

Diploma thesis

# Online Trimming of the CMS Pixel-Detector and Testing of the -X-Side of the Pixel-Detector

Manuel Waser  
Erikastrasse 2  
8003 Zürich  
mwaser@student.ethz.ch

Supervised by U. Langenegger and A. Starodumov

June 2008

# Contents

<b>1</b>	<b>Abstract</b>	<b>4</b>
<b>2</b>	<b>Introduction</b>	<b>5</b>
2.1	Standard Model . . . . .	5
2.1.1	Higgs mechanism . . . . .	7
2.2	LHC . . . . .	8
2.3	CMS . . . . .	10
2.4	Pixel Detector . . . . .	10
2.4.1	Signal . . . . .	13
2.5	Motivation . . . . .	14
<b>3</b>	<b>Trimming procedure</b>	<b>15</b>
3.1	Initial facts . . . . .	15
3.2	Trim parameters . . . . .	15
3.2.1	$V_{\text{ThrComp}}$ . . . . .	15
3.2.2	$V_{\text{Trim}}$ . . . . .	15
3.2.3	Trimbits . . . . .	17
3.3	Trimming algorithm . . . . .	17
<b>4</b>	<b>Measurements of temperature dependence</b>	<b>18</b>
4.1	Important parameters . . . . .	18
4.1.1	$V_{\text{Ana}}$ . . . . .	18
4.1.2	$V_{\text{OffsetOp}}$ . . . . .	18
4.1.3	$I_{\text{BiasDAC}}$ . . . . .	18
4.1.4	CalDel . . . . .	19
4.2	Temperature dependence . . . . .	19
4.3	Parametrized trimming . . . . .	19
4.4	Conclusion . . . . .	24
<b>5</b>	<b>Temperature sensor calibration</b>	<b>27</b>
5.1	Technical specifications of the temperature sensor . . . . .	27
5.2	Calibration . . . . .	27
5.3	Measurements . . . . .	29
5.3.1	Linearity . . . . .	29
5.3.2	Verification . . . . .	29
<b>6</b>	<b>Conclusion</b>	<b>33</b>
<b>7</b>	<b>Acknowledgment</b>	<b>34</b>

*Contents*

---

<b>A Perl script for the calculations</b>	<b>35</b>
<b>B Pixel testing manual</b>	<b>39</b>
<b>C References</b>	<b>63</b>

# 1 Abstract

The main goal of this diploma thesis is the online trimming of the CMS pixel detector. At the LHC the detector will be controlled with a framework called X-DAQ, which is operated by a web-interface.

Most of the parameters for controlling the detector are temperature dependent. To determine these parameters by measurements would take a long time. Therefore it is tested, during this diploma thesis, if it is possible to calculate the values of these parameters. All modules have been tested, before mounting on a support structure. During this module testing the parameters were measured only for two temperatures ( $-10^{\circ}\text{C}$  and  $+17^{\circ}\text{C}$ ). These values are the initial position for the calculations. To understand the behaviour of these parameters on different temperatures several measurements have been done in a cooling box, where the temperature can be controlled well. Also the built-in temperature sensors have been tested in accuracy and linearity. Furthermore the results had to be implemented in the X-DAQ software.

A second part of the thesis is the testing of one half of the detector. The objective of this part was to find damaged modules, so they could be replaced before the detector will be inserted in the CMS. During this testing the software has been improved as well. The outcome of this part is the "Pixel testing manual" which is added in the appendix.

In the introduction a very short outline in the standard model and a summary of the CMS-experiment is given. It includes the most important facts and specifications of LHC and CMS. Also a brief overview of the given settings and of the important parameters for the trimming is given.

## 2 Introduction

### 2.1 Standard Model

The Standard Model of particle physics is a theory, which explains three of the four known fundamental interactions. It includes the electro-magnetic force, the weak interaction and the strong interaction. The gravitation is not included. The Standard Model is a relativistic quantum field theory. All matter is composed of 12 point-like particles, called fermions. For each particle an anti-particle exists as well. The fermions are divided in two groups: the leptons and the quarks. It is also possible to group the particles in generations depending on their mass. Therefore two leptons and two quarks are grouped. In table 2.1 the particles are listed in order of their masses, with the lightest generation on top. The charge of the electron,  $\mu$  and  $\tau$  is the elementary charge  $-e = -1,602176487 \times 10^{-19}$  C. The neutrinos have no electric charge and very low masses. Until 1999 it was assumed they have no mass at all. At this time the neutrino oscillation was discovered. Which means it is possible for neutrinos to switch between the generations. This only is possible if the neutrinos have masses.

The quarks are classified as the particles which carry a color charge and therefore interact via the strong force. The quarks are bound together as they have to form color-neutral particles called hadrons. This is achieved by either binding three quarks (baryons) or a quark and an anti-quark (mesons). An example for the first type is the proton. It consists of two up-quarks and one down-quark ( $u u d$ ). The electric charge of the upper quarks is  $+\frac{2}{3}e$ , of the lower ones  $-\frac{1}{3}e$ . Therefore a proton has an electric charge of  $+1e$ . The  $\pi^+$  consists of an up- and anti-down-quark ( $u \bar{d}$ ), which is a meson.

The energy of a particle in LHC can be transformed into other particles corresponding to Einstein's formula  $E = mc^2$ . If the energy is high enough it is therefore possible to create particles with higher mass than the original particles. The production of new particle has to obey different rules as for example energy and momentum conservation. The only stable particles are the ones from the first generation. Particles of higher generations will decay into particles of lower generations. Therefore in the universe mainly particles of the first generation exist.

The interactions are explained by the exchange of particles called bosons. They are listed in table 2.2 according to the corresponding force. The graviton is a highly speculative particle. No quantum gravity has been established yet. In figure 2.1 the interactions between the different particles are shown.

Besides the particles mentioned before in the Standard Model also the Higgs boson is introduced. It is a massive elementary particle. In the Standard Model it is responsible for the masses of the other point-like particles. Because of its weight it has not been detected yet. There is hope to find the Higgs boson at the Large Hadron Collider at CERN.

Even if the Standard Model explains a lot of the observed phenomena there are theories which extend the Standard Model. The most discussed candidates are supersymmetric theories

Table 2.1: List of fermions in the standard model.

Leptons	Quarks
$(\nu_e)$ $(e)$	$(\text{up})$ $(\text{down})$
$(\nu_\mu)$ $(\mu)$	$(\text{charmed})$ $(\text{strange})$
$(\nu_\tau)$ $(\tau)$	$(\text{top})$ $(\text{bottom})$

Table 2.2: Interaction particles of the Standard Model

Force	Interaction boson	Relative strength
electro-magnetic force	photons	$10^{-2}$
weak interaction	$W^\pm, Z$	$10^{-5}$
strong interaction	8 gluons	1
gravity	graviton	$10^{-38}$

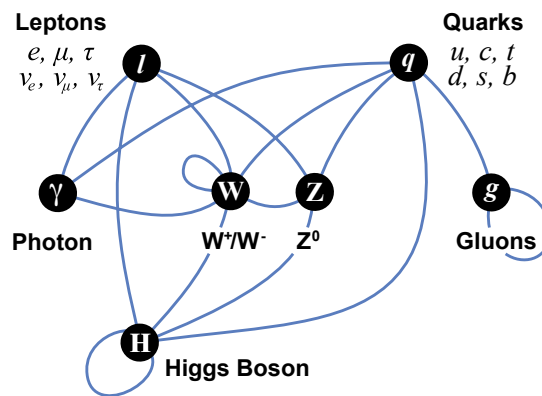


Figure 2.1: Interactions between the particles of the Standard Model.

or the string theory. At the moment it is not known which will be the right one. Possibly the LHC experiments show some signs for one of these theories.

### 2.1.1 Higgs mechanism

A problem of the Standard Model of particle physics is the appearance of the mass of the  $W^\pm$ - and the Z-bosons. The Higgs mechanism is a mathematical instrument to introduce the masses of these particles.

In physics the behaviour can be described by Lagrangians. This principle was introduced in mechanics and is well confirmed. Nowadays this concept is used in mostly all parts of physics. The symmetries of the Lagrangian define the conservation laws, for example the invariance under translation guides to the conservation of momentum.

The Higgs mechanism will be shown by breaking the  $U(1)$  gauge symmetry, which is the simplest example. We start with the Lagrangian

$$L = (\partial_\mu \Phi)^* (\partial^\mu \Phi) - \mu^2 \Phi^* \Phi - \lambda (\Phi^* \Phi)^2.$$

It is now needed to make the Lagrangian invariant under  $U(1)$  local gauge transformations,

$$\Phi \rightarrow e^{i\alpha(x)} \Phi.$$

This can be achieved by substitute the derivative  $\partial_\mu$  by the covariant derivative,

$$D_\mu = \partial_\mu - ieA_\mu.$$

The resulting Lagrangian has the form of

$$L = (\partial^\mu + ieA^\mu) \Phi^* (\partial_\mu - ieA_\mu) \Phi - \mu^2 \Phi^* \Phi - \lambda (\Phi^* \Phi)^2 - \frac{1}{4} F_{\mu\nu} F^{\mu\nu}.$$

For  $\mu^2 < 0$ , this is the QED Lagrangian for a charged scalar particle of mass  $\mu$ . Which is not what we want. We want to generate the masses by spontaneous symmetry breaking. Therefore we take  $\mu^2 > 0$ . Now  $\Phi$  is replaced by

$$\Phi(x) = \sqrt{\frac{1}{2}} [v + \eta(x) + i\xi(x)].$$

The Lagrangian then becomes

$$L' = \frac{1}{2} (\partial_\mu \xi)^2 + (\partial_\mu \eta)^2 - v^2 \lambda \eta^2 + \frac{1}{2} e^2 v^2 A_\mu A^\mu - ev A_\mu A^\mu - \frac{1}{4} F_{\mu\nu} F^{\mu\nu} + \text{interaction terms}.$$

The resulting particle spectrum contains a massless Goldstone boson  $\xi$ , a massive scalar  $\eta$ , and a massive vector  $A_\mu$  which we were looking for. We now have

$$m_\xi = 0, \quad m_\eta = \sqrt{2\lambda v^2}, \quad m_A = ev.$$

The problem now is to modify the Lagrangian such that the massless Goldstone boson does not appear anymore. Therefore we can change the gauge. Instead of the gauge taken before, we now take

$$\Phi \rightarrow \sqrt{\frac{1}{2}} (v + h(x)) e^{i\theta(x)/v}$$

$$A_\mu \rightarrow A_\mu + \frac{1}{ev} \partial_\mu \theta.$$

$\theta(x)$  has to be chosen, such that  $h$  is real. The Lagrangian then becomes

$$L'' = \frac{1}{2}(\partial_\mu h)^2 - \lambda v^2 h^2 + \frac{1}{2}e^2 v^2 A_\mu^2 - \lambda v h^3 - \frac{1}{4}\lambda h^4 - \frac{1}{2}e^2 A_\mu^2 h^2 + ve^2 A_\mu^2 - \frac{1}{4}F_{\mu\nu}F^{\mu\nu}.$$

The Goldstone boson does not appear in the theory anymore. The appearance of the extra degree of freedom is corresponding only to the freedom to make a gauge transformation. The Lagrangian describes now only two interacting massive particles, a vector boson  $A_\mu$  and a massive scalar particle  $h$ , which is called Higgs particle. The Goldstone boson has been turned into the longitudinal polarisation of the massive gauge particle. This modification is called the "Higgs mechanism"[1].

In the Standard Model this has to be done for the  $SU(2)$  gauge symmetry, which represents the weak interactions. The result of these modifications are three massive gauge fields ( $W^\pm$  and  $Z$ ) and one massive scalar  $h$ , the Higgs boson.

## 2.2 LHC

The large hadron collider (LHC) is a new particle accelerator which is located at CERN. It will be the most powerful particle accelerator built so far. It will accelerate protons to an energy of 7 TeV. This leads to a center of mass energy of  $\sqrt{s} = 14$  TeV. It will be also possible to operate the LHC with heavy ions. If there are no big problems, it will start working in summer this year. It is located in a tunnel with circumference of 27 km in the underground between the Jura Mountains and the lake Geneva. This tunnel already was used from 1989 to 2000 for the LEP accelerator which operated with electrons and positrons. The protons are held on their orbit by 1232 superconducting dipole magnets. They are cooled down to  $-271^\circ\text{C}$ . Also there are 392 quadrupole magnets to focus the beam. Just before the collision points some special magnets are installed to squeeze the beam to increase the probability of collisions. Besides the high energy also the instantaneous luminosity of  $10^{34} \text{ cm}^{-2}\text{s}^{-1}$  is a characteristics of the LHC. At this luminosity the ring is filled with nearly 3000 bunches of protons. The spacing between the bunches is 25 ns on average. A schematic of the LHC and its interaction points is shown in figure 2.2.

There are four experiments located at the LHC:

- CMS
- ATLAS
- LHCb
- ALICE

CMS and ATLAS are the main experiments. They are designed to detect new particles. The hope is to find the Higgs boson, the only undetected particle in the Standard Model, supersymmetric particles or other evidences of new physics.

LHCb is designed for B-physics measurements. Also it is expected to find indirect evidence for new physics in flavor-changing neutral currents.



