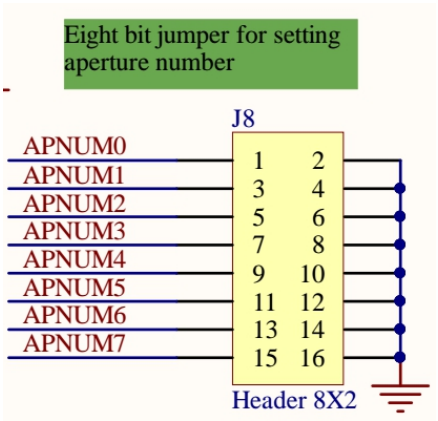
Rick Raffanti, Wei Liu **rev5.1 Feb 22, 2022**

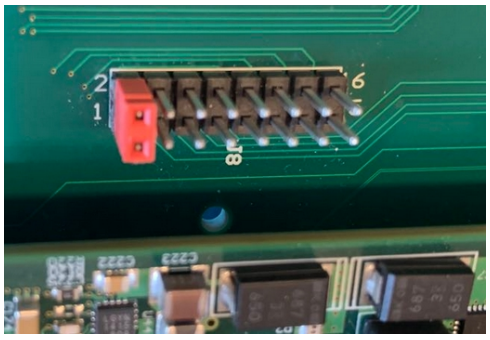
PANO-SETI QUADRANT BOARD PACKET DEFINITION

Each Quadrant board will interface via a 1Gb/s Ethernet connection. Quadrant boards have ARP functionality. We'll set a fixed IP address based on the aperture and quadrant numbers.

IP Setting:
Four quabos will be installed on one mobo. The last two bits of the IP address for the four quabos are 2'b00, 2'b01, 2'b10, 2'b11, which depends on the quabo position.
J8 on Mobo is used for setting bit2 to bit9 in the IP address. APNUM0-7 are pulled up by default, which means the default IP addresses of four quabos on the mobo are
192.168.3. 252 - 192.168.3.255.



For example, if a APNUM0 is connected to GND by a jumper, the IP addresses of the four quabos on the mobo are
192.168.3.248 - 192.168.3.251



Quabo Reboot:
Quabo reboot is based on panoseti_tftp, which is a python package.
You can get the python package and user manual here:
https://drive.google.com/file/d/1hRyxMg7WtEDAkEqUhYwegE81hu_xeOqk/view?usp=sharing

Commands to the Quadrant board from the host:
UDP packets sent to port 60000, length dependent on command type.
The byte at payload offset 0 indicates command type. Command types that don't necessarily produce a response can be set to echo the command packet by setting the MS bit of the command_type byte. Just the SetASICs and Calibrate_Baseline commands need to return a packet.

Echo Response packets of various lengths will be sent from port 60000 of the Quadrant board to the host

Set ASICs command Packet Length 492 bytes

Payload offset	Byte Contents
0	0x81 or 0x01
4	ASIC0, setup[7:0]
5	ASIC0, setup[15:8]
…	….
107	ASIC0, setup[828:824] (3 MS bits unused)
108-131	unused
132	ASIC1, setup[7:0]
133	ASIC1, setup[15:8]
…	….
235	ASIC1, setup[828:824] (3 MS bits unused)
236-259	unused
260	ASIC2, setup[7:0]
261	ASIC2, setup[15:8]
…	….
363	ASIC2, setup[828:824] (3 MS bits unused)
364-387	unused
388	ASIC3, setup[7:0]
389	ASIC3, setup[15:8]
…	….
491	ASIC3, setup[828:824] (3 MS bits unused)

setup[828:0] is the 829-bit shift register described in the MAROC data sheet used for setting amplifier and shaper parameters, trigger masks, test enables, etc.

The Echo response to this command (only) will consist of the data read back from the MAROCs rather than that written to the MAROCs. The first time the chips are read, the response will not match the command; thereafter a repeated load of the same data should produce a match of the 829 bits from each chip (the unused bits in the packet will not necessarily match).

