Sample Implementation of Internet of Things in Small and Medium Enterprises

A thesis submitted for the

Bachelor of Science in Information Systems

by

Thomas Willems

Student ID: 210200426

E-Mail: twillems@uni-koblenz.de

Faculty 4: Computer Science

Institute for IS Research

University of Koblenz-Landau, Germany

Supervisors:

Prof. Dr. Susan P. Williams

Patrick Nitschke

Koblenz, October 2018

Declaration/Erklärung

I declare that,

This thesis presents work carried out by myself and does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any university. To the best of my knowledge, it does not constitute any previous work published or written by another person except where due reference is made in the text.

Ich versichere,

dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Mit der Einstellung dieser Arbeit in die Bibliothek bin ich einverstanden. Der Veröffentlichung dieser Arbeit im Internet stimme ich zu.

Thomas Willems

Koblenz, 16.10.2018

Abstract (English)

The Internet of Things (IoT) is a concept in which connected physical objects are integrated into the virtual world to become active partakers of businesses and everyday processes (Uckelmann, Harrison and Michahelles, 2011; Shrouf, Ordieres and Miragliotta, 2014). It is expected to have a major impact on businesses (Council, Nic and Intelligence, 2008), but small and medium enterprises' business models are threatened if they do not adopt the new concept (Sommer, 2015). Thus, this thesis aims to showcase a sample implementation of connected devices in a small enterprise, demonstrating its added benefits for the business.

Design Science Research (DSR) is used to develop a prototype based on a use case provided by a carpentry. The prototype comprises a hardware sensor and a web application which can be used by the wood shop to improve their processes. The thesis documents the iterative process of developing a prototype from the grounds up to useable hard- and software.

This contribution provides an example of how IoT can be used and implemented at a small business.

Abstract (German)

Das Internet der Dinge (IoT) ist ein Konzept, bestehend aus vernetzten physischen Objekten, welche in die virtuelle Welt integriert werden um aktive Teilnehmer von Geschäfts- und Alltagsprozessen zu werden (Uckelmann, Harrison and Michahelles, 2011; Shrouf, Ordieres and Miragliotta, 2014). Es wird erwartet, dass dieses Konzept einen großen Einfluss auf Unternehmen haben wird (Council, Nic and Intelligence, 2008). Geschäftsmodelle kleiner und mittelständischer Unternehmen (KMU) sind bedroht, sollten sie den sich abzeichnenden Trend nutzen (Sommer, 2015). Daher ist das Ziel dieser Arbeit, eine exemplarische Implementierung von vernetzten Geräten in einem kleinen Unternehmen um seine Vorteile darzustellen.

Diese Arbeit verwendet Design Science Research (DSR) um einen Prototyp zu entwickeln, der auf dem Anwendungsfall einer Holzwerkstatt aufbaut. Der Prototyp besteht aus einem physischen Sensor und einer Webapplikation, welche von dem kleinen Unternehmen zur Verbesserung seiner Prozesse genutzt werden kann. Die Arbeit dokumentiert den iterativen Entwicklungsprozess der Prototypen von Grund auf zu nutzbarer Hard- und Software.

Der Hauptbeitrag dieser Arbeit ist die beispielhafte Anwendung und Nutzung von IoT in einem kleinen Unternehmen.

Table of Contents

List of Abbreviations

API	Application Programming Interface
BLE	Bluetooth Low Energy
DSR	Design Science Research
EEPROM	Electrically erasable programmable memory
GDC	General Design Cycle
IoT	Internet of Things
LiPo	Lithium Polymer Battery
LoRa	Long Range
LoRaWAN	Long Range Wide Area Network
MCU	Microcontroller
NiMH	Nickel-metal hydride Battery
RFID	Radio frequency identification
RPiZW	Raspberry Pi Zero W
TTN	The Things Network
SME	Small and Medium Enterprise
SoC	System on a Chip
UUID	Universally Unique Identifiers

List of Figures

List of Tables

1 Introduction

This first chapter of this thesis gives a brief introduction to the to the topic “Sample Implementation of Internet of Things in Small and Medium Enterprises”. Starting with the problem statement which initiated the performed research (section 1.1). Based on this statement the research aim, objectives and questions are depicted (section 1.2). These are fulfilled by using the design science research method, which are introduced in the research method section (section 1.3). Further an outline of the following chapters is given (section 1.4).

1.1 Problem Statement

Internet of Things (IoT) are interconnected, communicating pieces of equipment with optional capabilities like gathering data, actuating or data processing (International Telecommunication Union, 2012).

It is expected that businesses spend 250 billion Euro to IoT related fields by 2020 (Hunke *et al.*, 2017). It is also expected that IoT will impact nearly six of the global economy percent of the global economy (Blanter, 2014). Currently IoT passed the “Peak of Inflated Expectations” in Gartner’s Hype Cycle for 2018(Gartner, 2018). The Internet of Things is seen as a big driver for Industry 4.0 (Shrouf, Ordieres and Miragliotta, 2014). Because of this high expectations large Enterprises are already investing in infrastructure (Barr, 2015; White, 2018), processes (Löffler and Tschiesner, 2013) and products(Wortmann and Flüchter, 2015). These companies are at the forefront of the so-called fourth industrial revolution (Schwab, 2016).

Small and Medium Enterprises (SME), the “backbone” of the German economy might be left behind (Söllner, 2014). Though these enterprises are often praised for their role as innovators they often lack specialized employees as well as funding thus innovation investments are stagnating (EFI, 2016). If they do not adopt, even their livelihood could be threatened (Sommer, 2015). To keep their status, SMEs need to consider the new possibilities of connected devices. IoT can be used in many sectors relevant to SMEs like manufacturing & production or maintenance (Roy *et al.*, 2016; Zhang *et al.*, 2016). Asset management (e.g. remote monitoring of tools and resources) and automation can speed up manufacturing and lower the manufacturing costs (Butner and Lubowe, 2015).

To profit from the improvements of IoT, SMEs might consider buying into existing, proprietary technologies and infrastructure. Doing this they risk “data”- or “vendor”- lock in(Roman, Zhou and Lopez, 2013; Mineraud *et al.*, 2016). A possible solution to these problems could be open source software and protocols (Ven, Verelst and Mannaert, 2008).

1.2 Research Aim, Objectives and Questions

The aim of this thesis is to evaluate use cases of IoT for SMEs. The research will focus on sensor-based, connected devices. It should show that there is a low cost, easy-to-use alternative to proprietary solutions. This alternative will consider the “Things” as well as an IoT Platform as a backend. A connected device will be built, which will show its collected data in an easy-to-understand web application.

The research is done with the help of the carpentry “Holzgespür”. Holzgespür is an exemplary digitized business of the Handwerkskammer Koblenz. The small business evaluates the use of sensor-based devices to improve their processes. Holzgespür has valuable domain knowledge in woodworking which helps to determine use cases, measuring methods and restrictions which may apply.

To achieve the research aim, several Research objectives will be fulfilled (see Figure 1). The first research objective (RO1) is to find existing uses of IoT in Enterprises. This might give further ideas how connected devices can be used in a carpentry.

RO1 Research existing approaches of using IoT in enterprises.

To achieve the first research objective, research questions (RQ) are expressed which can serve as guidance. Existing research regarding IoT in enterprises should be evaluated. Success stories of IoT implementations can be utilized to find already used infrastructure and tools

RQ1.1 Which research has been concluded regarding IoT in enterprises?

RQ1.2 Are there successful implementations of IoT in enterprises?

RQ1.3 Which infrastructure and tools have been used?

Further it is important to find use cases for a small enterprise like the carpentry Holzgespür. Since the research will focus on sensor-based devices, useful sensors for a workshop should be found.

RO2 Research possible use cases for a carpentry.

To find these sensors, useful data needs to be identified. In a further step its necessary to collect this useful data. This leads to the following RQ:

RQ2.1 What data could a local carpentry need?

RQ2.2 How can this data be obtained?

To use the collected data meaningfully, the data needs to be stored in a data processing backend. The data can then be evaluated and visualized. Networking and connection types for IoT devices should be evaluated to send the data to a more powerful backend.

RO3 Research networking options for IoT devices.

To preserve energy the networking stack should be low-powered. Also, the data should be sent to an open platform to avoid data or vendor lock in.

RQ3.1 Which low-powered networking stacks can be used for sensor devices?

RQ3.2 Are there existent platforms to gather the data?

RQ3.3 How should a backend be designed to process gathered data?

After the above ROs are fulfilled, a working prototype and backend should be developed. The prototype should gather data and send it to a backend.

RO4 Develop a sensor-based prototype for a local small enterprise.

Components and restrictions need to be considered building a working prototype. This leads to the following questions.

RQ4.1 Which components are suitable for a prototype?

RQ4.2 Are there restrictions for devices in a carpentry?

The collected data needs to be accessible for further use in the carpentry.

RO5 Make the sensor data accessible.

For the scope of this thesis the data should be visualized meaningfully.

RQ5.1 How to visualize data?

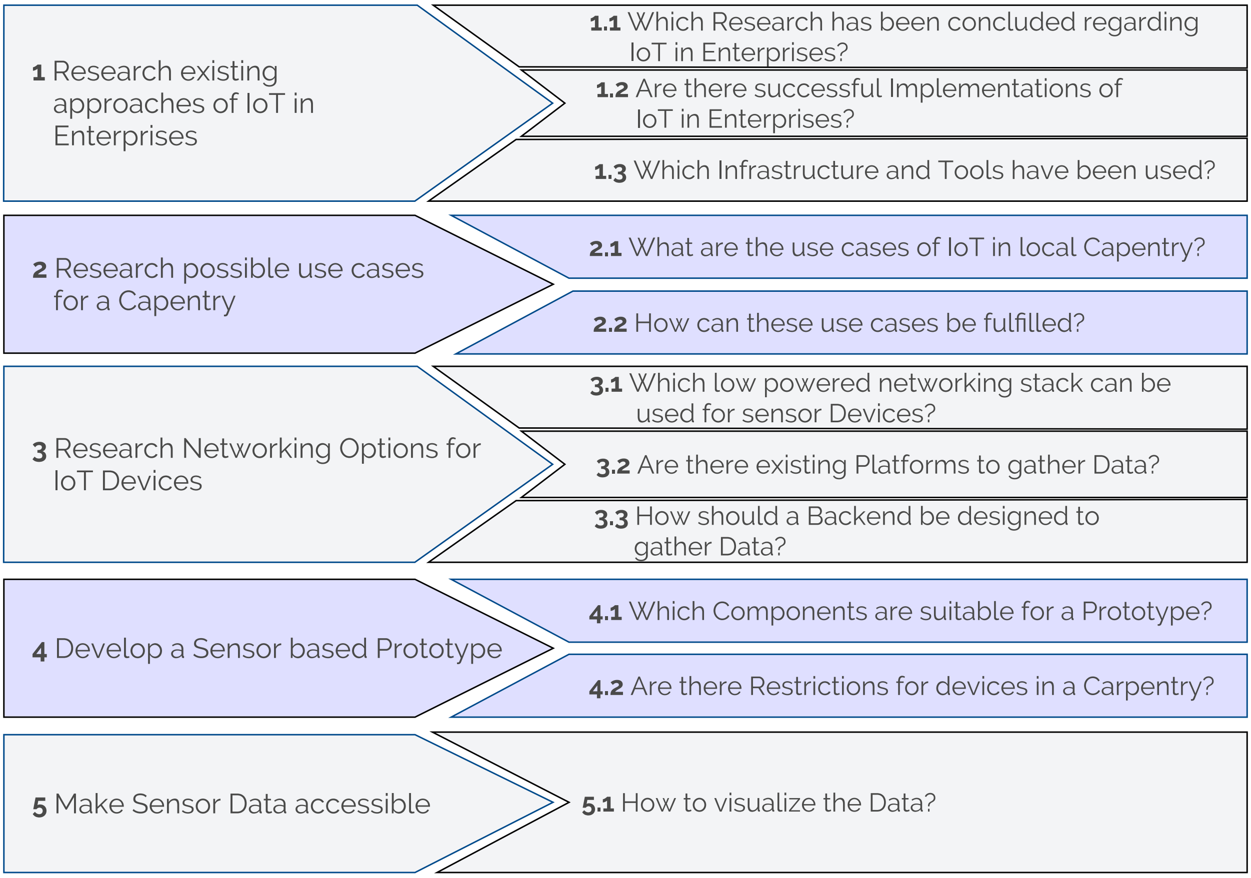


Figure 1: Research objectives and corresponding research questions (own illustration)

1.3 Research Design

To develop and evaluate a prototype for a small business, Design Science Research (DSR) is used as the research methodology. Considering that IoT is a relatively new field, DSR is deemed useful. It provides a framework to develop an artifact – the prototype – based on the researcher's general knowledge to identify ways to solve a problem. Further the General Design cycle (GDC) by Takeda *et al.* (1990) and Vaishnavi and Kuechler (2008) is used to structure the research process. This structure comprises five steps, which can be cyclical (see Figure 2).

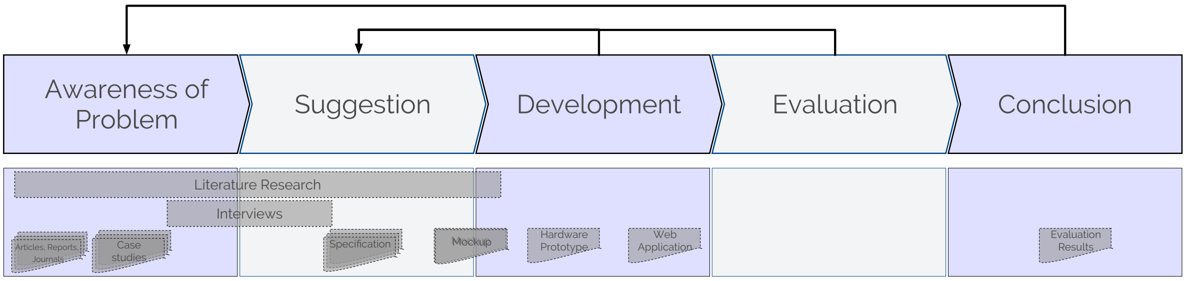


Figure 2: Design Science Research Framework (based on Vaishnavy and Küchler, 2007)

The GDC starts with the awareness of problem, in which a problem is identified. The problem can be identified through literature research or can be deduced from a previous cycle. For this thesis, the problem was identified through literature research and interviews with employees of the carpentry.

Having a general awareness of the problem, suggestions are drawn. These suggestions are based on the interviews and by reading specifications of tools used. Methods and appropriate hardware to solve the problems are researched. During the suggestion step, first mockups are created to see the general viability of the proposed approach.

During the third step – the development – the identified suggestions are used to create prototypes. For this thesis, the prototypes are a sensor-based device and a web application. The sensor-based device measures the wood moisture content and determines its location. The web application visualizes the data from the device. During development, possible occurring problems or improvements can lead to a next iteration starting at a previous step.

The developed prototypes are tested in the evaluation step to see if the stated problems can be solved. Failure to do so, can lead to a further iteration of the GDC. The following conclusion is the last step of a DSR project.

The conclusion ends with an artifact which could be the awareness of a new problem, leading to a new cycle.

During the first two steps of the process, literature research was conducted to become familiar with the domain of IoT. For the literature research the databases ACM Digital Library[1], Springerlink[2],

IEE Xplore Digital Library[3], ResearchGate[4] and Google Scholar[5] are used. The Focus was based on IoT and the usage of IoT in Enterprises. Lacking scientific papers for the later, the focus was widened to “Industry 4.0”, or “Fourth Industrial Revolution”, to see how emerging technologies are used to improve processes and how these can be used for SMEs.

1.4 Outline of the Thesis

This section summarizes the following chapters and describes each of them in the following paragraphs.

Chapter 1 gives a brief introduction to the topic. A Problem statement and the research aim are specified, guiding the further research conducted in this thesis.

Chapter 2 defines the theoretical foundations on which this thesis is based on. Section 2.1 introduces Internet of Things, section 2.2 describes the use of IoT in enterprises.

Chapter 3 documents the development of the prototype. The use case for the sensor-based device is described in a scenario (section 3.1). Section 3.2 introduces the chosen sensors. Methods for moisture measuring and the respective hardware are presented (section 3.2.1) The other sensors – a positioning system – functionality is described in section 3.2.2. Further the used networking stack (section 3.3), and additional used components (section 3.4) are depicted. Chapter 3 also examines the case to protect all components (section 3.5). The design of the software, especially the web application is presented (section 3.6).

Chapter 4 shows the setup and testing of the prototype. The build process of the hardware (section 4.1.1) and the deployment of the web application (section 4.1.2) is described. Further, each sensor and the web application are tested in regard to the use cases. (section 4.2).

Chapter 5 is the final chapter of this thesis, providing summarized answers to the research questions (section 5.1). It further discusses the limitations of the research conducted (section 5.2). Possibilities for future work based on this thesis are showcased (section 5.3).

---

[1] http://dl.acm.org/

[2] http://link.springer.com/

[3] http://ieeexplore.ieee.org/

[4] https://www.researchgate.net/

[5] https://scholar.google.de/

2 Theoretical Foundations

This chapter gives a definition of Internet of Things, the historic development and an overview of different perspectives (section 2.1). The usage of IoT in businesses and how it correlates with Industry 4.0 is also assessed (section 2.2).

2.1 Internet of Things

There is no unique definition of Internet of Things. The Term was first coined by Kevin Ashton in 1999, by linking the use of RFID technology with the internet (Ashton, 2010). He described how the supply chain can be improved by connecting physical things to networked machines to share and store information. Today most definitions have the commonality of physical things, accessible through computers and networked devices (ITU, 2005; Uckelmann, Harrison and Michahelles, 2011; Madakam, Ramaswamy and Tripathi, 2015).

Internet of things is seen to have a high impact in the coming years and centuries. The term IoT itself, or derived technologies and models (i.e. IoT Platform) are continuously on the Gartner Hype Cycle (Gartner, 2013, 2014, 2016, 2017, 2018) as one of the emerging technologies. IoT Platforms – a middleware for machine-to-machine or application communication (Mineraud *et al.*, 2016) – passed the peak of inflated expectations and is supposed to reach its plateau in five to ten years (Gartner, 2018). It is expected that by 2020, 5 to 50 billion devices will be connected to the internet (Evans, 2011; Gubbi *et al.*, 2013). Connected devices are deemed to have a major impact on both businesses and private life (Council, Nic and Intelligence, 2008; Barnaghi *et al.*, 2012). Business spending is projected to reach 250 billion euros by 2020 (Hunke *et al.*, 2017).

Atzori, Iera and Morabito (2010) divide the Internet of Things into three visions: the "Things"-oriented, the "Internet"-oriented and the "Semantic"-oriented vision (Figure 3).

