Created:February 11, 1997
Modified:June 19, 1997

Near IR Imager (NIRI)—Contents

Hubert Yamada

niri_hty_TOC/02

This report gives a table of contents for the documentation for the Gemini Near Infra-Red Imager (NIRI).

niri_hty_002.fm **Near IR Imager (NIRI)—Detailed Design Document**

niri_hty_004.fm Near IR Imager (NIRI)—Software Calibration and Maintainance

niri_hty_006.fm Near IR Imager (NIRI)—Software Release Notes

niri_hty_007.fm **Near IR Imager (NIRI)—User Interface Requirements**

niri_hty_003.fm **"Near IR Imager (NIRI) to Observatory Control System"**

niri_hty_008.fm **"A&G to NIRI/GNIRS On Instrument Wavefront Sensors"**

niri_hty_005.fm **“NIRI/GNIRS Science Instrument to On Instrument Wavefront Sensor”**

Created:February 6, 1997
Modified:June 19, 1997

Gemini
Near
Infra-Red
Imager

NIRI

Near IR Imager (NIRI)—Definitions, Abbreviations and References

Hubert Yamada

niri_hty_009/00

This report lists all abbreviations and references which are specific to the Gemini "Near Infra-Red Imager" and "On Instrument Wavefront Sensor."

1 Introduction

1.1 Purpose of the document

This is a master list of all abbreviations and acronyms that are used by the software documentation for the Near Infra-Red Imager (NIRI) and On Instrument Wavefront Sensor (OIWFS).

1.2 Scope

The software product to be developed from this document will be the control system for the Near Infra-Red Imager (NIRI) and the On Instrument Wavefront Sensor (OIWFS).

1.3 Abbreviations and Acronyms

This list is incomplete. For a more complete listing, see the Abbreviations and Acronyms list in [16].

A&G	Acquisition and Guiding
A/D	Analog to Digital Converter
CICS	Core Instrument Control System
CC	Components Controller
DC	Detector Controller
GNIRS	Gemini Near Infra-Red Spectrograph
HS	Hall-effect Sensor
ICD	Interface Control Document
IS	Instrument Sequencer
N/A	Not Applicable

NIRI	Near Infra-Red Imager
OCS	Observatory Control System
OIWFS	On Instrument Wave-Front Sensor
OMS	Oregon Microsystems
TBD	To Be Determined
TCS	Telescope Control System
WCS	World Coordinate System
UH IfA	University of Hawaii, Institute for Astronomy

1.3.1 Stylistic Conventions

References to documents in Section 1.5, “References,” on page 2 are given like this [1].

1.4 Definitions

This list is incomplete. For a more complete listing, see the CICS definitions list in [17].

Hall-effect Sensor. An electronic component which is used to detect and to measure magnetic field strength.

1.5 References

[1] niri_hty_000.fm (obsolete, superseded by [2] and by [3]), “NIRI Optical-Components Controller: Implementation Issues,” Hubert Yamada, Ev Irwin, Tony Young, University of Hawaii, Institute for Astronomy.

[2] niri_hty_001.fm[1], “Near IR Imager (NIRI)—Architectural Design Document,” Hubert Yamada, University of Hawaii, Institute for Astronomy.

[3] niri_hty_002.fm, “Near IR Imager (NIRI)—Detailed Design Document,” Hubert Yamada, University of Hawaii, Institute for Astronomy.

[4] niri_hty_003.fm, “Near IR Imager (NIRI) to Observatory Control System,” Hubert Yamada, University of Hawaii, Institute for Astronomy.

[5] niri_hty_004.fm, “Near IR Imager (NIRI)—Software Calibration and Maintenance,” Hubert Yamada, University of Hawaii, Institute for Astronomy

[6] niri_hty_005.fm “NIRI Science Instrument to On Instrument Wavefront Sensor”, Hubert Yamada, University of Hawaii, Institute for Astronomy.

[7] niri_hty_006.fm “Near IR Imager (NIRI)-Software Release Notes”, Hubert Yamada, University of Hawaii, Institute for Astronomy; Steven Beard, Royal Observatory Edinburgh.

[8] niri_hty_007.fm “Near IR Imager (NIRI)-User Interface Requirements”, Hubert Yamada, University of Hawaii, Institute for Astronomy.

[9] niri_hty_008.fm “A&G to NIRI/GNIRS On Instrument Wavefront Sensor”, Hubert Yamada, University of Hawaii, Institute for Astronomy.

[10] niri_hty_009.fm, “Near IR Imager (NIRI)—Definitions, Abbreviations and References, Hubert Yamada, University of Hawaii, Institute for Astronomy

[11] NIRI Hardware-specification document(s), under preparation, Tony Young

1. The NIRI and OIWFS software documentation is currently under development. Working drafts are available by ftp from ftp://gemftp.ifa.hawaii.edu/gemini_IFA/niri.

[12] NIRI Electronics-specification document(s), under preparation, Ev Irwin

[13] NIRI Wavefront-sensor detector document(s), under preparation, Ev Irwin

[14] cics_smb_041, "CICS—Detailed Design Document," Steven Beard, Royal Observatory Edinburgh.

[15] cics_smb_022, "CICS—Architectural Design Document," Steven Beard, Royal Observatory Edinburgh.

[16] cics_smb_013, "Science Instrument to Observatory Control System", Steven Beard, Royal Observatory Edinburgh.

[17] cics_smb_039, "CICS Project Glossary and References," Steven Beard, Royal Observatory Edinburgh.

[18] GNAAC.ICD.193_195ab.fm, "ICD 1.9.3[ab]—Detector Controller to NIRI/NIRS ICS Interface", Nick C. Buchholz, NOAO.

[19] ic16110.doc, "ICD 1.6/1.10 A&G to On-Instrument Wavefront Sensors", Nick Dillon, Malcolm Stewart, Steven Beard, Brian Leckie.

[20] ag_jms_004, "Control of Wavefront Sensors", Malcolm Stewart

[21] SPE-C-G0037, *"Software Design Description"*, Gemini 8m Telescopes Project.

[22] SPE-I-G0009, "*Software Programming Standards",* Gemini 8m Telescopes Project.

[23] *"Instrumentation (Facilities and Science) ICD's Guidelines"*, Susan Wieland, Gemini 8m Telescopes Project.

[24] gscg.kkg.009, *"ICD 1a — The System Command Interface"*, Kim Gillies, NOAO.

[25] gscg.kkg.024, *"ICD 1b — The Baseline Attribute/Value Interface"*, Kim Gillies, NOAO.

[26] gscg.kkg.010, *"ICD 2 — Systems Status and Alarm Interfaces"*, Gemini 8m Telescopes Project.

[27] gscg.grp.015, *"ICD 7a — ICS Subsystem Interfaces"*, Steven Beard, Steve Wampler and Chris Mayer, Gemini 8m Telescopes Project.

[28] gscg.grp.009, "*ICD 7b— TCS Subsystem Interfaces",* Philip Taylor

[29] gscg.grp.???, "*ICD 12—Interlock System*", Peregrin McGehee

[30] gscg.grp.???, *"ICD 13 — Standard Controller*", Bret Goodrich & Andrew Johnson, Gemini 8m Telescopes Project.

[31] gscg.grp.025, *"ICD 16 — The Parameter Definition Format"*, Steve Wampler, Gemini 8m Telescopes Project.

[32] ocs.kkg.031, *"Sequence Command Specifications"*, Kim Gillies, Shane Walker & Steven Beard, NOAO/Gemini 8m Telescopes Project.

Created:February 10, 1997
Modified:June 19, 1997

Gemini
Near
Infra-Red
Imager

Near IR Imager (NIRI)—Architectural Design Document

Hubert Yamada

niri_hty_001/02

This report gives a architectural software design for the Gemini “Near Infra-Red Imager software.”

1 Introduction

This report gives an architectural software design for the Near Infra-Red Imager (NIRI) and provides description of its software decomposition.

1.1 Purpose of the document

This document describes the difference between the NIRI design and the template which is provided by the CICS, described in [15]. It is assumed that the reader of this document is familiar with that document.

The intended audience of this document is:

- the NIRI reviewers;
- the developers of the Gemini Observatory Control System (OCS);
- the developers of the Gemini Acquisition and Guidance units
- the developers of the GNIRS OIWFS components controller and other groups which will be using the design of the NIRI OIWFS.

1.2 Scope

The software product to be developed from this document will be control software for the Near Infra-Red Imager (NIRI).

1.3 Definitions, Acronyms and Abbreviations

See “Near IR Imager (NIRI)—Definitions, Abbreviations, and References”.

1.4 Stylistic Conventions

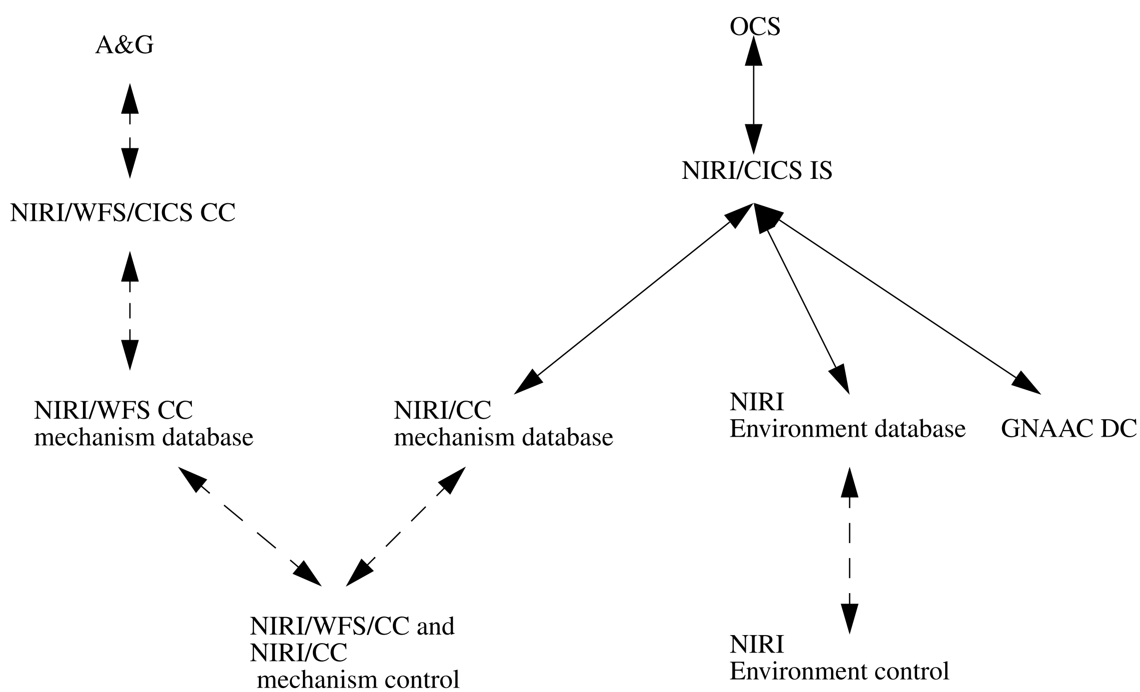
References to documents in Section 1.5, “References,” on page 2 of “Near IR Imager (NIRI)—Definitions, Abbreviations, and References” are given like this; [1].

1.5 References

See “Near IR Imager (NIRI)—Definitions, Abbreviations, and References”.

2 Implementation

At this time, the NIRI software is expected to follow the CICS architectural design very closely. The differences will be detailed in a later version of this document.



Solid lines represent commands that are linked by apply records

Created:February 6, 1997
Modified:June 19, 1997

Gemini Near Infra-Red Imager

NIRI

Near IR Imager (NIRI)—Detailed Design Document

Hubert Yamada

niri_hty_002/04

This report gives a detailed software design for the Gemini "Near Infra-Red Imager" and On Instrument Wavefront Sensor.

1 Introduction

This report presents a the software design for the control software for the Gemini Near Infra-Red Imager.

This report supersedes the preliminary document, niri_hty_000.fm, "Optical-Components Controller: Implementation Issues," which was released as part of the NIRI PDR documents.

Note that the design of many of the hardware elements is not finalized at this time. Some control algorithms may need to be modified.

1.1 Purpose of the document

This report gives a describes the differences between the design of the software for the Near Infra-Red Imager (NIRI) and the prototype instrument which is described in the CICS documents [14]. This document includes a description of the software for both NIRI and the NIRI OIWFS[1] components controllers. At some time, the OIWFS documents will be placed into a separate document.

The intended audience of this document is:

- the NIRI reviewers;
- the developers of the Gemini Observatory Control System (OCS);
- the developers of the Gemini Acquisition and Guidance units;

1. For purposes of this document, the terms NIRI OIWFS and OIWFS will be considered synonymous, though the NIRI OIWFS will be used by other systems.

- the developers of the GNIRS OIWFS components controller and other groups which will be using the design of the NIRI OIWFS.

It is assumed that readers of this document are familiar with the CICS prototype instrument system which is described in [14], [15], [16]. Throughout this document, unless otherwise stated, it is assumed that the software design for the NIRI control software is the same as the software design for the CICS prototype instrument

1.2 Scope

The software product to be developed from this document will be the control system for the Near Infra-Red Imager (NIRI).

1.3 Definitions, Acronyms and Abbreviations

See “Near IR Imager (NIRI)—Definitions, Abbreviations, and References”.

1.4 Stylistic Conventions

References to documents in Section 1.5, “References,” on page 2 of “Near IR Imager (NIRI)—Definitions, Abbreviations, and References” are given like this; [1].

1.5 References

See “Near IR Imager (NIRI)—Definitions, Abbreviations, and References”.

2 Implementation

2.1 Overview

The NIRI and NIRI OIWFS control software will be based on the Core Instrument Control System (CICS). Whenever possible, CICS commands will be duplicated. New commands and their corresponding records will be as similar to existing CICS commands as possible. If there is nothing in the CICS which can be adapted to our purposes, a new command will be designed and implemented, using the specifications in the ICD’s (especially ICD 7a and ICD 1b).

The instrument sequencer (IS) will forward sequence commands to the components controller (CC) and to the detector controller (DC); it will get the World Coordinate System (WCS) information from the TCS; it will modify the WCS information to include instrument-specific information; it will forward the modified coordinates to the DC; and it will set the TCS pointing origin.

The detector controller is a separate work package, and is described in [18].

FIGURE 1. Module Dependencies

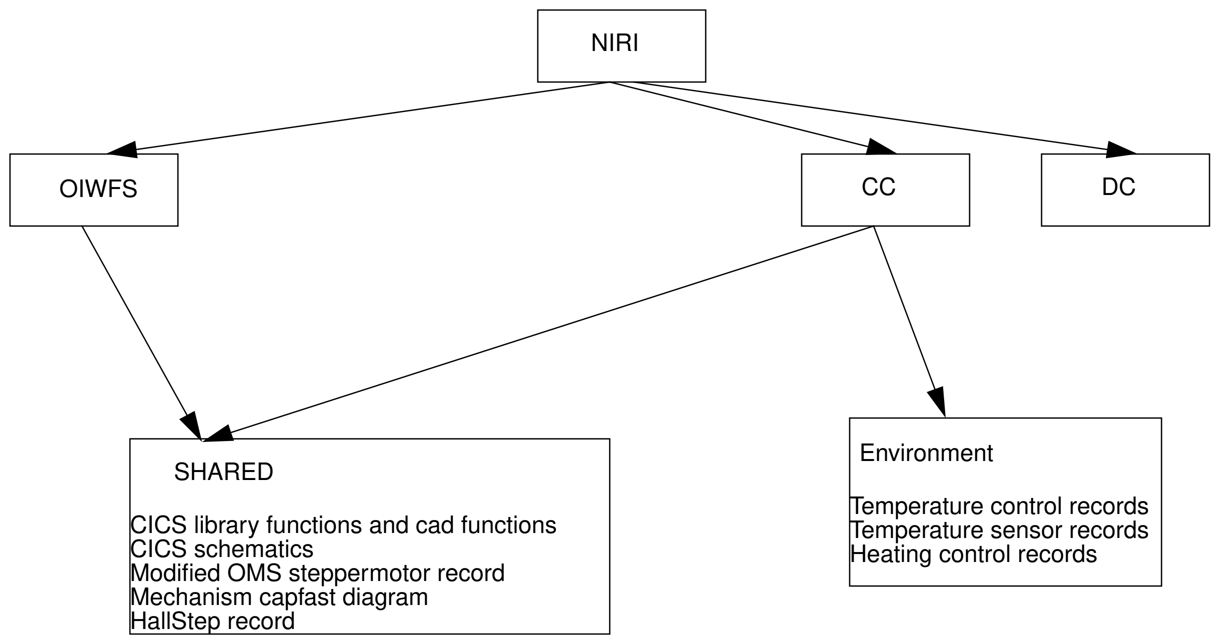
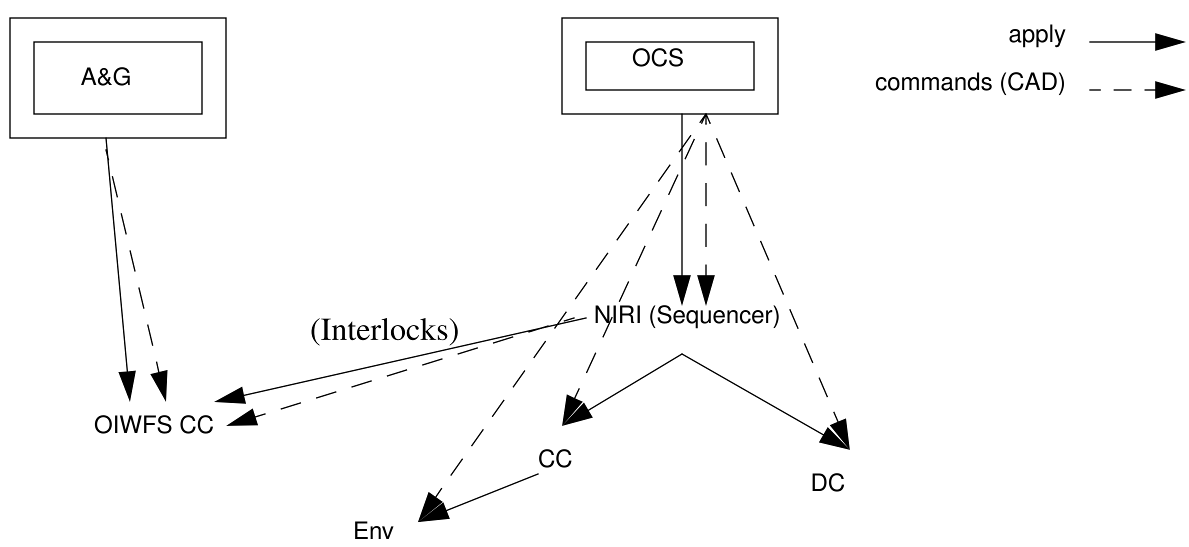


FIGURE 2. Command Flow



2.2 Static Storage

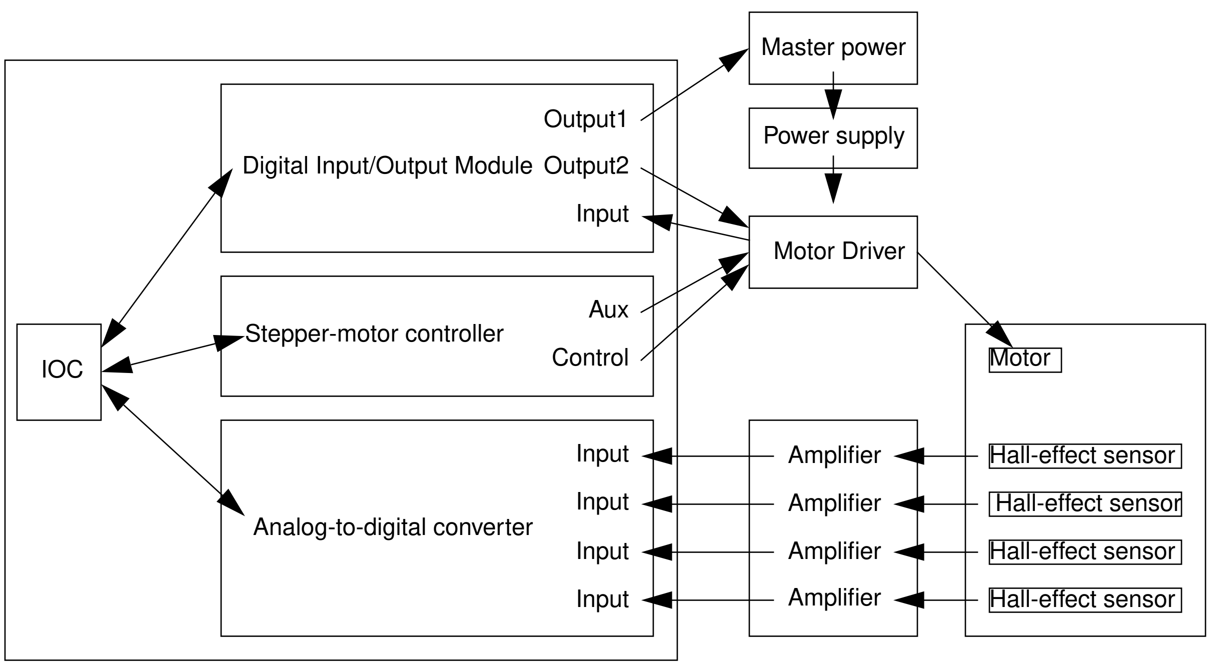
2.2.1 Static Look-Up Tables

Upon initialization of the system, calibration information will be loaded into appropriate EPICS process variables using pvload. This information will be necessary to calibrate the Hall-effect sensors. The calibration data will be generated by a separate program, which will be described in [5].

