Deutsches Elektronen-Synchrotron (DESY)

Universidad Autónoma del Estado de Hidalgo

Upgrade of Mimosa-DAQ of beam telescopes

Author:
Omar Mancilla Martinez

Supervisor:
Jan Dreyling-Eschweiler

September 6, 2017

Abstract

In this report a promising upgrade of the data acquisition of the Mimosa planes is presented.

The objective of this project was to explore the possibility of upgrading the beam telescope framework to a modern more efficient framework which will be able to test new devices in order to develop them for the future detectors.

Contents

1 Introduction

In this report first it's given a short introduction to the generalities of the actual hardware and software of the Beam Telescopes framework followed by the generalities of the intended new hardware, firmware and software along with the motivation to change the actual setup of the telescopes. Then there are explanations of the changes on the hardware, firmware and software along with some diagrams and pictures after this, the results of the new setup are showed and discussed along with the next modifications that should be done to the setup to make it totally implemented.

1.1 The beam telescopes

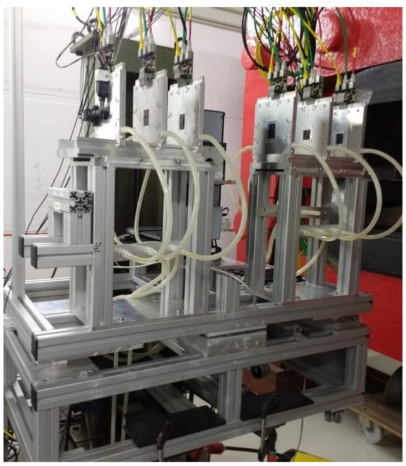

Figure 1: The DURANTA beam telescope

The beam telescopes are particle trackers, detectors with a space resolution (σ) of around 2 μm and a readout time of approximate 115 μs. The main purpose of the beam telescopes is to work as tools for detector development, adding a device to the telescope makes it a device under test (DUT), like a pixel or a strip detector. The high space resolution of the beam telescope and the reconstruction software allows the telescope to track the path of a electron coming from the test beam, when a DUT is added to the setup is possible to test it and find the space resolution with accuracy along with another characteristics like noise, readout time after and before radiation between others.

There are a total of seven beam telescopes, all of them works with six Monolithic Active Pixels Sensors (MAPS), also called Mimosa26 sensors, aligned in two arms with three sensors each and between the two arms a space to place a DUT.

1.1.1 Trigger

At both ends of the telescope there is a pair of crossed scintillators tubes each with a photomultiplier (PMT) which are used for triggering, that means that only if the four triggers (unless something else is specified by the user) measure a signal the Mimosa26 and the DUT will readout, that make it unlikely to make a measurement which don't come from the test beam.

1.1.2 The Mimosa26 sensor

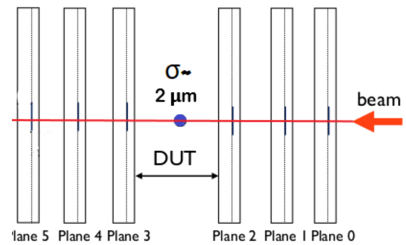

Figure 2: Diagram of the six Mimosa26 sensors, the resolution and the space for a DUT.

The component that allows the telescope to be so precise are the six Mimosa26 silicon pixel sensors, these sensors can be moved along its arm for different studies. The Mimosa sensors are composed by 576x1152 pixels, each pixel have a dimension of 18.5x18.5 μm that means that the sensor itself is 13.7x21.5 mm, the thickness of the sensor is 50 μm where only 14 μm are the active layer. The more material is in the beam path more scattering occurs and the reconstruction of the beam track becomes harder. Each sensor can read up to 10^6 hits/cm^2/s in that means a readout time of approx 115.2 μs of a frequency of 8.68 kHz, all this information is sent to an amplifier at the end of each column then to an analog to digital converter with a zero suppression circuit, that means that the output only contains the information of the pixels that detect a signal, the data stream goes to two memory banks which simultaneously read and write.

