

Pixel Phase 1 DAQ Documentation

Contents

1 Phase 1 DAQ Test Stand	5
1.1 Minimal set of components needed for the Phase 1 DAQ Test Stand	5
1.1.1 Useful tools for debugging and setting up	6
1.1.2 Software tools	6
1.2 Setting up infrastructure	7
1.2.1 Getting started	7
1.2.2 Preparing FC7 cards	7
2 AMC13	9
2.1 AMC13 prerequisites	9
2.2 Setting everything up for the test stand	9
2.2.1 TTC Configuration for test stand	9
2.2.2 AMC13 Configuration	9
2.2.3 Pixel Test Stand Configuration for standalone AMC13	10
3 Front End Driver (FED)	11
3.1 FED configuration	11
3.2 FED performance measurements	11
4 Pixel Front End Controller (pixFEC)	13
4.1 FEC configuration	13
5 Tracker Front End Controller (tkFEC)	15
5.1 FEC configuration	15
6 Operating the DAQ System	17
6.1 Setting up the CCU and Portcard	17
6.2 Running the microTCA FED	17
7 Debugging instructions	19
7.1 LEDs	19
7.1.1 FC7 revision2	19
7.1.2 FMC cards	20
Appendices	25

Chapter 1

Phase 1 DAQ Test Stand

This chapter gives a brief summary about all the components necessary to setup a test stand for the Pixel Phase 1 data acquisition system (DAQ). It is shown how all components are connected. The specifications of the shown parts and how to run them can be found in their respective chapter.

1.1 Minimal set of components needed for the Phase 1 DAQ Test Stand

For the DAQ system

- μ TCA Crate – hosts all the μ TCA cards, distributes signals over back plane
- MCH – MicroTCA Carrier Hub, manages all cards in crate
- AMC13 – interface to CMS TCDS (trigger & clock distribution system)
- Optical clock source – small card that creates a 40.08MHz clock
- FC7/CTA – CMS Tracker AMC, base board for most DAQ components
 - tkFEC – Tracker Front End Controller, sends control signals to the CCU board
 - pixFEC – Pixel Front End Controller, sends control signals to the detector modules
 - FED – Front End Driver, receives signals from the detector modules

For the test bench

- PortCard – equipped with POHs (Pixel Opto-Hybrids), conversion from electrical to optical signals
- DOH mother board – Digital Opto-Hybrids, conversion from optical to electrical signals
- CCU board – Communication & Control Unit
- Pixel Module
- Fibre adaptors
- Optical fibres
- A diverse set of power supplies and maybe an oszilloscop

1.1.1 Useful tools for debugging and setting up

- uSD card reader for JTAG-free CTA FW management
- Xilinx USB cable for debugging (FW upload)
- Lab PC with 2 Ethernet ports (SLC6)
- 1Ghz Oscilloscope
- opto-electrical converter for debugging of CLK lines
- differential probe, single ended probes for debug FMC on FED

1.1.2 Software tools

- Wireshark for spying on TCP traffic & determining MAC addresses (via yum)
- ic_mmc tools (imgtool for FW upload to SD card) (see FC7 user guide)
- AMC13 development tools (to set RARP mode on AMC13)
- natVIEW for crate-backplane configuration
- IPMItool (via yum)
- FED CMD line tools (https://github.com/gauzinge/FED_burnin/tree/FIFO)
- modified FEC SW including modified BPixelTools (available on FEC SW SVN)

1.2 Setting up infrastructure

1.2.1 Getting started

- uTCA Shelf, MCH & Power module should be plug & play
- MCH has a default IP address that can be changed via web interface (see NAT documentation)
 - simplest solution is to use a static IP & subnet mask
 - could also use DHCP server running on local lab PC for IP assignment (similar to P5 but maybe overkill)
- Lab PC needs to have 2 ethernet ports (1 for global network, one for local uTCA LAN)
 - configure the local LAN port to have static IP & subnet same as MCH
 - install uHAL (cactus.web.cern.ch)
 - install RARPD (daemon) for AMC13 / CTA IP assignment
- verify that you can ping the MCH (a USB format serial cable delivered with MCH can be used to connect via miniCOM terminal application in case ethernet does not behave at all)

