GemView, The UTA GEM DAQ Program

UTA-HEP/LC-0015

Soomin Kim
University of Texas, Arlington, TX 76016

August 29, 2006

Abstract

This document describes UTA DAQ program, GemView, written with Labview software toolkit. This document contains the description of the software organization and the manual for using GemView program. Since GemView works with ADLINK 2208 ADC card, it can read and process up to 96 individual channels. While the specified maximum data taking rate of the ADC card is at 3MHz, it has been determined that the optimal rate is the total rate of 2MHz. GemView takes data at a specified rate that does not exceed 2MHz and writes out the data into data files. An additional program is provided to read in already existing data from a file and plot it in a display screen.

1. Introduction

UTA HEP group is preparing to expose a 30cmx30cm double GEM chamber into particle beams at Fermi National Accelerator Laboratory in December, 2006. In order to accomplish this data taking with multiple channel readout, it was decided to use a DAQ program using the Labview toolkit. This DAQ program can also be used for general detector development and cosmic ray testing. This program is designed to utilize the 96 channel ADLINK 2208 ADC card. Labview

2. Design Requirements of GemView

GemView DAQ program was designed to meet the following requirements:

1. It must be able to read 96 channels simultaneously.
2. It must be able to write out the sampled peak values into the data file.
3. It must be able to sample a peak values every channel in every event.
4. It must be able to plot the peak values into one-dimensional histograms for all channels.
5. It must allow users to control several run parameters; including the total number of events, data recording to disk and
6. It must be able to display pulse shapes as if it were an oscilloscope.
7. It must be able to utilize the virtual instrument (VI) modules that are provided with ADLINK 2208 ADC card.

3. Organization and Design of GemView

GemView consists of one main virtual instrument (VI) and four sub-VI's, as shown in Fig. 1. The four different color codes represent sub-VI's; DAQ card control VI and histograms (light blue), filename search (blue-purple), data recording (grey), and check event (green). These sub-VI's then consists of the following 8 virtual instruments:

- *Chart* for waveforms
- *Graph* for histogramming
- *File generate* for output data files
- *Peak detect* for ADC peak sampling

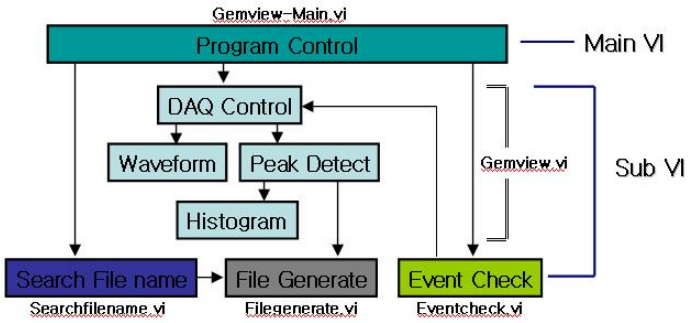


Figure 1 GemView block diagram. GemView consists of one main VI and four sub-VI's which themselves consist of many individual VI's.

- *DAQ control* for DAQ card control interface
- *Program control* for start, stop and other DAQ program control functions
- *Event check* for compare "Number of events per run" with the actual number of events to stop the running.
- *Search file name* for automatic determination of data filenames.

In the following subsections, each of the above function block diagrams are explained:

3.1 Program control: Gemview program control consists of five ***Event Structure Objects***; Start Run, Recording Toggle Switch, Channel Display Selection, Run Parameter Setup Selection and Stop Run. These objects wait until the relevant computational event

