University of Zurich[UZH]

BACHELOR THESIS – Communication Systems Group, Prof. Dr. Burkhard Stiller

MTT: My Thesis Title

Name Lastname
Zurich, Switzerland
Student ID: TBD

Supervisor: Name Lastname, Name Lastname, Prof. Dr. Burkhard Stiller
Date of Submission: Month DD, YYYY

University of Zurich
Department of Informatics (IFI)
Binzmühlestrasse 14, CH-8050 Zürich, Switzerland

ifi

Bachelor Thesis
Communication Systems Group (CSG)
Department of Informatics (IFI)
University of Zurich
Binzmuehlestrasse 14, CH-8050 Zurich, Switzerland
URL: http://www.csg.uzh.ch/

Declaration of Independence

I hereby declare that I have composed this work independently and without the use of any aids other than those declared (including generative AI such as ChatGPT). I am aware that I take full responsibility for the scientific character of the submitted text myself, even if AI aids were used and declared (after written confirmation by the supervising professor). All passages taken verbatim or in sense from published or unpublished writings are identified as such. The work has not yet been submitted in the same or similar form or in excerpts as part of another examination.

Zürich, _____

Signature of student

Abstract

Acknowledgments

Contents

Chapter 1

Introduction

Research questions:

- What are the advantages and disadvantages to using UWB over BLE for fast network communication.
- Can I build a system of sensors that detectes disturbances in transportation and alerts the user.
- What is the most adequate network topology to be used in such a system.

1.1 Motivation

Certify is an international cooperative project between twelve partners situated in Switzerland and the EU One of those partners is the university of zurich. Its focus is on the development of Internet of Things (IoT) systems for security, monitoring and detection Next to certifications and the development of frameworks, Certify also consirnes itself with the integration of IoT devices.

One of multiple currently running pilots of Certify is the "Tracking and monitoring of artworks" The goal of this pilot is to enable the constant tracking and monitoring of artworks by attaching a device to it. This device allows for unique identification, by using cryptographic methods. It is also intended to act as a cluster of sensors that collect information about the surounding of the artwork that are relevant to the wellbeeing of the artwork. The goal is to have constant data on the artwork throughout its lifecycle. This is intended to help with securing the artwork and helping with chain of custody monitoring.

1.2 Thesis Goals

The goal of this project is to develop a system that implements a localized version of the artwork tracking envisioned by the Certify project, ment for transportation in a truck.

Additionally the system will extend the Certify Projects goal by adding new detection methods and also informs the driver of the truck about potential problems.

The goal is to develop a system that tracks the state of artwork in a truck using different detection methods. The devices attached to the artworks, called tags, build a local decentralized network. The phone of the driver can query the network and displayed the collected metrics to them. If a metric is outside of the accepted norm, the system should allert the driver.

This thesis presents a proof of concept implementation. The used metrics are not intended to be a full representation of needed sensors to securley Transport art. Rather they are intended to show different types of sensors that can be used. This thesis assumes that the data transfer to a server using 4G, as planned by the Certify project, will work and will not implement it in this thesis.

1.3 Methodology

This Thesis was made in four stages:

Reserach

In a first step, the basis of the thesis had to be researched. This involved familiarizing with existing research on the toping of artwork tracking, local IoT networks and commonly used communication protocols. Existing artwork tracking methods need to be analyzed and evaluated, considering their strength and shortcommings during the transportation in a truck. The types of sensors that could be relevant need to be chosen, based on existing research, cost and avaliablity. Options for the network-architecture inside the truck needed to be researched and compared, based on performance, stability and security. A communication protocol needed to be chosen, based on the same criterias.

Design

Once the fundamental knowledge for the project had been aquired, the system had to be designed. The design was chosen based on feasability, security and stability.

Implementation

The desig then was implemented in a simplified manner based on the material that was avaliable. For this four tags were build, equiped with sensors, communication-capabilities and power suply. Then the required software was written, using existing implementations when possible and writing new code when required. A simple example app was also developed, based on an existing communications app published by Nordic Semiconductors and installed on a phone.

Evaluation

The developed system of tags and phone was tested in a series of five experiments. The first four experiments were intended to caputre a specific part of the system, while the last was a general purpose test. The tests were performed in a manner that insured minimal external influence. The resulting data from the testswere analyzed using statistical methods. The goal was to determin the reliability of the system, find limitations and look for emprovements.

