Commands and Replies Application Note
Firmware v1.3

Thursday, January 6, 2022

The GenIV controller uses a command-reply system, where both a command and a reply (status) are 32-bit values typically represented by ASCII letters up to four characters long. For example, the test data link command is represented by the ASCII letters 'TDL', which is `0x0054444C` in hexadecimal. All commands must return a reply (status). The standard return status is the ASCII value 'DONE' (`0x444F4E45`); indicating that a command was successfully received and processed by the controller. On error, the standard return status is 'EROR' (`0x45524F52`) and is typically augmented with an error code indicating the nature of the error. The augmented error codes can be found in the common *ArcErrorDefs.h* header and represent typical errors, such as invalid parameter or count, invalid command, timeout, range error, invalid board number, etc. The *ArcCommandDefs.h* header is another common file to both the micro-controller and the ARC API C++ library code and contains the standard command and status definitions.

The controller command processor is divided into two sub-processors; the primary base command processor and the local (user defined) command processor. The latter is optional and is defined as a stub by default. The primary command processor will automatically handle any commands that are defined within the *ArcCommandDefs.h* header file. If the processor is unable to find the incoming command, it will pass the command to the user-defined local command processor (see figure 1).

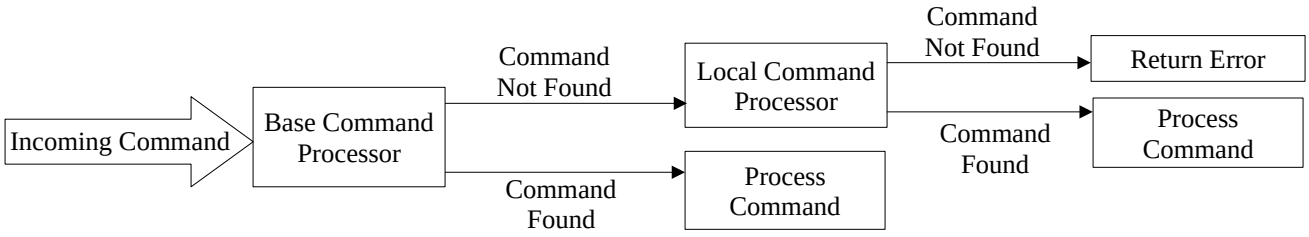


Figure 1: Micro-controller command processing

An error will be returned by the controller if both the primary and the local user-defined command processor fail to recognize any received command. The local command processor has the following definition:

```
bool localCommandHandler( ArcCommandPacket_t* pCmdPacket );
```

The function must return false if the command cannot be processed (invalid command); otherwise it must return true. The return of true or false from the local command processor only determines if the command definition is valid and says nothing as to whether or not the command was successfully

handled. The success of a command is handled directly by the command function itself. The definition of the *ArcCommandPacket_t* type can be found in the *ArcPacket.h* header of the *ArcGen4Base* library. It contains pointers to the header, command and arguments.

The list of commands accepted by the controller can be read by a host computer application. The G4 application will automatically read these values and display them in its commanding windows. Commands exist for reading both the primary and user-defined local command definitions and values. G4 will automatically call for all command definitions and provide a complete list of primary and user-defined values in it's commanding window.

In the micro-controller all local (user-defined) commands should be defined using the *cmd_t* structure found in the *ArcCommands.h* header of the *ArcGen4Base* library. While the use of the *cmd_t* structure is not explicitly required, it will guarantee smooth integration with the *ArcGen4Base* library.

GenIV Alerts

The controller has the ability to send alert messages to the host computer application if a potential health issue arises. An alert message is sent using the following command, which is defined in the *ArcGen4Base* library:

```
void sendAlert( const char* message );
```

The host app must register an alert handler with the *CArcDevice* class within the ARC C++ API.

Base Library Command Definitions

The following details the GenIV base library controller commands.

Test Data Link

Send a test value from the host computer to the controller. The controller should return the same value which the host computer will compare to be equal. Verifies the computer-controller communications link.

Command: 0x0054444C ('TDL')
Argument(s): Any 32-bit value
Return argument: The 32-bit command argument

Controller Ready

Returns true (1) only if the timing board micro-controller has successfully booted.