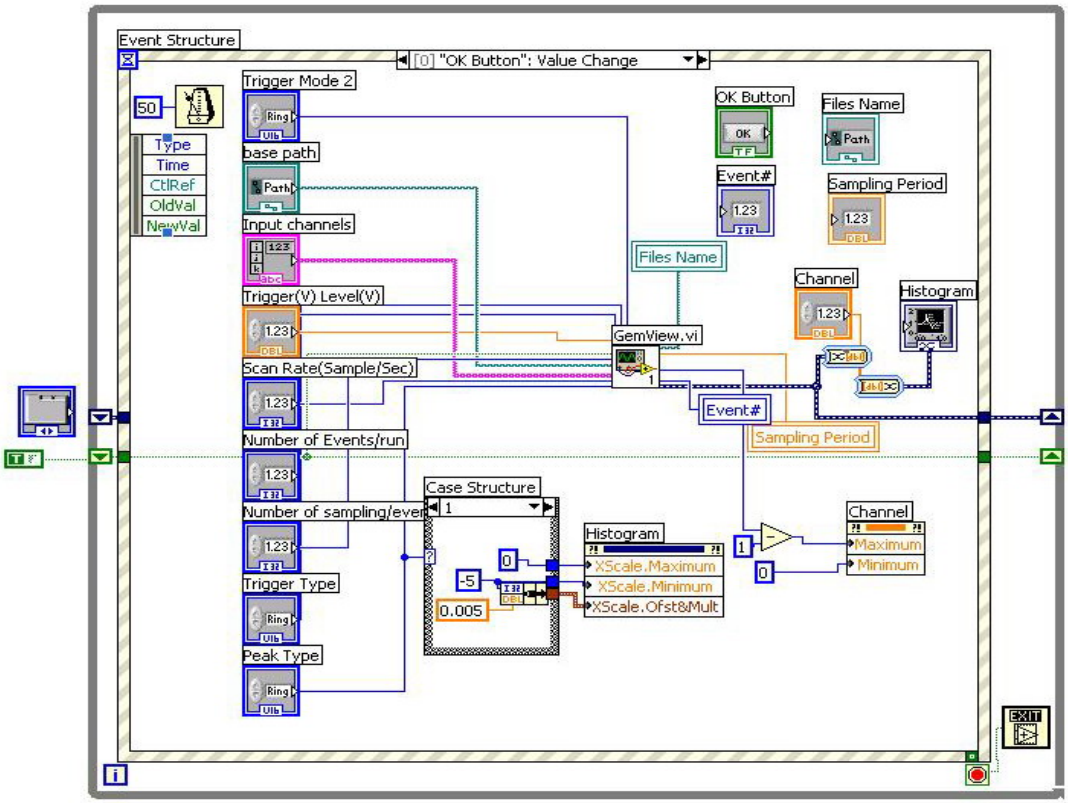
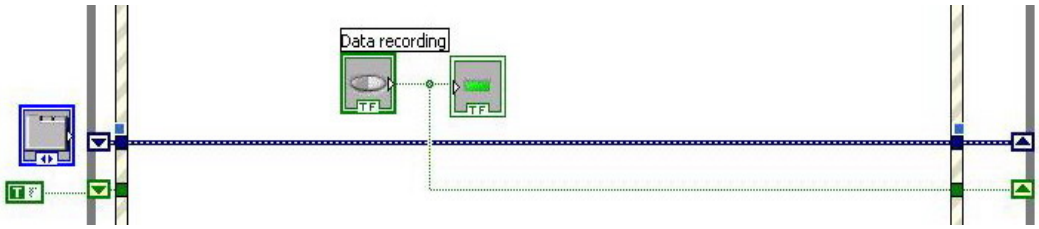


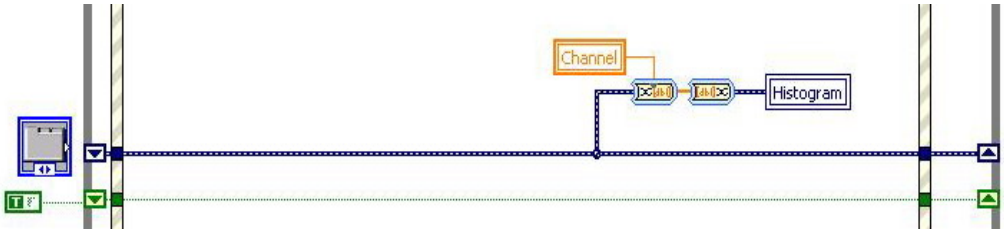
Figure 2. Event structure in an event where the "Run" button is pushed This shows various input parameter objects, program control objects and output indicator objects as block diagrams.

occurs and takes the appropriate actins for the given event.

