FC7 Setup on CentOS 7

JAHID HOSSAIN
University of Nebraska-Lincoln

Important Link:
https://gitlab.cern.ch/cmsinnertracker/Ph2_ACF
https://indico.cern.ch/event/986962/sessions/388630/attachments/2202617/3726619/FC7_Setup.pdf

Golden Image Firmware (If the SD card is not setup yet)

1. Insert your SD card in a USB dongle or PC integrated SD slot

2. Download the golden firmware from the cms-tracker-daq webpage

3. Check the name of the SD card on your system

```
sudo fdisk -l
```

It will result a list of Disks on the pc, by knowing the size of the SD you can find the name of the SD card, for example here the size of the SD card is 32GB

```
$ sudo fdisk -l
Disk /dev/sdb: 31.9 GB, 31914983424 bytes, 62333952 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
```

```
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

So the SD card name here is /dev/sdb

4. Write the golden image on the SD card

```
sudo chmod 744 /dev/sd_card_name
```

For example, here the command will be $ sudo chmod 744 /dev/sdb

5. Go to the folder where you saved the sdgoldenimage.img file

6. To write the image to the SD card, run,

```
dd if=sdgoldenimage.img of=/dev/sd_card_name bs=512
```

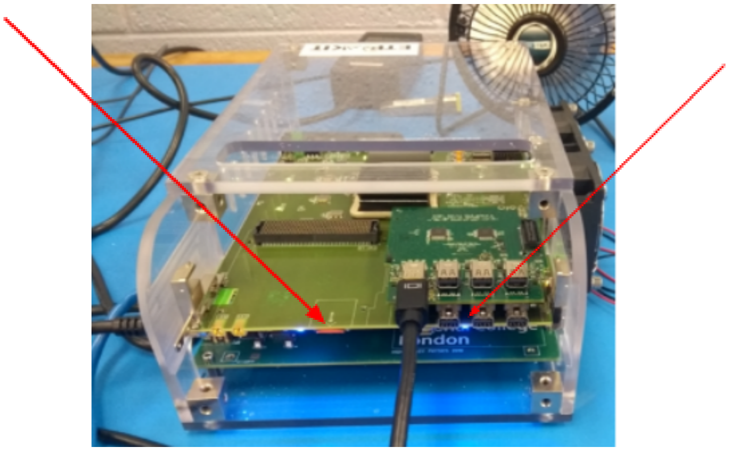
If the SD card is partitioned (formatted), pay attention to write on the block device (e.g. /dev/mmcblk0) and not inside the partition (e.g. /dev/mmcblk0p1)

7. Once the previous command is done, to check you can list the SD card:

```
./imgtool /dev/sd_card_name list
```

- there should be a GoldenImage.bin, with 20MB block size

8. Slide the FC7 in your nano-crate or uTCA crate and power it: when the SD card has the golden image on it, and it is not corrupted, you should see the blue LED of the FC7 blinking.



SD card

Blinking blue light

Middleware for the Inner-Tracker (IT) system

PC setup for communication with FC7

FC7 setup:

1. Check the name of your "second" ethernet card on your PC:

```
ifconfig -a
```

The output will be something like this,

```
$ ifconfig -a
enp3s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.64.156.212  netmask 255.255.0.0  broadcast 10.64.255.255
        inet6 fe80::e10c:38be:110c:7154  prefixlen 64  scopeid 0x20<link>
        ether 60:a4:4c:34:64:1a  txqueuelen 1000  (Ethernet)
        RX packets 3613844  bytes 585345078 (558.2 MiB)
        RX errors 0  dropped 2  overruns 0  frame 0
        TX packets 173705  bytes 43968309 (41.9 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp4s0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 00:0a:cd:3b:5f:49  txqueuelen 1000  (Ethernet)
        RX packets 216317  bytes 13199156 (12.5 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 217470  bytes 54564870 (52.0 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 435378  bytes 77137123 (73.5 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 435378  bytes 77137123 (73.5 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

```
…
```

In the example above it is enp4s0

2. Open as sudo the corresponding configuration file:

```
sudo nano  /etc/sysconfig/network-scripts/ifcfg-ethernet_card
```

where ethernet_card is the name of the ethernet card of your PC to which the FC7 is connected to (for example here it is enp4s0).

