**Introduction**

Control and Design System (CDS) is LIGO control system based on Linux Debian 10 (called Buster). This guide provides the required steps to install the system, and run a test model.

**A fresh copy of Linux Debian 10 (buster) needs to be installed, as follows:**

1. The version is AMD64 and the file is netinst CD image

https://www.debian.org/releases/buster/debian-installer/

Download as iso file

2. Make this iso file into a bootable usb. You need a tool for that called https://unetbootin.github.io/

Another tool is

https://rufus.ie/en/

The unetbootin worked for me. Use graphical install from the options when you boot from the usb.

3. Make the PC bootable from usb drive, from boot menu select usb drive or hit F11 at start and select usb drive.

4. BIOS settings:

Speed step/turbo mode Disabled

Quiet Boot Disabled

Hyper Threading Disabled

Advanced Power Management Configuration → Power Technology Disabled

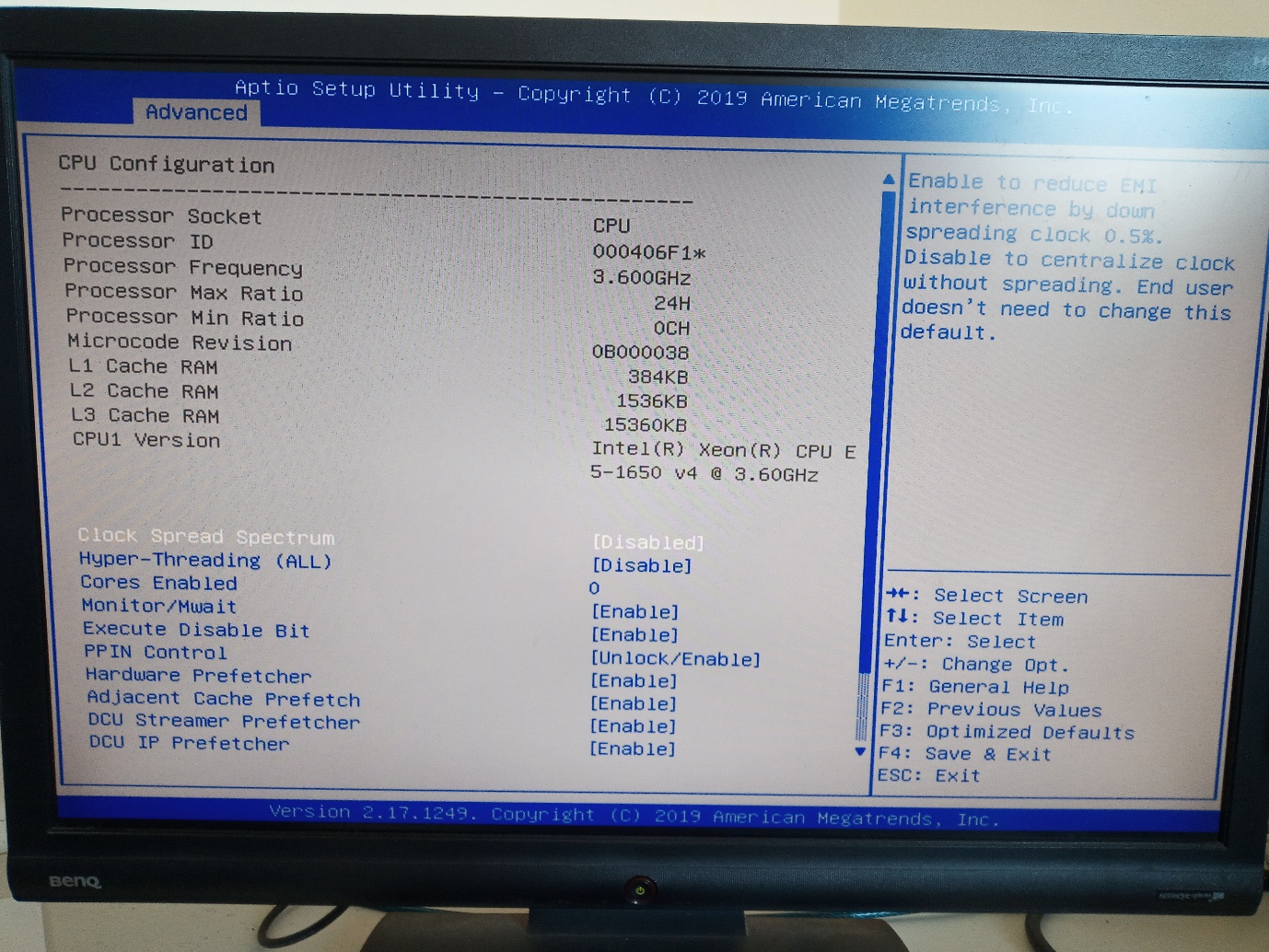
AES-NI Disabled

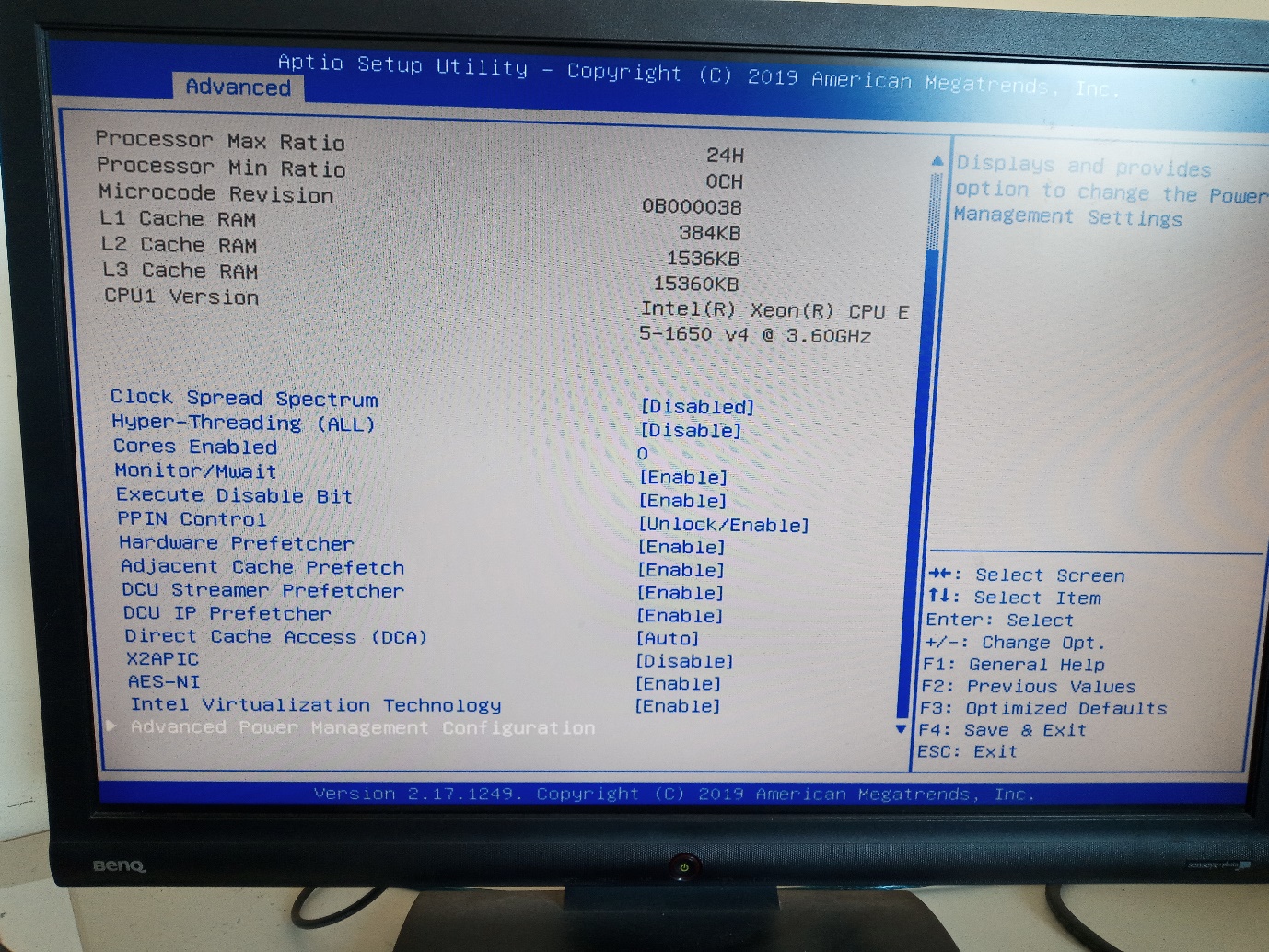
no UEFI

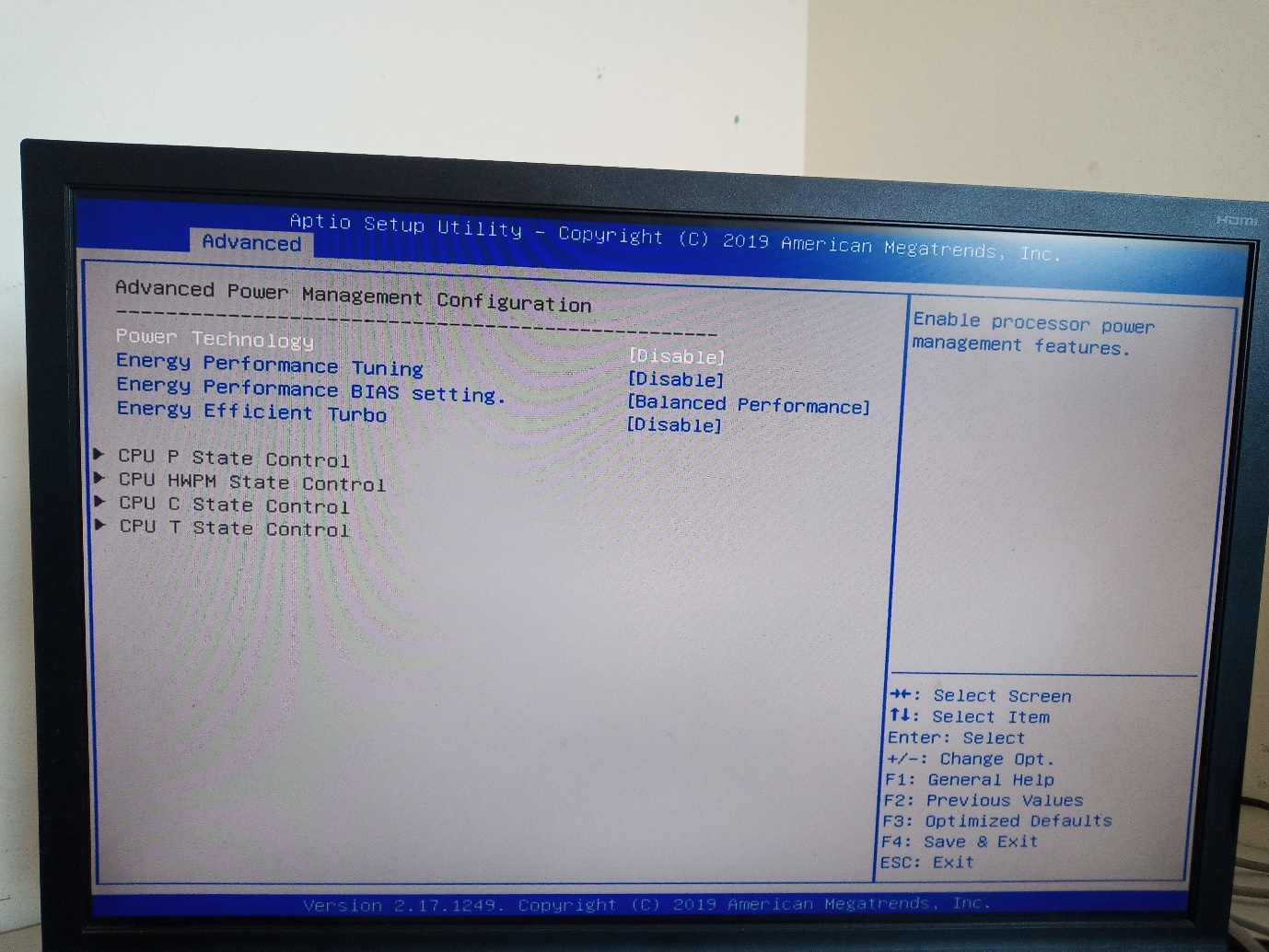
Also speed step needs to be turned off in the BIOS under advanced settings for CPU. Any setting that might change CPU speed must be disabled.