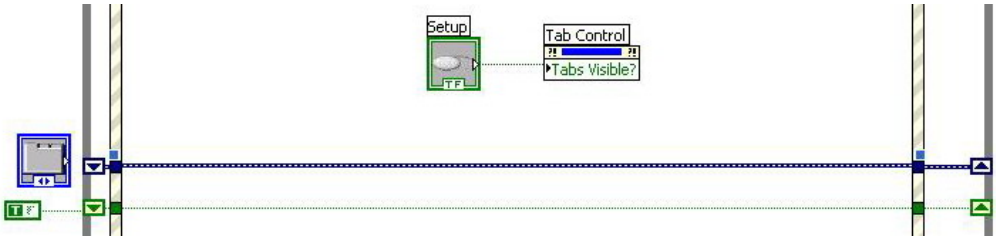
- **Start Run Event:** When the ***Run*** button is pushed after setting run control parameters, the ***Event Structure*** object calls the Gemview.vi which actually takes data and displays histograms and waveforms and passes all the input parameters determined before the start of the execution. This event structure is shown in Fig. 2.
- **Recording Toggle Switch**: When the ***Data Recording*** button is toggled on or off, the ***File Generate*** object is activated or deactivated. This event structure is shown in the figure below.



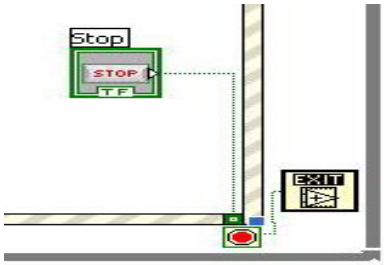
- **Channel Display Selection**: When the value of the ***Channel*** object changes, the histogram of the selected channel is displayed, as shown in the figure below.



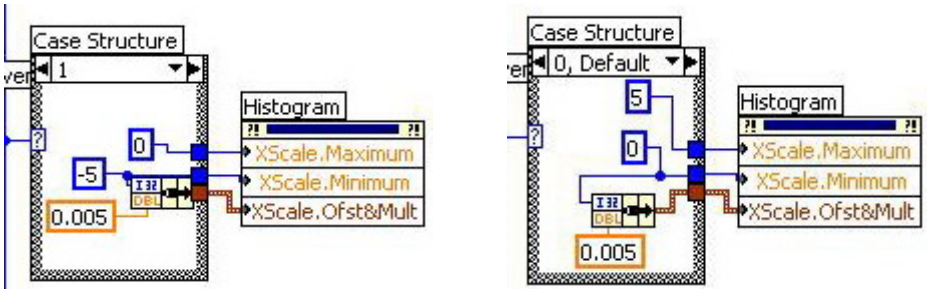
- **Run Parameter Setup Selection**: When the value of the ***Setup*** input parameter object changes by clicking on the setup button, the setup menu tab appears. This event structure is shown below.



- **Stop Run**: When the ***Stop*** button is pushed, this object stops Gemview operation. This event structure is shown in the figure below.



- ***Histogram Case Structure Objects*** is a part of the Event Structure Object. Depending on the selection of the input parameter Peak Type, either positive or negative histogram case is selected, and the histogram lower and upper bound values which have been predetermined in the setup are passed to the histogram display object.



3.2 DAQ Control Object: This object controls and interfaces to the ADLINK DAQ Card, as shown in the figure below. This object receives input parameters from the Program Control Object.

