GEORG-AUGUST-UNIVERSITÄT
GÖTTINGEN

Fakultät für
Physik

Bachelor's Thesis

Modultests für den ATLAS-Pixeldetektor

Module tests for the ATLAS pixel detector

prepared by

Timo Hüser

from Erlangen

at the II. Physikalischen Institut

Thesis number: II.Physik-UniGö-BSc-2018 /01

Thesis period: 9th April 2018 until 16th July 2018

First referee: Prof. Dr. Arnulf Quadt

Second referee: Priv.Doz. Dr. Jörn Große-Knetter

Contents

1. Introduction

The quest to better understand the fundamental physics governing the universe have driven humanity towards innovation and collaboration for centuries. The biggest and arguably most ambitious project trying to achieve this goal is the Large Hadron Collider (LHC) experiment at the European research organisation CERN. Since its comissioning in 2008 the LHC and its main detector experiments ATLAS and CMS managed to push the frontiers of particle physics significantly, the most prominent example being the discovery of the Higgs boson in 2012. After 10 years of collecting data in two runs, the first lasting from 2009 to 2013 and the second one from 2015 to 2018 with an upgrade period in between, the LHC will be shut down for major upgrades and repairs once the 2018 run concludes. This upgrade, called the High Luminosity Large Hadron Collider (HL-LHC) will try to increase the luminosity, a measure of the collision rate in the accelerator, by a factor of ten.

During this update not only the LHC but also many of the detector subsystems in the ATLAS and CMS experiments will undergo major upgrades. This includes the ATLAS pixel detector which will get replaced by an entirely new version. Building and testing the thousands of pixel modules required for this obviously is a very time consuming process. One step in the testing sequence, described in more detail later, involves irradiating the modules under test with a radioactive source. Previously the testing setup for this procedure allowed only one module to be tested at a time. This not only makes the process tedious and labour intensive but also requires the operator to handle the radioactive source for every module that gets tested.
The following chapters will investigate how a motorised platform to move the radioactive source can help to automate and parallelise this step in the testing procedure and the design of such a mechanism will be discussed in detail.

2. Hadron collider physics at the LHC

One way to gain insight into the underlying physics of the interaction between elementary particles is to collide them at very high energies, allowing to observe processes and interactions that do not happen at ordinary energy scales.
Starting in the first half of the 20th century physicists began to build these accelerators, starting with small cyclotrons reaching energies of a couple of MeV. New accelerator designs and technologies were rapidly developed, leading to higher and higher collision energies. In 1989 the predecessor of the LHC, the electron-positron collider LEP at CERN, already reached energies of 91 GeV, topping out at 209 GeV in the year 2000 after several upgrades [1]. At the same time the proton-antiproton collider TEVATRON at FERMILAB managed to reach center-of-mass energies of up to 1.96 TeV [2].
Both of those colliders, as well as the LHC are ring colliders. In a ring collider the particles are kept on a circular trajectory by a magnetic field produced by dipole magnets arranged around the ring. The big advantage of this architecture as compared to linear accelerators is that the particles can be stored in the ring, enabling multiple crossings of the accelerated beams. A major disadvantage is that the charged particles in the ring are under constant acceleration, making them loose parts of their energy through synchrotron radiation [3]. The energy loss produced by synchrotron radiation ΔE follows the equation:

$$\Delta E = \frac{(Ze)^2 \cdot E^4}{\epsilon_0 \cdot 3R \cdot (mc^2)^4}.$$

As is evident from this relation to minimise those losses the ring diameter needs to be as big as possible and the mass of the accelerated particles should be large. This is the main reason why newer accelerators trying to achieve energies of over 1 TeV tend to use protons and antiprotons instead of much lighter electrons.

To this day the LHC (Large Hadron Collider) at CERN is the most powerful particle accelerator ever built, colliding either protons or heavy lead ions at energies of up to

13 TeV. The LHC is a ring collider with a diameter of 27 km located in an underground tunnel under the city of Geneva in Switzerland [4]. CERN currently has 22 member states from all over the world providing the necessary funds and technology to accomplish this feat. To collect the data from the high energy particle collisions that happen in the LHC there are a total of seven detector experiments placed along the ring. The two biggest ones are the general purpose detectors ATLAS and CMS. Having two independent detectors collecting data makes it possible to cross validate the results and gain a much higher confidence in the discoveries made. The remaining two detectors serve specific purposes, ALICE is designed to study the heavy ion collisions produced at the LHC and LHCb is devised to study matter-antimatter asymmetries. In addition there are three smaller experiments called TOTEM, LHCf and MOEDAL placed at the sites of the bigger experiments taking additional specialised measurements.

The construction of the LHC was approved in 1994, 14 years later in 2008 the LHC was put into service for the first time. Shortly after the first start up a magnet quench caused by an electrical fault halted the experiments for about a year and delayed the start of the first run to the end of 2009. During the first run, which concluded in 2013 the centre-of-mass-energy of the colliding beams was raised from an initial 2.36 TeV to 7 TeV. The data collected during this run famously led to the discovery of the Higgs boson [5, 6].

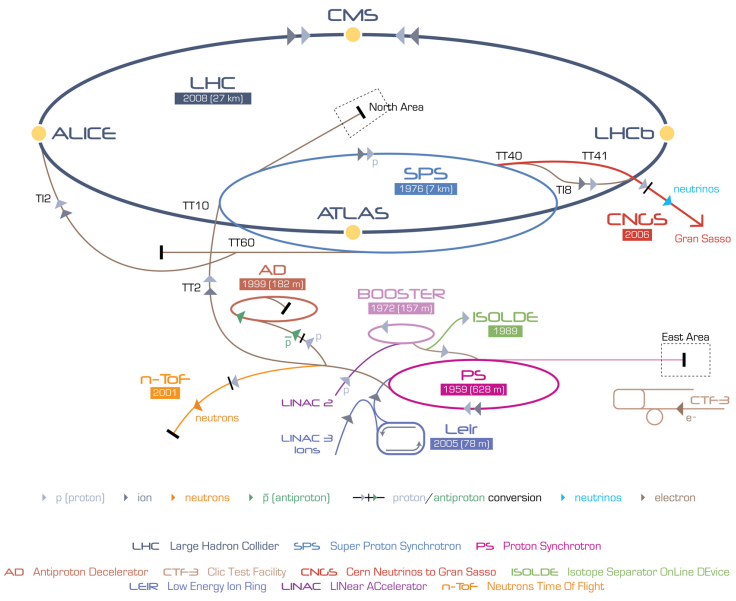


Figure 2.1.: Schematic overview of the accelerators at CERN including the LHC.

After this run the LHC was shut down for upgrades and maintenance for two years. The second run started in April of 2015 at a center-of-mass energy of 13 TeV and is still

ongoing [7].
Planned upgrades to the LHC to get implemented in the coming years after the second run is concluded focus on increasing the luminosity of the collider. While the integrated luminosity in the year 2017 alone reached a value of 50.2 fb^{-1} it still is not sufficient to study very rare events like certain decay channels of the Higgs boson with high precision [8]. Hence the High-Luminosity LHC project has been initiated to increase the luminosity by a factor of ten beyond the design value [9]. This update brings a series of new requirements not only to the accelerator architecture itself, but also to the ATLAS and CMS detectors and their hardware. The general architecture of the ATLAS experiment and some of the changes necessary for the HL-LHC upgrade will be discussed in the following chapters.

3. The ATLAS detector

The ATLAS (A Toroidal LHC ApparatuS) detector is one of the seven detectors at the LHC. The construction of ATLAS was finished in 2008 with the first data being collected during the first run of the LHC in 2009. Operation of the detector is currently planned to continue until at least 2035 with a major upgrade for the HL-LHC scheduled to be finished in 2025 [10].

3.1. Different detector subsystems

As the name suggests ATLAS uses a hybrid magnetic system consisting of a solenoid magnet producing a central 2 T strong magnetic field parallel to the beam axis and three magnets producing a toroidal field inside the muon detector and the end-caps with a strength in the range of $0.5 - 1$ T. All four magnets used are cryogenically cooled superconducting coils.
The detector can be divided into three sub-units, namely the tracking chamber, the electromagnetic and the hadronic calorimeters and the muon chamber [11].

Inner detector The inner detector is used to reconstruct the tracks of charged particles emerging from collisions in the detector. This tracking information can then be used to gain vertex and momentum information about those particles. For every bunch crossing event there are on the order of 10^3 particles produced. Such a bunch crossing event happens every 25 ns so the tracking system needs to process a huge number of events while still giving very good momentum and vertex resolution. This necessitates not only fast detection and readout electronics but also a very sophisticated trigger system that can identify and process relevant events in a very tight time frame.
The inner detector consists of three modules which can be seen in Figure 3.1. A Pixel tracker, discussed in more detail in section 3.3, forms the innermost layer, followed by silicone microstrip trackers (SCT) and straw tube transition radiation trackers (TRT). The Pixel and microstrip tracker cover pseudorapidity regions of $|\eta| = 2.5$ while the TRT

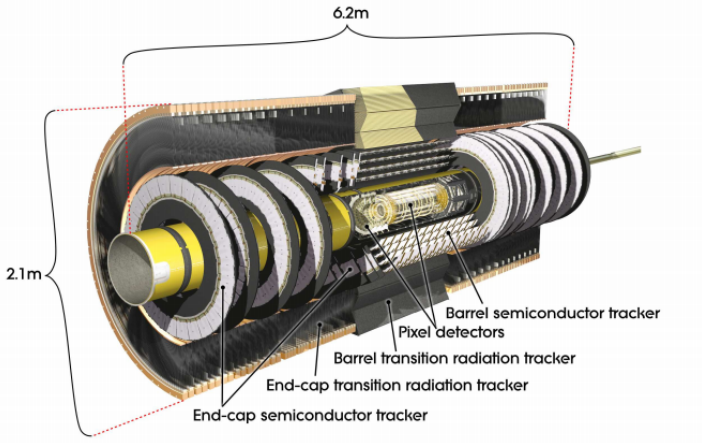


Figure 3.1.: Overview of the different components of the ATLAS inner detector, reprinted from [11].

covers the region $|\eta| = 2.0$ [11].

Calorimeters The ATLAS detector uses two calorimeters, an electromagnetic calorimeter, detecting mainly showers produced by electrons and photons and a hadronic calorimeter to detect hadronic showers. The construction and layout is shown in Figure 3.2. Both of the calorimeters used are sampling calorimeters. The electromagnetic one uses liquid Argon (LAr) as the active medium with lead as an absorber. The hadronic calorimeter consists of the tile calorimeter covering the $|\eta| < 1.7$ region using scintillating tiles as the active material and steel as the absorber in conjunction with LAr end-caps covering the pseudorapidity region of $1.5 < |\eta| < 3.2$ using copper plates as the absorbing material.

