PIRT 1280SciCam
1280x1024x12um Cooled InGaAs
Science Camera
1280SC-12-A1-InGaAs-1.7
Interface Control Document

InGaAs linear and area camera products are subject to the Export Administration Regulations (EAR) ECCN 6A003.b.4.a and require an export license from the U.S. Commerce Department.

1. Overview

The 1280SciCam (1280SC-12-A1-InGaAs-1.7) is a cooled scientific camera using Princeton Infrared Technologies 1280x1024x12um pitch InGaAs Focal Plane Array. This focal plane array is a hybrid device with a detector material made of $In_{0.53}Ga_{0.47}As$ (pin) mounted on top of silicon read out integrated circuit (ROIC). The detector turns the light into an electrical signal and this signal is gathered and digitized in the ROIC. This imaging array is backside illuminated, meaning the light comes in the side opposite of where the photodiodes are formed. This can be seen in Figure 1. The PIRT1280A1-12 is an InGaAs detector array that has had its substrate removed thus allowing a larger visible response than most InGaAs focal plane arrays. Figure 2 is a typical Quantum Efficiency curve for this device. This 12um pitch array has >99% fill factor. It also has low read noise <40e- allowing for low light level operation. The camera operates the focal plane array by providing clocks, voltages for signal and ground. It also communicates with the focal plane array to control bias to the photodiodes as well region of interest windows on the array. Finally the camera takes the multiple 14 bit digital outputs from the camera and recombines them into the correct sequence and transmits them out a Camera Link standard adapter. The camera controls the array temperate and has the ability to store non-uniformity corrections and apply them to the data stream if desired by the user.

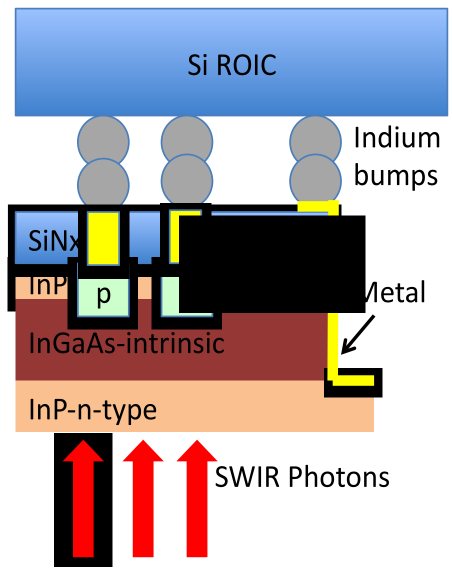


Figure 1. InGaAs imaging array attached so Si ROIC circuit.

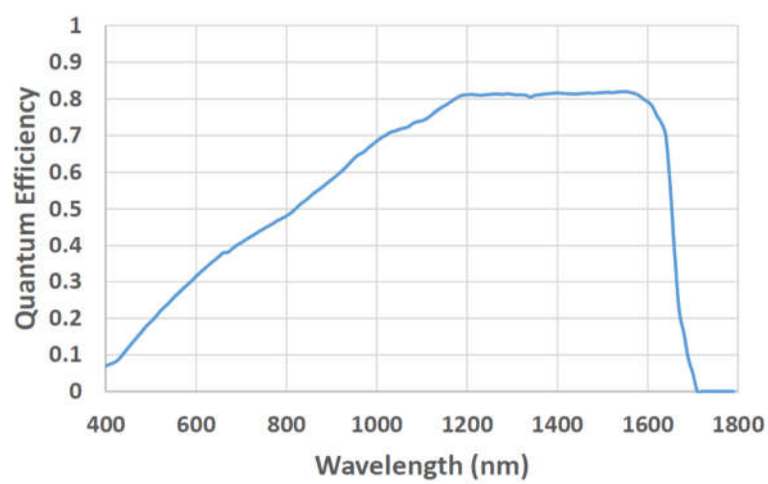


Figure 2. Typical quantum efficiency curve for PIRT InGaAs arrays.

2. Communication Protocol

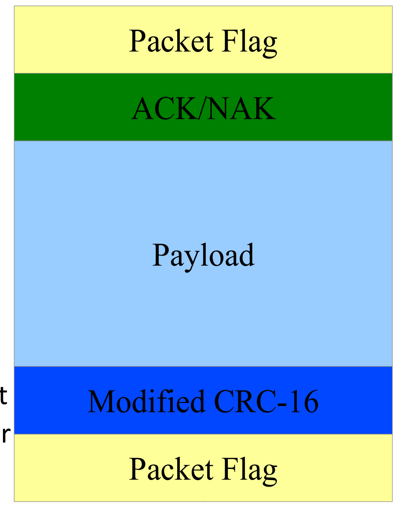
2.1. Overview

The camera communication protocol consists of two protocol layers: the data link and application layers. For wire-like communication interfaces that don't have a defined data-link level protocol (such as the Camera Link serial interface), the Science Camera communicates using a simple sliding window protocol. An application layer protocol is implemented to define the data payloads sent between the camera and the user system. The data link layer protocol guarantees that data sent to either system will arrive without errors, while the application layer protocol helps either system manage the incoming payload data.

2.2. Data-Link Layer Protocol

The data link layer protocol provides a high-probability of error-free data transfer through the use of a simple sliding window protocol. As this link is expected to be relatively inactive, the sliding window in this case has only a single frame. This decreases the resource burden on the camera system, but still provides a means for reliable communications between the host and camera system.

The general case data-link packet is illustrated below in Drawing 1. The packet is comprised of 5 basic fields: the *packet flag*, the *ACK/NAK*, the *payload*, and the *CRC-16*.



Drawing 1: Data-link protocol packet

The *packet flag* is a one octet (byte) field, denoted by the character 0x3E not preceded by the escape character, which is 0x5C. Because the packet flag denotes the beginning and the end of the packet, there is no need for a packet size field. The maximum packet size the camera can receive is 16383 bytes. The escape character allows the packet flag to be used in the payload without interrupting the packet. To use the escape character in the data payload, it must also be escaped (0x5C 0x5C).

The *ACK/NAK* is an octet (byte) field that denotes whether the previous frame has been successfully received by the receiving device (either the camera or PC). The ACK (0x20) byte is only used when the datalink layer is streaming data to the computer (see file transfer).   The NAK (0xA0) byte is used when an improperly formed packet is received. If this is not an ACK/NAK packet, the field remains but contains an null (0) value.

The packet flag and ACK/NAK comprise a static header at the beginning of each frame. The payload immediately follows this header, and is made up of an arbitrary number of octets. Any packet flags or escape characters occurring in this region must be escaped to avoid losing data.

The *modified CRC-16* is a two octet (byte) field providing a checksum for the packet. This is a modified CRC-16 because the generating polynomial is 0x755B (Koopman notation: 0xBAAD), or alternatively:

$$x^{16} + x^{14} + x^{13} + x^{12} + x^{10} + x^{8} + x^{6} + x^{4} + x^{3} + x + 1$$

This polynomial was chosen based upon the paper “Cyclic Redundancy Code (CRC) Polynomial Selection for Embedded Networks” by Koopman and Chakravarty. While one octet CRC solutions exist to detect 1 bit errors in large packet sizes, the two octet CRC provides detection of up to 3 bit errors for packets sized at 2048 bits. Thus, while there is no maximum packet size, the recommended maximum packet size is 254 bytes. The CRC calculation starts at the ACK/NAK field and ends at the byte immediately preceding the CRC field. Escape characters are added AFTER the CRC is calculated. Any packet flags or escape characters that occur in the CRC field must also be escaped.

2.3. Data-Link Layer Protocol Rules

Because the possibility exists for packet flags that have not been escaped (causing the transmitter and receiver to lose synchronization), there are a couple of additional features to this protocol.

1. The receiver must respond to every packet. (Single frame sliding window requirement.)
2. The packet is naturally ACKed if the sender receives a response prior to timing out. This indicates that most return (or what would naturally be considered ACK packets) do not have their ACK field set, but are rather responses to the given command. On the other hand, during file transmission, all packets sent are ACKed.
3. If the receiver receives a malformed packet, then it should assume loss of sync and flush the buffer and ignore next packet flag. A situation like this would occur if an unescaped packet flag occurred in the payload, causing the packet to suddenly end, and to be marked a bad packet for failing CRC. If a malformed packet is received, then it is assumed that the receiver is on an odd packet flag, and thus one packet flag must be ignored to return to the correct flag sequence. Ignored packets will timeout, and be resent by the transmitter.
4. ACK/NAK packets may contain command response data, and command response ACK packets does not need to continue the ACK chain. As an example, the host may send a packet requesting a current value of some parameter on the camera. In the ACK response to this request, the payload may contain the parameter value. In this case, the host does NOT need to send an acknowledgment for the receipt of the parameter value. However, if the CRC is invalid, then a negative response would be required to resend the correct data.

Because this is a 1-bit sliding window protocol, if the next packet sent by the host is NOT a negative acknowledgment, then the parameter value sent in the ACK packet was accepted by the transmitter.

The camera NEVER initiates communication using this data link protocol.

When the camera is turned on, it ignores all data on the bus until a packet flag is received.

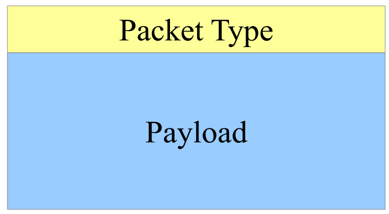
A RESET communications command is four (4) consecutive packet flags in a row. The reset clears the internal packet FIFO and any pending commands, as well as resets the sliding window by setting the frame number to zero. It is recommended to do this at the beginning of camera operations or if the camera returns an excessive number of negative acknowledgments.

2.4. Application Layer Protocol

The network layer protocol is a simple protocol used to deliver and transmit commands and files between the camera system and the host. Since this application protocol may be transmitted over various communication standards, the only limit on the size of the packet is that it must be received as a single coherent entity by the camera system.

The application layer protocol has two modes: command transfer mode and file transfer mode. In command transfer mode the application layer host transmits one or more commands to the camera system. The camera system executes those commands and responds appropriately according to the command sequence. In the file transfer mode, the application layer host either transmits or receives a file from the camera. This allows a more efficient means of transferring a large amount of data over fewer packets.

The difference between the two modes is determined by the type field (single octet) of the protocol, which is the only overhead the protocol requires.



Drawing 2: Application Layer Packet

2.5. Command Transfer Mode

The command transfer mode is extremely simple. There is only a single overhead byte per command, the command header (0xFF), which denotes that the following sequence of octets (bytes) is a command. The interpreter then scans until another command header is found. There must be one command header per command in the command transfer mode packet. All octets (bytes) between two command headers are assumed to be a single command. This indicates that all 0xFF values found internal to the command must be escaped with the escape character (0x5C). The escape character must also be escaped if found as part of a command.

Any number of variable-length commands may be placed into a single command transfer mode packet. No checksums are calculated for individual commands, since the data link layer protocol provides validation at a lower level. It should also be noted that each variable-length command sequence is attempted individually. Thus if one command is invalid, it does not necessarily affect following commands.

ENDIANNESS: Commands that require integer or float arguments are sent LSB first.

2.6. File Transfer Mode

A file is transferred to the camera by sending the “file write” command with the destination (absolute path on the camera) as the parameter. The “file write” command will return an error if there is currently a file open for reading or writing, or if the path is badly formed. If the “file write” command does not return an error, the camera is ready to receive the file data. At the application layer, file data is sent by sending packets with the packet type field set to the file flag (0xC0) and the file data as the payload, file flags (0xC0) contained in the payload should NOT be escaped. The payload may be arbitrarily sized, although it cannot exceed the maximum data link layer packet size. The file data received from each packet is linked together in RAM on the camera. After all data is sent, the “file close” command should be sent to write out the file to the file system.

In order to deal with slow transfer speeds over the Camera Link, files can be compressed before being transferred to the camera. If the path given with the “file write” command ends with “.bz2,” the camera assumes the file being sent is a bzip2 file. When the "file close" command is received, the camera will write out the bzip2 file, and then decompress it. The bzip2 file will be removed from the camera and the decompressed file will have the ".bz2" extension removed.

A file is received by sending the "file read" command with the path (absolute path on the camera) of the file to read as the parameter. The "file read" command will return an error if there is currently a file open for reading or writing, or if the file does not exist. If the "file read" command is successful, the camera will begin sending file packets. The application layer file packets will have the packet type field set to the file flag (0xC0) and the payload will contain data from the file being read. The next file packet will not be sent until the user sends an ACK in response to the received file packet. If the packet type field is not set to the file flag (0xC0) then all data has been sent from the camera. The "file close" command should be sent so that future file reads and writes may take place.

Compression is not currently supported for file reads.

2.7. Example Packets

The following is a series of example packets that demonstrate the structure of correctly formed packets using the IRC packet protocol.

COLOR CHART
RED – PACKET FLAG ORANGE – CRC GREEN – ACK/NACK BLUE – COMMAND FLAG PURPLE – COMMAND BLACK – DATA

```
COMMAND: 00 0D – Retrieve Serial Number

DATA: None

SEND:    3E 00 FF 00 0D 8E 85 3E

RECEIVE: 3E 00 FF 00 0D 31 33 39 33 39 39 00 E9 4F 3E


COMMAND: 05 16 – Set working directory

DATA: "/flash/"

SEND: 3E 00 FF 05 16 2F 66 6C 61 73 68 2F 00 D9 25 3E

RECEIVE: 3E 00 FF 05 16 A0 00 07 95 3E
```

```
COMMAND: 10 01 – Read VPOS bias
DATA: None
SEND: 3E 00 FF 10 01 A6 23 3E
RECEIVE: 3E 00 FF 10 01 3D 0A 57 40 9F DB 3E (3.36)

COMMAND: 10 64 – Set Window Column Size
DATA: 640 (80 02 00 00)
SEND: 3E 00 FF 10 64 80 02 00 00 BF 54 3E
RECEIVE: 3E 00 FF 10 64 80 02 00 00 BF 54 3E
```

3. 1280SciCam-Command List

3.1. Overview

The following command list provides a list of the commands and expected responses from the 1280SciCam. To verify the operation of these commands, it is recommended to using the “IRCommApp” application that is bundled with WinIRC.

NOTE: All values in the tables in the following sections are in hexadecimal. Floats and integers are sent LSB first. Strings are sent first character first and must be terminated with a null character.

3.2. Administrative Commands

3.2.1.1. Command: Reset Communications

Operation Code	00 04
Data	None
Response	None

This command resets the data link layer, destroying any unprocessed commands. This event is logged, and returns the operations code. Further command packets can be sent immediately after the response is received.

This command is also part of the "connection sequence."

3.2.1.2. Command: Revisions

Operation Code	00 0B
Data	Code to retrieve desired information: • 01 – returns the firmware version • 02 – returns the software version • 03 – returns the camera vendor • 04 – returns the camera family • 05 – returns the camera part number • 06 – returns the camera serial number • 07 – returns the camera FPA model
Response	00 0b string representing information desired OR the error code 0xe001 indicating invalid request code.

The revisions command provides a method for determining the current camera configuration. This event is logged, and returns both the operations code and some form of either string or error code response.

3.2.1.3. Command: Retrieve Serial Number

Operation Code	00 0D
Data	None
Response	00 0d string representing the serial number

This command just provides an alternative method for retrieving the serial number.

3.2.1.4. Command: Set Logging Level

Operation Code	00 20
Data	A single byte logging level code: • 00 – no logging • 01 – irregular logging of important events • 02 – default logging (logs all commands, recommended) • 03 – High (logs finer grain events)
Response	00 20 with either an error code 0xe001 to represent a bad

	level, or the level code sent

This command controls how much information is actually put into the log file. It is recommended to keep it at the default (regular) level, which logs all commands. This is particularly useful if there is an issue and the log file has to be pulled! Otherwise, re-creating the issue will be difficult, since the customer probably will not keep a log of his own events.

3.2.1.5. Command: Get Current Log

Operation Code	00 21
Data	None
Response	Initiates a file send event (from the camera) to send the current log file through the file transfer mode.

This command should not be entered into the IR Comm App. See further information in the chapter of common tasks.

3.2.1.6. Command: Write Log Event

Operation Code	00 22
Data	Null-terminated ASCII string to send to the log.
Response	00 22

The command allows the user to put an event directly into the log file to demark tasks. It allows commenting in the log for future readability, in case the log is going to be pulled.

3.2.1.7. Command: Get Previous Log

Operation Code	00 23
Data	None
Response	Initiates a file send event (from the camera) to send the previous log file through the file transfer mode.

This command should not be entered into the IR Comm App. See further information in the chapter of common tasks.

3.2.1.8. Command: Logging Enable

Operation Code	00 24
Data	Character of value 0/1 to determine if logging is off/on
Response	None

It is highly recommended to leave logging on, as turning it off will make debugging next to impossible.

3.2.1.9. Command: Get Logging Enable State

Operation Code	00 25
Data	None
Response	Returns a value of 1 if logging is enabled, 0 if disabled

It is highly recommended to leave logging on, as turning it off will make debugging next to impossible.

3.2.1.10. Command: Save Log to Flash

Operation Code	00 26
Data	None
Response	None

The Save Log to Flash command forces the current log to Flash. Currently the log may not be forced into writing to the Flash after ever entry. This command forces the Flash buffer to flash, and the data to actually commit to the hardware.

3.2.1.11. Command: Save Camera State

Operation Code	01 00
Data	<String> Filename of camera state to be saved. Filename must end in ".xml" and not contain a path
Response	A0 00 on successful completion. Otherwise, an error code will be returned: E0 08 – camera extension not ".xml" E0 09 – Null character missing at end of filename E0 0A – path instead of filename Other errors indicate XML library failures, see developer/code.

The Save Camera State command is useful because it allows the full camera state to be saved and retrieved by the Load Camera State at a later date. The only limit to the number of files that can be saved is the amount of storage space on the camera, which is currently about 495MB usable.

3.2.1.12. Command: Load Camera State

Operation Code	01 01
Data	<String> Filename of camera state to be loaded. Filename must end in ".xml" and not contain a path
Response	A0 00 on successful completion. Otherwise, an error code

	will be returned: E0 08 – camera extension not ".xml" E0 09 – Null character missing at end of filename E0 0A – path instead of filename Other errors indicate XML library failures, see developer/code.

The Load Camera State command allows the user to load a full camera state on demand. The State file is considered to be a full description of the camera, when saved from the Camera Head. Alternatively, the state file can be hand edited for toggling only system parameters that the user wants to uniquely modify. In this way, the "state" file is more like a script to automatically change a number of parameters in a single step.

3.2.1.13. Command: Set Startup State

Operation Code	01 02
Data	<String> Filename of camera state to be loaded at power on. Filename must end in ".xml" and not contain a path
Response	A0 00 on successful completion. Otherwise, an error code will be returned: E0 01 – Camera state file could not be opened (invalid) E0 02 – Camera state file given is not a valid state file E0 03 – Architecture file could not be opened E0 04 – Architecture file has been corrupted. E0 05 – Architecture file has been corrupted in a different way. E0 06 – Architecture file could not be written E0 07 – Camera state file is invalid. Other errors indicate XML library failures, see developer/code.

This command allows a different startup file to be loaded than what was originally setup. This allows the user to create his own unique state, then return to that state the next time the camera powers on.

Note that this command does NOT load the new state – it merely sets a camera parameter so that camera state file will be loaded in the future.

3.2.1.14. Command: Get Startup State

Operation Code	01 03
Data	None

Response	<String> of default state file name

This command returns the current default state (not necessarily the currently loaded state).

3.2.1.15. Command: Copy Default States

Operation Code	01 04
Data	None
Response	A0 00 on successful completion. Otherwise, an error code will be returned: E0 XY – basically indicates a failure to copy over 1 of 4 sets of files. The value X determines the type of default files: 1 - the architecture files 2 - camera state files 3 - the watchdog files 4 - the temperature calibration files. The value Y determines the type of error that occurred: 1 – could not open source 2 – could not open target 3 – could not write to target 4 – error during read of source 5 – error when closing source 6 – error when closing target 7 – error when syncing to flash Other errors indicate XML library failures, see developer/code.

Because of the flexibility allowed by the camera, it is quite easy for the user to get into an unresponsive video state. If this occurs, the final, and most drastic step would be to return to the default states, which is accomplished via this command.