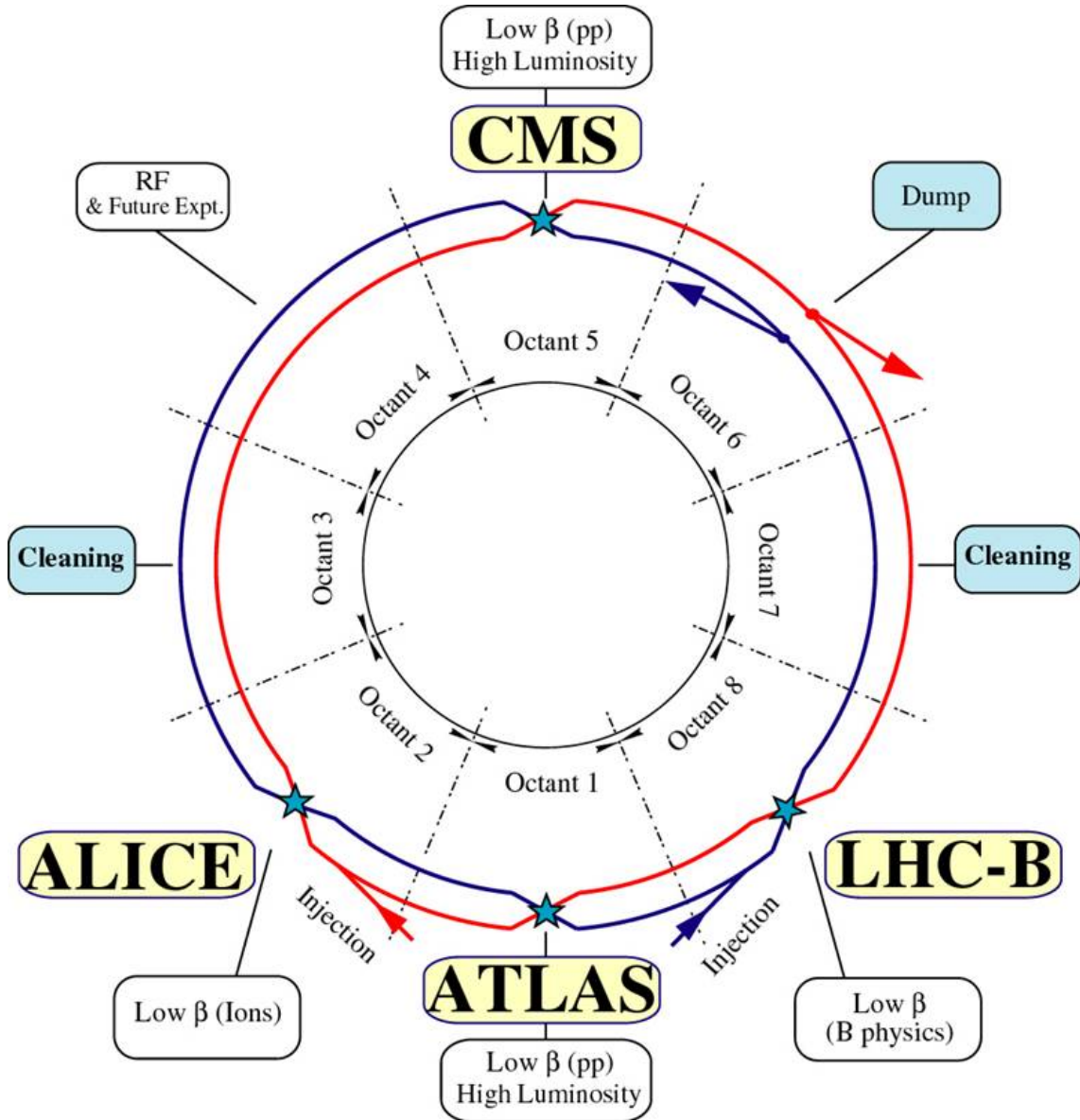


Figure 2.2: Schematic of the LHC with interaction points. [2]

ALICE is an experiment specialized in measurements with heavy nuclei. It is supposed to find evidence for a new phase of matter: the quark-gluon plasma.

## 2.3 CMS

The CMS (compact **m**uon **s**olenoid) detector is one of the two main detectors. It is relatively compact in volume, 8 times smaller than ATLAS, and has a very strong solenoid. Even if its name includes the word *compact* its dimensions are impressive. It is 21 m long, 15 m in diameter and weighs about 12500 tonnes. The production of the different parts was done at different countries. It was assembled at the surface and then piece by piece lowered to the cavern, which is located 100 m below the surface. The main goal of this experiment is to understand the TeV-physics, especially the detection of the Higgs boson. It also should detect supersymmetric particles, if they exist and can be produced at LHC.

The detector consists of four different parts, listed from the center:

- tracker
  - pixel detector**
  - silicon strip detector
- calorimeters
  - electromagnetic calorimeter
  - hadronic calorimeter
- solenoid
- muon detector

The tracker, the electromagnetic and the hadronic calorimeter are positioned inside the superconducting solenoid, which generates a magnetic field of 3.8 Tesla. The large muon detector does not fit inside the magnet.

The Higgs boson has different decay channels depending on its mass. They are shown in figure 2.3. The main decay channel for a Higgs boson with a mass smaller than  $2 W^\pm$  is to  $b\bar{b}$ . This channel can not be used, because of the high background. Good channels are  $\text{Higgs} \rightarrow \gamma\gamma$ ,  $\text{Higgs} \rightarrow \tau^+\tau^-$ ,  $\text{Higgs} \rightarrow WW$  and  $\text{Higgs} \rightarrow ZZ \rightarrow 4\text{ leptons}$ . The decay Higgs to  $ZZ$  has the best opportunity to be detected because the Z bosons will decay in four leptons. These leptons can be triggered well.

Because of the high luminosity, a normal siliconstrip tracker does not work at the innermost sections. So the first three layers are a pixel detector. The outer ones are conventional silicon strip trackers. On figure 2.4 the front of the CMS detector is visible.

## 2.4 Pixel Detector

The pixel detector includes the three innermost layers of the tracking system. The barrel has a length of 53 cm and radii of 4.4 cm, 7.3 cm, and 10.2 cm. It is closed by two end-caps on each side. For the read-out the detector is divided in two half shells. Each half shell is divided in

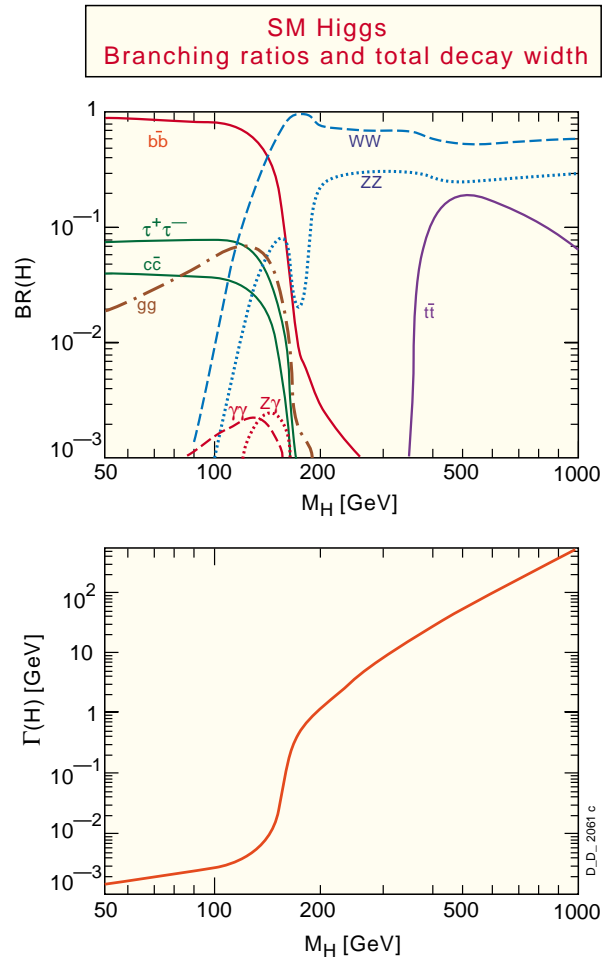


Figure 2.3: Branching ratio and width of the Higgs boson depending on its mass. [3]

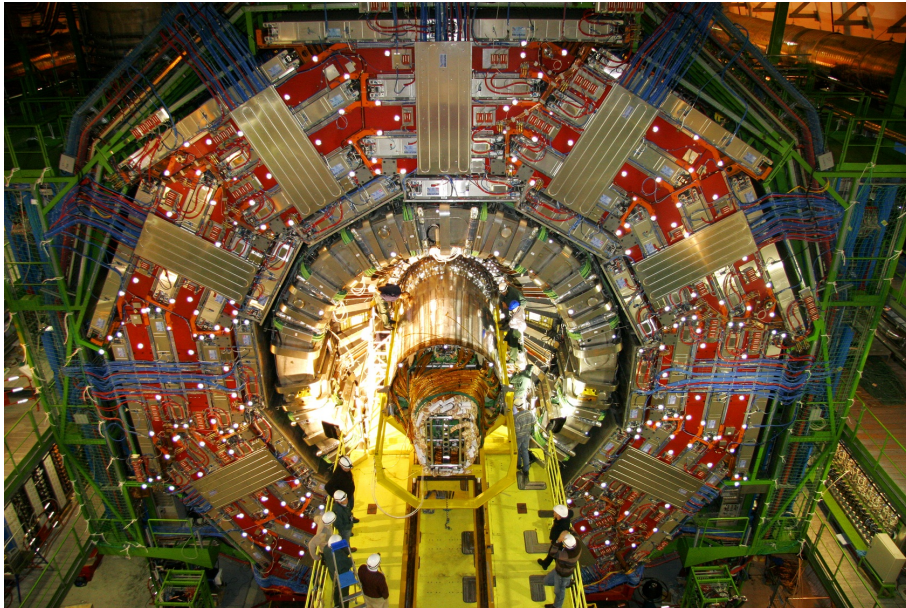


Figure 2.4: Front view of the CMS detector. [4]

two halves, which are segmented in eight sectors. One sector contains 24 modules. In total the detector consists of 768 modules. On figure 2.5 one half of the pixel detector is visible. The black spots in the middle of the picture are the protections of the token-bit-managers (TBM). Only the TBMs of the modules in the middle are visible. The others are covered by the kapton-cables of the modules. These cables are for the programming and the read-out of the modules.

The power supply and communication with the modules is done by two supply tubes per half shell. In these supply tubes the AOHs (analog optical hybrid), ccus and other supplies are located. The AOHs are used for the read-out of the signal. They convert the electronic signal into an optical signal and induct it in an optical fibre. The ccus are used for programming the ROCs. They receive the signal from the FEC (front-end controller) and send it to the TBMs.

One full barrel module contains 16 chips, also called ROCs (readout chips), each containing  $52 \times 80 = 4160$  pixels. The dimensions of the pixels are  $150 \mu\text{m}$  in  $z$ -direction and  $100 \mu\text{m}$  in  $(r, \phi)$ -direction. In total the pixel detector contains more than 65'000'000 pixels. There exist half modules with eight ROCs only as well. They are placed at the border of the half-shells.

It is possible to adjust several parameters of the modules. This is needed because the settings depend on the temperature and the target threshold. The programming is done by the TBM. It sends the commands to the ROCs. The TBM itself can be controlled by the computer via a FEC and a ccu on the supply tube.

For the read-out, the signals of the ROCs are combined and coded by a TBM and then sent to an AOH. The AOH converts the electrical signal into an optical signal. Then the signal is inducted into optical fibers and sent to the FED, which decodes the data and sends them to an event builder for further analyses. For the two innermost layers two channels per full module

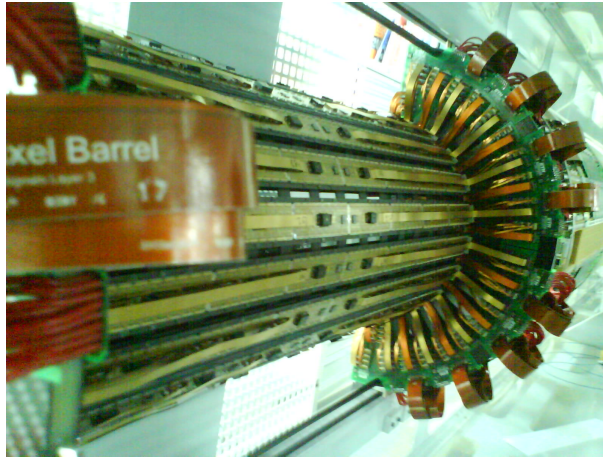


Figure 2.5: One half of the pixel detector.

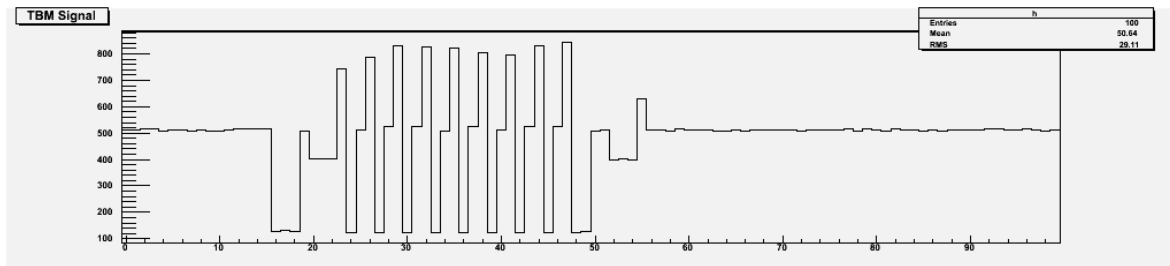


Figure 2.6: Typical signal of a half-module or of one channel of a module located on layer 1 or 2 with no hits.

are provided, because of the high data flow. For the third layer one channel per module should be enough.