<https://dcc.ligo.org/DocDB/0103/T1300430/003/LIGO-T1300430-v3.pdf>

The following pictures show the actual values of the BIOS







5. Start installing Debian, see this link for a tutorial, same link attached as pdf, Use xfce as the desktop environment (Don’t use GNOME, see Issues below).

https://phoenixnap.com/kb/how-to-install-debian-10-buster#ftoc-heading-7

6. User name: ligocontrol, password: control, or you can use controls/controls

**Issues**

The boot screen stuck at

[OK] Started GNOME Display Manager

This error due to the old graphics card on the PC

Reinstall and don't select GNOME, use xfce

**Installing CDS Steps**

This information is from the LIGO git-hub page

https://git.ligo.org/cds/advligorts/-/wikis/home

The steps are as follows:

1. Download cdssoft-release-buster.deb

from:

http://apt.ligo-wa.caltech.edu/debian/pool/buster-unstable/cdssoft-release-buster/

Navigate to the download folder then

$ sudo dpkg -i cdssoft-release-buster.deb

$ sudo apt update

2. start installing packages:

$ sudo apt install cds-workstation

3. make this file in this location

/etc/apt/sources.list.d/cdssoft-restricted.list

navigate to the location

cd /etc/apt/sources.list.d

then

$ sudo cat "deb [signed-by=/usr/share/keyrings/cdssoft-unstable-archive-keyring.gpg] https://restricted:excludedimpairmentstrand@apt.ligo-wa.caltech.edu/debian buster-restricted main" > cdssoft-restricted.list

4. Install the AdvLigoRTS, CyMAC support package advligorts-cymac:

$ sudo apt install advligorts-cymac

The default SITE of TST, and default IFO of X1 can be used for a default CyMAC installation.

5. Make folder in opt as follows

/opt/usercode

When prompted for the user model files, type

/opt/usercode

this folder is where you will store the \*.mdl files

6. Installing a Real-Time Kernel

$ sudo apt install linux-image-4.19.0-6-rtcds-amd64-unsigned

After doing that run this

$ sudo update-grub

Restart the PC and boot into the new debian with rtcds as following:

When rebooting wait until the GRUB menu appears and press the down arrow key to select the "Advanced Options ..." selection. then choice

Debian GNU/Linux, with Linux 4.19.0-6rtcds-amd64

and hit enter.

Later you can make this option the default one to avoid having to select it every time you boot. Google how to change the default boot option in GRUB.

7. Verify you are running the correct kernel, switch to root

$ su root

then use the following:

$ root@x2boot0:/root$ uname -r

The results should be:

4.19.0-6-rtcds-amd64

8. Installing Control Room Tools.

$ sudo apt install cds-workstation

This was done in step 2, this command should tell you that it is already installed.

**Building codes**

**\*\* Files locations\*\***

Upon installation of RTS packages;

all RTS executables will be installed in /usr/bin.

Source code needed by the Real-time Code Generator (RCG)

will be installed in /usr/share/advligorts/src.

