Implementation of the Synchronous DAQ Waveform

There will be two internal buffers of waveform data on each controller—one that is being updated and one that is not. When the trigger is received, the new waveforms will be written to the first buffer, leaving the second buffer static.

Email from Terry on 5/23/12:

To readout the ‘sync waveform data’

* Request data count must be a multiple of 640 (decimal)
* Plus 4 words for header

To read one buffer from master, command would be ‘rdx 01000000 644’

Right now any request less than 644 only returns OAC header plus 4 word time stamp.

Email from Terry on 11/12/12:

* x100-0x103 UDP TOD (4 words)
* x104-0x107 TELNET TOD Synchronous DAQ Waveform, sync time (4 words)
* x108 SDAQW RESET (16bits)
* x109 SDAQW TRIG (16bits)
* x10A SDAQW COUNT DOWN (16bits)
* x10B SDAQW ENABLE (16bits)
* x10C Trigger Counter, UDP r/w (16bits)
* x10D Test Register, UDP r/w (16bits)
* x10E uC Firmware Version
* x10F reserved registers (16bits)

Interpretation

|  |  |  |  |
| --- | --- | --- | --- |
| Address | Value |  | Comments |
| 100 | YYMM | From Linac Front end node | Time of Day from UDP host; Year, Month (January=1) |
| 101 | DDHH | Day (starting with 1 each month), Hour |
| 102 | MMSS | Minute, Second |
| 103 | CC00 | Cycle number (0000 through 0E00) |
| 104 | YYMM | From OAC | Time of Day to start looking for acquisition |
| 105 | DDHH |  |
| 106 | MMSS |  |
| 107 | CC00 |  |
| 108 | 0012 |  | Reset event (when to begin looking for trigger) |
| 109 | 0052 |  | Trigger event (when to take data) |
| 10A | 0001 |  | Countdown of trigger events |
| 10B | 0001 |  | 1: Enable; 2: TOD Match; 3: Trigger received; 4: Latched |
| 10C | nnnn |  | Trigger counter (R/O) |
| 10D | tttt |  | Test register |
| 10E | 0136 |  | Micro-controller version number, 1.36 (R/O) |
| 10F | xxxx |  | Reserved |

The signal to define the 15 Hz pulse on which the trigger occurs will contain these data:

* The value of the TCLK/LCLK reset event
	+ E.g., a supercycle reset
* The value of the beam trigger event
* The number of these trigger events to skip before actually triggering
* The expected time stamp of the reset event (in an appropriate internal format for the controllers).
	+ This is to disambiguate the reset event in the case when the trigger request message is received at about the same instant as a reset event (some controllers might get it just before and some get it just after).