Physik Institut der Universität Zürich

– Bachelor Thesis –

Extended Measurements of Charge Sharing at the TT Test Stand

Marco Tresch
Supervisor: Dr. Olaf Steinkamp

August 4, 2011

Charge sharing between strips in silicon micro-strip detecors can be exploited to improve the spatial resolution. Charge loss between strips can cause a drop in the hit efficiency. Early measurements on prototype detectors for the LHCb silicon Tracker with lasers and test beams show large charge sharing and charge loss in between strips. Measurements in LHCb show the opposite, small charge sharing and no charge loss in between strips.
This thesis covers studies of charge sharing and charge loss in between strips using a pulsed, focussed infra-red laser to generate charges at precisely known locations on the silicon sensor. For this purpose, measurements were taken under different conditions, for example, with different temperatures and amplifications. Furthermore it describes the laser alignment which gives an indication for geometrical charge sharing.

Acknowledgements

This thesis was only possible with the help and knowledge of many people from the LHCb group. Specially I would thank my supervisor Dr. Olaf Steinkamp for the numerous discussions and his support. Further thanks go to Christian Elsasser for the sharing of his knowledge of the Test Stand mannerism. I thanks also Dr. Roland Bernet for the help with the IT part of the experiment and Dr. Achim Vollhardt for the discussions and his knowledge. Next I thanks Dr. Mark Tobin for the fast help with the Data-Acquisition and the discussions on the problems of the thesis.

Special thanks go to Prof. Ulrich Straumann for the opportunity to write the thesis in his group and also for the discussions about it.

Finally I thank all the people who helped me with the thesis to find the right words or those who motivate me to go further, without them the thesis wasn't possible.

Contents

Introduction

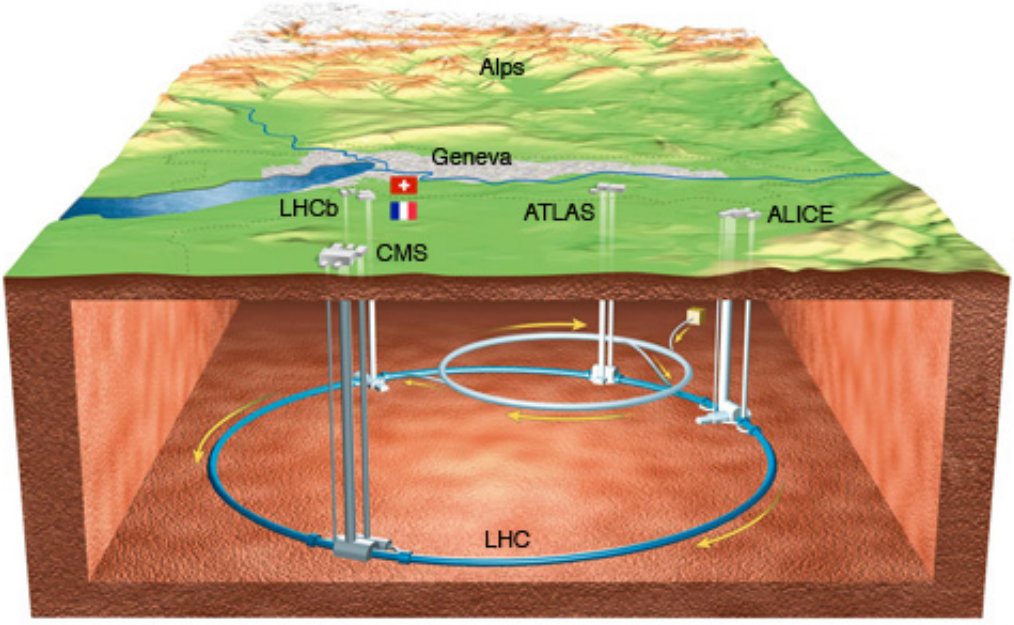


Figure 0.1.: CERN LHC

The Large Hadron Collider (LHC) is a particle accelerator near Geneva. Inside the accelerator, two beams of hadrons, either protons or ions, travel in opposite direction with a maximum energy of 7 TeV. Different experiments are installed at the LHC, one of them is LHCb. LHCb is optimized to study decays of B-mesons, which have a peaked cross-section on a small angle to the beam-pipe. The experiment consists of different components located around the beam-pipe in the forward direction (see figure 0.2). Each sub-detector measures a different characteristic of the particles produced by the colliding hadrons.
One of these subdetectors is the TT (Tracker Turicensis), which is designed to measure the track-position of charged particles in front of the magnet. The tracker is 150 cm wide and 130 cm high. It consists of four layers of silicon modules (fig. 1.1) and has an active area of about 8.4 m^2. The layers are arranged in pairs (x,u) and (v,x). Each of these contains pairs of half-modules which form one full-module. The modules are designed to reduce dead material inside the acceptance of the experiment, so the readout hybrids are located at the end of each half module. To have track information in x- and y-direction the modules in the u and v detection layer are rotated by the respective stereo angle of 5^o. To test the half modules, before the final build of the TT, the test stand

was constructed. Afterwards it would extended to be able to analyze the properties of the half modules [7, 14].

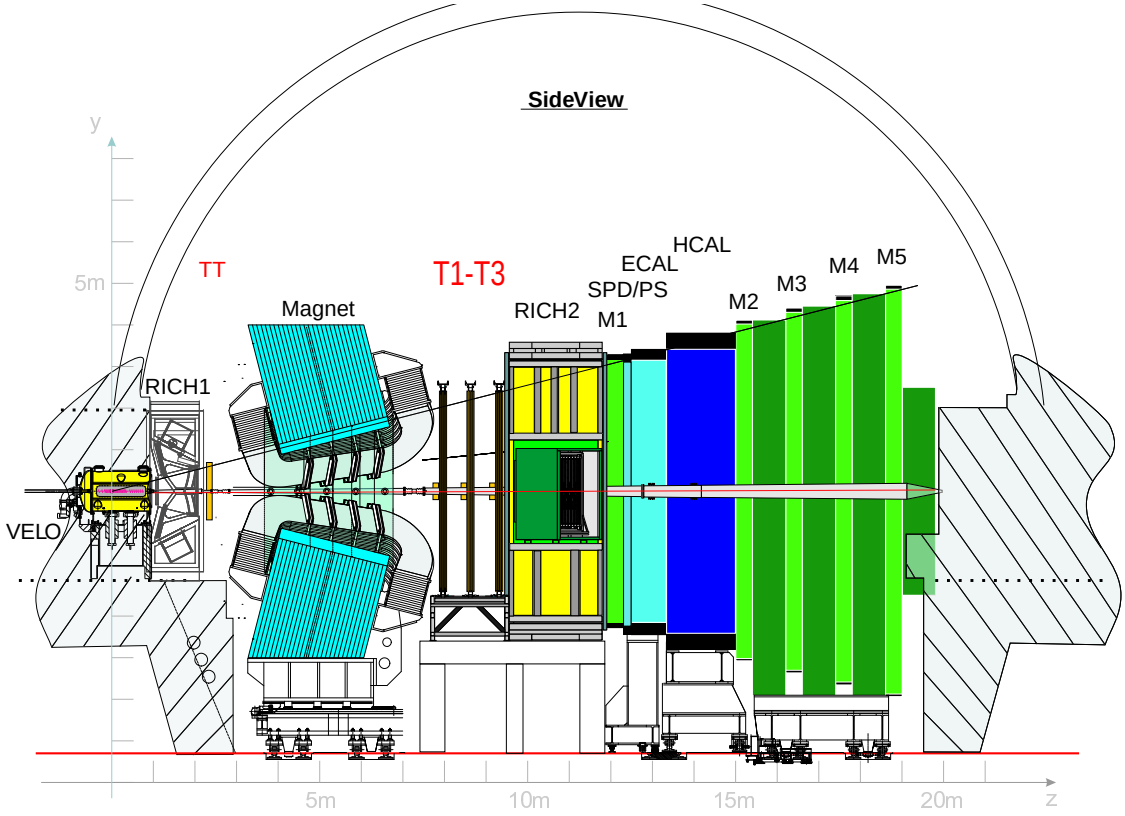


Figure 0.2.: LHCb

Early measurements with laser and test beams show large charge sharing and charge loss in between readout strips of the TT modules. On the other hand measurements in LHCb show small charge sharing and no charge loss in between strips. Because charge sharing improves spatial resolution and charge loss can reduce efficiency it is important to understand this discrepancy. The goal of this thesis was to understand charge sharing under different operation parameters, specially the charge sharing zone width and the charge loss.

Outline: Section 1 is an overview of the test stand and the silicon module with its properties. Section 2 contains a description of the data acquisition and preparation before the measurements. The focus measurements are described in section 3 and the charge sharing measurements in section 4. The thesis concludes with a short conclusion and outlook for further measurements. The appendix contains the test stand manual and the analysis code.

1. Test Stand Overview

Based on two bachelor theses [7, 14] only a short overview describing those parts of the test stand most relevant for the thesis is given here.
The three important parts for the thesis are:

- the Silicon Module
- the Laser System
- the Movable Structure holding the collimator focussing the laser beam onto the module

These three parts are inside a light tight box which holds the heatsink with the silicon module (see figure 1.9). Further the test stand consists of the cooling device, which controls the temperature inside the box during the measurements, and the control system, which provides the trigger for the laser signal and the read out, the bias voltage and the data acquisition.

1.1. Silicon Modul

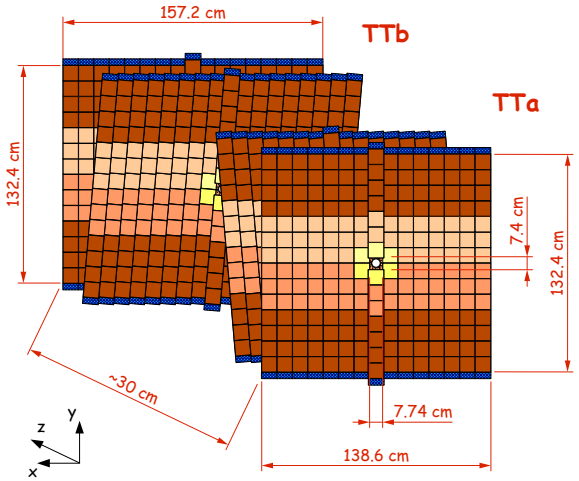


Figure 1.1.: TT layout

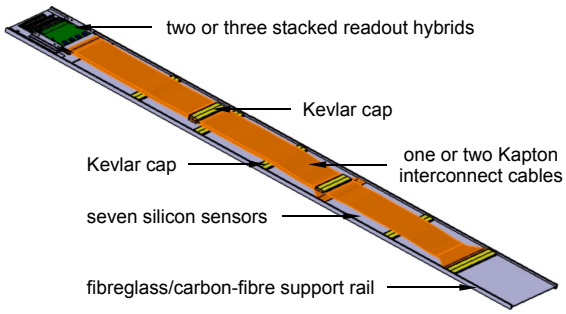


Figure 1.2.: TT module

Each half-module consists of seven silicon sensors (tab 1.1) with a stack of two or three readout hybrids at the end, that consist of 4 Beetle chips each. Every Beetle chip has

Table 1.1.: Dimensions of one silicon sensor [1]

Width	9.64 mm
Length	9.44 mm
Thickness	500μm
Strip pitch	183μm
Number of strips	512
p^+ implant width	46μm
Width metal strip	58μm
Full depletion	160 - 240 V

Table 1.2.: Beetle Chip Parameters [17]

Parameter	Value
Ipre	600 μA
Vfp	0 mV
Vfs	400 mV

four ports, which transfer data from 32 strips to a service box. The beetle chip stores pulse height information in an analogue pipeline memory (Table 1.2 shows the beetle parameters which influence the pulse shape, for further information see [17]). This information will be converted with analogue-to-digital converter and processed further. More information can be found in [23]. The modules installed near the beam-pipe in LHCb have four silicon sensors close to the readout-chips (L-sector), as well as a two sensor region (M-sector) and a one sensor region (K-sector) at the end -"4-2-1 type". The L- and M-sector are connected with Kapton interconnect cables of 38 cm, respectively 57 cm length. All other modules consist of a L-sector with four silicon sensors and a M-sector with three sensors -"4-3 type". A thin Kapton cable is connected with the backplane to provide bias voltage to the sensors possible [20]. During the thesis only sector M of a 4-3 module has used.

1.1.1. Function of Silicon Sensors

The detection of charged particles in the TT is based on a single-side silicon micro-strip detector module. Such a module is more or less a reverse-biased diode. The figure 1.3 shows a schematic profile of a strip detector. The module consists of a n-doped silicon bulk and p^+-doped strips so that the depletion zone extends into the bulk. To avoid that the leakage current from the detector flow through the Beetle chip, the modules have a SiO_2 layer between p^+ and readout metal strip. For the operation the module is reverse biased, where a positive bias voltage is applied at the backplane. The bias voltage has been chosen to increase the depletion zone to full depletion, which means it extends over the full thickness of the sensor. The electric potential is at maximum close to the backplane and zero by the p^+ strips. To avoid loss in charge collection, the detectors are over-biased (higher voltage then full depletion voltage). For the TT silicon detector modules the bias-voltage used is 300 V.

If an ionizing particle goes through the detector it produces electron-hole pairs along its track. The electric field generate of by the bias-voltage separates the electron-hole pair before it annihilates. The electrons drift to the anode (backplane) and the holes to the cathode (p^+ strip). The drifting charges produce an electric current on one or more specific strips, which is then processed by the readout-chips. The sharing of the current

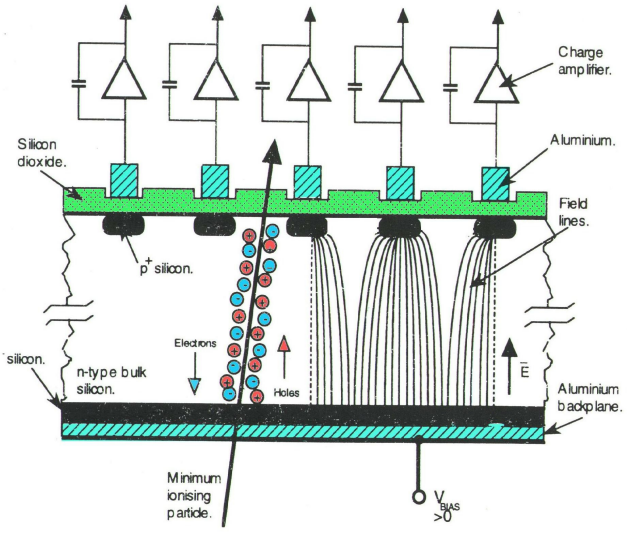


Figure 1.3.: Scheme of the silicon sensor with description of the electric field

to more then one strip specifically to two, is described in chapter 4 [13, 5].

For the operation of a silicon strip detector is it important to know the signal behavior of a minimal ionizing particle - MIP. By definition MIP's have an energy loss rate near the minimum and are the worst case to measure. For the TT detector a MIP should have a signal/noise ratio of 12 to 14.[19]. The charge sharing measurements described in this thesis (see chapter 4) were performed at signal/noise ratio around 10.

1.2. The Laser System

The test stand at the Physics Institute at the University of Zurich was extended for specific position measurements and tests on the silicon modules [7]. A Nd:YAG-laser[1] with spectral maximum at $\lambda = 1066$ nm ($\lambda_{min} = 1055$ nm, $\lambda_{max} = 1075$ nm) was used to mimic ionizing particles traversing the detector [18]. The band gap in silicon is approximately ≈ 1.11eV but the average energy to create electron hole pair is 3.6 eV, which is the energy of minimum three single laser-photons. Charges are therefore produced by multi-photon events. For more information see [13, p. 19].
The laser system consists of the pulsed laser diode, a passive attenuator, a collimator and a power supply. The trigger signal from the data acquisition would be converted by NIM-modules and then go to the laser diode which produced the laser pulse. The laser pulse goes over optical fibers to the attenuator and then to the collimator which focuses the laser beam at a focus distance of $\approx$ 12mm. To synchronize the readout and the laser beam an offset could be set between the laser trigger signal and the read out signal. The measurement of the delay time is described in section 2.2.

[1]Neodymium-doped yttrium aluminium garnet laser

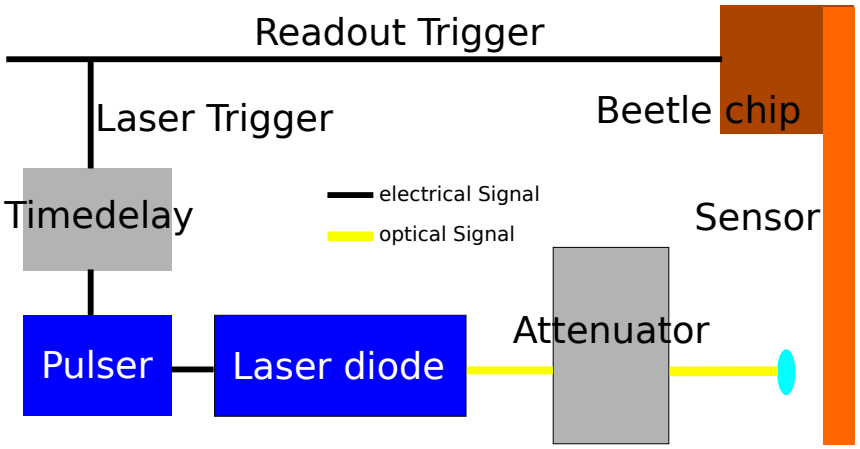


Figure 1.4.: Schematical overview of the laser system

1.3. The Movable Structure

The collimator from the laser is connected to a movable structure. This consists of two step-motors to move the collimator in x- and z-direction, where x means the direction the sensor and perpendicular to the readout strips and z means the distance between collimator and sensor (see Fig 1.5). The precision of the move in x-direction is 5μm and in z-direction 2.5μm [7]. Furthermore all x coordinates are only relative values to an arbitrary start point. Figure 1.10 shows the structure which is fixed with a block of lead to have better mechanical stability. The whole structure is placed by hand as good as possible perpendicular to the sensor. Beside the movable structure are also the attenuator and the laser diode with the glass fibers. All these things are positioned in a way that the fibers are not bent too much and the movable structure can move freely. Due to the possibility that geometrical charge sharing could be produced by

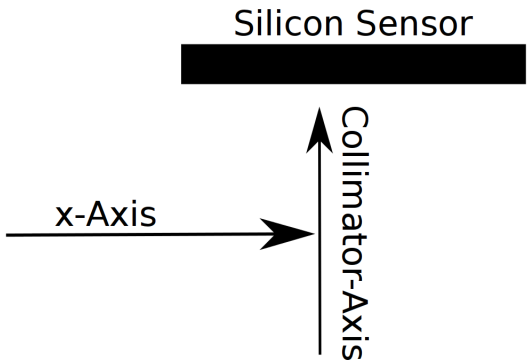


Figure 1.5.: x- and Collimator-axis perpendicular,where the x coordinates are only relative values to some start point

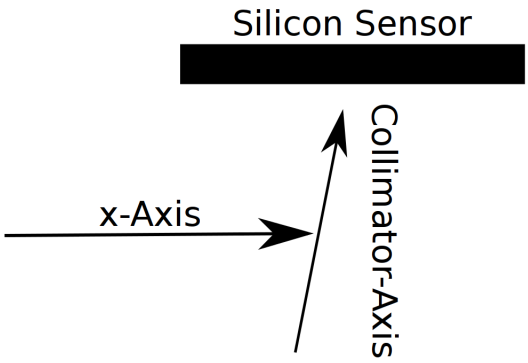


Figure 1.6.: x-axis perpendicular

non perpendicularity of the laser, the correct alignment of the laser must be proven.