1.2.2 Preparing FC7 cards

Before plugging the card in

- FC7 manual, although not complete, provides enough pointers to get started: <https://svnweb.cern.ch/cern/wsvn/ph-ese/be/fc7/trunk/fc7/doc/>
- 2 configurations to be done on the board:
 - set the powering switch to crate operation
 - push the switches 1 & 5 of the CPLD DIP switch down (includes FGPA in JTAG chain, sets FW loading from SD card after power on)
- carefully insert card in crate - attention, the switch that is actuated by the hot-swap handle is rather fragile
- if the card is properly seated in the crate, the blue LED at the bottom should light up and the corresponding LED on the MCH should blink
- insert the hot-swap handle - the card should power up & load the FW image from SD card (see 1.2.2)

Once the card is plugged in

- use an external card reader to format the SD card following the instructions from the FC7 manual p13
- upload an FC7 GoldenImage and / or a pixel specific image following the instructions
- caveat 1: one image must always be called “GoldenImage.bin” - this is what gets loaded on power-up
- caveat 2: FW files *must* be .bin format

Network configuration with RARP

- all pixel-specific FW uses RARP mode for the CTA
- MAC address is easily found out using wireshark
- requires edit of /etc/ethers & /etc/hosts & ‘sudo /etc/init.d/rarpd restart’
- arp -e command is your friend
- see <https://twiki.cern.ch/twiki/bin/viewauth/CMS/Ph1PixelDAQTest>

Chapter 2

AMC13

2.1 AMC13 prerequisites

- TTC links from AMC13 to AMCs needs to be enabled explicitly on per-slot basis
- AMC13 needs a LHC-like input (clock) signal for QPLL to lock:
 - internal 40MHz oscillator does not work
 - either TTC data-stream from TTCCi (80.16 MHz bi-phase encoded)
- external (BU) clock board (80.16 MHz clock)
- these need to be provided via the TTC SFP (Rx)
- AMC13 TTC simulator mode will not work for pixels

2.2 Setting everything up for the test stand

2.2.1 TTC Configuration for test stand

- we usually use the following TTCCi config for our uTCA system:
 - BC0 every orbit
 - BGO Ch15: 0x2c - charge inject @ BX 101, repetitive, optional prescale
 - cyclic L1A: BX 199 (98 BX after the charge inject), repetitive, optional prescale

2.2.2 AMC13 Configuration

- AMC13 offers 4 BGO channels (BC0 is sent automatically every orbit)
 - can configure any command and send single or cyclic BGOs (orbit prescale is possible)
 - this is rather flexible
 - EC0 and OC0 can be user enabled
- L1A is available in 3 modes:
 - 0: periodic triggers spaced by rate orbits at BX=500
 - 1: periodic triggers spaced by rate BX
 - 2: random trigger at 2 * rate Hz
- these are configured with rate and rules (`amc13.info`) attributes
- this config can be used to either send burst triggers (also single) or continuously

2.2.3 Pixel Test Stand Configuration for standalone AMC13

- restrictions on BGO & L1A config leave 1 possible config for the AMC13
- BGO Ch0: cmd=0x2c, repeat=1, prescale=0, BX=380 (sent 20 BX later than that, makes BX 400)
- L1A: mode=0, rate=1 burst=1 rules=0
 - this equals to: 1 L1A every orbit @ BX 500, 1 BGO (0x2c) every orbit at BX 401
 - 20 BX delay for the BGO have to be considered
- L1A is rescaled via ‘rate’ attribute, BGO via ‘prescale’ attribute

Chapter 3

Front End Driver (FED)

