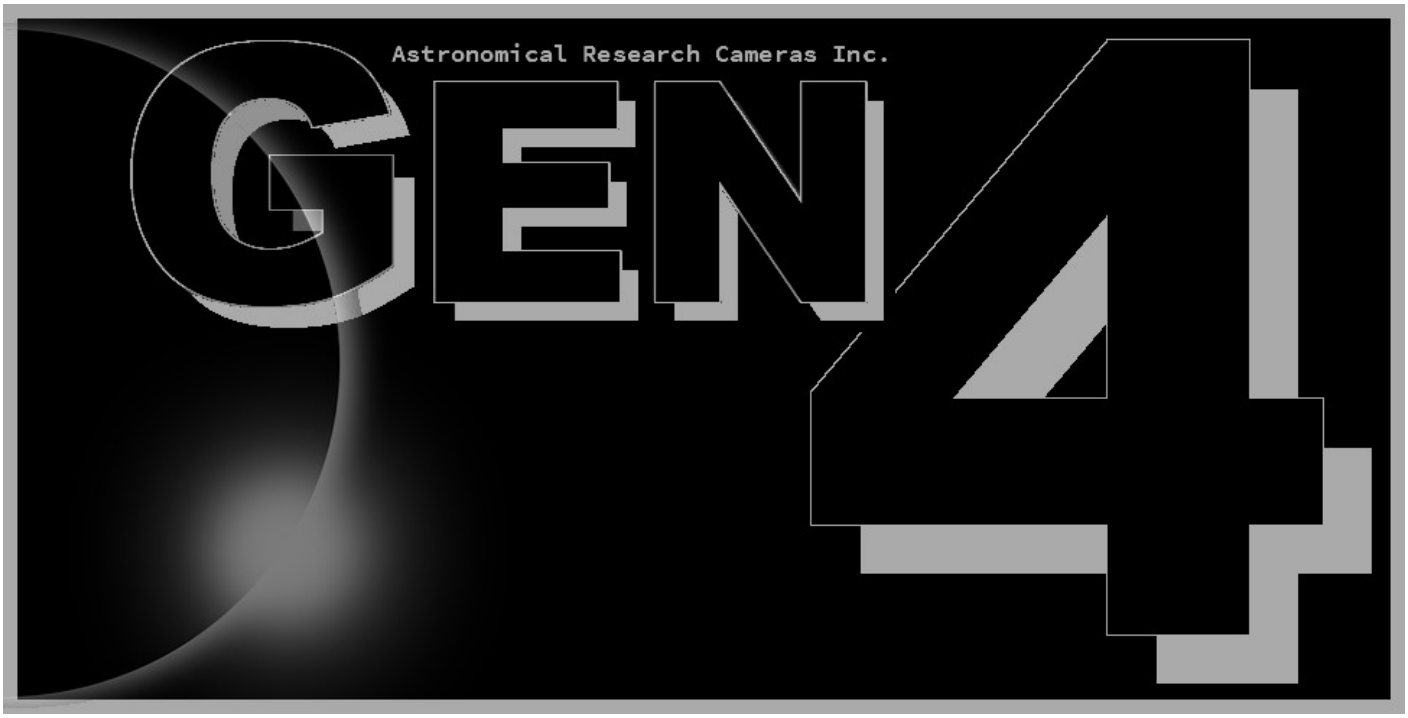


ARC Gen IV
Users Manual

v.002

Scott Streit
Astronomical Research Cameras, Inc.
Tuesday, October 19, 2021

Table of Contents

System Overview

A Gen IV controller consists of a timing board, a clock driver and DC bias board, one or more video boards, a power supply and a housing communicating over a serial link (copper or fiber) at 10 Gb/s to a PCIe interface board located at a distance in a host computer. The timing board contains an ARM micro-controller tightly coupled to an FPGA. The ARM micro-controller executes commands from the host computer, manages exposures and readouts and supervises the FPGA. The FPGA performs real-time functions, especially including the generation of imaging array clocking and video processing timing signals.

