

TALLINN UNIVERSITY OF TECHNOLOGY  
School of Information Technologies

Andree Orasson 204765IVEM

# **Affordable Internet of Things Solution for Transmitting SIA DC – 09 Messages**

Master's thesis

Supervisor: Ivo Mürsepp  
PhD

Tallinn 2022

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond

Andree Orasson 204765IVEM

# **Soodne nutistu lahendus SIA DC – 09 sõnumite saatmiseks**

Magistritöö

Juhendaja: Ivo Mürsepp  
Doktorikraad

Tallinn 2022

## **Author's declaration of originality**

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Andree Orasson

09.05.2022

## **Abstract**

There are different possibilities for exchanging data between users in today's world and the Internet of things (IoT) solutions have become affordable, secure and cost effective. The purpose of the current master thesis was to find an alternative transmitter solution for an outdated custom-made ultra high frequency (UHF) transmitter system by developing the most suitable system. The developed solution was proposed for an Estonian company which was looking for an updated system for remotely monitoring the sensors' network.

There were some problems with the present system such as the transmitter not being updated (due to being custom-made) and all the available solutions not fulfilling all essential criteria which the company had set. The author of the current thesis conducted an overview of scientific publications and patents which describe different alternatives for data transmission. The overview indicated that GPy modem with custom-made printed circuit board (PCB) was the best alternative for the current system.

The conducted test showed that the custom-made PCB main functions worked as planned. A special software was developed for the GPy to read general purpose input/output pins (GPIO) status and generate a SIA DC – 09 messages. The code was tested thoroughly which indicated that the protocol could be implemented on the edge device. The tests were done in laboratory conditions and some connection losses were noticed during the test runs without finding the root cause. Field trials should be done before large-scale deployment in order to collect more data for future analysis.

In conclusion, the main goal of the given master thesis was achieved. A new solution for transmitting SIA DC-09 message was proposed and the tests proved that the protocol could be implemented on the edge device. The developed system had less elements in the communication path and was more affordable compared to ready-made products.

This thesis is written in English and is 53 pages long, including 5 chapters, 29 figures and 4 tables.

## **Annotatsioon**

### **Soodne nutistu lahendus SIA DC-09 sõnumite saatmiseks**

Tänapäeval on olemas erinevaid võimalusi andmete edastamiseks kasutajate vahel ning nutistu lahendused on muutunud kulutõhusamaks, turvalisemaks ning odavamaks. Käesoleva magistritöö eesmärk oli pakkuda ning välja töötada alternatiivne informatsiooni edastamise lahendus kasutuses oleva erilahendusena tehtud UHF saatjate-sensor süsteemile. Väljatöötatud süsteemi pakuti Eesti ettevõttele, mis oli parasjagu otsimas võimalike uuenduslike lahendusi kaugloetavatelt sensoritelt andmete kogumiseks.

Praegusel kasutusesoleval süsteemil oli mitmeid probleeme. Kasutatavat saatmislahendust ei saanud enam uuendada (kuna oli valmistatud erilahendusena) ning pakutavad alternatiivid ei sobinud ettevõtte seatud kriteeriumitega. Magistritöö käigus analüüsiti teaduslikke publikatsioone ja patente, mis kirjeldavad erinevaid informatsiooni edastamise viise. Koostatud analüüs näitas, et GPy modem koos erilahendusena disainitud trükkplaat on parim alternatiiv praegusele kasutuses olevale süsteemile.

Läbiviidud katsed magistritöö käigus valmistatud trükkplaadiga näitasid, et plaadi põhifunktsioonid töötavad plaanitult. Arendusplaadi jaoks kirjutati eraldi programmikood, mis suudab tuvastada andurilt tulnud informatsiooni, koostada SIA DC-09 andmepaketi ning saata kasuliku infot kasutaja poolt määratud SIA DC-09 serverile. Laboratoorsetes tingimustes testimisel täheldati, et GPy modem võib aegajalt kaotada ühenduse serveriga. Läbi tuleks viia pikaajalisem testimine, et välja selgitada võimalik juurpõhjus.

Magistritöö käigus seatud põhiülesanded said täidetud. Välja pakuti alternatiivne lahendus kasutusesolevale süsteemile ning demonstreeriti, et SIA DC – 09 protokoll on võimalik kasutada nutistus. Välja arendatud lahendus on vähem elemente võrreldes kasutuses oleva süsteemiga. Pakutud süsteem on kulutõhusam ja väiksema omahinnaga

kui praegu turul pakutavad tooted. Eelpool mainitud ettevõtte kaalub tõsiselt pakutud lahendust ning on huvitatud lahenduse rakendamisest oma igapäeva töös.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 53 leheküljel, 5 peatükki, 29 joonist, 4 tabelit.

## List of abbreviations and terms

2G	Second Generation
3G	Third Generation
3GPP	3rd Generation Partnership Project
4G	Fourth Generation
5G	Fifth Generation
AC	Alternative Current
ACK	Acknowledgement Messages
AMR	Automatic Meter Reading
ASCII	American Standard Code for Information Interchange
BOM	Bill of Materials
CRC	Cyclic Redundancy Check
CRS	Central Station Receivers
DC	Direct Current
DUH	Unable Acknowledgement
EC-GSM-IoT	Extended Coverage GSM for Internet of Things
eGPRS	enhanced General Packet Radio Service
GPIO	General Purpose Input/Output
GSM	Global System for Mobiles
HEX	Hexadecimal
I/O	Input/Output
I2C	Inter-Integrated Circuit
IoT	Internet of Things
IP	Internet Protocol
LDO	Low Dropout Regulator
LED	Light-Emitting Diodes
LPWA	Low Power Wide Area
LTE	Long Term Evolution
LTE MTC CAT M1	LTE Machine Type Communications Category M1



M2M	Machine to Machine
MCU	Microcontroller Unit
MIB	Master Information Block
MITM	Man-In-The-Middle Attack
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
MQTT	Message Queuing Telemetry Transport
NAK	Negative Acknowledgement
NB-IoT	Narrowband IoT
OTA	Over the Air
PC	Personal computer
PCB	Printed Circuit Board
PE	Premises Equipment
REPL	Read-Evaluate-Print-Loop
SD	Secure Digital
SIA	Security Industry Association
SIM	Subscriber Identity Module
SPI	Serial Peripheral Interface
SSL/TLS	Secure Sockets Layer and Transport Layer Security
TCP	Transmission Control Protocol
UART	Universal Asynchronous Receiver-Transmitter
UDP	User Datagram protocol
UE	User Equipment
UHF	Ultra High Frequency
USB	Universal Serial Bus

## Table of contents

1 Introduction .....	14
1.1 The problem.....	14
1.2 Overview of the current system.....	15
1.3 Project Specification.....	16
1.4 The project objectives.....	17
2 State of art.....	18
2.1 Low power wide area technical standards .....	21
2.1.1 NB-IoT .....	22
2.1.2 EC-GSM-IoT.....	22
2.1.3 LTE-M.....	23
3 Hardware .....	25
3.1 Printed circuit board design.....	29
4 Software.....	38
4.1 Safety aspects of the IoT systems.....	38
4.2 SIA DC – 09 protocol overview .....	40
4.3 The program code and implementation .....	44
4.4 Further work .....	50
5 Summary.....	52
References .....	54
Appendix 1 – Non-exclusive licence for reproduction and publication of a graduation thesis .....	59
Appendix 2 – Printed circuit board drawings.....	60
Appendix 3 – Printed circuit board bill of the materials .....	63
Appendix 4 – The GPy with printed circuit board .....	66
Appendix 5 – The boot.py program code .....	67
Appendix 6 – The main.py program code .....	68

## List of figures

Figure 1. Currently used monitoring system working principle schematic.....	16
Figure 2. Communication path from the sensor unit to the SIA DC-09 server.....	16
Figure 3. Voltage divider schematic.....	30
Figure 4. Showing board to board connections. ....	31
Figure 5. DC-to-DC converter schematic.....	33
Figure 6. Battery voltage measurement circuit.....	34
Figure 7. USB to UART bridge schematic.....	35
Figure 8. USB cable connected to the USB port: (a) USB cable did not reach the USB port fully, (b) the USB cable reached the port successfully. ....	37
Figure 9. The template for the event based on SIA DC-07-2001.04.....	41
Figure 10. SIA event message showed with elements. ....	43
Figure 11. The proposed solution working principle schematic. ....	44
Figure 12. Firstly used LTE firmware upgrade command. ....	45
Figure 13. Secondly used LTE firmware upgrade command.....	45
Figure 14. AT-commands used to search Telia network.....	45
Figure 15. SIA DC – 09 example message.....	47
Figure 16. Received DUH message from the server. ....	47
Figure 17. Received ACK message from the server. ....	48
Figure 18. LTE attach command used for establishing connection. ....	48
Figure 19. The main.py flowchart schematic. ....	49
Figure 20. Received SIA DC–09 message in the IoBroker.....	50
Figure 21. Printed circuit board drawing sheet 1.....	60
Figure 22. Printed circuit board drawing sheet 2. ....	61
Figure 23. Printed circuit board drawing sheet 3. ....	62
Figure 24. The GPy with printed circuit board.....	66
Figure 25. The boot.py program code. ....	67
Figure 26. The main.py program code. ....	68
Figure 27. The main.py program code. ....	69
Figure 28. The main.py program code. ....	70

Figure 29. The main.py program code..... 71

## **List of tables**

Table 1. LPWA cellular standard overview. ....	23
Table 2. Service providers comparison table.....	24
Table 3. Development board comparison table. ....	28
Table 4. Printed circuit board bill of the materials. ....	63
Table 4. Printed circuit board bill of the materials. ....	64
Table 4. Printed circuit board bill of the materials. ....	65

# **1 Introduction**

There has always been a need for developing better information transmission systems in order to offer a better customer experience, lower the running costs and have better coverage. Competition between companies guarantees continuous development and research for improving the existing systems and proposing new solutions.

The purpose of the current master thesis is to find an alternative transmitter solution for an outdated custom-made UHF transmitter system by developing the most suitable system and conducting test runs. The new solution should simplify the currently used system and be more affordable. The problems with the present system are defined and overviews of scientific publications and patents are conducted which are used for proposing a new solution. The developed solution is proposed for an Estonian company which is looking for an updated system for remotely monitoring the sensors' network.

## **1.1 The problem**

The current master's thesis is going to propose a solution for an Estonian company, which has an emerging problem with their currently used UHF transmitter units. The given units have been used for relaying information from the sensors to company's own server. Over ten thousand devices are deployed all over Estonia. The information from the monitoring units are received on a daily basis. The transmitter units has been used over 20 years and now the company has started looking for new solutions which can replace the legacy systems. The main problems with the old transmitters are that the maintenance costs are very high and it is hard to find replacement devices for the given application because they are no longer being manufactured. Also, the current solution was custom-made for user needs and cannot be upgraded for the new technologies. It is no longer reasonable for the company to invest in outdated systems while the new devices have lower operating and setup costs as well as increase the reliability of system. Additional concerns have risen from the aspect of safety because the used UHF

frequencies are publicly accessible even though the company is paying a frequency fee the transmission can still be easily interrupted.

The Company has set three major objectives which the given master's thesis follows:

- The proposed solution has to be completely ready for use, no additional investment needed;
- The system should be easily deployable;
- The cost of the device should be under 50 €.

## **1.2 Overview of the current system**

The current device set consists of the transmitter unit, sensor and power supply. The working principle schematic of the sensor and transmitter can be seen in Figure 1. The UHF transmitter is responsible for receiving information from the sensor and relaying it to the company's server. Every device has a unique code which is added in front of the data package before the transmission so that the server can separate different units. Additionally, the server does not send out the reception confirmation because the current communication link is one way, therefore retransmissions have been added to the currently used UHF transmitters to ensure that the information is received and decoded by the server. The working principle schematic of the communication path is shown in Figure 2. The sensor has six 12 volt outputs which are connected to the transmitter and every digital signal has a defined meaning. If any output goes high, then information about it is transmitted to the server. The sensor unit and UHF transmitter have same energy source which duplicated with a 12-volt battery and direct current (DC) adapter because the currently used transmitters take a lot of energy during broadcast and it drains the battery. Also, the battery is used for back-up when there is a problem with the DC adapter.

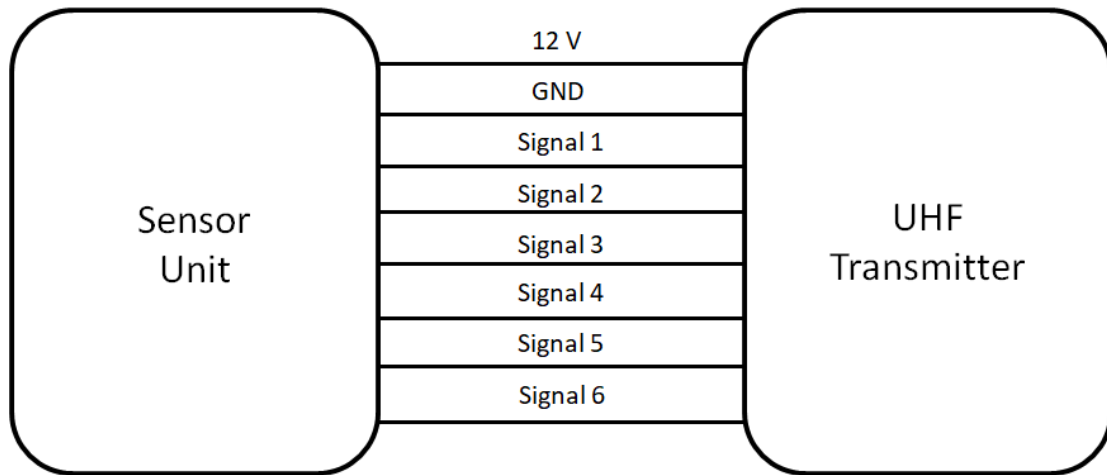


Figure 1. Currently used monitoring system working principle schematic.

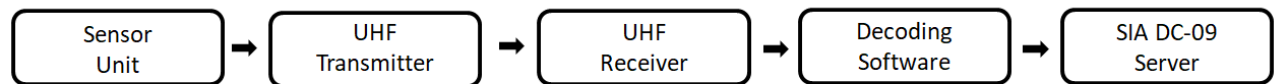


Figure 2. Communication path from the sensor unit to the SIA DC-09 server.

### 1.3 Project Specification

The purpose of this specification is to describe technical aspects of the project. These principles are going to be used for selecting and verifying a suitable solution.

The Company has set the following criteria:

- The offered solution has to use cellular connectivity;
- The transmission unit should have six digital input 12 V pins;
- The used communication protocol has to be SIA DC – 09;
- The system must relay sensor information to the server in less than 10 seconds;
- The solution should be able to transmit 500 kB data per month.