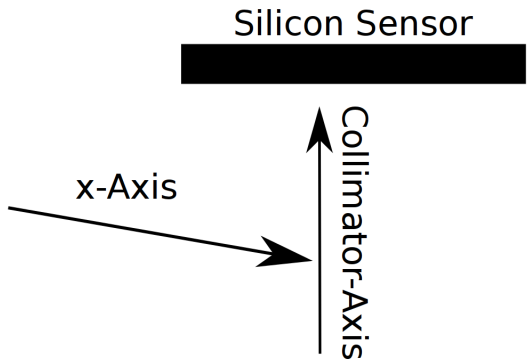


Figure 1.7.: Collimator axis perpendicular

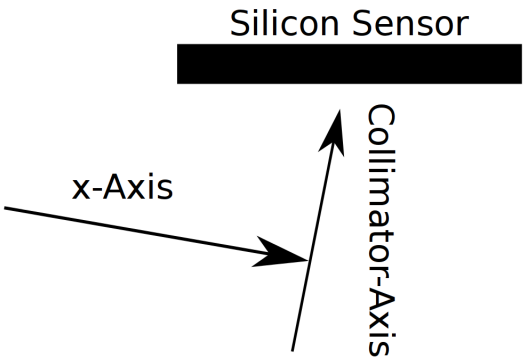


Figure 1.8.: no-axis perpendicular

The perpendicularity between the two axis of the movable structure was assumed to be guaranteed by the mechanical precision of the construction. The alignment of the collimator to the axis of the movable structure result in a new definition of the z-axis for the laser where the x-axis is the same as for the movable structure. Different possibilities for a displacement of the collimator exist; in the case of the test stand it has mainly three possibilities (see figure 1.5,1.6 and 1.7), which change the position of the focus point relative to the sensor. The effect for the measurements of these possibilities is described in chapter 3. Figures 1.9, 1.10 and 1.11 show the movable structure in detail and an overview of the hole box.

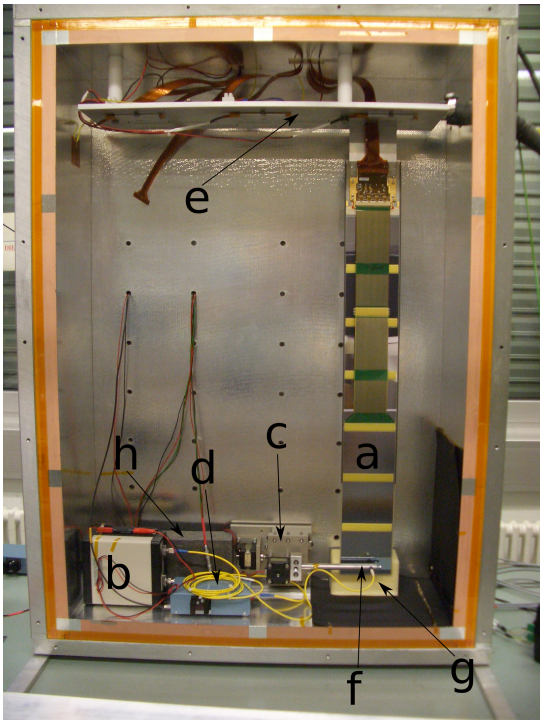


Figure 1.9.: Front view of the test stand box

a) Detector module

b) Passive attenuator

c) Movable structure

d) Laser amplifier

e) Heatsink (conduct the heat and cooling the electronics)

f) x-Axis with Collimator

g) Foam block (to stabilize the silicon sensor)

h) Lead block (to stabilize the movable structure)

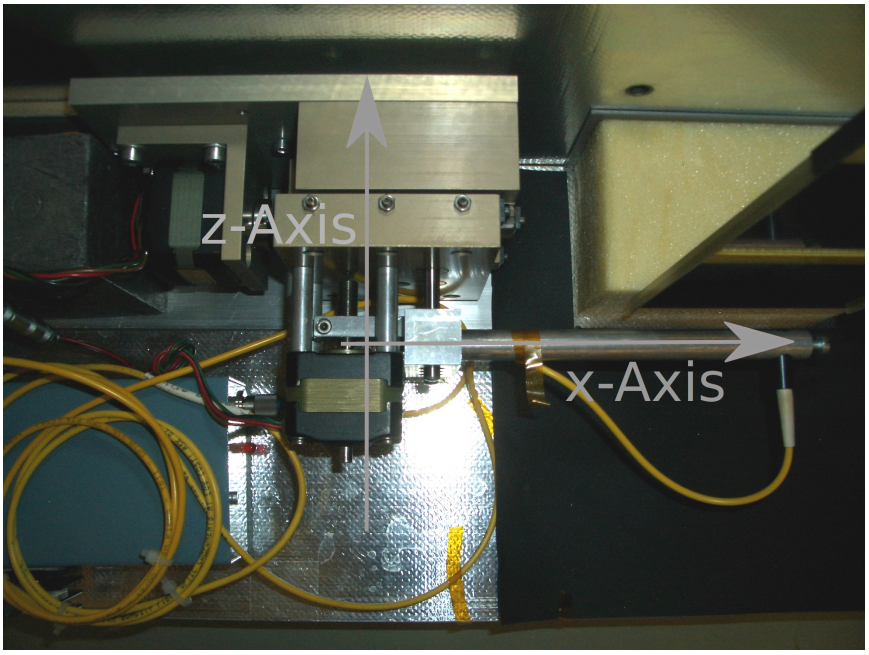


Figure 1.10.: View on the xz plane of the movable structure

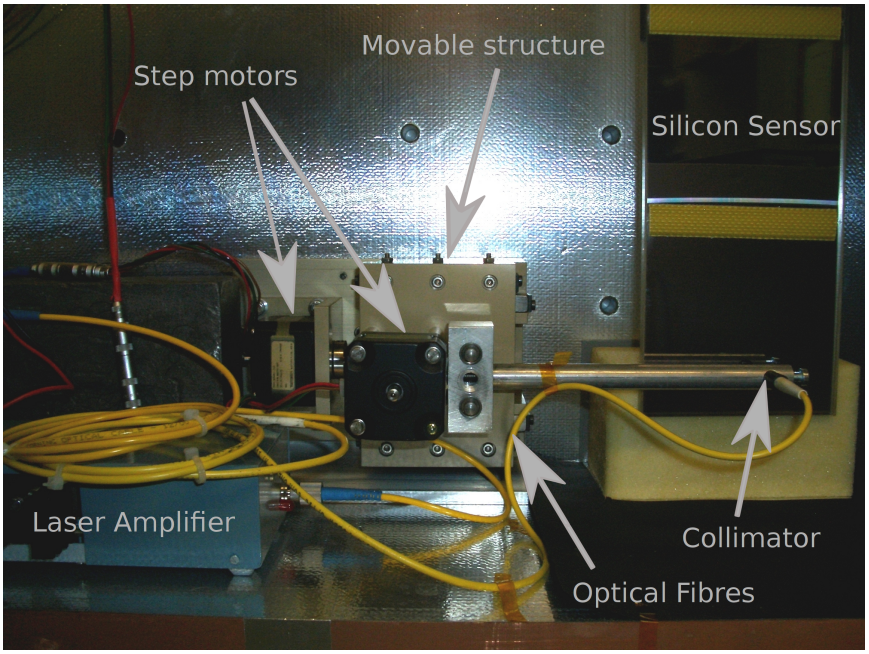


Figure 1.11.: Front view of the movable structure

1.4. Optical Properties of Silicon Sensors

The laser photons should not be absorbed completely at a distance less then 500 μm in silicon to ensure the production of charges over the whole thickness of the sensor. For the wavelength of the laser (1066 nm), the average absorption length in silicon is approximately 1 mm. This guarantees the production of charges over the full thickness of the sensor.

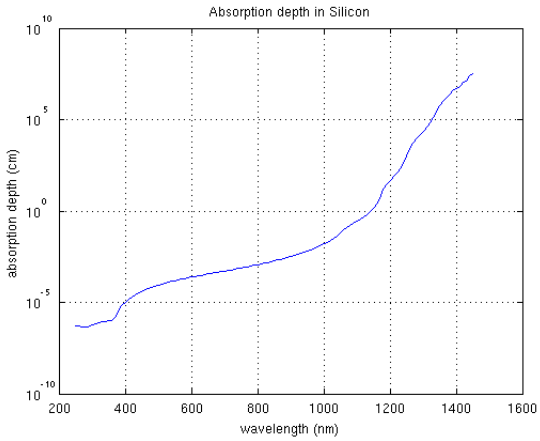


Figure 1.12.: Optical properties of silicon: absorption depth

Another important detail is that the refraction index for silicon at $\lambda = 1066$ nm is n = 3.55. Due to that, a displacement of the focus point of 1 mm in silicon corresponds to 3.55 mm displacement in air, as illustrated in figure 1.13. This was considered in chapter 4 for the measurements with a focus point inside the silicon bulk [12].

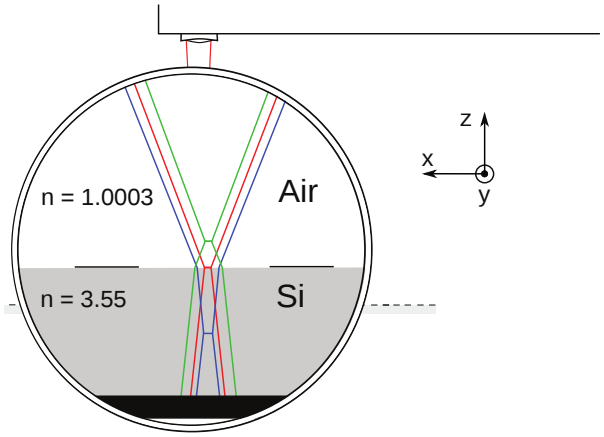


Figure 1.13.: The picture shows the light rays represented by their waist, it represent three different z-position. The difference between focus point and silicon surface for the green line is the same as for the blue line. Its clearly visible that for the same movement as in air it has to be done a smaller movement in silicon. From [7]

2. Preparation

This section describes the procedure for the data acquisition. Further it describes the measurements of pulse shape and noise which should be done before the experiments. For the further measurements, its important to know that the read out performs at the right time and about the stability of the noise. All the described files are in the appendix.

2.1. Data Acquisition

The data acquisition of the TT at LHCb is based on the analysis program *GetData*. It decodes the raw-data and processes them and provides for example pedestal calculation and common mode subtraction. The software structure is difficult and has many processing features which aren't required in this thesis. Therefore, the data processing was performed by a Matlab file *GetData.m*. At the end the only future used of the *GetData* program was dumping the raw ADC-values to n-tuples (matrices with row=event and column=strip) and to make the appropriate histograms. The root program *plotTuple2.C* copies the ADC-values of a specific strip and its neighbors to a tab separated ASCII-file, which can be imported into Matlab.

The Matlab file *GetData.m* performs several steps:

pedestal calculation: The pedestal is the mean signal height without laser signal. It is different for every strip and should be constant over time. Therefore it can be determined with the mean of a set of events. Runs with 60000 events without laser signal were taken before and after each measurement. The program takes the mean (corresponds to the pedestal) and the rms (corresponds to the noise) from it, but as seen on the figure 2.1 the histogram of the runs without laser has some outliers outside 3σ from the mean. It was decided to remove these outliers for the calculation of the noise and the pedestal. The m-File *Smean.m* does this in five iteration steps and stores the mean as p_i and the rms as σ.

rms calculation: The program calculates the standard deviation σ_i of the ADC-values per strip over all events after outlier removal:

$$\sigma_i = \sqrt{\frac{\sum_{j=1}^{n}(x_j^i - \mu^i)^2}{n}} \tag{2.1}$$

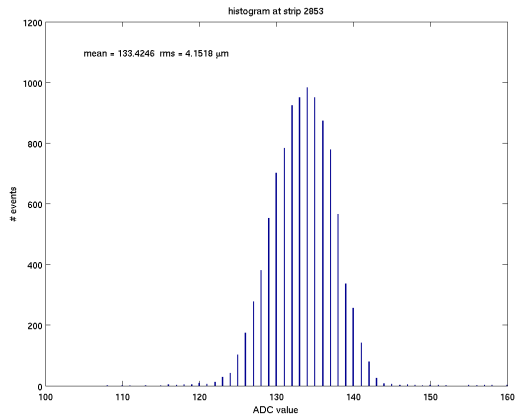


Figure 2.1.: Histogram without Smean.m

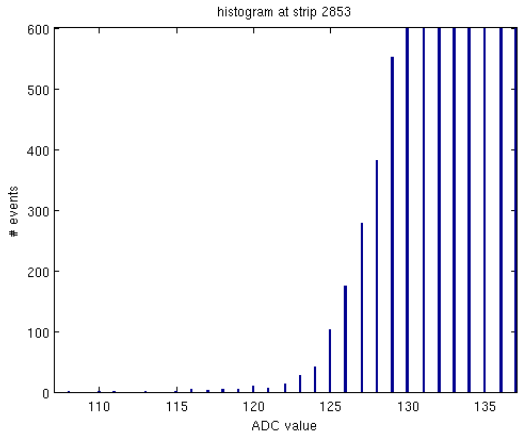


Figure 2.2.: Histogram without Smean.m with zoom on the left side

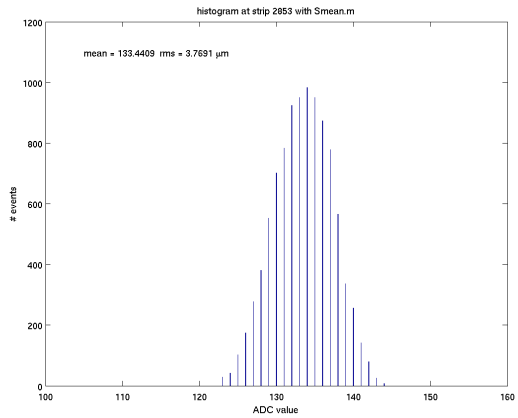


Figure 2.3.: Histogram with Smean.m

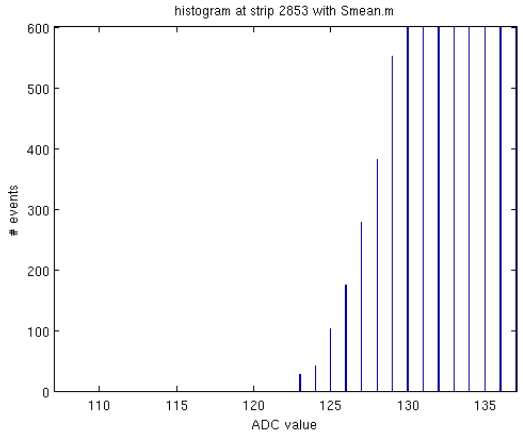


Figure 2.4.: Histogram with Smean.m with zoom on the left side

where x_j^i is the ADC-value of the i^{th} strip and j^{th} event and μ^j the mean of the i^{th} strip. The rms (noise) differs form strip to strip due to gain variations in the Beetle chip. The signal/noise ratio was taken as signal amplitude to get rid of this dependence on the amplifier gain.

data calculation: The program subtracts the pedestal p_i from the mean value of signal-measurements with laser pulse (also calculated using *Smean.m*) and makes the error propagation. At the end it stores the data values and the errors in a file for further calculations. Next, *GetData.m* plots the data values as in a 3D plot as a function of the x- and z-position fo the collimator as shown in figure 2.5

common mode subtraction: Another contribution to the signal is the common mode. Each beetle chip has four output ports and the common mode is for each event different

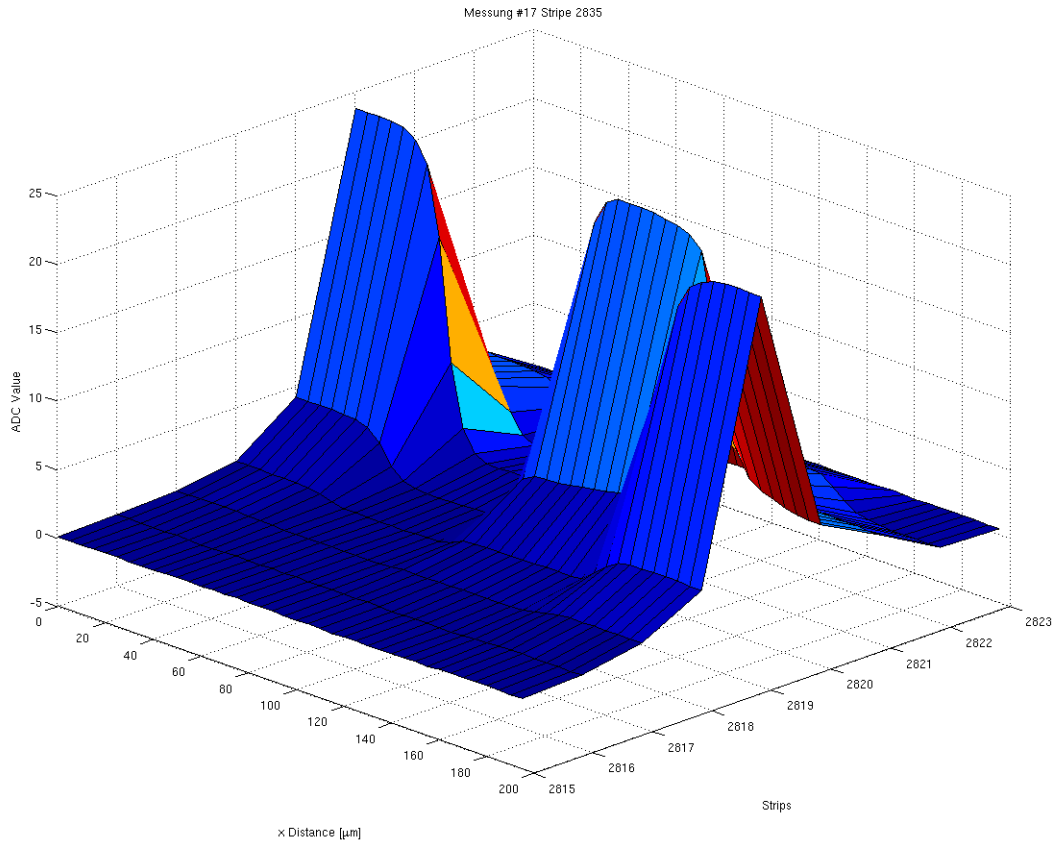


Figure 2.5.: On the figure it can be seen a region where the signal goes to zero. This is the case if the laser move over the aluminium strip. Around $140\mu m$ it can be seen that the signal change to the next strip, this is the charge-sharing between those two strips. More in chapter 3.

but for one port the same. The common mode can for example be calculated with a fit of a straight line to all signals from one port. Common mode makes the spread of the signal peak greater and this implies a bigger noise on the other hand, a common mode subtraction could cause unwanted biases in the measurement of the signal amplitude. The most interesting part for the thesis is the mean of the signal distribution and this does not depend on the spread. Based on these considerations was no common mode subtraction implemented in the Matlab analysis.

