LabVIEW User's Manual

Digital Gamma Finder (DGF)

Pixie-4/Pixie-500

Version 1.0, September 2011

XIA LLC

31057 Genstar Road
Hayward, CA 94544 USA

Phone: (510) 401-5760; Fax: (510) 401-5761
http://www.xia.com

Table of Contents

1 Introduction

This document describes the LabVIEW user interface of the Pixie-4 and Pixie-500 modules. For a full description of the modules' functions and features, please refer to the Pixie-4 and Pixie-500 User Manuals. In particular, references in this manual to sections 4 and higher imply these sections are located in the Pixie-4 User Manual.

Section 2 is intended to give a quick overview of installation and getting started with the Pixie Viewer. A description of all VI is found in section 3. For full details of the advanced controls in the VIs, please consult the Online Help manual released as part of the software distribution. It mainly describes the Igor Pro version of the Pixie Viewer, but the controls are generally named the same.

As far as the interface is concerned, the difference between Pixie-4 and Pixie-500 manifests in only three issues:

1) The correct DSP file name has to be selected for booting: “P500code.xxx” vs “PXIcode.xxx”
2) The interface automatically rounds user specified times (e.g. trigger filter length) to closest multiples of the base digitization or processing time (2ns or 13.33ns).
3) When viewing captured waveforms, the time unit has to be set correctly

2 Setting Up

2.1 Installation

2.1.1 Hardware Setup

The Pixie-4 and Pixie-500 modules can be operated in any standard 3U CompactPCI or PXI chassis. Chassis that adhere only to the CompactPCI standard can only operate modules individually. To communicate between modules a backplane adhering to the PXI standard must be present. To begin, place the embedded computer (or remote controller) in the system slot of your chassis. Place the Pixie modules into any free slots with the chassis still powered down, then power up the chassis (Pixie modules are not hot swappable). If using a remote controller, be sure to boot the host computer *after* powering up the chassis[1].

2.1.2 Drivers and Software

When the host computer is powered up the first time (after installing the controller and Pixie modules in the chassis) it will detect new hardware and attempt to find the appropriate drivers. (A Pixie module will be detected as a new device every time it is installed in a new slot.) While

[1] In some systems, “scan for hardware changes” in the Windows device manager may detect and install a remote chassis when the PC was booted first.

there is no required order of installation of the driver software, the following sequence is recommended:

1. If you have a remote controller, first install the driver software for the controller itself. Otherwise, skip to step 4.
   Unless directed otherwise by the manufacturer of the controller, this can be done with or without the controller and Pixie-4 modules installed in the host computer and/or chassis. If the modules are installed, ignore attempts by Windows to install drivers until step 7.

   NI controllers come with a multi-CD package called "Device Driver Reference CD". For simplicity it is recommended to install the software on these CDs in the default configuration.
2. Unless already installed, power down the host computer, install the controller in both the host computer and chassis, and power up the system again (chassis first).
3. Windows will detect new hardware (the controller) and should find the drivers automatically. Verify in Window's device manager that the controller is properly installed and has no "resource conflicts".
4. Install LabVIEW, version 8.5 or higher
5. Install the Pixie-4/500 software provided by XIA (see section 2.1.3)
6. Unless already installed, power down the host computer and install the Pixie-4 modules in the chassis. Check the input jumper settings for the appropriate signal termination: 50 Ω or 5 kΩ (see section 10.1 for details). Then power up the system again (chassis first).
7. Windows will detect new hardware (the Pixie modules) and should find the drivers automatically. If not, direct it to the "drivers" directory in the Pixie software distribution installed in step 5. Verify in Window's device manager that the modules are properly installed as "PLX Custom (OEM) PCI 9054 Boards" and have no "resource conflicts". Currently, the driver must be version 6.4. The previously used driver version 4.1 will identify the modules as "Custom (OEM) PCI 9054 Boards" without the "PLX"

2.1.3 Pixie LabVIEW User Interface

To "install" the Pixie LabVIEW interface (called Pixie Viewer), copy the complete distribution from the installation CD (or zip file) to the folder C:\XIA\Pixie4, or to a custom folder. Then follow the instructions in drivers/readme to install the drivers. The Pixie Viewer distribution contains several LabVIEW Virtual Instruments (VIs) and eight subfolders (Configuration, Doc, Drivers, DSP, Firmware, MCA, PixieClib, and PulseShape). Make sure this folder organization is preserved, as future updates will be based on this structure. Feel free, however, to add folders and subfolders at your convenience.

Functional (programming) or cosmetic (appearance) changes to any of the VIs may be saved by clicking on *File -> Save* from the top menu. This action saves the current state of the interface. This same action must be performed on all open VIs to save their status.

Generally, for any action in the VIs to be executed, the user first has to click the button (or other widget), then click on the "run" button at the top of the panel.

2.2 Getting Started

2.2.1 Booting (PIXIE4BOOT.VI)

To initiate communication with a Pixie module, double-click on the file PIXIE4BOOT.VI in the LabVIEW folder. Prior to booting the Pixie module, the correct paths to the firmware, DSP files and settings files must be set (blue underlined text suggests a likely locations for the pertinent files) as shown in Figure 2.1. The procedure to change and save the defaults settings is

1. Ctrl-M to enter the VI edit mode
2. Edit the text fields to insert the correct paths
   If booting a P500 module, then replace the DSP files with "P500code.xxx"
3. Choose Edit->Make Current Path Default in the Edit menu
4. Save Pixie4Boot.vi