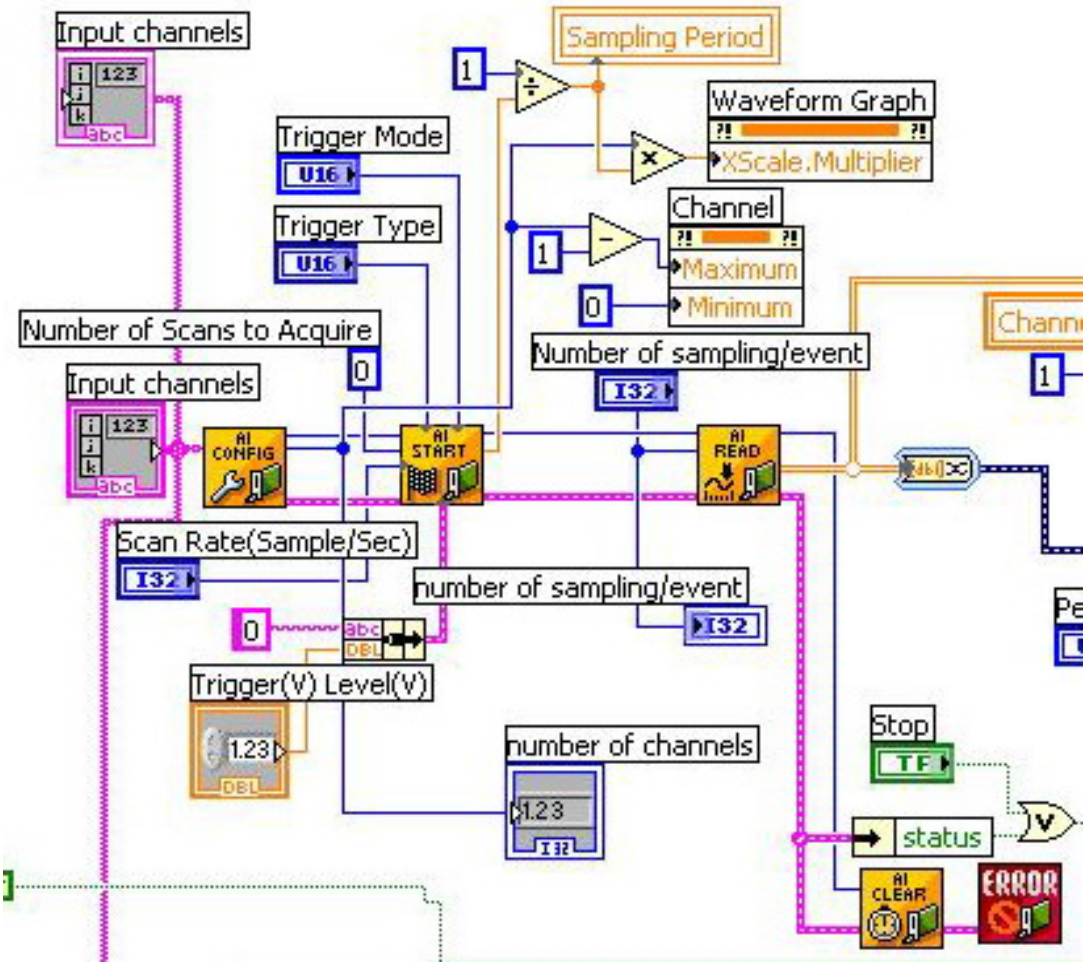
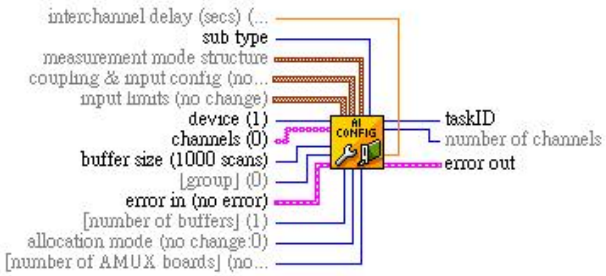
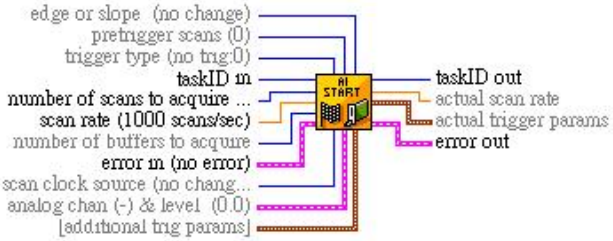


Figure 3. DAQ Control Object portion of Gemview.vi.

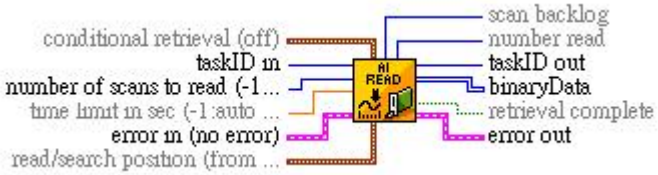
- **Analog Input (AI) Config VI**: This is a VI provided by ADLINK. This VI controls the buffered analog input operation, including ADC card hardware configuration and card internal buffer allocation.



- **AI Start VI**: Starts the buffered analog input operation. This VI sets the ADC sampling rate, the number of scans to acquire per event, ADC clock source, and the trigger conditions and starts data acquisitions.

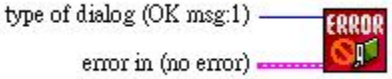


- **AI Read VI**: This VI reads the data from the buffered analog input acquisition by the specified number of scans.