G4 Image Acquisition Software

The G4 app is the control and image acquisition software for the Gen IV controller. It is designed to be used for both development and scientific image acquisition. G4 is the simplest and fastest way to get started. Features include controller initialization, temperature, voltage and current monitoring, custom video channel assignment options, individual board settings, developer tools, Java beanshell scripting for customization, image display, FITS and image processing. It is available for both Windows and Linux (Ubuntu LTS 20.04 or newer). It requires the PCIe interface board device driver to be installed.

API Software

The ARC application interface (API) is a set of modern C++ libraries that provide classes used for control, image acquisition, FITS and post-processing. The classes in the API are used by the G4 app. Available for both Windows and Linux (Ubuntu LTS 20.04 or newer), it requires the PCIe interface board device driver to be installed.

Controller Firmware

The Gen IV controller contains an ARM micro-controller manufactured by NXP Semiconductors for command processing and image sequence generation. The ARM micro-controller firmware is C code that can be edited and compiled using the Eclipse-based *MCUXpresso* IDE. The IDE is available from the NXP website for both Windows and Linux. The firmware source code is distributed on the ARC website as MCUXpresso projects that can be imported into the IDE.

The firmware is divided into two parts: (a) user-editable code, which includes waveform generation, imaging array readout and clocking, DC bias voltage setting, exposure sequences, and user customizable commands; and (b) a pre-compiled static library containing all operations independent of the imaging array.

Packets and Communication

The Gen IV controller transfers all data between the controller and the PCIe interface board using packets sent over a fiber or copper cable following the 10GBASE-SR (fiber) or 10GBASE-T (copper) protocol. The fiber operates at 850 nm and the cable can be up to 400 m in length, while the copper cable can be up to 100 m in length. The transceiver is contained in an industry standard SFP+ metal package that is hot-pluggable

into a small metal cage. These packet types include commands, replies, alerts and images. New to Gen IV is the ability for the controller to send an alert message to the host computer. Image packets contain image data and headers that are assembled by each video board for transmission over high speed serial links to the timing board. The timing board assembles these into image packets that adhere to the 10GBASE link protocol for transmission to the PCIe interface board.

Commands and Replies

Every command sent from the host computer generates an acknowledging reply from the controller. This handshaking ensures proper command sequencing across the system. The controller generates a "done" reply packet for all successful commands and an error reply packet, containing a descriptive error code, for unsuccessful commands. A command packet can support up to 252 32-bit arguments and a reply packet can return up to 252 32-bit data values. The command and acknowledge replies consist of printable ASCII characters (0x20 to 0x7E), up to four bytes in length. Any command arguments and additional reply data are not constrained to the printable ASCII character set and may be any 32-bit integer or floating point value. The current list of available commands and acknowledgment replies can be found in the command definition header of the Gen IV C++ application interface (API) device library and the ARM micro-controller firmware source code. The API device library contains all the classes necessary to form command packets and to extract data from the acknowledge reply packets. The ARM micro-controller firmware contains similar functionality. For example, the test data link command, which has a value of 0x0054444C ('TDL' in ASCII), verifies the communications link between the host and controller by sending any 32-bit date value to the controller and verifying that the replied value matches the data value sent.

Waveform Generation

The timing board contains circuitry that writes 16-bit data words to the controller backplane over the pins SS15 ~ SS0 ('SS' for Signal State) at a maximum rate of 50 MHz, or every 20ns. The twelve least significant signals SS11 ~ S0 are connected to twelve clock driver circuits, while SS1 ~ SS0 are also used by the video processor. The four most significant bits SS15 ~ SS12 are used for addressing the video or clock driver board and for selecting some special functions. The current data word is written to the backplane immediately, while a 16 bit word linked to each data word causes a delay until the next data word is written. The delay time is 20 ns times the 16-bit delay number, which is the same time it takes to write a single data word.

