

Table of Contents

How to setup a CCT VP 717 Single Board Computer.....	1
Installing vme drivers.....	1
Copying missing library shared objects.....	1
Setting up the system so that drivers are loaded at every boot.....	1
Setting up the A32 and A24 mapping in the VME memory.....	2
Running DAQ School Exercise 4.....	3
Setting up the VME SBC.....	3
Checking the database.....	3
Running monitoring.....	3
Running BL4S Daq.....	4
Setting up computers.....	4
Downloading source code.....	4
Setting up database.....	4
Setting up hosts.....	4
Monitoring.....	4
Labview Event Display.....	4
Controlling HV mainframe remotely.....	5
Connecting to Webcam PC's.....	6
Installing AE1200 Wireless adapter to RH6.....	7
Using DIP to publish data for a client in Technical Network(TN).....	8
Installing DIM.....	8
Installing DIP and test programs.....	8
Starting DIP Server that publish BL4S Variables.....	8

How to setup a CCT VP 717 Single Board Computer.

VP717 VME single board computers are items that can be rented from CERN Electronics Pool. One can plug in a monitor, keyboard and mouse and use it as a normal computer. After installing SLC6, one should log in as root, and do the following steps.

Installing vme drivers

```
cd root
mkdir ATLAS
cd ATLAS
mkdir bin lib driver
cd driver
cp ~joos/public/tdaq-05-03-00/drivers-05-03-00-x86_64/vme_rcc_tsi._KERNELNO__x86_64.ko vme_rcc_tsi.ko
cp ~joos/public/tdaq-05-03-00/drivers-05-03-00-x86_64/io_rcc._KERNELNO__x86_64.ko io_rcc.ko
cp ~joos/public/tdaq-05-03-00/drivers-05-03-00-x86_64/cmем_rcc._KERNELNO__x86_64.ko cmем_rcc.ko

cd ../bin/
cp ~joos/public/tdaq-05-03-00/installed/x86.../bin/tsiscope .
cp ~joos/public/tdaq-05-03-00/installed/x86.../bin/scanvme_tsci .
cp ~joos/public/tdaq-05-03-00/installed/x86.../bin/vme_rcc .
cp ~joos/public/tdaq-05-03-00/installed/x86.../bin/tsiconf .

cd ../lib/
```

Copying missing library shared objects

ldd command will show the missing libraries for each binary file, and they should be copied to /ATLAS/lib directory. One should repeat the following 3 lines until the command does not give more missing libraries such as "libSomething.so.3 => not found"

```
ldd ../bin/* |grep found
cp ~joos/public/tdaq-05-03-00/installed/x86.../library/MISSING_LIB_NAME
export LD_LIBRARY_PATH=.
```

During the installation for BL4S, we had to copy the following ones.

```
#Copy until nothing else appears. Example:
cp /afs/cern.ch/atlas/project/tdaq/inst/tdaq/tdaq-05-03-00/installed/x86_64-slc6-gcc47-opt/lib/lib...
cp /afs/cern.ch/atlas/project/tdaq/inst/tdaq/tdaq-05-03-00/installed/x86_64-slc6-gcc47-opt/lib/lib...
cp /afs/cern.ch/atlas/project/tdaq/inst/tdaq/tdaq-05-03-00/installed/x86_64-slc6-gcc47-opt/lib/lib...
cp /afs/cern.ch/atlas/project/tdaq/inst/tdaq/tdaq-05-03-00/installed/x86_64-slc6-gcc47-opt/lib/lib...
cp /afs/cern.ch/atlas/project/tdaq/inst/tdaq/tdaq-05-03-00/installed/x86_64-slc6-gcc47-opt/lib/lib...
cp /afs/cern.ch/atlas/project/tdaq/inst/tdaq/tdaq-05-03-00/installed/x86_64-slc6-gcc47-opt/lib/lib...
cp /afs/cern.ch/atlas/project/tdaq/inst/tdaq/tdaq-05-03-00/installed/x86_64-slc6-gcc47-opt/lib/lib...
```

Setting up the system so that drivers are loaded at every boot

```
cd /etc/init.d
cp ~joos/public/drivers_tdaq_vme .
cd ../rc5.d
ln -s ../init.d/drivers_tdaq_vme S98drivers_tdaq_vme
cd ../init.d
./drivers_tdaq_vme start
```

Setting up the A32 and A24 mapping in the VME memory.