3.1 FED configuration

- 1 CTA + 1 FITEL Rx FMC (in upper slot for current PixFED beta FW - in lower slot in the future)
- Xilinx FMC105 supported on the other slot for debugging (https://indico.cern.ch/event/482128/contribution/0/attachments/1211056/1766370/Phase_1_update_13_01_2016.pdf for Pin mapping)
- current FW is rather basic (phase finding, FE DESER block, basic BE readout functionality)
 - single-event readout mode / TTS / C-DAQ implementation in progress
 - photocurrent measurement via ADC on FMC implemented
 - OSD read back functionality implemented
- integration in POS ongoing - standalone CMD line tool available: https://github.com/gauzinge/FED_burnin/tree/FIFO - gcc > 4.7 required!!
- caveat: uTCA FED needs male MPO on input fibre (different from digital VME FED)

3.2 FED performance measurements

Here we could collect some results and 'How-to's for the performance measurements for the FEDs

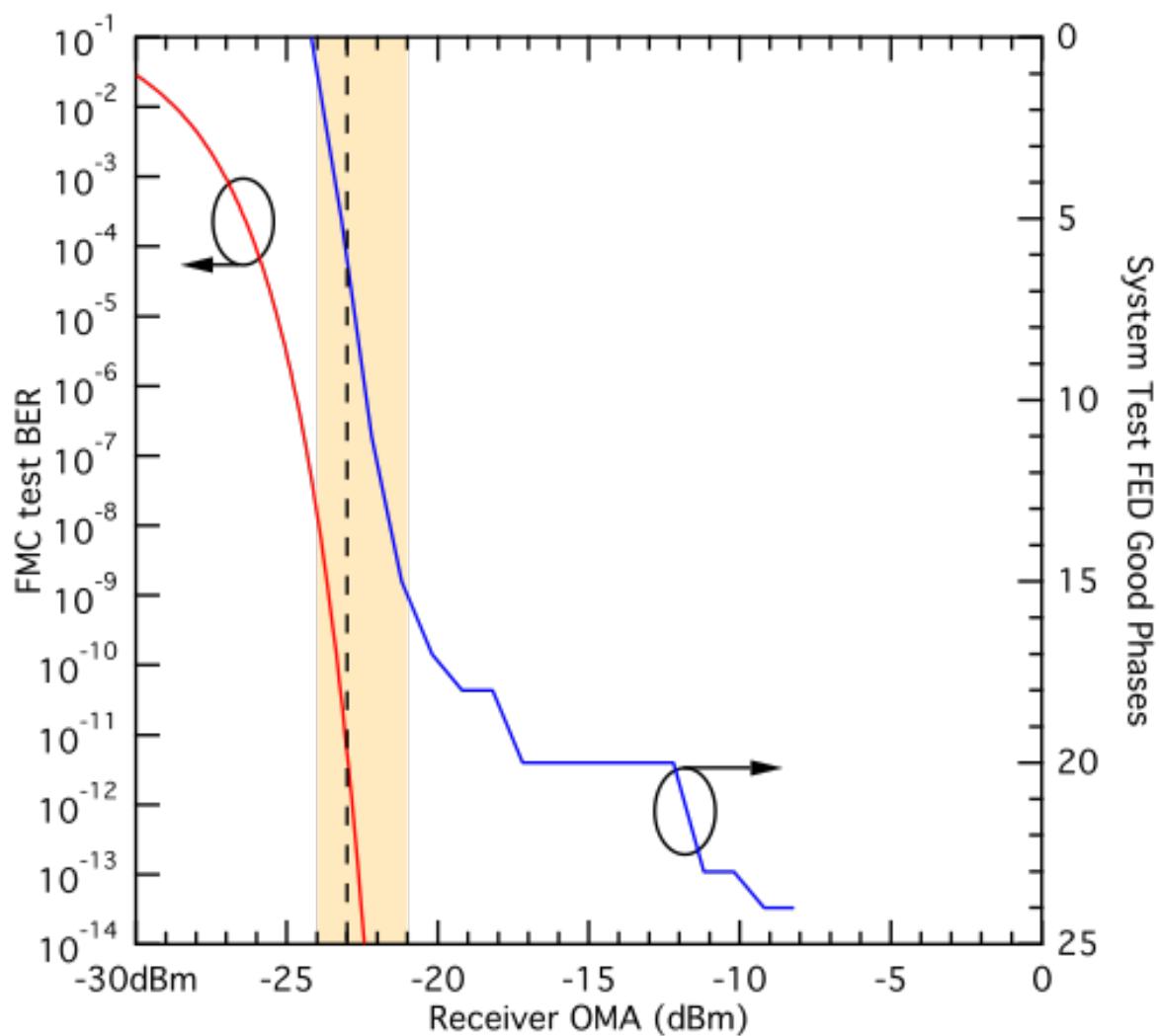


Figure 3.1: Number of good phases vs optical signal attenuation

Chapter 4

Pixel Front End Controller (pixFEC)

4.1 FEC configuration

- TkFEC: modified version of FEC SW available on SVN (make Generic, make APIConsoleDebugger for CMD line operation)
 - https://svnweb.cern.ch/cern/wsvn/cmsptdaqup/CMS_UPGRADES_IPHC/SOFTWARE/PHASE_1/CTRL/tags/TK_FEC_FMC8SFP_4CTRLRING_V01/?#a9e50ea3b3bff0b870e7f5f38e685e2a
 - <https://twiki.cern.ch/twiki/bin/viewauth/CMS/AddingAnUtcaRingDevice>
- PxFEC: SW adaptation in progress (no “release” yet)
- Ring 0 / Link 0 is on the bottom of bottom FMC
