ACQ2xx Labview User Guide

Prepared By: Peter Milne

Date: 11 June 2010

Rev	***Date***	***Description***
1	8 Feb 2010	Init
2	11 June 2010	Update screen shots for MPLOtChannel16

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:

Table of Contents

1 Overview

1.1 Example Code

Download example code : acq2lvrc.zip

1.2 Control and monitoring

ACQ2xx offers remote access to its command interpreter using the Web Service WS interface. Labview 2009 running on Windows is able to use .NET 2.0 to import the web-service and auto-generate an appropriate VI. The VI allows a client application to execute commands and queries direct on the ACQ2xx.

So, the Web Service offers a very easy way to make a remote procedure call on the ACQ. After that, the remote client application controls the ACQ by means of the highly functional ASCII-character command set supported by ACQ2xx.

An example remote control client is supplied.

This can connect to the client, run an initialization command on it, start, stop and monitor acquisition.

The full set of commands is defined in D-Tacq_2G_UserGuide

However, it's most likely that your application will only require a small subset of the available commands. D-TACQ is happy to provide a simple base application for any scenario.

1.3 Design time Procedure to create a Web Service remote VI:

NB: this procedure has been proven to work under Windows XP.

It didn't work under Windows 7..

Following the example in:

Using LabVIEW to Call SOAP Web Services

ACQ2xx Web Service Definition WSDL is stored on each card at
URL http:IP/acq2sh.wsdl

There's a public copy also available at: acq2xx.wsdl

1. Launch LabVIEW.
2. Select **Tools»Import»Web Service,** enter **URL**
3. Use the generated VI - see working example below.

Or download a pre-created library:
acq2xx_WebService_install_in_user.lib.zip

```
Download and unzip under:
C:\Program Files\National Instruments\LabVIEW 2009\user.lib

2.0 Access example project:
The example project acq2lvrc.lproj contains a number of examples.
```

2 Control

Control is vi SOAP webservice.

2.1 Init

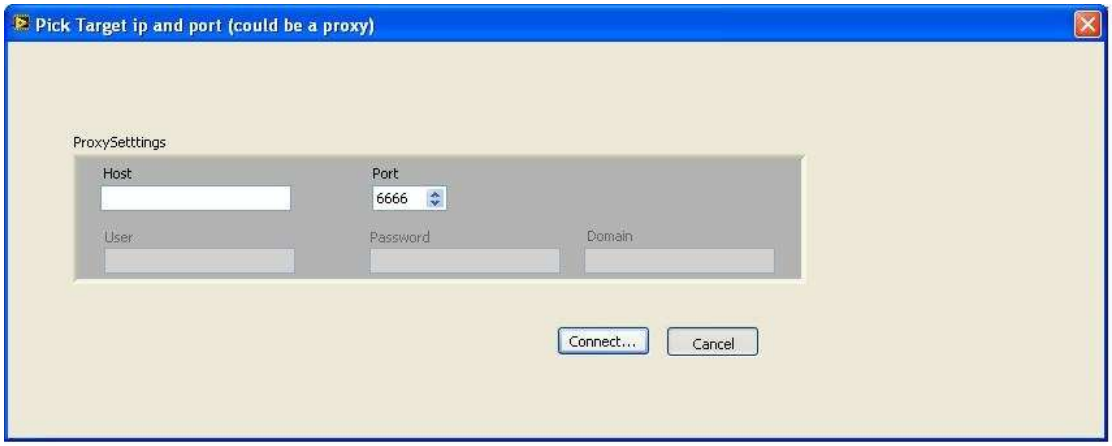
Uses a text init script to initialize the device.

`dt100_init.vi`

Controls shot setup using an command script file.

```
Two scripts are provided:
init.txt : set up a simple SOFT_TRANSIENT capture (for initial test)
init_trig.txt : set up a triggered captured (synchronized to outside
world)
```

2.1.1 Pick Target IP dialog



Enter target IP address.

Port 6666 is the default Web Service port - leave unchanged,

Press Connect...

