Bachelor's Thesis

Czech
Technical
University
in Prague

F3

Faculty of Electrical Engineering
Department of Measurements

LoRaWAN based monitoring system

Ondřej Fidra

Supervisor: doc. Ing. Radislav Šmíd, Ph.D.
May 2018

CTU
CZECH TECHNICAL
UNIVERSITY
IN PRAGUE

BACHELOR‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name:	**Fidra Ondřej**	Personal ID number:	**456869**
Faculty / Institute:	**Faculty of Electrical Engineering**		
Department / Institute:	**Department of Measurement**		
Study program:	**Cybernetics and Robotics**		
Branch of study:	**Sensors and Instrumentation**		

II. Bachelor’s thesis details

Bachelor’s thesis title in English:

LoRaWAN Based Monitoring System

Bachelor’s thesis title in Czech:

Monitorovací systém s LoRaWAN komunikací

Guidelines:

Develop a system for monitoring physical quantities with LoRaWAN IoT communication. Design and create two modules:
1. Battery-powered temperature (or other variables) sensing device with minimized power consumption
2. Edge device for monitoring of supply voltage and current consumption for common appliances with local evaluation of supplied and consumed energy. Implement a backup power supply and output switching using the relay.

Use microcontrollers from STmicroelectronics, realize the devices on PCBs. Create a gateway using the Raspberry Pi module. Examine the possibility of systém modification from WAN to point-to-point LoRa communication.

Bibliography / sources:

[1] Pham, C.: Building low-cost gateways and devices for open LoRa IoT test-beds. 11th EAI Intl. Conference TridentCom, Hangzhou, China, June 13-15, 2016.
[2] Ganssle, J.: The Art of Designing Embedded Systems, Elsevier Science, 2008.
[3] HopeRF datasheets, http://www.hoperf.com/rf_transceiver/lora/
[4] Raspberry Pi documentation, https://www.raspberrypi.org/
[5] STMicroelectronics datasheets, http://www.st.com

Name and workplace of bachelor’s thesis supervisor:

doc. Ing. Radislav Šmíd, Ph.D., Department of Measurement, FEL

Name and workplace of second bachelor’s thesis supervisor or consultant:

Date of bachelor’s thesis assignment: **10.01.2018** Deadline for bachelor thesis submission: __________

Assignment valid until:
by the end of summer semester 2018/2019

doc. Ing. Radislav Šmíd, Ph.D. Supervisor’s signature	Head of department’s signature	prof. Ing. Pavel Ripka, CSc. Dean’s signature

Acknowledgements

I would like to thank the supervisor doc. Ing. Radislav Šmíd, Ph.D. for his continous support during this thesis. Furthermore, I would like to thank my parents for the constant positive mood and psychic support during the whole study.

Declaration

I hereby declare that I worked out the presented thesis independently and I quoted all the sources used in this thesis in accord with Methodical instructions about ethical principles for writing academic thesis.

In Prague, 24th May 2018

Prohlašuji, že jsem předloženou práci vypracoval samostatně, a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze, 24. května 2018

Abstract

This thesis aims at development and production of monitoring modules of physical quantities using a microcontroller. It also points to the communication chain with LoRa technology. The microcontroller will process measured values and send via wireless LoRa communication using LoRa gateway to appropriate storage. The work aims in more detail on the design of two end-devices (Node). The task of the first one is to measure temperature and humidity in normal conditions. The second one is used to monitor the voltage and current drawn from the electric power distribution for conventional electronic devices.

Keywords: LoRa, microcontroller, low-power devices, Edge computing, Raspberry Pi, Node-RED, The Things Network, Internet of Things

Supervisor: doc. Ing. Radislav Šmíd, Ph.D.

Abstrakt

Tato bakalářská práce se zaměřuje na návrh a výrobu modulů pro monitorování fyzikálních veličin pomocí mikrokontroléru a na komunikační řetězec s technologií LoRa. Mikrokontrolér bude naměřené veličiny zpracovávat a poté prostřednictvím bezdrátové komunikace LoRa odesílat za využití LoRa gatewaye na vhodné uložiště. Práce podrobněji rozebírá vývoj a výrobu dvou koncových zařízení (Node). Úkolem prvního zařízení je monitoring teploty a vlhkosti v běžných podmínkách. Druhé zařízení slouží pro sledování odebíraného proudu a napětí distribuční elektrické sítě pro běžná síťová zařízení.

Klíčová slova: LoRa, mikrokontrolér, nízkopříkonová zařízení, Edge computing, Raspberry Pi, Node-RED, The Things Network, internet věcí

Překlad názvu: Monitorovací systém na bázi LoRaWAN

Contents

Figures

Chapter 1

Introduction

The world of the IoT (internet of things) is full of sensors which sense any physical quantities. The phenomenon IoT is a label for the network used by electronics devices which implement hardware and software to measure, monitor and send data. It allows controlling or monitoring any objects remotely, from any part of the world through the existing network. As this phenomenon grows, the number of sensors increases, and the network gets complex.

These sensors can communicate with the central unit through many different technologies, which every single technology has its advantages and disadvantages. For example, data transfer based on physical connection via wires does not care about power consumption. On the other hand, the devices are physically connected to each other, so there are limited possibilities of mobility, and the installation is complicated.

Take a look at wireless communication. Wireless sensors have practically unlimited movement; the distance between communicating devices limits it. However, its main disadvantage is that the sensors must have independent power supply. The particular case of using wireless communication is when we have an available power supply, but there is no possibility to attach wired communication physically between the sensor and other devices. The problem of wireless devices with independent power supply means that the devices have limited lifetime due the lifetime of a battery. Nowadays, sensors get smaller, so we cannot extend its lifetime by increasing dimensions of batteries. To extend the lifetime of wireless sensors to maximum, we reduce power consumption to the minimum by using low power technologies such as deep sleep modes and using low power data transfer technologies.

There are many different types of communications providing low power wireless data transfer, for example, Bluetooth low energy (also known as Bluetooth Smart or BLE), ZigBee, ANT, SigFox, LoRa. Some of these communications belong to network family LPWAN (Low Power Wide Area Network). LPWAN is designed for battery-powered devices with low data rate and with a requirement of long range. To achieve these specifications, LPWAN uses many different platforms. Thw most used platforms are based on ultra narrowband modulation (SigFox) or chirp spread spectrum (LoRa).

Chapter 2

Aims

This thesis describes an development and making of two different end-nodes with LoRa connectivity. The first device is a battery powered humidity and temperature sensor with optimised power consumption (the Thermometer). The second one is the Plug Manager which is used for monitoring the proper function of an ordinary AC powered appliance (such as refrigerator, for example). The device measures voltage and current from which computes properties of mains electricity, current and power drawn by an appliance and it also does the spectral analysis, fast Fourier transformation (FFT), of current. The Plug Manager has a relay to switch the appliance on or off if it is not properly working. Both devices will use LoRa network and local area network (LAN) to make the data available to the user. The web page will be used to represent data visually.

2.1 The Thermometer

The main task of the Thermometer end-node will be to measure ambient temperature and humidity and to achieve low power consumption. The power supply of the device is 3V button cell CR2032. To achieve low power consumption, there will be used microcontroller (MCU) and sensor with low power modes and LoRa modem to transmit measured values. The values can be visually displayed on every device with access to LAN network and with web browser. The solution will be developed on the solderless breadboard with development board by ST Microelectronics, STM32L-DISCOVERY. For the final realisation, the custom printed circuit board (PCB) will be designed, and the PCB will be made.

2.2 The Plug Manager

The Plug Manager will be the device, which monitors the properties of the mains electricity and power consumption of a typical AC powered appliance. It will measure instantaneous voltage and current (512 samples per period) and computes the root-mean-squared (RMS), maximum values, a period of the voltage, the phase of voltage relative to current and active, reactive and apparent power. And it will do the FFT for the spectral analysis. It will send the values via LoRa every error or event (overcurrent, loss of the electric power) and also every predefined time slot. For the development purposes, the solderless breadboard will be used with the development board as mentioned earlier. To control the device, there will be organic light-emitting diode (OLED) display and three capacitive sensing buttons. The standard push buttons could not be used, because the device will not be galvanic isolated from the mains. In the final realisation, there will be 230V AC, this value is dangerous, so for the development, the 12V AC will be used. As the Thermometer, the Plug Manager will also be realised on the PCB. For the safety, the whole device will be closed in the power strip with one socket.

Chapter 3

LPWAN

LPWAN (low-power wide-area network) is family of a wireless telecommunication network which has common features. It is designed to allow long-range communications at a low bit rate and with low power consumption. Primarily it is used in battery-powered sensors. Usually, there are gateway concentrators to receive the LPWAN and sending to the internet, but it can also use private wireless sensor network.

3.1 LoRa Introduction

The term LoRa stands for Long-Range, and it represents patented wireless communication. The range of this standard is circa 10 kilometres, and it can receive a signal under the noise. The LoRa technology is divided into two parts - LoRa, the physical layer and LoRaWAN, the upper layers.

3.2 LoRa Architecture

The LoRa network consists of an application server, a network server, a join server, gateway radio and end-device. It has a star topology. The diagram of the LoRa network architecture can be seen in the Fig. 3.1.

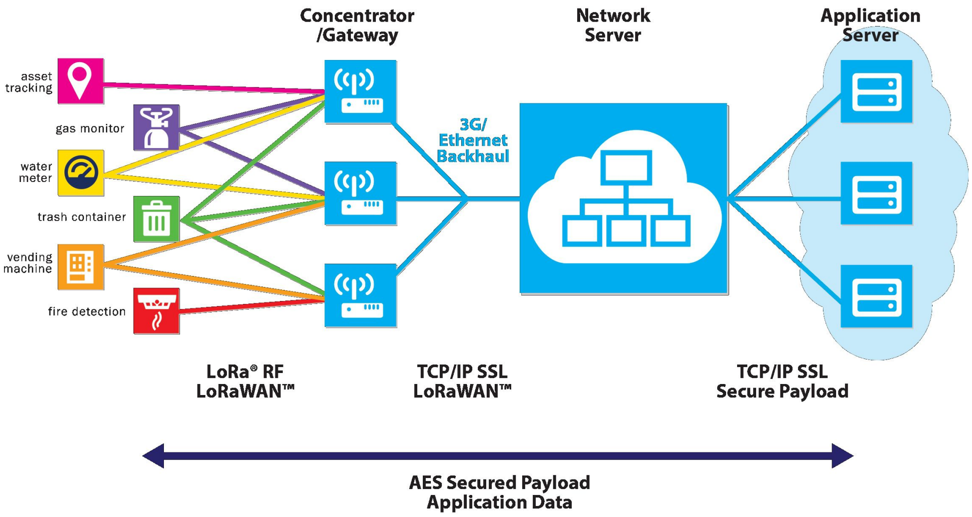


Figure 3.1: Architecture of LoRa network

3.2.1 Application Server

The Application Server handles all the application layer payloads of the associated End-Devices and provides the application-level service to the end-user. It also generates all the application layer downlink payloads towards the connected End-Devices.[9]

3.2.2 Network Server

The Network Server terminates the LoRaWAN MAC layer for the End-Devices connected to the network. It is the centre of the star topology.[9]

3.2.3 Join Server

The Join Server manages the Over-the-Air (OTA) End-Device activation process.[9]

3.2.4 Radio Gateway

The Radio Gateway forwards all received LoRaWAN radio packets to the Network Server that is connected through an IP backbone. The Radio Gateway operates entirely at the physical layer. Its role is to decode uplink radio packets from the air and forward them unprocessed to the Network Server. Conversely, for downlinks, the Radio Gateway executes transmission requests coming from the Network Server without any interpretation of the payload.[9]

3.2.5 End-Device

The End-Device is a sensor or an actuator. The End-Device is wirelessly connected to a LoRaWAN network through Radio Gateways. The application layer of the End-Device is connected to a specific Application Server in the cloud.[9]

3.3 LoRa Physical Layer

The physical layer is closed, and there is no freely available official document of the LoRa physical layer specification. The LoRa sends messages on the license-free radio frequency bands (169 MHz, 433 MHz, 868 MHz for Europe and 915 MHz for North America). The LoRa protocol is based on chirp spread spectrum (CSS) modulation which uses frequency chirps with a linear variation of frequency over time in order to encode information. Because of the linearity of the chirp pulses, frequency offsets between the receiver and the transmitter are equivalent to timing offsets, easily eliminated in the decoder. This also makes this modulation immune to the Doppler effect, equivalent to a frequency offset.[1] This type of modulation also provides excellent distortion immunity and can receive a signal that has down to -20dB signal-to-noise ratio (SNR).

The LoRa modulation can adjust parameters of transmitting signal. They are:

- bandwidth (BW)

- spreading factor (chirp rate) (SF)

Bandwidth is a width of the frequency spectrum which uses communication, spreading factor (chirp rate) is the first derivative of chirp frequency it determines the number of bits which are encoded in one chirp.[6] In the Fig.3.2 (adapted from [6]) is shown how the frequency spectrum of LoRa modulation looks like and meaning of the parameters.

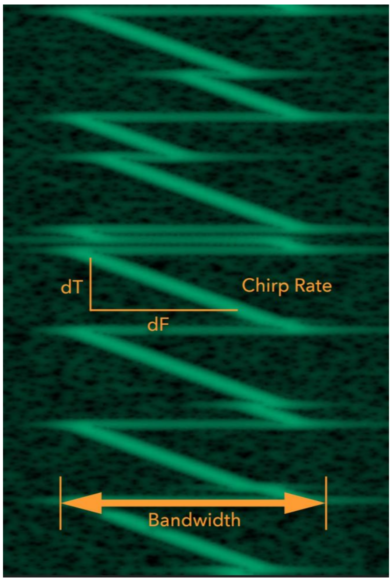


Figure 3.2: Frequency spectrum of LoRa modulation

The possibility of changing parameters allows using adaptive data rate (ADR) feature. ADR saves the air-time, reduces power consumption, but it is suitable only for stationary devices.

The decoding of this signal is done by inverting the direction of the chirp signals. Then these signals are multiplied. The Fig.3.3 (adapted from [6]) shows how the result looks. After that, symbols can be read.

3.4 LoRaWAN

LoRaWAN standard defines medium access control (MAC) sublayer and specifications of devices with LoRa communication. The message formatting is shown in the Fig.3.4, Fig.3.5, Fig.3.6, Fig.3.7 (adapted from [8]. All LoRa uplink and downlink messages carry a PHY payload (Payload) starting

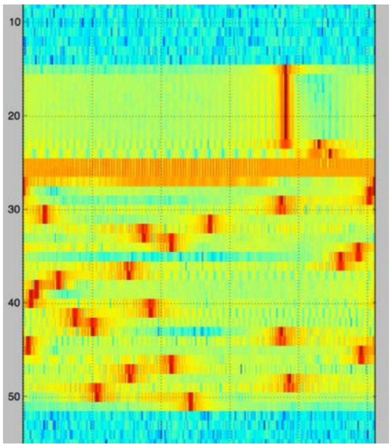


Figure 3.3: Decoded signal

with a single-octet MAC header (MHDR), followed by a MAC payload (MACPayload), and ending with a 4-octet message integrity code (MIC)[8].

