

# Tutorial for the installation of Outer tracker hybrid test system

P. Chatagnon, F. Ferro

November 16, 2023

All the instructions below can be found at the following links:

- <https://cms-tracker-daq.web.cern.ch/cms-tracker-daq/> : All information on how to connect the FC7, upload firmwares, etc
- [https://gitlab.cern.ch/cms\\_tk\\_ph2/Ph2\\_ACF](https://gitlab.cern.ch/cms_tk_ph2/Ph2_ACF) : Repository for the acquisition software
- [https://gitlab.cern.ch/cms\\_tk\\_ph2/cmsph2\\_tusb](https://gitlab.cern.ch/cms_tk_ph2/cmsph2_tusb) : library handling the usb communication between the PC and the crate

## 1 Overview of the test system

This tutorial describes the steps to install the hybrid test system for the Phase 2 tracker of CMS. A schematic description of the test system is shown in Figure 1. The following instructions describe the installation of the connections between the PC and the multiplexing crate, the PC and the FC7 and the power supply of the crate.

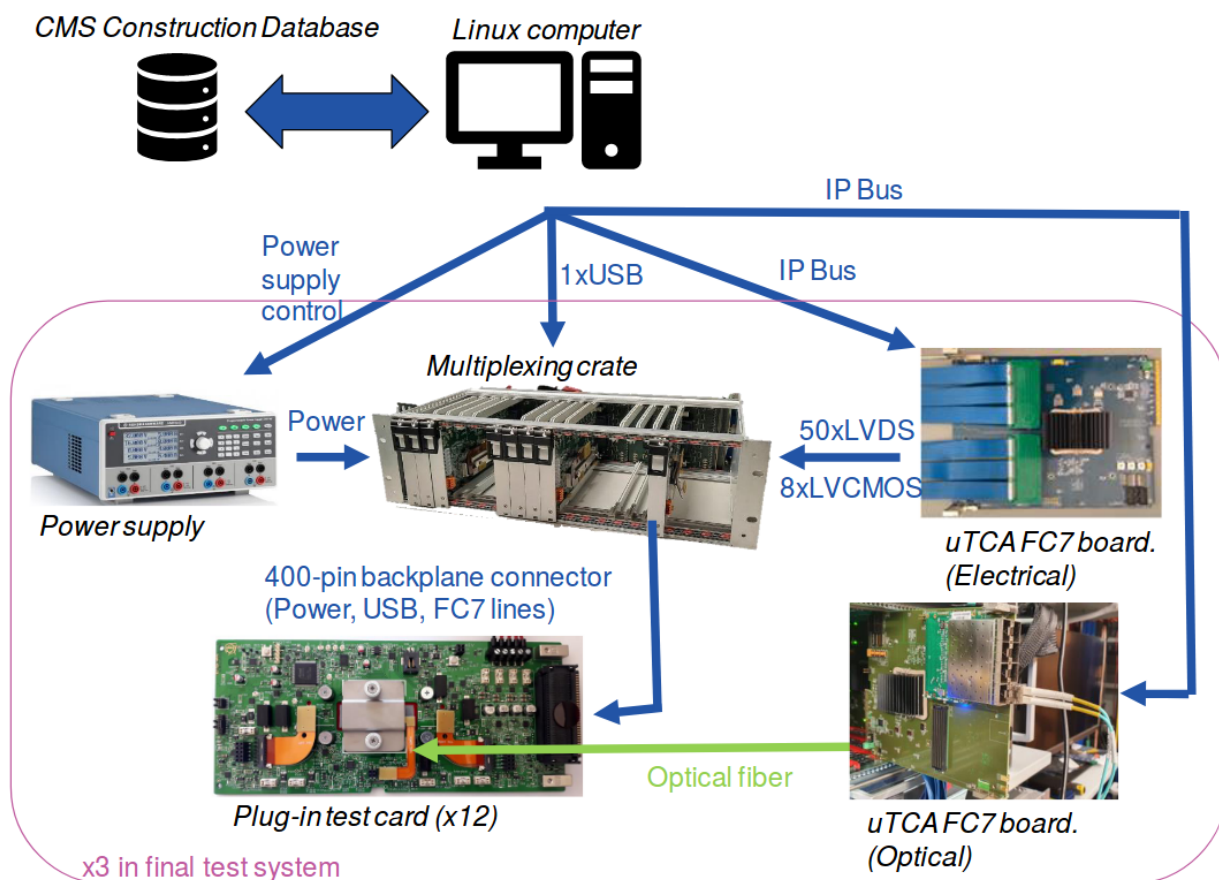


Figure 1: Scheme of the complete test system

Figure 2 shows the test system installed in Genova. The FC7 are hosted in the uTCA (top of the picture). The test crate is visible in the bottom of the picture.

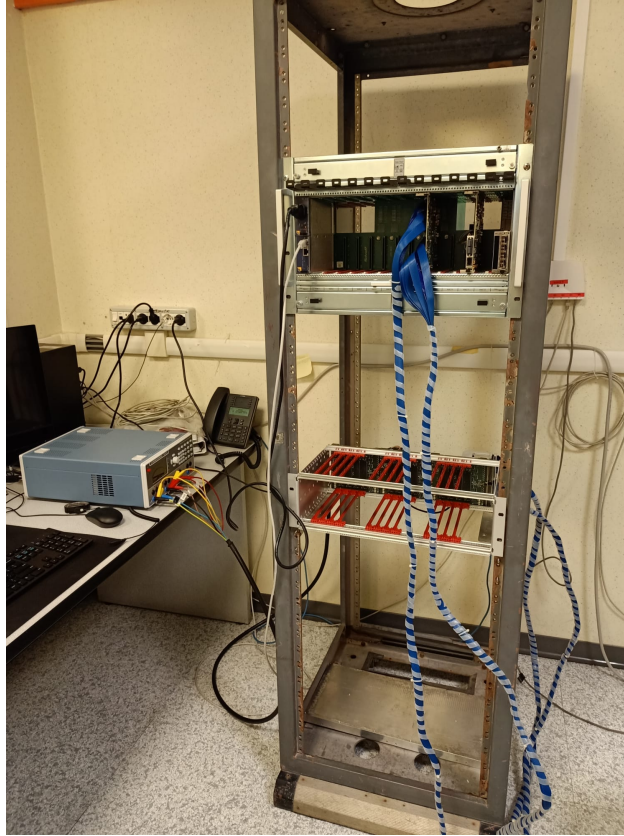


Figure 2: Overview of the test system in Genova

## 2 Hardware

In this section the steps to install the various hardware parts of the system are detailed.