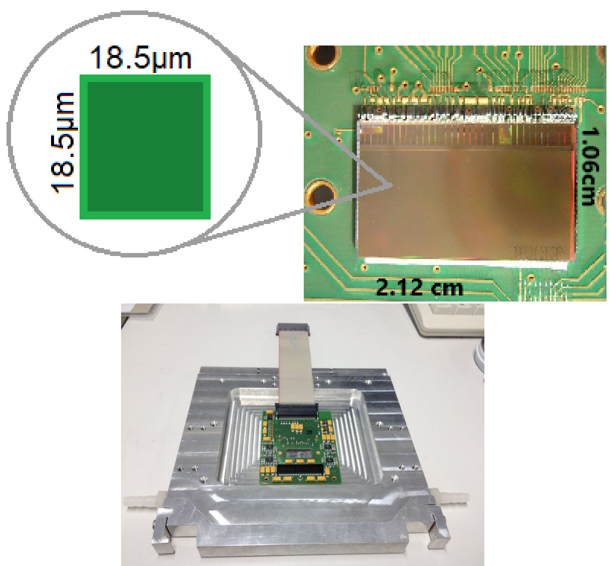

Figure 3: Pixel of the Mimosa sensor, the sensor and its housing.

1.2 Mimosa DAQ

The current method of the data acquisition of the Mimosa sensors and the alternative proposed in this project are exposed

1.2.1 Current setup

In the current setup the output of the Mimosa26 sensors follow the next scheme:

6x		Data		Nationals		PC
Mimosa	⇒	Reduction	⇒	Instruments	⇒	(Storage)
Sensors		Unit		Crate		

This setup is based around the National Instruments (NI) crate which is a commercial hardware with a price around the 15,000 €, in order to send the information to the NI crate a data reduction unit is required.

The NI crate have a FPGA module, this module requires a firmware to be uploaded to it in order to work, in this setup the firmware is based on Lab View. Also the readout software is based on Lab View and C++. All these make the development or change of the software inflexible.

2 New Mimosa DAQ

With the update of the LHC and the ILC it is necessary to be able to develop and test new and better detectors with higher spacial and time resolution, that is the mission of the beam telescope and in order to keep testing newer DUT's is required to upgrade it constantly, by upgrading to the proposed setup a wider variety of devices will be able to be tested.

The new proposed setup eliminates the need of the National Instruments hardware and the data reduction unit:

6x		MMC3		PC
Mimosa	⇒	Board	⇒	(Storage)
Sensors		(FPGA)		

This setup along with being cheaper than the actual one, is based on a modular framework which allows comfortable development and usage of different sensors as the Mimosa26 and the FEI4 pixel sensor.

2.1 Hardware: The MMC3 Board with a FPGA

The MMC3 Board is a custom board with the next components:

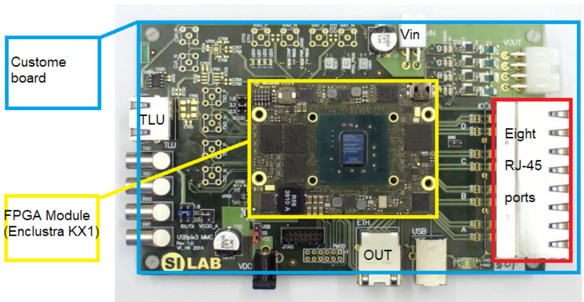

Figure 4: The MMC3 board and its components and ports

It count with a connection for the trigger unit (TLU), 4 trigger inputs to serve as a trigger unit, 8 RJ-45 ports for the sensors (Mimosa, FEI4 or other), USB port, power port and most importantly on the center of this board there is a Xilinx FPGA module. The FPGA is a device which can be programmed to do a simple logic operation or emulate a complex digital chip.

In this board we are using a FPGA module from the Xilinx company, even if the FPGA comes from a private company the connections of the custom board are common, and the price is low, a price of around 1,500 €, a 10% of the price of the NI crate currently used.

2.2 Firmware

The firmware is the software that is uploaded to the hardware, controlling it in order to make it work.

The firmware is uploaded to the FPGA module so it can receive the data from the RJ-45 ports, make a handshake with the TLU and send the resulting information through the Ethernet output port. All this is programmed in Verilog, a hardware description language which in this case describe all of the already mentioned functions, the code is compiled and uploaded to the module by VIVADO, which is the development software required to work with the Xillinx FPGA.

