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

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

Gemini
Near
Infra-Red
Imager

NIRI

Near IR Imager (NIRI)— Definitions, Abbreviations and

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[1] niri_hty_000.fm (obsolete, superseded by [2] and by [3]), “NIRI Optical-Components Controller: Implementation Issues,” Hubert Yamada, Ev Irwin, Tony Young, University of Hawaii, Institute for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

[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

NIRI

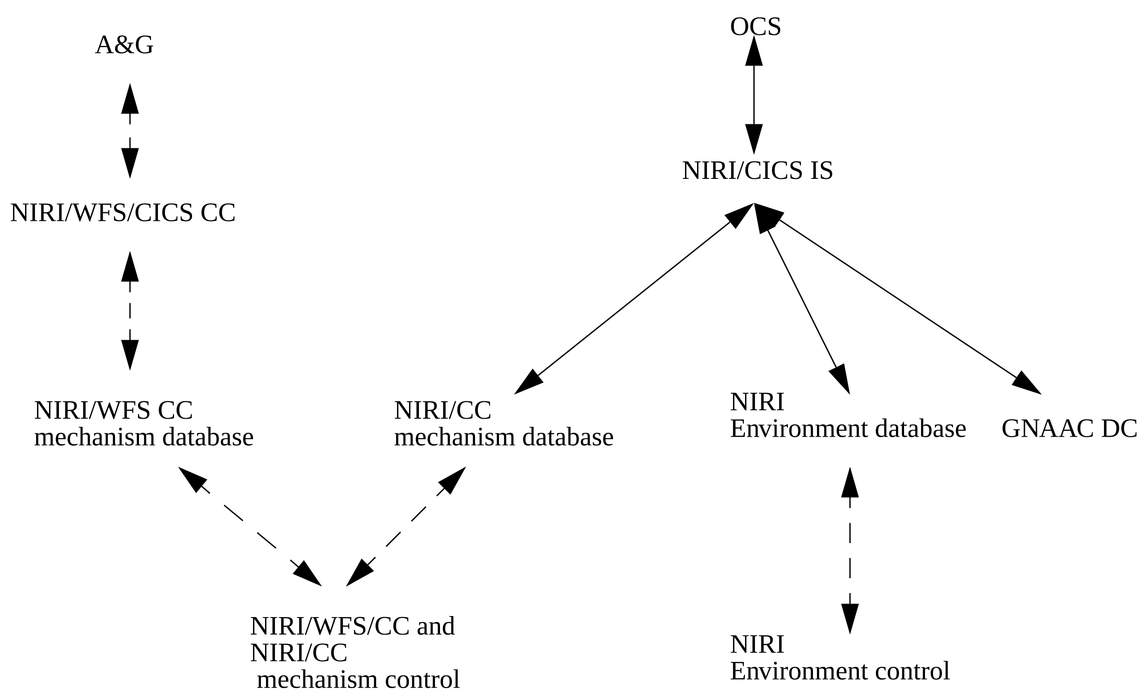
Near IR Imager (NIRI)—

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”.