3 Stepper Motor / Hall-Effect Sensor Control

All mechanisms will be controlled using a combination of stepper motors and Hall-effect sensors.

FIGURE 3. Control system overview (single mechanism)



3.0.1 Notes on Figure 3

- All inputs in Figure 3 lead in to the same amplifier and A/D boards. This will probably not be true. The inputs may be distributed among several boards, so that the system will remain partially functional if a single board fails.
- Although the master power control is shown in this single-component diagram, this will actually be shared among all mechanical components. The master power will be shut off only when all components are idle.
- The Digital Input/Output, “Output2” and “Input” are used to check the status of the stepper motor driver, and to reset it if there is a fault.

- The Stepper-motor controller Aux output will be used to shut down the power to the motor when it is not in motion.

3.1 Hall-Effect Sensors and the Home Position

The stepper motors are not encoded. Instead, each component will have a magnet at its datum (home) position. A Hall-effect sensor will be used to detect this magnet to locate the datum position. For some components there may also be other magnets giving physical limits and other reference positions. These secondary magnets may be distinguished from the home-position magnet by using the opposite polarity, and may be distinguished from each other by a different peak field strengths at the sensor[2].

One sensor is designated as the "primary" sensor. This sensor will be connected to the "home" input of the steppermotor controller, which will speed up the datuming process. If this sensor fails, then the datum position may still be found, but will require a slower search process.

2. Different field strengths at the sensor may be produced either by varying the depth of the magnet or by using magnets with different strengths.

3.1.0.1 A discrete positioning mechanism (4-position filter wheel)

Figure 4 describes a simple, 4-position filter wheel. For mechanisms with discrete positions (e.g., filter wheels) there will be reference magnets for each distinct position, as well as the magnets for the home position. For the filter wheel, the reference magnets (and their corresponding sensors) are mounted on two concentric rings. By using a combination of magnets with different field strengths at the sensor, each position may be uniquely encoded. Note that if either sensor fails, the system will continue to operate, though the encoding scheme may no longer be unique.

FIGURE 4. A simple mechanism (wheel)

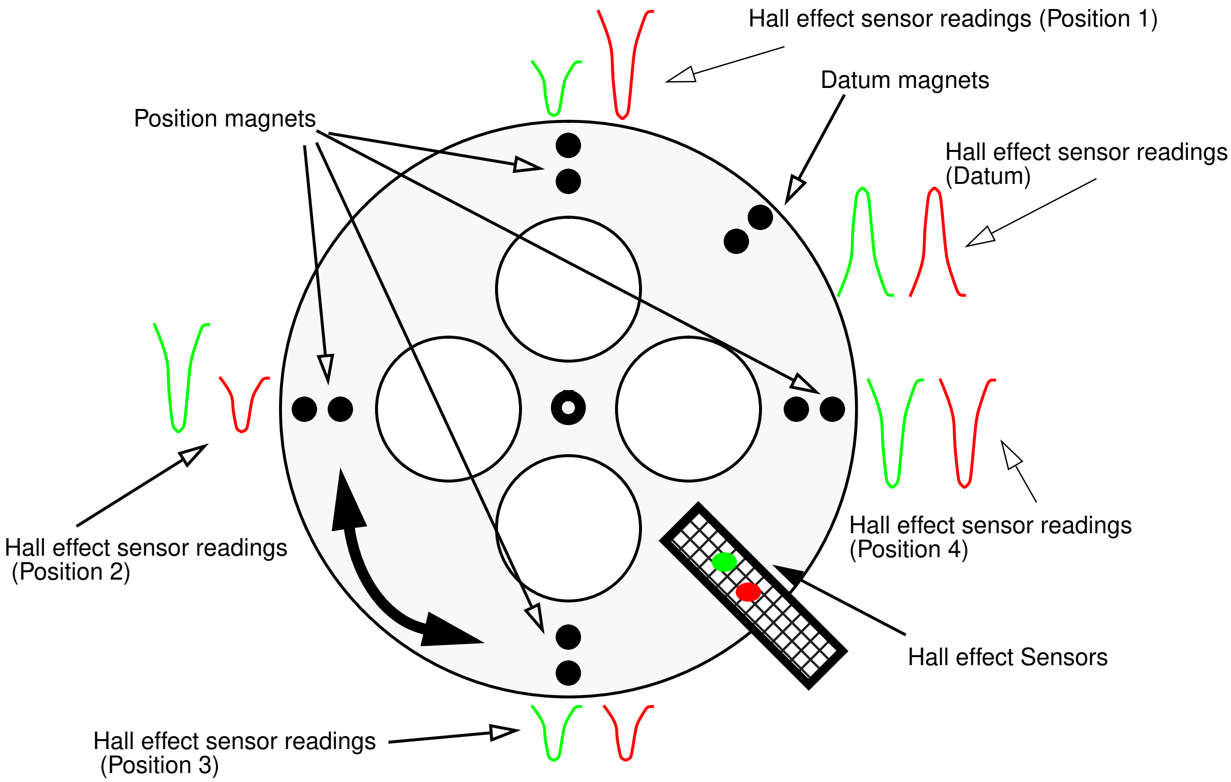


Table 1 describes the sensor values corresponding to Figure 4.

TABLE 1. Sensor Values for Figure 4

Position	Sensor 1 (V)	Sensor 2 (V)
1	-2.0	-4.0
2	-4.0	-2.0
3	-2.0	-2.0
4	-4.0	-4.0
Datum	+4.0	+4.0

Table 2 describes typical sensor values for a twelve-position wheel. Note that it is assumed that it is only possible to distinguish four different magnet strengths. In practice, it may be possible to distinguish more values.

TABLE 2. Typical Sensor Values (12 position wheel)

Position	Sensor 1 (V)	Sensor 2 (V)
1	-4.0	-4.0
2	-4.0	-3.0
3	-4.0	-2.0
4	-4.0	-1.0
5	-3.0	-4.0
6	-3.0	-3.0
7	-3.0	-2.0
8	-3.0	-1.0
9	-2.0	-4.0
10	-2.0	-3.0
11	-2.0	-2.0
12	-2.0	-1.0
Datum	+4.0	+4.0

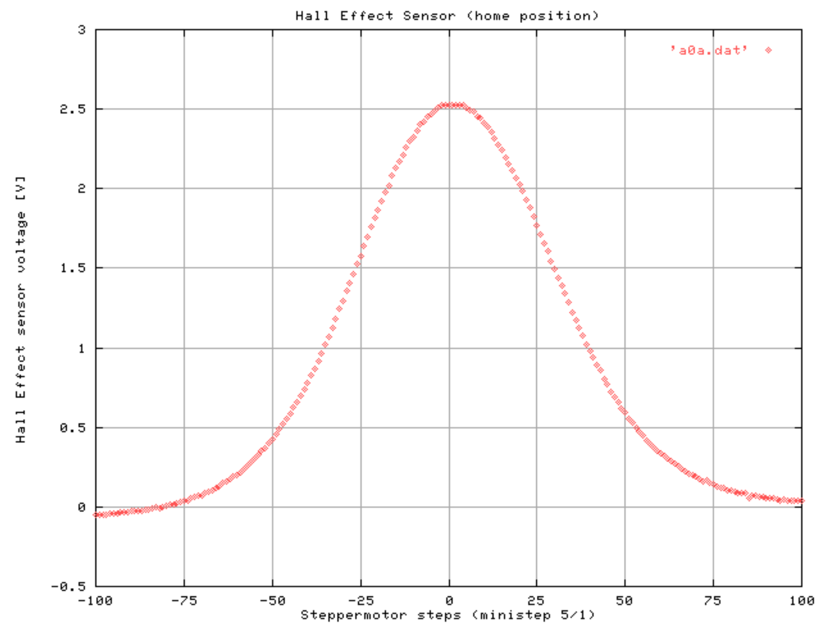
For smaller mechanisms with a small number of positions, the two-sensor arrangement will be modified to use a single back-to-back sensor pair.

3.1.0.2 Sensor data plots (home magnet)

Figure 5 is a data plot of the Hall-Effect sensor values near a datum magnet. Note that the peak is sharply defined and nearly symmetric, though careful inspection shows a slight asymmetry. The data values become negative in the wings of the peak. This is probably a real effect, due to the reversal of the direction of the magnetic field near a magnet.

This data was taken using a 5/1 mini-step stepper motor controller. In practice, the mini-steps may prove to be useless, because the motor will tend to fall into a detente when power to the motor is removed.

FIGURE 5. Hall effect sensor readings (Datum magnet)

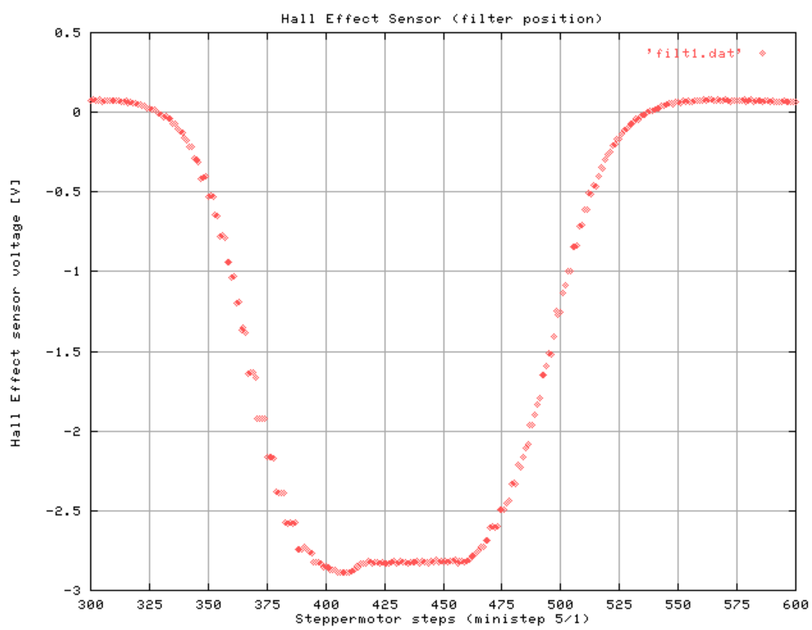


3.1.0.3 Sensor data plots (Confirmation magnet)

Figure 5 is a data plot of the Hall-Effect sensor values near a datum magnet. Note that the (negative) peak is flattened. This demonstrates the effectiveness of the Geneva mechanism which is used to turn the filter wheel. (The Geneva mechanism causes the motion of the wheel to be non-uniform. The motion of the wheel is slowed in the vicinity of the filters, making the hardware relatively insensitive to positioning errors.) The data values become positive in the wings of the peak. This is probably a real effect, due to the reversal of the direction of the magnetic field near a magnet.

This data was taken using a 5/1 mini-step stepper motor controller. In practice, the mini-steps may prove to be useless, because the motor will tend to fall into a detente when power to the motor is removed.

FIGURE 6. Hall effect sensor readings (Filter magnet)



3.1.1 A continuous-positioning mechanism (focusing stage)

In some cases, a single sensor will be unable to provide adequate resolution. Higher resolution will be provided by using a combination of magnets and sensors on different parts of the mechanism (e.g., for the focusing stage, one magnet/sensor pair on the moving part of the stage, and a second pair on the main gear).

Figure 7 describes an (over)simplified model of a focusing stage, which requires two sensor pairs to define the datum position. The coarse sensor is able to position the stage in the vicinity of the datum, unambiguously, but with poor resolution. The gear sensor is able to determine the position very accurately, but the true position can be incorrect by an integral number of gear rotations.

A dual-magnet linear sensor arrangement is shown. The coarse-positioning or "linear" magnets are placed with opposite polarities, producing two adjacent peaks, one positive, and one negative. Between the peaks, there is an ramp, which is (almost) linear. The stage is designed so that it always operates in the linear region, providing a useful (though not extremely accurate) encoding of the position of the stage. Note that any changes in the sensitivity of the sensor electronics or constant offset can affect this encoding scheme. This is sufficient, because the encoding scheme needs only to position the stage at the datum, to within one turn of the main gear.

FIGURE 7. A two-sensor arrangement (Focusing stage)

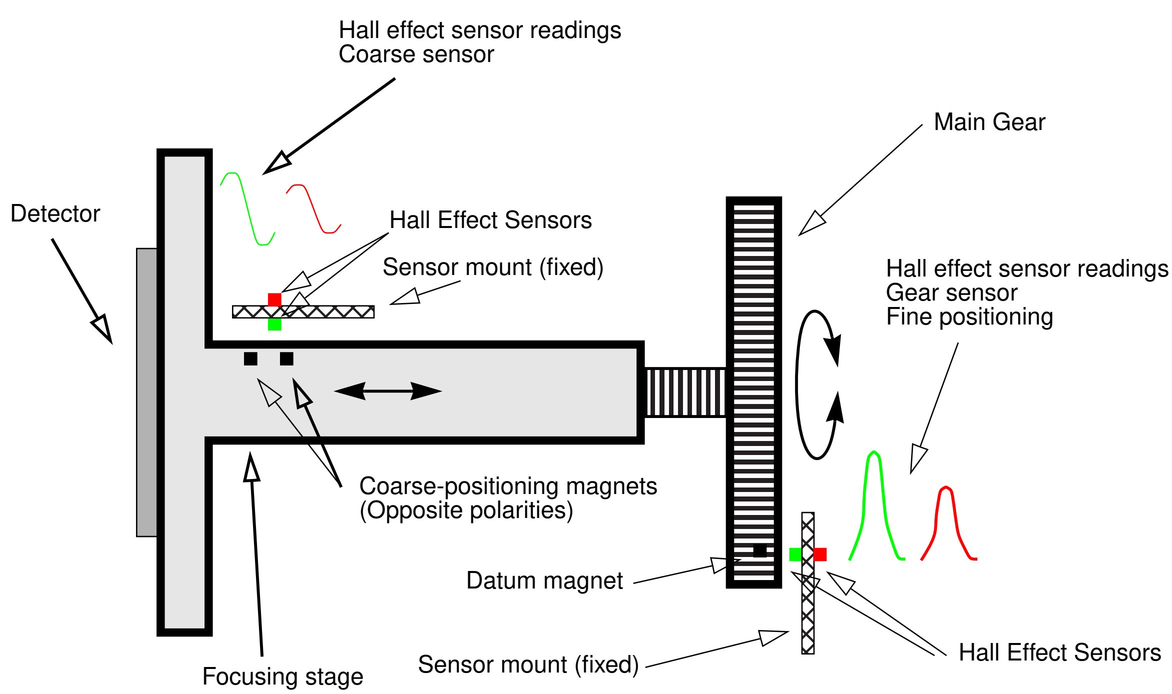
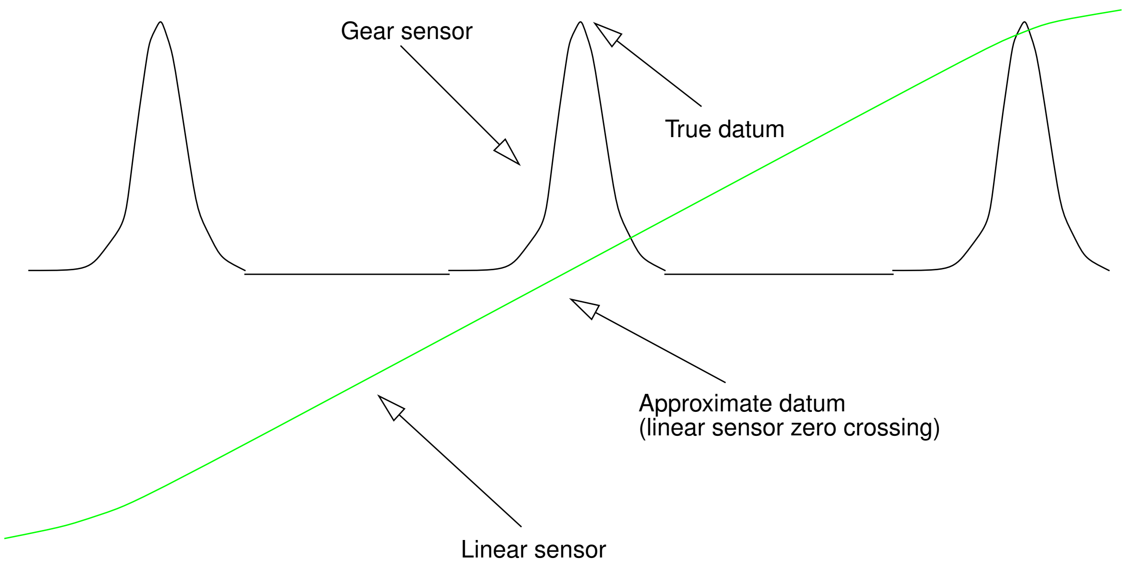


Figure 8 shows the expected reading from the Hall-effect sensors.

FIGURE 8. Sensor values (peak width highly exaggerated)



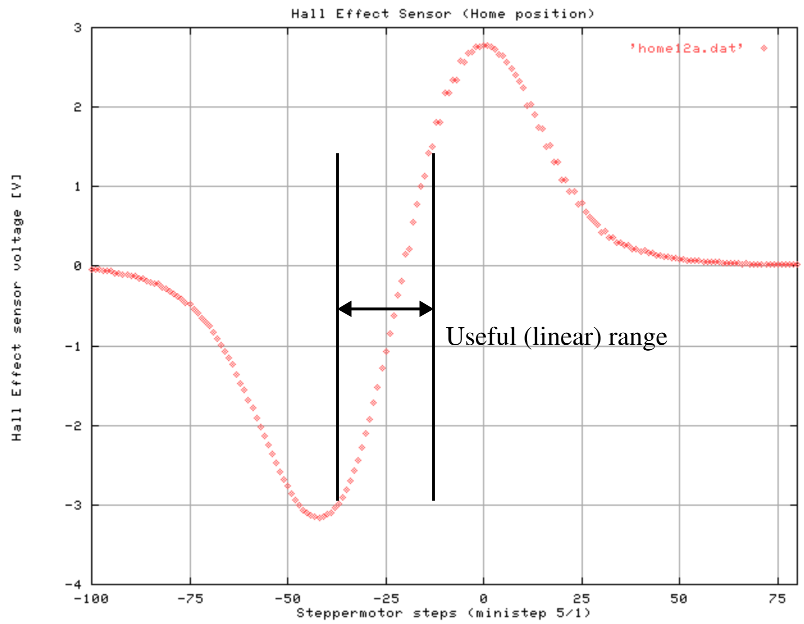
Unlike the filter wheel, the Hall-effect sensors here are mounted on top of each other. This arrangement is simpler than for the filter wheel, because it requires only a single magnet, not a pair of magnets. Unfortunately, because it is not possible to insure that the sensors are precisely concentric with each other, there is a slight difference in the peak position. This must be compensated for via an offset value, which must be determined empirically. Furthermore, the signal strength for the sensor which is further away from the magnets is noticeably weaker.