The RCG sources required to build the real-time code will be installed in the /usr/share/advligorts/src directory. All executable RTS code will be installed in the /usr/bin directory.

**\*\*Matlab version and required libraries\*\***

Clone githubadvligorts into your local PC. In

/opt/advligorts/

$ git clone <https://git.ligo.org/cds/advligorts.git> /opt/advligorts

Matlab 2019a should be used. The blocks library needs to be added to the Matlab path

addpath /opt/advligorts/src/epics/simLink/

addpath /opt/advligorts/src/epics/simLink/lib

To add those files permanently to Matlab path do the following

In Matlab:

1. Go to "File->Set Path" from within MATLAB or type "pathtool" at the MATLAB prompt.

2. Use the "Add" button to add your desired folder(s) to the MATLAB path.

3. Click "Save" so that this path is used in future MATLAB sessions.

That should save the "pathdef.m" for you.

**\*\*Control Model Build and Install Procedure\*\***

You need two files the iop model and the user model, both attached.

The naming is like below

x1iopxxx.mdl

x1: is the interferometer

iop is for the iop model

xxx any letters and numbers , this can be more than three characters.

Every mdl file has cdsParamters block, the cdsParamters block for the iop model should be set as the following

ifo=X1

rate=64K

dcuid=78

host=dhcp-154

iop\_model=1

no\_sync=1

Make sure to write the correct interferometer name in ifo and the host. host is the computer name you can find it by typing hostname in the terminal.

iop\_model=1 this should be =1 to indicate this is an iop model

no\_sync=1 this to enforce the synchronization to be done internally instead of the 1 pulse per second 1pps from the gps.

dcuid=78 this is unique identifier and should be different for each model 5-13 or 16-255

The user code is made in the different model as follows

x1tst.mdl

The cdsParamters has the following

ifo=X1

rate=2K

dcuid=11

host=dhcp-154

shmem\_daq=1

specific\_cpu=2

no\_sync=1

To buid the code

rtcds build <modelname>

rtcds install <modelname>

rtcds start <modelname>

you do that for the iop model first then you do that for the user code.

After building and installing the user app, you need to start the local\_dc using

local\_dc -s [yourmodel] -d /opt/rtcds/tst/x1/target/gds/param/

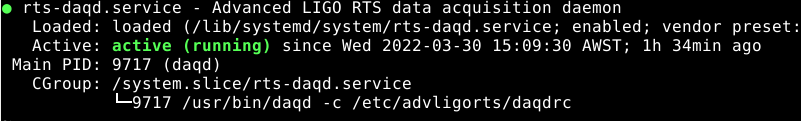
replace the [yourmodel] with the user model name (the control model and NOT the iop model) without the brackets. After running this command, keep the shell open as it will have a service running in the background.

The next thing to do is to check if the daqd is working. daqd is the frame builder , it is responsible for collecting all the channel signals and broadcast them.

$ systemctl status rts-daqd

if it is working it should show the following

Active (running) in green



Another way to check is to use,

sudo journalctl -f –unit rts-daqd.service

the above command will give you more information in case of any error.

To restart the rts-daq use the following

systemctl restart rts-daqd

If rts-daqd is not working then you will not be able to read any signal, and you will need to fix the issues to make the daqd working again.

If it does not work, you need to check that the master and .par files are existing in these locations and have the right information

/etc/advligorts/master

/etc/advligorts/testpoint.par

The master file contain the following information about all the model files (iop and user models), the first half contain the ini files which contain the channels information, the second half is the parameters files.