### 2.4.1 Signal

The data collected in a module is coded into an electronic signal. In figure 2.6 a typical signal of a half of a module with no hits is shown. As described before the modules on layer 1 and 2 have one channel per half-module. The black-level is adjusted to  $\sim 512$ , the ultra-black (UB) to  $\sim 150$ .

The signal contains a header, information about the hits and a trailer. The header contains the first three UBs, which defines the beginning of the signal. Also an event counter is added to the header (the following three clocks at the same level). Then follow the 8 ROCs, separated by one UB. They contain information about the hits on pixels. This information consists of the position and the strength of the hit. Also it is possible to set a last-DAC. With this it is possible to read out an additional register, as for example the temperature. The parameter to be read out has to be chosen before the event is read out. After the ROCs follows the trailer, which consists of two UB levels. It signals the end of the event.

## 2.5 Motivation

For a working pixel detector some parameters have to be in a certain range, others have a predefined value. Some of these parameters are temperature dependent, mainly the ones which are important for the trimming (see chapter 3) of the detector. During the tests of the modules several measurements for the determination of the parameters were made at two different temperatures. At this time it is not known at which temperature the detector will be operating. Therefore it is necessary to be able to set the values of the parameters dependent on temperature. To collect these values by measurements would take too much time. So it would be useful to be able to calculate these values. The main part of this diploma thesis is therefore to study the computability of the parameters in dependence of the temperature and the determination of the temperature.

## 3 Trimming procedure

### 3.1 Initial facts

The trimming is important to suppress the noise and to assure a low threshold at the same time. The noise reduction is needed to avoid a buffer overflow, because then hits can be lost. With the parameters listed below it is possible to equalize the threshold of the single pixels of one ROC. The threshold corresponds to the amount of electrons, which have to be produced by the particle traversing the detector, so a signal is produced in the ROC. The threshold is identified by sending an electronic pulse to the pixels. The strength of this pulse can be varied and is known. This is done multiple times for several pulse strengths. Then the efficiency of recognized hits is evaluated and plotted versus the signal strength. Through these points a function is fitted. The threshold is defined as the point where this function crosses 50% efficiency (see figure 3.1). On figure 3.2(a) and 3.2(b) the difference of the threshold distributions between an untrimmed and trimmed ROC is shown.

For the trimming three parameters are important:

- $V_{\text{ThrComp}}$
- $V_{\text{Trim}}$
- Trimbits

In the following sections these parameters are discussed.

### 3.2 Trim parameters

#### 3.2.1 $V_{\text{ThrComp}}$

With  $V_{\text{ThrComp}}$  it is possible to move the threshold of all pixels of one ROC. With the other two parameters it is only possible to lower the threshold of the pixels. Therefore the value of this parameter is set at a value that the pixel with the lowest threshold has its threshold as close as possible to the target threshold. The temperature dependance is shown in figure 4.2.

#### 3.2.2 $V_{\text{Trim}}$

$V_{\text{Trim}}$  also is a parameter which can only be set per ROC and not per pixel. This parameter defines the maximum value the threshold of a single pixel can be shifted downwards. This parameter has to be chosen carefully, because it is divided by the trimbits in only 16 steps. The problem if this parameter is chosen too big is, that the distribution of the thresholds of the pixels gets too coarse. Therefore it is needed to balance between how far the pixel with the highest threshold can be shifted down and how smooth the other pixels can be adjusted. The temperature dependance of  $V_{\text{Trim}}$  also is shown in figure 4.2.

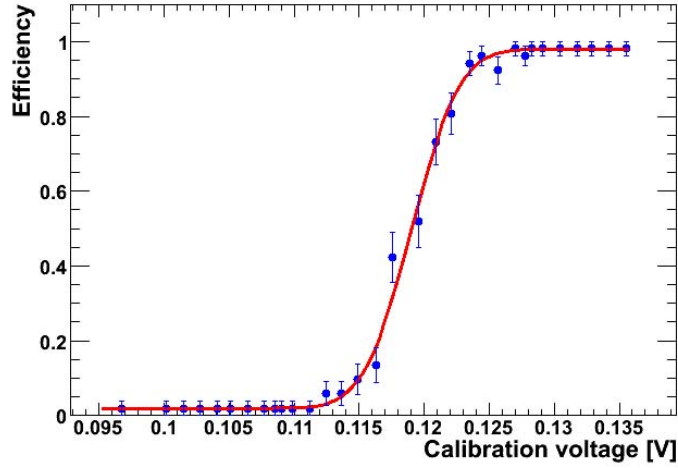


Figure 3.1: Example of a measured SCurve in units of calibration voltage vs. efficiency. Threshold  $\approx 0.12$  V. [5]

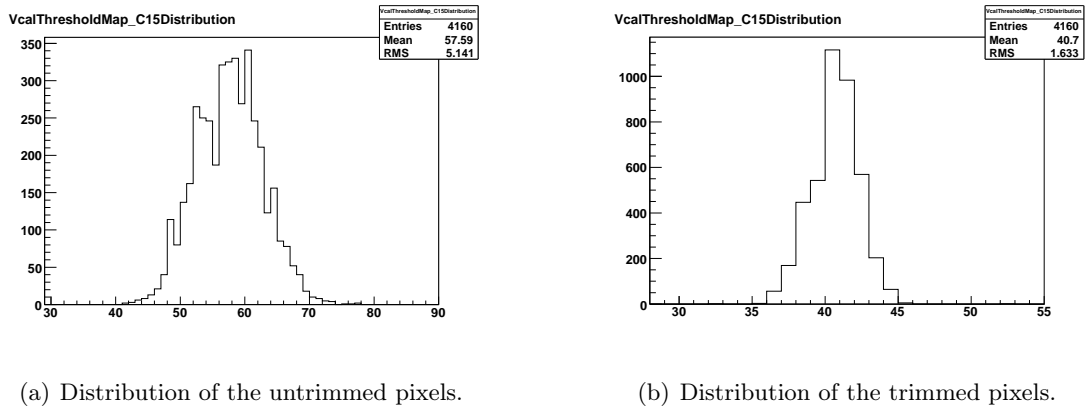


Figure 3.2: Trimmed and untrimmed distribution of the threshold of one ROC. [7]



### 3.2.3 Trimbits

The trimbits are the only parameter which can be set per pixel. It is possible to shift the pixels individually to the target threshold. This is done by dividing the  $V_{\text{Trim}}$  in 16 parts. The trimbits define the fraction of the  $V_{\text{Trim}}$  one pixel is shifted (0 means full  $V_{\text{Trim}}$ , 15 means no  $V_{\text{Trim}}$ ). Therefore the value of the trimbit is set by a binary search step by step until the threshold is as close as possible to the target threshold. Here it is obvious what happens if the  $V_{\text{Trim}}$  is chosen too big. It is not possible to bring the threshold in a desired region, because the step size is too big. The trimbits seem to be independent in temperature changes.

## 3.3 Trimming algorithm

The algorithm pursues the following scheme [6]:

1. **Determine  $V_{\text{ThrComp}}$**

For the determination of the global  $V_{\text{ThrComp}}$ , the  $V_{\text{ThrComp}}$ -threshold for all pixels is measured. Since it is only possible to lower the threshold of the pixels with the other parameters, the minimum value of this distribution determines the global  $V_{\text{ThrComp}}$  value.

2. **Determine  $V_{\text{Trim}}$**

To find the proper value for  $V_{\text{Trim}}$  the  $V_{\text{Cal}}$  thresholds of all pixels are measured. The pixel with the highest threshold is then used to determine the  $V_{\text{Trim}}$  value. For this pixel the trimbit is set to zero and  $V_{\text{Trim}}$  is increased until the threshold of this pixel reaches the target threshold. To exclude outliers the mean value ( $\langle T \rangle$ ) and the RMS of the distribution of the thresholds is determined. Only pixels with a threshold lower than  $\langle T \rangle + 5 \times \text{RMS}$  are included.

3. **Determine trimbits**

The setting of the trimbits is done by a binary search. It gives the threshold as close as possible to the target threshold. The search starts at a value of 7 and comprises four iterations.

4. **Verification**

At the end all thresholds are measured again to validate the procedure.

## 4 Measurements of temperature dependence

### 4.1 Important parameters

In addition to the parameters mentioned in the chapter before some other parameters are important for a working detector. The ones which also are temperature dependent and were studied are listed below:

- $V_{\text{Ana}}$
- $V_{\text{OffsetOp}}$
- $I_{\text{BiasDAC}}$
- CalDel

In the following paragraphs the functions of these parameters are described.

#### 4.1.1 $V_{\text{Ana}}$

This parameter controls the analog current in the ROCs. It is important for the analog current to be in a certain region. If it is too high the power consumption is too heavy. If the current is too low the ROC will not work at all.

At the beginning this parameter was thought to be temperature independent. But during the testing of the detector  $V_{\text{Ana}}$  turned out to be a function of temperature. The temperature in the testing room was remarkably higher than 17°C. It occurred that no signal was observed. After studying the problem it turned out to be a problem with  $V_{\text{Ana}}$ . So temperature measurements were done. It turned out to be dependent on temperature. The value for this parameter was calculated for a temperature of 35°C. After these modifications the problem disappeared.

#### 4.1.2 $V_{\text{OffsetOp}}$

$V_{\text{OffsetOp}}$  is a parameter which has an influence on the pulse height of the signal. Therefore it has an influence on the linearity of the signal. While the absolute value of the pulse height is already adjusted by another DAC,  $V_{\text{OffsetOp}}$  is used to shift the pulse height curve in the target ADC range [6].

#### 4.1.3 $I_{\text{BiasDAC}}$

With this parameter it is possible to adjust the address levels. It is responsible for the width of the gaps between the address levels. Also it is needed for the adjustment of the ultra-black of the ROCs (see "Pixel testing manual").

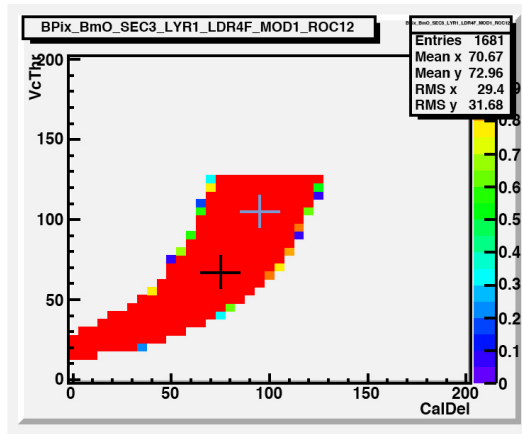


Figure 4.1: Example of a CalDel vs.  $V_{\text{ThrComp}}$  scan with its working region (red). CalDel is on the x-axis in DAC units.  $V_{\text{ThrComp}}$  is located on the y-axis also in DAC units.

#### 4.1.4 CalDel

This parameter is used for the delay control of the signal. This means it defines the timing of the signal. During the configuration of the detector a 2-dimensional scan versus  $V_{\text{ThrComp}}$  is done. The result is a working region (see figure 4.1). If this parameter is shifted too much, the signal can not be decoded anymore.

## 4.2 Temperature dependence

For the calculation it was assumed the parameters behave linearly in temperature. To make sure this assumption is not completely wrong measurements at six different temperatures have been done. The measurements were taken at  $-10^{\circ}\text{C}$ ,  $-5^{\circ}\text{C}$ ,  $0^{\circ}\text{C}$ ,  $+5^{\circ}\text{C}$ ,  $+10^{\circ}\text{C}$  and  $+17^{\circ}\text{C}$ . As one can see in figure 4.2 and 4.3 the behavior of the six parameters discussed in the paragraphs before is close to linear. For the three parameters  $V_{\text{Ana}}$ ,  $V_{\text{Trim}}$  and  $V_{\text{ThrComp}}$  the calculated values at  $0^{\circ}\text{C}$  were compared with the measured values. The results are shown in figures 4.4, 4.5 and 4.6. On the x-axis the number corresponds to the ROC number, the y-axis is the value of the parameter in ADC units. On the upper graphs the measured values for  $17^{\circ}\text{C}$ ,  $0^{\circ}\text{C}$  and  $-10^{\circ}\text{C}$  is shown. The lower ones contain the calculated and measured values for a temperature of  $0^{\circ}\text{C}$ . Most of the calculated values are within a few DACs compared with the measured ones.

## 4.3 Parametrized trimming

To test the results of the investigations above, four modules were put in the cooling box and the threshold of the pixels was measured. In the cooling box it is possible to work with the modules at a controlled temperature. For the tests the temperature was set at  $0^{\circ}\text{C}$ . The parameters were calculated based on the values measured at  $-10^{\circ}\text{C}$  and  $+17^{\circ}\text{C}$  during a full module test. These values also will be available at the real detector from the module testing. With these values it is possible to determine the slope and offset for each ROC individually. At

## 4 Measurements of temperature dependence

---

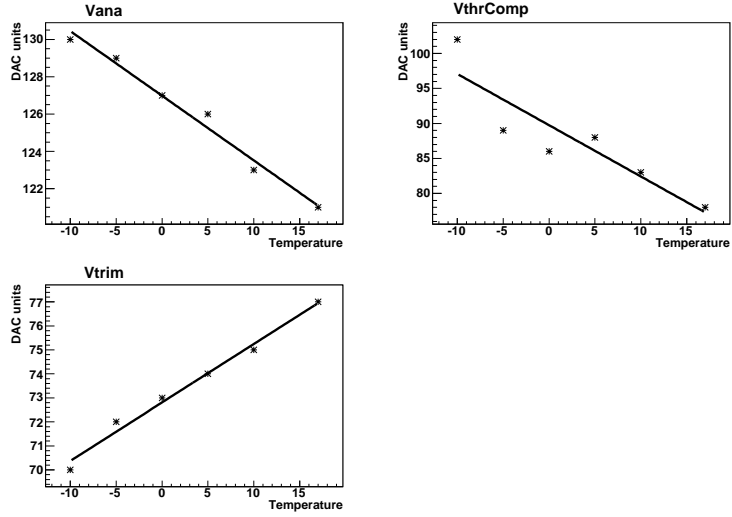


Figure 4.2: Temperature dependence of  $V_{Ana}$ ,  $V_{Trim}$  and  $V_{ThrComp}$ .

