Multivent User Guide

Prepared By: Peter Milne

Date: 2010 -09-21

Rev	***Date***	***Description***
1.0	!5 Aug 2010	initial
2.0	21 Sept 2010	Update to reference MULTIDTACQ

Table of Contents

Copyright and Attribution.

Document created using OpenOffice.Org www.openoffice.org.

This document and D-TACQ Software comprising platform Linux port, Linux kernel modules and most applications are released under GNU GPL/FDL:

Document:

Software:

1 Introduction

1.1 Summary

ACQ132 is a 32 channel intelligent simultaneous digitizer. It is capable of continuously capturing data at high rates to internal memory. For a host computer to handle all this data either after a shot or continuously presents a significant problem.

Multivent is an operational mode that allows multiple events to be extracted and saved from a long-running data stream. The mode allows capture over long periods, and data is presented in a way that is directly accessible (and plottable) on a host computer. The host computer can be a standard *MS-Windows* PC, with no additional software loaded other than a suitable plot program; *Kst* [1.5.3] an open source, no-cost plot program is recommended. Critically, no device driver software is required at all, making the system zero-install, maximum portability.

1.2 Intended Audience

Users, potential users.

1.3 Scope

This document covers a simple setup scenario and discusses data handling.

It's assumed that the user is already able to execute commands on the *ACQ132*, using any of the techniques outlined in [1.5.1].

1.4 Glossary

ACQ132 : 32 x 2MHz Intelligent Simultaneous Digitizer *Ref*

MSPS : Mega Sample Per Second (per channel)

MS-Windows : generic term for Microsoft(R) desktop os eg Win XP,

FPGA : Field Programmable Gate Array

ES : Event Signature : event position marker embedded in data.

TRG : Front panel Trigger Input [2]

TBLOCKS : 6MB block subdividing capture buffer.

DirFile : A simple, self describing binary data format.

Samba : Free Software implementation of *MS-Windows* file server.

1.5 References

1.5.1 2GUG – D-TACQ 2G User Guide

1.5.2 ACQxxx Hardware Installation Guide

1.5.3 Kst – a cross-platform, high performance plot tool. Plots DirFile data.

1.5.4 7-zip – MS-Windows tool handles tar, tgz archive files.

1.5.5 Cygwin – additional command environment for MS-Windows, includes ssh client.

1.5.6 Dt100rc – D-TACQ GUI capture tool.

1.5.7 MultiDtacq **–** Controls multiple ACQ cards in a fixed scenario

1.6 Notation

- **`command`** : indicates name of a program (command)
- `preformatted text` : literal input or output from terminal session.
- *Defined Term* : some term or acronym specific to this domain (perhaps referenced in the glossary)

2 Summary of Requirement

2.1 Overview

A seismic application requires continuous data capture at 1 *MSPS* for some 10's of seconds. The duration of the experiment is longer than the available capture memory for a simple one-shot transient (5s). The experiment is characterized by a sequence of randomly timed events, and only the data immediately before and after the event is required to be saved at high speed., although it is also required to log data at low speed for trending. The events are limited in number, with a low average rate, although events can be back to back.

The data must be made available to an *MS-Windows* system with minimum programming required.

2.2 Parameters

1. Number of channels 32 or more
2. Output sample rate: 1MHz
3. Duration of experiment: 600s
4. Maximum number of events: 100.

5. Samples to store both pre- and post- event : 10000
6. Logging rate : 10Hz, nominal

3 The Solution: Multivent

Multivent is an operating mode that allows continuous capture with multiple events.

ACQ132 is set to capture data continuously to the local 1GB memory capture buffer.

For processing convenience, the capture buffer is divided into a series *TBLOCKS*, each 6MB long.

The *FPGA* hardware features an event capture mechanism that inserts an Event Signature *ES* into the data stream at the point of event occurrence. Downstream software can reconstruct the exact point of the event by locating and decoding the ES. An event is a pre-defined edged on a digital input line. Typically this is a falling edge on the front-panel *TRG* input.

The Event capture mechanism is traditionally used in the single-event "`triggeredContinuousMode`" [1]. Capture starts off continuously overwriting the capture buffer. When an event is detected, software allows the capture to continue for a pre-determined number of (post-) samples. Then the shot is stopped, software locates the exact point of the *ES*, removes it from the data, and adjusts the application view of the data so that applications see exactly the pre-specified number of *pre*- and *post*- samples.