Each measurement was taken 10000 laser pulses unless otherwise mentioned. The axes was defined as in section 1.3.

2.1.1. Noise Behavior on Time

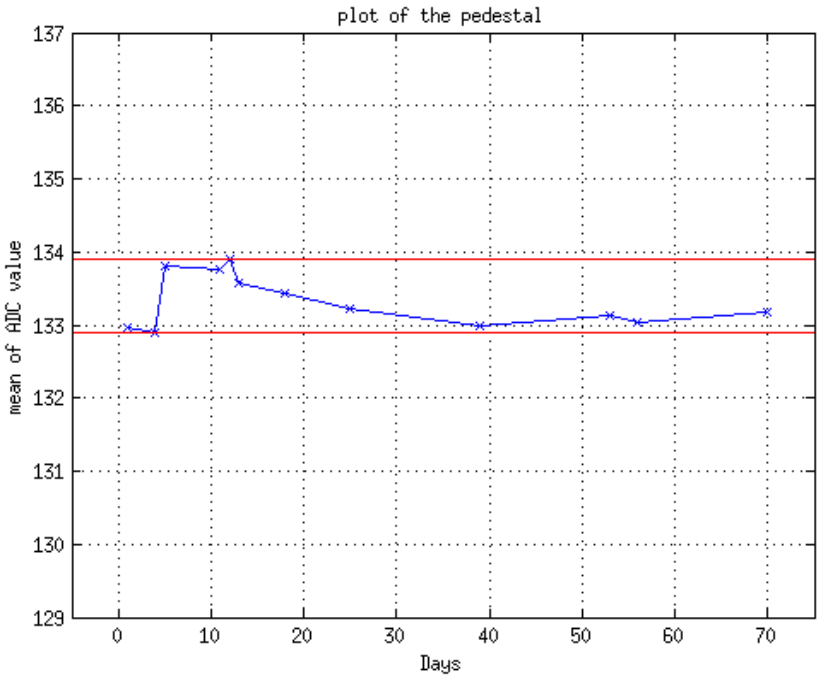


Figure 2.6.: Plot of the mean values from the pedestal-noise measurements

One of the important questions on the behavior of the silicon detector is its stability, especially the stability of the noise and the pedestal. As mentioned before both are measured before each measurement. For each pedestal-noise measurement were 6 runs taken without laser (one run consists of 10000 laser pulses). The noise and pedestal values are shown as function of time (figure 2.6 and 2.7.
The noise is stable over the 70 days, but the pedestal varies in a range of 1 ADC value (132.9 to 133.9). These variations do not affect our measurement and were not investigated further. They could be caused by e.g. temperature variations. This gives a nice noise behavior with small fluctuation on a long measurement period and verify the procedure of the applied noise measurement.

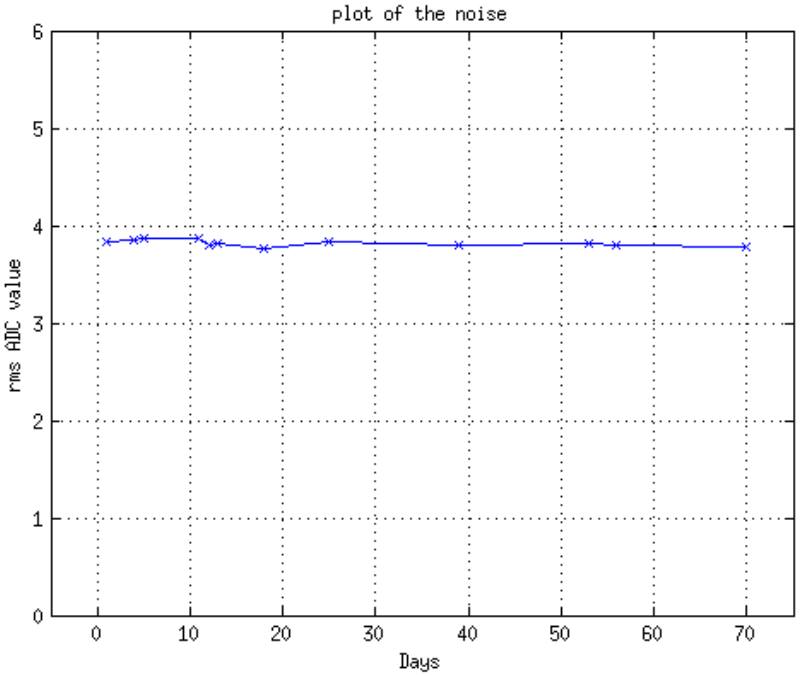


Figure 2.7.: Plot of the rms values from the pedestal-noise measurements

2.2. Signal Pulse

In a previous thesis [7] the time delay between laser signal and read out was defined. The pulse shape (measured singal amplitude as a function of the time difference between the two signals which was measured for this is shown in figure 2.8. The best time delay is at the maximum of the pulse shape. For the measurements of this thesis, the pulse shape was measured only around the maximum to verify that the beetle chip still reads out at the right time. All measurements were taken at the strip 2853 after a short focussing as described in section 3.2. The attenuator in the laser system defines the intensity of the laser pulse, where a high dB means small intensity (high attenuation) and vice versa. More information can be found in [21]. The signal pulse measurement was performed at an attenuation of 24.5dB resulting in a signal/noise ratio ≈ 10. All focus measurements were taken at 20dB. The charge-sharing experiments were taken at different values of attenuation to test a possible effect of the intensity on the charge sharing. The result implies that it made no difference at which dB the measurements were taken (see section 4.1.1).
The time delay was adjusted to the optimum value found in the previous thesis before, shown as the zero point in figure 2.9. Only a few measurements around this point were needed to verify the maximum. As it can be seen in figure 2.9, the time delay was correct.

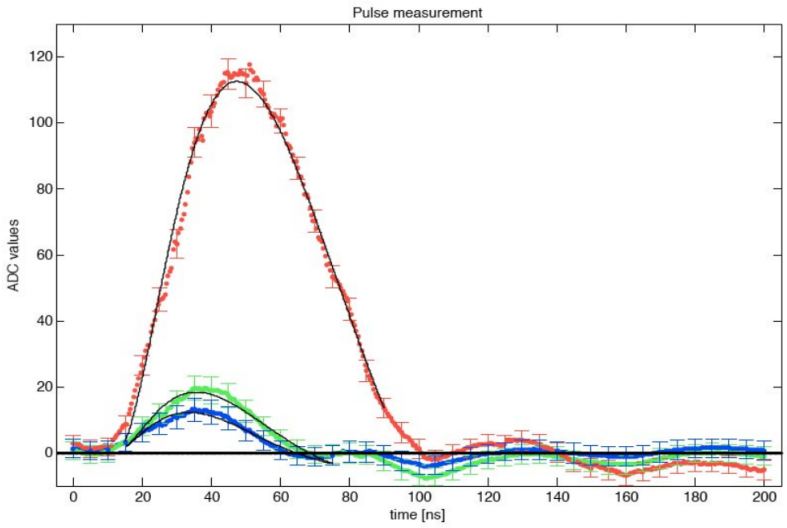


Figure 2.8.: The figure shows the pulses measured in 0.5 ns intervals inclusive their fit functions describing the main pulse. The red plot represents Channel 2853 while the blue one is 2852 and the green one 2854. Based on [7, page 24]

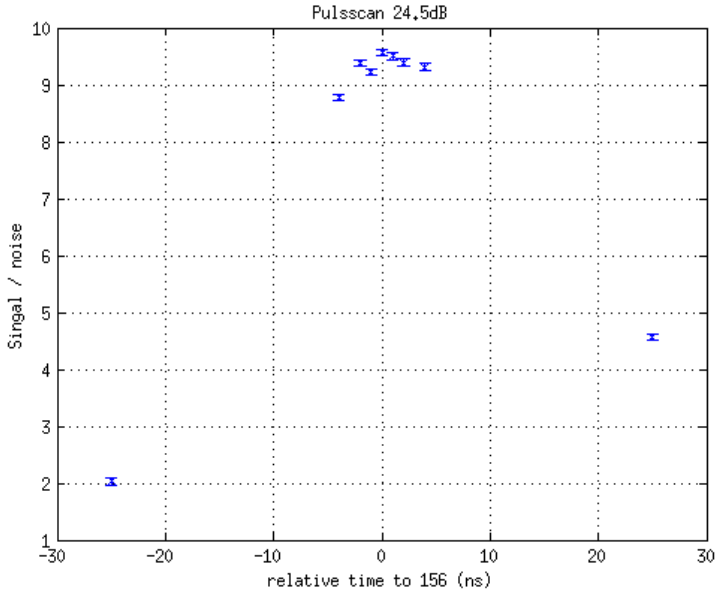


Figure 2.9.: Pulsescan at 24.5dB.

3. Laser Alignment

One of the most important parts of this thesis was to be sure that the axes of movable structure and the laser collimator are perpendicular to the silicon sensor. This would minimize charge sharing effects created by geometrical properties as mentioned before (section 1.3).
This section will describe first how the focus position can be determined and how this can be used to check the alignment. Further it contains the result of the alignment and an estimation on the accuracy of the position measurement.

3.1. Example Measurement

A measurement, as figure 3.1 shows, consists of a scan over 40 steps of 5μm each along the x axis. A region where ADC-values are small is well visible. This is the area where the laser is behind the aluminium strip. It has a width of about 50-55 μm, compatible with the width of the aluminium strip (58 μm). Further the figure shows that the strips left and right from the main signal strip also see a small signal. This small signal is a result of the crosstalk between neighboring strips. There are two types of crosstalk, one is due to the capacitive coupling between the neighboring readout strips on the sensor, which is symmetric due to the geometry of the sensor. The second one is caused due to design of the front end electronic on the Beetle chip, which cause an asymmetric crosstalk. More on crosstalk in section 4.1 and 4.2. Further away from the edge of the strip, the charge sharing begins and the jump can be seen from one strip to the next. More about charge sharing follows in chapter 4.

3.2. Identification of the Focus Position

The basic idea of the alignment is that the focus point of the laser beam, produced by the collimator, should be on the silicon surface. In the following this is named *best focus*. Two methods are possible to accomplish this, one uses the steepness of the slope at the edges of the area, where the laser beam is behind the aluminium strip; the other method uses the width of the area where the laser beam is covered by the aluminium strip.

Focus on aluminium strip: Because the laser beam can not penetrate the aluminium strip, the difference of the ADC-values between measurements on the strip and besides the strip can be used as an indicator for the focus. While the laser moves across the edge of the strip, the measured signal amplitude increases as the beam spot moves into or out

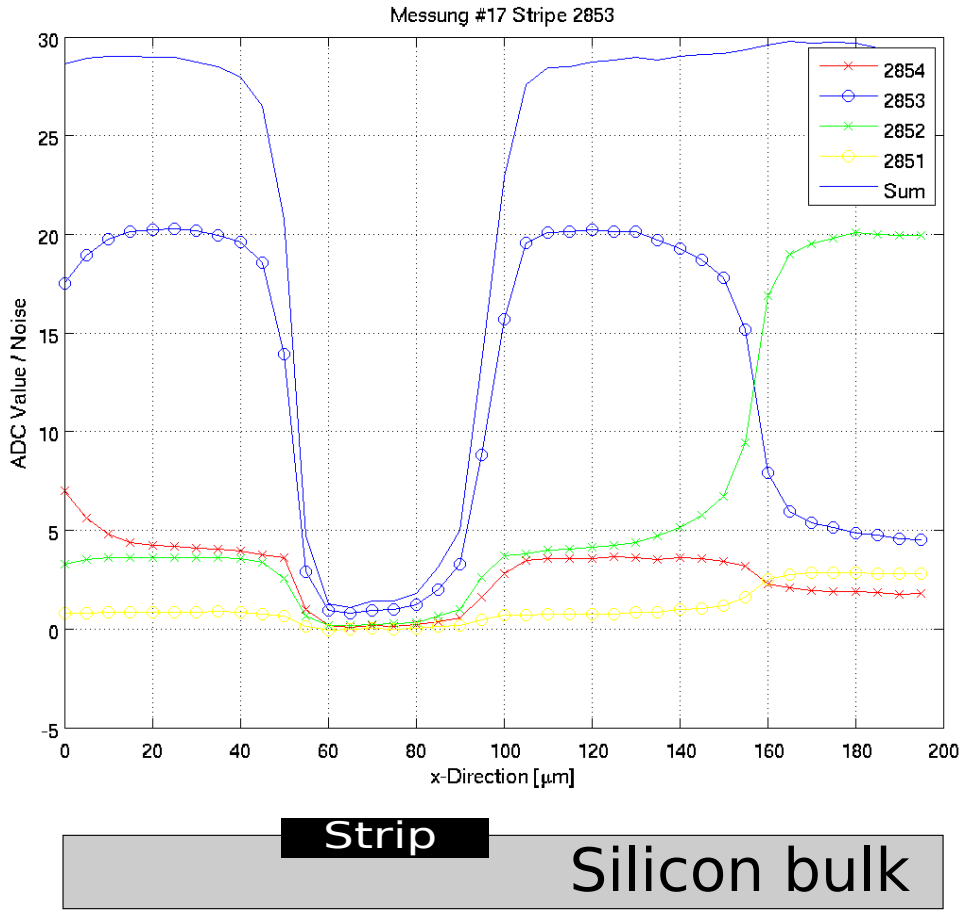


Figure 3.1.: scan over 40 steps

the shadow of the aluminium strip. The best focus occurs when the slope of the signal at the corner of the region is steep. Measurements were taken at different z-positions and the slopes for each z position was determined to find the best focus position (see appendix B for the code and the next section for the procedure).

Fast focusing: The signal amplitudes can be displayed on a oscilloscope via a Digital to Analogue Converter. This allows a fast focusing; it is necessary to make a range of measurements along the x axis with different focus-distances. The z-position for which the width of the x-range in which the signal is small or invisible is largest, is the best focus position.

The second method is much faster then the first one and the obtained results are comparable. For the measurements after the laser alignment, this method would be used to focus.

3.2.1. Analysis

This section describes the extraction of the focus position from the data. A focus measurement consists of 25 scans over 200 μm (40 steps) in the x-direction. The scans were taken at z-positions from -1000 μm to 1000 μm with smaller steps around zero. The zero position was defined by *fast focusing*. Further scans were taken on two different strips with a distance of 6.05 mm in between the two strips to define the angle between x-axis of the laser and the sensor surface, more in the next section.
As mentioned before the slope at the edges of the aluminium strips was used to find the best focus. The signal amplitude at edges can be described with the following formulae:

$$f(x) = a_L \cdot \left(1 - erf\left(\frac{x - \mu_L}{\sqrt{2}\sigma_L}\right)\right) + b_L(leftedge) \tag{3.1}$$

$$f(x) = a_R \cdot erf\left(\frac{x - \mu_R}{\sqrt{2}\sigma_R}\right) + b_R(rightedge) \tag{3.2}$$

The μ defines the position of the edge in x-direction, while a smaller width σ stands for a better focusing. The variables a and b depend only on the laser intensity and on the pedestal. They are constant over different z-positions as expected. The fit was performed at each z-position, which gives the focus size and edge position as a function of z.
As an example two of these fits are shown in fig. 3.2 and 3.3. For all fits there was also a χ^2-test performed and, where needed, a scale correction on the errors of the fitted parameters (all errors treat identical with factor S = $\sqrt{\frac{\chi^2}{ndf}}$, ndf = numbers of degree of freedom).

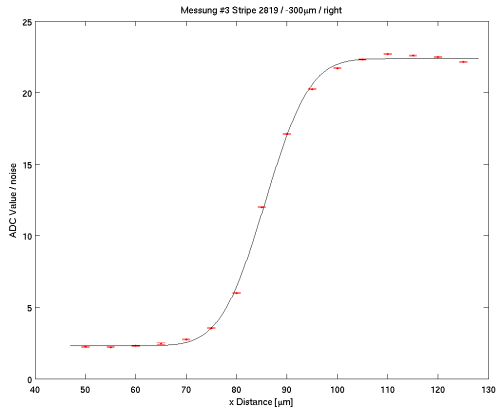


Figure 3.2.: fit of the right edge, at focus distance $-300\mu m$. The rising is much slower the in figure 3.3.

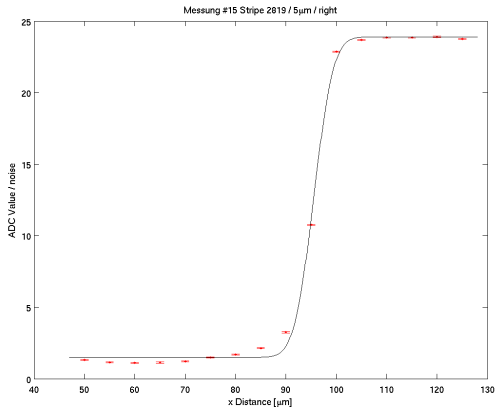


Figure 3.3.: fit of the right ege at focus distance $5\mu m$. The rising is much faster, it follows that the focus is better then in figure 3.2

Extraction of the Focus Position

In figure 3.4, the fitted focus size σ from each run is plotted versus the z-position. As mentioned in figure 1.13 it can be described with the formula:

$$\sigma(z) = s_1|z - s_0| + s_2 \tag{3.3}$$

where s_0 describes the z-position of the focus point, s_1 stands for how fast it defocuses and the s_2 stands for the minimal focus size. Table 3.1 shows the measured values for the two strips.

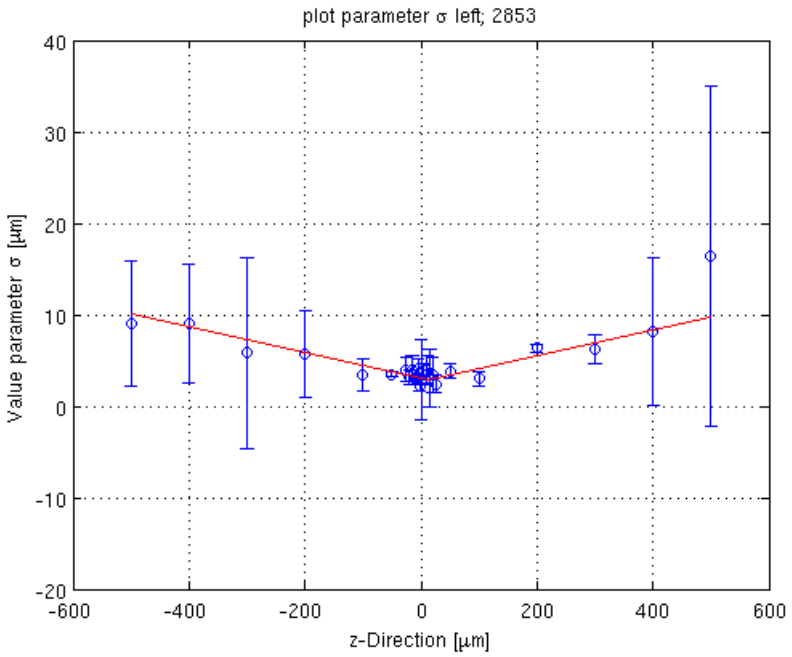


Figure 3.4.: plot of σ versus the z-position including the fit of form 3.3

Table 3.1.: Fitted parameters $\sigma(z)$

	2853		2819	
Parameter	right	left	right	left
s_1	$(1.9\pm0.2)10^{-2}\mu$m	$(1.3\pm0.2)10^{-2}\mu$m	$(1.3\pm0.2)10^{-2}\mu$m	$(1.5\pm0.2)10^{-2}\mu$m
s_2	3.2±0.2 μm	3.0±0.2 μm	3.7±0.2 μm	3.4±0.5 μm
s_0	-2.9±11.3 μm	-2.5±14.1 μm	3.4±12.2 μm	33.3±33.7 μm

Discussion: From the fits for both channels it can be seen that the minimal focus size σ is around 3.3μm, which means that 95% of the laser beam is inside of $4\sigma = 13.2\mu$m. Also it can be seen that the error on the focus position s_0 is large. These big errors come from the dispersion of the measurements around the focal point. The reason is that for scans with steep edges, only two or three measurement points fall inside the slipe, the fit isn't perfect and the result changes significantly if only one point is lower or higher.