In MAC Payload there is structure named FPort. FPort is one-byte long register. An application can use values of FPort from an interval from 1 to 223. The value 224 is used for testing purposes. If the FPort has a value of 0 than the FRMPayload contains only MAC commands. LoRaWAN implementation SHALL discard any transmission request from the application layer where the FPort value is not in the 1..224 range.[8] In frame header structure (FHDR) is 2 byte frame counter (FCnt). The FCnt must never be reset during the device's lifetime.

The LoRaWAN uses double encryption to the data. Both encryptions are AES128 (advanced encryption standard). FRMPayload must be encrypted before the message integrity code (MIC) is calculated. MIC is calculated over all the fields in the message using AES128 CMAC encryption.

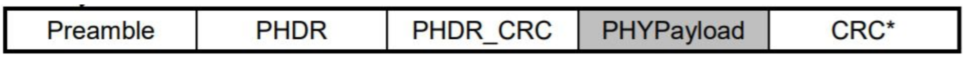


Figure 3.4: Radio PHY structure (* is only available on uplink messages)

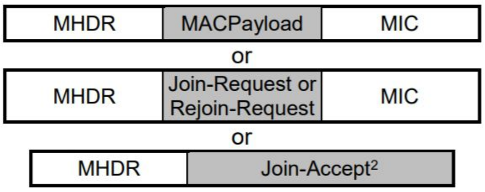


Figure 3.5: PHY payload structure

Figure 3.6: MAC payload structure

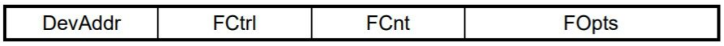


Figure 3.7: Frame header structure

3.4.1 Activation of devices

Before the device can be used, it must be activated. LoRaWAN defines two types of activation, Over-The-Air (OTA) activation and Activation-by-Personalization (ABP).

ABP

Device with Activation-by-Personalization has statically defined its addresses and keys for the encryption (Application Session Key, Network Session Key). These data are programmed by a manufacturer.

OTA

The OTA device has assigned only some of the addresses (Device EUI, Join EUI). It has no device ID and no encryption keys from a manufacturer. At this point, the device is called Generic End-Device[9]. After the commissioning, the end-device gets the rest of the addresses and IDs and also encryption keys.

3.4.2 LoRaWAN end-device classes

LoRaWAN defines three classes of devices (class A, class B, class C). The differences of these classes is in operation which makes LoRaWAN more flexible.

Class A

The default class which must be supported by all LoRaWAN end-devices, class A communication is always initiated by the end-device and is fully asynchronous. Each uplink transmission can be sent at any time and is followed by two short downlink windows, giving the opportunity for bi-directional communication, or network control commands if needed. This is an ALOHA type of protocol. Because downlink communication must always follow an uplink transmission with a schedule defined by the end-device application, downlink communication must be buffered at the network server until the next uplink event. The time flow is shown in the Fig.3.8 (adapted from [8]).[7]

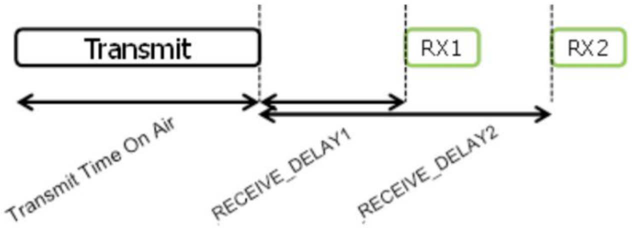


Figure 3.8: End-device receive slot timing

Class B

In addition to the class A initiated receive windows, class B devices are synchronised to the network using periodic beacons, and open downlink 'ping slots' at scheduled times. This provides the network the ability to send downlink communications with a deterministic latency, but at the expense of some additional power consumption in the end-device. The time flow of beacon is shown in the Fig.3.9 (adapted from [8]).[7]

Class C

In addition to the class A structure of uplink followed by two downlink windows, class C further reduces latency on the downlink by keeping the receiver of the end-device open at all times that the device is not transmitting (half duplex). Based on this, the network server can initiate a downlink transmission at any time on the assumption that the end-device receiver is

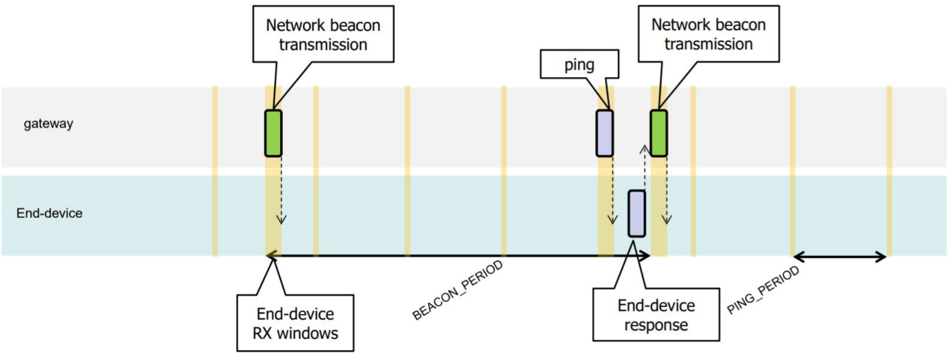


Figure 3.9: Beacon reception slot and ping slots

open, so no latency. The compromise is the power drain of the receiver (up to 50mW) and so class C is suitable for applications where continuous power is available.[7]

3.4.3 LoRa in private network

The LoRa physical layer and LoRaWAN are two separated standards which can be used independently. There is no restriction to use LoRa modulation with LoRaWAN standard. As it was mentioned above, LoRaWAN is an open standard and in End-Devices is made by software. The LoRa transceiver modules receive and transmit data on LoRa physical layer, the layers above are done in controller, and that is why we have freedom in choosing the right architecture (for example point-to-point). The custom network topology can be made. Some limitations must be respected, for example, do not start transmitting while the other device did not end transmitting yet (the collision may occur and loss of both packets). As is known that the LoRa works on license-free radio frequency bands, it can be freely used with all of the benefits of the LoRa modulation.

Chapter 4

HW Design

4.1 Microcontroller

Microcontroller (MCU) contains the processor core, memories and many peripherals in one chip. It is the central part of peripheral devices, it controls and manages all of the data flow, makes all the components to work correctly and does every data computation. Both devices built within this thesis contains the same microcontroller. Because of previous experiences with programming powerful MCUs from ST Microelectronics, it was chosen to use their microcontrollers. During the making devices, there was one limitation on the MCU due to the LoRa library. The LoRa library was compatible only with ST microcontrollers from the low-power family. The microcontroller that is used on both devices is STM32L072.

4.1.1 Overview

The STM32L072 is based on ARM Cortex M0+ core, and it has 192kB of flash memory and 20kB of RAM (random access memory).[16] The internal block diagram of this microcontroller is shown in the Fig.4.1 (adapted from [16]).

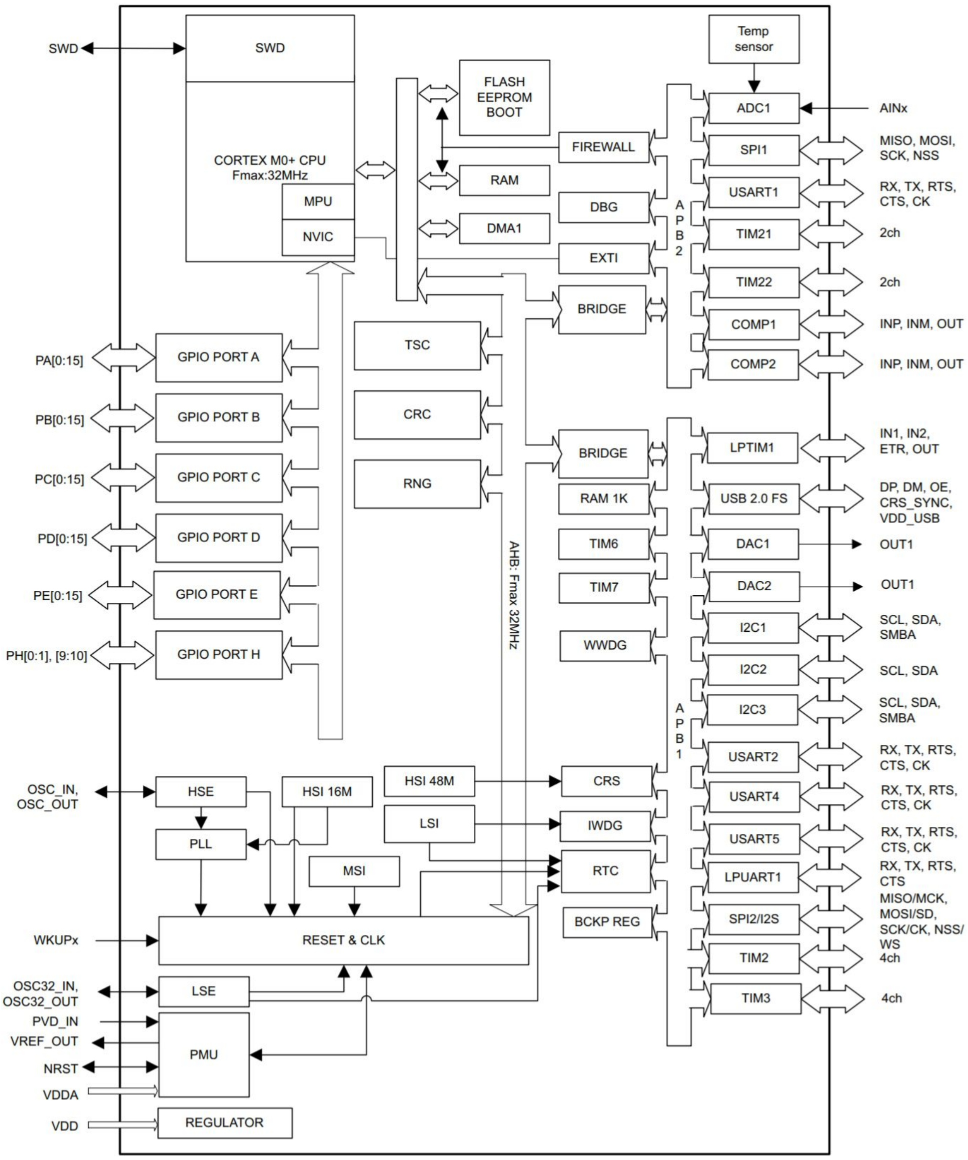


Figure 4.1: STM32L072xx block diagram

4.1.2 Peripherals

The applications of both devices do not use all of the peripherals. There will be mentioned peripherals that have been used.

GPIO

Let's start with GPIO (general purpose input/output) ports[17], which provides the physical connection with all the hardware. In the Fig.4.2 (adapted from [17]) is shown an internal structure of one I/O pin. Each of the GPIO pins can be configured by software as follows:

- Output states: push-pull or open drain + pull-up/down
- Output data from output data register (GPIOx_ODR) or peripheral (alternate function output)
- Speed selection for each I/O
- Input states: floating, pull-up/down, analog
- Input data to input data register (GPIOx_IDR) or peripheral (alternate function input)
- Bit set and reset register (GPIOx_BSRR) for bitwise write access to GPIOx_ODR
- Locking mechanism (GPIOx_LCKR) provided to freeze the I/O port configurations
- Analog function
- Alternate function selection registers
- Fast toggle capable of changing every two clock cycles
- Highly flexible pin multiplexing allows the use of I/O pins as GPIOs or as one of several peripheral functions

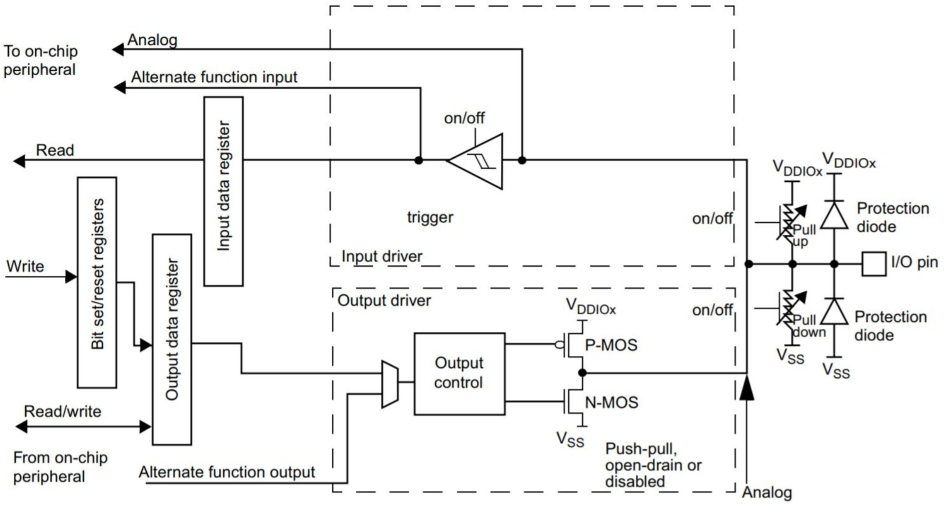


Figure 4.2: Basic structure of an I/O port bit

Timers

The ultra-low-power STM32L072xx devices include three general-purpose timers, one low-power timer (LPTIM), one basic timer, two watchdog timers and the SysTick timer.[16][17] Here will be overviewed general-purpose timers (the most complex). In the picture Fig.4.3 (adapted from [17]) is internal structure of a timer. It's main features are:

- 16-bit up, down, up/down auto-reload counter.
- 16-bit programmable prescaler used to divide the counter clock frequency by any factor between 1 and 65535.
- Up to 4 independent channels for:
  - Input capture
  - Output compare
  - Pulse Width Modulation (PWM) generation (Edge- and Center-aligned modes)
  - One-pulse mode output
- Synchronization circuit to control the timer with external signals and to interconnect several timers.
- Interrupt/Direct Memory Access (DMA) generation on the following events:

  - Update: counter overflow/underflow, counter initialization (by software or internal/external trigger)
  - Trigger event (counter start, stop, initialization or count by internal/external trigger)
  - Input capture
  - Output compare
- Supports incremental (quadrature) encoder and hall-sensor circuitry for positioning purposes
- Trigger input for external clock or cycle-by-cycle current management