Figure 5: Vivado Xilinx logo

2.3 The new Software Framework

The new framework called PyBar is based on C++ and Python, which makes the software understandable without losing computing performance. The software in general lacks of GUI and works with python scripts which are called from the

Linux terminal, between those scripts we have the possibility to take data from the FEI4 sensor or from the Mimosa26.

The scripts are called from the Linux terminal but no necessarily from the computer next on the test beam area, the software runs on a PC in the local network.

3 Development and Commissioning

The project started with only the MMC3 board along with the code of software and firmware downloaded from GitHub, due to the easy access to this platform, and contact with the developers of the setup.

3.1 Hardware components

The first issue was to power the board, a 5v generic power supply was used, also a jumper on the board was changed to power the board with the supply instead of via USB, as is seen in the next figure:

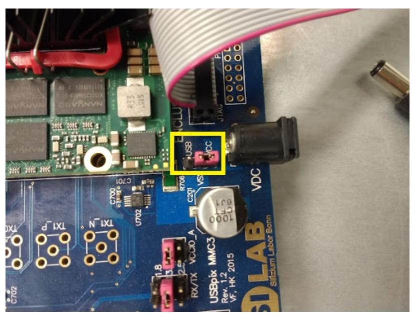

Figure 6: The jumper was changed from the USB pin to the VCC pin.

To enable the RJ-45 ports eight shorts were made on the back of the MMC3 board as seen in the next figure:

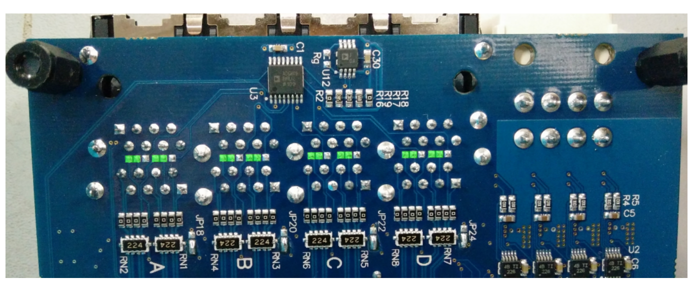

Figure 7: The yellow marked pins were soldered together to enable the RJ-45 ports.

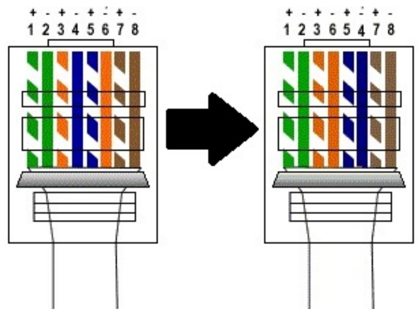

Figure 8: The 4th and 6th pin are switched on only one side.

Once enabled the RJ-45 ports it is necessary to connect the Mimosa26 sensors to the ports, for that custom ethernet cables are required, the difference between these cables and a normal ethernet cable is that the 4th and 6th wire inside only one side of the cable are switched like in the figure 8.

The custom cables are only to connect the Mimosa26 sensors to the RJ-45 ports, all the other ethernet connections will remain with the wires of the actual setup.

Additionally, in order to prepare all the hardware for testing and troubleshooting, the power cords for the pixel sensor FEI4 were created and a special dual power supply acquired. The power cord and the connections to the power supply are shown on the next figure:

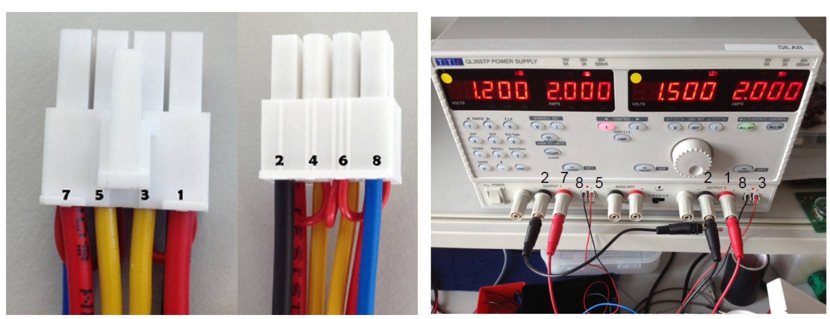

Figure 9: The created power cord of the FEI4 at both ends.

