Smart Case for Remote Radio Kit

Emil Östlund
Abstract

Smart Case for Remote Radio Kit

Emil Östlund

The thesis aims to develop a prototype for a Smart Case for Remote Radio Kits at the department of Demo & Event at Ericsson in Kista.

The smart case consists of a mechanical structure (the case itself with) and an electronic system that includes a temperature sensor, a LCD display showing the temperature, a GPS (global positioning system) module for positioning the case, a GSM (Global System for Mobile Communications) module and a microcontroller Arduino UNO. The Case is modelled in 3D with the help of CAD software and then printed with a 3D printer. A down-scaled prototype is built with the help of the 3D printer and the 2D drawing will be used when the full scaled model is produced. The Arduino UNO handles temperature sensor and GPS measurements, LCD display, and the transmission of measurement data using GSM module via text message (SMS) to a cell phone or to a server over the Internet.

The projected ended up with all the drawings and models finished for the Case as well as the implementation of down-scaled prototypes. The electrical system was tested and finished individually. But the complete system cannot be assembled inside the Case due to the time limitation. This means that the project can be further extended, where a full scale model can be developed and the electrical control system can be assembled together and mounted inside the Case.
Acknowledgements

I would like to thank the Department of Demo & Event at Ericsson, Kista and everyone that I have gotten to work with there for giving me the opportunity to execute my thesis project there. The project have been educational and I am grateful that I got to participate and see how a large tech company works and operates. I also personally want to thank Magnus Sandström, Manager Service Operator at Ericsson for being my supervisor and mentor during this project.

Lastly, I want to thank my subject reviewer Ping Wu, PhD, Associate Professor of Electrical Engineering specialized in signal processing at Uppsala University for helping me and pushing me during this thesis project.
Table of Contents

1. Introduction .................................................................................................................. 1
   1.1. Background .............................................................................................................. 1
   1.2. Overview .................................................................................................................. 1
   1.3. Purpose and goals ................................................................................................... 2
   1.4. Tasks and scope ...................................................................................................... 2
   1.5. Method ..................................................................................................................... 3
   1.6. Outline ..................................................................................................................... 4
2. Theory .............................................................................................................................. 6
   2.1. Overview of the smart case ...................................................................................... 6
   2.2. Data transfer and communication ............................................................................ 7
       2.2.1. Serial Peripheral Interface ................................................................................. 7
       2.2.2. Inter-Integrated Circuit ..................................................................................... 7
       2.2.3. Universal Asynchronous Receiver-Transmitter .................................................... 8
       2.2.4. Transmission Control Protocol ......................................................................... 8
       2.2.5. User Datagram Protocol ................................................................................. 9
   2.3. GSM - Global System for Mobile Communication ................................................... 9
   2.4. GPS - Global Positioning System ............................................................................ 10
   2.5. PWM - Pulse width modulation .............................................................................. 11
3. Implementation ................................................................................................................ 12
   3.1. Flight Case ............................................................................................................... 12
       3.1.1. Temperature monitoring .................................................................................... 13
       3.1.2. LCD Display ...................................................................................................... 13
       3.1.3. Mobile communication ...................................................................................... 13
       3.1.4. GPS Location tracker ....................................................................................... 13
       3.1.5. Led Panel and diodes ........................................................................................ 13
       3.1.6. Fans and Cooling ............................................................................................... 13
       3.1.7. Hinges for the Case Lid ..................................................................................... 14
   3.2. CAD modelling ........................................................................................................ 14
       3.2.1. FlightCase & Trolly .......................................................................................... 15
       3.2.2. Case Base .......................................................................................................... 17
       3.2.3. Case lids & cover ............................................................................................... 17
       3.2.4. Branding ............................................................................................................. 18
       3.2.5. Support beams ................................................................................................... 18
       3.2.6. Railing ................................................................................................................. 19
       3.2.7. Hinges ................................................................................................................ 19
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>AC</td>
<td>Alternating current</td>
</tr>
<tr>
<td>ADC</td>
<td>Analog to Digital Converter</td>
</tr>
<tr>
<td>BJT</td>
<td>Bipolar Junction Transistor</td>
</tr>
<tr>
<td>CAD</td>
<td>Computer-aided design - software for modelling</td>
</tr>
<tr>
<td>COM</td>
<td>Communication port - serial port interface</td>
</tr>
<tr>
<td>dBi</td>
<td>dB(isotropic) - Measurement for forward gain of an antenna compared to the theoretical isotropic antenna</td>
</tr>
<tr>
<td>DC</td>
<td>Direct current</td>
</tr>
<tr>
<td>DHCP</td>
<td>Dynamic Host Configuration Protocol</td>
</tr>
<tr>
<td>DNS</td>
<td>Domain Name System</td>
</tr>
<tr>
<td>EDGE</td>
<td>Enhanced Data Rates for GSM Evolution</td>
</tr>
<tr>
<td>EGPRS</td>
<td>Enhanced General Packet Radio Service - EDGE</td>
</tr>
<tr>
<td>FTP</td>
<td>File Transfer Protocol</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
</tr>
<tr>
<td>GSM</td>
<td>Global System for Mobile Communication</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical-User Interface</td>
</tr>
<tr>
<td>HMI</td>
<td>Human-Machine Interface</td>
</tr>
<tr>
<td>HTTP</td>
<td>Hypertext Transfer Protocol</td>
</tr>
<tr>
<td>HTTPS</td>
<td>Hypertext Transfer Protocol Secure</td>
</tr>
<tr>
<td>I/O</td>
<td>Input / Output</td>
</tr>
<tr>
<td>I2C</td>
<td>Inter-Integrated Circuit - single ended serial computer bus</td>
</tr>
<tr>
<td>IC</td>
<td>Integrated Circuit</td>
</tr>
<tr>
<td>IDE</td>
<td>Integrated development environment</td>
</tr>
<tr>
<td>IMSI</td>
<td>International Mobile Subscriber Identity</td>
</tr>
<tr>
<td>Iot</td>
<td>Internet of things - connectivity from a device to the internet</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>26</td>
<td>LCD</td>
</tr>
<tr>
<td>-----</td>
<td>-------</td>
</tr>
<tr>
<td>27</td>
<td>LED</td>
</tr>
<tr>
<td>28</td>
<td>LTE</td>
</tr>
<tr>
<td>29</td>
<td>MISO</td>
</tr>
<tr>
<td>30</td>
<td>MOSI</td>
</tr>
<tr>
<td>31</td>
<td>PLA</td>
</tr>
<tr>
<td>32</td>
<td>PWM</td>
</tr>
<tr>
<td>33</td>
<td>QWIIC</td>
</tr>
<tr>
<td>34</td>
<td>RF connector</td>
</tr>
<tr>
<td>35</td>
<td>RISC</td>
</tr>
<tr>
<td>36</td>
<td>RTK</td>
</tr>
<tr>
<td>37</td>
<td>RX</td>
</tr>
<tr>
<td>38</td>
<td>SCKL</td>
</tr>
<tr>
<td>39</td>
<td>SIM</td>
</tr>
<tr>
<td>40</td>
<td>SMA</td>
</tr>
<tr>
<td>41</td>
<td>SPI</td>
</tr>
<tr>
<td>42</td>
<td>SS</td>
</tr>
<tr>
<td>43</td>
<td>TCP</td>
</tr>
<tr>
<td>44</td>
<td>TTL</td>
</tr>
<tr>
<td>45</td>
<td>TWI</td>
</tr>
<tr>
<td>46</td>
<td>TX</td>
</tr>
<tr>
<td>47</td>
<td>UART</td>
</tr>
<tr>
<td>48</td>
<td>UDP</td>
</tr>
</tbody>
</table>
Chapter 1

1. Introduction

1.1. Background

Ericsson has over many years used their Remote Radio Kits to be able to set up networks, firewalls and servers on remote locations. In the current situation the company’s radio equipment is sent in shock-resistant boxes, so called flight cases. The cases meet the requirements to send the fragile equipment worldwide without damaging it, but the cases does not have any other function then to protect the equipment. The cases does not have any smart and clever functions that would increase the safety and useability of the cases.

When the cases and radio equipment is put in place to be used for demoing, the boring boxes do not give a professional and serious impression.

Ericsson has recently collaborated with companies that are at the forefront of their technological areas, for example see [1]. When Ericsson visit the companies and offers their services like network, firewalls and server solutions they want to impress the customer by updating their equipment in such a way that it will attract the eyes and raise questions and thoughts. At the moment, Ericsson set out their so called flight cases and not many spectators or customers are impressed and Ericsson wants to change this.

Ericsson wants to change the way their equipment is displayed and how its exterior looks so that it holds the same level as their software, hardware and services. The new case that should be modeled and manufactured should give the appearance of a serious and modern approach.

Since a new case is created the company also saw the opportunity to implement new smart functions that will help and make it easier for employers while working with the equipment. The functions should make their radio cases more efficient and safer. To be able to track where their equipment is all the time is a good way to increase the safety aspect as an example.

This thesis project was written and completed by one student from the Bachelor Program in Electrical Engineering at Uppsala University. The project was done at Ericsson at the department of Demo and Event.

1.2. Overview

The project is built on both hardware and software, both are used to create a system for the different functions. Software is used to control the hardware, they need to work together. A block diagram on how the system is build is illustrated in figure 1.1. It is divided into two sections, hardware and software and the diagram shows how each part of the system works. As an example how the GSM function works, the hardware is a breakout board with a GSM module, this is explained in section “3.4.5. Radio and telecom communication” and it uses a SIM Card to operate the telecommunication with the help of the LTE Antenna. This hardware implementation is then controlled by the Arduino IDE software.

Hardware such as electrical components and various circuit boards is bought and connected together, the electrical components and part of the system that isn’t bought as a complete circuit is designed in both Fritzing and LtSpice, where electrical schematics is drawn. This is explained in chapter 3.
Arduino IDE software with the help of its libraries is used to control the system, where a circuit board named Arduino Uno is used as the brain.

The system is then mounted to the new prototype case that is constructed. The new case is modelled in a software called AutoDesk Inventor, which is a 3D CAD modelling software. Each part of the new case and also the flight case that is used as the base is modelled in the software, where both 3D models is made and also 2D drawings is exported, this can be seen in chapter 3 of the report.

The project will not immerse much in complicated theory other than different types of data- and telecommunication and also a brief explanation on how GPS systems work. The circuit boards that are bought and used in the project will be assumed to work as intended and no complicated troubleshooting or measurements will be done if not necessary, this is done if the system is not working as intended or as expected. Same goes for the electrical components, as an example, will the project assume that the antennas work as stated in their datasheets and specifications. The circuits is explained, why and how it is used and its main components is presented in chapter 3.

1.3. Purpose and goals

The goal of this project is to develop and produce a finished prototype of a product. The product itself is a new smart case solution that provides both a new exterior of their equipment and also smart and useful functions to expand the usability of the radio kits.

1.4. Tasks and scope

The scope of the project covers the areas literature study and both hardware and software implementations where the tasks are explained below, the task is divided into different parts:

- **Literature study**: This is done in the beginning of the project, to collect and study relevant theories, information and data that will be used when both the hardware and software is implemented later on.
Information on different types of data communication is studied, how SPI, I2C and UART works and operate and also how it is implemented in different types of circuit boards and IC circuits. How it is used and what type of data it can receive or send.

Different IP and server solutions is also studied such as UDP and TCP, where the goal was to get relevant information and an understanding of what type of protocol should be used to send data collected by the system.

Since the system will use GPS to collect positioning data and send that and other information via IP and radio, both GPS system and radio communication were necessary to be researched further in order to get an understanding on how it works and operates.

Each component and electrical circuit bought have been studied and information on how they operate and how they should be connected and what type of communication that should be used between them.

A lot of time has been spent on the online forums related to the Arduino community [2] where information related to both hardware and software have been studied.

- **Software**: This task is done with the help of multiple computer software, where each software is used to create something hardware related or to control hardware.

  CAD software is used to create and model all the parts and assemble the new prototype case solution. Different electrical schematic and simulation programs is used to draw schematics and to test different circuit solutions that will be implemented as hardware and components in the system.

  Arduino IDE software with the help of its libraries is used to control the hardware and everything is coded in the IDE. The code is written on a PC and transferred onto hardware and in this case it is a Arduino Uno circuit board. The code has the information and it controls all the functions that the system should have and be able to operate.

- **Hardware**: The hardware is both related to electrical components and circuits but in this project it also consists of the case, flight case and other hardware used when mounting everything together.

  Electrical components are used and need to be soldered together in order to create and construct the circuits that were designed. Electrical circuit boards need to be connected together, some boards are connected via quick connected cables but some also need to be soldered.

  The case prototype that is constructed needs to be mounted to the flight case and the electrical system needs to be mounted inside that as well.

- **Evaluation**: Evaluation of each part of the project is done, this is written about in the discussion section of the report, chapter 6.