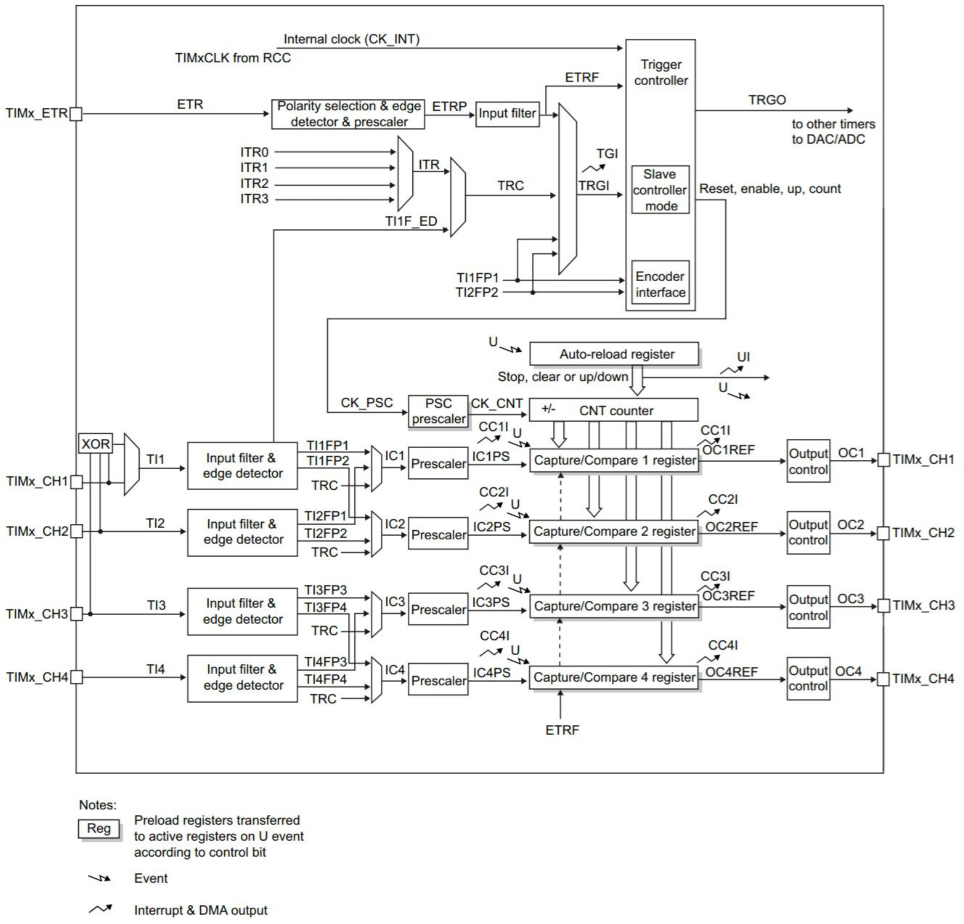


Figure 4.3: General-purpose timer block diagram

SPI

SPI stands for a serial peripheral interface. It is serial synchronous full-duplex communication. In this microcontroller, the SPI peripheral also combines I2S protocol.[16][17] The SPI in the devices is used to communicate with LoRa module RFM95W. Fig.4.4 (adapted from [17]) shows an internal diagram, and there are main features of this peripheral:

- Master or slave operation
- Full-duplex synchronous transfers on three lines
- Half-duplex synchronous transfer on two lines (with bidirectional data line)
- Simplex synchronous transfers on two lines (with unidirectional data line)
- 8-bit to 16-bit transfer frame format selection
- Multimaster mode capability
- 8 master mode baud rate prescalers up to $f_{PCLK}/2$.
- Slave mode frequency up to $f_{PCLK}/2$.
- NSS management by hardware or software for both master and slave: dynamic change of master/slave operations
- Programmable clock polarity and phase
- Programmable data order with MSB-first or LSB-first shifting
- Dedicated transmission and reception flags with interrupt capability
- SPI bus busy status flag
- SPI Motorola support
- Hardware CRC feature for reliable communication:
  - CRC value can be transmitted as last byte in Tx mode
  - Automatic CRC error checking for last received byte
- Master mode fault, overrun flags with interrupt capability
- CRC Error flag
- 1-byte/word transmission and reception buffer with DMA capability: Tx and Rx requests

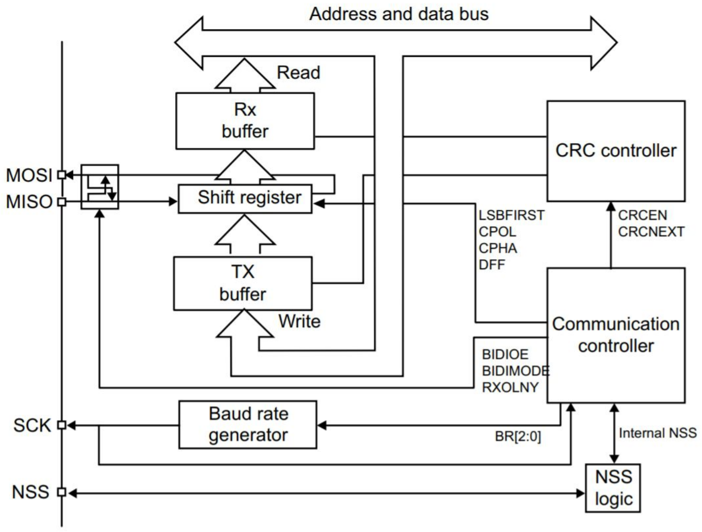


Figure 4.4: SPI block diagram

I2C

The I2C (inter-integrated circuit)[16][17] bus interface handles communications between the microcontroller and the serial I2C bus. It provides multi-master capability and controls all I2C bus-specific sequencing, protocol, arbitration and timing. It supports Standard-mode (Sm), Fast-mode (Fm) and Fast-mode Plus (Fm+). The I2C interface is used in the Power Manager device for the OLED display. The internal diagram is shown in the Fig.4.5 (adapted from [17]). Main features of I2C peripheral:

- I2C bus specification rev03 compatibility:
  - Slave and master modes
  - Multimaster capability
  - Standard-mode (up to 100 kHz)
  - Fast-mode (up to 400 kHz)
  - Fast-mode Plus (up to 1 MHz)
  - 7-bit and 10-bit addressing mode
  - Multiple 7-bit slave addresses (2 addresses, 1 with configurable mask)
  - All 7-bit addresses acknowledge mode
  - General call
  - Programmable setup and hold times

  - Easy to use event management
  - Optional clock stretching
  - Software reset
- 1-byte buffer with DMA capability
- Programmable analog and digital noise filters

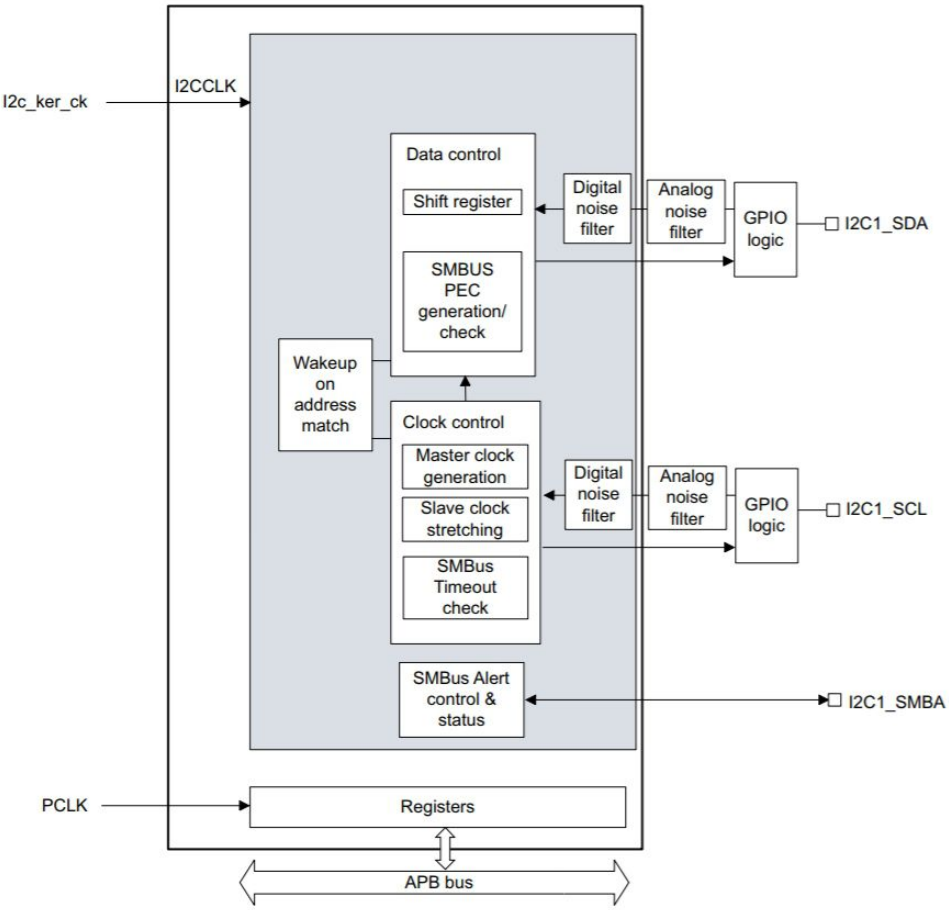


Figure 4.5: I2C block diagram

ADC

The 12-bit ADC is a successive approximation analog-to-digital converter. It has up to 19 multiplexed channels are allowing it to measure signals from 16 external and 2 internal sources.[16][17] A/D conversion of the various channels can be performed in single, continuous, scan or discontinuous mode. The result of the ADC is stored in a left-aligned or right-aligned 16-bit data register. The internal ADC is used in the Power Manager to measure the properties of mains electricity. The internal diagram of ADC is in the Fig.4.6 (adapted from [17]). Main features of ADC:

- High performance
  - 12-bit, 10-bit, 8-bit or 6-bit configurable resolution
  - ADC conversion time: $0.87\mu s$ for 12-bit resolution (1.14 MHz), $0.81\mu s$ conversion time for 10-bit resolution, faster conversion times can be obtained by lowering resolution.
  - Self-calibration
  - Programmable sampling time
  - Data alignment with built-in data coherency
  - DMA support
- Low-power
  - Application can reduce PCLK frequency for low-power operation while still keeping optimum ADC performance. For example, $0.87\mu s$ conversion time is kept, whatever the frequency of PCLK)
  - Wait mode: prevents ADC overrun in applications with low frequency PCLK
  - Auto off mode: ADC is automatically powered off except during the active conversion phase. This dramatically reduces the power consumption of the ADC.
- Analog input channels
  - 16 external analog inputs
  - 1 channel for internal temperature sensor (VSENSE)
  - 1 channel for internal reference voltage (VREFINT)
- Start-of-conversion can be initiated:
  - By software
  - By hardware triggers with configurable polarity (internal timer events or GPIO input events)
- Conversion modes
  - Can convert a single channel or can scan a sequence of channels.
  - Single mode converts selected inputs once per trigger
  - Continuous mode converts selected inputs continuously
  - Discontinuous mode

- Interrupt generation at the end of sampling, end of conversion, end of sequence conversion, and in case of analog watchdog or overrun events
- Analog watchdog
- Oversampler
  - 16-bit data register
  - Oversampling ratio adjustable from 2 to 256x
  - Programmable data shift up to 8-bits
- ADC supply requirements: 1.65 to 3.6 V
- ADC input range: $VSSA \leq VIN \leq VDDA$

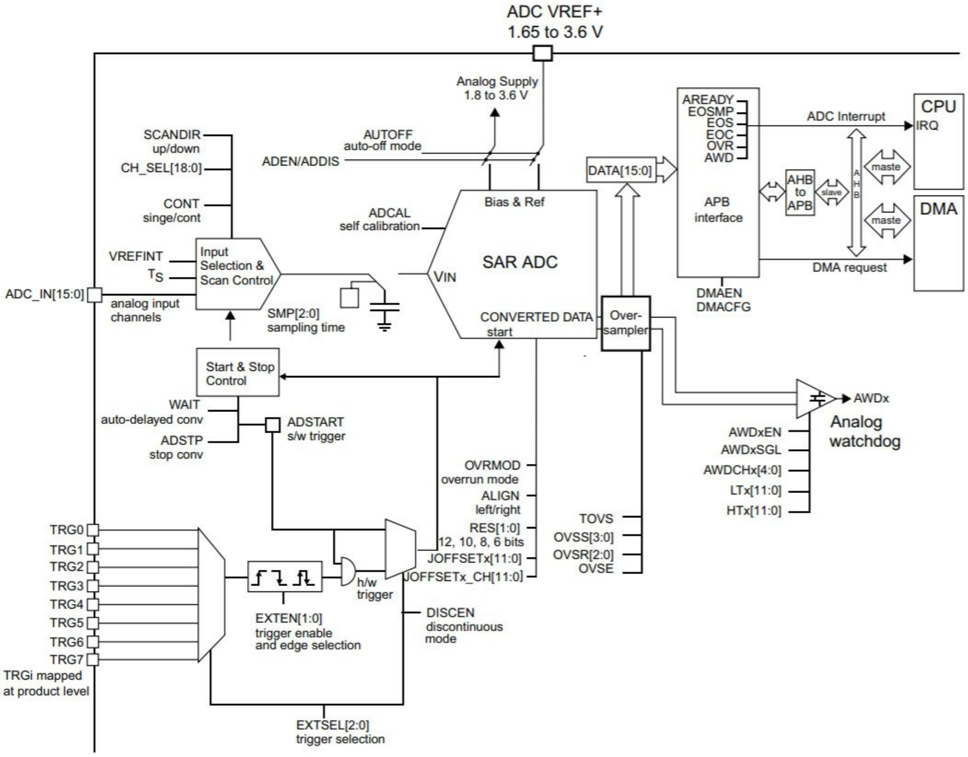


Figure 4.6: ADC block diagram

COMP

STM32L0x2 devices embed two ultra-low-power comparators COMP1, and COMP2[16][17] that can be used either as standalone devices (all terminal are available on I/Os) or combined with the timers. Both comparators are used in the Plug Manager to find passes through zero. The internal diagram of comparators is in the Fig.4.7 (adapted from [17]). Here are main features of COMP:

- COMP1 comparator with ultra low consumption
- COMP2 comparator with rail-to-rail inputs, fast or slow mode
- Each comparator has positive and configurable negative inputs used for flexible voltage selection:
  - I/O pins
  - DAC
  - Internal reference voltage and three submultiple values (1/4, 1/2, 3/4) provided by scaler (buffered voltage divider)
- Programmable speed/consumption (COMP2 only)
- The outputs can be redirected to an I/O or to timer inputs for triggering:
  - Capture events
- COMP1 and COMP2 can be combined in a window comparator. Each comparator has interrupt generation capability with wake-up from Sleep and Stop modes (through the EXTI controller)

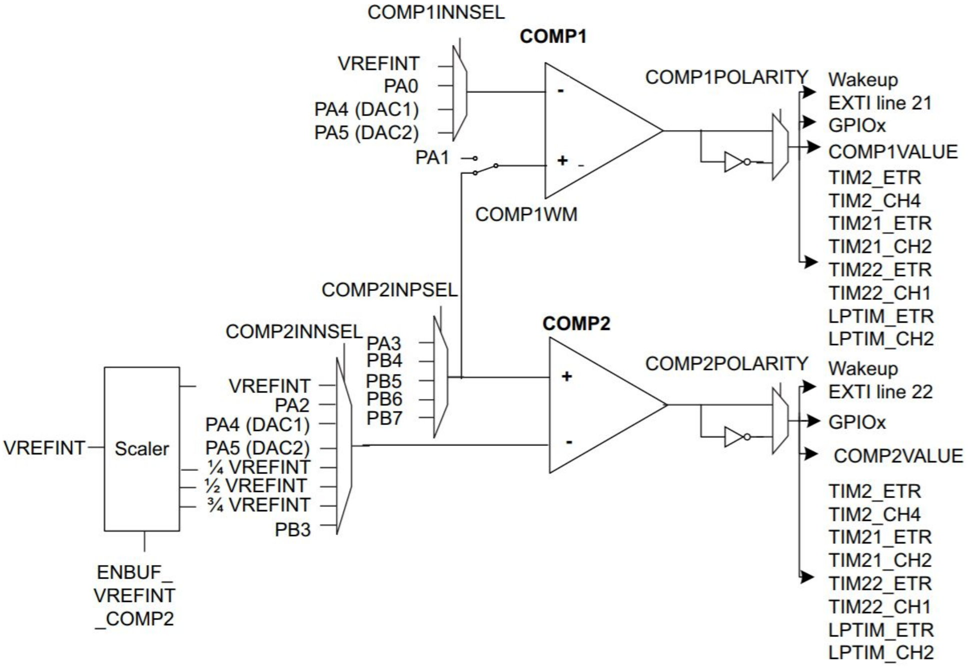


Figure 4.7: COMP block diagram

4.2 LoRa Module

At this time, there are not many available LoRa modules. Almost every module is based on Semtech's chips. The most affordable was a module RFM95W which operates on frequency 868MHz.