Configuration file should similar as below, where the ethernet must be a Static Network,

```
$ cat /etc/sysconfig/network-scripts/ifcfg-enp4s0
DEVICE=enp4s0
BOOTPROTO=none
ONBOOT=yes
USERCTL=no
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
NAME=enp4s0
UUID=60ba0ae5-ddb2-42bd-9f36-58bc81e72b83
HWADDR=00:0A:CD:3B:5F:49
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
IPV6_ADDR_GEN_MODE=stable-privacy
DNS1=192.168.1.1
IPADDR=192.168.1.1
GATEWAY=192.168.1.1
PREFIX=24
```

Add the following line in the file: **IPADDR=192.168.1.1, DNS1=192.168.1.1, GATEWAY=192.168.1.1**

3. Restart the network:

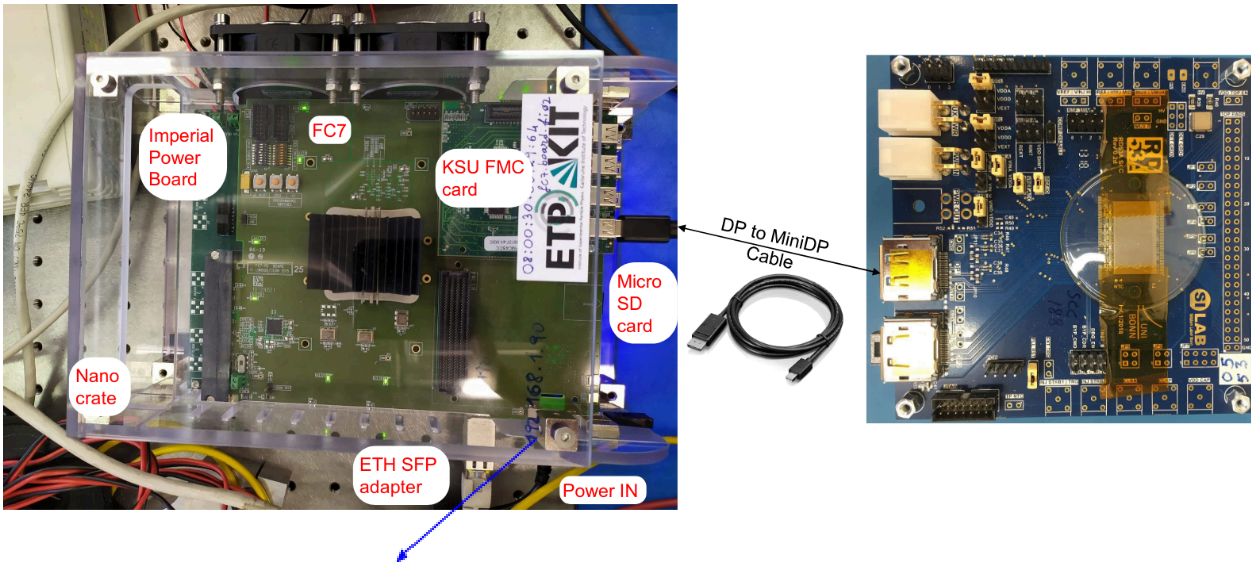
```
sudo /etc/init.d/network restart
sudo systemctl restart NetworkManager    #on CentOS9/ Almalinux9
```