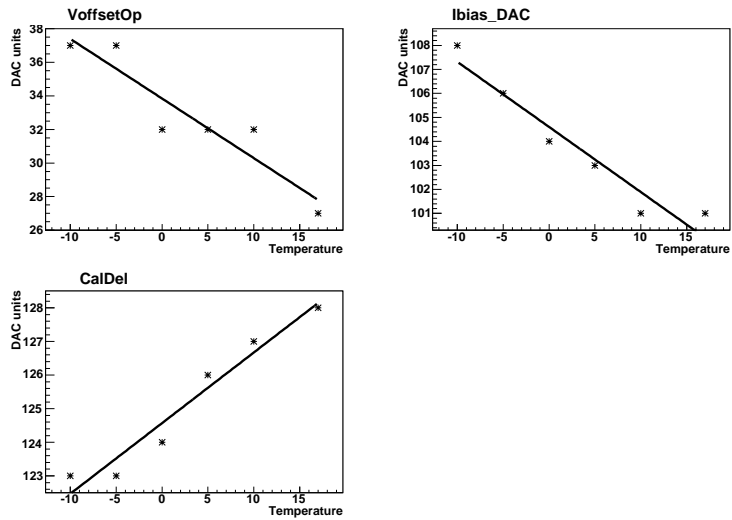


Figure 4.3: Temperature dependence of  $V_{OffsetOp}$ ,  $I_{BiasDAC}$  and  $CalDel$ .  $V_{OffsetOp}$  is measured with a step size of 5 units.

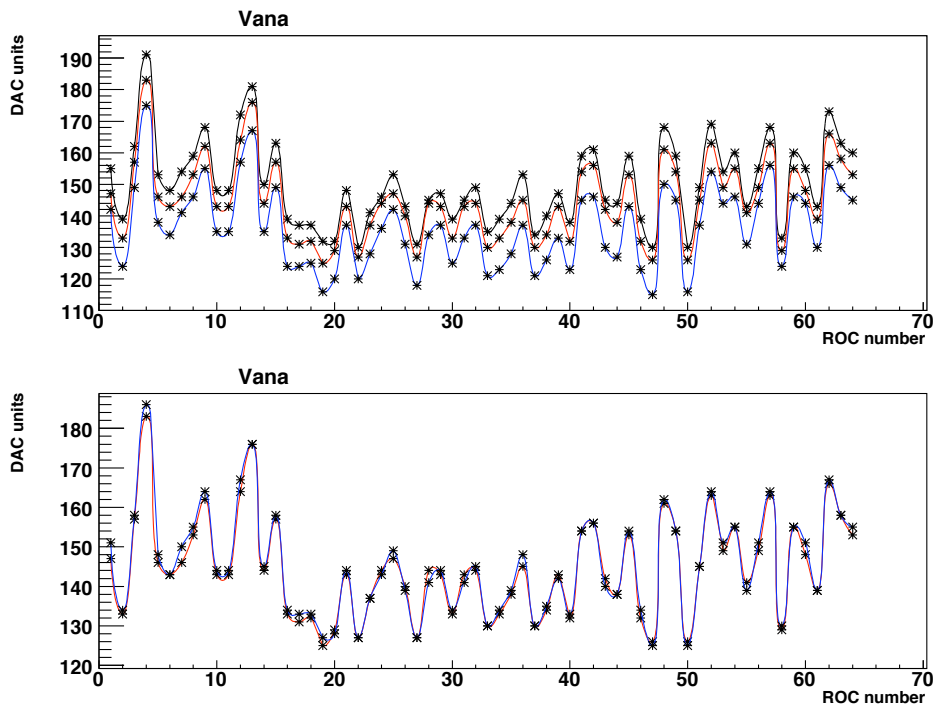


Figure 4.4: Difference between calculated and measured value for  $V_{\text{Ana}}$ , at  $0^\circ\text{C}$  for several ROCs. In the upper graph the measured values for  $T = -10^\circ\text{C}$  (black),  $T = 0^\circ\text{C}$  (red) and  $T = 17^\circ\text{C}$  (blue) are shown. In the lower graph the measured values at  $T = 0^\circ\text{C}$  (red) and the calculated values (blue) are compared.

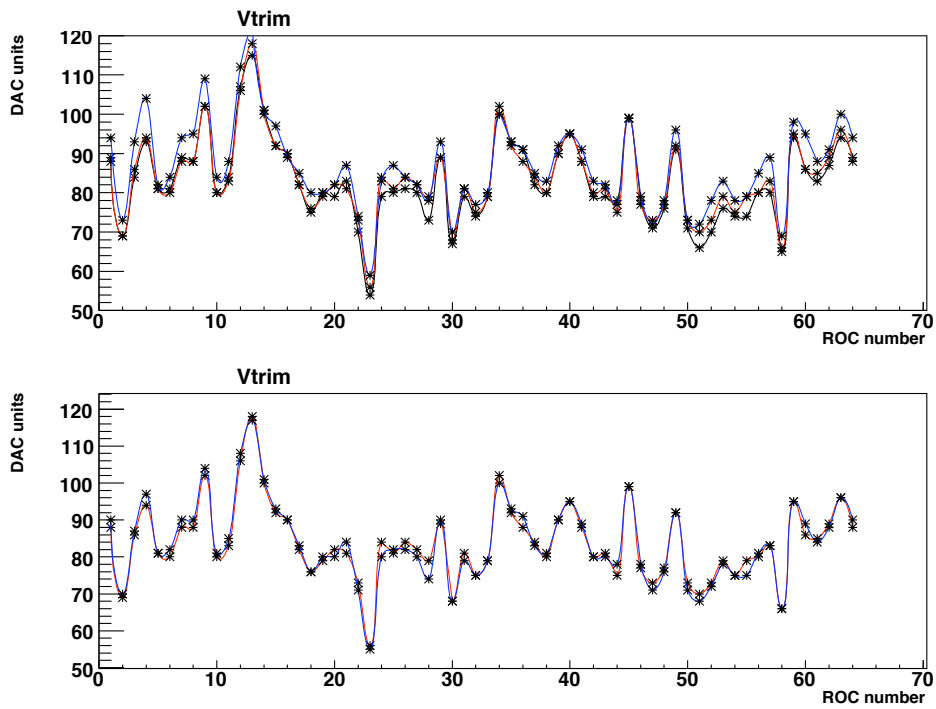


Figure 4.5: Difference between calculated and measured value for  $V_{\text{Ana}}$ , at  $0^\circ\text{C}$  for several ROCs. In the upper graph the measured values for  $T = -10^\circ\text{C}$  (black),  $T = 0^\circ\text{C}$  (red) and  $T = 17^\circ\text{C}$  (blue) are shown. In the lower graph the measured values at  $T = 0^\circ\text{C}$  (red) and the calculated values (blue) are compared.

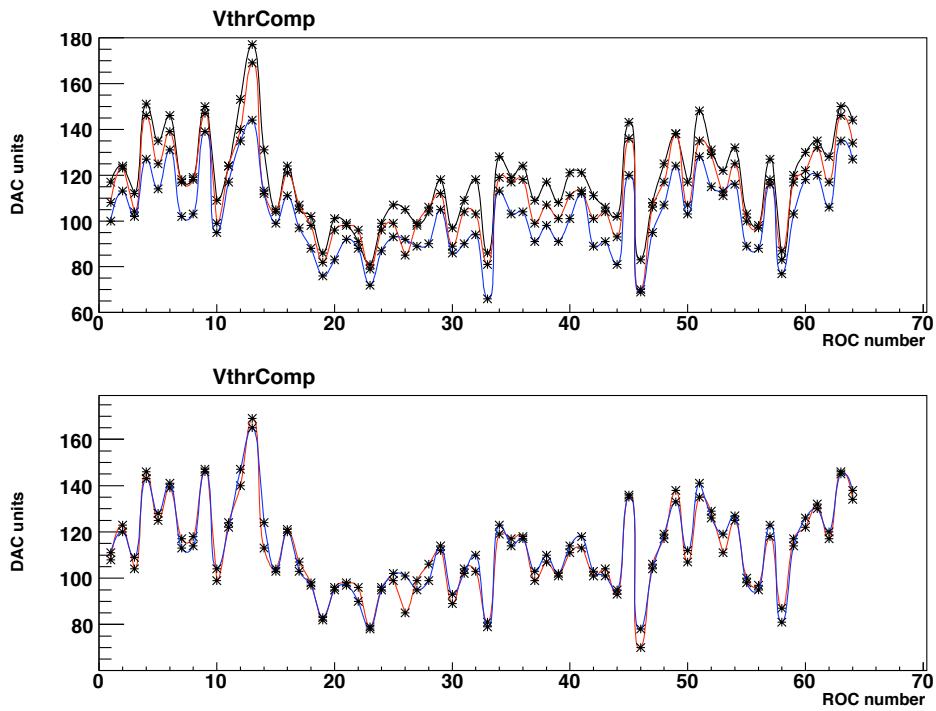


Figure 4.6: Difference between calculated and measured value for  $V_{Ana}$ , at  $0^\circ\text{C}$  for several ROCs. In the upper graph the measured values for  $T = -10^\circ\text{C}$  (black),  $T = 0^\circ\text{C}$  (red) and  $T = 17^\circ\text{C}$  (blue) are shown. In the lower graph the measured values at  $T = 0^\circ\text{C}$  (red) and the calculated values (blue) are compared.

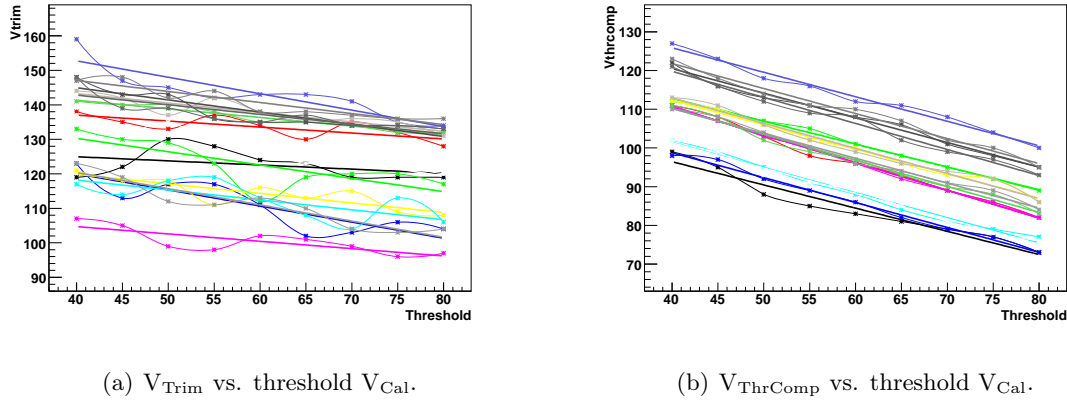


Figure 4.7: linearity of  $V_{Trim}$  and  $V_{ThrComp}$  vs. threshold  $V_{Cal}$ .

the beginning only  $V_{Ana}$ ,  $V_{Trim}$  and  $V_{ThrComp}$  were calculated. With this settings the modules did not work properly. So the decision was made to modify the algorithm so all parameters are linearly interpolated based on the values measured during module testing. For the ones which have a predefined value this does not matter, because they have the same value for both temperatures and for the others (the six mentioned before) it was needed. After this correction the modules worked properly.

In summer 2007 a semester work was done [7] which analyzed the threshold in dependance of  $V_{Trim}$  and  $V_{ThrComp}$ . The graphs are shown in figures 4.7(a) and 4.7(b). The result from this work also is included in these tests. The formulas for  $V_{Trim}$  and  $V_{ThrComp}$  dependent on  $V_{Cal}$  are:

$$V_{ThrComp}(V_{Cal}) = V_{ThrComp}(60) - 0.65 \times (V_{Cal} - 60)$$

$$V_{Trim}(V_{Cal}) = V_{Trim}(60) - 0.45 \times (V_{Cal} - 60).$$

According to these results it is now possible to calculate the values of the parameters depending on the temperature and desired threshold. In figure 4.8 the threshold of all pixels of the four modules used in this test are put in one histogram. The measurements were done with calculated parameters for a temperature of  $0^{\circ}\text{C}$  and a threshold of  $V_{Cal} = 60$ . In figure 4.9 the same is done for a threshold of  $V_{Cal} = 50$ . The mean values are close to target threshold and the RMS is in a good region as well.

