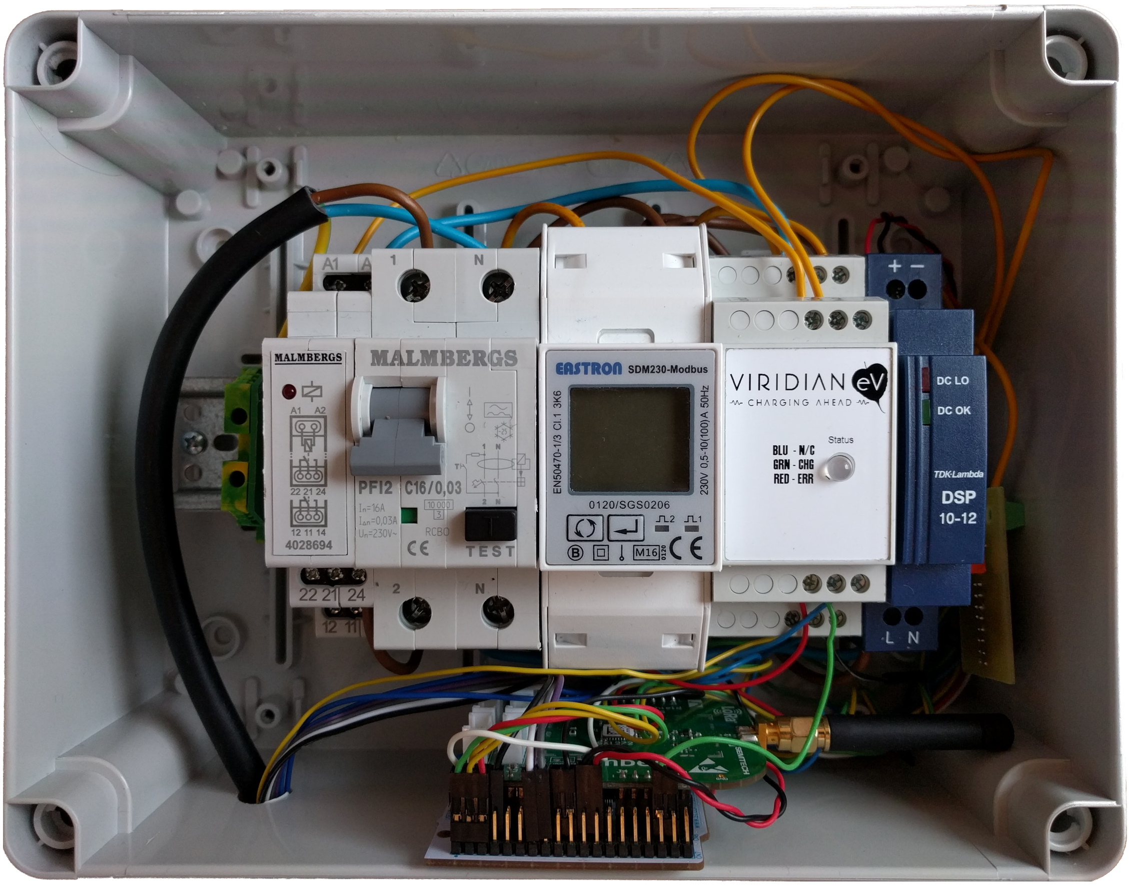


# Controlled Charging of Electrical Vehicles on Residential Power Grid

Master's thesis in Embedded Electronic System Design

JOHAN HINDÉR

SIMON WIJK

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2018

Master's thesis 2018

# Controlled Charging of Electrical Vehicles on Residential Power Grid

JOHAN HINDÉR
SIMON WIJK

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2018

Controlled Charging of Electrical Vehicles on Residential Power Grid
JOHAN HINDÉR
SIMON WIJK



Supervisor: Lena Peterson, Department of Computer Science and Engineering
Advisor: Joakim Fernstad, Endian Technologies AB
Advisor: Tord Andersson, Endian Technologies AB
Examiner: Per Larsson-Edefors, Department of Computer Science and Engineering

Master's Thesis 2018
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

The picture on the cover is the smart electric vehicle charging box prototype created in this project.

Typeset in LaTeX
Gothenburg, Sweden 2018

Controlled Charging of Electrical Vehicles on Residential Power Grid
JOHAN HINDÉR
SIMON WIJK
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

# Abstract

The market for electric vehicles (EVs) is growing explosively and shows no signs of stopping. With more and more owners of EVs, a significant load increase on the power grid is to be expected. EV owners in residential areas often share a connection to the power grid, where the maximum current is split evenly among a number of electric vehicle supply equipments (EVSEs) sharing the connection. If the EVSEs using this connection could have a common knowledge about how much power is available at the power grid connection and shared this among currently connected EVSEs the available power would seem higher than if the available current is shared equally among residents not using it for the moment.

This thesis report covers an attempt to make an intelligent EVSE that can be installed at already existing charging lots in residential areas where the parking lots are located far apart. The EVSEs should use a wireless connection to remove the cost of laying new cables and enable easy installing. The report contains a comparison of new long-range communication technologies to find the best suited solution for the intelligent EVSE. The charging shall be controlled by a centralised server, from where charging can be enabled and disabled based on a schedule or human interaction.

With this prototype system we hope to enable easy and low-cost installation of smart EV charging in residential areas.

Keywords: LoRa, LoRaWAN, electric vehicle, smart grid, EV charging, EVSE.

## Acknowledgements

We would like to thank Endian Technologies AB for hosting us during this thesis and giving us the possibilities to implement our work on a real situation. We would like to thank our supervisor Lena Peterson for the all the feedback on the report an having patience with us during our slow progress on this thesis. We would than and our company advisor Joakim Fernstad for always having good ideas and feedback on both implementation and writing, and Tord Andersson for all help with ideas when things went wrong.

And lastly we would like to thank our examinator Per Larsson-Edefors for all the help and feedback received during this thesis and still having interest in our report when receiving it so late.

Johan Hindér, Gothenburg, 09 2018
Simon Wijk, Gothenburg, 09 2018

# Contents

# Acronyms

**APB** authentication by personalisation. 10

**CP** control pilot. 3–5

**CSS** chirp spread spectrum. 9

**DDoS** distributed denial-of-service. 2

**DE** drive output enable. 7

**DI** data input. 7

**EPC** electric vehicle protocol controller. 5, 29

**EV** electric vehicle. v, 1–5, 18, 29, 31

**EVSE** electric vehicle supply equipment. v, 1–4, 18, 19, 28–30

**IC** input current. 5

**IoT** internet-of-things. 2

**LPWAN** low-power wide-area network. 9

**M2M** machine-to-machine. 9

**NB-IoT** narrowband IoT. 12

**OTA** over-the-air. 10

**OTAA** over-the-air activation. 10

**PE** protective earth. 3–5

**PHY** physical-layer. 9

**PKI** public key infrastructure. 10

**PP** proximity pilot. 3

**$\overline{\textbf{RE}}$** receive output enable. 7

**RO** receiver output. 7

**TTL** transistor-transistor logic. 7

# 1

# Introduction

The market for EVs is growing explosively and shows no signs of stopping [1]. EVs are charged using a charging station, also known as EVSE. The EVSE controls the maximum charging current that the EV charger is allowed to draw. EVSEs are often controlled independently with a fixed maximum load. These EVs are increasing the demands on the power grid [2].

If all EVSEs at a parking lot had the ability to cooperate, the load could be balanced for all present vehicles. Vehicles not used more during the day could be charged during the night when the electricity prices and demand are at their lowest. Higher current could be allocated to vehicles that need to be charged faster while lowering supply for the rest.

Recently much research has been done into algorithms and technologies for optimally charging vehicles on a shared grid [2–5]. To be able to implement these charging algorithms the EVSEs need communication between each other. Wireless communication would be preferable for existing groups of EVSEs since it gets rid of the need to lay new cables for every EVSE.

EVSEs sharing a power connection can be spread out over distances that are too long to be covered by Bluetooth, Wi-Fi or other common wireless communication technologies. There is however a growing number of wireless communication technologies that operate in the sub-GHz range which can communicate over distances larger than a kilometre [6]. This thesis projects aim is to design and build a charging controller for EVSEs distributed over a residential area sharing a power grid connection. The controller should prioritise charging when the demand on the grid is low. Prior research in the field of distributed EV-charging should be used to achieve this [3].

## 1.1 Problem statement

The highest demand on the electric grid is in the evening between end of the working day and midnight [7]. This is the time when most EVs are parked and connected for charging. If charging could be moved to low-load times during the night it would reduce demand on the power grid and result in reduced cost for charging the EVs.

Residential power grids have a maximum allowed current drain. To not exceed this limit, all connected EVSEs must limit the current so that when all EVSEs are in use, the total current drain does not exceed the limit. The easiest solution would be a static limit for each EVSE. However, with dynamic current regulation of each EVSE, the maximum current drain for the charging EVs can be increased, lowering the charging times.

EVSEs in a residential area are located at each house or group of houses. But these houses and EVSEs still share the same power grid. Because of the distances, long-range communication up to at least one kilometre is required to reach and control all EVSEs. Recently some new long-range low-power communication technologies meant for low-throughput industrial applications have entered the market [6].

In a network of nodes that communicate with each other it is important to consider security, privacy and trust. This is a hot topic especially for internet-of-things (IoT) devices [8]. There have been cases recently where massive numbers of compromised IoT devices are responsible for distributed denial-of-service (DDoS) attacks. One of these attacks managed to shut down large parts of the Internet for several hours by targeting a DNS provider [8].

The problem statement can be summarised in the following research questions:

- Which functionality is required to control charging of EVs sharing residential power grid and balance the charging depending on the amount of connected vehicles for the highest utilisation of the available current?
- How can we combine the charge control functionality with EV charging during the night to lower the peak load on the power grid?
- How do we make a prototype that can guarantee that the EVs will be fully charged for use every morning?

## 1.2 Aim

The aim of this thesis is to investigate the field of kilometre-range communication technologies available today. Using this research a prototype system should be developed, the prototype shall contain a central server, a user interface and a number of EVSEs. It should be used to distribute current for charging EVs in a residential power grid.

## 1.3 Limitations

The goal of this project is to develop and demonstrate a prototype for centralised charging of EVs, however, this project should not attempt to make a working product.

Most of the low-throughput communication technologies that will be evaluated are also made for low-power operation. Optimising the microcontroller code for low power is time consuming and not necessary in this project.

There are many long-range sub-GHz technologies on the market, to test all of them requires lots of hardware (gateway, transceiver, controller) and would be expensive and time consuming. Because of this, we shall only compare specifications and use previous studies.

# 2
# Theory

In this chapter we present theory about the different parts and technologies used in this project. Section 2.1 explains the theory for the connection and communication with the EV charger. Section 2.2 explains the theory for recording the energy consumption of the charging EVs.

## 2.1 Electric vehicle supply equipment (EVSE)

The EVSE is the equipment connecting the EVs to the power grid. The EVSE includes connectors, cables, conductors and other devices installed specifically with the purpose of enabling safe transfer of energy between the power grid and the EV battery [9].