One sensor is designated as the “primary” sensor. This sensor will be connected to the “home” input of the steppermotor controller, which will speed up the datuming process. If this sensor fails, then the datum position may still be found, but will require a slower search process.

3.1.1.1 Sensor data plots ("linear" arrangement)

Figure 9 shows measured values for the Hall effect sensors in a "linear" arrangement. Note that the data values are almost linear in the region between the peaks. This provides a useful partial encoding scheme; however it is too susceptible to instabilities in the electronics to be used as a precise encoding scheme.

FIGURE 9. Hall effect sensor readings ("linear" magnet arrangement)



3.2 Peak finding algorithm

For the rest of this document, we will refer to "finding the peak value" associated with each magnet/sensor pair. The exact algorithm is TBD, but will probably be:

```
Locate the largest sensor value.
Find the two nearby points with 50% of the maximum value.
Set the datum at the midpoint of the two 50% points.
```

This symmetric algorithm should be relatively insensitive to slight changes in the gain of the electronics, and to any biases in the electronics.

(The peak-finding algorithm is implemented using a custom record [the HallStep record], which still needs to be documented.)

3.3 Stepper-motor errors

The hardware will be designed so that the stepper motors will never lose counts if all mechanisms are working properly. If any counts are lost, a problem exists which must be detected and corrected. The Hall-effect sensors may be able to detect the problem. If the sensors are able to detect a problem, an ERROR will be generated and the mechanism will be disabled to protect the hardware.

Error recovery TBD.

3.4 Backup sensors

Many, (all? TBD) of the Hall-effect sensors have backup sensors. The system will be designed to use a primary sensor to datum the corresponding mechanism and to confirm the instrument configuration, and a secondary sensor for confirmation purposes only. Either sensor may be disabled if it fails. If either sensor is disabled, it may not be possible to confirm the configuration of the corresponding mechanism. If the primary sensor fails a slower datuming algorithm will be used. If both sensors are disabled, it will not be possible to confirm the current position nor will it be possible to datum the mechanism. Disabling either sensor will generate a WARNING.

3.5 Hard Limits and Limit Switches

The software will be designed so that the hardware will never hit a hard limit or a limit switch. If a hard limit or a limit switch is ever encountered, an ERROR will be generated and operator-intervention will be required. At the very least, it will be necessary to datum the hardware.

Most components do *not* have limit switches; components that do not have limit switches are designed so that they can *occasionally* hit a hard limit without damage. In any case, hitting a hard limit or a limit switch can only happen due to a serious error, and will generate an ERROR which must be reset by appropriate operator intervention.

3.6 Simultaneous motor motion

It will not be possible to move all motors in the system simultaneously, because of power restrictions. (This will be handled by the lower level software, and should be transparent to the upper level interface.) Details TBD.

***Possible approach*:** The motors will be divided into several, disjoint sets. Each set of motors will be moved and the next set will not be moved until the previous set has completed moving.

Advantages: Easy to implement reliably.

Disadvantages: Configuration may take much longer than required.

***Alternate approach*:** Whenever a motor becomes active, a counter will be incremented. When the maximum number of motors is running, new operations will be delayed until the motor count drops below the maximum.

Advantages: Configuration takes place as quickly as possible.

Disadvantages: Harder to implement, more complicated failure modes.

3.7 Stepper Motor Control Module Failure

If the motor draws too much current, the stepper-motor control modules can enter into a fault state. The motor is unable to operate if a fault occurs. This fault generates an immediate ERROR until the stepper-motor control module is reset. (Routinely done during INIT, or a separate command, or both? TBD)

Note: A fault condition also exists if power is not applied to the control module. This is normal. Obviously, it is impossible to reset this fault. It is not possible to distinguish this from the over-current fault condition.

3.8 Diagnose

A DIAGNOSE command will be provided for each component (This should not be confused with the standard TEST and VERIFY commands.) This command will move the hardware and monitor the Hall-effect sensors in order to determine if the hardware is functioning. It should be possible to discover a single, failed, sensor. It will not be possible to distinguish a double sensor failure from a failed motor. Unlike the TEST command, the DIAGNOSE command *will* move the hardware.

3.9 Redatum

A REDATUM command will be provided for each component. (This should not be confused with the standard TEST and VERIFY commands.) This command will move to the expected datum position and confirm that it really is at the datum. If the verification fails, then an error will be returned. This command will not be used as part of the normal configuration-change procedure. It will only be used if a possible hardware problem is suspected.

4 Environment Control

4.1 Cooling motor

It will be necessary to adjust the rate of the cooling motors, by sending them commands via a serial connection. Gemini is expected to provide a serial communications record at some time in the future. If none is available, we will locate or write a special-purpose motor-control record or will add appropriate support to a stringout record. TBD.

4.2 Temperature Control

The way that the temperature-control loop will work is TBD. It is expected that temperature control will be handled with a pair of smart off-the-shelf temperature controllers. These controllers will accept ascii commands via their serial ports. It will probably be necessary to locate or to develop an EPICS temperature-controller record.

If the control algorithms are simple enough, it may be sufficient to write a custom device and driver for the standard EPICS ao record, and control the system using the EPICS state notation language.

Warm-up may be handled using heater resistors (controlled by hardware connected to a digital output port). Calculation records, look-up records, and/or subroutine records will be used to set the warm-up temperature profile.

It will be necessary to coordinate the warming of the components with the warming of the detector, to insure that the detector is always slightly warmer than the surrounding instrument.

4.3 Temperature Sensors

An undetermined number of temperature sensors will monitor the behavior of the system if the temperature is out of the valid range a WARNING (ERROR?) will be generated.

The temperature sensors will be read using a XY566 analog input board. Selected sensors will be connected to two input channels, one channel with a gain of 1, and the other with a gain of 10. This is necessary, in order to provide a dynamic range that is large enough to monitor the entire temperature range from room temperature to the lowest possible operating temperature, but at the same time providing higher resolution at the operating temperature.

5 Special Calibration Programs

There will probably be a need for special purpose software for engineering purposes, e.g., to generate the look-up tables of filter positions. These programs will never be used in normal operation. After delivery of the instrument to gemini, the programs will only need to be run after daytime engineering in which significant changes are made to the NIRI hardware.

All of these programs will reside on the host computer. These exact nature of these programs is TBD.

These programs will be documented in [5].

6 Special Epics Records

6.1 Modified records

6.1.1 Steppermotor record (OMS VME6 device/driver)

The epics steppermotor and OMS VME6 device provides support for the Oregon Microsystems VME8-8 board. We will produce a modified steppermotor record and device support. This record will be called the ‘nirimotor’ record, and may replace the steppermotor record in the standard Gemini EPICS distribution.[3]

- A timing bug in the OMS device support is fixed.

3. There are currently some significant problems with the steppermotor record. The nirimotor record may become the standard Gemini steppermotor record.

- The record will be modified to allow direct control over the AUX output. (The AUX output is used to control the power to the stepper motors from the stepper motor driver modules.) A new field will be added with three possible values: ON, OFF, AUTO. AUTO will be the default and is backwards compatible with the existing record.
- The record and device/driver support *may* be modified to allow access to the parameters which control the acceleration profile. *If* it is necessary to make this change, the default mode will be backwards compatible with the existing record.

6.2 Custom Records, Device Support, and Subroutine Records

The following will probably be necessary. They will be some combination of custom records, new device support for existing records, subroutine records, gensub records, and SNL code:

- Records/code to control the cooling and warming of the instrument;
- Records/code to optimize the motions of the filter wheels;
- Records/code to control the cooling motor;
- Records/code to control the temperature controller.

7 Implementation

7.1 Overview

The NIRI top-level directory (niri_VXXXX, where XXX is a release number) is made up of the following subdirectories

LIB. Shared utilities and schematics

CICSCC. NIRI CC code, adapted from the CICS

CICSWFS. NIRI WFS code, adapted from the CICS

NIRICC. NIRI mechanisms, WFS mechanisms, and NIRI environment control

DC. Currently the CICS DC, to be replaced by the GNAAC DC

Parts of this system can be built by including their directory together with LIB. For example, a stand-alone wavefront sensor can be built by including just the “LIB”, “CICSWFS”, and “NIRICC” directory.

The NIRI top-level directory, the LIB directory, and the CICSCC directories are adapted from the corresponding CICS directories, with the following changes:

- The niri_VXXXX/ascii directory includes files from the NIRICC/ascii directory.
- The CICSCC directory has additional mechanisms, to simulate the entire NIRI instrument

- The channel-access links to the CICS motor simulator has been replaced with channel-access links to the NIRICC databases.

In addition, the CICSWFS directory is essentially a copy of the CICSCC directory, customized with the NIRI components.

7.2 The NIRICC directory

Note: The NIRICC directory is designed to be run as a standalone package for low-level engineering of the niri mechanisms, or as a part of the complete niri system.

The NIRICC directory contains three independent databases for controlling different mechanisms: The CC database, the WFS database, and the environment-control database. It also contains databases which may be loaded to simulate the corresponding subsystems. There are also some special records, device support, and other C code which are required to support the databases.

It is possible to simulate the NIRI mechanisms, the WFS mechanisms, and the NIRI environment controls by loading the appropriate simulation database(s), which are also included in the NIRICC directory. The top-level user interface should be have identically with the simulation or with the actual hardware. Some slight differences may be noted in the lower-level engineering interfaces.

7.2.1 The CC Database

The components-controller database is made up of several instances of the same capfast hierarchal symbol, the mech symbol.

7.2.2 The WFS Database

The wavefront-sensor database is structured identically to the CC database, with a different set of mechanisms.

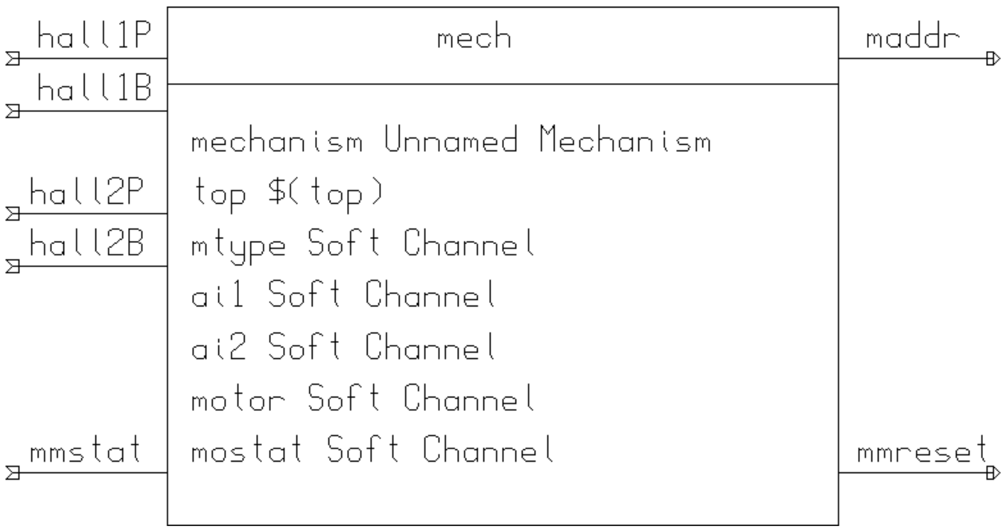
7.2.3 The ENV Database

The environment-control database currently is very simple, containing only a few records to monitor the temperature sensors. Much more needs to be added. Details TBD.

7.3 The Mech Symbol and Diagram

All components will be controlled by instances of a single capfast hierarchal symbol and the associated capfast diagram. Custom parameters are passed in via input and output ports (mostly hardware addresses) and via macro substitution.

7.3.1 The Mech Symbol



The following table describes the macro variables, inputs, and outputs of the mech hierarchal symbol.

TABLE 3. Mech.sym macro variables and ports

Macro/Port Name	Type	Value
hall1P	Input record hardware address	Primary Hall Effect sensor (wheel or main gear)
hall1B	Input record hardware address	Backup Hall Effect sensor (wheel or main gear)
hall2P	Input record hardware address	Primary Hall Effect sensor (linear sensor or unused)
hall2B	Input record hardware address	Secondary Hall Effect sensor (linear sensor or unused)
mmstat	Input record hardware address	Motor control module status (working or fault)
maddr	Motor control record hardware address	Motor controller board
mmreset	Output record hardware address	Motor control module reset (resets fault state)
mechanism	Macro variable	Text string (name of mechanism)
top	Macro variable	Prefix for generating record names
mtype	Macro variable	Type of motor controller (EPICS Device or Soft HS [simulator])
ai1	Macro variable	Type of input sensor (EPICS Device or Soft HS [simulator])
ai2	Macro variable	Type of input sensor (EPICS Device, Soft Channel [unused], or Soft HS [simulator])
motor	Macro variable	Type of motor (EPICS Device or Soft HS [simulator])
mostat	Macro variable	Type of motor status input (EPICS Device or Soft Channel [unused])

7.3.2 External interfaces (the NIRI/CICS interface)

This section describes the fields that the NIRI/CICS interface uses to control the mechanisms in the mech hierarchal symbol. These are the only fields that should be considered “public”[4]. All other fields are to be considered a private to the implementation, and may be changed at any time.

TABLE 4. External Interfaces

Field	Action	Read or Write	Action	Comments
$(top)MoveTo.VAL	Demand Position	Write	Move to new position	If an operation is already in progress, a new operation will fail.
$(top)Op.VAL	Operation Code	Write	Perform operation.	See Table 5 on page 20 for a list of operation codes If an operation is already in progress, a new operation will fail.
$(top)Stop.VAL	Stop	Write	Interrupt operation	Write a 1 to this field to interrupt an operation

4. The low-level engineering interface interacts with other fields; do not use it as an example of the correct way to interface with the mech hierarchal symbol.

TABLE 4. External Interfaces

Field	Action	Read or Write	Action	Comments
$(top)Busy.VAL	Busy	Read	Operation in progress?	Non-zero means operation in progress
$(top)Done.VAL	Done	Read	Status after operation	Non-zero means error
$(top)Hallstep.MESS	Completion message	Read	Error/Status message after an operation	Cleared by the next operation
$(top)Hallstep.DATM	Datumed?	Read or Write	Mechanism datumed?	Can be set or cleared externally or internally at any time. Set after successful datum, cleared if datum failed, cleared upon reboot. For informational purposes only; does not affect workings of record.
$(top)Hallstep.HSWN	Hall effect sensor disabled?	Read	Hall-effect sensor disabled?	Can be set or cleared externally or internally at any time.
$(top)Moving.VAL	Moving?	Read	Mechanism moving?	Non-zero means that it is moving. This is primarily for informational displays. Monitor $(top)Busy.VAL instead, because the mechanism may briefly stop moving while an operation is still in progress.
$(top)Mpos.VAL	Current Position?	Read	Mechanism current position	This field takes approximately 0.1 seconds to update. It is intended to be used for informational displays. For precise positioning, wait at least 0.1 second after Moving.VAL becomes 0 before reading this field.
$(top)Speed.VAL	Desired speed?	Read	Mechanism speed	This is the desired speed after the mechanism has accelerated, *not* the current speed.

The following table describes all operation codes. *These codes might be out of date. Check values in hsConsts.h.*

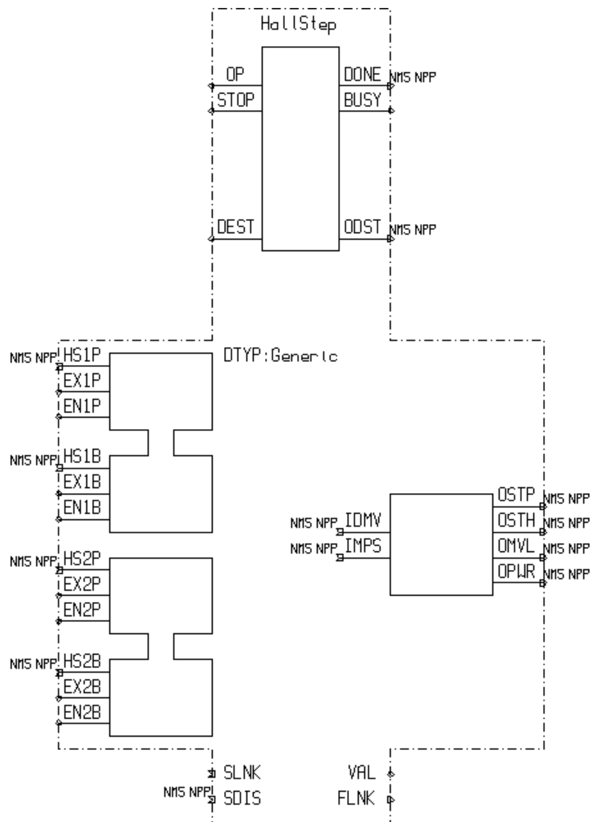
TABLE 5. Mech Hierarchical Symbol Operation codes

Code	Action	Public?	Comments
0	No Op	No	Reserved for internal use
1	Move	No	Use $(top)MoveTo.VAL instead
2	Datum mechanism	No	For engineering. Normal programs should use Datum + Diagnose
3	Redatum mechanism	Yes	
4	Cycle mechanism	No	For engineering
5	Diagnose mechanism	Yes	
6	Datum + Diagnose	Yes	

7.4 The HallStep Record and Symbol

The “mech” diagram controls the hardware via the (custom) “hallStep” record. All commands to the hardware will pass through this record. (Hall effect sensor values may be read directly from the ai record.) Nothing should interface directly to the steppermotor record, except for the records in the mech symbol.

In general, it should never be necessary to interface to the HallStep Record. Use the public fields in the mech symbol instead.



The following table describes the fields of the HallStep record.