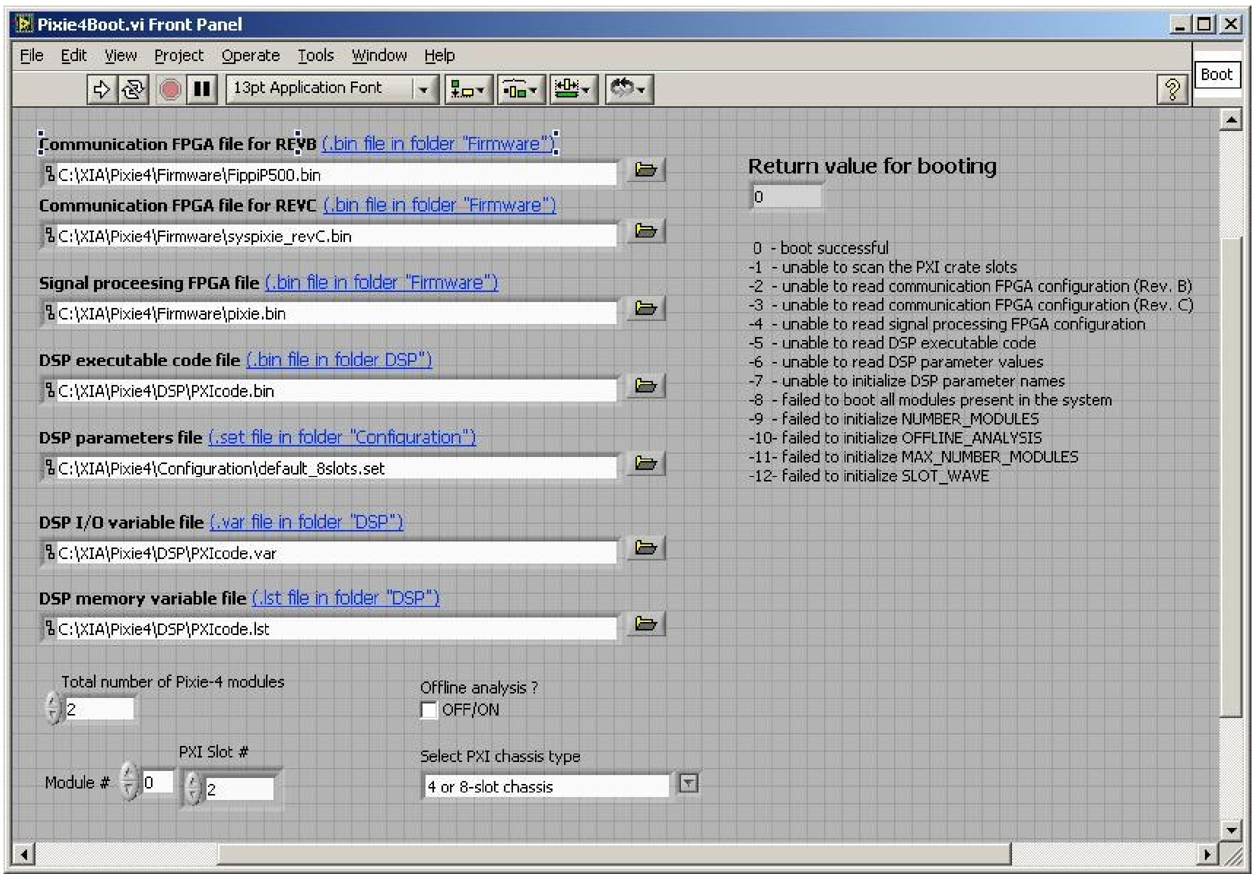

Figure 2.1: The PIXIE4BOOT VI

In addition to the paths, the total number of Pixie-4 modules must be entered, as well as the PXI slot mapping for each module. This action is executed by entering the PXI slot number for module 0, toggling the module number, and entering the slot for the next module and so on. Lastly, the chassis type must be selected. If you want to run the software without a chassis or module attached, check *Offline Analysis*.

The Pixie-4 is booted by clicking the run button (arrow) in the top panel[2]. A series of return values indicates the boot status - a return value of 0 indicates a successful boot sequence. Otherwise, refer to the troubleshooting section for possible solutions.

2.2.2 View ADC traces (Pixie4ADCTraces.vi)

With the Pixie-4 booted it is strongly suggested that the ADC waveforms be inspected to ensure that the sampled waveforms fall into the ADC voltage range before data is acquired.

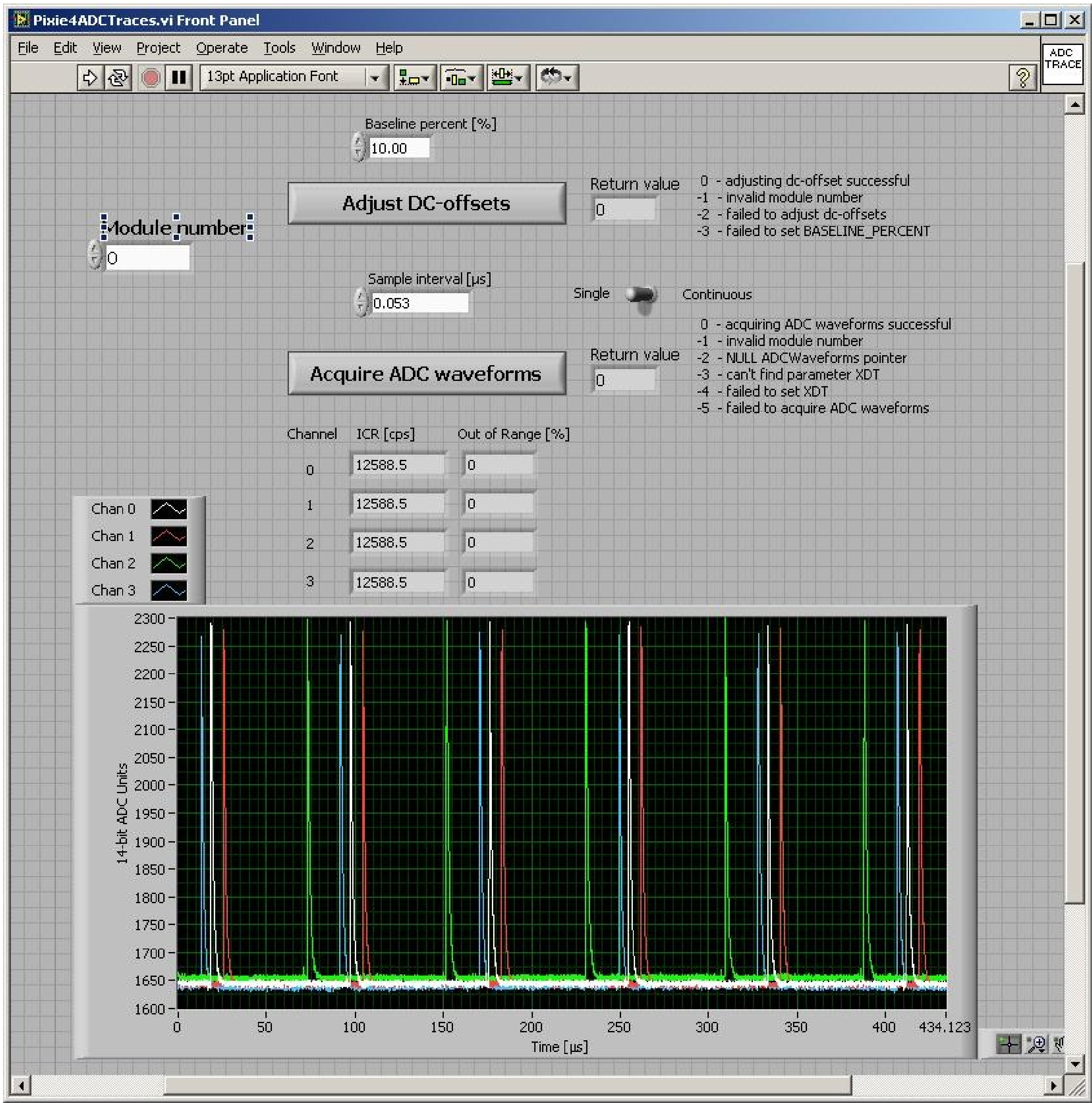

Figure 2.2: The Pixie4ADCTraces VI

[2]All subsequent action issued from any of the VI modules require that the run button be pressed for the command to be processed. This additional step is implied in all further descriptions.

Waveforms can be viewed by launching Pixie4ADCTraces.vi (Figure 2.2). Executing *Acquire ADC waveforms* displays traces from all 4 channels spanning 8192 time steps**.** *Sample interval* sets the time between successive steps. The toggle widget allows the user to switch between *Single* trace or C*ontinuous* trace acquisition. As each channel may have slightly analog different gains and/or offsets when a module is booted, *Adjust DC-Offsets* automatically sets a common DC-offset level for each channel. *Baseline* allows the user to define that DC-offset level.