### 2.1 Power cabling and power supply setting

The test crate is powered by a *ROHDE&SCHARTZ HMP4040* power supply. The power cable is connected following the schematic of Figure 3. The various power cables are clearly labeled as shown in Figure 4.

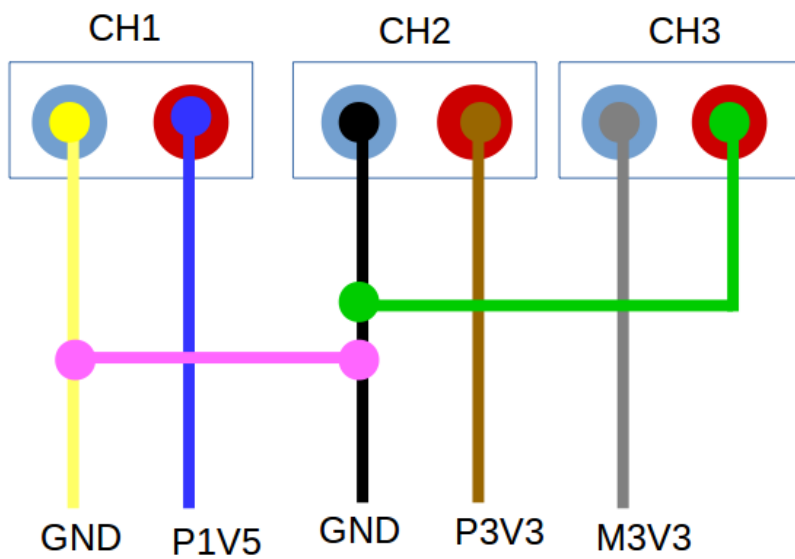


Figure 3: Schematic display of the connection of the crate power cable on the power supply.

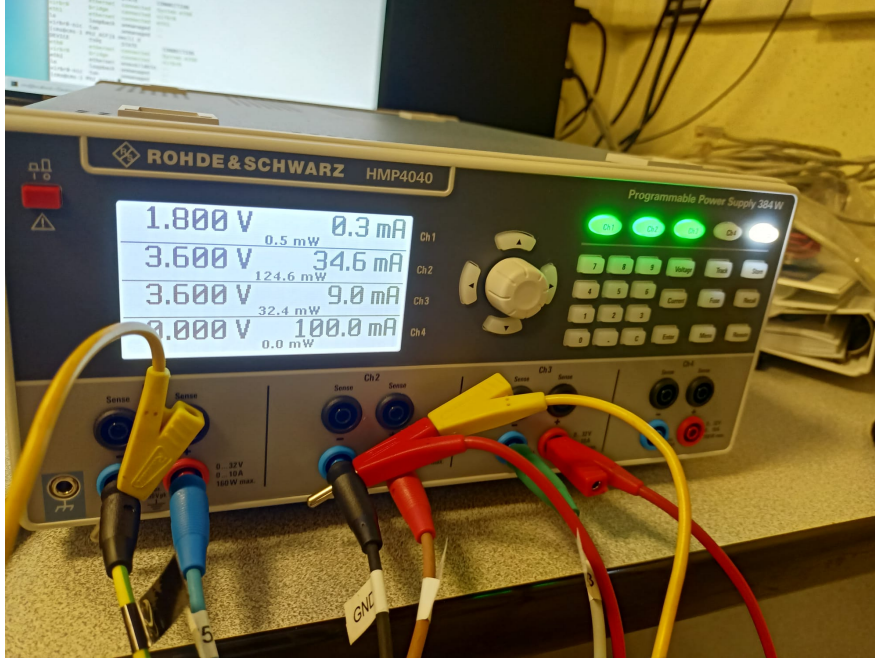


Figure 4: Fully configured power supply.

The voltages and current limit have to be set as indicated in Table 1. Once the voltage and current are set, the cable can be plugged in the back of the crate in any power plug, as shown in Figure 5.

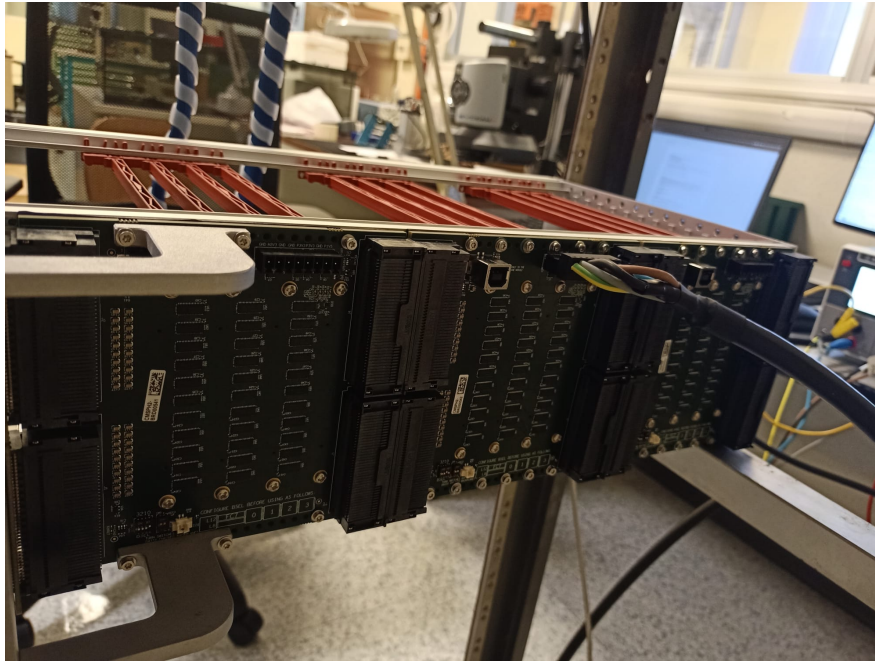


Figure 5: Picture showing the power cable connection to the crate. The cable can be plugged in any of the three backplanes (here it is on the middle one).

