On the Development and Application of a Photon Counting Imaging System for Broad Spectrum Luminescence Analysis

Iain McCulloch
May, 2014

A thesis submitted for the degree of Doctor of Philosophy of The Australian National University.

Volume I
Statement of Authorship

I hereby declare that, except where explicitly stated, this thesis including its appendices is entirely my own original work. Such exceptions are limited to Appendices A and N, and parts of Appendix L.

Iain McCulloch,
15 May 2014.
To my grandfather,
Charles Holt Smith
who showed me at an early age that learning was good,
but finding out for myself was even better,
and so set my life on a path of discovery.

And to my parents,
Don and Jill McCulloch
who stoically encouraged me on that journey,
and have done so ever since.

Whatever I may have achieved,
your steadfast love and support has made possible.
Acknowledgements

My sincere thanks go to the many people who contributed directly to the PCIS development, and to those who gave me personal support or assistance during the course of project. A few made such significant contributions that they require specific acknowledgement. I apologise to anyone whose contribution may have been overlooked.

First I must thank the caving community – the cavers, scientists and cave managers – who welcomed me into their unique tribe, and from the first moment treated me as family. You opened my eyes to the amazing world underground, and encouraged and helped me to explore and understand it.

Rob Palmer, cave diver extraordinaire, undertook the arduous task of training and qualifying me as a cave diver; and Mervyn "Nipper" Maher, cave dive instructor par excellence, spent the next year continuing my training, putting me in the water at every opportunity, until he believed that I was competent and safe in the enchanting but unforgiving world of the phreatic. I thank you both.

The Mount Gambier crew – Ian Lewis, for stimulating discussions on the regional geology, and for the extended use of his house; Kevin Mott, keeper of the maps, for freely sharing all his hard-won data; George Yarra, who while running an advanced cave diver's course around Mount Gambier took the time to show me most of the diveable caves in the area; and Peter "Puddles" Horne, whose remarkable work the Lower South East Cave Reference Book I value nearly as much as his friendship. I thank you all for making my trips to the Lower South East both enjoyable and productive.

Cave managers and cave guides everywhere have welcomed me into their domains, have taken the time to share their vast knowledge of their caves, and have always welcomed what little I could offer in return. Deserving particular mention for their co-operation and support are Steve Riley and the guiding staff at the Jenolan Caves; Steve Bourne, Caves Manager at Naracoorte Caves; and Trevor Wynyatt, the forestry officer with responsibility for many of the caves around Mount Gambier.

Many technical staff, at the Research School's workshops and elsewhere, contributed to the PCIS development, but three went far beyond any call of duty in their personal commitment to the work. I would like to thank Andrew Wilson of the RSES Mechanical Workshop; Daniel Cummins of the RSES Electronics Workshop; and Damien Jones of Prime Optics for their special contributions.

To my supervisors, Nigel Spooner and Brad Pillans, I offer sincere thanks. Your belief in me, and commitment to helping me through a long and difficult project, will not be forgotten. And to all the other academic, technical and administrative staff who did so much to help me – thank you.

For repeatedly proof-reading parts of my thesis and improving it enormously by their suggestions, my supervisors again, but also my father, Don McCulloch, and my ex-wife, Anne Ransley. Any remaining errors or omissions are certainly not due to any lack of care or diligence on their part.

I wish to thank all of my family for their support, but especially my children, who throughout their lives have paid an unfair price for my obsessions. I promise I will try to make it up to you.

And finally the caves, for entertaining me, for challenging me, for fascinating me – but above all, simply for always being there, patiently guarding their treasures and their secrets until I return.
Abstract

Novel luminescence equipment, the Photon Counting Imaging System (PCIS), with broad spectrum high sensitivity imaging capabilities, incorporating both thermal and multi-source optical stimulation, and providing flexible facilities for the optical filtering of signals, was designed and constructed.

Capable of high quantum efficiency, low-noise imaging from infrared to near UV (~ 1050 - ~ 200 nm), the PCIS is based on a liquid Nitrogen-cooled charge-couple device (LN-CCD) detector, and very wide spectrum, high photon capture optics. The three element expanded Offner reflective optical design avoids chromatic aberration and meets the spectral requirements, but demands extreme precision of manufacture and assembly. A highly rigid, adjustable and focusable optics assembly was designed and constructed, and special optical alignment tools and methods were developed and applied.

A bespoke programmable control unit (the IICU) was designed and built, providing independent numerical control of multiple optical stimulation sources, with programmable generic interfaces to support a wide range of ancillary equipment and experiment designs. Configurable control tables installed in the IICU define synchronised sequences of functions which implement a highly flexible range of basic luminescence operations. A PCIS hardware simulation environment supports the development and testing by experimenters of new IICU control tables.

Custom PCIS control software, with an informative graphical user interface, uses the IICU control table functions to implement sophisticated, high level luminescence protocols and experimental sequences. Flexible parameterised protocol templates have been developed for the Single Aliquot Regeneration (SAR) dating protocol using Optically Stimulated Luminescence (OSL), and for the exploratory broad spectrum analysis of materials with thermoluminescence (TL).

A developer's version of the PCIS control software is integrated with hardware and software PCIS simulators to provide a fully-functioned development environment for the construction and testing of luminescence protocols before samples are committed to the imaging system. A comprehensive, three-level logging scheme ensures that all runs, including simulator runs, can be fully audited and verified, and can be precisely repeated at any time.

Studies undertaken with the PCIS include analyses of quartz, Sodium Chloride (table salt), Calcium Fluoride, porcelain, mobile phone SIM cards, and samples of specialised glasses designed to detect ionising radiation. Future PCIS applications are surveyed, as are future developments of the PCIS, including re-development around an electron-multiplying charge-coupled device (EM-CCD) to give the PCIS upgraded temporal and spatial resolution, and improved low-signal sensitivity.

Initially intended in this project to address unresolved questions in karst geomorphology involving sedimentary processes and structures at the Jenolan Caves and Naracoorte Caves World Heritage Areas, the PCIS has proved to have great capability for wider applications in dating, sedimentary process studies, analysis of minerals and other materials, radiological event analysis, and for the direct study of luminescence, fluorescence, phosphorescence and tenebrescence phenomena. In conjunction with novel micro-sampling and OSL site survey techniques, the PCIS may enable entirely new methodologies, transforming the role of luminescence in archaeology, palaeontology, and Quaternary geomorphology.
Table of Contents

Statement of Authorship........................................................................................................... i
Acknowledgements.................................................................................................................... iii
Abstract ........................................................................................................................................ iv

1 Introduction................................................................................................................................. 1
  1.1 Background............................................................................................................................ 2
  1.2 Goals, Scope and Structure.................................................................................................... 20

2 PCIS Design and Development – Opto-Mechanical Components ............. 24
  2.1 Overview.................................................................................................................................. 24
  2.2 Inherited Components............................................................................................................ 26
  2.3 Selection of the Optical Design.............................................................................................. 35
  2.4 Broad Architecture of the PCIS............................................................................................ 42
  2.5 Broadband Optics Assembly ............................................................................................... 53
  2.6 Source Turret ........................................................................................................................ 102
  2.7 Shutter Housing .................................................................................................................... 153
  2.8 Assembly and Alignment of the PCIS.................................................................................... 158

3 Operation of the Phase 1 PCIS .......................................................................................... 167
  3.1 The Phase 1 PCIS Control Electronics .................................................................................. 167
  3.2 Phase 1 PCIS – Optical and Performance Testing ............................................................... 174
  3.3 Phase 1 PCIS - Control Architecture Investigations ............................................................. 196
### 4 PCIS Design and Development – Automation and Control

<table>
<thead>
<tr>
<th>Section</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1 System Integration Paradigm</td>
<td>204</td>
</tr>
<tr>
<td>4.2 Background to the IICU Development</td>
<td>210</td>
</tr>
<tr>
<td>4.3 The IICU Hardware Components and Interfaces</td>
<td>221</td>
</tr>
<tr>
<td>4.4 The IICU Command Table Structure</td>
<td>228</td>
</tr>
<tr>
<td>4.5 The IICU Management Interface</td>
<td>241</td>
</tr>
<tr>
<td>4.6 IICU V2 – Summary of Changes from the V1 Functional Specification</td>
<td>246</td>
</tr>
<tr>
<td>4.7 The IICU Command Tables Sets</td>
<td>251</td>
</tr>
<tr>
<td>4.8 The IICU Bench Testing Environment</td>
<td>262</td>
</tr>
</tbody>
</table>

### 5 Development of the PCIS Sequence Software

<table>
<thead>
<tr>
<th>Section</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1 Overview of the PCIS Automation Software</td>
<td>268</td>
</tr>
<tr>
<td>5.2 Sequence Templates for Standard PCIS Operations</td>
<td>270</td>
</tr>
<tr>
<td>5.3 Sequence and Include Files for Custom PCIS Operations</td>
<td>274</td>
</tr>
<tr>
<td>5.4 PCIS Information Logging</td>
<td>281</td>
</tr>
<tr>
<td>5.5 The PCIS_Seq Runtime Software</td>
<td>288</td>
</tr>
<tr>
<td>5.6 The PCIS Sequence Language</td>
<td>304</td>
</tr>
<tr>
<td>5.7 The PCIS Sequence Development Environment</td>
<td>313</td>
</tr>
</tbody>
</table>

### 6 PCIS Applications to Date

<table>
<thead>
<tr>
<th>Section</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1 Phase 2 PCIS Performance Testing</td>
<td>324</td>
</tr>
<tr>
<td>6.2 Optical Glass Studies</td>
<td>325</td>
</tr>
<tr>
<td>6.3 Analysis of Porcelain Slice</td>
<td>327</td>
</tr>
<tr>
<td>6.4 Table Salt (NaCl) Characterisation Studies</td>
<td>329</td>
</tr>
<tr>
<td>6.5 Opportunistic Retrospective Dosimetry with SIM Cards</td>
<td>331</td>
</tr>
</tbody>
</table>

### 7 Future Developments and Applications

<table>
<thead>
<tr>
<th>Section</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.1 PCIS Hardware developments</td>
<td>335</td>
</tr>
<tr>
<td>7.2 PCIS Software Developments</td>
<td>337</td>
</tr>
<tr>
<td>7.3 Remote Monitoring of the PCIS</td>
<td>341</td>
</tr>
<tr>
<td>7.4 PCIS Application Directions</td>
<td>342</td>
</tr>
<tr>
<td>7.5 Optical Upgrades and PCIS II – Sensitivity and Resolution</td>
<td>351</td>
</tr>
</tbody>
</table>

### 8 Conclusion

<table>
<thead>
<tr>
<th>Section</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>353</td>
</tr>
</tbody>
</table>

### References

<table>
<thead>
<tr>
<th>Section</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>357</td>
</tr>
</tbody>
</table>
Appendices

Appendix A  Optical Report: Damien Jones, Prime Optics
Appendix B  LED Installation Specifications: Iain McCulloch
Appendix C  Initial Analog Control Unit Design: Iain McCulloch
Appendix D  IICU Functional Specification: Iain McCulloch
Appendix E  IICU Command Table Set 2: Iain McCulloch
Appendix F  PCIS Sequence Templates Library: Iain McCulloch
Appendix G  PCIS Universal Includes Library: Iain McCulloch
Appendix H  PCIS Command Table-Specific Includes Library: Iain McCulloch
Appendix I  PCIS_Seq Language Reference: Iain McCulloch
Appendix J  PCIS Operations Guide: Iain McCulloch
Appendix K  PCIS Data Processing Specification: Iain McCulloch
Appendix L  PCIS_Seq Command Code: Iain and Brendan McCulloch
Appendix M  PCIS Simulator Command Code: Iain McCulloch
Appendix N  Minisys Low-level Command Specification: Risø Corporation
1 Introduction

A journey of a thousand miles begins with a single step.

"The Way of Lao Tzu"

Lao Tzu, Chinese philosopher.

This thesis describes the design and development of the Photon Counting Imaging System (PCIS) — a novel item of luminescence research equipment conceived and concept-tested by Dr Nigel Spooner (Spooner, 2000) but in the end, due to his unexpected departure from the Institution and therefore in a day-to-day sense from the project, primarily designed and constructed by myself. The thesis describes the emergence of the design from the goals and constraints present at commencement; the detailed design, construction and testing of the components of the PCIS; the integration of the components and development of the electronics and software to control and manage the PCIS; and the final form of the PCIS — its capabilities, and how to program and operate it to achieve the best performance and results.

A number of applications that the PCIS has performed to date are described. These derive in large part from runs where the primary goal was testing of the PCIS, and they therefore represent different stages of its development. Nevertheless, some of these applications produced highly useful results, and several publications ensued (McCulloch et al, 2011; Spooner et al, 2011; Oks et al, 2011).

Finally some future directions are investigated. Some pertain to future developments and improvements to the capabilities of the PCIS itself; others pertain to novel applications that the unique capabilities of the PCIS render open to investigation. They include software tailored to the PCIS to support the visual selection of sub-sample areas, and processing of their data into a form compatible with palaeodose estimation utilities; and the development of micro-sampling and minimal-chemical-processing techniques.
1.1 Background

This PhD project arose from the serendipitous confluence of several factors – my personal situation allowed me to commit myself to an interesting and challenging development project; the initial success of Dr Spooner’s work in luminescence imaging meant that there was an important technical development in need of someone with the time, skills and motivation to complete it; and there were questions in karst geomorphology requiring a novel approach, such as the PCIS, to address them. These background factors, and how they came together, are described below.

1.1.1 Personal Background

The origins of this project – for me – lie deep underground. Mid-life crisis had led not to a red sports car, but rather to a less glamorous (but equally fanatical) devotion to caving. Virtually living in the Jenolan Valley in eastern New South Wales, first vertical caving and then cave diving expanded my range of underground explorations. Finally, a fascination for the development and history of the caves (and a lack of ready and convincing answers) led me to uncertake a Graduate Diploma in Geology, studying karst geomorphology under the invaluable guidance of Professor John Chappell.

Involvement in the project to dive, explore and map the underground rivers at the Jenolan Caves (during which I had the honour to make the first through trip from the southern show cave complex to the outside Blue Lake via River – Styx) led to the observation that sediments in the rivers displayed quite different behaviours at different locations. The phreatic environment was clearly capable of performing the most remarkable feats of sediment sorting and imbrication (on scales from clay to pebbles), and while some underwater features, when dug through, would then remain open (e.g. The Long Low Flat Horrible Thing upstream from Twin Bridges), others (such as the 36 m constriction in Slug Lake, Mammoth Cave) would refill themselves and need to be dug afresh on each visit.

Some little thought led to the realisation that a cave stream, particularly in a phreatic setting, would inevitably behave very differently to an unconstrained surface stream. The question of how sediments move through a cave – and the related question of how to study such phenomena – began to pre-occupy my thinking. It occurred to me that if one could establish the statistical distribution of grain “ages” for a number of locations along a cave stream (where “age” represents the time since that grain entered the cave system from above ground), then it should be possible to deduce more than was presently known about the patterns of movement of sediments in phreatic settings (a subject which for practical reasons had received little study).

And so I attended a seminar presented by Dr Nigel Spooner on luminescence dating, which my very little knowledge had encouraged me to hope might be of some assistance in my quest. (A course at undergraduate level addressing briefly the applications, capabilities and above all the limitations of various dating techniques would, I believe, be of great value.) At the conclusion of the seminar I questioned Dr Spooner about the application of luminescence techniques to the active cave stream setting, and his reply was somewhat equivocal; luminescence techniques could in principle address the problem, but the current generation of luminescence dating equipment could not.
1.1.2 Luminescence Project Background

Dr Spooner explained that he was in the process of developing imaging equipment for luminescence – the Photon Counting Imaging System (PCIS) – and that one thing such a machine would be capable of was large scale single-grain dating applications such as I was proposing.

The design criteria set for the PCIS was that it be capable of performing the most difficult challenge yet defined in luminescence dating – dating single zircon grains using the "auto-regeneration TL" technique (Templer, 1986). This method requires an initial TL measurement of the zircon natural TL, followed by approximately 6 months storage in a dark low-radiation environment to enable TL regrowth from dose by the internal alpha particle irradiation – "auto-regeneration TL". The age is the ratio of these two TL signals; the auto-regeneration TL is not only extremely faint but has an emission spectrum exhibiting peaks spread across the UV and visible wavelength range, emitted from various rare earth recombination centres.

Hence the PCIS optics would ideally collect a large fraction of the emitted photons regardless of wavelength from each grain of an array of zircons, and transfer them to create a grain array image on the LN/CCD chip; this requires the optics to be both achromatic and to have the largest feasible numerical aperture. Spatial resolution is a lesser consideration, requiring only that the light emitted from each grain is unambiguously projected onto its corresponding location in the array image.

Detector selection was based on the specifications of the PI back-illuminated LN/CCD camera in comparison to other types of detector available in the late 1990’s: these were position sensitive photomultipliers, micro-channel plate image intensifiers and intensified front-illuminated CCD cameras. The choice of LN/CCD camera remains valid today, even since the advent of electron multiplication CCD (EMCCD) cameras. The major apparent advantage that an EMCCD has over a CCD is that the amplification stage (on-chip gain) allows better S/N at very low photon fluence (< 10 photons/pixel/readout), but the gain process introduces an additional stochastic noise term which degrades the utility of the EMCCD for high quality quantitative data collection. Basden et al., 2003 suggest a strategy to reduce this effect to insignificance, but only through the means of operating at high frame rates, a generally undesirable mode for luminescence applications due to the needless expansion of the data stream this entails.

O’Grady, 2006 assessed the suitability of EMCCDs in comparison with CCDs, comparing sensitivities, dark signal, noise factors and readout noise with regard to their impact on detection limits, and concluded that traditional CCD cameras have superior detection limit and equal sensitivity to EMCCDs for low light level applications where high frame rates are not required. More recently, Joubert and Sharma, 2011 have compared EMCCD cameras with the newly-emerging Scientific-Grade CMOS sensors, and concluded that the former are significantly superior for extreme low light level sensing, such as is required of the PCIS.

Although other designs and technologies had been used for imaging luminescence in the past (Huntley and Kirkey, 1985; Duller et al, 1997; McFee, 1998), none of these had achieved the combination of spectral range and sensitivity needed for the most demanding tasks planned for the PCIS, notably single-grain optical dating and zircon auto-regeneration TL dating. Nor had any been capable of performing as broad a range of luminescence investigations as the PCIS.
Several years after the commencement of the PCIS project a series of publications appeared from the Heidelberg group, Germany, describing a CCD camera-based luminescence imaging system. The first of these articles, Greilich et al., 2002, reported the detection system to be a Princeton Instruments LN-cooled CCD spectrographic camera, optically interfaced to the luminescence reader using a Olympus Zuiko 50 mm f 1.4 camera lens. These optics limited its light collection and spectral transmission performances to closely resembling those of the prototype PCIS as described in Spooner, 2000, and hence significantly inferior to those of the completed reflective optics PCIS.

The Heidelberg system was subsequently proposed for application to the dating of granitic stone surfaces (Greilich et al., 2005) and then to enable introduction of a more general method for spatially resolved dating of surfaces (Greilich and Wagner, 2006); it was supported by the development of a software program “AgesGalore” for evaluating spatially resolved luminescence data (Greilich et al., 2006), following which no further publications appear to have been produced.

More recently, three groups have presented imaging systems based on EMCCD cameras interfaced to automated TL/OSL readers, either Risø machines or the new Lexsyg system (made by Frieberg Instruments GmbH, Germany).

The Oxford, England group, described a luminescence imaging system based on an EMCCD interfaced to a Risø reader and using refractive optics (Clark-Balzan and Schwenninger, 2012). The lens assembly consisted of two fused silica, anti-reflection coated, plano-convex lenses each of 25 mm diameter and 35 mm focal length, giving f 1.4 and 0.37 Sr solid angle collection, just under half that of the PCIS.

The Lexsyg system (Richter et al., 2013) has as its imaging detector a Princeton Instruments ProEM:512B EMCCD camera, which is used in CCD mode for lowest noise. An unspecified optic assembly having numerical aperture 0.5 (close to f 0.9) is provided for the UV waveband; several other optic assemblies were also developed in order to utilise the spectral detection range provided by the EMCCD, but details are not provided.

The most recent of these systems appeared in the literature subsequent to the submission of this thesis; that of Kook et al., 2013, described as a “high sensitivity imaging attachment to the Risø TL/OSL platform”. It includes a Peltier cooled Evolve EMCCD camera (Photometrics) optically interfaced to the reader using AR-coated fused silica refractive optics, and with performance very similar to that of the Oxford unit of Clark-Balzan and Schwenninger, 2012 on which it appears based.

At the time of writing, no access has been possible for direct comparison of the performance of the PCIS with any of these three most recently disclosed systems, but the published data on them is sufficient to permit some broad assessment. All three systems have considerable functional similarity, an inevitable result of the choice of all three groups to utilise refractive rather than reflective optics. The three EMCCD cameras each have superior QE and lower readout noise compared to the PCIS LN/CCD, but this advantage is fully offset by the greater solid angle collection efficiency of the PCIS reflective optics, therefore in any given waveband the four systems are closely equivalent in detection efficiency.

However the PCIS has two major advantages - it remains uniquely capable for broad spectrum imaging, and its reflective optics enable achromatic imaging at the highest numerical aperture of any imaging system designed for luminescence analysis.

In comparison with a conventional PMT-based reader, measurement has shown that the PCIS has
approximately 1.5 times the sensitivity in the UV of the Risø Minisys when equipped with a bialkali PMT, the advantage accruing from the comparable solid angle collection of the PCIS optics combined with the greater quantum efficiency of the LN/CCD camera in that waveband. The PCIS advantage grows rapidly with wavelength through the visible and near-IR (NIR) wavebands due to the increasing relative QE of the LN/CCD camera to that of the Risø bialkali PMT at longer wavelengths; this advantage is approximately a factor of 100 by 700 nm, and far greater in the NIR.

At the time I commenced work on the PCIS project, the state was as follows:

- A Risø Minisys TL/OSL-DA 15 had been acquired, capable of both thermo-luminescence (TL) and Optically Stimulated Luminescence (OSL), using a standard photomultiplier tube for photon counting in the mid to short wavelength bands.
- A very high sensitivity liquid nitrogen-cooled charge-coupled device (LN-CCD) of 512 by 512 pixels had been acquired, along with its controlling electronics and standard software suite.
- A pair of standard 50 mm focal length, f/1.2 camera lenses had been mounted “back-to-back” to provide 1 to 1 imaging at approximately f/1.4 for the visible spectrum only.
- The above components had been assembled, and TL measurements had been made on a variety of materials. Signals emitted in the red part of the visible spectrum were recorded (and reported in Spooner, 2000), demonstrating that the LN-CCD had the sensitivity required to record meaningful images from just the photons emitted by the luminescence process.
- A design had been commissioned from Damien Jones of Prime Optics for an optical system to provide 1 to 1 images, at very wide aperture (low equivalent f-number), across a spectral range from IR (~1050 nm) to near UV (~200 nm). A reflective “expanded Offner” design was selected in order to achieve the required performance. (The design appears at Appendix A.)

Clearly not recognising the full implications of my actions, I then offered to “help finish building the machine”, if I could then have use of it for my cave sediment studies. Dr Spooner, with perhaps only a slightly better understanding of the implications than I had, quickly accepted the offer. Accordingly I was accepted as a visitor at the Research School of Earth Sciences, and commenced work on the design of the complete broad spectrum PCIS, starting with the reflective optics.

Shortly after commencing work on the PCIS project I was invited to undertake a PhD, combining development of the PCIS with a study of the fossiliferous sediment cones in caves of the Naracoorte Caves World Heritage Area in the south-east of South Australia. (Fossil Chamber in Victoria Fossil Cave contains what certainly was then, and may still be now, Australia’s premier megafauna fossil deposits.) I accepted the invitation, albeit (for medical reasons) on a part-time basis.

The understanding was that I would spend around half my time assisting with the PCIS development project, and the remainder continuing my karst and cave studies. In fact, with the early departure of Dr Spooner and the emerging scale and complexity of the PCIS development, less time was spent on the karst studies, and while that work has led to some conclusions (McCulloch, 2009), and further publications are planned, the karst studies do not form a significant part of this thesis. In the end my efforts were focussed totally on the PCIS project, and that focus is reflected herein.
1.1.3 Karst Geomorphology Background

The PCIS was never intended as a single-application development; indeed, great pains were taken to make its capabilities as broad as possible. Nevertheless, the proposed single-grain quartz OSL dating study of the fossiliferous sediment cones at Naracoorte Caves, South Australia, provided a focus both for the PCIS development goals, and for my broader studies of cave and karst geomorphology. This section introduces some peculiarly Australian issues in karst geomorphology, outlines the kinds of karst geomorphological problems which the PCIS might help to address, and describes in detail the proposed study site at Naracoorte Caves.

1.1.3.1 Karst Geomorphology Overview

It is apparently appropriate that I provide at this point a general overview of, or introduction to, karst geomorphology. I will attempt to do so briefly, focussing on the geomorphology of Australian karst, since it has become clear that the traditional literature describing European and American karst concepts cannot be simply transposed onto Australian karst systems.

First, it should be noted that karst geomorphology refers to all types of karst features, from the surface drainage patterns characteristic of entire karst landscapes down to such small-scale features as the various classes of karren; however, I shall focus, as do most karst-related publications, on caves, as being the karst features generally of greatest interest. I shall adopt the speleologist’s definition of a cave; that it must be enterable by a human, and must have a dark zone.

Karst geomorphology may be studied in one of two ways: using a regional, or “top down” approach; or using a process-oriented, or “bottom up” approach. The former approach tries to interpret a cave or, more usually, a karst area in the context of the regional geomorphic history, relating the karst features and their imputed times (and modes) of development to regional geologic and climatic history, with the aim of achieving a better understanding of that regional history, as well as a deeper understanding of the caves themselves. Clearly such an approach must be region-based, and no global (or even national) overview is practical.

Process-based approaches to karst geomorphology attempt to determine the particular physical, chemical, and biological processes that have contributed to the development of karst features, sometimes on a cave-wide or even regional scale, but more usually relating different processes to separate (and often quite small) features within a cave or cave system. This approach, although limited in its import with respect to events outside the caves, at least has the potential for more general – even in some cases global – applicability.

True progress in our understanding of the caves of a region occurs only when regional approaches are synthesised with process-based approaches to produce a complete picture of the history and development of the cave system. Such syntheses are, as our mathematical friends would say, non-trivial, and in my view no completely successful synthesis of Australian karst has yet been achieved.
1.1.3.2 Region-based karst geomorphology

It is impossible to study (let alone understand) a region’s caves until those caves have been found, explored, surveyed and mapped — tasks overwhelmingly undertaken by “amateur” cavers and rarely by professional karst geomorphologists. Such work, which often takes thousands of hours, would be prohibitively expensive if performed professionally (not to mention the OH&S nightmares that would arise), generally leads to the production of high quality surveys, maps and publications describing a particular region and its caves. (Cavers love maps, and the map itself often provides the only acknowledgement of their association with the exploration of the cave. Surveys are conducted to incredible standards of accuracy, given the conditions, and the detailed maps, including a wide range of accurately located in-cave features, are often works of art as well as of science.)

These caver publications are however in the nature of a “karst feature inventory” and, although often extending to the secondary mineralogy and biology of the caves, do not usually explore the geomorphology or processes that have produced the features; rather, such works (and especially the cave maps invariably included) are a necessary pre-cursor to a serious study of the regional karst geomorphology. Some of these descriptive works relating to the principal karst provinces of Australia are reviewed below.

Australia must be considered as at least three distinct provinces in karst geomorphology terms, with little commonality between them with respect to the cave systems and operative processes (see Figure 1.1.1 below). These are (1) the tropical North; (2) the East coast (including Jenolan) and Tasmania; and (3) the southern and western regions, including Naracoorte and the Nullarbor. I shall not discuss the tropical North any further, as I have no personal experience of the caves of that region, most of which are anyway still in the “karst inventory” phase. Work on such caves is further complicated by their geographic remoteness, and the highly seasonal nature of the climate limits visitation to particular times of year. (For example, the Bullita cave system in the Northern Territory, the most substantial such system with over 100 km of surveyed passage, is visited only for a single trip in July each year, and is only now emerging from the “scooping booty” phase.)

The East coast and Tasmanian karsts are characterised by development over very long periods of time, in small, isolated (“impounded karst”) exposures of extremely hard, massive and often extensively folded limestones of very low porosity and, generally, permeability. (Only Mole Creek in Tasmania represents a significant expanse of karst, at approximately 10 km by 25 km. It is the only Australian region which expresses true karst-type surface drainage patterns, and is properly comparable to karst regions studied in Europe. Other large Australian karst areas — in the South and West province — are too arid to show such karst-type surface expressions.)

Although very much better studied than the karsts of the tropical North, the Eastern Australian karst areas generally have very complex histories and are not necessarily much better understood. “Inventory” publications produced by the caving community exist for a significant number of these areas. These include “Caves of Jenolan, 1: The Exploration and Speleogeography of Mammoth Cave, Jenolan” (Dunkley and Anderson, 1971 — usually called “The Yellow Book”); “The Caves of Jenolan 2: The Northern Limestone” (Bruce Welch (ed), 1976 — usually called “The Blue Book”); “Wombeyan Caves” (Dyson, Ellis and James, 1982); “Caves and Karst of Wombeyan” (Ross Ellis (ed), 2004); “Bungonia Caves” (Ellis et al (eds), 1972); “Under Bungonia” (Bauer and Bauer, 1998); “Wee Jasper
Development and Application of a Photon Counting Imaging System

Caves” (J. N. Jennings, 1985b); “Wee Jasper Caves” (Dunkley, Spate and Welch, 2010); “Tuglow Caves” (Cooper, Scott and Vaughan-Taylor, 2006); “Timor Caves” (Rutledge, Smith, Brainwood and Baker, 2008); “Tasmania: Tasmanian Cave Exploration in the 1980’s, Volume 1” (Hume, Nicholas and Wailes (eds), 1992); and “Caves of Gunns Plains” (Blanden, 2004). Note that most of these excellent works are produced and published, generally at a loss, by amateur cavers and their clubs, often with the assistance of the Australian Speleological Federation.

The best Australian synthesis of regional and process-oriented karst geomorphology exists for this East coast region, and resides in the many publications by Armstrong Osborne (e.g. Osborne, 1999; Osborne, 2007). The most recent, and the most “synthetic” of these papers, is “Rethinking eastern Australian Caves” (Osborne, 2010).

The southern and western Australian karst areas are characterised by relatively large (in the case of the Nullarbor, extremely large) areas of horizontal or sub—horizontal limestone (whether marine-laid or aeolian), of extremely high porosity and permeability. The limestones themselves, their patterns of outcropping and the caves that develop within them are starkly different to those of the eastern province. These southern and western province limestones are both porous and permeable; they are also very soft, and break-down becomes an important process in the history of these caves.
The best “inventory” works for the southern and western regions are Peter Horne’s excellent “Lower South East Cave Reference Book” (Horne, 1993) – which describes some 400+ caves in the Mount Gambier area of South Australia; Ian Lewis’s “South Australian Cave Reference Book” (Lewis, 1976) and “Discover Naracoorte Caves” (Lewis, 1977); the “Nullarbor Caving Atlas” (Pilkington and Mott (eds), 1982); “Caves of the Nullarbor” (Dunkley and Wigley, 1967); “Mullamullang Cave Expeditions 1966” (Hill, 1966).

Further information on all of Australia’s karst regions and their caves exists in the many journals published by caving clubs, in the clubs’ own records, and now in the Karst Information Database (“The KID”) maintained by the caving clubs and the Australian Speleological Federation. The KID supplants the earlier (and now out of print) “Australian Karst Index 1985” (Matthews, 1985). The abundance of caver-produced “inventory” works for Australia’s cave areas compares favourably with the number of effective syntheses that have been produced, and suggests that the professional scientific community is lagging well behind the efforts of “amateur” cavers. The extent to which this is due to the low priority given to karst studies in Australian science funding is open to argument.

1.1.3.3 Process-based karst geomorphology

Process-based karst geomorphology examines the individual physical and chemical processes responsible for the production of particular karst features which may appear in many different cave settings. Publications on karst processes therefore lack the strongly regional flavour of the earlier described publications, and may have a quite broad – even international – relevance.

Despite the broader applicability of process studies, there is still the risk of assuming that a process determined in one place is responsible for similar-looking features elsewhere; this may or may not be so, and needs to be determined in each case. That said, there is one process which dominates in the formation phase of a cave’s history, and that process is mixing corrosion.

All karst process publications mention and describe mixing corrosion, but fail to ascribe to it the dominant role that it actually plays in the formation (if not necessarily in the later development) of caves everywhere. I have personally found it useful, in determining the unique characteristics of a particular region’s caves, to look for those features which appear not to be the result of mixing corrosion and to attempt to understand them. Few other generalisations are possible in a field where variety is ubiquitous, and the devil is indisputably in the detail.

Useful works which address karst geomorphology from a process perspective include “Processes in Karst Systems: Physics, Chemistry and Geology” (Dreybrodt, 2011); “Karst Hydrogeology and Geomorphology” (Ford and Williams, 2007); and “Cave Geology” (Palmer, 2007). However, none of these can fairly be described as introductory works, nor are any specifically Australian in outlook.
1.1.3.4 Australian Karst Syntheses

The “short story” of cave morphology is that caves form due to dissolution of the soluble rock (usually limestone), predominantly due to the process of mixing corrosion (see Dreybrodt, 19xx among other references). An often long process of development and change in the cave ultimately terminates in one (or sometimes both) of two ways; either the cave fills completely with allochthonous sediments or secondary minerals (almost invariably calcite); or the cave “bubbles to the surface” via stoping or similar processes (possibly assisted by regional surface lowering). When either process completes, there is no longer a cave. (Karst geomorphologists will disagree, arguing for a “palaeokarst cave” or a “cave without a roof”, respectively. Cavers deny the existence of such things – or at least, that such things are caves.)

Karst syntheses attempt to detail these development phases, and explain the operation of each in terms both of the karst processes that have been operative, and the regional conditions applying at the relevant times. Early Australian karst syntheses (including Jennings (1971, 1985a) – and, disappointingly, at least one work published as recently as 1996) suffer from attempts to graft descriptions from the classical overseas literature onto examples from Australian caves, sometimes comprising an uncomfortable juxtaposition of a borrowed (European or North American) diagram and a photo from an Australian cave, with little or no attempt to justify the applicability of the former to the latter. More recently there has been a realisation that Australian karst and conditions require their own analysis of the operative processes and their relation to the resultant cave and karst systems.

Although early, and suffering somewhat from the above problem, Joe Jennings’s “Karst” (Jennings, 1971), later revised and re-published as “Karst Geomorphology” (Jennings, 1985a) remains the best broad synthesis actually based on Australian caves and karst areas, most if not all of which Joe actually visited before speculating as to their histories. Although, like Douglas Adams’s “The Hitchhiker’s Guide to the Galaxy”, it “contains much that is apocryphal, or at least wildly inaccurate”, it represents a genuine attempt to come to an understanding of uniquely Australian karst environments and processes, and their relationships in real Australian cave settings. As such it sets a reference against which all later work is judged. Thus, despite its now-apparent faults, it provides a necessary basis for discussion – much as Darwin’s work on evolution and Freud’s work on psychiatry have done. Each is seminal and important despite its evident yet excusable shortcomings.

With particular reference to the Nullarbor (which had been little studied prior to his time), Jennings’s “A Preliminary Report on the Karst Morphology of the Nullarbor Plains” (Jennings, 1961) provides an interesting, and again in many ways seminal overview of this most important karst region.

A later work, suffering less from Euro-centricity, is “Beneath the Surface” (Finlayson and Hamilton-Smith, eds, 2003). Although presenting a less coherent view than Jennings (being a compilation of chapters from a variety of authors), it has the advantage of a further 40 years work on Australian karst processes and histories. Chapter 1 in particular (“Black Holes: Caves in the Australian Landscape” by John Webb, Ken Grimes and Armstrong Osborne) provides a modern overview of cave development in the different karst regions and limestone types of Australia.
1.1.3.5 Other Useful Karst Publications

Many regional karst inventory publications include sections on the biota of the caves they describe, often representing significant programs of collecting and identifying samples. However, they obviously have only a regional relevance, and do not purport to provide an overview of cave biology generally. Two books which do provide such an overview are “Cave Life” (Culver, 1982) and “Caves and Cave Life” (Chapman, 1993). Also of interest in this context is “Dark Life” (Taylor, 1999), presenting a lay perspective on the current state of knowledge of and research into extremophiles — many (although not all) of which are cave forms.

The “Encyclopedia of Caves and Karst Science” (John Gunn, 2004), although not an introductory or overview work, covers both regional and process-oriented aspects of karst geomorphology, and is a useful reference work for any karst researcher. (It also offers the valuable and interesting insight that while the consumption of cave salamanders in conjunction with alcohol is fatal in 50% of cases, the toxic effects on humans of cave salamanders when not combined with alcohol are unknown).

The standard reference work on cave mineralogy is “Cave Minerals of the World” (Hill and Forti, 1997). This is an indispensable resource for karst geomorphologists, and is the only reasonably comprehensive global survey of the many minerals found, and the many found only, in caves.

Finally, an interesting review of the development of theories of karst geomorphology — and cave development in particular — over the last 2,000 years or more is provided by Trevor Shaw in “History of Cave Science” (Shaw, 1992). This fascinating work makes one wonder how our own efforts might be viewed by the readers of future editions!

1.1.3.6 The Proposed Project Study Site

The main study site proposed for testing and proving the capabilities of the PCIS was an exposed sediment wall in a non-fossiliferous sediment cone in Alexandra Cave at Naracoorte Caves — on the assumption that it formed a good proxy for much more restricted sites such as the megafauna-containing sediment cone in Fossil Chamber, Victoria Fossil Cave. This assumption I later came to question. The wall in Alexandra Cave was created by excavations (in the 1930’s or earlier) to open the cave for tourism, and is easily accessed for sampling. A 4 or 5 metre section of the (30 or 40 metre long) wall is shown below at Figure 1.1.2.

But just as the emerging capabilities of the PCIS led to a far broader range of applications than single-grain quartz dating (see Chapters 3, 6 and 7), so my interest in Australia’s regions of soft, porous, structurally undisturbed limestone — including Naracoorte, the Mount Gambier area, the Nullarbor and Roe Plains and the south-west corner of Western Australia — became much broader than the occurrence and structure of the sediment cones that their caves contain. As Professor Chappell had promised, these caves were very different to those developed in the ancient, massive and highly folded limestones of eastern Australia (Osborne, 2010).
That said, extensive study of large numbers of these caves and their sediment cones eventually led to the development of a hypothesis about the structure of the cones - a hypothesis which could be tested with very high grain count, single-grain quartz OSL dating, using the PCIS.

The predominant feature of the Alexandra Cave sediment cone, apparent in Figure 1.1.2, is the succession of distinct and near horizontal strata seen in the lower parts of the wall. (I had cleaned the entire wall with a paint brush to enhance visibility prior to taking these photos.) By contrast, the upper parts of the wall show a significant degree of re-working and disturbance. The parts of the sediment cone that I have examined in Fossil Chamber, Victoria Fossil Cave, show similar structural disturbance. (I did not have permission to sample, or even to clean exposures in Victoria Fossil Cave.)

So why might the Alexandra Cave sedimentary structures not be representative of the megafauna-containing cone structures seen in Victoria fossil Cave and elsewhere? For a start, there may be good reasons why some caves contain megafauna fossils, but Alexandra cave does not.\footnote{It is of course possible that Alexandra Cave entrance simply wasn't open at the right time, when megafauna were roaming the regional landscape falling into things; but other explanations also deserve consideration.} Alexandra Cave may not have had the "pit-fall trap" type of entrance that is usually associated with fossiliferous cave deposits. Although the entrance has been significantly modified by tourist infrastructure, evidence around the artificial entrance suggests that the original entrance may have been smaller, and under a rock shelf. Unfortunately I have found no pre-development photographs of this entrance.

Small rodent and marsupial bones have been found in the Alexandra Cave sediments, and this is also consistent with a cave that birds and bats could enter, but megafauna could not. The entrance by which Victoria Fossil Cave was discovered is just such an under-shelf entrance (close to the present...
artificially-cut tourist entrance), and there are no megafaunal remains found anywhere in that area of Victoria Fossil Cave. (The entrance by which megafauna entered the Fossil Chamber has long been closed by sediments accumulating to the ceiling of the cave, and then to the surface land level.)

It is apparent from Figure 1.1.3 below that the sedimentary strata in the lower parts of the cone in Alexandra Cave are closer to horizontal than they are to the slope of the “sides” of the cone. (The photo is taken from the “far” chamber, and the floor is sloping down towards the viewer.)

![Figure 1.1.3](image)

Figure 1.1.3 Horizontal strata in ~2 m high cut section of sediment cone, Alexandra Cave.

On close inspection, it appears that many of the lower strata in the Alexandra Cave sediments may have been water-deposited, and the nature of the rest of Alexandra Cave supports that idea. The cave extends down a gently sloping passage into a larger chamber with a flat mud floor, and at least two pools. A small dug extension at the end of the cave suggests water-borne sedimentary infilling, and reveals no lower level that would have quickly drained water away. The two pools, although now artificially maintained, are in the place of older natural pools which appear in many old photographs (1930s and earlier) in my private collection. There is every indication that Alexandra Cave has been significantly flooded at some past periods, reworking the contained sediments. Alexandra Cave was starting to look a lot less like Fossil Chamber than it had initially seemed.

Then a chance encounter at a conference lunch created an opportunity to spend some weeks in Mount Gambier, 100 km south of Naracoorte, first being guided around most of the diveable caves in the area by a senior instructor for the Cave Divers Association of Australia; and then, armed with comprehensive maps of the area and the caves (Horne, 1993), an open access permit, a compass, GPS, rope and vertical gear, I was able to make a leisurely study of a great number of the smaller caves of the region. These caves contain large numbers of sedimentary cone structures – but of a significantly different character to those in the caves around Naracoorte.
The sediment cones in what I call the “corrasion tube” caves of the Mount Gambier area tend to be much steeper – much closer to a reasonable repose angle for the sediments – and often reach the ceiling of the cave thus blocking their own entry hole. In the few places where parts of their internal structures are visible, they appear to be less disturbed than the broader cones, and to have strata more nearly parallel to their sides. I began to consider what differences there may have been in the environments the cones had formed in; and why the broader cones typical of Naracoorte (especially the fossiliferous examples) should apparently have suffered so much more structural disturbance than the steeper cones which are typical of the Mount Gambier region.

Two observations in particular provided pointers to possible answers – weather events, and disturbance by animals. We shall first consider the effects of severe weather events.

During a visit to Naracoorte Caves on my way back from Mount Gambier, it was suggested that I look at the “Tomato Cave” (U11) entrance to the Wet Cave (U10, U11) system. Here the “floor” of the cave at the large (10 m plus diameter) doline entrance is actually the top of a very broad (30 m plus radius), gently dipping sediment cone. A heavy storm a few days earlier had carved a channel into the sediments, of the order of half a metre wide and deep. I followed this channel for approximately 30 metres into the cave until it disappeared into a hole behind a large boulder. The channel did not appear to be following earlier erosional channels, but rather to be arbitrarily cross-cutting existing sedimentary structures. The total amount of material transported by this one event was quite large – in the region of 5 to 10 cubic metres.

Now no-one is suggesting that the megafauna were particularly stupid or clumsy, and it is likely that in order to collect significant deposits of their fossils, a cave entrance needs to be large, and open for a long period of time. The cumulative effects of sub-aerial weathering can therefore reasonably be expected to be particularly evident in the fossiliferous sediment cones.

The second observation concerns an effect that would only occur in the fossiliferous cones, since it is caused by the previous owners of the now-fossilised bones themselves. Figure 1.1.4 below shows the skeleton of a modern kangaroo that was caught in a floor slot in Morgan’s Cave near Mount Gambier. Deep scratch marks, consistent with having been produced by the claws of the kangaroo, can be clearly seen around the edges of the slot. Similar damage has been observed at other locations where an animal would encounter rock while trying to get to the cave’s nearest entrance (or, from the animal’s perspective, exit). Loose sediments neither resist nor preserve such marks as well as rock, so one can only imagine how much more damage a desperate animal would cause to the top of a sediment cone under a classic pit-fall entrance.

---

2 By my observations underground, the blocked entrances significantly outnumber the open ones. Curiously, this creates the possibility of estimating an answer to the classic tourist question, “How many undiscovered caves are there in this area?” By plotting the number of open corrasion tubes against the number of blocked holes, and using luminescence to estimate the age and duration (i.e. time from formation to blockage) of the blocked holes, it should be possible to estimate the number of caves with one or more blocked entrances, but no unblocked entrances. Since the landscape makes caves with unblocked entrances very easy to find, the number of unknown caves is just the number with only blocked entrances. The other classic tourist questions – “Is it dark in the caves?” and “What time is the 3 o’clock tour?” – are less amenable to luminescence analysis.
Based on the totality of these observations, I formed the hypothesis that there were two distinct classes of sediment cones to be found in these caves.

1. Large, broad, gently dipping cones, occasionally containing megafauna fossils. These generally have severely disturbed internal structures, which would make it difficult to obtain reliable OSL dates from the sediments, or to establish the age of the contained fossils.
2. Smaller, steeply dipping sediments cones, which do not contain large fossils, but whose undisturbed structure would render them more susceptible to OSL analysis and dating.

This hypothesis is well supported by field observations; its determination would have interesting implications for megafaunal studies in Australia; and it might be rigorously tested using massive grain count single-grain sediment dating with the PCIS.
1.1.3.7 Problems and Difficulties in Karst Geomorphology

It has been seen that karst geomorphology is a broad and complex subject. However, for various intrinsic reasons it is also an extremely difficult subject to study. A cave – a hole in the ground – is essentially an absence – the rock that isn’t there (hence the classic answer to “Why do you go caving?” – “Because it isn’t there.”) Fundamentally, deriving the history of something that is not there has certain obvious difficulties associated with it. The necessary resort is to study the things that now are where the rock used to be. The air – and any water contained in the cave – are likely to be very recent and disclose little of the cave’s early history. One is left with little left to work on but

(a) the shape of the resultant cave, and
(b) the nature and properties of any solids filling the cave (i.e. those sediments and secondary minerals that are busy trying to make it no longer a cave at all.)

Unfortunately, the latter badly disguises the former, to the extent that I generally refer to calcite deposits – the “pretties” for the sake of which many people enter the cave in the first place – as “cave cholesterol”. There is little of benefit in it, but it blocks (and, in the modern conservation-focussed age) prevents entry to or knowledge of, much of the distinctive dendritic shape of the cave.

This leaves for study only the deposited sediments and the secondary mineral deposits. Where secondary mineral deposits – especially calcite – exist, and can be related directly by stratigraphic principles to the geomorphic events of interest, these can be easily dated by radio-isotope techniques, and much information may be derived.

Unfortunately (or, in light of the “cave cholesterol” comment, fortunately) many caves contain little, if any, calcite. It is a fortunate and unusual event indeed when a piece of calcite has clearly formed in a location, and at a time, that would be useful to determining the geomorphic history of the cave. (Note these difficulties do not apply – or at least not in the same degree – to palaeoclimatologists masquerading as karst geomorphologists, as the calcite they study – i.e. date – does not need to provably relate to a particular aspect of or event in the cave’s geomorphic history.)

Furthermore, “stratigraphic principles” (like OH & S rules) are honoured more in the breach than the observance once underground. Sediments do not deposit from bottom up (especially in water), but may attach to any appropriate surface – wall, floor or ceiling – with, like McCavity (the mystery cat), little appreciation for the laws of man or gravity. Sediments may also erode from the bottom up, and later refill the cavity (from any direction), making any concept of a “stratigraphic sequence” at best dubious, and at worst highly misleading. Above ground the significance of sedimentary events are often derived from an assumed sequence of deposition; in a cave the actual sequence of sedimentary events must be determined before the sediments’ significance can be determined. This leaves dating of the sediments in the cave as an essential component – if not in practice the only applicable technique – for determining the geomorphic history of the cave.

When we speak of “dating” the sediments in the cave we are of course referring to the date of deposition of the sediments in the cave, and not to the date of formation of the sediments. (In very rare cases sediments – especially clays – may form in the cave, and then their date of formation may have some relevance; but such cases are rare, and are in any case better treated as cases of
secondary mineral formation, rather than as sediment deposition events.) Dates of formation of (many) minerals may be measured to good accuracy using a wide variety of conceptually simple radio-isotopic decay techniques – all that is required is that the mineral be a closed system for the relevant elemental isotopes, and that is usually the case. But knowing when the minerals formed is generally irrelevant to the cave’s history; what is needed is the date at which they were deposited in the cave.

In rare cases where there is “good” included carbon in the deposits, radio-carbon techniques may provide an insight into depositional history (although only out to an age of 60,000 years or so). However, the only technique generally available for determining a depositional age is luminescence dating, with all of its associated complexities and its inherent inaccuracies. (Luminescence ages are the only ones routinely quoted to one standard deviation rather than two – i.e. a “2 in 3” confidence rather than a 95% confidence – and this must be borne continually in mind when reading papers, specially where both luminescence and radiometric ages are used in the same study. If a luminescence age “doesn’t fit”, there is at least a 1 in 3 chance that it is simply wrong.)

Unfortunately (a common word in this field of study), cave sediments are notoriously difficult to study using luminescence techniques. It appears that the best thing for quartz (in terms of rendering it susceptible to luminescence dating) is a hard life. A few million years rattling around the landscape seems to result in an abundance of both signal storage and recombination sites within the crystal lattice, the essential components of a good (and long-running) “geological clock”. Australian desert quartz, which will usually have had such a long landscape history, is generally regarded as easy to work with and date, and this fact may even influence the selection of luminescence-reliant projects tackled by researchers.

Cave sediments often lack this kind of history, and the “pleasant” properties that result from it. They may therefore be unable to store a long-term “clock” signal (lack of storage sites); or it may be difficult to “read” the signal using normal quartz OSL dating protocols (lack of re-combination sites). Indeed, much of the quartz material contained in cave sediments may actually be autochthonous, and have had neither a “hard life” history, nor an actual “date of deposition” in the cave at all. The former case makes it difficult to “date” the sediments using luminescence; the latter case makes it impossible.

Furthermore, from personal observation, most sediments enter a cave during (necessarily brief, at least in geological terms) storm events. Such events may erode sediments from prior deep storage sites directly to the cave so that, especially if the storm event occurs at night, the grains may not have properly bleached out any signal acquired and stored prior to their in-cave deposition. This is referred to as “partial bleaching”; it makes dating of sediments difficult even using single grain techniques, and near to impossible using more traditional “bulk grain” techniques.

So the effective study of cave morphology all too often depends critically on the luminescence dating of poorly sensitised, partially bleached autochthonous grains randomly mixed with autochthonous grains which contain no useful luminescence signal whatsoever. In summary, it is difficult to the point of being non-trivial.

Effective karst sedimentology therefore requires single-grain luminescence dating of hard-to-date quartz grains, and thus requires access to all the data that can possibly be extracted from the grain.
Existing single-grain luminescence techniques and equipment (principally the RISO laser-scanning reader discussed further in Chapter 2) fail in this critical regard due to the following limitations:

1. They have the ability to look at only one stored signal and one re-combination process — the blue-stimulated UV-emitting OSL of the $\Delta 20^\circ$ C luminescence band, which some “difficult” quartzes may not express at all, or at least not adequately for single-grain dating.

2. They completely lack any true spatial discrimination, preventing the detection of any artificially “old” ages due to internal radiation doses (e.g. due to included alpha-emitting minerals). A spatially discriminating system such as the PCIS can show sufficient of the internal structure of a grain (and its emitted luminescence signal) to allow such internally-dosed grains (which would otherwise lead to an excessively large age estimate) to be identified and excluded from any age estimate. (Temporal discrimination between grains that happen to be at different locations, as with the laser reader system, does not constitute spatial discrimination in any meaningful sense.)

3. The existing techniques do not allow the extraction of individual grain data by thermoluminescence (since the heating of the sample aliquot cannot be raster scanned in the way that the laser optical stimulation can be). The PCIS, being a true imaging system rather than a “fake” or raster-scanning system, allows the same extraction of single grain data from thermoluminescence protocols (or thermoluminescence stages of a Single Aliquot Regeneration dating protocol) as from Optically Stimulated Luminescence protocols (or protocol stages).

This last feature perhaps represents the most damning shortcoming of the existing single-grain technology, and the greatest leap forward (at least in terms of dating applications) provided by the PCIS. The availability of the individual grain TL data is of enormous assistance in dealing with otherwise hard-to-date quartz sediments (Nigel Spooner, pers. comm.). In particular, it appears to provide the best way to discriminate partially bleached grains, internally alpha-dosed grains (both of which tend to lead to excessive age estimates) and non-sensitised autochthonous grains (which may tend to lead to an under-estimate of age) from those grains which can actually provide a good estimate of the age or date of the depositional event of interest. It may also be possible to look at other re-combination sites (via different wavelength emitted signals), using the broad spectral capability of the PCIS. These factors are discussed further in Chapters 6 and 7.

So true karst geomorphology remains problematic but, with the right equipment and techniques, not totally intractable. The PCIS is the right equipment, and renders possible the right techniques.
1.1.3.8 The Importance of Karst geomorphology

This section should not be required. Achieving or contributing to a greater understanding of the world and the Universe in which we live is the only meaningful and honourable goal of human existence, and the conduct of investigative science should require no external justification. But we also live in a world of plebeians and economic fundamentalists, and so some justification is called for, if only in order to secure resources against competing human “needs”, like paid matrimonial leave schemes for the wealthy.

Cavers of course need no external reason for the study of caves, understanding that they are important in their own right, and human knowledge and human nature are improved by a closer acquaintance, and a deeper understanding of them. Other less enlightened folk may feel the need for “good reasons” for studying caves. These are fortunately not difficult to supply, from the fields of palaeontology, archaeology, anthropology and climatology, all of which rely heavily on cave sites as repositories of information relevant to those fields. Accessing and interpreting that information requires that the caves not only be entered (requiring the contribution of cavers), but also understood in time, requiring the input of karst geomorphologists.

These fields of study are also honourable and important, and speleologists are usually keen and willing to assist, though we often end up wishing that the relevant scientists would show more respect for, and do less damage to, the caves which have preserved their data for them. But the fact remains that the caves are often the best or indeed the only repositories of such information. Speleothems, for example, are excellent and easily dated recorders of past climatic conditions, and help provide essential information to convince the intellectually challenged (i.e. “climate sceptics”) that climate actually changes in ways that affect human populations.

In summary, karst geomorphology is not only complex and difficult, but also crucially important to various fields of human endeavour. Better tools and techniques for undertaking karst geomorphology are clearly required. The PCIS is just such a tool, making possible techniques of analysis that were previously either impossible or infeasible.
1.2 Goals, Scope and Structure

1.2.1 Principal Goals of the Project

The principal goal of the work described in this thesis was the development of a flexible and reliable research equipment to address the problems and issues in karst geomorphology that were described in the preceding sections. A strictly secondary goal was to conduct a useful karst geomorphological study of the fossiliferous sediment cones of the Naracoorte Caves – also discussed above.

The principal goal required that the PCIS be developed not just to the stage of a “proof of concept” instrument, as would be the case in most PhDs; but rather that I deliver a sophisticated research tool that could be used with confidence by a variety of researchers addressing a variety of karst geomorphology-related problems. This required the extensive automation, monitoring, logging and auditing facilities that were successfully developed, and which are described in detail in Chapters 4 and 5. Also required was a substantial body of documentation, without which the machine would be effectively unusable and therefore essentially useless. This thesis, as well as describing the process of the PCIS’ development, provides just that required body of technical documentation.

The secondary goal of the work, that of undertaking the Naracoorte sediment cones study, was absolutely secondary. Although useful and interesting in its own right, it could have been achieved with less total effort using existing techniques (although not, for reasons discussed above, the laser-based Riso reader – rather a time-consuming project of single-grain-per-aliquot dating using PM tube-based OSL equipment could have been employed). This would have perhaps answered one specific question, but would not have materially advanced the fields of luminescence science or karst geomorphology in the way the delivery of a new technique – and equipment to implement it – might do. The real reason for undertaking such a study, in the context of the PCIS development, would be to prove the effectiveness and performance of the developed equipment.

In the end – and as described in the following section – lack of institutional resources prevented the completion of the Naracoorte Caves project, and the proof of performance of the PCIS was achieved in different ways, testing the PCIS development phases with the various materials analysis applications which are described in Chapter 6. Importantly, the primary project goal was successfully satisfied, and equipment and techniques now exist which could successfully complete a properly funded and resourced study such as that proposed for the Naracoorte Caves.

1.2.2 Scope of the Project

The initial intention when I joined the Research School of Earth Sciences and commenced this PhD was that I would put in a contained effort contributing to the development of the PCIS (which at that time was primarily the project of Dr Spooner), and then perform a body of work that would form the basis of a thesis focussing on the structure and, hopefully, the developmental history of selected sediment cones at Naracoorte Caves.
A further hope of my own was that, if successful I might then develop a project to use the PCIS for a study of the undeniably more complex history and sedimentology of Jenolan Caves (see Osborne, 1999 and Osborne, 2007 on the extended history and complexity of the Jenolan cave system).

Life is often what happens when you’re busy making plans, and so it turned out in this case. The development of the PCIS to the standard I aspired to was a much larger project than any of those involved had anticipated – as I hope a reading of this thesis will show. The early departure of Dr Spooner, and my decision to remain and attempt to complete the project on my own, with only the resources available to a part time PhD student undertaking an orphaned project, did nothing to shorten the time frame or reduce the workload.

Indeed at times my supervisors feared that I would continue to develop and improve the PCIS forever – that it had in fact become a banana project. It is certainly true that improvements and enhancements could continue to be made (see Chapter 7 for a discussion of some of these ways), but I had in mind a point at which “the loop would be closed” on the complex structure of automation software, and it would become possible to use the PCIS with confidence on rare or valuable samples, employing complex protocols that would exercise the most advanced capabilities of the PCIS. That point was reached with the successful development of the PCIS simulator (see Section 5.7), which allows any sequence, implementing any protocol, to be observed running on a standard PC, with a complete audit trail of the sequence (including all the standard PCIS log files), exactly as it would run on the PCIS. However, I really was, now, out of time.

When pressured by time and budget constraints it is always better to sacrifice scope rather than quality, and I believe that the PCIS, as delivered, is a high quality research tool. Lacking infinite budget or resources, however, some of the original scope needed to be sacrificed. In the end, what had to be sacrificed was the software to transform the images captured during a PCIS run into a data stream suitable for submitting to existing palaeodose and age estimation programs – and, therefore, the ability to actually attach ages to sediment grains. (A detailed specification for the software that I would have liked to develop, had time and resources allowed, does however appear at Appendix K.)

In addition, the loss of the luminescence group from the Research School during the development of the PCIS meant that there was not even any laboratory technical support available for the preparation and processing of samples (and I had neither the skills, the qualifications nor the inclination to undertake HF-based low-light sample preparations). Consequently, the detailed analysis of sediments from the sediment cones at Naracoorte Caves had to be deferred. (The samples were collected, and I would like to thank the staff and management at the Naracoorte Caves Reserve for access to their sites and their support of my work. I sincerely hope that the work can be completed in the not too distant future.) Since it was no longer feasible to undertake a significant cave dating project, reliance was placed on the use of existing samples, or samples not requiring extended preparation procedures. Development and testing of the PCIS would therefore be performed primarily with materials analysis (rather than geological dating) applications.

A number of applications not requiring the dating of geological samples were undertaken, and are described principally in Chapter 6. These demonstrate the capabilities, performance and essential

---

3. This delightful phrase is attributed – alas probably apocryphally – to a young girl who claimed that she “knew how to spell ‘banana’, she just didn’t know when to stop”.

Page 21
sensitivity of the PCIS, and clearly establish its ability to perform dating applications when the data manipulation software is implemented. They also clearly establish the functionality and performance of the PCIS automation systems (electronics and software), the simplicity of operation of the PCIS, the repeatability of PCIS runs, and the comprehensive audit trail provided by the multi-level run logging systems incorporated into the automation software.

The sequence software to support an extremely flexible Single Aliquot Regeneration (SAR) protocol (the usual protocol for optical dating applications, and the only one applicable to single-grain dating), including all of the latest “extra cycles” preferred by some luminescence practitioners (e.g. see Murray and Wintle, 2000), has been developed and used in the materials studies referred to above. This SAR sequence software also includes some optional additional features that make use of the specific capabilities of the PCIS. Both this SAR sequence, and an equally flexible “TL probe” sequence (designed for investigating materials, using a broad spectrum search for detectable thermo-luminescence signals) are described in this thesis (see Section 5.2).

1.2.3 Structure of the Document

This thesis is largely the story of the development of the Photon Counting Imaging System. However, it must necessarily also be a description of the machine that resulted from that process, what its performance capabilities are, and how it can be operated to extract the best of its capabilities. With any machine of the overall complexity of the PCIS, such a set of operations and maintenance documentation must be either quite detailed and lengthy, or inadequate. I hope the documentation I have provided here is not inadequate. I apologise for its necessary length.

To demonstrate the performance of the PCIS, and to establish its applicability to the types of problems at which it was targeted, relatively brief descriptions are given of a number of applications to which it has been put, several of which have already resulted in publications.

Finally, some future development directions for the PCIS, and interesting application areas to which it could be applied, are briefly outlined.

The thesis therefore contains the following main sections:

- Chapter 2 describes the opto-mechanical construction of the Phase 1 or “manual” PCIS;
- Chapter 3 describes the applications undertaken to prove the performance of the Phase 1 PCIS, and the investigation of the best techniques for automating its operation;
- Chapter 4 describes the design and development of the PCIS control electronics;
- Chapter 5 describes the development and use of the PCIS control software suite;
- Chapter 6 reviews the major applications undertaken to date with the PCIS; and
- Chapter 7 describes some future developments and applications of the PCIS.

Volume 2 comprises a number of appendices containing supporting information about components of the PCIS. (Three of the appendices are partly or wholly the work of other parties.) Five of the key referenced papers (McCulloch et al, 2011; Spooner, 2000; Spooner et al, 2011; Oks et al, 2011; and Clark-Balzan and Schwenninger, 2012) are reproduced in a pocket inside the back cover of Volume 2.
1.2.4 Some notes on style

While I have some history in writing technical documents in a commercial setting, I have less experience in writing for an academic audience, and less knowledge of the expectations thereof. So as to prevent confusion, I will make clear here a few points regarding my style.

Firstly, the first person pronoun is used by, and in preference to such circumlocutions as, "the present author". For this I make no apology. A preposition is a perfectly acceptable part of speech to finish a sentence with. I occasionally start a sentence with a conjunction. And I freely exercise my right to boldly split infinitives no man has split before. And where a light-hearted or (dare I presume) humorous touch will not detract from meaning or clarity, I indulge myself.

I may mix my units. I respect the technicalities of the SI system, and adhere closely to them where appropriate. However, we live in a time when both metric and imperial units are in general use, and should take advantage of the fact, and not try to force an imperial foot into a metric 0.3048 m shoe. Where something was clearly designed and made to imperial measures, I use those in describing it.

Above all, my goal in writing is clarity of meaning. Ease of style generally, but not always, contributes to clarity. In particular there is often a need in technical documents to repeat a precise phrase or wording, where a variant would give a more pleasing flow but might introduce uncertainty as to whether the same, or a slightly different meaning, was intended. In such cases I will favour precision over style, and apologise in advance for any ugliness thereby created.

1.2.5 Intellectual Property

The author, Iain McCulloch, asserts that, except for those appendices which are clearly identified as being wholly or partly the work of another party, the work described in this document is produced entirely by himself. I hereby assert my ownership of the intellectual property embodied, including but not limited to all engineering drawings, software, language specifications and software and hardware specifications. I retain the copyright and all intellectual property rights in this work.

It is my intention to make all such intellectual property available on appropriate terms to any organisation, team or project developing equipment or techniques that build on the work described herein. In particular, all associated property (intellectual and otherwise) that is in the author’s possession will be made available to the Institute for Photonics and Advanced Sensing at the University of Adelaide to support the ongoing use and development of the PCIS. I have granted a licence in perpetuity to that group to use all runtime software, sequence files and include files developed up to the conclusion of my PhD, including the PCIS Simulator software.

4 The custom of avoiding particular parts of speech in certain contexts is puzzling, and led the founders of the Flowtrack company to adopt as their corporate mission statement, "We will use verbs." Whether this actually says anything about their company is a moot point; but it at least says something about mission statements.

5 The attribution to Winston Churchill of the quote "This is the sort of bloody nonsense up with which I will not put", although it appears in the Oxford Companion to the English Language, would appear to be apocryphal.

6 With apologies both to Fowler and to the writers of Star Trek.
Development and Application of a Photon Counting Imaging System

2 PCIS Design and Development - Opto-Mechanical Components

“I have trodden the winepress alone,
And from the peoples no one was with Me.
For I have trodden them in My anger,
And trampled them in My fury;
Their blood is sprinkled upon My garments,
And I have stained all My robes.”

Isaiah 63:3, New King James Version

2.1 Overview

The goal of the PCIS design process was to preserve the maximum flexibility of application, taking full advantage of the capabilities of the Minisys TL/OSL-DA-15 reader, and of the sensitivity and spectral range of the liquid Nitrogen-cooled charge-coupled device (LN-CCD) detector. To this end a study had been commissioned (see Appendix A) to examine optical designs that would achieve maximum possible photon capture consistent with specified image “sharpness”, and operation over an extreme spectral range.

In order to obtain full value from the spectral capabilities of the system, and to allow the use of the PCIS for investigative work identifying and characterising the various wavelength signals emitted by materials under analysis, I decided that it would be necessary to allow the fast and simple interchange of optical filters during a run, without the need to dismantle the detector system (as needs to be done with the original Minisys). A “filter drawer” would need to be designed as part of the Source Turret to facilitate these operations, and would need to have enormous rejection both of room light, and of stimulation photons leaking past the filters. This would need to be 15 orders of magnitude or more of rejection, and would present a significant design challenge.

To support the range of investigative tasks envisaged for the PCIS, it was also desirable to incorporate the widest possible range of optical stimulation sources. It was proposed to include infrared, red, blue and UV LEDs, as well as supporting the use of either the original Minisys liquid light guide stimulation source, or any other appropriate and controllable external source (which might be of coherent or non-coherent light of any chosen spectrum), via a “general purpose stimulation port”. These goals would further complicate the Source Turret design.

Finally, as it was my objective to deliver a useful research tool rather than just a proof of concept implementation that would support the work for a PhD project but nothing more, it would be necessary to provide a comprehensive and flexible, but simple to use, automation and data logging system for the PCIS. As a general rule, the fewer functions a system supports, the simpler it can be made to operate; and flexibility of application implies a more complex user interface to support the greater range of choices offered – thus a typewriter is easier to learn and use than a word processor (and doesn’t require re-learning every time a new model is released!) The PCIS would clearly require
a multi-level control system, allowing simplicity of operation and a short learning period for users undertaking relatively “standardised” operations, with deeper (and more complex) levels of control available to more demanding users, wanting to use the full range of PCIS capabilities to undertake novel studies using new luminescence protocols. The design and implementation of automation facilities to meet these requirements would become a task of similar or even greater magnitude than the development and implementation of the broadband optics or the source turret assembly.

I was also determined to address two clear shortcomings of the Minisys that I had seen seriously affecting a number of researchers using the standard Minisys facilities – reliability and auditability.

The Minisys acquired a reputation for either not starting correctly at all, or for failing for unidentified reasons during a run. Since a run may involve irreplaceable samples that can only be measured once, any lack of reliability was totally unacceptable and would not be tolerated in the PCIS. I was eventually able to diagnose the cause of many of the Minisys reliability problems as being associated with a failure to correctly initialise the serial communications for the low level command interface, and to avoid them by implementing a more robust start-up procedure in the PCIS software.

A second significant source of unreliability during a run with the standard Minisys, is the failure of the high wattage quartz halogen (i.e. incandescent) globe that is the ultimate source of the blue stimulation light used for quartz OSL. With the PCIS I would try to avoid this problem, as well as the long warm-up times associated with the quartz halogen globe, by using blue LEDs as the source of OSL stimulation photons. All of the LEDs (infrared, red and UV LEDs would also be installed) would need to be mounted as close as possible to the sample in order to achieve adequate intensity of optical stimulation. Provision would be made to allow resort to the original external light box and liquid light guide arrangement should the LEDs prove to be inadequate.

As to auditability, I was asked by a number of researchers to try and confirm the precise treatment to which a sample had been subjected by the Minisys, usually some significant time later when the final analytical results came in and were not what had been expected. In no case was I able to definitively do so, as the Minisys software does not log the operations actually performed along with the output data, and there is no guarantee that the sequence specification file – even if its name has been recorded and is known, and a sequence file of that name still exists – is unchanged from when the run was performed. The methods eventually employed by the PCIS to provide full auditability and repeatability of any run are described in the appropriate sections below.
2.2 Inherited Components

When I joined the PCIS development, the Minisys luminescence reader and the Princeton Instruments LN-CCD camera system had already been acquired using funds from the ANU Major Equipment (Boardman) Committee, and had been subjected to limited performance testing measuring TL signals from various samples via refractive glass optics as described in the introduction.

A design had also been commissioned from Damien Jones of Prime Optics for a broadband lens system that would allow the PCIS to extend its operations over the full spectral range of sensitivity of the CCD – around 200 to 1050 nm (see Appendix A).

These three pre-existing items would form the nucleus around which the PCIS would be developed, and a thorough understanding of them was necessary before the PCIS architecture could be developed. They are discussed, and their important features identified, in the following sections.

2.2.1 Minisys TL/OSL-DA 15 Luminescence Reader

The standard Minisys TL/OSL-DA 15 luminescence reader is supplied by the Risø Corporation of Denmark, and is documented in the paper and electronic data supplied with the machine. That documentation is not reproduced here.

The Minisys reader had been acquired prior to my arrival, and was performing routine dating work on multi-grain quartz samples using its supplied photomultiplier tube for signal acquisition. With the opportunity to dismantle and “play” afforded by the early need to identify and repair a vacuum leak, I made myself thoroughly familiar with its mechanical, electronic and software components.

The complete Minisys reader comprises:

1. The base unit, with attached Sr-90 beta source, turret (containing filters and IR diodes) and photomultiplier tube
2. An electronics assembly, providing vacuum pump control and vacuum measurement, as well as high voltage supply for the PM tube, and also some of the control electronics (the rest of which are in the base unit)
3. A computerised control unit, which supports the Minisys low level control language on a serial (RS-232) interface, and which links to both the electronics assembly and the base unit. This also contains the PM tube counting circuitry.
4. A PC running Risø's supplied sequence control software (Seq117.exe), an Excel spreadsheet-based application which supports the SAR protocol (only), and generates Minisys low-level commands over the RS-232 interface to the Minisys control unit.

In its role in the PCIS, the Minisys base unit would provide the evacuated sample chamber, sample irradiation (using the attached Sr-90 beta source), sample transport, and digitally controlled heating of the sample. In the end, the Minisys also provides some of the timing functions of the PCIS, as this avoids the introduction of indeterminate timing errors due to the (rather slow – approx. 100 mS per
message) serial communications, and provides better synchronisation of the optical stimulation sources with the controlled-rate heating of the sample.

The base unit, the electronics assembly and the control unit form a tangled web of interconnected pieces and functions (as indicated by the proliferation of cables – electrical and optical – which connect them together, with at least one cable connecting to all 3 components). Separating these as part of the PCIS design would be a nightmare, and would involve the unnecessary replication of significant amounts of existing functionality.

The PCIS design would therefore treat these components as a single functional entity, “breaking in” to assert control at the point of the serial interface supporting low-level Minisys commands. The original control PC with its Excel-based sequence application would be discarded, and replaced with custom PCIS software which would control both the Minisys, and the rest of the PCIS.

The related “Center.exe” application provided by Risø, which allows the maintenance of various internal registers of the Minisys, as well as supporting the execution of single low-level Minisys commands entered at the keyboard, would be retained.

The photomultiplier tube was removed from the Minisys (and placed in a dark cupboard), as was the filter and IR diode-containing turret assembly. (Due to the complex cabling of the Minisys, including a single cable with both electrical and optical parts connecting the turret to both the electronics assembly and the Minisys control unit, the turret cannot actually be completely removed from the machine, and continues to sit wrapped in cloth behind the Minisys.) Tests were then conducted to confirm that there were no interlocks preventing the Minisys from operating in this configuration (with the PM tube removed). When operating as the PCIS, no commands would be issued requesting data gathered from the PM tube, and in this circumstance all other features of the Minisys were able to be operated normally.

I also determined that we would need to install the optional I/O card that can be purchased from Risø, which is supported by commands within the low-level Minisys command language, in order to support synchronisation of Minisys operations with other components of the PCIS. Accordingly we purchased what turned out to be simply an over-priced 3rd party external card with 8 relay outputs, and 16 optically isolated inputs, either TTL or dry contact (jumper selected on the card).

The “optional I/O card” does not actually install into the Minisys controller, although the Minisys controller does need to be very significantly disassembled in order to connect the flying lead (supplied) to the I/O card, which must then be mounted in a separate external case (not supplied). This I did, fitting the external case with a D25 connector to support the 8 (3-pin) relay output connections, and a D37 connector to support the 16 (2-wire) isolated input connections. I confirmed (using the Center.exe application) that these could be set and read respectively via the appropriate low-level interface commands, and the other shortcomings were forgiven. I could now control and monitor external components of the PCIS via the Minisys, using the known and already supported low level command language.

Since the documentation of the Minisys low level language is at best terse, I felt that how and why the various commands were intended to be used could best be determined with the aid of seeing how Risø themselves used those commands from their own Sequence program. Accordingly I set up
a spare PC of my own, with 2 serial ports, to monitor all traffic on the low-level command interface during “normal” Minisys sediment dating runs. To achieve this I made a special cable, which split out signals from both the Transmit Data and the Receive Data lines between the Minisys and the control PC, directing each to the Receive Data line of separate D-shell connectors, and then connected them to the 2 serial ports on the monitoring PC. I ran 2 copies of HyperTerm (the serial terminal emulation program then supplied with Windows), causing each to log the received data to a file called respectively “ReceiveDataRunnnn” and “TransmitDataRunnnn”. (A commercially produced data monitor - such as the excellent HP units I used in a previous life - would have been both easier and better, but was not available to me.)

I left this arrangement running for a number of runs, and eventually made it permanent as it was useful in determining the cause of periodic failures of the Minisys when running its supplied software. Thus I learned not only how to use the various low level commands, but also some important cases of how not to use them! Although these files, being 2 separate logs without a 1 to 1 correspondence between their lines, were far from convenient or simple to use (and in the end I devised much better logging facilities for the PCIS run software), their analysis was sufficient to confirm that it would be possible to develop PCIS software which used those commands to control the Minisys and, via the installed I/O card, other synchronised components of the PCIS.

Unfortunately an interlock was discovered to exist within the Minisys equipment, and it does not allow the vacuum pump to run concurrently with the Nitrogen feed to the sample chamber. (If that sequence of commands is given, the Minisys reports an error and stops). Given the existence of a small leak into the sample chamber (believed to be around the central spindle, instructions for the removal of which Risø had refused to supply), far better oxygen molecule removal could be achieved if Nitrogen were used to flush the chamber for a period before turning off the vacuum pump. Risø’s stated reasons for not allowing this did not appear reasonable to me. They were:

1. That starting the N₂ flow while the pump was running would cause sample disks to be blown off the carrier wheel. My view (as supported by my experience) was that if the initial blow of N₂ into a chamber that was at less than 1 mBar pressure didn’t cause the disks to be blown off, then the existence of a vacuum pump a metre or more away maintaining a lowered pressure in the chamber wouldn’t cause it to happen either.

2. That pumping pure N₂ wasn’t good for the vacuum pump. My view (as supported both by in-house technical advice taken at the time, and later by experience) was that if the pump could pump air which was 80% N₂, then removing the more corrosive 20% O₂ wasn’t going to make any difference – it would pump pure N₂ just fine.

Accordingly, it was decided that the vacuum pump (which runs from an electronically switched but otherwise standard 240 v AC supply) would need to be disconnected from the Minisys electronics assembly, and controlled separately as one of the PCIS ancillary components (i.e. indirectly via the I/O card installed in the Minisys).

Finally, it was noted that the design and construction of the Minisys absolutely required that the sample be held horizontally, face up, somewhere in the region of 200 mm above the bench on which the Minisys stood. No other configuration is possible, and both the optical design chosen, and the mechanical architecture of the PCIS, would need to accommodate this fact.
2.2.2 Princeton Instruments LN-Cooled CCD with ST-138 Controller

The LN-CCD camera system acquired from Princeton Instruments prior to my joining the project comprises the following three major components:

1. The TEK 512x512 DB camera head itself, containing the LN-chilled CCD detector, low-noise analogue/digital converter and their associated electronics (as well as the LN dewar);
2. The ST-138 controller, providing high speed electronic control of the detector and associated shutter, and supporting the set-up of operating modes of the CCD; and
3. The control PC running the WinView camera control and image manipulation software.

The image data is acquired from the detector by the ST-138 controller, in a format determined by the current set-up of the detector. The data is then passed via a high speed serial interface to the control PC, where the WinView application allows some manipulation of the images, and supports their display and analysis. The features and functions of these 3 components need to be examined in quite some detail here, as their proper understanding underlies the design of the overall PCIS architecture that is described in the following sections.

It was long an open question whether the PCIS would require custom software to control the camera system, or whether the operation of the PCIS could be accomplished using only the facilities supported by the supplied WinView software.

Producing custom software for the camera would require obtaining the Software Development Kit, and learning to program using the supplied library of camera control routines. Given the number of other custom software environments that needed to be learnt or created from scratch for the PCIS, there were obvious attractions in avoiding yet another system of some depth and complexity.

To avoid the development of custom camera software would require full use of the facilities of the ST-138 controller and the camera control functions of the WinView software, so these were investigated in detail. For a number of these control parameters, examined individually below, custom camera software would allow them to be changed “on the fly” at any time during a PCIS run, whereas reliance on the existing WinView software would likely imply that they could be set only before a run commenced, and would then remain as set for the whole of the PCIS run. How critical the loss of “on the fly” control would be in each case would need to be assessed, and measured against the development effort required to produce fully customised camera control software as part of the PCIS automation process.

2.2.2.1 Principal Control Parameters of the LN-CCD

The most important set-up parameters affecting the performance of the LN-CCD are:

- Readout rate;
- Binning factors; and
- Timing and synchronisation parameters.
2.2.2.1.1 Readout Rate Settings

The readout rate setting (actually controlling the frequency at which charges extracted from the CCD wells are presented to the A/D converter) allows the choice between a reasonably fast camera, or an extremely low noise camera.

At the lowest supported readout rate (50 kHz) it takes in the order of 5 seconds to read an image from the CCD, or to clean the CCD surface of charge preparatory to recording an image. (Reading and cleaning the CCD are identical operations, except that in the latter case the image data is not stored or transferred to the control PC.) At higher read-out rates the “cycle time” for the camera can be reduced to as little as half a second, but the read-out noise increases so dramatically (by a factor of 100 or more) that the system can no longer be considered as an ultra-high sensitivity detector.

Given that luminescence work is characterised by the very low level of the optical signal being detected and measured, it is extremely unlikely that increased read-out rates would be called for at all, let alone “on the fly” in the middle of a run. However, 5 seconds of dead time between image collections is not ideal either, especially where a sample is being measured at elevated temperatures, and signal will continue to be lost during the readout time.

For those rare cases where the quantity of signal is so high that the severe degradation of signal to noise ratio can be tolerated, and where the timing constraints are critical for the experiment being conducted, the camera can be set to a faster read-out rate, via the WinView software, prior to the commencement of the run. I could not envisage a need for mixed high speed and low noise components of a run, and decided that sacrificing the ability to vary this parameter on the fly was acceptable. If necessary, a researcher could split the work into two separate PCIS runs, with the camera set to a different readout rate for each, and combine the data for the total experiment.

2.2.2.1.2 Binning Factor Settings

Binning factors are a much more important way of trading sensitivity against other factors – in this case, a way to increase effective sensitivity at the expense of image resolution (i.e. number of pixels).

It should be understood that sensitivity is effectively the same thing as signal-to-noise performance. With an inherent quantum efficiency of about 70% across the spectrum of interest, the camera is clearly capable of detecting signals of extremely low total energy. Whether the collected signal is of any value, however, depends on whether it can be identified “above” the detector noise, or whether the signal is swamped by that noise. Thus raising the background noise and lowering the detector’s sensitivity amount to the same thing; and the concepts of signal-to-noise ratio and sensitivity may be used somewhat interchangeably.
The detector itself is a 512 by 512 pixel Charge-Coupled Device (CCD). Each pixel is considered as a “charge well” of a certain size (i.e. charge capacity). Each pixel also has a physical size – in this case about 24 µm square – that determines the total number of photons that will fall on it (and hence the total charge that it will accumulate) in a given exposure. When the detector is read out, the accumulated charges are cycled sequentially to a “transport column” at the end of the rows; and thence sequentially to a read-out cell at the end of the column. At the read-out cell, the charge is converted to a voltage that is compared to a reference voltage by the analogue/digital converter, and thus becomes a number representing the “brightness” or, in this case, the number of photons that were detected striking the given pixel.

The detector, the A/D converter and the most immediate related circuitry are kept on a cold finger (the other end of which is immersed in liquid nitrogen), and held at a temperature of -110C +/- 0.05 C. This minimises the noise introduced in this stage, and is fundamentally what gives this system its enormous sensitivity (i.e. its huge signal-to-noise ratio) when compared to a typical CCD application in a consumer digital camera.

[Note that achieving very low noise levels on the signal read-out also requires that the charges be read out slowly, so that the voltages presented to the A/D converter have time to stabilise, and can be converted accurately to photon counts. That is why the system shows the huge sensitivity losses that were noted in the preceding section, when the read-out rate is increased. It is also why early digital cameras suffered such a pause before taking a photo – cleaning the CCD (which as we noted above is the same as reading out an image) was slow, until further development of CCDs addressed the problem. To reduce the delay, modern digital cameras incorporate special circuitry to support high-speed cleaning of the CCD before a photo is taken – the PCIS CCD is not so equipped.]

It turns out that the overwhelmingly predominant source of noise lies in the conversion of the accumulated charge to a voltage via the read-out cell, and not in the conversion of photons to charge in each pixel, nor in the transport of the charges from the pixel to the read-out cell.
cell. Therefore if multiple pixels are combined, and their aggregated charge is converted to a voltage (and hence to a count) as a single unit, then the signal is larger, but the noise component is not, and the signal-to-noise ratio – and hence the sensitivity – has been increased by a factor equal to the number of pixels aggregated. This is exactly what the “binning” function of the LN-CCD does.

Binning the CCD allows the face of the CCD to be divided into rectangles by defining groups of columns and rows of pixels that will be read out as a group. Although the columns and rows can be set arbitrarily, this discussion will assume that binning is always square – i.e. that pixels are grouped into squares of 4, 9, 16 etc pixels. If binning is used in the PCIS then I expect it would be square binning anyway, and the assumption makes the discussion easier.

Binning provides a way to increase the sensitivity of the CCD camera by a factor of ten or more, at the sole expense of spatial discrimination – i.e. of pixel count. This is extremely powerful, and a facility that should be available to the PCIS.

Against the above it could be argued (albeit wrongly) that the whole point of the PCIS is to provide spatial discrimination (i.e. imaging), so that to sacrifice it for any other benefit is a retrograde step. However, as will become clear when we look at the optical designs, the ¼ megapixel resolution of the CCD detector is effectively better than the imaging that can be expected from the optics, and so there is actually some spatial resolution “to spare” in the CCD itself.

Since the very small optical signal energies generated is perhaps the limiting factor for the application of luminescence techniques to a wider variety of samples and materials (a problem which certainly becomes worse, not better, when attempting to form an image, rather than just measuring the signal), any gain in sensitivity is a very important benefit indeed. Accordingly, I expect that square binning in 2 by 2 or 3 by 3 format, yielding a fourfold or ninefold increase in sensitivity, is likely to be a common practise with the PCIS, both when analysing faint samples, and when pushing short exposure times to improve the temporal discrimination of signals.

Some more on Binning

Binning is not just a facility provided for high end, high sensitivity scientific cameras. Every consumer digital camera “bins” the several megapixel CCD to provide a real time moving image on the – usually about 100 kilopixel – viewfinder screen, even in low light when the real exposure may take ½ a second or more. Other techniques are also used – like lowering the reference voltage to the A/D converter, as is done for high ISO ratings – but the main benefit still comes from extreme (up to 8 by 8 or more) binning factors.

The transport column and read-out cell of all CCDs are designed with significantly larger charge capacities than the individual pixels, so that they can handle the multi-pixel charges being shifted when binning is used. Nevertheless, these charge wells can become saturated, producing the bright streaks of light often seen emanating from a bright point on the viewfinder (but not on the final image) of a digital camera. These streaks always run in one direction only - the CCD shift column direction.
So the need for the PCIS to be able to run with any specified binning factor at the CCD is established. Whether it is necessary that the PCIS be able to change the binning factors of the CCD camera in the middle of a run is less clear. There is a cogent argument to be made that having part of the data set at a higher spatial resolution than other parts would be unlikely to be of benefit, since the final analysis would necessarily be performed at the lower resolution anyway. By this argument, the whole of any given run should always be performed at a single resolution – i.e. without change to the binning factor mid-run.

So it would seem advantageous, but not necessary that the PCIS take control of the CCD binning parameters. And so the matter will be left, for further consideration during the PCIS system architecture design.

2.2.2.1.3 Timing and Synchronisation

The ST-138 controller, supplied with the LN-CCD camera, supports the high speed acquisition of image data from the camera, and its delivery to a custom I/O card in an otherwise standard PC.

While the front panel of the ST-138 is quite minimalist, the rear panel is richly provided with cryptically labelled BNC connectors supporting a wide range of signal exchange and system control functions. Three of these would prove crucial to the final PCIS control architecture, and so will be described here.

The three signals of interest are labelled Sync, Trig and NotScan, but I’ll explain them anyway.

When a particular Run Mode is enabled via the WinView software menus, an accumulation (CCD-speak for an image taken, or a picture) that has been initiated via the software will not actually be undertaken until a falling edge signal is detected on the Sync input. (All TTL signals are active low, so that the normal indication of a signal going active is a “falling edge”.) Rise however chose to adopt a different logic between the low level command language and the external I/O connections, so that extreme care needs to be taken with the polarity of signals in this part of the design. Where confusion might otherwise arise, the terms “active” and “inactive” will be preferred.