The company prefers to use cellular connectivity because Estonia has good coverage and the communication link is more robust compared to the existing solution. The offered solution could preferably be easily upgraded to fifth generation (5G) later. The



company continues to use sensors which have six 12 V digital outputs after having replaced the UHF transmitters which are currently used. When there is a change in any of the sensor's output pin statuses, the data has to be relayed to the company's server in less than 10 seconds. The company has estimated that 500 kilo bytes of information is transmitted by one transmitter unit every month.

#### **1.4 The project objectives**

The goal of the current master's thesis is to develop a prototype solution for the company by following the project specification and objectives described in the problem statement. Firstly, a market research is carried out to get an overview of the available technologies and it should indicate the best possible data transmission solution. Secondly, a suitable hardware is going to be selected for this project. The reliability and functionality of the prototype solution are tested during field trials and the results will be analysed to see if the targets of the project specification and objectives are achieved.

## 2 State of art

The following section of the thesis observes different ways of collecting data which are used and described in other research papers. The main focus is on the remote monitoring systems which can collect the data and transmit it to the user specified location for future analysis. The given overview shows what kind of technologies and standards are used for remote monitoring and should indicate a suitable solution for this master thesis.

The automatic meter reading (AMR) system has been used for remote monitoring utilities and data collection at the customer's site. The AMR includes a battery operated measuring device, which measures and collects the data e.g. gas consumption. The information is transmitted to the network node and relayed to service provider [1] [2] [3]. In AMR wireless communication industrial, scientific and medical bands are used, which means the anti-interference and data management capabilities of such networks vary greatly and unstable data transmission and issues in security and reliability are major concerns [3] [4]. The downsides of AMR are one way communication and limited information, therefore more infrastructure is needed compared to Internet of Things (IoT), where the modem and measuring unit can be combined in one device [3] [5]. IoT enables stable, real-time traffic data collection from meters, device status monitoring, command delivery and additional remote operations [4]. When a local telecommunication service provider Telia first opened their Narrowband IoT service in 2018, the launch customer was gas company OÜ Energate who used this network for monitoring gas meters. In their press release OÜ Energate stated that they were planning to change all gas meters to remote readouts using NB-IoT technology, which means they can offer better and comfortable solutions for clients and at the same time save money while using a more efficient system [6].

Cellular communicators are another type of technology which can be used for remote status monitoring. Additionally, they have controllable ports which can be used in different applications such as automatic gate opening or controlling ventilation. The system is capable of operating in second generation (2G), third generation (3G) or

fourth generation (4G) networks. The status information is then sent to the server which can be accessed via a web application. Also, the user can be notified with an Short Message Service message. The communicator usually has two power supply terminals, one for the alternative current (AC) adapter and the other one for the battery. The system offers a two-way communication and is preconfigured and therefore the user only needs to define the status of the input/output pins. One of the drawbacks of the cellular communicators is having a limited number of input/output pins [7] [8].

The following examples describe different cases of using IoT applications in real life scenarios which were discussed in various research papers. As one example, a group of scientists developed IoT prototypes for monitoring river behaviour to help prevent flooding in Colima state, Mexico. Fixed position and moving drifter nodes were used for collecting real time data and transmitting it to the server. The one monitoring system consisted of 32-bit microcontroller, ultrasonic water level sensor, 3G cellular modem and power supply, which had a battery and solar panel with a controller. The system was designed so that it could support a variety of sensors and it would use a Message Queuing Telemetry Transport (MQTT) protocol to send data through a cellular communication module which was integrated into its main board. The device sends data strings using AT-Commands from the Microcontroller Unit (MCU) to the cellular module to communicate with an external MQTT broker which was installed on a server-side environment that manages all subscription and production topics [9]. Additionally, MongoDB was implemented on the server side to resolve potential issues related to different sensors having their own log structures. The authors had considered vulnerability aspects of the wireless system which were solved by using encryption and Secure Sockets Layer and Transport Layer Security (SSL/TLS) certificates. The fixed nodes were deployed in different locations near the Colima river where the 3G signals were strong enough to send messages to the data acquisition platform for future analysis. The authors say in the conclusion that this kind of data collection and analysis can be used for monitoring the river, but data storing should be used to avoid losing any measurements during connection losses [9]. A similar solution was developed in Estonia by Flydog Solutions OÜ, where ultra-low-cost environmental monitoring devices were designed to enable cloud-based data collection and extensive real-time monitoring of the status of floods in urban areas. The proposed early detection solution was aimed at creating a shift from just dealing with the consequences of a flood towards increasing

the time to prepare for a flood and thus reducing the consequential (in)direct costs, which could result in saved lives and decreased damage to land and property [10].

Another example is an IoT-based continuous vacuum-packed food monitoring system developed by a research group consisting of Rumanian, Indian and Vietnamese scientists in order to have intelligent packaging and also ensure quality and safety of the food products. The constructed system was divided into two parts for lowering the costs and power consumption. The data acquisition module consisted of an Arduino board connected with gas, pressure, temperature and humidity sensors which were exposed to the analysed product. The sensors used Inter-Integrated Circuit (I2C) and Serial Peripheral Interface (SPI) to communicate with the Arduino. The Zigbee protocol was used for transmitting information from Arduino to the computer, where it was stored in excel spreadsheet. Additionally, the LabVIEW software was used for graphic user interface and MATLAB for further analyzation. The authors say that the whole system was user-friendly and low-cost as it could be constructed by a hobbyist for less than 100 €. An onion was chosen as a test sample because it is used widely in foods. When the onion was put into a vacuum container and dried under vacuum, the gas sensor measured large fluctuations during the first 2 hours after which it stabilised and the humidity increased 30% at the beginning of the test. Later, both values stabilized at the same time, which meant the food had entered into preservation state. The authors say in the conclusion that food dried under vacuum spoils less than food dried in traditional ways, because the process takes place in a closed chamber and therefore a lower moisture level is achieved. Furthermore, using this type of IoT solution can alert people when the food starts deteriorating [11].

As a final example, a methodology for retrofitting a legacy building with IoT capabilities was proposed in Malaysia. The idea was to obtain information about the energy consumption and indoor climate of the building. Additionally, the paper also brought out a number of issues behind retrofitting a 20-year-old building. In Malaysia, where the research was carried out, more than 50% of the building energy is consumed by cooling systems. Retrofitting old buildings with new technologies helps to improve and analyse energy usage and predict the building's maintenance needs. Firstly, the researchers developed and proposed a protocol for managing a large scale retrofitting project. The described process should be as follows: involving the local community, observing the indoor environment and identifying possible sensing variables, mapping

the floor plan and testing the selected sensors. The testing should bring out possible problems while setting up and connecting the sensors, e.g the location and availability of the local area network and power sockets if wired connection is used. The paper identified a number of issues and challenges which may happen during the retrofit: cost overrun, the security of data, discrepancies in IoT sensor communication protocols and human cooperation. The measured information should be stored in the local server to avoid data leaking and ensure free access to the data. IoT sensors may use different communication protocols to communicate to the server and this could make retrofitting more complicated than initially planned. The authors conclude that the proposed protocol should be used for large scale retrofitting [12].

The overview shows that low power wide area (LPWA) technologies are widely used for remote monitoring and direct internet connection is preferred. The connection between the sensor equipment and server can be established with cellular modems or communicators. The collected information should be transmitted to the server where final data processing takes place.

## **2.1 Low power wide area technical standards**

There are many low power wide area technical standards which can be divided into two categories: one is a proprietary patented technology that works on unlicensed spectrum, such as LoRa, Sigfox, RPMA, etc [13]. The other three solutions operating in licensed spectrum bands are: Extended Coverage GSM for Internet of Things (EC-GSM-IoT), Long Term Evolution Machine Type Communications Category M1 (LTE MTC CAT M1, also referred to as LTE-M) and NB-IoT – have emerged to address the diverse requirements of the IoT market which have been standardised by 3rd Generation Partnership Project (3GPP). The main advantage of 3GPP-standardised LPWA solutions, compared to proprietary technologies, is that they have the support of a huge ecosystem with more than 400 individual members. 3GPP stipulates that standardised technologies deliver a minimum level of performance, regardless of the vendor. Standards also ensure interoperability across vendors and mobile operators [14]. The perspective of coverage capability, LTE-M based on LTE network is relatively weak, while low-rate LoRa, Sigfox, NB-IoT, and EC-GSM are relatively strong, and Sigfox

can reach 50km coverage. The data transmission rate, LoRa, Sigfox and EC-GSM transfer rates are not sufficient to transmit large amounts of information, while LTE-M and NB-IoT can perform large data transmission [13]. Also, they are more robust and secure compared to unstandardized LPWA technologies [15]. NB-IoT and LTE-M are 3GPP standards that are both set to coexist with other 3GPP 5G technologies, therefore fulfilling the long term 5G LPWA requirements [16].

### **2.1.1 NB-IoT**

NB-IoT is a combination of NB-CIoT and NB-LTE technology. Among them, NB-CIoT was jointly proposed by Huawei, Qualcomm and Neul. NB-LTE was proposed by Ericsson, ZTE, Nokia, Samsung, Intel and other manufacturers [13]. NB-IoT has been designed to offer extended coverage compared to the traditional GSM networks. The complexity of NB-IoT devices can be even lower than that of GSM devices due to the changes to the synchronisation signal design and simplified physical layer procedures simplifying the received signal processing [14]. NB-IoT is ideal for low-throughput, delay-tolerant use cases with low mobility support, such as smart meters, remote sensors and smart buildings [13]. It is meant to be used to send and receive small amounts of data (generally in two - or three - digit numbers of bytes) over a period of time generated by low data - producing IoT devices [17]. NB-IoT can coexist with LTE. In addition, NB-IoT in standalone operation can coexist with GSM/3G/LTE, according to 3GPP evaluations [14].

### **2.1.2 EC-GSM-IoT**

The optimisations made in EC-GSM-IoT allow the technology to be introduced into existing GSM networks as a software upgrade on the radio network and also on the core Network [14]. It is based on enhanced General Packet Radio Service (eGPRS) and designed as long communication distance, low power consumption and low complexity cellular system for IoT communications [13]. At the same time, this backwards-compatibility includes resource sharing between EC-GSM-IoT and legacy packet-switched services to allow for a gradual introduction of the technology in the network without the need to reserve dedicated resources for IoT. EC-GSM-IoT has been

designed to offer coverage for machine to machine (M2M) devices in locations with challenging radio coverage conditions [14].

### 2.1.3 LTE-M

Many cellular operators and companies such as Nokia, Ericsson and Qualcomm introduced LTE-M as a potential technology to optimize LTE for the IoT. However, EXALTED was the first project of the European Union's Seventh Framework Program (FP7) to present LTE-M as a new system that extends LTE specifications for M2M communications and supports future wireless communication systems in terms of extended coverage, lower cost, better security, energy efficiency and the ability to support a larger number of connected devices [18]. LTE-M is a cellular radio access technology specified by 3GPP in Release 13, 14 and 15 [19]. In general, LTE-M can be used to refer to all use of LTE for M2M and IoT and the evolution of LTE MTC features. This includes both CAT-0 and CAT-M1 (even CAT-1) user equipment (UE), and other features such as Power Saving Mode (PSM) and extended DRX cycles [20]. LTE-M connected devices can have a battery life of up to 5 - 10 years for a wide range of use cases, while enabling a reduction in modem costs up to 20% - 25% compared to the cost of eGPRS modems [21].

Table 1. LPWA cellular standard overview.

	<b>LTE - M</b>	<b>NB-IoT</b>	<b>EC-GSM-IoT</b>
Downlink peak rate	1 Mbps	200 kbps	70 kbps
Uplink peak rate	1 Mbps	144 kbps	70 kbps
Duplex mode	Half	Half	Half
Range	< 11 km	< 11 km	< 20 km
Cellular connectivity	4G	2G/3G/4G	2G
Devices	18 000	50 000	50 000
Latency	6.7 s	2 s	2 s

Telia, Elisa, and Tele2 are the three main telecommunication service providers in Estonia who also offer cellular internet of thing services. The UE could work in 2G-4G

or NB-IoT if the configuration allows it. The Uploading and downloading speeds are the same as for other users, only the data usage is limited. The comparison between the providers is presented in the Table 2 [22] [23] [24].

Table 2. Service providers comparison table.

	<b>Telia</b>	<b>Tele2</b>	<b>Elisa</b>
Data usage per month	15 MB/month	30 MB/month	25 MB/month
Price	0.99 €/month	0.98 €/month	1.19 €/month

Remote data monitoring can be done in several ways; however the most suitable option is to use cellular connectivity because it offers direct connection to the internet. Additionally, cellular technologies are standardized by 3GPP and therefore are more robust and secure compare to unstandardized LPWA technologies. The research papers shows that MCU should be used for preliminary data analysis and the final processing needs to be done in the server. The given analysis demonstrates that NB-IoT with the MCU is the most suitable technology for solving the problem of the current master thesis.



### 3 Hardware

The project specification required a suitable hardware for transmitting the sensor information to the customer specified server. A market research was carried out to find out if an appropriate device was available and could be bought or there was a need for a custom-made solution to be designed for the current thesis. The results of the market research were presented in the Table 3.

Arduino MKR NB 1500 is a development board which is equipped with LTE CAT M1/NB – IoT modem which supports various bands used all over the world. The board is designed for 5 V supply and the modem and controller are on a single board [25]. The program can be written on the Arduino IDE or web application in C++ language, which is later processed and compiled to machine language [26]. Additionally, the web application support firmware updates over the air (OTA). Arduino MKR NB 1500 has a Li-Po charging circuit that allows the board to run either on battery or on external power. The board has 8 digital input/output (I/O) pins, 7 analogue input pins, 1 analogue output pin, a single Micro subscriber identity module (SIM) slot, a U.FL connector for antenna and also SPI, universal asynchronous receiver-transmitter (UART) and I2C communication ports. Additionally, the board is equipped with USB Micro B port for software configuration [25].

Pycom GPy is a development board combined with a microcontroller unit and CAT-M1/NB-IoT modem, which supports bands with global coverage. Additionally, the GPy has WiFi and Bluetooth capability. The operating voltage is 5 V and it has 20 GPIO, where 4 pins support UART and SPI connection each and 2 pins could be used for I2C. The board has 2 U.FL connectors, one of which is for WiFi/Bluetooth and the other for LTE as well as nano SIM slot. Micropython language is used for programming the microcontroller unit and it can be done by using Atom or Visual Studio Code. In both cases, a Pymakr plugin is required for uploading the written code to the development board [27]. Also, the plugin enables the user to communicate to the board, using the built-in command line read-evaluate-print-loop (REPL) [28]. Pycom offers a web application called Pybytes, which allows the owner to monitor the used devices and update the firmware OTA if needed [27].

Actnius Icarus IoT Board is built around Nordic Semiconductor nRF9160 modem and combines LTE-M, NB-IoT, GPS, accelerometer, universal serial bus (USB), lithium-polymer charger as well as an eSIM with free data out of the box and a nano SIM slot. Additionally, the board can measure the battery voltage level. It has 18 GPIOs, where up to 4 pins can be utilized for SPI, I2C or UART connections, as well as the USB Micro B port. The board operating voltage is 5 V during normal usage but it has a safety rating for 28 V for input voltage. It can be programmed in the nRF Connect SDK or in the Visual Code Studio where C++ is used as programming language. The nRF9160 chip manufacturer Nordic Semiconductor offers cloud service for remote controlling and monitoring [29].