4. Check again with ifconfig –a that the IP has been properly assigned

The output should be as follow,

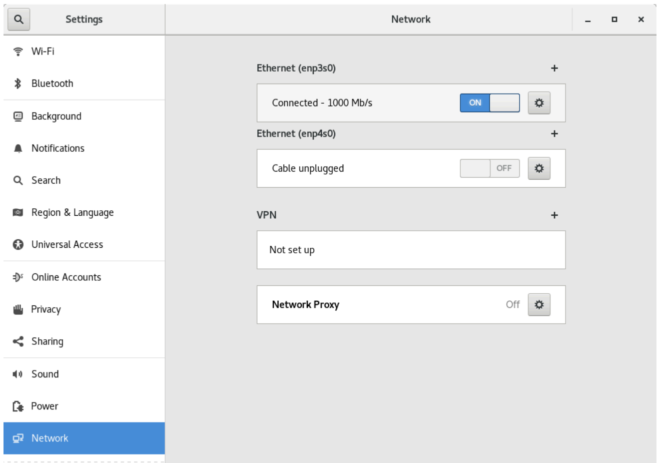
```
$ ifconfig -a
enp3s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.64.156.212  netmask 255.0.0.0  broadcast 10.255.255.255
        inet6 fe80::62a4:4cff:fe34:641a  prefixlen 64  scopeid 0x20<link>
        ether 60:a4:4c:34:64:1a  txqueuelen 1000  (Ethernet)
        RX packets 3869495  bytes 620832633 (592.0 MiB)
        RX errors 0  dropped 12  overruns 0  frame 0
        TX packets 202369  bytes 56945253 (54.3 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp4s0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 192.168.1.1  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::3326:110f:ad57:6a49  prefixlen 64  scopeid 0x20<link>
        ether 00:0a:cd:3b:5f:49  txqueuelen 1000  (Ethernet)
        RX packets 216317  bytes 13199156 (12.5 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 217470  bytes 54564870 (52.0 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
….
```

5. Physically connect the (nano)crate to this second ethernet card of your PC



6. Turn on the FC7 system and check the network settings on your PC, it should display connected.

Before



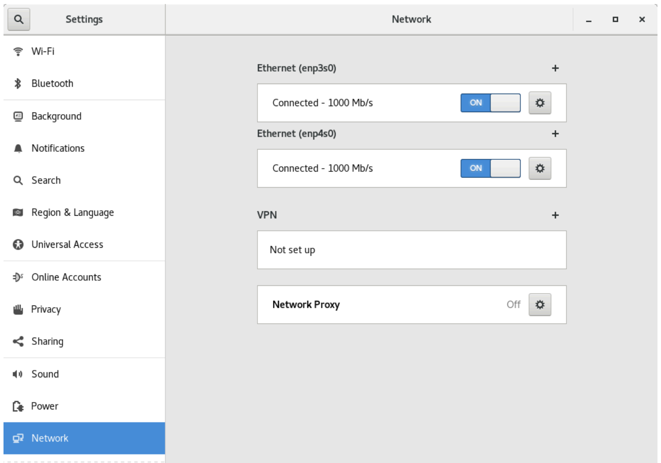
Ethernet (enp3s0)
Connected - 1000 Mb/s ON

