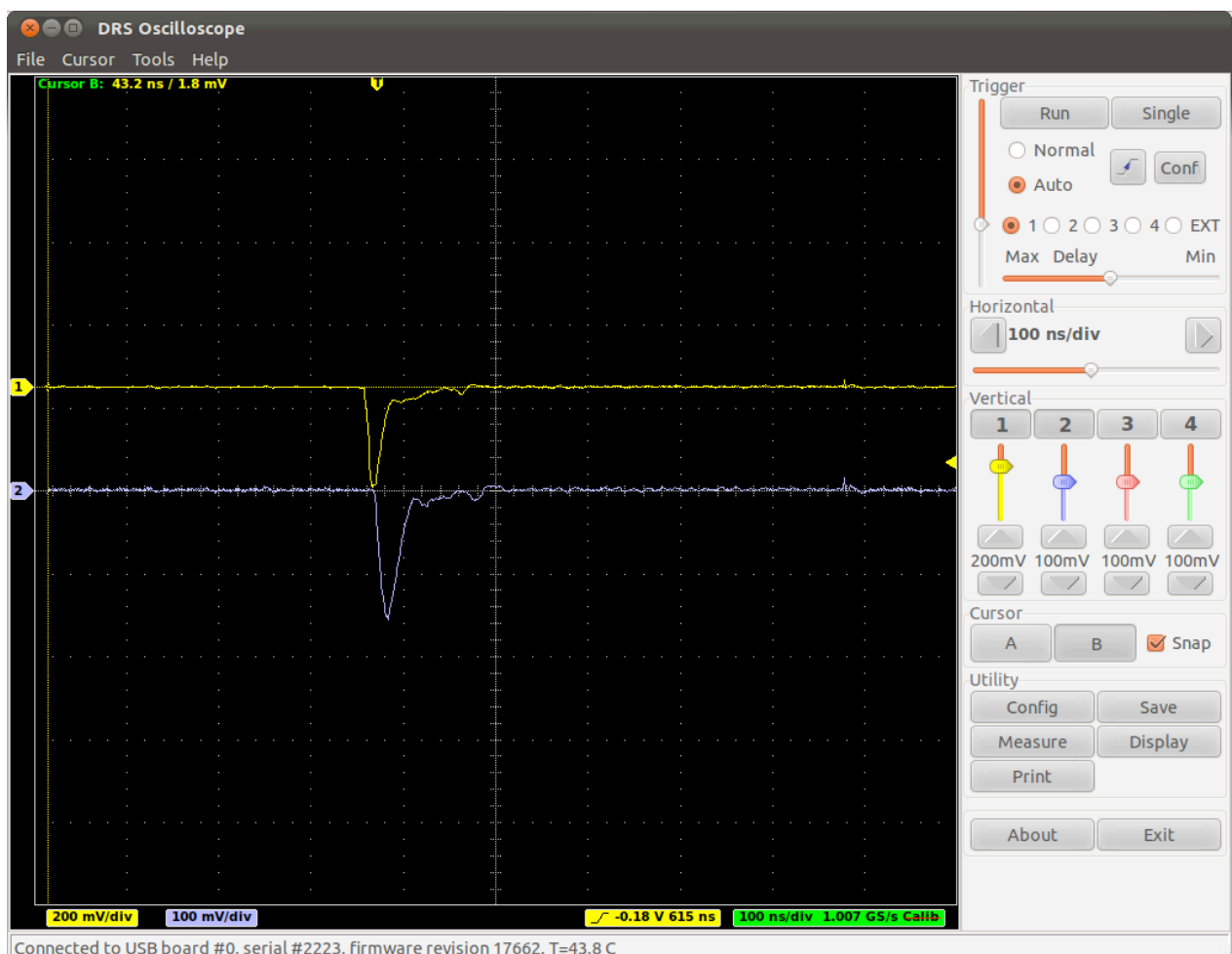
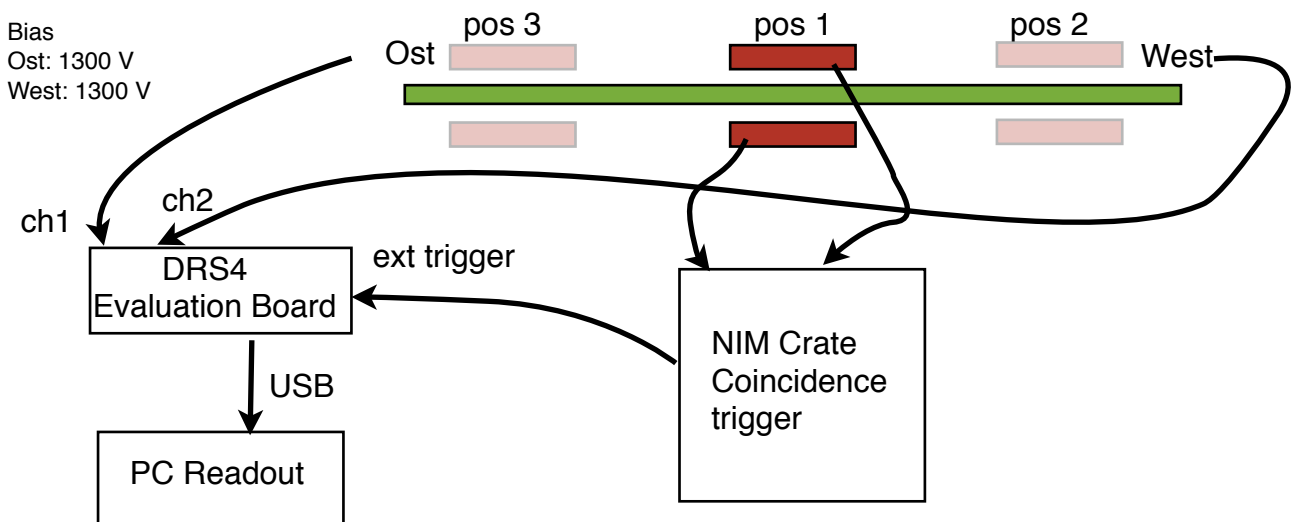