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

- 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

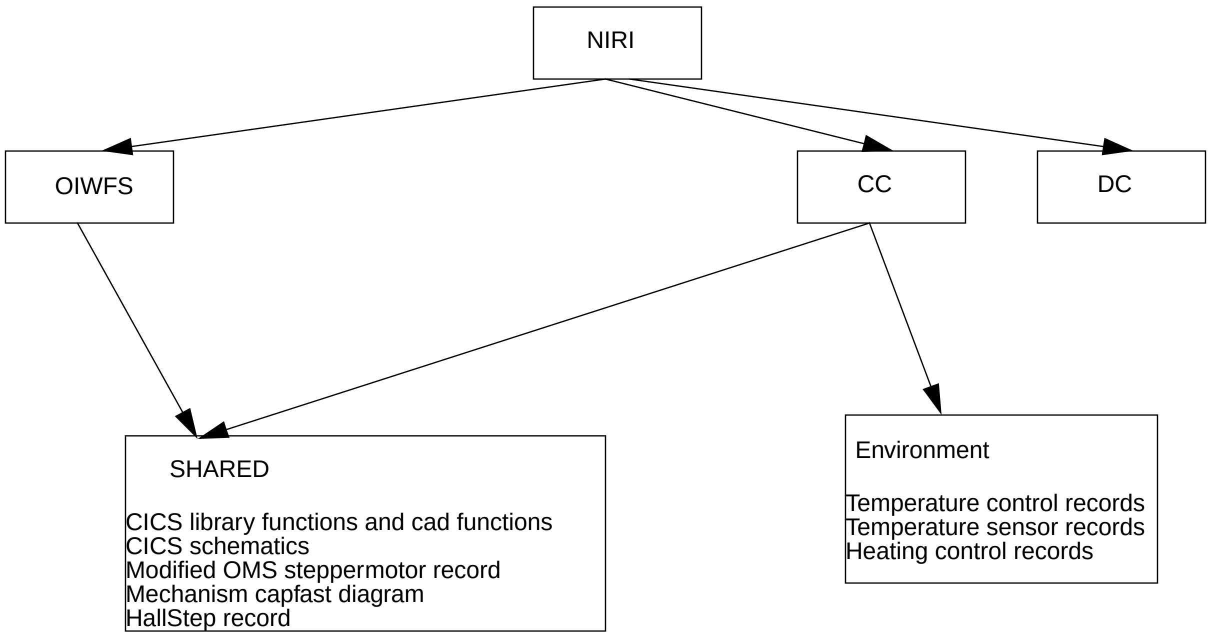
See “Near IR Imager (NIRI)—Definitions, Abbreviations, and References”.

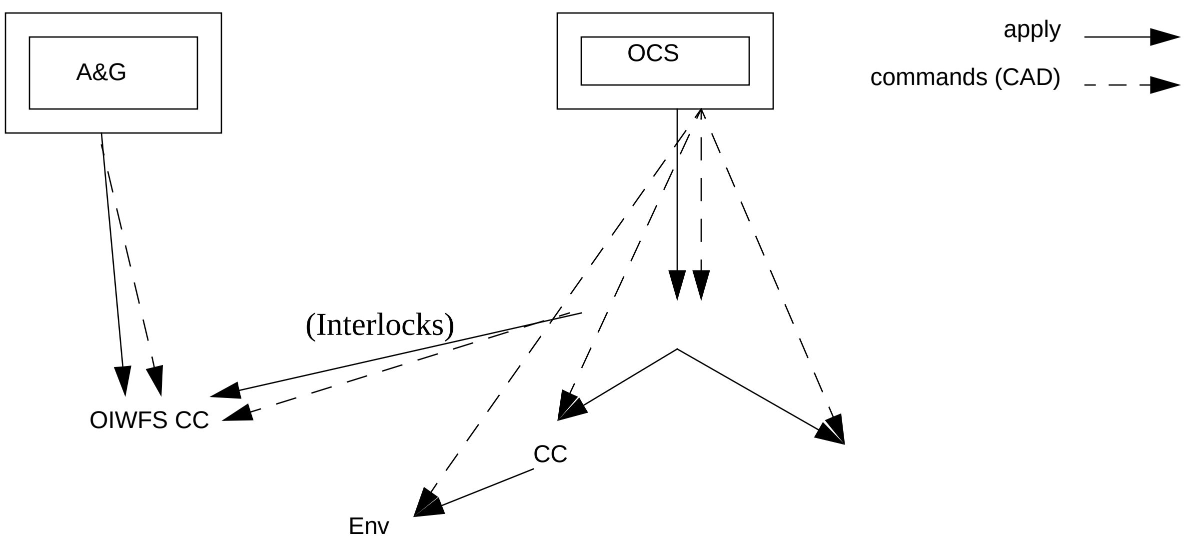
1.4 Stylistic Conventions

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”.