/opt/rtcds/tst/x1/chans/daq/X1IOPTLT2.ini

/opt/rtcds/tst/x1/chans/daq/X1TST.ini

/opt/rtcds/tst/x1/target/gds/param/tpchn\_x1ioptlt2.par

/opt/rtcds/tst/x1/target/gds/param/tpchn\_x1tst.par

The other file is the testpoint.par file and it should exist in the following location:

/etc/advligorts/testpoint.par

This file normally exists in

/opt/rtcds/tst/x1/target/gds/param/testpoint.par

And it has the following information:

[X-node11]

hostname=dhcp-154

system=x1tst

[X-node45]

hostname=scipe18

system=x1iopfe1

[X-node56]

hostname=dhcp-154

system=x1cdsbasictest1

[X-node78]

hostname=dhcp-154

system=x1ioptlt2

[X-node170]

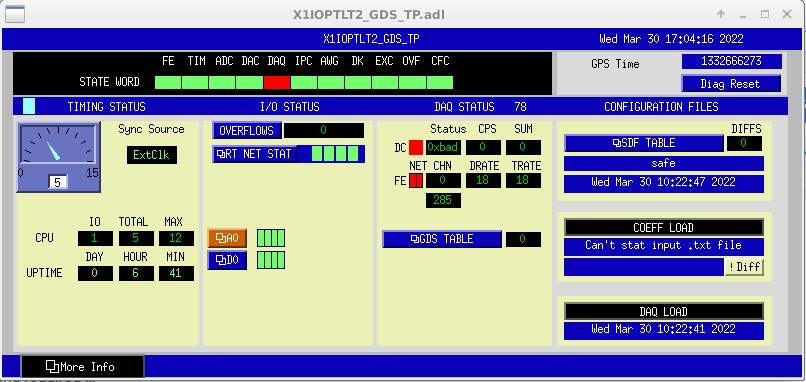
hostname=dhcp-154

system=x1ioptlt3

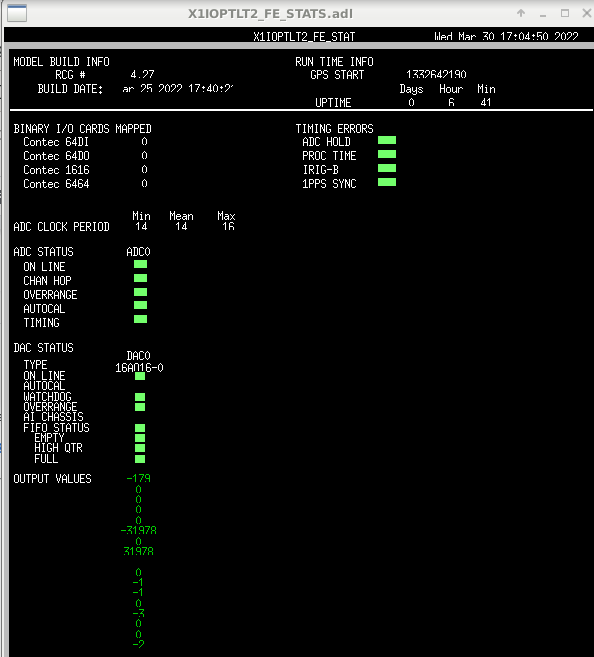
To check if the real time system works properly:

$ cd /opt/rtcds/tst/x1/medm/x1ioptlt2

$ medm X1IOPTLT2\_GDS\_TP.adl



When clicking on the More Info you get



Then you can start the diaggui, awggui to plot/inject signals. They should be able to see the available channels if you have the daqd up and running.

**Reading channels in realtime and offline**

Using ndscope:

NDSSERVER=dhcp-154:8088 ndscope

LIGONDSIP=dhcp-154 dataviewer

To store a channel you need to add a block of DAQ channels in the Simulink file, and write the channel name with a star \*. This means this channel will be recorded.

To read the channel use ndscope and add\_DQ after the channel name, the \_DQ is necessary to indicate that this is the recorded channel. Now when you scroll it will show the historical stored data of that channel.

**Setting Permissions:**

Users need to have read and write permissions to

/opt/rtcds/tst/x1

Otherwise, you will have error when updating the filter coefficients.

Add your user and advligorts to the group of each other and then give permissions to them.