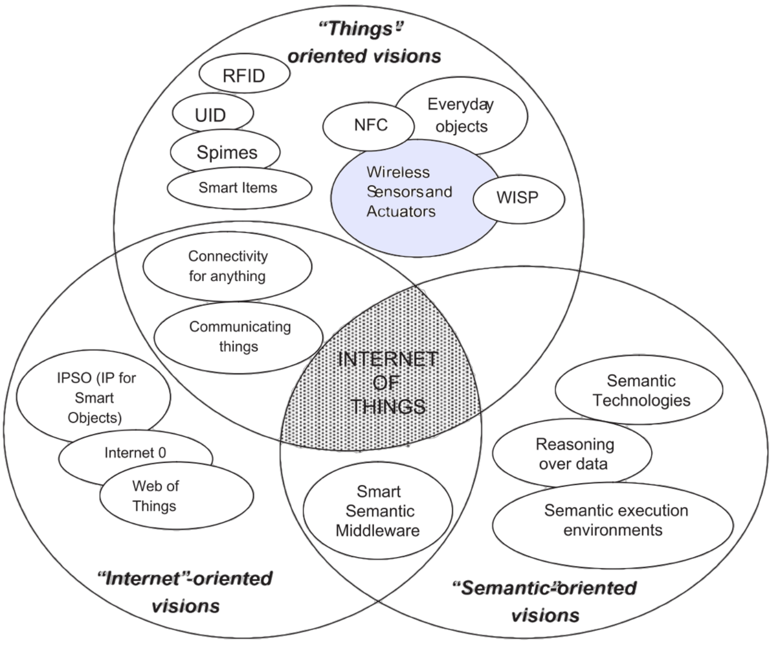


Figure 3: "Internet of Things" paradigm (from Atzori, Iera and Morabito, 2010)

The thesis will focus on the "Things"-oriented vision of the above Internet of Things model (Atzori, Iera and Morabito, 2010). These things are connected devices which are trackable and uniquely identifiable. A vision is the integration and combination of the physical world and the digital world (Fleisch, 2010). Things, which can generate data with sensors or execute actions through actuators, can capture or interact with the physical world. The physical world becomes more accessible through networked devices like computers. Things can further communicate with themselves and other applications (machine-to-machine communication) to achieve common goals (Zuehlke, 2010). The International Telecommunications Union sees the potential, that connections are not only between anyone at any time, but also by anything. Through the accessibility, connected devices can enhance business process and everyday scenarios (Uckelmann, Harrison and Michahelles, 2011).

The internet-oriented vision describes how Things can be incorporated into the current Internet, or how the current Internet can be enhanced to support billions of connected devices. The researchers propose to simplify the IP-stack to accommodate the limited power of connected devices (Atzori, Iera and Morabito, 2010).

The semantic-oriented vision discusses ways of describing new services and things to harmonize the communication (Atzori, Iera and Morabito, 2010). These semantics are supposed to enable interoperability of connected devices and their functionality, streamlining the heterogenous IoT environment.

The built prototype is a wireless sensor device which communicates with a web application. Sensor data is made accessible for further use by the application. The prototype uses already developed technologies (i.e. LoRaWAN), and semantics (i.e. MQTT) to enable the communication between the device and the web application. Thus, the thesis focuses on the things-oriented vision, more specifically on the build process of a wireless sensors.

2.2 Fourth Industrial Revolution

As stated above, IoT is expected to have a major impact on businesses (Council, Nic and Intelligence, 2008; Barnaghi *et al.*, 2012). Businesses can use IoT to create new, or improve their existing processes with the help of connected devices (Shrouf, Ordieres and Miragliotta, 2014). Additionally, whole new business models are opening up with the use of evolving technology (Leminen *et al.*, 2012). The use and impact are labeled the "Fourth Industrial Revolution", or "Industry 4.0" (Bloem *et al.*, 2014; Lasi *et al.*, 2014; Schwab, 2016).

With the advance of IoT, connected physical objects can be integrated into business processes (Shrouf, Ordieres and Miragliotta, 2014). Changes detected by things can alter processes to improve performance (Zuehlke, 2010). Connected devices can even become active participants of based processes (Shrouf, Ordieres and Miragliotta, 2014). One such example could be the production line detecting the product being worked on. Based on the gathered data, the product could be routed to different stations. Connected devices might even act independently to achieve business goals (Zuehlke, 2010). The production line could autonomously route a product based on sensor data. In a wood working shop, wood logs can notify an employee as soon as a required wood moisture is reached.

This automation can change the production process significantly (World Economic Forum, 2018). Manual labor could be reduced and process could be sped up (Bauernhansl, Ten Hompel and Vogel-Heuser, 2014). Because of these benefits and potential cost savings, large enterprises are already investing to improve their processes (Löffler and Tschiesner, 2013; Huber, 2016; Roth, 2016).

Besides the improvements to processes, new business models can be established using connected devices (Yang, Liu and Liang, 2010; Leminen *et al.*, 2012). For example, connected cars are used to create a floating car sharing fleet, in which cars can be tracked and rented through the internet (Leminen *et al.*, 2012). Another example is predictive maintenance, in which products can foresee upcoming failures and request maintenance (Huber and Kaiser, 2015). In a wood working shop, predictive maintenance could be furniture, detecting damaged or missing coating and alerting the owner.

Although the fourth industrial revolution promises several benefits for businesses, SMEs are not investing heavily in these modern technologies. The stagnating investments are on account of missing specialized employees or funding (EFI, 2016). Because of this, SMEs are jeopardizing their livelihood and might become victims of the Fourth Industrial Revolution (Sommer, 2015).

3 Prototype Development

This chapter covers the development of the prototype. A scenario for IoT in a carpentry is illustrated (section 3.1). Based on this setup, the development is described. Beginning with the used sensors (section 3.2) over the chosen network protocol (section 3.3), to building the actual hardware (section 3.5). Further the programming of the prototype and the web application are illustrated (section 3.6).

3.1 Scenario

In a traditional trade as wood working, several tasks are done manually by an employee. The prototype should support the staff in their day-to-day work. During the initial interview one of these manual tasks got described in detail: Wood needs to have a specific moisture content to be processed and installed. An example was, a stair case which is supposed to be installed in a historic castle. The wood moisture of the staircase needs to match the humidity in the castle to avoid a shrinking or expanding of the wood after its installed. As an organic and hygroscopic[6] material, wood can change its size depending on the amount of water saved within. A change of its size could lead to damage of the product.

To guarantee the wood will not change its size after its processed, the moisture content needs to be monitored regularly. This measurement is currently done manually. Smart sensors could be used to reduce the manual labor by monitoring the moisture of a wood log.

To control the drying rate of wood, the logs are moved frequently through, or outside the shop. A warmer, more sunny area leads to faster drying. Currently there is no system to keep track of the wood logs location. Only the employee who moved the logs knows the current location. IoT can help to trace the movement of wood logs through the shop. Storing the location data and making it accessible, gives all employees the chance to find the wood type they are looking for.

Since the wood is moved through the shop and even to outside locations, the device needs to be able to send data over a wide area of several hundred square meter. Having a wide network coverage helps to keep accurate and up-to-date data available for look up by employees.

To use the collected moisture and location data, it should be presented in a quick and easily understandable way. Employees should be able to access the data on the go, to decrease the lookup time. To display the data, it needs to be sent to a web application. This application will display the gathered data. This makes it easier for employees to find logs they can work on. Additionally, the application is supposed to send out notifications if a wood log reaches a certain moisture, or is moved to a different location.

---

[6] Absorbing or attracting moisture from the air

A protective case needs to be built to ensure the sensors and components of the hardware prototype do not get damaged in the rough environment of a carpentry. This case needs to be small enough to be deployed throughout the shop without interrupting the daily business. It should also withstand rough handling and even drops.

Building the web application and hardware prototype is done based on the design science research method (see chapter 1.3). Each section can be mapped to the five steps (Figure 4). After literature research and an interview, use cases are expressed. Based on these and the acquired knowledge, a tentative design is proposed. This design is the starting point for the web application and hardware prototype. Developing the software and hardware is an iterative process and may be repeated. The final prototypes are tested based on their functionality.

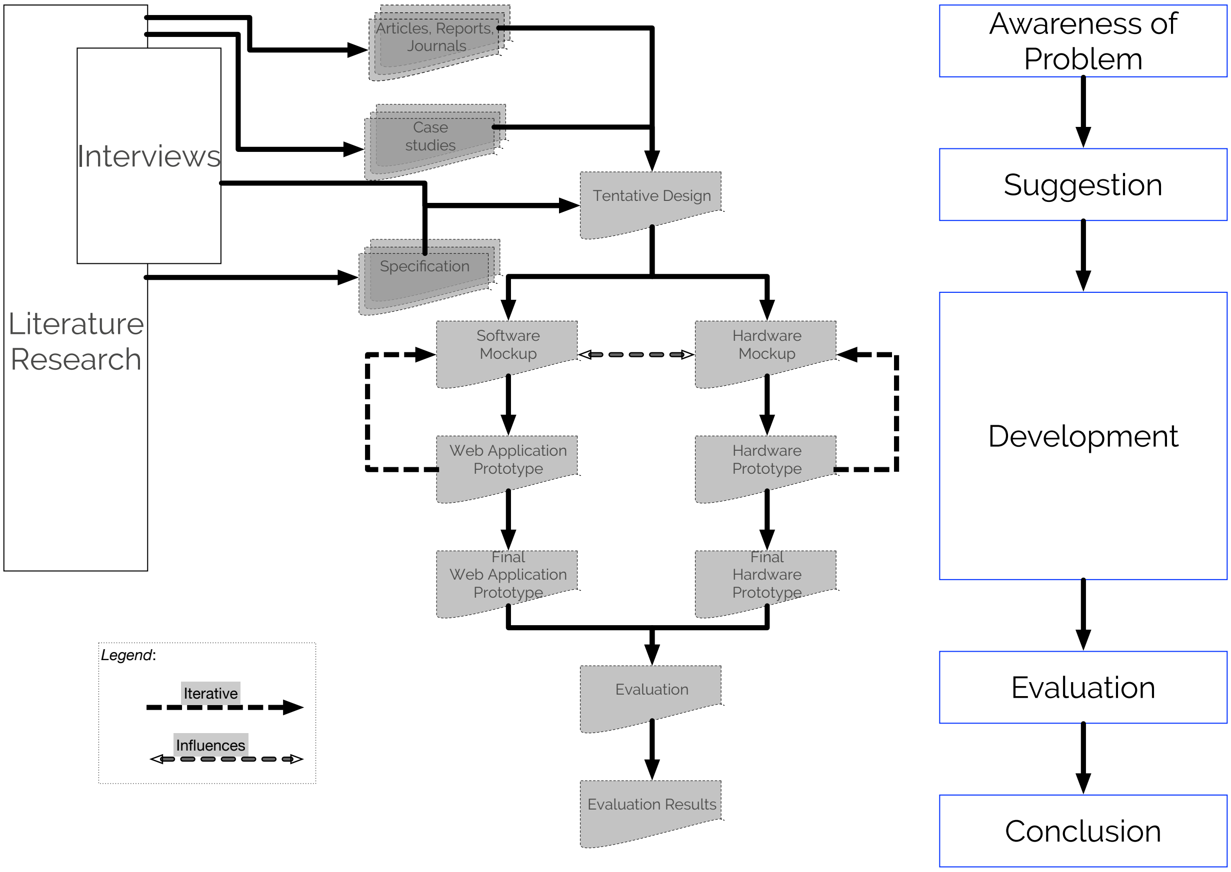


Figure 4: Research Steps (based on Vaishnavi and Kuechler, 2008)

3.2 Sensors

To fulfill the use cases – measuring wood moisture and determining location of wood logs – two sensors need to be designed. A wood moisture sensor to monitor the moisture content (section 3.2.1) and a positioning system to determine the location of wood logs (section 3.2.2). Finding suitable methods and hardware to achieve both uses cases is described below.

3.2.1 Wood Moisture

Wood as an active material, can shrink or expand depending on its moisture level. As mentioned in the Scenario, the wood worker needs to keep track of the wood's moisture content and match the humidity of the installation location. This ensures a finished product, e.g. a staircase, will not change its size and possibly break. Currently these measurements are done manually and regularly, at the shop and remote install locations. Only if the moisture content is adequate (e.g. equal to the remote location), the wood can be processed.

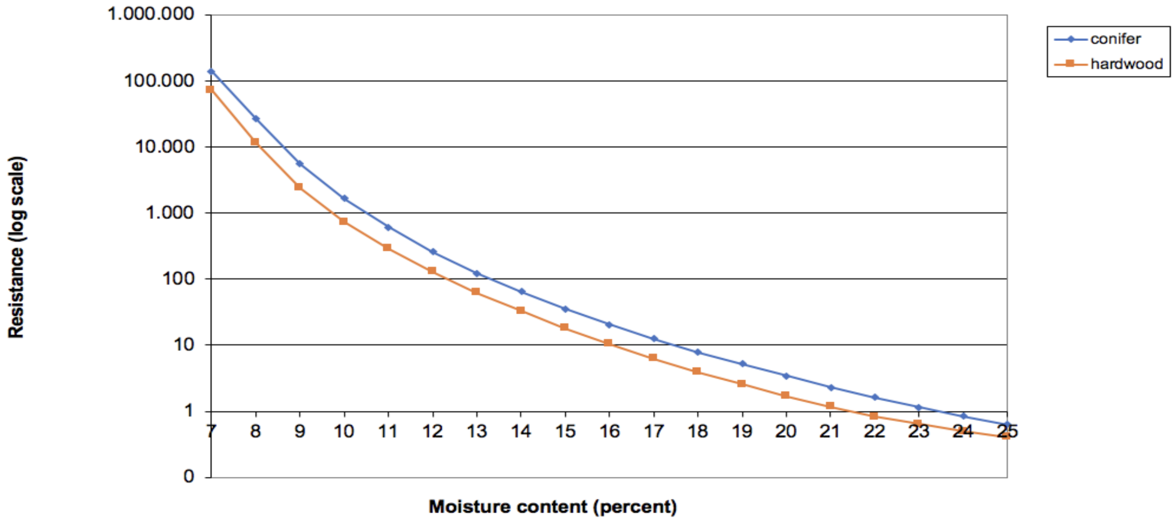


Figure 5: Mapping Moisture Content to Resistance (average) (based on James, 1988)

Wood moisture can be determined by measuring the wood resistance after DIN EN 13183-2. A lower resistance represents a higher moisture content. This mapping is a logarithmic function: a wood moisture of greater than seventeen percent typically has a resistance of less than ten megaohm; wood moisture of less than ten percent typically has a resistance greater than 1.000 megaohm (see Figure 5). Wood needs to have between thirteen and seven percent water content to be processed (Niemz *et al.*, 2007). This corresponds to between about 100 to 100.000 megaohm (James, 1988). The exact wood type needs to be considered getting a mostly accurate mapping of resistance and wood moisture. Conifers have a higher resistance at a lower wood moisture than hardwoods (compare Figure 6 & Figure 7).

The analog input of the prototyping hardware is to imprecise to determine the wood moisture of a specific wood type (see section 3.2.1.2**)**. Due to this limitation, a distinction is drawn only between conifer and hardwoods for this thesis.

It should be noted that the used values are based on measurements by the United States Department of Agriculture and only reflect American wood types (James, 1988). These measurements can be used to determine the moisture content of European hardwoods and conifers at around 1% accuracy, as reported by sample measurements at the carpentry (see Table 7). To accurately determine the moisture of specific wood types, measurements with a calibrated tool need to be recorded per type.

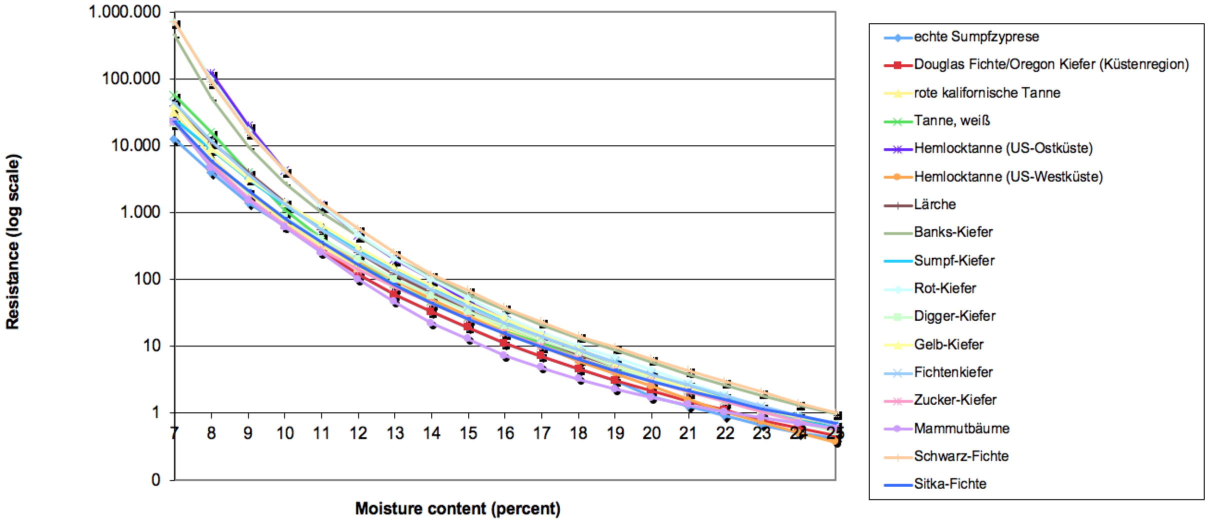


Figure 6: Mapping Moisture Content to Resistance (conifers) (based on James, 1988)

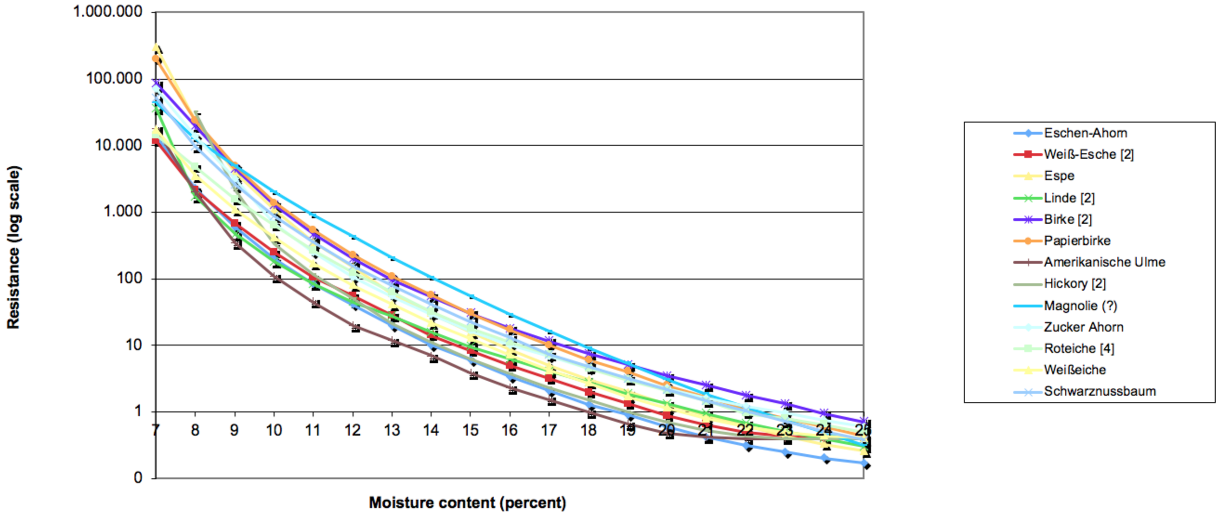


Figure 7: Mapping Moisture Content to Resistance (hardwoods) (based on James, 1988)

A sensor measuring the wood moisture can reduce the manual labor at the carpentry. Although the prototype will only differentiate between hardwoods and conifers, the measurements are accurate enough to be used meaningfully, according to employees. As described in the scenario (section 3.1), the staff should also be notified if the moisture content reaches a desired amount.

Since the processing power and storage is limited on the prototypes, they will only measure the resistance. The mapping of resistance to a moisture content and notifications is done on a more powerful application (section 3.6.3).

3.2.1.1 Wood Moisture Measuring Method

A way to measure the resistance, and thus the moisture content, of wood is a voltage divider as described in DIN EN 13183-2. Voltage dividers return a resistance R by multiplying a known resistance R_{known} with a provided voltage $V_{provided}$ divided by V_{read} and subtracting one:

$$R_{wood} = R_{known}(\frac{V_{provided}}{V_{read}} - 1)$$

This formula can be proved by Ohm's law. The voltage divider is illustrated by the below circuit diagram.