3.2.1.16. Command: Set Bad Pixel Replacement Calibration File

Operation Code	01 06
Data	None

Response	A0 00 on successful completion. Otherwise, an error code will be returned: E0 0X – basically indicates a failure to set the BPR calibration file The value X determines the type of error that occurred: 1 – could not open source 2 – calibration file is invalid 3 – Architecture file could not be opened 4 – Architecture file has been corrupted 5 – Architecture file has been corrupted 6 – Architecture file could not be written 7 – calibration file is invalid 8 – poorly formed packet (usually invalid filename) 9 – Null character missing at end of filename Other errors indicate XML library failures, see developer/code.

This command sets the BPR calibration file. For most users, the factory setting will never need to be changed. The only time this changes is basically if the focalplane changes.

3.2.1.17. Command: Program Main Flash

Operation Code	03 00
Data	<String> absolute path to flash file
Response	<String> echoed path if successful else E0 01 00 00 – No null byte E0 02 00 00 – Path not absolute

This command will begin programming the flash. Once programming begins, the power to the camera must not be turned off until it is finished else it will have to be reprogrammed through JTAG. Use the Program Main Flash Status command to determine when programming is finished.

3.2.1.18. Command: Program Main Flash Status

Operation Code	03 01
Data	None

Response	<Integer> Between 0 and 100 indicates percent complete. Negative indicates an error.

This command will indicate error or progress in programming the flash. The program flash command will not fail if the command is formed correctly, so this command must be queried to check for errors, such as the file not existing. If there are no errors, this command will indicate the completion percentage, the flash is finished being programmed when this command returns 100. Programming the flash takes ~5 minutes.

3.2.1.19. Command: Kill Code

Operation Code	0F FF
Data	None
Response	None

This command effectively shuts down the camera, and in order to send further commands the user must power cycle. This command is mainly used during debugging, and should not be issued to the camera when in a production state.

3.3. File Commands

3.3.1.1. Command: File Write

Operation Code	05 10
Data	<String> Destination path of the file
Response	E0 02 – error code indicating a file is already open E0 04 – error code that path must begin with /flash or /ramfs E0 06 – error code indicating no path was provided A0 0A – successfully opened file handle

This command uses the file transfer section of the serial protocol to transfer files to the camera head such as NUC files and state files. Note that each file section is appended to the end of the open file until the file is expressly closed. Only one file handle may be open at a time.

Note that if the file ends in a ".bz2" extension, the camera assumes that the file has undergone bzip2 compression. In this case, the camera will attempt to decompress the file and write out the uncompressed file to the destination.

Note also that files may be sent to the /ramfs, which is a convenient way to store files only used for this camera session.

3.3.1.2. Command: File Read

Operation Code	05 11
Data	<String> Path of the file to read
Response	E0 02 – error code indicating a file is already open E0 04 – error code that path must begin with /flash or /ramfs E0 06 – error code indicating no path was provided E0 08 – error code indicating file unable to be opened A0 0A – successfully opened file handle

This command uses the file transfer section of the serial protocol to transfer files from the camera head such as NUC files and state files. Note that each file section is appended to the end of the open file until the file is expressly closed. Only one file handle may be open at a time.

Note that NO COMPRESSION is available in retrieving files FROM the camera.

3.3.1.3. Command: File Close

Operation Code	05 12
Data	None
Response	E0 02 – error code indicating no file is open A0 0A – successfully opened file handle

This command closes any open file, and writes out that file to the requested destination. In the case of compressed files, it decompresses and then writes out the file. IF FILES ARE REFUSING TO BE RETRIEVED, IT IS USUALLY BECAUSE OF A DANGLING FILE OPEN. MAKE SURE ALL FILES ARE CLOSED WITH THIS COMMAND.

3.3.1.4. Command: List Files Verbosely

Operation Code	05 13
Data	None
Response	<String> Filenames, size, and more in the current working directory E0 02 – error code indicating the working directory could not be opened

This is a verbose version of the list command, which allows the user to see the files in the directory, their permissions, and size. The dates/times are not logged by the file system, and should not be used.

3.3.1.5. Command: File Delete

Operation Code	05 14
Data	<String> Relative file path to remove
Response	E0 01 – error code indicating no null character at the end of the file path string E0 02 – error code due to parent directory in the path E0 03 – error code indication file path is not relative E0 04 – error invalid permissions to access E0 05 – error file does not exist E0 06 – error invalid permissions to delete E0 07 – error during removal A0 0A – successfully removed file

Note that the file path to send is relative, which means that the following commands (05 15 and 05 16) must be used to move the current camera path to the desired directory in order to remove the file.

3.3.1.6. Command: List Files

Operation Code	05 15
Data	None
Response	<String> Filenames in the current working directory E0 02 – error code indicating the working directory could not be opened

This command is useful to list all current files in the working directory. That way you can verify if the file you want to use is in the location you suspect.

3.3.1.7. Command: Set Working Directory

Operation Code	05 16
Data	<String> the absolute destination path to set the working directory.
Response	E0 01 – Error code because null string received E0 02 – Error code because cannot move to the parent directory of /flash or /ramfs E0 03 – Error code indicating path was not absolute E0 04 – Unable to change to a relative path E0 05 – Unable to change to absolute path

	E0 06 – Unable to resolve current directory E0 10 – No Null character found at the end A0 0A – Successful change of directory

This command is useful when working with files. It allows you to move the working directory for listing and deleting files.

3.3.1.8. Command: Get Working Directory

Operation Code	05 17
Data	None
Response	<String> current working directory E0 02/E0 01 – error code indicating the working directory could not be resolved

This command is useful to get the path of the working directory. That way you can verify if you are at the current path location.

3.3.1.9. Command: Make Directory

Operation Code	05 20
Data	<String> Relative path of directory to create
Response	E0 01 – error code indicating the path requested was not relative E0 02 – error code indicating the path already exists E0 03 – error code with system error message E0 05 – No NULL character found at the end of the path A0 0A – successful command/directory created

This command is useful if you wish to group a subset of files in a new directory. This could be done to group localized NUC files or state files.

3.3.1.10. Command: Remove Directory

Operation Code	05 21
Data	<String> Relative path of directory to remove
Response	E0 01 – error code indicating the path requested was not relative E0 02 – error code indicating the path is not a directory

	E0 03 – error code directory is not empty E0 04 – error code unable to remove directory system error. E0 05 – No NULL character found at the end of the path A0 0A – successful command/directory removed

This command is used to remove a directory from the file system.

3.3.1.11. Command: File Status

Operation Code	05 23
Data	None
Response	<Signed Byte> the number of decompression threads running or (if a negative value) the error code while decompressing

This command is used immediately after transferring a compressed file to the system, polling until the camera returns a 0 or a negative number. If the camera returns a negative value, then the file was unable to be uncompressed and should be manually removed from the camera head with the delete command. If the camera returns a 0 value, then the file is decompressed and ready for use on the camera system. If the camera returns a 1, the target file should NOT be used until decompression has completed.

3.3.1.12. Command: List NUC Attributes

Operation Code	05 33
Data	None or <String> filename
Response	<String> Attributes of NUC filename received or all NUC files if no data is sent. E0 01 – error code indicating filename must have ".nuc" extension E0 02 – error code indicating the file does not exist E0 03 - error code indicating no null character

This command returns the attributes for a specific NUC file or all NUC files in the /flash/nuc directory. If a 0 byte payload is sent, this command will return the attribute string for all NUC files found in the nuc directory. The format of the string for each file is "[filename],[radiometric flag (0/1)],[min calibrated temperature (float)],[max calibrated temperature (float)],[integration time (float)]\n". If the NUC file does not support these attributes it will return "[filename],0,0,0,0\n".

3.3.1.13. Command: Disk Free Space

Operation Code	05 25
Data	None
Response	<String> – returns the name, total size, used space, usable size, percentage complete, and mount point of the camera hard drive or an error value if an error occurs.

This command allows the user to see how much space is available for NUC and State Files on the camera. For large format focalplanes, the NUC files can be on the order of 8-10MB apiece. This indicates that the camera can comfortably store 40-50 such files before space becomes critical. If more files are needed, it is recommended to copy the NUC files onto an external storage and put the NUC files back on the camera as needed. An example response string looks like "ubi0_0 450.7M 29.7M 416.3M 7% /mnt/ubi_mnt"

Basic Camera Operation Commands

3.3.1.14. Command: Write/Read #of Data Channels Output from FPA

Operation Code	10 26
Data	<Byte> indicating how many data channels to enable: 00 – single channel 01 – dual channel 02 – four channel
Response	<Byte> returns byte received if successful E0 FF – error in attempt to change channels. Check camera log E0 00 – No value sent to change

This command modifies the number of serial data channels streaming from the focal plane. In general, increasing the number of channels increases the data rate, but there are some limitations. For base Camera Link output (single connector), the maximum number of channels is two. In full Camera Link output (two connectors) all four channels can be used.

Note that increasing the number of data channels does not automatically increase the data rate – the frame time of the image should also be modified to achieve the desired frame rate. In addition, changing the number of channels modifies the DC bias of the focalplane in the majority of digital sensors. This results in invalidating the non-uniformity correction requiring either a 1-pt update or another NUC to be loaded.

For additional information, see the sensor ICD.

Operation Code	10 27
Data	None
Response	<Byte> indicating how many data channels to enable: 00 – single channel 01 – dual channel 02 – four channel

3.3.1.15. Command: Pixel Clock Enable/Read

Operation Code	10 28
Data	<Byte> 01 – to enable pixel clock to the FPA 00 – disables pixel clock to the FPA
Response	E0 01 – data payload wrong size (single byte only) E0 03 – No data payload sent A0 00 – Pixel clock successfully modified

The pixel clock is required for clocking the FPA and allowing the FPA to operate. However, the pixel clock should only be running if and only if the FPA is powered on with all it's biases in steady state. Turning off the pixel clock will stop the data flow from the sensor, and the SERDES in the FPGA will need to be reset when the pixel clock is enabled again. The SERDES needs to be reset because it is synchronized to a phase of the clock, which will have changed my turning the pixel clock off/on.

This command is usually issued as part of the startup sequence.

Operation Code	10 29
Data	None.
Response	<Byte> 01 – if pixel clock is enabled 00 – if pixel clock is disabled

3.3.1.16. Command: Pixel Clock Select/Read

Operation Code	10 2A
Data	<Byte> 00 – 20 MHz pixel clock <default> 01 – 16 MHz pixel clock

	02 – 13.33 MHz pixel clock 03 – Programmable oscillator pixel clock
Response	E0 01 – No payload data sent E0 03 – Incorrect payload pointer E0 FF – Invalid clock select code A0 <data> - successful operation

This command modifies the frequency of the pixel clock, which in turn has a direct effect on data rate and maximum frame rate. The default operation is 20 MHz, and it is recommended that it not be changed. Use of the programmable oscillator is strongly discouraged due to excessive jitter in the clock circuit. **NOTE:** Be sure to turn off the pixel clock when changing the frequency so that the receive SERDES on the FPGA will be reset.

Operation Code	10 2B
Data	None
Response	<Byte> 00 – 20 MHz operation 01 – 16 MHz operation 02 – 13.33 MHz operation 03 – programmable oscillator operation

3.3.1.17. Command: Set/Read Programmable Oscillator Frequency

Operation Code	10 2C
Data	<Float> representing the frequency requested
Response	E0 01 – data payload is not a float E0 02 – Frequency request rejected – too low E0 03 – Frequency request rejected – too high A0 00 – Frequency set

This command enables and sets the programmable oscillator frequency. It is recommended to NOT use the programmable oscillator, as it introduces too much jitter for stable, high-quality imagery.

Operation Code	10 2D
Data	None
Response	<Float> representing the frequency of the programmable oscillator. Note that this is in little-endian order (LSB is

	first).

3.3.1.18. Command: Select/Read JAMSYNC state

Operation Code	10 30
Data	<Byte> JAMSYNC state • 00 – JAMSYNC off • 01 – JAMSYNC always on • 02 – 30 HZ JAMSYNC • 03 – Counts to sync counter then SYNC • 04 – External SYNC in • 05 – SYNC OUT at integration start • 06 – SYNC OUT at frame start • 07 – Reserved • 08 – CC1 SYNC (selective integration time) • 09 – External SYNC in (selective integration time) • 10 – CC1 SYNC (selective integration time, CC1 input is output on SYNC OUT)
Response	E0 01 – payload data invalid E0 02 – payload data code not recognized A0 00 – successfully modified JAMSYNC state

This command effectively controls the rate at which the focalplane outputs data. The user can set the mode to 4, and input their own sync pulses to start integration. Alternatively, the user can attempt to control the sync pulses with the sync counter.

The preferred method for controlling the frame rate is to set the JAMSYNC to a SYNC OUT state, and control the frame time through the frame time command.

In the selective integration time modes (8,9,10), commands that change the FPA settings are forbidden (integration time, frame time, column/row size/offset, etc).

Operation Code	10 31
Data	None
Response	<Byte> corresponding to a JAMSYNC state (see above)

3.3.1.19. Command: Set/Read Programmable Sync Register

Operation Code	10 32
Data	<Integer> representing ticks to wait prior to internally generated sync pulse
Response	E0 01 – data payload is not a 4-byte integer A0 <byte> – Register set. The byte responded is the LSB of the integer received.

This command sets the programmable sync register. It is recommended to NOT use the programmable sync register, as it introduces too much variability for stable, high-quality imagery.

Operation Code	10 33
Data	None
Response	<Integer> current Programmable Sync Register

3.3.1.20. Command: Set/Read Pixel Data Inversion

Operation Code	10 50
Data	<Byte> 01 – invert the data 00 – no data inversion
Response	<Byte> - returns set state E0 01 – correct data payload (single byte) was not sent

This command is used to set the state of data inversion. If the data inversion is turned on, then pixel data from the focalplane is inverted (cold appears hot, etc.)

Operation Code	10 51
Data	None
Response	<Byte> data inversion state

3.3.1.21. Command: Set/Read Pixel Data Depth

Operation Code	10 52
Data	<Byte> 00 – 13 bit pixel data

	01 – 14 bit pixel data (default)
Response	E0 01 – correct data payload (single byte) was not sent <Byte> - returns set state

This command is used to set the state of pixel data depth. This is necessary to change to the highest frame rates on some digital focalplanes.

Operation Code	10 53
Data	None
Response	<Byte> pixel data depth state

3.3.1.22. Command: Write/Read Window Column Size

Operation Code	10 64
Data	<Integer> Column Size
Response	E0 01 – Data payload was not correct size (4 bytes) <Integer> - column size set

This command is used to set the camera window column size (both in the FPA and FPGA). Note that this command ONLY changes window column size and does not update frame time, etc. This command uses the immediate bit state to determine when the window size is changed. Some restrictions are put on the size for each FPA in terms of column size, dependent upon the FPA. Also, the column size MUST be divisible by 8 for NUC to work properly.

Operation Code	10 65
Data	None.
Response	<Integer> Column Size

3.3.1.23. Command: Write/Read Window Column Offset

Operation Code	10 66
Data	<Integer> Column Offset
Response	E0 01 – Data payload was not correct size (4 bytes) <Integer> - column offset set

This command is used to set the camera column offset. This is typically used to offset a subwindow to either a certain portion of the scene, or alternatively, a better section of the focalplane (one with fewer bad pixels, for instance). This command has limits set by the FPA and must be divisible by 4.

Operation Code	10 67
Data	None.
Response	<Integer> Column Offset

3.3.1.24. Command: Write/Read Row Size

Operation Code	10 68
Data	<Integer> Row Size
Response	E0 01 – Data payload was not correct size (4 bytes) <Integer> - row size set

This command is used to set the camera row size. It can be set to any size from 1 to max rows on the focalplane. It can be set in increments of one.

Operation Code	10 69
Data	None
Response	<Integer> Row Size

3.3.1.25. Command: Write/Read Row Offset

Operation Code	10 6A
Data	<Integer> Row Offset
Response	E0 01 – Data payload was not correct size (4 bytes) <Payload> - column row offset set

This command is used to set the camera row offset. It can be incremented by 1, and can be used to set a sub-window anywhere in the focalplane window.

Operation Code	10 6B
Data	None.
Response	<Integer> Row Offset

3.3.1.26. Command: Write/Read Integration Time

Operation Code	10 6C

Data	<Integer> Integration Time
Response	E0 01 – Data payload was not correct size (4 bytes) <Payload> - integration time set

This command is used to set the camera integration time. For valid integration times, see the focalplane ICD.

Operation Code	10 6D
Data	None
Response	<Integer> Integration Time

3.3.1.27. Command: Write/Read Frame Time

Operation Code	10 6E
Data	<Integer> Frame Time
Response	E0 01 – Data payload was not correct size (4 bytes) <Payload> - frame time set

This command is used to set the camera frame time. For valid frame times, see the focalplane ICD.

Operation Code	10 6F
Data	None.
Response	<Integer> Frame Time

3.3.1.28. Command: Set/Read Test Pattern Enable

Operation Code	10 38
Data	<Byte> 01 – turn the test pattern ON 00 – turn the test pattern OFF
Response	E0 01 – data payload was not sent A0 <Byte> - test pattern modified

This command sets the enable for the test pattern. It should be noted that turning off the test pattern, the camera attempts to sync to the video from the focalplane. If the video cannot be synced, then the camera forcefully shows a test pattern output until video can be synced.

Upon removing the test pattern the FPGA SERDES receivers and transmitters are reset, and attempt to resync to the sensor video stream.

Operation Code	10 39
Data	None
Response	<Byte> 01 if test pattern ON, 00 if OFF

3.3.1.29. Command: Set/Read Test Pattern Selection

Operation Code	10 3A
Data	<Byte> Code selects one of the following states: • 00 – full count 0 – 16367 • 01 – line count pattern 0 – number of columns • 02 – full frame of 0s • 03 – full frame of 1s
Response	E0 01 – data payload was not sent A0 <byte> - test pattern modified, returns code selected.

This command sets the actual pattern that is output when testing. If the test pattern is not on, then no visible is change is made, but will happen the next time the test pattern is enabled.

Operation Code	10 3B
Data	None
Response	<Byte> the test pattern selection

3.3.1.30. Command: Read the ADC4 Voltage

Operation Code	10 19
Data	None
Response	<Float> representing the voltage on the ADC4 input. Note that this is in little-endian order (LSB is first).

This command reads the voltage at the ADC4 point from the ADC. The return value is a float that provides the last read voltage value.

3.3.1.31. Command: Read the ADC5 Voltage

Operation Code	10 1B
Data	None
Response	<Float> representing the voltage on the ADC5 input.

	Note that this is in little-endian order (LSB is first).

This command reads the voltage at the ADC5 point from the ADC. The return value is a float that provides the last read voltage value.

3.3.1.32. Command: Read the ADC6 Voltage

Operation Code	10 1D
Data	None
Response	<Float> representing the voltage on the ADC6 input. Note that this is in little-endian order (LSB is first).

This command reads the voltage at the ADC6 point from the ADC. The return value is a float that provides the last read voltage value.

3.3.1.33. Command: Read the ADC7 Voltage

Operation Code	10 1F
Data	None
Response	<Float> representing the voltage on the ADC7 input. Note that this is in little-endian order (LSB is first).

This command reads the voltage at the ADC7 point from the ADC. The return value is a float that provides the last read voltage value.

3.3.1.34. Command: Enable\Read ADC #1

Operation Code	10 20
Data	<Byte> determining if the ADC should be enabled (01) or disabled(00)
Response	A0 00 – ADC was modified correctly E0 01 – There was no payload sent to the command.

The ADC should ALWAYS be enabled, and is set to enable during camera startup in the camera state file. It is allowed to be disabled for testing purposes, but in general this should not be changed after startup.

Operation Code	10 21
Data	None
Response	<Byte> 01 – if the ADC is enabled 00 – if the ADC is disabled

Example	10 21 returns 10 21 00 when the ADC is disabled.

To verify that the ADC is functioning correctly, read back the ADC #1 enable. If it is 1, then the voltages read back are correct. This is useful to verify that the camera state XML loaded correctly, since the ADC are set in that file.

3.3.1.35. Command: Enable\Read ADC #2

Operation Code	10 22
Data	<Byte> determining if the ADC should be enabled (01) or disabled(00)
Response	A0 00 – ADC was modified correctly E0 01 – There was no payload sent to the command.

The ADC should ALWAYS be enabled, and is set to enable during camera startup in the camera state file. It is allowed to be disabled for testing purposes, but in general this should not be changed after startup.

Operation Code	10 23
Data	None
Response	<Byte> 01 – if the ADC is enabled 00 – if the ADC is disabled
Example	10 23 returns 10 23 00 when the ADC is disabled.