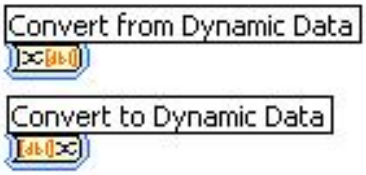


- **AI Clear VI**: This VI stops the data acquisition operation. Before beginning a new acquisition, the AI Config VI must be called to reinitialize.

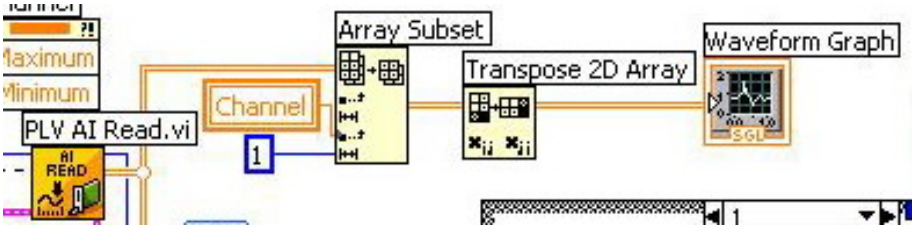
- **ERROR Handler VI**: This VI displays a dialog box with information about the in case of error.



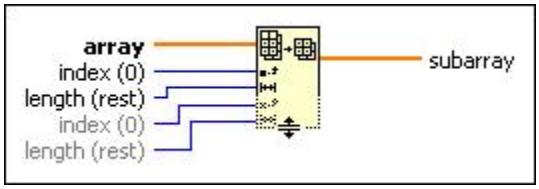
- Convert to Dynamic Data : Converts numeric, Boolean, waveform and array data types to the dynamic data type for use with Express VIs[1].
- Convert from Dynamic Data : Converts the dynamic data type to numeric, Boolean, waveform, and array data types for use with other VIs and functions.



3.3 Chart for waveform: This function displays waveforms of the input pulse from all channels, one at a time. It extracts one-dimension data from the output of DAQ. And transpose the data, after that display it.



- Array Subset: Returns one-dimension at index what “Channel” selected.

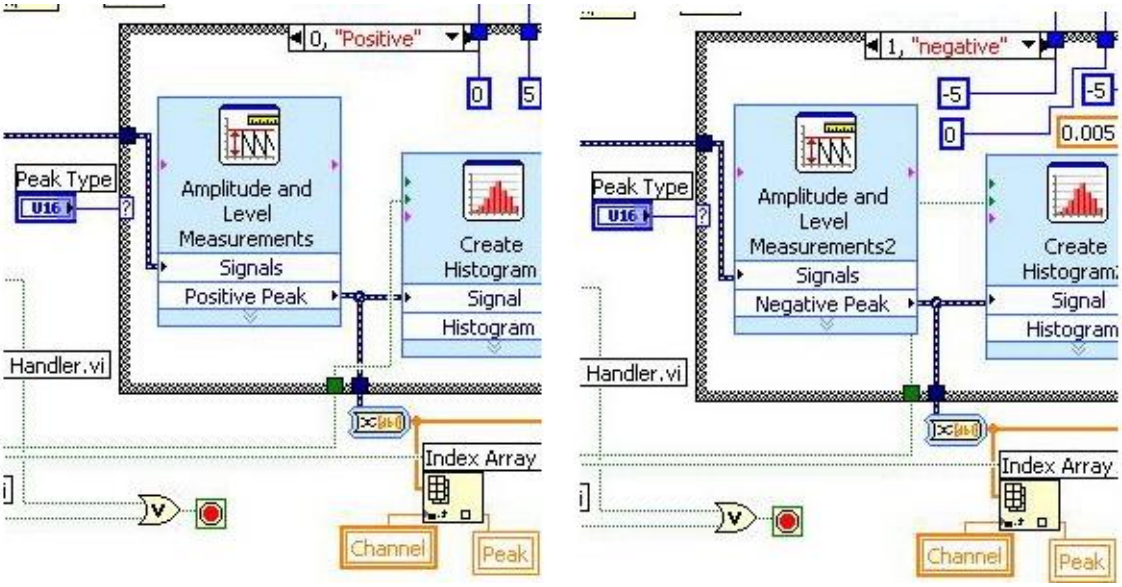


- Transpose 2D array : Rearranges the elements of **array** such that one column **array**[1,i] becomes **transposed array**[i,1].