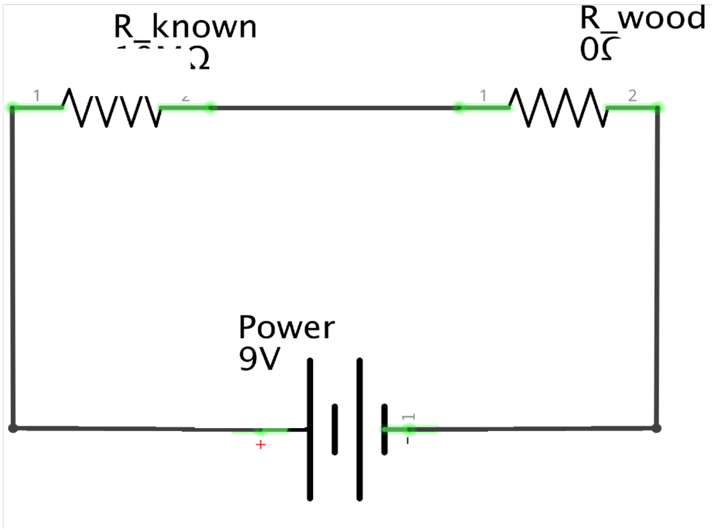


Figure 8: Circuit Diagram - Voltage Divider (own illustration)

3.2.1.2 Wood Moisture Measuring Hardware

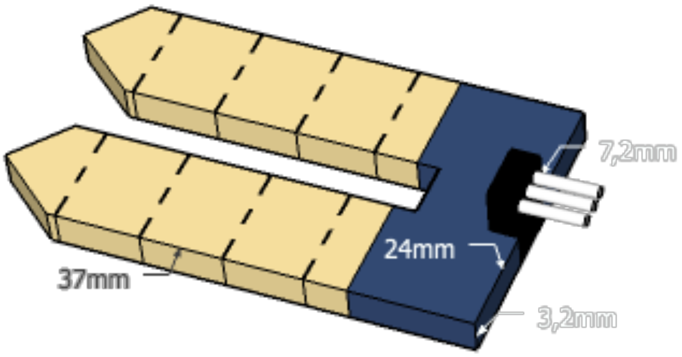


Figure 9: Example of a Prebuilt Soil Moisture Measurement Part (own illustration)

As a first approach, to measure the moisture content, prebuilt parts are used. Prebuilt parts reduce the build time for early prototype devices. There are several "soil moisture" sensors available (Figure 9). These sensors are typically built using a LM393 comparator (Texas Instruments, 2014). Function wise its comparable to the voltage divider: The comparator measures the voltage difference between the two pins. This voltage drop could be mapped to a resistance and thus a wood moisture (section 3.2.1). These "soil moisture" sensors further simplify the process by returning an analog output between 0 (dry) and 1023 (wet).

Early in the prototyping phase, it was assumed, that soil moisture sensors can measure the water content of wood logs. Although these sensors return an analog value for moist woods with a moisture content of above approximately twenty percent, they are not able to provide a reading for a lower moisture content. This is due to the resistance logarithmic increase with less moisture. A low moisture level has several times the resistance of a wood log with a high moisture content (Figure 5). Wood should have a moisture content less than fifteen percent to process them without risk of shrinking (Niemz *et al.*, 2007).

A way to read a higher resistance is by increasing the voltage. The prebuilt sensor is rated for a voltage between 3,3V and 5V. By using a 9V power source for the voltage divider, it is possible to measure resistances up to 2000 megaohm. This would suffice to measure a wood moisture content of around nine to ten percent (see Figure 5). The voltage divider is built according to the above circuit diagram (Figure 8). R_{known} is a 10 megaohm resistor, R_{wood} – the unknown wood resistance – is built using two stainless steel nails (see Appendix 1). These nails are inserted into the wood at a distance of 1cm with a depth of 2cm. A transistor is added to turn the circuit on or off. The complete circuit can be seen below.

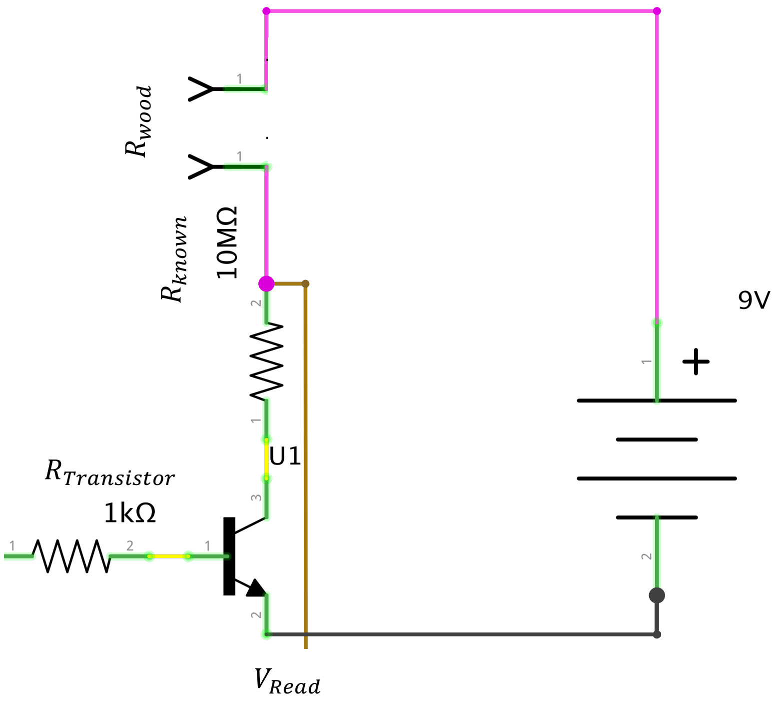


Figure 10: Circuit Diagram - Moisture Measurement (own illustration)

The microcontroller has a precision of 10bits (0-1023), and a voltage reference of either 1,1V or 3,3V. This means V_{read} can either be between $0V - 1{,}1V$ in steps of 0,001V, or $0V - 3{,}3V$ in steps of 0,0032V. This corresponds to a readable moisture content of 8 – 13% or 9 – 15%. The more precise reference of 1,1V is used to more accurately determine the moisture in the low end. Wood with more than 13% moisture content likely will not be processed (see section 3.2.1.1).

To mitigate measurement errors of an analog reading, a hundred readings are done in a short time. An average of all readings is used.

3.2.2 Positioning System

This section discusses suitable approaches to determine a location of a device (3.2.2.1). After evaluating an appropriate method, hardware is chosen to implement it (section 3.2.2.2)

3.2.2.1 Positioning Method

Holzgespür has many wood logs stored in shelves on premise. To control the drying process, wood logs are moved regularly to different shelves: e.g. logs inside dry faster than those outside. Because the wood is moved regularly, only few employees have a complete overview where the inventory is stored. A sensor can help to get an accurate position, e.g. a specific shelf, for each log. This location data can be viewed by all staff, thus making the information available for all employees.

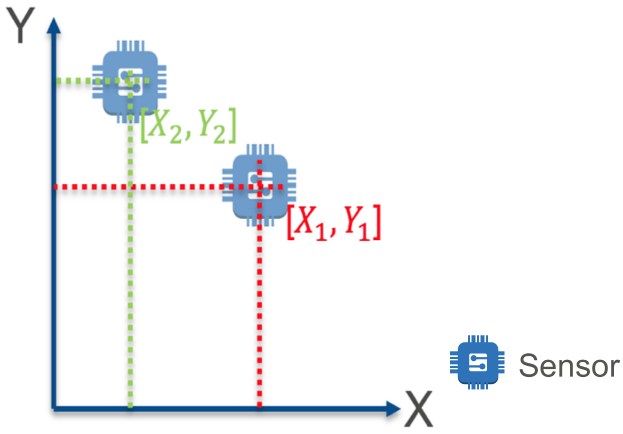


Figure 11: Absolute Position (own illustration)

Positioning systems can either provide an absolute or a relative position. An absolute position can be placed on a map through X and Y values as seen in Figure 11. Each sensor can be identified by its position. A relative position gives a measurement near a known location and is less accurate. Regardless, relative positions should suffice to determine a specific shelf.

Absolute positioning systems require significant more power than relative positioning systems (Kjaergaard, 2012). As runtime is a major factor for the prototypes, and relative positioning systems can be accurate enough for storage environments, only methods for relative positioning are considered.

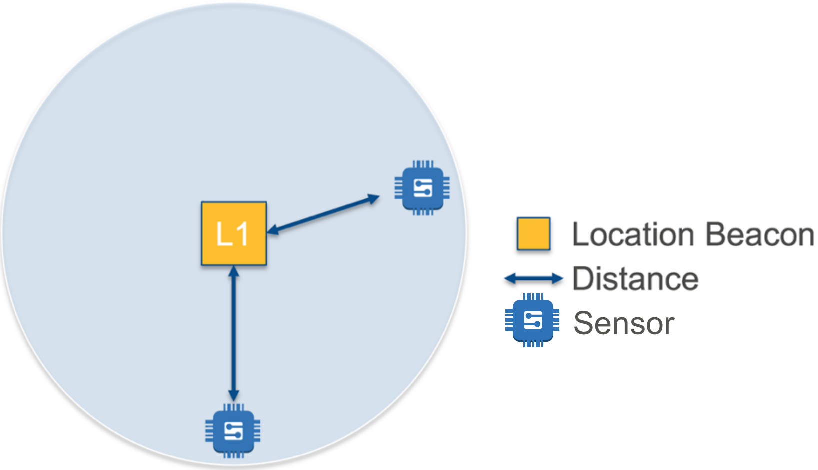


Figure 12: Relative Position - Single Signal (own illustration)

Relative positioning is done by measuring the signal strength between a beacon and a sensor (Patwari *et al.*, 2003; Banks *et al.*, 2005). The absolute location of the beacon should be known (e.g. attached to a shelf). A sensors location can be determined by finding the closest beacon based on its signal strength. The sensor is within a (small) radius of its closest beacon (see Figure 12), e.g. in a shelf. With this approach, single sensors cannot be identified by their position, because they are in a radius around the beacon.

The positioning of relative location tracking could be made more accurate by using triangulation. In this approach, the sensor measures the signal strength of several beacons. The sensor can then be placed at the intersection point of the discovered beacons (Figure 13). This approach however is discarded because of memory constraints of the prototype. The RAM of the microcontroller is not sufficient to store two or more universal unique IDs (UUID) of beacons. Also, mapping only a single signal between sensor and beacon leads to a simpler application logic. Positioning with a single signal should be sufficient to determine a specific shelf in the wood shop.

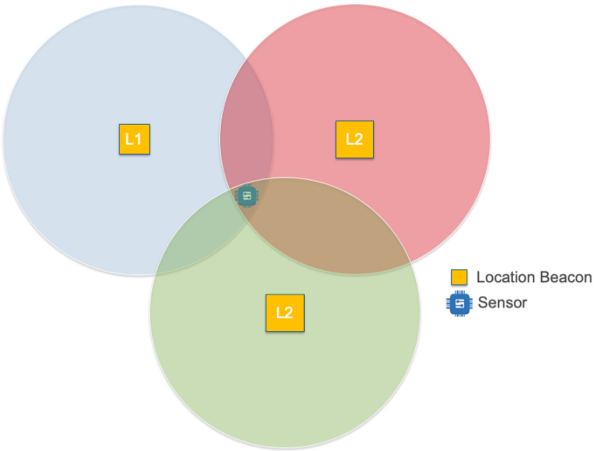


Figure 13: Relative Position - Triangulation

For the chosen relative positioning with a single beacon, a sensor can either send a signal to a beacon or receive a signal from a beacon.

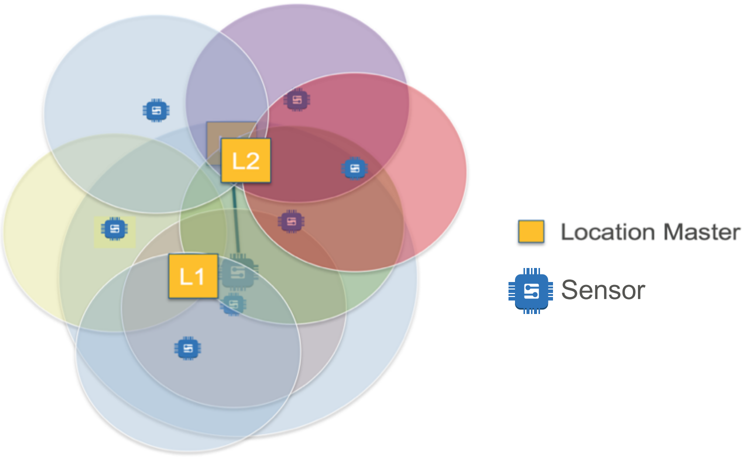


Figure 14: Sending Location - Network Congestion (own illustration)

If the sensor is sending a signal, the location beacon detects near sensors. Thus, the positioning logic and calculation can be done outside the sensor on a more powerful system. Single sensor nodes can be found by tracing its signal with a mobile device. However, actively sending sensors have several drawbacks: power consumption, network congestion and a more complicated positioning logic. Sending a signal uses more power than receiving a signal. Also, the usable bandwidth per channel is limited, as only a few devices can use a frequency at a time. Many actively sending nodes can be congest the channel, thus limiting the scalability (Figure 14). If a frequency is used to capacity, devices need to wait for a free slot to send their signal (Hull, Jamieson and Balakrishnan, 2004). Doing the positioning calculations outside the node, leads to an overall more complicated logic as one node can be near two or more beacons. To determine the nearest beacon the signal strength needs to be compared between several beacons (compare Figure 13). This complex logic could be used to triangulate the position of each sensor. As described above the advantage of more accurate positioning is not needed.

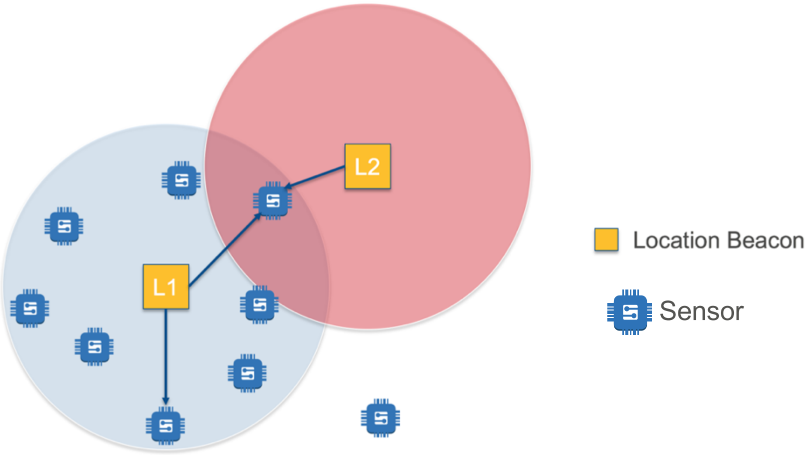


Figure 15: Receiving Location from Beacon (own illustration)

A receiving sensor with an active beacon uses less power. Each node only needs to scan the frequency shortly to determine near beacon's signals. It is also more scalable as only a few beacons are sending a signal compared to many nodes (see Figure 15). If a node is in range of two beacons, the node itself can determine its closest beacon by signal strength. For this approach however, more programming logic needs to be on the nodes themselves. Considering the limited computation power, this might be problematic. Another drawback is a potential less reliable positioning: If a node is not near a beacon, it cannot determine its position. As a fallback, if a log is not in range of a beacon, the previous location can be used, or an employee can be notified of a potential problem. As wood logs should be near a shelf and thus a location beacon most of the time, this less reliable positioning method is acceptable. Because of its scalability and the lower power consumption, sensors receiving signals are built.

3.2.2.2 Positioning Hardware

For relative positioning several technologies are suitable. Most wireless networking protocols can be used, since only the signal strength between the receiving sensor node and the location beacon are used. By comparing several technologies based on range, power consumption and cost, Bluetooth Low Energy was chosen as seen in the table below.

Table 1: Comparison possible Positioning System Technology (own listing)

	RFID	UHF RFID	WIFI	Bluetooth 2.0	Bluetooth Low Energy	Zigbee
Range (up to)	6cm	1-6m	70m	10m	100m	~100m
Bitrate	10Mb/s	10Mb/s	150Mb/s	0,72Mbs/s	2 Mb/s	0,25Mb/s
Power consumption (Idle)	-[7]	-[1]	~10mA	~10mA	~100uA	~15uA
Power consumption (Transmit)	-[1]	-[1]	~85mA	~50mA	~4,5mA	~50mA
Cost/Module	2 €[8]	160 €[2]	3 €	2 €	2 €	25 €

With a Range of up to a hundred meters, Bluetooth Low Energy (BLE) modules have an adequate reach to find the nearest location beacon. RFID and ultra-high frequency (UHF) RFID, as passive technologies, lack this range. These technologies need to be scanned by a handheld device. On average BLE's power consumption is the lowest of the sending technologies. WIFI for comparison uses a hundred times more power at idle and over fifteen times at transmission. BLE cost per module is comparatively cheap. Zigbee as another technology with a low-power consumption and adequate range, has a higher price point. This makes BLE the ideal component for positioning.

The concrete unit for the prototype is a BLE HM-10 module (JNHuaMao Technology Company, 2014). It supports Bluetooth beacon scanning and advertisement modes and can controlled by other hardware through a serial connection.

---

[7] RFID uses active beacons/scanners and passive trackers; power usage is not applicable for the sensors

[8] Cost per active beacon/scanner; cost per passive tracker is negligible

The HM-10 modules support Bluetooth beacon advertisement, so they could serve as location beacons. However, for deployment at a carpentry, prebuilt Bluetooth beacons are used (Shenzhen Minew Technologies Co. Ltd., 2014). These are chosen because they already have a durable case and long battery life at moderate cost.

3.3 Networking

The IoT prototypes themselves lack the computing power to work with the collected data in a meaningful way. So, collected data from the wood moisture sensor and the positioning system needs to be send to a more powerful system through some networking stack. The chosen hardware (section 3.3.1) and the technology (section 3.3.2) is described.

3.3.1 Networking Hardware

Although the prototypes already have a BLE module to send and receive data, the range is limited to up to a hundred meters (see Table 1). A carpentry can span several hundred cubic meters. The exemplary shop spans $650m^2$, exceeding BLEs range. The prototypes however should be able to send data to ensure the system always has valid data. This is necessary to notify an employee of a potential problem, e.g. a prototype not in range of a Bluetooth beacon. To guarantee all prototypes can send their data, a wide radio coverage is needed. Thus, a new technology for data transfer needs to be evaluated.

Table 2: Comparison of Networking Technology (own listing)

	Bluetooth Low Energy	WIFI	LTE-M	LoRaWAN
Range (up to)	100m	70m	100km[9]	15km[3]
Bitrate	2 Mb/s	150Mb/s	400Kb/s	400b/h
Power consumption (Idle)	~100uA	~10mA	~10mA	~20mA
Power consumption (Transmit)	~4,5mA	~85mA	~420mA	~100mA
$\frac{\text{Cost}/\text{Module}}{\text{Monthly}}$	-	-	from 1,5€	-
$\text{Cost}/\text{Module}$	2 €	3 €	55 €	20 €

As seen in Table 1, BLE and WIFI have a limited range which makes those technologies unsuitable for long-range communications. Thus, other technologies need to be evaluated (Table 2). LTE-M, a subgroup of the LTE wireless communication standard for IoT devices, enables IoT devices to send data over a long-range at a fast speed. LTE-Ms power consumption at idle is only slightly higher than other networking technologies such as WIFI and Bluetooth 2.0. Its peak power consumption however is rather high and would reduce the runtime of the prototypes power source (section 3.4.2). This technology is cost prohibitive: besides a high module cost of 55€ per unit, LTE-M also requires a subscription to a

---

[9] In rural areas. Visual obstructions greatly reduce distance

carrier to use its network (M2M, 2018; Telekom, 2018). These two points discard LTE-M as a suitable technology.