Several of these data and delay words are put in what we call a waveform table. Each waveform table contains its length as its first data word. The FPGA circuitry writes the data word at the starting address (plus one) of each waveform table and continues writing each data word in sequence until it counts up to the number 'length'. The circuitry then loops back and repeats itself a programmable number of times. The ARM keeps a list of the waveform table starting addresses, and passes each address and its associated loop counter value to the FPGA to initiate a waveform sequence. The FPGA memory can accommodate up to 1024 waveform data words. These waveform tables are compiled from the ARM micro-controller source code and written to the FPGA after the ARM micro-controller software is download to the controller from the host computer.

These addresses and loop counter pairs are stored in a FIFO (First In First Out) memory in the FPGA. After execution, these pairs are dropped as the FIFO advances to the next pair. The ARM micro-controller needs to write these pairs fast enough that the FIFO is never empty, but not so much that it gets full. The ARC micro-controller generally writes a short burst of these address and loop counter pairs, and is free to process commands in between.

Clock Driver and DC Bias Generation

The clock driver and DC bias board contains twelve DC bias circuits with programmable output voltages from 0 to 5.0 volts with no load. Under a load of 35 mA they supply a maximum voltage of 3.5 V. Their drift with temperature is an excellent 1.5 ppm per °C. There are twelve clock drivers operating between zero and a programmable voltage that has a maximum value of 4.5 volts. They each have a drive capability of 24 mA and 5 ns rise times. The maximum voltages of all these circuits can be set lower with a simple resistor change, in order to keep the imaging array safe. Jitter reducing circuit techniques keep the clock driver jitter under 10 ps rms, as measured with a fast sampling oscilloscope.

Video Processor

The analog signal chain of the video processor consists of a high input impedance, fully differential, balanced input circuit from which a programmable offset voltage can be subtracted to get the signal within optimal range of the A/D converter. This is followed by a differential amplifier that subtracts the two input voltages and drives the differential input of a successive approximation 16-bit A/D converter. The overall gain of the circuit is x6.0, which maps an input voltage range of 0.68 volts to the full range of the A/D converter. The A/D converter produces unitary straight binary, with 0x0 at the highest input voltage (VideoIn+ - VideoIn-) and 0xFFFF at the lowest input voltage. Note that this is effectively an inversion so lower input voltages produce higher image counts. The analog -3 dB bandwidth of the circuit is 4.4 MHz, chosen to be close to the maximum conversion rate of 4 MHz. Both input signals should be kept within the range of -0.9 to +3.6 volts to avoid non-linearities in the input op amps.

The A/D conversion starts at the rising edge of the start conversion signal, labeled CONV in the waveform generator. The falling edge has no effect. The conversion takes 200 ns, and the transfer of its output data an additional 40 ns, resulting in 240 ns as the minimum overall conversion time. The A/D can perform multiple conversion on each pixel, with the FPGA on the video board averaging the resulting digital values. The number of pixels to average can be anything from 2 to 255.

Exposure and Readout

Exposures are initiated by the ARM micro-controller by stopping the imaging array clocking and possibly opening the shutter. The exposure is timed by a 32-bit timer in the FPGA at a resolution of 10 µs, for a maximum exposure time of 12 hours. Readout is initiated by the ARM micro-controller writing readout waveforms to the waveform generator.

The Gen IV controller has a specific constraint on the number of columns in the image. The image column dimension must be evenly divisible by the channel count AND the resulting number of columns-per-channel value must also be a multiple of four. This requirement is imposed by the controllers' FPGA 64-bit packet processing interface. As an example, consider a controller with one video board and all sixteen video channels enabled. An image with 1616 columns is not valid because the number of columns-per-channel equals 101, which is not a multiple of four. However, column numbers 1664 and 1728 are both valid since they are multiples of sixteen (104 and 108 respectively), and their number of columns-per-channel is a multiple of four. Note that this restriction applies to the number of *virtual* channels, as discussed below.
The firmware supports imaging array resets, Fowler reads, up-the-ramp and any combination of these. Up-the-ramp is just a repetition of this sequence, recording all the image data along the way. The sequence does N resets followed by N1 reads, expose, then N2 more reads. Each read gets assigned its own image buffer, so the image data will be stored in N1 + N2 separate image buffers in the host computer.