### 2.1.1 Charging cable and contact

There are three different connector types used in the EV industry today. They are SAE J1772 [10] known as type one or Yazaki contact (manufacturer), VDE 0623-5-2 [11] known as type two or Mennekes contact (manufacturer) and JEVS G105-1993 [12] known as type four or CHAdeMO contact. The third type of contact is a proposal connector produced by EV Plug Alliance [13], which was competing with the Mennekes contact and follows the same IEC 62196 standard but uses another contact layout.

The Yazaki contact is a single-phase connector defined by SAE J1772 [10] standard. It includes three pins for AC connection; live, neutral and protective earth (PE) and two smaller pins for communication with the EV; proximity pilot (PP) and control pilot (CP) as shown in Figure 2.1.

The Mennekes contact is a single and three-phase connector following the IEC 62196-2 standard [11]. The IEC 62196-2 standard includes the pins from the SAE J1772 specification and adds two extra AC pins for the two extra phases. The VDE-AR-E 2623-2-2 contact is shown in Figure 2.2.

### 2.1.2 Charge control

The EVSE communicates with the built-in battery charger in the EV and specifies the amount of current it is allowed to drain from the grid.

The maximum charging current depends on the rating of the charging cable and the current available at the EVSE. The rating of the charging cable is read by

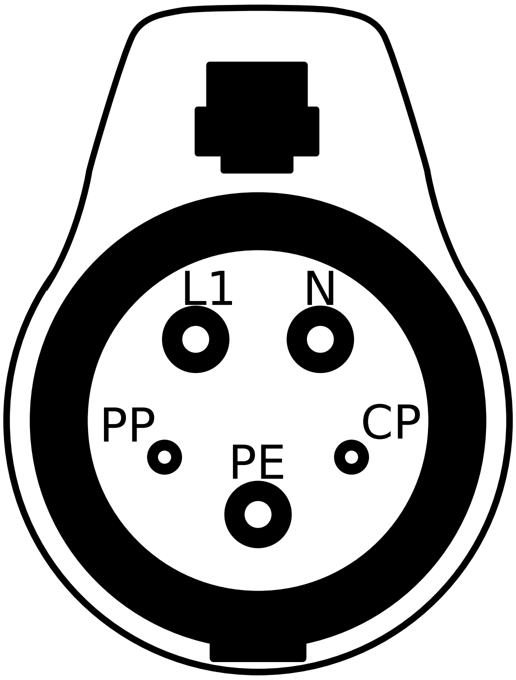


**Figure 2.1:** SAE J1772 ("Type 1") male contact.

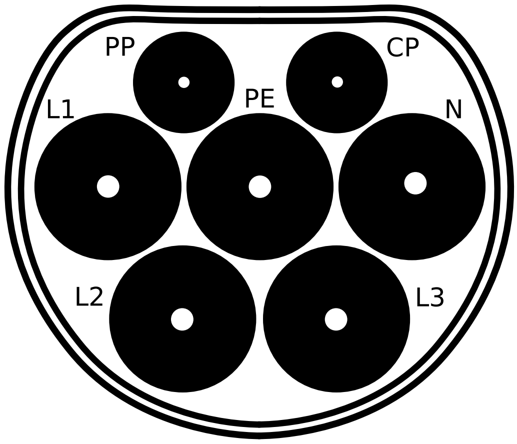


**Figure 2.2:** VDE-AR-E 2623-2-2 ("Type 2") male contact.

the EV charger by measuring the resistance between the PP and PE signals. The current available at the EVSE is communicated over the CP signal while the EV charger uses resistance between CP and PE to communicate charging state. The EV changes this resistance depending on the state of the charging. The EVSE then reads the state by measuring the voltage drop over this resistance. The different states are represented in Table 2.1.

**Table 2.1:** EV states read by resistance between CP and PE

| State | | Resistance Ω |
|---|---|---|
| A | No EV connected | Open |
| B | EV present | 2740 |
| C | EV charging | 882 (1300 \|\| 2740) |
| D | EV charging (vent[1]) | 246 (270 \|\| 2740) |
| E | No power | NA |
| F | EVSE Error | NA |

When the EV charger is connected, charging state (B, C or D), the EVSE starts to control the duty cycle of the PWM signal on the CP pin. The duty cycle represents the maximum allowed current drain by the EV charger following (2.1) for 6 A to 51 A and (2.2) for 52 A to 80 A.

$$\text{Current}[A] = \text{Dutycycle} \cdot 0.6 \tag{2.1}$$

[1]Some EV batteries needs ventilation while charging

$$\text{Current}[A] = (\text{Dutycycle} - 64) \cdot 2.5 \tag{2.2}$$

### 2.1.3 Viridian EVSE Protocol Controller (EPC)

The Viridian electric vehicle protocol controller (EPC) is a controller module for easily connecting and controlling EV charging [14]. The EPC controls the communication with the EV charger. The state of the EV charger is read by measuring the resistance between CP and PE and the current limitation is communicated with a PWM signal between the contacts input current (IC) and 0 V on the device. The EPC is demonstrated in Figure 2.3 with all connection terminals marked.

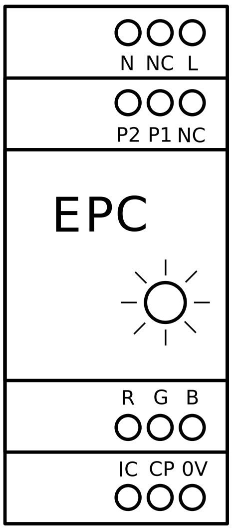


**Figure 2.3:** Viridian EPC for "one phase" layout

The EPC represents the charging state with three 5 V signals (B, G and R) or a three coloured LED (blue, green and red). Only one signal is active at a time and the signal can either be constant high or flashing with 1 Hz frequency. Table 2.2 describes the signals for each state of the EV charging. The current limitation is controlled by changing the resistance between the IC contact and 0 V. The resistance of 191 Ω generates a voltage of 0.8018 V between IC contact and 0 V and represents 6 A charging for the EV [14] while a resistance of 9090 Ω generates a voltage of 4.5045 V and represents 80 A maximum charge.

**Table 2.2:** EV states output by the EPC [14]

| State | | Signal |
|---|---|---|
| A | Not connected | Flashing B/blue |
| B | Connected EV | Constant B/blue |
| C | EV charging | Constant G/green |
| D | Ventilation required | Constant R/red |
| E | Not powered | No signal |
| F | Error | Flashing R/red |

### 2.1.4 Digital potentiometer: AD8400

A digital potentiometer is a digital controllable resistance. It is not continuously variable as an ordinary potentiometer, but is rather divided in digital taps (steps) [15]. These steps can be selected through serial interface such as I2C or SPI. The selection is done by writing the digital code for a requested resistance to the digital potentiometer. The digital code is translated to a position for a wiper inside digital potentiometer. By moving the wiper, the resistance between the end point and the wiper changes.

The AD8400 is a 1-channel, 256 taps, linearly divided 1 kΩ digital potentiometer controlled through SPI [16]. The AD8400 has a terminal on each side of variable resistance (terminal A and B) and the wiper moves between these two terminals. The resistance between the wiper terminal and terminal A is the opposite of the resistance between the wiper and terminal B (1 kΩ minus the resistance of wiper to terminal one). The lowest possible resistance is 50 Ω (tap value zero) and each tap moves the wiper approximately 4 Ω.

## 2.2 Easton SDM230-Modbus RTU energy meter

Easton SDM230-Modbus RTU is an energy meter providing the energy readings through a built-in display and over RS485 modbus [17]. There are 48 input registers holding the energy data, each data value is stored as 32-bit IEEE754 floating-point number [18] and is therefore taking two register for each data value, most significant register first. There are also nine holding registers containing settings for the energy meter.

### 2.2.1 Modbus

Modbus is a serial communication protocol standard used on the application layer in the OSI model. It can be used on multiple different psychical layers such as Ethernet, RS485/RS232 or other serial communication techniques [19]. The Modbus protocol is a request/reply protocol where a master/server sends a request message to a device and the device answer the master with a data response.

There are three different message types; request, response and exception response. The message is built up of four segments; an address, a function code , a data segment and an error checking segment.

The Modbus request message contains the address to the slave, a function code, defining what kind of request there is, a data field with length depending on what kind of function there is and last two bytes of CRC.

The slave answers with the same address and function code in the address and function field, so the master knows where the response come from. The response data is dependent on what function was called. And the CRC ends the message.

In case of an error in the request the slave answers with an exception response. This response also starts with the address but the function code is changed to the same code but the MSB is flipped to one. The data part contains an exception code depending on the error.

The function code 0x04 is for reading input registers. This function code is sent with four bytes of data. The first two bytes contain the address of the register on the device, the second two bytes are for representing the number of register that should be read after the first address. The example 0x0002 represents reading two registers starting on the "address" and reading "address" +1. The response to this request is function code 0x04 together with one byte representing the numbers of data bytes to be received. In this case four bytes (each register is two bytes). This is followed by the data, the register of the address first. All data uses big-endian representation, that is most significant byte comes first.

### 2.2.2 RS485

RS485, also known as TIA-485-A or EIA-485, is an electrical characteristics standard for a type of serial communicating [20]. It communicates using differential receiver levels for representing zeroes and ones. A negative voltage of 200 mV between signals A and B represents a logic one and a positive voltage of 200 mV represents a logic zero [20]. RS485 can use two wires A and B for half duplex communication or four wires for full duplex communication, using one pair for transmitting and one pair for receiving. A network is set up with one driver/master, who decides who can utilise the signal wires, and multiple devices/slaves that always listens and only respond when the master ask them for information.

### 2.2.3 SP3485 RS-485 Transceiver

The SP3485 is a RS-485 transceiver converting transistor-transistor logic (TTL) serial data to RS-485 and back. The SP3485 communicates over TTL with four signals; data input (DI), drive output enable (DE), receive output enable ($\overline{\text{RE}}$) and receiver output (RO) [21]. When the DE signal is high the SP348 converts the data on DI signal to RS-485 output on pin A and B. When the $\overline{\text{RE}}$ is low the SP348 converts the RS-485 signals on pins A and B to a TTL signal on pin RO.

# 3

# Long-range wireless communication technologies

Low-power wide-area networks (LPWANs) are wireless communication technologies for machine-to-machine (M2M) communication which are designed to have very long range and low power consumption. To achieve this most LPWANs have low data rates and operate at sub-GHz frequencies.

The primary use case for LPWAN technologies is to get sensor data from places where wired internet or Wi-Fi is not available. With the low power requirements, many companies advertise battery life of up to and beyond ten years.

This chapter will elaborate on LPWAN technologies and will describe the technologies relevant to this project. As a conclusion, different LPWAN technologies will be compared and a decision will be made on which technology to be used in this project.

## 3.1 Sub-GHz ISM-bands

Many LPWAN technologies use unlicensed sub-GHz frequency bands that are open for anyone to use. These bands use different frequencies for different parts of the world: 915 MHz in the Americas and 433/868 MHz in Europe. In Europe there are regulations for how the band can be used, most notably you are not allowed to have a duty-cycle greater than 1% per device per hour [22]. This means that you are only allowed to have 36 seconds of airtime each hour. The duty cycle limit is removed if the end device implements "Polite spectrum access" which means that the device has to listen to the channel it is about to transmit on to see if it is free, if the channel is busy the device has to switch channel or wait until it is free before transmitting.

