DEGREE PROJECT IN ELECTRONCS AND COMPUTER ENGINEERING, FIRST LEVEL
STOCKHOLM, SWEDEN 2016

Design of microcontroller circuit and measurement software for SiC and MOREBAC experiment

KTH Student Satellite MIST

HANNES PAULSSON
MIKAEL ANDRÉ

KTH ROYAL INSTITUTE OF TECHNOLOGY
INFORMATION AND COMMUNICATION TECHNOLOGY

Design of microcontroller circuit and measurement software for SiC and MOREBAC experiment

KTH Student Satellite MIST

Hannes Paulsson, Mikael André

2016-06-02

Bachelor's Thesis

Examiner
Carl-Mikael Zetterling

Academic adviser
Bengt Molin

KTH Royal Institute of Technology
School of Information and Communication Technology (ICT)
Department of Communication Systems
SE-100 44 Stockholm, Sweden

Abstract

This paper describes the development of an experiment to test the characteristics and functionality of Silicon Carbide (SiC) components in a space environment. The experiment is a part the "Miniature Student Satellite" (MIST) project, and the "Work on Venus" project, both situated at KTH, Stockholm, Sweden

The paper primarily covers the development and implementation of the experiments microcontroller and its software, whilst the construction and development of the test circuit for the transistors is carried out at the same time by another team, and therefore described in a separate paper.

A microcontroller is selected for this experiment after consideration is taken to both the Low Earth Orbit environment where the experiment will take place, end the power consumption restrictions due to the limited amount of power available at the satellite itself. The software on the microcontroller is then developed to read temperature and voltage input from the different transistors under test, and transform the input data to a readable format sent to the satellites On Board Computer, who can then communicate the readings to the Earth Base Station.

Apart from the software of the SiC experiment, a similar software solution on a similar microcontroller is also developed for another experiment called MOREBAC, which will be placed on the same satellite. The main difference on the MOREBAC project will be the type of data read on the input, the number of inputs and the format of the package sent to the On Board Computer.

The final stage of the work for this thesis is the design and construction of a Printed Circuit Board. The board contains the microcontroller and connected components, the transistors to be tested, as well as power supplying components, developed in yet another thesis work.

Keywords

Silicon Carbide, satellite, embedded systems, microcontroller, spaceflight

Acknowledgments

We would like to thank our Academic Advisor Bengt Molin and our Examiner Carl-Mikael Zetterling for their help during the development of this project.

We would also like to thank our teammates Simon Johansson, Matthias Ericsson and Johan Silverudd for their participation and help.

Lastly, we would like to thank the whole MIST team, especially Dr. Sven Grahn and Dr. Christer Fuglesang for making this project possible.

Stockholm, June 2016
Hannes Paulsson
Mikael André

Table of contents

List of Figures

List of Tables

List of acronyms and abbreviations

ADC	Analog to Digital Converter
BJT	Bipolar Junction Transistor
DAC	Digital to Analog Converter
DEC	Decimal value
GPIO	General Purpose Input and Output
HAL	Hardware Abstraction Layer
HEX	Hexadecimal value
I/O	Input and/or Output
I2C	Inter Integrated Circuit. A serial communication protocol.
ICT	Information and Communication Technology
MIST	Miniature Student Satellite
OBC	On Board Computer
PCB	Printed Circuit Board
RTOS	Real Time Operating System
SiC	Silicon Carbide
SWD	Single Wire Debug
USART	Universal Synchronus/Asynchronus Receive Transmit. A serial communication protocol.

1 Introduction

This chapter describes the background of the thesis and provides an overview of the project outlines, as well as the goals and purposes of the project.

1.1 Background

To land a spacecraft on Venus and have it work there, the spacecraft have to be constructed of components capable of withstanding extreme conditions. The surface of Venus has a temperature of 460°C, and the surface pressure is 92 bar[1], and while traveling from Earth to Venus the spacecraft will absorb large amounts of radiation. One material tested to withstand this extreme environment is Silicone Carbide (SiC)[2]. SiC components with similar electrical properties as regular Silicone components, can be manufactured out of SiC, with the advantage of environmental sturdiness. This material has successfully been tested in different environments on earth to simulate how they would perform. A new test is now to be conducted, to test the properties of these components in an actual space environment. This experiment is called SiC in Space, and will be launched as a component on a miniature satellite, MIST[3], developed by students at KTH. The experiment will then measure the performance of a SiC transistor and a regular Silicone transistor, while in orbit around the earth, and relay the results back to earth for analysis.

This thesis will cover the work of developing and designing parts of this experiment, focusing on the microprocessor that runs the test program for the circuit. The time frame of the project is 10 weeks, and the final result is not expected to be the finished product to be put on the satellite, but a testable prototype, leaving work for other students to complete the project before the launch of the satellite.

Another MIST-experiment called MOREBAC, developed parallel to SiC in Space, will need a microprocessor that works similar to the one developed in this thesis. The student responsible have asked for help to choose and configure a microprocessor for the MOREBAC project, and therefore, that part of their project is also a part of this thesis.

1.2 Problem definition

To measure a circuit and send the data from the satellite down to earth, some kind of microprocessor must be available on the experiment. To decide what microprocessor to use, and how to best configure and program it, a lot of different variables must be taken into consideration. The problem can be summarized into three different questions.

1. What microprocessor can be used, that can make all the required measurements, withstand the environmental forces present in orbit around the Earth, and still consume as little power as possible, since the power supply on a satellite is very limited?
2. How do you place the microprocessor on a Printed Circuit Board (PCB), with all necessary components to do the measurements required by the experiment, on a minimal surface area? Since the whole experiment is limited to 94x94mm PCB, and the actual test circuit together with components for power management also have to be fitted, the circuit of the microcontroller has to be as compact as possible, while still meeting EMC requirements, to

not emit or radiate unnecessary electromagnetic fields that could interfere with, or damage, other experiments fitted on the satellite.

3. How do we design a software that is capable of doing all the measurements accurate, but with minimal power consumption? The software must also be stable, handle the fact that the power to the microprocessor will be switched off between measurements, and be able to communicate with the On Board Computer of the satellite, in order to transmit the data that will then be relayed down to earth.

1.3 Purpose

The purpose of this thesis is to evaluate and choose a microcontroller suitable for the task of doing experimental measurements in space and to write a software that is optimized for the same task. The purpose of the microcontroller itself can be divided into three main tasks.

1. Provide the test circuit with different reliable and stable voltage levels during the test phase, to allow the test to be conducted under different circumstances.
2. To read the analog signals from the test circuit and store them as readable values.
3. Send the stored values to the On Board Computer (OBC) of the satellite, which will then transfer the stored data down to earth for further analysis.

The choice of the microcontroller should be based on requirements regarding the space environment, the required interfaces for the experiment, and the power consumption constraints set by the satellite.

The purpose of the developed software is effectively and accurately execute the tasks stated above, while being both stable and reliable during runtime. Besides that, the code should be used to configure the microcontroller in a way that it meets the requirement regarding power and communication.

As a secondary purpose, this thesis is to be used as a reference for any future work on the Sic in Space project.

1.3.1 MOREBAC

The purpose of the involvement in the MOREBAC project is to aid them with the microcontroller and software of their project since the student working on MOREBAC is mainly focused on biological science rather than computer science. Our part of the work on MOREBAC will be to replace their Arduino-based [4] test circuit with a circuit based on a microcontroller that meets the same requirements as the one used on the SiC in Space project.

1.4 Goals

The goal of this project is to design and implement two microprocessor units capable of measuring the results of the two tests SiC in Space and MOREBAC in accordance with the requirements specification (Appendix A). Due to the time frame for this project, the complete requirements specification will not be fulfilled. Mainly two aspects will not be covered; this project will not produce the final PCB that will be placed on the satellite for either one of the experiments, and the software written for the MOREBAC experiment will not be a final version, since the final experiment parameters won't be decided while this thesis work is being conducted.

The main goal has been divided into the following sub-goals:

1. Create software for the SiC experiments, as final and launch ready as possible under the circumstances.
2. Create software for the MOREBAC experiment that can be used as a base for the final version.
3. Design a circuit design and PCB as close to a finished version as possible for the SiC in Space experiment.
4. Design a circuit design and PCB that can work as a test circuit for the MOREBAC experiment and also be used as a base for the final circuit.

1.4.1 MOREBAC

The goal for our involvement in the MOREBAC project is to finish the development and construction of an experimental board based on a microcontroller equal to that of the SiC in Space project. Besides that, the software on the current Arduino based test circuit will be ported to the new hardware with maintained functionality.

1.5 Research Methodology

This thesis is based on two primary methodologies. First, a qualitative literature study, to provide the knowledge on how to best solve the task, based primarily on earlier work in the field, and data available about the hardware and software used. Second, an empirical phase, using the knowledge acquired in the literature study. In this phase hardware and software are developed in an iterative way, by a combination and research and testing to solve one problem at the time. The reason for working iteratively is to make sure errors and other issues are discovered early and to make the impact of them as small as possible.

1.6 Delimitations

This thesis will not focus on the design of the test circuit, what effects or functions are being monitored or the calculations involved. The scope of this thesis is the microcontroller used on the Sic in Space experiment, its software and the layout and construction of a prototype PCB, as well as the microcontroller used on the MOREBAC project, and its initial software and configuration.

The main focus of the report will be the development of the software and hardware of the first prototype of the SiC in Space experiment. This will not be the final product launched with the MIST satellite, so this thesis can be seen as a reference for future work on the project, together with the work on the test circuit by Ericsson and Silverudd[5], and the work on the power supply by Johansson[6], conducted parallel to this thesis.

The work related only to the MOREBAC project will have separate subsections under each chapter.

1.7 Structure of the thesis

Chapter 2 presents relevant background information about the SiC in Space project, MIST, and other related work.

Chapter 3 presents the methodology and methods used in this thesis.

Chapter 4 presents the actual work done on both hardware and software development and testing.

Chapter 5 presents an analysis of the work and its results.

Chapter 6 presents the conclusion of the thesis and recommendations of future work.

2 Background

This chapter provides basic background information about the SiC in Space project, MIST and the environment in space related to the requirements set on the hardware. Additionally, this chapter describes the MOREBAC project at a glance.

2.1 MIST

The project covered in this thesis is a part of a larger project at KTH called Miniature Student Satellite (MIST). This section will give you a brief introduction to the project to better understand the context of this thesis.

2.1.1 What is MIST?

The MIST project is initiated by the KTH Space Center[7]. The project itself is driven by student teams of about 10 people, working on it as different thesis projects lasting one semester, and then handing it over to a new team of students to continue the work in the following semester until the satellite is ready for launch. Apart from the students, Dr. Sven Grahn, and Dr. Christer Fuglesang are supervising the project, and mentoring the student teams.

2.1.2 The satellite

The satellite itself is based on a model called the CubeSat[8], a commonly used structure for experimental satellites. Since the first CubeSat was launched in 2003, more and more CubeSats have been successfully launched, and their popularity is still growing. In 2014, over 75 CubeSats were launched[9].

In the case of MIST, this satellite will host a number of experiments. Besides these, the satellite will be equipped with small thrusters, which is actually itself one of the experiments. It will also have solar panels that charge an internal battery on the satellite. This battery can deliver 14 V, and power all experiments on the satellite, along with the OBC (On Board Computer). The OBC also controls the power supply system, which takes the 14 V and steps them down to four 5V and two 3.3V buses that are then distributed to the different experiments to be used for low voltage components, primarily microcontrollers.

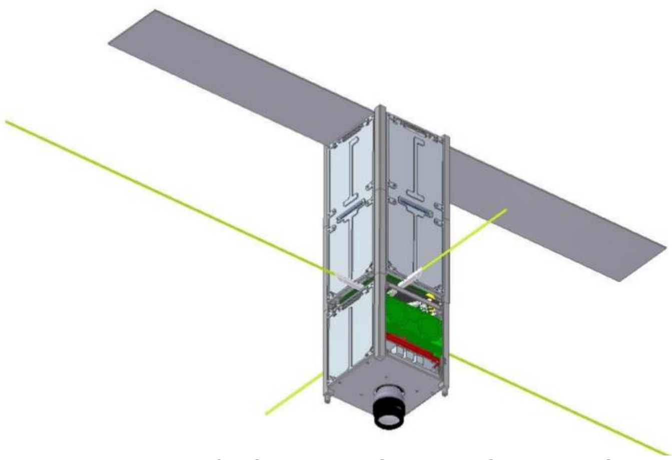


Figure 1 - 3D rendering of MIST satellite. Source: MIST Project. Source: MIST Project document M110-006

2.1.3 Purpose of MIST

There are two main purposes of the MIST project: education and experimentation.

The educational part of the project is, of course, the main reason why the project is conducted by students. Because of the complexity of building a functional satellite, the work is suited for thesis work in many different areas such as physics, electronics, computer science and mechanical engineering.

The other purpose of the project is to fit 8 different experiments as payload on the satellite. Most of these projects are also made by students as master or bachelor theses and conducted in close cooperation with the student teams who build the actual satellite. The project handled in this thesis, SiC in Space, is one of the eight experiments to be launched with MIST. Two of the other projects mentioned in this report, MOREBAC and Piezo LEGS, are also among the experiments to be launched.

2.2 SiC in Space

The main goal of this experiment is to test the performance of Silicon Carbide transistors in the harsh environment of space.

Silicon Carbide, SiC, occurs in several different crystal configurations, called polytypes, in nature. Most interesting for commercial use are configurations called 4H and 6H SiC[10]. Other common materials in semiconductors are among others pure silicon (Si) and Gallium Arsenide (GaAs).

The main reason SiC is suitable as a material in semiconductors is its high energy band gap, the energy needed for its electrons to jump from the valence band to the conduction band. The energy needed differs for different crystal combinations. For 4H, it is 3.2 eV and 3.0 eV for 6H. This can be compared to the energy bandgap for Silicon, 1.12 eV, and Gallium Arsenide, at 1.43 eV. The high energy bandgap is the reason why SiC can withstand higher temperatures than other semiconductor materials. When a material is heated up, the vibrations can excite the electrons enough to jump from the valence band to the conduction band. This, in turn, makes the semiconductor fail. With a high energy bandgap, more heat is needed for this to happen. The high energy bandgap also allows semiconductors made of SiC to be smaller for the same breakdown voltage. The on-resistance is also much smaller compared to other materials.