1.4 Thesis Outline

Chapter 2

Fundamentals

2.1 Background

2.1.1 Ultra Wideband

IEEE

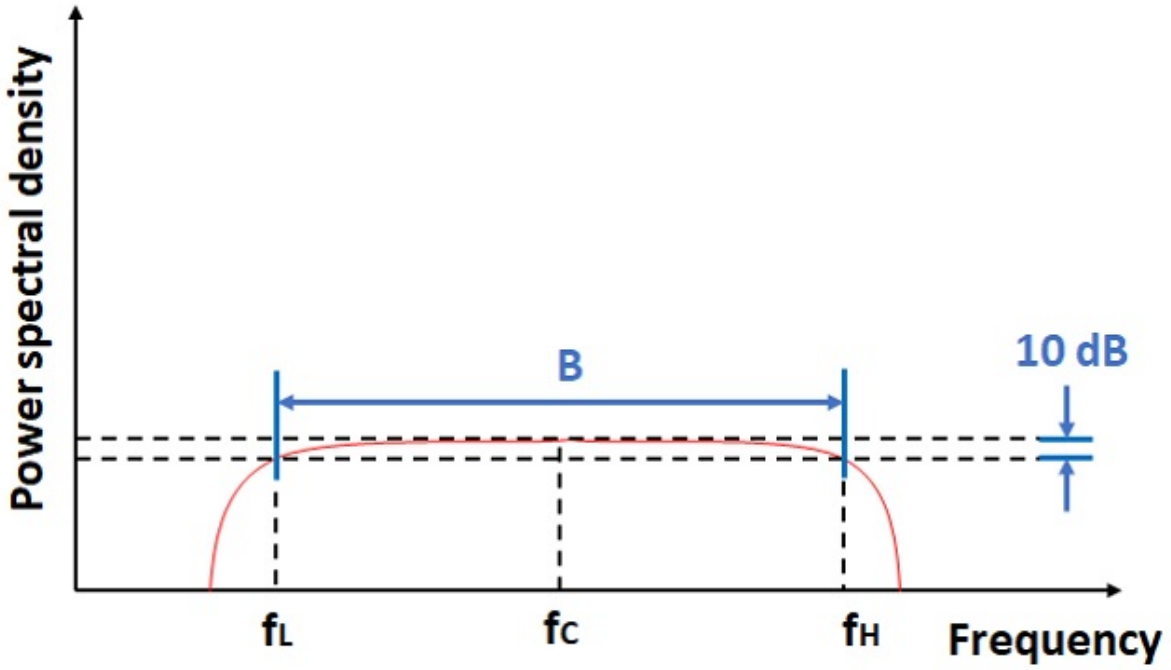
The Ultra Wideband (UWB) communication protocoll was introduced in 2003 by the Institute of Electrical and Electronics Engineers (IEEE) as part of the IEEE 802.15.4 standard. In 2020 updates were made to the protocoll when the IEEE 802.15.4z-2020 standard made improvements to the PHY layers of UWB connections. It achieved this by introducing a more robust timestamping system on the PHY layer. This is suplemented by changes to the MAC layer, that allow for the exchange of ranging information. The result is short frames, that are transmitted fast, between devices, leading to short bursts of communications that are fast, secure and ideal for ranging.

UWB works by using short radio frequency pulses, resulting in a large bandwidth. UWB is a lower power communication form. This prevents it from interfering with other communication forms it is sharing its wavelength with, such as WLAN or Bluetooth. Since UWB uses very short, distinc pulses over a short range, it has found use in ranging systems. UWB is split into high rate pulse (HRP) UWB and low rate pulse (LRP) UWB. Since ranging is part of this work and LRP is generally not used for ranging, I will not discuss it further in this thesis. Since UWB devices tend to be small and have a low energy consumption, in combination with the capability of ranging as well as data transfer, they have become popular as Internet of Things (IoT) devices.

The standard defines the PHY and MAC layer as well as frequency bands for communication. The 4z expantion tries to integrate UWB into the the WPAN standard. In Section ... and ... I will discuss the PHY and MAC layer.

The sending devices emits pulses in a pre-set band of frequencies, using short bursts to transmit the bits. The signal forms a concave curve in this band, where the two points that are 10 db below the maximum power spectral density are called the lower- and upper-frequency point, see figure 2.1. These two points must at least 500 Hz apart. The maximum power spectral density must be below the noise level. This proces prevents confilcts with other communications, that use a single frequency with a high power spectral density and modulate signal transmission, such as WI-FI or Bluetooth. The UWB protocoll has the added benefit of beeing usefull for high accuracy localization.

Figure 2.1: Power Spectral Density: Bandwidth B, lower-frequency f_L, upper-frequency f_H, [1]



UWB supported Nodes

The IEEE 802.15.4 standard distinquishes between two types of devices. Full-function device (FFD) are capable of connecting to multiple other devices, receive, transmit and coordinate. Reduced-function device (RFD) on the other hand can only connect to one other device and act as worker. In Topological terms RFDs can only operate as leaves, while FFDs can be any node in a network, including leaves. RFDs therefore are strictly worse, but make up for it by requiering fewer resources, such as memory and power. When FFDs work as PAN coordinators, they can use short adresses to address any node. The PAN also has a PAN identifier, to help communication accross multiple networks, while still using the short address. Each device also has a extended address, that is not assigned by any coordinator, that serves as a universal unique identifier (UUID).

2.1.2 UWB MAC

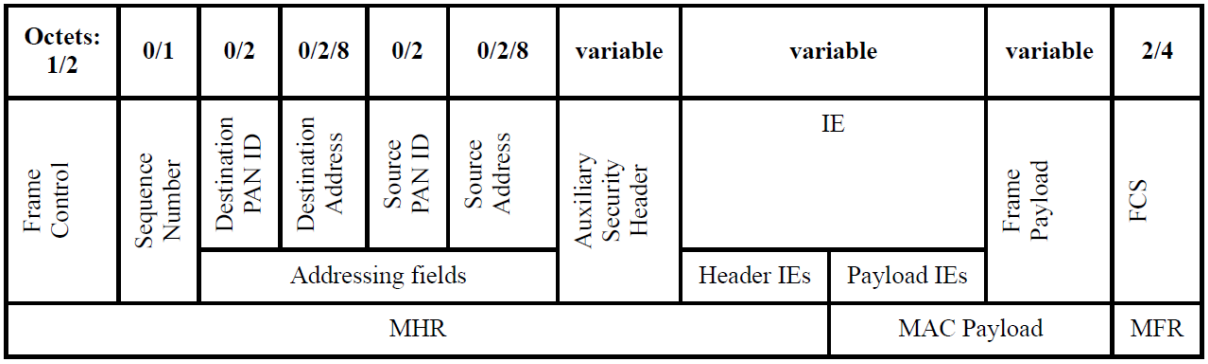


Figure 2.2: General MAC Frame Format [2]

The Mac Layer is part of the Data link layer. The Mac Frame is the payload of the PHY frame. It carries information about the type of frame, frame-format, security mechanism, adressing and frame validation. The Mac Layer additionaly provides rules for beacon managment, chanel access.

MAC Frame Format

Figure 2.2 shows the composition of a UWB-MAC frame.

