MAUNA KEA INFRARED

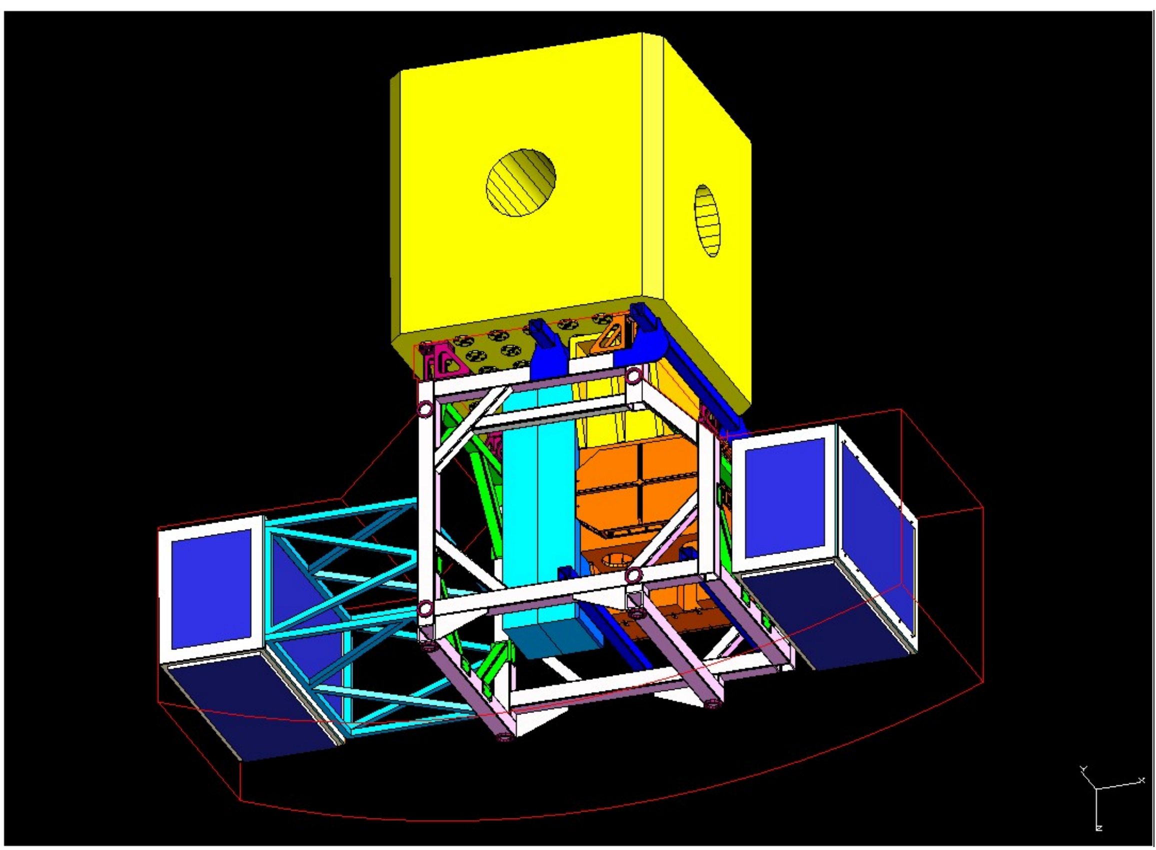
SOFTWARE ARRAY CONTROL

INTERFACE DESCRIPTION

MKIR# NICI SW - 800-202-02

REVISION 0.80 (Preliminary)

1/20/05



Revision History

Revision	Author	Summary of revisions	Date
0.80	Mike Thompson	Prepared for preliminary 80% release of manuals to Gemini. See version 0.79 for issues.	1/20/05
0.10	Mike Thompson	Original revision.	5/29/03

Table of Contents

Table of Figures and Photos

1 Introduction

This document provides a subsystem level description of the software portion of the Redstar3 Array Controller developed by Mauna Kea Infrared. The Redstar3 provides control of and data acquisition from four quadrant Aladdin type III 1024x1024 arrays. The Array Control Interface can be used for configuring and driving arrays.

High image throughput rates are designed into this controller. The controller is implemented with Independent control and data pathways and processing. That is, image capture processes are separated from instrument control processes for increased throughput.

This document describes the instrument level control for acquiring images through the Aladdin III array. The following is a discussion of basic image acquisition concepts, usage scenarios, and the user level commands for image acquisition.

Section 2 lists the technical specifications, that is, the capabilities, of the Array Control Interface.

Section 3 provides a discussion on basic image acquisition concepts, a use scenario, definitions of the array control commands, and a description of the GUI.

Section 4 provides definitions of terms and acronyms used throughout this document.

2 Technical Specifications

The technical capabilities of the array controller are listed in this section.

Operates Raytheon Aladdin III 1024x1024 FPA

- This Aladdin III array is supported with all basic astronomical modes.
- Provides programmable control of select bias voltages such as detector bias (VGGCL, VDETCOM, and VDDUC).
- Supports several readout modes:
  - Single reads.
  - Double correlated reads.
  - A noise reduction (Fowler sampling) read mode is desired and will be included as a goal.

Stand-alone/Remote Operation

- Standalone: The controller can operate as a standalone interface that provides full control and image display.
- Remote: The controller can operate in remote mode in which all functions can be issued through a socket type interface.

Coadd buffers

- A minimum of eight, 32 bit, coadd buffers are required with a goal of 16 buffers.
- Coadd buffers are user selectable from the GUI or socket.
- The macro language allows the user to select the coadd buffer.

Subarray operation

- Subarray operation is supported in all modes.
- Minimum required subarray size is 64x64 with a goal of 8x16.

Macro capability

- A scripting language is provided that allows all commands to be executed from a script file.
- The program that executes the scripting language resides in the user interface when running stand-alone.
- In remote mode the facility provides the macro or scripting capability built into the sequencer.

Image Monitor (also known as Data Viewer, DV)

- A near real time display, or image monitor, is provided to aid in the setup and adjustment of the instrument. Adjustment of focus is an example. See the User Manual addendum Data Viewer Description for details on using DV.
- The requirements for the image monitor are flexible in terms of resolution but it must operate quickly to allow feedback for manual adjustments.
  - The numbers of bits of resolution can be set in order to meet frame rate requirements (i.e. an 8 bit display is allowed). The pixels can be sub-sampled as well to meet the speed requirements.
  - Subarrays may be used to speed up the frame rate and is supported on the image monitor. This does not change the array readout, just what is displayed.

3 Functional Description

The imaging function of the Array Controller is to gather information from an image detector and combine that information with ancillary data and prepare the images for subsequent computer processing and human viewing. Some of the ancillary data is in the form of camera and telescope setup information. This ancillary data is used to form a final raw image or FITS file.

For this discussion please see the block diagram of the Array Controller provided in Figure 1. The Array Controller hardware consists of an Instrument Control (IC) Server, a Pixel Server, Array Control Electronics, a Terminal Server (DigiPort), and an array. The Instrument Control Server is the main controller of the instrument. Software running in the IC Server drives all of the major functions of the Array Controller. In the Array Control Electronics the IPEngine1 (the hardware Clocker) drives clock signals to the array and the IPEngine2 (Catcher) prepares the raw data from the array for transmission over the fiber link to the Pixel Server. The Pixel Server assembles coherent images from the raw pixel data stream, can perform some image processing, and prepares the image for storage and viewing. The Terminal Server provides an Ethernet to Serial interface for mechanism control.

Following is a discussion of the user level commands used to gather the images. There is also a discussion on Basic Image Acquisition Concepts, a Use Scenario, Array Clocking Patterns, User Programmable Array Bias Voltages, and the GUI. All of the commands for controlling the Array Controller are accessible through the GUI, the GUI's command line, the macro interface, and the socket interface.

The software's command set is divided into System Setup Commands, System Commands, General Setup and Configuration Commands, and Data Viewer (DV) Commands. All of the commands defined in this document are accessible to the user. The commands are executable via the command line in the GUI or via a command socket. These commands are defined in the sections that follow.