After any adjustments first execute *Adjust DC-Offsets,* then *Acquire ADC waveforms*. *Return value* indicates the status of these actions. Waveforms should be comfortably within ADC range, i.e. baseline at the specified level, pulses starting with a rising edge, and not clipped at the upper end of the ADC range. If any of the pulses are inverted, *Trigger positive* in the Register A tab of the Pixie4SetUserPar VI should be toggled (Section 3.2). If pulses go out of range, adjust the *Gain* in the *Analog Signal Conditioning* tab in the Pixie4SetUserPar VI, then again execute *Adjust DC-Offsets.* You can also manually specify an offset in the *Analog Signal Conditioning* tab. Signals that still fall out of the ADC range must either be terminated differently or the detector gain reduced.

Generally speaking, changes to parameters made in the Pixie4SetUserPar VI should follow this sequence: First ensure panel values are identical to the values present in the module's DSP by executing *Upload.* Then change values and apply them to the DSP by executing *Download* at the bottom of this same VI .When changes are applied, values are rounded or limited according to the module's valid range, so the final step should be to execute *Upload* again to see true values used by the module. At any point modified DSP parameter settings may be saved to file. To do so, launch the Pixie4SaveDSPPartoFile VI. It is advisable to create a new file name to avoid overwriting the default settings file.

2.2.3 Data Acquisition

Once the ADC traces have been properly adjusted, the pulse decay time constant has to be specified. In Pixie4SetUserPar VI, click on the *Decay time* tab and enter an estimated preamplifier exponential RC decay time for *Tau.* Save settings as described above.

To start an data acquisition run, open the Pixie4DAQ VI. Click on the *Run control* tab, set *Run Type* to “0x301: MCA Mode”, *Poll time* to .1 second, and *Run time* to 30 seconds or so, then click on the run button.

To view the result, open the Pixie4MCA VI and execute *Refresh Histogram.*

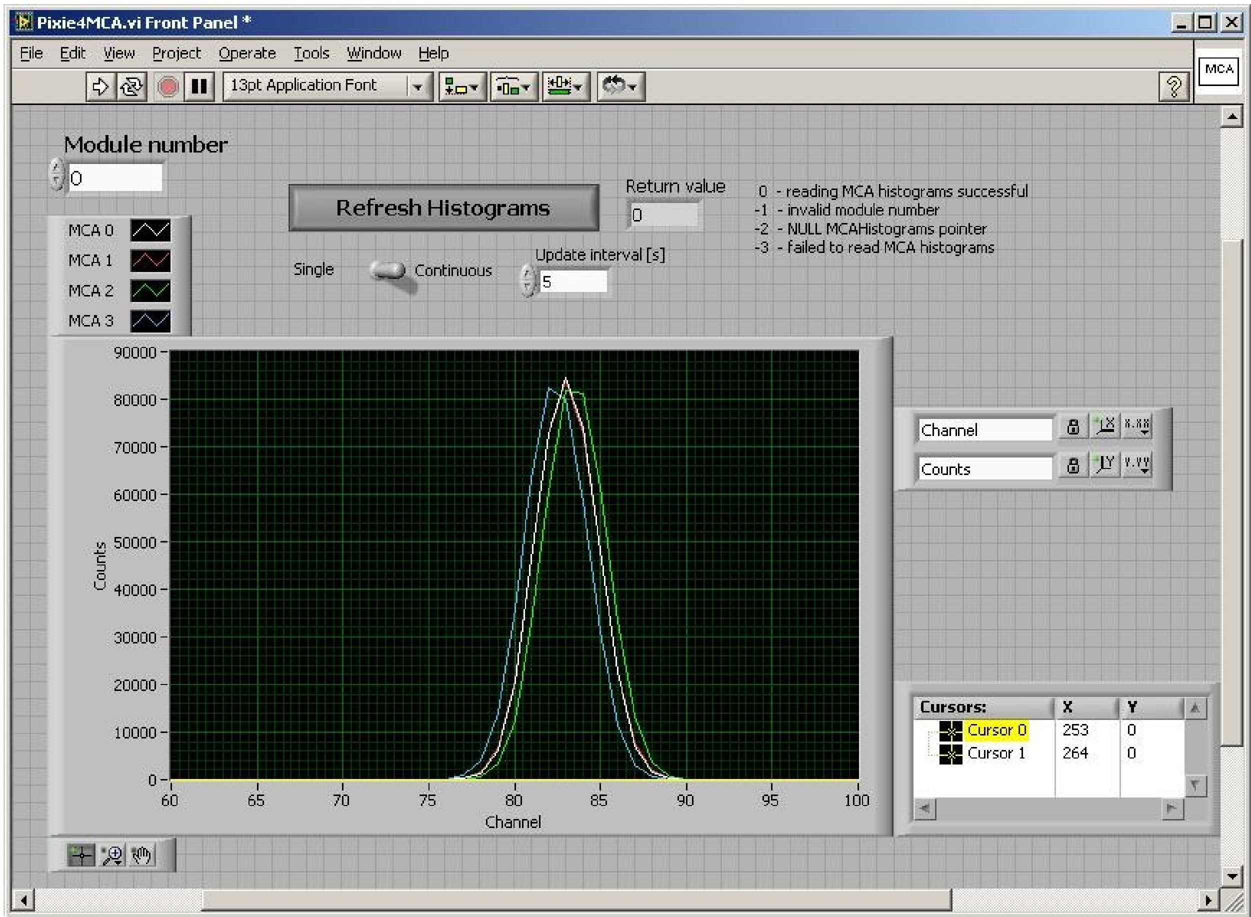

Figure 2.3: The Pixie4MCA VI

This brief overview is intended to quickly get the user up and acquiring data with the Pixie-4. Undoubtedly there are user parameters settings such as energy filter rise time and flat top that may need adjustment to optimize your results. These parameters, along with other a host of logical control parameters are described in the next section.

3 Navigating the LabVIEW VIs

3.1 Overview

The LabVIEW interface consists of eight VIs, each of which can be independently launched on an as-needed basis. Loosely speaking they can be grouped into three categories: Setup Control (PIXIE4BOOT.VI, PIXIE4ADCTRACES.VI, PIXIE4SETUSERPAR.VI and PIXIE4SAVEDSPPARTOFILE.VI – Section 3.2 and 3.3), Run Control (PIXIE4DAQ.VI – Section 3.4) and Results (PIXIE4MCA.VI, PIXIE4PULSESHAPE.VI and PIXIE4READRUNSTATISTICS.VI – Section 3.5). This grouping is shown in Figure 3.1

Setup Control

Pixie4Boot.vi (2.2.1)
Boot modules
Set location of firmware, settings, and output files

Pixie4ADCTraces.vi (2.2.2)
View Traces
Adjust offsets

Pixie4SetUserPar.vi (3.2)
Triggers and threshold (3.2.1)
Pule shape analysis (3.2.2)
Gate and veto (3.2.3 and 7.4)
Tau (3.2.4)
Analog signal conditioning (3.2.5)
Channel coincidence (3.2.6 and 7.6.1)
Module coincidence (3.2.7 and 7.2.2, 7.6.2)
Histogramming (3.2.8)
Baseline (3.2.9)
List mode settings (3.2.10)
Advanced Features (3.2.11)

Pixie4SaveDSPParToFile.vi (3.3)

Run Control

Pixie4DAQ.vi (3.4)
Run control (3.4.1)
Synchronization (3.4.2)
Output file (3.4.3
Data record options (3.4.4)

Results Control

Pixie4MCA.vi
View energy histograms (3.5.1)

Pixie4PulseShape.vi
View waveforms and event data (3.5.2)

Pixie4ReadRunStatistics.vi
Live times and count rates (3.5.3, and 6.6)

Figure 3.1: Block diagram of the main panels in the Pixie LabVIEW Interface. Numbers in brackets refer to the corresponding sections in the manual.

