

Muon Detector Prototype



Presented by:
Paulus Jacobus de Bruyn


Prepared for:
Dr. Jane Wyngaard
Department of Electrical and Electronics Engineering
University of Cape Town

Submitted to the Department of Electrical Engineering at the University of Cape Town
in partial fulfilment of the academic requirements for a Bachelor of Science degree in
Electrical and Computer Engineering

7 November 2021

Declaration

1. I know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is one's own.
2. I have used the IEEE convention for citation and referencing. Each contribution to, and quotation in, this report from the work(s) of other people has been attributed, and has been cited and referenced.
3. This report is my own work.
4. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as their own work or part thereof.

Signature:.....
P.J. de Bruyn

Date: 7 November 2021

Abstract

Muon radiography has been used since the 1950s to create scans of large and complex objects such as mountains, volcanoes, and pyramids. In this project, a system that can detect muons will be created in order to muon-image Devil's Peak. A previous final year engineering student built a muon detector subsystem. The purpose of this project is to create the microcontroller subsystem to which the muon detector subsystem can connect that can send out data about the muon detection to a distant computer using LoRa. The data that will be sent is the following: longitude and latitude coordinates; accelerometer data; temperature and humidity; and voltage levels over time measured by the peak detector.

Contents

1	Introduction	1
1.1	Background to the study	1
1.2	Objectives of the study	2
1.2.1	Problems to be investigated	2
1.2.2	Purpose of the study	2
1.3	Scope and limitations	2
1.4	Plan of development	2
2	Literature Review	4
2.1	Muons	4
2.1.1	Introduction to muons	4
2.1.2	Muon radiography	4
2.1.3	Operation of a muon detector	5
2.1.4	Use cases	6
2.2	The Internet of Things (IoT)	7

2.2.1	Introduction	7
2.2.2	History of IoT	8
2.2.3	The use of IoT	9
2.3	Edge computing	10
2.3.1	Introduction to edge computing	10
2.3.2	History of edge computing	10
2.3.3	Benefits of edge computing	10
2.3.4	Challenges of edge computing	11
2.4	Communication technologies	12
2.4.1	Introduction	12
2.4.2	Cellular networks	13
2.4.3	Sigfox	13
2.4.4	LoRa	14
3	Methodology	15
3.1	Problem definition	16
3.1.1	Previous work done on the project	16
3.1.2	Requirements for this project	19
3.2	Practical aspect	22
3.2.1	Initial component selection and design	22
3.2.2	PCB design	22

3.2.3	Coding design	23
3.2.4	Coding implementation	23
3.3	Report	23
4	Design	24
4.1	PCB Design	24
4.1.1	Daughter board	24
4.1.2	Main PCB	28
4.2	Coding design	35
5	Implementation	37
5.1	Schematic	37
5.1.1	Power supply	38
5.1.2	Peripherals	39
5.1.3	IO expanders	40
5.1.4	LoRa module	42
5.1.5	Microcontroller	44
5.1.6	NRST and BOOT	45
5.1.7	ST-Link	46
5.1.8	Oscillators	46
5.1.9	Interface with the daughter board	47

5.1.10	Other pin headers	49
5.2	PCB layout	51
5.2.1	Microcontroller	51
5.2.2	LoRa	52
5.2.3	Power supply	53
5.2.4	Accelerometer and temperature sensor	54
5.2.5	IO expanders	55
5.2.6	VDD	56
5.2.7	Issues with initial PCB designs	57
5.3	Timing	58
5.3.1	Timing of the daughter board	59
5.3.2	Timing of the ADCs	59
5.3.3	Timing of the IO expanders	60
5.3.4	Timing of the accelerometer	61
5.3.5	Timing of the temperature and humidity sensor	62
5.3.6	Timing of the GPS module	62
5.3.7	Timing of the LoRa communication	63
5.3.8	Timing conclusion	63
6	Results	65

7	Conclusion	67
8	Recommendations	68
A	Additional files and schematics	72
A.1	GitHub link	72
A.2	Full PCB layout	72
B	Addenda	77
B.1	Ethics form	77

List of Figures

3.1	Flow chart of the methodology	16
3.2	Layout of scintillator planes [24]	17
3.3	Layout of the muon detector subsystem [24]	18
4.1	Diagram of the daughter board and its interface with the microcontroller subsystem	25
4.2	Pulse shaping circuit output [24]	27
4.3	Microcontroller subsystem design	29
4.4	Attempted STM32WL55JC routing	32
4.5	FSM of the control program	35
5.1	Power supply of the PCB	38
5.2	Peripherals on the PCB	39
5.3	IO expanders on the PCB	40
5.4	Schematic of the LoRa module	42
5.5	Connections of the microcontroller	44
5.6	NRST and BOOT connections	45

5.7	ST-Link header	46
5.8	Oscillators for the microcontroller	46
5.9	JST GH header footprint	48
5.10	3D view of the JST GH header	48
5.11	19-pin 2.54 mm pitch male header appearance on the PCB	49
5.12	8-pin 2.54 mm pitch male header appearance on the PCB	50
5.13	Microcontroller routing on the PCB	51
5.14	LoRa routing on the PCB	52
5.15	Power supply routing on the PCB	53
5.16	Accelerometer and temperature sensor routing on the PCB	54
5.17	IO expanders routing on the PCB	55
5.18	3.3V supply across the PCB	56
5.19	Acid trap formed by copper tracks	57
5.20	An initial design with poor track spacing	58
5.21	Timing diagram of the daughter board	59
5.22	Final timing diagram	64
A.1	Front side of the PCB	73
A.2	Back side of the PCB	74
A.3	Front side of the PCB without copper fill	75
A.4	3D view of the PCB	76

List of Tables

3.1	Functional requirements of the system	21
3.2	Non-functional requirements of the system	22
5.1	Address selection of the IO expanders	40
5.2	10-pin JST GH header pin description	48
5.3	9-pin JST GH header pin description	49
5.4	3-pin JST GH header pin description	49
5.5	19-pin 2.54 mm pitch male header pin description	50
5.6	8-pin 2.54 mm pitch male header pin description	51

Nomenclature

BGA Ball Grid Array

EMC Electromagnetic Capability

FSM Finite State Machine

PCB Printed Circuit Board

RTC Real Time Clock

SiPM Silicon Photomultiplier

SPI Serial Peripheral Interface

SWD Serial Wire Debug

Chapter 1

Introduction

1.1 Background to the study

Muons are elementary particles that are part of the lepton group and were discovered by American physicists Carl David Anderson and Seth Neddermeyer in 1936 [1]. It was later found that muons provide a good way to image the internal structure of large objects such as mountains. This is because of a property muons have where they change direction and lose a certain amount of energy after they penetrated an object. This information can be used to find the size and type of material the muon travelled through. Muons are a good method for imaging because they occur naturally in the environment and do not have to be produced by an external source like other imaging techniques such as using x-rays or gamma rays. Muon radiography is also a much safer way of imaging that does not have the potential to expose humans to stray radiation as with other methods [2]. Additionally, the foremost reason muons are efficient for imaging is because of their deep penetrative power. Muons can travel through thick layers of rock, concrete, and iron which usually block other particles that are used for detection such as x-rays, gamma rays, neutrons, alpha particles, and beta particles [3].

1.2 Objectives of the study

1.2.1 Problems to be investigated

The main problem that will be investigated is how to design a PCB with a microcontroller that can communicate with the muon detector subsystem, GPS module, accelerometer, and temperature and humidity sensor, and transmit information from those devices via LoRa to a remote computer. This information will then be used for muon imaging.

1.2.2 Purpose of the study

This study will produce a way to detect muons and measure their direction of motion. This will allow one to muon image large objects such as Devil's Peak, which is the case in this study. This way, geological research can be done on Devil's Peak such as finding hidden tunnels and caves.

1.3 Scope and limitations

This study will only include the following: the design of the PCB of the microcontroller subsystem; the coding of the microcontroller to interface with the muon detector, GPS module, accelerometer, and temperature and humidity sensor; and then finally the sending of the data from those sensors via LoRa. The study will not include the design of the muon detector subsystem, or the actual muon-imaging of Devil's Peak.

The biggest limitation to this study is time as it has to be completed within a period of three months. There is also a limited budget of R1500 for the project.

1.4 Plan of development

The study starts off with a literature review which will be an overview of the research and development that has been done in the field of muon detection, edge computing, IoT, and communication technologies. It will then continue to the methodology section which

will explain what approach was used to satisfy the requirements of the project as well as exploring previous work done on the project. Thereafter, the schematic and PCB design of the microcontroller subsystem will be documented together with the design choices that were made to come to that design. Then, the results of the study will be reported. And finally, conclusions will be drawn and recommendations will be made for further study in this field.

Chapter 2

Literature Review

2.1 Muons

2.1.1 Introduction to muons

A muon is an elementary particle which, similar to an electron, has a negative charge and is part of the lepton group, but is approximately 207 times heavier than an electron. It was discovered by the American physicists Carl David Anderson and Seth Neddermeyer in 1936. A muon is fairly unstable, and has a lifetime of just 2.2 microseconds. Although their lifetime is short, muons travel at speeds close to the speed of light and can penetrate thousands of metres into the Earth [1].

Muons detected on Earth come into being due to primary cosmic rays. These are produced from a variety of sources in outer space such as solar flares, or novas. The primary cosmic rays travel through space and some collide with Earth's atmosphere. Then, they generate a shower of secondary cosmic rays of which muons are one type [4].

2.1.2 Muon radiography

A valuable property that muons have is that they scatter after penetrating objects. The degree at which their angles change depend on the object. For dense objects with a high number of protons, such as uranium and plutonium, the change in angle is more prominent. Data can be collected on the degree of scattering of muons throughout the

body of the object to create scans of the interior of the object. This field is known as muon radiography and has been implemented since the 1950s [5].

Due to their high momentum, muons can penetrate deeper into thick and dense objects than x-rays, and so they are the better choice when scanning thick samples [5]. Muon radiography is also a safer option than x-rays which can produce stray radiation, unlike muons that are present naturally [2]. About 10 000 muons reach every square metre of Earth per minute which is approximately one muon passing through an area the size of a human hand every second [5]. Thus, there is an ample supply of muons to create accurate scans.

2.1.3 Operation of a muon detector

The main parts of a muon detector are a scintillator and a SiPM. A scintillator is a material that luminesces (produces photons) when it is hit by ionizing radiation (which includes muons). This is a property of the material scintillators are made of. Scintillators can be made from a variety of different materials that can be solid, liquid, or gas and can be organic or inorganic. But in general, high density materials are usually chosen. In the case of inorganic crystal scintillators, the way they operate is when they are hit by ionizing radiation, it excites the electrons within the crystal which then move from the valence to the conduction band leaving holes behind in their place. These electron-hole pairs move around the structure until they reach and are trapped by a defect such as a dopant which are known as activators when it comes to scintillation. At these activator sites, the electrons decay and emit photons which causes luminescence [6].

Although the property of scintillators to luminesce is interesting from a physics perspective, to turn it into something useful an electrical signal is required so that processing and analysis of the luminescence can be made. This is where SiPMs come in. SiPMs are devices that convert light into electrical pulses. They are highly sensitive such that even a single photon can trigger them. The most vital component in a SiPM is the APD (Avalanche Photomultiplier Diode). This component streams an avalanche of electrons when it is triggered by a photon. This avalanche of electrons only lasts for a very short time in the order of 1 nanosecond. This electrical pulse is then what can be passed to an electronics device for processing. One disadvantage of SiPMs is that they need a high voltage power supply. In some cases, an older variant of the SiPM is used which is the PMT (Photomultiplier Tube). However, SiPMs are becoming the more popular choice because of their lower operating voltage, lack of susceptibility to magnetic fields, compact

size, and sturdiness [7].

2.1.4 Use cases

Decision Sciences

Decision Sciences is a company that has been given the exclusive licence of the Intellectual Property (IP) by U.S. Department of Energy's Los Alamos National Laboratory (LANL) to create a system they call Discovery. Discovery is a system that uses muon radiography as a non-intrusive and safe way to generate scans of cargo trucks. They have found that dense shielding such as metal plates around nuclear materials and contraband can pass many types of scanning techniques, like x-rays and gamma rays, completely undetected. However, this does not happen with muons as the dense shielding makes them scatter more which actually makes the materials easier to find. Many smugglers know that water can defeat x-ray systems and would place contraband in pallets of water. However, water still cannot pass by muon radiography undetected [3]. The system can learn the configuration of different types of vehicles to ignore the known features such as the engine and transmission and focus on any unusual materials that might be hidden [2].

Pyramids

A major use for muon radiography is to scan the internal structure of pyramids. This is often used for the purpose of finding hidden rooms inside the pyramids. Luis Alvarez used muon radiography to search for hidden chambers in the Pyramid of Chephren in Giza. Alvarez and his team started recording data in early 1968 on magnetic tape and through the course of their research they were unable to find any hidden chambers in the 19% of the pyramid that they scanned. Although they were unable to find anything, their research did demonstrate the potential that muon radiography holds [8].

In December 2015, a physicist called Kunihiro Morishima and his team placed muon detectors in the Queen's chamber of the Great Pyramid of Giza. After a few months, the researchers came upon something promising. To validate their findings, two other teams of physicists placed muon detectors of different types at different locations inside and outside the pyramid. All three teams agreed that their results showed a cavity above the Grand Gallery. The space was found to have a similar cross section as the Grand Gallery and at least 30 metres long. This was the largest discovery of a new space inside

the pyramid since the 19th century [9].

Volcanoes

Muon radiography has also been used to scan magma chambers in order to estimate when the volcanoes will erupt. Hiroyuki K. M. Tanaka from the University of Tokyo and his colleagues investigated the density distribution of magma in Mount Iwodake. A $1m^2$ detector was placed at the foot of the volcano to obtain an image of the conduit on the inside. This information can be used to predict the development of eruptions and determine which areas close to the volcano are in danger [5].

Cosmic Pi

Cosmic Pi is a low-cost cosmic ray detector kit. The kit contains scintillators and SiPM modules as well as a PCB with an amplifier for the detection of muons. The PCB also contains a GPS module, temperature and humidity sensor, and accelerometer. The kit also contains a Raspberry Pi Zero WH which helps with wireless communication and some processing. The idea is for many of these kits to be assembled and used by people all over the world so that the data gathered by these devices can be collected in order to create a global telescope [10].

2.2 The Internet of Things (IoT)

2.2.1 Introduction

IoT is the growing trend whereby more and more devices are being connected to the internet. Through the development of IoT, devices that initially had very limited functionality have become more useful and responsive not only through their ability to connect to the internet, but more specifically because of their ability to collaborate with other devices. This makes the environment smarter as more internet-capable devices are integrated with it. These devices collect a vast amount of data about the environment which then leads to a challenge with IoT which is how to store and process such large quantities of data that are related to each other. The challenge is made harder by the fact that the data can be produced by many different types of devices running on different platforms that

are connected from a variety of remote locations. This is where cloud computing arises to offer a solution [11].

Cloud computing allows a way for IoT devices to connect to each other through central servers that are very powerful and reliable. Cloud computing also offers analytics tools and various ways to visualize data. In addition, it also allows one to access applications from anywhere that has an internet connection [11]. There are some issues with the use of cloud computing when it comes to IoT which are explained further in Section 2.3.

Through the large collection of data about an environment (such as a home environment or work environment like construction), it helps one produce a common operating picture (COP) [11] of the environment. In addition, through this collection of data, a new realm of possibilities are opened in the field of data analytics, and it allows one to understand, optimize, and control the environment better.

Another important factor of IoT is that it should be able to operate without the interference of humans or, put differently, it should "disappear from the consciousness of the user"[11]. To achieve this, IoT devices need to: be able to communicate relevant contextual information; have a shared comprehension of the environment and context; and have strong analytics tools. This will allow IoT devices to act autonomously [11].

In order to fully utilize the available Internet technology, large-scale wireless sensor networks that are platform independent must be deployed and include the collection, management, storage, and analysis of data [11].

2.2.2 History of IoT

The term 'Internet of Things' was first introduced by Kevin Ashton in a presentation he made at a Proctor and Gamble (P&G) conference in 1999. The presentation specifically addressed the use of RFID in supply chain management [12]. But since then, the term has expanded to include a whole variety of applications and smart devices that can operate autonomously.

2.2.3 The use of IoT

Radio Frequency Identification (RFID)

RFID is the use of small electronic chips for wireless data communication. They assist with the identification of the object they are attached to. One major benefit is that they are passive elements and not battery powered, which saves a lot of energy. They use the power of the RFID reader's probing signal to communicate their ID. The major usage of RFID tags is currently with bank cards to allow for quick transactions. Another major usage is in the monitoring of cargo by attaching RFID tags to port containers [11].

Lighting

The control of lighting is one of the classic examples of the use of IoT because it can take place fully without the need for human intervention. Light sensors that are positioned outside a building can communicate the light intensity to the lights that are inside the building which would then adjust their brightness accordingly. This can save much energy and ensures that lights are not too bright or left on unnecessarily. This is often done using Bluetooth mesh nodes that are attached to all the sensors and lights.

Automotive

Predictive maintenance technology can be used in one's car to have a better understanding about the internal state of the car. Data that is collected by many sensors in your car can be sent for analysis on the cloud to notify the driver of any potential problems that might present themselves in the future. This allows the user to take action to avoid potential incidents [13].

2.3 Edge computing

2.3.1 Introduction to edge computing

Edge computing is a distributed computing paradigm that brings computing closer to the edge where it is needed. The edge can be defined as any computing resources that are at a distance from cloud data centers and closer to data sources. In the last couple of years, the amount of data produced at the edge has grown significantly with the great expansion of IoT and the increase of data producers. As an example, a Boeing 727 can produce 5 GB of data every second, and an autonomous car can produce 1 GB of data every second. This has put much pressure on networks and cloud data centers and they cannot keep up with the big volume of data being produced. This is why edge computing is needed because it lightens the load on cloud data centers and decreases network traffic [14].

2.3.2 History of edge computing

The idea behind edge computing started in the late 1990s when Akamai launched content delivery networks (CDNs) to increase the speed of web services. CDNs use nodes closer to the user devices to prefetch and cache content so that it can be delivered to users faster. CDNs are especially useful when it comes to video streaming because there is no more reliance on one central server to do all the work. The potential for edge computing truly became evident in 1997 when Brian Noble and his coworkers implemented speech recognition on mobile devices by having the resource intensive work delegated to a nearby server and then the results would be only have to be sent to the mobile device.

2.3.3 Benefits of edge computing

Edge computing increases responsiveness because it decreases the distance data exchanges have to travel. Data does not have to traverse various networks to get to the cloud for processing and so latency is decreased [15]. As a test for edge computing, researchers built a platform to run a facial recognition application, and found that the response time improved from 900 to 169 ms when using edge computing instead of a cloud based solution [14]. According to Ha *et al.* [16], using cloudlets to do the processing for wearable cognitive assistants improved the response times between 80 and 200 ms. In addition,

the energy consumed was lowered by 30% to 40% [16].

A great example of the potential of edge computing was given by Shi *et al.* [14], which is in the case of video analytics. If a child goes missing, the video data from street cameras can be used to help locate the child. However, to send all of that large video data to a central server to be analyzed would be a very slow task which would not be suitable for such an urgent matter. But if edge computing is utilized, the central server can signal the devices at the edge to search through their local data. Then, an edge device only needs to report back the result to the cloud server. This is very beneficial because not only is the overhead of transferring large amounts of video data eliminated, the search for the child is done in parallel by all the devices which massively reduces the time to retrieve a result [14].

Privacy and security is a major issue of concern with the increasing prevalence of IoT. IoT devices can collect a vast amount of private data that should not be available to unauthorized third parties. And so, sending data to the cloud for processing could be a problem because it exposes the data to potential threats such as unauthorised parties intercepting the data. For this reason, it is more secure to process the data where it is collected - at the edge [14].

