NIRC2 Software Status Report for the Preship Review

1.0 Introduction

This report provides details that supplement the slides that will be presented at the NIRC2 preship review. These slides appear at

http://www2.keck.hawaii.edu:3636/realpublic/inst/nirc2/remsoft/preship/soft_pres

When reading this report please refer to the figures at the above URL, or the hardcopy that was mailed along with this report.

2.0 Background

The timeline given in slide #2 summarizes major NIRC2 software milestones. Here we provide some detail for each of these milestones

- NIRC-1 P3 software commissioned - P3 (short for "phase 3") refers to a revision of the NIRC-1 software that incorporated the Keck Keyword Layer, a standard used on all Keck instruments, and use of CSH as a command line interpreter.
- Decision to use Alad electronics and software plus P3 - Transputer-based electronics and software from UCLA to drive either the Aladdin-2 or Aladdin-3 detectors was delivered with the NIRSPEC instrument, commissioned on Keck-2 in 1999. Considerable NIRC2 software is inherited from these two systems: NIRC-1 P3 and the NIRSPEC Aladdin system.
- Software CDR - The NIRC2 software design passed its critical design review in January 1999. The review materials, report from the review board, and response of the design team are included in section 6 of the Software Documentation Binder and are discussed in section 5.0 below.
- Host computer procured (UCLA). Alad keywords & electronics tested at UCLA - In spring of 1999 the NIRC2 host, a Sun Ultra-60 running Solaris 2.6, arrived at UCLA. The alad software was then installed and tested with the transputer electronics.
- Single motor tested via P3 at UCLA - In June 1999, a motor assembly was provided to UCLA and used to test P3 motor control.
- Two motors tested via P3 at CIT. Alad keywords and electronics tested at CIT - The computer system and detector electronics were transported to the NIRC2 lab in Pasadena during Fall, 1999. At this point we began testing concurrent communication to multiple motors.
- Keyword layers complete. CLI 60%. Solaris 2.7 - By early 2000, all low level software was complete, but not fully tested. The CLI scripts to move motors to named positions and take basic images were also complete. We upgraded to Solaris 2.7. Unlike 2.6, Solaris 2.7 defaults to 64-bit mode. Due to problems in the transputer driver, we reconfigured Solaris 2.7 to run in 32-bit mode.

- CLI 80%. First users - In Spring 2000, the software was being used by non-programmers to begin mechanical tests.
- Full-up software testing begins - In June 2000, multiple user accounts were created and all software excepting scripts for grism spectroscopy and the daemon for pupil mask rotation was being exercised.
- Released to selected users for beta test - In October 2000, a selected set of end-users began formal testing. The results are discussed in section 6.0 below.
- PMR software installed - In February of 2001, the servo daemon that tracks the telescope pupil and drives the rotating mask was first released.
- Preship Review - April 23, 2001.

3.0 Computers and Networks

As per slide #3, the NIRC2 host computer communicates with the instrument as follows

- Alad electronics via private SCSI controller - Like NIRSPEC, a dedicated SCSI controller is required to isolate transputer communications from SCSI communications with disk and tape. The maximum data rate along this link is approximately 800 kilobytes/second.
- Ten motors via RS-485 bus - Data flows only during motor moves or queries. Concurrently moving all ten motors generates approximately 200 bytes/second. The RS-485 operates at 9600 baud (1200 bytes/second). More details appear in section 6.0 below.
- PMR motor via dedicated RS-485 connection - Because this motor is sometimes driven continuously by the PMR daemon, a dedicated RS-485 connection is assigned.
- Lakeshore temperature monitor and control via RS-232
- Spectral lamps via RS-485 - Lamps are mounted on the AO bench.
- Cold head speed control via RS-485 - CCR compressors are located in the mechanical room downstairs from the instrument. All other equipment referenced above is near the instrument on the Nasmyth platform.

4.0 Overview

The twelve subsystems that comprise NIRC2 software are listed in the table on slide #4. The relationship between these subsystems is given in the top level diagram on slide #4A. Here we give a brief description of each of these subsystems together with a reference to where more detailed information can be found within the Software Documentation Binder.

- CLI Scripts - The 75 NIRC2 commands are documented in section 1.2 of the Software Documentation Binder. Each performs its action either by invoking lower level facility scripts, by accessing keywords directly, or both.
- QuickLook Image Display - See slide #4B. This IDL application is essentially the same tool used to view images on NIRSPEC.