2.3 Environment in Low Earth Orbit

In Low Earth Orbit (LEO), the main aspect of the environment that has to be accounted for in this experiment is the temperature fluctuations. This is important when choosing what components to use when constructing the experiment. A thorough analysis of the thermal aspects of the MIST satellite has been done by Andreas Berggren[11].

The temperature of the spacecraft is decided by a number of factors. The external factors are mainly three, the radiated energy from direct sunlight, the sunlight that is reflected by the Earth, and the albedo of the earth, i.e. the infrared light radiated from Earth[12]. Heat conducted from the components on the spacecraft itself also radiates to all parts of the spacecraft. The satellite itself will also radiate some of this heat into space. The amount of radiation is based on the temperature of the spacecraft itself. This is controlled by a thermal control system. Taking all the above into consideration, the SiC experiment is expected to see temperatures ranging from -2 to +30 degrees Celsius.

Another important aspect of the environment in LEO is the radiation. There are three types of radiation to consider; Trapped Radiation, Galactic Cosmic Rays, and Solar Cosmic Rays. [13]

Trapped radiation consists of trapped energetic charged particles held in place by the Van Allen Belt, the magnetic field around the earth. These particles are electrons, protons, and heavy ions. For LEO, the SAA (South Atlantic Anomaly) is of extra importance. The SAA is a portion of the Van Allen Belt where the magnetic field reaches much closer to the earth, situated around Brazil, thus trapping more high energy particles.

The Galactic Cosmic Rays are radiation emitted from outside the solar system, consisting mainly of protons.

The Solar Cosmic Rays are generated on the sun from bursts caused by events like solar flares. These rays, like the Galactic Cosmic Rays, consist mostly of protons.

The geomagnetic field of the Earth shields us from most of the Galactic Cosmic Rays and the Solar Cosmic Rays in LEO. Though the total amount of radiation from all three of these sources are necessary to evaluate.

The main issue with the radiation is the so-called Single Event Upsets. The small size of the circuitry in a microprocessor means that the necessary charge needed to differentiate between a '1' and a '0' is small enough that a single charged particle can flip a bit from a '1' to a '0', or from a '0' t a '1'. [14]

Depending on where this occurs, the implication of such an event differs. If a bit is flipped somewhere in RAM during runtime, the program could stop working during this run, but would not impact the next execution of the program. If a bit is flipped in ROM, where the software itself is stored, the whole program could be irreversibly lost.

2.4 MOREBAC

The experiment Microfluidic Orbital Resuscitation of Bacteria (MOREBAC) aims to investigate the revival of freeze-dried microorganisms in orbit and is proposed by the Division of Proteomics and Nanobiotechnology, KTH.

The idea of the MOREBAC project is to revive freeze-dried microorganisms at different time intervals during the satellite's orbit, and then investigate the growth of the bacteria by measuring how much light from an LED that passes through the chamber where the micro-organisms are stored. The microcontroller used in this experiment will not only have to control the LED and read the light-sensors but also control the valves regulating the pressure in the bacterial chamber since it will change as they grow.

If the experiment is successful, this method could be vital for the development of miniature ecosystems in the case of a future human occupied planetary base.

2.5 Summary

The MIST project is a big project initiated by KTH to create a satellite entirely designed and developed by students. Placed on this satellite are several different experiments, of which SiC in space is one. The SiC in space project is a small part of the research on Silicon Carbide components and the advantages of using them in extreme environments. By launching the SiC in space experiment with MIST, tests can be conducted to analyze how the Silicon Carbide components behave in the environment present in a Low Earth Orbit. Another experiment on MIST is MOREBAC, a biological experiment testing the revival of freeze-dried bacteria in space. This thesis covers the part of that project that is also included in this project.

3 Methodology

3.1 Literature study

The first part of this project is to gather information regarding the two main subjects of the thesis. Low power microprocessors, and satellites in Low Earth Orbit (LEO). The task of finding a suitable microprocessor will depend on what requirements that are set on the microcontroller, so the first step is to set up a list of requirements. The requirements are set, taking into consideration what the microprocessor must be able to do during the experiment, the power consumption restraints set by the MIST satellite team, the available space on the final PCB, and the environment in the LEO. The microprocessor might need to withstand a wide temperature range, large doses of radiation and vacuum, and still perform deterministically.

After the requirements are set, a study of the microprocessor market is done. Different manufacturers are considered, and available datasheets and similar technical documents are being used to find out what processors are suitable for the task. Since many manufacturers offer similar products, all meeting the requirements set, other factors will be weighed in, such as availability, price, and development environment. Another factor the was proven important in the choice of microprocessor was the fact that some processors were available as development kits, making it possible to start developing basic software and try out the processor, without first having to design and assemble a PCB or similar custom development environments.

To find out about the working environment for the experiment, when mounted on the satellite and launched into orbit, many different sources are considered to find as accurate data as possible. Since the environment is dependent on the actual orbit, the main source will be the documents produced by the MIST teams. The data provided by these should be the most accurate for the final environment for the satellite. Also, the data from the thermal control team of the mist is important, since the thermal control system might affect the environment for the different experiments. A problem with this source is that the thermal control work is under development at the same time as this experiment, meaning that they might change during the work on this thesis resulting in new requirements being set. To avoid having to rely on only one source for this data, other sources are considered in this research, one of them being the data available from the developers of the CubeSat standard and ISIS[15], the company producing the actual frame for MIST. As a third major source, data from bigger satellite projects, mostly done by NASA or similar agencies, are being used.

3.2 Software development

The software will be developed in two main stages. First, a development kit will be used, for simplicity reasons, to develop and test the main features of the program. Then, if everything seems functional, a PCB will be designed similar to what the final experiment PCB will look like, and the software will be refitted and tested on this board. If everything has been done correctly in the first development phase, the transition to the final layout should be relatively seamless.

The software will be developed step by step, testing one major function at the time. Every part will be tested individually before being put together into a running program. This way of testing and developing one unit at the time will make debugging easier since each functionality will have fewer possible error sources, in case things are not working as intended. The main functions to be tested separately are:

- Initialize peripherals and oscillators.
- Sending I2C messages.

- Receiving I2C messages.
- Using DAC to provide the experiment with a reliable voltage source.
- Using ADC to measure several inputs from the experiment.
- Store the measured data temporarily and create a data package to be transferred via I2C.
- Setting the processor in sleep mode after the experiment is done, to save power.

After each unit is tested and approved, the main program will be constructed, and the units will be tied together. The software is developed using the free version of the software Keil µVision 5[16]. The main initialization code is created with the tool STM32CubeMX[17], available from the MCU manufacturer ST-Microelectronics.

3.3 Software testing

For each unit to be tested, a test case is set up to make sure all functionality is tested. After the test case is set up, the software is developed with the intention to pass the current test case. If the test case is clear, work starts on the next thing to be tested. A cleared test case, however, does not make the code final. The entire program is developed iteratively, since new requirements may come up during the development process, and some units may have to be rewritten. Also, a working unit may still be optimized to be faster, more energy efficient or just easier to understand, just to make a better final product.

Some of the functionality can be tested and developed on the microprocessor or development kit itself, but some of the tests require external components to simulate the final way the code will be used on. One of the main functions for this is the measurement since this is the main task for the microprocessor. In order to develop this part of the code, without having to rely on the team working on the actual experiment, simple test circuits will be designed on a breadboard, to simulate a possible input scenario for the real experiment. These will then be used for the purpose of testing the ADC and DAC unit, but also to evaluate power consumption for the whole program at a later stage. In the case of the real circuit for the experiment will be finished, this will then be used to repeat the tests in an even more accurate way.

3.4 Hardware Development

The initial software development will be conducted on a factory made development-kit. However, when the main part of the software is done, a transition has to be made to a hardware layout similar to what will be implemented in the final product.

The first part is to develop a simple circuit to test the microprocessor as it is, without the many peripherals implemented on the development–kit. This will result in a simple PCB with the microprocessor, a voltage regulator, input/output pins, some LEDs for debugging, and basic necessary components like decoupling capacitors and resistors. This board can then be used to test the different software units in order to test if something has to be changed from the previous versions. One of the main difference may be the change of some I/O pins input impedance caused by all the peripherals connected to the development board. Since all of the microprocessors available I/O:s will be available on the new test board, this could not only be used together with the test circuits developed for the unit tests, but also to connect the other parts of the final product, developed by the other teams, and test the whole unit. If these tests are successful, the next step would be to create a layout for the entire experiment, following the design requirements for the MIST satellite, see Appendix E. The schematics and the PCB layout is created using the free software DipTrace[18].

3.5 Experimental design/Planned Measurements

A test plan for the software is given in Appendix B.

3.5.1 Evaluation board

The STM32L0-Discovery[19] board is used to test the initialization of the peripherals and communication protocols. The test programs are based on the demo programs available from the manufacturer and edited to mimic the behavior that we would expect from the final product. The I2C communication is tested between two Discovery boards, the ADC is tested by feeding the different channels from a signal generator, or a basic voltage divider circuit and the DAC is tested by simply measuring the output with a voltage meter.

Figure 2 - STM32L0 Discovery

3.5.2 The microcontroller experimental PCB

This PCB features only the processor and the necessary components like decoupling capacitors and filters. All of the available I/O:s are routed to connectors, to allow the experimental card being connected to other components or breadboards. The initial tests on this board are the same as the ones made on the Discovery board. After that, the board is connected to a test circuit similar to the one to be fitted on the actual experiment, and test readings of this circuit are made. The results are also sent via a UART serial interface to a PC and displayed graphically in an application created in e software environment called Processing.

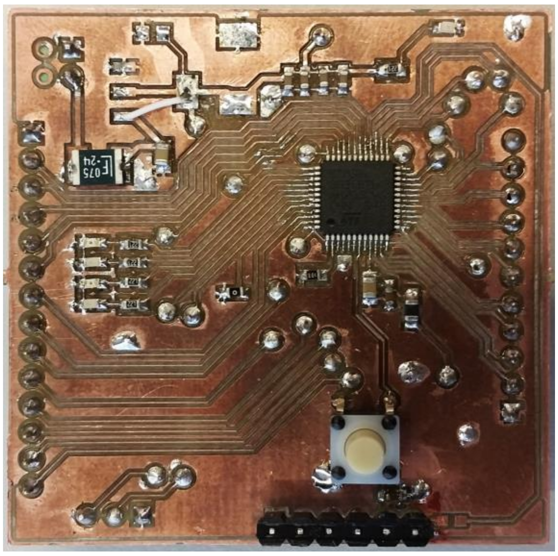


Figure 3 - Experimental PCB, Milled

3.5.3 The Prototype PCB

The prototype PCB is a prototype version of the final circuit to be used in the experiment, including the MCU, the test circuit and the power supply (BILD). The tests conducted here are basically the same as with the experimental PCB mentioned above. Apart from that, the power consumption on both the MCU power supply and the power supply providing the test circuit with 10V can now be measured by reading the voltage drop across a small resistor in series with the power supply input.

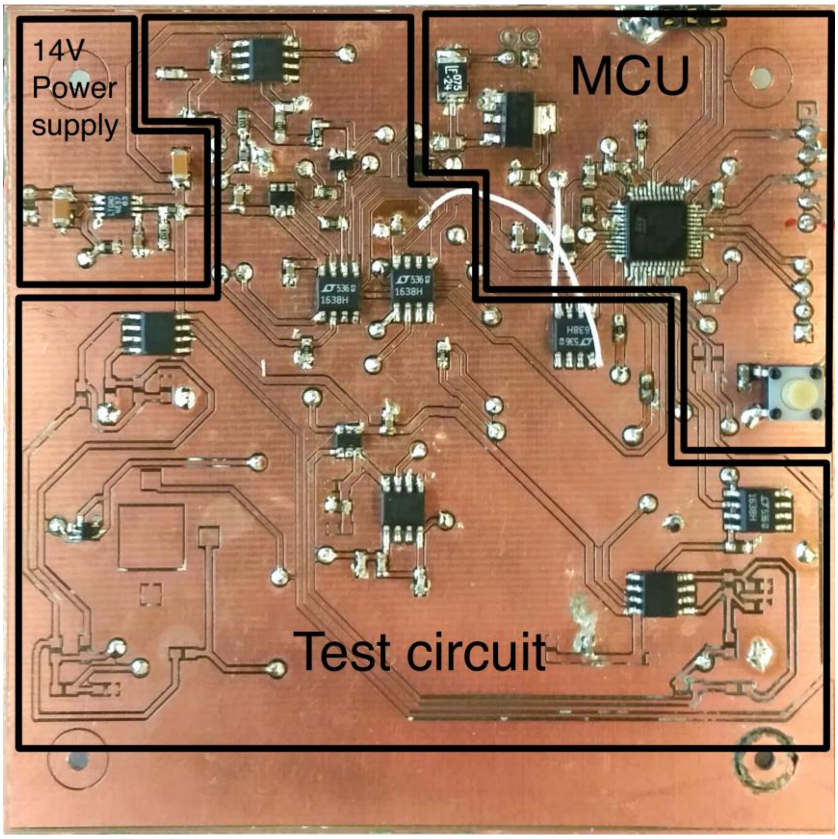


Figure 4 - Prototype PCB, Milled

4 System design and implementation.

4.1 Choice of MCU

The MCU market is big. Huge in fact. There are numerous of producers each with hundreds of different models for different purposes and solutions. The first step in choosing what MCU to use in the SiC and MOREBAC project is therefore to establish some requirements. There are three main areas to consider: the environment in which the MCU will operate, the power consumption, and the number available ADC to be able to test as many inputs as required.

4.1.1 Environment

The MCU will operate on a satellite in LOE. As mentioned earlier, the estimated temperature span for the experiment is -2 to +30 degrees Celsius. Most commercial MCU are specified for -40 to +85, or better, so no extra consideration should have to be taken to operating temperature requirements. According to the data available from the MIST -project, the background radiation in LOE does not exceed what standard commercial components can handle, so most MCUs would be suitable. Finally, the vacuum. The main effect on components in a vacuum is the lack of heat radiation. This matter should have been taken into consideration in the thermal control calculation, and therefore, should the above-mentioned temperature range of the experiment still be valid.

4.1.2 Power consumption

Since the only power source for the satellite is its solar panels, and a battery storing energy from those, each of the experiment should try to consume as little power as possible. Therefore, the power consumption of the MCU is highly prioritized, and should be kept to a minimum. Besides choosing an MCU with a naturally low power consumption due to its physical construction, the configuration of the processor will also be important. The MCU must have the possibility to run on a very low frequency, and also any unnecessary features an I/O:s must be able to be switched off. Most producers have “Ultra Low Power” series of MCU, and these should be considered.