To verify that the ADC is functioning correctly, read back the ADC #2 enable. If it is 1, then the voltages read back are correct. This is useful to verify that the camera state XML loaded correctly, since the ADC are set in that file.

3.3.1.36. Command: Set/Read FPA Communications Output Lines

Operation Code	10 58
Data	<Byte> 00 – SCLK and SDIN output lines swapped 01 – no swap
Response	E0 01 – correct data payload (single byte) was not sent A0 01 – state set correctly

This command is used to set the output FPA communication lines. On some CPE/motherboard combinations these are swapped from other combinations. This command allows the output to be software selectable.

Operation Code	10 59
Data	None
Response	Returns the FPA comms output line state
Example	10 59 returns 10 53 00 when the FPA output lines are swapped.

3.3.1.37. Command: Sync FPA FPGA Registers

Operation Code	10 5A
Data	None
Response	None

This command is used to synchronize the FPGA registers to what the FPA is currently set at. This command should only be used during internal setup. Once setup is complete the firmware keeps the registers synchronized manually.

3.3.1.38. Command: Reset FPA

Operation Code	10 5C
Data	None
Response	None

This command is used to reset the FPA. This should only be done at startup by the firmware. Doing this after startup will cause the FPA/FPGA to lose sync.

3.3.1.39. Command: Reset FPA Communications

Operation Code	10 5E
Data	None
Response	None

This command is used to reset the FPA communications. This is generally controlled by the firmware; however, if the FPA is unresponsive while the imaging pipeline is changing, then issuing this command should solve the problems. (This is only an issue during testing with long/noisy lengths between the FPA and electronics.

3.3.1.40. Command: Write/Read FPA Registers

Operation Code	10 60
Data	<Byte FPA Address> <Byte FPA Data>

Response	E0 01 – Data payload was not correct size (2 bytes) <Payload> - address and data sent to the FPA

This command is used to set FPA registers. For more information on the FPA registers, refer to the specific FPA ICD. NOTE: This is a dangerous command, use only if confident that this is what you want. Changes are NOT REFLECTED in FPGA behavior!!! Use specific change commands to effect both FPA and FPGA behavior.

Operation Code	10 61
Data	<Byte> FPA Address
Response	<Byte> FPA Data at FPA Address requested

3.3.1.41. Command: Write/Read FPA Communications Immediate Bit

Operation Code	10 62
Data	<Byte> 00 – immediate bit is off (default), FPA changes set at NEXT frame 01 – immediate bit is on, FPA immediately changes
Response	E0 01 – Data payload was not at least 1 byte <Payload> - Immediate bit set to either 0 or 1

This command is used to set FPA registers either on the start of the next integration time or as soon as the command is fully latched. For more information on the FPA registers, refer to the specific FPA ICD. NOTE: This is a dangerous command, use only if confident that this is what you want.

Operation Code	10 63
Data	None
Response	<Byte> Immediate Bit state

SERDES Commands

3.3.1.42. Command: Reset/Read SERDES FIFO

Operation Code	10 40
Data	<Byte> 00 – Stop SERDES FIFO reset on all channels 01 – Reset SERDES FIFO channel 1

	02 – Reset SERDES FIFO channel 2
	03 – Reset SERDES FIFO on channels 1 & 2
	04 – Reset SERDES FIFO on channel 3
	05 – Reset SERDES FIFO on channels 1 & 3
	…..
	0F – Reset SERDES FIFO on all channels
Response	E0 01 – correct data payload was not sent
	A0 <byte> - Reset SERDES FIFO mask written

This command is used to reset the SERDES serial bit FIFO. It is recommended that the user not modify these directly. It is instead recommended to reset the total SERDES path by turning the test pattern on/off.

Operation Code	10 41
Data	None
Response	Returns the SERDES FIFO reset mask

3.3.1.43. Command: Read DPA Locked

Operation Code	10 43
Data	None
Response	<Byte> indicating if dynamic phase alignment is linked on all 4 channels (or channels running)

This command is used to verify that SERDES channels are dynamically phased aligned. The least significant bit represents channel 1, the next least significant bit represents channel 2, etc.

3.3.1.44. Command: Reset/Read SERDES Channels

Operation Code	10 44
Data	<Byte>
	00 – Stop SERDES reset on all channels
	01 – Reset SERDES channel 1
	02 – Reset SERDES channel 2
	03 – Reset SERDES on channels 1 & 2
	04 – Reset SERDES on channel 3
	05 – Reset SERDES on channels 1 & 3
	…..

	0F – Reset SERDES on all channels
Response	E0 01 – correct data payload was not sent A0 <byte> - Reset SERDES FIFO mask written

This command is used to reset the SERDES channels. It is recommended that the user not modify these directly. It is instead recommended to reset the total SERDES path by turning the test pattern on/off.

Operation Code	10 45
Data	None
Response	<Byte> SERDES Channel reset mask

3.3.1.45. Command: Read FPA Clock Locked

Operation Code	10 47
Data	None
Response	<Byte> indicating if FPA clock is locked (01)

This command returns whether the clock from the FPA to the FPGA's SERDES has been locked by the SERDES. Until this happens, the FPGA can't receive sensor data.

3.3.1.46. Command: Set/Read Data Alignment on SERDES Channels

Operation Code	10 48
Data	<Byte> 00 – Stop SERDES data alignment on all channels 01 – Reset SERDES channel 1 02 – Reset SERDES channel 2 03 – Reset SERDES on channels 1 & 2 04 – Reset SERDES on channel 3 05 – Reset SERDES on channels 1 & 3 ….. 0F – Reset SERDES on all channels
Response	E0 01 – correct data payload was not sent A0 <byte> - Reset SERDES FIFO mask written

This command is used to reset the SERDES channels. It is recommended that the user not modify these directly. It is instead recommended to reset the total SERDES path by turning the test pattern on/off.

Operation Code	10 49
Data	None
Response	<Byte> SERDES Channel reset mask

3.3.1.47. Command: Read Data Alignment Locked

Operation Code	10 4B
Data	None
Response	<Byte> indicating if SERDES data alignment is locked

This command returns whether the FPGA's SERDES has been locked by the pixel data alignment module for each channel. Until this happens, the FPGA can't receive correct sensor data. The least significant bit represents channel 1, the next least significant bit represents channel 2, etc.

3.3.1.48. Command: Reset/Read Data SERDES Channels

Operation Code	10 4C
Data	<Byte> 00 – Stop SERDES data alignment on all channels 01 – Reset SERDES channel 1 02 – Reset SERDES channel 2 03 – Reset SERDES on channels 1 & 2 04 – Reset SERDES on channel 3 05 – Reset SERDES on channels 1 & 3 ….. 0F – Reset SERDES on all channels
Response	E0 01 – correct data payload was not sent <Payload> - Data channels held in reset

This command is used to reset the SERDES channels. It is recommended that the user not modify these directly. It is instead recommended to reset the total SERDES path by turning the test pattern on/off.

Operation Code	10 4D
Data	None
Response	<Byte> the SERDES Channel Data Alignment reset mask

Example	10 4D returns 10 4D 00 when the SERDES data channel reset mask is off

3.3.1.49. Command: Set/Read FPGA Channel Control

Operation Code	10 56
Data	<Byte> E4 – FPA channel 3 to FPGA channel 3, FPA channel 2 to FPGA channel 2, etc. 00 – FPA channel 0 to FPGA channel 3, FPA channel 0 to FPGA channel 2, etc.
Response	E0 01 – correct data payload (single byte) was not sent <Byte> - returns set state

This command is used to set the incoming FPGA channels to the corresponding FPA channel. For the vast majority of FPAs/cameras, there should be no reason to change this. However, in some case bond wires are burnt out, buffers are overloaded and burnt out, etc. requiring channels to be changed This is done by setting 2 bits for each channel, as shown:

Bit	Signal Name	Description	Default Value
0	IN0	FPGA Input Number Bit 0 for input 0	0
1	IN0	FPGA Input Number Bit 1 for input 0	0
2	IN1	FPGA Input Number Bit 0 for input 1	1
3	IN1	FPGA Input Number Bit 1 for input 1	0
4	IN2	FPGA Input Number Bit 0 for input 2	0
5	IN2	FPGA Input Number Bit 1 for input 2	1
6	IN3	FPGA Input Number Bit 0 for input 3	1
7	IN3	FPGA Input Number Bit 1 for input 3	1

For example, if FPA channel 0 should be output on FPGA channel 1, then bits 2 and 3 should read “00”. When doing this, it is important to modify the FPA's output control register as well. See the respective FPA ICD.

Operation Code	10 57
Data	None

Response	<Byte> FPGA channel control state

FPA Bias Commands

3.3.1.50. Command: Turn off ALL DACs

Operation Code	10 24
Data	None
Response	Echoes all bytes received

This command turns off all DAC voltages, effectively turning off the FPA sensor. This command is used internally when monitored voltages/currents exceed their effective limits. In that case, the biases are shut off to avoid damaging sensitive bond wires or the sensor itself.

3.3.1.51. Command: Set VPOS/VPH Bias Voltage

Operation Code	10 00
Data	<Float> representing voltage to set the VPOS/VPH bias. Note that this is in little-endian order (LSB is first).
Response	A0 0A – Successful modification of VPOS/VPH bias E0 01 – Command does not have enough bytes to be a float E0 02 – Voltage requested greater than max allowed E0 03 – Voltage requested less than minimum and not 0.0V

This command sets the digital high positive rail voltage bias to the FPA and the analog bias supply. Note that different FPAs may have different turn on bias sequences, and thus the individual ICDs of the focalplane should be consulted prior to using this command. This command is protected by internal configuration files to prevent the user from applying too high a voltage to the FPA. If these configuration files are not loaded then the FPA cannot be turned on.

3.3.1.52. Command: Read the VPOS Bias

Operation Code	10 01
Data	None
Response	<Float> representing the voltage on the VPOS bias. Note that this is in little-endian order (LSB is first).

This command reads the VPOS bias from the ADC. Note that this is the bias that directly feeds the focalplane sensor. The return value is a float that provides the last read bias voltage value.

3.3.1.53. Command: Read the VPOS1 Bias

Operation Code	10 03
Data	None
Response	<Float> representing the voltage on the VPOS1 bias. Note that this is in little-endian order (LSB is first).

This command reads the VPOS1 bias from the ADC. Note that this is the bias that is on the high side of the current sense resistor. This reading, coupled with the voltage given by VPOS, can provide current information through the formula (VPOS1 – VPOS) / 5. (Current is in amps.)

The return value is a float that provides the last read bias voltage value.

3.3.1.54. Command: Read the VPH Bias

Operation Code	10 05
Data	None
Response	<Float> representing the voltage on the VPH bias. Note that this is in little-endian order (LSB is first).

This command reads the VPH bias from the ADC. Note that this is the bias that directly feeds the focalplane sensor. The return value is a float that provides the last read bias voltage value.

3.3.1.55. Command: Read the VPH1 Bias

Operation Code	10 07
Data	None
Response	<Float> representing the voltage on the VPH1 bias. Note that this is in little-endian order (LSB is first).

This command reads the VPH1 bias from the ADC. Note that this is the bias that is on the high side of the current sense resistor. This reading, coupled with the voltage given by VPH, can provide current information through the formula (VPH1 – VPH) / 5. (Current is in amps.)

The return value is a float that provides the last read bias voltage value.

3.3.1.56. Command: Set the VPOUT Bias

Operation Code	10 08
Data	<Float> representing voltage to set the VPOUT bias. Note that this is in little-endian order (LSB is first).
Response	A0 00 – Successful modification of VPOUT bias E0 01 – Command does not have enough bytes to be a

	float E0 02 – Voltage requested greater than max allowed E0 03 – Voltage requested less than minimum and not 0.0V

This command sets the LVDS supply bias to the FPA. Note that different FPAs may have different turn on bias sequences, and thus the individual ICDs of the focalplane should be consulted prior to using this command. This command is protected by internal configuration files to prevent the user from applying too high a voltage to the FPA. If these configuration files are not loaded then the FPA cannot be turned on.

3.3.1.57. Command: Read the VPOUT Bias

Operation Code	10 09
Data	None
Response	<Float> representing the voltage on the VPOUT bias. Note that this is in little-endian order (LSB is first).

This command reads the VPOUT bias from the ADC. Note that this is the bias that directly feeds the focalplane sensor. The return value is a float that provides the last read bias voltage value.

3.3.1.58. Command: Read the VPOUT1 Bias

Operation Code	10 0B
Data	None
Response	<Float> representing the voltage on the VPOUT1 bias. Note that this is in little-endian order (LSB is first).

This command reads the VPOUT1 bias from the ADC. Note that this is the bias that is on the high side of the current sense resistor. This reading, coupled with the voltage given by VPOUT, can provide current information through the formula (VPOUT1 – VPOUT) / 5. (Current is in amps.)

The return value is a float that provides the last read bias voltage value.

3.3.1.59. Command: Set the VPL Bias

Operation Code	10 0C
Data	<Float> representing voltage to set the VPL bias. Note that this is in little-endian order (LSB is first).
Response	A0 00 – Successful modification of VPL bias E0 01 – Command does not have enough bytes to be a float

	E0 02 – Voltage requested greater than max allowed E0 03 – Voltage requested less than minimum and not 0.0V

This command sets the digital low positive rail bias to the FPA. Note that different FPAs may have different turn on bias sequences, and thus the individual ICDs of the focalplane should be consulted prior to using this command. This command is protected by internal configuration files to prevent the user from applying too high a voltage to the FPA. If these configuration files are not loaded then the FPA cannot be turned on.

3.3.1.60. Command: Read the VPL Bias

Operation Code	10 0D
Data	None
Response	<Float> representing the voltage on the VPL bias. Note that this is in little-endian order (LSB is first).

This command reads the VPL bias from the ADC. Note that this is the bias that directly feeds the focalplane sensor. The return value is a float that provides the last read bias voltage value. Note that with no load on these pins, this voltage is being pulled low.

3.3.1.61. Command: Read the VPL1 Bias

Operation Code	10 0F
Data	None
Response	<Float> representing the voltage on the VPL1 bias. Note that this is in little-endian order (LSB is first).

This command reads the VPL1 bias from the ADC. Note that this is the bias that is on the high side of the current sense resistor. This reading, coupled with the voltage given by VPL, can provide current information for the bias through the formula (VPL1 – VPL) / 5. (Current is in amps.)

The return value is a float that provides the last read bias voltage value.

3.3.1.62. Command: Set the VCommon Bias

Operation Code	10 10
Data	<Float> representing voltage to set the VCommon bias. Note that this is in little-endian order (LSB is first).
Response	A0 00 – Successful modification of VPL bias E0 01 – Command does not have enough bytes to be a

	float

This command sets the analog ground rail bias to the FPA. Note that different FPAs may have different turn on bias sequences, and thus the individual ICDs of the focalplane should be consulted prior to using this command. This command is protected by internal configuration files to prevent the user from applying too high a voltage to the FPA. If these configuration files are not loaded then the FPA cannot be turned on.

IT IS HIGHLY RECOMMENDED TO LEAVE THIS RAIL AT THE VOLTAGE SET BY THE FACTORY. THIS COMMAND IS MEANT FOR CALIBRATION USAGE AT THE FACTORY ONLY.

3.3.1.63. Command: Read the VCommon Bias

Operation Code	10 11
Data	None
Response	<Float> representing the voltage on the VCommon bias. Note that this is in little-endian order (LSB is first).

This command reads the VCommon bias from the ADC. Note that this is the bias that directly feeds the focalplane sensor (via VDETCOM if connected). The return value is a float that provides the last read bias voltage value. Note that with no load on these pins, this voltage is being pulled low.

3.3.1.64. Command: Read the VCommon1 Bias

Operation Code	10 13
Data	None
Response	<Float> representing the voltage on the VCommon1 bias. Note that this is in little-endian order (LSB is first).

This command reads the VCommon1 bias from the ADC. Note that this is the bias that is on the high side of the current sense resistor. This reading, coupled with the voltage given by VPL, can provide current information for the bias through the formula (VCommon1 – VCommon) / 5. (Current is in amps.)

The return value is a float that provides the last read bias voltage value.

3.3.1.65. Command: Write/Read SBF FPA Detector Bias

Operation Code	10 90
Data	<Byte> Detector Bias
Response	E0 01 – Data payload was not correct size (1 or more bytes) <Payload> - SBF FPA detector bias set

This command is used to set the camera detector bias if it has an SBF sensor. For valid detector bias settings, see the focalplane ICD.

Operation Code	10 91
Data	None
Response	<Byte> Current Detector Bias

3.3.1.66. Command: Write/Read SBF Ramp High Bias

Operation Code	10 92
Data	<Byte> Ramp High Bias
Response	E0 01 – Data payload was not correct size (1 or more bytes) <Payload> - SBF FPA ramp high bias set

This command is used to set the camera ramp high bias if it has an SBF sensor. For valid ramp high bias settings, see the focalplane ICD.

Operation Code	10 93
Data	None
Response	<Byte> Current Ramp High Bias

3.3.1.67. Command: Write/Read SBF FPA Ramp Low Bias

Operation Code	10 94
Data	<Byte> Ramp Low Bias
Response	E0 01 – Data payload was not correct size (1 or more bytes) <Payload> - SBF FPA Ramp Low bias set

This command is used to set the camera ramp low bias. For valid ramp low bias settings, see the focalplane ICD.

Operation Code	10 95
Data	None
Response	<Byte> Current Ramp Low Bias

3.3.1.68. Command: Read Instantaneous Bias Currents

Operation Code	10 B1
Data	None.
Response	<Float> - instantaneous current (A) of the VPOS bias

Operation Code	10 B3
Data	None.
Response	<Float> - instantaneous current (A) of the VPL bias

Operation Code	10 B5
Data	None.
Response	<Float> - instantaneous current (A) of the VPH bias

Operation Code	10 B7
Data	None.
Response	<Float> - instantaneous current (A) of the VPOUT bias

Operation Code	10 B9
Data	None.
Response	<Float> - instantaneous current (A) of the VCOMMON bias

These commands provide the results of the instantaneous bias currents and are used in the watchdog to be sure that they don't exceed safe values.

Number of Channels Through Modules

This set of commands sets the number of channels through each module. In general, these should all be set to the same value. However, it is possible when incorporating the frame averaging to have one incoming number of channels and a different output number of channels, thus necessitating all of these commands.

3.3.1.69. Command: Write/Read Raw Data Number of Channels

Operation Code	10 C0
Data	<Byte> - the new number of raw channels 00 – 1 channel 01 – 2 channel 02 – 4 channel 03 – 4 channel
Response	E0 01 – Data payload was not correct size (more than 1 byte) A0 <Byte> – Number of channels set

This command is used to set the number of data channels the RAW DATA module outputs. Note that this command does NOT change the number of channels output from the FPA (use the 10 26 command for that).

Operation Code	10 C1
Data	None
Response	<Byte> - the current number of raw channels

3.3.1.70. Command: Write/Read Frame Average Number of Channels

Operation Code	10 C2
Data	<Byte> - the new number channels for the frame average module 00 – 1 channel 01 – 2 channel 02 – 4 channel 03 – 4 channel
Response	E0 01 – Data payload was not correct size (more than 1 byte) A0 <Byte> – Number of channels set

This command is used to set the number of data channels the FRAME AVERAGING module outputs.

Operation Code	10 C3
Data	None
Response	<Byte> - the current number of channels for the frame

	averaging module

3.3.1.71. Command: Write/Read NUC/BPR Number of Channels

Operation Code	10 C4
Data	<Byte> - the new number channels for the BPR module 00 – 1 channel 01 – 2 channel 02 – 4 channel 03 – 4 channel
Response	E0 01 – Data payload was not correct size (more than 1 byte) A0 <Byte> – Number of channels set

This command is used to set the number of data channels the NUC/BPR DATA module outputs.

Operation Code	10 C5
Data	None
Response	<Byte> - the current number channels for the BPR module

3.3.1.72. Command: Write/Read AGC Number of Channels

Operation Code	10 C6
Data	<Byte> - the new number channels for the AGC module 00 – 1 channel 01 – 2 channel 02 – 4 channel 03 – 4 channel
Response	E0 01 – Data payload was not correct size (more than 1 byte) A0 <Byte> – Number of channels set

This command is used to set the number of data channels the AGC module outputs.