2.3.4 Challenges of edge computing

One challenge experienced with edge computing is that of programmability. With cloud computing, programs are usually written in one language and compiled for a specific platform. In this way, there is more control over where and how a program is run. But because edge computing platforms are more diverse, it becomes more difficult to create a program that will function properly everywhere it is used [14].

And next is the issue of data abstraction. It should be avoided to send raw data from IoT devices to applications that run on the gateway. Firstly, because of security, and secondly, because the extra load of data could overwhelm the gateway. However, one then has the issue of usability of the data because information describing the data is removed which could make it more difficult for the gateway application to make sense of the data and use it. The reliability of the data can also be a problem because firstly the sensing of the data might not be completely accurate and the wireless communication medium might be unreliable. And so, a major challenge for edge computing and IoT is how to extract useful data from an unreliable source. [14].

According to Shi *et al.* [14], there are four essential issues when it comes to service management for edge computing which are summarized as follows:

- Differentiation: some services are more important than others and so there needs to be some form of prioritization.
- Extensibility: the network should be able to grow and easily accept new nodes to connect to it.
- Isolation: several applications should have access to a resource so that if one application fails, the resource should still be accessible. In this way, if one application fails, it does not affect the rest of the system.
- Reliability: if a service is not working properly, it is a challenge to find what exactly the problem is. A way to overcome this problem would be if the service could provide a message of what component failed.

The reliability of a service is mostly centred around data collection and communication. When the communication medium is not reliable, one way to overcome this is to use predictions from historical and reference data, but this is still an ongoing challenge.

2.4 Communication technologies

2.4.1 Introduction

In order for IoT devices to interact with each other, they need to be able to communicate. This can be either with short range communication technologies such as Bluetooth, Zigbee, LoWPAN, RFID, Zwave, and NFC as well as long range technologies such as cellular networks (such as 3G and 4G), Sigfox, and LoRa. All of these technologies utilize radio waves instead of dealing with the complications of physical wiring such as not being easily extendable. These technologies can be based locally on a more private network with a smaller range, or they can make use of the cloud for interconnection and take advantage of cloud services [17]. The following subsections will take a look at the different long range communication standards.

2.4.2 Cellular networks

Cellular communication, especially through the evolution of 4G and 5G, allows for communication at very high speeds. 4G has a theoretical maximum of 300 Mbps while typical values are around 40 Mbps. And 5G has theoretical speeds of 10 Gbps and potentially even more, but currently speeds of around 200 Mbps are experienced [18]. Cellular networks also have a very wide range with LTE being able to reach 100 km. The main disadvantage of cellular networks, however, is that they are relatively power hungry and might not be suitable for small low-powered devices that should be able to run on a battery for a number of weeks [17].

2.4.3 Sigfox

Sigfox is a narrowband technology that uses binary phase-shift keying (BPSK) for radio transmission. The narrowband technology allows a receiver to only listen for incoming signals on a smaller spectrum of frequencies which reduces the effects of noise [19]. Sigfox uses a very low data rate of 100 to 600 bps but has a range of approximately 40 km in rural areas and 10 km in urban areas.

Sigfox was found in 2010 by Ludovic Le Moan and Christophe Fournet with the intention to improve the world of IoT and allow devices to connect to each other in a more efficient way. Sigfox places great emphasis on providing a service that can extend a long range to reach low-powered devices that need to transfer small packets of data. All the complex computations for Sigfox are performed on the cloud instead of the low-power devices for additional energy savings. Another advantage is that Sigfox is capable of operating very efficiently and has reduced signalling and protocol overhead [20].

One usage of Sigfox is in the monitoring of the fill levels of dumpsters. In this way, dumpsters do not have to be manually checked, and the refuse collection provider does not have to be manually sent a request for a pickup. Instead, these processes happen automatically by having a sensor send a signal to the pickup collection provider, using Sigfox, that the dumpster is full and ready for a pickup. In addition, a temperature sensor and fire alarm can be added to the system for extra safety [20].

2.4.4 LoRa

LoRa (Long Range) is a technology provided by Semtech. It is used to transmit data of low data rates (0.3 to 27 kbps) [17] over a long range and can transmit 5 km in urban areas and 20 km in open space. LoRa uses the sub-GHz bands of the ISM radio band spectrum and the specific band it uses depends on the geographical location. A major advantage of LoRa is that it is very low-power, making it suitable for IoT applications [21].

It is important to make a distinction between LoRa and LoRaWAN. LoRa is the modulation technique to get a high receiver sensitivity and reduce bit errors. LoRa also refers to the systems and chips that support this type of modulation [22]. However, LoRaWAN is quite different. It is a MAC layer protocol that is built on top of LoRa. LoRaWAN also includes encryption and identification functionality at the gateway level. Additionally, LoRaWAN has a cloud component which the gateways connect to [23].

Chapter 3

Methodology

The approach used for this project was mostly similar to that of the waterfall methodology with some touches of the agile methodology. The work took a mostly linear approach with problem definition, design, and implementation, but there were numerous times where the design would have to change or new components had to be added to the design and then the project had to adapt to those changes in an agile manner.

The project had a team of five members. The undergrad student doing this project, a master's student, Alan Pohl, who previously worked on this project for his final undergraduate project, the supervisor for the project, Jane Wyngaard, and a team from the physics department consisting of two members. The physics team was made up of a student, Daniel Weiss, doing his honours project on muon imaging and his supervisor, James Keaveney. The team would meet weekly for updates on everyone's progress together with direction from the supervisors. Much collaboration was done with Pohl as he pursued the design and implementation of the muon detector subsystem further as his master's project which had to interface with the microcontroller subsystem of this project. The physics team acted as a sort of client for this project as the eventual goal was to provide them with the muon detector to be used in their research.

Below is a diagram showing the sequence of steps in the project, followed by a more detailed description of each step.

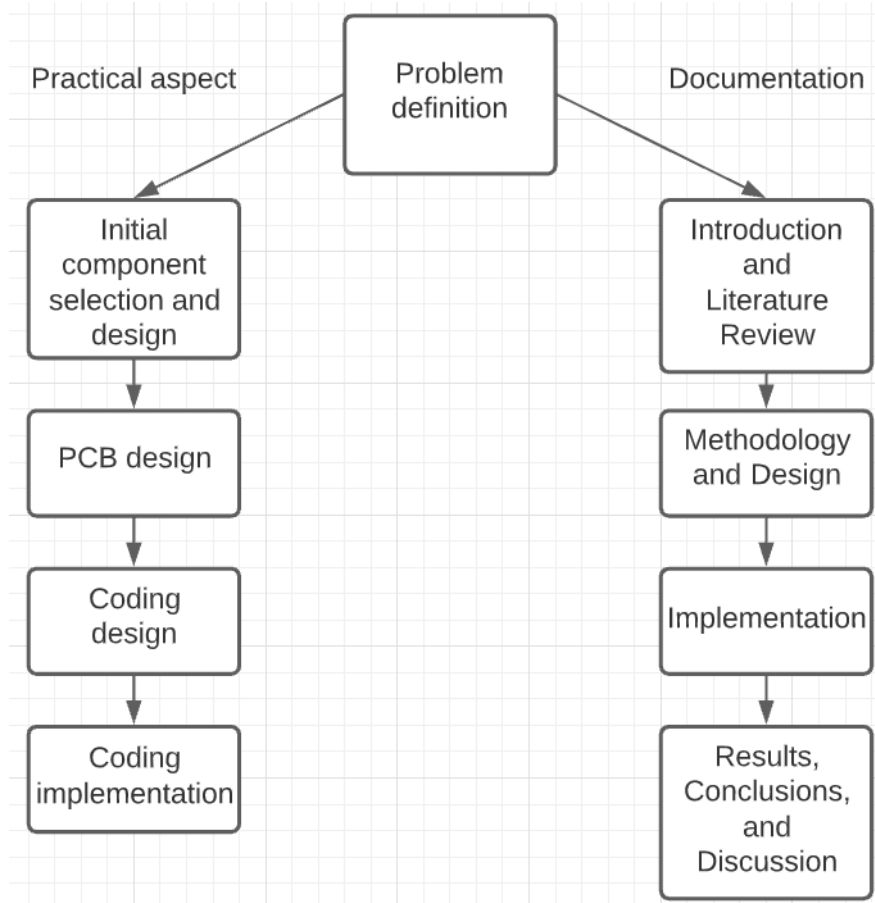


Figure 3.1: Flow chart of the methodology

3.1 Problem definition

This was the initial step to start off the project. Through introductory meetings with the team, the goals and requirements of the project were established. The work that was done by Pohl during his undergraduate project was investigated as a starting point for the project.

3.1.1 Previous work done on the project

Pohl designed the muon detector subsystem in his undergraduate final project which includes scintillator planes, SiPMs, comparators, pulse shaping circuits, ADCs, and encoders.

The scintillator planes had the following layout:

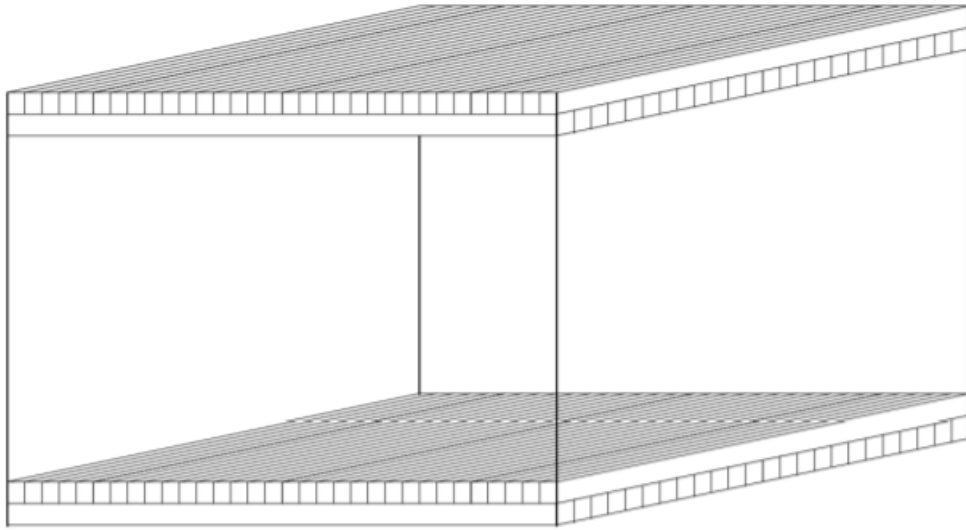


Figure 3.2: Layout of scintillator planes [24]

The scintillator has two sets of planes - one at the top and one at the bottom with a space between the two sets. Each set has one plane of vertical scintillator blocks and one plane of horizontal scintillator blocks. This was done to create a grid so that when a muon passes through the plane, the exact x- and y-coordinates of where the muon hit the plane can be found. The reason for having two sets of planes with a distance between them is so that one can determine the trajectory of the muon. When a muon hits the scintillator, the coordinates of the top plane and the bottom plane can be compared to find the trajectory. For the trajectory to be found with good accuracy, there must be enough scintillator blocks to create a grid that is sufficiently fine. By weighing the costs and the accuracy required, it was decided to use 32 scintillator blocks for each of the 4 planes.

The overall design of each plane in the scintillator subsystem is as follows:

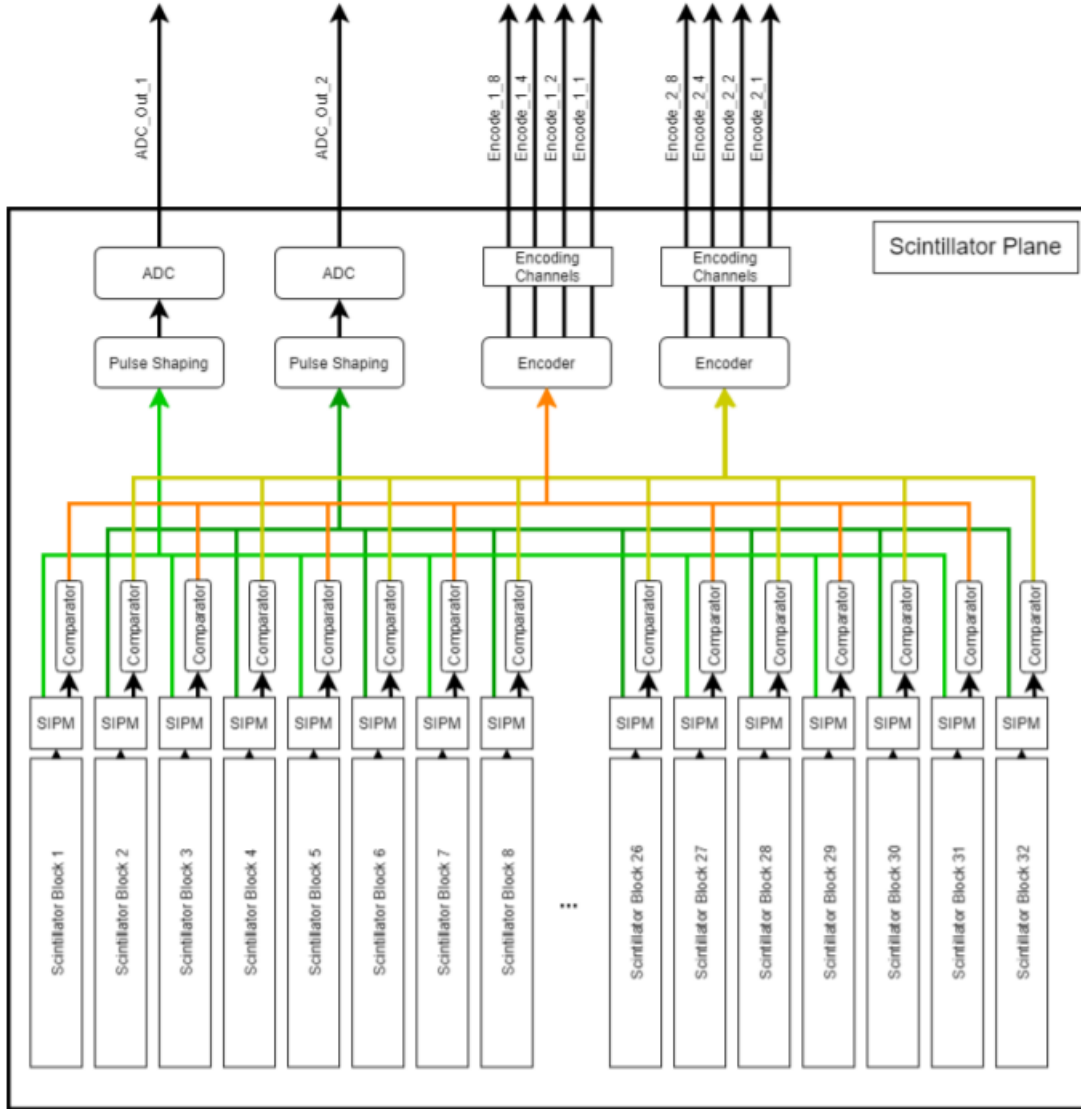


Figure 3.3: Layout of the muon detector subsystem [24]

When a muon hits the plane, the scintillator that was hit absorbs the muon's energy and converts it into photons. These photons then travel to the SiPM (silicon photomultiplier) which produces an electrical pulse. Thereafter, two things happen with the pulse.

Firstly, the pulse is sent to an ADC so that the pulse can be read. However, because the pulse is so short, it first has to pass through a pulse shaping circuit to extend it so that it can be picked up by the ADC. Then, the digital output from the ADC is sent to the microcontroller subsystem.

Secondly, in order to find which scintillator block was hit, a logic signal must be passed

to the microcontroller from the specific scintillator block. In order to generate a logic high signal, the pulse from the SiPM is passed through a comparator. And because there are 32 scintillator blocks which would require many pins on a microcontroller, an encoder is used to reduce the number of pins required from 16 to only 4 pins for each of the two sets of scintillator blocks which brings it to a total of 8 encoder pins.

Another important design choice to mention is the separation of the scintillator blocks into two sets of 16 and the use of two ADCs and encoders instead of one of each. There is a problem that a muon might come at the scintillator plane at such an angle that it activates two scintillator blocks that are next to each other. But a muon will never be able to activate more than two scintillator blocks. For this reason, every second scintillator block is connected to its own ADC and encoder so that no information is lost.

3.1.2 Requirements for this project

The aim of this project was to continue from where Pohl left off and implement the microcontroller subsystem for the project. This subsystem had two major elements - a hardware part and a software part. The hardware element consisted of the selection of all the hardware modules and components and the placement and routing of those components on a PCB. The software element consisted of the program that would run on the microcontroller, that was on the PCB, which would utilize the hardware components selected in order to perform the major functions needed for the project.

The subsystem needed the following hardware components:

- GPS module: this is used to find the location of the device. During muon detection, several scintillator planes are usually used. It is then necessary to know where each of these planes are in relation to each other in order to determine how to piece together their outputs to produce a muon image.
- Accelerometer: this component is often used to determine the amount of acceleration of a device, but in this case it is used to determine the angle of inclination of the plane which helps with determining the trajectory of a muon.
- Temperature and humidity sensor: temperature and humidity can have an effect on the muons and their detection, and so this is useful information to be sent to the physics department for analysis.

- Headers for interfacing with the muon detector subsystem: the muon detector subsystem needed to get its power from the microcontroller subsystem. In addition, the microcontroller subsystem needed to interface with the ADCs and encoders from the muon detector subsystem.
- LoRa module: this module enables LoRa communication which is necessary to transmit the information gathered from the muon detector subsystem and the sensors on the microcontroller subsystem.

In addition, collaboration had to be done with Pohl to decide which ADCs to use on the daughter board of the muon detector subsystem to sample the output of the pulse shaping circuits. This is because the ADCs fell in the middle ground between the muon detector subsystem and the microcontroller subsystem. The ADCs needed to be fast enough to capture the pulse produced by the pulse shaping, but at the same time have the right specifications that it could work with the chosen microcontroller.

Functional requirements

Functional requirement number	Description
FR001	The system must include a functioning PCB that can be programmed and can communicate with its peripherals.
FR002	The PCB must contain a microcontroller that can store and run a moderately sized program and have enough I ² C, SPI, and UART interfaces to support all the peripherals that need to connect to it.
FR003	The PCB must be able to interface with the muon detector subsystem.
FR004	The system must be able to find its GPS coordinates.
FR005	The system must be able to determine its angle of inclination.
FR006	The system must be able to determine the surrounding temperature and humidity.
FR007	The system must be able to send the information it gathered via LoRa.
FR008	The microcontroller must be fast enough to extract the values from the ADCs, GPS module, accelerometer, and temperature and humidity sensor, and work out where exactly on the scintillator planes the muon made contact, and then communicate this information via LoRa all in under one second. This is because approximately one muon will hit scintillator every second.
FR009	An ADC must be selected for the daughter board of the muon detector subsystem which must be fast enough to sample at least 10 values of the output of the pulse shaping circuit.

Table 3.1: Functional requirements of the system

Non-function requirements

Non-functional requirement number	Description
NF001	Standard components that are available from JLCPCB must be selected.
NF002	On the PCB, there should be a minimum track spacing of 0.2 mm.
NF003	The minimum via to pad and via to track spacing should be 0.2 mm.
NF004	Vias with a drill hole size of no less than 0.2 mm with a diameter of at least 0.4 mm must be used.
NF005	Blind and buried vias must not be used.

