Booster VME-DDS Module Bench Tests

September 27, 2017

Craig Drennan

Table of Contents

Table of Figures and Tables

I. Introduction

This document is a procedure for checking out the VME-DDS Modules, drawing # 173718, 9/2009. The base setup for the VME-DDS Module bench tests include a collection of test equipment, FPGA programming tools, a serial terminal to communicate with the MVME crate controller. We use software that runs from the "nova.fnal.gov" server to write and use commands to interface to the VME-DDS module through the MVME crate controller.

II. Programming the FPGA's

The Upper and the Lower FPGA's are both programmed from a single EEPROM configuration memory. The configuration memory is loaded with a single Altera **.pof* file using the Quartus II programming application and a USB Blaster.

The two FPGA's are configured sequentially from the serial EEPROM, first the Upper FPGA and then the Lower. When the individual FPGA configurations are compiled by the Quartus II program, ****.sof*** files are generated. The current procedure is to copy a newly generated "DDS_LOWER.sof" file from the Lower FPGA project directory to the Upper FPGA project directory. Within Quartus, with the Upper FPGA project active, use the "Convert Programming Files" utility, under the "File" pull down menu. Within the utility, we press the "Open Conversion Setup Data" button and load the file "UPPER-LOWER convert file.cof". The UPPER_LOWER convert file was previously setup to append the "DDS_LOWER.sof" file to the "DDS_UPPER.sof" file to create the "UPPER-LOWER.pof" file, with which we will program the configuration memory. Figure II.1 is a screen shot of what this utility looks like after the conversion setup file has been loaded. The "Generate" button is clicked to execute the conversion.

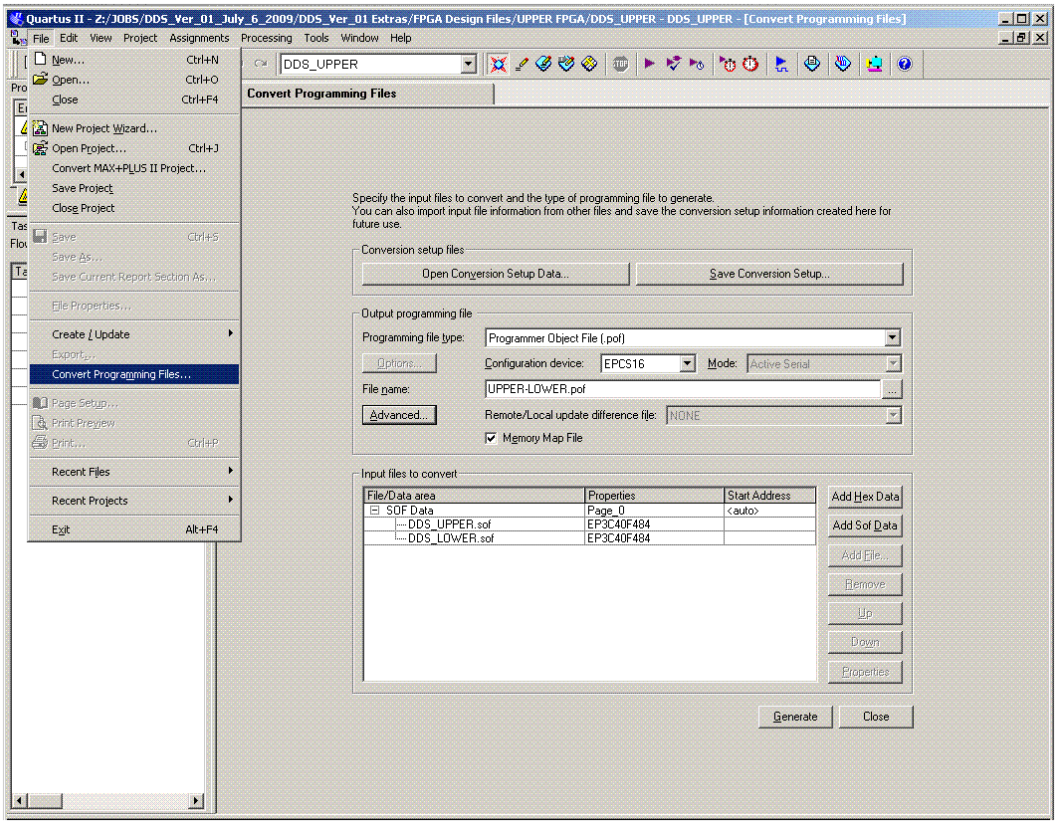

Figure II.1 Screen shot of the programming file conversion utility.

Once the “UPPER-LOWER.pof” file has been generated, or re-generated, we use the configuration device programmer utility within Quartus with an Altera USB-Blaster Cable. To program the EPCS16SI8N device on the DDS module the USB-Blaster, or similar programming cable, is connected to J22. The programmer connection is shown in Figure II.2.

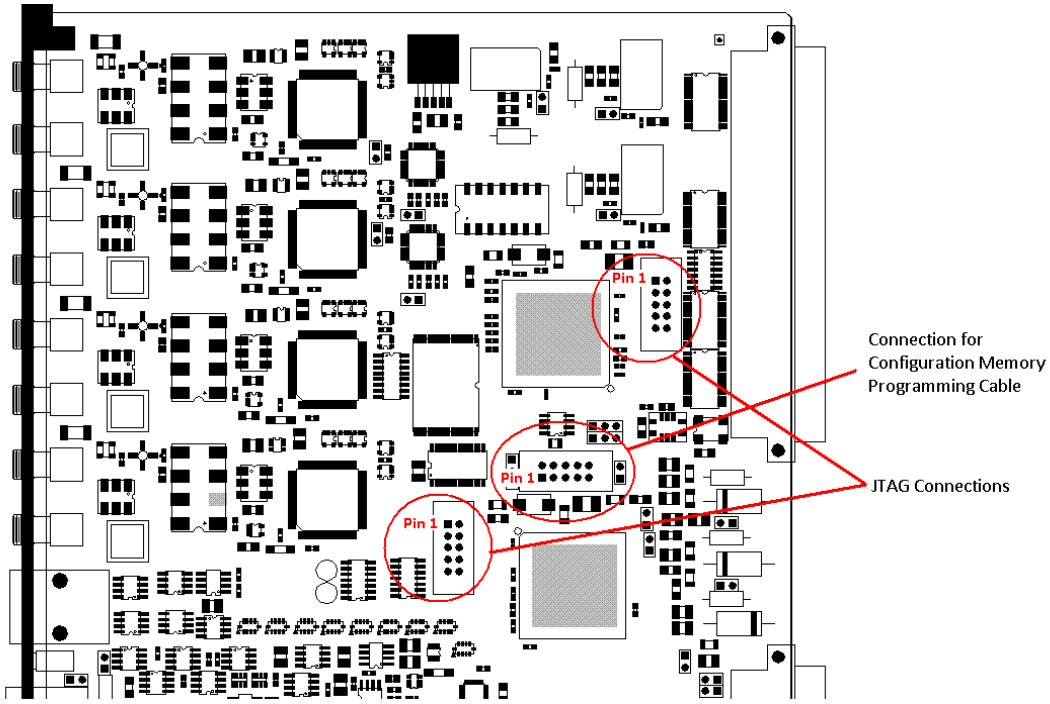

Figure II.2 FPGA programmer connections and JTAG interface connection.

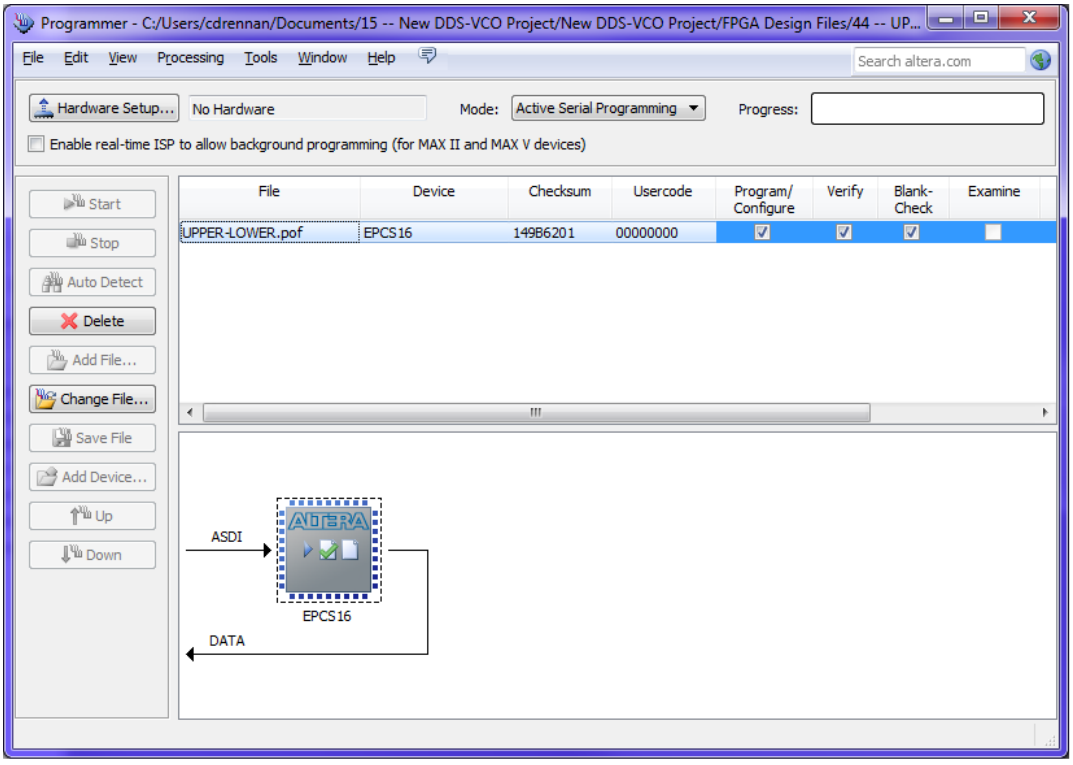

Figure II.3 The Quartus II Programmer utility screenshot.

In the Quartus II programmer utility, Figure II.3, note the Checksum value to uniquely identify the code being loaded. The Examine function can be used to read the data from the EEPROM configuration device and identify the Checksum of the code loaded.

III. Interfacing Through the MVME Crate Processor

We read and write parameters and curves and execute functions on the VME-DDS module via serial port commands to the MVME crate processor. The MVME communicates with the VME-DDS module over the VME bus. When the MVME crate processor is powered up, or reset, it must be setup and loaded with software routines from the "nova.fnal.gov" server. The MVME 2300 processor, on the test bench in room BGW-104, is currently node BLRFD3. We must ensure that the startup script for this node is selected to be "mini_startup" as seen in file

"/fecode-bd/vxworks_boot/fe/blrfd3/startup" on nova.fnal.gov.

A particular startup script is enabled by removing the '#' sign denoting comment text, if it is not already. A '#' comment symbol should be placed at the start of any other line with a script command. See a listing of the current startup file in Appendix A.

```
#CURRENT VME STARTUP
< mini_startup
```

Among other startup functions, the MVME is loaded with software functions specifically written to interact with the VME-DDS module using a serial terminal connected to the DEBUG port of the MVME. These software functions are compiled into the file and copied to

"/fecode-bd/vxworks_boot/fe/blrfd3/vme_test1.o"

The source directory for the C files and Make file for these software functions is

"/export/home/cdrennan/code/dds_vme1/"

Use Teraterm or PuTTY, or a similar serial terminal program, to connect to the MVME debug serial interface. The serial port settings for the MVME 2300 are: 9600 BAUD, 8 Bit, 1 Stop Bit, no parity, no flow control.

In the future, the crate processor may be upgraded to an MVME 5500 whose serial port settings are: 38400 BAUD, 8 Bit, 1 Stop Bit, no parity, XON/XOFF flow control.

To set or read the curves in the VME-DDS Module, you need access to the ACNET, JAVA application "B30: Booster DDS Curves". Instructions on using this application can be found in the Accelerator Division DocDB database at Beams-doc-5307.

III.1 Serial Terminal Commands Used in Testing the VME-DDS Module.

Table III.1 lists several serial terminal commands useful in testing the VME-DDS module.

Table III.1 Serial terminal commands.

Command	Arguments	Results
params	none	Displays a table of 14 parameters used by the module. See Figure V.2.
rdparam(uint index)	index => an unsigned integer selecting parameter 0 - 14	Displays the hexadecimal value of the Setting, Reading and Flag for a single parameter.
wrparam(uint index, uint value)	index => an unsigned integer selecting parameter 0 – 14 value => a number between 0 and 65535.	This writes a value to the indexed parameter's Setting and sets the update Flag for this parameter. A brief pause is implemented before reading back the Reading and Flag. These will have been updated if the module is receiving a 15 Hz trigger.
ena_store(void)	none	This command sets the enable backup bit allowing a store to FLASH when a parameter or curve is changed. The enable backup bit is cleared when a backup is made, requiring the ena_store be re-issued for another backup to occur.
restore(void)	none	VME Control Mode Only. Causes shared memory to be loaded from the FLASH memory.
store(void)	none	VME Control Mode Only. Causes shared memory to be written to the FLASH memory.
upd(void)	none	VME Control Mode Only. Triggers the parameter update process.
setdacs(int volts)	nominal volts out of all 4 dacs (-10 … +10)	With SW[6] and SW[7] in the down position this command nominally sets all the dac outputs to the parameter "volts".
caltest("filename")	Filename => name of file to store calibration results.	This routine sequences the ADC voltage test source through voltages from −10V to +10V asking for a measurement of each voltage from the user and storing the info to a file.
testchan(int channel)	channel => a value from 1 to 4 indicating the ADC channel to test.	This function sequences through the voltages for which the test source has been recently calibrated and makes comparisons of the resulting ADC code output to the ideal codes for the voltages set.