In *Multivent*, the card is set to *SOFT_CONTINUOUS* mode, but *event0* is also enabled. Now the card will capture data indefinately until capture is aborted. Any number of events may be generated, with the corresponding ES embedded in the data.

Kernel-level firmware provides a virtual device for monitoring events. An application, **`acq_demux-lpp`** blocks on the virtual device, and receives the relevant *TBLOCK* id whenever an event it detected. The *TBLOCK* is marked as BUSY and is not subsequently overwritten as the capture continues. Running at low priority, **`acq_demux-lpp`** locates the *ES* while the capture is still running, and demuxes the required pre-, post- amounts of data surrounding the *ES*. For the convenience of host applications, the demuxed data is extracted into *DirFile* format on a subdirectory of a *ramdisk*. The *ramdisk* is exported as an *MS-Windows* compatible share.

The net result is that standard software running on an *MS-Windows* system is able to plot the captured data around the event directly from the network file share, with no further post processing at all. *D-TACQ* recommends the *KST* program to plot the data.

A further low-priority program, **`acq_demux-ll`** creates a record of the entire data set, subsampled at a very low rate. The output of **`acq_demux-ll`** is presented as in *DirFile* format on the same network file share, and users are able to monitor the data trend in real time without further programming.

This solution works because there is a limited number of events, with a relatively small number of samples required per event. With 1GB capture memory, *ACQ132* has 148 *TBLOCKS*, so in principle the limit to the number of events is close to 148

3.1 No missing data at block ends ..

Each *TBLOCK* contains nearly 100K samples, so there's a 90% chance of a 10K data set fitting in any one *TBLOCK*. The kernel driver compares the event position in the TBLOCK with a threshold. If the event is under the threshold at the beginning of the block, the previous TBLOCK is reserved and presented to the demux process, if the event is over the threshold at the end of the block, the next TBLOCK is presented to the demux process. So the system is able to present data sets specified as 10K pre-, 10K post without missing any data. In other words, for the common case pre==post, the event is always in the middle of the data set. This is 100% effective for up to 20K samples.

The **`acq_demux-lpp`** process is relatively slow (about 5s/*TBLOCK*) while capture is running at 1MHz. The *TBLOCK* is recycled once the demux is complete (since the demuxed data has been copied to *ramdisk*), and so the TBLOCK is returned to pool.

The remote client has write access to the demultiplexed data. So the remote client has the possibility to consume the data, then delete it on the ramdisk, freeing up ramdisk space, giving the possibility that the capture process can run for ever.

Requirement for a system that can run for ever is that the average event rate should not exceed about 1 per 10s. The on-board memory can accomodate up to 100 events at any one time.

4 Instructions for use.

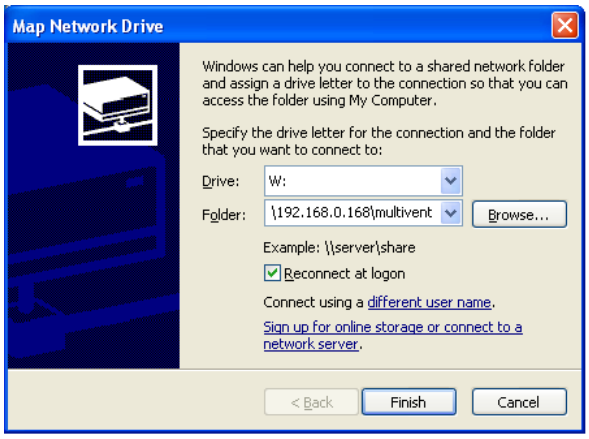
4.1 Pre-requisites

1. *ACQ132* cards have been preconfigured according to [1.5.2].
2. User is able to log into the card to execute commands (using ssh, telnet for initial testing), or programmatically (eg using the WebService interface.) We recommend that the *MS-Windows* host pc should have an ssh client installed (we recommend *cygwin*), and the plot tool *Kst* see [1.5.3].
3. The *ACQ132* cards are setup on the *LAN*, and ip addresses are known.
4. Connect known signals to the card. We used a free running signal in *AI01-16*, and "sampled the trigger" in *AI17*.
5. Connect trigger to front panel *TRG* input
6. Use **`dt100rc`** [1.5.6] to run a regular transient capture at 1MHz to confirm that data is as expected, and the trigger works. Leave **`dt100rc`** connected and on the Capture page to monitor state when running multivent.
7. Use **`MultiDtacq`** [1.5.7] to control a multiple card scenario.
8. Take a look at our data. Load a copy of Kst 2.0, (select "View All Files", then select image to download (eg Kst-2.0.0.exe).