### 1.5. Method

A prestudie was done where information was collected and discussion was conducted between relevant parties and involved people.
First the project needed to be specified, what was expected to be accomplished. What was the case going to look like? What dimensions should it have? What functions should it have? What materials were going to be used? How should the case be constructed? What electrical components is going to be needed? What software was going to be used? What should be ordered? All these questions were discussed in small meetings and information were collected and a plan were decided.

The 5G Remote radio kits were going to be slimmed down to a smaller sized case, this were discussed and measured out with these things in mind:

- The size of the hardware used.
- The extra space needed for airflow.
- The space to be able to reach and work with the hardware.

A Flight case with the size of 12U were chosen.

The design of the case has been implemented in many iterations and the look of the case have been tweaked a lot. Glass fiber were decided to be used as the material for the case, the 2D CAD drawings is going to be used when the full scaled version is produced and the drawings will be sent to a company that can create the product.

The smart functions implemented is written about in the section “3.1. Flight Case.” The functions implemented is temperature reading, LCD display, Mobile communication, GPS tracking, LED Panel and control of smaller LED:s and electrical fans.

Before the project started a meeting was held where a project plan was written. The project plan involved a Gantt schedule where every step of the project were decided and when it should start and be finished. About 6 weeks after the project started the project plan was renewed and a new Gantt schedule was made.

The Gantt schedule are given in appendix C and the date that the project should be finished at was decided to be 14/06 - 2019.

The scope of the project and end goal were specified early in the project. The goal is to make a new exterior for Ericsson’s Remote Radio Kits, a new refreshed version that is up to date. The design should be simple and clean but it should also have the ability to attract people's attention when walking by. The exterior should have the same standard as their equipment inside and services they provide. The slogan that were hatched during the prestudie became “Simple but advanced”. The case should look simple but have advanced smart functions and gimmicks.

1.6. Outline

The technical report is structured in six different chapters and the references and appendices is presented at the end of the report.

In chapter 1 the introduction is presented, where the projects background, its purpose & goal and its task & scope is presented. The introduction is supposed to give the reader an overview of the project and its specifications.

In chapter 2 the relevant theory that is used in the project is presented and described. Different data types and different types of communications and the concept behind them. This gives the reader an understanding on how the data collected is sent between different hardware and software solutions and also how different electrical circuits boards and IC circuits are connected and how they
communicate. The project uses different sensors and GPS data is collected and sent via GSM, the meaning of both GSM communication and a basic overview over GPS is presented.

Chapter 3 explains the implementation of the system and how the different parts of the project is executed, both hardware and software. It explains the different types of software used, such as how the CAD modelling were done and how the electrical schematics were drawn and also how the programming were executed and coded. This chapter also describes and present the hardware implementation, what types of electrical components and circuits that is used and why. How the system is assembled and the troubleshooting is presented and documented in this chapter as well.

In chapter 4 the result is presented, the finished prototype with the system working as intended and each part assembled together to create the new smart case solution.

Chapter 5 presents the evaluation of the project were the discussion can be found. The discussion presents and defends why and how different parts were executed in this way during the time of the project. Things like laws and various specifications that the project has taken into account is explained.

In the last chapter, chapter 6 the report presents the conclusion. The conclusion is related to the results and what went good or bad during the project is written about. Improvements and what can be done to perfect the case is presented and further studies.

The references is following the IEEE (Institute of Electrical and Electronics Engineers) standard style and is presented in this way. The IEEE standard is the citation style that is used in electronics, engineering, computer science, telecommunication and information technology reports [3].

The Appendices can be found at the end of the report, models, prototypes and software that were developed and created during the project is presented.
Chapter 2

2. Theory

2.1. Overview of the smart case

The smart case is built to be mounted on top of a 19 inch flight case, where the case and various hardware and mounting solutions is modelled in CAD software. The smart case is illustrated as a 3D model in figure 2.2.

The smart case is constructed in glass fiber and the case is divided into three parts, the base, the front lid and the rear lid. The front and back lid is mounted on hinges so that they can be opened, so that hardware inside can be accessible.

The case will have an electrical control system mounted inside, this is illustrated in figure 2.3. The control system will have many different functions, these are explained in “3.1. Flight Case.” The control system will collect data from sensors and other modules and the data collected will be able to be sent via both radio and IP solutions, to either a cellphone as a text message or to a server where information can be diagnosed and stored.

Figure 2.2. An overview of the smart case, mounted on top of the 19 inch flight case.
2.2. Data transfer and communication

Different types of data communication is suitable for different applications [4]. The different electrical circuits communicate between each other with the help of protocols and clusters or packets of data.

2.2.1. Serial Peripheral Interface

SPI is a four wire bus protocol for synchronous serial communication, this is shown in figure.2.4. It is optimal for short distance communication and is almost used in every embedded system [5]. It communicates in full Duplex meaning that it can communicate in both directions simultaneously [6].

The master side is creating a clock signal on SCKL that the rest of the communication synchronice with. MISO and MOSI is used for transmitting and receiving data between both ends. The MISO (Master Input Slave Output) line is receiving data from the slave side and MOSI (Master Output Slave Input) line is sending data to the slave side. The SS (Slave Select) decides if the slave end is active or not, this is usable if a master is sending data to multiple slaves [7].

SPI communication is done in this case by sending a digital 8-Bit sequence each clock cycle [8].

![Figure.2.4. Basic overview on how SPI communication works.](image)

2.2.2. Inter-Integrated Circuit

I2C is a computer bus that communicate on two wires, SDA (Serial Data) and SCL (Serial Clock) [9]. The communication is half-duplex, meaning that it can send and receive data in both ways but it can't happen simultaneously. I2C is commonly used in short distance lower-speed peripherals between processors and microcontrollers [10].

Any numbers of masters or slaves can be created virtually and then connected to the two wires. As long as it uses a protocol that defines a unique 7-bits slave address for each device and that the data
sent is in 8-Bit sequence. Some bits will be used for controlling the communication, where it starts, ends and in what direction it should operate [11].

The physical connection needs so called pull up resistors in order for it to work. The pull up resistors task is to set the line to “high” when its not driven low by the open-drain interface. The impedance value is important to be right in order for the signal to transmit data without any signal losses [12].

The calculation for pull up resistors are [13]:
\[
\text{tr}=\text{Rise time of both SDA and SCL signals}, \quad \text{Cb}=\text{Capacitive load for each bus line} \\
\text{VOL}=\text{Low-level output voltage} \\
\text{Rp}_{\text{min}}=\text{VCC}-\text{VOL}_{\text{max}}\text{IOL} \\
\text{Rp}_{\text{max}}=\text{tr}\left(0.8473*\text{Cb}\right)
\]

![Figure 2.5](image)

**Figure 2.5.** Basic overview on how TWI and I2C communication work.

### 2.2.3. Universal Asynchronous Receiver-Transmitter

UART is a data communication technique that uses two wires. It is a hardware implementation, it supports asynchronous serial communication in both directions. The Tx line transmits data and the Rx line receives data. The transmitting wire is connected to the receiving wire on the other end and vice versa [14] as illustrated in figure 2.6.

UART can only communicate between two devices. It can operated in three different modes [15].
- Simplex - where data is sent in one direction.
- Half Duplex - where it can send and receive data but not simultaneously.
- Full Duplex - where data is sent and received simultaneously.

![Figure 2.6](image)

**Figure 2.6.** Basic overview on how UART communication works, Rx - Tx.

### 2.2.4. Transmission Control Protocol

TCP is in the transport layer of the Internet protocol suite and is an internet protocol (IP) that is used in network implementations. TCP is a reliable way of transporting data or packages between different hosts [16]. TCP is used in almost every internet application, common examples are email, file transfer and the world wide web. Services that uses TCP are HTTP, HTTPS, FTP and also commonly used in computer games [17].

TCP keeps track so that every packet that is sent is also received on the other end, it ensures that the packets will reach its destination in the same data order as sent. If a packet is scrambled in the
transmission the TCP will sort the data. All these extra functions makes TCP a reliable way of transmitting data [18], but since it always checks if the data sent is received the TCP transmission are slow compared to other protocols, UDP for an example is faster [19]. UDP is explained in the next section.

2.2.5. User Datagram Protocol

UDP is similar to TCP but is a faster way of transmission. UDP transmission doesn't notice or care if all the data sent is not received, it does not have a check ups as TCP. This is one of the reasons why it is a faster way of transmitting data. UDP cannot ensure reliable data transfer and cannot guarantee delivery of the data sent, it doesn't have error correction either [20].

Even though UDP isn't as reliable as TCP it has it uses, such as DNS, IP telephony and DHCP [21].

![Figure 2.7. TCP vs UDP.](image)

2.3. GSM - Global System for Mobile Communication

GSM or Global System for Mobile communication is a mobile network, it is used all over Europe and also in other parts of the world. GSM is a second generation of standards for mobile networks, often referred to as 2G. The GSM operate on three different frequency bands [22]:

- 900 MHz, this was the original frequency for the GSM system.
- 1800 MHz, this was added later on to keep up with the demand and growing number of users.
- 1900 MHz, this is mainly used in the United States.

GSM is based on TDMA (Time division multiple access) system, where it uses the technology of digital signaling and speech channels and it is the backbone of both GPRS and EDGE technology [23]. GSM phones use a SIM Card to be able to identify the users subscription information [24].

The Network is built on four different main parts, they all need to work together in order for the network to function. The parts are the mobile device, the base station subsystem (BSS), the network switching subsystem (NSS) and the operation and support subsystem (OSS). The device connects via hardware to the network and the SIM Card provides the network with information and identity from the user, figure 2.8. illustrate the four different parts [25].
2.4. GPS - Global Positioning System

The Global Positioning System is a location tracking and navigation system that is based on satellites, the GPS system is made up of at least 24 different satellites that are accessible. The satellites were put up in the first place for military use by the department of defense in United States and later in the 1980s the public was given access to them [26]. The satellites used circles the earth twice a day in a precise orbit and each satellite transmit an orbital parameter and a unique signal [27]. This information is used and the signal is decoded and a precise location is computed from the information collected.

The GPS system calculates the users position by measuring the distance to each satellite and the time it takes for the system to receive the signal transmitted from the satellite. In order to calculate the users positioning in 2 dimensions, latitude and longitude, the GPS needs information and communicate with at least 3 satellites. If the system manage to communicate with four or more satellites, the system can calculate the users positioning in 3 dimensions, latitude, longitude and altitude [28]. This is illustrated in figure2.9. below.

GPS used today will normally communicate with at least 8 satellites, but this varies depending on where you are located and what time of day it is [29].

There are different types of satellites in orbit, these are GNSS, GPS, BeiDou and GLONASS [30]. The GPS module and technique used in this project is described in section “3.4.6. GPS” and it uses a GPS/GNSS antenna and information from GPS and GLONASS satellites.
2.5. PWM - Pulse width modulation

PWM or Pulse Width Modulation is a technique where the goal is to reduce the average value of an analog signal. The PWM signal is a square wave where it oscillates between “high” and “low”, it acts as a digital signal. When the signal is “high” it will work as normal and when its “low” it will not do anything, the ratio between how long time the signal is “high” versus “low” is the technique which create the average output value of the signal [31].

The period time of the signal is varied and when talking about PWM signals the ratio between how much percent of the period of time the signal is high is called Duty cycle [32], also shown in figure.2.10. So if the Duty cycle is set to 100% the signal is constant high which means that the average value of the signal is the same as the output from the electrical board and in this case the voltage level out from the Arduino I/O. When it is set to 50% the average value will be half of what was set on the I/O.

Pulse width modulation is illustrated in figure.11, where pulse width is the duty cycle and the digital signal is multiplied with the sinusoidal signal to create the Analog PWM signal.
Chapter 3

3. Implementation

3.1. Flight Case

Flight Case also known as Road Case or ATA Case is a box where fragile equipment is shipped inside [33]. The case can be in many different shapes and sizes, the inside will vary depending on what type of equipment is supposed to be stored inside [34]. An area where these cases are commonly used is when shipping musical instruments, these boxes can often be seen near a music stage with instruments inside. Ericsson uses these Cases to ship their sensitive radio and server equipment all around the world both by lorries and by air travel.

The boxes are built to handle abuse, when they are thrown around at the airport or when shipped in other ways. The construction is made from panels that are made from both ABS plastic, fiberglass and cabin-graded plywood sheets that are mounted together with the help of a metal frame. Inside the case the sides are covered with polyurethane foam to protect the equipment. Some cases can also be constructed with server racks mounted to shock absorbing dampers for extra safety.

The Flight Case used in this prototype is a 12U 19 inch Flight Case with shock mounted server racks with the dimensions of: width 728mm, height 944mm and depth of 1200mm with the lids. The case is sitting on top of 120mm castors, the flight case is illustrated in figure.3.1.

The Flight case will act as a base and the prototype case will be built around it. This is written and discussed about in chapter 5 of the rapport.