IV. Test Equipment and Module Setup

The base setup for the VME-DDS Module bench tests include a collection of test equipment. The following describes what is needed.

1. The VME-DDS Module under test in a VME crate with an MVME, Slot 0 processor. It is preferred that the VME-DDS module is on an extender card so that it is easier to access the DIP switches.

<u>NOTE !</u> The VME-DDS module always needs forced air cooling from the crate fans or an individual bench type fan.

2. An oscilloscope, 4 channel, 100 MHz bandwidth or better with A then B type triggering. The setup used when writing these instructions used a Tektronix MSO3054, 4 channel, 500 MHz scope.

3. Dual channel signal generator like a Tektronix AFG3102 or and Agilent 81150A. These can generate sinewaves up to 100 MHz, and pulse type signals whose period and pulse width can be adjusted.

4. A phase detector with a bandwidth of 800 kHz or better. An MI Phase-Lock Controller has a phase detector with a direct analog output that meets this requirement, and is used in these instructions.

5. The Figure VI.1 shows the DIP switch settings and the initial configuration for testing.

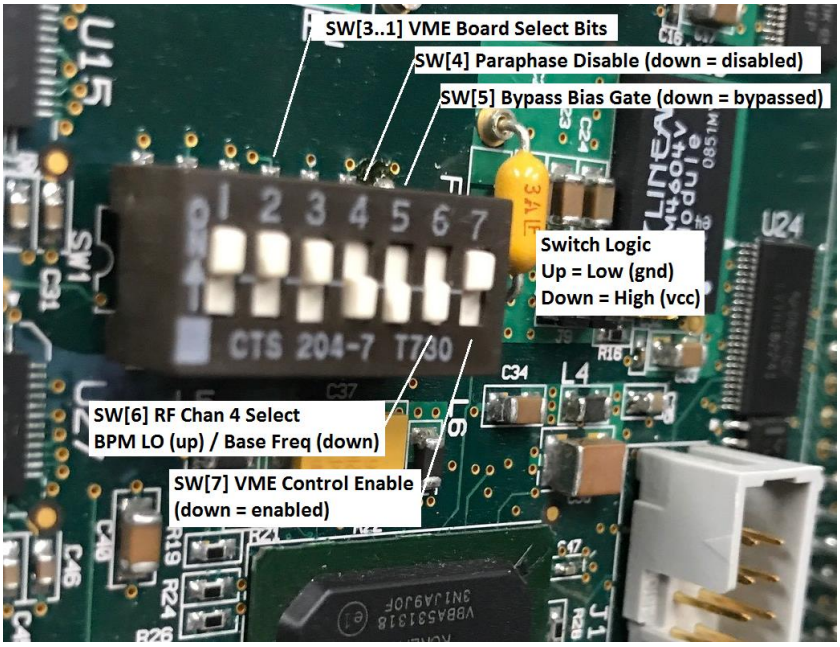

Figure IV.1 DIP Switches for addressing and setting test modes.

6. An adapter is typically needed for testing on the bench. This adapts the digital inputs and outputs on the front panel ribbon connector to input Lemo connectors and a pin header for monitoring the outputs on a logic analyzer probe.

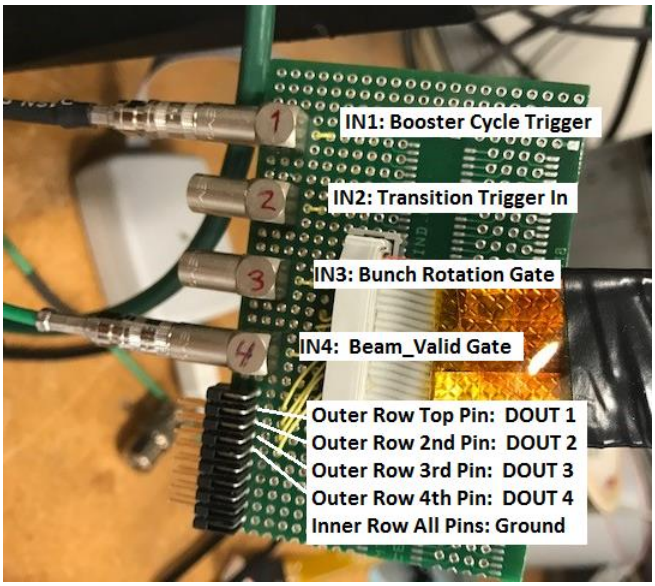

Figure IV.2 Digital IO Adapter.

V. Setting Up FLASH Memory for a New Module.

The ACNET control system reads and writes parameters and curves to the VME-DDS module through the MVME processor, over the VME bus. Under normal operating conditions, the VME bus reads and writes only exchange data with the dual-ported "shared memory" on the VME-DDS module. Processes running in the FPGA's of the VME-DDS access this shared memory to process the data and fill in all the other memories used on the module. It is, therefore, only necessary to back up the shared memory to the FLASH memory.

The VME-DDS uses two blocks of memory in the FLASH to store the shared memory data. When shared memory is being backed up it is written to the memory block that was not used to bootup the module. When the shared memory backup is complete, it is tagged as the block to be booted from and the other block of memory will be the one we back up to next time. This way we avoid corrupting our boot up data should the data backup be interrupted by a loss of power.

When the VME-DDS module is powered up several bootup processes run. The first process will load parameter and curve data from the FLASH memory into the working FPGA memory. If a module is new, or something has corrupted the FLASH memory, it is necessary to store good values into the FLASH. The FPGA's have internal RAM memory for which initial memory values can be set. This is done with the **.mif* memory initialization files. These files are associated with the blocks of memory created with the Quartus II MegaWizard. At power up, the FPGA memory will assume these initial memory values.

As seen in the previous section, DIP switch SW[7] is the VME Control Enable. If the module is powered up with SW[7] in the downward position, the normal bootup processes do not run and we can execute serial port commands through the MVME processor to manually execute bootup commands. By executing the "store" command, we store the FPGA initial memory values, for the shared memory, into the FLASH memory. Executing the "store" command twice ensures there are valid values in both FLASH memory blocks.

Once the FLASH memory has been setup, we can test the normal functioning of the backup and boot up processes. The VME-DDS module needs to be triggered to cause the FPGA logic to process new parameter values. The normal Booster trigger happens at 15 Hz. Figure V.1 illustrates the connections and signal generator setup to get this trigger.

V.1 Procedure for Setting Parameters and Testing the FLASH Memory Process

1. Connect and start the cycle trigger as illustrated above and power the module.

2. Execute the serial terminal command “params” and note the value of the parameters.

3. If there are known values you wish to set the parameters to, you can start setting a few of these using the command wrparam(index, value) (e.g. wrparam(2, 0xea6)). The value can be given in either decimal or hexadecimal, but the wrparam command does not convert from engineering units. Figure V.2 is a screen capture with reasonable values for the parameters.
   Note that when you are writing the parameters the left LED on the faceplate labeled ‘X’ does not light.

4. Execute the serial terminal command “ena_store” watching the ‘X’ LED as you hit enter. You should see the LED light up and then go out.

5. Now you can power cycle the board.

6. Once the serial terminal shows that the MVME processor has finished booting up, you can execute the “params” command again and see that the parameter changes you had made persisted after the power cycle.

7. Continue to setup all the parameters to the desired value and execute the “ena_store” command.

8. Power cycle the module and ensure the parameters have retained the values that you set.

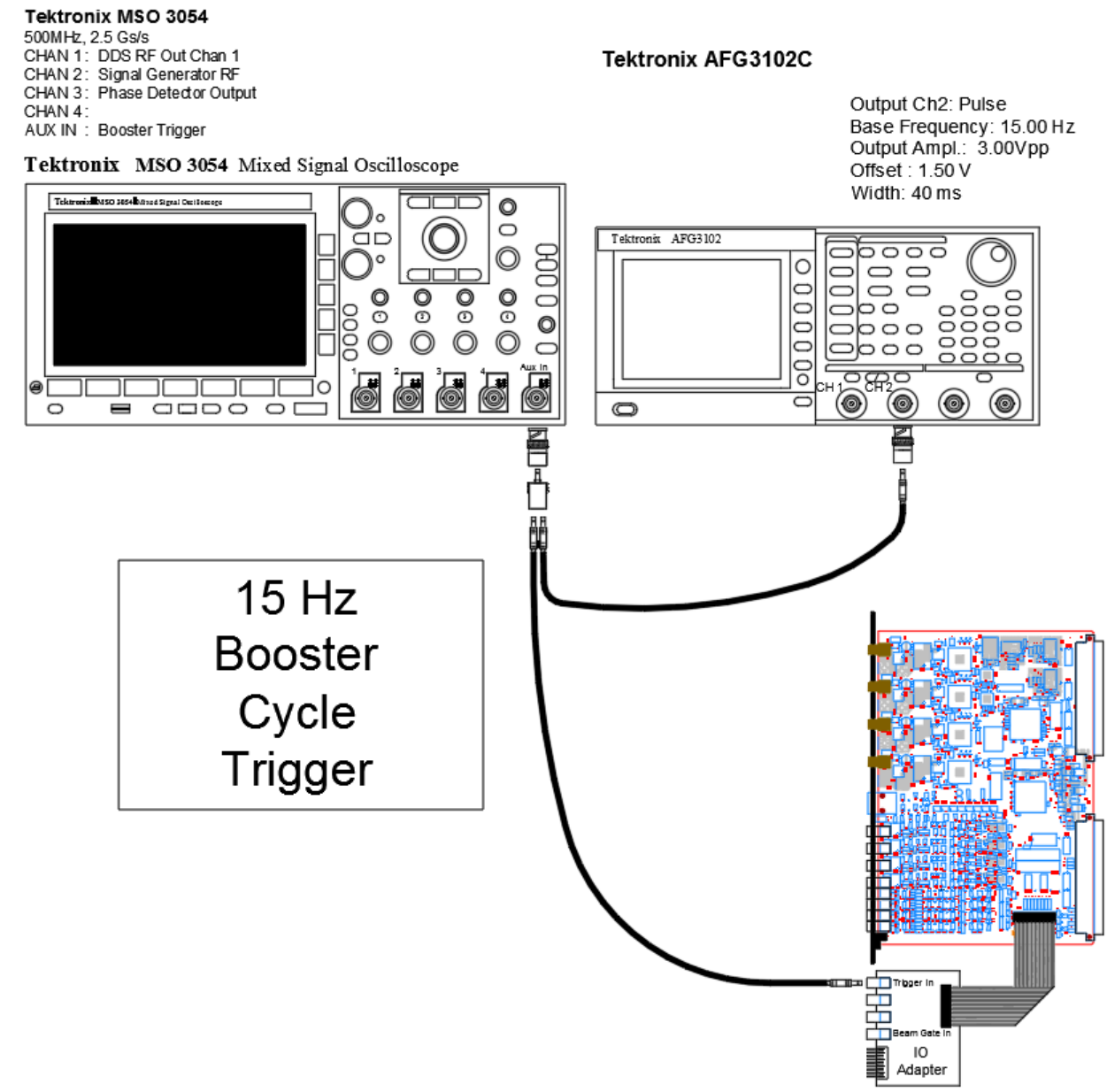

Figure V.1 15 Hz Trigger connection.