   Download the test data set from here. Extract, eg using 7-zip. [1.5.4] From **`Kst`**, Open| and search for the pre-recorded .kst state file `template4-tmp.kst`. The plot shows two tabs, the first with log data, the second with the first four events.

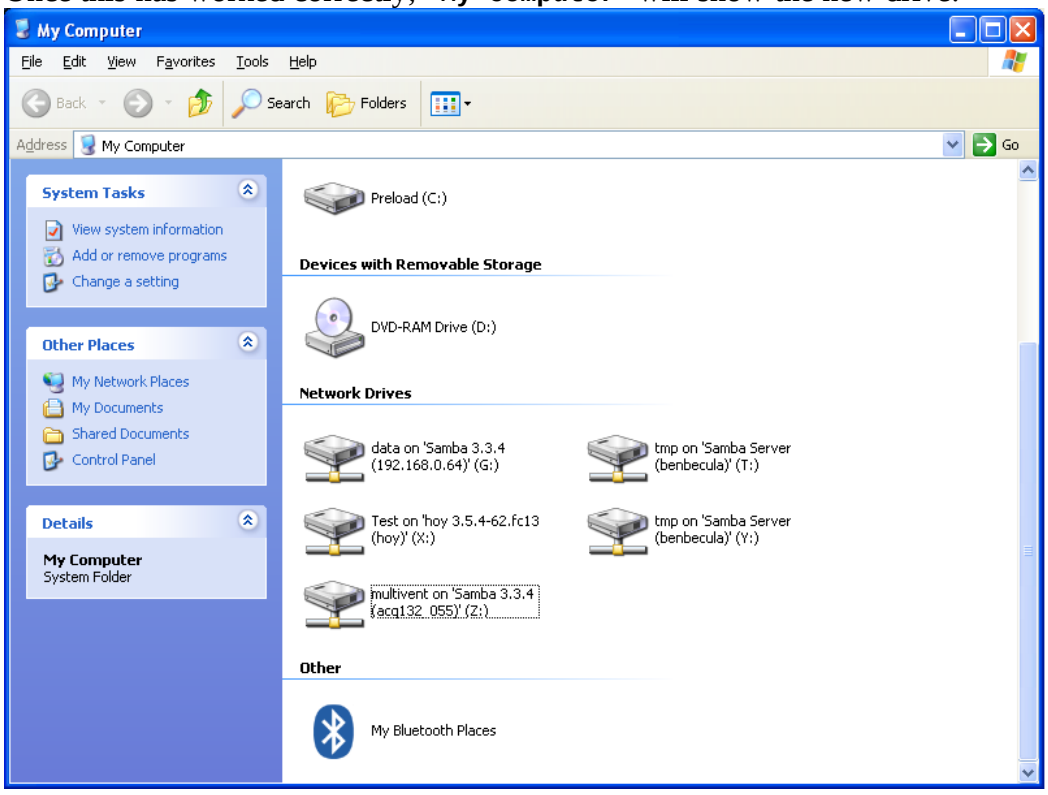
4.2 Mount file shares.

Use "**`My Computer, Map Network Drive`**" to map the file share from each *ACQ132* device to a local drive. At the password dialog, specify User dt100, password the same. It's also possible to access the data anonymously, but then teh host cannot delete