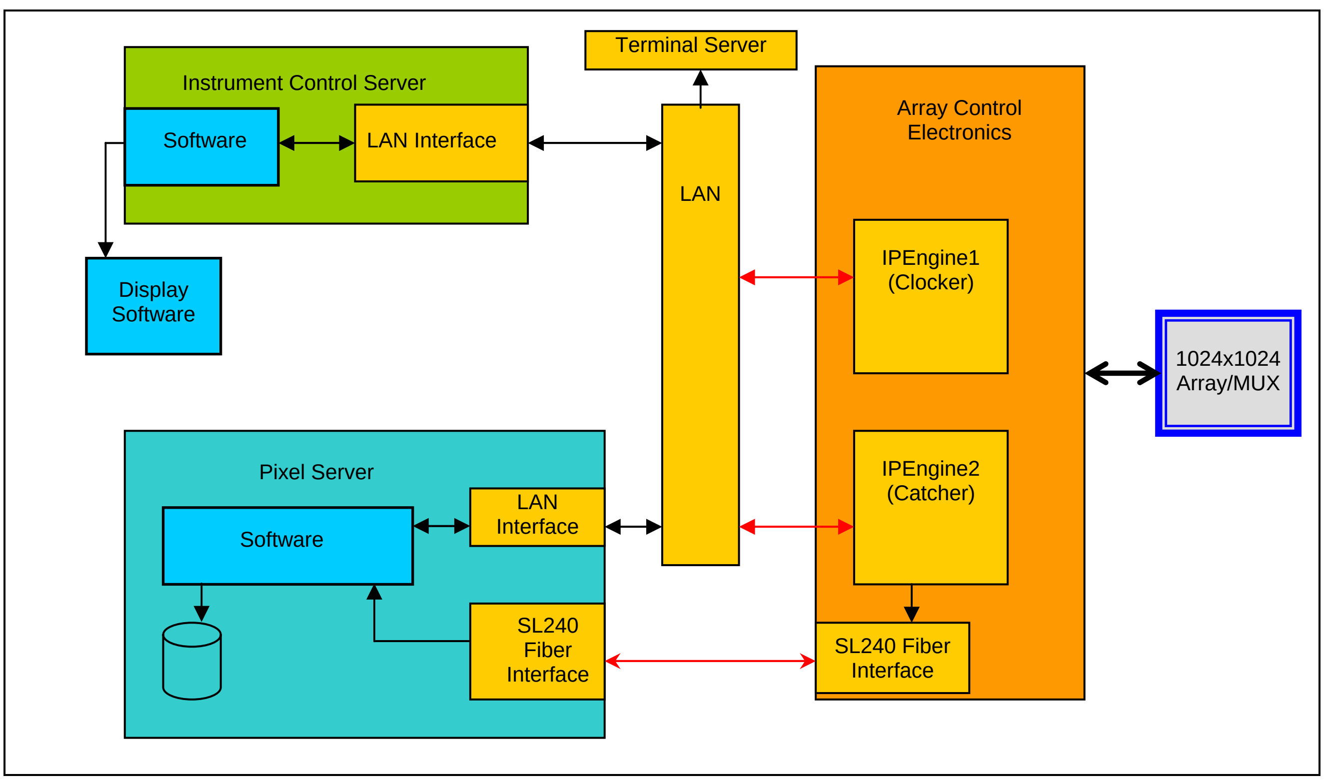


Figure 1 Block Diagram of the Array Control Electronics

3.1 Basic Image Acquisition Concepts

This section describes some basic concepts related to image acquisition.

3.1.1 Resets

During times in which images are not being acquired, light may still be falling on the array. The resulting accumulation of electrons in the wells of the array will overexpose the image and may cause other aberrations. The system will flush these extra electrons at continual and specific times. This is called a background reset (BGR). A timing diagram of BGRs is illustrated in Figure 2. There are 3 parameters that can be set to control background resets. The first parameter, NS, specifies the time for which the reset line, Vrstg, is held high. The second parameter, MS, specifies the time between BGRs. The last parameter, Min.MS, specifies the time from the last BGR to the beginning of the acquisition process.

There is an additional reset just before the image is scanned (so that an exact duration of integration is known). Note that the time of integration of any particular pixel depends on how many pixels are ahead of it in the scan sequence as well as how fast each pixel can be scanned. This information can be computed downstream if needed, so only the time between last reset and first pixel acquisition is used to compute the duration of integration.

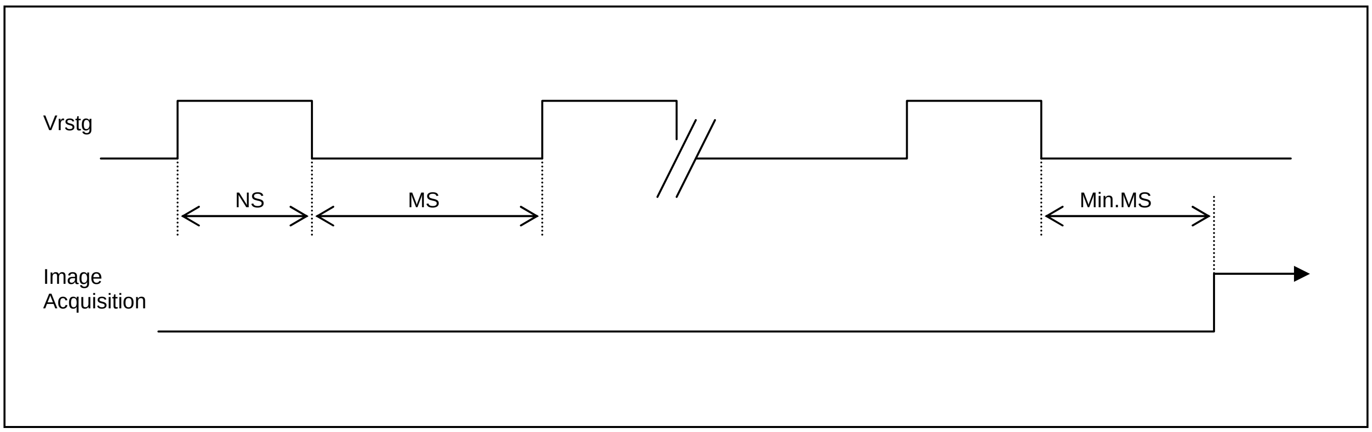


Figure 2 Background Reset Timing Diagram

3.1.2 Subarrays

Some areas of the image may be deemed more important than others. To increase the speed of scanning, the important areas can be identified as subarrays. A specific scanning sequence (and the corresponding decoding sequence) must be generated that will read out only those desired regions, discarding the data from unwanted regions. The scanning sequence is called the ‘Clocking Pattern’ and must be sent to the clocker as needed. The image comes off of the Aladdin imaging device in a non-intuitive and hardware dependent order, thus, the image decoding sequence must correspond to the clocking pattern. This decoding sequence is sent to the pixel server to reorder the image into the rows and columns of the image.

Subarrays are enabled by the ‘ARRAYMODE’ command or in the GUI’s Observation Window. The subarray size is set by the ‘ARRAY’ command or set in the GUI’s Observation Window.

3.1.3 Pedestal Frames

When the Aladdin III device is reset to clear excess electrons, the reset itself generates some small number of electrons on the surface of the image array. If the noise is recorded immediately after the reset operation (called a pedestal frame) it can be subtracted from the subsequent scan. This results in a sharper, less noisy image. The pedestal frame is taken after the 'rtime' parameter. Subsequent scans will have this data subtracted out. Usually, if not always, each reset may need a separate pedestal. Note that the subarray settings will not change between pedestal and subsequent scans.

3.1.4 Coaddition

Multiple scans of the same image may be taken. These scans will result in data that needs to be accumulated into separate buffers. When all the scans are taken, the buffers may be combined by averaging, adding or subtracting, or may be written out to the final destination separately.

Coaddition is enabled by the 'COADD {num}' command or in the GUI's Observation Window.

3.2 Use Scenario

To make an observation, at least the steps described here must be executed. First a qualitative description is given and then a list of the steps and their associated commands is given. Note that 'user' actions referenced are equally applicable to any scripts run remotely from the instrument sequencer.

First an image of a distant object is focused on the array. The MKIR controller is initialized and ready to gather this image. The image is to be scanned in its entirety and sent to the pixel server which decodes the data stream and saves it for viewing. The exact time of the image capture is needed, as well as other information, such as where the telescope is pointing and the filters in the light path.

Here the basic steps to make an image capture are listed along with their command line equivalents in parentheses. Note that this is a generic example. Additional steps and commands may be necessary for some observations.

- The user points the telescope at the desired object.
- The user sets an integration time. (ITIME {time})
- The user sets an exposure mode. (ARCMODE {arc_s | arc_d})
- The user sets an integration count. (NDR {count})
- The user sets a cycle count. (CYCLES {count})
- The user selects the data destination buffer. (DESTBUF {bufsel})
- The user selects between Full Array and Subarray mode. (ARRAYMODE {full | sub})
    - If Subarray Mode is selected, set the subarray (region of interest). (ARRAY {x} {y} {wid} {hgt})
- The user connects the IC to the peripherals. (CONNECT)
- The user starts the exposure. (GO)

3.3 Array Clocking Patterns

There are 16 signals that are used in this Redstar3 implementation for controlling the Aladdin III array. All of these signals are referred to as “clocks” although some of the signals may be constant voltages. It is by driving these clock signals that images are captured from the array. The specific sequence of manipulating these clocks for array control is called a clocking pattern.

Clocking pattern generation is accomplished by a combination of software and hardware. Software in the Instrument Control Server generates a specific clocking pattern for each observation based on the configuration parameters set by the user. This clocking pattern, called a Superpattern, is sent to the hardware Clocker in the Array Control Electronics which executes the pattern on the array.

The Array Controller is shipped with built-in, carefully optimized clocking patterns. It is recommended that users use the patterns that shipped with the array controller. If the customer wishes to modify the clocking patterns, extreme care must be taken as the arrays are very sensitive devices. Improper clocking patterns can result in unpredictable array behavior or cause damage to the arrays.

The following discussion describes how to generate array clocking patterns.

3.3.1 Clocking Pattern Building Blocks, pFiles and patFiles

For modular and flexible control over array operation, the Array Controller utilizes a suite of basic pattern building blocks to build the Superpattern. Each of these building blocks contains a clocking pattern that executes one atomic operation on the array. For example there are atomic operations for an array reset, row increment, and clocking out 16 pixels of array data.