LoRaWAN is a wireless communication technology using license free radio frequency bands (868MHz in Europe) to send data to a remote gateway at a distance of up to 15km. Its costs per module is less than half of an LTE-M unit. By using license free RF bands, LoRaWAN can be used without a monthly cost. These bands however are highly regulated, and each device is limited by duty cycles, in the amount of data it can send (Bundesnetzagentur, 2018). The limited bandwidth is no restricting factor as no real-time data is needed and the collected data packages will less than 60 Bytes. The duty cycles are respected by only sending data in intervals of around six hours. Using LoRaWAN gives access to an existing and extendable infrastructure.

As LoRaWAN seems to be a suitable networking technology to send collected data, a RFM95 LoRa radio is used for this prototype (Hope Microelectronics Co., 2014). It is a widely used radio for LoRaWAN with accompanying libraries for easy programming. This radio supports sending and receiving data on several frequency bands, including the license free 868MHz band for Europe.

3.3.2 LoRaWAN and the Things Network backend

Viewed from an OSI model perspective, the LoRaWAN protocol consists of a long range (LoRa) physical layer and a media access control layer of the OSI model (International Telecommunication Union, 1994). The specification for this protocol was released in 2015 and is maintained by the Lora Alliance (LoRa Alliance, 2017). This specification also describes a system architecture for communicating between devices and a gateway, while respecting frequency, duty cycles and timing set by local RF regulations. Devices can use LoRaWAN to send and receive data. In the context of this thesis, prototypes will only be able to send data. This choice has been made to save memory on the MCU and increase the sleep cycle duration to save battery power.

Devices send data through their radio over LoRaWAN to a gateway. These data packages can then be routed via internet protocols to a network server, such as The Things Network (TTN). Further the backend can route the data packets to an application server for data processing (see Figure 16).

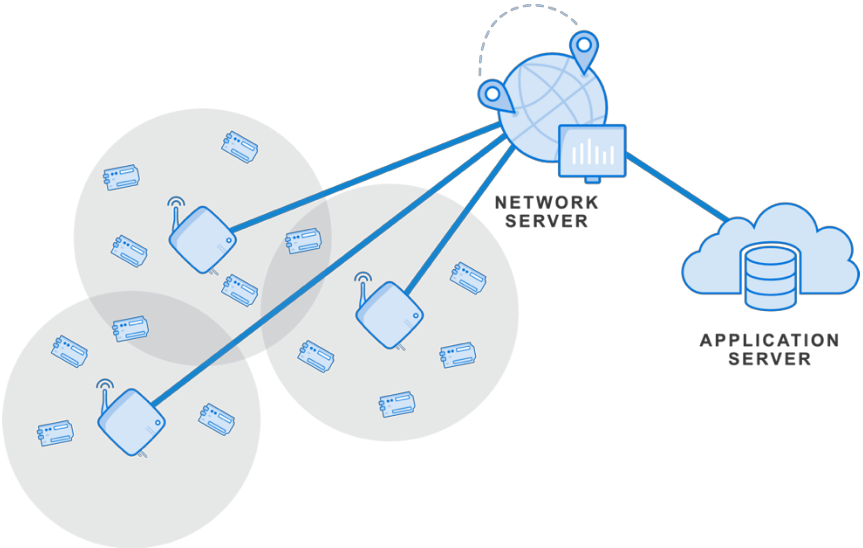


Figure 16: The Things Network Overview (from *TTN Network Architecture*, 2018)

The Things Network is an open source routing system for things. It implements the LoRaWAN specification to accept data packages send from things through gateways. Devices have to be registered at the Things Network with a unique id. Data packages are encrypted with this id before being send to the backend. The backend decrypts the messages and routes them to a linked application via different APIs (see Figure 17). TTN supports basic monitoring of devices and gateways to ensure they are still responding. Data from an application can also be send through TTN to devices, but this feature is not implemented in the prototypes for this thesis.

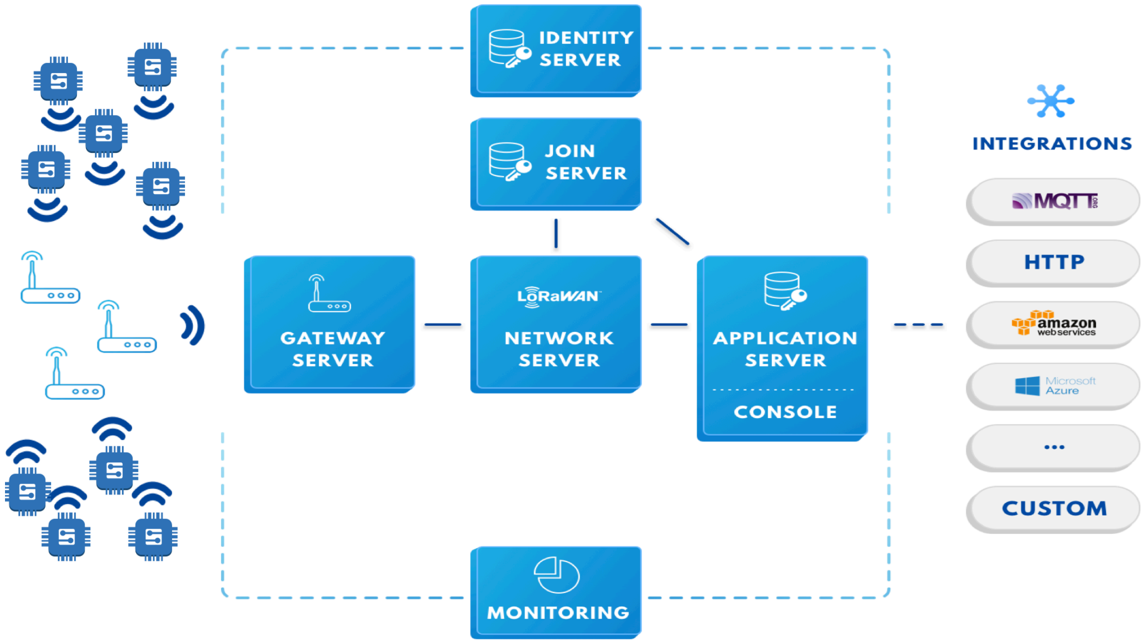


Figure 17: The Things Network Stack (based on *TTN Network Architecture*, 2018)

3.4 Additional Components

The components for data collection and networking need to be controlled by a microcontroller (MCU) or -processor. These components also need a power source. In this chapter several controlling units (section 3.4.1) and power sources (section 3.4.2) are compared to find suitable parts for the prototype.

3.4.1 Microcontroller or Microprocessor

For the prototype several microcontroller (MCU) boars and a System on a chip (SoC) board are considered. The MCU or SoC will control all components from the previous chapters attached to it (sections 3.2 & 3.3). The unit reads measured raw data and does basic operations on them. These operations include finding the nearest location beacons ID (section 3.2.2) and calculating the average of several moisture readings 3.2.1.2). Further the data is combined to a single packet and sent through the LoRaWAN radio.

Most available MCUs and SoCs will have enough processing power and memory to handle these tasks. The main requirements to consider are power usage, ease of use, size and cost. To achieve the requirement of ease of use only prebuilt prototyping boards are evaluated. These include the “Raspberry Pi Zero W” (RPiZW) (Raspberry pi foundation, 2014) and Arduino Boards (Arduino, 2014, 2016). Both have accessible data pins to attach the used sensors and components.

The RPiZW is a shrunk version of the teaching board Raspberry Pi. The “W” version includes wireless LAN and Bluetooth, so a separate component could be saved. However, the added wireless capabilities increase the power usage of the board. The RPiZW has a SoC with a clock speed of 1 GHz and 512 MB RAM and is by far the fastest considered Board here. Since it is a SoC, the RPiZW by default runs on a full operating system. A high clock speed, many integrated peripherals and multi-purpose OS pushes the power consumption above 100 mAh and thus makes the board unsuitable for running it prolonged periods of time on batteries.

The Arduino company builds open source computer hardware and platform. It designs and builds several single-board microcontrollers – Arduino Boards. There are several board designs with ranging specifications (e.g. 8MHz to 400Mhz). To limit the selection, only boards with the “minimal” form factor are considered. Currently available are the “Nano” and “Pro Mini” with an ATmega328 MCU, and the “Micro” Board with an ATmega32U4 MCU. Since the “Nano” and “Pro Mini” Boards are mostly the same, except for an additional USB I/O on the Nano board, the Pro Mini is considered since it has an even smaller form factor.

Table 3: Comparison of suitable Computers (own listing)

	Raspberry Pi Zero W	Arduino Micro	Arduino Pro Mini
Size mm (LxW)	65x30mm	48,3x17,8mm	33x18mm
Features	Multi-Purpose CPU WLAN Bluetooth 4.0	Micro Controller ATmega32U4	Micro Controller ATmega328
Clock speed	1000MHz	16 MHz	8 MHz
RAM	512MB	-	-
Flash	microSD	32KB	32KB
SRAM	-	2,5KB	2KB
EEPROM	-	1KB	1KB
Power (Idle)	120mA	50mA	5mA
Power (Sleep)	not available	unknown	1,1mA
Power V	5V	5V	3,3V
Cost	10 €	7 €	2 €

In the above stated requirements – power usage, size and cost – the Arduino Pro Mini bests its competition (Table 3). It is the cheapest board with the lowest power usage and also smaller than the Micro board. Drawbacks like the missing USB I/O for easy reprogramming and low clock speed of 8MHz are negligible for the use cases. The biggest drawback is the limited SRAM and EEPROM: the LoRa library stack is rather large and might push the memory limit.

For the prototype the Arduino Pro Mini is chosen.

3.4.2 Power Source

Two factors are important for the power source: the provided voltage to run each component and capacity to run the whole prototype for an extended period. Batteries can be divided in two groups: primary and secondary cell batteries (Winter and Brodd, 2004).

Secondary cell batteries like nickel-metal hydride batteries (NiMH) or lithium polymer batteries (LiPo) have advantages over primary cell batteries. These types of batteries are rechargeable and provide a greater variety of forms and voltages. This could lead to a smaller case as those batteries do not have to be replaced and can be integrated more tightly. However secondary cell batteries pose a higher fire hazard (Amon *et al.*, 2012). Since a carpentry is prone to fire damage, secondary cell batteries must not be used to minimize the risk.

Primary cell batteries have a lower risk of fire or explosions than secondary cell batteries (Amon *et al.*, 2012). Another benefit of primary cells is a higher capacity compared to secondary cells. However, the prototype needs to accommodate a way to replace drained cells. Alkaline type batteries are used as they have the highest energy density for primary cells.

A 3,3V power source can run every component, except for the moisture sensor. The moisture sensor itself needs 9V to measure the wood moisture (section 3.2.1.2). A raw input voltage between 3,35V – 12 is supported by the 3,3V Arduino Pro Mini (section 3.4.1). Its internal regulator can power it down to an operating voltage of 3,3V to power the BLE component (sections 3.2.2) and the LoRa radio (section 3.3.1). Considering the high usable voltage range, capacity is the driving factor for choosing a battery size. Another consideration is the size to keep the overall prototype small.

Table 4: Comparison of common Battery Types (own listing)

	AAA	AA	D	E (9-Volt)
Nominal Voltage	1,5V	1,5V	1,5V	9V
Typical capacity (Alkaline)	1200mAh	2700mAh	12000mAh	565mAh
Size	10,5x44,5mm (dia x H)	14,5x50,5mm (dia x H)	34,2x61,5mm (dia x H)	48,5x26,5x17,5 mm (H x L x W)
Volume (rounded)	3,85cm³	8,34cm³	56,5cm³	22,49cm³
Density	311,4 mAh/cm³	323,8 mAh/cm³	212,4 mAh/cm³	25,12mAh/cm³

As seen the table above AA type batteries have the highest density of all widely used primary cell battery sizes. Since it only has a nominal voltage of 1,5V at least three AA batteries have to be wired in series to stay above the input voltage for the Arduino Pro Mini. The moisture sensor needs at least 9V. Six AA batteries would need to be connected in series to output 9V. A single E battery can output 9V but has a low capacity. One of these is smaller than three (or six) AA batteries. Since the moisture sensor can be turned off when it is not in use, capacity can be negligible for this sensor.

Thus, a mix of three AA batteries and one E battery is used to power all components in the final prototype.

3.5 Designing the Hardware

In this sub chapter the design and build process for the prototype is described. The hardware prototype went through several iterations. Each step improves on previous designs based on feedback or noted limitations.

3.5.1 Prototype MKI

The first Iteration is a sketch up of all the proposed components with a simple case around it to protect them. This design was created after the first interview at Holzgespür to have an approximate prototype size. The prototype should not interfere with day-to-day business so it has to be as small as possible. At this stage a prebuilt moisture sensor was proposed to determine the wood moisture content (section 3.2.1.2). To measure the wood moisture, the prototype needs to be placed in a proximity of few centimeters to the wood log. With around $28cm^3$, this case design is a lot smaller (Figure 18) than the final version. This is due the initial usage of a high capacity LiPo battery. Since these secondary cell batteries, a prone to explode if handled wrongly (section 3.4.2), the design was revised.

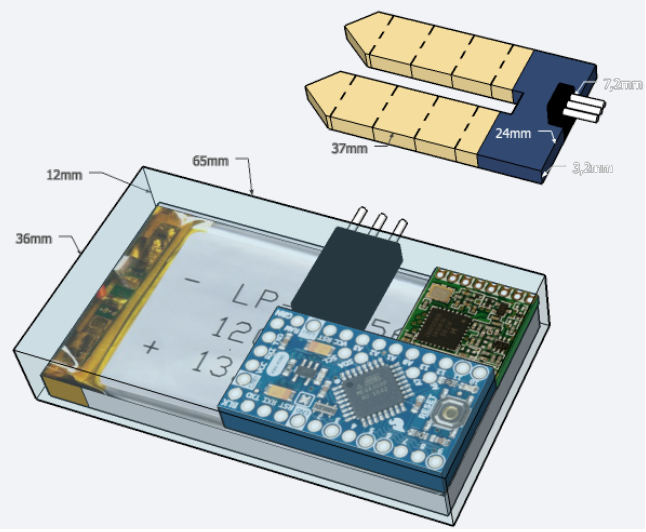


Figure 18: First Prototype sketch up (own illustration)

3.5.2 Prototype MKII

The second iteration uses three AA alkaline batteries to reduce the fire risk (section 3.4.2). It still uses the prebuilt moisture sensor (section 3.2.1.2). Since every component uses 3,3V, each component can be powered by a single power source, keeping the wiring diagram rather simple as seen in the figure below.

The starting point for the wiring diagram is based on the LMIC Library for the RFM95 radio. The six used digital pins are pre-defined and it is recommended to use these pins. Every other component needs to be connected without collision to already used pins. The RFM95 radio gets its 3,3V power from the Arduino Pro Minis power regulated VCC pin. The HM-10 Bluetooth module and the prebuilt moisture sensor uses the same power output. Also two digital pins are used by the HM-10 for serial I/O. The moisture sensor uses one analog pin. On the I/O side there is room for additional moisture sensors. These could be used to either measure several logs or several locations of one log to improve accuracy.

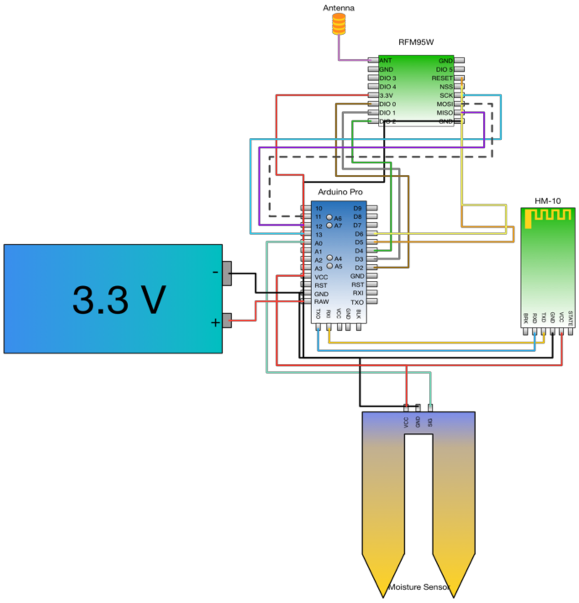


Figure 19: Wiring first Prototype (own illustration)

The RFM95 radio can be stacked on the Arduino. This increases stability compared to longer, looser wires. The other components are placed rather loosely in the case. The case has “cable ducts” – openings – to route pins outside. These can be used to reprogram the Arduino and lead sensor cables outside. Using AA batteries increases the size compared to the first iteration (Figure 20). To increase stability of the case and make the 3D print process easier, the width of the walls has also been increased. Overall the volume increased to $66cm^3$. It grew in height and length. According to the carpentry the size is still suitable for deployment although it should not be higher.

The second iteration also has holes to screw the case to wood logs. This makes it possible to attach the prototype to wood logs. It should be noted that this approach is destructive to the wood logs and thus is not used in later iterations.

Wiring each component to the Arduino proved to be prone to breakage. Wires got loose through vibrations and movements which broke a build of this iteration of the prototype.

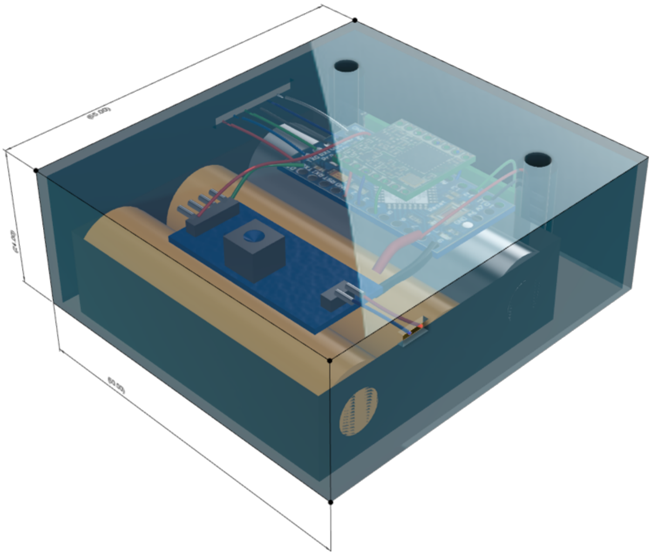


Figure 20: Case Design first Prototype (own illustration)

3.5.3 Prototype MKIII

The third iteration uses the voltage divider seen in Figure 10**,** to measure the wood resistance. Since the new prototype needs a higher voltage for measurements of high resistances, a second power source has to be added for this circuit. To persevere power, the lower capacity E batteries (section 3.4.2) can be controlled by the Arduino via a transistor connected to a free digital I/O pin. The other components are connected as in Figure 19. A switch was added to turn the prototype on or off (Figure 21). This makes it easier to reset, or store it for extended periods of time.