3.2 Setup Group - PIXIE4SETUSERPAR.VI

All user parameters are accessed via the PIXIE4SETUSERPAR VI. Figure 3.2 is an overall view of the VI with tabs so designated as to give the user some idea of the their functionality. The PIXIE4SETUSERPAR VI is itself divided into two subpanels: channel (left) versus module/system parameters (right).

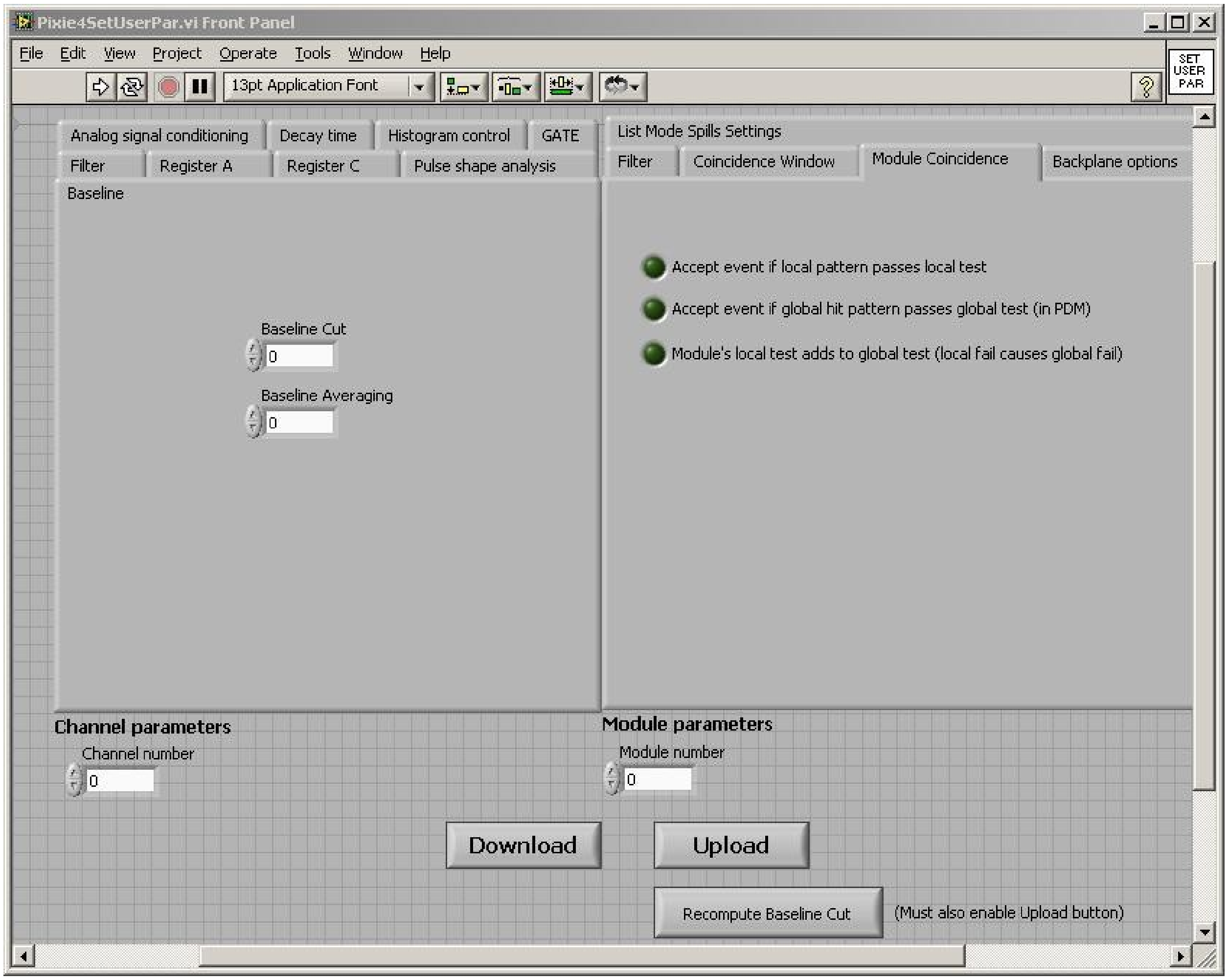

Figure 3.2 PIXIE4SETUSERPAR VI.

User parameters in the PIXIE4SETUSERPAR VI are available to tailor the Pixie module to the particular requirements of a given experimental setup. As befits a digital device, all parameter settings are stored in a settings file. This file is separate from LabVIEW VIs and provides flexibility by saving and restoring different settings for different detectors and applications. At boot time the parameters from the DSP parameter file specified in the PIXIE4BOOT VI are automatically downloaded to the module. After the module is booted changes to any of the user parameter are applied to the module(s) by executing *Upload*, user input, *Download*, *Upload* (to verify) on the PIXIE4SETUSERPAR VI. Unless these changes are committed by writing them to a configuration file using the PIXIE4SAVEDSPPARTOFILE VI they will be lost on the next module reboot. Should an entirely different DSP parameter file be desired a reboot is required.

When a user enters a parameter value for one channel, the same numerical value is automatically copied to all channels (for the sake of convenience; the rationale being that the most commonly encountered experimental setup is multiple channels of the same type of instrument). However, the value entered for that channel *only* is written to the DSP when executing *Download* with this channel selected. For example, if the *Tau* parameter in the *Decay Time* tab is changed for channel 0 and module 0, this same tau value will be copied to the *Tau* text entry boxes of all channels and modules. Executing *Download* would then download this revised tau value to the DSP only for channel 0 and module 0. If this same tau value happens to be appropriate for all channels and and modules, then the user must toggle through each combination of channel and module and follow this same procedure. In a similar vein, if a separate tau value is required for some or all of the channels and modules then the user must again toggle through all combinations of channels and modules, entering a different tau value each time, executing *Download* before moving on to the next channel. Confusion may arise if the user were to toggle back to a channel for which the tau value has already been committed to the DSP since the most recently entered value will instead appear in the *Tau* text entry box. Nonetheless, one can confirm that the correct value for that channel was entered by executing *Upload*.

We now proceed with a detailed description of the functionality of each of the tabs in the PIXIE4SETUSERPAR VI. We start by describing the functionality of the tabs in the left panel before doing the same for tabs in the right panel. Before proceeding, we note that associated functionality for many of the tabs described in this section are found in tabs *Register A* and *Register C*. This will be noted on a case-by case basis.

3.2.1 Filter Tab

The *Filter* tab contains controls to set the trigger and energy filter parameters. These two filters are described in the following subsections

3.2.1.1 Trigger filter

The *Trigger filter* column defines the trigger filter times and *threshold.* Except for the threshold, the trigger settings have rarely to be changed from their default values.