Section 1. Setup

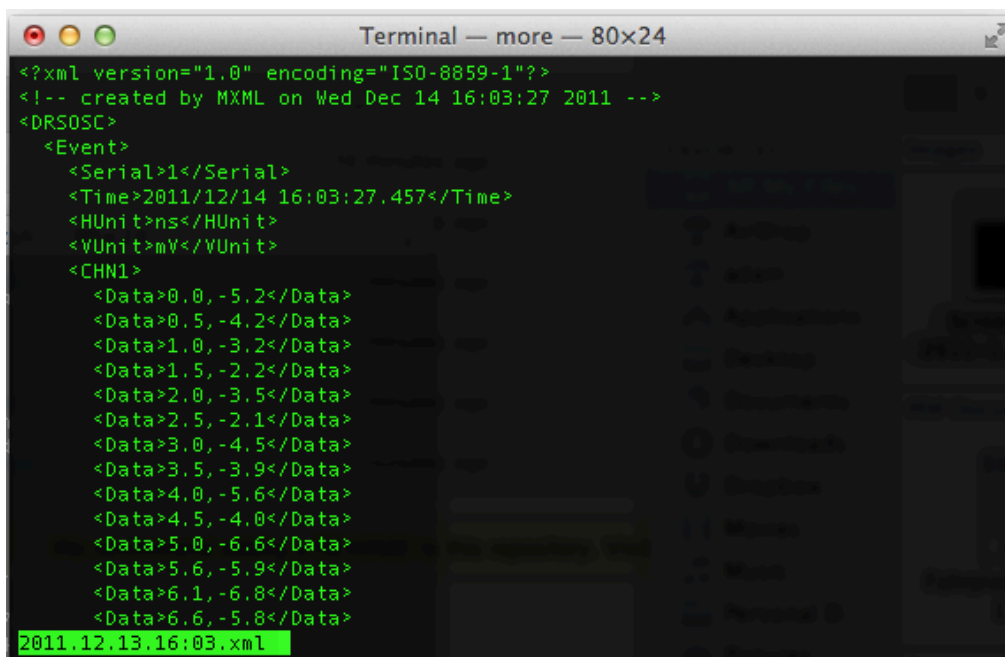
The test bench set up at KIT includes a ~3-m plastic scintillator “Edelweiss” panel and two smaller scintillator panels to act as a coincidence trigger. At each end of the Edelweiss panel are two PMT groups, labeled Ost and West, each with a summed signal output that are recorded by the DRS4.