For human viewing and editing these atomic operations are stored in pFiles. A pFile is a human readable text file that provides a method for manipulating the 16 clock signals in such a way as to accomplish an array operation.

The pFiles are compiled into patFiles with the cpat utility. It is possible, although much more difficult, for the user to create patFiles directly. The patFiles are text files that contain sequences of clocking entries for the 16 clock signals and durations for which the clocking entries should be asserted to the array.

The atomic operations contained in the patFiles must be applied to the array in specific sequences for clocking out the array. **The patFiles are sourced by the Instrument Controller software at startup** and are used to assemble the Superpattern.

There are two sets of files, one for fast readout with acceptable noise levels, and another with a slower readout that has been optimized for the lowest noise levels possible. Selection between fast and slow patterns is done with the ‘DOFASTMODE’ command.

The pFiles and patFiles can be found in the following directories.

/home/nici/redstar3/solar/fastpatterns
/home/nici/redstar3/solar/slowpatterns

3.3.1.1 pFile Format

Each pFile contains a command oriented description of a clocking pattern. Each line in a pFile contains one command. The commands provide a method for describing a clocking pattern. Care must be taken to assert and de-assert all of the clock signals at the appropriate times. Once a clock signal is set, it remains set until it is explicitly cleared. Figure 3 provides a list of the commands that can be used in pFiles.

The commands in pFiles are compiled into patFiles by the cpat utility. See Section 3.3.1.4 for an explanation of the cpat utility.

Command	Description
BCLEAR <mask>	Turn off the specified bits in the mask.
BDEF <mask> <value>	Defines a 16 bit hexadecimal mask- <mask> is a text string that will represent the 16 bit (hex) mask <value>.
BSET <mask>	Turn on the specified bits in the mask.
DESC <text>	Description text for the pattern in the pFile.
OUTFILE <filename>	Sets the output file name.
PROCESS <filename>	Execute the commands in file <filename>.
RESET	Resets all internal variables to startup condition. **This is not an array reset.**
SETENV <var>=<value>	Create an environment variable. For example setenv SettleTicks=12 allows you to put this in your pFiles: tick $SettleTicks
SHOW	Show all internal variables.
TICK <n>	Hold the current mask for n clock ticks. This command is basically a no-op that waits the specified number of clock ticks.
WRITE	Generates a patFile which is written to the file specified by the OUTFILE command.
# <comment>	The comment character '#' is used as the first character of a line.

Figure 3 Table of pFile Commands

Here is an example of a pFile.

```
#Example pFile
reset
pmask on
outfile test.pat
desc   Description of Pattern 75
bdef  frame       0x00000001
bdef  convert     0x00000002
bset   frame
tick 2
bset   convert
tick 1
bclear convert
tick 2
write
```

3.3.1.2 patFile Format

The patFiles contain a list of alternating Clocking Entry Durations and Clocking Entry Values. Comments are made with the '#' character as the first character on a line. Ignoring lines occupied by comments, odd numbered lines contain on Clocking Entry Duration. Even numbered lines contain 16 bit hex Clocking Entry Values which specify which clock bits are to be set and clear. A Clocking Entry Value is held at the array inputs for the duration specified by the preceding Clocking Entry Duration. Here is an example.

```
# Example patFile
# Initial condition.  Hold for 6 ticks.
0006
8440
# Assert bit 0.  Hold for 1 tick
0001
8441
# Clear bit 0.  Hold for 15 ticks.
000F
8440
```

3.3.1.3 Bit Definitions

Generally the pFile commands provide a way to turn on and off the 16 clock signals. So there are 16 bits to be controlled in clocking patterns. Each bit is assigned a number from 0 to 15. The following table defines the clocks that correspond to bits in the pFiles and patFiles.

Bit	Clock Signal
0	vggcl
1	vrowon
2	vrstr
3	vrstg
4	read_data
5	convert
6	Frame
7	phiSS
8	vddcl
9	phiS1
10	phiS2
11	phiSOE
12	phiDES
13	phiFS
14	phiF1
15	PhiF2

Figure 4 Table of Bit Positions for Clock Signals

3.3.1.4 The cpat Utility

The cpat utility is a clocking pattern table generator. The utility takes a command oriented description of a clocking pattern from a pFile as input. The output is an ASCII file containing a clocking pattern table in a patFile. The patFiles are used by the IC software to generate Superpatterns.

The supported commands are listed in Section 3.3.1.1.

Name:

cpat – compile pattern

Synopsis:

cpat [options] <pFile filename>

Options:

-v verbose. Produces very verbose output. Default is off.
-h help. Displays summary of usage and options.

3.3.2 pFiles and patFiles Shipped with the Array Controller

There are a number of pFiles that ship with the Array Controller. Each contains a clocking pattern for a specific array operation.

A naming convention is used for the pFiles for most of the operations. The first two characters specify the general type of array operation.

- p1: Frame Start.
- p2: Address Next Row, i.e.: advance the row counter register.
- p3: Address Next Column.
- p4: Reset a Row/Pair.
- p5: Idle State.

For array operations, rows are grouped into four row sets. The next character, if any, may be ‘a’, ‘b’, ‘c’, or‘d’ which refer respectively to rows 1, 2, 3, and 4 of the four row set.

The final character, if any, refers to specifics for the particular operation.

Note that there is special treatment for clocking the first and last 16 pixels of a row. All pixels in between the first and last 16 pixels are referred to as “middle pixels”.

The pFiles and their associated operations are listed in the table in Figure 5. There are patFiles associated with each pFile that perform the specified function.

Toggle/No Toggle: Commands with the Toggle/No Toggle option permits control over the change of state to phiS1 array clock. When toggled, the array's row register counter is advanced. When not toggled the counter register is not advanced, permitting a double sampling of the row.

pFile	Operation
global_reset.p	Performs a reset on the array.
integration.p	Initiates an integration.
p1.p	Frame Start with Global Reset.
p2an.p	Jump from row A to row B with no toggle of phiS1.
p2at.p	Jump from row A to row B, toggle phiS1.
p2bn.p	Jump from row B to row C with no toggle of phiS1.
p2cn.p	Jump from row C to row D with no toggle of phiS1.
p2ct.p	Jump from row C to row D, toggle phiS1.
p2dn.p	Jump from row D to row A with no toggle of phiS1.
p3af.p	Clock out first 16 pixels of row A (assumed to be current row).
p3al.p	Clock out last 16 pixels of row A (assumed to be current row).
p3am.p	Clock out middle 16 pixels of row B from current position (assumed to be current row).
p3as.p	Skip 16 pixels of row A from current position (assumed to be current row).
p3bf.p	Clock out first 16 pixels of row B (assumed to be current row).
p3bl.p	Clock out last 16 pixels of row B (assumed to be current row).
p3bm.p	Clock out middle 16 pixels of row B from current position (assumed to be current row).
p3bs.p	Skip 16 pixels of row B from current position (assumed to be current row).
p3cf.p	Clock out first 16 pixels of row C (assumed to be current row).
p3cl.p	Clock out last 16 pixels of row c (assumed to be current row).
p3cm.p	Clock out middle 16 pixels of row C from current position (assumed to be current row).
p3cs.p	Skip 16 pixels of row C from current position (assumed to be current row).
p3df.p	Clock out first 16 pixels of row D (assumed to be current row).
p3dl.p	Clock out last 16 pixels of row D (assumed to be current row).
p3dm.p	Clock out middle 16 pixels of row D from current position (assumed to be current row).
p3ds.p	Skip 16 pixels of row D from current position (assumed to be current row).
p4b.p	Reset the A and B row pair.
p4d.p	Reset the C and D row pair.
p5_idle.p	Executes a noop, used for timing.
pnull.p	Executes a noop, used for timing.

Figure 5 Table: pFiles and Their Specific Array Operations

3.4 User Programmable Array Bias Voltages

There are three user programmable array bias voltages VGGCL (DAC0), VDETCOM (DAC10), and VDDUC (DAC11) voltages are programmable. **It is absolutely critical that VDETCOM be set to a more positive voltage than VDDUC or the array can be critically damaged**.

3.5 System Setup Commands

The System Setup Commands provide control over system configuration parameters like hostnames and port numbers for various parts of the Array Controller. This section specifies the System Setup Commands.

3.5.1 PsrvHostname – Set Pixel Server Hostname

PsrvHostname - Set the hostname of the Pixel Server.

- Parameters – hn
- Range - hn: This is a text string. It must be the valid hostname of the Pixel Server system.
- Initial Values - At start up, hn is "".
- Syntax - "PSRVHOSTNAME {hn}"

3.5.2 PsrvPortNum – Set Pixel Server Port Number

PsrvPortNum - Set the port number of the Pixel Server.

- Parameters – pn
- Range - pn: This is a valid port number. It must be the valid port number for the server software on the Pixel Server system.
- Initial Values - At start up, hn is 2000.
- Syntax - "PSRVPORTNUM {pn}"

3.5.3 ClockerHostname – Set Clocker Hostname

ClockerHostname - Set the hostname for the system on which the Clocker software is running. Parameters – hn. Normally this is in the IC.

- Parameters – hn
- Range - hn: A text string for the Clocker system's hostname.
- Initial Values - At start up, the hn is set to "".
- Syntax - "CLOCKERHOSTNAME {hn}"

3.5.4 ClockerPortNum – Set Clocker Port Number

ClockerPortNum - Set the port number on which the Clocker is listening.

