Low-power processing and communication

Bachelor Thesis

E.E. Bronkhorst
D.L.J. Kappelle

TU Delft

Low-power processing and communication

Bachelor Thesis

by

Erné Bronkhorst
Daniël Kappelle

to obtain the degree of Bachelor of Science
at the Delft University of Technology,
to be defended on Monday July 2, 2018 at 1:30 PM.

TU Delft

Contents

List of Figures v

1 Introduction 3
- 1.1 Problem definition . . . 3

2 Programme of Requirements 5
- 2.1 Requirements of whole system . . . 5
  - 2.1.1 Technical specifications . . . 5
  - 2.1.2 Geometric specifications. . . . 6
  - 2.1.3 Environmental specifications . . . 6
- 2.2 Requirements of subsystem. . . . 7
  - 2.2.1 Power consumption . . . 7
  - 2.2.2 Range and applicability . . . 7
  - 2.2.3 Sensing . . . 7

3 Design 9
- 3.1 Communication Protocol . . . 9
  - 3.1.1 Wi-Fi. . . . 10
  - 3.1.2 Bluetooth Low Energy (BLE) . . . 10
  - 3.1.3 GSM . . . 10
  - 3.1.4 ZigBee . . . 10
  - 3.1.5 Narrowband IoT (NB-IoT) . . . 10
  - 3.1.6 LoRaWAN . . . 10
- 3.2 LoRa Module . . . 11
- 3.3 Micro controller. . . . 12
  - 3.3.1 XLP series (PIC) . . . 13
  - 3.3.2 Picopower series (AVR). . . . 13
  - 3.3.3 ARM . . . 13
- 3.4 Temperature Sensor. . . . 13
- 3.5 Transmission cycle . . . 13
  - 3.5.1 System at rest . . . 13
  - 3.5.2 Triggering a cycle . . . 14
  - 3.5.3 Total cycle . . . 14
- 3.6 Overvoltage protection . . . 15

4 Prototyping 17
- 4.1 System overview . . . 17
- 4.2 Voltage source . . . 17
- 4.3 SPI and I^2C bus. . . . 18
- 4.4 LoRa Parameters . . . 19
- 4.5 Prototype setup . . . 20
  - 4.5.1 Receiving circuit . . . 20
  - 4.5.2 Transmission circuit (Development board) . . . 21
  - 4.5.3 Transmission circuit (PCB Implementation) . . . 21

5 Measurements 23
- 5.1 Method . . . 23
  - 5.1.1 Power Consumption . . . 23
  - 5.1.2 Range . . . 23

5.2 Power consumption results . . . 24
5.2.1 Total system . . . 24
5.2.2 Effects of LoRa parameters . . . 26
5.2.3 Sleep currents . . . 27
5.3 Range results . . . 29

6 Discussion **31**
6.1 Receiving network . . . 31
6.2 Receiving LoRa gateway placement . . . 31
6.3 LoRa configuration . . . 31
6.4 Power consumption . . . 31
6.4.1 Sleeping components . . . 32
6.4.2 LoRa parameters . . . 32
6.5 Sensing . . . 32
6.6 Geometry . . . 32
6.7 Environment . . . 32

7 Conclusion **33**

Bibliography **35**

List of Figures

1.1 System overview 4

3.1 A schematic overview of the entire subsystem. 9

4.1 The implementation of the entire subsystem. 17
4.2 Schematic overview of how the ground can be disconnected from the whole subsystem by switching the N-channel MOSFET. 18
4.3 The SX1272 LoRa transceiver on break-out board. 20
4.4 The implementation of RX circuit. 21
4.5 The Arduino acts as an AVR programmer to program the ATtiny84a. The Arduino is connected to a computer over USB. 22
4.6 PCB layout of the entire IoT sensor. The right side of the PCB is the subsystem described in this report. 22

5.1 Either one of the three components with a shunt connected. 24
5.2 The drop in voltage over the capacitor during one cycle, from 3.34V to 2.46V 25
5.3 Current profile of the temperature sensor 25
5.4 Current profile of the LoRa module. 25
5.5 Current profile of the micro controller. 25
5.6 Power consumption profile of the micro controller 26
5.7 Power consumption profile of the LoRa module 26
5.8 Power consumption profile of the temperature sensor 26
5.9 The consumed energy for transmission of one byte with differen TX output powers. 27
5.10 Energy consumption for different spreading factors. 27
5.11 Time-on-air for different spreading factors. 27
5.12 Energy consumption for different bandwidths. 28
5.13 Time-on-air for different bandwidths. 28
5.14 Energy consumption for different coding rates. 28
5.15 Time-on-air for different coding rates. 28
5.16 Energy consumption for different packet sizes. 28
5.17 Time-on-air for different packet sizes. 28
5.18 Received packets during line of sight measurement. 30
5.19 Received RSSI with incrementing number of concrete floors on different TX levels. 30

Preface

This report discusses one part of a project consisting of three parts. The goal of the project is to design a battery less sensor powered by RF energy fields present in an office environment. The sensor is able to measure the temperature and is able to communicate wirelessly with a low power communication protocol. In this project the use case of a building will be explored, where the climate and lighting of different rooms can be adapted, be monitored and be adjusted remotely. Proposed by the dean of faculty EEMCS of the University of Technology Delft Prof. dr. John Schmitz this project has started as a Bachelor Graduation Project for the B.Sc. Electrical Engineering. This project was assigned to a group consisting of two students per subgroup, six students in total.

In this report we will discuss the development of the subsystem providing the processing, sensing and communication of the environmental variable of interest. The power gained from the harvesting and storage subsystems will be used to perform the measurement, and to send the data to the cloud. The main focus of the development of the subsystem will be to make it as low power as possible. Decreasing the amount of power used in a measurement results in a shorter measurement interval of the complete sensor, and thus creating a more accurate representation of the environmental variable of interest.

Enjoy reading this thesis, but keep in mind, don't spend too much energy while reading!

Erné Bronkhorst
Daniël Kappelle
Delft, June 2018

1 Introduction

A lot of electronic devices today consume a lot of energy, which is often stored in a battery. This battery can be recharged with a battery charger or can be replaced by another battery. Today a lot of electronic devices can be charged wirelessly by using inductive coupling, capacitive coupling or magnetodynamic coupling. Applications of these techniques are charging mobile phones, electrical toothbrushes, biomedical microsystems [1] and even electrically driven cars can be charged wirelessly by a coil in the transmitter with a varying electric current passing through which induces a current in a receiver coil placed in the car itself [2].

These charging techniques are based on the experiments of inventor Nikola Tesla. He experimented with lighting a lamp by using resonant inductive coupling of two LC-circuits, varying the physical distance between the transmitter and the receiver. These experiments of Tesla still have impact on today's technology, like he wanted in his days:

> "As regards the production of light, some results already reached are encouraging and make me confident in asserting that the practical solution of the problem lies in the direction I have endeavored to indicate. Still, whatever may be the immediate outcome of these experiments I am hopeful that they will only prove a step in further developments towards the ideal and final perfection." - Nikola Tesla [3]

All these applications are dependent on an external energy source, but what if a device does not need a battery and can operate autonomously? One of the solutions is Radio-Frequency (RF) energy harvesting. An energy conversion technique where electromagnetic waves on ultra high frequencies (UHF), which are present almost everywhere in the air, are transformed into electric power. Billions of radio transmitters everywhere in the world are broadcasting RF energy, including mobile telephones, mobile base stations (GSM), television broadcast stations, radio broadcast stations and internet modules (WiFi/3G/4G). This RF energy can be harvested from ambient sources and can be used for wireless charging. With this technique a very low energy consuming device, like a temperature sensor, can operate without using a battery and can be placed anywhere.

1.1. Problem definition

RF-energy harvesting is a present-day technique which was not a common alternative for energy harvesting in the past. The reason for this was the availability of RF energy. A couple of years ago very little RF energy was available compared to now, due to the fact that today mobile phones are used more and more and internet becomes more important. At this moment some companies are experimenting with autonomous sensors, like NOWI-energy [4], Freevolt [5] and Powercast [6]. These products make use of RF energy harvesting to stretch the lifetime of the battery of the device. The achievement of RF energy harvesting techniques is to have a device that does not need a battery and consumes only energy from electromagnetic waves on radio frequencies. This is very useful for all Internet of things (IoT) applications. Millions of sensors will be used for this technology, and one can imagine that it is not feasible to change the batteries of all these sensors every now and then or to install cables everywhere to connect the sensors to the grid.

Therefore a solution will be presented. In this project a device is designed that will power a temperature sensor with RF energy harvesting and communicate the data wirelessly to a base station, *without using a battery*. This base station will upload the data to "the cloud", where the data is stored. This feature is necessary for the IoT-applications. An overview of the system can be seen in figure 1.1. To harvest the RF energy an antenna will be used and will be matched to a rectifier, which converts the ultra high frequency signal to DC. The energy will be stored in a capacitor and will be used to power the temperature sensor and the data processing unit, which collects the data and transmits it to a base station. With this system design no battery will be needed. The goal is to harvest as much energy as possible and to use this energy as efficiently as possible.

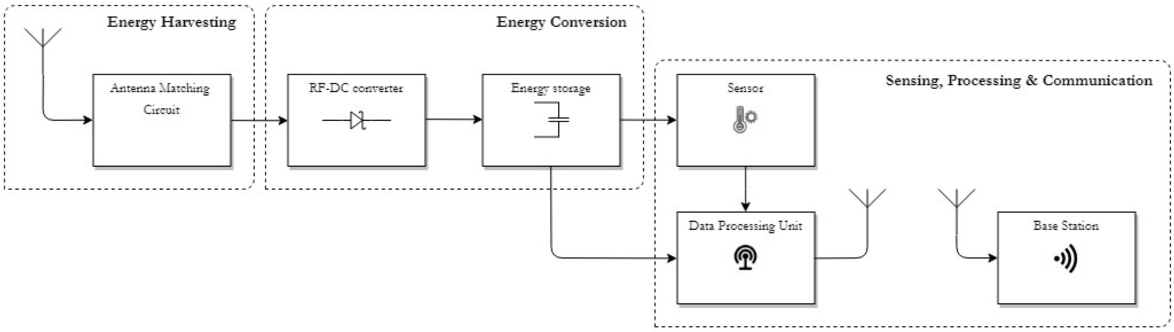

Figure 1.1: System overview

This device can be placed anywhere and operates for an infinite time, without the need of substituting the battery every once in a while and without using cables to provide the device with energy from an external energy source. With this device it is very easy to measure environmental parameters at every location if enough RF energy is available. By collecting all the data in "the cloud", these devices can be used in Smart Buildings: building automation; a category of control and communications technologies that links building systems that are typically controlled separately. These can be systems for electrical distribution, fire alarming, security, heating and air conditioning [7]. Our design will succeed if only one sensor can be powered by RF energy harvesting, without the use of a battery.