Operation Code	10 C7
Data	None

Response	<Byte> - the current number channels for the AGC module

3.3.1.73. Command: Write/Read Convolution Number of Channels

Operation Code	10 C8
Data	<Byte> - the new number channels for the convolution module 00 – 1 channel 01 – 2 channel 02 – 4 channel 03 – 4 channel
Response	E0 01 – Data payload was not correct size (more than 1 byte) A0 <Byte> – Number of channels set

This command is used to set the number of data channels the CONVOLUTION module outputs.

Operation Code	10 C9
Data	None
Response	<Byte> - the current number channels for the convolution module

3.3.1.74. Command: Write/Read OUTPUT Number of Channels

Operation Code	10 CA
Data	<Byte> - the new number channels for the output module 00 – 1 channel 01 – 2 channel 02 – 4 channel 03 – 4 channel
Response	E0 01 – Data payload was not correct size (more than 1 byte) A0 <Byte> – Number of channels set

This command is used to set the number of data channels the OUTPUT module outputs.

Operation Code	10 CB
Data	None
Response	<Byte> - the current number channels for the output module
Example	10 C1 returns 10 C1 01 when the OUTPUT module is set to output 2 channels

3.3.1.75. Command: Write/Read CAPTURE Module Number of Channels

Operation Code	10 CC
Data	<Byte> - the new number channels for the capture module 00 – 1 channel 01 – 2 channel 02 – 4 channel 03 – 4 channel
Response	E0 01 – Data payload was not correct size (more than 1 byte) A0 <Byte> – Number of channels set

This command is used to set the number of data channels the CAPTURE module outputs.

Operation Code	10 CD
Data	None
Response	<Byte> - the current number channels for the capture module

Image Acquisition (Frame Grabber) Window Coordinates

This set of commands records the coordinates of the frame for the framegrabber/WinIRC to retrieve. That is, if the software is performing additional windowing, then it stores the coordinates in these registers between sessions. This is considered part of the camera state, and saved when the camera state is saved.

3.3.1.76. Command: Write/Read Framegrabber Column Offset

Operation Code	10 D0

Data	<Integer> - column offset for the framegrabber
Response	E0 01 – Data payload was not correct size (4 bytes) A0 01 – Columns offset value set

Operation Code	10 D1
Data	None
Response	<Integer> - 4 bytes indicating the column offset

3.3.1.77. Command: Write/Read Framegrabber Row Offset

Operation Code	10 D2
Data	<Integer> - row offset for the framegrabber
Response	E0 01 – Data payload was not correct size (4 bytes) A0 01 – Row offset value set

Operation Code	10 D3
Data	None
Response	<Integer> - 4 bytes indicating the row offset

3.3.1.78. Command: Write/Read Framegrabber Columns

Operation Code	10 D4
Data	<Integer> - number of columns for the framegrabber
Response	E0 01 – Data payload was not correct size (4 bytes) A0 01 – Columns value set

Operation Code	10 D5
Data	None
Response	<Integer> - 4 bytes indicating the number of columns

3.3.1.79. Command: Write/Read Framegrabber Rows

Operation Code	10 D6
Data	<Integer> - number of rows for the framegrabber
Response	E0 01 – Data payload was not correct size (4 bytes) A0 01 – Rows value set

Operation Code	10 D7
Data	None
Response	<Integer> - 4 bytes indicating the number of rows

3.4. Video Output Commands

3.4.1.1. Command: Reset Image Processing Pipeline

Operation Code	20 00
Data	None.
Response	<Payload> and resets the image processing pipeline.
Example	20 00 returns 20 00 and resets the full image processing pipeline (not the sensor, nor the SERDES)

Not yet fully implemented.

3.4.1.2. Command: Select/Read Camera Link Multiplexor Output

Operation Code	20 02
Data	<Integer > - new Camera Link Mux selection 00 00 00 00 – (0) – Raw data 01 00 00 00 – (1) – Frame Average data 02 00 00 00 – (2) – NUC'd/BPR data 03 00 00 00 – (3) – Bilinear interpolation data 04 00 00 00 – (4) – No video 05 00 00 00 – (5) – Reserved 06 00 00 00 – (6) – Reserved 07 00 00 00 – (7) – No video

Response	<Payload> and sets the Camera Link mux to the output requested

This command is used to set the Camera Link output mux. In the case of raw data, turning off the NUC and BPR modules will output raw data, but moving the Camera Link mux will reduce the latency of the frame from the camera to the frame grabber. Obviously, in order to get BPR data the mux must be moved back to the 02 selection.

Operation Code	20 03
Data	None
Response	<Integer> current Camera Link multiplexor selection

3.4.1.3. Command: Select/Read Ethernet Frame Capture Multiplexor Output

Operation Code	20 04
Data	<Integer > - new Frame Capture Mux selection 00 00 00 00 – (0) – Raw data 01 00 00 00 – (1) – Frame Average data 02 00 00 00 – (2) – NUC'd/BPR data 03 00 00 00 – (3) – No video 04 00 00 00 – (4) – No video 05 00 00 00 – (5) – Reserved (future) 06 00 00 00 – (6) – Reserved (future) 07 00 00 00 – (7) – No video
Response	<Payload> and sets the frame capture mux to the output requested

This command is used to set the frame capture mux. The video stream off of this mux is used to capture data to a frame buffer.

Operation Code	20 05
Data	None
Response	<Integer> current frame capture multiplexor selection

3.4.1.4. Command: Write/Read Alternate Dual Base Camera Link Mode

Operation Code	20 60

Data	<Byte> - set dual base mode is on (1) or off (0) 0 – Only output base Camera Link on connector #1 1 – output base Camera Link on connectors #1 and #2
Response	E0 01 – data is not a byte A0 00 – Alternate mode turned on/off successfully

NOTE: This command has no effect in 4 channel output mode!

This command is used to set the Alternate Dual Base Camera Link mode on or off.

Operation Code	20 61
Data	None
Response	<Byte> with current mode selection

3.4.1.5. Command: Write/Read Connector #2 Output Selection

Operation Code	20 62
Data	<Integer> - output selection for the second Base Camera Link output 0 – Raw data 1 – Frame Averaged data 2 – NUC/BPR data 3 – AGC data 4/7 – no video output 5 – post convolution filter #1 6 – post convolution filter #2
Response	E0 01 – data is not an integer <Payload> - selection successfully changed

NOTE: This command has no effect in 4 channel output mode!

This command is used to set the output of the second connector when in Dual Base Mode. This allows a different video output than what is being output on connector #1, allowing the user to view 2 different streams.

Operation Code	20 63
Data	None

Response	<Integer> with current mode selection

3.5. Metadata Commands

3.5.1.1. Command: Select/Read Metadata Data Enable

Operation Code	20 06
Data	<Byte > - new Metadata Enable selection 00 – (0) – Metadata disabled 01 – (1) – Metadata on first row only 02 – (2) – Metadata on last row only 03 – (3) – Metadata on both first and last rows
Response	E0 01 – error code indicating that no selection payload was sent <Payload> and sets the metadata enable as requested

This command is used to set the metadata on the image. Note that there are multiple commands that control the metadata.

When adding metadata to the first line, the image video is shifted down by 1 row, since the metadata is placed on the first row. Likewise, when the special Metadata Row Overwrite Flag is not set, enabling the last row of metadata waits until the frame is complete, then adds metadata at the end of the frame. Setting the Metadata Row Overwrite Flag allows the last row of metadata to be set to any row (1- end of frame + 1).

It should be noted that adding metadata can cause issues if less than 4 full lines of timing is not provided after the image by use of the frame time register.

Operation Code	20 07
Data	None
Response	<Byte> current metadata enable selection

3.5.1.2. Command: Reset Metadata Frame Counter

Operation Code	20 08
Data	None.
Response	<Payload> and resets the frame counter to 0.

Sending this command to the camera will cause the metadata frame counter to be 0 on the next frame.

3.5.1.3. Command: Write/Read Metadata Replicate

Operation Code	20 0A
Data	<Byte> - indicates whether metadata is replicated at the halfway point of the enabled lines 00 – metadata replication OFF 01 – metadata replication ON
Response	E0 01 – no data was supplied to the command <Payload> and sets the metadata replication as requested

This command is used to set the metadata replication. Metadata is 134 bytes long in Revision C. If the halfway point is less than 134, this command does not execute. Also, if the frame column size is less than 268, then the second set of data is truncated.

Operation Code	20 0B
Data	None
Response	<Byte> current metadata replication

3.5.1.4. Command: Write/Read Metadata User Defined Data

Operation Code	20 0C
Data	<24 Bytes> - user defined metadata to be striped
Response	E0 01 – no data was supplied to the command A0 00 – user defined metadata set

This command is used to set the user defined metadata, and up to 24 bytes are available to be set. This is not saved by the camera!

Operation Code	20 0D
Data	None
Response	<24 Bytes> current user defined metadata

3.5.1.5. Command: Write/Read Metadata Overwrite Row

Operation Code	20 0E
Data	<Integer> - the row of data to overwrite (range is row 1 to max frame row + 1)

Response	E0 01 – data is not an integer E0 02 – row requested is out of bounds <Payload> - metadata last row set to row requested

This command is used to set the row number to overwrite when metadata last row is enabled AND the Metadata Overwrite Row Flag is set. Some error checking is handled in this function, but it is still possible to change the window size and create an invalid metadata row situation. In that case, just use this command to set the row to a valid selection, and the metadata striping will continue as normal.

Operation Code	20 0F
Data	None
Response	<Integer> current row to overwrite

3.6. NUC/BPR Commands

3.6.1.1. Command: Write/Read NUC Input Mux Selection

Operation Code	20 1C
Data	<Byte> - input that NUC should use 0 – Raw data 1 – Averaged frame data
Response	E0 01 – data is not a byte A0 <Byte> – Successfully set NUC input mux selection

This command is used to set the data input to the NUC/BPR modules. If 0 (default), then the input is raw data from the sensor. If 1, then the raw data from the sensor is averaged for X number of frames, then sent to the NUC/BPR modules.

Operation Code	20 1D
Data	None
Response	<Byte> - current input that NUC is using
Example	20 1D returns 20 1D 00 when the NUC input mux is set to raw data (default)

3.6.1.2. Command: Write/Read Metadata Overwrite Row Flag

Operation Code	20 2A
Data	<Byte> - overwrite flag

	00 – Metadata last row appended (default) 01 – Metadata last row striped
Response	E0 01 – data payload was not 1 byte A0 0A – Metadata Overwrite Row Flag set as requested
Example	20 2A 01 returns 20 2A A0 0A and sets the last row metadata to be striped on the requested row

This command is used to set the Metadata Overwrite Row Flag, which determines whether the last row is appended to the end of the frame, or overwrites a row on the image. It should be noted that the data on the overwritten row is lost.

Operation Code	20 2B
Data	None
Response	<Byte> overwrite flag status

3.6.1.3. Command: Enable/Read Non-Uniformity Correction

Operation Code	20 10
Data	<Byte> - NUC module enable 00 – Disable NUC module (default) 01 – Enable NUC module
Response	<Payload> and NUC is enabled/disabled as requested

This command is used to enable the Non-Uniformity Correction Module. The NUC module must be enabled in order for Bad Pixel Replacement to work.

Operation Code	20 11
Data	None
Response	<Integer> indicates whether NUC is enabled

3.6.1.4. Command: Enable/Read Bad Pixel Replacement

Operation Code	20 12
Data	<Byte> - BPR module enable 00 – Disable BPR module(default) 01 – Enable BPR module

Response	<Payload> and BPR is enabled/disabled as requested

This command is used to enable the Bad Pixel Replacement Module. The BPR module must be enabled in order for Bad Pixel Replacement to work. **NOTE** that BPR will only correctly function if there are more than 3 rows and more than 48 pixels per row.

Operation Code	20 13
Data	None
Response	<Integer> indicates whether BPR is enabled

3.6.1.5. Command: Write/Read Active NUC Slot

Operation Code	20 30
Data	<Integer> - NUC slot to make active (range:0-11)
Response	<Payload> and sets the active NUC slot to the requested value. Anything over 11 will be set to 11.

This command is used to set the active NUC slot. This allows the user to rapidly switch between NUC tables (on end of frame).

Operation Code	20 31
Data	None
Response	<Integer> current active NUC slot

3.6.1.6. Command: Write/Read NUC File to Slot

Operation Code	20 32
Data	<Integer> - NUC slot (range:0-11) <String> - the filename and path of the NUC file to write to chosen NUC slot
Response	E0 00 00 00 – data too short E0 01 00 00 – no null terminating byte in string E0 03 00 00 – error loading NUC, see log E0 04 00 00 – path does not begin correctly, see log A0 0A 00 00 – loaded requested NUC file into requested slot

This command is used to load a NUC file from the camera's Flash drive into a NUC memory slot for active correction. Be sure to turn active NUC correction off prior to using this command!

Operation Code	20 33
Data	<Integer> - the desired NUC slot (range: 0-11)
Response	<String> “BAD SLOT” if integer slot request is outside of range, “NONE” if no file in current slot else filename of current file in requested slot

3.6.1.7. Command: Remove File from Slot

Operation Code	20 34
Data	<Integer> - NUC slot to depopulate(range:0-11)
Response	E0 01 – integer not provided E0 02 – slot requested out of range A0 00 – NUC slot successfully removed

This command is used to remove a NUC file from being associated with a NUC slot. IT DOES NOT REMOVE THE NUC DATA FROM MEMORY. This command is mainly used to remove a NUC file from the slot so that it is not saved to the slot when saving a camera state. When the camera state is saved, all NUC files are saved in their associated slots, which adds overhead for file copies during camera boot or whenever the state is loaded. For large arrays, these delays can be significant.

3.6.1.8. Command: Perform 1 point update

Operation Code	20 36
Data	None
Response	A0 0A – NUC slot successfully removed

A 1 point update updates the current NUC coefficients (offset only) to correct for drift. This does NOT perform a BPR calibration. If no NUC file is available in the current slot, then no bad pixels are discovered, and the bad pixels remain with no correction. This is mainly to be used as a short cut to the full 1pt calibration which discovers the complete bad pixel map.

3.6.1.9. Command: Perform 1 point calibration

Operation Code	20 38
Data	None
Response	A0 0A – NUC successfully calibrated

A 1 point calibration performs an offset only NUC correction and discovers the bad pixel map. For bad pixel discovery it uses a cosine^4 map from the default file, as well as floor average and threshold values supplied by the user. If a NUC file is in the current slot, then the gain values from the NUC file are used, but the offset and bad pixel map are overwritten. Otherwise, the gains are set to 1.

3.6.1.10. Command: 2-point Calibration – Collect Set 1

Operation Code	20 3A
Data	None
Response	A0 01 – Set 1 frames successfully averaged

This command grabs a number of frames and averages them for a 2 point Non-Uniformity Calibration. The number of frames is set buy the 20 3E command. The number of frames to grab can be anywhere from 2 to 512. This set of frames total average needs to be lower than the total average for set 2, or the NUC calculation will fail.

3.6.1.11. Command: 2-point Calibration – Collect Set 2

Operation Code	20 3C
Data	None
Response	A0 01 – Set 2 frames successfully averaged

This command grabs a number of frames and averages them for a 2 point Non-Uniformity Calibration. The number of frames is set buy the 20 3E command. The number of frames to grab can be anywhere from 2 to 512. This set of frames total average needs to be greater than the total average for set 1, or the NUC calculation will fail.

3.6.1.12. Command: Perform 2 point calibration

Operation Code	20 82
Data	None
Response	A0 01– 2pt NUC successfully calibrated and set to active slot

A 2 point calibration calculates the gain and offset coefficients for NUC correction and discovers the bad pixel map. This command relies on the 2 point calibration sets having already been captured before it is run. For bad pixel discovery it uses a cosine^4 map from the default file, as well as floor average and threshold values supplied by the user. If the current slot is occupied, this command overwrites those values. The most common cause for this command to fail is too tight a threshold on the bad pixel discovery, increase the threshold value.

3.6.1.13. Command: Write/Read NUC frames to average

Operation Code	20 3E
Data	<Integer> - number of frames to average during NUC calculations (2 - 512)

Response	E0 01 – data is not an integer A0 0A – Frames number set correctly to some value

This command is used to set the number of frames to average when calculating the non-uniformity coefficients. The number can range between 2 and 512, and is capped to those numbers if a value outside the range is requested.

Operation Code	20 3F
Data	None
Response	<Integer> current frames value

3.7. Frame Averaging Commands

3.7.1.1. Command: Write/Read Frame Averaging Mode

Operation Code	20 70
Data	<Byte> - frame averaging mode 00 – no frame averaging (no frames output!) 01 – single shot frame averaging meant for calibration 02 – continuous block frame averaging 03 – continuous streaming frame averaging (future – not yet implemented)
Response	E0 01 – data is not a byte A0 01 – Frame averaging mode successfully modified

This command is used to set the frame averaging mode. The single shot selection is meant to be used for calibrations (1pt and 2pt), while the block averaging mode allows faster data rates to be converged for base Camera Link output.

Operation Code	20 71
Data	None
Response	<Byte> with current frame averaging mode selection
Example	20 71 returns 20 71 00 when frame averaging is off

3.7.1.2. Command: Write/Read Frame Averaging Single Shot Complete

Operation Code	20 73
Data	None

Response	<Byte> - frame averaging single shot state, complete(1) or not (0)

This function is mainly used internally during calibration.

3.7.1.3. Command: Write/Read Number of Frames to Average

Operation Code	20 74
Data	<Integer> - number of frames to average. Range is 2-512 for block frame averaging.
Response	E0 01 – data is not an integer A0 01 – Frame averaging mode successfully modified

This command is used to set the frame averaging mode's number of frames to average. Note that if a value is given outside the range, the value is capped to be the largest (or smallest) value in the range.

Operation Code	20 75
Data	None
Response	<Integer> with current number of frames to average

3.7.1.4. Command: Write/Read Frame Averaging Sum Buffer Offset

Operation Code	20 76
Data	<Integer> - the offset within the 256MB Frame Capture Buffer to store the intermediate sums
Response	E0 01 – data is not an integer A0 01 – Sum Buffer Offset set

This command is used to set the frame averaging mode sum buffer offset within the frame capture memory. This is set automatically by the embedded software and does not usually have to be modified.

Operation Code	20 77
Data	None
Response	<Integer> with current sum buffer offset
Example	20 77 returns 20 77 C0 DB FA 01 (default value)

3.7.1.5. Command: Write/Read Frame Averaging Output Buffer Offset

Operation Code	20 78

Data	<Integer> - the offset within the 256MB Frame Capture Buffer to store the averaged output
Response	E0 01 – data is not an integer A0 01 – Sum Buffer Offset set

This command is used to set the frame averaging mode averaged output buffer offset within the frame capture memory. This is set automatically by the embedded software and does not usually have to be modified.

Operation Code	20 79
Data	None
Response	<Integer> with current sum buffer offset

3.7.1.6. Command: Read Frame Averaging Total Average

Operation Code	20 7B
Data	None
Response	<Integer> - the total average of the averaged frame.

This command is used to return the frame averaging total average calculated on the last block frame average.

3.7.1.7. Command: Read Frame Averaging Total Sum

Operation Code	20 7D
Data	None
Response	<Long Long> - 8 bytes providing the total sum of all pixels of the averaged frame.

This command is used to return the frame averaging total sum calculated on the last block frame average.

3.7.1.8. Command: Write Calculate the Default Output Buffer Address

Operation Code	20 7E
Data	None
Response	<Integer> equaling E0 6D FD 01
Example	20 7E returns 20 7E E0 6D FD 01

This command is used to calculate and set the default frame averaging output buffer address. THIS COMMAND SHOULD ONLY BE GIVEN IMMEDIATELY AFTER THE CALCULATE DEFAULT SUM BUFFER ADDRESS COMMAND.

3.7.1.9. Command: Write Calculate the Default Sum Buffer Address

Operation Code	20 80
Data	None
Response	<Integer> equaling C0 DB FA 01

This command is used to calculate and set the default frame averaging sum buffer address.

3.8. Bad Pixel Replacement Module Customization

This set of commands customizes how the camera discovers bad pixels during 1-point and 2-point calibrations.

3.8.1.1. Command: Write/Read BPR Floor Average

Operation Code	21 00
Data	<Integer> - the floor average for the cos^4 bad pixel replacement algorithm
Response	E0 01 – data is not an integer A0 01 – floor average set

This command is used to set the floor for the Bad Pixel Replacement algorithm. Default is 500.

Operation Code	21 01
Data	None
Response	<Integer> with current floor average.