In this same operating mode, an accumulation gathered when Sync is activated will, at its conclusion, be stored to the PC only if an active Trig pulse has been received (since the preceding accumulation was dealt with). If no Trig pulse has been received then the image is not stored – the CCD has just performed a surface charge clean without taking a picture!

Normally, the CCD surface is cleaned at the start of (i.e. prior to) an accumulation. However, an available mode called “Shutter PreOpen” will cause the CCD to open the shutter and wait – and when the accumulation is initiated (or the Sync line cycles) the requested accumulation time is started immediately, and then the image is stored (or discarded if Trig has not been cycled).

By using these modes (and setting the CCD accumulation time to zero), the PCIS can use a CCD Clean cycle (or just open the shutter) to start an accumulation; and cycle the Sync line to end the
accumulation. In this way the PCIS can take control of both the timing and duration of the accumulations, while running just the standard supplied WinView software.

The third line of interest is the NotScan output from the ST-138. This line goes active when the ST-138 is not reading data from the actual CCD detector. Thus the PCIS can use this signal to determine when the CCD camera has finished processing one accumulation (or CCD Clean cycle), and is ready to undertake another.

The relevant signal timing diagrams appear in the Appendices to the ST-138 Controller manual, and are not reproduced here. It was an examination of these signal timing diagrams that led me to the solution outlined above for synchronising the LN-CCD camera to the rest of the PCIS. Testing was conducted to confirm that the signals did in fact operate as195(195,195),(582,263) I had understood, and being successful, this eventually became the adopted camera synchronisation design, as described in the later section titled “PCIS System Architecture”.

There are numerous other parameters available to be set via the WinView Experiment Setup and Hardware Setup menus, but most of these fall into one of two categories:

1. Those for which only one value is sensible, or which refer to functions (such as dynamic frame shift modes) not used by the PCIS; and
2. Those which are equally or only applicable post data collection, such as spatial cosmic ray removal and background subtraction.

Since none of these would require active control during a PCIS run, it did not seem that the PCIS control systems and architecture would need to take account of them. Decisions as to their use could safely be left for a later date (indeed, a final decision on the benefits of cosmic ray removal algorithms has still not been reached). They will not be considered further here.

2.2.3 Broadband Reflective Optics design, Prime Optics

The third significant pre-existing component was in the form of a design document only – the document produced by Damien Jones of Prime Optics which discussed and analysed 5 classes of optics, and 4 specific representative designs, to meet the needs of the PCIS. This document is titled “Broadband Cameras for Luminescence Dating” and appears at Appendix A.

The selection of a design for the PCIS optics was probably the single most crucial technical decision to be made. The entire success of the project depended on the ability of the optics to capture as many as possible of the available photons, and create from them a good quality unity magnification image of the sample, across the whole spectral range of interest, onto the detector face of the camera.

The analysis of the designs presented, and the eventual selection of the recommended Offner design, was carried out jointly by Dr Spooner and myself, and is presented in the following section.
2.3 Selection of the Optical Design

The optical design report produced for the PCIS by Damien Jones of Prime Optics (Appendix A) considered a number of designs for the PCIS Optics before recommending the Offner full reflective design. Dr Spooner and I discussed the various options, and came to the same conclusion in favour of the Offner design — and for essentially the same reasons — as had Damien. The considerations leading to the adoption of the Offner reflective design — and the difficulties which that decision would inevitably create — are discussed below.

A perfect lens would produce an absolutely accurate, sharp and colour-true image of a scene or object, in a perfect flat plane that would then become the film-plane or detector-plane. No lens is perfect, and the imperfections are referred to as aberrations. The aberrations are classified into a number (often but not always 7) of types, and different techniques are used in lens design to reduce the impact of the different types of aberrations. These techniques are usually in competition with each other, and any real lens design represents a compromise between the different aberration types. The optimum compromise depends on the application to which the optics will be put.

However the aberrations are classified, they are always considered in two main classes; chromatic aberrations, and monochromatic (or sometimes just “other”) aberrations. Chromatic aberrations are those which cause different wavelengths of light from the same point source to arrive at different parts of the image plane, and are due to the differing refraction angles of different wavelengths of light at each refractive surface of a lens. Monochromatic aberrations are (largely) independent of the wavelengths of light involved, and so cause a distortion of the image, or a variation in sharpness across the image area, even in monochromatic light.

The usual approach to reducing chromatic aberrations is the introduction of multiple lens elements of different dispersions, which will tend to cancel out each other’s chromatic aberrations, but without cancelling out the fundamental optical properties of the lens. These attempts to reduce chromatic aberration often tend to introduce or worsen the spherical aberrations, and further refinement of the design (often implying further lens elements) are required to reduce those.

Monochromatic aberrations arise because most lens systems contain at most one stigmatic point, but are required to form extended images. For example, a paraboloidal mirror forms a stigmatic (geometrically perfect) point image of an on-axis point source at infinity. Off-axis object points do not image stigmatically, producing coma (spreading) in off-axis images. Good optical design aims to approximate stigmatism over extended images, whilst minimising element numbers and complexity.

The larger the aperture of a lens — that is, the more of the available light you wish it to capture — the worse all forms of aberration become, and the more corrections need to be applied.

The wider the spectrum over which a lens must work, obviously the worse the effects of chromatic aberrations become and the more corrections need to be applied in the optics design to reduce the chromatic aberrations to acceptable levels.

(This would appear to be why all animals which have developed high acuity vision using refractive optics — such as ourselves — have given up vision in both the infrared and in the ultraviolet, though certainly in the case of human beings the receptors in the eye are still sensitive in the near UV. This
curiously resulted in some early recipients of replacement lenses in the eye acquiring vision – albeit somewhat difficult to focus – in UV light invisible to the rest of us mortals. Insects on the other hand evolved what are fundamentally reflective optics in their compound eyes, and although not achieving the same acuity as human vision, have retained vision into the UV. Many plant and animal camouflage systems only make sense when photographed including the UV part of the spectrum.)

In most lens systems, the end result of applying techniques to successively reduce or remove the different types of aberration is a proliferation of elements; the resort to non-spherical elements; the use of bizarre materials with extreme optical properties; or some combination of all three. Non-spherical elements are difficult to design, and even harder to produce accurately (unless the other design constraints allow the use of molded “optical plastics”, which in any case can only be done cost-effectively with mass-produced consumer items such as modern digital cameras). “Bizarre” materials also tend to be expensive and difficult to produce as lens elements, and may have poorer total transmissive properties than glass. A proliferating number of elements – whatever their material – not only involves expense and complexity, but more importantly implies rapidly increasing losses of the transmitted light – i.e. a reduction in the effective aperture of the lens – due to reflective losses at each refractive surface.

The PCIS optics – with the concurrent requirements for extremely large effective aperture and unusually wide spectral range – would clearly constitute a significant challenge in both design and production.

2.3.1 The 4 Optical Designs Assessed

The four different optical designs considered and assessed by Damien Jones of Prime Optics are discussed below. These designs include a fully refractive design, a fully reflective design (though with transmissive filter and window elements), and two designs combining both reflective and refractive elements. Before considering the four designs in detail, it is worth considering some general aspects of reflective versus refractive designs.

Reflective optics – that is, optics comprising an assembly of mirrors rather than lenses – are achromatic (since the law of geometric reflection is independent of wavelength), but offer fewer degrees of freedom (represented in a refractive design by the introduction of new elements, each with 2 curvatures and a refractive index to be chosen) by which monochromatic (mainly spherical) aberrations may be managed. Refractive optics suffer from both chromatic and monochromatic aberrations, but offer more design options for addressing these issues. However the proliferation of surfaces, and the accumulation of losses from partial reflection at those surfaces, places a limit on the process of adding corrections to eliminate aberrations.

It should also be noted that reflective optics generally require far finer tolerances on both the shape and the placement of the elements, with respect to each other, the object and the image plane, than do transmissive (i.e. refractive) optics. This is due to the fact that a small misalignment of a transmissive element will lead to a smaller change in the angle by which light is refracted, whereas a small angular error on a mirror will lead to double that error in the reflected angle. Reflective optics
must be polished to far closer tolerances, and are significantly more difficult to align correctly than are refractive optics – which are often mounted in a simple tube with spacers (and a couple of helical threads for focussing) so that no separate alignment of individual elements is required.

These factors will show up in the assessment of the lens designs discussed below.

It is also necessary before examining the lens design options to understand the term “RMS radius spot size”. This is the root mean square of the distances on the image plane, from a calculated ‘central’ position, to the actual positions at which photons from a single point source arrive, as some parameter of the source light is varied. The varying parameter may be the wavelength of the light (for assessing purely chromatic aberrations); the angle at which the emitted light rays strike the first surface of the lens (when assessing purely monochromatic aberrations); or both, providing a measure of the “sharpness” of the lens under both wide aperture and wide spectral range conditions. The preferred target performance of the PCIS optics on this parameter was 50 µm.

Finally, with a targeted wavelength range for the optic design of 200 to 1050 nm, it is worth quoting from Damien Jones's report some comments on the choice of glass for any transmissive design:

“The specified short wavelength coverage of the system makes the use of special UV glasses mandatory. There are 3 to choose from: fused silica, Calcium Fluoride and Lithium Fluoride. Achromatization is best achieved with silica and Calcium Fluoride. Lithium Fluoride is more problematic because of its solubility in water and dispersion mismatch.”

### 2.3.1 Fully refractive design

To explore this approach Damien Jones developed the design below, which includes 8 each of fused silica and Calcium Fluoride elements.

![Figure 2.3.1 Fully refractive PCIS Optics Design (D. Jones).](image-url)
The large number of silica and CaF$_2$ elements would imply a high construction cost, which is not rewarded with especially good “sharpness”, having an RMS radius spot size of 90 µm. The large number of internal surfaces also means that the effective transmission of light through the system would be significantly less than that calculated from its solid capture angle. To quote again from Damien Jones, “Such a mediocre system would not be a cost-effective solution because of the materials and fabrication expenses.”

Due to the reasonable availability of only two UV-transmitting materials (i.e. only two refractive indexes) with which to control chromatic aberrations, no refractive design is going to perform significantly better than the one above with any fewer elements. When it is also considered that this design would require the LN-CCD camera to be suspended upside down some distance above the samples in the Minisys, it can be concluded that our hunt for the ideal lens design is not yet over.

### 2.3.1.2 Combined Reflective/Refractive design with Internal Foci

This class of design (according to Damien Jones) “encompass[es] Schmidt, Maksutov, Houghton and other systems.” The design developed for analysis is a Houghton system, and is shown in the figure below.

![Figure 2.3.2 Houghton Refractive/Reflective Design with Internal Foci (D. Jones).](image)

Although the predicted optical performance of this system is excellent (16 µm RMS spot radius at f/1), and the number and shapes of elements are reasonable in terms of fabrication costs, the
insurmountable difficulty is that the sample and detector are not isolated objects, but are embedded in machines of significant size. As those machines (the Minisys and the LN-CCD camera) are not transparent, the design fails.

To quote again from Damien Jones, these systems “all suffer from obscuration effects and accessibility of the focal plane.”

2.3.1.3 Combined Reflective/Refractive design with External Foci

Without repeating all of Damien Jones’s discussion of the available design choices in this class, we will go straight to his chosen representative design for this class which appears below.

![Diagram of Houghton Refractive/Reflective Design with External Foci](image)

Figure 2.3.3 Houghton Refractive/Reflective Design with External Foci (D. Jones).

This design delivers excellent light gathering and very good imagery (~50 µm RMS radius spot), but as can be seen from the diagram it requires transmissive elements that are sized according to the distance between the sample and the detector (~200 mm), rather than being sized according to the size of the sample and detector (~10 to 12 mm).

To again quote Damien, “material costs alone are prohibitive.” However, the performance would be very good, the physical placement of the sample and detector are ideal, and it was felt that this option should be kept under consideration for a possible future, better funded project.
Damien in his report briefly considers reflective systems with internal focii (which “encompass some Cassegrain and all Gregorian configurations”), but concludes that “None deliver fast focal ratios easily and all have complex surface shapes.” As was noted above, complex – i.e. non-spherical – surface shapes are extraordinarily difficult (and correspondingly expensive) to produce, especially as one-offs and to the tolerances required in a reflective optic system.

2.3.1.4 Reflective (Offner) Design

In the class “reflective systems with external foci”, Damien considered both the Offner as an example of a concentric (or perturbed concentric) design, and the triple-mirror anastigmat (TMA) as an example of off-axis systems. He concluded that “TMAs are a fabrication nightmare, having complex surface figures and large decentrations” – advice that I am strongly inclined to take at face value. He goes on to develop for analysis the Offner system (shown in the figure below) as it is “based on purely spherical surfaces” and has “quite reasonable tolerance demands” (for a reflective system).

![Image of the selected Dioptric (Offner) System with External Foci](image)

Figure 2.3.4 The selected Dioptric (Offner) System with External Foci (D. Jones).

This design delivers a capture angle equivalent to f/1.4 or better, with imagery somewhat poorer than desired (at 75 µm RMS radius spot size off axis; better than that on axis), but still acceptable (and more than adequate for the purpose of allocating a photon to a given grain on a 1 mm grid).
There are only three reflective elements which—a though of a large size and high curvature that would create some difficulties in polishing and coating—are all of spherical section. The fabrication costs should therefore be the most reasonable of the options examined—possibly even within budget! In addition, the placement of the sample and detector are ideally suited to the constraints imposed by the existing (Minisys and LN-CCD) components.

We therefore decided to proceed with the Offner design, ordered the appropriate Pyrex glass blanks for the fabrication of the elements, and organised with Gabe Bloxham from Mt Stromlo Observatory to undertake the grinding and polishing of the three mirrors. I was meantime free to start the design of the numerous bespoke components that the full PCIS would require.

Another view of the Offner system appears below, showing better in perspective how the light rays traverse the 3 mirrors to provide the 1:1 imaging at the detector. The diagrams include the necessary transmissive components—the optical filter pack and the vacuum windows—which are properly accounted for in both the lens design, and the expected image quality calculations.

Figure 2.3.5 The Selected Offner Reflective Design, Perspective View (D. Jones).

Now, with a clear understanding of what the Minisys and LN-CCD systems offer to the PCIS—both in terms of facilities and functions, but also in terms of limitations and constraints—and with the optics design chosen, it was time to set the design goals for the PCIS, and then to start to develop an architecture that would support those objectives, and within which the individual bespoke components of the PCIS could be designed and developed.

The following section looks at those design goals, and the PCIS architecture that was developed to support them.
2.4 Broad Architecture of the PCIS

2.4.1 Overall design objectives

When I undertook the development of the PCIS, and when I also undertook a PhD, my commitment was to produce a fully developed item of luminescence research equipment that would hopefully have a useful research role well beyond the completion of my studies. It must be capable of as wide a range of tasks as possible (not just the work I wanted to do), and needed to be able to be used simply and effectively by researchers other than myself.

The PCIS would therefore need to be:

- Flexible enough to take advantage of all its inherent capabilities, and able to perform a wide variety of analytic tasks combining both thermal and optical stimulation techniques
- Able to discriminate, measure and image luminescence signals across the whole of the spectral range of the LN-CCD detector
- Simple to operate for established luminescence protocols, yet able to be programmed for any sequence of supported operations for more unusual analytic tasks
- Reliable – sufficiently so for rare or irreplaceable samples to be committed to its clutches without undue fear of machine failure and consequent loss of the sample
- Complete with data recording functions that aid with the management of the vast quantities of data that would be produced for each and every sample; and
- Auditable – i.e. it must be possible to trace after the fact the exact sequence of operations performed on the sample, and to replicate it precisely for another sample.

It should be noted that the pre-existing Minisys equipment performs poorly on a number of these criteria – particularly on flexibility, reliability, auditability, and repeatability – and that the minimal goal in each case was to exceed the corresponding performance of the Minisys.

2.4.2 Initial PCIS Architecture

The PCIS architecture emerged as the logical, if not the only viable way to meet the design goals, given the constraints represented by the existing components, with their individual capabilities and limitations. The limited resources available to the project also limited the design choices available, though not in the end, to any significant degree, the capabilities or usefulness of the PCIS.

The PCIS architecture is considered below in two main components: the mechanical (or more precisely, mechanical and optical) architecture; and the system, or PCIS control architecture.
2.4.2.1 Mechanical Architecture

The chosen design of the optics and the dimensions of the existing components (the Minisys and the LN-CCD) almost dictated that the physical layout of the PCIS would consist of the Minisys to the left and the LN-CCD Camera to the right on a rigid bench, with the 3-mirror lens system housed in some form of rigid and lightproof housing sitting above, and bridging the gap between, the two items.

As noted above, the camera has no designed flat surfaces to rest on, and so would need to be either:

1. Suspended from the optics assembly via the large threaded ring around the face containing the CCD detector; or
2. Provided with a special cradle that would hold it accurately and rigidly in alignment so that the optical assembly could mate to it.

Option 1 above appears to be the manufacturer’s intention for the camera (no other good mounting surface being provided), and would automatically ensure accurate mating of the camera to the optics. However, if the optics support the camera, what supports the optics? The Minisys top plate (which covers, and moves to provide access to, the wheel of samples) is not capable of supporting that mechanical load cantilevered off it to one side; and the motor that lifts the cover plate (so that samples can be changed) certainly could not lift it. So option 1 would require the combined camera/optics assembly to be supported in some fashion anyway, and then to be moved as a combined unit to provide access to the sample chamber.

Option 2 involves making a carrier for the camera to hold its front face extremely rigid and level (and at the exact right height with respect to the top of the Minisys), then bringing the optics assembly to the correctly positioned Minisys and LN-CCD camera, and attaching it simultaneously to both components. The optics assembly would still need to be moved for access to the sample chamber, but only the optics assembly; the LN-CCD camera would then remain in place beside the Minisys.

The optics assembly would inevitably be quite heavy. The glass alone would be over 5 kg, and the very rigid full enclosure, plus focussing and adjusting mechanisms, would yield a total weight of at least 30 kg. It was felt that with good alignment of the Minisys and LN-CCD, the lens could achieve adequate alignment and light sealing simply by resting on the Minisys and LN-CCD under its own weight. Mating rings incorporating O-ring light seals would provide positional accuracy and light exclusion, but the optics would be easily lifted and moved aside to provide access to the samples. And no complex locking mating assemblies would need to be operated for each sample change.

It was decided to adopt option 2, and this has proved completely successful, although it did in the end require not only the manufacture of a special camera support stand, but also:

- The installation of a completely flat 30 mm thick alloy bench top for the Minisys and the LN-CCD to sit on;
- The manufacture of a custom, adjustable base for the Minisys to allow its height to be accurately matched to that of the LN-CCD; and
- The development of an interesting capacitance-based procedure (that will be described later) to allow the height adjustment to be made accurately.
These considerations, and the constraints imposed by the features and limitations of the pre-existing equipment (the Minisys and LN-CCD) as described above, inexorably guided the physical design of the PCIS towards the final implementation of:

- Near-cubical optics assembly including the focussing and focus indicator assemblies
- Source turret to fit the space between the top of the Minisys and the base of the optics assembly, and containing: a UV-transmitting window to preserve the vacuum in the sample chamber; multiple optical stimulation sources; and an optical filter pack to discriminate emitted photons from stimulation photons. The source turret would also provide a positionally accurate light-tight mating assembly to join it optically to the lens assembly.
- Destination turret providing for the electronic shutter, excess light level detectors, and optical mating to the LN-CCD. The destination turret (or shutter housing) also provides the correct physical placement of the LN-CCD with respect to the optics and the sample.
- An assembly to raise and lower the optics to allow for the insertion and removal of samples from the Minisys while preserving the accuracy of placement of the lens with respect to the Minisys, the sample and the LN-CCD;
- Electronically controlled power supplies to drive the stimulation LEDs for OSL work;
- Electronics to support the synchronisation of the LN-CCD with the rest of the PCIS (primarily with Minisys operations); and
- Software to provide automation and sample tracking capabilities for the completed system.

A wide range of other ancillary components would also be designed and constructed, including a range of alignment jigs and component testing tools.

Figure 2.4.1 below shows the major mechanical components of the PCIS in what is close to their final configuration, although neither the source turret nor the shutter housing have yet been anodised. The photo shows Dr Spooner and myself lowering the lens onto the Minisys base and the LN-CCD prior to conducting a run. The blue LEDs seen shining from the source turret were turned on for photographic effect only, and would not normally be lit at this stage of proceedings.

Figure 2.4.1 Dr Spooner and myself lowering the broadband optics into place during assembly of the PCIS.
2.4.2.2 PCIS System Architecture

Although the total PCIS architecture as it finally emerged, and is shown in overview in the figure below, was not known in detail this early in the planning of the project, the most significant aspects of it were already emerging. In fact, the architecture shown below, with the Integration and Interface Control Unit (IICU) providing control of the full range of functions shown there, and being a fully programmable unit with its own management software, did not emerge fully until all of the opto-mechanical components described so far had been produced, and shown to work.

In the initial stages of running the PCIS, the camera, shutter and LED drivers were all manually controlled, using simple “alpha testing” electronics that I built for that purpose. There was little automation of any operations until the final IICU had been designed and commissioned, and the initial versions of the PCIS Sequence program had been written. There was no way to reliably give an exact or repeatable treatment to a sample, and certainly no possibility of unattended operation!

However, by the time that the opto-mechanical components were operating, sufficient of the final design was clear to me that the “alpha electronics” were designed, and operations of the PCIS were developed, in a fashion that allowed the viability of the final design to be confirmed. Only then did I fully commit to the system integration approach represented in the diagram below, and commission the development of the IICU by the RSES Electronics Workshop to a detailed functional specification produced by myself, and appearing at Appendix D.

Although it is getting ahead of our story to present the final PCIS Architecture so much ahead of its time, I think the story of the design and development of the major components of the PCIS will be easier to follow if you have some idea of the architecture into which they were being designed to fit. As I’m not sure how to convey “some idea” in a useful fashion, I will instead attempt to convey a clear picture of the final system architecture, and then move on to a detailed description of the individual components that constitute it.

Once the story of developing the PCIS up to the “alpha testing” or “proof of concept” stage has been told, and the full task of aligning the mirrors and proving the performance of the broadband optics has been explained, the following sections will return in detail to the development of the system integration and automation components – the IICU, its internal command tables, and the various software items that support the full functioning of the automated PCIS.

In Figure 2.4.2 certain components will be recognised as “pre-existing components”. These include:

- The Minisys Base Unit (with its photomultiplier tube and filter turret removed); the Minisys Controller (with the optional I/O card installed and wired); and the Minisys Electricals;
- The LN-CCD Camera; the LN-CCD Camera Controller; and the LN-CCD Camera Control PC and Software;
- The vacuum pump; and
- The shutter controller (essentially just a signal amplifier, required to drive the large high speed electronic shutter in the shutter housing).

The other components, including the PCIS Lens Unit; the Source Turret; the Shutter Housing; the IICU; the PCIS Sequence software running in the PCIS Control PC; the firmware and control tables
running in the IICU; and the IICU Management software – would all need to be bespoke developments, and are described in detail later in the document. Of these, only the IICU Management Software would prove to be a simple development.

It will be seen from the Figure that the Minisys Base Unit, Controller and Electronics have been left as much as possible connected as they originally were. The new PCIS Control PC and PCIS Sequence software communicate with the Minisys trio using the Minisys low level command language over the RS-232 Serial interface to the Minisys Controller. In turn, the Minisys Controller communicates with the IICU via the 8 Outputs and 16 Inputs of the installed optional I/O card.
The IICU then controls various items including the vacuum pump (note the wider arrow from the Minisys base unit to the vacuum pump represents the air flow, not a control circuit). The IICU also controls the synchronisation of image capture by the LN-CCD with the requirements of the whole PCIS, using three TTL level connections (Trig, Sync and NotScan) between the IICU and the LN-CCD Controller.

Importantly, the IICU controls the current, timing and duration of the electrical drive to the four strings of stimulation LEDs (infrared, red, blue and UV) in the source turret. The IICU also monitors the source turret and shutter housing for signals indicating either that the lens is not down on the Minisys and LN-CCD, or that excessive light conditions exist in the shutter housing (which might damage the detector if the shutter were allowed to open).

Control of the shutter is diverted from the LN-CCD Camera Controller via the IICU, which acquires full control of the shutter state. The requested state from the Camera Controller is just one of the inputs looked at by the IICU in determining whether and when to open the shutter.

The IICU also supports a number of individually configurable TTL level inputs and outputs, as well as a second switched power outlet, to support any additional ancillary equipment (such as external sources of optical stimulation) that particular experiments may require.

The manner in which the IICU exercises all of this control can be set and managed via the IICU Configuration Control software, which is shared between the IICU Configuration Control PC and the IICU Management Interface software.

We will now consider in detail the design, fabrication and assembly of the principle bespoke opto-mechanical components — the Optics assembly; the Source Turret; and the Shutter Housing — with a clear picture of where they fit into the overall structure of the PCIS, and of the roles each component must perform. The much vexed alignment of the optics assembly will also be described.

2.4.3 Guiding Design decisions

The first significant mechanical design decision, curiously, was the height of the base plate of the optics assembly. This decision determined the amount of space below the base plate into which to fit the source turret; and the space between the base plate and the base of the central mirror (M2) into which the focussing mechanism must be fitted (The expanded Offner design achieves focussing by the movement up and down by a few millimetres of M2. The exact range of movement required was not known at this point, and so a conservative design allowing at least +/- 5 mm or more was required. The final design allowed for approximately +/- 8 mm, and this has proved adequate.)

This decision made, the design of the source turret and of the optics assembly — the two most critical components of the opto-mechanical design — could now proceed independently.

It was also decided at this stage that the assembly of the PCIS (required whenever new samples are installed in the Minisys) would be achieved by firmly mounting the Minisys/Source Turret assembly and the LN-CCD/Shutter Housing assembly, and then lowering the lens onto them. The weight of the
lens assembly (in the end some 30 to 40 kg) would be sufficient, with appropriate design of the mating surfaces, to ensure both accurate mechanical alignment and adequate light-proofing.

Since a luminescence analysis of any material (whether to provide an age estimate or for other analytical reasons) implies hundreds of processing steps, and also because any such run produces enormous amounts of data (particularly so in an imaging system), it was clear that if the PCIS was to be of practical use then a comprehensive automation and data management capability would also be required. The lack of resources available to the project (myself – part time) meant that full use must be made of any existing capabilities, rather than re-developing such components. It was therefore decided to use as much as possible of the existing communications and automation capabilities of the Minisys.

Consequently it was decided to install the optional I/O board into the Minisys (providing 8 relay outputs and 16 optically isolated dry contact or TTL level inputs, addressable within the standard Minisys command set via the serial interface). The other components of the PCIS (including the stimulation LEDs and the LN-CCD interface) would then be controlled via the Minisys in a hierarchical architecture. This would provide two main advantages (in addition to the hoped-for reduction in development effort) versus a “flatter” design with the various components separately controlled from a single software suite:

1) Tighter control of the timing and synchronisation of the various components and functions of the PCIS (as against a design which put unpredictable serial communications delays into each branch of the control structure); and

2) The potential to avoid any low-level programming of the LN-CCD system, instead relying on the programmable parameters of the supplied generic “WinView” software.

The downside of this approach is the loss of the ability to change the parameters of the LN-CCD system during a PCIS run. However, on analysis it appeared that the only parameter that might usefully be changed during a run was the binning factor (which allows a trade-off between image resolution, and sensitivity or signal to noise ratio). Furthermore, a work-around exists by breaking a single run into multiple runs, and re-configuring the camera between the runs. It was decided that this was an acceptable loss, considered against the technical advantages and the reduced development effort of the hierarchical design.

This design further requires that the run name be entered twice – once into the PCIS control software, and again into the camera system software – as a means of managing and correlating the data produced by a single run. All sequences to date have been written so that the PCIS runtime software reminds the operator to enter the run name into the camera system software.

It was also clear that “proof of concept” of the mechanical and optical design – of the ability of the lens and camera system to provide the required spectral range, resolution, focus and above all sensitivity – would need an interim “quick and dirty” solution to the LED drive and system synchronisation issues. This interim solution would also need to provide the facilities to prove the viability of the intended final synchronisation and automation architecture, prior to commissioning the final electronics and developing the automation and data management software.

The various “beta test” interim components are also described in this thesis.
2.4.4 Implementation Phasing and Interim Components

The development of the PCIS was not an orderly process from requirements to architecture, through to the detailed design and construction of each component and subsystem, and finishing with system integration - assembly, alignment and testing. Indeed I have never seen a project – even a well-resourced and independently managed one – that was.

Rather, the development was a constant process of overlapping phases, emergent design and parallel activities, encompassing concurrent design, construction and testing at all different stages.

If this document were to present the whole of the PCIS development in a strictly chronological fashion the only result would be confusion. Living it chronologically often had that effect on me.

Unlike life, the order of a document is under some sort of control. The development of each component and subsystem is therefore presented in the sections that follow in a contained, independent and hopefully comprehensible fashion.

The order of development, and how the results of each stage helped determine the right direction for the next is still, however, an important part of the total story. This section attempts to explain the main aspects of the history and chronology of the PCIS development, without getting caught up in too much detail of the individual parts. It is my hope that the context this provides will assist in the understanding of the more detailed component descriptions that follow.

2.4.4.1 Overview

In broad terms the development of the PCIS can be seen as consisting of two main phases. In the first phase the major mechanical and optical components – the optics, source turret and shutter housing – were built to a usable, though not final, state. Interim electronics (the “lunch-box electronics”) were constructed to drive the LEDs installed in the source turret, and to allow the camera and shutter signals to be monitored and generated manually. Using the supplied software for both the camera (WinView) and the Minisys (seq117.exe and center.exe), this setup allowed the PCIS to be operated, albeit in a highly complex and inconvenient manual fashion.

The phase one system served two purposes: to act as a proof-of-concept for imaging luminescence, and for the broadband optics and LN-CCD combination in particular, proving that the PCIS could be a viable and useful research tool; and helping to establish the requirements of the final PCIS, especially in the area of system integration and automation.

Phase two saw the completion to final specifications of the major components, including the final (and successful) optical alignment. It also saw the development of the Integration and Interface Control Unit (IICU) and its associated software, and of the PCIS run-time software (pcis_seq.exe) and its associated sequence definition files. The PCIS Sequence Development Environment – consisting of the Developer’s Version of the PCIS_Seq software, and the PCIS Simulator hardware and software – completed the phase two development, and delivered the fully functional, automated PCIS.
2.4.4.2 Phase One – The Manual PCIS

The primary goal of phase one was to develop the main complex components of the PCIS – the broadband optics and the source turret – to the point that they could be tested, and to develop the detailed design of the PCIS so that a clear view existed of the final machine being built – its capabilities and its limitations – and of how to go about building it.

Before work could commence on the construction of any PCIS components, the overall optical and mechanical architecture had to be determined in some detail. At this stage too, the first ideas were beginning to emerge on how the final control or system architecture of the PCIS should appear. Any interim electronics built for phase one to support testing of the optics and turret assemblies would also need to support investigation and testing of the control architecture and synchronisation methods being considered.

Only such other components as were needed to support phase one operations (including the shutter housing and a basic camera support stand) were produced at this stage. To the extent possible, additional development was deferred until phase one had proved the underlying concept and capability, and the costs and benefits of developing the full automated PCIS could be assessed.

The Offner design was selected for the broadband optics, glass blanks were purchased, and the three required mirrors were figured, cut and coated. The metal components for the optics housing, and for the mirror mounting, aligning and focussing mechanisms, were designed and manufactured. The major components of the optics housing were anodised, and the optics were assembled and aligned (see "Alignment – First Attempt").
The source turret was designed and the metal parts fabricated, though only one filter drawer and the more important filter drawer inserts were made at this time. The source turret was assembled without anodising, and with only a subset of its LEDs installed. It would be disassembled later, and the parts anodised (including the extra filter drawer and inserts) prior to re-assembly with all of its contained electronics. The complete source turret assembly is a significant operation (which is described later), and the interim assembly saved a lot of effort while allowing the performance of the stimulation LEDs to be assessed.

The destination turret, or shutter housing, was also required for any operation of the PCIS, so it too was constructed (though not anodised, and without its final connectors and light detector). A simple stand was made for the LN-CCD camera, which brought it up to the approximate correct height, but did nothing to ensure its levelness and good mating with the optics (which had to be set up by hand for each run). Levelling the camera afresh for each run on a sloping and slightly curved standard laboratory bench proved so difficult that a milled flat aluminium alloy slab was also attached to the bench (and levelled), and the stand for the Minisys was upgraded to provide height adjustment.

![Image of LN-CCD camera with shutter housing](image)

**Figure 2.4.4** The LN-CCD camera with shutter housing attached, in the original simple stand.

In this configuration the PCIS was run, successfully, and useful data on samples of opto-electronic glasses was obtained (see Section 4). The optical alignment was revealed to be faulty (being able to give good positioning of the image or good focus, but not both), and would need to be re-thought. The overall sensitivity of the system, the quality of the imaging from the optics (on the off-centred images) and the sufficiency of LEDs for stimulation were all established, as was the viability of the intended control architecture.

All of the important questions about the viability and capabilities of the PCIS were now answered positively, and I proceeded to the development of the final automated PCIS.
2.4.4.3 Phase Two - The Automated PCIS

The main features of the phase two PCIS development were:

- The design of a new alignment jig and procedure to achieve the correct alignment of the broadband optics (see “Broadband Optics – Final Assembly and Alignment”);
- Completion of the source turret accessories, and the anodising and final assembly of the source turret with its full complement of electronics and internal wiring (see “Design and Development of the PCIS – Source Turret”);
- The design and construction of an internally programmed, computerised control unit for the PCIS and its internal interfaces (the IICU), and the development of its firmware and control tables (see “System Integration – Development of the IICU”);
- The development of the PCIS run-time software (PCIS_Seq) to enable unattended, automatic operation of the PCIS via simple text files defining the sequence, and of the associated suite of sequence file development tools (see “Development of the PCIS Sequence Software”);
- The development of a library of sequence support files for specific functions of the PCIS, and the complete configurable sequence implementations for the Single Aliquot Regeneration (SAR) protocol, and a Thermoluminescence Probe sequence (TL_Probe) developed for the broadband scanning of materials, and the identification and isolation of any emitted luminescence signals.

There were also a range of smaller improvements and modifications made to the ancillary components of the PCIS. These included a new camera stand to hold it in better alignment to the optics (and to make the LN filling operation easier); Better procedures for levelling and aligning the Minisys base unit, the LN-CCD camera and the broadband optics; and the development of a complete hardware emulation of the Minisys and CCD controller interfaces to support acceptance testing and control table development for the IICU.

![Figure 2.4.5 The IICU in the test bed environment built for control table development and testing.](image)

In its final form the PCIS is reliable, repeatable, and simple to set up and run for standard tasks, while retaining the ability to be programmed for any task within its fundamental optical capabilities.
2.5 Broadband Optics Assembly

The role of the broadband optics assembly in the PCIS is to capture as many as possible of the photons emitted by a sample, across as wide as possible a portion of the spectrum to which the LN-CCD is sensitive, and to focus those photons as accurately and consistently as possible, and with as little loss as possible, onto the detector – an area just slightly larger than the sample.

It was shown above why these requirements led to the selection of a three mirror reflective optics design of the Offner style.

A consequence of selecting a reflective design is that the tolerances are a lot tighter than for a corresponding refractive design, not only on the shape of the element, but also on the accuracy of mounting and alignment of the surfaces. The physical design of the lens box, and the procedures for aligning the optics, would therefore become critical factors in the overall success of the project.

The implementation, construction and alignment of the Offner optic system are described below.

2.5.1 The Expanded Offner Design

The Offner design comprises three spherical surfaces (mirrors) designated, for simplicity, M1, M2 and M3. These are arranged to produce a 1:1 image, some distance away from and in a plane parallel and roughly congruent to that of the sample. (The surfaces are not truly congruent due to the effects of different thicknesses of filter glass and vacuum windows in the two light paths.)

As oriented in the PCIS, the concave M1 sits some 400 mm directly above the sample, and concave M3 a similar distance above the detector, both mirrors facing downwards. M2 is a convex mirror facing upwards, about half way between the sample and the detector, and about 100 mm above them. Focus adjustment (required for changes of filter or changes in sample thickness) is achieved by moving M2 up or down. Refer to Figure 2.5.1 for a perspective view of this arrangement.

![Diagram of Offner System](image)
In the case of the PCIS the design was “expanded” to minimise spherical aberrations, whilst accommodating the necessary spacing between sample and detector. Consequently the sphere centres of M1 and M2 are not congruent, but are separated by about 1.5 mm.

The three mirrors are rectangular cut-outs of a spherical section. M1 and M3 (concave) are approximately 350 mm by 130 mm with a 345 mm radius of curvature, and M2 is approximately 170 mm by 62 mm with a 172.5 mm radius of curvature. These large sizes for their relatively small radii of curvature would make coating of the mirrors a technical challenge that BAE of Adelaide was the only Australian company willing – and with the facilities – to accept.

The design supplied by Damien Jones of Prime Optics included a table of the required x, y, z co-ordinates of the mirrors with respect to the sample and detector, along with tolerance specifications for the surface accuracy and alignment of the mirrors. The design also included simulated images displaying the calculated sharpness or focus performance of the lens for different parts of the field of view. (The design field of view is a 12.5 mm square, matching the size of the CCD detector.) These tables are reproduced as Figures 2.5.2 to 2.5.4.

### TABLE 1a: Offner Specifications
(as at 08 August 2000)

<table>
<thead>
<tr>
<th>Surf</th>
<th>Ident</th>
<th>Y mm</th>
<th>Z mm</th>
<th>xRot deg</th>
<th>Radius mm</th>
<th>Next medium</th>
<th>Aperture/Term</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>BnP</td>
<td>9396.264</td>
<td>0.000</td>
<td>-90.549</td>
<td>plane</td>
<td>AIR</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>Sce</td>
<td>0.000</td>
<td>90.000</td>
<td>-90.000</td>
<td>plane</td>
<td>AIR</td>
<td></td>
</tr>
<tr>
<td>1 Window</td>
<td>19.800</td>
<td>90.000</td>
<td>-90.000</td>
<td>plane</td>
<td>SiO2</td>
<td>71.0</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>23.800</td>
<td>90.000</td>
<td>-90.000</td>
<td>plane</td>
<td>AIR</td>
<td></td>
</tr>
<tr>
<td>3 Filter</td>
<td>24.800</td>
<td>90.000</td>
<td>-90.000</td>
<td>plane</td>
<td>BK7</td>
<td>44.0</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>27.800</td>
<td>90.000</td>
<td>-90.000</td>
<td>plane</td>
<td>BK7</td>
<td>44.0</td>
</tr>
<tr>
<td>5 Filter</td>
<td>28.800</td>
<td>90.000</td>
<td>-90.000</td>
<td>plane</td>
<td>AIR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>31.800</td>
<td>90.000</td>
<td>-90.000</td>
<td>plane</td>
<td>AIR</td>
<td></td>
</tr>
<tr>
<td>7 M1</td>
<td></td>
<td>351.691</td>
<td>0.000</td>
<td>-90.130</td>
<td>-345.000</td>
<td>AIR</td>
<td></td>
</tr>
<tr>
<td>8 M2</td>
<td>Window</td>
<td>182.418</td>
<td>0.000</td>
<td>-90.000</td>
<td>-172.500</td>
<td>AIR</td>
<td></td>
</tr>
<tr>
<td>9 M3</td>
<td></td>
<td>351.691</td>
<td>0.000</td>
<td>-89.870</td>
<td>-345.000</td>
<td>AIR</td>
<td></td>
</tr>
<tr>
<td>10 Window</td>
<td>15.659</td>
<td>-90.000</td>
<td>-90.000</td>
<td>plane</td>
<td>SILICA</td>
<td>25.0</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>14.639</td>
<td>90.000</td>
<td>-90.000</td>
<td>plane</td>
<td>AIR</td>
<td></td>
</tr>
<tr>
<td>12 Det’r</td>
<td>3.459</td>
<td>-90.000</td>
<td>-90.000</td>
<td>plane</td>
<td>AIR</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 2.5.2 Final lens component location and dimension calculations (D. Jones).
Table 3: Component Tolerances

<table>
<thead>
<tr>
<th>Ident</th>
<th>Axial Displacement (mm)</th>
<th>Tilt (min)</th>
<th>Figure (Fr@Na/HeNe)</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1, M3</td>
<td>0.25</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>M2</td>
<td>0.25</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>Filters</td>
<td>2.00</td>
<td>15</td>
<td>2</td>
</tr>
</tbody>
</table>

Figure 2.5.3 Tolerance specifications for the lens components (D. Jones).

The following diagram (Figure 2.5.4) shows the expected image appearance for a point source at various locations within the sample area, identified with respect to the long axis (front and back) and the short axis (left / right) of M1.

Figure 4c: Dioptric (Offner) System spot diagrams

Figure 2.5.4 Detector imagery predictions for sample centre, edge and corner spot sources (D. Jones).
The points (identified by their millimetres offset from the centre of the sample area) are, reading from top left to bottom right: centre of outer edge; outer edge front or back corner; centre of sample area; centre of front or back edge; centre of inner edge; inner edge front or back corner. Note that while the inner and outer edges of the sample (as with the corresponding edges of M1 and M3) differ from each other due to the nature of the Offner design, the front and back are symmetrical, and the corresponding optical performance predictions would be identical.

Below (Figure 2.5.5) we see all 3 mirrors of the Offner design, after grincing and polishing, cutting to shape, and having their reflective coatings applied. The application of accurately thickness-controlled layers of reflective and coating materials onto the mirrors was made significantly more difficult by the large size and extreme curvatures. Figure 2.5.6 shows the author measuring the cut and coated mirrors while designing the support and adjustment mechanisms that would suspend them from the top of the lens box.

Figure 2.5.5  All three mirrors, with the mounting plate for M2. Note the large sizes and extreme curvatures.

Figure 2.5.6  Mirror 1 (or 3?) being measured for mounting after polishing and coating.
2.5.2 Manufacture of the Mirrors

Once the optical design had been selected, Dr Spooner and I ordered the blanks from which the mirrors would be produced, and I commenced the detailed design of the lens housing and other opto-mechanical components. The grinding, polishing and coating of the mirrors took several months, and overlapped in time with much of the work described in the following sections.

The material selected for the mirrors was Pyrex™ glass. Forming and polishing of the mirrors would be undertaken by Gabe Bloxham at the Mt Stromlo Observatory optical workshop. We would also need to find a facility able to coat mirrors of such large size and high curvatures with the necessary precision and consistency.

Pyrex™ was selected for a number of reasons:

- It has a low co-efficient of thermal expansion compared to other glasses;
- It can be readily formed into large, bubble-free slabs;
- For the above reasons it is the material of choice for reflective telescopes, and is therefore readily available in the sizes that we needed, and for an affordable price; and
- Because Pyrex™ is so popular for telescope mirrors, those who grind mirrors are very familiar with it, and its grinding and polishing properties are particularly well understood.

Three 16” diameter by 4” thick (for M1 and M3), and two 12” diameter by 3” thick (for M2) optical grade Pyrex™ glass blanks were purchased. Each spherical section mirror, whether concave or convex, generally requires that its complement also be ground and polished along with it – that’s just how the process is done. In this case the same convex counterpart would be used for both M1 and M3, so only the three larger blanks were required. This not only saved some money, it had the more important effect of ensuring that M1 and M3 came out with exactly matching radii.

The blanks were delivered to Gabe at Mt Stromlo, along with the mirror specifications from Damien Jones (with which Gabe was already familiar; he had advised during the study and selection of the Offner design), and he began to work his very special magic.

The production of the mirrors took several months, partly due to the need to give the large glass blanks several hours before each measurement of the surface, prior to deciding the arrangement for the next stage of polishing. The measurement facility comprised a long optical table, with an arrangement allowing diagonal diffraction lines to be generated by the (uncoated) mirror surface through the eyepiece. Distortion of the lines out of the straight would represent an imperfection of the surface; the specified tolerance here was 2 diffraction fringes in Sodium light. In final form, and when fully temperature stabilised, the distortion amounted to less than half a Sodium fringe (except at the very margins of the mirror, where cutting stresses inevitably introduce distortions, but affecting less than 1 percent of the lens surface). As soon as a hand was placed on the back of a mirror, introducing thermal variation, the diffraction lines would become severely distorted – significantly in excess of the allowed two fringes. Temperature stability of the optics would obviously be important in their final operating environment.

Progress of the convex M2 was tested by measuring its concave polishing counterpart.
During the grinding and polishing process, when the mirrors were being cut to their final rectangular shapes, a slot was cut at my request into each end of M1 and M3. This would support the attachment of a safety “catcher cable” which would hopefully prevent the complete loss of a mirror if its glued mounts should fail. I trust glues, but not more than I need to.

When the polishing was completed, the mirrors were carefully packaged, and delivered (along with some special mounting jigs) to British Aerospace Engineering (BAE) in Adelaide, who owned what appeared to be the only facility in Australia capable of coating such large and high curvature mirrors to the required standard. The mirrors were coated with Aluminium and then with appropriate protective layers, providing good long term reflectivity throughout the required spectral range. The supplied reflectivity performance measurements from 200 nm to 1000 nm appear as Figure 2.5.7.

![Figure 2.5.7 Reflectivity characteristics of the mirrors (BAE).](image)

The mirrors were returned with a chip – about the size of my small fingernail – missing from one of M1/M3, but otherwise undamaged, and finally looking like mirrors! The mounting plates – presumably seen as a one-off and disposable – were not returned with the mirrors, but had now done their job.
2.5.3 Housing and Optical Mating

The selected Offner optical design, and the nature of the existing Minisys base unit and LN-CCD camera, meant that a rectangular prism shape – in fact very close to a cube – would be the obvious shape to use, and would be the simplest to design and manufacture. This structure would support the upper mirrors suspended from its upper face, and the lower mirror (M2) via some focussing mechanism mounted on the lower face. The lower face would also have two large ports for photon entry and exit, with light tight mating rings, enabling the optics to rest on the Minisys and LN-CCD, and maintaining alignment by the optics’ own weight.

In view of the need for rigidity to maintain the orientation of the mirrors, and the desirability of weight, the selected design paradigm for the optics housing was over-engineering. Accordingly (and because the material was left over from an earlier job and was therefore free), the approximately 390 mm cube of the optics housing was constructed entirely from 12 mm aluminium alloy plates. Each plate has machined steps at each edge, which interlock in a fashion to act as a light baffle and exclude extraneous light from the optics. Any such light would be well off axis, and unlikely to reach the detector (hence the open framework often used for reflective telescopes). However, considering the extremely low photon levels we were hoping to detect and measure, complete exclusion of stray light seemed the safer course. Indeed, it may have been better to include O-rings at every mating surface, but this seemed excessive at the time, and experience has shown that the design chosen is adequate.

The front, back and side plates of the box are relatively simple, including only holes for the placement of handles; for the attachment of a lens lifting bracket at the back of the box (not currently used); and for attaching lifting chains to the upper corners of the side plates. The upper and lower plates are more complex.

The upper plate includes six machined and recessed ports for the placement of the M1/M3 adjustable mounts, along with associated holes for alignment dowels and attachment bolts; four threaded holes for attachment of the version 2 optics alignment laser guide; and a central hole for the laser alignment guide plug.

The lower plate of the lens box – the Optics Assembly Base Plate – is quite complex, and was the first component designed. The choice of its placement and design would affect much of the detailed design of other components. It is discussed in detail in the following section.
2.5.3.1 Optics Assembly Base Plate

The positions of the sides, back and top of the optics box were not critical, so about 1 cm was allowed outside the mirror dimensions to allow installation clearance, and possible room for bolt heads (a little more – around 2 cm – at the top, to allow for adjustment mechanisms in the M1 and M3 mounts, necessary for aligning the optics).

The position of the base plate however would be critical. With the top of M2 approximately 180 mm above the sample, and M2 likely to be about 50 mm thick, there was only about 130 mm between the base of M2 and the sample – itself a few millimetres thick. Any placement of the base plate would need to fit, from the bottom up:

- the source turret, containing (in order) the vacuum window, the stimulation LEDs, and the replaceable optical filter assembly;
- a mating ring to ensure mechanical alignment and light exclusion;
- the base plate itself;
- the focussing mechanism to raise and lower M2; and
- a mounting plate for M2.

The decision was made to place the inner face of the base plate 100 mm below the top of M2 (82.4 mm above the sample), leaving 50 mm above the plate for the M2 mounts and focussing gear; and about 70 mm below the plate for the source turret and mating ring. (A similar space would also be available for the destination turret/shutter housing, but that was not a limiting constraint.)

Any higher placement of the base would limit not only the vertical space available for the focussing mechanism, but also the width available between the entry (source) and exit (destination) ports, as the optic path becomes wider as you go higher. Any lower placement of the base would lead to difficult interference problems between the optics and the lid lifting mechanism on the Minisys, as well as limiting the design of the source turret. Even at the chosen height, a few millimetres would need to be filed off one point on the lower left edge of the optics box to prevent interference with the angled arm that attaches the Minisys sample chamber lid to the lifting mechanism.

The broad shape of the optics box could now be drawn – see Figure 2.5.8 below – and precise calculations made to determine the size, shape and location of the entry and exit ports. In Figure 2.5.8 the adjustment and focussing mechanisms for all three mirrors are conceptual only – it is known at this stage that such items must exist, and must be placed essentially as shown, but their details have yet to be designed.

Note that the 15 degree clockwise rotation of the optics relative to the Minisys – required to provide physical clearance between the optics and the Minisys cover lift mechanism – has no implications for the design of the optics. It may equally be considered to be a widdershins rotation of the Minisys relative to the rest of the PCIS (accommodated where the source turret bolts to the Minisys cover plate). Since the sample is circular, and without any particular orientation, the rotation of the Minisys base unit may be – and is – ignored.
The pages reproduced below as Figures 2.5.9 to 2.5.11 show the process of calculating the final size and placement of the first of the two optical ports in the base plate (the source aperture). The actual calculated shape of the intersection of the base plate with the pyramid-like shape defined by the available light paths between the sample and M1 is not rectangular, or even rhomboid, but rather a distorted, curved collar shape. The final design for the apertures is the smallest rhombus (allowing for 3 mm radius curved corners) that will contain the calculated shape. Additional calculations derive the required chamfer angle for each of the four sides of the rhombus.

Corresponding calculations were undertaken for the destination aperture, but are not reproduced here. The slightly larger detector area (compared to the sample area), and the different thicknesses of transmissive elements in the two sides, lead to slightly different figures for the destination aperture. Also, the chamfers around the destination turret are cut so that they are larger than the calculated “light cone” at the lower surface of the base plate (rather than the upper surface, as for the source aperture).

The purpose of these severe chamfer angles on the light ports is to ensure that no photons coming from the preceding optical stage (the sample in the case of the source aperture; M3 in the case of the destination aperture) can strike the inner face of the aperture (where it cuts through the 12 mm thick base plate), be reflected on to the next optic stage, and eventually reach the detector. The port cut-out inner faces are chamfered so as to be invisible from the preceding optical stage.
Figure 2.5.9 Initial planning for Source Aperture location and dimensions.
### Development and Application of a Photon Counting Imaging System

**Source Aperture**

<table>
<thead>
<tr>
<th>Inner Corners</th>
<th>Outer Corners</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Corner of Sample</strong></td>
<td><strong>Centre:</strong> (0,0, -95)</td>
</tr>
<tr>
<td>(±5, θ, -85)</td>
<td>(±5, θ, -95)</td>
</tr>
<tr>
<td><strong>Corner of Mirror</strong></td>
<td><strong>Centre:</strong> (0,318.08, -149.33)</td>
</tr>
<tr>
<td>(±175, 303.82, -11.24)</td>
<td>(±175, 270.97, -137.02)</td>
</tr>
<tr>
<td><strong>Aperture @ Source Face (Y=100)</strong></td>
<td><strong>Centre:</strong> (0,100, -112.08)</td>
</tr>
<tr>
<td>(±60.95, 100, -60.72)</td>
<td>(±67.74, 100, -110.51)</td>
</tr>
<tr>
<td><strong>Aperture @ Outer Face (Y=87)</strong></td>
<td><strong>Centre:</strong> (0,100, -112.08)</td>
</tr>
<tr>
<td>(±53.68, 87, -63.99)</td>
<td>(±58.58, 87, -108.49)</td>
</tr>
<tr>
<td>θ Y = 90:</td>
<td>(±61.46, 100, -108.98)</td>
</tr>
<tr>
<td>(±55.36, 90, -63.15)</td>
<td><strong>Centre:</strong> (0,87, -109.86)</td>
</tr>
<tr>
<td><strong>Cross-Face Angle in Z-Dim'n</strong></td>
<td><strong>Centre:</strong> (9, 40, -110.27)</td>
</tr>
<tr>
<td>from (x, z, -86) to (x,87, -62.88)</td>
<td>(2, 57)</td>
</tr>
<tr>
<td>= tan⁻¹(31.12/87)</td>
<td>= 18.68° from vertical</td>
</tr>
<tr>
<td>≈ 19.68° from vertical</td>
<td><strong>Centre:</strong> (5, 87, 63.99)</td>
</tr>
<tr>
<td><strong>Cross-Face Angle in X-Dim'n</strong></td>
<td><strong>Centre:</strong> (±15.11° from vertical)</td>
</tr>
<tr>
<td>from (5, 0, z) to (x,87, -63.99)</td>
<td>(±15.11° from vertical)</td>
</tr>
<tr>
<td>= tan⁻¹(68.66/87)</td>
<td>≈ 36.59° from vert.</td>
</tr>
</tbody>
</table>

**Using x-face angle**

<table>
<thead>
<tr>
<th>Z-angle @ inner = 25°</th>
<th>Z-angle @ outer = 20°</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Aperture @ Inner Face (Y=100)</strong></td>
<td><strong>Centre:</strong> (±64.59, 100, -67.82)</td>
</tr>
<tr>
<td>BUT USE (±65, 100, -56)</td>
<td>(±70.49, 100, -113.22)</td>
</tr>
<tr>
<td><strong>Max. Aperture @ Outer Face (Y=90)</strong></td>
<td><strong>Centre:</strong> (±57.51, 90, -60.66)</td>
</tr>
<tr>
<td>BUT USE (±57, 90, -112.26)</td>
<td>(±63.61, 90, -111.36)</td>
</tr>
</tbody>
</table>

**Corners of Source Aperture can be cut to a radius of (up to) 3 mm.**

---

Figure 2.5.10 Intermediate calculations of Source Aperture location and dimensions.
### Source Aperture

<table>
<thead>
<tr>
<th>Source Aperture</th>
<th>Inner Corners</th>
<th>Outer Corners</th>
<th>Outer Edge Centre</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sample</td>
<td>(±5, 0, -85)</td>
<td>(±5, 0, -95)</td>
<td>(0, 0, -95)</td>
</tr>
<tr>
<td>Mirror</td>
<td>(±175.30382, -11.2)</td>
<td>(±175.270.97, -137.02)</td>
<td>(0.318.08, -149.33)</td>
</tr>
<tr>
<td>Aperture @ Y = 90</td>
<td>(±55.36, 90, -63.15)</td>
<td>(±61.46, 40, -108.96)</td>
<td>(0, 90, -110.37)</td>
</tr>
</tbody>
</table>

#### X-face and Z Dim in Z Dimin (* from Vert)
- From (±5, 0, -95) to (±5, 0, -95)
  - tan⁻¹(±5/90) = 19.49°
- From (±5, 0, -95) to (±5, 0, -108.96)
  - tan⁻¹(±5/90) = 19.49°
- From (0, 0, -95) to (0, 90, -110.37)
  - tan⁻¹(0/110.37) = 15.74°

#### X-face and Z Dim in X Dimin (* from Vert)
- From (5, 0, ±z) to (-5, 36, 90, ±z)
  - tan⁻¹(5/36) = 33.85°
- From (5, 0, ±z) to (-61.44, 90, ±z)
  - tan⁻¹(5/61.44) = 36.46°

#### Allowance on X-face ±z
- Use: Z-A @ inner ± 25° X-angle = 40°

#### Project onto Y = 100
- (±63.75, 100, -58.49) → (±69.85, 100, -112.60) → (0, 100, -114.01)

#### Add 1.15m all round.
- (±65.0, 100, -57.5) → (±71.1, 100, -115.0)

#### New X-f.d in Z @ Y = 100
- From (±5, 0, -95) to (±5, 0, -95)
  - tan⁻¹(5/95) = 16.70°
- From (0, 0, -85) to (0, 100, -115.0)
  - tan⁻¹(0/115.0) = 16.70°

#### New X-f.d in X @ Y = 100
- From (5, 0, ±z) to (-5, 0, 100, ±z)
  - tan⁻¹(5/100) = 34.99°
- From (5, 0, ±z) to (-7.10, 0, ±z)
  - tan⁻¹(5/7.10) = 37.23°

#### Project onto face Y = 90
- (±56.61, 90, -62.16) → (±62.61, 90, -111.36)

---

Cut Source Aperture from Top or Inner Surface of Base Plate (Y = 100) at corners & angles:

**Inner Corners**: (±65, 0, ±57.5)

**Inner Edge**: at angle of 25° from vertical

**Outer Corners**: (±71, 0, ±115.0)

**Outer Edge**: at angle of 20° from vertical

"End of Aperture" Edges: at angle 40° from vert.

All angles cut to make Aperture smaller at Bottom Surface.

---

Figure 2.5.11 Final Source Aperture location and dimension calculations.
The under-side of the lower plate also has attached to it the mating rings for the source and destination turrets. These mating rings will be “mirrored” at the upper edge of the two turrets, but with the addition of an O-ring to ensure better light exclusion. The O-ring will bear against the step of the mating ring that is not pierced for the mounting bolts (see Figure 2.5.12 Mating Ring Cross-section). The mating surfaces of the turrets are a part of the turrets themselves, and do not require mounting bolts that might otherwise interfere with the O-ring.

The mounting rings are located to the underside of the lens base plate using dowels, to ensure the accurate placement of the sample and detector with respect to the optics. There is a 0.2 mm clearance built in to the design of the mating rings and the turret assemblies, allowing clearance for fitting the optics to the Minisys and LN-CCD when the PCIS is run, but not allowing sufficient movement to affect the sample-to-detector alignment. The 1.0 mm radius curve shown on the inner lip of the mating surface also assists with correct location of the optics.

![Figure 2.5.12 Optics assembly Base Plate showing all dimensions and overlay of M2 and focussing gears.](image)

The “lower step” in Figure 2.5.12 (with the mounting bolt holes through it) is designed to come into direct contact with the corresponding part of the source turret or shutter housing, and to ensure precise mechanical and optical alignment of the parts. The “upper step” in the diagram is designed to have a 0.1 mm clearance to the turret or housing, where an O-ring is protruding by 0.2 mm from the corresponding surface. The 0.1 mm compression of the O-ring ensures proper light exclusion.

The mounting ring for the source turret is rectangular (with rounded corners to further aid the location of the optics). There are 2 reasons for this. One is to save space under the base plate for the focussing assembly that has yet to be designed. The other, and over-riding reason, is that the optics assembly must maintain a rigid orientation with respect to the Minisys – 15 degrees clockwise rotation with respect to the Minisys when viewed from above. This rotation is set by the attachment of the source turret to the Minisys base unit, so that the mounting ring can be (and is) aligned to the base plate major axes.
The destination turret mounting ring – under which the LN-CCD camera sits – was made circular, in order to allow the LN-CCD camera to be turned, if necessary, under the optics. It was thought that this facility may be required in order to bring samples that were rectangular arrays of grains into alignment with the X- and Y-pixel dimensions of the camera, thereby simplifying the digital extraction of square areas of data relating to each grain. Although this has not been a requirement (better techniques of data extraction based on circles of data relating to the grains were devised later), the circular port has rendered easier the correct placement of the camera, and lowering of the lens assembly onto the Minisys and camera when running the PCIS.

The 15 degree offset of the lens with respect to the Minisys was calculated to be just sufficient to allow the lens assembly to clear the Minisys motor tower and top plate support arm, while still allowing the LN-CCD camera to clear the edge of the Minisys base unit when turned under its circular mounting ring. It was calculated that 2 or 3 mm of material would need to be filed from the lower left edge of the lens assembly to clear the Minisys lid support arm, and that the camera – in its yet-to-be-designed cradle – would have 5 to 10 mm clearance on the Minisys at its critical angle of 45 degrees to the Minisys. These calculations were (fortunately) borne out in practice, and the camera may actually be rotated through more than 90 degrees, even while the PCIS is running.

The mounting rings are recessed 2 mm into the lower surface of the base plate to gain back those 2 mm for the source turret design, and to aid in the exclusion of possible light leaks between the mounting ring and the optics base plate. This can be seen in the base plate cross-section (Figure 2.5.13). The source turret mating ring is shown below. The destination turret mating ring is similar, but circular and, therefore, slightly larger.

![Figure 2.5.13 Optics assembly Base Plate Source Turret Mounting Ring.](image)

The optics assembly base plate design is now complete for the moment. The mounting, alignment and focusing assembly for M2 has yet to be designed, and is described later. However, the space into which this assembly will need to fit is now defined.
Figure 2.5.14 below shows the final base plate design. Shown are the source and destination ports as well as the corresponding turret mounting rings. Also shown is the position of M2, and the focussing gear assemblies as they were finally designed, clearly showing the space constraints with which the focussing assembly design would have to contend. The focussing assembly would require five holes to be machined into the optics assembly base plate, each fitted with a custom-designed brass bush.

Finally, note the large number of bolt placements. This is repeated around the optics assembly – requiring a total of 90 bolts – to ensure both mechanical rigidity and also good light exclusion.
2.5.4 M1 and M3 mounting and Adjustment

Mirrors M1 and M3 – the larger, concave mirrors – are suspended from the top plate of the optics assembly. Although they do not have to move for focusing, the need for initial alignment, to very close tolerances (0.25 mm axial displacement and 1 minute-arc of tilt) would require precision adjustable mounts.

The choice was made to use three screw threaded adjusters with custom-built ball joint assemblies on the back of each mirror, with fixed, internally threaded brass parts mounted to the lens box top plate for them to turn in. Turning the threaded shaft (from outside the lens assembly, using a screwdriver) would cause it to move in and out of the fixed mount point, thereby adjusting the angle of M1 or M3. Turning all 3 adjusters for one mirror equally would cause axial movement of the mirror. Lock nuts would prevent further movement once the alignment was complete.

The support posts were machined from 0.75 mm pitch (by 12 mm) bolts. A specially adapted screwdriver was used to adjust them. This had a side bar of approximately 230 mm length welded to it, yielding a circumference of movement at the end of the bar of 1.5 m. This created a simple 2000:1 ration between movement at the end of the screwdriver handle, and axial movement at that part of the mirror. Thus the axial placement could easily be adjusted by increments of 5 or 10 microns (being 1 to 2 cm at the handle), yielding an ability to adjust the tilt on the mirror by increments of 0.3 minutes of arc or better – exceeding the specified requirements.

![Figure 2.5.16 Optics assembly Base Plate showing all dimensions and overlay of M2 and focussing gears](image)

The ball joint assemblies would need to be glued to the flat back of the mirrors, either separately or as a single assembly. Since the mirrors M1 and M2, each 2 kg or more in weight, were to hang rigidly and immovably from these mounts, a strong thin-film glue would be needed. (In the end super glue was used.)
Because of the thin inflexible glue layer, differential thermal expansion between the glass and metal components would be a problem if a large single metal mount were used – potentially leading to the ultimate separation of mirrors and mounts. Accordingly three separate mounts, each machined from a 30 mm diameter disc of 12 mm alloy, were used on each mirror.

The bolts were each turned down to a 10 mm sphere at one end. (The opposite end was cut off and a flat screwdriver slot machined in.) The back-of-mirror mounts each comprised three parts: a 30 mm diameter disc of 10 mm aluminium plate with a 10 mm diameter hemisphere machined in the centre of one face; and 2 semicircular components, obtained by taking a similar disc-shaped component, drilling a 5 mm hole through the centre, and then cutting it into two semicircular pieces. By assembling these three components around the machined spherical end of each bolt (using locating dowels as well as small bolts), six full ball joint assemblies were created. These were each assembled, and then lapped by hand to ensure they operated smoothly and without any excess play.

These would be glued three to the back of each of M1 and M3. Two would be at the extreme ends of the mirror edges closer to each other (at the centre of the lens assembly), since this is the high side of the installed mirrors and would support most of their weight. One adjustable support would be placed midway along the outer edges of the mirrors, providing an adjustable 3-point stand-off for each mirror.

The height of the top plate of the lens box was now confirmed, as allowing sufficient room above the “centre” edges of the mirrors for the upper and lower components of the adjusters; and allowing the outer adjusters to just fit within the side walls of the lens box (1.33 mm clearance – see Figure 2.5.16 above and Figure 2.5.17 below).

![Figure 2.5.17 Optics assembly showing placement of M1 and M3 mounts.](image-url)
2.5.4.1 Attaching the Mounts to M1 and M3

According to the design, the mounting pads for M1 and M3 needed to be glued to the backs of the mirrors symmetrically around the “optic centre” of each mirror. Given that M1 and M3 are spherical sections with true flat backs, the optic centre is simply the thinnest point of the mirror. The location of each optic centre relative to the geometric centre (the intersection of two diagonals drawn on the back of the mirror) had been marked by Gabe Bloxham with a different coloured spot on one corner of each mirror, and notes as to the offset of the optic centre from the geometric centre relative to the spotted corner. This would allow the correct position for each pad to be calculated relative to the identified edges of each mirror.

While the M1 and M3 mounts (described above) allow radial movement of the mirror as well as adjusting the angle, there is no adjustment for lateral displacement. Any inaccuracy in the placement of the mounting pads would therefore have two effects.

1. In effect a slice along one edge of the intended reflective surface would be missing, replaced by a (presumably ineffective, or at least less effective) slice added to the opposite edge. Given the large size of the mirrors, this effect would be trivial for any sub-millimetre displacement of the mirrors.
2. Since the M1 and M3 mounts allow pivoting only at the mirror end, and are rigidly fixed where they mount to the lens box, any misalignment would introduce lateral stresses into the mounts and could, ultimately, cause their glued attachments to the mirrors to fail. An error in the shape of the pattern of attachments would lead directly to such stresses; any error in the position of the pattern of attachments would require that the angle of the back of the mirror be adjusted (to bring the front face into coincidence with the correct sphere). Any large angular adjustment would reduce the apparent separation of the mounting pads (viewed from the perspective of the mounting points on the lens box), and would thus indirectly introduce stresses into the mounts.

The glueing of the mounting pads to the backs of M1 and M3 would clearly require the assistance of an alignment jig to ensure accurate placement. The target for placement accuracy would be matched to the general machining tolerances applied during the construction; +/- 0.03 mm.

This stage of the design produced one of the few wrong turns of the project. An attempt was made to design a plate which could be positioned on the back of each mirror, and against which the pads could be aligned whilst glueing. The plate would then be carefully “weaved” out from between the pads, leaving the pads in place while the glue dried. The original design required repeated modifications and adjustments in the effort to find a shape that would both give the required placement support, and allow its own removal without disturbing the pads. As Figure 2.5.18 shows, the envisaged plate became more and more complex, and the possibility of removing it without disturbing the pads more and more remote.

Now there is a general principle, in technology if not in science, that when solutions are getting more complex, and the final result doesn’t seem to be getting any closer, that you are probably headed up a dead end, and need a fresh approach (fusion energy and speech recognition seem to fit this category). The mirror mounts alignment jig seemed to be a case in point.
If the investment is large, and some successes seem to have been achieved (as in the examples cited above), then it can be extremely difficult, if not commercially impossible, to abandon the progress achieved and make a fresh start with new ideas. Fortunately in this case there was no more investment than a few drawings, which I happily abandoned in favour of a fresh, and much simpler approach.

It would be better to have the mirrors facing upwards, resting on the mounting pads during glueing, rather than the other way around, as this would both allow the mirror’s weight to hold the parts together while the glue set, and also avoid the risk of damage to the mirrored surface. However, the pads must be held in full contact with the back of the mirror, so that if the pads rested on a surface with the mirror resting on them, then the surface would have to be supremely flat to match the back of the mirror.

Or – and this was the innovative part of the idea – the pads could be made to align themselves to the back of the mirror under the same mirror weight that would hold them together while the glue set. This could be achieved by using the hemisphere machined into the back of each pad as a way of allowing the pad to rotate and align itself correctly to the back of the mirror.

So the revised and much better alignment jig shown at Figure 2.5.19 was designed and made. It consists of a single (moderately flat) plate of aluminium, with three conical “divots” and three blind holes. The blind holes will contain tall dowels against which the mirror can be pushed (two dowels against a long edge; one dowel against a short edge), ensuring accurate placement of the mirror relative to the jig. A 10 mm ball bearing was placed in each of the conical divots; and a mounting pad
placed with its hemispherical cavity over the ball bearing. This firmly fixed the position of each pad, while still allowing the pads to tilt and come into proper alignment with the back of the mirror.

Figure 2.5.19 Final – and better – alignment jig for attachment of M1 and M3 mounts.

Using the measurements supplied by Gabe Bloxham, shims could be calculated which, placed between the dowel posts and the mirror edges, would allow the mounting pads to be correctly placed with respect to the optical centre of each mirror, using the one alignment jig.

The jig, the shims and ball bearings, and the mirrors were taken to the Mt Stromlo Optical Workshop, where Gabe Bloxham used his experience with reflective optic construction (and the glues appropriate thereto) to assist with the gluing of the pads to M1 and M3. The glue of choice for a hard permanent attachment between clean aluminium alloy and glass was a cyanoacrylate, commonly known as super glue.

The attachment was a great success, and when the glue dried the rest of the M1 and M3 mounts were attached to the mirrors. (Note that each mount has been individually lapped in to operate smoothly and avoid stresses to the mirror mounts during adjustments. The parts of the different mounts are numbered, and must not be mixed up during assembly.)

The photos below show the details of the M1 and M3 mounting and alignment assemblies. Note that these photos were taken during a later disassembly of the lens and cleaning of the mirrors, with the mirrors already mounted to the lens box top plate. During the initial assembly of the optics (described later), mirrors M1 and M3 are mounted to the lens box top plate with the lens box
already largely assembled. For the later re-assembly shown below, the mirrors and top plate were removed, cleaned and re-installed as a single assembly.

Figure 2.5.20 Detail showing one of the six custom-made ball-joint mounts used for M1 and M3.

Figure 2.5.21 Wide view of the M1 (left) and M3 (right) mounts, showing the "catcher" wire in place.
The safety “catcher” wires for M1 and M3 (there in case the glue should fail) needed to be both strong and, since the corners of the cut grooves are quite sharp, also tough or cut resistant. Twist-laid stainless steel wire was considered, but the need to operate a swaging tool so close to the mirrors was a disadvantage. In the end, a plaited mixture of stainless steel and carbon fibre was used – purchased from a fishing store as a trace material. With a 50 kg breaking strain and supposedly resistant to shark bite, the material has the great advantage that ordinary fishing knots can be used to attach it, avoiding the need for tools. Loops of this were tied around each mirror, and then tied (down in the photos; up in the final application) to the mounting posts.

Mirrors M1 and M3 are now complete, and ready to be installed into the optics box.
2.5.5 M2 Mount and Focussing System

The mounts for M2 would not only need to provide for fine adjustment to support the alignment of the optics (as the mounts for M1 and M3 do), but would also need to incorporate a focussing mechanism. The focussing mechanism needs to raise and lower M2 without introducing any tilt, thus preserving M2's angular alignment with respect to M1 and M3. Given the required accuracy of angular alignment (maximum 5 minutes of tilt), this represented a very stringent requirement on the focussing system design.

One fact in our favour, however, is that since M2 sits facing up, it is possible to mount it using a controlled thickness of soft glue (RTV or Room Temperature Vulcanising silicone was used). This provides the small degree of flexibility needed to accommodate any differential thermal expansion, so that it is possible to mount M2 to a single hard metal mounting plate, and engineer the focussing and adjustment system to the metal plate.

Initial advice I received was that a Crayford focuser would be appropriate. This is the focussing mechanism usually used on microscopes, consisting of two co-axial tubes, with a friction wheel or gear rack mechanism causing one tube to slide within the other.

The Crayford mechanism can maintain reasonable accuracy of alignment – sufficient for a tolerant transmissive design such as a microscope – when the tubes are long with respect to their diameter. The PCIS however needed to raise a large (170 mm long) mirror on a mechanism that had to fit into about 50 mm of space – i.e. calling for a squat mechanism, which would translate into a very short, large diameter Crayford tube, in which all the geometry militates against maintaining accuracy of horizontal alignment under vertical movement.

Given the high precision of horizontal alignment required of M2 in the Offner design – maximum 5 minutes of tilt – I could not in the end convince myself that a Crayford focussing mechanism would work sufficiently well, and sought an alternate solution. Failing to find any existing “off the shelf” design that would suit the focussing needs of the Offner design, I determined to design my own.

To obtain the required precision of adjustment I decided that, as for M1 and M2, I would fundamentally rely on fine threads to turn a manageably large angular movement into a very small, controlled linear movement. Again there would be three such threaded mounts, but this time they would need to be able to operate either independently (for alignment of M2), or in unison (for focussing). A gear train with large tooth count gears could keep the turning threads properly synchronised for focussing, but would need to be able to be disengaged to allow independent adjustment for alignment of M2.

In order that M2 remain level, even when changing the direction of focus adjustment, it is important that the gears driving each of the three threaded supports be the “same distance down the gear train”, so that any lash between the gears would as far as possible cancel out. The worst case would be to have one threaded support gear driving (another gear driving) another threaded support gear. In that case the effective change in level when changing focus direction would be twice that of the gear train between the threaded support gears.
Alternatively, if a single central driven gear were to drive the three threaded support gears simultaneously, then the effective change in tilt would be limited to that produced by the greatest difference of the lash between the three gear trains. Since each “gear train” would then be just two gears (i.e. one meshing) long, the net difference in lash should be small compared to the actual gear lash itself (and smaller compared to twice the gear lash).

The latter option, of a single central or “sun” gear driving three subsidiary or “planet” gears, was the one chosen for implementation. To preserve both symmetry and space at the front of the lens, two threaded support gears would be placed as far apart as possible at the back of M2, and the third at centre front of M2. (Space is required at the front of the lens box for the focus drive gear, the focus indicator shaft, the liquid light guide to the source turret, and for removal and replacement of the optical filter drawer. There is a little more room to spare at the back of the lens.)

Figure 2.5.24 shows the size and location selected for the sun and planetary gears, as well as the spur gear to drive the sun gear from the focussing knob, overlayed on the lens base plate.

Figure 2.5.24  Re-installing M1 and M3 into the lens box, as a complete assembly with the top plate.
For stability the threaded support posts should be as far apart as possible; for fine focus control (and minimisation of lash), the gears should be as large as possible. But the gears, which are above the base plate, must also remain clear of the light paths from sample to detector — which at the height of the base plate just fit within the apertures shown, and above the base plate are larger yet.

This led to the arrangement shown in Figure 2.5.24 above, with the sizes shown there, and with 20, 80 and 40 teeth each in the spur drive gear, the sun gear and the planetary gears respectively. This gives a reduction of 2:1 from the spur drive gear to each planetary gear, with cancellation of gear lash. The spur gear is driven from the focus knob shaft via a 1:1 right angle drive bevel gear set (bought commercially), meaning that one complete turn on the focussing knob will raise or lower M2 by 0.5 mm. This should yield a reliable and repeatable adjustment sensitivity of 1 or 2 hundredths of a millimetre in the height of M2 (represented by 7.5 to 15 degrees rotation of the focussing knob), over a focussing range of + / - 5 mm or more.

M2 would be invisible inside the lens box; there would be no indication from an unfocussed image as to which way the focus should be adjusted; and the focussing knob would clearly be multi-turn in normal use. It appeared therefore that there was a considerable and unacceptable risk that M2 might be driven too high, and wind all the way off one or more of the supporting posts — requiring disassembly of the lens and re-alignment of M2 (a non-trivial task — see Section 2.5.5 below). It was decided that an indicator of the present focussing position was a necessity rather than a luxury.

The focus position indicator is driven from the axle of the sun gear (which itself is moving at only $\frac{1}{4}$ of the rate of the focus shaft) via a worm or screw gear (commercially bought) giving a further reduction of 10:1, for a total ratio of 40:1 between the focus knob and the focus indicator. This is a low enough ratio to make it easy to discern when the focus knob is on the correct rotation (almost 10 degrees of indicator shaft movement per turn of the focus knob), but high enough to allow plus or minus twenty rotations of the focus knob (providing for + / - 10 mm movement on M2) before the indicator shaft cycles around. Plus or minus 5 mm on M2 is accommodated in a single “legal” semicircle of the indicator shaft.

Note that there is no mechanical interlock to prevent damage to M2. The user must ensure that the focus indicator stays in the “legal” range. In practice this has not been a problem, as focussing is limited to plus or minus a couple of turns on the focussing knob, so that the indicator shaft always remains close to its marked “central” position; and if focus is lost, it is usually recovered by moving back towards the “central” focus position.

To accommodate the system described, five holes were machined into the lens box base plate during manufacture, and were later (after anodising) fitted with brass bushes to support their respective gears. The sun gear axle is exactly central to the lens, a fact which would prove important during the final alignment of the lens. The bushes for the sun and planet gears can be seen in Figure 2.5.25 below, as can the spur drive gear itself. Note that the spur drive gear is tall. This is so that when the central sun gear is raised to disengage it from the three planet gears (as must be done to adjust the planet gears separately during alignment of M2), it remains in engagement with the spur gear. In this way the spur gear can be jigged (by plus or minus half a tooth) to help the sun gear re-engage with the planet gears when the alignment is complete.
Development and Application of a Photon Counting Imaging System

Figure 2.5.25 Lens box base plate showing the bushes for the focusing gears.

In Figure 2.5.25 the focussing knob, bevel gears, spur drive gear and focus indicator shaft are already installed. The central bush will take the sun gear on its steel axle. The other three bushes will take the internally threaded geared support mechanisms for M2. The detail of these is shown below.

Figure 2.5.26 Threaded adjuster/support for M2, shown with brass bush in lens base plate.
Note the hex shape cut into the very bottom of the threaded M2 adjuster gear. This allows each planet gear (i.e. M2 support) to be adjusted separately (and easily) from outside the lens box during the alignment of M2, using a 12 mm ring or open-ended spanner. The circlip fixes the lens support firmly in place in the base plate bush, avoiding any unintended movement at this point. (The bushes themselves are firmly interference-fitted to the base plate.)

The recess in the top of the gear is to allow the nylon fitting at the top of the threaded shaft to come as low as possible, so as not to limit the downward focusing movement of M2. The threaded shaft, and the nylon fitting via which it is threaded into the M2 base plate, are shown in Figure 2.5.27. The purpose of the nylon bushes is to allow a slight movement of the threaded posts with respect to the M2 base plate as M2 is aligned – otherwise alignment of M2 might be impossible, or might cause the focusing mechanism to bind up mechanically and become difficult to operate. Small stresses can be sustained at this point, as they are transferred to the rigid base plate supporting M2 (see below), and not to the glued attachments to the mirror itself. This is why nylon inserts are adequate here, whereas the M1 and M3 mounts required full ball joints. (Due to the design, M1 and M3 also required rotational freedom at the mounts, which the ball joints also provide.)

![Figure 2.5.27 Threaded shafts and nylon mounting plate inserts for M2 supports.](image-url)
At this point in time I made an error involving glue. When inserting the threaded posts into the nylon mounting plate inserts, I knew that they must not move at that point in future — the movement must occur between the threaded shaft and the internally threaded gear if the focussing was to work correctly. So rather than thinking it through properly I added a drop of hard setting thread locker (essentially superglue) to the shaft before threading it into the bush.

And it stuck fast part way in. Actually (and fortunately), most of the way in, and my judgement was that it could be left as was without critically limiting the downward range of M2. It would reduce the downward movement available from the calculated nominal-focus M2 position from 8 mm to about 5 mm, which should be (and fortunately has proven to be) adequate.

Nevertheless it was a foolish mistake; the parts would have been difficult to disassemble, since the nylon inserts were also glued into the mounting plate, and M2 was already attached. The really galling thing is that the error could have been so easily avoided in either of two ways:

1. A trial fit-up of the parts before final assembly would have revealed that the threaded steel shafts were a very snug fit in the nylon threads, and that no glue would be needed; and
2. A moment’s calm reflection would have revealed the truth — that if a steel thread into nylon needed thread locking glue, then we wouldn’t have Nylock™ nuts!

The first major lesson about glues then, is to think first, and to be quite sure that any glue is needed at all! Unfortunately, as events will prove, it was a lesson that I was slow to learn.

Figure 2.5.28 M2 mounting plate, plan view. Note cut-outs to accommodate spur gear and raised sun gear.

---

7 If this statement creates a slight eerie feeling that you are going to encounter it again, and again — a faint and unidentified future echo — then that feeling is called *uja dève*, and you would be right. Glues are a recurring theme. *Uja dève*, incidentally, is a recent expression in English (I know this, as I created it myself) so you may not have heard it before. However, I have a slightly eerie feeling that you’re going to hear it again...
The plate to which M2 was glued, and to which the adjustable support posts are attached via the nylon inserts, is shown above (cross section is below). Note the two circular-section recesses in the plate. These allow room for the spur drive gear, and for the sun gear when it is raised during M2 alignment, so that the downward movement range of M2 is not limited. This, in conjunction with the recesses in the planet gears, was designed to preserve a full (and fortunately generous) +/- 8 mm range of vertical movement for M2.

![Diagram of M2 mounting plate](image)

Figure 2.5.29 M2 mounting plate. The nylon inserts were threaded and glued into the recesses shown.

Also to be seen in the sides of the M2 mounting plate are a set of threaded holes. These are provided to mount two optical baffles to the sides of M2 (described below). During gluing of M2 to the base plate they also serve for attaching a guide rail, against which M2 can be pushed to ensure it is central on the mounting plate. (The mounting plate is slightly wider than M2 both to provide the best location for the baffles, and to allow room for thick walled nylon inserts on the support posts.)

In this fashion, the M2 mounting plate (which is the same length as M2) becomes its own jig for positioning M2 during gluing. The plate is firmly mounted, with the side guide rail attached (overhanging the top of the plate by 3 mm), and with its end pressed against a hard straight edge. Four blobs of RTV silicone are placed on the plate, and two narrow, 1 mm thick shims are laid down clear of the silicone. M2 is then placed onto the silicone blobs, pressed against the guide rail and the hard edge at the end of the plate. The silicone is allowed to partially cure before the shims are removed. When the silicone is fully cured the nylon inserts and the support posts are screwed into the bottom of the plate, the baffles are attached to the sides, and M2 is ready for installation.
2.5.5.1 Internal Baffle Plates

The internal baffle plates sit either side of M2 to eliminate any photons straying from the correct optic path. As can be seen in Figure 2.5.30, there is a wedge-shaped gap between (a) the light path from the sample to M1 and (b) the light path from M1 to M2. The figure shows these light paths for both the front (or rear) edges of the mirrors (the lower marked intersection), and also for the centre edges of the mirrors (the higher marked intersection). The "wedge" between the two light paths thus forms a fan shape, as shown in the final dimensions drawing of the baffle (Figure 2.5.31).

The intersection of the light paths (a) from M2 to M3 and (b) from M3 to the detector, form an almost identical shape on the right-hand side of M2. Since the detector is slightly larger than the sample, the right (or detector) side light paths are slightly larger, and the corresponding fan shape between them slightly smaller, than on the left (or sample / M1) side. The calculations for the fan shape into which the baffle must fit are therefore carried out on the right hand or detector side of the lens.

The plane defined by the edge of this fan shape meets the plane of the base of M2 about 3 mm outside the edge of M2 (see Figure 2.5.30). This is why the mounting plate for M2 is made 6 mm wider than M2 itself. This lets the baffles to be attached to the sides of the M2 mounting plate, with a single bend of 11 degrees putting the whole baffle into the right location to block as many stray photons as possible, but without any occlusion of the desired light path from sample to detector.

Figure 2.5.30 Calculations for the dimensions and placement of the M2-attached optical baffles.
The final dimensions of the baffle were based completely around straight lines and circular arc segments, and are shown in Figure 2.5.31 below. The baffles were not anodised, but were spray painted matt black before being attached to the M2 mounting plate.

Figure 2.5.31 Dimensions of the M2-attached optical baffles.
2.5.6 Alignment – First Attempt

The first attempt at alignment of the optics was what I can generously describe as a qualified success, and honestly describe as a deep learning experience.

I had always recognised that construction of the optics was one thing, but aligning them to work at their best was another, and one for which I was woefully ill-equipped. (My total experience in optical assembly had been a period spent repairing cameras and projectors. But taking a cylindrical lens apart, and reassembling it with all the shims and spacers back where they came from – whilst highly instructive in the techniques of hidden assembly screws and the uses of helical threads – is a long way short of designing an alignment procedure ab initio. Besides, as I stated earlier, transmissive lenses are easy, and very forgiving. Reflective optic systems are not.)

I had been assured by the relevant optical expert at the University (no names, no pack drill) that he understood my situation, and would provide the assistance required when the time came. However, when the time duly came, the offered assistance turned out to be the gift of an optically flat front surface mirror, and the advice that a sphere has a centre, to which a light ray will return if that’s where it started. I appreciated the gift of the mirror.

Accordingly I designed and constructed a dual purpose jig. In mode 1 it would support the lens box without the base plate, and allow a plate to be attached nominally at the point where the centres of M1 and M3 should be. In mode 2 it supported the lens box with the base plate in (and therefore also with M2 installed), and allowed a screen to be placed at each of the nominal locations of the sample and detector. The problem with this approach turned out to be the number of links (through the body of the lens box and the several components of the jig) between the mirror mounting points, and the supposed location on the jig of the various target points.

The procedure, carried out at night to get adequate low light conditions, was as follows:

1. Mount the optics box, with M1 and M3 fitted but with the floor and front of the box absent, on the alignment jig in Mode 1.
2. Fit the target plate which has two holes, 0.5 mm diameter and 1.5 mm apart, as close as possible to the geometric centre (in plan view) of the lens box.
3. Place an incoherent light (a green LED) under each of the two holes in turn, and adjust the corresponding mirror until the reflection appears to return to its origin.
4. Place a laser pointer under each hole in turn, pointing it variously at the corners, the centre sides and centre of the corresponding mirror, and adjust the mirror until the best possible return of the beam to its origin is achieved for all parts of the mirror.
5. Repeat steps 3 and 4 for each of M1 and M3, ad nauseam if not actually ad infinitum, until happy or tired.
6. Raise the lens box, and install the base plate. Return the lens box to the alignment jig, this time in mode 2.
7. Install M2 and all its associated focussing gear into the base of the lens box.
8. Support the bottom of the shaft of the central focussing gear, disengaging it from the three planetary gears that complete the focussing system, and allowing the three mounting points for M2 to be adjusted separately.
9. Mount a plate with a square array of 0.5 mm holes onto the alignment jig, as close as possible to the calculated position of the sample.