The RFM95W transceiver features the LoRa^{TM} long range modem that provides ultra-long range spread spectrum communication and high interference immunity while minimising current consumption. This device also supports high performance (G)FSK modes for systems including WMBus, IEEE802.15.4g. The RFM95W deliver exceptional phase noise, selectivity and receiver linearity for significantly lower current consumption than competing devices. [5] As can be seen in the Fig.4.8 (adapted from [5]) the block diagram of the chip, it is complete transceiver with a digital interface to control it. The pinout of the module is in the Fig.4.9 (adapted from [5]).

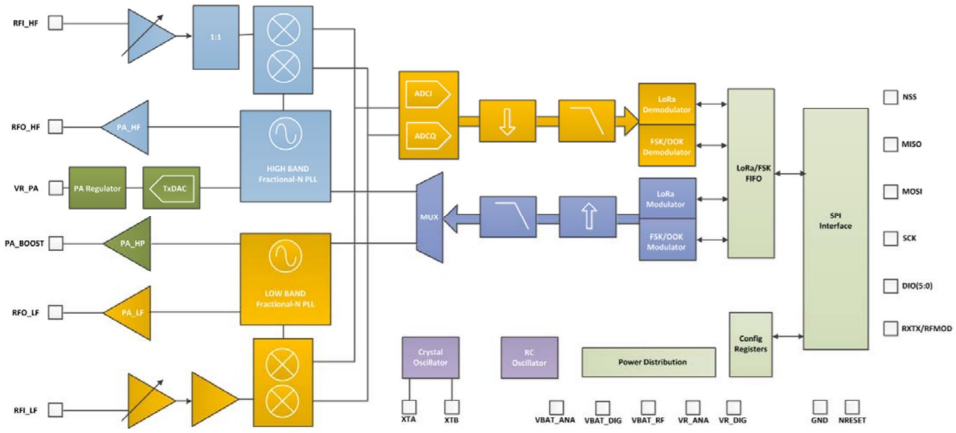


Figure 4.8: RFM95W block diagram

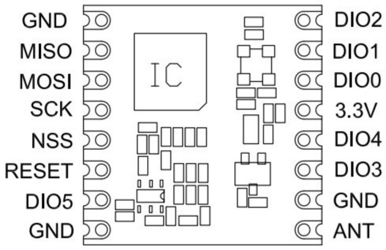


Figure 4.9: RFM95W module pinout

The significant part of the transceiver module is its antenna. Both devices are small, so I decided to use a helical antenna for 868MHz SW868-TH13. In the Fig.4.10 (adapted from [12]) is a graph of antenna's VSWR. The technical drawing of the antenna is in the Fig.4.11 (adapted from [12]).

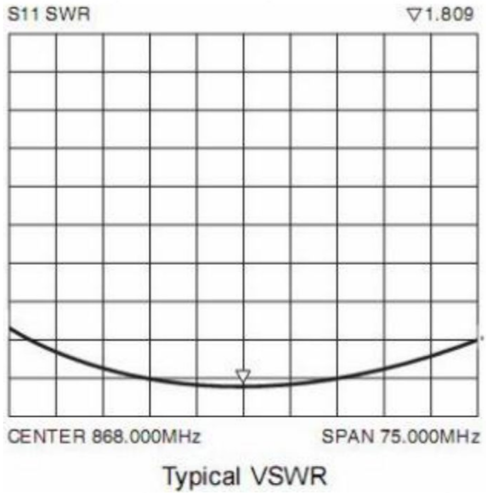


Figure 4.10: VSWR diagram of SW868-TH13 antenna

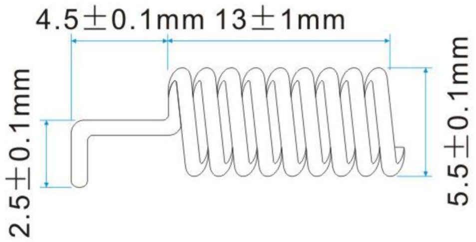


Figure 4.11: The technical drawing of SW868-TH13 antenna

4.3 LoRa Gateway

As it was mentioned above, a gateway is a necessary part of the network. At my location was not any public gateway. The most affordable solution was to make single-channel gateway using single-board computer Raspberry Pi (RPi) and LoRa module RFM95W.

Specifically, the Raspberry Pi 3 B V1. 2 is used. It has (adapted from [3]):

- Quad-Core 1.2GHz Broadcom BCM2837 64bit CPU 1GB RAM
- BCM43438 wireless LAN and Bluetooth Low Energy (BLE) on board
- 40-pin extended GPIO
- 4 USB 2 ports
- 4 Pole stereo output and composite video port

- Full size HDMI
- CSI camera port for connecting a Raspberry Pi camera
- DSI display port for connecting a Raspberry Pi touchscreen display
- Micro SD port for loading your operating system and storing data

The RFM95W is connected to the 40-pin Raspberry Pi GPIO as it is shown in the Fig 4.12. The PCB has been made for the connection of RFM95W to the Raspberry Pi pin header (it is in the Fig. 4.13).

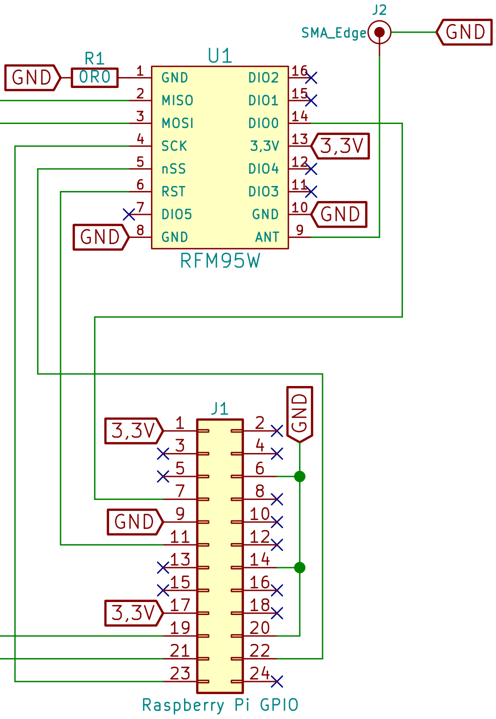


Figure 4.12: RFM95W connection to RPi

4.4 Sensors

This subchapter includes two parts. First one will describe sensors used in the first device of the thesis (the Thermometer) and the second part focuses on sensors of the second device (the Plug Manager).

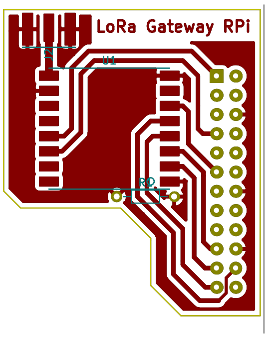


Figure 4.13: PCB for the gateway

4.4.1 Sensors of the Thermometer

The thermometer uses only one sensor to measure temperature and relative humidity. SHT11, humidity and temperature sensor, was chosen. The SHT11 is fully calibrated sensor with digital output, and it has low power consumption. In one chip it contains a band-gap temperature sensor, humidity capacitive sensor element, an analog amplifier, A/D (analog-to-digital) converter, memory and digital interface. With all of this elements, the chip is supplied in surface mountable LCC (Leadless Chip Carrier) package. In the Fig.4.14 (adapted from [14]) is package and pinout of the sensor.[14]

Pin	Name	Comment
1	GND	Ground
2	DATA	Serial Data, bidirectional
3	SCK	Serial Clock, input only
4	VDD	Source Voltage
NC	NC	Must be left unconnected

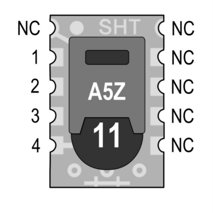


Figure 4.14: Pinout and package of SHT11

The temperature sensing element in SHT11 is based on the measurement of the forward voltage of silicon diode, which is temperature dependent. This principle is known as a band-gap temperature sensor. The humidity measurement uses a unique capacitive sensor element.[14] Both sensors are seamlessly coupled to a 14bit analog to digital converter and a serial interface circuit. To communicate with the sensor, the SHT11 uses specific synchronous, similar to I2C (Inter-Integrated Circuit), serial communication consisting of bidirectional data signal line and the clock signal. The sensor can be connected to I2C bus, but the controller has to switch between the protocols, because SHT11 cannot be addressed by I2C. On the beginning of every communication sequence, the controller has to send a specific transmission start token. After that, the controller sent address and required command. The upcoming communication depends on the sent command. For requesting to measure temperature or humidity, there is time delay determined for sampling the ADC. In this time delay, the data line has to be left open by controller because after the sensor ends measurement, it pulls data low. The controller records this event, sends a clock signal and reads the data from the sensor. For reading/writing command there is no time delay. On the end of every sequence, there is CRC-8 (cyclic redundancy check) checksum. In the Fig.4.15 (adapted from[14]) can be seen time diagram of reading relative humidity from the sensor.

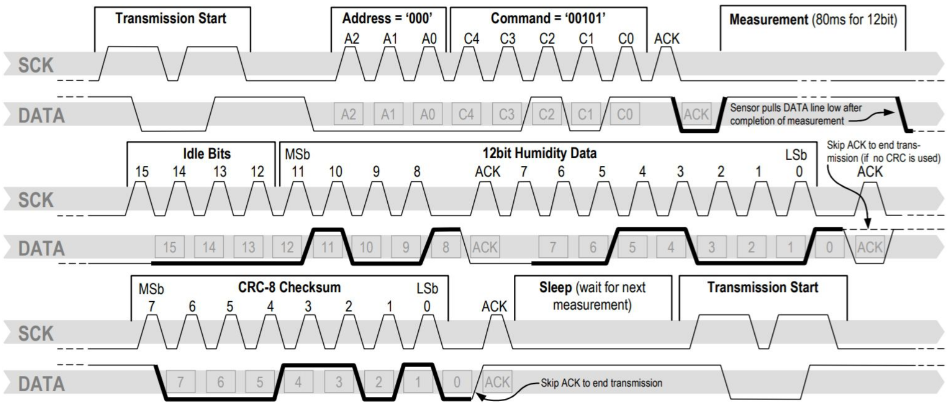


Figure 4.15: Example of measure humidity

The raw data from the sensor has to be converted to get the value in the right quantity. The band-gap temperature sensor is very linear, and the formula is following:

$$t = d_1 + d_2 \cdot SO_T \qquad [^\circ C/^\circ F].$$

Where t is temperature in $[^\circ C]$ or $[^\circ F]$, d_1 and d_2 conversion coefficients and SO_T is temperature readout from the sensor.

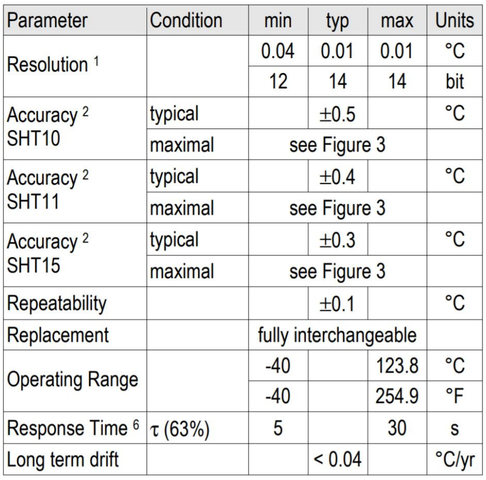
For the converting relative humidity, there are two formulas. One for linear relative humidity and the second one for the real relative humidity:

$$RH_{linear} = c_1 + c_2 \cdot SO_{RH} + c_3 \cdot SO_{RH}^2 \qquad [\%],$$

$$RH_{real} = (T_{^\circ C} - 25) \cdot (t_1 + t_2 \cdot SO_{RH}) + RH_{linear} \qquad [\%].$$

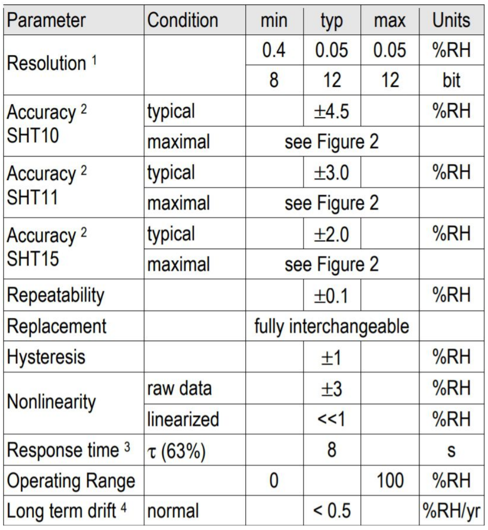
Where RH_{linear} is linear relative humidity, RH_{real} is real relative humidity, c_1, c_2 and c_3 are humidity conversion coefficients, $T_{^\circ C}$ is temperature in $^\circ C$, t_1 and t_2 are temperature compensation coefficients and SO_{RH} is humidity readout from the sensor.[14]

In the Fig.4.16, Fig.4.17, Fig.4.18 (adapted from[14]) are typical characteristics of the SHT11 sensors.