3.3. Alignment Checks

As mentioned in section 1.3, there are three possibilities how the structure could be misaligned:

1. Both the x-axis of the movable structure and the collimator axis are not perpendicular (see figure 1.8). In this case, if the x position changes then the position of the focus point (best focus position) also changes. Further it follows from the nonperpendicularity of the collimator axis that if the z-position changes then the edge position would also vary due to the fact that the intersection between laser beam and surface would depend on the z position.

2. Only the x-axis is perpendicular. Then the best focus position should be equal for different x-positions, but for the edge-position μ changes with the z position as in 1.

3. Only the collimator axis is perpendicular, then the edge-position μ does not change as a function of the z position, but the position of the best-focus changes with different x-position.

To test the alignment of the structure, one can therefore use the measurement of the best-focus position on the surface $\sigma(x)$ as a function of x and the edge position $\mu(z)$ as a function of z. Both functions should be constant in the case of perfect alignment.

3.3.1. x-Axis and Sensor Surface

As mentioned in the previous section the smallest focus size and the position of this focus were defined for two readout strips at a distance of 6.05 mm from each other. From the difference in z for the best focus and the distance in between the two strips the angle between the x axis and the sensor surface was determined. The tangent of the angle is equal to the ratio of $\frac{\Delta z}{\Delta x}$. To determine the error on the angle, were taken the largest and smallest slope compatible with the errors on the positions of the best focus. The figures 3.5 and 3.6 show the plots. Table 3.2 shows the parameters of the fitted straight lines.

Table 3.2.: Fitted parameters $\sigma(x)$

parameter	right	left
m_0	$(1.0 \pm 3.9)10^{-3}$	$(5.9 \pm 7.8)10^{-3}$
m_1	$(-2.9 \pm 14.2)\mu$m	$(-2.5 \pm 14.0)\mu$m
angle [°] (atan(m_0))	0.06 ± 0.22	0.33 ± 0.44

Discussion: The angle between the x-axis and sensor surface is compatible with zero as can be seen in table 3.2.

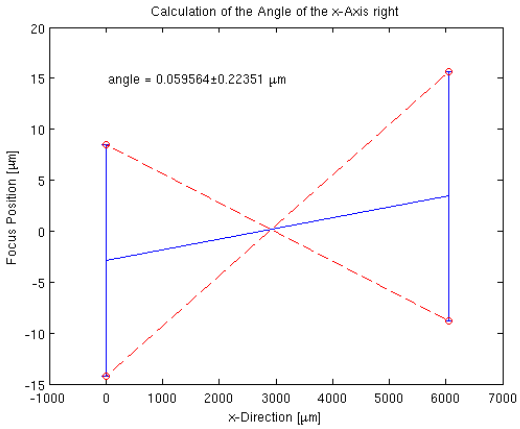


Figure 3.5.: Plot of the minimal focus size as function of x for the right edge. The red dashed lines are the minimal and maximal slope compatible with the errors.

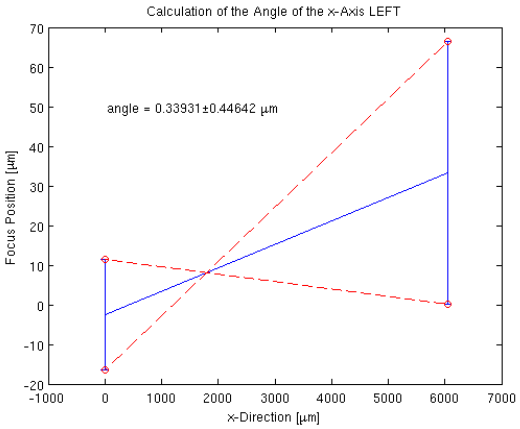


Figure 3.6.: Plot of the minimal focus size as function of x for the left edge. The red dashed lines are the minimal and maximal slope compatible with the errors.

3.3.2. Collimator Axis and the Normal of the Sensor Surface

The edge position μ should not depend on z if the collimator axis and the sensor surface are perpendicular. A straight line was fitted to the data (tab. 3.3 and figure 3.7) where $m_0 = \frac{\Delta\mu}{\Delta z}$:

$$\mu(z) = m_0 \cdot z + m_1 \tag{3.4}$$

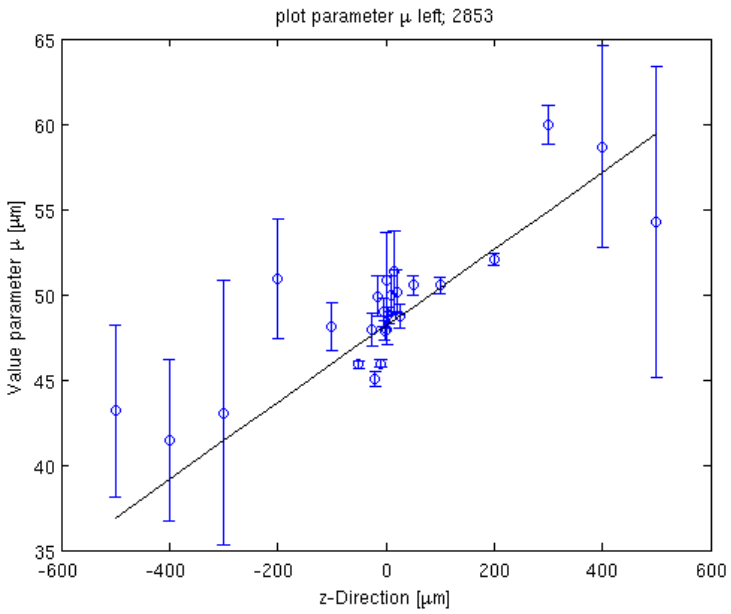


Figure 3.7.: plot of μ versus z-position with the fit of form 3.4

Table 3.3.: Fitted parameters $\mu(z)$

	2853		2819	
Parameter	right	left	right	left
m_0	$(2.6 \pm 1.0)10^{-2}$	$(2.2 \pm 0.8)10^{-2}$	$(2.1 \pm 0.4)10^{-2}$	$(1.8 \pm 0.6)10^{-2}$
m_1	94.7 ± 1.1 μm	48.3 ± 0.9 μm	93.4 ± 0.5 μm	36.6 ± 0.6 μm
$angle[^o]$	1.0 ± 0.5	0.9 ± 0.4	0.9 ± 0.2	0.8 ± 0.3

Discussion: The angle between the collimator axis and the normal-axis on sensor surface is approximately 0.9^o. From the negligible angle between the x-axis and z-axis of the movable structure follows that the measured angle is due to a misalignment of the collimator axis to the movable structure. But fortunately the measured angle is so small that it has no great influence on the charge sharing (see section 4.1.1).

In conclusion, the focus point was defined with respect to the strips on the sensor surface and the axis are perpendicular within the errors.

3.4. Large Measurement

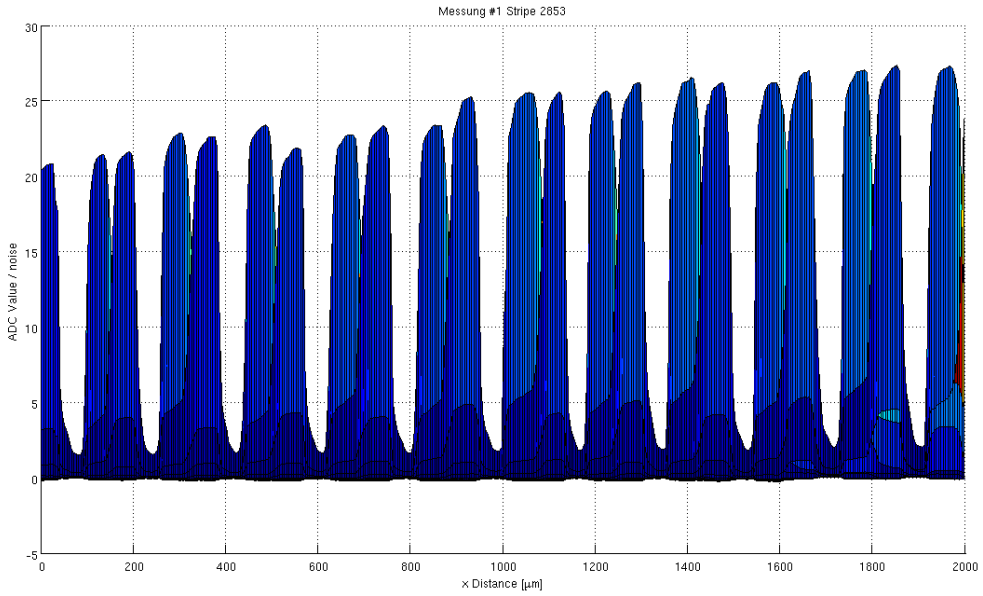


Figure 3.8.: Scan over 10 strips

To control if the step motor functions properly a scan over 10 strips was taken. The result of this measurement is shown in figure 3.8. It shows that the maximal amplitude grows for smaller strip number (highest strip on the left side). Figure 3.9 shows the plot of the noise as function of the strip number. As it can be seen the noise does not depend

on the strip number. This indicates that the trend observed in figure 3.8 is not due to gain variations in the readout chain. As this effect does not affect the studies performed in this thesis, it was not investigate further. Further, an analysis of the steps size of the step-motor was made. Each right side of the areas where the laser is behind a strip was fitted with the function 3.2. After this the obtained $\mu's$ were fitted as a linear function of x as in fig 3.10.

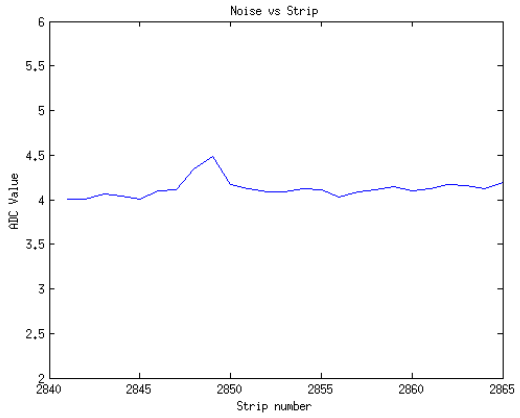


Figure 3.9.: Noise of the scan over 10 strips as function of the strip number

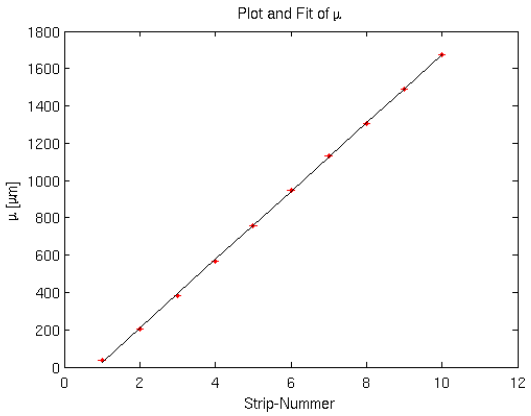


Figure 3.10.: Fit of μ as function of the strips

It fit gives for the slope $(182.8 \pm 0.8)\mu$m, which agrees well with the strip pitch of 183μm (tab. 1.1). Further the variation of the slope gives an estimation for the error per step. Figure 3.11 shows the histogram of the calculated difference between left and right edge of the areas with small signal. This difference gives the width of the measured aluminium strip. The weighted mean of the width is $54.7 \pm 1.3\mu$m, which is compatible with the aluminium strip width of 58μm.

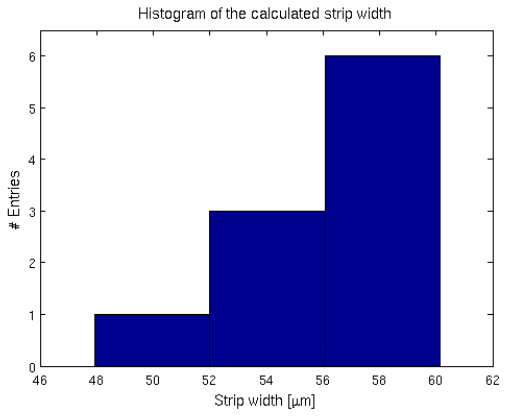


Figure 3.11.: Histogram of the aluminium strip width calculated for 10 different strips from the difference between right and left edge.

4. Charge sharing

Charges (electron-hole pairs) are produced by ionizing particles crossing the silicon or by laser beam (see section 1.4). The holes drift to the p^+ implant and the electrons to the back plane (n^+ implant). The p^+n diode is segmented and an electric field exists between each of these p^+ strips and the back plane as shown in figure 1.3. The produced holes diffuse during the drift. Because of this effect the holes can be collected on either the left strip or the right strip if the particle/laser beam cross the central region in between the two strips resulting in charge sharing. Charge sharing could also be introduced by geometrical properties, if the beam (laser or particle) is not perpendicular to the silicon bulk. This results in the production of charges at different positions in between the strips which gives additional charge sharing.

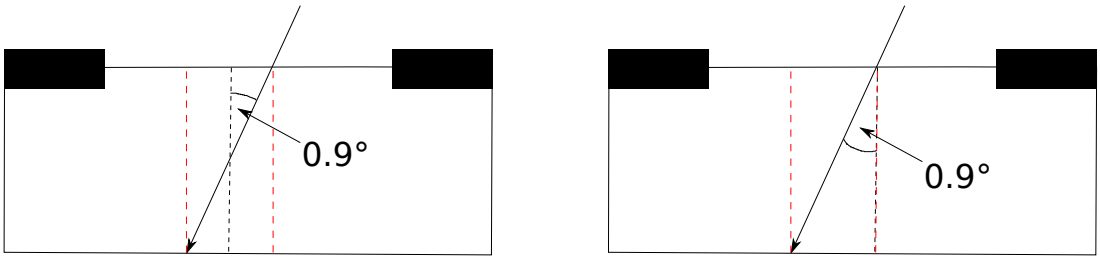


Figure 4.1.: *left figure* - It shows the laser beam through the bulk for the focus position in the middle of the bulk. Because of the symmetry of the induced charge sharing zone (area in between red dashed lines) should be the charge sharing symmetrical and as mentioned in the text is the charge sharing zone width biased. *right figure* - The focus position is on the surface. As mentioned in the text induce this an asymmetry for the charge sharing zone and as for the left figure case the charge sharing zone width is biased.

As shown in chapter 3 the laser beam has an angle $\approx 0.9^o$ to the normal of the silicon sensor, which over the thicknes of the silicon sensor of 500μm results in a zone with a width of 7.8μm (see figure 4.1). For the assumption that most charges are produced close to the focus point, it follows that if the focus point is on the surface of the sensor then the charge sharing should be asymmetrical. That means that if the laser is positioned exactly in the center in between the two strips the sharing zone is not in the middle between the strips. If the focus position is in the middle of the bulk, the charge sharing is symmetrical in any case. Therefore the measurement of the charge sharing asymmetry as a function of the z-position of the focus could give an indication of the influence from geometrical properties on the charge sharing.
The zero position for the z-axis (focus point on the surface) was defined in section 3 with the focus measurements. The position in the middle of the bulk is 250μm from the

zero point into the silicon (250μm/3.55 = 70.4μm for the step motors, for explanation of the factor 3.55 see section 1.4).
In previous measurements with a laser test stand and in testbeams at CERN, it was observed that some charge loss occured in the central region in between two strips. Because not all electric field lines go to the p^+ implants, it could be that charges drift to the SiO_2 layer and are captured. As a result, the sum over four strips show a dip in the center of the charge sharing zone. As figure 4.2 shows the charge loss observed in those measurements not a small effect. On the other side, measurements at LHCb have not observed any charge loss.

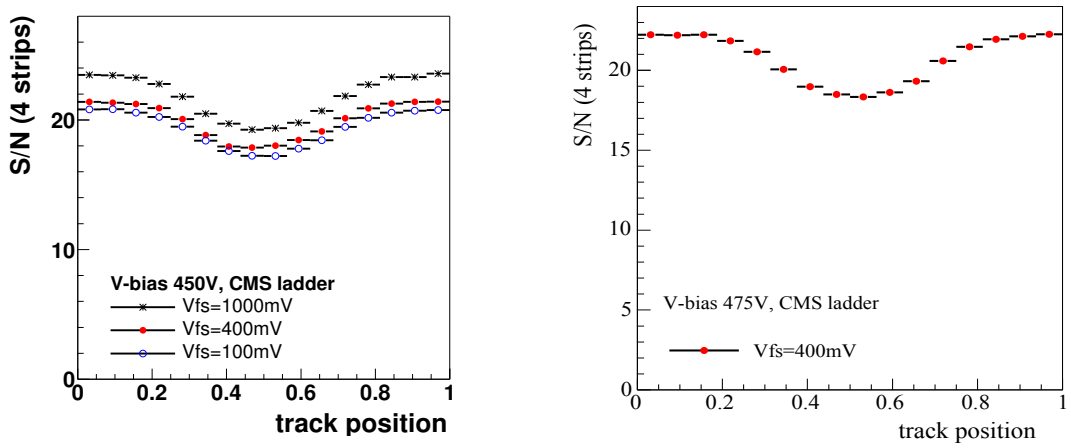


Figure 4.2.: The S/N was determined from the summed charge on four strips. In both pictures was V_{fs} set to 400 mV.*left figure* - It shows a measurement of the summed charge at a 120 GeV/c π^- beam at the CERN X7 test facility in 2003 [8]. *right figure* - The figure shows an equivalent measurement at the same test facility in 2004 [9].

This chapter describe the measurements to study this charge loss and the determination of the charge sharing width.

4.1. Charge Sharing Width

To quantify the charge sharing, the *charge sharing coefficient* η is defined as:

$$\eta(x) = \frac{A_i(x) - A_{i+1}(x)}{A_i(x) + A_{i+1}(x)} \tag{4.1}$$

where $A_i(x)$ is the signal measured on strip i. This coefficient was measured as a function of the x position of the laser. An example is shown in figure 4.3. Formula 3.2 was used to fit $\eta(x)$. The fit paramter σ indicates here the width of the sharing zone and μ represents the centre of the charge-sharing zone. As the properties of the electric field (close to the surface of the sensor there is a region with low electric field) and the non

perpendicularity of the laser beam would indicate a dependence of the charge sharing on the focus position, measurements were taken with different focus distances, namely on the surface of the sensor (z = 0) and in the middle fo the bulk (z = 70.4 μm into the silicon sensor). Further, the offset of the charge sharing center (μ from the fit) and the

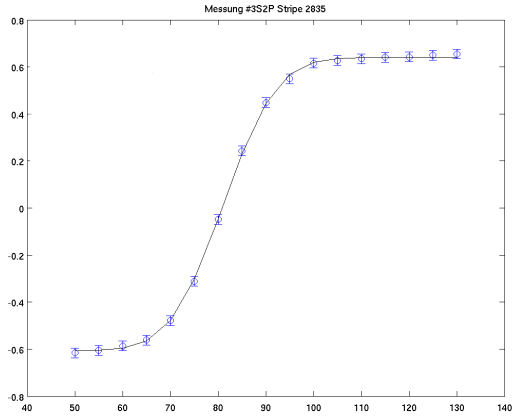


Figure 4.3.: Fit of the η function with formula 3.2

geometrical center between two strips was calculated for each measurement. To define the geometric center between two strips, the positions of the strip edges were determined by fitting the signal of the main strips on the outside of the charge-sharing zone with the formulae 3.1 and 3.2. An example is shown in figure 4.4. The geometrical center of the two strips is calculated as the difference between the two μ's from the fit.