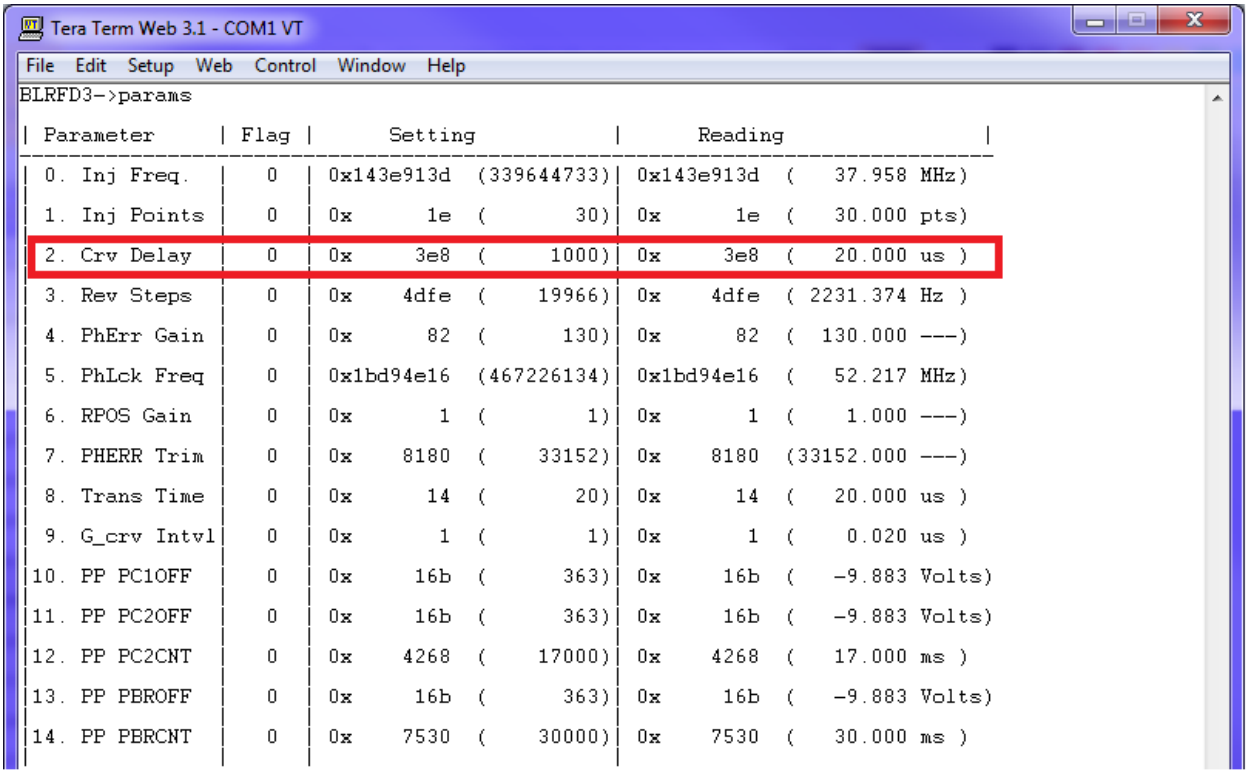

Figure V.2 Good parameter values.

V.2 Setting up Frequency and Bias Curves

To set or read the curves in the VME-DDS Module, you need access to the ACNET, JAVA application “B30: Booster DDS Curves”. Instructions on using this application can be found in the Accelerator Division DocDB database at Beams-doc-5307.

The basic procedure for loading good curves is as follows.

1. Start and ACNET console and run application B30.

2. When started, the application is as in the left pane of Figure V.3. The left-most pull-down menu will indicate that we are targeting the VME DDS module in the LLRF Room, running the Booster (so be careful).

3. Select the second pull-down menu as shown and select “Freq”. The application will then retrieve the Frequency curve in BLLRF.

   The curve you see in the application window is in a local buffer. If you inadvertently change something it will not affect the curve in the running “bllrf” module unless you click “Load” with “bllrf” selected in the pull-down menu

4. As shown in the right pane of Figure V.3, select “blrfd3”, then load the Frequency Curve retrieved from “bllrf” into “blrfd3” by clicking the button near the bottom of the application window labeled “Load”.

The Frequency Curve is now loaded into “blrfd3”. Later we will see ways to verify this.

5. Select “bllrf” again from the pull-down menu.

6. Select “bias” from the second pull-down menu. The Bias Curve from the bllrf module will be displayed in the application window.

7. Select “blrfd3” from the first pull-down menu.

8. Click “Load” to load this Bias Curve into blrfd3.

9. Execute the serial terminal command “ena_store” and watch that the ‘X’ LED on the front of the VME DDS module lights. The curve data should now be stored in the FLASH memory.

10. After power cycling the VME DDS module you should be able to verify that the Frequency and Bias curves are established in the module. On the Curve Editor application select “blrfd3” in the left-most pull-down menu. When you select “Freq” from the second pull-down the Frequency Curve in blrfd3 will appear in the application window. Examine the curve to see if it is as expected.

11. With “blrfd3” selected, now select “Bias” from the second pull down and the Bias Curve from blrfd3 should appear. Examine this to see if it is as expected.

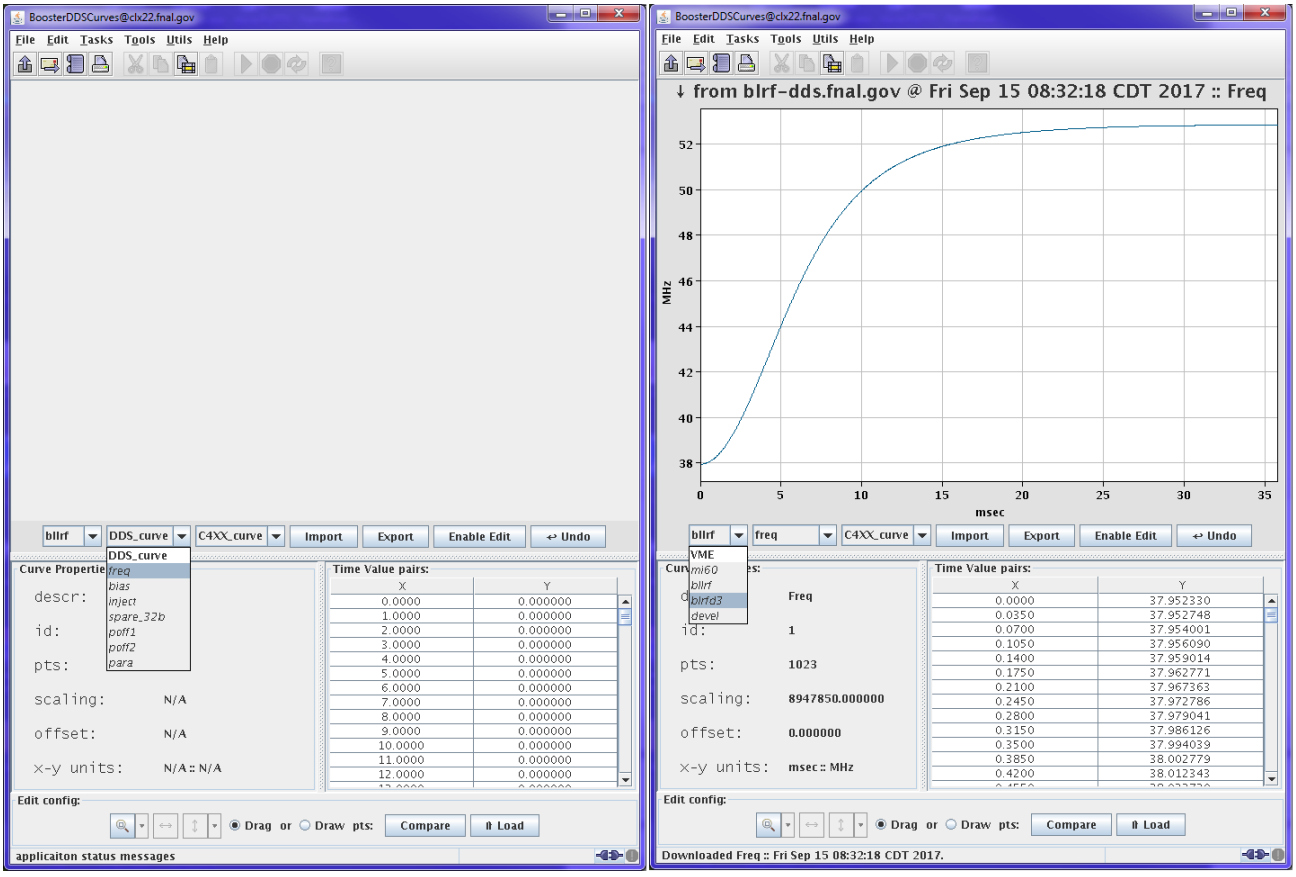

Figure V.3 Curve Editor Application B30.

VI. Diagnostic DAC Outputs

Four analog outputs are produced by Digital to Analog Converters (DAC's) controlled by the FPGA's. The analog outputs are assigned as follows.

AO 1 is the base frequency curve, with the modified injection curve portion, but without the phase error feedback.

AO 2 is the Bias Curve. This is the bias curve that is amplified and transmitted to the High Level RF systems.

AO 3 is the base frequency curve, with the modified injection curve portion and phase error feedback.

AO 4 is a software selectable diagnostics output.

Use the following procedure to calibrate the DAC outputs.

Note: The input resistors between the final op-amp and the x4 range selection jumpers { R131, R152, R171, R191} have been decreased from 442 ohms to 424 ohms by adding approximately 10.2K ohms in parallel. This increased the gain of each channel so the desired gain was within the range of the dpot gain adjustment.

1. Connect the DAC output to be calibrated to a voltmeter.

2. Set DIP switches SW[6] and SW[7] in the down position. If it is necessary to power cycle the module it is best to return SW[7] to the up position when the module is powering up, booting up. Then return SW[7] to the down position before proceeding with the DAC calibration.

3. To adjust the gains of the DAC channels we use the digital potentiometers on the module. We adjust the dpots with the "dpot" command shown in Figure VI.1.

4. First setup the channel gain by determining the difference between the DAC voltage out when nominally set to +5.00V and -5.00V. Use the commands setdacs(5.0) and setdacs(-5.0). The difference between the two DAC output voltages should be 10.000, or as close as you can get. It will typically take several iterations between using the "dpot" command and checking the gain with the "setdacs" command. It is helpful to keep a log of the voltages measured and the digital pot

5. After the gain for a channel has been set, set the DAC output to 0.00V, setdacs(0.0), and use the dpot command to set the offset for the channel.

```
BLRFD3->dpot

------------------------------
  dpot( index, dir, steps, store )

  index:
        0 => ADC 1 Offset
        1 => ADC 1 Gain
        2 => ADC 2 Offset
        3 => ADC 2 Gain
        4 => ADC 3 Offset
        5 => ADC 3 Gain
        6 => ADC 4 Offset
        7 => ADC 4 Gain

        8 => DAC 1 Offset
        9 => DAC 1 Gain
       10 => DAC 2 Offset
       11 => DAC 2 Gain
       12 => DAC 3 Offset
       13 => DAC 3 Gain
       14 => DAC 4 Offset
       15 => DAC 4 Gain

       16 => Test Source Adjust
--------------------------------
  dir:  direction when sdjusting error
        from Expected minus Result code
     -- Offset --
      '1' -> less positive (down)
      '0' -> more positive (up)

     --  ADC Gain (slope) --
      '1' -> slope more positive
      '0' -> slope more negative

     --  DAC Gain (slope) --
      '0' -> slope more positive
      '1' -> slope more negative
--------------------------------
  steps: 64 possible steps.
         Holds when limits reached.

--------------------------------
  store:  store change in non-volatile memory
      '1' -> make permanent
      '0' -> make temporary

--------------------------------
```

Figure VI.1 The command that manipulates the digital potentiometers.

VII. Bias Curve and Bias Gate Operation

Ensure the following conditions to check the operation of the Bias Curve and Bias Gate

1. Ensure that a 15 Hz trigger and the other connections are as shown in Figure VII.1.

2. Ensure the Bias Curve is loaded as described in Section V.2.

The Bias Curve is the output of DAC channel 2. Connect this to the scope with a high impedance input. The Bias Gate input is on a digital input on the rear VME module. This is expected to be between 3V and 4V when gating on the Bias Curve. The Bias Gate is bypassed, meaning the Bias Curve is always enabled, if DIP switch SW[5] is in the down position.