#Setting the A32 and A24 mapping in the memory.

```
cd /ATLAS/lib/  
../bin/tsiconfig -a ../driver/vmetab
```

To see what to do inside the tsiconfig, here's a howTo from the master himself:
/afs/cern.ch/user/c/cyildiz/public/vme_mastermapping.mp4

To see the mapping, one can use tscope

```
../bin/tscope
```

```
=====
```

LSI	VME address range	PCI address range	EN	MRPFD
0	0x0000000000000000 - 0x0000000000ffffff	0x0000000040000000 - 0x0000000040ffffff	Yes	No
1	0x0000000000000000 - 0x0000000000ffffff	0x0000000041000000 - 0x0000000050ffffff	Yes	No
2	0x0000000000000000 - 0x000000000000ffff	0x0000000000000000 - 0x000000000000ffff	No	No
3	0x0000000000000000 - 0x000000000000ffff	0x0000000000000000 - 0x000000000000ffff	No	No
4	0x0000000000000000 - 0x000000000000ffff	0x0000000000000000 - 0x000000000000ffff	No	No
5	0x0000000000000000 - 0x000000000000ffff	0x0000000000000000 - 0x000000000000ffff	No	No
6	0x0000000000000000 - 0x000000000000ffff	0x0000000000000000 - 0x000000000000ffff	No	No
7	0x0000000000000000 - 0x000000000000ffff	0x0000000000000000 - 0x000000000000ffff	No	No

```
=====
```



Running DAQ School Exercise 4

Exercise 4 was originally run on CCT VP 110 VME SBC(32 bit), with operating system SLC4, and tdaq version 02-00-03. With help of experts the exercise is upgraded to run on CCT VP 717 VME SCB(64 bit), with operating system SLC6, and tdaq version 05-03-00.

Running exercise on this hardware with newer software requires the following steps.

Setting up the VME SBC.

- Install SLC 6
- Login as root. Create account daquser

```
addusercern daquser
```

- Install tdaq locally

Checking the database

Running monitoring

```
./emon_task -p part_Scintillator1 -t ReadoutApplication -e 1000 -v2
```

Running BL4S Daq

Setting up computers

The computers have to be set up to be able to handle the DAQ. By default SLC6 has ulimit set to restrict the maximum number of files and process open to 1024. It also rejects traffic to unknown ports in its iptable config.

On all the computers you'll use as DAQ components do the following as root:

Write the following in `/etc/security/limits.d/99-max-files.conf`

```
# set the max number of open files - issue
*          soft    nofile      65536
*          hard    nofile      65536
```

Write the following in `/etc/security/limits.d/99-nproc.conf`

```
*          soft    nproc       192751
```

You can list the iptables rules with `iptables -L INPUT --line-numbers`. This prints rule numbers. To delete rule number X, call `iptables -D INPUT X`.

Downloading source code

There are 2 repositories that have the BL4S daq related source code.

- DAQ Software and the database [↗](#): The source code and OKS database of the DAQ,
- DAQ Extras [↗](#): BL4S Software not related to DAQ (Analysis etc.)

Setting up database

Setting up hosts

The hosts that will be used in the DAQ system should be created. To do this a bash script is used, and this script takes a python program as input. Script can be found in `/bl4sextras/db_programs/` with instructions inside the README

Monitoring

There are several monitoring programs, most run with following command line arguments

```
emon_task2 -p part_BL4S -t ReadoutApplication -e 1000 -v 2 -c current_config.cfg
```

Labview Event Display

- Install labview by doing following steps. Note that we use 32 bit version, because VIPM does not work on 64bit (At least I(Cenk) could not get it to run). Answer all questions yes when you run the install script. You can just omit installing NI VISA Drivers, they are not needed.

```
cd /afs/cern.ch/sw/natinst/LabVIEW2013/32bit/labview_core
./INSTALL
```