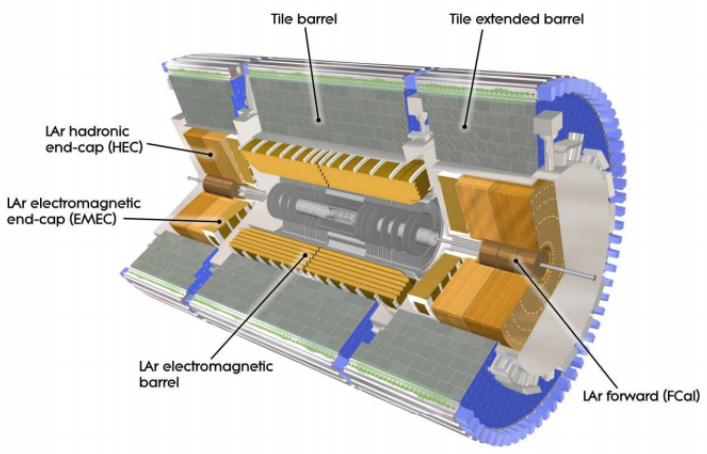


Figure 3.2.: Overview of the calorimeters used in the ATLAS detector, reprinted from [11].

To prevent showers extending over the hadronic calorimeter into the muon chamber an adequate thickness of the calorimeters is required. The thickness of the electromagnetic calorimeter is 22 radiation lengths χ_0 in the barrel region and 24 χ_0 in the end-cap regions. The hadronic calorimeter has a thickness of 11 interaction lengths λ which provides sufficient punch-through prevention.

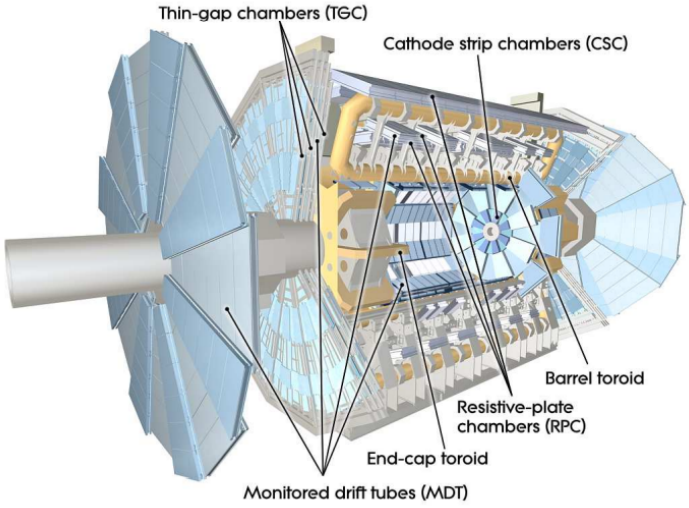


Figure 3.3.: Overview of the muon chamber used in the ATLAS detector, reprinted from [11].

Muon Chamber Muons usually pass through the tracking detector and the calorimeters, so an additional detector layer, the muon chamber is needed to provide energy and momentum measurements for these particles. The layout of the muon spectrometer is shown in Figure 3.3.

The muon chamber is designed to measure the tracks of traversing muons under the influence of the magnetic field produced by the toroidal barrel and end-cap magnets. The muons in the region $|\eta| < 1.4$ get deflected by the barrel toroids while the muons traversing in the $1.6 < |\eta| < 2.7$ region are influenced mainly by the end-cap toroids with a transition region covering $1.4 < |\eta| < 1.6$ where both magnets contribute to the bending of the track. In the inner region ($|\eta| < 2$) the precision tracking is done with cathode strip chambers using resistive plate chambers for triggering. In the outer region monitored drift tubes are used for the precision measurement and thin gap chambers provide the triggering information.

Using the more complex toroidal magnetic field in the muon chamber has the advantage that the bending of the tracks happens in a plane orthogonal to the bending caused by

the solenoid field in the inner detector. This provides two independent measurements of the momenta of traversing particles.

3.2. Trigger System

Because of the high collision rate in the ATLAS detector it is necessary to implement a trigger system that can pick out potentially interesting events and thus reduce the data rates so they can be handled by the following analysis and storage systems. In the ATLAS detector the trigger system has two levels, a hardware implemented first level (L1) and a software implemented high level trigger (HLT). Those two trigger levels manage to reduce the event rate from an initial 40 MHz to approximately 1 kHz. A schematic overview of the currently used trigger system can be found in Figure 3.4.

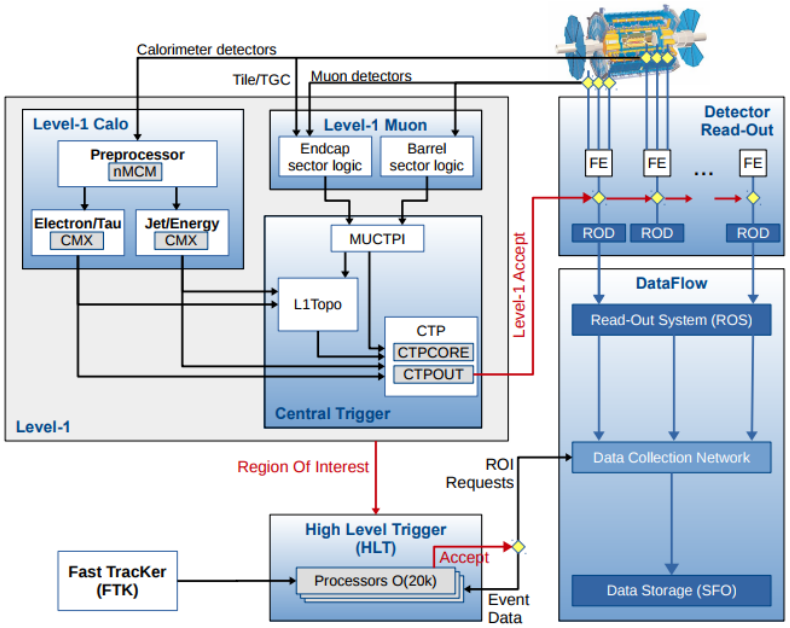


Figure 3.4.: Overview of the two level trigger system used in the ATLAS detector, reprinted from [12].

Level-1 Trigger The hardware implemented level-1 trigger uses coarse granularity output data of the two calorimeters and the muon spectrometer to pick out events containing high transverse momentum (p_T) electrons, photons, jets, muons and high missing transverse energy. If such an event is present it determines a so called region of interest (RoI) in the detector that is subsequently used to seed the HLT. The L1 trigger reduces the data rate from 40 MHz to 100 kHz with a maximum latency of 2.5 μs required by the

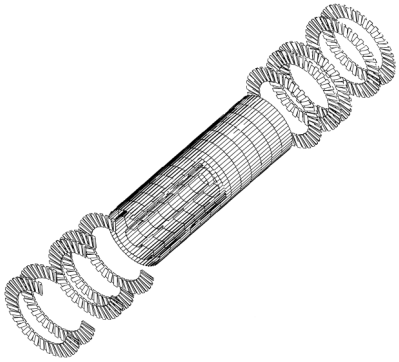


Figure 3.5.: Overview of the ATLAS Pixel detector showing the three barrel layers and the two end-caps, reprinted from [13].

collision rate [11].
The initial L1 trigger system used in the first run of the LHC relied mainly on application specific integrated circuits (ASICs), in the upgraded version for the second run field-programmable gate array (FPGA) technology was used which enabled the use of more complex algorithms like auto-correlation filters or bunch-by-bunch dynamic pedestal correction. This change improved pile-up suppression and fake trigger rejection. Another upgrade implemented to allow for more detailed reconstruction to be performed later in the HLT is the addition of the Fast TracKer (FTK) to the L1 trigger system. It can perform a global track reconstruction after each L1 trigger event, this information can then be used to feed the HLT and thus enable it to perform more advanced triggering tasks, like for example triggering on b-jets. Without this information this task would be computationally too expensive for the HLT and its required latency [12].

High Level Trigger The High Level Trigger (HLT) is almost entirely implemented on commercially available server hardware making it as flexible and cost effective as possible. The HLT gets seeded with the RoIs detected by the L1 trigger and consequently performs more detailed analysis and reconstruction of those areas, taking into account data from all the available detectors including the tracking system. If an event passes the HLT requirements the HLT sends out this data set including all the analysis and reconstruction data it has produced for permanent storage and subsequent offline analysis. At this point the data rate is reduced to a final rate of roughly 1 kHz. The HLT accomplishes this with a latency of roughly 0.2 s [11, 12].

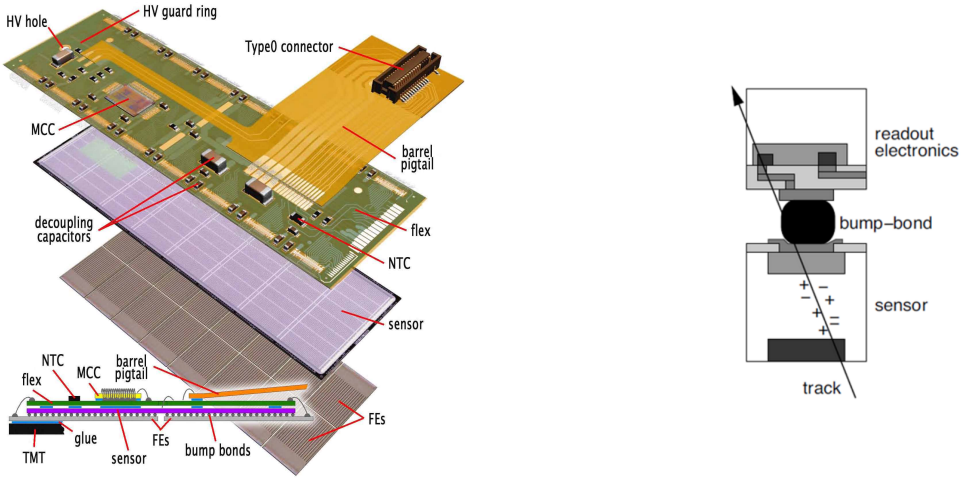


Figure 3.6.: Construction of a single ATLAS Pixel module including its module control chip, reprinted from [13, 15].

3.3. The ATLAS Pixel Detector

At the LHC bunch crossings happen at a rate of 40 MHz with each bunch crossing producing over 10^3 particles. Very precise primary vertex tracking is necessary to identify the collisions those particles originate from. Additionally the precise determination of secondary vertices is important for the high energy collisions at the LHC. This secondary vertex information is for example needed for identifying b-jets by their impact parameter (b-tagging). In the ATLAS detector vertex determination is mainly accomplished by the Pixel detector, making up the innermost layers of the detector. Being so close to the beam pipe leads to very high irradiation of the Pixel modules making it necessary to make them as radiation hard as possible [14].