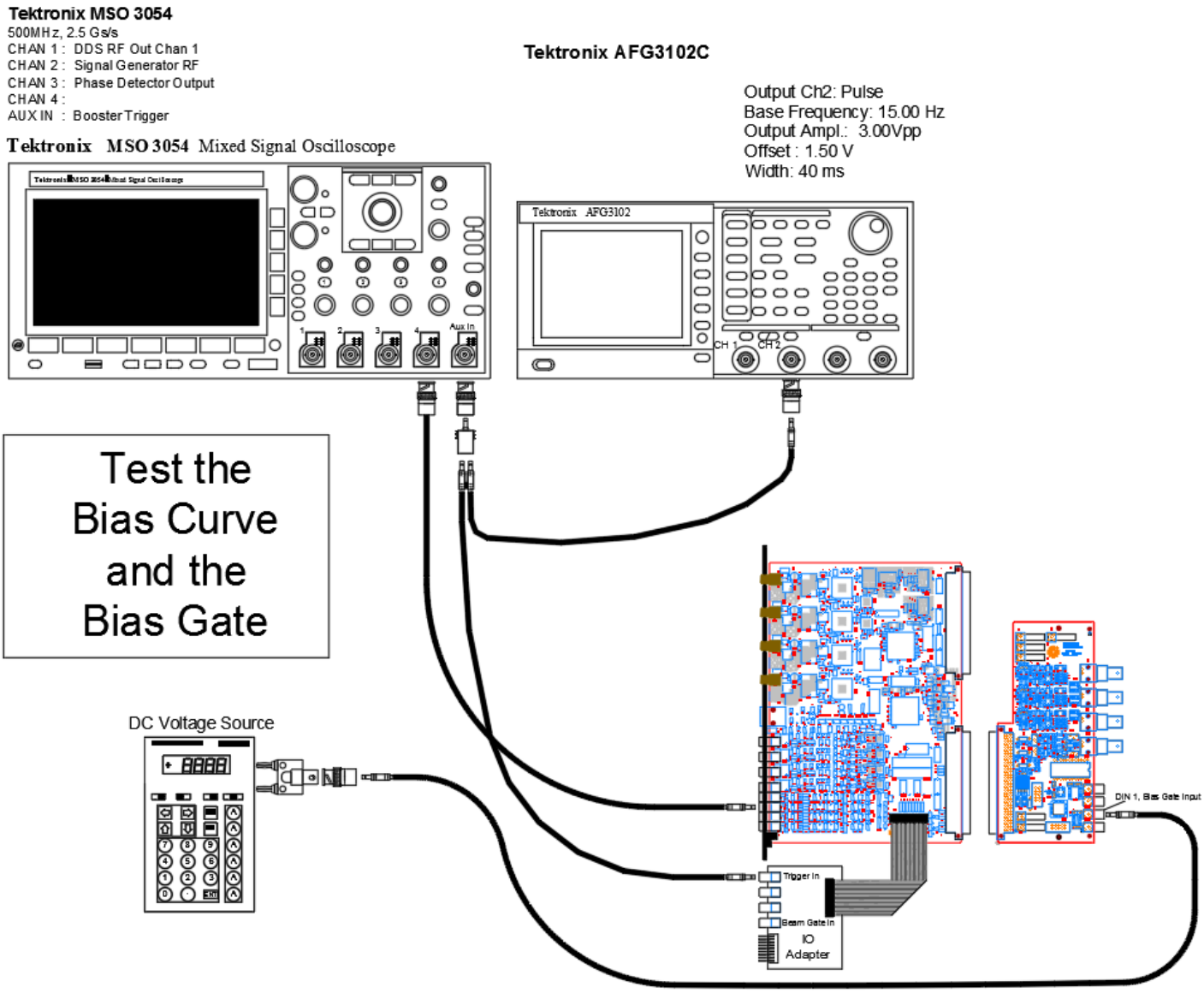

Figure VII.1 The setup for the Bias Curve and Bias Gate Test.

3. With DIP SW[5] down we should see the Bias Curve output regardless of the Bias Gate.

4. With DIP SW[5] up we should see the Bias Curve output only when the Bias Gate voltage is a logic High (e.g. 3V).

VIII. Analog Input, ADC Calibration

There are four analog inputs to the module. These inputs are assigned as follows.

IN 1 is the phase error feedback that, when the Beam Valid Gate is present, modulates the frequency of all the DDS outputs (except DDS channel 4 when SW[6] is down).

IN 2 is the RPOS phase drive feedback. This phase drive feedback will modulate the phase of DDS channels 2 and 3. This modulation is always active.

IN 3 is not currently used.

IN 4 is not currently used.

Use the following procedure to calibrate the ADC inputs.

Note: it is best if you let the VME-DDS module run and warm up for at least 30 minutes before doing this calibration. Also, there should be a typical amount of forced air cooling on the VME-DDS module.

1. With the SW[7] switch in the up position, power up the VME-DDS module and crate processor.

2. Once the processor has finished booting up, as seen with the debug serial terminal, switch SW[6] and SW[7] to the down position.

3. Monitor J32 with a voltmeter. See Figure VIII.1.

4. From the serial terminal enter "caltest("filename")", where "filename" is a short name of the file in which you want to keep the test voltage values. Note that the command uses quotes around the filename (e.g. caltest ("tf")).

5. After entering the caltest command you will be asked to enter the Test Voltage value as measured on the voltmeter. The program will sequence through 7 negative values, a pause, and then 7 positive values.

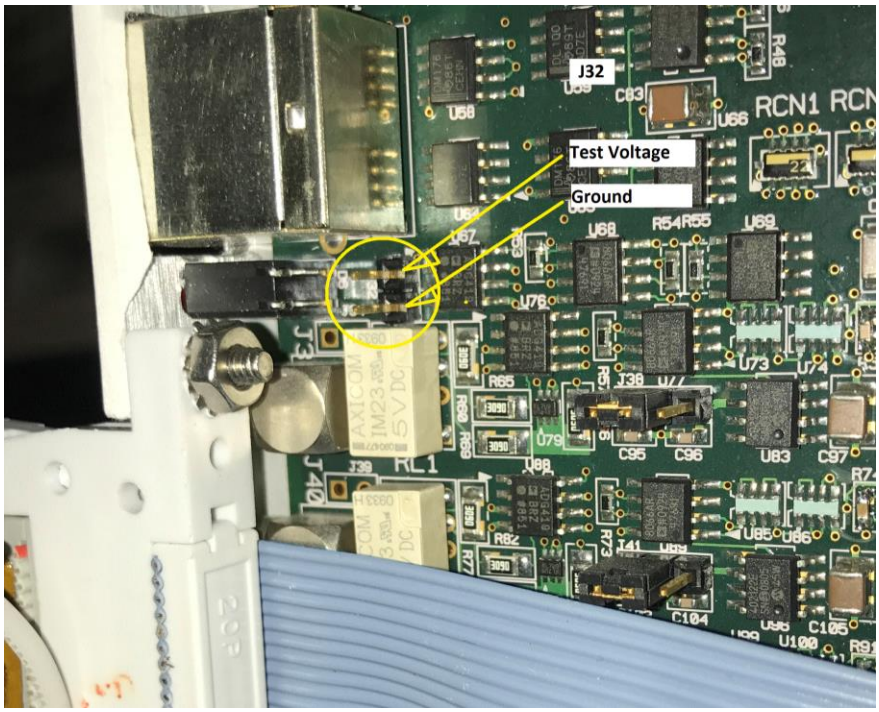

Figure VIII.1 Location of Test Voltage test point, J32.

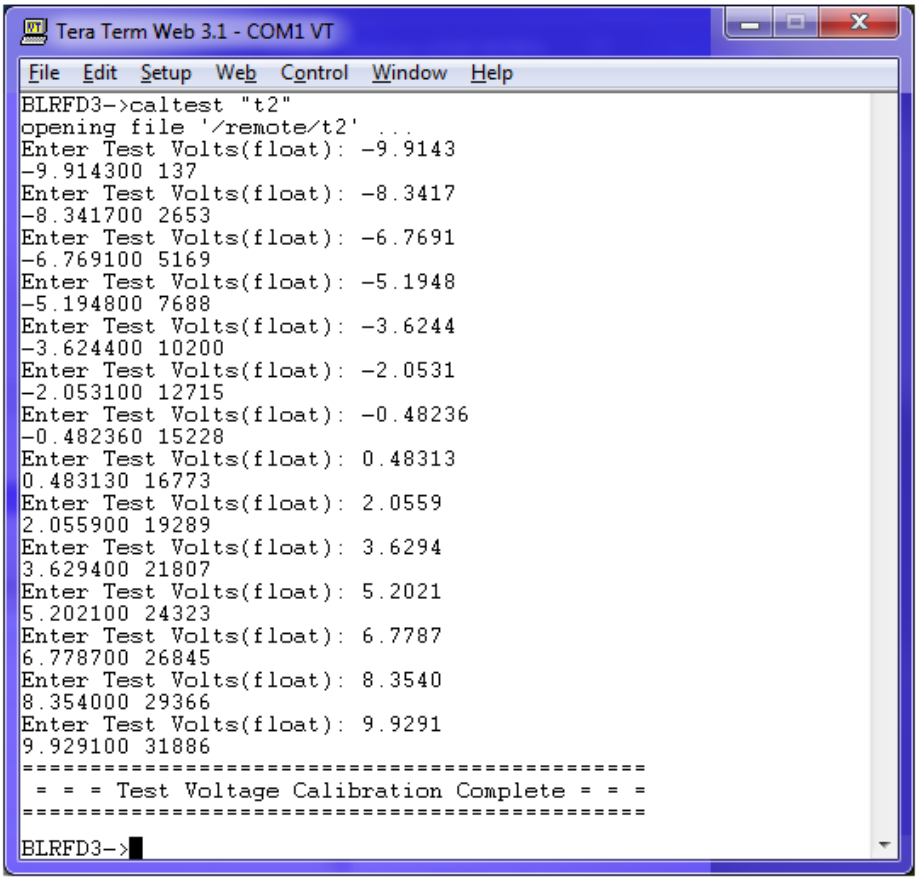

Figure VIII.2 "caltest" screen shot.

6. For each channel, we will enter the serial terminal command "testchan X" where "X" is the channel number 1, 2, 3 or 4. See Figure VIII.3.
   Note that there is a longish pause between the negative test voltage near zero and the first positive value.

7. The testchan results will show the average and standard deviation of many ADC samples at each of the test voltages. The digital pot function, Figure VI.1, can be used to minimize the numbers in the "delta" column. Each column, except for volts, is given in ADC counts. The resolution of the analog inputs with a +/-10 Volt range is 0.6 mV / count.

Note: You will see that there is a larger error (delta) near the zero volt input. There is a tradeoff between the size of this error and the size of the standard deviation result. The tradeoff is made by changing the size of the two capacitors on the input to the ADC's.

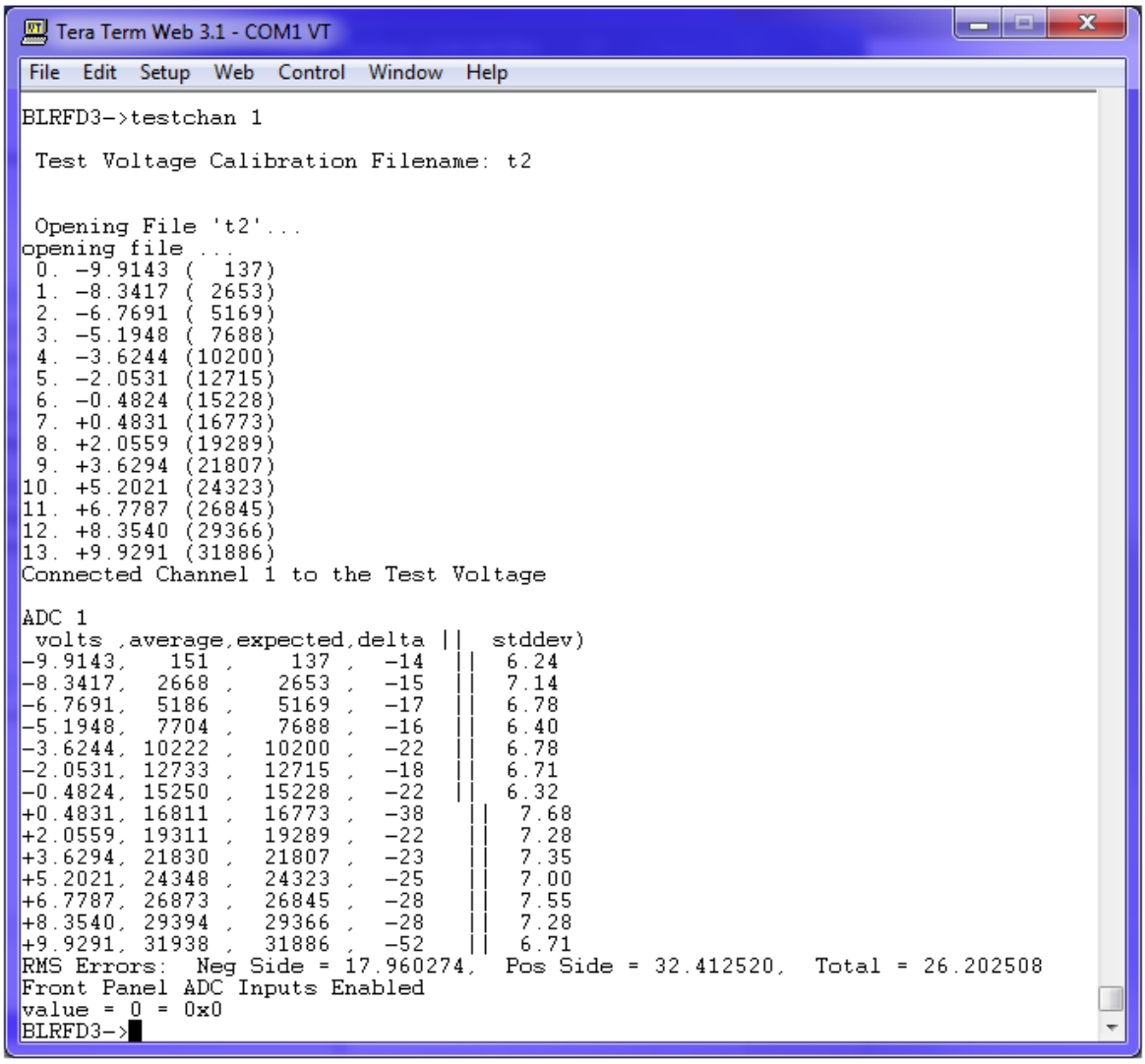

Figure VIII.3 “testchan” screen shot.