3.8.1.2. Command: Write/Read BPR Threshold

Operation Code	21 02
Data	<Integer> - the threshold for the cos^4 bad pixel replacement algorithm
Response	E0 01 – data is not an integer A0 01 – threshold set

This command is used to set the threshold for the Bad Pixel Replacement algorithm. Default is 500.

Operation Code	21 03

Data	None
Response	<Integer> with current BPR threshold.

3.8.1.3. Command: Write/Read BPR Filename

Operation Code	21 04
Data	<String> - filename and absolute path ending with ".cos4" which provides a valid BPR calibration file
Response	E0 00 00 00– Data is too short and/or not null-terminated E0 01 00 00– No null terminating byte EX XX XX XX- see log A0 0A 00 00– BPR calibration file set

This command is used to set an alternate BPR calibration file. Recommended behavior is to use this only for test. Instead, update the default BPR calibration file and restart the camera.

Operation Code	21 05
Data	None
Response	<String> with current BPR calibration file

3.8.1.4. Command: Read BPR FPA Type

Operation Code	21 07
Data	None
Response	<Integer> - BPR Calibration File FPA Type (207, 209,etc)

This command is used to read the FPA Type from the BPR calibration file, providing what the calibration file **thinks** it is correcting.

3.8.1.5. Command: Read BPR Pixel Size

Operation Code	21 09
Data	None
Response	<Float> - BPR Calibration File FPA Pixel Size in microns

This command is used to read the FPA Pixel size from the BPR calibration file, providing what the calibration file **thinks** it is correcting.

3.8.1.6. Command: Read BPR Total Width

Operation Code	21 0B
Data	None
Response	<Integer> - BPR Calibration File FPA total width in pixels

This command is used to read the FPA total width from the BPR calibration file, providing what the calibration file **thinks** it is correcting.

3.8.1.7. Command: Read BPR Total Height

Operation Code	21 0D
Data	None
Response	<Integer> - BPR Calibration File FPA total height in pixels

This command is used to read the FPA total height from the BPR calibration file, providing what the calibration file **thinks** it is correcting.

3.8.1.8. Command: Read BPR F Number

Operation Code	21 0F
Data	None
Response	<Float> - BPR Calibration File optical system's F Number

This command is used to read the Optical system F Number from the BPR calibration file, providing what the calibration file **thinks** it is correcting.

3.9. Radiometry Camera Operations

3.9.1.1. Command: Load Radiometry Calibration File

Operation Code	15 00
Data	<String> - name of file
Response	E0 01 – File does not exist or error parsing E0 02 – Empty File E0 03 – Not a radiometry cal file A0 0A – successfully set

This command looks for the file name in /flash/radiometry/ and if found it will load calibration values. This value is stored when the state saved. If the first byte received is a null character, it will clear the stored filename so that a subsequent save state will not included a radiometry file load command in it.

Operation Code	15 01
Data	None
Response	String of loaded file or a single null character if no file is loaded

This returns the currently loaded radiometry calibration file.

3.9.1.2. Command: Load Radiometry Lens File

Operation Code	15 02
Data	<String> - name of file
Response	E0 01 – File does not exist or error parsing E0 02 – Empty File E0 03 – Not a radiometry lens file A0 0A – successfully set

This command looks for the file name in /flash/radiometry/lens/ and if found it would load lens values. This value is stored when the state saved. If the first byte received is a null character, it will clear the stored filename so that a subsequent save state will not included a radiometry file load command in it.

Operation Code	15 03
Data	None
Response	String of loaded file or a single null character if no file is loaded

This returns the currently loaded radiometry lens file.

3.9.1.3. Command: Lens Temperature

Operation Code	15 04
Data	<float> - Lens Temperature
Response	E0 01 – data is not 4 bytes E0 02 – data is outside range [? – ?] A0 01 – set the lens temperature

This command sets the lens temperature (K) which is outputted in the meta data.

Operation Code	15 05
Data	None
Response	<Float> - Lens Temperature

This returns the current lens temperature (K)

3.9.1.4. Command: Object Temperature

Operation Code	15 06
Data	<float> - Object Temperature
Response	E0 01 – data is not 4 bytes E0 02 – data is outside range [? – ?] A0 01 – set the lens temperature

This command sets the object temperature (K) which is outputted in the meta data.

Operation Code	15 07
Data	None
Response	<Float> - Lens Temperature

This returns the current object temperature (K)

3.9.1.5. Command: Object Emissivity

Operation Code	15 08
Data	<float> - Object Emissivity
Response	E0 01 – data is not 4 bytes E0 02 – data is outside range [? – ?] A0 01 – set the emissivity

This command sets the object emissivity which is outputted in the meta data.

Operation Code	15 09
Data	None
Response	<Float> - Lens Temperature

This returns the current object emissivity.

3.9.1.6. Command: Reflected Apparent Temperature

Operation Code	15 0A
Data	<float> - Reflected Apparent Temperature (K)
Response	E0 01 – data is not 4 bytes
	E0 02 – data is outside range [? – ?]
	A0 01 – set the Reflected Apparent Temperature

This command sets the Reflected Apparent Temperature (K) which is outputted in the meta data.

Operation Code	15 0B
Data	None
Response	<Float> - Reflected Apparent Temperature

This returns the current Reflected Apparent Temperature (K).

3.9.1.7. Command: Atmospheric Temperature

Operation Code	15 0C
Data	<float> - Atmospheric Temperature (K)
Response	E0 01 – data is not 4 bytes
	E0 02 – data is outside range [? – ?]
	A0 01 – set the Atmospheric Temperature

This command sets the Atmospheric Temperature (K) which is outputted in the meta data.

Operation Code	15 0D
Data	None
Response	<Float> - Atmospheric Temperature

This returns the current Atmospheric Temperature (K)

3.9.1.8. Command: Relative Humidity

Operation Code	15 0E
Data	<float> - Relative Humidity
Response	E0 01 – data is not 4 bytes
	E0 02 – data is outside range [? – ?]
	A0 01 – set the Relative Humidity

This command sets the relative humidity (%) which is outputted in the meta data.

Operation Code	15 0F
Data	None
Response	<Float> - Relative Humidity

This returns the current relative humidity (%)

3.9.1.9. Command: Object Distance

Operation Code	15 10
Data	<float> -Object distance
Response	E0 01 – data is not 4 bytes E0 02 – data is outside range [? – ?] A0 01 – set the Object distance

This command sets the Object distance in meters which is outputted in the meta data.

Operation Code	15 11
Data	None
Response	<Float> - Object distance

This returns the current Object distance in meters

3.10. Sync Selective Integration Time Control

Enabling sync selective integration time is done with the JamSync command (10 30). In this mode, the length of the external sync in pulse determines the integration time for the next frame. There are 8 selectable slots which correspond to input pulse lengths of 100us to 800us (+/-10us) in 100us increments. If the input pulse length is valid, the corresponding slot's integration time (set with 10 E2) will be loaded for the next frame. If the input length is invalid, there will be no change to the integration time.

3.10.1.1. Command: Sync Selective Integration Select

Operation Code	10 E0
Data	<Byte> Integration time slot select 0 - 7
Response	A0 0A – set integration time selector E0 01 – Time slot out of range or no data

This command selects which slot to edit when issuing the integration time slot command or NUC slot command.

Operation Code	10 E1
Data	None
Response	<Byte> - integration time slot selected

This returns the current selected slot

3.10.1.2. Command: Sync Selective Integration Time

Operation Code	10 E2
Data	<Float> Integration time
Response	A0 0A – set integration time E0 01 – Data is not a valid float

This command sets the integration time for the currently selected slot

Operation Code	10 E3
Data	None
Response	<Float> Integration time

This returns the integration time of the currently selected slot.

3.10.1.3. Command: Sync Selective NUC Slot

Operation Code	10 E4
Data	<Byte> NUC slot (0 – 11)
Response	A0 0A – set NUC slot E0 01 – No data or out of range

This command sets the NUC slot for the currently selected integration slot.

Operation Code	10 E5
Data	None
Response	<Byte>

	NUC Slot

This returns the NUC slot of the currently selected integration slot.

3.11. Bi-linear Interpolation Output Control

The bilinear interpolation is used in the camera for digital zoom capability. The interpolation output is a maximum of 1280x1024, so zooming past this window will lose information.

3.11.1.1. Command: Bi-linear Interpolation Enable

Operation Code	21 40
Data	<Byte> 1 – enable 0 – disable
Response	E0 01 – data is not 1 byte A0 01 – set the enable

This command enables or disables the bilinear interpolation. Note that it should be paired with the multiplexor selection of the desired output.

Operation Code	21 41
Data	None
Response	<Byte> - Enable status

3.11.1.2. Command: Bi-linear Interpolation Frame Rate

Operation Code	21 42
Data	<Byte> 0 – 15 Hz 1 – 30 Hz 2 – 45 Hz 3 – 60 Hz
Response	E0 01 – data is not 1 byte A0 01 – set the frame rate

This command sets the frame rate of the output of the Bilinear interpolation. This is the rate at which frames are removed from the frame buffer by the interpolation. Should be kept at 30 Hz as there is no compelling reason to change.

Operation Code	21 43
Data	None
Response	<Byte> - Interpolation frame rate setting

3.11.1.3. Command: Bi-linear Interpolation Fixed window

Operation Code	21 44
Data	None
Response	A0 01 – sets up interpolation for given window size

This command sets up the bilinear interpolation frame retrieval for the current fixed window size. First, this command calculates the required offsets, scaling factor, and other variables, then it turns the interpolation on. This is a convenient command if you don't wish to hand jam the values into the register, and wish the current window to enlarge to fill a 1280x1024 window.

Note that the scaling factor is the same in both directions to maintain the same height to width ratio as the original frame.

3.11.1.4. Command: Bi-linear Interpolation Original Frame Number of Columns

Operation Code	21 50
Data	<Integer > value of the number of columns in the original frame
Response	E0 01 – data is not 4 bytes or an integer A0 01 – set the number of columns

This command sets the number of columns in the original frame, not the interpolation resulting frame. This is used to calculate the scaling factor and expected linear differentiation.

Operation Code	21 51
Data	None
Response	<Integer> - number of columns

3.11.1.5. Command: Bi-linear Interpolation Original Frame Number of Rows

Operation Code	21 52
Data	<Integer > value of the number of rows in the original frame

Response	E0 01 – data is not 4 bytes or an integer A0 01 – set the number of rows

This command sets the number of rows in the original frame, not the interpolation resulting frame. This is used to calculate the scaling factor and expected linear differentiation.

Operation Code	21 53
Data	None
Response	<Integer> - number of rows

3.11.1.6. Command: Bi-linear Interpolation Frame Number of Columns

Operation Code	21 54
Data	<Integer > value of the number of columns in the new frame, generally set to 1280
Response	E0 01 – data is not 4 bytes or an integer A0 01 – set the number of columns

This command sets the number of columns in the interpolated frame. This is used to calculate the scaling factor and expected linear differentiation.

Operation Code	21 55
Data	None
Response	<Integer> - number of columns

3.11.1.7. Command: Bi-linear Interpolation Frame Number of Rows

Operation Code	21 56
Data	<Integer > value of the number of rows in the new frame, generally set to 1024
Response	E0 01 – data is not 4 bytes or an integer A0 01 – set the number of rows

This command sets the number of rows in the interpolated frame. This is used to calculate the scaling factor and expected linear differentiation.

Operation Code	21 57
Data	None

Response	<Integer> - number of rows

3.11.1.8. Command: Bi-linear Interpolation Frame Number of Offset Columns

Operation Code	21 58
Data	<Integer > value of the number of offset columns in the new frame
Response	E0 01 – data is not 4 bytes or an integer
	A0 01 – set the number of offset columns

This command sets the number of offset columns in the interpolated frame. This is used to calculate the scaling factor and expected linear differentiation.

Operation Code	21 59
Data	None
Response	<Integer> - number of offset columns

3.11.1.9. Command: Bi-linear Interpolation Frame Number of Offset Rows

Operation Code	21 5A
Data	<Integer > value of the number of offset rows in the new frame
Response	E0 01 – data is not 4 bytes or an integer
	A0 01 – set the number of offset rows

This command sets the number of offset rows in the interpolated frame. This is used to calculate the scaling factor and expected linear differentiation.

Operation Code	21 5B
Data	None
Response	<Integer> - number of offset rows

3.11.1.10. Command: Bi-linear Interpolation Original Frame Offset Columns

Operation Code	21 5C

Data	<Integer > value of the number of original offset columns
Response	E0 01 – data is not 4 bytes or an integer A0 01 – set the number of original offset columns

This command sets the number of offset columns in the original frame. This is used to calculate the scaling factor and expected linear differentiation.

Operation Code	21 5D
Data	None
Response	<Integer> - number of offset columns

3.11.1.11.Command: Bi-linear Interpolation Original Frame Offset Rows

Operation Code	21 5E
Data	<Integer > value of the number of offset rows
Response	E0 01 – data is not 4 bytes or an integer A0 01 – set the number of original offset rows

This command sets the number of offset rows in the original frame. This is used to calculate the scaling factor and expected linear differentiation.

Operation Code	21 5F
Data	None
Response	<Integer> - number of original offset rows

3.11.1.12.Command: Bi-linear Interpolation Max Columns

Operation Code	21 60
Data	<Integer > value of max columns
Response	E0 01 – data is not 4 bytes or an integer A0 01 – set the max number of columns

This command sets the maximum number of columns in the interpolated frame. This is used to calculate the scaling factor and expected linear differentiation.

Operation Code	21 61
Data	None
Response	<Integer> - number of maximum columns

3.11.1.13. Command: Bi-linear Interpolation Max Rows

Operation Code	21 62
Data	<Integer > value of the number of offset rows
Response	E0 01 – data is not 4 bytes or an integer A0 01 – set the number of original offset rows

This command sets the maximum rows in the original frame. This is used to calculate the scaling factor and expected linear differentiation.

Operation Code	21 63
Data	None
Response	<Integer> - number of max rows

3.11.1.14. Command: Bi-linear Interpolation X-Ratio

Operation Code	21 64
Data	<Float> value of the x-ratio
Response	E0 01 – data is not 4 bytes A0 01 – set the x-ratio

This command sets the ratio in the X direction. This is generally calculated by dividing the original number of columns (less any offset) by the new number of columns in the interpolated image.

Operation Code	21 65
Data	None
Response	<Float> - x-ratio

3.11.1.15. Command: Bi-linear Interpolation Y-Ratio

Operation Code	21 66
Data	<Float> value of the y-ratio
Response	E0 01 – data is not 4 bytes or an integer A0 01 – set the y-ratio

This command sets the ratio in the Y direction. This is generally calculated by dividing the original number of rows (less any offset) by the new number of rows in the interpolated image.

Operation Code	21 67

Data	None
Response	<Float> - y-ratio

3.11.1.16.Command: Bi-linear Interpolation Scale Factor

Operation Code	21 68
Data	<Float > scale factor
Response	E0 01 – data is not 4 bytes or an integer A0 01 – set the scale factor

This command sets the scale factor used to scale the original image in the X and Y directions.

Operation Code	21 69
Data	None
Response	<Float> - number of max rows

4. Examples of Common Tasks on the 1280SciCam

4.1. Setting up the Digital Focalplanes

When the electronic board set first powers on, the main signals to the focalplane are at ground. This includes the biases, the serial control interface, and the video control interface. The biases are comprised of five DC signals: VPOS, VPH, VPL, VPOUT, and VDETCOM. Only four biases are controlled via the software interface. The VPH and VPOS biases are tied together, and thus constitute one controllable bias. The VCOMMON bias is as the VDETCOM bias when the jumper is set; otherwise the VDETCOM bias is also tied to VPOS (common case). Setting VDETCOM equal to VPOS allows "back bias" to be put on the readout, which can be useful for InGaAs. The serial control interface is a 3-wire bus consisting of a serial clock, serial data in, and serial data out. The video control interface is comprised of two signals going to the FPA, the reference clock and the sync signal, and four LVDS video data channels output from the FPA.

Under common conditions, the first step to turn on the digital focalplanes is to set up the biases to specific voltages in a specific order:

(1) Set VPOS/VPH to 3.3V (Opcode: 10 00)

(2) Set VCOMMON/VDETCOM to required voltage, usually 0.0V (Opcode: 10 10)

(3) Set VPL to 1.6V (Opcode: 10 0C)

(4) Set VPOUT to 0.8V (Opcode: 10 08)

Timing during setting up these biases isn't an issue, since the commands internally force the minimum timing. Other material/readouts may have a different bias sequence. See the readout ICD for more information.

After setting up the biases, the next step is set up the video control interface by starting the reference clock. This is the main clock for the FPA operation, and is nominally set to 16.5MHz. When cold, the focalplanes can be "overclocked" to 20MHz, which is what is normally sent. *To see video data during warm operation, the reference clock needs to be less than 16.5MHz.* To set the frequency of the reference clock use opcode 10 2A. Once a frequency has been selected, use the opcode 10 28 to enable the reference clock to be output from the electronics to the FPA. It is best at this point to leave the synchronization pulse low (the default) until we update the FPA registers with the setting we would like to use.

Before reading or writing the FPA registers it is highly recommended to familiarize and/or actively use the readout ICD.

To begin setting up the readout for the desired behavior, the first step is to reset the communications link with the FPA, as the serial interface is in an unknown state. The opcode 10 5E performs this action. To return the FPA to a known state, use the opcode 10 5C to reset the FPA. Since the synchronization is not being accomplished, set the immediate bit to high using the 10 62 opcode, in order to actually change the registers. To read individual registers from the FPA, use the 10 61 command. This is also a useful command to verify that communications with the FPA are taking place. To write individual registers, use the 10 60 command. Most of the common operations on the focalplane have been wrapped in their own commands, such as setting frame time (opcode: 10 6E) or setting integration time (opcode: 10 6C). For more such commands, check the Basic Camera Operation Commands in the Science Camera Command List.

To output a signal synchronous with the start of integration time, be sure to use the 10 3C command and be sure to set the digital BIT output on the FPA to output integration time using the correct control registers. Usually this is done by sending the 10 60 25 23 command, but may very for newer FPAs.

Finally, be sure to disable the immediate communications mode, since normally you want changes made to FPA operation to occur on the next frame, not the current one. Finally, set the FPA synchronization signal to be constantly high (using opcode 10 30), and then control the frame frequency using the frame time to set the period. Alternatively, set the synchronization so that it pulses at the desired frame rate or on an input signal.

4.2. Camera Link Output Configuration

The Camera Link output on the Science Camera can be configured a number of ways depending on the end user application. The most common method of output is "base" Camera Link. Either single or dual channel operation from the focalplane can be output across base Camera Link mode. Since the second Camera Link connector is not in use during base mode, there is an alternative "dual" base mode option that allows video output on the second connector. For more information, see opcode 20 60. The image on the second connector may also be modified so that it is different than the video output on the first connector, see opcode 20 62.

In the final mode, “full” Camera Link can be output from the camera, allowing up to 5.12Gbps output from the electronics. For most SBF digital focalplanes, this is far more than actually required, about 320MBps. To switch between “full” and “base” Camera Link output modes, use the opcode 10 CA to set the number of output channels. Setting the number of channels to one or two will automatically set the Camera Link output into “base” mode, while setting the number of channels to four will set the output into “full” mode.

The Camera Link output is constantly on and cannot be turned off.

4.3. Enabling Frame Averaging

Frame averaging is one of the more complex blocks in the 1280SciCam system. This is mainly due to the number of options, and the requirement of being memory-aware through the frame capture block. Frame averaging uses the upper 2*i* of the frame capture (256MB) memory, where *i* is the size of a single video frame. Because the frame capture state machine uses the lower *Ni*, where *N* is a user-supplier number of frames, to store the frames captured for the Pleora and HDMI video outputs. Normally, this is computed by the software, but the frame averaging offsets can be set by the user. In this case, be sure not to overlap the frame capture memory!

Frame averaging can currently be set into 3 modes: off (default), single-shot, and continuous block. The single-shot mode performs exactly as it sounds: the firmware captures *k* frames, averages them, then leaves the averaged frame in the frame capture memory at the specified output offset. This mode is used when generating a NUC table in the electronics set. The other interesting mode is the continuous block mode, which averages *k* frames, outputs the result to the video pipeline and the frame capture memory at the specified offset, and continuously performs this operation on the next *k* block of frames until disabled. This allows for interesting downstream applications by mimicking a long integration time. It also provides a method for accumulating a much higher frame rate into a lower-bandwidth interface. Note that no image registration is provided – this is a simple pixel-averaging operation.