With all this we have also a ready to function FEI4 to test the hardware and firmware once this last one is uploaded.

3.2 Generating the firmware

As it have been already said the MMC3 board works with a FPGA module, this module need to be configured with a firmware in order to work. The basic resources of the Firmware were extracted from GitHub, after that the firmware was compiled with VIVADO, getting a bitfile, which is the file that is uploaded to the board, then the bitfile is uploaded to the board.

After uploading successfully the bitfile the firmware was changed so the IP address of the board matches the ones of the local network in the testbeam area.

3.3 Using the software framework

After installation, the main scripts used in this project were the "scan_m26_telescope.py" and the "scan_fei4_self_trigger.py", which are the scripts to read from the beam telescope and the FEI4 respectively. The script for the beam telescope have the option to program the Mimosa26 sensors replacing another part of the actual setup, in this project we didn't take that possibility, the script reads all the information of each Mimosa26 sensor and save it all together in a file with h5 format. The FEI4 script interpret this h5 file and generate an pdf file with the hit-maps, on the other side the telescope's script doesn't and is necessary to run an interpreter script to get the hit-maps.

The Mimosa script works with the next principle:

- Initialize and configuration: here the script configure the Mimosa26 sensor if the option is enabled, also configure the FEI4 since all the scripts from PyBar use the sensor.
- Starts scan: In this stage the script reads the information from all the sensors simultaneously, after the defined time runs out the script goes to the next stage.
- Saving: here the data is stored in a h5 file, if the scan is without an external trigger the h5 file can be quite big.

Also there are another PyBar scripts to:

- Scan the FEI4 with an external trigger
- Enable the detected trigger for all the scans
- Tune the FEI4
- and others

4 DAQ tests

With the hardware setup finished and the firmware uploaded to the MMC3 board it is possible to run a test to be sure that everything is working.

4.1 FEI4 read-out at the DESY II test beam

To test the FEI4 the Mimosa26 sensors were disabled, the FEI4 connected to the MMC3 board and mounted on the beam telescope; after that a electron beam was turned on went trough the FEI4. Then the script to scan from the FEI4 was executed there a h5 and a hit-map was generated (See figure 10).

The hit-map clearly shows the region where the beam strikes the FEI4 and there is no doubt that is running and working like it should.

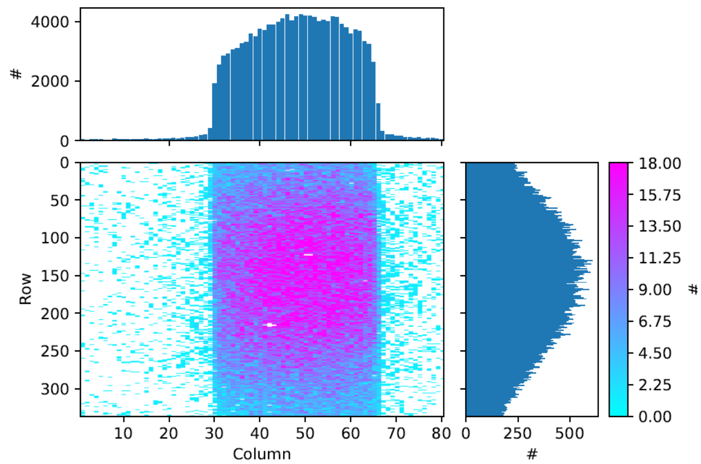

Figure 10: The FEI4 hit-map with the test beam, the sharp borders of the hit-map are due to the beam collimator.

4.2 Mimosa26 read-out with a β radioactive source

To test the Mimosa26 sensors the test beam area wasn't available so instead of a beam of electrons a β radioactive source was used, strontium 99 , since the amount of electrons of the source that reach the sensor is probably less than the amount in the test beam a measurement of the noise was made by running the telescope script without any source resulting in a h5 file. After the scan of the noise the radioactive source was placed over the sensor (as seen in the figure 11) and the script was run again giving the another h5 file. After the measurements the script for the interpretation of the h5 files was executed on both files, resulting in the hit-maps of the figure 12 and 13.