TABLE 6. HallStep fields

Name	Summary	Description
VAL	Subr. return value	Error code returned if operation is rejected
OVAL	Old return value	
PREC	Display Precision	
STOP	Stop operation	Setting this field to non-zero cancels the current operation
OP	Operation	See Table 7 on page 24
DEST	Destination	Destination for Move operation
OPST	Operation status	Text description of current action

TABLE 6. HallStep fields

Name	Summary	Description
MESS	Operation error msg	Text description of error or other important information
HOPR	High Operating Range	Upper limit of motion (Ignored during datum)
LOPR	Low Operating Range	Lower limit of motion (Ignored during datum)
RAW	Raw mode	Device dependant (e.g. ignore limits)
DONE	Operation error code	Output link, writes zero [no error] or non-zero [error] after operation completes
BUSY	Currently processing?	Output link, writes 1 when operation begins, 0 when operation ends
DSLP	“Datum slope P”	Exact meaning depends on device support (Wheel or Stage), used for datuming
DSLB	“Datum slope B”	Exact meaning depends on device support (Wheel or Stage), used for datuming
DOFP	“Datum offset, rough P”	Exact meaning depends on device support (Wheel or Stage), used for datuming
DOFB	“Datum offset, rough B”	Exact meaning depends on device support (Wheel or Stage), used for datuming
DBSH	“Datum backlash”	Exact meaning depends on device support (Wheel or Stage), used for datuming
DOF1	“Sensor offset, 1P-1B”	Exact meaning depends on device support (Wheel or Stage), used for datuming
WTHP	“Peak width 1P”	Estimated width of sensor peak (gear or wheel)
PKP	“Peak height 1P”	Estimated height of sensor peak (gear or wheel)
WTHB	“Peak width 1B”	Exact meaning depends on device support (Wheel or Stage)
PKB	“Peak height 1B”	Exact meaning depends on device support (Wheel or Stage)
CSTP	“Cycle stop”	Stop cycle test
CTOL	“Cycle tolerance”	Sensor error for automatic stop
CSET	“Cycle set values”	(Flag) On next cycle, set expected values for cycle test from observed value
CDSA	“Cycle A dest.”	First position for cycle test
CDSB	“Cycle B dest.”	Second position for cycle test
CDSC	“Cycle C dest.”	Third position for cycle test (normally 0)
E1PA	“Cycle A expect”	Expected primary (gear or wheel) sensor value at first test position
E1PB	“Cycle B expect”	Expected primary (gear or wheel) sensor value at second test position
E1PC	“Cycle C expect”	Expected primary (gear or wheel) sensor value at third test position
V1PA	“Cycle A read”	Actual primary (gear or wheel) sensor value at first test position
V1PB	“Cycle B read”	Actual primary (gear or wheel) sensor value at first test position
V1PC	“Cycle C read”	Actual primary (gear or wheel) sensor value at first test position
E1BA	“Cycle A expect”	Expected secondary (gear or wheel) sensor value at first test position
E1BB	“Cycle B expect”	Expected secondary (gear or wheel) sensor value at second test position
E1BC	“Cycle C expect”	Expected secondary (gear or wheel) sensor value at third test position
V1BA	“Cycle A read”	Actual secondary (gear or wheel) sensor value at first test position
V1BB	“Cycle B read”	Actual secondary (gear or wheel) sensor value at second test position
V1BC	“Cycle C read”	Actual secondary (gear or wheel) sensor value at third test position
E2PA	“Cycle A expect”	Expected primary (linear) sensor value at first test position
E2PB	“Cycle B expect”	Expected primary (linear) sensor value at second test position
E2PC	“Cycle C expect”	Expected primary (linear) sensor value at third test position

TABLE 6. HallStep fields

Name	Summary	Description
V2PA	"Cycle A read"	Actual primary (linear) sensor value at first test position
V2PB	"Cycle B read"	Actual primary (linear) sensor value at second test position
V2PC	"Cycle C read"	Actual primary (linear) sensor value at third test position
E2BA	"Cycle A expect"	Expected secondary (linear) sensor value at first test position
E2BB	"Cycle B expect"	Expected secondary (linear) sensor value at second test position
E2BC	"Cycle C expect"	Expected secondary (linear) sensor value at third test position
V2BA	"Cycle A read"	Actual secondary (linear) sensor value at first test position
V2BB	"Cycle B read"	Actual secondary (linear) sensor value at second test position
V2BC	"Cycle C read"	Actual secondary (linear) sensor value at third test position
HSWN	"Warn HS"	Flag set when any hall effect sensor is disabled
HS1P	"Input Link HS1P"	Link to Hall-effect sensor input record
EX1P	"Expect HS1P"	Expected value of sensor after next move command
EN1P	"Enable HS1P"	Enable/Disable Hall-effect sensor
HS1B	"Input Link HS1B"	Link to Hall-effect sensor input record
EX1B	"Expect HS1B"	Expected value of sensor after next move command
EN1B	"Enable HS1B"	Enable/Disable Hall-effect sensor
HS2P	"Input Link HS2P"	Link to Hall-effect sensor input record
EX2P	"Expect HS2P"	Expected value of sensor after next move command
EN2P	"Enable HS2P"	Enable/Disable Hall-effect sensor
HS2B	"Input Link HS2B"	Link to Hall-effect sensor input record
EX2B	"Expect HS2B"	Expected value of sensor after next move command
EN2B	"Enable HS2B"	Enable/Disable Hall-effect sensor
IDMV[a]	"Motor finished moving"	Input link, used to monitor motion flag on motor-control record
IMPS[a]	"Motor Cur. position"	Input link, used to monitor motion flag on motor-control record
OMVL[a]	"Motor New position"	Output link, used to write new position on motor-control record
OSTH[a]	"Motor Set home"	Output link, used to trigger "set home" operation on motor-control record
OSTP[a]	"Motor Stop motion"	Output link, used to write "stop" on motor-control record
OPWR[a]	"Motor Power"	Output link, used to control power to on motor-control record (not-implemented)
ODST	"Destination Link"	Output link, after a move, writes the current position

a. IDMV, IMPS, OMVL, OSTH, OSTP, and OPWR are very closely tied with the behavior of the steppermotor record. Since the steppermotor record may be replaced, these links may need to be changed to work with the replacement record.

The following table describes all operation codes. *These codes might be out of date. Check values in hsConsts.h.*

TABLE 7. HallStep Record Operation Codes

Code	Action
0	No Op
1	Move
2	Datum mechanism
3	Redatum mechanism
4	Cycle mechanism
5	Diagnose mechanism
6	Datum + Diagnose

7.4.1 Record processing

This record is processed only when an operation code is written to the OP field.

7.4.1.1 When an operation code is processed:

```
If the record is processing another operation, the new operation fails,
   and the VAL field is set to a non-zero value.
If the operation code is invalid, the new operation fails and the VAL
   field is set to a non-zero value.
If the operation code is valid, the operation code is passed to the
   completion task, and the completion task is awakened
Record processing completes, and 0 is returned in the VAL field.
```

7.4.1.2 When the completion task is given a new operation

```
The task writes a non-zero value to the BUSY link.
The task performs the desired operation (described below), reading
   sensor values from the sensor input records, writing control codes to
   the motor-control record. If there is an error, the task sets the
   MESS field to a descriptive message.
The task writes an error code (or 0 if there is no error) to the OPST
   field.
The task writes a 0 to the BUSY link.
```

For the move command:

```
The DEST field is read.
The hardware is moved to the new location.
The sensor values are compared to the expected sensor values (not yet
   implemented).
```

For the datum command:

```
The DATM field is set to 0
The hardware performs whatever operations are necessary to datum the
   hardware (see Section 3.1 on page 5).
If the operation succeeded, the new datum is set.
If the operation succeeded, the DATM field is set to 1
```

For the redatum operation

```
The hardware performs whatever operations are necessary to datum the
   hardware (see Section 3.1 on page 5)
The new datum is not set; the MESS field indicates how different the new
   datum is from the last datum
```

For the cycle operation

```
The hardware moves to the first test position (set by the CDSA, field.
At the first test position, it reads the Hall effect sensors and
   compares them to the expected values.
If the values are not within the stated tolerance, the operation fails
   and processing terminates.
The hardware moves to the second test position (set by the CDSB, field.
At the second test position, it reads the Hall effect sensors and
   compares them to the expected values.
If the values are not within the stated tolerance, the operation fails
   and processing terminates.
The hardware moves to the third test position (set by the CDSC, field.
At the third test position, it reads the Hall effect sensors and
   compares them to the expected values.
If the values are not within the stated tolerance, the operation fails
   and processing terminates.
The process begins again, starting with the first test position, until
   the operation fails, or until the CSTP field is set to a non-zero
   value.
```

For the diagnose operation

```
The hardware moves to the first test position (set by the CDSA, field.
At the first test position, it reads the Hall effect sensors and
   compares them to the expected values.
If the values are not within the stated tolerance, the operation fails
   and processing terminates.
The hardware moves to the second test position (set by the CDSB, field.
At the second test position, it reads the Hall effect sensors and
   compares them to the expected values.
If the values are not within the stated tolerance, the operation fails
   and processing terminates.
The hardware moves to the third test position (set by the CDSC, field.
At the third test position, it reads the Hall effect sensors and
   compares them to the expected values.
If the values are not within the stated tolerance, the operation fails
   and processing terminates.
```

For the datum + diagnose operation

```
The DATM field is set to 0
The hardware performs whatever operations are necessary to datum the
   hardware (see Section 3.1 on page 5).
If the operation succeeded, the new datum is set.
The hardware moves to the first test position (set by the CDSA, field.
At the first test position, it reads the Hall effect sensors and
   compares them to the expected values.
If the values are not within the stated tolerance, the operation fails
   and processing terminates.
The hardware moves to the second test position (set by the CDSB, field.
At the second test position, it reads the Hall effect sensors and
   compares them to the expected values.
If the values are not within the stated tolerance, the operation fails
   and processing terminates.
The hardware moves to the third test position (set by the CDSC, field.
At the third test position, it reads the Hall effect sensors and
   compares them to the expected values.
If the values are not within the stated tolerance, the operation fails
   and processing terminates.
If the operation succeeded, the DATM field is set to 1
```

Created:February 3, 1997
Modified:June 19, 1997

Gemini
Near
Infra-Red
Imager

Near IR Imager (NIRI)—Software Calibration and Maintainance

Hubert Yamada

niri_hty_004/03

This document describes the software calibration programs and procedure for NIRI.

1 Introduction

This is currently an outline for a document which will be written later.

1.1 Purpose

This document describes the software calibration programs and procedure for NIRI.

The intended audience of this document is:

- the NIRI reviewers;
- the developers of the Gemini Observatory Control System (OCS);
- groups which will be using the design of the NIRI OIWFS (e.g. GNIRS);
- the maintainers of NIRI after it is delivered to Gemini.

1.2 Definitions, Acronyms and Abbreviations

See "Near IR Imager (NIRI)—Definitions, Abbreviations, and References".

1.3 Stylistic Conventions

References to documents in Section 1.5, "References," on page 2 of "Near IR Imager (NIRI)—Definitions, Abbreviations, and References" are given like this; [1].

1.4 References

See "Near IR Imager (NIRI)—Definitions, Abbreviations, and References".

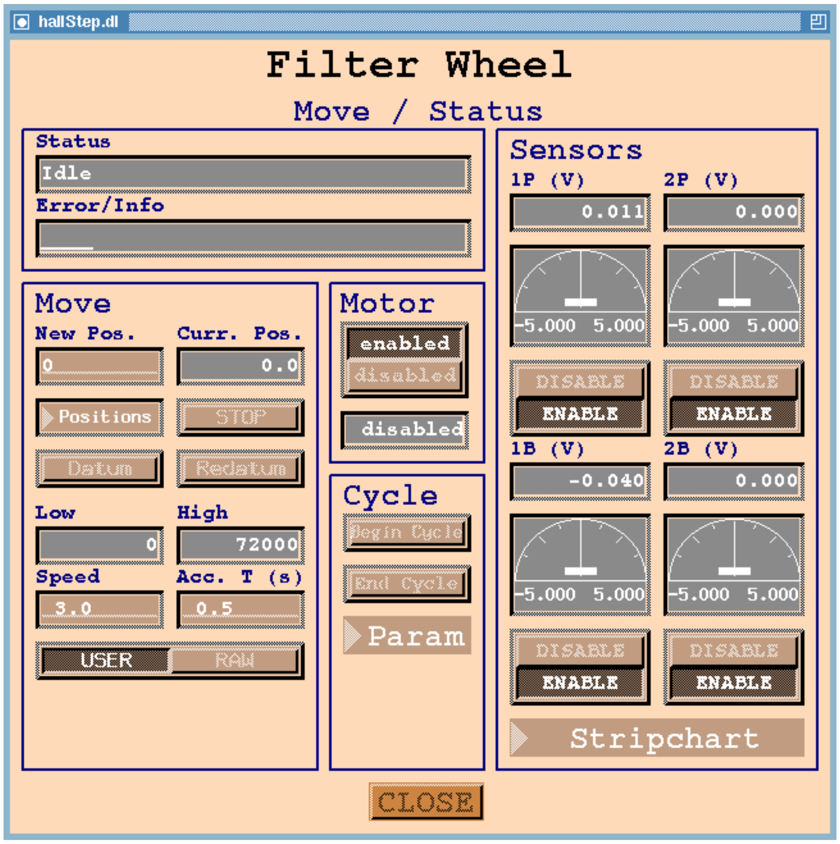
2 Calibration and Diagnostic Programs

2.1 NiriEng

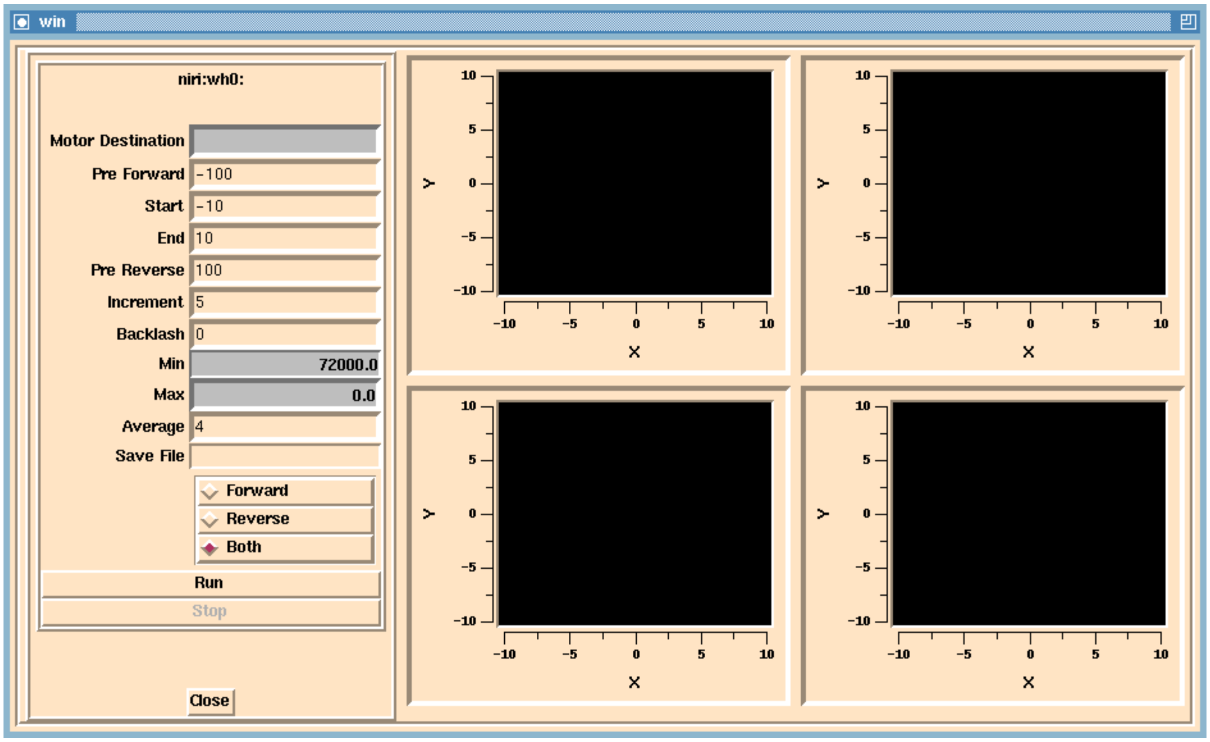
NiriEng is the low-level engineering user interface.

2.1.1 Running niriEng

This is a prototype only. The final version will probably be quite different from this version.



This is a prototype only. The final version will probably be quite different from this version.



3 Calibration Procedure

Whenever a mechanism is modified, it will be necessary to recalibrate it. This section will describe the calibration procedure which is required for each mechanism.

4 Troubleshooting

5 Warnings

Created:February 3, 1997
Modified:June 19, 1997

Gemini
Near
Infra-Red
Imager

NIRI

Near IR Imager (NIRI)—Software Release Notes

Hubert Yamada and Steven Beard

niri_hty_006/002-03

This document describes the procedure for building and installing the NIRI/CICS/OIWFS, NIRI CC, and OIWFS software, release 002 (pre-alpha prototype and simulator).

1 Introduction

1.1 Purpose

This document describes the procedure for building and installing the NIRI CC and OIWFS low-level software. This is distributed as a separate package from the NIRI CICS-based interface. This is a standalone package which provides all the tools that are necessary to interface to the NIRI CC and OIWFS hardware, to verify that the hardware is working correctly, and to calibrate the hardware.

This package does not provide all of the software that is necessary to control NIRI or the OIWFS according to gemini specifications. The higher level CAD/CAR/APPLY interface is provided by niri_V002 which is based on the CICS and was written by Steven Beard; it is documented in [TBD].

The intended audience of this document is:

- the NIRI reviewers;
- the developers of the instruments which will be using the design of the NIRI OIWFS.

1.2 Definitions, Acronyms and Abbreviations

See "Near IR Imager (NIRI)—Definitions, Abbreviations, and References".

1.3 Stylistic Conventions

References to documents in Section 1.5, "References," on page 2 of "Near IR Imager (NIRI)—Definitions, Abbreviations, and References" are given like this; [1].

1.4 References

See "Near IR Imager (NIRI)—Definitions, Abbreviations, and References".

1.5 Changes since the last release

1.5.1 Release V001

This is the initial release and was not widely distributed.

1.5.2 Release V002

Mostly bug fixes and changes in the way that configuration is handled. There is now a sequencer control screen.

2 Prerequisites

This software was developed under Solaris 2.5.1 and epics3.12.2Gem4. I do not guarantee that it will work correctly with any other release. At this time no customization is necessary. However, a modified steppermotor record is provided, for better performance. At some time in the future, the modified steppermotor record will become mandatory.

Also required:

- gunzip.

3 Building, Loading, and Using the NIRI CC Software

The niriCC package is a minimal package which is designed to run the NIRI and OIWFS hardware, and to simulate the hardware. It does not satisfy the Gemini interface requirements. It does provide all functions which are necessary to calibrate and debug the NIRI hardware.

Note: the full NIRI/CICS package, described below in "Building, Loading, and Using the CICS/NIRI/OIWFS software" on page 5, uses the niriCC package for it's low-level functions. Since all functions which are provided by the niriCC package are available in the CICS/NIRI package, there are really only three reasons to use the niriCC package directly:

- It loads a *lot* faster
- It compiles a *lot* faster
- It uses less memory (probably not a real problem)

3.1 Obtaining the Software

Ftp to gemftp.ifa.hawaii.edu, and download

```
/pub/gemini_IFA/niri/source/niriCC_V002.tar.gz
```

There is another package, niri_V002.tar.gz in that same directory. This is described below in "Building, Loading, and Using the CICS/NIRI/OIWFS software" on page 5.

3.2 Unpacking the Software

Execute the following commands:

```
mkdir niriCC_V002
cd niriCC_V002
cp (...)/niriCC_V002.tar.gz .
gunzip < niriCC_V002.tar.gz | tar -xvf -
```

Create an environment variable named NIRI_CCDIR containing the name of this directory (absolute path, not relative path).

Add $NIRI_CCDIR/bin/solaris to your path.

3.3 Configuration

3.3.1 Initial Setup

In the $NIRI_CCDIR directory, type:

```
./niriSetup
```

You will get a few warning messages, that ".cvsignore" does not exist. These are harmless and can be ignored.

3.3.2 Adjusting the prefix

You should change the prefix of the files from "niri:" to the appropriate prefix for your system. Edit the following files, replacing all instances of "niri:" with the prefix that is appropriate to you system:

- $NIRI_CCDIR/capfast/*Top.sch
- $NIRI_CCDIR/diag/niriEng
- $NIRI_CCDIR/startup/*.vws

3.3.3 Configuring a standard EPICS system

If you have an unmodified EPICS system, you must comment out the following lines in $NIRI_CCDIR/startup/startup.vws and $NIRI_CCDIR/startupSim.vws:

- ld < $(version)/bin/$(arch)/drvOms.o
- ld < $(version)/bin/$(arch)/devSmOms6Axis.o

Then comment out the following line in $NIRI_CCDIR/ascii/devSup.ascii

- "steppermotor" VME_IO "devSmOms6Axis" "OMS 6-Axis"

Finally, comment out the following line in $NIRI_CCDIR/ascii/drvSup.ascii:

- "drvOms"

This procedure will be streamlined in a later release.

The real system will not work properly using the standard OMS steppermotor driver. However the simulator will work correctly.

3.3.4 Environment subsystem

Currently, there is no simulator for the environment subsystem. If you do not have the XYCOM 566 board installed, or if you do not wish to access it with the NIRI software, you should comment out the following lines in $NIRI_CCDIR/startupSim.vws and $NIRI_CCDIR/startup.vws:

- dbLoadRecords "$(version)/data/envTop.db"
- pvload "$(version)/data/env.pv"

3.4 Building the software

In the $NIRI_CCDIR directory, type

```
gmake
```

This should build the system. The software should compile without warnings, *except*

- devSmOms6Axis.c generates a lot of warnings
- drvOms.o generates a lot of warnings
- All modules will warn "'rcsid' defined but not used."
- If your DISPLAY environment variable is not set, then the .dl files will not be created, and you will be unable to start the control screens.

3.5 Loading the NIRI CC and OIWFS prototype or simulator

Connect to your IOC and set the start-up script to (...)/bin/mv167/startupSim (simulator) or (...)/bin/mv167/startup (prototype), where (...) represents the full, absolute path of the niriCC directory. Then reboot the IOC. The IOC should reboot without any errors. At this time, you are running the NIRI CC and OIWFS low-level simulators or prototypes.

If you wish to run only the OIWFS simulator or prototype, and not the full NIRI simulator, you should edit startupSim.vws to remove any lines containing the strings "cc". *I will figure out a cleaner way to handle this at some time in the future.*

Note: the prototype requires the appropriate hardware to run, and is probably useless to anyone outside of the niri project. It is provided only for purposes of evaluation and code review.

3.6 Running the diagnostic software

- Make sure that NIRI_CCDIR is defined.
- Make sure that $NIRI_CCDIR/bin/solaris is in your path

Type:

```
niriEng
```

This should start up the niri low-level engineering screen.