3.3.1. The current ATLAS Pixel Detector

A schematic overview of the layout of the ATLAS Pixel detector can be found in Figure 3.5. The detector consists of four barrel layers and two end-caps with five discs each. In total the detector has a diameter of 26.5 cm and spans a length of 184 cm measured between the two outermost end-cap discs. ATLAS uses 1744 identical pixel sensor modules each containing 47232 pixels. The modules used are hybrid semiconductor sensors using n-type implants on a n-type substrate being bump bounded to their corresponding front end readout chips. The big advantage of using N-in-N sensors is that they can be used partially depleted so they do not get destroyed by a radiation induced type inversion and a relatively low bias voltage can be used. These properties ensure that the high radiation hardness specifications are met [13].

The innermost layer of the Pixel detector, called the Insertable B-Layer (IBL) was installed in May 2014 between the original Pixel detector and a new beam pipe with a smaller radius than the original one. This update was necessary to mitigate radiation damage done to the original pixel detector and to deal with increasing bandwidth requirements. With the IBL the FE-I4 front-end chip using 130 nm CMOS technology and the newly developed 3D Silicone Detectors were used inside the detector for the first time [16].
Figure 3.6 depicts a single Pixel sensor module showing the semiconductor sensor being bump bonded to the front end electronics which in turn get wire bonded to the module control chip (MCC) [13]. While the bump bonding technique allows for very compact module designs and reduces the area of the detector that is not covered by active sensors to a minimum it is very difficult to manufacture and causes very high production costs for these modules.

3.3.2. Planned upgrades for the HL-LHC

As previously mentioned the LHC and with it the ATLAS detector will undergo major upgrades after the second run is completed to significantly increase the luminosity, planning to achieve an additional integrated luminosity of 2500 fb^{-1} over ten years. The so called High Luminosity LHC (HL-LHC) is scheduled to be completed in 2024 and development and production of all the required new hardware is currently underway.

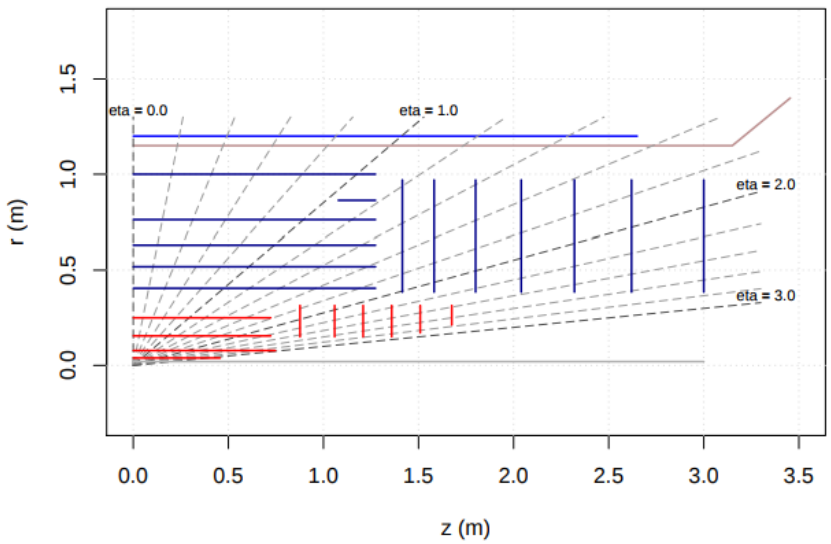
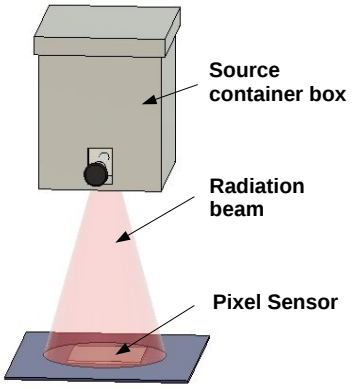


Figure 3.7.: Planned layout of the tracker system for the HL-LHC upgrade of ATLAS showing the pixel detector in red, reprinted from [10].

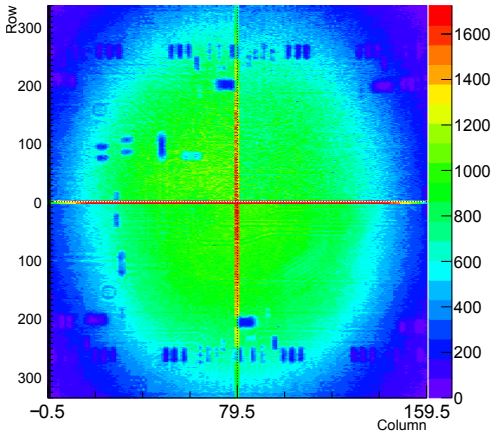
The Pixel detector is the detector stage closest to the beam pipe, this position subjects it to the highest irradiation of all the components of ATLAS. This makes it necessary to replace the entire Pixel detector (and in fact the entire inner detector) for the HL-LHC update to guarantee sufficient radiation hardness. The higher luminosity also leads to the production of even more particles per bunch crossing, which would saturate the bandwidth of the currently installed Pixel readout system.
Besides those changes necessary to keep the pixel detector operational, the new version will also increase angular coverage and employ smaller pixels (25×150 μm^2 as opposed to 50×400 μm^2 for the old detector) to increase the granularity of the detector and use new read-out chips to handle the increased bandwidth requirements.
The currently planned layout of the Inner Tracker (ITk) for the HL-LHC can be seen in Figure 3.7, showing the four pixel barrel layers and six discs per end-cap in red. This layout has a total active sensor area of 8.2 m^2 containing $638 \cdot 10^6$ pixels. Alternative layouts like a conical shape of the barrel layers to reduce the required sensor surface area or adding a fifth pixel layer are being considered. The fifth layer would improve the patter recognition seeding capabilities of the pixel detector significantly and enable better two-particle separation in high p_T jets at the cost of higher complexity, increased production cost and additional material [10].

4. Teststand design

Before being integrated into the ATLAS detector the newly designed and built pixel modules need to be tested and qualified. This testing procedure includes irradiating the modules with a radioactive source as shown schematicaly in Figure 4.1a. This produces a so called hit map as shown in Figure 4.1b that can be used to check if all the pixels are working as expected and all the bump bonds are intact. Handling the radioactive source is time consuming and human exposure should be kept to a minimum. This is achieved by building a teststand that makes it possible to test four modules at once. In contrast to the previous setup, which could only irradiate one module at a time, the new setup requires an automatically movable radiation source to irradiate all the modules.



(a) Schematic view of the source scan procedure



(b) Example of the hit map produced by the source scan

Figure 4.1.: Source scan procedure.

This is done using a motorised moving platform able to travel in the plane parallel to the modules. Commercially available solutions (e.g. the RK E linear unit[1]) would have only allowed travelling along a single axis for roughly $10 - 15$ cm which is not enough to irradiate four modules. Thus it was decided to build a completely custom solution that would allow enough movement to cover the required area. This custom design also allows

[1] `https://www.rk-rose-krieger.com/fileadmin/catalogue/lineartechnik/en/lt_e_en.pdf`

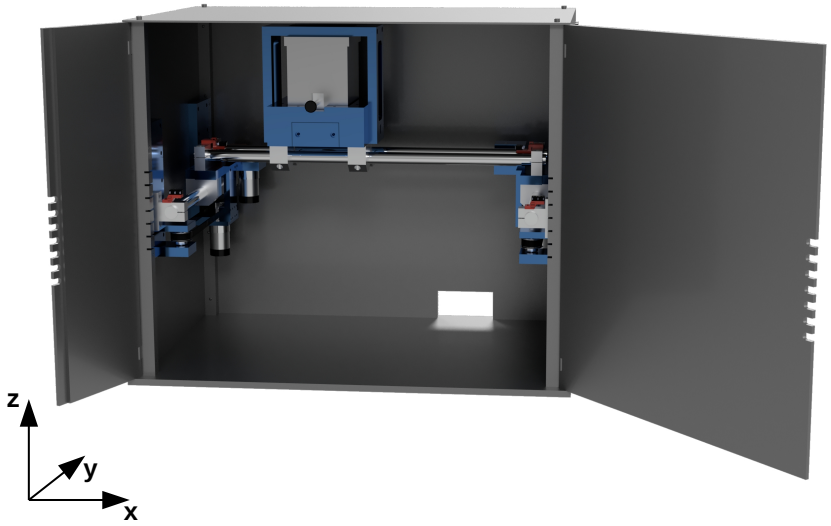


Figure 4.2.: Overview of the design of the teststand.

to implement all the safety and interlock features for working with the radioactive source. In the following sections the design of the hardware, the controller software and the safety mechanisms as well as the integration into the STcontrol software will be described in detail.

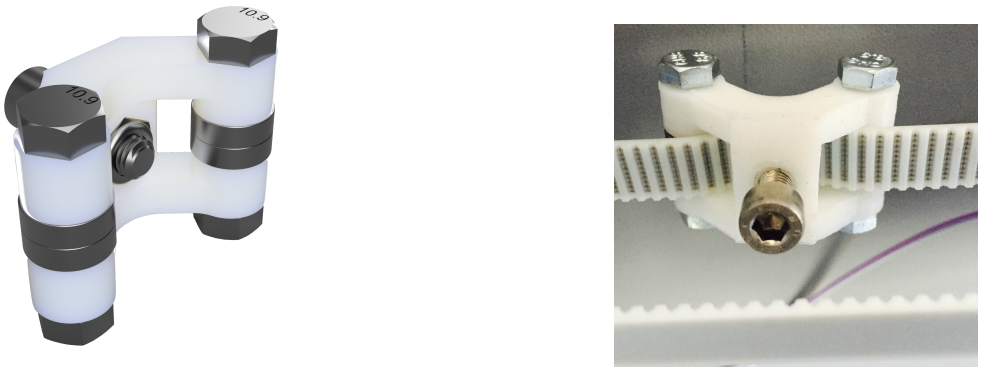
4.1. Hardware design

The previously used setup for doing the source scans consisted of a big steel box for radiation shielding with a smaller box containing the radioactive Americium or Strontium sample mounted on a horizontal plate inside. The module under test was mounted directly below this plate and a manual shutter mechanism was used to shield the source until the doors of the big box were closed. The new motorised platform replaces the horizontal plate that was used to mount the source box in a fixed position by a movable mount.
The overall design of the moving platform is strongly inspired by the design of commercially available 3D printers and other CNC machines. Because all the hardware has to fit into an already built steel box used for radiation shielding a compact design based on belts and pulleys was chosen. The pulleys will be driven by highly geared down (131:1 and 164:1 for the x- and y-Axis respectively) DC motors[2]. To make sure the platform can easily carry the source box, weighing 5 kg, 2 cm thick steel rails with correspondingly sized linear bearings were used to build the base structure. The bearings used are rated to carry a static load of up to 800 N. Given that the weight of the moving parts of the

[2] `https://www.robotshop.com/media/files/pdf2/003-rb-dfr-233.pdf`

	Box size [mm]	Platform travel [mm]
x-axis	650	325
y-axis	420	165
z-axis	550	132 (in steps of 44)

Table 4.1.: Overview of platform travel and box size.