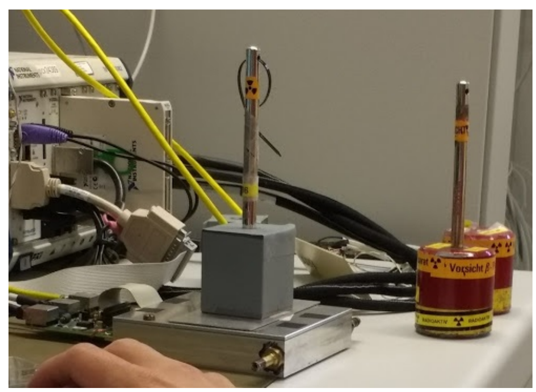

Figure 11: β Radioactive source over a Mimosa26 sensor.

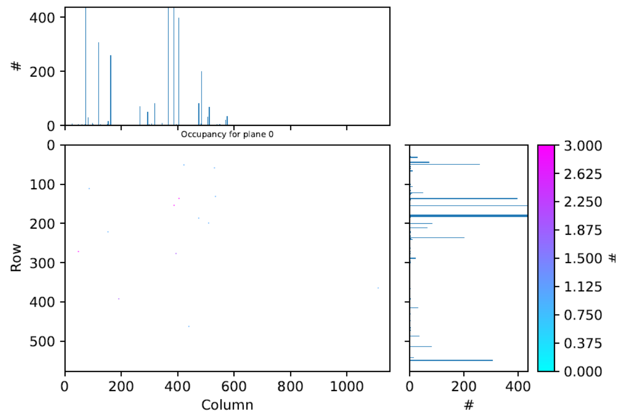

Figure 12: Hit-map of the Mimosa26 Sensor without any source of electrons.

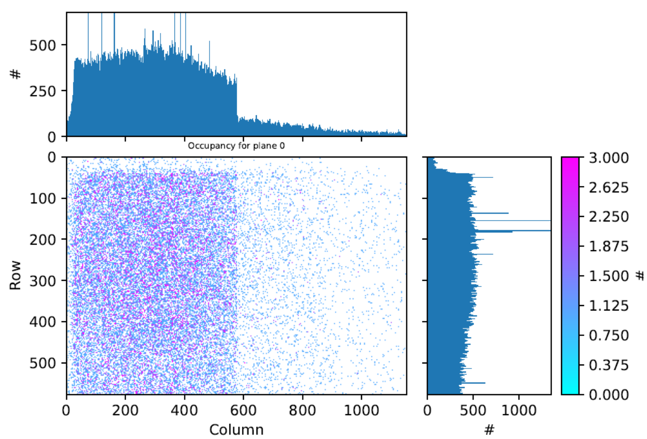

Figure 13: Hit-map of the Mimosa26 sensor with a β radiation source over it.

The difference between the hit-maps is obvious, even if there is some noise and hot pixels, the Mimosa26 sensor is running and taking data from the radioactive source.

5 Summary and outlook

In the project we accomplished many tasks, like:

- Power the MMC3 board and make the necessary changes so it can work
- Create, modify and upload the firmware to the board
- Test the FEI4 sensor and scripts
- Get a reading from the Mimosa sensors
- Graph the data of the Mimosa sensor

But some there are some improvements to make:

- Put the MMC3 in a protective housing
- Change the firmware to avoid custom cables
- Test the 6 Mimosa26 in the test beam with the PyBar scripts
- Make this new setup available for the users of the test beam

Acknowledgements

First of all I want to thank my supervisor Jan Dreyling who guide me through a amazing hardware project and inspired me with it to decided that I want to work in hardware physics for more than just a summer. Also I want to thank the whole DESY ATLAS team for being the perfect working group with the weekly meetings, the beautiful working environment and the social events making me feel part of ATLAS and part of DESY. Finally but not least I want to thank the organizers of the summer school program for giving me the chance to live an experience like this, so beautiful, funny and instructive.
Thank you everyone for one of bests summers in my life!