Ethernet (enp4s0)
Cable unplugged OFF

VPN
Not set up

Network Proxy Off

After



Ethernet (enp3s0)
Connected - 1000 Mb/s ON

Ethernet (enp4s0)
Connected - 1000 Mb/s ON

VPN
Not set up

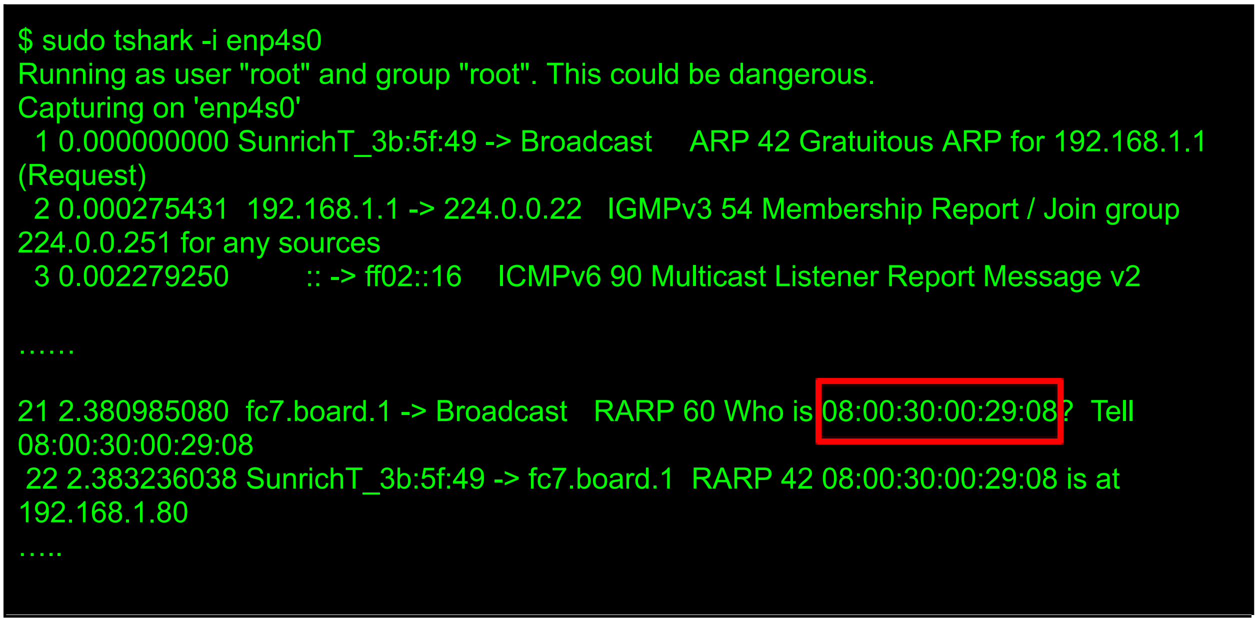
Network Proxy Off

7. Install wireshark in order to figure out which is the MAC address of your FC7 board and run tshark where ethernet_card is the name of the ethernet card of your PC to which the FC7 is connected to (for example here it is enp4s0).

```
sudo yum install wireshark
sudo tshark -i ethernet_card
```

To get the desired results turn off the FC7 if it is on then run the above command then turn on the FC7 again

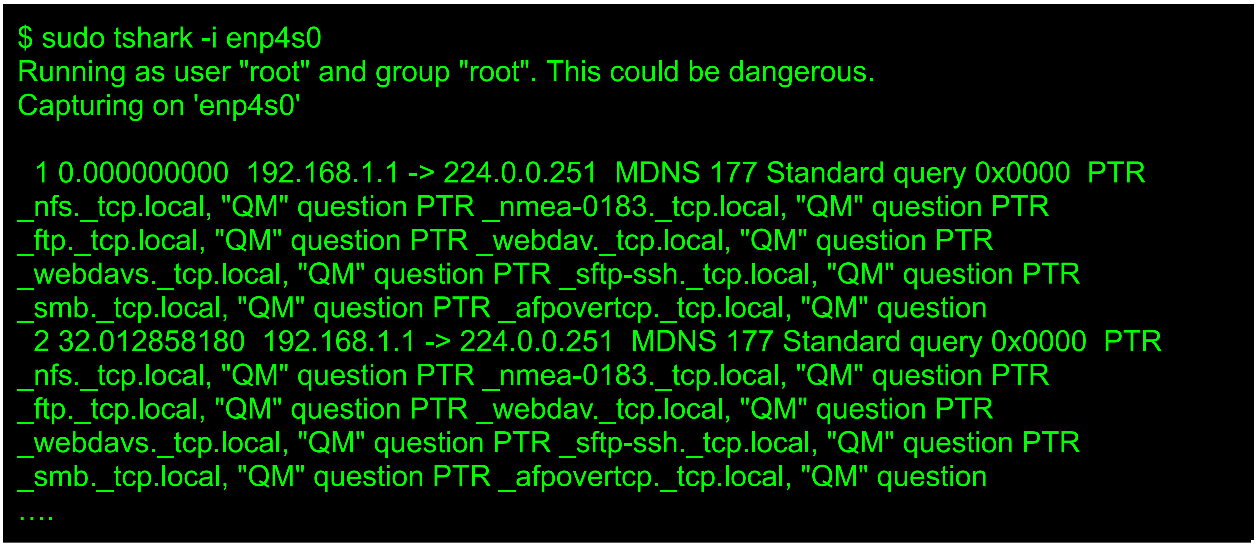
You should get the output similar like this,



```
$ sudo tshark -i enp4s0
Running as user "root" and group "root". This could be dangerous.
Capturing on 'enp4s0'
  1 0.000000000 SunrichT_3b:5f:49 -> Broadcast    ARP 42 Gratuitous ARP for 192.168.1.1
(Request)
  2 0.000275431  192.168.1.1 -> 224.0.0.22   IGMPv3 54 Membership Report / Join group
224.0.0.251 for any sources
  3 0.002279250         :: -> ff02::16    ICMPv6 90 Multicast Listener Report Message v2

……

21 2.380985080  fc7.board.1 -> Broadcast   RARP 60 Who is 08:00:30:00:29:08?  Tell
08:00:30:00:29:08
 22 2.383236038 SunrichT_3b:5f:49 -> fc7.board.1  RARP 42 08:00:30:00:29:08 is at
192.168.1.80
…..
```