The threshold value corresponds to ¼ of the pulse height in ADC steps, e.g. with a threshold of 20, triggers are issued for pulses above 80 ADC steps. This relation is true if the trigger filter *rise time* is large compared to the pulse rise time and small compared to the pulse decay time. A pulse shape not meeting these conditions has the effect of raising the effective threshold. The threshold value is scaled with the trigger filter *rise time*, therefore it is not limited to integer numbers. The *Register A* tab provides advanced trigger filter functionality via the buttons

- *Respond to group triggers only* to capture waveforms based on distributed, not local trigger,
- *Good channel* to enable or disable a channel,
- *Read always* to force read out of channels even in absence of a hit,
- *Enable trigger* to allow a channel to trigger acquisition (itself and other channels), and
- *Trigger positive* to invert incoming signals (trigger on inverted waveforms).

The remaining settings are described in section in the online help manual.

3.2.1.2 Energy filter

The *Energy filter* column contains the settings for the energy filter and the subsequent energy computation. These settings are paramount for obtaining the best possible energy resolution with a Pixie system. The energy filter *rise time* (or peaking time) essentially sets the trade off between throughput and resolution: longer filter *rise times* generally improve the resolution (up to a certain optimum) but reduce the throughput because more time is required to measure each pulse. The *Integrator* parameter modifies the energy computation. For a detailed description of the filter operation see Section 6 of the user manual.

3.2.2 Pulse shape analysis tab

The *Pulse Shape Analysis* tab contains the controls to set the length and pre-trigger delay of the waveforms to be acquired. Advanced options include parameters for online pulse shape analysis

3.2.3 GATE Tab

The *GATE* tab, in tandem with tabs *Register A* and *Register C* contains the controls to set the window for gating acquisition with external signals. The GATE tab itself contains the text entry boxes

- *Gate Window* to start a counter of length Gate Window, and
- *Gate Delay* to start a counter of length Gate Delay

with toggle options for related controls located in the *Register A and C tabs*. Generally speaking we distinguish between a

- GATE, a dedicated signal for each individual channel. It is active at the rising edge of the pulse e.g. to suppress a detector pulse with a coincident pulse from a BGO shield. *Register C* tab contains four radio buttons related to the this GATE function
    - *GATE signal required to accept events* to validate a recognized event based on the GATE,
    - *Invert GATE polarity for acceptance* to accept or reject if GATE had a pulse,
    - *Route VETO signal to be used for GATE* to use the VETO as gate input, and
    - *Invert GATE polarity* to start GATE delay counter on a rising or falling edge.
- VETO, a signal distributed to all modules and channels, but each channel is individually enabled to require or ignore this signal. VETO is active during the validation of a pulse (after pileup inspection), an energy filter rise time plus flat top after the rising edge. With suitable external logic, the decision to veto a pulse can be made from information obtained at the rising edge of the pulse (e.g. multiplicity from several channels) and therefore this function is also called Global First Level Trigger (GFLT). The *Register A* tab includes two radio buttons related to the this VETO function
    - *GLFT Required* so that a VETO signal must occur to accept a pulse, and
    - *Inverted* VETO signal to accept a VETO (logic 1 or 0).

For a detailed description of the GATE and VETO operation, see sections 6.6.2 and 7.4 of the user manual.

3.2.4 Tau Tab

The pulse decay time *Tau* is used to compensate for the decay of a previous pulse in the computation of the pulse height. The user can enter a known value or use the waveforms from the PIXIE4ADCTRACES.VI to estimate it.

3.2.5 Analog signal conditioning Tab

The *Analog signal conditioning* tab has two text entry boxes whose values can be changed through the increment widget or by manual entry. The two settings are

– *Gain (V/V)* which sets the analog gain before digitization, and

– *Offset (V),* which directly sets the offset voltage.

3.2.6 Coincidence Window Tab

The *Coincidence* tab contains the controls to set the acceptable hit pattern, and the coincidence window after validation during which channels can contribute to the hit pattern. There is a check box for each possible hit pattern. For example, if the check box with pattern 0100 is checked, events with a hit in channel 2 and no others are accepted. Selecting multiple check boxes accepts combinations of hit patterns, e.g. any event with exactly one channel hit. For a detailed description of the coincidence operation, see section 7.2.1 of the user manual.

3.2.7 Module Coincidence Tab

The *Module Coincidence* tab is used to set parameters that affect the system as a whole. Examples are trigger distribution between modules, coincidence settings between modules, and the operation of the Pixie-4's front panel input. Controls for coincidences between modules are

– *Accept event if the local pattern passes the local test* (default),

– *Accept event if the global hit pattern passes the global test*, and

– *Module's local test adds to global test (local fail adds to global fail).*

See sections 7.2.2 and 7.6.2 of the user manual for details.

3.2.8 Histogram Tab

The *Histogram* tab contains basic functionality for histogram accumulation. There are two self-explanatory text boxes

– *Minimum energy*, to set the minimum energy to be histogrammed (bin 0 = min. energy)

– *Binning factor* to combine 2^N bins, which reduces the total number of bins but does not change the energy range in the spectrum.

3.2.9 Baseline Tab

The *Baseline* tab contains the controls for modifying baseline measurements. There are two text boxes

- *Baseline Cut*, and
- *Baseline Averaging*.

Section 6.4 addresses the role that the baseline plays in noise minimization. Once averaging values are input an improved estimate of the baseline cut can be obtained by highlighting the *Recompute Baseline Cut* button on the bottom right of the VI.

3.2.10 List Mode Spill Settings Tab

The radio button *32 buffers* is used in conjunction with the PIXIE4DAQ VI and dumps 32 buffers to file in a single write operation. This mode results in much faster readout and lower dead time. The default is a single buffer.

3.2.11 Advanced Features

This section defines some of the more advanced features of the LabVIEW interface. The description of infrequently used features is omitted.

3.2.11.1 Pulse Shape Pileup

Register C has four radio buttons dedicated to pileup inspection

- *Disable pileup inspection* to accept all events, piled up or not
- *Ignore out-of-range* to accept events even if the signal gos out of range
- *Invert pileup inspection* to accept only events that are piled up
- *Pause pileup inspection* to ignore “double triggers” with a few hundred nanoseconds.

Section 6.4 of the user manual describes how the Pixie-4 treats pileup

3.2.11.2 Backplane options tab

This tab enables

- *Trigger Share Mode* Set to 1 to sharing triggers over the backplane, 0 otherwise

and two buttons

- *Front panel drives GFLT line* to route the front panel input to the GFLT/VETO line
- *Front panel contributes to status line* to record front panel status in hit pattern word

These latter controls are described in Section 7.6 of the user manual.

3.3 Setup Group - PIXIE4SAVEDSPPARTOFILE.VI

Changes to DSP parameters that are to be preserved for future runs must be saved to file using the PIXIE4SAVEDSPPARTOFILE VI as shown in Figure 3.9. A reboot via the PIXIEBOOT VI applies the parameters from the file to the module.