Channel	CH1	CH2	CH3
Pair	P1V5	P3V3	M3V3
V	1.8 V	3.5 V	3.6 V
$A_{max}$	2 A	4 A	1 A

Table 1: Voltage and current limit to apply to each channel of the power supply



## 2.2 Backplanes setting

In order for the test card multiplexing to work, each backplane has to be given a unique adress. This is possible using physical switches located on the back of every backplanes. The switches are located as shown on Figure 6. Each switch corresponds to a unique address from 0 to 3 and is activated when in its upward position. The addressing of each backplane must be done in the following way: the backplane hosting the FMC connections must be addressed 0, the following one 1, and the furthest from the FMC connector 2.

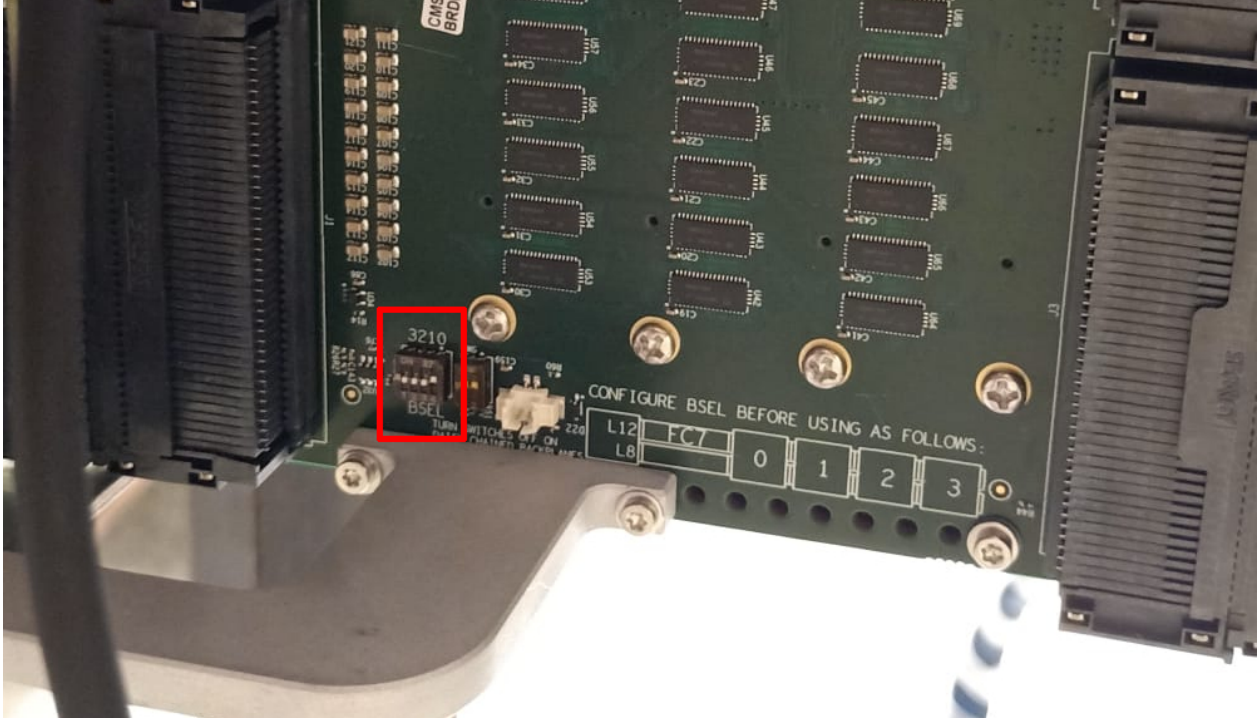


Figure 6: Switches used to provide a physical address to each backplane of a crate. The address corresponding to each switch is printed above each one of them, in this case the address is 0 corresponding to the backplane with the FMC connections to the FC7.

## 2.3 Data transmission cables

Once the power cable is correctly installed, the FMCs and usb cables have to be installed.

### 2.3.1 FMC cables connections

Two FMC cables link the test crate with the FC7 used to control the backplanes. The FC7 have two FMC connectors denoted L12 and L8, while the backplane connectors are denoted J1 and J2. The FMC cables have to be plugged as:

- L8 → J2
- L12 → J1.
- i.e. the upper part of the FC7 connected with the lower part of the mini-crate.

The blue flat canles should be connected to the backplane (bp) 0, as well as the USB cable.

Schematically the FMC connections are shown in Figure 7.

On both the FC7 and on the crate, the cables are secured using metallic pieces screwed in place. They can be seen on Figure 8 and 9. Finally, note that an adaptator is needed to plug the FMC cables on the crate. They can be seen in place on Figure 9.

### 2.3.2 Usb cable connection

The USB connection between the PC and the crate is achieved using a USB-A/USB-B cable. The USB-B plug is connected in the back of the first backplane (i.e. the closest to the FMC connectors) as shown in Figure 10. Note that no usb device will be seen by the PC until a test card is correctly configured.