The dialog disappears:

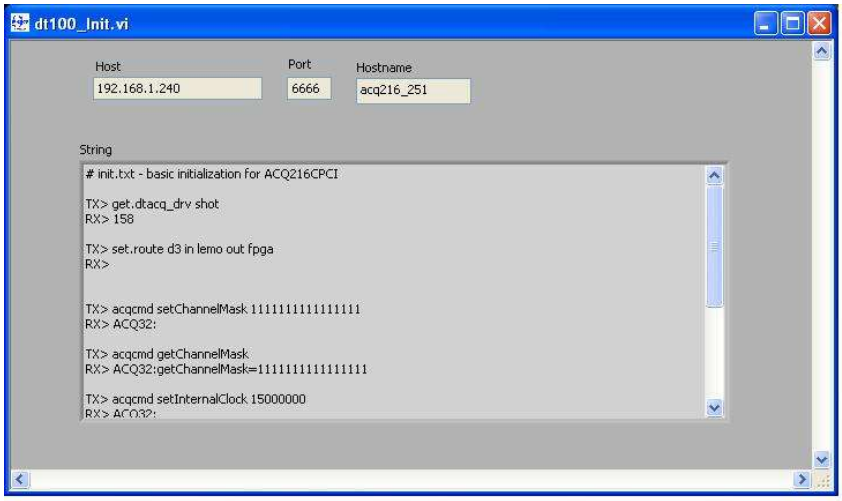
2.1.2 dt100_Init main screen:

A file dialog appears. Pick an appropriate init script

eg

```
init.txt : set up a simple SOFT_TRANSIENT capture (for initial test)
init_trig.txt : set up a triggered captured (synchronized to outside
world)
```

The script then runs, screen looks like this:

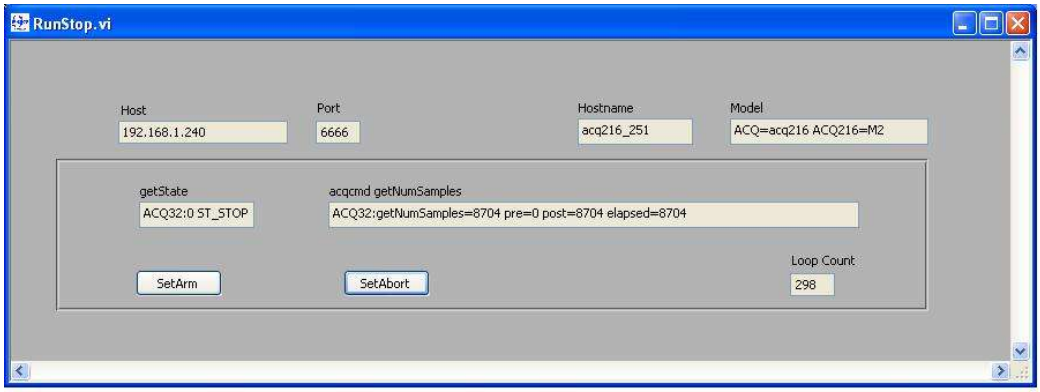


You may exit Init (once per series of identical shots)

2.2 RunStop

`RunStop.vi:`
Controls arm and abort of the shot.
Provides status
Same Pick target IP dialog as before.

Press SetArm. The card will run and status display will update.



Leave RunStop running to allow subsequent shots.