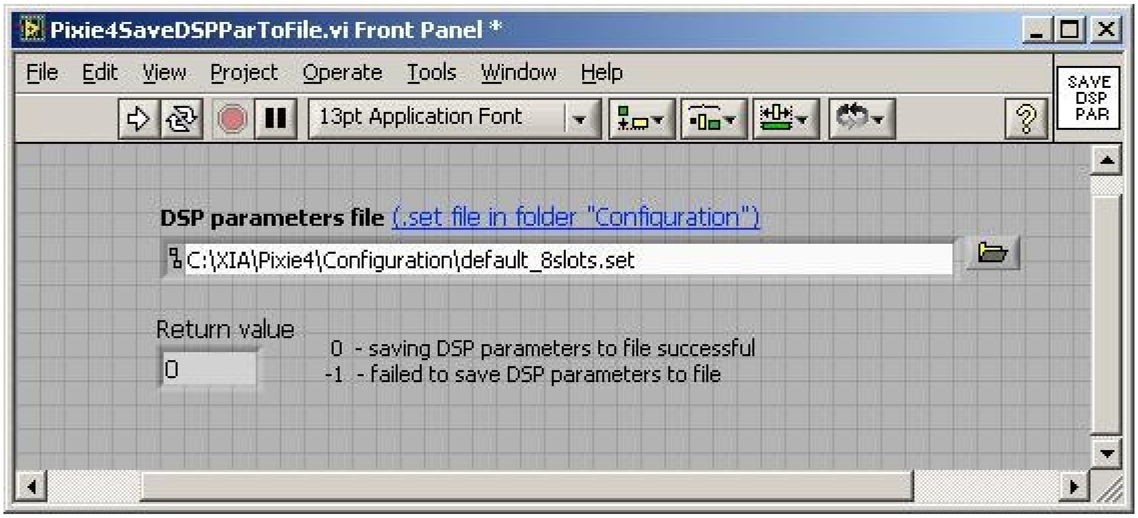

Figure 3.9: THE PIXIE4SAVEDSPPARAMETERSTOFILE VI

3.4 Run Control - PIXIE4DAQ.VI

The Run Control group has the most essential controls to start and stop runs, and to define or monitor the run time and the number of spills.

3.4.1 Run control tab

The *Run Control* tab in the PIXIE4BOOT VI defines the settings for data acquisition. This tab includes

- *Run type* to select MCA or list mode runs – refer to Section 4 for a detailed description.
- *Polling time* to set the polling time (period for checking if list mode data is available for readout and/or run time is reached),
- *Run time* to specify the duration of the run,
- *Number of spills* to specify the number of spills in list mode runs. (In list mode runs, data is accumulated in on-board memory until full, at which time it is read out by the host PC. Each readout is termed a spill. The number of spills thus sets the amount of data to collect.)
- *Spill Timeout* to set a timeout for each spill

3.4.2 Synchronization Tab

The controls include

- *Simultaneously stop/start modules* to stop/start all modules at the same time, and
- *Synchronize clocks to synchronize acquisition between modules* to set all times to zero at runstart.

This functionality is described in Section 7.2 of the user manual.

3.4.3 Output File Tab

In this tab, output files and paths are specified

- *Base name*, *Run number*
  The compete filename is formed from base name and 4-digit run number, e.g. test002.bin. File extensions are .bin and .dat for list mode data, .mca for spectra, .set for settings
- *MCA Data Path* to specify the location of the MCA files. If the default install location is C:XIA\Pixie4\ the MCA files are placed in C:\XIA\Pixie4\MCA
- *List Mode Data Path* to specify the location of the list mode files. If the default install location is C:XIA the list mode files are placed in C:\XIA\Pixie4\PulseShape

3.4.4 Data record options tab

The controls include

- *Auto increment run number* to increment the run number after each run,
- *Auto store spectrum data as binary .mca file* to write the MCA spectrum to file after each run,
- *Auto store setting in .set file after run* to record the settings after each run,
- *Auto store list mode data into .dat file* to convert the binary list mode data into a text file, and
- *New files after every xxx spills* to write a a file and start a new run after the specified number of spills.
  When taking long data acquisitions, it may be beneficial to break up the run into smaller sub runs. This helps to save data in case of power failure or system crashes, since only the most recent sub run is lost. Also list mode files tend to get large and unwieldy for analysis in longer runs.

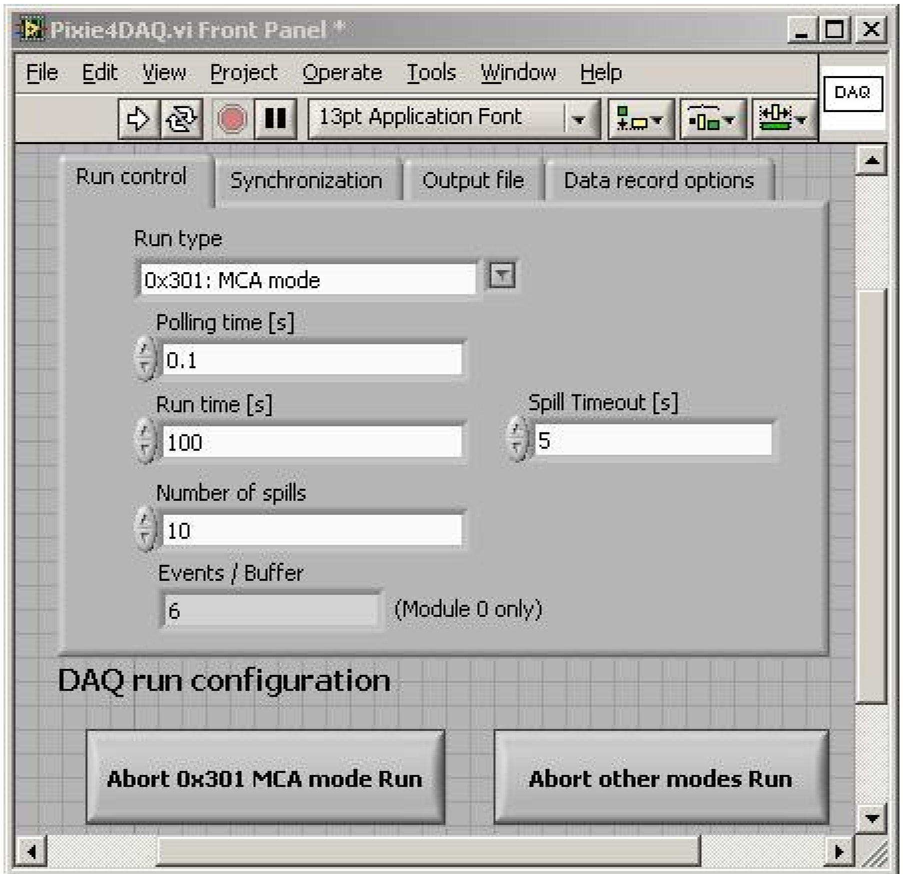

Figure 3.4: The PIXIE4DAQ VI

3.5 Results Group

The Results group includes VIs to view the output data from the data acquisition in detail. It includes the PIXIE4MCA, the PIXIE4PULSESHAPE and PIXIE4READRUNSTATISTICS VIs.

3.5.1 MCA Spectrum (PIXIE4MCA.VI)

The MCA SPECTRUM display shows the spectra accumulated in on-board memory. To observe the spectrum in real time the Pixie4MCA VI must be open and running simultaneously with the PIXIE4DAQ VI. Alternatively, the MCA spectrum can be viewed after the run has ended. The spectrum is updated at a refresh interval set by the text widget *Update interval*. The MCA spectrum graph shows the MCA histograms for all four channels. The display can be rescaled by overwriting any of the text labels on the x- or y-axes. Multiple cursors can be inserted by right clicking in the *Cursor* box, and choosing *Create Cursor.* Once clicked, a cursor labeled *Cursor 0* is created and the user is prompted for *x* and *y* positioning. A second cursor can be similarly created. With the region of interest expanded the cursors can be dragged to either side of the peak to read off the FWHM. This VI operates only on a module's MCA memory. The saved .mca file (32bit unsigned integer binary numbers) can be imported into any suitable application.