It is possible to design a system with these features, but the system is dependent on the available RF energy at the location where the device is placed. The level of ambient RF energy affects the frequency at which the system can sense and communicate and also the functionality of the system. A reduction in RF energy means that the period between each time the data can be communicated increases and a minimum level of available energy has to be reached to convert the RF energy to a voltage level that can be used by our system. The efficiency of each subsystem quantifies the capability of that part. To achieve the best performance each of the subsystems has to be optimized and matched to each other. The performance of the whole system is best observed with the sense-and-communicate frequency.

2

Programme of Requirements

2.1. Requirements of whole system

The goal of this project is to build an *Autonomous battery less sensor for IoT-applications in Smart Buildings*. This chapter will present the requirements of the whole system and the special requirements for the subsystem discussed in this report.

2.1.1. Technical specifications

1-1 Cloud
The sensor should be an IoT-device, which means that it is connected to the cloud. This can be concretized as follows: the device must communicate its measured values to another device (base station or service) which on its turn will make the information available over internet. In other words, the device does not have to be directly connected to the internet per se, as long as its information becomes available on the internet.

1-2 Smart building
The device is intended for use in so called *Smart Buildings*. A Smart Building in this sense uses many sensors and actuators to control for example the climate (temperature, humidity, air quality) or security autonomously, so without the intervention of a human person [7].

1-3 Self powered
There should be no need to physically access the device on a regular basis. This means the device cannot use batteries, because they need replacement every now and then. Also, in order to be versatile and easy to place anywhere, the device should not use a wired power source. The device should therefore be able to extract energy from its environment and this should be done by the use of RF harvesting.

1-4 Measured quantity
For this project, it is sufficient to just measure ambient temperature. This should be possible with a precision of about 0.1 °C. Furthermore, the device must be designed in such a way that at a later stage, the temperature sensor could easily be replace with a different sensor (e.g. humidity or light intensity).

1-5 Integrity
The device should ultimately be used in a Smart Building, thus be connected to certain actuators in an autonomous fashion. It is therefore essential that the data produced by the device is correct. There must be a form of error checking, so that faulty data is discarded.

1-6 Redundancy
The device completely depends on ambient RF sources and in case these are non-existent or the power that can be extracted from them is too little, the device will be unable to function. This is acceptable and should be considered in the application of the IoT-sensors, they might not function from time to time due to lower levels of RF energy. Having many sensors in different locations obviously reduces the chance that no sensor in a certain vicinity functions, so this is a way to overcome the problem.

1-7 Production cost
The devices should be used on large scale, i.e. many sensors should be placed in a building. This requires that the production cost of a single sensor is limited and should not exceed €50.

1-8 Configuration
Either the device must be designed in such a way that no configuration is needed, i.e. it can just be placed somewhere and work right away. Otherwise, the device must have an interface so a serviceman can configure the necessary settings on site.

1-9 Security
The data coming from the device must be secured, so that it is not possible for anyone to interfere. It should not be possible to mimic the device, nor should the data be readable by anything other than the intended receiver.

2.1.2. Geometric specifications

2-1 Office Environment
The IoT-sensor will be designed for operation in an office environment. This office environment is considered to be on the mainland, i.e. to be covered by the GSM-network and not to be exposed to abnormal amounts of RF (e.g. near a microwave oven).

2-2 Location
The IoT-sensor will be designed for operation in the EU. Transmission of RF fields is regulated by the EU defined rules for RF transmission. Also the harvesting of RF is based on operating frequencies in the EU.

2-3 Dimensions
Since the IoT-sensor is designed for use in an office environment, the actual size of the sensor is of great importance. The sensor should not interfere with the architectural and constructive aspects of the Smart Building. The maximum size of the sensor should be somehow equal to that of a normal enterprise grade WiFi access-point. The maximum dimensions are therefore defined as 200x200x100 mm (LxWxH).

2.1.3. Environmental specifications

3-1 Water & dust resistance
The IoT-sensor is designed for indoor use only. Since office environments are known to be not extremely dusty only a case to protect the hardware to big objects is necessary. Therefore the case will comply to IP1X dust protection (protection against objects bigger than 50mm). And since the sensor is designed for indoor use only, it will comply to IPX0 water protection (no special protection) [8].

3-2 Operating temperature
Since the IoT sensor is designed for indoor use only, the operating temperature will be 0-40°C. Assuming that an office will always be above 0°C because it is indoors. The maximum temperature of +40°C gives the capability to measure in (relative) hot rooms, like server rooms.

3-3 Overvoltage & Brown-out
As the input power (RF power) fluctuates quite randomly over time, there might be moments where there is a lot of power available and moments where there is very little. The device must be protected for sudden high levels of power, so that the device is not damaged. Also, the device must be able to cope with very little amounts of power, it is okay if it just stops working until there is enough RF power again.

3-4 Orientation
Since the antenna for the RF harvesting, implemented on the IoT sensor, is direction-sensitive, the direction of installation will be important. The direction of installation will be defined by the antenna design, and therefore determined at installation.

3-5 Minimal available RF power
In order to be able to harvest energy from the RF waves available in the office environment, the amount of available power has to be at least -25 dBm.

2.2. Requirements of subsystem

2.2.1. Power consumption

The most important aspect of the processing and communication subsystem is its power consumption. The power that can be harvested from RF is very little.

The continous available power from the RF harvesting is estimated at roughly 1 μW. Basically, the subsystem must consume less than that. It is not feasible to continuously be processing and communicating with this available power, therefore sensing, processing and transmission will only be done in intervals.

The desired transmission interval is in the order of minutes: 15 - 20 minutes per measurement and transmission. Given a 1 μW continuous power supply and assuming that sensing, processing and transmission takes roughly 100ms, the average power consumption during this time must be $(20 \cdot 60 \cdot 1 \cdot 10^{-6})/(100 \cdot 10^{-3}) = 12$mW. In other words, 1.2mJ of energy may be used.

2.2.2. Range and applicability

The system must be able to send its measured data to either a base station or to an existing service. The key is that the data reaches the *cloud* and can be accessed over internet. This can be accomplished by installing one or more base stations in the building and connecting them to the internet or by directly communicating to the internet (over Wi-Fi for example).

In the case of using base stations, the range must be so that only one or a few base stations are needed in a building. This means that the signal must be able to reach one location from anywhere in the building.

In the case of using an existing service, the range must be so that the nearest radio tower operating the service can be reached.

2.2.3. Sensing

The system must be capable of sensing temperature with at least 0.1°C resolution. The system should however be designed in such a way that the temperature sensor could easily be replaced by an arbitrary other sensor, for example a humidity or light intensity sensor. In order to achieve this, the system must use a standardized method to connect the sensor.

Furthermore, the sensor must have low power consumption, in the order of micro watts.

3 Design

The subsystem that is to be designed here basically consists of three parts (as mentioned before). These parts are: sensing, processing and communication. The requirements of the subsystem have been specified in the previous section.

A schematic overview of the subsytem is shown in figure 3.1. The overview clearly shows how the three components are linked together. The implementations of the components and the interconnections will be explained in this section.

3.1. Communication Protocol

In order for the IoT sensor to transfer measured data, it has to be able to communicate to the cloud. In order to communicate to the cloud, the sensor's data has to be received by either a base station placed inside (or nearby) the sensor, or to a network which is connected to the internet (e.g. the GSM/4G network). Since there are multiple options of communication protocols, some of them will be discussed, indicating their advantages and disadvantages. In this stage no distinction is made between using a base station (peer to peer communication), or using a wide covered network. During choosing a communication protocol, the following points are interesting;

- **Extreme low power**
  Increasing the amount of power needed for communication will end up in a longer measurement interval, making (depending on the application) the measurements less interesting.
- **Sufficient range**
  The range at which the IoT sensor is able to communicate indicates (if needed) how many base stations

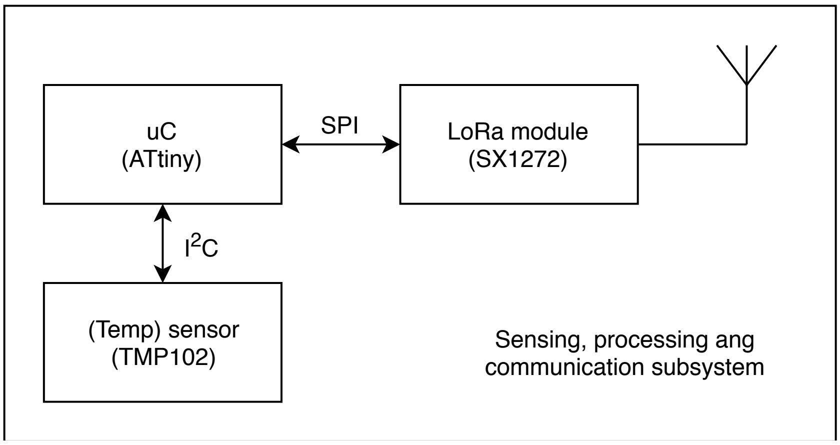

Figure 3.1: A schematic overview of the entire subsystem.

have to be placed in the measurement environment. If the number of base stations is high, the number of cables and base stations that need to be installed makes the use of the (wireless) IoT sensor less interesting. Installing one base station for an environment (an office) is acceptable, since is decreases the number of cables/base stations significantly compared to wired sensors.

- **Easy to connect to the cloud**
  The measurement data generated by the IoT sensor needs to be transferred to the cloud. Since the cloud is accessible through any available internet connection, the receiving network or base station needs to be connected to the internet in some way.

A summary of the advantages and disadvantages of the different protocols can be found in table 3.1.

3.1.1. Wi-Fi

WiFi (IEEE 802.11) is a communication protocol which communicates at the 2.4 GHz and 5 GHz band. Since the WiFi network is commonly used in office environments, it is able to guarantee coverage and connection to the cloud. When the existing WiFi network in a office environment is used, no additional base stations have to be placed to be able to communicate to the cloud. One of the drawbacks of WiFi is its power efficiency. WiFi uses long and difficult handshakes to setup a connection. These handshakes require multiple packets to be received and transmitted by the sensor as well as the base station/access point. Implementing the receive capabilities and encrypting/processing of the handshakes will use more energy than available in the sensor.

3.1.2. Bluetooth Low Energy (BLE)

Bluetooth Low Energy (IEEE 802.15) is a low energy implementation of the Bluetooth protocol. It is designed for low power applications in an personal area network configuration. The relative short range of BLE (~100m) makes the use of on site base stations inevitable. The protocol is also equipped with a procedure to keep the connection alive, which requires packets to be sent often.

3.1.3. GSM

GSM is one of the most broadly supported long range communication protocols in the world. GSM800 operates around 800MHz and has a range from a couple of hundred meters to several tens of kilometers. Since GSM is widely supported by several providers, it has a good network coverage and is able to communicate to the cloud from nearly any given location. GSM is a relatively old protocol and therefore is not really power efficient. The chips needed for GSM communication are relatively expensive and since GSM is provided through a service provider, only paid connections are available. These properties of GSM make it less interesting for the application in the IoT sensor.

