TALLINN UNIVERSITY OF TECHNOLOGY
SCHOOL OF ENGINEERING
Department of Mechanical and Industrial Engineering

DEVELOPING AND TESTING AN ADC FOR KUUPKULGUR LUNAR ROVER

ADC TARKVARA ARENDAMINE JA TESTIMINE KUUPKULGURI PROJEKTI JAOKS

BACHELOR THESIS

Student: Allar Kütt

Student code: 210830MVEB

Supervisors: Tauno Otto, Programme Director
Laur Edvard Lindmaa, System engineer

Tallinn 2024

AUTHOR'S DECLARATION

Hereby I, Allar Kütt, declare that I have written this thesis independently.

No academic degree has been applied for based on this material. All works, major viewpoints and data of the other authors used in this thesis have been referenced

30. Dec. 2024

Author: ....(*signed digitally*)

/signature /

Thesis is in accordance with terms and requirements

30. Dec. 2024

Supervisor Tauno Otto: …........................(*signed digitally*)

/signature/

Accepted for defence

“.......”....................20… .

Chairman of theses defence commission: ....

/name and signature/

Non-exclusive licence for reproduction and publication of a graduation thesis

(signed digitally)

I, Allar Kütt,

1. grant Tallinn University of Technology free licence (non-exclusive licence) for my thesis "DEVELOPING AND TESTING AN ADC FOR KUUPKULGUR LUNAR ROVER"

supervised by: Tauno Otto and Laur Edvard Lindmaa

1.1 to be reproduced for the purposes of preservation and electronic publication of the graduation thesis, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright;

1.2 to be published via the web of Tallinn University of Technology, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright.

2. I am aware that the author also retains the rights specified in clause 1 of the non-exclusive licence.

3. I confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights, the rights arising from the Personal Data Protection Act or rights arising from other legislation.

30.12.2024

Department of Mechanical and Industrial Engineering

THESIS TASK

Student: Allar Kütt, 210830MVEB

Study programme: MVEB, Integrated Engineering

Supervisor(s): Programme Director, Tauno Otto, 53090118

Kuupkulgur MTÜ, System engineer, Laur Edvard Lindmaa

Thesis topic:

Eng. Developing and testing an ADC for KuupKulgur lunar rover

Est. ADC tarkvara arendamine ja testimine KuupKulguri projekti jaoks

Thesis main objectives:

1. Gain diagnostic data from the ADC-s within the EPS through programming a working driver for the given ADC-s.
2. Display the results within a simple UI

Thesis tasks and time schedule:

No	Task description	Deadline
1.	Write introduction, literature review	31.03
2.	Description of the methodology	15.04
3.	Research section (write the driver, test hardware and software)	30.04
4.	Analysis of the results	10.05
5.	Summary and final formatting	20.05

Language: English **Deadline for submission of thesis:** 30. Dec. 2024

Student: Allar Kütt ……………........ 30. Dec. 2024

/signature/

Supervisor: Tauno Otto …………………….. 30. Dec. 2024

/signature/

Head of study programme: …………… .... “.......”......................20…..a

/signature/

Terms of thesis closed defence and/or restricted access conditions to be formulated on the reverse side

CONTENTS

PREFACE

The given thesis topic was offered by Tartu observatory, most of the work and testing was done there. The task and instructions for the work were given by the KuupKulgur project system engineer Laur Edvard Lindmaa.

List of abbreviations and symbols

EPS – Electrical power system

ADC – Analog to digital converter

PCB – Printed circuit board

PCBA - Printed circuit board assembly

IDE - Integrated Development Environment

UI - user interface

CAD – Computer Aided Design

DC – Direct current

LSB - east significant bit

SCLK – serial clock

CS - chip select

MISO - main input/subnode output

MOSI - main input/subnode output

LRV – lunar roving vehicle

CAM – computer aided manufacturing

SMD - surface mount device

INTRODUCTION

Tartu Observatory is the most distinguished space centre in Estonia. Their primary goal is to promote the science of astronomy and space technology and to train young scientists and engineers. This is accomplished through projects, such as KuupKulgur, EstCube 1 & 2, OPIC and other such projects [1], which pioneer outer space research.

This thesis will be covering a part of the KuupKulgur project. The mission of the KuupKulgur project is to develop and deliver the first Estonian lunar rover to the moon. The objective of this rover is to be a relatively low-cost and modular platform for various payloads which collect data from the surface of the moon.

The aim of this thesis is to measure the battery cell voltage, current, and power consumption of the rover. To achieve this, it is needed to write drivers for the ADC (Analog Digital Converter) integrated circuits on the EPS (Electrical Power System) circuit board which translates the raw signals into human readable data and to display the gained data within a simple UI (user interface).

The data from the ADC will be mainly diagnostic data, such as battery cell voltage, temperature, current, and power efficiency.

The code for the drivers is written in the C++ programming language using Visual Studio Code IDE (Integrated Development Environment).

This thesis consists of three chapters, the first describes the problem statement, methods used, and the requirements for the work. In the second chapter an overview of the theoretical part is given as well as an overview of the hardware used. In the third chapter the code is analysed on how it works, what it does and the results that are produced with it.

1 PROBLEM STATEMENT

An essential part of developing and testing any remote or autonomous system, such as Kuupkulgur, is to gather data about the current status of the lunar rover such as KuupKulgur and report when a failure occurs. In case of a failure in the EPS or any other part of the electronic stack the raw data translated by the sensors on board, such as ADC-s can be used to diagnose the problem, fix it or minimize the damage. This data is also useful to keep track of power consumption and the state of the battery pack - its overall health, and the battery percentage. It is also used for the digital twin of the KuupKulgur in a simulated environment developed in the observatory.

Having gained the electrical diagnostic data from KuupKulgur it is needed to display the results within an UI along with data from other systems such as the motor drivers, communication modules etc. This UI will help the KuupKulgur engineers monitor the health and status of the KuupKulgur. Therefore, help with the continued development, eventual deployment and the following operations.

2 Theory and technologies used

2.1.1 Drivers for hardware

In computer engineering a driver is software with the primary function of translating commands from operating systems into commands that the hardware can understand. The driver handles operations such as initialization, data transfer, and error handling. Drivers are usually device specific, meaning each device usually requires a separate driver [2].

2.1.2 Integrated circuit

Integrated circuit (IC) is a single assembly unit of miniaturised electronic components such as transistors, diodes, capacitors, resistors etc. Their connections are made up of microscopic semiconductor material, usually silicon. The resulting circuit is a small monolithic chip, whose sizes typically range from a few square centimetres to a few square millimetres. The most basic types of IC-s are analog and digital circuits [3].

Analog circuits typically consist of only a few components, making them the least complicated type of ICs. In the general case analog circuits are connected to devices which acquire continually changing signals from the environment. For example, a microphone converts sound waves into an electrical signal, that signal can be then amplified and filtered by an analog circuit [3].

Digital circuits handle discrete binary signal values which fluctuate between logical 0 and 1 (signal low or high). A discrete signal has a predetermined step size, with each step there is a corresponding binary value. Digital ICs mainly consist of diodes, microprocessors, and transistors. In modern digital ICs there are billions or even trillions of transistors within a single chip. Digital integrated circuits are the basis of all digital computing, they are also used to process digital information without the need to connect to a computer. Some examples include timers, calculators, counters, and microprocessors [4].

2.1.3 ADC introduction and use cases

An Analog-to-Digital Converter (ADC) is an IC that takes an analog signal and converts it into a digital signal, making it possible to process them by digital systems.

An ADC takes the continuous input voltage and samples it at defined regular intervals, then it converts each sample into a binary value according to the magnitude of the input signal shown in figure 1. The conversion is accomplished by quantization, it involves dividing the continuous range of analog values into discrete levels relative to a constant direct current (DC), called reference voltage (Va in figure 1), then each level is represented by a corresponding digital code [5]. Each discrete level is separated by 1 LSB (least significant bit), which is the smallest change in input an ADC can pick up. The width of 1 LSB for a given device is determined by the relation between the reference voltage and the number of possible binary values for one measurement of the device. The number of possible binary values for a measurement is determined by 2 to the power of the bit resolution of the device. For an example in figure 1. a 10-bit device is shown, meaning its number of possible binary values is $2^{10} = 1024$.

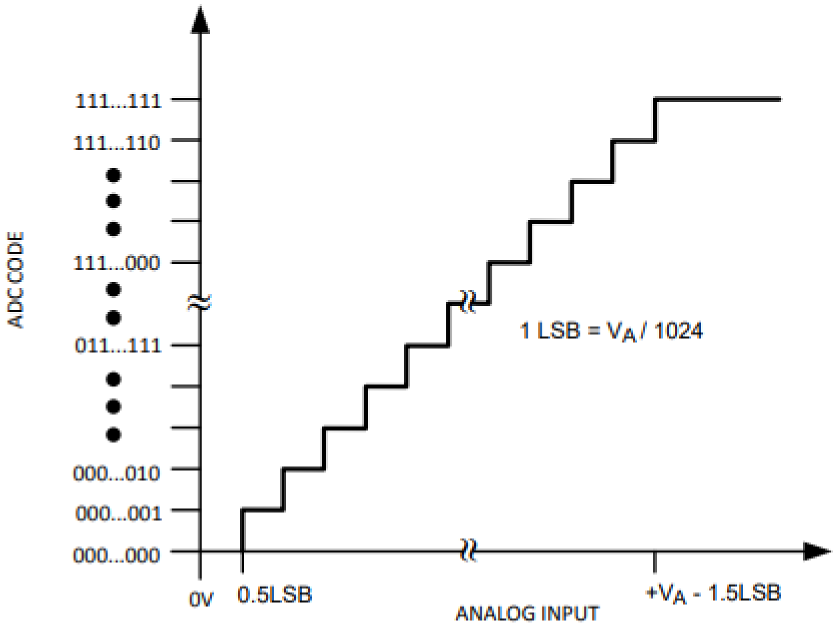

Figure 1. ideal transfer Characteristic of ADC108S022 [6]

ADCs are used in various industries, including telecommunications, medical instrumentation, industrial automation, space technology, and consumer electronics. ADCs are essential in the digitization of various signals [5] such as audio signals, sensor measurements, and environmental data to name a few.