## 3.2 LoRa

LoRa is a physical-layer (PHY) proprietary modulation technology from Semtech that focuses on long range and low power consumption. LoRa is a chirp spread spectrum (CSS) technology, this means that the data is encoded in sinusoidal pulses with continuously increasing frequency over time. With CSS the entire allocated bandwith is used to transmit the signal making it resilient against noise. LoRa operates in the sub-GHz unlicensed bands.

### 3.2.1 LoRaWAN

LoRaWAN is a MAC-layer specification [23] for LoRa created by the LoRa Alliance. LoRaWAN is open and free to use. It uses a star topology network architecture where the end-devices communicate directly with a LoRa gateway that is connected to the internet.

With a bandwidth of 125 kHz in Europe LoRaWAN sends messages at 0.3 kbit/s to 5.5 kbit/s [24], the bit rate depends on how far the message needs to be sent.

LoRaWAN features symmetric encryption [23] using AES with 128-bit key length. The messages are encrypted twice, the data is encrypted with an application key and the whole message is then encrypted with a network key. The gateway knows the network key and uses it to decrypt the message headers, using the headers the message is forwarded to the correct target server on the internet. The target server knows the application key and uses it the decrypt the message data.

There are two methods available to activate end-devices on a LoRaWAN network. Over-the-air activation (OTAA), each session the end-devices goes through a join procedure with the network where it is assigned network and application keys. The second activation method is authentication by personalisation (APB), using this method the network and application keys are generated before deployment and are stored on the end-devices. Using this method no join procedure is required.

Both uplink and downlink messages are possible using LoRaWAN, there are three different modes of operation available for end-devices called Class A, B and C. Class A is the default mode and using this the end-devices are initiating the communication with an uplink transmission followed by two short downlink receive windows where the gateway can transmit data to the end-device. Class B adds additional receive windows at scheduled times enabling downlink messages without waiting an uplink message from the end-device. Using Class C the end-devices are receiving nearly continuously, the only times they are not receiving is when they are transmitting.

LoRaWAN uses adaptive data rates to find the correct balance between speed and range.

### 3.2.2 Symphony Link

Symphony Link is a proprietary protocol built by Link Labs using the LoRa PHY. It was created to compete with LoRaWAN and Symphony Link has some big advantages. Encryption is handled using a public key infrastructure (PKI) with Diffie Hellman algorithm and AES encryption [25]. There is no duty cycle limit on Symphony Link since it implements "Polite spectrum access". All messages in Symphony Link are acknowledged (ACK). To keep ACKs from taking up valuable airtime the gateway multicasts multiple ACKs in a single transmission. Symphony Link has built in support of over-the-air (OTA) firmware upgrade. This protocol is currently only available in the USA on the 915 MHz band.

## 3.3 Sigfox

SigFox is a proprietary worldwide ultra-narrowband communication service provided by the company with the same name. The technology uses the Sub-GHz ISM-bands and modulates the signal with binary phase-shift keying (BPSK) for uplink and Gaussian frequency-shift keying (GFSK) for downlink [26].

Sigfox is deployed as one large network covering large parts of Europe and some locations in America, Asia and Australia. SigFox works much like the cellular network, in each country there is a carrier responsible for selling subscriptions to the network. There are different levels of subscriptions depending how many uplink and downlink messages that are needed per day.

Downlink messages in SigFox is possible but very limited. To send a message to an end device, the downlink message first has to be sent from the user application to the SigFox backend where it is queued. The message is sent when the end device sends an uplink message with an acknowledgement flag set, this flag informs the back end that the end device will be in receive mode for 30 seconds.

Message sizes in Sigfox are 12 bytes for uplink and 8 bytes for downlink. Sigfox occupies a bandwidth of 100 Hz in Europe and 600 Hz in the USA. The uplink data rate is 100 bit/s or 600 bit/s depending on region.

## 3.4 Weightless-P

Weightless-P is one of Weightless Special Interests Groups (SIG) open LPWAN standards.

Weightless-P uses Gaussian minimum shift keying (GMSK) and offset quadrature phase-shift keying (QPSK) modulation with a 12.5 kHz bandwidth and it can operate over the entire sub-GHz spectrum. The data rate is adaptive from 0.2 kbit/s to 100 kbit/s depending on how far the message need to be sent, the range in an urban environment is up to 2 km.

Communication is bidirectional and fully acknowledged, if a message is dropped it is retransmitted automatically. Authentication to the network is handled with keys using AES 128/256 encryption.

## 3.5 Waviot NB-Fi

NB-Fi stands for Narrowband Fidelity and is a proprietary LPWAN protocol owned by Waviot. The physical layer is a radio module also developed by Waviot and uses narrow band technique and BPSK for signal modulation [27]. The NB-Fi module is available in multiple frequency bands from 315 MHz to 915 MHz depending on region. The available data rate is between 10 bit/s to 100 bit/s per node [28] and a gateway supports up to 5000 transmitting devices simultaneously. The average latency is 30 seconds for uplink and 60 seconds for downlink communication [27].

## 3.6 Ingenu RPMA

Random phase multiple access (RPMA) is LPWAN technology from Ingenu that unlike most other LPWAN technologies utilizes the 2.4 GHz band. RPMA is a direct-sequence spread spectrum (DSSS) technology with code-division multiple access (CDMA). They claim up to 18 km range for one base station, there is no way to set up your own RPMA basestation and currently they only operate in the USA.

## 3.7 LTE Cat-M1

LTE Cat-M1 is a completely new technology standard from 3GPP that builds on the existing LTE technology. The difference between LTE Cat-M1 and LTE is that it uses a bandwidth of 1.4 MHz channel instead of the full 20 MHz. LTE Cat-M1 only uses one antenna and is half-duplex which means that it cannot receive and transmit at the same time. These changes make the hardware cheaper and lower the power consumption. Still being compatible with existing LTE networks means that this technology has good coverage in large parts of the world. LTE Cat-M1 benefits from high data rates of up to 1 Mbit/s. Because it is compatible with LTE base stations the end-devices can connect to any server using a TCP/IP connection, this however requires the end-device to be much more secure, having to implement IP security. LTE Cat-M1 has support for full mobility which means that a device can move between base stations without having to redo a join request.

Trials of LTE Cat-M1 is being deployed around the world, USA will be first with a nation wide network deployed by Verizon which is planned to be completed ending Q1 of 2017. Since this technology operates in the licensed spectrum there will be monthly subscription fees to use it.

## 3.8 NB-IoT

Narrowband IoT (NB-IoT) is another new technology standard from 3GPP that is designed to fit inside a 200kHz GSM/UMTS channel. It supports uplink speeds of up to 144 kbit/s and is half duplex. NB-IoT can be deployed in three different ways, stand-alone in a GSM band, in the guard bands of LTE, or in a LTE band. NB-IoT has no support for mobility so when a device changes base station it will have to redo the join request. NB-IoT devices will not have a direct TCP/IP connection to the internet like LTE-Cat-M1, it will have a similar soloution to LoRaWAN where the gateway converts the packages and puts them on the Internet. NB-IoT will be deployed by mobile carriers in the licensed spectrum which means that there will be a monthly subscription fee to use it.

## 3.9 Cellular networks

Regular 2G/3G cellular networks obviously have much better coverage than any of the other communication technologies mentioned above. The drawback is that each device must have a SIM card and pay for the data being transmitted/received.

## 3.10 Comparison and decision

This section will compare the different technologies with respect to this project's requirements and in the end a decision will be made on which LPWAN technology to use.

### 3.10.1 Project requirements

For this prototype to be successful the communication link will have to fulfil some requirements. The communication technology should have a range of at least one kilometre. The reason for this is that parking lots in urban areas can be large and typically not be located close enough to an internet connection for Wi-Fi to work.

Since this project involves building a working prototype, hardware for the communication technology will have to be available to buy. It must also be possible to self-host a gateway for the communication technology or good coverage in urban areas available from carrier hosted communication technologies.

The communication technology must have some form of built in security in the form of encryption of data and authentication to join networks. To ensure that the data that is being transferred is secure and to prevent any malicious acts against the prototype.

The price per device should be reasonable to allow for small scale production, less than 50 euro per communication device.

### 3.10.2 Comparison

An elimination matrix is used to eliminate the technologies that could not be used at all for this project, table 3.1 shows this matrix. This removes all but the cellular, LoRaWAN and Waviot NB-Fi communication technologies. A Pugh matrix is used to decide which of the three technologies to use, seen in table 3.2. LoRaWAN is the best fit for this project because of the possibility to host your own gateway combined with the wide market adoption and multitude of end devices to choose from.

**Table 3.1:** Elimination matrix for possible communication technologies.

| | Range >1 km | Encryption and authentication | Available in Sweden | Hardware available | Comment |
|---|---|---|---|---|---|
| Cellular | X | X | X | X | |
| Ingenu RPMA | X | X | | | Only available in the USA |
| LTE-Cat-M1 | X | X | | | Only trials in Sweden so far |
| LoRaWAN | X | X | X | X | |
| Waviot NB-Fi | X | X | X | X | |
| NB-IoT | X | X | | | Not available in Sweden yet |
| SigFox | X | X | | | Coming to Sweden 2018 |
| Symphony Link | X | X | X | | No hardware for Europe available |
| Weightless-P | X | X | X | | Not possible to buy hardware yet |

**Table 3.2:** Pugh matrix for deciding communication technologies.

| | Cellular | LoRaWAN | Waviot NB-Fi |
|---|---|---|---|
| Self hostable | - | + | + |
| Gateway cost | 0 | + | - |
| Subscription cost | - | 0 | 0 |
| Device cost | - | + | + |
| Market adoption | + | + | - |
| Device options | + | + | - |
| Grade | -1 | 5 | 0 |
| Rank | 3 | 1 | 2 |

# 4
# Implementation

This chapter describes how the prototype is implemented as a whole, starting with the components in each EVSE node, section 4.1, followed by the communication link, section 4.2, and how the centralised control system is implemented, section 4.3.