3.1.4. ZigBee

ZigBee is a standard communication protocol designed as an extension for Bluetooth and WiFi. It is low-power and has a slow data rate. It is designed for communication over short distances, and therefore has a relatively short range (~100m). When implementing ZigBee as the communication protocol of the IoT sensors, the use of on site base stations is inevitable.

3.1.5. Narrowband IoT (NB-IoT)

Narrowband IoT is defined as a Low Power Wide Area Network (LP-WAN) and focuses on indoor usage. NB-IoT operates at a the frequency band of 900MHz and uses a small bandwidth (~200kHz). NB-IoT is a low power communication protocol. It is able to communicate in an ad-hoc environment and has the possibility to connect to a service provided network as well. Since NB-IoT is relatively new, it is not yet very widely supported.

3.1.6. LoRaWAN

LoRaWAN is also part of the Low Power Wide Area Network (LP-WAN) family. LoRaWAN is specially designed for low-power communication over long distances. Its application focuses on transmission of small data packets at a low data rate with relatively long transmission intervals. Since LoRa is a Wide Area Network, it is designed for transmission over large ranges (~10-15km), making it the ideal communication protocol for implementation in the IoT sensor. LoRa is able to communicate in an ad-hoc environment and has the pos-

Protocol	Advantage	Disadvantage
Wi-Fi	+ Directly connected to the internet + No additional base station needed + Already present in office environments	- Not very power efficient - Long & difficult handshakes needed to connect
Bluetooth Low Energy (BLE)	+ Very power efficient	- Short range (~100m) - Need for additional base stations - Need to send packets often to keep connection alive
GSM	+ Good network coverage	- Very power inefficient - Expensive chips needed - Only via paid network service
ZigBee	+ Power efficient	- Short range (~100m) - Need for base stations
Narrowband IoT	+ Power efficient + Long range (~10km) + Network service present (T-Mobile/Vodafone), so base stations not needed	- Currently less supported
LoRaWAN	+ Power efficient + Long range (~10km) + Network service present (KPN), so base stations not needed	

Table 3.1: Advantages and disadvantages of different communication protocols

sibility to connect to a service provided network. In The Netherlands such a LoRa network is provided by KPN.

While choosing a communication protocol for the IoT sensor, the following considerations have been made.

- Since BLE and ZigBee have a short range of approximately 100 meters, multiple base stations will be needed in the measurement environment, requiring the need for installing cables and mounts for the stations, and therefore are not interesting for implementation in the IoT sensor.

- GSM and WiFi do not meet the requirement of low-power communication, therefore increasing the measurement interval to a length in which the IoT sensor will no longer deliver interesting information.

- Narrowband IoT and LoRa both seem good options and are more or less equal competitors. In our application neither of the two appears to be significantly more appropriate than the other [9]. However, at this time LoRa seems to be a little bit more widely supported than NB-IoT, which is why LoRa wins this race in our case.

Concluding these drawbacks and comparing the advantages and disadvantages of the different communication protocols, the best option is LoRaWAN. It is low power by design, and has many transmission parameters in order to be able to control the amount of transmit power, range and data integrity. Since LoRaWAN has the possibility to be configured as an ad-hoc network, but can also be part of a provided network such as that of KPN, the best configuration can be found depending on the application of the IoT sensor.

3.2. LoRa Module

Even though LoRa is a relatively new protocol. It is widely supported and implemented. This also means that there is a handful of LoRa modules available off-the-shelf. These have been optimized for low power consumption and the LoRa protocol. Therefore there is no point in designing a custom LoRa module. The point is in choosing a proper existing module. The following aspects have been taken into account while comparing different modules;

Property	HopeRF RM95W	Semtech SX1272
Operating frequency	868/915 MHz	868/915 MHz
Max. output power	20 dBm	20 dBm
Communication interface	SPI	SPI
Sleep current (register retention)	200 nA	200 nA
Unit price (/1000)	~20 EUR (on breakout-board)	~3.50 EUR
County of origin	China	US

Table 3.2: Comparison of LoRa transceivers [11] [12]

- **Operating frequency**
  Since the IoT sensor is produced for use in the EU, the LoRa transceiver has to be able to receive and transmit data over the EU standardized LoRa operating frequency of 868MHz [10].
- **Communication interface**
  Since the IoT sensor will be controlled by a micro controller, the module needs to be able to communicate with the micro controller over a standardized communication interface (I^2C , SPI, etc).
- **Low-power operation**
  Power consumption is one of the key concepts in the IoT sensor. Since most of the energy of a measurement cycle is lost during transmission of the data, the LoRa module needs to be equipped with enough low-power parameters. Sleep current and processing power needs to be minimized while choosing a module.
- **Availability and support**
  In order to deliver a high quality and low data-loss communication link, the selected module needs to meet the LoRa standards, and since the module is closed for development (while in-chip development is seen as impossible) the support from the manufacturer needs to be of a decent level.

Researching the available LoRa modules results in two broadly used LoRa transceivers, comparison of these modules can be found in table 3.2.

Both transceivers support the correct operating frequency, deliver enough sleep power optimization settings and communicate over SPI. Since Semtech is the market-leader in LoRa modules, documentation and support is better. Since availability and support are an aspect of importance, the Semtech SX1272 is the module of choice for the IoT sensor.

3.3. Micro controller

The subsystem must be capable of reading a value from a (temperature) sensor and then transmitting this value over the chosen protocol (LoRa). Performing these tasks needs a bit of computing and the easiest option to achieve this is by using a micro controller. They can be easily programmed to do all of those tasks. The following aspects are of importance when choosing a suitable micro controller

- **Sleep current**
  The current consumption during sleep mode is very important in case it is desired to keep the micro controller turned on while the storage capacitor is recharging.
- **Active current**
  The active current (i.e. when the micro controller is processing) should be low, so not too much energy is used on processing.
- **Peripherals**
  In order to communicate with the LoRa module, an SPI interface is needed. In order to be able to connect arbitrary sensors, an I^2C interface is also needed.

Micro controllers come in many different types. Researching the available options, a few seem very well suited. For example, the XLP (eXtreme Low Power) series of the PIC family, such as the PIC16LF15313. Another option is from the Picopower series of the AVR family (by Atmel), such as the ATtiny84a. ARM also offers some low-power chips, but they are unnecessarily complicated for this application.

3.3.1. XLP series (PIC)

Microchip produces a series of microprocessors they call the eXtreme Low Power (XLP) series. These chips are optimized for power consumption with sleep currents as little as 9nA and active currents of 30 μW/MHz [13].

For example the PIC12F1822 would be a suitable microprocessor. It has a standby current of 20nA, active current of 30μW/MHz and offers both SPI and I^2C [14].

3.3.2. Picopower series (AVR)

Microchip also offers a series of microprocessors of the AVR family with low power capabilities, the picopower series. The ATtiny84A is an example of this series. It has a power-down current of 100nA and an active current of 210μA at 1MHz [15]. It also supports both SPI and I^2C . Furthermore, the AVR family is widely supported with many available libraries.

3.3.3. ARM

The ARM family also offers low power microcontrollers. These have sleep currents down to 170nA [16]. Most of the ARM chips also offer a number of SPI and I^2C buses. The ARM chips are not very desirable in this case, because they are unnecessarily complex (i.e. they are large and offer a huge number of peripherals) and they are only available in small package that make them difficult to prototype with.

Based on the observations above, eventually the choice was made to use an ATtiny84A. It uses slightly more energy than its PIC competitor, however the AVR family seemed somewhat more supported and easier to implement. The ATtiny84A was chosen over its two variants, the 24A and 44A, because of the flash size. The ATtiny84A, ATtiny44A and ATtiny24A have 8kB, 4kB and 2kB flash sizes respectively. Having 8kB available assures that all code will fit on the microprocessor.

3.4. Temperature Sensor

First the second requirement (from 2.2.3) is addressed. This states that it should be easy to replace the sensor by another type of sensor. This could be achieved in different ways as long as there is a standardized connection. One way is an analog approach where the sensor acts as a variable resistance (like a thermistor). The problem with this is that more complicated sensors might not be easily implemented in this way.

A standard that is used for many different types of sensors is the I^2C protocol. This protocol uses two data lines to digitally communicate. It is possible to connect multiple devices on the same bus, however this is not necessary in our system [17].

A temperature sensor that works with I^2C is the TMP102. It has a low current consumption of 50μA when it is active. It has a resolution of up to 0.0625°C and an error of $\sigma = 0.2$°C [18]. This sensor meets most of the requirements, only the error is slightly larger than specified, but this is acceptable.

3.5. Transmission cycle

As mentioned earlier, not enough power is available to continuously measure, process and transmit. This has to happen only at certain intervals, so that between the intervals the storage capacitor can be charged. This poses two questions: what does the system do between the intervals? And how does the system determine when to initiate a measuring, processing and transmission *cycle*?

3.5.1. System at rest

First it should be determined what happens with the system when it is at rest, i.e. a cycle has not been initiated. All three components offer sleep/stand-by modes, so it should be determined whether the components are put in sleep mode or actually turned off (cut off from the power supply).

The most logical options would be the following:

- **Keep the whole system at sleep**
  In this scenario each of the components is put in its respective sleep mode, thus stay connected to the voltage source.
- **Keep only the microprocessor at sleep**
  In this scenario only the microprocessor stays connected to the voltage source and the other components are disconnected so their current consumption is removed completely.
- **Disconnect the whole system**
  In this scenario none of the components are put in sleep mode, but instead they are all disconnected completely from the voltage source.

The advantage of keeping a component in sleep mode instead of powering it off is that it its registers are kept. This reduces the need to do all sorts of initializations each cycle, which costs energy. On the other hand, sleep mode consumes some energy. There is a crossover point where the one becomes more efficient than the other based on the interval and the power consumptions of setting up and sleeping. From performed measurements, as described later in this report (5.2.3), keeping the components in sleep consumes too much power. Therefore, the choice is made to disconnect the entire subsystem from the power source.

3.5.2. Triggering a cycle

The second question, i.e. how to know when to start a cycle, is discussed now. There are two basic ways to go about the intervals.

- **Time based**
  This assumes the intervals are of same length between each cycle.
- **Charge based**
  The charge of the storage capacitor determines when to send.

The former option needs some kind of oscillator to keep track of time. This will need some energy to operate, although not necessarily much. There is a bigger problem however: the interval is always equal regardless of the received power. This is problematic as the available power varies quite a lot with time. The interval should be chosen long enough so that most probably enough charge has built up in the capacitor to perform a full cycle. There are two risks to this. First of all, if the capacitor was already charged enough earlier, the voltage would increase to a larger value than needed and would most likely be dissipated in voltage regulation, thus wasting energy. The other risk is that the capacitor is not charged enough and the cycle is not completed, this depletes the capacitor of energy without having a successful transmission, thus also wasting energy.