Table 3.2: Non-functional requirements of the system

3.2 Practical aspect

3.2.1 Initial component selection and design

In this stage, the major components for the microcontroller subsystem were selected such as the microcontroller, of course, and the accelerometer, temperature and humidity sensor, GPS module, and LoRa module. It had to be confirmed that all components that were selected were available from JLCPCB in order to be manufactured. If a critical component was needed that was not available from JLCPCB, other plans were made such as ordering from RS Components or DigiKey so that the component could be soldered onto the PCB manually. In addition, in this stage diagrams of the design of the microcontroller subsystem were drawn up using online tools such as diagrams.net, and the interfaces between the muon detector subsystem and microcontroller subsystem were determined. This process is explained in detail in Chapter 4.

3.2.2 PCB design

In this section, the schematic of the PCB for the microcontroller subsystem was created using KiCAD. After the schematic was completed, the PCB layout and routing was done so that it could be manufactured by JLCPCB. All specifications laid out by JLCPCB

were followed to make sure they could produce the PCB without any issues. This section is demonstrated in Chapter 5.

3.2.3 Coding design

This is where the planning for the programming of the microcontroller took place. Diagrams such as an FSM (Finite State Machine) were drawn to plan out the flow of the logic of the main program that would be running on the microcontroller. This section is demonstrated in Chapter 4.

3.2.4 Coding implementation

During this stage, the actual coding of the microcontroller was to be done and tested. The software that was planned to be used was the STM32CubeIDE. The various communication protocols for the components used in the microcontroller subsystem had to be investigated as part of the coding design. UART was used for the GPS module, SPI for the ADCs and IO expanders, and I²C for the accelerometer and temperature and humidity sensor.

3.3 Report

The report was written in parallel with the development of the project. The introduction in the report gave a basis for the project by detailing the history and background to the project as well as its aims. Then the literature review was where all the research had been gathered that was needed to implement the project. The methodology provided the structure for how the project was developed and implemented. The design section was where all the design decisions were made such as component selection and schematic design. This section also included diagrams that show the operation of the system and its programs. Thereafter, the implementation section was where the PCB layout was produced. Finally, the results of the investigation were reported, and conclusions of the project were drawn.

Chapter 4

Design

4.1 PCB Design

4.1.1 Daughter board

The muon detector subsystem designed by Pohl has two main parts: the scintillator planes and the daughter boards. The scintillator planes contain the 32 scintillator blocks which are used to detect muons. The scintillator planes then connect to the daughter boards which contain the electronics of the muon detector subsystem. After the completion of Pohl's design, the interface of the daughter boards to the main PCB, which is the microcontroller subsystem, was determined. Below is the simplified design of a daughter board which was used to determine the interface with the microcontroller subsystem:

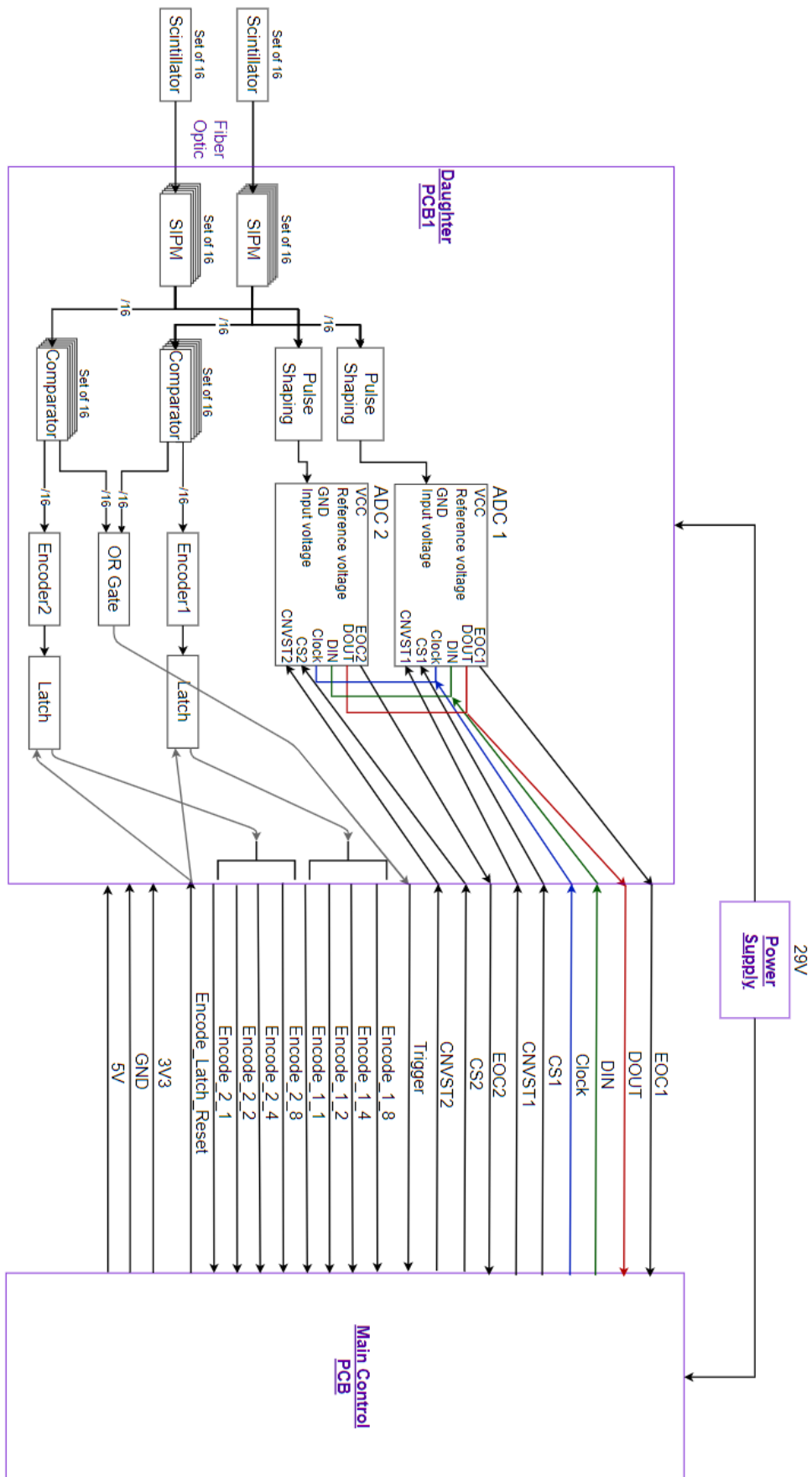


Figure 4.1: Diagram of the daughter board and its interface with the microcontroller subsystem

Each of the four scintillator planes will have one daughter board which will have the same design as the one above. Two major changes were made to the design of the daughter board from the initial design in Chapter 3.

Firstly, the outputs of the comparators were connected to an OR gate. Each plane has its own OR gate. The OR gate will function as a trigger to the microcontroller as it indicates that an event on the scintillator has occurred, i.e. that a muon hit the scintillator. The alternative to using OR gates to check for an event would be to check every encoder pin to see if any of them went high. The problem with this is that it takes much longer to read the 32 encoder lines than checking the 4 OR gates.

The other change was that the output of the encoders were connected to a latch so that their outputs are remembered. This is necessary because after the microcontroller is triggered by the OR gate, it takes a while for the microcontroller to gather all the data from the daughter board. The comparators, and therefore the encoders, only stay at a high output for a few microseconds. Thus, in order not to lose the output of the encoders, they must be stored by latches. Because of this, an additional reset line is required to clear the values of the latches after they have been read by the microcontroller.

The daughter board is powered by 5V, 3V3, and GND lines. The 5V supply is required to power the ADCs, while the standard 3V3 is required to power the rest of the electronics and logic lines.

The ADCs have several useful pins beyond the clock, MISO, MOSI, and NSS pins that are necessary for SPI communication. There is a conversion start (CNVST) pin that allows you to enable or disable the ADC. This functionality will be explained further in the ADC selection section. Then, there is also an end of conversion (EOC) pin which can signal the micro that a new sample has finished being converted by the ADC.

ADC selection

One of the vital requirements for the ADC was to ensure that it is fast enough to capture at least 10 samples of the output of the pulse shaping circuit. Below is shown two graphs: the output of the SiPM is on top which is also the input to the pulse shaping circuit, and then the output of the pulse shaping circuit is shown at the bottom.

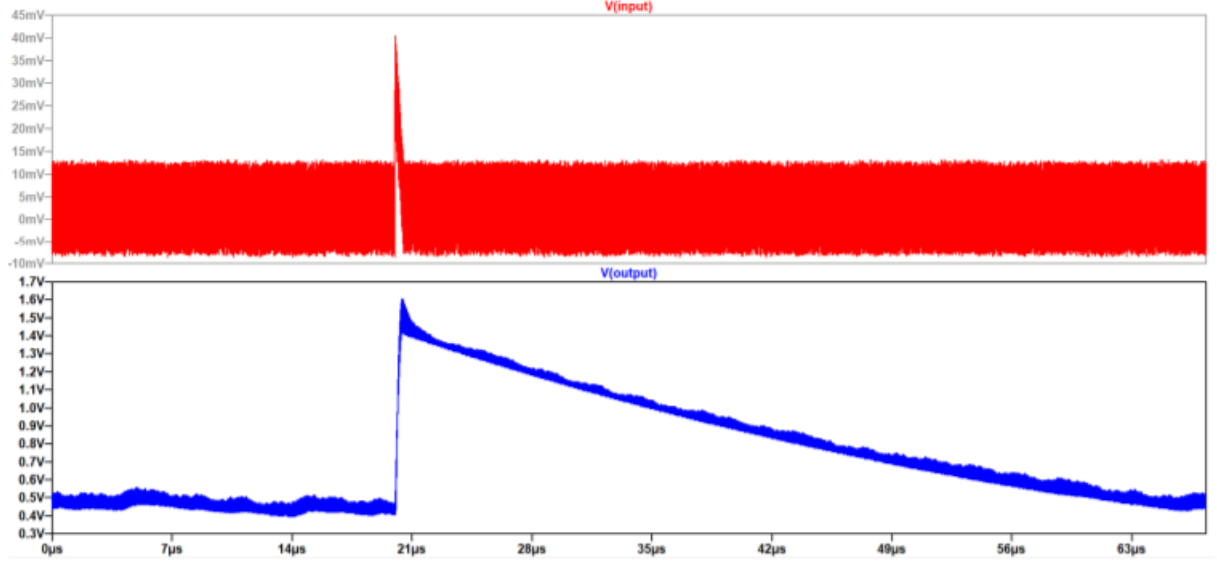


Figure 4.2: Pulse shaping circuit output [24]

It can be seen from the above figure how the very short output from the SiPM is significantly extended. On the output of the pulse shaping circuit, the pulse starts at approximately $20 \mu\text{s}$ and ends at approximately $66 \mu\text{s}$. Thus, the pulse has a duration of approximately $46 \mu\text{s}$. As 10 samples must be taken from this pulse, the ADC needs to be able to take a new sample at least every $4.6 \mu\text{s}$. Then, the required sampling rate of the ADC can be calculated as follows:

$$\begin{aligned}
 \text{Sampling rate} &= \frac{1}{\text{sampling period}} \\
 &= \frac{1}{4.6 \mu\text{s}} \\
 &= 217391 \text{ Hz}
 \end{aligned}$$

Therefore, when looking at standard ADC values, an ADC that has a sampling rate of at least 220 kHz was required.

The other major requirement was that the ADC should include a FIFO to store the most recent values it converted. There are a number of reasons why this was required. Firstly, because the required sampling rate is quite fast and the microcontroller would have to read from 8 ADCs that are producing new data at 220 kHz, which would mean that a fast microcontroller would have to be chosen and timing would be very critical to make sure no values are missed. Secondly, all 8 ADCs needed to be connected to the same SPI interface because of the limited number of SPI interfaces on the available microcontrollers, which means that the ADCs would have to take turns to send their data

which makes it take even longer to read all the data from the ADCs. This challenge can be demonstrated with a calculation. A standard 12-bit ADC would need to send 16 bits over SPI (SPI sends one byte at a time), and taking into account the start and stop bits that are sent to control communication, a total of 18 bits will need to be sent to send one sample from an ADC to the microcontroller. Taking a standard SPI speed of 18 MHz (which is the speed allowed by the chosen microcontroller [25]), we get the following:

$$\begin{aligned}\text{Time to send 18 bits} &= \frac{\text{Size of data}}{\text{Frequency in Hz}} \\ &= \frac{18 \text{ bits}}{18 \text{ MHz}} \\ &= 1 \mu s\end{aligned}$$

Because 8 ADCs need to be read from in the worst case, it will require $8 \mu s$ to read one sample from each ADC which violates the requirement of $4.6 \mu s$. Therefore, a FIFO is absolutely required and needs to be big enough to store at least 10 values.

With these requirements in mind, the MAX11635EEE+T was found. It has a sampling rate of 300 kHz which satisfies the requirement of at least 220 kHz. In addition, it has a FIFO that can store up to 16 samples, making the MAX11635EEE+T the perfect choice for the daughter board.

The ADC will be operated as follows: when the scintillator plane is hit by a muon, the microcontroller will wait for about $50 \mu s$ to allow the entire pulse to be produced by the pulse shaper and converted by the ADC. After this point, the ADC is disabled by using the CNVST pin. This will prevent the ADC from adding in more values into the FIFO which will push out the desired pulse shaper output values before they can be captured by the microcontroller. Then, after the values have been read by the microcontroller, the ADC will be enabled once again so that it can be ready to convert values from the pulse shaper at any time.

4.1.2 Main PCB

The design for the microcontroller subsystem is shown below:

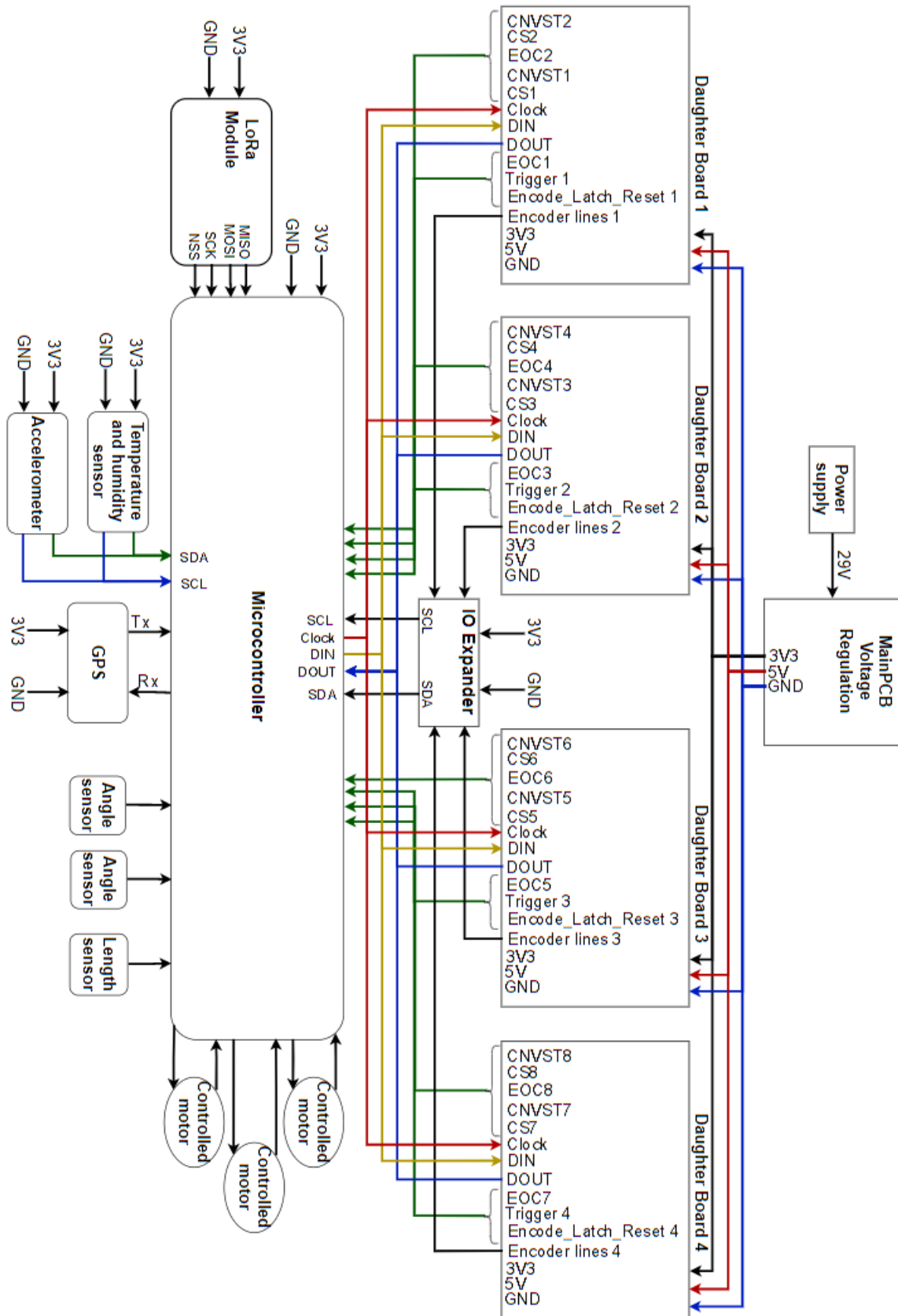


Figure 4.3: Microcontroller subsystem design

Some simplifications were made for a less messy demonstration of the subsystem. The 8 encoder lines of each daughter board were joined together as one line. In addition, some of the connections of the peripherals such as the accelerometer and LoRa module were omitted. However, the connections that are of most interest such as the supply connections and communication interfaces are shown.

This subsystem design satisfies the hardware requirements laid out in the methodology by including a GPS module, accelerometer, temperature and humidity sensor, and LoRa module. The subsystem includes the correct interface to the muon detector subsystem. In addition, the subsystem also attempts to use components that are available from JLCPCB as far as possible.

It can be seen in the diagram that angle sensors, a length sensor, and controlled motors are included in the diagram. These components were not developed in this project, but will be included in a future project. One motor will allow one to change the ‘top-down’ angle of the scintillator, while another motor will allow one to change the ‘left-right’ angle of the scintillator. The angle sensors will measure the extent of these two angles. The final motor will control the distance between the two planes. The distance between the planes changes the angle of acceptance that the scintillator allows. When the planes are close together, a wider angle of muons are accepted. This is because a muon can come at the scintillator nearly parallel to the plane, but still manage to travel through all 4 planes because they are close together. But when the planes are further apart, a more focused angle of muons are accepted. The length sensor would measure this distance between the planes.

The following sections detail the components that were selected and reasons for the choice.

Microcontroller: STM32F051R8T6

One of the major reasons for this choice was because it is a mainstream microcontroller with much community support. Its core is an ARM 32-bit Cortex-M0 CPU with a frequency of up to 48 MHz, making this microcontroller sufficiently powerful for this project. It has a flash size of 64 kB, making it unlikely to run out of memory for a medium-sized program. It also has the perfect amount of communication interfaces needed of two SPI (ADCs and LoRa module), two I²C (accelerometer and temperature sensor connected to one interface and the IO Expander to the other), and two USART interfaces (of which only one will be used for the GPS module). Another major reason for this choice for a microcontroller, was because it is a 64-pin package with many GPIO connections which

are necessary to interface with the muon detector subsystem which requires many pins [25].

However, for the greater part of the project another microcontroller was chosen and worked with before the selection of the STM32F051R8T6, which is the STM32WL55JC. This was because of its large flash size, extensive 73-pin package, and the fact that it has two processing cores (ARM Cortex M4 and M0+). But the major reason it was chosen was because LoRa is integrated with it. This meant that no additional external LoRa module would be required and it removes the risk of communication problems between the microcontroller and the LoRa module [26].