Command: 0x43524459 ('CRDY')
Argument(s): none
Return argument: 1 (true) if micro-controller has successfully booted

Get Command Count

The timing board micro-controller contains a list of available commands that is readable by the host computer. This command returns the total number of commands defined in the micro-controller. This number can be used to repeatedly call the *Get Command At* and *Get Command Value* functions to obtain command details, such as its description and value. Two versions are available, one for the standard base command list and one for user (local) defined commands.

Description: Get the command list for the base set of commands
Command: 0x00474343 ('GCC')
Argument(s): none for the standard set of base commands
Return argument: The total number of commands supported by the controller.

Description: Get the command list for the user (local) set of commands
Command: 0x00474343 ('GCC')
Argument(s): 0x4C ('L') for the number of local (user) defined commands.
Return argument: The total number of commands supported by the controller.

Get Command At

The timing board micro-controller contains a list of available commands that is readable by the host computer. This command returns the text description for the micro-controller command defined at the specified list index. The total number of commands in the list can be obtained by calling the *Get Command Count* function. Two versions are available, one for the standard base command list and one for user (local) defined commands.

Description: Get the standard base command description
Command: 0x00474341 ('GCA')
Argument(s): The index of the command whose value should be read
Return argument: The command textual description. The length of the text will vary and non-ascii printable characters represent the end of the text.

Description: Get the user (local) command description
Command: 0x00474341 ('GCA')
Argument 0: The index of the command whose value should be read
Argument 1: 0x4C ('L') for the number of local (user) defined commands.
Return argument: The command textual description. The length of the text will vary and non-ascii printable characters represent the end of the text.

Get Command Value At

The timing board micro-controller contains a list of available commands that is readable by the host computer. This command returns the numerical value of the micro-controller defined command at the specified list index. The total number of commands in the list can be obtained by calling the *Get Command Count* function. Two versions are available, one for the standard base command list and one for user (local) defined commands.

Description: Get the standard base command value
Command: 0x47435641 ('GCVA')
Argument(s): The index of the command whose value should be read
Return argument: The 32-bit command numerical value

Description: Get the user (local) command value
Command: 0x47435641 ('GCVA')
Argument 0: The index of the command whose value should be read
Argument 1: 0x4C ('L') for the number of local (user) defined commands.
Return argument: The 32-bit command numerical value

Get System State

Returns the current state of the controller.

Command: 0x475353 ('GSS')
Argument(s): none
Return argument 0: The image column pixel dimension (32-bits)
Return argument 1: The image row pixel dimension (32-bits)
Return argument 2: The image channel count (16-bits)
Return argument 3: The maximum valid channel count (16-bits)
Return argument 4: The number of image columns-per-channel in pixels (16-bits)
Return argument 5: The number of resets to be performed (32-bits)
Return argument 6: The number of pre-expose read counts (N1) (32-bits)
Return argument 7: The number of post-expose read counts (N2) (32-bits)
Return argument 8: The exposure mode (32-bits)
Return argument 9: The number of signal averages (32-bits)
Return argument 10: The synthetic image mode (32-bits)
Return argument 11: 1 (true) if abort has been set on the controller; 0 otherwise (32-bits)
Return argument 12: 1 (true) if idle mode is currently executing; 0 otherwise (32-bits)
Return argument 13: 1 (true) if resets are currently executing; 0 otherwise (32-bits)

NOTE: All arguments are returned as 32-bits regardless of actual size

Get or Set Image Dimensions

Set or get the current image dimensions. Note that the column dimension is dependent on the number of columns-per-channel. Refer to the user's manual for details.

Description: Read function
Command: 0x0044494D ('DIM')
Argument(s): none for reading the current dimensions
Return argument 0: The 32-bit image column dimension in pixels
Return argument 1: The 32-bit image row dimension in pixels
Return argument 2: The 32-bit image columns-per-channel in pixels

Description: Write function
Command: 0x0044494D ('DIM')
Argument 0: The column dimension, in pixels, to write
Argument 1: The row dimension, in pixels, to write
Return argument: 'DONE' on success or 'EROR'

Get Board Map

Get the current controller board mapping. The mapping gives the type and position of all circuit boards in the controller.