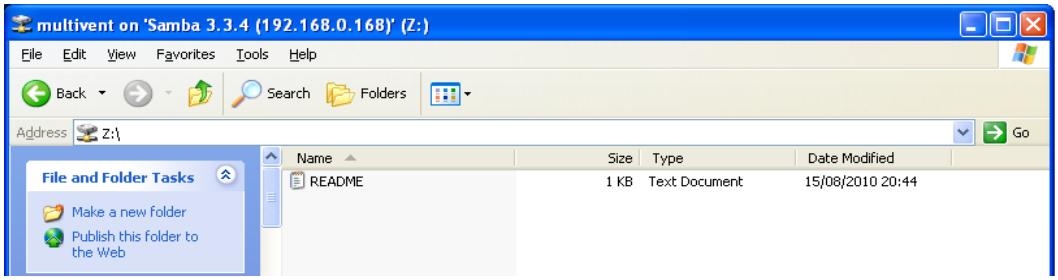


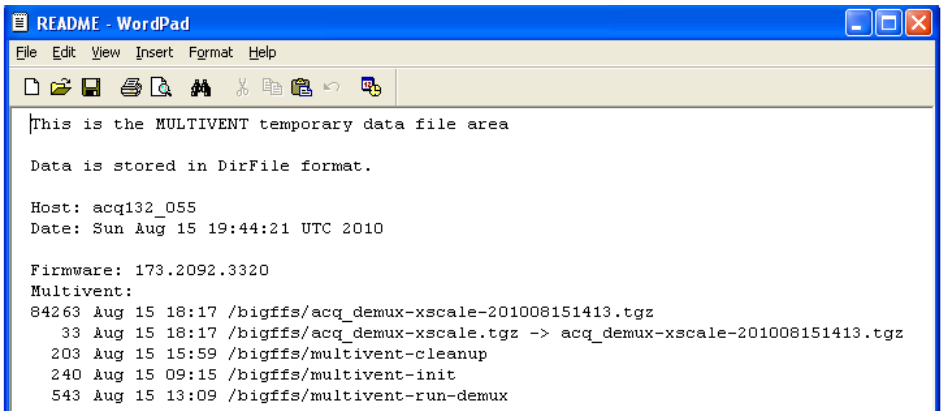
Actual text:
\\192.168.0.168\multivent

Once this has worked correctly, "**`My Computer`**" will show the new drive:



And the initial listing shows one `README` file, open with "**`Wordpad`**" :


multivent on 'Samba 3.3.4 (192.168.0.168)' (Z:)
File Edit View Favorites Tools Help
Back Search Folders
Address Z:\
Go
File and Folder Tasks
Make a new folder
Publish this folder to the Web
Name Size Type Date Modified
README 1 KB Text Document 15/08/2010 20:44


README - WordPad
File Edit View Insert Format Help
This is the MULTIVENT temporary data file area
Data is stored in DirFile format.
Host: acq132_055
Date: Sun Aug 15 19:44:21 UTC 2010
Firmware: 173.2092.3320
Multivent:
84263 Aug 15 18:17 /bigffs/acq_demux-xscale-201008151413.tgz
33 Aug 15 18:17 /bigffs/acq_demux-xscale.tgz -> acq_demux-xscale-201008151413.tgz
203 Aug 15 15:59 /bigffs/multivent-cleanup
240 Aug 15 09:15 /bigffs/multivent-init
543 Aug 15 13:09 /bigffs/multivent-run-demux

4.3 Initialize the card

Use MULTIDTACQ with appropriate scenario set.

The scenario provided is:

`ACQ132-multivent.com.d_tacq.multicon.Multicon`

This scenario requires a trigger on the front-panel TRG input. The TRIGGER, EVENT buttons on MULTIDTACQ simulate edges internally on the ACQ132. So the digitizers will respond to the simulated events, but there will be no event to see in the data For our testing, we used an alternate scenario `ACQ132-multivent.com.d_tacq.multicon.MulticonDIO32` that assumes a DIO32 port looped back to both TRG and AI inputs – sampling the trigger is always a good way to test; Unfortunately DIO32 is not a standard deliverable.

4.4 Trigger

There needs to be an initial edge on *TRG* to start the process.

The MULTIDTACQ scenario includes TRG, EVENT controls, but this assumes a test loopback from the DIO32 port to the LEMO TRG input. This isn't available on shipped systems, in this case a physical external trigger has to be used.