Set HVs command Packet Length 64 bytes

Payload offset	Byte Contents
0	0x82 or 0x02
1	unused
2	HV0[7:0]

3	HV0[15:8]
4	HV1[7:0]
5	HV1[15:8]
6	HV2[7:0]
7	HV2[15:8]
8	HV3[7:0]
9	HV3[15:8]
10-63	unused

HVx is set in steps of -1.14mv, -75v max

Set Acquisition command Packet Length 64 bytes

Payload offset	Byte Contents
0	0x83 or 0x03
1	unused
2	acq_mode
3	unused
4	acq_interval[7:0]
5	acq_interval[15:8]
6	4'h0, hold1[3:0]
7	0
8	4'h0, hold2[3:0]
9	0
10	0 (adc_clk_phase is hard-coded in firmware)
11	0
12	PH_stop_channel
13	0
14	7'h0, stim_on
15	0
16	stim_level,
17	0
18	5'h0, stim_rate[2:0]
19	0
20	7'h0, en_WR_UART
22[2:0]	flash_rate, 0 to 7
24[4:0]	flash_level, 0 to 31
26[3:0]	flash_width, 0 to 15
28-63	unused

acq_mode[7:0] = 0x0 for data-taking disabled
0x01 for Image Mode(Packets generated in MB core),
0x02 for PulseHeight Mode(PHM)
0x11 for PH Mode, no BL subtract
0x03 for 16-bit High Speed Image Mode(HS-IM)
0x06 for 8-bit High Speed Image Mode
0x07 for Simultaneous Mode(8-bit HS-IM and PHM)

acq_interval[15:0] for image mode, interval, in steps of 1us, over which pixel triggers are accumulated.

hold1 and **hold2** are the 4-bit hold delay values; they set the delay of the HOLD signal assertions after trigger reception, in 5 ns ticks. We are only using one HOLD signal, and probably will leave the delay value at 0, since the internal MAROC delays are so large that a zero setting results in the slow shaper signal being held at its peak, when the fastest shaper settings are used

adc_clk_phase It's hard-coded in firmware now

PH_stop_channel stops the analog readout of the MAROCs at a given channel, to permit looking at the fast shaper signal from that channel on the SMA testpoint output, or the slow shaper signal on the PCB testpoint. Because there is a 6-sample latency through the LTC2170 this value must be set to 70 to do a complete readout of the 64 pixels per chip.

stim_on is set to 1 to turn the on-board stimulation generator on,
stim_rate can be set from 0 to 7, to vary the rate from 190 to 24,400 Hz

stim_level is an 8-bit value that sets the amplitude of the stim pulse. 255 corresponds to a 3.3v step, and each channel has a Ctest input capacitor of 2pF, so full-scale is nominally 6.6pC of charge, or 40M e-. Given the nominal 1.7 Me- gain of the SiPMs, this is about 23 pe.

flash_rate is a 3-bit value controlling the rate of flashes from the Mobo LED:

0	1PPS (sync'ed to the internal 1PPS)
1	95 Hz
2	191 Hz
3	381 Hz
4	763 Hz
5	1526 Hz
6	3052 Hz
7	6104 Hz

flash_level controls the level of the DC supply to the pulser, 0 to 10v in 312mv steps
flash_width controls the width of the pulse driving the LED, 1.5 to 24 ns in 1.5ns steps

en_WR_UART is a bit that, when set, directs the White Rabbit shell command and reporting to the UART connection on J3, rather than using the UDP interface.

Reset command — Packet Length 64 bytes

Payload offset	Byte Contents
0	0x84 or 0x04
1-63	tbd

Resets state machines and packet counter.