- NOTE: Only in one of the PC's (bl4sdaq) I had to install libXinerama.i686 and mesa-libGL.i686 to get 32 bit labview to run.

```
yum install libXinerama.i686 mesa-libGL.i686
```

- Install "Asynchronous Message Communication" extension
 - ◆ Download VI Package Manager (VIPM) from <http://jki.net/vipm/download>, extract the zip file, start the executable "vipm"
 - ◆ When the VIPM window opens, search for "Asynchronous Message Communication" on right top. Install the package (If it has problem communicating with labview, in labview go to Tools->Option->VI Server, and click on TCP/IP to activate port 3363, then press OK)
- Get the event_display source code from Joseph Tag

Controlling HV mainframe remotely

The SY2527 H.V. Mainframe has ethernet connection, It can be controlled remotely following the steps:

- Register it to network going to CERN Network DataBase. Current one is registered as bl4shv on Cenk Yildiz.
- Start telnet in a windows or linux PC (Windows: go to Command Prompt, type "telnet" and press enter, Linux: Open up a terminal, type "telnet" and press enter)
- Connect to the HV mainframe with the following command:

```
o bl4shv 1527
```

- The rest is same as using the HV module from its own screen.

Connecting to Webcam PC's

```
rdesktop -d cern -u daquser -g 95% pcbl4swebcam
```

```
rdesktop -d cern -u daquser -g 95% pcbl4swebcam2
```

```
rdesktop -d cern -u daquser -g 95% pcbl4swebcam3
```

Installing AE1200 Wireless adapter to RH6

Warning: It probably won't work stably. In 2014 we used windows PC's with wireles adapters.

```
su
yum --enablerepo=elrepo install ndiswrapper-utils kmod-ndiswrapper
cd /afs/cern.ch/user/c/cyildiz/public/ae1200wireless
ndiswrapper -i bcmwlhigh5.inf
depmod -a
modprobe ndiswrapper
```


Using DIP to publish data for a client in Technical Network(TN)

Installing DIM

This step is only needed, if you want to start your own dns server. If you want to communicate with TN, this will not help, since you need to use one of public servers.

http://lhcb-daq.web.cern.ch/lhcb-daq/cgi-bin/rpmrepo_scripts/list_rpms.sh

For instance, SLC6 64 bits DIM :

```
cp /afs/cern.ch/lhcb/project/web/online/online-rpm-repo/slc6X/x86_64/RPMS/dim-v20r7-1.RHEL6.x86_64
yum localinstall dim-v20r7-1.RHEL6.x86_64.rpm dim-programs-v20r7-1.el6.x86_64.rpm
```

It should resolve all dependencies for you and install the DNS daemon in /etc/init.d/dnsd

To have a DIP Name server running, all you need is to run following commands to setup and restart dns daemon

```
echo "export DIM_DNS_PORT=2506" >> /etc/sysconfig/dnsd
service dnsd restart
```

You can then point your DIP API to "localhost".

Installing DIP and test programs

- Download DIP from <http://j2eeps.cern.ch/wikis/display/EN/DIP+API>
- unzip the file.
- In the folder run "source setenv64.sh"
- Go to examples directory, compile Server.cpp and Client.cpp by saying "make all"
- If there are errors, try to fix them by adding relevant C header files. (You may need to add "stdlib.h" and change a function Sleep to sleep (big to small caps))

Starting DIP Server that publish BL4S Variables

- It is in the daq_extras repo, there is already a working version of BL4S Dip Server. Do following to start the server after logging on to a PC with daquser account

```
cd /afs/cern.ch/user/d/daquser/public/daq_extras/dip_server
source setenv64.sh
./ServerBL4S
```

To check if the variables are published:

```
cd /afs/cern.ch/user/d/daquser/public/daq_extras/dip_server
source setenv64.sh
./browser64.sh
```

In the window, enter dipnsgpn1 as server, then you should see published variables under name bl4s in the tree.

-- CenkYildiz - 13 Mar 2014 - 27 Jun 2014 -- CenkYildiz - 07 Jul 2014 -- CenkYildiz - 29 Aug 2014

This topic: BL4S > HowTo

Topic revision: r22 - 2015-04-21 - TimBrooks



Copyright &© 2008-2019 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? Send feedback