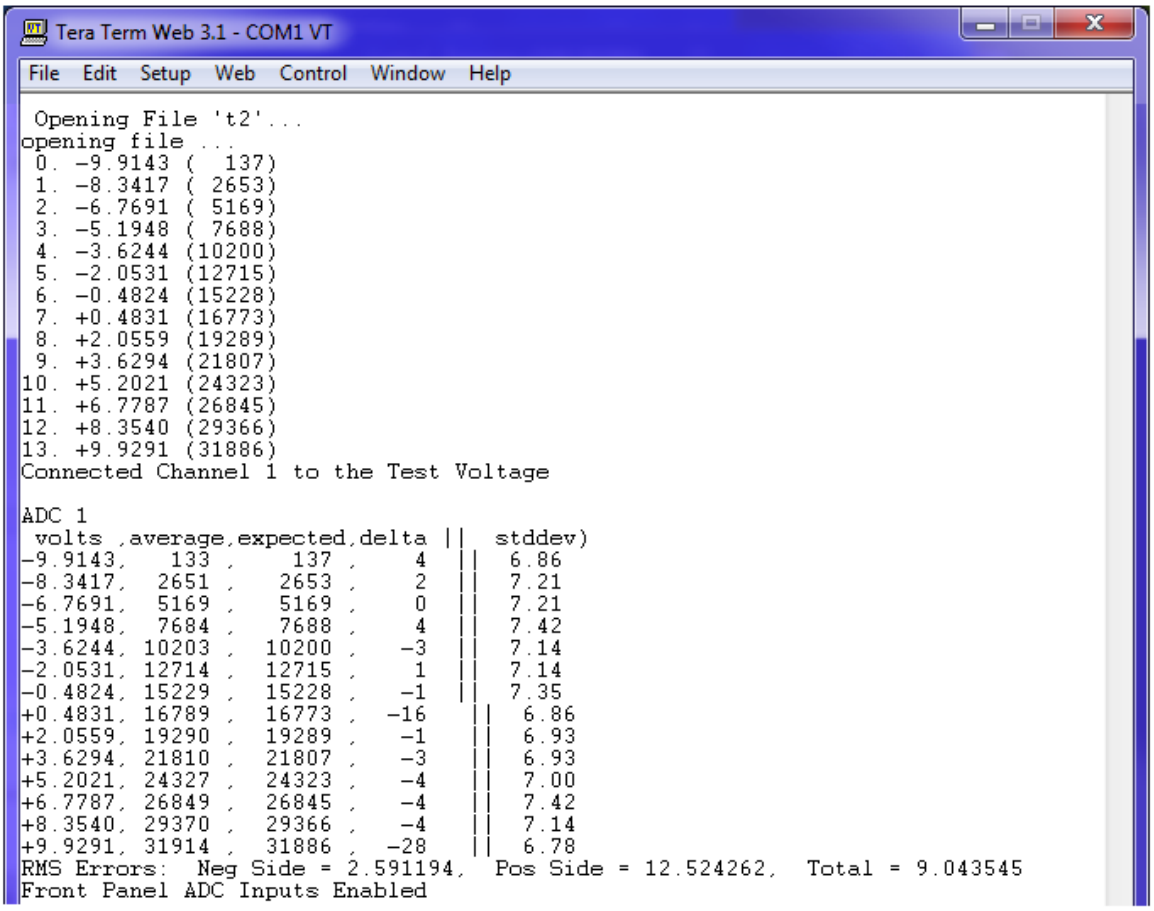

Figure VIII.4 "testchan" results after adjusting the digital offset pot for channel 1.

IX. Measuring the Phase Error Feedback

The phase error feedback voltage comes into the module through analog input IN 1. When the Beam Valid Gate input on digital input DIN 4 is active (High) the value of the phase error feedback is multiplied by a settable gain and added to the Frequency Curve frequency word that sets the frequency of DDS channels 1, 2 and 3. The frequency of DDS channel 4 is, in normal operation, is set to be 28.3 MHz higher than the other channels. DDS channel 4 is used as the BPM Local Oscillator RF. In test mode, with DIP switch SW[6] down, DDS channel 4 outputs RF at the Frequency Curve frequency, unaffected by the phase error feedback.

To measure the effect of the phase error feedback we can measure the frequency difference between DDS channel 1 and DDS channel 4 in test mode. Figure IX.1 shows the setup for this measurement. The MI Phase Lock Controller shown in the setup is just used for its phase detector. Recall that the output of a phase detector will be a triangle-wave with a period that will be the reciprocal of the difference in frequency of the two inputs $(1/\Delta freq)$.

Figure IX.2 show an oscilloscope displaying the phase detector output whose input signals are different in frequency by 50.0 Hz.

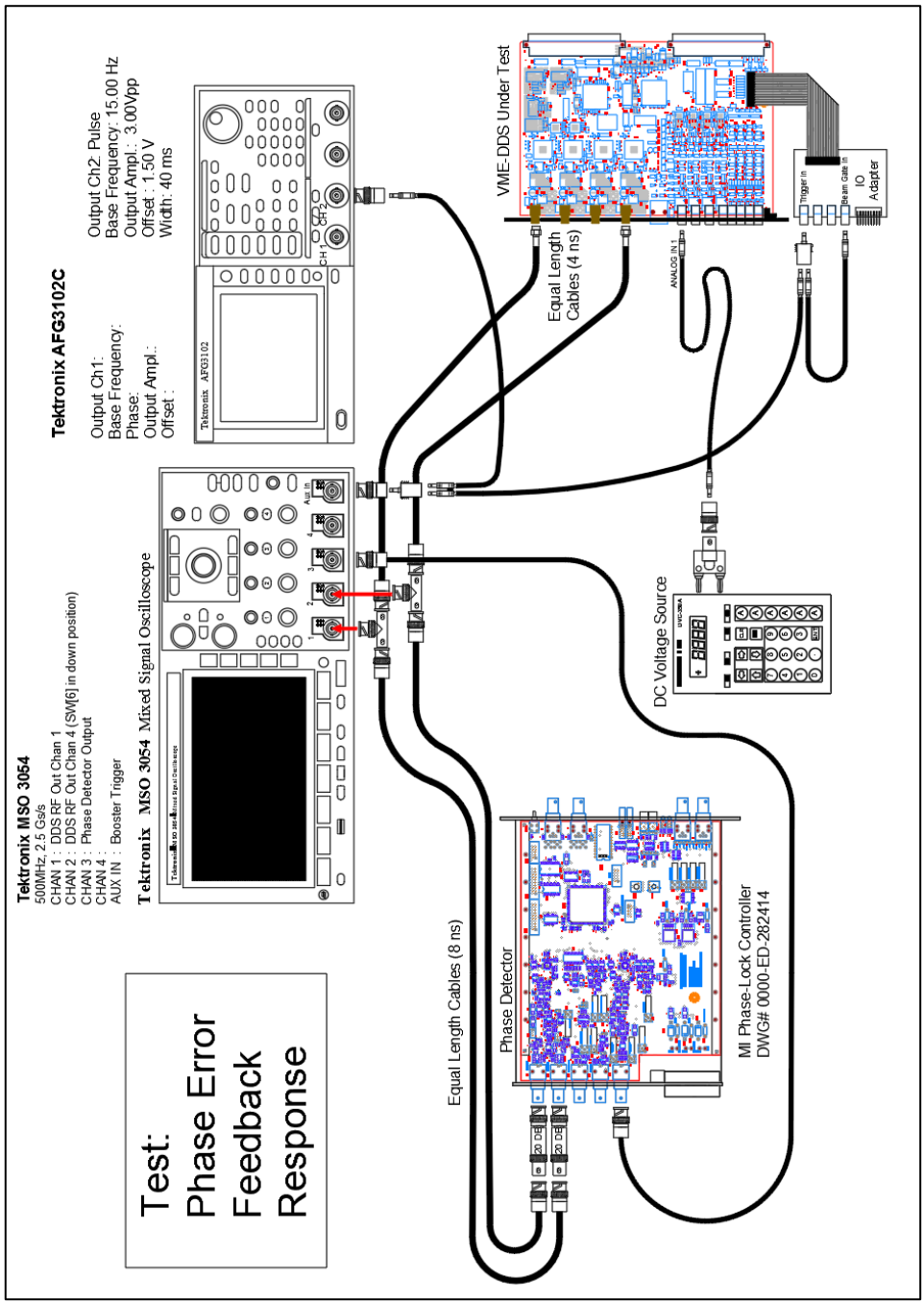

Figure IX.1 Setup for measuring the effect of the phase error feedback.

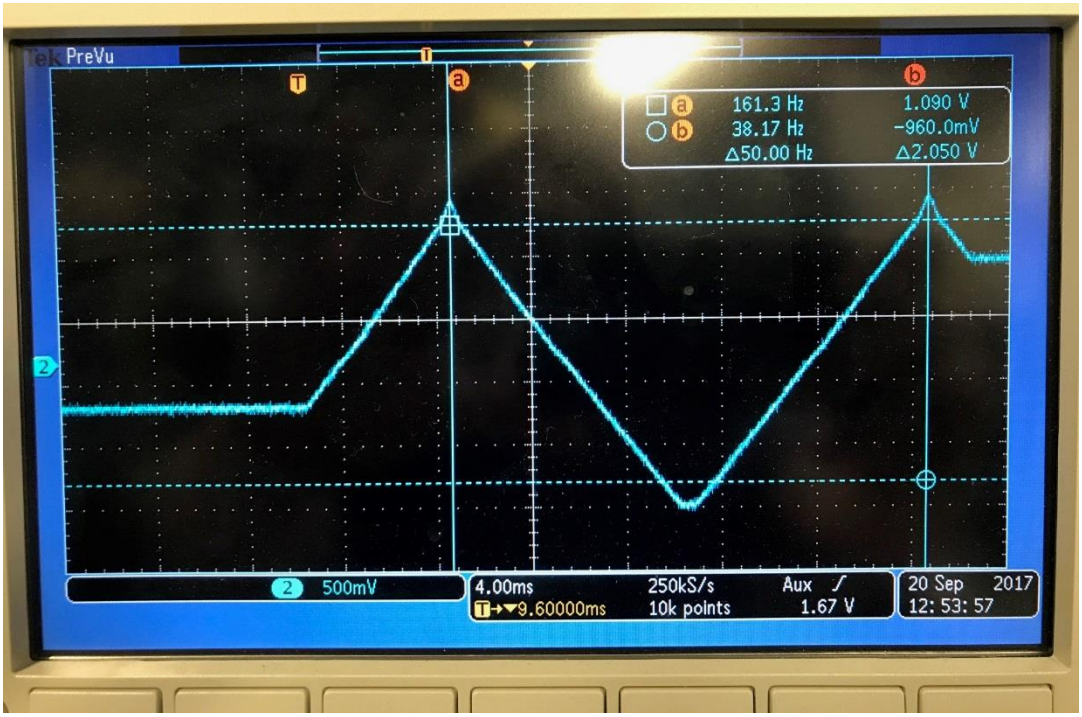

Figure IX.2 Scope trace of phase detector output for a frequency difference of 50 Hz..

The procedure for measuring the phase error feedback is as follows.

1. Using the serial terminal interface set the Phase Error Gain parameter to 100 (0x64). The command is wrparam(4,100). You can use the command “params” to ensure the setting was made. If the setting does not take and or the Flag for the parameter remains ‘1’, ensure that the 15 Hz trigger is firing and that SW[7] is in the up position.

2. Likewise, adjust parameter 7, Phase Error Trim, for a minimum slope on the phase detector output when the input phase error feedback is set to zero volts.

3. Using the DC Voltage Source, set the phase feedback error to the values in the table below and record the resulting frequency difference.

Table IX.1 Data for Phase Error Feedback Measurement

V pherr, Volts	Delta Freq. Hz
-1.000	
-0.800	
-0.400	
-0.200	
-0.060	
-0.020	
0.000	
0.020	
0.060	
0.200	
0.400	
0.800	
1.000	

Table IX.2 Sample Measurement

V pherr, Volts	Delta Freq. Hz
-1.000	-17,670
-0.800	-14,120
-0.400	-7,072
-0.200	-3,526
-0.060	-1,064
-0.020	-339
0.000	42
0.020	397
0.060	1,099
0.200	3,597
0.400	7,299
0.800	14,350
1.000	17,860

X. Measuring the Radial Position Phase Drive Feedback

The Radial Position Phase Drive feedback will shift the phase of the RF out of DDS channels 2 and 3 +/- 90 Degrees given a +/-10 Volt signal at analog input IN 2. The response of RF to this signal can be measured using the setup in Figure X.1.

1. With the setup in Figure X.1, the sinusoidal phase drive feedback from the signal generator (red cable) will provide a signal that swings +/- 9V since we do not provide the 50 Ohm termination expected by the signal generator.