In the MAC header (MHR), the Frame Control Field includes information about:

- the frame-type
- if the Auxiliary Security Header Field is used and in what capacity
- if additional frames will follow
- if an acknowledgment message is expected
- if the message is between different PAN-Networks.
- of what type the receiver is (PAN coordinator, device, PAN-Network)
- the used frame-format standard
- where to find the source address

The Sequence Number counts up, helping to keep track of the order in which frames arive. The Adressing Fields carry the IDs of sender and recipient for the frame. The Auxiliary Security Header Field only exists if it was specified in the control Field. It contains additional information needed for the chosen security methode.

There are two parts to the information element (IE). The header IE specifies additional information about the frame, for example data formating information or chanel time

allocation. The the payload IE specifies the length and data-type of the payload field. The payload containes the data that is sent. It and the IE are of variable length, depending of the frame-type and data-length.

The MAC footer (MFR) marcs the end of the frame. It only contains the frame checking sequence (FCS), that can be used to detect corrupted frames using cyclic redundancy checks.

2.1.3 UWB PHY

PHY Chanel

The IEEE 802.15.4z amendment defines 16 channels for communication for HRP UWB. A chanel is defined by its center frequency. UWB devices can transmit on three different bands, high band, low band and sub-gigahertz. For each band their is one chanel that is mandatory to support, if a device suports the band. The other channels are optional, but if two devices want to communicate with each other they need to use the same band. The bands, 16 chanels and their ranges and which chanels are mandatory can be found in table (see table 2.1).

Channel number	**Center frequency (MHz)**	**HRP UWB band**	**Mandatory**	
0	499.2	sub-gigahertz	✓	
1	3494.4	Low band		
2	3993.6			
3	4492.8		✓	
4	3993.6			
5	6489.6	High band		
6	6988.8			
7	6489.6			
8	7488			
9	7987.2		✓	
10	8486.4			
11	7987.2			
12	8985.6			
13	9484.8			
14	9984			
15	9484.8			

Table 2.1: HRP UWB Frequency and Channel Assignments [2, 3]

Scrambled timestamp sequence

The 4z amendment added the option to include a scrambled timestamp sequence (STS) into the frame. The STS is a cyphered sequence that includes the timestamp and is used for ranging. It is ment to increase the accuracy and integrtity of the raging results.

Before transmition receiver and sender exchange a randomly generated key. The key is then used to encrypt the timestamp using the advanced encryption standard (AES) with 128 bits. This ensured that the signal has not been intersepted and changed, to manipulate the ranging result. Devices that support STS are called HRP-enhanced ranging capable devices (HRP-ERDEV).

Pulse Repetition Frequency

The pulse repetition frequency (PRF) is the frequency at wich bursts are sent by the transmitter. The mean PRF is the average PRF while sending the payload (power switching service data unit PSDU). The higher the mean PRF, the shorter the airtime of each frame and allows for faster communication. HRP-ERDEV use a different mean PRF than general devices. They can work in Base pulse repetition frequency (BPRF) operating at mean PRF 64 MHz or in higher pulse repetition frequency (HPRF) mode operating above BPRF (Table 2.2).

Standard	**HRP UWB mode**	**mean PRF**
802.15.4	Non HRP ERDEV	3.9 MHz, 15.6 MHz, 62.4 MHz
802.15.4z	HRP-ERDEV BPRF	62.4 MHz
	HRP-ERDEV HPRF	124.8 MHz, 249.6 MHz

Table 2.2: HRP UWB Mean PRF (Based on IEEE 802.15.4 and IEEE 802.15.4z, [2, 3])

Symbol Encoding

UWB sends symbols by transmitting a burst of pulses that encode the symbol. Since the pulses have clean edges, the arival time can be measured precisly. This leads to the burst having two ways to carry information([4]):

- Binary phase-shift keying (BPSK: Encoding zeros and ones shifting the pulses phases so the burst beak for one has an oposite amplitude to the other. Figure 2.3 shows the singal 101 binary phase-shift keyed. Each bit is set twice, to detect problems with transmission.
- Burst position modulation (BPM): Changing the timing of the burst so it falls into a different time-slot inside of the possible burst position. Figure 2.4 shows how the burst can be placed in a BPM-interval. The burst can't be placed in the guard interval. The guard exists to minimize inter-symbol interference from the signals taking multiple paths.

One or both of these encoding-strategies can be used in a uwb transmission. The poition of the pulses inside of the burst (see figure 2.3) relative to each other can be used to detect the presence of multipath effects and ajust for them. Using this, precises arival times for the wohle signal can be calculated.

Non-HRP ERDEV use BPM and BPSK. Some HRP-ERDEV can use only BPSK, using a higher PRF and therefore reducing airtime.

Figure 2.3: UWB signal transmission byte encoding, [4]