Figure.3.1. 12U Rack Flight Case.

The new solution with a case mounted on top of the flight case will have an electrical system implemented with functions that will ease the use of the radio kits, the functions that the smart case should have been decided in the prestudie and why it was implemented is explained below.
3.1.1. Temperature monitoring

The temperature monitoring is done by measuring the temperature in the case in different locations. The temperature is measured with an Analog TMP36 sensor. The temperature is important to measure and keep track off [35], the optimal temperature for big server rooms is between 20 and 22 degrees celsius according to [36]. So the best case scenario would be to keep the temperature down close to that even though it is in a smaller and tighter space. The lower the temperature the better working condition for the hardware inside.

The data collected from the temperature sensor is then used to send information via GSM to a cell phone and also to be displayed on an LCD mounted to the case. The complete system will have warnings for when the temperature reaches critical levels, the warnings will be able to be seen with the help of coloured LED:s.

3.1.2. LCD Display

An LCD display is used and mounted to the case. The LCD should act as the GUI and HMI, it should be able to show the data collected by the rest of the control system. The code will be written so that the lcd will have different menus for displaying different things.

3.1.3. Mobile communication

The mobile communication is implemented so that information that is collected by the system can be sent from the case to a cell phone, the system has the ability to send messages via SMS when critical moments happens. This is also used to send data and information to a server via IP solutions, where data is monitored. The data sent is average temperature and location. Since the system will have the ability to receive and send text messages, various functions can be added later if there is time and a use for it. Functions that can be added to operate via SMS can for example be a shutdown command where the user can send a text message from a phone with the message “Shutdown” and then the system can shutdown and hibernate until its restarted.

3.1.4. GPS Location tracker

The ability to see the position of the case and its expensive radio equipment is a good safety measurement. A GPS module is implemented in order to achieve that. The data from the GPS can be sent both via SMS and to a server with the help of the mobile communication.

3.1.5. Led Panel and diodes

The case is supposed to stand out and bring attention. The lid of the case will have the Ericsson name milled out on it, behind that a LED Matrix or Panel is mounted. The LED Matrix will light up and have different lightning scenes displayed on it.

Diodes will be implemented to illuminate light where it is needed and also act as warning light for the temperature monitoring.

3.1.6. Fans and Cooling

In order to keep the temperature inside the case down, multiple fans were installed. The fans were placed in positions that would optimize airflow [37]. The fans can be turned on with the help of a switch mounted on the case.
3.1.7. Hinges for the Case Lid

The Case were designed so that the front and back part could be opened up and removed if needed. The Lids covering the front and back were decided to be opened and operated in a certain way. The hinges should operate in such a way that the lids could move out from the main case and then up to be stored on top of the case. More in detail why the lids and hinges were designed in this way is written about in chapter 5 of the report.

3.2. CAD modelling

CAD - Computer Aided Design is software that is used to create models in both two-dimensional and three-dimensional environments. The program provides tools for optimization, simulations and also great analytic data. It is used in many applications and engineering areas, such as automotive, shipbuilding, aerospace industries and industrial architectural designs [38].

3D Modelling
A three-dimensional model is created in the software to represent how the product should look like. A 3D model is created by first drawing a shape or object in two-dimensions with the help of a common used x-y coordinate system. The 2D object is then used as a mold for setting up a 3D object in a x-y-z coordinate system where x and y is on the same axis as before. The shape is often drawn with its start in origo, this is done so that you always have a reference point when you keep on building the object. The 3D shape can be made to look like anything, hence why it is so commonly used in almost every construction and product development.

Simulations
The program supports different types of simulations, examples can be stress analysis test, wind flow simulations, simpler movements of objects or even to control the strength and solidity of an object [39].

In this project no complicated simulations have been studied. The simulations that have been used is how the hinges should operate to be able to open and close the Case lid in a safe and smart way.

Importing files
Every component that is used in the finished product have been designed and modelled except the electrical components used. The electrical circuits models were downloaded from a website, see [40] and then later imported into the CAD program.

Assembly
Once all the 3D objects are created it's time to assemble them together. This is done by first importing all the objects that are going to be assembled together into the program. In the program you set rules and limits on how the objects should relate to each other. For two objects moving in correlation to each other you assemble objects with limitations in how much the objects are supposed to move or rotate around each other. The non-moving objects are merged together and locked in place with the limitations the user choose.

2D drawings
2-dimensional drawings are exported from the 3-dimensional object. The drawing shows one object and all its dimensions. A proper 2D drawing should display the object so that there is no uncertainties where all sides are shown and all its dimensions. For bigger projects with multiple parts every object has its own 2D drawing but it also comes with an exploded view showing all the components and how they relate to each other [41].
The 2D drawing is created so that it can be handed over to the companies that are going to create the object. For an example in this project the case is exported to a drawing and then sent off to be developed.

The design were discussed and decided between the people involved. The design was first drawn by hand (see appendix C, Misc.19) to get an idea on how the model should look like and after that modelled in 3D in many different iterations. All the drawings or models referred to in the CAD modeling and drawing can be found in appendices A and B.

### 3.2.1. FlightCase & Trolley

The Flight Case used is a 19 inch SPS server rack 12U from ProCase and its dimensions are specified in (3.1. Flight Case).

A model of this case or box were drawn in order to measure and to get an overview of how everything should be assembled and so that nothing were interfering with each other. Since the only thing that is important with this model is the outer dimensions and where the locking mechanism is located. The model were never created to be an exact replica of the ProCase, it was just a placeholder since it didn't have to be constructed, it was just a placeholder in the 3D model in order to give a good aesthetic look for the complete model.

**Wheels**

The wheels were measured out to have a diameter of 120 mm and the height from ground to top of the assembly 165 mm and is shown in “Drawing.1”. The wheels were constructed with four different parts being assembled together, every part were individually modeled. First the wheel hub were modeled and then the outer plastic for the wheel, these two were assembled to create the wheel. The bracket were modeled and a cylinder for connecting the wheel to the bracket. Aluminium were selected as the material for the bracket, rod and wheel hub and ABS plastic were chosen as the material for the outer wheel.

**Trolley base**

The trolley base have the basic dimensions of length 1200 mm, width 728 mm and height 12 mm, the trolleys full dimensions is shown in “Drawing.2”. It was constructed as a 2D rectangle and then extruded in 3D and then the chamfer on the edges and holes were then made. MDF fiberboard were chosen as the material.

**Locking mechanism**

The locking mechanism is shown in “Drawing.3” and it was never modeled to be an exact replica of the one that comes mounted on the Flight Case. It were modeled to be used in the assembly to show where the locking mechanism would sit and so that the case around it wouldn't interfere when operating it. The drawing doesn't have any dimensions on it since they are not relevant and all the parts were made in aluminum. The mechanism where constructed in four parts, the base, the cylinder, the butterfly and the rod connecting the butterfly to the cylinder.

The base were drawn from the side view so that the shape could be created in 2D before making the 3D model, to make it simpler. The 2D view were extruded and mirrored to create a symmetric base and later fillets were added to the sharp edges and corners.

The cylinder was created as two 2D circles on top of each other and then extruded to the desired height. A hole were created through the cylinder from the center (XZ-Plane) and fillets were added.
The butterfly was created as a 2D drawing almost looking like mickey mouse ears and then extruded into a 3D object. The big holes in the YZ-plane were included in the 2D drawing and didn't have to be made later. The small holes in the XY-plane were made with the dimensions to fit the connecting rod. The rod is just a 3D cylinder with the dimensions to fit the circle and butterfly.

**Metal shields for the locking mechanism**

The metal shields is mounted on both the base and the lids of the 12U Flight Case, so two different types had to be constructed and they were created in aluminum. They were both made in the same way but with different dimensions, each of the parts dimensions is illustrated in “Drawing.4” and “Drawing.5”.

It started by creating a 2D shape as seen in the base view in the drawing, a rectangle with a radius on the edges on one side. It were extruded to the full height of the part and then material were removed from the underside of the part to create the U shape. Fillets were added in the corners to make it look right and symmetric.

The rivets on the topside of the part were created by using rectangular pattern and circles in 2D and then all the circles / rivets were converted to 3D at the same time.

**12U Flight Case Base**

The Base of the 12U Flight Case started as a two rectangles, with one inside the other with 18 mm between them. The full dimensions of the base is shown in “Drawing.6” and the dimension of the outer rectangle is 960x728 mm. The space between the two rectangles were extruded to the height of 944 mm. After that an 18 mm plate were created on both sides that were open so that the Case now were a hollow box with 18 mm thick walls.

In the XY Plane a 2D drawing were placed, it was a rectangle that were 18 mm offset from every side of the outer rectangle. This was used to create a hole through the box. The basic shape of the Flight Case base were now finished.

Now the indentation for the metal shields were created on the outside of the base, at the end of the sides. A 2D drawing were created on the side were one shape were drawn and that were used with the help of rectangular pattern and mirror to replicate it on all four corners. Fillets were added and this is where the metal shields will mount to.

**12U Flight Case Lid**

The Lid started as a 944x728 mm rectangle and were extruded to the width of 120 mm. The object were now a rectangular box with the dimensions of 944x728x120 mm. Same way as the hole through the base were created the embedment were made to the depth of 102 mm, 120 - 102 = 18 mm, which is the thickness of the wall.

The indentation were created in the same way as in the base but with smaller dimensions, the full dimensions of the lid is shown in “Drawing.7”. Dark grey wood birch were chosen as the material for the Base and the Lids.

The fully assembled 12U Flightcase is illustrated in “Drawing.8”.

---

16 | Page
3.2.2. Case Base

The Base of the case starts as a rectangle with the dimensions of 1014x780 mm. This dimensions are chosen so that there is 50 mm between the case and the flight case. The height of the case is 1014 and the height of the flight case is 944, with the thickness of the case in mind (20 mm) the calculation are, 944 + 50 + 20 = 1014 mm. The width 780 mm of the case are measured so that it wouldn't cover the flight case locking mechanism, the mechanism should be reachable and a worker should be able to operate the mechanism without any interference by the case. The solution to this was to make the base shorter and to lengthen the front and back lids or covers.

The rectangle were extruded 868 mm to create the width of the base and the calculation for the width is 728 mm + 100 mm + 40 mm, where 728 mm is the width of the flight case, 40 mm is the thickness of both sides of the case and 100 mm is to give 50 mm of room on both sides between the case and flight case.

A fillet with the radius of 125 mm were created on both sides on top of the base vertically, different radiuses were tested. If the radius were to big it would interfere with the flight case outer dimension, since it were going to be mounted inside and if it were to small it didn't look right.

The shell function in the program were then used on the object. The thickness were set to 20 mm and this results in the case being 20 mm thick, the basic shape of the base were now created.

On top of the base four identical holes were created, the holes have a diameter of 55 mm. These holes are created in order to fit the railing, this is explained in (3.2.6. Railing).

After the basic shape of the base were done, the company branding had to be implemented. The Ericsson logo were milled on the sides and the department name were embossed 10 mm deep, this is discussed more in-depth in (3.2.4. Branding) and all the dimensions of the base is illustrated in “Drawing.9”.

3.2.3. Case lids & cover

This section explains how the front and rear cover were modelled and the branding part for both objects are explained in the section (3.2.4. Branding). The full dimension for the front and back cover is shown in “Drawing.10” & “Drawing.11”.

Front lid

The front lid started as a 2D drawing from the side view of the lid. The shape is a rectangle with the dimension 1014x210 mm, 1014 mm because of the same way as explained in (3.2.2. Case Base). 210 mm is to cover for the length of the flight case lid and also to cover for the shortened base, the lid is supposed to cover the locking mechanism on the flight case when it is closed as well. One side of the rectangle has its bottom extended with a line of 110 mm and a shape with the radius of 1794.5 mm is connected from the end of the line to the top of the rectangle. This shape is extruded 868 mm and this dimension is the same width as the base and the explanation is the same.

Fillets are now applied to the object, a radius of 125 mm is given to the top vertical edges to match with the base. The front part where the big radius are located is given a fillet with a radius of 75 mm. This radius was tested with different values and if the radius were to big it created problems when the object were hollowed out, since the thickness of the object should be 20 mm everywhere.

The shell function were used on the solid object with the thickness set to 20 mm, the basic shape of the lid were now finished.
Rear cover

The rear cover were also started viewed from the side, it was done this way since it seemed like the easiest way to be able to create the shape with the “wing”. The shape started as a rectangle with the dimensions of 1014 x 250 mm where the height is explained in the chapters above and the width as well.

The wing part were made to be symmetrical around the left edge of the rectangle. It was made with an angle of 135 degrees and length of 100 mm out from the edge, the side on the edge were drawn to be 20 mm to match the rest of the case. The bends starts from the edge and stretches 150 mm in both x and y direction with a radius of 400 mm.