Parameter	Condition	min	typ	max	Units
Resolution [1]		0.04	0.01	0.01	°C
		12	14	14	bit
Accuracy [2] SHT10	typical		±0.5		°C
	maximal	see Figure 3			
Accuracy [2] SHT11	typical		±0.4		°C
	maximal	see Figure 3			
Accuracy [2] SHT15	typical		±0.3		°C
	maximal	see Figure 3			
Repeatability			±0.1		°C
Replacement		fully interchangeable			
Operating Range		-40		123.8	°C
		-40		254.9	°F
Response Time [6]	τ (63%)	5		30	s
Long term drift			< 0.04		°C/yr

Figure 4.16: SHT11 typical temperature sensor characteristics



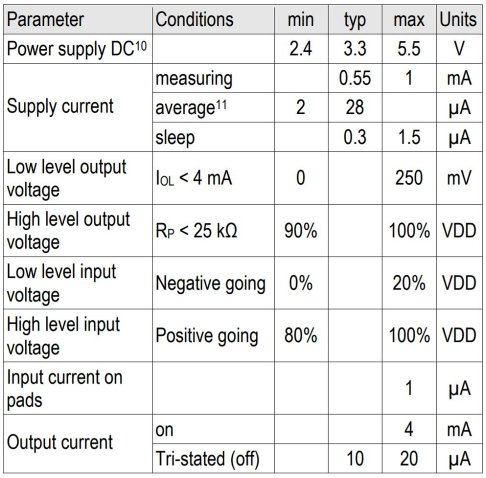
Parameter	Condition	min	typ	max	Units
Resolution [1]		0.4	0.05	0.05	%RH
		8	12	12	bit
Accuracy [2] SHT10	typical		±4.5		%RH
	maximal	see Figure 2			
Accuracy [2] SHT11	typical		±3.0		%RH
	maximal	see Figure 2			
Accuracy [2] SHT15	typical		±2.0		%RH
	maximal	see Figure 2			
Repeatability			±0.1		%RH
Replacement		fully interchangeable			
Hysteresis			±1		%RH
Nonlinearity	raw data		±3		%RH
	linearized		<<1		%RH
Response time [3]	τ (63%)		8		s
Operating Range		0		100	%RH
Long term drift [4]	normal		< 0.5		%RH/yr

Figure 4.17: SHT11 typical humidity sensor characteristics

4.4.2 Sensors of the Plug Manager

For the getting and computing data about the consumption of an AC powered device and the actual state of electrical grid, the two quantities have to be measured. It is the instantaneous voltage and the instantaneous current.

For the instantaneous voltage measurement is used a voltage divider. It consists of six 47k ohms resistors and one 1k ohms resistor in 0603 packages. This circuit drops down the mains voltage from 230V RMS (root mean square), 650V peak-peak to 0.813V RMS, 2.3V peak-peak. So it has a reserve from the nominal mains values. The maximum mains voltage it can measure depends



Parameter	Conditions	min	typ	max	Units
Power supply DC[10]		2.4	3.3	5.5	V
Supply current	measuring		0.55	1	mA
	average[11]	2	28		μA
	sleep		0.3	1.5	μA
Low level output voltage	I_{OL} < 4 mA	0		250	mV
High level output voltage	R_P < 25 kΩ	90%		100%	VDD
Low level input voltage	Negative going	0%		20%	VDD
High level input voltage	Positive going	80%		100%	VDD
Input current on pads				1	μA
Output current	on			4	mA
	Tri-stated (off)		10	20	μA

Figure 4.18: SHT11 typical electrical characterictics

on the supply voltage of microcontroller (with circa 3V supply voltage it can measure up to 300V RMS, 850V peak-peak on the mains). Any of resistors is not accurate, so to get the precise voltage measurement the calibration had to be done.

Because of high current (10A) that can flow to the mains appliance, it was chosen to use a current transformer which converts high current on the primary winding into the small current on the secondary winding. To get the voltage as an output, instead of the current, the load resistor has to be connected to the secondary winding. The voltage drop on the load resistor is directly proportional to current on the primary side. As current transformer AC-1010 is used. The AC-1010 is designed for 10A continuous and 60A peak. The major aspect of current transformers is secondary turns and primary turns ratio. The AC-1010 has a ratio of 1000:1.[13] That means that the current on the secondary side is 1000 smaller than on the primary side. By this aspect, we determine the value of the load resistor. In my case, load resistor has a value of 100 ohms (10A RMS on primary means 1V RMS on load resistor).

There is one important thing to consider about the measurement. The device measures AC (alternating-current) current and voltage, but the microcontroller can measure analog values from GND to VDD. To offset this values precisely to the middle of the GND and VDD a symmetrical voltage divider is used (look at the schema in the Fig.4.19 on the measurement part).

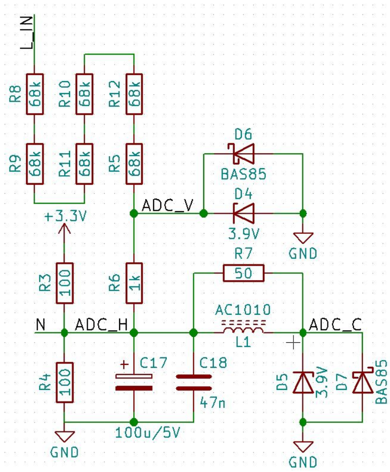


Figure 4.19: Analog part of the Plug Manager

4.5 Overall design

In this section is described the hardware design of the whole devices. The hardware design of both devices consists of all necessary components, wiring among them and design of printed circuit boards (PCB).

4.5.1 The Thermometer design

The hardware design of The Thermometer device contains only necessary components. There is the microcontroller, the LoRa module RFM95W, the helical antenna, the humidity and temperature sensor SHT11, a 3V battery CR2032 and two bypass capacitor. There is also pin header determined to program and to debug the MCU. After that, it can be removed. On the pin header are led out reset pin, data and clock pin, serial out and power supply. The RFM95W's SPI interface is connected on pins with an alternate function of SPI peripheral (PA5, PA6, PA7). Other pins of RFM95W are fully programmable on the microcontroller. The parts are connected to the microcontroller the way that the dimensions of the device's PCB are minimised. The schematic and PCB design is shown in the Fig.4.20, Fig.4.21, Fig.4.22 below.

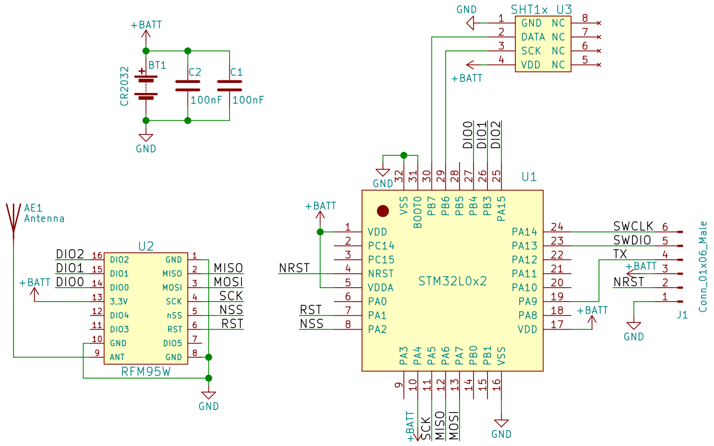


Figure 4.20: Schematic of the Thermometer

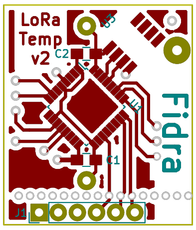


Figure 4.21: Front layer of PCB of the Thermometer

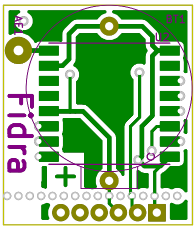


Figure 4.22: Bottom layer of PCB of the Thermometer

4.5.2 The Plug Manager design

The Plug manager device contains much more components, and more operational states than the Thermometer. Let's start to analyse the power supply. The Plug Manager is powered by small 2W AC/DC 230V power supply with 3.3V/600mA DC output[10]. This power supply is directly connected to a filtering capacitor, relay and OLED display. The same power supply also powers the microcontroller and the LoRa module, but it is separated by Schottky diode to achieve powering these components while there is no mains voltage. In this divided power branch there is supercapacitor, to provide energy for a more extended period.

As it was mentioned above, there are circuits for converting voltage and current that is measured. The outputs are connected to the two pins of MCU, PA1 is for voltage and PA3 for current. The Zener and Schottky diode are connected to each analog pin to protect against voltage that could damage them. There is also a measurement of voltage offset. It is used to deduct it from the measured array and for internal comparators for measurement of a period, phase shift and trigger.

The LoRa module RFM95W is connected the same way as in the case of the thermometer. There are differences only in connection with programmable pins of RFM95W, slave select and reset pin.

To control the state of the plug, there is a relay switching the phase of the plug. The relay can switch up to 10A on 230V AC. The operational condition of relay's coil is 150mA at 3V[19]. The microcontroller's pin cannot drive that high current, so there is a driver for the coil. It is made by bipolar NPN transistor BCP56. The disconnecting of inductance load (coil) will cause negative polarity peak, which could damage the transistor. There is a connected diode on the coil which short-circuits this peak. The Fig.4.23 shows it.

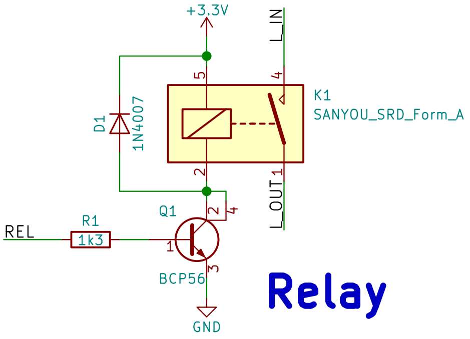


Figure 4.23: Schematic of a relay driver

The user interface of the Plug Manager consists of three capacitive buttons and OLED display. The display is graphical and monochromatical with resolution 128x64 pixels. It has SSD1306 driver inside and communicates via I2C[15]. The display has standardised wiring. The buttons are made of conductive planes isolated by 3mm thick transparent acrylate. The buttons are connected to the microcontroller's pins through a resistor to improve the ESD (electrostatic discharge) immunity. A significant part of this capacitive solution is sampling capacitor (C7, 47nF). In the Fig.4.24 is block diagram of Power Manager and in the Fig.4.25, Fig.4.26 is the designed PCB. The complete schematic is in the Appendix section in the Fig. B.4.

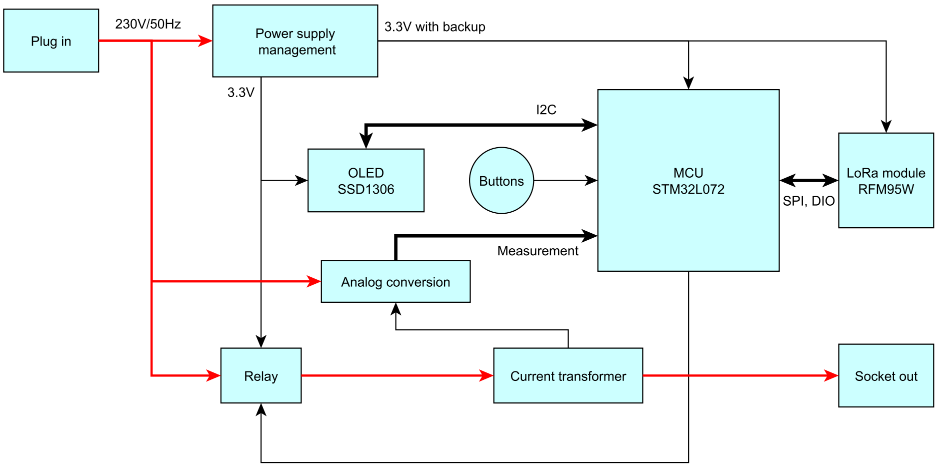


Figure 4.24: Block diagram of the Plug Manager

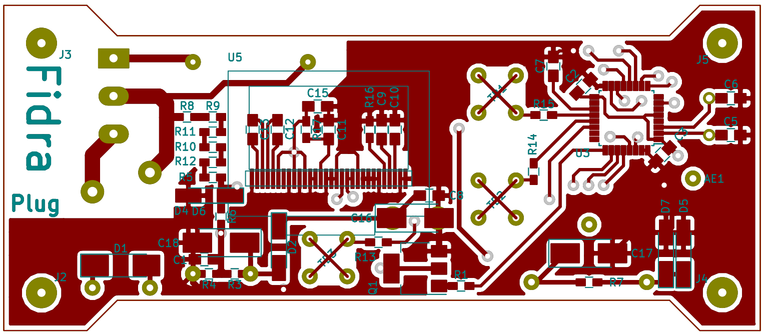


Figure 4.25: Front layer of PCB of the Plug Manager

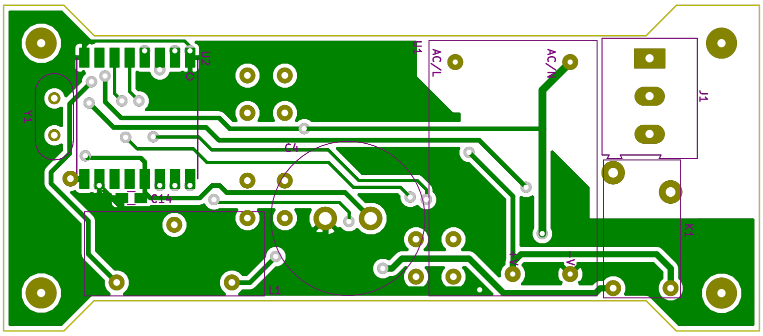


Figure 4.26: Bottom layer of PCB of the Plug Manager

Chapter 5

SW Implementation

5.1 MCU SW

Software for the microcontroller was written in programming language C using Eclipse IDE with System Workbench for STM32 (SW4STM32) extension. Both projects are based on the LoRaWAN software expansion by ST Microelectronics. This software expansion contains libraries supporting LoRa standards and example projects. The library also supports communication with LoRa module RFM95W, UART communication and measurement of battery voltage.

5.1.1 The Thermometer

The Thermometer software is based on example project for End-Node. The difference is in addition of a library for the sensor SHT11. The library is written so that the code does not load processor. It is using interrupts from the timer that is generating a clock signal and whole communication protocol. There is also modified time management for sending and (it sends a data packet every 10 minutes). The program flow is seen in the Fig.5.1.