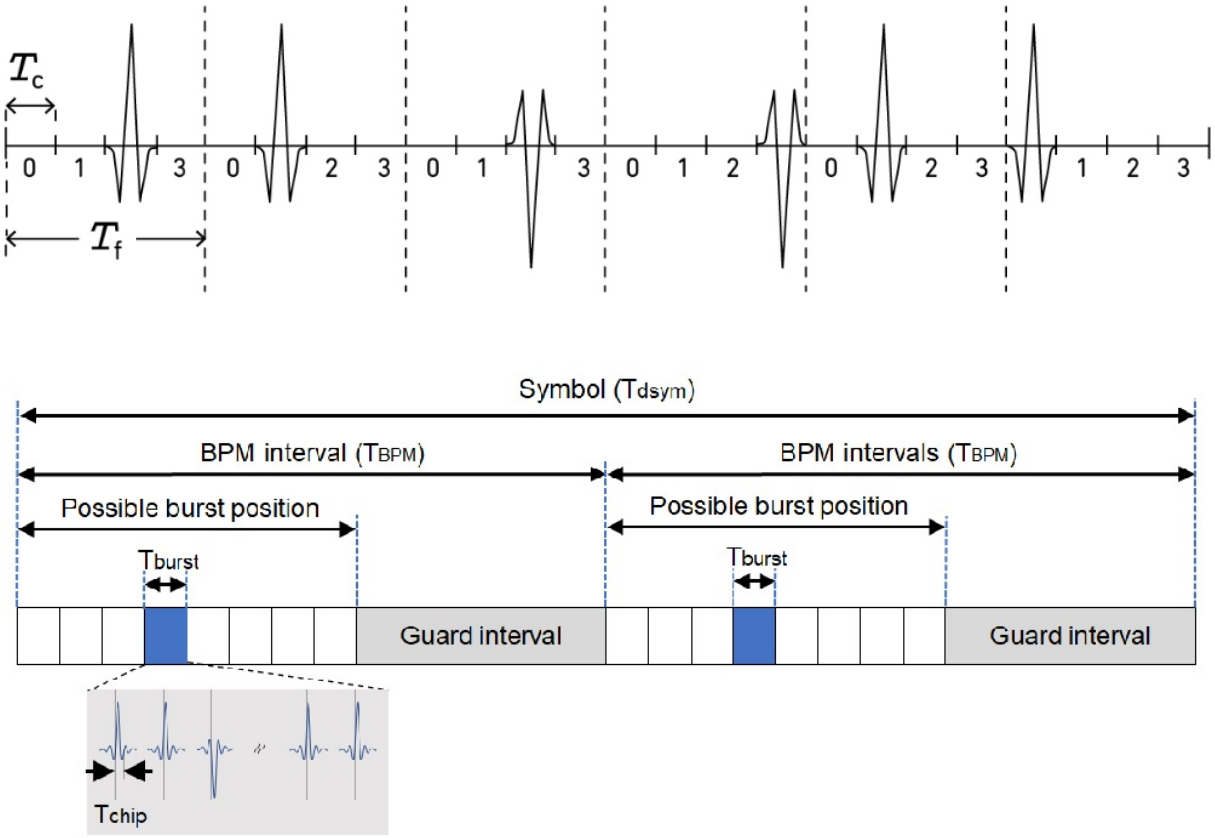


Figure 2.4: HRP UWB PHY Symbol Structure [1]

PHY Frame

Figure 2.5 shows a schematic view of a PHY frame as defined by the IEEE 802.15.4 standard. The Synchronziation header (SHR) containes the information needed to detect the signal and ajust to its parameters. The PHY header contains meta information about the payload and its encoding. The PHY payload contains the data that is to be send, namly the MAC frame.

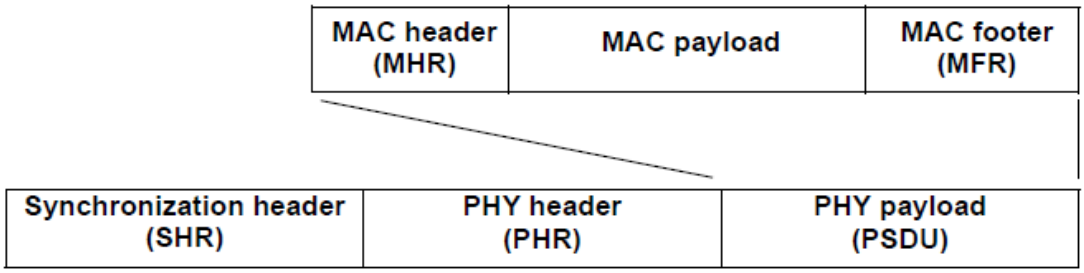


Figure 2.5: Schematic View of a PHY frame defined by IEEE 802.15.4 [2]

Figure 2.6 shows the synchronization header, consiting of two parts. The SYNC section is detectable by the receiver and informs it that a transmission has started. Depending

on the predefined mode, pulses of different length. The sequence of pulses specify a set of chanels that can be used for communication. The preamble can also be used to identify a PAN coordinator.

The SHR ends with the Start of Frame Delimiter (SFD). It indicates that the synchronization has ended and the comming signals will be data, starting with the PHY header. It also contains a timestamp which can be used for ranging using time difference of arrival (ToA), see section **??**

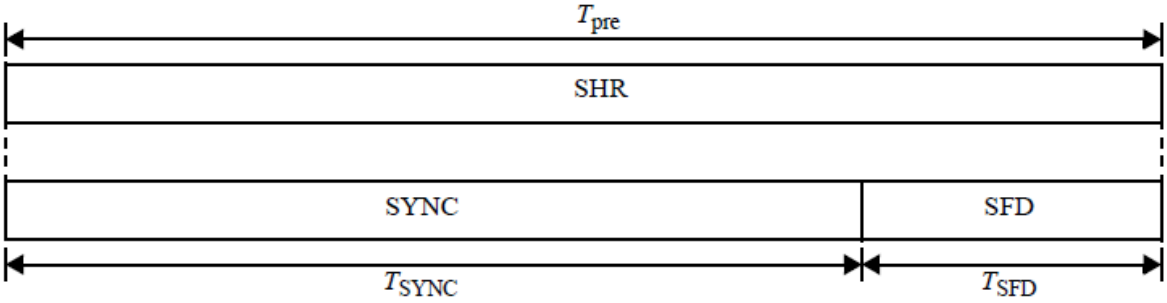


Figure 2.6: SHR Field Structure [2]

The PhY header contains all information needed to read the PHY payload (see Figure 2.7). The first bit defines the data rate that will be used during the playload transfer (see section 2.1.3). The next seven bits define the length of the frame, with a frame length of maximal 128 bytes. the 10th bit shows if ranging will be used with this frame. The next bit is reserved. Bits 11 and 12 define the preamble duration. It specifies how many repetitions are used, which can range from 16 to 4096. The last 6 bits are single error correct, double error detect (SECDED) bits that form a Hammock block and can be used to correct single bit errors and detecting, but not fixing, double bit errors.

The last part of the PHY frame is the The PHY payload (PSDU). This contains the the MAC frame, as defined in section 2.1.2.