The shape were then extruded 868 mm, same explanation as for the width of the base and front lid. The fillet of 125 mm were added on both edges on top of the shape and also on the wing. Now the basic shape were created and the shell function with thickness of 20 mm were used.

Some extrusions were done on the back side of the cover in order to make room for mounting fans and also a pocket for cables to be threaded through.

3.2.4. Branding

The base and the front lid of the case is branded with both Ericssons name, logo and also the department were the project is executed (Demo & Event).

When it comes to branding the tolerances and specifications are very strict and as illustrated in “Model.12”, Ericsson has its own font for branding and logo, the three diagonal stripes has its set dimensions and angles. This also applies to colours, if something is going to be coloured it have to follow the companies standard colours.

The Branding is shown in “Drawing.12” and also in “Model.21 & Model.22”. Ericsson uses the text font named Ericsson Hilda and this has been used in the bold format with the dimension of 55 mm in both the Ericsson name that is milled out in the front lid and also in the Demo & Event branding that have been embossed in the side of the base. The texts were created with the text function inside a 2D drawing in the program where the official Ericsson font were imported to, the text were then used as a template to mill and emboss the object.

The Logo had to have the angle of 18.435 degrees for the lines and the three stripes are all parallel to each other. This was done in the program by downloading a figure of the official Ericsson logo and then the logo were imported into a 2D drawing and the figure was used as a mask when the logo were drawn. A grid were made in the 2D drawing to ensure that all the lines were parallel and in the right place with the right angles, the grid can be seen in “Model.29”.

3.2.5. Support beams

The support beams are supposed to connect the base of the case to the 12U Flight Case, where there will be one support beam in each corner on the top and on the bottom of the flight case.

The lower support beam is just a rectangle with the dimensions of 780x50x50 mm where 780 mm is the width of the base of the case and 50 mm is the distance between the case and the flight case, the height of 50 mm is to make place to screw the beam into the flight case. The beam is made of aluminum and has three 15 mm holes with M15 threads inside.
The upper support beam is more complicated but it has the same basic dimensions as the lower support beam. The dimension is length of 780 mm and height and width of 100 mm, 50 mm in each direction, both in x and y direction. It will sit on the corner of the flight case. The backside of the beam has a radius of 105 mm to match the inside of the base of the case. It has six 15 mm holes with M15 threads.

How the support beams look like and how it is mounted is illustrated in “Model.6, Model.10, Model.13 & Model.18” and its full dimensions is shown in “Drawings.9 & Drawing.10”.

3.2.6. Railing

The railing were modelled to be a prototype of how it could look like and how to solve the problem with mounting the rail when assembling the case and flight case. This system or railing were never used in the finished prototype, this is talked about in discussion. Instead an already made railing were going to be bought and used.

**Rail**

The railing was made by the help of two 2D sketches and 3D sweep function. A circle with a diameter of 50 mm was placed in XZ-plane and an arch were place in the YZ-plane. The Sweep function uses the circle as a shape to sweep along the arched line that were created in the other plane, after the sweep the basic shape of the rail is done.

The shell function were used to hollow out the rail, the thickness were set to 5 mm and after that a circle were extruded around the bottom of the both ends of the rail to create a lip where the base plate could attach to. The rails full dimensions is shown in “Drawing.11”.

**Base plate**

The base plate were created as a two half circles with one smaller than the other, the outer diameter is 85 mm and the inner is 50 mm, 50 mm to match with the size of the rail and 85 mm to have enough room to be able to mount it to the flight case. This half moon like shape were extruded 20 mm. The cross section of the circle in the XZ-plane where used to create a hole on one side of the section and a pin on the other. The pin and hole have the same dimensions to be able to be merged together when two base plates are used together. Fillets were added and the base plate is used in such a way that when two are used together they work as a locking mechanism for the railing. The base plate has holes on the bottom with threads for mounting it to the flight case. Its full dimensions is shown in “Drawing.12”.

3.2.7. Hinges

Hinges were created in the program and assembled to be able to test how the front lid and rear cover should operated, some simulations were done and a clear figure of what type of hinges were needed got figured out.

3.2.8. Electrical circuits

As mentioned in section above electrical circuits were imported to the program. The models of the electrical circuits were downloaded from the online forum GrabCAD and imported to the CAD program. The electrical circuits were assembled inside the case to get an overview on how they should be mounted and fitted later on and if some brackets or holders needed to be created with the help of the 3D printer.
3.3. 3D Printing

3D Printing is a relatively new technology, it is often used to create prototypes. The process of 3D printing is started as a CAD file or other 3D model file, then the model is used in the process where material is joined or solidified under computer control to create a three-dimensional object, with material being added together. 3D printing is mostly done in plastic (PLA), where it is melted and then cooled to solidify it, see [42] and [43].

Learning how to use the 3D Printer

In order to create prototypes and various brackets with the help of the 3D printer, the basics on how to operate the printer and what settings to be used had to be tested and learned. Simplistic objects were modeled in CAD and then different types of mesh were tested and then sent to the printer. The different results were analyzed and an understanding on which mesh and settings were the optimal for the material and printer used. The 3D printer used is a Creality CR-10S [44] and the material is standard 1.75 mm PLA filament [45]. One of the test objects that got printed can be seen in appendix F, the Cura file in figure 1 and the result in figure 2 and 3.

Prototypes

Prototypes got printed in downscaled size in order to get an overview. For an example a downscaled Ericsson logo got printed to see that it looked right with the dimensions given in the program. Also the different text that were used for branding got printed for the same reasons as the logo. The models for these can be seen in appendix F and figure 4 and the results can be seen in figure 5.

Support & holder for electrical circuits

The electrical circuits is mounted on top of the flight case and under the case, where the height is limited to 50 mm. The circuits and various components have to be secured to the case so that they stay in place when the case is moved around. Some of the circuits and boards have mounting holes that is used to mount the circuits, but in some cases distances and brackets is needed and were 3D printed to help with the mounting.
3.4. Hardware and electrical circuits

This section explains how the implementation of the functions were done, what hardware used and the connection between the circuits, figure 13 below illustrate a block diagram over the system.

![Block diagram over the system, from software to hardware.](image)

3.4.1. Electrical schematic over the system

In figure 14 the schematic is illustrated with the different boards and components and how they are connected together. The schematic shows the boards and their output pins, for full documentation and schematic over every individually circuit or board it can be seen in the datasheets for each component in the appendix or in the references that is referred to in the report.

![Fritzing electrical schematic over the system.](image)
The system uses a lot of breakout boards and circuit boards, these are explained in the sections below and what function they have and how they were implemented. The system also uses various electronics both passive and active components such as resistor, potentiometer, transistors and antennas, these are used to control different currents and signals/communication going to components and circuits.

3.4.2. Arduino Uno & Mega

Arduino Uno and Mega is both open-source microcontroller boards, see [58] and [59]. The Uno is based around the ATmega328P microcontroller and the Mega is based on the ATmega2560 microcontroller, the specifications for each microcontroller can be seen [60] & [61]. The circuit boards are equipped with both digital and analog I/O pins that is used to communicate with other circuits or components. The Uno has 14 Digital pins where 6 of them can be used as PWM outputs, it also has 6 Analog pins where the Mega is a much bigger board with its 54 Digital pins (12 PWM) and 16 Analog pins.

The Microcontroller board is programmable with the Arduino IDE (Integrated Development Environment) via a type B USB cable and its is programmed with the use of a PC. It is powered with voltages from 7 to 20 volts via a power jack or 5 Volts from the USB port.

The board is used because of its ability to adapt to what is needed in order to control the electrical system. The product is fairly cheap and has great documentation and since its open-source, a lot of information can be found online. The Unos clock speed is at 16MHz and it has power output pins of both 3.3 and 5.0 Voltages.

Even though the Arduino Uno board are provided with a lot of different analog and digital pins as illustrated in figure.22, there are some special pins.

Where the digital pins are:

- Pin 0 is RX and Pin 1 is TX which is Uart data.
- The pins 3, 5,6, 9, 10 and 11 can be used as 8-bit PWM outputs.
- SPI communication can be done via the pins 10 (SS), 11 (MOSI), 12 (MISO) and 13 (SCKL).

The special analog pins are:

- A4 and A5 which supports TWI communication also known as I2C.

Full specification of the Arduino Uno Board can be seen in its datasheet [62].
ATmega328P microcontroller

ATmega328P is a high performance, low power 8-bit microcontroller and its block diagram is shown in figure 3.5. It is based on the AVR enhanced RISC architecture [63] & [64]. The microcontroller is commonly used in embedded system applications, which gives the microcontroller a wide range on systems it can be applied and used in. As mentioned above the microcontroller is used in the Arduino Uno and the pinout from the microcontroller is shown in figure 3.6. and what pin it is connected to is shown in Table 1. The main features why the microcontroller is so popular is that it is a Non programmable data and program memory, it is high performance, low power consumption, it can operate fully static operations, it has on chip analog comparator, its advanced RISC architecture, 32KB flash memory and 2KB SRAM. The microcontroller has an operating voltage between 1.8 and 5 V, its crystal oscillator can operate in speeds up to 20 MHz [65], the microcontrollers full specification can be found in its datasheet. The controller supports SPI, I2C and USI (Universal Serial Interface) and it is using three types of memories [66].

- The flash memory with its 32KB capacity, the flash memory is ROM (Programmable Read Only Memory) which is a non volatile memory.
- SRAM (Static Random Access Memory) with 2KB of capacity. The memory is volatile and when the microcontroller loses power the data stored will be removed and erased.
- The EEPROM (Electrically Erasable Programmable Read Only Memory) which stores the longtime data.
Figure 3.5. Block diagram for ATmega328P.

Figure 3.6. ATmega328P and its pinout.
<table>
<thead>
<tr>
<th>ATmega328P microcontroller</th>
<th>Arduino Uno breakout board</th>
<th>ATmega328P microcontroller</th>
<th>Arduino Uno breakout board</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pin</td>
<td>Indication</td>
<td>Pin</td>
<td>Indication</td>
</tr>
<tr>
<td>1</td>
<td>PC6</td>
<td>Reset</td>
<td>Reset</td>
</tr>
<tr>
<td>2</td>
<td>PD0</td>
<td>Digital Pin 0 (Rx)</td>
<td>16</td>
</tr>
<tr>
<td>3</td>
<td>PD1</td>
<td>Digital Pin 1 (Tx)</td>
<td>17</td>
</tr>
<tr>
<td>4</td>
<td>PD2</td>
<td>Digital Pin 2</td>
<td>18</td>
</tr>
<tr>
<td>5</td>
<td>PD3</td>
<td>Digital Pin 3 (PWM)</td>
<td>19</td>
</tr>
<tr>
<td>6</td>
<td>PD4</td>
<td>Digital Pin 4</td>
<td>20</td>
</tr>
<tr>
<td>7</td>
<td>VCC</td>
<td>Vcc</td>
<td>Voltage</td>
</tr>
<tr>
<td>8</td>
<td>GND</td>
<td>Gnd</td>
<td>Ground</td>
</tr>
<tr>
<td>9</td>
<td>PB6</td>
<td>Crystal</td>
<td>23</td>
</tr>
<tr>
<td>10</td>
<td>PB7</td>
<td>Crystal</td>
<td>24</td>
</tr>
<tr>
<td>11</td>
<td>PD5</td>
<td>Digital Pin 5</td>
<td>25</td>
</tr>
<tr>
<td>12</td>
<td>PD6</td>
<td>Digital Pin 6</td>
<td>26</td>
</tr>
<tr>
<td>13</td>
<td>PD7</td>
<td>Digital Pin 7</td>
<td>27</td>
</tr>
<tr>
<td>14</td>
<td>PB0</td>
<td>Digital Pin 8</td>
<td>28</td>
</tr>
</tbody>
</table>

Table 1. ATmega328P pin connection to the Arduino Uno breakout board pins.

### 3.4.3. Power supply and batteries

The system needs power in order to work and operate. The basic power for the system is 5V that will be provided from the Arduino Uno. The Arduino Uno itself is powered with an AC/DC power supply that is plugged into the wall were the 220V AC is converted to DC and with the help of a step-down transformer down to 12V and 1A, the Uno can be powered with DC voltages between 7-20V according to its datasheet. The power supply used is a I.T.E Power Supply from AMIGO, this is shown in appendix C in misc.12.

The 5V from the Arduino is delivered through the I/O pin marked as 5V, the voltage is provided to the GSM module, GPS module, the LCD display and the temperature sensor. The GPS can also be provided with the 3.3V from the Arduino Uno but the lower voltage signal is more sensitive to noise [67].

25 | Page
The LED Matrix needs its own power source since it uses a lot of current [68]. This is talked about in (3.4.7 Flexible LED Matrix and diodes), a power supply that can deliver 5V DC and at least 2.5 A is used. The power supply used is a Switching Adapter from DVE, this is illustrated in appendix C in misc.11.