For example here the MAC address of the FC7 is **08:00:30:00:29:08**

If you don't turn off the FC7 before running the command you would get something like this,



```
$ sudo tshark -i enp4s0
Running as user "root" and group "root". This could be dangerous.
Capturing on 'enp4s0'

  1 0.000000000  192.168.1.1 -> 224.0.0.251  MDNS 177 Standard query 0x0000  PTR
_nfs._tcp.local, "QM" question PTR _nmea-0183._tcp.local, "QM" question PTR
_ftp._tcp.local, "QM" question PTR _webdav._tcp.local, "QM" question PTR
_webdavs._tcp.local, "QM" question PTR _sftp-ssh._tcp.local, "QM" question PTR
_smb._tcp.local, "QM" question PTR _afpovertcp._tcp.local, "QM" question
  2 32.012858180  192.168.1.1 -> 224.0.0.251  MDNS 177 Standard query 0x0000  PTR
_nfs._tcp.local, "QM" question PTR _nmea-0183._tcp.local, "QM" question PTR
_ftp._tcp.local, "QM" question PTR _webdav._tcp.local, "QM" question PTR
_webdavs._tcp.local, "QM" question PTR _sftp-ssh._tcp.local, "QM" question PTR
_smb._tcp.local, "QM" question PTR _afpovertcp._tcp.local, "QM" question
….
```

You could also be able to identify the MAC address of the FC7 by reading its board id as described here and then find the corresponding MAC address using this table

Alternatively you can connect the MCH to the LAN and identify its MAC address as described here.

8. Add this address and the name that you want to give your FC7 in the /etc/ethers file using a text editor,

```
sudo nano /etc/ethers
```

Add the following line in the file,
08:00:30:00:29:08   unl_fc7

For example,

```
$ cat /etc/ethers
08:00:30:00:29:08 unl_fc7
```

08:00:30:00:29:08 (MAC address of the FC7) unl_fc7 (You can put any name here)

9. And finally give the card an IP address that belongs to your subnet, adding it to /etc/hosts:

```
sudo nano  /etc/hosts
```

Add the following line in the file,

192.168.1.80 unl_fc7

Here **192.168.1.80** is the IP address that should be the same as the CMSIT.xml file that we will use later to run the DAQ. And **unl_fc7** is the name that is given previously in the /etc/ethers.

```
$ cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6

192.168.1.80 unl_fc7
```

You can also comment the first two lines with a harsh (#) mark.