(a) Design of the belt tensioner

(b) Installed belt tensioner

Figure 4.3.: Belt tensioning mechanism.

machine including the source box is around 16.5 kg and the load will be evenly spread on four bearings the static load on each bearing is not expected to exceed 50 N. Since the machine will not undergo high accelerations dynamic loads can be neglected.
A complete 3D model of the machine was created using the CAD package *Autodesk Fusion 360*. All the CAD files, along with technical drawings for all the custom made parts, can be found in the GitLab repository listed in Table A.2. This 3D model was used to make sure the design did not contain any errors, like parts intersecting at some point along the movement and also provided the necessary drawings required by the workshop to manufacture the custom made parts.

As apparent in Figure 4.2 the belt and pulley based design allows to put the motors out of the way of the radioactive source to maximise the travel of the machine. This results in a fairly efficient use of the build volume defined by the steel box. An overview of the machine travel and overall size is given in table 4.1. One disadvantage of a belt and pulley design is that it inherently has large amounts of backlash. This is caused by the necessary flexibility in the belts. To minimise this effect and to make the assembly as simple as possible the belt tensioners shown in Figure 4.3 were added to every belt. These enable very precise tensioning of the belts so the previously mentioned disadvantages can be minimised.

For better clarity the different directions of motion of the platform from now on will be

referred to as x-, y-, and z-axis as indicated in Figure 4.2.
To reduce cost and build time as much as possible and to allow easy sourcing of replacement parts care was taken to use as many standard DIN parts as possible. The parts coloured in blue in Figure 4.2 are custom made. Aluminium was chosen as the material for most of them since it is relatively lightweight and commonly available while still being strong enough to provide the necessary structural stability. An overview of all the used commercially available parts can be found in Table A.1.
The machine, shown in some more detail in Figure 4.4, can be split up into two functional units, called x-, and y-axis. The y-axis describes the stationary part mounted directly to the steel box that contains the motor and pulley assembly responsible for driving the machine along the y direction, as well as the moving part containing the motors and pulleys for driving the machine in x direction. The x-axis describes the platform the radioactive source gets mounted on. It contains the source shielding interlock mechanism that allows the source to be electronically opened and closed.

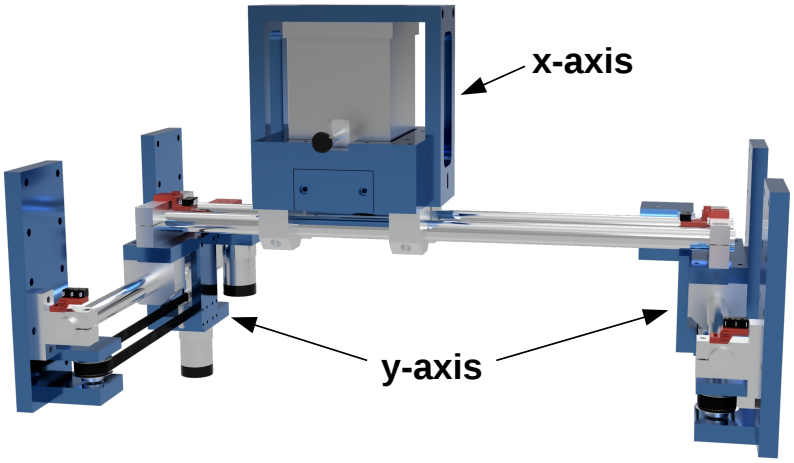


Figure 4.4.: Overview of the moving platform mechanism design.

4.1.1. y-Axis design

Because the y-Axis has to carry the weight of the whole x-axis in addition to the weight of the radiation source box it uses two geared DC motors. Despite providing double the torque of a single motor using two motors minimises the amount of radial force on the rails of the machine which in turn prevents it from locking up and improves the positioning accuracy. The disadvantage of the two motor solution is that it requires the motors to be precisely synchronous. This is taken care of by the software running on the microcontroller

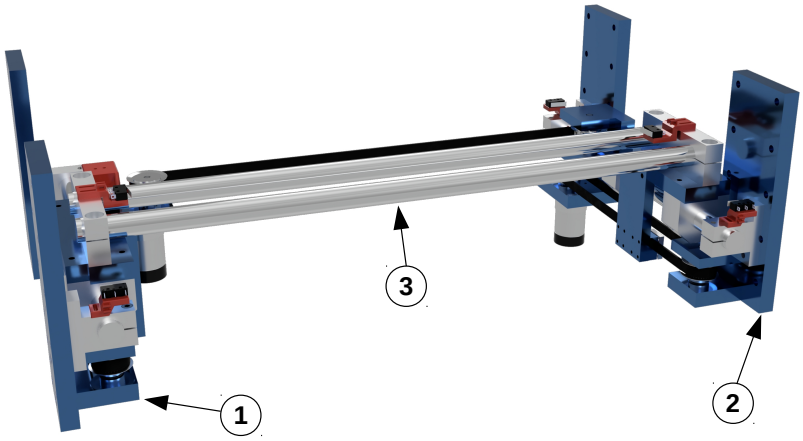


Figure 4.5.: Overview of the design of the y-Axis.

controlling the motors. This software will be discussed in more detail in section 4.2.

The two stationary parts of the y-Axis, indicated by the numbers (1) and (2) in Figure 4.5 each carry one of the motors and get bolted directly to the side plates of the containing box via 16 M8 bolts. Each of those bolts is rated to carry static loads of up to 3000 N[3], this ensures a solid and reliable mounting. While the platform is not motorised in the z-direction, multiple mounting holes in the y-Axis mounting plates still make it possible to adjust the height of the platform over a range of 132 mm in steps of 44 mm. Once the setup is fully assembled and adjusted to the height of the modules and their supporting infrastructure changing the height will not be necessary any more so an automated adjustment mechanism is not needed for this axis. To keep the radiation shielding of the box intact the mounting plates are made of stainless steel. Given that these plates are stationary the weight of 2 kg per plate does not cause any disadvantages or additional loads on the motors, rails or bearings. The moving parts of the y-axis, indicated by the number 3 in Figure 4.5, slide on two steel rails with four linear bearings. Since the source only needs to be moved once for each source scan the moving speed can be slow (6.5 $\frac{\text{cm}}{\text{s}}$) without causing significant delays in the testing procedure. This minimises the risk of fast movements exerting high forces on the machine and consequently damaging it or causing parts to come loose. Using linear ball bearings furthermore reduces friction in the system to a minimum which in turn reduces the amount of torque the motors need to apply to move the platform.

The moving part of the y-axis contains the motor and pulley assembly driving the x-axis as well as the two steel rails it slides on. In addition the y-axis contains two software limit

[3] `https://us.misumi-ec.com/pdf/press/us_12e_pr1271.pdf`

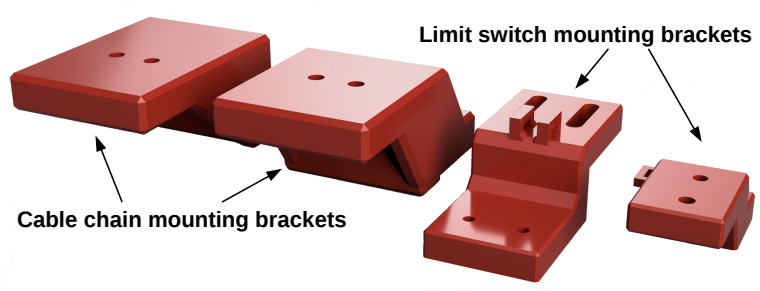


Figure 4.6.: 3D printed switch and cable chain mounting brackets.

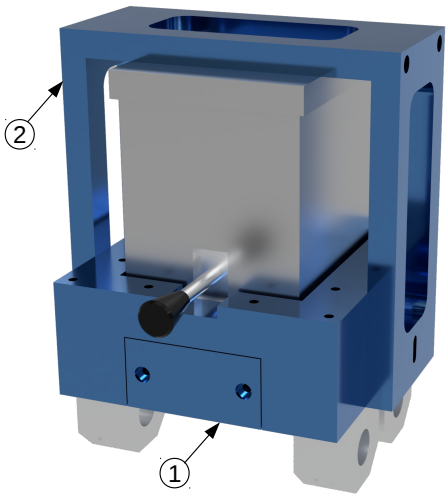
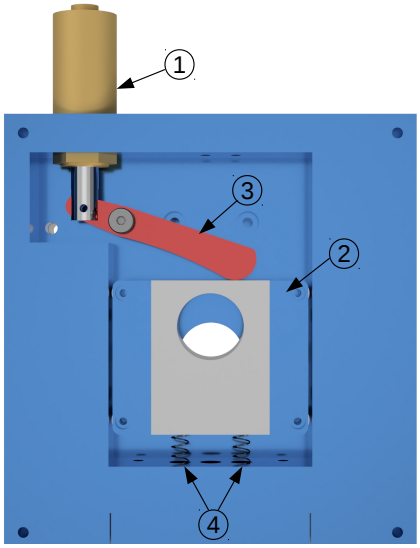
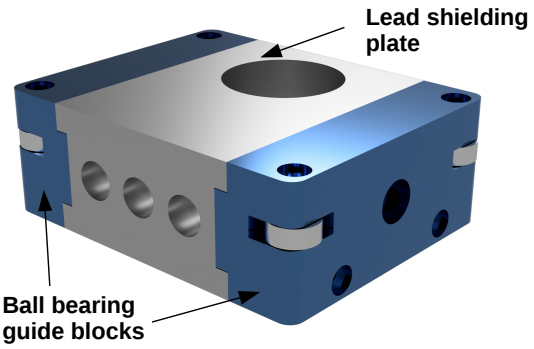


Figure 4.7.: Overview of the design of the x-axis.

switches enabling the homing of the machine as well as four emergency limit switches that physically cut the power to the motors and the source shielding solenoid in case one of the axes travels over its allowed range. This safety mechanism is discussed in more detail in section 4.2.2. Mounting those limit switches in the correct positions requires small mounting brackets that screw onto different parts of the y-Axis. Because of the small size and the complex shape of those parts they can not be machined from aluminium. Since they do not carry any loads they are designed to be 3D printed using PLA plastic. Figure 4.6 shows all the limit switch mounting brackets as well as two mounting plates needed to mount the cable chain for the x-Axis needed to make sure there are no loose wires close to the radioactive source.



(a) Components of the source shielding mechanism



(b) Source shielding shutter

Figure 4.8.: Overview of the source shielding interlock mechanism.

4.1.2. x-Axis design

The purpose of the x-axis assembly is to securely carry the source box and enable remote opening and closing of the radioactive source. The later task is accomplished by an electro-mechanical shutter mechanism discussed in more detail later. Similar to the y-axis the x-axis is mounted on two steel rails via four linear ball bearings and gets driven by the previously described belt and pulley mechanism mounted on the y-axis. An overview of the design is shown in Figure 4.7. To ensure the source box can not come loose or tip over inside the machine the x-axis uses an aluminium mounting clamp (2) fastened to the base via two thumb screws that can be fastened by hand.