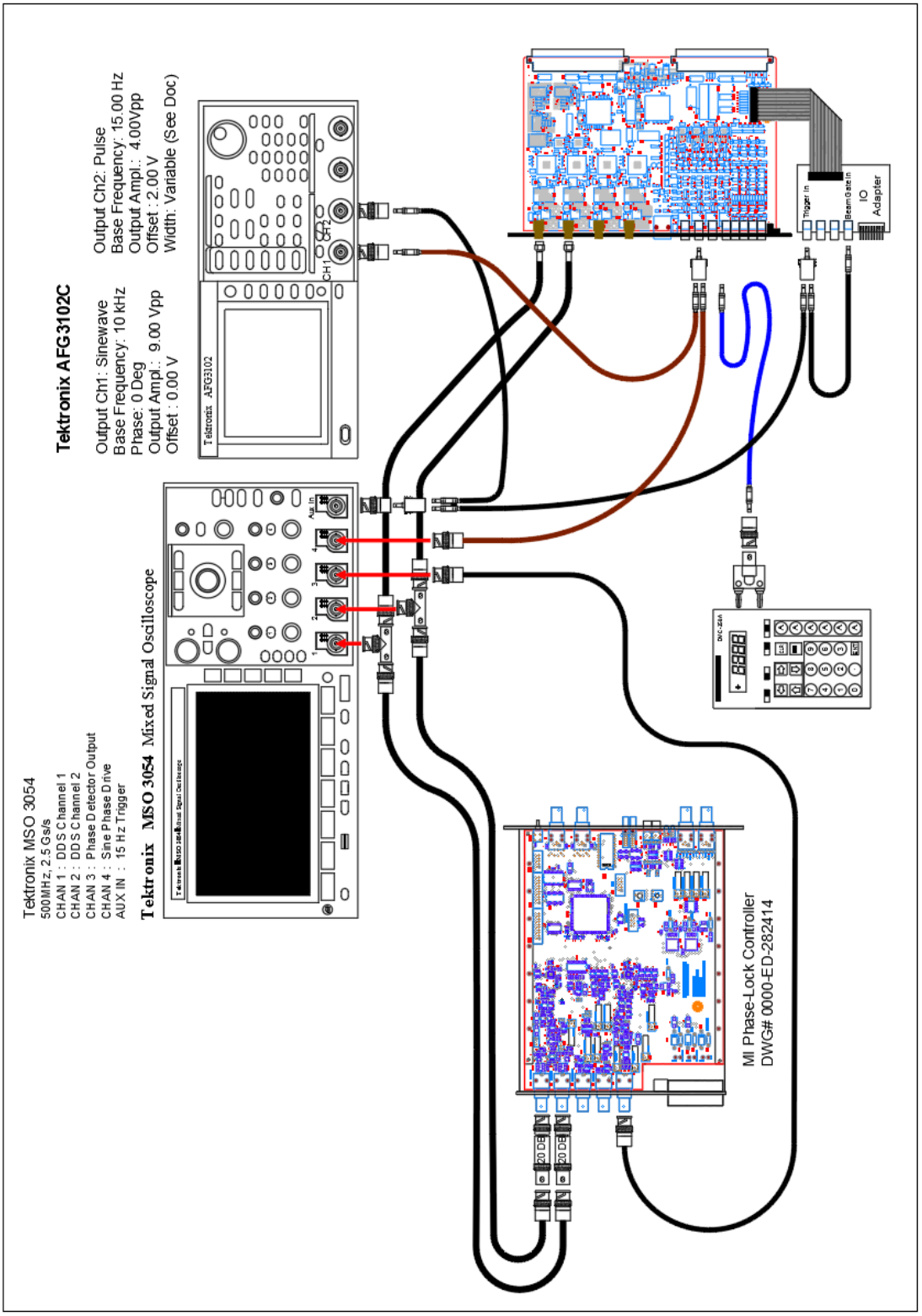

Figure X.1 Phase Drive measurement setup.

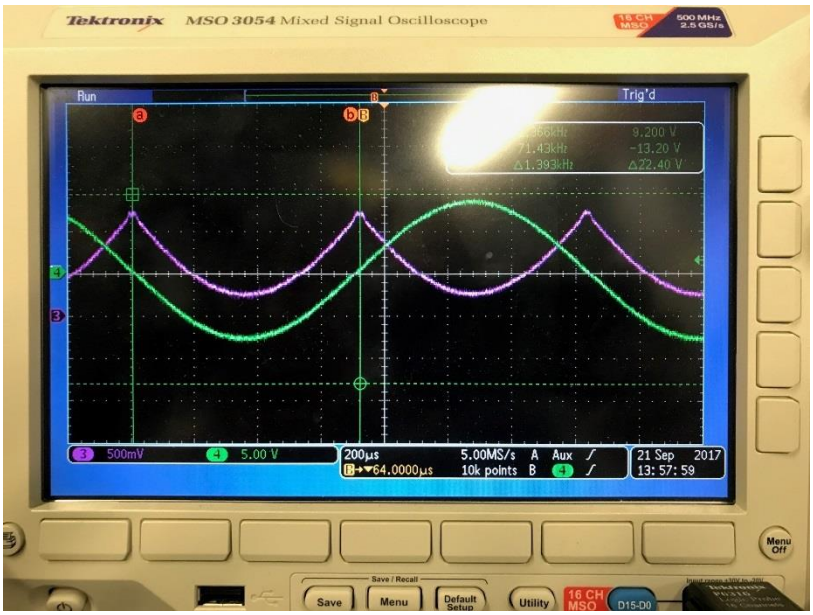

Figure X.2 Green trace is the phase drive. The magenta trace is the phase response.

2. Monitoring the phase difference between DDS channels 1 and 2 we will see something like Figure X.2.

3. The transfer function of the phase detector should be kept in mind when looking at the phase response. As seen in Figure X.3 the output of the phase detector, both a 90 degree shift and a −90 degree shift approach zero volts. The maximum positive phase detector output results when the phase difference is zero degrees.

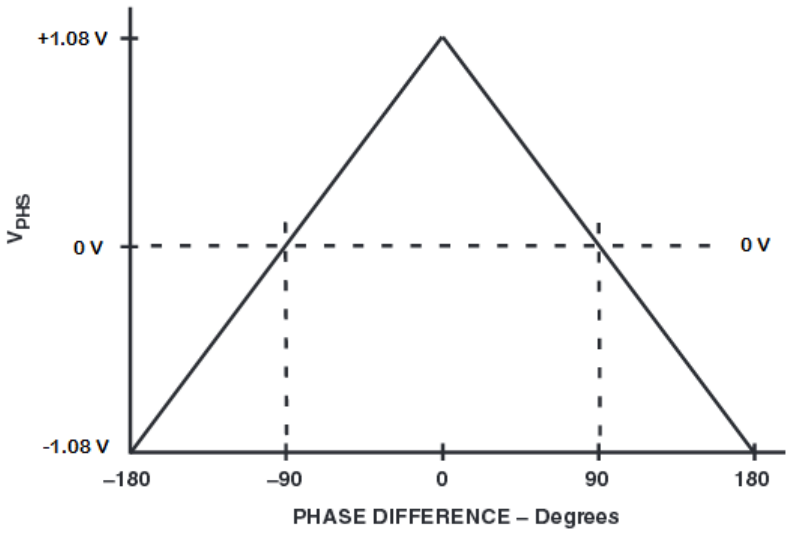

Figure X.3 Phase detector transfer function.

4. To distinguish between 90 degrees and −90 degrees and get an overall calibration of the RPOS phase drive feedback we change the setup to use the DC voltage source (blue cable) and make some RF phase measurements with the scope.

5. To make phase measurements with the scope
   i. AC couple the two RF signals.
   ii. Ensure that each scope channel's reference position and offset is zero.
   iii. Expand the time scale (4.00 ns/div) so that you can see two rising edges of the RF for the frequency verification step. The time scale can be expanded further (2.00 ns) when measuring the phase difference.
   iv. Expand the volts per division down to as small as 20 mV/div. This will give you sharp discernable zero crossings.
   v. Change the acquisition setting of the scope from "sample" to "average". This will average out the phase variations do to the harmonic distortion of the DDS outputs.

6. The frequency of the RF that we are measuring the phase of needs to be known to convert the nanoseconds of time between zero crossings into degrees of phase difference. Since we are triggering our scope with the trigger that starts the frequency sweep of the RF, we expect the RF frequency to be the Injection Frequency that is set by the Injection Frequency parameter, parameter 0. Use the "params" command to determine this value. Also, measure the time between successive rising edge zero crossings of one of the RF signals.

7. Fill out the following table by setting the phase drive feedback voltage with the DC voltage source and measure the resulting time between zero crossings of DDS channel 1 and DDS channel 2 RF.
   Note that if DDS 1 Leads DDS 2, then the phase shift is positive.

Table X.1 Table to Record Phase Drive Feedback Measurements.

Note: Highlight the calculated column and press F9 to recalculate the numbers.

Frequency, MHz	V psdrv, Volts	delta T, ns	Phase Detector Volts	Phase Difference, Degrees *Calculated Column* =R2C1*1e6*360*1e-9*RxC3
37.94	-9	-5.60	0.111	-76.49
	-8	-4.94	0.228	-67.47
	-7	-4.30	0.347	-58.73
	-6	-3.68	0.466	-50.26
	-5	-3.00	0.584	-40.98
	-4	-2.38	0.703	-32.51
	-3	-1.72	0.821	-23.49
	-2	-1.08	0.939	-14.75
	-1	-0.46	1.057	-6.28
	0	0.20	1.226	2.73
	1	0.84	1.087	11.47
	2	1.48	0.971	20.21
	3	2.12	0.853	28.96
	4	2.76	0.733	37.70
	5	3.44	0.616	46.98
	6	4.04	0.497	55.18
	7	4.72	0.378	64.47
	8	5.40	0.259	73.76
	9	5.96	0.140	81.40

Table X.2 Sample Data for Phase Drive Feedback Measurement.

Frequency, MHz	V psdrv, Volts	delta T, ns	Phase Detector Volts	Phase Difference, Degrees *Calculated Column* =R2C1*1e6*360*1e-9*RxC3
37.94	-9	-5.60	0.111	-76.49
	-8	-4.94	0.228	-67.47
	-7	-4.30	0.347	-58.73
	-6	-3.68	0.466	-50.26
	-5	-3.00	0.584	-40.98
	-4	-2.38	0.703	-32.51
	-3	-1.72	0.821	-23.49
	-2	-1.08	0.939	-14.75
	-1	-0.46	1.057	-6.28
	0	0.20	1.226	2.73
	1	0.84	1.087	11.47
	2	1.48	0.971	20.21
	3	2.12	0.853	28.96
	4	2.76	0.733	37.70
	5	3.44	0.616	46.98
	6	4.04	0.497	55.18
	7	4.72	0.378	64.47
	8	5.40	0.259	73.76
	9	5.96	0.140	81.40

XI. Test the Phase Offset Curves

The VME-DDS module has Phase Offset Curves that can be setup and loaded to memory the same as the Frequency Curve, using ACNET application B30. There are two phase offset curves. There is one for DDS channel 2, "poff1", and one for DDS channel 3, "poff2". Refer to section "Setting up Frequency and Bias Curves" to recall how to use ACNET application B30. To enable editing in B30 you need to click the "Enable Edit" button. You can then vary the phase offset curve values.

The previous section described how to monitor the effect of the phase changes with the phase detector and scope. Below is some sample data taken.

Frequency, MHz	Phase Setting (UNIT)	delta T, ns		Phase Difference, Degrees *Calculated Column* =FREQ*1e6*360*1e-9*delta_T
52.22	-46000	13.42		252.285264
	-32000	9.36		175.960512
	-12000	3.46		65.045232
	-4000	1.14		21.431088
	2000	-0.57		-10.715544
	4000	-1.17		-21.995064
	8000	-2.32		-43.614144
	12000	-3.49		-65.609208
	16000	-4.67		-87.792264
	20000	-5.82		-109.411344
	24000	-7.01		-131.782392
	28000	-8.16		-153.401472
	32000	-9.38		-176.336496
	36000	-10.46		-196.639632

Linear Fit	
deg/unit	offset
-0.005483	-0.0036294
unit/deg	
-182.3741	

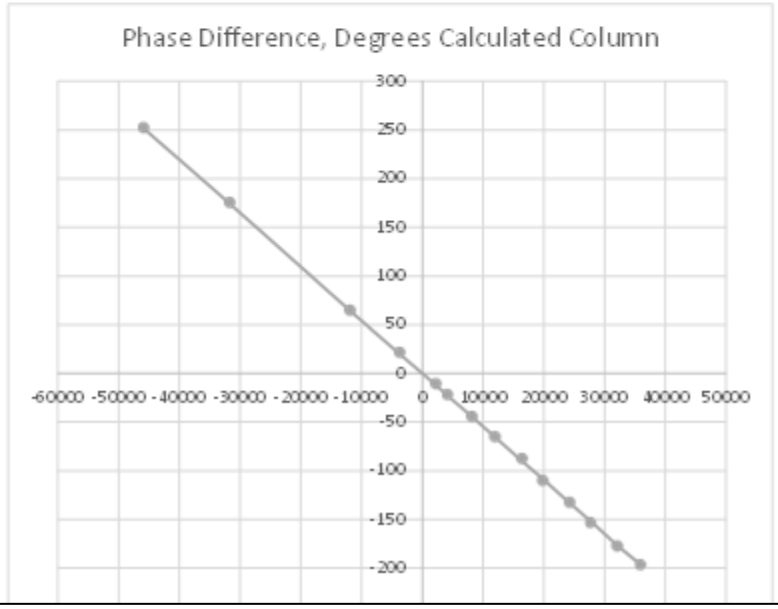

Figure XI.1 Sample Data take to see the effect of the Phase Offset

XII. Technique for Measuring Specific Points in the Frequency Sweep

The DDS RF outputs are swept in frequency according to the Frequency Curve in its memory. The sweep starts a settable time delay (CRVDLY) from the occurrence of the cycle trigger input's **rising edge**. The pulse output of the signal generator produces this rising edge trigger. The **falling edge** of the pulse signal can be used to trigger the oscilloscope, where the pulse width adjustment of the pulse signal can accurately offset the triggering of the scope with respect to the triggering of the frequency sweep.