10. Restart the network:

```
sudo /etc/init.d/network restart
sudo systemctl restart NetworkManager     #on CentOS9 / Almalinux9
```

11. Install the rarpd daemon

```
sudo yum install ~/path/rarp_file_name.rpm
```

Taking rarp_file_name.rpm file from here.

12. Start rarpd daemon:

```
sudo systemctl start rarpd
```

Or,

```
sudo rarpd -e -A   # on Almalinux9
```

To start Docker
sudo systemctl restart docker

To start rarpd automatically after bootstrap:

```
sudo systemctl enable rarpd
```

Output:

```
rarpd.service is not a native service, redirecting to systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable rarpd
```

13. Try to ping FC7 on the IP address or the name that you gave it in the hosts file, for example,

```
ping 192.168.1.80
```

The output should be like,

```
$ ping 192.168.1.80
PING 192.168.1.80 (192.168.1.80) 56(84) bytes of data.
64 bytes from 192.168.1.80: icmp_seq=1 ttl=64 time=0.058 ms
64 bytes from 192.168.1.80: icmp_seq=2 ttl=64 time=0.054 ms
64 bytes from 192.168.1.80: icmp_seq=3 ttl=64 time=0.050 ms
….
```

Then you should be able to upload firmware on the FC7 with the fpgaconfig tool in the Ph2_ACF middleware.

IT-DAQ (Ph2_ACF middleware) setup and run:

1. Install prerequisite softwares on CentOS7, If you would like to setup on CentOS8/RHEL 9.1 or AlmaLinux 9.1 check instructions here

Create a python environment before

```
sudo yum install python3-virtualenv
python3 -m venv myenv
source myenv/bin/activate
```

Don't forget to activate your env every time you restart terminal or pc

```
sudo yum install pugixml-devel
```

If necessary run this following line before the previous command,

```
sudo yum install epel-release
```

2. Install boost by running,

```
sudo yum install boost-devel
```

3. Install CERN ROOT

```
sudo yum install root
sudo yum install root-net-http root-net-httpsniff root-graf3d-gl root-physics root-montecarlo-eg
root-graf3d-eve root-geom libusb-devel xorg-x11-xauth.x86_64
```

4. Install IPbus (uhal & controlhub together) from here.

```
sudo yum groupremove uhal
sudo curl https://ipbus.web.cern.ch/doc/user/html/_downloads/ipbus-sw.centos7.x86_64.repo
-o /etc/yum.repos.d/ipbus-sw.repo
sudo yum clean all
sudo yum groupinstall uhal controlhub
sudo yum groupinstall uhal-debug
export LD_LIBRARY_PATH=/opt/cactus/lib:$LD_LIBRARY_PATH
```

5. Checkout the DAQ code from git:

```
sudo yum install git
git clone --recurse-submodules https://gitlab.cern.ch/cms_tk_ph2/Ph2_ACF.git
cd Ph2_ACF
git checkout tags/v4-06 -b Jahid
```

6. Then

```
sudo yum intstall cmake
cd Ph2_ACF; source setup.sh; mkdir myBuild; cd myBuild; cmake ..; make -j4; cd ..
```

7. Next run the DAQ

```
mkdir choose_a_name
cp settings/RD53Files/CMSIT_RD53.txt choose_a_name
cp settings/CMSIT.xml choose_a_name
cd choose_a_name
```

8. Edit the file CMSIT.xml in case you want to change some parameters needed for the calibrations or for configuring the chip