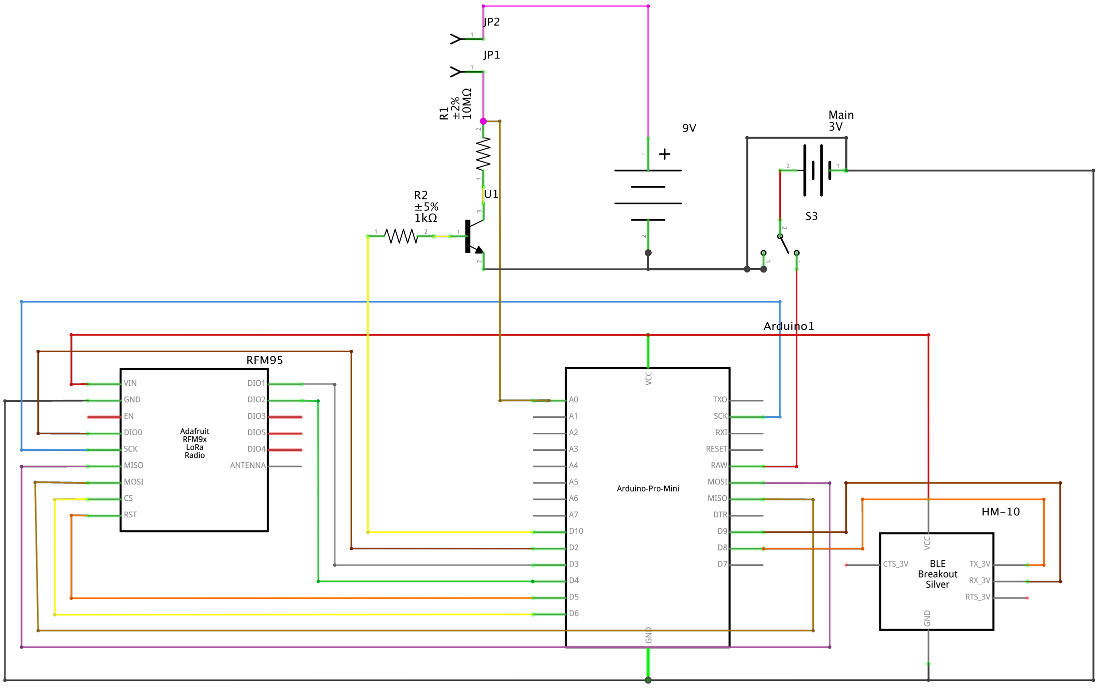


Figure 21: Wiring third Prototype (own illustration)

The figure below is a schematic of the final wiring design on a breadboard. To ensure a more resilient design, all components are wired to a prototyping board. Soldering to the board instead of to the Arduino decreases the difficulty of soldering, leading to more stable solder joints. This design keeps the prototype confined in a small space while still being able to place the largest components, the batteries, freely.

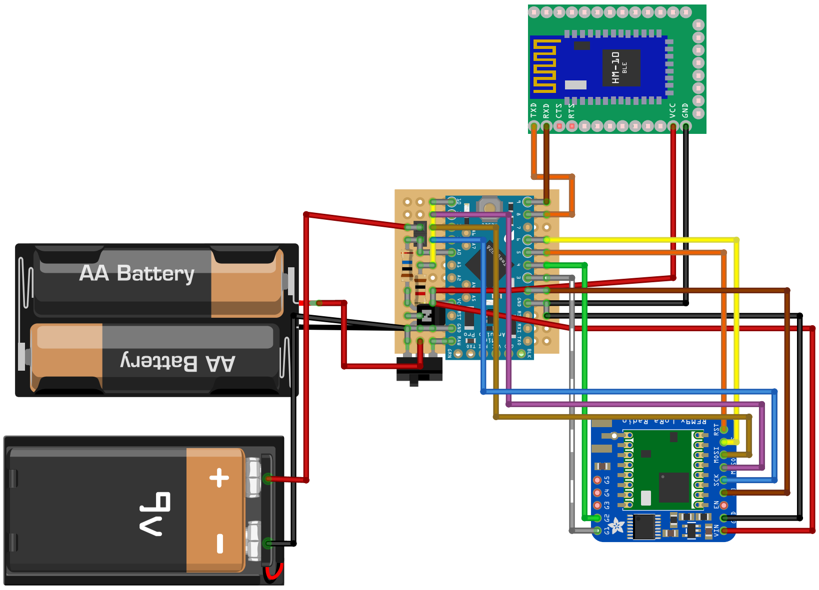


Figure 22: Breadboard Diagram third Prototype (own illustration)

The case increased in length by the size of a 9V battery. Since the layout is more stretched out, the components can be rearranged, leading to a decreased height. The overall volume of the new case is $145cm^3$ (Figure 23). Since the E-type battery is rather large, the main breadboard can be secured by additional walls, which create a separate compartment. This compartment makes the components fit tightly and should reduce the risk of broken solder points even more. The screw holes got removed from this design because screwing the case to the wood log proved to be too destructive. Instead the case is supposed to be attached by zip ties or rope.

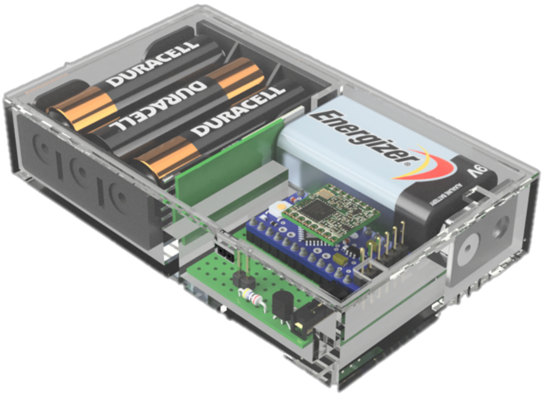


Figure 23: Case Design third Prototype (own illustration)

Prototype MKIII (improved)

During assembly of the third prototype, it became clear that the fit was too tight. As a fix, the wall thickness got decreased by 0.5mm. Further the compartment walls got angled to give the fitting more tolerance by being able to slide the breadboard in. The top case's corners got also angled to accommodate 3D printing tolerances. To keep the case closed during movement, snap clips got added. Also, two grooves got added as a lead for a rope or zip tie to attach the case to a log (Figure 24).

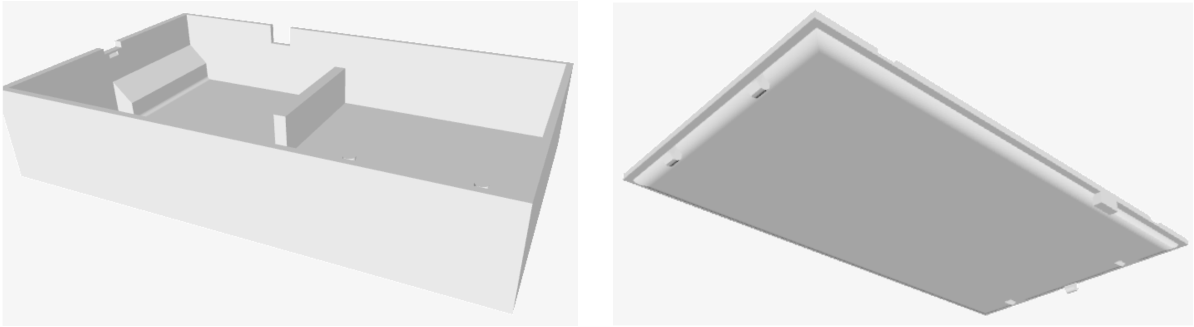


Figure 24: Revised Case Design Third Prototype (own illustration)

The final prototype has a dimension of 108mm * 64mm * 21mm.

3.6 Building the Software

This sub chapter describes the programming of the prototype itself (section 3.6.1). It also gives a brief description of the TTN network code to decode the received package (section 3.6.2). Further the architecture and development of the application is portrayed (section 3.6.3).

3.6.1 Programming the Prototype

The prototype is controlled by the Arduino Pro Mini. It is supposed to sleep for long periods. On wake-up it scans for Bluetooth beacons, measures the resistance of the attached wood log and sends a package through LoRaWAN.

The Arduino Pro Mini is programmed in C++ through Arduino Studio IDE[10]. On startup the Arduino sets up the HM-10 BLE module and the RFM95 radio. The HM-10 is configured to scan for BLE beacons, the RFM95 registers itself at The Things Network. After a successful setup its onboard LED light up for 5 seconds to give an indication of successful initialization. Afterwards the Arduino enters its low-power mode. If the setup fails, the Arduino needs to be reset manually because it is stuck waiting for a proper reply from one module.

The Arduino Pro Mini is supposed to wake-up about every six hours from its low-power mode and activate its connected modules. This is implemented with the "LowPower" library[11]. It provides easy-to-use functions to put the Arduino in low-power mode with periodic wake-ups without an external signal.

---

[10] https://www.arduino.cc/en/Main/Software (Accessed 15.09.2018)

[11] https://github.com/rocketscream/Low-Power (Accessed 15.09.2018)

This function is looped till the six-hour interval is reached. As the Arduino has no real-time clock, it can shift by a few minutes over the time span of six hours. Since the data it collects is not needed at fixed times, this shift can be ignored.

Every time the Arduino finishes its sleeping loop, it activates its components as needed. First it scans for BLE beacons in its area. This is triggered by sending commands to the HM-10 module over a serial connection. The module detects universally unique identifiers (UUID) and signal strengths among less relevant data (JNHuaMao Technology Company, 2014). These 16-byte UUIDs have to be filtered: Invalid beacons from other broadcasting devices (e.g. smartphones) need to be discarded. Valid UUIDs need to be compared by their signal strength. The UUID with the strongest signal, indicating a close beacon, need to be saved. This filtering needs to be done in real-time. Otherwise the Arduinos buffer will be filled by the returned data, leading to unexpected behavior. The HM-10 might return invalid data through its serial connection. To ensure better results, the Arduino retries up to three times to find a valid beacon UUID. If no UUID could be found, a NULL[12] UUID is saved as a placeholder.

After an UUID is saved, the Arduino measures the resistance of the wood as described in section 3.2.1.2. This is done by activating the wood moistures power circuit and reading the analog value returned by the sensor. Values can fluctuate between readings, so an arithmetic mean of one hundred tries is calculated. The analog reading needs to further divided it with the Arduinos analog resolution. The result then equals the voltage V_{read}. The resistance R_{wood} can be calculated using the formula for the voltage divider (as seen in 3.2.1.1). R_{wood}, a two-byte value, is saved.

With an UUID and a resistance value, a transportable package for LoRaWAN is created. The 16-byte UUID and 2-byte moisture reading is encoded to an 18-byte array. This array can be sent in one cycle through the RFM95 radio as part of a LoRaWAN package. The “lmic” library[13] creates the package. Besides the array, the TTN- Application and Device ID is added. The library also handles all technically aspects of using the RF band: It keeps track of its duty cycle, chooses an appropriate channel and sends the package.

The Arduino returns to its sleep cycle after a package is sent.

3.6.2 The Things Network configuration

The Things Network can do most of its work without further configuration. It can route the package from a device to an application based on its Application and Device ID. After routing the message, TTN decrypts the message. This raw data could already be used by external applications. To make further work easier, the message gets decoded at TTN. Via JavaScript the package gets split to a resistance and a UUID reading. The values get named and are now accessible through APIs.

---

[12] “00000000-0000-0000-0000-000000000000“

[13] https://github.com/matthijskooijman/arduino-lmic (Accessed 15.09.2018)

3.6.3 Designing the Web Application

To give employees access to the data, a responsive web application is needed (see Table 5). The application can be used on the work floor by a smartphone. It displays the device name, the measured moisture and the nearest beacon's name . Additionally, the wood moisture is displayed as a graph to show its drying trend. Users can choose a wood type to have a more accurate calculation of the moisture content. A notification can be toggled to be informed when a specific moisture content is reached.

Table 5: Web Application - Responsive Wireframe (own illustration)

To get the data from the Things Network an application is built with Node-RED[14]. The Node-RED environment collects the data from TTN, saves them to a database and provides a frontend (Figure 25).

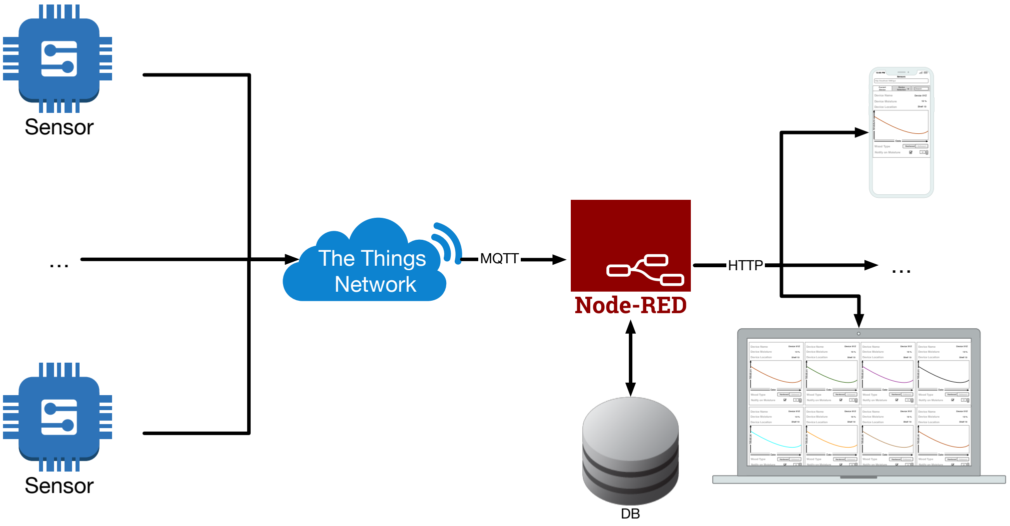


Figure 25: Web Application Overview (own illustration)

---

[14] https://nodered.org (Accessed 15.09.2018)

Node-RED is a visual, flow-based programming tool. Originally developed by IBM, now part of JS Foundation, the tool was designed to easily connect Things with online services and APIs. Programming can be done by dragging, dropping and connecting function-nodes together to create flows. These flows can be seen as simple applications. Because of its ease of use, basic software can be built without a programming background. Employees at SMEs could build flows for their Things and thus can easily and quickly deploy new prototype applications.

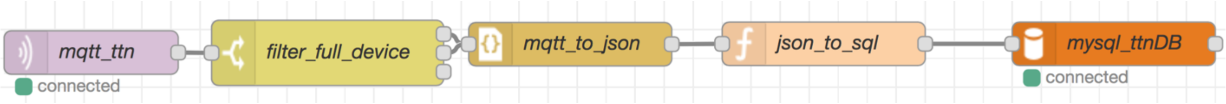


Figure 26: Node-RED Flow MQTT to DB (own illustration)

One example is the saving of device readings in a database as seen the figure above: With the function “mqtt_ttn”, Node-RED can subscribe to new data from TTN. MQTT[15] (Message Queuing Telemetry Transport) is a machine-to-machine messaging protocol. Machines, such as things or services, can publish or subscribe to messages. TTN publishes one message for each sensor-reading and for full-devices – including its sensor readings. These messages need to be filtered to avoid duplicate data points. The function “filter_full_device” discards the separate sensor-readings and only lets full-device readings through. An MQTT object consists of a topic to identify the device and a message with metadata of its LoRaWAN connection and the payload itself (e.g. moisture and location).

To easily work with the mqtt-data, it gets converted to JSON (mqtt_to_json). Further this JSON-object is converted to an SQL query. (json_to_sql), to save the relevant data in a connected database (mysql_ttnDB). The json_to_sql node contains the only self-written code of the flow: a simple “Insert” SQL statement.

The data is stored to a MariaDB[16] SQL database, to keep a record of all incoming data**.** For this prototype, not all attributes are saved. The saved attributes are the timestamp, device id and the sensor readings (Table 6). Reducing the saved attributes keeps the SQL query simple and the database small. The stored data can be used to make long-term observations and audit movements or the drying processes.

---

[15] http://mqtt.org (Accessed 15.09.2018)

[16] https://www.mariadb.org (Accessed 15.09.2018)

Table 6: Node-RED Example Database Table (own listing)

Field	Type	Null	Key	Default	Extra
id	Int(11)	NO	PRI	NULL	auto_increment
timestamp	datetime	NO		NULL	
dev_id	text	NO		NULL	
payload_raw	blob	NO		NULL	
moisture	decimal(7,4)	NO		NULL	
nearUUID	text	NO		NULL	

MariaDB, a relational database, is chosen because of its widespread usage. According to the annual Stackoverflow survey, relational Databases are most commonly used (Figure 27). Employees might already be familiar with relational databases either, through MySQL[17] or Microsoft Access[18]. The practical experience could make it easier for them to create and service such databases. There is also a variety of graphical tools available to manage relational databases, test or create SQL queries.

For a better scalability the use of a NoSQL database like MongoDB[19] might be more suitable. These document-oriented databases allow the direct storing and interaction of JSON objects.

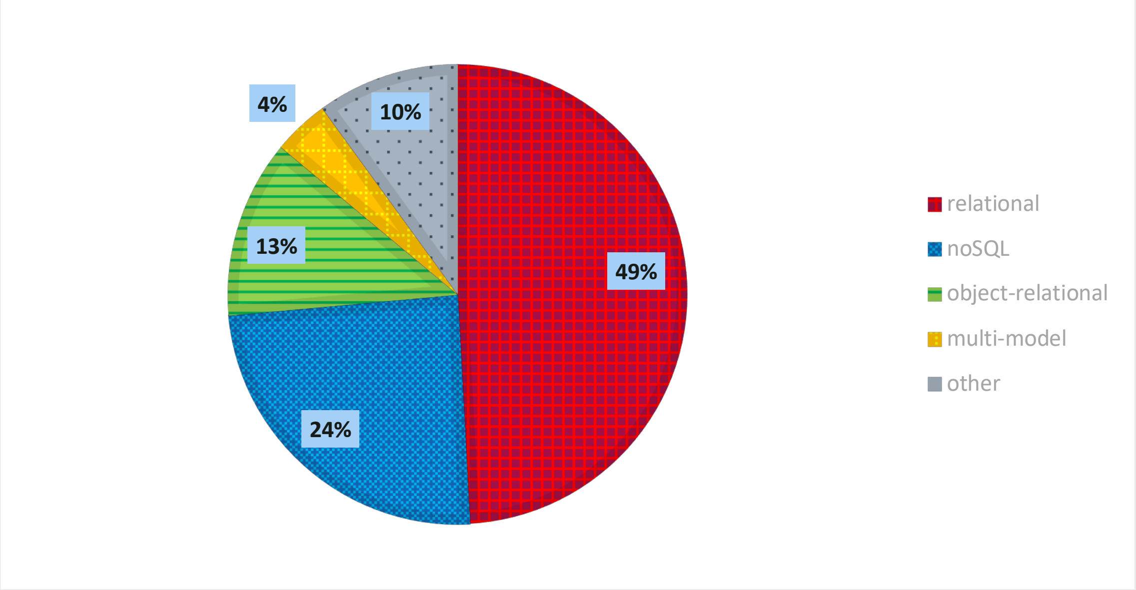


Figure 27: Commonly used Databases (based on StackOverflow, 2018)

---

[17] https://www.mysql.com/ (Accessed 15.09.2018)

[18] https://products.office.com/en-us/access (Accessed 15.09.2018)

[19] https://www.mongodb.com/ (Accessed 15.09.2018)

Node-RED also has the capabilities to create a Responsive Frontend. The previously generated JSON object can be parsed to GUI nodes to display Graph and Texts for each device (Figure 28).