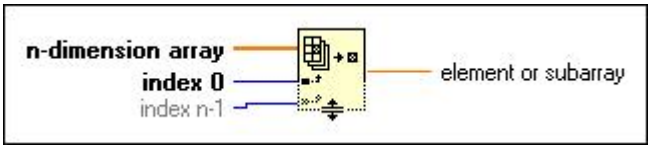
[1] Express VIs for common measurement tasks. Express VIs are nodes that require minimal wiring because you configure them with dialog boxes. The inputs and outputs for the express VI depend on how you configure the VI.

3.4 Peak detect: This function provides the detection of the peak from input signals. It has two cases according as "peak type". It connects "Amplitude and Level Measurements" block. Then it has one-dimension data of peak.

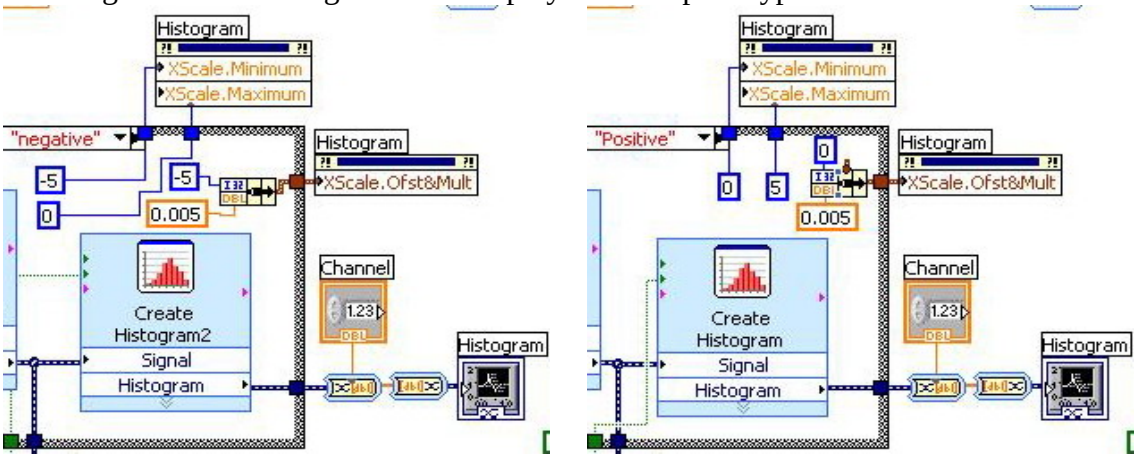
- Amplitude and Level Measurements : Measures the most positive or negative peak in Signals. It has two cases was controlled by "peak type" values.



- Index Array : Returns the peak what "Channel" selected.



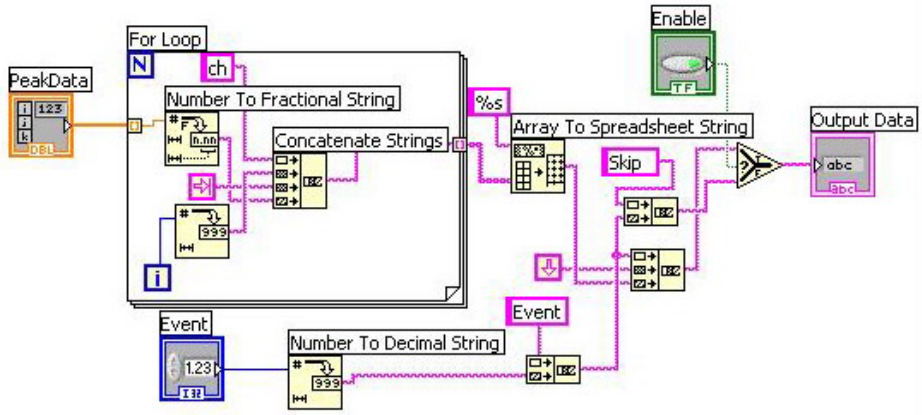
3.5 Graph for histogram : This function shows peak values in one-dimensional histograms. This histogram was setup by case of "peak type".



- **Create Histogram Express VI** : Compute a histogram for signal. It has two cases was controlled by "Peak type" values.