The latter option takes care of these problems. The interval is based on the charge in the capacitor. The challenge is to determine when the capacitor is charged to a sufficient level. This can be done in a number of different ways. The key requirement is that it takes as little power as possible, at least less than the constant input power (from the RF harvesting).

In the conversion and storage subsystem, a DC-DC booster is used to charge the storage capacitor [19]. This is an integrated chip (BQ25504RGTT [20]) which offers many features. One of those features is a *Battery OK* signal. This signal acts as a comparator with hysteresis and becomes a logic high whenever the capacitor is charged beyond a predefined threshold voltage and becomes a logic low whenever the voltage has dropped below a predefined other threshold voltage. This signal can be exploited to turn on the subsystem. More on this in 4.2.

3.5.3. Total cycle

Taking all above into account, a complete cycle can be designed. The following list shows the different parts of the cycle and their order.

1. Subsystem is powered on

2. Micro controller initializes
3. LoRa module is initialized and put to sleep
4. Temperature is read from the sensor
5. LoRa packet is transmitted
6. LoRa module and micro controller are put to sleep

At this point, the energy in the storage capacitor will have depleted much and will start to recharge. Once it has charged enough, the next cycle is started.

3.6. Overvoltage protection

In this project we are mainly focusing on keeping everything as low energy as possible, taking into account very tiny currents and input powers. It is important however, to not forget the possibility of a sudden excess of energy. Assume a serviceman is installing the IoT sensor when he receives a call on his cell phone just next to the RF harvesting antenna. The harvested power might be orders of magnitude larger than expected. In this case, the IoT sensor must not be damaged.

The sensing subsystem is protected from such excesses of power by the DC-DC booster in the conversion subsystem. The chip that provides this functionality (BQ25504RGTT) has a built-in overvoltage protection that can be set to a desired voltage level. By setting this level correctly, the chip will not charge the storage capacitor beyond that point and thus protects the sensing subsystem from suffering from too high voltages.

4

Prototyping

This chapter describes the final implementation of the entire subsystem as well as some prototyping steps. First the total overview will be discussed and then some of the choices in implementation will be elaborated on.

4.1. System overview

As mentioned in the previous chapter and depicted in figure 3.1, the system mainly consists of three parts. In figure 4.1 the entire subsystem is shown as it is implemented. The three components are marked.

The voltage source Vcc and ground are connected to the conversion and storage subsystem. This is further discussed in 4.2. The interconnection between the three components is a combined SPI and I^{2}C bus, this is discussed in 4.3.

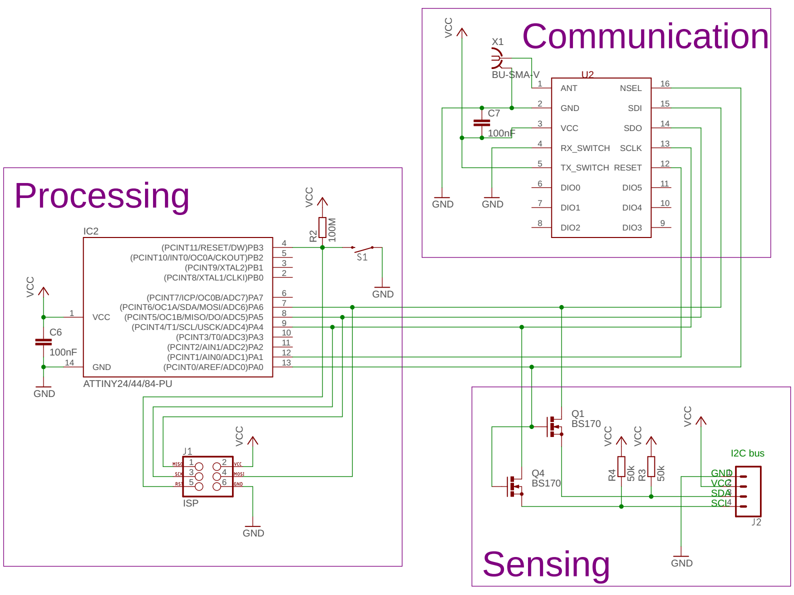

Figure 4.1: The implementation of the entire subsystem.

4.2. Voltage source

The subsystem is supplied by the conversion and storage subsystem of this project [19]. As mentioned in 3.5.1, the system is completely cut off of the supply voltage when it is at rest (i.e. not sensing/transmitting).

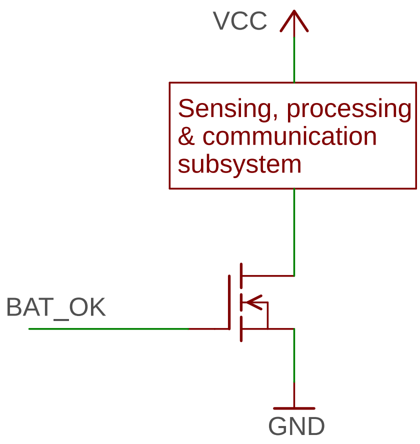

Figure 4.2: Schematic overview of how the ground can be disconnected from the whole subsystem by switching the N-channel MOSFET.

This is achieved by using an N-channel MOSFET that opens up the ground connection of this subsystem. This is schematically shown in figure 4.2. The BAT_OK is a signal coming from the DC-DC converter that is used in the conversion and storage subsystem [19] and becomes a logic high whenever the storage capacitor is charged sufficiently, as described in 3.5.2.

One might argue that it would be better to disconnect the supply voltage instead. However, this poses some difficult problems. In order to understand these problems, it is important to consider the fundamentals of the entire IoT sensor. The voltage source that is used in this subsystem is coming directly from the storage capacitor that is charging. This means that the voltage source can range anywhere from 0V to ~3.5V. To disconnect the supply voltage from the subsystem, a P-channel MOSFET would be used. This is a problem, because in order to open up the connection, a positive voltage must be supplied, but if the voltage across the storage capacitor is still close to zero, this will never happen. Consequently, the MOSFET will act as a conductor, thus supplying the subsystem with power that is then dissipated. This power is then not used to charge the capacitor further and the system will stay in a balance where the MOSFET is conducting and a small current flows through the whole subsystem (but not enough for it to function properly). Hence, the N-channel MOSFET approach is chosen.

4.3. SPI and I^2C bus

To keep the IoT sensor a compact device, a smaller variant of the Atmel chips is used: the ATtiny84A (as discussed in the previous section). This device features what the manufacturer calls a Universal Serial Interface (USI), which supports multiple communication protocols. Among them are SPI and I^2C, which are needed in this case [15]. Note that the micro controller only has one USI bus, so SPI and I^2C will have to share the bus.

Both SPI and I^2C are designed to connect multiple devices on the same bus [17], [21]. They are, on the other hand, not necessarily designed to share a bus. This requires some additional components and also some thought in programming.

SPI uses a *channel select* signal to select a slave device. This signal is normally logic high, only when the master wants to communicate to the slave, the signal is pulled low. This signal can be exploited to disconnect the I^2C device. As can be seen in figure 4.1, the channel select signal (PA0 from the ATtiny) is connected to the gates of two N-channel MOSFETS. Whenever the SPI bus is not used, the channel select signal is high and the I^2C device is connected. Conversely, if SPI is in use, channel select is pulled low and the I^2C device is disconnected. This way, the two devices can communicate over the same bus without interfering.

In software, some considerations have to be made. The USI needs some configuring to use either protocol. Every time the USI is switched from SPI to I^2C or vice versa, this configuration has to be performed.

Both the SCL and SDA signals of the I^2C connection have to be pulled up using a resistor. The value of this resistor was chosen to be 50kΩ. In order to reduce the current flowing through the resistor, the largest possible resistance should be used. However, the maximum resistance that can be used is limited by a few factors. The resistor in combination with the total bus impedance (combination of output and input impedances and the wire itself) defines an RC time, which must be fast enough for the chosen data speed. The following equation can be used to determine the maximum resistance [22]:

$$R_{p,max} = \frac{t_r}{0.8473 \cdot C_b} \tag{4.1}$$

Assuming the conventional frequency of 100kHz ($t_r = 1/f = 1$µs and a bus capacity of ~15pF [1], the maximum resistance is $R_{p_max} = \frac{1 \cdot 10^{-6}}{0.8473 \cdot 15 \cdot 10^{-12}} \approx 78.7$ kΩ. To be on the safe side, a resistor with value 50 kΩ is used.

One might argue that a 50kΩ resistance is quite small and would leak too much current. Indeed, the leakage current when the bus voltage is at 0V would be $I = V/R = 3.3/(50 \cdot 10^3) = 66$ µA, which is a rather large current in this application. However, this leakage current is only drawn whenever a logic 0 is transmitted over the bus for the SDA line or every half clock cycle for the SCL line. As long as the sensor is at rest, no current is drawn.

A quick calculation illustrates the energy consumption. During one sensing cycle, a total of 4 bytes is sent over the I^2C bus: 1 address byte, 1 byte of data (addressing the proper register on the temperature sensor), 1 byte of data (the actual temperature) and 1 byte to put the device in sleep mode. This totals to $4 \cdot 8 = 32$ bits. Assuming that about half of the bits are zeroes, the SDA line will have 16 zeroes, they are zero for an entire period. The period is 1 µs so a total of 16 µs. The SCL line is zero half of every period, so also for 16 µs [17]. In total the consumed energy will be $E = (t_{SDA} + t_{SCL})VI = (16 \cdot 10^{-6} + 16 \cdot 10^{-6})3.3 \cdot 66 \cdot 10^{-6} \approx 7$ nJ. This is 6 orders of magnitude smaller than the total energy consumed by one cycle, so it is safe to say this contribution is negligible. Hence, the 50 kΩ resistance is acceptable.

4.4. LoRa Parameters

The LoRa protocol is defined by the LoRa Alliance [23], the physical layer protocol is closed and proprietary. LoRa uses a proprietary chirp spread spectrum (CSS) modulation to encode data onto sweeped frequency chirps via instantaneous changes in frequency [24]. Some parameters are available for transmission configuration;

- **LoRa Class**
  The LoRa protocol supports 3 operating classes: A, B and C [23]. The LoRa system is uni-direction by design, being from the sensor (node) to the receiver (gateway). Since in some applications bi-directional communication is necessary for the application, the different LoRa classes define receive windows for the sensors. During these receive windows the sensor is able receive messages queued at the gateway. In our application a uni-directional connection suffices.

- **Spreading Factor**
  Spreading Factor (SF), defined as $SF = \log_2(R_c/R_s)$, where R_s is the symbol rate and R_c is the chip rate [25]. A chip is defined as a rectangular pulse onto which the data sequence and the carrier signal are multiplied before it is transmitted. Increasing the SF increases the communication range, but extends the time-on-air.