Cellular communicator G16 is used with security systems which allow the status information to be transmitted to the server and Trikdes software is used for receiving the information. The same software can be used for monitoring the communicators and relaying the gathered information into central command system by using SIA DC-09 protocol. The communicator has 2 input ports for listening to the status information, a USB Micro B port, a SMA port and a SIM card slot. The modem can operate in 2G, 3G and 4G. The system is designed to operate 12 V and can be equipped with a port expander for additional cost [30].

Uplink 5500ATT is a cellular alarm communicator, which is specifically designed to be used in AT&T cellular networks. The 5500ATT is compatible with the 3G and LTE networks and transmitted signals over the 850&1900 MHz frequency bands. The communicator can report occurred events in the following formats: Contact ID, SIA, Pulse 4/2 and Modem IIe/IIIa2. It has four 12-V inputs and two outputs which can be configured by the user. Additionally, the communicator can be configured by using a smartphone app or web interface [31].

There are several readymade solutions available on the market, but they do not fulfil all criteria described in the project specification. Arduino MKR NB 1500, Pycom Gpy and Icarus IoT Board nRF9160 offer a large amount of GPIO, but all these three boards have a 5-V operating voltage. A logic level converter from 12 V to 5 V is necessary for conversion if one of these microcontrollers are to be selected. Additionally, a special software needs to be written in order to use the SIA DC-09 protocol for transmitting data. The cellular communicator G16 and Uplink 5500ATT have a 12-V operating

voltage and can be easily configured for the SIA DC – 09 protocol, but they do not have 6 GPIO which are required. Cellular communicator G16 has the possibility to connect the port extender, but this would make the total cost of the device over 200 €. Also, Uplink 5500ATT is configured to work in the United States market with local carrier and therefore additional investments might be necessary for customizing it to the local carriers here in Estonia.

One of the alternatives in order to reduce the cost of the device would be to design the modem from scratch. The preliminary study showed that such self-made design could be almost half the cost compared to a ready-made solution. However, this option would mean that more investments to the prototypes would be necessary to validate the reliability of the system and therefore the author of the current thesis decided not to proceed with the described alternative.

The overview compiled by the author of the current thesis showed that the SIA DC – 09 protocol was not widely used and the author did not find any research paper which would describe the protocol implementation on IoT device. The protocol was used in some patents [32] [33] [34], but the application area was different compared to the current master thesis. Additionally, the market research revealed some available solutions, however, they did not fulfil the company criteria. The author proposed to proceed with the PyCom GPy modem with a custom-made PCB, which would be responsible for the signal conversion and powering the modem. This option was estimated to be more cost-effective and more convenient to use than the current readymade systems. Additionally, the described prototype should indicate whether the system works as planned and can be implemented in the future and give the company an advantage in the market.

Table 3. Development board comparison table.

<b>Board</b>	<b>Operating voltage</b>	<b>I/O ports</b>	<b>Communication protocols</b>	<b>Carrier compatibility</b>	<b>Price</b>	<b>Stock Status</b>
Arduino MKR NB 1500	5 V	7 analogue inputs 1 analogue output	SPI UART I2C	LTE-M / NB-IoT	66,90 €	Out of Stock
PyCom GPy	5V	20 GIPO	2x SPI 2x UART I2C SPI	LTE-M / NB-IoT	48,40 €	In stock
Icarus IoT Board - nRF9160	5 V	18 GIPO	4x SPI 4x UART 4x I2C	LTE-M / NB-IoT	130 €	Out of stock
Cellular communicator G16	12V	2 Inputs 1 output	USB Mini-B	2G/3G/4G	120 €	In stock
Uplink 5500ATT	12V	4 Inputs 2 Output	micro-USB	4G	170€	In stock

### 3.1 Printed circuit board design

The printed circuit board was designed for converting input signal levels, programming and powering the GPY. In the signal conversion part of the schematic the 12 V logical signals from the device were converted to 5 V levels for GPY. The DC-to-DC conversion was used for step down 12 V supply voltage to 5 V for supplying the development board. GPY itself did not have a USB port and as a result, programming and debugging capability were added to the designed board.

The schematic of the circuit was divided into 9 subsections (Appendix 2). The signal conversion from 12 V to 5 V was done in the first section, the second and the ninth part showed the solutions for board-to-board connections, the third and the fourth sections described the power supply, the battery voltage measurement was explained in the fifth part. The sixth and the seventh section described the USB connection and the eighth part showed LED indicators, which were used for visual control. Online shops Farnell and Digi-Key were used for finding the suitable components for PCB.

The 12 V logical signals from the sensor needed to be converted to a suitable level for GPY. The signal levels had to be converted by using the voltage divider, which was simplest passive linear circuit. It consisted of two resistors connected in series where the input voltage was applied across the resistor series and the output voltage was taken from between them [35]. The GPY input pin high status ranged from 2.475 V to 3.6 V and the low status is between -0.3 V and 0.825 V [27]. The converted 12 V logical input signal for the GPY should be 3 V, which means there could be  $\pm 10\%$  voltage fluctuation from the sensors side [27]. The current through the voltage divider was chosen 0.65 mA and it was expected that the load current would have been much smaller. The resistor values could be calculated by using the voltage divider equation [35], which was deduced from the Ohm law as follows. Firstly, bleeder resistor value was calculated:

$$R_2 = \frac{V_{R2}}{I} = \frac{3\text{ V}}{0.65\text{ mA}} = 4615\ \Omega \quad (1)$$

Where:

$R_2$  was the value of the bleeder (second) resistor of voltage divider;

$V_{R2}$  was output voltage;

$I$  was current through the voltage divider

Secondly, first resistor value was calculated:

$$R_1 = \frac{V_{R1}}{I_{R1}} = \frac{9V}{0.65mA} = 13846 \Omega \quad (2)$$

Where:

$R_1$  was the value of the first resistor of voltage divider;

$V_{R1}$  was voltage drop on  $R_1$  a subtraction between  $V_{OUT}$  and  $V_{IN}$ ;

$I$  was current through the voltage divider

Theoretically calculated resistor values should had been  $4615 \Omega$  and  $13846 \Omega$  but selected values were  $5100 \Omega$  and  $14000 \Omega$  because they had better stock margins.

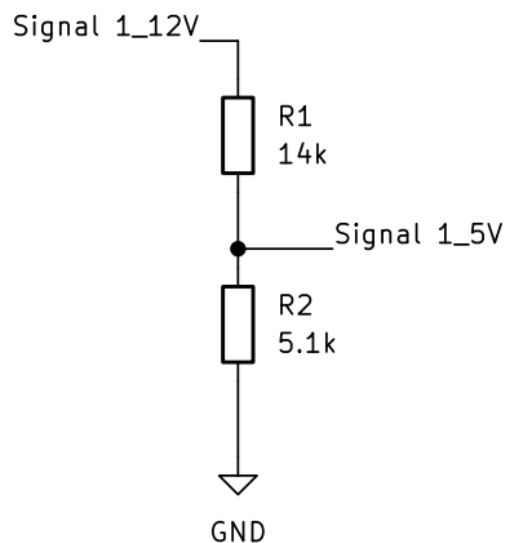


Figure 3. Voltage divider schematic.

Board to board connectors were used for making a connection between the GPY and the designed PCB as shown in Figure 4. One row with an eight-position screwed (marked as J3) terminal had been used for connecting the sensor wires to PCB because soldering in the field would have been impractical and not comfortable. Another one row with a two-position screwed (marked as J4) terminal was chosen to use in the battery voltage measuring circuit for the same reason as described above. Different terminals were used to reduce the possibility of making any wrong connections while being set up by a technician. The logical signals 1 to 6 from the sensor were connected correspondingly to the GPY logical inputs pins from 5 to 11. The 12 V input supply voltage from the power source was connected directly to the DC-to-DC converter input.

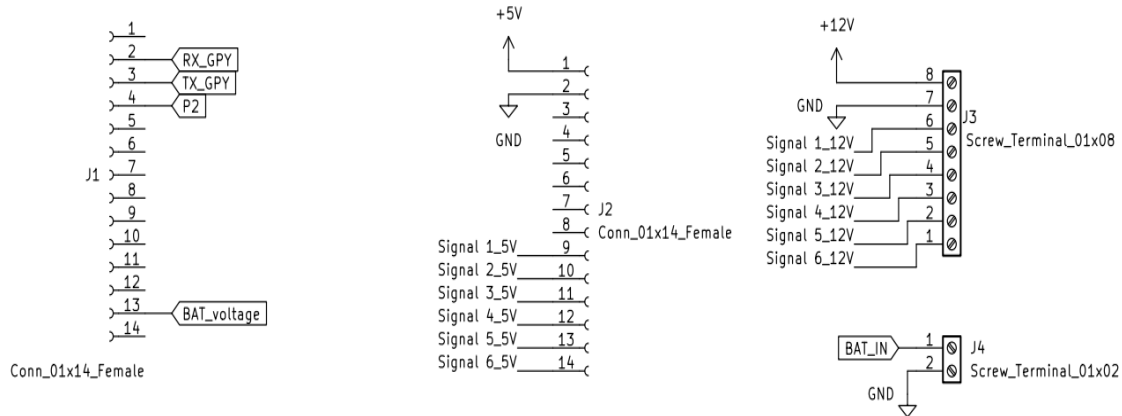


Figure 4. Showing board to board connections.

The supply voltage conversion could be done by using the low dropout regulator (LDO) or the switching regulator in order to offer stable power for the GPY. The LDO had a simpler layout and tended to cost less but they took up more PCB area compared to switching regulators and had a lower efficiency [36]. For example, in application, where input voltage of 12 V needed to be converted output voltage of 3 V, the LDO efficiency was less than 27.5% compared to the switching regulator where it was more than 90% [37] [38]. However, the switching regulator produced a switching noise and the PCB design was more complex than the LDO [37] [38]. The GPY supply input voltage was between 3.5 to 5.5 V and the board maximum current consumption was 1.2 A during transmission. The input voltage was chosen 5 V to manage any fluctuation from the power supply. The data described above allowed to calculate the theoretical power

dissipation of the LDO and the switching regulator. Firstly, the maximum output power was calculated by using power formula (3).

$$P_{out} = I_{out} \cdot V_{out} = 1.2 \cdot 5 = 6 \text{ W} \quad (3)$$

Where:

$P_{out}$  was the output power;

$V_{out}$  was the output voltage and

$I_{out}$  was the output current.

Secondly, the input power was calculated by dividing output power with regulator efficiency. The switching regulator's efficiency was 90% and the LDO's 30%.

$$P_{In\_SR} = \frac{P_{out}}{0.9} = 6.6 \text{ W} \quad (4)$$

$$P_{In\_LDO} = \frac{P_{out}}{0.3} = 20 \text{ W} \quad (5)$$

Where:

$P_{In\_SR}$  was the input power for switching regulator and

$P_{In\_LDO}$  was the input power for LDO;

The difference between the input and the output power was a power loss which was converted to heat. Theoretically, the switching regulator produced 0.6 W and LDO 11.2 W heat while doing DC-to-DC conversion. The rule of thumb was approximately 1 W of dissipation in 1 mm<sup>2</sup> of board area resulted in a 100°C temperature rise [38]. If the LDO had been used, then a radiator would have been required for cooling, while switching regulator could have been used without it. The given analysis showed that the best power converting solution was the switching regulator.



The MP4560DN-LF-P switching regulator manufactured by MONOLITHIC POWER SYSTEMS was selected for DC-to-DC conversion because it offered 5 V<sub>out</sub> with 2 A constant output as well as a suitable stock margin. The switching regulator datasheet provided typical application schematic for 5 V output with a recommended components list. Manufacturer's recommendations were followed during the circuit and PCB design (Figure 5) [39].

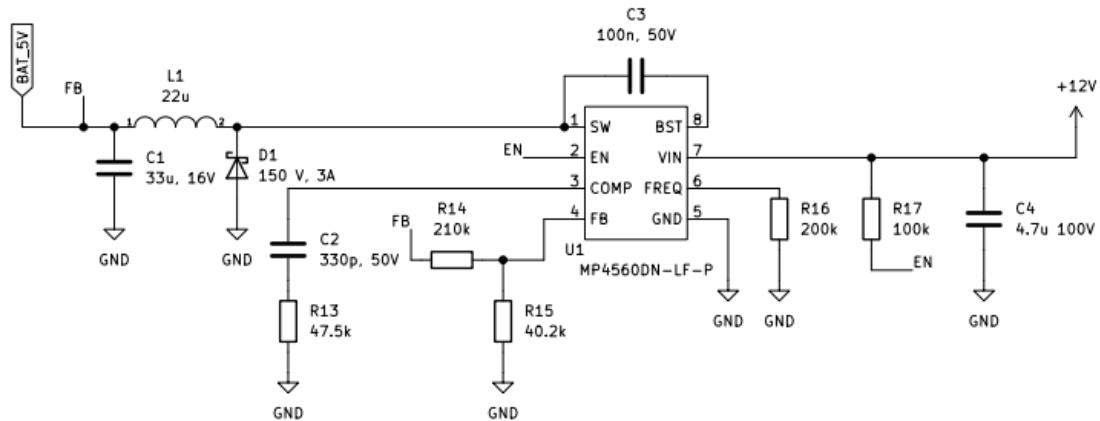


Figure 5. DC-to-DC converter schematic.

The microcontroller unit could be powered from the USB connection and the external power supply, which would be the main energy source during normal operation. While configuring GPY via the USB, the battery source could be also connected to PCB, which could cause a disturbance in the circuit. The supply input switch was designed and the P-channel metal oxide semiconductor field effect transistor (MOSFET) was selected for switching input power for the GPY. The MOSFET drain to source voltage ( $V_{ds}$ ) and continuous drain current ( $I_d$ ) were selected as high as possible 60 V<sub>ds</sub> and 5.1 A<sub>id</sub> to tolerate possible voltage fluctuations in the system. The Schottky diodes were added for reverse polarization protection, the same diode components were used in the other parts in the schematic to lower the cost of the bill of materials (BOM) and keep the circuit as simple as possible. A capacitor was added for voltage smoothing at the end of supply input switch.