The device driver for the PCIe interface board allocates two kernel buffers to receive image packets, each of approximately 1.15 MB. The PCIe interface board writes image packets to one buffer while the host computer is reading image packets from the second buffer, alternating as each buffer becomes empty or full. The host computer writes the image packets to a buffer in user space. After completion of the readout, the host computer separates image data from the headers and writes only image data back to the same buffer that contained the image packets, all of this occurring in user space. This re-assembling of the image uses positional information from each packet header that was assigned when the packets were first created by the video board.

Board Numbers and Placement

Board identification jumpers have been eliminated in favor of a mixed hardware/software protocol that assigns board numbers based on their position in the controller board stack and type of board. These board numbers are used by the software for writing to peripherals and transferring image data. The timing and power supply boards have fixed numbers, 0x8 and 0xF respectively. The boards above the timing board will be assigned increasing board numbers going upwards, starting with 0x9. The boards below the timing board will be assigned increasing board numbers, but going downwards, starting with 0x1. The clock driver board must always be the top board in the system. It's recommended that the video boards be evenly distributed above and below the timing board. The board map is displayed in G4's "Controller Setup" window after the controller is initialized. The board map for a 64-channel controller is shown below:

Gen IV Board Map for a 32-channel Controller	
Board number	**Board**
0xA	Clock Driver Board
0x9	Video Board
0x8	Timing Board
0x1	Video Board
0xF	Power Supply Board

Gen IV Board Map for a 64-channel Controller	
Board number	**Board**
0xB	Clock Driver Board
0xA	Video Board
0x9	Video Board
0x8	Timing Board
0x1	Video Board
0x2	Video Board
0xF	Power Supply Board

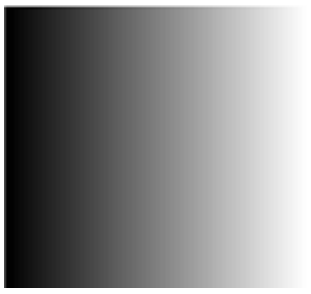
Video Board Channel Assignment

Each video board contains sixteen physical video processor channels. Every physical channel processes its video signal and completes the A/D conversion once initiated. The transfer of each channels' A/D converter data to the timing board can be enabled or disabled in software. A/D converter data transfer can be enabled for a particular physical channel and assigned to a unique virtual channel number that increments sequentially from zero to the desired number of virtual channels (minus one). Image data are transmitted to the host computer in packets that each contain data from only one virtual channel. The virtual channel number is written to the header of each packet and used by the host computer software to re-assemble (or de-interlace) the packets into a proper two-dimensional image. The details of this assembly algorithm will depend on the detector array geometry and the user's intent.

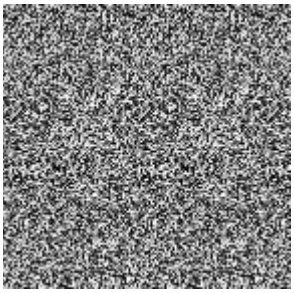
Each virtual channel number can be assigned only once to any given physical channel. The order in which these assignments are made is immaterial, as long as the virtual channel numbers all get assigned with no gaps. Any virtual channel number can be assigned to any physical channel number. This assignment can be made using the ARC G4 channel assignment window or in the ARM micro-controller code. The default assignment is to enable all video channels in the controller. In the figure above, virtual channel numbers are assigned bottom to top, with channels 0 to 15 assigned to the board at the bottom of the stack (0x2), 16 to 31 to the one above it (0x1), and so on. In the final de-interlaced image for an HxRG imaging array the channel wired to channel 0 of the bottom board will appear in the left-most group of columns.