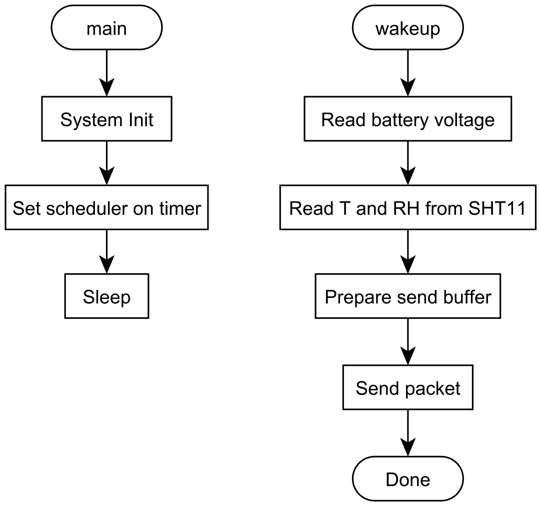


Figure 5.1: Program flow of the Thermometer

5.1.2 The Plug Manager

The software of the Plug Manager It is also built on example project for the End-Node. To the project are added libraries to compute complex fast Fourier transformation (FFT), to manage ADC with a trigger, timings and all computations, the library to create simple User Interface (UI) with capacitive buttons and the library which controls OLED display.

The project uses real-time edge computing. Edge computing is a method used in small devices for optimising cloud computing and reducing data flow to the cloud. The only requirement is to have enough powerful microcontroller. This allows sending data only when a significant event happens (for example overvoltage, overcurrent...). The device measures two periods of mains voltage and current. Simultaneously it saves the measurement into the RAM. The measurement of the period and the phase shift is also simultaneously done. Before every measurement, there is calibration of the ADC and measurement of the offset voltage. Then the device computes voltage RMS, voltage maximum, current RMS, current maximum, power factor, complex, active and reactive powers. The program then calls the FFT function for the current waveform. The user-interface and the OLED display update follows. This sequence is repeated. The service of touch buttons is running independently on the program flow. The timer interrupt is used to cyclic control of the button state. The simplified program flow is also shown in the Fig.5.2.

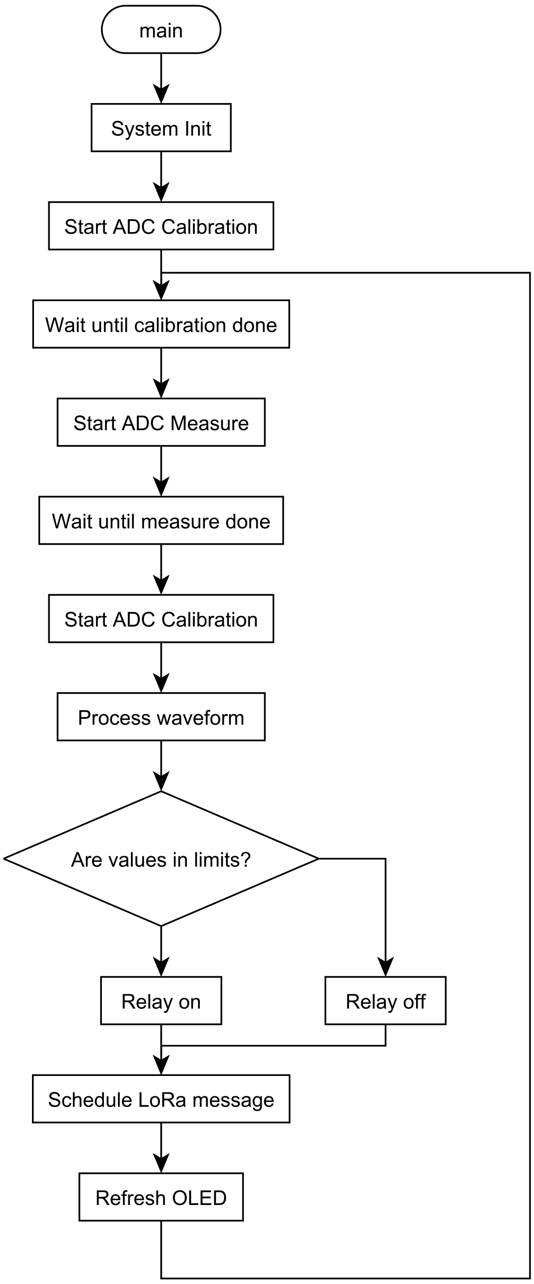
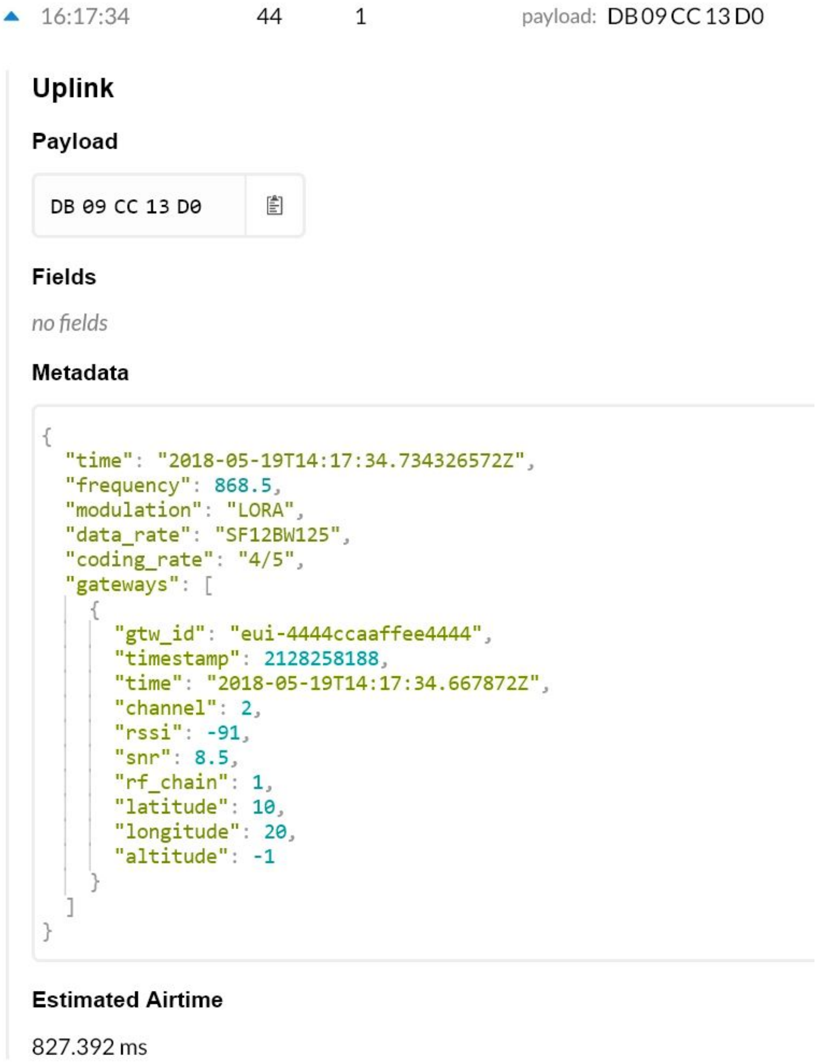


Figure 5.2: Program flow of the Plug Manager

5.2 Raspberry Pi Gateway

The Raspberry Pi runs on Raspbian, the free operating system based on Debian optimised for the Raspberry Pi hardware[4]. The software for the single channel LoRa gateway is freely available on the GitHub[18]. In the code can be chosen what frequency it will receive and on what server will data be loaded. In this case, The Things Network (TTN) was chosen.[11]

The Things Network provides the network for IoT devices based on LoRaWAN standard. It is based on an open community of developers, which can register LoRa gateways and applications for free. The community makes the coverage of LoRa radio network by registering its gateways. The TTN provides the whole network needed to make fully functional applications with generating gateways and end-nodes addresses and keys. The Things Network does not save the received data. The user has to download it from the TTN server. In the Fig.5.3 is TTN received packet from the Thermometer end-node.



▲ 16:17:34 44 1 payload: DB09CC13D0

Uplink

Payload

DB 09 CC 13 D0

Fields

no fields

Metadata

```
{
  "time": "2018-05-19T14:17:34.734326572Z",
  "frequency": 868.5,
  "modulation": "LORA",
  "data_rate": "SF12BW125",
  "coding_rate": "4/5",
  "gateways": [
    {
      "gtw_id": "eui-4444ccaaffee4444",
      "timestamp": 2128258188,
      "time": "2018-05-19T14:17:34.667872Z",
      "channel": 2,
      "rssi": -91,
      "snr": 8.5,
      "rf_chain": 1,
      "latitude": 10,
      "longitude": 20,
      "altitude": -1
    }
  ]
}
```

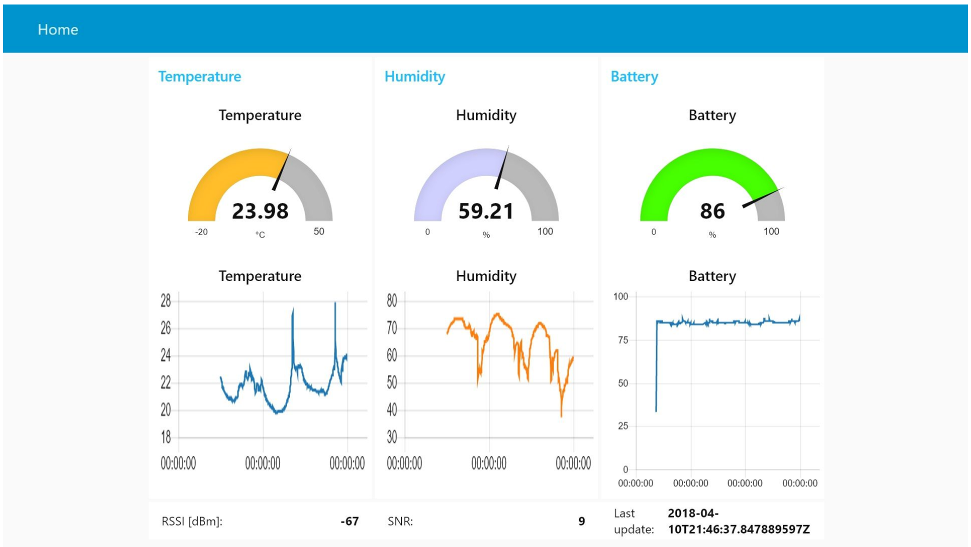
Estimated Airtime

827.392 ms

Figure 5.3: The Things Network received packet

5.3 Node-RED

The last segment of the projects is the visualisation of the received data. It is done by programming tool Node-RED. The Node-RED has a lot of extensions among them belongs getting data from TTN and Dashboard user interface (in the Fig.5.4 can be seen the dashboard user interface of the Thermometer).



RSSI [dBm]: -67

SNR: 9

Last update: 2018-04-10T21:46:37.847889597Z

Figure 5.4: NodeRED Dashboard for the Thermometer

Node-RED is a flow-based programming tool, original developed by IBM's Emerging Technology Services team and now a part of the JS Foundation.[2] The Node-RED application runs on the Raspberry Pi which is used as the gateway. Its editor opens as a web page. In the Fig.5.5 is the example of the program flow of the Thermometer, which gets data from TTN, editing it and making a simple user interface to display data. The data from the TTN are downloaded, and then the data has to be modified for the charts and gauges. This is done by the block 'Data to numbers'. It has one input (whole LoRa packet) and three outputs (battery voltage, temperature and relative humidity). The converting code is in the Fig. 5.6.

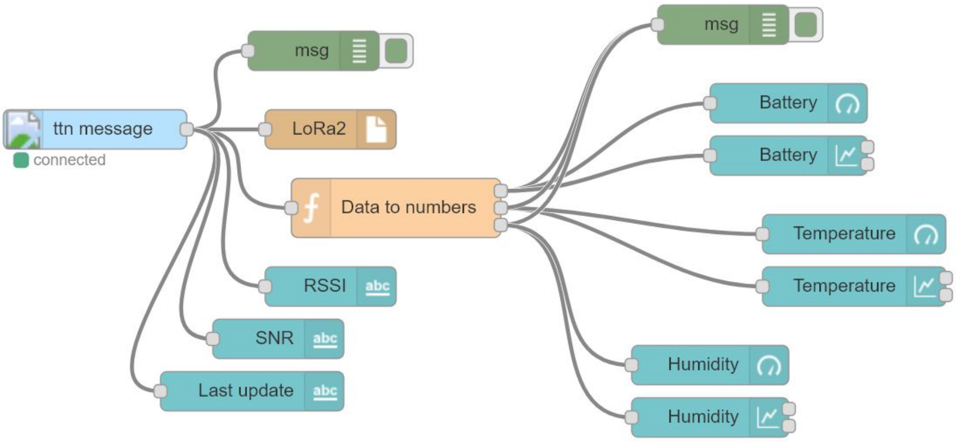
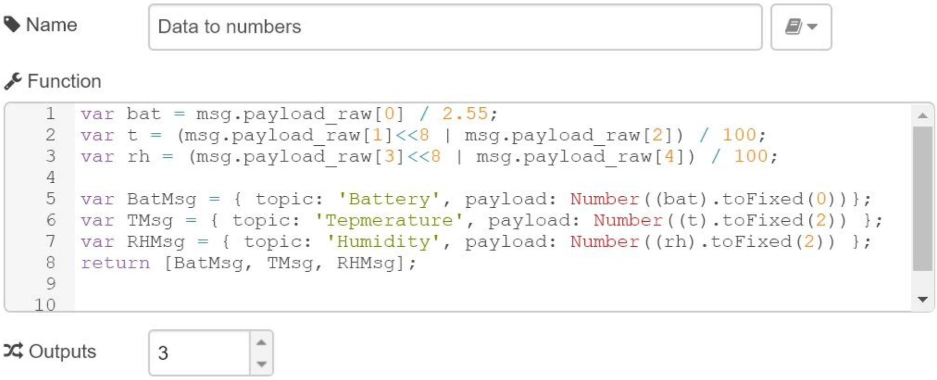


Figure 5.5: Program flow in the Node-RED for the Thermometer



Name: Data to numbers

```
var bat = msg.payload_raw[0] / 2.55;
var t = (msg.payload_raw[1]<<8 | msg.payload_raw[2]) / 100;
var rh = (msg.payload_raw[3]<<8 | msg.payload_raw[4]) / 100;

var BatMsg = { topic: 'Battery', payload: Number((bat).toFixed(0))};
var TMsg = { topic: 'Tepmerature', payload: Number((t).toFixed(2)) };
var RHMsg = { topic: 'Humidity', payload: Number((rh).toFixed(2)) };
return [BatMsg, TMsg, RHMsg];
```

Outputs: 3

Figure 5.6: The converting code of the ’Data to numbers’ block

Chapter 6

Results

This chapter takes a closer look at the final results of this thesis. In the frame of this work, two end-nodes were created, single channel LoRa gateway running on Raspberry Pi (can be seen in the Fig.6.1) and data visualisation. The battery-powered Thermometer, which senses ambient temperature and humidity. And the Plug Manager device which monitors the functional state of an ordinary AC appliance. Both devices were realised on PCBs. The Thermometer's PCB has solder mask surface finish, and the surface of the Plug Manager's PCB is silver plated.

