Table of Contents

Introduction

This page is supposed to be a set of instructions on how to get the automated HV control working with SCTDAQ.THe following is how I managed to get it to work at Glasgow

Hardware

In Glasgow I use a Keithley 2410 attached to the DAQ PC by GPIB (using a GPIB-usb cable)

RS232 "should" work as well, however i haven't tested this as yet

Other Voltage sources that can be controlled are

- Keithley 2410
- 487 picoammeter/voltage source
- ??

On my DAQ PC i have NI-VISA installed, which allows communication with the Keithley

SCTDAQ

The HV Control software is contained in the latest version of SCTDAQ that can be downloaded from the SVN

It is possible that newest version has the HV Control sub-project already contained in sctdaq_hsio project

If it does, then one of the sub projects will be called 'TkHVdll.'

If not, it can be added by doing the following:

- Open scdaq_hsio
- Go to File, then Add, then existing project
- Navigate to \sctdaq\khvnew and add 'TkHVdll.vcproj'

In both cases, compile the SCTDAQ files as per usual (see https://twiki.cern.ch/twiki/bin/viewauth/Atlas/StripsUpgradeDAQFreiburg for more details)

If successful, the HV control can be used via root

Using in ROOT

Run Stavelet.cpp as normal

in the ROOT window, setting up is initialised using the following commands

- gSystem->Load("khvdll.dll")
- keith= new TkHV("GPIB::24::INSTR")

24 is the address of my 2410 via the GPIB

At this point some error codes appear on the device bit it "should" still work fine

To ramp the voltage to a desired V:

keith->Ramp(V,I,b)

where

- V is the required voltage
- I is the current compliance limit (in uA)
- b is the ramping factor , which is either 1,2,3 or 4 *(50, 20 1 and 5V/s repectively)

To Perform an IV scan

keith->IVScan(Float_t vStart, Float_t vStop, Float_t vStep, Float_t vEnd, Float_t iLimit, Int_t msDelay)

where

- vStart is the starting Voltage
- vStop is the end voltage
- vStep is the voltage step
- vEnd is the voltage at which you want the source to return to
- iLimit is the current compliance (uA)
- msDelay is the delay between points (ms)

eg

keith->IVScan(1,10,1,0,10,5000);

(Information on how to log the data and write to file to follow)

LV Control

Again, my LV sourve is connected to the DAQ PC by GPIB

Open Measurement and automation to check the address the LV device is operating.

For me it was 6. (My device is a TTi TSX3510P)

Now open rootlogon.c file in the sctdaq directory

In my setup the number in bold changed to '6' as below

```
TTi* CSI = 0;
// GPIB address, for others use resource string
CSI = new TTi( 6);
if(CSI->isValid==0) CSI=0;
break;
```

Run Stavelet.cpp as normal.

Is all is working, the following code should appear in the ROOT window:

tokenTSX3510P
TTi::PrintStatus for TSX3510P at resource GPIB::6::INSTR
Max voltage 35V
Max current 10A
Set voltage 6.2V
Set current 5.2A
Output voltage 0.2V
Output current 0A

(the numbers will obvouisly chamge depending on the V and I that the device was set to)

The LV bias can now be controlled with the drop down menu in the GUI

Note: The LV supply must be switched on when stavelet.cpp is run, otherwise ROOT will crash

---

This topic: Sandbox > HVCONTROL
Topic revision: r2 - 2013-05-09 - AndrewBlue