The base part of the x-axis (1) contains the interlock mechanism necessary to safely open and close the radioactive source without having to open the doors of the containing box. The interlock mechanism shown in detail in Figure 4.8a uses a magnetic solenoid linear actuator (1) to drive the source shutter slider (2) back and forth via a lever (3) which in turn covers the radioactive source with a 3 cm thick lead plate in the closed position. The opened position is reached by pulling in the actuator by magnetising the solenoid coil which aligns the opening in the source box with a hole in the lead shielding plate. Once the coil gets demagnetised two springs (4) return the slider into its normally closed position. This can happen either through the software controls or a power cut caused by the limit switches or other safety mechanisms. To ensure the slider carrying the heavy lead shielding plate does not get locked up it uses eight small ball bearings as

guiding wheels. The design of the slider is shown in detail in Figure 4.8b.
As mentioned previously the solenoid coil is supplied by the same 12 VDC power supply that supplies the motors. Despite the datasheet[4] rating the coil for continuous operation at this voltage it does heat up significantly when this voltage is applied over an extended period of time. To mitigate this effect the coil is driven by a pulse width modulated signal with a duty cycle of 50 %. The full power is only applied for one second right after the interlock gets toggled to supply the necessary torque to pull the slider.
Correct functioning of the interlock mechanism is critical to the safety of the machine so care was taken to design it as robust and reliable as possible. Still regular testing and inspection of the mechanism are required to make sure it works as expected and there are no unexpected issues caused by wear on the moving parts or other faults.

4.2. Control electronics and software

As described in the previous section the machine uses three geared DC motors with digital encoders and an electromagnetic solenoid actuator to drive the platform and the interlock mechanism. In this section the control electronics and the software running on them will be described in detail.
The basis of the control system is an Arduino UNO[5] using the ATmega328 microcontroller with a plug in motor control shield[6]. The motor control shield contains four h-bridge motor drivers that can drive the DC motors as well as the solenoid. Both the motors and the solenoid operate on 12 VDC with an expected combined maximum current draw of two to three Ampere. The datasheet of the motor control shield specifies a maximum current draw of 1.2 A per channel which is enough to power the motors drawing an average of 300 mA when driving the platform at maximum speed. The Arduino gets connected to a computer via USB and uses a software serial port to communicate. It also receives the output of the encoders installed on the three motors and reads out the limit switches. Monitoring the encoders allows the controller to monitor all relative movements of the platform and the limit switches enable homing the machine to a known origin.

[4] http://www.produktinfo.conrad.com/datenblaetter/500000-524999/503346-da-01-en-ZYLINDERMAGNET_ITS_LZ_2560.pdf
[5] https://www.farnell.com/datasheets/1682209.pdf
[6] https://media.digikey.com/pdf/Data%20Sheets/DFRobot%20PDFs/DRI0039_Web.pdf

4.2.1. Motor control software

To keep track of the motor positions and drive the motors with acceleration and deceleration ramping the microcontroller uses a purpose made motor control library. The library is strongly interrupt driven and implements all necessary motor driving and readout functions. In addition it implements a storage structure in the EEPROM of the microcontroller to permanently store settings like the axis limits or the acceleration and deceleration slopes. This on the one hand makes it possible to only initialise the controller once for the specific machine it will be used on without the need to permanently store the machine information on the computer the device is interfacing with, making it possible to use the device on multiple computers without having to transfer the savefiles between them. On the other hand storing critical information like the maximum travel or speed directly on the device ensures that a communication error can not damage the machine because safety and validity checks for the send commands can be done directly on the controller. Furthermore it contains a mechanism to check the synchronicity of several motors and correct for misalignment or stop the machine if the misalignment overshoots a settable value.
With this interrupt based motor controller running in the background the main program loop implements the protocol to communicate over a software serial port using a custom instruction set. Each instruction send to the controller has a length of six bytes. The controller will reply to each instruction with a five byte long reply package. Each instruction package starts with a mode select byte, used to set what information is to be read out or set by the controller. This is followed by a byte to choose if the following command is a read or a write operation and two 16 Bit long integers making up the command itself. The following reply send back by the microcontroller starts with an acknowledge byte and two 16 Bit long integers. An overview of all the available instructions and query replies is given in Tables 4.2 and 4.3 and a graphical overview of the code running on the microcontroller is shown in Figure 4.9. While the serial protocol is very specific to the moving platform design used in this project, the motor controller library could be very easily adapted to drive many kinds of DC motor powered projects using up to four DC motors with encoders.
The performance of the motor controller software regarding precision, repeatability and reliability will be discussed in chapter 5.

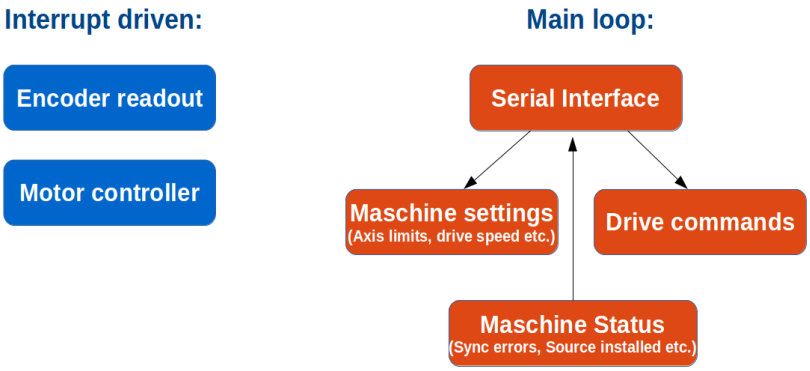


Figure 4.9.: Overview of the motor control software running on the Arduino.

Instruction	Mode byte	Command 1	Command 2
En-/ disable motors	0	0: disable, 1: enable	ignored
Home machine	1	ignored	ignored
Axis Limits	2	Limit x-Axis	Limit Y-axis
Speed Limit	3	Speed Limit	ignored
Slopes	4	Acceleration Slope	Deceleration Slope
Endstop Tolerance	5	Tolerance	ignored
Synchronisation	6	Sync Tolerance	Sync Error Limit
Position	7	Position x-Axis	Position y-Axis
Jog	8	Jog distance x-Axis	Jog distance y-Axis
Speed	9	Speed for all Motors	ignored
Interlock	10	0: disable, 1: enable	ignored

Table 4.2.: Instruction set for the serial communication with the motor controller.

Instruction	Reply 1	Reply 2
En-/ disable motors	0: disabled, 1: enabled	ignored
Home machine	0: uninitialised, 1: initialised	ignored
Axis Limits	Limit x-Axis	Limit Y-axis
Speed Limit	Speed Limit	ignored
Slopes	Acceleration Slope	Deceleration Slope
Endstop Tolerance	Tolerance	ignored
Synchronisation	Sync Tolerance	Sync Error Limit
Position	Position x-Axis	Position y-Axis
Jog	ignored	ignored
Speed	Speed for all Motors	ignored
Interlock	disabled, 1: enabled	ignored

Table 4.3.: Replies send by motor controller when queried.

4.2.2. Electronics and safety mechanisms

Since the machine is designed to carry a highly radioactive source (either Strontium or Americium) radiation safety is one of the most important design concerns. The machine needs to be designed to securely prevent the user from accidentally getting exposed to radiation. The safety mechanisms need to be able to safely handle faults of the machine itself, like a power cut or a software error as well as mistakes made by the user, like for example opening the doors of the steel box without first closing the source shutter.
Similar to the old design there is a red warning light on top of the box indicating to the user wether the radioactive source is installed. To guarantee safe operation the machine is equipped with an array of sensor switches consisting of a total of four emergency limit switches that cut the power going to the motors and the interlock solenoid in case the machine drives over its desired range. Those emergency limit switches also trigger a relay that cuts the power and latches into this disconnected state until a reset button is manually pressed. This is to ensure that the platform can not unexpectedly start moving again after it has hit one of the emergency stops.

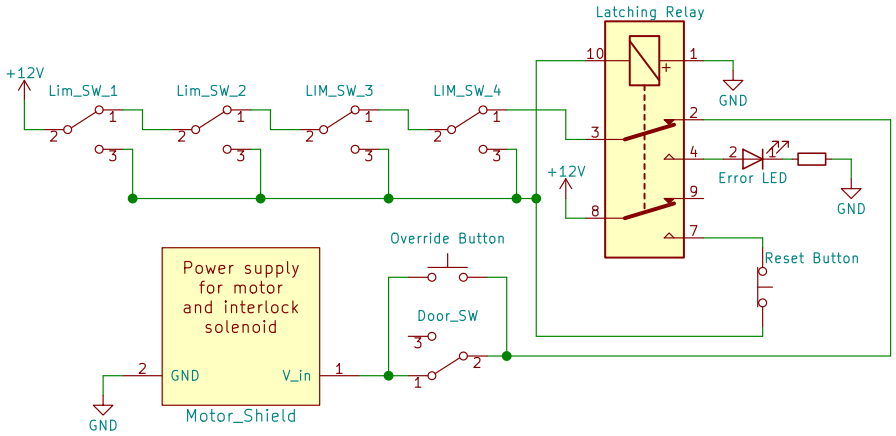


Figure 4.10.: Circuit diagram of the safety interlock mechanism.

To prevent injuries caused by the platform moving while the operator is still working on it, the power also gets cut by interlock switches installed on the two doors of the steel box. These switches are not connected to the latching relay so they only temporarily disable the motors and close the source shielding shutter while the doors are open. The circuit diagram for all the interlock switches is shown in Figure 4.10 and an overview of possible faults and following machine behaviour is given in Table 4.4. With this interlock mechanism it is impossible for the user to ever unintentionally get in contact with the radioactive source.

Fault	Motor Power	Source shutter	Indicators
Power cut	cut off	closed	nothing
Door opened	cut off	closed	nothing
Limit switch hit	cut off (latching)	closed (latching)	Error LED
Motors out of sync	stopped (software)	closed (software)	Error message

Table 4.4.: Overview of fault handling for different scenarios.

To connect all the motors and switches to the Arduino, a hand soldered circuit board is used. It contains a mount for the Arduino, connectors for all the motors, switches and the solenoid as well as the relay. The power supply currently used is a lab power supply able to supply up to 3 A of current at the operating voltage of 12VDC.

4.3. Software design

As described previously the microcontroller controlling and supervising the moving platform is not a standalone controller but needs to communicate to a computer over a software serial port running at a baud rate of 115200 Bd. The high baud rate is necessary so the motor control loop running on the microcontroller does not get interrupted for too long while the communication takes place. The serial protocol is described in detail in section 4.2.1.
Once the machine is used to test modules as intended it can be controlled over the PrimList interface in the STcontrol software. STcontrol is based on the ATLAS PixLib package and provides a graphical user interface to control and readout the pixel modules and perform the tuning and testing procedures. This can either be done directly over a software serial port or remotely using a TCP protocol as described in section 4.3.2. For troubleshooting or in case the machine ever gets used for other purposes a standalone program with a graphical user interface (GUI) was developed as described in section 4.3.1. All the source code used in this project can be found in the GitLab repositories listed in Table A.2.