Given the time in the frequency sweep we want to present on the oscilloscope we can set the pulse width to that time plus the settable time delay value (CRVDLY). Additionally, the signal generators typically allow the user to dial in the value of the pulse width so we can see how frequency, phase and amplitude of the DDS RF signal change as we sweep through the frequencies, by dialing the dial.

See Figure XII.1 for the setup.

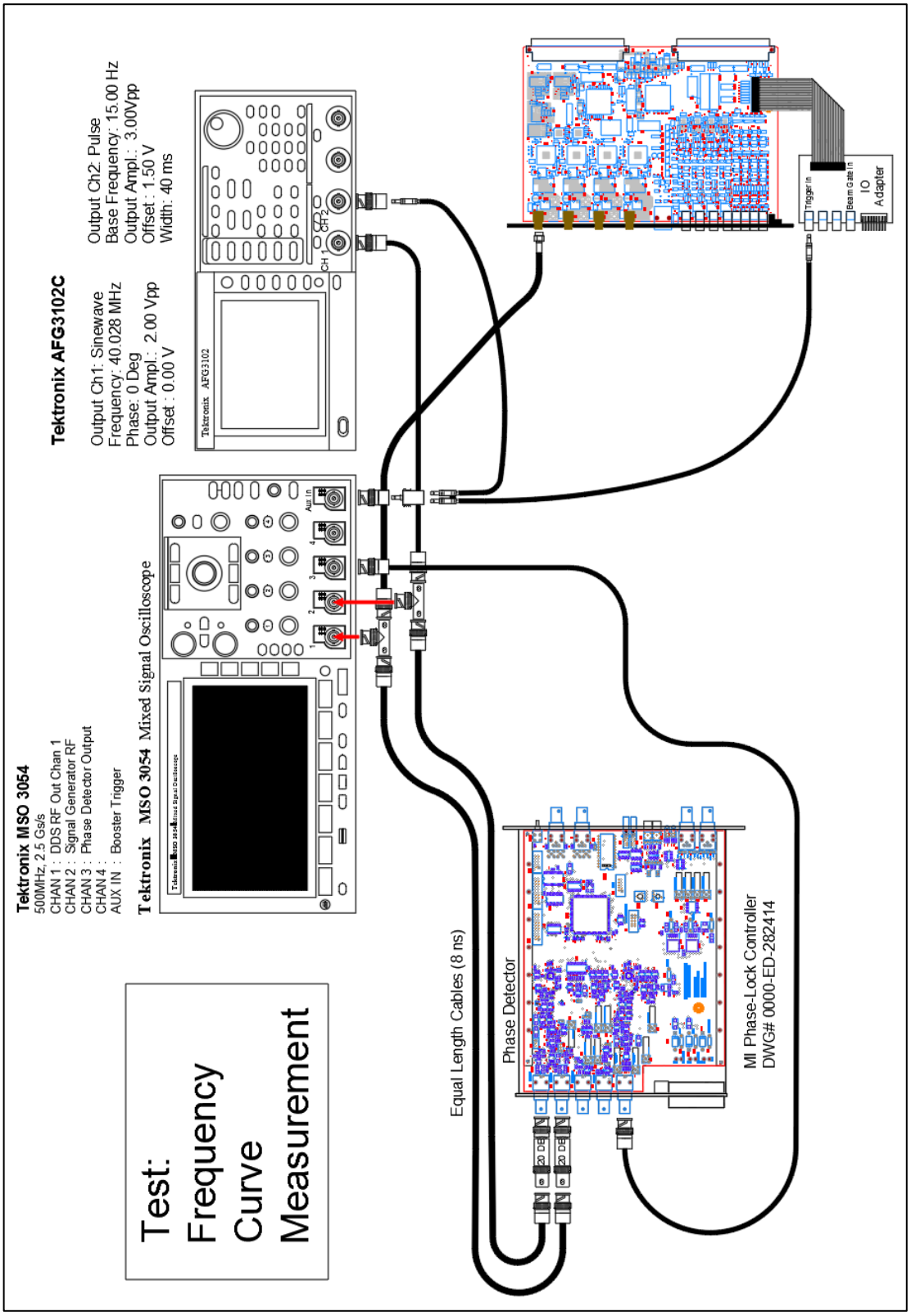

Figure XII.1 Setup for measuring parameters of the frequency sweep.

XIII. Measure the RF Amplitude Throughout the Frequency Sweep

Using the method in Section XI., measure the amplitude of the RF amplitude through the 36 ms sweep. To get an accurate amplitude measurement, the oscilloscope time scale must be small enough to present an un-aliased display of the wave form. By using the falling edge of the signal generator trigger and changing the pulse width of the trigger, you can scroll through the entire sweep noting any changes in RF amplitude.

The criteria we want the RF amplitude to meet is

1. The RF amplitude is at least 1.45 Volts peak to peak.

2. The RF amplitude does not vary more than 4% (approx.. 60 mV out of 1.5 Vpp).

3. Fill in the table below. Use averaging on the scope to get smooth sinewaves to measure. Use the smallest volts/div. on the scope channels that provides a full peak to peak view of the waveforms.

Table XIII.1 RF Amplitude Measurement Data.

Offset from Trigger, ms	RF Amplitude, Vpp		
	DDS Ch 1	DDS Ch 2	DDS Ch 3
1	1.492	1.540	1.512
2	1.496	1.532	1.512
4	1.484	1.532	1.508
8	1.468	1.504	1.496
12	1.460	1.496	1.496
18	1.456	1.488	1.496
25	1.452	1.488	1.492
Max	1.496	1.540	1.512
Min	1.452	1.488	1.492
Range	0.044	0.052	0.020
Percent Var.	2.9%	3.4%	1.3%

Table XIII.2 Sample Data for RF Amplitude Measurement.

Offset from Trigger, ms	RF Amplitude, Vpp		
	DDS Ch 1	DDS Ch 2	DDS Ch 3
1	1.492	1.540	1.512
2	1.496	1.532	1.512
4	1.484	1.532	1.508
8	1.468	1.504	1.496
12	1.460	1.496	1.496
18	1.456	1.488	1.496
25	1.452	1.488	1.492
Max	1.496	1.540	1.512
Min	1.452	1.488	1.492
Range	0.044	0.052	0.020
Percent Var.	2.9%	3.4%	1.3%

XIV. Verify Function of the Tuning Parameter

The following sections cover how to verify that the settable tuning parameters in the VME-DDS module are working as expected. We will be using the setup in Figure XIV.1 to measure the RF frequency at various points in the sweep.

XIV.1 Measure the Injection Frequency

The Injection Frequency is parameter 0, as set by the serial terminal interface. We will be using the signal generator to find the frequency that causes the phase detector output to flatten out. At this point the frequency of the DDS channel and the signal generator are nearly equal. Follow the procedure below.

1. With the setup, back in Figure XII.1, set the frequency of the signal generators sinewave output to the Injection Frequency read through the serial terminal interface.

2. Observe the phase detector output on a 2.00 ms / div. time scale with the scope trigger point at mid screen.

3. Adjust the frequency of the signal generators sinewave output by first hundreds of Hertz then tens of Hertz, then single Hertz to get the flat portion of the phase detector trace to be level. See Figure XIV.1.

4. Note the values of the parameter 0 Injection Frequency and the setting on the signal generator.

Injection Frequency Setting (param 0)	Signal Generator Setting
37,945,000 Hz (sample data)	37,944,808 Hz (sample data)

5. Change the Injection Frequency setting, parameter 0, by 10 kHz. Perform the steps above to find the signal generator setting and note the frequencies below.
   Note to set parameter 0, enter a value that is 8.947791 bits/Hz times the desire frequency setting using the "wrparam(0, "value")" command. Enter the "params" command to view your changes.

Injection Frequency Setting (param 0)	Signal Generator Setting
37,955,000 Hz (sample data)	37,954,801 Hz (sample data)

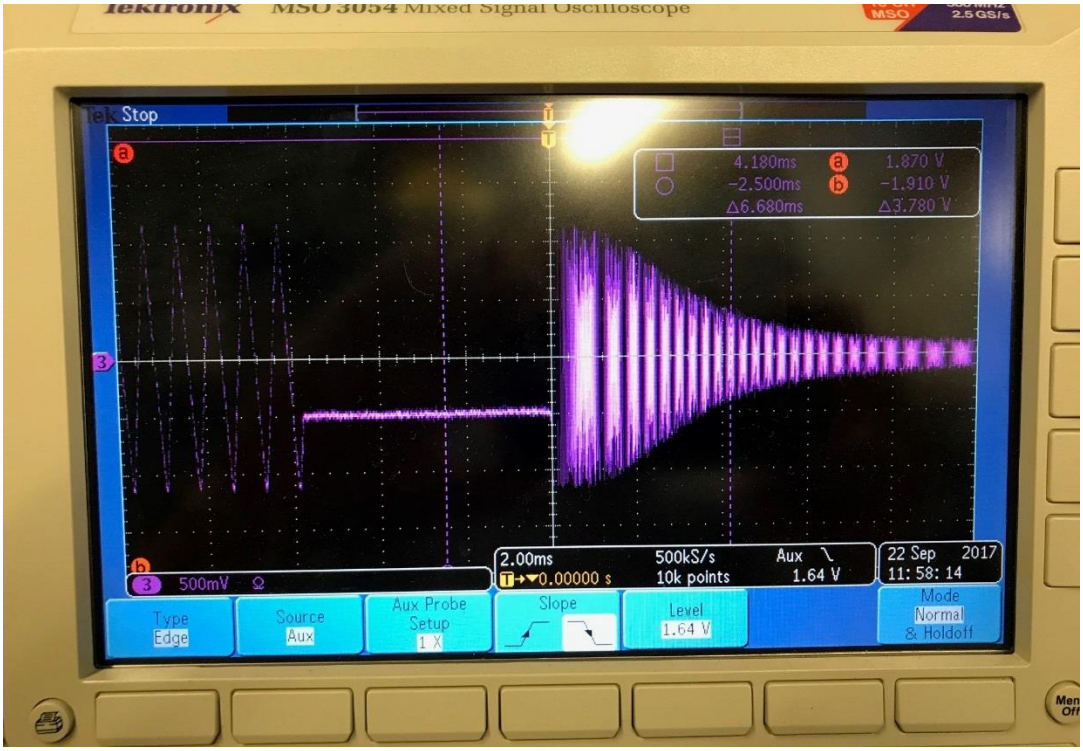

Figure XIV.1 Scope shot of Injection Frequency Measurement.

XIV.2 Measure the Curve Delay

The Curve Delay is the delay between the receipt of the trigger and when the Frequency Curve starts to sweep. It's the amount of time we hold the DDS frequency at the injection frequency after the trigger. There are a couple ways to determine when the first frequency update is made after the trigger. Digital output DOUT 2 is by default pulses indicating a transfer of data from the Frequency Curve memory. See Figure XIV.2 to see where this is on one of the adapters. Analog Output 1, the Frequency Curve, DAC waveform, can be put on the scope with averaging and a 2.00 mV/div. scale. At this small voltage scale, you can see when it starts to be updated. See Figure XIV.3.

The Curve Delay is determined by counting 20 ns clocks. So, the Curve Delay will be the raw Curve Delay value, parameter 2, times 20e-9.

Try two or three different values of curve delay and verify the delay on the scope.

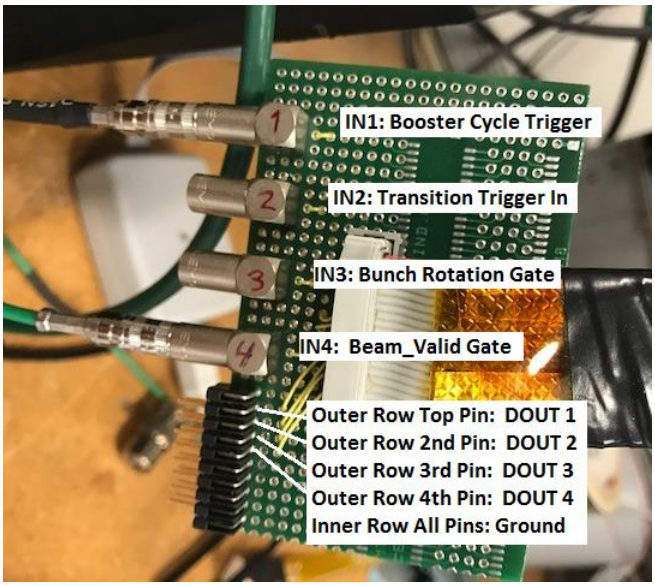

Figure XIV.2 Adapter for Digital IO.