4.1.3 Analog and digital interface and peripherals

The owner of the SiC project asked for an Analog to Digital Converter (ADC) with at least 10 channels, to make room for a few different experiments to be carried out. The ADCs should also have at least 10-bit resolution to provide sufficient precision. Besides the ADC input, the number of I/O should be kept to a minimum to keep the size of the MCU footprint small, due to the limited space on the PCB.

Besides the ADC, the test method chosen for the components requires a variable voltage source, to generate different base currents to the test circuit. This requires the MCU to have at least one Digital to Analog Converter (DAC).

After taking all the requirements above into consideration there are still a lot of MCUs to choose from, most of them offering similar specifications and features.

The final choice of MCU for this project was the STM32L053C6 from ST Microelectronics[23]. This decision was made due to two major factors. First, the writers have previous experience of working with MCUs from ST Microelectronics, and therefore, no unnecessary time have to be spent on getting to know new developing tools and API:s. Also, there is a development kit based on the same processor available, “STM32L0538 - Discovery”, allowing most of the code to be tested in an early phase of the project, without having to create a temporary PCB for testing purposes.

Other advantages of the chosen processor are low power consumption (88uA/MHz), a 7x7mm/48pin LQFP package, a 10 channel/12 bit ADC, one 12bit DAC and a minimum clock speed of 65kHz.

4.2 Software Design

The complete code for the experiment is available on GitHub. Links to these are provided in Appendix D

4.2.1 STM32CubeMX and the HAL-library

In the beginning of the software development, the previous mentioned software tool STM32CubeMX is used to generate a lot of the initialization code. The software allows the user to choose configuration parameters like what functions to use the different input and output pins for, what clock speed to use, or what interrupts to listen to. The generated code can then, of course, be modified during the software development if circumstances change or new features are needed. Most of the code generated by CubeMX, as well as a lot of the code used during the development phase of this project, is based on a Hardware Abstraction Layer (HAL) library provided by the microcontroller manufacturer ST-Microelectronics[20]. The HAL-library is used to simplify the hardware part of the code, that is, all the code written intended to use or manipulate the MCU or any peripherals.

4.2.2 Main Program

Since the task for the main program is very straight forward, no real-time operating system (RTOS) or watchdog is needed. The basic task for the program, as a function of time, is shown in Figure 5

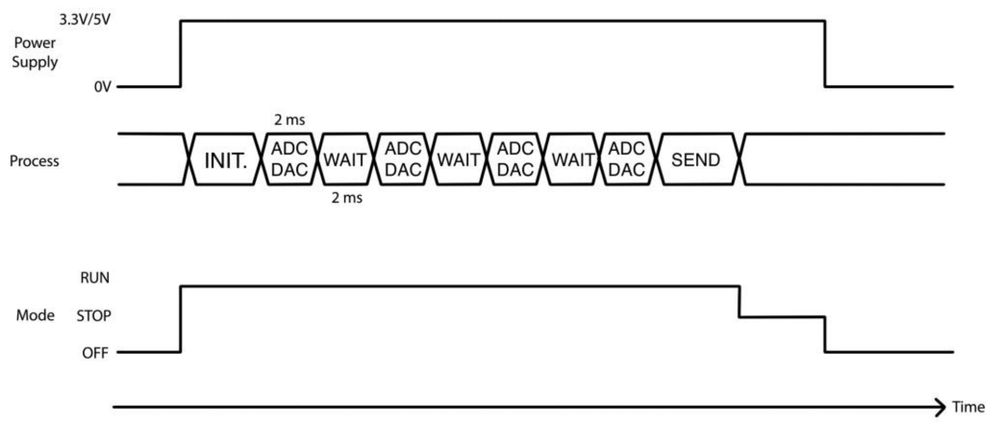


Figure 5 - Basic task of main program

When the supply power is turned on for the experiment, the software begins performing the test by setting the DAC to four different voltage levels, while reading the results from the test circuit with the ADCs. When all tests have been performed and handled, the program enters the low power STOP-mode, until the power supply of the experiment is turned off.

A more comprehensive flow chart of the program is shown in Figure 6.

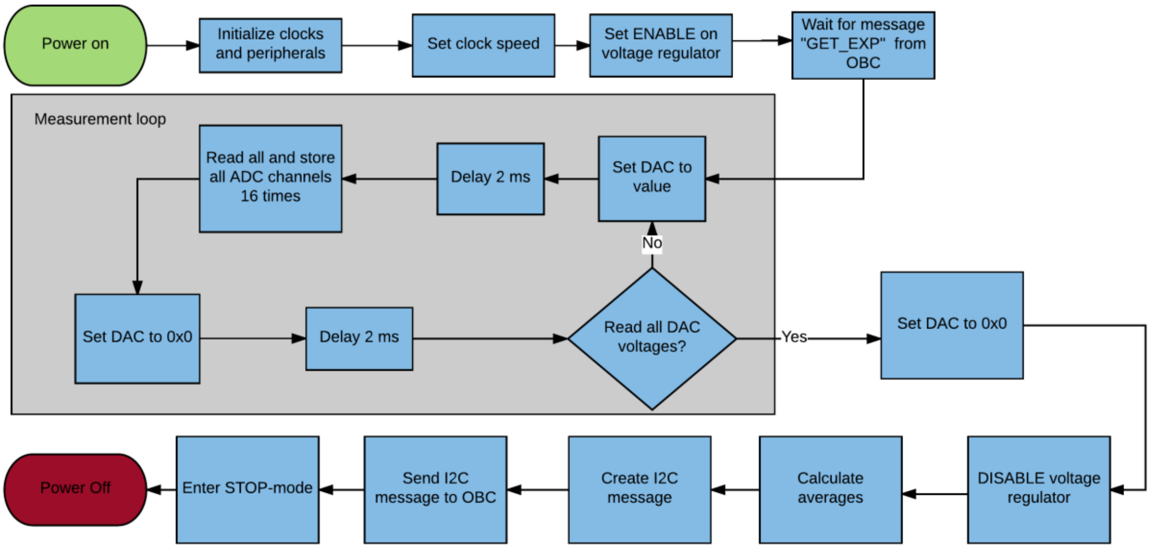


Figure 6 - Flowchart of main program

4.2.3 Initialization

When the power is turned on, the software initializes the peripherals (the ADCs and the DAC), and the clocks. The internal multispeed oscillator is used as System Clock, configured to run at 524 kHz.

When this is done, we wait for the message from the OBC, that they are expecting data from our experiment.

In order to minimize the power used on the battery bus, the voltage regulator that regulates the supply voltage for the test circuit itself is turned on by software just as the experiment starts, and turned off as soon as the measurements are done.

4.2.4 ADC Initialization

The ADC is configured in continuous conversion mode[21, p. 313]. In this mode, all ADC ports that should be read are initialized as the program starts up. When ADSTART is called, all ADC ports are read in sequence. After each read, the EOC (End of Conversion) flag is set in the ISR register, and the value is stored in the register DR. By reading the DR register after each EOC flag, all enabled ADCs can be read in a quick fashion. When all ADC channels are read, the EOSEQ (End of Sequence) flag is set.

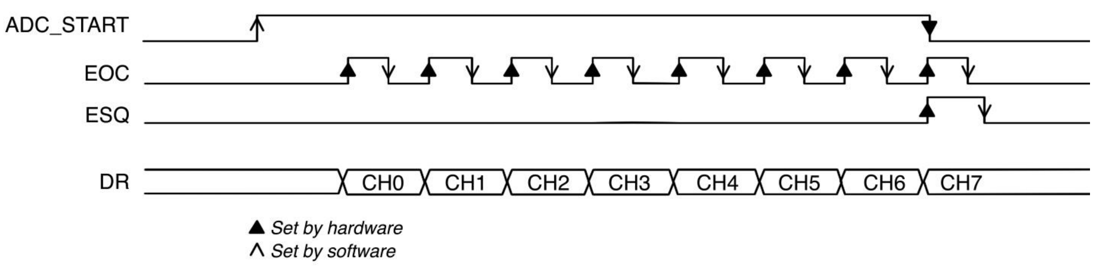


Figure 7 – ADC register flow

4.2.5 Measurement loop

When the experiment starts, the DAC is set to the first value, 3.1 Volts. Since the DAC has a 12-bit resolution, passing 0x000 to the DAC is equal to 0 V, and 0xFFF is equal to 3.3 Volts.

The voltage level of the DAC as a function of the provided argument *Value* can be calculated as shown in equation 1. From this follows that we can calculate the argument *Value* as a function of the voltage provided from the DAC as shown in equation 2.

$$V_{DAC}(Value) = \frac{Value}{4095} * V_{DD} \quad (1)$$

$$Value(V_{DAC}) = \frac{4095 * V_{DAC}}{V_{DD}} \quad (2)$$

With the supply voltage V_{DD} at 3.3 Volts, the four voltage levels are calculated and shown in Table 1.

Desired voltage (V)	DAC value (DEC)	DAC value (HEX)
3.1	3847	0xF07
2.1	2606	0xA2E
1.1	1365	0x555
0.5	620	0x26C

Table 1 - Calculated DAC values

The voltage levels chosen is based on suggestions from the experiment design team, and is subject to revision.

After each voltage is set on the DAC, all eight points of measurements are read with the ADC. The measurements are explained in Table 2 and Figure 8

ADC Channel	*Measurement*
0	Temperature sensor, Si BJT
1	V_{BE}, Si BJT
2	V_{RB}, Si BJT
3	V_{RC}, Si BJT
4	Temperature sensor, SiC BJT
5	V_{BE}, SiC BJT
6	V_{RB}, SiC BJT
7	V_{RC}, SiC BJT

Table 2 - Measurement points

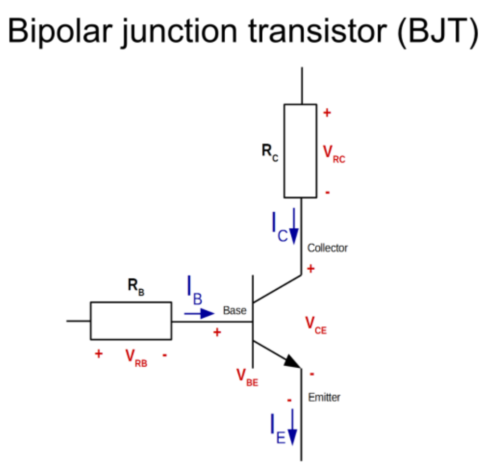


Figure 8 - Illustration of measurement points [5].

All points are measured 16 times in quick succession. These sum of the 16 measurements for each point is stored in a struct, consisting of one field per measurement.

When the measurements are done, the DAC is set to zero volts for a period of 2 ms. This is done due to a requirement from the experiment design team since the transistors run a risk of overheating if the power is turned on for too long. After this delay, the DAC is set to the next value, and the measurement loop runs again. The resulting values from each voltage level are stored in separate structs.

4.2.6 Calculating average

Important to note is that doing exactly 16 readings to compute an average have a few advantages. First of all, taking several measurements and computing the average reduces the risk of incorrect readings. By letting the number of readings be a power of two, in this case, 2^4, it is possible to shift the accumulated value four steps to get the average. The reason to choose 2^4, and not for example 2^5, makes it possible to store the data in a 16-bit unsigned integer. Since the ADC has a resolution of 12 bits, the highest value we can read, 3.3 V, is represented as $0xFFF_{16}$ (4095_{10}). When accumulating 16 of these values, the highest possible reading would be $0xFEE2_{16}$ (65250_{10}), which still fits within 16 bits, where the maximum value is $0xFFFF_{16}$ (65535_{10}). If we were to use 2^5 readings, the highest possible value would be $0x1FFE0_{16}$ (131040_{10}), which would need a 32-bit integer to be stored.

So, by shifting all the read values in the structs by 4 bits to the right, the average measurements are obtained.

4.2.7 Creation of I2C Data Package

The I2C message has to follow a set structure, decided by the OBC team. The first 4 bytes must contain the length of the message to be sent. Those 4 bytes should not be included in the length. The following bytes must be the payload. The last byte must be a checksum based on the payload. A simple modular checksum is used.

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	...	Byte n
Length	Length	Length	Length	Payload	Payload	Payload	Payload	Payload	Checksum

Figure 9 - I2C message example

Since our data is stored in 16 bits integers, all values will be represented by two bytes. The size of our payload is represented in four bytes as 0X00000041, as shown in equation 3.

$$Size\ (bytes) = ADC\ Readings * (DAC\ levels * 2) + Checksum = 8 * 4 * 2 + 1 = 67_{10} = 0X00000041_{16} \qquad (3)$$

All our measurements will have a value between 0x000 and 0xFFF, and will as such always fit within two bytes. For example, the value 0xFFF can be represented in two bytes as shown in Figure 10

Byte 0	Byte 1
0x0F	0xFF

Figure 10 - Payload example

All measured values are packaged in this fashion. At the same time, all values are added to a single 8-bit unsigned integer. As this integer overflows, it will represent the modular checksum of our message. This is then added to the end of the message. An example of how this complete package might look can be seen in Appendix C.

4.2.8 Sending message and entering STOP-mode

The last important step is to send the data to the OBC using the 12C protocol. Since the experiment is configured as an I2C slave, it cannot request to send data. Instead, the master (the OBC in this case), must request to receive data from the experiment. To handle this, the software sets itself in a state where it is polling for an I2C read request. After the data has been sent, the software puts the MCU in the so-called STOP-mode. In this mode, the processor draws very little current (about 1μA), and this is used to minimize the power consumption for the time between the I2C message is sent, and the OBC shut down the experiment by disconnecting the power. The STOP-mode is entered in two stages. First, all peripheral clocks are disabled, and the “Ultra Low Power” mode is enabled with a call to the I2C message example

```
HAL_PWREx_EnableUltraLowPower();
```

After that, stop mode is entered using:

```
HAL_PWR_EnterSTOPMode(PWR_LOWPOWERREGULATOR_ON, PWR_STOPENTRY_WFI)
```

4.3 Analyzing software

In order to validate the measured data, and test the format of the I2C package, a secondary software was developed. The software was developed with a development tool called Processing (www.processing.org).