Synthetic Images

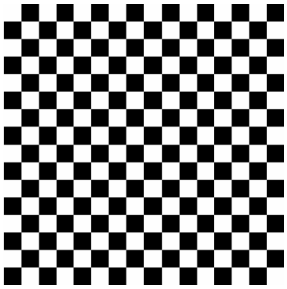
The video board can produce two types of synthetic images. The first type contains a constant in each channel, with channel 0 containing the value 255 and each subsequent channel being incremented by 256 from the previous one. The values will be 255, 511, 767, … etc. The second type is produced by a feature built into the A/D converters that produces a pseudo random image, and has the appearance of a salt-and-pepper image. Histograms of the resulting images should show statistically equal occurrences of data values from 0 to 64k.



Video Board Synthetic Image



Video Board Synthetic Image (ADC)



Timing Board Synthetic Image

The timing board can also produce a synthetic image. This image, however, is fixed in size at 2048x2048 with 16-channels and has a checkerboard pattern. The micro-controller auto sets the dimensions and channel count when the timing synthetic command is received. The micro-controller also contains registers to change the default values of the pattern, which are 0x1234 and 0x6789.

Synchronizing Controllers

The exposure and readouts of two or more controllers can be synchronized. This requires one PCIe interface board for each controller and cabling between the controllers to carry a clock signal and a pulse to indicate the start of readout.

Measuring the Temperature of the Imaging Array

Measurements of the imaging array temperature accurate to 0.1 ºC can be made with RTD temperature sensors connected in a four-wire topology. Less accurate measurements can be made with diodes, thermistors, thermocouples or RTDs connected in a two-wire topology. Several temperature sensors can be connected at one time, even of mixed type. A temperature measurement IC, the LTC2983, needs to be programmed by the user to handle these custom configurations. A default configuration is supported that consists of three RTDs, each one in a four-wire topology. The *Pinouts* document describes the wiring from the timing board to these three RTDs. Procedures for implementing alternative temperature sensors are described in the *Measuring the Temperature of the Imaging Array* Application Note.

Measuring the Circuit Board Temperatures

All the controller circuit boards contain a centrally located IC to measure the board temperature. It operates over the range of -40 to +105 °C to an accuracy of 0.5 °C. The data interface is over the SYS-SPI, a Serial Peripheral Interface controlled by the FPGA on the timing board. Because the conversion time is quite long (240 ms min) it's not feasible to have the host computer initiate a measurement and wait for the result. Instead, the timing board FPGA reads from each of the selected boards one at a time in a continuous loop, storing the results in a table. When the host computer requests a board temperature the value is read from this table. The loop needs to be started and stopped, and is stopped after download. Software on G4 can read these temperature values on demand or plot them as a time series.

Measuring Voltages and Currents

Voltages and currents can be measured at several places and read by the host computer. The twelve DC bias circuits and clock driver generating supply are connected to a multiplexer that connects to a 24-bit slow A/D converter. The FPGA operates a continuous loop as in the circuit board temperatures discussed above. The power supply contains a similar circuit to measure all the currents it supplies, also operating in a continuous loop that is started and stopped. Voltages are not measured but instead monitored by an IC that flags if any of them are out of range. Software on G4 can read these values on demand or plot the total controller power dissipation as a time series.

Shutter

The controller generates an optically isolated digital control signal for controlling a shutter. Power for the shutter can be supplied either by the controller or externally, jumper selectable.

Digital I/O

Eight general purpose signals of digital I/O between the outside world and the ARM micro-controller are provided. Each pin can be used as either an input or an output and used with simple bit manipulation commands in the ARM micro-controller software. The driver is a 3.3 volts part.

Analog Outputs

Two 16-bit DACs are provided for general purpose use. They have an output settling time of 10 µs. Their outputs are buffered to provide 0 to 3.3 volts at 10mA.