4.3.1. Pixel Box Controller Software

The so called Pixel Box Controller software (PBC) is a standalone control interface designed to interface with the Arduino running the motor control software described in section 4.2.1. It is written in C++ and uses the *QT* framework to implement the GUI as well as the serial and TCP communication. The GUI allows to toggle between two main modes. The direct control mode gives access to all features related to moving the platform. The TCP mode sets up a TCP Server over which the platform subsequently can

Parameter	Functional description
x-Axis Limit	Maximum travel along the x-axis
y-Axis Limit	Maximum travel along the x-axis
Maximum Speed	Maximum speed of the platform
Acceleration Slope	Number of steps the machine takes to accelerate to the set speed
Deceleration Slope	Number of steps the machine takes to decelerate to the set speed
Endposition Tolerance	Number of steps the controller allows as tolerance between the set and the actual motor position
Synchronisation Tolerance	Number of steps two motors in a sync group can be out of sync without the controller correcting
Out of sync break	Number of steps that two motors in a sync group can be out of sync before controller stops the machine and throws an error

Table 4.5.: Overview of all the settable machine parameters.

be remotely controlled. Those two modes are described in detail in the following sections.

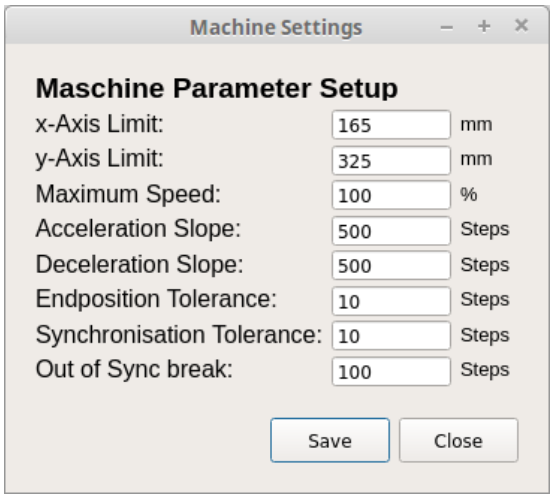


Figure 4.11.: Available Machine settings.

In addition to these two main modes a machine parameter settings window can be accessed over the menu bar. This settings window, shown in Figure 4.11, allows to set and read out all available machine parameters. An overview of those parameters with a functional description is given in Table 4.5. The machine parameters do not get stored in a savefile on the computer the PBC is running on but directly in the microcontrollers EEPROM. This gives the advantage that connecting the controller to different computers will not result in loosing or altering important machine parameters.

Direct control mode

For testing and assembly as well as for possible future use cases of the machine that are not using the STcontrol software it is desirable to have a simple and clear control interface that gives access to all the functions of the machine.
The direct control mode of the PBC software enables the user to drive the machine to a desired position given in x-y-coordinates as well as to jog the machine in x or y direction with a settable step size. The speed those movements are executed at is settable via a slider. The software also continuously reads out the current platform position and displays it in the top left corner. Furthermore there are a number of push buttons that enable the user to home the machine, enable and disable the motors and to toggle the interlock solenoid. The GUI of the direct control mode is shown in Figure 4.12.

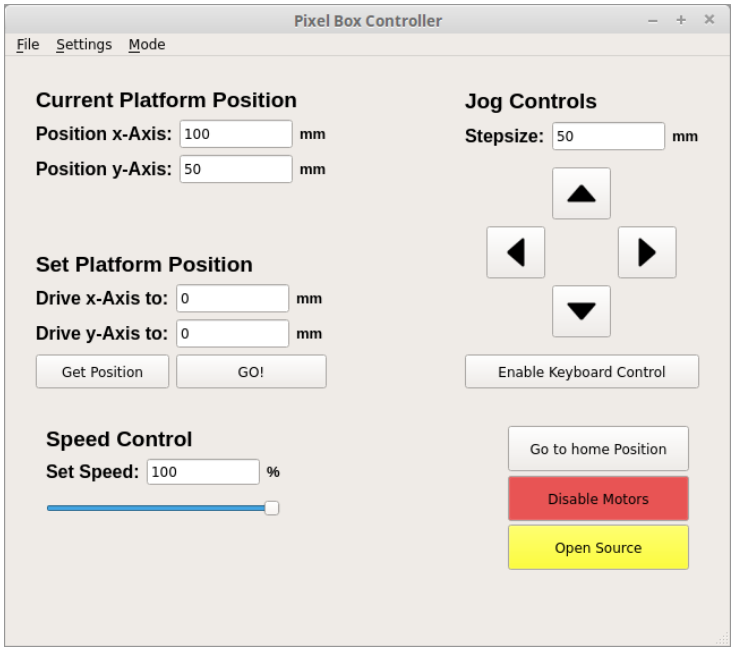


Figure 4.12.: GUI of the PBC software in direct control mode.

Since the UI is designed to be as simple and safe to use as possible care was taken to not allow any inputs that could damage the machine, like driving the motors before a successful homing procedure was executed. This is done by enabling and disabling input buttons depending on wether using them is safe given the current state of the machine.

TCP mode

Testing multiple pixel modules at once requires running multiple instances of STcontrol on multiple computers, each controlling and monitoring a single module. For an efficient

testing procedure it is thus necessary to remotely control the platform from multiple computers at a time. This is not possible using the direct control mode.
The TCP mode integrated into the PBC enables this remote controlling. It sets up a TCP server using the *QtNetwork* library on TCP port 15555. The communication protocol implemented by the TCP mode is very similar to the one used to communicate to the motor controller directly over software serial. The instruction send to the server by the module controller starts with a mode select byte, followed by two 16 Bit command integers. The server replies to an instruction by resending the mode byte followed by two 16 Bit reply integers.

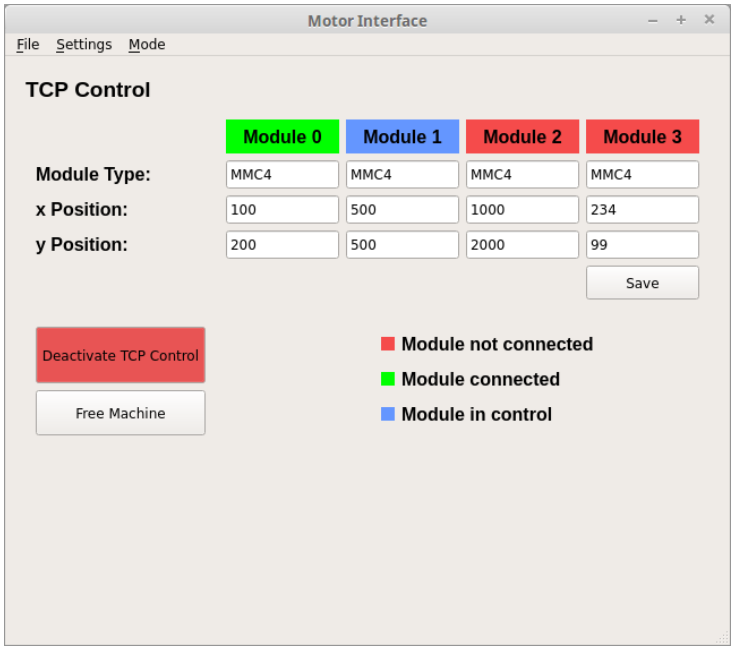


Figure 4.13.: TCP Interface in the PBC software.

The software allows a settable number of modules to connect to the server, this number by default is set to four since this is the number of modules that most likely will be installed in the setup. It can be changed via the *NUM_MODULES* macro in the *motor_interface.hpp* header file.
To initialise the communication a module first needs to send its module ID to the server. Via the *get control* command a module can then take over control over the platform allowing it to move it around and open and close the source shutter through the *drive axis* and *interlock* commands, respectively. Using the *drive to module* command the platform can be driven to the modules position, settable in the TCP mode UI for each module. The set module positions get stored in a savefile so they only need to be set once for a given setup. Once a module controller finishes its testing procedure and does not require

Command	Mode	Cmd 1	Cmd 2	Reply 1	Reply 2
status	0	0: occupation status 1: module connected	ignored ignored	occupation status connection status	
get control	1	0: get control 1: set free	ignored	setting successful	
drive to module	2	ignored	ignored	driving successful	
drive axis	3	x-pos	y-pos	driving successful	
get position	4	ignored	ignored	x-pos	y-pos
interlock	5	0: close 1: open	ignored	setting successful	

Table 4.6.: TCP communication protocol for TCP communication with PBC.

access to the platform any more it can give it free again using the *free machine* command to allow other module controllers to take over control.
The TCP mode software keeps track of which modules are connected and in control and indicates it by colour coding the modules as visible in Figure 4.13. Control commands send by modules that are not currently in control over the machine get ignored. In case one of the module controllers fails to free the machine after it takes control the *Free Machine* button in the PBC GUI makes it possible to manually free the machine and allow other modules to take over control again.

TCP module controller

The main purpose of the TCP mode in the PBC software is to be used in conjunction with the STcontrol software, allowing multiple pixel modules connected to multiple computers to be tested at once. The *TCP module controller* is a standalone program that uses the same *TCPClient* class as the STcontrol TCP module described in more detail later. The GUI, as shown in Figure 4.14, implements all the control features that are available using TCP communication. The module controller software is mainly useful for troubleshooting and testing since using the significantly more complex STcontrol software can be very time consuming. Should the machine ever be used outside of the initial project of testing pixel modules the controller also provides a simple to use interface to remotely control the machine without all the overhead of STcontrol.

4.3.2. STcontrol integration

The USBpix system used to read out and test the FE-I4 pixel modules uses the STcontrol software to interface with the modules and provide a GUI. The GUI is implemented using *QT* and *Root*. STcontrol already provides an interface to communicate with external

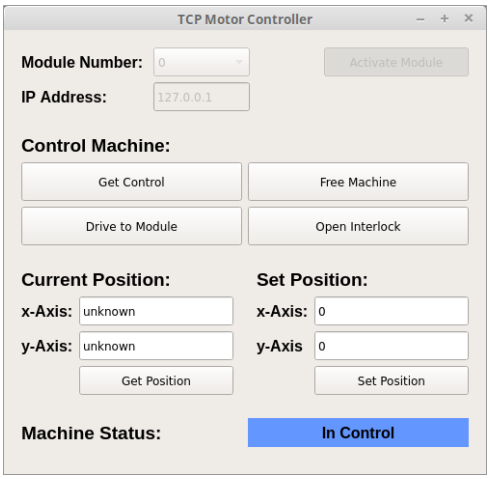


Figure 4.14.: TCP controller interface for remote control.

devices like power supplies and all kinds of measurement instruments over a variety of different protocols, for example USB, GPIB and RS232. Those external devices are called *Detector Control Systems* or DCSs for short. STcontrol also provides the option to send commands to DCS devices inside the PrimList module. The PrimList module is intended to quickly set up and execute simple automated test sequences and thus is very useful for the planned module testing.
As explained previously there are two possible ways of how STcontrol can communicate to the Arduino that controls the machine. The first way is direct communication over a software serial port using the protocol described in Tables 4.2 and 4.3. The advantage of this control mode is that it requires no additional software to be running. For some use cases it will be necessary to control the platform from multiple instances of STcontrol simultaneously. The TCP communication mode makes this possible. In this mode STcontrol does not interface with the Arduino directly but through the TCP server hosted by the PBC software.