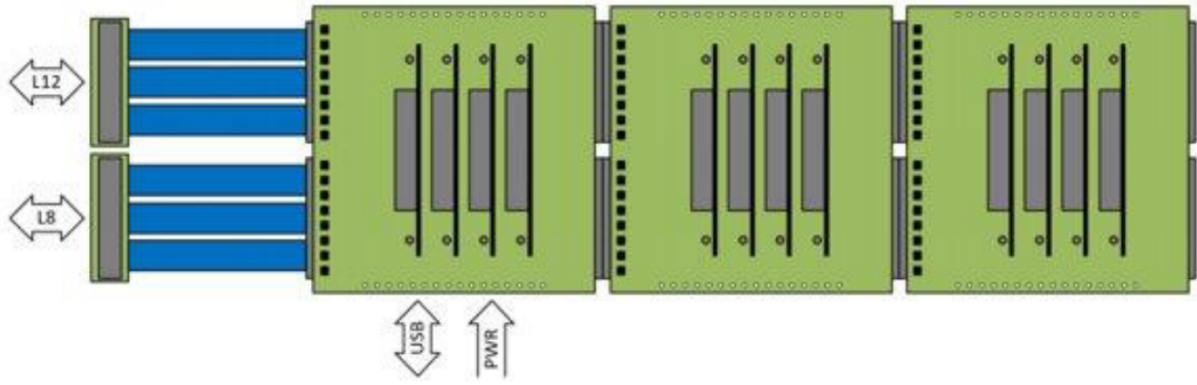


Figure 7: FMC Connection scheme

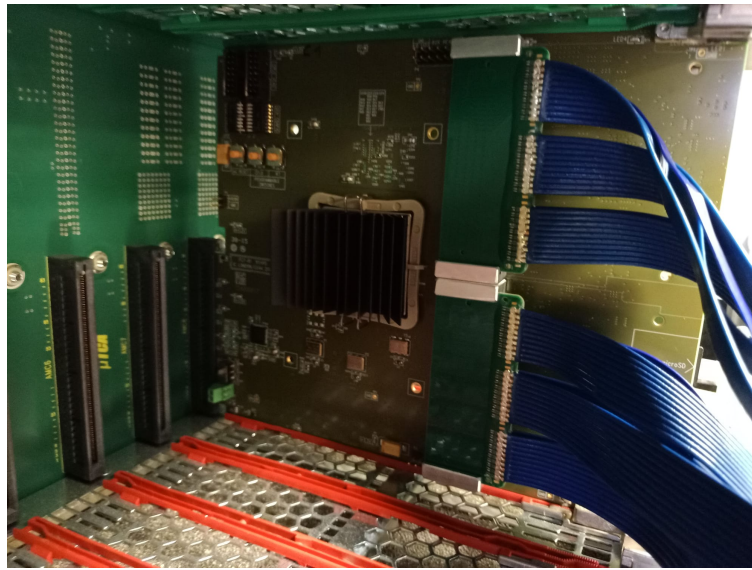


Figure 8: Picture of the FMC cable connection on the FC7. The connection is secured using rectangular metallic pieces, visible on the picture on each side of the connectors, which are screwed directly on the FC7.

## 2.4 AMC13 module

In order to provide a clock to the system the AMC13 module can be used (<http://www.amc13.info>). The module can be plugged as in Figure 11 and a loopback connection fiber must be plugged in. Note that even if the module is correctly powered no front panel led may be on.

## 3 Software

The following instruction complete the following tutorials: [https://cms-tracker-daq.web.cern.ch/cms-tracker-daq/tutorials/pc\\_connection/](https://cms-tracker-daq.web.cern.ch/cms-tracker-daq/tutorials/pc_connection/). Make sure the PC you use has two ethernet cards (enp1s0 and enp2s0). We will set the connection on enp1s0.

A few more details on how to prepare the FC7 and SD cards can be found here: <https://cernbox.cern.ch/s/stFuUdRMd07cxF6> or alternatively <https://www.ge.infn.it/~ferro/CMS/OTdocs/PreparingFC7-1.pdf>

Before proceeding install a rarp daemon as explained in [https://cms-tracker-daq.web.cern.ch/cms-tracker-daq/tutorials/pc\\_connection/](https://cms-tracker-daq.web.cern.ch/cms-tracker-daq/tutorials/pc_connection/).

### 3.1 Setting up the ethernet connection between the uTCA crate/FC7 and the PC

Once the rarp daemon has been installed and configured, do the following to verify the state of the enp1s0 interface and restart all the necessary services.