- Fibre mapping for 1 FMC can be seen in 5.1

SFP left				SFP right	Ring pxFEC	Ring tk FEC
H (Clock)	Rx			Tx	G (Data)	Ring 2 B
	Tx			Rx		Link 4
F (Clock)	Rx			Tx	E (Data)	Ring 2 A
	Tx			Rx		Link 3
D (Clock)	Rx	purple	orange	Tx	C (Data)	Ring 1 B
	Tx	green	yellow	Rx		Link 2
B (Clock)	Rx	light blue	orange	Tx	A (Data)	Ring 1 A
	Tx	dark blue	red	Rx		Link 1

Figure 4.1: FEC fibre mapping

Chapter 5

Tracker Front End Controller (tkFEC)

5.1 FEC configuration

- TkFEC: modified version of FEC SW available on SVN (make Generic, make APIConsoleDebugger for CMD line operation)
 - https://svnweb.cern.ch/cern/wsvn/cmsptdaqup/CMS_UPGRADES_IPHC/SOFTWARE/PHASE_1/CTRL/tags/TK_FEC_FMC8SFP_4CTRLRING_V01/?#a9e50ea3b3bff0b870e7f5f38e685e2a
 - <https://twiki.cern.ch/twiki/bin/viewauth/CMS/AddingAnUtcaRingDevice>
- PxFEC: SW adaptation in progress (no “release” yet)
- Ring 0 / Link 0 is on the bottom of bottom FMC
- Fibre mapping for 1 FMC can be seen in 5.1

SFP left				SFP right	Ring pxFEC	Ring tk FEC
H (Clock)	Rx			Tx	G (Data)	Ring 2 B
	Tx			Rx		Link 4
F (Clock)	Rx			Tx	E (Data)	Ring 2 A
	Tx			Rx		Link 3
D (Clock)	Rx	purple	orange	Tx	C (Data)	Ring 1 B
	Tx	green	yellow	Rx		Link 2
B (Clock)	Rx	light blue	orange	Tx	A (Data)	Ring 1 A
	Tx	dark blue	red	Rx		Link 1

Figure 5.1: FEC fibre mapping

Chapter 6

Operating the DAQ System

This section will show how one can run and operate the system to do basic tests. For now it only consists of the basics tools we developed at CERN but as soon as possible POS should also be integrated.