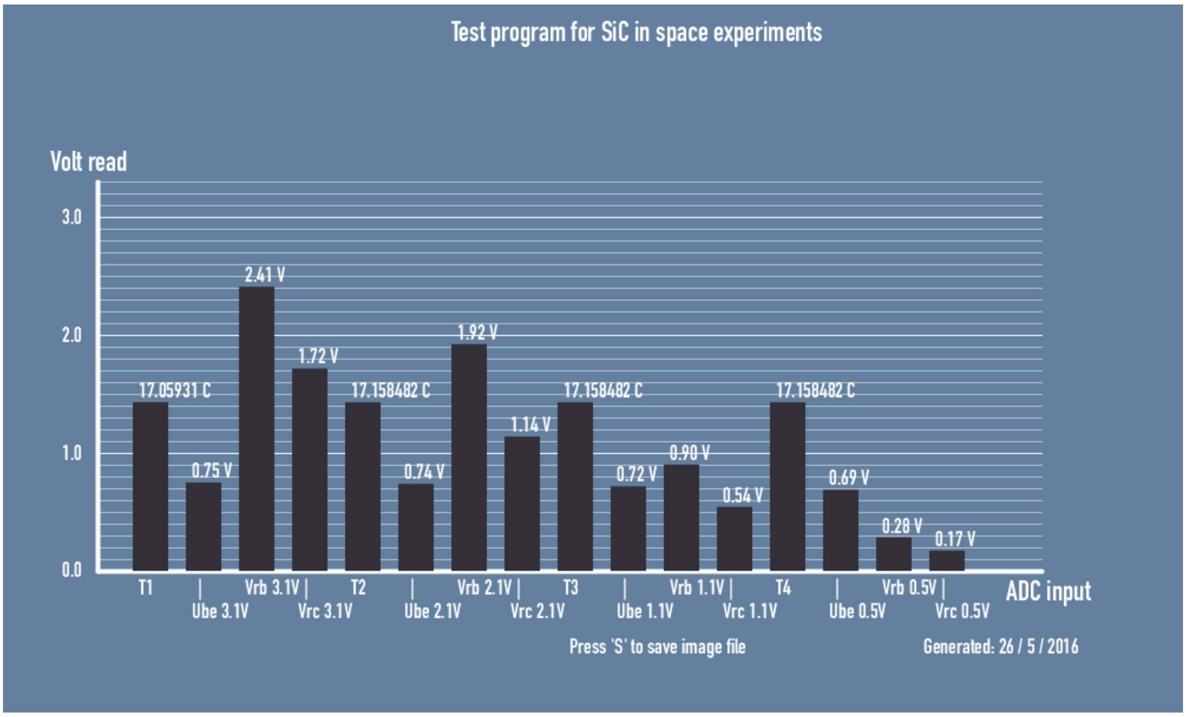


Figure 11 - Example of data read with analyzing software. Only showing data from one of the transistors (Si BJT).

The main purpose of this program was to show the measured values in a graphical and easy to read way. The program communicates directly with the experiment PCB via USART, where it receives a message identical to the one sent to the OBC via I2C. This USART connection is of course not implemented in the final version of the software and is therefore not explained in this report. This did, however, give us the opportunity to define a method to read the data in the message, and calculate the voltages read, along with the temperature.

4.3.1 Reading the data package

The basic structure of the data package is explained in chapter 4.1.6. In order to read the data, the bytes has to be stitched together. This is done by shifting the first byte eight bits to the right, and then add them together with the bitwise operator OR. Written as code, it would look like:

```
int value = ((byte1 << 8)||(byte2));
```

The bitwise OR operator can be described mathematically as shown in equation 3, where b is the number of bits, in this case, 16, and x and y are the two bytes.

$$x\ OR\ y = \sum_{n=0}^{b} 2^n \left[\left[\left(\left\lfloor \frac{x}{2^n} \right\rfloor mod2\right) + \left(\left\lfloor \frac{y}{2^n} \right\rfloor mod2\right) mod2\right] + \left(\left\lfloor \frac{x}{2^n} \right\rfloor mod2\right)\left(\left\lfloor \frac{y}{2^n} \right\rfloor mod2\right)\right] mod2\right] \quad (3)$$

This would then yield an integer value. For the voltage levels, this can be calculated as shown in equation 4.

$$Voltage = \frac{Value}{4095} * 3.3 \quad (4)$$

These voltages can be used further to calculate beta (β). This is explained in more detail by Eriksson and Silverudd[5].

The temperature based on the voltage read on the temperature sensors can either be read using the table in the data sheet for the thermometer [22], or be calculated with equation 5, which is a parabolic equation derived from the same table.

$$T = \frac{8.194 - \sqrt{(-8.194)^2 + 4 * 0.00262 * (1324 - V_{TEMP}(mV))}}{2 * -0.00262} + 30 \quad (5)$$

4.4 Hardware Design

4.4.1 Schematics and PCB – Evaluation board circuit

During development, a schematic and PCB was created for testing purposes only. This design was equipped with pin headers connected to all used inputs and outputs for the SWD connection, the ADCs, the DAC, the I2C and the USART connections. The schematics also included four LEDs for testing purposes, and an LED to indicate that the supply voltage is applied, and a reset button.

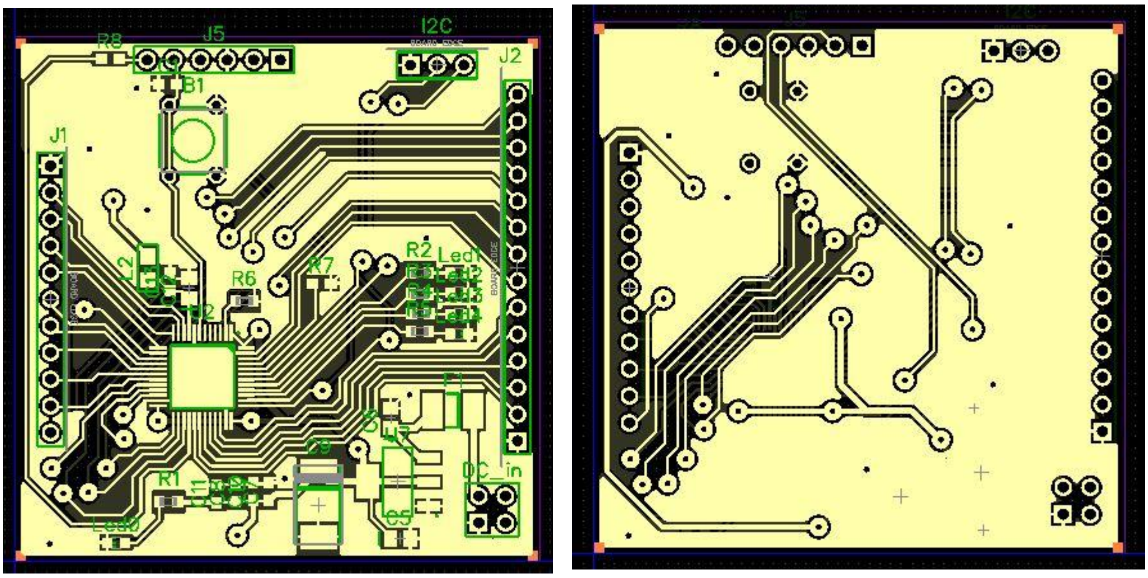


Figure 12 - Front and back of experimental PCB

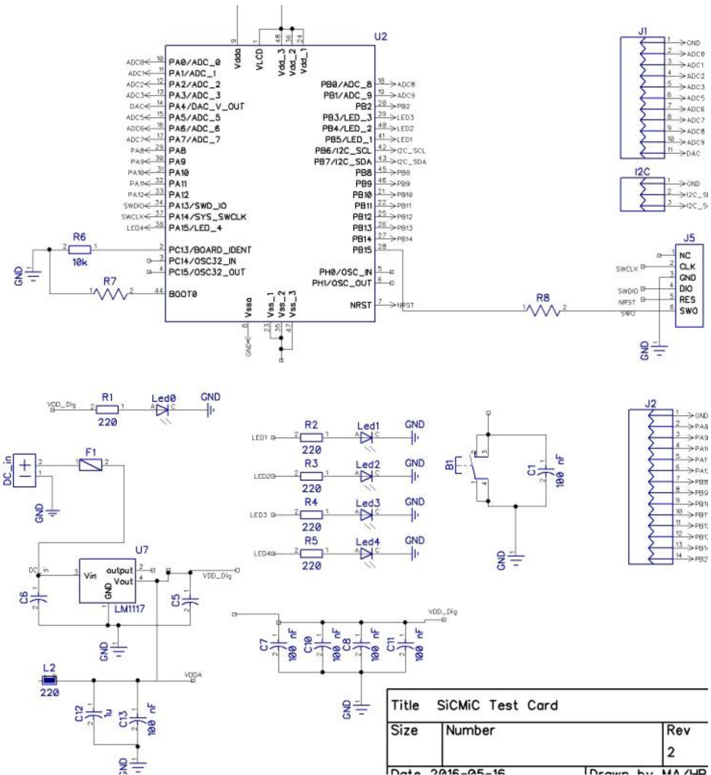


Figure 13 - Schematic of experiment circuit

In order to ensure correct functionality of the MCU, a few external components are needed. The choice of placement and values of these components are based on the design recommendations provided by the manufacturer of the microprocessor, St-Microelectronics. Decoupling capacitors are used between all VDD/VSS-pair, in order to lessen the interference on the power supply to the MCU [24, p. 25]. One 100nF ceramic capacitor is used for each VDD/VSS-pair, and one ceramic 10μF is used in parallel with those. Since the VDDA is supplied by the same source as the VDD signal, a filter consisting of one ferrite and two capacitors is used to filter out high-frequency noise. The capacitors used are one ceramic 100nF and one ceramic 1μF. A capacitor is also used to remove parasitic resets on the reset button. A 100nF ceramic capacitor is used[25, p. 92].

Since it was not yet decided whether or not our experiment would be supplied with 3.3 V or 5 V, the first test design for the MCU PCB includes a voltage regulator, to step the voltage down from 5 V to 3.3 V. The performance of the voltage regulator became an issue, since the quiescent current consumption of most OTS voltage regulators is substantially greater than the consumption of the MCU itself.

The first voltage regulator chosen was the popular LM1117, often used in similar situations, stepping down external power sources from up to 20V [25], down to the 3.3 V most MCUs uses. This proved to be problematic since the LM1117 has a quiescent current of as much as 5-10 mA. Since the MCU only draws about 1 mA during the measurement phase, this was not acceptable. For this reason, a different voltage regulator, LT1129IST, was used. This meet the important specifications and draws a quiescent current of about 50μA[25].The effect of this change is discussed further in chapter 5.1.2.

The voltage regulator needs some capacitors for optimal performance. The recommendation from Linear Technology is one capacitor between 1uF and 10μF between the input and ground, and one capacitor of at least 3.3 μF between the output and ground[25, p. 1]. For both of these, a 10 μF capacitor was used.

On the test circuit, a 750 mA fuse is placed on the input for protection.

4.4.2 Schematics and PCB - proposal for final design

Other than the test circuit, a proposal for a final design was developed. This design has a few key differences from the previous design. Most notably, this design contains the complete experiment. This includes the test circuit for the transistors and the voltage regulator on the battery supply for this experiment. Information about those part of the experiment is available in [6] and [5].

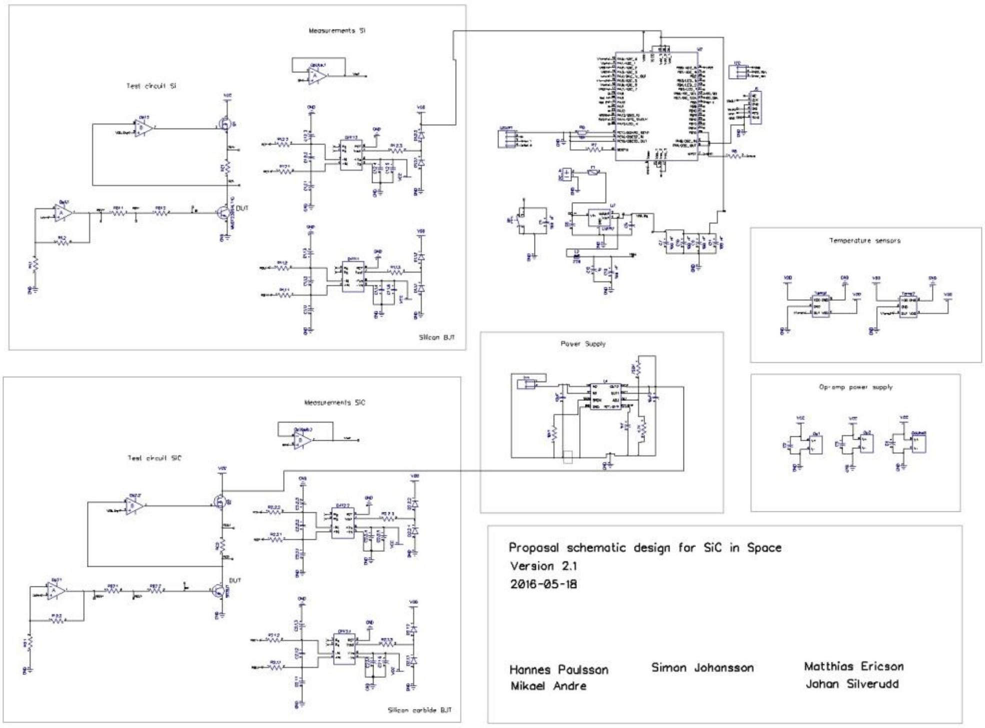


Figure 14 - Schematics proposal for final design

The differences in the MCU part of the design is that the voltage regulator is removed since the most recent decision from the MIST team is that we will be supplied with 3.3V directly. The fuse is also removed since a fuse will be positioned at main power distribution of the satellite.

Most parts that will not be useful in orbit are also removed. In this design, no LEDs are used, and all pin headers for the ADCs and the DAC are removed. The pin headers for the programming port, the I2C, the USART and the power supply are not removed. The I2C and power supply will be connected to the experiment via cables that could be soldered to the pin header connections. The USART connection will not be used on the final product, but is left on this design in order to simplify further testing and analysis. These pinouts could either be left unsoldered or be removed on a final design of the PCB.

All used ADCs and the DAC is connected via traces on the PCB. A separate trace for the shutdown signal to the voltage regulator on the battery input is also incorporated in the design.

The schematic also includes the reset button. This can be left unsoldered or removed for the final design.

