Booster Wire Scanner Integrator

October 30, 2009

Jim Lackey, Craig Drennan

I. Introduction

II. Charge Input Integrator

The Booster Wire Scanner Integrator is a 4 channel 6U VME module used to integrate and digitize charge signals from the wire scanner. The input integrator section can provide a continuous charge measurement. Signals for each input are switched between the two channels of a TI/Burr-Brown ACF2101 integrator chip every 20 us. Figure I.1 is a simplified illustration of the integrator circuit. As one channel is integrating, the other is being digitized and reset. The digitized, 21 us integration results can subsequently be summed into longer integration intervals or charge currents (Amps) over various intervals can be computed via different methods of averaging the 21 us integration results.

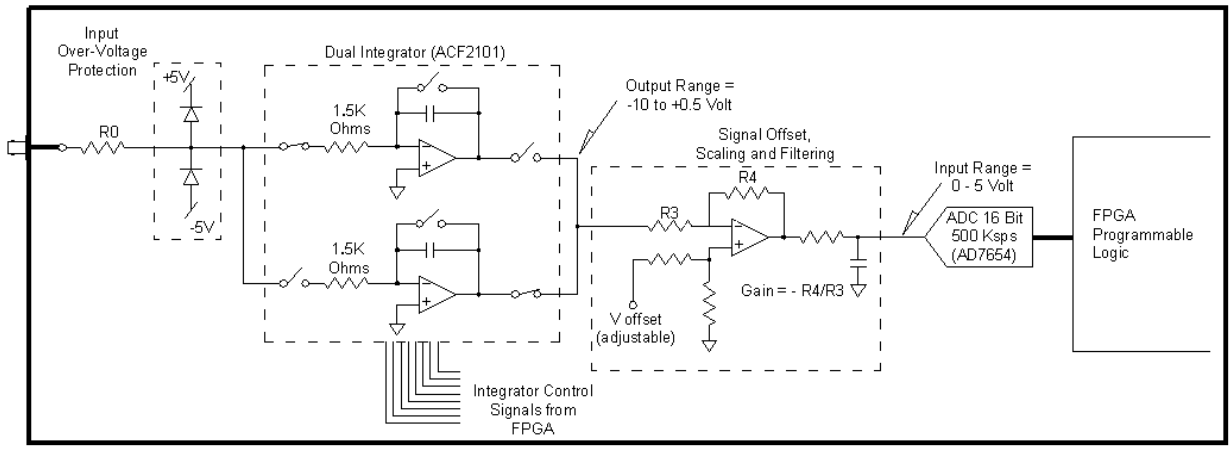


Figure I.1 Simplified illustration of the integrator circuit.

For the current wire scanner application, measurement begins upon receipt of a TTL trigger. The initial 4, 20 us integration results are averaged and set as a baseline for the following measurements. The baseline value is subtracted following 32 integration results and the resulting values are output as analog signals using digital to analog converters.

III. FPGA Integrator Control and Signal Processing

Two FPGA's on the integrator module are employed to control the integrators, interface with the signal digitizers (ADC's) and the digital to analog converters (DAC's), and perform the digital signal processing. As mentioned, the integration is performed by ping-ponging between two integrating opamps. This involves sequencing the Hold, Reset, and Select switches of each integrator. This task and the control of the signal digitizer is performed in the Upper FPGA. The digital output of the ADC's is routed to both the Upper FPGA and the Lower FPGA. Since it is the Lower FPGA that controls the DAC's, it is in the Lower FPGA that we do the signal processing.

The FPGA's are manufactured by Altera and hence the development environment used to write and compile the FPGA code is Quartus. The code is a number of files in a project folder. The folder name will indicated whether the code is for the Upper or Lower FPGA and provide some indication as to whether the code is current or at least provide a revision date (yymmdd). In order to examine and or edit the code the user should open the Quartus project file within the project folder. This file may have an odd name, but will likely be the only file with a ".qpf" filename extension.

In order to load new FPGA code into the integrator module you must use the ATMEL, "CPS.EXE" programming application and the ATMEL programming cable. The FPGA's load their code at power up from ATMEL EEPROM memory. Figure III.1 shows a photo of the programming cable connected to program the Upper FPGA and Figure III.2 shows a photo for programming the Lower FPGA.