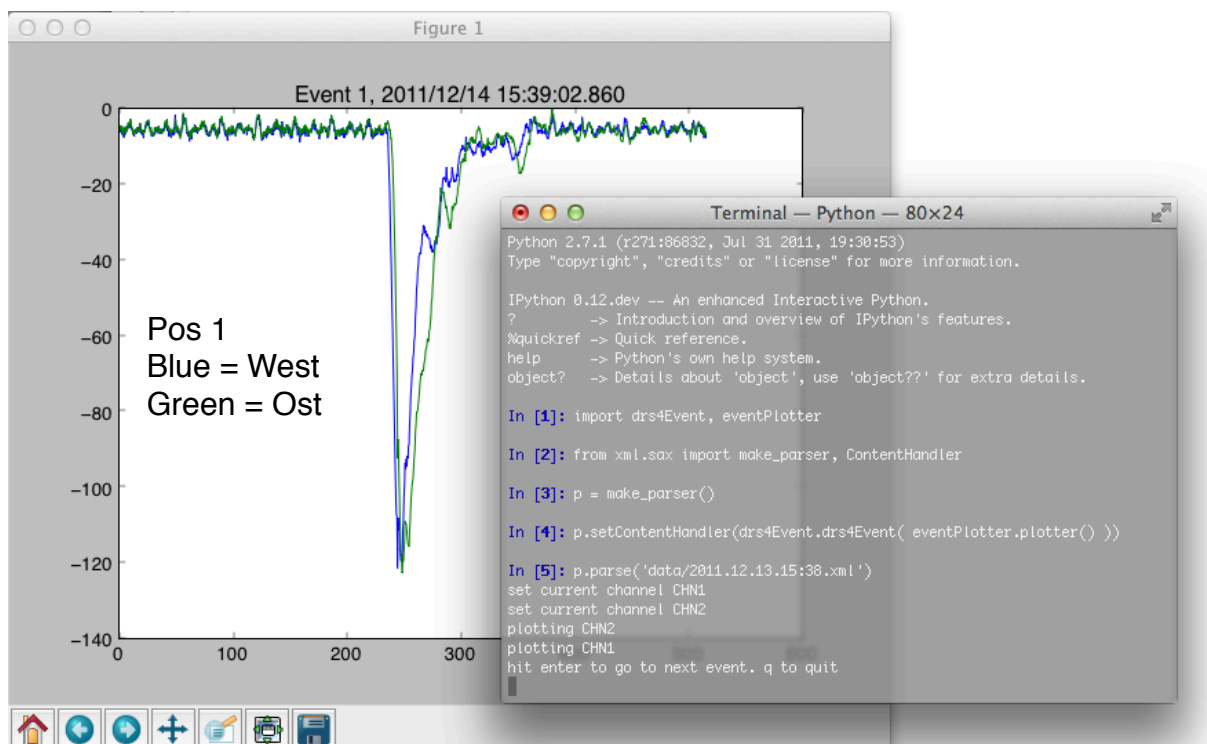
The PC readout software was supplied by the PSI group that produces the DRS4 chip. This allows one to view the output of the DRS4 in “oscilloscope” mode and to save the triggered data to disk in a binary or XML format. While XML data files are significantly larger than binary data files, they are programmatically easier to read. Some Python code was written to read these XML data. They are currently located at the following public repository: <https://github.com/gadamc/drs4>

The image above is a screenshot of the PSI group’s software. The software was installed on a PC running the Ubuntu 11.10 linux distribution.

The next screenshot shows the data in XML format.