#### 4.4 Conclusion

After these investigations it should be possible to calculate the values of important parameters for the trimmed detector. The only problem remaining is to determine the temperature at a certain ROC. In each ROC a temperature sensor is built in. In the next chapter the measurements for the calibration and temperature measurements are described. At the moment it is



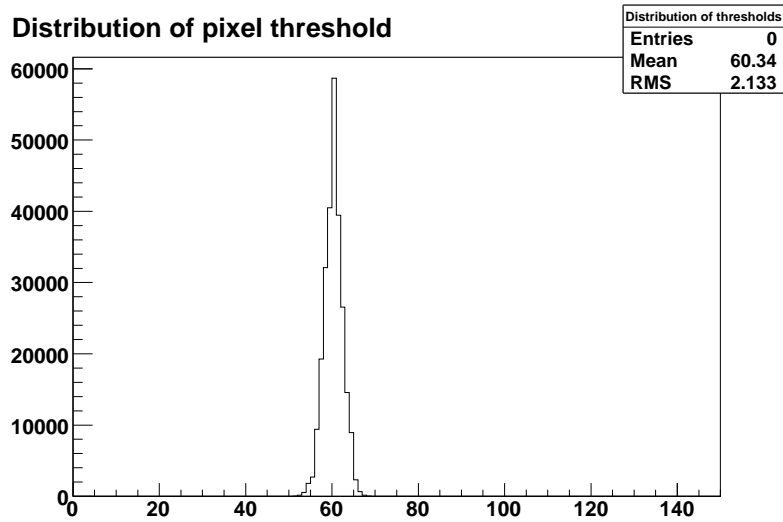


Figure 4.8: Threshold distribution of all pixel with calculated parameters at a threshold of  $V_{\text{Cal}} = 60$  and a temperature of  $T = 0^\circ \text{C}$ .

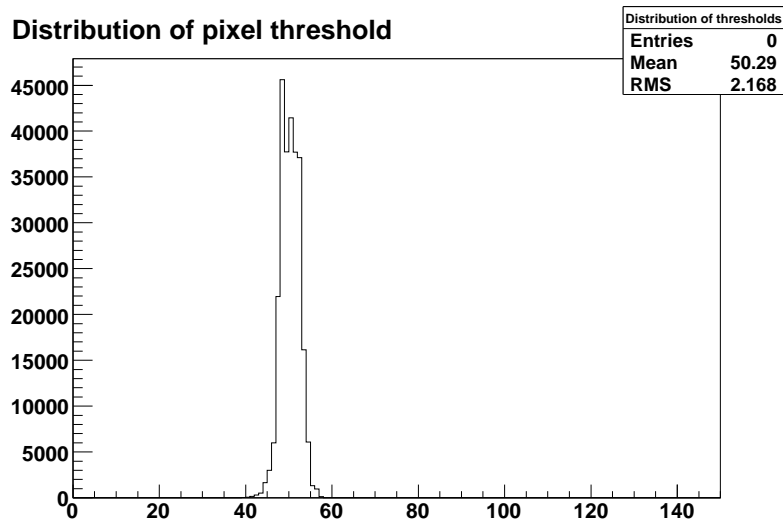


Figure 4.9: Threshold distribution of all pixel with calculated parameters at a threshold of  $V_{\text{Cal}} = 50$  and a temperature of  $T = 0^\circ \text{C}$ .

possible to calculate the parameters for a given temperature and a target threshold. The values will be calculated on the basis of the measured parameters at  $T = -10^{\circ}\text{C}$  and  $T = +17^{\circ}\text{C}$ . Therefore a perl script is produced, which can be executed from the X-DAQ user-interface. It calculates the values of the parameters depending on the temperature and the target threshold based on the measured values during module testing (see appendix for the source code).

## 5 Temperature sensor calibration

### 5.1 Technical specifications of the temperature sensor

In each ROC a temperature sensor is built in. The sensor works with a temperature dependent voltage source. This voltage is compared to a known voltage source which is almost temperature independent. The difference is converted by an ADC. This value can be read out through the last-DAC.

The sensor is designed to measure temperatures in the range  $[-30^{\circ}\text{C}, +70^{\circ}\text{C}]$ . To reach a better accuracy, the gain has been chosen such that not the full range of temperatures can be measured with the same reference voltage. Instead one has to program the appropriate voltage using a 3 bit DAC. This defines 8 different windows of measurement [8]. The reference voltage has to be chosen such that the signal is not too small. But also it must not be too big, because in this case some non-linear effects take place during the signal transmission and the result is not usable anymore. In figure 5.1 a schematic of the temperature sensor is shown. During the module tests the sensor calibration and the values of the temperature dependent source were measured at  $T = -10^{\circ}\text{C}$  and  $T = +17^{\circ}\text{C}$  as well. Assuming the voltage source has a linear dependence in temperature it is possible to calculate the temperature by measuring the voltage.

### 5.2 Calibration

Before the temperature measurement can be done the sensor has to be calibrated. In this case the voltage of the temperature independent sources is taken and compared to a known voltage (470 mV). This is possible because there are eight voltage sources with well known voltage. The difference then is read out of the signal by the last-DAC. With this measurements it is possible to relate the pulse-height to the voltage-difference. The value is set to zero if the difference gets too small or even below zero. Therefore it is needed to choose only "good" points. In figure 5.2 a typical result of the measurement for the calibration of the sensor is shown. In this case the first three points are taken for the calibration. The ADC value is related to the corresponding value in millivolts. Based on the ADC value of the temperature sensitive measurement it is possible to calculate the corresponding voltage. The voltage is then compared with the voltages at  $T = -10^{\circ}\text{C}$  and  $T = +17^{\circ}\text{C}$ . Assuming the voltage source is linear in temperature it is possible to calculate the corresponding temperature.

The problem to find the right measurement window was solved by taking the window with the smallest, but still remaining, signal at both temperatures. This makes sure the linearity of the signal transmission is retained and still readable even if the temperature is higher than  $+17^{\circ}\text{C}$ .

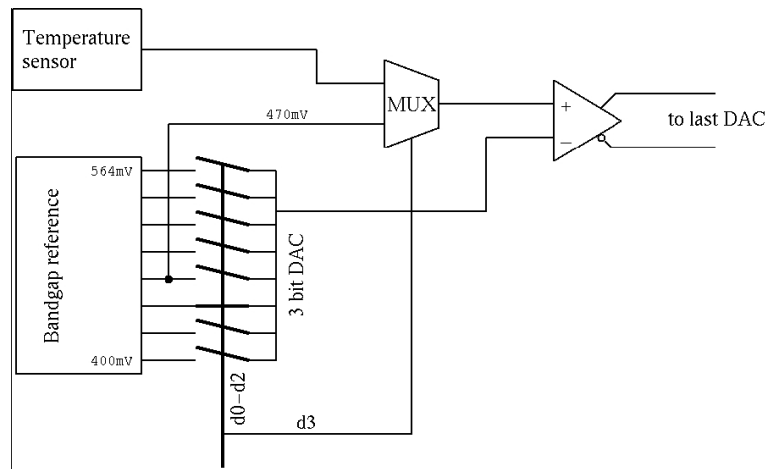


Figure 5.1: Schematic of the temperature sensor built in on each ROC. [8]

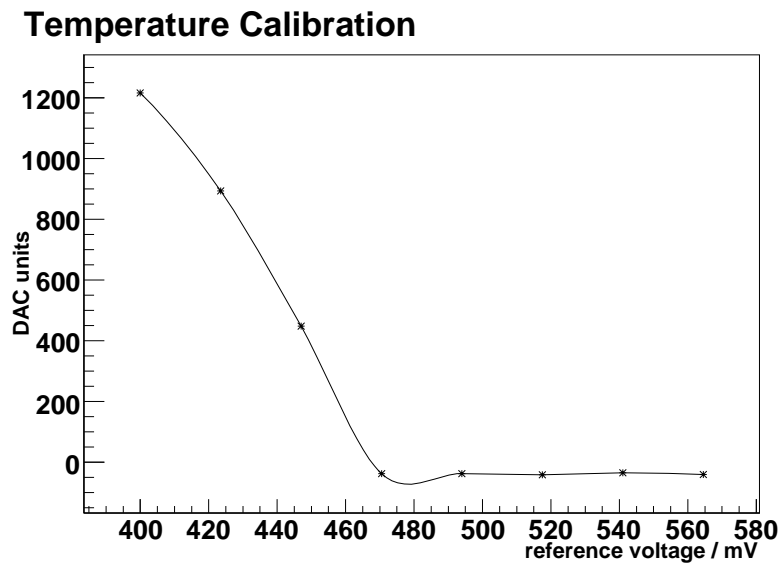


Figure 5.2: Example of a temperature calibration in ADC units vs. reference voltage. Each step of the reference voltage corresponds to a measurement window. The measurement with temperature sensitive source looks quite the same.

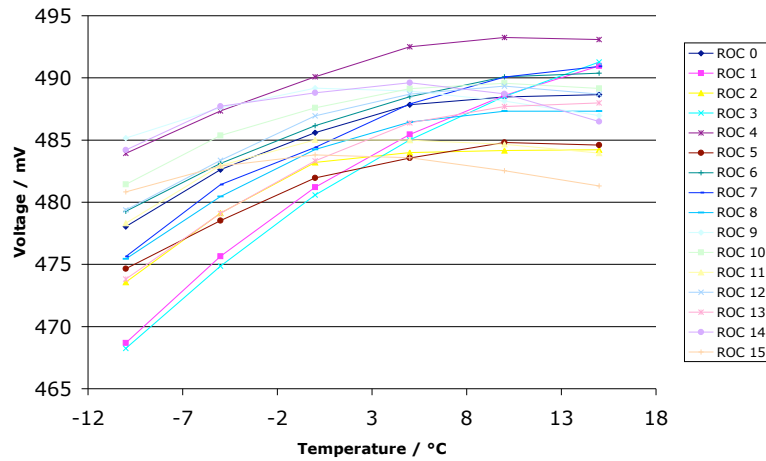


Figure 5.3: Voltage vs. temperature for the temperature window 0.

## 5.3 Measurements

### 5.3.1 Linearity

The measurements were taken in the cooling box. The goal of these measurements was to verify the linearity of the temperature sensor in dependence of temperature. Therefore the measured value for different measurement windows at different temperatures were taken and analyzed. Figure 5.3 contains the data for the measurement window 0. The non-linearity is evident. For most of the ROCs this window can not be used. The non-linearity is assumed to be an effect of the signal transmission, because the signal is very big in this window. As described in the section before, the transmission of the signal gets non-linear if the signal is too big. On the other hand in figure 5.4 the curves are almost linear. In this case it should be possible to calculate the temperature in dependence of the voltage. This is the reason why the window with the smallest signal is chosen for the measurement.

### 5.3.2 Verification

To verify the results from the linearity tests the cooling box was set to different temperatures. The temperature then was measured with the temperature sensors of the ROCs. These results were compared to the temperature in the cooling box. In figures 5.5(a) and 5.5(b) the results are shown. As one can see, the results of a single ROC are not very precise. But for the mean value averaged the whole module the result is usable.

What also had been tested is the reproducibility of the measurement. Therefore the temperature was measured ten times with the same method like before. The results are shown in

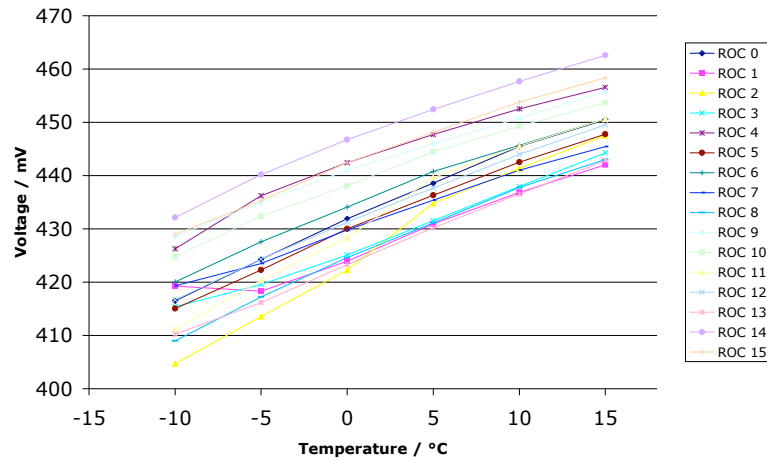
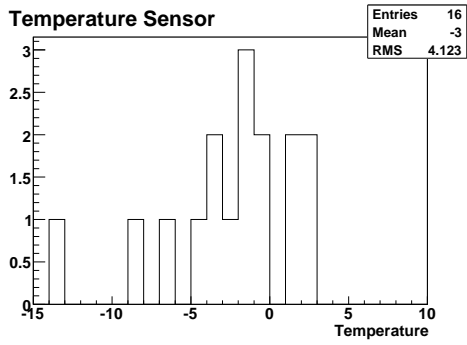
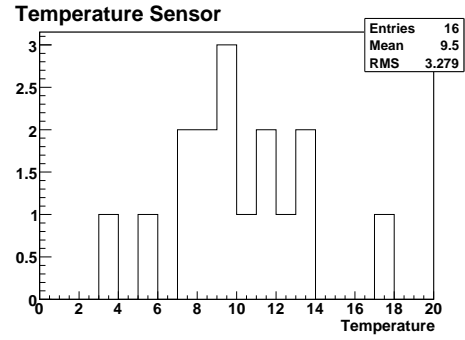


Figure 5.4: Voltage vs. temperature for the temperature window 2.

figures 5.6(a)/(b) and 5.7(a)/(b). The result is only showed for two ROCs per temperature, because the results for the other ROCs looks similar to the showed ones. The RMS of the measurements is between 0 and 0.52. As one can see the result can not been improved much by measuring the temperature more than once. Nevertheless it is possible to determine the temperature with an accuracy of a few degrees.