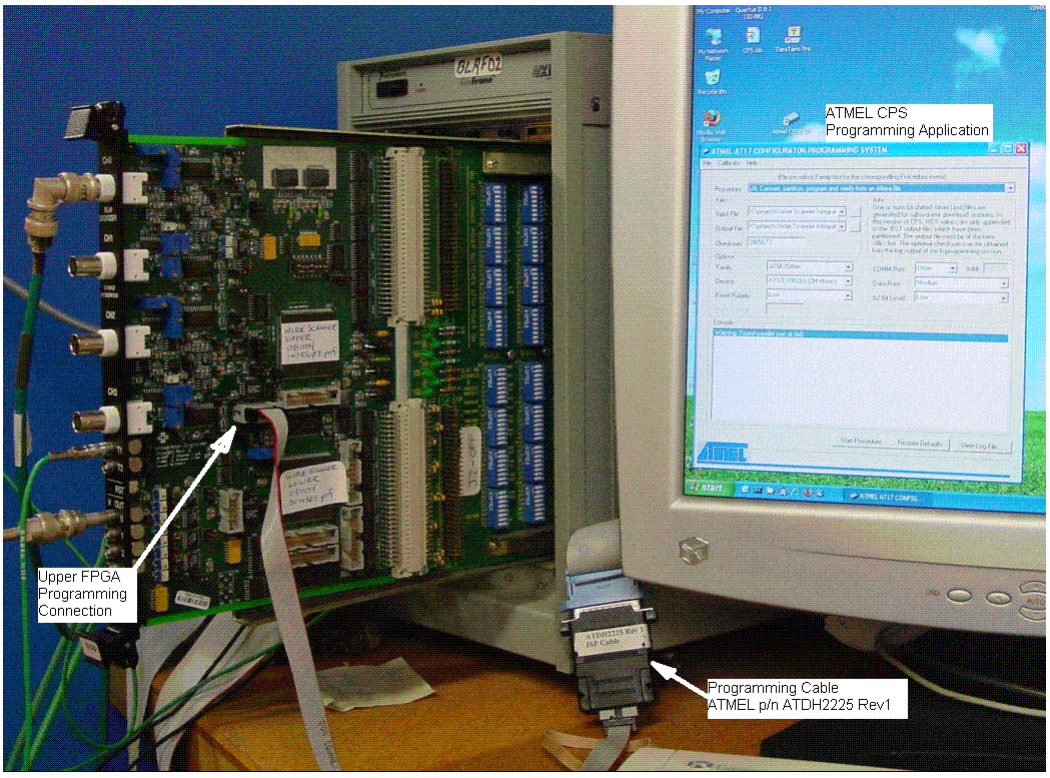


Figure III.2 Setup for programming the Upper FPGA.