The reason the STM32WL55JC was not chosen even though it seemed to be the perfect selection for the project was because of the type of pins. It had BGA (Ball Grid Array) pins with a pitch of 0.5 mm [26]. This means that the pins are very densely packed, and routing between the pins is impossible. The only way to route this type of microcontroller is through placing blind and buried vias under the microcontroller pins (called via-in-pad). Firstly, the use of these vias is not ideal because of the high manufacturing complexity which makes them very expensive and time consuming to make. Secondly, because the BGA pins are so small, vias with a very small diameter would be required which violate JLCPCBs minimum specifications. The vias needed a drill hole size of 0.17 mm and a diameter of 0.34 mm which is lower than JLCPCB's minimum specifications of a 0.2 mm drill and a 0.4 mm diameter. It is likely that microvias would be required in order to get these small sizes which is also a problem because microvias are more specialized and expensive than normal vias [27]. Thirdly, because these vias are of the via-in-pad type, they need to be capped and filled with epoxy. This is because of the small size of the BGA pins. The pins are so small that they can only take a little bit of solder. If the via used under the pad is an open hole, the via might wick all the solder away from the pad which creates a weak or broken connection between the pad and microcontroller pin [28]. In addition, when it came to routing the microcontroller this was the result:

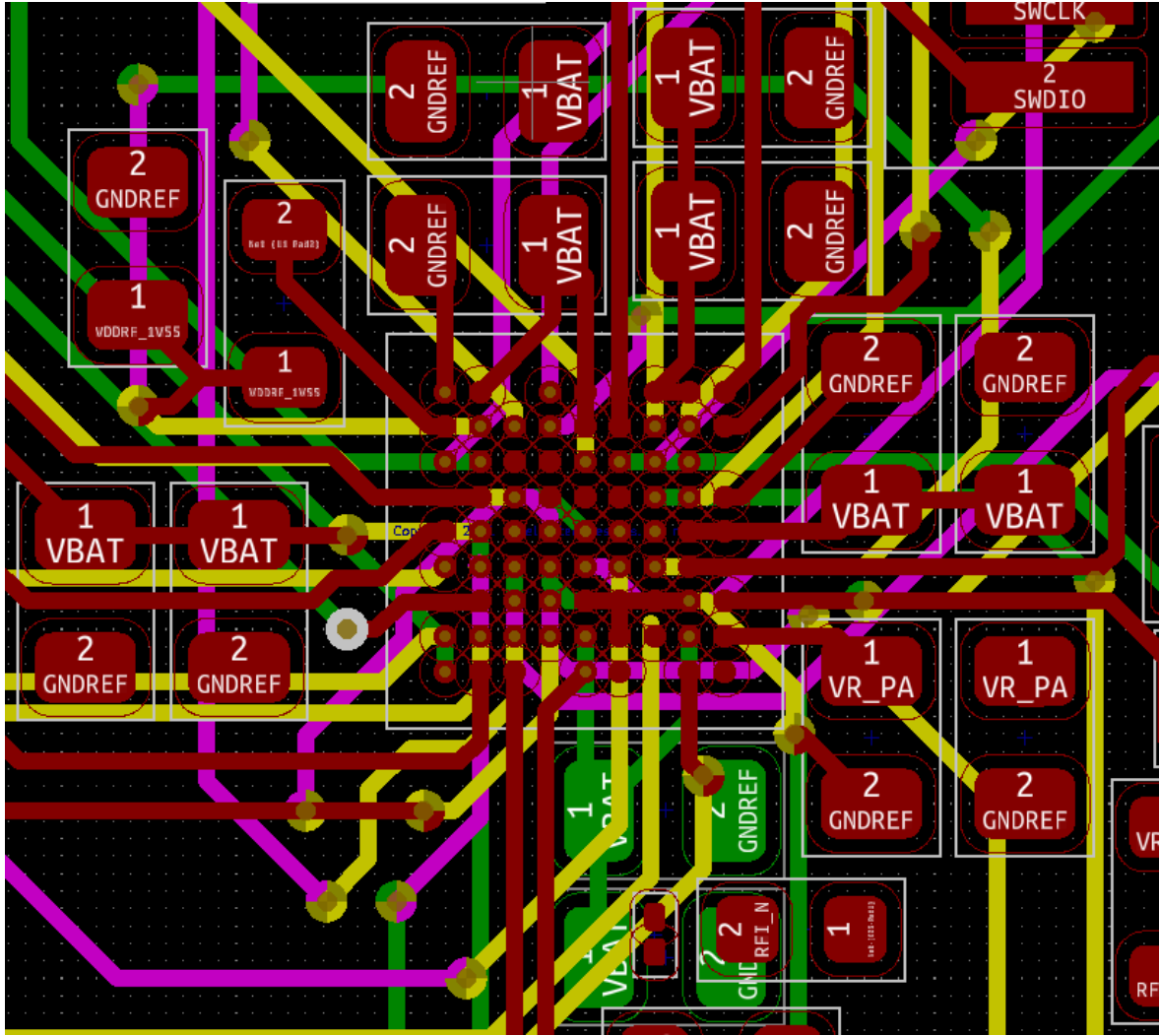


Figure 4.4: Attempted STM32WL55JC routing

Even through the use of 4 signal layers, the routing was highly difficult and there were still a number of unconnected pins which were stuck on the inside. It was likely that the design would have needed to be changed to 6 signal layers to complete the routing. And so, because of the high complexity, high costs, and greater potential of making a mistake and getting a faulty PCB, it was decided to change the design and use another microcontroller which turned out to be the STM32F051R8T6.

GPS module: ATGM336H-5N31

The ATGM336H-5N31 is one of the most popular GPS modules at JLCPCB which makes it a trustworthy choice. This GPS module uses the standard NMEA0183 protocol which is the standard with GPS modules making it easy to work with. This module requires

an external antenna, and so the KYOCERA AVX GPS Patch Antenna 1004138 will be used. This antenna is not available from JLCPCB and would have to be procured from RS Components [29].

Accelerometer: LIS3DHTR

This component was mainly selected for its availability, and the fact that it is an STMicroelectronics product which makes it more reliable. It also includes a 32 level FIFO for each of the X, Y, and Z output channels which can be quite useful [30].

Temperature and humidity sensor: SHT30-DIS

This component was chosen firstly because it includes a temperature sensor and a humidity sensor in one package instead of needing two separate components. It also has a compact size which aids with the layout and routing of the PCB. In addition, it was also chosen because of its high reliability, accuracy, and high signal to noise ratio [31].

LoRa module: SX1276IMLTRT

This component was mainly selected for its availability at JLCPCB, and the fact that there are libraries readily available for it. Another benefit is that it has a wider frequency range than some of the other models in the same line such as the SX1278 and SX1279. Its frequency range stretches from 137 - 1020 MHz whereas the SX1278 and SX1279 only range from 137 - 525 MHz and 137 - 960 MHz respectively [32].

IO expanders: PCF8575, PCF8574

Because of the 36 pins needed for the encoders from the daughter board (8 pins each for 4 planes plus 4 pins to reset the latches that hold onto the data of the encoders), it adds extra complexity for the routing of the PCB and requires a microcontroller with a very high number of GPIO pins which are not always readily available. For this reason, it was chosen to add IO expanders to the design to which the encoder lines can connect. Then, only two I²C pins (SDA and SCL) are needed from the microcontroller to communicate with all 36 encoder pins. This also has the added benefit of making the design more

modular. And so, two PCF8575 modules and one PCF8574 module were chosen. The reason the PCF8575 was chosen was because it was the biggest IO expander JLCPCB had available with 16 IO pins. The PCF8575 is also one of the most popular IO expanders which makes it more reliable. Because the PCF8575 only has 16 pins, two of them were put in the design which brought the number of IO pins to 32. But because an additional 4 pins were needed, the smaller version of the PCF8575 was chosen which is the PCF8574 with 8 IO pins [33][34].

4.2 Coding design

An FSM of the control program that will be running on the microcontroller is shown below:

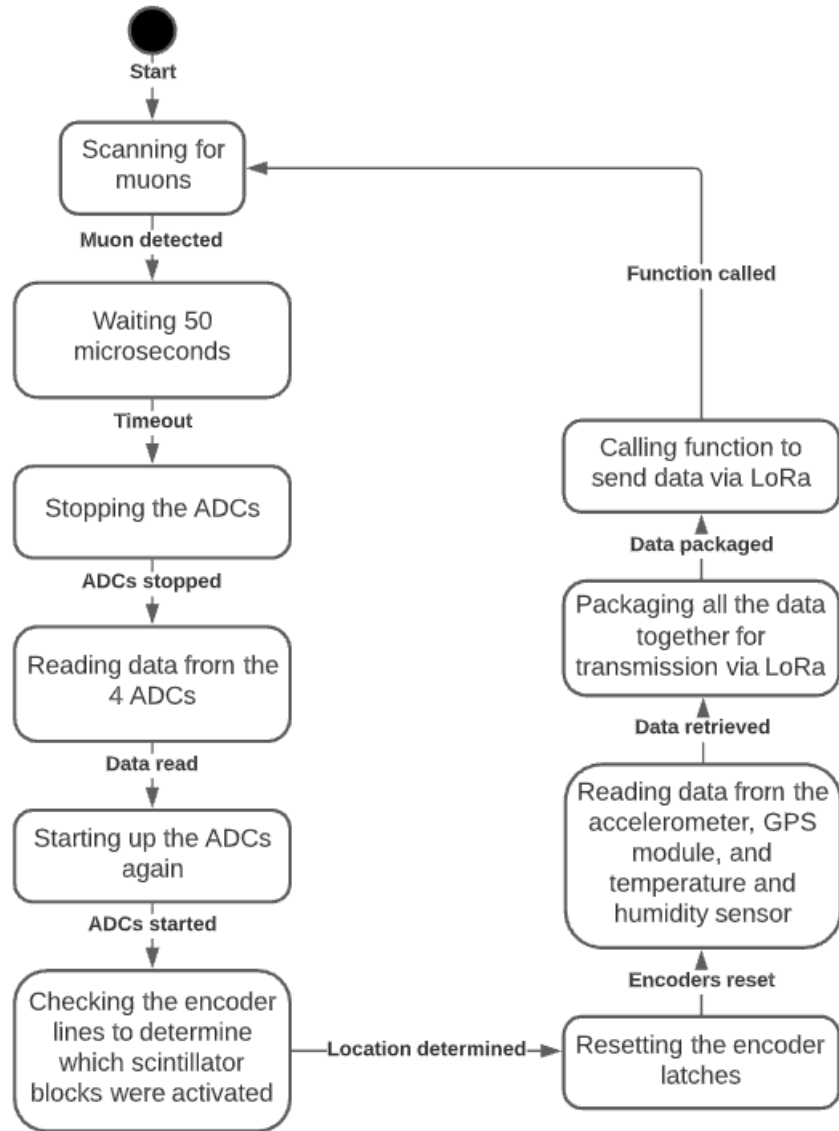


Figure 4.5: FSM of the control program

The program starts off in a loop where it continuously checks the OR gates from the daughter board to see if any of them are high which indicates that a muon hit the scintillator. When an event is detected, the microcontroller has to wait for approximately $50 \mu\text{s}$. This will give enough time for the pulse shaping circuits to extend the pulse, as well as giving the ADCs the opportunity to convert the 10 samples from the pulse. After those $50 \mu\text{s}$ when the ADCs have finished the conversions, they need to be deactivated so that they do not push more values into the FIFO which might cause the desired pulse

shaping output values to be lost. All the data stored in the FIFOs of the ADCs will then be read and stored, and thereafter the ADCs will be started again so that they are ready to sample new pulses as soon as they arrive. Then, the encoder lines will be checked to determine which scintillator blocks were activated in order to find where exactly the muon hit the scintillator. This will allow the microcontroller to calculate the trajectory of the muon, and together with the GPS coordinates, the exact geographical coordinates of where the muon hit the scintillator can be calculated. Thereafter, the angle of inclination of the scintillator plane will be determined using the accelerometer, the GPS coordinates will be read from the GPS module, and the temperature and humidity will be read from the temperature and humidity sensor. All of this data that was read from the peripherals and calculated (the trajectory and exact position of the muon) will be transferred to a remote computer using LoRa. And finally, the program returns to the loop where it continues checking for new events from the muon detector subsystem.

Chapter 5

Implementation

5.1 Schematic

The following sections follow the schematic of the PCB for the microcontroller subsystem and the major decisions that were made about the connections between the components. Any calculations used to work out component values will also be covered.

5.1.1 Power supply

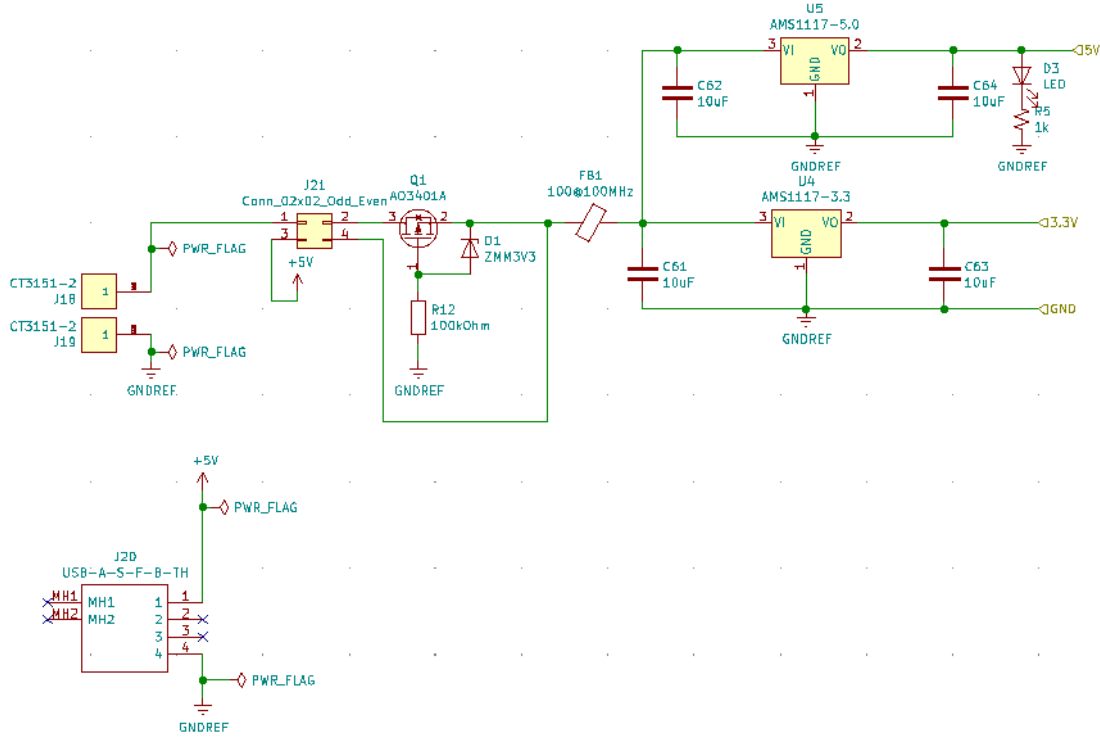


Figure 5.1: Power supply of the PCB

There are two ways to power the PCB - through banana connectors or through USB. The USB connection is convenient for programming the microcontroller outside an electronics lab as it can simply be powered by a laptop. The banana connectors are what will be used when the project will be deployed in the field. A 29V input will then power the system. The USB connector is a standard female USB-A connector that is available from JLCPCB. However, the CT3151-2 connectors are only available from DigiKey and no equivalent connectors were found from JLCPCB, and so these will have to be ordered separately from DigiKey and manually soldered.

To select between the banana connector power supply and USB power, a header is used. The appropriate pins can just be connected for the desired power supply.

The power supply circuit also includes reverse polarity protection on the banana connectors using a p-channel enhancement type MOSFET in case the input jacks are swapped by mistake. Using a MOSFET solution instead of using a simple diode is more energy efficient because the MOSFET does not have a significant voltage drop like a diode does [35]. Because USB-A inherently enforces polarity protection by only allowing the USB connector to connect with a specific orientation, the reverse polarity protection circuit is

not necessary and can be chosen to be bypassed if USB power is used.

Next, a ferrite bead is included to remove unwanted high frequencies from the circuit. Then, the components that follow thereafter are the voltage regulators. The AMS1117-5.0 and the AMS1117-3.3 are used to regulate the voltage supply to 5V and 3.3V respectively. They were selected because of their popularity and because of their use in PCBs that were examined as reference during the development of the schematic.

The final component included in the power supply circuit is an LED to indicate that the power has been connected successfully. It is connected in series with a resistor to prevent it from burning out.

5.1.2 Peripherals

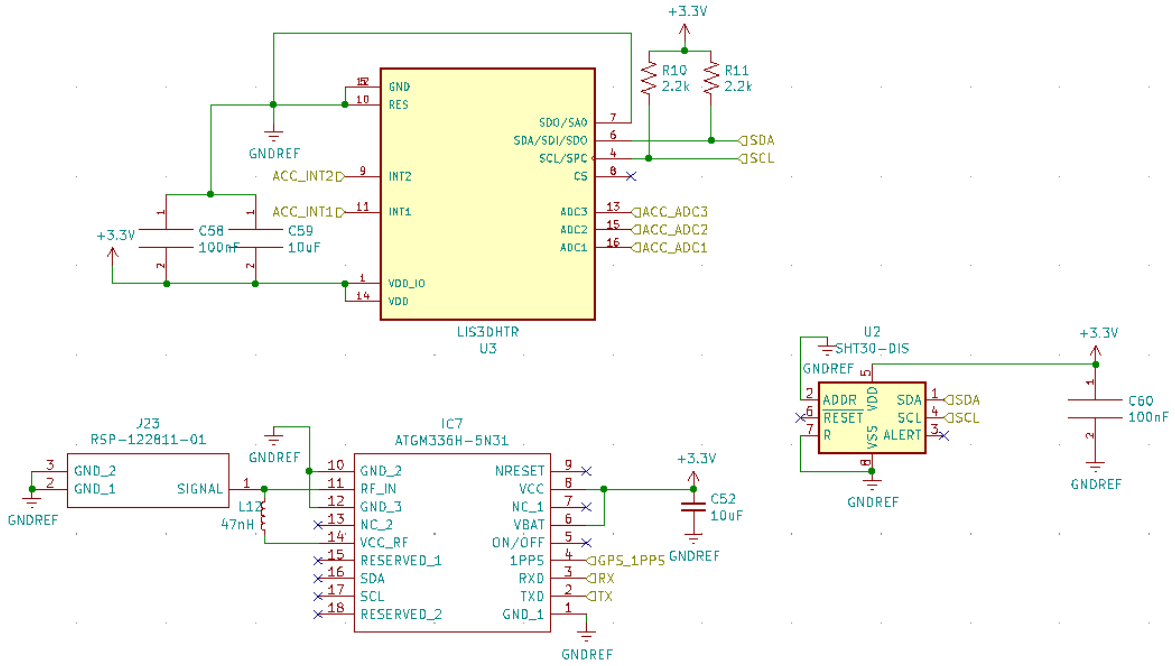


Figure 5.2: Peripherals on the PCB

The three peripherals on the PCB - accelerometer (LIS3DHTR), GPS module (ATGM336H-5N31), and temperature and humidity sensor (SHT30-DIS) - will be discussed in this section. The peripherals mostly have their connections similar to that which is recommended in their respective data sheets with the correct voltage and ground connections as well as the appropriate decoupling capacitors. Any unused pins of interest such as the three input pins of the auxiliary ADC of the accelerometer as well as its two interrupt

pins and the 1PPS pin from the GPS module are all connected to a header so that they can be accessed externally when needed.

The accelerometer and temperature and humidity sensor share the same I²C interface because of the limited I²C interfaces on the microcontroller. The GPS module has its own separate UART interface to the microcontroller.

The GPS module requires an external antenna and therefore the KYOCERA AVX GPS Antenna was chosen. This antenna uses an IPEX MHF connector which the GPS module is connected to using pins 11 and 14 as shown in Figure 5.2.