2.1.4 Serial peripheral interface (SPI) introduction

The Serial Peripheral Interface (SPI) is a synchronous serial communication protocol used for connecting peripheral components in embedded systems, such as microcontrollers, sensors, memory devices. SPI provides high-speed, full-duplex communication between a main device and one or more subnode devices over short distances, usually within the same device.

Normally SPI has four signal lines: **serial clock (SCLK)** main sets the rate at which the data is sampled from the subnode, **chip select (CS)** which enables the data communication with the given SPI device, **main input/subnode output** (**MISO**) transmits data from the subnode to the main device, and **main output/subnode in (MOSI)** transmits data from the main device to the subnode device [7].

As briefly mentioned in the previous paragraph, SPI operates in a main-subnode configuration, where the main device initiates communication with a subnode by sending it a clock signal and pulling its CS low. The subnode device then records one bit of binary data for each clock pulse, the data is then sent to the output as a binary value. The number of bits transferred for each binary value depends on the resolution of the subnode device [8].

SPI can use various data transfer modes, from mode 0 to mode 3. The modes differ in clock polarity (CPOL) and clock phase (CPHA). They specify at which conditions of clock and phase a bit of data is sampled (table 1). For mode 0 data sampled on rising edge and shifted out on the falling edge, mode 1 data sampled on the falling edge and shifted out on the rising edge, mode 2 data sampled on the falling edge and shifted out on the rising edge, mode 3 data sampled on the rising edge and shifted out on the falling edge [9].

Table 1. Table of SPI modes based on [7] and [9]

SPI mode	CPOL		CPHA		Clock Polarity in Idle State
0	0		0		Logic low
1	0		1		Logic low
2	1		0		Logic high
3	1		1		Logic high

SPI is used due to its simplicity, cost effectiveness and efficiency in sending data, it is suitable for applications requiring low latency data transfer. Additionally, SPI is highly versatile, meaning it can support a wide range of peripheral devices with minimal hardware overhead [8]. It is because of these properties that SPI has become a standard communication interface in many microcontrollers and integrated circuits (IC-s).

2.1.5 Breadboard

A breadboard is a platform that is used to build test circuits. It consists of a rectangular plastic body with a square grid of small holes, which are designed to hold and connect electric components. The grid is spaced to accommodate standard wire gauges and component leads. Typically, the distance between each hole is 2.54 mm (0.1 inch) [10].

Breadboards usually consist of two sections, which are terminal strips and bus strips. Terminal strips are located in the middle of the board, usually between two bus strips. The terminal strips consist of columns of interconnected holes, typically each column contains five holes. Bus strips are located at the top and/or bottom of the breadboard. Each bus strip contains two interconnected rows, the top row is generally used for power input and the bottom row is for ground [11].

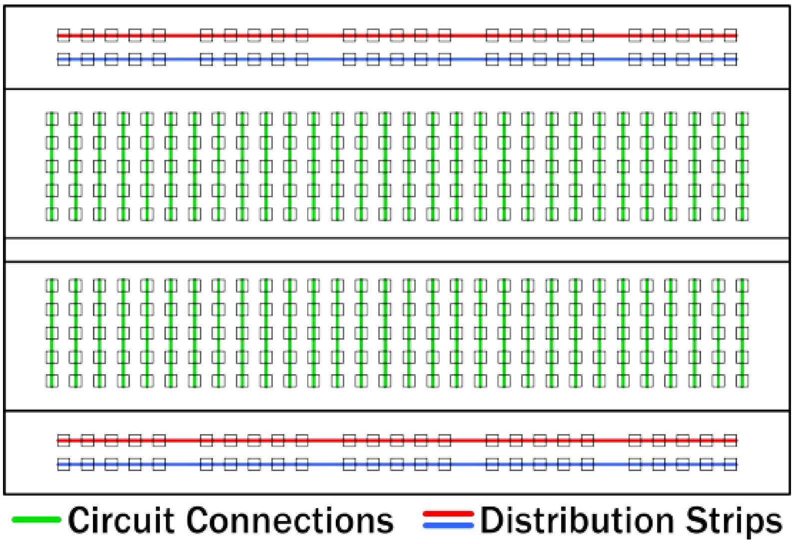

Figure 2. Breadboard connections [11]

The breadboard is constructed in such a way to make it easy to add and remove components to the schematic. To add a component or a wire it is simply needed to insert its leads into the holes, no soldering needed. The easy nature of inserting and removing components makes the breadboard a versatile tool for prototyping circuits.

2.1.6 Printed circuit board

A printed circuit board (PCB) is an electronic assembly which essentially consists of copper conductors and electrical insulators. The conductors provide connections between the components soldered onto the PCB and the insulators provide the separation between those connections. PCBs also provide mechanical support to the electronic components by allowing the device to be mounted in an enclosure [12].

PCBs are assembled from alternating layers of conductive copper and electrical insulators. The conductive details printed on PCBs include copper traces, tin/gold pads, vias, and conductive planes. The pads are at the top layer of the PCB to provide an area for the components to be soldered onto, they are covered with tin or gold to protect the copper from corrosion. Vias are drilled holes with copper walls within the PCB that provide conductivity between copper layers. The walls of the drilled hole are covered in the same fashion as the pads. The insulators also provide mechanical rigidity by enclosing the layers of conductors within themselves. The top layer is made of a non-conductive solder mask, and on top of which is printed a silkscreen to provide a legend for electronic components. Functionally it is not critical, however the solder mask provides long term protection for the top copper layer from the elements and accidental shorting of the components. When the aforementioned fabrication steps are completed, the board is ready for the components to be soldered onto the board. After the components are soldered onto the board the PCB assembly (PCBA) can be tested [12].

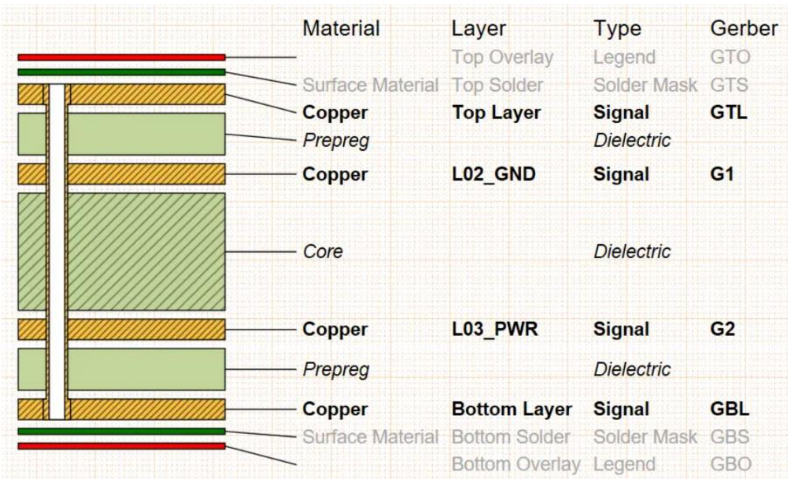

Figure 3. Layers of a multilayer PCB [12]

PCBs vary in complexity with the number of conductive layers within the board. Single-sided boards have components mounted on one side of the board, with the bottom surface typically being fully copper to provide a large ground plane. The connection between the ground plane and the components is provided by vias. Double-sided circuit boards have components on both sides of the PCB and the layers may be interconnected by vias if necessary. Multi-layer PCBs have internal copper layers in addition to the top and bottom, as can be seen on layers figure 3. These internal layers are used to carry electrical signals between components, or they are used as conductive plane layers [12].

2.1.7 Prototype PCB production

Saving development time is critical when developing new electrical systems. This creates a necessity for ways to quickly test developed electrical circuits. Simpler circuits can be tested using a breadboard or other such prototyping solutions. However, when the circuits become more complicated it becomes impractical to test them on simpler solutions. Especially if surface mount device (SMD) components are used. Installing SMDs on a normal prototype board is simply impractical. As seen on figure 4, it becomes very messy very fast and is time consuming for larger circuits. Meaning there is a need to look for an alternative solution.