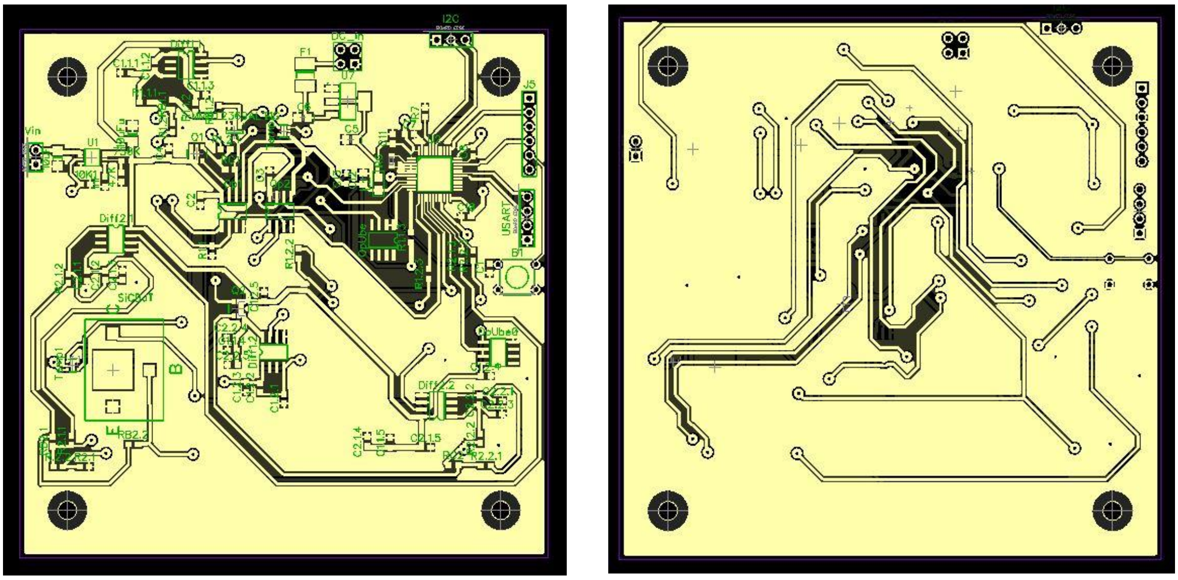


Figure 15 – Front and back of PCB design of proposal design

4.5 MOREBAC

4.5.1 Hardware

The hardware developed for the MOREBAC project is very similar to the first experimental PCB designed for the SiC in Space project. The purpose as said before, is to create a development board to replace the Arduino in the current test circuit used in the MOREBAC project. To do that a suitable MCU have to be picked out. Since the only real difference in requirements from those of the SiC project is the number of ADC needed, the obvious choice was a bigger version of the SiC processor, with 64 pins, of which 16 can be used as ADCs. This processor is called STM32l053R6[26]. To make the card useful even for further experiments, we placed connector pins for all 16 ADC channels, as well as plenty of GPIO pins, allowing for most of the final MOREBAC experiment being developed using this PCB. The design and layout can then be used as a reference to the final design by the next student to work on the project. The final design of the board and the schematic can be seen in Figure 16.

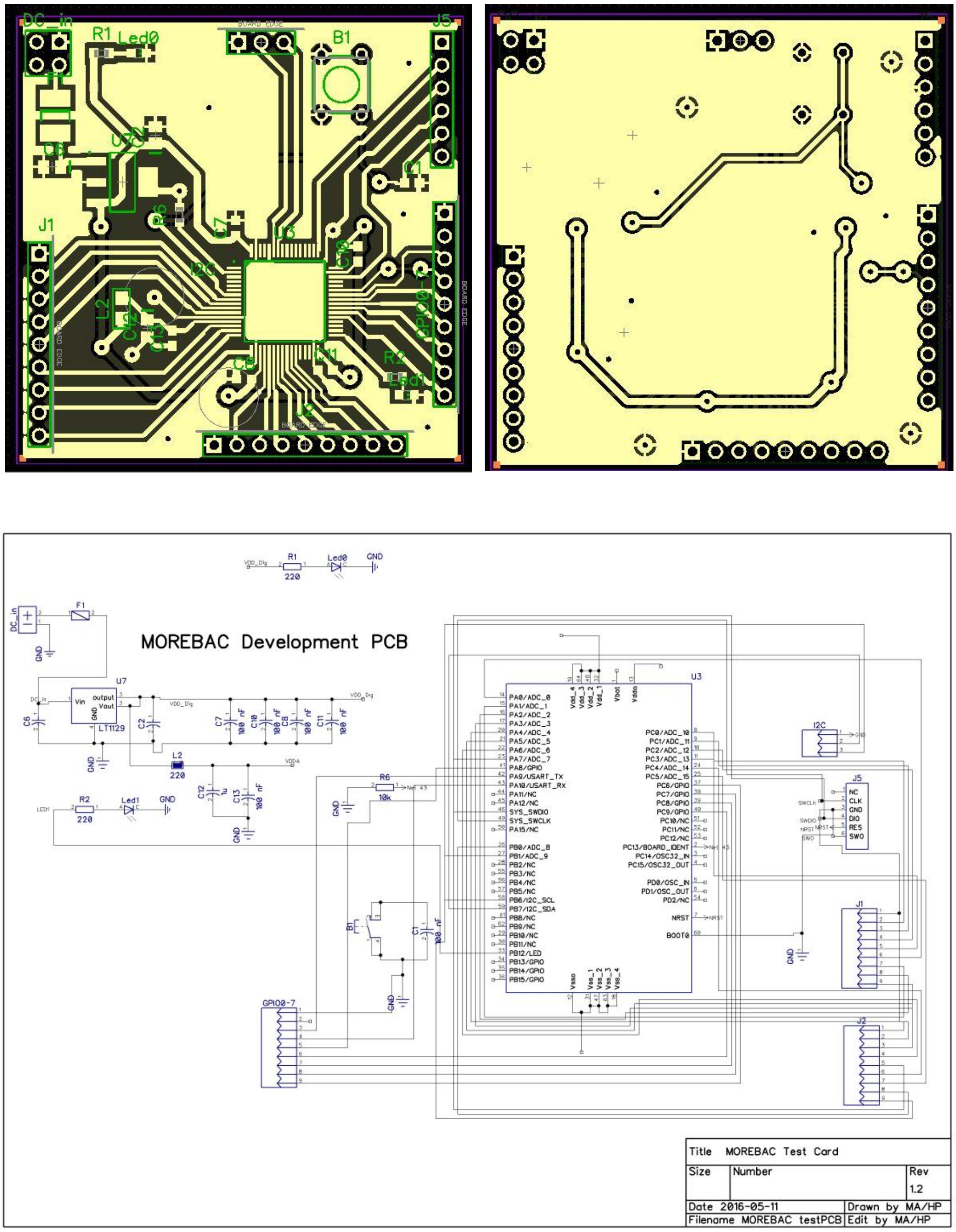


Figure 16 – PCB and schematics for MOREBAC design

4.5.2 Software

The original software for the MOREBAC test circuit was designed for an Arduino Nano and developed in the Arduino IDE. The experiment simply turned on a light, read the value from an optical sensor, then turn off the light and read the same sensor again.(Figure XX) The data was then sent via a serial interface and analyzed in another software developed in the program Processing. The software developed for the new hardware is meant to be as identical to the original software as possible. Not only in functionality but in the structure of the code, making it easy for anyone comfortable in the Arduino environment to understand and modify the code. To achieve this, a lot of the available Arduino functions have been rewritten in regular C code, usable in the developing environment for the STM32 MCU. These function can be viewed as another abstraction layer on top of the existing libraries.

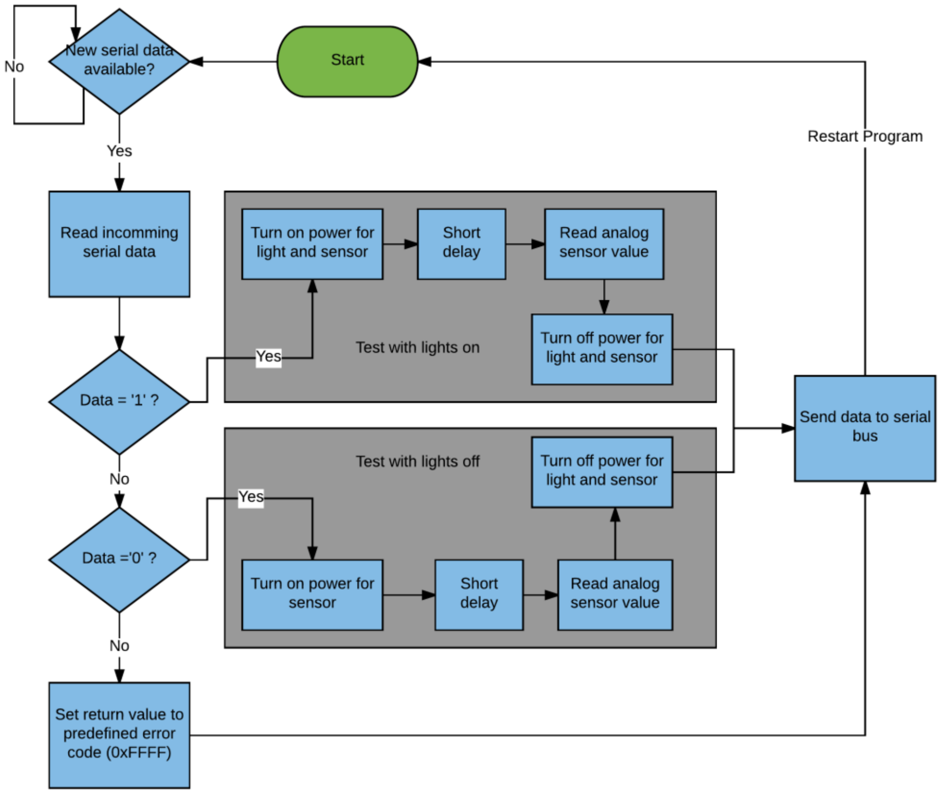


Figure 17 MOREBAC software flowchart

5 Analysis

5.1 Major results

5.1.1 ADC and DAC performance

A reasonable precision of the ADC and DAC is crucial for reliable results from the experiment. Since both the ADC and the DAC has a resolution of 12 bits, the maximum precision is calculated as shown in equation 6.

$$Minimum\ value = \frac{1}{2^{12}} * V_{DD} = \frac{1}{4096} * 3.3 \approx 0.8\ mV \tag{6}$$

The ADC is capable of measuring values between $V_{SS} \leq V_{In} \leq V_{DD}$, using the reference V_{DDA}, where $V_{DDA} \geq V_{DD}$ [21, p. 299].

Measurements on the performance of the ADC is shown in **Fel! Hittar inte referenskälla.**. As can be seen, the ADC performs well according to the expected values, and the error is within the range specified in the datasheet. The error is calculated by comparing the value read by the ADC with the value measured directly on the test circuit with a voltage meter.

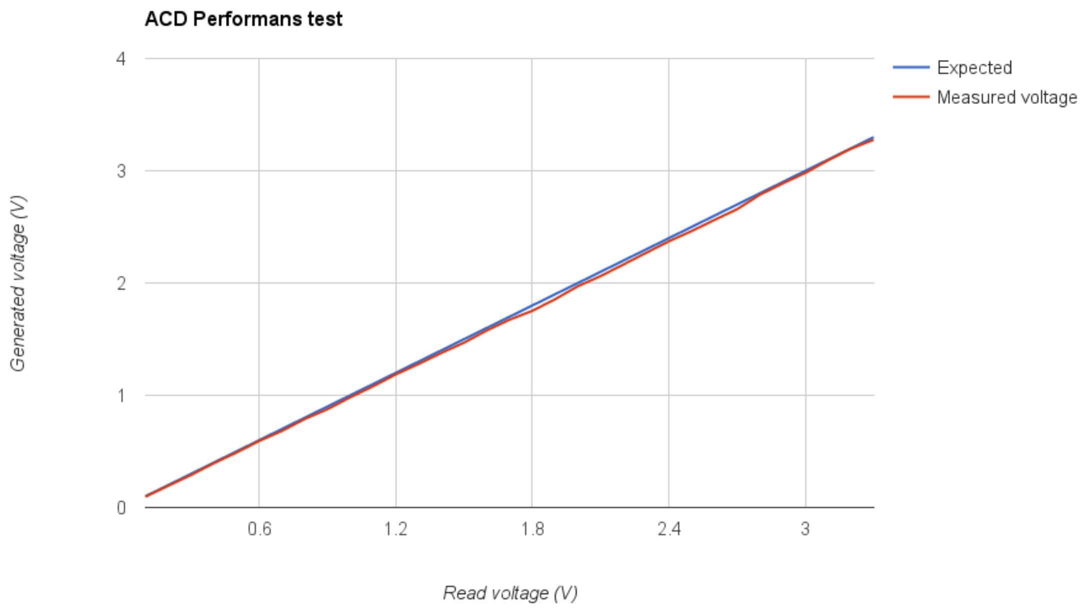


Figure 18 ADC readings vs. expected value

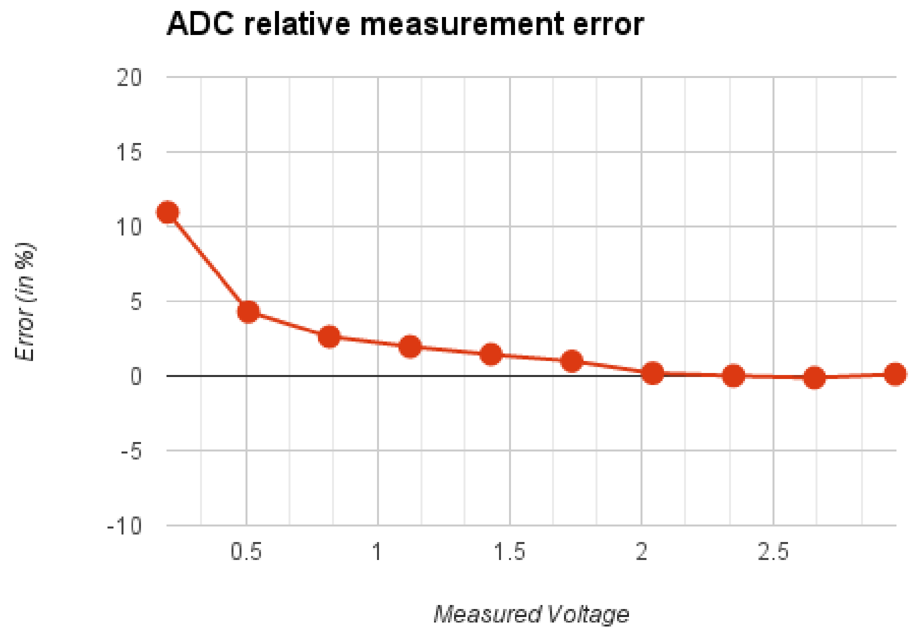


Figure 19 ADC relative measurement error

The DAC also performed as expected. An important thing to notice about the DAC output levels, which corresponds well to the data available in the reference manual, is that the specified output range of the DAC is from 0.2V to V_{REF} - 0.2V[20, p. 98]. V_{REF}, in the processor used, is the same as V_{DDA}, the effective use of the DAC for this experiment from 0.2V to 3.1V. The desired voltage range from is 0.5V to 3V, making our configuration sufficient for the experiment.