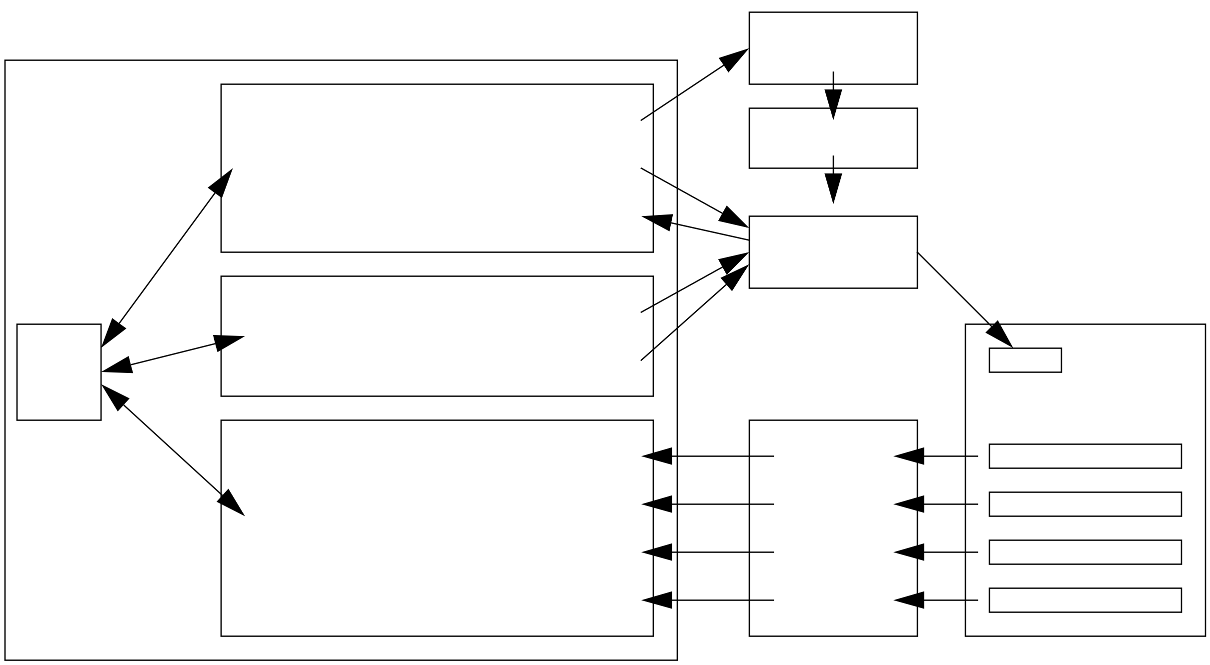

NIRI
OIWFS
CC
DC
SHARED
CICS library functions and cad functions
CICS schematics
Modified OMS steppermotor record
Mechanism capfast diagram
HallStep record
Environment
Temperature control records
Temperature sensor records
Heating control records


A&G
OCS
apply
commands (CAD)
(Interlocks)
OIWFS CC
CC
Env

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].



- 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.

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.