- **Transmission bandwidth**
  The transmission bandwidth of LoRa packages is either 125, 250 or 500 kHz. Increasing the channel bandwidth decreases the time on air, but also has a negative influence to the receivers sensitivity [26].

- **Coding rate**
  The SX1272 LoRa module employs a form of Forward Error Correction (FEC) in order to recover corrupted bits during transmission [26]. This correction adds a small overhead in the transmitted packet. The coding rate (CR) defines a measure for the amount of added overhead in order to perform the redundancy check. Increasing this coding rate results in a longer air-time, but improves data integrity.

[1] This is a typical value that seemed to work properly. This value depends on the sensor that is connected.

During development the power impact of the different LoRa parameters will be compared. LoRa class A supports simple transmission with an optional acknowledgement from the gateway. In order to minimize the amount of power consumed by the sensor, LoRa class A without acknowledgement will be implemented in the IoT sensor. Following the LoRa design guide [26], the optimum settings for our application should be a spreading factor of 7 (smallest on-air time), a transmission bandwidth of 125kHz (largest receiver sensitivity) and the smallest (4/5) coding rate (smallest on-air time).

4.5. Prototype setup

4.5.1. Receiving circuit

In order to receive LoRa packets sent by the IoT sensor while testing, a development receive circuit is necessary. The receive circuit has to be able to measure the following aspects;

- **LoRa packets**
  The receiver has to be able to receive LoRa packets sent by the IoT sensor. Since LoRa has many transmission parameters, the receiving circuit has to be able to change the transmission parameters easily.
- **RSSI and SNR**
  In order to determine the performance of the IoT sensor, the receiving circuit has to be able to measure the RSSI (Received Signal Strength Indicator) and the signal to noise ratio (SNR). The RSSI displays the total signal power received at the receiving circuit, and the SNR displays the distance from the maximum RSSI to noise floor.
- **Logging of packets**
  The receiving circuit has to be able to log the received packets including their RSSI, SNR and whether the packet payload is corrupted or not.

For the implementation of the receiving circuit, an SX1272 LoRa module mounted on a break-out board was connected over SPI to an Arduino Uno. The break-out board of the SX1272 is displayed in figure 4.3. It is equipped with the SX1272 chip, an antenna matching circuit and an RX/TX multiplexer. Since the receiving circuit is only used for receiving packets, and no acknowledgements have to be sent back to the transmitting module, the RX/TX multiplexer will be locked in the RX (receive) state. The micro controller of the Arduino is able to read and write the registers on the SX1272.

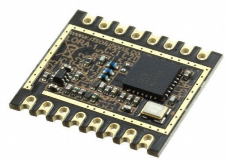

Figure 4.3: The SX1272 LoRa transceiver on break-out board.

The SX1272 has multiple registers which store the RSSI and SNR of the latest received packet. Reading these registers while receiving packets from the LoRa module gives the possibility to store the RSSI and SNR on a computer. The data acquired by the receiving circuit is transferred to the computer via a serial USB connection. On the computer the data is stored in a file for investigation of the packets. The computer is also able to set the transmission parameters (spreading factor, coding rate, transmission frequency etc) by writing registers in the SX1272. An overview of the receiving circuit can be found in figure 4.4.

The receiving circuit as shown in figure 4.4 is able to receive LoRa packets with different transmission parameters, is able to read out the SNR and RSSI and is capable of sending the data to a computer for logging, and thereby meets the requirements of the receiving circuit.

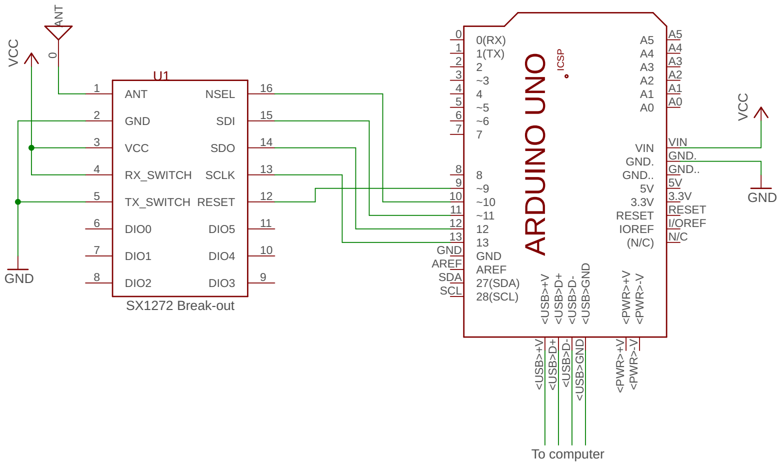

Figure 4.4: The implementation of RX circuit.

4.5.2. Transmission circuit (Development board)

While developing the sensing, communication and processing part of the IoT sensor, the optimum parameters for the micro controller, LoRa transceiver and temperature sensor have to be determined for application. In order to determine the optimum parameters, this subsystem of the IoT sensor is first build on a development board. While designing this development board, the following aspects are important;

- **Configurable**
  It should be easy to program the micro controller in its configuration to try out new parameters and new firmware. It should also be somewhat easy to rearrange components, i.e. desolder them and hook them up differently.

- **Measurable**
  It should be configured in such a way that the current consumptions of the different components can be measured. Also the total supply voltage and current should be easy to measure.

Programming the micro controller
The micro controller that is used, an ATtiny84a, is an AVR micro controller and it can be programmed over an SPI connection. Many AVR programmers are available, however it is also easy to use an Arduino microcontroller as programmer. The Arduino offers an easy interface to any PC over a serial connection and can easily be programmed to act as a programmer. Figure 4.5 shows how the ATtiny84a can be programmed using the Arduino [27].

The ATtiny can be programmed in either C, C++, assembly, or a combination of the three. Some libraries for standard functionally are used, such as the AVR IO library that facilitates a lot of references to IO registers and libraries for SPI and I^2C . For communication with the LoRa module, an adapted version of the Arduino LoRa library [28] is used.

All the code is compiled and linked using avr-gcc (gcc compiler for avr processors) and then uploaded to the ATtiny using avrdude (using the Arduino as programmer).

4.5.3. Transmission circuit (PCB Implementation)

Eventually the total circuit is implemented on a printed circuit board (PCB). The PCB also incorporates the other two subsystems, so the entire IoT sensor is one device. In order to be able to test the different subsystems individually and also to reduce the risk of the whole device failing due to one faulty subsystem, the PCB is designed in such a way that it can be separated. The three subsystems are interconnected with jumpers, which can be removed in case a subsystem is not working. The PCB layout is shown in figure 4.6. The right side of the PCB (to the right of the vertical line) is the subsystem described in this report.

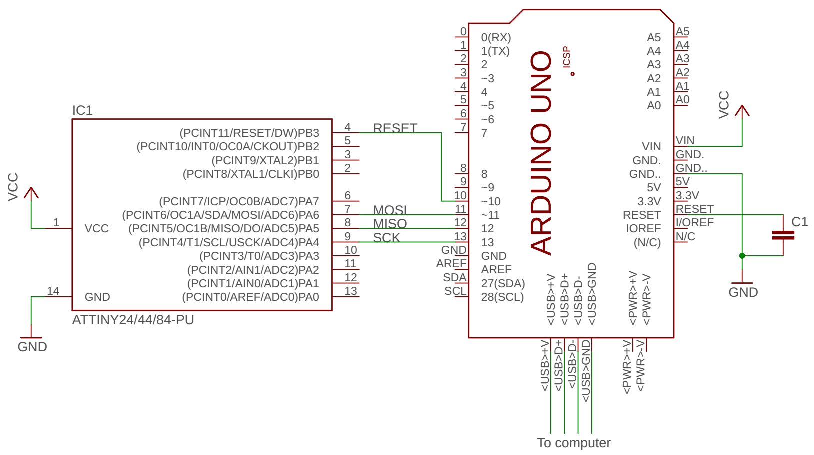

Figure 4.5: The Arduino acts as an AVR programmer to program the ATtiny84a. The Arduino is connected to a computer over USB.

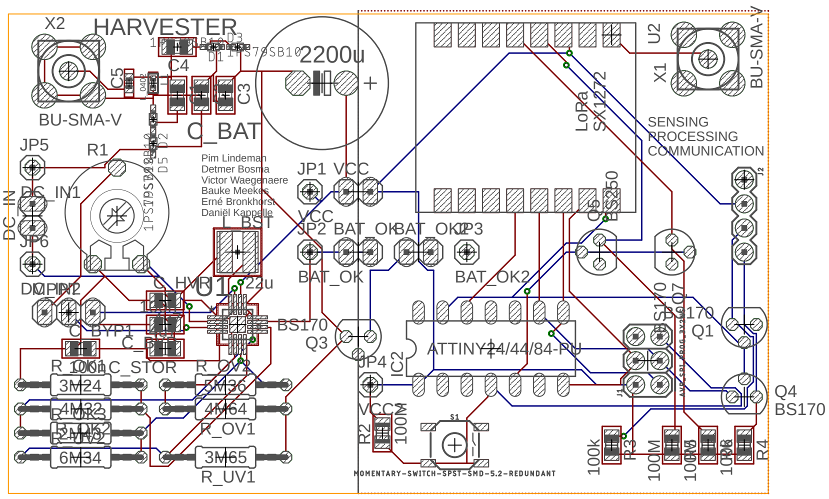

Figure 4.6: PCB layout of the entire IoT sensor. The right side of the PCB is the subsystem described in this report.

5
Measurements

5.1. Method

5.1.1. Power Consumption

It is of utmost importance to know exactly how much current each of the component consumes. Measuring the currents can be categorized into two different types of measurements: static current consumption and time-domain current consumption.

Static current consumption The static current consumption of the components is of particular interest when the components are in sleep/standby mode. In these modes the components consume very little current, but it is important to know exactly how little since there is only in the order of micro watts of power available to the system. The currents in these modes are in the order of a few micro amps or even a few hundred nano amps. Measuring such small currents requires some special equipment. A source measurement unit, like the Keithley 2400 SourceMeter can be used. This device acts as a power supply and measures to great accuracy the current through and voltage across its terminals. Currents down to 10pA can be measured [29].

Measuring these small currents takes some time, so it is not possible to measure these currents in the time domain. That is why this technique is only used for measuring the static current consumption.

Time-domain current consumption A cycle (as described in 3.5) consists of multiple phases, e.g. first the micro controller initiates, then the LoRa module initiates and then a packet is transmitted, etc. It is important to know the current consumption profile in the time-domain so it can be determined how much current each of the phases consumes.

To measure the current in the time-domain, a shunt resistor is placed in series with the component as depicted in figure 5.1. The value of this resistor should be small, so there is only a small (negligible) voltage drop across it. A value of 10Ω is used. An oscilloscope is used to measure the voltage across the resistor and by tweaking the trigger settings, a time-domain current consumption profile is obtained. The current is calculated from the measured voltage by dividing by the used shunt resistance.