The low level engineering screens are documented in [5].

4 Building, Loading, and Using the CICS/NIRI/OIWFS software

The niri package is a minimal package which is designed to run the NIRI and OIWFS hardware, and to simulate the hardware. This package is intended to satisfy all Gemini interface requirements for a science instrument. It provides all functions which are necessary to calibrate and debug the NIRI hardware.

Note: the full NIRI/CICS package uses the niriCC package (described above in “Building, Loading, and Using the NIRI CC Software” on page 2) for it’s low-level functions.

4.1 Obtaining the Software

Ftp to gemftp.ifa.hawaii.edu, and download

```
/pub/gemini_IFA/niri/source/niri_V002.tar.gz
```

There is another package, niriCC_V002.tar.gz in that same directory. This is described above in “Building, Loading, and Using the NIRI CC Software” on page 2.

4.2 Unpacking the Software

Execute the following commands:

```
mkdir niri_V002
cd niri_V002
cp (...)/niri_V002.tar.gz .
gunzip < niri_V002.tar.gz | tar -xvf -
```

4.3 Configuration

Edit niriLogin. You will need to edit this file to set the path to your installed NIRI software, and to set your path PATH.[1]

```
source niriLogin
```

The csh/tcsh configuration file “niriLogin” sets up various environment variables and your path. You must source niriLogin before running any of the niri software. It would probably be wise to source “niriLogin” in your .login or .cshrc file.

```
source niriSetup
```

This will configure the software for your EPICS system, and will copy in some standard EPICS files.

If necessary, search all for the string “niri:” and change it to the appropriate prefix for your system.

4.4 Building the system

Execute the following command:

```
gmake
```

1. This assumes that you are using csh, tcsh, or some other shell that is compatible with the csh/tcsh syntax. If not, you will need to make an equivalent file for your shell, using niriLogin as a starting point.

Note: If your DISPLAY environment variable is not set, then the .dl files will not be created, and you will be unable to start the control screens.

4.5 Loading the NIRI/CICS OIWFS/CICS prototype or simulator

Connect to your IOC and set the start-up script to (...)/bin/mv167/startupSim (simulator) or (...)/bin/mv167/startup (prototype), where (...) represents the full, absolute path to the niri_V002 directory. Then reboot the IOC. The IOC should reboot without any errors. At this time, you are running the NIRI/CICS and OIWFS/CICS low-level simulators or prototypes.

Note: the prototype requires the appropriate hardware to run, and is probably useless to anyone outside of the niri project. It is provided only for purposes of evaluation and code review.

4.6 Running the software

The system may be started by running the command:

```
niriStart
```

This will start a top level control screen from which you can start the NIRI CC, WFS, DC, and low-level engineering screens.

5 Building, Loading, and Using the CICS/OIWFS software

The CICS/OIWFS software is built identically to the CICS/NIRI/OIWFS software, except that the startup.vws (or startupSim.vws) file must be edited to start only the OIWFS packages.

This will be explained in full detail in a future release of this document.

5.1 Running the software

The system may be started by running the command:

```
niriWfsStart
```

This will start a top level control screen from which you can start the NIRI CC, WFS, DC, and low-level engineering screens.

6 Limitations of the simulation

The simulation is a reasonably accurate representation of the what the working niri system will be like, except:

- Mechanisms are idealized, i.e., all Hall-effect sensors lie on top of each other, there is no noise in the signals, there is no backlash, and the mechanisms cannot fail.
- The number of named positions for simulated mechanisms is incorrect. This aspect of the simulation will become much more realistic as the real, working mechanisms become available
- The Hall effect sensors do not read the same values as the real mechanisms.

- The "FAST" simulation mode is not really very fast for datuming.
- The environment (temperature) subsystem doesn't do anything.

Created:February 3, 1997
Modified:June 19, 1997

Gemini
Near
Infra-Red
Imager

NIRI

Near IR Imager (NIRI)—User Interface Requirements

Hubert Yamada

niri_hty_007/01

This document describes the requirements for the general user interface for NIRI.

1 Introduction

This is currently an outline for a document which will be written later.

1.1 Purpose

This document describes the requirements for the general user interface for NIRI. It fulfils the Gemini requirement that the NIRI developers provide a list of requirements for the OCS developers.

The intended audience of this document is:

- the NIRI reviewers;
- the developers of the Gemini Observatory Control System (OCS);
- the developers of the GNIRS OIWFS components controller and other groups which will be using the design of the NIRI OIWFS.

1.2 Definitions, Acronyms and Abbreviations

See “Near IR Imager (NIRI)—Definitions, Abbreviations, and References”.

1.3 Stylistic Conventions

References to documents in Section 1.5, “References,” on page 2 of “Near IR Imager (NIRI)—Definitions, Abbreviations, and References” are given like this; [1].

1.4 References

See “Near IR Imager (NIRI)—Definitions, Abbreviations, and References”.

ICD 1.9a/3.1

"Near IR Imager (NIRI) to Observatory Control System"

niri_hty_003

Hubert Yamada
1997-06-19

Version: 0.4

Issued By: Hubert Yamada

			Date:
Approved By:	____________________________	Group Manager	__________
	____________________________	Group Manager (if req)	__________
	____________________________	Systems Engineering	__________

Revision Control

1. Revision Version # 0.0
   Date: 1997-02-10
   Revised by: Hubert Yamada
   Reason for / items changed: Original prerelease version
2. Revision Version # 0.1
   Date: 1997-03-13
   Revised by: Hubert Yamada
   Reason for / items changed: Second prerelease version (with input from S. Beard). Too many changes to list. Major restructuring, a lot of new material.
3. Revision Version # 0.2
   Date: 1997-04-02
   Revised by: Hubert Yamada
   Reason for / items changed: Third prerelease version.
4. Revision Version # 0.3
   Date: 1997-04-29
   Revised by: Hubert Yamada
   Reason for / items changed: Critical Design Review version.
5. Revision Version # 0.4
   Date: 1997-06-19
   Revised by: Hubert Yamada
   Reason for / items changed: Changes in response to CDR, and comments from Steven Beard, Andy Foster, Bret Goodrich. Changes necessary to simplify interface to CICS. OIWFS commands have been moved into a separate document.

Last Modified: Thursday, June 19, 1997 5:43 pm

Chapter 1 Introduction

1–1.0 Description

1–1.1 Purpose

This Interface Control Document (ICD) serves the following purposes:

- To describe the commands and parameters that will pass between the OCS and the Near Infra-Red Imager (NIRI) sequencer and subsystems.

  This document only describes the ways in which NIRI differs from the prototypical Gemini instrument, described in ICD 1.9/3.1 [16]. It is assumed that the reader is familiar with that document. *Much of this document is copied verbatim from ICD 1.9/3.1, without further acknowledgement.*
- To provide the necessary declarations of commands, parameters and Status and Information records which is required in an "Instrument to OCS ICD." This document provides the required tables, using the required "Parameter Definition Format", described in [11].

In addition to the documents mentioned in the previous paragraphs, the specific interfaces described in this document are governed by the following parent Interface Control Documents, which describe the general properties of particular kinds of interface:

- ICD 1a [24], which describes how the OCS deals with commands and attributes and the behavior the OCS expects from the systems it commands.
- ICD 1b [25], which gives a general description of the way the Observatory Control System communicates with the other principal systems. It is also related to ICD 7a, [27], which describes the way an Instrument Control System communicates with its subsystems.
- ICD 2 [26], which describes the way in which an Instrument Control System makes information available to the outside world (and the OCS) by means of "Status Information Records".

The intended audience for this document is:

- the NIRI reviewers;
- the developers of the Gemini Observatory Control System (OCS).

1–1.2 Scope

This document describes only aspects of the NIRI software system which differ from, or are more specific than, the Gemini standard instrument, described in [16]. It is assumed that the reader is familiar with that documents.

This document does *not* describe the following topics:

- the science detector, except for sequence commands which will be forwarded to the science detector controller (see [18]);
- the NIRI hardware and electronics (see [11] and [12], summarized in [3]);

- the NIRI On Instrument Wavefront Sensor (see [9]).

The software design is not described here. For that, see [2] (currently under preparation), and [3] (currently under preparation).

1–2.0 Related Documents and Drawings

1–2.1 References

See “Near IR Imager (NIRI)—Definitions, Abbreviations, and References”.

1–2.2 Definitions, Acronyms and Abbreviations

See “Near IR Imager (NIRI)—Definitions, Abbreviations, and References”.

1–2.3 Stylistic Conventions

References to documents in Section 1.5, “References,” on page 2 of “Near IR Imager (NIRI)—Definitions, Abbreviations, and References” are given like this; [1].

Chapter 2 Details of the NIRI to OCS Interface

2–1.0 Introduction

2–2.0 Physical System Interfaces

2–2.1 Mechanical Interface

Not Applicable for a software ICD.

2–2.2 Optical Interface

Not Applicable for a software ICD.

2–2.3 Electronic Interface

The system hardware architecture for the NIRI control software is the same as described in [16], with the following exception:

- The Instrument Sequencer, Components Controller, and OIWFS Components Controller will share the same IOC.

There will also be the following electronic VME modules:

- The XVME-240 Digital I/O Board, (shared between the NIRI CC, the NIRI OIWFS, and the interlock bus):
  - To support the interlock bus (Gemini standard, see [29]);
  - To control the stepper-motor master power supply;
  - To monitor the stepper-motor control modules;
  - To reset the stepper-motor control modules;
  - To control the heating system.
- The serial ports on the VME-167 (via the MVME-712/M module) and a serial I/O board (board TBD, depending on available EPICS support.):
  - To control the NIRI/OIWFS temperature controllers (three ports);
  - To control the NIRI cooling motors (one port);
- Four custom-built VME sensor-support boards:
  - To read out the NIRI Hall-effect sensors;
  - To read out the NIRI temperature sensors;
  - To read out the NIRI OIWFS Hall-effect sensors (one board, shared);

- Two XVME-566 Analog Input Modules
  - To read out the VME sensor-support boards for the NIRI temperature sensors
  - To read out the VME sensor-support boards for the NIRI Hall-effect sensors
  - To read out the VME sensor-support boards for the NIRI OIWFS Hall-effect sensors (one board, shared)
- Three Oregon Microsystems VME8-8 Stepper-motor controller boards
  - To control the NIRI stepper-motors
  - To control the NIRI OIWFS stepper-motors (one board, shared)

2–2.4 Mass/Balance

Not Applicable for a software ICD.

2–2.5 Thermal Interface

Not Applicable for a software ICD.

2–3.0 Software/Control Function Interface

2–3.1 Overview

2–3.1.1 Communications infrastructure

The communications infrastructure is the same as described in [16].

2–3.1.2 Software architecture

This section is not yet written.

2–3.2 Behavior

The behavior of the NIRI/OCS interface is the same as described in [16].

2–3.3 Implementation

This section contains tables summarizing the EPICS records used for the interface. The next sections (Section 2–3.4 on page 16 and Section 2–3.5 on page 23) contain the details of what those records do.

It is assumed in all cases that the record names given in this document will be prefixed by the string “niri:” when the database is implemented. For the current (prototype) release, records in the SAD database are prefixed by “niri::”; this will change to “niri:” before the final release.

2–3.3.1 Command (CAD/CAR) records

The columns of the command record tables are as follows:

Description. A brief description of the command.

Attribute. The name of the attribute which the Observatory Control System associates with a particular parameter of a particular command.

CAD record/field. The name of the EPICS CAD record (minus the "niri:" prefix[1]) together with the field in that CAD record associated with each parameter of the command.

CAD ordering number. The recommended order in which this command should be executed. CAD records should be connected to their associated APPLY record in the order 1, 2, 3, 4, 5, etc... CAD records within the same ordering number set can be connected to the APPLY record in any order in that set.

CAR record. The CAR record associated with the command. Note that several CAD records may share the same CAR record.

2–3.3.1.1 Sequence command CAD/CAR records

The NIRI sequencer recognizes all the OCS sequence commands and uses the CAD/CAR records specified in ICD 1b [25]. See also ICD 1a [24], for a description of what these commands do.

TABLE 1. NIRI sequence command CAD/CAR records

Description	Attribute	CAD ("niri:" +)	CAR record ("niri:" +)
Test.	(none)[a]	test	testC[b]
Initialize.	(none)[a]	init	initC
Datum mechanisms.	(none)[a]	datum	datumC
Verify	(none)[a]	verify	verifyC
End verify	(none)[a]	endVerify	endVerifyC
Guide	(none)[a]	guide	guideC
End guide	(none)[a]	endGuide	endGuideC
Observe	(none)[a]	observe	observeC
Pause	(none)[a]	pause	
Continue	(none)[a]	continue	
Stop	(none)[a]	stop	
Abort	(none)[a]	abort	
End Observe	(none)[a]	endObserve	endObserveC
Park	(none)[a]	park	parkC
Diagnose	(none)[a]	diagnose	applyC[c]

a. No arguments.

b. Currently, it is required to have separate TestC, InitC, etc. CAR records. There some disagreement over this requirement, and all of these may CAR records may disappear, and will be replaced by the global applyC.

c. This is not a standard record, and does not need it's own diagnoseC CAR record.

1. Note that the Detector Controller commands have a "niri:dc:" prefix; the Components Controller commands have a "niri:cc:" prefix; and the environment commands have a "niri:env:" prefix.

2–3.3.1.2 General observing command CAD/CAR records

TABLE 2. NIRI general observing command CAD/CAR records

Description	Attribute	CAD record:field ("niri:" +)	CAD ordering number	CAR record ("niri:" +)
Select beam-splitter position.	Position name	cc:spltSel.A	6	cc:spltC
Park beam-splitter.	(none)[a]	cc:spltPark	14	
Select filter in wheel 1.	Position name	cc:filt1Sel.A	6	cc:filt1C
Park filter wheel 1.	(none)[a]	cc:filt1Park	14	
Select filter in wheel 2.	Position name	cc:filt2Sel.A	6	cc:filt2C
Park filter wheel 2.	(none)[a]	cc:filt2Park	14	
Select pupil mask or filter in pupil-mask/filter wheel.	Position name	cc:filt3Sel.A	6	cc:filt3C
Park pupil-mask/filter wheel.	(none)[a]	cc:filt3Park	14	
Select focal-plane mask.	Position name	cc:foplSel.A	6	cc:foplC
Park focal-plane mask wheel.	(none)[a]	cc:foplPark	14	
Select beam-steerer positions	Position name (1) Position name (2)	cc:sterSel.A cc:sterSel.B	6	cc:sterC
Park beam-steerers.	(none)[a]	cc:sterPark	14	
Select pupil-viewer position	Position name	cc:puvwSel.A	6	cc:puvwSelC
Park pupil-viewer.	(none)[a]	cc:puvwPark	14	
Set focus.	Focus position	cc:focMove.A	6	cc:focC
Park focus.	(none)[a]	cc:focPark	14	
Open/Close window cover	State [OPEN | CLOSED]	cc:covSel.A	6	cc:covC
Park window cover.[b]	(none)[a]	cc:covPark	14	

a. No arguments.

b. Equivalent to closing the window cover.

2–3.3.1.3 Engineering command CAD/CAR records—System Commands

The following commands are used for engineering purposes, and are not expected to be used during normal observing. These are normally activated by the engineering user interface.

These commands affect the overall behavior of NIRI and the NIRI OIWFS. Note, in particular, that it is not possible to reboot NIRI and the NIRI OIWFS separately

TABLE 3. NIRI system engineering command CAD/CAR records

Description	Attribute	CAD record/field ("niri:" +)	CAD ordering number	CAR record
Reboot.	(none)[a]	reboot	N/A	N/A
Select debug mode.	Debug mode	debug.A	3	debugC
Flush data from internal buffers.	(none)[a]	flush	13	flushC

a. No arguments.

2–3.3.1.4 CC Sequence command CAD/CAR records

The NIRI components controller control software recognizes all the OCS sequence commands and uses the CAD/CAR records specified in ICD 1b, [25]. See also ICD 1a, [24], for a description of what these commands do.

TABLE 4. NIRI CC sequence command CAD/CAR records

Description	Attribute	CAD record.FIELD ("niri:" +)	CAR record ("niri:" +)
Test.	(none)	cc:test	cc:testC
Initialize.	(none)	cc:init	cc:initC
Datum mechanisms.	(none)	cc:datum	cc:datumC
Verify	(none)	cc:verify	cc:verifyC
End verify (no action)	(none)	cc:endVerify	cc:endVerifyC
Guide	(none)	cc:guide	cc:guideC
End guide (no effect)	(none)	cc:endGuide	cc:endGuideC
Park	(none)	cc:park	cc:parkC

2–3.3.1.5 Engineering command NIRI CC CAD/CAR records

The following commands are used for engineering purposes, and are not expected to be used during normal observing. Note that, although these commands are connected to the APPLY record, they are normally activated directly by the engineering user interface.

The following commands affect the overall behavior of NIRI.

TABLE 5. NIRI CC engineering command CAD/CAR records

Description	Attribute	CAD record/field ("niri:" +)	CAD ordering number	CAR record ("niri:" +)
Datum filter wheel 1.	(none)[a]	cc:filt1Datm	3	cc:filt1C
Set filter wheel 1 position in engineering units.	Filter wheel 1 position	cc:filt1MoveEng.A	6	
Enable/disable Hall-effect sensors for filter wheel 1.	Use sensor P[b] (bool[c]) Use sensor B[d] (bool)	cc:filt1Hall.A cc:filt1Hall.B	1	
Confirm that filter wheel 1 has not lost its datum position.	(none)[a]	cc:filt1Rdtm	6	
Confirm that filter wheel 1 and its Hall-effect sensors are operational.	(none)[a]	cc:filt1Diag	6	
Datum filter wheel 2.	(none)[a]	cc:filt2Datm	3	cc:filt2C
Set filter wheel 2 position in engineering units.	Position	cc:filt2MoveEng.A	6	
Enable/disable Hall-effect sensors for filter wheel 2.	Use sensor P (bool) Use sensor B (bool)	cc:filt2Hall.A cc:filt2Hall.B	1	
Confirm that filter wheel 1 has not lost its datum position.	(none)[a]	cc:filt2Rdtm	6	
Confirm that filter wheel 2 and its Hall-effect sensors are operational	(none)[a]	cc:filt2Diag	6	
Datum pupil-mask/filter wheel.	(none)[a]	cc:filt3Datm	3	cc:filt3C
Set pupil-mask/filter wheel position in engineering units.	Position	cc:filt3MoveEng.A	6	
Enable/disable Hall-effect sensors for pupil-mask/filter wheel.	Use sensor P (bool) Use sensor B (bool)	cc:filt3Hall.A cc:filt3Hall.B	1	
Confirm that pupil-mask/filter wheel has not lost its datum position.	(none)[a]	cc:filt3Rdtm	6	
Confirm that pupil-mask/filter wheel and its Hall-effect sensors are operational	(none)[a]	cc:filt3Diag	6	
Datum focal-plane mask wheel.	(none)[a]	cc:foplDatm	3	cc:foplC
Set focal-plane mask wheel position in engineering units.	Focal-plane mask wheel position	cc:foplMoveEng.A	6	
Enable/disable Hall-effect sensors for focal-plane mask wheel.	Use sensor P (bool) Use sensor B (bool)	cc:foplHall.A cc:foplHall.B	1	
Confirm that focal-plane mask wheel has not lost its datum position.	(none)[a]	cc:foplRdtm	6	
Confirm that focal-plane mask wheel and its Hall-effect sensors are operational	(none)[a]	cc:foplDiag	6	