Bits: 0–1	**2–8**	**9**	**10**	**11–12**	**13–18**
Data Rate	Frame Length	Ranging	Reserved	Preamble Duration	SECDED

Figure 2.7: General PHR Field Format [2]

The 802.15.4z amendment contains optional changes to the PHY frame format if the participaiting devices are HRP-ERDEV devices. Figure 2.8 shows the newly allowed structures for a UWB frame. Configuration 1 is equivalent to the already existing PHY frame. The others additionaly contain a scrambled time stamp. This can be placed in different places after the SHR. Since UWB can also be used only for ranging without transmitting a message, configuration 3 only contains the SHR and STS, without a payload.

Aditionaly the PHR can be formatted differently (see gigure 2.9. The reserved field and preamble duration is removed to make more space for the frame length. This allows to send more data in one frame, increasing the throughput of the UWB communication.

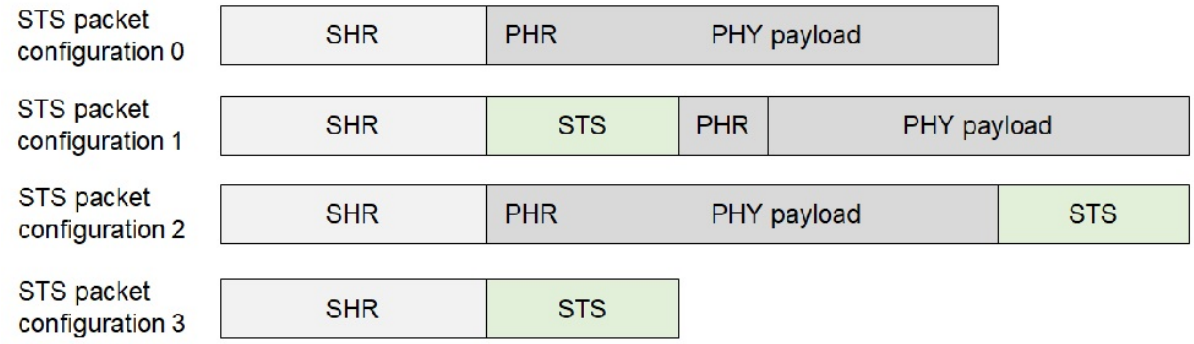


Figure 2.8: HRP-ERDEV Frame Structures [1]

Bits: 0	1	2–11	12	13–18
A1	A0	PHY payload length	Ranging	SECDED

Figure 2.9: PHR Field Format for HRP-ERDEV in HPRF Mode [3]

2.2 Related Work

2.2.1 Artwork Tracking

Since art preservation is an old field and temperature, humidity, light and vibrations have been known to be detrimental to most artworks, especialy paintings, most research in this direction is older than 20 years [5, 6, 7]. Still, the envention of new technologies, such as pattern recognizion using artificial inteligence, improvement on existing tools like infrared imaging and a active need for solutions have kept the resaech into artwork preservation an active field [8, 9]. One such new technologies are sensor networks, which have become widespread in the field of art-preservation [10].

Artwork tracking during transportation has not been a major focus in academia. The most relevant related research was done by Fort et al. [11]. They developed a low-cost, low powered sensor node to track temperature, humdity, pressure and vibrations of artwork and wooden structures. The sensor node would then report its findings to a remote server. They confirmed the validity of their sensor in a series of experiments, that were performed in a static building. They also presented a theoretical framework for their sensor to be used in a transportation scenario, but they do not report having implemented or tesed this system. Their sensor used an ascelerometer to detect vibrations, and the Bosch BME280 sensor to detect pressure, temperature and humidity. Their sensors did not build a network and were not queried, but reported their findings directly to either a wlan router or a ble-capable smart-device. The research of Fort et al. showed the value of low-cost sensors in the detection of threats to artwork.

2.2.2 Sensor Networks

Wireless Sensor Networks (WSN) have become a central aspect if IoT. Researchers have tried to focus on the most prevelant problems arrising from the development of NSWs, mainly power managment, security and privacy, data integrity and avaliability [12].

[13] researched WSNs outside of the controlled environment of a house. They propose a WSN that can track thevitals of mountainiers and call for help when measurements have dangerous values. They used an Ardruino Mega board equiped with a radio transiever, using LoRa with a star-topology, was used.

[**?**] created a WSN of NRF24101 board that is intended to monitor linear infrastructures like deepsee wires, using radio and wifi for communication. Using deep sleep they were able to optimize energy usage so the sensor is predicted to last five years on batery.

[**?**] used an acelerometer to detect vibrations in pipeline to discover leaks. They used a narrowband connection for communication and GPS for localization. Their sensors could query each other for data, to provide a more complete image of the situation.

2.2.3 Wireless ranging

[**?**] made an overview of publications envolving positioning systems for industrial settings. They looked at the positioning systems in papers using RFID, BLE, UWB, Wi-Fi and ZigBee. They found that UWB consistently reported the highest accuracy of these methods. UWB was the least affected by multipath-effects, although it was still the most common issue with this technology.

Early research of ranging using UWB was done by Gezici et al. [**?**, **?**]. These papers gave an overview of the different positioning systems for UWB, angle of arival, recived signal strength, time of arrival and time difference of arival. Time of arival and time difference of arrival were studied further in these publications, presenting errorsources and mitigation tools.

Early research focused on augmentation of UWB ranging methods. [**?**] proposed using integer programs for midigating the error for ranging without line-of-sight. [**?**] tried to solve the same issue by using methods based on the statistics of multipath-effects. BiasSub and BiasRed was proposed to reduce the bias in time difference of arival, by appling of a well-known algebraic explicit solution for source localization [**?**]. [**?**] emproved uwb ranging by eliminating random error. They did this by pre-filtering, using a anti-magnetic ring to eliminate outliers and using the double-state adaptive Kalman filter to improve position accuracy. Newer research has also begone incorporating neural networks into UWB positioning systems [**?**, **?**, **?**].