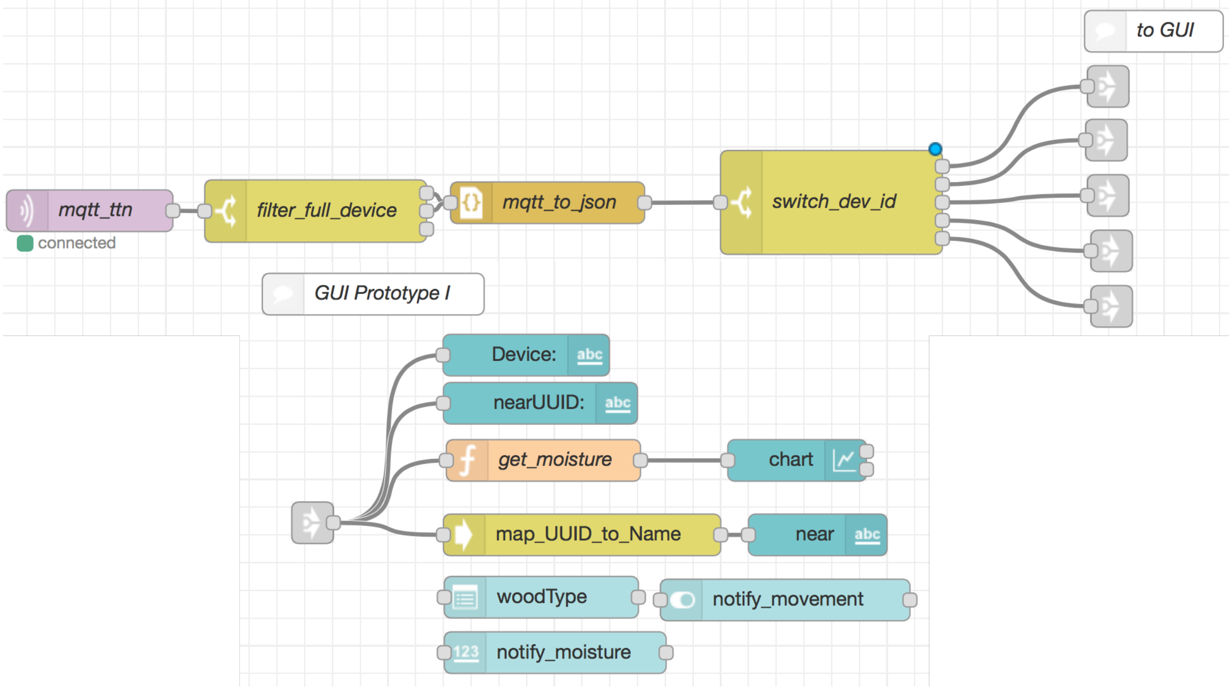
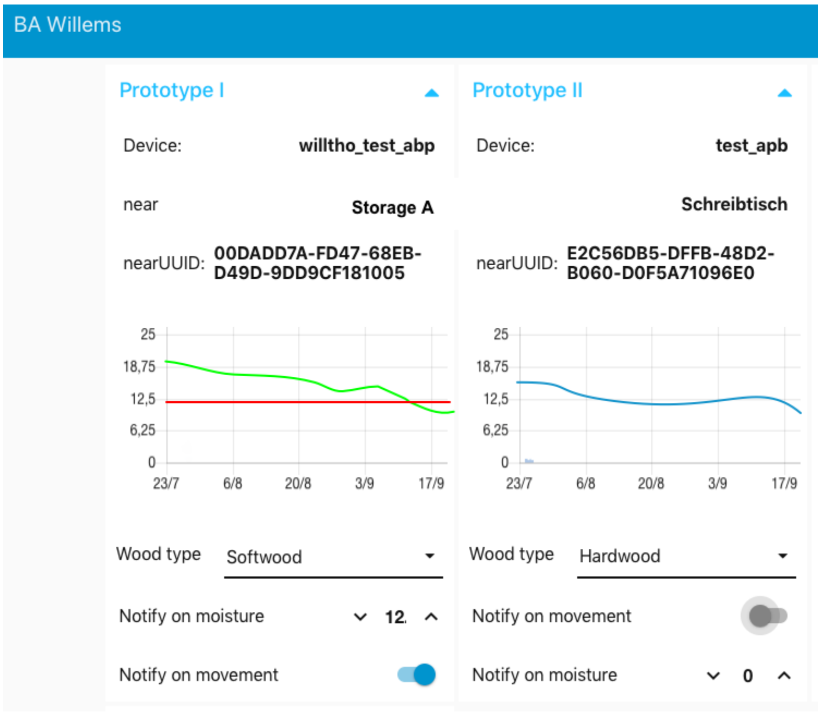


Figure 28: Node-RED Flow MQTT to Frontend (own illustration)

The created JSON object gets routed to its corresponding GUI group by its device id (“switch_dev_id”). Each device has its own group of GUI elements. These GUI groups can include text fields, charts, buttons sliders, or switches among other elements. For this prototype, every group comprises text fields, a graph and switches (Figure 29). The device name and the nearest UUID can be read from the corresponding JSON attributes. The resistance reading gets converted to a moisture content (“get_moisture”) depending on the wood type (compare Figure 5). An employee can set the wood type (hardwood, conifer) via a drop-down menu. To give the UUID a more human-readable name, the value gets mapped to a name (“map_UUID_to_Name”) and displayed as a text field. An employee can also set notification alarms for specific events. Notifications can be sent out if the wood log changed its location, or the wood moisture reached a desired content. The GUI itself can be accessed at the “gui/” page (Figure 29)



Device: willtho_test_abp

near Storage A

nearUUID: 00DADD7A-FD47-68EB-D49D-9DD9CF181005

Wood type Softwood

Notify on moisture 12.

Notify on movement

Device: test_apb

Schreibtisch

nearUUID: E2C56DB5-DFFB-48D2-B060-D0F5A71096E0

Wood type Hardwood

Notify on movement

Notify on moisture 0

Figure 29: Node-RED Frontend Example with two Devices (own illustration)

A drawback of using Node-RED to draw a responsive GUI, is that GUI groups cannot be created programmatically. Each device, needs to have its own group within Node-RED. A new group has to be created for each new device manually. Creating these groups for each prototype leads to redundant code which cannot be reduced. This limits the scalability of using Node-RED as a single, complete system for backend and frontend software. For production environments different approaches are recommended.

One such approach could be to connect other services to Node-RED. The frontend could be created by a different software like Grafana[20]. Node-RED can still act as a receiver for MQTT messages and parse them to a database (Figure 26). Further the database could be queried through Node-RED. If needed, Node-RED can prepare the data for the frontend software before passing it.

For this thesis Node-RED is adequate. A simple-to-use tool like Node-RED can motivate employees of SMEs to fiddle and create their own ideas, without being discouraged by complicated programming tasks. To showcase and test prototypes the simple interface of the tool should be sufficient for most task. It is however encouraged to use more suitable and scalable tools for production environments. As noted, a small application with MongoDB and Grafana could be used. These tools however have a higher learning curve than Node-RED.

---

[20] https://grafana.com (Accessed 15.09.2018)

4 Deployment and Tests

This chapter is about the build process for hard- and software (section 4.1). Further the testing methods are described (section 4.2). These include all hardware components (moisture sensor, positioning system, networking, and battery), and the web application.

4.1 Setup

This section illustrates the build process of the prototype hardware (section 4.1.1) and the setup of the web application (section 4.1.2)

4.1.1 Building the Hardware

The prototype itself was built four times. Two early iterations were built for testing and showcase purposes (see 4.1.1.1). Two prototypes with the final design were built (see 4.1.1.2).

4.1.1.1 Prototype Iterations

The first prototype was built using the design described in section 3.5.2. It was built to test the functionality at the carpentry. As no cases were printed yet, all components were loosely placed inside a generic case (see Appendix 2). This first prototype broke solder points due to vibrations during transportation. Separate testing of the prebuilt moisture sensor proved it to be not powerful enough (see 4.2), rendering this design obsolete.

The second built prototype was based on the MKIII design (section 3.5.3). For this prototype the Handwerkskammer Koblenz printed a case. It was primarily built to showcase the hardware for one of their events and has been assembled on short notice. The case itself is rigid and stable, however to small: The components had to be pushed into place (see Appendix 3). The connector for the E-type battery doesn’t fit in the small case. Wires had to be directly soldered to the battery, making it not user replaceable. Due to printing inaccuracies, the top does not fit perfectly, and the case has to be sealed with tape (see Appendix 4).

4.1.1.2 Final Prototype

The previous design was revised as seen in the section “Prototype MKIII (improved)”. The university printed three cases after this improved design. Due to the decreased wall size and a different printer, the cases themselves are more fragile. The layers of the walls separated (see Appendix 5). These separating walls got melted together to improve the stability. However, during assembly one case broke beyond repair (see Appendix 6).

Some further modifications have been made to the cases and the design to regard the place and stability restrictions: The inside compartment wall had to be shortened to fit the Bluetooth module and E-type battery. Also, internal holes have been added for cable routing and the pin outs for the Bluetooth module (see Appendix 7). The breadboard design has also been altered to give the prototype a more cleaned up look (see Appendix 8). The LoRa radio and parts for the voltage divider are now on the bottom of the breadboard. This makes the reset button and the probe connector for the voltage divider more accessible (see Appendix 9).

Each prototype is flashed with a unique device ID assigned by TTN. This is a manual process and limits scalability. One of the final prototypes got an extra opening for easy access to the serial port. This allows for quick reprogramming and testing of the hardware. Because this opens another possible ingress point, this modification was only applied to one of the cases.

4.1.2 Deploying the Web Application

The web application (see 3.6.3) is hosted on a Linux server platform running a Docker[21] environment. Docker allows containerization of all needed applications. This has the benefit of easy deployment across a variety of host OSs.

The web application itself consists of two containers: A Node-RED and a MariaDB container. These containers are each based on their respective sample Dockerfile[22] [23]. The containers are configured and linked through a Docker-Compose file[24] (see Appendix 1).

A Node-RED environment set up via the docker-compose file, can be accessed locally through port 1880[25].

4.2 Testing the use cases

In this section, the components and web application are tested to determine if they can fulfill their respective use case. First the moisture content measuring method is tested (section 0), followed by the

---

[21] https://www.docker.com (Accessed 15.09.2018)

[22] https://hub.docker.com/r/nodered/node-red-docker/ (Accessed 15.09.2018)

[23] https://hub.docker.com/_/mariadb/ (Accessed 15.09.2018)

[24] https://docs.docker.com/compose/ (Accessed 15.09.2018)

[25] http://localhost:1880/

4.2.1 Wood Moisture Content Measuring

The moisture measurement was tested twice each for a hardwood and a conifer. Once at the carpentry, to test the circuit design itself (Figure 10) and a second time with the finished prototypes.

Table 7: Wood Moisture Measurements (own listing)

Wood Type	Professional Tool	Circuit	Multimeter
Ash Tree (Hardwood)	6.9%	0%	7%
Oak Tree (Hardwood)	12%	12%	12%
Spruce (Conifer)	9,4%	9% - 10%	9% - 10%
Walnut Tree (Hardwood)	9%	8% - 9%	8%

At the carpentry, the moisture contents of several wood types were tested (Table 7). Each wood type was tested by the built circuit, a multimeter and the professional tool[26] the carpentry generally uses. The multimeter is a more accurate voltage reader, having a higher precision than the used Arduino (compare section 3.2.1.2). Voltage read by the circuit and the multimeter was mapped to a wood moisture content as seen in section 3.2.1.

The multimeter and voltage divider circuit had readings equal to the professional tool. Only a dry ash tree log with a moisture content of 6,9% could not be measured. An ash wood tree with ~7% moisture content, has a resistance between 12.000 – 14.000 megaohm (James, 1988). This resistance can be seen as an upper limit for the measurement the prototype can read. If the prototypes reading is 0, the moisture can be assumed to be less than 8%. Since, for the most part the moisture content needed for processing is between 7% and 13%, the circuit was deemed adequate.

After the circuit was successfully tested, a second test with the full prototype was performed on an elm tree, a hardwood, and a spruce, a conifer. Due to no access to a professional tool, the prototypes readings were confirmed by a separate measurement with a multimeter. During the test, attachment methods were tested: The cases can be attached semi-permanent with double-sided adhesive tape. The prototype itself is light enough (~190g) to stay attached with this method. Alternatively, the case can be attached with zip ties or rope (see Appendix 11). This makes redeployment at a different log easier than adhesives.

---

[26] BTI - Hydrofix Holz

4.2.2 Positioning System

Two scenarios were tested for the location scan: long range (indoor and outdoor) and overlapping signal detection. The indoor test was performed in a room with seven to eight unrelated Bluetooth signals. These signals were likely from smartphones or notebooks as they were without an assigned UUID. The prototype itself was connected over a serial connection to a notebook. This made it possible to run the tests repeatedly without exceeding LoRaWAN duty cycles. Each scenario was run three times to ensure a high data quality.

For the long-range test, the sensor and a beacon were placed at opposites sides of the room. The distance was about 15 meters with some furniture (chairs & tables) in between. The sensor had no problem picking up the signal: With each scan, the expected UUID got returned. When additional beacons were added, the UUID of the closest was returned.

To determine the maximum range, the location scan was also performed outside in a public location. The sensor could reliably pick up a signal at a distance of up to 50 meters with a clear line of sight. Increasing the distance further lead to unreliable detection. At 75 meters only every third scan returned the expected UUID with the other scans returning NULL UUIDs.

The second test scenario – overlapping signals – was performed in the same room as above. Two beacons got placed near the sensor. The first beacon at a distance of about one meter, the second at about 20 centimeters (see Appendix 10). The sensor reliable returned the UUID of the beacon closest to itself.

To determine the minimum distance at which the sensor can distinguish between two beacons two beacons got placed at around eight- and ten-centimeters distance to the HM-10 antenna (see Appendix 10). The Bluetooth module could reliable distinguish the signals and always returned the closest. Moving them closer together, the sensor returned one of the UUIDs seemingly at random. As both beacons were measured with the same signal strength, the beacon’s UUID which got detected first was returned.

The scalability of this approach could not be tested efficiently as only four prototypes were built. To at least ensure the positioning method works in a small scale, all four devices, and eight beacons were deployed within a $40m^2$ room. Each prototype could pick up its nearest beacon and send a signal to the LoRa gateway.

4.2.3 Battery Life

On average, the prototype consumes around 20mA during its measurements and around 3mA during its sleep cycle. These values divided from the battery capacity of AA batteries (Table 4), leads to an expected runtime of 5 to 6 days of continuous load and 35 to 40 days of sleep time. The Arduino however needs a minimum of 2,4V to run. Batteries will probably fall below this required Voltage before their capacity is depleted. The voltage divider expects 9V for accurate measurements and calculations. It is expected that the voltage drop of the E type battery is negligible because the moisture measurement itself does not consume power.

To test the battery life, the prototype was connected to a multimeter. The multimeter measured the remaining battery power of the AA and E type batteries automatically every thirty minutes for 72 hours and added the value to a csv file. The AA type batteries were tested in three scenarios: continuous load without sleep, normal modus and sleep modus (Figure 30).

The E type battery was measured once during the continuous measurement scenario since the stress on the battery would be the highest. The Battery dropped by an expected, low amount of 0,02V.

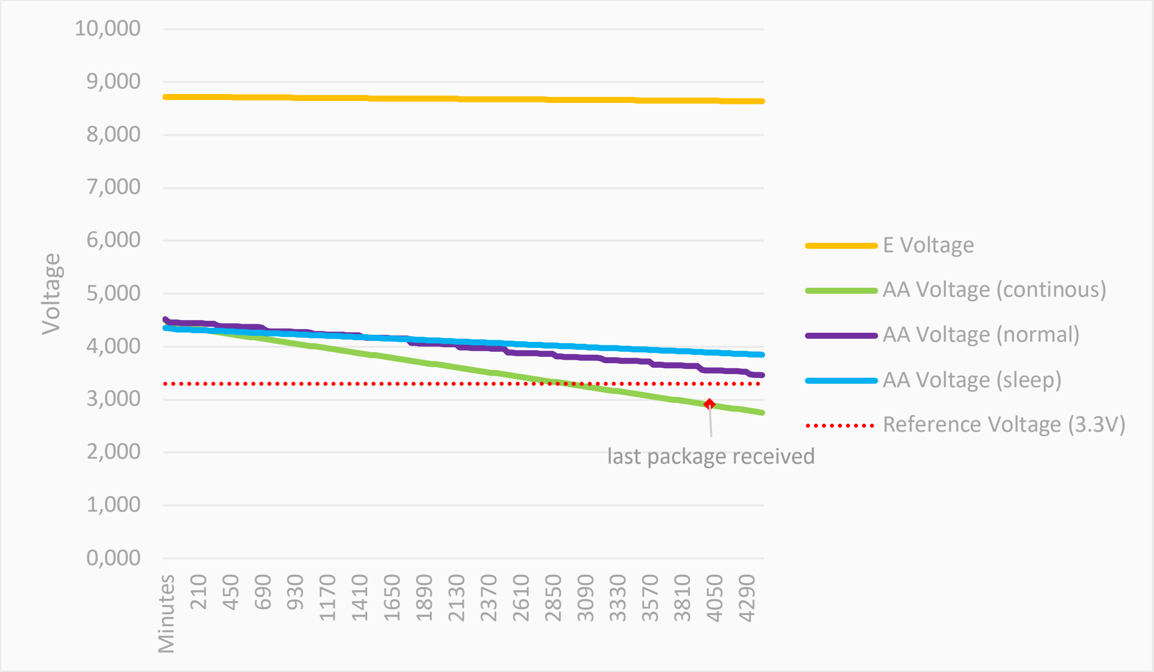


Figure 30: Battery voltage drop in use (own illustration)

For the continuous test, the six-hour low-power cycles got disabled. The prototype continuously scans for Bluetooth beacons, measures the moisture content and sends the collected data over LoRaWAN. During the continuous test, the voltage of the AA batteries dropped to below 2,9V after about 68 hours. This is below the specified recommended voltage of 3,3V. After 68 hours, the prototype did not send further messages to The Things Network. It likely did not have a high enough voltage to keep sending data.

The voltage drop of the AA batteries during the sleep scenario – a repeated loop of the low-power routine – is about 0,004V per hour. The voltage dropped by 0,5V during the 72 hours test. A fully charged AA array with an initial charge of 4,5V would reach the minimum viable voltage of 2,9V after around 10 days.

The normal scenario – a measurement followed by a six-hour sleep cycle – ran continuously for the 72 hours test. The voltage dropped by 0,3V – 0,4V per day. After 72 hours, the AA batteries had a remaining Voltage of 3,45V, a drop of around 1,1V. In this scenario, the prototype could run for around 5 days before dipping below 2,9V at which the LoRa radio cannot send data.

4.2.4 Web Application

To test the functionality of the web application, its features got tested to work as expected: Data should be received, stored in a database and be visualized. Further, the application should send out notifications if the moisture content reaches a certain level or the storage location changed.

Every time a data package was sent from a prototype, the package was successfully routed to the Node-RED environment. Each of these packages got also written to the database. The GUI refreshes automatically on changes.

If a wanted moisture level is set, the application sends out an email as soon as the measured value is less than the setup value. With the current implementation, this notification is repeated with every consecutive reading, e.g. every sixth hour. This notification behavior could be seen as excessive by the employees.

5 Summary and Conclusion

This chapter begins by recapitulating the findings in relation to the research questions described in section 1.2. Further the discovered limitations are stated (section 5.2), as well as future work which could built upon the findings of this thesis (section 0).

5.1 Research Questions

The research objections and questions stated in section 1.2, which are the foundation of this thesis, are answered in summary. Each object will have references to their corresponding section(s) in this thesis for a more elaborate answer.

RO1 Research existing approaches of IoT in Enterprises

Existing research for IoT and usage of IoT in enterprises is discussed in chapter 2 Theoretical Foundations. Research defining the term IoT is used to give an overview of the field (section 2.1). Further research of IoT usage in the industry is described in section 2.2. Literature of IoT usage is citied although the research focused on large enterprises.

RO2 Research possible use cases for a carpentry

During interviews with employees of the carpentry, a scenario was developed to use IoT to decrease manual labor (section 3.1). A sensor-based prototype should gather the moisture content of wood logs and keep track of these logs' locations. The wood moisture is determined by measuring the electrical resistance of a wood given wood log (section 3.2.1). The location is identified by scanning for existing Bluetooth signals and finding the strongest (closest) signal (section 3.2.2).

RO3 Research networking options for IoT devices

To send data over long ranges in a warehouse, LoRaWAN is chosen as a networking stack (section 3.3). Using LoRaWAN has the benefit to use an existing and extendable infrastructure. The Things Network, a routing system for things can be used, minimizing the work to design a custom backend system (section 3.3.2).

RO4 Develop a sensor-based prototype

The prototype is developed to be built quickly and cheaply using existing hardware modules. These components are chosen to fulfill the use cases of gathering wood moisture content (section 3.2) and positioning (section 3.3). An Arduino is selected as the controlling unit (section 3.4). To keep the used components safe, a case was designed to protect all components, while still being small enough to be used in the carpentry (section 3.5). Special attention has to be placed on the power source: a secure, non-flammable solution has to be used to reduce risks of fire hazards (section 3.4.2).

Further the hardware was assembled (section 4.1.1) and tested. The tests were performed to see if the stated use cases can be fulfilled (section 4.2).