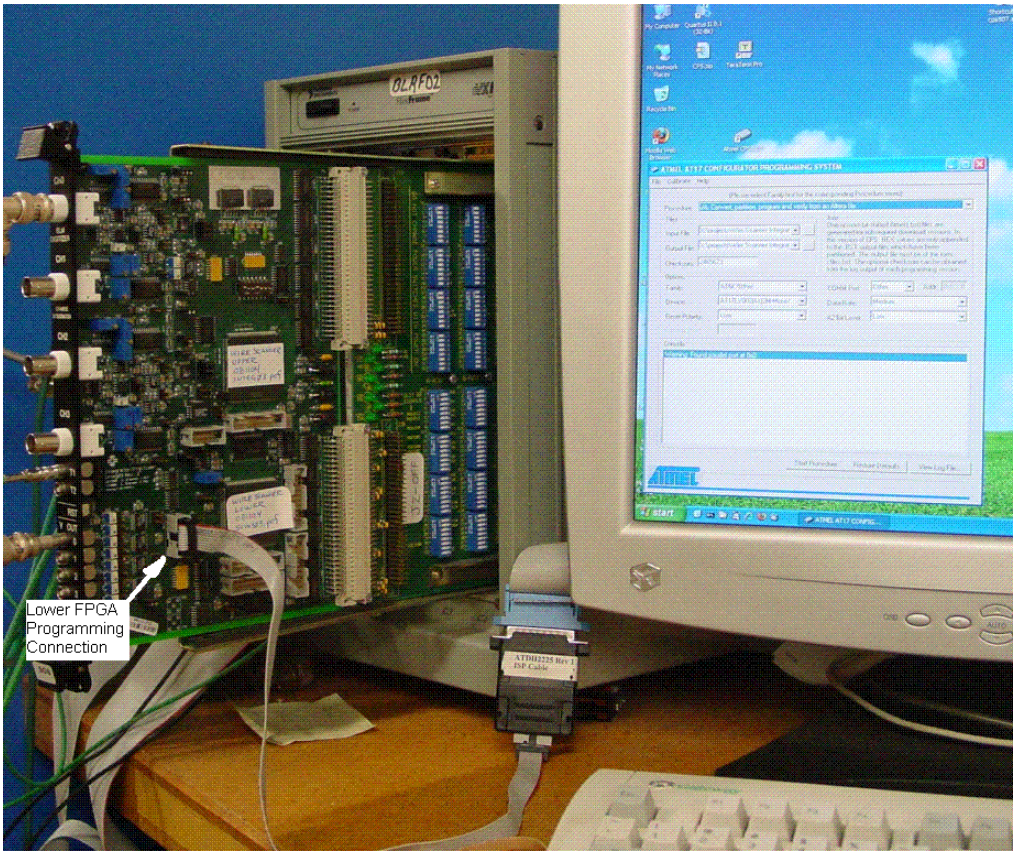


Figure III.2 Setup for programming the Lower FPGA.

Figure III.3 shows the control panel for the ATMEL CPS programming application. Note that it is required that the options box be setup first in order to be able to select the "/A:Convert, partition, program and verify an Altera file" choice in the Procedure pick box. The options are setup as shown, except for possibly the COMM Port and Addr boxes. A typical value for COMM Port would be LPT1 (a parallel port) and the Addr box would set itself. In the example in Figure III.3, the computer uses a PCI parallel port card. By choosing "Other" we are able to set Addr to the hardware address of the PCI card. ATMEL is coming out with a USB version of the programming cable.

The Input File box in the control panel should be set to point to the "POF" extension file for the FPGA project you wish to install. This is a file generated by Quartus when the FPGA Code is compiled. The Output File can be any file name ending with a ".bst" extension. This can be a file that does not yet exist. This is the file that is written when the Altera POF file is "converted" to an ATMEL format file.