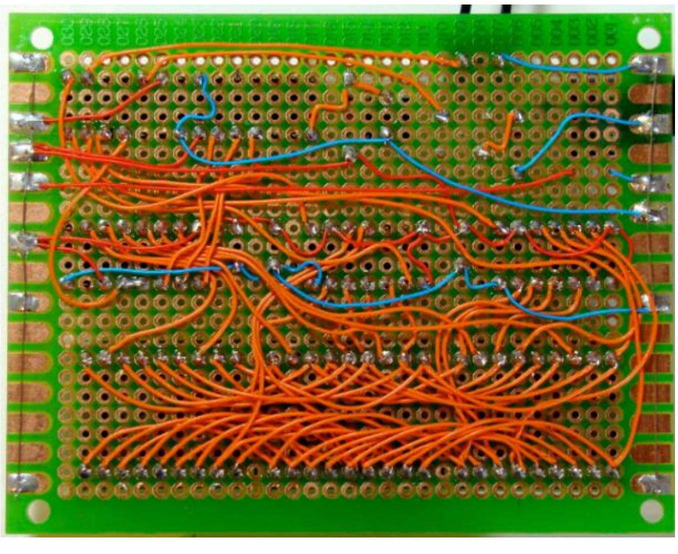

Figure 4. Example of a prototype board created by hand [13]

It is possible to order full prototype boards from producers such as Aisler or JLCPCB. However these solutions have a minimum three days of lead time, which can be extended by issues in logistics and can be costly depending on the complexity of the PCB. Meaning when rapid prototyping is required it can be advantageous to produce the prototype PCB in house.

For in house prototype PCB production there exists many computer aided manufacturing (CAM) solutions. These solutions allow for a time efficient and reliable way to produce prototype PCBs. With the solution available in house the lead time is less than a day in most cases making in house production cost and time efficient [14]. With simpler circuits it's possible to have multiple revisions in a single day. These solutions use a blank premade two-layer PCB material consisting of two copper layers and an insulating layer. Which is then etched and drilled according to the PCB layout provided. In most cases the CAM software included creates paths for the milling and drilling routes automatically. One of such mills is LPKF ProtoMat 44 [15].

2.1.8 Breakout boards

Breakout boards are PCBs which provide an easy way to interface with an IC, sensor or module by providing a simple way to connect external electronics and signal/data lines to specific pins. The easy access is achieved by a separate circuit board, example shown in figure 5, where the pins of the device are broken out into pin headers, or other easily accessible terminals, which are usually labelled [16]. Usually, the pin headers have a pitch with of 2.54 mm for ease of us in tandem with a breadboard. This makes breakout boards perfect for prototyping circuits and debugging software written for the device.

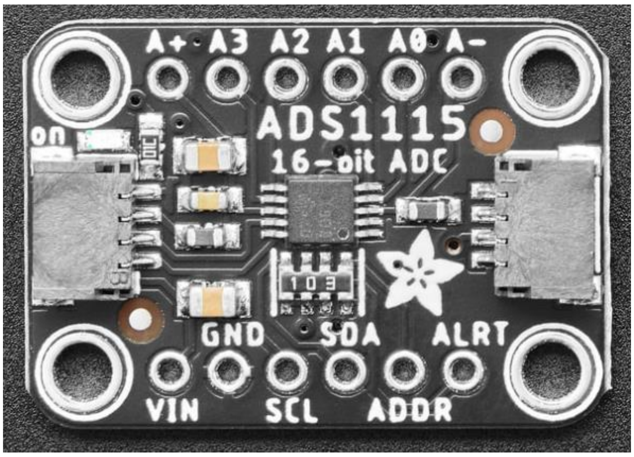

Figure 5. Example of a breakout board [17]

2.1.9 Logic analysers

A logic analyser is a device used to capture, display, and measure multiple electronic signals at the same time in a digital circuit. It can show the relationship and timing between various signals in a digital or analog system. Digital logic analysers often have the capability to analyse and record digital communication protocols like I2C, SPI, and Serial. To record data logic analysers come with software, which provides a concise UI to navigate through large amounts of signal data. This UI also displays information such as pulse width, frequency, and period of periodic signals, as well as duty cycle among other signal parameters. Most modern communication systems being implemented are based on a set of standard protocols. Logic analysers also have the ability to identify and then decode signals using these standard protocols. These features make digital logic analysers one of the most effective tools for debugging digital circuits and communication systems [18].

In 1973, Hewlett Packard introduced the first logic analyser [19], capable of measuring and displaying logic states using a set of LEDs. The HP 5000A, was the first logic analyser available on the market but it was limited to just two channels. In subsequent years, commercial logic analysers evolved to feature dozens of channels, allowing them to read and display digital logic in parallel.

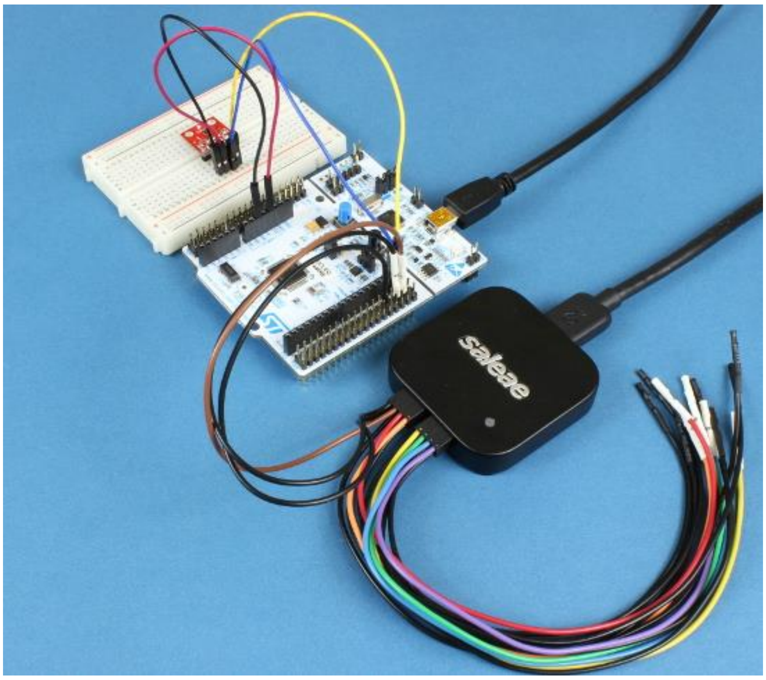

Figure 5. PC-based logic analyser connected to a circuit [18]

Modern logic analysers have the advantageous capability to monitor and record a large number of digital signals simultaneously, in most cases 8 to 100 channels at once [18]. They have the ability to begin recording at a trigger event defined by the user. These trigger events can be configured to trigger taking into account states of multiple channels. Modern logic analysers also have the ability to record and analyse analogue signals, logic analysers with this capability are known as mixed-signal oscilloscopes.

PC-based logic analysers utilise computers to handle the intensive tasks of displaying and analysing captured data. These analysers consist of a separate device that connects to one of the computer's accessory ports. This device contains the necessary data acquisition circuitry needed to capture multiple high-speed digital signals. The captured data is then transmitted to computer software through the connected port. Given the need for fast transmission of large amounts of data, most PC-based logic analysers use high-speed wired connections like USB or Ethernet [18].

2.1.10 Lunar rovers

Lunar rovers, also known as moon rovers, are specialized robots and vehicles designed for exploring the surface of the Moon. These vehicles have a crucial role in lunar exploration missions, enabling scientists and astronauts to conduct scientific experiments, gather data and explore the moon's landscape without human presence [20].

The first lunar rover, named the Lunar Roving Vehicle (LRV), was deployed during the Apollo 15 mission in 1971. To be able to traverse the surface of the moon the LRV was designed to be able to operate in low-gravity and vacuum. The electric LRV vehicle allowed the Apollo mission astronauts to reach further away from their landing spots to perform their surface extravehicular activities. The LRV had a mass of 210 kg and was designed with an additional payload of 490 kg in mind. The frame was 3.1 metres in length with a wheelbase of 2.3 metres. The highest point of the LRV was 1.14 metres from the base of the wheels. The frame was constructed with an aluminium alloy tubing assembled by welding them together, consisting of a 3-part chassis which could be folded, thanks to the hinged centre and hung in the Lunar Module quad 1. The two side-by-side seats were also foldable and made of tubular aluminium with nylon webbing and aluminium floor panels. For each seat there were adjustable footrests and a velcro seat belt. In the front centre of the rover a large mesh dish antenna was mounted to a mast, as can be seen in figure 5. A fully loaded LRV had a ground clearance of 36 cm [21].

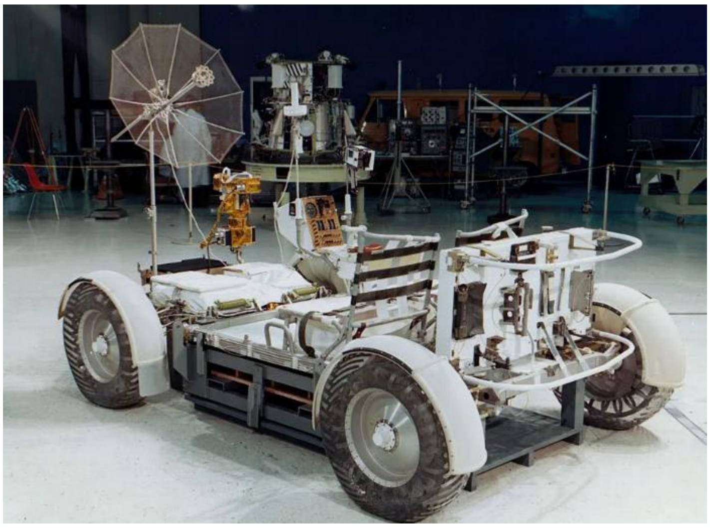

Figure 5. LRV 1 before being shipped to KSC [22]

Since the Apollo era, lunar exploration has continued to grow and develop, leading to the development of new lunar rover designs and concepts. One of these concepts is cube rovers, inspired by the success of CubeSats, they follow the same 20x10x10 cm payload standard as CubeSats. By being able to be constructed from off-the-shelf components Cube rovers allow for a modular and relatively cheap platform for lunar exploration and research [23]. The compact size also makes them scalable to larger quantities.