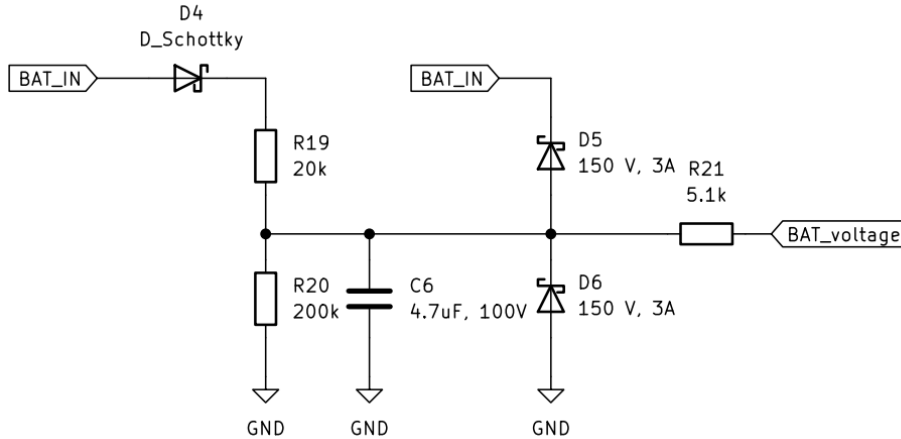


Figure 6. Battery voltage measurement circuit

The battery voltage measurement circuit was designed for monitoring the battery voltage status and inform the company if the battery needed to be replaced (Figure 6). This part was added to the demo board for testing purpose and could be removed if the customer did not need it. The voltage divider was used in battery voltage measurement where  $12 V_{in}$  was made  $5 V_{out}$  for the GPY. A current limiting resistor and a pair of clamping diodes were added to protect the GPY I/O ports. There could be a difference between the calculated and measured voltage value due to voltage drops on resistors and therefore the final configuration for measuring the battery voltage was done in software level.

Pycom offered a separate Expansion board where the GPY could be connected for software uploading, but using this solution in the field could be challenging if all the hardware installation had been completed already. In order to solve this issue, a software uploading capability was added to the designed PCB where USB to UART bridge was exploited to communicate the GPY (Figure 7). The CP2102-GM was selected for the conversion where the bridge's TX pin was connected to the GPY's RX pin and the GPY's TX to the bridge's RX pin [40]. Because the CP2102-GM supported the USB 2.0 standard, the USB Type C port with USB 2.0 capability was chosen for physical connection [41]. Additionally, the European Commission made it mandatory for common chargers for electronic devices to have USB type C [42]. A separate switch was added to the PCB for allowing the GPY to enable a bootloader mode.

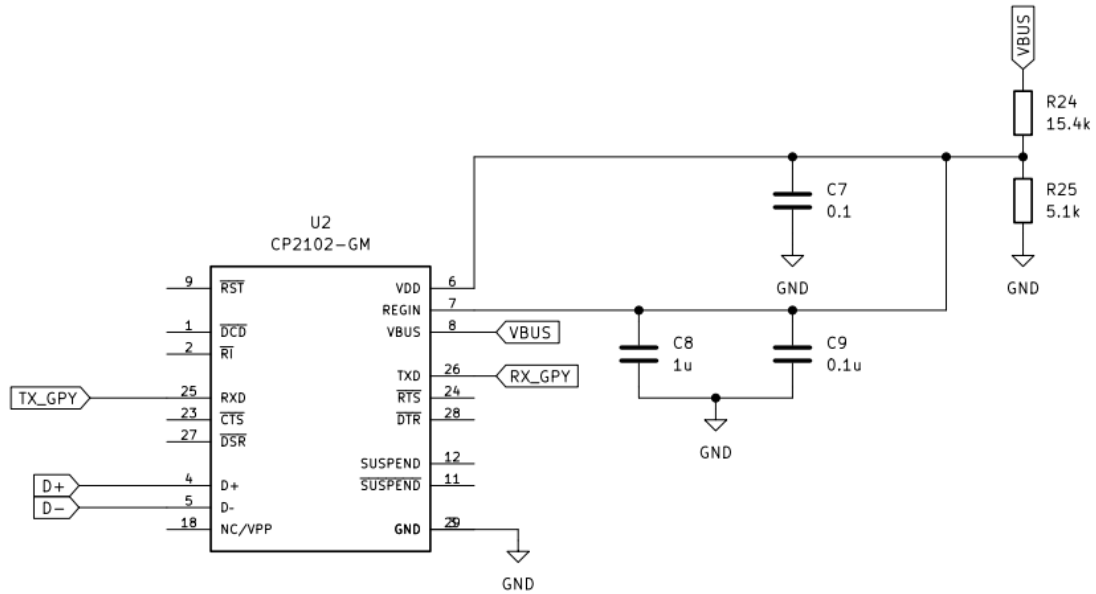


Figure 7. USB to UART bridge schematic

Two light-emitting diodes (LED) were added to PCB for visual control to confirm if 12 V supply arrived from the sensor's side to board and if 12 V to 5V conversion was working properly. A resistor was needed before LED, which is also called a ballast resistor, to prevent excessive current burn out LED. The resistor values had been calculated by using formula (6) [43]:

$$R = \frac{V_S - V_f}{I_f} \quad (6)$$

Where:

$V_S$  was a supply voltage;

$V_f$  was a LED forward voltage and

$I_f$  was a LED forward current.

The selected LED forward voltage was 2.4 V and the current 20 mA [44]. The calculated resistor value for 12 V supply voltage was 480  $\Omega$  and for 5 V it was 130  $\Omega$ . A 510  $\Omega$  resistor was selected for both voltages to keep the bill of material as simple as

possible and the given component had more suppliers compared to calculated values. When the bigger resistor value was used compared to the calculated value then the LED was dimmer.

The size of the designed printed circuit board was 83x110x1.5 mm and it had two sides which were both connected to the ground. Additionally, M4 holes were added to the PCB corners for enabling the technician to mount the board to the case. The signal track width was 0.250 mm and the power supply track was 1 mm, whereas the copper thickness for both was 1 oz. The best practise recommendations of the printed circuit board manufacturer were followed during the design process [45]. The board was ordered from JLCPCB, who offered affordable manufacturing and fast delivery. The PCB components cost 32 € and the manufacturing 6.89 € for five PCB. The total cost for one prototype was around 37 €, which included shipping. The price of one board could be reduced in the future if the order quantities became bigger. The bill of the materials list was presented in appendix 3.

The components were soldered into the board by the author. After soldering, a functional test was done to the board where different sections were checked separately. Firstly, short-circuit tests were done to avoid any malfunctions and no mistakes were discovered. The measured value of the voltage divider was 3.35 V. The DC-to-DC converter test showed that it worked as planned, the measured voltage was 4.6 V. The USB to UART bridge was able to make a connection with the computer. The given test showed that the USB cable did not fully reach the port because the USB port was not close enough to the PCB edge (Figure 8).

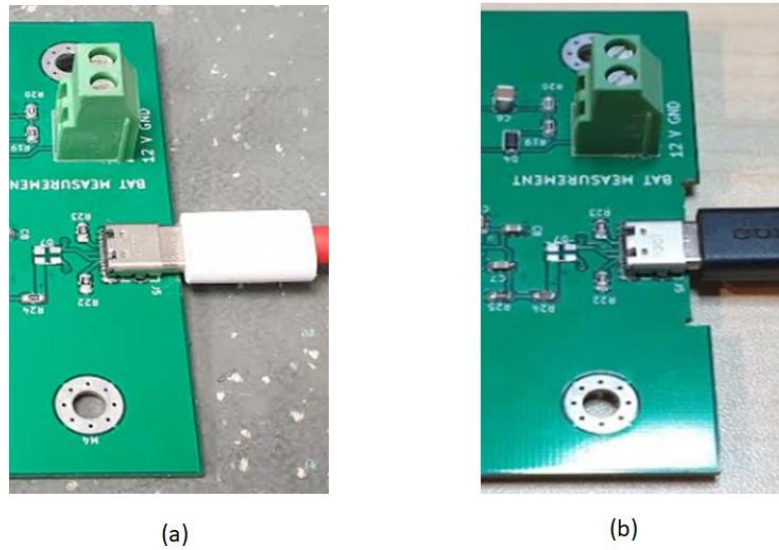


Figure 8. USB cable connected to the USB port: (a) USB cable did not reach the USB port fully, (b) the USB cable reached the port successfully.

The author of the thesis decided not to use the USB to UART bridge in the second revision of the PCB because it did not serve the purpose in the update process flow. The first board configurations need to be done on the company premises where all the necessary software updates, SIA DC – 09 and NB-IoT program codes would be uploaded to the board. The field technician would only need to add the customer ID which could be done OTA. Also, these changes would have affected the PCB layout which could be made smaller and more compact. The cost of the BOM would have been reduced by around 8 € compared to the first revision. The PCB of the second revision should include: DC-to-DC converter, signal conversion, LED indicators and the battery voltage measurement circuit.

## **4 Software**

A special software needed to be developed for the GPy to communicate with the company's server. The safety aspects of the IoT systems and SIA DC-09 standard were studied before the program code was written to avoid possible mistakes. The program scripts with testing results were presented at the end of chapter 4.

### **4.1 Safety aspects of the IoT systems**

The security aspect of the IoT devices are a concern for the company and therefore possible weaknesses of the system are studied. Different parts of the communication link are observed and the current overview indicates which kind of preventive actions has to be done.

In cellular networks base-stations provide connectivity between the core network and the nearby cellular devices [46]. One of the possible attack type is a man-in-the-middle attack (MITM) using a rogue base station. When the victim UE camps on the rogue base-stations, the attacker may perform an attack such as sniffing or blocking a message transmitted from the victim UE to the outside or reject communication from the outside to the victim UE through a MITM [47] [48]. In LTE network the common features of the Attach and Paging procedures include both Random Access and Radio Resource Control Connection procedures. They both occur after the UE receives Master Information Block (MIB) information by broadcasting from base-station. The vulnerabilities appear in the MIB messages that are broadcasted in the Attach and Paging procedures. Initially all messages are delivered in plain text before the Security Context Setup for encryption or integrity verification are completed and therefore if a malicious attacker intends, they can sniff messages in the middle of the transmission [47]. During the attach procedure the UE sends an attach request message to base-station in plain text. After receiving the attach-request the mobility management entity challenges the UE with an authentication challenge. This is necessary procedure for the UE to authenticate with the network [49]. When a UE has no data to send, it enters an idle mode and wakes up periodically which is called the paging occasion. During the paging procedure the data exchange between the UE and base-station is not encrypted and commands are sent in plain text over the air [49]. The lack of

confidentiality guarantees to the paging protocol's original goal of balancing between the device's battery consumption and quality-of-service [46].

Another possibility for MITM is in local area network while lightweight application-layer protocol is used. The MQTT has gained tremendous support and wide use to the extent of becoming "the de facto standard of IoT". However, MQTT does not have a sophisticated security mechanism on its own, but relies on SSL/TLS to encrypt MQTT messages. SSL and TLS are cryptographic protocols that allow communications to be encrypted, thereby making attacks like sniffing and MITM more difficult [50]. The TLS (and SSL) protocols are located between the application protocol layer and the Transmission Control Protocol (TCP) / Internet protocol (IP) layer, where they can secure and send application data to the transport layer. Because the protocols work between the application layer and the transport layer, TLS and SSL can support multiple application layer protocols. TLS and SSL assume that a connection-oriented transport, typically TCP, is in use [51]. Unfortunately, cryptographic protocols are poorly enforced in IoT deployments and some devices do not have the resources to support SSL/TLS. MITM attacks is grants the attacker a large degree of freedom for manipulation, by having both read and write access (within a network of IoT devices) to sensitive or critical information without being noticed. In comparison, eavesdropping has no write access, and both jamming and Denial of Service attacks can be easily detected [50].

Jamming happens when the attacker generates a radio signal in order to interfere with the legitimate wireless signal between the subscriber and the base station. Jamming affects all users who are being serviced by the base-station, which makes the attack easy to detect [52]. In addition to affecting the LTE, jamming also has effect on other wireless systems which transmit and receive useful data.

The IoT device message could be influenced by MITM or jamming. One option would be a MITM with rogue base-station which could be used for hijacking the UE and disrupting false information. Second possibility would be MITM while lightweight application-layer protocol is used without encryption. MITM attacks are technically challenging to design because the attackers need in-depth understanding of protocol and network details [50]. Lastly, jamming could be used for blocking the communication path, but this would affect other users. Both attack methods are considered to be highly

unlikely to happen, but the recommendation is to use SSL/TLS protocols if they are not already required by the SIA DC – 09 standard to ensure data security.

## **4.2 SIA DC – 09 protocol overview**

One of the criteria described in the project specification is to use a SIA DC – 09 communication protocol for transmitting information from the edge device to the server. The given standard is developed by the Security Industry Association and used by the industry volunteers. The latest version of the standard was published in 2020 and it is an enhancement of the older versions of the standard [53]. The first SIA standard (SIA DC-03-1990.01) was published in 1990 [54]. SIA standards establish the minimum performance requirements and are intended neither to preclude additional product features or functions nor to act as a maximum performance limit. Any product the specifications of which meet the minimum requirements of a SIA standard should be considered in compliance with that standard [53]. The standard document defines compatibilities, requirements, testing and possible simulation cases. In the current master thesis, the minimum requirements of the SIA DC – 09 are going to be implemented on the prototype solution. Additional features described in the standard could be implemented on the MCU if the first results are positive and they are needed.

The standard required that the Premises Equipment (PE) and the Central Station Receivers (CSRs) either support User Datagram protocol (UDP) or TCP. When the PE or CSR support only one protocol, UDP is the preferred implementation but TCP may be used [53]. The standard can be implemented on any media that carries IP. The CSRs should have static IP address but the PE may have dynamic or static IP address. Advanced Encryption Standard can be used for encryption and it has to full fill Federal Information Processing Standards Publication 197. Additionally, encrypted messages shall use Cipher Block Chaining. Encryption support is optional for PE but mandatory for CSRs. When encryption is used, only the data, timestamp and padding content of a message are encrypted [53]. When encryption is selected, the user may use a key length of 128, 192 or 256 bits and a matching key value (and therefore matching key length) must be programmed at the PE and the CSR [53].