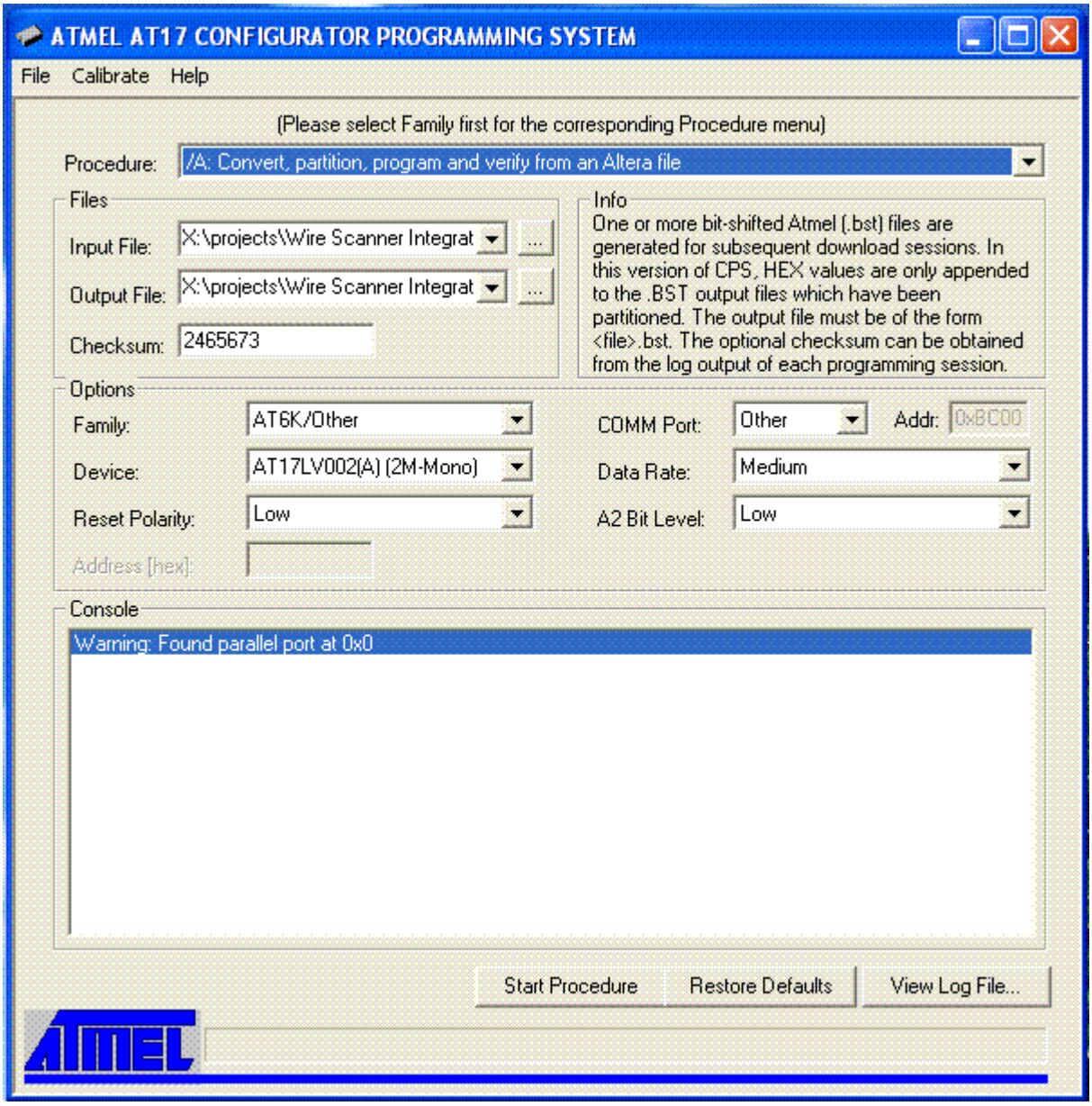


Figure III.3 The control panel for the ATMEL CPS programming application.

IV. Operational Test Setup

An operational test can be performed to ensure that the integrator module is working as intended. Figure IV.1 illustrates the test equipment connections. Basically two pulse inputs are required. The first provides a trigger to begin a measurement cycle. The second pulse, delayed in time with respect to the first pulse, represents the charge pulse from the wire scanner. The delay between the trigger pulse and the signal pulse allows for the measurement of the baseline before the active integration interval begins. Table IV. Lists the parameters of the two pulses we wish to generate.

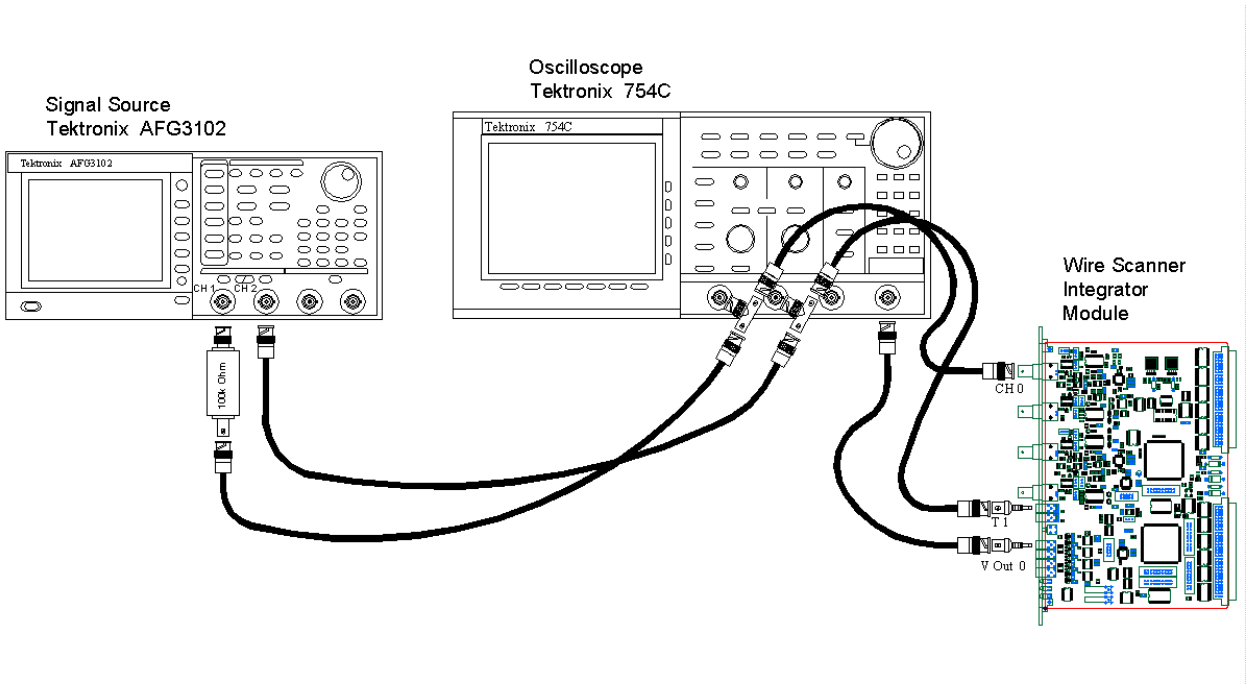


Figure IV.1 illustration of the test equipment connections.

