Booster Beam Loss Monitors

Local application support

Mon, May 1, 2000

The local application BLMS provides the support for processing Booster beam loss monitor signals. It captures the waveforms digitized by the Swift digitizer and writes a data stream record for each digitized waveform. It also calculates dose summations over a time interval for each portion of the Booster acceleration cycle.

Waveform data

Using the Swift digitizer, eight waveforms are digitized and stored into memory on an IndustryPack card. These beam loss monitor signals are actually outputs of a log amplifier, so they must be de-logged first into real beam loss signals before any calculations can be done. The resulting waveforms are written into a data stream, one for each signal, including a header that describes parameters relating to the waveform measurement. The data streams can be accessed via RETDAT, in order to permit up to 15Hz monitoring. They can also be accessed from FTPMAN, when an isolated snapshot is to be captured for display. Although the FTPMAN client cannot keep up with 15Hz, it is the Acnet standard client interface for displaying waveforms.

The data stream record header can include the following information:

Field	*Size*	*Meaning*
waveform	4	Address of waveform in Swift digitizer memory
reset	2	Booster reset clock event number
spare	2	—
time	8	Time-of-day to a 15Hz cycle
delay	2	Delay time in μs
rate	2	Digitize rate in kHz
spare	12	—

The size of the header is 32 bytes, allowing for room to add more fields.

Within a single data stream, the waveform address should be constant. The reset clock event may vary according to a parameter of WFDS, which describes which resets should cause a waveform data stream capture. The delay and digitize rates are set by a channel for each. They will normally remain fixed, unless it is decided to alter them on behalf of all users.

Selected data point from waveforms

In the current system support for BLMs, there is an application page B36 that allows seeing data values from all BLMs at one time in the waveforms. It works by requesting data from those A/D channels based upon a clock event 0x0B. In the front end, the signals are measured by the A/D to yield this value. One disadvantage of this solution is that all users interested in such data must share the same sampling time. It would be better if each user of this page B36 could specify a time for sampling without interference with others.

To support such capability, one could define a database name that is accessible via FTPMAN to obtain waveforms, but it is accessible via RETDAT to obtain a sampled data point from the waveform. One way to do this is to use the offset word in the RETDAT

request to specify which data point of the waveforms is to be sampled. The new version of B36 could make the requests for these devices using the offset word to indicate the point within each waveform. The application would know that the digitization is being done at 12.5KHz, so that it can convert a time into a point number, and the point number could be the offset value. Alternatively, the offset value could be twice the point number, so that it represents a kind of byte offset into the waveform array. If there is a nonzero delay after a reset event before digitizing the waveform, the program may need to take that into account. It would be simplest if the program supported setting the delay in a way such that all delay devices were targeted at once. There may be 10 IRMs that support access to these waveforms, and it may be convenient if all of them used the same delay. There may be a way to enforce this in the front ends; one front end node could copy such a delay setting to all others that support Booster BLMs.

To add this capability to RETDAT requires new support for the offset word. Currently, it is used to modify the 2nd word of an ident such as the channel number. It can also be used to modify the 3rd word of an ident. To use this second method, we need to design a new listype that is appropriate. Its ident would include the channel number and a base point, which would be defined as zero in the Acnet database. The application could specify via the offset word the data point desired. In this way, the current RETDAT support would be sufficient to support this case.

The support for this new listype would involve checking the CINFO table for a match on the channel number. This would include the location of the waveform data. The internal pointer could simply be the address of the data point indicated. If there is no match in the CINFO table, simply return a zero for the value. One should not normally make such requests. Alternatively, one could return a new error code.

Another method is to formally add support for this case in RETDAT, using a different flag value for offset support. The offset flag value is a 4-bit field in the upper nibble of the low byte of the first word in the SSDN. Values of 0, 1, and 2 are used, so a value of 3 could be added as another option. For a reading listype, the CINFO table would again have to be checked in order to determine the location of the waveform.

For an ordinary request for 2 bytes of data via RETDAT, with a zero offset, what should be returned? Perhaps it could be the last value of the waveform, since that represents the total accumulated beam loss for this location. This would mean that the first data point in the arrray could not be accessed, unless the offset word were treated as an index in the range 1–n rather than 0–m. An offset word of zero would yield the final, maximum value of the waveform.

Dose calculations

Beam loss should be measured across certain time intervals with the waveforms. Since the waveforms are an integrator measurement of beam losses, one should take the reading of loss at the end of the interval minus the loss at the end of the previous interval. (Of course, one must convert the log amplifier output to rads/second units first before computing differences.) For the first interval, there is no previous interval, so the first data point of the waveform should be addressed. The specification of the ranges to be used for intervals should be determined globally, perhaps by a sequence of channels in each front end that has such waveform data.