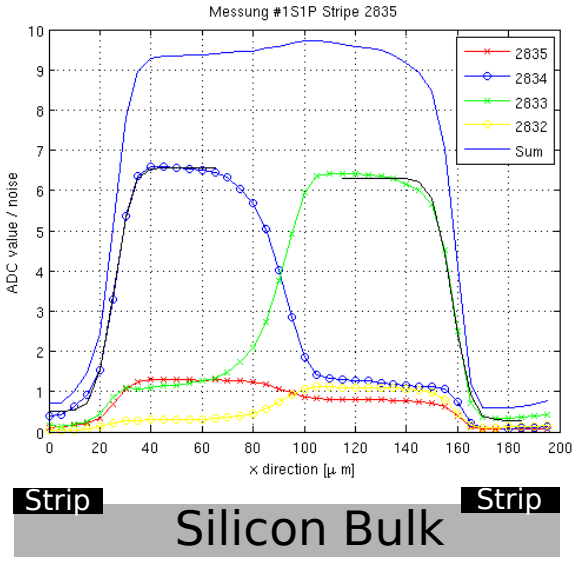


Figure 4.4.: Plot of the charge sharing with fit to calculate the geometrical center. Bias Voltage 300 V and focus on the surface

4.1.1. Measurements

Measurements on the charge sharing width were taken for different conditions. First were measured the width for different strips for different focus depth (on the sensor surface and in the middle of the bulk) and different bias voltages. Then measurements were also performed at different amplification and temperature.
Unless mentioned otherwise, the charge sharing was measured at bias voltage of 300 V, on the surface of the sensor, in between strip 2835 and 2834, at Temperature 293 K and at the saturation of 20 dB. In the following this is named *standard configuration*.

Different Focus Position

Measurement runs of 40 steps (200 μm) were taken at two focus positions (focus on the surface of the sensor and in the middle of the bulk) and fixed bias voltage. These runs were taken between strips 2833 and 2834 and between strips 2834 and 2835. Due to the geometrical properties of the silicon sensor, the charge-sharing for both runs (33-34 and 34-35) should be identical and symmetrical. This was verified by the measurement as seen in figure 4.5. Further the figure 4.6 show that the behavior of the crosstalk changed for even or odd strips. The crosstalk is for odd strips higher then for the even strips, due to the asymmetry of the crosstalk in the Beetle chip.

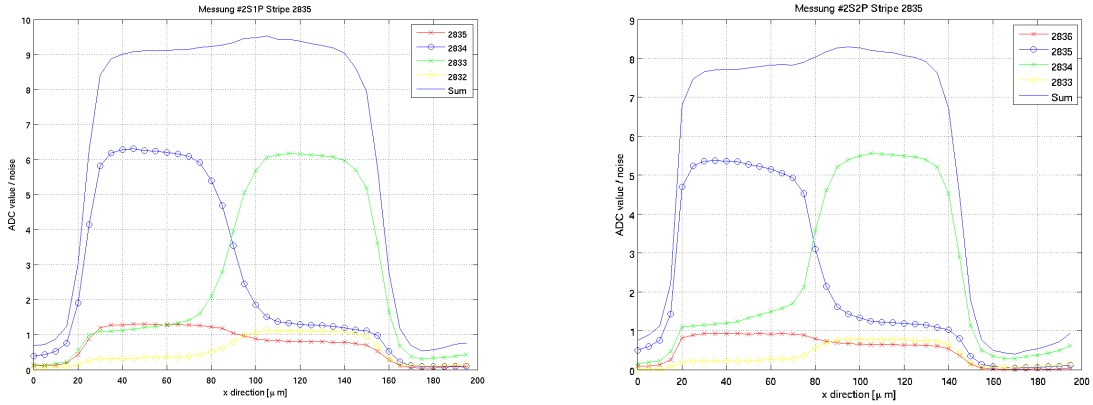


Figure 4.5.: Different Strips; *The left figure* - shows charge sharing measured at bias voltage of 250 V and in between strip 2834 and 2833. *The right figure* - shows charge sharing measured at bias voltage of 250 V.

Discussion: All results for the charge sharing zone width for different strips and different focus position agree with each other within errors. This indicates that the zone width is independent of the focus position and the same for different strips.

Different Bias Voltage

The drift velocity and diffusion of the charge carriers depend on the strength of the electric field. Larger diffusion results in a greater charge sharing. To investigate this

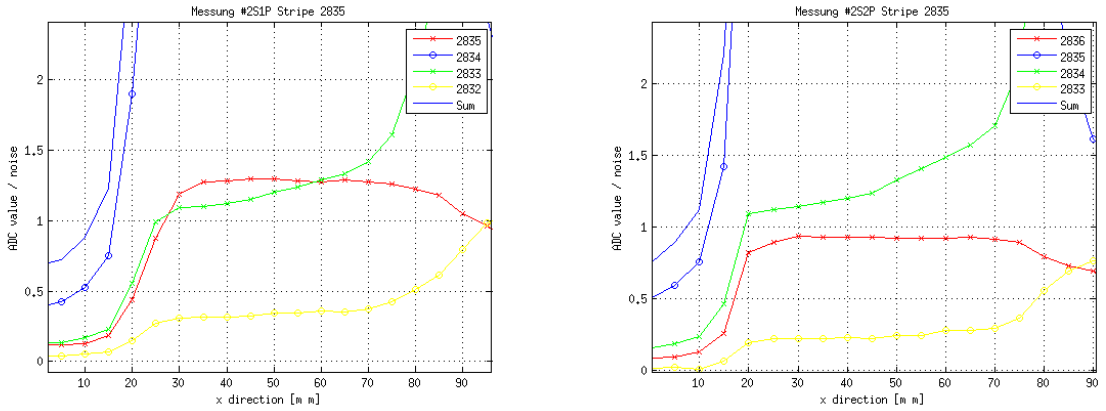


Figure 4.6.: Crosstalk; *The left figure* - shows the cross talk for odd strips at the standard configuration as mentioned. The signal amplitude is higher on 2835 then on 2834 *The right figure* - shows the cross talk for even strips at the standard configuration as mentioned. The signal amplitude is smaller on 2836 then on 2835.

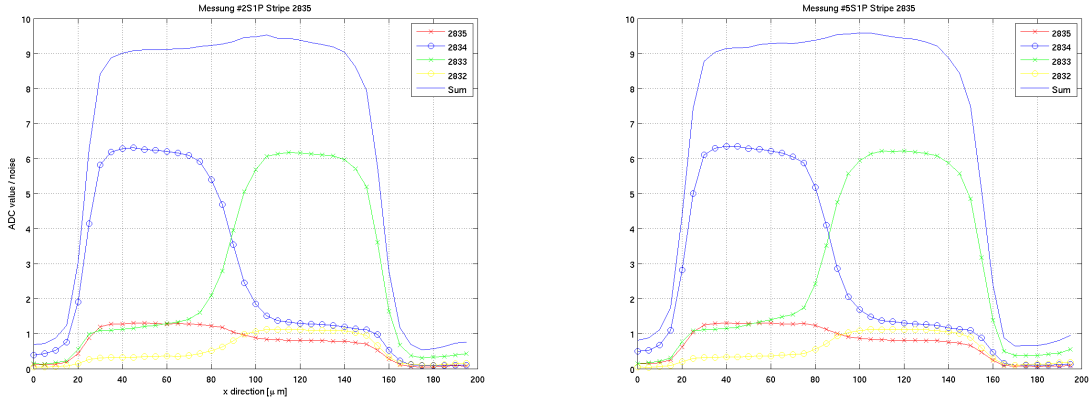


Figure 4.7.: Different Focus Position; *The left figure* - shows charge sharing measured at bias voltage of 250 V. *The right figure* - shows charge sharing measured at bias voltage of 250 V and at the focus position in the middle of the bulk.

Table 4.1.: Fitted parameter σ on surface

	2834 / 2833	2835 / 2834
Bias Voltage	Sharing Zone width	Sharing Zone width
300 V	(9.4±0.7)μm	(6.4±0.8)μm
250 V	(9.1±0.3)μm	(6.4±0.7)μm
200 V	(8.7±1.2)μm	(8.8±0.2)μm

Table 4.2.: Fitted parameter σ in the bulk

Bias Voltage	2834 / 2833 Sharing Zone width	2835 / 2834 Sharing Zone width
300 V	(7.7±0.5)μm	(9.7±0.2)μm
250 V	(8.2±0.3)μm	(9.6±0.2)μm
200 V	(5.4±1.5)μm	(7.9±2.1)μm

effect, charge sharing measurements were performed for different bias voltages.

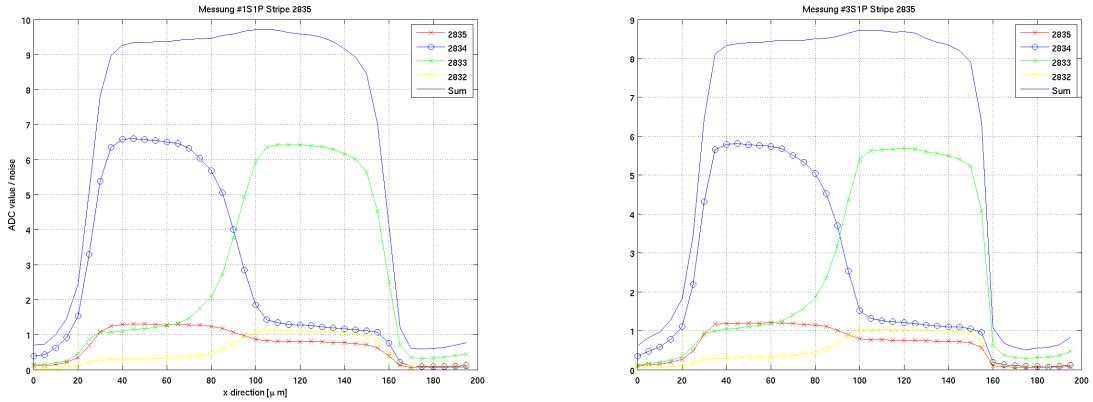


Figure 4.8.: Different Bias Voltage; *The left figure* - shows charge sharing measured at the standard configuration as mentioned. *The right figure* - shows charge sharing measured at bias voltage of 200 V.

Discussion: Table 4.1 and 4.2 shows the results of these measurements. The measurements at bias voltage of 200 V have a large error. Around 200 V the diode just starts to deplete which results in a highly fluctuated signal. The charge sharing does not have a significant dependence on the bias voltage, all measurements agree within their errors. The weighted mean over all data (different strips, bias voltage and focus position) gives a charge sharing zone width of 8.5±0.1μm.

Different Laser Beam Intensity

The laser signal amplitude was changed with the attenuator in the laser system to simulate different charge depositions in the sensor. The measurements made to investigate for some charge loss or dependence of the charge-sharing on the signal amplitude. The figure 4.9 shows measurements at 27 dB and at 20 dB.

Discussion: The charge sharing shows no significant dependence on attenuation (charge sharing width at 20dB: $\sigma = 7.6 \pm 0.3\mu$m).

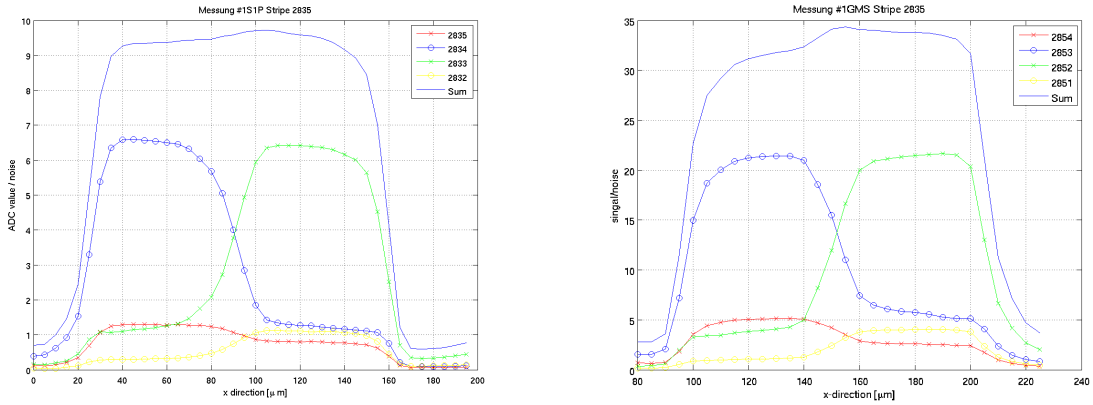


Figure 4.9.: Different dBs; *The left figure* - shows charge sharing measured at the standard configuration as mentioned. *The right figure* - shows charge sharing measured at the saturation 27dB.

Different Temperature

As the TT at LHCb was cooled down to 273 K, measurements were made of the charge-sharing at different temperature. Measurements were taken at 293 K and 273 K. The results are shown in the figure 4.10.

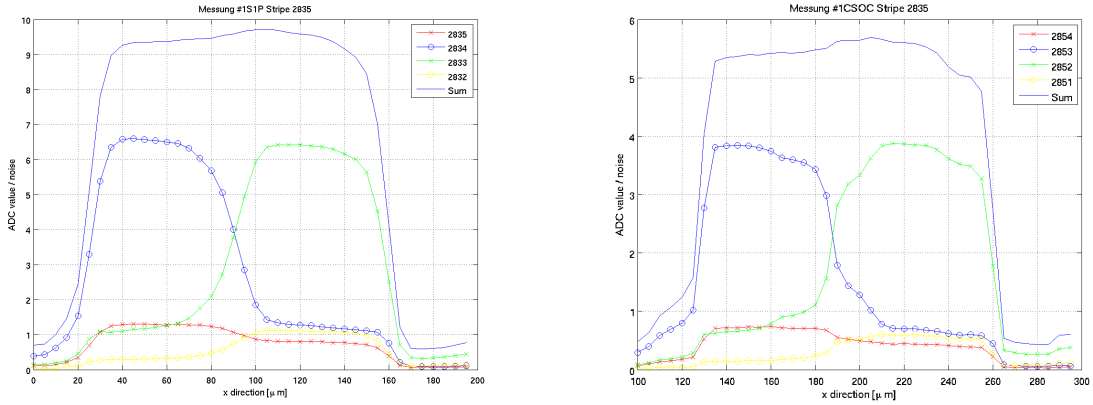


Figure 4.10.: Different Temperature; *The left figure* - shows charge sharing measured at the standard configuration as mentioned. *The right figure* - shows charge sharing measured at 273 K

Discussion: The charge sharing shows no dependence on the temperature ($\sigma = 9.2 \pm 1.9\mu$m at 273 K and $\sigma = 6.4 \pm 0.8\mu$m at 293 K). The reason for the large error on this measurements is a bad fit due to the angled curve as figure 4.10 shows. To distinguish the reason for this more specific measurements are needed. Further, the measured signal amplitudes are lower at 293K because the Beetle gain depends on temperature [17].

Asymmetry of the Charge Sharing

As mentioned before, the charge sharing could have some asymmetry due to misalignment of the laser beam. Mainly the charge sharing center could be different from the geometrical center between two strips. The geometrical center and the analytical center were calculated as mentioned before for different bias voltages, focus positions and different strips.

Table 4.3.: Fitted parameter μ on surface

	2834 / 2833		2835 / 2834	
Bias Voltage	Sharing Zone	SZ calculation	Sharing Zone	SZ calculation
300 V	(90.0±0.5)μm	(91.6±0.6)μm	(79.9±0.6)μm	(80.6±0.4)μm
250 V	(88.9±0.2)μm	(89.6±0.2)μm	(80.0±0.5)μm	(81.0±0.4)μm
200 V	(90.3±0.9)μm	(91.7±0.7)μm	(81.1±0.1)μm	(82.5±0.4)μm

Table 4.4.: Fitted parameter μ in the bulk

	2834 / 2833		2835 / 2834	
Bias Voltage	Sharing Zone	SZ calculation	Sharing Zone	SZ calculation
300 V	(85.1±0.4)μm	(86.2±0.2)μm	(80.0±0.2)μm	(81.1±0.6)μm
250 V	(86.2±0.2)μm	(87.9±0.4)μm	(78.9±0.2)μm	(80.6±0.8)μm
200 V	(87.6±1.1)μm	(89.7±0.7)μm	(81.1±1.5)μm	(82.5±1.2)μm

Discussion: As table 4.3 and 4.4 show, the differences between the calculated geometrical center and analytical center are small, We concluded from this that the small non perpendicularity of the collimator (see section 3.3 and 4) has only a small effect on charge sharing.

4.2. Charge Loss

To determine the charge loss the sum of the measured signals over two strips and over four strips were taken. A sum over four strips was taken to look also at the behavior of the crosstalk. The sum in each case should be constant over the area between two aluminium strips, if no charge loss exists. Otherwise a clear dip should exist as seen in earlier laser and test beam measurements. This analysis was performed for the same measurements as for the charge sharing width.