2.1.11 KuupKulgur

KuupKulgur, seen on figure 6, is a cube lunar rover being developed by the Tartu Observatory Space Exploration group.

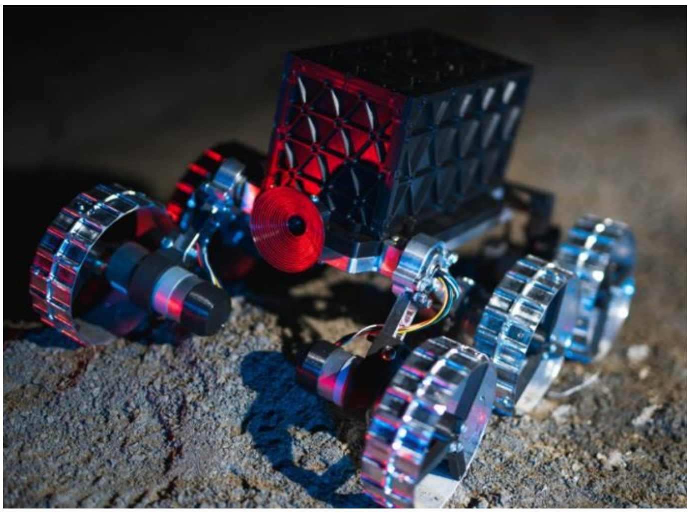

Figure 6. KuupKulgur Payload Demonstrator Model [24]

KuupKulgur consists of two main parts: the payload and the base, which carries the payload. The electronical components of KuupKulgur will be housed within the base. The subsystems are as follows: flight controller module, avionics systems, motor drivers, batteries etc. Onto the base platform it will be possible to install the payload module. This form factor allows for a modular design approach [24].

The objective of this rover is to be a low-cost and modular platform for various CubeSat standard sized payloads, which collect scientific data and perform scientific experiments on the surface of the Moon. These payloads will be developed by third parties, having access to a standardised lunar rover platform these third parties do not need to develop their own rover. Therefore, making the process of conducting scientific research on the moon more accessible to the world and through that accelerating space exploration and research.

3 Development

3.1.1 Gathering information from the data sheet

To write a driver for any IC it is necessary to read the datasheet of the device first, as it contains crucial information about the chip. The datasheet [6] for the ADC being used, model nr. ADC108S022, contains information important to programming the driver, such as the minimal and maximum clock frequency, timing diagrams, control register bits, input channel selection ID-s, SPI conversion process breakdown, and the conversion formula which is used turn the binary output into human readable data.

3.1.2 Writing the code

The basis for the written code was the Spidev library developed by Andrea Paterniani [25]. The library contains the necessary code for system calls to support communication with SPI devices within Linux.

To use any SPI device, it is required to configure it in such a way that it matches with the written code that will later utilize the SPI device. The initialization of the SPI utilizes Spidev-s ioctl, which is a system call to modify parameters within special files. In this case ioctl modifies parameters related to SPI. During initialization clock frequency, SPI mode, and the sampled ADC is set. The clock frequency was set to 500 kHz, SPI mode to zero, and the battery ADC was selected for testing.

To get data from all the ADC input pins at the same time the developed transfer function is used. The SPI transfer function utilizes SPI-s ability to send and receive data at the same time. The function activates an ADC according to the ID given to the function, by pulling CS low on the given ADC the data transmission begins. The amount of bytes input and output of the transfer is specified by an integer value input to the function. The function also requires an input of control register IDs on which the measurements will be performed. With the aforementioned input the transfer function then utilizes ioctl, which in reality performs the measurements by communicating with the core of the operating system. The results of these measurements are saved within an array of 8-bit unsigned integers. To make the transfer function work it was important to determine the precise length of the arrays input to the function.

By taking a look at the ADC108S022-s serial timing diagram see on figure 7, it can be seen that the output data is sent in sixteen clock cycles or in this case one serial frame, meaning 16 bits are sent to the output with the most significant bit (MSB) first. As each byte is eight bits, the output can be divided into two bytes. With four leading zeros and two trailing zeros it leaves 10 bits of actual output data.

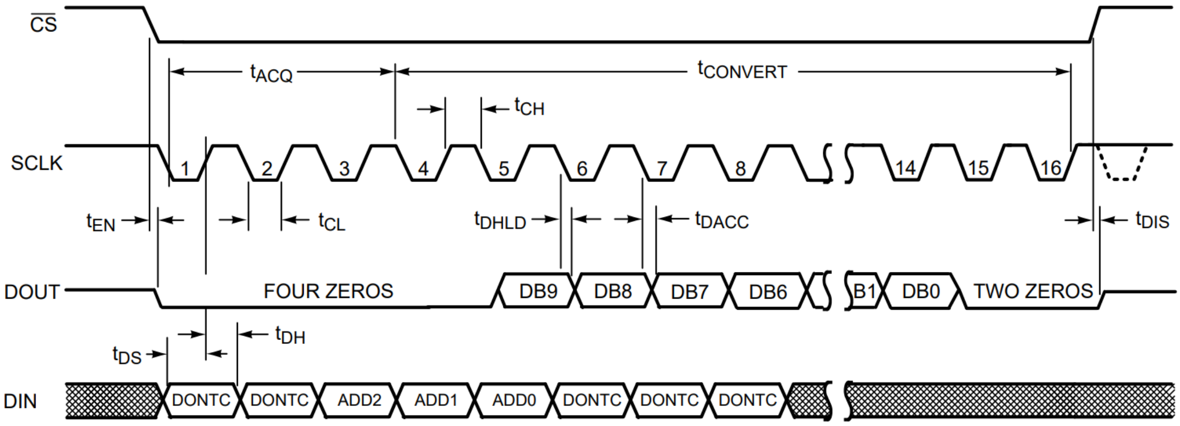

Figure 7. ADC108S022 Operational Timing Diagram [6]

During the same sixteen clock cycles the control register is also sent as an input to the ADC, which determines which input data pin is sampled to the output in the next serial frame. The input channel control registers are simple binary values, translated to decimal values they match the pin numbers 0 - 7. According to the control register bits table in the datasheet (table 2) the binary value needs to be shifted three bits to the left to match the given control register format.

Table 2. ADC108S022 control register bits and channel ID-s [6]

Bit 7 (MSB)	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DONTC	**DONTC**	**ADD2**	**ADD1**	**ADD0**	**DONTC**	**DONTC**	**DONTC**

ADD2	ADD1	ADD0	Input Channel
0	0	0	IN0 (Default)
0	0	1	IN1
0	1	0	IN2
0	1	1	IN3
1	0	0	IN4
1	0	1	IN5
1	1	0	IN6
1	1	1	IN7

The output for one pin measurement is sixteen bits, which is divided into two bytes during the conversion process, meaning the data bits are divided into those two bytes, as can be seen from the timing diagram (figure 7). The byte pairs are saved as eight-bit unsigned integers, which perfectly contain the eight bits of output. To get the real raw value of the measurement it is needed to combine the two bytes into a single binary value using the logical bitwise OR operation. However, first the bits for each byte need to be shifted. The first byte must be shifted by six bits to the left and the second byte two bits to the right. To avoid data loss the first byte and the result is treated as a sixteen-bit unsigned integer, meaning leading zeros are added to fill the missing bits. When the shifted values are combined in this fashion a ten-bit binary value is achieved, which is the raw value of the measurement.

To better illustrate the process in the previous paragraph let us define the single measurement sixteen-bit output as an array of O-s (don't care bits) and X-s (data bits). And then divide the sixteen-bit output into two bytes B1 and B2. Using these definitions the table shown in figure 9 was made.

Table 3. Bitwise operations performed to gain the raw value from a measurement

Nr	Operation	Byte pair		Result
		B1	**B2**	
0	Define data bits and dontC bits	0000 XXXX	XXXX XX00	0000 XXXX XXXX XX00
1	Shift B1 six bits to the left	0000 XXXX	-	0000 00XX XX00 0000
2	Shift B2 two bits to the right	-	XXXX XX00	00XX XXXX
3	Applying bitwise OR (B1 | B2)	0000 00XX XX00 0000	00XX XXXX	0000 00XX XXXX XXXX

3.1.3 Debug process

When attempting to run the program for the first time to test the written code, calling the ioctl function crashed the program due to ioctl not being able to find the Spidev framework. However, the author of this thesis observed that Spidev existed on the given system. Consulting the system engineer it was found that every time the KuupKulgur was rebooted it was needed to run the modprobe command within the terminal. The modprobe command manually adds modules to the core of the operating system of Linux. Having done so the program ran, and the raw data was gained from the pins.

Having gained the raw data from all the input pins using the SPI transfer function it is needed to translate the raw voltage values into real voltage values. This is achieved by using the transfer formula given in the datasheet, also shown in figure 1. Using this formula to translate the results it was clear that the results were inaccurate and seemingly random values. This was also confirmed by manually measuring the voltage at the input pins with a multimeter.