10. Mount a plate with an interchangeable (black painted or white rice paper) screen onto the jig at the calculated detector position. (In fact, this calculated position – as with that of the sample – has been adjusted for the fact that there is no vacuum window or filter pack in the alignment assembly. The two plates are therefore of different thicknesses.)

11. Place a diffused white light source under the source perforated plate to produce a corresponding array image at the detector plate, and separately adjust the M2 mounts until a sharp image of the perforated plate appears at the desired location on the detector screen. This will turn out to be a compromise between sharpness and location.

12. Place an optically flat front-coated mirror at the detector location (using the right thickness of support plate) and check that a laser beam from any one of the sample grid holes, pointed at any part of M1, returns after 7 reflections to the hole from whence it came.

13. Repeat steps 11 and 12, but this time also allowing adjustments of M1 and M3 mounts, in order to improve the compromise between best focus and best location – i.e. attempting as far as possible to optimise both. Repeat until happy or tired.

Step 13 unfortunately involves 9 separate points of adjustment – that is, 9 independent variables – and turns out not to be possible.

The PCIS run following this alignment produced quite good sharp images, but with the image off-centred sufficiently badly that some 20% of the sample disk was off the detector area, and was not imaged. (See Figure 2.5.32.) In addition, the focusing gear was extremely tight, and in fact had to be turned with pliers rather than using a light finger touch. This indicated that the base plate of M2 was not flat with respect to the base of the box, exceeding the capability of the nylon spacers incorporated into the M2 mounts to absorb the consequent tilt of the mounting posts. This further suggested that the alignment of M1 and M3 had been at fault, with the alignment jig not placing the alignment holes at the precise centre of the lens box.

Figure 2.5.32 Off-centred image produced after the first alignment of the PCIS optics.
After this PCIS run had been completed, I felt that my experience with the lens alignment (two long nights of adjuster twiddling) had given me sufficient “feel” for the behaviour of the optics, that I should be able to jiggle the image back towards the centre of the detector “by hand and eye”. This would involve a synchronised adjustment of all 9 mounting points.

I was able to move the image back towards the centre of the detector, getting the whole of the sample disk into the field of view, but at the significant expense of focussing sharpness across the sample area (see Figure 2.5.33).

![Figure 2.5.33 Correctly aligned but out-of-focus image.](image)

I appeared now to be in a situation where I could traverse a locus of alignments, trading accuracy of the image alignment with consistency of focus across the image area.

Obviously a different approach to the alignment of the optics was required – one that would provide concurrently both adequate centring and correct focussing of the image.
2.5.7 Revised Alignment Procedure and Jigs

I sought the assistance of my friend and optical designer Damien Jones. Sitting on his verandah contemplating the Sunshine Coast hinterland bush and eating scones (the perfection of which was then his latest mania, and a very acceptable one at that), Damien produced the nub of the solution, literally, on the back of an envelope. At the heart of the idea was to establish a reference laser beam that could be aligned to the actual base of the lens box, before removing the base and using the laser reference to position the jig more accurately (in Mode 1) to the centre of the lens assembly.

The enhanced accuracy of the alignment of the jig would carry right through the alignment procedure, reducing the (apparently impossible) 9 point adjustment to three separate, and achievable, 3 point adjustments. M2 would be aligned so that it reflected the laser directly back up to its source, making its alignment independent of M1 and M3. The use of appropriate temporary baffles in the lens box would allow M1 and M3 to be adjusted separately (though in the end removal of the baffles allowed a series of reflections of the two centres in both M1 and M3, producing a very distinctive pattern of reflections that made any inaccuracy of alignment visually very obvious).

The alignment jig that I designed for this, and which was manufactured by the RSES mechanical workshop, comprised two large plates with fine-thread lockable (and spring loaded) screw adjusters to set their separation at three points – i.e. to adjust how parallel they were to each other. The lower plate is firmly attached to the top plate of the lens box using the four threaded holes put in for that purpose; and a laser pointer (with an external battery pack to last for a week) mounted to the upper plate via a special rigid assembly that I designed. This assembly allows some lateral movement of the pointer before it is clamped down, so that it can be aligned with the relevant holes.
A brass plug with a hole through its centre (0.5 mm diameter, opening out to 0.6 mm at the bottom end) is inserted into the centre hole of the top plate. Note that this hole was drilled for a support assembly, and was not milled in during the initial manufacture of the plate. By measurement it is approximately 0.1 mm off the true centre of the plate, and so the plug is also manufactured with a 0.1 mm offset of its "central" hole. The direction of offset is indicated by a stamped mark on the plug, and must be installed with the mark facing to the right hand side wall of the lens box. The laser pointer mount is adjusted so that the beam passes through the hole in the top plate plug.

With the base plate of the lens box installed, but with M2 and the focussing gear not installed, the lens is placed on the alignment jig in Mode 2. The second brass plug (with its 0.5 / 0.6 mm diameter hole in its actual centre) is inserted into the hole which normally carries the shaft and bush for the centre focussing gear. This hole is at the precise centre of the base plate.

![Image of the alignment setup](image)

Figure 2.5.35 The revised alignment setup for the PCIS optics.

The top alignment tool is now adjusted (via its 3 fine-thread screws) so that the laser beam also passes directly through the centre of the lower brass plug. A magnifying assembly was used to allow this to be viewed through a camera, with its viewfinder diverted to a 14" computer screen (see photo, Figure 2.5.35). This allowed any touching of the edges of the lower (0.6 mm) hole by the laser beam to be observed easily, with an appearance very reminiscent of the "diamond ring" effect just prior to totality of a solar eclipse. The top plate adjusters are then clamped securely into place, and must remain that way for the remainder of the alignment process. The reference beam against which the alignment will be performed is now established.

This reference beam is now used to assist the independent and separate alignments of M1, M3 and M2, eliminating the "9-point adjustment" problems of the first alignment attempt. The procedure used is described below.
2.5.8 Broadband Optics - Final Assembly and Alignment

This section describes the procedure for the assembly and alignment of the broadband optics as it should have happened; with hindsight. All of the steps and operations actually occurred at some time, but not in the ideal order described here.

The reason for this lies in the initial and unsuccessful attempt at aligning the lens, following which a partial disassembly was sufficient to allow cleaning of the components, and application of the final alignment procedure. The complete assembly and the final alignment procedure were therefore never actually carried out together. That is nevertheless the simplest and best way to describe them.

First though, a confession – I made a mistake involving glue.

During the first assembly of the broadband optics, I had for some reason decided that the light seal around the edges of the lens box might need improving, and for even less reason that a layer of black RTV silicone at each join was a sensible way to achieve it. Of course it contributed little to the light seal, but placed all of the carefully tolerated dimensions of the mechanical components – and therefore of the optics themselves – in jeopardy. Something I might have thought of beforehand...

Removing every last tiny scrap of the silicone from all 24 stepped edges of the alloy plates required the use of sulphuric acid-based silicone removal paste, and was extremely tedious, boring and unpleasant. A labour of penance for stupidities committed, and no labour of love.

Later tests of the complete PCIS suggest that light leakage at the edges of the lens box is not and never has been any issue.

2.5.8.1 Preparations for Assembly of the Optics

It is assumed that all of the components of the lens are available, properly cleaned and ready for assembly, and also that the two main optical alignment jigs and ancillary items required for the alignment are to hand.

The assembly and alignment process may take several days, and requires a clean, undisturbed room with at least 2 metres of available, clear bench space. There needs to be some point above the bench to attach a pulley system directly over the main lens support jig, making it possible to raise and lower the lens for fitment and removal of the base plate during the alignment.

It is further assumed that the mirrors are already attached to their mounting hardware. M1 and M3 should have their ball joint units assembled, the steel threaded shafts protruding from the backs of the mirrors, and the ball joints lapped and operating smoothly. M2 should be on its mounting plate, with the three nylon inserts and threaded steel shafts already installed to the underside. If these are not already assembled, refer to the sections above on the mirror mount assemblies for details of the procedures to be followed, and assemble the mounts before continuing.
2.5.8.2 Initial Assembly Stages

The first stage is to assemble the main plates that comprise the lens box. The front plate only requires no preparation. The other plates should be prepared as follows.

1. Attach the carrying handles to the centre outside of each side plate, and the lens lifting eye bolts to the outside upper corners.
2. Insert eight bolts (to provide optical blanking of the threaded holes) into the outside of the rear lens plate. (Alternatively, if the new lens lifting mechanism has been built, attach the lens lifting bracket to the same eight holes.)
3. Attach the source turret (rectangular) and shutter housing (circular) engaging rings to the underside of the lens base plate. Use dowels for accurate location, and bolt to the threaded holes using recessed cap-head bolts.
4. Attach the two alloy support brackets and the focus drive shaft, with the focussing knob and right-angle-drive gear attached, to the underside of the lens base plate.
5. Install the focus drive spur gear through the appropriate brass bushing from the upperside of the lens base plate. Secure it in place with the right-angle-drive gear grub screw, making sure the two bevel gears engage correctly. Refit the gears with washers as shims if necessary for correct meshing of the two gears.
6. Attach the two alloy brackets and the focus indicator shaft, with the focus pointer and the worm-driven gear attached, to the underside of the lens base plate. A trial installation of the central sun gear now is a good idea to ensure the correct alignment of the worm drive gears for the focus indicator shaft. The sun gear should now be removed again.

The top, sides and back plate now need to be assembled and bolted rigidly together. Start with three plates that meet at a corner (top, back and one side), and get two bolts into each of the three edges. Nip the bolts, but don’t give them a final tighten. Attach the other side, again with a couple of nipped-tight bolts in each edge.

At this stage I prefer to also attach the base and front, using only the outer two bolts on each edge. These will be removed again, but ensure the correct alignment of all the box plates before all 70-odd bolts are installed.

Install all of the rest of the bolts connecting the back, top and side plates, and tighten them all properly. The front plate and then the bottom plate should now be removed again, and the lens box placed on the alignment stand in mode 1 (see “Alignment – First Attempt” above).

The four lifting chains and turnbuckle adjusters should be attached to the four eye-bolts in the side plates, and to their central joining ring, so that the overhead pulley arrangement may now be used to raise and lower the lens via the lifting ring.
2.5.8.3 Installing M1 and M3

It is assumed that M1 is to be fitted first, into the left hand side, or sample side of the lens box, followed by M3 fitted into the detector side. It doesn’t in fact matter which mirror goes in which side, or the order in which they are fitted.

The procedure for fitting M1 into the lens box requires 2 people, and is as follows.

1. Align the threaded shafts of the M1 mounts so that they are perpendicular to the back of the mirror. (This will be their final orientation when installed.)
2. Insert the twelve dowels for the six brass lens mounts into the upper surface of the lens box top plate.
3. Calculate how far down the threaded shafts the bottom of the threaded brass inserts will come, with the mirror in its calculated correct position. (This will be one figure for the central or inner mount, and another – larger – figure for the two outer mounts.) Mark the threaded shafts with a piece of insulation tape on the ball joint side of the calculated position.
4. Wearing dust-free rubber gloves and handling the mirror with lint-free lens tissues, raise M1 carefully into place so that the threaded shafts protrude through the three large holes in the lens box top plate. Have the second person thread the brass mounting points onto the three threaded shafts until they just reach the insulation tape, and then align each brass mount so that it will mate with its dowels. Have the second person remove the insulation tape markers with tweezers.
5. Lower M1 carefully into place, with the second person assisting in seating the brass mounts, until all three brass mounts are fully seated on their locating dowels. Install and tighten the bolts that attach the lens mounts to the lens box.

The procedure for fitting M3 is identical to that for M1 except that, unlike M1, M3 cannot now be installed by moving it perpendicularly to its back face into its final position (M1 is in the way); instead M3 has to be lifted vertically into place. For this reason the threaded shafts need to be positioned somewhat differently so that they will reach the holes in the top plate, and will need to be carefully adjusted back to perpendicular by the second person as the mirror is carefully lifted to the top of the box and then (after the brass mounts are threaded on) lowered to its final position.

The brass lock nuts can now be threaded onto the six adjuster shafts where they protrude from the brass mounts, but should not be tightened at this stage.

The reason for measuring and marking (with insulation tape) the initial positions of the brass mounts on the adjuster shafts is to ensure that the mirrors are mounted at close to their final correct angle. This avoids introducing stresses into the glued mount attachments to the mirrors, which would arise if the mounts deviated significantly from perpendicular to the back of the mirror. During alignment care will be taken to avoid the same stresses by making sure the mirror is kept “pointed towards” the same central focus, even when its radial position is being adjusted. An error between the relative positions of the three adjusters of 1 mm (corresponding in the worst case to an angle of about half a degree) can be easily tolerated without inducing excessive stresses at the glued joins.
2.5.8.4 Establishing the Centre Reference

The next stage of the optics assembly is to establish a laser line down the exact centre of the lens box that will be used as a common reference in the separate alignment of each of the three mirrors. The jig used for this was described earlier.

Install the 0.5 mm drilled brass plug into the central hole in the lens box top plate, with the stamped offset mark pointing to the right (or detector side) of the lens. The jig must now be attached to the top of the lens box using the 4 machined stand-offs and the corresponding bolts. (The threaded holes do not penetrate the lens box top plate, and so the correct length bolts are essential.)

Note: although this jig uses a low power laser pointer device, the beam is still capable of damaging the eyes, and it will be on a lot of the time during the alignment. Please take care, especially when using mirrors around the laser beam.

Install the laser pointer into the carrier on the jig, and attach its external battery pack (the pointer should come on). Loosen the bolts attaching the laser holder to the jig and adjust the position of the laser carrier until the beam is cleanly penetrating the 0.5 mm hole in the brass plug.

![Image of laser alignment jig](image-url)

Figure 2.5.36 The laser alignment jig in place on the PCIS optics.

The lens must now be raised temporarily so that the lens base plate can be installed, using at least two bolts on each of the three edges. The lens box is now lowered back onto the lens support jig (in mode 2). The lower brass plug (without a registration mark) should now be placed into the centre of the base plate, using the brass bush provided for the focussing sun gear.
It is now necessary to establish a good magnified image of the lower plug so that the laser can be adjusted accurately. After playing with a number of lens options, I settled on a 7 megapixel digital camera, in macro mode, pointing at the lower plug through an inverted golf range-finding scope. The camera was attached to a laptop computer using remote camera software so that the viewfinder image appeared on the computer screen, and the camera controlled (and images captured) from the computer. This gave a large, easily viewed image of the crucial point while all the alignments were conducted. The arrangement is shown below in Figure 2.5.37.

![Image of the monitoring setup for the PCIS optics alignment.](image)

Figure 2.5.37 The monitoring setup for the PCIS optics alignment.

The laser alignment jig is now adjusted (by loosening the three lock screws and using the three fine thread height adjuster screws) until the laser beam is precisely penetrating the centre of the 0.6 mm hole in the bottom plug. Images showing incorrect and correct alignment of the laser beam appear in the following two figures. Even a slight misalignment is immediately visible as the laser touches one edge of the hole, producing a distinctive eclipse-like "diamond ring" effect, as was described earlier. This makes precise alignment of the laser beam a relatively simple affair. The laser alignment jig must now be carefully clamped in place via the locking screws, without disturbing the alignment.
In the following photo, a green LED has been placed below the alignment plug in the base plate, to make the position of the laser beam relative to the hole easier to see.

Figure 2.5.38 The laser alignment beam missing the alignment hole by 1 mm or so.

Figure 2.5.39 Alignment almost correct – laser beam just touching edge of alignment hole.
2.5.8.5 Alignment of M1 and M3

Now that the laser reference beam is centred, it can be used to correctly position the M1/M3 alignment plate on the lens jig. M1 and M3 can then be aligned.

First, the lens must be raised, the base plate removed, and the lens replaced on the jig in mode 1.

Now, the M1/M3 alignment plate, which has 2 holes, each 0.5 mm diameter and with centres 1.56 mm apart (see Figure 2.5.41 below), should be attached over the centre aperture of the lens jig. The screws should not be fully tightened yet.

With the red laser reference beam switched on, and the green LED under the alignment plate, position the plate so that the red dot is precisely central between the green dots (see Figure 2.5.42), and tighten the screws. The two holes in the alignment plate are now at the correct position for the centres of the spheres that are M1 and M3. Tighten the screws attaching the M1/M3 alignment plate to the lens jig. The laser may be switched off.

M1 and M3 are now aligned in turn, using the green (non-coherent) LED and a red laser pointer under the M1/M3 alignment plate, and observing the reflections. Starting with the alignment of M1, the M1/M3 baffle is installed under M3 to eliminate irrelevant and distracting reflections.

The M1/M3 baffle is a piece of thin alloy about 360 mm square, bent in a right angle in the middle and sprayed matt black. It can be attached under either M1 or M3 by a stationery clip to the side of the lens box while the other mirror is aligned.
For adjusting the M1 and M3 mounts, a good quality screwdriver was selected that fitted the heads of the adjusters snugly, and a 315 mm bar was welded perpendicular to the shaft, close to the handle. With this, 1 cm of movement at the end of the handle would correspond to 10 microns of movement at the mirror mount. This made the adjustments very easy to control.
The first step is to bring the green reflected spot to align with the M1 centre hole. (The M3 centre hole can be obscured during this initial phase, though it turns out that for most of the alignment procedure it is better for the second hole to be left open.) This brings M1 to the approximate correct orientation; it remains to bring it to the correct distance from the centre.

For this the (second) red laser pointer is used, being aimed through the centre hole at different parts of M1 in turn (centre, edges and corners) and the position of the reflection observed. From the pattern of the observed reflections it can be determined whether the laser beam paths are crossing over themselves on their return from the mirror, and so whether M1 is too close to, or too far from, its centre. As the axial location of M1 is improved, small errors in its orientation will be observed, and can be corrected. Eventually, the beam will be returning from all parts of M1 correctly back to its origin hole, and M1 is correctly positioned. It can now be re-checked using the green light, and observing the reflections of the light from both the M1 and the M3 centre holes (see Figure 2.5.43).

![Figure 2.5.43 Correct alignment of M1, with reflection of M2 centre hole also visible.](image)

From the experience of performing the alignment, it is clear that the beam can be returned to better than 0.2 mm accuracy, representing an angular alignment to within 2 minutes of arc of the correct angle. With axial displacement, a movement in either direction of 2 cm on the screwdriver handle on all three adjusters was sufficient to render a noticeable deterioration of edge alignment versus centre alignment. I therefore conclude that axial placement was achieved to within +/ - 10 microns of the nominal placement. These placements are noticeably more accurate than the tolerances required by the design specification.

The M1 / M3 baffle is now moved to the left side of the lens box, and the alignment procedure is repeated for M3.
Interestingly, at the conclusion of aligning both M1 and M3, the baffle can be removed completely from the lens box and the green LED placed under the two lens-centre holes. What will now be observed is an evenly-spaced, dead straight row of green spots, being the two holes with the green LED shining through; the reflection of each of those holes on the other mirror (the one it is not the centre of); then the reflection of those two reflections in each of M1 and M3, producing two spots co-incident with the original two holes, and two further spots extending the dotted line; and etc. A surprising number of visible dots appear (3 or 4 each side), when considering the tiny percentage of light energy that is scattered from each dot (on matt black paint) back towards the mirrors.

![Image](image_url)

Figure 2.5.44 Adjusting M3 using the enlarged image on the laptop screen. The first image shows the set-up with the room lights on; the second shows the actual lighting conditions.

By experiment, it was found that the tiniest adjustment of any one of the six mounts produced an irregularity in the row of dots that was immediately obvious to the eye. (It seems we are particularly good at this kind of pattern matching, and a small error in a pattern is much more noticeable than a similar displacement of a single object.) This regular green dotted line is therefore a very good cross-check that the alignment of both M1 and M3 has been performed accurately.
2.5.8.6 Assembly and Installation of M2

It is assumed that M2 is already attached to its mounting plate, and that the nylon inserts and threaded support posts have already been installed into the underside of the M2 mounting plate. These steps are described in “M2 Mount and Focussing Assembly” above.

If the lens internal light baffles (see “M2 Mount and Focussing Assembly”) are not already attached to the sides of the M2 mounting plate, attach them now, with the baffles fanning outwards, away from M2, using the correct 10 mm long cap-head bolts.

The final stage of preparation of M2 is to attach the internally threaded brass “planet” gears of the focussing mechanism to the M2 assembly. These are threaded onto the steel M2 support posts so that they are all equidistant from the underside of the mounting plate. The correct distance corresponding to the calculated “correct” height of M2 for a nominal 6 mm thickness of filter glass can easily be calculated and used, but this is not critical since M2 can easily be raised and lowered, via the focussing mechanism, after it has been aligned.

The lens box should now be raised and the lens base plate re-installed. The good news is, this is the last time that the base plate needs to be removed and re-installed; the bad news is that, as a consequence, this time all 20-odd bolts need to be installed and tightened. The lens box is lowered back onto the lens jig in mode 2.

Next, the central (and larger) “sun” gear should be installed into the central hole of the lens base plate. (It will be found that the laser-alignment plug needs to be removed first.) Do not fit the worm gear (for the focus indicator) to the sun gear at this stage.

Carefully install the complete M2 assembly into the lens box base plate, so that the three internally threaded extensions of the planet gears drop into the three corresponding brass bushes. It may be necessary to “jiggle” the focussing knob slightly to get the sun gear to engage correctly with all three planet gears.

Install the circlips onto the planet gears where they protrude from the bottom of the brass bushes. These hold the M2 assembly firmly down in place, while still allowing the planet gears to turn for alignment of M2 and focussing.

Figure 2.5.45 The focussing knob, indicator and adjuster assemblies seen from under the lens box.
2.5.8.7 Alignment of M2

The central sun gear needs to be disengaged again from the three planet gears. This is done by lifting it from below (where its shaft protrudes from the central brass bush in the lens base plate). The shaft will need to be raised by just over 10 mm to disengage the gears, and it should then be supported in place by placing an appropriate object underneath it.

Note that the sun gear remains engaged with the focus drive spur gear (because of its taller design), and therefore also with the focussing knob. The three planetary gears, however, are now free to rotate independently of each other, or of any other part of the focussing assembly. They can be turned using a 12 mm spanner on the exposed lowest part of each planet gear.

The laser alignment beam should now be turned on again, and its reflection from M2 observed. This reflection is now in the "ceiling" of the lens box (rather than the "floor" as it was earlier), nestled between the inner edges of M1 and M3. The sighting scope was set up to enable this point to be observed properly, but due to access difficulties, the camera and PC arrangement was not used.

Using the 12 mm spanner on the three planet gears, the angle of M2 is adjusted until the laser reference beam (nominally 0.5 mm diameter) correctly enters the underside of the upper laser alignment plug (0.6 mm diameter). This is fortunately very easy to achieve with great accuracy, making up for the difficulties of observation and the uncomfortable posture required.

Calculations based on the small spanner-end movement (2 to 3 cm only) necessary to visibly de-align the mirror, the length of the spanner (~ 20 cm), the pitch of the threads (1 mm) and the separation of the gear centres (46.8 mm at the back), suggest that the resultant alignment is accurate to within 1 or 2 minutes of arc – again, well within the specified tolerances.

Remove the object supporting the sun gear shaft, and pull the sun gear shaft downwards so that the sun gear re-engages with the three planet gears. It may be necessary to jiggle the focussing knob (and therefore the sun gear) very slightly to achieve the engagement. However, care should be taken to limit the movement to + / - half a gear tooth or less; otherwise the correct angular alignment of M2 will be lost.

The worm gear that drives the focus indicator shaft should now be installed on the bottom of the sun gear shaft (lining up the focus indicator to somewhere near vertical), and the grub screw tightened to engage the focus indicator, and to hold the sun gear firmly in place.

The focussing knob should now be operated through a couple of turns in either direction to ensure that the focussing gears and focus indicator are all operating smoothly and correctly. The alignment of the laser beam from M2 back to the upper laser alignment plug should now be re-checked. If all is correct, then all three mirrors are now correctly aligned.
2.5.8.8 Completion of Optical Alignment

The various lasers and LEDs used for alignment may now be turned off, and the laser alignment jig and the upper laser alignment plug removed from the lens box upper plate. The four attachment holes for the jig are blind, and can be left as they are. The central hole from which the plug was removed however must be blinded off, and an eye bolt is used for this. (This was the original lifting point for the lens, and still provides an additional lifting point if required). Take great care accessing the nut for the eye bolt between M1 and M3 in the ceiling of the lens box.

The front plate of the lens box should now be installed, and all of its bolts installed and tightened.

The lens should now be lifted with the pulley arrangement one last time, the lens support jig and associated pieces removed from under it, the lens base attached, and the lens lowered back to the bench. The beam and pulley arrangement should now be removed and hidden before someone from OH&S sees it and demands a load rating and engineer's report.

The optical alignment is now complete, and the optics are (hopefully) ready for use.

Figure 2.5.46 The completed and aligned PCIS broadband optics.

---

8 This actually happened.
2.6 Source Turret

The source turret provides the mating between the Minisys and the Broadband Optics. In addition, it provides for:

- Vacuum seal between the sample chamber and the optics;
- Optical stimulation sources, comprising in the final implementation infrared, red, blue, and ultraviolet LEDs, plus a general purpose stimulation port that can accept the liquid light guide used in the original Minisys reader;
- A photo-transistor to detector excessive light levels in the sample chamber area, or to detect the failure of an external light source; and
- A changeable optical filter pack to separate optical stimulation photons from emitted luminescence photons.

An early design decision (made by myself perhaps to the concern, if not actually against the advice of my supervisor) was that the optical filters should be able to be changed without the need to remove the optics or disassemble the PCIS in any way. Although this would be a difficult design, and in the end would make the entire mechanical design and assembly of the source turret extremely complex, I felt that it was necessary if the PCIS was to succeed as an instrument to explore the full range of luminescence signals arising from a particular sample. I believed this to be a critical capability to take full advantage of the broad spectral range of the LN-CCD, and to justify the expense and difficulty of the broad spectrum reflective design of the optics.

Accordingly a “Filter Drawer” was designed with extreme baffling to eliminate extraneous light either from outside (room lighting) or from around the filters (stimulation photons). This baffling would need to achieve 15 to 18 orders of magnitude light rejection, and was to dominate the physical design and assembly of the source turret. But it worked!

2.6.1 Design

The defining constraint for the source turret design was to avoid any obstruction of the signal light path while providing the other features necessary, to wit:

- the vacuum window and retaining ring;
- the stimulation LEDs, which must be as close as possible to the sample;
- the filter pack, ideally accommodating 10 mm or more of filter thickness;
- the filter drawer and its light baffling components;
- the wiring for the stimulation LEDs and light detector; and
- a light proof mating assembly for the optics box.

Since the light path from the sample to M1 is essentially an inverted rectangular pyramid, it is clear that the higher up the filter pack is placed, the larger it needs to be in order to contain the entire light path, and so avoid any obstruction of signal photons and consequent loss of PCIS sensitivity.
Now good optical filter materials are quite expensive, and it was immediately clear that to use our existing, extensive set of 44 mm diameter filters would cause a significant obstruction of the light path, and the loss of somewhere in the region of 50% of the available photons. (Refer to the later discussion on vignetting for the approximate nature of this number). Larger rectangular filters would be needed, but the lower the filter pack could be placed, the smaller the rectangular filters could be.

The design would also need to include the ability to use 44 mm diameter circular filters (since replacement of the full set of filters with rectangular versions would be well beyond the budget available), and an appropriate adapter plate would need to be designed. This would allow the full range of spectral experiments to be conducted, albeit at a reduced sensitivity, and full size rectangular filters could be obtained for those applications that would particularly benefit from the full unconstrained PCIS sensitivity. In particular, full size UG11 filters with a red-excluding notch coating on one surface would clearly be wanted for quartz blue/UV OSL, and were ordered as soon as the required physical size was determined.

The ring of stimulation LEDs also needed to be as close as possible to the sample, for two reasons:

1. Given the best blue LEDs available at the time (Nichia BP500S luminance-selected 20 mA single chip GaAs LEDs in a clear plastic T5 5 mm package), the intensity of optical stimulation for OSL might be borderline. I was trying to replace the liquid light guide source, which is powered by a 300 watt QH globe, with about 2/3 of a watt – electrical – of LEDs. Since the LEDs are essentially a non-collimated source, stimulation intensity at the sample would be inversely proportional to the square of the sample-LED distance. Even accepting that the PCIS would shine signals out more slowly than the original system (the strength of the PCIS being its spatial discrimination, and temporal discrimination matching that of a single channel photomultiplier tube having never been an objective of the design – see the LN-CCD description above), placing the LEDs close to the sample would be critical.

2. The LEDs have to be below the filter pack (or rather, the filter pack needs to be between the LEDs and the detector), and as discussed the filter pack needs to be as low as possible – therefore the LEDs must be even lower.

Finally, since the vacuum window needs to sit between the sample and the LEDs, it must be as close to the sample as can possibly be arranged. (Placing the window above the stimulation LEDs would lead to enormous difficulties maintaining the vacuum around the numerous electrical connections.)

How these constraints combined to determine the final design of the source turret is explained below.

### 2.6.1.1 Conceptual Design of the Source Turret

Since the principal components of the source turret need to be as close as possible to the sample, it is necessary first to consider the placement of the sample disks within the Minisys evacuated sample chamber. This arrangement is shown in Figure 2.6.1 below.
The sample disks are carried on a sample wheel (carrying up to 48 disks), which rotates within the sample chamber to bring particular disks into line with either the beta source or the read-out port (as shown). At the readout port the sample is lifted from the wheel by an electrically powered heater plate (shown in red). The sample disks generally are of stainless steel, 9.7 mm diameter and approximately 0.9 mm thick, although other materials and thicknesses, and indeed even hollowed planchettes, are available. The clearance between the disk and the underside of the Minisys lid is just over 1 mm (with standard sample disks), and this sets the maximum sample thickness (unless a hollowed planchette is used). Thicker samples are generally swept off the disk when the sample wheel is turned.

The source turret must not further restrict the sample thickness allowed, and so no part of the source turret (including the retaining hardware for the vacuum window) may extend below the lower surface of the Minisys lid. The actual vacuum window itself must maintain at least the same clearance from a sample raised on the heater plate as exists between a lowered sample and the Minisys lid. The arrangement of the vacuum window to achieve this is shown below (Figure 2.6.2). Note the placement of O-rings to maintain the vacuum seal, avoiding the assembly bolt holes.
Next, the internal shape of the source turret was adjusted to provide for the installation of LEDs, and to provide optical baffling for a filter drawer. In Figure 2.6.3 below can be seen the conical collar through which the LEDs will protrude, and the multi-step shape which forms the light baffle against stimulation photons around the sides and back of the filter drawer.

Please note that these diagrams are representational only, and are not to final scale. They show the emergence of the necessary features of the source turret as the conceptual design progressed, but before the detailed design and dimensioning of the components. The final design, with all correct dimensions and angles, is shown in the representative engineering drawings which appear in the following sections. In particular, the LED collar does not over-shadow the sample disk as appears in these diagrams. In fact, the total obstruction created by the LED collar amounts to less than ½%, at the extreme corners of the rectangular-pyramidal light path.

![Diagram of LED mounting collar and filter drawer lower light baffles.](image)

**Figure 2.6.3** The LED mounting collar and the filter drawer lower light baffles.

From this outline, calculations showed that the LEDs should be aligned at 30 degrees to the horizontal. Since the LEDs should point directly at the centre of the sample disk (after allowance for the change in the light path due to the Spectrasil™ vacuum window itself), any gentler an angle would lead to interference with the O-rings and the vacuum window retaining ring attachment hardware; any steeper an angle would reduce the size of the LED collar’s central aperture, and lead to significant obstruction of the light path. A steeper angle would also force the base of the filter drawer too high, increasing the size (and cost) of the filters required.

This arrangement is shown in Figure 2.6.4 below (though note the dimensions and angles are not true), which shows the LEDs protruding through the LED collar and almost touching the vacuum window. This arrangement does not obstruct the light path, and reduces the LED chip to sample distance by about 5 mm (from approximately 30 mm to approximately 25 mm), resulting in an increase in stimulation intensity of around 45%.
In the following photo (Figure 2.6.5) we can see the vacuum window retaining ring (if not the window itself, but trust me, it’s there), the main vacuum seal O-ring and the bolts for attaching to the Minisys lid. The LED wiring groove is covered by black insulation tape (to keep the wires contained), and cannot be seen. Five blue LEDs and one UV LED can be seen protruding from the LED collar, and are almost contacting the vacuum window. The blue LEDs would later need to be drawn back to flush with the collar so that filter material could be placed in front of them. The red area is a photographic artefact caused by a reflection from a filter in the filter drawer.
According to this design a drawer containing 48 mm by 72 mm rectangular filters could be accommodated (with 2 mm thick walls reducing to 1 mm where the baffle steps occur). This would allow an internal drawer depth of 14 mm before the top of the drawer would cause obstruction of the light path, and therefore allow up to 14 mm of optical filters to be used. This exceeds the design objective of supporting at least a 10 mm thickness of filters. At this stage, however, there was still the possibility that the filters would need O-rings and a positive retaining mechanism to prevent light leakage around them, and so the nominal 14 mm available was not regarded as excessive.

From a brief inspection of the above diagram it is obvious that the cut-out for the filter drawer intersects the tubes in which the LEDs are to be mounted. The problem is actually slightly worse than appears, because (a) the filter drawer is actually wider than appears in the diagram, and (b) the diagram shows a central cross-section, and the LED tube that has the most direct angle away from the filter drawer. As we go round the circle of LEDs in the LED collar, the angle of the tube in front projection comes closer to vertical, and the interference between the filter drawer cut-out and the LED tubes becomes more extreme. Most of the LED tubes are in fact completely severed by the filter drawer, especially those at the front of the source turret where the filter drawer must slide out.

The problem with this design is obvious – where will the wiring for the LEDs run, and how will it get there? The first question at least is answered by Figure 2.6.6 below, where it will be seen that fine holes (2 to 3 mm diameter) would need to be drilled or bored from the edge of the turret to intersect the LED tubes, and the LED wires brought out below the filter drawer via those holes. The tubes could then be blocked (for external light exclusion) above the LEDs but below the filter drawer.

Unfortunately the source turret sits in a recess in the Minisys lid, so that all the wiring would need to run in a groove in the lower lip of the turret, and be brought out at the rear of the turret where the filter drawer was not an issue. The groove would need to sit outside the mounting bolts (which radially would be placed between adjacent LED access tubes); and the mounting bolts to sit outside the main vacuum seal O-ring groove. These requirements limit the dimensions of the groove to 3 mm wide by 3.5 mm deep, requiring thin wires and careful placement and packing.

![Figure 2.6.6 Source Turret design, showing wiring access and light blocks in LED tubes.](image-url)
The answer to the second question – how the wires will actually get there – turned out to be a little more involved, and will be explained when the assembly of the source turret is described.

The filter drawer itself – and the arrangement used to baffle its front edge against the entry of room light – are described later. For the moment it remains to provide a top lip for the filter drawer cut-out (to hold the filter drawer firmly against the lower baffles, and to help block any stimulation photons coming past the edge of the actual filters), and the light-proof mating flange to the optics assembly. If only to allow access for milling the complex shape of the filter drawer cut-out, this is best done by attaching a separate part (somewhat prosaically called the Source Turret Upper Body) to the top of what we will now call the Source Turret Lower Body.

The parts described so far, with a circular vacuum window over the sample and a ring of stimulation LEDs, are essentially circular in format (albeit with a rectangular cut-out for the filter drawer). The upper body is essentially rectangular in shape, and provides the mating flange to match the rectangular mating flange on the lens box. The basic arrangement is shown above in Figure 2.6.7. The overall design of the source turret lower body is now essentially complete. With the front of the circle cut off to form a "D" shape so that there is a flat surface to mount the filter drawer mating flange, but with a 4.6 mm base left as a full circle to carry the LED wiring groove, and with the various assembly and mounting bolt holes added, the source turret lower body appears somewhat as shown in the perspective sketch shown below in Figure 2.6.8. This sketch was provided to the workshop staff to aid their understanding of the (plan and section) engineering drawings.

The recesses seen in the side of the lower body are necessary to provide the facility to bolt the source turret to the Minisys lid, and because of the LED tubes, the main vacuum seal O-ring and the
LED wiring groove, are in the only place they could possibly fit. Their location meant that they could only be produced using electrical discharge machining, and an appropriate brass die was made for the purpose. (New blind threaded holes were tapped in the Minisys lid to match these locations. Fortunately there was no interference with the three holes used to mount the original turret and photomultiplier tube.)

Figure 2.6.8 Perspective sketch of the Source Turret Lower Body.

Figure 2.6.9 The source turret. Note the side-recesses accommodating the mounting bolts.
The recesses are also used for four upwards-facing bolts helping attach the upper body to the lower, leading to a useful consequence that I would like to, but can’t, claim was planned. The downwards-facing bolts attaching the turret to the Minisys are longer (at 16 mm) than the upwards-facing bolts above them (10 mm). The 16 mm long bolts, with washers, are placed through the lower holes, and the shorter (10 mm) bolts still have room to fit into their (upper) holes and be tightened during assembly of the turret. The lower bolts no longer have room to come out (a fact deduced during one assembly, when I discovered that they no longer had the room to fit in once the upper bolts were installed). The lower bolts are now captive bolts, and cannot be dropped into the interior of the Minisys when the turret is being attached (in the dark and with difficult access) to the Minisys lid.

The final major component of the source turret is the filter drawer mating flange. This attaches to both the lower and upper bodies (concealing the join between them at the front of the turret), and provides a complex light baffle (a series of grooves and fingers running around the edge of the drawer front). This protects against external (room) light by interlocking with the correspondingly-shaped front of the filter drawer. The bolts attaching the mating flange to the source turret need to be recessed into the bottom of (the larger) groove so they do not interfere with the light baffling; and their positions need to be carefully chosen so that the bolts meet metal, and not LED tubes. The photo below reveals that for the lower two attaching holes this required two attempts; the LED tubes just would not hold a thread! The photo shows the front of the source turret with the mating flange attached to the upper and lower body parts.

Figure 2.6.10 detail of the source turret filter drawer mating flange and light baffle.
2.6.1.2 Detailed Design and Dimensioning

The detailed design and dimensioning of the source turret is presented via a selection of the engineering drawings from which the components were produced. The engineering drawings for the PCIS were all produced by me using traditional pencils and paper, and my daughter’s old school drawing board. There were several reasons for electing to do it that way, rather than using a Computer Aided Design (CAD) tool:

- A CAD tool doesn’t do a design for you; the designer still has to fully understand what is needed, and how to shape the parts to meet the needs. The computer system simply documents the design. I felt that drawing each part with pencil and paper, calculating each dimension or clearance using trigonometry and a calculator, would help me to understand and check the design as it emerged; a computer drawing package would not do that.
- A real and complex job is not the right setting for learning new tools and methods. I feared that using an unfamiliar and complex drawing package might lead to design errors not being recognised. By calculating the dimensions and then producing scale drawings by hand, any errors in the calculations should be revealed.
- Although I had learned hundreds of new computer systems in my life, I had never learnt to do engineering drawing (Tech Drawing had not been considered sufficiently “academic” for me when I was at school). This would be my opportunity to learn how you design and draw parts so that things can actually be made. A CAD package would not teach me that.

By no means are all of the hundred-plus original A4 and A3 drawings presented here (though they are all available as full-size pdf scans). The aim is to present just those drawings which best illustrate the actual implementation that emerged from the design process; and which help to demonstrate the precision with which the parts were dimensioned to ensure that stimulation intensity, signal photon capture and extraneous light baffling were all optimised.

Figure 2.6.11 Representative “fore-aft” cross-section through the centre of the Source Turret Lower Body.
The first drawing (Figure 2.6.11 above) shows a central “fore-aft” section through the source turret lower body. Being parallel to the major axis of the filter drawer it is a “true” section, in that there is no distortion of the position of the back of the filter drawer due to perspective, and so it shows the true degree of interference between the filter drawer and the relevant LED tubes (in this case the tubes for the two 5 mm UV LEDs).

As can be seen in the photo at Figure 2.6.9 above, the outer rear LED tubes are also only partially occluded by the filter drawer. This fact was used to advantage during the initial test phase assembly to allow the rear 5 LEDs to be installed “the easy way”, without diverting their wires to the small secondary access tubes. The front LEDs can only be installed the “proper” way, and only one front LED was installed for the test phase (and that only to prove that it was possible).

The left-right central cross-section at Figure 2.6.12 below is also a true section, but does not show any of the large side LED tubes, since there are none at that location. It shows how the base of the filter drawer completely occludes the edges of the inverted cone aperture restricting the leakage of stimulation photons into the filter drawer light baffle area. It also shows the nominal central line at 30 degrees to horizontal along which all the LED tubes and the utility stimulation port will run, showing that they all point to the centre of the sample position (as corrected to allow for the vacuum window).

This section contains threaded holes for both vacuum window retaining ring, and for upper-to-lower body assembly, precisely because it doesn’t contain any LEDs or other elements. All such bolts had to be placed where they would fit, as can be seen in the plan projection of the source turret lower body at Figure 2.6.13.

![Figure 2.6.12 True "sideways" cross-section through the centre of the Source Turret Lower Body.](image-url)
All of the labelled cross sections identified in Figure 2.6.13 were produced so that the calculated dimensions and clearances could be checked, but also so that the workshop had complete specifications of the shapes and locations of the four types of access tubes required. Different access tubes were needed for:

- the 5 mm clear plastic package blue LEDs (8 of);
- the 5 mm metal can package UV LEDs (2 of);
- the 12.7 mm clear plastic package red and IR LEDs (total 3 of); and
- the utility/LLG stimulation port (16 mm, 1 of)

Figure 2.6.13 Vertical projection of the source turret showing placement of the LED and detector tubes.

The two nominally 3 mm holes shown in Figure 2.6.13 at the rear of the source turret are provided for two light detectors, and actually sit just below the 5 mm LED tubes, but still within the face of the LED collar, and are shown in separate cross sections (not reproduced here). They are also connected to the LED wiring groove via fine access holes, so that all of the source turret wiring can be neatly hidden and protected from damage. Calculations were made to ensure that there would be no interference between the LED tubes and the detector tubes.

Figure 2.6.14 below shows the cross section containing the liquid light guide (LLG) port. With this drawing (as with others of the cross sections) I started by photocopying a "standard" transverse...
cross-section and then adding the individual access tube details to it. Because of this – and as noted on the drawing – the edge of the filter drawer cut-out appears (slightly) further from the LED tube than it should, and so doesn’t represent the true clearance between the tube and the filter drawer. (The clearances between the tube and the steps at the base of the turret are shown correctly, as the latter are circular not rectangular.)

In the case of the LLG in particular, no interference between the filter drawer and the access tube can be allowed (since this case involves not simply wires, which can be run elsewhere, but the whole light guide itself). Separate calculations were carried out for all such potential interference points, but were not transferred to the final engineering drawings – only the dimensions they proved. A small sample of the calculations involved in determining the LLG placement shown in Figure 2.6.14 are reproduced below as Figure 2.6.15.

Figure 2.6.14 Source Turret Lower Body cross-section showing the Liquid Light Guide access tube.

Note that the back of the LLG access tube intersects not just the side wall but also the top face of the source turret lower body. This dimension would be used to determine the maximum width of the (rectangular) upper body, and the placement of the assembly holes, so that the LLG (or other auxiliary stimulation source) could be inserted and removed.
Figure 2.6.15 Calculations determining the placement of the Liquid Light Guide access tube.

While the vertical plan in Figure 2.6.13 above defines all of the stimulation and detection access ports, it does not show the details of the filter drawer cut-out. These are shown in Figure 2.6.16 below, along with assembly bolt location co-ordinates calculated from the angular positions shown in Figure 2.6.13. The cross sections F and G referred to in Figure 2.6.16 are not reproduced here. They are similar to Figures 2.6.12 and 2.6.11 above respectively, but include dimensional details of the filter drawer cut-out and light baffle steps, rather than of the LED access tubes.

Figure 2.6.16 Vertical projection of the source turret showing the cut-out for the filter drawer.
The source turret upper body is shown in the plan and selected cross sections below. Note that the rectangular aperture in the upper body, at 44 mm by 68 mm, is smaller than the filters and provides a 2 mm overlap all around to optically mask the edges of the filters, and to hold the filter drawer securely in place. These figures (2.6.17 to 2.6.19) also clearly show the rectangular mating flange that corresponds to the source turret mating flange on the lens assembly.

Figure 2.6.17 The Source Turret Upper Body, plan view.

Figure 2.6.18 The Source Turret Upper Body, lateral cross-section.
The longitudinal cross section below at Figure 2.6.19 shows the front/rear asymmetry of the upper body. The rear (at the left of the diagram) contains four bolt holes for attaching directly down to the lower body. The front (right in the diagram) shows the higher step height to provide room for the filter drawer underneath, and the horizontal threaded holes that allow the filter drawer baffle to attach to the upper body. The filter drawer baffle (shown at Figure 2.6.20) also attaches in a similar way to the lower body, thus indirectly connecting the two major turret body parts at the front.

Figure 2.6.19 The Source Turret Upper Body, length-wise cross-section.

Figure 2.6.20 Filter Drawer MATING Flange. Note the steps in the opening, matching the internal baffle design.
2.6.1.2.1 Filter Drawers and Inserts

The filter drawer body (shown at Figure 2.6.21) is essentially a square-cornered (48 mm by 72 mm) rectangular recess almost fully piercing a slightly larger, round-cornered (at the back) rectangular block of alloy. A small (1 mm) lip is left around the base of the recess (with larger rounded corners) to prevent the filters falling right through. (The extra support at the corners is perhaps unnecessary but, being right at the bottom of the filter pack, is well out of the light path and does no harm.)

It will be seen that the stepped baffle incorporated into the source turret lower body filter drawer rebate is reflected in the shape of the bottom of the filter drawer except that the step seen closely following the edge of the circular aperture in the source turret is clearly not replicated here. That baffle step is represented in the filter drawer by the fact that the front support corners, and that whole end of the filter drawer, are 1 mm thicker than the rear filter support corners. The extra millimetre representing the first baffle step disappears at the inner end of the 10 mm radius filter support corners; it has to, or the filter drawer wouldn’t close.

![Figure 2.6.21 Plan and cross-sections of the filter drawer body.](image-url)
The filters in any case need to sit 1 mm above the floor of the lower body recess, since that is the thickness of the lip around the drawer that supports them – and the lip itself needs to slide over the turret body baffles when it is inserted. That is where the apparent “missing millimetre” has gone.

The inserts which carry the 44 mm circular filters however are able to make use of the extra millimetre, and the more complex shape of their lower surface (see Figure 2.6.22) more closely reflects the shape of the recess in the turret lower body. Because the insert uses the extra millimetre, and because its own filter support lip is 1 mm thick, the circular filter ends up at exactly the same base height as the rectangular filters, and the same 14 mm total depth can be accommodated. It should be born in mind however that the degree of occlusion of the light path (reducing the PCIS sensitivity) will increase with the maximum height of the 44 mm circular filters, in a manner that is difficult to calculate, but which will in any case be less than 50 % attenuation.

Figure 2.6.22 generalised drawing of filter drawer inserts to carry 44 mm circular filters.
Two inserts were made for each of the two filter drawers, to accommodate different thickness filters. The figure above shows the generalised drawing from which the two different-thickness dimensioned drawings were produced. In addition, a set of shims was made for each drawer, again in varying thicknesses, to allow the inserts to be extended for great depths of 44 mm filters. Each such shim is a simple 48 mm by 72 mm milled flat alloy rectangle with a central through hole of 44 mm diameter, and then black anodised.

A full set of aperture masks was also produced for each filter drawer. These are like the shims described above, but with smaller central through holes. Aperture masks were produced with 24 mm, 12 mm, 6 mm, 3 mm and 1.5 mm aperture (with the three smaller hole sizes edge chamfered) in 2 mm thick aluminium alloy flat milled plates (black anodised). Although nominally representing 4-fold (or 2 f/stop) reductions in the light path between each pair, the shape of the light path, and the fact that the plates are not placed at the calculated correct focal point for an aperture, means that the true reduction in light intensity is difficult or impossible to calculate. Nevertheless the aperture plates will restrict photons, so that they will reduce saturation of the CCD for particularly bright samples; and they will selectively mask the corners and ends of the mirrors, so that the smaller apertures will (and do) improve both the depth of field and the effective acuity of the optics.

The front flange of the filter drawer (Figure 2.6.23) has both a central boss that engages accurately with the corresponding recess in the filter drawer body, and also the baffle maze that mirrors the shape of the engaging ring on the source turret body. These features ensure the effective exclusion of ambient or room light photons from the filter area or the signal path. A handle (rectangular on drawer A, round on drawer B, to aid handling in the dark) attaches through the front plate into a blind threaded hole in the filter drawer body to hold the entire filter drawer assembly together.

Figure 2.6.23 Filter drawer front and light baffle, plan and cross-section.
Most of the major components of the source turret, including the two main body parts but excluding the filter drawer shims and aperture plates, are shown laid out ready for assembly in Figure 2.6.24 below. In fact the parts have just been disassembled after the early test phase of the PCIS and have been cleaned and prepared for anodising. The reduced test-stage LED set (5 blue LEDs and 1 UV LED) can be seen in the plastic bag at the upper left. The filter drawers themselves had not yet been disassembled when this photo was taken.

Figure 2.6.24 Major components of the source turret prior to anodising. Filter drawers are assembled.

Figure 2.6.25 The initial (unanodised) assembly of the source turret with the blue LEDs powered.
2.6.2 Initial Assembly

The initial assembly of the source turret occurred prior to anodising, and incorporated only a subset of the LEDs. It was designed to operate with the early test or “lunch box” electronics which in any case only provided drives for two strings of LEDs (blue and UV).

The purpose of the initial build was to test and prove the performance of the optics (see above for the defects in alignment exposed by this testing, and the revised alignment procedure adopted as a result); the performance of the stimulation sources and detector (determined to be adequate for blue-stimulated OSL in UV from quartz, but see below); and the viability of the integration architecture planned for implementation in the custom electronics that would support the final PCIS.

A further point of the test assembly was to prove that it was possible to install the LEDs into their tubes and get their wires coming out of the fine wire holes into the LED wiring groove. To this end one blue LED was installed at the front of the source turret where the wiring is most difficult. At the rear of the turret I installed the full complement of four blue LEDs and one UV LED. The wiring for these was done “the easy way”; the wires emerge unprotected from the LED tubes, and are joined externally to the turret. (It was determined earlier that it was physically possible – if messy – to wire the rear LEDs this way.) The different wiring methods are shown below (these photos were actually taken during disassembly of the source turret). In the first photo (Figure 2.6.26) showing the “easy way” wiring at the back of the turret, various other construction details are also visible: the unused light sensor tubes; the two vacuum seal O-rings; and the captive attachment bolts in their recesses.

![Figure 2.6.26 LEDs installed at the rear of the source turret, wired the “easy way”](image-url)
The side holes of the source turret (red and IR LEDs, and the LLG access tube) were not used at this stage, and were blocked with ball bearings and Q-compound. Q-compound was also used behind each LED installed, and can be seen around the wires emerging from the LED holes in the above photo of the rear of the source turret.

The photo below (Figure 2.6.27) shows the wires for a single blue LED emerging in the correct fashion from the access tube into the LED wiring groove. The basic means of achieving this will now be explained. (The full description of the source turret LED installation is split between the “Final Assembly” section below, and the appendices.)

![Image of LED installation](Image)

*Figure 2.6.27 One LED installed at the front of the source turret, wired the “correct way”.*

Due to the limited thickness of the flange below the filter drawer, the front LED wire access tubes could only be 2 mm diameter. Since 2 wires must emerge from each hole, each wire must be less than 1 mm in external diameter (i.e. including insulation). The wires would need enough copper to safely carry several hundred milliamps (preferably in multiple fine strands); and the insulation would need to be tough enough to withstand being dragged around a sharp 30 degree “V” edge in freshly machined alloy. A surprisingly extensive search ensued!

Eventually an appropriate wire was sourced, with adequate multi-strand copper, a tough Teflon insulation, and an external diameter of 0.9 mm. A 25 metre roll was obtained in each of red and black – sufficient for several assemblies of the turret.

---

Q-compound is a very dark, carbon powder-saturated non-hardening putty that is extremely effective at excluding light. Largely unknown elsewhere, it seems to be ubiquitous in the “low light” trades. I don’t know its origins; I inherited half a tub of it which, since it is largely re-usable, has been sufficient for all my needs.
The LEDs are installed into the source turret lower body part with no other parts attached — just a bare piece of metal. A piece of fuse wire (or similarly fine copper wire) is passed through the narrow wire access tube from the outside, and emerges via the corresponding LED access tube at the centre aperture of the turret. Here half the wire is pulled through the vacuum window aperture, and then threaded back out through the LED tube (not the wiring tube) to the outside of the turret. Both ends are pulled tight, and there is now a V-shaped fine wire running from outside the turret, down the LED tube, and back out the wire access tube to the outside again. This will be used to pull the final wires through after the LED is installed. (It actually turns out best to install 3 pull wires — see below.)

The LED terminals are now cut about 5 mm from the LED body, lengths of the Teflon-insulated wire are soldered on, and heatshrink insulation installed. Note that this lead length is less than the minimum specified for soldering, so a good heat sink must be used. I employ fine-nosed pliers, with an elastic band around the handles, held in a small bench vice. The tips of the pliers can be ground to give the most effective shape for the heat sink. In the final assembly, each LED was tested before and after installation to detect any damage. It seems that the shorter wavelength the LED, the easier it is to damage (with heat or static electricity), and the UV LEDs are particularly susceptible. An anti-static mat and earthed wrist strap were used for all stages of the electronics assembly.

The LED, with long wires attached, is now pressed down the LED tube past the fine wires installed earlier. (The plastic-encapsulated LEDs all have a flat that allows room for the fine wires, as do the stud-shaped 12.7 mm red and IR LEDs. The 5 mm metal can UV LEDs have a metal tab that needs to be snipped off, and this can be done so as to also leave a small flat in their base flanges.)

The exposed ends of the red and black wires are now each, in turn, stripped, tinned and securely and neatly soldered to the end of one of the fine wires emerging with it from the LED tube. The other end of the fine wire (emerging from the wire access tube) may now be used to carefully draw the fine wire, and with it the insulated wire, back down the LED tube, and finally out of the fine wire access hole. The technique involves inserting a tool with a softer edge down the LED tube into the inside of the “V” formed by the pull wire. Then, by gently and repeatedly pushing in on the tool, and then pulling out on the pull wire and the tool simultaneously, it is possible to gradually pull the whole wire assembly through. Care needs to be taken as the soldered join passes the 150 degree bend (especially on the second wire!), and the Teflon insulation must not be damaged. It is important not to twist either the pull wires or the insulated wires together as they are installed, or the pulling operation will not be possible. The tool I used was cut from a section of a plastic spiral binder, with a narrowing section and a small “V” cut into the end. Different size spiral binders supply tools for the different size LED access tubes.

If one of the pulling wires breaks while the insulated wire is still accessible in the LED tube, then the insulated wire is re-extracted, and the pull wire is removed and discarded. The third pull wire (remember I said to install three) is then used, not as you might expect to pull the insulated wire, but to pull two new pull wires into place. One of these is then used to pull the insulated wire.

If the pull wire breaks after the insulated wire has disappeared into the LED tube (or if the last pull wire breaks) it is generally necessary to remove the LED and start again from scratch.

It is not simple, but it is possible; and the concealed wiring design will work successfully for installing the full complement of LEDs and detectors in the final source turret assembly.
Once all the electronics are installed, the lower and upper body halves and the filter drawer engaging flange are loosely assembled. O-ring material should be cut (if necessary) and fitted in the grooves between each joined face. Remember to install the source turret mounting bolts before installing the four upwards-facing body half connecting bolts that make them captive. There are six downwards-facing connecting bolts, and six bolts attaching the filter drawer flange (four to the lower body half; two to the upper). All of these bolts must be left slightly loose, and will be tightened only when the filter drawer is fitted to the turret.

The filter drawer (only one existed at the first assembly) is now assembled (tighten its assembly bolt), and fitted to the source turret. The upper and lower body halves, and the engaging ring, must now all be hand-adjusted into place so that the filter drawer operates smoothly in and out, and snugs neatly into place when pressed home. The 16 connecting bolts should now be tightened gradually while continuing to check the fit of the filter drawer. This may take several attempts and some time, but a successful result will be achieved with patience.

The two lower vacuum seal O-rings can now be installed, the Spectrasil™ vacuum window and its retaining ring put in place, and the ten retaining bolts installed and tightened. (Some may feel a little rough, as these bolts had to be cut down to fit the threaded holes located between the LED tubes).

The upper O-ring may now be installed (providing the light seal to the optics assembly), and the D15 plug attached to the LED wires. (Refer to the wiring diagrams and pinout charts in the appendices, and be sure to bridge the pins intended for the “Lens Down” microswitch which has not yet been installed.)

The Source Turret is now assembled in its test form, and is ready for use.

![Figure 2.6.28 The completed Phase 1 source turret, ready for PCIS testing.](image)
2.6.2.1 Blue LEDs have an Ultraviolet tail!

The testing of the PCIS conducted with the interim source turret revealed two problems with the LED strings – actually two “well-known facts” about LEDs that turned out to be false. The first proved to be only a minor hiccup; the second was more troublesome. (The minor problem actually emerged later with the finer current control provided by the ICU, but is more logically described here.)

First the minor problem. It is a well-known fact that LEDs, unlike filament globes, retain (or even improve) their electrical efficiency when they are dimmed, all the way down to near-zero light levels. Since LEDs are necessarily current-controlled devices (their voltage changing only a small amount, and in a highly non-linear fashion, as the power dissipated in the device changes), this is equivalent to saying that there is a near-linear relationship between input electrical power and output optical power all the way from no current to the full rated current of the device. This turns out to be true only down to a couple of milliamps, below which there is a sudden and total cessation of light output. Again, the effect appears to be worse with shorter wavelength LEDs. (On the other hand, and surprisingly, it was possible in tests of blue LEDs to get increased brightness with increased current up to several times the device rating, for several hours, without destroying the LED.)

The low current behaviour of the UV LEDs caused an inability to record focussing images through UG11 filters using the focussing parameters that had been selected – 0.1 mA through the (then) two UV LEDs for 1 second. Given the known sensitivity of the CCD, and the desire not to saturate it excessively, low initial values had been deliberately chosen for these parameters. Increasing the exposure time had no effect; only when the UV LED current was increased to 0.3 mA did the problem (and with it the fears of some more serious system failure) disappear. I now simply run higher currents (minimum 0.5 mA), and use shorter exposure times (down to 0.1 second) when necessary (which is generally only when focussing for a TL run with no optical filters).

The second – and far more serious – problem emerged immediately tests were performed with the initial optics and turret. With a full 6 mm pack of notch-coated UG11 filters in place, and with stimulation provided by the five 20 mA blue LEDs, a sample of quartz grains was clearly visible on the CCD even when all the UV luminescence signal appeared to have been shone out.

Either the sample wasn’t being bleached properly by the blue LEDs and was continuing to give off a faint (but detectable) signal over long periods of stimulation; or light from the blue LEDs was penetrating the filter pack.

Tests with a daylight-bleached sample, and then with an empty sample disk, confirmed that it was not a slow, faint luminescence signal. The LEDs were clearly emitting some photons in the wavelength band passed by the UG11 filters. The blue LEDs had a UV tail!

The intensity of the image seen from the leakage photons with a 60 second exposure was around 10 times the background noise level (which is around 10 counts per pixel). At least one, and preferably two or more orders of magnitude reduction in the leakage photon signal would therefore be required in order to eliminate the problem.

(These tests at least confirmed that the blue LEDs, as it had appeared, were effectively shining the signal out of the quartz in reasonable time – 30 seconds or less – and so the blue LED stimulation...
was actually intense enough. The external light source and the liquid light guide, with all their attendant problems, would thankfully not be needed for normal quartz OSL dating applications.)

So the second “well-known fact” about LEDs – that they are to all intents and purposes mono-chromatic – was also false. No emission this far off band (in this case at about half the nominal wavelength) was evident from any graph of LED spectrums that I was able to find from any LED manufacturer, anywhere on the web. And really, fair enough – the LEDs do not produce any useful amount of light at these wavelengths – just enough to be detected by the most sensitive camera system available, which probably isn’t the reason LED makers think people are looking at the graphs.

(Although I knew that LEDs were not truly mono-chromatic, and indeed manufacturer’s graphs show a measurable spread of wavelengths, I was extremely surprised to find that an LED running at about 3.2 volts was able to emit any photons at all of short enough wavelength to pass the UG11. As the graph at Figure 2.6.29 shows, the long cut-off of the UG11 is at about 380 nm, or 3.25 eV photons. However, as Schott’s own data shows – see Figure 2.6.30 – there is a very small transmission out to about 400 nm, or 3.1 eV, and this, with possibly some assistance from available thermal energy, must be sufficient to account for the photons seen to pass the 6 mm UG11 pack.)

The only solution would be to block the UV photons with some sort of optical filter between the “blue” LEDs and the sample. This filter would have to sit directly in front of the blue LEDs. Either the filter material would need to conform closely to the shape of the 5 mm LED, or the blue LEDs would need to be pulled back by 5 mm to be flush with the retaining collar so that a flat filter material could be used. This would result in about a 30 % reduction in blue stimulation intensity (see the earlier calculations relating to the LED placement), and should be avoided if at all possible.

### SCHOTT UG11

<table>
<thead>
<tr>
<th>Thickness in mm:</th>
<th>1.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wavelength in nm:</td>
<td></td>
</tr>
<tr>
<td>Transmittance</td>
<td></td>
</tr>
<tr>
<td>Internal Transmittance:</td>
<td></td>
</tr>
</tbody>
</table>

![UG11 UV band pass filter spectrum](image.png)

Figure 2.6.29 UG11 UV band pass filter spectrum. Image from UQG Optics web site.
2.6.2.1.1 Selection of the UV Filter material

The UG11 transmittance spectra above show clearly the intended band pass between about 250 nm and 400 nm, and the secondary unintended transmittance peak centred at just over 700 nm that the notch coating was intended to block. (We had applied notch coatings to one 44 mm diameter circular UG 11 filter, and to one 48 mm by 72 mm rectangular filter.) Given that the centre of emission for the lnGaAs blue LEDs is at around 470 nm, it seemed most likely that the leaking photons were somewhere in the 390 nm to 400 nm region, and not in the longer wavelength UG11 transmittance windows. This was confirmed by adding a long wavelength blocking filter to the filter pack, and observing no difference in the level of the image produced by the leakage. (The particular filter used was selected from our extensive set of 44 mm filters by Dr Spooner, simply to confirm our belief that the leakage was in the UV band, and I do not have a record of the actual filter type.) The object of the filter to be applied to the blue LEDs would therefore be to block just that part of the blue diode emission that would pass the UV window of the UG11. The filter would need to be flexible and thin, and so would need to be plastic rather than glass. A thick sample swatch of 100 or more plastic filters was obtained from Lee filters, which included both optical filters (designed to pass or block specific spectra), and theatrical filters (designed for pleasing tonal qualities rather than specific spectral properties).

The spectral information supplied with the filters would be adequate for many purposes, but not for identifying whether the filters provided the degree of blocking that I required in the, say, 380 nm to 400 nm band. A rig would need to be built to test the properties of a selection of the filter materials, both for their blocking of the relevant band in the UV, and for their transmittance in the stimulation band around 470 nm. The filter materials selected for testing were two Lee Blues – so called because they pass blue light – and the “Lee UV”, so called because it blocks UV light.
The test jig was developed around a high sensitivity wide spectrum light meter, based on a 100 square millimetre silicon photo-diode detector. This meter (built in-house at the RSES) is not calibrated to any fixed luminance scale, but because of its design is quite linear across 7 or 8 orders of magnitude, and because of the size and technology of the detector is highly sensitive, and is sensitive in the relevant spectral band.

The jig comprised a tube, into which the 44 mm notch-coated UG11 filter could be securely mounted at one end, and a sample cut from the plastic filter swatches securely mounted at the other. In this case, “securely” means secure against light leaks; filter carriers fitted with O-ring light seals were used to achieve the required integrity. At one end of the tube sat an end cap fitted with two blue LEDs and two UV LEDs. At the other end was the high sensitivity detector.

The UV LEDs, passing through the UG11 filter, were used as a richer source of photons of the right energy (i.e. those that would pass the UG11 UV transmission band) than could be provided by the blue LEDs. The blue LEDs were used to test transmittance of the filter materials in blue. (The very small energy associated with the UV photons from the blue LEDs would not significantly affect the blue transmittance measurements, and could be ignored.)

Because of small measurement differences each time the jig was dismantled and assembled (attributed to small alignment differences and the highly asymmetric light distribution from LEDs generally), the measurements were repeated 10 times for each test and the results averaged.

Figure 2.6.31 The filter material test jig. The light tube is shown disassembled with the UG11 filter visible.

From the tests with UV LEDs, UG11 filter and the filter under test, it emerged that a single layer of the “Lee UV” filter material rejected over 99.9% of the UV photons passing the UG11 filter (3 orders
of magnitude rejection, exceeding the 1 to 2 orders required). In the tests using blue LEDs and the filter material under test only, the Lee UV passed over 90% of the blue photons (actually, of the total emission of the blue LEDs). A final test was performed was performed using blue LEDs, UG11 and the Lee UV filter to confirm that no detectable level of light passed that combination (the actual requirement to eliminate the leakage images from the PCIS).

The Lee UV plastic filter material was obviously an ideal material, and a full sheet of it (somewhat larger than A4; more than the few square inches required) was ordered and eventually arrived.

Because of the desire to maintain the intensity of stimulation for blue OSL there was a corresponding wish to maintain the blue LEDs in their present forward position (nearly touching the vacuum window), and not to draw them back (approximately 5 mm) to be flush with the surface of the source turret LED collar. To draw them back 5 mm would result in an approximate 30% reduction in stimulation intensity.

Extensive efforts were therefore made to deform pieces of the filter material into tiny little hats to fit the front of each blue LED. Highly polished brass internal and external moulds of the shape of a 5 mm LED were made, and various combinations of heat and pressure were tried in an attempt to get a successful "LED top hat". Unfortunately all such efforts failed, either not adopting the required shape, or tearing or melting through the filter material at the "top of the hat", and the need to withdraw the blue LEDs by 5 mm was acknowledged. A simple curved planar piece of filter material could then be (and was) glued to the LED collar in front of the blue LEDs.

The withdrawal of the blue LEDs would decrease the light intensity each contributed at the sample surface by around 30%; but the test source turret assembly had used only 5 blue LEDs, and these had been adequate to shine out the relevant quartz OSL signal in 30 seconds or less – slower than with the liquid light guide, but adequate and well tuned to the preferred cycle times of the LN-CCD detector. Going from five blue LEDs to the full complement of eight would slightly more than compensate for the reduced intensity from each (nett result about 110% of the test turret intensity).

The combination of LED stimulation and CCD detection for luminescence dating of quartz, the acid test for satisfactory optical performance of the PCIS, was proved, with only the one additional requirement that a piece of Lee UV plastic filter material be placed in front of the blue LEDs. The build of the final PCIS, including all of its automation electronics and software, could go ahead.

2.6.2.2 Some Final Points re Filters

At this point I made a mistake involving glue. Still concerned about effective light sealing around the edges of the filters sitting loose in the filter drawer, I super-glued a piece of 1 mm O-ring material to the filter-retaining lip around the base of the filter drawer. This was an error. 1 mm O-ring material is extremely unpleasant to deal with, twisting, kinking and bending with no consideration for one’s mental welfare whatsoever. (The thicker O-ring materials are far more co-operative, but I didn’t want to sacrifice 2 mm of drawer depth.) Without mechanical force on the filter to compress the O-ring, it was impossible that an O-ring would ever achieve a flatter result, or a better light seal, than the original machined alloy surface. The O-ring – and the super-glue – would need to be removed.
In fact the effectiveness of the light seal around the filter drawer and the filters, both against stimulation photons and against external room light, has been perfectly adequate in all tests with no supplementary O-rings at all. Not only are O-rings unnecessary inside the filter drawer; the O-rings I had planned to make and insert into the maze baffle around the filter drawer flange (protecting against room light) have also proved unnecessary, and have not been fitted.

The tests for stimulation photon leakage were just those which identified the UV tail on the blue LEDs — run the PCIS for a long exposure with the blue LEDs on (but with their UV filter attached), with a fully bleached sample (or no sample) at the heater plate, and with the camera shutter open. Then repeat the experiment, but with the blue LEDs off, and the shutter closed. There being no measurable difference between the two images (both are absolutely standard LN-CCD noise patterns), the total effectiveness of the seals against leakage of stimulation photons is proved.

The proofing against room light is not as absolute. Tests comprised comparing short and long exposures as well as shutter-open and shutter-closed experiments, at four different room light levels — (near) total dark, dim red room safe-lights (the normal operating condition), 60 Watt incandescent lamp pointed at the ceiling (close to normal office lighting levels), and 60 Watt lamp up close and direct on parts of the PCIS optics (the torture test). The requirement was that the first two (total dark and normal operating safe-lights) should be indistinguishable. In fact the first three (dark, normal and “office lighting”) were indistinguishable; only the torture test produced measurable photon counts at the CCD.

A little time was spent trying to identify the source of the light leak in the torture test, but without any real success. Attempts to mask various parts of the PCIS with black cloth or black insulation tape (neither of which are very good at preventing the transmission of photons; only their reflection) failed to produce conclusively identifiable patterns of change in the detected photon distributions. However, holding a lamp at arm’s length in one hand while operating the computer and the shutter with the other is not conducive to producing accurate repeatable results, so perhaps the experimental setup was to blame.

In any case, working out exactly why the PCIS failed to perform in conditions for which it was never designed did not seem to be a priority, and so the source of the leak remains a mystery. I have a suspicion that it was related to the mating flanges between the Minisys, the optics and the CCD not being perfectly aligned (the O-rings being designed to accommodate only 0.1 mm of alignment error), due to the inadequacies of the lab bench on which the components were mounted. The bench, the Minisys stand, the camera base, and also the procedure for aligning the PCIS components have all since been improved, and the problem (if indeed it is one) may have been resolved. New tests to determine this have not been undertaken at this stage, and are still not a high priority.

The only problem that has actually emerged from use of the filter drawer is the occasional observation of moiré effects when imaging a flat visual field (say an empty sample disk) using two filters together in the filter drawer. This was diagnosed as being due to interference effects from a small wedge angle existing between two very proximate glass surfaces, and was resolved by the insertion of a 0.5 mm thick spacer between the two filters, around their edges. Both aluminium and rubber sheet spacers were used (simple open rectangles, 48 mm by 72 mm with 2 mm wide edges), and both were completely effective in removing any appearance of moiré effects — which in any case were at a very low level, and would not have significantly affected any measurements.
2.6.3 Final Assembly

The final assembly of the source turret started with the dismantling and thorough cleaning of the test version, and then black anodising of all the metal parts. The cleaning needed to be particularly thorough, removing especially all of the Q-compound, so that the anodising would work correctly. Cleaning was performed mainly with ethanol-soaked pipe cleaners and cotton buds, and an acetone rinse. An ophthalmoscope is particularly useful for checking the cleanliness of the LED tubes, as the Q-compound has a tendency to stick to the LED retaining lands at the inner end of the tubes.

Anodising is an interesting process. After anodising the parts may be smaller, larger or even the same size as they were beforehand. Most anodisers will assume that you are mainly concerned with the appearance of the final result, and need to be told if you are more interested in dimensional invariance. By controlling the times and temperatures of the etching and deposition phases of the process, a good anodiser can provide excellent dimensional stability.

The anodiser I used operated out of a timber and gal shack in the old garden nursery quarter in Pialligo (a suburb which had delusions of being an industrial satellite of Fyshwick until it became a new access road for the Canberra Airport). There, surrounded by buckets and wires and heaters and bags of dye on every possible bit of floor, table or rafter space — an alchemist’s workshop in plastic and stainless steel — he practiced his mysterious arts. But, when asked, he proved most excellent at maintaining dimensional stability. (The mechanical workshop measured the bored holes in the lens box base plate before and after anodising, and they were the same to within a micron or two.)

Nevertheless, he complained about the alloys I had used for both the lens box plates and for the turret body components, telling me (correctly, as I later confirmed) what alloy numbers I had used for each, and which alloys I should have used for better results. Personally I was unconcerned by a small shell-mark here or there on the lens box, and actually thought the brown on black speckled effect on the turret body was quite appealing. I didn’t believe the light absorption of the surface would be measurably affected. I’m pretty sure I was happier than he was when I left. A master of his craft, who I wish was still in business.

The first stage of the actual assembly of the turret is a test fit-up of the upper and lower body parts, the filter drawer light baffle, and the two filter drawers. (The second drawer had by now been manufactured, speeding up the changing of filters during a PCIS run, and minimising any possible exposure of samples to even indirect room lighting.) The fit-up proved that it was still possible to get perfect snug, smooth operation of both filter drawers, and therefore that the anodising had not affected the parts’ dimensions, or the resultant closeness of fit.

The various inserts, shims and aperture plates had also been anodised, and are extremely close fits to the two filter drawers. Dimensional differences that I can’t measure mean that the best match for each accessory should be found, and the two sets of accessories then kept in separate containers marked “A” (rectangular-handled filter drawer) and “B” (round-handled filter drawer).

After dismantling the turret body from the test fit-up (the filter drawers are left assembled), the next step is the installation of the electronic components and wiring into the lower body. This is time consuming, extraordinarily finicky and described in the following section.
2.6.3.1 Installing the Source Turret Electronics

Shortly before the final source turret assembly, one of my then regular web trolls for news on LEDs revealed the sudden existence of 5-chip 100mA clear plastic 5 mm InGaAs blue LEDs. Being a standard T5 LED format, they would be direct physical replacements in the source turret for the single chip 20 mA Nichia NSPB500S LEDs that I had been using. The new LEDs consisted of five independent semiconductor chips (factory selected for matching voltage parameters so that they would current share correctly without internal ballast resistors), connected in parallel in the one package. Therefore they would run at essentially the same voltage as NSPB500s, and (since the power supplies for the PCIS were being designed to support the 250 mA IR LED anyway) would work as direct electrical replacements also.

No information was available as to the source of these LEDs, and the manufacturer in fact remains unknown to me. Nevertheless, I could purchase a bag of 30 of them out of China for less than $50 delivered – and did so. I was able to torture test a few of these LEDs until the smoke got out (see photo below), and concluded that their rating was if anything quite conservative. 200 mA for 8 hours a day for several days had no visible effect (though the actual quality and intensity of the emitted light may have been affected, and voltage drift was being noticed by the end of testing suggesting that some damage may have occurred. It finally took in excess of 1 Amp before the irritatingly bright blue glare was replaced by a gentle, soothing red glow, which was itself persistent.

![Image of smoke and LED](image)

Figure 2.6.32 Attempting to remove the smoke from one of the 100 mA 5-chip blue LEDs.

Clearly the LEDs should be good for their rated current, especially if well heat-sinked. (To preserve long-term quality of the stimulation sources, in the final PCIS all LEDs are restricted to 80% of their rated current, or 100% if over-ride mode is selected.) While some disparity between the chips, and between different LED packages in a current controlled series string, could be expected at very low currents (say less than 0.5 mA), this was not considered a particular problem, and on the basis of their high current performance the new LEDs were selected for installation into the source turret.
Although these LEDs would not provide five times the stimulation intensity at the sample (since the five chips cannot be focussed as well as a single chip, so more illumination would be off-sample and ineffective), they should provide 2 to 3 times the intensity at the sample, and would clearly settle any lingering uncertainty as to the adequacy of the blue LED stimulation.

Because the blue LEDs needed to sit 5 mm back from their original design position due to the need for the UV filter in front of them, and because I wished to provide both rigid placement and excellent heat sinking for the LEDs, individual copper collars were made for each of the eight blue LEDs. One of these is shown below, with a thin coating of heat conducting paste ready for installation into the source turret.

![Figure 2.6.33](image_url) One of the blue LED collars, prepped with heat conducting paste on the inside.

The IR, red and UV LEDs used would be those selected earlier. The IR and red LEDs are a half inch stud package containing 6 individual series-connected chips (see photo below); the UV LEDs are in a 5 mm metal can (the old standard transistor case used for BC108s and their ilk). Used were:

- **IR:** 1 of ELJ-880-228B 12.7 mm stud package 250 mA 10.5 V 880 nm peak wavelength
- **Red:** 2 of ELJ-660-225B 12.7 mm stud package 150 mA 13 V 660 nm peak wavelength
- **UV:** 2 of Nichia NSHU590 5 mm metal can 15 mA 3.5 V 375 nm peak wavelength

![Figure 2.6.34](image_url) One of the 12.7 mm stud-package LEDs (red or IR), prepared for installation.
Because of the risk of damaging the LEDs during installation into the turret (either through excessive heat, static electricity or mechanical damage to the wiring), it was decided to test each LED prior to, during and after installation. The tests would consist of measuring the voltage required to drive the LED at one half, and at its full rated current. This was the test most likely to identify any small-scale heat or static damage that could affect the performance or lifetime of the LED. Because of the difficulty of dismantling and re-assembling the source turret, it was preferable to identify any damage and replace the device immediately, prior to completing the assembly.

The photo below shows the test setup for all of the LED tests. The LED tester was made by me, and provides two different current ranges – 3 to 30 mA, and 30 to 300 mA. This would support testing all LEDs, from the 15 mA UV LEDs to the 250 mA IR LED. Simply built with batteries to supply current, and two 10-turn potentiometers to set the current for each range, the LED tester provides connection points for measuring current and voltage using two externally connected multi-meters. All test measurements were recorded, as was the final installed location of each tested LED. These records appear along with more detailed wiring instructions at Appendix B.

![Figure 2.6.35 Pre-installation testing of the LEDs. This blue LED requires 3.180 Volts to draw 100.0 mA.](image)

In order that the wiring will fit underneath the filter drawer, the leads of all LEDs need to be cut quite short. The 100 mA blue LEDs in particular come with data emphasising the minimum lead length for soldering as being 20 mm rather than the 5 mm or so which my design required. Soldering rapidly, and providing the best possible heat-sinking, would be critical if damage was to be avoided.

The photo below shows the rig used for the blue LEDs. The brass carrier is one of the components used in the failed attempt to mould the UV filter material into a LED-shaped hat; it has found a more useful role here. For the other LEDs, the pliers were used on the lead as shown below, and fingers were used as a heat-sink for the LED body. Unfortunately I had to use my own fingers, presumably explaining the tape on my left thumb in some of the following photos!
A good quality, static isolated temperature-controlled iron was used for the soldering, and all wires were tinned prior to attachment. With the iron set to a relatively high temperature, the soldering could be performed very quickly, minimising the heat transfer to the device. An anti-static mat and wrist strap were used throughout the assembly process, and the source turret itself was earthed.

Once the wires had been soldered, the LED was briefly re-rested in case of any damage from the soldering. No LED was determined to have been damaged by the soldering, so the heat sink and static protection arrangements appear to have been adequate.

The soldered LED terminals were then protected with heat shrink insulation, using a temperature-settable hair dryer rather than the cigarette lighter that I usually use. The same brass block was used to hold the 5 mm LEDs, and to provide heat dissipation during the shrinking.

Figure 2.6.36 The revised assembly for soldering the blue LEDs. Note the heat-sink built into the cathode.

Figure 2.6.37 The LED remains in the brass heat-sink while the heat-shrink insulation is applied.
Heat conducting paste was used on all of the installed LEDs (to improve the heat-sinking provided by the source turret itself), but in the case of the blue LEDs the copper tube spacers required heat conducting paste inside and out. The following two photos show the paste being applied to the inner wall of the LED tube, and then one of the copper collars (with heat-conducting paste already applied to the inside) being inserted into the source turret. In the first of the photos, three other blue LEDs can be seen, already installed with their front faces just behind the surface of the LED mounting collar of the source turret.

Figure 2.6.38 Applying heat conducting paste to the inside of the LED tube before installing the LED collar.

Figure 2.6.39 Inserting the collars for the blue LEDs into the source turret lower body.
Once the copper insert has been installed, the fine pull wires are installed as described above under "Initial Assembly". Note that although the LEDs will fit past these wires, the copper spacers will not, and must be installed first (ah, the voice of experience...). In the following photo the pull wires are seen installed ready for use in one of the front blue LED tubes. Two other front blue LEDs have already been installed, and their wires can be seen emerging into the LED wiring groove.

Figure 2.6.40 The pull wires installed after the copper collar, but before the LED is inserted.

Figure 2.6.41 Inserting the blue LED past the pull wires, into the LED tube and collar.
The preceding photo shows one of the blue LEDs being inserted past the pull wires and into its copper spacer at the bottom of the LED tube. A soft poker is used (in this case a cotton bud; plastic knotting needle pieces were also used) in order to avoid damage to the LED or the attached wires. In the following photo, the LED has been fully pressed home, and its attached Teflon-coated wires (the red and black wires in the photo) are carefully, and hopefully securely, soldered to the pull wires.

Figure 2.6.42 The Teflon-coated LED wires are soldered to the fine wires before pulling into the turret.

Figure 2.6.43 Pulling the first wire for the blue LED. Note the plastic tool protecting the wire from damage.
The preceding photo shows the first wire (in this case the black wire) being pulled into and through the LED wire access tube from the LED tube. The following photo shows the red wire being pulled; both wires are now close to home, and are starting to get in each other’s way around the back of the LED. Care must now be taken jiggling and pulling each in turn. The plastic tool (made from the spine of a comb binder) can be seen in these photos. It is used to push the wires down away from the sharp internal edge in the alloy, so that they can then be pulled a bit further from the wire access tube. Essentially it is push from the top, pull from the bottom, be patient, and don’t break the pull wire or the soldered connection! The following photo shows the wires in their fully installed position, bending just behind the heat-shrink at the back of the LED and running straight back down and out of the wire access tube. The heat-shrink needs to be visible, as in the photo.

Figure 2.6.44 Pulling the second blue LED wire, attempting to get both wires packed neatly.

Figure 2.6.45 Both wires for this LED are fully pulled, and the heat-shrink at the back of the LED is visible.
With the LED fully installed (but before gumming it all up with Q-compound), the LED is given its final test. The rig for this is shown in the following photo in which one of the blue LEDs has just been installed, and is being tested.

![Testing the blue LED for damage after installation. Showing 3.046 Volts at 468 mA.](image)

A curious feature emerged from this repeated testing of the LEDs; they require a slightly higher voltage to run after fitment to the source turret than they did beforehand. (Damage to the LED would be expected to lead to a reduction of the voltage for a particular current.) In fact in the pre-fitment testing there had always been a "settling period" of five or ten seconds during which the required voltage fell by a few hundredths of a volt, after which the voltage became stable. Once installed in the turret, the settling period disappeared - the LEDs would start at the same voltage as before installation - and stay there! Since the voltage had always been allowed to stabilise before being recorded, this leads to all of the post-installation voltages being recorded as a few tens of millivolts higher than the pre-installation figures (see Appendix B).

Some consideration led to the conclusion that the higher required voltage after installation was due to the better heat-sinking provided by the source turret (half a pound of aluminium alloy). This led to two realisations:

1. It may be possible to protect LEDs that need to be driven at or beyond their rated current by means of a current controlled supply that detects and responds to a drop in the output voltage, indicating that the LED chip has overheated. Fortunately, the PCIS does not require this as the LEDs are sufficient and do not need to be over-driven.
2. These results show that there is some contribution of thermal energy to the energy of the emitted photon – i.e. “thermal assistance” of the stimulated emission. This may help explain why the blue LEDs seem to be able to emit a few photons (the UV leakage) at photon energies greater (in electron volts) than the supply voltage to the LED. This suggests that some of the “UV tail” photons from the blue LEDs may actually be slightly shorter than 390 nm, and helps to explain their ability to pass the UG11 filters and create the “leakage image” that was observed during initial testing (and which is described above).

After these cogitations I concluded that the (surprising) higher voltage post-installation did not indicate damage to the LED, and assembly of the turret could proceed. The next step is to introduce light-blocking Q-compound behind the LED and pack it into thoroughly place, but without stressing the wires attached at the back of the LED. This is done initially with a piece of plastic knitting needle, allowing the Q-compound to be forced between the wires to the back of the LED, as shown below.

Figure 2.6.47 Packing Q-compound behind the installed LED using a cut-off plastic knitting needle.

When sufficient Q-compound has been introduced to the cavity, it is pressed and shaped with the special tool shown in the photo below. (This tool only fits the T5-size holes for the blue LEDs; the UV LEDs are slightly smaller, and the special tool cannot be used.)

The LED tubes at the front require particularly careful packing of the Q-compound, as an adequate filling is required but interference with the sliding of the filter drawer must be avoided – this is actually why the special tool was made, and a very neat result was obtained. Note that the 12.7 mm LED tubes are too large to successfully and permanently seal with Q-compound alone, and a close-fitting steel ball bearing is also used. This is shown in the photo at Figure 2.6.49 below.
Figure 2.6.48 Packing the Q-compound neatly below the filter drawer cut-out using the special packing tool.

Figure 2.6.49 The back of the 12.7 mm LED tubes were blocked with a ball bearing and Q-compound.
Finally all of the LEDs and the 3 mm light detector are installed, and the source turret lower body adopts a nightmare spider-like form, with a big fat central body and a profusion of skinny black and red legs. The wires that will eventually be brought out the back of the turret to the D15 connector need to be labelled now, while it is still possible to see where they come from. The source turret lower body now appears as in the following photo. The wires that will join to each other in the LED wiring groove do not need to be labelled.

![Image](image_url)

Figure 2.6.50 The lower body with the semiconductor devices installed, and the string-end wires labelled.

Next the wires need to be joined (to form the strings of red, blue and UV LEDs), laid in the LED wiring groove, and brought out the back of the turret for soldering to the D15 connector. Each join needs to be made in a precise way, and done so that it will sit in the right part of the wiring groove. The design is such as to minimise the greatest thickness of wires anywhere in the wiring groove - no more than four straight-through wires and one joined wire at any point in the groove.

The detailed instructions at Appendix B list:

- the initial length of wire to attach to each terminal of each device;
- the exact length to trim each wire to after installation;
- the exact path to be followed around the wiring groove by each wire;
- the soldering, folding and heat-shrink dimensions for each join;
- the pin-out diagram for the D15 connector;
- the voltage and current measurements for each LED before and after installation;
- a list of which specific LED is installed in each turret LED tube; and
- more detailed and precise instructions for the entire semiconductor installation procedure.
The following photo shows the front of the source turret after the front blue LED wires have been joined (to form the LED string), and packed into the wiring groove. The UV LED wires that can be seen in the foreground will be run over the top of the joins in the UV LED wiring, as will the wires for the 12.7 mm side LEDs, and the 3 mm light detector at the rear of the turret.