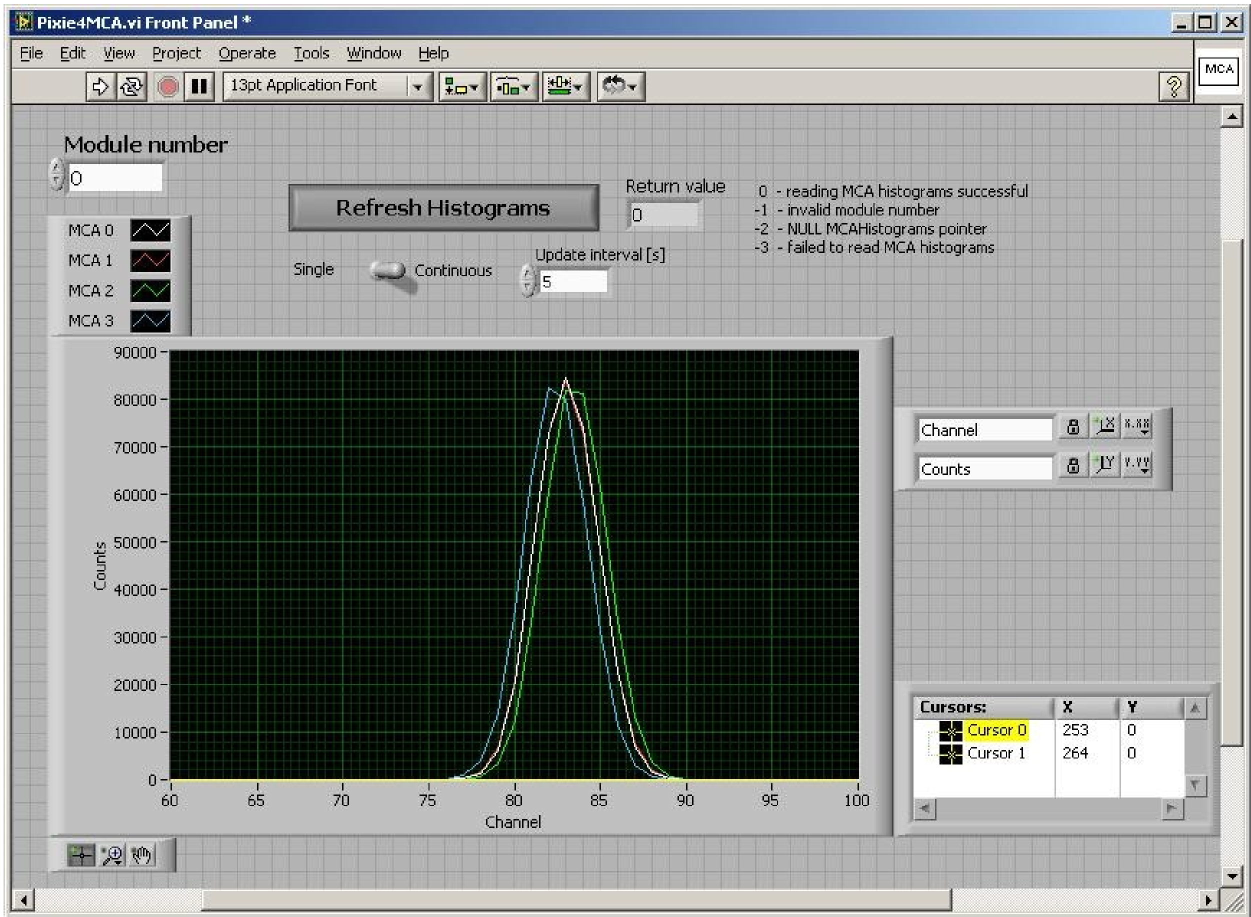

Figure 3.6: The Pixie4MCA VI.

3.5.2 List Mode Traces (Pixie4PulseShape.vi)

The List Mode Traces display shows data from the binary list mode files (.bin). If waveforms were collected, they are shown in the graph section of the panel. Event and channel header information – energy, time stamps, and hit patterns as described in section 4.1.2 – are shown in the fields above the graph section. As opposed to the Pixie4MCA VI the Pixie4PulseShape VI operates only on saved output files. Data can only be viewed once a list mode run has terminated. However, one can open any .bin file, not just that acquired during the last run. After specifying a data file, you can select an event to view by entering its number in the *Trace Number* field. To show the correct time units on the plot, be sure to select the appropriate *ADC sampling rate*.