PE may send two types of messages: events and link supervision (the latter is used for communication path status). CSRs send only one type of message, acknowledgment,



which may have three types: Acknowledgement Messages (ACK), Negative Acknowledgement (NAK) or Unable Acknowledgement (DUH). ACK means the messages were received without errors and positive acknowledgement is sent back to the PE. NAK message is sent back to the user when an encrypted message fails the timestamp test. DUH is used when the server was unable to process an otherwise correctly received message. The NAK and DUH message are never encrypted [53]. The event format which the PE uses for the events is based on SIA protocol DC-07-2001.04. The template for this event is shown in Figure 9.

```

<LF><crc><0LLL>
<"id"><seq><Rrcvr><Lpref><#acct>[<pad>]...data...[x...data...]<timestamp>
<CR>

```

Figure 9. The template for the event based on SIA DC-07-2001.04

The message elements are described below:

- LF is the American Standard Code for Information Interchange (ASCII) linefeed character, transmitted as a binary value 0x0A [53];
- Cyclic redundancy check (CRC) is used for detecting errors in raw data. The portion of the message starting with the first quote character of the ID and ending with the timestamp are included in CRC calculation. The CRC has to be transmitted as four ASCII characters [53];
- 0LLL indicates the length of the message. This length element consisted of the character "0" (ASCII zero) followed by 3 hexadecimal (HEX) digits (in ASCII). The characters counted are the same as are included in the CRC calculation [53];
- "id" field contains an ASCII token to indicate the format used in the data field of the message, and whether or not encryption is used. The quote characters are included in the message. PE has to support at least one of the tokens SIA-DCS and ADM-CID (Ademco Contact ID) and CSR has to support both of them. The tokens are defined in DC-07-2001.04. When the data and timestamp of a message are encrypted, the ID Token is modified to insert an ASCII "\*" after the quotation character and before the first character of the token. For example, an

unencrypted SIA DCS packet would use the token "SIA-DCS" and an encrypted SIA DCS packet would use the token "\*SIA-DCS" [53];

- Seq (sequence number) which the PE applies to each message as it is queued. The server has to echo the sequence number of the message to which it is replying in its acknowledgement messages. The user shall increment the sequence number to be used as each new message is queued. When the sequence number is 9999, the next sequence number is 0001 [53];
- Rrcvr (Receiver Number) is part of the account identification element. PE may be programmed to further extend the identification provided by the account number and account prefix by providing a receiver number. This is optional element and consists of an ASCII "R", followed by 1-6 HEX ASCII digits for the receiver number. When the PE does not need to transmit a receiver number, nothing is transmitted for this element [53];
- Lpref (account prefix) is part of the account identification element. It can be programmed into the PE to extend the identification provided by the account number. This element is required, and consists of an ASCII "L", followed by 1-6 HEX ASCII digits for the account prefix. When the PE does not need to transmit an account prefix, "L0" should be transmitted for this element [53];
- #acct (account number) is the most specific token and most important element of the account identification. It is always programmed into the premises equipment to identify. The account token appeared both in the header of the message (which is never encrypted) and in the data of the message (which may have been encrypted). This element consists of an ASCII "#", followed by 3-16 ASCII characters representing hexadecimal digits for the account number [53];
- [Data] or [<pad>|Data] is a message data and it is always represented in ASCII characters. The bracket characters "[" and "]" are included in the transmitted message. The data field format is dependent upon the ID token of the message. Where an account number is associated with a message (most message types), the account number data appeared at the start of the data [53];

- [x...data...] is an optional extended data which the PE can add to the message, by including one or more optional extended data fields. The start field is delimited with the ASCII character "[", followed by a single ASCII character (data identifier) that identifies the content of the data field. The data identifier may be any upper case ASCII character in the range "AG" to "Z" The field is terminated with the ASCII character "]" [53];
- <timestamp> has be included in encrypted messages, and may be included on messages that are not encrypted. This field is used to provide protection against message playback. The format of the timestamp is: <\_HH:MM:SS,MM-DD-YYYY> and is always transmitted with a reference of GMT. The braces are not part of the transmitted message, but the underscore, colon, comma and hyphen characters are included. The timestamp field is exactly 20 characters long. The CSR has to validate the timestamp against its own GMT reference. Encrypted messages with a GMT difference from the CSR greater than +20/-40 seconds shall be rejected with a NAK packet [53];
- CR is the ASCII carriage return character, transmitted as a binary value 0x0D [53].

An example message with elements is shown in Figure 10. The given message was in SIA DC – 04 format with no encryption and timestamp. The event expresses a fire alarm in zone 129 [53].

LF	CRC	OLLL	"id"	Seq	Rrcvr	Lpref	#acct
<x0A>	CE11	0032	SIA-DCS	9876	R579BDF	L789ABC	#12345A
			[#12345A	NFA129]	<x0D>		
	#acct	Data	CR				

Figure 10. SIA event message showed with elements.

The first goal is to compile and transmit the simplest SIA DC – 09 message to the server and later add additional features such as encryption and timestamp. This approach should indicate what kind of problems may occur while the first implementation is done on edge device.

### 4.3 The program code and implementation

The GPy had to be prepared and programmed for establishing a connection to the server and transmitting messages as described in SIA DC – 09 protocol. The proposed solution (Figure 11) simplified the system currently used in the company and removed some unnecessary parts while allowing direct connection to the server. Additional software and hardware solutions described in section 1.2 were not needed anymore. The software implementation was divided into three sections: firstly, firmware updates were done to the GPy, secondly, socket client server tests were conducted between the author’s personal computer (PC) and the GPy and lastly, the protocol implementation was carried out along with the testing.

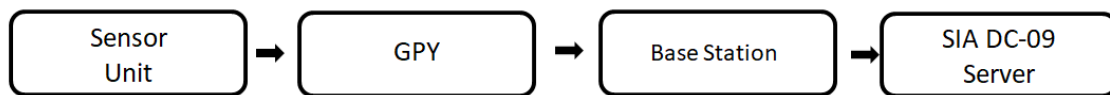


Figure 11. The proposed solution working principle schematic.

The modem configuration and code writing were both done in Visual Studio Code to which Python and Pymakr extensions were installed. Additionally, NodeJS was installed to the computer to support Pycom plugin. The Python plugin added language support as well as additional features such as auto complete, debugging, etc. Small subsets of the Python standard library were included in MicroPython and therefore no additional installations were needed. The MicroPython had to be used for programming MCU as described in chapter 3. The Pymakr plugin was used for uploading the code to the GPy and the REPL was run when code tests were being conducted.

The initial configurations were made after the modem was received. Firstly, the MCU and the modem firmware versions were checked. The modem firmware version was LR5.4.1.0-50523 which was the newest available option. The first number of the series was used to indicate LTE firmware edition, LR5.xx represented CAT-M1 and LR6.xx indicated NB-IoT. The firmware update was done to the modem to use its NB-IoT capability. The LTE configuration was hard coded and the user would not have been able to change it (use AT commands, for example) without updating the firmware version. There is a warning in the manufacturer datasheet about other set up methods

possibly damaging the modem. The Secure Digital (SD) Card update method was used for the firmware version change and the SD Card was formatted. The other update method options were flash, OTA and USB. Then the LTE configuration files were transferred to the SD Card and the upgrade command was used (Figure 12).

```
sqnsupgrade.run('/sd/upgdiff_old-to-new.dup')
```

Figure 12. Firstly used LTE firmware upgrade command.

The given command line gave an error message and therefore another command was used for changing the firmware version (Figure 13).

```
sqnsupgrade.run('/sd/name.dup', '/sd/updater.e1f').
```

Figure 13. Secondly used LTE firmware upgrade command.

The latter initiated a modem firmware version change which was successfully completed. Additionally, MCU and the Expansion Board 3.1v software updates were done in Pycom Upgrade software. The Expansion Board was used for uploading the program code to the MCU and the code was tested with REPL.

The first test was to validate if the GPy could make a LTE connection in NB-IoT mode. The LTE-M antenna kit (Pycom antenna for LTE) and Telia's nanoSIM were connected to the board. Firstly, AT-commands were used (Figure 14) to find Telia's network and establish a connection between the modem and base-station, however, this did not give any results.

```
import pycom
import time
from network import LTE
lte = LTE()
print(lte.send_at_cmd('AT+CPIN="0000"'))
print(lte.send_at_cmd('AT+CPIN?'))
print(lte.send_at_cmd('AT+COPS?'))
print(lte.send_at_cmd('AT+CSQ?'))
```

Figure 14. AT-commands used to search Telia network

The modem location was changed to evaluate if there would be any impact to the attaching procedure but it was concluded that the location did not have any influence on the connection. Secondly, an example code was used from Pycom website to which SIM card pin code unlocking command and Telia's Access Point Name address

(internet.emt.ee) [55] were added. The first attaching attempts to the network were not successful. It was noted that the attaching procedure could take several minutes before the connection is made. Also, location switching tests were run with the LTE example code and there was no impact on the attaching procedure. The modified Pycom example code proved to be working and a simplified version named boot.py was used for making cellular connection later in the thesis (Appendix 5).

Python Socket client-server example [56] was used to confirm if any information was received from the GPy. The server component was run on Visual Studio Code which was installed to the author's PC. The home router setting had to be changed to allow port forwarding which was necessary for enabling the incoming connection to connect to the PC through the internet. Firstly, home router firewall settings were changed to open port 8001 and then the PC local internet protocol (IP) address was matched with the open port. This should have allowed a direct connection between GPy and PC while the messages were sent. Also, it should be noted that the internet service provider (Telia) has to allow port forwarding. This option is by default closed for clients who have cable connection but the settings could be changed in the Telia's self-service on their website. Many attempts were made to establish a communication path between GPy and PC without any success. The router port status was checked on "you get signal" [57] website which showed port 8001 as closed. The same settings were used in an alternative router in a different location where the communication link worked as planned and "you get signal" [57] website showed the router 8001 port status as open. The root cause analysis indicated a problem with the home router firewall settings which could be changed, however in reality the router settings did not start working as planned. The problem was solved by allowing Universal Plug and Play (UPnP) service in the home router. Additionally, PortMapper 2.2.1 software was installed to the PC where the port configuration was done. The connection between the client (GPy) and the server (PC) was successfully made as the next step. While the tests were being contacted, it was noted that GPy had EHOSTUNREACH error which indicated that the communication path to the server was not established. This error occurred randomly and it was assumed that the module had crashed. Resetting GPy usually solved the problem and the root cause study did not reveal anything specific, therefore this should be studied during a longer test period.

The main problem with the UPnP is that it disregards a large portion of network security for the sake of convenience. Routers are generally well equipped to deal with incoming malicious connections. Safeguards like the firewall are put in place to filter out incoming threats. The UPnP often disregards a lot of these security measures, forwarding a port directly to a potential hostile device [58]. In relation to the current thesis the UPnP service was only needed for local testing with the home router and it was not used in real systems.

The SIA DC-09 server component named ELT SIA-IP receiver was installed to the author's PC for decoding incoming protocol packets. The Python socket client code example was modified in Visual Studio Code to send SIA DC – 09 messages. The initial test was conducted in the authors PC where the client code was run on the Visual Studio Code and the server in ELT SIA-IP. An example message was generated on North Latitude Technology website (Figure 15) for the purpose of testing and the local host IP 192.168.1.121 with port 8001 were added to the client code.

```
<LF>61AC0021"SIA-DCS"0002L0#1234[1234|NFA129] <CR>
```

Figure 15. SIA DC – 09 example message.

The linefeed binary value was 0X0A and it was expressed in Python \n and the carriage return binary value 0X0D was \r. The ELT SIA-IP program was configured to listen to IP 0.0.0.0 and port 8001. When a SIA example message was sent to the server program, an error message “Unrecognised data received from IP” was presented in the program. There was a case when a DUH message was received from the server component and the error message indicated a problem with the settings (Figure 16).

```
80B6000E"DUH"0000L0#[ ]
```

Figure 16. Received DUH message from the server.

The ELT SIA-IP program was in trial version and therefore limited adjustments could be done. Another program called IoBroker which had SIA plugin was installed to the author's PC. The same IP and port address were selected as on ELT SIA-IP program. Additionally, account ID was set up as 1234. The same test as with ELT SIA-IP program was contacted with IoBroker where ACK message was received. This confirmed that account ID was needed in the ELT SIA-IP program. The same client server (IoBroker) test was done with the GPy which gave positive results (Figure 17).

```
14310012"ACK"0002L0#1234[ ]
```

Figure 17. Received ACK message from the server.

The GPy had two codes uploaded to the board in the final prototype version. The first code (boot.py) was responsible for establishing the NB-IoT connection and the second code (main.py) was used for reading the GPy's GPIO status and generating a SIA DC – 09 message. The boot.py script was always run first for establishing the NB-IoT connection. The GPy red-green-blue LED changed its status to green when the connection was made. The attach command ensured that the scanning was done in a loop and program code was not proceeded (Figure 18).

```
lte.attach(band=20, apn="your apn")
```

Figure 18. LTE attach command used for establishing connection.

If cellular connection was not made, the modem continued to be in the scanning mode and during the scanning the GPy could restart itself and start scanning again [59]. The main.py file was run directly after boot.py and it contained the main code [60]. When the GPy was connected to the PCB and powered up, both program codes were executed as described above. The SIA DC – 09 message generation was triggered by the GPy's GPIO status changes which were checked one by one in the main.py file and the trigger status was reset after the message was send out. The program code could detect the status change while another event was in progress and a new message could be generated afterwards. All events were transmitted separately which was also required by the standard. The triggering with the message generation took less than two seconds during the test. Instantaneous triggering could be a problem if the sensor output floats which could lead to spamming the server with the event messages and crashing the GPy. The preventive action could be an additional capacitor in the PCB signal conversion section which smoothens the voltage fluctuation and adds a small delay. Also, an additional LED should be added to the PCB for visual control to check if the message generation had been done. If the sensor started acting abnormally it would be noticed by the CSR. The process flow from powering up the GPy to generating protocol message is shown in Figure 19.



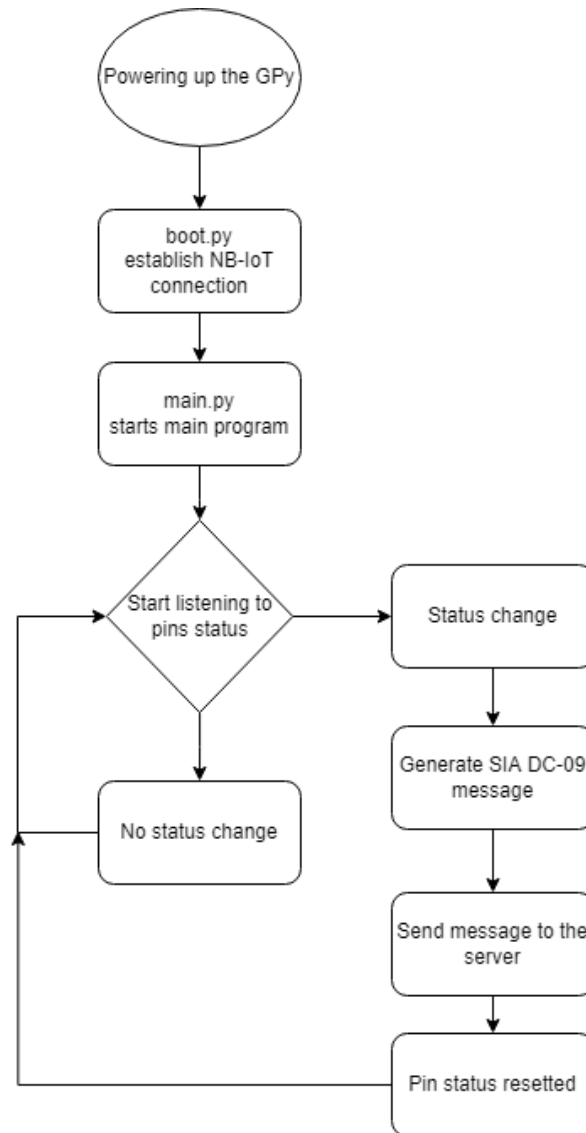



Figure 19. The main.py flowchart schematic.