Follow the guide here

https://git.ligo.org/cds/advligorts/-/wikis/sysadmin/ManagingTargetDirPermissions

copied below

Models are built and installed into the a target directory:

/opt/rtcds/<site>/<ifo>

This directory and the files in it should be owned by either controls or advligorts, and should be in either the controls or advligorts group.

Make sure that controls user is a member of the advligorts group and the advligorts user is a member of the controls group.

Any file that is not read-only should have group write permissions. Here is a useful command for adding group write permissions only to those files that already have owner write permissions.

find . ! -perm -g=w -perm -u=w -print0 | xargs -0 chmod g+w

**Disable the Nividia driver**

It was found that this graphics card driver cause ADC timeout error after running the real-time code for a few minutes. This driver needs to be disabled and the opensource nouveau driver should be used instead. The nouveau is already installed by default. To disable Nividia, check in this folder, which file is blacklisting the nouveau, once you find that, you need to open the file and comment that line, and black list Nividia instead.

$ sudo nano /etc/modprobe.d/nividia-blacklists-nouveau.conf

Comment nouveau, and blacklist nividia like below

#nouveau

blacklist nvidia

blacklist nvidia\_drm

blacklist nvidia\_modeset

Then run this command and restart PC afterwords

$ sudo update-initramfs -u

You can switch the comments in the file above if you want nivida back, you need to run the update command whenever you made such a change.

**Change the MAC address**

First run:

$ ip link show

This will list your network devices. Find the one you want to change. Next, run:

$ sudo ip link set dev <your device here> down

Then:

$ sudo ip link set dev <your device here> address <your new mac address>

Finally:

$ sudo ip link set dev <your device here> up

**Using diaggui**

Use the following command

NDSSERVER=dhcp-154:8088 LIGO\_RT\_BCAST=dhcp-154 diaggui -v

**Commands that you can use to start, stop models:**

If the RTS packages were installed, then the following basic start/stop commands are available.

• Start the model: rtcds start <modelname>

• Restart a running model: rtcds restart <modelname>

• Stop a running model: rtcds stop <modelnam>

Additional rtcds command line options can be found at rtcds command line interface.

Alternatively, the code can be started using the systemctl command

• Start the model: sudo systemctl start rts@modelname.target

• Restart a running model: sudo systemctl restart rts@modelname.target

• Stop a running model: sudo systemctl stop rts@modelname.target

When either rtcds or systemctl is used to start the control code, the following sequence of executables

will be loaded:

• EPICS interface software

-Setpoint monitoring software

-Sequencer to move data between the real-time kernel object and the CDS network

• Real-time control kernel module

• awgtpman, to support global diagnostic capabilities

**Information about the hardware:**

**\*\*ADC\*\***

The ADC modules presently being used for AdL FE systems are the General Standards model

PCI66-16AI64SSA-64-50M.

The datasheet for this module can be viewed at:

http://www.generalstandards.com/specs/pmc66\_16ai64ssa\_c\_spec\_092307.pdf

Note that the base module is actually a PMC bus module and can be used in PCIX and PCIe bus

systems by attaching it to the appropriate adapter.

**\*\*DAC\*\***

The DAC modules presently being used for AdL FE systems are the General Standards model

PCI66-16AO16-16-FO-DF.

The datasheet for this module can be viewed at:

http://www.generalstandards.com/specs/pmc66\_16ao16\_spec\_022508.pdf

Note that the base module is actually a PMC bus module and can be used in PCIX and PCIe bus

systems by attaching it to the appropriate adapter.

**\*\*Drivers link\*\***

<http://www.generalstandards.com/downloads/>

**SVN**

User apps developed in ligo can be copied to your local disk using

$ mkdir svn

$ cd svn

$ svn co https://redoubt.ligo-wa.caltech.edu/svn/cds\_user\_apps/ --username=albert.einstein

**Support:**

https://chat.ligo.org/ligo/channels/cds

Installation log, in case it could be of some help: http://www.sr.bham.ac.uk/dokuwiki/doku.php?id=cds:systeminstall2018:software\_installation\_log , login cdsguest, password readonly