- Parameters – pn
- Range - pn: The valid port number on which the clocker is listening.
- Initial Values - At start up, the pn is set to 2000.
- Syntax - "CLOCKERPORTNUM {pn}"

3.5.5 PixelHostname – Set Pixel Software Host

PixelHostname - Set the hostname for the system on which the Catcher software is running. Parameters – hn

- Parameters – hn
- Range - hn: A text string for the Pixel system's hostname.
- Initial Values - At start up, the hn is set to "".
- Syntax - "PIXELHOSTNAME {hn}"

3.5.6 PixelPortNum – Set Pixel Software Port Number

PixelPortNum - Set the port number on which the Catcher is listening.

- Parameters – pn
- Range - pn: The valid port number on which the clocker is listening.
- Initial Values - At start up, the pn is set to 2000.
- Syntax - "PIXELPORTNUM {pn}"

3.5.7 DigiHostname – Set DigiPort Hostname

DigiHostname - Set the DigiPort hostname (the DigiPort contains the remote serial channels)

- Parameters – hn
- Range - hn: A text string for the DigiPort hostname.
- Initial Values - At start up, hn is set to "".
- Syntax - "DIGIHOSTNAME {hn}"

3.5.8 DigiPortNum – Set DigiPort Port Number

DigiPortNum - Set the port numbers on which the DigiPort Serial Channels are listening.

- Parameters - pn1, pn2
- Range – pn1 and pn2
    - pn1: The valid port number on which the serial channel 1 is listening
    - pn2: The valid port number on which the serial channel 2 is listening
- Initial Values - At start up, the pn1 is set to 2001 and pn2 is set to 2002.
- Syntax - "DIGIPORTNUM {pn1} {pn2}"

3.6 System Commands

The System Commands set up connections from the IC to its clients, and starts and stops image acquisition. This section specifies the System Commands.

3.6.1 Connect – Connect IC to Clients

Connect - Tells the IC software to start up and connect the system.

- Parameters – None.
- Range – None.
- Initial Values - At start up the IC software is running and awaiting client connections, but is not connected to other system components (Clocker, Pixel, Pixel Server, Serial Ports).
- Syntax - "CONNECT"

3.6.2 Go – Start Image Acquisition

Go - Starts the system operation and image acquisition based on current parameters.

- Parameters – None.
- Range – None.
- Initial Values – None.
- Syntax - "GO"

3.6.3 Stop – Stop Image Acquisition

Stop - Stops any image acquisition in progress.

- Parameters – None.
- Range – None.
- Initial Values – None.
- Syntax - "STOP"

3.7 General Setup and Configuration Commands

The General Setup and Configuration Commands provide the methods for setting up the Array Controller for acquisitions. These commands prepare the Array Controller for an acquisition and should be used before the 'Go' command is executed. This section specifies the General Setup and Configuration Commands.

3.7.1 ArcMode - Set Exposure Mode

ArcMode - Selects the type of readout scheme for the acquisition, either ARC_S or ARC_D.

- Parameters – mode
- Range - mode: ARC_S or ARC_D.
  - o In ARC_S (arc single) mode the system will read out only signal images.
  - o In ARC_D (arc double) mode the system will read out both pedestal and signal images and will subtract off the pedestal image from the signal image to create the final image.
- Initial Values: On initial start up, ArcMode will be set to ARC_D.
- Syntax - "ARCMODE {arc_s | arc_d}"

3.7.2 Array – Set Subarray Size

Array - Set the size and location of the subarray. This subarray window determines which pixels are read out of the detector. Note that this command is only relevant if the ArrayMode has been set to subarray.

- Parameters - x, y, wid, hgt
- Range –
  - o x: 0 - 448. Must be some multiple of 64
  - o y: 0 - 448. Must be some multiple of 64
  - o wid: >= 64. Must be multiple of 64. The sum of the wid and x must be less than 512.
  - o hgt: >= 64. Must be multiple of 64. The sum of the hgt and y must be less than 512.
- Initial Values:
  - o On initial start up, system will be in FULLARRAY mode, and will process the entire image (1024x1024 pixels).
  - o On switching to SUBARRAY mode, initial values will be x = 0, y = 0, wid = 512, hgt = 512.
- Syntax - "ARRAY {x} {y} {wid} {hgt}"

3.7.3 ArrayMode – Set Array Mode (Full, Subarray)

ArrayMode - Set the system to either capture full images or to use the subarray parameters to capture sub-images.

- Parameters – mode
- Range - mode: full or sub.
  - o If mode=full the system captures data from the entire detector.
  - o If mode=sub the system captures data from only the part indicated by the subarray window parameters (set by the 'ARRAY' command).
- Initial Values: On initial start up, system will be in Full array mode.
- Syntax - "ARRAYMODE {full | sub}"

3.7.4 AutoSave – Set Save/Discard Data

AutoSave - Determines whether the data is saved to storage on the Pixel Server or is discarded.

- Parameters – a
- Range - a: on or off.
  - o If a=on, data is stored on the Pixel Server.
  - o If a=off, data is discarded at the Pixel Server, though system may be set to send images to the DV.
- Initial Values: On initial start up, AutoSave will be on.
- Syntax - "AUTOSAVE {on | off}"

3.7.5 BgrEnable – Background Reset Enable

BgrEnable – Enables and disables Background Resets.

- Parameters – mode
- Range - mode: on or off.
  - If mode=on the system performs BGR when the system is idle.
  - If mode=off the system does not perform BGR when system is idle.
- Initial Values: On initial start up, BGR is set to on.
- Syntax - "BGRENABLE {on | off}"

This command is password protected. See Section 3.7.15 for the engineering password command.

3.7.6 BgrMs – Set the BGR MS Parameter

BgrMs – Sets the BGR MS parameter. See Section 3.1.1 for an explanation.

- Parameters – time
- Range - time: The BGR MS time in milliseconds.
- Initial Values: On initial start up, time is 10 milliseconds.
- Syntax - "BGRMS {time}"

This command is password protected. See Section 3.7.15 for the engineering password command.

3.7.7 BgrMinMs – Set the BGR Min.MS Parameter

BgrMinMs - Sets the BGR Min.MS parameter. See Section 3.1.1 for an explanation.

- Parameters – time
- Range - time: The BGR Min.MS time in milliseconds.
- Initial Values: On initial start up, time is 10 milliseconds.
- Syntax - "BGRMINMS {time}"

This command is password protected. See Section 3.7.15 for the engineering password command.

3.7.8 BgrNs – Set the BGR NS Parameter

BgrNs - Sets the BGR NS parameter. See Section 3.1.1 for an explanation.

- Parameters – time
- Range - time: The BGR NS time in milliseconds.
- Initial Values: On initial start up, time is 10 nanosecs.
- Syntax - "BGRNS {time}"

This command is password protected. See Section 3.7.15 for the engineering password command.

3.7.9 CamMode – Set Basic/Streaming Camera Mode

CamMode - Specifies the operation mode of the system.

- Parameters – mode
- Range - mode: Basic or Streaming.
  - In BASIC mode the system takes individual images. Note that individual images may be made up of multiple samples, such as if taking multiple non-destructive reads or coadding multiple samples into an individual image.
  - In STREAMING (aka MOVIE) mode, the system continuously takes images. It continues to do so until the 'Stop' command is executed.
- Initial Values: On initial start up, CamMode is set to BASIC mode.
- Syntax - "CAMMODE {basic | streaming}"

3.7.10 ClkBiasVGGCL – Set VGGCL Bias Voltage

ClkBiasVGGCL - Sets the VGGCL clock bias voltage.

- Parameters – voltage
- Range - voltage: -3.15 to -3.75 Volts
- Initial Values - On initial start up, all clock bias voltages are set to -3.75 Volts.
- Syntax - "CLKBIASVGGCL {voltage}"

This command is password protected. See Section 3.7.15 for the engineering password command.

3.7.11 ClkBiasVDETL – Set VDET Bias Voltage

ClkBiasVDETL - Sets the VDET clock bias voltage.

- Parameters – voltage
- Range - voltage: -3.15 to -3.75 Volts.
- Initial Values - On in initial start up, all clock bias voltages are set to -3.75 Volts.
- Syntax - "CLKBIASVDET {voltage}"

This command is password protected. See Section 3.7.15 for the engineering password command.

Note that VDET is also referred to as VDETCOM. **It is absolutely critical that VDETCOM be set to a more positive voltage than VDDUC or the array can be critically damaged**.

3.7.12 Coadd – Set Coadd Count

Coadd - Sets the system coadd count.

- Parameters - num: the number of coadds to perform to generate an image.
- Range - num: num >= 0.
    - **Note**: An excessively high number of coadds could create an image that would exceed the 32 bit range of the integer based storage buffers.
- Initial Values - On start up, the coadd count is 1.
- Syntax - "COADD {num}"

3.7.13 Cycles – Set Cycle Count

Cycles - Sets the number of cycles in BASIC mode. This is the number of times to repeat the currently set up procedure.

- Parameters – count
- Range - count:
    - Applicable to Basic Mode image acquisitions only.
    - Sets the number of times to repeat the image acquisition procedure based on the current settings.
- Initial Values - On start up, the cycle count is 1.
- Syntax - "CYCLES {count}"

3.7.14 DoFastMode – Select Fast or Slow Clocking

DoFastMode - Select and use either the Fast or Slow clocking pattern sets.