Table IV Pulse parameters

Parameter	Value	Units
Channel 1 Setup (Trigger)		
Function	Pulse	
Frequency	1.0	Hz
Amplitude	3.00	Vpp
Offset	1.50	V
Width	10	micro-seconds
Run Mode	Burst (1 cycle)	
Channel 2 Setup (Signal)		
Function	Pulse	
Frequency	1.0	Hz
Amplitude	20 to 100	milli-Vpp
Offset	10 to 60 (baseline)	milli-Vpp
Width	220	micro-seconds
Run Mode	Burst (1 cycle)	
Delay	100	micro-seconds
Trigger (menu 2 of 2 of Burst Param Menu)		
Source	Internal	
Slope	Positive	

V. Installation and ACNET DAQ Details

Here is a collection of tables, screen shots and some simple operating instructions for the wire scanner.

Table V. ACNET devices associated with the VME Wire Scanner Integrator.

ACNET Parameter	Description	Details / Comments
B:WPNUH	Wire Position Setpoint	Motor control parameter
B:WNUH	Charge Signal Into Integrator	
B:WPSV	Integrator Voltage Output	Crate 30/Slot 20/Channel 13 MADC Num 10
B:IINTEG	Integrator “Start” Trigger	Crate 30/Slot 18/Ch 0 (front)

CHEAT SHEET FOR THE WIRE SCANNER APPLICATION

The Wire Scanner parameters can be found at

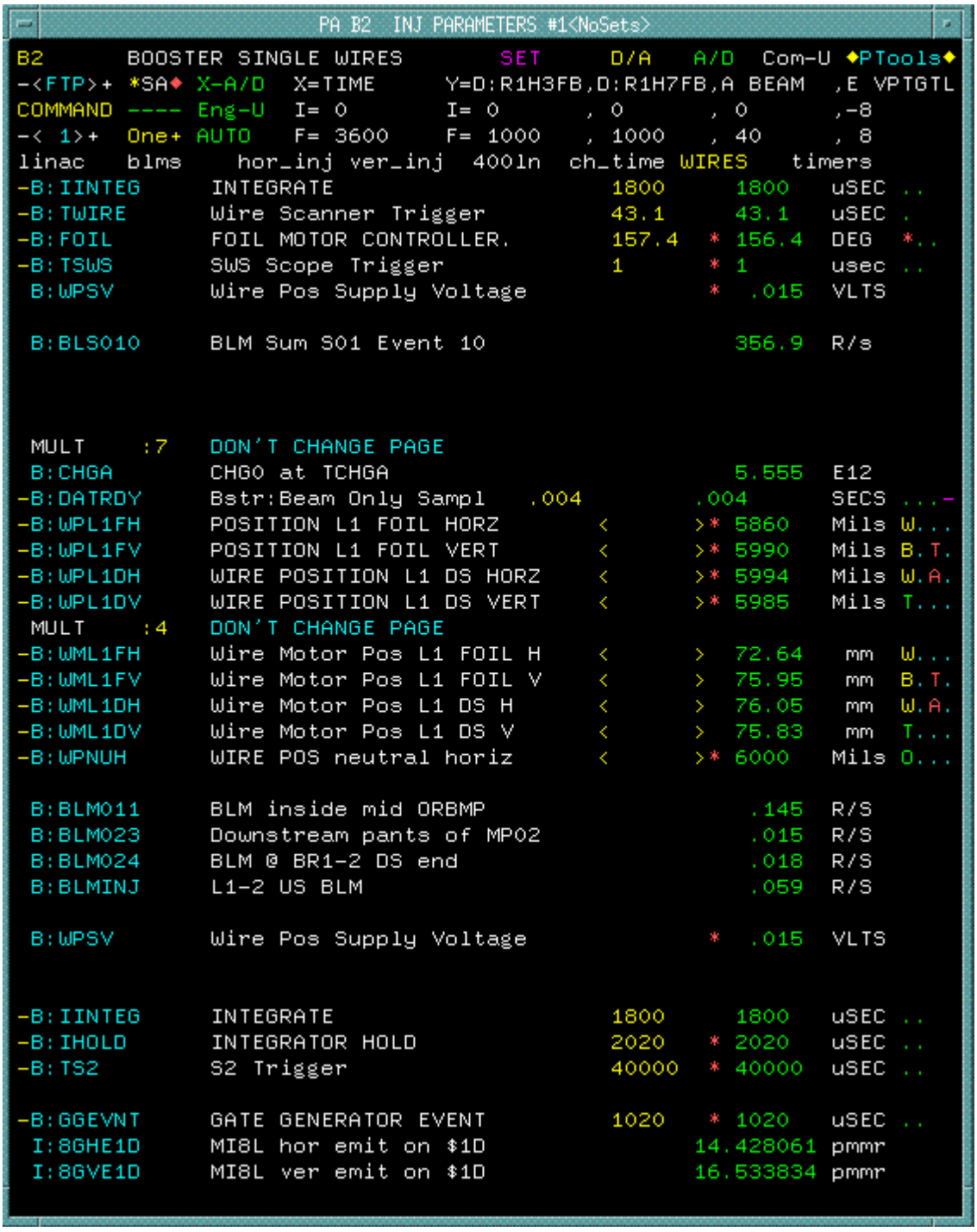
ACNET parameter page B2
page category "wires"
page -< 1>+