5.1.2. Range

In order to determine the range specifications of the IoT sensor, the receiving range of the transmitted signal has to be measured. This range measurement is done in two set-ups.

Line of sight range The line of sight measurement indicates what the maximum distance is between the IoT sensor and the receiving LoRa gateway. When the receiving LoRa module is placed outside the office environment, this measurement will indicate up to how far the receiver can be placed outside the building. Combining the data from this measurement with the inside-building measurement can give an indication of the optimum location of the receiver. This can be either inside the office, or just outside the building

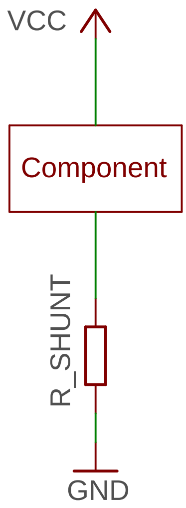

Figure 5.1: Either one of the three components with a shunt connected.

(across the street for example). This measurement is performed by placing a transmitting LoRa module on the 21^{st} floor of the EEMCS Faculty in Delft. The receiving LoRa gateway is placed on the roof of a car. While synchronizing the different LoRa parameters at the sending and receiving module, and logging the received packets, their RSSI and SNR, the range of the module can be determined.

Inside building range Since the IoT sensor will be deployed in an office environment, the inside building range is important in order to determine the number of LoRa gateways that need to be placed in the environment. The inside building range measurement will display the RSSI of received packages versus the number of concrete floors in between the transmitter and receiver. In order to measure the inside building range, a transmitting LoRa module is placed on he 21^{st} floor of the EEMCS Faculty in Delft, and will be received by a receiving LoRa module, which is moved a floor down every 10 transmitted packets. In order to determine the inside building range for different transmit powers, the transmit power is also varied during measurement.

5.2. Power consumption results

5.2.1. Total system

First of all the power consumption of the entire subsystem is evaluated as a function of time. This shows the total amount of energy needed to complete one cycle (3.5) and also shows which components consume the most energy.

For the measurement one cycle is performed including one packet consisting of 3 bytes. The output TX power is set to 2dBm. Figure 5.2 shows the decrease in the voltage over a 1500 µF storage capacitor. The voltage decreases from 3.34V to 2.46V. The energy stored in a capacitor is given by

$$E = \frac{1}{2}CV^2 \tag{5.1}$$

The difference in energy stored in the capacitor is the energy used by the subsystem during the cycle. This can then be calculated as

$$\Delta E = \frac{1}{2}C(V_{before}^2 - V_{after}^2) \tag{5.2}$$

Substituting the values gives

$$\Delta E = \frac{1}{2} \cdot 1500 \cdot 10^{-6}(3.34^2 - 2.46^2) = 3.63\text{mJ} \tag{5.3}$$

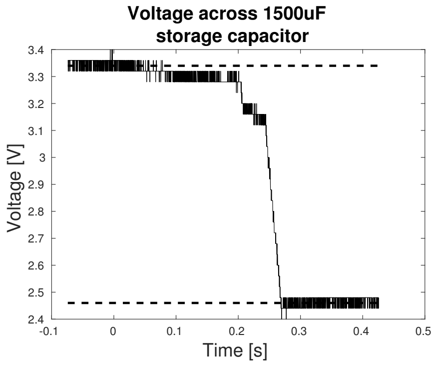

Figure 5.2: The drop in voltage over the capacitor during one cycle, from 3.34V to 2.46V

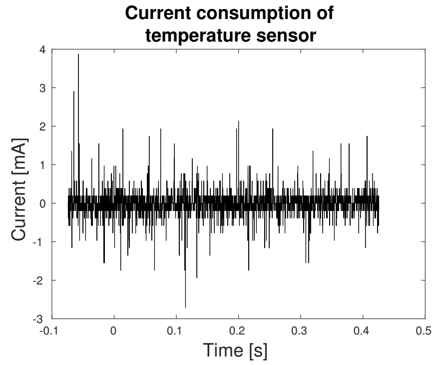

Figure 5.3: Current profile of the temperature sensor

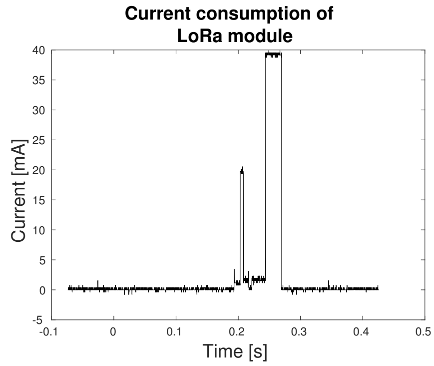

Figure 5.4: Current profile of the LoRa module.

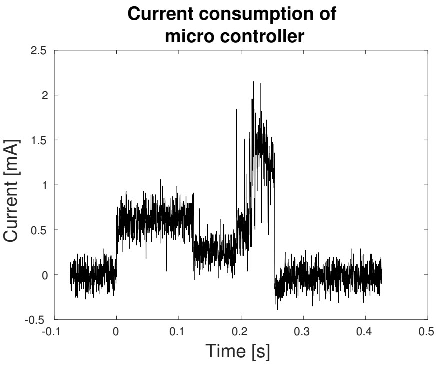

Figure 5.5: Current profile of the micro controller.

The total energy consumed by the system is thus 3.6mJ for one cycle at 2dBm output power.

Now the consumption of the different components is observed. Figure 5.4 shows the current consumption of the LoRa module over time. First, the current consumption is about zero. Then a first peak can be seen, this peak is during communication of the micro controller with the LoRa module. Then there is a second peak which dominates the current consumption, this peak is during the actual transmission of the packet. During this time the power amplifier connected to the antenna is turned on.

Similarly, figure 5.5 shows the current consumption of the micro controller over time. The consumption is in the order of a milliamp, which is close to the lower boundary of currents that can be measured with this setup. Hence, the profile is quite noisy. Nonetheless some observations can be made. From $t = 0$ the micro controller turns on. Around $t = 0.2$s communication with the LoRa module is in progress. After that the micro controller enters sleep mode and the current consumption goes to zero.

The temperature sensor shows no significant current consumption profile during the cycle, as seen in figure 5.3, its sleep current dominates and is discussed in the next section.

In order to determine the power consumption of the three components, a point wise multiplication of the current of the respective component and the voltage in the capacitor is performed in the time domain. Figures 5.6, 5.7 and 5.8 show the power consumptions of the micro controller, the LoRa module and the temperature sensor respectively.

From the power consumption profiles, the total consumed energy can easily be calculated as

$$E = \int P(t)\mathrm{d}t \tag{5.4}$$

which is the same as multiplying the average power by the total time. Table 5.1 shows the results. Two obser-

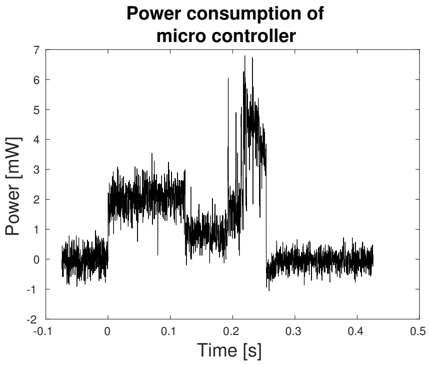

Figure 5.6: Power consumption profile of the micro controller

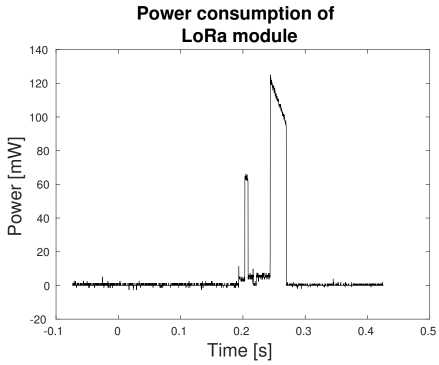

Figure 5.7: Power consumption profile of the LoRa module

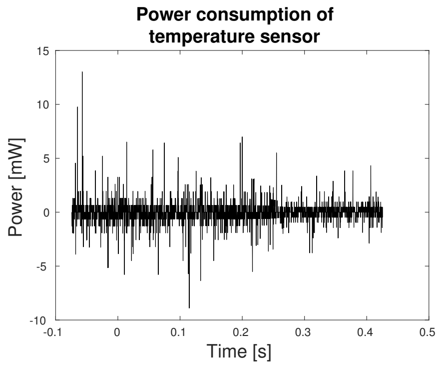

Figure 5.8: Power consumption profile of the temperature sensor

vations must be made first. The total energy consumed is 3.96mJ which is more than the amount of energy consumed from the capacitor. Furthermore, the energy consumed by the temperature sensor has a negative sign. It must be concluded that the measurements are not very precise, which can also be seen by the noisiness of the previous plots.

Nonetheless, the measurements give a good indication of the distribution of energy in the system. The temperature sensor consumes negligible energy during operation. The LoRa module consumes roughly 7 times as much energy as the micro controller. Especially the actual transmission, i.e. when the power amplifier for the antenna is turned on, dominates the power consumption of the subsystem. It is therefore important to lower the TX output power as far as possible.

Component	Energy used
Micro controller	0.50 mJ
LoRa module	3.47 mJ
Temperature sensor	-0.01 mJ

Table 5.1: Distribution of energy over the three components.

5.2.2. Effects of LoRa parameters

In the previous section it became clear that the transmission of packets with the LoRa module is dominant in the power consumption of the subsystem. As discussed in 4.4 LoRa offers a number of parameters that can be tweaked. The parameters all affect properties such as range, packet integrity and time-on-air. In this section the effect on the power consumption is observed. Note that the following measurements reflect the consumed energy by just the LoRa module. The other components are not regarded.

TX output power Perhaps the simplest parameter that can be changed is the TX output power of the LoRa module. This is simply the amount of power that is fed into the antenna. Figure 5.9 shows how choosing

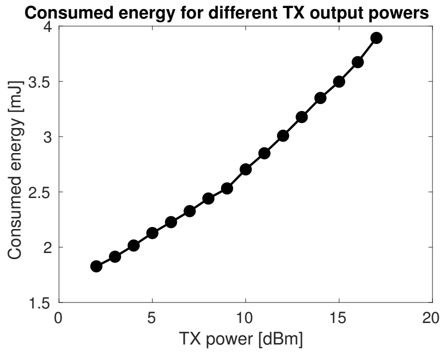

Figure 5.9: The consumed energy for transmission of one byte with differen TX output powers.

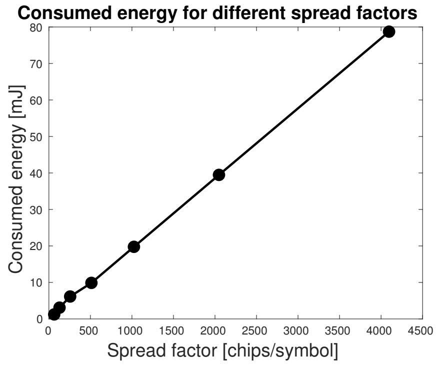

Figure 5.10: Energy consumption for different spreading factors.