3 Post-Shot Transient Data

3.1 Full, High Level Example

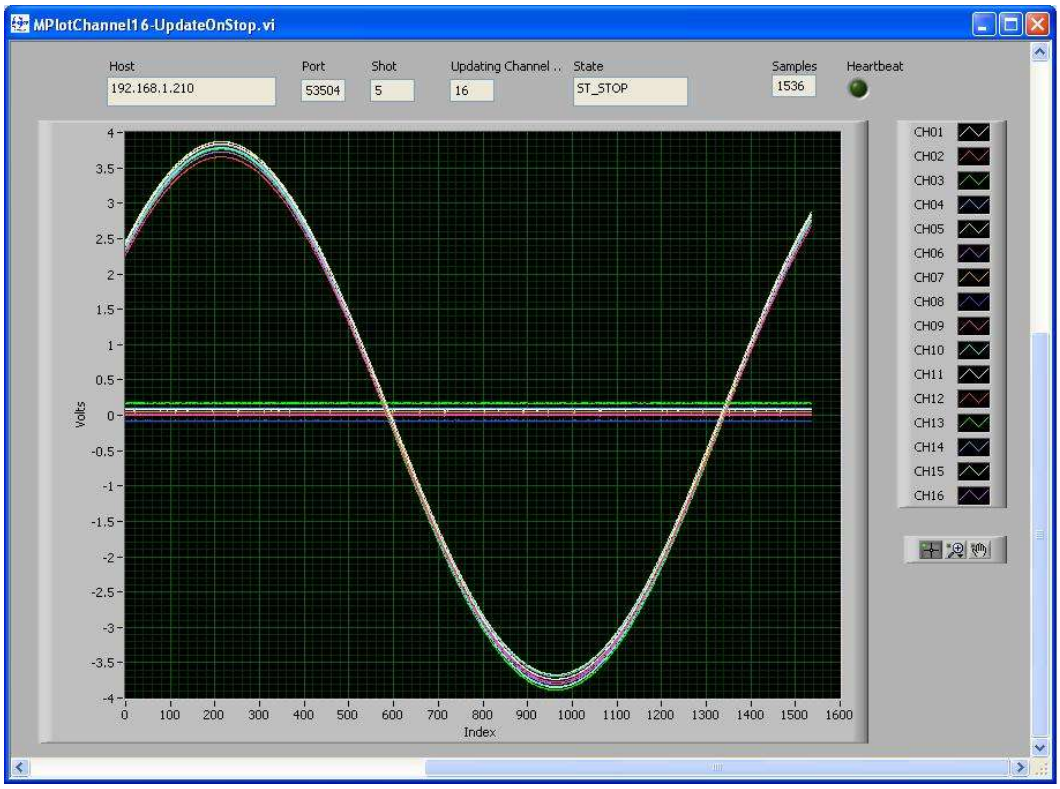
`MPlotChannel16-UpdateInStop.vi`
Monitors machine state, uploads and plots 16 channels at the end of a shot. Provides a heartbeat function.

3.1.1 MPlotChannel16 Pick Target IP and port

Enter target IP as before.

This VI uses 3 services on 3 ports. Accept the default and connect..

3.1.2 MPlotChannel16 Main Screen



- Shot: shows current shot number
- State: shows state of the card
- Samples: shows number of samples available
- Heartbeat: flashes to show link is alive
- Updating Channel...

On running the VI for the first time when the target is in ST_STOP state, or on transition to ST_STOP state, the VI will upload and plot data for each channel in sequence. Updating Channel.. gives a status indicating for this processing by showing the channel that is currently uploading.

3.2 HOST Pull on dt100d interface

Applies to post-shot transient data.