Flash Memory

All controller boards, except the power supply, contain a flash memory device, with one partition write-protected from the user that stores factory supplied data, and another partition dedicated to user supplied data. The user may store whatever data is desired. The user partition is not write-protected and can be mistakenly deleted if care is not taken. The user can access these data by opening the desired board window from the Controller Setup Window, then opening the User Block or I.D. Block window. Additionally, the timing board has a firmware block where the ARC micro-controller code may be stored and booted from instead of downloading from the host computer. Refer to the *Flash Memory* Application Note for details on the flash memory read and write commands.

Controller LEDs

The timing board's three LEDs are visible through a small slot in the rear panel. The video board's two LEDs are visible if the rear panel is removed. The center LED on the timing board indicates the status of the link to the PCIe interface board, and the top and bottom links on it and the video boards indicate the status of the links between the boards. The LEDs are OFF if the corresponding link is not functioning, always ON if the link is working but not active, and blinking if the link is active. The top LED indicates the status of communication with boards above it and the bottom LED indicating the status of communication with boards below it.

All controller LEDs can be turned ON or OFF using the 'LED' controller command. The LEDs are turned ON after program download.

Power Supply

The controller power supply is located at the bottom of the housing in a shielded enclosure. It contains a high voltage AC-DC or DC-DC module with input voltage ranges of 90 to 264 VAC and 127 to 300 VDC. There is also a low voltage DC-DC module with an input voltage range of 9 to 18 VDC, 12 VDC nominal. The selection of which module to use is made with two screw terminals. The output voltage of 5 VDC nominal is filtered and then down-regulated by several low EMI DC-DC switching regulators to the voltages required for the controller circuits (+3.3, +2.5, +1.5, +1.2 and -2.8 volts). The -2.8 V supply is used to power the input op amps of the video board so signals close to zero volts in differential pairs can be correctly processed. A 64-channel system consumes 70 watts maximum.

Thermal Control and Housing

There are no fans in the controller. Instead the circuit boards are cooled conductively with thermal plates installed on top of the electronic parts. Small compressible, thermally conductive sheets conduct heat from the electronic parts to the thermal plate, while the thermal plates are pressed against slots in the side plates of the housing by Wedgelocks to enhance thermal conductivity. These Wedlocks need to be tightened with screws so they press the thermal plates against the slots of the side plates. These screws are accessible from the rear of the controller once the rear panel is removed. The power supply also has Wedgelocks that are accessible through little holes in the rear panel. The housing is built from aluminum plates coated with a thin conductive coating, and is held together with metric M3 screws. It is available in two sizes, one to accommodate 16- or 32-channel systems (four slots + power supply) and one for 64-channel systems (six slots + power supply).

32-channel size	185 mm W x 195 mm D x 125 mm H
64-channel size	185 mm W x 195 mm D x 160 mm H
64-channel weight	5.0 kgs = 11 lbs

Programmable Circuits – FPGAs and CPLD

Each of the circuit boards contains a complex programmable element, a CPLD (Complex Programmable Logic Device) on the power supply, or FPGAs (Field Programmable Gate Array) on all the others. Much of the controller complexity and capability is contained in these devices. Revision control is maintained by storing the revision number of the circuit board and the program into the programmable device. These can both be read with the ‘BRI’ = Board Revision Level command. The first value returned is the board revision level, containing an ASCII number and a letter. The second value is the programmable device revision, an ASCII number. Note that the timing board revision number is large, reflecting the enormous effort required to develop it.

Each of these devices is programmed at the factory over a 10-pin connector located near the backplane connector. With the exception of the power supply they are accessible with the circuit boards plugged into each other and in the housing. Users are able to program these devices in the field by purchasing an inexpensive programmer that connects to a host computer USB port, and downloading programming software. Details are described in the *Programming FPGAs and CPLDs* document in the DOCS page of the www.astro-cam.com website.