5.1.2 Power consumption

The power consumption was measured in several ways, both on the separate test PCB and on the PCB with the complete experiment. The most exact readings were measured with a current meter hooked up in series with the power supply. The current meter used, a FLUKE 45 Digital Multimeter is not able to show any type of graph with current usage as a function of time. To be able to get a usable reading from this, the whole measurement loop in the program was placed in an infinite loop. In that way, a general idea of the average current used during runtime was obtained, at about 0.9 mA.

In order to be able to see the current as a function of time, another solution was used. A resistor with 100.1 Ω was placed in series with the power supply, and the voltage drop across this resistor was measured with an oscilloscope configured in time rolling mode. The current and the power could then be calculated as shown in equation 7 and 8.

$$I = \frac{U}{R} \qquad (7)$$

$$P = U * I \qquad (8)$$

As was mentioned in 4.4.1, two solutions with voltage regulators will generate a voltage drop across the resistor, the supply voltage seen by the circuit is lower than what is actually supplied. When the supply voltage is 5V, and the regulator steps this down to 3.3V, a voltage drop of a few hundred millivolts does not matter. However, when the supply voltage is 3.3V, the voltage drop across the resistor makes the MCU run at a lower voltage, which in turn might not draw the same current.

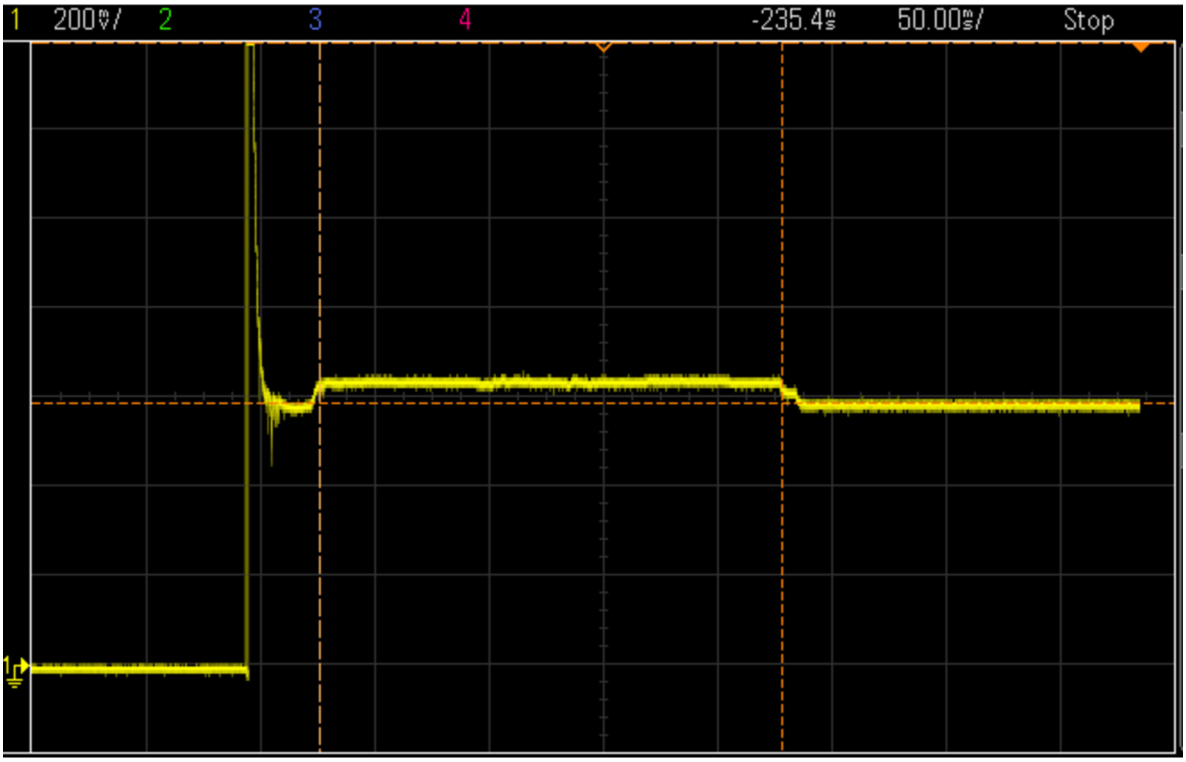


Figure 20 Voltage drop over 100ohm resistor during software runtime using LM1117

Moment	U (mV)	R (Ω)	I (mA)	Time (ms)	Power (mW)
Spike	4090	100.1	40.86	7	204.30
Initialization	575	100.1	5.74	20	28.72
Measurement loop	640	100.1	6.40	360	31.97
Clock shutdown	620	100.1	6.20	7.4	30.97
STOP mode	580	100.1	5.80	--	28.97

Table 3 - Current and power calculations, with LM1117

In Figure 21, a current spike is visible when the supply voltage is turned on. This is most likely due to the decoupling capacitors charging up. This spike lasts for about 7 ms, and has a peak of about 4090 mV, giving us a current of about 40.86 mA and a power consumption of about 204.3 mW.

When the MCU enters STOP mode, the current consumed is at about 5.8 mA. From the datasheet of the MCU, we know that the power consumption should be about one order of magnitude less than the observed consumption, at about 1μA. This is due to the voltage regulator LM1117, which according to the datasheet has a quiescent current consumption of about 5-10mA.

When changing the voltage regulator to LT1129, a significant decrease in power consumption can be noted.

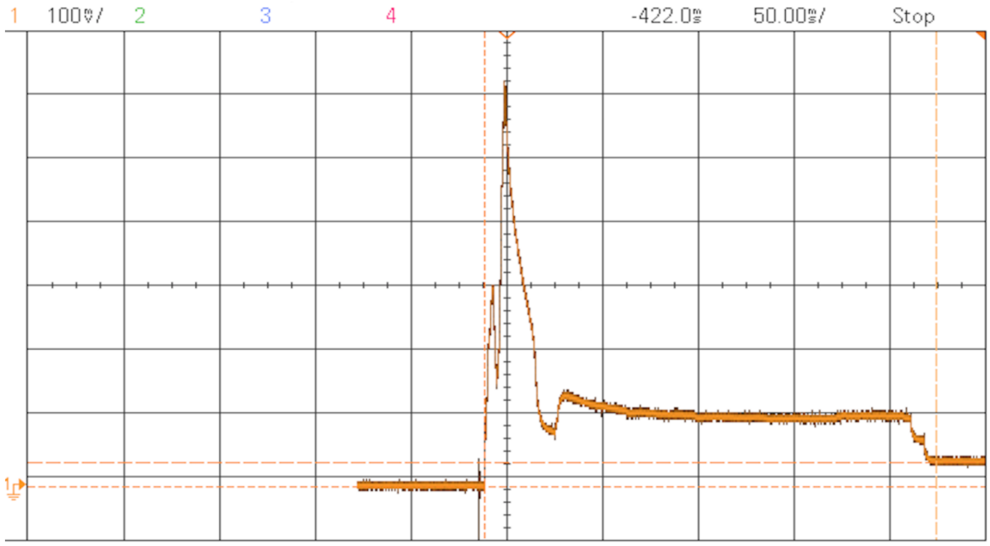


Figure 22 Voltage drop over 100ohm resistor during software runtime using LT1129

Moment	U (mV)	R (Ω)	I (mA)	Time (ms)	Power (mW)
Spike	600	99.86	6.008411776	37	84.11776487
Measurement loop	104	99.86	1.041458041	360	14.58041258
Clock shutdown	70	99.86	0.7009813739	7.4	9.813739235
STOP mode	34	99.86	0.3404766673	N/A	4.766673343

Table 4 - Current and power calculations, with LT1129

Even though we can see that the current consumption in stop mode still is well above 1µA, the maximum ratings for all parts of the program are significantly lower.

Important to note is that most of the power drawn by the experiment itself comes from the 14V battery supply. With the battery power turned off, the MCU actually draws more power, since the DAC has to power the whole test circuit.

5.2 Discussion

In a project like this, it can be hard to evaluate what parts of the work being done actually will be relevant for the final project since the deadline for the whole project is so far in the future. In this thesis, much of the focus has been on measuring the performance of the ADC and the DAC and trying to get them to operate as fast as possible. During the first few weeks, this seemed to be of high importance, based on a fear that the transistors would heat up too quick. Times in the microsecond range was discussed. Towards the end of the experiment, the importance of this was somewhat downplayed, and times in the range of milliseconds seemed to work well. An interesting insight gained from the work on the ADC/DAC performance is about the HAL library. Even though it in many cases provide easy and fast development, it is not always the best way to go. If a lower abstraction level is used when coding, it gives you a deeper understanding of how the code and the microprocessor actually works, and allows for more optimized code, at the cost of more time having to be spent reading reference manuals and datasheets.

Another part of the design that got a lot of focus was the power consumption. Although important both in respect of being able to give a precise value and in respect of using as little power as possible from the satellites battery, our experiment most likely uses a negligible amount of power compared to most other experiments. The focus of this did, however, reveal that we had an unnecessarily current draining voltage regulator, which we then could replace with a more suitable component.

Although a big part of this thesis work was to develop and program the software. This was also problematic since the variables of the experiment have not yet been decided. At the moment, we measure two different components that we provide with four different voltage levels. In the final project, there might be more than two components, and there might be more or less than four different voltage levels that have to be provided. Since the software is very straightforward, these measurements represent the bulk of the software written. Seeing how the experiment might change in the final version, this might mean that much of the code written will have to be discarded.

In the first few chapters of this thesis, the problematic environment in LEO is mentioned. The effect of his is not handled in this thesis other than used as a guideline when creating the initial requirement specification for the hardware components. Based on the information we have obtained both in the MIST project theses that were available at the beginning of this project, and on information gained at meetings with project teams working in parallel to this one, the main conclusion about the environment in which this experiment will operate, is that it will not be a big concern for the future development of this project. Still, the final product should be tested in environments tougher than the ones expected, to make sure there are margins. The intention from the beginning was to do at least some kind of environmental testing, for example running the test circuit under a longer time to see if it still behaved in an expected way, and to test the PCB in an oven and a freezer to see how extreme temperatures would affect the execution of the program and if the power consumption would stay the same. None of these test was done, however, mostly due to lack of time, and is left as future work for the project.

6 Conclusions and Future work

6.1 Conclusions

The basic goals described in 1.4 have all been met. Though, much of what has been developed requires further work in order to be considered final. For example, even though a proposal for a final PCB has been made, a more finalized version has to be drawn. There are several aspects of the PCB design that are not optimal, like some traces that are unnecessarily long, and the placement of many components are not optimal. For example, a few of the decoupling capacitors are placed too far away. The current version of the PCB is missing one key component, the voltage regulator, designed by Simon Johansson[6], which used to step up the battery supply voltage to 48V for the Piezo LEGS. Still, the state the project is in now is very suitable for a new student to use as a starting point for the final work on the project.

One of the important conclusions drawn from the project itself is the fact that when you work on a project that is heavily dependent on other projects developed in parallel, the exchange of updated information is crucial. It is also good to keep in mind that if whatever specifications you are using as a guideline for what you are developing are subject to change, don’t try to develop further than to the point where it meet your own requirements, but still are able to change and revert what you have done, without too much work. A good example of this the fact that when the project began, it was not yet decided whether this experiment would be supplied with 3.3V or with 5V, and if the experiment would be supplied with power only during the measurements, or if we would stay in an idle mode for some time. A similar situation arose with the OBC, regarding the exact specification of the I2C messages. These factors all resulted in time being spent on separate solutions for different situations and thus slow down the development process.

6.2 Limitations

The main limiting factor in the project was the time constraint. One of the most important factors was that much of the important parts of the MIST project was being developed in parallel to this experiment resulting either redoing things when new specifications appeared, or working with different solutions at the same time, as mentioned above.

The limitations of our results of the projects are mainly that no environmental tests have been done, so our measurements are only valid at room temperature, and so far we cannot say how relevant the work we have done during the entire project will actually be for the final product.

6.3 Future work

This chapter is one of the most important for the students taking over this project to finish the work on the experiment. Here, suggestions for future work are presented, based on what was not done during this part of the project.

6.3.1 Hardware improvement

The layout of the prototype board constructed in this project is in no way optimized, but rather just made to work and fulfill its purpose as a proof of concept. There are two main tasks to take care of when redesigning the PCB:

1. The step-up regulator developed by Johansson[6] to provide Piezo LEGS with 48 volts must be fitted on the experiment.
2. EMC regulations must be considered to be sure that the final experiment PCB that is put on the satellite is not susceptible to electromagnetic interference, or at risk of causing interference that may damage other experiments. Guidelines for this are available from the MIST team.

A suggestion that was discussed during the development but not realized because of the lack of time is to make the final version of the experiment as a 4-layer PCB rather than the current 2-layer version. This would be a good starting point in solving both of the above-mentioned concerns. A good idea would be to let an experienced constructor evaluate the final design.

The schematic for the final version also contains an error in the design provided by Matthias Ericsson and Johan Silverudd[5] that has to be corrected. This is the power supply to the operating amplifier named "OpUbe" that is not connected in the schematic, nor in the PCB layout. This is the reason for the two white wires that can be seen on the prototype board.

6.3.2 Software improvement

The current software, just as the hardware, is not intended to be the final version for the entire experiment. The running time of the program can probably be shorter by removing delays that may or may not have to be there. To do this, some more testing of the DAC and ADC performance will have to be done.