Figure 2.6.51 Wire joins for the front blue LED string, packed into the LED wiring groove.

Once all of the wires have been connected and laid in the wiring groove as per the instructions at Appendix B, the turret lower body will appear as in the photo below. All of the wires are now accessible at the rear of the turret for soldering to the D15 connector.

Figure 2.6.52 All wires joined, temporarily taped, and bundled for attachment of the rear connector.
It can be seen in these photos that the wiring in the groove has been covered with Sellotape to protect it from egress and possible damage. This is a temporary technique during construction only—it will later require a more permanent solution. This time I did not make a mistake with glue! See “Two Successes with Glues” below for a description of the method used to secure the wires.

In the following photo the upper body part has been temporarily attached (with a couple of bolts only) to aid the next stage of assembly—cutting all the labelled protruding wires to the right length and soldering them to the correct pins of the D15 connector. Note that each termination to the D15 connector is individually insulated with heat-shrink tubing, and all the wires are securely sleeved and clamped neatly in place. It is a matter of personal pride that the PCIS should never fail, or a sample be lost, due to a wiring or connection failure. The plastic sleeve and cable clamps are installed before cutting and soldering the wires, to help ensure they are cut to the correct length, and that the wiring does not need to be stripped out and done again at this late stage.

![LED wires covered and clamped, and soldered to the D15 connector with the micro-switch wires.](image)

The upper and lower body parts now need to be separated again so that the D15 connector can be fitted to its mounting cage, and attached to the back of the upper body. Note that the wires for the micro-switch have been sleeved, clamped and terminated next to the switch mounting plate. The micro-switch—which will be attached later—is used to detect whether the lens is down on the source turret. PCIS operation (and in particular the opening of the shutter to the CCD) are suppressed if the lens is not down, since the camera would presumably be exposed to room light.

The micro-switch is supported via a detachable plate so that, once the correct mounting height has been determined, the plate can be removed and taken to the workshop to be drilled and tapped for the micro-switch attachment bolts. That way, the assembled and clean source turret does not need to be exposed to the workshop environment.
The upper and lower turret body parts and the filter drawer light-excluding flange can now be assembled, using all 16 bolts, for the final time. The procedure is designed to ensure close fitting and smooth operation of the filter drawers, and is described in detail in the "Initial Assembly" section above. The only difference is that this time the correct fitment of both drawers must be achieved, so the process is just a little more prolonged. When correct operation of both drawers is achieved, all bolts are tightened. (Remember to fit the source turret mounting bolts into their holes before installing the upper/lower body connecting bolts that render them captive.)
The filter drawer closing latches (clearly visible in the previous and following photo) are now installed, and any necessary shim washers (to ensure firm closure of the drawers) selected and fitted. Note that the handle of the right hand latch later needed to be trimmed short to prevent interference with part of the optics focussing mechanism. A knurled-head bolt was added to the shortened handle so that it could still be easily operated by feel when under the lens assembly.

Figure 2.6.56 Assembled turret body, left side. Note the filter drawer retaining clips have been fitted.

Next the UV-excluding filter windows need to be attached in front of the blue LEDs. These can be seen in their final form in the photo below.

Figure 2.6.57 All of the installed LEDs are visible, with the UV-excluding filter over the blue LEDs.
Calculating and cutting the curved collar from which the window is made is simple: getting the hole for the UV LED in the centre of it, less so. In the end it was easier to punch the 5 mm hole first, and then to cut the collar accurately around it. One of the collars (the one visible in the photo above) then needs two small cut-outs to clear the 3 mm sensor tubes (only one of which is actually in use).

Glueing the windows in place on the source turret LED collar was fraught with risk. Any spillage of glue onto the front of the LEDs might damage them irrepairably; mis-placing the window and needing to remove it and its glue carried similar risks. The windows must be glued against the turret body all the way round each blue LED to prevent any UV leakage at the edge of the LEDs; yet the space around and between the LEDs is only a millimetre or so, making the correct application of the right amount of glue a problem in itself.

The windows need to stay securely in place long-term, but a re-build of the turret must be possible – so the glue must be “permanent”, but not absolutely permanent. A large number of different glues were purchased (concentrating on those that might be easier to clean up or remove), and were tested for adhesion and later removability using scraps of filter material and anodised aluminium.

The results of that search, and the method developed to apply the adhesive, are described below under “Two Successes with Glue”.

The final stage of assembly of the source turret is the attachment of the vacuum window to the base, using the ten specially machined cap-bolts. (These may feel rough going in due to the bolts having been cut, and the impracticability of cleaning up the threads with a die on such tiny bolts. Care needs to be taken to avoid cross-threading.) The underside of the source turret, with the vacuum window fitted over the LEDs, is shown in the next photo.
The completed source turret – except for fitment of the micro-switch and trimming of the right hand filter drawer latch – is shown in the following and final photo.

The correct placement of the micro-switch would be determined and marked on its mounting plate when the turret was bolted down to the Minisys base unit, and the PCIS is put into operation. (Liquid paper works well for marking anodised alloy.) It was when the full PCIS assembly took place that it was also discovered that the end of the handle of the right-hand filter drawer latch interfered with part the lens focussing mechanism, and that the handle would need to be trimmed.

Fortunately both of these parts are easily detachable from the turret (the switch carrier by design; the filter drawer latch by good fortune), and could be removed and taken to the mechanical workshop for the necessary work. The source turret was able to remain assembled and clean, and the modified parts were re-fitted in the lab.
2.6.3 Two Successes with Glue

I may have mentioned before that I occasionally made errors involving glues. I am happy to report now that on (at least) two occasions I managed to avoid doing so. This fact seems sufficiently momentous to warrant a separate description here.

2.6.3.2.1 Success No. 1: Securing the LED Wiring Groove

All of the LED and sensor wiring running in the LED wiring groove needs to be held securely in place to prevent it being damaged when the turret is being installed onto the Minisys base. The clearance between the turret and the Minisys is small; access is very limited; and the room is (usually) quite dark. If any wire came loose from the groove during installation, the chances of its being caught between the two components, and being damaged or even severed, are quite high.

A number of solutions were considered, including the filling of the groove with a "liquid insulation tape". Real insulation tape would not serve because of the lack of metal for it to stick to around various parts of the turret, and especially at the front under the filter drawer. It seemed likely that some sort of filler or glue would be needed.

But I had learned that removing glue can be difficult; and I did not want to create the situation where replacing one LED would require replacing them all, because all of their wires would be damaged during disassembly. For some time I prevaricated, and simply used a narrow piece of insulation tape and a great deal of care.

Eventually inspiration struck; the solution was that most wonderful of products, plumber’s tape.

Just sticky enough to adhere adequately to itself, but never any significant problem to remove from anything else, it has the added property that it will find a way to distort itself and wind into any thread, or other small gap, to which it is applied. Wrapped around the source turret wiring, it automatically snugs itself around the wires and deep into the groove at either edge. Also being extremely thin, there was room for several turns, resulting in a very neat, self-adhering closure.

So the successful solution to the glueing problem was, as it may often be, to use no glue at all.

As can be seen in the earlier photos, a single turn of neatly trimmed insulation tape was added, for aesthetic reasons but also to prevent the escape of any strands of plumber’s tape.

Figure 2.6.60 The final wiring groove, packed with plumber’s tape and then protected with insulation tape.
2.6.3.2.2 Success No. 2: Attaching the UV Windows

The difficulties and risks associated with glueing the UV-excluding windows over the blue LEDs, and the search for a suitable glue, were described above. Eventually I selected a craft “On and Off” glue.

You may be familiar with this product from its use in attaching give-aways – perfume sachets, disks of software etc. – to magazine covers. The glue has a much higher cohesion than adhesion, so that it tends to come off both parts, quite easily, as a single piece. Although claimed to be “re-usable”, and retaining some stickiness after removal, it never fully recovers its initial adhesive properties (which are surprisingly good). This product should provide the permanence of attachment required, yet tests showed that it could be easily removed for a re-build of the turret, or in case there was a blunder during its application.

Applying it with sufficient delicacy to achieve the all-round attachment required, without obstructing or smearing the front of any of the LEDs, was a separate problem.

Like plumber’s tape, syringes are so associated with the field where they are absolutely essential that we tend not to consider them for applications in which they are merely extremely useful. Unfortunately, due to personal and family medical issues in recent years, I had acquired a familiarity with syringes that I had never wanted; however, syringes were now very much “on my radar” when I was looking for a tool to solve a range of substance-application problems. I thought of them now.

The technique which I developed is as follows:

- Take a small syringe (I used 3 ml because I have them), and pull back the plunger to about half fill the syringe with air.
- Pour a small pool of the glue onto a non-absorbent surface and suck quarter to half a millilitre of glue into the syringe. This is best achieved by putting the tip of the syringe in the glue, pulling the plunger back by about 1 ml, and waiting.
- Keeping the syringe upright (so that the glue remains near the tip), put a relatively short, fat needle onto the syringe (I used 3/4” by 25 G, because I have them).
- Press the plunger of the syringe in until the contained air is compressed to about 3/4 or 2/3 of its original volume. (If you squeeze too far the plunger should expel itself somewhat from the syringe anyway due to the excess pressure. The maximum pressure that can be maintained is a function of the plunger wall friction and the syringe diameter. Other things being equal, a small syringe will maintain a higher internal pressure.)
- After a short while a thin, slow stream of glue should begin to emerge from the tip of the needle. The syringe may now be used to apply glue to the turret surface as though writing with a quill pen. If an excess of glue appears at the tip, simply wipe it off; if the flow rate is too low or high, simply press or withdraw the plunger slightly. To stop the flow, pull the plunger back to its original position. Pressing the plunger back in will restore the flow.

Using the above technique (and a pair of magnifying glasses) I was able to apply a well-controlled bead of the “On and Off” glue to the 1 mm wide lands between and around the LED tubes. The glue is allowed to partly set, and then the UV windows are applied. Success was achieved at the first attempt, and the result has proved excellent.
2.7 Shutter Housing

The shutter housing, aka the destination turret, is thankfully a much simpler construction than either the optics or the source turret. Basically a tube of the right diameter and length to join the optics destination-side (circular) optical port to the threaded ring at the front of the CCD, it contains the 45 mm diameter electronic shutter positioned just in front of the CCD window. It also contains a phototransistor connected to a circuit in the IICU that will prevent the shutter from opening, thereby protecting the CCD from damage, if the light level inside the housing is excessive.

The shutter itself is a purchased item that needs to be driven from a special external controller. The controller provides the shaped voltage drive to open the shutter in the shortest possible time without over-driving and damaging the shutter coil; a variety of TTL-level control and monitoring signals; and the ability to manually control the shutter from a front panel switch.

Note that the LN-CCD camera’s own electronics provide the facility to drive a shutter directly, but is only rated to drive up to the 35 mm shutter. The 45 mm shutter requires higher voltages and currents, and therefore the external controller. The external controller however provides control signals that the CCD’s internal driver does not, and which proved useful to the PCIS integration.

The shutter came supplied with a circuit to provide feedback (via a TTL output on the controller) of the actual state (rather than of the requested state) of the shutter. Since the shutter takes several milliseconds to fully open (or indeed to close), this signal allows for more accurate timing of exposures than would be possible by timing the signals requesting the controller to open or close the shutter. The feedback signal is provided by a small green LED, and a corresponding detector, that are separated from each other by the shutter blade when the shutter is fully open.

The LN-CCD of the PCIS is far too sensitive to tolerate a bright green LED, inadequately baffled, a few millimetres from the optical entry window. Fortunately, however, the PCIS is also too slow for the few milliseconds to be of concern (its forte, remember, is spatial not temporal discrimination). The green LED was defeated by the simple expedient of making a cable without the wires that drive it; the shutter and controller continued to operate correctly with the single exception that the feedback signal is lost, and all risks of light leaking to the CCD were resolved.

The front of the camera is a “D” shaped nose with the optical window in its centre, and a loose internally-threaded ring around it for securing the CCD to other equipment. The shutter is also a “D” shape, and in order to be as close as possible to the CCD needs to be fitted so that its operating coil nestles into the vacant part of the “D” at the camera’s front. This part of the shutter housing would therefore need to contain internally an offset “D” shape (one side of which would contain the CCD front; the other, the shutter operating coil), and an external thread to suit the clamping ring. The other end of the housing would be a simple ring to engage with the lens box optical mating flange.

For simplicity of manufacture, and to save materials, the shutter housing was constructed in two parts. The first is the more complex “D”-shaped part to which the shutter is attached; the other is a simple tube connecting the first part to the optics box. This overall design is shown below at Figure 2.7.1, in what has obviously been a working drawing during construction.
It should be mentioned here that the required height of the tube part of the shutter housing was surprisingly difficult to determine. To accord accurately with the optic design the actual face of the CCD had to be placed at a particular height with respect to the chosen reference; but the distance from the front face of the vacuum window over the CCD to the active face of the CCD was unknown. Several enquiries directed to the supplier and manufacturer of the LN-CCD each brought prompt, detailed and entirely different figures, and indeed diagrams of the front part of the camera.

In the end, and with a little trepidation, I took the very sensitive and clean camera to the brightly lit and slightly less clean engineering workshop. (The camera is meant to be quite resistant to bright light exposures when not powered up, but I prefer not to test such things. Exposure to that light level when operating would almost certainly destroy the camera.)

The camera, in its stand, was mounted to the base of an accurate numerically-controlled drilling and milling machine, with a high-powered microscope viewer (rather than a drill bit) mounted to the machine head. The high-powered microscope has a very short field of view—less than 0.1 mm. By focussing first on some of the (inevitable) fine dust on the surface of the window, and then moving the numerically-controlled head down until some surface detail at the edge of the CCD chip came into focus, a quite accurate measurement of the effective distance was obtained. (This effective distance is effectively already corrected for the optical effects of the vacuum window, whose thickness can now be ignored. The thickness of the vacuum window on the sample side had to be accounted for in the calculations of the correct true height of the sample.)
The measured height accorded fairly well with calculations based on an average of the "true" (i.e. not effective) figures and diagrams supplied. Construction therefore went ahead based on our measured effective distance, in the belief that any remaining error would be within the range of what the optics' focussing mechanism could compensate for, without loss of image quality.

Nevertheless, the design shown above in Figure 2.7.1 incorporates the ability to relatively easily change the height of the shutter housing should it in fact prove necessary. A shim (with another O-ring groove) could be placed between the two components to lengthen the tube; and the thick base part of the upper tube is long enough to allow several millimetres to be taken off (and a new O-ring groove installed) should the tube require shortening. Foresight that happily proved unnecessary.

The diagram below demonstrates the slightly more complex shape required for the lower part.

---

Figure 2.7.2 Plan and cross-section of the lower part of the shutter housing.
This degree of complexity is required because the operating solenoid of the shutter needs to sit beside the "D"-shaped front of the camera – in that part of the full circle that the "D" does not occupy. The shutter is clearly designed to be installed to this camera in that way, bringing the blades of the shutter as close as possible to the face of the CCD. This not only avoids any possible obstruction of part of the signal light path by the open shutter; it also assures the fastest possible clearing by the shutter blades of the whole CCD area during opening of the shutter – effectively a faster shutter opening and closing.

The large threaded base of the lower part is cut to fit the threaded retaining ring already mounted to the camera head, and serves to clamp the camera and shutter housing securely and rigidly together.

The major components of the shutter housing – including the shutter itself – are shown disassembled prior to anodising in the following photograph (Figure 2.7.3). The lower housing part is shown the right way up; the upper tube is upside-down, showing the assembly holes for bolting the two parts together. The shutter is installed after that assembly (needing to be twisted on its side to clear the narrower part of the tube at the optics engagement), and the electronics and connectors are then installed.

![Figure 2.7.3 The components of the shutter housing ready for initial assembly.](image)

Since the LED and detector mounted to the shutter have been disabled, the shutter is connected by only two wires (rather than the original seven), to a small 2-pin connector in the side of the housing. A second (3-pin) connector has a high-sensitivity photo-darlington pair soldered directly to the back of it, monitoring the shutter housing for dangerous (to the CCD) light levels. Two connectors are used, since one goes to the shutter controller and the other to the IICU; and I wanted to avoid Y-shaped cables such as those which had caused problems with the original Minisys.
In the test phase PCIS construction, a simple two-pin plastic bodied connector was used for the shutter drive. (The shutter housing light detector was not installed at that stage, in part at least because no circuitry existed for it to connect to.) This proved sloppy of fit and potentially mechanically weak in use, with no security against being pulled out during a run be any disturbance of the cable. Inserting the plug, at the back of the housing and in very low light, proved difficult.

For the final assembly metal-bodied plugs and sockets were used, with much greater mechanical strength, and a positive lock on insertion protecting against accidental disconnection. Their rigid construction and the clearly-felt engagement of the locking mechanism make them easy and certain to use, even in the dark. Their only down side is their ludicrous price – over $200 for the set! – but I felt that reliable operation of the PCIS required the expense.

The shutter and the light detector use connectors with identical bodies, but in two pin and three pin formats respectively. This makes it impossible to plug them in the wrong way round.

[In fact all custom components of the PCIS use different connectors, meaning that most of the electrical assembly of the PCIS can be done in the dark, without fear of disaster. The PCIS uses D9M, D9F, D15, D25, D35 and 7-pin mini DIN, with no connector type duplicated and all cables unique. All mains connectors are of course standard IEC fittings; and all single TTL level signals use the normal BNC connectors. But wiring errors between these identical connectors cannot actually damage anything (because the voltage levels are all the same within a connector type), and this approach allows different experimental set-ups to be implemented by patching different TTL level signals together using a set of standard BNC M-M cables.]

The photo below (Figure 2.7.4) shows the complete shutter housing, in its unanodised state as it was used in the original test phase PCIS assembly.

![Figure 2.7.4 The complete shutter housing, as used in the test phase of the PCIS.](image)
2.8 Assembly and Alignment of the PCIS

With all of the major opto-mechanical components of the PCIS now manufactured to precise designs and exacting tolerances, there remained the problem of assembling and accurately aligning the components of the PCIS. Since partial disassembly of the PCIS (removing and replacing the optics assembly) is required for sample changing, the precise and repeatable re-assembly must be a reasonably straightforward operation.

The first problem is the rapid removal and replacement of the quite heavy (30 to 40 kg) optics assembly from the Minisys base and LN-CCD camera assembly. With limited budget and resources, an accurately controllable pulley-and-rope based system was developed. This is described in the first sub-section below.

With the handling of the optics made manageable, there remained the problem of achieving accurate alignment of the optics to the other components. The existing arrangement — a lab bench built (either deliberately or accidentally) with both a slight slope and a slight roll-off towards the front; a non-adjustable wooden block stand for the Minisys; and a "V"-block stand for the LN-CCD that maintained flatness in one axis but allowed roll in the other — were not such as to make reliable and rapid re-alignment after a sample change possible.

The ultra-flat bench that I installed and the adjustable metal Minisys base and the fixed camera base that I designed are described in the following sub-sections. In the final sub-section below I describe the procedure for adjusting the Minisys base and other components so that accurate fit-up of the PCIS is achieved reliably and rapidly after each sample change.

2.8.1 Lens Positioning Rig

The lens lifting rig is essentially a chain and hangers, with a turn-buckle to allow slow and accurate lowering and raising of the optics onto (and off) the PCIS, with a "catcher chain" that allows it to be held at a point 4 or 5 inches above its rest height. A (nominally) 4 to 1 advantage pulley-and-rope parallel the chain arrangement, to aid in moving the optics between the lowered and the raised "catcher" positions; and in raising and lowering them from the bench to the chain rig and back.

At the top of the rig, three eye-bolts to the concrete ceiling have three chains, each with an included turnbuckle, meeting at a common attachment ring. Separate adjustment of the three turnbuckles allows fine control over the position at which the lens hangs; this will be used in the final adjustment to get the optics hanging directly over the source turret and shutter housing.

From the common attachment ring at the bottom of the lifting chain, four chains connect to the eye-bolts at the upper outside corners of the optics assembly. Again, each chain incorporates a turnbuckle. Careful adjustment of these four turnbuckles allows the optics to be made exactly level, so that they land on the source turret and the shutter housing simultaneously. The lower parts of the lifting rig — including the four point attachment to the optics — can be seen in the photo below. The side-pull rope-and-pulley system is hanging from the wall in the background.
To replace samples at the end of a PCIS run, the rig is operated in the following fashion:

- The large turnbuckle with the handle (visible at the top of the above photo) is operated to lift the optics a few millimetres, until they just clear the engagement rings on the source turret and shutter housing.
- The pulley system is used to lift the optics six inches or so, and the hook of the catcher chain is clipped to the lens attachment ring. The optics are lowered onto the catcher chain.

The optics are now high enough to allow the source turret to clear them when the Minisys lid is opened; and that can now be done and the sample wheel changed. If there is not another run happening immediately, the optics can be lowered safely to the bench as follows:

- First, the cover should be put over the shutter housing, and the camera assembly moved to one side and covered with a cloth. The flat bench is then covered with a fitted board.
- The side-pull rope (visible against the wall on the right in the above photo) is attached to the optics attachment ring, and the lens pulled about twelve inches to the right. The clip on the side-pull rope is now clipped to the attachment ring, and the optics are held in place.
- The optics base is now attached (with elastic straps) to the bottom of the optics box.
- The optics are again lifted an inch or so with the main pulley system; the chain is detached from the attachment ring; and the main pulley rig is used to lower the optics to the bench.
- The side pull rope may be left attached, or hung back against the wall for neatness.
The initial implementation of this system was installed using cheap components from a hardware store – galvanised steel pulleys, quick links and turnbuckles, and woven cotton utility rope. Although functional, this rig lacked smoothness and delicacy and, in my opinion and much more importantly, long term reliability. I therefore replaced it with stainless steel chain, quick links, attachment rings, turnbuckles, and pulleys; and with good quality 8 mm kernmantle construction climbing rope. This is the rig which appears in the photos, and which is still in use on the PCIS.

[Not liking to waste anything, I took the original and not completely trustworthy system home and used it to raise and store two heavy trestle tables against the ceiling of my garage – where it failed about a year later. Fortunately my distrust had led me to put catcher loops on that system too, and no damage was done to either person or property. Still, I was glad I had tested the bounds of the petty cash system, and replaced it with the good gear on the PCIS!]

2.8.2 PCIS Base

It was clear that to get a good mechanical alignment of the major PCIS components it would be necessary to start with a good, flat foundation. It would be easier to top an area of the existing bench with a new surface than to replace the whole bench, so I went on the scavenge and found a discarded air-table top. This was a slab of aluminium alloy, about 900 mm by 450 mm and about 30 mm thick, drilled with a regular array of 3 mm holes originally intended for air delivery. The slab was sent to the mechanical workshop at the Research School of Physical Sciences and Engineering, where its top surface was milled flat. I had my new bench top.

Because the bench had a significant slope, the slab would require a wedge of filler under it – I would use black silicone RTV from a pump cartridge. To make it possible to remove the slab later, and so that the silicone would not squeeze up through the 3 mm holes, I had a sheet of 1 mm aluminium cut and drilled to the same dimensions as the slab, to place under it. Four-inch alloy washers, a set of bolts, electric drill, spirit level and a quiet weekend with no-one around, and I was set.

The holes in the old air table top would allow me to locate a pair of guide rails using 4 mm dowels, so that the camera could be easily slid back into the correct position for PCIS assembly. (The camera needs to be moved – actually turned on end – for re-filling, or whenever the optics are lowered onto the bench.) The guide rails could be moved (on about a 20 mm grid) to achieve rough alignment of the PCIS to the lens lifting rig – fine adjustment is via the three turnbuckles at the top of the lens rig.

One small problem remained – the Minisys was sitting on the bench I needed to re-surface, and as it was attached by electrical, optical, nitrogen feed and vacuum lines – to the walls as well as top other components – moving it out of the way was impractical. Raising it a few inches and working under it, however, could be achieved. So I used the lens lifting rig to raise the Minisys base unit a few inches; built a support scaffold from an old bed frame and some other bits and pieces; taped it up good and solid with gaffer; and put the Minisys back down on it. (All perfectly safe and secure, but I didn’t necessarily want OH&S giving their opinion on it.)

The arrangement at this point can be seen in the photo below (Figure 2.8.2). The alloy slab for the new bench top can be seen in the background. The tapes around the Minisys are all climbing rated.
At this point I made an error involving glue\textsuperscript{10}. I underestimated the viscosity of the silicone, and assumed that the weight of the slab, plus the tension applied by eight strong bolts with big washers underneath, would be more than adequate to spread the silicone evenly. I WAS WRONG! I was surprisingly wrong. And by the time I realised how terribly terribly wrong I was, it was too late to try to lift the whole arrangement and start again. So we torqued down the back bolts until a measurable bow (actually three measurable bows) appeared in the alloy slab, and waited. (I had the assistance of my good friend and lab technician Wayne Cook for this particular adventure.) And loosened the front bolts, re-torqued the back ones and waited.

Eventually the slab was very close to horizontal, so we reduced the torque on the rear bolts to allow the slab to come flat again, lightly torqued the front bolts, ignored the wedge of silicone protruding from the front of the slab, and called it a night, relieved to have at last achieved an adequate result.

The photo below (Figure 2.8.3) shows the alloy slab finally installed and, I think, Wayne's feelings about the day's efforts and the results thereof.

\footnotesize
\textsuperscript{10} I know, the joke's wearing a bit thin, but this is the last one. Honest. The hardware build is almost done, and very little actual glue was used in the electronics and software builds anyway.
A quick visit later the same weekend to put the Minisys back down on its original base, but on the new bench, and to remove the temporary scaffolding, restored the room to normal operations by Monday. The silicone RTV had fully set, and for all the dramas and efforts on the Saturday, the end result was flat, smooth, and very close to horizontal. An acceptable outcome.

2.8.3 Camera Stand

Until this point the camera had rested in a stand that was just two "V"-shaped alloy plates on a flat alloy base — essentially a large V-block. This allowed the camera to be jiggled in one direction — the direction in which the bench curved — when trying to manually achieve a good fit of the optics on the original bench. The new bench would allow a stand to be used that held the camera completely rigid and parallel to the bench, and PCIS assembly would be simplified as a result.

In addition, the camera has to be placed on end for filling with liquid nitrogen, and this was quite difficult and unstable with the original stand. So a new camera base was designed, with solid clamping of the camera and a rigid end plate for filling. This can be seen, with the camera standing on end during the LN filling operation (itself more reminiscent of alchemy than what I ever expected from the earth sciences) in the photo below (Figure 2.8.4).
The camera was set level in the new base using an engineering flat plate and a sensitive pin-style movement gauge. The height was measured at various points around the lip of the shutter housing optics engaging ring, by turning the camera on the flat plate with the movement gauge mounted above it in a firm stand. Discrepancies in the direction along the length of the camera, which couldn't be removed by turning the camera in its cradle, were corrected with differing thicknesses of plumber's tape under each end of the camera. Stainless steel screw-adjustable straps at each end of the camera were then used to clamp the camera securely to the stand, and the camera was measured as being level to a tolerance of better than 20 microns.

2.8.4 Minisys Stand

The camera was now rigidly mounted, and could not be adjusted during PCIS assembly. To get the relative positions of the Minisys base and the camera correct, so that the optics would accurately align with both engagement rings when lowered, the position of the Minisys would need to be finely adjustable. The old wooden base would need to go, and an adjustable metal base replace it.

Four chunky alloy feet were made, each internally threaded and fitted with a large, fine thread bolt and lock nut. The heads of the bolts were machined into dome shapes to fit neatly into machined recesses in the underside of a Minisys-sized plate. Four alloy shims sit between the plate and the Minisys base unit, allowing a 10 mm slab of lead to be slipped in between. Whether necessary or not, this lead sheet had been a part of the original wooden base, and I chose to retain rather than eliminate it.
The photograph below (Figure 2.8.5) shows most of the features described in this section – the flat alloy slab bench top, the chains and ropes of the lens lifting rig, the rigid stand for the LN-CCD camera, and the new adjustable Minisys base. How this adjustment is performed to get the best possible opto-mechanical fit of the PCIS components is described in the final sub-section below.

Figure 2.8.5 The complete, assembled opto-mechanical components of the Phase 1 PCIS. The flat alloy PCIS base, the improved camera stand, the adjustable Minisys base and parts of the lifting rig can be seen.
2.8.5 PCIS Component Alignment

The opto-mechanical alignment of the PCIS major components occurs as follows.

- The approximate correct position for the camera is selected, the two guide rails are attached to the bench via the dowels, and the camera is slid firmly up against both guide rails to position it accurately.
- The Minisys is moved to the correct position, using the lowered lens as a guide and a rubber mallet for the fine adjustment, until the optics box engaging rings mate correctly and simultaneously with both the source turret and the shutter housing.
- The three upper turnbuckles of the lens lifting rig may now be adjusted so that the lens hangs in exactly the right position over the PCIS. (The four lower turnbuckles may also be adjusted so that the lens hangs level. The lengths of the main lifter chain, the catcher chain and the rope and pulley systems may also be adjusted if necessary.)
- The legs under the Minisys are adjusted to make the engaging rings of the source turret and the shutter housing co-planar, so that accurate light-tight mating of both optical connections can be achieved. This last step is the most problematic, and is described further below.

The initial height alignment is done using feeler gauges around the edges of both engaging rings, judging which way the Minisys has to move, and adjusting the Minisys legs (easily done with a spanner, and a second for the lock nut) to remove the measured gap. As a guide, a gap to the left or the right of the shutter housing engaging ring suggests that the Minisys needs to move up or down as a whole; a gap to the left or right of the source turret ring, or to the front or rear of either ring, suggests the Minisys needs to be tilted accordingly, but not raised or lowered.

The thinnest manageable feeler gauge for testing the gap is 0.05 mm. Since the design allows at least 0.1 mm before the O-ring light seal is compromised, it was felt that alignment within 0.05 mm would be acceptable, but a better alignment may lead to reliably (if marginally) better imaging.

As the photograph below will suggest, access all around the source turret and shutter housing is awkward at best, and some points are easier to measure at than others, perhaps biasing the priorities of the adjustment process. It was found that a white light in a dark room is visible through a gap smaller (I suspect significantly smaller) than 0.05 mm. But again, access all around for both a light source and my head is not possible, and only certain points can be checked. A combination of feeler gauge and light-sensing leads to a result in which reasonable confidence can be felt; but a more definite indication of correct alignment would be comforting.

At this point it occurred to me that since aluminium alloy is conductive; and since an anodising layer is thin and non-conductive; then the PCIS optics engaging rings, when exactly aligned, were actually flat metal plates separated by a thin insulator – i.e. capacitors. When incorrectly aligned the space between the conducting plates would become wedge-shaped, and the similarity to a capacitor – and perhaps therefore the measurable capacitance across the components – would diminish rapidly.

I went to the electronics workshop with this idea, and was met with something less like enthusiasm than scepticism. The alternative suggestion was made of using a megger (effectively an ohm meter for very high resistance values). I rejected this suggestion on two grounds:
1. Direct conduction being dependent on direct contact, I thought a megger would be less likely than a capacitance meter to respond to the difference between a small gap and a very small gap; and

2. Meggers often rely on high voltages to detect and measure high resistances, and I wasn’t comfortable with high voltages around the ultra-sensitive electronics in the LN-CCD.

Instead I suggested we get out the capacitance meter and try it, using two small scraps of anodised aluminium alloy sheet (sitting on the bench in front of us), either in direct contact or separated by a folded cigarette paper (which I happen to carry, and happen to know to be 0.05 to 0.06 mm thick when folded). The capacitance meter showed about a 2:1 difference between the two readings. (The absolute values are irrelevant and I do not recall them.)

I borrowed the capacitance meter and took it to the PCIS. There it was connected variously between the source turret and the optics; between the optics and the shutter housing; or across the two gaps, from the source turret to the shutter housing. Using the feeler gauge and optical gap-sensing to achieve an approximate alignment, the capacitance meter was watched for peaks in its readings while fine adjustments were performed. Eventually, a peak in the capacitance readings across both gaps was achieved simultaneously, and I accepted that an optimal alignment had now been reached.

The Phase 1 PCIS was now assembled, aligned and ready for testing. Most of the major components of the Phase 1 PCIS can be seen in the photo below (Figure 2.8.6).

![Figure 2.8.6 The complete Phase 1 PCIS. (The camera controller and the computers are not visible.)](image-url)
3 Operation of the Phase 1 PCIS

"The wonderful thing about a dancing bear is not how well he dances, but that he dances at all."

Anonymous.

The actual implementation of the PCIS was not a clean two-phase process as this document implies. Each time the PCIS was assembled and run, some part had been changed since the previous run, and the true picture is a more complex one of emerging detail and incremental functionality. So there is no single identifiable version of the PCIS that can be described as the Phase 1 implementation.

Nevertheless, the two phases exist. They are characterised both by different sets of control and synchronisation hardware, but also by the different goals of the two phases.

The Phase 1 implementation of the PCIS had as its primary goals the proving of the opto-mechanical components in conjunction with the LN-CCD detector (i.e. showing that it was possible to stimulate, record and measure a high-quality image from a stimulated mineral luminescence signal); and the determination of the best synchronisation and control architecture for the final (Phase 2) PCIS. The Phase 1 hardware is distinguished by including the “lunch-box electronics” (described below), which provide LED drives (blue and UV only), and very basic synchronisation support.

[The goal of the Phase 2 PCIS development, which is described in the later sections, was to deliver the final, fully automated and highly versatile PCIS, as per the architecture and design determined during the Phase 1 testing. The Phase 2 hardware is distinguished by including the Interface and Integration Control Unit (or “IICU” — an inelegant and supposedly temporary name which unfortunately stuck), for LED drive and system control functions. The IICU is described in Chapter 4.]

The temporary electronics built to meet the Phase 1 PCIS implementation goals are described below.

3.1 The Phase 1 PCIS Control Electronics

The design constraint for the Phase 1 electronics was to meet the necessary testing and development requirements for an absolute minimum of effort and resources. All of the resources left available to the project (and then some) would be required for the development of an effective final suite of electronics and software to provide full integration and automation of the PCIS.

For this reason the interim electronics were completely designed and built by myself, using a mixture of scrounged and scavenged parts. Since even this level of development would test what I remembered of my schoolboy electronics to the extreme, I should gratefully acknowledge the enormous assistance provided by “The Art of Electronics” (Horowitz and Hill, Second edition) and, where either that book or my understanding of it was insufficient, the generous assistance of the staff of the Electronics Workshop at the RSES. Any remaining errors are, of course, my own!
3.1.1 The Basic LED Driver

The first and most critical requirement for Phase 1 PCIS operations was to be able to drive at least the UV and blue LEDs. (The UV LEDs would allow focusing of the lens with a standard quartz OSL filter pack in place, and the blue LEDs would allow the optical stimulation of a UV signal from quartz. Thus both the optical performance and the sensitivity of the PCIS could be proved.)

According to the principle of minimum effort, only two LED drives were supported. Each comprises a dry cell battery (9 Volt or 18 Volt), and an appropriate 10-turn potentiometer to act as a “total loss” current limiter. A single moving coil meter (scavenged – who wants moving coil meters?) can be switched between the two LED drives to measure and set the current, after which it is assumed not to drift during the course of the experiment.

A bit rough perhaps, but very simple and cheap.

A small transistorised circuit allows each drive to be switched on and off by an external TTL-level signal. A front-panel switch allows the selection of either this “externally controlled” mode, or a “manual mode”. In the manual mode, each LED drive may be controlled via a simple front panel “on-off” switch. The manual mode would be used for focussing the optics and certain basic tests; the “externally controlled” mode would be used for timed exposures in OSL experiments.

This implementation is shown below at Figure 3.1.1. The obviously new components came from personal stock, or were scrounged from the RSES Electronics workshop. Since lunch boxes are made with sloping sides, and the unit kept falling over, half clothes pegs were used as feet to compensate. The “current monitor” points allow the attachment of separate external (and hopefully more accurate) meters.

![Image of the LED driver](image-url)

Figure 3.1.1 The interim LED driver electronics for the Phase 1 PCIS implementation.
3.1.2 The LED Driver Digital Upgrade

The inaccuracy of the meter, the lack of separate metering of each drive and a serendipitous two-for-the-price-of-one special at a well-known auto parts supplier encouraged the upgrade of the interim LED driver electronics to the Digitally Enhanced version shown below (Figure 3.1.2).

![Digital Upgrade to Interim LED Driver Electronics](image)

Figure 3.1.2 The digital upgrade to the interim LED driver electronics.

3.1.3 The Binary Timing Device

A timing device was designed and built to control either string of LEDs via the "external control" input(s) of the interim LED driver unit. There were three reasons for constructing such a device:

1. to provide tighter timing control for reproducible focussing and UV-emission luminescence image captures;
2. to provide a further level of testing of the synchronisation of the PCIS functions from the Minisys relay outputs; and
3. to test the adequacy of the timing method for use in the final implementation.

In an interesting throwback to my childhood, the timing circuit was built around an NE555 LSI Integrated Circuit – the first IC I ever bought and used. In my early teens I built a photographic exposure timer for darkroom use with the very first NE555s offered for sale in Australia. Other than the change from bipolar to CMOS IC technology, and the use of binary toggle rather than rotary switches, the main difference between the two is that whereas the PCIS timer is in a plastic lunchbox, the old darkroom timer was built in a metal ice-cream tin\(^{11}\).

\(^{11}\) The overlap in human history between ice cream coming in metal tins and LSI integrated Circuits being available is quite narrow, and places the earlier development around 1972 or 1973.
The earlier timer (as I discovered when I unearthed it again recently) also had at least some front panel markings, which the interim PCIS LED timer (as can be seen from Figure 3.1.3 below) — again built largely from scavenged parts — sadly lacks.

![Image of the interim LED drive timer device.](image)

Figure 3.1.3 The interim LED drive timer device.

Most of the front panel switches seen are used to select a duration by binary addition, the switches representing 1 second, 2 seconds, 4 seconds etc, through to 256 seconds. The push buttons act as “start timer” and “cancel timer”, while the red LED indicates whether the timer is active. One switch turns the whole unit on and off, and the final switch stops dirt getting in through the hole.

The unit is designed to have the timer started either by the front panel switch, or by the receipt of a falling edge TTL signal on an input (via the chocolate strip terminal seen at the top of the unit). The primary output is a TTL signal that is held low as long as the timer circuit is “active”. This is designed to trigger the “external control” input of the interim LED drivers.

The digital panel that can be seen at the top of the unit is a sad relic of a failed attempt to upgrade the unit to a full independent time-and-current-controlled LED driver with a digital current read-out. This attempt was undertaken largely to determine if I could design the digital panel meters into a
final LED timer/driver circuit for the final PCIS. This goal it fully satisfied, providing a clear and unequivocal answer in the negative. My schoolboy knowledge of electronics – even supplemented by "the Art of Electronics" – was insufficient to decipher the floating earth requirements from the single page of documentation provided (in Chinese). Still, at least now I knew! The panel was left in place after its smoke got out, partly as a monument to and reminder of my limitations, but mainly because it was doing no harm (and was keeping the dust out) where it was.

Of more note were the experiments to determine the accuracy and reproducibility of the NE555 timer. This circuit works essentially by charging a selected capacitor from 1/3 of the supply voltage to 2/3 of the supply voltage through a selected resistance. If one assumes a perfect (zero-leakage) capacitor being charged over a very small voltage range, then these things follow:

- the current flowing into the capacitor is inversely proportional to the series resistance;
- the time taken to charge the capacitor is inversely proportional to the current flow; so
- the time taken to charge the capacitor is directly proportional to the series resistance.

So in theory the drive time for the LEDs can be constructed in a binary fashion by simply adding in a series resistance corresponding to the time represented by that binary digit – so that each switch simply shorts out the corresponding resistor when that time unit is not selected. One goal of the development was to determine whether this worked well enough in practice to use in the final PCIS. The fact that the NE555 avoids some device non-linearities by only using the central third of the available voltage range for timing, combined with the use of modern very-low-leakage tantalum capacitors, gave me hope that the approach might be adequate. These hopes were sadly dashed.

In testing with a high-accuracy commercial counter/timer, the performance of the NE555 timer showed both non-linearity with the addition of resistances, and variability with small temperature variations, amounting to in excess of 1 percent. Although adequate for many purposes – including all of the proposed operations of the Phase 1 PCIS – this would clearly not meet the requirements of the final PCIS, and a crystal-controlled clock circuit would presumably be required.

Despite the failures and limitations noted above, I now had a set of electronics that could provide drive for a source turret LED string, at a pre-determined current and for a pre-determined duration, either at the push of a button or on the closing of a Minisys output relay. This is the form in which Phase 1 PCIS operations were undertaken.

An additional small lunch box, with a section of old telephone distribution frame in it, provided a mechanism whereby different wirings and cross-connections between the Minisys I/O card and the various other components of the Phase 1 PCIS could be rapidly implemented and tested.

One further item of electronics was required to support camera system synchronisation in the Phase 1 PCIS, and testing of the integration architecture proposed for the final automated PCIS. That item is a versatile, general purpose TTL signal monitor and generator, and is described below.
3.1.4 The Versatile TTL Signal Monitor/Generator

It is certainly possible to buy a commercial TTL signal generator and monitor, but (a) I had no budget, and (b) commercial units do not distinguish between a “high” input and a floating, or high impedance input. TTL logic is universal: signals float high (or are “gently pulled” high by a bias resistor in a TTL input stage) to represent the “0” or “inactive” state; and are “pulled hard” low by a transistor (or a corresponding CMOS structure) to represent the “1” or “Active” state. That is why TTL is referred to as “Active Low”; and disconnected inputs are sensed as “inactive”.

Except that Risø in their wisdom decided to interpret the digital inputs and outputs on the optional I/O card in the opposite sense. (This also impacted the design of the final PCIS Control Unit – see Section 4.) There were also some TTL ports on the ST-138 camera controller that could be programmed either as outputs or as inputs. A standard TTL monitor may not be able to distinguish a port programmed as an output but in a High or “0” state, from the same port programmed as an input, and floating. Similarly, a standard device could not distinguish a wiring break (high impedance) from a floating input — and in the PCIS a floating input may be interpreted as “Active” or “Inactive” depending whether it was sourced from the Minisys or elsewhere.

I therefore felt it was warranted to spend the time and effort to design and construct a specialised TTL monitor that, on at least some of its inputs, could distinguish 3 separate states; High, Low and High Impedance (or “floating” – i.e. disconnected). The device built included 2 switched TTL Outputs; 2 “normal” TTL Inputs; and 2 “special” TTL Inputs capable of distinguishing 3 states; High, Low and Floating (or Hi-Z). The 2 special inputs each have 2 buffered outputs associated with them; one where a Hi-Z input is translated to a Low output; and one where a Hi-Z input produces a High output. In the photo below (Figure 3.1.4), Input 1 is Low, Input 2 is floating (Hi-Z), and Inputs 3 and 4 could be High or Hi-Z. Both of the switched TTL Outputs are set to High.

![Figure 3.1.4 The TTL Signal Generator and Monitor in use in PCIS Phase 1.](image-url)
It was already clear that the TTL Tester built for Phase 1 PCIS system synchronisation testing would also have a role later in testing of the final automation electronics. (It is indeed a key component of the IICU Firmware Development and Testing environment described in Section 4.) I therefore splurged on a proper ABS plastic electronics box for it, and even made a proper front panel. (The front panel is surfaced with clear contact, so that in different experimental setups the functions of the different ports can be written on with a white-board marker, and later erased.)

In the Phase 1 PCIS the TTL unit would be used for both camera synchronisation via the TTL signals provided by the ST-138 controller (using the “External Sync” mode in the WinView software), and for monitoring and control of the shutter via TTL ports on the shutter driver. It would also serve to monitor the proper control of the Minisys external I/O ports using the Minisys low-level control language over the serial interface (at this time using a facility of the Risø-supplied “Centre.exe” utility software to send individual low-level Minisys commands). Since these ports not only adopt the unusual logic convention referred to above, but are also variously numbered from 1 to 8 (or 16); or from 0 to 7 (or 15 – there being 8 outputs and 16 inputs), the possibilities for confusion were nearly endless, and the TTL Tester would provide a simple means of resolving any logic or numbering errors that might emerge.

The complete Phase 1 PCIS integration electronics can be seen in Figure 3.1.5 below. On the top shelf is the Minisys controller (which receives and interprets the Minisys low-level language commands via the RS-232 connection), with beside it the case which holds the I/O card (which cannot be mounted inside the controller). That case has the D25 and D35 connectors via which the Minisys digital Outputs and Inputs respectively are brought out. In Phase 1, these ran to the wiring cross-connect box, which is hidden behind the digitally-enhanced LED driver on the lower shelf. To the left of the LED driver are the LED timer (described above), and the commercially-supplied shutter controller. To the right of the LED driver is the TTL Tester, connected to the TTL ports on the back of the ST-138 camera controller.
3.2 Phase 1 PCIS – Optical and Performance Testing

Many of the results produced during Phase 1 PCIS operations were important in their own right, with some of them contributing to publications on the properties and behaviour of the materials analysed. These results – along with others from the final Phase 2 implementation – are presented below in Section 6. The present section is concerned only with the performance of the PCIS during these experiments, and the implications of that performance for the PCIS’s future development path, and largely ignores the nature of the samples used and the specific results obtained.

In order to interpret what these Phase 1 tests reveal about the performance of the PCIS, it is important to understand the information recorded and displayed by the WinView software when an image (called an “accumulation” in CCD-jargon) is captured. Figure 3.2.1 below shows an example of an image displayed under WinView. This example is an actual thermo-luminescence image of some large grains (order of 1 mm) of Australian lake salt after beta irradiation. There is no incident light illumination of the grains, and no aperture plate or optical filters installed in the source turret.

Figure 3.2.1 Example of an image display using the WinView software.
Along the top of the display can be seen the name of the image file (always a .spe extension) and a specification of the size and number of images in the file. In this case "(512 x 512 x 2)" indicates that the image is at the maximum resolution of 512 by 512 pixels (i.e. no binning has been employed); and that there are two images in the image file. Also on the top row are the standard Windows buttons to minimise, restore or close the window.

The main part of the window is of course the captured image itself, which has been brightened and contrast-enhanced to make it clear and easy to view. This is normally done using buttons on the bottom row – more on that later. The cursor, which may be placed anywhere within the image boundaries, can also be seen. The cursor position affects what is displayed elsewhere.

To the left of and below the image area two graphs can be seen. These graphs display the image brightness data (the pixel photon counts) for sections taken vertically and horizontally respectively, through the current cursor position in the image. The graph axes show the pixel photon counts between which the display grades from black to white – not usually the actual darkest and brightest pixels of the image, as is explained below. Nevertheless, in a given display mode, these figures provide the first – and a very good – estimate of the overall “brightness” of a sample.

Below the horizontal section graph is a row of seven buttons. The last five of these control zooming of the display in a fairly obvious fashion and need no further explanation. The first two, however, control how the image information is displayed, and need to be properly understood.

The second button (the one that looks like a half moon) invokes or restores the standard, default display mode, in which the screen tones from black to white are allocated to pixel counts ranging from the lowest to the highest recorded values in the image. (The upper and lower “intensity” bounds on the graphs are also set to the same values). In theory this allows the full dynamic range of the image to be viewed in a fashion that is normalised for the overall “brightness” of the sample, and should be the preferred mode for displaying the luminescence images. In practice it suffers from the fatal flaw that with a “dim” image (low pixel photon counts – normal in luminescence work), a single bright pixel – say due to a cosmic ray strike somewhere on the CCD – will cause the display range to be artificially extended at the high end. That in turn causes the low pixel counts associated with the “real” image data to be all displayed as black, and the image effectively disappears.

The solution lies with the button labelled “5/95%”. When this display mode is selected, the screen tones from black to white are apportioned between the 5th and 95th percentile of the pixel photon count values. All pixels with counts lower than the 5th percentile display as black; all pixels with counts greater than the 95th percentile display as white. Although this often means (as in the above image) that detail in the brightest parts of the image is lost (in photographic terms, the image is unacceptably “washed out”), the problem related to cosmic ray strikes is eliminated, and the 95th percentile value (displayed as the upper value on the graphs) provides an excellent first guide to the overall brightness of the sample and the image. This is important in determining the appropriateness of the camera and shine parameters to the protocol in use, with the goal of producing images with a good signal to noise ratio, but no saturation of the CCD surface. 95th percentile values in the hundreds or thousands are a good sign; values in the 10s or 10s of thousands are less ideal.

Pull-down menus can be used to explicitly set display range values other than those produced by the two button options, and indeed display modes other than the image display shown above (including
3-D graphing options) may be invoked. However, the standard image display and the 5/95% button have proved to be the best way to display, observe and compare PCIS image data, and is the standard display method used in this thesis. (There is a good argument to be made that in the PCIS application, a “5/99%” function would give a better display of the brighter parts of the sample, while still avoiding the cosmic ray issues. However, “5/95%” is what is available; it is perfectly adequate; and so it is what I have used.)

The final row of the display shows the Intensity (“I” – actually the photon count) of the pixel under the image cursor. The “X” and “Y” values specify which image pixel is under the cursor being examined; and the “Z” value specifies which image of a multi-image .spe file is being viewed. In this case the intensity of the selected pixel of one of the bright salt grains is shown to be 14,678 counts – almost double the 95th percentile value for this image of 7,691. This shows both that (a) information about the brighter parts of the image is not well displayed with the “5/95%” option; and (b) that, as with many samples, a small number of grains are a lot brighter than the average; and that, using a traditional PM tube, a few bright grains might dominate the data for a whole disk. This observation would appear to put in question the statistical assumptions, based on the use of a large number of ‘similar’ grains, which underlie some of the luminescence protocols used in dating applications.

### 3.2.1 Image Quality Determination

The image quality with a good bright sample (as seen in Figure 3.2.1 above), or with adequate incident illumination (as will be seen in examples that follow) is perfectly adequate to determine the grain (or, with large grains, the region of a grain) from which the counted photons originated. The resolution of the optics at the centre of the image, when correctly focussed, is a good match to that of the 512-by-512 pixel detector (see Figure 3.2.2 below). The image quality at the corners of the image area degrades in a fashion very much in accord with the computer-generated predictions reproduced in Chapter 2 and in the appendices. With a bright sample and a small aperture plate inserted in the filter drawer, the corner image quality more closely approaches that of the centre of the image (see Figure 3.2.3 below).

The reaction of the designer of the optics, on being shown samples of the images produced, was that the performance of the optics was “as good or better than could have been expected.” (Damien Jones, pers. comm.) The second attempt at the alignment process had clearly been a success, and the optics were performing at or beyond specification.

Nevertheless, two apparent imperfections remain. The image is not fully centred in the detector; and the focus plane is not perfectly parallel to the plane of the sample disk. The latter effect is very visible in Figure 3.2.4 below. Note that the total depth of field of the optics, without inserted aperture plates, is only of the order of 0.1 to 0.2 mm. Any mismatch between the focal plane and the sample plane is therefore immediately apparent, and indeed appears exaggerated. (This image was actually taken with the Phase 2 PCIS; but the optics are the same, and the effect is particularly clear here. It shows the 180 – 212 micron fraction produced from supermarket-bought Saxa Rock Salt.)

I discussed these issues with the optical designer, with a possible view to trying to adjust the optical alignment to remove or ameliorate the effects.
Figure 3.2.2 Incident light image demonstrating moderate light, full aperture performance of the optics.

Figure 3.2.3 Incident light image showing performance of the optics with good light and reduced aperture.
The opinion of the optical designer was that the apparent defects were minor, and may well not be defects of the lens at all, but might rather be due to inaccuracies in the position of the sample holder (the heater plate) within the Minisys itself. This seemed a reasonable suggestion, since:

1. The precise alignment of the sample plate to the photomultiplier tube would not be a critical design goal of the Minisys, as any errors on the scale referred to would not make a measurable difference to the signal measured by the PM tube;
2. The mechanical design of the Minisys is such as to ensure precise control only of the height of the sample—the only factor that would make a difference to the received signal; and
3. I know of at least one occasion when another researcher placed an over-thickness sample into the Minisys (being run in its original PM tube-based form), leading to the sample being scraped off as the sample platter rotated. The same sample may well have reached the vacuum window when the heater plate was raised, transferring force to the (mechanically weak) heater plate. This would have distorted the heater plate in a fashion that could cause the sample to be both off-centre and non-horizontal.

In situ measurement of the position of the sample, with sufficient precision to determine whether it or the optics are responsible for the apparent defects, would be impractical. In any case, wherever the problem (if indeed it is such) lies, it is a moot point whether it would be best addressed by a repeat alignment of the optics, or by a judicious tweak applied to the heater plate.

On balance, the optical designer advised against making any further adjustments to an alignment that was already meeting or exceeding the design expectations. (His actual words were “Don’t you dare touch it!”, Damien Jones, v. pers. comm.) I elected to follow this sage advice, rather than chase a probably unachievable perfection of the optical alignment.
3.2.2 Dark Testing

Dark testing is intended to ensure that there is no false signal being detected as a result of leakage of external or room lighting into the optical path. It is based on comparison of “dark”, or background noise-only images such as the one shown below (Figure 3.2.5). The important detail here is the values applied to the 5th and 95th percentiles for graphing, as these are directly dependent on the overall brightness (or, more precisely, total photon count) of the image.

[It should be noted here that most of the “noise” seen in the image below is not noise in the sense of being random – of predictable overall intensity but entirely unpredictable in detail. Rather, most of what is seen in the image is an entirely reproducible signal characteristic of the individual CCD and its operating temperature. As such it may be subtracted from captured images (a subject discussed later with respect to optimising the sensitivity performance of the PCIS), leaving a “true” noise level of only a few counts per read-out (i.e. per pixel when binning is not used). In a testing situation it is better to work with “raw” data, as processing of the signal may hide defects of the equipment. In this case there would be no benefit in employing background subtraction, and so it was not used.]

Figure 3.2.5 Standard reference background or “dark” image.

Dark testing comprised first obtaining an absolute minimum-level background noise image; and then comparing it with further images taken under increasingly “difficult” test conditions. The object of this testing was to ensure that the entire PCIS was free from internal or external light leaks that would interfere with data gathered under any “normal” operating conditions.
It should also be noted that the “dark counts” in the above image are higher than those now produced by the camera system when thermally stable. This is due to the camera having been accidentally given an excessive light exposure while switched on during system testing. The extreme saturation this caused persisted, but gradually diminished over a period of several months. The excess signal was concentrated at the left edge of the image, reducing rapidly as the Y-offset increased, but initially affecting the entire sensor area. By the time these dark tests were undertaken the excess noise (which has now entirely disappeared) had reduced to a narrow strip down the left edge of the image (still visible in the image above), and a slight increase of the noise counts across the rest of the image. Now that repeated use has completely cleared the excess charge from the CCD, the fully-temperature stabilised camera system produces about 10 counts per pixel less noise than in these images. Nevertheless, the excess noise is again highly reproducible, and the ability of the camera system to detect extremely low levels of excess incident light is not affected.

The “minimum noise” or “minimum signal” background image to be used for comparison was obtained by operating the camera for a 1/10 second exposure, with no stimulation LEDs active; with the shutter closed; and with ALL room lighting extinguished. (A few equipment LEDs remain illuminated, but the overall room light conditions achieved are very low – and this, as a caver, I can say with confidence.)

Dark testing was then undertaken by successively:

- Opening the shutter;
- Increasing the accumulation time to 120 seconds; and
- Increasing the room lighting to its normal level.

All of these tests produced images indistinguishable from the reference image (except for increased cosmic ray strikes with increased duration). The adequate external light-proofing of the PCIS optical path in normal machine-room operating conditions, including the adequacy of the filter drawer front light baffle, was thereby clearly established.

“Torture testing” was then undertaken by first securing all light-sensitive items in the room, and then increasing the room lighting to successive levels above the conditions in which the PCIS was designed to operate. This was undertaken largely out of interest, but also to prove that the PCIS external light proofing was not just adequate, but was in fact ample.

Initial torture testing was performed by placing a desk lamp (with reflector) on a high shelf, pointed towards the ceiling, with a 60 Watt tungsten filament globe. This provides illumination somewhat short of recommended office lighting levels; but a perfectly workable light level for non-dark adapted eyes, and quite disturbingly bright in a room full of light sensitive items that is normally kept at a just-workable gloom. Although not measured, the increase in the number of photons striking the equipment is estimated to be at least 4, and probably 6 or more orders of magnitude.

This test showed no increase in the photon counts of the accumulated image, proving that the external light-proofing of the PCIS was significantly (several orders of magnitude) better than was required for normal operation.
In “super-torture” testing, the 60 Watt globe was directed at various parts of the PCIS from the closest range that was physically possible – between a few centimetres and a few inches. This test – representing an estimated further 4 to 6 orders of magnitude increase in illumination – finally showed a significant, and clearly visible, light leak into the optical path. Unsurprisingly, the leaked light showed no distinct form or pattern on the face of the CCD. Typically it showed as a general raised photon count, with a moderate tendency to peak along one edge, or at the centre of the CCD. Figure 3.2.6 below shows a typical image produced by the “super-torture” testing (displayed in the usual fashion, with “5-95%” display setting). Figure 3.2.7 shows the same accumulation, but with the display bound parameters set to the “5-95 %” levels appropriate to the reference dark image. This is how it actually appeared in the lab during the testing, since the run-time display application in Winview does not automatically re-set the display parameters for each successive accumulation.

Repeated attempts to localise the light leak – by re-directing the lamp, applying black tape to mating areas, or wrapping the source turret area in black cloths – produced no decipherable evidence helping to locate the leak. The most likely candidates remain the mating surfaces between the optics and the two turrets, or the filter drawer flange. The mating of the optics to the turrets was later improved by manufacturing and installing the new camera stand, and by applying the improved PCIS alignment procedure (described earlier). The filter drawer flange seal could possibly (though not definitely) be improved by installing the 1 mm O-ring material which was part of the original design, but which – mainly as a result of the successful dark tests described here – has never been installed.

Preparing the room for the extreme light levels involved in the “super-torture” testing takes some effort, and it has not been considered worthwhile or necessary to repeat any of the extreme light level tests described here with the improved camera stand and PCIS alignment.
Note that in Figure 3.2.6 above the light leakage is only at the level of 10 to 20 photons per pixel for the 120 second accumulation. Although this was not necessarily the brightest image obtained (not all of the many accumulations observed during this testing were saved or documented), it remains true that the light leaks observed even under this extreme regimen were such that any reasonably bright signal would have rendered them (a) invisible, and (b) of negligible significance. Also observe at least two obvious cosmic ray strikes in the upper third of the image. These are indicative of the longer exposure time (120 seconds) used to obtain the dark testing comparison images.

Finally, the vertical line in the lower half of the image is a defect of the CCD, and turns out to be of a much lower magnitude and importance than is suggested by these very low count images with extreme contrast enhancement applied. It has not been noticed in later runs of the PCIS, and may have been due to a dust particle that has since moved during re-vacuuming of the CCD chamber.

![Figure 3.2.7 Raised background noise image as it appeared during testing.](image)

If Figure 3.2.7 above is compared with the reference dark image shown at Figure 3.2.5, several things may be observed. Firstly, however low the leakage may be in terms of absolute photon numbers or photons per pixel, there is no possibility of overlooking it in the lab when the display is set for the photon count levels of a normal “dark” image. It is very bright, and extremely obvious.

Secondly, the CCD defect, although still visible, has already become less obvious just with the brighter display settings; and finally, the cosmic ray strikes have become completely invisible. There is a lesson here; the visual extraction and interpretation of the information contained in any low photon count image data is critically dependent on the display parameters selected.
Note that the fact that it was eventually possible to produce the type of light leakage shown in Figures 3.2.6 and 3.2.7 above does not suggest any problem with the PCIS. Indeed, the fact that it took light levels at least 8 to 12 orders of magnitude greater than the expected (and design) operating conditions of the PCIS to produce light leakages – and those only at the 10 to 20 counts per pixel level – indicates precisely the opposite. The PCIS, including the radical removable filter drawer design, is extremely proof against external light leakage.

### 3.2.3 Zero-Point Signal Testing

Zero-point signal testing is designed to ensure that with optical filtering appropriate to the (thermal or optical) stimulation being used, but in the absence of a sample which is emitting a signal, no “apparent signal” or increased photon count is detected – i.e. that the image produced is indistinguishable from the reference dark image used earlier in dark testing.

An “apparent” or “ghost” signal would be represented by an image that, with whatever brightness and contrast settings are required, shows some sort of “picture” corresponding to, or reminiscent of, the sample or internal Minisys hardware (sample holder or heater plate). This would be (and in certain tests was) indicative of the optical filter material being unable to exclude all of the reflected stimulation light on the optical path. Generally, this indicates that an incorrect filter type (as represented by its cut-off wavelength(s)) is being used or, more commonly, that an insufficient thickness of filter has been placed in the filter drawer.\(^{12}\)

An increased photon count (without any visible “ghost image”) would be indicative of stimulation photons off the optical path that somehow re-enter the optical path so as to strike the CCD, but at a location unrelated to their point of origin at the sample. (Any photons that start on the optical path should remain on it, and so – if not totally excluded – would contribute to a ghost image rather than to a raised background count. The number of photons starting on the optical path that get reflected by filter surfaces or dust particles, and somehow later re-enter the optical path in such a manner as to contribute to the background “noise” count, will be so small as to be ignored.)

An increased background count is therefore indicative of off-path photons leaking past the baffles and obstructions designed to prevent them doing so, and reaching the CCD. Excluding photons that survive a high number of reflections from the – mainly black anodised – internal surfaces of the optics assembly itself, there are only two such paths available:

1. by escaping the confines of the PCIS before the filter pack and re-entering it beyond; or
2. leaking around the edges of either the filters, or of the filter drawer containing them.

\(^{12}\) In one particular case, as documented earlier, it indicated that the blue LEDs had a short wavelength tail to their emission spectrum that crossed into the designed pass region of the UG11 or U340 filters generally used for quartz blue/UV OSL. This required the application of a UV-excluding filter to the front of the blue LEDs. Similar problems have not been encountered using UV LEDs for stimulation. Red LEDs have yet been employed for stimulation, but problems would appear unlikely unless a detection window especially close to the emission wavelength were selected. The UV LEDs are not designed for use as an OSL stimulation source, but rather to allow focussing images to be acquired when the filters for blue-stimulated UV OSL are in place.
Since the dark tests described earlier show that photons do not enter the PCIS, photons cannot be leaving and re-entering it. Therefore any increased background count during testing would indicate a leak around the filter drawer, and invalidate the stepped-baffle design.

Happily, I can report that no increased background photon count was detected in any test with appropriate filters of sufficient thickness to avoid ghost imaging. The earlier decision to incorporate a removable filter drawer, and the particular design employing simple stepped baffling around the front and base of the filter drawer, with rectangular drop-in filters and stepped adapter inserts, is totally vindicated.

3.2.3.1 Ghost Signal Zero Point Testing

It was determined both using TL runs to 300 °C, and using IR LED illumination of various fully bleached samples, that 3 mm of BG39 filter material was sufficient to fully exclude leakage of photons from IR stimulation LEDs, or of thermal photons associated with moderate temperature thermo-luminescence.

It was also determined that 6 mm of UG11 (with a red-excluding notch-coating on the lower surface) was sufficient to fully exclude photons from the blue LEDs in blue-stimulated OSL accumulations up to 60 seconds. Reducing the thickness to 5 mm produced “ghost images” at count levels similar to or below those of the background noise – peaking at perhaps 20 counts or so per pixel. (This was for a 60 second accumulation. The “ghost signal”, unlike the background read-out noise which has been shown to be constant, will be proportional to the accumulation time.)

Since the signal-photon loss due to an extra millimetre of filter is appreciable; and since the typical OSL accumulation (with later experience) turns out to be around 10 seconds, the choice of 5 mm or 6 mm of filter for optimised quartz OSL runs remains open. A conservative 6 mm of filter avoids any false signal for any reasonable length of OSL accumulation, and was used for all subsequent testing with blue LED stimulation. Using only 5 mm of UG11 filter may yield a larger percentage increase in the passed signal than it does of leaked stimulation photons, and so yield a better overall signal-to-noise ratio (and therefore better sensitivity) than using 6 mm of filter. This is one of a number of factors that require further investigation when performance tuning of the PCIS is undertaken.

3.2.4 Sensitivity Testing

Sensitivity testing is performed to ensure that the combination of the large capture angle reflective optics and the high quantum efficiency LN-cooled CCD are sufficient to allow viewable and measurable images to be formed solely from the photons emitted from a sample as luminescence.

Clearly some materials emit more luminescence photons than others; and the photons emitted from a given sample will depend not only on the experimental setup and the stimulation used, but also on the radiation dose received by the sample (whether natural or applied) prior to the experiment.
Since a natural dose in a sample can only be ‘shone out’ once, and since machine testing generally requires repeated runs, with different experimental parameters, to properly characterise the machine’s performance, most of these tests were conducted with beta doses applied in the lab. However, the whole principle of luminescence dating is that a rapidly applied artificial beta dose is indistinguishable from a slowly applied natural dose; so that testing with ‘artificial’ radiation doses is a perfectly adequate way of testing the sensitivity and performance of the PCIS.

In any testing program for developed technology, one applies ‘easier’ tests first, moving on to ‘harder’ tests, which will more fully exercise the machine’s capabilities, when the earlier tests are successfully concluded. Accordingly these tests start with a CaF₂ slice and then move on to more difficult ‘natural’ materials; salt (NaCl), porcelain, glass and finally quartz.

For similar reasons – ease of testing – most of these runs were TL, not OSL. Hand-operating the PCIS – separately controlling the Minisys, the camera system, the shutter and the stimulation sources – is quite difficult for TL runs; to do so for any meaningful OSL protocol, repeatedly and without errors, is very near impossible. This fundamentally is the reason that PCIS development was not terminated, and this thesis written up, at the end of the Phase 1 development. The Phase 1 PCIS, as will be shown, works; but it is not a useful or usable research tool. More extensive tests employing the OSL SAR protocol were necessarily deferred until the fully automated Phase 2 PCIS was complete.

Note that while the tests on salt, porcelain and glass were selected because the results, if successful, would also be valuable in their own right for what they disclosed about the behaviour of the material, these tests were conducted primarily to establish the performance of the PCIS. The particular tests were selected to suit the research objectives of other scientists uninvolved with the PCIS development, and have been the subject of publications by those researchers. These results and the publications to which PCIS data have contributed are discussed later under “PCIS – Results to date”. In the present section only the performance of the PCIS is discussed; the meaning of the results in terms of the materials studied is not considered.

3.2.4.1 Sensitivity Testing - Calcium Fluoride (TL)

Since Calcium Fluoride glows so brightly and reliably when heated even moderately, it was felt that imaging a homogenous 10 mm diameter, 1.0 mm thick CaF₂ disk would provide an ‘easy’ first test whereby the basic performance and sensitivity of the PCIS could be established. The CaF₂ disk could be re-irradiated between experiments, and a firm, easily measurable performance baseline could be established.

Unfortunately for the particular tests, but reassuringly in terms of PCIS performance, the sample was simply too bright! As shown in Figure 3.2.8 below, a first shine out (of what we shall call the ‘natural’ signal) caused immediate (but non-damaging) saturation of the CCD charge wells.

In Figure 3.2.9 below, we see the second accumulation taken from the same sample without any added beta dose. Two points are worth noting. Firstly, the material – although machined to a uniform 1 mm thickness – is not homogenous in its luminescence response. It clearly contains bright regions, which appear to be associated with the centres of saturation in the previous image.
Secondly, the heater plate being visible through the material is almost certainly due to luminescence photons reflected back from it, rather than differential heating of that part of the CaF₂ disk. The greater concentration of bright pixels in the area of the disk directly over the heater plate, however, probably is due to differential heating. Note that the parts of the heater plate visible outside the disk area are being imaged via reflected luminescence photons emanating from the edge of the CaF₂ disk.

Figure 3.2.8 TL glow of the “Natural” signal from CaF₂, showing severe CCD saturation.

Figure 3.2.9 Second “Natural” TL glow of the CaF₂ – a clear image with no CCD saturation.
A further accumulation of luminescence from a CaF$_2$ disk following an artificial beta dose produced such great saturation of the CCD that further experiments with CaF$_2$ were abandoned. The material is clearly far too bright to be of any use as a brightness reference, let alone as a meaningful test of the sensitivity of the PCIS.

### 3.2.4.2 Sensitivity Testing – Table Salt (TL)

Tests using salt were conducted both to test the PCIS performance, and to continue work conducted by Dr Spooner over recent years investigating the different luminescence signals from salt. Salt is ubiquitous in the human environment, and the ability to use it as a retrospective dosimeter would be of great value. The feasibility of such use depends on separating and characterising the different luminescence signals produced, and the PCIS, with its extreme spectral range, provides a means of examining longer wavelength signals, outside the spectral range of photomultiplier-based devices.

There is also a great benefit – especially to someone who has dived in zero visibility in constricted cave passages to collect sediment samples – in being able to purchase your sample material, by the 500g bag, at your local supermarket. (For some unknown reason it is possible to purchase a wide variety of table salts, of different marine, lacustrine and terrestrial origins, from this everyday retail outlet. This is odd, since – apart from grain size – it requires a machine like the PCIS to distinguish between them, and so few people actually have one. Nevertheless, for me it was most convenient.)

![Figure 3.2.10 TL of large grains of lake salt, 160 °C - 260 °C after 200 s (~20 Gy) Sr$^{90}$/$Y^{90}$ beta dose.](image)
Figure 3.2.10 above shows the collected luminescence image from the second half of a TL run on large (order of 1 mm) grains of Australian lake salt. (The first half of the run, representing the 160 °C to 260 °C temperature range, appears earlier as Figure 3.2.1.) Examination of these images shows that for each part of the TL run, the dimmer grains are producing luminescence signals in the thousands of counts per pixel; and the brighter grains, counts in excess of 10,000 counts per pixel. This represents luminescence counts of the order of millions of photons per grain per Gy.

[Note that in Figure 3.2.10 the heater plate has just become visible by the thermal IR photons it is emitting. This suggests using a short-pass IR-excluding optical filter would be beneficial for this sort of TL study, although the photon counts associated with this effect are too low to significantly affect the total photon counts in this temperature range. It is also interesting to note that the sample disk appears to be lagging behind the temperature of the heater plate. The extent of this effect could be studied with the PCIS, using different styles and materials of sample disks, to derive information about the likely true sample temperature for different heater plate temperatures and heating rates.]

The conclusion to be drawn at this stage of testing is that the PCIS clearly has the required sensitivity, as well as the resolution, spectral range and spectral discrimination, to successfully pursue this line of study into table salt, and into its potential as a retrospective dosimeter.

### 3.2.4.3 Sensitivity Testing – Porcelain (TL)

During one of the early PCIS testing runs, I performed TL on a slice of porcelain on behalf of Dr Spooner. The interest was in inspecting how uniform or otherwise the luminescence was across the sample of porcelain – a classic PCIS application – as part of a study into the centres of origin of porcelain luminescence. For the PCIS, it provided another material to test against, as well as providing an excellent example of the usefulness of an imaging luminescence reader.

A number of TL runs were performed with very similar results. The image shown at Figure 3.2.11 was chosen mainly because I like the enormous scatter of cosmic ray strikes that are visible. The sun was very busy that day; and clearly on a bad day cosmic ray strikes can be serious enough to affect the overall data, and techniques for “removing” cosmic ray strikes certainly need to be examined as part of the performance tuning of the PCIS. (Alternatively, a suitable location, under several metres of concrete or similar shielding, should be found for the PCIS.)

The TL image at Figure 3.2.11 shows per-pixel counts in the range of 100 to 200, giving a total photon count for the sample of close to 10 million photons (by measurement using the WinView “statistics” utility). It is also clear that the image gives good information about the distribution of luminescence centres in the porcelain sample. The PCIS is proven to have the sensitivity – as well as the spatial resolution – to enable it to undertake useful studies of this material. These results are reported in (Oks et al, 2011).

One further item of interest may be noted in these images. Just to the left of the porcelain slice, a small area (~ 15 by 20 pixels) of brightness is visible. This region – representing a total of 3,000 to 4,000 counts above background – is visible in all the TL images, but is not visible in the focussing image shown at Figure 3.2.12. (Unfortunately it is in the region of shadow due to the single UV LED
installed in the Phase 1 PCIS.) The nature of this signal, and the material producing it, is unknown; but it suggests that careful cleaning and preparation of disks between runs is of great importance.

Figure 3.2.11 Porcelain slice TL, room temp to 160 °C at 2 K/s, after 1 s (~0.1 Gy) beta; no optical filters.

Figure 3.2.12 Incident light focussing image of the porcelain slice.
3.2.4.4 Sensitivity Testing – Quartz (Blue/UV OSL)

Quartz can be a difficult material to study, and often tests the sensitivity limits even of a traditional photomultiplier tube-based luminescence reader – especially when working with a single grain, rather than a thousand or more on a masked mono-layer disk. Apportioning this scant signal across hundreds of separate detectors (i.e. pixels) would provide an extreme test for any detector system, even without the signal loss due to the inherently smaller capture angle of imaging optics, compared to placing a single detector as close as possible to the sample (as in the traditional PM-tube method).

It is important to understand that in this sense every PCIS run behaves like a single-grain run. When there is a larger number of grains on the sample disk (whether as a masked mono-layer, or as a separated grain array on one of the special disks), the now larger total light sum is apportioned over a proportionately larger number of detectors (pixels). Since each pixel is approximately 25 µ square, and the imaging is 1:1, the photons from a 400 µ diameter grain will illuminate around 200 pixels. A 200 µ grain will illuminate around 50 pixels, since the apparent area of the grain is reduced by a factor of four. But the true situation for the smaller grain may actually be worse than for the larger grain, since the volume of the grain – and so, presumably, the total emitted photon count – is reduced by a factor of eight. Depending on the efficiency with which photons from deep in a sample can escape the grain, small grains may be expected to appear, both at peak and on average, dimmer than larger grains. (If only near-surface photons escape to be detected, the effect will not exist. This is an open question, which the PCIS will in future be able to answer simply and definitively.)

Whatever grain size is chosen for study, the PCIS would likely require optimisation of its operating parameters to achieve the sensitivity and performance necessary to derive meaningful palaeodose estimates for individual quartz grains. It would also need the programming and automation facilities of the Phase 2 PCIS to perform accurately and reproducibly the number of individual measurements required for a normal Single Aliquot Regeneration (SAR) protocol run on even a single sample disk.

Nevertheless, it was desirable to undertake certain basic tests with the un-optimised, manually-driven Phase 1 PCIS to at least establish that it was possible to detect and measure a photon count associated with the stimulated emission of luminescence from quartz – whether due to the quartz’s “natural” received dose, or to an artificial beta dose applied in the lab. Given the difficulties of operating the Phase 1 PCIS in this manner, I took the easy way out with samples, and used some material that was readily available in the lab. This was a single laser-Minisys disk (indented to take a 10 by 10 array of 200 µ grains), with a scattering of 200 µ-fraction grains of an Australian quartz sample known only by its identifier, K0852.\(^{13}\)

The only incident light focussing image I have retained of this disk appears below as Figure 3.2.13. Note that due to the lower photon counts in this image, it does not appear as sharp or as high contrast as the similar image seen earlier at Figure 3.2.3, even though there is still a 44 mm diameter aperture plate in the optic path. This image was captured using a 3 second accumulation, with 2.5 mA of current in the (then single) UV LED, through the filters used for the OSL.

\(^{13}\) Unfortunately a number of laboratory notebooks were thrown away (not by me, but by a member of staff responsible for clearing other labs) when the building in which they were located was being demolished. It is now not known whether any records of the origins and preparation treatment of this sample still exist.
Figure 3.2.13 Incident light image of the quartz grains used for Phase 1 sensitivity testing.

The double image below (Figure 3.2.14) shows the first and second shine-outs respectively of the (unknown) natural signal in the quartz grains. (The quartz luminescence images in this section are easier to observe with the "false colour" option display turned on, and so are shown that way.)

Figure 3.2.14 First and second shines of the natural OSL – bright grain under cursor.
At least three of the grains are clearly visible by their own, natural, luminescence signal. This – given that there has been no tuning or optimisation of the PCIS whatsoever for this work – is an extremely positive result.

Conditions for these shines – and the further shines shown below – were:

- 10 second pre-heat to 220 °C;
- sample held at 125 °C during each shine;
- stimulation provided by four single chip 470 nm blue LEDs running at 25 mA for 64 seconds;
- using 44 mm diameter by 6 mm thick UG11 filter (actually two 3 mm slices) with a red-excluding notch coating on the lower surface.

The images appearing at Figure 3.2.15 below were acquired under exactly the same conditions, but following an artificial beta dose of 180 seconds (approximately 18 Gy). The same three grains are again visible, suggesting that a comparison of the luminosity between the natural and regen doses – at least for these few grains – should be possible.

These images were displayed with the same pixel under the cursor, so that their values could be compared. Although the first shine photon count is higher for the regen dose than for the natural (39 counts vs. 35), the difference between the first and second shine values is almost identical in each case (8 counts for the natural; 9 counts for the regen dose). This suggests (no more) a palaeodose estimate for that grain somewhere in the realm of 15 to 20 Gy.

A better estimate of the overall signal available to work with would be had by summing all the pixels corresponding to the grain; this is what is done, for the same images, in the following two figures.
Figure 3.2.16 Light sums for one grain, natural blue-stimulated OSL, first and second shines.

Figure 3.2.17 Light sums for one grain, OSL after 180 s (~ 18 Gy) beta, first and second shines.
In Figure 3.2.16, a box has been drawn around the grain highlighted under the cursor in the earlier figures. The Winview "Statistics" utility has then been used to produce a total photon count for the selected area. (Note that exactly the same area and cursor position are used in all of these figures.) These statistics show that the first shine of the natural OSL signal produced 756 more counts than the second shine did (17,437 vs. 16,681).

Figure 3.2.17 shows the same information relating to the two shines following an artificial beta dose of 180 seconds (~18 Gy). In this case the difference between the first and second shine photon counts for the grain is exactly 1000 (17,666 vs. 16,666). This is consistent with the single-photon data from Figures 3.2.14 and 3.2.15 above, but suggests a slightly lower palaeodose (perhaps 12 to 15 Gy) than the estimate based on the single pixel comparison.

[Note that normalisation or, in the jargon, "test" doses of 60 seconds (~6 Gy) were applied and shone out – in the same manner – before each regen dose was applied. However, no attempt has been made to apply the test dose data to "suggested" palaeodose estimates above. Such corrections would be made in specialist palaeodose estimation software, implementing some complex algorithm for determining a palaeodose estimate from the totality of the natural, regen and test dose data.]

Optimisation of the PCIS for quartz blue/UV OSL is discussed in the Chapter, "Future Development and Applications of the PCIS", and comprises a number of factors that can be set, individually or in combination, to provide the best signal-to-noise (and therefore sensitivity) performance when studying quartz grains. One of the more significant variables discussed there is binning of the CCD – a subject discussed earlier in section 2.2.2. It was therefore decided to attempt one further regen dose SAR cycle, but this time with the CCD set, via the Winview control software, to employ 4 by 4 binning across the entire chip area. The same disk of 100 µ grains was used again, after being given another 180 second (~18 Gy) applied beta dose. The following two figures show the results.

Figure 3.2.18 Blue-stimulated OSL after 180 s (~18 Gy) beta, with 4 by 4 CCD binning – first and second shines.
The improved visibility of the grains, and the far greater number of grains visible, are immediately obvious — highlighting the power of binning as a performance tuning technique. Unfortunately the loss of spatial resolution due to the effective 128 by 128 pixel count of the image (rather than the full resolution of 512 by 512 pixels) is also obvious.

In Figure 3.2.19 a different area has, of necessity, been selected than in the full resolution images examined earlier, and it may in fact include two or more grains — the lower resolution of the image makes this hard to determine. That said, the total difference between the first and second shine is now 1,849 counts (9,516 vs. 7,667). This — representing as it does some 25% of the total counts, including reproducible (and subtractable) noise — is undeniably a statistically significant result. More important perhaps than the total photon counts to our belief in the significance of these results (if the first image at Figure 3.2.18 is not sufficient), is the per-pixel average difference between the two shines of over 9 counts (compared to between 1 and 2 counts per pixel in the un-binned runs).

The PCIS clearly has the potential, albeit following more thorough performance tuning, to undertake significant studies of quartz luminescence, including palaeodose estimation for dating applications, on (at the very least the brighter grains of) sediments from a broad range of sites and settings.

---

14 Appendix K, “PCIS Data Processing Specification” includes a method whereby a full resolution incident light image is used to select areas (or grains) for analysis, and the corresponding pixels of lower resolution, binned luminescence images are then summed. This resolves the problem with grain selection that occurs here.
3.3 Phase 1 PCIS - Control Architecture Investigations

One of the principal goals of the Phase 1 PCIS operation was to determine the architecture to be used for operation and synchronisation of the various PCIS components in the final, fully automated implementation. The various architectures under consideration are discussed in detail in Chapter 4 below, but can be summarised here as the “Flat Architecture” and the “Hierarchical Architecture”.

For various reasons (again discussed in detail in Chapter 4) the hierarchical model was preferred, with the camera synchronised to the rest of the PCIS via TTL-level electrical interfaces, rather than being controlled directly by software. This approach is dependent on certain features of the LN-CCD camera system and the ST-138 controller working as per my understanding of the supplied documentation; but, if viable, would allow the PCIS to operate using the standard WinView software supplied with the camera system, and obviate the need to develop custom software to manage the camera system. Given the extremely limited resources available to the project, the advantages of this approach would be considerable.

The benefits and limitations of the architectural options are discussed fully in Chapter 4; in this section I describe the details of the ST-138 functions involved, and the testing conducted during Phase 1 PCIS operations to establish the viability of the electrical synchronisation methods underlying the proposed Hierarchical Architecture (referred to in the following as the “proposed architecture”). These tests employ the Phase 1 control electronics described in Section 3.1 above.

3.3.1 The ST-138 Controller Timing Modes

Deciding how the LN-CCD camera system should be controlled and synchronised with the other components would clearly be a critical factor in determining the preferred control architecture for the PCIS. The available timing and synchronisation modes of the LN-CCD camera system are documented in the ST-138 operations manual in the form of control flowcharts and signal timing diagrams.

The first decision to be made is between the “Synchronous” and “Asynchronous” operating modes. In the Asynchronous mode, each image captured is fully processed (including any image mathematics, background subtraction etc.) and displayed before the next image capture is commenced. In the Synchronous mode, the detector is made ready for the next image capture as soon as the current data has been read out from the CCD; post-processing and display of the image are performed in background, as processing time allows, and so may lag behind the image capture. These modes are detailed in Figure 3.3.1 below, taken from the ST-138 operation manual.

The Asynchronous mode would be chosen for an application where the continued operation of the experiment is dependent in some way on the data collected to date; in this case processing and examining the current image would be more urgent than collecting the next image. For an application such as the PCIS, where the course of the experiment is pre-determined (according to the chosen luminescence protocol), and the data is not processed or examined until the entire experiment is complete, the Synchronous mode is preferred; this allows the most rapid acquisition
of images, and offers the tightest timing control over the experiment. Importantly, selecting the Synchronous mode also enables the external electrical synchronisation facilities of the ST-138 controller, which are necessary for the proposed hierarchical architecture.

Figure 20. Chart of Synchronous and Asynchronous operation.

Figure 3.3.1 Synchronous and Asynchronous modes of the ST-138 LN-CCD Controller (Princeton Instruments).
3.3.2 The ST-138 External Synchronisation Options

Once the Synchronous timing mode has been selected via the WinView software, the External Synchronisation option may also be enabled. This allows the start of each accumulation to be controlled via external TTL inputs on the back panel of the ST-138 controller, and is detailed in the timing diagram below (Figure 3.3.2), taken from the ST-138 Controller Operation Manual.

![Timing diagram for the External Sync mode.](Image)

Figure 3.3.2 Operation of the ST-138 Controller External Synchronisation modes (Princeton Instruments).

The PCIS would clearly require custom electronics to provide, at the very least, current- and time-controlled drives to the 4 LED strings installed in the source turret. The obvious approach would have been to supply TTL signals from the custom electronics to control both the start and the end of the accumulation (with the shutter controlled either directly from the custom electronics, or from the ST-138 outputs). Unfortunately, the WinView/ST-138 combination does not offer the facility to control the end of an accumulation from an external signal – only the start of an accumulation.

The technique adopted to counter this limitation was to use the “Shutter Preopen” mode of the ST-138 in combination with the “External Sync” and “Free Run” modes (see the timing diagram above). The combination of External Sync and Free Run modes allows a series of accumulations to be taken without further software interaction (up to the limit set in the Experiment Setup options under WinView; set to its maximum value of 999 for all PCIS operations). The start of each accumulation is delayed until the detection of a falling edge on the External Sync TTL input.
The “Shutter Preopen” mode is designed to allow the shutter to be left open prior to the start of an accumulation, so that any shutter opening delays are eliminated. However, while ever the shutter is open, the CCD is actually accumulating charge from any incident photons – i.e. the accumulation is actually already underway. To counter this fact, a mode called “Continuous Cleans” is, by default, turned on with “Shutter Preopen”. In this mode, the charge is continuously read off successive strips of the CCD (thereby “cleaning” the CCD – see the sidebar information) until the External Sync signal is received. This will incur a small delay (dependent on the strip width selected) before an exposure starts; so in the case that the field of view is dark before the desired accumulation event, and no charge is accumulated during the pre-open period, the “Continuous Cleans” function can be switched off. In either case, the “real” accumulation then occurs for a duration specified (as the “exposure time”) via the WinView experiment setup parameters.

In the proposed PCIS architecture the actual accumulation of luminescence image data occurs during this “Shutter Preopen” period; and by the time the external synchronisation signal that would normally indicate the start of the desired accumulation is received, the actual PCIS accumulation is already complete. By setting the accumulation (or “exposure“) time in the WinView software to 0 seconds, the “start accumulation” external signal would actually trigger the immediate end of the PCIS accumulation. (To be conservative, this time was set to 0.1 seconds during testing.)

A consequence of this operating mode is that “Continuous Cleans” must be turned off; so if the shutter has been “pre-open” for any considerable period of time before the start of the desired accumulation, the surface of the CCD must be explicitly cleaned immediately before the actual accumulation. When accumulations occur in rapid succession (as is normally the case during the collection of luminescence data from a sample), the cleaning may be omitted between accumulations.