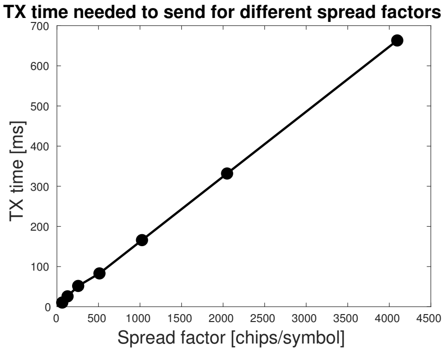

Figure 5.11: Time-on-air for different spreading factors.

higher output powers increases the total energy consumed. It is interesting to notice the linearity of the plot, which is not expected. An increase from 5dBm to 15dBm means a 10x increase in power, but the consumed energy is only about 1.6x as large. At this point there is no good explanation for this behavior. Possibly at lower TX output powers part of the input power is dissipated, but this is unconfirmed.

Spreading factor The spreading factor as described in 4.4 determines the number of chips per symbol. A higher spreading factor means an increased range, but longer time on air. Figure 5.10 shows the energy consumption for different spreading factors. It can be seen that increasing the spreading factor significantly increases the energy required. In figure 5.11 the increase in time-on-air is shown. Just like the energy, the time-on-air significantly increases with higher spreading factors. This is as expected according to 4.4.

Bandwidth LoRa can operate on three different bandwidths, 125kHz, 250kHz or 500kHz. Figure 5.12 shows the decrease in energy for wider bands. Similarly, figure 5.13 shows how the time-on-air decreases with wider bands. This is also in accordance with 4.4.

Coding rate As mentioned earlier the coding rate determines how much overhead is appended to the packet in order to be able to check the packet's integrity. Figure 5.14 shows the increase in energy consumption for higher coding rates. It is interesting to see that the increase is very little. The time-on-air also extends a little bit, as shown in figure 5.15.

Packet size Although the packet size is not really a LoRa parameter, it is interesting to see how LoRa deals with different packet sizes. Figure 5.16 shows the increase in consumed energy for longer packets. It is interesting to see that the consumed energy only increases after certain packet lengths. The same applies for the time-on-air. This is due to the nature of the LoRa protocol in which packets are padded to defined sizes.

5.2.3. Sleep currents

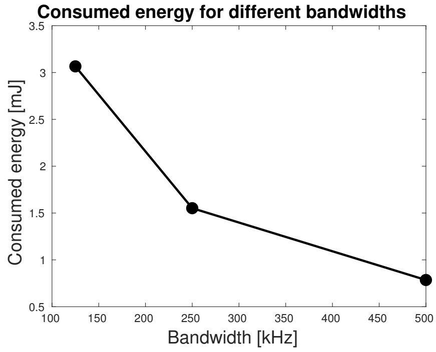

Figure 5.12: Energy consumption for different bandwidths.

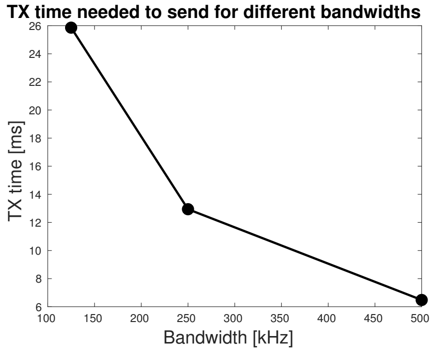

Figure 5.13: Time-on-air for different bandwidths.

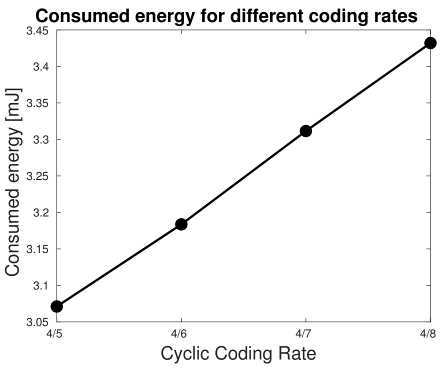

Figure 5.14: Energy consumption for different coding rates.

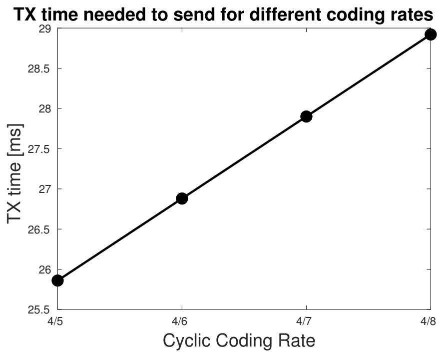

Figure 5.15: Time-on-air for different coding rates.

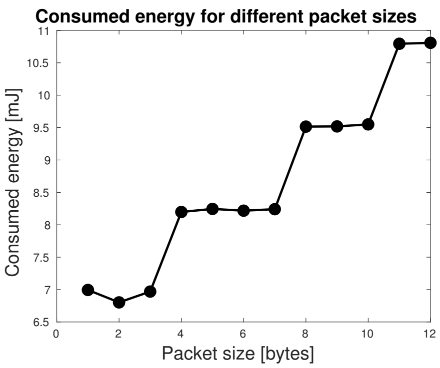

Figure 5.16: Energy consumption for different packet sizes.

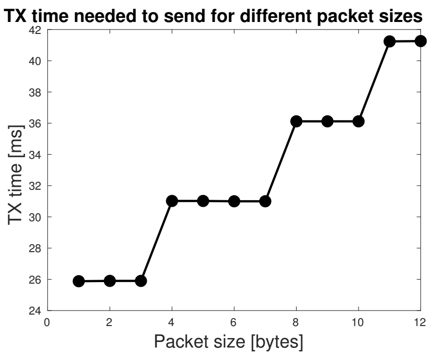

Figure 5.17: Time-on-air for different packet sizes.

LoRa module
The LoRa module offers both a sleep and a standby mode. In standby mode the crystal oscillator is still turned on. In sleep only the SPI bus and configuration registers are available [11]. Table 5.2 shows the current consumption of the LoRa module.

Mode	Specs	Measured
Sleep	0.1 μA	0.21 μA
Standby	1.4 mA	1.69 mA

Table 5.2: Specified current consumption of LoRa module in different modes versus measured values. (The typical value is used here, as presented in [11])

Micro controller
The ATtiny84A offers a few different sleep modes. The most relevant one is the Power-down mode. In this mode all peripherals are turned off and all oscillators are stopped. The micro controller can still be woken up by an interrupt on one of its I/O pins, which would be interesting in the case where the micro controller would be kept in sleep mode during rest.

Furthermore the micro controller has a Power Reduction Register which allows turning off some of the peripherals, such as timers, the USI[1] and the ADC [2] [15].

Table 5.3 shows the current consumption of the micro controller in some different modes.

Mode	Specs	Measured
Power-down	0.13 μA	0.1 μA
Active mode, PRR disabled	-	0.5 mA
Active mode, PRR enabled	0.25 mA [3]	0.5 mA
Idle mode, PRR disabled	-	0.246 mA
Idle mode, PRR enabled	0.04 mA [3]	0.226 mA

Table 5.3: Static current consumption of the micro controller for different modes, specified values are typical values from [15].

Temperature sensor
The TMP-102 also offers a sleep mode [18]. Table 5.4 shows the current consumption of the TMP-102. Clearly the measured current is much smaller than the specified current in normal mode. An explanation for this is that the normal mode is only used when the sensor is performing a measurement. This only takes a few tens of milliseconds. This is too short to get a good measurement using the SMU [4], but 50 μA is also too small to measure using the oscilloscope and shunt resistance. Hence the it seems like the current is only 8.1 μA.

Mode	Specs	Measured
Normal	50 μA	8.1 μA
Sleep	1.5 μA	5.8 μA

Table 5.4: Static current consumption of the temperature sensor for different modes, specified values are typical values from [18].

5.3. Range results

Line of sight range The line of sight range as defined in paragraph 5.1.2 is performed as described. The measurement was done under good weather conditions. While measuring the RSSI the EEMCS building was visible at the horizon at all times. The results of this measurement is plotted in figure 5.18. Each dot represents a received package. The measurement is performed at a fixed transmit power of 17 dBm. The maximum distance at which a packet is received is 14.9 km.

[1] Universal Serial Interface
[2] Analog to Digital Converter
[3] This value was specified with a supply voltage of 2V as opposed to the 3.3V we used, so it is not directly comparable.
[4] Source Measurement Unit

Figure 5.18: Received packets during line of sight measurement.

Inside building range The inside building measurement as defined in paragraph 5.1.2 is performed as described. The measurement results can be found in figure 5.19. The peak at 15 floors is outstanding. The reason for this peak is not investigated in this project. Some speculations are that it might be caused by reflections of the neighbouring building (Faculty of Civil Engineering). The roof of this building is at equal height of the measured peak. Another speculation is that the transmitted signal is guided thought the EEMCS building's elevator shafts.

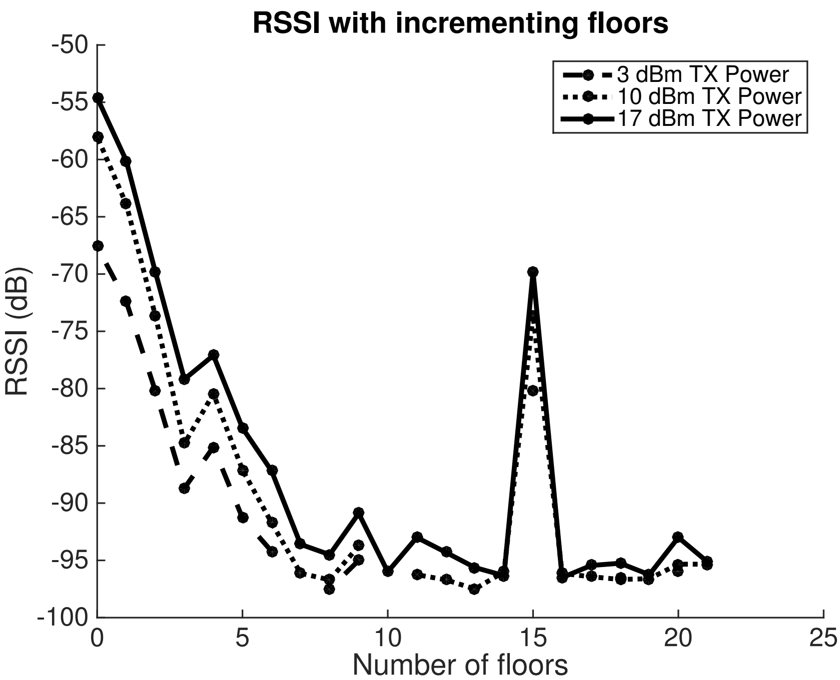

Figure 5.19: Received RSSI with incrementing number of concrete floors on different TX levels.