Command: 0x47424D50 ('GBMP')
Argument(s): none
Return argument 0: The board at slot position 14 (0xE00000000 | board id)
Return argument 1: The board at slot position 13 (0xD00000000 | board id)
. . .
Return argument 6: The ARC-420 timing board position (0x800000420)
Return argument 7: The board at slot position 1 (0x100000000 | board id)
Return argument 8: The board at slot position 2 (0x200000000 | board id)
Return argument 14: The ARC-480 power-control board position (0xF00000480)
. . .

Each 32-bit return argument contains the board number (as a hexadecimal value) in bits 0-27 and the slot position in bits 28-31. For example, the timing board (ARC-420) position is fixed at slot 8, so the 32-bit return value will be 0x800000420.

Get Address Map

Get the current controller address mapping. The mapping is different from the board mapping function and gives the slot position, SPI, and temperature register addresses of all circuit boards in the controller.

Command: 0x47414D50 ('GAMP')
Argument(s): none
Every three 32-bit return arguments are grouped into the data for one controller board. e.g. return arguments 0, 1, and 2 represent the data for the first controller board. The returned board order is identical to that returned by the Get Board Map function.

Return argument 0: The board slot position
Return argument 1: The board SPI register address
Return argument 2: The board temperature register address
Return argument 3: The next board slot position
Return argument 4: The next board SPI register address
Return argument 5: The next board temperature register address
. . .

Start Exposure

Start an image exposure and readout.

Command: 0x00534558 ('SEX')
Argument(s): The exposure time in tens of microseconds
Return argument: 'DONE' on success; 'EROR' otherwise

Stop Exposure

Stops an image exposure and readout.

Command: 0x53544F50 ('STOP')
Argument(s): none
Return argument: 'DONE' on success; 'EROR' otherwise

Read or Enable Exposure Mode(s)

Read one of the available exposure mode text labels, the number of modes available, or enable one of the available exposure modes. If no argument is given, the size of the exposure mode list is returned. Otherwise, if an ASCII 'r' or 'R' plus an index argument is given, the expose mode definition will be returned. Likewise, if an ASCII 'e' or 'E' plus and index argument is given, the specified exposure mode will be enabled.

Description: Read expose mode count
Command: 0x5245584D ('REXM')
Argument(s): none
Return argument: The number of expose modes available on the controller.

Description: Read expose mode description at a specified index
Command: 0x5245584D ('REXM')
Argument(s): 'r' or 'R' followed by a numerical index
Return argument: The expose mode textual description for the specified list index. 'EROR' is returned if the index exceeds the list size, which can be obtained by sending 'REXM' with no arguments.

Description: Enable mode
Command: 0x5245584D ('REXM')
Argument(s): 'e' or 'E' followed by a numerical index
Return argument: 'DONE' on success; 'EROR' otherwise

Get Pixel Count

Returns the pixel count for the current image readout.

Command: 0x47505843 ('GPXC')
Argument(s): none
Return argument 0: The upper 32-bits of the current 64-bit pixel count (typically not used)
Return argument 1: The lower 32-bits of the current 64-bit pixel count

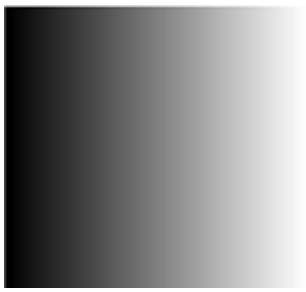
Read Elapsed Time

Returns the elapsed time, in tens of microseconds, for the current expose timer.

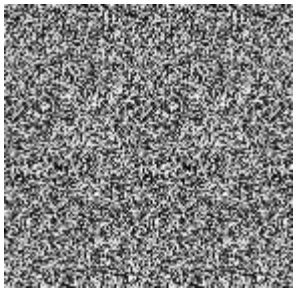
Command: 0x00524554 ('RET')
Argument(s): none
Return argument: The current elapsed exposure time in tens of microseconds

Enable or Disable Synthetic Image

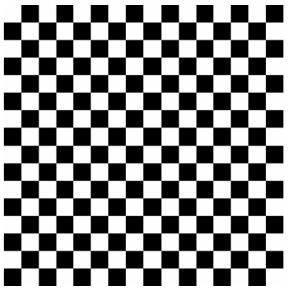
Enable or disable a synthetic image mode. The modes are defined in the ArcSyntheticDefs.h header file. There are four modes: none, timing board (ARC-420) generated, and two video board (ARC-440) modes.