## 4.1 EVSE node

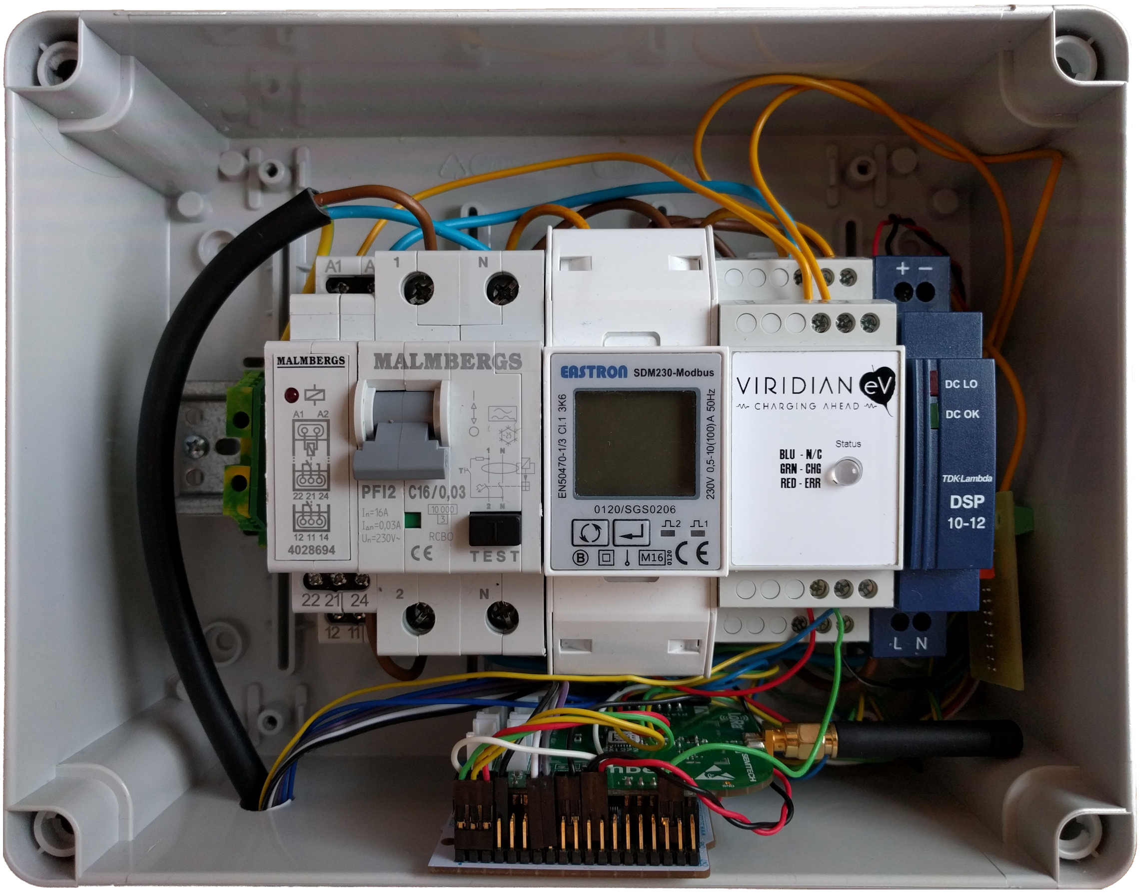


**Figure 4.1:** Picture of the EVSE

Each EVSE node contains seven components handling communication to the EV, energy metering and communication to the control system. A Viridian EPC [14] handles the communication between the microcontroller and the EV. An Easton

SDM230 Modbus [17] energy meter records the energy consumption and communicates with a microcontroller using Modbus over RS485. A 16 A circuit breaker protects against excessive current draw. A contactor controls the connection of the charging contacts on the charging plug. A 230 VAC-to-12 VDC converter, powers the microcontroller. A microcontroller with SX1272 LoRa module handling the communication and interfacing to the centralised control system and the components in the node. In extension to the microcontroller is a PCB containing RS485 and digital potentiometer chips and a signal relay for controlling the signal to the contactor.

### 4.1.1 Charging State

The Viridian EPC indicates the current charging state of the EV charger by three output signals. These signals are connected to the MCU and each signal is handled by an interrupt handler. The interrupt handlers register rising edges of the signals. On an interrupt the pin is read twice with ten millisecond separation to ensure the validity of the interrupt. If the interrupt is valid the pin number of the interrupt signal is stored and a 490 ms timer is started. When that timer runs out the signal is checked once more. If the signal is low there is a pulsing signal on the pin. Depending on the pin (R, G or B) and if there is a pulsing or solid signal the state is updated in the MCU, see table 2.2 for state description. This process is restarted if there is an interrupt from any of the R, G or B pins during the state evaluation. There does not exist a state with the green LED flashing, so when there is an interrupt on the G pin the 490 ms timer is skipped. The handling of state changes is illustrated in Figure 4.2. When a state is changed, a message is sent to the load balancer.

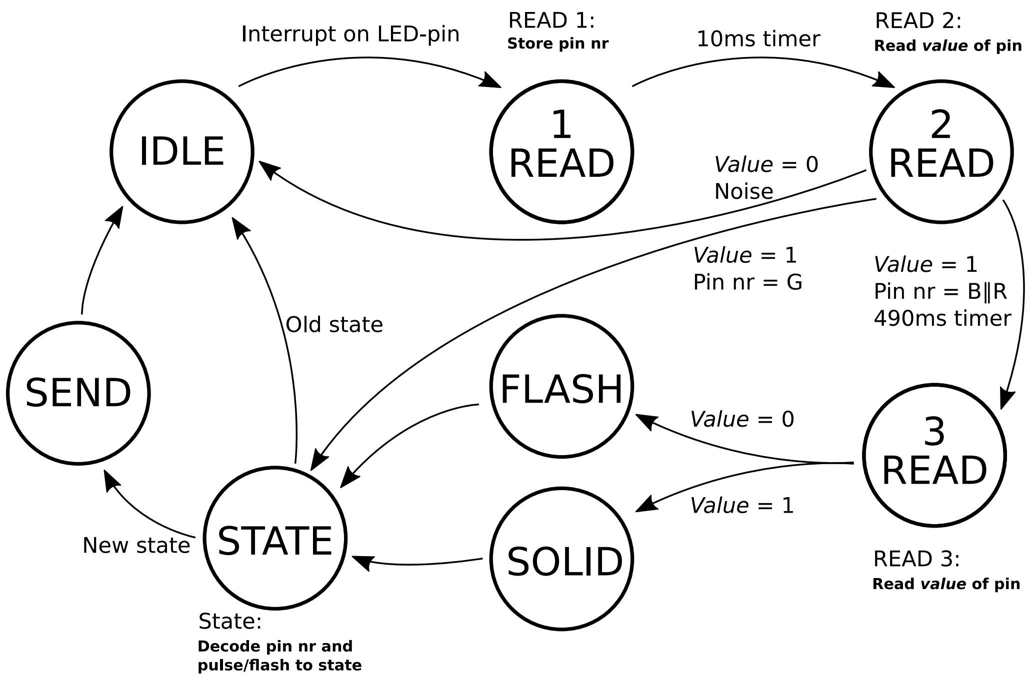


**Figure 4.2:** State machine handling the readings from the EPC LEDs. A new interrupt on a LED-pin will reset the state machine to Read 1.

During start up of the EVSE node the EPC flashes twice in each colour to show correct start up sequence. The LED interrupt handlers are therefore disabled during the start up sequence.

For testing the state readings from the EPC, an EV charger is simulated with a diode in series with three different resistances depending on the desired state, 2740 Ω, 1300 Ω and 270 Ω following Table 2.1. The state of the EPC is changed by connecting the different resistances between the 0 V and CP connectors.

### 4.1.2 Controlling current

To set the available current which the EPC communicates to the EV charger the resistance between IC and 0V on the EPC must be controlled. This is done with the 1 kΩ digital potentiometer AD8400 [16] from Analog Devices.

The wiper terminal and terminal B are connected to the 0V and IC contacts on the EPC. The MCU has digital codes for the correct resistances stored in an array and depending on the allowed charge current it chooses the respective digital code. This are then sent to the digital potentiometer and a response is sent back to the server to acknowledge that the current limit was updated.

When the resistance value is updated the EPC LED flashes on all three LED outputs, during this time the LED interrupt handlers are disabled to avoid any incorrect state change readings.

The resistance steps of the digital potentiometer are much smaller than the resistance step between each IC value on the EPC. This leaves most of the steps unused in the digital potentiometer. However the precision is necessary around the needed resistance values to avoid setting an incorrect charging current.

### 4.1.3 Energy readings

The Easton SDM230-Modbus energy meter communicates using Modbus over RS485. The microcontroller sends the modbus messages using a built-in UART module at TTL levels which are converted to RS485 using an SP3485 [21] RS-485 transceiver.

When a request for energy data is sent, a predefined buffer containing address, function code and registers is sent to the SP3485 chip. The SP3485 chip translates the data to RS485 and sends it to the energy meter. The energy meter answers with the data, in case of an error or an incorrect CRC is received, the request is sent again.

### 4.1.4 MCU and LoRa shield

The MCU used in the prototype is a Nucleo L476RG [29] devboard together with a SX1272 LoRa radio module shield [30] for LoRa communication. The Nucleo contains both the LoRaWAN stack and additional code for controlling and communicating with the components in the EVSE node.

### 4.1.5 Extension PCB

The extension PCB contains three components, a SP3485 transceiver for RS-485 communication with the energy meter, a digital potentiometer AD8400 for controlling the resistance between the Viridian EPC IC and 0 V contacts, and a signal relay for controlling a 230 VAC control signal to the contactor. The signal relay is supplemented with a flyback diode and transistor control to protect the MCU from voltage spikes and current drains when switching the relay. A 3D model of the PCB is displayed in Figure 4.3. Schematics and layout for the PCB can be found in the appendix A.

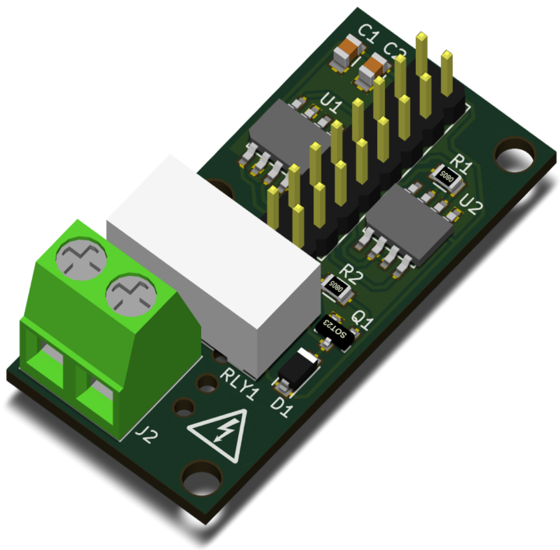


**Figure 4.3:** PCB containing digital potentiometer AD8400 (upper left chip), SP3485 transceiver (right chip) and a signal relay (white block) with connected screw terminals for AC connection.

### 4.1.6 Contactor

The contactor in the EVSE is used for switching the AC connection on and off to the EV. The contactor is controlled using a 220 V AC signal. This signal is in turn controlled from the MCU using the relay on the extension PCB.

### 4.1.7 LoRaWan modifications

The LoRaWAN specification says that the minimum time between two messages is 99 times the time on air for the first message. This rule exists to enforce the ISM limitation of 1 % duty cycle per hour. A drawback with the way the LoRaWAN specification implements the 1 % rule is that it is impossible to burst multiple small messages in a short time.

When an EV is connected to a EVSE there must be multiple messages uplink and downlink within a short time span. These messages are short (14 bytes with header) and only needed to be sent when an EV is connected (usually not more than once an hour). To be able to send these messages, we had to rewrite the LoRaWAN stack so that it allows for bursting multiple messages in a short amount of time while still following the ISM rules of 1 % duty cycle per hour.

This is done by storing all "time on air" data for new messages in an list together with an time stamp for each message. When something is to be sent first the list is updated, removing all messages that have a time stamp older than one hour and then all the "time on air" data is summed up. If the total sum of "time on air" plus the "time on air" for the new message is lower than 36 s (1 % duty cycle per hour) the new message is sent and stored last in the list. If the list is full, (36 s) or more, the message is delayed and not added to the list.