1. Set the number of frames to average. (Opcode: 20 74)
2. Set the number of input channels. (Opcode: 10 C0)
3. Set the number of output channels, this can me more, less, or equal to the number of input channels. (Opcode: 10 C2)
4. Set the NUC input mux if the output is to be NUC'd. (Opcode: 20 1C)
5. Set the number of NUC channels equal to the number of output channels (if going to NUC). (Opcode: 10 C4)
6. If not using the defaults, set the buffer offsets. (Opcodes: 20 76, 20 78)
7. Set the desired interface output mux to more than 1. (Opcodes: 20 02, 20 04)
8. Set to the desired frame averaging mode. (Opcode: 20 70)

Note that the input and output number of channels are not required to match. Thus downsampling (full bandwidth from the sensor, but base bandwidth from the frame averaging module) is possible. This is useful to users who only have access base Camera Link framegrabber options, but want to use the full bandwidth of the sensor. Note that it is possible to have raw output on one base Camera Link and frame averaged output on another in dual Camera Link mode, but that only one of those streams can be NUC'd.

4.4. Enabling Non-Uniformity Correction

Once raw video is being output across the desired interface, it is often desirable to remove the non-uniformity static noise in the image. This can be done by applying a per-pixel linear correction to the video image. In order to apply non-uniformity correction to the video stream, first create a Science Camera NUC file using two temperature sources and either the software or hardware NUC file creation options. Then load the file onto the camera, if using the software option by sending the NUC file to the "/flash/nuc" directory. Once that has been accomplished, the NUC file can be loaded.

1. Write the desired NUC file into one of the 12 NUC slots available. (Opcode: 20 32)
2. Set the slot that the file was written into as the "active" slot. (Opcode: 20 30)
3. Ensure that the number of channels the NUC operates on is equal to the sensor and output channels. (Opcode: 10 C4)
4. Enable the Non-uniformity correction. (Opcode: 20 10)
5. If desired, also enable the Bad Pixel Replacement. (Opcode: 20 12)
6. If sending NUC output to the Camera Link, set the Camera Link output mux. (Opcode: 20 02)
7. If sending NUC output to HDMI or Ethernet, set the Frame Capture mux. (Opcode: 20 04)

4.5. Performing On-board NUC Calibration

There are multiple options for performing on-board non-uniformity correction calibration on the camera: (1) update an existing or generate a new set of offset coefficients through a one-point update operation, (2) update an existing or generate a new set of offset coefficients and update the bap pixel information through a one-point calibrate operation, or (3) generate a new set of offset and gain coefficients through a two-point calibration operation.

The simplest operation is the one-point update. Since no new bad pixel information is being generated, it simply updates the offset coefficients of the active NUC slot based upon the bad pixel information provided by the slot. One-point update works best if the video input is a uniform temperature field.

1. Set the slot that the file was written into as the "active" slot. (Opcode: 20 30)
2. Set the number of frames to average for the NUC. (Opcode: 20 3E)
3. Set the frame averaging number of channels to the incoming number of channels. (Opcode: 10 C2)
4. Perform the 1 point update. (Opcode: 20 36)

It should be noted that if the default sum buffer address or output buffer address have been altered, the output will probably be incorrect. To reset to the default addresses, use the calculate sum buffer (opcode: 20 80) and calculate output buffer commands (opcode: 20 7E).

The calibration commands differ from the update commands since they also attempt to detect bad pixels in the incoming video stream and predict the best method for masking that pixel. The correction mask is a 5x5 window that allows averaging over a subset of the inner 3x3 window. For more information, see the table below, which shows the 5x5 window over a pixel and the corresponding replacement codes.

	4	3	2	1	0
4	88	87	86	85	84
3	78	77	76	75	74
2	68	67	XX	65	64
1	58	57	56	55	54
0	48	47	46	45	44

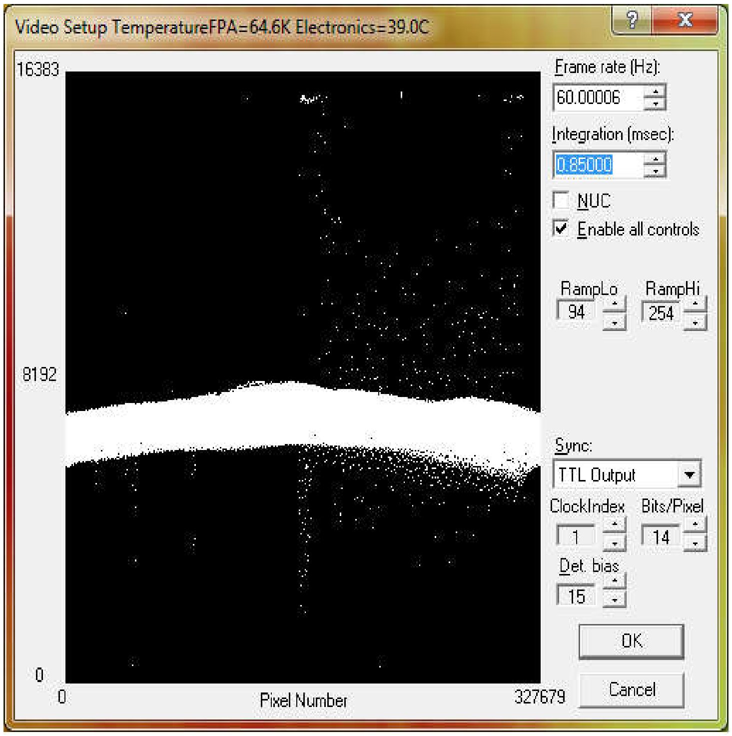
- 0xA0 – Average N, S, W, and E
- 0xA1 – Average N, S
- 0xA2 – Average E, W
- 0xA3 – Average NW, SW, NE, SE
- 0xA4 – Average NE, SW
- 0xA5 – Average NW, SE

Note: All codes in hexadecimal

Any code not in the table indicates that the pixel should be passed through without change.

To determine bad pixels, the camera uses an algorithm that automatically corrects for lens effects and random noise, then uses thresholding to attempt to determine bad pixels.

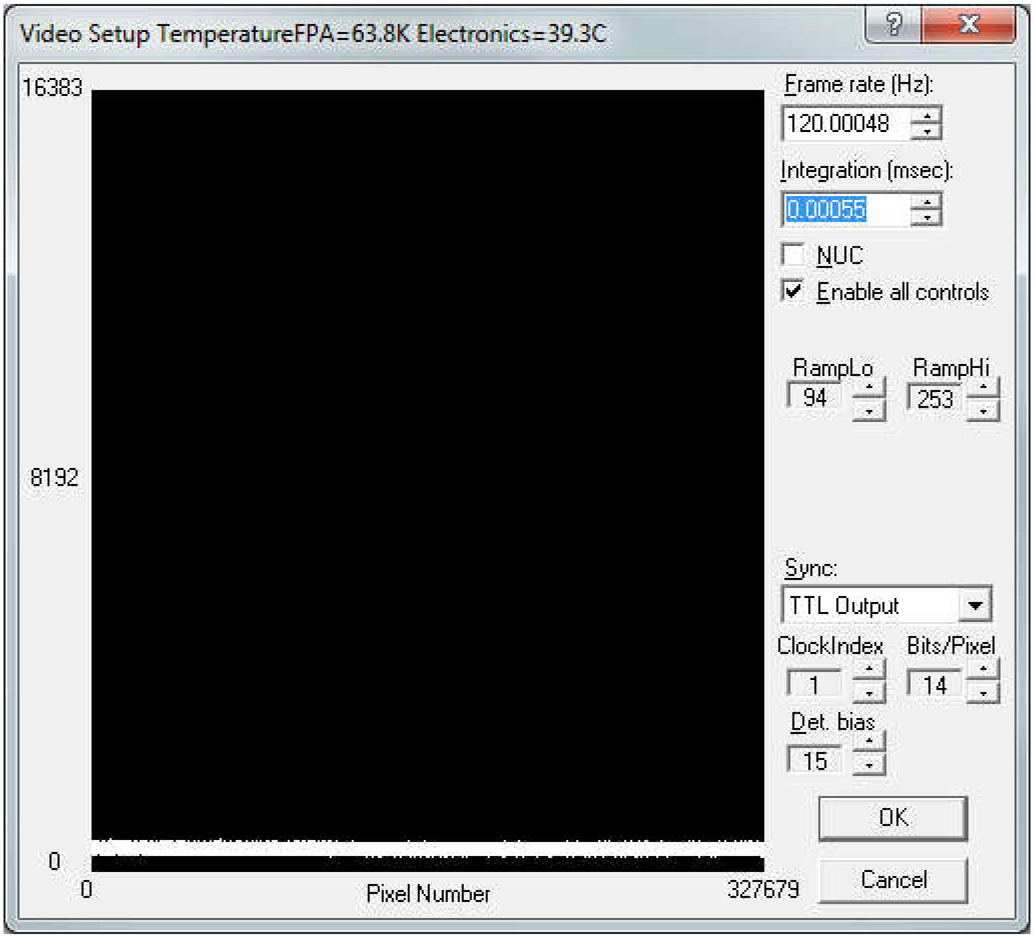
When looking at the video output from a standard Science Camera, vignetting is clearly visible in some of the larger arrays. In fact, due to lensing effects, the light falls off in intensity from the center of the image at a $\cos^4$ relationship. The visual effect is captured in WinIRC in the video oscilloscope:



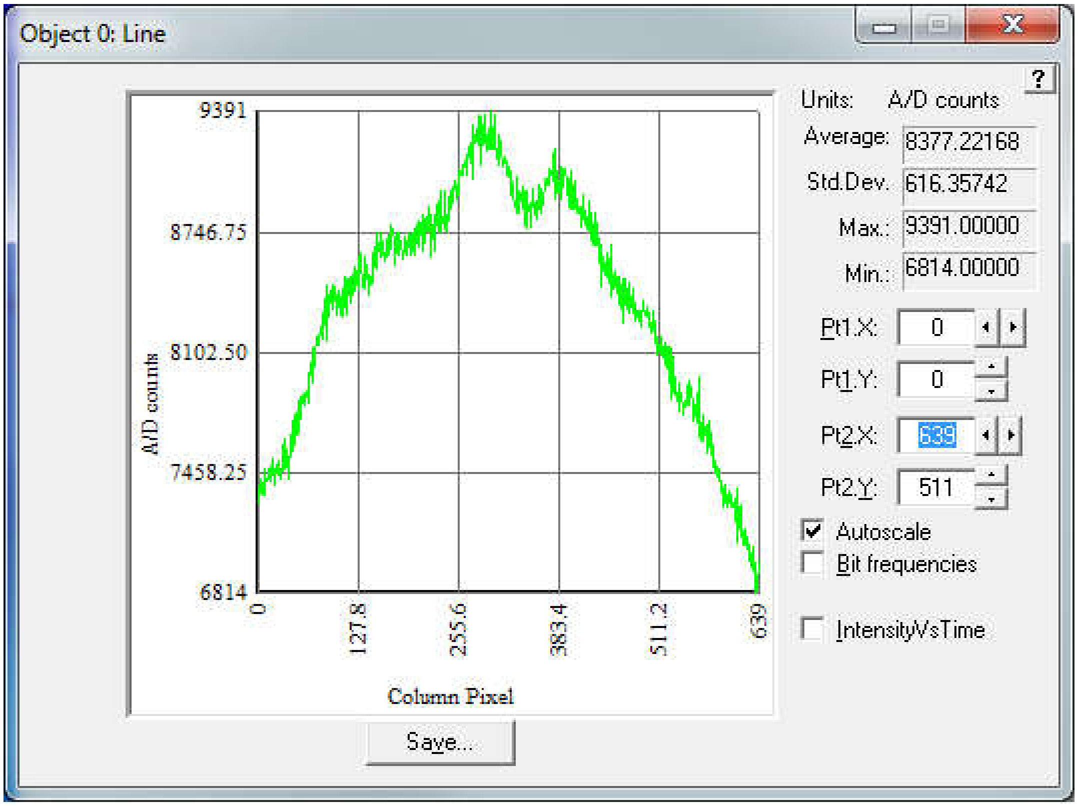
The image clearly shows that the intensity of the response is curved, with the highest response in the center of the image. In a well-centered image with a correctly thinned FPA, this can be readily seen when the FPA is presented a uniform scene. Concentric rings will appear in WinIRC as the equalization and color mapping algorithms attempt to depict the gradual degradation in intensity.

To find bad pixels, the algorithm takes the global average of the image – which represents the ideal response of every pixel to the constant-temperature scene – and compares it to each individual pixel. In order to better identify bad pixels, the algorithm "flattens" the $\cos^4$ response, using a predetermined correction factor based upon the camera's and FPA's unique geometries. This is the initial "expected" response. From this expectation, the algorithm subtracts the global noise estimation, which yields the final expectation value for each pixel. If this final expectation value is more or less than the required threshold, the pixel is marked as bad and the algorithm later attempts to find the subset of the 5x5 correction window which best matches its expected response.

So in order to get a good response from the bad pixel replacement algorithm, the user needs to understand the basics of how to estimate the global noise and the best values for thresholds. In order to approximate the global noise, take the integration time to its minimum value:



The values along the floor represent the changing noise. To get the global average, use the rectangle drawing tool and find the mean value reported. That is a good approximation for the global noise. To find a good approximation for the threshold, use the line drawing tool to draw a line from one upper corner to the opposite lower corner. This yields a response that will look something like the image below:



From this image, we can clearly see that the delta from the mean to the minimum value (approximately 1550 counts) is greater than the delta from the mean to the maximum value (approximately 1000 counts). So a good estimate for the threshold is 1550 counts (the larger of the two values).

To perform a one-point calibration, provide the FPA with a uniform scene and perform the following steps:

1. Set the slot that the file was written into as the "active" slot. (Opcode: 20 30)
2. Set the number of frames to average for the NUC. (Opcode: 20 3E)
3. Set the frame averaging number of channels to the incoming number of channels. (Opcode: 10 C2)
4. Set the global noise value. (Opcode: 21 00)
5. Set the bad pixel threshold. (Opcode: 21 02)
6. Perform the 1 point calibration. (Opcode: 20 38)

To perform 2 point calibration, the steps are a little more complicated:

1. Set the slot that the file was written into as the "active" slot. (Opcode: 20 30)
2. Set the number of frames to average for the NUC. (Opcode: 20 3E)

3. Set the frame averaging number of channels to the incoming number of channels. (Opcode: 10 C2)
4. Set the global noise value. (Opcode: 21 00)
5. Set the bad pixel threshold. (Opcode: 21 02)
6. Provide the FPA with a uniform temperature scene.
7. Collect the first sample set. (Opcode: 20 3A)
8. Provide the FPA a uniform scene of a higher temperature.
9. Collect the second sample set. (Opcode: 20 3C)
10. Perform the 2 point calibration. (Opcode: 20 82)

Any error codes that return due to an excessive number of bad pixels can usually be corrected by widening the threshold value. If the error continues to occur, check that you have the latest version of the cos^4 file.

4.6. Camera States

The 1280SciCam allows multiple states to be saved to its 512MB Flash hard disk. These states can be loaded at startup, to have the camera in a known state during initial operation, or during operation, to return the camera to a known state. In general, set the camera operation to a desired states, then set the state using the opcode 01 00. To load a camera state, use the opcode 01 01. Setting the startup state is a special operation, also requires an opcode: 01 02. If the camera ever becomes unresponsive to commands, but is communicating, you can return to the default factory preloaded states using the opcode 01 04.

4.7. Metadata

The following table details the structure of the metadata in the image stream if metadata is enabled on the camera. Each pixel is 16 bits with the most significant byte sent on Camera Link Port B and the least significant byte sent on Camera Link Port A. Floats and integers are comprised of two pixels with the most significant data contained within the first pixel. Strings are ordered so that the first character is the most significant byte of each pixel. Since the data will most likely be stored in little endian format on the frame grabber computer, direct byte conversion to int, float, or string will not be possible. The following shows examples for string and int.

```
/* Conversion of to int */

int value = (pixel[0] << 16) | (pixel[1]);

/* Conversion to string.  All strings have a max length, shown in the table below */

char str[num_words]

char* ptr = str;

for (int i = 0; i < num_words; ++i)
```

```
{
    if (pixel[i] != 0)
    {
        *ptr++ = pixel [i] >> 8;
        *ptr++ = pixel [i] & 0xff;
    }
}
```

Registers 128-191 are mirrors of the FPA registers in this space (non-used registers are 0's). See additional ICDs for FPA register details and descriptions of metadata use. In the table below the green sections are not fully operational in the 1280SciCam and may be activated in future versions. We have displayed them here as the user could allow for them in future versions.

Byte	Contents
0-1	0#AC - Marker for Packet beginning -- where # can be 00, 01, 10, 11 dependent upon position
2-33	Part number in ASCII (32 characters -- a null terminated string in here)
34-47	Serial Number (14 chars ASCII number -- null-terminated string)
48-63	Type of FPA (16 char ASCII string -- null terminated)
64-67	4 byte CRC, 802.3 Ethernet definition (first line CRC value is always 0; last line CRC value is calculated for image data only). This is a UINT32.
68-71	**4 byte frame counter. Please display frequency (1/counter) in the software in Hz.**
72-75	**4 byte float - frame time in seconds**
76-79	**4 byte float - integration time in seconds**
80-83	4 byte float - frequency to the FPA in Hz (REFCLK)
84-87	4 byte float - VPH bias
88-91	4 byte float - VPH source
92-95	4 byte float - VPL bias
96-99	4 byte float - VPL source
100-103	4 byte float - VPOUT bias
104-107	4 byte float - VPOUT source
108-111	4 byte float - VPOS bias
112-115	4 byte float - VPOS source
116-119	**4 byte float - FPA Temp Diode Voltage. IRC will provide information in the future on how to format this - for now it is TBD.**
120-123	4 byte float - Extra Temp 000 - Board Temp
124-125	**Data is RAW/NUC'd - 0x4E00 if NUC'd, 0x5200 if RAW DATA.**
126-127	**Filter wheel position/code**
128	FPA Reg00 = SYNC (sent after power on glitch to re-sync comm; typically is 0)
129	FPA Reg01 = COMM (Communication configuration; typically is 0)
130	**FPA Reg02 = COFF0 (Column Window Offset, low byte).**
131	**FPA Reg03 = COFF1 (Column Window Offset, high byte)**

132	**FPA Reg04 = CWS0 (Column Window Size minus 1, low byte)**
133	**FPA Reg05 = CWS1 (Column Window Size minus 1, high byte)**
134	FPA Reg06 = HB0 (Extra Horizontal Window Blank Pairs, low byte; normally 0)
135	FPA Reg07 = HB1 (Extra Horizontal Window Blank Pairs, high byte; normally 0)
136	**FPA Reg08 = ROFF0 (Row Window Offset, low byte)**
137	**FPA Reg09 = ROFF1 (Row Window Offset, high byte)**
138	**FPA Reg10 = RWS0 (Row Window Size minus 1, low byte)**
139	**FPA Reg11 = RWS1 (Row Window Size minus 1, high byte)**
140	FPA Reg12 = 0
141	FPA Reg13 = 0
142	FPA Reg14 = IT0 (Integration Time Ticks, byte 0)
143	FPA Reg15 = IT1 (Integration Time Ticks, byte 1)
144	FPA Reg16 = IT2 (Integration Time Ticks, byte 2)
145	FPA Reg17 = IT3 (Integration Time Ticks, byte 3; IntegrationTime reported in metadata bytes 76-79 is equal to IntegrationTimeTicks times Frequency as given in metadata bytes 80-83; actual integration time is about 12 ticks shorter)
146	FPA Reg18 = FT0 (Frame Time Ticks, byte 0)
147	FPA Reg19 = FT1 (Frame Time Ticks, byte 1)
148	FPA Reg20 = FT2 (Frame Time Ticks, byte 2)
149	FPA Reg21 = FT3 (Frame Time Ticks, byte 3; FrameRate = 1/FrameTime where FrameTime = FrameTicks * Frequency where Frequency is given in metadata bytes 80-83)
150	FPA Reg22 = VDBA (FPA Detector Bias Adjust)
151	FPA Reg23 = 0
152	FPA Reg24 = 0
153	FPA Reg25 = 0
154	FPA Reg26 = VHI (ADC Ramp High Bias; sets Full Well Output Level)
155	FPA Reg27 = 0
156	FPA Reg28 = VLO (ADC Ramp Low Bias; set Empty Well Output Level)
157-166	Additional FPA Registers; See individual FPA ICDs
167	**FPA Reg39 = CONF1** **Bits defined as follows for SBF207 & SBF204 only(!): Bit0=JamSync before reading out, Bit1=JamSync at edges of A/D cycles, Bit6=OM0, Bit7=OM1, where OM is OutputMode:** **1 Output => OM0=0, OM1=0** **2 Output => OM0=1, OM1=0** **4 Output => OM0=0, OM1=1** **Grayess to display 1/2/4 channel only (bit6 & 7).**
168-191	Additional FPA Registers; See individual FPA ICDs
192-193	**Relative Timer Data - Year; Relative times are initialized to 0 when power is turned on; use FUTURE-TBD software to set these values to real time values**
194-195	**Relative Timer Data - Day of the Year**
196-197	**Relative Timer Data - Hour**
198-199	**Relative Timer Data - Minute**
200-201	**Relative Timer Data - Second**
202-203	**Relative Timer Data - Millisecond**
204-205	**Relative Timer Data - Microsecond**
206-207	Relative Timer Source/Sync
208-243	Reserved
244-267	User defined fields (24 bytes)

268-331	GPS ASCII String
332-351	GSP Reserved
352-475	Radiometry Coefficients
476-479	4 byte float – FPA Temperature
480-483	4 byte float – Temperature Sensor 1
484-487	4 byte float – Temperature Sensor 2
488-491	4 byte int – Cooler Runtime Minutes
492-493	2 byte short – Encoder Counts
494-527	RESERVED
528-529	F1AC -- Marker for packet ending
530-639	RESERVED TO MAX

5. FPA Control Documentation

The standard configuration for the 1280SciCam is a 3 stage TEC cooled camera it includes a 1280x1024 InGaAs focal plane array, vacuum window, Nikon F- and C-mount optical lens adapters, support electronics, TEC control circuity, universal AC-to-DC brick power supply, and power supply cord. The 1280SciCam uses a two dimensional InGaAs focal plane array. It is a hybrid device involving a lattice matched $In_{.53}Ga_{.47}As$ p-i-n detector (InGaAs) array attached to a silicon CMOS readout integrated circuit (ROIC). Light strikes the detector array and is converted to electron-hole pairs. This charge is then collected, amplified and stored by the ROIC on a pixel by pixel basis. Each pixel in the ROIC has a capacitive transimpedance amplifier (CTIA) to amplify and capture the charge, this is called integrating the charge. The analog signal is then converted to a digital signal on the ROIC with a 14 bit analog to digital converter (A/D). This imager is a snapshot imager meaning all of the pixels capture signal simultaneously and then they are all read out. The imager does have a read while integrate mode so while one frame is being output by the FPA the next frame can be capturing signal.