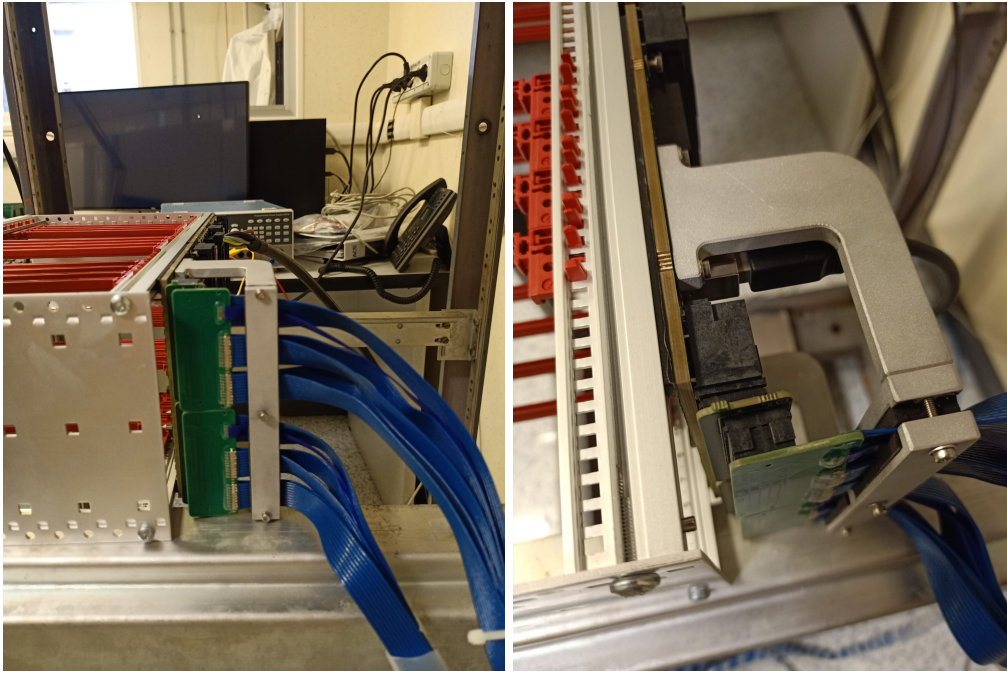


Figure 9: Picture of the connection of the FMC cable on the crate. The cables are secured using a long flat metallic piece with foam pad. The FMC adaptators are visible.

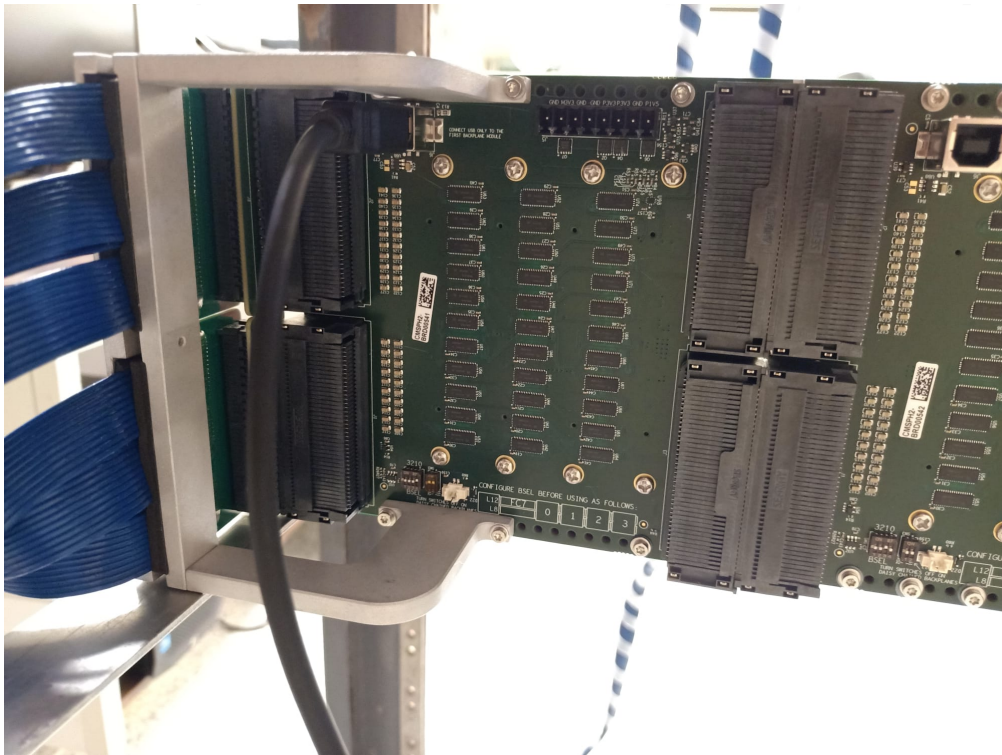


Figure 10: USB-B connection in the first backplane of the crate.

```
# Display the status of the ethernet interfaces of your PC
nmcli d

# Restart the various ethernet services
sudo systemctl daemon-reload
sudo systemctl restart rarpd
```



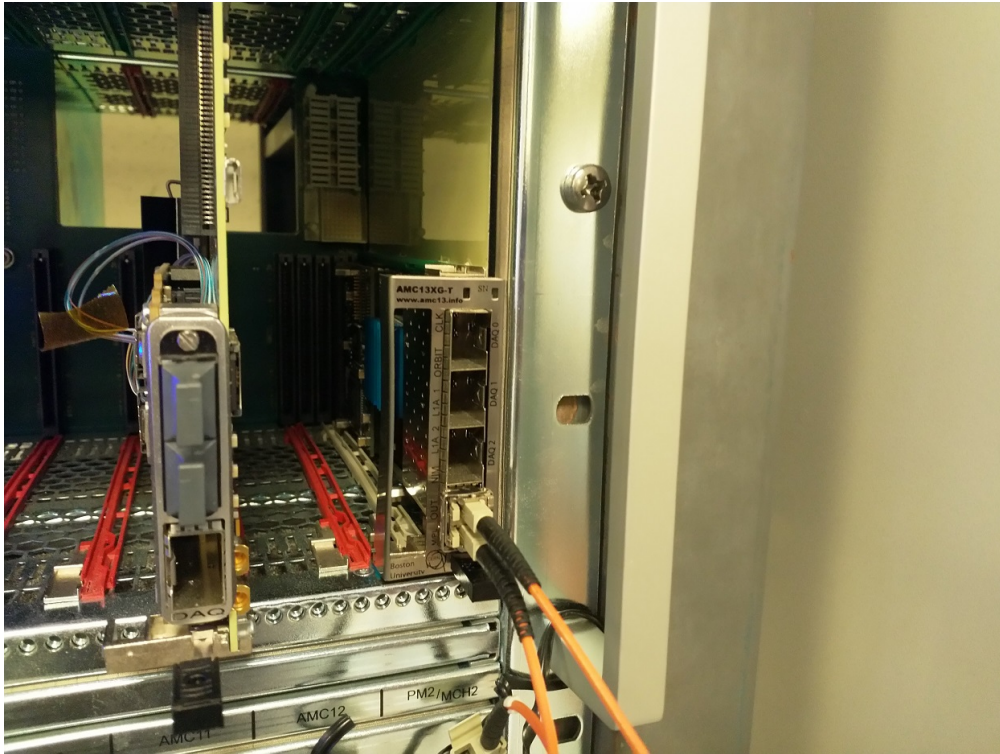


Figure 11: AMC13 module inside the crate.

```
sudo systemctl restart NetworkManager
```