6.1 The Thermometer

During development, the three versions of this device were made. First one was built on a solderless breadboard, and the others had custom made PCB. First PCB was one-sided and was powered by two AA batteries. The second one was double-sided with minimalistic dimensions. There was a problem with power consumption. The battery's lifetime was only 14 days. It was discovered, that the ADR was turned on, but there was not any available gateway which could regulate its data rate. That causes that the end-node transmitted packets with full power and the longest time (about 800ms). After fixing this issue, the battery life is not known, because the battery did not discharged until now. Another issue was with sensing element. The sensor works down to 2.4V, so as the battery voltage drops, the sensor stop works. The data that it sends are optimised so that any loss of resolution and to minimise airtime and power consumption. The Thermometer sends

Figure 6.1: Raspberry Pi single channel LoRa gateway

payload of 5 bytes length. One byte is used for the battery voltage, two bytes are for temperature and other two bytes for humidity. The whole device is in the Fig.6.2. More Figures of the development can bee seen in Appendix.

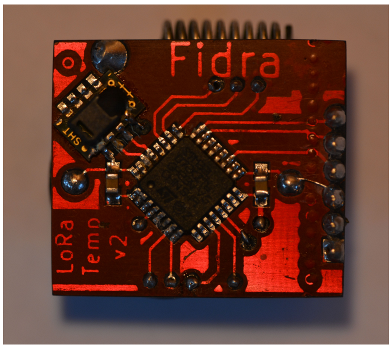


Figure 6.2: The Thermometer

6.2 The Plug Manager

The development of the Plug Manager was done on the solderless breadboard and with safe AC voltage (12V). For the debugging purposes, the OLED display was used. In the final version, the OLED display is used for the user interface which allows to control and set parameters of the device. The case of the final version is made from the modified power strip. The only problem with the Power Manager was with the current transformer. It induces the voltage, due to the surrounding electromagnetic disturbances. To minimise this effect, the shielding is needed, but the shielding would reduce the transmitting power of the LoRa signal. That is why the shielding is not used. The whole device is int the Fig.6.3. More Figures of the development can bee seen in Appendix.

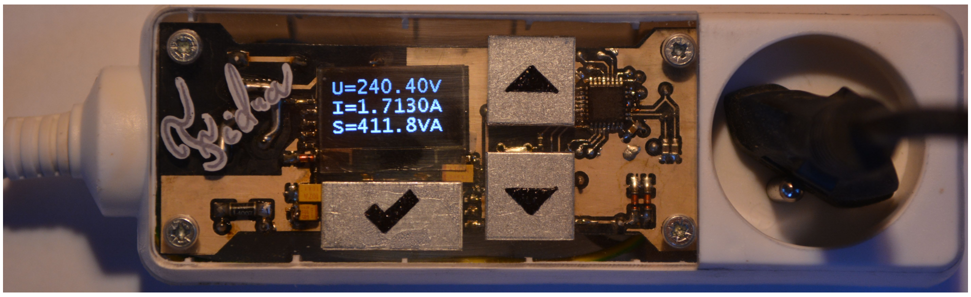


Figure 6.3: The Plug Manager

Chapter 7

Conclusion

This thesis describes the development of monitoring system with LoRaWAN based communication. Two different end-nodes based on ARM Cortex-M0+ core STM32L072 MCU with the LoRa network module RFM95W were designed and realized. Sent data are received via single channel LoRa gateway built with Raspberry Pi. The data visualisation was made using Node-RED programming tool with dashboard extension. For both end-nodes and gateway printed circuit boards were designed and manufactured.

First, simpler module (the Thermometer) measures temperature and humidity using the SHT11 sensor. The Thermometer end-node reads a value from the sensor every 10 minutes and sends it to the LoRa network. The device has low-power feature. Between the measurements, the device goes to the deep sleep mode. The end node is powered with CR2032 3V button cell. The PCB was designed to achieve small form factor (25x25mm).

Second end-node, the Plug Manager measures instantaneous voltage and current of an ordinary AC powered appliance (such as refrigerator, for example). It monitors a proper function of an appliance. The Plug Manager does local processing of measurement. It computes basic properties of mains electricity, current and power drawn by an appliance and it also does the spectral analysis, fast Fourier transformation (FFT), of drawn current. The device has a relay to switch the appliance on or off if it is not properly working. The Plug Manager offers a user interface made by the OLED display with SSD1306 driver and three capacitive touch buttons. The data are sent every predefined time slot or if any event happens (overcurrent, loss of the electric power).

Within this thesis was made single channel LoRa gateway built on Raspberry Pi with LoRa module RFM95W. The gateway works on 868.1MHz, spreading factor 7 and bandwidth 250kHz. The received data are through internet connection sent to The Things Network server. At the same time as the gateway receives packets, the Node-RED application runs on the Raspberry Pi. The two applications for the Node-RED were made. These applications are downloading LoRa packets from The Things Networks servers, modifying them and then the dashboard extension is used to visualise the data.

7.1 Future work

In the future, in the Thermometer will be fitted with BME280 sensor, which can work down to 1.71V. Another improvement will be made in resistance to the environment (waterproof).

The Plug Manager will be controlled only by the downlink packets by LoRa, without display. The power supply will be solved by low-pass filter directly from the mains with backup capacitor.