- Parameters – mode
- Range - mode: on or off.
    - If on, the system will reload the clocking patterns, using the Fast clocking pattern set to generate images. The Fast clocking pattern set has been optimized for fast acquisition of images within a tolerable noise range.
    - If off, the system will reload the clocking patterns, using the Slow clocking pattern set to generate images. The Slow clocking pattern set has been optimized for minimized noise.
- Initial Values - On start up, DoFastMode is off, and is using the Slow clocking pattern set.
- Syntax - "DOFASTMODE {on | off}"

3.7.15 EPassword – Send Engineering Password

EPassword - Send the engineering password.

- Parameters – pass
- Range - pass: This is the engineering password, it's a text string. Once sent and verified, the interface has access to engineering functionality, such as modifying BGR and Clock Bias Voltage parameters.
- Initial Value - On start up, the interface does not have access to engineering restricted functionality
- Syntax - "EPASSWORD {pass}"

3.7.16 FitsComment – Set the FITS Comment Header

FitsComment - Sets the comment header entry for all FITS files created.

- Parameters – comment
- Range - comment: Comment is a text string.
- Initial Values - On start up, comment is "".
- Syntax - "FITSCOMMENT {comment}"

3.7.17 FitsFilename – Set FITS Filename

FitsFilename - Sets the base filename for all FITS files created and stored.

- Parameters – name
- Range - name: The name of a file can be anything, but should end with a ".fit" file type extension. Filenames are appended with incrementing file numbers, which increment internal to the Pixel Server system as it stores files to disk, so the resulting filename will be in the format of "somefile.XXX.fit", with XXX being some number.
- Initial Values - On start up, the filename is "somefile.fit".
- Syntax - "FITSFILENAME {name}"

3.7.18 FitsFilenumber – Set FITS File Number

FitsFilenumber - Sets the base file number for all FITS files created and stored.

- Parameters – num
- Range - num: This is the base file number appended to the filename. Users can enter a base filename, then set the base file number, and begin acquiring images. As each image is stored, the file number is incremented, so images can be acquired sequentially without needing to rename each on individually.
- Initial Value:
- Syntax - "FITSFILENUMBER {num}"

3.7.19 ITime – Set Integration Time

ITime - Sets the integration time. Integration time is made up of two factors, the time to clock out the pedestal image (assuming the system is in ARC_D mode, otherwise this time is 0) and the wait time set in the electronics.

- Parameters – time
- Range - time: time must be greater than 0 and must be greater than the Pedestal time.
- Initial Values - time is initially set to the pedestal time plus the minimum integration time.
- Syntax - "ITIME {time}"

3.7.20 NDR – Set Non-Destructive Read Count

NDR - Sets the NDR count, the number of non-destructive reads to perform per acquisition.

- Parameters – count
- Range - count: count must be greater than 0, less than 20.
- Initial Values - count is initially set to 1 at start up.
- Syntax – “NDR {count}”

3.7.21 SlowCnt – Set Slow Count

SlowCnt - Changing the slow count changes the speed at which data is clocked out of the detector. It increases the duration of the amount of time each pattern entry is held at the detector.

- Parameters – count
- Range - count: Must be between 0 and 400.
- Initial Values - count is initially 0.
- Syntax – "SLOWCNT {count}"

3.7.22 DestBuf – Set the Image Destination Buffer

DestBuf - Sets the destination buffer to which an image is buffered.

- Parameters – bufsel
- Range - bufsel: There are 8 coadd buffers to choose from, numbered 0 to 7.
- Initial Values - At start up, buffer 0 is selected.
- Syntax - "DESTBUF {bufsel}"

3.7.23 FitsObject – Set FITS Object Header Entry

FitsObject - Set the FITS object header entry.

- Parameters – name
- Range - name: A text string. Typically this is the name or reference to the object viewed.
- Initial Values - At start up, name is "".
- Syntax - "FITSOBJECT {name}"

3.7.24 FitsObserver – Set FITS Observer Header Entry

FitsObserver - Set the FITS observer header entry.

- Parameters – name
- Range - name: A text string. Typically this is the name of the observer.
- Initial Values - At start up, name is "".
- Syntax - "FITSOBSERVER {name}"

3.7.25 RTVEnable – Enable High Speed Real Time Viewing

RTVEnable - Enable High Speed Real-Time viewing. This is accomplished by converting the image to 16 bits and binning from 1024X1024 pixels to 256X256 pixels.

- Parameters – enable
- Range - enable: on or off.
  - If enable=on real time viewing is activated, images are not saved to disk, but are binned down to 256X256 pixels, and are sent to DV.
  - If enable=off, image acquisition proceeds according to the current configuration.
- Initial Values - At start up, enable is off.
- Syntax - "RTVENABLE {on | off}"

3.7.26 SavePath – Set FITS File Path

SavePath - Sets the path to which FITS files are stored. Note, this does not create or modify any directory structure. The save directory must already exist.

- Parameters – path
- Range - path: A text string indicating the path to which data will be stored.
- Initial Values - At start up, path is set to "/home/nici/redstar3/data".
- Syntax - "SAVEPATH {path}"

3.7.27 SerialEnable – Enable Serial Communication
SerialEnable – Enables/disables communication from the IC over Ethernet and through the remote serial connections.

- Parameters – enable
- Range - enable: on or off.
  - If enable=on the IC system will attempt to connect to the two DigiPort channels via socket connection.
  - If enable=off, the IC system will disconnect from the DigiPort channels.
- Initial Values - At start up, enable is set to off.
- Syntax - "SERIALENABLE {on | off}"

3.7.28 ReadSerial1 – Read Serial Port 1
ReadSerial1 - Read a string from serial port 1.

- Parameters – string
- Range - string: A text string.
- Initial Values - none.
- Syntax - "READSERIAL1 {string}"

3.7.29 ReadSerial2 – Read Serial Port 2
ReadSerial2 - Read a string from serial port 2.

- Parameters – string
- Range - string: A text string.
- Initial Values - none.
- Syntax - "READSERIAL2 {string}"

3.7.30 WriteSerial1 – Write to Serial Port 1
WriteSerial1 - Write a string out of serial port 1.

- Parameters – string
- Range - string: A text string.
- Initial Values - none.
- Syntax - "WRITESERIAL1 {string}"

3.7.31 WriteSerial2 – Write to Serial Port 2
WriteSerial2 - Write a string out of serial port 2.

- Parameters – string
- Range - string: A text string.
- Initial Values - none.
- Syntax - "WRITESERIAL2 {string}"

3.7.32 TimestampEnable – Enable Timestamping
TimestampEnable - Turns timestamping on and off.

- Parameters – mode
- Range - mode: on or off.
  - If mode=on timestamping is enabled, and system will mark the FITS header with the time of image acquisition based on the NTP maintained time.
  - If mode=off the system will mark the FITS header only with the current system time.
- Initial Values - On start up, timestamping is off.
- Syntax - "TIMESTAMPENABLE {on | off}"

3.8 Data Viewer Commands

The Data Viewer Commands provide control and configuration of the Data Viewer (also known as Image Monitor).

3.8.1 DV1Enable – Enable Data Viewer 1

DV1Enable - Enables the sending of images to instance 1 of the DV.

- Parameters – enable
- Range - enable: on or off.
  - If enable=on images are forwarded to an instance of DV. The DV1 Hostname and DV1 Port number data are used to make the connection.
  - If enable=off images are processed and stored as specified by the General Setup and Configuration Commands.
- Initial Values - At start up, enable is off.
- Syntax - "DV1ENABLE {on | off}"

3.8.2 DV2Enable – Enable Data Viewer 2

DV2Enable - Enables the sending of images to instance 2 of the DV.

- Parameters – enable
- Range - enable: on or off
  - If enable=on images are forwarded to an instance of DV. The DV2 Hostname and DV2 Port number data are used to make the connection.
  - If enable=off images are processed and stored as specified by the General Setup and Configuration Commands.
- Initial Values - At start up, enable is off.
- Syntax - "DV2ENABLE {on | off}"

3.8.3 DV1Hostname – Set Hostname for Data Viewer 1

DV1Hostname - Sets the hostname parameter for the machine that the instance of DV indicated by DV1 is running on.

- Parameters – hn
- Range - hn: A text string. hn is the hostname that DV is running on.
- Initial Values - At start up, hn is set to "".
- Syntax - "DV1HOSTNAME {hn}"

3.8.4 DV2Hostname – Set Hostname for Data Viewer 2

DV2Hostname - Sets the hostname parameter for the machine that the instance of DV indicated by DV2 is running on.

- Parameters – hn
- Range - hn: A text string. hn is the hostname that DV is running on.
- Initial Values - At start up, hn is set to "".
- Syntax - "DV2HOSTNAME {hn}"

3.8.5 DV1Port – Set the Port Number for Data Viewer 1

DV1Port - Sets the port number parameter for the machine that the instance of DV indicated by DV1 is listening on.

- Parameters – pn
- Range - pn: A valid port number, the port number on which DV1 is listening.
- Initial Values - At start up, pn is set to 30123.
- Syntax - "DV1PORTNUM {pn}"

3.8.6 DV2Port – Set the Port Number for Data Viewer 2

DV2Port - Sets the port number parameter for the machine that the instance of DV indicated by DV2 is listening on.