An optional task, if deemed necessary, would be to improve the precision of the ADC readings by introducing a noise to the ADC channels. By creating a square wave with the MCU, for example by using one of the timers, and connecting this signal via a resistor and capacitor in series, a quasi-triangle wave can be generated. If this signal is connected in parallel with the input to the ADC, and an average is taken of this signal (just as in the current implementation of the code), a more accurate reading could be achieved. At this time, the 16 measurements taken are very likely to always have the same value, making the whole process of taking an average obsolete.

The current program contains a lot of unused code such as small test functions. All this should be removed when implementing the final version of the code.

The construction of the checksum for the I2C message is at the moment defined as a very simple modular checksum. Discussions with the OBC team is needed in order to confirm if this solution will work, or a more advanced checksum must be implemented.

The ADC is capable of reading the internal temperature of the MCU, without using a separate GPIO. This function could be implemented and used. If the MCU for some reason would overheat, this information would be useful not only for the SiC experiment but for the whole satellite since this would alter the expected thermal conditions of the satellite.

6.3.3 Software testing

The current I2C communication is based on tests done between two STM32 microcontrollers configured as Master and Slave. To make sure the communications actually work, tests should be made with the actual OBC.

Long term tests of the entire project should be considered to make sure the software behaves as expected every time it runs, even after longer periods of constant on and of switching of the MCUs power supply.

6.3.4 Hardware Testing

When evaluating the final power consumption for the whole experiment, new tests would have to be done, since the current tests are made without the SiC transistor being mounted on the PCB and used.

By using the current software for the prototype board, or an improved version of the software, basic functionality testing of the whole circuit can be made. The next step is to test the functionality of the experiment in a spacelike environment. This means exposing the experiment to things like extreme temperature, vacuum, and radiation, and run the software while exposed to these circumstances. Most of these test could hopefully be done in a laboratory at KTH. If no such equipment can be accessed, at least extreme temperatures can be tested by using basic household equipment like a freezer and a household oven. These tests could be done using the prototype board, to evaluate the components used, but should also be made with the final PCB to test the experiment as a whole.

Another important factor to test the final hardware for is vibrations. When the satellite is being launched the vibrations of the rocket is severe, and will require the physical construction of the experiment to be very rigid, and everything soldered and mounted to withstand those vibrations. Also, components like ceramic capacitors may be sensitive to vibrations. To avoid having the experiment damaged during launch, several test should be made on the actual implementation of the experiment, to make sure nothing is loose or at risk of being damaged.

The environmental tests can be conducted in several steps according to these document developed by NASA[27]. The testing for the SiC in Space experiment may not have to be as thorough as described there since the LEO environment is supposedly more forgiving, but if the possibility exists, as many tests as possible should be conducted.

An important thing to remember when testing is that the main purpose of the experiment is to test the Silicon Carbide components in a space environment. Therefore, to improve the design of the experiment, other than to make sure that the components that are NOT under test are working properly, would make the test results less meaningful.

6.3.5 Power supply

At the time when this report being printed, it is not yet confirmed whether the MCU on the experiment will be supplied with 5 or 3.3 Volts from the power supply on the satellite. In both cases, some work has to be done to make sure that the supply voltage is at the intended level for the processor. The current PCB layout is fitted with a 5 to 3.3V linear voltage converter. As mentioned in this report, the LM1117 used proved to have a higher power consumption than required. If the experiment will be supplied with 5 volts when mounted on the satellite, the LM1117 should be replaced by a lower power regulator, e.g. the LT 1129 as mentioned in chapter 5. If the component is replaced, it is important to redraw the schematic and PCB layout, since even if the two components are using the same footprint, the pads are connected differently. A correct usage of this component can be found in the MOREBAC implementation of the PCB.

If the supply voltage on the final version is 3.3V other things have to be taken into consideration. At the moment, the supply voltage for the MCU is the same as the supply and reference voltage for the ADC and DAC. (VDD_A). This means that if the 3.3V from the battery is not stable, the reference voltage for the measurements will not be the one expected, and the measurements will not only be useless, but there will not be any indication that they are actually wrong. Possible solutions would be to run the MCU at a lower voltage, for example, 3.1 V. By regulating the 3.3 V down to 3.1V, the regulator could keep the signal steady at 3.1, even though the 3.3V signal is noisy. Another solution would be to run the MCU at 3.3 V, but to keep a reference voltage at a lower, steady level. This could then be read with the one remaining ADC input, and used as a reference for the other readings. There are several workarounds for this problems, and one of the more important future work will be to solve this in the best way possible.

6.4 Reflections – Benefits, ethics, and sustainability

There is a growing concern of space debris in LEO. There are currently 22,000 space debris elements that are being tracked by the U.S surveillance networks [28]. This is seen as a big problem since the risk of the debris to collide with working satellites are getting higher. Space debris colliding with other space debris creates, even more, debris, making the problem bigger and bigger.

Our satellite is, of course, a part of this problem. If all goes to plan, the satellite will burn up when re-entering the atmosphere in a few years, but if the satellite were to collide with existing space debris, which is unlikely, but indeed possible, this would not only mean that our satellite is lost, it would also create even more debris.

There are of course also benefits to be gained in this project. Thanks to the CubeSat initiative, projects like this are more feasible than ever. This gives us a possibility to conduct more experiments that can give us new knowledge about the world we live in and the space around us. With projects like these, institutions like KTH can start new space programs that push us further into the space age.

While this particular CubeSat might not yield an economic return on investment, the experiments being done on SiC could help us learn new valuable insights that can be a part of further research in the field, which at some point might result in new products and new inventions.

One ethical aspect that can be connected to space projects, in general, is that much of what is being developed are used for military or surveillance purposes. One example of this is that the U.S. Defense Advanced Research Project (DARPA) wants to use CubeSat satellites to create a new global communications network [29].

References

[1] Larry W. Esposito, *Exploring Venus as a Terrestrial Planet*. Hoboken: Wiley, 2013.
[2] Muthu B. J. Wijesundara, *Silicon Carbide Microsystems for Harsh Environments*. Dordrecht: Springer, 2011.
[3] "MIST - the MIniature Student saTellite," *MIST - the MIniature Student saTellite*. [Online]. Available: https://mistsatellite.wordpress.com/. [Accessed: 27-May-2016].
[4] "Arduino - Home." [Online]. Available: https://www.arduino.cc/. [Accessed: 27-May-2016].
[5] M. Eriksson and J. Silverudd, "Design of measurements circuits for SiC experiment, KTH student satellite MIST," KTH, Royal Institute of Technology, Stockholm, Bachelor Thesis, 2016.
[6] S. Johansson, "Design of power supplies for Sic and Piezo LEGS experiment, KTH student satellite MIST," KTH, Royal Institute of Technology, Stockholm, Bachelor Thesis, 2016.
[7] "KTH | The Student Satellite MIST." [Online]. Available: https://www.kth.se/en/sci/centra/rymdcenter/studentsatellit. [Accessed: 27-May-2016].
[8] "CubeSat," *CubeSat*. [Online]. Available: http://www.cubesat.org/. [Accessed: 31-May-2016].
[9] E. Hand, "The rise of the CubeSat," *Science*, vol. 346, no. 6216, p. 1449, 2014.
[10] C.-M. Zetterling, *Process technology for silicon carbide devices*. 2002.
[11] A. Berggren, *Design of Thermal Control System for the Spacecraft MIST*. 2015.
[12] S. Corpino, M. Caldera, F. Nichele, M. Masoero, and N. Viola, "Thermal design and analysis of a nanosatellite in low earth orbit," *Acta Astronaut.*, vol. 115, pp. 247–261, 2015.
[13] E. G. Stassinopoulos and K. A. LaBel, "The Near-Earth Space Radiation for Electronics Environment," 01-Jan-2004.
[14] C. S. Guenzer, E. A. Wolicki, and R. G. Allas, "Single Event Upset of Dynamic Rams by Neutrons and Protons," *Nucl. Sci. IEEE Trans. On*, vol. 26, no. 6, pp. 5048–5052, 1979.
[15] "ISIS-Innovative Solutions In Space | The nanosatellite specialist," *ISIS - Innovative Solutions in Space*. [Online]. Available: http://www.isispace.nl/. [Accessed: 27-May-2016].
[16] "MDK for STM32L0 and STM32F0 - Installation & Activation." [Online]. Available: http://www2.keil.com/stmicroelectronics-stm32/mdk. [Accessed: 31-May-2016].
[17] "STM32CubeMX - STMicroelectronics." [Online]. Available: http://www.st.com/content/st_com/en/products/development-tools/software-development-tools/stm32-software-development-tools/stm32-configurators-and-code-generators/stm32cubemx.html?sc=stm32cubemx. [Accessed: 31-May-2016].
[18] "DipTrace - Schematic and PCB Design Software." [Online]. Available: http://diptrace.com/. [Accessed: 31-May-2016].
[19] "32L0538DISCOVERY - STMicroelectronics." [Online]. Available: http://www.st.com/content/st_com/en/products/evaluation-tools/product-evaluation-tools/mcu-eval-tools/stm32-mcu-eval-tools/stm32-mcu-discovery-kits/32l0538discovery.html. [Accessed: 27-May-2016].
[20] ST-Microelectronics, "'Description of STM32L0 HAL and Low-layer drivers', UM1749 user manual." 21-Mar-2016.
[21] ST-Microelectronics, "RM0367 Reference Manual | Ultra-low-power STM32L0x3 advanced ARM®-based 32-bit MCUs." 19-Feb-2016.
[22] Texas Instruments, "'LMT85/LMT85-Q1 SC70/TO-92, Analog Temperature Sensors with Class-AB Output', SNIS168C datasheet." Oct-2015.

[23] “STM32L053C6 - STMicroelectronics.” [Online]. Available: http://www.st.com/content/st_com/en/products/microcontrollers/stm32-32-bit-arm-cortex-mcus/stm32l0-series/stm32l0x3/stm32l053c6.html. [Accessed: 27-May-2016].
[24] ST-Microelectronics, “AN3216 - Getting started with STM32L1xxx hardware development | Application Note.” 20-Jun-2016.
[25] Linear Technologies, “‘LT1129/LT1129-3.3/LT1129-5 Micropower Low Dropout Regulators with Shutdown’, 112935ff datasheet.” 04-Dec-2008.
[26] “STM32L053R6 - STMicroelectronics.” [Online]. Available: http://www.st.com/content/st_com/en/products/microcontrollers/stm32-32-bit-arm-cortex-mcus/stm32l0-series/stm32l0x3/stm32l053r6.html. [Accessed: 30-May-2016].
[27] NASA Technical Standards System (NTSS), “General Environmental Verification Standard | GSFC-STD-7000.” [Online]. Available: https://standards.nasa.gov/standard/gsfc/gsfc-std-7000. [Accessed: 27-May-2016].
[28] Joseph N. Pelton, *New Solutions for the Space Debris Problem*. Cham: Springer International Publishing, 2015.
[29] “Inter-Satellite Communications Links - DARPA-BAA-15-43 (Archived) - Federal Business Opportunities: Opportunities.” [Online]. Available: https://www.fbo.gov/index?s=opportunity&mode=form&id=09ca0b003df747fea1a89d5e4e309afa&tab=core&_cview=0. [Accessed: 31-May-2016].

Appendix A - Requirements Specification

The requirements specification for the microcontroller (MCU) unit and its functions, its schematics, application and documentation are listed below. Note that the requirements differ for SiC and MOREBAC.

These requirements exceed what is accomplished in this report and in this iteration. These are the requirements for the complete launch ready experiments.

1. **Physical requirements MOREBAC/SiC**
    - The size of the MCU and other needed components must be kept to a minimum.
    - All components must be suited for the surrounding environment in low earth orbit.
    - All components must be surface mounted.
    - The radiated heat from the components must be kept to a minimum.
    - The radiated electromagnetic forces from the circuit must be kept to a minimum.
    - The MCU must be fitted on a printed circuit board (PCB) together with the rest of the experiment, following the exact specification regarding size and mounting to be placed on the CubeSat Satellite.
2. **Power requirements MOREBAC/SiC**
    - The power consumption of the MCU and the surrounding components must be kept to a minimum.
    - The total power used must be calculated with a precision of 0.05 W.
    - The MCU must be able to handle a supply voltage of 5 V. If the MCU has a maximum rating of less than that, the supply voltage has to be regulated to a suitable level.
3. **Interface requirements MOREBAC/SiC**
    - The MCU must be able to communicate with the On Board Computer (OBC), using I2C.
    - The MCU must be equipped with enough analog-to-digital-converters (ADC) to read all measurements necessary for the experiment.
4. **Interface requirements SiC**
    - The MCU must be equipped with a digital-to-analog-converter (DAC), to provide a variable voltage source for experiments.
5. **Software requirements MOREBAC/SiC**
    - The software written for the MCU must follow the expected behavior set by the OBC:
        - When the power bus for the experiment is turned on, the MCU must start and perform its experiments and readings.
        - When communicating with the OBC, the program must use I2C and act in the following way:
            - The experiment must be set to slave in the I2C interface
            - The experiment must be able to act according to the message sent by the OBC
            - When sending a message to the OBC, the experiment must package the payload in the following way:

- The first four bytes must contain the length of the payload (in bytes). These bytes are not considered in the payload length.
- Following those bytes, the data from the experiments are sent.
- The end of the payload must be a checksum value.

6. **Software requirements SiC**
   - The software must be able to provide a variable voltage source for the experiments.
   - The software must read the temperature for both experiments.
   - The software must be able to do three different measurements on both experiments
   - The software must be able to take several measurements in order to create an average reading.
   - The software must be able to control the voltage regulator that supplies power to the experiments.
7. **Software requirements MOREBAC**
   - The software should perform the same tasks, in a similar fashion, as the one written for the Arduino version of the current MOREBAC experimental circuit.
   - The software should be easy to understand for the student who wrote the initial Arduino software, or to someone with equal or better knowledge about c-programming.
   - The software should be easy to expand to a more complex circuit, based around the MOREBC hardware developed in this project.
8. **Documentation**
   - A document providing detailed information on how to read and use the collected data from the experiments must be produced.
   - A document detailing what applications has been used to create the software, the schematics, and the test environment, and how to correctly open and edit these files must be produced.
9. **Other**
   - All code, schematics, and other documentation must be made available for continued work.

Appendix B - Test Plan

These tests were conducted on the evaluation board from ST, the separate PCB and the final PCB proposal (where applicable). The test data shown in this thesis are the most relevant test situations. Test that was not possible to do is marked as ~~Strikethrough.~~

Basic software test

- Initialization of code at different clock speeds.
- Running of MCU in different run modes.