TABLE 5. NIRI CC engineering command CAD/CAR records

Description	Attribute	CAD record/field ("niri:" +)	CAD ordering number	CAR record ("niri:" +)
Datum beam steerers.	(none)[a]	cc:sterDatm	3	cc:sterC
Set beam steerer 1 position in engineering units.	Motor pos. (long)	cc:ster1MoveEng.A	6	
Enable/disable Hall-effect sensors for beam steerer 1.	Use sensor P (bool) Use sensor B (bool)	cc:ster1Hall.A cc:ster1Hall.B	1	
Confirm that beam steerer 1 has not lost its datum position.	(none)[a]	cc:ster1Rdtm	6	
Confirm that beam steerer 1 and its Hall-effect sensors are operational	(none)[a]	cc:ster1Diag	6	
Set beam-steerer 2 position in engineering units.	Motor pos. (long)	cc:ster2MoveEng.A	6	
Enable/disable Hall-effect sensors for beam steerer 2.	Use sensor P (bool) Use sensor B (bool)	cc:ster2Hall.A cc:ster2Hall.B	1	
Confirm that beam steerer 2 has not lost its datum position.	(none)[a]	cc:ster2Rdtm	6	
Confirm that beam steerer 2 and its Hall-effect sensors are operational	(none)[a]	cc:ster2Diag	6	
Datum pupil-viewer mechanism	(none)[a]	cc:puvwDatm	3	cc:puvwC
Set pupil-viewer position in engineering units.	Position (long)	cc:puvwMoveEng.A	6	
Enable/disable Hall-effect sensors for pupil viewer.	Use sensor P (bool) Use sensor B (bool)	cc:puvwHall.A cc:puvwHall.B	1	
Confirm that pupil viewer has not lost its datum position.	(none)[a]	cc:puvwRdtm	6	
Confirm that pupil viewer and its Hall-effect sensors are operational	(none)[a]	cc:puvwDiag	6	
Datum focus mechanism	(none)[a]	cc:focDatm	3	cc:focC
Set focus translation mechanism in engineering units.	Position (long)	cc:focMoveEng.A	6	
Enable/disable Hall-effect sensors for focus mechanism.	Use sensor 1P (bool) Use sensor 1B (bool) Use sensor 2P (bool) Use sensor 2B (bool)	cc:focHall.A cc:focHall.B cc:focHall.C cc:focHall.D	1	
Confirm that focus stage has not lost its datum position.	(none)[a]	cc:focRdtm	6	
Confirm that focus stage and its Hall-effect sensors are operational	(none)[a]	cc:focDiag	6	

TABLE 5. NIRI CC engineering command CAD/CAR records

Description	Attribute	CAD record/field ("niri:" +)	CAD ordering number	CAR record ("niri:" +)
Datum window cover.	(none)[a]	cc:covDatm	3	cc:covC
Set window cover position in engineering units.	Position (long)	cc:covMoveEng.A	6	
Enable/disable Hall-effect sensors for window cover.	Use sensor P (bool) Use sensor B (bool)	cc:covHall.A cc:covHall.B	1	
Confirm that window cover has not lost its datum position.	(none)[a]	cc:covRdtm	6	
Confirm that window cover and its Hall-effect sensors are operational	(none)[a]	cc:covDiag	6	

a. No arguments.

b. Primary

c. Strictly speaking, there is no boolean data type. "(bool)" is used to indicate that the range of possible values is either a zero (false) or a one (true).

d. Backup

TABLE 6. NIRI environment command CAD/CAR records

Description	Attribute	CAD record/field ("niri:" +)	CAD ordering number	CAR record ("niri:" +)
Set temperature controllers set point[a].	Set point 1 (double) Set point 2 (double)	env:tmpc1Set.A env:tmpc2Set.B	1[b]	tmpcC
Set speed of cooling motors.	Motor speed (double)	env:coolSet.A	1	env:coolC
Cool down / warm up / accelerated warm up[c] (TBD)	TBD	env:???	1	env:???C

a. It may not be necessary to be able to set the temperature controllers to different values.

b. Command may be issued whenever the system is otherwise idle.

c. Commands will be necessary here to begin the cool-down and warm-up cycles (and accelerated warm-up, if implemented). The format of these commands is TBD.

2–3.3.2 Status Information Records

The following tables show the status information provided by NIRI, together with the subset of that information which is written to the FITS header. The tables distinguish status information provided by the Instrument Sequencer, Detector Controller and Components Controller, but all the information shown is available over the OCS to NIRI interface. The tables have the following columns:

SIR record. The part of the name of the EPICS SIR record which follows the "niri:" prefix[2].

Units. The units in which the quantity is stored

Comments. A description of the item. The first paragraph of this will go into the SIR record description and also the comment field in the FITS header.

Note that only the type, units and comments are actually contained in the SIR records. The FITS keyword is provided by the DHS.

TABLE 7. NIRI Instrument Sequencer (IS) status Information

SIR record ("niri:" +)	FITS keyword	FITS included?	Type	Units	Comments
name	INSTRUME*	Start	string		Instrument name ("NIRI").
state	INSTATE •		string		Instrument state [BOOTING | INITIALISING | RUNNING | CONFIGURING]
health	INHEALTH •		string		Instrument health [GOOD | WARNING | BAD].
issPort	INPORT •		integer		Number of ISS port on which the instrument is installed.
historyLog		TBD	char array		Instrument log message.
heartBeat	N/A	Never	integer		Continuously changing variable used to detect whether the system is still alive.

TABLE 8. NIRI Components Controller (CC) status Information

SIR record ("niri:" +)	FITS keyword	FITS included?	Type	Units	Comments
cc:name	CCNAME		string		Components Controller name ("NIRICC").
cc:state	CCSTATE		string		Components Controller state [BOOTING | INITIALISING | RUNNING | CONFIGURING].
cc:health	CCHEALTH		string		Components Controller health [GOOD | WARNING | BAD].
cc:historyLog		TBD	char array		Components Controller log message.
cc:heartBeat	N/A	Never	integer		Continuously changing variable used to detect whether the system is still alive.
cc:wfsBeam	WFSBEAM		boolean		Flag to indicate that the NIRI is allowing light to fall on its on-instrument wavefront sensor.

2. Note that the actual system has been designed so the Status Alarm Database can easily be given a different prefix from the rest of the NIRI database (for example "niri:"). It is assumed in this document that the Status Alarm Database prefix will be "niri:".

TABLE 8. NIRI Components Controller (CC) status Information

SIR record ("niri:" +)	FITS keyword	FITS included?	Type	Units	Comments
cc:power			boolean		Master power on for steppermotor modules?
cc:filt1Name	CCFLT1 •	Start	string		Filter 1 name.
cc:filt1Eng	CCFLT1EN		integer	engineering units	Filter wheel 1 engineering position.
cc:filt1Parked	CCFLT1PK		boolean		Is filter wheel 1 parked?
cc:filt1Datumed	CCFLT1DT		boolean		Is filter wheel datumed?
cc:filt1Health	CCFLT1HT		string		Health of filter wheel 1.
cc:filt1Motor	CCFLT1MH		string		Health of filter wheel 1 motor.
cc:filt1State	CCFLT1ST		string		State of filter wheel 1.
cc:filt2Name	CCFLT2 •	Start	string		Filter 2 name.
cc:filt2Eng	CCFLT2EN		integer	engineering units	Filter wheel 2 engineering position.
cc:filt2Parked	CCFLT2PK		boolean		Is filter wheel 2 parked?
cc:filt2Datumed	CCFLT2DT		boolean		Is filter wheel datumed?
cc:filt2Health	CCFLT2HT		string		Health of pupil-mask/filter wheel.
cc:filt2Motor	CCFLT2MH		string		Health of filter wheel 2 motor.
cc:filt2State	CCFLT2ST		string		State of pupil-mask/filter wheel.
cc:filt3Name	FILTER3 •	Start	string		Pupil-mask/filter name.
cc:filt3Eng	CCFLT3EN		integer	engineering units	Pupil-mask/filter wheel engineering position.
cc:filt3Parked	CCFLT3PK		boolean		Is pupil-mask/filter wheel parked?
cc:filt3Datumed	CCFLT3DT		boolean		Is pupil-mask/filter wheel datumed?
cc:filt3Health	CCFLT3HT		string		Health of pupil-mask/filter wheel.
cc:filt3State	CCFLT3ST		string		State of pupil-mask/filter wheel.
cc:filt3Motor	CCFLT3MH		string		Health of pupil-mask/filter wheel motor.
cc:ster1Name	CCSTR1•	Start	string		Beam steerer 1 position name.
cc:ster1Eng	CCSTR1FE		integer	engineering units	Set beam steerer 1 engineering position.
cc:ster1Parked	CCSTR1PK		boolean		Is beam steerer 1 parked?
cc:ster1Datumed	CCSTR1DT		boolean		Is beam steerer 1 datumed?
cc:ster1Health	CCSTR1HT		string		Health of beam steerer 1.
cc:ster1Motor	CCSTR1MH		string		Health of beam steerer 1 motor.
cc:ster1State	CCSTR2ST		string		State of beam steerer 1.
cc:ster2Name	CCSTR2•	Start	string		Beam steerer 2 position name.

TABLE 8. NIRI Components Controller (CC) status Information

SIR record (“niri:” +)	FITS keyword	FITS included?	Type	Units	Comments
cc:ster2Eng	CCSTR2EN		integer	engineering units	Beam steerer 2 engineering position.
cc:ster2Parked	CCSTR2PK		boolean		Is beam steerer 2 parked?
cc:ster2Datumed	CCSTR2DT		boolean		Is beam steerer 2 datumed?
cc:ster2Health	CCSTR2HT		string		Health of beam steerer 2.
cc:ster2Motor	CCSTR1MH		string		Health of beam steerer 2 motor.
cc:ster2State	CCSTR2ST		string		State of beam steerer 2.
cc:foplName	CCFPM•	Start	string		Focal-plane mask wheel name.
cc:foplEng	CCFPMEN		integer	engineering units	Focal-plane mask wheel engineering position.
cc:foplParked	CCFPMPK		boolean		Is focal-plane mask wheel parked?
cc:foplDatumed	CCFPMDT		boolean		Is focal-plane mask wheel datumed?
cc:foplHealth	CCFPMHT		string		Health of focal-plane mask wheel.
cc:foplMotor	CCFPMMH		string		Health of focal-plane mask wheel motor.
cc:foplState	CCFPMST		string		State of focal-plane mask wheel.
cc:covEng	CCCOVEN		integer	engineering units	Window cover engineering position.
cc:covParked	CCCOVPK		boolean		Is window cover parked (closed)?
cc:covHealth	CCCOVHT		string		Health of window cover.
cc:covMotor	CCCOVMH		string		Health of window cover motor.
cc:covState	CCCOVST		string		State of window cover.
cc:puvwName	CCPV•	Start	string		Pupil viewer position name.
cc:puvwEng	CCPVEN		integer	engineering units	Pupil viewer engineering position.
cc:puvwParked	CCPVPK		boolean		Is pupil viewer parked?
cc:puvwHealth	CCPVHT		string		Health of pupil viewer.
cc:puvwMotor	CCPVMH		string		Health of pupil viewer motor.
cc:puvwState	CCPVST		string		State of pupil viewer.
cc:focName	CCFOC•	Start	string		Focus position name.
cc:focParked	CCFOCPK		boolean		Is focus stage parked?
cc:focDatumed	CCFOCDT		boolean		Is focus stage datumed?
cc:focHealth	CCFOCHT		string		Health of focus stage.
cc:focMotor	CCFOCMH		string		Health of focus stage motor.
cc:focState	CCFOCST		string		State of focus stage.

TABLE 8. NIRI Components Controller (CC) status Information

SIR record ("niri:" +)	FITS keyword	FITS included?	Type	Units	Comments
cc:focEng	CCFOCEN		integer	engineering units	Focus stage engineering position.
cc:focPos	CCFOCPOS		float	mm	Focus stage position (millimeters)

TABLE 9. NIRI Environment Information

SIR record ("niri:" +)	FITS keyword	FITS included?	Type	Units	Comments
env:health	CEHEALTH		string		Components Controller environment health [GOOD | WARNING | BAD].
env:tcsp1Set	CETPSET1		float	K	Temperature controller 1 set point.
env:tct1Temp	CETPVAL1		float	K	Temperature controller 1 temperature
env:tcsp2Set	CETPSET2		float	K	Temperature controller 2 set point.
env:tct2Temp	CETPVAL2		float	K	Temperature controller 2 temperature
env:dewarTempK1	CETPMON1		float	K	Dewar temperature, sensor 1.
env:dewarTempR1	CETPRAW1		float	V	Dewar temperature, sensor 1, raw voltage.
env:dewarTempK2	CETPMON2		float	K	Dewar temperature, sensor 2.
env:dewarTempR2	CETPRAW2		float	V	Dewar temperature, sensor 2, raw voltage.
. . .	. . .	. . .	. . .	. . .	. . .
env:dewarTempKn[a]	CETPMONn		float	K	Dewar temperature, sensor n.
env:dewarTempRn	CETPRAWn		float	V	Dewar temperature, sensor n, raw voltage.

a. n = total number of temperature sensors, currently TBD.

2–3.4 Detailed Command Description

A general table of the commands accepted by the NIRI control software, together with their arguments, is given in Table 2 on page 8. This section contains a detailed description of each of the commands that will be sent between the OCS and the NIRI control software. It is based on the required behavior of the commands specified in [21] and [32].

Any directive asserted on the top-level APPLY record will be forwarded to all the CAD records in the sequencer, CC, and DC (perhaps via lower-level APPLY records). Only the marked CAD records will respond. If the directive is accepted it will cause the NIRI control software to change its configuration. The "applyC" CAR record will go "BUSY" while the

NIRI control software is changing its configuration and go to "IDLE" if completed successfully or to "ERR" if the configuration change fails.

2–3.4.1 Sequence Commands

2–3.4.1.1 TEST

On receipt of this command, the NIRI IS will insure that an observation is not taking place, and then will forward this command to the DC and CC. The "testC" CAR record will go "BUSY" while the tests are being carried out and go to "IDLE" if completed successfully or to "ERR" if the tests fail.

The CC accepts, but ignores this command.

The behavior of the DC is documented in [18].

Because the TEST command is not permitted to move any mechanisms, it is unable to detect any problems with the components controller. Use the DIAGNOSE and REDATUM commands instead.

2–3.4.1.2 INIT

On receipt of this command, the NIRI IS will insure that an observation is not taking place. It will then execute its initialization sequence, in which it will: Read its hardware setup files and lookup tables, and reset itself to the start-up state. This command will be forwarded to the CC and the DC. The "initC" CAR record will go "BUSY" while the initialization is being carried out and go to "IDLE" if completed successfully or to "ERR" if the initialization fails.

The CC will read its hardware setup files and lookup tables, and reset itself to the start-up state. No mechanisms will be moved.

The behavior of the DC is documented in [18].

2–3.4.1.3 DATUM

On receipt of this command, the NIRI IS will insure that an observation is not taking place and then forward the command to the CC and DC. The "datumC" CAR record will go "BUSY" while the datum search is being carried out and go to "IDLE" if completed successfully or to "ERR" if the datum searched fails.

The CC will locate the datum (i.e. index, home or reference point) for all its mechanisms.

The behavior of the DC is documented in [18].

2–3.4.1.4 VERIFY

On receipt of this command, the NIRI IS will insure that an observation is not taking place and then forward the command to the CC and DC. The "verifyC" CAR record will be changed to "BUSY". The command will complete, and the "verifyC" CAR changed back to "IDLE" when the state of the NIRI control software can be changed safely.

The CC accepts, but ignores this command.

The behavior of the DC is documented in [18].

2–3.4.1.5 ENDVERIFY

On receipt of this command, the NIRI IS will insure that an observation is not taking place and then forward the command to the CC and DC. The command will change the "endVerifyC" CAR status to "BUSY" and then back to "IDLE".

The CC accepts but ignores this command.

The behavior of the DC is documented in [18].

2–3.4.1.6 GUIDE

On receipt of this command, the NIRI IS will insure that an observation is not taking place. It will then set a flag as a reminder that guiding is taking place (as this information may affect its interaction with any on-instrument wavefront sensor). The sequencer will also forward this command on to the CC and DC. The "guideC" CAR record will be changed to "BUSY" and will change to "IDLE" when all the mechanisms are tracking within specification.

The CC accepts but ignores this command.

The behavior of the DC is documented in [18].

2–3.4.1.7 ENDGUIDE

On receipt of this command, the NIRI IS will insure that an observation is not taking place. It will then reset the guiding flag to indicate guiding has stopped. The sequencer will then forward this command on to the CC and DC. The "endGuideC" CAR record will be changed to "BUSY" and then back to "IDLE".

The CC accepts but ignores this command.

The behavior of the DC is documented in [18].

2–3.4.1.8 OBSERVE

On receipt of this command, the NIRI IS will insure that an observation is not taking place and then forward the command to the DC. The IS will reject the command if its control software or the CC control software is still configuring its mechanisms. If accepted, the "observeC" CAR record will be changed to "BUSY". It will go back to "IDLE" when the data set observation completes (or "ERR" if the observation was not successful).

The CC will not respond to this command.

The behavior of the DC is documented in [18].

2–3.4.1.9 ENDOBSERVE

On receipt of this command, the NIRI IS will forward the command to the DC. No changes will be made to the "observeC" CAR record.

The CC will not respond to this command.

The behavior of the DC is documented in [18].

2–3.4.1.10 PAUSE

On receipt of this command, the NIRI IS will forward the command to the DC.

The CC will not respond to this command.

In the DC, this will cause the observeC CAR record to change to “PAUSED”. Further details of the behavior of the DC is documented in [18].

2–3.4.1.11 CONTINUE

On receipt of this command, the NIRI IS will forward the command to the DC. The command will be rejected if an observation has not been paused. If successful, the “observeC” CAR record will be changed to “BUSY”.

The CC will not respond to this command.

The behavior of the DC is documented in [18].

2–3.4.1.12 STOP

On receipt of this command, the NIRI IS will forward the command to the DC. The command will be rejected if an observation is not being made. If successful, the “observeC” CAR record will be changed to “IDLE”

The CC will not respond to this command.

The behavior of the DC is documented in [18].

2–3.4.1.13 ABORT

On receipt of this command, the NIRI IS will forward the command to the DC. If successful, the “observeC” CAR record will be changed to “IDLE”.

The CC will not respond to this command.

The behavior of the DC is documented in [18].

2–3.4.1.14 PARK

On receipt of this command, the NIRI IS will insure that an observation is not taking place and then forward the command to the DC and the CC.

The CC will move its mechanisms to a configuration in which it can safely be switched off. It will utilize all the low-level CAD records dedicated to parking the individual mechanisms. The behavior of the DC is documented in [18]. The “parkC” CAR record will go “BUSY” while the mechanisms are being parked and go to “IDLE” if completed successfully or to “ERR” if any mechanism fails to park successfully.

The behavior of the DC is documented in [18].

2–3.4.1.15 DIAGNOSE

On receipt of this command, the NIRI IS will insure that an observation is not taking place and then forward the command to the CC.

The CC will perform self tests to ensure that its subsystems are healthy. The “applyC” CAR record will go “BUSY” while the tests are being carried out and go to “IDLE” if completed successfully or to “ERR” if any mechanism failed to diagnose successfully.

The DC will not respond to this command.

Note: unlike the TEST command, the DIAGNOSE command will move mechanisms, as necessary to diagnose problems.

2–3.4.2 General Engineering Commands

2–3.4.2.1 REBOOT

On receipt of this command, the NIRI control software will reboot its IOC and go through the same start-up procedure it would normally do when first switched on. It will reboot into simulation mode NONE and debug mode NONE. The prototype and simulator will reboot into simulation mode FULL and debug mode NONE. There is no CAR record associated with this command because connection with the IOC will be lost while it reboots.

This command will also reboot the OIWFS components controller.

2–3.4.2.2 DEBUG

On receipt of this command, the NIRI control software will change to the given debug mode (NONE, MIN or FULL). This mode determines the amount and frequency of information logged by the NIRI control software, as described in the “Debugging Modes” section of ICD 1a, [24]. The “debugC” CAR record will go briefly “BUSY” and then back to “IDLE”.

2–3.4.2.3 FLUSH

This command will be forwarded to the DC.

The CC does not recognize this command.

On receipt of this command, the DC control software will transmit whatever is contained within its data buffers to the DHS data storage server for storage to disk. The action can be used for recovering data after a software problem. The “flushC” CAR record will go “BUSY” while the data are being transmitted and switch to “IDLE” if the data transfer succeeded or “ERR” if it failed. The behavior of the DC is documented in [18].

2–3.4.2.4 (Temperature controller commands)

TBD

NOTE: The design of these NIRI CC commands follows the CICS control software prototype, which is not finalized. They may change as the CICS is implemented.