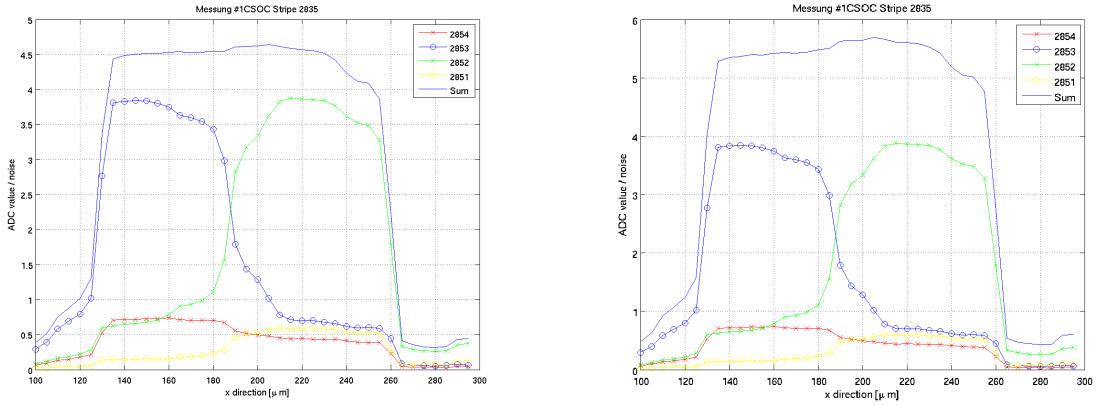


Figure 4.11.: *left figure* - It shows charge sharing at temperature of 273 K with summation over 2 strips. *right figure* - It shows charge sharing at temperature of 293 K with summation over 4 strips

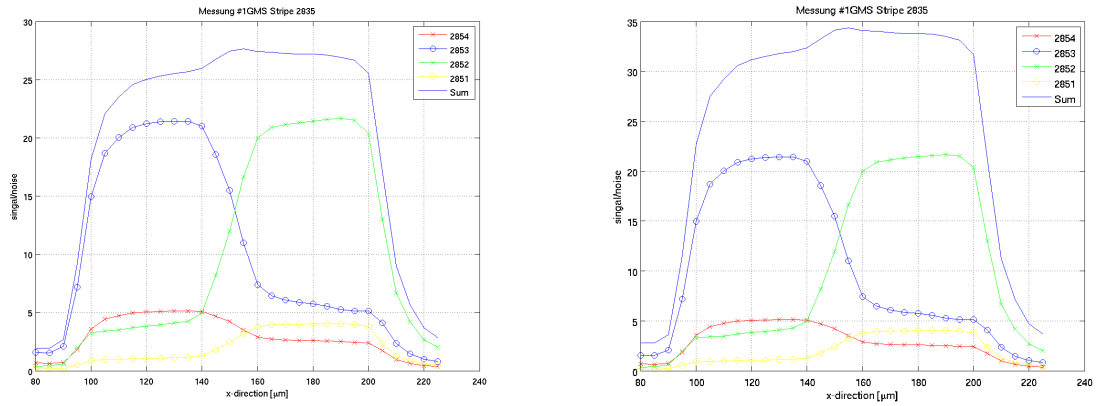


Figure 4.12.: *left figure* - It shows charge sharing at saturation of 27 dB with summation over 2 strips and in between strip 2853 and 2852. *right figure* - It shows charge sharing at saturation of 27 dB with summation over 4 strips and in between strip 2853 and 2852

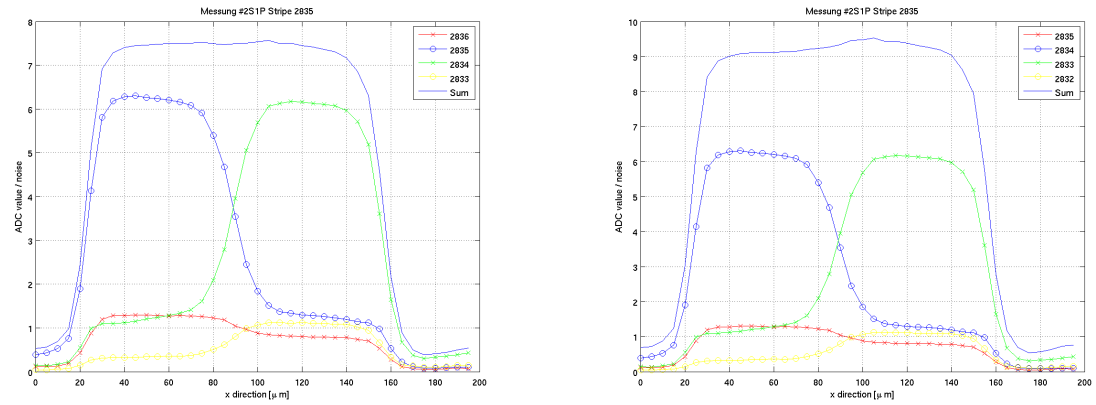


Figure 4.13.: *left figure* - It shows charge sharing with summation over 2 strips. *right figure* - It shows charge sharing with summation over 4 strips

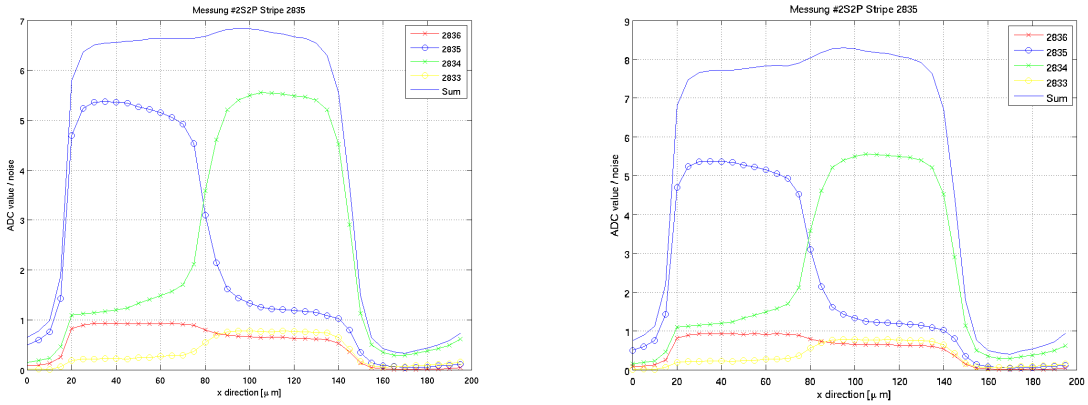


Figure 4.14.: *left figure* - It shows charge sharing in between strip 2834 and 2833 with summation over 2 strips. *right figure* - It shows charge sharing in between strip 2834 and 2833 with summation over 4 strips

Discussion: In all measurements, we observe no significant difference between summing over two or four strips. Every summation shows no charge loss. In the right half of the charge sharing zone, there is a little bump when summing over 4 strips. This bump is smaller for the sum over 2 strips. As mentioned in section 3.1, crosstalk in the Beetle chip causes some asymmetries. It looks like this small bump is an effect of the crosstalk, to verify this more measurements are needed for example on different strips.

5. Conclusion

At the end most time was invested in the focus alignment to verify the perpendicularity of the laser beam to minimize the geometrical charge sharing. That was the basis for the charge sharing measurements. In a series of measurements it was found, that the charge sharing does not depend clearly on temperature, attenuation, bias voltage, strip or position of the focus point. Width of charge sharing zone is measured to be in good agreement with results obtained from measurements in LHCb [22]. No charge loss was observed of the measurements.

"The important thing is not to stop questioning. Curiosity has its own reason for existing." - Albert Einstein

Bibliography

[1] *Properties of CMS-OB2-SSSD sensor type.* URL `http://lhcb.physik.uzh.ch/tt/sensorprob/F102821.JPG`. (Access: 18. February 2010).

[2] A. A. Alves et al. [LHCb Collaboration]. *The LHCb Detector at the LHC. JINST 3*, 2008.

[3] Claude Amsler. *Kern- und Teilchenphysik.* vdf Hochschulverlag AG, ETH Zürich, 2007.

[4] C. Bauer. *Grounding, Shielding and Power Distribution for the LHCb Silicon Tracking. LHCb-2004-101*, 2004.

[5] R. K. Bock and A. Vasilescu. *The Particle Detector BriefBook.* URL `http://rkb.home.cern.ch/rkb/titleD.html`. (Access: 17. Feburary 2011).

[6] Nicola Chiapolini. *Online Monitoring for the Silicon Tracker of the LHCb Experiment.* Institute of Physics University of Zurich, 2009.

[7] Christian Elsasser. *Extension of the TT Test Stand with a Pulsed Focused Infrared Laser.* Institute of Physics University of Zurich, 2009.

[8] M. Agari et al. *Test-beam measurements on prototype ladders for the LHCb TT station and Inner Tracker. LHCb Note 2003-082*, 2003.

[9] M. Agari et al. *Measurements of a prototype ladder for the TT station in a 120 GeV/c π^- beam. LHCb Note 2004-103*, 2004.

[10] J. Gassner. Measurements of prototype ladders for the tt station with a laser. *LHCb-2004-102*, 2004.

[11] St. Heule. *Simulation und Messung von Silizium-Streifen-Detektoren.* Institute of Physics University of Zurich, 2003.

[12] Virginia Semiconductor Inc. *Optical Properties of Silicon.* URL `http://www.virginiasemi.com/pdf/Optical%20Properties%20of%20Silicon71502.pdf`. (Access 17. February 2011).

[13] K. Kleinknecht. *Detectors for Particle Radiation.* Oldenbourg, 2^{nd} edition, 2006.

[14] Gabriel Landolt. *Setting up an LHCb TT-Detector Test Stand.* Institute of Physics University of Zurich, 2009.

[15] F. Lehner. *The LHCb Silicon Tracker. LHCb-2005-077*, 2005.

[16] LHCb-group. University of Zurich. URL `http://lhcb.physik.uzh.ch/`.

[17] S. Löchner and M. Schmelling. *The Beetle Reference Manual. LHCb-2005-105*, 2005.

[18] QPhotonics. *Datasheet QFLD-1060-30S-AR*. URL `http://www.qphotonics.com/product_image.php?imageid=3734`. (Access: 17. February 2011).

[19] Olaf Steinkamp. *Silicon strip detectors for the LHCb experiment. Nuclear Instruments and Methods in Physics Research A 541 (2005) 83–88*, 2005.

[20] Olaf Steinkamp. *Design and production of detector modules for the LHCb Silicon Tracker. Nuclear Instruments and Methods in Physics Research Section A*, 2007.

[21] JDS Uniphase. *Product Bulletin Manual Variable Optical Attenuators (VA4 Series)*. URL `http://www.datasheetarchive.com/VA4-datasheet.html`. (Access: 17. February 2011).

[22] J. van Tilburg. *Studies of the Silicon Tracker resolution using data. LHCb public note LHCb-PUB-2010-016*, 2010.

[23] Achim Vollhardt. *An Optical Readout System for the LHCb Silicon Tracker*. PhD thesis, University of Zurich, 2005.

A. The Test Stand Manual

This is an updated version of the manual describing how to operate the test stand (status: August 4, 2011).
The sections are taken over from [14] with only slight modifications.

A.1. How To Control Front-End Electronics

- Before starting the front-end electronics, make sure that the cooling system is running (how to control it is described in A.6).
- Make sure, that the DIM name server (DNS) is running on pc *lama*'s *Linux*-partition: Type in a console

  ```
  $ ps aux | grep /app/lhcb/pvss/PVSS_tfc/ ...
  fwComponents_20050810/bin/dns
  ```

  DNS is running if the corresponding process is prompted.

  The PVSS-project for controlling the Odin must not be running.
- Log into the *WINDOWS*-booted pc *DRACHE* as *Institut*.

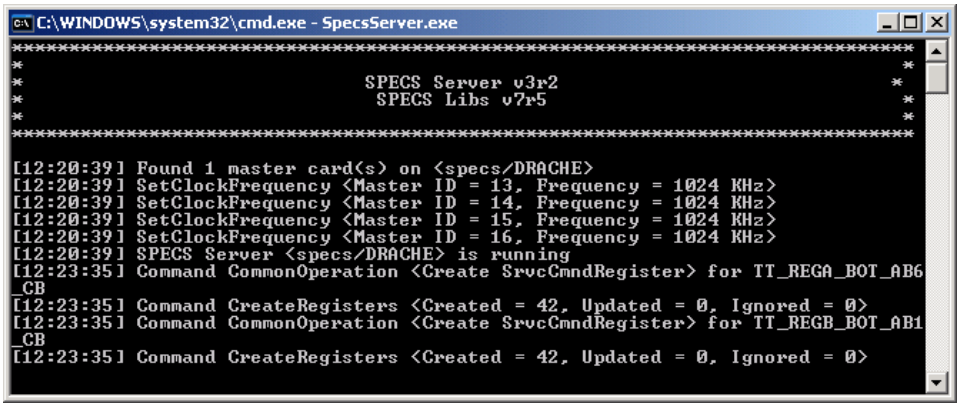


Figure A.1.: SPECS-server output after successful subscription of the Control Board hardware type

- Start the SPECS-server: Open a *WINDOWS* command prompt. Go to the project directory:

  ```
  : cd C:\ETM\PVSS2\3.6\TEST_STAND1\bin
  ```

Set the environment variable *DIM_DNS_NODE*:

```
: set DIM_DNS_NODE=lama.physik.uzh.ch
```

Start the SPECS-server:

```
: SpecsServer.exe
```

Don't close the window.

- Start the PVSS Administration in the *WINDOWS* Start menu.
- Select the project *TEST_STAND1*, hit the refresh button and then the green traffic light to start up the project. Wait until the whole project is loaded.

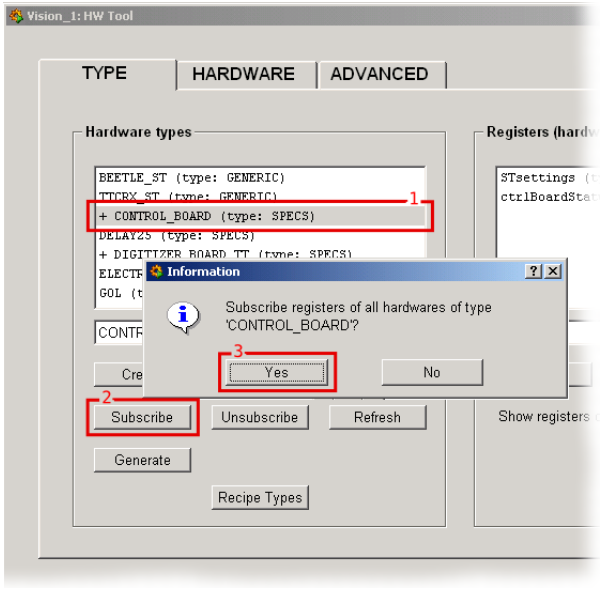


Figure A.2.: Subscription at the SpecsServer of hardware-types

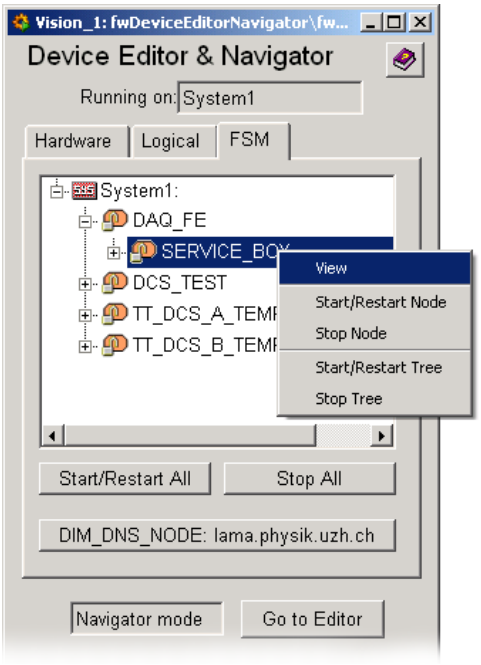


Figure A.3.: Device Editor & Navigator. Control unit objects marked with a lock, devices labelled with a yellow oval.

- Go to the panel called *HW Tool* and subscribe each hardware-type:
  1. select each one of the items listed under the tab *TYPE*
  2. hit `Subscribe`
  3. accept the prompts.

  If the subscription was successful one can see in the output in the SPECS-server window (fig. A.1). Otherwise restart the server and try again. See fig. A.2.

- Go to the panel *Device Editor & Navigator*. Change to the tab *FSM* and ensure that you are in the `Navigator mode`. Perform `Start/Restart All`. Wait until the prompt disappears. See fig. A.3.

- Initialize SPECS and power partitions: Right-click on `DCS_TEST`, in the appearing menu choose *View*.

  A window will open displaying all the sub-systems and their states. In this case the object is `DCS_BOX`. Take over control by clicking on the lock-sign and then pushing `Take`, the color of the drop-down menu displaying `NOT_READY` should change to yellow. Click on the `NOT_READY` and select from the drop-down menu `SWITCH_ON`. To configure send the recipe `PHYSICS` in the appearing window. The drop-down menu should change to green ground and the state `READY`. Further the red LED mounted on the box is on.

- Initialize front-end electronics: In the *Device Editor & Navigator* window right-click on the item `DAQ_FE`, choose *View*, take over the control. Click on the `NOT_READY` and select from the drop-down menu `Configure`. Use again the recipe `PHYSICS`. Because this is the wrong recipe for the ladders, the drop-down menu will turn into red and display `ERROR`. To correct this select from the drop-down menu `Recover`. The devices in error will go back to `NOT_READY` as well as the drop-down menu. Now select `Configure` again, but this time with the recipe `TEST|VFS400_NOTP`. The drop-down menu will show on blue ground `READY`.

- Start temperature readout: In the *Device Editor & Navigator* window right-click on `TT_DCS_B_TEMP_BOT_AB1`, then choose *View*. After setting *READY* `TT_AB1_HYBR_TEMP_P5X2` and `TT_TEMP_SBAB1`[1] the temperatures of the Hy-brid- and Control Board-sensors are readout.

  View the temperature curves: If not yet done start the *Trending Editor & Navigator* (look for `fwTrending` in the PVSS console). Ensure that it's in *Navigator mode*. Click on *Manage Plots/Pages*. Right-click on `System1:HYBR_TEMPS_3DB` to see the temperature plots (`CTRL`-`4` and -`8` allows zooming out in the x- and y-axis). Left-click allows configuration of the plot.

- A switch off has to be done by clicking on every drop-down menu displaying `READY` and choosing in the menu `Reset` or `Switch_OFF`. Note that objects can propagate commands to their sub-systems.

A.2. How To Send Triggers

- Login in the Linux-boot of *lama* as *lhcb*.

- Check whether DNS is running[2]:

[1] Due to a non-relevant programming error this device will change its state to *ERROR*.
[2] Use in general *XTerm* instead of normal consoles preventing memory overflows!

```
$ ps aux|grep dns
```

and look for `/app/lhcb/pvss/PVSS_tfc/ ... fwComponents_20050810/bin/dns`.
If not start it:

```
$ /app/lhcb/pvss/PVSS_tfc/fwComponents_20050810/. .. bin/dns
```

Important: If `ps aux|grep dns` delivers more than two processes, then the PVSS-project for Odin might be running. *The Odin-PVSS-project has to be shut down properly.*

- Check if sticky bit of `/tmp` is cleared:

  ```
  $ ls -ld /tmp
  ```

  If not clear it (as root):

  ```
  # chmod -t /tmp
  ```

- Start lock manager:

  ```
  $ source /app/lhcb/pvss/PVSS_tfc/setup
  ```

- Start Odin: Log in:

  ```
  $ ssh tfc@odinv100
  ```

  Password: `amex`
  Execute:

  ```
  $ start_TFC
  ```

  Don't close the shell.

- Start PVSS-project: In a new console type:

  ```
  $ startConsole
  ```

  Click on the green traffic light to start the project `TFC_LC`

- Go to the *Graphic Editor*, click on *Panel* → *Open* and select `TFC_LC/panels/TFC_top.pnl`. Start the panel by clicking *Panel* → *Save & View.*

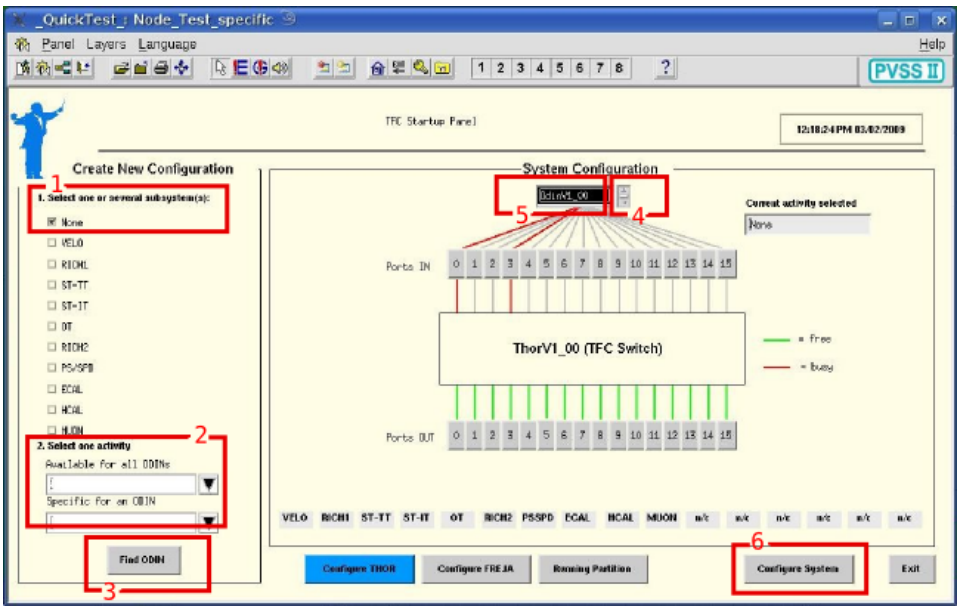


Figure A.4.: TFC Startup Panel

The following steps are visualized in fig. A.4.