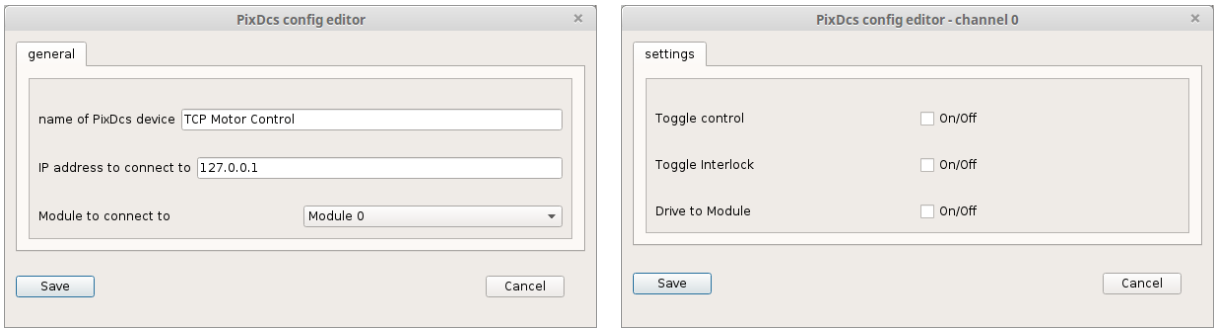
Software serial control

The STcontrol software already provides all the necessary functions for serial communication using the RS232 protocol in the *PixRs232Device* class. This class uses the functions defined in *ComTools.h* to directly interface with the serial ports of the computer. Some of those functions were extended to enable using software defined serial ports in addition to the hardware serial ports of the computer. Furthermore the *PixRs232Device* class and its functions were extended to support communications with the motor control Arduino. The motor controller has the device type *position* and uses three DCS channels, the first two being the x- and y-axis and the third one being the interlock shutter. Because it uses

a software serial port the *ACM0* port needs to be selected. Care should be taken that this port is not used by another device plugged into the computer.
The position of the x- and y-axis can be set in steps of 1 mm and the shutter gets opened and closed by setting the position to 1 or 0, respectively. The speed of the two axes can be adjusted from $0-100$ %, setting the speed of the shutter has no effect. By initialising the DCS item in STcontrol the machine automatically gets homed and the motors enabled so the machine is fully initialised and ready to use.

TCP control

As opposed to the RS232 interface discussed in the last chapter, STcontrol provides no implementation of a TCP interface. Because the PBC as well as the TCP module controller use the *QtNetwork* library it was decided to also use it for the STcontrol TCP mode to get it working under the given time constraints. However this is not a long term solution since the *PixLib* classes are not supposed to use any *QT* modules.
The TCP interface itself is implemented in the *Client* class contained in the *TCPClient.h* and *TCPClient.cxx* files. In addition to that the *TCPPixDcs* and *TCPPixDcsChan* classes were added to enable a TCP object to be added as a DCS device.



(a) DCS initialisation dialog ***(b)*** DCS control dialog

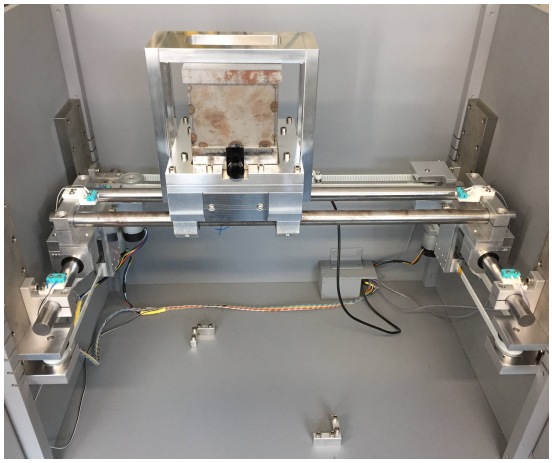
Figure 4.15.: DCS interface of the TCP control mode in STcontrol.

The motor controller can then be set up as a one channel TCP object through the *Add DCS item* dialog. In the initialisation dialog, shown in Figure 4.15a, the IP address of the TCP server hosted by the PBC and the module ID need to be selected. In case the PBC instance is running locally on the computer the default IP address is 127.0.0.1. The STcontrol software does not need to store any information about the position of the modules since this is taken care of by the PBCs TCP mode. This eliminates the need for additional savefile structures to be added to the STcontrol software.
To keep the user interface as clear as possible only the basic functionalities of the TCP

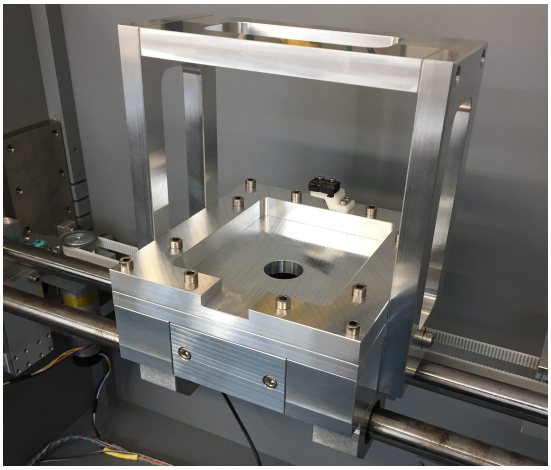
protocol have been implemented. The control dialog, shown in Figure 4.15b, allows the STcontrol instance to take control over the machine and set it free again, as well as to drive it to the modules position and to toggle the source shutter.

5. Assembly and testing

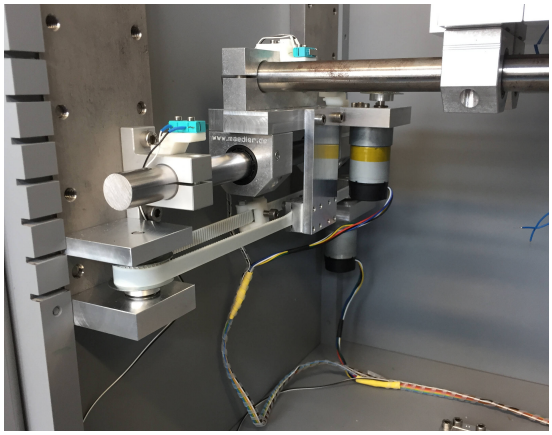
As mentioned before in Chapter 4 large parts of the machine were manufactured by the *Zentralwerkstatt*. After receiving those parts and wiring up the electronics the machine was assembled inside the steel box for testing and characterisation. Figure 5.1 shows a collection of images of the finished assembly.



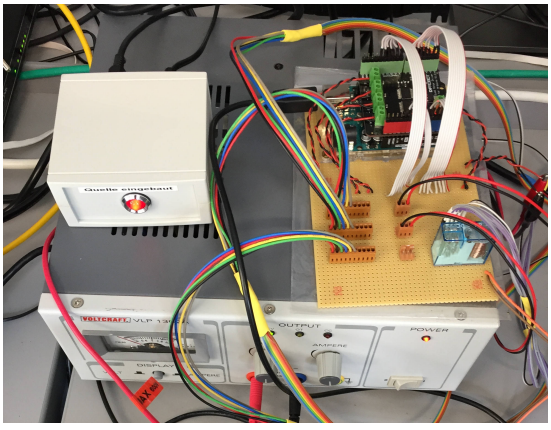
(a) Overview of both axis



(b) Detailed view of the x-axis



(c) Detailed view of the y-axis



(d) Power supply and control electronics

Figure 5.1.: Current status of the assembly installed in the steel box.

Since the machine will later carry a highly radioactive source, testing the reliability of the safety mechanisms was a priority. Furthermore several calibration parameters, such as

Parameter	Description	Default value
steps_per_rev[4]	Steps per motor revolution for for the four motor channels	$\{2080, 2080, 2675, 0\}$
pulley_diameter	Effective diameter (in mm) of the pulley connected to each motor	28.36
solenoid_delay	Time the solenoid is operated at 100 % duty cycle (in ms)	1000
solenoid_duty_cycle	Duty cycle of the PWM signal the solenoid is continuously driven with	200

Table 5.1.: Overview of the calibration parameters settable in the motor controller code.

the conversion factor for translating the encoder steps registered by the motor controller into the actual distance travelled by the platform were determined. An overview of the default values for all those parameters is given in Table 5.1. All those parameters can be set in the source code of the motor controller software running on the Arduino.
The emergency limit switches were tested by purposefully driving the machine over its designed travel range and thus letting it hit the limit switches 20 times. The relay switching off the motor and source shutter successfully cut the power in every single trial. Manually moving the platform away from the limit switch did in none of those trials reconnect the power, so the latching mechanism works as intended. Testing the reliability of the interlock shutter mechanism was not possible during the course of this thesis because the manufacturing of some of the required parts was not completed before the submission date.

To calibrate the conversion factor that translates the registered encoder steps into travelled distance in mm a pair of digital callipers was fixed to the stationary part of each axis. Subsequently the machine was commanded to drive 5000 steps in one direction. This was repeated ten times and the average travelled distance was noted. From those values the conversion factor was calculated. To verify the determined conversion factor the machine was then commanded to move 100 mm along each axis and the actually travelled distance was measured. Ten trials for each axis resulted in an average deviation of 0.12 ± 0.05 mm from the commanded position for the y-axis and 0.14 ± 0.05 mm for the x-axis. The individual trials are summarised in Figure 5.2.