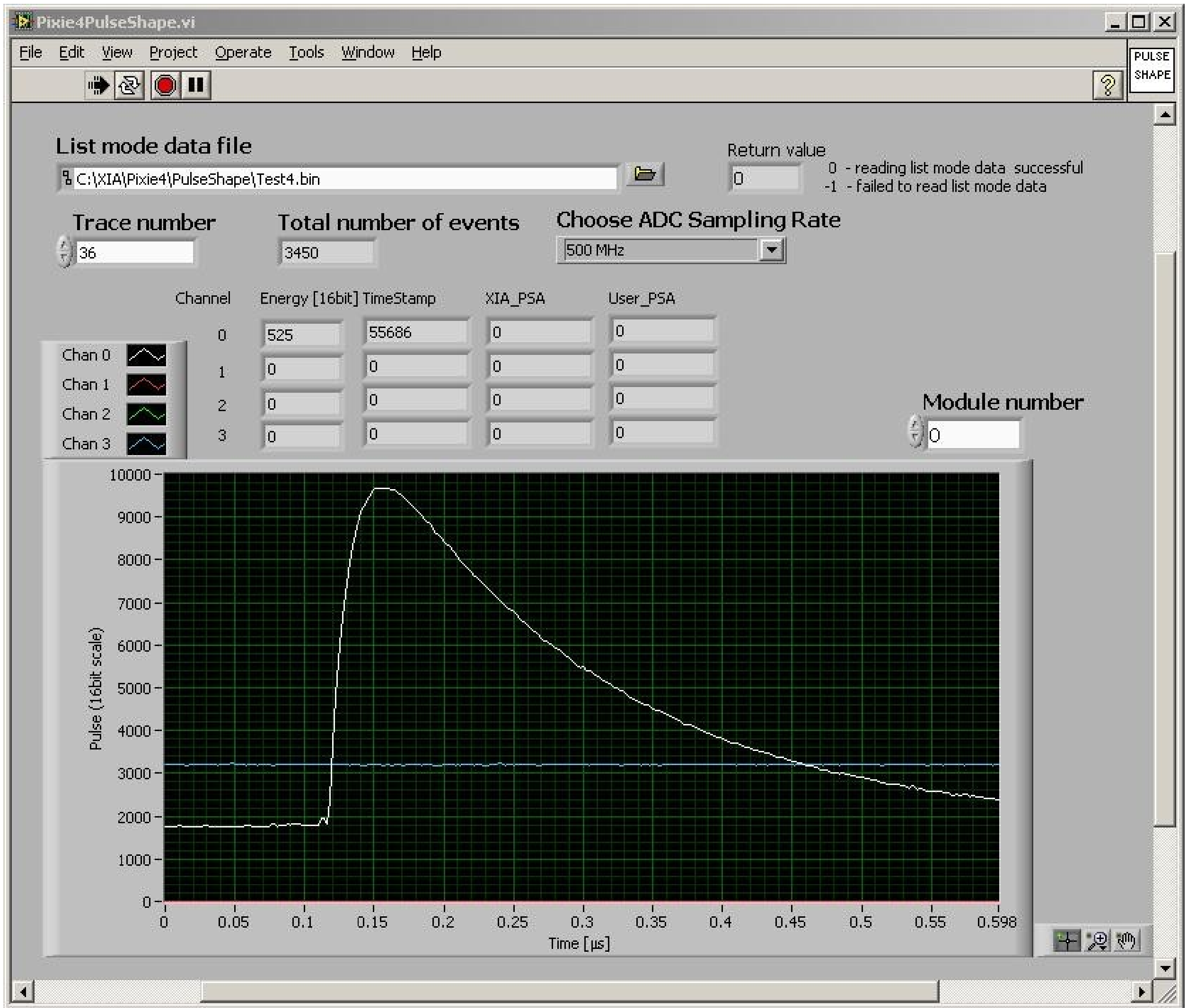

FIGURE 3.7: THE PIXIE4PULSESHAPE VI.

3.5.3 RUN STATISTICS (PIXIE4READRUNSTATISTICS.VI)

The PIXIE4READRUNSTATISTICS VI shows the live times and count rates measured by the Pixie-4. The numbers can be updated by clicking the Run button. Otherwise, the statistics shown are that from the end of the run. For a detailed description of the definition of these values, see section 6.6. Figure 3.8 shows the PIXIE4READRUNSTATISTICS VI

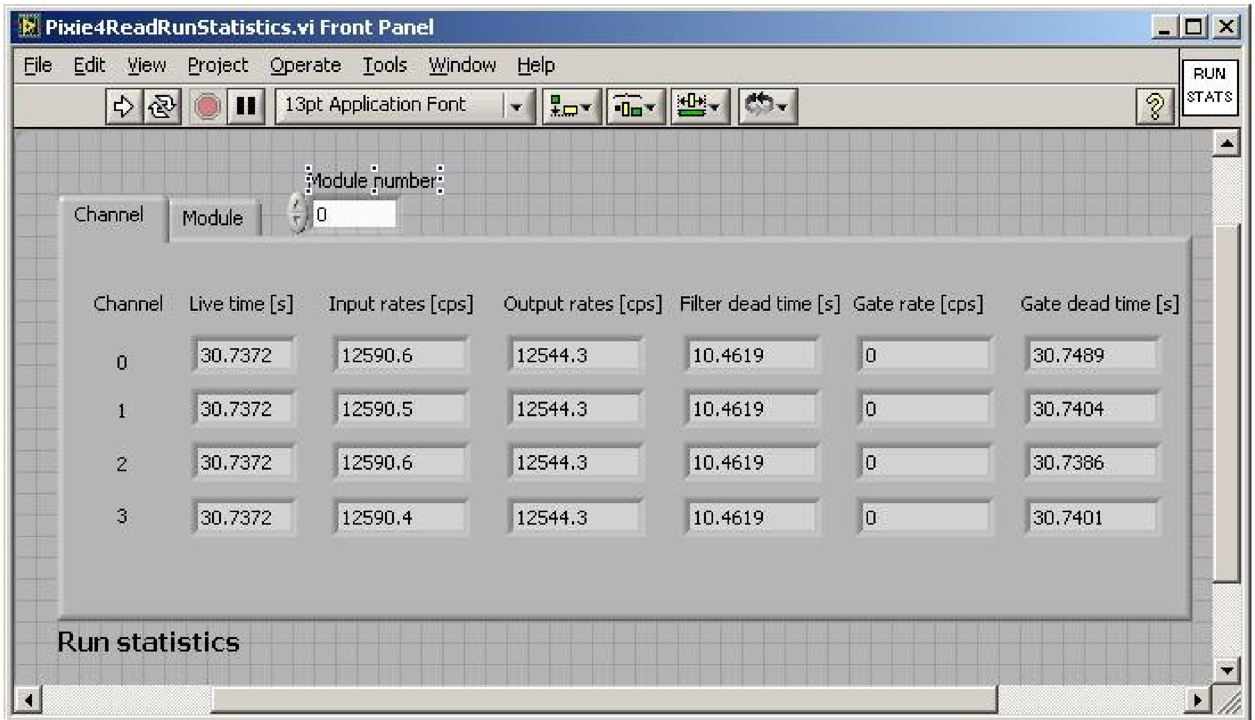

Figure 3.8: The PIXIE4READRUNSTATISTICS VI

3.6 Optimizing Parameters

Optimization of the Pixie-4's run parameters for best resolution depends on the individual systems and usually requires some degree of experimentation.

3.6.1 Energy Filter Parameters

The main parameter to optimize energy resolution is the energy filter rise time. Generally, longer rise times result in better resolution, but reduce the throughput. Optimization should begin with scanning the rise time through the available range. Try 2µs, 4µs, 8µs, 11.2µs, take a run of 60s or so for each and note changes in energy resolution. Then fine tune the rise time.

The flat top usually needs only small adjustments. For a typical coaxial Ge-detector we suggest to use a flat top of 1.2µs. For a small detector (20% efficiency) a flat top of 0.8µs is a good choice. For larger detectors flat tops of 1.2µs and 1.6µs will be more appropriate. In general the flat top needs to be wide enough to accommodate the longest typical signal rise time from the detector. It then needs to be wider by one filter clock cycle than that minimum, but at least 3 filter clock cycles. Note that a filter clock cycle ranges from 0.026 to 0.853µs, depending on the filter range, so that it is not possible to have a very short flat top together with a very long filter rise time.

3.6.2 Threshold and Trigger Filter Parameters

In general, the trigger threshold should be set as low as possible for best resolution. If too low, the input count rate will increase dramatically and "noise peaks" will appear at the low energy end of the spectrum. If the threshold is too high, especially at high count rates, low energy events below the threshold can pass the pile-up inspector and pile up with larger events. This increases the measured energy and thus leads to exponential tails on the (ideally Gaussian) peaks in the spectrum. Ideally, the threshold should be set such that the noise peaks just disappear.

The settings of the trigger filter have only minor effect on the resolution. However, changing the trigger conditions might have some effect on certain undesirable peak shapes. A longer trigger rise time allows the threshold to be lowered more, since the noise is averaged over longer periods. This can help to remove tails on the peaks. A long trigger flat top will help to trigger better on slow rising pulses and thus result in a sharper cut off at the threshold in the spectrum.

3.6.3 Decay Time

The preamplifier decay time τ is used to correct the energy of a pulse sitting on the falling slope of a previous pulse. The calculations assume a simple exponential decay with one decay constant. A precise value of τ is especially important at high count rates where pulses overlap more frequently. If τ is sub-optimal peaks in the spectrum will broaden and if τ is far from the true value the spectrum will be significantly blurred.

Fine tuning of τ can be achieved by exploring small variations around the original estimated value (±2-5%). This is best done at high count rates, as the effect on the resolution is more pronounced. The value of τ found through this way is also valid for low count rates. Manually enter τ , take a short run, and note the value of τ that gives the best resolution.