Video Board Synthetic Image (gradient)



Video Board Synthetic Image (ADC)



Timing Board Synthetic Image

Command: 0x0053594E ('SYN')
Argument 0: One of the synImgSource_e mode values from the ArcSyntheticDefs.h file
Argument 1: 0 to disable; 1 to enable
Return argument: 'DONE' on success; 'EROR' otherwise

Read ARC-440 Video Board Register

Read an ARC-440 video board transceiver register. This is primarily for debug purposes. The transceiver commands are defined as follows:

cmd	*description*	*cmd arguments*
0x0	Read back total row number	
0x1	Enabled channel(s) on the video board	
0x2	Assigned channel(s) Logic Number	bits 19-16 physical channel number
0x3	Column number for each channel	
0x4	Image readout Status	
0x5	Video FIFO Status	
0x6	Pixel average number	
0x7	Used channel(s) Logic Number	
0x8	Synthetic image status	
0x9	Channel DC offset value	bits 19-16 channel number
0xA	NXP reply to video board link test (TDL)	
0xB	Pixel count sent by local board	
0xC	Pixel count sent to upper board	
0xD	Pixel count sent to lower board	
0xE	Pixel count received from upper board	
0xF	Pixel count received from lower board	

Command: 0x00525652 ('RVR')
Argument 0: The slot number of the video board (ARC-440) to access
Argument 1: The transceiver command
Argument 2: An optional command argument (command dependent)
Return argument: The value read from the specified video board transceiver.

Write Video Board (ARC-440) DAC

Write to the video board DAC.

Command: 0x00575644 ('WVD')
Argument 0: The slot number of the video board (ARC-440) to access
Argument 1: The 16-bit value to write
Return argument: 'DONE' on success; 'EROR' otherwise

Assign Video Board (ARC-440) Virtual Channels

This command has three modes. The first one, which takes no arguments, will tell the controller to auto assign the default channel numbers. The next write mode will assign virtual video channel numbers to specific physical video channel numbers. Each ARC-440 video board contains sixteen physical video channels that can be re-assigned using virtual channels. These channels can also be enabled and disabled using the write command. The final mode is a read mode that will return all the current channel assignments for a specific video board (ARC-440).

Description: Auto assign default virtual channel numbers
Command: 0x00415643 ('AVC')
Argument: none
Return argument: 'DONE' on success; 'EROR' otherwise

Description: Read the assigned virtual channel numbers
Command: 0x00415643 ('AVC')
Argument: The slot number of the video board (ARC-440) to access
Return argument: Returns the list of virtual channels and their assigned values. Any physical video channel that is not enabled will return a virtual video channel value of 0x99.

Description: Write/assign virtual channel numbers
Command: 0x00415643 ('AVC')
Argument 0: The slot number of the video board (ARC-440) to access
Argument 1: The enable bits. Each bit in this 16-bit value represents one of the board's 16 channels. To enable channel 0, set bit 0 to 1, etc. A set bit enables the channel, a cleared bit disables the channel.
Argument 2: The 32-bit virtual to physical channel binding. The upper 16-bits must contain the physical channel number, while the lower 16-bits must contain the virtual channel number. e.g. to bind virtual channel 3 to physical channel 6 pass argument 0x00060003.
…
Argument 17: The final channel binding
Return argument: 'DONE' on success; 'EROR' otherwise

Enable(d) Video Board (ARC-440) Channels

This command can be used to read or write the enabled video channels on the ARC-440.

Description: Read function
Command: 0x00455643 ('EVC')
Argument: The board slot number

Return argument: A 16-bit value containing the enabled channel bits. Bit 0 represents channel 0, bit 1 represents channel 1, etc. If a bit is set, then the associated channel is enabled. There are 16-bit because each video board contains 16 video channels.