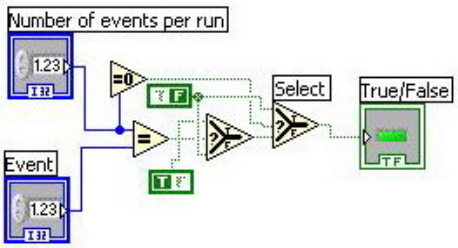
- Histogram : It is showing the histogram at one selected channel. And the parameters about this function was controlled by “Peak type” values.

3.6 File generate : It writes out sampled peak value from every channel into a data file. It connects peak data into the loop {ch+[i]+[tab]+peak data[i], it loops while size of peakdata array.} After that, it transfer into string. And connects it where after “Event##”.

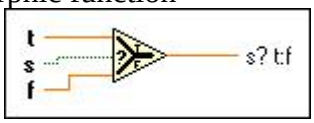


- For Loop : Executes its subdiagram n times, where *n* is the value wired to the count (**N**) terminal. The iteration (**i**) terminal provides the current loop iteration count, which ranges from 0 to *n*-1.
- Number To Fractional String : Converts **number** to an F-format (fractional notation), floating-point string at least **width** characters wide or wider if necessary.
- Concatenate Strings : Concatenates input strings and 1D arrays of strings into a single output string. For array inputs, this function concatenates each element of the array
- Number to Decimal String : Converts **number** to a string of decimal digits at least **width** characters wide or wider if necessary.
- Array To Spreadsheet String : Converts an array of any dimension to a table in string form, containing tabs separating column elements, a platform-dependent EOL character separating rows, and, for arrays of three or more dimensions, headers separating pages.

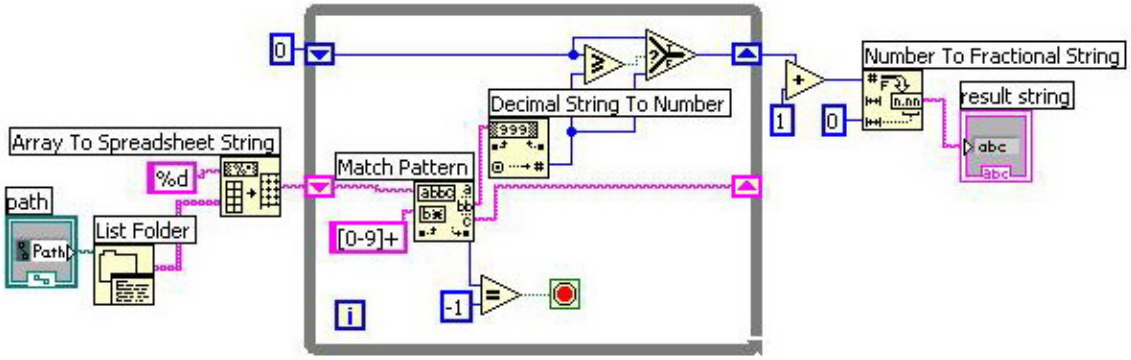
3.7 Event check: It compares “Number of events per run” with the actual number of events taken through the run and issues a bullion value. This function forces GemView to continue taking data till the total number of events is the same as the set parameter as the initialization stage.



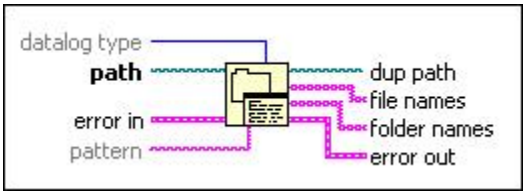
- Select : Returns the value wired to the **t** input or **f** input, depending on the value of **s**. If **s** is TRUE, this function returns the value wired to **t**. If **s** is FALSE, this function returns the value wired to **f**. The connector pane displays the default data types for this polymorphic function



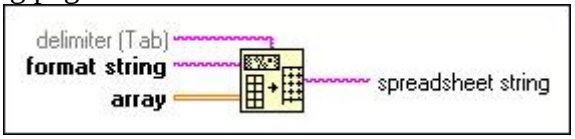
3.8 Search filename: It finds the last run within the given subdirectory where the data files are written and determines the output data filename for the new run.