9. Run the command to reset the FC7 (just once):

```
CMSITminiDAQ -f CMSIT.xml -r
```

Note that, If you get output like the following, that means the IP that is connected to FC7 is not properly static (you can check it by "ip r" command. You also might get this output when FC7 is not turned on or not connected to the PC.

```
$CMSITminiDAQ -f CMSIT.xml -r
|03:00:46|I|Trying to connect to the Power Supply Server...
bool TCPClientBase::connect(int, unsigned int)Connecting Client socket to server
name-127.0.0.1-serverPort: 7000
bool TCPClientBase::connect(int, unsigned int)Connecting Client socket to server ip
-127.0.0.1-serverPort: 7000
bool TCPClientBase::connect(int, unsigned int)Can't connect to 127.0.0.1
|03:00:46|I|Cannot connect to the Power Supply Server, power supplies will need to be
controlled manually
virtual TCPClientBase::~TCPClientBase()Closing TCPSocket #8
virtual TCPClientBase::~TCPClientBase()TCPSocket #8 closed.
virtual TCPSocket::~TCPSocket()Clean socket close!
|03:00:46|I|Resetting the backend board... it may take a while
Error while writing the following parameters: system.ctrl_2.fmc_pg_c2m,
system.ctrl_2.fmc_l8_pwr_en, system.ctrl_2.fmc_l12_pwr_en,
terminate called after throwing an instance of 'uhal::exception::ControlHubTargetTimeout'
  what():  The ControlHub did not receive any response from the target with URI
"chtcp-2.0://localhost:10203?target=192.168.1.80:50001"
ControlHub returned error code 3 = 'no reply to status packet'

Aborted (core dumped)
```

If we get output like the following that means we need to upload the firmware in the SD Card,

```
$ CMSITminiDAQ -f CMSIT.xml -r
|03:25:26|I|Trying to connect to the Power Supply Server...
bool TCPClientBase::connect(int, unsigned int)Connecting Client socket to server
name-127.0.0.1-serverPort: 7000
bool TCPClientBase::connect(int, unsigned int)Connecting Client socket to server ip
-127.0.0.1-serverPort: 7000
bool TCPClientBase::connect(int, unsigned int)Can't connect to 127.0.0.1
```

```
|03:25:26|I|Cannot connect to the Power Supply Server, power supplies will need to be
controlled manually
virtual TCPClientBase::~TCPClientBase()Closing TCPSocket #8
virtual TCPClientBase::~TCPClientBase()TCPSocket #8 closed.
virtual TCPSocket::~TCPSocket()Clean socket close!
|03:25:26|I|Resetting the backend board... it may take a while
terminate called after throwing an instance of 'uhal::exception::TransactionLevelError'
  what():  Bad response code (0x6 = 'bus timeout on read') received for Read-Modify-Write
bits at base address 0x40000040 (12 nodes under "RD53.user.ctrl_regs" match).  URI:
"chtcp-2.0://localhost:10203?target=192.168.1.80:50001". Sent/received headers: 0x2006014f
/ 0x20060046 (transaction 1/1 bytes into IPbus payload)
Aborted (core dumped)
```

So the next step is to upload the proper firmware in the SD card,

10. Run the following command to check which firmware is on the microSD card

```
fpgaconfig -c CMSIT.xml -l
```

The output should be as below,

```
$ fpgaconfig -c CMSIT.xml -l
15.02.2022 15:29:10: |140301307129216|I| Trying to connect to the Power Supply Server...
bool TCPClientBase::connect(int, unsigned int)Connecting Client socket to server
name-127.0.0.1-serverPort: 7000
bool TCPClientBase::connect(int, unsigned int)Connecting Client socket to server ip
-127.0.0.1-serverPort: 7000
bool TCPClientBase::connect(int, unsigned int)Can't connect to 127.0.0.1
15.02.2022 15:29:10: |140301307129216|I| Cannot connect to the Power Supply Server,
power supplies will need to be controlled manually
virtual TCPClientBase::~TCPClientBase()Closing TCPSocket #12
virtual TCPClientBase::~TCPClientBase()TCPSocket #12 closed.
virtual TCPSocket::~TCPSocket()Clean socket close!
15.02.2022 15:29:11: |140301307129216|I| 3 firmware images on SD card:
15.02.2022 15:29:11: |140301307129216|I|  - GoldenImage.bin
15.02.2022 15:29:11: |140301307129216|I|  - IT-KSU-L8-KSU-1G28
```