Description: Write function
Command: 0x00455643 ('EVC')
Argument 0: The board slot number
Argument 1: The 16-bit value representing the channels to be enabled or disabled. Bit 0 represents channel 0, bit 1 represents channel 1, etc. If a bit is set, then the associated channel is enabled. There are 16-bit because each video board contains 16 video channels.

ADC Mode Control

This command sets the mode ADC mode bits. An example of this would be enabling the generation of the ARC-440 video board random synthetic image data.

Command: 0x00414D43 ('AMC')
Argument 0: The board slot number
Argument 1: The 4-bit value

Return argument: 'DONE' on success; 'EROR' otherwise

Set Signal Averaging

This command sets the signal averaging value.

Command: 0x00535341 ('SSA')
Argument: 1 - 255

Return argument: 'DONE' on success; 'EROR' otherwise

Get Video Pixel Count

Returns the transferred pixel count for a specific ARC-440 video board. This is a debug command that returns the number of pixels sent and received by the specified video board.

Command: 0x47565043 ('GVPC')
Argument: The board slot number
Return argument 0: the pixel count sent to the video board immediately above
Return argument 1: the pixel count sent to the video board immediately below
Return argument 2: the pixel count received from the video board immediately above
Return argument 3: the pixel count received from the video board immediately below

Board Revision Info

Return the board and FPGA revision values. The revision numbers are returned as ASCII characters. e.g. a board revision number of 0x1E is returned as ' 1E', which is represented as 0x00003145.

Command: 0x00425249 ('BRI')
Argument: The board slot number

Return argument 0: The board revision number (as ASCII)
Return argument 1: The FPGA revision number (as ASCII)

Read, Enable or Disable Board LEDs

This command can read, enable or disable a boards LEDs. The read command actually reads the LED state for all the boards in the controller. Each returned element has the form board number in hex in the upper 16-bits and 0 or 1 in the lower 16-bits for disabled or enabled. A return value of 0 indicates no board or value. e.g. if board 9 has its LEDs enabled, then the LED command will return 0x00090001 as one of its return values. If it's disabled the return value will be 0x00090000.

Command: 0x4C454453 ('LEDS')
Argument: none
Return argument 0: The LED state of the top-most board in the controller stack.
Return argument 1: The LED state of the next top-most board in the controller stack.
. . .
Return argument N: The LED state of the bottom-most board in the controller stack.

Note: In all the returned arguments, 1 = LED enabled, 0 = LED disabled

Enable or Disable Board Temperature Loop

Enable or disable the temperature read loop on the timing board. When enabled, the board temperature, in Celcius, will be continuously read and updated to a FPGA register.

Command: 0x4542544C ('EBTL')
Argument: 1 to enable, 0 to disable

Return argument: 'DONE' on success; 'EROR' otherwise

Read Board Temperature

Read a board temperature in degrees Celcius. The return value is a float the must be converted back to a float.

Command: 0x52444254 ('RDBT')
Argument: The board slot number

Return argument: The 32-bit board temperature (in Celcius). This value must be converted back to a floating point value.

Read Detector Temperature

Read a detector channel temperature in Celcius. To define channels, the user firmware app must override the LTC2983_channelConfig[] array and LTC2983_channelLength values from the ArcLTC2983.h header file. The return value is a float that must be converted back to a float.

Command: 0x00524454 ('RDT')
Argument: The channel number. Current (default) values are:
Channel 2 – Sense resistor (currently not used)
Channel 4 – RTD PT-100
Channel 7 – RTD PT-100
Channel 10 – RTD PT-100

Return argument: The 32-bit detector temperature (in Celcius). This value must be converted back to a floating point value.

Read Detector Temperature Configuration

Read the detector temperature configuration. Each pair of returned values consists of a text label followed by the channel number of the configuration. Each text label consumes four 32-bit return values and all configuration values are sent back in a single list packet.

Command: 0x52445443 ('RDTC')
Argument: none

Return arguments 0-3: The first configuration text label.
Return argument 4: Channel number for the first configuration.
Return arguments 5-8: The next configuration text label.
Return argument 9: Channel number for the next configuration.
. . . etc.