Set an IP address for the enp1s0 interface

```
#Check the IP address of the ethernet interface
ifconfig
#the inet field of enp1s0 where the uTCA crate is plugged should show an IP
address, otherwise do the following line to provide an IP address to enp1s0
sudo ifconfig enp1s0 192.168.0.3
```

The IP address of the enp1s0 that we just configure as to be on the same network as the one of the FC7s. In other word with the enp1s0 address set to 192.168.0.1, the IP address of the FC7 have to be of the form 192.168.0.—

Also the subnet mask should be set to 255.255.0.0.

You can verify the state of the enp1s0 port using the following command:

```
# Verify that enp1s0 has state connected
nmcli d
```

If the ethernet port is correctly configured, the following output will be displayed:

In case, a FC7 is configured in the uTCA crate, one can "ping" it to verify the ethernet connection.

```
# ping an FC_7 if it is already installed
ping 'Name of your FC7'
```

DEVICE	TYPE	STATE	CONNECTION
enp2s0	ethernet	collegato	enp2s0
enp1s0	ethernet	collegato	enp1s0
lo	loopback	connected (externally)	lo

Figure 12: Screen display of the `nmcli d` command if the `enp1s0` port is well configured.

If the connection is established the output should look as in Figure 12, otherwise follow the instructions in the following section.

### 3.2 Adding a new FC7 to the uTCA crate

Switch off the uTCA crate, and plug the new FC7 in any available slot. Warning: to connect the FC7 it requires a bit of force, make sure it is well plugged. If the following phase fails, first verify the FC7 connection.

Before switching on the crate, open a separate terminal and do:

```
tshark -i enp1s0
```

This command listens to the `enp1s0` interface and displays all the packets going through it. Now the uTCA crate on, look for the broadcast of the newly connected FC7. It provides its MAC address. Note the MAC address before proceeding.

```
#add a line in /etc/ethers with the MAC and a name for the FC7 (any you like)
sudo vim /etc/ethers

#add a line in /etc/hosts with a IP address and the corresponding name
sudo vim /etc/hosts
```

The IP address of the new FC7 can be any address on the same sub-network as the ethernet port it is connected to : eg 192.168.0.— in our case. Obviously, it also has to be different from addresses already used in `/etc/hosts`.

```
ping 'FC7 IP address'
# or
ping 'FC7 name'
```

You should now be able to ping the new FC7.

```
[cms@cms-1 ~]$ ping fc7_ctl_1
PING fc7_ctl_1 (192.168.0.7) 56(84) bytes of data.
64 bytes from fc7_ctl_1 (192.168.0.7): icmp_seq=1 ttl=64 time=0.226 ms
64 bytes from fc7_ctl_1 (192.168.0.7): icmp_seq=2 ttl=64 time=0.210 ms
64 bytes from fc7_ctl_1 (192.168.0.7): icmp_seq=3 ttl=64 time=0.280 ms
64 bytes from fc7_ctl_1 (192.168.0.7): icmp_seq=4 ttl=64 time=0.206 ms
64 bytes from fc7_ctl_1 (192.168.0.7): icmp_seq=5 ttl=64 time=0.163 ms
```

Figure 13: Screen display of the ping command if the FC7 has been well configured.

Files `/etc/hosts` and `/etc/ethers` should look like in Fig. 14

### 3.3 Compiling Ph2\_ACF

Follow the instruction in [https://gitlab.cern.ch/cms\\_tk\\_ph2/Ph2\\_ACF](https://gitlab.cern.ch/cms_tk_ph2/Ph2_ACF). The README provides straight forward tutorial to compile the code. On Alma9, follow the instruction in the "Setup on RHEL 9.1 or AlmaLinux 9.1" section, and compile using the instruction in the "The Ph2\_ACF software" section.



```

127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.0.3    enp1s0
192.168.0.100   mch
192.168.0.7     fc7_ctl_1
#192.168.0.111  FED
192.168.0.104   fc7_roh_1
192.168.0.13    AMC13_T2
192.168.0.12    AMC13_T1
/etc/hosts (END)

00:10:18:31:7b:18    enp1s0
00:40:42:0b:29:59    mch
#08:00:30:00:22:da    FED
08:00:30:00:29:89    fc7_ctl_1
#08:00:30:00:22:e0    fc7_ctl_1
08:00:30:00:29:2d    fc7_roh_1
08:00:30:f3:01:ac    AMC13_T2
08:00:30:f3:01:ec    AMC13_T1
/etc/ethers (END)

```

Figure 14: Screen display of `/etc/hosts` and `/etc/ethers`.

Be careful to install all the required packages. For what concerns *protobuf* package beware that during the checks (*make check*) a big amount of memory is allocated, thus one of the checks can fail simply because of not enough available RAM.

After cloning the repository, be sure to select the branch you want. See the available tags with `git tag` and checkout the wished one with `git checkout < tagname >`, like v4-14.

### 3.4 Compiling TCUsb

Follow the instructions in [https://gitlab.cern.ch/cms\\_tk\\_ph2/cmsph2\\_tusb](https://gitlab.cern.ch/cms_tk_ph2/cmsph2_tusb). Once TCUsb has been compiled, you are asked to recompile *Ph2\_ACF*. Before doing so, change line 127 and 128 of the *Ph2\_ACF/setup.sh* file as:

```
export CompileWithTCUSB=true
```

### 3.5 Communicating with the FC7/Crate

Never forget to set your command line to *Ph2\_ACF* dir and run the setup script `. setup.sh` or `source setup.sh`.