RO5 Make sensor data accessible

To make the sensor data accessible and useful for the employees, a web application is designed (section 3.6.3). This application displays the wood logs moisture content and its development as a graph. Further it displays the location the log is stored in. The application can send notifications to employees if the moisture content reaches a specific value. To ensure the data can be checked without interrupting the existing workflows, the data is visualized in a responsive grid – making it accessible on mobile devices at the work floor. Further the applications functionality was tested (section 4.2.4).

Overall, the Research objectives could be fulfilled. It shows, that IoT can be used in SMEs and even a traditional craft as woodworking. The prototype has the potential to reduce the manual labor at the carpentry. It also makes knowledge more accessible, either by looking up the information or getting notified on a specific change.

5.2 Limitations

The prototype is mostly developed using prebuilt components. This is done to keep the cost of development lower than having custom made components. This however leads to an overall bigger size and higher power usage as some components come with additional features which are not needed to fulfill the tasks. Additionally, the prototyping case is rather fragile and not tolerant to 3D printing inaccuracies. The cases themselves have to be kept shut by adhesive tape or zip ties.

As stated in sections 3.2.1.2, the Arduino cannot measure a wide range of voltages and thus moisture contents. This is further confirmed during testing (section 0): The prototype can accurately measure wood moisture contents between 8% up to 13%, while a wood shop might need a wider range of 7% to 13% moisture content.

Battery life for the prototype is rather short at around 5 days (see section 4.2.3). The batteries have to be monitored and changed in regular intervals to keep the device running.

The prototypes register as APB devices at The Things Network. These devices have to be created at TTN and associated IDs have to be programmed to the corresponding prototype. This approach is chosen because the support of APB was more stable in the LMIC library used (section 3.6.1). The self-registering of devices did not work as reliably as having a hardcoded ID in the prototypes. This limitation hinders scalability as each device has to be registered and programmed manually.

A further limitation obstructing scalability is found in the web application. Each device displayed by the web application has to be created manually as Node-RED does not support programmatic creation of function nodes (see 3.6.3).

5.3 Future Work

As there is a lack of scientific papers regarding the usage of IoT in (small and medium) Enterprises, further research can be conducted in this domain. Researchers can determine how IoT can improve processes and reduce manual labor. They could create a framework to guide the introduction of this emerging technology into businesses.

Based on the limitations stated in section 5.2, the prototype can be improved and possibly mature to a product. Printed circuit boards and production-based circuits could reduce the size, improve stability and battery life. The prototype itself could be built as a more generic device with the ability to connect a variety of sensors.

References

Amon, F. *et al.* (2012) *Fire risks associated with batteries*. Borås.

Arduino (2014) *Arduino MICRO*, *Arduino Products*. Available at: https://store.arduino.cc/arduino-micro (Accessed: 8 August 2018).

Arduino (2016) *Arduino Pro Mini*, *Arduino Products*. Available at: https://store.arduino.cc/arduino-pro-mini (Accessed: 8 August 2018).

Ashton, K. (2010) 'That "Internet of Things" Thing', *RFID Journal*, p. 1. doi: 10.1145/2967977.

Atzori, L., Iera, A. and Morabito, G. (2010) 'The Internet of Things: A survey', *Computer Networks*. Elsevier B.V., 54(15), pp. 2787–2805. doi: 10.1016/j.comnet.2010.05.010.

Banks, D. *et al.* (2005) 'System and method for tracking the relative location of mobile stations belonging to a pre-defined group of mobile stations'. USA.

Barnaghi, P. *et al.* (2012) 'Semantics for the Internet of Things', *International Journal on Semantic Web and Information Systems*, 8(1), pp. 1–21. doi: 10.4018/jswis.2012010101.

Barr, J. (2015) *AWS IoT – Now Generally Available | AWS News Blog*, *AWS News Blog*. Available at: https://aws.amazon.com/de/blogs/aws/aws-iot-now-generally-available/ (Accessed: 8 April 2018).

Bauernhansl, T., Ten Hompel, M. and Vogel-Heuser, B. (2014) *Industrie 4.0 in produktion, automatisierung und logistik: anwendung, technologien und migration*. Springer.

Blanter, A. (2014) 'Silicon Valley Thought Leaders Series'. Available at: https://www.atkearney.com/documents/826061/826202/A.T.+Kearney_Internet+of+Things+2020+Presentation_Online.pdf/ef040ea5-97d3-729f-1fab-0b3010a188ad (Accessed: 8 April 2018).

Bloem, J. *et al.* (2014) 'The Fourth Industrial Revolution Things to Tighten the Link Between it and ot', *VINT research report*, pp. 1–39.

Bundesnetzagentur (2018) *Frequenzplan 2018*. Available at: https://www.bundesnetzagentur.de/SharedDocs/Downloads/DE/Sachgebiete/Telekommunikation/Unternehmen_Institutionen/Frequenzen/Frequenzplan.pdf?__blob=publicationFile&v=8 (Accessed: 27 June 2018).

Butner, K. and Lubowe, D. (2015) 'The Digital overhaul - Rethinkin manufacturing in the digital age'. doi: 10.4324/9780203935651.

Council, the N. I., Nic, N. and Intelligence, S. C. B. (2008) 'Disruptive Civil Technologies Six Technologies With Potential Impacts on US Interests Out to 2025', *National Intelligence Council*, (April), p. 48. doi: 10.1017/CBO9781107415324.004.

EFI, E. F. und I. (2016) 'Der Beitrag von KMU zu Forschung und Innovation in Deutschland', *EFI*. Available at: http://www.e-fi.de/fileadmin/Inhaltskapitel_2016/EFI_2016_B1.pdf (Accessed: 14 November 2017).

Evans, D. (2011) 'The Internet of Things How the Next Evolution of the Internet Is Changing Everything'. Available at: https://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf (Accessed: 13 December 2017).

Fleisch, E. (2010) 'What is the Internet of Things? An Economic Perspective', *Auto-ID Labs White Paper WP-BIZAPP-053*, pp. 1–27. doi: 10.1109/MCOM.2013.6476854.

Gartner (2013) *Gartner ' s 2013 Hype Cycle for Emerging Technologies Maps Out Evolving Relations ... Newsroom Gartner ' s 2013 Hype Cycle for Emerging Technologies Maps Out Evolving Relationship Between Humans and Machines Gartner ' s 2013 Hype Cycle for Emerging Techno*, *gartner.com*. Available at: https://www.gartner.com/newsroom/id/2575515 (Accessed: 12 January 2018).

Gartner (2014) *Gartner ' s 2014 Hype Cycle for Emerging Technologies Maps the Journey to Digital Business*, *gartner.com*. doi: http://www.gartner.com/newsroom/id/2819918.

Gartner (2016) *Gartner's 2016 Hype Cycle for Emerging Technologies Identifies Three Key Trends That Organizations Must Track to Gain Competitive Advantage*, *gartner.com*. Available at: https://www.gartner.com/newsroom/id/3412017 (Accessed: 12 January 2018).

Gartner (2017) *Top Trends from the Gartner Hype Cycle for Midsize Enterprises, 2017 - Smarter With Gartner*, *gartner.com*. Available at: https://www.gartner.com/smarterwithgartner/top-trends-in-the-gartner-hype-cycle-for-emerging-technologies-2017/ (Accessed: 28 March 2018).

Gartner (2018) *5 Trends Emerge in the Gartner Hype Cycle for Emerging Technologies , 2018*, *gartner.com*. Available at: https://www.gartner.com/smarterwithgartner/5-trends-emerge-in-gartner-hype-cycle-for-emerging-technologies-2018/ (Accessed: 20 August 2018).

Gubbi, J. *et al.* (2013) 'Internet of Things (IoT): A vision, architectural elements, and future directions', *Future Generation Computer Systems*, 29(7), pp. 1645–1660. doi: https://doi.org/10.1016/j.future.2013.01.010.

Hope Microelectronics Co. (2014) 'Datasheet: RFM95/96/97/98(W) v1.0', p. 121. Available at: http://www.hoperf.com/rf_transceiver/lora/RFM95W.html%5Cnhttp://www.hoperf.com/upload/rf/RFM95_96_97_98W.pdf.

Huber, D. and Kaiser, T. (2015) 'Wie das Internet der Dinge neue Geschäftsmodelle ermöglicht', *HMD Praxis der Wirtschaftsinformatik*. Springer, 52(5), pp. 681–689.

Huber, W. (2016) 'Industrie 4.0 in der Automobilproduktion', *Ein Praxisbuch: Springer Vieweg Verlag. Online verfügbar unter http://dx. doi. org/10.1007/978-3-658-12732-9*. Springer.

Hull, B., Jamieson, K. and Balakrishnan, H. (2004) 'Mitigating Congestion in Wireless Sensor Networks', in *Proceedings of the 2Nd International Conference on Embedded Networked Sensor Systems*. New York, NY, USA: ACM (SenSys '04), pp. 134–147. doi: 10.1145/1031495.1031512.

Hunke, N. *et al.* (2017) *Winning in IoT: It's All About the Business Processes, Boston Consulting Group*. Available at: https://www.bcgperspectives.com/content/articles/hardware-software-energy-environment-winning-in-iot-all-about-winning-processes/ (Accessed: 14 November 2017).

International Telecommunication Union (1994) 'X.200: Data Networks and open system communications', 4, p. 59. Available at: https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-X.200-199407-I!!PDF-E&type=items.

International Telecommunication Union (2012) 'Overview of the Internet of things', *Series Y: Global information infrastructure, internet protocol aspects and next-generation networks - Frameworks and functional architecture models*, p. 22.

ITU (2005) 'The Internet of Things', *Itu Internet Report 2005*, p. 212. doi: 10.2139/ssrn.2324902.

James, W. L. (1988) 'Electronics Moisture Meters for Wood', *General Technical Report*, (FPL-GTR-6), p. 19. Available at: https://www.fpl.fs.fed.us/documnts/fplgtr/fplgtr06.pdf.

JNHuaMao Technology Company (2014) 'HM Bluetooth module datasheet'. Shandong.

Kjaergaard, M. (2012) 'Location-based services on mobile phones: minimizing power consumption', *IEEE Pervasive Computing*, 11(1), pp. 67–73. doi: 10.1109/MPRV.2010.47.

Lasi, H. *et al.* (2014) 'Industry 4.0', *Business & Information Systems Engineering*. Springer, 6(4), pp. 239–242.

Leminen, S. *et al.* (2012) 'Towards IOT Ecosystems and Business Models', in Andreev, S., Balandin, S., and Koucheryavy, Y. (eds) *Internet of Things, Smart Spaces, and Next Generation Networking*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 15–26.

Löffler, M. and Tschiesner, A. (2013) *The Internet of Things and the future of manufacturing | McKinsey Company*. Available at: https://www.mckinsey.com/business-functions/digital-mckinsey/our-insights/the-internet-of-things-and-the-future-of-manufacturing (Accessed: 8 November 2017).

LoRa Alliance (2017) 'LoRaWAN 1.1 Specification', *LoRa Alliance*. Beaverton, p. 101.

M2M (2018) *Tarifübersicht*. Available at: https://www.m2m-mobil.de/m2m-mobilfunk-tarif-preisinformationen (Accessed: 23 March 2018).

Madakam, S., Ramaswamy, S. and Tripathi, R. (2015) ‘Internet of Things (IoT): A Literature Review’, *Journal of Computer and Communications*, 3, pp. 164–173. doi: 10.4236/jcc.2015.35021.

Mineraud, J. *et al.* (2016) ‘A gap analysis of Internet-of-Things platforms’, *Computer Communications*. Elsevier, 89–90, pp. 5–16. doi: 10.1016/J.COMCOM.2016.03.015.

Niemz, P. *et al.* (2007) ‘Grundlagen der Holzbe-und Verarbeitung’. ETH, Eidgenössische Technische Hochschule Zürich, Institut für Baustoffe IfB.

Normenausschuss Holzwirtschaft und Möbel (NHM) im DIN, B. 4. (2002) ‘DIN EN 13183-2 Feuchtegehalt eines Stückes Schnittholz - Teil 2: Schätzung durch elektrisches Widerstands-Messverfahren; Deutsche Fassung EN 13183-2:2002’. Berlin: Beuth Verlag.

Patwari, N. *et al.* (2003) ‘Relative location estimation in wireless sensor networks’, *IEEE Transactions on Signal Processing*, 51(8), pp. 2137–2148. doi: 10.1109/TSP.2003.814469.

Raspberry pi foundation (2014) *Raspberry Pi Documentation*, *3*. Available at: https://www.raspberrypi.org/documentation/hardware/raspberrypi/README.md (Accessed: 8 August 2018).

Roman, R., Zhou, J. and Lopez, J. (2013) ‘On the features and challenges of security and privacy in distributed internet of things’, *Computer Networks*. Elsevier, 57(10), pp. 2266–2279. doi: 10.1016/J.COMNET.2012.12.018.

Roth, A. (2016) *Einführung und Umsetzung von Industrie 4.0: Grundlagen, Vorgehensmodell und Use Cases aus der Praxis*. Springer-Verlag.

Roy, R. *et al.* (2016) ‘Continuous maintenance and the future – Foundations and technological challenges’, *CIRP Annals - Manufacturing Technology*. CIRP, 65(2), pp. 667–688. doi: 10.1016/j.cirp.2016.06.006.

Schwab, K. (2016) *The Fourth Industrial Revolution*. Geneva: World Economic Forum.

Shenzhen Minew Technologies Co. Ltd. (2014) ‘i3 MiniBeacon’. Shenzen: Shenzhen Minew Technologies Co. Ltd.

Shrouf, F., Ordieres, J. and Miragliotta, G. (2014) ‘Smart factories in Industry 4.0: A review of the concept and of energy management approached in production based on the Internet of Things paradigm’, in *2014 IEEE International Conference on Industrial Engineering and Engineering Management*, pp. 697–701. doi: 10.1109/IEEM.2014.7058728.

Söllner, R. (2014) ‘Die wirtschaftliche Bedeutung kleiner und mittlerer Unternehmen in Deutschland’, *WISTA – Wirtschaft und Statistik (01/2014)*, pp. 50–51. Available at: https://www.destatis.de/DE/Publikationen/WirtschaftStatistik/UnternehmenGewerbeanzeigen/BedeutungKleinerMittlererUnternehmen_12014.pdf?__blob=publicationFile (Accessed: 7 November 2017).

Sommer, L. (2015) ‘Industrial revolution - Industry 4.0: Are German manufacturing SMEs the first victims of this revolution?’, *Journal of Industrial Engineering and Management*, 8(5), pp. 1512–1532. doi: 10.3926/jiem.1470.

StackOverflow (2018) *Stack Overflow Developer Survey 2018 - Most Popular Databases*. Available at: https://insights.stackoverflow.com/survey/2018/#technology-databases (Accessed: 15 September 2018).

Takeda, H. *et al.* (1990) ‘Modeling Design Processes’, *AI Magazine*, 11(4), pp. 37–48. doi: 10.1609/aimag.v11i4.855.

Telekom (2018) *IoT Datentarif - Telekom IoT-Shop*. Available at: https://iot.telekom.com/shop/konnektivitaet/iot-datentarif.html (Accessed: 23 March 2018).

Texas Instruments (2014) ‘LMx93-N , LM2903-N Low-Power , Low-Offset Voltage , Dual Comparators’. Dallas, pp. 1–20. Available at: http://www.ti.com/lit/ds/symlink/lm393-n.pdf.

TTN Network Architecture (2018) *The Things Network*. Available at:

https://www.thethingsnetwork.org/docs/devices/node/.

Uckelmann, D., Harrison, M. and Michahelles, F. (2011) *Architecting the Internet of Things*. doi: 10.1007/978-3-642-19157-2.

Vaishnavi, V. K. and Kuechler, W. (2008) *Design science research methods and patterns: innovating information and communication technology*, *Order A Journal On The Theory Of Ordered Sets And Its Applications*. doi: 10.1201/9781420059335.

Ven, K., Verelst, J. and Mannaert, H. (2008) 'Should you adopt open source software?', *IEEE Software*, 25(3), pp. 54–59. doi: 10.1109/MS.2008.73.

White, J. (2018) *Microsoft will invest $5 billion in IoT. Here's why. - Internet of Things*, *Microsoft Blog*. Available at: https://blogs.microsoft.com/iot/2018/04/04/microsoft-will-invest-5-billion-in-iot-heres-why/ (Accessed: 8 April 2018).

Winter, M. and Brodd, R. J. (2004) 'What Are Batteries, Fuel Cells, and Supercapacitors?', *Chemical Reviews*. American Chemical Society, 104(10), pp. 4245–4270. doi: 10.1021/cr020730k.

World Economic Forum (2018) *The Future of Jobs Report*, *WEF 2018*. Available at: http://journals.sagepub.com/doi/10.1177/1946756712473437.

Wortmann, F. and Flüchter, K. (2015) 'Internet of Things', *Business & Information Systems Engineering*, 57(3), pp. 221–224. doi: 10.1007/s12599-015-0383-3.

Yang, D.-L., Liu, F. and Liang, Y.-D. (2010) 'A Survey of the Internet of Things', *Proceedings of the 1st International Conference on E-Business Intelligence (ICEBI2010)*. Elsevier B.V., 54(15), pp. 358–366. doi: 10.1016/j.comnet.2010.05.010.

Zhang, Y. *et al.* (2016) 'IoT-enabled real-time production performance analysis and exception diagnosis model', *IEEE Transactions on Automation Science and Engineering*. IEEE, 13(3), pp. 1318–1332.

Zuehlke, D. (2010) 'SmartFactory-Towards a factory-of-things', *Annual Reviews in Control*, 34(1), pp. 129–138. doi: 10.1016/j.arcontrol.2010.02.008.

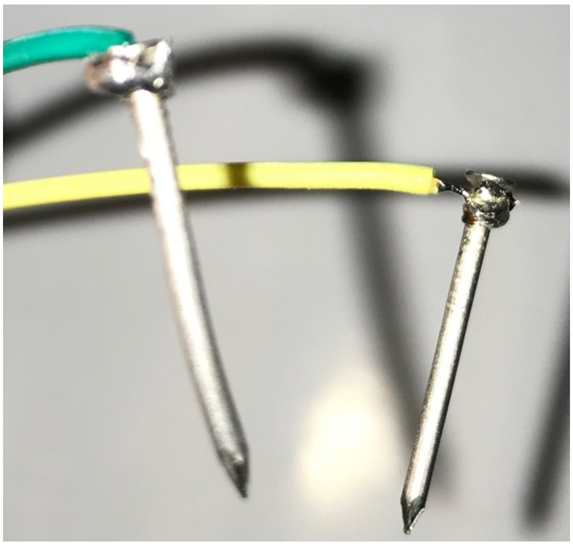
Appendix

Appendix 1 Source Code of the Prototype

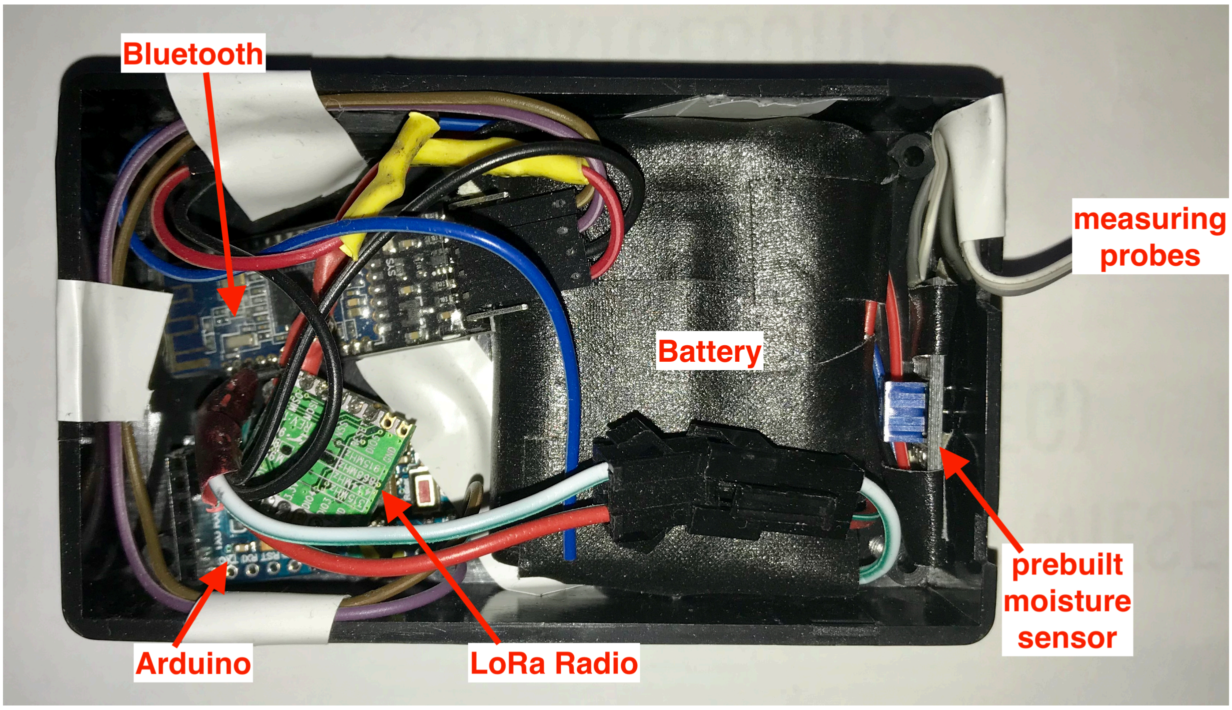
The source code of the prototype developed in chapter 3 is available at the following addresses. The Prototype code is in C++ and can be flashed using the Arduino IDE. The TTN Code is the used decoder in written in JavaScript. The Node-RED flow is a json file which can be imported into Node-RED. The Docker environment is a docker-compose file to setup a basic Node-RED environment. Each project includes a README file with a short description.