Set Stepper command — Packet Length 64 bytes

Payload offset	Byte Contents
0	0x85 or 0x05
4, 5	#steps, signed 16b number
6 [2]	focus limits on
8	fan speed, 0 to 15

Set Channel Mask command Packet Length 64 bytes

Payload offset	Byte Contents
0	0x86 or 0x06
1-3	unused
4	chip0mask[7:0]
5	chip0mask[15:8]
6	chip0mask[23:16]
7	chip0mask[31:24]
8	chip0mask[39:32]
9	chip0mask[47:40]
10	chip0mask[55:48]
11	chip0mask[63:56]
12-19	chip1mask[63:0]
20-27	chip2mask[63:0]
28-35	chip3mask[63:0]
36	OR-masks[7:0] (see below)
37	OR-masks[8] (see below)
38-63	unused

These are the channel mask bits, one per pixel, which can be set in the firmware to disable any number of channel triggers. Channel triggers can also be disabled by writing to the MAROC setup registers, but setting the mask using this command is faster and doesn't disrupt the MAROC settings. The channel mask bits are in channel order as numbered in the Maroc chip, these are related to the pixel locations through a lookup table (which is different for the BGA and QFP versions)

The OR-masks bits are as follows:

[0]	chip0_OR1
[1]	chip0_OR2
[2]	chip1_OR1
[3]	chip1_OR2
[4]	chip2_OR1
[5]	chip2_OR2
[6]	chip3_OR1
[7]	chip3_OR2
[8]	multi-chip OR
[15:9]	unused

Each MAROC has two ORed outputs and there is a single mask bit for each of these. There is also a wire-ORed connection among all four Quabos which we've never used, which can be masked by setting the multi-chip OR bit.

Calibrate PH Baseline command Packet Length 64 bytes

Payload offset	Byte Contents
0	0x07
1-63	unused

The quabo sends back a list of 256 numbers, starting at byte 4 of the packet, with the measured offset for each channel, a 16-bit little-Endian value. We don't actually use this info, and it might be useful for debugging.

shutter command Packet Length 64 bytes

Payload offset	Byte Contents
0	0x88 or 0x08
1 [0]	shutter open/close (high is open)

led flasher selection command Packet Length 64 bytes

Payload offset	Byte Contents
0	0x89 or 0x09
1 [0]	select new/old led flasher (high is for the new led flasher selection)

IM-PH IP setting command Packet Length 64 bytes

Payload offset	Byte Contents
0	0x8a or 0x0a
1	PH_IP_3 (e.g. 192)
2	PH_IP_2 (e.g. 168)
3	PH_IP_1 (e.g. 1)
4	PH_IP_0 (e.g. 100)
5	IM_IP_3 (e.g. 192)
6	IM_IP_2 (e.g. 168)
7	IM_IP_1 (e.g. 1)
8	IM_IP_0 (e.g. 99)

This command is used for setting IP addresses for PH packets and IM packets. The default IP address is 192.168.1.100. After you send this command, you will get a reply from quabo, which contains the Mac addresses for the PH and IM packets. The length of the reply is 12 bytes. The first 6 bytes are the Mac address for PH packets, and the last 6 bytes are the Mac address for IM packets.

HK IP setting command Packet Length 64 bytes

Payload offset	Byte Contents
0	0x8b or 0x0b
1	HK_IP_3 (e.g. 192)
2	HK_IP_2 (e.g. 168)
3	HK_IP_1 (e.g. 1)
4	HK_IP_0 (e.g. 100)

This command is used for setting IP addresses for HK packets. The default IP address is 192.168.1.100.

Set Housekeeping Interval command Packet Length 64 bytes

Payload offset	Byte Contents
0	0x20
1	hk_interval
2-63	unused

This command enables or disables the housekeeping output. If hk_interval is set to 0, housekeeping output is disabled. Else, the housekeeping packet rate is about (hk_interval * 3) seconds. At startup, this value is set to 1, so a packet is sent out every 3 seconds. (The delta-sigma housekeeping ADC is slow, getting through a complete set of 16 conversions in about 2 seconds, so sending out packets more frequently would result in a duplication of values.)