To get the last line,

Download the proper IT firmware version from here

For the case of SCC card, the proper firmware version is, v4.2-IT_CMS_FW
And the tar file is KSU-FMC(L12) & KSU-FMC(L8)
IT-uDTC_L12-KSU-4xSCC_L8-KSU4xSCC_ELECTRICAL_RD53A_x1G28.tar.gz

The command line for the uploading proper firmware is,

```
fpgaconfig -c CMSIT.xml -f ~/path/firmware_file_name_on_the_PC -i
firmware_file_name_on_the_microSD
```

Note that you need to untar the file before this command. You can also save the file in current working directory. "fpgaconfig" command has to run from the "choose_a_name" directory. Here, firmware_file_name_on_the_microSD is the given name of the firmware by the users.

Then run this command again, "fpgaconfig -c CMSIT.xml -l" you should see the name of the firmware in the list.

After this, run this following command to load a new firmware from the microSD card to the FPGA,

```
fpgaconfig -c CMSIT.xml -i firmware_file_name_on_the_microSD
```

Output:

```
$ fpgaconfig -c CMSIT.xml -i IT-KSU-L8-KSU-1G28
15.02.2022 15:53:03: |139991689293184|I| Trying to connect to the Power Supply Server...
bool TCPClientBase::connect(int, unsigned int)Connecting Client socket to server
name-127.0.0.1-serverPort: 7000
bool TCPClientBase::connect(int, unsigned int)Connecting Client socket to server ip
-127.0.0.1-serverPort: 7000
bool TCPClientBase::connect(int, unsigned int)Can't connect to 127.0.0.1
15.02.2022 15:53:03: |139991689293184|I| Cannot connect to the Power Supply Server,
power supplies will need to be controlled manually
virtual TCPClientBase::~TCPClientBase()Closing TCPSocket #12
virtual TCPClientBase::~TCPClientBase()TCPSocket #12 closed.
virtual TCPSocket::~TCPSocket()Clean socket close!
15.02.2022 15:53:03: |139991689293184|I| >>> Done <<<
```

Finally run this command again "CMSITminiDAQ -f CMSIT.xml -r" and should get this following results,

```
$ CMSITminiDAQ -f CMSIT.xml -r
|03:55:01|I|Trying to connect to the Power Supply Server...
bool TCPClientBase::connect(int, unsigned int)Connecting Client socket to server
name-127.0.0.1-serverPort: 7000
bool TCPClientBase::connect(int, unsigned int)Connecting Client socket to server ip
-127.0.0.1-serverPort: 7000
bool TCPClientBase::connect(int, unsigned int)Can't connect to 127.0.0.1
|03:55:01|I|Cannot connect to the Power Supply Server, power supplies will need to be
controlled manually
virtual TCPClientBase::~TCPClientBase()Closing TCPSocket #8
virtual TCPClientBase::~TCPClientBase()TCPSocket #8 closed.
virtual TCPSocket::~TCPSocket()Clean socket close!
|03:55:01|I|Resetting the backend board... it may take a while
|03:55:01|I|Waiting for DDR3 calibration...
|03:55:02|I|    --> DDR3 calibration done
|03:55:02|I|Now you can start using the DAQ ... enjoy!
```

11. Run the command:

```
CMSITminiDAQ -f CMSIT.xml -c name_of_the_calibration
```

For example, Latency scan, PixelAlive, Noise scan, SCurve scan etc.

One example is,

$ CMSITminiDAQ -f CMSIT.xml -c latency

For more information, check out these links,

Note: One needs to upload the firmware every time restarting FC7 and Ph2_ACF

https://gitlab.cern.ch/cmsinnertracker/Ph2_ACF
https://indico.cern.ch/event/842824/attachments/1920624/3179171/IT_DAQ_Calibrations.pdf

Thanks!!