5.1.3 IO expanders

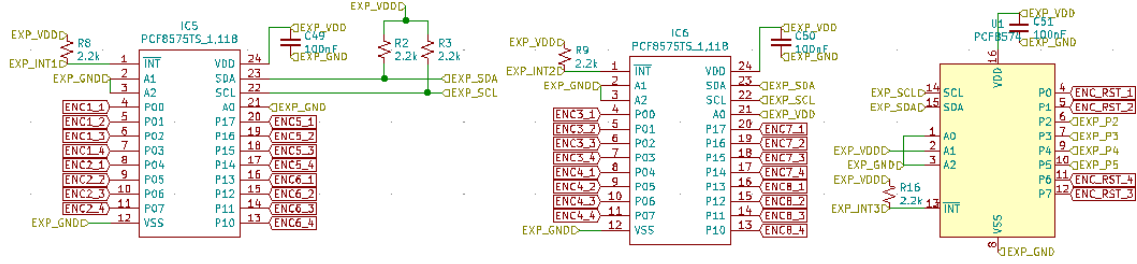


Figure 5.3: IO expanders on the PCB

Because of the shortage of pins on the microcontroller, IO expanders (PCF8575 and PCF8574) were chosen to interface with the 36 pins from the encoder section from the daughter board. The IO expanders have a useful feature of allowing one to change the address of the devices when multiple devices are used on the same I²C interface, which is the case in this design. This is done through hardware by setting the A0, A1, and A2 pins of the IO expanders high or low. The addresses for these devices have a leading 0100 and so the address for a particular device is 0100[A2][A1][A0]. The values that were selected for the devices, which corresponds to the IO expanders in Figure 5.3 from left to right, are shown below:

Device	A2	A1	A0	Address
PCF8575 (1)	0	0	0	01000000
PCF8575 (2)	0	0	1	01000001
PCF8574	0	1	0	01000010

Table 5.1: Address selection of the IO expanders

It can be seen that these values correspond with the connections in Figure 5.3. A2, A1, and A0 are pins 3, 2, and 21 respectively on the PCF8575 IO expanders, but pins 3, 2, and 1 respectively on the PCF8574 IO expander.

5.1.4 LoRa module

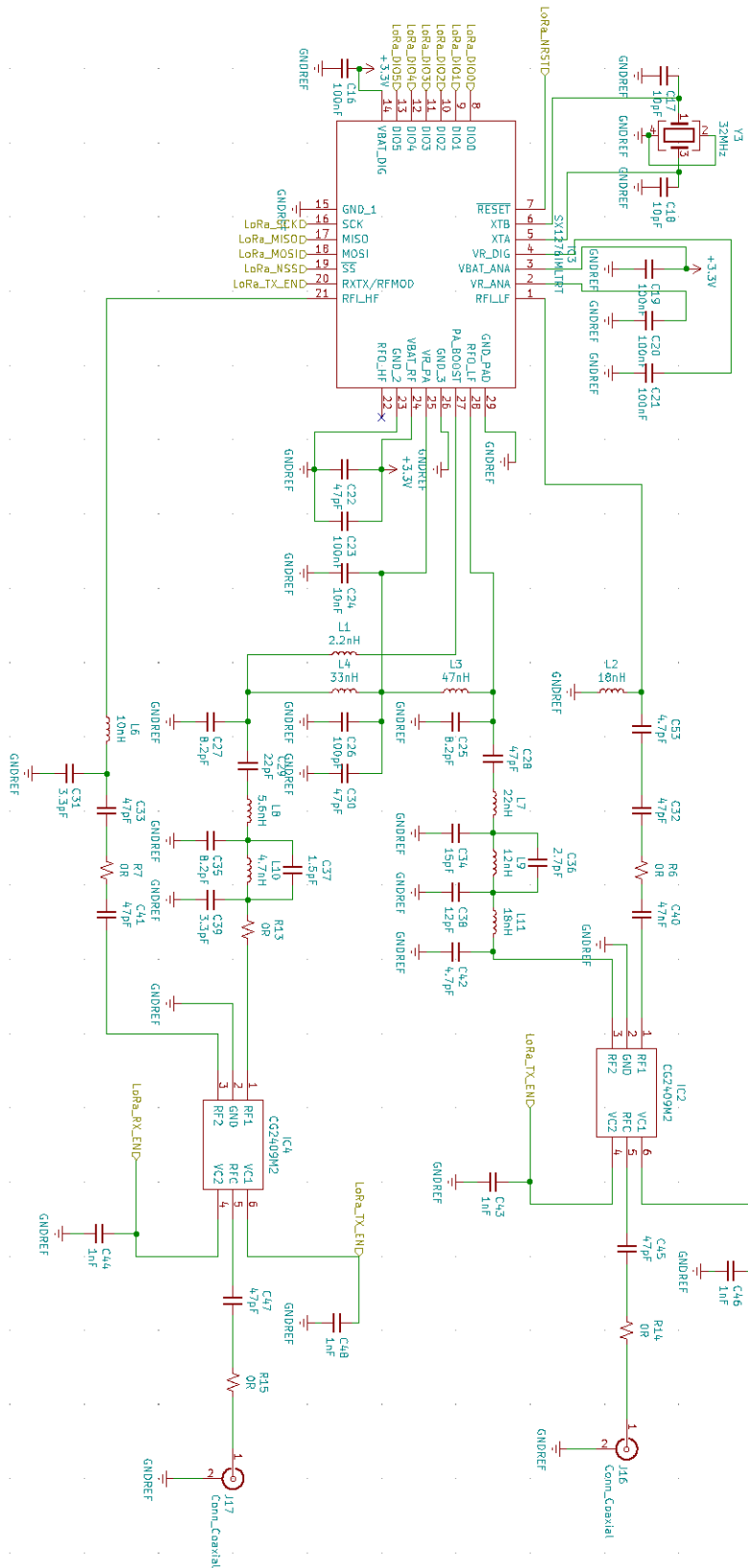


Figure 5.4: Schematic of the LoRa module

The above figure shows the LoRa module (SX1276) and how its pins are connected as well as the impedance matching circuits and filters that connect the module to the antennas. This schematic was drawn up with the aid of reference designs from Semtech which is the manufacturer of LoRa modules. The SX1276RF1JAS reference design was looked at specifically [36]. The configuration registers of the LoRa module are controlled using SPI, while the transmitted and received data as well as various states and clocks of the module are communicated using the digital IO (DIO) pins.

The LoRa module has three power amplifiers. By using the RFO_LP and RFO_HP, up to +14dBm can be delivered. But the third power amplifier, PA_BOOST, can deliver up to +20dBm. However, in the reference design the RFO_HP pin is unused, and so the RFO_LP and RFL_LP are used as the low-power transceiver pair, while the PA_BOOST and RFL_HP are used as the high-power transceiver pair. To switch the module to the transmitting state, the LoRa_TX_EN pin on the microcontroller must be set high while the LoRa_RX_EN pin must be set low. This is the opposite for when the module is in the receiving state where the LoRa_RX_EN pin is set high and the LoRa_TX_EN pin is set low.

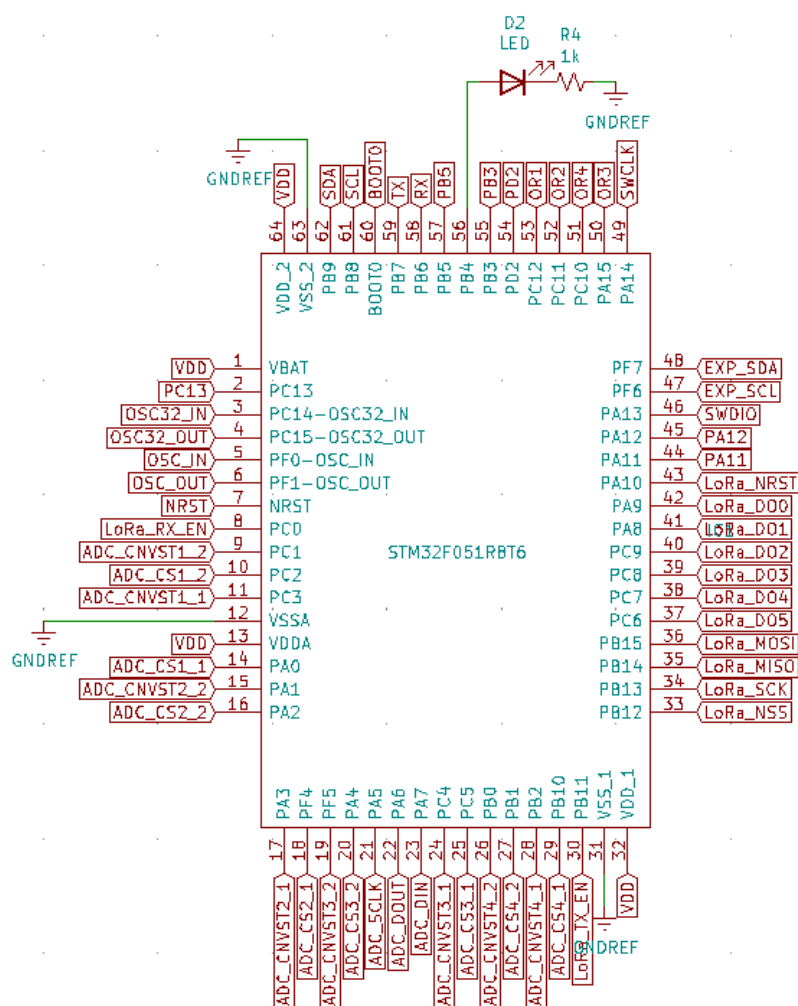


Figure 5.5: Connections of the microcontroller

The figure above shows how all of the pins of the microcontroller are connected with the aid of the STM32F051 datasheet [25]. This arrangement of pins was selected to optimize the routing of the PCB. The connections for the peripherals, such as the accelerometer and temperature and humidity sensor which use I²C and the GPS module which uses UART, are at the top of the microcontroller. The I²C connections for the IO expanders together with the OR gates from the daughter board are at the top right. The majority of the LoRa connections are on the right side. The ADC connections are mostly at the bottom of the microcontroller and some are on the left. And then finally, the other major connections on the left are for the 32 MHz (OSC) and 32.768 kHz (OSC32) oscillators. During the process of assigning these pins, the pins for the power connections were done first, followed by the connections for the communication interfaces. The remaining pins were assigned to the more basic connections such as the OR gates which are simple logic

level inputs. An LED was added for testing to check if the GPIO works as expected.

5.1.6 NRST and BOOT

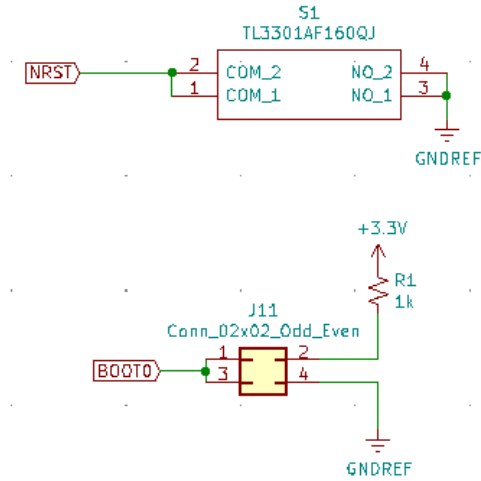


Figure 5.6: NRST and BOOT connections

The NRST (not reset) pin must be pulled low to signal a reset to the microcontroller. As can be seen in the above figure, the NRST pin is connected to a tactile push button which will connect the NRST to GND when pressed.

The BOOT0 controls the source from which the microcontroller boots. If the BOOT0 pin is set low (connecting pin 4 to either pin 1 or 3), then the user flash memory is selected as the boot space. If the BOOT0 pin is set high (connecting pin 2 to either pin 1 or 3), then the microcontroller activates the bootloader in system memory [37].

5.1.7 ST-Link

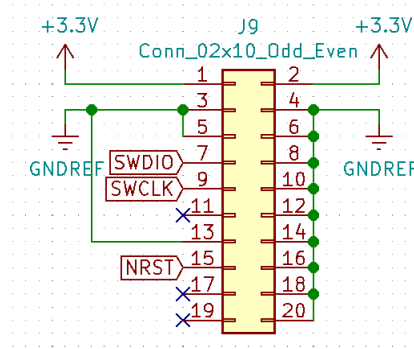


Figure 5.7: ST-Link header

To program the microcontroller through SWD, the ST-Link/V2 is used. A 2x10 ribbon cable is used to connect the ST-Link to the 2.54 mm pitch male header shown above. The SWDIO, SWCLK, and NRST pins connect to the microcontroller and are the major pins for programming the microcontroller.

5.1.8 Oscillators

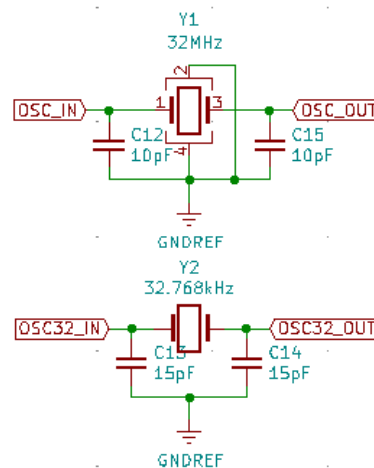


Figure 5.8: Oscillators for the microcontroller

The 32 MHz crystal (NX2016SA) is the main external oscillator, while the 32.768 kHz crystal (SC-20S) is the oscillator for the internal RTC. The following equation is used to determine the values of the external capacitors [38]:

$$C_L = \frac{C_1 \times C_2}{C_1 + C_2} + C_s \quad (5.1)$$

C_L is the load capacitance specified by the manufacturer, and C_1 and C_2 are the external capacitors that are connected to the sides of the oscillator. C_S is the stray capacitance in the PCB which can range between 2 pF and 8 pF [38].

To calculate the values of the capacitors for the 32 MHz crystal, the load capacitance was needed from the NX2016SA datasheet which was found to be 8 pF. By using a stray capacitance of 3 pF, and using the same capacitor value for C_1 and C_2 , then the value of the external capacitors can be calculated as follows:

$$C_L = \frac{C_1 \times C_2}{C_1 + C_2} + C_S$$

$$C_L = \frac{C_1 \times C_1}{C_1 + C_1} + C_S \quad (C_2 = C_1)$$

$$C_L = \frac{C_1^2}{2 \times C_1} + C_S$$

$$C_L = \frac{C_1}{2} + C_S$$

$$8 = \frac{C_1}{2} + 3$$

$$C_1 = 10 \text{ pF}$$

A similar process is followed to find the capacitance for the 32.768 kHz crystal. A load capacitance value of 12.5 pF was specified by the manufacturer. By using a value of 5 pF for the stray capacitance, and the derivation in the calculations above, the value of the external capacitors can be calculated as follows:

$$C_L = \frac{C_1}{2} + C_S$$

$$12.5 = \frac{C_1}{2} + 5$$

$$C_1 = 15 \text{ pF}$$

5.1.9 Interface with the daughter board

For the PCB of the microcontroller subsystem to interface with the daughter board, JST GH connectors were chosen. The footprint of a 9-pin JST GH header appears as follows:

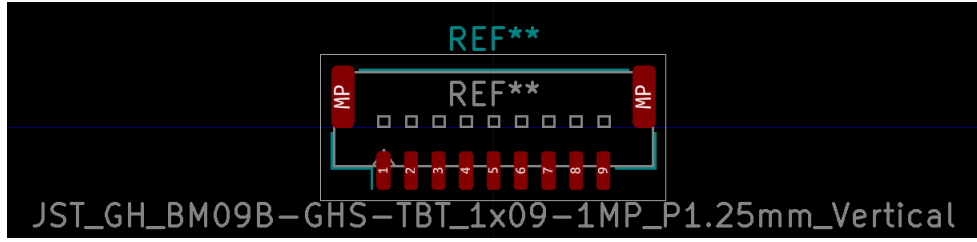


Figure 5.9: JST GH header footprint

A 9-pin JST GH header would appear as follows in an assembled PCB:

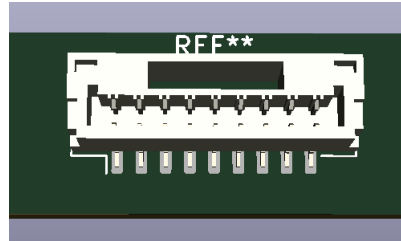


Figure 5.10: 3D view of the JST GH header

The pins are ordered from pin 1 to 9 from left to right in the figures above. The following tables describe the pins that are used for each JST GH header that is used to interface with the daughter board. Take note that there are four JST GH headers on the PCB for each header described because of the use of four daughter boards for the four scintillator planes in the muon detector subsystem.

10-pin JST GH header for the encoder section:

Pin#	Name	Function
1	OR1	Trigger that a muon was detected
2	ENC_RST1	Resets the latch connected to the encoders
3	ENC1_1	Encoder set 1 bit 0
4	ENC1_2	Encoder set 1 bit 1
5	ENC1_3	Encoder set 1 bit 2
6	ENC1_4	Encoder set 1 bit 3
7	ENC2_1	Encoder set 2 bit 0
8	ENC2_2	Encoder set 2 bit 1
9	ENC2_3	Encoder set 2 bit 2
10	ENC2_4	Encoder set 2 bit 3

Table 5.2: 10-pin JST GH header pin description

9-pin JST GH header for the ADC section:

Pin#	Name	Function
1	ADC_CNVST1_2	Active-low conversion start pin 2
2	ADC_CS1_2	Chip select 2
3	EOC1_2	End of conversion output 2. Conversion ready when pulled low.
4	ADC_CNVST1_1	Active-low conversion start pin 1
5	ADC_CS1_1	Chip select 1
6	ADC_SCLK	Clock pin for SPI
7	ADC_DIN	MOSI pin for SPI
8	ADC_DOUT	MISO pin for SPI
9	EOC1_1	End of conversion output 1. Conversion ready when pulled low.

Table 5.3: 9-pin JST GH header pin description

3-pin JST GH header for the power supply:

Pin#	Name	Function
1	GND	Ground connection
2	3.3V	Main power supply
3	5V	Power supply for ADCs

Table 5.4: 3-pin JST GH header pin description

5.1.10 Other pin headers

Any pins from the microcontroller or the peripherals that are unused but would still be helpful to be able to access are connected to headers. There are two headers that have this function. The first header is a 19-pin header with all the unused pins from the microcontroller, accelerometer, GPS module, and IO expanders are connected. The other header which is an 8-pin header is for the unused pins from the ADCs on the daughter board. The following tables describe the pins of these headers. The pins will be labelled on the silkscreen of the PCB.