- **1.** *Select one or several subsystem(s)* tick `None`.
  2. *Select one activity* choose `None` for *Available for all ODINs*.
  3. Click `Find ODIN`. Wait until in *System Configuration* Port 0 is highlighted red.
- **4.** Cycle through the prompt displaying `OdinDef` and pick `OdinV1_00`.
  5. Click into the prompt. The Port 3 will highlight red.
  6. Click on `Configure System`.

The window *TFC Local Run Control* should have opened. The numbers in the text refer to fig. A.5.

- **1a/b.** Verify that the states of `Partition_OdinV1_00` and `OdinV1_00` are both `RUN_NOT_READY` on orange ground. If this is not the case one has to perform a restart either of the project or of the entire PVSS environment.
  2. In the section *Initialization* click on `Subscribe Cnts` and `Counter Reset`.
- **3.** One can choose the desired triggers under section *L0 trigger*, e.g. *Periodic trigger A*.
- **4.** The configuration of the trigger can be done by clicking on `ODIN not ready` in the *Configuration* section.
  5. Configuration concerning the trigger frequency, offset and length or delay respectively for periodic and calibration trigger can be set in the *Periodic/calibration trigger SM* section. `Current` loads the current data (right column), `Default` loads the default data, `Apply` applies the new values (left column).

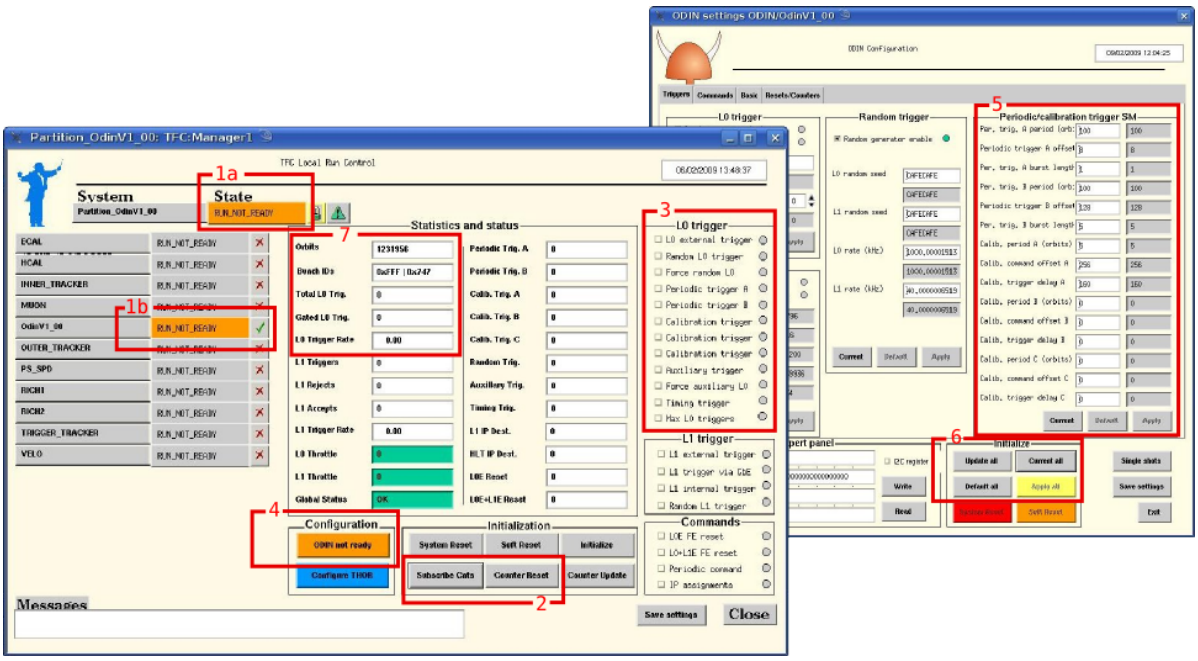


Figure A.5.: left: TFC Local Run Control, right: Odin Configuration

6. `Apply (all)` writes the desired configuration. (`Default all` can be useful after a loss of standard configuration.)

- **1a.** In order to start triggering click on the upper orange button `RUN_NOT_READY`, then on `GET_READY` and finally on `START_RUN`.

7. The L0 Trigger numbers should now count up.

- To avoid problems restoring the correct latency after the front end electronics have been off, it is advisable to make a *Soft Reset* by clicking on the `Soft Reset` in the window `Partition_OdinV1_00`. The Odin changes to the state described in **1a/b.**, but keeps the settings. Then perform a reset of the L0 Trigger by clicking in the window `ODIN_settings` on `Single shots`. A window appears with all the possible single shots. Click on `Shoot` next to `L0 Electronics Reset`.

A.3. How To Switch on the Laser

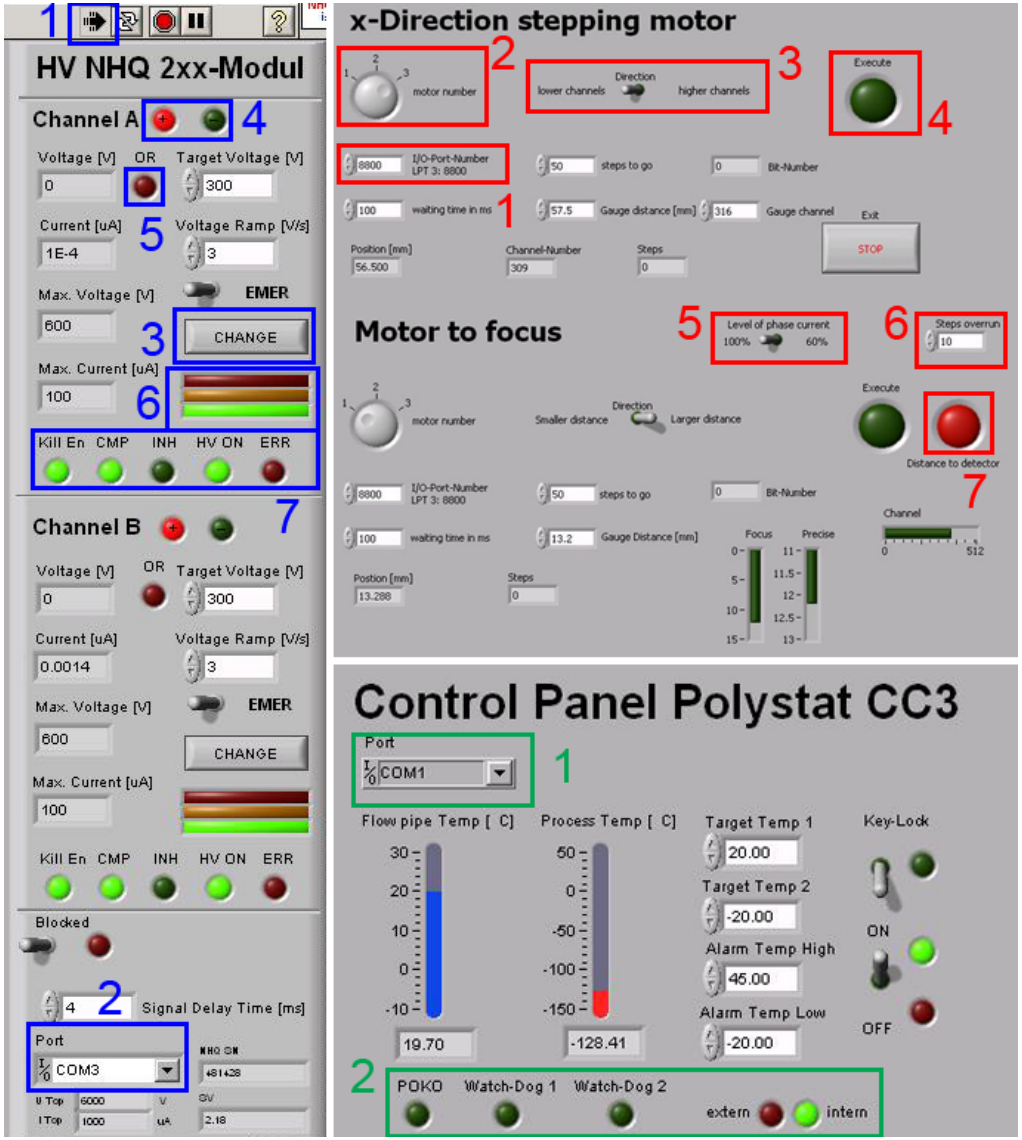
- Make sure that the flexes from Odin via NIM-crate into the test box are laid correctly (Connection `P1-R` on the Odin).
- Switch on the power supply next to the test box.

A.4. How To Control the High Voltage

- Open the program `St225.vi` or `St226.vi`[3] in the folder `HV-Steuerung` on the `Institut` account at `DRACHE`.
- Make sure that the selected port in the program (usually `COM3` or `COM4`) corresponds to the connection.
- Start the VI (Press the white arrow button in the prompt, turns to black). If the status display is green, the module is ready.
- If the status display is red, check the rocker switches `Contr.` (should be on `DAC`) and `HV` (should be on `ON`) at the modul.
- To change the target voltages, the ramp speeds or the delay time, put the designated value into the according box by keyboard and confirm it by clicking on to the control panel.
- To start changing the applied voltage to the target value hit `CHANGE`. While changing the applied voltage the status display is yellow.
- To bring the applied voltage immediately to $0\,V$, hit the switch `Emer.`.
- To avoid any accidental changing of the settings, hit the switch `Block`. A red light will appear.

Never turn high voltage on while the box is open! It immediately leads to breakdown current.

[3] `St225.vi` is to steer NHQ-225-modules whereas `St226.vi` is for NHQ-226-moduels



HV control (blue labels)
1: VI run button, **2:** port selection, **3:** button to start changing voltage, **4:** polarity indication, **5:** overrun indication, flashes when target voltage is higher than maximum voltage, **6:** status indication; green: ready, yellow: busy, red: error or not controlable, **7:** Kill En: Kill enable, CMP: control via RS232, INH: inhibiting signal over Lemo connection on control panel, HV ON: high voltage on, ERR: error

Step motor control (red labels)
1: port selection, **2:** port number on interface card, **3:** direction, **4:** execute command, **5:** motor current level (100% or 60% of maximum current), **6:** number of steps in reference direction, **7:** warning light flashing if collimator is nearer than 2 mm to the sensor

Polystat control (green labels)
1: port selection, **2:** POKO: dry contact; alarm lights; tempering mode

Figure A.6.: LabView panels; left: HV control, top right: step motor control, down right: Polystat control

A.5. How To Control the Stepping Motors

- Check power supply of the SMC800 interface.
- Open the program `Step_control.vi` in the folder `Step_control` on the `Institut` account at `DRACHE`.
- Make sure that the selected port in the program (usually `LPT3`) corresponds to the connection[4].
- Start VI.
- Be sure that the chosen motor numbers correspond to the actual connection.
- Select the number of steps and the direction.
- Hit `Execute`. Flashing green means the system waits until the previous running commands are finished.
- Before closing the program move back to the zero point. After restarting the program the actual point will be set to be zero.

A.6. How To Control the Cooling Device

- Open the program `Polystat_CC3.vi` in the folder `Polystat` on the `Institut` account at `DRACHE`.
- Make sure that the selected port in the program (usually `COM1`) corresponds to the connection.
- Start the VI.
- Put the rocker switch to `ON`.
- To change target or alarm temperatures, write down the desired value into the corresponding box.
- To avoid manipulations directly on the operator device, switch on the `Key lock`.

A.7. How To Start Data Aquisition

- You must be logged in the Linux-boot of *lama*.
- Connect to the TELL1's credit card pc:

[4]The hexadecimal code for `LPT3` is 0h8800.

```
$ ssh cc@ccpc18
Password: amex
```

- Start aquisition:

```
$ daq_tell1 ST5.v26_TTTELL01_labUZH.cfg
```

- Log into *Kamel*. Open a console. Write `<number>` events to binary file `<file>`:

```
# sudo /app/lhcb/ebuild/writeEventsToBinary/...
writeEventsToBinary <file> <number>
```

  (Binary files can be opened with `hexdump <file> | less`. The data is built out of the following elements: (magic pattern) (bank size) (version/type) (source ID). The magic pattern is in hex: `cbcb`.).

- Send desired triggers with Odin.

A.8. How To Open Data in ROOT

- Log into a machine on which the Gaudi anaylsis software is installed in the user's home directory and open a z-Shell. Append the following lines to the `~/.zshrc`-file.

```
# Environment for ROOT
export ROOTSYS=${CERN}/root ;
export PATH=${PATH}:${ROOTSYS}/bin ;
export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:
${ROOTSYS}/lib

# Environment for LHCb software
export MYSITEROOT=/app/lhcb/sw-slc4
export CMTCONFIG=slc4_ia32_gcc34
export CMTPATH=$HOME/cmtuser
export PATH=$MYSITEROOT/local/gcc-3.4.6/bin:
$MYSITEROOT/local/binutils-2.15.92.0.2/bin:
$MYSITEROOT/scripts:$PATH
export LD_LIBRARY_PATH=$MYSITEROOT/local/gcc-3.4.6
/lib:$MYSITEROOT/local/binutils- ...
2.15.92.0.2/lib:$MYSITEROOT
/lib:$LD_LIBRARY_PATH
. $MYSITEROOT/LbLogin.sh
```

- Make sure the folder `TTAnalysis.v7r1p1` is in the home directory (If not, copy it from `/home/hep/tobin`)

- Execute the following in a z-Shell:

```
$ SetupProject Vetra v7r1p1
$ cd TTAnalysis.v7r1p1
```

- Copy binary file to the current machine. Generate `.root`-file:

```
$ ./GetData <file>
```

  The names of the `.root`-files are displayed on the last line of the output.

- To open the histogram-file with *ROOT*:

```
$ root <file.root>
```

- To convert the n-tuples-file to an ASCII-file with *ROOT*:
  Open *ROOT*:

```
$ root
```

  Compile the converting program:

```
$ .L plotTuple2.C+
```

  Select the desired readout channels:

```
$ plotTuple2("<file.root>", ...
<chnl #>, <# of chnls on each side>)
```

B. Program Files

B.1. Data Processing

Listing B.1: plotTuple2.C; Program to convert root- to ASCII-file

```
#include "TTree.h"
#include "TFile.h"
#include "TProfile.h"
#include <vector>
#include <iostream>
#include <fstream>
#include <string>
void plotTuple2(const char* filename="/home/hep/tobin/data/...
2009.06.23/histo_190609_03-ntuples.root", const int strip=345, const
int width=3) {
  ofstream myfile;
  std::string fileout = filename;
  fileout += ".dat";
  myfile.open (fileout.c_str());
  //  myfile.open ("test.out");

  std::cout << "Dumping to " << fileout << std::endl;
  TFile* file = new TFile(filename);
  TTree* T = (TTree*)gDirectory->Get("TTDumpADCs/rawADCs");
  std::vector<float> adcs(3072,0.);
  T->SetBranchAddress("SourceID10",&adcs[0]);
  int n = T->GetEntries();
  for(int i=0; i<n; i++) {
    T->GetEntry(i);
    for(int j=-width; j<width+1; j++) {
        myfile << adcs[strip+j] << "\t";
    }
    myfile <<  "\n";
  }
   // myfile << strip << "\n";
  myfile.close();
  file->Close();

};
```

Listing B.2: GetData.m; Program to calculate signal and noise ADC-values per strip

```
%-------------------------------------------------------------------------
%       GetData
%-------------------------------------------------------------------------
%       Input: .dat files as pedestal data and raw data
%       Parameters: none
%       Output: mean.dat, std.dat, one figure
%       Command: GetData
%-------------------------------------------------------------------------

clear

%% Pedestal
%-------------------------------------------------
%singanture pedestal measurements (NoSignal)
ped_pre = 'FNS';
%Number of measurements for pedestal calculation
n_o_NS = 5;
ped = [];
ped_sig = [];

for Number = 1:1;
    for i=0:n_o_NS-1;
        data = load(['histo_',ped_pre,...
            sprintf('%d',Number*1000+i),'-ntuples.root.dat']);

        ped = [ped; data];
    end
end
[ped_data,ped_sig] = Smean(ped);
ped_sig = ped_sig./sqrt(length(ped));


%% Singal
%-------------------------------------
%Signatur of measurement (PulseScan)
pra = 'FS';

%Number of raw data measurments
n_o_D = 40;

for serieNumber = 1:25;
  cStrip = 2853;
  StripsOnSide = 4;
  save_format = 'pdf';
```

```
46  mes = [];
    coo = [];
    sdv= [];

    for i=0:n_o_D-1;
51    dat = load(['histo_',pra,sprintf('%d',serieNumber*10000+i),...
          '-ntuples.root.dat']);


      [data,data2] = Smean(dat);
56
      coo = [coo;i*5];
      mes = [mes;data];
      sdv = [sdv;data2];
    end
61
    sdv = sdv./sqrt(10000);
    dat = mes - meshgrid(ped_data,0:n_o_D-1);

    save(['data',sprintf('%d',serieNumber),'.dat']...
66    ,'dat','-ascii','-tabs')

    sdv = sqrt(sdv.^2 + meshgrid(ped_sig,0:n_o_D-1).^2);
    si = size(dat);
    measurement = 0:(si(1)-1);
71  measurement = measurement.*5;
    strip = (-StripsOnSide:StripsOnSide) + cStrip;
    measurementMesh = meshgrid(measurement,strip);
    stripMesh = meshgrid(strip,measurement);

76  %% Plot data
    screenSize = get(0,'Screensize');
    figure('Position',[1,1 screenSize(3)/1.6 screenSize(4)/2])
    surf(measurementMesh', stripMesh, dat)
    view(0,0)
81  hold on
    xlabel('x Distance [\mum]');
    ylabel('Strips');
    zlabel('ADC Value');
    title(['Messung #',num2str(serieNumber),' Stripe 314']);
86
    saveas(gcf,['focusmessung',sprintf('%d',serieNumber)...
      ,'.',save_format])
    save(['mean',sprintf('%d',serieNumber),'.dat'],'dat'...
      ,'-ascii','-tabs')
91  save(['std',sprintf('%d',serieNumber),'.dat'],'sdv'...
```

```
    ,'-ascii','-tabs')
  save(['coo',sprintf('%d',serieNumber),'.dat'],'coo'...
    ,'-ascii','-tabs')
end
```