- Parameters – pn
- Range - pn: A valid port number, the port number on which DV2 is listening.
- Initial Values - At start up, pn is set to 30123.
- Syntax - "DV2PORTNUM {pn}"

3.9 GUI

The Graphical User Interface for the Array Controller provides the capability to perform all observation functions and to set all configuration parameters. The GUI is divided into six windows. There are the Observation, Setup, Engineering, Macro, System, and FITS windows. Each of these windows' functionality is described below. Some of the settings have equivalent command line commands indicated in parentheses. Screen captures are provided that show the windows.

3.9.1 Common Functions

Each of the windows include a set of buttons for common operations. Here the buttons are listed with their associated function and their command line equivalent. These buttons are illustrated in Figure 6.

GO: Starts an observation. (GO)

STOP: Gracefully stops an observation after the current exposure. (STOP)

QUIT: Exits the GUI.

ABORT: Terminates the observation immediately. Not graceful.

CONNECT: Starts the IC software and connects the IC to its clients. (CONNECT)

There is also a command line and display of command history and status messages in each of the six GUI windows.

3.9.2 Observation Window

The Observation Window provides the functionality listed below. Command line equivalents are listed in the parentheses. A screen capture of the Observation Window is provided in Figure 6.

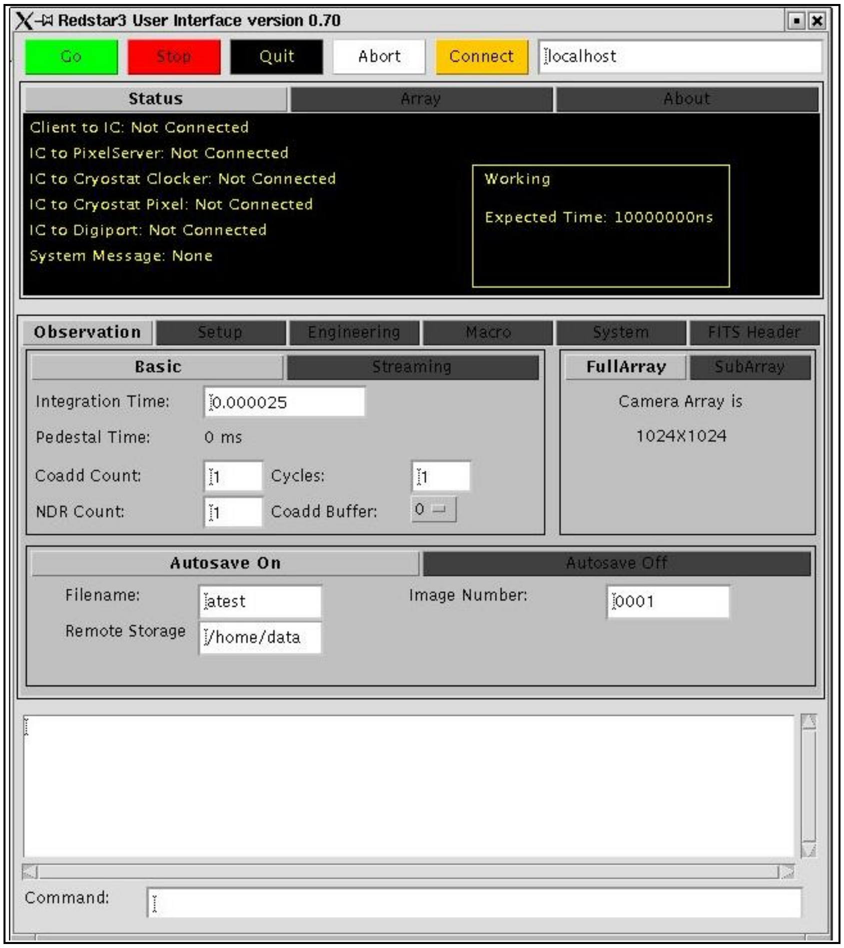


Figure 6 Screen Capture: GUI Observation Window

Observation Mode:

- Selects between image capture Basic and Streaming.
  - Basic: System performs one capture as specified by the configuration parameters.
    - Integration Time setting. (ITIME {time})
    - Pedestal Time monitoring.
    - CoAddition Count setting. (COADD {num})
    - NDR Count setting. (NDR {count})
    - Cycle Count setting. (CYCLES {count})
    - CoAdd Buffer selection. (DESTBUF {bufsel})
  - Streaming: System makes continuous image captures.
    - Integration Time setting. (ITIME {time})
    - Pedestal Time monitoring.
    - CoAddition Count setting. (COADD {num})
    - NDR Count setting. (NDR {count})
    - Cycle Count setting. (CYCLES {count})
    - CoAdd Buffer selection. (DESTBUF {bufsel})

Full Array/Subarray:
- Full Array: Displays the size of the full array, 1024 x 1024.
- Full Array/Subarray selection. (ARRAYMODE {full | sub})
- Subarray: Permits setting the size of the subarray. (ARRAY {x} {y} {wid} {hgt})

Autosave:
- Enable/Disable Autosave. (AUTOSAVE {on | off})
- Set FITS filename. (FITSFILENAME {name})
- Set FITS file storage path. (SAVEPATH {path})
- Set starting number for FITS image number. (FITSFILENUMBER {num})

3.9.3 Setup Window

The Setup Window provides the functionality listed below. Command line equivalents are listed in the parentheses. A screen capture of the Setup Window is provided in Figure 7.

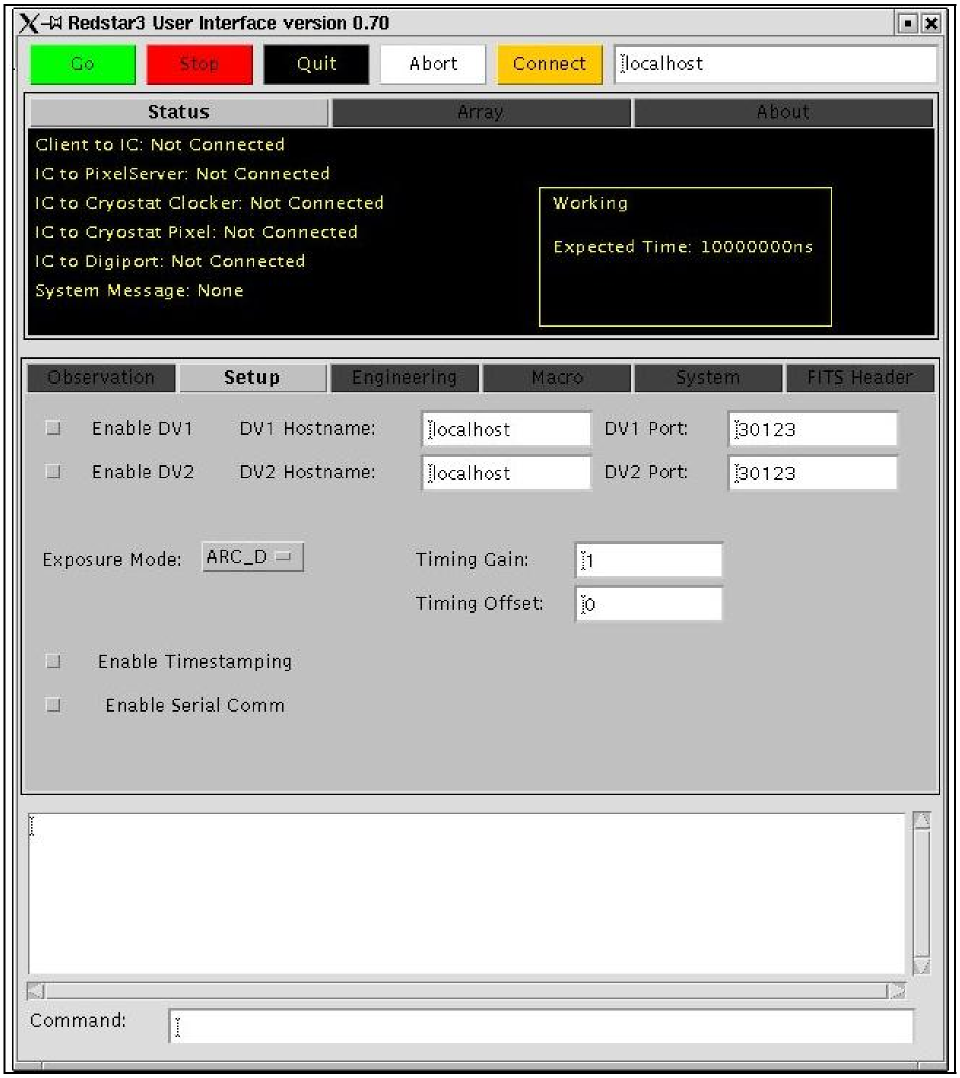


Figure 7 Screen Capture: GUI Observation Window

Image Monitor (DV, Data Viewer):

- Enable/Disable DV (DV1ENABLE {on | off}, DV2ENABLE {on | off})
- Set Data Viewer Hostname. (DV1HOSTNAME {hn}, DV2HOSTNAME {hn})
- Set DV Port. (DV1PORTNUM {pn}, DV2PORTNUM {pn})

Exposure Mode:

- Select Exposure Mode drop down box. (ARCMODE {arc_s | arc_d})
- Timing Gain
- Timing Offset

Timestamping: Enable/Disable Checkbox. (TIMESTAMPENABLE {on | off})

Serial Communication: Enable/Disable Checkbox. (SERIALENABLE {on | off})

3.9.4 Engineering Window

The Engineering Window provides password protected control over some critical Array Controller functions, listed below. Command line equivalents are listed in the parentheses. A screen capture of the Engineering Window is provided in Figure 8.