2–3.4.3 Wheel and Wheel-like Mechanisms

NIRI will have two filter wheels, a pupil-mask/filter wheel, a focal plane mask, two beam steerers, a pupil viewer, and a window cover. The following commands will act on all of the above mentioned mechanisms:

Each component can be set individually with the following commands:

2–3.4.3.1 Select position

On receipt of this command, the NIRI CC control software will:

```
Ensure an observation is not taking place.
Turn on the master power supply (if necessary).
Unlock the component. (Only the Beam-Splitter Wheel has a locking
   mechanism.)
Translate desired position name into an engineering position with the
   aid of a look-up table.
Look up the expected value for the Hall-effect sensors at the desired
   position location.
Move the mechanism to the engineering position corresponding to the
   desired position.
Lock the component (Only the Beam-Splitter Wheel has a locking
   mechanism.).
Turn off the master power supply (if last moving component).
Verify that the expected sensor value matches the measured value.
```

2–3.4.3.2 Park mechanism

On receipt of this command, the NIRI CC control software will:

```
Ensure an observation is not taking place.
Turn on the master power supply (if necessary).
Unlock the component. (Only the Beam-Splitter Wheel has a locking
   mechanism.)
Move the mechanism to its parking position.
Lock the component (Only the Beam-Splitter Wheel has a locking
   mechanism.).
Turn off the master power supply (if last moving component).
Verify that the expected sensor value matches the measured value.
```

2–3.4.3.3 Datum mechanism.

On receipt of this command, the NIRI CC control software will:

```
Ensure an observation is not taking place.
Turn on the master power supply (if necessary).
Unlock the component. (Only the Beam-Splitter Wheel has a locking
   mechanism.)
Move the mechanism and search for its datum.
Set the hardware datum position.
Lock the component (Only the Beam-Splitter Wheel has a locking
   mechanism.).
Turn off the master power supply (if last moving component).
```

2–3.4.3.4 Set filter wheel position in engineering units.

On receipt of this command, the NIRI CC control software will move the mechanism to the specified engineering position. The Hall-effect sensors will not be used for confirmation.

2–3.4.3.5 Redatum mechanism.

On receipt of this command, the NIRI CC control software will:

```
Ensure an observation is not taking place.
Turn on the master power supply (if necessary).
Unlock the component. (Only the Beam-Splitter Wheel has a locking
   mechanism.)
Move the mechanism and search for its datum.
Lock the component (Only the Beam-Splitter Wheel has a locking
   mechanism.).
Turn off the master power supply (if last moving component).
```

The hardware datum position will be left unchanged. If the datum position has changed significantly since the system was datumed, a warning will be generated.

2–3.4.3.6 Diagnose mechanism

On receipt of this command, NIRI will:

```
Ensure that an observation is not taking place.
Turn on the master power supply (if necessary).
Unlock the component. (Only the Beam-Splitter Wheel has a locking
   mechanism.)
Move the mechanism and attempt to verify that the motors and Hall-
   effect sensors are operating.
Lock the component (Only the Beam-Splitter Wheel has a locking
   mechanism.).
Turn off the master power supply (if last moving component).
```

2–3.4.4 Focus and offset commands

NIRI will have a focus stage. The following command will act on the above mentioned mechanism:

2–3.4.4.1 Select position

On receipt of this command, the NIRI CC control software will:

```
Ensure an observation is not taking place.
Turn on the master power supply (if necessary).
Move the focus mechanism to the engineering position corresponding to
   the desired focus.
Compute the expected values for the Hall-effect sensors.
Turn off the master power supply (if last moving component).
Verify that the final hall effect sensors match the expected values.
```

2–3.4.4.2 Park mechanism

On receipt of this command, the NIRI CC and control software will:

```
Ensure an observation is not taking place.
Turn on the master power supply (if necessary).
Move the focus mechanism to its parking position.
Turn off the master power supply (if last moving component).
```

2–3.4.4.3 Datum focus

On receipt of this command, the NIRI CC and control software will:

```
Ensure an observation is not taking place.
Turn on the master power supply (if necessary).
Move the mechanism and search for its coarse datum.
Move the mechanism and search for its secondary datum (if necessary).
Move the mechanism and search for its datum.
Record the new datum position and use this to update the position to
   engineering position look-up table for the mechanism.
Turn off the master power supply (if last moving component).
```

2–3.4.4.4 Redatum focus.

On receipt of this command, the NIRI CC and control software will:

```
Ensure an observation is not taking place.
Turn on the master power supply (if necessary).
Move the mechanism and search for its coarse datum.
Move the mechanism and search for its secondary datum (if necessary).
Move the mechanism and search for its datum.
Turn off the master power supply (if last moving component).
```

The hardware datum position will be left unchanged. If the datum position has changed significantly since the system was datumed, a warning will be generated.

2–3.4.4.5 Diagnose filter wheel

On receipt of this command, the NIRI CC and control software will:

```
Ensure that an observation is not taking place.
Turn on the master power supply (if necessary).
Move the mechanisms and attempt to verify that the motors and Hall-
   effect sensors are operating.
Turn off the master power supply (if last moving component).
```

2–3.5 Detailed Status information

Details of these records are still TBD.

2–3.5.1 General records

2–3.5.1.1 name

This record contains a string constant describing the name of the instrument. For NIRI this record will contain "NIRI".

2–3.5.1.2 state

This record will take the values BOOTING, INITIALISING, RUNNING and CONFIGURING[3] and reflect the state of the NIRI control software while it is initializing. The activities corresponding to these states are

- BOOTING — the state during IOC Init and while any database default values are being set.
- INITIALISING — the NIRI control software will be checking its own hardware and reading its hardware configuration files during this phase.
- RUNNING — the NIRI control software is ready to accept commands.
- CONFIGURING — the NIRI control software is ready to accept commands but it is busy configuring and unable to make observations.

2–3.5.1.3 health

This record will reflect the overall health of the NIRI control software (including the components controller and the detector controller). Expected values are GOOD, WARNING or BAD. If the overall health is BAD then the NIRI control software will be unusable. If the

3. Currently unimplemented

Health is set to WARNING then the NIRI control software will be able to execute some functions but not perhaps to specification.

2–3.5.1.4 historyLog[4]
The record used to contain messages to be recorded in the history log.

2–3.5.2 Wheel records

These are TBD.

2–3.5.3 Focus/Offset records

These are TBD.

2–3.5.4 Environment Records

These are TBD.

2–3.6 Alarm conditions

The following situations will generate an alarm or warning:

- Hall sensor disabled
- Motor-control module fault
- Diagnose failed
- Redatum failed
- Datum failed
- Move-confirmation failed
- Temperature changing
- Large temperature gradient
- Limit sensor reached
- Motion out of valid range

2–3.7 Debugging and Maintenance

This interface can be tested and debugged by means of an engineering user interface supplied as part of the NIRI.

2–3.7.1 Debugging Modes

The NIRI control software will have the standard debugging modes described in ICD 1a, [24].

2–3.7.2 Calibration

If a hall sensor fails or if a mechanism must be disassembled for maintenance, the sensors will need to be recalibrated. A separate program will be provided for this purpose. Calibration procedures will be described in a separate document [5].

4. Currently unimplemented

2–3.8 Simulation

The Gemini simulation modes are described in ICD 1a, [24]. The NIRI control software will use the FAST and FULL modes, and by default it begins in “NONE” mode. VSM mode may be supported, if time permits.

2–4.0 Safety Issues

It will be necessary to avoid moving NIRI components while the mechanical components are not at thermal equilibrium. Records will be provided to provide a temperature interlock. Details are described in [6].

ICD 1.6/1.10[ab]

"A&G to NIRI/GNIRS On Instrument Wavefront Sensors"

niri_hty_008

Hubert Yamada
1997-06-19

Version: 0.0

Issued By: Hubert Yamada

		Date:
Approved By: ____________________	Group Manager	__________
____________________	Group Manager (if req)	__________
____________________	Systems Engineering	__________

Revision Control

1. Revision Version #		0.0
	Date:	1997-06-19
	Revised by:	Hubert Yamada
	Reason for / items changed:	Original prerelease version

Last Modified: Thursday, June 19, 1997 5:43 pm

Chapter 1 Introduction

1–1.0 Description

1–1.1 Purpose

This Interface Control Document (ICD) serves the following purposes:

- To describe the commands and parameters that will pass between the A&G system and the On-Instrument Wavefront Sensor (OIWFS) Components Controller (CC).
- To provide a prototype which may, with minimal modification, be used to describe the GNIRS OIWFS CC, and the OIWFS for any system which copies the NIRI OIWFS hardware and software.

  This document only describes the ways in which the NIRI OIWFS differs from the prototypical OIWFS which is described in ICD 1.6/1.10 [19] and in [20].

  At some time in the future, the NIRI/GNIRS OIWFS information will be merged with ICD 1.6/1.10.

In addition to the documents mentioned in the previous paragraphs, the specific interfaces described in this document are governed by the following parent Interface Control Documents, which describe the general properties of particular kinds of interface:

- ICD 1a [24], which describes how the OCS deals with commands and attributes and the behavior the OCS expects from the systems it commands.
- ICD 1b [25], which gives a general description of the way the Observatory Control System communicates with the other principal systems. It is also related to ICD 7a, [27], which describes the way an Instrument Control System communicates with its subsystems.
- ICD 2 [26], which describes the way in which an Instrument Control System makes information available to the outside world (and the OCS) by means of “Status Information Records”.
- ICD 7b [28], which describes the way that the Telescope Control System (TCS) interacts with its subsystems. (From a control standpoint, the OIWFS CC is a subsystem of the A&G, which is a subsystem of the TCS.)

The intended audience for this document is:

- The NIRI reviewers;
- The GNIRS reviewers;
- Any other groups which will be using the design of the NIRI/GNIRS OIWFS;
- The developers of the Gemini Acquisition and Guidance units;
- The developers of the GNIRS OIWFS components controller.

1–1.2 Scope

This document describes only aspects of the NIRI OIWFS software system which differ from, or are more specific than, the Gemini standard on-instrument wavefront sensor

(OIWFS), described in [19]. It is assumed that the reader is familiar with both of those documents.

This document does *not* describe the following topics:

- The NIRI OIWFS CC hardware and electronics (see [11] and [12], summarized in [3]);
- The NIRI OIWFS DC software, hardware, and electronics (see [12] and [13]);
- The GNIRS software, hardware, and electronics;
- Other instruments which will be using the NIRI/GNIRS OIWFS;
- Interfacing an instrument to the NIRI/GNIRS OIWFS (see [6]).

The software design is not described here. For that, see [2], and [3].

1–2.0 Related Documents and Drawings

1–2.1 References

See “Near IR Imager (NIRI)—Definitions, Abbreviations, and References”.

1–2.2 Definitions, Acronyms and Abbreviations

See “Near IR Imager (NIRI)—Definitions, Abbreviations, and References”.

1–2.3 Stylistic Conventions

References to documents in Section 1.5, “References,” on page 2 of “Near IR Imager (NIRI)—Definitions, Abbreviations, and References” are given like this; [1].

Chapter 2 Details of the A&G to NIRI/GNIRS OIWFS Interface

2–1.0 Introduction

2–2.0 Physical System Interfaces

2–2.1 Mechanical Interface

Not Applicable for a software ICD.

2–2.2 Optical Interface

Not Applicable for a software ICD.

2–2.3 Electronic Interface

2–2.3.1 NIRI

The system hardware architecture for the NIRI OIWFS control software is the same as described in [16], with the following exception:

- The Instrument Sequencer, Components Controller, and OIWFS Components Controller will share the same IOC.

There will also be the following electronic VME modules:

- The XVME-240 Digital I/O Board, (shared between the NIRI CC, the NIRI OIWFS, and the interlock bus):
  - To support the interlock bus (Gemini standard, see [29]);
  - To control the stepper-motor master power supply;
  - To monitor the stepper-motor control modules;
  - To reset the stepper-motor control modules;
  - To control the heating system.
- The serial ports on the VME-167 (via the MVME-712/M module) and a serial I/O board (board TBD, depending on available EPICS support.):
  - To control the NIRI/OIWFS temperature controllers (three ports);
  - To control the NIRI cooling motors (one port);
- Four custom-built VME sensor-support boards:
  - To read out the NIRI Hall-effect sensors;
  - To read out the NIRI temperature sensors;
  - To read out the NIRI OIWFS Hall-effect sensors (one board, shared);

- Two XVME-566 Analog Input Modules
  - To read out the VME sensor-support boards for the NIRI temperature sensors
  - To read out the VME sensor-support boards for the NIRI Hall-effect sensors
  - To read out the VME sensor-support boards for the NIRI OIWFS Hall-effect sensors (one board, shared)
- Three Oregon Microsystems VME8-8 Stepper-motor controller boards
  - To control the NIRI stepper-motors
  - To control the NIRI OIWFS stepper-motors (one board, shared)

2–2.3.2 GNIRS

TBD

2–2.4 Mass/Balance

Not Applicable for a software ICD.

2–2.5 Thermal Interface

Not Applicable for a software ICD.

2–3.0 Software/Control Function Interface

2–3.1 Overview

2–3.1.1 Communications infrastructure

The communications infrastructure is the same as described in [16].

2–3.1.2 Software architecture

This section is not yet written.

2–3.2 Behavior

The behavior of the NIRI/GNIRS OIWFS/A&G interface is the same as described in [20] and [19].

2–3.3 Implementation

This section contains tables summarizing the EPICS records used for the interface. The next sections (Section 2–3.4 on page 12 and Section 2–3.5 on page 16) contain the details of what those records do.

It is assumed in all cases that the record names given in this document will be prefixed by the string “niri:” or “nirs”, as appropriate when the database is implemented, except for the SAD database which will be prefixed by “niri:”[1] or “nirs:”.

2–3.3.1 Command (CAD/CAR) records

1. In the prototype, this is “niri::”. This will be changed to “niri:” before final release.

The columns of the command record tables are as follows:

Description. A brief description of the command.

Attribute. The name of the attribute which the Observatory Control System associates with a particular parameter of a particular command.

CAD record/field. The name of the EPICS CAD record (minus the "niri:" or "nirs:" prefix[2]) together with the field in that CAD record associated with each parameter of the command.

CAD ordering number. The recommended order in which this command should be executed. CAD records should be connected to their associated APPLY record in the order 1, 2, 3, 4, 5, etc... CAD records within the same ordering number set can be connected to the APPLY record in any order in that set.

CAR record. The CAR record associated with the command. Note that several CAD records may share the same CAR record.

2–3.3.1.1 Engineering command CAD/CAR records—System Commands

The following commands are used for engineering purposes, and are not expected to be used during normal observing. These are normally activated by the engineering user interface.

These commands affect the overall behavior of NIRI and the NIRI OIWFS. Note, in particular, that it is not possible to reboot NIRI and the NIRI OIWFS separately. It may be possible to reboot the GNIRS OIWFS without rebooting GNIRS.

TABLE 1. NIRI/GNIRS system engineering command CAD/CAR records

Description	Attribute	CAD record/field ("niri:" or "nirs:" +)	CAD ordering number	CAR record
Reboot.	(none)[a]	wfs:reboot	N/A	N/A
Select debug mode.	Debug mode	wfs:debug.A	3	wfs:debugC
Flush data from internal buffers.	(none)[a]	wfs:flush	13	wfs:flushC

a. No arguments.

2–3.3.1.2 Engineering command NIRI/GNIRS CC CAD/CAR records

The following commands are used for engineering purposes, and are not expected to be used during normal observing. Note that, although these commands are connected to the APPLY record, they are normally activated directly by the engineering user interface.

2–3.3.1.3 OIWFS command CAD/CAR records—OIWFS components

The following commands are used for engineering purposes, and are not expected to be used during normal observing. Note that, although these commands are connected to the OIWFS APPLY record, they are normally activated directly by the engineering user interface.

This section will eventually be moved into [19].

2. Note that the OIWFS controller commands all have a "niri:wfs:" prefix.

The following commands affect the overall behavior of the OIWFS but do not affect the science detector.

TABLE 2. NIRI OIWFS CC engineering command CAD/CAR records

Description	Attribute	CAD record/field ("niri:" or "nirs:" +)	CAR record ("niri:" or "nirs:" +)
Datum filter wheel.	(none)[a]	wfs:filtDatm	wfs:filtC
Set filter wheel position in engineering units[b].	Filter wheel 1 position	wfs:filtMoveEng.A	
Enable/disable Hall-effect sensors for filter wheel.	Use sensor (bool[c])[d]	wfs:filtHall.A wfs:filtHall.B	
Confirm that filter wheel has not lost its datum position.	(none)[a]	wfs:filtRdtm	
Confirm that filter wheel and its Hall-effect sensors are operational	(none)[a]	wfs:filtDiag	

TABLE 2. NIRI OIWFS CC engineering command CAD/CAR records

Description	Attribute	CAD record/field ("niri:" or "nirs:" +)	CAR record ("niri:" or "nirs:" +)
Datum the gimbal mirrors.	(none)[a]	wfs:prbDatm	wfs:prbC
Park the gimbal mirrors.	(none)[a]	wfs:prbPark	
Test the gimbal mirrors.	(none)[a]	wfs:prbTest	
Move gimbal x mirror, given (x, y) position in engineering units.	(none)[a]	wfs:prbxMoveEng	
Move gimbal y mirror, given (x, y) position in engineering units.	(none)[a]	wfs:prbyMoveEng	
Move gimbal mirrors, to given (x, y) position and leave it stationary. (x, y) is the position in the focal plane measured in mm in the instrument's frame of reference.	x y	wfs:prbMove.A wfs:prbMove.B	
Equivalent to wfs:prbxMoveEng and wfs:prbyMoveEng. For compatibility with ICD 1.6/1.10 [19].	x y	wfs:moveEng.A wfs:moveEng.B	
Equivalent to wfs:prbMove. For compatibility with ICD 1.6/1.10 [19].	(none)[a]	wfs:move	
The probe is caused to follow a continuous stream of updated (x, y) positions. These positions are note defined as CAD record arguments but in separate records (see ICD 1.6/1.10).	(none)[a]	wfs:follow	
Enable/disable Hall-effect sensors for x-gimbal mirror.[e]	Use sensor 1P (bool)[d] Use sensor 1B (bool) Use sensor 2P (bool) Use sensor 2B (bool)	wfs:prbxHall.A wfs:prbxHall.B wfs:prbxHall.C wfs:prbxHall.D	
Enable/disable Hall-effect sensors for y-gimbal mirror[e].	Use sensor 1P (bool)[d] Use sensor 1B (bool) Use sensor 2P (bool) Use sensor 2B (bool)	wfs:prbyHall.A wfs:prbyHall.B wfs:prbyHall.C wfs:prbyHall.D	
Confirm that x-gimbal mirror has not lost its datum position.	(none)[a]	wfs:prbxRdtm	
Confirm that x-gimbal mirror and its Hall-effect sensors are operational	(none)[a]	wfs:prbxDiag	
Confirm that y-gimbal mirrors has not lost its datum position.	(none)[a]	wfs:prbyRdtm	
Confirm that y-gimbal mirror and its Hall-effect sensors are operational	(none)[a]	wfs:prbyDiag	

TABLE 2. NIRI OIWFS CC engineering command CAD/CAR records

Description	Attribute	CAD record/field ("niri:" or "nirs:" +)	CAR record ("niri:" or "nirs:" +)
Datum focus mechanism.	(none)[a]	wfs:focDatm	wfs:focC
Set focus stage translation in engineering units.	Focus position	wfs:focMoveEng.A	
Enable/disable Hall-effect sensors for focus stage.	Use sensor (bool)[d]	wfs:focHall.A wfs:focHall.B wfs:focHall.C wfs:focHall.D	
Confirm that focus stage has not lost its datum position.	(none)[a]	wfs:focRdtm	
Test the focus stage.	(none)[a]	wfs:focTest	
Confirm that focus stage and its Hall-effect sensors are operational	(none)[a]	wfs:focDiag	

a. No arguments.

b. ICD 1.6/1.10 [19] has three separate components: a field-stop, a neutral-density filter, and a color filter. For the NIRI/GNIRS OIWFS CC, these have been combined into a single wheel.

c. Strictly speaking, there is no boolean data type. "(bool)" indicates that the range of acceptable values will be zero or one.

d. If the primary sensor is ignored, a slower datuming algorithm will be used. If both the primary and the secondary sensors are ignored, it will not be possible to datum the system at all. If either is ignored, a warning will be generated.

e. Currently unimplemented

2–3.3.2 Status Information Records

The following tables show the status information provided by the OIWFS. The tables have the following columns:

SIR record. The part of the name of the EPICS SIR record which follows the "niri:" or "nirs:" prefix[3].