The first step in the debug process was to take a look at the code. This was done by using VS codes debug extension. The extension allowed for break points within the code, where the program is paused at a specific line of code. A simple mistake was found straight away. That mistake was that the input for the next register to be sampled was faulty. More specifically, for the input an array of register id-s was defined as 8-bit integers as such: {0x08, 0x10, 0x18, 0x20, 0x28, 0x30, 0x38}. Pin zero was missing from the input because pin zero is always the first pin sampled when CS is pulled low, therefore pin zero shouldn't be included within the input to avoid measuring it twice. However, the input expected 16 bits of data, meaning the array ran out midway through the transfer function. This caused an overflow, which in turn caused the seemingly random values. The fix was simple, just add zeros to the input after each register id. The input array was also expected to be the same length as the output array, the missing length was also filled out by zeros, giving us a new input array: {0x08, 0x00, 0x10, 0x00, 0x18, 0x00, 0x20, 0x00, 0x28, 0x00, 0x30, 0x00, 0x38, 0x00, 0x00, 0x00}.

While reading the datasheet it was noticed that the absolute minimum clock frequency found in page 3 of the datasheet under "*Operating Rating*" is 50 kHz [6], which was initially taken into account to set the clock frequency to 500 kHz during initialization, differed with the value given in the same datasheet on page 5 "*ADC108S022 Converter Electrical Characteristics*". There it was stated that the minimum clock frequency is 800 kHz. Due to this discovery the clock frequency was set to 1.2 MHz going forward.

With the previously mentioned changes the code still did not function as intended, the results were still faulty. It was time to take a closer look at the PCB the ADC was on to find out if there is a fault within the PCB design. When comparing the typical application circuit found within the datasheet, seen in figure 10, with the circuit on the EPS it was found that they were different. The theory was that the low pass filters were too strong for the inputs of the ADC. To test this theory, it was decided to locally produce a breakout board using the LPKF ProtoMat 44 [15], which was available at Tartu Observatory, to test the ADC separately from the rest of the electronics.

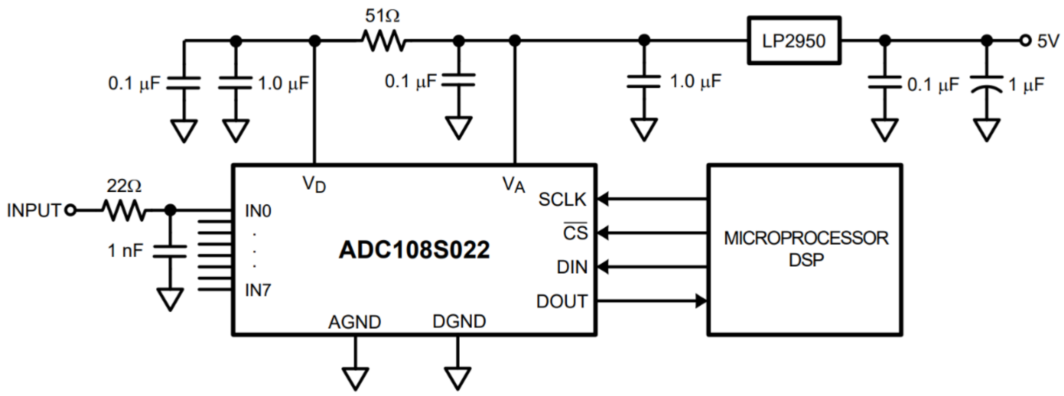

Figure 10. ADC108S022 Typical Application Circuit [6]

The breakout board was projected using KiCad using the same PCB template as the rest of the electronics from KuupKulgur. The only change needed in the template was to change the width of the copper track to 2 mm, increase the size of the vias and increase the clearances of the tracks to take into account the ProtoMat E44 circuit board cutters tolerances. On the schematic the order of the pin header pins was changed to make the routing as simple as possible. After the necessary changes, the new schematic was routed within KiCad (figure 11) and the files were exported to a format acceptable for the LPKF CircuitPro software.

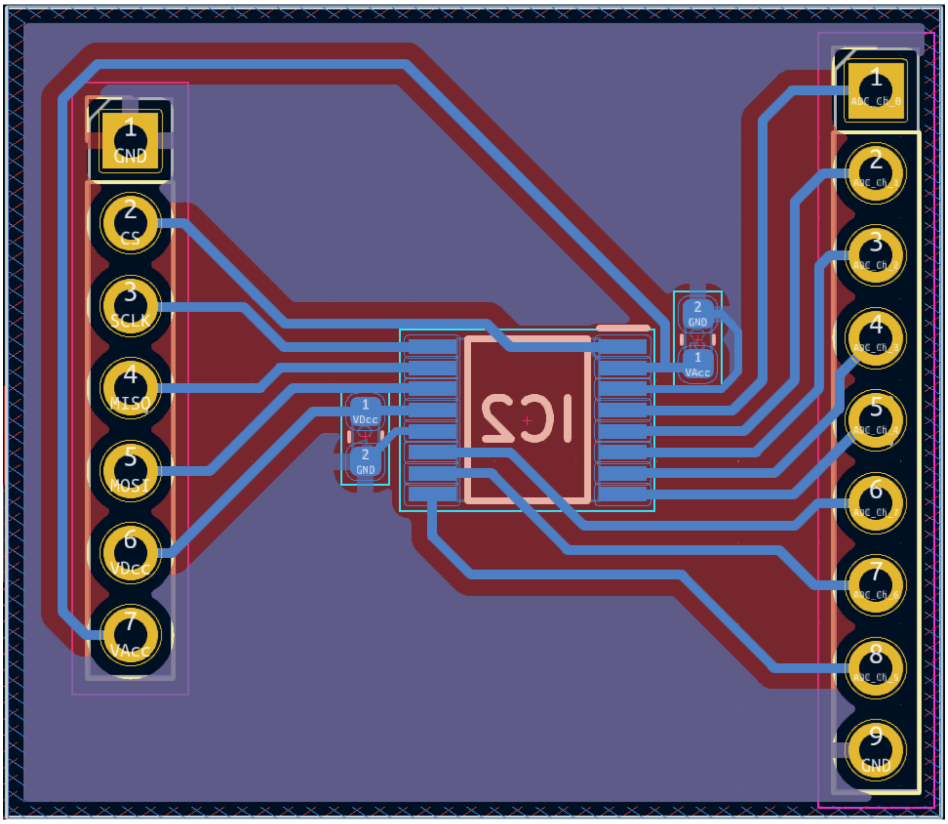

Figure 11. Projected breakout board layout in KiCad

The next step was to have the ProtoMat E44 cut out the projected PCB (figure 12). The process was relatively simple. The necessary breakout board CAD files were imported to the LPKF CircuitPro software. An area was defined in which the board will be cut from, CircuitPro required reference points to be defined outside of the defined PCB area, which were then defined. After the settings for the milling and cutting the PCB were set in accordance with the template of instructions provided by the system engineer. The production of the breakout board could then begin. The circuit board plotters software automatically planned the route for the milling process. The only things left to do was to manually change the endmills when the board plotter requested it and rotate the board after one side was done. When cutting of the board was complete the components were soldered onto the breakout board. Additionally, a 1 nF capacitor was added to one of the analog input channel 2 (AIN2). This was done to gain more accurate measurements during the debug process.

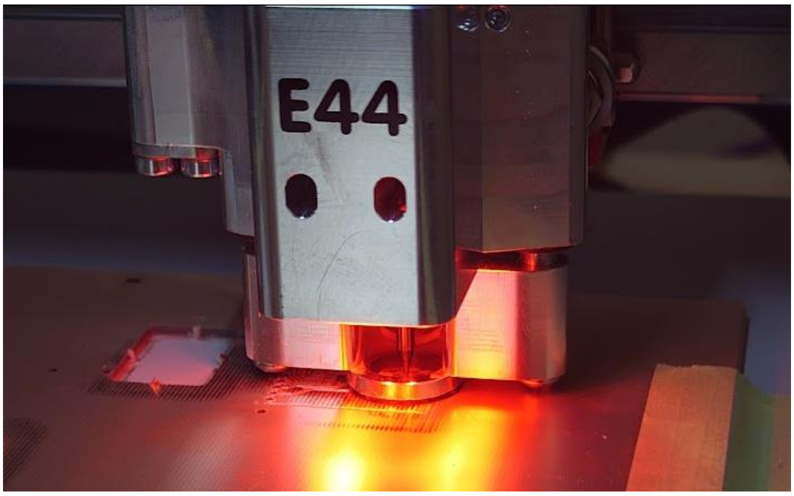

Figure 12. Projected breakout board being cut by ProtoMat E44

With the breakout board ready it was time to test the ADC with it. The breakout board was connected to the KuupKulgur with jumper wires and a breadboard. A Saleae logic 8-channel logic analyser [26] was also connected to the system. For the input a variable power supply was used. The aforementioned connections can be seen on figure 13. The logic analyser allowed the capturing and saving of the ADC input and output communication.