- Status Display - See slide #4C. This display was implemented using the generic *xshow* tool. The *xshow* tool will display values provided by any instrument that supports the Keck keyword interface. No NIRC2 specific code was written to produce this application.
- Log Display - See slide #4D. The log display is simply a unix “tail -f” of the status log written to by all daemons. It is monitored by the generic utility *tklogger. Tklogger* is used on all existing Keck instruments. No NIRC2 specific code was written to produce this application.
- Mechanism Keyword Library - The mechanism keyword library provides the standard Keck Keyword layer by mapping keyword requests into RPC calls that are forwarded to either the motor daemon or the io daemon.
- IO Daemon - The io daemon responds to RPC calls from the mechanism keyword library by issuing commands to one of the serial interfaces that connect the NIRC2 host computer to the temperature controller, lamp system, or cold head controls. Details for the motor and io daemons appear in section 2.1 of the Software Documentation Binder.
- Motor Daemon - The motor daemon responds to RPC calls from the mechanism keyword library by issuing commands to the serial interface that connects the NIRC2 host computer to the eleven Animatics motor controllers. Details appear in section 2.1 of the Software Documentation Binder.
- PMR Daemon - The Pupil Mask Rotator daemon tracks the telescope pupil by monitoring keywords that reflect the respective positions of the elevation axis and the AO rotator. This position is fed to the pupil mask rotation mechanism (see slide #10A).
- Motor Firmware - Ten subroutines that perform primitive functions such as reporting position, moving to a demanded position, and homing. These programs reside in the tiny processors which are part of each Animatics SmartMotor.
- Alad Keyword Library - The alad keyword library provides the standard Keck Keyword layer by mapping keyword requests into RPC calls that are forwarded to the alad daemon. Details appear in section 2.2 of the Software Documentation Binder.
- Alad Daemon - The alad daemon responds to RPC calls from the alad keyword library by issuing commands to the fiber interface that connects the NIRC2 host computer to the transputer electronics. Details appear in section 2.3 of the Software Documentation Binder.
- Alad Firmware - The transputer resident OCCAM code clocks the array and transmits pixel values back to the host. Details appear in section 3 of the Software Documentation Binder.

5.0 CDR Report Response

The requirements and design presented at the January, 1999, software CDR appear in sections 6.1 and 6.2 of the Software Documentation Binder.

The complete text of the review board report and the design team response to that report is given in section 6.3 of the Software Documentation Binder. These are paraphrased in the table that appears on slide #5. The table on slide #5 also gives a short description of how each response was implemented.

6.0 Testing

Testing milestones are summarized on the timeline shown on slide #6. This section provides more detail for each of those tests.

6.1 Unit Testing

6.1.1 Single Motor at CARA

No problems found. During Spring 1999, an internal code review of all C code was held.

6.1.2 Mux Readout at UCLA

The tests at UCLA with a bare mux, May 1999, were all nominal. Noise and dark current characteristics of the electronics and Aladdin-3 array (installed January 2001) will be presented separately at the review.

6.1.3 Motor Mechanism at UCLA (June 1999)

During approximately 80 hours of burn-in testing of the communication configuration, a serious problem was tracked to an incompatibility between the animatics serial protocol and the terminal server. The problem, which involved an inter-character delay required by animatics, was eventually resolved by exercising finer control of the unix communications driver and restricting the commands used in the firmware subroutines.

6.1.4 All Motors on the Bench and Final Comms to Lakeshore, etc (Feb 99 - Jun 00)

During September 1999 each motor was commanded over 4000 times during concurrent motor move tests. Several communications problems were uncovered and resolved. No known motor communication problems have occurred since those tests.

During February 2000, communication gear was configured as closely as possible to the final configuration that will exist on the telescope.

A problem with the repeatability of the travel limits was uncovered and resolved during Spring 2000.

6.2 Full-up Software Tests

6.2.1 Initial Testing (Jun-Oct 2000)

Twenty-three bugs were identified during initial testing. These are listed in the table on slide #7A. The twelve bugs with a dark check mark were fixed. The thirteen bugs with a grey check mark are listed in the table on slide #7B. This table characterizes each bug in terms of how serious the bug is, whether a work around exists, etc. Based on this characterization, weighed against our desire to maintain a stable software base during mechanical and optical testing, we chose not to make fundamental changes to address any of the bugs in this list until after shipment at the earliest.

The table on slide #7B is ordered with the most serious bugs first. All but two of the serious bugs either have a work around or have not recurred in subsequent testing. At the time of this writing, we are investigating the state of bugs 10 and 13.