#### 3.5.1 Provide FC7 IP address to the software

To communicate with the FC7 using *PH2\_ACF*, you need to provide the IP address of your FC7 to the program. This is done by changing the connection line in the xml description file (in our case *settings/PS\_ROH.xml*) as:

```

<connection id="board" uri="chtcp-2.0://localhost:10203?
target='address of your FC7':50001"
address_table="file://settings/address_tables/d19c_address_table.xml" />

```

and then do:

```
/opt/cactus/bin/controlhub_start
```

Note that if you have more than one FC7 board you need to provide a different (0, 1, 2, ...) board Id for each FC7.

### 3.5.2 Writing the right firmware to the FC7 boards

Firmware of the OT tracker can be download here: <https://udtc-ot-firmware.web.cern.ch/> Someone should tell you which is the latest version of the firmware for your hw configuration.

```
#Go in Ph2_ACF folder
cd Ph2_ACF #or Ph2_ACF_Latest or others depending on the latest installed version

#Display the available firmware on the SD card of the FC7
fpgaconfig -c settings/PS_ROH.xml -l

#Upload firmware from the PC to the FC7 SD card
fpgaconfig -b 'boardId' -c settings/PS_FEH_mainTest_test1.xml
-f 'firmware_file_name_on_the_PC.bin'
-i 'firmware_name_on_the_microSD'

#for the electrical FC7 (the one with the blue falt cables)
fpgaconfig -b 1 -c settings/PS_ROH.xml -f FC7_firmwares/ps_roh_5g_electrical.bin
-i ps_roh_5g_electrical

#for the optical FC7 (the one with the fibers)
fpgaconfig -b 0 -c settings/PS_ROH.xml -f ps_12m_5g_cic1_l12octa_l8quad.bin
-i ps_12m_5g_cic1_l12octa_l8quad

#Upload a firmware of the FC7 fpga
fpgaconfig -b 0 -c settings/PS_ROH.xml -i ps_12m_5g_cic1_l12octa_l8quad
fpgaconfig -b 1 -c settings/PS_ROH.xml -i ps_roh_5g_electrical
```

```
[cms@cms-1 Ph2_ACF]$ fpgaconfig -c settings/PS_ROH_el.xml -b 0 -i ps_12m_5g_cic1_l12octa_l8quad
14.07.2023 15:40:32: |140194246577280|I| Loading ps_12m_5g_cic1_l12octa_l8quad into the FPGA...
14.07.2023 15:40:35: |140194246577280|I| Firmware image: ps_12m_5g_cic1_l12octa_l8quad loaded on FPGA
14.07.2023 15:40:35: |140194246577280|I| >>> Done <<<
```

Figure 15: Screen display if the firmware has been correctly uploaded on the fpga

### 3.5.3 Communication with the mini-crate and test boards

First, a scan of the available test cards has to be done:

```
#Scan the crate, and display the slot occupied by a test card
mux_setup -f settings/PS_ROH.xml --mux_scan
```

From the output (see figure) one can see how many cards are connected to each backplane in which position. In order to configure the connection one can run

```
#Configuring card 1 in backplane 0
mux_setup -f settings/PS_ROH.xml --mux_configure 0,1
```

In order to setup the optical connection of the FC7 with the test card, one must identify the USB device via *lsusb* command looking for a line like

```

29.08.2023 16:20:31: |140543984221312|I| Setup is scanned
29.08.2023 16:20:31: |140543984221312|I| Available cards for bp 0:[ 1 ]
29.08.2023 16:20:31: |140543984221312|I| Available cards for bp 1:[ No cards]
29.08.2023 16:20:31: |140543984221312|I| Available cards for bp 2:[ 2 ]
29.08.2023 16:20:31: |140543984221312|I| *** End of the operation ***

```

Figure 16: Screen display after a crate scanning. Two test cards are found.

```
Bus 001 Device 005: ID 10c4:87a0 Silicon Labs PSRV2-CMSPH2-BRD00701
```

and then run the corresponding command

```

#device number should be checked with lsusb
#to check connectivity
PSROHTest -f settings/PS_ROH.xml -b --USBBus 1 --USBDev 5

#run the full test
PSROHTest -f settings/PS_ROH.xml --ep 0xCA --test-fcmd --test-clock --test-i2c 1000 \
--test-eom --test-vtrx --test-reset --hybridId CMSPH2-PRT00193 -b \
--test-eom --measure-input-iv --USBBus 1 --USBDev 5

```

All the channels are tested. At least one should be seen working. If all the channels fail there is a problem in the transmission/receiving of the optical signal.<sup>1 2</sup>

### 3.5.4 Using AMC13 clock

In order to have more reliable results, the clock provided by the AMC13 module should be used, instead the internal one. The AMC13 module should first set up and enabled. What follows has been extracted by the code and notes in gitlab repository at <https://gitlab.cern.ch/cms-ot-hybrids/amc13-mch-configuration-scripts>.

First thing to do is to provide two IP addresses to the two tongues of the module T1 and T2. The procedure is the same as that used for adding a new FC7: use tshark to spot the MAC addresses of the two tongues and then edit `/etc/ethers` and `/etc/hosts`. The rule of thumb to identify T1 wrt T2 is that the one that keeps talking is T1. T1 and T2 IP addresses should differ by 1 digit (see Figure 14).

In order to be able to configure the AMC13, one can have a look at [https://gitlab.cern.ch/cms-ot-hybrids/amc13-mch-configuration-scripts/-/blob/master/Getting\\_started\\_with\\_AMC13\\_module.pdf?ref\\_type=heads](https://gitlab.cern.ch/cms-ot-hybrids/amc13-mch-configuration-scripts/-/blob/master/Getting_started_with_AMC13_module.pdf?ref_type=heads) (note that the instructions are for Centos7 and that some of the required packages may already be installed. Note also that the PATHs are sometimes referred to the home `~` directory instead of the actual work dir. **Forget** about the IP assignment algorithm unless strictly needed for some reason).