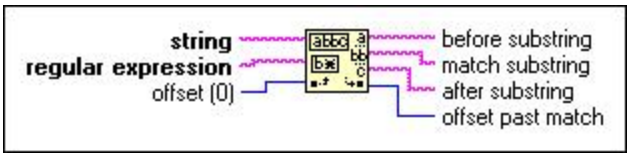
- List Folder : Returns two arrays of strings listing the names of all files and folders found in **path.**



- Array To Spreadsheet String : Converts an array of any dimension to a table in string form, containing tabs separating column elements, a platform-dependent EOL character separating rows, and, for arrays of three or more dimensions, headers separating pages.



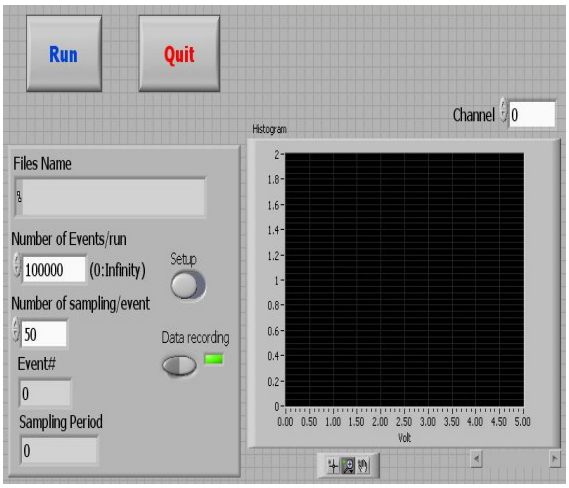
- Match Pattern : Searches for **regular expression** in **string** beginning at **offset**, and if it finds a match, splits **string** into three substrings.

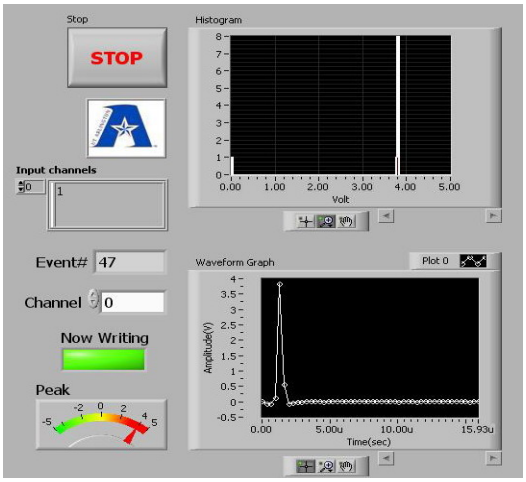


- Decimal String to Number : Converts the numeric characters in **string**, starting at **offset**, to a decimal integer and returns it in **number**.
- Number To Fractional String : Converts **number** to an F-format (fractional notation), floating-point string at least **width** characters wide or wider if necessary.

4. Feature & Gemview Enviroment

- **Feature**





  - **Logo :** It is about the UTA logo.
  - **Control Panel**
  - **Histogram**
  - **Waveform**
  - **Button :** Run, Quit, and Stop Button.
  - **Channel select button**
- **Gemview Enviroment : It needs ADLINK ADC card, D2K-DASK driver, and D2K-LVIEW driver(It is included in ADLINK Driver CD).**

5. Output Data File Format

Output Data file consist of [Channel gain queue, Event, channel number, value of peak].

Channel Gain Queue
1,2,3,4
Event0
ch0 0.000000 ch1 -0.004883 ch2 0.004883 ch3 -0.004883
Event1
ch0 1.972656 ch1 1.987305 ch2 1.972656 ch3 1.972656

This file contains peak values from four channels starting from channel 1. However, the file shows the channel starting from 0. This offset needs to be resolved.

6. Conclusions

A DAQ data taking program and a separate data plotter have been written using Labview tool kit to interface to the ADLINK PCI DAQ card. The data taker satisfies all six requirements laid out in section 1 above. However given the shortage of time, there are rooms for improvements in the following areas:

- The gain factor of 2 is applied to the peak values.
- The sampled peak values are not always correct.
- There is an offset for one channel in actual input channel and the selected channels.

It is my hope that these issues get resolved quickly.

7. Bibliography

[1] Labview fundamental Manual
[2] DAQ-2208 User Guide
[3] Help desk from Labview menu bar.("nudaqpci" provides context about daq-2208)
[4] http://zone.ni.com/zone/jsp/zone.jsp