19-pin 2.54 mm pitch male header:

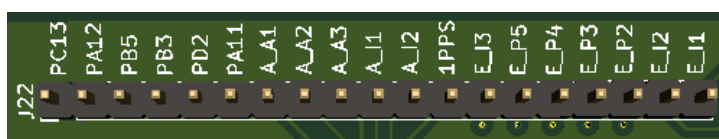


Figure 5.11: 19-pin 2.54 mm pitch male header appearance on the PCB

Pin#	Silkscreen name	Name	Function
1	PC13	PC13	GPIO pin PC13 on the microcontroller
2	PA12	PA12	GPIO pin PA12 on the microcontroller
3	PB5	PB5	GPIO pin PB5 on the microcontroller
4	PB3	PB3	GPIO pin PB3 on the microcontroller
5	PD2	PD2	GPIO pin PD2 on the microcontroller
6	PA11	PA11	GPIO pin PA11 on the microcontroller
7	A_A1	ACC_ADC1	ADC input 1 of the accelerometer
8	A_A2	ACC_ADC2	ADC input 2 of the accelerometer
9	A_A3	ACC_ADC3	ADC input 3 of the accelerometer
10	A_I1	ACC_INT1	Interrupt pin 1 of the accelerometer
11	A_I2	ACC_INT2	Interrupt pin 2 of the accelerometer
12	1PPS	GPS_1PPS	GPS pin which produces one pulse per second with a duration of 100 ms.
13	E_I3	EXP_INT3	Interrupt pin of IO Expander 3 (PCF8574)
14	E_P5	EXP_P5	IO pin P5 of IO Expander 3 (PCF8574)
15	E_P4	EXP_P4	IO pin P4 of IO Expander 3 (PCF8574)
16	E_P3	EXP_P3	IO pin P3 of IO Expander 3 (PCF8574)
17	E_P2	EXP_P2	IO pin P2 of IO Expander 3 (PCF8574)
18	E_I2	EXP_INT2	Interrupt pin of IO Expander 2 (PCF8575 (2))
19	E_I1	EXP_INT1	Interrupt pin of IO Expander 1 (PCF8575 (1))

Table 5.5: 19-pin 2.54 mm pitch male header pin description

8-pin 2.54 mm pitch male header:

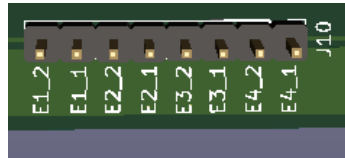


Figure 5.12: 8-pin 2.54 mm pitch male header appearance on the PCB

Due to lack of space, it was chosen to label the pins as ‘E1_1’ instead of using the form ‘EOC1_1’.

Pin#	Name	Function
1	EOC4.1	End of conversion pin for ADC 1 on daughter board 4
2	EOC4.2	End of conversion pin for ADC 2 on daughter board 4
3	EOC3.1	End of conversion pin for ADC 1 on daughter board 3
4	EOC3.2	End of conversion pin for ADC 2 on daughter board 3
5	EOC2.1	End of conversion pin for ADC 1 on daughter board 2
6	EOC2.2	End of conversion pin for ADC 2 on daughter board 2
7	EOC1.1	End of conversion pin for ADC 1 on daughter board 1
8	EOC1.2	End of conversion pin for ADC 2 on daughter board 1

Table 5.6: 8-pin 2.54 mm pitch male header pin description

5.2 PCB layout

The following sections examine the major parts of the PCB layout that are of interest. The thinking behind the placement and routing choices will be described. The full views of the PCB layers can be found in Appendix A.2.

5.2.1 Microcontroller

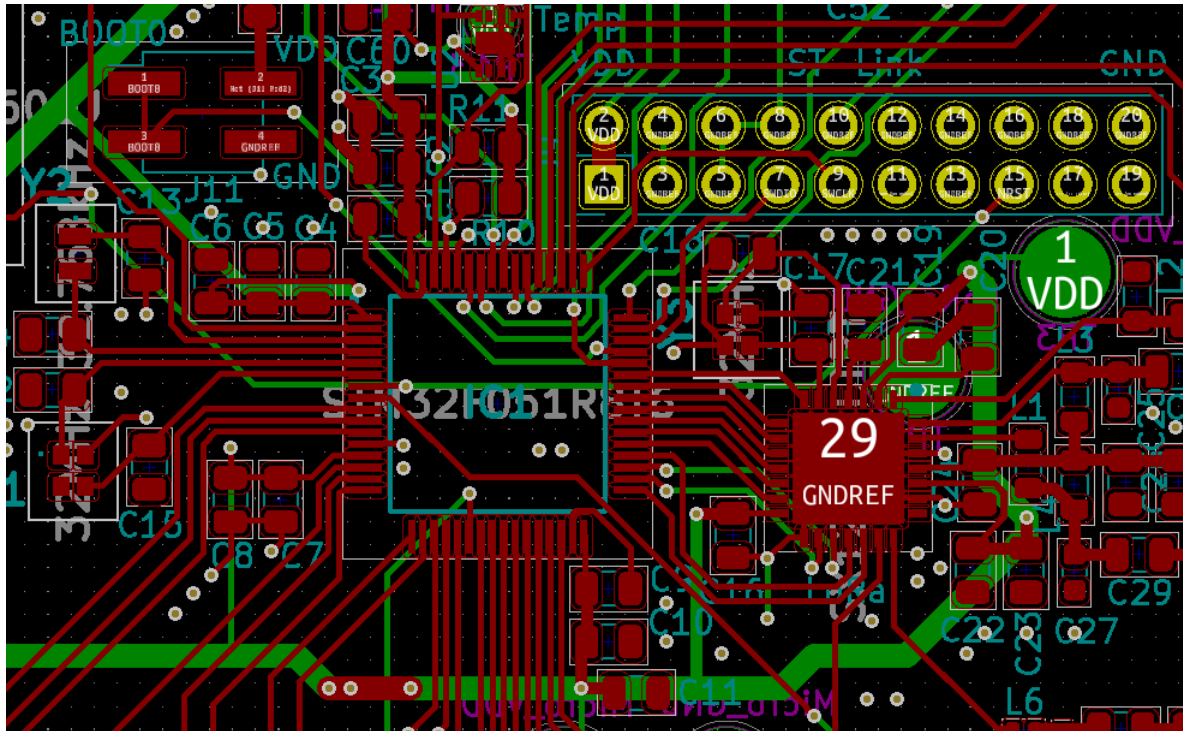


Figure 5.13: Microcontroller routing on the PCB

The microcontroller was the first component to be routed because of its vital role in the PCB. The decoupling capacitors were placed as close as possible to the microcontroller and from there, the rest of the components were placed around the microcontroller depending on where their connections were. For example, the accelerometer and temperature and humidity sensor were placed at the top of the microcontroller as explained in Section 5.1.5. Although it is not ideal for some of the copper tracks to be so close to each other, because of how close the pins of the microcontroller are to each other, there was no other choice.

The 2x10 2.54 mm pitch male header for the ST-Link ribbon cable can be seen at the top right in the image above. It was placed as close as possible to the microcontroller because of its great importance to have a usable PCB.

At the top left, the 2x2 header can be seen for the BOOT0 pin. The VDD and GND pins are labelled on the silkscreen so that the pins cannot be confused and that it is not necessary to rely too much on the documentation of the PCB.

5.2.2 LoRa

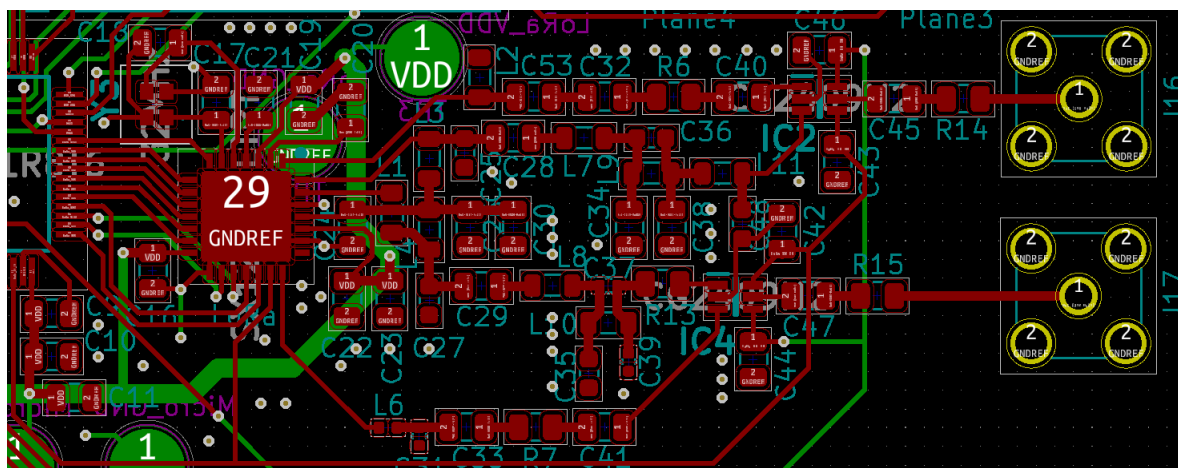


Figure 5.14: LoRa routing on the PCB

Much care had to be given to do the routing for the LoRa section. Using the AN5407 application note from STMicroelectronics, it was ensured that the copper tracks for the impedance matching circuits and filters were as straight as possible to reduce the possibility of reflections. In the places where there were bends, it would be ideal to have used rounded tracks with the same width instead. Unfortunately, these types of tracks were not available on KiCad, and external libraries did not end up working successfully.

The power connections were all placed at the top right of the PCB. The banana connectors can be seen at the very top left with the USB-A connector below them. All the tracks for the power supply were made 1 mm thick which is much thicker than the standard 0.25 mm tracks. This was done for better current flow for these tracks because they are vital for supplying power to not only this PCB, but also to the four daughter boards. The four 3-pin JST GH pins that are used to power the daughter boards can be seen at the bottom of the figure above. And because of the open space, the NRST tactile push button was placed right next to the BOOT0 header.

5.2.4 Accelerometer and temperature sensor

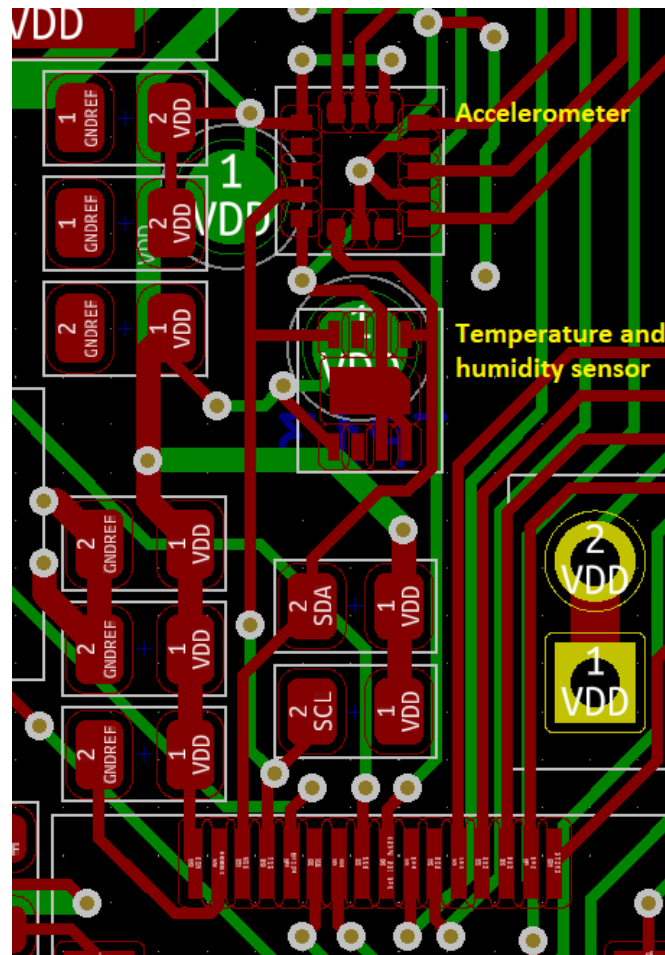


Figure 5.16: Accelerometer and temperature sensor routing on the PCB

The above image shows the routing for the temperature and humidity sensor as well as the accelerometer which proved to be one of the more difficult sections to route on the PCB. Because of their compact placement close the microcontroller and other components,

together with the fact that these components needed to share the SDA and SCL lines, it created a bit of a challenge to do the routing.

5.2.5 IO expanders

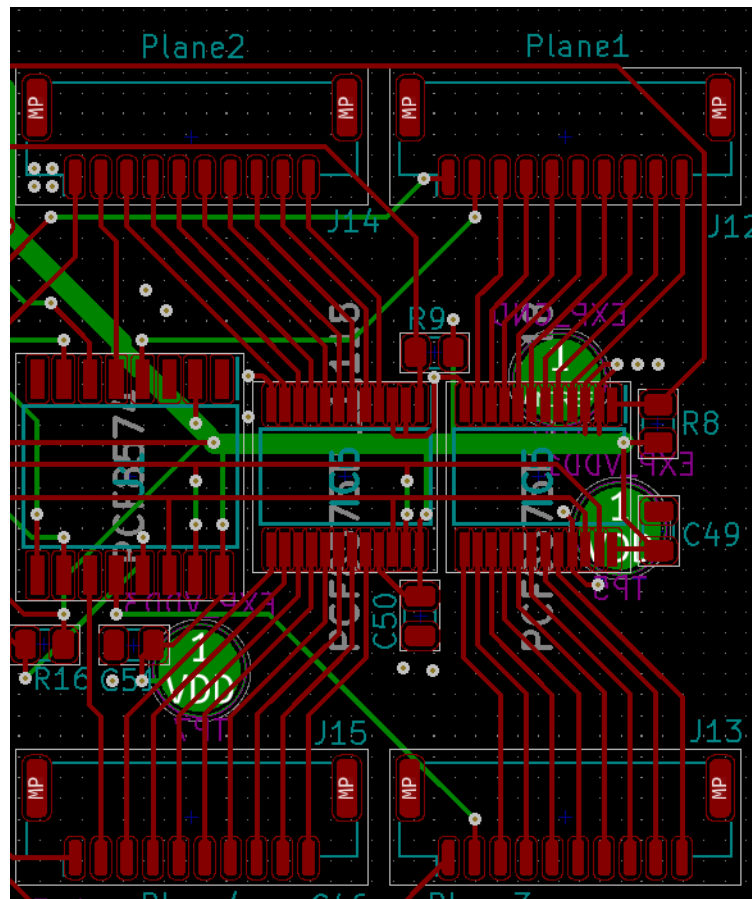


Figure 5.17: IO expanders routing on the PCB

The figure above shows the routing of the three IO expanders. This is an example of how tight spacing between tracks was avoided and how the tracks were placed a substantial distance away from each other. As can be seen in the image above and some of the previous images, test points were placed at the bottom layer of the PCB to aid with debugging. It was mostly the ground and supply pins to components that were connected to the test points. This is because, in the case that a component is not operating properly, it can be checked whether that component is getting power which will help in determining where the source of the problem could be. If the component is receiving power, but still does not function properly, it could be a sign that the problem lies with the component itself. But if it is found that the component is not getting power, then there could be

5.2.6 VDD



During the first attempt at the design, it was decided to make the bottom plane the 3.3V supply to make it easier to reach all the components that need power. However, this only made the connection of the ground pins (which are much more in number than the power pins) much more difficult and goes against the recommendation of the AN5407 application note [39]. So instead, the power was supplied by using large 1 mm tracks to supply all the required parts of the circuit.

5.2.7 Issues with initial PCB designs

Acid traps



Figure 5.19: Acid trap formed by copper tracks

The initial versions of the PCB designs contained many acid traps such as the one above. This is a problem because over the course of a few months, the corrosive material used to etch out the tracks on the PCB can erode through copper tracks causing broken connections. To understand why this happens, the process of creating the copper tracks will be explained briefly.

To start off, the core of the PCB, which is a non-conductive material such as fiberglass, has both sides completely covered with copper layers. Then, the layout of the tracks and pads are “printed” on the board with a protective layer that prevents the desired copper from being removed. Thereafter, the layers of the PCB are washed in corrosive chemicals that remove or “etch” the unwanted copper [40].

The problem with copper tracks meeting at an acute angle as shown above, is that some of these corrosive chemicals can get trapped in the area where those tracks meet. And then over time, these chemicals will continue eroding away slowly at those tracks which will eventually cause a broken connection and the PCB might fail [40]. And so, it is better if the tracks are always at obtuse angles or at least do not meet at less than 90°.

For this reason, these acid traps had to be fixed later on in the design to prevent this erosion from happening.

Track spacing

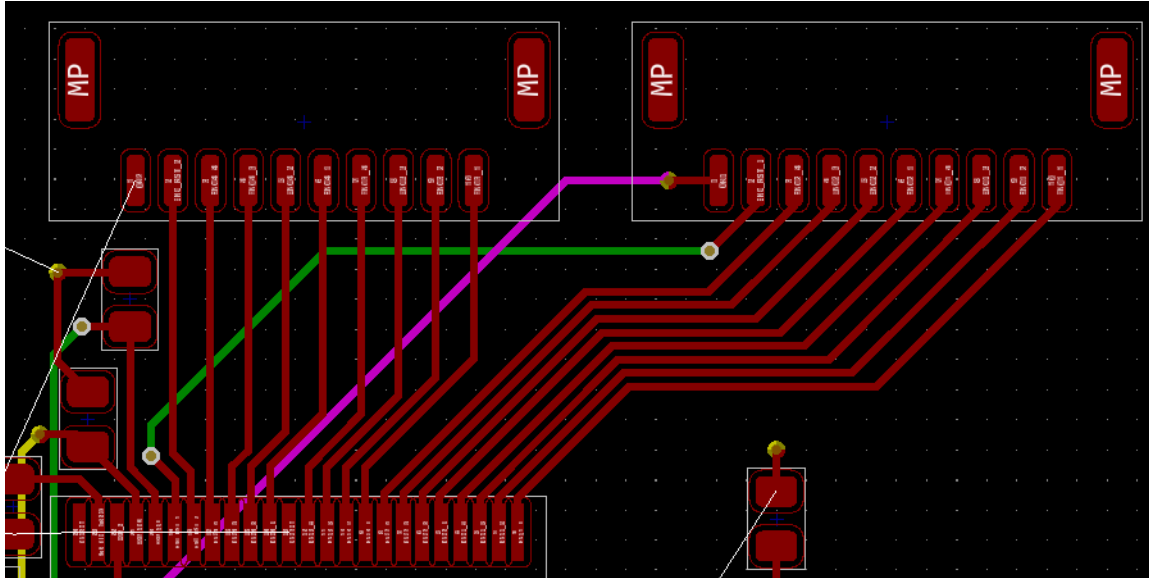


Figure 5.20: An initial design with poor track spacing

In initial designs, the tracks were often placed too close to each other as in the figure above. This can be a problem because there is a greater chance for these tracks to form a short circuit. It was unnecessary to have the tracks that close to each other. Sometimes there is no other choice which was the situation in Figure 5.13. But in this case, something similar to Figure 5.17 could have been done. It was ensured that in later designs the tracks had sufficient spacing.

5.3 Timing

In this section, the time taken from the moment a muon is detected by a scintillator block to the end of the process where data is sent via LoRa is calculated.

5.3.1 Timing of the daughter board

The following information, represented as a diagram, was received from Pohl which describes the timing in the daughter board:

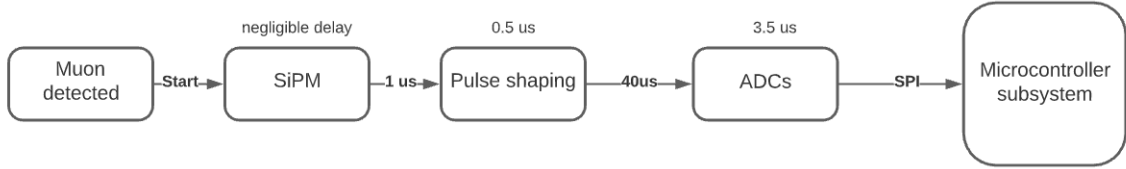


Figure 5.21: Timing diagram of the daughter board

The diagram starts with a muon detected by the scintillator. The luminescence generated from a scintillator block then passes to a SiPM which converts it into an electrical signal. The signal then passes to a pulse shaper which extends the signal. The extended signal then travels to an ADC where it is sampled and converted to a digital form. The values above the boxes indicate the amount of time a component takes to process a signal, and the value on the arrow indicates the amount of time to send that signal to the next device. The subsequent timing values will be calculated starting from the time it takes to retrieve data from the ADCs.