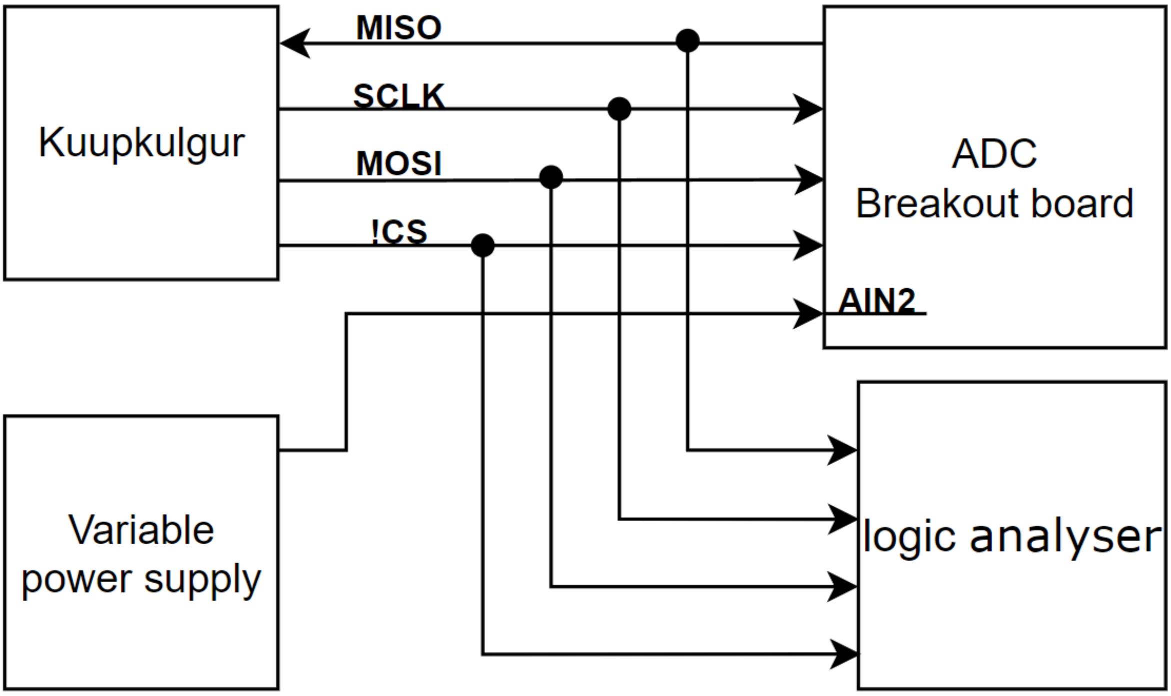

Figure 13. UML diagram of the circuit used for debugging

With the logic analyser it was seen that the SPI data was sampled on the falling edge and shifted out on the rising edge of the clock signal. Meaning SPI was initialized using mode 1 by mistake. This was easily fixed by changing the mode initialized to mode 0 as intended.

Another mistake was found within the transfer function. There was logic within the transfer function which set TX bits and RX bits to a non-zero value. Which should only be changed when the SPI operates with a dual or quad interface. Dual and quad interfaces have two and four datalines respectively, however the given ADC operates with a single data line. Meaning data loss or corruption is caused if the TX and RX bits are changed. Simply removing the logic changing the TX and RX bits solved this issue.

A quirk of the given ADC was found using the logic analyser when inspecting the raw output bytes of the ADC. According to the datasheet of the given ADC the leading bits of the first byte were supposed to be always zeros. However, in reality this was not the case, the bits had random values. When the leading bits by chance were nonzero it caused the translated value to be much larger than expected. This quirk of the ADC was handled by masking the first byte of the output before shifting the byte with 0x0f (00001111 in binary). Meaning a logical bitwise AND operation was performed between the first byte of each output frame and 0x0f. This operation assured the first four bits of each output frame were zeros as expected. The result of this operation is visible within the logic analyser output shown in figure 14.

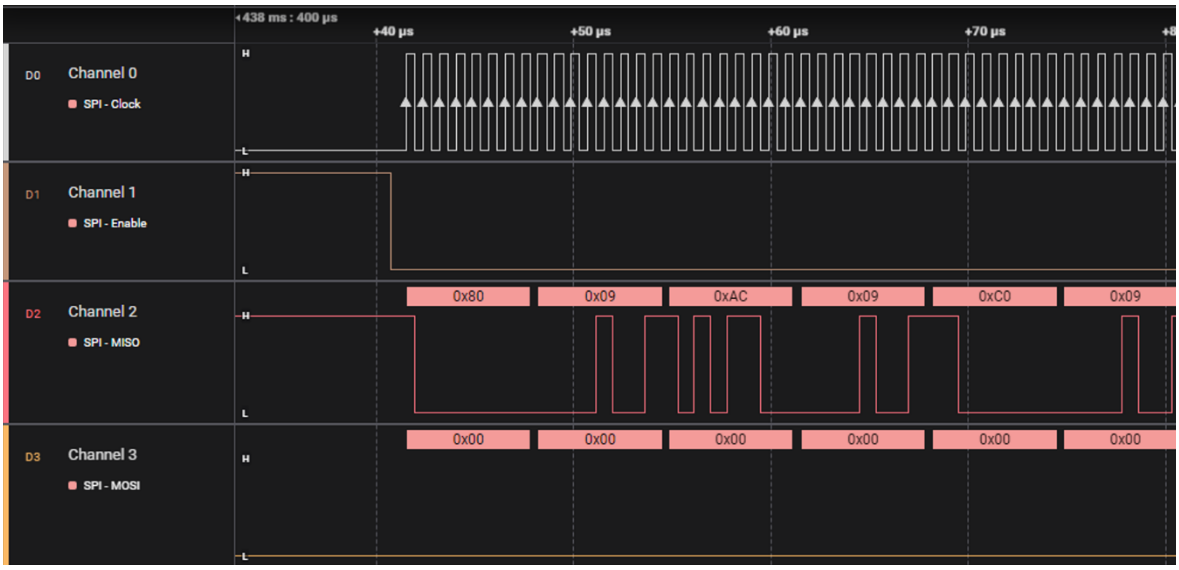

Figure 14. Logic analyser output while measuring output channel 0

While projecting the breakout board the author of this thesis did not take into account that the drilled holes do not act as vias, that would connect the ground planes when using the ProtoMat E44. Meaning the inputs and output headers had separate grounds, this mistake is also visible in figure 11. Before performing the rest of the measurements for the analysis of ADC the logic analyser was disconnected, as it was no longer needed, after which the results became inaccurate again. This happened because the logic analyser provided a common ground between the inputs and outputs. This mistake initially caused large inaccuracies when sampling and then translating the input signal using the breakout board. However, when the inaccuracy was investigated the mistake was discovered and then promptly corrected with a soldered jumper wire between the ground planes.

3.1.4 Conclusions from the debug process

After the changes mentioned in the previous chapter the ADC started to consistently output measurements with relatively small inaccuracies. However, it was noticed that as the input voltage rose the inaccuracy of the measurement increased dramatically. Using the variable power supply to apply different voltages it became clear that the increase in error was not linear.

To determine the error rate across the voltage range being measured within the KuupKulgur it was decided to make a graph with test voltages ranging from 0 to 5 volts. To achieve this the variable power supply was used to increase the voltage by 10 mV, which is the lowest increment possible on the given power supply. With each increment the average raw value of 40 consecutive measurements was recorded to a separate file by the written code. The average raw values were then input to a google sheets table which calculated the voltage using the transfer formula from the datasheet. With the input voltage known thanks to the variable power supply, the author was able to identify the error of the measurement. This was done by simply subtracting the expected voltage from the gained measurement. Using the data gained by this process the graph shown in figure 15 was made. The graph shows the input voltage and then the added voltage by the error.

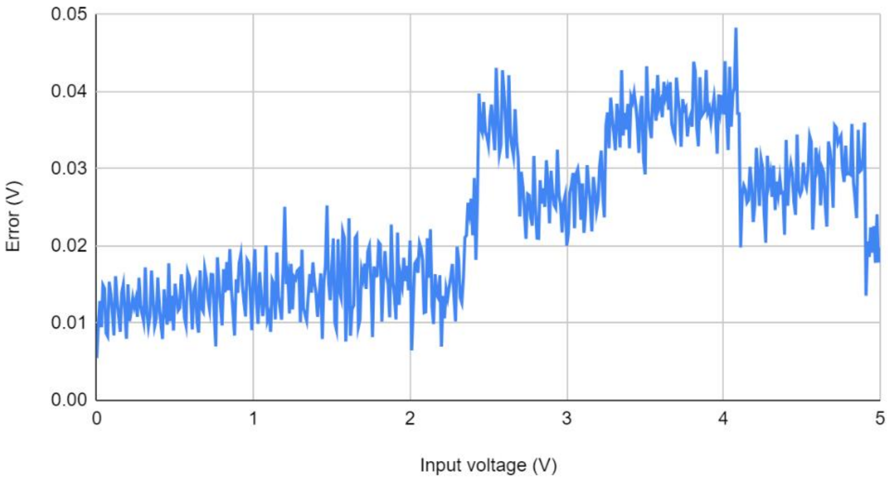

Figure 15. Error voltage vs. Input voltage graph of ADC108S022

With the error rate known it will be possible to take it into account when future measurements are performed. However, the error was too large at higher voltages to justify using it in future prototypes.

3.1.5 Market research for a more suitable ADC

To determine the next ADC to be used for the next prototype of KuupKulgur market research was conducted based on the required parameters. The requirements for the ADC were as at least 10-bit resolution, 8 analog input channels, and a measurement range of at least up to 5 V.

With many different options considered the author of this thesis recommended the purchase and further testing of three additional ADC-s shown in the table below (table 4).