Communication tests

- Sending and receiving I2C messages between two separate boards.
- Sending USART message to PC
- ~~Communicating with OBC via I2C~~

ADC test cases

- Measuring different voltage levels generated by separate source on a single ADC channel
- Measuring different voltage levels generated by separate source on all ADC channels
- Finding minimum stable ADC reading time
- Measuring accuracy of ADC

DAC test cases

- Generating variable voltages from DAC, read on Oscilloscope
- Defining maximum and minimum voltages

DAC/ADC test cases

- Measuring different voltages on ADC, generated by DAC

Hardware test

- Basic functionality of all components and connections
- Power consumption measurements, with and without experiment running.
    - On MCU supply port with 3.3V
    - On MCU supply port with 5V

Test on final PCB

- Running of complete experiment with Silicon transistor attached.
- ~~Running of complete experiment with both Silicon and SiC transistor attached.~~

~~Environmental tests~~

- ~~Temperature tests~~
- ~~Shake tests~~
- ~~Vacuum tests~~

Appendix C - I2C Message

The I2C message follows the structure described below. Each block represents one byte of data. Numbers in parenthesis denote what byte of the value this is.

Length (1)	Length (2)	Length (3)	Length (4)	Si Temp @ 3.1V (1)	Si Temp @ 3.1V (2)	Si V_{BE} @ 3.1 V (1)		
Si V_{BE} @ 3.1V (2)	Si V_{RB} @ 3.1V (1)	Si V_{RB} @ 3.1V (2)	Si V_{RC} @ 3.1V (1)	Si V_{RC} @ 3.1V (2)	Si Temp @ 2.1V (1)	Si Temp @ 2.1V (1)	Si V_{BE} @ 2.1 V (1)	Si V_{BE} @ 2.1V (2)
Si V_{RB} @ 2.1V (1)	Si V_{RB} @ 2.1V (2)	Si V_{RC} @ 2.1V (1)	Si V_{RC} @ 2.1V (2)	Si Temp @ 1.1V (1)	Si Temp @ 1.1V (1)	Si V_{BE} @ 1.1 V (1)	Si V_{BE} @ 1.1V (2)	Si V_{RB} @ 1.1V (1)
Si V_{RB} @ 1.1V (2)	Si V_{RC} @ 1.1V (1)	Si V_{RC} @ 1.1V (2)	Si Temp @ 0.5V (1)	Si Temp @ 0.5V (1)	Si V_{BE} @ 0.5 V (1)	Si V_{BE} @ 0.5V (2)	Si V_{RB} @ 0.5V (1)	Si V_{RB} @ 0.5V (2)
Si V_{RC} @ 0.5V (1)	Si V_{RC} @ 0.5V (2)	SiC Temp @ 3.1V (1)	SiC Temp @ 3.1V (2)	SiC V_{BE} @ 3.1 V (1)	SiC V_{BE} @ 3.1V (2)	SiC V_{RB} @ 3.1V (1)	SiC V_{RB} @ 3.1V (2)	SiC V_{RC} @ 3.1V (1)
SiC V_{RC} @ 3.1V (2)	SiC Temp @ 2.1V (1)	SiC Temp @ 2.1V (1)	SiC V_{BE} @ 2.1 V (1)	SiC V_{BE} @ 2.1V (2)	SiC V_{RB} @ 2.1V (1)	SiC V_{RB} @ 2.1V (2)	SiC V_{RC} @ 2.1V (1)	SiC V_{RC} @ 2.1V (2)
SiC Temp @ 1.1V (1)	SiC Temp @ 1.1V (1)	SiC V_{BE} @ 1.1 V (1)	SiC V_{BE} @ 1.1V (2)	SiC V_{RB} @ 1.1V (1)	SiC V_{RB} @ 1.1V (2)	SiC V_{RC} @ 1.1V (1)	SiC V_{RC} @ 1.1V (2)	SiC Temp @ 0.5V (1)
SiC Temp @ 0.5V (1)	SiC V_{BE} @ 0.5 V (1)	SiC V_{BE} @ 0.5V (2)	SiC V_{RB} @ 0.5V (1)	SiC V_{RB} @ 0.5V (2)	SiC V_{RC} @ 0.5V (1)	SiC V_{RC} @ 0.5V (2)	Checksum	

Appendix D – Source Code

Note that in order to program or debug the final PCB versions of the experiment, a separate MCU capable of programming via SWDIO is required. A suggestion for such an MCU is “Discovery kit for STM32L0 series with STM32L053C8 MCU” from ST.

The code for the main program is available on GitHub:

https://github.com/miandre/MiCSiC/tree/master/Master/SiCMiC

To run this code, download Keil from http://www.keil.com/ and open MiCSiC/Master/SiCMiC/MDK-ARM/SiCMiC.uvprojx

The code for the MORBAC project is available on GitHub:

https://github.com/miandre/MiCSiC/tree/master/MORBAC

To run this code, download Keil from http://www.keil.com/ and open MiCSiC/MORBAC/MDK-ARM/Morbac.uvprojx

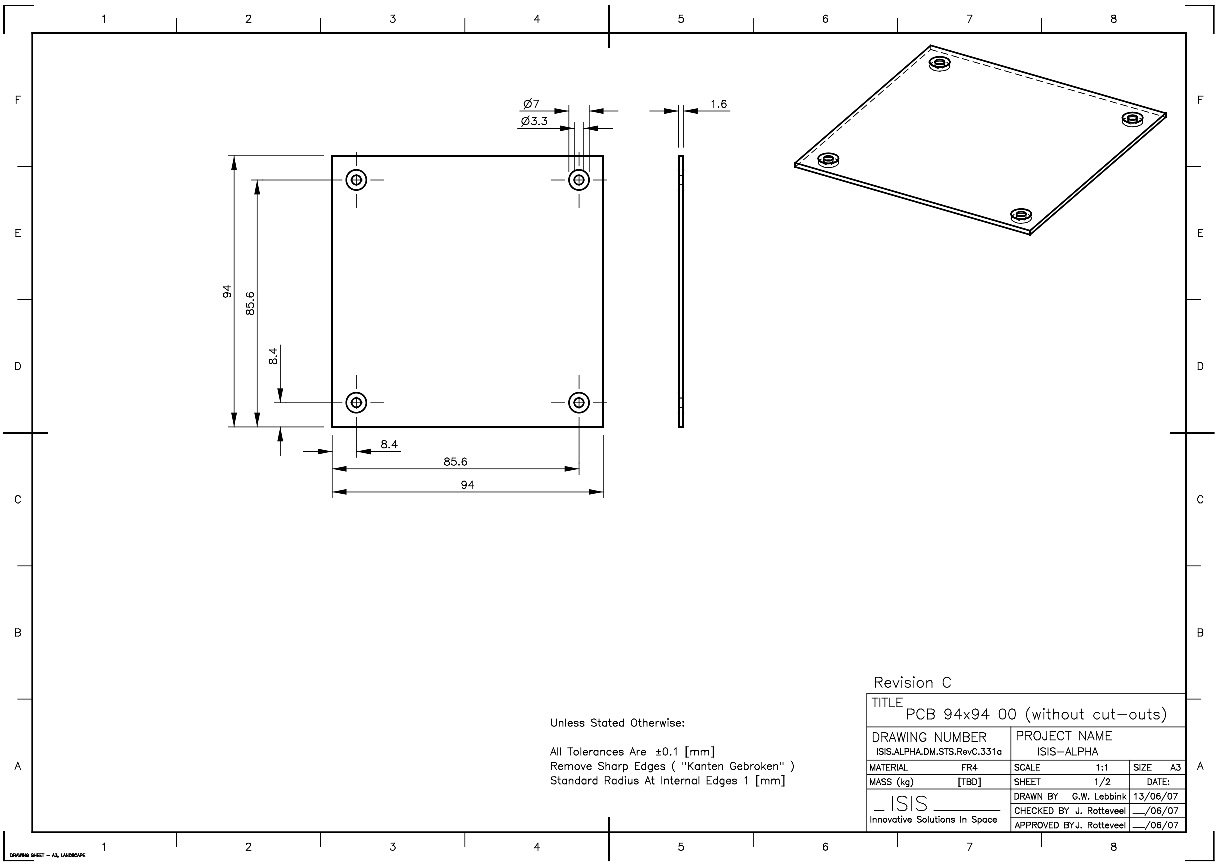
The code for the separate analyzing program is available on GitHub:

https://github.com/Happsson/SiC-Analyzing

To run this code, download and install Processing from https://processing.org/ and open SiC_reader.pde

Appendix E – PCB dimensions for MIST experiment

These dimensions are developed by ISIS.



Appendix F – PCB Component List

Component list MOREBAC PCB

Name	Type	Value
B1	Button	RESET
C1	CAP_0603	100 nF
C2	CAP_0805	10 uF
C6	CAP_0805	10 uF
C7	CAP_0603	100 nF
C8	CAP_0603	100 nF
C10	CAP_0603	100 nF
C11	CAP_0603	100 nF
C12	CAP_0805	1u
C13	CAP_0603	100 nF
DC_in	DC_in	
F1	Fuse_1812	750mA
GPIO0-7	644456-9	Pin-header
I2C	644456-3	Pin-header
J1	644456-9	Pin-header
J2	644456-9	Pin-header
J5	SWD	Pin-header
L2	Ferrit 0805	220
Led0	LED_1609	
Led1	LED_1609	
R1	RES_0603	220
R2	RES_0603	220
R6	RES_0603	10k
U3	STM32L053R6	
U7	LT1129	3,3V

Component list TEST PCB

Name	Type	Value
B1	Button	RESET
C1	CAP_0603	100 nF
C5	CAP_0805	10 uF
C6	CAP_0805	10 uF
C7	CAP_0603	100 nF
C8	CAP_0603	100 nF
C9	C 157 SMD	150u
C10	CAP_0603	100 nF
C11	CAP_0603	100 nF
C12	CAP_0805	1u
C13	CAP_0603	100 nF
DC_in	DC_in	
F1	Fuse_1812	750mA
I2C	644456-3	Pin-Header
J1	1-644456-1	Pin-Header
J2	1-644456-4	Pin-Header
J5	SWD	Pin-Header
L2	Ferrit 0805	220
Led0	LED_1609	
Led1	LED_1609	
Led2	LED_1609	
Led3	LED_1609	
Led4	LED_1609	
R1	RES_0603	220
R2	RES_0603	220
R3	RES_0603	220
R4	RES_0603	220
R5	RES_0603	220
R6	RES_0603	10k
R7	RES_0603	0
R8	RES_0603	0
U2	MCU, LQFP48	STM32L053C6
U7	LM1117	3,3V

Component list FINAL PCB

Name	Type	Value
10K1	RES_0603	1.1k
10uF	CAP_1206	10uF
10uFu	CAP_1206	10uF
1nF	CAP_0603	1nF
47K	RES_0603	47k
750K	RES_0603	750k
B1	Button	RESET
C1	CAP_0603	100 nF
C2	CAP_0603	0.1uF
C3	CAP_0603	0.1uF
C4	CAP_0603	0.1uF
C5	CAP_0805	10 uF
C6	CAP_0805	10 uF
C7	CAP_0603	100 nF
C8	CAP_0603	100 nF
C10	CAP_0603	100 nF
C11	CAP_0603	100 nF
C12	CAP_0805	1u
C13	CAP_0603	100 nF
C1.1.1	CAP_0603	47pF
C1.1.2	CAP_0603	470pF
C1.1.3	CAP_0603	47pF
C1.1.4	CAP_0603	1uF
C1.1.5	CAP_0603	0.1uF
C1.2.1	CAP_0603	47pF
C1.2.2	CAP_0603	470pF
C1.2.3	CAP_0603	47pF
C1.2.4	CAP_0603	1uF
C1.2.5	CAP_0603	0.1uF
C2.1.1	CAP_0603	47pF
C2.1.2	CAP_0603	470pF
C2.1.3	CAP_0603	47pF
C2.1.4	CAP_0603	1uF
C2.1.5	CAP_0603	0.1uF
C2.2.1	CAP_0603	47pF
C2.2.2	CAP_0603	470pF
C2.2.3	CAP_0603	47pF
C2.2.4	CAP_0603	1uF
C2.2.5	CAP_0603	0.1uF
D1.1.1	1N4148WS	
D1.1.2	1N4148WS	
D1.2.1	1N4148WS	
D1.2.2	1N4148WS	
D2.1.1	1N4148WS	
D2.1.2	1N4148WS	
D2.2.1	1N4148WS	
D2.2.2	1N4148WS	
DC_in	DC_in	
Diff1.1	AD8226ARZ	
Diff1.2	AD8226ARZ	
Diff2.1	AD8226ARZ	
Diff2.2	AD8226ARZ	
F1	Fuse_1812	750mA
I2C	644456-3	Pin-header
J5	SWD	Pin-header
L2	Ferrit 0805	220
MMBT2369ALT1G	MMBT2369ALT1G	
Op1	LT1638HS8	
Op2	LT1638HS8	
OpUbe	LT1638HS8	
OpUbe0	LT1638HS8	
Q1	FDC637AN	
Q2	FDC637AN	
R6	RES_0603	10k
R7	RES_0603	0
R.2.2	RES_0603	22k
R1.1	RES_0603	10k
R1.2	RES_0603	22k
R1.1.1	RES_0603	4.02k
R1.1.2	RES_0603	4.02k
R1.1.3	RES_0603	10k
R1.2.1	RES_0603	4.02k
R1.2.2	RES_0603	4.02k
R1.2.3	RES_0603	10k
R2.1	RES_0603	10k
R2.1.1	RES_0603	4.02k
R2.1.2	RES_0603	4.02k
R2.1.3	RES_0603	10k
R2.2.1	RES_0603	4.02k
R2.2.2	RES_0603	4.02k

R2.2.3	RES_0603	10k
RB1.1	RES_0603	47k
RB1.2	RES_0603	100k
RB2.1	RES_0603	15k
RB2.2	RES_0603	22k
RC1	RES_0603	300
RC2	RES_0603	
SiCBJT	SiC-transistor	

Temp1	LMT85DCKT	
Temp2	LMT85DCKT	
U1	3062	
U2	MCU, LQFP48	STM32L053C 6
U7	LM1117	3,3V
USART	644456-4	Pin-header
Vin	644456-2	Pin-header

TRITA-ICT-EX-20YY:XX