Wire Position
Wire position is changed with motor control parameter B:WPNUH
Rest Position = 4416 mils
H-minus Position = 4200 mils (estimate)

Higher values move wire radially **outward.**
Lower values move wire radially **inwards.**

Lower **limit** before hitting the beam = 3800 mils

Integrator Trigger
The integrator trigger is device B:IINTEG
The trigger delay from event = 1800 microseconds
The trigger event = 19 (NUMI) or 1D (MiniBooNe)


PA B2 INJ PARAMETERS #1<NoSets>
B2 BOOSTER SINGLE WIRES SET D/A A/D Com-U ♦PTools♦
-<FTP>+ *SA♦ X-A/D X=TIME Y=D:R1H3FB,D:R1H7FB,A BEAM ,E VPTGTL
COMMAND ---- Eng-U I= 0 I= 0 , 0 , 0 ,-8
-< 1>+ One+ AUTO F= 3600 F= 1000 , 1000 , 40 , 8
linac blms hor_inj ver_inj 400ln ch_time WIRES timers
-B:IINTEG INTEGRATE 1800 1800 uSEC ..
-B:TWIRE Wire Scanner Trigger 43.1 43.1 uSEC .
-B:FOIL FOIL MOTOR CONTROLLER. 157.4 * 156.4 DEG *..
-B:TSWS SWS Scope Trigger 1 * 1 usec ..
B:WPSV Wire Pos Supply Voltage * .015 VLTS
B:BLS010 BLM Sum S01 Event 10 356.9 R/s
MULT :7 DON'T CHANGE PAGE
B:CHGA CHG0 at TCHGA 5.555 E12
-B:DATRDY Bstr:Beam Only Sampl .004 .004 SECS ...-
-B:WPL1FH POSITION L1 FOIL HORZ < >* 5860 Mils W...
-B:WPL1FV POSITION L1 FOIL VERT < >* 5990 Mils B.T.
-B:WPL1DH WIRE POSITION L1 DS HORZ < >* 5994 Mils W.A.
-B:WPL1DV WIRE POSITION L1 DS VERT < >* 5985 Mils T...
MULT :4 DON'T CHANGE PAGE
-B:WML1FH Wire Motor Pos L1 FOIL H < > 72.64 mm W...
-B:WML1FV Wire Motor Pos L1 FOIL V < > 75.95 mm B.T.
-B:WML1DH Wire Motor Pos L1 DS H < > 76.05 mm W.A.
-B:WML1DV Wire Motor Pos L1 DS V < > 75.83 mm T...
-B:WPNUH WIRE POS neutral horiz < >* 6000 Mils O...
B:BLM011 BLM inside mid ORBMP .145 R/S
B:BLM023 Downstream pants of MP02 .015 R/S
B:BLM024 BLM @ BR1-2 DS end .018 R/S
B:BLMINJ L1-2 US BLM .059 R/S
B:WPSV Wire Pos Supply Voltage * .015 VLTS
-B:IINTEG INTEGRATE 1800 1800 uSEC ..
-B:IHOLD INTEGRATOR HOLD 2020 * 2020 uSEC ..
-B:TS2 S2 Trigger 40000 * 40000 uSEC ..
-B:GGEVNT GATE GENERATOR EVENT 1020 * 1020 uSEC ..
I:8GHE1D MI8L hor emit on $1D 14.428061 pmmr
I:8GVE1D MI8L ver emit on $1D 16.533834 pmmr