UWB localization has been used in many applied context. It has been proposed for pedestrian tracking [**?**], drone flying [**?**],robot navigation [**?**], navigation system for visualy impared people [**?**] and tracking people in buildings [**?**]. UWB positioning systems are particularly interesting for industrial IoT settings. [**?**] measured the performance of three

different UWB antennas, Qorvo, Sewio, and Ubisense. They measured a lot of multiplath-effects in such a complex environment. The midigated this by employing a Bayesian filtering method. [**?**] used UWB positioning in combination with Real-time kinematic positioning, to track workers while monitoring the factory. The goal was to trigger an alarm if a dangerous situation occured.

Chapter 3

Design

This section presents the principle design of the monitoring system. In the section 3.0.1 the components used are presented. Section **??** desctibes the functionalities and responsibilities of the system components. In Section **??** the network topology and data-flow is discueed

3.0.1 Hardware

This sectiondescribes the hardware used in the project. The setup consists of two distinct components: the artwork-tags, of which there are four, and one Phone that provides the interface to the user. The tag itself consists of 4 components:

1. nRF52840 Microcontroler
2. DWM3000 UWB Shield
3. DHT22 temperature and humidity sensor
4. MPU6050 accelerometer and gyroscope

Microcontroler

The fundament of the artwork-tag is build by the nRF52840 DK microcontroller developed by Nordic Semiconductors. It is part of the nRF52 series of microcontrollers intended for development. The nRF52840 DF is specialiezed for ble communication, for which it already includes the neccesary components. It is compatible with the nRF52 Software Development Kit (SDK), also developed by Nordic Semiconductors. The SDK makes it possible to use the ble functionalaties and to control the pins. It aslo includes implementations for a plethera of pin based protocols. It contains 58 pins, 48 of which are data-pins and manage the power suply for additional modules, which includes 3.5 and 5 Volt supply pins. 32 of the pins are installed in the same way as the pins on the Ardruino uno,

making it compatible with many peripherals that were designed with this common board in mind, such as the dwm3000. The remaining ten pins are enough to attach the sensors to. The nRF52840 DK includes a USB-B port that is used for Powersuply. Additionally it is connected to two pins and is used for UART-communication and for debugging. The nRF52 was chosen since it was avaliable and previous projects have been done with it in compination with the DWM3000 shield. As a result a lot of initial setup was already avaliable.

UWB shield

For communication between the tags as well as distance measurement the DWM3000 UWB-shield developed by Qorvo was chosen. The DWM3000 is a commonly used device for research involving UWB [14, 15, 16]. It allows low level access, but includes an SDK written in C that makes a lot of the processes transparent to the user, if they wish. The SDK uses the Serial Peripheral Interface (SPI) for communication between the shield and the microcontroler.

Humidity and temperature sensor

For humidity and temperature sensors I decided to use the DHT22(AM2302) produced by Guangzhou Aosong Electronic Co. [17]. It is a comonly used sensor in IoT monitoring systems [18]. The vendor claims a temperature range from -40Â° to 80Â° Celsius with a precision of 0.5Â°. [18] could experimentaly confirm that errors did not exceed 0.1Â°Celsius. They also concluded that the sensor is slow in detecting temperature change. This is also confrimed by the user manual [17], that states a read-interval of less than 2s is not possible.

The humidity sensor can detect the full range from 0% to 99.9% humidity, with an advertised maximum error of 2 percent-points [17]. I could not find any research that confirmed or denied these claims.

The DHT22 sensor uses three pins from the microcontroler, two pins for power suply and ground and one for single-bus communication. Since no SDK for this type of communication has been build for the nRF52 board series, it had to be implemented manually by reading the high and low voltage on the communication pin and, detecting headers and footers and parsing the binary messages. Dmitry Sysoletin published a project on github (`https://github.com/DSysoletin/nRF52_DHT11_example`) that handles the communication between an nRF52840 and a DHT11 sensor. Since the communications is mostly the same, I used his implementation, but changed the parsing of the actual data to fit the encoding used by the DHT22.

Accelerometer and Gyroscope

The MPU6050 sensor produced by InvenSense Incorporated provides accelerometer and gyroscope data. The accelerometer reports the acceleration in the three cardinal directions

in meters per second. The Gyroscope reports the rotation around the threee euclidean axis in degrees per second. In this project the accelermoterdata was not used, just the gyroscope.

The MPU6050 uses 4 pins, two for power suply and ground and two communication. The Sensor communicates using the I2C protocoll, a serial synchronus communication system. The microcontroller acts as the master and would in theory suport multiple workers on the same bus. Here only the MPU6050 uses I2C and is therefore the only worker. While the nRF52 SDK does not supply a I2C API, it offers a Two Wire Interface (TWI) implementation that is compatible with the I2C protocoll. It even used to offer MPU6050 specific support in the older SDK. I was able to port this older code to the current SDK.

Tag technical plan

The microcontroler builds the base of the Artwork-Tag. The other devices are attached to it over the avaliable pins. In the nRF52 SDK each data pin is assigned an integer value. These often correspon with the name of the pin according to the nRF52830 DK manual, but not always. I will use the names given in the manual to describe the pins. Pins are the methods by which a microcontroller controls its peripheras.

Some pins are intended for power supply. On the NRF52840 these pins are located in section P1, see table 3.1. The three VDD pins suply electricity with a Voltage of 3.5 Volts. A secondary power supply that uses 5 Volts is also avaliable. What voltage is needed depends on the periipheral. In this case, the DHT22 runs on 3.5 Volts, while the MPU6050 is made for 5 Volts. The P1 section also contains two ground pins, that need to be connected to the peripherals and a reset pin, so restart the microcontroler. The last pin is not connected (N.C.). There are additional ground pins in sections P4 and P24 of the board.