When the system is on the move and not able to be connected to a wall adapter, the system still needs power and in this case a Lithium battery will be used. Lithium Ion Batteries is a rechargeable battery where lithium ions move between the negative and the positive electrode. When the battery is discharging the ions move from negative to positive and when the battery is charging the ions move the other way instead, this is illustrated in figure.3.7. Lithium batteries are popular for portable electrical circuits and the batteries have a very high energy density and really low self discharge compared to other battery solutions [69].

The batteries are really popular and is commonly used in laptops, cell phones or other small electrical devices. The batteries are often much lighter than other rechargeable batteries with the same size. The batteries can handle hundreds of charge and discharge cycles and has no memory effect where the battery needs to be fully drained before it can be charged again. The L-ion batteries hold its charges, it only loses around 5 percent of the charges every month, compared to the NiMH batteries that loses 20 percent in the same period of time [70].

The battery will be used to power the Arduino UNO, GSM module and GPS module when the system is not stationary and on the move.

![Figure 3.7. Graphical overview of the Lithium battery.](image)

### 3.4.4. LCD Display

The LCD requires 11 I/O pins from the Arduino Uno. The connection between the Uno and the LCD is done through the digital pins, where 6 pins are needed and 3 is used in PWM mode. The LCD is powered by 5 Volts DC from the Uno. The data transfer and connection between the Arduino Uno and the LCD display is shown in table.2.

Contrast pin (VE) on the LCD Display, this is shown in figure.3.8. It is connected to the 5V from the Arduino through a potentiometer. This is done so that the voltage can be regulated and when the voltage on the LCD:s input (VE) is changed the contrast / backlight on the screen is adjusted, the higher the voltage the greater the contrast is. The connection is shown in the electrical schematic in figure.14 and also in the fritzing model in figure.3.23..
Liquid-Crystal Display is an electronically modulated optical device that uses light-modulating properties of liquid crystals to create a screen [71]. The LCD uses a backlight to light up individual pixels, the pixels are arranged in a rectangular grid and the grid is controlled by the displays chip or microcontroller. The backlight provides even light behind the screen, the light is then polarized by the liquid crystal layer and only some of the light are passed through the layer. The liquid crystal layer are made of a solid base and a then a liquid substance that can be modified or altered when an electrical voltage is applied across the layer. The substance blocks the light when it is turned off and it reflects the standard RGB (Red, Green & Blue) light when it is activated [72].

In this case an LCD display from Sparkfun is used [73] and its datasheet is shown in [74]. The display is a Basic 16x2 Character LCD with white text on black backlight. The display uses a Hitachi HD44780 chip as its brain.

Hitachi HD44780

Hitachi HD44780 is a dot-matrix display controller for LCD displays, the driver supports alphanumeric, Japanese kana characters and symbols. The chip works both in 4- and 8-Bit mode, it can be configured to be controlled by either one of them. A single chip can display up to two 8-character lines. The controller can only be used with monochrome text displays and is commonly used in fax machines, laser printers, networking equipment and industrial test equipment [75].

The specification of the HD44780 can be seen in its datasheet [76] and the block diagram over how the system is connected and how it operates can be seen in figure.3.9.

Figure.3.8. 16x2 LCD Display and its connections.
3.4.5. Radio and telecom communication (GSM)

The GSM module or shield will be used as a breakout board and not as a shield, why this is done is discussed in chapter 5. The GSM module (SparkFun LTE Cat M1/NB) will communicate with the Arduino Uno via Uart and it will be connected to the pins, Tx through digital pin 0 and Rx through digital pin 1, this is shown in table 5. The GSM module is used to access the mobile network and mobile communication, the information the system stores, such as temperature readings, positioning and location will be able to be shared via IP and server solutions.

The SparkFun LTE CAT M1/NB-IoT Shield [77] is an electrical board and it is shown in figure 28, it is equipped with a SARA-R4-R410M-02B LTE Cat M1/NB-IoT modem that provides a low power wide area network technology that provide cellular communication to Iot devices [78]. It works on standard LTE network bands just like the ones used in most smartphones, it is supported by almost every cellular network carrier. The module communicate via UART and simple AT commands. The board can be used to send text messages and communicate with servers through TCP and IP connections. The board also supports I2C interface for GPS communication. The board has a built in antenna but it also has a u.FL connection for connecting an external antenna if needed [79].

The SARA-R4:s interface can be configured via a micro-B USB cable. The module is powered by either a lithium battery, 5V pin from the Arduino or through the 3.3V USB port. There is a little switch on the board where the user choose if the arduino should send power to the module or if the module should send power to the arduino. This can be in good use if the user wants to power both boards with a battery. The board supports various GPS boards that are made from the same provider and they are connected via qwiic connection [80] that communicate with I2C protocol.
SARA-R4-R410M-02B

SARA-R4-R410M-02B is a cellular module created by U-blox. It is a compact LTE Cat M1/NB EGPRS module that works globally. The module can be setup to operate in different modes [81].

- LTE Cat M1 which is a LPWA (Low Power Wide Area) interface that connects IoT devices at a data speed of medium rate, which is in half duplex mode around 375 kb/s in upload and download speed. This is mostly used by network operators in the US [82].
- LTE Cat NB1 is slower than M1 with a speed of around 27.5 - 62.5 Kb/s, but the narrowband NB1 is better to be able to penetrate into buildings and it provides longer battery life and is easier to implement. It is being used in Asia, Pacific region and Europe [83].
- EGPRS also known as EDGE is a mobile phone technology which is a data transmission rate that is backward-compatible extension of GSM. EDGE is a pre-3G, 4G radio technology [84] & [85].

The module has a low-maintenance, low cost, low power consumption and has a long life (expected to last 10 years according to datasheet). The block diagram on how the module works is illustrated in figure.29 and its full specification can be seen in its datasheet, see [86].

![Figure.3.10. SparkFun LTE CAT M1/NB - Iot Shield](image)

![Figure.3.11. Sara-R4 Module block diagram](image)
The GSM module need to have a SIM Card connected in order for it to work and the GSM needs an LTE antenna as well. The GSM board has an antenna already mounted on it, but this antenna is bypassed and an external antenna is used via the u.FL connection, why the onboard antenna is not used is talked about in chapter 5.

SIM Card or Subscriber Identity module is an IC circuit that is supposed to store the users IMSI number. The SIM Card is used to operate the GSM module, it is used for mobile communication and to be able to access the telecom network from all around the world [87]. Pinout for the SIM Card is illustrated in figure.3.12. and a SIM from Telia is provided by Ericsson.

![Sim Card Pinout](image)

Figure.3.12.. Sim Card Pinout.

In this project a LTE Antenna - DA-24-04-SMR [88] is used, it is shown in figure.3.13.. Which has a SMA connection but it is reversed polarized which means that an adapter is needed for it to be connected to a standard SMA connection, a female to male adapter is needed [89]. The antenna operates at 2.4 GHz and it operates in the frequency range of 2400 - 2500 Mhz with a gain of 2.2dBi. The antennas specification can be seen in its datasheet [90].

The adapter used is a Reversed Polarized SMA Connector, the adapter converts the standard SMA male connector to female and vice versa.

![DA-24-04-SMR Duck Antenna](image)

Figure.3.13.. DA-24-04-SMR Duck Antenna.
3.4.6. GPS

The GPS module (SparkFun GPS-RTK - Neo-M8P-2) [91] is used in this project and it is shown in figure 3.14. The module is used for GPS location tracker. The data collected will be delivered to the GSM shield and from the shield to the Arduino. The GPS module communicate with the GSM via I2C through the qwiic connectors on both the GSM Board and on the GPS Board.

The SparkFun GPS-RTK Board is a breakout board based on the U-blox Neo-M8P-2 module that is a high accuracy component for GPS location tracking such as RTK. The accuracy is so precise that it can measure a position with an error of less than 2.5 centimeters according to the documentation [92].

The board is equipped with a backup battery that keeps the latest configuration and data for up to two weeks, the battery is also rechargeable.

The module Neo-M8P-2 can be configured to operate in many different ways. Variable I2C addresses, geofencing, different update rates and the RTK can be increased to operate at 4Hz. The module has four different types of communication options [93]:

- USB - which works as a COM port.
- UART - works via 3.3 Volt TTL.
- I2C - communicates through I2C with the qwiic connectors.
- SPI - SPI communication is viable, but then I2C and UART is disabled.

The GPS board has a u.FL connector and an antenna is needed to be connected there to be able to communicate with the satellites to achieve the GPS positioning.

Neo-M8P-2

Neo-M8P-2 is a high performance module with a positioning engine from U-blox M8 series that uses RTK (Real Time Kinematic) technology [94]. The module uses precise RTK and GNSS technology to be able to decide a objects positioning with tolerances of around two centimeters. The IC is designed to be used when accurate guidance is used such as in unmanned vehicles and other machined controlled applications. The module uses RTCM (Radio Technical Commission for Maritime Services) to send correction data to the rover module via a communication link, this is how the module can send positioning data with such an accuracy. The module is often used in applications where an object needs to be able to move fast, accurate and operate efficiently, examples of that can be moving robots that need positioning guidance, robotic lawn mower is a good example. Neo-M8P-2 are compatible with many different types of communications, such as wifi, bluetooth and cellular.

The module can receive different types of signals from different types of satellites according to the datasheet [95]:

- GNSS - it can receive and track multiple GNSS systems, it is default configured to concurrent GPS and GLONASS receptions.
- GPS - The module can receive and track L1C/A signals provided at 1575.42 MHz given from the GPS (Global Positioning System).
- BeiDou - The module can process and receive B1I signals at 1561.098 MHz from the BeiDou navigation satellite system. If the BeiDou signal is used in parallel with the GPS signal the result is a high coverage and improved reliability for better accuracy, but BeiDou is not available globally and is only available in Chinese region, the global coverage is said to be open in 2020.
- GLONASS - it can receive and process GLONASS signals with the GPS, it can track the L1OF signal at 1602 MHz + k * 562.5 kHz, k = the satellites channel frequency number.
The block diagram for the module is shown below, figure.3.15, and the module's full specification and data can be seen in the datasheet.

The GPS board doesn't come with an antenna installed and a powerful embedded antenna is needed to be used to ensure that the module can have reliable communication with the satellites.

A GPS/GNSS Magnetic Mount Antenna with a 3 m cable is used in the project. The antenna is designed to work with both GPS and GLONASS receptions [96]. Both GPS and GLONASS is explained above in Neo-M8P-2. The antenna uses standard male SMA connection, so no adapter is needed as in the GSM antenna, the antenna is shown in figure.3.16, and the specification for the antenna is shown in Table.2 since the datasheet never were provided by the manufacturer.
**Figure 3.16. The GPS/GNSS Antenna.**

<table>
<thead>
<tr>
<th>GPS/GNSS Magnetic Mount Antenna specifications:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dimensions: 50 x 38 x 17 mm</td>
</tr>
<tr>
<td>Weight: 75 g</td>
</tr>
<tr>
<td>Frequency Range: 1575 - 1610 MHz</td>
</tr>
<tr>
<td>GPS Center Frequency: 1575.42 MHz</td>
</tr>
<tr>
<td>GLONASS Center Frequency: 1602 MHz</td>
</tr>
<tr>
<td>LNA Voltage: 3 - 5 V DC</td>
</tr>
<tr>
<td>LNA Gain: 28 dB</td>
</tr>
<tr>
<td>LNA Current: 10 mA</td>
</tr>
<tr>
<td>Termination Connector: SMA male</td>
</tr>
<tr>
<td>Impedance: 50 Ohm</td>
</tr>
<tr>
<td>Right hand polarization: yes</td>
</tr>
<tr>
<td>Cable length: 3 meter</td>
</tr>
</tbody>
</table>

Table 2. The specification for the GPS/GNSS Antenna.

**3.4.7. Flexible LED Matrix and diodes**

The LED Matrix Panel uses the clever Neopixel libraries [97] in the Arduino IDE and this makes it possible to control the panel using a single digital output from the Arduino. As mentioned above the LED panel consists of 256 individually addressable LED:s, each LED draw current and the whole panel will use a lot of current. This means that even if the Panel operate on 5V it can't be powered by a standard 5V power source, the power source needs to be able to handle the high currents. So the Panel gets powered from a stand alone power supply that is rated at 5V and at least 2.5A.
Various diodes will be mounted throughout the case and these are controlled by one digital pin from the Arduino and a BJT circuit, this is presented in schematic 1 in appendix D.

LED Matrix or Display is based on multiple LED:s connected together to create a screen with the ability to change colour and display text, images and video. The more LED:s used the better resolution the screen can provide, each LED represents a pixel. In this case the LED Matrix is flexible to be able to adapt to the surface where it is supposed to be mounted.