Table 4. 8-channel ADC-s recommended for further development and their main parameters

	Names and links		Maximum operation ratings			
Nr.	Mfr. Part Nr.	Datasheet	Va [V]	I_{DCL}	sample rate [ksps]	Bits
0	ADC128S102	[27]	5.25	±1 μA	1000	12
1	ADS8344N	[28]	5.25	±1 μA	100	16
2	MCP3008T	[29]	5.5	±1 μA	200	10

3.1.6 Testing the recommended ADC-s

The first step in development for the new ADC-s was to make new breakout boards for the chosen ADC-s. Only minor changes were required from the previous breakout board. The changes included changing the IC to the desired ADC, adjusting the RC-filter resistors and capacitors for the inputs and reordering the header pins if needed. After these changes the PCB-s were ordered, assembled and then tested.

The driver written for ADC108S022 [6] was adjusted to work with the new ADC based on the datasheets of each device. The main thing that needed to be changed was how the input data was shifted and translated. Additional changes included changing the frequency of the clock speed and pin id adjustments in accordance with the datasheet for the given device. After the aforementioned changes and soldering the PCB, the testing could begin. The process for testing was similar to the process used for ADC108S022 in chapter 3.1.4, but with a different power supply DP832A [30] as the previously used power supply was no longer available. While using the DP832A power supply there was a noticeable difference between the set voltage and the actual output of 0.000 to 0.01 V. To mitigate inaccuracies caused by this output error the actual output of the power supply was used in the error calculations.

3.1.7 Results from testing recommended ADC-s

MCP3008T testing measurements (figure 16) show a noisy output and large inaccuracies within the whole measurement range.

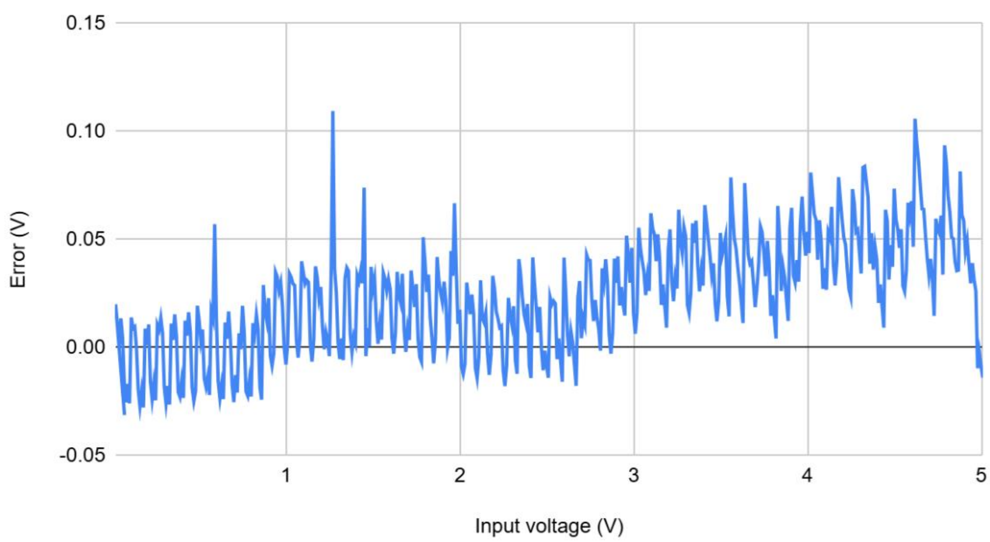

Figure 16. Error voltage vs. Input voltage graph of MCP3008T

ADC128S102 showed a drastically lower error rate and output noise than other tested ADC-s (figure 17). Additionally, the output was almost always lower than the real input. The low noise, low error rate, and the measurement being consistently below the real input makes error more predictable. Therefore, the error correction more straightforward.

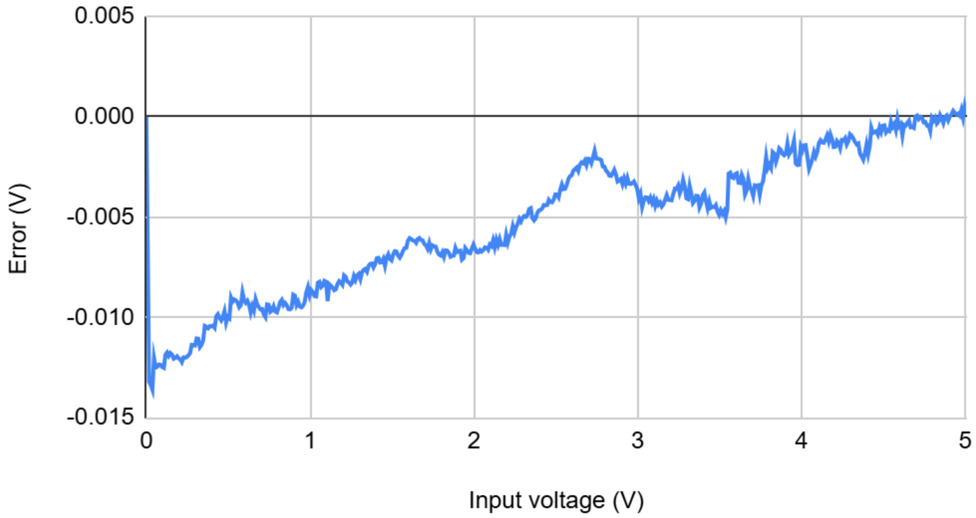

Figure 17. Error voltage vs. Input voltage graph of ADC128S102

3.1.8 Recommendation for a new ADC

Based on market research and testing of the chosen ADC-s the author of this thesis recommended ADC128S102 [27]. This recommendation was made based on the required properties for the new ADC. The recommended ADC has a higher resolution of 12 bits instead of 10 bits, which made it more accurate and a more predictable error pattern. With a measuring range of up to 5.25 V and 8 measuring channels it also fits with the required measurement range of at least up to 5 V and the 8-channel requirement. The recommended ADC was implemented in the next EPS version 1.1.

It should also be noted that to save the development time and money it was decided not to test the 16-bit ADS8344N [28], as ADC128S102 [27] was accurate enough for its purpose and produced more than acceptable results.

3.1.9 Calculating diagnostic data from ADC output

Having gained accurate enough voltage measurements, they can now be used to calculate the current, temperature, and divided voltages. This is accomplished by using the datasheets of the sensors used.

Current is measured by the INA180B3IDBVR [31]. The sensor works by measuring voltage drops across current-sense resistors, also known as shunt resistors. The resistance of these resistors is usually very low. For an example in this case a 2 mOhm resistor is used. The resulting voltage drop is very low, so the sensor amplifies the voltage drop value to level which is measurable by the ADC. Knowing this the formulas can be devised to calculate the current. To calculate the shunt voltage drop the voltage measured at the output by the ADC (V_{OUT}) must be divided by the gain:

$V_{SHUNT} = V_{OUT} / GAIN$

Then using the calculated shunt voltage the current can be calculated:

$I = V_{SHUNT} / R_{SHUNT}$

For the temperature measurement MCP9701AT [32] is used. The sensor in question outputs a voltage level based on how high the sensed temperature is. A higher output voltage means a higher temperature sensed by the sensor. Within the datasheet [31] is given the transfer function for the output voltage: $V_{out} = Tc * T_A + V_{oc}$.

Where:

- T_A = ambient temperature
- V_{out} = Sensor Output Voltage
- $V_{0°C}$ = Sensor Output Voltage at 0°C
- T_C = Temperature Coefficient

From this formula we can easily devise the sensed temperature formula:

$T_A = (V_{out} - V_{oc})/ Tc$

Some of the voltages that are needed to be measured are outside the measurement range of the ADC. Meaning they need to be divided by the use of a voltage divider. With the output voltage and the resistor values R_1 and R_2 known it is possible to devise the formula for the input voltage of the voltage divider:

$Vs = (V_{OUT} * (R_1 + R_2)) / R_2$

3.1.10 Analysing of measurements from KuupKulgur

Using the formulas from the previous chapter the voltages were translated to the appropriate measurements and displayed within the terminal in which the program was run (figure 18).

```
ADC initialized at speed: 10 MHz
Measurements from Battery ADC:
Pin 0 voltage = 4.158 V
Pin 1 voltage = 4.067 V
Pin 2 voltage = 4.293 V
Pin 3 voltage = 4.203 V
Pin 4 voltage = 4.046 V
Pin 5 voltage = 0.004 V
Pin 6 current = 0.003 A
Pin 7 temp. = 27.31 deg C

ADC initialized at speed: 10 MHz
Measurements from Power ADC:
Pin 0 converted voltage of 24 V source = 22.662 V
Pin 1 converted voltage of 12 V source = 12.376 V
Pin 2 converted voltage of 5 V source = 5.114 V
Pin 3 current = 0.003 A
Pin 4 current = 0.022 A
Pin 5 current = 0.672 A
Pin 6 current = 0.366 A
Pin 7 current = 0.000 A
```

Figure 18. ADC128S102 output measurements

The channels of the two ADC-s are named as follows:

Battery ADC

0. Battery pack cell 1
1. Battery pack cell 2
2. Battery pack cell 3
3. Battery pack cell 4
4. Battery pack cell 5
5. GND and battery pack voltage difference
6. Battery pack output current
7. Ambient temperature

Power ADC

0. 24 V source divided
1. 12 V source divided
2. 5 V source divided
3. 3.3 V current
4. 5 V current
5. 12 V current
6. 24 V current
7. Battery pack charging current

The measurement results for the most part closely match the expected values. However, there are some inaccuracies. The most obvious source of inaccuracy comes from the ADC itself (figure 17). Additionally, the signal that reaches the ADC inputs passes through various filters, such as op-amps, voltage dividers, RC filters. All these filters contain resistors, which have a deviation from the intended resistance value of up to ±

20 %. These deviations from the intended resistance values cause inaccuracies of the signals that reach the ADC. For an example measuring the 24 V source with a multimeter showed 23.4 V and the ADC measured only 22.662 V. This large output error is most likely due to the resistors within the voltage divider having higher or lower resistance values than intended.