Enable or Disable Power Supply Current Loop

Enable or disable the voltage and current monitor loop on the power supply board. When enabled, the power board FPGA will continuously monitor and report voltage, current and status values to a set of registers that can be monitored.

Command: 0x4550434C ('EPCL')
Argument: 1 to enable, 0 to disable

Return argument: 'DONE' on success; 'EROR' otherwise

Read Power Supply Currents

Returns the ARC-480 power supply voltage current values.

The power supply values begin at the following status packet payload indices:

index 0 => 3.3 volts
index 4 => 2.5 volts
index 8 => 1.8 volts
index 12 => 1.2 volts timing board
index 16 => 1.2 volts video boards
index 20 => 5.5 volts
index 24 => - 1.8 volts

Each returned current consists of four values in the following order:

N + index 0 => the voltage status (1 = okay, 0 = error)
N + index 1 => the current out-of-range status (0 = okay, 1 = out-or-range)
N + index 2 => a read error occurred (0 = okay, 1 = read error)
N + index 3 => the current value

The final value returned in the packet contains the all values are okay status (1 = all values are good, 0 = not all values are good)

Command: 0x52505343 ('RPSC')
Argument: none
Return argument 0: The 3.3V status (1 = okay, 0 = error)
Return argument 1: The 3.3V current out-of-range status (0 = okay, 1 = out-or-range)
Return argument 2: The 3.3V read error check (0 = okay, 1 = read error)
Return argument 3: The 3.3V current value
Return arguments 4-7: 2.5V current value data
Return arguments 8-11: 1.8V current value data
Return arguments 12-15: 1.2V current value data (ARC-420 timing board)
Return arguments 16-19: 1.2V current value data (ARC-440 video boards)
Return arguments 20-23: 5.5V current value data
Return arguments 24-28: 1.8V current value data
Return argument 29: 1 = all current values are within expected range; 0 otherwise

Read or Write Flash Memory

Read or write a byte of flash memory data. The user may only write to the firmware address 0 to 0x2FFFF and the user area located at address 0x38000 to 0x3FFFF. The user is free to write any text they wish to the user area.

Description: Read function
Command: 0x5257464D ('RWFM')
Argument 0: The board slot number
Argument 1: A valid flash memory address
Return argument: The 8-bit data value

Description: Write function
Command: 0x5257464D ('RWFM')
Argument 0: The board slot number
Argument 1: A valid flash memory address
Argument 2: The 8-bit data value to write
Return argument: 'DONE' on success; 'EROR' otherwise

Erase Flash Memory

Erase flash memory data block. WARNING: data in an erased memory block is irretrievable.

Command: 0x0045464D ('EFM')
Argument 0: The board slot number
Argument 1: A valid flash memory block id. See the ArcFlashMemoryDefs.h file for block ids.
Return argument: 'DONE' on success; 'EROR' otherwise

Read or Write a Timing Board (ARC-420) Register

Read or write a timing board (ARC-420) register. This is used for debug purposes.

Description: Read function
Command: 0x52575452 ('RWTR')
Argument 0: The 32-bit register address
Return argument: The 32-bit register value

Description: Write function
Command: 0x52575452 ('RWTR')
Argument 0: The 32-bit register address
Argument 1: The 32-bit value to write
Return argument: 'DONE' on success; 'EROR' otherwise

Read or Write the Timing Board (ARC-420) Waveform Generator Table

Read or write to the timing board (ARC-420) waveform generator table. The waveform generator table is 1024 dwords in length.

Description: Read function
Command: 0x52575747 ('RWWG')
Argument 0: The 32-bit table address (0-1023). The table is 1024 dwords in length.
Return argument: The 32-bit waveform value. The upper-16 bits are the delay value and the lower 16-bits are the waveform values.

Description: Write function
Command: 0x52575747 ('RWWG')
Argument 0: The 32-bit table address (0-1023). The table is 1024 dwords in length.
Argument 1: The 32-bit waveform value to write. The upper-16 bits are the delay value and the lower 16-bits are the waveform values.
Return argument: 'DONE' on success; 'EROR' otherwise

Write the Timing Board (ARC-420) DAC

Write to the ARC-420 timing board DAC. This command can write to one or both DACs. Providing one DAC number and value will write to a single DAC, while providing two pairs will write to both DACs.