## 4.2 LoRaWAN communication

The LoRaWAN messages between the EVSE nodes and the control system is separated in four types of uplink messages and three types of downlink messages. These message types is separated with use of a meta-data tag named "fport". The uplink messages uses fport 1 for energy messages, fport 2 for status messages, fport 5 for charge current acknowledgement and fport 6 for requesting fast charge from the control system. For the downlink messages port three is used for status request from the control system, port four for new current limits and port seven for turn off the fast charge request from the EVSE node.

## 4.3 Centralised control system

In this section we will begin with describing the layout and purpose of the centralised control system and how the different components communicate with each other. After that we will go into more detail how each component is implemented and how they work.

### 4.3.1 System overview

The centralised control systems have two tasks, the first one is to receive status messages from the LoRa connected EVSEs and calculate the current limits for each EVSE that wants to charge the connected EV. The other task is to provide a user interface where the user can request prioritised charging and view energy consumption and current draw.

The system is made up of five components, the first one is the LoRa gateway which receives and transmits the Lora messages from and to the EVSEs. The messages are converted to MQTT and sent to the network server which decrypts the messages, it also handles the over the air authentication of new LoRa devices to the network.

If the messages are from a authenticated EVSE they are forwarded to the load balancer using MQTT. The load balancer calculates new current limits for the EVSEs in the same group as the one that sent the message, the new limits are sent downlink to each affected EVSE. The current and energy data received from the EVSEs are also forwarded to the charging load balancer which inserts them into a SQL database.

The user interface is accessed on any device with a web browser, the web server is connected to the load balancer with MQTT to send prioritised charge requests. The energy and current data is fetched by the web server from the SQL database and displayed as graphs in the browser. A visualisation of this system can be seen in Figure 4.4.

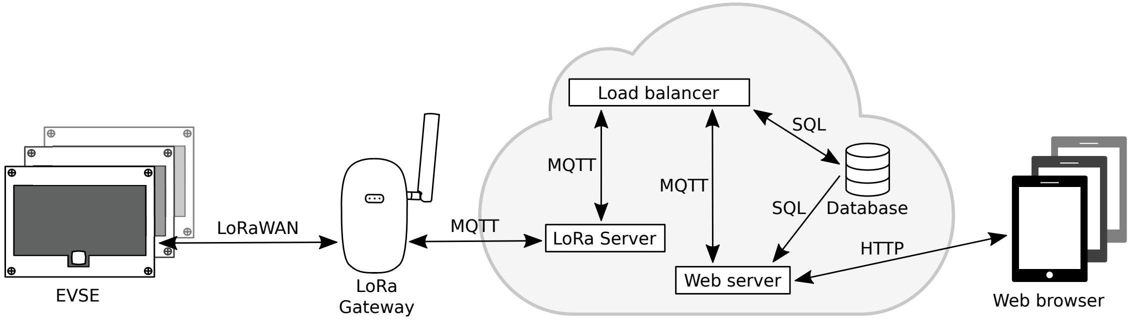


**Figure 4.4:** Overview of the communication between the components of the centralised control system.

### 4.3.2 LoRa gateway

The LoRa gateways task is to relay messages between the end devices and the network server. Uplink LoRa radio messages from the end devices are converted into UDP packages and sent to the network server. For downlink messages its the same but in reverse.

The gateway that we decided to use for this prototype is the Wirnet iFemtocell from Kerlink. It is a small form factor indoor gateway with built in Wi-Fi and Ethernet. It costs 300€ which is relatively cheap (it costs about the same as building one from scratch using a Raspberry Pi). The gateway runs Linux and is accessed using SSH.

Since UDP messages are not acknowledged by the receiver there is a risk for packet loss between the gateway and network server. To solve this problem we installed a UDP to MQTT converter on the gateway and sent the messages to a network server with support for MQTT. Since MQTT is transmitted using TCP any dropped packages are resent by the transmitter eliminating the packet loss in this part of the communication chain.

### 4.3.3 LoRa Server

The network servers task is to decrypt/encrypt messages and authenticate connections from new end devices. The decrypted messages are sent to the target application server and messages received from application servers are encrypted and sent to the gateway. New devices are authenticated by checking if their ID is approved by any application, if it is approved the device is handed a unique key to encrypt messages with.

We chose to use an open source network and application server called LoRa Server created by Orne Broocaar. It has support for Class C devices and uses

MQTT for communication between all of its components. The network server has support for both OTAA and ABP. The application server has a web browser based user interface that is used to create new applications and add approved device ids to them. The application server has a REST API where messages can be received and transmitted. It is also possible to transmit and receive messages using MQTT topics set up by LoRa Server.

A MQTT broker is needed to send and receive MQTT packages. We decided to use an open source broker called Mosquitto.

### 4.3.4 Programming language

To implement the load balancer and user interface a decision had to be made on which language to use. To save development time it must be possible to implement both the load balancer and user interface using the same language.

The requirements on the language are that it should be possible to deploy it on a Linux server, it should have a MQTT library and it should be possible to build a web server with it to serve the user application.

The languages that were considered was C#, Erlang, Go, Java, Javascript, PHP, Python and Ruby. The language we decided to use is Google's language Go since it is very well suited to be used as a web server and a server application running on Linux.

### 4.3.5 Load balancer

The task of the load balancer is to calculate current limits for EVSEs with connected EVs that needs charging. It should be able to prioritise charging for EVs that request priority charging and should prioritise charging during the night.

When the load balancer is started it fetches all registered EVSEs and their properties from the database. After that a connection to the MQTT broker is established. The connection to the MQTT broker is handled using Paho, a open source MQTT library for Go. The load balancer subscribes to the MQTT topic where LoRa Server publishes all LoRa messages the gateway receives. After this state requests are sent downlink to all EVSEs to find if they are running and which state they are in. Finally the load balancer enters an infinite loop waiting for MQTT messages from the EVSEs.

When a MQTT message is received the load balancer first decodes the base 64 encoded payload and then checks which LoRaWAN port it was sent on. The port decides how the load balancer should act on the payload.

If the port is "1" the payload is an energy reading from a EVSE, the load balancer inserts the energy and current values into the database with corresponding device ID and a time stamp.

If the port is "2" the message is a state update from a EVSE, a state change will require re-calculation of the current limit for all EVSEs in the same group as the one changing state. To calculate the new current limits the groups maximum current is divided by the combined weight of all EVSEs on that group, this is the new current limit for an EVSE with the weight of 1. The individual current limit for each EVSE is calculated by multiplying the weight of the EVSE with the new current limit. The new individual limits are sent downlink to the EVSEs using the LoRa Server.

If the port is "5" a EVSE has acknowledged a downlink message sent by the load balancer.

Finally if the port is "6" the message is a priority charging request sent from a user pressing the button on a EVSE. The load balancer gives the EVSE priority by setting its weight to four. After this the new current limits for EVSEs on the same group are calculated and sent downlink. An illustration of the load balancer's state transitions can be viewed in Figure 4.5

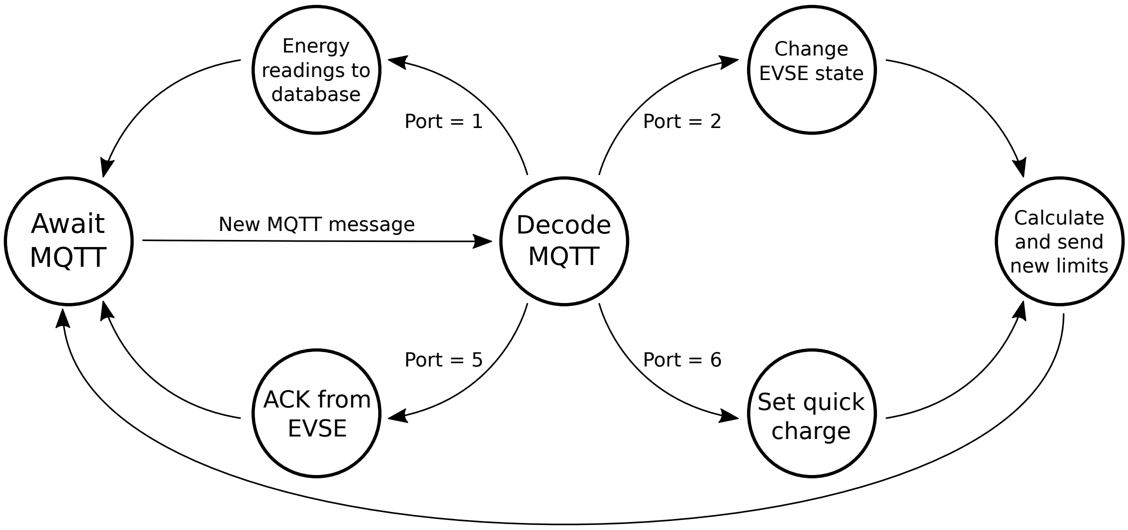


**Figure 4.5:** State transition diagram for the load balancer.

The load balancer reads a configuration file on startup that contains a start and stop time. These two times define between which hours the EVSEs are allowed to charge. The load balancer orders all EVSEs to stop charging by setting their weight to zero and then re-calculate the current limits which leads to all EVSEs getting zero amperes as limit. If a user requests prioritised charging however the weight is set to four and the EVSE is allowed to charge.

### 4.3.6 User interface

The user interface was implemented as a web page because it works on all devices with a web browser. The web page shows energy and current graphs and has a button to activate prioritised charging. To serve the web page to the user we created a web server using Go. When a user wants to view the page for a EVSE the web server gathers the related energy and current data from the database and inserts it into a web page which is served to the user.

The layout and design of the web page is created using Googles Material Design guidelines, because of this the web page scales seamlessly with the screen size of the device viewing it. This is important because the web page is meant to be used on a smart phone, which will have screens of very different sizes.

There are two different views available, one is a view of a specific EVSE which shows the energy and current graphs and has the prioritised charging button, this view can be seen in Figure 4.6. The other view is an overview of all registered EVSEs and groups, this view is meant to be used by the system administrator to add, remove or edit EVSEs and groups. The administrator view can be seen in Figure 4.7.