For temperature and current measurements small inaccuracies are caused by the sensors themselves.

3.1.11 Future development

Knowing the error rate of the ADC itself and other main sources of measurement error optimizations can be made to gain more accurate results.

Using more accurate resistors, or in the case of voltage dividers the resistor resistance values can be measured. The exact values can be used within the calculations to provide precise results.

The error rate of the ADC is very predictable, creating a trend line for each linear section of the error graph (figure 17) allows for easy error correction. Simply adding the expected error to the voltage measurement would yield more accurate results.

SUMMARY

The primary objective of this bachelor's thesis was to gain diagnostic data from the EPS through the ADC within the EPS of KuupKulgur. To achieve this, it was needed to develop a working driver for the ADC within the EPS.

The main objective was accomplished through a methodical testing and debugging process. It was possible to perform semi-accurate measurements with the written driver. However, the results were too inaccurate and inconsistent due to the ADC being used. Based on the requirements for the ADC and problems with the ADC in use before, the author of this thesis recommended a new ADC, which through testing appeared to be more accurate, has a more predictable error rate and therefore produced better results. The recommended ADC was then implemented within the next version of the EPS.

The secondary objective of showing the results in an UI was not accomplished in full due to the problems within the hardware. Those problems caused deeper UI integration to be postponed to a later date. However, the results of the measurements were displayed within the terminal in which the program was run, giving a good overview of the results.

KOKKUVÕTE

Käesoleva bakalaureusetöö peamine eesmärk oli saada diagnostilisi andmeid EPS-ilt kasutades KuupKulguri EPS-is asuvaid ADC-sid. Selleks tuli arendada töötav draiver ADC-le.

Peamine eesmärk saavutati meetodilise testimise ja tõrkeotsingu protsessi kaudu. Kirjutatud draiveriga oli võimalik saada suhteliselt täpseid mõõtmisi. Kuigi mõõtmis tulemusi oli võimalik kätte saada, olid need tulemused liiga ebatäpsed. Seda seetõttu, kasutatud ADC ei olnud piisavalt täpne. Arvestades ADC nõudeid ja varem esinenud probleeme kasutatud ADC-ga, soovitas käesoleva töö autor uut ADC-d, mis testide põhjal osutus täpsemaks, omas ettenähtavamat veamäära ja seetõttu andis paremaid tulemusi. Soovitatud ADC implementeeriti järgmises EPS-i versioonis.

Teisejärguline eesmärk, kuvada tulemused kasutajaliideses (UI), ei saanud täielikult täidetud riistvara probleemide tõttu. Need probleemid põhjustasid põhjalikuma UI integratsiooni edasilükkamist. Mõõtmistulemused kuvati terminalis, mis andis hea ülevaate diagnostilistest andmetest.

LIST OF REFERENCES

[1] “Tartu observatory active projects” Accessed 29.12.2024. [Online] Link:

https://tospexgroup.space/projects/

[2] SoniaLopezBravo, aviviano, alexbuckgit, and mhopkins-msft “What is a driver?”. Accessed 29.12.2024. [Online] Link:

https://learn.microsoft.com/en-us/windows-hardware/drivers/gettingstarted/what-is-a-driver-

[3] Judy Lynne Saint and Christopher Saint “Integrated circuit”. Accessed 29.12.2024. [Online] Link:

https://www.britannica.com/technology/integrated-circuit

[4] “What are Digital Integrated Circuits? Accessed 29.12.2024. [Online] Link:

https://technav.ieee.org/topic/digital-integrated-circuits

[5] Chris Pearson “High-Speed, Analog-to-Digital Converter Basics” Application Report SLAA510–January 2011. Texas Instruments, Dallas, Texas. [PDF] Link:

https://www.ti.com/lit/an/slaa510/slaa510.pdf

[6] Programmed ADC (ADC108S022) datasheet. [PDF] link:

https://www.ti.com/lit/ds/symlink/adc108s022.pdf?ts=1716559563858

[7] Piyu Dhaker “Introduction to SPI Interface” September 2018. Accessed 29.12.2024. [Online] Link:

https://www.analog.com/en/resources/analog-dialogue/articles/introduction-to-spi-interface.html

[8] “Serial Peripheral Interface (SPI) for KeyStone Devices User’s Guide”. User Guide SPRUGP2A – March 2012. Texas Instruments, Dallas, Texas. [PDF] Link:

https://www.ti.com/lit/ug/sprugp2a/sprugp2a.pdf

[9] Miguel Usach Merino “AN-1248: SPI Interface”. Accessed 29.12.2024. [Online] Link:

https://www.analog.com/en/resources/app-notes/an-1248.html

[10] Owen Bishop “Understand Electronics (Second Edition)”, 2001.

https://www.sciencedirect.com/topics/engineering/breadboard

[11] Breadboard BB400 datasheet. [PDF] link:

https://www.mouser.com/datasheet/2/58/BPS-DAT-(BB400)-Datasheet-932623.pdf

[12] Zachariah Peterson “What is a PCB and Intro to PCB Design” January 2024.

Accessed 29.12.2024. [Online] Link:

https://resources.altium.com/p/what-is-a-pcb

[13] Example picture of a prototype board created by hand

https://content.instructables.com/F2Q/270S/G9M4OS1A/F2Q270SG9M4OS1A.jpg?auto=webp

[14] " Research & In-house PCB Prototyping" Accessed 29.12.2024. [Online] Link:

https://www.lpkf.com/en/industries-technologies/research-in-house-pcb-prototyping/about-research-in-house-pcb-prototyping

[15] “LPKF ProtoMat 44” Accessed 29.12.2024. [Online] Link:

https://www.lpkf.com/en/industries-technologies/research-in-house-pcb-prototyping/products/lpkf-protomat-e44

[16] “Breakout boards – what are they and why you should use them” Accessed 29.12.2024. [Online] Link:

https://soldered.com/learn/breakout-boards-what-are-they-and-why-you-should-use-them/

[17] Adafruit breakout board of ADS1115

https://cdn-shop.adafruit.com/970x728/1085-09.jpg

[18] "What Is a Logic Analyzer?" Accessed 29.12.2024. [Online] Link:

https://articles.saleae.com/logic-analyzers/what-is-a-logic-analyzer

[19] Chuck House "Logic State Analyzer". Accessed 29.12.2024. [Online] Link:

https://hpmemoryproject.org/wb_pages/wall_b_page_12.htm

[20] "NASA's Network of Small Moon-Bound Rovers Is Ready to Roll" March 2024. Accessed 29.12.2024. [Online] Link:

https://www.nasa.gov/technology/nasas-network-of-small-moon-bound-rovers-is-ready-to-roll/

[21] Dr. David R. Williams "The Apollo Lunar Roving Vehicle" May 2016. Accessed 29.12.2024. [Online] Link:

https://nssdc.gsfc.nasa.gov/planetary/lunar/apollo_lrv.html

[22] LRV 1 before being shipped to KSC

https://www.nasa.gov/wp-content/uploads/static/history/alsj/a15/ap15-Boeing-LRV-2A297777.jpg

[23] Linda Herridge "Commercial CubeRover Test Shows How NASA Investments Mature Space Tech" Dec 2020. Accessed 29.12.2024. [Online] Link:

https://www.nasa.gov/missions/artemis/clps/commercial-cuberover-test-shows-how-nasa-investments-mature-space-tech/

[24] Tartu Observatory Space Exploration Group "KuupKulgur The Estonian Lunar Rover"

https://tospexgroup.space/projects/kuupkulgur/

[25] Andrea Paterniani: Linux spidev library

https://github.com/torvalds/linux/blob/master/drivers/spi/spidev.c

[26] Saleae logic 8-channel logic analyser datasheet. [PDF] link:

https://downloads.saleae.com/specs/Logic+8+Data+Sheet.pdf

[27] Recommended ADC (ADC128S102) for future development datasheet. [PDF] link:

https://www.ti.com/lit/ds/symlink/adc128s102.pdf?ts=1714919023234&ref_url=https

%253A%252F%252Fwww.google.com%252F

[28] Considered ADC (ADS8344N) datasheet. [PDF] link:
https://www.ti.com/lit/ds/symlink/ads8344.pdf?ts=1733400831778&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FADS8344

[29] Considered ADC (MCP3008) datasheet. [PDF] link:

https://ww1.microchip.com/downloads/aemDocuments/documents/MSLD/ProductDocuments/DataSheets/MCP3004-MCP3008-Data-Sheet-DS20001295.pdf

[30] DP832A power supply store page

https://rigolshop.eu/products/power-supply/dp800/dp832a.html

[31] Current sensor (INA180B3IDBVR) used within the EPS datasheet. [PDF] link:

https://www.ti.com/lit/ds/symlink/ina180.pdf?ts=1734194142374&ref_url=https%253A%252F%252Fwww.ti.com%252Fde-de%252Famplifier-circuit%252Fcurrent-sense%252Fanalog-output%252Fproducts.html

[32] Temperature sensor (MCP9701AT-E_LT) used within the EPS datasheet. [PDF] link:

https://componentsearchengine.com/Datasheets/3/MCP9701AT-E_LT.pdf