6
Discussion

6.1. Receiving network

The LoRa module implemented in the IoT sensor is capable of sending LoRa packets to a local receiving gateway and to a service provided network like KPN. When using the KPN network, the LoRa packets have to comply to the LoRaWAN standard [23]. Implementing this standard includes extra calculations to be made by the micro processor like encrypting and preparing a LoRaWAN header. A LoRaWAN header is also longer in size and will have a longer on-air time while transmitting. Since the application of the IoT sensor is in office environments and since office environments in general do not move, implementing the service provided network is not necessary. Due to this reason the IoT sensor will be using local LoRa gateways to receive the sensor's information.

This local LoRa gateway can easily be connected to the internet, so that the IoT sensor is indirectly connected to the cloud.

6.2. Receiving LoRa gateway placement

Since the IoT sensor will be using local LoRa gateways, every office environment will be equipped with at least one LoRa gateway. Looking at the measurement results of measurement 5.3, the IoT sensor will be able to penetrate at least 5 concrete floors when transmitting at 3 dBm. This means that a receiving LoRa gateway is needed every 5 floors. In order to decrease the amount of concrete floors that have to be penetrated, an option can be to place the LoRa gateway outside the measurement environment (for example across the street). This will increase the transmission distance. From the results of measurement 5.3, one can state that in a range of 500 meters outside of the building, the system is able to communicate without packet loss.

6.3. LoRa configuration

The LoRa module is configured using a set of parameters. These parameters are chosen in such a way that a long range is achieved, but there is also some data integrity checking to ensure non-corrupted packets.

6.4. Power consumption

By carefully choosing the components to use, we were able to get an energy consumption as little as ~4mJ for one measurement cycle. Although this is a little bit more than the desired 1.2mJ as described in 2.2.1, it is definitely in the same order of magnitude.

We have looked at different communication protocols and decided to use LoRaWAN, which seems a very good choice. It performs extremely well in terms of range and reliability using only a very tiny amount of energy. We have looked into the different parameters and in combination with existing literature determined which parameters to use, so we have good coverage with limited power consumption.

6.4.1. Sleeping components

Based on the measurements as discussed in 5.2.3, it is not possible to keep the components in sleep during rest. Even keeping just the micro controller in sleep is not an option. The micro controller in power down mode draws about 0.1 μA, which at 3.3V means a power consumption of about $0.1 \cdot 10^{-6} \cdot 3.3 = 0.33$ μW. In the best case, an input power of 1 μW can be reached, in which case just the micro controller would consume about one third of the power, so that only two third of the power is used to charge the capacitor. This will severely slow down the charging process and thereby decrease the rate at which sensing cycles can be performed. In case less power is available at the input, the micro controller might prevent the capacitor from charging altogether.

Not being able to keep the micro controller in sleep mode has a few drawbacks. First of all it is needed to initialize all the components again for each measurement. Secondly, it is not possible to use the micro controller to sense when the capacitor is sufficiently charged, or even to use an interrupt for the micro controller to trigger the start of a sensing cycle. A fully autonomous circuit has to be used to turn on the whole subsystem. Fortunately this has proven to be possible. The DC-DC booster that is used in the conversion and storage subsystem offers a feature that drives an output pin high whenever the capacitor is charged beyond a specified threshold and pulls it low when the capacitor voltage has dropped below another threshold [20].

In hindsight it might have been better to use the PIC as discussed in 3.3. Its sleep current is significantly smaller than that of the used ATtiny. By using the PIC it might have been possible to keep the micro controller turned on permanently instead of having to turn it off. This is a recommendation for any further research on this topic.

6.4.2. LoRa parameters

The LoRa parameters that were used are mainly based on literature. These parameters proved to meet the requirements in terms of range, but also in terms of power consumption. As discussed in 5.2.2, the parameters that we chose show good behavior in terms of power consumption. Hence there is no reason to choose them differently.

6.5. Sensing

The sensor is connected using I^2C , which is a standardized protocol used by many different types of sensors. This way it is easy to swap the sensor for any other sensor. This is useful in smart buildings where might want to measure temperature, humidity, light intensity and more. All can be done with this IoT sensor.

The used temperature sensor is a little less precise than the specified 0.1 °C, because it has an uncertainty of $\sigma = 0.2$ °C. It is very close to the specifications however. If a higher precision is desired, another sensor could be chosen.

6.6. Geometry

The device can be implemented in a relatively small fashion. The PCB that was designed during this project is about 80mm x 50mm. This size could be even further reduced by improving the PCB design and using just the SX1272 chip without its break-out board. This size is well within the specified dimensions of 200mm x 200mm x 100mm.

6.7. Environment

The IoT sensor is protected against overvoltage and undervoltage by the conversion subsystem [19]. By placing the sensor inside a case, it is protected from bigger items, thus complying with the specified dust resistance.

7
Conclusion

The objective of this project was to investigate the possibilities for an extreme low power implementation of the sensing, communication and processing subsystem of an RF powered IoT smart sensor. Considering the specifications and reviewing the measurements done, the designed prototype of the subsystem complies with the given specifications.

We have shown that it is possible to build a system that performs temperature measurements using as little as 4mJ of energy. At this energy level it is still possible to reach a reasonable range and reliability. We have seen that the choice of micro controller might not have been optimal, so that the energy consumption might be decreased slightly more.

The LoRaWAN standard implements a form of automatic data rate optimization (ADR). This ADR optimization is allows the LoRa parameters to be optimized for the current transmission distance, increasing the module's power efficiency. This ADR could be implemented in the peer to peer connection used in the subsystem to reduce the power consummation even more. This is something that could be futher looked into in future research.

The entire IoT sensor, including the harvesting subsystem [30] and the conversion and storage subsystem [19] has been built on a PCB. Some tests have been performed on the total system. It seems that the harvested power is very little when no RF sources are nearby. It does show however, that it is feasible to further develop this technique and achieve a useful sensor. More extensive testing of the entire IoT sensor and improving the harvested power output are also subject for future work.

Bibliography

[1] T. Sun, X. Xie, and Z. Wang, *Wireless Power Transfer for Medical Microsystems.* Springer Science & Business Media, 2013.

[2] S. Ashley, "Wireless charging: Pulling the plug on electric cars." `http://www.bbc.com/future/story/20121120-pulling-the-plug-on-electric-cars`, 2014.

[3] N. Tesla, "Experiments with alternate currents of very high frequency and their application to methods of artificiel illumination," *The Inventions, Researches and Writings of Nikola Tesla*, 1891.

[4] NOWI-energy, "Nowi: Enabling the internet of things." `https://www.nowi-energy.com/`, 2018.

[5] Drayson Technologies and LTD. Incorporated, "Freevolt rf energy harvesting - for the low energy internet of things." `http://www.getfreevolt.com`, 2018.

[6] Powercast Corp., "Powercast - wireless power products." `http://www.powercastco.com/`, 2018.

[7] D. Snoonian, "Smart buildings," *IEEE Spectrum*, 2003.

[8] Beghelli Canada, "Iec 60529-1 & nema standards." `https://www.brenclosures.com.au/wp-content/uploads/BR-Enclosures_IP-Ratings-Chart.pdf`, 2003.

[9] B. Reynders, W. Meert, and S. Pollin, "Range and coexistence analysis of long range unlicensed communication," in *2016 23rd International Conference on Telecommunications (ICT)*, pp. 1–6, May 2016.

[10] LoRa Alliance, "Lorawan 1.1 regional parameters." `https://lora-alliance.org/sites/default/files/2018-04/lorawantm_regional_parameters_v1.1rb_-_final.pdf`, 2018.

[11] Semtech, "Sx1272/73 datasheet." `https://www.semtech.com/uploads/documents/sx1272.pdf`, 2018.

[12] HopeRF, "Rfm95/96/97/98(w) datasheet." `http://www.hoperf.com/upload/rf/RFM95_96_97_98W.pdf`, 2018.

[13] Microchip, "Low power mcu and mpu family overview." `http://www.microchip.com/design-centers/lowpower`.

[14] Microchip, "Pic12f1822 datasheet." `http://ww1.microchip.com/downloads/en/DeviceDoc/40001413E.pdf`.

[15] Atmel, "Attiny24a/44a/84a datasheet." `http://ww1.microchip.com/downloads/en/DeviceDoc/doc8183.pdf`.

[16] STMicroelectronics, "Stm32 ultra low power mcus." `http://www.st.com/en/microcontrollers/stm32-ultra-low-power-mcus.html?querycriteria=productId=SC2157`.

[17] NXP, "Um10204, i2c-bus specification and user manual." `https://www.nxp.com/docs/en/user-guide/UM10204.pdf`, 2014.

[18] Texas Instruments, "Tmp102 low-power digital temperature sensor with smbus and two-wire serial interface in sot563." `http://www.ti.com/lit/ds/symlink/tmp102.pdf`, 2015.

[19] D. Bosma and W. Lindeman, "Autonomous battery less sensor for iot applications in smart buildings: Low-power energy conversion and storage for rf energy harvesting," tech. rep., University of Technology Delft, 2018.

[20] Texas Instruments, "bq25504 ultra low-power boost converter with battery management for energy harvester applications." http://www.ti.com/general/docs/lit/getliterature.tsp?genericPartNumber=BQ25504&&fileType=pdf, 2015.

[21] Schwerdtfeger, "Spi - serial peripheral interface." https://www.mct.net/faq/spi.html, 2000.

[22] Texas Instruments, "I2c bus pullup resistor calculation." http://www.ti.com/lit/an/slva689/slva689.pdf, 2005.

[23] LoRa Alliance, "Lora specification." https://lora-alliance.org/sites/default/files/2018-04/lorawantm_specification_-v1.1.pdf, 2015.

[24] M. Knight and B. Seeber, "Decoding lora: Realizing a modern lpwan with sdr." https://pubs.gnuradio.org/index.php/grcon/article/download/8/7, 2016.

[25] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melià-Seguí, and T. Watteyne, "Understanding the limits of lorawan." https://arxiv.org/pdf/1607.08011.pdf, 2017.

[26] Semtech, "Lora design guide." https://www.semtech.com/uploads/documents/LoraDesignGuide_STD.pdf, 2013.

[27] Arduino, "Arduino as isp and arduino bootloaders." https://www.arduino.cc/en/Tutorial/ArduinoISP.

[28] S. Mistry, "Arduino lora." https://github.com/sandeepmistry/arduino-LoRa.

[29] Keithley, "Model 2400 series sourcemeter user's manual." http://research.physics.illinois.edu/bezryadin/labprotocol/Keithley2400Manual.pdf.

[30] B. Meekes and V. Waegenaere, "Autonomous battery less sensor for iot applications in smart buildings: Ambient rf energy harvesting in office environment for iot sensor," tech. rep., University of Technology Delft, 2018.