The other pins are called data pins. By using voltage modulation these pins can transfer data and therefore be used for communication. The nRF52840 has a I/O voltage of 3.3 Volt. This means that a voltage of 3.3 Volt corresponds to a *Logic high* and 0 Volts represents a *Logic low*. This allows the data pin to transfer communication in a binary encoding. How a signal is interpreted is defined by the used communication protocol. The MPU6050 for example uses the I2C protocol and uses 2 datapins. This defines that one pin is used for a serial clock and the other pin transmits data. For the data transmission the protocol defines what a package looks like. This includes the start condition, the voltage characteristics that signal the beginning of a package, adressing, data encoding, acknowledgments and stop condition.

The DHT22 sensor does not use a given communication-protocol. It uses one data-pin to report its sensor data. How that data is encoded to high and low voltage is specified in the user manual [17] and has to implemented manualy.

The DWM3000 shield is mounted on the 32 pins ment for arduino one connections. All pins are forwarded and can be used by other devices, in a common ardruino-stackable

style. If they are data-pins they will share the data. Table 3.1 shows which devices use which pins. The only pin shared by multiple devices are power and ground pins. The microcontroler suplies enough power to support this.

The sensors are attached to the same powersource and ground as the shield, but use different data-pins. The DWM3000 leaves enough pins unused that both sensors could be attached to them. Since it is not visible which pins the shield leaves free, I decided to use data-pins that are not attached to the DWM3000 in any way. Table 3.2 shows how the sensors are connected to the remaining open pins.

3.1 Architecture

In order to discuss how the dataflow works, first the section 3.1.1 will establish what services are implemented in each part of the system. The section 3.1.2 will explain what triggers events and how they are handled inside the system.

3.1.1 Responsibilities

The system consists of the tags, the sensor network and the phone. These parts all have their own responsibilities.

Tag: The tag is responsible to manage its sensors. It has to do correct setup and converts its output into a understanable form. The tag can performe ranging with all its neighbours. Additionaly the tags are responsible to search for networks to join, and react aprobiatly to network request, be those queries for sensor data, ranging requests or network managment jobs. The tags provide a unique, secure universal identifier, to be used by queries or the network. How this is done is part of the certify project and will not be discussed in this thesis. The tag is also responsible for its own power managment. This is not the focus of this thesis and will only be mentioned when relevant. A guideline on powermanagment will not be provided here.

Network: The network is responsible to keep track of all tags taking part in the network. It offers a joining protocol for new devices and remains stable when devices leave or become unavaliable. It offers the possibility for phones to connect to the network. It ensures quieries from phones get transported to the correct tag and the answers to the correct phone. It ensures a network topology that corresponds to a graph that is at least 3-connected. On request it returns a list of connected devices to the phone.

Phone: The phone connects to the network via the provided method. It offers a graphical user interface (GUI) to be used by the driver. The GUI offers a method for the driver to set the acceptable ranges for all sensor data. Additionaly it offers a method to set query intervals-length. The phone is responsible to query sensor data for each tag and masurement once in each interval. The phone has to evaluate the answer. The phone has to report the results to the driver using the GUI. If a parameter falls outside of the acceptable range for its type, the phone is responsible to alert the driver to this fact.

Table 3.1: Adruino compatible pin assignment

	Pin	DWM3000	DHT22	MPU6050
P4	P1.10	✓		
	P1.11	✓		
	P1.12	✓		
	P1.13	✓		
	P1.14	✓		
	P1.15	✓		
	GND	✓		
	P0.02			
	P0.26	✓		
	P0.27			
P3	P1.01	✓		
	P1.02	✓		
	P1.03	✓		
	P1.04	✓		
	P1.05	✓		
	P1.06	✓		
	P1.07	✓		
	P1.08	✓		
	P1.10	✓		
P1	VDD			
	VDD			
	RESET			
	VDD	✓	✓	
	5V	✓		✓
	GND	✓	✓	✓
	GND	✓		
	N.C.			
P2	P0.03	✓		
	P0.04	✓		
	P0.28			
	P0.29			
	P0.30			
	P0.31			

Table 3.2: Non-Adruino compatible pin assignment

	Pin	DWM3000	DHT22	MPU6050
P6	P0.00			
	P0.01			
	P0.05			
	P0.06			
	P0.07			
	P0.08			
	P0.09			
	P0.10			
P24	P0.11			✓
	P0.12			✓
	P0.13		✓	
	P0.14			
	P0.15			
	P0.16			
	P0.17			
	P0.18			
	P0.19			
	P0.20			
	P0.21			
	P0.22			
	P0.23			
	P0.24			
	P0.25			
	P1.00			
	P1.09			
	GND			

The certify project also plannes to collect the sensor data on remote servers using a 4G connection. The plan is to equip each tag with antennas to allow it to send the data directly to the server itself. Since this is not a part of this thesis, the responsibility for the tag to do this was not added to a list. A known problem with this plan is, that a 4G connection is not always possible. Since small tags have very limited memory, the plan to store the sensor data on the tag is not pheasible. If the setup presented in this thesis is used, it would allow for the storage of the data on the phone, which has a much larger memory. This again was not added, since it is not part of this thesis.

3.1.2 Dataflow

How a tag connects to the network is described in the section 3.2. Figure 3.1 shows a sequece diagram ob the setup and main observation loop of the system. On the top the communicating parts are listed.

- Human is the driver of the truck
- Phone is the phone used by the Human
- Network consists of all the tags that are used and the network they build.
- Connected Tag is part of this network, but is listed seperatly. It represents the tag that is communicating to the phone
- Tag j is also part of the Netowrk. It represents the tag that is queried during the observation loop

Phone and Human communicate by using a GUI. Phone and Connected Tag commuincate using BLE. Every communication inside the network happens using UWB. This includes the communication between the Connected Tag, the network and Tag j.
When the phone wants to connect to the network, it looks for advertised BLE devices. It then displays the devices to the user and lets him pick one. The phone then pairs to the chosen tag, making it the connected tag and the phones connection to the network of tags. Once connected to the network, the phone will promt the human to enter the parameters. These consist of:

- Upper and lower limit for sensor data, like temperature and humidity
- Maximal displacement value for distance and gyro. These values represent the maximal difference in registered values that is allowed for positional measurements.
- Time between measurements. This gives the time period that will pass between measurements for each device and measurement type.