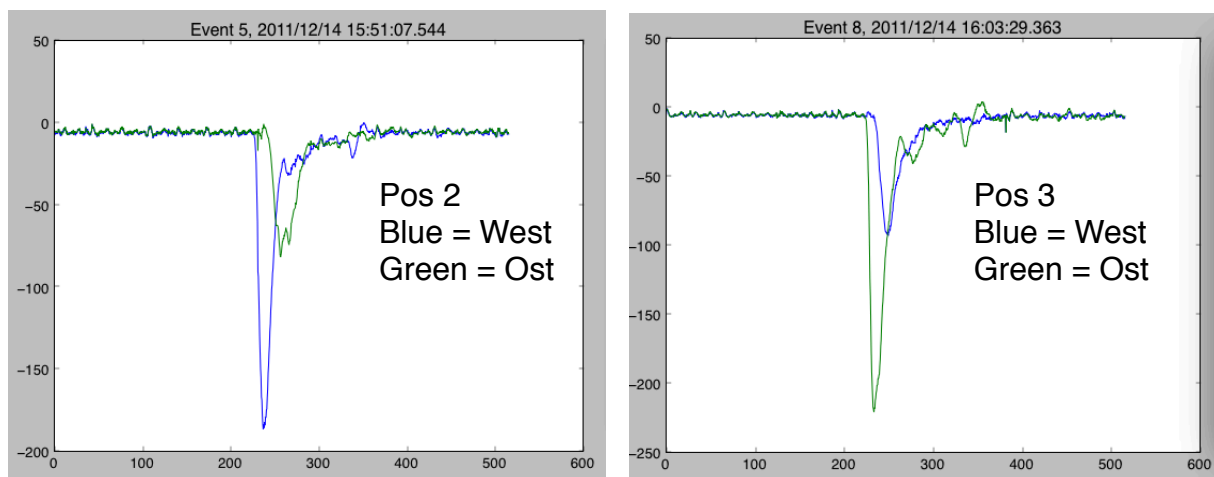
```
Terminal — more — 80x24
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- created by MXML on Wed Dec 14 16:03:27 2011 -->
<DRSOSC>
  <Event>
    <Serial>1</Serial>
    <Time>2011/12/14 16:03:27.457</Time>
    <HUnit>ns</HUnit>
    <VUnit>mV</VUnit>
    <CHN1>
      <Data>0.0,-5.2</Data>
      <Data>0.5,-4.2</Data>
      <Data>1.0,-3.2</Data>
      <Data>1.5,-2.2</Data>
      <Data>2.0,-3.5</Data>
      <Data>2.5,-2.1</Data>
      <Data>3.0,-4.5</Data>
      <Data>3.5,-3.9</Data>
      <Data>4.0,-5.6</Data>
      <Data>4.5,-4.0</Data>
      <Data>5.0,-6.6</Data>
      <Data>5.6,-5.9</Data>
      <Data>6.1,-6.8</Data>
      <Data>6.6,-5.8</Data>
    </CHN1>
  </Event>
</DRSOSC>
2011.12.13.16:03.xml
```



Using the Python code, one can scan through the data, event by event. Once you've downloaded the code, open a terminal in the directory where the code is located. Start a Python shell and issue the following commands:

```
In [1]: import drs4Event, eventPlotter  
  
In [2]: from xml.sax import make_parser  
  
In [3]: p = make_parser()  
  
In [4]: p.setContentHandler(drs4Event.drs4Event( eventPlotter.plotter() ))  
  
In [5]: p.parse('/path/to/data//2011.12.13.15:38.xml')
```

Data was taken with the coincidence scintillators placed in three different locations about the Edelweiss scintillator. The position and amplitude of the pulse peaks shift in time accordingly.



In addition to displaying the pulses, code was written to analyze each event with a trapezoidal filter to estimate the pulse position and amplitude. This is performed by the `drs4Analyzer` class found in the `dataAnalysis.py` file.

Current Problems:

Some events are readout properly by the XML Python code. Those events are skipped in the analysis.

The sampling may not be a rational number and so in the data the value for the time of each event does not appear to be consistently spaced. However, I believe that this is most likely a rounding error. I assume in the analysis that the data is equally spaced in time.