To check if the machine correctly registers all pulses emitted by the encoders on the three motors the machine was driven back and forth ten times over its full travel along both axis. If the encoder readout system works without errors the machine is expected to end up in the exact same spot after the movement is completed. Any deviations would hint at the encoder readout either missing or counting extra steps, this could either be

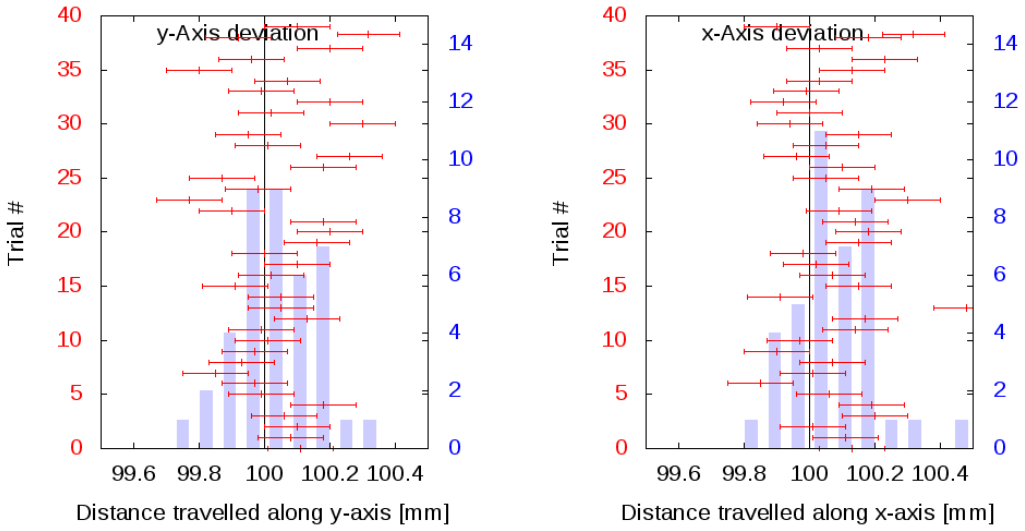


Figure 5.2.: Deviation of the actually travelled distance from the demanded one.

due to the encoders emitting a noisy signal or faults in the readout software. Driving the machine as described before and measuring the deviation between start and end position resulted in an average deviation of 0.20 ± 0.08 mm for the y-axis and 0.40 ± 0.08 mm for the x-axis over ten trials.

In addition to these motor control related inaccuracies mechanical backlash in the machine, caused for example by the flexibility in the belts or clearances between the moving parts, will also contribute to errors in the positioning of the platform. The amount of backlash in both the x- and y-axis was tested by pushing on the axis in both directions and measuring the deviation in the position. This was done again using digital callipers fixed to the stationary parts of the axis. With the belt tensioned appropriately the measured backlash was 0.65 ± 0.05 mm for the y-axis and 1.10 ± 0.05 mm for the x-axis. The backlash is bigger for the x-Axis because it uses a single belt as opposed to the two used by the y-Axis. The majority of this backlash is caused by the pulleys moving on the motor shafts and not the flexibility of the belts. This could significantly be reduced by exchanging the sharply tipped set screws used at the moment by flat tipped ones.

In conclusion, the positioning accuracy of the machine, taking into account all possible reasons for inaccuracies can be estimated to be roughly 1 mm. For module testing the acceptable positioning error is at least an order of magnitude bigger, since the procedure only requires the whole sensor to get irradiated and does not depend on precise positioning of the source.

During this testing the machine received the drive commands over the direct control

mode of the PBC software. Throughout the entire testing period no wrong commands were sent and the connection to the motor controller Arduino was never lost or interrupted. While more extensive testing of the software is necessary this indicates that there are no major errors in the implementation of the control interface.
Testing the TCP mode by connecting two STcontrol instances to the TCP server in the PBC software and sending all the available commands in the interface 10 times each was successful as well. The machine executed all the commands correctly and no connection errors were encountered in the testing process.

6. Summary and Outlook

The manufacturing and assembly of the whole setup was not completely finished during the course of this thesis, making complete testing and validation of the design impossible. This makes it very difficult to judge which parts of the system need to be improved. Nonetheless there are some lessons that can be learned from the design process and the assembly of the machine.

The approach taken to design the machine relied strongly on custom made components and custom software solutions. As an example instead of buying motors and corresponding motor drivers with commercially available control software, simple DC motors with a purpose made motor driver running on an Arduino were used. This brings the advantage that the system is fully customisable and can be tailored precisely to the designated application at a fraction of the cost of a commercially available system. This comes at the expense of a very long development time for the software and the lack of technical support that the respective manufacturer of commercial parts can provide.

The same arguments can be made for custom made hardware parts, that in addition carry the risk of containing design faults that make re-manufacturing necessary, adding additional cost and time. The big advantage of custom hardware parts is that they can be tailored to very specific needs, as an example including the source shutter interlock mechanism would have been a huge challenge with commercially available parts whereas it was relatively straight forward in the custom design because the whole x-Axis could be designed based around this mechanism.

Given the time constraints, this design did not utilise any of the simulation features the CAD package provides, this could have helped identify possible weak points or other structural faults. To avoid those weak points which could cause the machine to fail the parts in the design tend to be over-dimensioned for the loads they actually carry, this obviously increases the weight of the machine significantly and could have been avoided using simulations. Since the machine does not move fast the heavy weight is not a major concern and does not cause any disadvantages.

It remains to be seen how the moving platform actually performs during module testing since the modules and their supporting infrastructure were not ready to be installed in

the setup in time, but the results from preliminary testing look very promising as no major design flaws were encountered and the machine performed without any errors for the whole testing period, adding up to about two to three hours of machine runtime. The only component not tested is the source shutter interlock.
In contrast to the hardware the development of the software driving and controlling the machine is fully completed. The TCP remote control mode enables the desired parallelisation of the testing procedure and the motor control software running on the microcontroller allows safe and very precise positioning of the source as discussed in Chapter 5. The next steps in getting the machine ready for module testing should thus be to get the shutter mechanism assembled and thoroughly tested and consequently run the machine with the source installed to map out the area that can be irradiated. Some minor additions to the interlock electronics, such as installing the switches on the doors need to be done before safely being able to run the machine with an active source installed.

A. List of used hardware and links to source code repositories

Part description	Amount	Supplier	Link
Rail Clamp	4	Mädler	http://www.maedler.de/Article/64640620
Linear bearing	8	Mädler	http://www.maedler.de/Article/64652020
Rail	2 m	Mädler	http://www.maedler.de/Article/64752000
Double Rail Clamp	2	Mädler	http://www.maedler.de/Article/64640220
Pulley	6	Mädler	http://www.maedler.de/Article/16033600
Belt	3 m	Mädler	http://www.maedler.de/Article/16070000
Belt clamp	6	Mädler	http://www.maedler.de/Article/16079900
6 mm ball bearing	6	Mädler	http://www.maedler.de/Article/626-ZZ-MAE
3 mm ball bearing	8	Mädler	http://www.maedler.de/Article/623-ZZ-MAE
Springs	3	Gutekunst	https://www.federnshop.com/de/produkte/druckfedern/vd-063q.html
Cable chain	1	Conrad	https://www.conrad.de/de/energiefuehrungskette-lappkabel-silvyn-chain-light-sr200-61210381-fuer-kleinste-biegeradien-604595.html
Cable chain mount	1	Conrad	https://www.conrad.de/de/anschlusselement-lappkabel-silvyn-an-200012-k-61211271-604643.html?sc.ref=Product%20Details
Cable tube	3 m	Conrad	https://www.conrad.de/de/spiralschlauch-2-bis-15-mm-natur-sb-12-e-pb-fastener-meterware-540767.html

Table A.1.: Overview of all commercially available parts used in the design

Description	Link
Pixel Box Controller	`https://gitlab.cern.ch/tihuser/Pixel_Box_Controller`
TCP Mode Controller	`https://gitlab.cern.ch/tihuser/TCP_Motor_Controller`
CAD Files teststand	`https://gitlab.cern.ch/tihuser/Pixel_Module_Teststand_CAD`

Table A.2.: GitLab repositories for the source code and CAD files

Bibliography

[1] R. Assmann, M. Lamont, S. Myers, *A brief history of the LEP collider*, Nucl. Phys. Proc. Suppl. **109B**, 17 (2002)

[2] V. Shiltsev, *Achievements and Lessons from Tevatron*, Conf. Proc. **C110904**, 903 (2011)

[3] R. P. Walker, *Synchrotron radiation*, Conf. Proc. **C9209071**, 437 (1992)

[4] L. Evans, P. Bryant, *LHC Machine*, JINST **3**, S08001 (2008)

[5] ATLAS collaboration, *Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC*, Phys. Lett. **B716**, 1 (2012)

[6] S. Chatrchyan, et al. (CMS), *Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC*, Phys. Lett. **B716**, 30 (2012)

[7] R. Bruce, et al., *LHC Run 2: Results and challenges*, in *Proceedings, 57th ICFA Advanced Beam Dynamics Workshop on High-Intensity and High-Brightness Hadron Beams (HB2016)* (2016)

[8] *ATLAS experiment - Luminosity Public Results Run 2*, `https://twiki.cern.ch/twiki/bin/view/AtlasPublic/LuminosityPublicResultsRun2#Luminosity_summary_plots_for_201`, accessed: 2017-03-20

[9] G. Apollinari, et al., *High-Luminosity Large Hadron Collider (HL-LHC) : Preliminary Design Report* (2015)

[10] ATLAS collaboration, *Letter of Intent for the Phase-II Upgrade of the ATLAS Experiment*, Technical Report ATL-COM-UPGRADE-2012-040 (2012)

[11] ATLAS collaboration, *The ATLAS Experiment at the CERN Large Hadron Collider*, JINST **3**, S08003 (2008)

[12] A. R. Martinez (ATLAS), *The Run-2 ATLAS Trigger System*, J. Phys. Conf. Ser. **762(1)**, 012003 (2016)

[13] ATLAS collaboration, *The ATLAS silicon pixel sensors*, Nucl. Instrum. Meth. **A456**, 217 (2001)

[14] ATLAS collaboration, *ATLAS pixel detector electronics and sensors*, JINST **3**, P07007 (2008)

[15] N. Wermes, *Pixel Detectors for Charged Particles*, Nucl. Instrum. Meth. **A604**, 370 (2009)

[16] A. La Rosa, *The ATLAS Insertable B-Layer: from construction to operation*, JINST **11(12)**, C12036 (2016)

Acknowlegments

First and foremost I would like to thank Prof. Dr. Quadt for giving me the opportunity to write this thesis in his group and his support during the project. In addition I would like to thank Priv.Doz. Dr. Jörn Große-Knetter for agreeing to be the second referee for my thesis as well as for his support and guidance, especially by helping me out with fixing my bugs in the STcontrol software.
I would also like to thank Helge Beck for greatly helping me out all throughout the project and for proofreading my thesis. Furthermore I would like to thank Peter Arnsberger and Reinhard Mielke from the workshops in our Institute as well as Christof Schmidt and Alexander Gehrt from the *Zentralwerkstatt* for manufacturing all the parts I needed as well as for their very helpful advice on the design.

Erklärung nach §13(9) der Prüfungsordnung für den Bachelor-Studiengang Physik und den Master-Studiengang Physik an der Universität Göttingen:

Hiermit erkläre ich, dass ich diese Abschlussarbeit selbständig verfasst habe, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe und alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten Schriften entnommen wurden, als solche kenntlich gemacht habe.
Darüberhinaus erkläre ich, dass diese Abschlussarbeit nicht, auch nicht auszugsweise, im Rahmen einer nichtbestandenen Prüfung an dieser oder einer anderen Hochschule eingereicht wurde.

Göttingen, den 14. September 2018

(Timo Hüser)