The main.py program code had three bigger sections: declaring variables, functions and while true loop for monitoring the pins status. The communication path settings, pin status defining, SIA account parameters and events were defined at the beginning of the code. The CRC, message preparation and transmitting functions were used to reduce the code lines in the script. The CRC was used for detecting errors in digital data and the calculation was done accordingly to the SIA DC – 07 standard [61]. Firstly, a special library was used for calculating the CRC value, but the given library [62] could not be implemented to the MCU. The controller was not able to read different library files which were stored into the MCU and therefore a separate function for calculating the CRC was written [63]. The accuracy of the calculation was checked on

“crccalc.com” website [64] which gave the same result as in the CRC function. An anomaly was noticed in the OLLL length calculation which should include same elements as the CRC calculation according to the standard. The tests showed the fields of the element set "ID", "[" and "]" were not needed for the calculation and error messages for the server were received while the light version of the standard was used. North Latitude Technology website [65] was used for checking the message length calculation requirements which gave the same results. The calculation was double-checked with the example message shown in the SIA DC – 09 protocol, where the message length was the same as on the North Latitude Technology website. This could indicate a mistake in the SIA DC – 09 standard text. Separate event codes were used for generating the SIA DC – 09 message and also different functions were applied. The message transmitting function was taken from Python client example code [56], but it was customized for the current thesis. While True loop was used for continuously monitoring the status of the GPIO pins and when the status of one of the pins went high, the message generation function was triggered. The event counter was required by the standard and the status resetting capability of the GPIO pins was added into the while true loop. The main.py code (Appendix 6) was tested several times with positive results (Figure 20).



```
received from [REDACTED] following message: CE110032"SIA-DCS"9876R579BDFL789ABC#12345A[#12345A|NFA129]
```

Figure 20. Received SIA DC–09 message in the IoBroker.

#### 4.4 Further work

The conducted tests with the GPY showed that SIA DC – 09 could be implemented on the edge device. However, the tests were done in the laboratory conditions and therefore the life cycle test should be done (for one month, for example) in the field before using the proposed solution in real life conditions. Also, the field trials should be conducted in multiple locations to evaluate connectivity aspects. The program code should be made more robust and memory effect should be added to the code. Currently, when the GPY resets itself, the program code starts from the beginning. Additionally, the message

encryption should be implemented as described by the standard to ensure data security and supervision messages for checking the communication path status should be added.

## 5 Summary

The purpose of the current master thesis was to find an alternative transmitter solution for an outdated custom-made UHF transmitter system by developing the most suitable system and conducting test runs. The developed solution was proposed for an Estonian (security) company which was looking for an updated system for remotely monitoring the sensors' network.

There were some problems with the present system such as the transmitter not being updated (due to being custom-made) and all the available solutions not fulfilling all essential criteria (using Security Industry Association DC – 09 protocol, having six 12-V input pins and possible cellular connection) which the company had set. The author of the current thesis conducted an overview of scientific publications and patents which describe different alternatives for data transmission. The overview indicated that cellular connections should be used and therefore a market research was made for finding a suitable LTE modem. The selected NB-IoT modem required a custom-made PCB for converting sensor signals and DC-to-DC conversion. The proposed solution fulfilled all necessary requirements and the price difference between a possible ready-made system and custom-made solution indicated almost 40 € savings.

The marked research included comparing five different cellular modems and the best suitable NB-IoT modem was Pycom GPy. The given modem had more extras compared to other modules, a better stock margin and the price was more competitive compared to other products. However, the board operating voltage was 5 V and therefore additional PCB was designed mainly for signal conversion and DC-to-DC conversion. The PCB was tested with the GPy and the results were positive.

A special software was developed for the GPy to read GPIO pins status and generate a SIA DC – 09 messages. The light version of the standard was used without encryption and timestamp. The code was tested thoroughly which indicated that the protocol could be implemented on the edge device. The tests were done in laboratory conditions and therefore field trials should be done before large-scale deployment.

In conclusion, the main goal of the given master thesis was achieved. A new solution for transmitting SIA DC – 09 message was proposed and the tests demonstrated that the

protocol could be implemented on the edge device. The developed system had less elements in the communication path and was more affordable compared to ready-made products.

## References

- [1] SIEMENS, “Electronic heat cost allocator WHE5.. WHE6,” [Online]. Available: [https://hit.sbt.siemens.com/RWD/DB/ES/es/Assets/A6V10796494\\_Electronic%20heat%20cost%20allocator\\_en.pdf](https://hit.sbt.siemens.com/RWD/DB/ES/es/Assets/A6V10796494_Electronic%20heat%20cost%20allocator_en.pdf). [Accessed 15. 11. 2021].
- [2] SIEMENS, “Network node WTT56,” [Online]. Available: <https://sid.siemens.com/v/u/A6V10492030>. [Accessed 15. 11. 2021].
- [3] D. Hoelscher, “The Best Networks & Technologies for Advanced Metering Infrastructure,” [Online]. Available: <https://blog.huawei.com/2020/01/24/the-best-networks-technologies-for-advanced-metering-infrastructure/>. [Accessed 15. 11. 2021].
- [4] HUAWEI, “NB-IoT Smart Gas Solution,” [Online]. Available: <https://www-file.huawei.com/-/media/CORPORATE/PDF/News/NB-IoT-Smart-Gas-Solution-EN.pdf?la=en>. [Accessed 15. 11. 2021].
- [5] Holosys, “Five biggest benefits when implementing NB-IOT AMR solutions,” [Online]. Available: <https://www.holosys.hr/news/five-biggest-benefits-when-implementing-nb-iot-amr-solutions/>. [Accessed 15. 11. 2021].
- [6] Telia, “TELIA AVAS EESTIT KATVA ASJADE INTERNETI VÕRGU,” [Online]. Available: <https://www.telia.ee/uudised/telia-avas-eestit-katva-asjade-interneti-vorgu>. [Accessed 15. 11. 2021].
- [7] TRIKDIS, “Cellular communicator G16\_441W,” [Online]. Available: [https://www.trikdis.com/wp-content/uploads/2020/02/g16\\_441w-um\\_eng\\_2020-02-03.pdf](https://www.trikdis.com/wp-content/uploads/2020/02/g16_441w-um_eng_2020-02-03.pdf). [Accessed 15. 11. 2021].
- [8] Sierra wireless, “Cellular Communicator 5500 Datasheet,” [Online]. Available: [https://cdn.shopify.com/s/files/1/1659/9809/files/SW\\_5500\\_Datasheet-1\\_1.pdf?v=1593097013](https://cdn.shopify.com/s/files/1/1659/9809/files/SW_5500_Datasheet-1_1.pdf?v=1593097013). [Accessed 15. 11. 2021].
- [9] C. Moreno, R. Aquino, J. Ibarreche, I. Pérez, E. Castellanos, E. Álvarez, R. Rentería, L. Anguiano, A. Edwards, P. Lepper, R. M. Edwards and B. Clark, “RiverCore: IoT Device for River Water Level Monitoring over Cellular Communications,” [Online]. Available: <https://www.mdpi.com/1424-8220/19/1/127/htm>. [Accessed 15. 11. 2021].
- [10] Flydog Solutions OÜ, “SMART ENVIRONMENTAL MONITORING,” [Online]. Available: <https://www.flydogmarine.com/products/smart-city/>. [Accessed 15. 11. 2021].
- [11] A. Popa, M. Hnatiuc, M. Paun, G. Oana, D. J. Hemanth, D. Dorcea, L. H. Son and S. Ghita, “An Intelligent IoT-Based Food Quality Monitoring Approach Using Low-Cost Sensors,” [Online]. Available: <https://www.mdpi.com/2073-8994/11/3/374/htm#B19-symmetry-11-00374>. [Accessed 15. 11. 2021].
- [12] S. Dzulkiyfly, H. Aris, B. N. Jorgensen and A. Q. Santos, “Methodology for a Large Scale Building Internet of Things Retrofit,” [Online]. Available:

- <https://ieeexplore.ieee.org/abstract/document/9243304>. [Accessed 15. 11. 2021].
- [13] S.-H. Hwang and S.-Z. Liu, "IEEE," [Online]. Available: <https://ieeexplore.ieee.org/document/8851631>. [Accessed 13. 10. 2021].
- [14] GSMA, "Low Power Wide Area Technologies GSMA," [Online]. Available: <https://www.gsma.com/iot/wp-content/uploads/2016/10/3GPP-Low-Power-Wide-Area-Technologies-GSMA-White-Paper.pdf>. [Accessed 13. 10. 2021].
- [15] S. Dawaliby, A. Bradai and Y. Pousset, "In depth performance evaluation of LTE-M for M2M communications," [Online]. Available: In depth performance evaluation of LTE-M for M2M communications. [Accessed 13. 10. 2021].
- [16] "GSMA 5G mobile iot," [Online]. Available: <https://www.ericsson.com/4a8d35/assets/local/reports-papers/5g/doc/gsma-5g-mobile-iot.pdf>. [Accessed 13. 10. 2021].
- [17] M. E. Klicpera, "WATER METER AND LEAK DETECTION," [Online]. Available: <https://patentimages.storage.googleapis.com/cf/be/c3/5df9edf42e1be8/US20190234786A1.pdf>. [Accessed 13. 10. 2021].
- [18] S. Dawaliby, A. Bradai and Y. Pousset, "In depth performance evaluation of LTE-M for M2M communications," [Online]. Available: <https://ieeexplore.ieee.org/document/7763264>. [Accessed 13. 10. 2021].
- [19] GSMA, "LTE-M Deployment Guide to Basic Feature Set Requirements," [Online]. Available: <https://www.gsma.com/iot/wp-content/uploads/2019/08/201906-GSMA-LTE-M-Deployment-Guide-v3.pdf>. [Accessed 13. 10. 2021].
- [20] T. Tirronen, "Cellular IoT alphabet soup," [Online]. Available: <https://www.ericsson.com/en/blog/2016/2/cellular-iot-alphabet-soup>. [Accessed 13. 10. 2021].
- [21] GSMA, "LTE-M Commercialisation Case Study," [Online]. Available: [https://www.gsma.com/iot/wp-content/uploads/2019/02/201901\\_GSMA\\_LTE-M\\_Commercial\\_Case\\_Study-ATT\\_Telstra.pdf](https://www.gsma.com/iot/wp-content/uploads/2019/02/201901_GSMA_LTE-M_Commercial_Case_Study-ATT_Telstra.pdf). [Accessed 13. 10. 2021].
- [22] Telia, "M2M EUROOPA," [Online]. Available: <https://www.telia.ee/ari/mobiil/m2m-ja-iot-teenused/m2m>. [Accessed 04. 04. 2022].
- [23] Elisa, "ELISA IOT," [Online]. Available: <https://www.elisa.ee/et/ari klient/internet/mobiilsed-lahendused/elisa-iot>. [Accessed 04. 04. 2022].
- [24] Tele2, "M2M PAKETID," [Online]. Available: <https://tele2.ee/ettevotja/internet/m2m#paketid>. [Accessed 04. 04. 2022].
- [25] Arduino, "Arduino MKR NB 1500," [Online]. Available: <https://store.arduino.cc/products/arduino-mkr-nb-1500>. [Accessed 02. 02. 2022].
- [26] CIRCUITO TEAM, "EVERYTHING YOU NEED TO KNOW ABOUT ARDUINO CODE," [Online]. Available: <https://www.circuito.io/blog/arduino-code/>. [Accessed 02. 02. 2022].
- [27] Pycom, "GPy," [Online]. Available: <https://pycom.io/product/gpy/>. [Accessed 02. 02. 2022].
- [28] Pycom, "Pymkr Atom Package," [Online]. Available: <https://atom.io/packages/pymkr>. [Accessed 02. 02. 2022].

- [29] Actinius, “Icarus IoT Board,” [Online]. Available: <https://www.actinius.com/icarus>. [Accessed 02. 02. 2022].
- [30] Alarmtec, “Trikdís G16T GSM kommunikaator,” [Online]. Available: <https://www.alarmtec.ee/tooted/trikdis-g16t-gsm-kommunikaator/>. [Accessed 02. 02 2022].
- [31] Alarm System Store, “Uplink 5500ATT Universal LTE AT&T Cellular Communicator,” [Online]. Available: <https://www.alarmsystemstore.com/products/uplink-5500att-universal-lte-at-t-cellular-communicator>. [Accessed 02. 02. 2022].
- [32] P. J. Dawes, C. DeCenzo and C. Wales, “Premises system management using status signal,” [Online]. Available: <https://patents.google.com/patent/US11201755B2/en?q=SIA+DC09&oq=SIA+DC09>. [Accessed 04. 04. 2022].
- [33] M. Lamb, “DIY monitoring apparatus and method,” [Online]. Available: <https://patents.google.com/patent/US10706715B2/en?q=SIA+DC09&oq=SIA+DC09>. [Accessed 04. 04. 2022].
- [34] P. J. Dawes, C. DeCenzo and C. Wales, “Integrated security network with security alarm signaling system,” [Online]. Available: <https://patents.google.com/patent/AU2017201585B2/en?q=SIA+DC09&oq=SI A+DC09>. [Accessed 04. 04. 2022].
- [35] J. Braza, “HOW VOLTAGE DIVIDERS WORK,” [Online]. Available: <https://www.circuitbasics.com/what-is-a-voltage-divider/>. [Accessed 02. 02. 2022].
- [36] Z. Peterson, “Using a Linear Voltage Regulator vs Switching Regulator in Your PCB,” [Online]. Available: <https://resources.altium.com/p/using-ldo-vs-switching-regulator-your-pcb>. [Accessed 02. 02. 2022].
- [37] H. Zhang, “AN-140: Basic Concepts of Linear Regulator and Switching Mode Power Supplies,” [Online]. Available: <https://www.analog.com/en/app-notes/an-140.html>. [Accessed 02. 02. 2022].
- [38] R. Nowakowski and R. Taylor, “Linear versus switching regulators in industrial applications with a 24-V bus,” [Online]. Available: [https://www.ti.com/lit/an/slyt527/slyt527.pdf?ts=1642007751748&ref\\_url=https%253A%252F%252Fwww.google.com%252F](https://www.ti.com/lit/an/slyt527/slyt527.pdf?ts=1642007751748&ref_url=https%253A%252F%252Fwww.google.com%252F). [Accessed 02. 02. 2022].
- [39] MONOLITHIC POWER SYSTEMS, “MP4560,” [Online]. Available: <https://www.farnell.com/datasheets/3161503.pdf>. [Accessed 02. 02. 2022].
- [40] Pycom, “USB Serial Converter,” [Online]. Available: <https://docs.pycom.io/gettingstarted/programming/usbserial/>. [Accessed 02. 02. 2022].
- [41] Universal Serial Bus Implementers, “USB Type-C Spec R2.0,” [Online]. Available: <https://www.usb.org/sites/default/files/USB%20Type-C%20Spec%20R2.0%20-%20August%202019.pdf>. [Accessed 02. 02. 2022].
- [42] European Commission, “Pulling the plug on consumer frustration and e-waste: Commission proposes a common charger for electronic devices,” [Online]. Available: [https://ec.europa.eu/commission/presscorner/detail/en/IP\\_21\\_4613](https://ec.europa.eu/commission/presscorner/detail/en/IP_21_4613). [Accessed 02. 02. 2022].
- [43] Digi-Key Electronics, “LED Series Resistor Calculator,” [Online]. Available: LED Series Resistor Calculator. [Accessed 02. 02. 2022].