Surface Charge Cleaning of CCDs

Cleaning any extraneous charge (whether of thermal or optical origin) from the surface of a CCD is a necessary noise reduction step before acquiring an accumulation or “photo”.

In the early days of CCD-based digital cameras there was an extended and annoying delay between pressing the shutter button and the picture actually being taken. This was the surface cleaning period, and it rendered early digital cameras virtually useless for action photography.

Surface cleaning was accomplished by reading the CCD surface – just as for an actual accumulation or “photo” – and then discarding, rather than storing, the data. Much development was done devising special circuitry within the CCD to allow surface cleaning to occur much more rapidly, leading to the highly usable digital cameras of today.

The LN-CCD is optimised for low noise (and hence high sensitivity) rather than for rapid operation, and does not contain the special cleaning circuitry. Cleaning the CCD is still achieved by reading out (but not storing) an accumulation, and so with the slow (50 KHz) read-out rate required for ultra-low noise operation, the clean cycle takes approximately 5 seconds.
(Since the cleaning operation is slow, avoiding extra clean cycles during a multi-accumulation shine is important.) The proposed architecture would therefore require explicit control over the timing of cleaning cycles, as well as of accumulations.

Control of cleaning cycles is provided by the use of the External Trigger TTL level ST-138 input in conjunction with the External Sync input described above. (Refer again to the timing diagram at Figure 3.3.2 above.) When a falling edge is detected on the External Sync input, and reading of the surface of the CCD is initiated, what is done with the data depends on the External Trigger input. If there has been a falling edge on the External Trigger input since the previous reading of the CCD, then the data is stored – and an image has been acquired. If there has not been a falling edge transition on the external Trigger input since the last such transition on the External Sync input, then the data read from the CCD is discarded – and a surface clean has been performed!

The end of an image accumulation (or of a CCD surface clean) is indicated by an output signal from the ST-138 controller, “NotScan” (see Figure 3.3.2). This indicates that the CCD surface is not being scanned – i.e. that the surface read-out process is complete. This signal may be routed to the custom electronics so that the PCIS can detect when the LN-CCD is ready for the next accumulation.

The following section reviews the ST-138 controller signals involved in implementing the proposed hierarchical architecture. The testing undertaken with the Phase 1 PCIS to prove the signals operated according to my understanding, and that the proposed architecture was viable, is then described.
3.3.3 The ST-138 Synchronisation Signals

The viability of the proposed hierarchical architecture depends on the use of three TTL signals (two inputs and one output) of the ST-138 controller. These signals are reviewed briefly below.

3.3.3.1 External Sync (ST-138 TTL Input)

In the appropriate software run mode, a falling edge on this input causes the camera system to acquire an accumulation. The duration of the acquisition (i.e. the “exposure time” of the “photo”) is determined by a value previously set in the software. In the proposed PCIS architecture the “exposure time” would be set to (effectively) zero, and the signal would therefore indicate the end of the accumulation. The actual duration of the accumulation would be determined by the timing of the control signal from the control electronics, and may therefore be controlled without access to the software-set accumulation duration.

3.3.3.2 External Trigger (ST-138 TTL Input)

The External Trigger input to the ST-138 is used to determine whether an accumulation triggered by the External Sync input should be stored or discarded. If there has been a falling edge on this input since the previous falling edge on the External Sync input, then the camera system will read and store the CCD data as an accumulation, or image. If there has not been such a falling edge on the External Trigger input since the last External Sync falling edge, then the CCD data is read but then discarded. This sequence acts to perform a “clean cycle” on the CCD, removing any accumulated noise signal and preparing the LN-CCD for the next “real” accumulation. Thus a combination of these two ST-138 inputs allows external electrical control of both CCD surface cleaning cycles and image acquisitions.

3.3.3.3 NotScan (ST-138 TTL Output)

This curiously-named output of the ST-138 provides an active-low TTL-level output when the ST-138 is not busy. In this case, “not busy” means that it is not actively reading out an accumulation (whether for storage or for discard) from the LN-CCD. When the ST-138 is “not busy”, the camera system is ready to perform another accumulation. Thus a falling edge on this output following a Sync-triggered clean or accumulation cycle would indicate the end of that read-out, and the readiness of the camera system for the next operation. In the hierarchical architecture(s), this signal would be monitored by the control electronics to support synchronisation of the LN-CCD camera system with the rest of the PCIS.
3.3.4 Phase 1 Testing of the Synchronisation Signals

Testing of the proposed control architecture functions using the Phase 1 control electronics was actually a relatively simple procedure. The versatile TTL Signal Monitor/Generator (see Section 3.1.4 above) was connected to the 3 TTL signals on the back of the ST-138 controller, providing switches to toggle the Sync and Trig inputs, and a 3-state LED display to monitor the state of the NotScan output. Combined with single low-level Minisys commands issued from the 1-line interface of the Risø-supplied “center.exe” application (and the appropriate software modes in the WinView camera control software), it was possible to show that all ST-138 signals operated as expected. It would indeed be possible to control both cleaning of the CCD and the acquisition of images using external electronics, and without access (during a PCIS run) to the LN-CCD control software.

Additional testing included use of the TTL Monitor/Generator to demonstrate external control of the opening and closing of the shutter using a TTL signal via the commercially-supplied shutter controller (which generates the higher voltage waveform necessary for fast and reliable operation of the large 45 mm shutter required by the PCIS; the normal 35 mm shutter, requiring less power to operate, can be controlled directly from the LN-CCD camera). No electronic monitoring was necessary to confirm the correct operation of the shutter, as its operation is quite loud and easily monitored by ear. (As was described in Chapter 2.7, “Shutter Housing”, electronic detection of the shutter state – and the green LED by which it operated – had been disabled.)

The TTL Monitor/Generator was also used to confirm the correct operation of the optional 8-output 16-input I/O card installed by myself into the Minisys control hardware. This quick test confirmed that it was indeed possible to generate and monitor TTL signals using the supplied commands of the Minisys low-level control language. A further test confirmed that specification of a relay number as the light source in a Minisys OSL or TOSL (Optically or Thermo-Optically Stimulated Luminescence) command did in fact cycle the relay as expected at the time point that the optical stimulation should commence. These tests also confirmed the inverted logic (active-high rather than the usual active-low) employed by the Minisys TTL signals.

Finally, the wiring cross-connect box that formed part of the Phase 1 PCIS control system was used to directly connect the Minisys I/O card outputs and inputs to the three ST-138 TTL control lines, and to the shutter controller drive input, to confirm that the entire system could be controlled appropriately using only low-level Minisys commands over the Minisys RS232 interface, and without any mechanical manipulation of switches.

These tests proved the viability of fully automated control of the PCIS using the electrical interfaces to the camera system, and thus the viability of the hierarchical architecture(s) under consideration. Note however that the extremely large number of low-level Minisys commands required to implement even a single image acquisition cycle precluded use of this as a viable PCIS control mode without the development of fairly sophisticated PCIS control software, which would generate the large number of low-level commands from a much simpler high-level description of the required operations. The development and implementation of this software is described in Chapter 5.
4 PCIS Design and Development – Automation and Control

“People who are really serious about software should make their own hardware.”

Alan Kay, Computer scientist

The experiments conducted on the Phase 1 PCIS revealed three major tasks requiring attention before the final automated PCIS build. These were:

1. re-aligning the optics to achieve better centring and focus of the image concurrently;
2. filtering the blue LEDs to eliminate the short wavelength emission tail that was producing a leakage image through the UG11 filters used for blue/UV OSL (Optically Stimulated Luminescence with “blue” stimulation photons and detection in the UV) on quartz; and
3. development of the final automation and control system for the PCIS.

The procedure developed and applied for the improved alignment of the optics, and the successful results thereof, were described earlier in Sections 2.5.7 and 2.5.8.

The use of a small section of Lee UV plastic filter in front of the blue LEDs (and the techniques used to support drawing the LEDs back by 5 mm so that the filter could be fitted in front of them) were described earlier in Sections 2.6.2 and 2.6.3.

Although both of these developments were undertaken as a result of early testing of the Phase 1 PCIS, and were essential parts of the Phase 2 PCIS deliverables, the solutions were actually implemented and tested prior to the development of the final control electronics (the IICU), using the interim electronics described in section 3. They have therefore been described as part of the Phase 1 PCIS development, along with the development of the optics and the source turret respectively.

At the conclusion of what has been described as the Phase 1 development therefore, the PCIS was in essentially its final state as far as opto-mechanical components and performance is concerned. (Certain components were subject to disassembly for anodising, and a few ancillary components, including aperture plates for the second filter drawer, were yet to be produced.) Indeed, it was suggested to me that I should cease development at that point, and commence writing it up as a thesis. However, my goal of delivering a usable and useful machine for luminescence experiments required that a comprehensive automation system be developed for the final PCIS.

This chapter describes the development of the control and synchronisation hardware (the Interface and Integration Control Unit, or IICU), its firmware, and the control tables that specify and define its functions and capabilities.

The following chapter describes the suite of software that finally provides for the simple definition and automated execution of complex luminescence protocols.
4.1 System Integration Paradigm

As has been described above, the broad PCIS integration architecture is determined by the existing command/polling control architecture of the Minisys, and the need to re-use as much as possible of the existing control systems. Because of the delays imposed between commands by the slow serial connection over which the Minisys command structure is passed (approximately 0.1 seconds for each command and response), as much as possible of the timing control must be devolved “down the hierarchy”, closer to the actual components with critical timing constraints. This needs to be achieved via existing supported command structures, which are executed within the Minisys hardware without further serial commands and responses, rather than being controlled at every step by the PCIS control software via the slow serial connection.

Two major architectural approaches to these requirements are described here as the “Flat Architecture” (see Figure 4.1.1) and the “Hierarchical Architecture” (Figure 4.1.2).

In the Flat Architecture, a single PC is used to run all the necessary control software, and to collect the images associated with the run. The control PC is directly connected to the Minisys controller, the ST-138 camera controller and to the PCIS custom control electronics. The control software directly operates three external I/O channels: one running the Minisys low-level control language over an RS-232 serial line; one running a custom control language over a serial (RS-232 or USB) connection to the custom electronics; and a high speed serial interface to the camera controller.

![Diagram](image-url)

**Figure 4.1.1** The Flat Architecture model for the PCIS.
In this architecture, the custom PCIS software would need to incorporate code to manage all three of these interfaces, including code to control the camera system (and to collect and record the images produced) which would use the facilities of the Princeton Instruments software development kit (SDK) for the camera/controller system. The SDK comprises a collection of runtime libraries providing subroutine calls to control all of the important camera parameters and operating conditions. These libraries are written in Visual Basic, and are intended to be called by code written in the same language, or one of a small set of languages with compatible subroutine-call structures.

The need, with the Flat Architecture, to manage the synchronisation of all three communications channels in such a manner as to prevent indeterminate timing delays due to the different speeds of the interfaces would add significant complexity to the software development. It would also imply abandonment of the synchronous command/poll-for-response paradigm underlying the existing Minisyx control systems.

In addition, the Flat Architecture model requires the development and implementation of a command/response communications structure for the custom PCIS electronics; and the learning of the complete system for software management of the camera system (since all aspects of the camera system – not just those unique to the PCIS – would need to be controlled by the custom PCIS runtime software). These represent not inconsiderable development tasks on a project that is already severely resource-limited.

The Hierarchical Architecture model – diagrammed below at Figure 4.1.2 – avoids many of these problems and additional development tasks.

As is seen in Figure 4.1.2, the Hierarchical Architecture requires that the custom PCIS control software manage only a single interface – the serial RS-232 interface to the Minisyx controller. Since facilities for RS-232 serial communications are well supported in nearly all existing languages (including Visual Basic); since I have many years of experience in developing software for RS-232 communications ports; and since the Minisyx control language already exists and is adequately defined and described in the Minisyx documentation, this is clearly the easiest (in terms of expected development effort) of the three interfaces required by the Flat Architecture.

The management of the camera system, and the collecting of images from the ST-138 controller, are managed by independent software. All of the important PCIS facilities can be implemented and supported by the standard, supplied WinView software package. (Some small limitations involved in this approach are described below.) Whether the custom PCIS and the WinView camera control software now run on the same or different PCs is not critical to the overall operation and functionality of the PCIS. In the end it was easier to leave the camera software running on a separate PC, and simply to replace the original Minisyx “sequence” software with custom PCIS control software running on the original Minisyx control PC. This is the option represented in Figure 4.1.2.

In this Hierarchical Architecture, the custom PCIS electronics is controlled directly by the optional relays and optical inputs already fitted to the Minisyx; and indirectly by “OR” (Operate Relay) and “RO” (Read Optical) commands embedded into the stream of Minisyx control and monitoring commands on the serial interface to the Minisyx controller. In many instances the Minisyx will now manage the timing and synchronisation of functions of the Minisyx and of the PCIS control electronics, both ensuring good timing control and simplifying the custom software development.
In the Hierarchical model, synchronisation of the camera system to the rest of the PCIS is managed by TTL-level electrical signals exchanged between the custom PCIS electronics (which will become the IICU), and the ST-138 camera controller. This method was investigated and proven during the Phase 1 PCIS operations described in Chapter 3.

In the Hierarchical model the shutter is controlled directly from the custom PCIS electronics. (In the Flat Architecture it is unclear whether the shutter is best controlled in this fashion, or from the ST-138 controller, and this would require further investigation.)

The considerations discussed above, taken in combination, argued heavily in favour of the Hierarchical Architecture, both on functional and timing grounds, but also, and critically, on the grounds of minimised development effort. On completing this analysis (and the investigations described in Chapter 3), the Hierarchical Architecture was selected as the preferred model.

The advantages and limitations of the preferred Hierarchical Architecture, and the possibilities of its future development should additional resources become available, are discussed below.
4.1.1 Benefits of the Integration Paradigm

There are two principal benefits arising from the adopted integration paradigm. They are: a reduced development effort (for the same end functionality) compared to alternative approaches; and the retention of tight timing control of critical events.

4.1.1.1 Development Effort vs. Functionality

The reduced development effort with the hierarchical architecture relates principally to two factors:

1. the lack of a requirement to manage and synchronise three interfaces, each with a different logical structure and physical implementation (transport layer in network terminology); and
2. avoiding the need to learn the camera control software development environment, and to use it to control a broad range of camera set-up options and operating parameters.

While there may be increased requirements (and therefore development effort) on the custom PCIS electronics with this approach, the general rule of “divide and conquer” applies. Two problems are easier to design around and resolve if they are kept separate, than if they are rolled together into a single “super-problem”. The flat architecture creates out of the PCIS control software a single “super-implementation” dealing simultaneously with the idiosyncrasies of both the Minisys and the camera controller – as well as any complexities that may emerge from the design of the custom control electronics interface.

The hierarchical architecture model separates these issues into separate developments, or indeed – in the case of the camera control facilities – eliminates the problem altogether.

The compromises in terms of functionality associated with the hierarchical model are minor, and are discussed in section 4.1.2 below.

4.1.1.2 Event Timing Control

The tight timing control offered by the hierarchical architecture is attributable to the ability (described in Chapter 3) to use one of the Minisys relays as the “light source” in Minisys OSL and TOSL (Thermo-Optically Stimulated Luminescence) commands, allowing nearly instant electrical triggering of the custom electronics to initiate the next step of a stimulated luminescence operation.

In addition, with this approach the camera system is sitting ready for the next accumulation (actually, already undertaking the next accumulation), waiting only for an electrical signal to trigger the completion of the operation. Camera system delays are therefore also minimised.

It is important to realise here that all serial communications methodologies incur what may be called a “packet delay”, whereby the entire message (even if only a single bit) must be received (including any message headers, error correction codes etc. principally designed to support the successful
Development and Application of a Photon Counting Imaging System

transfer of longer messages), and checked for any errors, before the message is passed on to the receiving system. Many of these considerations also apply to parallel communications technologies, which are generally designed with message header and error correcting facilities similar to those of serial communications techniques. This is because nearly all parallel communications standards are designed to support messages longer (in bit count) than the "width" of the parallel interface.

However fast the communications path for longer messages (in terms of bits or bytes per second), in the case of a single bit message it is the message header processing and similar delays that will dominate. Thus there is still a "packet delay" associated with each communication. Depending on the actual transport layer (or "communications hardware") used, the packet delay may amount to tens of milliseconds. (With the 9600 baud RS-232 connection used by the Minisys controller the typical packet delay – for a 2 to 4 byte command and response – totals around 100 milliseconds.)

With a single-bit communications channel (the division into serial or parallel is meaningless here), there are no such delays. The TTL-level signals proposed for use at critical communications points in the hierarchical model (such as the camera system control signals described in Chapter 3 and tested during Phase 1 PCIS operations) avoid the complication of packet delays.

The single electrical signal can be detected at electronic speed (i.e. in microseconds), and can be used to generate a hardware interrupt in the receiving electronics, so that the managing software can react to the generated signal, again with delays limited to the microsecond scale.

Ensuring tight timing control over luminescence experiments will clearly be more achievable with the hierarchical architecture – where all timing-critical signals are conveyed via single-bit TTL-level electronic signals – than would be the case with the flat architecture model.

4.1.2 Limitations of the Integration Paradigm

The limitations of the hierarchical model with respect to the flat model are all associated with the implicit inability to re-program the camera system during a PCIS run. (With the hierarchical model the WinView software is pre-set into the appropriate operating modes, and there is no further control of the camera system available to the PCIS runtime software.)

On examination of the camera system parameters that are able to be managed via the software interface (whether the standard WinView software or custom software built using the software development kit), it emerges that there is only a single parameter that a user might realistically wish to alter during a run – and that is the binning factor applied to the CCD.

Even this one parameter would not generally be changed during a run; it would be set to achieve the sensitivity required for the particular experiment, and, once the data is limited to the spatial resolution implied by the binning factor, there would be little if any benefit in acquiring some of the images (those with a lower sensitivity requirement such as focussing images) at a higher resolution (i.e. with less radical binning applied). In the unlikely case that an unusual experiment did require changing the binning factor, the experiment would need to be designed as two or more separate PCIS runs, each with its own associated binning factor set in the WinView camera control software.
A minor limitation imposed by the separation of PCIS control and camera system control in the hierarchical model is that it will now be necessary for the operator to start two software processes, and to enter certain information (in fact, just the run name) into both systems. This is only a minor problem, and is significantly mitigated by incorporating (into the PCIS runtime software) operator prompts for the necessary with the camera control system actions. The operator will also be prompted to gather the output data from both systems into a single directory for later analysis.

The obvious conclusion is that the limitations imposed by adoption of the hierarchical architecture model are minor compared to the additional complexity of, and implementation effort required for, the flat model. When also considering the timing control issues, the decision to favour the hierarchical model basically makes itself.

4.1.3 Future Development of the Selected Approach

It is worth noting here that adoption of the hierarchical model now does not prevent a future development that adds camera control facilities to the PCIS runtime software. Additional commands would need to be designed and implemented in whatever “protocol definition file” format is eventually adopted; camera initialisation procedures would need to be included in every protocol definition (though they would probably only need to be developed and written once); and the camera control and PCIS runtime systems would both need to operate on the same PC.

Adoption of this development strategy – discussed further in the later section on “Future developments of the PCIS” – would obviate all of the restrictions noted above as being associated with the hierarchical architecture, whilst retaining all of the benefits of tight timing control over experiments.

Thus the hierarchical architecture provides a development path that should allow all necessary features to be implemented within the limited resources presently available to the project; but will also provide a sensible development path towards the more complete control of the PCIS from a single point that would be desirable if the project were not resource-constrained.

There remain no substantive arguments in favour of adopting the flat architecture – or any of its variants – and so the hierarchical architecture is adopted. The custom PCIS control electronics can now be designed to support the role required of them in the hierarchical model. That design and development is described in the sections that follow.
4.2 Background to the IICU Development

Experiments conducted with the Phase 1 PCIS had clearly established the fundamental requirements of the control and automation system to be designed for the final PCIS. These requirements were:

- four independent current drivers for the LED strings;
- provision for synchronisation of the PCIS with Minisys operations;
- stimulation and image-capture timing requirements; and
- synchronisation of the LN-CCD camera system to the PCIS.

The overall PCIS system architecture implied by these requirements, and by the selection of the Hierarchical Architecture model, is shown below at Figure 4.2.1. (This diagram is taken from the IICU Functional Specification at Appendix D.)

Following the visit recounted earlier to Damien Jones (Prime Optics) to discuss the problems with the original optics alignment, I found myself visiting my old friends at FlowTrack Electronics (remember, the ones who use verbs – see footnote 4 on page 10). In this macho environment of big transformers and discrete power semiconductors (FlowTrack engage mainly in the development of custom remote area power systems), I designed a PCIS control system using fixed logic, discrete transistors and basic (7400 series TTL) LSI logic chips.

Facing the defining limitation of 8 bits of control data from the 8 Minisys relay outputs, the design used front panel switches to pre-select the meaning or effect of combinations of input bits. The input bits, along with data about the switch positions, would then be de-multiplexed to provide a discrete control line for each preselected combination of functions.

The front panel switches would select:

- the pair of LED strings in use for an experiment (a focussing string and a stimulation string, of which the latter may be Null to support pure thermoluminescence operations);
- the range of currents from which the input “command bits” could select (each switch position specifying both the set of focussing LED currents available, and the set of stimulation LED currents available); and
- the range of stimulation times from which the input “command bits” could select.

The input “command bits” would directly specify:

- the issuing of a new command (the “Command Trigger”);
- timing of the stimulation phase of the command (the “Secondary Trigger”);
- whether to use the focussing or stimulation LED string; and
- whether to cause the camera system to record and store an accumulation.

This would leave 4 bits to select one of sixteen LED current and drive duration options, from the set of such options currently selected via the front panel switches. The timing selection was designed around “additive resistances” with an NE555 timer, as discussed in the interim electronics description above. The current control was also specified using switched resistances across the sensor terminals of a standard three-leg current controller IC.
The original 3-page design outline for the non-computerised (or analog) PCIS Control Electronics is reproduced at Appendix C.

Figure 4.2.1 PCIS System Architecture Diagram – from the IICU Functional Specification.
4.2.1 Design of the Digital IICU

The discrete component design for the PCIS control electronics was presented to Daniel Cummins in the Electronics workshop at the RSES. Considering what I was trying to achieve, and the inherent complexity of the implementation, he was of the opinion that a programmable design based on a digital microprocessor would be a better option, for the following reasons:

- the initial design and construction complexity (and therefore costs) would be similar;
- implementing any bug fixes or requested enhancements to the discrete component design would require changes to the electronics, and would require significantly more effort than similar changes (which would be made in software) with the programmable design; and
- the programmable design would offer more flexibility, and therefore more potential functionality for the resultant Phase 2 PCIS.

Accepting this advice with gratitude, as I could clearly see the increased functionality and flexibility that the software approach would offer, I went away to produce a revised specification for a programmable, microprocessor-based Integration and Interface Control Unit (IICU) for the PCIS.

4.2.1.1 Overview of the Digital Design

The fundamental problem in designing and implementing a control unit for the hierarchical architecture is the limitation to 8 bits of data in the control interface. This limit is imposed by the number of relays in the optional Minisys I/O card that was installed earlier. (This card sports 16 optically-isolated inputs, so the channel width for retrieving status information would not be a limitation.)

There are clearly more than eight concurrent 1-bit functions that the control unit needs to implement, so the problem may be described as one of appropriately de-multiplexing the eight control lines into the larger number required to directly control the needed functions.

Perhaps due to my background in computing, during which I had designed and built several devices with similar de-multiplexing requirements, I was immediately drawn to the control table design that was eventually adopted. This is highly reminiscent of — and was probably sub-consciously based on — a common technique of processor implementation called micro-coding. With this technique (which I had used many years before when designing a small computer around a basic bit-slice processor chip), the bit patterns representing individual instructions from the processor’s instruction set are used to address a relatively short and wide table of values stored in (permanent) memory.

This table is short because it typically contains only a few entries for each processor instruction code that it implements; it is wide because it must contain a bit (or group of bits) for every single electronic function in the hardware. Each and every control line or address line of every gate, multiplexer, register, or other device — including the basic processor chip itself — must be represented by a separate bit in the control table. There is no overlaying of functions, because there is no further de-multiplexing of the signals represented by the data bits in the micro-code table.
In the IICU this process is modified slightly, in that each received command (the equivalent of an executed instruction in the micro-code model) “executes” only a single line of the control table. There is no equivalent in the IICU architecture of the “basic processor” hiding – like a homunculus, the “ghost in the machine” – inside the micro-coded processor design, so there is no equivalent of the calling, looping or even sequential processing that the micro-code design relies on. In the PCIS, the execution of a sequence of IICU functions would require the issuing of a sequence of IICU commands by the PCIS runtime software.

[Note that the original IICU design, which appears at Appendix D, did call for the ability to have a “Reset” command execute up to three further instructions following sequentially in the table, to support initialisation functions that one might wish to have associated with the Reset. This, however, was not implemented in the final IICU, as it proved to be inconsistent with the overall single-instruction design and would have been difficult to implement; and in any case the need could be just as easily met by designing the runtime software to call for the additional commands to be executed. As the overall lower implementation effort – and equally functional – option, this was the approach finally adopted for the PCIS operation, and the IICU reverted to a consistent design executing only a single control table entry for each command implemented.]

With this design it is clear that only one line of the control table can be executed at any one time, so only one line of the table can be “active” at a time – even though many of the functions it controls may then be “active” simultaneously. The available input lines – any combination of which may be active simultaneously – can therefore be de-multiplexed to provide an “address” into the control table, so that if there are n available control lines, there may be 2^n rows in the control table.

Nominally there are eight control lines available from the eight Minisys relays. However, one line is necessarily required to indicate to the IICU that it is time to execute the instruction addressed by the other lines – the line referred to as the “Command Trigger”. Also, as discussed earlier, there must be a relay allocated as the “light source” for OSL and TOSL Minisys commands in order to achieve the tight timing control over the stimulated luminescence process that the hierarchical architecture makes possible. This line is referred to as the “Secondary Trigger”. This leaves six control lines available to address up to 64 rows in the IICU control table. Therefore the IICU has at any time a repertoire of 64 different available commands, each comprising any desired combination of the individually controlled functions which are represented by the columns of the table.

[In the final implementation a Primary and Secondary table were provided, with special commands defined to switch between them. This provides the capability of having up to 126 functions in the repertoire, even though half of them will require two commands to be issued for their execution. Four such sets of Primary and Secondary command tables are installed, the operative pair being selected by an IICU front panel switch. This allows different researchers to have different sets of 120-odd commands available to them, possibly representing quite different sets of PCIS functionality.]

The range of different individual internal functions that could now be controlled was no longer limited by the number of control lines available (6); nor by the number of table entries that could be addressed (64); but was only dependent on the width of the control table. And the width of the control table (memory being cheap these days) was a free choice that I could make. The possibility of introducing extra functions into the IICU now existed.
4.2.1.2 Functionality Enhancements with the Digital IICU

The minimal requirements for the control electronics were already understood as a result of the investigations described in Chapter 3, and the decisions made regarding the overall PCIS architecture described above. Indeed, the absolute minimum requirements are embodied in the documentation of the original discrete electronics design that appears at Appendix C. I readily concede that the discrete-component design did no more than the absolute bare minimum to achieve effective automation of the PCIS, operating as a stand-alone research tool.

With the decision to design a programmable, microprocessor-controlled IICU, and with the further decision to go with the control table design described above, the range of functions supported could now be expanded - just the control table would need to be wider. I therefore decided to consider what additional functions could usefully be included in the IICU.

The first and obvious choice was to allow independent specification of the drive currents for the 4 LED strings, rather than the selection from a small number of pre-determined current values for a single focussing LED string and a single stimulation LED string, as in the discrete design. It was decided to allow 0.1 mA increments from 0 mA to 409.5 mA (the highest-rated LED string currently installed in the source turret being 250 mA for the IR LED). Thus four 12-bit fields (one for each string) would be required in the control table. (It had been decided, in consultation with Daniel Cummins in the Electronics workshop, to specify four identical current-controlled power supplies, all designed to meet the maximum voltage and current requirements of any LED string - around 32 volts for the blue LEDs, and 250 mA for the IR LED. Although slightly wasteful of power, this would be simpler to design and implement than four different supplies suited to each LED string, and would also preserve the maximum flexibility for possible future PCIS upgrades with different LEDs.)

No benefit could be seen in allowing different durations for the different LED strings in a single shine, so just one duration field would be required - again, replacing the limited choices offered in the discrete design. A 16 bit field was allocated, allowing times from 0.1 seconds to 6,553.5 seconds (or just over 109 minutes). This would adequately cater for any foreseeable use of the PCIS.

Nearly as obvious was the decision to include two software-switched mains power outlets. One would be used to control the Minisys vacuum pump, thus freeing the PCIS from the restriction imposed within the Minisys that the vacuum pump and Nitrogen flow cannot both be active simultaneously. (As discussed earlier, I disagreed with Risø's stated reasons for disallowing this, and firmly believed that starting the Nitrogen flow before the vacuum pump was stopped would improve the scavenging of oxygen from the sample chamber.) The second switched outlet would be available for other associated equipment, such as an external light source that might be used for stimulation via the general purpose port in the source turret. (The original Minisys external light box and liquid light guide could be one such external source, allowing different selections of filters to be used to control the stimulation spectrum.)

It was already known that the IICU would require three TTL-level ports (two outputs and one input) to support synchronisation with the camera system (refer to the discussions in Chapter 3). With the new flexibility of the digital design, it was decided to add a number of auxiliary TTL ports to support integration of the PCIS with other laboratory equipment in the future construction of larger, more
complex experimental setups. In the end two input ports and four output ports were specified, with different functions available to link them with each other, with IICU status outputs to the Minisys, and to the switched power outlets. This would provide great flexibility in the design and assembly of future experiments involving the PCIS. (Precise details of these ports are described in the IICU Functional Specification which appears at Appendix D.)

The digital IICU design offered the opportunity to take more explicit control of the shutter state. To support this two extra TTL ports were specified – “Shutter In” and “Shutter Out”. “Shutter In” would monitor the state that the camera system was attempting to put the shutter in; “Shutter Out” would drive the shutter controller and determine the actual state of the shutter. The linkage between the two would be under software control.

I also decided to incorporate support for light sensors in the source and destination turrets, and micro-switches on each turret to detect when the lens was correctly in place on those components. These inputs would be used by the IICU to force the shutter to close (or to disable it from opening) if there was an excessively high light level in the PCIS optical path (which might cause saturation of and possible damage to the CCD); or if the optical path was open to the room light conditions because the lens was not in place. (In the end, only the destination turret light detector and the source turret micro-switch were implemented, as these were sufficient to provide the protection required.)

Although the digital IICU design did not increase the number of bits of data that could be directed to the Minisys I/O card’s sixteen optically-isolated inputs, the programmable design would make it possible to monitor and report more items of information about the state of the IICU – such as the state of the turret/shutter interlocks described above. In the end 8 bits of status information were specified to be reported back to the Minisys, where they could be read by the PCIS runtime software. Since there are 16 Minisys inputs available, and only 8 are used by the IICU, it was decided to provide “pass-through” connectors for the other eight inputs on the IICU back panel. These are “invisible” to the IICU, but can be read directly by the PCIS runtime software. Routing them via the IICU is done primarily for neatness of cabling in more complex experimental setups, and to avoid the use of split “Y” cables such as have been so annoying on the original Minisys.

An extended range of front panel controls and displays – including a direct display of the total current passing through all of the LED strings at any point in time – was also specified. These are described in more detail later. The PCIS internal architecture that was designed around the revised requirements is shown at Figure 4.2.2, taken from the IICU Functional Specification at Appendix D.

The full Functional Specification for the IICU appears at Appendix D, and it is highly recommended that “if you only read one Appendix this year, make it this one”. It contains a lot of detail about the functionality and implementation of the IICU that is not repeated in the sections that follow. Although a number of changes were made to the functionality during the development and testing of the IICU, and a few new functions were actually added, the functional specification remains the best description, and in great detail, of the greater part of the IICU and its functions.

The following sections describe the IICU, including the enhancements that were made to the IICU specification during the Phase 2 PCIS development. Much of this stands alone, but some parts will make little sense without a prior reading of the original specification. Where the two descriptions differ, the information in the following sections, written after the IICU implementation, is definitive.
Figure 4.2.2 IICU Internal Architecture – from the IICU Functional Specification.
4.2.2 The Digital IICU Implementation

The IICU was developed according to my Functional Specification (see Appendix D) by Daniel Cummins of the RSES Electronics workshop. The hardware platform selected for the development was the Motorola MC9S12A256BCPV MicroController Unit (MCU) and the Freescale CodeWarrior Development Studio for S12(X) Version 5.1.

The circuit design, component selection, PCB layout, and (within the guidelines laid down in the specification) the final front and back panel layouts were all performed by Daniel, with little involvement from myself other than final approval. Daniel also developed the firmware for the IICU that implements the functions of the control tables as described in the specification and in this chapter. (Pseudocode implementations of many of the functions were included in the original specification. These however were intended simply to provide an unambiguous communication of the required functionality, and not in any way to constrain Daniel’s control over the final implementation.) The firmware is written in C, and compiled under the Code Warrior HC12 C compiler.

Following the original implementation, changes to the IICU firmware for bug fixes and functional enhancements were a shared responsibility between Daniel and me. Working from a full copy of the C source code, I would generally provide actual suggested code changes to implement the fix or upgrade required. Daniel would then make the actual changes as he saw fit, and perform initial testing of the changes with the IICU in the development environment. Final performance and acceptance testing was then performed by myself, either within the IICU Bench Testing Environment (see Section 4.4), or in the context of the live PCIS.

The very few electronic fixes and upgrades required during the development were all undertaken entirely by Daniel, according to bug reports or upgrade requests written by me.

Daniel also provided the original Excel spread-sheet under which command tables are developed – a tool he had built for his own needs when installing command table entries to allow testing of the firmware under development. Maintenance and improvement of the spread-sheet was then undertaken by myself. Daniel also made the initial spread-sheet changes to support the change in IICU V2 to the new command table structure. Again, I then acquired responsibility for maintenance and improvement of the spread-sheet software.

All development, installation and maintenance of IICU Command Table Sets for actual use in the PCIS was undertaken solely by myself. Ad-hoc command tables for the testing of individual IICU functions were a shared development, and a shared resource, between Daniel and me. (All such testing tables are distinguished by Command Table numbers in the 90’s, and are usually installed in table slot D. “Real” command tables are numbered from 1 to 89, and may be installed in slots A, B and C.)

Photos of the final IICU front and back panels, and of the principal internal printed circuit boards (PCBs) appear below at Figures 4.2.3 to 4.2.5.
Development and Application of a Photon Counting Imaging System

Figure 4.2.3 The IICU Front Panel showing controls and indicators.

Figure 4.2.4 The IICU Back Panel connectors.

Figure 4.2.5 The IICU Main PCB and Front Panel PCB.
4.2.2.1 IICU Version History

During the implementation of the IICU a small number of changes and enhancements were implemented that place the final IICU at variance with the specification at Appendix D. These changes, and the reasons for them, are discussed in detail in the sections that follow.

It was only very close to the end of the development, however, that any changes significant enough to affect the definition of the command table structure were made. Because these changes for the first time rendered all earlier command tables incompatible with the new IICU code, the final delivered system is referred to as the IICU V2.

The changes made for IICU V2 were primarily aimed at improving the command table structure so that a greater range of operations could be specified within a single command table pair (Primary and Secondary). This in turn delivers greater flexibility of operation of the PCIS without the need to develop and install new command table sets.

Development of new command tables to support new combinations of PCIS operations is a fairly technical process, requiring a detailed knowledge of the entire PCIS software suite, of the Minisys low-level commands and operations, and a particularly detailed understanding of the IICU Functional Specification. Although great flexibility of operation is supported by the facility to develop and install new command tables (and then to develop Sequence and Include files to operate with them), this is not something that a luminescence researcher would typically be keen, or confident, to undertake.

The greater flexibility available with the IICU V2 derives from the exclusion of 3 bits (that turned out not to be required) from the Camera Control Parameters field, and their replacement with 3 bits originally in the Shine Command Parameters field, but actually involved with the LN-CCD camera system synchronisation; the re-structuring of the Shine Command Parameters field into two separately-defaultable 4-bit sub-fields (reflecting the existing structure of the Camera Control Parameters field); and the more logical allocation of functions to bits in the Shine Command Parameters sub-fields.

The effect of these changes is to allow more different combinations of parameters to be represented in fewer rows of the Command Table, since use of the Set Defaults command allows the desired combination to be built up from individual sub-field specifications. Representing all possible combinations of six bits in a single field requires $2^6$, or 64 rows (an entire command table); splitting the six bits into two separately-defaultable 3-bit fields reduces the number of rows required to $2 \times 2^3$, or 16 rows – only a quarter of a command table.

The end result of the IICU V2 changes is that all (meaningful) combinations of Shine and Camera parameters, along with an extended range of current and duration specifications, can be represented in a single command table set.

Also in IICU V2, additional bits were implemented in the Shine Control Parameters to:

- suppress the turning ON of the LEDs during a Shine command;
- suppress the turning OFF of the LEDs during a Shine command; and
- specify the use of the External (IICU back panel) Secondary Trigger input.
The operation of these bits – and their intended use – is described along with the rest of the Shine Command Parameters field in Section 4.4.6 below.

The only other changes made to IICU V2 comprised “cleaning up” of the firmware that implements the Shine command and manages the camera synchronisation signals, to remove some “spaghetti” code that had resulted from the ad-hoc enhancements made to the V1 code. These changes rendered the firmware more maintainable, but made no difference to the functions of the IICU.

At present the IICU V2 is running with an installed command table (Command Table 2) that is the functional equivalent of the table (Command Table 1) that was used with the IICU V1 software for the operations of the Phase 2 PCIS described in this thesis. This maintains full compatibility with the significant body of Sequence files and Include files that have already been developed (see Chapter 5). However, it also means that there has not yet been any tangible benefit, either in increased functionality or ease of use, from the IICU V2 implementation.

An improved command table set for the IICU V2 – actually claiming the potential benefits outlined above – is being designed (as Command Table Set 3), but some details are still under discussion and their resolution awaits further use of the PCIS, and a better understanding of its requirements.

The new command table will require the creation of alternate versions of some of the low-level Include files, to ensure they call the correct commands from the new table. The PCIS software structure supports the use of different low-level Include files for different command table sets, so that the old table and old Sequence and Include files can still be installed and used. (The techniques for achieving this are described in chapter 5.) Thus neither the change from IICU V1 to IICU V2, nor the upgrade from Command Table 2 to Command Table 3, will cause any compatibility issues with the existing library of Sequence definition files, nor preclude the exact repeat of any earlier (IICU V1) experiment using the Sequence specifications automatically saved with the earlier run’s output data.

With the IICU V2 and Command Table Set 3, the PCIS will be flexible enough to make it extremely unlikely that new command tables would need to be constructed for any foreseeable PCIS operations. Since the construction and testing of new command tables is an exacting and laborious task, that needs to be completed with absolute precision, the development effort of moving to the IICU V2 is fully justified.

The following sections all describe the IICU V2. Differences from the original IICU V1 specification are highlighted, and the details in the following sections take precedence over the Functional Specification at Appendix D as a description of the current state of the IICU and the PCIS.
4.3 The IICU Hardware Components and Interfaces

This section provides a summary of the various components, controls and interfaces of the IICU. It does not generally attempt to provide the detail of function or electrical characteristics that are contained in the Functional Specification at Appendix D. The original specification should be referred to for all details not covered here, with the following exceptions:

- the LED Current Driver specifications, which were implemented as four identical units rather than being individually specified to the particular Stimulation LED strings as in the original specification; and
- the External Secondary Trigger TTL Input (which is a new feature not covered in the original specification at all).

In these two cases, all necessary details are provided in the sections that follow.

4.3.1 IICU Front Panel Controls and Displays

The IICU front panel controls are designed to allow the minimum necessary “hardware” control of the IICU. The IICU front panel displays, however, are designed to give a comprehensive view into the operations of the IICU, and of the PCIS, without the “benefit” of software interpretation of the system’s activities. (Extensive software monitoring of the PCIS is also provided by the PCIS runtime software – see Chapter 5).

The individual IICU controls and displays are shown in the photo at Figure 4.3.1 below (which has all of the front panel indicators and displays lit), and are described in the sections that follow.

![Figure 4.3.1 The IICU Front Panel with all indicators and displays lit.](image)
4.3.1.1 IICU Front Panel Controls

The front panel controls of the IICU are minimal. They comprise:

- A power switch – to power the IICU On or Off;
- A Reset push-button switch – to invoke a Hard Reset of the IICU. (Note that this is slightly different from executing a Reset command, which invokes a “Soft Reset”);
- The Command Table Selector switch – a four-way rotary switch (labelled “A” to “D”) allowing the selection of one of the four installed Command Table Sets. (Note that the Reset button must be pressed before the selection comes into effect); and
- Blank Display – a two-state push switch that blanks the entire IICU display (except for a dim red neon in the Power switch to show that the unit is active). This allows the room light level to be minimised when samples are being loaded into the PCIS.

4.3.1.2 IICU Front Panel Displays

The IICU front panel displays are quite comprehensive as, when things don’t work, staring at a bunch of dark, silent black boxes in a dark, red room can be unenlightening and eventually frustrating. For system debugging purposes it seemed important that the IICU itself should display plentiful information about its own state and activities. These displays are also a great comfort and benefit when the PCIS is operating correctly, and normal luminescence analysis runs are under way.

All IICU front panel indicators are either single round red LEDs, or red 7-segment red LED displays. The entire front panel is also covered by a deep red Perspex cover, screen printed with the display-identifying information. This minimises the impact of the IICU on the low-level PCIS room lighting.

The front panel displays are discussed in groups below.

4.3.1.2.1 Command Number and Table Number

These two- and three-digit displays normally show the current selected Command Table Set Number (decimal, 00 to 99), whether the Primary or Secondary table of the set is in use, and the currently active or last executed Command Number (decimal, 00 to 63).

The decimal point of the Command Table Number display is used to indicate that there is a currently active command – i.e. that the IICU is busy.

In the case of an IICU error, all command execution halts, and the Command Table Number display is replaced with a display of the internal IICU Error Number. (Interpreting IICU Error Numbers is best done by reference to the IICU firmware source code. The list of error numbers in the IICU Functional Specification is incomplete, as additional possible error types emerged and were given unique numbers during the IICU development.)
In general, IICU errors are a feature of code development and IICU Command Table development. With mature IICU firmware, and with properly tested Command Tables, the only IICU errors a user is likely to see relate to attempting to perform operations that require “Over-Ride Mode” in order to be available. This feature is provided to help prevent the accidental use of LED currents or shutter states that could potentially harm the PCIS or the LN-CCD.

4.3.1.2.2 Minisys Status Output Displays

A number of the IICU status bits that are reported to the PCIS (via the Minisys optically-isolated inputs) are replicated on the IICU front panel. These comprise:

- Command Active (described in the previous section);
- Error – indicates that an IICU Error has occurred;
- Warning – indicates that some IICU command has generated a Warning condition; and
- Over-Ride – indicates that the IICU is currently operating in Over-Ride mode.

In addition there are LED indicators to show when each of the two switched power outlets is active. These are not part of the status information reported to the Minisys.

4.3.1.2.3 LED Drive Displays

The LED Drive displays comprise four separate LED indicators (to show which subset of the four source turret stimulation LED strings are currently being driven), and a 4-digit display to show the total current (in tenths of a milliamp) being delivered to the four LED strings.

Note that the LED current display is a true current measurement (taken in the common ground return of the four LED strings), and not a calculated sum of the current that should be being delivered. It will thus provide an indication of an open or short circuit (or even a dislodged connector) in the source shutter or the LED drive wiring.

4.3.2 The Current Drivers

In the original functional specification, the four required LED current drive circuits were specifically allocated to the blue, red, UV and IR LED strings. The electrical specification of each of the four drives was set in terms of what was expected to be installed in the source turret in each of the four strings – thus the “blue” drive specified a high voltage (around 38 volts) but a low maximum current (only around 20 mA with the best blue LEDs then available, although 100 mA 5-chip blue LEDs later became available, and were eventually installed); while the “IR” drive specified a low voltage (only around 3 volts), but a much higher current (250 mA or more).
In the end it was decided, in the interests of future PCIS upgrade flexibility but also to simplify the design and construction of the IICU, to implement four identical current supplies, each capable of the highest of the voltage and highest current requirements found amongst the four LED strings.

Accordingly, each of the four power supplies is capable of delivering up to 40 volts, at any current up to 410 mA. (This figure – actually 409.5 mA – was chosen as it exceeded the requirements of the LEDs presently being installed; and it could be specified, to the nearest 0.1 mA, in a 12 bit field of the IICU Command Word – see Section 4.4 below.)

Each power supply is digitally current-controlled by the IICU firmware according to the specifications in the Command Tables. Current control rather than voltage control is used due to the very flat voltage-to-current curve of LEDs – to control the power consumed by an LED, it is sufficient to control its current. Effective power control cannot be achieved by controlling the voltage.

The power supplies operate by total loss of the excess voltage at the supplied current – and thus require significant heat sinks. The waste of electrical power implied by this approach (as opposed to using individual switch-mode power supplies) is not considered to be a problem in a one-off item of laboratory equipment.

Note that two separate current limits are applied to each LED string – a “Soft Maximum Current” and a “Hard Maximum Current”. These are set differently for each LED string to match the characteristics of the LEDs. The Soft Maximum Currents are set in the Reset command (see section 4.4.1.3), while the Hard Maximum Currents are set via the IICU Management Interface (see Section 4.5). The Soft Maximum Currents – usually set to 80% of the rated LED currents – may not be exceeded by any Shine or Set Defaults command, unless the IICU is in Over-Ride Mode. The Hard Maximum Currents – presently set to 100% of the rated LED currents – may not be exceeded under any circumstances.

4.3.3 Interface to the Minisys

The IICU interface to the Minisys comprises three groups:

- command inputs from the Minisys;
- IICU state outputs to the Minisys; and
- “pass-through” Minisys inputs located on the IICU back panel.

4.3.3.1 Minisys Command Input group

The eight command inputs to the IICU are derived from the eight relay outputs situated on the Minisys optional I/O card. They are received at the IICU via a D25 back panel connector.

Six of these inputs specify the Command Number (i.e. the entry or line number of the Command Table) which is to be executed next. A falling edge transition on the seventh input, referred to as the
“Command Trigger”, causes the IICU to execute the command number addressed by the current state of the first six inputs.

The eighth input is the “Secondary Trigger”, and is used by the IICU (when specified by the command being executed) to control the timing of the optical stimulation phase of a Shine command.

4.3.3.2 IICU State Outputs to the Minisys

The IICU State Outputs are delivered to the first eight optically-isolated Minisys inputs via the D37 connector on the IICU back panel. The bits of status information carried are:

- Command Active (i.e. IICU Busy);
- Error (an Error has occurred and the IICU has stopped);
- Warning (a Warning has occurred but the IICU is still running);
- Over-Ride (a command has put the IICU into Over-Ride mode);
- Shutter Too Bright (an excess-light condition exists in the shutter housing);
- Lens Not Down (the lens is not assembled onto the source turret and shutter housing); and
- Minisys TTL Outputs A and B (refer to Section 4.4.4 below).

For details on the meaning of these status bits – and in particular for the functions of Over-Ride Mode and the operations it allows – refer to the IICU Functional Specification at Appendix D.

4.3.3.3 IICU Pass-through Minisys Inputs

All sixteen Minisys optically-isolated inputs are delivered to the back panel of the IICU via a single D37 connector, but only the first eight are actually used by the IICU (as just described).

The other eight inputs are brought back out on the back panel of the IICU – four programmed as TTL level inputs via BNC connectors, and four programmed as dry contact inputs via an 8-way screw terminal connector.

These Minisys inputs are made available at the IICU back panel for cabling convenience only. They are not visible to the IICU logic, and so are not really a part of its “functional” specification.

4.3.4 Interface to the Camera System

The interface to the camera system consists of three TTL outputs and two TTL inputs terminated at BNC connectors on the IICU back panel. These are considered in two groups – the camera synchronisation group and the shutter control group.
4.3.4.1 Camera Synchronisation Group

The camera synchronisation group comprises two TTL outputs, Trig and Sync; and one TTL input, Camera Ready. These appear as BNC connectors on the IICU back panel, and connect to the ST-138 controller ports "Trigger", "Ext Sync" and "NotScan" to implement synchronisation of the camera controller by the IICU. This is achieved in the manner described under discussions of the PCIS architecture options in Chapter 3.

4.3.4.2 Shutter Control Group

This comprises one TTL input, "Shutter In", and one TTL output, "Shutter Out".

The Shutter Out output directly controls the state of the PCIS shutter via the appropriate TTL input on the shutter (electrical) controller.

The Shutter In input connects to the Shutter Out connector on the ST-138 controller, so that the IICU can monitor what the camera controller is trying to do with the shutter. This may be relevant in some operating modes of the IICU, and this signal may (under IICU software control) be replicated onto the Shutter Out signal – effectively restoring control of the shutter to the ST-138 camera controller.

4.3.5 Interfaces to Other Components

The interfaces to other components (i.e. other pieces of equipment that may form part of a complete experiment based on the PCIS) comprise four TTL outputs, two TTL inputs, and two software-switched mains power outlets (one of which is normally used to control the vacuum pump). All these appear on the back panel of the IICU – the TTL connections via BNC, the power outlets via IEC connectors.

The two TTL inputs (General Purpose TTL Inputs A and B) can be linked under software control to the Minisys state outputs Minisys TTL Out A and B, to the first two TTL Outputs (General Purpose TTL Output A and B), and to the two switched mains power outlets. The nature of these links, and how they are controlled, is described in Sections 4.4.4 and 4.4.5 where the relevant IICU Command Word fields are described.

The final two TTL Outputs, Auxiliary TTL Output A and B, are independently controlled as described in section 4.4.5. Each may be programmed into the Low (Active) state, the High (Inactive) state, or programmed to send a short negative-going (Active) pulse. (The duration of the TTL pulses sent by the IICU may be set via the IICU Management Interface – see section 4.5).
4.3.6 IICU – Additional Back Panel Connectors

Several other connectors appear on the IICU back panel. They are:

- a D15 Source Turret (LED Drive) connector;
- a Mini-Din connector for the destination turret (shutter housing);
- a BNC connector for the TTL External Secondary trigger input;
- a D9 connector for the RS-232 IICU Management Interface; and
- a Banana terminal for an external earth connection.

The External Secondary Trigger provides an alternate to the eighth relay input from the Minisys (see Section 4.3.3.1 above) for triggering the start of the optical stimulation phase of a Shine command. This will allow the shine to be synchronised with an external light source, such as the original Minisys Liquid Light Guide optical stimulation source (for which there is an appropriate optical port in the PCIS source turret).

The choice between the Minisys-sourced or the External Secondary Trigger is specified in the Shine Command Parameters field of the IICU Command Word – see Section 4.4.6.
4.4 The IICU Command Table Structure

The IICU is based around a command table, individual entries or "rows" of which are addressed and invoked under the control of the relays on the Minisys I/O card. The structure of each row of the table is shown in Appendix D, but is repeated below as Figure 4.4.1. At this word-and-byte level, the structure has not changed since the original specification.

<table>
<thead>
<tr>
<th>Command Nibble</th>
<th>I-Drives Selection</th>
<th>Default Values Control Byte</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auxiliary Control Byte 1</td>
<td>Auxiliary Control Byte 2</td>
<td>Camera Control Parameters</td>
</tr>
<tr>
<td>Shine Command Parameters</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Duration of current drive – 16 bit value in tenths of a second</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Not used</td>
<td>Current Drive Number 1 – 12 bit Current value x 0.1 mA</td>
<td></td>
</tr>
<tr>
<td>Not used</td>
<td>Current Drive Number 2 – 12 bit Current value x 0.1 mA</td>
<td></td>
</tr>
<tr>
<td>Not used</td>
<td>Current Drive Number 3 – 12 bit Current value x 0.1 mA</td>
<td></td>
</tr>
<tr>
<td>Not used</td>
<td>Current Drive Number 4 – 12 bit Current value x 0.1 mA</td>
<td></td>
</tr>
</tbody>
</table>

Figure 4.4.1 IICU V2 Command Table structure.

The individual control table fields, and their function and purpose in the PCIS, are described briefly below. Full and complete descriptions of each field, with detailed descriptions of their functions in the form of pseudo-code specifications, appear in the Functional Specification. Where a field has changed since the IICU V1 Functional Specification at Appendix D, a complete description of the field's structure and function under IICU V2 is given here.

4.4.1 Command Table Fields – Command Nibble

This field has not changed from the IICU V1 Functional Specification at Appendix D.

The Command Nibble\(^\text{15}\) comprises two sub-fields; the Main Command, and the Sub-command. The Main Command is a two-bit field, allowing the specification of one of three available IICU command types, as well as "No-Op" ("No Operation") – indicating an unused line of the Command Table. (Attempting to execute a No-Op causes an IICU Error.)

\(^\text{15}\) Reflecting what passes for humour amongst computer types, a nibble is half of a byte, or four bits.
The three supported command types are “Shine/Aux” (the command that actually “does stuff”); “Set Defaults” (which sets defaults for Shine/Aux commands); and “Reset” (which resets the IICU).

The meaning of the Sub-command di-bit depends on the command type with which it is associated, and is discussed with each of the command types, which are described separately below.

4.4.1.1 The IICU Shine/Aux Command

The Shine/Aux command is the one that actually does stuff – both undertaking PCIS “shines” (including camera synchronisation and shutter control), and operating all of the auxiliary inputs and outputs of the IICU. Whether the command executes the Shine part, the Auxiliary part, or both, is specified by the two bits of the Sub-command field.

Except for the interpretation of some bits of the Shine Command Parameters and the Camera Control Parameters (which are explained below where those fields are described), the Shine/Aux command operates exactly as described in the original specification.

Where a Shine/Aux command calls for both a Shine and an Auxiliary command (i.e. both bits of the Sub-command field set), the Auxiliary functions are performed first. There is then a short pause before the Shine part is executed. In IICU V1 this pause was set to 100 ms; in the IICU V2 the length of the pause can be set – in 0.1 ms increments – via the IICU Management Interface.

4.4.1.2 The IICU Set Defaults Command

The Set Defaults command is used to build up, either at one go or piece by piece, the internal Default Command Word. The Default Command Word (or specified fields of it) can then be used by a Shine/Aux command to supplement the fields specified in the Shine/Aux command itself.

Using this technique it is possible to build and invoke a much larger set of different Shine/Aux commands than could be stored directly in the Command Table Set.

The operation of the Set Defaults command is controlled by the contents of its Default Values Control field; and by individual bits and sub-fields of the Auxiliary Control bytes, the Shine Command Parameters and the Camera Control Parameters. The details of how these bits interact in the setting and using of default values are fully explained in the IICU Functional Specification, and in the relevant field descriptions below.

The use of defaults in the IICU is necessarily complex, but provides enormous flexibility in the setting and use of default values not only for entire fields, but for individual sub-fields. A careful reading of the Functional Specification, and a proper understanding of all the subtleties of the defaults management, is essential to the task of building new control tables, or of building low-level Sequence Include files (see Chapter 5).
Apart from the splitting of the Shine Control Parameters into two separately-defaultable sub-fields in IICU V2 (documented below), the Set Defaults command operates exactly as described in the original specification.

The Sub-command field of a Set Defaults command is normally set to “00”. However, other values of the Sub-command field cause the Set defaults command to have the following functions:

- “01” – Switch to Secondary Command Table
- “10” – Switch to Primary Command Table
- “11” – Toggle Over-Ride Mode

These functions can be combined with other functions of a fully-populated Set Defaults command. However, this is regarded (by me) as bad practice, and is not done in any of the existing command tables. If the functions are combined, the Set Defaults action happens first. This prevents a single command from both setting Over-Ride Mode and performing an action that requires it, as this would defeat the “safety fence” against unintended actions that the Over-Ride Mode is meant to provide.

4.4.1.3 The IICU Reset Command

The Reset command does what a RESET normally does – it stops any current operations; clears any internal stored states; and puts the machine into a known state ready to commence operating.

In the IICU, the Reset command:

- turns off any LED drives that may be active;
- closes the shutter;
- sets all TTL outputs to their Inactive state;
- turns off the two switched mains power outlets;
- clears any Warning condition;
- cancels Over-ride Mode if it is set;
- restores the IICU to operating on the Primary Command Table; and
- clears any default values stored in the internal Default Command Word.

The Reset command therefore leaves the IICU in a completely known state, from which a new sequence of operations may safely be commenced.

The Reset command also re-populates the internal Default Command Word with default field values to be used in any Shine or Auxiliary commands that specify that default values be used. The Default Values Control Byte of the Reset command specifies which fields of the Reset command should be copied to the Default Command Word, allowing many of the default fields related to Shine and Auxiliary commands to be initialised. The I-Drive Current fields of the Reset Command contain the Soft Maximum Current values for the four LED strings (see the description of Soft and Hard Maximum Currents in Section 4.3.2 above). The Default Currents for the LED strings therefore cannot be set by the Reset command, and must be set by a Set Defaults command instead.
Note also that the mandatory Reset command at Word 0 of the Primary Command Table contains the Command Table Number in the Shine Duration field. The Reset command therefore cannot specify a default Shine Duration, and that field is always initialised to 0.0 seconds.

A Reset command can of course only be invoked when the IICU is idle and ready to execute a command. Therefore it cannot be used to interrupt a command that is "stuck", or simply taking too long; nor can it be used if an IICU Error has occurred, since all command execution is then disabled. The Reset button acts like a Reset Error command, but can be invoked at any time. It will terminate any executing command, and clear any Error condition. It then causes the execution of the Reset command at Primary Command Word 0, including any initialisation of the Default Command Word called for by that Reset command.

Note that in the original IICU specification, the Sub-command field of a Reset command contained a number from zero to three, specifying the number of command table rows consecutively following the Reset command that should be automatically executed by the IICU following execution of the Reset. This was intended to allow a comprehensive initialisation of the IICU and of any attached ancillary equipment by the execution of a series of Set Default and Auxiliary commands after the Reset. This would be the only circumstance in which the IICU would execute more than the one addressed instruction when the Command Trigger input was cycled.

This capability was never implemented, and does not exist in either IICU V1 or IICU V2. The reasons for this absence are:

1. it would, on the advice of Daniel Cummins the IICU developer, be difficult to implement within the code structure that had been designed to support the single-instruction-per-interrupt paradigm of the rest of the IICU specification;
2. lacking any additional budget, I needed something to trade against the additional IICU functions that I wanted (such as the Rapid Repeat Shine mode and the External Secondary Trigger – see later in this chapter), which would not be difficult to implement but which had not been in the specification agreed to; and
3. it would be very simple to implement any extended initialisation of the IICU by the simple expedient of explicitly invoking the additional IICU commands from an "Initialisation" Sequence Include file (see Chapter 5).

The Reset is therefore now a single command like any other, and the Sub-command field is not used.

4.4.2 Command Table Fields – I-Drives Selection

This field has not changed from the IICU V1 Functional Specification at Appendix D.

It specifies what combination of the four LED drives should be used for a Shine operation. Any combination may be specified, from no LEDs to all four strings active.
4.4.3 Command Table Fields - Default Values Control Byte

This field (shown in Figure 4.4.2 below) controls both the way default values are set with the Set Defaults and Reset commands, and the way they are used with the Shine/Aux command. The functioning of this field is actually more complex than it first appears.

In general, the effect of any given bit of this byte being set is:

- for a Set Defaults command – that the entire corresponding field of the Set Defaults command should be copied to the internal Default Command Word; and
- for a Shine/Aux command – that the entire corresponding field from the Default Command Word should be used, instead of any values specified in the Shine/Aux command itself.

Some Command Word fields support selective defaulting of sub-fields, and are even more complex:

- Bit 7 ("Set/Use Default Shine/Camera Control Parameters") is interpreted in combination with the relevant "Use Specified Values" bits of the Shine Command Parameters and Camera Control Parameters fields, to determine how defaulting of individual subfields of those fields is handled, and to determine the post-Shine shutter state; and
- Bit 6 ("Set/Use Default Auxiliary Control Parameters") is interpreted in combination with the values of the sub-fields of the Auxiliary Control Bytes 1 and 2, to determine how the defaulting of Auxiliary functions is handled.

The details of these interactions are explained in Sections 4.4.4 to 4.4.7 below.

<table>
<thead>
<tr>
<th>Bit Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>Set / Use Default Shine/Camera Control Parameters</td>
</tr>
<tr>
<td>6</td>
<td>Set / Use Default Auxiliary Control Parameters</td>
</tr>
<tr>
<td>5</td>
<td>Set / Use Default Shine Duration</td>
</tr>
<tr>
<td>4</td>
<td>Set / Use Default I-Drives Selection</td>
</tr>
<tr>
<td>3</td>
<td>Set / Use Default Drive 1 Current Setting</td>
</tr>
<tr>
<td>2</td>
<td>Set / Use Default Drive 2 Current Setting</td>
</tr>
<tr>
<td>1</td>
<td>Set / Use Default Drive 3 Current Setting</td>
</tr>
<tr>
<td>0</td>
<td>Set / Use Default Drive 4 Current Setting</td>
</tr>
</tbody>
</table>

Figure 4.4.2 IICU Default Values Control Byte, internal structure.

The complexity of this field certainly makes the construction of Command Tables more intricate; but it is necessary in order to achieve the flexibility of operation that allows a comprehensive range of PCIS capabilities to be available within the constraint of having only a single, 128-entry Command Table Set active at any time.

Even though the Shine and Camera Parameters fields that this field operates in conjunction with have changed in IICU V2, the Default Values Control Byte itself has not changed, and the definition of its functioning in the IICU V1 Functional Specification at Appendix D, although terse, is accurate.
4.4.4 Command Table Fields – Auxiliary Control Byte 1

This field (shown at Figure 4.4.3 below) allows specification of the behaviour of Minisys TTL Outputs A and B (two of the eight IICU State outputs reported back to the PCIS via the Minisys optically-isolated inputs); and of the General Purpose TTL Outputs A and B (two of the BNC-connector TTL Outputs on the back panel of the IICU).

<table>
<thead>
<tr>
<th>Bit No.</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Minisys TTL Output A</td>
<td>Minisys TTL Output B</td>
<td>General Purpose TTL Output A</td>
<td>General Purpose TTL Output B</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 4.4.3 Auxiliary Control Byte 1, internal structure.

The Minisys TTL Outputs A and B exist to allow the PCIS to detect and respond to the states of the IICU General Purpose TTL Inputs A and B (respectively). The Minisys outputs can be set to slavishly follow the state of the corresponding General Purpose Input; to latch “On” when a rising Edge is detected on the Input; or to latch “On” when a Falling edge is detected. The latching functions are provided to allow the PCIS to detect a transient input TTL signal, even though only intermittent polling of the IICU state outputs by the PCIS is possible.

The General Purpose TTL Outputs A and B (available via IICU back panel BNC connectors) can be programmed High; Low; or to follow the state of the corresponding Minisys TTL Output (including any latching currently active on the Minisys TTL Output). This mode makes it possible for the IICU to monitor an external TTL signal (including for transient states); re-output to other ancillary equipment an optionally latched copy of that signal to trigger some external event; and report on all these goings-on to the PCIS runtime software.

For all of these sub-fields, any zero value is interpreted as a “no-op”, effecting no change on the corresponding output. This fact is used in conjunction with Bit 6 of the Default Values Control Byte to support the defaulting of individual sub-fields as follows:

- for a Set Defaults command – if Bit 6 of the Default Values Control Byte is set, then the entire contents of the Auxiliary Control Bytes are copied to the Default Command Word. But even if Bit 6 of the Default Values Control Byte is not set, then non-zero sub-fields of the Auxiliary Control Bytes are still copied to the Default Command Word; zero-valued sub-fields will not be copied, leaving the corresponding Default Command Word sub-fields unchanged. (Thus the only way to clear a sub-field in the Default Command Word is to re-write the entire field, using a Set Defaults command with Bit 6 of the Default Control Byte set to 1.)

- for a Shine/Aux command – if Bit 6 of the Default Values Control Byte is clear, then all sub-fields of the Auxiliary Control Bytes in the Shine/Aux command are used, whether they contain a zero or a non-zero value. If however Bit 6 of the Default Values Control Byte is set, then any sub-fields which are zero in the Shine/Aux command will be taken from the Default Command Word instead; auxiliary control sub-fields that are not zero in the Shine/Aux command will be used (rather than the corresponding Default command Word sub-fields)
whatever the state of Bit 6 of the Default Values Control Byte. This allows a Shine/Aux command to default all Auxiliary behaviours except those explicitly coded (with non-zero values) in the Shine/Aux command itself.

This behaviour, although apparently complicated, allows the separate and extremely flexible control – either by defaulting or by explicit specification – of the Auxiliary functions of the IICU.

This field has not changed from the IICU V1 Functional Specification at Appendix D.

4.4.5 Command Table Fields – Auxiliary Control Byte 2

This field (shown at Figure 4.4.4 below) allows specification of the behaviour of the IICU’s two switched mains power outlets, and of the IICU Auxiliary TTL Outputs 1 and 2. All of these IICU outputs appear on the back panel of the IICU, via IEC and BNC connectors respectively.

<table>
<thead>
<tr>
<th>Bit No.</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>Switched Mains</td>
</tr>
<tr>
<td>6</td>
<td>Power Outlet A</td>
</tr>
<tr>
<td>5</td>
<td>Switched Mains</td>
</tr>
<tr>
<td>4</td>
<td>Power Outlet B</td>
</tr>
<tr>
<td>3</td>
<td>Auxiliary TTL</td>
</tr>
<tr>
<td>2</td>
<td>Output 1</td>
</tr>
<tr>
<td>1</td>
<td>Auxiliary TTL</td>
</tr>
<tr>
<td>0</td>
<td>Output 2</td>
</tr>
</tbody>
</table>

Figure 4.4.4 Auxiliary Control Byte 2, internal structure.

The Switched Mains Power outlets can be programmed to “On”; to “Off”; or to follow the state of the Minisys TTL Output A or B respectively. (See the preceding section for the available programmed behaviours of Minisys TTL Outputs A and B.)

The two Auxiliary TTL Outputs can be programmed to “Low” (Active); to “High” (Inactive); or to send an Active (negative-going) TTL pulse (of the duration set via the management interface). Note that these outputs are incorrectly labelled on the IICU back panel as “TTL Out C” and “TTL Out D”.

The defaulting of individual sub-fields of Auxiliary Control Byte 2 is handled in exactly the same manner as for Auxiliary Control Byte 1. Note that the single Bit 6 of the Default Values Control Byte governs the defaulting behaviour of both of the Auxiliary Control Bytes.

This field has not changed from the IICU V1 Functional Specification at Appendix D.

4.4.6 Command Table Fields – Shine Command Parameters

This field and the following field (Camera Control Parameters) are the two fields that have changed significantly with the implementation of IICU V2, and the descriptions that follow supersede those in the Functional Specification at Appendix D.
The Shine Command Parameters field is now (like the Camera Control Parameters) divided into two separately defaultable, and quite independent, sub-fields – the upper, or “Suppress LEDs” nibble; and the lower, or “Secondary Trigger” nibble. These are defined separately below.

The defaulting behaviours of these two sub-fields are now identical. The upper bit of each nibble (i.e. Bits 7 and 3 of the byte-long field) is designated as “Use Values Specified”. Each such bit interacts with Bit 7 of the Default Values Control Byte (the “Set/Use Default Shine/Camera Control Parameters”) in the following fashion:

- for a Set Defaults command – if Bit 7 of the Default Values Control Byte is set, then the entire contents of the Shine Command Parameters field (and also the Camera Control Parameters) are copied to the Default Command Word. But even if Bit 7 of the Default Values Control Byte is not set, then sub-fields (nibbles) of the Shine Command (or Camera Control) fields that have the “Use Specified Values” bit set are still copied to the Default Command Word; sub-fields without the “Use Specified Values” bit set will not be copied, leaving the corresponding Default Command Word sub-fields unchanged. Thus individual sub-fields of these Parameter fields may be defaulted independently of each other.

- for a Shine/Aux command – Bit 7 of the Default Values Control Byte is not consulted. If the “Use Specified Values” bit of either sub-field is set in the Shine command, then the corresponding sub-field is taken from the Shine command word, if the “Use Specified Values” bit is clear in the Shine command, then the corresponding field from the Default Command word is used instead. This allows a Shine/Aux command to default all Shine Command Parameters except those explicitly coded (with the “Use Specified Values” bit set) in the Shine/Aux command itself. [Note that in a Shine command, Bit 7 of the Default Values Control Byte is used to determine the behaviour of the shutter following the Shine command – see the description in Section 4.4.7 below.]

This defaulting behaviour differs from the description in the IICU Functional Specification of this field, which was not originally divided into two separately defaultable nibbles.

4.4.6.1 The Suppress LEDs Nibble

This nibble contains only two active bits: “Suppress LEDs ON”; and “Suppress LEDs OFF”. Each is quite simply defined; in an otherwise normal Shine command, then either the single step at which the LEDs are switched ON, or the single step at which the LEDs are switched off, is omitted.

The purpose of these bits is to replace – and to provide a more flexible alternative to – the old “Cancel Shine” and “Indefinite Shine” functions. These functions were late additions to the original IICU specification, and were implemented in a rather ad-hoc fashion that complicated both the Command Table structure, and the documentation thereof.

The Indefinite Shine function was provided to support experiments – and particularly those aimed at early testing of the PCIS – where it was desirable to leave the stimulation LEDs actively illuminating the sample at the end of an IICU command, while image capture operations were performed directly from the camera control software (rather than being controlled by the PCIS and the electrical
synchronisation facilities). This was particularly relevant during testing of the optical and electrical components before the PCIS automation (described in Chapter 5) was available; but it may also have applications in future PCIS experiments.

An “Indefinite Shine” was originally indicated by an explicit zero duration associated in the command with at least one active LED string – an admittedly ad-hoc method inconsistent with the rest of the IICU internal structure. A “Cancel Shine” (necessary to turn the LEDs OFF following an Indefinite Shine) was indicated by a Shine/Aux command with a “00” sub-command di-bit – i.e. a Shine/Aux command with neither Shine nor Auxiliary components specified. Both the Indefinite Shine and the Cancel Shine were “hard coded” with respect to many of their parameters, making them relatively inflexible, single-purpose functions.

By changing these ad-hoc functions to the current specification the inconsistencies with the design paradigm were removed; and the commands were made much more general, since any of the normal parameters to a Shine command may now be specified with any Indefinite Shine or Cancel Shine function. Indeed a command could be specified with both bits set; it would have no action on the stimulation LEDs, but would undertake all other (specified) timing and synchronisation functions of a normal Shine command.

Thus the new structure makes the Command Table logically more consistent; makes the two functions considerably more flexible; and makes the Command Table easier to construct and to use.

These bits did not exist at the time the original functional specification was written, and are not documented therein. This description of the “Suppress LEDs” bits is therefore definitive.

4.4.6.2 The Secondary Trigger Nibble

The lower nibble of the Shine Command Parameters (also separately defaultable by use of its own “Use Specified Values” bit) contains three bits that, together, fully define the way the PCIS responds to and uses the Secondary Trigger input(s). The three bits are:

- Use External Secondary Trigger;
- Secondary Trigger Wait; and
- Secondary Trigger Duration.

The Use External Secondary Trigger bit specifies that the external (IICU back Panel BNC connector) Secondary Trigger input should be used instead of the Minisys relay output assigned as the “normal” Secondary Trigger. Commands that delay the stimulation phase until a falling edge is detected on the secondary trigger (as described immediately below) will now monitor the back panel BNC input, rather than the relevant Minisys relay output. This allows external equipment associated with a PCIS experiment (such as the original Minisys liquid light guide source) to trigger the stimulation phase of a luminescence procedure.

Note that the External Secondary Trigger was not part of the original specification, and is not mentioned in Appendix D. It actually came about as a result of there being a spare BNC connector on
the back panel. (Indeed the External Secondary Trigger input is still labeled “SPARE”.) This in turn was a result of the BNC connectors being installed using “stacked pairs” of connectors (as a single component), so that inevitably there was an even number of connectors physically present. As the original specification called for an odd number of TTL inputs and outputs, there was obviously a spare connector. Since there was also a spare logical port address available, I asked for the “spare” connector to be wired up and programmed to operate as described here. This enhancement improves the flexibility with which complex experiments based on the PCIS may be designed and implemented.

The “Secondary Trigger Wait” bit specifies that the IICU should delay the stimulation phase of a Shine command (and also the relevant shutter and camera synchronisation functions) until the Secondary Trigger input (whether from the Minisys or External) is received. The PCIS will then run the stimulation LEDs for the duration specified (either in the Shine command, or in the Default Command Word, according to the value of Bit 5 of the Default Values Control Byte).

The “Secondary Trigger Duration” bit specifies that the stimulation phase should be delayed (as per the Secondary Trigger Wait”), but that the duration of the stimulation should then be controlled by the Secondary Trigger input remaining in the Active state – so that the Secondary Trigger input controls both the start and the duration of the Shine. All duration fields – whether in the command or in the default Command Word – are ignored in this mode. This mode would normally be used to allow the Minisys to take control of stimulation timing, as well as heating rate of the sample, during a combined TOSL run.

If both the “Secondary Trigger Wait” and the “Secondary Trigger Duration” bits are set, then the “Rapid Repeat Shine” mode is invoked.

4.4.6.2.1 Rapid Repeat Shine Mode

In the Rapid Repeat Shine mode both the duration of the Secondary Trigger input, and the duration specified in the command word are relevant. Multiple stimulations and image accumulations are undertaken, each for the duration specified in the command’s Duration field. The sequence of shines and accumulations is continued as long as the Secondary Trigger Input in use remains Active.

The purpose of this mode is to allow the sequence of accumulations to occur – possibly while a temperature ramping is happening – with the minimum possible “dead time” between accumulations (during which a sample may be continuing to emit signal, despite the lack of optical stimulation). This mode allows the next accumulation to begin as soon as the camera reports that it is ready (via the Camera Ready Input), without the extended serial communications delays (of the order of half a second) that are involved if the next accumulation requires the setting up and issuing of a new IICU Shine command.

A new stimulation stage is undertaken if the Secondary Trigger input is still Active when a falling edge is detected on the Camera Ready input. The accumulation, once commenced, will continue for the specified duration, even if the Secondary Trigger goes Inactive during that shine stage. Thus all recorded accumulations are collected over the correct, specified time period.
The number of accumulations actually recorded is communicated back to the PCIS via an IICU output port that can be specified via the IICU management interface. (The selected port is toggled at the start, and half way through, each Shine. Because of the serial communications delays associated with the polling of port states, the Rapid Repeat Shine mode should not be used with exposure durations of less than 1 second.) The state of the selected port will be re-routed by cable to the IICU pass-through back panel connector marked "Minisys Input 8". Special routines in the PCIS Runtime code monitor the corresponding optically-isolated Minisys input (Optical Input 9), and record the correct number of accumulated images to the PCIS Log files (see Chapter 5).

The Rapid Repeat Shine mode can of course be used with no stimulation LED strings specified, to support the effective and timely gathering of pure TL data. The Rapid Repeat Shine mode did not exist when the Functional Specification at Appendix D was written, and it is not referred to in that document. This description is therefore definitive.

4.4.7 Command Table Fields – Camera Control Parameters

The overall structure of the Camera Control Parameters field, and the way in which its sub-field defaulting behaviour is managed, have not changed from the definition in the Functional Specification at Appendix D. The contents of the second sub-field however, and its operation, have changed significantly, and are described below.

The two sub-fields in IICU V2 are the Shutter Behaviour nibble, and the Camera Synchronisation nibble. Each is separately defaultable via its own "Use Specified Values" bit. Note however that the defaulting behaviour of the Shutter Behaviour nibble in a Shine/Aux command is unique, and is described below. Otherwise, the defaulting behaviour of this field (for both Set Defaults and Shine commands) is as described above for the Shine Command Parameters field and its sub-fields.

4.4.7.1 Shutter Behaviour Nibble

The three active bits of the Shutter Control nibble specify one of eight available states or behaviours of the Shutter. These allow the shutter behaviour to be linked to the IICU Shine activities; to be linked to the camera system activities (using the "Shutter In" state information); or to be set to an Open or Closed state. The eight available values of the Shutter Control nibble are:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>Closed</td>
</tr>
<tr>
<td>001</td>
<td>Follow Shutter Drive Input</td>
</tr>
<tr>
<td>010</td>
<td>Inverse of Shutter Drive Input (Override Mode only)</td>
</tr>
<tr>
<td>011</td>
<td>Open (available in Override Mode only)</td>
</tr>
<tr>
<td>100</td>
<td>Open for Shine duration only</td>
</tr>
<tr>
<td>110</td>
<td>Shine duration + Latency before</td>
</tr>
<tr>
<td>101</td>
<td>Shine duration + Latency after</td>
</tr>
<tr>
<td>111</td>
<td>Shine duration + Latency before and after</td>
</tr>
</tbody>
</table>
The four “Shine duration” shutter modes (100 to 111) cause the shutter to be open while the LEDs are active. They also allow the shutter to be opened just before the stimulation starts (to allow for the physical shutter-opening time); to remain open briefly after the stimulation stops (to capture any slow-release photons from the sample); or both. The duration of the two time extensions (called “Latency Before” and “Latency After”) may be set via the IICU management interface.

Note that two of the shutter modes — “Inverse of Shutter Drive Input” and “Open” — require that the IICU be in Over-ride Mode when they are specified. These modes would only be required in unusual PCIS experiment set-ups.

The behaviour of the shutter after the stimulation period controlled by the Shine command is determined by the value of Bit 7 of the Default Values Control Byte (the “Set/Use Default Shine/Camera Control Parameters” bit). In a Set Defaults command this bit is used, as described above, to control the sub-field or full field setting of Default Command Word values.

In a Shine command, however, the use of default values versus values specified in the command is controlled solely by the “Use Specified Values” bit of each sub-field (as described above for the Shine Control Parameters field). Bit 7 of the Default Values Control Byte now (if set) specifies that at the end of the stimulation phase of the Shine command, the shutter should be restored to whatever state or behaviour is specified in the Default Command Word. If Bit 7 of the Default Values Control Byte is clear, then the shutter is left in whatever state or behaviour was specified in the Shine command for the stimulation phase of the command.

Bit 7 of the Default Values control byte would normally be set in a Shine command, so that at the end of the command the shutter is returned to its default state (which is normally “Closed”). However, in an Indefinite Shine command, where data collection is expected to continue past the end of the command execution by the IICU, this bit would normally be clear, leaving the shutter in whatever state was specified in the Shine command so that further accumulations may be acquired.

4.4.7.2 Camera Synchronisation Nibble

The Camera Synchronisation nibble comprises three bits (plus the Use Specified Values bit):

- Send Trig
- Send Sync
- Wait Camera Ready

“Send Trig” being set in a Shine command causes the IICU to send a negative-going pulse to the “Trigger” input of the ST-138 camera controller. This will cause the next accumulation to be stored by the camera system (implementing a PCIS image capture) rather than being discarded (which would implement a CCD surface clean).

“Send Sync” being set in a Shine command causes the IICU (at the end of the stimulation phase, if any) to send a negative-going pulse to the “Sync” input of the ST-138, causing it to immediately collect the image data from the CCD (for storage or discard as determined by the “Send Trig” bit).
"Wait Camera Ready" being set causes the IICU to wait for a falling edge signal on the “NotScan” output of the ST-138 (indicating that the read-out is complete and the camera is ready for a new acquisition), before completing the IICU command and clearing the “Command Active” status bit to the Minisys. (This bit must be set in Rapid Repeat Shine mode, where it forces the IICU to wait for the NotScan signal before commencing the next accumulation.)

These bits were moved from the Shine Command Parameters with the development of IICU V2, and replaced the earlier fields specifying the nature of the state or transition required on the Camera Ready Input (Low, High, Falling Edge or Rising Edge). These proved to be unnecessary (only Falling Edge was ever used), and so were removed as part of the IICU V2 “rationalisation”.

The setting and use of default values for this sub-field is controlled by the sub-field’s own “Use Specified Values” bit, and by Bit 7 of the Default Values Control Byte, in the manner described above for the Shutter Behaviour nibble.

### 4.4.8 Command Table Fields – Duration of Current Drive

This field contains a 16-bit value specifying the duration of the stimulation phase of a Shine command in tenths of a second. The available range is thus from 0 seconds to 6,553.5 seconds (over 100 minutes). Its defaulting behaviour is determined by Bit 5 of the Default Values Control byte.

This field has not changed from the IICU V1 Functional Specification at Appendix D.

### 4.4.9 Command Table Fields – Current Drive Numbers 1 to 4

These four fields specify the drive currents to be used with each of the four LED strings in a Shine command. Each field is 12 bits long, and specifies a current between 0 mA and 409.5 mA in 0.1 mA increments. The defaulting behaviour of these fields, both for Set Defaults and Shine commands, is controlled by Bits 0 to 3 of the Default Values Control Byte.

Each field sits within a 16 bit region of the Command Word, so that there are four “spare” bits associated with each current specification. These could potentially be used in the future to support the implementation (and specification) of, say, ramped LED drive currents for the PCIS. Any such use would of course require enhancements to the IICU firmware, as well as the development of new Command Table Sets to invoke the new behaviours.

Note that in a Reset command, these fields contain Soft Maximum Current Limits which the LED strings must not exceed (unless the IICU is in Over-ride Mode), rather than Default Current settings. Default Currents for the four LED strings must be set using a Set Defaults command. (Hard Maximum Current Limits – which may not be exceeded even in Over-ride Mode – are set via the IICU Management Interface.)

These fields have not changed from the IICU V1 Functional Specification at Appendix D.
4.5 The IICU Management Interface

The Management Interface as implemented for the IICU V2 is quite different from that described in the original IICU Functional Specification, and is documented in detail below.

Although the style of the interface is simpler than that in the specification, the overall level of functionality is very similar. It supports the maintenance of PCIS operating parameters, and the inspection of many of the IICU internal states and variables (including the internal Default Command Word), as well as providing for the upload and download of IICU Command Tables.

4.5.1 The Management Interface Paradigm

The IICU management functions are implemented via a simple command-line interface over the RS-232 serial port on the IICU back panel (labelled "SERIAL"). This interface is accessed by a PC running a terminal emulator program – HyperTerm has been used.

This approach – initially developed by Daniel Cummins in the Electronics workshop simply to support loading command tables into the IICU during the IICU firmware development – was eventually adopted as the basis for the complete IICU management facility. Its main advantage is that it avoids the need to develop a specific application to run on the management PC. Its use also minimised the development required in the IICU firmware to support the required management functions.

The serial interface supporting the management functions is set to run at 9600 baud asynchronous, 8 data bits, No parity and 1 stop bit. No flow control is used. All data transfer is in ASCII, using CRLF as the line terminator. All management interface commands start with the “#” character.

4.5.2 Uploading and Downloading Command Tables

The basic (and original) management commands are “#P” to Put a command table to the IICU; and “#G” to Get a command table from the IICU. Either command will elicit a response, asking for a number from one to eight to be entered, specifying the particular command table (from Slot A Primary, Slot A Secondary ... through to Slot D Secondary) to be Put or Got.

In a Put operation, the ASCII hexadecimal representation of the entire command table data is then "Cut" or "Copied" from the spread-sheet under which the tables are developed (see Section 4.7 below), and "Pasted" into HyperTerm (or similar application), which then sends it to the IICU.

The Get operation causes the IICU to return (to the HyperTerm application) the same ASCII hexadecimal representation of the selected command table, as 64 CRLF-separated lines of text. The Get operation is less useful than the Put, since the retrieved data cannot (for technical reasons to do with how the data is calculated) be pasted back into the table development spread-sheet. The Get is used primarily to confirm that the preceding Put has worked correctly.
4.5.3 Setting and Reporting IICU Internal Parameters

The IICU management interface allows a number of important IICU operating parameters to be examined and set. These comprise the Hard (LED) Maximum Current Limits; various timing parameters; and the Rapid Repeat Shine Reporting Port.

In each case the reporting is invoked by a command of the form "#x?", where x is a character representing the parameter to be reported. Setting of parameters is invoked by a command of the form "#xnnnnn", where x is as for a reporting command, and nnnnn represents the new value to be given to that parameter.

4.5.3.1 Hard LED Maximum Currents

As described earlier, the Reset command contains the Soft Maximum Currents for each of the LED strings, which cannot be exceeded by any Shine (or relevant Set Defaults) command unless Over-ride Mode is set. The management interface supports the setting of Hard Maximum Current Limits for each LED string, that cannot be exceeded by a Shine or Set Defaults command even when the IICU is in Over-ride Mode.

The soft current limits are set to preserve the long-term performance and stability of the LEDs, and are currently set at 80% of the rated maximum current of the LEDs in each string. The hard limits are intended to prevent any immediate damage to the LEDs, and are currently set equal to the LEDs’ rated maximum currents.

The command ID characters used are I (for LED string 1), J, K and L (for LED string 4). The currents are set in tenths of a milliamp, from 0 to 409.5 mA (i.e. “nnnn” can range from 0 to 4095). For example:

- #I?  - will report the present Hard Max Current setting for LED String 1
- #K1500  - will set the Hard Max Current for LED String 3 to 150 mA
- #L4096  - will be rejected as a command (and cause no change to the internal state of the IICU), since the requested current value is outside the allowed range.

4.5.3.2 IICU Timing Parameters

The format of the IICU timing parameter commands is the same as that for the hard current limits, except that the values are now reported and set in tenths of a millisecond, rather than tenths of a milliamp. The values may range from 0 or 1 (0.0 or 0.1 milliseconds) through to 30000 (3,000 milliseconds, or 3 seconds). The values available for reporting and setting, and their corresponding command ID characters, are as follows.
• T – reports or sets the pulse width used by the IICU for all outgoing TTL pulses (camera synchronisation and auxiliary outputs). Value may be 1 (0.1 ms) to 30000 (3.0000 seconds).

• Q – reports or sets the “Quiet time” (the Pause) between the Aux and Shine phases of a fully-specified Shine/Aux command. Value may be 0 (0.0 ms) to 30000 (3.0000 seconds).

• B – reports or sets the Latency Before the shutter opens in relevant shutter modes (see Section 4.4.7.1). Value may be 0 (0.0 ms) to 30000 (3.0000 seconds).

• A – reports or sets the Latency After the shutter closes in relevant shutter modes (see Section 4.4.7.1). Value may be 0 (0.0 ms) to 30000 (3.0000 seconds).

### 4.5.3.3 Reporting and Setting the Rapid Repeat Shine Reporting Port

When images are acquired one at a time in response to PCIS commands, the PCIS runtime software is able to record each one to the PCIS Log file (see Chapter 5). When the Rapid Repeat Shine mode is invoked, however, the PCIS runtime software does not know how many images will be recorded, and so is unable to keep the Log file up to date with the recorded image numbers.