The text follows the instructions found on Mia Liu's twiki <https://twiki.cern.ch/twiki/bin/viewauth/CMS/PixelPhase1SetupMicroTCA>.

All examples are made for the TIF test stand for now but should be kept as generic as possible.

6.1 Setting up the CCU and Portcard

First make sure the correct environment variables have been picked up. On `cmsuppixpc001` you would do for example

```
source /setup.sh
```

We are working in the following with the BPixelTools version that can run on the microTCA system. Change the directory there and also source the needed environment variables.

```
cd /FEC_SW/Pixel/BPixelTools/  
source setenv.sh
```

Then change to the CCU directory

```
cd ccu
```

From here we configure the CCU by running

```
bin/ccu -utca /FEC_SW/connections.xml -fechardwareid board -ccu 0x3f  
-channel 0x11
```

This will open an interactive shell where first

```
scanccu
```

and

```
scanringdevice
```

needs to be run. If everything worked fine and there were no errors the Portcard can be initialised by running the script

```
Init4ChPortcard_uTCA.sh
```

To control the sending of triggers with the TTCci system we have to start HyperDAQ. To do this, change to the directory

```
cd /home/fectest/FEC_mTCA/pixel/PixelRun
```

There run the appropriate shell script

```
./run_noRU_TIF.sh
```

From the browser `IStillNeedTheURL` you can control the TTCci system then.

6.2 Running the microTCA FED

Chapter 7

Debugging instructions

7.1 LEDs

The first step in any debugging operation should be to look at the LEDs of all the system components and to check that everything is running fine. This section will cover the LEDs of the FC7 card and the FMC cards for the pixFED, pixFEC and tkFEC.

7.1.1 FC7 revision2

Since revision2 is the final production board for the Pixel Detector Phase1 upgrade, we will describe it here. Earlier revisions are similar, with a slightly different placement of voltage indicator LEDs on the side of the FC7 card.

FC7 - voltage indicators

As seen in figure 7.1 the LEDs indicating different supply voltages are distributed over the side of the PCB board. All LEDs have a label next to them what makes it easy to identify what voltage they show. All LEDs should be **ON**.

FC7 - MMC heart beat

The LED labeled **LED4** indicates the heart beat of the Module Management Controller (MMC). This LED should be flashing.

FC7 - front panel

The FC7 is equipped with six LEDs on the front panel. Four of those LEDs have a defined behaviour on all FC7 cards. The remaining two LEDs are user defined and specific to whether the FC7 is configured as pixFED, pixFEC or tkFEC. The naming convention of the LEDs can be found in figure 7.3. The purpose of the four standard LEDs can be seen in figure 7.2. For the different flavours of FC7 cards in the Pixel DAQ system, the purpose of the user defined LEDs is summarized in table 7.1.

Table 7.1: User defined LEDs on FC7 front panel

FC7 flavour	TOP User LED	BOTTOM User LED
pixFED	unused	toggles with L1A triggers
pixFEC	PLL locked	unused
tkFEC	unknown	unknown