Appendix A

The Thermometer

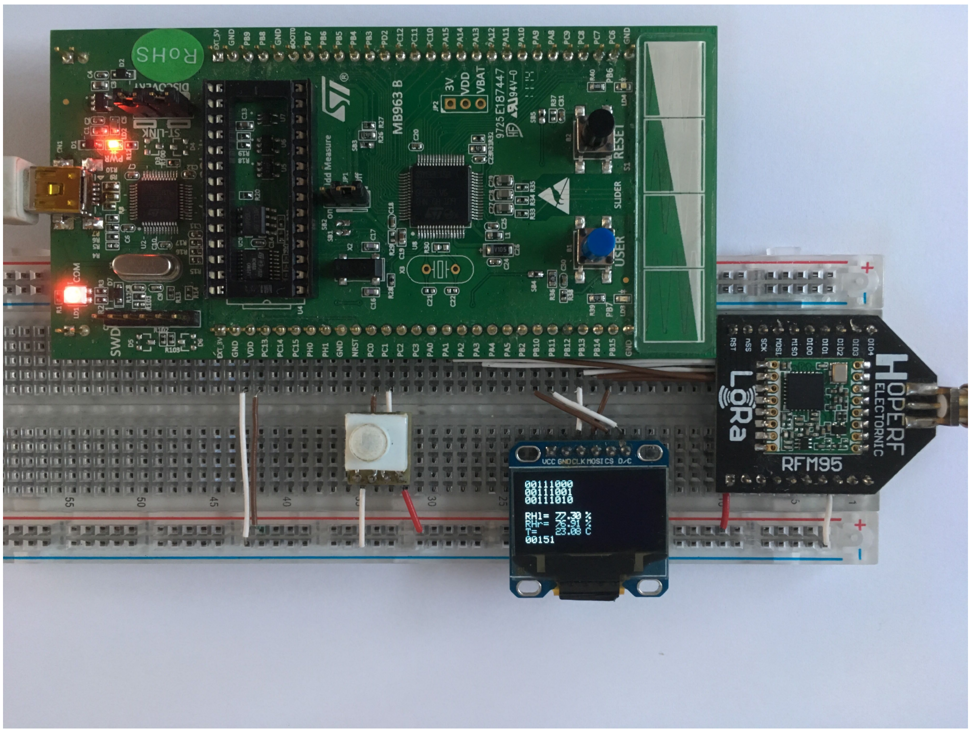


Figure A.1: Developing of the Thermometer device

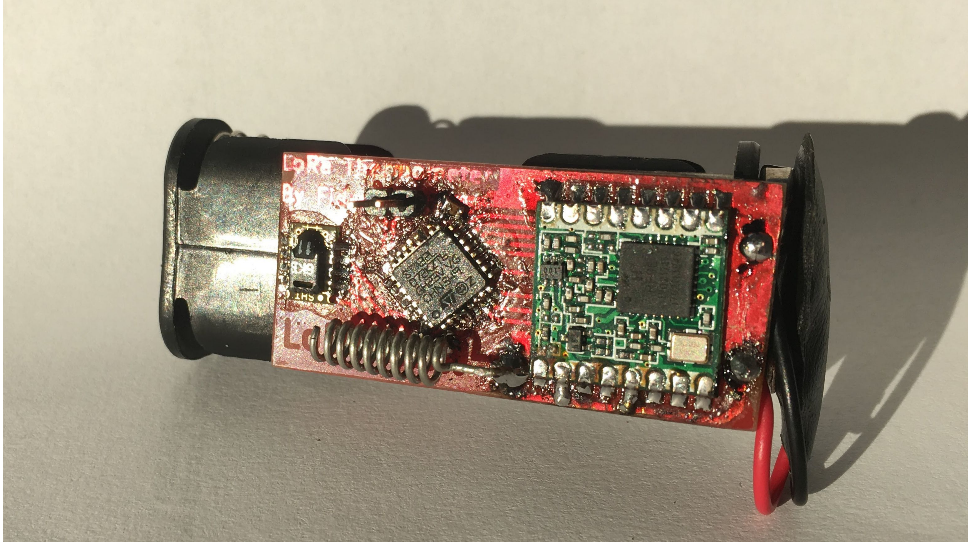


Figure A.2: First version of the Thermometer

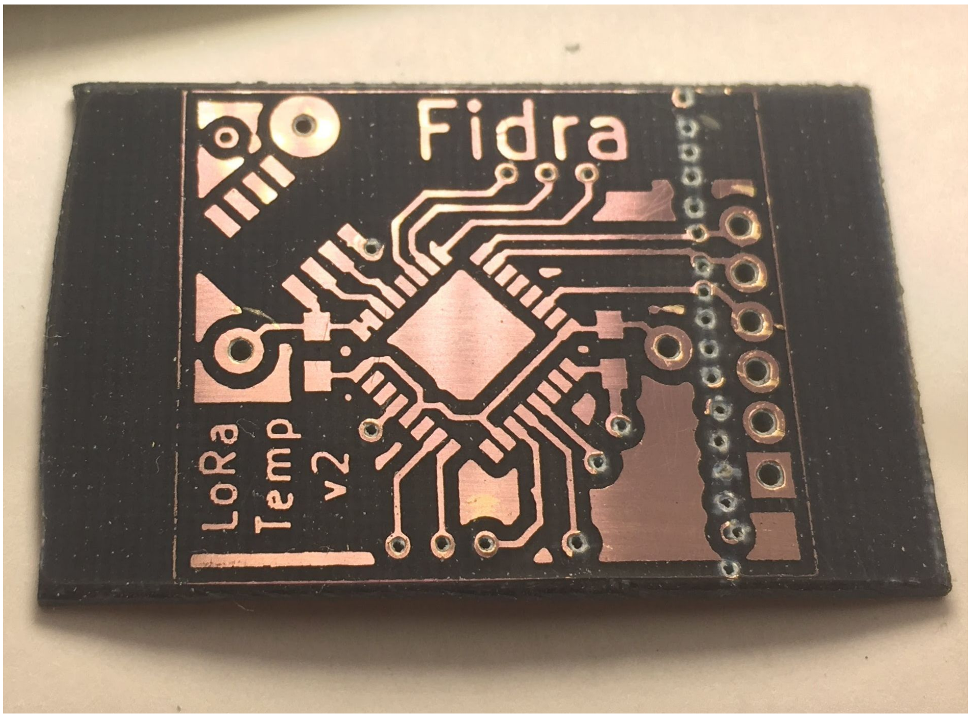


Figure A.3: Making of the Thermometer's PCB

Appendix B

The Plug Manager

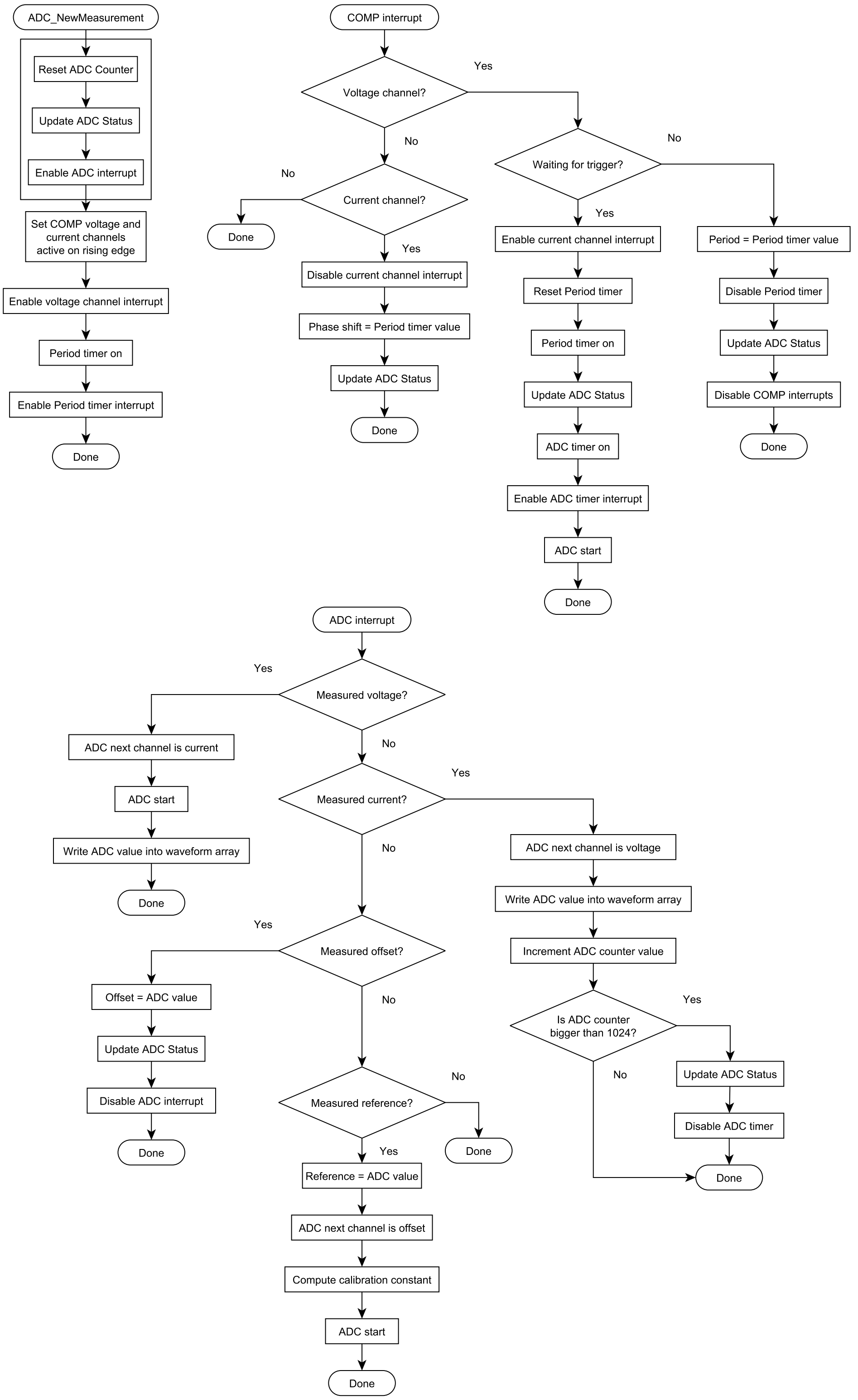


Figure B.1: Flowchart of measurement part of the Plug Manager

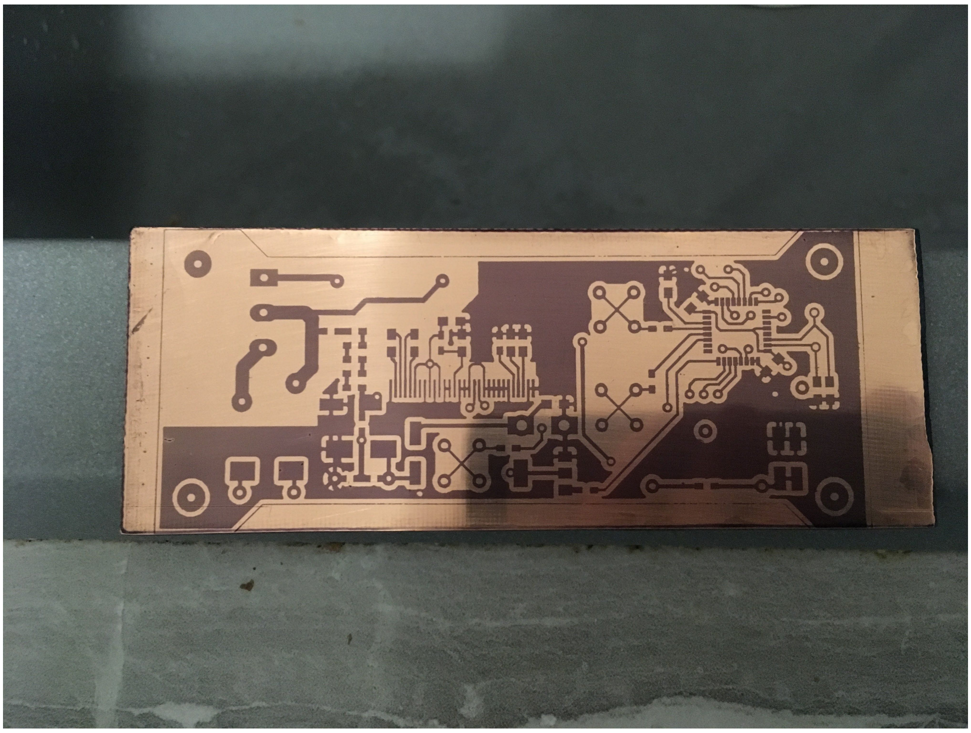


Figure B.2: Making of the Plug Manager's PCB

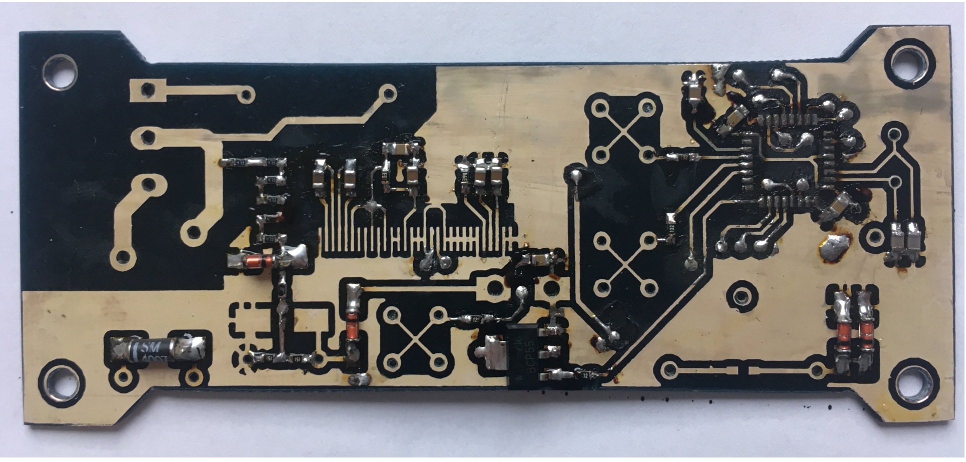


Figure B.3: Soldering of the Plug Manager

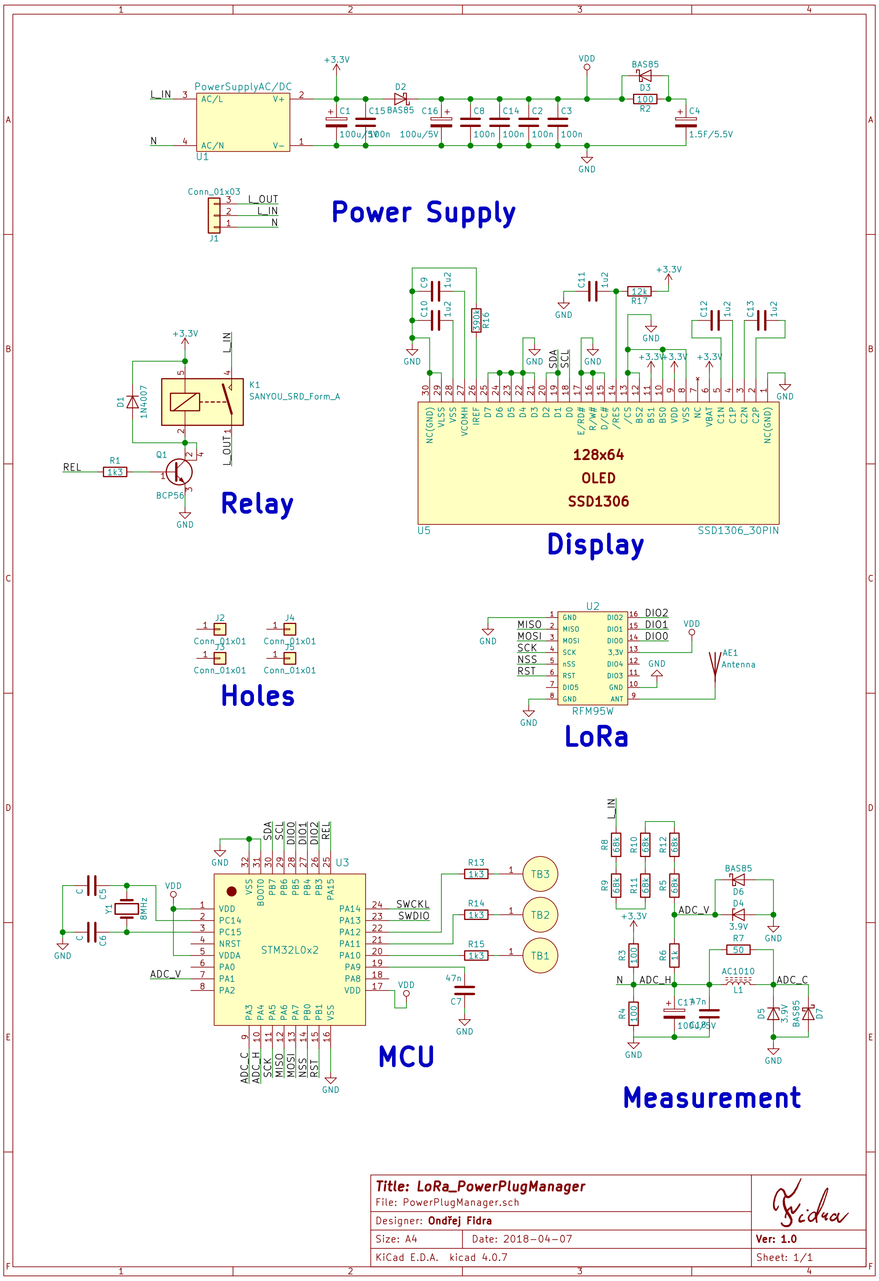
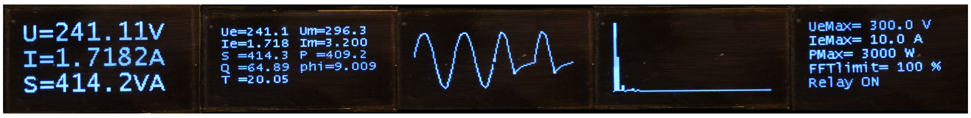


Figure B.4: The Plug Manager schematic



```
U=241.11V
I=1.7182A
S=414.2VA
```

```
Ue=241.1  Um=296.3
Ie=1.718  Im=3.200
S =414.3  P =409.2
Q =64.89  phi=9.009
T =20.05
```

```
UeMax= 300.0 V
IeMax= 10.0 A
PMax= 3000 W
FFTlimit= 100 %
Relay ON
```

Figure B.5: The screens of the Plug Manager's UI

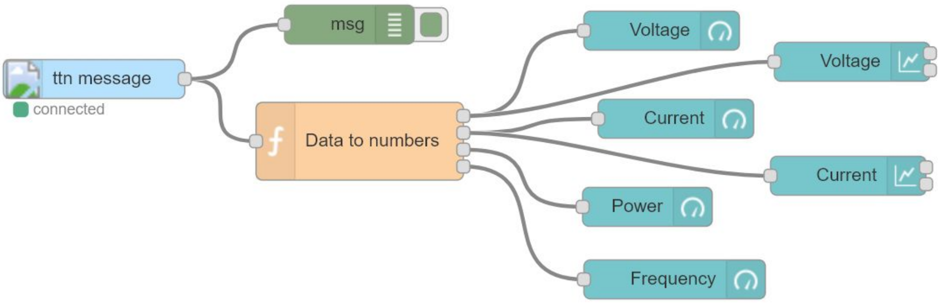


Figure B.6: The Node-RED flow for the Plug Manager

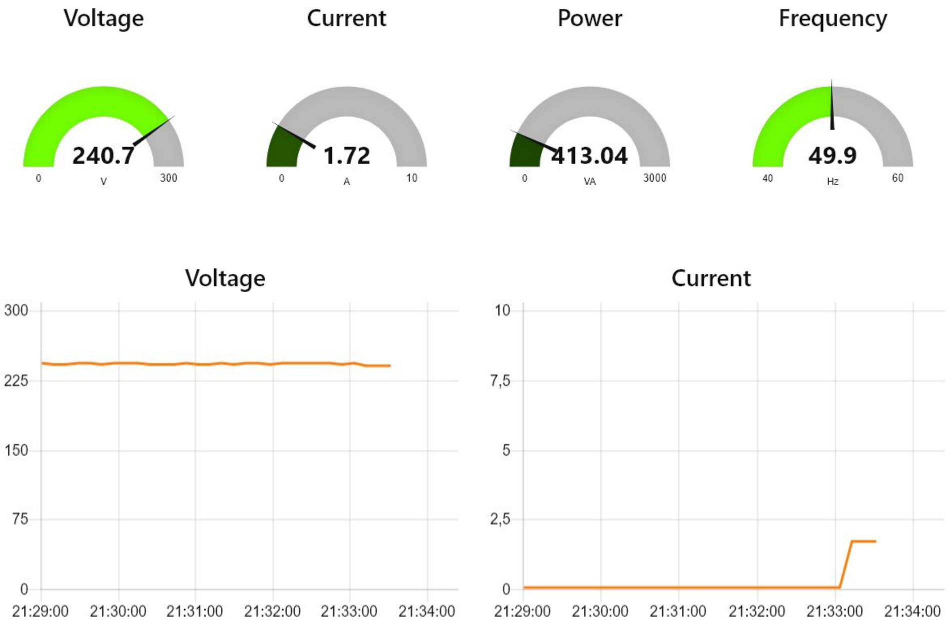


Figure B.7: The Node-RED dashboard for the Plug Manager

Appendix C

Bibliography

[1] Thomas Clausen Aloÿs Augustin, Jiazi Yi and William Mark Townsley. A study of lora: Long range and low power networks for the internet of things. `http://www.mdpi.com/1424-8220/16/9/1466/htm`. Accessed: 2018-05-12.

[2] JS Foundation. About node-red. `https://nodered.org/about/`. Accessed: 2018-05-12.

[3] Raspberry Pi Foundation. Raspberry pi 3 model b. `https://www.raspberrypi.org/products/raspberry-pi-3-model-b/`. Accessed: 2018-05-02.

[4] Raspberry Pi Foundation. Raspbian. `https://www.raspbian.org/`. Accessed: 2018-05-09.

[5] HOPE MICROELECTRONICS CO.,LTD. *RFM95/96/97/98(W) - Low Power Long Range Transceiver Module V1.0*. Accessed: 2018-04-21.

[6] Matthew Knight. Reversing lora. `https://static1.squarespace.com/static/54cecce7e4b054df1848b5f9/t/57489e6e07eaa0105215dc6c/1464376943218/Reversing-Lora-Knight.pdf`. Accessed: 2018-05-08.

[7] Inc. LoRa Alliance. What is the lorawan$^{\mathrm{TM}}$ specification? `https://lora-alliance.org/about-lorawan`. Accessed: 2018-04-25.

[8] LoRa Alliance, Inc. *LoRaWAN*^{TM} *1.1 Specification*, 2017. v 1.1.

[9] LoRa Alliance, Inc. *LoRaWAN*^{TM} *Backend Interfaces 1.0 Specification*, 2017. v 1.0.

[10] MEAN WELL Enterprises Co., Ltd. *IRM-02 series*, 2016. Rev 5.

[11] The Things Network. The things network. `https://www.thethingsnetwork.org/`. Accessed: 2018-05-17.

[12] Ltd. NiceRF Wireless Technology Co. Sw868-th13. `http://nicerf.com/Upload/ueditor/files/2018-04-17/868MHz%20copper%20spring%20antenna%20SW868-TH13%20V2.2-708de313-1676-4bee-82db-cf516b61cf7f.pdf`. Accessed: 2018-04-05.

[13] Nuvotem Talema. *AC1010 10 Amp Current Transformer*, 2006.

[14] Sensirion AG Switzerland. *Datasheet SHT1x*, 2008. V 4.0.

[15] Solomon Systech Limited. *Advance Information SSD1306 128 x 64 Dot Matrix OLED/PLED Segment/Common Driver with Controller*, 2008. Rev 1.1.

[16] STMicroelectronics. *STM32L072x8 STM32L072xB STM32L072xZ*, 2017. Rev 4.

[17] STMicroelectronics. *Ultra-low-power STM32L0x2 advanced Arm®-based 32-bit MCUs*, 2017. Rev 5.

[18] Thomas Telkamp. Single channel lorawan gateway. `https://github.com/tftelkamp/single_chan_pkt_fwd`. Accessed: 2018-01-04.

[19] Tyco Electronics Corporation, a TE Connectivity Ltd. company. *Miniature PCB Relay OJ/OJE*, 2014. Rev 0114.

Appendix D

CD