Listing B.3: Smean.m; Function to remove values outside of 3σ

```
%-------------------------------------------------------------------------
%   Smean
%-------------------------------------------------------------------------
%   Input: matrix of data, each row is a different data set
%   Output: mean of the 3sigma correction and std
%   Parameter: none
%   Command: Smean(matrix)
%-------------------------------------------------------------------------


function [tempMean, tempStd] = Smean(da_ta)

tempMean = [];
tempStd = [];
lines = size(da_ta,1);

for row = 1:size(da_ta,2)
  temp1 = da_ta(:,row);
  st = std(temp1);
  me = mean(temp1);
  j = 5;

  while (j >= 1)
    std3 = 3*st;
    uline = me + std3;
    dline = me - std3;
    newMean = [];

    for i=1:length(temp1)
      if (temp1(i)<=uline) && (temp1(i)>=me)
      newMean = [newMean;temp1(i)];
      elseif (temp1(i)>=dline) && (temp1(i)<me)
        newMean = [newMean;temp1(i)];
      else
      end
    end

    me = mean(newMean);
    st = std(newMean);
    j = j-1;
```

```
  end

  tempMean = [tempMean,me];
  tempStd = [tempStd,st];
end
```

B.2. Focus Alignment

Listing B.4: focus.m; Program to calculate the fit parameter for the left and right edge

```
%-------------------------------------------------------------------------
%   Focus
%-------------------------------------------------------------------------
%   input: singal and rms n-tuples, coordinate ntuple, error
%          functions
%   parameter: none
%   output: three figures
%   command: focus
%-------------------------------------------------------------------------

save_format = 'png';

for measurement=1:25
  zname_vector = [-500,-400,-300,-200,-100,-50,-25,-20,-15,-10,...
    -5,-2.5,0,2.5,5,10,15,20,25,50,100,200,300,400,500];
  zname = zname_vector(measurement);

  dat = load(['mean',sprintf('%d',measurement),'.dat']);
  sdv = load(['std',sprintf('%d',measurement),'.dat']);
  coo = load(['coo',sprintf('%d',measurement),'.dat']);
  si = size(dat);

  %Region of interest
    %RegOfI = [7 16];   %% left edge
  RegOfI = [17 28];   %% right edge

  dat = dat(RegOfI(1):RegOfI(2),5);
  sdv = sdv(RegOfI(1):RegOfI(2),5);
  coo = coo(RegOfI(1):RegOfI(2));

  extrema = [min(coo) max(coo) min(dat) max(dat)];
  sig = sqrt(sdv);

  figure;
  errorbar(coo,dat,sdv,'r.')
```

```
hold on
xlabel('x Distance [\mum]');
ylabel('ADC Value');
coo_contin = extrema(1)-3:0.1:extrema(2)+3;


%Determation of side
if dat(1) > dat(length(dat))
  side = 'left'
else
  side = 'right'
end

title(['Messung #',num2str(measurement),' Stripe 2853',...
  ' / ',num2str(zname),'\mum / ', side]);

%Fitting of error function
if strcmp(side,'left')
  start_para = [dat(1)-dat(length(dat)),dat(length(dat)),...
  (coo(1)+coo(length(coo)))/2, 1];
  [f para check iter corr_para cov_para] = ...
      leasqr(coo,dat,start_para,'edgeleft',0.00001,2000,1./sig);
  plot(coo_contin, edgeleft(coo_contin, para),'k-');


  %%Chi quadrat Test
  %-------------------------------
  chi = sum(((dat-edgeleft(coo,para))./sdv).^2);
  ndf = length(dat)-length(para);
  para_chi_left = [chi ndf]
else
  start_para = [-dat(1)+dat(length(dat)),dat(length(dat)),...
  (coo(1)+coo(length(coo)))/2, 1];
  [f para check iter corr_para cov_para] = ...
      leasqr(coo,dat,start_para,'edgeright',0.00001,2000,1./sig);
  plot(coo_contin, edgeright(coo_contin, para),'k-');


  %%Chi quadrat Test
  %-------------------------------
  chi = sum(((dat-edgeright(coo,para))./sdv).^2);
  ndf = length(dat)-length(para);
  para_chi_right = [chi ndf]
end

scale = chi/ndf
```

```
    if scale>1
     para_err = sqrt(diag(cov_para)).*sqrt(scale);
    else
      para_err = sqrt(diag(cov_para));
    end

    saveas(gcf,['FE',sprintf('%d',measurement),'_',side,'.'...
      ,save_format])
    save(['para',sprintf('%d',measurement),'_',side,'.dat']...
      ,'para','-ascii','-tabs')
    save(['para_err',sprintf('%d',measurement),'_',side,...
      '.dat'],'para_err','-ascii','-tabs')
end
```

Listing B.5: edgeleft.m; Function to describe left edge

```
%Error function for focus edge left

function F = edgeleft(x,p)
% p(1) = amp  p(2) = offset    p(3) = mu  p(4) = sig
F = p(1)*(1-erf((x-p(3))/(sqrt(2)*p(4))))+p(2)
```

Listing B.6: edgeright.m; Function to describe right edge

```
%Error function for focus edge right

function F = edgeright(x,p)
% p(1) = amp  p(2) = offset    p(3) = mu  p(4) = sig
F = p(1)*(erf((x-p(3))/(sqrt(2)*p(4))))+p(2)
```

Listing B.7: testp3.m: Program to calculate the edge position as dependence on z

```
%-------------------------------------------------------------------
%   Test mu
%-------------------------------------------------------------------
%   Input: fitparameter from focus.m
%   Output: fit-parameter for mu and errors
%   Parameter: none
%   Command: testp3
%-------------------------------------------------------------------

pL = [];
pR = [];
pL_err = [];
pR_err = [];
mes = [-500;-400,;-300;-200;-100;-50;-25;-20;-15;-10;-5;-2.5;0;...
   2.5;5;10;15;20;25;50;100;200;300;400;500];
```

```
for measurement = 1:25
  para_left = load(['para',sprintf('%d',measurement),'_',...
    'left','.dat']);
  para_right = load(['para',sprintf('%d',measurement),'_',...
    'right','.dat']);

  paraerr_left = load(['para_err',sprintf('%d',measurement),...
    '_','left','.dat']);
  paraerr_right = load(['para_err',sprintf('%d',measurement),...
    '_','right','.dat']);

  pL = [pL;para_left(3)];
  pR = [pR;para_right(3)];
  pL_err = [pL_err;paraerr_left(3)];
  pR_err = [pR_err;paraerr_right(3)];
end


%% left
  figure
  errorbar(mes,pL,pL_err,'o')

  xlabel('z-Direction [\mum]')
  ylabel('Value parameter \mu [\mum]')
  title('plot parameter \mu left; 2853')
  hold on

  %Fit data linPlot
  sig = sqrt(pL_err);
  start_para = [(pL(1)-pL(25))/2, min(pL) ];
  [f para check iter corr_para cov_para] = ...
    leasqr(mes,pL,start_para,'linPlot',0.00001,2000,1./sig);
  plot(mes, linPlot(mes, para),'k-');

  %%Chi quadrat Test
  %-----------------------------------
  chi = sum(((pL-linPlot(mes,para))./pL_err).^2);
  ndf = length(pL)-length(para);
  para_chi_left = [chi ndf]

  scale = chi/ndf

  if scale>1
    para_err = sqrt(diag(cov_para)).*sqrt(scale);
  else
```

```
    para_err = sqrt(diag(cov_para));
  end

  save(['para','_pleft','.dat'],'para','-ascii','-tabs')
  save(['para_err','_pleft','.dat'],'para_err','-ascii','-tabs')


%% right
  figure
  errorbar(mes,pR,pR_err,'o')

  xlabel('z-Direction [\mum]')
  ylabel('Value parameter \mu [\mum]')
  title('plot parameter \mu right; 2853')
  hold on

  %Fit data linPlot
  sig = sqrt(pR_err);
  start_para = [(pR(1)-pR(25))/2, min(pL) ];
  [f para check iter corr_para cov_para] = ...
    leasqr(mes,pR,start_para,'linPlot',0.00001,2000,1./sig);
  plot(mes, linPlot(mes, para),'k-');

  %%Chi quadrat Test
  %-----------------------------------
  chi = sum(((pR-linPLot(mes,para))./pR_err).^2);
  ndf = length(pR)-length(para);
  para_chi_left = [chi ndf]

  scale = chi/ndf

  if scale>1
    para_err = sqrt(diag(cov_para)).*sqrt(scale);
  else
    para_err = sqrt(diag(cov_para));
  end

  save(['para','_pright','.dat'],'para','-ascii','-tabs')
  save(['para_err','_pright','.dat'],'para_err','-ascii','-tabs')
```

Listing B.8: testp4.m: Program to calculate the focus position and focus size

```
%-------------------------------------------------------------------
%   Test sigma
%-------------------------------------------------------------------
%   Input: fitparameter from focus.m
%   Output: fit-parameter for sigma and errors
```

```
%    Parameter: none
%    Command: testp4
%-------------------------------------------------------------------------

pL = [];
pR = [];
pL_err = [];
pR_err = [];
mes = [-500;-400,;-300;-200;-100;-50;-25;-20;-15;-10;-5;-2.5;0;...
   2.5;5;10;15;20;25;50;100;200;300;400;500];

extrema = [min(mes) max(mes) min(mes) max(mes)];
mes_contin = extrema(1)-3:0.1:extrema(2)+3;

for measurement = 1:25
   para_left = load(['para',sprintf('%d',measurement),'_',...
     'left','.dat']);
   para_right = load(['para',sprintf('%d',measurement),'_',...
     'right','.dat']);

   paraerr_left = load(['para_err',sprintf('%d',measurement),...
     '_','left','.dat']);
   paraerr_right = load(['para_err',sprintf('%d',measurement),...
     '_','right','.dat']);

   pL = [pL;para_left(4)];
   pR = [pR;para_right(4)];
   pL_err = [pL_err;paraerr_left(4)];
   pR_err = [pR_err;paraerr_right(4)];
end


%% plot left
   figure
   errorbar(mes,pL,pL_err,'o')

   xlabel('z-Direction [\mum]')
   ylabel('Value parameter \sigma [\mum]')
   title('plot parameter \sigma left; 2853')
   hold on
   axis([-600 600 -20 40])
   grid

   %fit data with spotfcn
   sig = sqrt(pL_err);
   start_para = [pL(1)-pL(length(pL)),min(pL),...
```

```
    (mes(1)-mes(length(mes)))/2];
  [f para check iter corr_para cov_para] = ...
    leasqr(mes,pL,start_para,'spotfcn',0.00001,2000,1./sig);
  plot(mes, spotfcn(mes, para),'r-');

  %%Chi quadrat Test
  %-----------------------------------
  chi = sum(((pL-spotfcn(mes,para))./pL_err).^2);
  ndf = length(pR)-length(para);
  para_chi_left = [chi ndf]

  scale = chi/ndf

  if scale>1
    para_err = sqrt(diag(cov_para)).*sqrt(scale);
  else
    para_err = sqrt(diag(cov_para));
  end

  para_left=para
  para_err_left=para_err
  save(['para','_P4_left','_','.dat'],'para','-ascii','-tabs')
  save(['para_err','_P4_left','_','.dat'],'para_err','-ascii','-tabs')


%% Plot right
  figure
  errorbar(mes,pR,pR_err,'o')

  xlabel('z-Direction [\mum]')
  ylabel('Value parameter \sigma [\mum]')
  title('plot parameter \sigma right; 2853')
  hold on
  axis([-600 600 -20 40])
  grid

  %fit data with spotfcn
  sig = sqrt(pL_err);
  start_para = [pR(1)-pR(15),min(pR),(mes(1)-mes(length(mes)))/2];
  [f para check iter corr_para cov_para] = ...
    leasqr(mes,pR,start_para,'spotfcn',0.00001,2000,1./sig);
  plot(mes_contin, spotfcn(mes_contin, para),'r-');

  %%Chi quadrat Test
  %-----------------------------------
  chi = sum(((pR-spotfcn(mes,para))./pR_err).^2);
```

```
    ndf = length(pR)-length(para);
    para_chi_right = [chi ndf]

    scale = chi/ndf

    if scale>1
     para_err = sqrt(diag(cov_para)).*sqrt(scale);
    else
     para_err = sqrt(diag(cov_para));
    end

    save(['para','_P4_right','_','.dat'],'para','-ascii','-tabs')
    save(['para_err','_P4_right','_','.dat'],'para_err','-ascii','-tabs')
```

B.3. Charge Sharing

Listing B.9: chargeSharing.m: Program to calculate the sharing zone

```
%-------------------------------------------------------------------------
%   Charge sharing
%-------------------------------------------------------------------------
%   Input: data and std from GetData.m
%   Output: sharingzone width and sharingzone position
%           (geoemtrical and   from fit)
%   Parameter: none
%   Command: chargeSharing
%-------------------------------------------------------------------------


save_format = 'png';
dataID = 'S2P';

for measurement =1:6
  dat = load(['data',dataID,sprintf('%d',measurement),'.dat']);
  sdv = load(['std',dataID,sprintf('%d',measurement),'.dat']);
  coo = load(['coo',dataID,sprintf('%d',measurement),'.dat']);
  stripe = 5;

  plot1 = dat(:,stripe+1);
  plot2 = dat(:,stripe);
  plot3 = dat(:,stripe-1);
  plot4 = dat(:,stripe-2);

  sdv1 = sdv(:,stripe+1);
  sdv2 = sdv(:,stripe);
```

```
sdv3 = sdv(:,stripe-1);
sdv4 = sdv(:,stripe-2);


%%Charge sharing
  plotTotal = plot1 + plot2 + plot3 + plot4;
  figure
  plot(coo,plot1,'r-x',coo,plot2,'b-o',coo,plot3,'g-x',...
    coo,plot4,'y-o',coo, plotTotal)
  grid on
  hleg1 = legend('2835','2834','2833','2832','Sum')
  hold on
  title(['Messung #',num2str(measurement),dataID,...
    ' Stripe 2835']);
  xlabel('x direction [\mu m]')
  ylabel('ADC value / noise')
  saveas(gcf,['ladungsverteilung',sprintf('%d',measurement)...
    ,dataID,'.',save_format])


%% calculate the center

%left
left_Reg = [1 14];

sdv2b = sdv2(left_Reg(1):left_Reg(2));
sig2 = sqrt(sdv2b);
plot_left = plot2(left_Reg(1):left_Reg(2));
coo_left = coo(left_Reg(1):left_Reg(2));

start_para_left = [plot_left(1)-plot_left(length(plot_left))...
  ,plot_left(length(plot_left)),(coo_left(1)+...
    coo_left(length(coo_left)))/2, 1];
[f para check iter corr_para cov_para] = ...
  leasqr(coo_left,plot_left,start_para_left,'edgeleft'...
    ,0.00001,2000,1./sig2);
plot(coo_left, edgeleft(coo_left, para),'k-');

  %%Chi quadrat Test
  %-------------------------------
  chi = sum(((plot_left-edgeleft(coo_left,para))./sdv2b).^2);
  ndf = length(plot_left)-length(para);
  para_chi_left = [chi ndf]

  scale1 = chi/ndf
```

```
    if scale1>1
      para_err = sqrt(diag(cov_para)).*sqrt(scale1);
    else
      para_err = sqrt(diag(cov_para));
    end

    edge_left = para(3);
    edge_left_err = para_err(3);

  %right
  right_Reg = [24  38];

  sdv3b = sdv3(right_Reg(1):right_Reg(2));
  sig3 = sqrt(sdv3b);
  plot_right = plot3(right_Reg(1):right_Reg(2));
  coo_right = coo(right_Reg(1):right_Reg(2));

  start_para_right = [plot_right(1)-plot_right(length(plot_right))...
    ,plot_right(length(plot_right)),(coo_right(1)+...
      coo_right(length(coo_right)))/2, 1];
  [f para check iter corr_para cov_para] = ...
    leasqr(coo_right,plot_right,start_para_right,'edgeright'...
      ,0.00001,2000,1./sig3);
  plot(coo_right, edgeright(coo_right, para),'k-');

    %%Chi quadrat Test
    %-------------------------------
    chi = sum(((plot_right-edgeright(coo_right,para))./sdv3b).^2);
    ndf = length(plot_right)-length(para);
    para_chi_right = [chi ndf]

    scale = chi/ndf

    if scale>1
        para_err = sqrt(diag(cov_para)).*sqrt(scale);
    else
        para_err = sqrt(diag(cov_para));
    end

    edge_right = para(3);
    edge_right_err = para_err(3);


  center = edge_left + (edge_right-edge_left)/2;
  center_err = 0.5 * sqrt(edge_left_err.^2 + edge_right_err.^2);
```

```
120
    %% Charge sharing zone

    RegOfI = [11  27];

125 coo2 = coo;
    coo = coo(RegOfI(1):RegOfI(2));

    plot1 = plot1(RegOfI(1):RegOfI(2));
    plot2 = plot2(RegOfI(1):RegOfI(2));
130 plot3 = plot3(RegOfI(1):RegOfI(2));
    plot4 = plot4(RegOfI(1):RegOfI(2));

    sdv1 = sdv1(RegOfI(1):RegOfI(2));
    sdv2 = sdv2(RegOfI(1):RegOfI(2));
135 sdv3 = sdv3(RegOfI(1):RegOfI(2));
    sdv4 = sdv4(RegOfI(1):RegOfI(2));

    etha = [];

140 for i = 1:length(plot1)
      et = (plot3(i)-plot2(i))/(plot2(i)+plot3(i));
      etha = [etha;et];
    end

145
    % fit errorfnc

    sdv = sqrt((2*plot2/(plot3+plot2).^2)*sdv2.^2+ (2*plot3/...
    (plot3+plot2).^2)*sdv3.^2);
150 sig = sqrt(sdv);

    start_para = [etha(1)-etha(length(etha)),etha(length(etha)),...
      (coo(1)+coo(length(coo)))/2, 1];
    [f para check iter corr_para cov_para] = ...
155   leasqr(coo,etha,start_para,'edgeright',0.00001,2000,1./sig);

      %%Chi quadrat Test
      %-------------------------------
      chi = sum(((abs(etha)-abs(edgeright(coo,para)))./sdv).^2);
160   ndf = length(etha)-length(para);
      para_chi_center = [chi ndf]

      scale = chi/ndf

165   if scale>1
```

```
        para_err = sqrt(diag(cov_para)).*sqrt(scale);
    else
        para_err = sqrt(diag(cov_para));
    end

  %% plot data + fit
  figure
  errorbar(coo,etha,sdv,'o')
  hold on
  plot(coo, edgeright(coo, para),'k-');
  hold on
  text(40.5,0.6,['SZ width = ', num2str(para(4)),'\pm',...
    num2str(para_err(4)),' \mum']);
  text(40.5,0.5,['SZ position= ', num2str(para(3)),'\pm',...
    num2str(para_err(3)),' \mum']);
  text(40.5,0.4,['mPoint calcPic= ', num2str(center),'\pm',...
    num2str(center_err),' \mum']);
  text(40.5,0.2,['Scale = ', num2str(scale)]);
  title(['Messung #',num2str(measurement),dataID,' Stripe 2835']);

  saveas(gcf,['chargeShareZone',sprintf('%d',measurement),...
    dataID,'.',save_format])
end
```