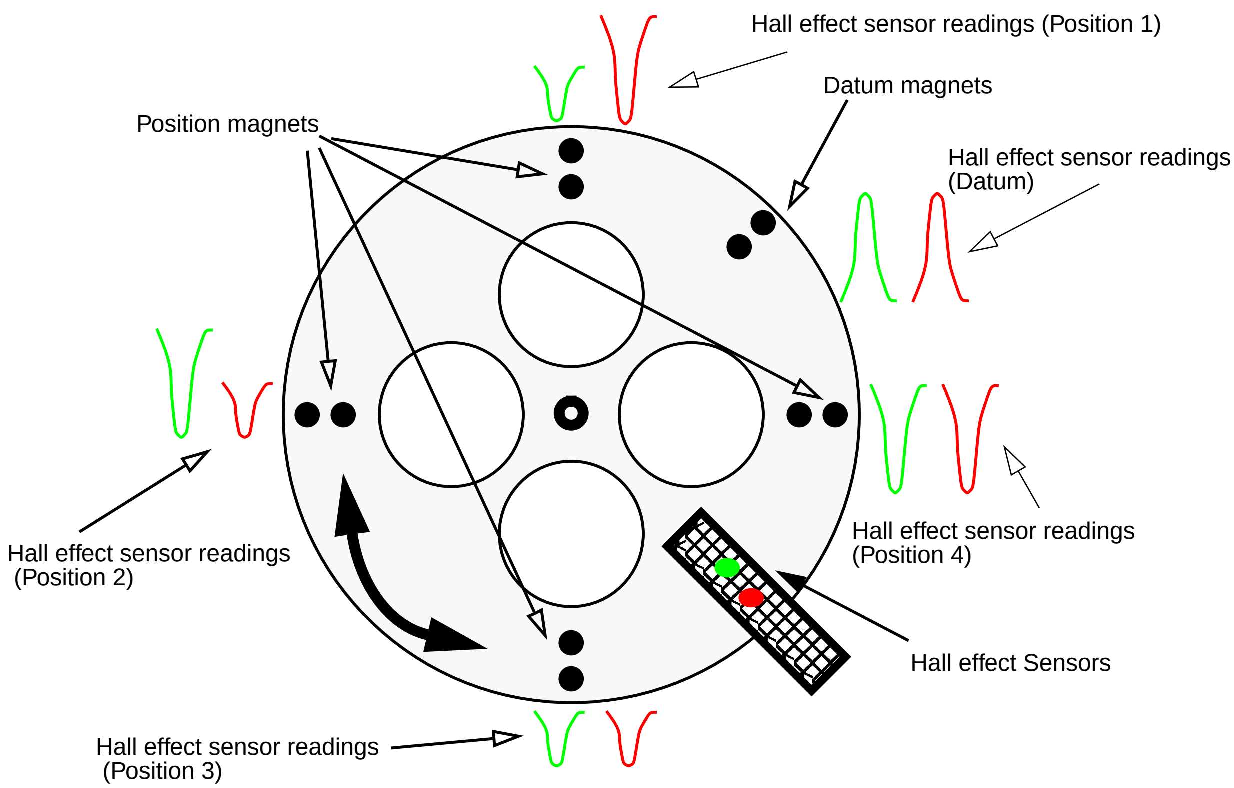


Table 2 describes typical sensor values for a twelve-position wheel. Note that it is assumed that it is only possible to distinguish four different magnet strengths. In practice, it may be possible to distinguish more values.

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.

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.

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.