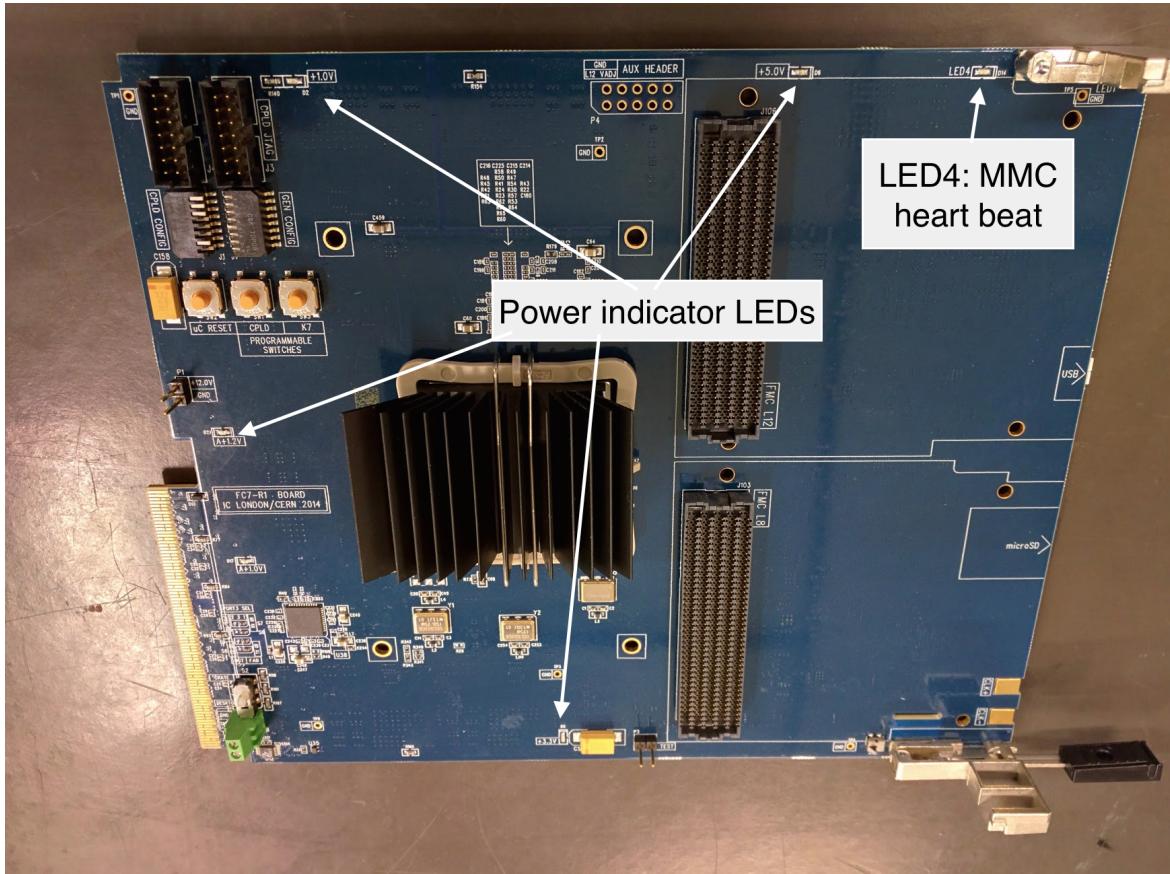


Figure 7.1: Side-view of the FC7 board

7.1.2 FMC cards

The two different FMC cards that make a FC7 board to either a FED or a FEC have a number of LED added to them.

FED FMC

The FED FMC shown in figure 7.4a is equipped with 7 LEDs. None of them are currently used. All LEDs are ON.

FEC FMC

Tracker FEC and Pixel FEC use the same FMC card. Two LEDs on the card indicate the status of the +3.3V and the V_{adj} power supply, see figure 7.4b

LED	COLOUR	STATUS
CPLD LED	RED	FPGA in reset
	GREEN (blink)	CPLD clock
	GREEN	FPGA configured
	BLUE	SPI PROM configuration mode
SYS LED TOP	RED	Golden Image Firmware loaded
	BLUE (blink)	IPBus 1Hz clock
SYS LED BOTTOM	RED	CDCE PLL not locked
	ORANGE	CDCE TTC clock out of phase
	GREEN	CDCE ok
MMC LED	GREEN	Loading firmware from SD