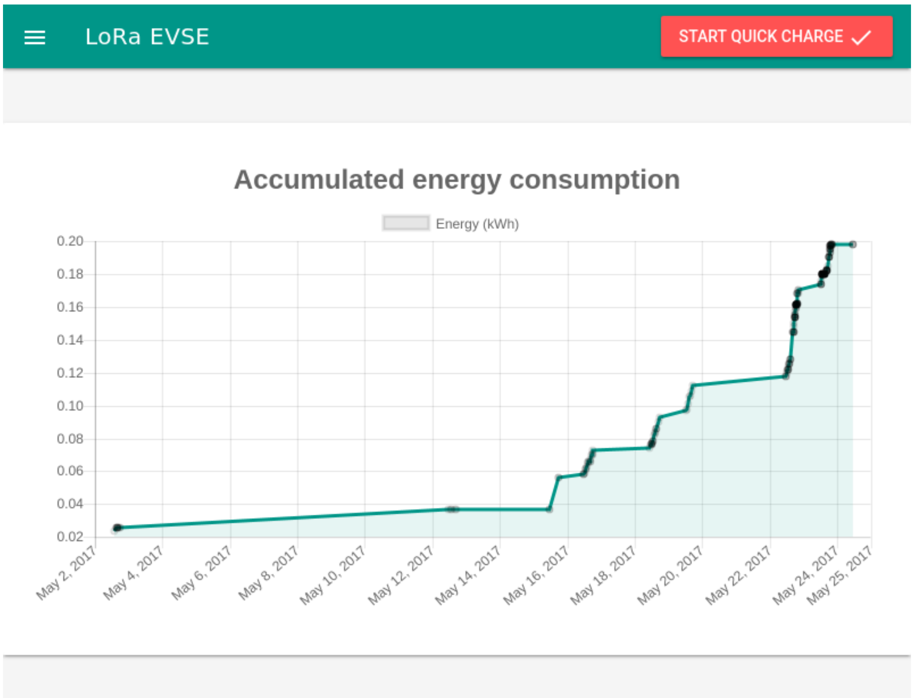


**Figure 4.6:** The user interface for a specific EVSE, used to view energy consumption and to request prioritised charging.

The graphs are drawn by a open source javascript library called Chart.js, the data is supplied as JSON by the web server.

When the user clicks the prioritised charging button it is sent as a HTTP POST request containing the device id and a bool describing if it was a request to turn on or off prioritised charging. The web server routes the request to the prioritised charging request handler, this handler sends a message over MQTT to the load balancer requesting prioritised charging for the EVSE in question. The load balancer responds with the new prioritised charging state to the web server which in turn replies to the original POST request showing the state to the user by changing the text in the button to "Stop quick charge".

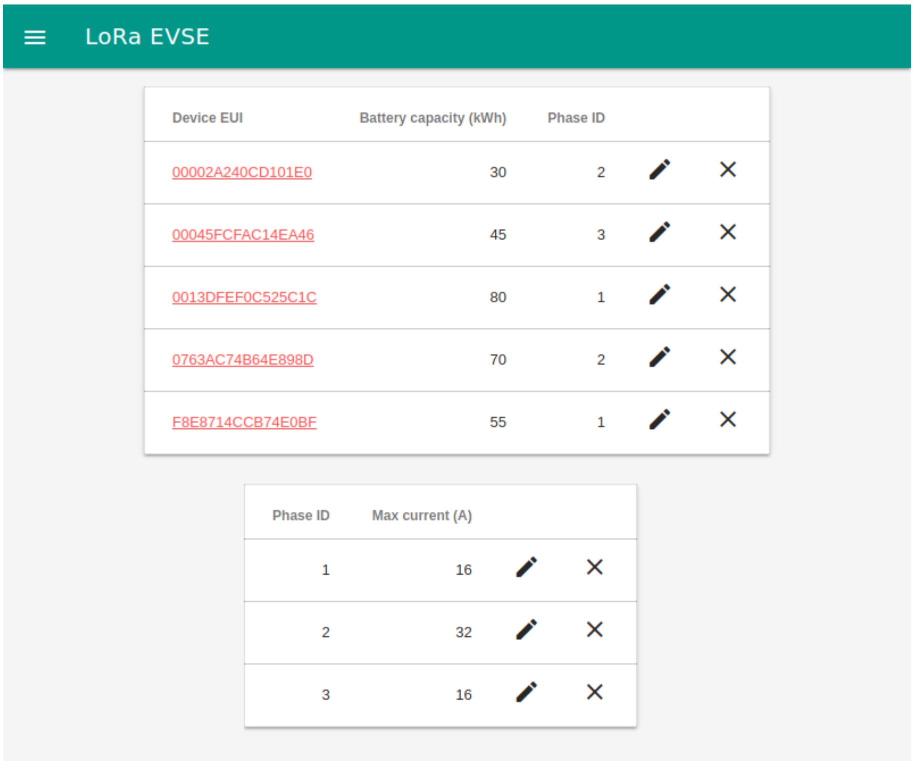


**Figure 4.7:** Interface for system administrators used for deleting and adding EVSEs and groups.

# 5

# Results

The result chapter brings up results for individual parts of each sub system, EVSE nodes in section 5.1, LoRa result in section 5.2, control system results in section 5.3 and ends with a summary of how the prototype meets the goals we set in the beginning of the project, section 5.4.

## 5.1 EVSE

The project used two EVSE test nodes but has scalability to add more nodes through the web interface. This section will contains result for the different functions of the EVSE nodes have.

### 5.1.1 LEDs reading

The state updates from the EV charger is read from the EPC with half a second delay for all states but state "C". This because of that all LEDs but the green LED has states represented by flashing 1 Hz or solid signals.

### 5.1.2 Digital potentiometer

When the microcontroller updates the digital potentiometer and the EPC is in state B or C it changes the PWM modulation of the signal on the CP pin. This accordingly to the resistance values requested by the MCU.

The digital codes of the resistances for the digital potentiometers used in the two EVSE nodes differs, thus making the translation array of digital codes specific for each EVSE node. The digital codes for the two EVSE nodes resistances can be found in table 5.1

### 5.1.3 Voltage spikes

When charging begins the signal relay switches a AC signal to the contactor which in turn switches the AC connection to EV charger. These switches create disturbance that some times disable the LoRa module. After this disturbance, the MCU still manage to communicate with the LoRa module but the radio fail to send any messages to the gateway. The disturbance exists when any of the relays that switch AC has any galvanic connection to the LoRa module. The EPC, the signal relay and one independent relay development board has been tested independently and all three causes the disturbance that causes the LoRa module to fail.

**Table 5.1:** Digital code for the two digital potentiometers used in the two EVSE nodes.

| Current (A) | Resistance (Ω) | Digital code | |
|---|---|---|---|
| | | Node 1 | Node 2 |
| 6 | 191 | 33 | 26 |
| 7 | 205 | 36 | 29 |
| 8 | 221 | 39 | 33 |
| 9 | 237 | 43 | 36 |
| 10 | 249 | 46 | 39 |
| 11 | 267 | 51 | 43 |
| 12 | 280 | 54 | 47 |
| 13 | 301 | 59 | 52 |
| 14 | 316 | 62 | 55 |
| 15 | 332 | 66 | 59 |
| 16 | 348 | 70 | 63 |

### 5.1.4 Energy meter

The energy data is received from the energy meter and sent to the control system once an hour without any problems.

## 5.2 LoRa

The low through put of a LoRa network was not a problem for the implementation. There is no big amount of data that needs to be communicated. The reliability of the communication link is however critical and very depending on node placement. If the node has a lot of obstacles in close range to the antenna the transmission has higher probability to fail even in short ranges to the gateway. A node with free air in close range tend to handle much longer distances without failing in transmission. Parking lots tend to be flat and the EVSE nodes can therefore be placed with mostly free air all the way to the gateway, resulting in high probability of good connectivity.

### 5.2.1 Range test

To find out if LoRaWAN meet this projects range requirement we carried out a range test. We started with the included indoor antenna mounted on the gateway. With this setup we achieved a maximum range of one kilometre. Since LoRaWAN is meant for outdoor use we mounted an outdoor antenna and continued the test. With this antenna we got a maximum range of 5.3 kilometre in our test. The test locations can be seen in Figure 5.1.

The testing environment was very varied with industrial, urban and countryside terrain all in the 5.3 kilometre. The positioning off the LoRa node was more crucial than what spreading factor was used for sending the message. A message node in mid to short range from the gateway that was placed with bad radio conditions was as likely to fail with SF7 as with SF12. A node with good conditions usual manage to send the data on all spreading factors. The change in reliability depending on spreading factor started to show a little first with the longer ranges (over one kilometre). The data from the testing can be found in Appendix B

We can conclude that LoRaWAN clearly meets the range requirements for this project from this test with even the lowest spreading factor.


**Figure 5.1:** Range test locations represented as red dots.

### 5.2.2 Message delays

The load balancer worked as expected but we quickly realised that delays were going to be a problem because of how LoRaWAN works. When a EV is plugged in to start charging the load balancer has to send new current limits to all other charging EVs on the same group before the EV can start charging. Two messages will be sent for each charging EV, one downlink with the new current limit and one uplink that acknowledges the new limit. These messages are one byte, it takes one second to transmit one byte of data with spreading factor 12. This means that each EV that is charging on the same group adds a delay of at least two seconds before the most recently connected EV will start charging. Any dropped dropped messages would also add delays of one second each. These delays will significantly impact the user experience, the user might have to wait for several seconds before the car starts charging.

There is no way of getting rid of these delays and still allow the slower spreading factors. The delays could be mostly eliminated if we were to only use spreading factor 7 which only takes 46 ms to send one byte.

## 5.3 Centralised control system

The open source network and application servers for LoRaWAN by Orne Brocaar was easy to set up and worked without any problems for the entire duration of the project.

The programming language Go ended up being a very good choice to implement the load balancer and the user interface. It was easy to learn thanks to the excellent introduction website "A Tour of Go" by the Go authors. Go was as expected well suited to run on a Linux host and do the load balancing calculations and communicate with the LoRa network server using MQTT messages. The user interface was also easy to implement as a web application using Go, much because of the abundance of good examples on the internet. Thanks to Google's Material Design framework the user interface ended up looking good as well.

## 5.4 Summary

The prototype controls the maximum available current of every connected EVSE node and share the current between the nodes as intended. The fast charging functionally gives the EVSE node four times the weight compare to a normal EVSE node when the current is shared between the EVSE nodes. When a EV is connected to a EVSE the control system it makes sure that all already charging EVSEs is lowered to the new lower limit before the new EV is allowed to start charging. The control system can be controlled so that all the weights of the EVSEs is zeroed between different time stamps so that charging is only enabled during the night or if fast charging is requested.

# 6
# Conclusion

## 6.1 Discussion

In this section we discuss problems, solutions and possible new development of the different components in this project.

### 6.1.1 EPC

When the current limit is updated the EPC quickly turns all LED signals on and off three times. During this time the interrupt handlers are disabled to make sure that this flashing does not trigger an incorrect state change. This workaround could result in a missed state change if the car switches state during the time the interrupt handlers are disabled. This problem should not occur in normal use of the EVSE, but should anyways be handled in a finished product.

For larger scale production of these EVSE nodes the control values for the digital potentiometer should be the same so that no individual EVSE node configuration is needed. It is possible that this is the case for most digital potentiometers and that our case just was a bad example, but this has not been tested.

A possible solution for most of our problems with the EPC would have been to remove the EPC as a unit of our system and implement the communication directly to the EV. The communication protocol only consists of a resistance that need to be read on the EV side and a 1 kHz $\pm 12$ V PWM signal that needs to be generated on the controller side. This solution however would have required full access to the IEC 62196 standard and probably a lot more work with certifications to get a finished product.

### 6.1.2 AC voltage spikes

Switching the AC power lines to the EV charger on and off causes voltage spikes on all conductors galvanically connected to the contactor. The size of these spikes seems to depend on the load of the switching power line. There are multiple possible solutions that may work for lowering the spikes to a manageable level which does not disable the LoRa module.

If the EV communication protocol had more control over the EV charger the maximum charge current when switching the AC power lines could be set to zero instead of the lowest current rating of six ampere. The EV charger is probably not drawing the maximum current at once when the EV is connected to the EVSE but this is nothing we can guarantee for different implementations of EV chargers.