White Rabbit Shell Command — Packet Length

Payload offset	Byte Contents
0	0x30

Housekeeping packet of 64 bytes:

UDP packets sent to port 60002

Offset	Byte Contents
0	0x20
1	bootbyte
2	BOARDLOC[7:0]
3	BOARDLOC[15:8]
4	HVMON0[7:0]
5	HVMON0[15:8]
6	HVMON1[7:0]
7	HVMON1[15:8]
8	HVMON2[7:0]
9	HVMON2[15:8]
10	HVMON3[7:0]
11	HVMON3[15:8]
12	HVIMON0[7:0]
13	HVIMON0[15:8]
14	HVIMON1[7:0]
15	HVIMON1[15:8]
16	HVIMON2[7:0]
17	HVIMON2[15:8]
18	HVIMON3[7:0]
19	HVIMON3[15:8]
20	RAWHVMON[7:0]
21	RAWHVMON[15:8]
22	V12MON[7:0]
23	V12MON[15:8]
24	V18MON[7:0]
25	V18MON[15:8]
26	V33MON[7:0]
27	V33MON[15:8]
28	V37MON[7:0]
29	V37MON[15:8]

Offset	Byte Contents
30	I10MON[7:0]
31	I10MON[15:8]
32	I18MON[7:0]
33	I18MON[15:8]
34	I33MON[7:0]
35	I33MON[15:8]
36	TEMP1[7:0]
37	TEMP1[15:8]
38	TEMP2[7:0]
39	TEMP2[15:8]
40	VCCINT[7:0]
41	VCCINT[15:8]
42	VCCAUX[7:0]
43	VCCAUX[15:8]
44	UID[7:0]
45	UID[15:8]
46	UID[23:16]
47	UID[31:24]
48	UID[39:32]
49	UID[47:40]
50	UID[55:48]
51	UID[63:56]

52[0]	SHUTTER_STATUS	58	FWTIME[23:16]
52[1]	LIGHT_SENSOR_STATUS	59	FWTIME[31:24]
53	Unused	60	FWVER[7:0]
54	Unused	61	FWVER[15:8]
55	Unused	62	FWVER[23:16]
56	FWTIME[7:0]	63	FWVER[31:24]
57	FWTIME[15:8]		

*52[0] means bit0 in byte52.
bootbyte is 0xaa for the first housekeeping packet sent out after CPU boot; 0 thereafter.
TEMP2, VCCINT, and VCCAUX are read from the FPGA and have different conversion formulas, see below

BOARDLOC[15:0] is the aperture numer (8b) concatenated with the quadrant number (2b) padded with zeroes
HVMONx[15:0] is the bias voltage to detector array x, in steps of 1.22mv/LSB (0 to -80v)
RAWHVMON[15:0] is the nominal -70V supply voltage in steps of 1.22mv/LSB (0 to -80v)
HVIMONx[15:0] is the current drawn by detector array x: Iarray = (65535 – HVIMON) * 38.1nA (2.5mA max)
V12MON[15:0] is the voltage of the 1.2v supply to the FPGA MGT, 19.07 uv/LSB
V18MON[15:0] is the voltage of the 1.8v supply to the MAROCs, 19.07 uv/LSB
V33MON[15:0] is the voltage of the 3.3v 1.2 to the MAROCs, 38.1 uv/LSB
V37MON[15:0] is the voltage of the 3.7v switching supply, 38.1 uv/LSB
I10MON[15:0] is the current of the 1.0v supply to the FPGA core, 182uA/LSB
I18MON[15:0] is the current of the 1.8v supply to the FPGA, 37.8uA/LSB
I33MON[15:0] is the current of the 3.3v supply to the FPGA, 37.8uA/LSB
TEMP1[15:0] is the temperature read from the TMP125, temp = .0625*N, in degrees C (signed quantity)
TEMP2[15:0] is the temperature read from the FPGA, temp = N/130.04 -273.15 in deg C
VCCINT and VCCAUX are measured by the FPGA volts = N*3/65536 in volts
UID[63:0] is the 64-bit unique ID that is read from the DS18B20 chip
UTC[31:0] is Universal Time, in seconds, from the White Rabbit infrastructure
NANOSEC[31:0] is number of nanoseconds since the last tick of UTC