- [44] MULTICOMP PRO, “0805 SMD Chip LED,” [Online]. Available: <https://ee.farnell.com/multicomp-pro/mp008275/led-green-90mcd-577nm-0603/dp/3796304?st=led%20smd>. [Accessed 02. 02. 2022].
- [45] JLCPCB, “Capabilities,” [Online]. Available: <https://jlcpcb.com/capabilities/Capabilities>. [Accessed 02. 02. 2022].
- [46] S. Ankush, H. Syed Rafiul, C. Omar, B. Elisa and L. Ninghui, “Protecting the 4G and 5G Cellular Paging Protocols against Security and Privacy Attacks,” [Online]. Available: <https://sciendo.com/downloadpdf/journals/popets/2020/1/article-p126.pdf>. [Accessed 15. 04. 2022].
- [47] M. Kim, J. Park, D. Moon, J. Jang, Y. Kim and J. Lee, “Long-Term Evolution Vulnerability Focusing on System Information Block Messages,” [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/9289440/authors#authors>. [Accessed 15. 04. 2022].
- [48] H. Kim, J. Lee, E. Lee and Y. Kim, “Touching the Untouchables: Dynamic Security Analysis of the LTE Control Plane,” [Online]. Available: <https://ieeexplore.ieee.org/document/8835363>. [Accessed 15. 04. 2022].
- [49] V. Kumar V and L. K V, “Averting Paging Related Attacks in 4G LTE Communication System,” [Online]. Available: <https://www.ijrte.org/wp-content/uploads/papers/v8i2S4/B10290782S419.pdf>. [Accessed 15. 04. 2022].
- [50] H. Wong and T. T. Luo, “Man-in-the-Middle Attacks on MQTT-based IoT Using BERT Based Adversarial Message Generation,” [Online]. Available: [https://www.researchgate.net/publication/345890033\\_Man-in-the-Middle\\_Attacks\\_on\\_MQTT-based\\_IoT\\_Using\\_BERT\\_Based\\_Adversarial\\_Message\\_Generation](https://www.researchgate.net/publication/345890033_Man-in-the-Middle_Attacks_on_MQTT-based_IoT_Using_BERT_Based_Adversarial_Message_Generation). [Accessed 15. 04. 2022].
- [51] Microsoft, [Online]. Available: <https://docs.microsoft.com/en-us/windows-server/security/tls/schannel-security-support-provider-technical-reference>. [Accessed 15. 04. 2022].
- [52] R. Ghannam, F. Sharevski and A. Chung, “User-targeted Denial-of-Service Attacks in LTE Mobile Networks,” [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/8589140>. [Accessed 15. 04. 2022].
- [53] Security Industry Association, “SIA Digital Communication Standard – Internet Protocol Event Reporting,” [Online]. Available: [https://www.securityindustry.org/wp-content/uploads/2017/10/dc09\\_r2021\\_20201027.pdf](https://www.securityindustry.org/wp-content/uploads/2017/10/dc09_r2021_20201027.pdf). [Accessed 15. 04. 2022].
- [54] ANSI, “SIA DC-03-1990.01 (R2000.11),” [Online]. Available: <https://webstore.ansi.org/Standards/SIA-Security/SIADC03199001R200011>. [Accessed 15. 04. 2022].
- [55] Telia, “APN-I SEADISTAMINE ANDROID TARKVARAGA TELEFONIS,” [Online]. Available: <https://www.telia.ee/abi/juhend/463/apn-i-seadistamine-android-tarkvaraga-telefonis>. [Accessed 05. 04. 2022].
- [56] Python, “socket — Low-level networking interface,” [Online]. Available: <https://docs.python.org/3/library/socket.html#socket.socket.sendmsg>. [Accessed 04. 05. 2022].
- [57] you get signal, “Port forwarding tester,” [Online]. Available:

- <https://www.yougetsignal.com/tools/open-ports/>. [Accessed 05. 04. 2022].
- [58] W. Moors, “This Router Setting is Putting Your Business at Risk,” [Online]. Available: <https://www.securiwiser.com/blog/this-router-setting-is-putting-your-business-at-risk/>. [Accessed 05. 04. 2022].
- [59] Pycom, “LTE Examples,” [Online]. Available: <https://docs.pycom.io/tutorials/networks/lte/>. [Accessed 05. 04. 2022].
- [60] Pycom, “Getting Started,” [Online]. Available: <https://docs.pycom.io/gettingstarted/>. [Accessed 05. 04. 2022].
- [61] Security Industry Association, “DC-07 Going Through SIA Public Review,” [Online]. Available: <https://pdfcoffee.com/dc-07-going-through-sia-public-review-pdf-free.html>. [Accessed 05. 04. 2022].
- [62] M. Scharrer, “crccheck 1.1,” [Online]. Available: <https://pypi.org/project/crccheck/>. [Accessed 05. 04. 2022].
- [63] Stack overflow, “Convert CRC16 CCITT code from C to Python,” [Online]. Available: <https://stackoverflow.com/questions/67115292/convert-crc16-ccitt-code-from-c-to-python>. [Accessed 05. 04. 2022].
- [64] “Online CRC Calculator,” [Online]. Available: <https://crccalc.com/>. [Accessed 05. 04. 2022].
- [65] North Latitude Technology, “SIA DC-09 Message Generator,” [Online]. Available: <https://dc09gen.northlat.com/>. [Accessed 05. 04. 2022].

## **Appendix 1 – Non-exclusive licence for reproduction and publication of a graduation thesis<sup>1</sup>**

I Andree Orasson

1. Grant Tallinn University of Technology free licence (non-exclusive licence) for my thesis "Affordable Internet of Things Solution for Transmitting SIA DC-09 Messages", supervised by Ivo Mürsepp
  - 1.1. to be reproduced for the purposes of preservation and electronic publication of the graduation thesis, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright;
  - 1.2. to be published via the web of Tallinn University of Technology, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright.
2. I am aware that the author also retains the rights specified in clause 1 of the non-exclusive licence.
3. I confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights, the rights arising from the Personal Data Protection Act or rights arising from other legislation.

09.05.2022

---

<sup>1</sup> The non-exclusive licence is not valid during the validity of access restriction indicated in the student's application for restriction on access to the graduation thesis that has been signed by the school's dean, except in case of the university's right to reproduce the thesis for preservation purposes only. If a graduation thesis is based on the joint creative activity of two or more persons and the co-author(s) has/have not granted, by the set deadline, the student defending his/her graduation thesis consent to reproduce and publish the graduation thesis in compliance with clauses 1.1 and 1.2 of the non-exclusive licence, the non-exclusive license shall not be valid for the period.

# Appendix 2 – Printed circuit board drawings

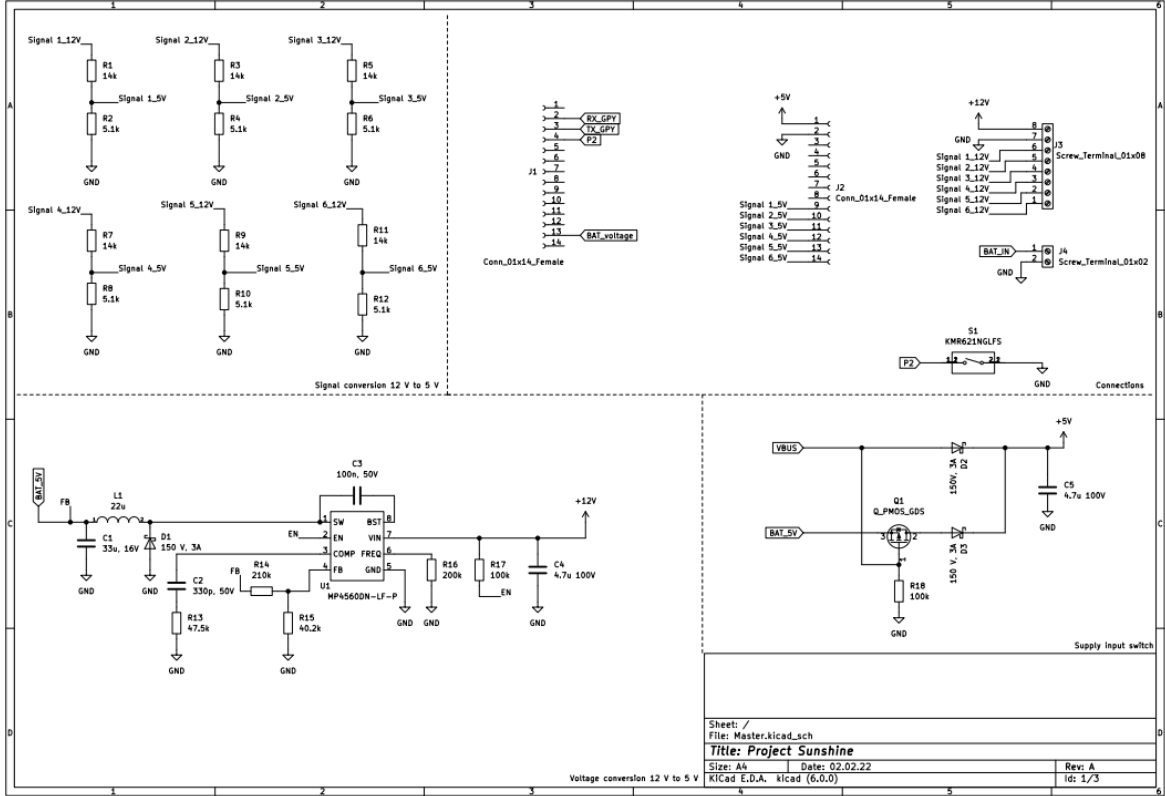


Figure 21. Printed circuit board drawing sheet 1.

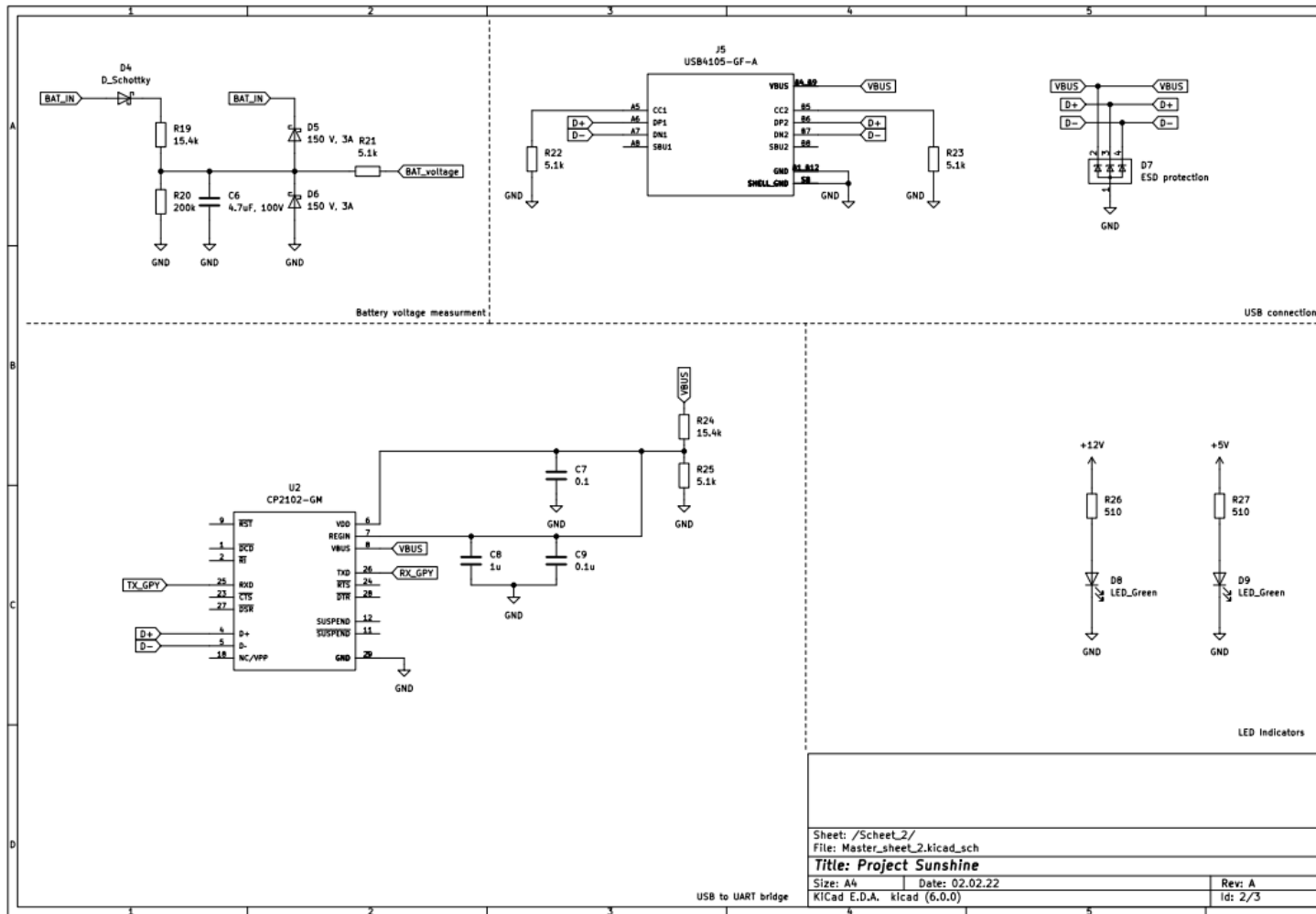


Figure 22. Printed circuit board drawing sheet 2.