Another possible solution would be to ensure galvanic isolation between the radio module and the rest of the EVSE. This could be done by replacing the AC signal relay with a solid state relay and adding opto-isolators to the EPC LED- and RS485-signals. This way the only galvanic connection would be the power supply witch in turn given some time probably could have been filtered to get a working module without spikes.

### 6.1.3 EV communication protocol

The standard protocol for communication charging information (IEC-62196) is missing lots of functionality for enabling smart charging. As only the state and maximum charge current is shared between the EV and the EVSEs the possibilities to make the charging smart is limited. If the protocol instead could be a digital protocol where information such as battery capacity and state of charge would be shared the collaboration between multiple EVSE nodes would work better. A car with a small battery could be allotted a lower maximum current while a car with much capacity to charge could be allotted a higher one. The current used for charging the last percents of a battery is also lower than the current used for charging an empty battery. If this could to be communicated to the EVSEs this could be used for lowering the available current for a EV that does not need it and give it to a EV that is not close to being fully charged.

The current each connected EV draws could be measured by adding a current meter for each charging station and then actively changing the maximum allowed current to follow what the EV actually draws. This would remove the loss in allocated current for each EV but adds both complexity in the EVSE and the control system while increasing the amount of data that needs to be sent between the EVSEs. This increase in data and real time information would lead to higher requirements on the communication technique in booth speed and throughput, which LoRa is not close to handle.

### 6.1.4 LoRa

Using LoRa as communication technique added a lot of limitations of how the system could be implemented. The low throughput resulted in a very small message with limited information about the current charging state of the EVSE. With a higher throughput communication technique information from the energy meter could be added to the control algorithm for a more controlled charging of the EVs. This would also require a lot more messages between the control system and the EVSEs. For such a implementation a higher throughput technique is needed.

Each state change of the EVSEs is critical for when to change charging currents and therefore need acknowledgement from the control system. Serving a big parking lot with many EVSE nodes will add a lot of acknowledgements that needs to be sent from the control system. This would increase the numbers of gateways needed as each gateway probably will reach the limit of how many messages is allowed to be sent. Adding more gateways in the same area could then increase the risk of messages interfering with each other as the two gateways will send messages.

These two problems interfere much with the rest of the charging system as this is the most critical communication path. By chaining communication technique a faster and more expandable system could have been created.

This however would require a rework of the project requirements for adding more high throughput techniques. By chaining the self-hostable and cost requirement for this project a cellular technique would be interesting, or by lowering the range requirement a meshed Wi-Fi solution could be interesting.

There is also a lot of long range techniques that could work better than LoRa however not yet released as of the date when starting this project.

## 6.2 Conclusion

The goal of this project was to design a prototype that could manage the charging of groups of EVs. We achieved this by controlling each EVSE individually. To communicate with the EVSEs we had to use a communication technology with a very long range, a new type of wireless communication technology called LPWAN fit this task. To choose which LPWAN to use we evaluated and compared all relevant technologies. We decided to use LoRaWAN because of its availability of hardware and the possibility of setting up our own network. With the EVSEs connected to the internet the next task was to organise the charging. We created a load balancer application with the task of calculating the current allocations for all connected and charging EVs. To allow users to see their energy consumption and request quick charge we created a web based user interface.

The early decision to build the EVSE based on a already developed third party EPC to lower the complexity of the problem created more problems than it solved. The prototype would probably have achieved a better result if the EV charger control hardware were to be built from scratch and where all communication signals would be controlled directly by the prototype.

The choice of using LoRaWAN as our communication technique limited the prototype a lot in terms of responsiveness and functionality. LoRa was however the only available technique fitting our requirements at the start of this project.

## 6.3 Future improvements

There are some parts of this system that could benefit from future development. Some of the most interesting improvement would be better protection against voltage spikes that occur when using relays, add more functionality and data representation in the user interface, add functionality for handling acknowledgement of the LoRaWAN messages or try another communication technology more suitable for usage with critical data messages.

# Bibliography


[1] EV statistics in sweden. 2017, Jan 31. [Online]. Available: http://elbilsstatistik.se/startsida/se-statistik/

[2] K. Clement-Nyns, E. Haesen, and J. Driesen, "The Impact of Charging Plug-In Hybrid Electric Vehicles on a Residential Distribution Grid," *IEEE Transactions on Power Systems*, vol. 25, no. 1, pp. 371–380, Feb 2010.

[3] J. de Hoog, T. Alpcan, M. Brazil, D. A. Thomas, and I. Mareels, "Optimal Charging of Electric Vehicles Taking Distribution Network Constraints Into Account," *IEEE Transactions on Power Systems*, vol. 30, no. 1, pp. 365–375, Jan 2015.

[4] J. de Hoog, D. A. Thomas, V. Muenzel, D. C. Jayasuriya, T. Alpcan, M. Brazil, and I. Mareels, "Electric vehicle charging and grid constraints: Comparing distributed and centralized approaches," in *2013 IEEE Power Energy Society General Meeting*, July 2013, pp. 1–5.

[5] E. Sortomme, M. M. Hindi, S. D. J. MacPherson, and S. S. Venkata, "Coordinated charging of plug-in hybrid electric vehicles to minimize distribution system losses," *IEEE Transactions on Smart Grid*, vol. 2, no. 1, pp. 198–205, March 2011.

[6] R. Sanchez-Iborra and M.-D. Cano, "State of the art in LP-WAN solutions for industrial IoT services," *Sensors*, vol. 16, no. 5, 2016. [Online]. Available: http://www.mdpi.com/1424-8220/16/5/708

[7] B. Sun, Z. Su, D. Wei, and Y. Li, "Research on optimal control of electric vehicle charging in residential area," in *2016 35th Chinese Control Conference (CCC)*, July 2016, pp. 8617–8621.

[8] U. Lindqvist and P. G. Neumann, "The future of the internet of things," *Commun. ACM*, vol. 60, no. 2, pp. 26–30, Jan. 2017.

[9] National Fire Protection Association®, *NFPA 70: National Electrical Code 2017®*, ser. International Electrical Code® Series. Delmar Cengage Learning, 2016.

[10] *SAE Electric Vehicle and Plug in Hybrid Electric Vehicle Conductive Charge Coupler*, SAE International, 2 2016, rev. 6.

[11] *Plugs, socket-outlets, vehicle connectors and vehicle inlets - Conductive charging of electric vehicles - Part 2: Dimensional compatibility and interchangeability requirements for a.c. pin and contact-tube accessories*, International Electrotechnical Commission, 2 2016, rev. 2.

[12] *Plugs, socket-outlets, vehicle connectors and vehicle inlets - Conductive charging of electric vehicles - Part 3: Dimensional compatibility and interchangeability requirements for d.c. and a.c./d.c. pin and contact-tube vehicle couplers*, International Electrotechnical Commission, 3 2014, rev. 1.

[13] EV Plug Alliance, "To connect electrical vehicles." [Online]. Available: http://www.smartgrids-cre.fr/media/documents/dossiers/vehicule/EVPlugAlliance_Plug.pdf

[14] *EVSE Protocol Controller (EPC) Manual*, Viridian EV, 8 2016, rev. 1.2. [Online]. Available: https://static1.squarespace.com/static/56274e10e4b05fed125a4df9/t/57a49565d482e9189b0f0606/1470403966850/EVSE_Protocol_Controller_Manual-+iss1.2+050816.pdf

[15] K. R. Fowler, C. L. Silver, S. (e-book collection), and B. (e-book collection), *Developing and managing embedded systems and products: methods, techniques, tools, processes, and teamwork.* Amsterdam;Boston;: Elsevier, 2015;2014;.

[16] *1-/2-/4-Channel Digital Potentiometers*, Analog Devices, Inc, 2010, rev. E. [Online]. Available: http://www.analog.com/media/en/technical-documentation/data-sheets/AD8400_8402_8403.pdf

[17] *Eastron SDM230 Modbus Protocol*, Jiaxing Eastron Electronic Instruments Co.,Ltd, v1.2. [Online]. Available: http://www.flanesi.it/blog/download/sdm230modbus/SDM230-Modbus%20protocol%20V1.2.pdf

[18] "IEEE standard for floating-point arithmetic," *IEEE Std 754-2008*, pp. 1–70, Aug 2008.

[19] *MODBUS Application Protocol Specification*, Modbus Organization, Inc, 4 2012, v1.1b3. [Online]. Available: http://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf

[20] Telecommunications Industry Association (TIA), "Standard TIA-485: Electrical characteristics of generators and receivers for use in balanced digital multipoint systems," December 2012.

[21] *+3.3V Low Power Half-Duplex RS-485 Transceiver with 10Mbps Data Rate*, Exar Corporation, 6 2012, rev 1. [Online]. Available: https://www.exar.com/content/document.ashx?id=639

[22] ETSI, "En 300 220-2 v3.1.1," 2016. [Online]. Available: http://www.etsi.org/deliver/etsi_en/300200_300299/30022002/03.01.01_30/en_30022002v030101v.pdf

[23] N. Sornin (Semtech), M. Luis (Semtech), T. Eirich (IBM), T. Kramp (IBM), O.Hersent (Actility), "Lorawan specification v1.0.2," 2016. [Online]. Available: https://www.lora-alliance.org/Contact/Request-Specification-Form

[24] L. A. T. committee, "Lorawan regional parameters v1.0," 2016. [Online]. Available: https://www.lora-alliance.org/Contact/Request-Specification-Form

[25] L. Labs. Symphony Link - Internet of Things Wireless LPWA. (2017, May 4). [Online]. Available: https://www.link-labs.com/symphony

[26] Sigfox. Radio Technology Keypoints | Sigfox. (2017, May 4). [Online]. Available: https://www.sigfox.com/en/sigfox-iot-radio-technology

[27] WAVIoT™. NB-Fi FAQ. (2017, Mar 20). [Online]. Available: http://dgmatics.com/technology/faq

[28] ——, "White paper: Waviot nb-fi lpwan technology," Tech. Rep., June 2016.

[29] *DB2196: STM32 Nucleo-64 board*, STMicroelectronics, 11 2016, rev. 8.0.

[30] *SX1272MB2DAS Hardware Description*, Semtech, 5 2016, rev. 1.0.

