MiniBooNE BPMs

MBPM local application

Tue, May 7, 2002

The BPM signals for the MiniBooNE target area need software support via a local application. This note describes what `LOOPMBPM` does. It is based on the document by Craig Drennan called *Requirements for BPM Data Processing*.

There are 4 BPMs, each of which uses two analog channels of an 8-channel Quicker Digitizer IndustryPack module in one IRM. The QD will operate at a 10 MHz sampling rate, triggered by a clock event plus a delay that brings it near to the time of the 1.6 μs MB beam pulse. (With newer digitizers installed, the maximum rate can be increased to 20 MHz, if needed.) The QD digitizes all 8 channels simultaneously, storing the results into a memory. (It actually digitizes 16K samples for each channel, capturing the readings from each channel into a separate FIFO, then depositing all these FIFO outputs into memory over about 105 ms of time. If a new trigger occurs sooner, however, it starts over at the beginning of memory.)

There is concern about pedestals, or baselines, for each of the 8 signals, so these are measured prior to the captured beam pulse data. This means that the event delay should be set so that the QD is triggered at least a microsecond before the beam pulse occurs. At least 5 points should be averaged to determine these pedestals.

The exact time at which the beam pulse occurs may exhibit some jitter, so the software must notice the rising edge of the beam pulse by examining the baseline-subtracted data samples. For any given beam pulse, all 8 channels should exhibit the same delay, but it may be useful to check that this is so. The algorithm to be used in detecting the rising edge is somewhat in question. A specified number of samples after this rising edge defines where the summations begin for deriving the beam position results.

LA parameter layout

The parameters of the LA are as follows:

Prompt	*Meaning*
`ENABLE B`	`LA enable bit`
`#BPMS`	`number BPMs (1–4)`
`SIGNAL C`	`signal base channel#`
`POSITN C`	`position base channel#`
`CALIB  C`	`calibration coefficients base channel#`
`#BASE`	`#points used to compute baseline pedestals`
`#RISE`	`rising edge offset to first data point`
`#PTS`	`#points used to compute positions`
`EVENT# C`	`event/Bit number channel#`
`--`	`(spare)`

From 1–4 BPMs can be processed, each one consisting of two consecutive channels, A and B. Each pair of channels follows in succession. It is expected that the `#BPMS` parameter will be 4, and all 8 channels of the QD will be used to support these 4 BPMs. The `SIGNAL` base channel# will be the base channel for the consecutive set of channels that map to 8 channels of the QD.

The `POSITN` base channel# points to the 4 position result channels. The following channels will hold the 4 average data values, the 8 baselines, and the 8 rising edges.

The `CALIB` base channel number points to a series of channel numbers that will include 4 channels for each BPM. Each set of 4 channels will yield the `m1`, `m2`, `m3`, and `Sx` values, all of which are used for the position calculations. In case there is no calibration base channel specified, default values for `m1`, `m2`, `m3`, `Sx` of 0.98947, 0.02825, 0.01275, 30.842 are used. (This default values feature has probably already outlived its usefulness.)

The `#BASELN` parameter specifies how many consecutive samples are used to derive an average that will be the baseline value to be subtracted from each raw data value when calculating positions. There is a separate baseline value computed for each of the 8 signals. The starting point for computing this average is a constant in the program that is currently set for the 2nd point.

The `#RISE` (rising edge offset) parameter specifies the number of points beyond the signal rising edge at which the position calculation summation begins. The `#PTS` parameter specifies the number of points to be used in the summations. The minimum threshold above the baseline for detecting a rising edge is a constant in the program, which is currently set to 100 mv. This constant is used in case the mean between the baseline and the maximum data value is less than this amount above the baseline, implying the basic signal is very small. The current version of the program scans the data for the maximum data point. The mean of this value and the baseline is used as a threshold for detection of the rising edge. (This threshold is also made at least 100 mv higher than the baseline.)

The current first version of the code assumes its own knowledge of where in memory it can find the QD digitized data arrays. An alternative is to use the `CINFO` table, in which an entry would be made for each of the channels indicated by the `SIGNAL` parameter.

The LA is invoked at 15 Hz, a few milliseconds after the time for Booster extraction. By that time, all necessary digitized data values are already stored, as we need no more than 64 points for each of the 8 channels. (This small amount of data should already be in memory by 0.5 ms after the QD trigger.) On MiniBooNE cycles, which are marked by `0x1D` reset clock events, the necessary computations are performed and the position results stored into the result channel readings. The `EVENTS#` parameter allows specifying a channel number whose reading specifies a clock event number or a nonzero Bit#, in which case position calculations are triggered by the event or a “1” state of this bit reading. (To be interpreted as a Bit#, the reading should be >= `0x0100`.)

First, the baselines, or pedestals, are determined for each of the 8 signals. Then the rising edges are detected for each signal. (These should be assumed to be equal.) If the rising edge is not recognizable for either A or B, the position P is set to 32.767 mm; otherwise, this formula is used:

```
P = Sx * numer / denom
```

where

```
numer = m1*sum(A) – sum(B) + nPts*(m2 – m3)
denom = m1*sum(A) + sum(B) + nPts*(m2 + m3)
```

The `sum(A)` notation indicates the sum of the pedestal-subtracted data points for one signal.

The `m1`, `m2`, and `m3` coefficients are designed to correct for small deviations of the gain and offset of each of the signals. These values can be derived either online or off-line. The `Sx` value is the detector sensitivity scale factor. It is expected to be the same for all 4 BPMs, but the program allows them to be different.

The units of position are in millimeters, and the position channels use full scales of 32.768, so that the least significant bit represents one micron resolution.

The program captures the 64 data points of each waveform to facilitate checking and debugging. In the static memory, one can find the last set of data on which the program calculated positions.

Any of the parameters of the program can be modified while the program is running. About every 2 seconds, the parameter values are reviewed for changes. Changes that can be modified online include the calibration constants. (The two exceptions for which modification would require disable/enable toggling of the enable bit are the base channel#s for the signal and positions.)

The 16 calibration channels are raw floating point channels. These calibration constants can be set via an Acnet parameter page. They can also be set via the latest version of the "little console" parameter page. The parameters listed above may be set using Page E on the little console or by using the analogous page on an Acnet console.

The result channels can of course be viewed on an Acnet parameter page, or on a little console.

The current set of parameters in use for test `node0642` are:

```
E LOCAL APPS      05/07/02 0946
NODE<0642>  NTRY<24> 64  H<0508>
NAME=MBPM   CNTR=58  DT= 0.5  MS
TITL"MBOONE BEAM POS MONITORS"
SVAR=00046D16     05/07/02 1515
ENABLE  B<00E1>*MBPM ENABLE
#BPMS    <0004>
SIGNAL  C<02F0> BP1AQK 0     MM
POSITN  C<02C0> BPM1P  0.919 MM
CALIB   C<0280> BPM1M1 0.989 NA
#BASE    <0006>
#RISE    <0003>
#PTS     <0008>
EVENT#  C<00D0> MBPMEV 29    HEX
         <0000>
```