To address this problem, the Rapid Repeat Shine mode of the IICU cycles a TTL Port for each image that is acquired. The IICU cycles the selected port for half of each acquisition duration (i.e. sets the port low half way through the acquisition, and high again at the end of it). This ensures that the TTL signal remains stable in each state long enough for the PCIS runtime software to reliably poll for its state, and record the acquired images (as long as each acquisition is for a second or more).

Because the Rapid Repeat Shine Mode was a late addition to the IICU functionality, no TTL signal was created and allocated to this purpose, and the signal has to be “overlaid” on one of the existing IICU TTL outputs. Since I was unable to decide which TTL port the signal should be overlaid on (not being able to predict all possible experimental configurations of the PCIS, and therefore not knowing which TTL outputs might already be in use), I elected to implement the following solution:

- The IICU TTL Output which would be cycled as each image was acquired would be user-selectable, and would be configured via the IICU Management Interface.
- The selected TTL Output would then be connected to a fixed Minisys optically-isolated input via the loop-back connectors on the back panel of the IICU, using a short BNC co-axial cable. The first available Minisys TTL input (referred to as Minisys Input 9 in the Minisys command language but unfortunately labelled “Minisys Input 8” on the IICU back panel) was selected.
- The PCIS runtime software is then written to poll the state of logical Minisys Input 9, and records an image to the PCIS Log file on each falling edge detected.

The IICU Output ports that may be selected for this signal to be “overlaid” on, and the Management Interface code letter used for each, are:

A Minisys TTL Output A

B Minisys TTL Output B
Note that simply cycling the selected output port at the start and end of each acquisition would only work if the camera system read-out time (the time between acquisitions) was guaranteed to be greater than half a second or so. This would not be the case if, for a high brightness but time-critical experiment, the read-out rate of the CCD was set to a faster value. (This trade-off was discussed in Chapter 2.) That is why the half-duration cycling of the output port was specified. However, if both the acquisition and read-out times will be greater than half a second (the normal situation), then a signal that cycles at the start and end of each acquisition is perfectly adequate. Such a signal is available from the ST-138 NotScan output, or from the Shutter State Monitor output on the shutter controller. In this case, the reporting port can be set to “N” (for “None”), and the NotScan or shutter monitor output may be connected directly to the TTL port labelled “Minisy Input 8” on the IICU back panel.

### 4.5.4 Reporting IICU Internal State Variables

The IICU management interface can also report the state of various internal state values that it would not make sense to set. For these commands, the only available form is the “#x?”, where x is the command ID character as listed below.

- **C** - reports the Command Number of the last IICU command executed, along with the Command Table Number, and whether the command was from the Primary or Secondary table of the identified Command Table Set.

- **E** - reports the Error Number of the last IICU Error that has occurred (if any since the latest IICU Reset). Also reports the Command Number of the IICU command being executed when the Error occurred, along with the Command Table Number, and whether the command was from the Primary or Secondary table of the identified Command Table Set.

- **W** - reports the Warning Number of the last IICU Warning that has occurred (if any since the latest IICU Reset). Also reports the Command Number of the IICU command being executed when the Warning occurred, along with the Command Table Number, and whether the command was from the Primary or Secondary table of the identified Command Table Set.

- **D** - reports the complete contents (as an ASCII Hexadecimal screen dump) of the current internal Default Command Word. This is useful for the testing and debugging of Set Defaults commands, and of Shine/Aux commands that use the stored default values.

- **S** - invokes the reporting of all of the above status values, as though the user had issued #C, #E, #W and #D as a single management interface command line.
### 4.5.5 IICU Management Interface Command Summary

The following table summarises all of the commands available on the IICU management interface.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#P</td>
<td>Put Command Table. Follow prompts and enter Command Table number to be Put.</td>
</tr>
<tr>
<td>#G</td>
<td>Get Command Table. Follow prompts and enter Command Table number to Get.</td>
</tr>
<tr>
<td>#I?</td>
<td>Reports the present value of HardMaxCurrent1 (x 0.1 mA)</td>
</tr>
<tr>
<td>#Innnn</td>
<td>Sets the value of HardMaxCurrent1 to ( nnn.n ) mA (from 0 to 409.5)</td>
</tr>
<tr>
<td>#J?</td>
<td>Reports the present value of HardMaxCurrent2 (x 0.1 mA)</td>
</tr>
<tr>
<td>#Jnnnn</td>
<td>Sets the value of HardMaxCurrent2 to ( nnn.n ) mA (from 0 to 409.5)</td>
</tr>
<tr>
<td>#K?</td>
<td>Reports the present value of HardMaxCurrent3 (x 0.1 mA)</td>
</tr>
<tr>
<td>#Knnnn</td>
<td>Sets the value of HardMaxCurrent3 to ( nnn.n ) mA (from 0 to 409.5)</td>
</tr>
<tr>
<td>#L?</td>
<td>Reports the present value of HardMaxCurrent4 (x 0.1 mA)</td>
</tr>
<tr>
<td>#Lnnnn</td>
<td>Sets the value of HardMaxCurrent4 to ( nnn.n ) mA (from 0 to 409.5 mA).</td>
</tr>
<tr>
<td>#T?</td>
<td>Reports the TTL Pulse Width used by the IICU outputs (x 0.1 ms).</td>
</tr>
<tr>
<td>#Tnnnnn</td>
<td>Sets the TTL Pulse Width to ( nnnnn.n ) milliseconds (0.1 to 3000.0 ms).</td>
</tr>
<tr>
<td>#Q?</td>
<td>Reports the Pause between the Aux and Shine phases of a Shine/Aux (x 0.1 ms).</td>
</tr>
<tr>
<td>#Qnnnnn</td>
<td>Sets the Pause between Aux and Shine to ( nnnnn.n ) milliseconds (0.0 to 3000.0 ms).</td>
</tr>
<tr>
<td>#B?</td>
<td>Reports the Latency time used Before Shutter Opens (x 0.1 ms).</td>
</tr>
<tr>
<td>#Bnnnnn</td>
<td>Sets the Latency time Before Shutter Opens to ( nnnnn.n ) ms (0.0 to 3000.0 ms).</td>
</tr>
<tr>
<td>#A?</td>
<td>Reports the Latency time used After Shutter Closes (x 0.1 ms).</td>
</tr>
<tr>
<td>#Atnnnn</td>
<td>Sets the Latency time After Shutter Closes to ( nnnnn.n ) ms (0.0 to 3000.0 ms).</td>
</tr>
<tr>
<td>#R?</td>
<td>Reports the Rapid Repeat Shine Reporting Port setting (A, B, G, H, X, Y or N).</td>
</tr>
<tr>
<td>#Rx</td>
<td>Sets the Rapid Repeat Shine Reporting Port setting to ( x ) (A, B, G, H, X, Y or N).</td>
</tr>
<tr>
<td>#C</td>
<td>Reports the Last Command No. executed; its Command Table No., and P or S.</td>
</tr>
<tr>
<td>#E</td>
<td>Reports the Last Error No. to occur; its Command No., Table No., and P or S.</td>
</tr>
<tr>
<td>#W</td>
<td>Reports the Last Warning No. to occur; its Command No., Table No., and P or S.</td>
</tr>
<tr>
<td>#D</td>
<td>Reports the entire current internal Default Command Word (in hexadecimal).</td>
</tr>
<tr>
<td>#S</td>
<td>Reports all Status information reported by #C, #E, #W and #D.</td>
</tr>
</tbody>
</table>
4.6 IICU V2 – Summary of Changes from the V1 Functional Specification

The major changes affecting the functionality of the IICU between the original specification and the final IICU V2 implementation have already been detailed throughout the previous sections. This section reviews those changes, and lists the minor differences between the specification and IICU V2 that have not previously been discussed. It thus provides a single reference that, read in conjunction with the specification at Appendix D, provides a complete description of the IICU V2 functionality.

4.6.1 Suppress LEDs ON and OFF Bits

The Suppress LEDs On and Suppress LEDs Off bits are in the Suppress LEDs Nibble of the Shine Command Parameters Command Word field, described in Section 4.4.6.1 above. These bits provide a more logical and more general ability to exit an IICU Shine command with the LEDs left on – and later to turn those LEDs off – than the earlier “Indefinite Shine” and “Cancel Shine” operations included in the original specification.

Designed primarily to support testing of the PCIS before the full software automation described in Chapter 5 was available, the more flexible implementation in IICU V2 may also find a role in future more complex PCIS experiments, such as in investigating the bleaching behaviours of minerals.

4.6.2 Rapid Repeat Shine Mode

The Rapid Repeat Shine mode is functionality that is provided to compensate (to some degree) for the slow readout rates required to achieve the desired very low noise levels (and thus high sensitivity) from the LN-CCD. It is invoked by settings of the Secondary Trigger Nibble in the Shine Command Parameters field, described in Section 4.4.6.2 above.

This mode causes the IICU to perform repeated Shines, each of the length specified as the Shine Duration, for as long as the Secondary Trigger is Active. Each succeeding Shine is performed as soon as the previous Shine has completed, without the need for the PCIS runtime software to issue any new IICU commands. This saves around half a second or more between each pair of Shines, minimising any signal lost from a sample that is still hot, or even still being heated, during the “dead” period between accumulations. Its main application is to TL or TOSL runs, rather than to normal OSL runs performed at moderate temperatures.

No equivalent of this functionality was included in the original IICU specification.
4.6.3 External Secondary Trigger Support

This feature was added during the development of the IICU, but its use was only properly established with the re-arrangement of the Shine Command and Camera Control Parameter fields in IICU V2. Its implementation was largely motivated by my noticing there was a “Spare” BNC connector on the IICU back panel, and realising that programming it as an alternate Secondary Trigger input would place the source of that signal under software control, and avoid the use of messy split cables if an external source of the signal was required in future experimental setups.

The operation of the External Secondary Trigger when it is selected (using a bit in the Secondary Trigger Nibble described in Section 4.4.6.2) is exactly the same as the operation of the internal (Minisys-sourced) Secondary Trigger in “normal” operations. The function of the Secondary Trigger input is to control the timing of the optical stimulation (and image accumulation) phase of a Shine command, and has already been described.

No provision for an External Secondary Trigger was included in the original IICU specification.

4.6.4 Management Interface Changes

The IICU Management Interface as finally delivered with the IICU V2 (described in detail in Section 4.5 above) differs quite significantly in form and description from that originally envisaged, and described in the Functional Specification. The total functionality, however, is not very different.

The changes were prompted mainly by the need to minimise the development effort from the Electronics workshop late in the IICU development when, with no extra funds available, I was negotiating the implementation of the principal IICU V2 upgrades described above.

Accordingly it was decided to specify a new set of Management Interface requirements based on the interim software that Daniel Cummins had already built to support the development and debugging of the IICU firmware. This was a basic command line interface accessed by running a terminal emulation program (such as the HyperTerm application supplied with Windows) on the management PC. The interface supported only two commands — “G” to Get a table from the IICU; and “P” to Put a table to the IICU. This simple interface style was extended with a range of additional commands to support the setting and reading of a wider range of PCIS parameters and state variables.

The original specification called for the ability to have some internal IICU values displayed with constant “live” updating of the value while the IICU executes a command. This implied some sort of windowed interface, rather than the simpler command line interface finally adopted. This facility was not necessary (as the value could simply be read again by the user), and so was discarded.

The “Debug Mode Control Functions” included in the original specification were also sacrificed in the name of reduced development effort, as these functions could be met almost as easily using the facilities of the IICU Bench Testing Environment (described in Section 4.8 below), including the hardware Minisys Simulator that had by now been built.
4.6.5 Minor Changes from the Specification.

The IICU V2 as finally implemented differs from the original specification in a number of minor details that do not significantly affect the functionality, as the changes already described have done. These minor changes are still relevant to a complete understanding of the final IICU, and in particular to the maintenance of the IICU firmware or the construction of new Command Tables.

4.6.5.1 LED Current Driver Specifications

In the original specification the LED current drivers were specified with individual current and voltage specifications according the LED string they were intended to drive. During development of the IICU circuitry we decided to implement four identical current drivers, each capable of meeting the requirements of any LED string. This preserved flexibility for future upgrades of the source turret and the stimulation LEDs; but it also simplified the design and construction of the IICU.

At the same time the somewhat arbitrary association of LED strings to current drive numbers in the original specification was changed to the more logical (and more easily remembered) paradigm of associating higher drive numbers with higher energy photons – so that Drive 1 is the IR LEDs; Drive 2 is red; Drive 3 is blue; and Drive 4 is UV.

4.6.5.2 Simplified Reset Command Behaviour

As described in Section 4.4.1.3 above, the Reset command was originally specified as being able to execute up to three IICU commands following it in the Command Table, to support extended initialisation after a Reset. This feature was not implemented, and any necessary initialisation of the IICU is now performed by the PCIS runtime software issuing a series of IICU commands. This is a simpler, and in many ways more flexible solution, since different luminescence protocols can now issue different sets of initialisation commands. The small time delay inherent in issuing the extra commands is of no consequence at the start of a run when a Reset is normally performed.

4.6.5.3 Camera Ready Transition Required Field Not Implemented

The original specification allowed a Shine command to specify the nature of the state or transition on the Camera Ready input (High, Low, Rising Edge or Falling Edge) that the IICU would interpret as meaning the LN-CCD was ready for the next accumulation. This was done as a safeguard against any possible misinterpretation of the various ST-138 signal timing diagrams on which the PCIS hierarchical architecture was based.
During initial PCIS testing it was confirmed that the only value required was “Falling Edge”, so this value was “hard coded” in the firmware, and the field was removed from the Command Word structure. It was this change that gave the flexibility and space in the Command Word to allow the principal changes that comprise the IICU V2 upgrade to be made.

4.6.5.4 Turret/Shutter Interlocks

The original specification called for there to be light level detectors both in the source turret and in the shutter housing, as well as micro-switches on both units to detect whether the lens is down, and the optical path sealed from external light entry. These are used via an interlock arrangement to prevent the shutter from opening (and the LN-CCD possibly being saturated and damaged) if the PCIS is not assembled correctly, or if there is excess light over the shutter for some other reason (such as an incorrect optical filter pack for the stimulation LEDs being used). In the end only the light detector in the shutter housing, and the micro-switch on the source turret, were implemented.

If the light levels are safe for the LN-CCD immediately above the shutter, then the light levels in the source turret (below the optical filter pack) are somewhat irrelevant. Indeed, even though the source turret detector was specified as part of the shutter interlock to prevent CCD damage, its main purpose in reality would have been to provide positive evidence (should there ever be any doubt) that the stimulation LEDs were working. Since the IICU front panel displays the true current flowing through the LEDs (not just the current requested), and since LEDs do not pass current (at any reasonable voltage) unless they are re-emitting the energy as photons, the current display provides the positive confirmation required. There was therefore no real loss of functionality in not installing the source turret light level detector.

Since it is physically impossible to lower the lens onto only one or the other of the source turret and shutter housing, there was no real need to have two micro-switches to confirm proper PCIS assembly. The source turret provided an easier location than the shutter housing to mount and connect a micro-switch, and so the micro-switch on the shutter housing was omitted.

The interlock between these two detector types and the shutter drive was originally specified as a hardware interlock, so that it could not be defeated by software errors. The interlock was eventually implemented in the IICU firmware instead, but (now that the relevant code has been fully tested) it provides the same level of protection against operator errors as would an interlock in hardware.

4.6.5.5 Shutter Housing Connector

The Shutter Housing connector was originally specified as a D9, consistent (but not confusable) with the D15 connector used for the Source Turret. In the end a 7-pin Mini DIN connector was used instead, to avoid confusion with the D9 connector used for the RS232 serial connection to the management PC.
Apart from the unavoidable plethora of BNC connectors for TTL signals, there are no two identical connectors on the back panel of the IICU. This is particularly important for a machine that has many cables, and is generally assembled in extremely low light conditions.

4.6.5.6 Switched Power Outlets

The original specification calls for physical On/Off switches on each of the (software-switched) mains power outlets. This was not implemented as (a) the IEC connectors (which would fit the back panel design) have no such switches, and (b) it was a silly idea in the first place, more likely to lead to failed runs (by an outlet being unintentionally switched off) than to serve any useful purpose.

4.6.5.7 Back Panel TTL Port Labelling

For reasons I don’t understand, the naming of the General Purpose and Auxiliary TTL ports changed during the physical design of the IICU and its housing, and appears quite differently on the IICU back panel than in the specification.

The mapping between the names used in the specification (and in this thesis), and the labels that appear next to the connectors on the IICU, is as follows:

<table>
<thead>
<tr>
<th>Functional Specification</th>
<th>IICU Back Panel Labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>General Purpose TTL Input A</td>
<td>AUX IN A</td>
</tr>
<tr>
<td>General Purpose TTL Input B</td>
<td>AUX IN B</td>
</tr>
<tr>
<td>General Purpose TTL Output A</td>
<td>AUX OUT A</td>
</tr>
<tr>
<td>General Purpose TTL Output B</td>
<td>AUX OUT B</td>
</tr>
<tr>
<td>Auxiliary TTL Output A</td>
<td>AUX OUT C</td>
</tr>
<tr>
<td>Auxiliary TTL Output B</td>
<td>AUX OUT D</td>
</tr>
</tbody>
</table>

Note also that the Shutter Housing Mini-DIN connector is labelled “LENS”; and that the back panel BNC connector labelled “SPARE” is now the External Secondary Trigger TTL input, supporting new functionality introduced in IICU V2 and described above in Section 4.6.3.

4.6.5.8 Execution of Reset[0] Command

The specification called for the IICU to be able to execute Command 0 (which would always be a Reset command), even if another IICU command was active. This was not implemented, due to the complexity it would have added to the IICU firmware. Any “locked up” instruction (such as one waiting for an external event that has not occurred), or an IICU Error condition, must now be cleared by a Hard Reset – i.e. by pressing the Reset button on the IICU front panel (or, alternatively, by powering the IICU down and back up).
4.7 The IICU Command Tables Sets

This section describes the tools used for creating and maintaining IICU Command Table Sets; the conventions applied to their design and construction; the various Command Table Sets that have been built and used to date; and Command Table Set 3 that is currently under construction.

4.7.1 Creating and Maintaining IICU Command Tables

The basic tool used to create and maintain IICU Command Tables is an Excel spread-sheet that represents the IICU Command Word as described in Section 4.4, and allows a complete table of 64 command words to be created. The original spread-sheet was written by Daniel Cummins of the Electronics workshop, but has been modified for easier use and better readability by me.

While it is possible to make small changes to a command table with little specialist understanding beyond the information contained in this section, the actual construction of new command tables requires an extremely detailed understanding of the entire IICU specification, as well as of the Minisys command language, and of the anticipated operations and requirements of the PCIS.

This section describes the mechanics of creating and installing IICU Command Table Sets, and also the design and contents of the currently-installed Command Table Set (Set Number 2) which supports all of the present automated PCIIS protocols described in Chapter 5. The emerging design of Command Table Set Number 3, which will take full advantage of the more flexible operations made possible by the IICU V2 updates, is also briefly described.

4.7.1.1 Command Table Creation – the Excel Spread-sheet

The IICU Command Table Creation spread-sheet allows the individual fields of each command word to be entered, generally either as single bits (0 or 1), or as decimal numbers (e.g. shine duration and LED currents). Some multi-bit code fields are entered by a pull-down list (e.g. Shutter Behaviour, one of eight named values); others (the Sub-command di-bit and the Auxiliary control sub-fields) do not, and the correct binary value must be known and entered.

The calculated parts of the spread sheet (all of those not on a white background in the figures that follow – all fields with a white background are user-entered) accumulate all of the entered values into hexadecimal fields, and finally into a single column (the wide purple column in Figure 4.7.1) that contains the entire command table as sixty-four 128-bit numbers, represented by ASCII hexadecimal characters. This is the column that, once the command table is complete, is copied and pasted to the IICU during the table installation process (as described earlier in Section 4.5.2).

The leading section (the first few columns) of the Primary Command Table Number 2 (the current Primary Command Table) is shown in Figure 4.7.1 below. The entire Primary and Secondary tables of Command Table Set 2 appear at Appendix E.
Main Command
bits
1

,q

0~
01

"'Tl
(JQ

6

05

6

06

7

07

600FOOOOA88BOOOOOOOOOOOOOOOOOOOO
00000000000000000000000000000000
60BF0000898800000000000000000000

C
--,

ro

Shine/Aux
NoCommand
Sh in e/Aux
SetDefau Its
SetDef au Its

Ori
00
01

I
I

-w

+:>,
-..J
fJ

n
n

13

14
U l lt llllCI' • U . I U IICl' \. I l'IU V YJl\,. r\. llDO I.UI V , 1,11110 1.- ....-1 11,.
1

001111011
00'\1110

,v

...., ,..,,.,..,., ...

17
18
19

o:iooo·I
0100'10
011 00~n

OD 60BF 00'00 000B 0000 0000 0000 OO(JO 0000
OE 60Eii" 0'000 OOOE 0000 0000 00.00 0000 0.0'00

..., ...... ..,,.,,

A
60BFOOOOOOOBOOOOOOOOOOOO.OOOOOOOO
60BFOOOOOOOEOOOOOOOOOOOOOOOOOOOO

Sh in e/Aux

..

..,..,..,, ..,.., ..................... ..,,.., .., ,..,,,,...,.., ..,,.,,,.,..,,..,..,,,,.,,,,..,.,,..,,..,

11 609F ODOO 1OOC 000 11 0000 0000 0000 0-000
12 609F 0000 OOOA 000 11 0000 0000 0000 00'00
13 609F 0000 000B 000 I\ 0.0'00 0000 0000 0000

....,, .,,,..,,,,.... "'

609FOOOO 1OOC00010000000000000000
609FOOOOOOOA00010000000000000000
609FOOOOOOOB00010000000000000000

Shin e/Aux
Shin e/Aux
Sh in e/Aux

,v

..,..,

...,..,..,

...

,.,..,.., ., '-'"· "

..,..,,...,.., ..,..,,

.....,.,..,

CJ

n

3

3

0..

01
0~
01

-f
CJ

CJ"

01
01
0,1

N

-u
--,

n
on
01
0~
0~

3

CJ

-<
ro

CJ

0..
::J
(JQ
--+,

ro
~

n

Oil
0'1
1
0~
0 11
OIL
0

s, D
ne - . ec ,
C , e ore , estt
Shine - NO LEDs, 0.1 Sec , Restore, Rest-Ont
Shine - ALL DEFAULTS , Do NOT Restore Shutter behavi our
Shine - ALL DEFAUL TS Restore Shutter behaviour

Aux - M/Sys TTL Out A - Latch, Falling Edge GP TTL In A
Aux - M/Sys TTL Out A - Follow GP TTL In A
Aux - M/ s TTL Out B - Latch , Fallin Ed e GP TTL In B

"1J

ro
N

u,
N

I
I
I
I

011
on
Af

011
ux - ower
e
Aux - Power Outl et B - ON
Aux - Power Outl et B - OFF
Aux - GP TTL Out C - HI GH
ux .
u
Aux - GP TTL Out C - SEND PULSE
Aux . GP TTL Out D . HI GH
Aux · GP TTL Out D - LOW
ux .
u
Toggl e Ove rride
Switch to Secondary
Switch to Prima

0·11
01
·1
Oil
OR
on

I In I

61
62
63

11,1111011 3D 8000 0000 0000 0000 0000 0000 0.000 0-000
ln-11(10 3E 9000 OOQO 0000 000.0 0000 0000 0_000 0000
'L1'1 1t1 rtJ 3F AOO.o 0000 0000 oodo_oooo 0000 oooo_oow

B00000.00000000000000000000000000 .
90000000000000000000000000000000
AOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOO

1in e Aux
Shine/Aux
Shin e/Aux
Shin e/Aux
1in e/Aux
SetDefau lts
SetDefaults
SetDefaults

!I
0~
0 11
_ Odl
04
10
rt'O
110

-

--

011

-C

3

I
I

on

0

::J
~

I

~

ro

OJ

Vo

::J

0

OQ

on
on

01
on
01\
1
on
O,
Oil

C

Sub Command
un ction
bits L.XIve1 I l.X!Ve2 \DriVe3 ID"ive4 Shine/Camera Con tro
00
0
0
0
0
1
00
0
0
0
0
1
01
0
0
0
0
0
·10
0
0
0
0
1
10
10
0
0
0
0
1
00
0
0
0
0
0
.,
10
0
0
0
0
0
0
0
0
00
0
0
0
0
0
10
0
0
0
0
1
10
0
0
0
0
1
1
10
0
10
0
'\O
0
1
10
0
1
.,
·10
0
10
0
1
10
1
.,
·10
0
0
0
0
10
0
0
0
0
1
10
0
0
0
0
1
.,
'IO
·10
0
0
0
0
1
.,
·10
0
0
0
0
10
0
0
0
0
1
·o
·1
0
0
·10
0
0
0
0
1
10
0
0
0
0
1
10
0
0
0
0
1
·1
0
1
0
0
0
1
.,
·1
0
0
0
0
0
1
0
1
·1
0
0
0
1
1
0
0
0
'\
0
1
0
0
1
0
1
0
0
1
·1
0
1
a
0
0
0
0
1
0
0
0
0
0
0
0
0
0
1
0
~
0·1
0
0
0
0
0
01
0
0
0
0
0
01
0
0
0
0
0
·1
0
0
01
0
0
0
0
0·1
0
0
0
0
0
01
0
0
0
0
0
1
0·1
0
0
0
0
0
0·1
0
0
0
0
0
01
0
0
0
0
0
.,
0
0
0·1
0
0
0
0
0
~
0·1
0
0
0
0
0
0·1
0
0
0
0
0

I

.,

1·1
0·1
10

0

ro
<
ro
0
"O

3

ro

::J

rl'

OJ

::J

a..
)>
"O
"O

-

n

OJ

rl'

0
::J

0

--t,

OJ

""CJ
::T

0
0

rl'

::J

n

0

C

::J

rl'

::J

(J'Q

3
OJ

(J'Q

0
0
0

0
0
0

0
0
0

0
0
0

0
0
0

::J
(J'Q
(/)

-<
VI

rl'

ro

3


The column at the extreme left of the command table is not generated by the spread-sheet, but is a short-hand description of the command on that line that must be entered (and maintained in the case of any changes) by the user. Its accuracy is essential, as a printout of the first two columns of the command tables is an essential document when creating the low-level PCIS Sequence files that call the IICU commands. (The creation of Sequence files is described in Chapter 5.) Copies of these printouts for Command Table Set 2 – showing the commands implemented in the command tables currently in use – appear at Figures 4.7.2 and 4.7.3 below.

4.7.2 Command Table Conventions

The IICU Command Table spread-sheet does not enforce any control over the contents that the user decides to enter (though it does not allow data to be entered into the 17 unused bits in each word). The creation of legal, let alone sensible commands, is entirely the responsibility of the user. In particular, any standards about how commands are arranged in the table, and what commands are chosen for inclusion, have at best the standing of convention.

Nevertheless certain conventions have emerged as useful and sensible, and are described below.

4.7.2.1 Conventional Values of Particular Command Table Locations

Certain Command Table Locations (or Command Numbers) have by convention to contain particular IICU commands. Word 0 and Words 61 to 63 contain the commands listed below. That leaves Words 1 to 60 of each command table – 120 commands in all – at the discretion of the creator of the table.

4.7.2.1.1 Word[0] of each table is a Reset command.

This is the only really mandatory convention, since the IICU expects a Reset Command at Word 0, and executes it during a power-up or a Hard Reset (when the front panel Reset button is pushed).

The Reset command at Word 0 of the Primary Command Table contains the Command Table Number in the Shine Duration field. (This is displayed on the IICU front panel and used to synchronise the PCIS runtime software with the Command Table Set in use.) The LED Current fields contain the Soft Maximum allowable currents for each LED string. Most other fields of the Reset command may contain default values to be copied to the internal Default Command Word.

Another reason for this allocation of function to Word 0 is that if anything goes wrong with the command communication structure (which, going via Minisys commands and relay toggling is not impossible, especially during early testing), then the most likely command to be accidentally invoked – all zeros – will be a harmless Reset.
4.7.2.1.2 Word[63] of each table is “Switch to Primary Table”

Since there is no IICU status bit telling the Minisys which command table (Primary or Secondary) is in use, separate commands for “Switch to Primary” and “Switch to Secondary” are supported. Since “all ones” (corresponding to Command 63) is the second most likely command number to be accidentally invoked (given the confusion over the sense of TTL level inputs and outputs with the Minisys), and “Switch to Primary” also seems a harmless command to execute accidentally or repeatedly, Command Word[63] is by convention always a “Switch to Primary Table” command.

4.7.2.1.3 Word[62] of each table is “Switch to Secondary Table”

In theory, only the Primary Table needs a “Switch to Secondary” command; and only the Secondary Table needs a “Switch to Primary”. But, since the Primary/Secondary state cannot be read by the PCIS, such an approach would require the PCIS runtime software to keep permanent track of which table was in use. The only way to resolve any doubt would be to issue a Command[0], which is always a Reset and would switch the IICU back to using the Primary Command Table.

A neater solution – and the one implemented – is to have both commands, in identical locations, in each table. (Command 62 – adjacent to the “Switch to Primary” – just seemed a logical and convenient location to choose.) In this way, a Command[63] will always place the IICU in the Primary command table, whatever its previous location; similarly, a Command[62] will reliably place the IICU in the Secondary Command Table.

4.7.2.1.4 Word[61] of each table is either Toggle Over-ride Mode or a No-Op.

It seemed sensible to have a conventional location for the Toggle Over-ride Mode (the only “mode-change” IICU command other than the table select commands), and Word[61] was selected. This conveniently leaves Words 1 to 60 available for the “real” commands of the Command Table Set; and additionally leaves the locations immediately following the Reset command at Word[0] available as the logical location for any extended initialisation commands.

Note that the PCIS can read the state of the Over-ride Mode via a reported status bit, and so can resolve any uncertainty that might arise as to whether an earlier stage of the run had selected Over-ride Mode or not. Therefore a single command that toggles the Over-ride state is sufficient.

Command tables that do not include the facility to select Over-ride Mode should have a No-Op (No Operation) at Word 61. In this way, any protocol attempting to select Over-ride Mode with a table that doesn’t allow it will cause an IICU Error and stop executing. Thus two versions of a Command Table Set could be used, differing only in this command in each table. The version without the “Toggle Over-ride” Command could be used for work that did not require the Over-ride Mode to be available, avoiding any possibility of accidentally specifying any “high risk” commands.
4.7.2.2 Conventional Use of Primary and Secondary Tables

The conventions around what commands are implemented in a command table set, and where in the table set they go, is based on the concept of using:

1. a small number of relatively specific commands that provide for the quick and simple execution of common ICU functions; and
2. a very broad range of Set Defaults commands, in combination with very general Shine/Aux commands that use all, or a large majority, of the defaulted values. This is slower (requiring more commands), but preserves the flexibility that allows the PCIS to implement the broadest possible range of ICU commands.

An associated concept is that the ICU should normally – and especially during time-critical operations – be running from the Primary Command Table. This translates to the idea that commands that actually do things – mainly Shine/Aux commands – should all be in the Primary table, while the Secondary table should only contain commands that are used to set up the ICU for the “active” commands – i.e. Set Defaults commands that would normally only be called during PCIS and ICU initialisation.

To the extent that this approach can be consistently followed (space limitations in the two tables force some compromise), the Primary table can be populated with commands that may take some time to complete, and generally require that the PCIS runtime software poll the ICU Command Active bit for completion of the command before continuing; and the Secondary table can be populated with effectively instantaneous Set Default commands that do not require polling for completion. (This approach is assumed by the PCIS Simulator software described in Chapter 5 when checking for correct ICU Command Completion polling during protocol testing.)

The trick in developing ICU Command Tables is to predict which are the Shine and Aux operations that will be used most commonly to implement the usual TL, OSL and TOSL cycles of the PCIS protocols; and which functions should be supported via the slower but more general approach of building up, and then invoking, a Default Command Word. Increasing experience with the use of the PCIS is assisting this process, and is reflected in the emerging design of Command Table Set 3. Also under study is the question of which Auxiliary functions should be available directly as independent Shine/Aux commands (preferably in the Primary command table), and which should be supported via Set Defaults commands in the Secondary command table.

A final and important convention regarding the overall design of command tables is that, other than the Over-ride and non-Over-ride versions of tables mentioned earlier, the existence of different versions of the same command table number should be avoided. Command Table Sets that are edited or changed in any significant way should be given a new Command Table number, rather than over-writing or replacing the earlier version. In this way all earlier PCIS runs remain precisely (and easily – see Chapter 5) repeatable. This is often a requirement (which the Minisys was rarely able to meet) when the results of a luminescence experiment produce “interesting” results that encourage the repeat of the experiment on new or different samples.
4.7.3 Development Command Tables

By convention also (since I get to choose the conventions at this stage), ICU Command Table Slot D (selected via the ICU front panel switch), and all Command Table Numbers from 90 up, are reserved for development and testing of the ICU. Command Table numbers in this range could be free, at my discretion, of any of the conventions and rules applying to “ordinary” command tables.

Command Tables numbers in this range can be freely re-used, as their purpose in testing is often very temporary. Also, there is usually only one Command Table of each number, rather than a Primary and a Secondary, as 60 commands is usually ample for the testing being conducted. On occasion the same table is loaded into both the Primary and Secondary slots so that table switching commands can be tested.

Development and testing tables can (again, by my convention) be loaded into ICU Slot D at any time, without consultation with other users who may have loaded tables into Slot D.

Development tables were numbered from 99 down, and to date four single development tables have been used, numbered from 99 to 96.

4.7.4 Command Table Sets 1 and 2

Command Table Set 1 and Command Table Set 2 are functionally equivalent in their specifications, but built to the different Command Word structures of ICU V1 and ICU V2 respectively. (Command Table 2 actually does not implement four commands in Secondary Table 1, related to selecting the Camera Ready Input Transition Required, since ICU V2 no longer supports this selection. No existing protocols had used those commands in Command Table Set 1.)

Command Table Set 1 underwent a few revisions during early development (breaking the rule espoused above, but only while there was no existing body of PCIS work to be affected), but remained to the end significantly as it was originally designed – a design that has proved effective (if not perfect) in supporting all of the PCIS TL and OSL protocols that have been developed to date. It has some weaknesses that will be addressed in Command Table Set 3, but has nevertheless proved highly functional.

The discussion that follows is based on the current Command Table Set 2 (as it is consistent with the ICU V2 Command Word description given above), but applies in equal measure to Command Table Set 1, which was actually used for most of the PCIS work featured in this thesis.

Figures 4.7.2 and 4.7.3 below show the Summary Print-outs for the Primary and Secondary tables respectively of Command Table Set 2. (The full tables appear at Appendix E.) These printouts are the principal references for the command tables during individual command testing (using the ICU Test Environment – see section 4.8), and during the development of low-level Sequence files that invoke the ICU commands (see Chapter 5). They will also serve as the focus for discussion here of the structure and contents of Command Table Sets 1 and 2.
### Command Table Number 02 - Primary

<table>
<thead>
<tr>
<th>Decimal</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Reset - Aux=0, T/S/CR, Closed, I Limits = LOW (10%)</td>
</tr>
<tr>
<td>1</td>
<td>Reset - Aux=0, T/S/CR, Closed, I Limits = HIGH (80%)</td>
</tr>
<tr>
<td>2</td>
<td>Aux - Use Defaults - ALL</td>
</tr>
<tr>
<td>3</td>
<td>Cancel Shine - No Sync, Restore Dflt Shutter</td>
</tr>
<tr>
<td>4</td>
<td>Cancel Shine - S/CR, Restore Dflt Shutter</td>
</tr>
<tr>
<td>5</td>
<td>CLEAN CCD - Shine - 0.0 Sec, NO LEDs, Closed, S/CR, Restore</td>
</tr>
<tr>
<td>6</td>
<td>!cancel -1.0 Sec, IR LEDs, !=FOCUS, Restore, Rest=Dflt</td>
</tr>
<tr>
<td>7</td>
<td>!cancel -1.0 Sec, Red LEDs, !=FOCUS, Restore, Rest=Dflt</td>
</tr>
<tr>
<td>8</td>
<td>!cancel -1.0 Sec, Blue LEDs, !=FOCUS,Restore, Rest=Dflt</td>
</tr>
<tr>
<td>9</td>
<td>!cancel -1.0 Sec, UV LEDs, !=FOCUS, Restore, Rest=Dflt</td>
</tr>
<tr>
<td>10</td>
<td>Shinee - 0.1 Sec, No Sync, Restore, Rest=Dflt</td>
</tr>
<tr>
<td>11</td>
<td>Shinee - 0.1 Sec, T, Restore, Rest=Dflt</td>
</tr>
<tr>
<td>12</td>
<td>Shinee - 0.1 Sec, S, Restore, Rest=Dflt</td>
</tr>
<tr>
<td>13</td>
<td>Shinee - 0.1 Sec, S/CR, Restore, Rest=Dflt</td>
</tr>
<tr>
<td>14</td>
<td>Shinee - T/S, Restore, Rest=Dflt</td>
</tr>
<tr>
<td>15</td>
<td>Shinee - T/S/CR, Restore, Rest=Dflt</td>
</tr>
<tr>
<td>16</td>
<td>Shinee - 2T Dur'n Int, No Sync, Restore, Rest=Dflt</td>
</tr>
<tr>
<td>17</td>
<td>Shinee - 2T Dur'n Int, T/S/CR, Restore, Rest=Dflt</td>
</tr>
<tr>
<td>18</td>
<td>Shinee - 2T Dur'n Int, T/S/CR, Rest=Dflt</td>
</tr>
<tr>
<td>19</td>
<td>Shinee - 2T Ext Delay, T/S/CR, Restore, Rest=Dflt</td>
</tr>
<tr>
<td>20</td>
<td>Shinee - 2T Ext Delay, T/S/CR, Rest=Dflt</td>
</tr>
<tr>
<td>21</td>
<td>Shinee - 2T Ext Delay, T/S/CR, Rest=Dflt</td>
</tr>
<tr>
<td>22</td>
<td>Shinee - 2T Ext Delay, T/S/CR, Restore, Rest=Dflt</td>
</tr>
<tr>
<td>23</td>
<td>Shinee - 2T Ext Delay, T/S/CR, Rest=Dflt</td>
</tr>
<tr>
<td>24</td>
<td>Shinee - 2T Ext Delay, T/S/CR, Rest=Dflt</td>
</tr>
<tr>
<td>25</td>
<td>Shinee - 2T Ext Delay, T/S/CR, Rest=Dflt</td>
</tr>
<tr>
<td>26</td>
<td>Shinee - 2T Ext Delay, T/S/CR, Rest=Dflt</td>
</tr>
<tr>
<td>27</td>
<td>Shinee - 2T Ext Delay, T/S/CR, Rest=Dflt</td>
</tr>
<tr>
<td>28</td>
<td>Shinee - 2T Ext Delay, T/S/CR, Rest=Dflt</td>
</tr>
<tr>
<td>29</td>
<td>Shinee - 2T Ext Delay, T/S/CR, Rest=Dflt</td>
</tr>
<tr>
<td>30</td>
<td>Shinee - 2T Ext Delay, T/S/CR, Rest=Dflt</td>
</tr>
<tr>
<td>31</td>
<td>Shinee - 2T Ext Delay, T/S/CR, Rest=Dflt</td>
</tr>
<tr>
<td>32</td>
<td>Shinee - 2T Ext Delay, T/S/CR, Rest=Dflt</td>
</tr>
<tr>
<td>33</td>
<td>Shinee - 2T Ext Delay, T/S/CR, Rest=Dflt</td>
</tr>
<tr>
<td>34</td>
<td>Shinee - 2T Ext Delay, T/S/CR, Rest=Dflt</td>
</tr>
<tr>
<td>35</td>
<td>Shinee - 2T Ext Delay, T/S/CR, Rest=Dflt</td>
</tr>
<tr>
<td>36</td>
<td>Shinee - 2T Ext Delay, T/S/CR, Rest=Dflt</td>
</tr>
<tr>
<td>37</td>
<td>Shinee - 2T Ext Delay, T/S/CR, Rest=Dflt</td>
</tr>
<tr>
<td>38</td>
<td>Shinee - 2T Ext Delay, T/S/CR, Rest=Dflt</td>
</tr>
<tr>
<td>39</td>
<td>Shinee - 2T Ext Delay, T/S/CR, Rest=Dflt</td>
</tr>
<tr>
<td>40</td>
<td>Shinee - 2T Ext Delay, T/S/CR, Rest=Dflt</td>
</tr>
<tr>
<td>41</td>
<td>Shinee - 2T Ext Delay, T/S/CR, Rest=Dflt</td>
</tr>
<tr>
<td>42</td>
<td>Shinee - 2T Ext Delay, T/S/CR, Rest=Dflt</td>
</tr>
<tr>
<td>43</td>
<td>Shinee - 2T Ext Delay, T/S/CR, Rest=Dflt</td>
</tr>
<tr>
<td>44</td>
<td>Shinee - 2T Ext Delay, T/S/CR, Rest=Dflt</td>
</tr>
<tr>
<td>45</td>
<td>Shinee - 2T Ext Delay, T/S/CR, Rest=Dflt</td>
</tr>
<tr>
<td>46</td>
<td>Shinee - 2T Ext Delay, T/S/CR, Rest=Dflt</td>
</tr>
<tr>
<td>47</td>
<td>Shinee - 2T Ext Delay, T/S/CR, Rest=Dflt</td>
</tr>
<tr>
<td>48</td>
<td>Shinee - 2T Ext Delay, T/S/CR, Rest=Dflt</td>
</tr>
<tr>
<td>49</td>
<td>Shinee - 2T Ext Delay, T/S/CR, Rest=Dflt</td>
</tr>
<tr>
<td>50</td>
<td>Shinee - 2T Ext Delay, T/S/CR, Rest=Dflt</td>
</tr>
<tr>
<td>51</td>
<td>Shinee - 2T Ext Delay, T/S/CR, Rest=Dflt</td>
</tr>
<tr>
<td>52</td>
<td>Shinee - 2T Ext Delay, T/S/CR, Rest=Dflt</td>
</tr>
<tr>
<td>53</td>
<td>Shinee - 2T Ext Delay, T/S/CR, Rest=Dflt</td>
</tr>
<tr>
<td>54</td>
<td>Shinee - 2T Ext Delay, T/S/CR, Rest=Dflt</td>
</tr>
<tr>
<td>55</td>
<td>Shinee - 2T Ext Delay, T/S/CR, Rest=Dflt</td>
</tr>
<tr>
<td>56</td>
<td>Shinee - 2T Ext Delay, T/S/CR, Rest=Dflt</td>
</tr>
<tr>
<td>57</td>
<td>Shinee - 2T Ext Delay, T/S/CR, Rest=Dflt</td>
</tr>
<tr>
<td>58</td>
<td>Shinee - 2T Ext Delay, T/S/CR, Rest=Dflt</td>
</tr>
<tr>
<td>59</td>
<td>Shinee - 2T Ext Delay, T/S/CR, Rest=Dflt</td>
</tr>
<tr>
<td>60</td>
<td>Shinee - 2T Ext Delay, T/S/CR, Rest=Dflt</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>!Switch</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>!Switch</td>
<td>to Secondary</td>
</tr>
<tr>
<td>!Switch</td>
<td>to Primary</td>
</tr>
</tbody>
</table>

**Figure 4.7.2 IICU Command Table 2 Primary – Command Summary.**

It will be seen that the Primary command table largely follows the table conventions described above, with the exception that there are two Set Defaults commands (which do not require polling for completion) in the Primary table (commands 8 and 9). These two commands set the Camera Synchronisation sub-field of the Camera Parameters field to “T/S” and “T/S/CR” respectively.
"T/S" sets the IICU so that a Shine Command (using the relevant default settings) will cause the camera system to record an image, but the IICU will not wait for the camera read-out to complete before reporting to the PCIS that the IICU command has completed. "T/S/CR" will likewise cause an image to be recorded, but the IICU will wait for the camera system to complete the read-out before reporting the IICU command as complete. It was felt that this level of immediate control was required, and warranted these commands being available in the Primary command table. Experience has shown that there are no circumstances where time is critical, but the completion of the camera system readout before proceeding is not; and so the "T/S" option has not been used. Therefore the "T/S/CR" alternative does not need to be in the Primary table either, and neither of these Set Defaults commands will appear in the Primary table of Command Table Set 3.

The Primary Table 2 starts with some general commands, including the much-used “Clean CCD”. In this Table the Reset at Word[0] sets particularly low current limits (10% of rated LED currents). This is to provide a high degree of protection to the source turret LEDs (which are very difficult to replace) during system testing. There is a second Reset command at Word[1] which sets the Soft Maximum LED currents to the more normal value of 80% of the rated LED currents.

Most of the Primary table – 30 + commands – consists of groups of Shine commands. Each group expresses the different available values of one only of the parameter fields, while defaulting the rest of the Shine/Aux command parameters. Note that exposure times of 0.1 seconds are offered in this group, essentially implementing an “immediate” shine, as in IICU V1 a time of 0.0 seconds would represent an “Indefinite Shine”. Longer exposure times – as actually used for TL and OSL accumulations – are set using the Set Defaults commands in the Secondary table.

The Primary table also offers immediate access to four Focussing commands, each implementing a 1.0 second accumulation using one of the four LED strings. Experience has shown that it is difficult to get a “good exposure level” that works both for TL runs (with no optical filters present) and for OSL runs (with optical filters that inevitably reduce the CCD photon counts), and future Command Table Sets will therefore include eight focussing commands, rather than the four (commands 37 to 40) in Table Sets 1 and 2. The accumulation times and LED currents required to obtain the best focussing images in each circumstance, and for each LED string, require further investigation.

The final group of sixteen commands in the Table 2 Primary are Auxiliary commands (i.e. Shine/Aux commands with an Aux component but no Shine part), implementing a selected subset of the available Auxiliary functions. The rest of the Auxiliary functions – with some overlap with the Primary Aux commands – are available via Set Defaults commands in the Secondary table. The division of Auxiliary functions between Aux commands (which have immediate one-off effect) and Set Defaults commands (which have effect when future Aux commands are issued) is one for further study. The approach presently preferred is that those Aux functions which perform an immediate action – e.g. Power On or Off to Outlet A or B, or Send TTL Pulse on Aux Out C or D – should be implemented as Shine/Aux commands in the Primary Table, while those Aux functions that define a behaviour – e.g. Follow General Purpose TTL Input – should be provided as Set Defaults commands in the Secondary table. Command Table Sets 1 and 2 approach, but do not entirely conform to, this design paradigm.

Toggle Over-ride Mode, Switch to Secondary Table and Switch to Primary Table appear at the end of the Primary table as per the recommended convention, but are not present in the recommended positions in the Secondary table. This will be rectified in future Command Table sets.
Apart from the lack of Toggle Over-ride and Switch to Secondary at commands 61 and 62, the Command Table Set 2 Secondary table (shown in summary form above) follows the recommended conventions. There is a Reset command at Word[0], paralleling the conservative (low LED current) Reset command at Word[0] of the Primary table. The rest of the Secondary table comprises groups of Set Defaults commands, each group affecting only one of the Command Word fields.
The first half of Table Set 2 Secondary comprises Set Defaults commands affecting Shine operations. Four groups of commands affect the LED drive strings selected; the LED drive currents used; the duration of the Shine stimulation phase; and the Shutter Mode to be used. Note that due to table space limitations, not all combinations are supported. In particular, only a limited range of drive currents and shine durations are available. (These will be expanded in Command Table Set 3 with the extra flexibility that the IICU V2 structure allows.)

The second half of the Secondary command table contains Set Defaults commands implementing all of the available auxiliary I/O functions, including those also represented by Shine/Aux commands in the Primary table – thus there is significant overlap of these commands. (These Secondary table Set Defaults commands are supported by the "Aux – Use All defaults" command in the Primary table.) Note that there is no “Set Defaults – All Auxiliary Sub-fields to Zero” command (an oversight), so that in Command Table Sets 1 and 2 the only way to clear the auxiliary function sub-fields is by using a Reset command. This oversight will be rectified in Command Table Set 3.

Note that the meaning of the “Set Defaults – Duration = 0.0” command (Secondary command 18) has changed in Table Set 2, since under IICU V2 this no longer represents an “Indefinite Shine”. This has been left as a zero-duration Shine command (rather than being changed to an IICU V2 Indefinite Shine command), as no existing protocol uses the Indefinite Shine anyway, so no incompatibility with existing code is introduced. Finally, there is a group of four commands supporting Set Defaults operations on the “Camera Ready Transition Required” field under IICU V1. Since IICU V2 does not support this function (the Transition Required being hard coded as “Falling Edge”), these fields could not be supported in Command Table Set 2. They have been replaced with IICU No-Op commands, so that any attempt by a PCIS Sequence to execute those commands will trigger an IICU Error condition.

4.7.5 Command Table Set 3

Command Table Set 3 is currently being designed. Its purpose is partly to benefit from the lessons learnt with Command Table Sets 1 and 2 (see above), but mainly to take advantage of the greater flexibility available with the IICU V2 Command Word structure to implement a more comprehensive range of command options in a single table set than was possible under IICU V1.

With the new division of the Shine Command and Camera Control Parameters under IICU V2 into four independently-defaulatable sub-fields, it will be possible to represent all possible settings of these fields as Set Defaults commands in the Secondary table.

The auxiliary I/O functions will be fully represented, but will be divided between (Primary table) Aux commands (which have immediate effect) and (Secondary table) Set Defaults commands, without the overlap that occurs in the earlier table sets. (This will recover some command table entries for other uses.) The precise division of the auxiliary functions between Aux and Set Defaults commands is decided in most cases, but remains “for further study” in some other cases. Command Table Set 3 will rectify the oversight in earlier tables, and provide a Set Defaults command setting all sub-fields of the two Auxiliary Control Bytes to zero values.
The present Command Table Set 2 requires that a different default Shine be set up (using Secondary table Set Defaults commands) for a PCIS thermoluminescence (TL) stage than for an optically-stimulated luminescence (OSL) stage. PCIS runs performing both TL and OSL stages (actually the majority of runs) require a new set-up of the Default Command Word between different stages of the run. In Command Table Set 3 two Shine Commands will be provided, each using different subsets of defaulted parameters, so that re-setting the defaults between run stages is no longer required. This will (slightly) improve the total time efficiency of most PCIS runs.

It has become evident using the PCIS that there is a need for both a “Short Focus” and a “Long Focus” command for each LED string – i.e. a total of eight “Focus” commands instead of the four in Table Set 2. The Short Focus commands will be set (LED current and duration) to produce good exposures (pixel photon counts in the thousands) with no optical filters installed. The “Long Focus” parameters will be set to produce good exposure levels when the “normal” filters used to pass luminescence photons of the corresponding energy are installed. For example, the “UV Long Focus” will be set for use with 5 or 6 mm of UG11 filters, as normally used for blue/UV quartz OSL.

Despite increasing the number of Set Defaults commands to provide complete coverage of the Shine Parameter options (including the LED strings in use), the improved structure of the IICU V2 Command Word means that there will still be more table entries available in Table Set 3 for setting Drive Current and Shine Duration options than were available in Table Sets 1 and 2.

What additional current range and duration options are made available is a matter for further discussion, but is likely to include a greater range of short durations. PCIS runs to date suggest that the signal stored in most samples is stimulated out quite quickly, and in most runs the second accumulation of each OSL stage is devoid of signal – i.e. it has all been shone out during the first accumulation. (This discussion is less applicable to TL runs, where the accumulation time tends to be linked to the rate at which the temperature is ramped, which in turn depends more on the experiment being conducted than on the capabilities of the equipment being used.) Since the stimulation power (particularly of the blue LEDs) has proved adequate, the longer Shine Durations available in Table Set 2 have received less use than the shorter durations, and greater benefit is likely to derive from supplying more options for short Shine Durations than from more long options.

Finally, and despite the extra functionality being supported, the greater efficiency of the IICU V2 structure means that Command Table Set 3 will have some “spare” entries in the Primary table. These locations will be “Reserved for Custom Table Functions”, and may be used by a researcher to extend a Command Table Set without needing to change the existing commands it contains. This allows a researcher to implement a few extra custom commands to support specific research needs without needing to generate a complete new Command Table Set with a new Table Set Number. Other users of the basic Command Table Set will not be affected by the extensions, since no existing protocol code will use those command numbers. Researchers using the Extended Command Table Set facilities will need to ensure that their own particular version of the (Primary) table is installed in the IICU prior to a PCIS run; but most researchers, using standard protocols running on the standard table set, will be unaffected by whichever extended version is installed, and will not need to load a particular version into the IICU (and therefore do not need to know how to do so).

Command Table Set 3 will conform fully with the command table conventions described earlier.
4.8 The IICU Bench Testing Environment

Testing of the IICU firmware that implements the Command Table structure, as well as the development and testing of the Command Tables themselves, would clearly require the ability to exercise individual IICU commands, and to monitor their performance.

One possibility was to perform IICU firmware and Command Table testing in the full PCIS environment, with the real Minisys, LN-CCD camera system, optics and turrets of the PCIS. This would necessarily need to be performed in the low-light laboratory in which the Minisys and other PCIS components are installed. This could be undertaken using the Minisys low-level command entry facility of the Minisys’ Centre.exe application, or the then-emerging functionality of the PCIS runtime software (see Chapter 5). In fact some PCIS system integration testing did occur this way, but it was not the ideal way to test components of the PCIS, especially the IICU, for the following reasons:

- The laboratory is a difficult environment in which to work, for reasons of both space and light levels. IICU testing requires working with multiple pieces of equipment and various source code print-outs, and is easier in a better-lit and less cluttered environment.
- There is a risk during the testing of a system’s components that an error in one component may lead to damage being caused to other components. Since I had no budget to replace even those parts of the PCIS that were available off-the-shelf, nor time to repair or re-build custom components such as the source turret, it was important to avoid this risk.
- Incredibly, at the time of the IICU development, I was being denied any sort of regular or frequent access to the Minisys to undertake any sort of PCIS testing at all. It was necessary to perform all possible PCIS component testing without the Minisys, so that the occasional access I was granted could be devoted to integration testing of the PCIS as a whole.

Accordingly I decided to devote the necessary resources to building a test-bed environment for the IICU. This environment comprises various hardware units that simulate the functions and behaviour of the PCIS components normally connected to the IICU — the Minisys, the source and destination turrets, the camera and shutter systems, and the various optional auxiliary components.

The camera and shutter systems could be simulated using the general purpose TTL signal generator and monitor built for the Phase 1 PCIS interim electronics, and described earlier in Section 3.1.4. Simulators for the Minisys itself, and for the source and destination turrets, were custom built for the IICU test environment. The Minisys simulator also includes facilities for exercising the IICU’s auxiliary TTL inputs and outputs.

The custom components designed and built for the IICU test environment are described in the following sections.
4.8.1 The Minisys Simulator

The Minisys simulator (pictured below at Figure 4.8.1) primarily serves to replicate the functions of the optional I/O card installed in the Minisys. It provides eight toggle-switched outputs (representing the eight Minisys relay outputs) and eight LED-indicator inputs (showing the state of the eight IICU State outputs to the Minisys).

These principal IICU/Minisys connections are carried on the 25-way and 35-way D-connector cables seen at the rear of the Minisys simulator. Note that the 35-way cable carrying the IICU State outputs to the Minisys also carries the signals from the “loop-back” ports on the IICU back panel that normally provide access to the other eight optically-isolated inputs on the Minisys I/O card. These eight inputs do not form any part of the IICU’s functional definition (as explained earlier, they are made available at the IICU back panel purely to simplify the cabling of PCIS experiments), and so they are not monitored by the Minisys simulator built for IICU testing.

The eight switches in the bottom row of the Minisys simulator allow an IICU command number to be set up in binary (using the first six switches), and then to be “Command Clocked” to the IICU to invoke the execution of that command in the currently-active command table. For testing of Shine commands that depend on the Secondary Trigger input from the Minisys, the eighth switch allows that signal to also be simulated.

![Figure 4.8.1 IICU Test Harness – Minisys Simulator.](image)

As can be seen in the photo, the simulator also provides two switches to exercise the IICU General Purpose TTL Inputs A and B; and four LEDs to monitor the IICU Auxiliary TTL Outputs. These inputs and outputs are connected to the IICU via the BNC co-axial cables that can be seen in Figure 4.8.1.
4.8.2 The Source Turret Simulator

The source turret simulator (pictured below at Figure 4.8.2) provides a plug-compatible version of the PCIS source turret in which LEDs can, if necessary, be replaced quickly and easily. As in the real source turret, there are four strings of LEDs, but unlike the source turret, all four LED strings are accessible and visible. There is also a front panel switch that replicates the function of the source turret micro-switch that detects whether the PCIS optics are assembled properly (switch down), and disables the shutter if they are not (i.e. if the switch is up).

Note that the source turret simulator has a light detector fitted. In the final implementation of the IICU there is no interlock between the source turret light levels and the shutter (only the destination turret light level is used), so the source turret simulator’s light detector was covered with tape during testing, and no testing of its functioning was conducted.

![Figure 4.8.2 IICU Test Harness - Source Turret Simulator](image)

Note that the term “LED string” is used slightly loosely here; each LED string in the source turret is represented by a single LED in the simulator, and the IR and UV LEDs are replaced by visible red LEDs. One high-current LED is installed to allow higher current settings in the control tables to be tested, but there are no strings requiring a high drive voltage (as the blue string in the real source turret does). The voltage and current performance of the digitally-controlled current drives for the LED strings had been previously tested in the Electronics workshop, and the IICU test environment did not need to provide for further testing the current driver hardware.
4.8.3 The Destination Turret Simulator

The destination turret – or shutter housing – simulator is a very simple device indeed. It comprises a light detector and a switch mounted in a black plastic film canister. To simulate dark (normal) conditions over the shutter, the lid is placed on the film canister; to simulate (abnormal) illuminated conditions at the shutter (and so exercise the shutter interlock at the start of a Shine command), the lid is simply removed. (Note that this does not test the sensitivity of the light detector, and the light level at which the interlock works; it simply uses the very high light levels of normal office illumination to test the basic operation of the interlock. The sensitivity of the light detector must be adjusted – using a trim-pot internal to the IICU – in the actual laboratory lighting conditions.)

The destination turret simulator also includes a switch to detect if the lens is down (PCIS correctly assembled). As noted earlier, this was not fitted to the real destination turret, the micro-switch on the source turret being relied upon instead. Since these two switches are simply wired together inside the IICU to construct an effective “OR” gate of the two “Lens Not Down” signals, the IICU functions can be tested with either the source turret or the destination turret switch.

The destination turret simulator (shown at Figure 4.8.3 below) is fitted with a Mini-DIN plug to make it plug compatible with the real destination turret.

Figure 4.8.3 IICU Test Harness – Destination Turret Simulator.
4.8.4 The IICU Bench Testing Setup

The IICU Test Bench is designed to exercise all of the available inputs and outputs, and all of the available functions of the IICU. Figure 4.8.4 below shows the back panel of the IICU installed in the test rig. The only ports not connected are the Management PC Interface (since the laptop I use for Command Table development was not in the office when the photo was taken), and the "loop-back" connectors for the upper eight Minisys optically-isolated inputs. (Note that one of the loop-back inputs is occupied as part of some testing of the management interface facility to nominate the Rapid Repeat Shine Reporting Port. The nominated port is normally looped back to the Minisys via "MINISYS INPUT 8", as shown in this photo, so that it can be read by the PCIS runtime software.)

Also seen in Figure 4.8.4 are the two "night lights" used to test the switched power outputs. These contain incandescent globes, and so present a "well-behaved" resistive load of a few watts to the power outlets. They are clearly visible even in a well-lit office, without being unpleasantly bright when they switch on. Each is labelled (front and back) for the power outlet to which it is attached.

This immediately identified an IICU problem in which – somewhere between the Command Table Builder spread-sheet and the physical port addresses used in the IICU firmware – the two outlets identities had been confused, and the wrong outlet was being switched.

The complete IICU Test Environment – including the laptop used as the IICU Management PC – is shown below in Figure 4.8.5. Note that none of the equipment is powered on in this photo.
In the above photo of the IICU test environment, the Minisys simulator can be seen to the left of the IICU, with the source turret and destination turret simulators just behind it. To the right of the IICU is the management PC (laptop). The management PC would normally be showing a section of a command table, or running the IICU management interface (displaying IICU internal parameter values) via the HyperTerm terminal emulator program. Just to the left of the management PC is the general purpose TTL signal generator/monitor, acting as a simulator for the camera system and shutter control functions.

At the back of the IICU are the two night-lights connected to the switched power outlets, and behind them is a dual-trace digital oscilloscope. The oscilloscope is connected to various points in the test environment to measure any time delays (whether requested, as with shutter latencies, or whether undesired) in the execution of IICU commands.

Also visible at the back left of the bench are the LED timing controller and current driver from the Phase 1 PCIS interim electronics. These are not part of the IICU test environment, and should not have been on the IICU test bench.
5 Development of the PCIS Sequence Software

"Simple things should be simple; complicated things should be possible."

Alan Kay, Computer scientist

The PCIS Sequence Software was developed to automate the operation of the PCIS, and to comply with the dictates of this chapter’s introductory quote. The PCIS now had the proven capacity to undertake a wide range of luminescence experiments, but was enormously complex to operate. The goal of the Sequence Software – PCIS_Seq and its associated components – was to make “normal” luminescence operations (such as the Single Aliquot Regeneration (SAR) protocol usually applied to quartz dating) simple, while preserving the ability to run (albeit at the cost of greater complexity of operation) any luminescence experiment or protocol of which the PCIS was physically capable.

This chapter tells the story of that software development.

5.1 Overview of the PCIS Automation Software

With the completion of the IICU and its command tables, it is now possible to control the whole PCIS, through any combination of the wide range of operations of which it is capable, solely by issuing a series of commands and status requests over a single RS-232 interface. The obvious problem is that the required sequence of commands is extremely long; the correct selection and sequencing of the commands is quite complex; and many of the commands themselves are quite cryptic, quite detailed, or both.

The cryptic nature of the command sequence is shown by the short example of a typical sequence of commands which appears below at Figure 5.1.1. (This is taken from a log file produced by an early version of PCIS_Seq while performing a single TL accumulation from the porcelain sample described earlier. It shows a clean of the CCD; an IICU command to start a secondary trigger-controlled Shine with no LEDs; a Minisys command to do a Thermo-Optical run to 100° C at 2° per second, holding at 90° C, with Relay 8 as the “light source”; and polling the Minisys and IICU for command completion.)

As an indication of the large number of commands required, a later mixed TL, IRSL and OSL run on four sample disks produced a serial log file just over 56,000 lines long, exclusive of repeated status polling commands and responses, which form the majority of the traffic on the serial connection. Although the new log file format includes some information in addition to the actual serial port commands to, and responses from, the Minisys, many commands do not produce responses, and running the entire sequence requires somewhere in excess of 30,000 commands to be issued.

There is no doubt that some custom software is required, that will translate a simple description of a luminescence protocol or experiment into the bizarrely lengthy and complex sequence of Minisys serial commands that is required to implement it. That software is the PCIS Sequence application, and its development, definition and use are described below.
Development and Application of a Photon Counting Imaging System

Figure 5.1.1 PCIS Commands required to perform a single TL accumulation.
5.2 Sequence Templates for Standard PCIS Operations

The best way to present and explain the PCIS Sequence Software is, in effect, the reverse order to its actual development; but it matches the way a user of the PCIS would encounter it. We will start at the “top” or “outside” layer, where simple or standard experiments may be performed easily by an inexperienced user, using a sophisticated PCIS Sequence Template\(^{16}\) which supports a given class of work. We will then delve deeper into the software structure to see how different or more complex procedures can be designed and implemented.

The great majority of work done with luminescence readers is the Single Aliquot Regeneration (SAR) Protocol analysis of quartz grains for dating purposes. There is even more emphasis on Single Aliquot protocols with the PCIS (as opposed to protocols that compare separate but supposedly “identical” samples), since at the spatial resolution of the imaging capabilities of the PCIS the different samples are not identical. Only if the data is averaged over the entire sample disk is the “identity assumption” even arguably justifiable; but then most of the benefit of the PCIS would have been discarded.

The SAR protocol is reasonably complex, and can include TL, OSL and IRSL cycles additional to the standard Natural, Regen and Test dose OSL cycles of the “basic” SAR protocol. Further, there are particular capabilities of the PCIS, including the capture of image data during sample pre-heats, and the capture of incident light images before or after the SAR cycles (to assist with sub-sample selection, and to detect any movement of the sample relative to the heater plate during the run). These should also be available to any PCIS SAR sequence. A substantial goal of the PCIS Sequence software (and an acid test of its success) would be the development of a comprehensive and flexible SAR sequence template, which would be simple to use for a researcher with general technical competence, but would require no deep knowledge of the PCIS hardware or software.

The approach I selected – and had always preferred – was that of a human-readable and easily editable text file, created under a simple application like Notepad, that would be interpreted and implemented by custom-written software in order to execute the PCIS run that it describes. The text file must clearly conform to some formal specification to make it machine-readable (the PCIS Sequence Language Definition, which I developed and will describe later), but can still have sufficient flexibility of syntax, verbosity (as opposed to the terseness so beloved of computer programmers) and use of keywords, to make it clear and simple to read and understand – and to modify as needed.

The degree to which I succeeded in these goals may be judged from the section of the final SAR_Template Sequence that I developed, which appears at Figure 5.2.1 below (over two pages). The full SAR sequence definition file (which appears in Appendix F) is two to three times as long as the extract shown, and offers fully-documented options to control many additional run parameters. However, for the great majority of SAR dating experiments, the additional parameters would not need to be changed, and the operator would need to understand no more than the two pages of text shown below in order to operate the PCIS. As observed, simple things should be simple.

\(^{16}\) A note on terminology. The term “protocol” is used to refer to a generalised scheme for using luminescence data for a given purpose (often the dating of a sample). The term “sequence” is used to denote a particular implementation of a luminescence protocol or experiment, either for a particular sample run, or for a specific machine such as the Minisys or the PCIS. A “Sequence Template” is an especially flexible PCIS Sequence implementing a particular protocol, but designed to be customised for individual PCIS runs.
Development and Application of a Photon Counting Imaging System

SAR_TEMPLATE_V1.SQR

This sequence is designed to run a full SAR protocol on a set of sample disks.

THIS FILE SHOULD BE COPIED, RENAMED AND ALTERED TO SUIT YOUR PARTICULAR REQUIREMENTS.

Update the various process parameters described and set below to customise the protocol for your own use.

All of these parameters are checked in SAR_EDITCHECK_V1.PSI before the Sequence is Run on the samples.

PLEASE UPDATE THESE COMMENTS TO REFLECT THE APPLICATION AND PROPERTIES OF YOUR PARTICULAR SAR PROTOCOL.

```
#DEBUG ON ' Re-instating this line logs extra debug info to Input and Serial Logs.

CmdTblNum

Include SAR_INITIALISE

The following Parameters should be changed to suit the requirements of your specific SAR Run.

' Log Sample descriptions to the RunInfo file.

*********************** SAMPLES CURRENTLY IN MINISYS ***********************
UserLogDesc SAMPLE: 1 (Disks 1, 9, 17, 25, 33): Salt (1) - Australian Lake Salt
UserLogDesc SAMPLE: 2 (Disks 3, 11, 19, 27, 35): Salt (3) - Woolies Home Brand Sea Salt
UserLogDesc SAMPLE: 3 (Disks 5, 13, 21, 29, 37): Salt (10) - Himalayan Rock Salt
UserLogDesc SAMPLE: 4 (Disks 7, 15, 23, 31, 39): Salt (14) - Evaporated Seawater

' If DiskSet is set to NULL then the First / Last / Spacing parameters will be used. If DiskSet is specified (as below) then it will be used in preference.

The two alternate specification types below are equivalent.

Set DiskSet ( 17 25 33 )
Set FirstDiskNum 17
Set LastDiskNum 33
Set SampleSpacing 8

' If FocusDiskNum is set to NULL, the actual Sample disks will be used for focussing. Otherwise, only the selected "FocusDiskNum" will be used for focussing.

Set FocusDiskNum 9
SAR Dose selections. All applied doses are logged to RunInfo.

Set RegenDoseSet (100 250 500 1000 2000)
Set TestDose 30

Set the OSL (stimulation) colour (IR, RED, BLUE or UV), the OSL Temperature, the Filter Set to be used and the Focussing colour (selected to pass the filter Set) for each of the 3 SAR Stages - Pre-Treatment, Main (Natural and Regens) and Post-Treatment.

Values for later Phases may be referenced to those of preceding Phases, as shown below.

The FilterDesc values are used to determine the AUTO sample processing sequence (see below), and to automatically issue requests to the user change the Filter Set and re-focus the optics when required.

Set PreTreatOSLColour IR
Set PreTreatOSLTemp 50
Set PreTreatFilterDesc BG 39 by 3mm, Aperture 42mm
Set PreTreatFocusColour RED
Set SARMainOSLColour BLUE
Set SARMainOSLTemp 125
Set SARMainFilterDesc UG 11 by 6mm Notch Coated, Aperture 42mm
Set SARMainFocusColour UV
Set PostTreatOSLColour SARMainOSLColour
Set PostTreatOSLTemp SARMainOSLTemp
Set PostTreatFilterDesc SARMainFilterDesc
Set PostTreatFocusColour SARMainFocusColour

Set parameters for OSL Pre Heat (i.e. TL) Stages.

The PreHeat Temperature and PreHeat Hold Time may be set separately for the Natural, Regen and Test Dose Stages.

All PreHeats are performed at TLHeatRate Degrees C per second.

At the end of each PreHeat the sample is held at the PreHeat Temperature for the Hold Time, then falls to FinalTemp.

Set PreHeatTempNat 260
Set PreHeatHoldNat 10
Set PreHeatTempRegen PreHeatTempNat ' In this case use same values for
Set PreHeatHoldRegen PreHeatHoldNat ' Regen Cycles as for Natural
Set PreHeatTempTest 160
Set PreHeatHoldTest 0
Set TLHeatRate 5
Set TLFinalTemp 0

The Parameters below should be suitable for most SAR Runs, and will not usually need to be changed.

Figure 5.2.1 Principal section of the SAR Sequence Template file.
The section of Sequence file shown above allows the user to change all the “normal” variables of an SAR Protocol dating run – the sample descriptions; the disks used (here just the last three disks of the Australian Lake Salt sample); the test and regen beta doses to be used; and the temperatures, heating rates, stimulation wavelengths and optical filtering to use both for the main SAR cycles, and for the (optional) Pre- and Post-treatment cycles that some luminescence practitioners like to use.

The full SAR Template Sequence file (seven A4 pages in landscape orientation – see Appendix F) provides for the changing of a range of other run parameters, some of them specific to the PCIS and its imaging capabilities. These additional, fully documented and user-settable parameters include:

- the processing order of the run (disk priority, protocol step priority or Automatic, which minimises operator intervention for filter changes);
- the use of chamber evacuation and Nitrogen gas flow according to the sample temperature;
- whether to wait for the sample to cool down between processing stages;
- options for capturing incident-light sample images at the start, middle and end of the run;
- various SAR main cycle, Pre-treatment and Post-treatment cycle parameters, such as:
  - whether to include an IRSL stage in each SAR cycle; and
  - how images should be captured during both OSL and Pre-heat stages.

The result is a very flexible, advanced SAR protocol, which can be read, understood, modified and used, within a few hours, by any reasonably technically competent luminescence practitioner.

Of course the PCIS is not only capable of SAR analysis, or indeed of dating work generally, and the PCIS Sequence Software suite has been constructed to support any class of work of which the PCIS is physically capable. Sequence Templates (and their supporting bodies of sequence code) can be constructed to make the conduct of any class of luminescence work as simple as SAR Protocol runs.

A number of one-off experiments were conducted during the PCIS development, for which custom Sequence files were created; but one other class of work emerged as an important category, and has been supported by the development of a sophisticated and flexible Sequence Template. That class of work is the broad-spectrum probing of materials with thermoluminescence to detect any signals they may be capable of storing and emitting. (What these signals then tell us about the materials is a later stage of analysis, and would involve other PCIS capabilities, including various-wavelength OSL runs.) This work is supported by the TL_Probe Sequence Template, which appears in Appendix F.

As with the SAR Sequence Template, the TL_Probe Sequence Template comprises about two pages of documented parameters that would usually be changed for each run; and another two to three pages of parameters that would normally not be changed between runs. The parameters include:

- the beta doses to be applied;
- the maximum temperature, hold time and heat rate for the TL runs;
- the temperature ranges over which TL images should be accumulated;
- any optical filtering to be used;
- and various other PCIS run parameters as per the SAR_Template Sequence.

Thus the materials analysis capability of the PCIS is also supported, and made simple to use, by the extended PCIS Sequence software. Other classes of work can be similarly supported in the future.
5.3 Sequence and Include Files for Custom PCIS Operations

The Sequence Template files discussed above, which make certain classes of work so simple to undertake, do not of course exist in isolation. They work because they are supported by a large body of lower level code files (called Include files, with the extension .psi for "PCIS Sequence Include file"), and beneath that by the PCIS_Seq program and the Sequence Language that it implements. And beneath that again, of course, by the IICU control tables, and the ICU firmware that implements them. Of this hierarchy of software, the IICU is delivered including facilities and documentation to support the creation or modification of Sequence files, all the levels of Include files, and the IICU control tables. (The IICU firmware and the PCIS_Seq program are less accessible for modification by a user of the PCIS, and no support is provided for so doing. However, these components provide for complete flexibility in the control tables and Sequence code files that they support respectively, and should not ever need modification in order to exploit the full capabilities of the PCIS).

The creation and modification of IICU Control tables was described in Section 4.7; the PCIS_Seq program and the Sequence language are discussed in Sections 5.5 and 5.6 below; here we consider the blocks of function implemented by the Include files, and how they may be assembled and used to implement specific experimental procedures not covered by the existing Sequence Template files – or to construct new Sequence Template Sets for whole new classes of luminescence experiments.

When humans discuss something, they usually like to invent categories for the something to help them do so. The categories are human inventions, and are not unique – and there are two different ways of categorising Sequence and Include files, useful in two different contexts.

First, from a technical or implementation perspective, the PCIS_Seq software clearly recognises three different categories of files containing PCIS Sequence language code:

1. Sequence files – files ending in the .seq suffix, which are the only files that PCIS_Seq will recognise and run. These should include Initialisation and Finalisation code for the various PCIS hardware components, and will generally call PCIS Include files to do most of the work.
2. Universal Include files – files with the .psi suffix, excluding the Command Table-Specific Include files described below. Universal Include files do not contain any IICU Commands (although they may contain Minisys Commands). They may be called from any Sequence file or any other Include file, to perform a defined body of work.
3. Command Table-Specific Include files – files with the .psi suffix which also contain IICU Commands, and are therefore dependent upon which Command Table Set is installed in the IICU. They must contain a CmdTblNum Command, specifying the number(s) of the Command Table Set (or Sets) with which they are compatible. They are stored in subdirectories (called CmdTbllnn, where nn is the Command Table number) of the main Include files directory. This allows multiple “parallel” versions to exist, each compatible with different Command Table Sets. PCIS_Seq will automatically select the right version, so that higher level sequence code may be used without modification, with any Command Table Set that supports the functions it requires. In order to minimise the work of updating PCIS Sequence code when a new Command Table Set is implemented, these Include files generally contain the smallest units of PCIS functionality that can be sensibly wrapped around the IICU Commands.
How the PCIS_SEQ implements a Sequence file, where it looks for the various Include file types (and in what order), and how this may be used by an experimenter to implement specific PCIS behaviours without affecting other users, is discussed in greater detail in Section 5.5 below.

Although an understanding of the above “technical” categories is essential to the building or modifying of Sequence and Include files, the second way of categorising them is more relevant to an understanding of the logical distinctions between the various files and their functions that underlies the good design of Sequences and Sequence Templates. The three logical categories are:

1. **Sequence Template Files** – this includes all .seq files (specific or “one-off” sequences as well as those implementing Protocol Templates), as well as all those Include files which are specific to (and often implement the real logic of) a particular Sequence Template .seq file.

2. **High Level Include Files** – these generally implement significant blocks of luminescence operations (e.g. a complete TL or OSL stage, possibly including several image accumulations), or of other PCIS functions (e.g. performing image focussing cycles), and are designed to be called either from Sequence files, or from other “higher level” Include files. These files would be directly called – and may even be copied and changed – by PCIS users going beyond the capabilities of the existing Sequence Template files, and writing their own Sequences.

3. **Low Level Include Files** – these generally implement small individual PCIS tasks, such as acquiring a single accumulation, or even just turning a power point on or off. These are generally called only from higher level Include files (and not directly from a Sequence file) to support the higher level blocks of function. PCIS users writing their own sequences would not generally need to use them, and therefore would not need to know their details.

The distinction between High Level and Low Level Include files is obviously not definitive. In some cases, an Include file clearly exists only to be called by a single other Include file, to implement some very specific, locally-meaningful function, and would never be called directly from any other Sequence code file. This is clearly a Low Level file. Other Include files would be called from most or all Sequence files (Initialise and Finalise are the obvious examples), and are clearly High Level files. In other cases the distinction really represents my own judgement as to whether or not the file would be likely to be used directly by PCIS users writing their own Sequence or Sequence Template files.

Command Table-Specific Include files are generally considered to be Low Level. Although some may occasionally be called from a “one-off” Sequence file (e.g., “Turn Power A On”, or “Send Pulse on TTL Output X”), they implement functions that would not normally be called directly from a well-constructed Sequence Template file. Universal Include files, however, are by no means all High Level; the category also includes all Low Level Include files that do not directly execute ICU Commands.

[Note that the arrangement of the Appendices is a blend of these two categorisation systems that is deemed most useful for the use and maintenance of Sequence and Include files. There they are divided into Sequence Template Files (Sequence files and Sequence-specific Include files – Appendix F); Universal Include Files (Appendix G); and Command Table-Specific Include Files (Appendix H). It is assumed that there are no Include files that are both Sequence- and Command Table-specific. This seems reasonable (and is currently true), since Sequence-specific Include files tend to be functionally “high level”, and Command Table-specific Include files generally implement “low level” functions.]

Actual Sequence (.seq) files, by fiat, are not allowed to be Command Table-specific.
PCIS_Seq does not include any passing of parameters or returning of function values in the
traditional sense, and any “parameters” required by the called code, or variables into which results
will be returned, must be set up (using Set or Declare commands) prior to calling the Include file.
There is therefore no formal definition or checking of the “interface” that the Include file “exports”
to borrow some Object-Oriented Programming terms). Instead, every Sequence or Include file must
contain a detailed “comments field” at its head that defines:

- what variables it expects to find, set to what values, on entry;
- where relevant, the legal range of values for any such variables;
- any specific state the PCIS is expected to be in (e.g., Sample on position, Lift Up)
- exactly what functions the Include file performs, including any information logging;
- what variables must exist to be set, and the values they will contain on file exit; and
- where relevant, the state the PCIS is left in on exit.

It is then possible to use the existing Include files in new Sequences by only reading the explanatory
comments field; reading the actual Sequence code they contain should be unnecessary. The best
guide to the required level of internal Include file documentation is to refer to the existing body of
Sequence and Include files in the appendices. (Indeed, the quickest way to learn to write code – in
any language, including the PCIS Sequence language – is by examining existing, well-written code.
The actual language reference documentation – in this case appearing in Section 5.6 and at
Appendix I – is generally only referred to for details not adequately covered by the example code.
This is certainly the approach I would recommend for learning to program the PCIS.)

A section of the SAR Sequence Template file SAR_TEMPLATE_V1.SEQ was shown earlier at Figure
5.2.1. (The complete file appears in Appendix F.) Supporting it in the SAR Sequence Template File Set
are the following Sequence-specific Include files:

SAR_EDIT_CHECK_V1.PSI – This file checks all of the user-entered parameter values in the
SAR_TEMPLATE_V1 Sequence file for individual legality, and for their validity in
combination. If any illegal values are found, an identifying message is logged to the RunInfo
file and the SAR Run is aborted.

SAR_EXECUTE_V1.PSI – Sorts out the sequence of operations necessary to perform the multi-sample
SAR Run, according to the processing order selection made in SAR_TEMPLATE_V1. It then
executes the SAR Run by making calls to SAR_MAIN_RUN (to implement the “main SAR
cycles”), and the single-SAR-Cycle implementation selected in SAR_TEMPLATE_V1 (in this
case SAR_CYCLE_GENERIC.PSI) to implement the Pre-treatment and Post-treatment cycles.

The TL_Probe Sequence Template file (TL_PROBESEQ) is supported by a similar set of Sequence-
specific Include files. The SAR and TL_Probe Sequence Template File Sets, including both the
Sequence files and the Sequence-Specific Include files, appear at Appendix F.

Figure 5.3.1 below shows a typical Universal, High Level Include file. This file, SAR_CYCLE_GENERIC,
displays many of the typical features of a PCIS Include file, including the extensive documentation of
its interface – i.e. the values it needs, what it does, and what it returns to the calling code. It is called
indirectly by SAR_EXECUTE_V1, and directly by SAR_MAIN_CYCLE, to perform the actual SAR cycles,
according to parameter values set at a higher level of the Sequence Template.
Development and Application of a Photon Counting Imaging System

This Include File provides a generic implementation of a single Cycle of an SAR Protocol. It may be called from both SAR Pre-Treatment and SAR Post-Treatment Include files, as well as from SAR_MAIN_RUN.PSI.

SARCycleCount is used to indicate the Natural Cycle (= 0) or the Regen Cycles (> 0).

For each Cycle the following processes are applied to the sample:

1) (Regen Cycles only) applies Beta Irradiation for "Regen Dose" Seconds
2) PreHeats the sample, using the Temperature and Hold Time specified for the Natural or Regen Cycle.
3) Optionally (if SARincludeIR is True), performs an OSL Stage using IR Stimulation at SARinclIRTemp.
4) Performs an OSL Stage (using OSLType Include file), using the Stimulation Colour and Temperature specified for the OSL Phase (Pre-Treatment, Main Phase and Post-Treatment)
5) Applies Beta Irradiation for "Test Dose" Seconds
6) PreHeats the sample, using the Temperature and Hold Time specified for the Test Dose Cycle.
7) Performs an OSL Stage using the Stimulation Colour and Temperature specified for the Phase.

If requested (PhotoAfterNat=True), a Photo Image of the Sample is acquired at the end of the Natural Cycle.

Each SAR Cycle, each Irradiation and each OSL are recorded to the RunInfo file.

Requires: OSLColour
           OSLTemp
           OSLType
           SARCycleCount
           SARincludeIR (Boolean)
           RegenDose
           TestDose
           PhotoAfterNat
           PhotoColour (may be NULL)
           FocusColour

Returns: FirstRegenDose is set from Regen Dose if SARCycleCount = 1 (for use in Post-treatment Phase).
          SARCycleCount incremented by 1

If SARCycleCount > 0
  GOTO REGEN_CYCLE

  Perform a Natural SAR Cycle.
  LogMsg "INFO: SAR " SARPhaseDesc ", Natural Cycle (No Regen Dose) for Disk No: " DiskNum
  Set TLMaxTemp = PreHeatTempNat
  Set TLHoldTime = PreHeatHoldNat
  Call PreHeatType
  GOTO COMMON_CODE

  Perform a Regen SAR Cycle.

LABEL REGEN_CYCLE
LogMsg "INFO: SAR " SARPhaseDesc ", Regen Cycle No: " SARCycleCount ", (Regen Dose: " RegenDose ") for Disk No: " DiskNum
Set BetaTime = RegenDose
Call BETA_IRRADIATE
Set TLMaxTemp = PreHeatTempRegen
Set TLHoldTime = PreHeatHoldRegen

Call PreHeatType

; Common Code for Natural and Regen SAR Cycles. Perform IR OSL?

LABEL COMMON_CODE

If SARIncludeIR <> True
    GOTO SKIP_IR

; Perform optional Included IR OSL before normal OSL Stage.

Declare SavedOSLColour = OSLColour
Declare SavedOSLTemp = OSLTemp

Set OSLColour IR
Set OSLTemp InclIROSLTemp

Call OSLType

Set OSLColour SavedOSLColour
Set OSLTemp SavedOSLTemp

; Normal OSL for Natural or Regen Stage.

LABEL SKIP_IR

Call OSLType

; Perform the Test Dose Stages of the SAR Cycle.

Set BetaTime TestDose

Call BETA_IRRADIAITE

Set TLMaxTemp = PreHeatTempTest
Set TLHoldTime = PreHeatHoldTest

Call PreHeatType
Call OSLType

If SARCycleCount > 0
    GOTO EXIT_SAR_CYCLE

; If requested, take "After-Natural-Cycle" Photo of the Sample.

If PhotoAfterNat = True
    Call PHOTO_DISK

; Finished this SAR Cycle - increment Cycle Count and Exit.

LABEL EXIT_SAR_CYCLE

If SARCycleCount = 1
    Set FirstRegenDose = RegenDose
SetPlus SARCycleCount 1

Figure 5.3.1 The SAR_CYCLE_GENERIC.PSI Universal, High Level Include File.
SAR_CYCLE GENERIC is an extremely versatile file, performing a single OSL cycle (Natural or Regen, or a single Pre-treatment or Post-treatment Cycle), including any specified Regen dose, and also the Test dose and second OSL Shine. It performs any pre-heats specified, as well as the optional IRSL stage. The times, temperatures, heating rates and stimulation wavelengths to be used are specified externally via parameter variables. However, SAR_CYCLE GENERIC could be called by any of a family of Sequences using SAR-like processing cycles, and is not intimately bound to the SAR_TEMPLATE_V1 Sequence file. It is therefore classed as a Universal Include File, and not as a Sequence-Specific file.

Other High Level, Universal Include files which are called by SAR_TEMPLATE_V1, but which might also be used by other SAR implementations, include SAR_INITIALIZER.PSI, which performs initialisation and default value setting of some SAR-specific variables before it calls the generic INITIALIZE.PSI to perform the normal PCIS hardware initialisation; and SAR_MAIN_RUN.PSI, which implements the Main Cycles (Natural, Test Dose and Regen) of the SAR Protocol for a single disk.

A typical Low Level, Command Table-Specific Include file is shown below at Figure 5.3.2. This is the OSL_SINGLE_SHINE.PSI file, which is called (indirectly) from the SAR_CYCLE GENERIC High Level Universal Include file shown above (Figure 5.3.1), as well as from other OSL Include files.

```
OSL_SINGLE_SHINE.PSI

This Include File performs a single OSL Accumulation on the sample in position at the reader.