# A

# Extension PCB

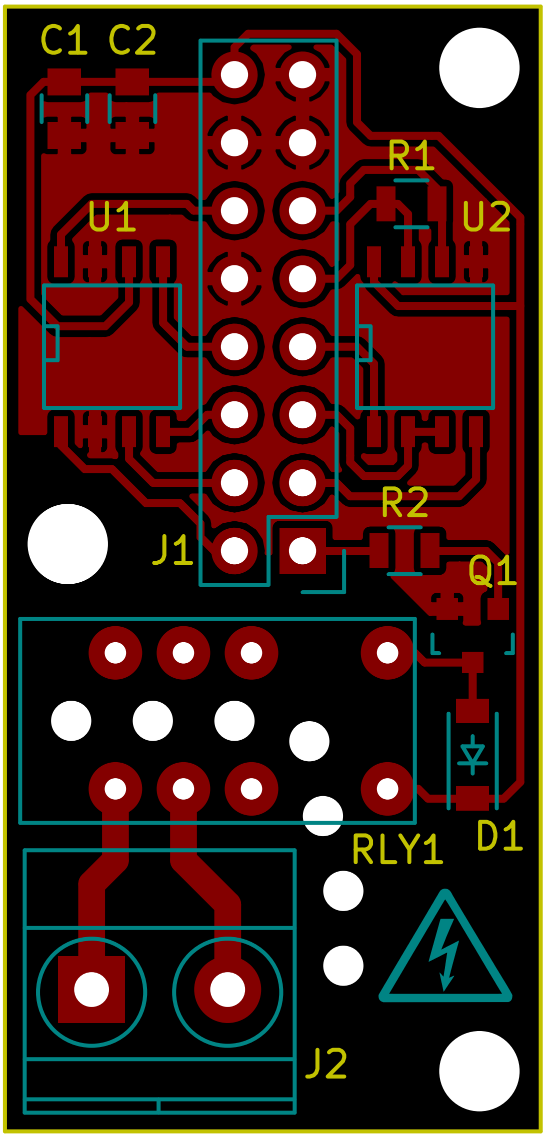


**Figure A.1:** PCB layout.

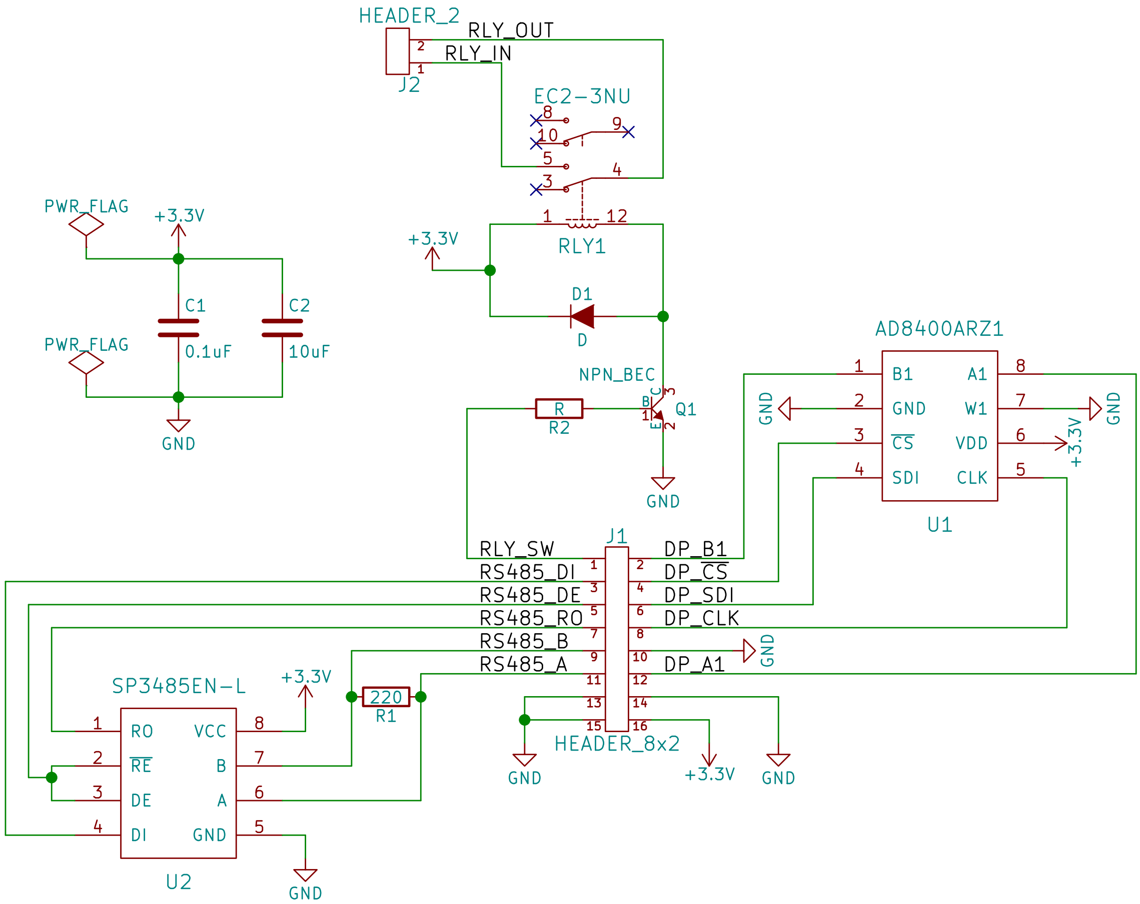


**Figure A.2:** PCB schematic.

# B
# Range Data

LoRa Range test data

| | | | DR_5 (SF7) | | DR_4 (SF8) | | DR_3 (SF9) | | DR_2 (SF10) | | DR_1 (SF11) | | DR_0 (SF12) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Location** | **Plats** | **Distance** | **RSSI** | **LoRaSNR** | **RSSI** | **LoRaSNR** | **RSSI** | **LoRaSNR** | **RSSI** | **LoRaSNR** | **RSSI** | **LoRaSNR** | **RSSI** | **LoRaSNR** | **Node-Antenna** | **Gateway-Antenna** |
| 57.67118,12.01549 | 1 | 1 | -25 | 7.5 | -33 | 8.8 | -53 | 12.5 | -51 | 10.8 | -40 | 10.2 | -27 | 8 | Short | Standard |
| 57.67099,12.01551 | 2 | 21 | -76 | 8.5 | -81 | 11.2 | -75 | 8.5 | -69 | 6.8 | -70 | 12.5 | -62 | 8.5 | Short | Standard |
| 57.67057,12.01584 | 3 | 71 | -80 | 8.8 | -79 | 10.5 | -83 | 7.2 | -86 | 7.5 | -76 | 8.5 | -75 | 7.2 | Short | Standard |
| 57.66977,12.01588 | 4 | 160 | -91 | 8.5 | -91 | 10 | -92 | 10.2 | -85 | 9 | -96 | 7 | -88 | 7.2 | Short | Standard |
| 57.66884,12.0165 | 5 | 270 | -107 | -2.2 | -104 | 0.2 | -100 | 7 | -106 | 0.2 | -100 | 6.5 | -101 | 2.5 | Short | Standard |
| 57.66736,12.01693 | 6 | 430 | -110 | -4 | -107 | -1.2 | -108 | -3 | -107 | -2.8 | -111 | -7.8 | -106 | -2.8 | Short | Standard |
| 57.66502,12.01746 | 7 | 700 | -111 | -5.2 | -109 | -6 | -109 | -6.8 | -111 | -7.5 | -108 | -3.5 | -110 | -5.5 | Short | Standard |
| 57.66502,12.01746 | 7 | 700 | -106 | 0 | -106 | -1 | -107 | -0.2 | -106 | -0.2 | -109 | -1.5 | -105 | 0 | Bump | Standard |
| 57.66502,12.01746 | 7 | 700 | -104 | 2.2 | -105 | 2 | -105 | 3.8 | -109 | -2.2 | -108 | -1.8 | -105 | -0.5 | Short | Standard |
| 57.66495,12.01761 | 7 | 700 | x | | -113 | -5.2 | -111 | -13.5 | -111 | -13 | -111 | -15.8 | -110 | -10.2 | Short | Standard |
| 57.66495,12.01761 | 7 | 700 | -112 | -6.8 | | | -112 | -12.8 | -110 | -9.2 | -112 | -12.5 | -111 | -9.8 | Bump | Standard |
| 57.66495,12.01761 | 7 | 700 | -106 | 6.8 | -107 | 6.5 | -106 | 8.8 | -105 | 5.8 | -104 | 7 | -105 | 6.5 | Bump | Outside big |
| 57.66495,12.01761 | 7 | 700 | -104 | 7 | -109 | 8.5 | -109 | 8 | -103 | 6.8 | -107 | 7 | -106 | 5.8 | Short | Outside big |
| 57.66392,12.0182 | 8 | 820 | -109 | -7 | -108 | -4.8 | -110 | -9.2 | -111 | -9.2 | -109 | -7.5 | -109 | -5.2 | Bump | Standard |
| 57.66392,12.0182 | 8 | 820 | -103 | 5 | -107 | 2.5 | -106 | 1.2 | -104 | 3.2 | -107 | 4.5 | -105 | -1.8 | Short | Standard |
| 57.66169,12.01902 | 9 | 1080 | x | | x | | x | | -115 | -11.2 | -113 | -11 | -113 | -11.2 | Short | Standard |
| 57.66169,12.01902 | 9 | 1080 | x | | x | | x | | x | | -112 | -17.8 | x | | Bump | Standard |
| 57.66026,12.0201 | 10 | 1250 | x | | x | | x | | | | x | | -115 | -17 | Short | Standard |
| 57.66026,12.0201 | 10 | 1250 | x | | x | | x | | x | | x | | x | | Short | Outside big |
| 57.659,12.02089 | 11 | 1390 | -117 | -1 | -115 | 0.8 | -118 | 1.2 | -119 | -7 | -119 | -4.5 | -116 | 0.2 | Short | Outside big |
| 57.659,12.02089 | 11 | 1390 | x | | x | | x | | x | | x | | x | | Short | Standard |
| 57.65767,12.01984 | 12 | 1530 | -113 | 5 | -115 | 2 | -115 | 2.2 | -118 | -0.5 | -119 | -3.8 | -119 | -9 | Short | Outside big |
| 57.65602,12.019 | 13 | 1700 | -113 | 4.5 | -102 | 5 | -115 | 5.2 | -109 | 5.2 | -111 | 7 | -110 | 3.8 | Short | Outside big |
| 57.6534,12.02059 | 14 | 2000 | -117 | -3.5 | -119 | -5 | -123 | -8.5 | -118 | -2 | -117 | -0.2 | -121 | -6.5 | Short | Outside big |
| 57.65049,12.02204 | 15 | 2340 | -115 | 1 | -118 | -5.5 | -121 | -11 | -121 | -7.2 | -121 | -8.2 | -119 | -5 | Short | Outside big |
| 57.64679,12.02505 | 16 | 2780 | -118 | -4.8 | x | | -119 | -11.2 | -121 | -11.8 | -121 | -12.5 | -119 | -11 | Short | Outside big |
| 57.64512,12.02744 | 17 | 2990 | -118 | -5.2 | -119 | -5.5 | -121 | -4 | -104 | -10 | -119 | -6.8 | -119 | -7.2 | Short | Outside big |
| 57.64099,12.03226 | 18 | 3510 | -121 | -7.8 | -119 | -7.5 | x | | -119 | -10.5 | -119 | -6.2 | -116 | -7.5 | Short | Outside big |
| 57.63199,12.03607 | 19 | 4530 | x | | -121 | -8.8 | -119 | -13 | -121 | -12.2 | -123 | -8.8 | -121 | -6.8 | Short | Outside big |
| 57.62547,12.04096 | 20 | 5310 | x | | -121 | -8.5 | x | | x | | -119 | -15.2 | x | | Short | Outside big |
| 57.62447,12.04162 | 21 | 5430 | x | | x | | x | | x | | x | | x | | Short | Outside big |