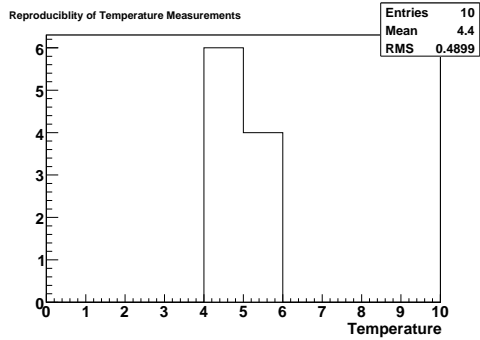


(a) Result at 0° C.

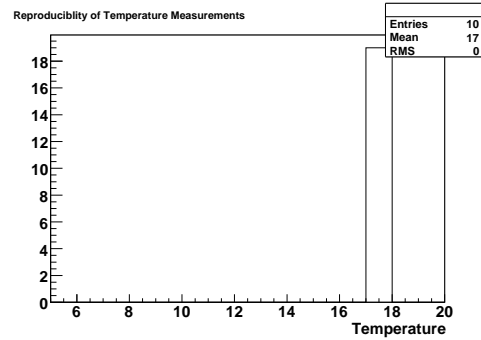


(b) Result at 10° C.

Figure 5.5: Distribution of the temperature measured with the temperature sensor of the ROCs.



(a)



(b)

Figure 5.6: Reproducibility of the temperature measurements with the built in sensor at 10° C. To check if the result can be improved by taken multiple measurements the two ROCs with the highest deviation were taken.

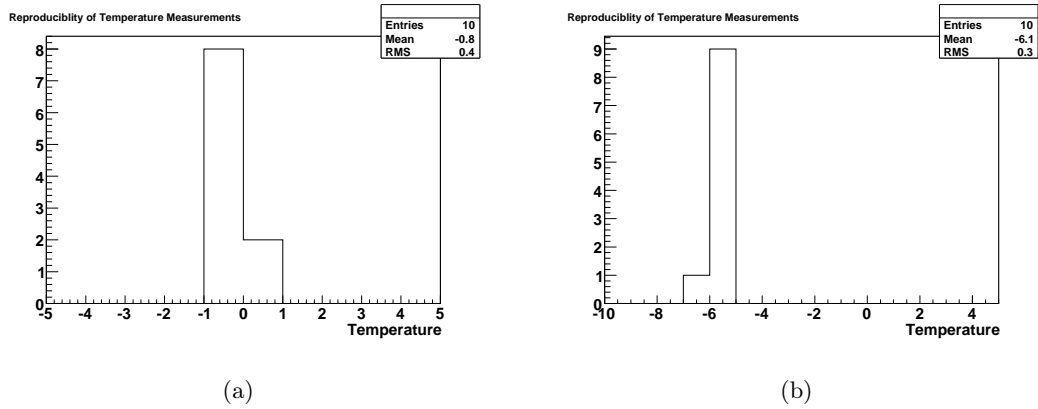


Figure 5.7: Reproducibility of the temperature measurements with the built in sensor at 0° C. For these measurements the ROCs with a mean value of 0° C and -6° C were taken.



## 6 Conclusion

These measurements confirm the assumption that the behavior of the parameters is close to linear. This is correct for the analyzed parameters but also for the temperature measurement itself. For the second one the problem of the right temperature window still exists. In the tests above the window was chosen by looking at the measured values during module testing. This seemed to work for the tested temperatures. It is hard to say if this test still holds if the temperature is far below  $T = -10^{\circ}\text{C}$  or above  $T = +17^{\circ}\text{C}$ .

According to these measurements it can be said, it is possible to measure the temperature of a module. If this temperature is determined it is possible to calculate the values of the temperature dependent parameters as well. The parameterized trimming of the detector therefore is feasible. With these investigations the trimming procedure will be much faster compared to measuring all the parameters. At the moment it is not clear how big the temperature differences in the detector will be. Also it is not known how the temperature gradient looks for a single module. In the cooling box the whole module has contact to the cooling plate, therefore the module has a constant temperature. In the detector the cooling pipes will be at the borders of the modules. In this case the modules probably contain a temperature difference between the ROCs. For this information the temperature measurement has to be implemented in the X-DAQ software. This task is not done yet. The temperature can be measured with some sensors placed on the cooling tubes as well. For the measurement of the temperature with these sensors, the temperature gradient of the modules placed on the detector has to be studied. These sensors will be used as back-up for the built in temperature sensors.

## **7 Acknowledgment**

I want to thank Andrey Starodumov for his patience and supervising and all other members of the pixel group from PSI for the good time. Also I want to thank Urs Langenegger for making this diploma thesis possible.

## A Perl script for the calculations

```
#####  
#                                                                 #  
# Script for calculating the dac settings depending on threshold and #  
# temperature                                                                 #  
#                                                                 #  
# Author: Manuel Waser                                                                 #  
# Date: 5. June 2008                                                                 #  
#                                                                 #  
# usage: perl Mod_Vana -t temperature -s threshold                                                                 #  
#                                                                 #  
#####  
  
#!/usr/bin/perl  
  
use Getopt::Std;  
getopts('t:s:');  
  
if (!$opt_t)  
{  
    $opt_t = 20;  
}  
  
if (!$opt_s)  
{  
    $opt_s = 60;  
}  
  
$k = 1;  
  
$file = '/home/l_tester/manuel/Diploma_thesis/inhalt'; # list of all modules for calculating  
  
open(INFO,$file);  
@lines_a=<INFO>;  
close(INFO);  
  
foreach $lines (@lines_a)  
{  
    if ($k != 1)  
    {
```

```
@line=split(' ', $lines);

#--- --- directory of the dac files for temperature T = -10
$dir_10a=join(' ', '/home/l_tester/manuel/Diploma_thesis/T-10a/', @line[8]);

#--- --- directory of the dac files for temperature T = +17
$dir_17=join(' ', '/home/l_tester/manuel/Diploma_thesis/T+17a/', @line[8]);

#--- --- directory of the final dac files
$dir = join(' ', '/home/l_tester/manuel/Diploma_thesis/testdac/', @line[8]);

if (@line[8] ne "inhalt")
{

#--- --- reading values out of the dac files ----

    open(INFO_2, "$dir_10a");
    @dac_file_10a=<INFO_2>;
    close(INFO_2);

    open(INFO_2, "$dir_17");
    @dac_file_17=<INFO_2>;
    close(INFO_2);

    $linenr = 0;

#--- --- create new file for calculated dacs --- ---

    open(INFO_3, ">$dir");

#--- --- loop over all lines of the dac files --- ---

    foreach $dac_file_10a (@dac_file_10a)
    {

        @dacs_10=split(' ', $dac_file_10a);
        @dacs_17=split(' ', @dac_file_17[$linenr]);

        my $name_10 = @dacs_10[1];
        my $value_10 = @dacs_10[2];

        my $name_17 = @dacs_17[1];
        my $value_17 = @dacs_17[2];
```

```
if ($name_17 ne "ROC:")
{

#--- --- start of caluclating new dacs depending on temperature --- ---

    $slope = ($value_17 - $value_10)/17;

    $value = $value_17 + ($opt_t - 17)*$slope;

    $value_int = int $value;
    if ($value - $value_int > 0.5)
    {
        $value_17 = $value_int + 1;
    }
    else
    {
        $value_17 = $value_int;
    }

#--- --- start of calculating new Vtrim and Vthr depending on threshold --- ---

    if ($name_17 eq "Vtrim:")
    {
        $value = $value_17 - ($opt_s - 60) * 0.45;

        $value_int = int $value;
        if ($value - $value_int > 0.5)
        {
            $value_17 = $value_int + 1;
        }
        else
        {
            $value_17 = $value_int;
        }
    }

    if ($name_17 eq "VcThr:")
    {
        $value = $value_17 - ($opt_s - 60) * 0.65;

        $value_int = int $value;
        if ($value - $value_int > 0.5)
        {
            $value_17 = $value_int + 1;
        }
    }
}
```

```
        else
        {
            $value_17 = $value_int;
        }
    }
}

#--- --- --- end of calculations --- --- ---

#--- --- --- print out nnew values to the file --- --- ---

#    print STDOUT "$name_17 $value_17\n";

    printf INFO_3 '%-15s' , $name_17;
    printf INFO_3 $value_17;
    printf INFO_3 "\n";

    $linenr++;

}
}

#--- --- --- end of loop for one file --- --- ---

    close(INFO_3);
}
    $k++;
}
```

**B Pixel testing manual**

# Contents

<b>1</b>	<b>Introduction</b>	<b>5</b>
<b>2</b>	<b>Testing procedure</b>	<b>6</b>
<b>3</b>	<b>Running the program</b>	<b>7</b>
3.1	Configuration . . . . .	7
3.1.1	detectconfig.dat . . . . .	8
3.1.2	param_fed_XX.dat . . . . .	8
3.2	Power . . . . .	9
3.3	The program crashes during configuration on any test . . . . .	9
3.4	Checking HubID . . . . .	9
<b>4</b>	<b>AOHBias</b>	<b>10</b>
4.1	Procedure . . . . .	10
4.2	Problems . . . . .	10
4.3	Solutions . . . . .	10
4.3.1	Program crashes . . . . .	10
4.3.2	BadTimeSlotInfo . . . . .	12
4.3.3	NoSaturationPt . . . . .	12
4.3.4	Only2TimesSlotReadings . . . . .	12
4.3.5	Difference B-UB is 0 . . . . .	12
<b>5</b>	<b>FEDBaselineCalibration</b>	<b>13</b>
5.1	Procedure . . . . .	13
5.2	Problems . . . . .	13
5.2.1	The calibration does not converge . . . . .	13
5.2.2	Small difference between B and UB . . . . .	14
5.2.3	UB too low . . . . .	14
5.2.4	UB too high . . . . .	14
5.2.5	All shifted up or down . . . . .	14
<b>6</b>	<b>TBMUB</b>	<b>15</b>
6.1	Procedure . . . . .	15
6.2	Problems . . . . .	15
6.3	Solutions . . . . .	15
6.3.1	Difference does not reach zero . . . . .	15



*Contents*

---

6.3.2	The difference of the AnalogOutputGain is too big . . . . .	15
6.3.3	Difference is always below zero . . . . .	15
<b>7</b>	<b>ROCUB</b>	<b>16</b>
7.1	Procedure . . . . .	16
7.2	Problems . . . . .	16
7.3	Solutions . . . . .	16
7.3.1	Difference starts below zero . . . . .	16
7.3.2	Difference does not reach zero . . . . .	16
<b>8</b>	<b>VcThrCalDelFIFO1</b>	<b>17</b>
8.1	Procedure . . . . .	17
8.2	Visualization . . . . .	17
8.3	Problems . . . . .	17
8.4	Solutions . . . . .	17
8.4.1	Low efficiency . . . . .	17
8.4.2	Too small working area . . . . .	17
8.4.3	No working area found . . . . .	17
<b>9</b>	<b>AddressLevelCalibration</b>	<b>19</b>
9.1	Procedure . . . . .	19
9.2	Problems . . . . .	19
9.3	Solutions . . . . .	19
9.3.1	One peak is out of range . . . . .	19
9.3.2	The peaks are too broad . . . . .	19
9.3.3	Small peaks occur next to the main peaks . . . . .	19
<b>10</b>	<b>PixelAlive</b>	<b>21</b>
10.1	Procedure . . . . .	21
10.2	Visualization . . . . .	21
10.3	Problems . . . . .	21
10.4	Solutions . . . . .	21
10.4.1	No hits are produced . . . . .	22
10.4.2	Periodic pattern appears on a ROC . . . . .	22
10.4.3	Hits are produced at the wrong position . . . . .	22
<b>11</b>	<b>HV-Test</b>	<b>23</b>
11.1	Procedure . . . . .	23
11.2	Problems . . . . .	23
11.3	Finishing . . . . .	23
<b>12</b>	<b>Appendix</b>	<b>24</b>
12.1	List of commands to update configuration-files . . . . .	24

*Contents*

---

12.2 Final thoughts . . . . . 24

# 1 Introduction

In this manual the normal procedure for testing the CMS pixel detector is described. It is based on my experience testing the -x side of the detector. There is a chapter for each test, where the procedure is defined, the possible problems are listed and at the end of each chapter the (hopefully successful) solutions.

The used software framework is called X-DAQ. The configurations are shared on different files. Their locations and usage is described in the chapter where they are used. Further there are configuration files, which contain information about the modules to be tested or channels which are needed. Also there are files which contain the information which channel corresponds to which module. In the following chapters it is described when and how these files have to be adjusted. These files are located in different directories at `TriDAS/PixelConfigDataExamples/`. The proper directories are listed in the chapter ‘Configuration’. For each calibration exists a configuration-file, which contains the important settings for the configuration. These files are located at `TriDAS/PixelConfigDataExamples/calib/NUMBER/calib.dat`. The NUMBER corresponds to the calibration. It is listed in the chapter testing procedure.

Each module has its channel and HubID. These values are unique for each sector. There is one file which relates all these identifications. With this information each module can be identified. In chapter 3.4 it is described how to check these relations.

The results of the calibration will be stored in the running directory. After each test one has to manually update the proper configuration file. The corresponding commands are listed at the end of this document in table 12.1. After some calibrations it is needed to look at the results manually before updating. In this case either a html-file or a postscript-file is produced.

For the solution of some problems external programs are needed. Their usage is described at <https://twiki.cern.ch/twiki/bin/view/CMS/PixelOnlinePSI> in the corresponding section.