After having installed the AMC13 code inside the `amc13` dir, edit `connectionSN43.xml` with the right IPs and run

```

. env.sh
tools/bin/AMC13Tool2.exe -c amc13/etc/amc13/connectionSN43.xml

```

If the AMC13 is connected no error will be reported and with the `list` command it will be shown as in Figure 17. In order to configure the AMC13 the following line can be entered instead of the `list` command:

```

> en 1-12 f t
> exit

```

<sup>1</sup>FC7 version has to be checked.

<sup>2</sup>If an optical mezzanine is missing channel 0 might be seen as locked (!). Disable these channels to prevent errors.



```
[cms@cms-1 amc13]$ tools/bin/AMC13Tool2.exe -c amc13/etc/amc13/connectionSN43.xml
No address table path specified.
Using .xml connection file ...
Using AMC13 software ver:0
Read firmware versions 0x224e 0x2e
flavor = 2 features = 0x000000b2
>list
Connected AMC13s
*0: SN: 211 T1v: 224e T2v: 002e cf: amc13/etc/amc13/connectionSN43.xml
>exit
```

Figure 17: Checking AMC13 connection.

The HW needs now to be instructed to use the clock provided by the AMC13 module. The file settings/PS.ROH.xml needs to be modified setting the clock\_source\_u7 register to 0 for each board:

```
<Register name="clock_source_u7">0</Register>
```

a new file can be created (settings/PS\_ROH\_AMC.xml) and it can be used to configure the boards and rerun the test:

```
# e.g.
mux_setup -f settings/PS_ROH_AMC.xml --mux_configure 0,2
PSROHTest -f settings/PS_ROH_AMC.xml --ep 0xCA --test-fcmd --test-clock --test-i2c 1000 \\\
--test-eom --test-vtrx --test-reset --hybridId CMSPH2-PRT00193 -b --test-eom \\\
--measure-input-iv --USBBus 1 --USBDev 46
```

### 3.6 Installing and using the GUI

In order to install the GUI, see the repository in gitlab

<https://gitlab.cern.ch/imateosd/gui-for-hybrids-testing-on-crate-systems>

Don't forget to edit *configuration.json* file if needed.

The commands to run the GUI are:

```
#inside gui-for-hybrids-testing-on-crate-systems
source setup.sh
python3 cratetesting.py
```

The first time the GUI is used, the file hybrid/PSROH.py might need to be updated. Especially the following variables:

- PH2\_ACF\_WORKING\_DIRECTORY
- DESC\_FILES
- FIRMWARE

configuration.json file must also be updated.

The following parameters need to be changed carefully:

- default\_Ph2\_ACF\_directory : the directory where Ph2\_ACF is installed
- default\_fw\_image : the firmware image to be installed on the electrical FC7

```

# CONFIGURATION VARIABLES
PH2_ACF_WORKING_DIRECTORY=" ../Ph2_ACF_Latest/"
DESC_FILES = [ ["settings/PS_ROH.xml"],
                ["settings/D19CDescription_PS.xml"],
                ["settings/D19CDescription_PS.xml"] ]
# SERIAL_NUMBER_VALIDATOR = "^PS(-)?ROH\d{2}-\d{9}$"
SERIAL_NUMBER_VALIDATOR = ".*"
TEST_PROCEDURE = 'PSROHTest'
TEST_ARGUMENTS = [ ['--test-external-pattern', '0xCA', '--test-fcmd', '--test-clock', '--test-reset', '--test-i2c', '100', '--measur
e-input-iv', '-b'] ] # Removed '--test-vtrx', '--test-eom',
FIRMWARE = [ ["ps_12m_5g_cic2_l12octa_l8quad", "ps_12m10gcic2_l12ol8q_v1_1.bin", "ps_roh_5g_electrical", "ps_10g_el_preIreneMrg.bin"
],
              #Crate 1: [Optical for 5G, Optical for 10G, Electrical for 5G, Electrical for 10G]
              ["ps_12m5gcic2_l12ol8q_v1_1.bin", "ps_12m10gcic2_l12ol8q_v1_1.bin", "ps_5g_elec_TC_v1_1.bin", "ps_10g_el_preIreneMrg.bi
n"],
              #Crate 2: [Optical for 5G, Optical for 10G, Electrical for 5G, Electrical for 10G]
              ["ps_12m5gcic2_l12ol8q_v1_1.bin", "ps_12m10gcic2_l12ol8q_v1_1.bin", "ps_5g_elec_TC_v1_1.bin", "ps_10g_el_preIreneMrg.bi
n"] ]
              #Crate 3: [Optical for 5G, Optical for 10G, Electrical for 5G, Electrical for 10G]

```

Figure 18: Screen display of a portion of hybrid/PSROH.py

```

{
  "default_ph2_acf_directory": " ../Ph2_ACF_Latest",
  "default_fw_image": "ps_roh_5g_electrical",
  "default_settings_files": [
    [
      "settings/PS_ROH_elOnly.xml"
    ]
  ],
  "default_serial_number_validator": "((((((PS)|(2S))-?(((FEH[0-9]{2}|R|L))|ROH[0-9]{2}|SEH|POH)))|(8CBC3((18)|(40))))-[0-9]{9}))|(CMSP
H2-PRT[0-9]{5}))",
  "number_of_crates": 1,
  "number_of_hybrids": 1,
  "vendor_mode": true,
  "db_target": "development",
  "db_run_type": "VAT",
  "location": "Genova",
  "institute": "INFN",
  "tester": "Fabrizio Ferro",
}

```

Figure 19: Screen display of a portion of configuration.json.

- default\_settings\_files : a HW configuration xml file done on purpose where the electrical FC7 is board 0. One can get PS.ROH.xml, rename it, delete the optical FC7 part and change the board number from 1 to 0.