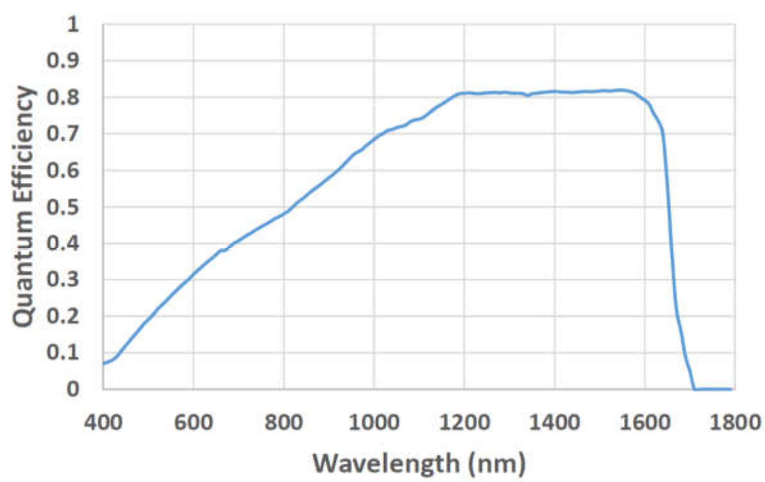


Figure 2. Typical quantum efficiency curve for PIRT InGaAs arrays.

This FPA is a 1280x1024 imager meaning it has 1.31Mpixels. Each of these pixels are spaced on a 12um pitch. The entire array can be read out or a window or region of interest (ROI) can be read out instead. The advantage of the ROI is it allows one to reach a higher frame rate. The imager has one to four outputs each maximized at 560Mpixels/s. The increase in the frame rate is not linear to the number of pixels readout. The window size to frame rate is discussed in more detail in section 6.1. Data from the array of pixels are multiplexed to a multi-line A/D that converts the signal to 14 or 13 bits depending on the setting. Each analog pixel has ping pong circuitry to enable one line to be read while another is converting charge thus enabling the snapshot imagery.

6. Specifications

All specifications list in Table 1 were conducted at 25C.

Some specifications will improve at lower temperatures.

Table 1: FPA Specifications

Parameter	Unit	Min	Typical	Max	Comments
Resolution	Pixels		1280x1024		
Pixel Pitch	μm		12		
Frame Rate 1280x1024 640x512 312x128	Frames/s	95		105 382 1490	
Data Output	Bits	13		14	User selectable
Peak Wavelength	μm		1.55		
Fill Factor	%	99		100	
Responsivity	Photoetectron count	2.5	3	4.5	
Integration time	Seconds	100e-6		>0.08	Dependent on array setpoint, this is for 25C
Dark Signal Rate 25C OC -50C	Electrons/second		 20k 140	250k	
Read Noise	Electrons		28	30	
Full Well	Electrons	46k	50k		
Inoperable pixels	%		0.4	0.6	

7. Timing

7.1. Windowing functions

The PIRT1280A1-12 SWIR FPA is a digital 1280 x 1024 pixel InGaAs FPA with a CTIA input and the ability to limit the output data to a desired window size and position.   (Window size and position can be specified in 1 x 1 pixel increments.) The window can be as small as 312 cols x 128 rows up to the full 1280 cols x 1024 rows. Thus, the frame rate can be increased with smaller window sizes without an inordinately high output data rate. The FPA can be configured to operate with 1, 2, or 4 digital output ports, with each limited to a maximum of 560 Mbits/second. Highest frame rates are achieved with four output ports enabled. Two line start characters (pixels) are added to each port, for a total of 8 in 4 port mode. Some example frame rates and window sizes for 14 bit and four port mode are shown in Table 8 below.

Table 8: Typical Maximum Frame Rates at 25C

window			FR	Out	Refclk	A/D	HB
col		row	(Hz)		(MHz)	13/14	(Pixels)

1280	X	1024	95	4	16.5	14	0
1024	X	1024	120	4	16.5	14	0
640	X	512	365	4	16.5	14	0
312	x	128	1450	4	16.5	14	0

7.2. Register Consistency

All registers are double buffered to enable consistent switching between operational modes. This method prevents the FPA from using a 16 bit register when only the first byte has been written. Also, multiple registers may have to be set up in a consistent manner before allowing the FPA to utilize the new programmed state. This is accomplished using two methods: the IMM bit and a combination of INH_EOI and INH_FS bits in the com register.

If the IMM bit is at 0 in the normal serial data command then the FPA will not start using the new value until after the next end of integration (or frame start from the frame time register). If IMM is set to 1 then the register changes immediately and the FPA is effected immediately.

If the INH_EOI and INH_FS bits are both set to 1 multiple register can be preloaded with the possibility of an end of integration (or frame start) arriving and transferring the new data to the destination registers. Once the collection of new registers is completed the INH_EOI and INH_FS bits are cleared and the FPA is switched into its new state smoothly with no dropped frames between the two FPA states at the next end of integration (or frame start for the frame time register). If a 16 bit register is used then both bytes can be updated simultaneously with the IMM bit even though they are still address separately. The SCLK can run as high as 100kHz providing sufficient time to change the entire FPA operating state on a frame by frame basis.

7.3. Register Assignments

The register assignments are shown in TABLE 9. The registers control frame length (which determines frame rate), integration time, window size as well as window position, A/D conversion to 14 or 13 bits as well as detector bias and other converter rail adjustments.

Table 9: Available Register Command Assignments

Reg No. (6 bits)	Name	Preload (8 bits)	Function
0	SYNC	N/A	Send after power-on or glitch to resync communications Send two null bytes. Then IMM write to addr 0 w/ 0 data.
1	COM	0	Communication Configuration
2	COFF0	8	Column Offset Byte 0
3	COFF1	0	Column Offset Byte 1

	4	CWS0	255	Column Window Size Byte 0
	5	CWS1	4	Column Window Size Byte 1. Bit15 = Horizontal Reflection
	6	HB0	0	Horizontal Blank Byte 0. (Extra Blank Column Pairs.)
	7	HB1	0	Horizontal Blank Byte 1
	8	ROFF0	0	Row Offset Byte 0
	9	ROFF1	0	Row Offset Byte 1
	10	RWS0	255	Row Window Size Byte 0
	11	RWS1	3	Row Window Size Byte 1. Bit15 = Vertical Reflection
	14	IT0	92	Integration Time Byte 0
	15	IT1	17	Integration Time Byte 1
	16	IT2	0	Integration Time Byte 2
	17	IT3	0	Integration Time Byte 3
	18	FT0	65	Frame Time Byte 0
	19	FT1	131	Frame Time Byte 1
	20	FT2	2	Frame Time Byte 2
	21	FT3	0	Frame Time Byte 3
	22	N/A	N/A	N/A
	26	VHI	235	Ramp High Bias
	28	VLO	85	Ramp Low Bias
	36	N/A	N/A	N/A
	37	N/A	N/A	N/A
	38	CONF0	22	Configuration Byte 0
	39	CONF1	164	Configuration Byte 1
	40	CONF2	228	Configuration Byte 2
	41	CONF3	13	Configuration Byte 3
	42	N/A	N/A	N/A
	43	N/A	N/A	N/A
	44	N/A	N/A	N/A
	46	NCP	0	Null Column Pairs Inserted Between Pairs of Columns
	47	N/A	N/A	N/A
	48	N/A	N/A	N/A
	49	CONF4	146	Configuration Byte 4
	50	N/A	N/A	N/A
	51	N/A	N/A	N/A
	52	N/A	N/A	N/A
	53	N/A	N/A	N/A

54	**N/A**	N/A	**N/A**
55	**N/A**	N/A	**N/A**
56	**N/A**	N/A	**N/A**

Shading indicates 16-bit registers (two 8-bit registers with shared updating)

7.4. Register Functions

The following register functions are the ones that are accessible by users. Some of the registers listed above are for testing and we will not explain their usage here. Please contact your representative if you believe they are needed.

7.4.1.Serial Command Interface Synchronization

The SCI may need to be resynchronized after a glitch or interruption in the serial communication. Address 0 is reserved for this purpose. There is no physical register at address 0. A read back of the registers should be performed after a glitch to make sure that no registers were corrupted while attempting to write to or read from a register during a glitch.

The following command sequence should be sent to the SCI to the resynchronize the serial command interface in Table 10:

Table 10: Serial Command sequence

IMM	W/R	ADDR[5:0]	DATA[7:0]	
1	1	00000	00000000	(Sync Word)
0	0	00000	00000000	(Null Word)
1	1	00000	00000000	(Sync Word)

*7.4.2.*Communication Register

The 8-bit Communication Register COM configures the serial communication. It is used to command a global reset of the FPA, configure the serial data output buffer, and control the updating of the registers.

Table 11: Communication Register COM

Addr: 1		**COM** Preload: 0
0	RESET	Global Reset Of Command Circuitry
1	~EN_OD	Enable Of Open Drain On Serial Word Readback
2	EOI_INH	Inhibit End Of Integration Update to Registers (not FT[3:0])
3	FS_INH	Inhibit Frame Start Update to Registers (FT[3:0] only)
4	X	
5	X	
6	X	
7	X	

The CMDRST signal allows the FPA to be reset without having to power cycle the FPA. When CMDRST is asserted high (set to 1), a global reset is generated for approximately 3 ms. This resets the timing circuits and also causes the registers to be reset to their preload values—clearing the CMDRST bit.

The active low ~EN_OD signal configures the serial data output SDOUT as an open drain output, requiring an external resistor pull-up. De-asserting ~EN_OD (set to 1) configures SDOUT with a CMOS buffer output.

The EOI_INH signal can be set to globally inhibit all of the registers (except the Frame Time registers) from updating at the End of Integration. The FS_INH signal can be set to globally inhibit the Frame Time registers from updating at Frame Start. This allows new values to be sent to several registers over an indefinite period of time without disturbing the present register values. In this case, no registers will be updated unless the IMM bit is set to 1 in the address byte. If the EOI_INH and FS_INH bits are cleared, all registers will then be updated at the next end of integration (or Frame Start for the Frame Time registers). Note: Writes to the COM register to clear the EOI_INH or FS_INH bits should always have the IMM bit set to 1.

The default command sequence for most applications:

1) Set EOI_NHI and/or FS_INH

2) Write to registers

3) Clear EOI_NHI and/or FS_INH with IMM set.

7.4.3.Column Offset Register

The 16-bit Column Offset Register COFF (two 8-bit registers COFF0 and COFF1 with shared updating) determines the first column to be read out. The resolution of the Column Offset Register is equal to one column. The preload value is COFF=8—the first “active” column. The first eight columns (COFF=0-7) may be used but are considered border columns (may have response rolloff due to processing effects at the borders).

Table 12: Column Offset Register COFF

Addr: 2	COFF0	Preload: 8
0	COFF0	Column Offset Bit 0
1	COFF1	Column Offset Bit 1
2	COFF2	Column Offset Bit 2
3	COFF3	Column Offset Bit 3
4	COFF4	Column Offset Bit 4
5	COFF5	Column Offset Bit 5
6	COFF6	Column Offset Bit 6
7	COFF7	Column Offset Bit 7

Addr: 3	COFF1	Preload: 0
0	COFF8	Column Offset Bit 8
1	COFF9	Column Offset Bit 9
2	COFF10	Column Offset Bit 10
3	COFF11	Column Offset Bit 11
4	X	
5	X	
6	X	
7	X	

7.4.4. Column Window Size Register

The 16-bit Column Window Size Register CWS (two 8-bit registers CWS0 and CWS1 with shared updating) determines the number of columns to be read out in addition to the first column (offset column). Bit 15 is the sign bit (horizontal reflection). The resolution of the Column Window Size Register is equal to one column. The actual number of columns read out will be a quad number of columns (4 pixels per REFCLK cycle). Thus, for column window sizes divisible by four, there will be one extra column read out. For column window sizes with a remainder of one when divided by four, there will be two extra columns read out. For column window sizes with a remainder of two when divided by four, there will be three extra columns read out. For column window sizes with a remainder of three when divided by four, there will be no extra columns read out. If the column window size extends beyond the physical array, then the missing columns in the last quad column will be filled with the ODB control word indicating the End Of Data. The rest of the column window will then be filled with horizontal blanks (HB control word).

Note: If the Horizontal Reflection Bit is set, then the start column from the COFF registers must be set to 1296 minus the no horizontal reflection COFF settings.

Table 13: Column Window Size Register CWS

Addr: 4	**CWS0**	Preload: 255
0	CWS0	Column Window Size Bit 0
1	CWS1	Column Window Size Bit 1
2	CWS2	Column Window Size Bit 2
3	CWS3	Column Window Size Bit 3
4	CWS4	Column Window Size Bit 4
5	CWS5	Column Window Size Bit 5
6	CWS6	Column Window Size Bit 6
7	CWS7	Column Window Size Bit 7

Addr: 5	**CWS1**	Preload: 4
0	CWS8	Column Window Size Bit 8
1	CWS9	Column Window Size Bit 9
2	CWS10	Column Window Size Bit 10
3	CWS11	Column Window Size Bit 11
4	CWS12	Column Window Size Bit 12
5	CWS13	Column Window Size Bit 13
6	X	
7	HREFL	Horizontal Reflection. (Sign Bit)

7.4.5.Horizontal Blank Register

The 16-bit Horizontal Blank register HB (two 8-bit registers HB0 and HB1 with shared updating) determines the number of blank quad columns (4 pixels per REFCLK cycle) to add at the end of each line read out.

Horizontal blanks will also be added automatically at the end of super lines (groups of 4 lines) that are shorter than an A/D conversion cycle (converting the next 4 lines). A minimum size super line is equivalent to 2448 pixels (612 REFCLK cycles) in 14-bit A/D mode or 1268 pixels (317 REFCLK cycles) in 13-bit A/D mode.

Table 14, Horizontal Blank Register HB

Addr: 6	**HB0**	Preload: 0
0	HB0	Horizontal Blank Bit 0 (Blank octuple columns at end of line)
1	HB1	Horizontal Blank Bit 1
2	HB2	Horizontal Blank Bit 2
3	HB3	Horizontal Blank Bit 3
4	HB4	Horizontal Blank Bit 4
5	HB5	Horizontal Blank Bit 5
6	HB6	Horizontal Blank Bit 6
7	HB7	Horizontal Blank Bit 7

Addr: 7	**HB1**	Preload: 0
0	HB8	Horizontal Blank Bit 8
1	HB9	Horizontal Blank Bit 9
2	HB10	Horizontal Blank Bit 10
3	HB11	Horizontal Blank Bit 11
4	HB12	Horizontal Blank Bit 12
5	HB13	Horizontal Blank Bit 13
6	HB14	Horizontal Blank Bit 14
7	X	

7.4.5.1. Row Offset Register

The 16-bit Row Offset Register ROFF (two 8-bit registers ROFF0 and ROFF1 with shared updating) determines the first row to be read out. The resolution of the Row Offset Register is equal to one row. The preload value is ROFF=0, but the first “active” row is ROFF=8. Despite differing from the preload, this should be considered the nominal operation mode. The first eight rows (ROFF=0-7) may be used but are considered border rows (may have response rolloff due to processing effects at the borders).

TABLE 15: Row Offset Register ROFF

Addr: 8	ROFF0		Preload: 0
0	ROFF0	Row Offset Bit 0	
1	ROFF1	Row Offset Bit 1	
2	ROFF2	Row Offset Bit 2	
3	ROFF3	Row Offset Bit 3	
4	ROFF4	Row Offset Bit 4	
5	ROFF5	Row Offset Bit 5	
6	ROFF6	Row Offset Bit 6	
7	ROFF7	Row Offset Bit 7	
Addr: 9	**ROFF1**		**Preload: 0**
0	ROFF8	Row Offset Bit 8	
1	ROFF9	Row Offset Bit 9	
2	ROFF10	Row Offset Bit 10	
3	X		
4	X		
5	X		
6	X		
7	X		

7.4.6. Row Window Size Register

The 16-bit Row Window Size Register RWS (two 8-bit registers RWS0 and RWS1 with shared updating) determines the number of rows to be read out in addition to the first row (offset row). Bit 15 is the sign bit (vertical reflection). The resolution of the Row Window Size Register is equal to one row. The test row (row address 1296) is sent out in the output data stream for the remaining portion of row windows that extend beyond the actual detector array. At the end of data readout, vertical blanks (VB control word) will automatically be added to the output data stream to make up the rest of the frame.

Note: If the Vertical Reflection Bit is set, then the start row from the ROFF registers must be set to 1024 minus the no vertical reflection ROFF settings.

Table 16, Row Window Size Register RWS

Addr: 10	RWS0	Preload: 255
0	RWS0	Row Window Size Bit 0
1	RWS1	Row Window Size Bit 1
2	RWS2	Row Window Size Bit 2
3	RWS3	Row Window Size Bit 3
4	RWS4	Row Window Size Bit 4
5	RWS5	Row Window Size Bit 5
6	RWS6	Row Window Size Bit 6
7	RWS7	Row Window Size Bit 7

Addr: 11	RWS1	Preload: 3
0	RWS8	Row Window Size Bit 8
1	RWS9	Row Window Size Bit 9
2	RWS10	Row Window Size Bit 10
3	X	
4	X	
5	X	
6	X	
7	VREFL	Vertical Reflection. (Sign Bit)

7.4.7.Row and Column Addressing

The row and column addressing of the 1280 x 1024 detector array is shown in Figure 15 below. Window size and position can be specified in 1 x 1 pixel increments. The window can be specified as small as 1 col x 1 row, but due to internal timing constraints the minimum uniform-size output window is 612 cols x 8 rows (317 cols x 8 rows in the 13-bit mode)—the window will be automatically filled with the appropriate number of horizontal blank and vertical blank pixels.