UTC is not yet implemented (will come from the White Rabbit system). UID is not implemented. NANOSEC is the value from the 10MHz/1PPS system and is actually in 3.125ns ticks (320MHz clock).

Science We have two kinds of packets sent from each quadrant board to the host's UDP port 60001, 528 bytes payload, as follows:

Payload offset	Byte Contents
0	acq_mode[7:0]
1	packer_ver
2	packet_no[7:0]
3	packet_no[15:8]
4	BOARDLOC[7:0]
5	BOARDLOC[15:8]
6	UTC[7:0]
7	UTC[15:8]
8	UTC[23:16]
9	UTC[31:24]
10	NANOSEC[7:0]
11	NANOSEC[15:8]
12	NANOSEC[23:16]
13	NANOSEC[31:24]
14-15	unused
16	pixel0[7:0]
17	pixel0[15:8]
…	…
526	pixel255[7:0]
527	pixel255[15:8]

272 bytes payload(only for 8-bit image mode), as follows:

Payload offset	Byte Contents
0	acq_mode[7:0]
1	packet_ver
2	packet_no[7:0]
3	packet_no[15:8]
4	BOARDLOC[7:0]
5	BOARDLOC[15:8]
6	UTC[7:0]
7	UTC[15:8]
8	UTC[23:16]
9	UTC[31:24]
10	NANOSEC[7:0]
11	NANOSEC[15:8]
12	NANOSEC[23:16]
13	NANOSEC[31:24]
14-15	unused
16	pixel0[7:0]
…	…
272	pixel254[7:0]
271	pixel255[7:0]

In PulseHeight Mode, when any (unmasked) trigger is asserted, the system reads the 12-bit amplitude from all 256 pixels. The 12-bit value and the trigger assertion are combined into a 16-bit word.

Now, High-Speed Image mode is implemented. The image packets are generated in FPGA, instead of microblaze software, so the minimal integration time can get to 10us. Here are two High-Speed Image Modes: 16-bit Image Mode and 8-bit Image Mode.

(1) In 16-bit Image Mode the system will count trigger pulses occurring on each (unmasked) pixel's trigger_out signal and report back the total, pixel__[15:0] every acq_interval *1 microseconds. Counters will be reset after each interval.

(2) In 8-bit Image Mode the system the system will count trigger pulses occurring on each (unmasked) pixel's trigger_out signal and report back the total, pixel__[7:0] every acq_interval *1 microseconds. Counters will be reset after each interval. The pixel__[7:0] is the low 8bits in pixel__[15:0] in the 16-bit Image Mode, and it will be frozen to 255, when the counter gets to 255.

acq_mode is the mode, set by the acq_mode command

packet_ver is the packet version. For now, the packet_ver in the IM packets is 0. The packet_ver in the old PH packets is 0, which means the ph data in the packets is 12-bit, unsigned data; the packet_ver in the new PH packetsd is 1, which means the ph data in the packets is 16-bit, signed data.

packet_no[15:0] is a 16-bit packet sequence number, can be used to check if the software is getting them all. Reset with the Reset command, wraps at full-scale.

BOARDLOC[15:0] is the aperture number (8b) concatenated with the quadrant number (2b) padded with 0s.

UTC[31:0] is Universal Time, in seconds, from the White Rabbit infrastructure
NANOSEC[31:0] is number of nanoseconds since the last tick of UTC

NANOSEC is captured when the trigger happens in PH mode, and it's also included in High-Speed Image mode(16-bit Image mode and 8-bit Image mode).