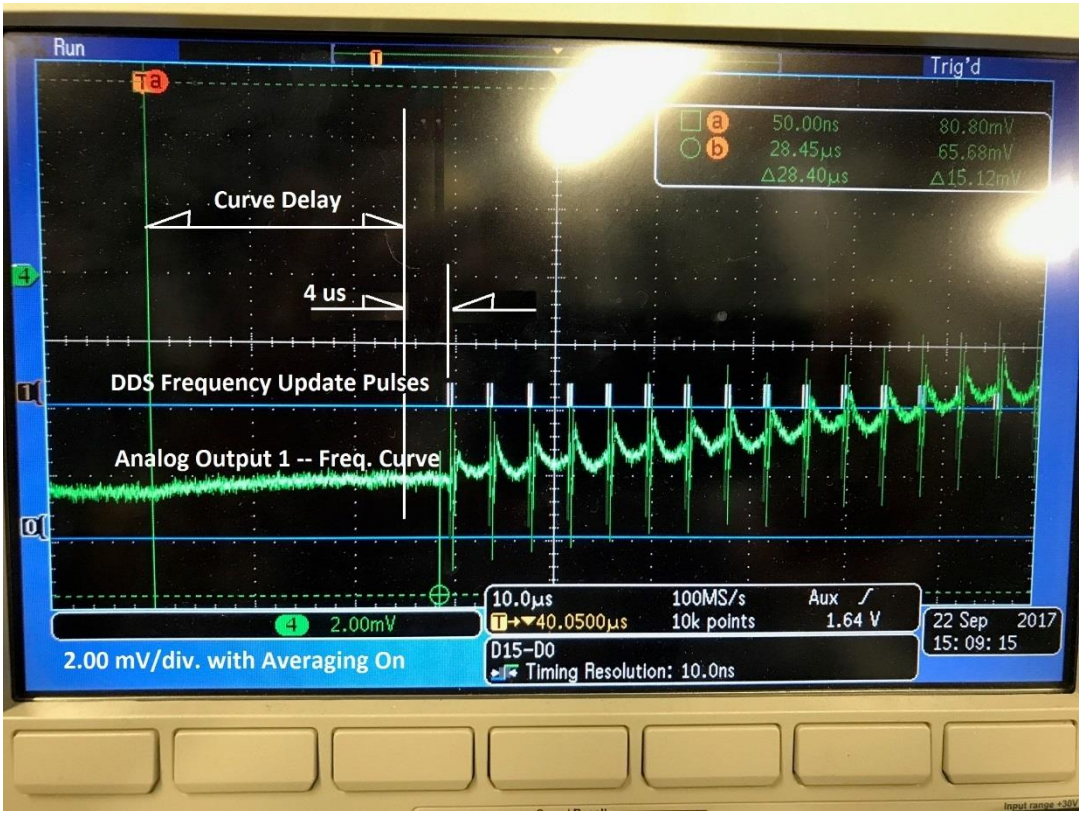

Figure XIV.3 Scope shot measuring the Curve Delay.

XIV.3 Measure the Transition Trigger Frequency

The Transition Trigger fires when the DDS frequency word, going out to the DDS's, rises above a set frequency. The set frequency is B:TTXFRQ on ACNET or parameter 5 through the serial terminal interface. The parameter scales the same as the Frequency Curve or the Injection Frequency, 8.947791 bits/Hz.

The Transition Trigger is output on digital output DOUT1 (see Figure XIV.2). When setting the Transition Trigger Frequency, we want to see that it triggers at the right frequency. By using the phase detector, we can locate where in the frequency sweep a particular frequency is.

With the signal generator set to a frequency within the sweep, the phase detector output will show a lobe, or a flattening, or a turning around in the triangle wave output whose period is the reciprocal of the frequency difference. The scope shots in Figure XIV.4 show this point where frequencies match at two different time scales.

To measure the frequency where the Transition Trigger fires, set the signal generator sinewave to a frequency equal to the Transition Trigger Frequency. Then make fine adjustment of the signal generator frequency in the thousands and hundreds of Hertz to center the lobe of the phase detector output.

To center the Transition Trigger on the scope we could simply trigger the scope with it. However, if we trigger with the adjustable falling edge of the cycle trigger pulse out of the other signal generator channel, we can observe the movement of the Transition Trigger as we vary the Transition Trigger Frequency.

To test the Transition Trigger Frequency, do the following.

1. Setup the test equipment as in Figure XII.1.

2. Using the serial terminal note the value of parameter 5 using the "params" command, then enter wrparam(5,0x1bd9c86e) to set the Transition Trigger Frequency to 52.22 MHz.

3. Monitor Transition Trigger, DOUT1, on the scope.

4. Triggering the scope on the falling edge of the signal generator cycle trigger pulse, center the rising edge of the Transition Trigger.

5. Monitor the output of the phase detector on the scope and set the frequency of the signal generator sinewave to 52.220000 MHz. The scope should look like Figure XIV.5.

6. With the scope set to 200 us / div., change the Transition Trigger Frequency by entering wrparam(5,0x1be76cd8), then set the signal generator sinewave to 52.320000 MHz. The Transition Trigger should move to the right of the scope display.

7. Change the Transition Trigger Frequency again by entering wrparam(5,0x1bcc1e56), then set the signal generator sinewave to 52.120000 MHz. The Transition Trigger should move to the left of the scope display.

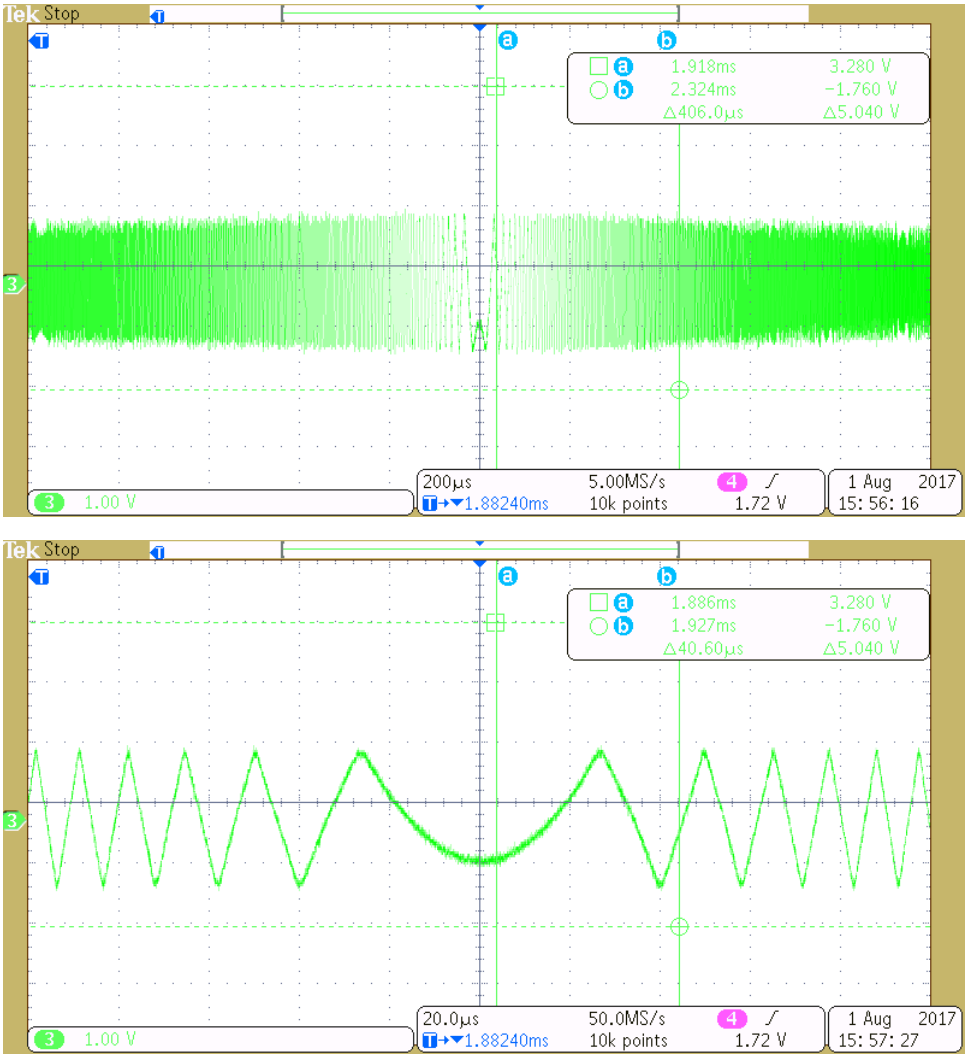

Figure XIV.4 Scope shot showing a lobe at the point where frequencies match.

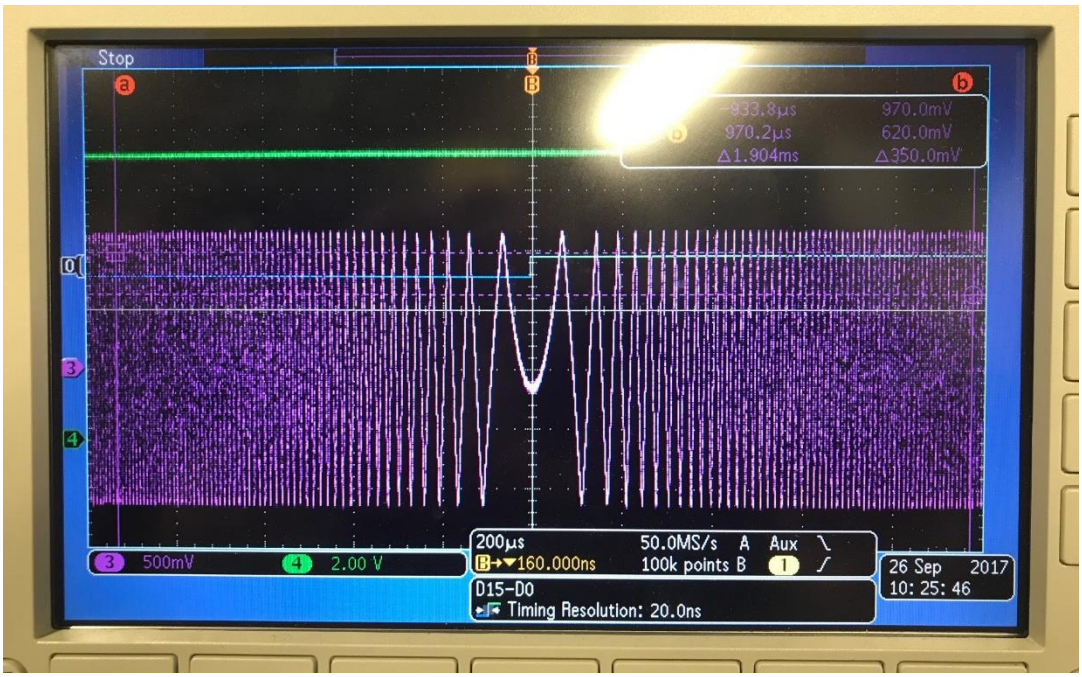

Figure XIV.5 Scope shot showing the Transition Trigger at the predicted frequency.

XV. APPENDIX A.

"nova.fnal.gov" Files used for Bench Testing

The listings below are important file for interfacing with VME and VXI modules under test.

Listing A.1 Startup file for MVME processor node BLRFD3.

```
# ID: $Id: startup,v 1.1.1.1 2004/01/12 22:02:40 waller Exp $
# For blrfd3 FE
# Use this startup for AnIntFlt class ACNET/VXI interface

# Run desired starup up script file
# Comment out others

cd "/fecode-bd/vxworks_boot/fe/blrfd3"

# ----  Startup file choices ----
#========= NEW STARTUP ============
#>>> Used for Booster VXI LLRF stuff
#< startup_hybrid

#========= NEW STARTUP ============
#>>> Used for misc. tests.
# was using this on prior to troubleshooting phdev #
#< startup_wallerTests

#========== SIMPLE STARTUP ============
# Use the following startup for checking out the Paraphase Module
#< phase_ctrl_startup

#>>> Used for BLM Digitizer Card Testing <<<
#< blm_startup

#>>> Used for the LLRF Utility Module Testing <<<
# OLD startup < vxidsp_startup
#< mini_vxi_startup  <<<<<<<< previous VXI startup

#CURRENT VME STARTUP
< mini_startup

#=========================#
# R E M I N D E R         #
#-------------------------#
# For VME Bus (non-VXI RM)
# Base Addresses Are
# MVME2300 --> 0xfa00 0000
# MVME5500 --> 0xbf00 0000
#-------------------------#
```