## 2 Testing procedure

During testing the following calibrations have to be done:

1. Delay25 (40)
2. AOHBias (45)
3. FEDBaseLineCalibration (43)
4. TBMUB (70)
5. ROCUBEqualisation (71)
6. ClockPhaseCalibration (49)
7. VcThrCalDelFIFO1 (42)
8. AddressLevel (44)
9. PixelAlive (47)
10. HV-test (47)

The number in the brackets corresponds to the directory where the calib.dat-file is located. In the following sections the problems encountered and their handling will be discussed. After each test the proper configuration file has to be updated. The list is located in Table 12.1 at the end of this document.

## 3 Running the program

To run the program one has to be in this directory:

`~/TriDAS/pixel/PixelRun/`

The program is started with `./run.sh`.

The program is controlled by a web-interface. The link can be found at the bookmarks of the browser. In this directory also the results will be saved.

### 3.1 Configuration

Before running the program one has to prepare a few configuration files which are located in the `/TriDAS/PixelConfigDataExamples` directory.

- `detconfig/4/detectconfig.dat`
- `fedcard/4/param_fed_XX.dat`
- `calib/40/delay25.dat`

If one switches from -z-side to +z-side or opposite, also the following files should be checked:

- `portcardmap/4/portcardmap.dat`
- `nametranslation/4/translation.dat`

These files are located in:

`/home/l_pixel/TriDAS/PixelConifgDataExamples_final_psi/portcardmap/4`

for the `portcardmap.dat`, respectively

`/home/l_pixel/TriDAS/PixelConifgDataExamples_final_psi/nametranslation/4`

for the `translation.dat`.

It also has to be checked if the portcard-file for the sector exists. The file is located at:

`/home/l_pixel/TriDAS/PixelConifgDataExamples_final_psi/portcard/4`

and is called `portcard_BPix_B(m/p)(I/O)_SECX_PRT(1/2).dat`. If this file does not exist for the sector to be tested, one can simply copy from an other sector, but has to ensure the ccu-address is right. This address can be checked with the `pxfec` program. The usage of this program is described at <https://twiki.cern.ch/twiki/bin/view/CMS/PixelOnlinePSI> in the chapter `pxfec`.

### 3.1.1 detectconfig.dat

The detectconfig.dat file is located at:

/home/l\_pixel/TriDAS/PixelConifgDataExamples\_final\_psi/detconfig/4

The easiest way to modify this file is to open it in an editor, and then simply copy the names of the modules to be tested from the detectconfig\_B(m/p)(O/I)\_SECX.dat. These files can also be found in the same directory.

### 3.1.2 param\_fed\_XX.dat

This file can be found at:

/home/l\_pixel/TriDAS/PixelConifgDataExamples\_final\_psi/fedcard/4

In this file close to the end the channel enable bits are located. Here the used channels have to be enabled (0) and the unused have to be disabled (1). The coding is simply the binary number translated to hex. The enumeration starts from right to left (the first channel is the rightmost bit). For example  $0x13b = 0b100111011$ , in this case channel 3,7 and 8 are enabled while the others are disabled. The channels are grouped by 9. In the following tables one can see the configuration for standard tests. The difference between sector 4,5 and the other ones, is that they only contain 8 modules on layer 3 instead of 12. Therefor more channels have to be closed.

Table 3.1: Channel enable bits for sectors 1-3, 6-8.

Layer 1/2	chnls 1-9: 0x000 chnls 10-18: 0x000 chnls 19-27: 0x1c0 chnls 28-36: 0x1ff
Layer 3	chnls 1-9: 0x1ff chnls 10-18: 0x1ff chnls 19-27: 0x03f chnls 28-36: 0x000

Table 3.2: Channel enable bits for sectors 4, 5.

Layer 1/2	chnls 1-9: 0x000 chnls 10-18: 0x000 chnls 19-27: 0x1c0 chnls 28-36: 0x1ff
Layer 3	chnls 1-9: 0x1ff chnls 10-18: 0x1ff chnls 19-27: 0x0ff chnls 28-36: 0x006

## 3.2 Power

To turn on and off the power one can either connect to the power supply with telnet:  
telnet 128.129.202.35:1527

the username is: 'user' and the password is also : 'user'. The other possibility is to use the keyboard connected to the caen.

Once connected first one has to choose 'channels' in the menu 'main'. This is done by first pushing <TAB> and then <RETURN>. For layer 1/2 one has to turn on L12 position VD12. The high voltage is turned on at HV0 and HV1. For layer 3 one has to turn on L3 position VD3, here also the high voltage is located at HV0 and HV1. The power of the AOHs is turned on at Sec1-4 and Sec5-8 all eight and for the ccu CCU position CH00. If the power supply is turned on the first time for the AOHs and the ccu the switches are locked. To unlock them one has to enable the last column and change the value from Dis to En.

## 3.3 The program crashes during configuration on any test

There are different problems which cause this effect. One can try the following options.

1. check if the detectconfig is properly set
2. check if the power is on for the right layer, the aohs and the ccu
3. check the connections
4. turn power off and on again
5. turn off and on the power of the crate

Probably this procedure has to be repeated multiple times.

## 3.4 Checking HubID

The easiest way to check the HubID is to send periodic triggers with the X-DAQ software. Then it is possible to check the signal with the FedConsumer. The usage is described at <https://twiki.cern.ch/twiki/bin/view/CMS/PixelOnlinePSI>. To check the HubID it is useful to change the 'tbn inputbias' to 50. This should decrease the amplitude of the signal. So it is possible to see which channel has changed. With this it should be possible to check the HubID. The other possibility is to compare the table with the channels with the list Beat Meier made during the mounting of the modules.

## 4 AOHBias

### 4.1 Procedure

During this test the AOHBias is scanned and set to an optimal value. First the software scans the signal to find the position of the TBM-header. This is done for all channels simultaneously. Then for each channel the difference between black and ultra-black is measured. The characteristic then is stored in a ps-file (AOHBias\_FED\_XX\_channel\_XX.ps). It is useful to have a look at these pictures because it is possible to see if there is a problem with a fiber. If the fibre is dirty the slope is remarkably lower than normal and also the noise is very high (see figure 4.1). The AOHBias values are stored in the portcard-file (portcard\_BPix\_B(m/p)(O/I)\_SECX\_PRT(1/2).dat; PRT1 is related to layer 1/2, PRT2 is related to layer 3). It is located at:  
/home/l\_pixel/TriDAS/PixelConifgDataExamples\_final\_psi/portcard/4  
The proper AOHBias can be found on the 'channel table', the printout which relates the module name to the channel number.

### 4.2 Problems

During this test the following problems can occur:

- program crashes
- BadTimeSlotInfo
- NoSaturationPt
- Only2TimesSlotReadings
- Difference B-UB is 0

### 4.3 Solutions

#### 4.3.1 Program crashes

If the program crashes and the message 'too few entries 0' is printed, do a 'reset' and 'piareset all' in the ccu-program (the usage is described on the webpage) and have a look at the signal with FEDBaseLineCalibration. It also is possible that there is no signal at all. In this case the connections should be checked. If nothing helps one can also exclude this module from the detectconfig.dat.



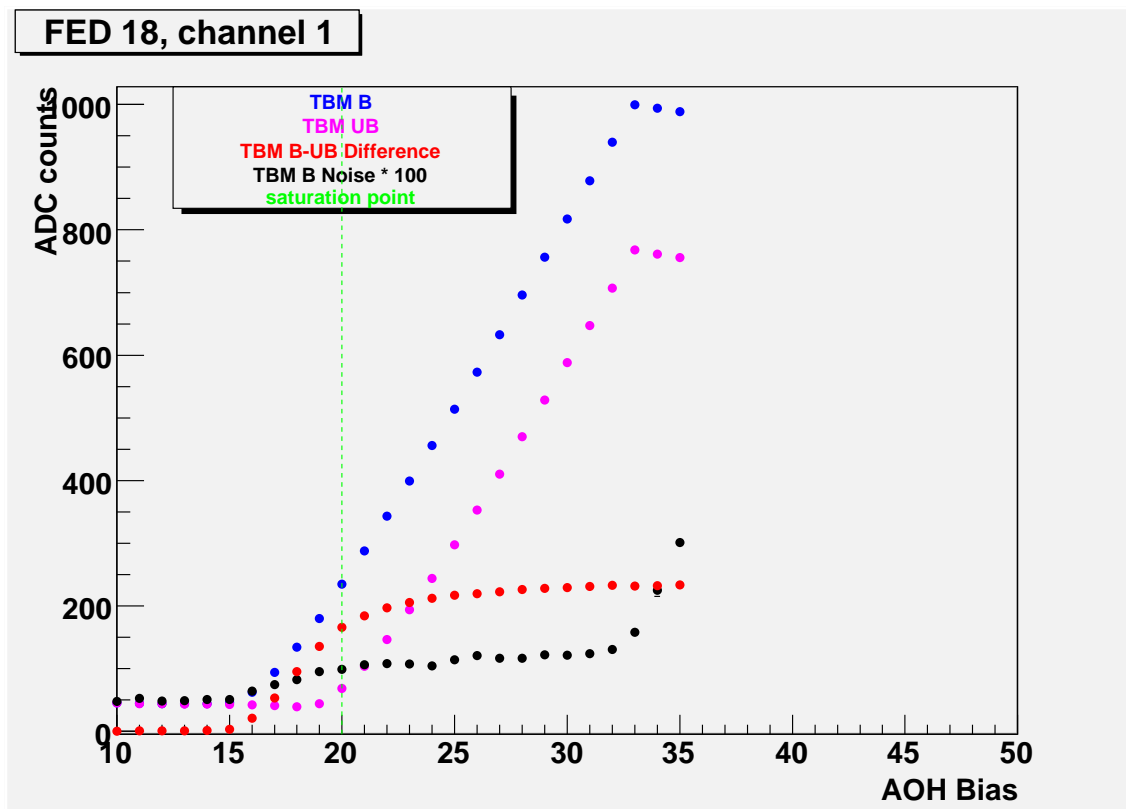


Figure 4.1: Example of an AOHBias-scan.

### 4.3.2 BadTimeSlotInfo

One can try to run FEDBaseLineCalibration to look at the signal. If the signal has a bad timing (the three clocks of the TBM-header are not properly aligned), one can try to fix the problem by running ClockPhaseCalibration. If this does not fix the problem or the signal looks good, one should make a remark. Then it is possible to ignore this problem for further tests and continue.

### 4.3.3 NoSaturationPt

This is a problem of the algorithm. To cure this you have to look at the ps-file, which is produced during this test and the value has to be set by hand in the portcard-file or simply try to rerun this test. Often this problem is related to a bad point during measurement. This means for one point the measurement did not work. It is then not possible for the algorithm to interpret the result. If one looks at the curve manually, normally the value for the AOHBias can be set.

### 4.3.4 Only2TimesSlotReadings

Run FEDBaseline and look for a problematic channel. This helps to figure out the problem.

### 4.3.5 Difference B-UB is 0

Run FEDBaseline and look for a problematic channel. Usually one can see a signal but with a strongly reduced amplitude. Fiber cleaning helps to recover the signal.

## 5 FEDBaselineCalibration

### 5.1 Procedure

In this test the baseline gets adjusted to the targeted black-value (512). The program stops if the value of the offset does not change anymore or if it does not converge after 16 iterations. At the end a HTML-file is produced for a closer look at the signal and the position of the black-level. This file is produced only if the test converges. In figure 5.1 a good signal is shown.

### 5.2 Problems

- The calibration does not converge
- Small difference between B and UB
- UB too low: close to 0 or even below 0
- UB too high
- All shifted up or down

#### 5.2.1 The calibration does not converge

In this case it should be tried to evaluate the ‘bad’ channel to have a closer look at the signal. It is possible to have a look at the picture of the signal with kvview even when the test does not converge because the .gif files are produced anyway. The name of the .gif file is: FIFO1Signal\_ITERATION\_FEDNR\_CHANNEL.gif. If the signal of the bad channel looks fine, just try to rerun the test. If there is no signal or it has a bad shape,

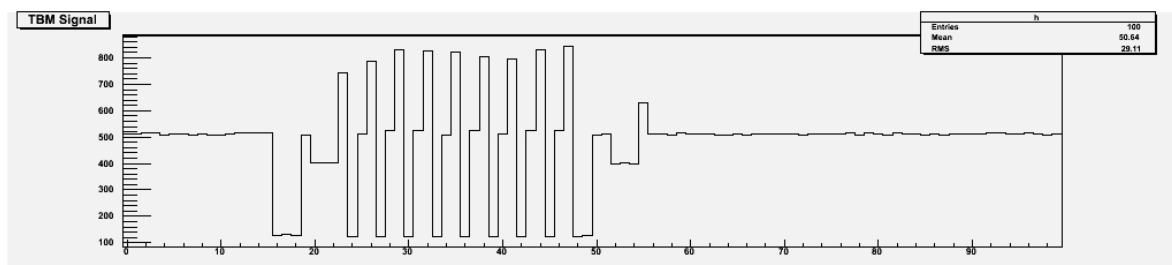


Figure 5.1: Shape of a good signal.

one can try to fix in by sending a hard reset by using the ccu-program. The usage is described at <https://twiki.cern.ch/twiki/bin/view/CMS/PixelOnlinePSI>. The command is 'piareset all'.

### 5.2.2 Small difference between B and UB

Fiber cleaning helps.