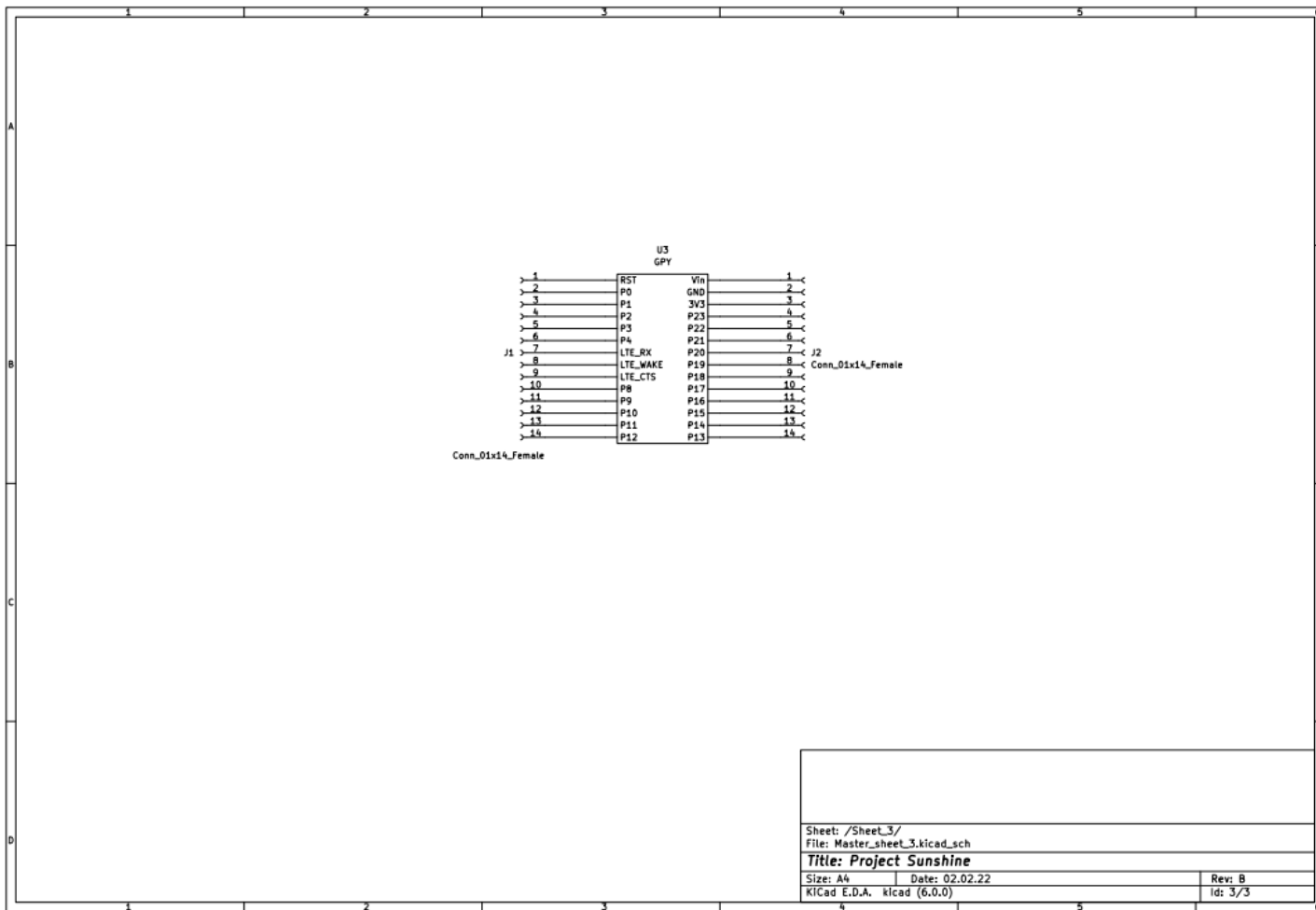


Figure 23. Printed circuit board drawing sheet 3.

## Appendix 3 – Printed circuit board bill of the materials

Table 4. Printed circuit board bill of the materials.

<b>Product</b>	<b>Subdivision</b>	<b>Value</b>	<b>Quantity</b>	<b>Unit price</b>	<b>Total price</b>
Resistor	Signal conversion	15.4k	6	0,74 €	4,45 €
Resistor	Signal conversion	5.1k	6	0,04 €	0,22 €
Wire-To-Board Terminal Block	Board To Board		1	3,45 €	3,45 €
Board-to-Board	Board To Board		2	0,36 €	0,72 €
Wire-To-Board Terminal Block	Board To Board		1	0,91 €	0,91 €
Switching Regulator	LDO		1	4,92 €	4,92 €
Inductor	LDO	22uH	1	1,57 €	1,57 €
Diode Schottky	LDO	150V,3A	1	0,54 €	0,54 €
Capacitor	LDO	33uF, 16V	1	1,02 €	1,02 €
Capacitor	LDO	4.7uF, 100V	1	0,63 €	0,63 €
Capacitor	LDO	330pF, 50V	1	0,07 €	0,07 €
Capacitor	LDO	0.1uF, 50V	1	0,07 €	0,07 €

Table 5. Printed circuit board bill of the materials.

Resistor	LDO	47.5k	1	0,04 €	0,04 €
Resistor	LDO	210k	1	0,06 €	0,06 €
Resistor	LDO	40.2k	1	0,04 €	0,04 €
Resistor	LDO	200k	1	0,01 €	0,01 €
Resistor	LDO	100k	1	0,02 €	0,02 €
Diode Schottky	Supply input switch	150V,3A	1	0,54 €	0,54 €
Diode Schottky	Supply input switch	150V,3A	1	0,54 €	0,54 €
MOSFET	Supply input switch	60 V, 5.1A	1	0,86 €	0,86 €
Capacitor	Supply input switch	4.7uF, 100V	1	0,63 €	0,63 €
Resistor	Supply input switch	100k	1	0,02 €	0,02 €
Resistor	Battery voltage measurement	100k	1	0,02 €	0,02 €
Resistor	Battery voltage measurement	15.4k	1	0,08 €	0,08 €
Resistor	Battery voltage measurement	5.1k	1	0,04 €	0,04 €
Capacitor	Battery voltage measurement	4.7uF, 100V	1	0,63 €	0,63 €
Diode Schottky	Battery voltage measurement	150V,3A	1	0,54 €	0,54 €
Diode Schottky	Battery voltage measurement	150V,3A	1	0,54 €	0,54 €
Diode Schottky	Battery voltage measurement	150V,3A	1	0,54 €	0,54 €
USB	USB connection	USB port	1	0,85 €	0,85 €
Resistor	USB connection	5.1k	2	0,04 €	0,07 €



Table 6. Printed circuit board bill of the materials.

ESD Protection Devices	USB connection		1	1,02 €	1,02 €
Botton	USB connection		1	0,36 €	0,36 €
USB Interface	USB to UART bridge		1	5,54 €	5,54 €
Capacitor	USB to UART bridge	0.1uF, 50V	2	0,07 €	0,13 €
Capacitor	USB to UART bridge	1u, 25V	1	0,07 €	0,07 €
Resistor	USB to UART bridge	15.4k	1	0,08 €	0,08 €
Resistor	USB to UART bridge	5.1k	1	0,04 €	0,04 €
Resistor	LED Indicators	510	2	0,01 €	0,02 €
LED	LED Indicators	20 mA	2	0,09 €	0,17 €

## Appendix 4 – GPy with printed circuit board

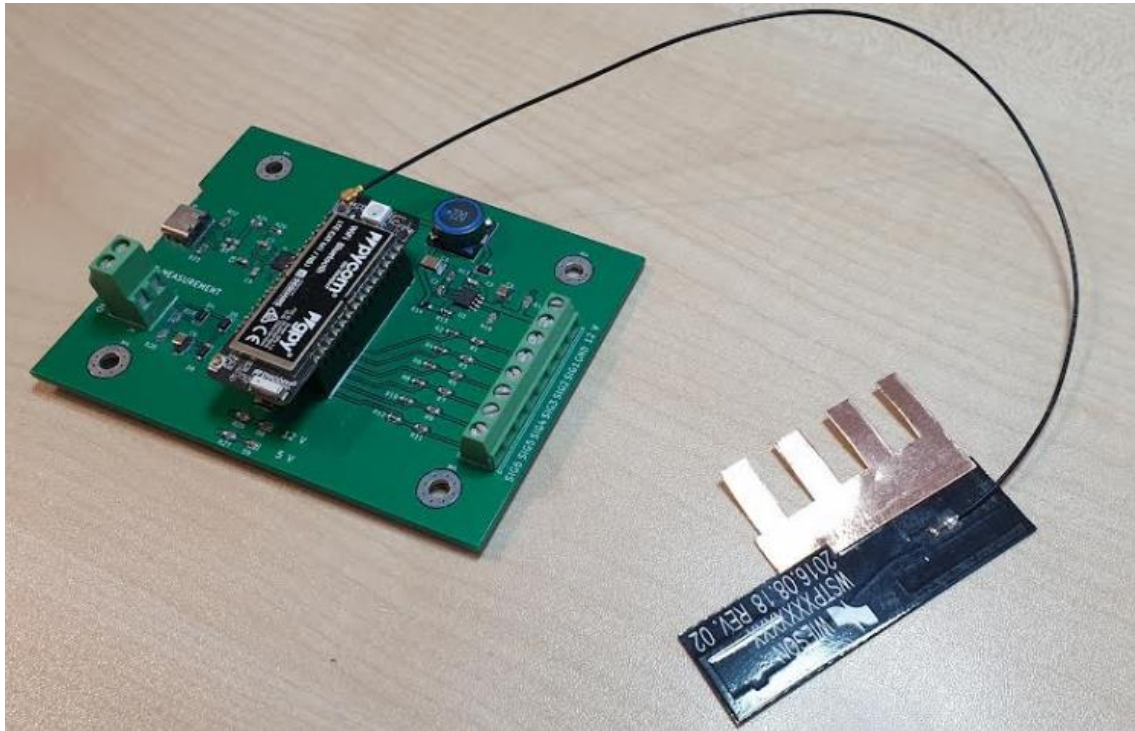


Figure 24. The GPy with printed circuit board

## Appendix 5 – The boot.py program code

```
import pycom
import time

from network import LTE
lte = LTE()
pycom.heartbeat(False)

time.sleep(2)
lte.send_at_cmd('AT+CPIN="0000"')
time.sleep(0.1)
lte.attach(band=20, apn="internet.emt.ee")

while lte.isattached():
    pycom.rgbled(0xff00)
```

Figure 25. The boot.py program code.

## Appendix 6 – The main.py program code

```
import time
import socket
import pycom
from machine import Pin

PORT = 8001
FORMAT = 'cp1252'
SERVER = "84.50.164.143"
ADDR = (SERVER, PORT)
client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
client.connect(ADDR)

acct = "1234"
counter = 9990
Lpref = "L0"

led = Pin('P9', mode = Pin.OUT)
Test = Pin('P12', mode = Pin.OUT)
#button = Pin('P10', mode = Pin.IN)
#Test.value(1)

PIN1 = Pin('P18', mode = Pin.IN)
PIN2 = Pin('P17', mode = Pin.IN)
PIN3 = Pin('P16', mode = Pin.IN)
PIN4 = Pin('P15', mode = Pin.IN)
PIN5 = Pin('P14', mode = Pin.IN)
PIN6 = Pin('P13', mode = Pin.IN)

def send(msg):
    message = msg.encode(FORMAT)
    print(message)
    client.sendall(message)
    print(client.recv(512).decode(FORMAT))

def crc16(data : bytearray): # offset , length
    crc = 0
    length = len(data)
    for i in range(0, length):
        crc ^= data[0 + i] #data[offset + i]
        for j in range(0,8):
            if (crc & 1) > 0:
                crc = (crc >> 1) ^ 0xA001
            else:
                crc = crc >> 1
    return crc

def message_crc(event):
```

Figure 26. The main.py program code.

```

    messagecrc = str("SIA-DCS"+ '%04d' % counter + Lpref + "#" + acct +
"[" + acct + event + "]" ) #
    messagecrc_enocde = messagecrc.encode()
    crc = crc16(messagecrc_enocde)
    crc_hex = hex(crc)
    return crc_hex

while True:

    if (PIN1() == 1 and is_active == False):

        is_active = True
        counter = counter + 1
        event1 = "|NFA129"
        event1_1 = message_crc(event1)
        msg_length = len( '%04d' % counter + Lpref + acct + acct +
event1 )
        print(msg_length)
        lsls = "\n" + event1_1.upper()[2:] + '%04d' % msg_length + "SIA-
DCS" + '%04d' % counter + Lpref + "#" + acct + "[" + acct + event1 + "]"
+ "\r"
        print(lsls)
        send(lsls)
        pycom.rgbled(0x7f0000)

    if (PIN1() == 0):
        is_active = False
        client.close()

    if (PIN2() == 1 and is_active_2 == False):

        is_active_2 = True
        counter = counter + 1
        event2 = "|NFA128"
        event2_1 = message_crc(event2)
        msg_length = len( '%04d' % counter + Lpref + acct + acct +
event2 )
        print(msg_length)
        lsls = "\n" + event2_1.upper()[2:] + '%04d' % msg_length + "SIA-
DCS" + '%04d' % counter + Lpref + "#" + acct + "[" + acct + event2 + "]"
+ "\r"
        print(lsls)
        send(lsls)
        pycom.rgbled(0x7f7f00)

    if (PIN2() == 0):
        is_active_2 = False
        client.close()

```

Figure 27. The main.py program code.

```

if counter == 9999:
    counter = 0

if (PIN3() == 1 and is_active_3 == False):
    is_active_3 = True
    counter = counter + 1
    event3 = "|NFA127"
    event3_1 = message_crc(event3)
    msg_length = len( '%04d' % counter + Lpref + acct + acct +
event3 )
    print(msg_length)
    ls1s = "\n" + event3_1.upper()[2:] + '%04d' % msg_length + '"SIA-
DCS"' + '%04d' % counter + Lpref + "#" + acct + "[" + acct + event3 + "]"
+ "\r"
    print(ls1s)
    send(ls1s)

if (PIN3() == 0):
    is_active_3 = False

if (PIN4() == 1 and is_active_4 == False):
    is_active_4 = True
    counter = counter + 1
    event4 = "|NFA126"
    event4_1 = message_crc(event4)
    msg_length = len( '%04d' % counter + Lpref + acct + acct +
event4 )
    print(msg_length)
    ls1s = "\n" + event4_1.upper()[2:] + '%04d' % msg_length + '"SIA-
DCS"' + '%04d' % counter + Lpref + "#" + acct + "[" + acct + event4 + "]"
+ "\r"
    print(ls1s)
    send(ls1s)

if (PIN4() == 0):
    is_active_4 = False

if (PIN5() == 1 and is_active_5 == False):
    is_active_5 = True
    counter = counter + 1
    event5 = "|NFA125"
    event5_1 = message_crc(event5)
    msg_length = len( '%04d' % counter + Lpref + acct + acct +
event5 )
    print(msg_length)
    ls1s = "\n" + event5_1.upper()[2:] + '%04d' % msg_length + '"SIA-
DCS"' + '%04d' % counter + Lpref + "#" + acct + "[" + acct + event5 + "]"
+ "\r"
    print(ls1s)
    send(ls1s)

```

Figure 28. The main.py program code.

```

if (PIN5() == 0):
    is_active_5 = False

if (PIN6() == 1 and is_active_6 == False):
    is_active_6 = True
    counter = counter + 1
    event6 = "|NFA124"
    event6_1 = message_crc(event1)
    msg_length = len( '%04d' % counter + Lpref + acct + acct +
event6 )
    print(msg_length)
    ls1s = "\n" + event6_1.upper()[2:] + '%04d' % msg_length + '"SIA-
DCS"' + '%04d' % counter + Lpref + "#" + acct + "[" + acct + event6 + "]"
+ "\r"
    print(ls1s)
    send(ls1s)

if (PIN6() == 0):
    is_active_6 = False

```

Figure 29. The main.py program code.