The LED Matrix used is an 8x32 pixel screen with every single led individually addressable. This enables the ability to create different scenes and illuminating images. The Matrix is powered with 5 Volts DC and it uses a lot of current, since it is powering 256 RGB LED:s. The datasheet specifies that the nominal rating for current is around 5 A and max current can reach up to 15 A when all LED:s is set to bright white [98]. This means that the LED Matrix can’t be powered by the Arduino and require its own power source with the ability to handle the high currents. The LED Matrix has JST SM connectors with three cables, where one is power, one is data signal and the last one is ground or com for the signal, this is illustrated in figure 3.17.

The LED Matrix is build with the WS2812B LED:s and controlled by the WS2811 IC drivers, these components are explained below.

![Figure 3.17. Flexible LED Matrix with its connections showing.](image)

WS2812B

WS2812B is an intelligent control LED integrated light source, it is a 5050-package containing both the control circuit and the RGB chip. It is not only SMD LED, it has an integrated control circuit embedded inside. The brightness of each color can be adjusted by the use of a serial string to control the brightness in 256 different ways, this gives the user the ability to choose which color they want. The LED can display 16 777 216 different types of colors. The LED has low voltage, high brightness, the light emitting angle is wide, low power and long life. The WS2812B is overall good and reliable, its pinout is shown below, figure 3.18. Where VDD is the power pin, VSS is the ground pin, DI is data input pin and DO is data output pin. The LED:s full specification can be seen in its provided datasheet, see [99].

WS2811

WS2811 is a signal line 256 Gray level 3 channel constant current LED driver IC, its specifications can be seen in the datasheet [100]. The chip is internally connected in such a way that it will reshape and amplify the signal that passes through the data latch on the digital port. It also has a precision oscillator and a 12V programmable constant current output driver. The chip uses single NZR (No-return-to-zero) communication mode.
The pinout for the WS2811 can be seen in figure.3.19, and the pins are:

- **VDD** - Power supply voltage
- **SET** - Sets the work mode for the IC, either low speed or high speed.
- **DIN** - Data signal input.
- **DO** - Data signal cascade output.
- **GND** - Ground
- **OUTB** - Output for the Blue PWM control.
- **OUTG** - Output for the Green PWM control.
- **OUTR** - Output for the Red PWM control.

LED stands for Light Emitting Diode and it is a diode that emits light when currents flow through it. The semiconductor contains electrons and when they are recombined with electron holes energy is released in the form of photons and this is the effect of electroluminescence. The colour of the light emitted is determined by the energy that is released by the photons. The photon energy determines the wavelength of the light and different wavelength of the light leads to different colors emitted [102]. An overview of the LED and its connectors anode and cathode is illustrated in figure.3.20.
3.4.8. Temperature sensor

The Analog TMP36 is a temperature sensor that can measure temperatures between -40 to +125 degrees Celsius with an accuracy of (+/-) 1 degrees Celsius according to the datasheet [103]. The sensor has three terminals, these are shown in figure 3.21, where one is “ground”, one is “voltage in” and one is “voltage out”. The “voltage in” is the power source to the sensor, it can be powered with voltages between 2.7 and 5.5 V. The “voltage out” is the signal with the measured temperature, the ratio between degrees celsius and voltage out is linear and corresponds to 10 mV per degree celsius, this is illustrated in figure 3.22. The operating voltage is 750 mV at 25 degrees celsius.

![Figure 3.21. TMP36 sensor and its terminals.](image)

![Figure 3.22. Graph for the correlation of degrees celsius and voltage out, line “b” is for the TMP36 sensor.](image)

3.4.9. Fans and cooling

Fans will be mounted on the back lid of the Case and these will be powered by 5V and a BJT switch controlled from the arduino. No PID-control or any control, just an on or off switch is implemented, fans will run on full speed when turned on.

3.4.10. Connections

This section show a graphical schematic over the system and this section also presents tables on how the different circuit boards and components are connected. Where each pin is connected and what type of data is sent or received. Below two fritzing schematics are presented, figure 42 and 43. These show the electrical system and how each circuit or component is connected to each other. Figure 42 is
slightly modified in paint to include the components that couldn't be implemented in Fritzing and some of the boards aren't identical to the once used but the size and connections are the same.

Figure.3.23. Fritzing graphical schematic over the electrical system and components used.

Figure.3.24. Fritzing graphical schematic over the first version of the electrical system.
<table>
<thead>
<tr>
<th>Pin</th>
<th>Description</th>
<th>Pin</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A0</td>
<td>Analog signal - Input from TMP36</td>
<td>0</td>
<td>Tx - Rx on GSM Shield</td>
</tr>
<tr>
<td>A1</td>
<td>*</td>
<td>1</td>
<td>Rx - Tx on GSM Shield</td>
</tr>
<tr>
<td>A2</td>
<td>*</td>
<td>2</td>
<td>Data 7 LCD</td>
</tr>
<tr>
<td>A3</td>
<td>*</td>
<td>3</td>
<td>PWM Signal - Data 6 LCD</td>
</tr>
<tr>
<td>A4</td>
<td>I2C SDA -</td>
<td>4</td>
<td>Data 5 LCD</td>
</tr>
<tr>
<td>A5</td>
<td>I2C SCL -</td>
<td>5</td>
<td>PWM Signal - Data 4 LCD</td>
</tr>
<tr>
<td>Power</td>
<td>Power</td>
<td>6</td>
<td>LED Matrix</td>
</tr>
<tr>
<td>Vin</td>
<td>Power</td>
<td>7</td>
<td>LED Warning</td>
</tr>
<tr>
<td>GND</td>
<td>LED Matrix &amp; TMP36</td>
<td>8</td>
<td>LED Warning</td>
</tr>
<tr>
<td>GND</td>
<td>GND</td>
<td>9</td>
<td>LED Warning</td>
</tr>
<tr>
<td>5V</td>
<td>TMP36, LCD, GSM &amp; GPS</td>
<td>10</td>
<td>SPI SCK -</td>
</tr>
<tr>
<td>3.3V</td>
<td>*</td>
<td>11</td>
<td>SPI MISO PWM - Enable LCD</td>
</tr>
<tr>
<td>Other</td>
<td></td>
<td>12</td>
<td>SPI MOSI - Register select LCD</td>
</tr>
<tr>
<td>Reset</td>
<td>*</td>
<td>13</td>
<td>SPI SS -</td>
</tr>
<tr>
<td>IORef</td>
<td>*</td>
<td></td>
<td>GND LCD, GSM</td>
</tr>
<tr>
<td>ARef</td>
<td>*</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* = I/O not in use

Table.3. The Arduinos pinout and connections.
= Not in use in 4-Bit mode, $ = Data transfer between MPU and HD44780U

<table>
<thead>
<tr>
<th>Type of Signal</th>
<th>LCD Display output</th>
<th>I/O</th>
<th>Description</th>
<th>Arduino Uno</th>
</tr>
</thead>
<tbody>
<tr>
<td>GND</td>
<td>1</td>
<td>Gnd</td>
<td>Ground potential</td>
<td>GND</td>
</tr>
<tr>
<td>Vcc</td>
<td>2</td>
<td>Power</td>
<td>5V DC</td>
<td>5V</td>
</tr>
<tr>
<td>VO</td>
<td>3</td>
<td>D0</td>
<td>contrast</td>
<td>5V through the potentiometer</td>
</tr>
<tr>
<td>RS</td>
<td>4</td>
<td>DO</td>
<td>Register select</td>
<td>D12</td>
</tr>
<tr>
<td>R/W</td>
<td>5</td>
<td>Gnd</td>
<td>Read / Write</td>
<td>GND</td>
</tr>
<tr>
<td>Clk</td>
<td>6</td>
<td>DO</td>
<td>Clock (Enable)</td>
<td>D11</td>
</tr>
<tr>
<td>$</td>
<td>7 - 10</td>
<td>$</td>
<td>$</td>
<td>$</td>
</tr>
<tr>
<td>PWM</td>
<td>11</td>
<td>DO</td>
<td>$</td>
<td>D5</td>
</tr>
<tr>
<td>Data</td>
<td>12</td>
<td>DIO</td>
<td>$</td>
<td>D4</td>
</tr>
<tr>
<td>PWM</td>
<td>13</td>
<td>DO</td>
<td>$</td>
<td>D3</td>
</tr>
<tr>
<td>Data</td>
<td>14</td>
<td>DIO</td>
<td>$</td>
<td>D2</td>
</tr>
<tr>
<td>Backlight +</td>
<td>15</td>
<td>Power</td>
<td>Backlight anode</td>
<td>5V</td>
</tr>
<tr>
<td>Backlight -</td>
<td>16</td>
<td>Gnd</td>
<td>Backlight cathode</td>
<td>GND</td>
</tr>
</tbody>
</table>

Table 4. The LCD:s outputs and connection to the Arduino.

<table>
<thead>
<tr>
<th>Type of Signal</th>
<th>LTE CAT M1/NB</th>
<th>I/O</th>
<th>Description</th>
<th>Arduino Uno</th>
</tr>
</thead>
<tbody>
<tr>
<td>GND</td>
<td>Gnd</td>
<td>Gnd</td>
<td>Ground potential</td>
<td>GND</td>
</tr>
<tr>
<td>Vcc</td>
<td>Vcc</td>
<td>Power</td>
<td>5V DC</td>
<td>5V</td>
</tr>
<tr>
<td>Uart</td>
<td>Tx</td>
<td>DO</td>
<td>Uart Output signal</td>
<td>D0 (Rx)</td>
</tr>
<tr>
<td>Uart</td>
<td>Rx</td>
<td>DI</td>
<td>Uart Input signal</td>
<td>D1 (Tx)</td>
</tr>
</tbody>
</table>

Table 5. The LTE CAT M1/NB:s pins and connection to the Arduino.
### Table 6. The GPS-RTK - Neo-M8P-2 pins and connection to the LTE CAT M1/NB.

<table>
<thead>
<tr>
<th>Type of Signal</th>
<th>Flexible LED Matrix</th>
<th>I/O</th>
<th>Description</th>
<th>Arduino Uno</th>
</tr>
</thead>
<tbody>
<tr>
<td>GND</td>
<td>Yellow - GND</td>
<td>Gnd</td>
<td>Ground potential</td>
<td>*</td>
</tr>
<tr>
<td>Vcc</td>
<td>Red - DC5V+</td>
<td>Power</td>
<td>Power supply 5V, 5A DC</td>
<td>*</td>
</tr>
<tr>
<td>Data</td>
<td>Green - DAT</td>
<td>DIO</td>
<td>Serial Data line for led control</td>
<td>D7</td>
</tr>
<tr>
<td>COM</td>
<td>Yellow - COM-</td>
<td>Gnd</td>
<td>Serial Data ground</td>
<td>Digital GND</td>
</tr>
</tbody>
</table>

### Table 7. The Led Matrix pins and connection to the Arduino, colour codes from figure 3.17.

<table>
<thead>
<tr>
<th>Type of Signal</th>
<th>Flexible LED Matrix</th>
<th>I/O</th>
<th>Description</th>
<th>Arduino Uno</th>
</tr>
</thead>
<tbody>
<tr>
<td>GND</td>
<td>Yellow - GND</td>
<td>Gnd</td>
<td>Ground potential</td>
<td>*</td>
</tr>
<tr>
<td>Vcc</td>
<td>Red - DC5V+</td>
<td>Power</td>
<td>Power supply 5V, 5A DC</td>
<td>*</td>
</tr>
<tr>
<td>Data</td>
<td>Green - DAT</td>
<td>DIO</td>
<td>Serial Data line for led control</td>
<td>D7</td>
</tr>
<tr>
<td>COM</td>
<td>Yellow - COM-</td>
<td>Gnd</td>
<td>Serial Data ground</td>
<td>Digital GND</td>
</tr>
</tbody>
</table>

### Table 8. The TMP36’s pins and connection to the Arduino.

<table>
<thead>
<tr>
<th>Type of Signal</th>
<th>TMP36</th>
<th>I/O</th>
<th>Description</th>
<th>Arduino Uno</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vcc</td>
<td>1</td>
<td>Power</td>
<td>5V DC</td>
<td>5V</td>
</tr>
<tr>
<td>Input</td>
<td>2</td>
<td>AO</td>
<td>Analog output signal</td>
<td>AI</td>
</tr>
<tr>
<td>GND</td>
<td>3</td>
<td>Gnd</td>
<td>Ground potential</td>
<td>GND</td>
</tr>
</tbody>
</table>

### 3.4.11. Soldering

Soldering were done for the connection between the Arduino and the LCD, the connection between the Uno and the GSM and TMP36 sensor signal to the Uno. Cables were also soldered to the pins on the potentiometer and also resistors were soldered onto the cathode leg on the rgb leds, where they were soldered with an individual cable to the anode and a common ground cable. Power cables were
soldered from the power supply to power the LED Matrix. This can be seen in Appendix C, in misc.2-8.

### 3.5. Software and Programming

This section explains how the implementation of the functions were done, how the software and code were written. Why and what type of software that were used is also explained.