The Image accumulation is logged to the RunInfo File before the OSL Shine is performed.

Requires:  DiskNum
            OSLColour
            OSLTemp
            OSLTime

Assumes:   Sample is already in position at the reader, lift in Up position, sample temperature set.
            CCD has already been cleaned and is ready for use.
            SET_OSL_TIME and OSL_PREPARE have already been called

Returns:   Sample still in position at the reader, lift in Up position, sample still at OSLTemp.

CmdTblNum 1 2

Log and perform the OSL Shine.

LogImg "OSL: DISK: " DiskNum " at " OSLTemp " Degrees, using " OSLColour "
        LEDs at HIGH Power (80%) for " OSLTime " Seconds."

SetPlus TotalOSLTime OSLTime

ICC 15    'IICU Default Shine command to perform the OSL

' if #DEBUG is ON, DUMP variable stack (once only) to Log files.'

#DUMP 2 - At End of OSL_SINGLE_SHINE.PSI
```

Figure 5.3.2 The OSL_SINGLE_SHINE.PSI Low Level, Command Table-Specific Include File.
The OSL_SINGLE_SHINE Command Table-specific Include file again shows the documentation of its calling interface. It also shows the use of the CmdTblNum Command, to show that this include file is compatible with Command Table Set Numbers 1 and 2 only. It also uses a general programming command, “SetPlus”, to accumulate the total OSL Time for reporting at the end of the run. (This feature is used mainly in the PCIS Simulator environment – see Section 5.7 – to help calculate an estimated elapsed time for the real PCIS Run.) Finally, the file shows one of the Debug group commands, “#Dump”, that is only effective if “#Debug” is set to ON in the calling Sequence file. Section 5.6 gives more detail of the available PCIS Sequence Language commands.

The final PCIS Sequence code example shown here (Figure 5.3.3 below) is of a short Command Table-Specific Include File, that might feasibly be called directly from a Sequence Template Set file, and may thus actually be considered a High Level Include file (though it would more usually be called from another Include file implementing a larger block of functionality). This file, CLEAN_CCD.PSI, contains only a single IICU Command (in addition to the CmdTblNum Command declaring its compatibility with Command Table Set Numbers 1 and 2). Two other examples of High Level, Command Table-specific Include files are INITIALISE.PSI and FINALISE.PSI – see Appendix H.

The reason for creating and using CLEAN_CCD.PSI – rather than just issuing the “ICC 4” command directly from the higher level file – is simply to avoid every file which wants to perform a CCD Clean becoming a Command Table-Specific Include File, and needing to be updated when a new Command Table Set is created and installed. It is far more effective to place the “ICC 4” Command in a separate file to “quarantine” the Command Table dependence.

```
CLEAN_CCD.PSI

This Include File causes the Camera System to read the CCD without storing an image. This has the effect of cleaning any charge off the CCD, thus removing any accumulated thermal noise or cosmic ray strikes.

This should be called immediately prior to commencing any series of image accumulations.

CmdTblNum 1 2
ICC 4  Clean CCD
```

Figure 5.3.3 The CLEAN_CCD.PSI High Level, Command Table-Specific Include File.

An examination of the Sequence Template files and the Include files in the Appendices will reveal how the individual blocks of function may be successively built up to provide, at the top level, an easily read, highly parameterised and extremely flexible implementation of a complex luminescence protocol, or of other broad classes of luminescence experiments.

In particular, the three-file hierarchy composed of the Universal Include files FILTER_AND_FOCUS.PSI and FOCUS_DISKNUM.PSI, and the Command Table-specific Include file RECORD_IMAGE.PSI, provides a good example of the use of User-input commands to control sequence flow, and the devolution of IICU Commands to a single Low Level Command Table-specific Include file.
5.4 PCIS Information Logging

A single PCIS Run can produce many hundreds of images, each representing a different sample and stage of the run. The PCIS_Seq software clearly needs to produce a record of each image and what it represents; and further, a record of all the other processes (pre-heats, beta irradiations etc.) to which each sample is subject. This information file, as with the sequence files, should be human-readable, but may also need to be read by software which automates the analysis of the images and samples (see Section 7.2.1 and Appendix K). It therefore needs a somewhat formalised syntax.

There is also, periodically, a need to delve deeper into the progress of a run, whether because of a runtime error, or simply because the results are unexpected, and the researcher feels a need to confirm that the sample was in fact treated exactly as intended. For this, some lower-level logging capability is required; and because experience shows that runs only ever fail when the low-level logging is switched off, it was decided to always record the low-level logging information.

Finally, I was asked a number of times by other researchers using the Minisys equipment if I could help them repeat exactly an earlier run, on a new sample. This was only ever possible if (a) they could identify the precise Minisys sequence file they had run, and (b) it could be shown (using the Windows’ File Manager – sorry, “Explorer” – utility) that the Minisys sequence file had not been altered since the original run. These conditions could not usually be met, and it was impossible to claim, with sufficient scientific rigour, that the two runs were identical. I was determined that with the PCIS it would always be possible to prove – and to repeat – the exact detail of an earlier run.

5.4.1 Overview

To meet the above goals, each and every run of the PCIS_Seq software produces the following:

- an output directory named for the Run Name entered into PCIS_Seq (and located in the nominated output directory – see Section 5.5). This directory will contain the other PCIS_Seq outputs, and the user will be prompted to copy the image file (created on the camera system PC), into it.
- to meet the requirements to identify the images and the operations performed on each sample: a “RunInfo.txt” file, called just “the RunInfo file”.
- a subdirectory called “Support”, containing:
  - a mid-level log file, recording every PCIS Sequence language instruction executed;
  - a low-level log file recording every command and response on the RS-232 serial port to the Minisys;
  - a copy of the original sequence (.seq) file used for the run; and
  - a subdirectory called “Includes”, containing a copy of all the Include files used.

The last two items support the ability to re-run the precise same sequence, no matter what changes you or other researchers may have made to the original Sequence and Include files. The various items logged by PCIS_Seq are discussed separately below.
The most important output produced by the PCIS_Seq software is undoubtedly the RunInfo file. Indeed, in the vast majority of cases, where the run proceeds normally and produces results within the expected range, the RunInfo file will be the only PCIS_Seq output ever looked at.

The RunInfo file is intended to record all the parameters of the operations performed on the samples, and to identify which images (in the .spe image file produced by the camera system) relate to which sample and processing stage. As will be seen in the (short) RunInfo file shown below at Figure 5.4.1, it also contains some overall run parameters (such as the use of vacuum pumping and Nitrogen flow); and some identifying information about the run which does not directly relate to the samples and their treatment (including, importantly, the Command Table Set in use).

The two lines highlighted in the RunInfo file shown below will be examined more closely when the lower-level logging is described in the following section.

```
17:01:34: {START}: 15/11/2011 Run name: 'SIM Card TL Probe Test Run 1' - Run started.
17:01:34: {INFO}: Input Sequence: C:\Documents and Settings\PCIS\Desktop\PCIS Sequences\TL_Probe SIM Chip V1.SEQ
17:01:34: {INFO}: Output Directory: C:\Documents and Settings\PCIS\Desktop\PCIS SIM Card Work
17:01:36: {ICU}: Command Table Number in use: 01
17:02:21: SAMPLE: 1 (Disk 32): First SIM Chip, large piece GOLD Side, small piece reflective side
17:02:25: VACUUM: Pump started.
17:13:20: NITROGEN: Flow ON.
17:15:36: NITROGEN: Flow OFF.
17:15:36: FILTER: None
17:21:22: INFO: Commence TL Probe Run (Natural and Regen Cycles) for Disk Number: 32
17:21:35: INFO: TL Probe, Regen Cycle No: 1 (Regen Dose: 300) for Disk No: 32
17:27:04: NITROGEN: Flow ON.
17:27:04: TL PHASE: DISK: 32 to 300 Degrees at 1 Degrees per Second, Hold for 0 Seconds.
17:27:11: Image 7 TL: DISK: 32 To 50 Degrees at 1 Degrees/Sec, Relax to 40 Degrees.
17:28:11: Image 8 TL: DISK: 32 To 100 Degrees at 1 Degrees/Sec, Relax to 90 Degrees.
17:29:22: Image 9 TL: DISK: 32 To 150 Degrees at 1 Degrees/Sec, Relax to 140 Degrees.
17:30:32: Image 10 TL: DISK: 32 To 200 Degrees at 1 Degrees/Sec, Relax to 190 Degrees.
17:31:43: Image 11 TL: DISK: 32 To 250 Degrees at 1 Degrees/Sec, Relax to 240 Degrees.
17:32:54: Image 12 TL: DISK: 32 To 300 Degrees at 1 Degrees/Sec, Hold for 0 Seconds.
17:35:01: COOL: DISK: 32 allowed to cool to under 60 Degrees. (T = 59.8).
17:35:03: NITROGEN: Flow OFF.
17:35:08: Image 13 PHOTO: DISK: 32 using RED LEDs.
17:35:17: TIME: Total Beta Irradiation Time: 0 Hours 5 Minutes 0 Seconds.
17:35:17: TIME: Total PreHeat and Hold Time: 0 Hours 5 Minutes 0 Seconds.
17:35:17: TIME: Total Heating Time for OSLs: 0 Hours 0 Minutes 0 Seconds.
17:35:17: TIME: Total OSL Stimulation Time: 0 Hours 0 Minutes 0 Seconds.
17:35:17: TIME: Total of above logged Times: 0 Hours 10 Minutes 0 Seconds.
17:35:31: {END}: SIM Card TL Probe Test Run 1 Run finished - Actual Run Time 00:33:57
```

Figure 5.4.1 A short RunInfo file, from a PCIS Run based on the TL_Probe Sequence Template.
[Note that all of the log file examples shown in this section are taken from a real PCIS run, based on the TL_Probe Sequence Template Set, examining a mobile phone SIM card for any luminescence following an artificial beta dose (none was found – see Section 6.5). The run in question was very short, but includes many of the “normal” PCIS operations, including all of the normal initialisation and finalisation procedures.]

Entries are logged to the RunInfo file by sequence code files containing “LogMsg” and “LogImg” commands (the latter maintaining the internal image-number counter as well as logging the message). Thus the content and format, and even the number of messages logged, is entirely under the control of any user who learns how to write or modify PCIS Include files.

Any standards applying to the format of messages are therefore not enforced, and really have only the status of conventions. Nevertheless, they are important conventions if software is to be produced to automatically process the image files on the basis of the information in the RunInfo file. To date I have tried to adhere to a format that will make both the automated interpretation of the file, and the introduction of new message types as required, as simple as possible.

Should it be necessary to alter the formats used when automated data processing facilities are developed (again, see the discussion of this at Section 7.2.1 and the detailed software specification at Appendix K), this may be easily achieved by the simple editing of the relevant PCIS Include files. The RunInfo logging commands are generally restricted to certain low level Include files, in order to keep the logging commands directly associated with the (usually) “low level” events which they record, and to help maintain consistency in the formats used; so there would only be a small number of Include files that would need to be changed. I have therefore allowed the development of a formal definition of the RunInfo file syntax to be deferred to a later date.

5.4.3 PCIS Low-Level Log Files

There are actually two levels of event logging lower than the RunInfo file: a mid-level file called “InputLog.txt” (or just “the Input log”); and a low level log called “SerialLog.txt” (or “the Serial log”). Both of these files are created automatically by PCIS_Seq for every run, and are placed in the “Support” subdirectory of the Run Output directory.

Various optional low-level logging schemes were tried during the PCIS_Seq software development, but in the end it was decided that:

- optional logging facilities are never in effect when the hard-to-recreate problems occur;
- the logging functions do not affect the performance of the PCIS_Seq software (which is ultimately limited by the speed of the serial port, and nothing else); and
- they are only text files anyway, and are very small compared to the image files; so
- the low level log files would be recorded as part of the output of every PCIS run.

This decision made, it was decided that there should be the two levels of log files described above. The Input log, recording every Sequence language command that PCIS_Seq was asked to execute, would help resolve any problems with the logic or structure of the Sequence and Include files. The
Serial log – recording every command and response on the serial link to the Minisys, on which all Minisys and IICU commands are carried – would help resolve any deeper problems with the PCIS, its software, firmware and hardware components.

Figure 5.4.2 below shows an extract from the Serial log of the same run for which the RunInfo file is shown above. This extract records all of the Sequence language instructions that were executed by PCIS_Seq to implement the PCIS operations covered by the two RunInfo file entries highlighted in yellow in Figure 5.4.1 above.

Note that every RunInfo file entry is mirrored, in its correct place, in the lower level Input log. Similarly, all Input log entries (including the RunInfo file entries) are mirrored correctly to the lowest level Serial log. This simplifies the process of locating an error or problem, and tracing it down to the lower levels to diagnose and rectify it. The mirrored entries are clearly identified; and a location in a lower level file may be found by copying the higher level file entry, and then using a text editor to search for it in the lower level file.

```
17:27:04 : Run Info...TL PHASE: DISK: 32 to 300 Degrees at 1 Degrees per Second, Hold for 0 Seconds.
17:27:04 : (INPUT) : SetPlus TotalPreHeatDegrees TLMaxTemp
17:27:04 : (INPUT) : SetPlus TotalPreHeatTime TLHoldTime
17:27:04 : (INPUT) : Declare PreHeatFinished = False
17:27:04 : (INPUT) : Call TL_PREPARE
17:27:04 : (INPUT) : CmdTblNum 1 2
17:27:04 : (INPUT) : ICC 62
17:27:05 : (INPUT) : ICC 38
17:27:05 : (INPUT) : ICC 33
17:27:05 : (INPUT) : ICC 1
17:27:06 : (INPUT) : ICC 63
17:27:06 : (EXIT) : End of File TL_PREPARE.PSI
17:27:06 : (INPUT) : Call CLEAN_CCD_NO_COMP
17:27:06 : (INPUT) : CmdTblNum 1 2
17:27:06 : (INPUT) : iCmd 4
17:27:06 : (EXIT) : End of File CLEAN_CCD_NO_COMP.PSI
17:27:06 : (INPUT) : MSC LU
17:27:10 : (INPUT) : iCmd
17:27:11 : (INPUT) : Declare StageTemp
17:27:11 : (INPUT) : For StageTemp In PreHeatTempSet
17:27:11 : (LOOPON) : Call TL_SINGLE_STAGE
17:27:11 : (INPUT) : CmdTblNum 1 2
17:27:11 : (INPUT) : If PreHeatFinished = True
17:27:11 : (INPUT) : If StageTemp >= TLMaxTemp
17:27:11 : (INPUT) : Declare EndTemp = StageTemp
17:27:11 : (INPUT) : SetMinus EndTemp TLRelaxDegrees
17:27:11 : (INPUT) : Logging "TL: DISK: " DiskNum " To " StageTemp " Degrees at " TLHeatRate " Degrees/Sec, Relax to " EndTemp " Degrees:"
17:27:11 : Run Info...Image 7 TL: DISK: 32 To 50 Degrees at 1 Degrees/Sec, Relax to 40 Degrees.
17:27:11 : (INPUT) : iCmd 23
17:27:11 : (INPUT) : MSC TO StageTemp TLHeatRate 20 EndTemp R8 20 0 0
17:28:06 : (INPUT) : iCmd
17:28:11 : (INPUT) : GoTo FINISHED_TL_STAGE
17:28:11 : (INPUT) : #DUMP 3 - At End of TL_SINGLE_STAGE.PSI
17:28:11 : (EXIT) : End of File TL_SINGLE_STAGE.PSI
```

Figure 5.4.2 Section of a PCIS Input Log, relating to the RunInfo file shown earlier.
The lowest level of logging performed by PCIS_Seq is contained in the Serial log file (which also contains all RunInfo file and Input log entries, and therefore comprises a highly detailed record of the entire PCIS run). The Serial log contains every command to and response from the Minisys, which of course includes all ICU commands since they are executed via Minisys Operate Relay (OR) and Read Optical (RO) commands. These are correctly interspersed with the Sequence language commands (the Input log entries) which caused the Minisys commands to be executed.

The portion of Serial log file appearing below at Figure 5.4.3 shows the serial port commands and responses which implement the set of Sequence language commands highlighted in yellow in the Input log at Figure 5.4.2 above.

```
17:27:11: {CMD}: RO 2
17:27:11: {RSP}: 0
17:27:11: Input.... {INPUT} : Declare StageTemp
17:27:11: Input.... {INPUT} : For StageTemp In PreHeatTempSet
17:27:11: Input.... {LOOPON} : Call TL_SINGLE_STAGE
17:27:11: Input.... {INPUT} : CmdTblNum 1 2
17:27:11: Input.... {INPUT} : If PreHeatFinished = True
17:27:11: Input.... {INPUT} : If StageTemp >= TLMaxTemp
17:27:11: Input.... {INPUT} : Declare EndTemp = StageTemp
17:27:11: Input.... {INPUT} : SetMinus EndTemp TLRelaxDegrees
17:27:11: Input.... {INPUT} : Loglmg "TL: DISK: DiskNum To StageTemp Degrees at TLHeatRate Degrees/Sec, Relax to EndTemp Degrees."
17:27:11: {CMD}: RO 1
17:27:11: {RSP}: 0
17:27:11: {CMD}: RO 1
17:27:11: {RSP}: 0
17:28:11: .................. above Cmd/Rsp pair repeated a further 547 times.
17:28:06: {CMD}: RS 3
17:28:06: {CMD}: RS 3
17:28:06: {CMD}: RS 3
17:28:06: {RSP}: 91
17:28:06: {CMD}: RS 3
17:28:06: {RSP}: 91
17:28:06: Input.... {INPUT} : ICmd 23
17:28:06: Input.... {INPUT} : OR ON 46
17:28:06: Input.... {INPUT} : OR ON 7
17:28:06: {CMD}: OR OFF 1234567
17:28:06: Input.... {INPUT} : MSC TO StageTemp TLHeatRate 20 EndTemp R8 20 0 0
17:28:06: {CMD}: TO 50 120 40 R8 20 0 0
17:28:06: {CMD}: RS 3
17:28:06: {RSP}: 27
17:28:06: Input.... {INPUT} : IComp
17:28:06: {CMD}: RO 1
17:28:06: {RSP}: 1
17:28:06: {CMD}: RO 1
17:28:06: {RSP}: 1
17:28:06: Input.... {INPUT} : IComp
17:28:11: .................. above Cmd/Rsp pair repeated a further 42 times.
17:28:11: {CMD}: RO 1
17:28:11: {RSP}: 0
17:28:11: {CMD}: RO 2
17:28:11: {RSP}: 0
```

Figure 5.4.3 Section of a PCIS Serial Log, relating to the Runinfo file and Input Log shown earlier.
5.4.4 Repeating PCIS Runs

The exact repeatability of PCIS runs is supported by having the PCIS_Seq runtime software copy the Sequence file, and every Include file used, to the output directory created for the run. This way, the entire sequence used is “carried everywhere” with the actual output data (the .spe image file and the PCIS RunInfo file), all contained in a single directory, eponymous with the PCIS Run Name used.

The Sequence file used is copied to the Support subdirectory of the RunName directory created by PCIS_Seq, along with the low level log files. (Thus the typical user performing a normal PCIS run will never see these files: just the Image file, RunName.spe; the RunInfo file; and a directory called Support, which they know contains the extra stuff if it is ever needed.)

The Support subdirectory also contains a further subdirectory called “Includes”, which contains all of the Universal Include Files that the PCIS run used; and which also contains yet a further subdirectory called “CmdTblnn”, where nn is the Command Table Set Number that was used when the run was performed. This last subdirectory contains all of the Command Table-Specific Include Files that were used by the run. These will all be versions that are consistent with Command Table Set nn.

The reason for this apparently complex structure of subdirectories (shown at Figure 5.4.4 below) is so that a future run of the PCIS, intended to exactly repeat an earlier run, may be performed simply by “pointing” PCIS_Seq at the Support subdirectory of the earlier run, and selecting the saved, original, sequence. All of the include files will then be in the correct place, according to the structure described earlier, which allows PCIS_Seq to select the right version of the Command Table-Specific Include Files. (Alternatively, and better, the saved Sequence file and the complete Includes directory — or indeed just the entire Support directory — may be copied to a different place, and then edited for the new run. Even if only sample descriptions are changed, it is still bad form to change any part of the output data created by a PCIS run, if a proper audit trail of the earlier run is to be retained.)

There are two potential, and small, exceptions to the exact repeatability of the earlier run.

The first possible exception is that some parts of a PCIS Run (even if only the image focussing routines), offer the possibility of user interaction that, to some degree, affects the total course of the run. The present suite of Sequences and Sequence Templates do not offer the potential for user interactions to affect which Include files actually get called, but it is easy to imagine a Sequence which would. For example, the option to perform an IRSL stage during each SAR cycle could be made subject to a user interaction, rather than being parameterised in the Sequence file; and if it was not requested in the original run, then potentially an Include file may not have been used — and would therefore not have been copied to the Support directory — by the original run. It is probably good practice when writing Sequence software not to make parameters which will significantly affect the course of processing subject to run-time user interaction — where they will be recorded only in the Input and Serial logs — but rather, to embody them in the Sequence file itself.

It is also fair to say that if user interactions cause a different Include file to be called, then the user was not attempting to exactly repeat the original run anyway. And, should this situation arise, the repeat run will not fail at that point; PCIS_Seq will automatically find the requested Include file in the PCIS system Includes library, and the new run should correctly achieve its intended effect.
The second potential exception to exact repeatability lies in the possibility that, when the repeat run is performed, the IICU contains a different Command Table Set than was used for the original run.

In this case, the PCIS_Seq software will search the PCIS system library for different versions of the Command Table-Specific Include Files originally used, these ones compatible with the Command Table Set Number now in use. (And will hopefully find them there, in the appropriate “CmdTblnn” subdirectory.) These “equivalent” files for the new Command Table Set should perform exactly equivalent functions to the old ones. Nevertheless, if exact repeatability is sought, it would be better to re-install the original Command Table Set in a spare slot of the IICU (Slot D may be used if it is only for one run, and you don’t mind it being over-written again later – see Section 4 above), and then perform the repeat run using the exact same Minisys and IICU operations as the original.

![Figure 5.4.4 Structure of the PCIS_Seq Output Files and Directories](image-url)
5.5 The PCIS_Seq Runtime Software

This section describes the actual PCIS_Seq program that interprets the Sequence and include files, executes the specified PCIS run by issuing Minisys and IICU commands over the Minisys serial interface, and produces the output files just described.

First, I describe the rather tortuous history of the evolution of the PCIS_Seq program, from the original utility designed to test whether the Microsoft serial communications library calls could be made to work under Visual Basic, and in so doing provide the ability to copy characters typed at the keyboard to the Minisys serial interface, and display responses on the screen — through to the fairly sophisticated and highly functional program that successfully performs all the functions described earlier in this chapter. An understanding of this evolution will help to explain why certain features of PCIS_Seq — and of the Sequence language it implements (itself described in Section 5.6) — operate the way they do, and how that affects the construction of Sequence and include files.

Secondly, I describe the resultant, end software, PCIS_Seq Version 3.5\(^1\) — its installation and operation, and its major features.

5.5.1 Overview of the Sequence Software Development

The entire PCIS Sequence software suite comprises:

- PCIS_Seq, the actual PCIS runtime software;
- PCIS_Dev, the developer's version, incorporating debugging tools; and
- PCIS_Sim, the PCIS Simulator that is used with PCIS_Dev.

PCIS_Dev and PCIS_Sim are offshoot programs from recent versions of PCIS_Seq. They support the PCIS Sequence Development Environment, and are described in Section 5.7 below. The present section describes the gradual emergence of the PCIS_Seq runtime program only.

When the IICU was completed and put into operation, and the PCIS was capable of being operated via a single interface (the Minisys RS-232 command interface), the only tool available for doing so was the extremely limited command line interface of the Minisys Centre.exe application.

A small, quickly assembled utility that improved the ability to run the PCIS step-by-step from a command line appeared to be an essential tool for continued PCIS testing and operation. At the same time, knowing that a full software development to automate the PCIS would be required, I was deciding what platform or language it should be built on or written in. Perhaps surprisingly, Visual Basic 6 (hereafter called VB6) emerged as a logical option for two main reasons.

\(^1\)This last revision of the PCIS software implemented the final, richly flexible syntax of many commands, and cleaned up some of the earlier, extremely terse versions. I had intended to note this significant upgrade with the appellation "Version 4.0"; but I could not discover, and my son could not remember, how to change the major Version number under Visual Studio! So Version 3.5 it became, and Version 3.5 it remains.
Firstly, the Princeton Instruments library of Camera Function Calls, provided as the central component of their Software Development Kit (SDK) and essential to any custom control of the camera system from the PCIS automation software (whether now or as a later enhancement), are all written in Visual Basic (VB). Although it is possible to call them from other language code (and indeed I considered using C instead, being a better structured language than Basic), certain complexities often arise in doing so, and would certainly be avoided if VB were used.

Secondly, my son Brendan, who had dropped out of his own Computing degree, offered to help me build the simple utility referred to above; and he was familiar with VB, but had never written in C. Of the sensible options for the software development, VB was the closest we had to a language and development environment in common. (VB code is produced within the Microsoft Visual Studio environment. I had previously worked with various Basic interpreters and compilers, but never with the Visual Studio development environment. I felt the learning curve would be manageable.)

And so VB6 was chosen for the development of the quick utility, and in so doing its capabilities and appropriateness for the final, full PCIS Automation software would be assessed. A quick utility was developed that proved the ability to read and write from the serial port; and gave the ability to copy lines of text from the keyboard and screen to the Minisys serial port.

As testing and the capabilities of the evolving IICU Command Tables grew, the simple utility (which, for want of a better name, had been called “PCIS_Seq”), grew alongside. The capabilities were extended to include the ability to execute simple commands (which were distinguished from lines of text to be copied to the Minisys by virtue of starting with the hash symbol, “#”), and the ability to invoke a common series of commands stored in a text file by typing the “#Include filename” command. Since the “included” files could themselves contain #Include commands, the beginnings of a simple PCIS Command Interpreter were emerging. The logic was “turned inside out” so that PCIS_Seq started from a file of commands (the origins of the “.seq” file), from which the original command line interface could be evoked by an on-screen button, and some PCIS runs were performed with this early, very simple utility.

At this stage, I still knew that a “proper” PCIS Sequence program – probably, though not necessarily a full two-pass compiler – would need to be written, and was turning over the various design options in my head as the existing PCIS_Seq functionality grew. In the meantime, since it was only a temporary utility anyway, some of the new PCIS_Seq capabilities and command types were added in a somewhat hurried – even ad hoc – manner, and with a terseness of syntax not designed to aid their long term use. Some of these, sadly, remain.

Eventually, and very suddenly one night\(^{18}\), I realised that with the next logical upgrade to PCIS_Seq, it would become capable of the full richness of functionality needed to support the kind of Sequence file structure that has been presented above – i.e. one that would protect the casual user from the need to learn any more than a few pages of easily-understood protocol description, yet allow the more serious user full access to the PCIS’ capabilities.

\(^{18}\) This was the first of two moments of sudden revelation, where I realised that a short-cut could be taken to avoid a significant software development. (The second was realising that the PCIS Simulator could be developed very quickly – in fact over-night – from the then existing PCIS_Seq code.) It is almost certainly due entirely to these two flashes of inspiration that the PCIS was able to be brought to a level of completion which I could accept, and this thesis then written within a time-frame with which the University could accept.
There would be certain compromises which would affect the way Sequence files and Include files would be constructed (discussed in Section 5.6), but the essential features would be supported. The key feature was the ability to write code that used (i.e. Called) blocks of pre-existing functionality based purely on a defined interface to the called block of code, and not in any way dependent on how that block of code was implemented, or on what other blocks of pre-existing code it called.

In computing terms, the environment of the calling code needs to be protected from activities that occur at a lower level of the calling hierarchy. Without such environment protection, a user wishing to “Call” a pre-existing block of code would need to understand in detail the entire called hierarchy of Include files, and every variable that they use, to make sure there were no clashes with variable names already used “higher up” in the hierarchy. Environment protection, however implemented, insulates the calling code and its variables from this interference, and only the interface defined in the header comments of an Include file need be understood for the file to be used. This single feature allows the PCIS user to develop their knowledge of the PCIS Sequences progressively, and only as their need for more detailed control develops.

It turned out that a single pass interpreter (which I had, in PCIS_Seq) could do the job; and a full compiler (which I didn’t have) would not be needed. All further development of PCIS_Seq was then undertaken with the goal of it becoming the final end-user PCIS automation software. But even now some commands (particularly the user interaction command group, which was assembled hastily from a copy of the original command line interface code, and employs too many positional parameters), show distinct signs of their origins in a temporary, own-use utility.

5.5.2 History of the Sequence Software Development

The PCIS_Seq program was never designed and implemented in the traditional sense; even “evolved” somewhat over-states the case. “Accreted” might be the most honest description. The principal steps in the emergence of the final PCIS_Seq application, and the order in which they occurred are summarised (as well as memory can re-construct) in the following list.

1. Initial development of the keyboard-to-serial port copying utility, proving that VB6 supported the necessary capabilities for the PCIS automation software development.
2. Addition of “Hash Commands” (starting with the “#” character), to read a line of data from the Serial Port to an accessible (and displayable) Sequence variable; and to perform polling of the Minisys and of the IICU for command completion.
3. The ability to “Include” blocks of code stored in text files was added.
4. The first PCIS_Seq “inversion” – instead of starting with the command line window (from which an Include (.PSI) file might be run using “#INCLUDE”), PCIS_Seq now commenced by executing a specified Sequence (.SEQ) file – from which the command line window might then be invoked by a “#PAUSE” command, or by the user clicking on the “PAUSE” button.
5. Additional commands were implemented, including the “#ICC n” command, which generated all the Minisys Operate Relay commands necessary to implement the IICU instruction number n, and basic user interaction commands to allow the user to enter information (typically sample names or optical filter types used) into the sequence.
6. The second PCIS_Seq inversion – it was realised that copying “everything not interpreted as being a valid command” to the Minisys was not a terribly error-proof approach, and the “#MSys text” command was implemented to copy the specified text to the Minisys. Any line not starting with a Hash sign was now considered to be a comment, and ignored.

7. A range of pre-defined variable names were implemented, and made available to the Sequence code using “#Set” and related commands. This supported the further parameterisation of sequences, and improved the readability of Sequence and Include files.

8. Since the Hash symbol was no longer necessary to distinguish commands from text to be copied to the Minisys (and is an ugly affectation), it was removed from all commands and indeed, at that time, from the entire PCIS software. However, the code implementing Sequence language commands retained (and retains still) the name “HashHandle”.

9. The conditional or “IF” statement was implemented with a scope of a single line of Sequence code. This single-line style is ideally suited to simple implementation in an interpreter (as opposed to a multi-pass compiler). I borrowed the idea from Hewlett Packard programmable calculators (the only calculators I use; parentheses on calculators confuse me), where it was presumably used for the same reason. A conditional statement with multi-line scope is quite possible within an interpreter, but the extra implementation effort was not justified.

10. Borrowing from the concept of the 1-line scope IF statement, a “FOR” command with a single-line scope was implemented. Implementation of a FOR command with multi-line scope would be significantly more difficult in an interpreter, requiring either the storage of multiple command lines in memory; or the ability to “jump backwards” in a file. Neither of these capabilities exist in PCIS_Seq. The single-line scope often requires that the code to be repeated be held in a separate .psi file, which is called using the single line of code inside the FOR statement. This adds some complexity to the structure of include files for a sequence, but significantly less complexity than supporting multi-line FOR statements in PCIS_Seq.

11. The GOTO command was implemented. Capable only of forward jumps (for the same reason that FOR statements have only a single line scope), these are used mainly to overcome the shortcomings of the single-line IF statement. (The principle of “GOTO-less programming” is an affectation that can be afforded only by those working in richly-structured languages).

12. The “RESTART.FILE” command was implemented. Used in conjunction with the one-line IF statement and forward GOTOS, this command allows the full flexibility of all normal program flow-control constructs to be “simulated” within the low level Include files.

13. The RunInfo file, and the first of the commands which log messages to it (i.e. LogMsg and LogImg), were implemented.

14. The CmdTblNum command was implemented. This important command is used (without parameters) in a Sequence file to have the user enter the Command Table Set Number currently in use; and (with parameters) allows a Command Table-specific Include file to specify the Command Table Set Number (or Numbers) with which it is compatible. This solved, in a simple but functional fashion, the vexed issue of synchronising the Sequence language code implementing a protocol with the ICU Command Table Set in use.

15. The Input Command Line parsing routines were upgraded to properly support the use of single quotes to introduce comments, and of double quotes to delimit quoted string parameters. (This code became surprisingly complex, and it was decided not to bother implementing the facility to split PCIS language commands over multiple lines, but rather just to print Sequence and Include files in landscape format.)
16. Support for parenthesis-delimited "set" variables was added, allowing the development of the "FOR Variable IN Set" construct.

17. A stack was created for Sequence variables, and the DECLARE command was implemented. This allows the creation of meaningful variable names to support a sequence, and made possible the development of readable and easily maintained Sequences. (See Section 5.6 for the behaviour of Set and Declare commands). The #IMPLICIT_DECLARE compiler\(^{19}\) directive was created (the first re-introduction of the hash symbol since step 8) to allow compatibility with earlier Sequence code, which did not "Declare" variables.

18. The critical development that meant PCIS_Seq could meet the long-term PCIS automation needs, and that a separate software would not be required – Environment Space protection was added to the Variables stack, and the INCLUDE command was split into two commands – INCLUDE, which does not invoke Environment Protection; and CALL, which does. The enormous importance of this step should not be under-estimated just because its implementation, using the Variables stack, was so simple. The implementation of the CALL statement – and hence of environment space protection – required only four lines of code (to save and restore stack pointers), wrapped around a call to the existing INCLUDE function.

19. All mouse interactions that could cause the Runtime software to pause execution were eliminated. This required the elimination of all scrollable windows, as well as the elimination of the program's title bar. (Holding down a mouse button on the title bar halts all program execution under Windows. Therefore the PCIS_Seq program is neither re-sizeable nor moveable. This is achieved by removing its caption, which removes its title bar. It also means that there can be no name – only an icon – on the task bar tab for the application.)

20. Due to code changes which generalised the implementation of individual PCIS Sequence commands, and made the more complex commands simpler to implement, the original command-line interface to PCIS_Seq had been unavailable for some time; but its source code still existed. In a moment of bravery, I re-instated the command line interface, and implemented the (supposedly temporary) BACKDOORPASUSE command to allow access to it. To my surprise (considering the enormous changes to PCIS_Seq since its last use), it worked perfectly, and it can now be invoked (with significantly enhanced functionality, especially in the Developer's Edition – see Section 5.7) via the "PAUSE" button.

21. Optional Low-level logging was implemented, initially using a single file. The current two-level permanently-enabled logging scheme was developed later.

22. The PCIS_Seq screen interface was enhanced to support the display of either the latest (i.e. most recent) section of the RunInfo file (the start-up default), or (using the "Swap Display" button), the latest sections of both the Input and Serial logs.

23. Support for PCIS User Names was added. The plain-text user files store the RS-232 Port Number being used; the sequence file and directory; the output directory; and the run name used. This allows them to be defaulted to these values for the next run (by that user), minimising the user interaction required when starting PCIS_Seq. (This is especially valuable during testing, when large numbers of short runs are performed.)

The list goes on.

\(^{19}\) The term "compiler directive" is in such common use for source code statements that set parameters controlling the way compiler software runs, but don't directly generate any object code themselves, that I shall use it here rather than the possibly more technically correct, but unfamiliar, "interpreter directive".
24. The #Debug Command group was added. #DEBUG OFF (another compiler directive) disables all the other debug commands, so that moving Sequence code into production requires only that this single statement be inserted. Many low level include files in the PCIS system library include Debug commands (principally #DUMP, which copies all variables and their values to the Serial Log), which can be activated by any calling sequence containing “#DEBUG ON”.

25. Error handling was improved with the implementation of the #ON_ERROR directive. This declares a set of Include files (often just FINALISE) that, should a PCIS Error occur, will be called before the PCIS halts. Internal PCIS_Seq code to detect different types of errors, report them properly and then stop “gracefully”, was significantly extended and improved.

26. The ABORT_RUN command was enhanced to also allow the specification of a list of Include files to be run before exiting.

27. The Error handling code was improved so that the #ON_ERROR processing was not called if a PCIS Error occurred during the “#ON_ERROR” Include file execution.

28. The PCIS_Seq Source code was split into two versions, using VB6 compiler directives to control two different compilations. The RunTime version, PCIS_Seq, has restricted user controls, and disables all #DEBUG group commands. This ensures that its running (and hence the real-time operation of the PCIS) cannot be accidentally interrupted. The Developer’s Edition, PCIS_DEV (see Section 5.7), allows significantly enhanced user interactions, including the use of all #DEBUG commands, and an enhanced “Pause” window allowing command line entry of additional PCIS Sequence language commands during a PCIS Run.

29. Following the failure, several hours in, of one of the first long automated PCIS runs because I had typed “Set FocusColour = BLUE”, instead of “Set FocusColour BLUE”, I significantly enhanced the flexibility of the syntax of the relevant PCIS commands (principally IF, SET and FOR) to try and reduce the incidence of such events in future. The syntax of some commands was improved, and the range of recognised keywords significantly extended.

30. In the second of the two major development-avoiding shortcuts, I realised one evening that PCIS_Seq could be “turned inside out” just one more time, and a PCIS Simulator created that would allow the development and testing of Sequence code without running on the real PCIS with real samples. Although considerable later improvement occurred (see Section 5.7), the first working version of PCIS_Sim was produced before the sun rose the next day.

31. Code was added to trap any VB internal errors that might occur, and translate them into a PCIS Sequence Error, so that they could be handled “gracefully”, including any requested “#ON_ERROR” processing. The PCIS_Seq code was so stable by that point, already trapping and correctly handling so many error types, that the code had to be deliberately nobbled to allow an error to “slip through” to VB, so that the VB6 Error-trapping code could be tested.

(The list could go on, but doesn’t.)

Given the history of its development, the spaghetti-like nature and overall lack of aesthetic appeal of the final PCIS_Seq VB6 source code is not surprising. Its incredible robustness under the duress of its enormously extended functionality, may be. It is described more fully in the following section.

---

20 Indeed, this is a significant part of the reason the complete PCIS_Seq source code is not submitted as part of this thesis – the other reason being its potential commercial value if re-developed for related applications. However, the sections which are important to a full understanding of the PCIS Sequence language, and of the operation of the PCIS Simulator, are reproduced at Appendices L and M respectively.
5.5.3 The PCIS_Seq Application

This section examines the actual PCIS_Seq application that emerged from the somewhat piecemeal
development process described above - its strengths and features, its limitations, and how to install
and run the software.

5.5.3.1 Overview

PCIS_Seq was developed under, and to run under, Windows. This may seem an odd choice for a real-
time machine control application, but it is familiar, well supported and universally available; there is
an enormous range of Windows software available to assist with the development of applications
(such as the Visual Studio system that I actually used); and, with appropriate precautions, it works
sufficiently well in this role - as was proved by the existing Minisys runtime software being used in
the lab, an Excel spread-sheet application running under Windows 98.

The precautions required are several:

- Don’t expect lightning-fast response to external events, as Windows will take processor
  priority for screen updating and some other user interaction events.
- Don’t let the computer anywhere near a network, ever. If it was stable in 1999, let it still
  think it is 1999, and never see any system software later than that. Avoid any access, ever, to
- No anti-viral software, and turn off all possible security, firewall or other protective features
  that you can. This won’t be a problem, because it will never get near a network.
- Within the application itself, completely avoid any of a large class of user interface features,
  as use of these will cause all processing of your own application code to stop until the user
  action is completed. This includes any moving, re-sizing or even scrolling of any window.

PCIS_Seq simply does not require fast response to events - or even speed in any of its processing.
The PCIS is a real-time machine; but a very slow real-time machine, whose response to anything is
limited by the serial connection to the rest of the machine running at 9600 baud. Software
processing speed is simply not an issue to PCIS_Seq.

The second and third points relate to isolating the underlying computer from any change that could
make things that used to work no longer do so; and from interruptions when Windows decides to
access the web and monopolise the processor while it downloads updates (or whatever it does).
These requirements were easily met by running old PCs with no network cards, with the added
advantage that such machines, running Windows 98 or XP, are free for the asking.

The final point against using Windows - its total commitment to giving total priority to the user
interface - was addressed by careful, but not excessively difficult design of the PCIS_Seq user
interface, and is described later.

So, Windows - a good old solid reliable version - it was, for both development and run time.
The development environment chosen, for reasons discussed earlier, was Visual Basic Version 6 (VB6) running within the Visual Studio development environment. Although the VB6 compiler has some flaws (particularly in the area of declaring, initialising and controlling the scope of variables), it was possible to work around these, and all the other facilities required (including simple serial port communications) were adequately supported.

Visual Studio proved to be an extremely effective and easy to learn environment for developing, testing and debugging code, providing a rich array of event and variable tracking capabilities linked to an interpreted execution of the source code. This was important, particularly given the final complexity and somewhat “unplanned” structure of the PCIS_Seq source code.

Visual Studio supports the simple and effective construction of quite elegant user interfaces (within the constraints of what the developers expected you to want to do, of course), and allows standard procedural Basic code – the code that will actually implement the program that the interface exists to support – to be attached to the user interface items (windows, text boxes, buttons etc.). The standard opening interface to the PCIS_Seq program (after a user-name has been entered) is shown at Figure 5.5.1 below. All the code that actually runs PCIS_Seq – reading the Sequence and Include files, translating them into a series of Minisys RS-232 Commands and Responses, creating the log files, everything described so far – lies behind the “Start” button. Note that at this stage PCIS_Seq has a title bar, and although it cannot be re-sized, it can still be moved around the screen.

![PCIS_Seq User Interface at start-up, before a Run has commenced.](image_url)
5.5.3.2 Running PCIS_Seq

When the PCIS_Seq application is started, a dialogue box is displayed asking for a User Name to be entered. This is used to access the /users/username.pud text file, allowing all of the text fields shown in Figure 5.5.1 above (including the Com Port number) to be set to the values used when the last PCIS_Seq Run was performed under the same User Name. (The new values are stored back when the run starts.) Often a new run requires only that the Run Name be updated before “Start” is pressed.

Figure 5.5.2 below shows PCIS_Seq shortly after a run has started. Initial focussing of the PCIS optics is being performed before the run proper commences. It demonstrates a typical user interaction window under PCIS_Seq – the user may continue asking for more images to be acquired (they will be displayed on the adjacent camera system PC), until they are happy that the focussing is correct.

Also notice that now a Sequence is running, there is no longer a title bar. It is removed when the run starts to prevent the user moving the window, which would halt all processing of the Sequence. (Holding down either mouse button on a title bar will halt all application processing until the button is released.) The solution is not to have a title bar, and the way to achieve this (in VB6 at any rate) is not to have a title – to set it to null. This then means that the application’s tab in the Windows task bar has no name, and it is necessary to attach a recognisable icon to PCIS_Seq after compilation so that it (or a daughter window) can be re-displayed if it gets hidden by another Windows application. For similar reasons, there are no scroll bars on any of the windows which display log file information.

![PCIS Control Centre - v3.5](image)

Figure 5.5.2 PCIS_Seq User Interface after a Run has started, during focussing of the PCIS optics.
Although all efforts have been made to avoid accidental interruptions to PCIS_SEQ, the user may have a need to deliberately pause the machine (perhaps while attending to some previously forgotten task), and the “Pause” button seen in the figures allows this.

In PCIS_SEQ, the only actions allowed by the Pause window (invoked by the Pause button) are to Continue (“it’s ok, I’ve put the samples in now”); or to Abort the run (“I can’t find the samples”) – see Figure 5.5.3 below. (The Pause window of PCIS_DEV allows additional functions, which are described in Section 5.7.)

Also in Figure 5.5.3 it will be seen that the “Swap Display” button has been used, and the interface is no longer showing the RunInfo file, but is now showing the trailing portions of both the Input (mid-level) and Serial (low-level) log files. Again, both displays are trimmed by custom code so that they fit the windows precisely, and no scroll bars appear.

Note that whenever PCIS_SEQ is Paused, all three log files are closed and re-opened, so that the latest entries written to them are “flushed” from the output buffers to disk. This means that if it is necessary to see more than the portion of the files displayed by PCIS_SEQ, the log files can be opened simultaneously under, say, NotePad, and all entries, including the very latest, will be visible. This is a good technique when developing and de-bugging complex Sequence Templates, but has never been required during normal running of the PCIS.

Figure 5.5.3 PCIS_SEQ User Interface showing a Run that has been Paused by the user.
When the PCIS Run has finished, whether successfully or with a PCIS error, an appropriate message is written to RunInfo, and hence gets displayed on the screen. With most Sequences, the PCIS equipment will have been properly closed down (by the FINALISE include file) before the ERROR or Run Finished message is displayed. The application stays open, so that the user can see the completion state of the Run, until the “Exit” button is pressed.

That’s about all that’s involved in actually running the PCIS. (Preparing the PCIS for a run is described in the operating procedures at Appendix J.) From this point on, follow the screen directions displayed by a well-written Sequence for operations such as filter changes; and at the end of the run, copy the Output directory created by PCIS_Seq onto a memory stick, and then copy the Image data from the camera system PC into the same directory (called RunName.dir) on the memory stick.

### 5.5.3.3 Installing PCIS_Seq

The PCIS_Seq software is installed using a “Distribution Package” created under Visual Studio. The Package contains an “install.exe” program which conducts the install. Once the PCIS_Seq executable file has been installed (follow the default install locations, and allow an icon to be placed on the desktop), it may be necessary to create the directory structure for the PCIS System Sequence and include files, and for the User files. The required structure is shown at Figure 5.5.4 below.

---

**Figure 5.5.4** PCIS_Seq Software and System Libraries, installation directory structure.
When the directories shown in Figure 5.5.4 have been created and populated from the install disk, PCIS_Seq should be run so that a UserName file will be created, and defaults can be set for the various directory and file names. The Com Port number being used should also be entered—Windows Control Panel may be required to identify the number to enter. UserName files are created automatically whenever a previously unknown user name is entered – the User Name is for convenience, not security, and no passwords are used. The short “Config” sequence should then be run so that the new UserName.pud file will be written to the Users directory.

The PCIS Library of Sequence and Include files just installed should not themselves be modified. It is recommended that Sequence files, and any Include files that are to be modified, should be copied to a “private” location for the User or project involved. The directory structure required if “private” Include files are to be used is shown at Figure 5.5.5 below. Note that this is not normally required – usually only the Sequence file needs to be copied and modified; the original Include files can be run directly from the PCIS System Library. PCIS_Seq will find Include files in either location.

Due to a bug in the Visual Basic Browse Directories function, it is sometimes necessary to run the installation two or even three times before the “Select Output Location” button will work. This is a well-recognised VB6 problem, but extensive web research disclosed that the only sure cure was to move from VB6 to the “.net” version of Visual Basic, at the cost of considerable extra complexity in quite unrelated areas. Given the small number of PCIS_Seq installs that are likely to ever occur, it was decided that the problem could be lived with.
5.5.3.4 The Sequence Command Interpreter

Behind the “Start” button shown in the figures above lies the code that actually implements the Sequence and Include files – the Sequence Command Interpreter.

I have made the distinction between an interpreter and a compiler several times, but now need to define precisely what I mean, and what distinguishes the two. For the purposes of this discussion:

- A compiler reads through the source code twice (hence “two-pass compiler”), building a symbol table on the first pass and resolving all references on the second. An interpreter makes only a single pass of the code (as it executes it), and no symbol table is built.
- The symbol table and reference resolution allows the compiler to enforce typing of variables (also of Function calls etc.), allowing it to detect many types of source code error before execution of the code occurs. The interpreter cannot detect type mismatch or unidentified symbols until the code is actually executing.
- A compiler builds a machine-readable (but not human-readable) “object” file (familiar in DOS and Windows as an “exe” file), which will be executed later. The interpreter executes the code as it reads it, and no intermediate file is required. It is generally understood that the compiler’s executable will run faster than an interpreter, as the compiler usually undertakes code optimisation when it writes the executable. An interpreter has no opportunity to do so.
- An interpreter, since it doesn’t see code until it executes it, is less capable than a compiler in implementing asynchronous event routines, or parallel “streams” of simultaneously-executing code – essential features for building flexible user interfaces.

In general, there is a tendency amongst the computer cognoscenti to belittle interpreters, and regard them as “the poor man’s compiler”. The belief is that:

- Interpreters don’t enforce strong typing and so encourage poor coding techniques;
- Interpreters are slow; and
- Interpreters don’t support modern object-oriented programming techniques (whatever they are this week).

I shall argue here that an interpreter implementation is well suited to the task required of PCIS_Seq, and in fact has several advantages over a compiler; and that the recognised limitations of an interpreter are of little or no import in this case.

PCIS_Seq, and the Sequence language that it implements, were designed to fit one purpose only; running the PCIS. This is inherently a sequential process. The PCIS can only perform one function at a time, so a plain old-fashioned procedural language is entirely appropriate. There is just no need for the parallel threads or other “object-oriented” techniques that a modern two-pass compiler can support. And as explained before, speed and performance are just not an issue.

The differences between a relatively simple interpreter and a more complex two-pass compiler that actually do affect the design, construction and operation of PCIS_Seq are examined below.
5.5.3.4.1 Limitations of the Interpretative Approach

The fact that the one-pass interpreter neither builds a symbol table nor resolves the source code references (variable names and procedure — or Include file — names) prior to execution is a very significant shortcoming, with serious implications for the early detection of Sequence errors.

Early detection of errors — that is, before the actual execution of the code on the target machine — is particularly important in the case of the PCIS, as it is with any scientific machine handling possibly irrereplaceable samples. The natural luminescence signal can only be shone out once. If a run fails part way through (because of a mis-spelt variable name, or an errant “equals” sign), it may not be possible to simply correct the error and repeat the run. The only sample may have been destroyed.

This is an extremely serious issue, and I had always recognised that it would need to be addressed in some fashion. With the decision to go with the one-pass interpreter, rather than starting a separate development to produce a two-pass compiler, the problem became even more acute.

Source code errors — in any programming system — may be considered in two classes:

1. Syntactic errors and typographic errors — essentially, poor grammar and poor spelling respectively. These are the most common forms of error, especially when only relatively small changes are made to existing code (the usual way of developing new Sequences). A compiler will almost always detect these errors, either as an “illegal instruction” (anything a compiler doesn’t understand is called “illegal”), or as an unresolved reference (since the mistyped name will not be found in the symbol table).

2. Semantic errors — essentially, errors of logic. In this case, a perfectly legal (and well spelt) set of instructions specifies a plausible behaviour, but not the one you intended. Although they can arise from poor typing, they are more usually due to unclear thinking. Semantic errors cannot be detected even by a compiler, since a compiler cannot know what you meant to ask for, and must accept any syntactically legal source code, however illogical.

So, a compiler implementation of PCIS_Seq would have detected (most) syntactic errors, but not semantic errors. And semantic errors have an equal, or even greater potential to destroy samples than syntactic errors, since a syntactic error may cause the run to fail before the samples are affected, whereas a semantically incorrect Sequence will run the incorrect procedure to completion.

Given the critical importance for the PCIS of not destroying samples with software errors, a PCIS Sequence development environment was required that would detect, or make visible, both syntactic and semantic errors before any samples were placed in the machine. This could be achieved with a PCIS Simulator, which would allow the PCIS_Seq software and the entire target Sequence to be run in a “safe” environment, and the output RunInfo and Log files examined to confirm complete correct operation, before the Sequence was run on the real PCIS with real samples. The combination of an interpretive PCIS_Seq with a full PCIS simulator-based development environment would provide even better protection against software errors than a two-pass compiler with a symbol table.

I was extremely concerned at the effort and resources that the development of a full PCIS simulator would require. Section 5.7 explains the short-cut that allowed its very rapid development, and describes the PCIS Sequence Development Environment that resulted.
Since, as I have stated, the PCIS does not require speed, and is suited to a simple procedural style of programming, the only remaining limitation of the interpretive approach is that certain types of code construct (those involving backwards looping or jumping) are more difficult to implement.

PCIS_Seq lacks a number of code constructs that are typically supported by general purpose procedural computer languages. For the most part, these limitations exist because the relevant construct would have taken some effort to implement, and the construct was not essential for the writing of well structured, effective Include files.

The one construct that is extremely difficult to support in the interpretive paradigm is that of Loop commands (For, While, Loop etc.) with a scope of more than one line of source code. (Supporting a single such loop would not be excessively difficult; but supporting a general capacity for multi-line loop commands – including within conditionals or within other loop commands – would be quite complex and time-consuming.)

The work-around for this limitation is simply to place the multiple lines of code into a separate Include file, and to Call (or possibly Include) that file from the supported single-line “For” statement. This adds an extra level to the calling hierarchy of the Sequence; and adds a small degree of extra complexity to the construction of PCIS Sequence and Include Files.

Since this extra complexity is generally limited to the low-level include files, so that the normal PCIS user is quite insulated from it; and since the complexity it adds to the Include files is actually quite minimal, the limited code constructs it imposes on the PCIS_seq interpreter are not considered to be a significant issue.

5.5.3.4.2 Benefits of the Interpretative Approach

The interpretive approach has one significant benefit that is often over-looked; it does not produce an object (executable, or .exe) file. Although the use of an executable object file does allow greater code efficiency, and results in a faster performance of the software, it also represents a part of the system that is not human-readable, and so cannot be easily and directly checked for errors. This may be acceptable for a mass-produced compiler such as Visual Basic, where the performance of the compiler has been tested by millions of users and the chances of an error intruding at the compilation stage is slight. For the PCIS however, where the end-user testing and proving of the software could never be as comprehensive, avoiding the introduction of an inscrutable and potentially error-prone stage to the software is far more important than outright processing speed.

It cannot be denied, however, that for me as the sole developer of the PCIS system, the over-riding advantage of the interpreter was the enormously reduced development effort required, especially once the supposedly temporary PCIS_Seq utility had reached the level of functionality that it had acquired when the decision was made not to undertake a separate compiler development.

By further developing the PCIS_Seq interpreter utility into the final PCIS automation program, and then designing the Sequence and Include code to operate in that environment, it became possible to deliver the fully automated PCIS within a manageable time frame.
5.5.3.4.3 Variable Scoping and Environment Spaces

The key facility that allows clean re-usable Include file source code to be developed, and which allowed the PCIS_Seq interpreter to be used in preference to a separate compiler development, is the ability to control the hierarchical scope of PCIS variables, and therefore to provide protection for the environment space of the calling code. This section discusses those facilities, and their particular implementation in PCIS_Seq.

Early PCIS_Seq source code included a range of pre-declared and pre-named variables that could be used by the PCIS Sequence and Include code. (Note that although PCIS variables are sometimes interpreted as being Numeric or Boolean, all variables in PCIS_Seq are actually implemented as strings. Therefore assigning types to the pre-declared variables was not an issue.) It soon became clear that use of these arbitrarily-named pre-defined variables was very limiting to the capabilities that could be easily implemented, and tended to make the resultant code unacceptably difficult to read, understand and modify. There was clearly a need to allow users to declare their own variables, with names that were meaningful to their role in the code being developed.

I therefore implemented a two-part list, containing variable names and values respectively; and a Declare statement that allowed users to create new variables with new names.

At the same time, I was pondering the vexed question of whether the statement that caused a new file of code to be opened and executed should be named “Call” (the more intuitive term for “calling” a separate pre-defined block of function), or “Include” (the existing name, and technically a more accurate description of how the statement operated at that time).

And then inspiration struck! The solution, which would greatly enhance the functionality of PCIS_Seq, and ultimately avoid the need for a PCIS compiler development, was to have both!

The “Include” statement would continue to operate without change, acting in all respects as though the code in the Included file appeared in line in the calling file, in the place of the Include statement. The “Call” statement, however, would save a pointer to the end of the current list of variables, and would restore that pointer on exit from the Called file. This would effectively remove from the list any variables created by the Called file (or by any files it in turn Called or Included), restoring the environment of the original calling code before execution returned to that module. This would provide the level of environment space protection necessary to support the creation of independent re-usable modules of function as callable Include files.

The saving and restoring of pointers through multiple levels of file Calls is supported by the facilities of VB6 for the protection of local variables, and thus the functionality and complexity is effectively “borrowed” from the VB compiler, rather than having to be implemented (by me) in PCIS_Seq.

When a file is Included (rather than called) these stack management and environment space protection features are not applied. This allows the Include statement to be used for the traditional programming “Include” function – using header files that declare and initialise variables that will be available to the Calling Sequence code. This is seen in the frequent use of “Include INITIALISE” in Sequence template files, and neatly hides the necessary declarations.
5.6 The PCIS Sequence Language

This section describes the instructions and features of the PCIS Sequence language which the PCIS_Seq application implements. This is the language in which all of the Sequence code and Include files examined earlier were written. It was developed by me specifically to meet the needs of specifying a luminescence Sequence to be run on the PCIS equipment.

A detailed specification of the syntax and function of each of the Sequence Language commands, in alphabetic order, appears at Appendix I. The present section is intended to give a more general overview of the language and its most important features.

5.6.1 Overview

Although the Sequence language implemented by PCIS_Seq is of sufficient richness and complexity to satisfy the Turing requirements of a Universal Computing Machine, and so in theory is capable of undertaking any information processing task that any computer is able to perform\(^{21}\), it was never intended to be used for general data processing applications. Its sole purpose is to control the PCIS, and this is clearly evident in the instruction set it supports.

However, as has been seen in the Sequence and Include files examined earlier, the language is clearly sufficient to enable clear and maintainable code to be produced, implementing luminescence operations (Protocols or Sequences) of significant complexity. Whatever limitations remain in the language and the instructions it supports are there because, fundamentally, they do not hinder the production of good Sequence code, or the effective operation of the PCIS.

It remains true, though, that while the PCIS Sequence language allows good programming techniques, it does not enforce them.

5.6.2 The PCIS Sequence Language Instruction Set

This section examines briefly the main members and general features of each of the main classes or groups of instructions supported by PCIS_Seq. (How the instructions are best used in combination to implement functional and maintainable PCIS Sequence code is addressed by the brief Application Notes in the following section.)

The commands will be reviewed below in the following groups: PCIS-Specific Commands; Program Flow Control Commands; General Programming Commands; User Interaction and Logging Commands; and Interpreter Directives and Debug Commands.

\(^{21}\) Academic discussions of programming languages and their capabilities generally ignore anything to do with I/O, mainly, I suspect, because it is messy and difficult to formalise. Any ability to output the results of a computation is enough for a language or machine to qualify, on that count, as a Universal Computing Machine.
Note that the current Version 3.5 Sequence Language supports a number of “heritage” commands, which were current in PCIS_Seq Version 2 or earlier, and were retained to maintain compatibility with earlier Sequence code. With the upgrade of the entire PCIS to IICU Version 2, backwards compatibility with earlier versions of PCIS_Seq is no longer required, and these commands will not be supported in the future when PCIS_Seq Version 4 is released. The Heritage Commands are included in the formal Language Definition at Appendix I, but are not discussed in this section.

5.6.2.1 PCIS-Specific Commands

**MSys, MSC, MComp, ICmd, ICC, IComp, ICountComp, l, CmdTblNum, ForAllDisks, ReadSer / ReadSerial, ReadSerDiscard, FlushSerialBuffer**

These commands exist purely to support the interaction with the PCIS components, and would not exist in a “general” programming language.

The group **MSys, MComp** and **MSC** are used to issue a command to the Minisys over the serial port; to poll the Minisys for completion of the command; and both (issuing the command and polling for its completion), respectively. The **ICmd, IComp** and **ICC** group perform the same functions for IICU commands, automatically generating the required Operate Relay and Read Optical Minisys commands. **ICountComp** polls for IICU Command Completion, but also counts the transitions on Optical Input 9, logging each to the RunInfo file as an acquired image, until the IICU command completes. This is used to support the “Rapid Repeat Shine” mode of the IICU (see Chapter 4).

The **I** command issues the “!” or Reset command to the Minisys, and reads and discards the returned Minisys Software Version Number. It is used by the “Initialise” Include file.

**CmdTblNum** exists to support the co-ordination of Sequence and Include file code with the commands in the currently-installed IICU Command Table Set. Its uses – either to request the number of the Command Table Set in use, or to specify the Set Numbers(s) with which a Command Table-Specific Include file is compatible, have been described earlier.

**ForAllDisks** is a convenient short-hand, allowing an operation to be performed on each of the sample disks currently in use in a PCIS Run. It is translated internally into either “For DiskNum equals FirstDiskNum to LastDiskNum by SampleSpacing”, or to “For DiskNum in DiskSet”, depending which approach the Sequence uses to specify the sample disk numbers being used. See the extract from OSL_Template_V1.Seq shown earlier at Figure 5.2.1 for an explanation of the two different ways to specify the sample disks being used.

The Serial Communications commands could equally have been placed in the General Programming Commands, but in PCIS_Seq the serial port is used exclusively for Minisys commands and responses, and so they have been included here. **ReadSer** (or ReadSerial – they are the same command) reads a (CrlF terminated) line of characters from the serial port into a PCIS_Seq “special variable” called varSerialIn. **ReadSerDiscard** does the same but discards the result. **FlushSerialBuffer** reads the serial port repeatedly until a serial read time-out occurs. It is used by Initialise.psi to clear the port to ensure successful initialisation of communications with the Minisys.
5.6.2.2 Program Flow Control Commands

Include, Call, For, If, Goto, Label, RestartFile, AbortFile, AbortRun, PauseTime

Include and Call have already been discussed. Basically, “Include” executes the statements in the named Include file exactly as though they had appeared in the original file in place of the “Include” statement. “Call” provides a new environment space for any variables declared by the called file, and removes them, restoring the original variables environment, when control returns to the calling file.

For and If each have a scope of only the single statement immediately following the command. “For” will cause the following statement to be executed once for each of the values specified. “If” will cause the immediately following statement to be executed if and only if the attached conditional expression evaluates to true. (The actual implementation is, that if the condition evaluates to False then the following statement is read and discarded. If it is true, no action is required, and execution continues – at the statement controlled by the “If” command.) “For” and – especially – “If” have quite complex and variable syntaxes, which are specified in full at Appendix I.

GoTo and Label are used to implement forward jumps (only) within a Sequence or Include file. The “GoTo LabelName” statement causes PCIS_Seq to discard all lines up to and including one containing the statement “Label LabelName”. If no such line is found, a PCIS Error is caused. There is at present no support for jumping to a label earlier than the GoTo statement – constructs using the RestartFile command are used instead (see the Application Notes in Section 5.6.3).

RestartFile, as its name implies, causes the execution to re-commence from the start of the current Sequence or Include file. It is the only way to implement a backwards jump in the code, and is used in routines like the image focussing, which need to be repeated an unpredictable number of times.

AbortFile causes execution of the current Include file to stop, and control to return to the statement that called (or included) the file. It is like an “exit” statement in a function call, and allows Include files to have multiple exit points.

AbortRun causes the entire PCIS run to terminate, following the execution of any Include files specified in the AbortRun command (typically “Finalise” and “Report_Times”). This is usually used when a Sequence detects that a run has gone wrong, and it cannot recover. It allows the PCIS hardware to be shut down “gracefully” pending the presence of an operator.

PauseTime causes all execution of the PCIS_Seq application to stop for the number of seconds specified. This is a dangerous command, since it leaves the PCIS (and particularly the Minisys) unattended. If the Minisys times out while the Run is paused, it will shut itself down and the run will fail. (The maximum value the Minisys time-out can be set to is 300 seconds.) The PauseTime command, if it is used at all, should only be used with a short specified time period. Generally, it is better to ask the Minisys to implement the pause, and then poll it until the command completes (using “MSC PA nnn”). This avoids any risk of a Minisys time-out.
5.6.2.3 General Programming Commands

Set, Declare, SetPlus, SetMinus, SetTimes, SetDiv

This group of Sequence Language commands is surprisingly sparse, but meets the needs of the PCIS.

Set is used quite simply to set the contents of a variable to the value specified. (Note that all PCIS_Seq variables are strings, although for some operations they may be interpreted as numbers.) Unless Implicit_Declare is specified (which, as stated earlier, is not recommended), the variable must exist before it can be set.

Declare is the statement used to create PCIS variables. Declare will create the named variable in the internal PCIS_Seq variable stack, in the environment space of the current include file, even if a variable of that name already exists outside the current environment space. ("Declare" may also specify a value to which the newly-created variable will be initialised.) Since any search for a variable name starts at the top of the stack (i.e. at the limit of the current environment space), it will always be the most-locally-declared copy that is found and affected, protecting any variables of the same name Declared in "higher" calling files (i.e. "lower" in the stack) from being destroyed.

A special feature of the Declare statement is that, if the variable already exists in the current environment space then the entire Declare statement is ignored. This allows "RestartFile" to be used without overwriting the value of any variable Declared during the first execution of the file, preserving information across the "backward jump" or "loop" in the code. This feature is used to implement the repeated focussing on multiple disks performed by the "Filter_And_Focus" include file, and the files that it calls.

Although PCIS_Seq implements user-defined variables and effective variable scope management, there is no expression evaluation supported at all. SetPlus, SetMinus, SetTimes and SetDiv are used instead, allowing the Sequence writer to parse their own expression, and implement the individual components. Each command sets the first parameter to its original value Plus (Minus, Times, Div) the value of the second parameter. The operations are carried out as integer or real operations according to the value of the variables. If the variables do not parse as numbers, a PCIS Error occurs.

5.6.2.4 User Interaction and Logging Commands

LogMsg, LogImg, UserMsg, UserLogDesc

LogMsg logs a message to the RunInfo file (and so also to the Input and Serial logs). LogImg performs a similar function, but logs the message as a numbered "Image:" record, and maintains the Run Image counter. (The Run Image counter is also maintained, and Images logged, by the ICountComp statement – see above.)

22 I dislike Infix notation (and all those (very confusing) brackets ("()")), and refuse to implement yet another Infix expression parser. I might have enjoyed implementing a Post-fix, or Reverse Polish Notation parser, but few other PCIS users would have understood it. How our schools have failed us!
UserMsg is a somewhat clumsy, seven-parameter command that puts up a message box with one to three buttons. Although both the labels and the descriptions of the buttons – as well as which of the three buttons will be shown – can be specified in the command, the function of each button is fixed. The three functions are “Restart File”, “Continue” and “Abort Run”. Using these three capabilities to implement effective user choices takes some ingenuity in the construction of the Include files, and the implementation of a better-structured command for user interactions is one of the higher priority items when further development is done on PCIS_Seq.

UserLogDesc allows a user-entered or user-modified string to be written to the RunInfo file. It is currently used (by Filter_And_Focus.psi) to record the optical filters installed in the source turret. The filter description specified in the sequence code is displayed to the user, who may then modify it to match the actual filters installed, and the modified description is logged.

5.6.2.5 Interpreter Directives and Debug Commands

#Implicit_Declare, #On_Error
#Debug, #BreakPoint, #Dump

The Interpreter Directives that are effective in PCIS_Seq are “#Implicit_Declare (On or Off)”, and “#On_Error (list of Include files)”. The Debug group are only operative in PCIS_Dev. This allows Debug commands used for Sequence development to be left in the code when it moves to production – they will not affect the operation of the real PCIS under PCIS_Seq.

#Implicit_Declare means that an attempt to set a variable that does not exist, instead of causing a PCIS Error and terminating the Run, will cause that variable to be created, and then Set its value. This was implemented purely to provide compatibility with Sequence code written before the Declare statement existed. No such Sequence code still exists. Since #Implicit_Declare seriously reduces the ability of the interpreter to recognise typographic errors, its use is strongly discouraged.

#On_Error allows the Sequence writer to specify a set of Include files to be executed, before the PCIS stops, should a PCIS Error occur. Usually a Sequence should include “#On_Error Finalise” to ensure that the PCIS equipment is stopped properly pending the arrival of an operator.

#Debug controls the operation of the entire debug group of commands. These operate only in PCIS_Dev (used in the PCIS Sequence Development Environment – see Section 5.7), and then only if there is a “#Debug On” statement. Otherwise, all #BreakPoint and #Dump statements are ignored.

#BreakPoint causes the Run execution to stop at that line of code, and the user interaction window of PCIS_Dev to be displayed. This allows the user to undertake a variety of debugging actions, including single-stepping the code, or entering Sequence Language commands from the keyboard.

#Dump causes all of the PCIS Sequence variables currently Declared to be written, with their values, to the Serial log file where they can be examined. Options on the #Dump statement allow dumping of the values only on certain selected entries to the code, rather than on every execution of the Include file. This helps limit the Dumps to the Serial log to those that are actually wanted.
5.6.3 Sequence Language Application Notes

This section contains a few explanatory notes on how certain PCIS Sequence Language statement types should be used, alone or in combination, in order to effectively implement the types of constructs that good Sequence code requires.

Because of the idiosyncrasies of the Sequence language, there are a few things that need to be done somewhat differently than they would be in a typical general-purpose programming language.

5.6.3.1 The Variables Stack

It has been explained that the user-declared variables are maintained in a LIFO (Last-In-First-Out) stack as a means of implementing control over the hierarchical scope of the declared variables. The stack is considered to be built from the bottom up; but when the stack is searched for a variable name to obtain (or set) its value, it is searched from the top down, so that the most recent occurrence of the variable name is used – that’s what environment space protection for the parent code is all about. But the picture is just slightly more complex than that.

When PCIS_Seq is started, the stack is “seeded” with a number of constants and keywords, and a pointer is created showing the “top” of this group of entries. Constants and keywords created here as “variables” include “NULL” (as “”), “BLUE” (as “BLUE”), “RED” (as “RED”), “EQUAL” (as “EQUAL”), “IN” (as “IN”) etc. Note that all of the names and the assigned values are in upper case.

When PCIS_Seq wants to “Get” the value of a “variable” (it may actually be a keyword or constant), it searches the variables stack from the top to the very bottom (and generates an error if the name is not found). When PCIS_Seq wants to “Set” the value of a variable, however, the stack is searched from the top down only to the “top of Constants’” pointer (and generates an error if the name is not found – unless #Implicit_Declare is ON, in which case it Declares and then Sets the variable).

The Declare statement will search the Current Environment Space (at the top of the stack), and the Keywords and Constants Space (at the bottom of the stack), and will not create the variable if the name is found in either of these two places.

The purpose of these arrangements is twofold:

1. It prevents the values of keywords and constants from being changed (and protects against the names being used as variables and those values being changed); and
2. It provides case-insensitivity for all these keywords and constants. Since variable names are already case-insensitive (they are all translated to upper case before being created or searched for), and now the keywords and constants get treated in the same fashion, they automatically inherit the case-insensitivity. This eliminates an entire class of potential sequence code errors (“True” or “true” instead of “TRUE” etc.).

The behaviour of the Declare statement when the variable name is found to exist already in the Current Environment Space is discussed in Section 5.6.3.3, “Declare and RestartFile”, below.
5.6.3.2 Call and Include

“Call” creates a new environment space for the called Include file, and removes any variables Declared by that file (or any other code files it Includes or Calls) when control returns to the original calling (or “parent”) code. “Include” does none of these things.

“Include” should only be used where you specifically want variables Declared within the Included code to be available to the parent file code when control returns there. This implements the function of “header” files, as they are called in C and some other languages. Their specific purpose is to Declare (and usually initialise) variables for the “parent” code. This promotes code re-usability, and avoids cluttering up the parent code with a whole lot of Declare statements.

A good example of this use of the “Include” statement is provided by “SAR_Template_V1.seq”, which uses the Include statement to execute “SAR_Initialise.psi”, which in turn uses Include to execute the generic “Initialise.psi”. Because this occurs at the start of the Sequence file, the hierarchical scope of variables means that all of the variables created by the two sets of initialisation code are available to the whole of the rest of the SAR Sequence code.

If an Include file does not contain any Declare statements, or “Include” any files which contain Declare statements, then technically it does not really matter whether it is Called or Included. Nevertheless, in order to protect the environment space of the parent code it is advisable always to use “Call”, unless there is a specific reason to use “Include” instead.

5.6.3.3 Declare and RestartFile

As noted above, Declare will not create a new instance of a variable name that already exists in the Current Environment Space. (For one thing, it would make the previous occurrence impossible to access – with either Get or Set – since the new occurrence would always be found first.)

Nor will a Declare statement in such a situation change the current value of the existing variable, even if an initialisation value appears as part of the Declare statement. The reason for this specific behaviour is quite subtle, and is to do with the use of the RestartFile statement to implement a form of backward looping in Include files.

It is occasionally necessary (especially when the number of times something occurs depends on a user interaction) to actually jump backwards in the code – and the only way to do this in the PCIS Sequence Language is with the “RestartFile” statement. (A recursive call to the same Include file could also be used – and is supported by PCIS_Seq – but the resulting construct will generally be more complex than the method described here.)

If the RestartFile statement is used, but the code needs to know whether it is the first or a subsequent iteration (or perhaps the actual iteration number) then a problem might arise. If the loop counter is maintained in a local variable (as it should be if the external environment space is to be protected from unwanted effects), then it must be Declared and initialised inside the called
Include file. But then it would be re-Declared (or at least re-initialised) when the file was re-started, and so the iteration count would be lost.

The solution to this is the one stated above – that a second or subsequent attempt to Declare a variable in the one environment space will be completely ignored by PCIS_Seq. Of course the variable can still be Set, so the implementation will look something like:

```
Declare       LoopCounter   0
SetPlus       LoopCounter   1
{  Code implementing Include file function ... }
If  Condition
  RestartFile
{  Finalisation code ... }
```

This achieves the effect of maintaining the loop counter across the series of file re-starts, but allowing the LoopCounter variable to be removed when the file finally Exits so that no other variable of the same name used by an “ancestor” file will be affected.

### 5.6.3.4 For Statements and the Use of Set Variables

The lack of simple backwards jumping capabilities in a one-pass interpreter such as PCIS_Seq makes it extremely difficult to implement a “For” statement (or any of the related looping constructs such as “While”), that has to execute the same piece of code multiple times. It would have been possible to store the code inside the For loop in memory while it was interpreted; but it would have been as much work as doing so for all the Sequence code – at which point I might as well have implemented a symbol table and a full two-pass compiler – and I didn’t want to do that.

The solution was to implement a “For” statement with a scope of just one line of source code. This implementation fits easily into the interpretive paradigm, and if necessary the one line of code in the For loop may be a Call statement to an Include file containing any number of statements. With this easy work-around, there was no necessity to try and implement a much more complex solution.

I have said that the For statement also supports a form of “For Variable in Set Variable ... “. Now there is actually no such thing as a “Set Variable” in PCIS_Seq – all variables are in fact strings. But a string consisting of a series of comma or space separated values enclosed in parentheses will be treated as a “Set Variable” for the purposes of this “For” construct only. This is used in OSL_Template_V1 to support statements like “For RegenDose In RegenDoseSet ... “, allowing a simple set-like specification of the set of SAR Regen doses to be used.

In fact the parentheses (like the commas) are optional, and are stripped off, so that a construct like:

```
For  StimulationColour In  IR Red Blue, UV
    Call   OSL_Routine
```

may be used. OSL_Routine will be called four times, with StimulationColour set to IR, RED, BLUE and UV (because of the automatic case-conversion) for the successive calls.
Note that the following form may be used with no need to previously Declare the loop variable “I”.

        For I Equals Value1 To Value2
            { Single line of loop code ... }

The Call statement automatically creates a variable called “I” for the private use of each called Include file, precisely so that this very common style of For statement, where the loop control variable is of no consequence outside the For loop, may be used.

5.6.3.5 If Statement Constructs and the Use of GoTo

As is the case with the “For” statement, the “If” statement has a scope of only a single line of Sequence source code. While it would have been possible to implement an “If ... Then ... Else ... EndIf” block construct within the PCIS_Seq interpretive paradigm, it did not seem necessary to do so.

The If statement can of course be written to control the execution of a Call to an Include file which may contain as many lines of code as are required. And where a variable simply needs to be set to one of two values, “Else ...” type constructs can be avoided with a structure like:

    Set      EditOK = True
    If Condition
        ' Test for an edit failure
        Set      EditOK = False
    EndIf

Where it is really necessary to execute multiple lines of code in both the “If” and “Else” parts, a structure like the following, using GoTo statements, is simple and clear.

    If Not Condition
        GoTo Else_Code
        { ... “If Code” – executed if Condition is True ... }
    GoTo End_If
    Label Else_Code
        { ... “Else Code” – executed if Condition is False ... }
    Label End_If
        { ... Rest of module code ... }

A “GoTo” statement always finds the next following occurrence of a “Label” statement with the matching label name; the rest of the file is not looked at. Therefore multiple “If ... Else” constructs like the one above could appear sequentially in a source code file, all using exactly the same label names. The correct instance of the “Label” statement would be found in each case.

It is only if nested “If ... Else” constructs were to be used that complications would arise, and unique names would need to be used. The same complexity would have affected any implementation of an “If ... Then ... Else ... EndIf” construct directly into the PCIS Sequence Language, and is the main reason (given the existence of the simple workaround shown) that it is not supported.
5.7 The PCIS Sequence Development Environment

I had always known that to “close the loop” on the PCIS development, and to make it realistic for other researchers to develop and use their own Sequences to implement their own, novel luminescence experiments (on rare or valuable samples), a development environment for Sequence and include files would be mandatory. Until this step was achieved, I would be unable to regard the PCIS development as acceptably complete.

This section tells the story of that development, and describes the PCIS Sequence Development Environment that was produced.

5.7.1 Overview

The PCIS Sequence Development Environment (sometimes called the Software Development Kit – or PCIS SDK – since Sequences are the PCIS software that a user sees) comprises two software components and, at a later stage, a PCIS_Simulator hardware unit as well.

The two software components are the Developer’s Edition of the PCIS Run-time code, PCIS_Dev; and the PCIS Simulator software, PCIS_Sim.

PCIS_Dev was an obvious off-shoot of PCIS_Seq. Indeed, one could almost say the opposite was true, that PCIS_Seq is generated from PCIS_Dev by disabling all the features that might possibly interfere with a real-time run on the real PCIS (using real samples). These were basically the features that allowed user interaction with the running software and the user interface. Once the split had been made, additional useful user interaction features could be added to PCIS_Dev without fear of their possible effect on real PCIS Runs. The final (or at least the current) features of PCIS_Dev are described in the following section.

The development of the PCIS Simulator, by contrast, appeared to be a non-trivial problem.

Without having thought about it in any great depth, I had always assumed that a complete PCIS Simulator, although definitely required, would entail a reasonably significant new software development. The extent to which my supervisors were aware of this was a subject I had been carefully avoiding.

Then one night, in the midst of working on the constant flow of upgrades to PCIS_Seq, I paused to consider the simulator problem and inspiration hit like a totally painless thunderbolt! PCIS_Seq could be turned inside-out one last time, and turned into a PCIS Simulator, with a hugely reduced effort compared to what I had been anticipating. To quote Douglas Adams, “This time it was right. It would work. And no-one would have to get nailed to anything!”.

I had the first working version of PCIS_Sim running successfully before the sun came up.

The development and the current state of the PCIS Simulator software, and the PCIS Simulator hardware unit that was developed to support it, are described in Section 5.7.3.
5.7.2 The PCIS Run-Time Software Developer’s Edition – PCIS_Dev

The Developer’s Edition of the PCIS Run-time software, PCIS_Dev, removes the constraint that user actions (other than clicking the “Pause” button) should not be able to pause or halt the execution of the Sequence. (Thus it retains its title bar when the Run starts, and the window may – with care – be moved during the Run.)

The two main differences between PCIS_Seq and PCIS_Dev are that:

1. the “Debug” group of instructions are enabled in PCIS_Dev. Once a “#Debug On” statement is encountered, all subsequent #BreakPoint and #Dump statements are enabled. All “Debug” commands are disabled in PCIS_Seq, so that normal execution of the Sequence is not interrupted; and
2. the User Pause window (containing only “Continue” and “Abort Run” options in PCIS_Seq) is considerably enhanced, offering additional options including “Single Step” and Keyboard Command Entry.

The Developer’s edition also defaults to the display mode showing the Input and Serial logs (rather than the RunInfo file) when the Run is started, on the assumption that the lower level information would be wanted during development, but probably not when a tested Sequence is run on the PCIS.

The Developer’s Edition is clearly identified by the prominently displayed words:

PCIS Control Centre - v3.5 *** Developer’s Edition ***

These features can be seen in Figure 5.7.1, which shows PCIS_Dev performing a simulated OSL Run.

![Figure 5.7.1 PCIS_Dev User Interface, showing the enhanced functions of the Pause window.](image-url)
Development and Application of a Photon Counting Imaging System

Figure 5.7.1 demonstrates the #Dump Debug statement, which has just caused the PCIS variable values to be written to the Serial Log, and hence also to be shown in the lower display window. The command which caused this can be seen at the bottom of the upper (Input Log) window:

```
#DUMP 1 - At End of RECORD_IMAGE.PSI
```

This Debug statement would be ignored by PCIS_Seq, and the Serial Log dump would not happen.

The “1” in the #Dump command ensures that the dump only occurs once, not every time the RECORD_IMAGE Include file is executed, so that the Serial Log does not fill up with unwanted information. The rest of the line is just text that will be recorded in both the Input and Serial Logs, identifying the point at which the dump occurred.

The Pause window in Figure 5.7.1 shows the other special features of PCIS_Dev. Of the six buttons shown, only two – “Continue” and “Abort Run” – appear in the Pause window of PCIS_Seq. The four buttons exclusive to PCIS_Dev will be explained here.

“Restart File” is simply a re-instatement of the original third button when this window was first implemented, for private personal use as a temporary testing utility. However, it has a real purpose, which is why it was re-instated.

When “Restart File” is executed, PCIS_Seq (or PCIS_Dev) actually closes the file, and then re-opens it for Input. This means that any changes that have been made to the file since it was opened the first time, will be there in the file opened the second time. This means that if, during Sequence testing, it is apparent that a line of code was omitted or incorrect, it may be possible to Pause the Run, edit the offending source code file using a text editor, and then click “Restart File”. It is a matter for the user’s judgement whether this will lead to the same end result as executing the correct file, once.

“Swap Display” functions exactly as the same button on the main window, but the main window (and its buttons) are inaccessible when the Pause window is active (“has focus” in the jargon), so a copy was placed on the Pause window as well.

The “Single Step” function allows the Sequence source code to be executed one line at a time, with the Pause window being re-displayed each time. It was surprisingly simple to implement, and is extremely useful in tracking down exactly where a Sequence error occurs. Interestingly, I then realised that I could implement a “#BREAKPOINT” statement, pausing Sequence execution and popping up the Pause window allowing single-stepping from that point, with a single line of VB6 source code (“If DebugEnabled Then Pause = True”). So I did.

The final extra button, “Command Entry”, is extremely powerful. It causes the text entry window (visible in Figure 5.7.1 above) to appear as part of the Pause window. Using this, the user can enter any valid Sequence language command, and PCIS_Dev will execute it as though it had appeared in the current source code file. This can be used for “real” commands, like setting a parameter you had forgotten, so that the simulated run can continue; or for Debug commands like #Dump, to assist in the Sequence code debugging process. (Note that the complete Input and Serial Logs can be seen at any time by opening the Log files written to the hard disk. These will be fully up-to-date.)

The capacity with PCIS_Dev to destroy a Run – and hence the need to split the Sequence software into the two versions described – should by now be obvious.
5.7.3 The PCIS Simulation System

The PCIS Simulation System consists of two components; the Simulator software (based on PCIS_Seq); and the PCIS Simulator hardware. The next section describes the operation and capabilities of the Simulator software, PCIS_Sim.

The Simulator hardware was developed to enable the complete Sequence Software Development Environment to be run on a single, normal, modern laptop or desktop computer, and is described in the following section.

5.7.3.1 The PCIS Simulator Software – PCIS_Sim

The basic idea of PCIS_Sim was amazingly simple. I realised that PCIS_Seq already had the facilities required to support a PCIS Simulator – support for reading and writing the serial port, detailed multi-level run logging capabilities, and a functional and effective user interface. All I had to do (for the first version) was implement a single Sequence Command called “#PCIS_Simulator_Mode”; create a “Sim_Config” Sequence file containing that single command; and then run PCIS_Dev and PCIS_Sim on two separate computers connected by a serial cable. (These were initially in separate rooms.)

The #PCIS_Simulator_Mode Sequence Command would sit in an infinite loop, patiently waiting for a line of input to arrive on the serial port; examining that input (using a “Case” or “Switch” statement based on the first text string in the line, assumed to be a Minisys command); if necessary, reporting an Error or Warning to the user interface (via RunInfo messages, and later also by changing the screen colour); and finally sending a response that “looked like” what the Minisys would normally send. Then back to patiently waiting for another line of input.

PCIS_Sim does not stop if a Warning or Error occurs. It just reports it, and waits for the next line to arrive. In the case of an Error, nothing is sent back to the Runtime Sequence software running on the other machine. So PCIS_Seq (or PCIS_Dev) may get an Error and Abort the Run, if a response was expected to the Command that caused the PCIS_Sim Error; but PCIS_Sim keeps on running until the operator clicks on the Exit button.

The first version of PCIS_Sim was working very quickly. Improvements were then gradually made, to the user interface; to the log file recording; and to the range of Minisys commands recognised, and the “sophistication” of the responses sent to them, leading finally (and with pleasingly small total development effort) to the highly functional current version of PCIS_Sim, which is described below.

[Note that PCIS_Seq, PCIS_Dev and PCIS_Sim are all produced from the same VB6 source code. VB6 compiler directives are used to control “conditional compilation” of various code segments to produce the different executable programs. The whole process is controlled by setting a single compiler variable – “#Developers_Edition” or “#Simulator_Version” – to “True” before compiling the code (and selecting the correct executable file name). Thus there is little, if any, extra code maintenance workload as a result of having the three versions of the software.]
The obvious changes that can be seen in the PCIS Simulator user interface (shown below at Figure 5.7.2) are the removal of any ability to specify an input Sequence file or location (the special Simulator Config Sequence file in the PCIS Program directory is used automatically); and the changes to the labels used in the Log Files (and the display windows), which were made to improve clarity and readability. (This last change, to the already grossly over-loaded PCIS_Seq logging routines, were more messy and difficult than I had expected – or than would have been the case with well-designed source code!)

The PCIS Simulator starts, as does PCIS_Dev, with the low level log files displayed. The "Swap Display" option is available, but the RunInfo file of the Simulator run actually has nothing of interest in it (just the PCIS Run Start messages, and a Simulator Mode Start-up message). Mind you, very little more than that appears in the Input Log either – the action is all in the Serial Log window.

![PCIS Simulator V1.0](image.png)

Figure 5.7.2 PCIS Simulator User Interface – Run progressing normally.

The improvements made to the “quality” of the simulation – i.e. more complex but more realistic responses to the various Minisys commands – were a bit more involved. The complete Minisys Command Set document was examined (see Appendix N), and all possible commands were assigned to one of several categories. These, and the type of Simulator response elicited by each, are examined below.
1. Unrecognised commands – not listed in the Minisys Command Set documentation (and so usually a typo in the Sequence code). These will all cause a Simulator Error.

2. Commands that should never occur in the PCIS – principally changing the serial port parameters; and Monochromator commands. (The Monochromator Minisys option is physically as well as logically incompatible with the PCIS, so the commands should not occur.) These too cause an immediate Simulator Error.

3. Legal Command names that contain a syntax error (too many or too few parameters), or an illegal parameter value. These too elicit a Simulator Error and no other response.

4. Legal commands that do not require a response – e.g. “Cancel Beta Irradiation” (BC), or “Position Sample n at Heater” (PS n).

5. Legal commands that require one or more standard, fixed responses – e.g. “Read Software Version” (RV) or “Read System Parameters i to j” (RA i/j). PCIS_Sim sends fixed, pre-coded (but appropriate) strings to the serial port, then returns to waiting for the next input line.

6. Commands that require an “intelligent” response from the Simulator, which PCIS_Sim has to calculate in some fashion before sending. These are mainly the Minisys commands to do with issuing IICU commands, and with polling Minisys and IICU commands for completion. These have a more sophisticated level of support in PCIS_Seq, giving the Simulator system the ability to detect and report a much greater range of possible Sequence errors.

Commands in groups 4 through 6 above are further divided into those that require polling for Minisys Command completion (those that take time, such as sample positioning or handling commands); and those that do not require polling (because their operation is instantaneous). If the command used to poll the Minisys for command completion (“Read Status Byte 3” (RS 3)) is received after a Minisys command that doesn't require polling, then a Simulator Warning is generated. If a command that does require polling for completion is followed by a Minisys command other than Read Status Byte 3, then a Simulator Error is generated.

The PCIS Simulator also tracks all “Operate Relay” (OR) Minisys commands, and determines when an IICU Command has been issued. The Command Number is decoded, and the Command Table conventions are then used to recognise the following IICU commands:

- Command 63 – Switch to Primary Table;
- Command 62 – Switch to Secondary Table; and
- Commands 0 or 1 – assumed to be Hard and Soft Reset, respectively.

These are used to keep track of which Command Table (Primary or Secondary) is in use, and to control the Simulator's expectations for IICU Command Completion polling. (Polling of the IICU for command completion is performed using the Minisys “Read Optical Input 1” (RO 1) command.)

None of these special recognised IICU commands require polling of the IICU for their completion. All other Primary Table commands require IICU completion polling before another IICU Command is issued; Secondary Table commands (assumed to be instantaneously-executed “Set Defaults” commands) do not require command completion polling.

If an IICU Command which does not require completion polling is followed by an “RO 1” Minisys command (polling the IICU for completion), then a Simulator Warning is generated. If an IICU command that requires polling is not followed by “RO 1”, then a Simulator Error is generated.
When polling for either Minisys or IICU Command completion is required, the relevant Minisys command ("RS 3" or "RO 1") must be issued several times (the current setting is 3 times) before a "Command complete" response is issued. ("Command still running" responses are issued to the first two polls.) This simulates more realistically the timing of real PCIS operations, without unduly slowing down the Simulated PCIS Run.

The other Minisys Command given special treatment is "RT 1" – Read Sample Temperature. To support realistic testing of the "Wait_Sample_Cool" Include file code, the Simulator returns a geometrically-descending set of Temperature readings in response to "RT 1" commands, until they fall below 25 degrees, and are set to a high value again to start a new descending sequence. This means that a subset of the Wait_Sample_Cool calls in a Sequence will be simulated with a proper sequence of calls and (descending) responses. Others will immediately receive a response below the Sample Cool Temperature specified in the Sequence, and only one "RT 1" command will be issued.

PCIS_Sim also performs extensive edit checks on the parameters specified with many of the Minisys commands that are usually used by the PCIS, especially the TL, OSL and TCSL commands. Values of these parameters (either individually or in combination) may cause a Simulator Warning or Error.

Figure 5.7.3 below shows PCIS_Sim after a Simulator Warning has occurred. In this case the Simulator Warning was generated by issuing an “IComp” Sequence command (using the Command Entry function of the PCIS_Dev Pause window) when IICU completion polling was not expected.
It will be seen in Figure 5.7.3 that the Simulator Warning has caused the user interface to turn a sickly pink. This makes it very difficult for the operator to miss that the Warning has occurred.

When a Simulator Error occurs, the user interface turns an even more un-missable red. This can be seen in Figure 5.7.4 below, where a Simulator Error has been caused by issuing two Primary Table ICU commands, without polling for ICU completion in between.

When a Simulator Error or Warning is seen, the user can Pause the Sequence execution. The PCIS_Dev debug facilities may then be used, and the Log windows (or the actual PCIS_Dev and PCIS_Sim Log files) examined, to determine whether an error exists in the Sequence code.

![PCIS Simulator V1.0](image)

Figure 5.7.4 PCIS_Seq User Interface after a Simulator Error has been generated.

It may be seen in the above figures that a Simulator Warning or a Simulator Error also activates the "Reset Err(or)" button. Clicking on the Reset Err button will restore the interface to pale blue.

(PCIS_Sim is given a blue interface to make it easily and obviously distinguishable from either PCIS_Seq or PCIS_Dev, one or other of which will always be running at the same time as PCIS_Seq.)

It is important to understand that when a Simulator Error or Warning occurs, the only actions taken by PCIS_Sim (other than recording the event to the log files) is to change the screen colour, and to enable the "Reset Err(or)" button. The only function of the Reset Err button is to restore the screen colour to pale blue. This allows subsequent Warnings or Errors to be observed.
Recall that PCIS_Sim does not halt when a Warning or Error occurs. It simply goes back to waiting for the next instruction to arrive, and then deals with that as best it can. The run-time software that delivered the command may of course fail, especially if the intended command required a response from the Minisys, but PCIS_Sim itself will stop only when the Exit button is clicked.

The VB6 source code which implements the PCIS Simulator appears at Appendix M. It may be referred to in order to identify the exact PCIS_Sim behaviour associated with any given Minisys command, sequence of commands or combination of command parameters.

5.7.3.2 The PCIS Simulator Hardware

The Sequence Development Environment is meant to make the development and testing of PCIS Sequences a safe and comfortable operation that can be performed in the office, or at home – and not in a dark, dangerous lab. But few home or office computers have one serial port, let alone two. To serve its purpose, the SDK should be able to run anywhere, and on a normal laptop computer.

The solution was the USB-connected PCIS Simulator hardware unit Version 1.1 shown below at Figure 5.7.5. (Version 1.0 wasn’t in a box, and didn’t have a label.)

Figure 5.7.5 The PCIS Simulator Hardware Unit, allowing operation via a USB port.

This “Simulator Hardware” is actually a bit (though not a lot) of a con. Internally it comprises a USB-to-dual-RS-232 adaptor cable purchased from a well-known electronics supplier, with the USB plug left available outside the box. The two serial plugs are securely connected to each other, via an RS-232 serial cross-over (or “null modem”) cable (obtained from the same source). The adapter provides two logical Serial (“ComN”) ports, which may be used by the two applications (PCIS_Dev and PCIS_Sim). The entire SDK may thus be run on any computer with a spare USB port available.
**5.7.4 The Complete Sequence Development System**

The complete PCIS Sequence Software Development Environment is shown in the photograph at Figure 5.7.6 below. This shows it running on my "other" laptop (not the one I am currently writing on) with a second screen attached. The use of two screens is highly recommended – handling the required number of open windows on a single screen is extremely difficult.

PCIS_Dev is running on the left-hand screen, with the Pause window open (so that the Run is halted, and may be examined). The right hand screen (the laptop’s own screen) is displaying the running Simulator software. (No Error or Warning has occurred, so the Simulator is its usual pale blue.) Also running on the right hand screen (just to the right of the Simulator) is a window displaying the Input Log from the PCIS_Dev Run. A third window (visible just below the Simulator) is looking at the PCIS Outputs/RunName/Support directory where the Log files are being created. This allows the Input Log to be re-opened in the right hand window whenever the PCIS_Dev Run is paused, ensuring that the complete and up-to-date log file can be examined.

The PCIS Simulator hardware unit can be seen at the right of the picture, plugged into a USB port at the right side of the laptop computer.

![Figure 5.7.6](image-url) The complete PCIS Sequence Development Environment. The Simulator hardware is at the right.
6 PCIS Applications to Date

"I thought of images as shapes produced by artists. Roger thought of images as numbers produced by light."

"The Archimedes Codex" by Reviel Netz and William Noel.

This chapter provides a brief review of the various luminescence experiments that have been conducted to date with the PCIS. The emphasis is placed on the demonstration of the performance and versatility of the PCIS, rather than on the details of the experiments themselves, their goals and their results.

Many tests of the PCIS’ performance were, of course, conducted using whatever samples happened to be in the Minisy, applying an artificial beta dose, and then ensuring that the PCIS operated as intended, and that the results were broadly as expected. These tests are not described in this chapter. From time to time, however, when a major development stage of the PCIS was considered complete, my supervisor Dr Spooner would participate in the testing, and was able to propose (and manage the selection and loading of samples for) luminescence experiments that were of interest in their own right, either to him or to his research colleagues. Even in these cases, my attention was primarily focussed on the PCIS and its performance; the interpretation of the particular results remained the responsibility of Nigel or his colleagues. The attention I was able to pay to the actual experiments and their results was dependent on how well the PCIS was operating at the time, and how much operator attention and intervention it required. During most of these tests, the PCIS was far from the simple-to-operate equipment that it is today (see Appendix J), and running the equipment required a high degree of timing-critical manual operation.

Accordingly, this chapter does not attempt to interpret the results of the experiments in any great detail. The goal is to provide an overview of the nature of the luminescence experiments, and to show the performance of the PCIS in undertaking them. Descriptions of the scientific work undertaken, and of the results obtained, may in some cases be found in referenced publications based, in part, on these PCIS experiments.

The one exception to the above considerations is the examination of mobile phone SIM cards as ubiquitous opportunistic retrospective dosimeters, described below in Section 6.5. (An opportunistic dosimeter is a material that happens to be there, as opposed to one that is deliberately placed at the location ahead of time.) This was the final PCIS run that I conducted before I was banished to writing this thesis. The real purpose of this experiment was to prove to myself that the PCIS no longer required close attention during a run, and that I could in fact concentrate on the experiment design and sample preparation, rather than on the operation of the PCIS, and could therefore now undertake genuine and interesting scientific investigations with complete confidence in the performance of the equipment. In this it was a complete success, although, as we shall see, the scientific results for the particular TL tests conducted were negative.
6.1 Phase 2 PCIS Performance Testing

Testing of the Phase 1 PCIS had already demonstrated that the optical performance and sensitivity required of the system had been achieved. The Phase 2 development was primarily about programming of sequences and the full automation of PCIS runs. As such, all experiments successfully concluded with the Phase 2 PCIS were tests, and proofs, of the Phase 2 performance, and there were no major new runs conducted purely in the context of proving the PCIS automation facilities.

The PCIS automation facilities performed nearly flawlessly through these tests. Occasional bugs in newly-developed code and facilities were universally quick and easy to locate and rectify using the Sequence development environment described in Chapter 5 (in conjunction with the Visual Studio development environment for the VB6 source code).

Following completion of the Sequence development environment, and my acceptance that the PCIS was now in its final deliverable form, only a single run (the SIM card tests described in Section 6.5) was conducted prior to abandoning the PCIS in order to write this thesis. That test was a complete success in terms of PCIS operation, allowing a reasonably complex experimental protocol to be developed and tested “off-line”, and to operate correctly at the first run on a real sample. I was able to focus my attention on the samples and the experiment, rather than on the experimental equipment, and in this one case I am therefore competent to discuss the results of the experiment and their implications for further investigations.

The automation facilities which comprise the Phase 2 development of the PCIS are proved to be a complete success, and no further runs aimed at testing the automation facilities are required.
6.2 Optical Glass Studies

The optical glass studies were conducted for a colleague of Dr Spooner, Christopher Kalnins from the University of Adelaide. Christopher is working on the development of direct-sensing fibres for detecting ionising radiation (as opposed to the “traditional” approach of attaching a scintillator to the end of an optical fibre, with the loss of signal and therefore sensitivity implied by that approach). The proposed detection system would also have the advantage of allowing radiation to be detected anywhere along the length of the fibre.

The samples comprised standard, commercially available Fluoride Phosphate (FP) and borosilicate glasses, along with a number of specially prepared FP glasses doped in varying concentrations with Manganese, Chromium, Europium, Copper, Terbium and Cerium, plus some undoped FP glasses with different conditions of manufacture. The various dopings were designed to control the wavelength of the luminescence produced in the glass, to meet various experimental and detection criteria.

The importance to this work of the PCIS lay primarily in the ability to detect long wavelength emissions – Christopher was particularly interested in red emissions, which traditional PM tube-based systems are unable to detect. The unique ability of the PCIS to directly show how the emission centres were distributed within the sample was also, of course, of great interest.

These experiments were conducted using an early release of the IICU, and my primary goal was the testing and debugging of a wide range of newly-implemented features. In spite of the debugging of the IICU facilities consuming two of the three days available for the experiments, the third day produced successful, bug-free runs, and useful and interesting results from the glasses tested.

In particular, the manganese-doped Fluoride Phosphate glass produced very strong emissions in the red. To quote from Christopher’s communication to me describing the results:

“The TL intensity from this sample was orders of magnitude higher than for other samples, such as the chromium doped fluoride phosphate sample which was also expected to emit in the red. The PCIS let us conclude that doping the glass with manganese is worth considering if we want to produce luminescence in the red.”

Figure 6.2.1 below shows the 100 °C to 200 °C TL (at 2 degrees per second), with no optical filters and no CCD binning, from the Mn-doped FP glass after approximately 10 Gy applied beta dose. Notice the magnitude of the light sums, with the 95th percentile at over 1,000 counts per pixel, for a total of about $3 \times 10^7$ counts for the entire sample. Such large numbers leave no doubt as to the statistical significance of the results obtained. The obvious ring of light results from luminescence photons reflecting from the edges of the planchette in which this relatively thick sample is carried.

During the TL runs on the glass samples, a disk of Aluminium Oxide (“Sample 13 – Al2O3:C dosimetry crystal”) was included. The TL for the Aluminium Oxide, with the same treatment and following the same beta dose as the glass samples, is shown at Figure 6.2.2. (Note that the TL peak widely used for dosimetry lies well above the temperature range used here.) In this case the 95th percentile counts are about 2,670, with a total of close to $10^8$ counts for the entire sample. The slight inhomogeneity seen in the signal is probably caused by slightly different heating across the sample due to poor thermal contact from the heater plate, to the sample planchette, to the sample itself.
Figure 6.2.1 TL from Manganese-doped FP glass, 100 °C to 200 °C after 10 Gy beta dose.

Figure 6.2.2 TL from an Al₂O₃ dosimeter chip, 100 °C to 200 °C, after 10 Gy beta dose.
6.3 Analysis of Porcelain Slice

In the run-up to preparing my paper for Radiation Measurements disclosing the development and existence of the PCIS (McCulloch et al, 2011), measurements were made of the luminescence behaviour of a porcelain sample, and of various samples of common salt (NaCl). These results were included in two papers delivered at the same conference, and also published in Radiation Measurements. These two applications are described in this and the following section.

The study of a porcelain sample was performed by Dr Spooner and myself on behalf of Hugo Oks of the Institute for Photonics and Advanced Sensing at the University of Adelaide (Oks et al 2011). The goals of the study to which the work with the PCIS contributed were twofold:

1. to determine the spatial uniformity of the emissions from porcelain; and
2. to determine whether significant emissions existed at wavelengths too long for detection with normal PM tube-based equipment.

The porcelain sample used for the PCIS study, called HOPORA, comprises a 0.35 mm thick slice, approximately 7 mm square, from a telegraph post insulator. The insulator was manufactured in South Australia in around 1980. This is one of 5 samples, including three pre-Chernobyl samples from Russia, Belarus and the Ukraine. An image of this material undergoing TL measurement to 160 °C appeared earlier as Figure 3.2.11, and is reproduced below as Figure 6.3.1.

![Figure 6.3.1 Porcelain sample HOPORA, TL measured to 160 °C at 2 K/s, after 1 s (~0.1 Gy) beta dose.](image-url)
The TL image shown at Figure 6.3.1 was acquired without any optical filters installed in the PCIS, and therefore represents all the detectable emission wavelengths from the sample. Analysis of the image reveals that the visible “bright spots” have a brightness about 3 times that of the rest of the slice. Oks et al (2011) concludes that the images are consistent with the hypothesis that the porcelain luminescence originates from inclusions of quartz in the porcelain material. He postulates that the thicker slice used (0.35 mm as against the 0.2 mm used in previous studies) accounts for the relatively low contrast between the bright spots and the background, with photons from deeper quartz inclusions being diffused through the bulk of the porcelain sample. There are also in Figure 6.3.1 a few noticeable dark spots. In my own view (Oks et al (2011) expresses no opinion on this), these dark spots probably indicate the presence of opaque mineral inclusions at or near to the top surface of the sample, blocking any photons from below.

Oks et al (2011) does not comment on the overall grading of photon counts from a peak at the lower right of the sample (as seen in Figure 6.3.1) to a minimum at the upper left of the sample. This may be due to uneven contact between the porcelain slice and the sample carrier, leading to better thermal conduction (and faster heating) at one end of the sample. If this is the case, then the uneven contact is probably due to the lower surface of the porcelain slice not being perfectly flat. The porcelain sample was sliced using a water-cooled Buehler diamond wafering saw, but it is likely that a small burr was not subsequently polished away and has caused the uneven thermal contact.

On the second objective – that of determining the existence of any long wavelength emissions – the results were less conclusive. Tests conducted with both the PCIS in Canberra and with the Fourier Transform Spectrometer in Adelaide did not reveal any previously-unknown emissions, confirming the known emission bands at 390 nm, 420 nm and 620 nm. What was surprising was the intensity of the 620 nm emission, and subsequent work concentrated on the 230 °C, 620 nm emissions. While this emission band is certainly detectable with “traditional” luminescence readers, it is outside the primary sensitivity range of the PM tubes normally used in luminescence work, and Oks et al (2011) comments “that this red emission is usually removed by blue-pass filters in conventional TL readers”.

The extreme spectral range and very flat spectral response of the PCIS, together with its ability to perform low to moderate temperature TL without any optical filters installed, makes it ideal for the recognition and assessment of these longer wavelength signals. Whether or not the PCIS is the ideal system for the later detailed analysis of these signals, it certainly would appear to have a valuable role in the initial determination of the existence and spatial origin of luminescence signals, and particularly of signals of longer wavelengths. Spatial analysis of the origins of the different wavelength signals is then a simple operation with the PCIS.

The stated objective of these porcelain studies was to investigate further the potential of porcelain – and particularly of porcelain insulators, which are virtually ubiquitous in urban areas around the world – for use as an opportunistic retrospective dosimeter. Opportunistic dosimeters have huge potential for the analysis of radiological events that cannot be predicted, and so cannot be monitored by pre-placed dosimeters or dosimetry equipment. Such events could include the accidental or illicit storage and transport of nuclear material, or the detonation of a “dirty bomb”.

Overall, the PCIS was able to make a significant contribution to the porcelain study, and in particular provided an immediate visual insight into the spatial distribution of the luminescence centres.
6.4 Table Salt (NaCl) Characterisation Studies

A number of PCIS runs were conducted with common table salt (NaCl, hereafter just called “salt”). While this was primarily due to the ongoing interest of my supervisor, Dr Spooner, in the analysis of luminescence from salt, the material also offered me several advantages while developing the PCIS.

The first of these advantages is being able to walk into a supermarket and purchase, for a few dollars, a wide variety of samples of different production methods and geographic origins. While I am still at a total loss as to why anyone not undertaking a detailed mineralogical analysis could possibly need 10 or more varieties of table salt to be available (my local urban supermarket in Canberra even sells Pink Himalayan Rock Salt!), I will take a lucky break when I get one.

Secondly, as we are not trying to work out the age of the salt, all of the analysis would be carried out using applied beta doses. This allowed all of the sample preparation – grinding, sieving etc. – to be carried out in full light. Sample preparation was in fact mainly done in my kitchen, with the unwanted size fractions simply being put on my dinner.

Thirdly, it transpires that salt, in addition to the well-known 590 nm (orange) emission, also has emissions in the blue and UV, and is susceptible to a range of thermal, IR and visible light stimulations. These factors make salt an ideal material for PCIS studies; and at the same time make the PCIS, with its broad and very flat spectral response, and its ability to apply thermal, IR, red and blue stimulation, an excellent tool for studying salt. (The PCIS also contains UV LEDs, but these are not generally regarded as a stimulation source, their main role being to allow focussing images to be acquired with optical filter packs that exclude the blue and longer wavelength photons.)

The overall goal of Dr Spooner’s work with salt (a ubiquitous material wherever people live, work and, primarily, eat) is to investigate its potential as an opportunistic retrospective dosimeter with application to radiological event analysis. A primary goal of the work described here, and in Spooner et al (2011), was to investigate the similarities and the differences between salt samples of widely differing geographic origins and production or extraction methods – hence the wide variety of salt types purchased and tested.

The contribution of the PCIS to this work was primarily to be an assessment of the homogeneity or heterogeneity of the various emission signals across separate grains of a given salt type, for a wide variety of salt types. This objective the PCIS was easily able to meet.

In summary, the inter-grain heterogeneity of emitted signals does vary widely with the origin and type of the salt. Spooner et al (2011) concentrates mainly on Sample 3, an evaporated sea-water salt, and includes TL images of this sample acquired with the PCIS. These images clearly show a high degree of inter-grain homogeneity of luminescence response for both the JV emission (using blue optical stimulation and 3 mm of U 340 filter), and for the sum of emissions in the orange to blue region (using IRSL with 3 mm BG 39 plus 3 mm GG 445 filter material). By contrast, the large grains of Australian Lake Salt display a much higher degree of inter-grain heterogeneity. (These grains were shown earlier at Figure 3.2.10, undergoing pure TL from 160 °C to 260 °C after a 20 Gy applied beta dose, with no optical filters installed, and therefore representing the sum of all detectable signals.)
These differences in homogeneity of response are perhaps unsurprising given the different origins of the samples. A recently-evaporated sea-water salt might be expected to show greater inter-grain homogeneity than a salt of “geological” origins, such as a European mined salt, or our Australian Lake Salt samples. Even though the Lake Salt is another evaporative salt, I know from my own observations that the conditions vary greatly across these lake beds as they dry out following a rain event. By contrast, the commercial sea-water salts are artificially dried under conditions of relatively great uniformity, and this is perhaps reflected in the homogeneity of their luminescence responses.

Attempts have also been made, using both the data from the PCIS and that from the Fourier Transform Spectrometer, to relate the relative intensities of the orange, blue and UV emissions to the source and nature of the salt crystals. To the best of my knowledge no clear relationship has yet been identified, and this area is still under study by Peter Hunter at the Defence Science Technology Organisation (DSTO).

One further observation emerged — essentially serendipitously — from the PCIS studies of salt, and warrants mentioning here.

During and immediately following the PCIS salt runs described in Spooner et al (2011) I was making final changes to the PCIS Sequence Language, and was developing the TL_Probe Sequence Template from the individual TL Sequences that had previously been used. These developments required numerous short runs of the PCIS to test and prove each small change, or set of small changes, that were made to the software. Basically out of laziness I left the same salt samples in the PCIS, and kept repeating variants of the TL analysis on the same 1 or 2 disks of Australian Lake Salt.

Although I was again more interested in the performance of the equipment than of the samples, at some point I realised that the signals were disappearing! With the same repeated beta dose (usually a 200 second, or approximately 20 Gy dose), less and less luminescence was detected each time the machine was run. The loss of sensitivity was repeatable, and significant.

Up until that time each TL test on salt had been performed with a sequence of “beta, TL, TL”, with the second TL being to confirm that all of the available signal had been extracted by the first TL. (A similar approach had also been adopted with the IRSL and visible light OSL experiments.) Following the above observation, the Salt TL Sequence was changed (now a simple operation!) to implement a sequence of “beta, TL, beta, TL, TL”, with the second set of “beta, TL, TL” being intended to show any sensitivity change between the two runs.

The data providentially gathered during PCIS development has not yet been analysed to quantify this effect; nor have the tests been repeated with different types of salt. Quantification of the effect for the different emission wavelengths for each salt type would also be an interesting study, and should add to our understanding of the dynamics of luminescence in salt, and possibly in other materials.

This study, across both emission types and salt types, would require a significant experimental workload. However, this is precisely the type of work at which the PCIS, now easily programmable, excels, and could well provide an interesting and useful PCIS application when this thesis is (finally) finished, and I am allowed to go back into the lab and play with my shiny black toy.
6.5 Opportunistic Retrospective Dosimetry with SIM Cards

Given the high profile of “radiological event analysis” applications, and the potential value of personal retrospective dosimetry, I was pondering the materials associated with an individual that might be able to record a measurable signal when subjected to ionising radiation. This limits us to something crystalline or semi-crystalline that all (or most) people carry on their person all (or most) of the time.

I considered spectacles, but there were a number of disadvantages:

- not enough people wear or carry them;
- most eye-glasses are not glass, but acrylic — according to my optometrist friend, the vast majority are CR39 (Columbia resin number 39); a few are polycarbonate; and less than one percent are glass (Horst Reiss, pers. comm.)
- even glass spectacles are unlikely to be made from one of the “good” glasses for OSL, experience having shown that it is the highly doped glasses that give the best OSL signals;
- by their very nature, eye-glasses tend to be frequently exposed to light, which would bleach any stored signal before the glasses could be recovered for analysis; and
- people tend to need their eye-glasses, and the luminescence testing (certainly in existing equipment) would necessarily be destructive.

Then I realised that the personal item that is now ubiquitous, certainly in the western world and to an increasing degree in the developing countries, is the mobile phone. And buried inside all mobile phones, away from any bleaching light exposure, is a small, cheap crystalline item — the SIM card. Ninety percent or more of people carry a mobile phone almost everywhere they go; SIM cards are cheap (I have purchased them, retail, for 7 cents each); and (with the co-operation of the service provider) a SIM card can be quickly and easily copied, so that the individual may continue to use their phone. I decided to test the potential use of SIM cards as personal retrospective dosimeters.

The second motivating factor was that the PCIS automation systems were now complete, and it was now, in theory, so easy to run the PCIS that I could concentrate on the application, the design of the protocol and the selection and preparation of the samples, rather than focussing my entire attention on ensuring the correct operation of the equipment. That the experiment might also produce valuable scientific results was a bonus. What I really wanted to prove, at least to myself, was that the PCIS was now a usable and reliable scientific instrument that could be easily applied to a real science question, and would produce meaningful results quickly and easily. This I achieved.

The sequence used was based on the TL_Probe sequence template. It applies a beta dose of approximately 30Gy to the sample, then, with no optical filters installed (so that the full spectrum of emitted signals might be detected), TL runs to 50, 100, 150, 200, 250 and 300 °C were performed. The protocol was tested under the Sequence Development Environment described in Chapter 5, and was found to be working correctly.

As is supposed to now be the case, preparation of the sample was the more difficult and time-consuming task. The construction of a SIM card is quite interesting. The actual chip is glued to a piece of tape which has the conductors (as seen on the surface of the final SIM card) connected
through to the opposite surface. The chip is attached to the tape with an opaque black epoxy cement. The tape is then cut, and the piece (with the chip attached) is inverted and glued into a well cast into the plastic card.

Removal of the tape from the plastic card is a simple operation using a scalpel. Removal of the epoxy (and the chip) from the tape is less trivial. Commercially, this operation is performed with high pressure jets of high temperature, high molar Nitric acid. This was not a procedure I relished the idea of replicating at home. The advice of our “rock lab” manager, Shane Paxton, was to soak the chip and epoxy (at room temperature) in a bath of a nitric acid-based brick cleaning product available from our local hardware store. This I did; the epoxy softened, and the chip was easily removed and cleaned ready for analysis. Note that I conducted this procedure in good light (intending to artificially dose the chip before testing for luminescence), but it could easily and safely be carried out in dark room conditions (dim red lighting).

The SIM card has a coated side (appearing under an optical binocular microscope to be vapour-deposited gold) and an un-coated side. Since I had broken the chip while working out how to remove it from the epoxy, I placed the two parts on the sample disk with opposite surfaces showing. A focussing image of the SIM chip pieces on the sample disk appears below as Figure 6.5.1.

![Focussing image (using red LEDs) of the SIM chip fragments.](image-url)

Figure 6.5.1 Focussing image (using red LEDs) of the SIM chip fragments.
Unfortunately, I was unable to coax a luminescence signal from the prepared SIM card. But, in science, a negative result is still a result, and the PCIS had produced it with a minimum of fuss and bother. In terms of proving the value of the PCIS as an investigative tool for exploring luminescence phenomena, the experiment was a total success.

On pondering the reasons for this lack of success with a material that is, in essence, pure silicon, I hypothesized that the reason may simply be that the material in the chip is too pure. In the manufacture of silicon wafers for the production of integrated circuits, enormous lengths are gone to in purifying the material before it is then doped to form a P or N semiconductor substrate. Since luminescence signal storage and recombination sites are believed to be associated with impurities and dislocations in the crystalline material, it seems reasonable to suppose that the lack of signal from a SIM chip may be due to a lack of the impurities and dislocations required. I am at present uncertain as to how this hypothesis might be tested.

The experiment does not totally rule out the use of SIM cards for retrospective dosimetry. Further tests using IRSL and both red- and blue-stimulated OSL, with the chip held at a variety of temperatures, should still be conducted, as should higher-temperature TL runs employing optical filters to exclude thermal photons.

23 One method employed to "dope" the substrate is by irradiating the cylinder of pure silicon with neutrons, thereby converting some of the Silicon to Phosphorus (IAEA-TECDOC-1681, 2012). I cannot resist the observation that perhaps the chemists owe an apology to the alchemists for their scathing condemnation of the search for transmutation of elements! This is now everyday engineering in the semiconductor industry.
7 Future Developments and Applications

"Don’t make fun of Grad students! They just made a terrible life choice."

Marge Simpson to Bart Simpson, in "The Simpsons”.

The goal of the PCIS project was to design, build and demonstrate an advanced item of luminescence research equipment that would transcend the limitations of existing luminescence readers. The PCIS would combine, in one device, high sensitivity, extreme spectral range and fine spatial resolution, with precise control of the sample environment and emission stimulation methods, and convenient and flexible optical filtering for the selection of emitted signals. My personal further goal was that the PCIS must be completely reliable, and as simple as possible to program and operate without limiting its functions and capabilities. These goals have now been achieved.

But very few projects of the magnitude of the PCIS development are ever truly complete. All milestone definitions, including that called “completion”, are arbitrary, and there will always remain things that should have been or could be done, or done differently.

The point at which I declared the development of the PCIS “complete”, and of necessity put it aside so that I could document the system in the form of this thesis and its appendices, was when the Sequence Development Environment (the PCIS Simulator and the Developer’s Edition of PCIS_Seq) enabled the safe, secure testing of PCIS Sequences prior to their execution on real samples. This, I believe, finally made the PCIS a useful and usable research tool, applicable to a broad range of luminescence investigations. But it does not mean that no further development is possible.

And it goes without saying that there are many applications to which I would have liked to put the PCIS, but which the time constraints on a PhD simply did not allow.

This chapter addresses some of the developments which I hope will occur in the future to enhance the already significant capabilities of the PCIS, and to improve its ease-of-use for an ever wider range of luminescence experiments; and a few of the most interesting and productive of the applications which the PCIS uniquely could undertake.

That there is so much remaining to be done to explore, and then to improve the capabilities of the PCIS could be seen as a condemnation of what has been achieved, suggesting that the PCIS as it stands is somehow deficient. I do not believe that to be the case. Rather, I see it as a sign of a successful development that it is not limited to one particular set of goals, but instead has the capacity, flexibility and fundamental sturdiness not only to form the basis for new and more extensive capabilities and applications, but by the very nature and quality of its present operations to actually suggest the directions in which further development could productively occur.

I hope to see the PCIS undertaking important investigations of luminescence and related phenomena for many years to come. A few of the developments and applications that may help this come about are described as briefly as possible in the sections that follow.
7.1 PCIS Hardware developments

The opto-mechanical components of the PCIS actually seem to be very well matched. There is no component which is clearly limiting the potential performance of other components and so limiting the sensitivity, signal-to-noise, spatial resolution or spectral performance of the complete PCIS.

That said, there are a few minor items, and two expensive items, that would improve the operator experience in running the PCIS.

The minor items include the provision of a better cover for the source turret (and perhaps also for the shutter housing) when the PCIS is not assembled; a more useful set of containers for the customized aperture plates, filter carriers and other inserts for the two optical filter drawers, which would simplify the selection of inserts and the setting-up of filter drawers in the dark; and a lighter and more sightly base/stand for the optics assembly. But these are all minor items, and the present solutions, although inelegant, suffice.

There are in fact only two items in the opto-mechanical category which I would have liked to implement, but could not, due their significant cost and my insignificant budget. Although they would not make the PCIS in any way more functional, they would make its operation significantly easier. They are described below and will, I hope, be implemented in the fullness of time.

7.1.1 Lens Lifting Rig

The arrangement of ropes, pulleys and turnbuckles that is used to raise and lower the optics was an excellent solution to a difficult problem given that the solution had to be implemented via petty cash payments of no more than $50 each. It is not a solution that would be contemplated were a budget appropriate to the requirements available. It is a method that is appropriate for the very high concrete ceiling (and lack of other substantial mechanical support) in the PCIS’ present location, and to a single user – me – who is very familiar with rope and pulley systems. I use such systems extensively for life support in my caving and diving activities, and am therefore confident to commit the integrity of the optics assembly to their – and my – care.

Other users may lack the same confidence, or indeed competence.

At present the PCIS must be disassembled (i.e. the optics assembly removed and replaced) whenever the samples are changed, or whenever the LN-CCD camera needs to be re-filled with liquid Nitrogen (presently after about 6 to 8 hours of PCIS run time). There is a plan to have a Minisys lid made with a removable vacuum-proof port via which samples may (with extreme care) be changed without the need to open the Minisys, and therefore without the need to raise the optics assembly. Similarly, I have recommended (see Section 7.1.2 below) the provision of a constant feed LN system for the LN-CCD camera. These two developments together would remove the need for regular removal and replacement of the optics assembly; nevertheless there would remain the occasional need to do so (if only when samples were dropped during replacement via the new port), and a method requiring less care, expertise and risk is definitely required.
When the PCIS is installed in its future location in the Institute for Photonics and Advanced Sensing at the University of Adelaide, a firm floor base will be provided, on which an electrically or hydraulically operated lifting mechanism will need to be mounted. While this will probably be fairly expensive, there is no real technical difficulty in placing the approximately 40 kg weight of the optics assembly onto the PCIS with the required accuracy and gentleness.

I foresaw this need right from the start of the project, and the lens box (constructed from 12 mm aluminium alloy plates) has eight through-threaded holes, arranged in an upper group and a lower group of four holes each, in the back panel. Thus no new machining of the lens assembly (which would require its complete disassembly) is required.

The placement of the eight mounting holes has been designed to allow the weight of the assembly to be cantilevered from a rear support without distorting the lens box, and therefore without affecting the alignment of the mirrors which is, as was seen in Chapter 2, incredibly precise. The holes are presently blocked with short bolts and washers to prevent light entry.

### 7.1.2 Constant Liquid Nitrogen Feed

The LN-CCD camera has its own containing dewar which is filled with liquid Nitrogen to keep the detector chip (and associated very-low-voltage circuitry) at the depressed temperature that allows low noise (and hence high sensitivity) operation. The desired temperature is considerably warmer than LN, so when the camera is operating an electric heater is used to "re-warm" the components. (The temperature is held stable at -110 °C +/- 0.05 °C.)

The combination of the fairly small (approximately 1.5 litre) dewar, and the fairly fierce heating needed to maintain the temperature accurately, means that the camera can only operate for about six to eight hours before it requires re-filling. This currently limits the longest PCIS run that can be undertaken. (There is presently also a limit in the WinView software of 1,000 images in a Run, but the time limit of the LN will nearly always be reached first.)

The LN-CCD camera has to be turned on end in order to re-fill its LN dewar. This of course requires its removal from the PCIS, and hence the raising and replacing of the optics assembly. All in all, this is a delicate and somewhat time-consuming operation that would be better avoided.

The simple, albeit somewhat expensive solution, is the provision and attachment of a constant feed liquid Nitrogen system to the LN-CCD camera. This is a bit less simple (and more expensive) than one might at first think. The tendency of liquid Nitrogen to become just Nitrogen in any environment you or I can survive means that all LN feed systems want to create vapour locks, and continual flow or stirring in all parts of the system becomes necessary. And of course all the pumps and stirrers must operate at LN temperatures, increasing the cost, if not necessarily the difficulty.

Nonetheless, such things are standard technology, and can be provided for a price. The improvements to the usability of the PCIS from the provision of a constant feed LN system would be significant, and this enhancement of the PCIS should be a high priority.
7.2 PCIS Software Developments

This section describes the important software developments that would improve the usability or the usefulness of the PCIS.

The only critical item is the development of software to support, and as far as possible to automate, the analysis of the enormous amount of data that is produced by a PCIS Run. The capabilities of such software to a large degree determine the real usable capabilities of the PCIS, and Section 7.2.1 describes both the problem, and the software I have proposed to address it.

The succeeding sub-sections describe improvements that could be made to the existing software suite. These are basically aimed at making the PCIS slightly easier to program and operate, rather than at extending its capabilities, and so are seen as less critical developments.

7.2.1 PCIS-ODA (Output Data Analysis) Software

As has been seen, many interesting and useful PCIS applications are possible without the existence of any software designed specifically to manage and manipulate the PCIS data. These applications have relied on qualitative interpretation of the images produced, supported by a limited number of calculated or derived results produced by largely manual procedures. However, the full capabilities of the PCIS can never be exploited in this fashion.

There is clearly a need for custom software, designed to read both the RunInfo file produced by PCIS_Seq and the .spe image file produced by the camera system, and to integrate and present them in a fashion that supports the flexible selection, analysis, re-formatting and output of the data.

I have written a detailed functional specification for such software, incorporating features to exploit to the full the particular capabilities of a broad-spectrum imaging system, but time and resources have not yet allowed its production. The full specification appears at Appendix K, but there are a few features of the software it describes that I would like to highlight here.

The most obvious and immediate need is for the ability to produce palaeodose estimates from, in particular, quartz samples. Ideally this should be done without re-inventing, and indeed without re-implementing, all the complexities of the algorithms used to produce palaeodose and error estimates from the totality of luminescence data produced by an SAR-protocol run.

The simplest way to achieve this is to convert the data contained in the PCIS images to a form that can be directly fed into existing palaeodose and age estimating packages. Such packages exist to process data in the .bin file format produced by the standard Minisys. Since that file format is sufficient to support the needs of the PCIS, it has been chosen as the preferred output data format for the PCIS Data Analysis software.

Once the region of a set of images corresponding to a particular sample has been determined, producing the required output file is a relatively straightforward process. It is in the selection of regions to define as separate samples that the complexities, and the possibilities, arise.
The necessary user interactions clearly require a graphical display, which will in essence be an overlay made from (a subset of) the images corresponding to a given sample disk. This disk, and the corresponding images, may contain a number of samples. Regions corresponding to analysable samples will be defined by the user using a mouse to manipulate tools on the displayed image. In the simplest case, a rectangular grid may be moved and adjusted over the image to define the regions corresponding to the separate grains on one of the special grid-array sample disks. Individual “cells” may then be clicked for inclusion or exclusion from the set of analysable grains for output.

However, much more than this may be achieved, especially with samples more complex than square arrays of carefully selected quartz grains.

The software has access to all the images pertaining to the disk—focussing images, Pre-Heat emission images, and of course all optically-stimulated luminescence images. Any IRSL pre-wash, or any specified “post-SAR” cycles are also always recorded, and the images are available. If desired, additional cycles could be incorporated into protocols (particularly the SAR Sequence Template) specifically to support the later image-analysis of the data. This richness of available data opens up some interesting possibilities, which (with apologies to Adobe and their “Digital Darkroom” idea) I have grouped together in the specification under the heading of the “Digital Wet Lab”.

The idea behind the Digital Wet Lab is to perform on the data, after the Run, equivalent processes to those currently performed on the actual samples, before the run, in a sample preparation laboratory (a “wet lab”). The current wet lab processes are difficult, time-consuming, costly, and perhaps above all, involving as they do considerable quantities of Hydrofluoric acid, dangerous. The advantages of working on the data instead would be obvious to anyone familiar with the existing processes.

First contender for the Digital Wet Lab approach is “Digital Mineral Separation”. Rather than removing the non-quartz components with acid and alkali washes, and with polytungstate solution heavy mineral separation, the quartz grains might be identified on the basis of their luminescence response to all of the different conditions encountered during the SAR run. If on the totality of the TL, IRSL, Pre-heat and OSL data for the Natural, Test and Regen doses what is seen doesn’t look like a quartz response, then the grain can be deleted from the set of samples for analysis.

A colour display of the sample disk, assembled from false-colour images derived from different stages of the SAR process, could be designed so that the different “quartz-like” or “non-quartz-like” responses would result in different display colours. Such an algorithm—tuned by the user perhaps, so that quartz appears grey and other minerals do not—could support the automated identification of quartz grains and selection of image regions for analysis as samples.

But such a system, if developed, would have application outside the identification only of quartz, and beyond the simple square array of grains. Other minerals would also display “signature responses” to the luminescence conditions, and might also be identified for analysis. But the more interesting application is the ability to recognise and select randomly-shaped regions of quartz or other minerals, within simple slices of materials such as epoxy-set sediment samples.

The details of this proposed approach are described in the Output Data Processing Software Specification at Appendix K. The possibilities such a technique would create for micro-sampling and the cost-effective use of OSL for broader site analyses are described below in Section 7.4.
Another “Digital Wet Lab” technique which is proposed and described in Appendix K is the “Digital HF Etch”, whereby a bright outer ring may be digitally removed from a grain before analysis. This obviously corresponds to its traditional wet lab namesake, which is undertaken primarily to remove the outer surface of the grain which may have been affected by alpha radiation, and might therefore disturb the palaeodose estimates obtained. (Claims that the HF etch also improves the optical clarity of the grain, and therefore the proportion of any emissions that are detected, are ignored here. Such effects are usually relatively minor, while actual HF remains absolutely nasty.)

I do not pretend that the removal of a bright annulus from a two-dimensional projection is equivalent to removing the entire surface of the original grain – obviously the “top” and “bottom” surfaces remain in the image. But an analysis of the geometry shows that the technique could be surprisingly effective in supporting the rapid determination of the approximate relative palaeodoses of a set of samples, and the technique is so simple and cheap that it is possible to apply it to a usefully large set of samples, and statistically meaningful results could be obtained rapidly and cheaply (terms not normally associated with luminescence analysis or dating).

It is also possible to select or group grains for analysis by their apparent size (“Digital Sieving”); by factors for which there is actually no wet sample lab equivalent, such as the relative brightness of the grains; or by any other characteristics derived from the full PCIS data set available.

The Digital Wet Lab concept and its potential are discussed further in the software specification at Appendix K. It is my earnest hope that Output Data Processing software for the PCIS will be produced according to that specification, and by someone other than myself. The ability of the PCIS to undertake many potential projects, including quartz dating applications, depends on it. This is the highest priority of all the future PCIS developments described in this chapter.

### 7.2.2 Improvements to the PCIS Sequence Language

The PCIS Sequence software emerged from what was originally a temporary utility, and still shows some signs of its origins. The lack of a consistent design paradigm underlying the supported instruction set is particularly evident.

The most serious deficiency lies in the area of the user interaction commands, used when a PCIS Run must pause for operator actions (such as changing filters) or data entry (such as entering sample descriptions). There are two issues with this command group:

1. the syntax of the commands is sub-optimal, having too many positional text parameters, making the commands difficult both to write and to read; and
2. the commands use the same VB6 window definition as the “PAUSE” button interface, leading to some unexpected behaviours in PCIS_Dev.

Neither of these issues is critical, or demands immediate attention. Because the user interaction commands are usually called from low-level utilities, rather than from the Sequence file itself, their construction is hidden from most PCIS users, who would simply call the packaged functionality via an existing, well-defined include file.
The second issue arises when the user-interaction “PAUSE” window is open at the same time that the Sequence executes a user-interaction command requiring the same window, but with different features enabled. The situation can then arise that the Sequence continues, but the “PAUSE” window is still displayed. The operator must then click the “Continue” button to clear the display of the window.

This behaviour can only arise when the Sequence is being single-stepped; and since that capability is only available in the Developer’s Edition of the software (PCIS_Dev), the issue can never arise with the Runtime Edition (PCIS_Seq), and so cannot affect a real PCIS Run.

The solution to both issues is to implement a new set of user interaction statements in PCIS_Seq, which use their own, newly-defined VB6 window. Existing Sequence and Include code would then be modified to use the new, better-constructed commands.

Because this issue only affects Sequence development, and does not affect real PCIS Runs, it is not regarded as critical or urgent and will not be attended to soon. However, because the Sequence Development Environment (running under the Visual Studio VB6 development environment) enables both the PCIS_Seq code and Sequence files to be developed and tested at home, without any of the actual PCIS system being required, the changes will no doubt be made at some time.

Other improvements to the syntax and command set of PCIS_Seq could certainly be made, but I am not aware of any other significant deficiencies in the PCIS Sequence software.

7.2.3 Integrated PCIS Run-time Software

Integrating camera-control commands into the PCIS Sequence software, as was discussed earlier in Chapter 4, would yield slight benefits for both the functionality and the usability of the PCIS.

Functionally, it would become possible for a Sequence to change camera parameters in the middle of a PCIS Run. The only parameter I can envisage being usefully changed during a run is the binning factor applied to the CCD chip. Since there is a work-around for this (breaking the PCIS Run into multiple runs, and changing the camera parameters via the WinView software between the runs), the present restriction is not critical and does not, of itself, justify the development effort needed to integrate the camera control and PCIS Sequence control functions.

Integrated runtime software would also remove the need for the operator to start two software applications; to enter the same Run Name into both; and to gather the output data from two computers at the end of the run. Again, the present procedures are only a minor inconvenience, and do not warrant a new software development at this time.

However, if the re-development of the PCIS as “PCIS II” were to be undertaken, allowing the newest CCD devices from Princeton Instruments to be used, then the integrated runtime software would be the best solution to certain technical difficulties that would arise due to the new camera controller design. In that case, the integrated software should also be “retro-fitted” to the existing PCIS. See Section 7.5 below for further discussion of the PCIS II development and its implications.
7.3 Remote Monitoring of the PCIS

Luminescence runs on any equipment can be quite extended, and always seem to finish at inconvenient hours of the day (or, more usually, night). Scheduling demands often mean that the next run must be started as soon as possible thereafter. Provision of a constant feed LN system for the PCIS will allow much longer runs, and will only exacerbate this problem.

The benefits of being able to remotely observe and monitor the progress of a run, ideally from the comfort of home, are obvious. But remote monitoring of a system that is not allowed to come into contact with a network, or ideally with any external communications function that might disrupt a run in progress, presents certain challenges.

I spent some time considering not only how this might be achieved, but how it might be achieved very cheaply. The answer I came up with, although not yet implemented, is described here.

A colleague made the suggestion that a network-enabled web-cam pointed at the PCIS and its computer screens would meet the need. Although an excellent and innovative suggestion, I perceived three potential problems:

1. network-enabled web-cams are not very cheap;
2. the optical performance of the more affordable web-cams is inadequate; and
3. it does not solve the problem of turning on the PCIS screens so that they can be observed.
   (The PCIS computer screen savers must be set to blank the screens in order to preserve low light conditions in the machine room.)

The first two problems are solved by taking a throw-away old PC, requiring only a network card and available USB ports, and connecting to it a couple of old mid-quality digital cameras. (If you do not already have these, they can now be purchased very cheaply from most op shops.) The PC is set up so that it – and the cameras connected to it – can be operated from a remote PC over the network.

Remotely waking up the PCIS PCs so that their screens come to life is more challenging. Screen savers mean that all that is required is to move the PC’s mouse – but how to do that remotely without expensive custom equipment? The solution is quite novel.

Take a spare mouse and plug it into an available USB port on the PCIS PC (or the camera PC). (All computers will happily support multiple mice.) Tether the mouse on the bench or a shelf, using its USB cable, so that it can move but not fall off the shelf. Then take an old mobile phone (there should be one in a bottom drawer somewhere) and put a pre-paid SIM card in it. (No call credit is required, so this will only cost two or three dollars). Set the phone to no screen backlighting (or tape over the screen); then set it to “Silent” and to “Vibrate”. Plug the phone and charger into a power point near the PCIS and glue the phone to the mouse!

Now, when you want to observe either PCIS PC’s screen via the “web-cam” PC, you simply ring the phone that is glued to the mouse. The phone vibrates; the mouse moves; and the screen comes to life. One phone may be glued to both mice, or two phones and SIM cards may be used, enabling independent control of the two PCIS computer screens. This technique has been tested, and works delightfully well.
7.4 PCIS Application Directions

This section looks at a small selection of the applications I envisage will be undertaken with the PCIS in the future. Some of these, such as the optimisation of the PCIS for quartz dating, should be done as soon as is practicable, and would have been done as a part of this thesis had time and resources allowed. These applications, in a sense, feed back into the PCIS development, as their goal is to improve or demonstrate the capability of the PCIS to undertake certain important classes of work.

Other applications described here are also important, but are not in the same sense urgent. These applications, such as the dating of out-of-context pottery samples, aim to open up new ways of using luminescence analysis that are only possible with an imaging system. They demonstrate capabilities that are currently unique to the PCIS, but are not expected to contribute directly to the further development and use of the PCIS in the manner of the applications in the first category.

Finally, we will look at a development that has the potential to change the way luminescence is applied to its traditional roles in archaeology, palaeontology and geomorphology. The sediment slice analysis and micro-sampling site survey techniques are intended to reduce both the site impact and the cost of OSL analysis, to a degree that will allow its use on large numbers of samples from a given study site. OSL could then be used to help in the analysis and understanding of an entire site, instead of being restricted to the “precise” dating of a few selected samples. More precise dating techniques – including “traditional” OSL – might then be targeted at better-selected samples, genuinely associated with the events, fossils or artefacts of interest.

The applications described below comprise only a small, representative subset of the possible new applications for the PCIS. Many others have been considered and discussed – from the phreatic streamway sediment studies at Jenolan (and elsewhere) that inspired my original involvement in the project, through to studies of fluorescence, phosphorescence, and the relationship of tenebrescence to these phenomena – but space does not permit a more comprehensive survey.

7.4.1 PCIS Optimisation for Quartz Blue/UV OSL

The optical dating of sediment-derived quartz grains for palaeontological, archaeological and geomorphic investigations, using blue-stimulated OSL with UV detection, has become the defining application for luminescence readers. When the Output Data Analysis software has been written, so that palaeodose and age estimates can be derived from PCIS data, it is likely that such applications will also form a significant part of the work undertaken with the PCIS. While the ability of the PCIS to detect small UV emissions from quartz has been clearly established, all such PCIS work to date has used conservative settings appropriate to the early testing then being undertaken, and the best operating conditions and procedures to produce the optimum results are yet to be established.

Optimising the PCIS’ performance for blue/UV OSL on quartz requires that a large number of PCIS runs be undertaken with a sample whose characteristics are well known, and which is not one of the “difficult to date” samples. Repeated runs would be required to determine the optimum choices, individually and then in combination, for the following factors:
• Background subtraction should be applied to the acquired images, either at the time of acquisition or as a post-acquisition process. This subtracts the fixed (and major) component of the apparent noise (seen in PCIS images in this thesis), by subtracting a fixed “dark image” from each acquisition. This works because most of the apparent noise seen is actually a fixed pattern specific to the particular camera (and its operating temperature), and can be successfully subtracted. The remaining noise is then, of course, just noise. Background subtraction has not been applied to runs performed so far, because during development and testing of a system the data should be subject to as little manipulation as possible, and examined in its “raw” state. Background subtraction has great potential to improve the effective signal-to-noise, and hence the effective sensitivity of the PCIS for this type of work.

• The cosmic ray removal algorithms in WinView should be investigated, and the preferred algorithm routinely applied to PCIS data (either at acquisition or as a post-acquisition process). This has not been done so far, for the reasons quoted above. But cosmic ray strikes can contribute large count numbers to a small group of pixels, and at their worst could significantly affect the luminescence data. It seems likely that the spatial algorithm—which compares pixels with adjacent pixels in the same image—would be best suited to the PCIS.

• The thickness of UV-passing (UG 11 or similar) optical filters for the PCIS filter drawer. Present indications are that between 5 and 6 mm are required. The less filter used, the less attenuation of the signal photons, but the greater the potential for “shine-through” of the stimulation photons. The minimum thickness that excludes, or reduces to an acceptably small level the “shine through” photons detected, should be used.

• Perhaps most critical of all, experiments with different CCD binning factors are required to test the final sensitivity limits of the PCIS’ optics and detector. The potential for sensitivity improvements by factors of 16, 25 or even higher, while retaining a useful degree of spatial discrimination, offers the potential to study even “dim” and “hard to date” samples.

• Parameters such as the duration of each OSL “Shine” (and its associated accumulation), and the current levels to use in the stimulation LEDs, require further study. These may be adjusted to give better time discrimination so that the signal decay curve can be observed; or so that the entire luminescence signal is shone out quickly in a single accumulation. The best choices might depend on the type of quartz, and the type of study being conducted.

• All of the normal parameters applicable to an SAR OSL run—the preheat times, rates and temperatures used, the selection of Test and Regen doses, the choice of pre- and post-cycles (including IRSL stages), and the temperatures at which all OSL (including IRSL) stages are performed, may also be varied with the PCIS, as with any other system. It is hoped that these parameters will not be especially critical to the performance of the PCIS, and values that have proved appropriate with other (non-imaging) systems may be adopted.

Once this optimisation work has been performed, the ultimate detection and measurement limits of the PCIS may be compared with those of a non-imaging (PM tube-based) reader running with a single grain on the sample disk. This comparison could be conducted over a range of different quartz samples, but also over a range of grain sizes. (Larger grain sizes might appear to give an edge to the PM tube-based system because of the increased total light sum; but larger grains would also allow larger CCD binning factors for the same effective or “scaled” resolution of each grain.) These comparisons would help to determine the best grain size to use for future PCIS single-grain quartz dating applications, a further factor in optimising the PCIS for such work.
7.4.2 High Grain Count Single-Grain Quartz Dating

Traditionally, OSL “dating” of quartz, whether with single or multiple aliquot techniques, has been performed on “bulk” assemblages of grains – typically several thousand grains to a sample. The advantages of working with the large total light sum that this delivers are undeniable; but arguments that larger grain numbers per sample inherently give better and more accurate results are dubious.

First of all, the argument rests on the assumption that all the grains analysed have received, and should therefore show, the same palaeodose – i.e. that there is in fact a single palaeodose to be estimated. This in turn assumes that there has been no significant mixing, turbation, or removal and replacement of the sediments during their burial history – an assumption that appears more dubious the more I look for, and find, sources of disturbance in sedimentary structures. (This is true even in caves, where one might expect the effects of bio-turbation at least to be at a minimum.)

Secondly, the argument assumes that all of the grains make a similar contribution to the light sum. If a few very bright grains dominate the total light sum (and experience suggests that many samples are highly non-homogenous, and that this condition frequently arises), then any assumptions regarding the greater accuracy of results averaged over a larger grain count are called into question. The effective grain count over which results are averaged is little more than the number of bright grains. Multiple aliquot protocols (where the assumption of similarity is applied across multiple sample disks, rather than only within a single disk) are particularly susceptible to such effects, and this (along with the ability to date much smaller quantities of sample) may in part explain the preference in recent times for Single Aliquot Regeneration protocols.

It is also possible that bright grains have more luminescence signal storage sites, and so accumulate signal more quickly; but they may not have proportionally more of the relevant re-combination sites, and so may also saturate more quickly. This could lead to an under-estimate of the bulk sample palaeodose. Or the bright grains could even represent a different source population, brought in by turbation, with a completely different history and palaeodose to the bulk of the sample. In this case, an age determination based largely on the few bright grains could actually be quite misleading.

Finally, there is the emerging desire (and my own motivating need) to examine and understand geomorphic processes, with less or no emphasis on the date at which a particular instance occurred. In this case we assume that the sediments in a sample are not homogenous, but are mixed by the process under study; and it is the nature and extent of that mixing that we want to determine. In this case, of course, bulk sample palaeodose estimation is of no assistance.

For all of these reasons, there has been a growing interest in, and demand for, single-grain quartz dating. Prior to the PCIS there were just two ways of achieving this, each with significant drawbacks.

The first technique is simply to use a standard Minisys reader and SAR protocol, but to put only a single grain on each sample disk. The only problem with this approach – but it is an absolute showstopper – is that it is slow. Every grain is processed entirely sequentially, including its Regen beta doses, so that a reasonably thorough analysis of just a few samples from a site could consume many months of machine time. This is unsupportable for most projects, and led to the development of the second technique referred to – the laser-based single-grain reader attachment produced by Risø for the standard Minisys TL DA-15 and later luminescence readers.
Although the laser reader allows the Regen beta dose to be applied to many (up to one hundred) grains at a time, thereby dramatically reducing the run lengths, there are a number of other problems or compromises inherent in this approach.

Firstly, there is no ability with the laser reader system to collect per-grain data for any heating stage, whether that be pre-heats forming part of an OSL stage, or TL stages of a protocol. Pure TL studies of single grains cannot be conducted at all with such a system. The laser reader appears to be designed purely to undertake SAR dating of quartz grains using blue/UV OSL; yet even that is compromised by the inability to study any of the pre-heat responses of the grains.

Secondly, there are problems with correct alignment of the lasers (which is necessarily re-performed for each new sample disk), and with the potential for cross-talk between the grains. If reflected stimulation photons excite signals from grains other than the one currently being analysed, then part of their signals will be incorrectly attributed to the current grain, thereby skewing the results.

Finally, there is the assumption that stimulation with coherent (laser) light is equivalent to stimulation with non-coherent light of a similar average wavelength. To the best of my knowledge this assumption has not been rigorously tested; and since it is well known that laser light can elicit unexpected behaviours from crystals, it seems to me that it should be.

The PCIS avoids all of these shortcomings and compromises. It delivers the time savings of the laser reader (and more, since OSL shines are now also paralleled), with all of the per-grain data (and more) that the PM tube-based reader, operating on one grain per sample disk, can provide. The eventual conduct of large scale single-grain studies of quartz with the PCIS should help to build a body of knowledge regarding the behaviours of quartz, which in turn will help with the design and interpretation of experiments conducted with the more available laser-based single-grain readers.

### 7.4.3 IRSL and OSL Analyses of SIM chips

The initial attempt to stimulate a signal from a mobile phone SIM card using the TL_Probe Sequence Template, described above in Section 6.5, failed to produce any usable results. However, the potential value of SIM cards as opportunistic retrospective dosimeters for the general population remains attractive, and further investigations of this material should be conducted.

Tests for optically or thermo-optically stimulated luminescence should be conducted, using at least the standard stimulation wavelengths easily available to the PCIS (IR, red and blue), at a variety of sample temperatures. These tests could be conducted as a single PCIS Run, with increasing temperatures, and then with increasing stimulation photon energies, using a series of short-pass optical filters selected to just exclude thermal or stimulation photons for each stage. The stages would be ordered so as to minimise the operator intervention for filter changes.

This approach requires the use of multiple samples to avoid de-sensitisation effects distorting the results, and should employ samples prepared in various ways, including crushed samples. These experiments may then form the basis for a generalized Material_Probe Sequence Template, employing all the stimulation modes discussed above, for broader materials analysis applications.
7.4.4 Dating Out-of-Context Pottery Samples

"Dating" of objects using luminescence is actually a process of determining a palaeodose estimate for the object, and then turning that into an "age" estimate, essentially by just dividing it by the best estimate of the dose rate to which it has been subject throughout its life. But the dose rate can only be determined by an analysis of the object’s context – typically the sediments in which it has been buried. (Since the “age” is the time since the object’s last exposure to bleaching radiation – such as sunlight – it may be assumed that the object has been buried for the interim.) Indeed for most objects it is the context (the quartz grains within the sediments surrounding the object), rather than the object itself, for which the palaeodose is estimated, and the age determined.

But many important museum objects are “out-of-context”, meaning that we no longer have knowledge of or access to the sediments (or the context) in which they were previously buried. Even if the item itself is susceptible to luminescence analysis (and pottery objects are), the best that can be obtained is a palaeodose estimate; there is no way to turn that into a meaningful age estimate. A means of determining even an approximate age for such objects – sufficient, say, to differentiate an ancient object from a modern replica – could be of enormous value to the owning institutions.

Firstly it must be understood that if a forgery is suspected, then a simple palaeodose estimate, even where the alleged context is known, is not necessarily decisive. Modern forgeries may well have been subjected to an artificial radiation dose precisely because it is common knowledge that the palaeodose can be measured. A technique is required that is immune to such attempted deceits.

Templer (1983, 1986) recognised the potential of the internal alpha dose of single zircon grains, and explored the imaging of zircon TL. In the zircon auto-regeneration TL dating technique which was developed, the TL of individual zircon grains extracted from the pottery is measured; the zircons are stored in a dark, low-radioactivity container for at least six months; and the TL is then re-measured. The ratio of the two light sums yields an age estimate. The light sums are heavily dominated by the alpha self-irradiation component, so other (external) dose sources may be ignored.

The PCIS offers the potential to address certain issues and so improve the technique. Firstly, the PCIS allows the zircon extraction step to be avoided. A small, freshly-broken fragment is still required, but it would not be destroyed, and could eventually be returned to the original object. This could be an important consideration for rare and valuable museum artefacts. Furthermore, the extreme spectral capabilities of the PCIS would deliver a unique ability to image and measure the full range of very faint luminescence signals known to be emitted by zircon.

Finally, the PCIS would be able to detect, and separately measure, luminescence from the pottery itself – or, more correctly, from the included quartz grains which appear to provide the actual luminescence signal for pottery palaeodose estimates (Oks et al, 2011). This measurement from the region immediately surrounding the zircon would provide an independent confirmation of the age estimated by the zircon auto-regeneration, while measurements from other regions of the pottery could yield an estimate of the external "palaeodose" received. By comparing the zircon auto-regeneration age estimate with the pottery “palaeodose” estimate, the PCIS may be able to detect whether an artificial external radiation dose has been applied. Such artificial dosing would indicate a deliberate forgery, rather than the accidental mis-attribution of antiquity to an object.
7.4.5 Photo-Transfer Techniques Using Long Wavelength Emissions

Not all quartz grains emit measurable quantities of UV photons when subjected to appropriate thermal or optical stimulation, even when they have received a substantial radiation dose. They may be referred to variously as “non-sensitive”, “non-responsive” or “hard to date” samples. Photo-transfer techniques (Bailiff, 1976; Bowman, 1979) represent a possible method of deriving palaeodose estimates, and hence ages, for at least some of these samples.

The apparent lack of sensitivity in these quartz samples seems likely to be due either to a lack of the relevant storage sites for a signal; or a lack of the relevant photon-emitting re-combination sites.

[ In the blue/UV OSL quartz dating discussed here, the relevant sites are the 325 °C storage sites, which are stable over sufficiently long time periods for dating purposes, and the UV-emitting re-combination sites. However, the principles explained here may well be applicable to other minerals, and other combinations of storage and re-combination sites. ]

If the shortfall is of signal storage sites, then no long term signal is present, and the sample is probably not dateable by luminescence techniques. But if the shortfall is in UV-emitting re-combination sites, then other re-combination sites, emitting lower energy photons, might be made to serve instead. The problem lies in detecting the lower energy emitted photons against a background of optical stimulation photons, or thermal photons if using TL.

Photo-transfer techniques attempt to use lower energy storage sites as a temporary store for the charges associated with the higher energy (hence longer term and useful for dating) 325 °C sites. By performing the blue OSL at room temperature, some of the electrons freed from the 325 °C sites may end up being trapped in lower energy storage sites (which have earlier been cleared by an appropriate pre-heat), rather than reaching photon-emitting re-combination sites. In a second TL or OSL run, the signal now stored in the lower energy traps can be stimulated with much lower temperatures or photon energies, and much lower-energy emitted photons can be detected. This provides an indirect method of measuring the original, 325 °C, long-term stored signal; and, by performing the blue OSL at room temperature, largely circumvents any thermal quenching effects.

[ Note that to prevent loss of signal, photo-transfer effects are deliberately avoided in normal OSL runs by performing the OSL at an elevated temperature. Any electrons landing in storage sites with characteristic temperatures lower than the sample temperature are immediately re-evicted. ]

The PCIS, with its broad spectral range, versatility of optical and thermal stimulation and easily changed optical filters for signal selection, is an ideal tool for undertaking these long wavelength emission studies. The further ability to identify which grains, or regions of a sample, are responsible for particular emissions greatly enhances the potential of the PCIS as a tool to advance this work. Studies should be undertaken with the PCIS, using known “hard to date” quartz samples. Once developed, the method could be applied to other minerals and materials, such as the SIM card chip, which is proving resistant to more conventional luminescence analysis (see Sections 6.5 and 7.4.3).

Eventually, optional Photo-Transfer stages could be included in the generalised Material_Probe Sequence Template foreshadowed earlier, and these techniques could then be routinely applied to explorations of the luminescence behaviours of new samples or new materials.
7.4.6 Potted Sediment Slice Analysis

The greatest part of the cost of a luminescence dating study lies in the preparation of the samples. (Isotopic analysis of the sample “ends” for dose rate determination can also be costly, but can be avoided – with some loss of accuracy – by the use of in-situ NaI gamma ray spectrometry.) If this cost could be avoided for even some luminescence applications, the number of samples processed, and the range of projects to which luminescence could be applied, might both increase significantly.

The “Digital Wet Lab” functions described in Section 7.2.1, with the PCIS’ imaging capabilities, open the door to novel sample preparation methods. What I propose is that the original sample core, with as little disturbance as possible, be potted in epoxy, and then sliced into 1 mm or thinner sections. These could then be cut or broken down into pieces for analysis in the PCIS, and subjected to a normal SAR dating protocol run, involving only a little machine time per slice.

Age estimates for each of the samples could be produced quickly and cheaply by running the PCIS SAR protocol on multiple slices from each sample; and then:

- identifying the quartz regions in each slice by their luminescence response;
- applying a “Digital HF Etch” to each identified quartz grain or region;
- outputting the selected regions as a data file for palaeodose estimation; and finally
- applying the field gamma spectrometry dose rate data to produce age estimates.

The results would not achieve the same precision as “traditional” luminescence methods, but could be applied to many projects for which OSL is presently just too expensive; and could certainly help to identify those sites, or those parts of a site, that might be of the era or age of interest to the researcher, and therefore warrant further study – including, possibly, “traditional” OSL. The possibilities of this technique certainly warrant further investigation.

7.4.7 Micro-Sampling and OSL Site Survey Techniques

The Digital Wet Lab components of the Output Data Processing software (see Section 7.2.1 and the specification at Appendix K), with the potted sediment slice techniques described above, might dramatically reduce the time and cost – admittedly with some loss of precision – of OSL dating.

The present section discusses a way to combine those advantages with a micro-sampling technique that allows dramatically reduced site impact, while allowing the higher sample numbers necessary to provide an overview of the history and development of the site, and allow the optimum selection of (probably larger) samples for more detailed and precise analysis.

As the only tool which can provide an age-of-formation for any part of (most quartz-bearing) sedimentary structures, luminescence should offer a way to “survey” the site, so that the event or item of interest can be properly identified, and its relationship to dateable sediments properly established. It is my own view that, all too often, this proper understanding of the site is not established prior to the collection of samples for OSL dating.
Fossils and artefacts are rare; the sites are often small; and traditional OSL samples are quite large—typically 38 mm in diameter by 125 mm long, but larger tubes (50 mm or even 75 mm diameter) are often used. So, too often, the sample selection process is one of finding a big enough undisturbed section where not too much further damage will be caused; taking the sample (and possibly doing gamma spectrometry in the hole); and then, if the date obtained is “good”, arguing in support of the close relationship between the sediments dated and the item or event of interest.

My proposal is to use samples collected with a 3/8” (9.5 mm) tube, 100 mm or less in length. If successful, this would greatly reduce the per-sample impact on the site, and at most sites would allow a larger number of samples to be dated, and a sedimentary context to be established for the item of interest. Furthermore the cores, once epoxy potted and thin sliced, could be placed directly onto PCIS sample disks (9.7 mm diameter) with no further preparation.

Experiments were conducted using 10 mm tubes of aluminium alloy (normal workshop stock with walls approximately 1 mm thick) and 3/8” thin-walled brass tubing obtained from a modelling shop. These were tested with both straight-cut ends, and with ends chamfered on the outer face. Tests were conducted both with dry tubes, and tubes internally sprayed with “SilkoSpray” silicon oil.

It was determined that the critical factor is wall thickness, and dramatically better results were obtained with the thin-walled brass tube. Chamfering offered no benefit, and is not worth the extra cost; and dry tubes worked better than sprayed tubes, both on sample collection and on removal of the sample from the tube. The fundamental problem is that the smaller tubes tend to push the sediment away, rather than enveloping it, and a method of delivering the tube faster than I achieved with a hammer is probably required — if necessary, nail gun technology (based on .22 calibre cartridges), or something bearing some of the features of a cross-bow, could be tried. The best results obtained with a dry, thin-walled brass tube, a simple anvil and a hammer are shown below.

Figure 7.4.1 3/8” (9.5 mm) micro-sample taken in dry, non-chamfered thin-wall brass tube.
Figure 7.4.1 above shows that it is possible, in some sediments at least, to obtain adequate samples in 3/8" tubes using only an anvil and a hammer. Better sampling techniques, delivering the tube to the sediments at a higher velocity than I can achieve by hand, should dramatically increase the range of sediment types from which it is possible to reliably and successfully obtain the micro-samples. A minimum sediment fill depth of about half the tube length would be required in order to be able to cut several 1 mm thick sample slices, well away from the possibly light-affected end of the tube.

Potting of the samples could easily be conducted in the field, avoiding any disturbance to or mixing of the samples during transport. A small portable vacuum pump would be required, in order to avoid air bubbles and failure to correctly epoxy-pot the entire contents of the tube.

The sample tube is assumed to be capped at both ends with a light-proof rubber or plastic cap. The end of the tube that was hammered or fired into the sediments (the “front” of the tube) should have its cap pierced, and a clear pipe from the vacuum pump placed over the capped tube end. The sample tube is now pumped down to a decently hard vacuum. Then, with the tube still attached to the vacuum line and pump, the “back” end of the tube should be placed into a reservoir of the liquid epoxy, and the back-end cap pierced with a sharp implement. Epoxy should now flow into the tube, compacting the contained sediments towards the front end of the tube. When epoxy appears at the hole pierced in the front end (hence the clear vacuum pipe), the pump is switched off, the vacuum pipe is removed, and the sample is set aside for the epoxy to dry.

The complete potted sample will be removed from the brass tube at a later date, back at the home laboratory. This may be achieved by using the different thermal expansion rates of the epoxy and the brass. Chilling the entire sample and tube (whether in ice, dry ice or LN would be determined by experiment), and then rapidly heating the brass tube from the outside (say with water at 80 or 90 °C to avoid any effect on low-temperature luminescence storage sites), should allow the entire core to be easily tapped out of the brass tube. Then all that is required is to cut the sample into the thin slices needed for PCIS analysis using a water-cooled diamond saw.

To maintain the small sample sizes and low per-sample costs necessary for the site survey technique, gamma spectrometry (rather than off-site activity analysis of the sample ends) would be required for the determination of the historic dose rates. This might use in-situ NaI gamma spectrometry with small detector heads designed to fit the sample holes; or it might use existing laboratory-based high-resolution Germanium gamma spectrometry equipment able to accept the micro-samples. The former would unavoidably be very slow (although the detectors would presumably also be cheap, and multiple detector crystals might be used); the latter would require a larger initial investment, but would retain the low per-sample costs required for the site survey micro-sampling techniques.

Many such samples could be collected, prepared and analysed for the present cost of a single OSL sample that requires full wet lab sample preparation and off-site dose rate determination. Even if any ages so obtained were quite imprecise in absolute terms, their relativities would still provide a good overview of the site’s history, and a better guide to the selection of meaningful and dateable samples than is often available at present.

The potential of the micro-sampling site survey technique described here certainly warrants the development of the Digital Wet Lab components of the PCIS Output Data Processing software, and the continued investigation of the potted sediment slice analysis techniques.
7.5 Optical Upgrades and PCIS II – Sensitivity and Resolution

Some while ago – after the PCIS was operating in its current form – my supervisor, Dr Spooner, asked me to consider the feasibility of upgrading the existing LN-CCD detector to one of the new range of electrically cooled high-sensitivity EM-CCD devices now being offered by Princeton Instruments. The new devices have a similar quantum efficiency and spectral range to the LN-CCD, but have lower noise levels, and so should provide an improved overall system sensitivity for very small signals. As well as avoiding the need for filling with liquid Nitrogen, they also offer higher spatial resolution (1024 by 1024 pixels), and faster read-out rates than the current LN-CCD detector.

The short answer to the query is yes, it is feasible. Although the new camera controllers do not offer any equivalent to the External Trigger Input of the ST-138 (which is used in the PCIS to indicate that a clean cycle rather than an image acquisition is required), there is a simple work-around. A small modification could be made to the Include files which implement CCD Clean cycles, so that they recorded the (necessarily acquired but unwanted) images to the PCIS RunInfo file under a title like “CLEAN:”. These images would then be ignored by the PCIS Output Data Processing software, for the simple reason that the software has not been told to do anything with “CLEAN:” records.

As regards the faster read-out speed of the new detectors, while it would obviously be of benefit (especially in the case of fast heating rate TL runs), I do not feel that the present PCIS is being limited in its applications by its lack of temporal discrimination. As we have seen, there is an enormous variety of important work that may be done, exploiting the spectral range and spatial discrimination of the PCIS, which is in no way hindered by slow image read-out rates.

So, certainly feasible, but not necessarily the only option. I suggested that it would be better to build a second Imaging Reader – PCIS II – built around the new detector and completely new optics, and optimised for specific applications. The reasons for that suggestion are explained below.

The existing PCIS optics were designed to maximise the competing performance criteria of:

- Wide spectral range;
- High photon capture; and
- Fine spatial resolution.

The highest priority was given to extreme spectral range. Simultaneous recording of photons from NIR to UV would greatly improve the feasibility of zircon auto-regeneration techniques (Templer and Walton, 1983; Templer, 1986; Smith et al, 1991); and the red-NIR waveband is potentially of great importance for feldspar TL and OSL applications (Zink and Visocekas, 1996). Spatial resolution was given the lowest priority, needing only to match the 512 by 512 pixel resolution of the LN-CCD.

However, the complete PCIS allows trade-offs to be made between spatial resolution and photon counts (actually signal-to-noise, or system sensitivity), by binning of the CCD (favouring sensitivity), or by inserting aperture plates (favouring resolution). Spectral range cannot be “traded off” against sensitivity or resolution (though it may be deliberately limited by the use of optical filters).

[Note that the aperture plates in current use were designed to reduce the signal for ultra-bright samples, and any improved resolution is a side-effect. Aperture plates closer to the first mirror (M1)
inside the optics box – would probably yield a better improvement in resolution for the same factor of sensitivity loss, compared to the current filter-drawer mounted aperture plates. ]

The proposed PCIS II, with the 1024 by 1024 pixel EM-CCD, would provide the ability to optimise both sensitivity and spatial resolution, by compromising on spectral range – the one trade-off combination that the present PCIS does not offer.

To achieve both the desired resolution and the required photon capture (and thus sensitivity), the optics would need to be optimised for a much more specific region of the spectrum than the broad spectrum PCIS optics (see the Optical Report at Appendix A). However, multiple interchangeable optics assemblies could be built, each covering a section of the EM-CCD’s spectrum. (This may be seen as analogous to the use of multiple lenses of varying focal length – or effective resolution – on a camera.) The multiple optics assemblies – or lenses – could each be of a transmissive design, so that they would be far simpler to build and align than the reflective optics used in the current PCIS (see Chapter 2). Note, however, that due to the materials some of these lens designs require, simpler does not necessarily imply cheaper. In fact, the opposite is likely to hold.

The PCIS II would provide both sensitivity and spatial resolution, concurrently, at a level matching or bettering that which the PCIS can provide on either. It would, of course, be an extremely useful development. But the PCIS II would not provide (within any single experimental setup) the spectral range of the present PCIS. Both machines would have their role, and indeed complementary roles in some applications. For example, the present PCIS, with its extreme spectral range and tuned for sensitivity, would remain the ideal tool for detecting “new” signals or probing “new” materials for their luminescence properties. But the PCIS II may then be the better tool for investigating the spatial distribution and temporal properties of any signals – especially weak signals – so identified.

[ Note that the PCIS II might achieve spectacular temporal performance compared to the present PCIS if kinetic frame transfer techniques (which are supported by the EM-CCD range) were used in combination with the EM-CCD’s faster read-out rates. This would allow accumulation times down to 100 ms or less, with “gaps” or dead times between frames of less than a millisecond. This would completely eliminate temporal performance as a limiting factor in imaging applications. If Princeton Instruments bring out a 1024 by 2048 EM-CCD – which would match options offered in their LN-CCD range – then an extremely fast, highly sensitive, 1024 by 2048 pixel PCIS II could be constructed. ]

In summary, it would seem optimal to keep the existing PCIS performing the wide range of novel and interesting work of which it is already capable, and to start a new parallel development of the PCIS II. The new development would incorporate a 1024 by 1024 pixel EM-CCD (or a 1024 by 2048 pixel kinetics-mode EM-CCD); a range of interchangeable optics assemblies; and redesigned control facilities based around a revised IICU and the integrated runtime software described in Section 7.2.

Following successful completion of the PCIS II, an informed decision could be made regarding the benefits of retro-fitting an EM-CCD detector, the revised control electronics and the integrated runtime software to the existing PCIS.

My own enthusiasm for involvement in these significant developments is a matter on which, at the present time, I reserve judgement.
8 Conclusion

"Don’t let it end like this. Tell them I said something!"

Last words of Pancho Villa, Mexican freedom fighter\(^{24}\).

As we saw in the last chapter, the story of the PCIS and its applications is far from over. But this account of its conception, of its design and construction, of the applications it has performed and of the performance and potential of the PCIS as it now stands, is complete. So now it is time to review, briefly, what has, and what has not, been achieved.

I came to the PCIS project from a background of questions concerning cave geomorphology, to which there were no ready answers, and no obvious means by which answers might be obtained. These questions concerned the movement and behaviour of sediments in the two major active streamways at Jenolan Caves, and their effects on the flow of those streams. They arose from my observations as a caver and a cave diver, that sedimentary changes affecting significant parts of the stream systems were occurring on an annual to decadal time scale, incompatible with the millennia-plus time scales which are generally supposed to apply to processes in cave morphology.

Now it must be conceded that no specific answers to the geomorphic conundrum that is Jenolan Caves has emerged from my work (so far); but to be fair, I did not have specific questions. The motivation for my original involvement in the PCIS development lay in my recognition of a whole class of cave sedimentology problems, at Jenolan and presumably elsewhere, for which not only were there no adequate answers, there were no appropriate tools and methods that could be applied to obtain answers. That gap I attempted to fill with the PCIS, and we shall consider shortly the degree to which I succeeded.

And what of the more general goals to further my study and understanding of karst geomorphology alongside my efforts on the PCIS development? These were expected to occupy half of my time during my PhD – a target I fell short of as the PCIS development progressed – but were not to involve the PCIS, as they would occur in parallel with its development.

Ultimately the PCIS development project – as we have seen – simply became larger, and consumed more of my time and energy, than any of us had foreseen. If a fully usable PCIS were to be delivered, then something – either my health or my cave studies – had to give. It was to be the latter.

Nevertheless, over the years, and with the encouragement and support of the School (for which I am very grateful), I have been able to attend speleological conferences and, more importantly, to make extended field trips into karst regions of Australia far afield from the eastern Australian cave systems of which Jenolan is perhaps just the most complex example. Thus I have tried to fulfil the urging of Professor John Chappell, so many years ago now, to “pull my head out of Jenolan”, and consider the far wider variety of limestones, landscapes and caves that Australia has to offer.

\(^{24}\) This is quoted from The Ultimate Collection of Finales and Farewells (2004) by Laura Ward and Robert Allen, p. 92. Unfortunately, further research shows that, as with so many of the best quotes, this is “probably apocryphal” – it appears that none of Pancho’s companions survived the fire-fight in which he died.
And it is in those other caves that my understanding has advanced furthest. Introduced to the Naracoorte area by the fossiliferous sediment dune project to which I shall return shortly; to the Mount Gambier region by the interesting and diveable caves there (and its proximity to Naracoorte); to the Nullarbor – repeatedly – by my sheer love of the landscape and its caves, wet and dry; and to the dune cave systems of south west Western Australia by the serendipity of a conference there (and the extended field trip which that, with its two Nullarbor crossings, already implied), I became interested in the common features of, and the differences between, these caves developed in large, low relief landscapes composed of young, soft and porous limestones.

I have become convinced that beyond the obvious differences in their present forms, there are important common processes in the inception and early development of many of these caves from widely different areas. I believe that a better understanding of these common origins, from a study of all of the related caves in the context of their disparate locations, will eventually lead to a fuller understanding of each cave region, and the geomorphic histories of the individual caves.

In these goals, of better understanding the history and processes involved in the development of the cave systems of Australia’s coastal dunes and plains, I believe I have made some progress. Some conclusions concerning the development of certain types of cave in the lower south east of South Australia, and some speculations on how those might relate to the apparently very different large caves of the Nullarbor, were presented to the Australian Speleological Federation Conference (McCulloch, 2009). It is also pleasing to me, that theories about the early development of these soft limestone caves seem to be more susceptible to study with luminescence tools – such as the PCIS – than the admittedly fascinating but possibly less tractable questions concerning Jenolan Caves.

So, on the whole, I am pleased with the progress I have made with my pursuit of an understanding of the processes of karst geomorphology, despite the limited time that was eventually devoted to it. Those studies have not benefitted yet from the application of the PCIS, of course, but I shall be able to return to them, armed not only with emerging hypotheses to put to the observational test, and refine into testable theories, but also with access to what may be just the right tool to test the theories with. For what had become a part-time side study, I am reasonably satisfied.

After I had commenced the development of the PCIS (at that stage, assisting Dr Spooner), it was suggested that I undertake a PhD, and a particular project at Naracoorte Caves in South Australia was proposed as an appropriate case study for the PCIS when complete, as well as being an interesting application, of merit in and of itself. That project – a study of the fossiliferous sediment cones in the caves of the Naracoorte Caves World Heritage Area, to assess the likely effectiveness of luminescence sediment dating as a means of establishing the age of the fossils – has not been completed. As we have seen, it unavoidably awaits the development of the PCIS Output Data Processing software, as well as the running of the prepared samples in the PCIS.

But I have made an extensive survey of both fossiliferous and non-fossiliferous sediment cones and their cave contexts, both in the Naracoorte area and elsewhere, and based on the evidence seen, have developed a hypothesis (expounded in the Introduction) as to the differences we might expect to see between two different classes of sediment cone. This hypothesis, if correct, would imply that the non-fossiliferous cones are generally more susceptible to analysis by sediment dating than the more interesting fossiliferous examples. So here again I am satisfied, that both an interesting and plausible hypothesis and, in the PCIS, a means of testing it, have been developed.
And finally, of course, what of the PCIS itself? Well, I find myself very pleased with that achievement. I must thank all involved, and most especially Dr Spooner, for giving me such full control over the technical design. Faults remain, of course, and there is much that has yet to be done, but few compromises were made, and none of those fundamental. What Dr Spooner gave me responsibility for, he gave me authority over, and I thank him for the host of battles that I didn’t have to fight.

I was able to complete the PCIS development to what I believe is a highly satisfactory standard of performance and versatility, fully satisfying the principal goals of the project as detailed in Chapter 1. With completion of the Sequence Development Environment, the PCIS became a flexible and highly usable research tool that should contribute to science for many years, and may also help to address some of my cave questions. The PCIS is a mature scientific research instrument, of importance not only to traditional and emerging luminescence applications in palaeontology, archaeology and the earth sciences, but also with broad application to studies in materials science, and to the further investigation of the physics of luminescence phenomena.

To reiterate, the major achievements of the project described in this thesis are:

- Construction of novel high numerical aperture reflective optics for fast achromatic imaging from infrared to near UV wavelengths (~1050 to ~200 nm);
- Design and construction of an integrated photon-counting imaging system incorporating an automated luminescence reader, custom illumination assembly, reflective optics and liquid nitrogen-cooled charge-coupled device (LN/CCD);
- Creation of automation software for integrated system control, enabling automatic implementation of complex protocols including, but not limited to, SAR protocols;
- Implementation of comprehensive facilities to support the auditing and, when required, the precise repetition of PCIS runs;
- Demonstration of capability including IRSL, OSL and TL imaging for materials analysis, retrospective dosimetry and luminescence dating applications; and
- Achievement of world-leading capability for achromatic IRSL, OSL and TL imaging.

So I feel pleased with what has been achieved, and comfortable with what has not; and I am satisfied that the years devoted to the PCIS have been well spent.

And now it is time to rest. To go caving – perhaps to go diving. To sit, to relax, to think.

And look, and observe, and reflect.

Until something piques my curiosity.

Again...
Figure 8.1 Dr Spooner and the author, looking suitably pleased with the operation of the final automated PCIS.
References


Development and Application of a Photon Counting Imaging System


Kook, M. H., Lapp, T., Thomsen, K. J., Jain, M. and Murray, A. S., 2013, A luminescence imaging system for routine single grain OSL dating, UK Luminescence and ESR Meeting 2013, University of St Andrews, abstract only.

Lewis, Ian. D., 1976, South Australian Cave Reference Book, Cave Exploration Group (South Australia) Inc. (Occasional PaperNo. 5).


McCulloch I., Spooner N.A., Jones D., 2011, A high-sensitivity photon counting imaging system (PCIS) for luminescence analysis, Radiation Measurements 46, 1566-1570.


Spooner N.A., 2000, A photon-counting imaging system (PCIS) for luminescence analysis, Radiation Measurements, 32 (5-6), 513-521.


End of Volume 1
Printed on: 15 May 2014