Figure 7.2: Purpose of the 4 standard LEDs on the FC7 front panel



Figure 7.3: Front-view of the FC7 board

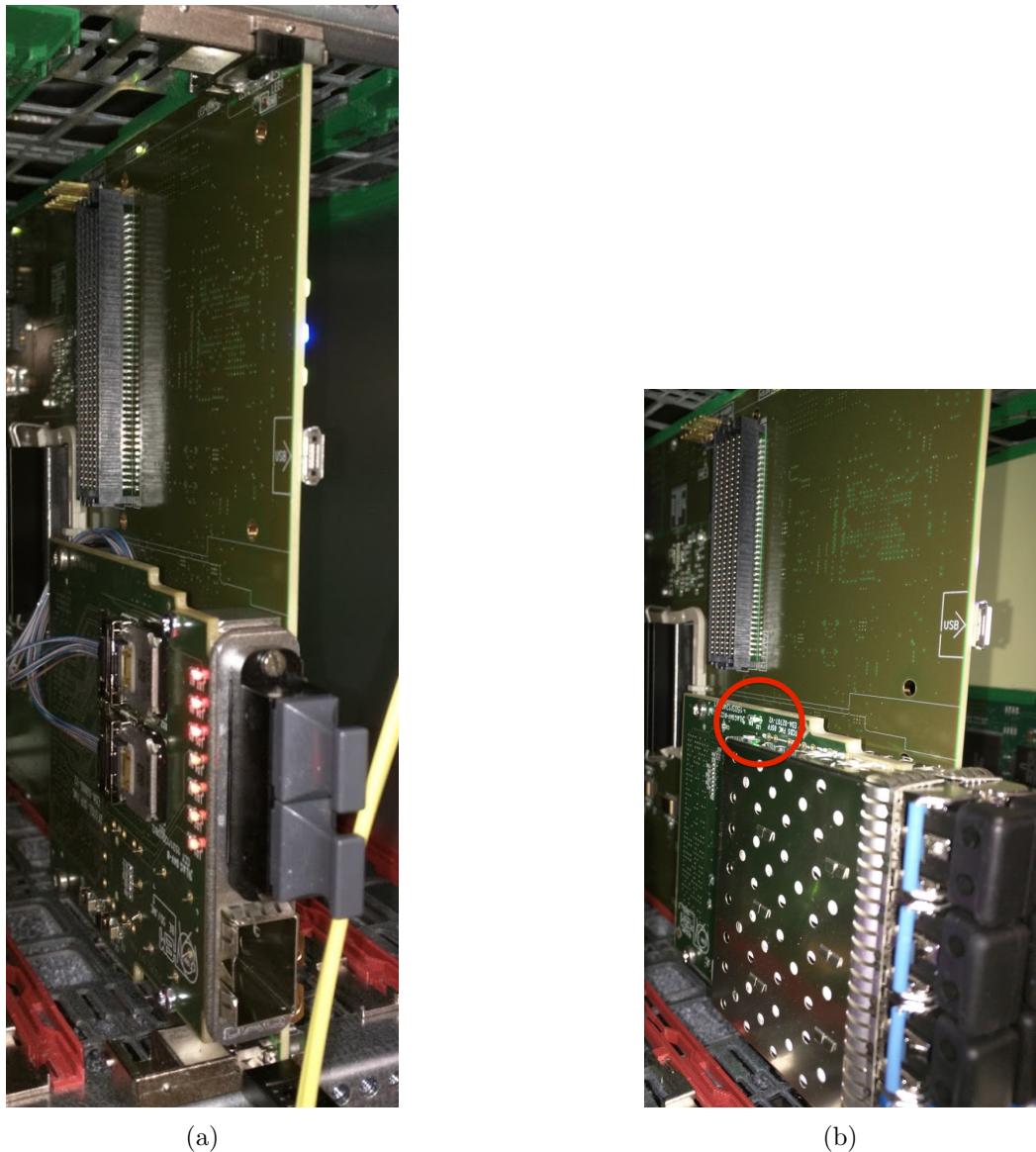


Figure 7.4

List of Figures

3.1	Number of good phases vs optical signal attenuation	12
4.1	FEC fibre mapping	13
5.1	FEC fibre mapping	15
7.1	Side-view of the FC7 board	20
7.2	Purpose of the 4 standard LEDs on the FC7 front panel	21
7.3	Front-view of the FC7 board	21
7.4	FMC cards of the Pixel DAQ system	22

List of Tables

7.1 User defined LEDs on FC7 front panel	19
--	----