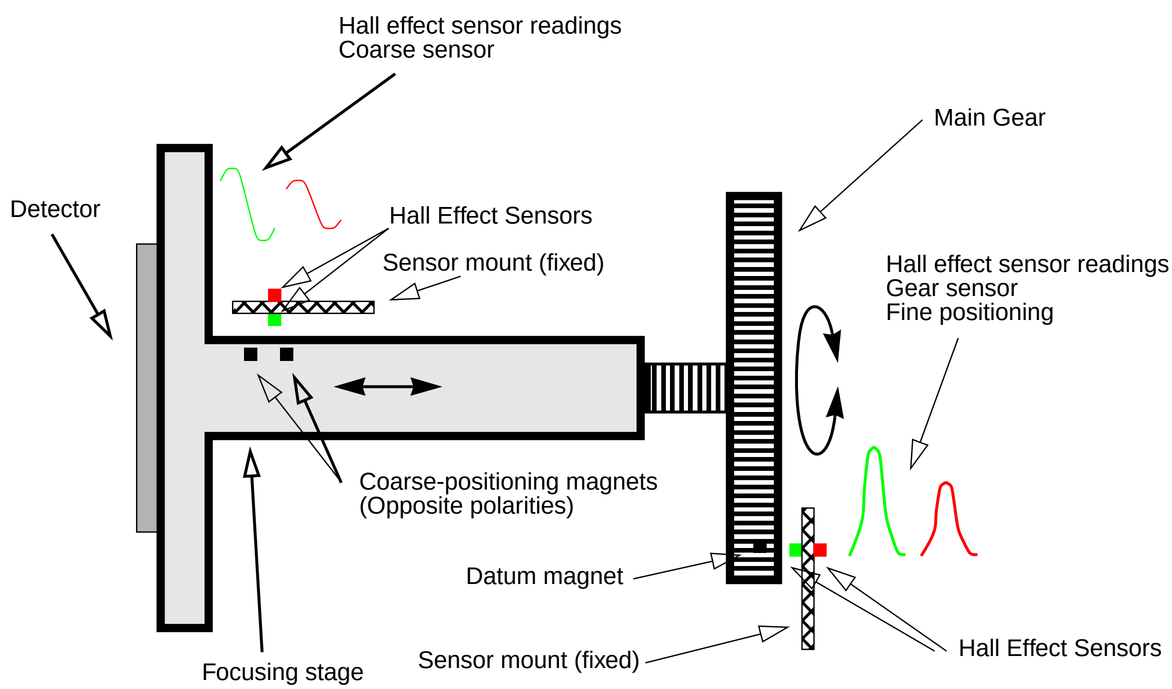
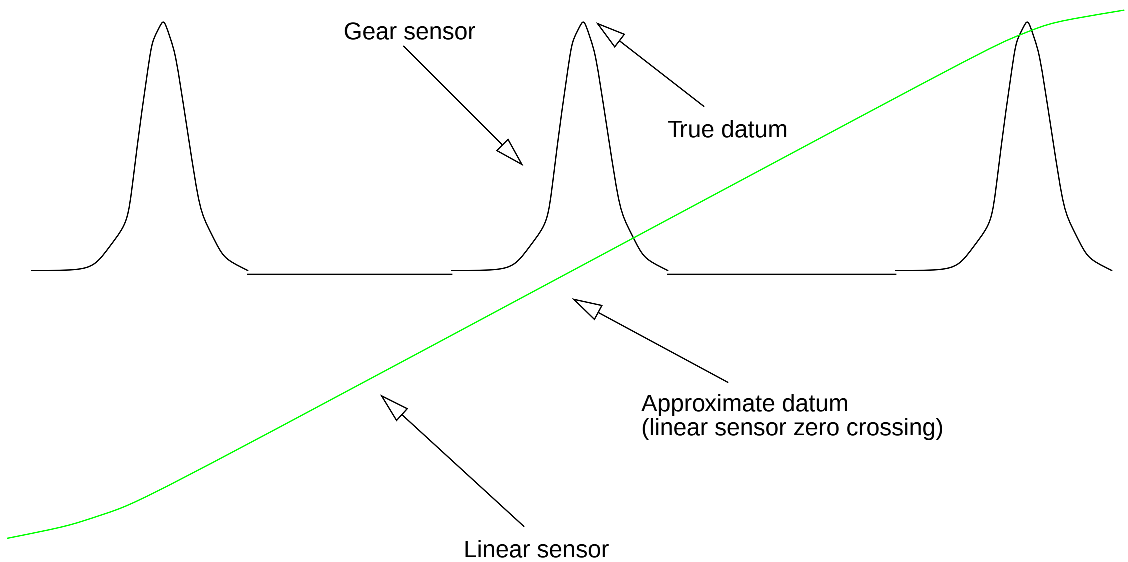
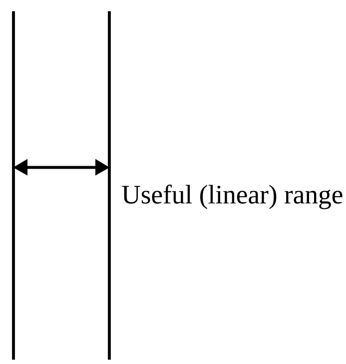


Figure 8 shows the expected reading from the Hall-effect sensors.



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.



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.

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.

- 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.

- 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.