Software is used to design and test the structures, draw model schematics, electrical circuit schematics and also software is needed to control all the electrical circuits and boards. The smart functions such as GSM, GPS is configured via software on a PC. The software used in this project is:

**AutoDesk Inventor**

AutoDesk is a american software company that makes services and software solutions for the engineering, architecture, manufacturing, construction, entertainment industries and the media areas.

AutoDesk Inventor is one of their software and it is a CAD program. Inventor has been used in this project to model three-dimensional objects such as the case and all its various components around it. Every component have also been converted into a 2D drawing. All prototypes have been designed in Inventor and all the files for the 3D printing have been modified in the software as well.

**Fritzing**

Fritzing is a open-source software. Quoted from their website [104], “We offer a software tool, a community website and services in the spirit of processing and arduino, fostering a creative ecosystem that allows users to document their prototypes, share them with others, teach electronics in a classroom, and layout and manufacture professional pcb’s”.

The software has been used to draw graphical schematics over the electrical system and how the different components and electrical boards should connect and communicate with each other.

The software is really simple to use and is mostly only used to get an overview over the system. For electrical simulations and measurements, a different program or software would have been preferred.

**Arduino IDE Software**

Arduino Integrated development software is a cross-platform software coded in Java that works in multiple operative systems, it works in Windows, Linux and MacOS. The software is open-source and is created by the company named Arduino. The company is an open-source hardware and software company. The Arduino community is a big community with a lot of activity on online forums where people discuss different ideas and engineering solutions. Since everything is open source, everyone can upload they’re Arduino project with its schematics and code. The Arduino IDE code is called a “sketch” and can be uploaded to their community [105].

Arduinos website has a lot of information and tutorials and simple sketches for commonly used electrical boards and arduino shields. An electrical board can easily be tested by just plugging it into a computer and uploading a code from their website, then the code can be tweaked to fit the users need. For bigger project multiple samples codes can be combined to create a bigger control system.

The software is downloaded from their website and runs on a computer where the code is written. The software is really simple and the code or sketch can be divided into four different sections.
The first part is libraries that is imported to the program, this is explained below in (Arduino compatible libraries). The second part is the creation of variables, numbers or strings that are going to be used in the code, these have to be defined in the beginning of the program just like in every other coding language. The third part is where the setup is done, this code executes one time once the program is started. For example the data rate in bits per second (baud) is set, this is the speed for the serial data transmission. The configuration is done in the setup, after importing libraries they need to be configured and settings need to be changed in the setup. The last part is the main loop, in this loop the program runs all the time as long as the code does not have any delays. Any type of calculation or algorithm is written in this loop. As an example is if a temperature is measured its signal is provided to an analog input on the arduino board and then the algorithm for converting the signal in volts to temperature in degrees celsius is ran in the loop.

In the second part of the I/O are defined. Here the user tells the program on which I/O pin it should send or receive data. The Arduino board such as Uno or Mega has both digital and analog I/O pins, where the digital is a signal that is referred to high or low and the analog pin can read voltages between 0 and 5 Volts. The Arduino boards have the ability to use some of its digital pins as PWM signals, this is explained in section 2.5. PWM.

When first starting the sketch the user choose which board is used. If a Arduino Uno is used then Uno is selected in the settings and then the program knows how many pins the board has and which pin can be used for what operation.

Arduino compatible libraries

The IDE software can be extended with the use of libraries. Libraries are imported to the program and they provide extra functionality, the libraries can be used to manipulated data or used when communicating with hardware [106]. The IDE software comes with already installed libraries that are commonly used and then the user import the libraries that will be needed to control and operate the system. The libraries that come installed are written by Arduino them self, but since its open-source anyone can write new libraries and share with the community.

Standard libraries that is used in this project are:

- GSM Library - this is used to connect the GPRS/GSM network to the hardware.
- LiquidCrystal Library - this is used to control the standard 16x2 LCD Display.
- SPI Library - this is used for SPI data communication.
- SoftwareSerial Library - this is used for serial communication on any digital pin on the Arduino board.
- Wire Library - this is used for TWI / I2C communication for sending and receiving data.

Community provided libraries used are:

- TinyGPS Library - this is used for GPS tracking [107].
- Adafruit_NeoMatrix - this is used to control the LED Matrix [108].
- Adafruit_NeoPixel - same as above [109].
- FastLED - this is used for demo and animation patterns in the LED Matrix [110].
- Adafruit-GFX - Used by the NeoMatrix library [111].
- SparkFun LTE Shield Arduino Library - used to control the GSM [112].

LtSpice

LtSpice is a software for creating electrical circuits, schematics and simulations. The software is created by Linear Technology and is free for everyone to use. LtSpice is referred as a high performance SPICE simulation software with waveform viewer for analog electrical circuits [113].
The program works in such a way that the user draws electrical schematics and puts out electrical components from provided libraries. Once a schematic is drawn the user sets values and configure the components to its desired values.

Both DC and AC analysis can be simulated in the program, the simulations is often used to test filters and check if the circuit is providing the desired output when it is fed with a set input.

In this project LtSpice have been used to test different voltage dividers and BJT transistor solutions.

3.5.1. Arduino Uno

The Arduino Uno is used as the main part in the system, in the software the user choose UNO in the settings and then the I/O needs to be configured. What pin is supposed to send or receive what type of data is configured. At what speed the system should operate and the speed for data transmission is set. The PC communicate with the Arduino Uno via a USB to type A/B cable. The settings for using the Arduino Uno in the IDE software can be seen in appendix E, figure.E-1.

3.5.2. LCD Display software

The LCD communicate with the Arduino on 6 digital output pins that all have to be defined, this is done by giving constants numbers, where the number represents what I/O pin is used. The “LiquidCrystal” Library is imported and the function “LiquidCrystal lcd” is used where the constant is used to select which pin it should operate on. The function from the library handle how each I/O should operate, example if the I/O should be high/low or if it should work as a PWM output. In the setup the size of the LCD display is defined and in this case the LCD have the size 16x2 and this is done by the function “lcd.begin”. The lcd.print function is used to print texts onto the display defined display connected to the outputs from the Uno.

This can be seen in appendix E, code.2, where a test code is shown. This code was written to test the LCD under the development process and the LCD implemented with the temperature sensor code can be seen in figure.E-10.

3.5.3. Mobile communication software

The Mobile communication or GSM is using software provided by SparkFun. The Library imported and used in the IDE software is called “SparkFun LTE Shield Arduino Library”. The example code named “00_Register_Operator” were used and modified. The code starts the GSM module on the breakout board and it configure its settings and what pins it should send and receive data on. The APN and number of operators were modified in order for it to work with the provided telia sim card.

The code initialize the LTE Shield and sets the mobile network operator, then it sets the APN. After that the user has the option to scan for available mobile network to connect to, then the user gets to choose what network to connect to among the ones found. When this is done the configuration of the GSM module is done and it can now be used to send and receive data over the telecom network, this can be seen in appendix E, code 3 and 4.

The example code “01_SMS_Send” were modified and then used to send text messages from the module. Since the module is already configured and connected to a network this code only has to decide where the text message should be sent to. The destination number is modified to the desired receiver. The user gets to type what the text message should be in the serial monitor. Tests of this can
be seen in appendix E, code 5 and 6. Where code.6 shows the text messages sent and received on a mobile phone.

The code uses the library and its function “lte.sendSMS” with the variables text message and destination number.

3.5.4. GPS location software

The GPS code were first written to use another GPS module and were never tested with the hardware. The code used TinyGPS library and SoftwareSerial, the code were going to work with the NEO-6M GPS module and it were using the functions from the library. The code can be seen in appendix E, figure.E-7.

Since the GSM module came with the qwiic connection that uses I2C to communicate with another module were used. This made it possible to use the “SparkFun LTE Shield Arduino Library” and the example code “05_GPS_Request” and “06_GPS_GPRMS” were used. This never worked and it was decided to not be implemented in the end, why is written about in the discussion. The code can be seen in appendix E, figure.E-8.

3.5.5. Temperature reading software

The temperature code can be seen in appendix E, figure.E-9. The code creates three variables named: sensePin, sensorInput and temp, where the first two are integers and the last double. The standard function “analogRead” is used to store the value of the signal that is connected to the analog input on the Arduino Uno and in this case on the first analog I/O (A0). The value read on the analog input is an analog signal and have to be converted to digital inorder to be used in the code. The analog number is converted to a digital 10-Bit number by the Arduino Unos ADC.

The code takes the number from the ADC and divide it by 1024 because of 10-Bit, 1024 is the number of values the 10-Bit sequence can represent [114].

The value is then multiplied by 5 since 5V is used as the reference, an offset from the signal is subtracted and the value is multiplied by 100 to get the value in degrees.

The function “Serial.print” is used to print a text and the value stored in the variable “temp”.

3.5.6. LED Matrix panel and diodes software

The Adafruit_NeoMatrix library is used and the example code “matrixtest” is modified and used. The code is written so that it prints a message on the matrix.

The code uses the function “Adafruit_NeoMatrix” and sets the number of led:s, how many on each row and how many rows and in this case it’s a 8x32 LED Matrix. The matrix is made up of RGB LED:s and in the code the user can choose between 16 777 216 different colours by mixing the “RGB” values.

The brightness on how much light the LED:s should illuminate can be set by altering the value used by the “matrix.setBrightness” function, were the values is in percentage from 0-100 percent. The delay alter how fast the text should scroll by on the Matrix. This can be seen in appendix E, code 11.
3.6. Troubleshooting

3.6.1. Specifications for the Case

The dimensions for the case suit the specification and the drawings were finished for the prototype case. A full size case were never created in this project, but downscaled versions were created and they were working as expected and as intended. It looks professional and gives a clean and serious impression.

3.6.2. Electrical circuits working as intended

The electrical system were not assembled together but every part of the system were tested individually and in some cases together and everything worked as intended except after some troubleshooting and modification in both hardware and software except the GPS part.

3.6.3. Supervisors approval

The supervisor made the decision to not create a full size prototype of the case at this time, but all the files, drawing and electrical system with documentation were handed over to be able to create it in the future when there is time for it. The supervisor is happy over the result and this project have been both stressful and fun for both of us.

3.7. Assembly and Construction

3.7.1. Assembling the Case

The prototype scale of the case were created and assembled together, this can be seen in appendix F, figure 8-10, 12 & 14 - 19.

3.7.2. Electrical circuits assembly

The electrical system can be seen in figure.3.23. and in appendix C.
Chapter 4

4. Results

The result was delimited from the original plan, this was due to delays in orders, shipments and a time limit.

Everything software related to the case itself got finished, 2D and 3D models for the new prototype case were created and also 2D drawings for each part. The 3D models created can be seen in appendix B and all the related drawings can be seen in appendix A, these were later used to 3D print a scaled prototype of the case.

The electrical system and everything hardware related to it got finished and can be seen in appendix C.

The hardware were tested and cables and components was soldered together, some breakout boards needed cables soldered onto them and some came with connections already mounted.

Misc.2-8 in appendix C shows the components or parts of the system that got soldered or cables soldered onto them, everything from power supplies to active or passive components. Misc.9-18 shows the components used in the system and also the whole system mounted together with every hardware connected to each other.

The electrical system was tested mostly with each component or part of the system individually, but in some cases some parts were tested together. The complete system with all different parts working simultaneously were never finished due to time limit. The hardware is controlled by software and in this case it is a code or algorithm written in the Arduino IDE software that is uploaded to the Arduino Uno circuit board.

The software were finished for each individually part of the system except for the GPS system. The software was mostly written and used for controlling each part one by one and not simultaneously. But some systems got implemented and tested together. The Arduino code can be seen in appendix E, where each code used is presented.

Code.3-6 is used for the GSM part of the system. From the configuration part to the network connection and lastly the GSM module sending a text message from the Arduino Uno to a mobile phone via the GSM module and its connected antenna.

In appendix F the results from the 3D printing is presented. The prototype case modelled in CAD got scaled down to the size of 10:1 and then got sliced in a software called Cura to be able to be used by the 3D printer (Creality 10s Pro).
Chapter 5

5. Discussion

A lot of decisions that have been made during the project have evolved around, that the prototype case should be able to be replicated easily by someone that doesn't have the understanding or skills to make or create everything from scratch and due to time limit. Hence why completed electrical circuits or boards have been bought instead of being made. In today's market it can be more cost efficient to buy a finished product or part of a product then creating it yourself in small quantities.

The fact that the Case should be able to travel all around the world, the project had to adapt to worldwide logistic principles and laws. Materials used for example needed to be materials that were allowed to enter and leave other countries without any interference, hence why no wood could be used in the project. Same goes for batteries, when travelling by air batteries it is a big no-no. Therefore batteries couldn't be bought from another country and had to be bought here in sweden and the Case can not be shipped with a battery connected during flights.