4.5 Check that data is accumulating:

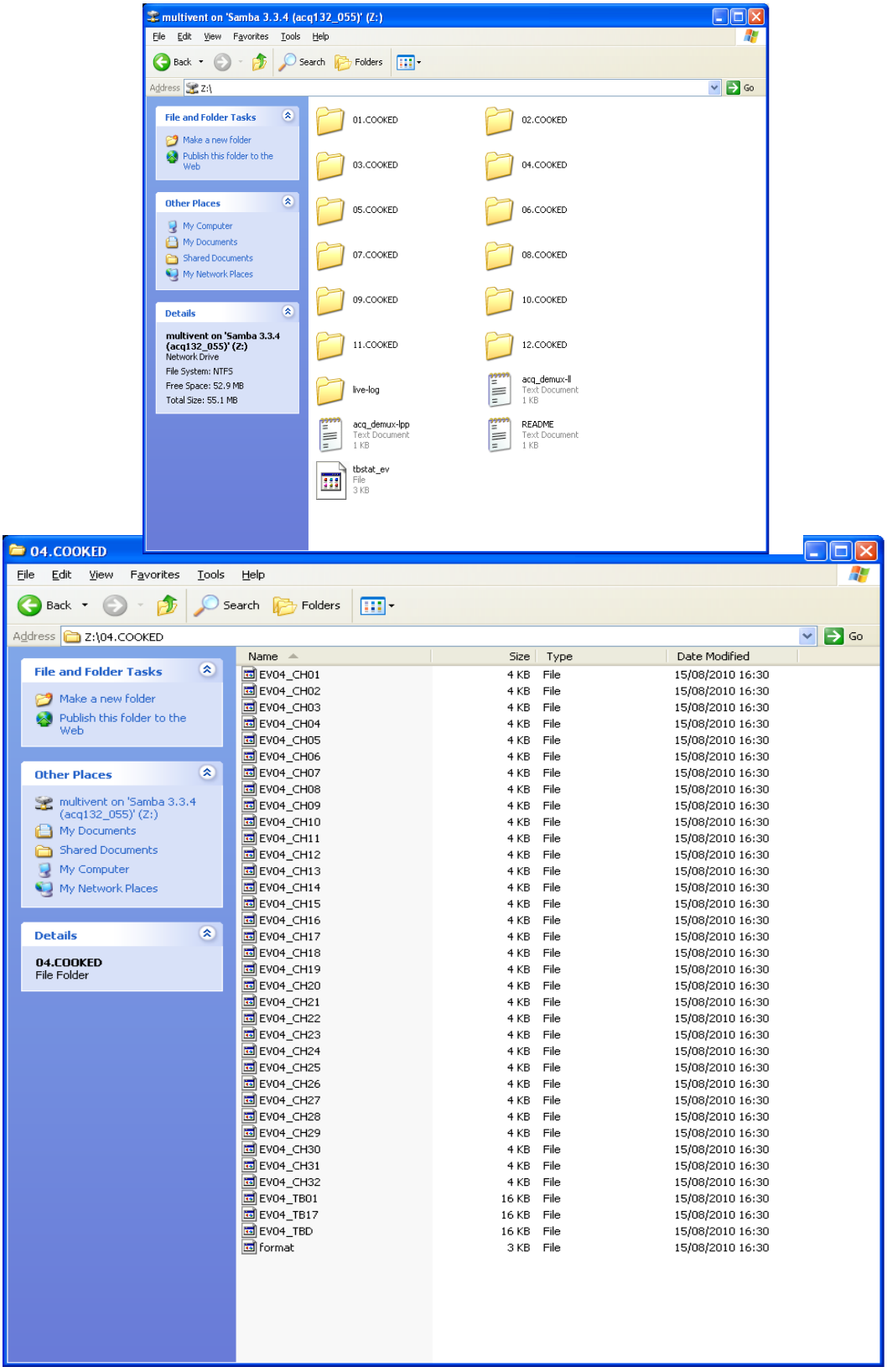
Initial directory listing show the `live-log` subdirectory.

The signal files inside should be gradually increasing in size.

NB: on KST, under MS-Windows, it may be necessary to press the refresh button to get updates.

4.6 After some events, the directory looks like this

Nb: it takes several seconds for the Event directory to show up – the embedded microprocessor is working hard!