5.3.2 Timing of the ADCs

The maximum SPI clock speed the STM32F051 allows is 18 MHz [25]. However, the nominal speed of the internal clock of the ADC limits the maximum SPI clock speed to 4.4 MHz [41]. In the worst case, all 8 ADCs will be activated. Because it is a 12-bit ADC, and SPI communication sends 8 bits at a time, a total of 16 bits will have to be sent to transfer the data. The additional start and stop bit for every communication that takes place must also be taken into account which brings it to a total of 18 bits that are transferred for every sample of data. It must also be taken into account that 16 samples must be read for each ADC since that is the size of the FIFO and the desired data can exist between any of those samples. Using this information, the time to transfer all the

data from the ADCs can be calculated as follows:

$$\begin{aligned}\text{Clock period} &= \frac{1}{\text{frequency}} \\ &= \frac{1}{4.4 \text{ MHz}} \\ &= 0.227 \text{ } \mu\text{s}\end{aligned}$$

$$\text{Total number of bits} = 18 \text{ bits} \times 16 \text{ samples} \times 8 \text{ ADCs} = 2304 \text{ bits}$$

$$\begin{aligned}\text{Total time} &= \text{number of bits} \times \text{clock period} \\ &= 2304 \times 0.227 \text{ } \mu\text{s} \\ &= 0.524 \text{ ms}\end{aligned}$$

However, the total amount of ADC data that will need to be sent via LoRa, which excludes the start and stop bits, can be calculated as follows:

$$\text{Total number of bits} = 16 \text{ bits} \times 16 \text{ samples} \times 8 \text{ ADCs} = 2048 \text{ bits}$$

5.3.3 Timing of the IO expanders

The next step is to calculate the time it takes to read from the IO expanders that connect to the encoder section. Because of the significant amount of time taken to extend the pulse and read from the ADCs, the time taken for the signal to pass through an encoder and a latch is negligible. By the time the encoders need to be read from, the data will be available.

In the worst case, all encoders will have to be read from. Because there are 8 encoder pins per daughter board and a total of 4 boards, 32 pins will have to be read. The PCF8575 IO expanders are the only ones that will be read from to get the values of the encoder pins. The PCF8574 is only used for the reset pins of the latches.

The I²C communication on the PCF8575 works as follows. A start bit is sent by having the SDA line go low while the SCL line is high. To set the IO expander to write mode, the bit after the 7-bit slave address for the specific IO expander is set to a zero. To read, which would be the case in this design, the last bit must be set to a one. So after these 8-bits are sent, two bytes (16 bits) are sent from the slave device with each bit

corresponding to the value on a specific IO pin. After each byte sent, an acknowledge bit is sent from either the master or the slave, depending on which one is receiving data. This means that a total of 3 acknowledge bits are sent. Then, at the end of the communication, a stop bit is sent. Adding up the 3 bytes, 3 acknowledge bits, and the start and stop bits, a total of 29 bits need to be sent to read the values of one IO expander. And because there are 2 IO expanders that will have to read from, a total of 58 bits need to be sent to read from all 32 encoder lines.

Because the PCF8575 has a maximum transfer rate of 400 kbps, the time taken to read the all the encoder lines can be calculated as follows:

$$\begin{aligned}
 \text{Total time} &= \frac{\text{size of data}}{\text{transfer rate}} \\
 &= \frac{58}{400 \text{ kbps}} \\
 &= 145 \mu s
 \end{aligned}$$

5.3.4 Timing of the accelerometer

The next step is calculate the time taken to read from the accelerometer. The I²C communication on the LIS3DHTR works as follows. Similar to the IO expanders, start and stop bits must be sent as well as acknowledge bits after every byte of data transferred. The first byte sent is the slave address of the LIS3DHTR. Next, 8 bits for the specific register to be read from must be sent. To make communication faster, the MSB bit of the register address can be set to a one to allow the address to be automatically incremented to the next register address. Because of this, the next register address to be read from does not have to be specified by the master every time. For the x-, y-, and z-coordinates of the accelerometer, there are two register addresses for each coordinate meaning that there are 2 bytes for each coordinate. Because there are 3 coordinates, a total of 6 bytes (48 bits) need to be read. Therefore, the total number of bytes that need to be communicated when reading from the accelerometer is 8 bytes. And together with the start and stop bits as well as the 8 acknowledge bits, a total of 74 bits must be sent. And because the LIS3DHTR has a maximum transfer rate of 400 kbps, the time taken to read

the values from the accelerometer can be calculated as follows [30]:

$$\begin{aligned}
 \text{Total time} &= \frac{\text{size of data}}{\text{transfer rate}} \\
 &= \frac{74}{400 \text{ kbps}} \\
 &= 185 \mu s
 \end{aligned}$$

5.3.5 Timing of the temperature and humidity sensor

The I²C communication for the SHT30-DIS works as follows. As with the previous sections, a start and stop bit must be sent as well as an acknowledge bit after every byte sent. The first byte sent is the slave address for the SHT30-DIS. Next, two command bytes are sent to start the measurement for temperature and humidity. It then takes 4.5 ms (which is the fastest possible speed) to make the measurement. Then, two bytes for the temperature value are sent followed by another byte for the CRC checksum. Similarly for the humidity reading, two bytes must be sent for the value followed by one byte for the CRC checksum. As a result, a total of 9 bytes must be sent, together with 9 acknowledge bits, and the start and stop bit. This brings the total number of bits to 83. And because the SHT30-DIS has a maximum transfer rate of 400 kbps, the time taken to read the values from the temperature and humidity sensor can be calculated as follows [31]:

$$\begin{aligned}
 \text{Total time} &= \frac{\text{size of data}}{\text{transfer rate}} \\
 &= \frac{83}{400 \text{ kbps}} \\
 &= 207.5 \mu s
 \end{aligned}$$

Due to the time it takes to do a measurement, the total time taken to perform the communication with the temperature and humidity sensor is 4.7 ms (4.5 ms + 207.5 μ s).

5.3.6 Timing of the GPS module

The ATGM336H-5N uses the NMEA0183 protocol through UART. In this protocol, different types of messages are constantly being sent in a loop by the GPS device. Each message can have a maximum of 82 characters meaning that the messages have a maximum size of 82 bytes. Assuming that the ATGM336H-5N provides the 10 messages that are normally sent by GPS devices [42], there is an absolute maximum of 820 bytes

(6560 bits) that need to be sent before the desired information is found. Because the ATGM336H-5N has a default baud rate of 9600 bps, the transmission time can be calculated as follows:

$$\begin{aligned}\text{Total time} &= \frac{\text{size of data}}{\text{transfer rate}} \\ &= \frac{6560}{9600 \text{ bps}} \\ &= 683.3 \text{ ms}\end{aligned}$$

The only data required from the GPS to be transmitted via LoRa is the latitude and longitude. Each of these require 9 bits (8 bits for the coordinate, and one bit for either north/south or east/west), which brings it to a total of 18 bits.

5.3.7 Timing of the LoRa communication

The final stage of the timing is to send the data via LoRa. The following is a rough estimate of how much data needs to be sent via LoRa: 2048 bits from the ADCs; 32 bits from the encoders; 48 bits from the accelerometer; 32 bits from the temperature and humidity sensor (2 bytes for temperature and 2 bytes for humidity); and 18 bits for the GPS coordinates. This brings the total amount of data that needs to be sent via LoRa to 2178 bits. Using a standard LoRa transfer rate of 27 kbps [17], the total time it takes to send the data can be calculated as follows:

$$\begin{aligned}\text{Total time} &= \frac{\text{size of data}}{\text{transfer rate}} \\ &= \frac{2178}{27 \text{ kbps}} \\ &= 80.7 \text{ ms}\end{aligned}$$

5.3.8 Timing conclusion

Using the above information, the final diagram for the timing is as follows:

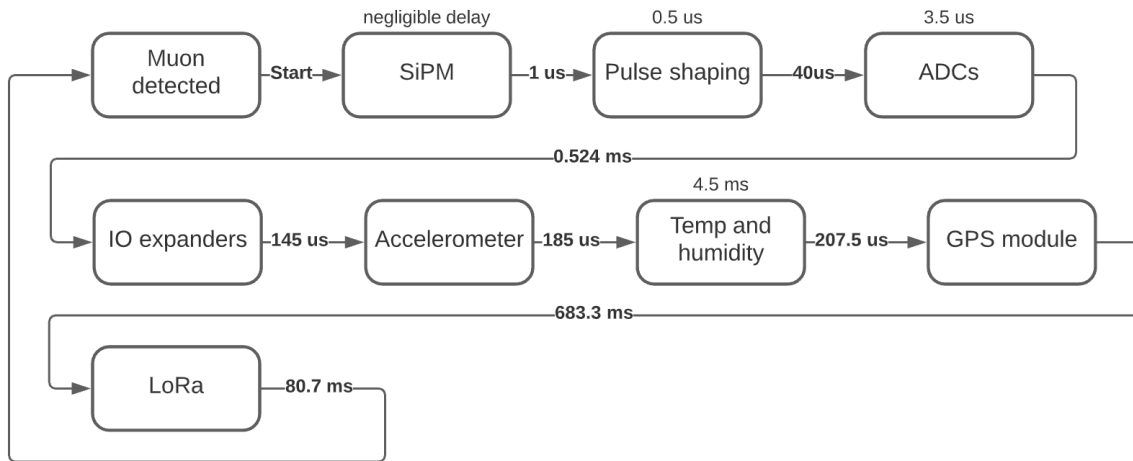


Figure 5.22: Final timing diagram

This diagram shows how long it takes from detecting a muon, to sending the data about the detection to a remote computer. The total time it takes can be calculated as follows by adding up the values of all the delays and time it takes to do the communication of data:

$$\begin{aligned}
 \text{Total time} &= 1 \mu\text{s} + 0.5 \mu\text{s} + 40 \mu\text{s} + 3.5 \mu\text{s} + 0.524 \text{ ms} + 145 \mu\text{s} + 185 \mu\text{s} + 4.5 \text{ ms} + \\
 &\quad 207.5 \mu\text{s} + 683.3 \text{ ms} + 80.7 \text{ ms} \\
 &= 769.61 \text{ ms}
 \end{aligned}$$

Chapter 6

Results

In terms of the PCB design, all the functional requirements laid out in the Methodology and the aims described in the Introduction were achieved by this project. A PCB for the microcontroller subsystem was designed that could interface with the muon detector subsystem as well as gather additional data surrounding the muon detection and be able to send this data to a remote computer. The PCB contains the appropriate hardware components to satisfy all the functional and non-functional requirements. The PCB has an appropriate microcontroller that satisfies requirement FR002 as it would be able to run a moderately sized program. The microcontroller also has enough interfaces available to support all the peripherals that connect to it. When it was found that the microcontroller would run out of pins, IO expanders were used to compensate for it. The LoRa module is of good quality and would be able to send all the required information to a distant computer, and thereby it satisfies requirement FR007. The accelerometer, temperature and humidity sensor, and GPS module are all of good quality and would be able to perform their functionality well which satisfies requirements FR004, FR005, and FR006. Through collaboration with Pohl, the interface of the microcontroller subsystem with the muon detector subsystem was determined which satisfies requirement FR002. Using the requirements set out by Pohl's design that an ADC must be selected that can sample at least 10 values of the output of the pulse shaping circuit, an appropriate ADC was found which satisfies requirement FR009. And through timing calculations, it was found that the time taken from the moment a muon is detected to gathering data from peripherals and sending it to a distant computer would take about 769.61 ms which satisfies requirement FR008 that this process should be completed in less than 1 second.

All the non-functional requirements laid out in the Methodology were achieved in this project. Standard components were chosen that were available from JLCPCB. The few

components that were not available could easily be ordered from DigiKey and RS Components. Through this, requirement NF001 is satisfied. The track spacing on the PCB was carefully examined and design rules were used to ensure that no tracks had a spacing of less than 0.2 mm, thereby satisfying requirement NF002. The same applies for the via to pad and via to track spacing which was ensured to be more than 0.2 mm which satisfies requirement NF003. Vias that have a diameter of 0.6 mm and a drill size of 0.3 mm were used throughout the PCB design. This satisfies requirement NF004 that the diameter of the vias should be at least 0.4 mm wide and have a drill size of at least 0.2 mm. And because only two layers were used for the PCB design, no blind or buried vias were used which satisfies requirement NF005.

Regrettably, because the project only had a duration of three months, not all of the requirements and practical work could be implemented. Due to the time restrictions and the amount of time it took to design the PCB, there was not sufficient time to have the PCB manufactured at JLCPCB. Because of this, the microcontroller subsystem could not be programmed, and so it could not be tested whether all the components work well enough to satisfy the requirements of the client, the physics team, that the system should be able to be used for muon imaging. So although functional requirements FR002, FR003, FR004, FR005, FR006, FR007, and FR009 were satisfied in terms of component selection and design, there was not an opportunity to implement the programming and use these components to ensure that they satisfy the requirements. And because the physical PCB was not manufactured, FR001 could not be met. Although the theoretical timing calculations do satisfy requirement FR008, it cannot be determined for certain if it will work out the same way if it is implemented in real life.

Chapter 7

Conclusion

The aim of the project was to design a PCB for a microcontroller subsystem which consists of a microcontroller and additional peripherals such as an accelerometer, GPS module, and temperature and humidity sensor with which the microcontroller should be able to communicate. In addition, the PCB had to be able to interface with the muon detector subsystem designed by a master's student, Alan Pohl, and then send the information about the muon detection, such as the GPS coordinates, trajectory of the muon, temperature, humidity, and angle of inclination of the scintillator, to a distant computer using LoRa. The system in its complete form would then be a fully developed muon detector that can be used to muon image Devil's Peak as per the requirements of the client, the team from the physics department. Through the development of the project, a complete PCB was designed which fulfilled the hardware requirements of the microcontroller subsystem. The PCB includes all the necessary components that are necessary to produce the microcontroller subsystem such as the microcontroller, accelerometer, GPS module, temperature and humidity sensor, and LoRa module. And through collaboration with Pohl, an interface between the microcontroller subsystem and the muon detector subsystem was designed. Due to time constraints, the PCB could not be manufactured and therefore not programmed, but it lays a solid foundation for future development to create a muon detector.

Chapter 8

Recommendations

The following recommendations are made for future development of the project:

- Use rounded tracks in the LoRa section of the PCB layout.
- Have the PCB manufactured.
- Program the microcontroller with the control program described in the Design section.
- Integrate the PCB with the muon detector subsystem and test the operation of the system as a whole.
- Use a dual-core microcontroller so that the microcontroller does not have to wait for the LoRa module to finish sending data before it checks for the arrival of another muon. The LoRa communication should run as its own process separate from the main control program.
- Include angle sensors and controlled motors to manipulate the orientation and pitch of the scintillator plane.
- Include a length sensor together with a controlled motor to manipulate the separation between the scintillator planes.

Bibliography

- [1] Britannica, *Muon — subatomic particle — Britannica*, <https://www.britannica.com/science/muon>, 2021. (visited on 08/14/2021).
- [2] M. Wolverton, “Muons for Peace,” *Scientific American*, vol. 297, no. 3, pp. 26–28, 2007.
- [3] D. Sciences, *MMPDS - The Next Generation of Non-Intrusive Inspection — Decision Sciences : Decision Sciences*, <https://decisionsciences.com/our-product/>. (visited on 08/17/2021).
- [4] L. T. Science, *What are Cosmic Rays? — Let’s Talk Science*, <https://letstalkscience.ca/educational-resources/backgrounders/what-are-cosmic-rays>, 2019.
- [5] C. J. Rhodes, “Muon tomography: looking inside dangerous places,” *Science Progress*, vol. 98, no. 3, pp. 291–299, 2015.
- [6] Dynasil, *What are Scintillators? How does Scintillation Work? - Hilger Crystals*, <https://www.dynasil.com/radiation-detection/what-are-scintillators-how-do-they-work/>. (visited on 10/30/2021).
- [7] S. Piatek, *What is an SiPM and how does it work? — Hamamatsu Photonics*, <https://hub.hamamatsu.com/jp/en/technical-note/how-sipm-works/index.html>, 2016.
- [8] L. W. Alvarez, J. A. Anderson, F. E. Bedwei, J. Burkhard, A. Fakhry, A. Girgis, A. Goneid, F. Hassan, D. Iverson, G. Lynch, Z. Miligy, A. H. Moussa, M. Sharkawi, and L. Yazolino, “Search for Hidden Chambers in the Pyramids,” *Science*, vol. 167, no. 3919, pp. 832–839, 1970.
- [9] J. Marchant, *Cosmic-ray particles reveal secret chamber in Egypt’s Great Pyramid*, <https://www.nature.com/articles/nature.2017.22939>, 2017. DOI: 10.1038/NATURE.2017.22939.
- [10] J. Devine, *Cosmic Pi - The Recipe — CosmicPi*, <http://cosmicpi.org/2019/11/01/Cosmic-Pi-Recipe.html>. (visited on 10/28/2021).

- [11] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, *Internet of Things (IoT): A vision, architectural elements, and future directions*, <http://dx.doi.org/10.1016/j.future.2013.01.010>, 2013. DOI: 10.1016/j.future.2013.01.010. arXiv: 1207.0203.
- [12] K. Ashton, *That ‘Internet of Things’ Thing — RFID JOURNAL*, <https://www.rfidjournal.com/that-internet-of-things-thing>, 2009.
- [13] V. Hayden and P. Ryabchuk, *How Can the Automotive Industry Use Internet of Things (IoT) Technology?* <https://www.intellias.com/how-can-the-automotive-industry-use-internet-of-things-iot-technology/>, 2018.
- [14] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, *Edge Computing: Vision and Challenges*, 2016. DOI: 10.1109/JIOT.2016.2579198.
- [15] IBM, *What Is Edge Computing - South Africa — IBM*, <https://www.ibm.com/za-en/cloud/what-is-edge-computing>. (visited on 08/29/2021).
- [16] K. Ha, Z. Chen, W. Hu, W. Richter, P. Pillai, and M. Satyanarayanan, *Towards Wearable Cognitive Assistance*, <http://dx.doi.org/10.1145/2594368.2594383>, 2014. DOI: 10.1145/2594368.2594383.
- [17] A. A. Bahashwan, M. Anbar, N. Abdullah, T. Al-Hadhrami, and S. M. Hanshi, *Review on Common IoT Communication Technologies for Both Long-Range Network (LPWAN) and Short-Range Network*, <https://www.researchgate.net/publication/344933088>, 2021. DOI: 10.1007/978-981-15-6048-4_30.
- [18] 4g.co.uk, *How fast are 4G and 5G? - Speeds and UK network performance*, <https://www.4g.co.uk/how-fast-is-4g/>. (visited on 10/28/2021).
- [19] L. Labs, *SigFox Vs. LoRa: A Comparison Between Technologies and Business Models*, <https://www.link-labs.com/blog/sigfox-vs-lora>, 2018.
- [20] Sigfox, *Sigfox - The Global Communications Service Provider for the Internet of Things (IoT)*, <https://www.sigfox.com/en>. (visited on 10/28/2021).
- [21] B. Foubert and N. Mitton, *Long-Range Wireless Radio Technologies: A Survey*, <https://www.mdpi.com/1999-5903/12/1/13/htm>, 2020. DOI: 10.3390/FI12010013. (visited on 10/28/2021).
- [22] L. Labs, *What Is LoRa? A Technical Breakdown*, <https://www.link-labs.com/blog/what-is-lora>, 2018.
- [23] —, *What Is LoRaWAN? [Technical Breakdown]*, <https://www.link-labs.com/blog/what-is-lorawan>, 2018.
- [24] A. Pohl, *Muon Detector Prototype*, 2021.
- [25] STMicroelectronics, “STM32F051x4 STM32F051x6 STM32F051x8,” 2017.