6.2.2 Nov-Dec 2000 Testing

During Nov-Dec 2000, all of the NIRC2 commands documented in section 1.2 of the Software Design Book were tested extensively by astronomers Dave Thompson and Eiichi Egami. Each used one of the numbered accounts that will be assigned to observers after NIRC2 has been installed at Keck. The text of their December 12th report appears in the table found on slide #7C.

Of the 79 comments contained in that report, approximately one third were addressed by improving the syntax checking in the command scripts. A general utility was developed for this purpose during December. Each command was then retested to confirm that invalid parameters were properly screened.

Of the 26 remaining problems reported, 13 were suggested wording changes in the output. These are being incorporated where appropriate.

Of the 13 remaining problems, all but two have been fixed. The remaining two problems lie in the *corona* and *loadstate* commands and are being debugged at the time of this writing.

6.2.3 Jan-Apr 2001 Testing

Bugs reported during Jan-Apr 2001 testing are listed in the table on slide #7D. Problems with setting up subframes, aborting exposures, and setting the sampling rate were all resolved. As above, work on debugging the *corona* and *loadstate* commands continues. In addition, some issues with the scripts for setting up grism spectrography are still outstanding.

6.3 Reliability

As per slide #8, of the twelve subsystems described in section 4.0 above, nine are persistent (i.e., have the potential to crash). Of those nine, seven have had no known crashes since full-up testing began with beta users last November.

The PMR daemon has had no known crashes, but has had little test time.

Of the four types of Alad_server crash types

1. Invalid parameters.
2. Insufficient CPU resources.
3. DCS unavailable.
4. crash_alad script.

all but #2, insufficient resources, has been discussed in sections 6.2.1 and 6.2.2 above.

The insufficient resource problem cropped up during initial PMR testing in early February. Five “asynchronous” crashes of the alad server occurred during that week. (i.e., unexplained crashes during normal operation). No asynchronous alad server crashes had occurred during the several months prior to that week, nor have they occurred during the several months of testing that followed.

During that week, the host computer was being used to test the PMR daemon, to run the openwindows desktop (normally openwindows is not run on the host), to make tape backups, and to recompile software. We believe the alad_server was sensitive to the increased CPU and memory loading.

The five crashes were exacerbated by an open device problem caused by *caRepeater*, a component of the telescope simulator. The same problem has bit us with other Keck instruments. Until it was discovered, the *caRepeater* problem required a reboot whenever the alad server crashed, thus increasing the recovery time for an alad server crash from half a minute to seven minutes.

In early March 2001 we expanded the host memory from one eighth of a gigabyte to one gigabyte. We've seen no recurrence of asynchronous crashes since that week in February. This includes periods during which the PMR daemon has been run during data acquisition.

6.4 Summary

Nirc2 software has undergone over 400 hours of burn in testing. A few scripts are still being worked on, but these do not affect the overall reliability of the software. Of the nine subsystems that are persistent and therefore have the potential to crash, only the alad server has known failure modes and these are all unlikely in practice.

The primary concerns are with insufficient test time for PMR software and grism spectroscopy scripts. As of this writing, changes were being made to the animatics firmware to work around a possible switch bounce problem. Test time for these changes is also a little short.

6.5 PMR Testing

In addition to non-rotating, circular pupil masks, NIRC2 provides four rotating pupil masks. The requirement for these masks as specified in NIRC2 Functional Requirements is to track the telescope pupil to within one milliradian (0.057 degrees) at the maximum pupil rotation rate of 0.45 degrees/second.

We set up a test with a strobe light synched to the host computer clock and simulated inputs from the telescope.

These tests showed that we are able to track the telescope pupil to within 0.32 milliradians at a pupil rotation rate of 0.20 degrees/second, but with an offset of 3.05 milliradians (corresponds to 0.9 seconds of time). Graphs of these results appear on slide #10A.

A second test showed that we are able to track the telescope pupil to within 0.24 milliradians at a pupil rotation rate of 0.0083 degrees/second, but with an offset of 2.50 milliradians. Graphs of these results appear on slide #10B.

Some questions still remain regarding absolute versus relative time and initial positioning.

7.0 Documentation

Titles for the 24 documents that appear in each of the five copies of the Software Documentation Binder are given in slide #11. Each external review board member received a copy of this binder along with this report. A cross check against the documentation required by KSD-3 appears in the following section.

8.0 KSD-3 Cross Check

Keck Software Document #3 (KSD-3) defines a checklist for software presented at a Keck preship review. Slide #12 provides this checklist applied to NIRC2 software. The primary outstanding items at this writing are

- Complete extracted function headers
- Plan for integrating and testing software at CARA