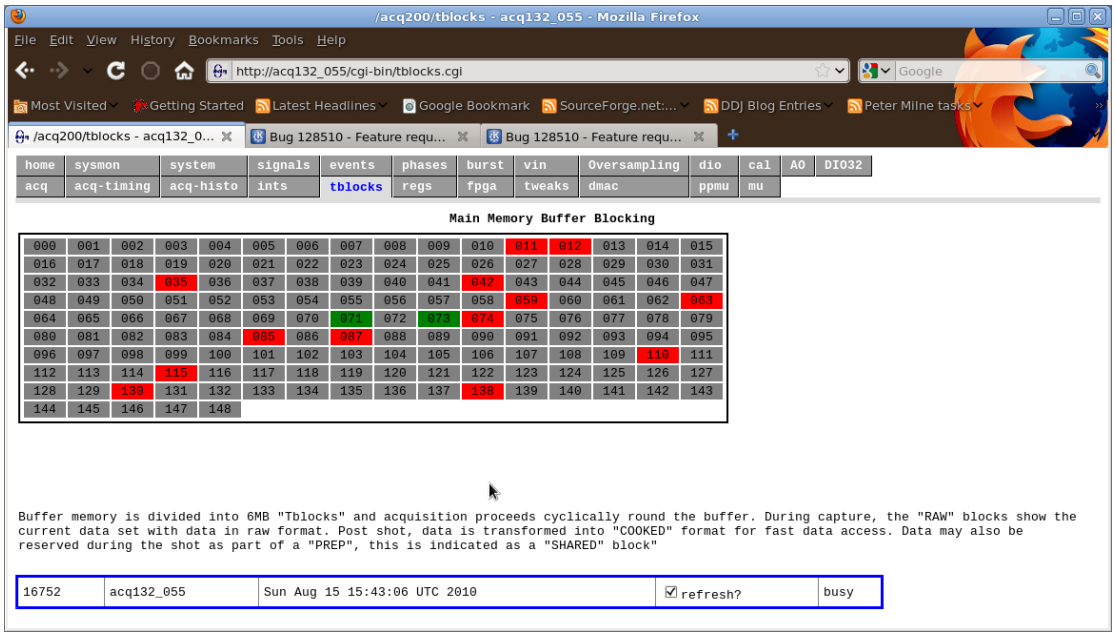
... and drilling down into the Event directory, we see the demuxed data in *DirFile* format, ready to plot.

4.7 Embedded Web Pages.

Leave Firefox monitoring the *TBLOCKS* web page.

During capture, a “snake” of three red blocks traverses the buffer map. These are the current capture *TBLOCKs*.

Each event leaves a *TBLOCK* permanently marked in red. The capture snake continues, avoiding the marked blocks. Clearly if we carried on long enough, with enough events arriving faster than **`acq_demux-lpp`** can deal with them, then we'd run out of blocks. As after each *TBLOCK* has been processed by **`acq_demux-lpp`**, it is returned to the pool, (changes red->grey). This takes about 5..10s depending on system load and event capture length.



4.8 To Stop the shot

Run

`acqcmd setAbort`

Once you have backed up the data, run this command and it's ready to start over:

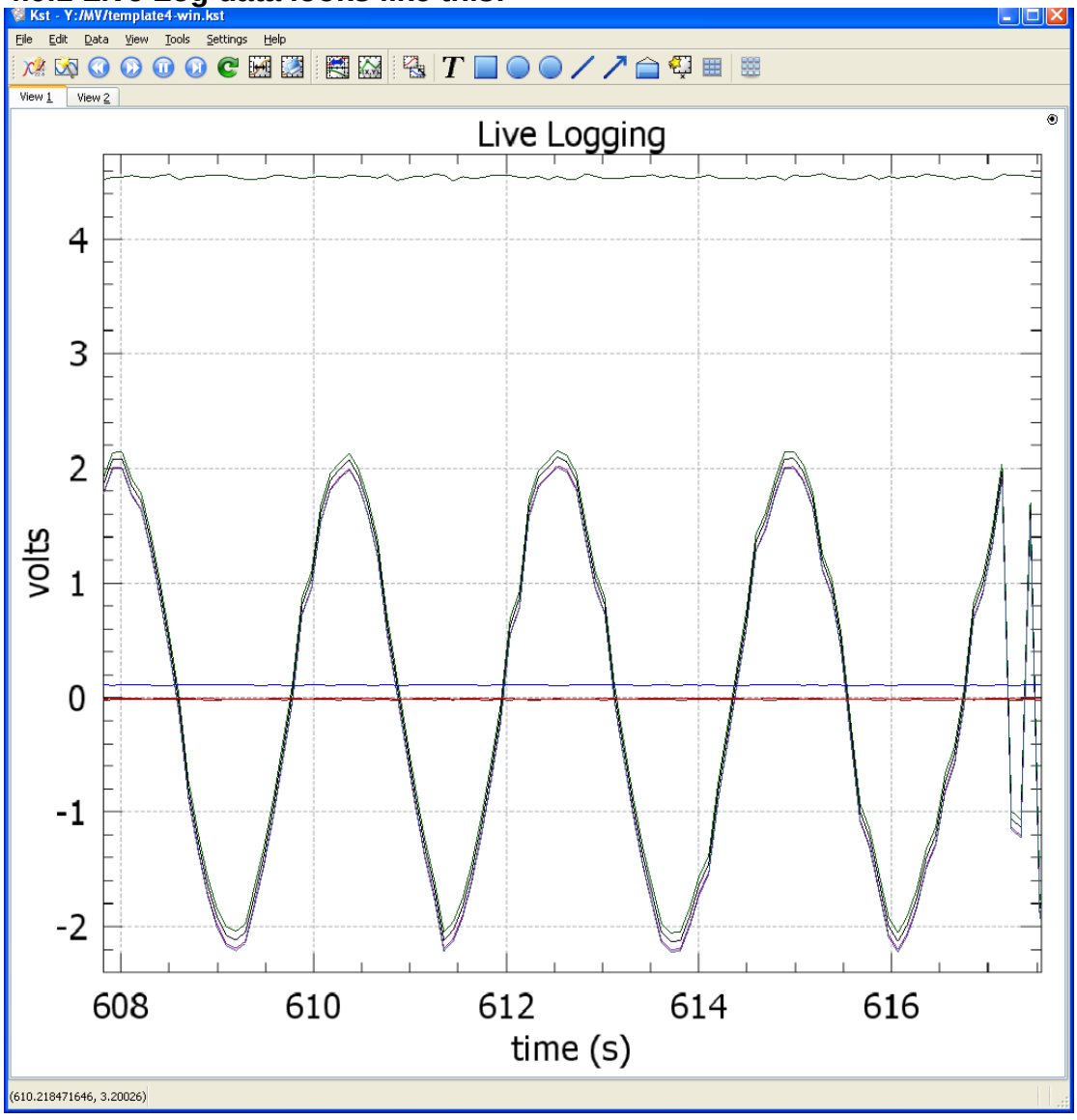
`multivent-cleanup`

4.9 Plotting the Data

Use **KST** to plot the data. You can configure the plot using the Data Wizard, and or you can take advantage of a setup we made. Two example `.kst` setups (for 4 events and 8 events respectively) are available on the shared directory, so you can load and plot directly (please make sure that 4 or 8 event files are available first).

It's specific for drive Z:\, first four events. To make it work with other drives, edit and substitute for Z:\. It's also possible to make a stored setup with relative paths. You can copy the .kst file with relative path to the network drive, or copy the contents of the network drive to local disk and save the relative path `.kst` file in the same directory.

4.9.1 Live Log data looks like this:



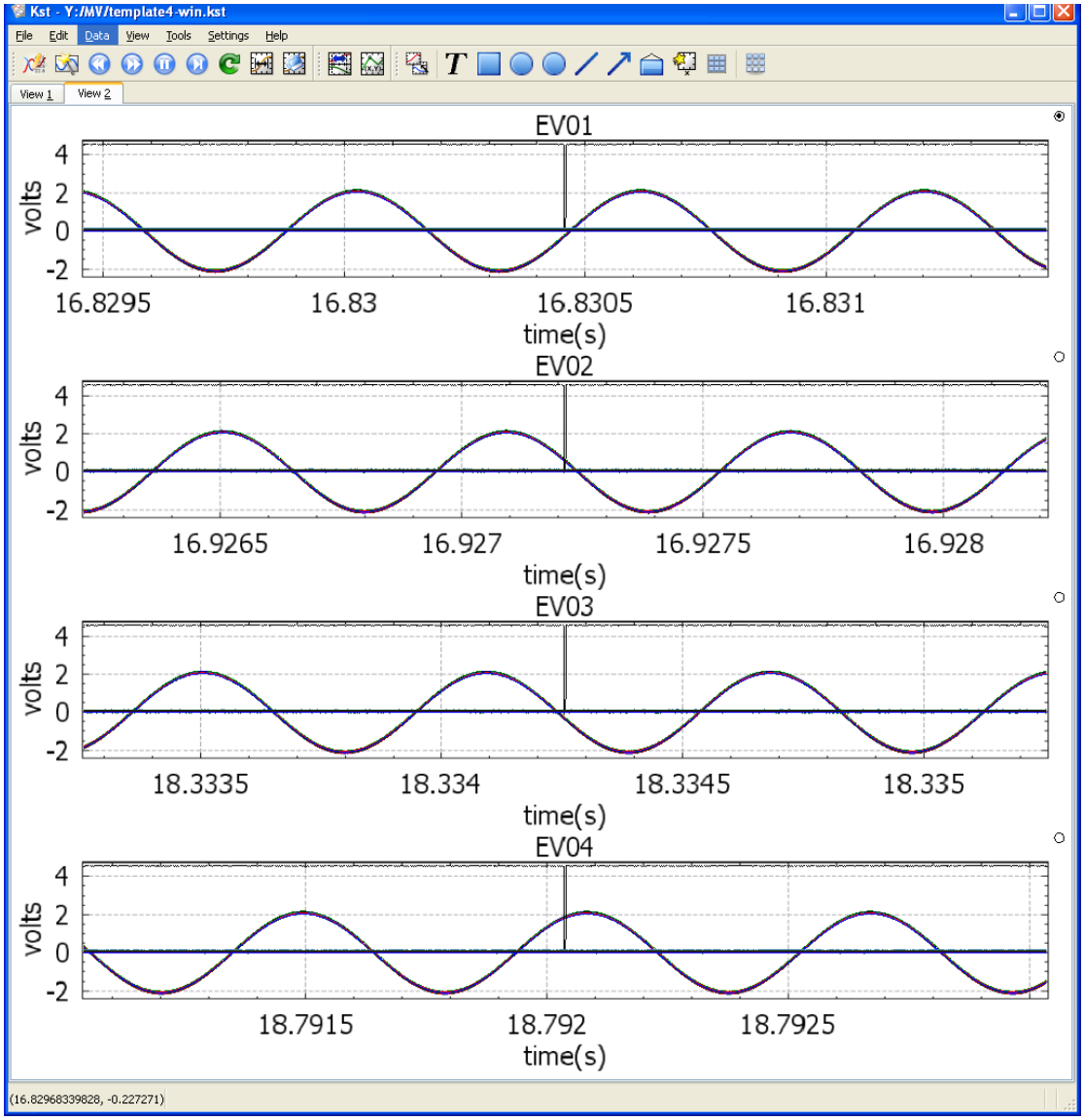
4.9.2 Event Data looks like this:

The Event is a falling edge on a digital line.

The digital line is sampled by the capture system.

The digital line is immediately pulses high, while the event capture is showing 10ms either side of the Event. The timebase shows seconds since the start of the shot.

Our test signal generated a pair of pulses every second, proving that the system can hande back-to-back pulses.



5 Appendix Kst Plot Tool Notes

We recommend Kst 2.0 (first version with port to MS-Windows).

Please be sure to use the latest version from sourceforge, it includes a number of fixes that make it more convenient to use. The file is a self-extracting zip archive, run it, then drill down to the “bin” directory and double-clock the Kst icon to run.

5.1 View Test data.

A test data set is provided here:

http://www.d-tacq.com/data/multivent-8x10K-example.zip

Download and extract the data, then use Kst to open the stored setup file

`template-8-2.kst`

The plot shows three tabs:

Live – snapshot of live data,

View 4, View 3: plot 8 events, 4 per page. Note that the digital event signal (captured on an analog input for reference is in the centre of each data set.

5.2 Current Limitations.

Generating an appropriate plot can be tedious. A .kst setup file with relative paths to be useful. It includes plots for 4 events. Of course, for more events, one would have to create the additional plot definitions, this is probably as simple as it can be, but is still slow (no replication macro). Do it once, get what you need and save the state.

NB: it's convenient to save a relative state file to the ACQ132 file share, however please note that behind this is a ramdisk, it is VOLATILE – so please back up to disk as well.

6 Appendix: Firmware Customisation

Cards are delivered fully customised. For the record, this was:

1. *Samba* Installed.
2. **`acq_demux`** installed
3. custom multivent scripts added.

```
ls -l /bigffs | cut -c 38-
84263 Aug 15 18:17 acq_demux-xscale-201008151413.tgz
   33 Aug 15 18:17 acq_demux-xscale.tgz -> acq_demux-xscale-
201008151413.tgz
   52 Aug 15 09:15 mps
  203 Aug 15 15:59 multivent-cleanup
  240 Aug 15 09:15 multivent-init
  543 Aug 15 13:09 multivent-run-demux
  375 Aug 15 11:17 rc.multivent
09520 Aug  3 12:03 samba-acqX00-0905191153.tgz
   27 Aug  3 12:03 samba-acqX00.tgz -> samba-acqX00-0905191153.tgz
  786 Aug 15 20:29 smb.conf
```

4. /ffs/user/rc.user
   calls /ffs/rc.multievent