- [26] —, “STM32WL55xx STM32WL54xx,”
- [27] Z. Peterson, *Everything You Need to Know About Micro Via PCB Design Techniques — Altium Designer*, <https://resources.altium.com/p/everything-you-need-know-about-microvias-printed-circuit-design>, 2017.
- [28] P. Dillmann, *Via in Pad PCB Design - MacroFab*, <https://macrofab.com/blog/via-in-pad-pcb-design/>, 2019.
- [29] ZHONGKEWEI, “ATGM336H-5N Datasheet,” 2016.
- [30] STMicroelectronics, “MEMS digital output motion sensor: ultra-low-power high-performance 3-axis ”nano” accelerometer,” 2016.
- [31] Sensirion, “Datasheet SHT3x-DIS,” 2019.
- [32] Semtech, *SX1276/77/78/79*, https://datasheet.lcsc.com/lcsc/2010281834_SEMTECH-SX1276IMLTRT_C80171.pdf, 2016.
- [33] HGSEMI, “PCF8575,” 1999.
- [34] —, “PCF8574,” 2018.
- [35] Components101, *Design Guide - PMOS MOSFET for Reverse Voltage Polarity Protection Circuit*, <https://components101.com/articles/design-guide-pmos-mosfet-for-reverse-voltage-polarity-protection>, 2020.
- [36] Semtech, *SX1276 — 137MHz to 1020MHz Long Range Low Power Transceiver — Semtech*, <https://www.semtech.com/products/wireless-rf/loro-core/sx1276>. (visited on 11/05/2021).
- [37] STMicroelectronics, “AN2606 - STM32 microcontroller system memory boot mode,” 2021.
- [38] Suntsu, *Crystal Load Capacitance - Suntsu Electronics*, <https://suntsu.com/services/engineering-services/suntsu-application-notes/crystal-load-capacitance/>. (visited on 11/05/2021).
- [39] STMicroelectronics, “AN5407 - Optimized RF board layout for STM32WL Series,” 2020.
- [40] R. Golombick, *4 less obvious PCB DFM violations — PCBflow*, <https://blogs.sw.siemens.com/pcbflow/2020/03/16/4-less-obvious-pcb-dfm-violations/>, 2020.
- [41] M. Integrated, “MAX11634–MAX11637,” 2011.
- [42] TRAMsoft, *NMEA data*, <https://www.tramsoft.ch/downloads/garmin/NMEAda.ta.htm>. (visited on 11/07/2021).

Appendix A

Additional files and schematics

A.1 GitHub link

https://github.com/Paolo-dono/Muon_Detector_Prototype

A.2 Full PCB layout

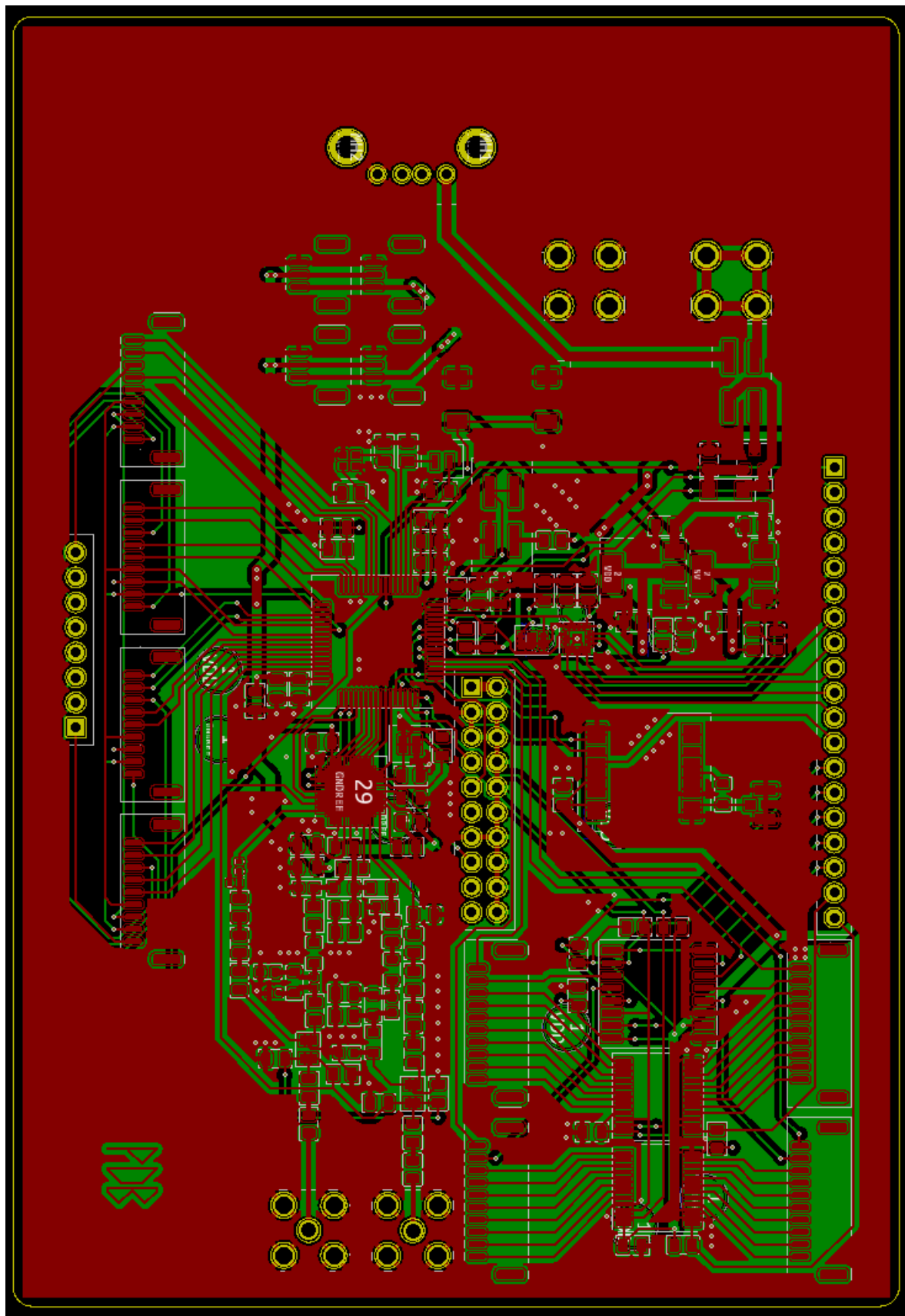


Figure A.1: Front side of the PCB

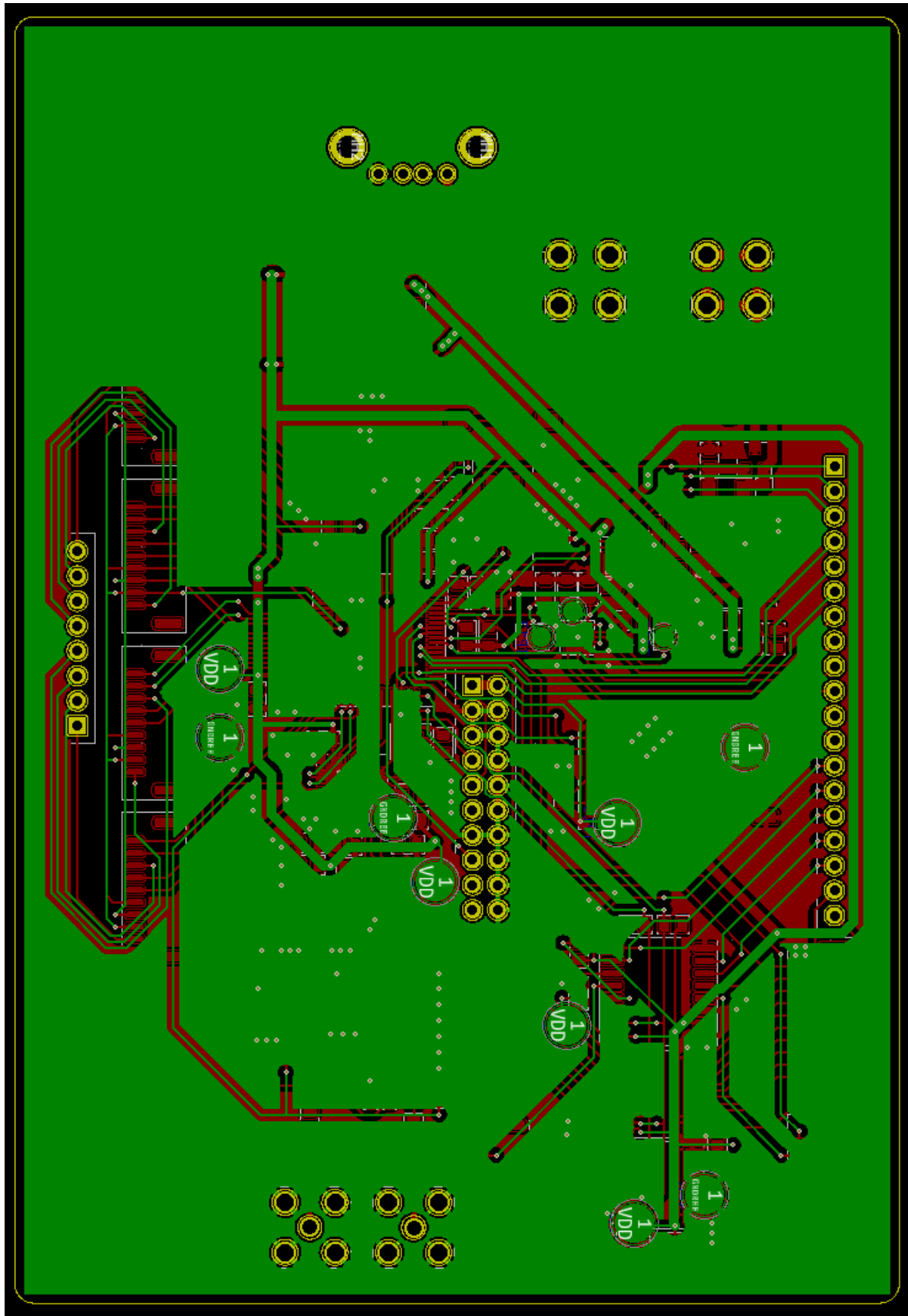


Figure A.2: Back side of the PCB

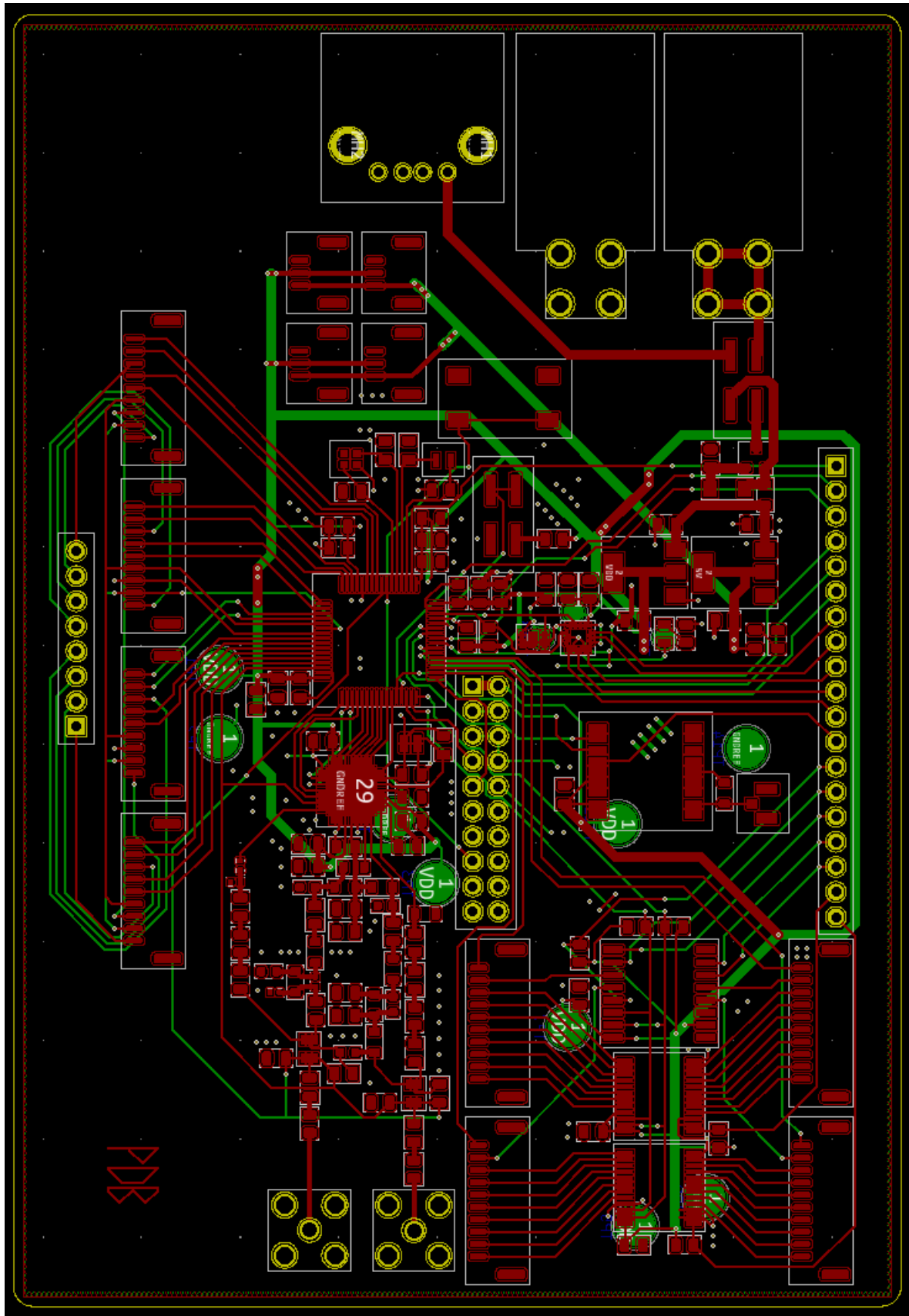


Figure A.3: Front side of the PCB without copper fill

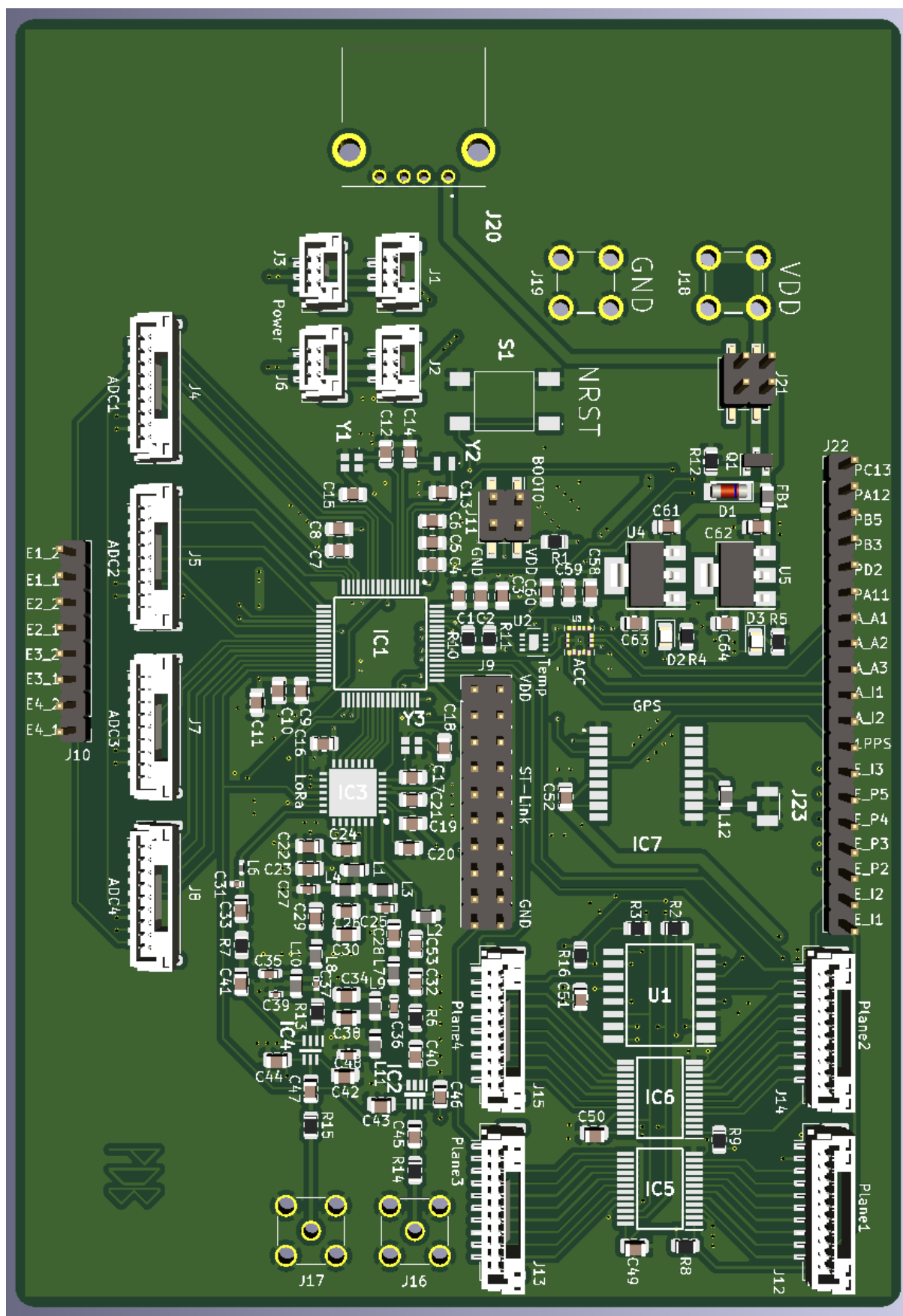


Figure A.4: 3D view of the PCB

Appendix B

Addenda

B.1 Ethics form

ETHICS APPLICATION FORM



Please Note:

Any person planning to undertake research in the Faculty of Engineering and the Built Environment (EBE) at the University of Cape Town is required to complete this form **before** collecting or analysing data. The objective of submitting this application *prior* to embarking on research is to ensure that the highest ethical standards in research, conducted under the auspices of the EBE Faculty, are met. Please ensure that you have read, and understood the **EBE Ethics in Research Handbook** (available from the UCT EBE, Research Ethics website) prior to completing this application form: <http://www.ebe.uct.ac.za/ebe/research/ethics1>

APPLICANT'S DETAILS		
Name of principal researcher, student or external applicant		Paulus Jacobus de Bruyn
Department		Electrical Engineering
Preferred email address of applicant:		mailforpaolo@yahoo.com
If Student	Your Degree: e.g., MSc, PhD, etc.	BSc(Eng)
	Credit Value of Research: e.g., 60/120/180/360 etc.	40
	Name of Supervisor (if supervised):	Jane Wyngaard
If this is a research contract, indicate the source of funding/sponsorship		
Project Title		Muon Detector Prototype

I hereby undertake to carry out my research in such a way that:

- there is no apparent legal objection to the nature or the method of research; and
- the research will not compromise staff or students or the other responsibilities of the University;
- the stated objective will be achieved, and the findings will have a high degree of validity;
- limitations and alternative interpretations will be considered;
- the findings could be subject to peer review and publicly available; and
- I will comply with the conventions of copyright and avoid any practice that would constitute plagiarism.

APPLICATION BY	Full name	Signature	Date
Principal Researcher/ Student/External applicant	Paulus Jacobus de Bruyn		14/08/2021
SUPPORTED BY	Full name	Signature	Date
Supervisor (where applicable)	Jane Wyngaard		17/08/21

APPROVED BY	Full name	Signature	Date
HOD (or delegated nominee) Final authority for all applicants who have answered NO to all questions in Section 1; and for all Undergraduate research (Including Honours).			
Chair: Faculty EIR Committee For applicants other than undergraduate students who have answered YES to any of the questions in Section 1.			