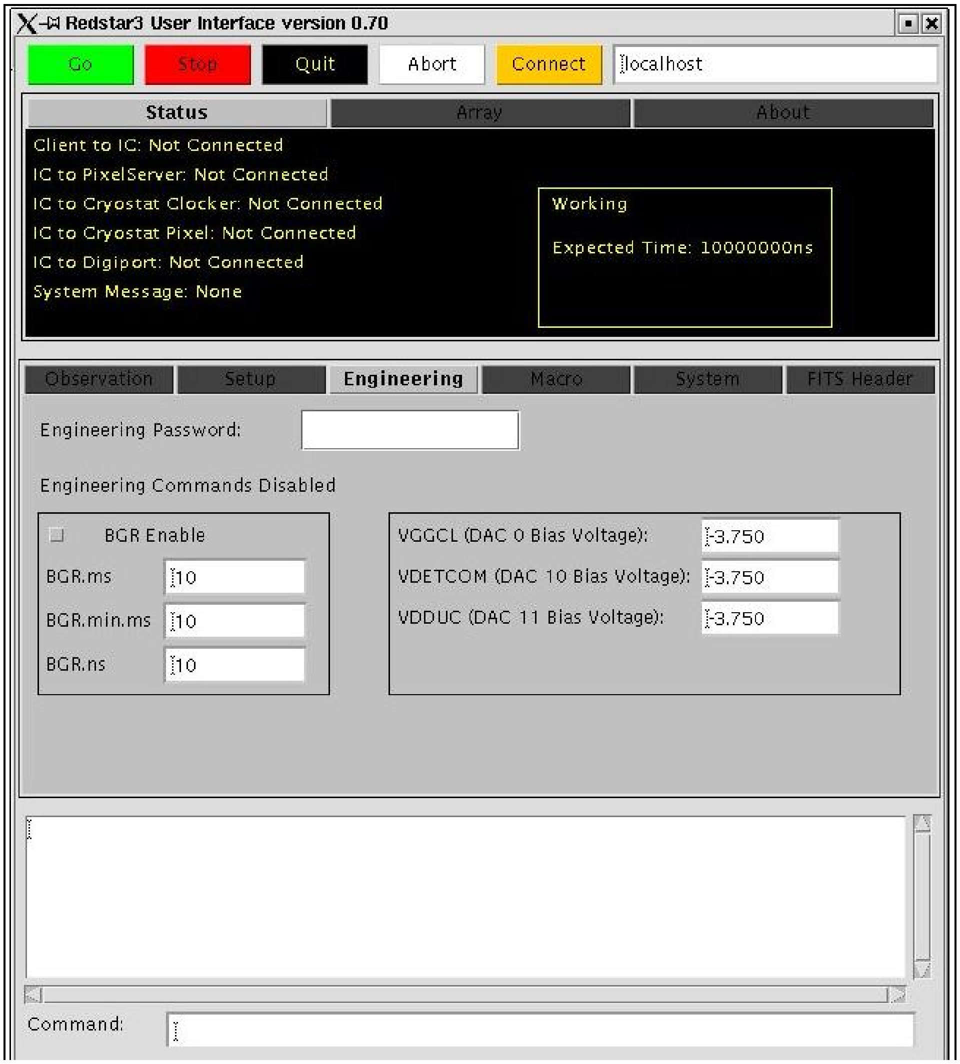


Figure 8 Screen Capture: GUI Engineering Window

Background Resets (BGR):

- Enable/Disable Checkbox. (BGRENABLE {on | off})
- BGR.ms (BGRMS {time})
- BGR.min.ms (BGRMINMS {time})
- BGR.ns (BGRNS {time})

Array Bias Voltages:

- Set array bias VGGCL. (CLKBIASVGGCL {voltage})
- Set array bias VDETCOM. (CLKBIASVDET {voltage})
- Set array bias VDDUC.
- **It is absolutely critical that VDETCOM be set to a more positive voltage than VDDUC or the array can be critically damaged**.

3.9.5 Macro Window

The Macro Window provides the capability to define, edit, and execute macros. A screen capture of the Macro Window is provided in Figure 9.

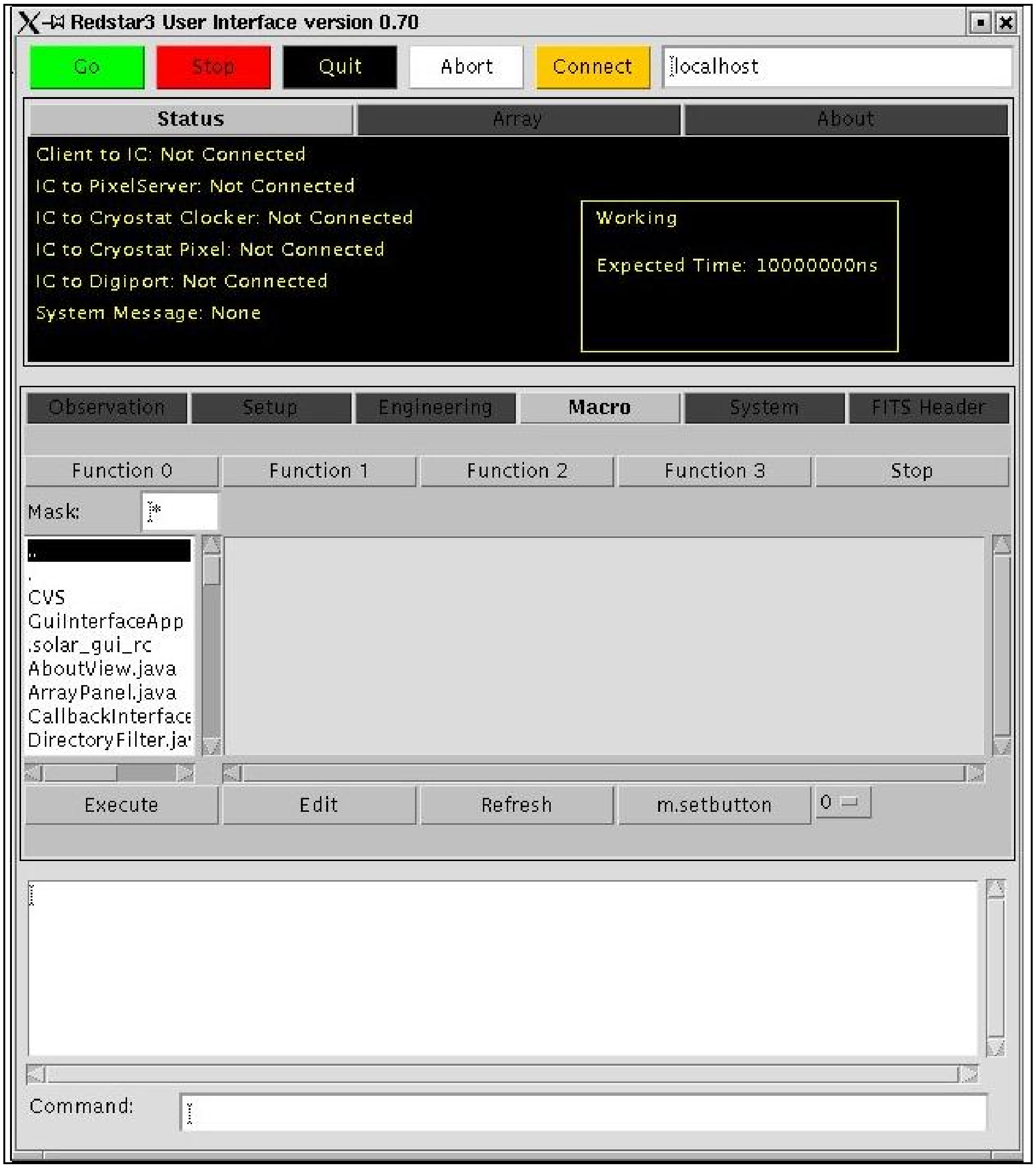


Figure 9 Screen Capture: GUI Macro Window

- **Execute**: Executes the selected macro.
- **Edit**: Opens a text editor for editing the selected macro.
- **Refresh**: Refreshes the list of macros.
- **m.setbutton**: Assigns the selected macro to the Function # button where # is the number selected in the drop down menu to the right of the m.setbutton button.

3.9.6 System Window

The System Window provides the functionality listed below. Generally these parameters need not be modified by the user. Command line equivalents are listed in the parentheses. A screen capture of the System Window is provided in Figure 10.