The “useable” 1280 x 1024 array has row addresses 8 to 1287 and column addresses 8 to 1031. The rows are read out in the slow multiplexed direction. The columns are read out in the fast multiplexed direction. Thus, the address of a given detector is shown in the Figure 15 as (row, column).

There are 8 extra columns/rows surrounding the “useable” area (for process uniformity). The ODB control word will finish out a quad column (2 pixels per REFCLK cycle per output port) indicating the end of valid data when the column window extends beyond the physical array. The line number provided with each row that is read out can be used to indicate valid rows.

Horizontal blanks will fill out the rest of the window for windows extending beyond column address 0 or column address 1296. The test row (at row address 1296 with selectable on-chip or off-chip bias inputs) will fill out the rest of the window for windows extending beyond row address 0 or row address 1040. Column and/or row offsets (starting column/row) that are outside of the physical limits will output floating data bus values.

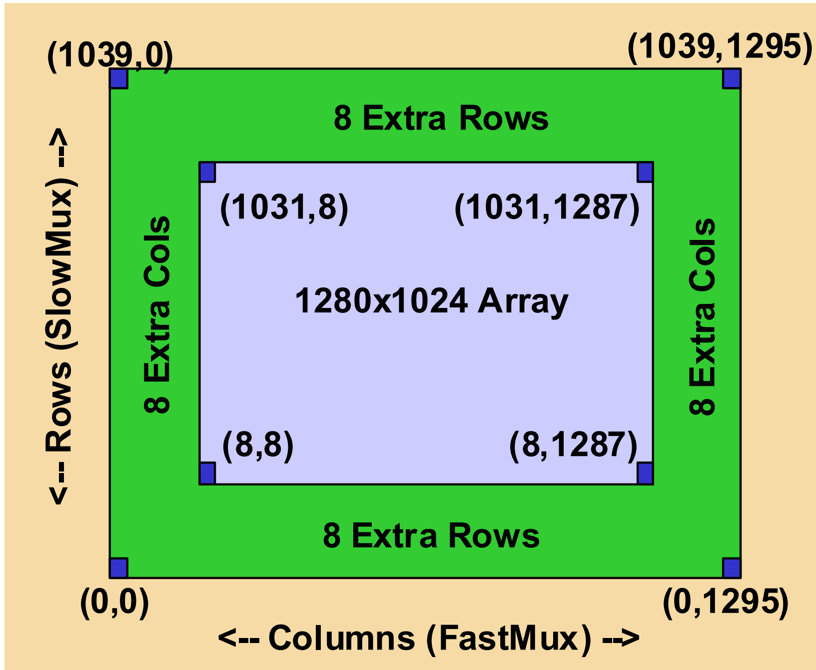


Figure 15 Row and Column Addressing scheme for FPA.

7.4.8. Integration Time Register

The 32-bit Integration Time register IT (two 16-bit registers without shared updating each composed of two 8-bit registers with shared updating, IT0 & IT1 and IT2 & IT3) determines the integration time in REFCLK cycles. The minimum integration time is internally limited to 12 REFCLK cycles. But to ensure proper unit cell operation the integration time should be set to greater than 512 REFCLK cycles

Table 17, Integration Time Register IT

Addr: 14	**IT0**	Preload: 92
0	IT0	Integration Time Bit 0 (Time in Master Clock Period)
1	IT1	Integration Time Bit 1
2	IT2	Integration Time Bit 2
3	IT3	Integration Time Bit 3
4	IT4	Integration Time Bit 4
5	IT5	Integration Time Bit 5
6	IT6	Integration Time Bit 6
7	IT7	Integration Time Bit 7

Addr: 15	**IT1**	Preload: 17
0	IT8	Integration Time Bit 8
1	IT9	Integration Time Bit 9
2	IT10	Integration Time Bit 10
3	IT11	Integration Time Bit 11
4	IT12	Integration Time Bit 12
5	IT13	Integration Time Bit 13
6	IT14	Integration Time Bit 14
7	IT15	Integration Time Bit 15

Addr: 16	**IT2**	Preload: 0
0	IT16	Integration Time Bit 16
1	IT17	Integration Time Bit 17
2	IT18	Integration Time Bit 18
3	IT19	Integration Time Bit 19
4	IT20	Integration Time Bit 20
5	IT21	Integration Time Bit 21
6	IT22	Integration Time Bit 22
7	IT23	Integration Time Bit 23

Addr: 17	**IT3**	Preload: 0
0	IT24	Integration Time Bit 24
1	IT25	Integration Time Bit 25
2	IT26	Integration Time Bit 26
3	IT27	Integration Time Bit 27
4	IT28	Integration Time Bit 28
5	IT29	Integration Time Bit 29
6	IT30	Integration Time Bit 30
7	IT31	Integration Time Bit 31

7.4.9.Frame Time Register

The 32-bit Frame Time register FT (two 16-bit registers without shared updating each composed of two 8-bit registers with shared updating, FT0 & FT1 and FT2 & FT3) determines the frame time in REFCLK cycles. The actual frame time will be long enough to ensure that integration will complete. The frame time should be set greater than eight lines; otherwise the output data frame start may not be generated. The minimum frame time is internally limited to 1224 REFCLK cycles in 14-bit mode and 640 REFCLK cycles in 13-bit mode.

Table 18, Frame Time Register FT

Addr: 18	FT0	Preload: 65
0	FT0	Frame Time Bit 0 (Time in Master Clock Period)
1	FT1	Frame Time Bit 1
2	FT2	Frame Time Bit 2
3	FT3	Frame Time Bit 3
4	FT4	Frame Time Bit 4
5	FT5	Frame Time Bit 5
6	FT6	Frame Time Bit 6
7	FT7	Frame Time Bit 7

Addr: 19	FT1	Preload: 131
0	FT8	Frame Time Bit 8
1	FT9	Frame Time Bit 9
2	FT10	Frame Time Bit 10
3	FT11	Frame Time Bit 11
4	FT12	Frame Time Bit 12
5	FT13	Frame Time Bit 13
6	FT14	Frame Time Bit 14
7	FT15	Frame Time Bit 15

Addr: 20	FT2	Preload: 2
0	FT16	Frame Time Bit 16
1	FT17	Frame Time Bit 17
2	FT18	Frame Time Bit 18
3	FT19	Frame Time Bit 19
4	FT20	Frame Time Bit 20
5	FT21	Frame Time Bit 21
6	FT22	Frame Time Bit 22
7	FT23	Frame Time Bit 23

Addr: 21	FT3	Preload: 0
0	FT24	Frame Time Bit 24
1	FT25	Frame Time Bit 25
2	FT26	Frame Time Bit 26
3	FT27	Frame Time Bit 27
4	FT28	Frame Time Bit 28
5	FT29	Frame Time Bit 29
6	FT30	Frame Time Bit 30
7	FT31	Frame Time Bit 31

7.4.10. Analog Ramp Adjustment Registers

The analog ramp waveform is adjusted using two on-chip programmable 8-bit DAC's. The Ramp Low Bias (VLO) sets the reset level of the ramp. The Ramp High bias (VHI) sets the upper end of the ramp.

The analog ramp is compared to the sampled integrated signal. Thus, the analog ramp waveform determines the mapping of the FPA dynamic range to the digital output count. Use VLO to adjust the lower end of the dynamic range (the offset level). Use VHI to adjust the upper end of the dynamic range (and thus the gain). VHI and VLO are programmable through two separate 8-bit registers VHI and VLO.

Table 19: Ramp Bias High Register VHI

Addr: 26	**VHI**	Preload: 235
0	VHI0	Ramp High Bias Bit 0
1	VHI1	Ramp High Bias Bit 1
2	VHI2	Ramp High Bias Bit 2
3	VHI3	Ramp High Bias Bit 3
4	VHI4	Ramp High Bias Bit 4
5	VHI5	Ramp High Bias Bit 5
6	VHI6	Ramp High Bias Bit 6
7	VHI7	Ramp High Bias Bit 7

Table 20: Ramp Bias Low Register VLO

Addr: 28	**VLO**	Preload: 85
0	VLO0	Ramp Low Bias Bit 0
1	VLO1	Ramp Low Bias Bit 1
2	VLO2	Ramp Low Bias Bit 2
3	VLO3	Ramp Low Bias Bit 3
4	VLO4	Ramp Low Bias Bit 4
5	VLO5	Ramp Low Bias Bit 5
6	VLO6	Ramp Low Bias Bit 6
7	VLO7	Ramp Low Bias Bit 7

The predicted ramp bias values (with VPOS = +3.3 V) are shown in Table 3.3.4.3.12C.

Table 21: Predicted Ramp Bias DAC Voltages

Count	(V) CDAC	Count	(V) CDAC	Count	(V) CDAC	Count	(V) CDAC	Count	(V) CDAC	Count	(V) CDAC	Count	(V) CDAC	Count	(V) CDAC
0	0.000	32	0.410	64	0.825	96	1.235	128	1.658	160	2.069	192	2.483	224	2.894
1	0.013	33	0.423	65	0.837	97	1.248	129	1.671	161	2.081	193	2.496	225	2.906
2	0.025	34	0.436	66	0.850	98	1.261	130	1.683	162	2.094	194	2.508	226	2.919
3	0.038	35	0.448	67	0.863	99	1.273	131	1.696	163	2.106	195	2.521	227	2.931
4	0.051	36	0.461	68	0.876	100	1.286	132	1.709	164	2.119	196	2.534	228	2.944
5	0.063	37	0.473	69	0.888	101	1.298	133	1.721	165	2.132	197	2.546	229	2.957
6	0.076	38	0.486	70	0.901	102	1.311	134	1.734	166	2.144	198	2.559	230	2.969
7	0.088	39	0.499	71	0.913	103	1.324	135	1.746	167	2.157	199	2.571	231	2.982
8	0.102	40	0.512	72	0.927	104	1.337	136	1.760	168	2.170	200	2.585	232	2.995
9	0.114	41	0.524	73	0.939	105	1.349	137	1.772	169	2.183	201	2.597	233	3.008
10	0.127	42	0.537	74	0.952	106	1.362	138	1.785	170	2.195	202	2.610	234	3.020
11	0.139	43	0.550	75	0.964	107	1.375	139	1.798	171	2.208	203	2.622	235	3.033
12	0.152	44	0.563	76	0.977	108	1.388	140	1.810	172	2.221	204	2.635	236	3.046
13	0.165	45	0.575	77	0.990	109	1.400	141	1.823	173	2.233	205	2.648	237	3.058
14	0.177	46	0.588	78	1.002	110	1.413	142	1.836	174	2.246	206	2.661	238	3.071
15	0.190	47	0.600	79	1.015	111	1.425	143	1.848	175	2.258	207	2.673	239	3.083
16	0.204	48	0.615	80	1.029	112	1.440	144	1.862	176	2.273	208	2.687	240	3.098
17	0.217	49	0.627	81	1.042	113	1.452	145	1.875	177	2.285	209	2.700	241	3.110
18	0.229	50	0.640	82	1.054	114	1.465	146	1.888	178	2.298	210	2.713	242	3.123
19	0.242	51	0.652	83	1.067	115	1.477	147	1.900	179	2.311	211	2.725	243	3.136
20	0.255	52	0.665	84	1.080	116	1.490	148	1.913	180	2.323	212	2.738	244	3.148
21	0.267	53	0.678	85	1.092	117	1.503	149	1.926	181	2.336	213	2.750	245	3.161
22	0.280	54	0.690	86	1.105	118	1.515	150	1.938	182	2.349	214	2.763	246	3.174
23	0.292	55	0.703	87	1.117	119	1.528	151	1.951	183	2.361	215	2.776	247	3.186
24	0.306	56	0.716	88	1.131	120	1.541	152	1.964	184	2.374	216	2.789	248	3.199
25	0.318	57	0.729	89	1.143	121	1.554	153	1.977	185	2.387	217	2.802	249	3.212
26	0.331	58	0.741	90	1.156	122	1.566	154	1.989	186	2.400	218	2.814	250	3.225
27	0.343	59	0.754	91	1.168	123	1.579	155	2.002	187	2.412	219	2.827	251	3.237
28	0.356	60	0.767	92	1.181	124	1.592	156	2.015	188	2.425	220	2.840	252	3.250
29	0.369	61	0.779	93	1.194	125	1.604	157	2.027	189	2.438	221	2.852	253	3.262
30	0.381	62	0.792	94	1.206	126	1.617	158	2.040	190	2.450	222	2.865	254	3.275
31	0.394	63	0.804	95	1.219	127	1.629	159	2.052	191	2.463	223	2.877	255	3.288

7.4.11. Configuration Register0

The 8-bit Configuration Register CONF0 allows some flexibility in the operation of the FPA.

Table 22, Configuration Register CONF0

Addr: 38	**CONF0**	Preload: 22
0	REDHI	Enable Rail To Rail or reduced Swing For Reset of Unit cell
1	A2D14	1 = 14-bit A/D, 0 = 13-bit A/D
2	EQBLK	HB and VB Set Equal to the BLK Control Word
3	ENIT	Enable Instrumentation Readout After Last Row
4	ENDBIT	Enable DBIT LVDS Output
5	FCRST	Reset Frame Counter
6	RMPROW	Select one of two ramp buffer rows
7	TSTPAT	Select one of two NSF test rows

- The REDHI signal allows the resetting of the Unit Cell clock rails to be adjusted from rail to rail to ground to a reduced high rail. If REDHI is set to logic zero and FSPRSTINT is set to logic one, the clock will swing rail to rail.
- The A2D14 signal allows the A/D's to be configured for 14 or 13 bits. Preload: A2D14 = 1 (14-bit A/D mode).
- The EQBLK signal allows the horizontal blank HB and vertical blank VB control words to be set equal to the blank BLK control word. Preload: EQBLK = 1 (HB = VB = BLK).
- The ENIT signal enables the instrumentation data to be read out after the last row is read out. Preload: ENIT = 0 (no instrumentation data at end of frame).
- The ENDBIT signal enables the LVDS output for the DBIT. This saves power if the DBIT LVDS is terminated with a 100 ohm load, but not used. Preload: ENDBIT = 1 (LVDS output for the DBIT enabled).
- The FCRST signal resets the 28-bit frame counters. Preload: FCRST = 0.
- The RMPROW signal selects one of two ramp buffer rows. Preload: RMPROW = 0.
- The TSTPAT signal selects one of two NSF test rows. See Test Mode register description upcoming for a detailed description of the test patterns. Preload: TSTPAT = 0.

7.4.12. Configuration Register1

The 8-bit Configuration Register CONF1 allows some flexibility in the operation of the FPA.

Table 23: Configuration Register CONF1

Addr: 39	**CONF1**	Preload: 164
0	JBR	Jam Before Reading out
1	JAD	Jam at edges of A/D cycles
2	SHW0	Sample/Hold pulse Width adjustment bit 0
3	SHW1	Sample/Hold pulse Width adjustment bit 1
4	SHEOR	Sample/Hold pulse starts just at End Of Row
5	FSPRSTINT	Full Swing PRSTINT clock rails
6	OM0	Output Mode Bit 0
7	OM1	Output Mode Bit 1

The JBR signal (active high signal) allows the user to JAMSYNC the chip to control when the readout cycle begins instead of the beginning of integration. The JAMSYNC signal initiates the beginning of the readout sequence, but there is a 8 line latency due to internal pipelining. If JBR is set to logic zero, then the chip is in Jam then Integrate mode. Preload: JBR = 0 (jam then Integrate).

The JAD signal (active low signal) allows the integration to start within 3 to 4 REFCLK cycles of the JAMSYNC signal (low jitter JAMSYNC mode) else to delay the start of integration to the edges of the A/D conversion cycle (high jitter JAMSYNC mode). Preload: JAD = 0 (low jitter JAMSYNC mode, normal mode). ***This circuit however is for internal test purposes only and the user must refrain from using this feature.***

- SHW [1:0] control the width of the Sample/Hold clock. 00 is 256 master clock cycles, 01 is 512 cycles, 10 is 1024 cycles and 11 is 2048 cycles.
- SHEOR enables the Sample/Hold clock to go high at the end of the readout cycle. Active high.
- FSPRSTINT commands the Prstint clock to be rail to rail. If set to logic zero, the high rail is reduced by internal voltage diveders. This function works in concert with REDHI bit from CONF0 register.

The Output mode control bits OM[1:0], control the number of outputs enabled. In four output mode OM1 is set to logic 1, and OM0 is set to a logic0. In two output mode OM1 is set to logic 0, and OM0 is set to a logic 1. In one output mode both OM1 and OM0 are set equal to each other.

7.4.13. Configuration Register2

The 8-bit Configuration Register CONF2 allows output numbering to be configured.

Table 24: Configuration Register CONF2

Addr: 40	**CONF2**	Preload: 228
0	ON0	Output Number Bit 0 for output 0
1	ON1	Output Number Bit 1 for output 0
2	ON2	Output Number Bit 0 for output 1
3	ON3	Output Number Bit 1 for output 1
4	ON4	Output Number Bit 0 for output 2
5	ON5	Output Number Bit 1 for output 2
6	ON6	Output Number Bit 0 for output 3
7	ON7	Output Number Bit 1 for output 3

The Output numbering register allows each output to be reconfigured to be any one of the four outputs. Each output has two bits for this function. If you set these two bits to 00, then this output will be output 1. If you set them to 01, then this output will be output . . If you set them to 10, then this output will be output 3. . If you set them to 11, then this output will be output 4. This option is basically for mother board layout in one or two output mode if the layout is proving difficult. The default is 4 output, straight documentation.

7.4.14. Configuration Register3

The 8-bit Configuration Register CONF3 allows some flexibility in the operation of the FPA.

Table 25: Configuration Register CONF3

Addr: 41		CONF3	Preload: 13
0	~ENFR	Enable Ramp Filter (Active Low)	
1	X		
2	~ENRBF	Enable Ramp Bias Filters (Active Low)	
3	~ENRBUPF	Enable Ramp Bias Update Per Frame (Else Per Ramp)	
4	ENCVPSF	Enable Clocking of VPSF	
5	X		
6	X		
7	X		

- The ~ENRF signal (active low signal) enables the switch capacitor filtering of the bias to the gate of the current source that generates the ramp. Preload: ~ENRF = 0 (ramp filter enabled).
- The ~ENRBF signal (active low signal) enables the switch capacitor filtering of the ramp biases (VHI and VLO). Preload: ~ENRBF = 0 (ramp bias filter enabled).
- The ~ENRBUPF signal (active low signal) enables the updating of the ramp biases once per frame (otherwise once per A/D cycle). Preload: ~ENRBUPF = 0 (update once per frame).
- The ENCVPSF signal allows the source follower in each unit cell to effectively be shut down during the transfer of signal from the Integration capacitor to the sample/hold capacitor. This is done to reduce some of the parasitic gate capacitance of the source follower. The default is logic 0, not clocked.

7.4.15. Unit Cell Configuration Registers

There are 8 registers that control the operation parameters of the unit cell. These registers are typically not adjusted by most users due to the need of detailed information of the unit cell design as well as the circuits that these registers control.

TABLE 26: Configuration 4 Registers, CONF4

Addr: 49		CONF4	Preload: 146
0	UCIEN0	Unit Cell Current Adjust Bit 0	
1	UCIEN1	Unit Cell Current Adjust Bit 1	
2	UCIEN2	Unit Cell Current Adjust Bit 2	
3	DTIEN0	Detector Bias Current Adjust Bit 0	
4	DTIEN1	Detector Bias Current Adjust Bit 1	
5	DTIEN2	Detector Bias Current Adjust Bit 2	
6	CPT	UC Power Clock Trigger (0 = FS, 1 = EOI)	
7	UCCLKEN	Enable Unit Cell Clock (Active Low)	

The Unit Cell Current Adjust bits are used control the current sources in the CTIA unit cell amplifiers. Nominally the CTIA was designed to have a 30nA per amplifier unit cell current, but this can be reduced to 15nA to reduce the overall chip power dissipation. Larger unit cell currents may be used, but this will greatly increase the chip's power dissipation. The higher current may be needed for fast integration times or fast events where the amplifier speed needs to be increased.

Table 27: CTIA Current on Amplifiers

UCCLKEN	UCIEN2	UCIEN1	UCIEN0	CTIA Current	Mode
X	0	0	0	120nA	DC
X	0	0	1	60nA	DC
X	0	1	0	30nA	DC
X	0	1	1	15nA	DC

The unit cell AC clocking features are controlled by the rest of the registers. ***We do not recommend touching any of these other features as they could cause damage to the circuit and/or hurt the performance of the array.***