The design of the Case were designed and made with specifications in mind that would make it easy to assemble, disassemble and work with. The radio equipment inside shouldn't interfere with the case. The case was designed to be built around a 12 U Flight case and to be mounted on top of it. The front and back lid were designed to be able to be opened or even removed if needed. A solution was designed on how to mount the case to the flightcase and the support beams were modelled in CAD, the same goes for the railing that is supposed to be mounted on the top of the case. The railing was implemented in order to be able to stack heavy equipment or stuff on top of the flightcase without damaging the case, the railing is also taller than the “spoiler” of the case to protect it.

The aesthetic design of the case were done in many interactions, the final design was inspired by airplanes and sports car designs and curves. The design is simple, looks clean and give a serious and modern expression.

The reasoning behind why the size of 12U were chosen for the flightcase was specified by the department, they were going to slim down the existing 20 U cases and compress the equipment and pack it into smaller sized cases and at the same time change the appearance of the cases.

Every function implemented in the electrical system was implemented to fulfill a certain purpose, it could be either to ease the use of the equipment or for safety reasons. But some functions implemented was to make the case stand out and bring attention.

The Case have the company and the department name and logo displayed. If a product should be used professionally by ericsson and displayed with the company name, branding have to follow specific rules and specifications. The text font, placement and size have all to match the company policy. As an example when designing and implementing the Ericsson logo on the case in the 3D program it had to be precise and the angle for the logo had to be right with a tolerance of three decimals.

The electrical components used have been studied and research have been made and components that matched the desired specification have been used. The Arduino Uno was used as the brain and main component of the system and the Uno was chosen over the Mega due to the faster logic controller and smaller size, the amount of I/O on the Uno was enough to cover the systems needs. The temperature reading was done with the help of an analog sensor instead of a digital one in order to not interfere with the numbers of I/O on the breakout board.
What type of data transmission or Internet protocol used were based on the signal or data sent and on how important or often it needed to be updated.
Chapter 6

6. Conclusions

The project as a whole have been educational. Both within the technical difficulties, but also to take responsibility, control and to follow up a larger project from start to finish. Although it is based on electrical engineering and other engineering fields, it has been instructive to work within a large company and get an insight and experience on how work and projects is executed.

Delays in deliveries, miscommunication and problem with outsourcing work, all these obstacles have been encountered during the project and many more. But every obstacle have had a solution to it.

The project was finished but delimated from the original plan, but the end result i am proud of. The end result is a 3D printed scaled model of the 3D modeled case, The CAD Drawings, The software developed and used and also the electrical system, both hardware and software.
References


Appendices

A. CAD Drawings

Figure A-1. Wheel assembly for the 12U Flight Case.

Figure A-2. Base Plate for the 12U Flight Case.
Figure A-3. Locking Mechanism for the 12U Flight Case.

Figure A-4. The metal shield for the locking mechanism for the base of the 12U Flight Case.
Figure A-5. The metal shield for the locking mechanism for the lids of the 12U Flight Case.

Figure A-6. The Base of the 12U Flight Case.
Figure A-7. The Lids of the 12U Flight Case.

Figure A-8. The Assembled 12U Flight Case.
Figure A-9. The Base of the Case with the Ericsson and Demo & Event Branding.

Figure A-10. The front lid of the Case with the Ericsson Branding.
Figure A-11. The back lid of the Case.

Figure A-12. The Ericsson and Demo & Event Branding on the Case.
Figure A-13. The lower support beam for the Case, connecting the Case to the 12U Flight Case.

Figure A-14. The upper support beam for the Case, connecting the Case to the 12U Flight Case.
Figure A-15. The prototype of the railing that goes on top of the Flight Case.

Figure A-16. The prototype of the lower / locking part for the railing.
Figure A-17. The assembled Case mounted to the 12U Flight Case.

Figure A-18. The assembled Case mounted to the 12U Flight Case.
Figure A-19. The assembled Case mounted to the 12U Flight Case with the lids opened.
B. CAD 3D Models

Figure B-1. The Case mounted to the 12U Flight Case with the lids moved out of position.

Figure B-2. The Case mounted to the 12U Flight Case with the front lid moved out of position.
Figure B-3. The Case mounted to the 12U Flight Case with the lids closed.

Figure B-4. The Case mounted to the 12U Flight Case with the front lid being transparent.
Figure B-5. The Case mounted to the 12U Flight Case, the front lid and base being transparent.

Figure B-6. The Case mounted to the 12U Flight Case, transparent case.
Figure B-7. The Case mounted to the 12U Flight Case, viewed from rear left side.

Figure B-8. The Case mounted to the 12U Flight Case, with the back lid transparent.
Figure B-9. The Case mounted to the 12U Flight Case, base and rear lid transparent.

Figure B-10. The Case mounted to the 12U Flight Case, base and lids transparent.
Figure B-11. The Case mounted to the 12U Flight Case, viewed from left side.

Figure B-12. The Case mounted to the 12U Flight, front and rear lid transparent.
Figure B-13. The Case mounted to the 12U Flight, the case being transparent.

Figure B-14. The Case mounted to the 12U Flight, viewed from the front.
Figure B-15. The Case mounted to the 12U Flight, viewed from the front with the lid transparent.

Figure B-16. The Case mounted to the 12U Flight, front and back lid transparent.
Figure B-17. The Case mounted to the 12U Flight, viewed from the back.

Figure B-18. The Case mounted to the 12U Flight, viewed from the back with the back lid transparent.
Figure B-19. The Case mounted to the 12U Flight, back and front lid transparent.

Figure B-20. The Case mounted to the 12U Flight, viewed from the under side.
Figure B-21. The Case mounted to the 12U Flight, Demo & Event and Ericsson logo branding.

Figure B-22. The Case mounted to the 12U Flight, Ericsson and logo branding.
Figure B-23. The Case mounted to the 12U Flight, viewed from the top.

Figure B-24. The Case mounted to the 12U Flight, viewed from the side with the front lid open.
Figure B-25. The Case mounted to the 12U Flight, front lid open.

Figure B-26. The Case mounted to the 12U Flight, viewed from the front with the front lid open.
Figure B-27. The Case mounted to the 12U Flight, front and back lid open.

Figure B-28. The Case mounted to the 12U Flight, viewed from the back with both lids open.
Figure B-29. The Grid created when drawing the official Ericsson logo.
C. Miscellaneous

Figure C-1. The Potentiometer with the cables soldered onto its pins and the rotation knob mounted.

Figure C-2. The LCD Display and the cables soldered onto the circuit board.
Figure C-3. The cables soldered onto the GSM breakout board.

Figure C-4. The cables soldered on to the GSM breakout board.
Figure C-5. The electrical system when testing the LCD display, warning LEDs and the temperature sensor.

Figure C-6. The LCD display showing data collected by the temperature sensor.
Figure C-7. The qwiic connection between the GSM Breakout board and the GPS module. Also the u.fl to SMA adapter which connects the circuits to the antennas.
Figure C-8. The qwiic connection between the GSM Breakout board and the GPS module. Also the u.fl to SMA adapter which connects the circuits to the antennas.

Figure C-9. The SparkFun LTE CAT M1/NB-Iot Shield with the SARA-R4-R410M-02B LTE Cat M1/NB-Iot modem.
Figure C-10. The SparkFun GPS-RTK - Neo-M8P-2 Board with the Neo-M8P-2 GPS module.

Figure C-11. The whole electrical system test assembled on a breadboard, with all the connectors and circuits connected.
Figure C-12. The hand drawing with an overview over the project/case.
D. Electrical Schematics

Figure D.1. Electrical circuit for the diode switching with the help of a digital out signal and BJT.
E. Arduino Code

Figure E-1. Settings for the Arduino Uno and what data port it should communicate on.

```c
#include <LiquidCrystal.h>

const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;

LiquidCrystal lcd(16, 2);

void setup() {
  lcd.begin(16, 2);  // Starts the LCD command with the settings 16x2 dot matrix
  lcd.print("Hello World!");  // LCD Print command
}

void loop() {
  lcd.setCursor(0, 1);  // Sets the cursor at the first letter of the second row
  lcd.print(millis() / 1000);  // Prints a timer counting from 0
}
```

Figure E-2. Arduino sketch for testing of the LCD Display.
Figure E-3. GSM module configuration code and the serial monitoring.

Figure E-4. GSM configured and connected to a network, seen on the serial monitor. The APN, IP and RSSI is censored.
Figure E.5. The code and serial monitoring for sending a text message from the GSM module. The mobile phone number is censored.

Figure E.6. The code and text message received on the mobile phone, the mobile phone number is censored.
```cpp
#include <SoftwareSerial.h>
#include<TinyGPS.h>

float lat = 28.5458, lon = 77.1703;

SoftwareSerial gpsSerial(3, 4);
TinyGPS gps;

void setup() {
    Serial.begin(9600);
    gpsSerial.begin(9600);
}

void loop() {
    while (gpsSerial.available()) {
        if (gps.encode(gpsSerial.read())) {
            gps.f_get_position(&lat, &lon);
            println("GPS Signal");
            println("Lat: ");
            print(lat);
            println();
            println("Long: ");
            print(lon);
        }
    }

    String latitude = String(lat, 6);
    String longitude = String(lon, 6);
    Serial.println(latitude + ";" + longitude);
    delay(1000);
}
```

Figure E-7. The first GPS test code.
Figure E.8. The code for the GPS configuration and serial monitor.

```
TemperaturesensorCode25
//Enri Ostaund 2019-05-16
//Temperature sensor-code

int sensePin = A0;             // The Arduino Pin that will be used
int sensorInput;              // The variable with the sensor input
double temp;                  // The variable with the temperature in degrees.

void setup() {                   // Start the Serial Port at 9600 baud (default value)
  Serial.begin(9600);
}

void loop() {

  sensorInput = analogRead(A0); // read the analog sensor and store the value from the analog pin A0
  temp = (double)sensorInput / 1024; // find percentage of input reading
  temp = temp * 5;                  // multiply by 5V to get voltage
  temp = temp - 0.5;                // Subtract the offset from the signal
  temp = temp * 100;                // Convert to degrees Celsius

  Serial.print("Current Temperature: ");
  Serial.println(temp);
  delay(1800);
}
```

Figure E.9. Arduino sketch for the temperature sensor.
```cpp
// Emil Ostlund 2019-05-17
// Temperature sensor and LCD code

#include <LiquidCrystal.h>

const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;

LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

int sensePin = A0;
int sensorInput;
double temp;

int tempHigh = 7;
int tempMed = 8;
int tempLow = 9;

void setup() {
  Serial.begin(9600);
  lcd.begin(16, 2);
  lcd.print("Current Temp: ");

  pinMode(tempHigh, OUTPUT);
  pinMode(tempMed, OUTPUT);
  pinMode(tempLow, OUTPUT);
}

void loop() {
  sensorInput = analogRead(A0);
  temp = (double)sensorInput / 1024;
  temp = temp * 5;
  temp = temp - 0.5;
  temp = temp * 100;
```
```c
int temp2 = temp;

Serial.print("Current Temp: ");
Serial.println(temp);

if (16 < temp) {
    digitalWrite(temphigh, HIGH);
    digitalWrite(tempmed, LOW);
    digitalWrite(templow, LOW);
}
else if (14 < temp < 16) {
    digitalWrite(tempmed, HIGH);
    digitalWrite(temphigh, LOW);
    digitalWrite(templow, LOW);
}
if (temp < 15) {
    digitalWrite(templow, HIGH);
    digitalWrite(tempmed, LOW);
    digitalWrite(temphigh, LOW);
}
lcd.setCursor(0, 1);
lcd.print(temp);
lcd.print(" °C");
delay(100);
}
```

Figure E-10. Arduino sketch for the temperature sensor, LCD display and temp monitoring.
Figure E-11. The code for the LED Matrix Panel.
F. 3D Printing

Figure F-1. The Cura file, the 3D test model sliced and ready to 3D print.

Figure F-2. The result of the test print that got stopped halfway through the printing process.
Figure F-3. The result of the test print that got stopped halfway through the printing process.
Figure F-4. The Cura file, the test for the scaled logo and texts.

Figure F-5. Results for the 3D printed logo and text.
Figure F-6. The Cura file, the scaled prototype front and back lid.

Figure F-7. The Cura file, the scaled prototype front and back lid.
Figure F-8. The result of the 3D printed front and back lid.

Figure F-9. The result of the 3D printed front and back lid.

Figure F-10. The result of the 3D printed front and back lid.
Figure F-11. The Cura file, the scaled prototype base.

Figure F-12. The result of the 3D printed Base.
Figure F-13. The Cura file, the scaled prototype of the support beams and railing.

Figure F-14. The result of the 3D printed support beams and railing.
Figure F-15. The result of the 3D printed pieces mounted together.

Figure F-16. The result of the 3D printed pieces mounted together.
Figure F-17. The result of the 3D printed pieces mounted together.

Figure F-18. The result of the 3D printed pieces mounted together.
Figure F-19. The result of the 3D printed pieces mounted together.