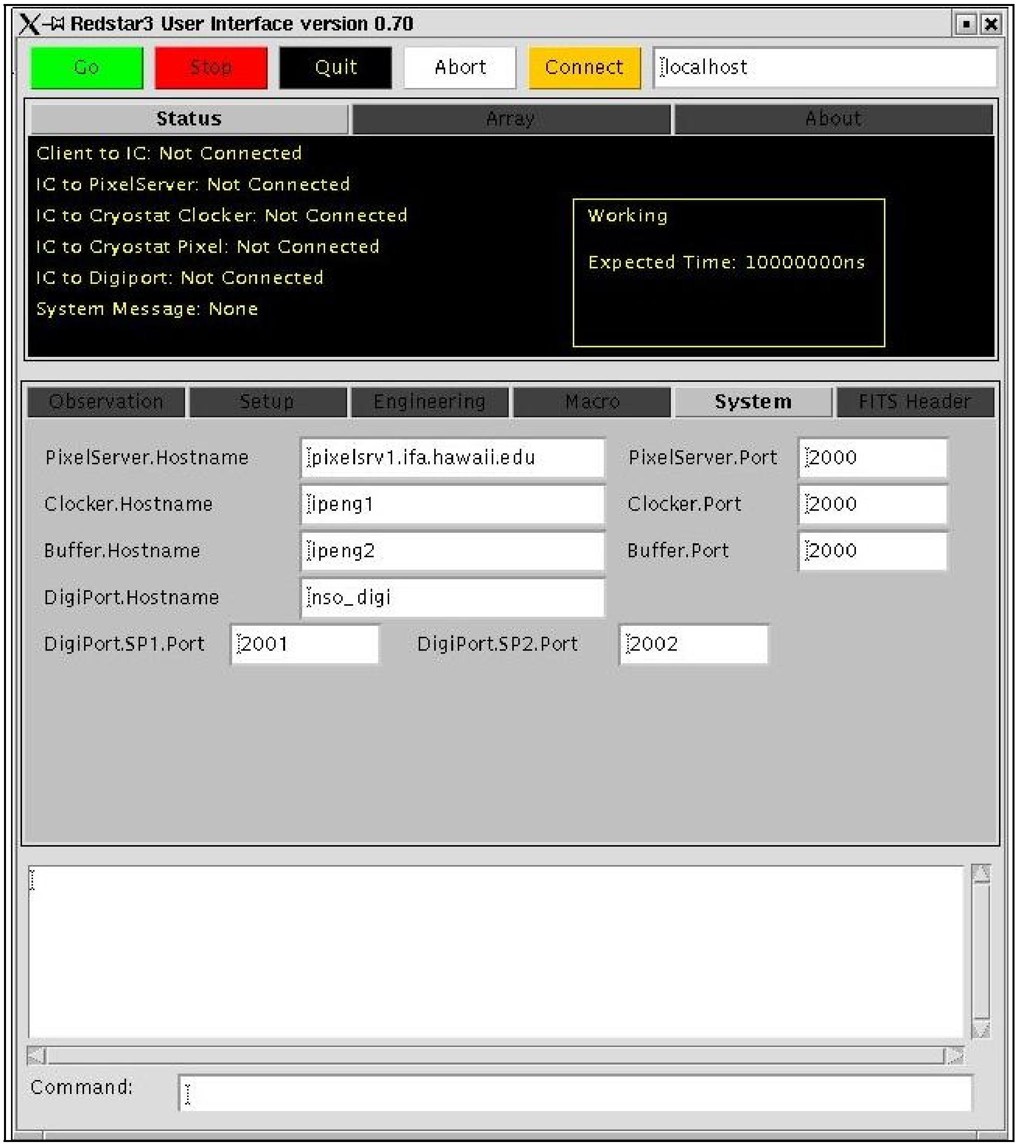


Figure 10 Screen Capture: GUI System Window

Pixel Server:
- Set the Pixel Server hostname. (PSRVHOSTNAME {hn})
- Set the Pixel Server port. (PSRVPORTNUM {pn})

Clocker:
- Set the Clocker hostname. (CLOCKERHOSTNAME {hn})
- Set the Clocker port. (CLOCKERPORTNUM {pn})

Catcher:
- Set the Catcher hostname. (PIXELHOSTNAME {hn})
- Set the Catcher port. (PIXELPORTNUM {pn})

DigiPort:

- Set the DigiPort Hostname.
- Set DigiPort Channel 1 Port. (DIGIPORTNUM {pn1} {pn2})
- Set DigiPort Channel 2 Port. (DIGIPORTNUM {pn1} {pn2})

3.9.7 FITS Header Window

The FITS Header Window provides control over what information is inserted into FITS file header for the observation. The functions that can be set in the FITS window are listed below. The command line equivalents are listed in parentheses. A screen capture of the FITS Header Window is provided in Figure 11.

FITS Header:

- Set FITS Object Header Entry. (FITSOBJECT {name})
- Set FITS Observer Header Entry. (FITSOBSERVER {name})
- Set FITS Comment Header Entry. (FITSCOMMENT {comment})
- Set FITS Pos. Angle Header Entry.
- Set FITS Plate Scale Header Entry.

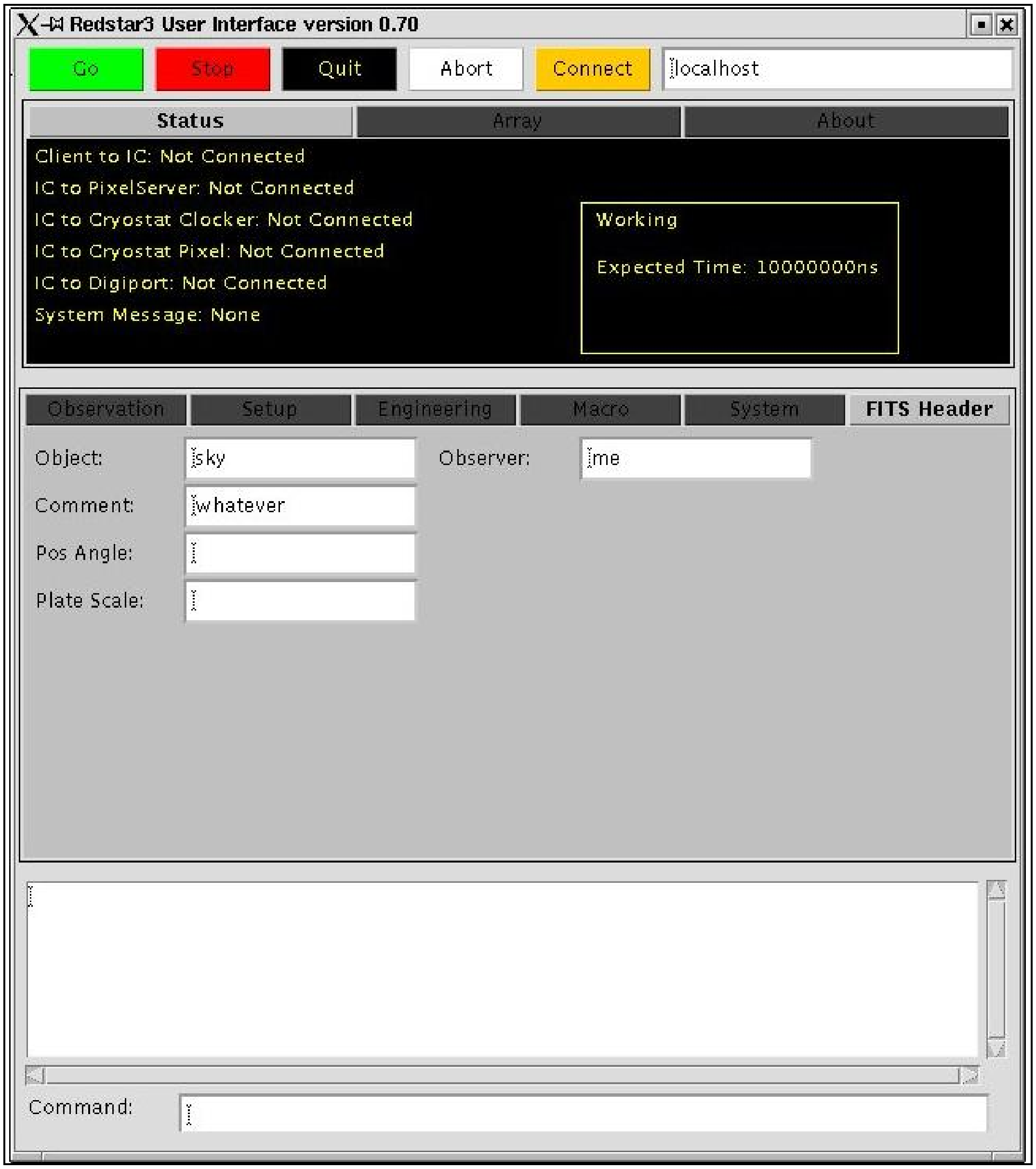


Figure 11 Screen Capture: GUI FITS Header Window

4 Acronyms and Definitions

Aladdin	A 1024 x 1024 In:Sb 1-5 micron focal plane array.
BGR	Background Reset.
DV	Data Viewer, also known as Image Monitor
FCRYO2	Fiber Cryostat Mounted subassembly version 2, a board in the Redstar3
FPA	Focal Plane Array
GUI	Graphical User Interface
IC	Instrument Controller
LAN	Local Area Network
LSB	Least Significant Bit
MKIR	Mauna Kea Infrared
NDR	Nondestructive Read
Redstar3	MKIR's Infrared Array Controller
ROI	Region Of Interest, also called Subarray