Descripton: Single DAC write
Command: 0x00575444 ('WTD')
Argument 0: DAC number
Argument 1: DAC value
Return argument: 'DONE' on success; 'EROR' otherwise

Description: Dual DAC write
Command: 0x00575444 ('WTD')
Argument 0: DAC number
Argument 1: DAC value
Argument 2: DAC number
Argument 3: DAC value
Return argument: 'DONE' on success; 'EROR' otherwise

Enable Clock Driver (ARC-430) DC Bias Loop

Enable or disable the DC bias loop. This must be enabled in order to read any of the DC bias values. The loop will run continuously until disabled, updating the values in the read back registers on each loop.

Command: 0x4544424C ('EDBL')
Argument 0: Clock driver board slot number
Argument 1: 1 to enable; 0 to disable
Return argument: 'DONE' on success; 'EROR' otherwise

Read or Write a Clock Driver (ARC-430) DC Bias Value

Read or write a clock driver DC bias value.

Description: Read function
Command: 0x52574443 ('RWDC')
Argument 0: Clock driver board slot number
Argument 1: The channel number. Range 0-11 or 0x1C for the upper level and 0x1D for lower.
Return argument: The DC bias value for the specified channel number.

Description: Write function
Command: 0x52574443 ('RWDC')
Argument 0: Clock driver board slot number
Argument 1: The channel number. Range 0-11 or 0x1C for the upper level and 0x1D for lower.
Argument 2: The value to write
Return argument: 'DONE' on success; 'EROR' otherwise

Read Clock Driver (ARC-430) Reference Voltage

Read the clock driver reference voltage and the 5V value.

Command: 0x00525256 ('RRV')
Argument 0: Clock driver board slot number
Return argument 0: The reference voltage, in volts
Return argument 1: The 5V value

Get or Set the Synchronization Mode

Get or set the controller synchronization mode. This mode allows multiple controllers to be linked together so they all start readout at the same time.

Description: Read function
Command: 0x4753534D ('GSSM')
Return argument: The current mode. 0 = primary mode; 1 = secondary mode.

Description: Write function
Command: 0x4753534D ('GSSM')
Argument 0: 0 = primary mode; 1 = secondary mode
Return argument: 'DONE' on success; 'EROR' otherwise

Local (HxRG) Command Definitions

The following details controller commands specific to the HxRG:

Enable or Disable HxRG Windowing Mode

Enable/Disable the windowing mode using the bottom-most ARC-440 video board in the controller stack, which is auto-detected by the function.

Description: Enable function
Command: 0x0053574D ('SWM')
Argument 0: The window channel number. 7 for H1RG and H2RG, 15 for H4RG.
Argument 1: The row start pixel
Argument 2: The row end pixel
Argument 3: The column start pixel
Argument 4: The column end pixel
Return argument: 'DONE' on success; 'EROR' otherwise

Description: Disable function
Command: 0x0053574D ('SWM')
Argument 0: The window channel number. 7 for H1RG and H2RG, 15 for H4RG.
Return argument: 'DONE' on success; 'EROR' otherwise

Reset HxRG Internal Registers

Resets the HxRG internal registers

Command: 0x00524952 ('RIR')
Argument: none
Return argument: 'DONE' on success; 'EROR' otherwise

Transmit HxRG Serial Interface

Transmit a value to the HxRG over its serial interface.

Command: 0x00545349 ('TSI')
Argument: The 16-bit value to transmit
Return argument: 'DONE' on success; 'EROR' otherwise

Set the Number of Resets

Sets the number of reset reads to be performed before each exposure.

Command: 0x00535243 ('SRC')
Argument: The reset count value
Return argument: 'DONE' on success; 'EROR' otherwise

Set the Number of Fowler Reads

Sets the number of reset reads to be performed before each exposure.

Command: 0x00534652 ('SFR')
Argument 0: The number of pre-exposure reads
Argument 1: The number of post-expose reads
Return argument: 'DONE' on success; 'EROR' otherwise

Reset the HxRG

Reset the HxRG

Command: 0x00525354 ('RST')
Argument: none
Return argument: 'DONE' on success; 'EROR' otherwise