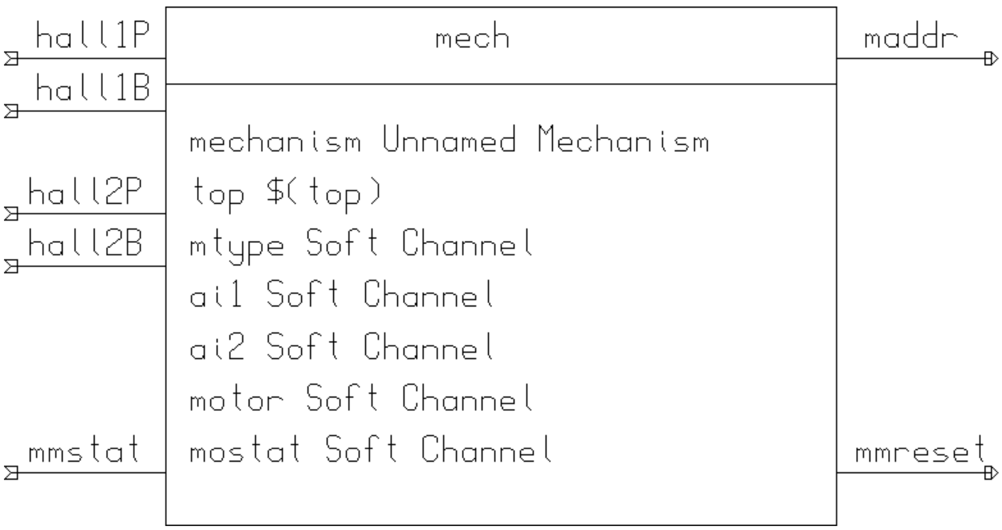
7.2.3 The ENV Database

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.

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.

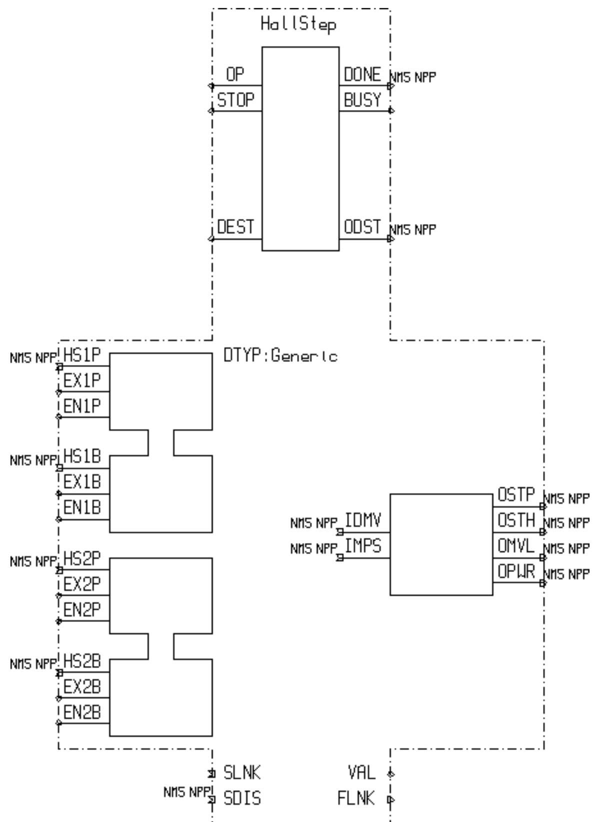
$(top)Busy.VAL	Busy	Read	Operation in progress?	Non-zero means operation in progress
$(top)Done.VAL	Done	Read	Status after operation	Non-zero means error
$(top)Hallstep.MESS	Completion message	Read	Error/Status message after an operation	Cleared by the next operation
$(top)Hallstep.DATM	Datumed?	Read 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.*

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.

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

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

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

Near IR Imager (NIRI)—Software

Gemini
Near
Infra-Red
Imager

NIRI

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

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

Gemini
Near
Infra-Red
Imager

NIRI

Near IR Imager (NIRI)—Software

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.

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].

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.

- 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

1.4 References

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

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.

2–3.3.1.2 General observing command CAD/CAR records

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.

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.

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.

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.

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.

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.

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).
```

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

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].

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.

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].

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.

2–3.4.1.1 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

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).

2–3.5.1.1 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