Arduino Prototype	https://gitlab.uni-koblenz.de/FGEIM/eot_ba_willems_smart_carpenter/arduino_prototype.git
TTN Decoder	https://gitlab.uni-koblenz.de/FGEIM/eot_ba_willems_smart_carpenter/TTN_payload_decoder.git
Node-RED flow	https://gitlab.uni-koblenz.de/FGEIM/eot_ba_willems_smart_carpenter/node-red-flow.git
Docker	https://gitlab.uni-koblenz.de/FGEIM/eot_ba_willems_smart_carpenter/docker-environment.git

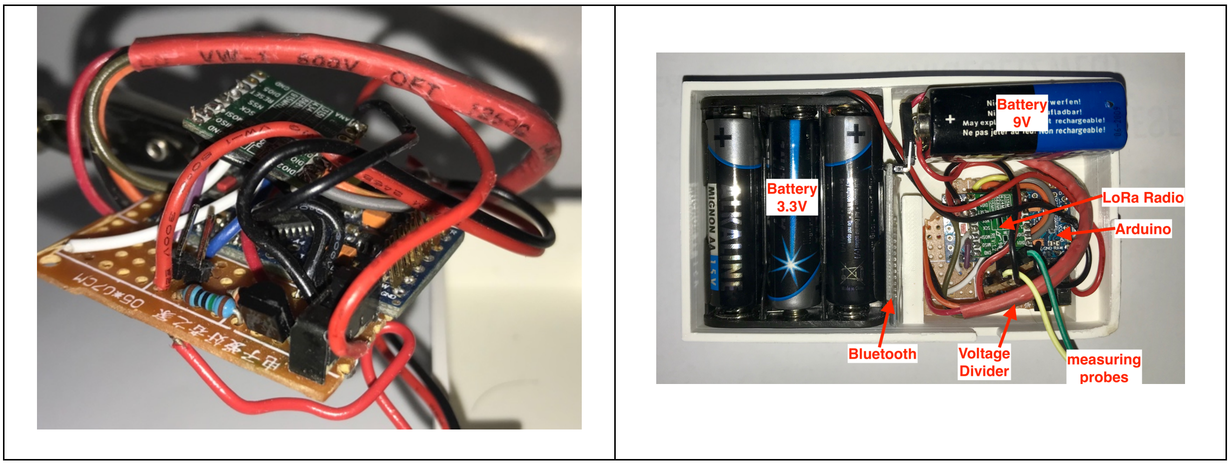
Appendix 2 Pictures Build Process



Appendix 1: Moisture Measurement probes (own illustration)

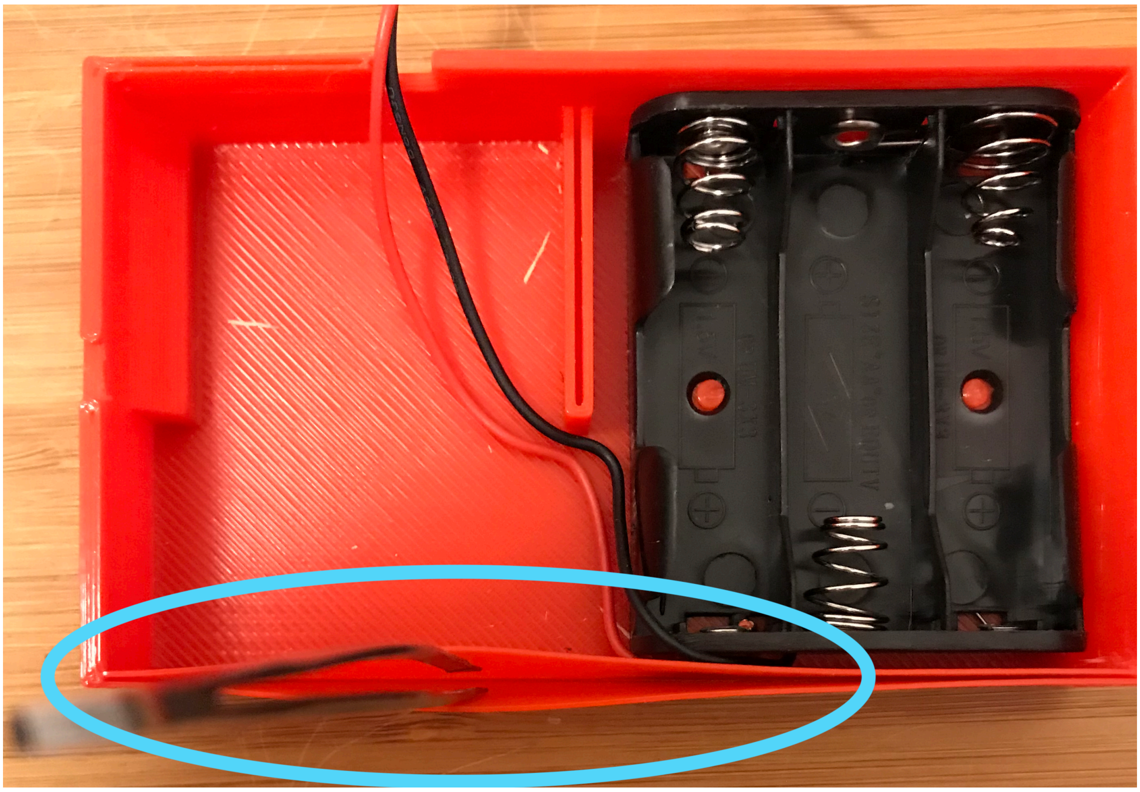


Appendix 2: First Hardware Prototype (own illustration)



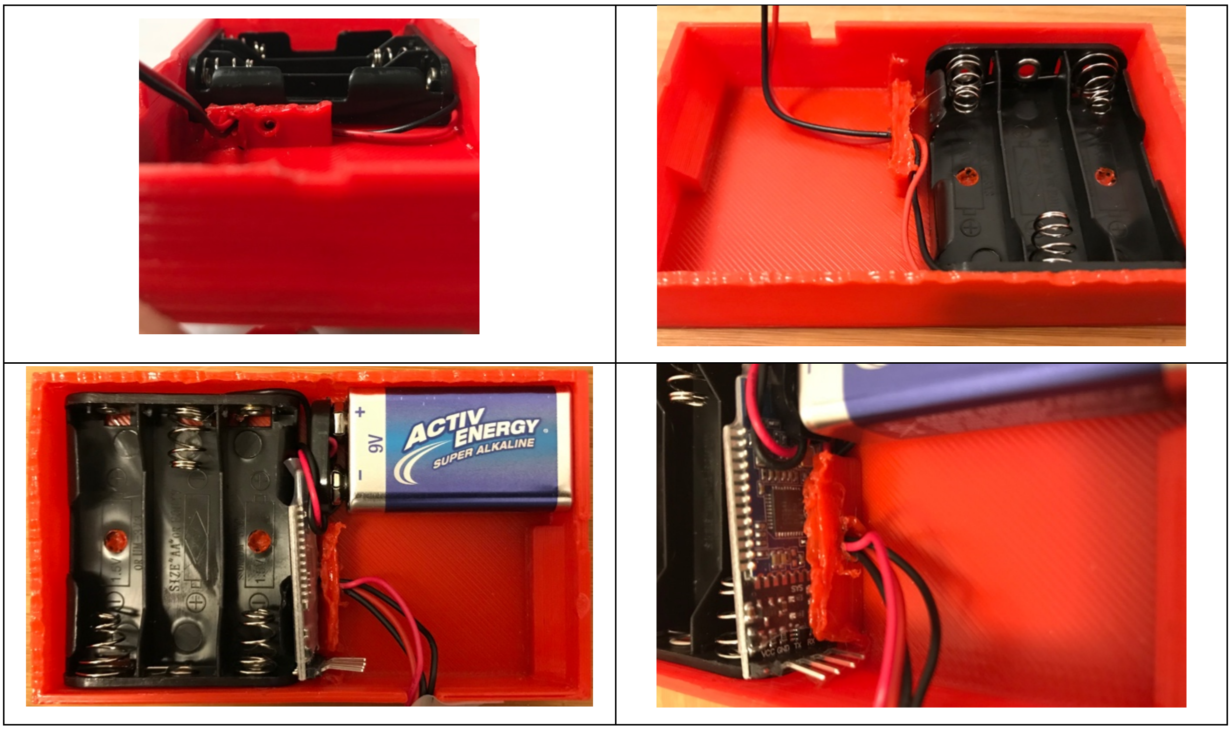
Appendix 3: Second Hardware Prototype – breadboard and case (own illustration)

Appendix 4: Second Hardware Prototype - semi closed (own illustration)

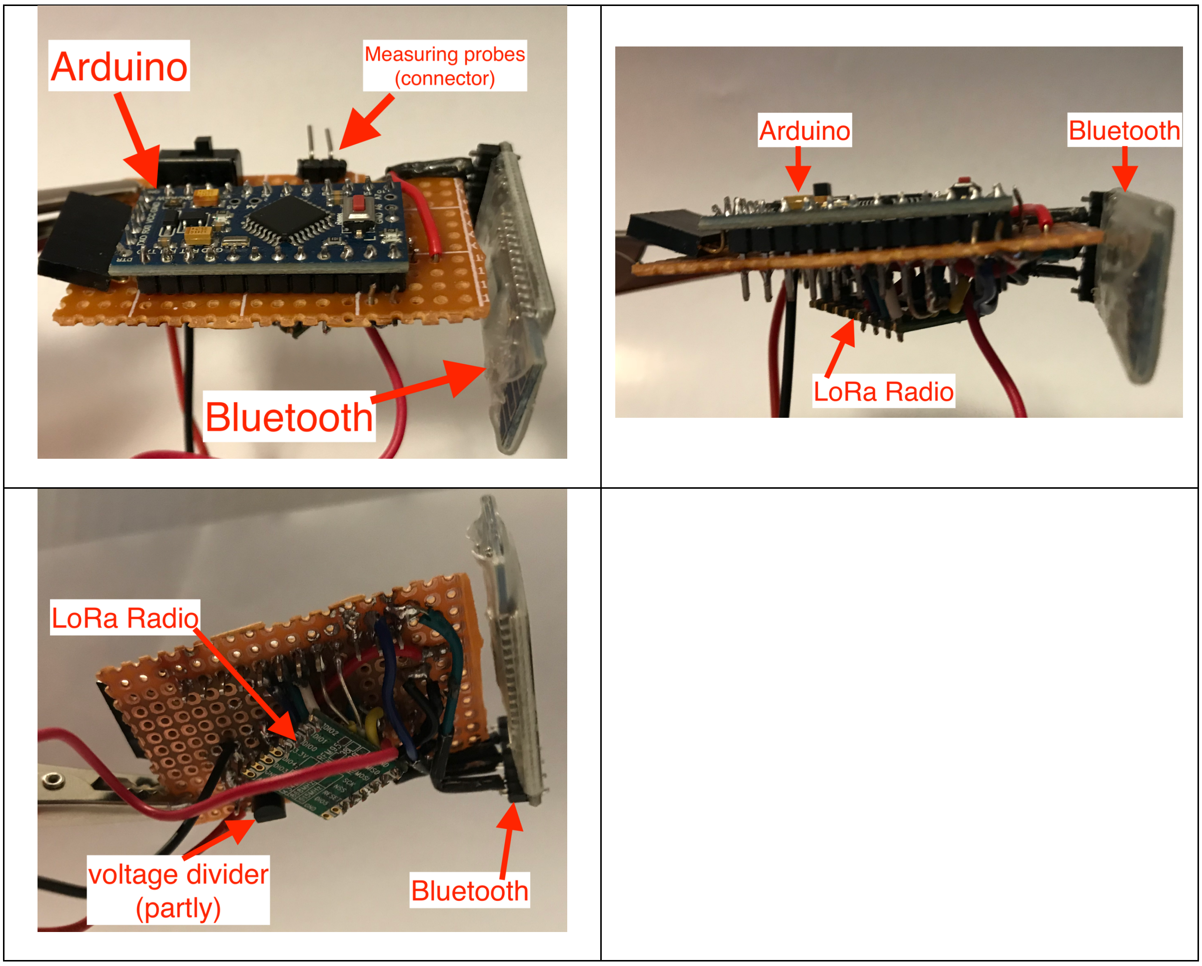


Appendix 5: Third Hardware Prototype - fragility (own illustration)

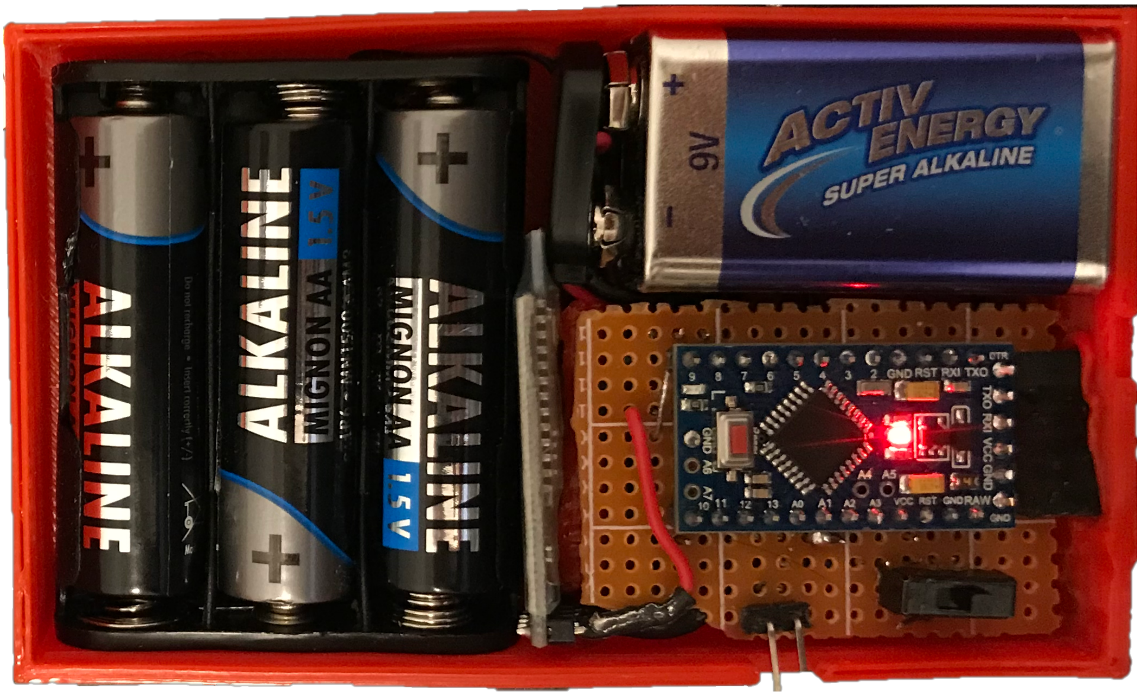
Appendix 6: Third Hardware Prototype - broken wall (own illustration)



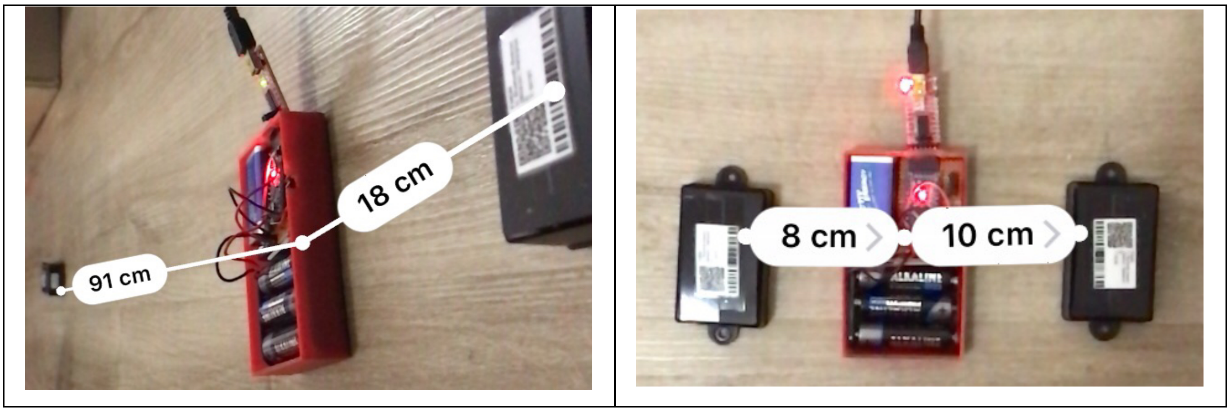
Appendix 7: Third Hardware Prototype - modifications case (own illustration)



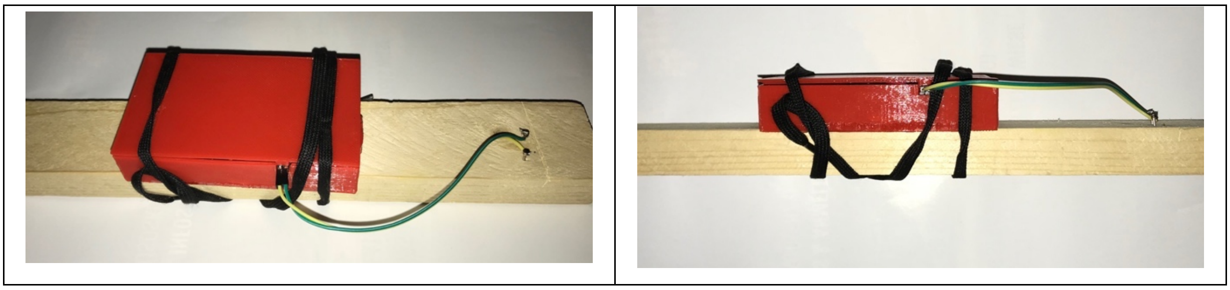
Appendix 8: Third Hardware Prototype - Breadboard (top | middle | bottom) (own illustration)



Appendix 9: Third Hardware Prototype - Assembled (own illustration)



Appendix 10: Test Location Scan - overlapping signals (own illustration)



Appendix 11: Example of Prototype attachment with rope (own illustration)