Units. The units in which the quantity is stored

Comments. A description of the item. The first paragraph of this will go into the SIR record description and also the comment field in the FITS header.

Note that only the type, units and comments are actually contained in the SIR records. The FITS keyword is provided by the DHS.

3. Note that the actual system has been designed so the Status Alarm Database can easily be given a different prefix from the rest of the NIRI/GNIRS database(s) (for example "niri::" or "nirs::"). It is assumed in this document that the Status Alarm Database prefix will be "niri:" or "nirs:".

TABLE 3. NIRI OIWFS status Information

SIR record ("niri:" +)	FITS keyword	FITS included?	Type	Units	Comments
wfs:name	WSNAME		string		OIWFS Components Controller name ("NIRIWFS" or "NIRSWFS").
wfs:state	WSSTATE		string		OIWFS Components Controller state [BOOTING | INITIALISING | RUNNING | CONFIGURING].
wfs:health	WSHEALTH		string		OIWFS Components Controller health [GOOD | WARNING | BAD].
wfs:historyLog		TBD	char array		OIWFS Components Controller log message.
wfs:heartBeat	N/A	Never	integer		Continuously changing variable used to detect whether the system is still alive.
wfs:filtName	WSFLT •	Start	string		OIWFS Filter name.
wfs:filtEng	WSFLTEN		integer	engineering units	OIWFS Filter wheel engineering position.
wfs:filtParked	WSFLTPK		boolean		Is OIWFS filter wheel parked?
wfs:filtDatumed	WSFLTDT		boolean		Is OIWFS filter wheel datumed?
wfs:filtHealth	WSFLTHT		string		Health of OIWFS filter wheel.
wfs:filtState	WSFLTST		string		State of OIWFS filter wheel.
wfs:prbxPos	WSXPOS•	Start	double	arcsec	OIWFS x-gimbal mirror offset
wfs:prbxEng	WSXEN		integer	engineering units	OIWFS x-gimbal mirror engineering position.
wfs:prbxParked	WSXPK		integer	engineering units	Is OIWFS x-gimbal mirror parked?.
wfs:prbxDatumed	WSXDT		boolean		Is OIWFS x-gimbal mirror datumed?
wfs:prbxHealth	WSXHT		string		Health of OIWFS x-gimbal mirror.
wfs:prbxMotor	WSXMH		string		Health of OIWFS x-gimbal mirror motor.
wfs:prbxState	WSXST		string		State of OIWFS x-gimbal mirror.
wfs:prbyPos	WSYPOS		double	arcsec	OIWFS y-gimbal mirror offset in arcseconds
wfs:prbyEng	WSYEN		integer	engineering units	OIWFS y-gimbal mirror engineering position.
wfs:prbyParked	WSYPK		integer	engineering units	Is OIWFS y-gimbal mirror parked?.
wfs:prbyDatumed	WSYDT		boolean		Is OIWFS y-gimbal mirror datumed?

TABLE 3. NIRI OIWFS status Information

SIR record ("niri:" +)	FITS keyword	FITS included?	Type	Units	Comments
wfs:prbyHealth	WSYHT		string		Health of OIWFS y-gimbal mirror.
wfs:prbyMotor	WSYMH		string		Health of OIWFS y-gimbal mirror motor.
wfs:prbyState	WSYST		string		State of OIWFS y-gimbal mirror.
wfs:focParked	WSFOCPK		boolean		Is OIWFS focus stage parked?
wfs:focDatumed	WSFOCDT		boolean		Is OIWFS focus stage datumed?
wfs:focHealth	WSFOCHT		string		Health of OIWFS focus stage.
wfs:focMotor	WSFOCMH		string		Health of OIWFS focus stage motor.
wfs:focState	WSFOCST		string		State of OIWFS focus stage.
wfs:focPos	WSFOCP		double	mm	OIWFS Focus stage position.
wfs:focEng	WSFOCE		integer	engineering units	OIWFS Focus stage engineering position.

2–3.4 Detailed Command Description

A general table of the commands accepted by the NIRI/GNIRS OIWFS CC, together with their arguments, is given in Table 1 on page 7. This section contains a detailed description of each of the commands that will be sent between the A&G and the NIRI/GNIRS control software. It is based on the required behavior of the commands specified in [21] and [32].

If a directive is accepted it will cause the NIRI/GNIRS OIWFS CC to change its configuration. The "applyC" CAR record will go "BUSY" while the OIWFS control software is changing its configuration and go to "IDLE" if completed successfully or to "ERR" if the configuration change fails.

2–3.4.1 Sequence Commands

2–3.4.1.1 INIT

On receipt of this command, the OIWFS control software will execute its initialization sequence, in which it will: determine which simulation mode is required, read its hardware setup files and lookup tables, and reset itself to the start-up state. The "initC" CAR record will go "BUSY" while the initialization is being carried out and go to "IDLE" if completed successfully or to "ERR" if the initialization fails. No mechanisms will be moved.

2–3.4.1.2 DATUM

The OIWFS CC will locate the datum (i.e. index, home or reference point) for all its mechanisms. The "datumC" CAR record will go "BUSY" while the datum search is being carried out and go to "IDLE" if completed successfully or to "ERR" if the datum searched fails.

2–3.4.1.3 PARK

On receipt of this command, the OIWFS will move its mechanisms to a configuration in which it can safely be switched off. It will utilize all the low-level CAD records dedicated to parking the individual mechanisms. The "parkC" CAR record will go "BUSY" while the mechanisms are being parked and go to "IDLE" if completed successfully or to "ERR" if any mechanism fails to park successfully.

2–3.4.1.4 DIAGNOSE

On receipt of this command, the OIWFS CC will perform self tests to ensure that its subsystems are healthy. The "diagnoseC" CAR record will go "BUSY" while the tests are being carried out and go to "IDLE" if completed successfully or to "ERR" if the tests fail.

Note: unlike the TEST command, the DIAGNOSE command will move mechanisms, as necessary to diagnose problems.

2–3.4.2 General Engineering Commands

2–3.4.2.1 REBOOT

On receipt of this command, the OIWFS control software will reboot its IOC and go through the same start-up procedure it would normally do when first switched on. It will reboot into simulation mode NONE and debug mode NONE. The prototype and simulator will reboot into simulation mode FULL and debug mode NONE. There is no CAR record associated with this command because connection with the IOC will be lost while it reboots.

This command will also reboot the NIRI CC and IS. It may or may not reboot the GNIRS CC and IS (TBD).

2–3.4.2.2 DEBUG

On receipt of this command, the NIRI control software will change to the given debug mode (NONE, MIN or FULL). This mode determines the amount and frequency of information logged by the NIRI control software, as described in the "Debugging Modes" section of ICD 1a, [24]. The "debugC" CAR record will go briefly "BUSY" and then back to "IDLE".

2–3.4.3 Wheel and Wheel-like Mechanisms

The OIWFS will have a filter wheel. The following commands will act on the above mentioned mechanism:

Each component can be set individually with the following commands:

2–3.4.3.1 Select position

On receipt of this command, the OIWFS CC control software will:

```
Turn on the master power supply (if necessary).
Unlock the component. (Only the Beam-Splitter Wheel has a locking
   mechanism.)
Translate desired position name into an engineering position with the
   aid of a look-up table.
Look up the expected value for the Hall-effect sensors at the desired
   position location.
Move the mechanism to the engineering position corresponding to the
   desired position.
Lock the component (Only the Beam-Splitter Wheel has a locking
   mechanism.).
Turn off the master power supply (if last moving component).
Verify that the expected sensor value matches the measured value.
```

2–3.4.3.2 Park mechanism

On receipt of this command, the OIWFS CC control software will:

```
Turn on the master power supply (if necessary).
Unlock the component. (Only the Beam-Splitter Wheel has a locking
   mechanism.)
Move the mechanism to its parking position.
Lock the component (Only the Beam-Splitter Wheel has a locking
   mechanism.).
Turn off the master power supply (if last moving component).
Verify that the expected sensor value matches the measured value.
```

2–3.4.3.3 Datum mechanism.

On receipt of this command, the OIWFS CC control software will:

```
Turn on the master power supply (if necessary).
Unlock the component. (Only the Beam-Splitter Wheel has a locking
   mechanism.)
Move the mechanism and search for its datum.
Set the hardware datum position.
Lock the component (Only the Beam-Splitter Wheel has a locking
   mechanism.).
Turn off the master power supply (if last moving component).
```

2–3.4.3.4 Set filter wheel position in engineering units.

On receipt of this command, the OIWFS CC control software will move the mechanism to the specified engineering position. The Hall-effect sensors will not be used for confirmation.

2–3.4.3.5 Redatum mechanism.

On receipt of this command, the OIWFS CC control software will:

```
Turn on the master power supply (if necessary).
Unlock the component. (Only the Beam-Splitter Wheel has a locking
   mechanism.)
Move the mechanism and search for its datum.
Lock the component (Only the Beam-Splitter Wheel has a locking
   mechanism.).
Turn off the master power supply (if last moving component).
```

The hardware datum position will be left unchanged. If the datum position has changed significantly since the system was datumed, a warning will be generated.

2–3.4.3.6 Diagnose mechanism

On receipt of this command, NIRI will:

```
Turn on the master power supply (if necessary).
Unlock the component. (Only the Beam-Splitter Wheel has a locking
   mechanism.)
Move the mechanism and attempt to verify that the motors and Hall-
   effect sensors are operating.
Lock the component (Only the Beam-Splitter Wheel has a locking
   mechanism.).
Turn off the master power supply (if last moving component).
```

2–3.4.4 Focus and offset commands

The OIWFS will have a focus stage and two gimbal mirrors. The following commands will act on the above mentioned mechanism:

2–3.4.4.1 Select position

On receipt of this command, the OIWFS CC control software will:

```
Ensure an observation is not taking place.
Turn on the master power supply (if necessary).
Move the focus mechanism to the engineering position corresponding to
   the desired focus.
Compute the expected values for the Hall-effect sensors.
Turn off the master power supply (if last moving component).
Verify that the final hall effect sensors match the expected values.
```

2–3.4.4.2 Park mechanism

On receipt of this command, the OIWFS CC control software will:

```
Ensure an observation is not taking place.
Turn on the master power supply (if necessary).
Move the focus mechanism to its parking position.
Turn off the master power supply (if last moving component).
```

2–3.4.4.3 Datum focus

On receipt of this command, the OIWFS CC control software will:

```
Ensure an observation is not taking place.
Turn on the master power supply (if necessary).
Move the mechanism and search for its coarse datum.
Move the mechanism and search for its secondary datum (if necessary).
Move the mechanism and search for its datum.
Record the new datum position and use this to update the position to
   engineering position look-up table for the mechanism.
Turn off the master power supply (if last moving component).
```

2–3.4.4.4 Redatum focus.

On receipt of this command, the OIWFS CC control software will:

```
Ensure an observation is not taking place.
Turn on the master power supply (if necessary).
Move the mechanism and search for its coarse datum.
Move the mechanism and search for its secondary datum (if necessary).
Move the mechanism and search for its datum.
Turn off the master power supply (if last moving component).
```

The hardware datum position will be left unchanged. If the datum position has changed significantly since the system was datumed, a warning will be generated.

2–3.4.4.5 Diagnose filter wheel

On receipt of this command, the OIWFS CC control software will:

```
Ensure that an observation is not taking place.
Turn on the master power supply (if necessary).
Move the mechanisms and attempt to verify that the motors and Hall-
   effect sensors are operating.
Turn off the master power supply (if last moving component).
```

2–3.5 Detailed Status information

Details of these records are still TBD.

2–3.5.1 General records

2–3.5.1.1 name
This record contains a string constant describing the name of the instrument. For NIRI this record will contain "NIRI WFS"; for GNIRS the record will contain "NIRS WFS".

2–3.5.1.2 state
This record will take the values BOOTING, INITIALISING, RUNNING and CONFIGURING and reflect the state of the NIRI control software while it is initializing. The activities corresponding to these states are

- BOOTING — the state during IOC Init and while any database default values are being set.
- INITIALISING — the NIRI control software will be checking its own hardware and reading its hardware configuration files during this phase.
- RUNNING — the NIRI control software is ready to accept commands.
- CONFIGURING — the NIRI control software is ready to accept commands but it is busy configuring and unable to make observations.

2–3.5.1.3 health
This record will reflect the overall health of the OIWFS. Expected values are GOOD, WARNING or BAD. If the overall health is BAD then the NIRI control software will be unusable. If the Health is set to WARNING then the OIWFS control software will be able to execute some functions but not perhaps to specification.

2–3.5.1.4 historyLog
The record used to contain messages to be recorded in the history log.

2–3.5.2 Wheel records
These are TBD.

2–3.5.3 Focus/Offset records
These are TBD.

2–3.5.4 Environment Records
These are TBD.

2–3.6 Alarm conditions

The following situations will generate an alarm or warning:

- Hall sensor disabled
- Motor-control module fault
- Diagnose failed
- Redatum failed
- Datum failed
- Move-confirmation failed

- Temperature changing
- Large temperature gradient
- Limit sensor reached
- Motion out of valid range

2–3.7 Debugging and Maintenance

This interface can be tested and debugged by means of an engineering user interface supplied as part of the NIRI.

2–3.7.1 Debugging Modes

The NIRI control software will have the standard debugging modes described in ICD 1a, [24].

2–3.7.2 Calibration

If a hall sensor fails or if a mechanism must be disassembled for maintenance, the sensors will need to be recalibrated. A separate program will be provided for this purpose. Calibration procedures will be described in a separate document [5].

2–3.8 Simulation

The Gemini simulation modes are described in ICD 1a, [24]. The NIRI control software will use the FAST and FULL modes, and by default it begins in "NONE" mode. VSM mode may be supported, if time permits.

2–4.0 Safety Issues

It will be necessary to avoid moving OIWFS components while the mechanical components are not at thermal equilibrium. Records will be provided to provide a temperature interlock. Details are described in [6].

ICD 1.9/1.10ab

"NIRI/GNIRS Science Instrument to On Instrument Wavefront Sensor"

niri_hty_005

Hubert Yamada
1997-06-19

Version: 0.2

Issued By: Hubert Yamada

		Date:
Approved By:___________________________	Group Manager	_________
___________________________	Group Manager (if req)	_________
___________________________	Systems Engineering	_________

Revision Control

1. Revision Version # 0.0
   Date: 1997-04-02
   Revised by: Hubert Yamada
   Reason for / items changed: Original prerelease version
2. Revision Version # 0.1
   Date: 1997-04-29
   Revised by: Hubert Yamada
   Reason for / items changed: Critical Design Review version
3. Revision Version # 0.2
   Date: 1997-06-19
   Revised by: Hubert Yamada
   Reason for / items changed: Merged references and definitions with master NIRI reference and definition list.

Last Modified: Thursday, June 19, 1997 5:43 pm

Chapter 1 Introduction

1–1.0 Description

1–1.1 Purpose

This Interface Control Document (ICD) serves the following purposes:

- To describe the commands and parameters that will pass between the Near Infra-Red Imager (NIRI) and its on instrument wavefront sensor (OIWFS) and between the Gemini Near Infra-Red Spectrograph (GNIRS) and its OIWFS

The intended audience for this document is:

- the NIRI reviewers;
- the GNIRS reviewers and developers.

1–1.2 Scope

This document describes commands which pass between NIRI/GNIRS and their respective OIWFS's.

This document does *not* describe the following topics:

- the NIRI detector and components, (see [18], [11], and [12], summarized in [3]);
- the GNIRS hardware and electronics, (see [18] and (???))
- the OIWFS hardware and electronics, (see [11] and [12], summarized in [3]);
- the NIRI/GNIRS OIWFS detector and components (see [11], [12], and [13], summarized in [3]);

The software design is not described here. For that, see [2] (currently under preparation), and [3] (currently under preparation).

1–2.0 Related Documents and Drawings

1–2.1 References

See "Near IR Imager (NIRI)—Definitions, Abbreviations, and References".

1–2.2 Definitions, Acronyms and Abbreviations

See "Near IR Imager (NIRI)—Definitions, Abbreviations, and References".

1–2.3 Stylistic Conventions

References to documents in Section 1.5, "References," on page 2 of "Near IR Imager (NIRI)—Definitions, Abbreviations, and References" are given like this; [1].

Chapter 2 Details of the NIRI/GNIRS to OIWFS Interface

2–1.0 Introduction

2–2.0 Physical System Interfaces

2–2.1 Mechanical Interface

Not Applicable for a software ICD.

2–2.2 Optical Interface

Not Applicable for a software ICD.

2–2.3 Electronic Interface

Not Applicable for this ICD.

2–2.4 Mass/Balance

Not Applicable for a software ICD.

2–2.5 Thermal Interface

Not Applicable for a software ICD.

2–3.0 Software/Control Function Interface

2–3.1 Overview

2–3.1.1 Communications infrastructure

The communications infrastructure is the same as described in [16].

2–3.2 Implementation

This section contains tables summarizing the EPICS records used for the interface. The next sections (Section 2–3.2.1.1 on page 5) contain the details of what those records do.

It is assumed in all cases that the record names given in this document will be prefixed by the string "niri:" or "nirs:" when the database is implemented, except for the SAD database which will be prefixed by "niri::" or "nirs::".

2–3.2.1 Command (CAD/CAR) records

The columns of the command record tables are as follows:

Description. A brief description of the command.

Attribute. The name of the attribute which the Observatory Control System associates with a particular parameter of a particular command.

CAD record/field. The name of the EPICS CAD record (minus the "niri:" or "nirs:" prefix) together with the field in that CAD record associated with each parameter of the command.

CAR record. The CAR record associated with the command. Note that several CAD records may share the same CAR record.

These commands are executed by asserting the START directive on the CAD record.

TABLE 1. NIRI/GNIRS OIWFS interlock CAD/CAR records

Description	Attribute	CAD record:field ("niri:" +)	CAR record ("niri:" +)
lkCfg	Lock for configuration (Bool)	wfs:lkCfg.A	wfs:lockC
setTmp	Lock for cool down / warm up (WARM|COLD|CHANG-ING)	wfs:lkEnv.A	
lkObs	Lock for observation (Bool)	wfs:lkObs.A	
lkHW	Lock for hardware protection (Bool)	wfs:lkHW.A	

2–3.2.1.1 Detailed Command Description

2–3.2.1.2 LkCfg

When this lock is enabled, no cpu-intensive commands will be executed by the OIWFS CC.

An attempt to set this lock will be rejected if a command is currently being executed.

If the OIWFS CC is currently in "follow" mode, it will *not* update its probe position; and the positioning error will be *incorrect*.

2–3.2.1.3 SetTmp

This command informs the OIWFS about the current temperature of the system. There are three possible states:

WARM. System is at room temperature.

COLD. System is at its operating temperature.

CHANGING. System is currently being cooled down or warmed up.

This command will be rejected if a command is currently in progress.

If the OIWFS CC is currently in "follow" mode, it will *not* update its probe position; the positioning error will be *meaningless*.

When the temperature of the system is CHANGING all temperature-sensitive components will be disabled. When the temperature is first set to WARM or to COLD, the OIWFS will enter the not-datumed state. (The datum shifts slightly, depending on the temperature of the system.)

The OIWFS CC should never be in “follow” mode unless the system is already cold and its temperature has stabilized. Probe update commands will be ignored. The positioning error will be meaningless unless they system is COLD.

2–3.2.1.4 LkObs

When this lock is enabled, the system will reject all motion commands. This command is provided to provide a low noise environment for sensitive observations.

An attempt to set this lock will be rejected if the system is currently if a command is currently in progress.

If the OIWFS CC is currently in “follow” mode, it will *not* update its probe position; however the positioning error will be *correct*.

2–3.2.1.5 LkHW

When this lock is enabled, the system will reject all motion commands. This command should never be used in normal operation. It is provided as a “catch-all” command to lock all hardware for any unspecified reason.

This command is *always* accepted and is always processed as soon as possible. Any commands that are currently being processed will FAIL. If the OIWFS CC is currently in “follow” mode, it will *not* update its probe position; the positioning error will be *meaningless*.

2–3.2.2 Safety Issues

lkEnv (or lkHW) must always be issued when the system is not at an equilibrium temperature. Various components may not operate correctly (and could be damaged) if they are moved when they are not at a uniform temperature.