Connect to dt100d socket,

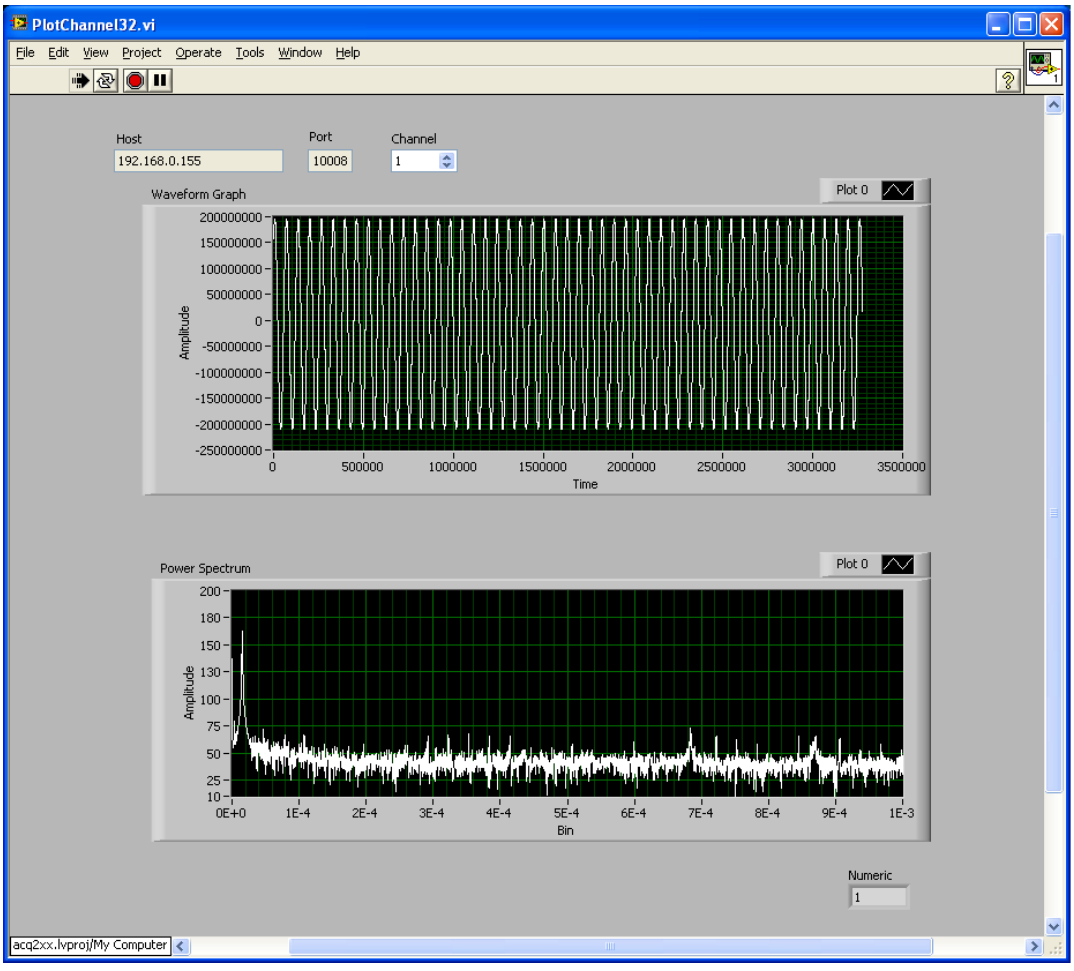
Command: dt100 open data [file]

Read byte count

Read binary data

See examples:

- ReadChannel16.vi (16 bit data)
- ReadChannel32.vi (32 bit data)



3.3 Windows Network share

ACQ2xx can export data on a Windows Network share.

Data from a transient capture if presented as a binary file per channel.

A client application simply has to open the file and read the data.

3.4 Target Push via ftp : Post Shot Transient

This is the highest performance upload method. It is assumed that the HOST provides an ftp service and disk space. The ACQ2xx sends data as files to the ftp server as soon as it is available:

Immediately after the shot, data is automatically uploaded to the ftp server.

The most efficient upload sends the data as a single raw data file "XX".

Alternatively, the transform process on ACQ2xx can be used to split the data into channels (de-multiplex), and data can be send one file per channel to the HOST computer. This method is the simplest for the HOST computer.

After the shot, the HOST computer target directory contains a series of binary files that are easy to process on the HOST:

01 .. NN (eg 64 on ACQ164CPCI, 96 on ACQ196CPCI),

Alternatively, the raw data file can be processed on the HOST eg by acq_demux, this will split the file into channels and save it in DirFile format.

API doc

4 Continuous Data

4.1 HOST Pull low rate streaming data on dt100d interface

Connect to dt100d socket,

Command: dt100 open stream

Read binary data in frames.

This technique may be applicable for low rate streaming (<10kHz).

4.2 Target Push via ftp : In-Shot Continuous

ACQ2xx sends data as a series of "TBLOCK" files, each TBLOCK is 6MB of raw data and numbered 000, 001, ... 130 (maximum number in 1GB local memory).

The data is transferred as a sustained rate of up to 40MB/s, and the target directory of the HOST computer becomes a mirror of the local buffer. At 40MB/s, the local buffer (860MB) is overwritten every 20s. This means the HOST-side software has 20s to deal with the data before it is overwritten.

Typically this raw data will be processed using acq_demux, to split into channels.

It's recommended that the ftp target directory is a RAMDISK, while the output of acq_demux can be stored to physical disk, and/or made available for downstream processing or plotting. D-TACQ recommends kst for plotting large multi-channel data sets.

4.3 Streaming High rate data on PCI backplane

Htstream program. Please contact D-TACQ for details.

4.4 Streaming High rate data on PCI-Express cable

Contact D-TACQ for details.