### 5.2.3 UB too low

Decrease the gain of the corresponding channel. If TBM runs in dual mode, increase the gain of the partner channel of the same TBM

### 5.2.4 UB too high

Increase the gain of the corresponding channel.

### 5.2.5 All shifted up or down

The output signal is a differential one. So, if one wire is broken such a pattern can be observed. Non recoverable defect. Channel should be disabled.

Once I made a mistake in the param\_fed\_XX.dat file after which the program tried to adjust the baseline to 0. If this happens check the channel enable bits for bad characters.

## 6 TBMUB

### 6.1 Procedure

This test scans the TBM AnalogOutputGain and sets a proper value, so the ultra-black (UB) is located at 150. In layer 1/2 only the value can only be set once for both channels of a module. Therefor the higher value of both channels is taken, so one can be sure both channels have their UB at 150 or below. In the printout the difference of TBMUB and the target-UB is printed.

### 6.2 Problems

If the test fails for a module there can be the following problems:

- difference does not reach zero
- the difference of the AnalogOutputGain is too big
- difference is always below zero

### 6.3 Solutions

#### 6.3.1 Difference does not reach zero

Mostly it works, if one increases the AOH\_Gain in the portcard-file.

#### 6.3.2 The difference of the AnalogOutputGain is too big

Increase the AOH\_Gain for the channel with the higher AnalogOutputGain.

#### 6.3.3 Difference is always below zero

If this happens the AOH\_Gain is set too high. So one should try to decrease this value. If TBM is in a dual mode, increase the gain of the partner channel of the same TBM.

## 7 ROCUB

### 7.1 Procedure

During this test the ROC-ultra-black (ROCUB) also gets adjusted to 150. The scanning parameter is  $V_{IBiasDac}$ . This parameter is also important for the separation of the address-levels. The value in the print-out is the difference between ROCUB and the target-UB.

### 7.2 Problems

The following problems appeared during this calibration:

- difference starts below zero
- difference does not reach zero

### 7.3 Solutions

#### 7.3.1 Difference starts below zero

If the difference starts below zero the AOH\_Gain probably is set too high. One can try to decrease this value. In this case the TBMUB has to be repeated to get sure the TBMUB is still at 150 or below.

#### 7.3.2 Difference does not reach zero

If the difference does not reach zero the opposite has to be done. Try to increase the AOH\_Gain. In this case the TBMUB also has to be redone.

## 8 VcThrCalDelFIFO1

### 8.1 Procedure

In this calibration a 2-dimensional scan is done to find the working region. For this a hit on four pixels is produced and read out. The varied parameters are  $V_{cThr}$  and CalDel. The result is stored in one file per ROC, it is called: ThrCalDelScan\_BPix\_...dat.

### 8.2 Visualization

The result has to be visualized by a root macro: thresholdCalDelay.c. This will produce a ps-file with the working region and it also puts a black cross at the point of the chosen values. This file has to be checked if the test worked for all ROCs. A good result is if the red working region has the shape of a 'tornado' (see figure 8.1).

### 8.3 Problems

- A problem which can appear in this test is that one of the chosen pixel is a dead pixel. Then the efficiency will never reach 100% but a multiple of 25%
- Too small working area
- No working area found

### 8.4 Solutions

#### 8.4.1 Low efficiency

In this case one has to set the values by hand or choose four different pixels unless one knows which pixel is dead.

#### 8.4.2 Too small working area

One can try to run ClockPhaseCalibration and Delay25 to adjust time to send commands to the chip.

#### 8.4.3 No working area found

Just leave the default settings for the chip which has this problem.

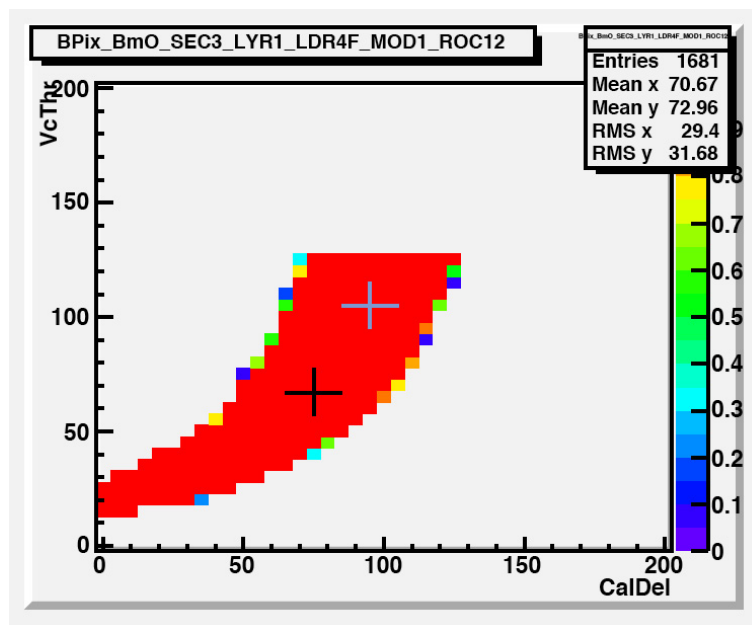


Figure 8.1: Example of a working area.



## 9 AddressLevelCalibration

### 9.1 Procedure

In this calibration the address-levels for the pixel-position are defined. During this test all pixels get a hit and the signal is analyzed. The results are filled into a histogram which can be inspected with the web-browser (see figure 9.1).

### 9.2 Problems

During this test the following problems appeared:

- one peak is out of range
- the peaks are too broad
- small peaks occur next to the main peaks

### 9.3 Solutions

#### 9.3.1 One peak is out of range

If this happens the  $V_{IBiasDac}$  is set to high. It is either possible to reduce this value by hand (it is located in the ROC\_DAC\_BPix\_...dat file) or if the ROCUB is far below 150 one can simply rerun the ROCUBCalibration.

#### 9.3.2 The peaks are too broad

This case points to a problem with noise on the signal. Try cleaning the fibers. If this does not help one can try to run ClockPhaseCalibrations if there is a bad timing on the signal.

#### 9.3.3 Small peaks occur next to the main peaks

If this happens one should check in the log-file if the BaselineCorrection is large. If this correction is too big the test does not run properly anymore. So one can simply rerun FEDBaseLineCalibration and then rerun AddressLevelCalibration.

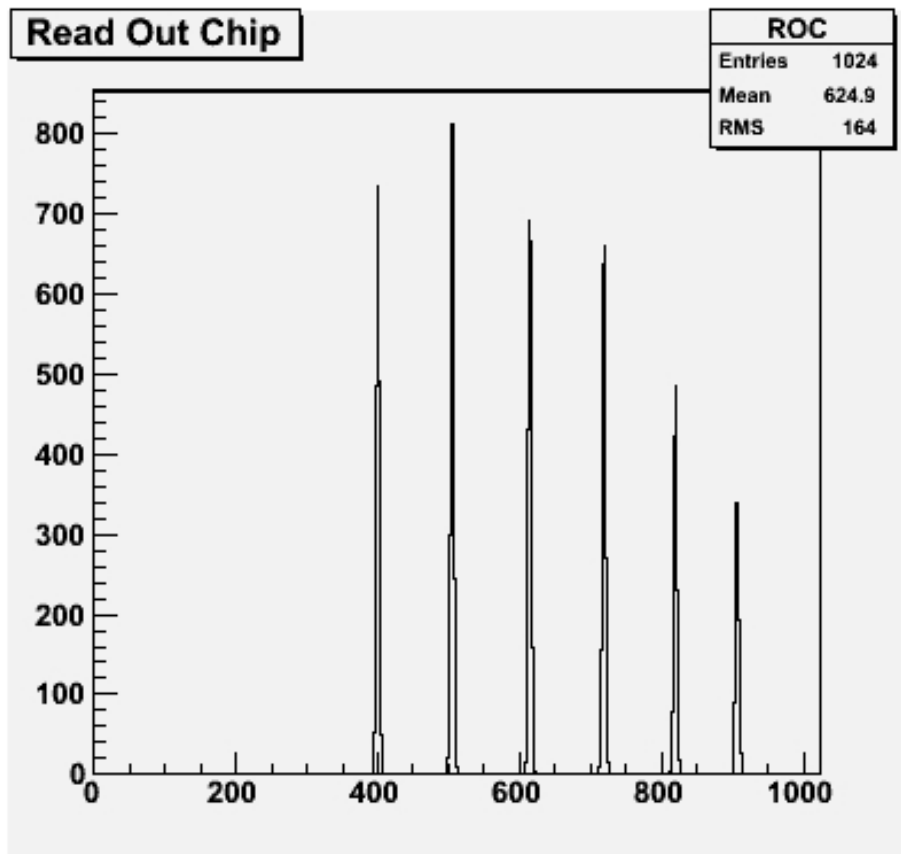


Figure 9.1: Example of well separated address levels.

## 10 PixelAlive

### 10.1 Procedure

This test is the main goal in testing the detector. It tells which pixels are dead and which are working. During this test in each pixel 10 hits will be produced and read out. The result is stored in PixelAlive\_RUNNR.dmp.

### 10.2 Visualization

The result can be visualized by a root macro called pixel\_alive.c. For this one has to link PixelAlive\_RUNNR.dmp to PixelAlive.dmp. Also a calib\_full.dat\_copy has to be copied to calib.dat. One has also to modify the script for layer 1/2 respectively layer 3. This is at the beginning of the script. For layer 1/2 the three lines below L3 links have to be commented, for layer 3 the three lines below L12 links.

### 10.3 Problems

During this test one has no more the possibility to look at the signal because the FED runs in 'intelligent' mode. It just gives the information at which pixel a hit was produced to the computer. If anything goes wrong the error-message will be the same. Here is a list of problems that occurred during this test:

- no hits produced
- periodic pattern appears on a ROC
- hits are produced at the wrong position

### 10.4 Solutions

In general it is difficult to locate the problem during the PixelAlive scan. If nothing works one can try to look at the signal with the fedscope or the FEDBaseLineCalibration. If the signal looks fine try one of the following procedures.

#### **10.4.1 No hits are produced**

In this case one can try to rerun AddressLevelCalibration. If the time between the last tests and PixelAlive is big one has probably to repeat the whole calibration procedure, because most of the parameters are dependent on temperature. One should start with FEDBaseLineCalibration so one can check if the UBs still are well adjusted. In this case it is possible to start with VcThrCalDel.

#### **10.4.2 Periodic pattern appears on a ROC**

For this problem we do not have a proper solution yet. Probably the reason for this pattern is only one noisy pixel. In one case the pattern disappeared, when we excluded this one pixel. To find this pixel it is possible to compare the pattern with the order of the pixels in the calib.dat file. It is possible to change the order to get sure the right pixel has been chosen.

#### **10.4.3 Hits are produced at the wrong position**

This problem can happen if the high voltage is not turned on. If high voltage is turned on there is probably a problem with the address-levels. So this test has to be repeated.

## 11 HV-Test

### 11.1 Procedure

This test is in principle a repetition of the PixelAlive. The PixelAlive-test has to be done twice, once with and once without high voltage turned on. By comparing the results (visualized by the root macro) it is possible to see if the high voltage supply is working. With high voltage turned on remarkably more hits have to be visible then with no high voltage.

### 11.2 Problems

Until now no problems occurred during this test.

### 11.3 Finishing

After the HV-test is done one has to move the data to a backup directory. For this one has to create a new directory in:

/home/l\_pixel/data

The new directory should be named like: B(m/p)(I/O)\_SX\_L(12/3).

## 12 Appendix

### 12.1 List of commands to update configuration-files

The following table (12.1) contains the list of commands to update the proper configuration files after the calibrations.

Table 12.1: List of the commands to update the proper files.

Test	file to update
AOHBias	./updatePortcard
FEDBaseLineCalibration	./updateFed
TBMUB	./updateTbm
ROCUBEqualization	./updateRoc
VcThrCalDel	./updateRoc
Addresslevel	./updateFed

### 12.2 Final thoughts

After a few changes in the program code the procedure has become more stable. Less problems occurred during running the system. Most of the problems should be solvable with the solutions I have given here. If some new problems appear feel free to add them to this document.

## C References

- [1] "Quarks & Leptones: An Introductory Course in Modern Particles Physics"; Francis Halzen, Alan D.Martin; Wiley 1984
- [2] [http://lhc-machine-outreach.web.cern.ch/lhc-machine-outreach/lhc\\_in\\_pictures.htm](http://lhc-machine-outreach.web.cern.ch/lhc-machine-outreach/lhc_in_pictures.htm) (23.6.2008)
- [3] <http://cmsinfo.cern.ch/outreach/CMSdocuments/CMSplots/CMSplots.html> (3.7.2008)
- [4] <http://doc.cern.ch//archive/electronic/cern/others/PHO/photo-bul//bul-pho-2007-079.jpg>; Michael Hoch (December 2007)
- [5] "How do we calibrate in the lab?"; P. Trüb, A. Starodumov, U. Langenegger (January 2007); [http://kamor.ethz.ch/pres/2007/as\\_2007-01-14.pdf](http://kamor.ethz.ch/pres/2007/as_2007-01-14.pdf)
- [6] "CMS Barrel Pixel Module Qualification"; S.Dambach, C.Eggel, U.Langenegger, A.Starodumov, P. Trüb
- [7] "Parameterized trimming of the CMS pixel detector modules"; M. Waser (June 2007) <http://www.phys.ethz.ch/~ursl/home/v0/projects/p05.pdf>
- [8] "PSI46 - External Specifications"; K. Gabathuler (2004) <http://kamor.ethz.ch/ref/PSI46v2.pdf>