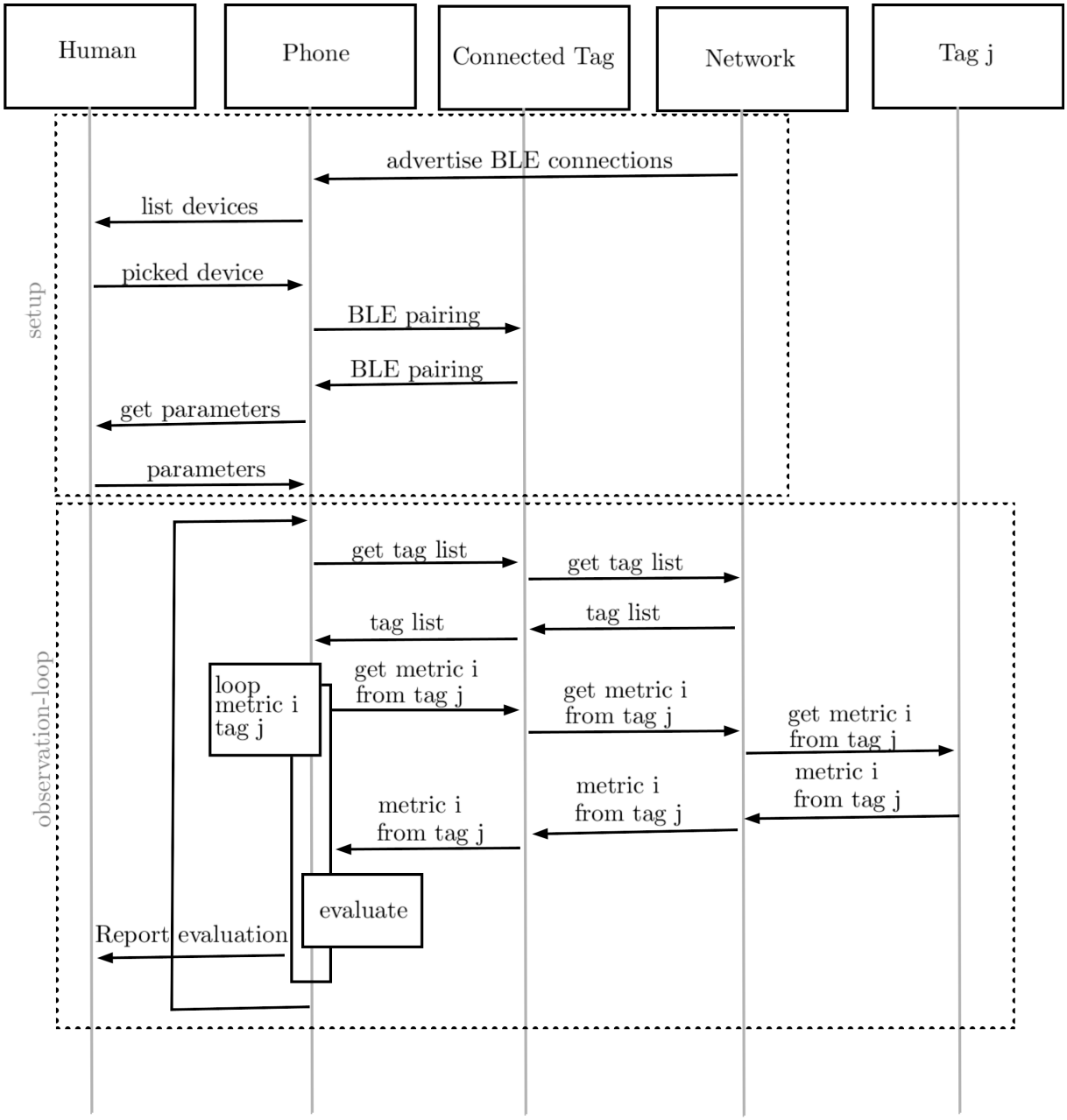


Figure 3.1: Sequence diagram of setup and oberservation loop. Setup is berformed one, oberservation loop repeats until stopped.

Once the parameters chosen, the user can start the observation.
Each iteration of the observation loop begins with a call to the network for a list of all tags currently in the network. Since the tag network is a dynamic sensor network, the tags in the network can theoretically change. In practice this should onyl happen, when artwork is unloaded or if a tag becomes faulty. The request for the list is transmitted to the connected tag ober BLE, which than queries the network for all conected devices. The responde is returned to the Phone. The phone then starts a nested loop, iterating over the list of tags and the list of metrics caputred by the system. For each measurement and tag combination (i,j) the phone contacts the connected tag for the value, which in turn queries the network. Once the message has arrived at the tag j, tag j gets measurement

i. In case of sensors this intails contacting the sensor and requesting a value. If metric i is a distance measurement, tag j will commence a two-way ranging operation over UWB with all its registered neighbours and will report the list of distances, together with the tag-adresses they correspond to. Metric i is then transported over the network back to the connected tag and finaly to the phone. The phone must then evaluate the retrieved data.
During the evaluation process, the phone creates an evaluated measurement, and marks it as problematic or unproblematic. What the evaluation looks like depends on the metric.

- For most metrics, like humidity and temperature, the evaluated measurement is equivalent to the received measurement. It is then checked, if the measurement falls into the aceptable measurement parameters, set by the human. If it does not, the evaluated value is marked as problematic.
- Some metrics require comparisment to the previous data. The gyroscope reports the current orientation of the tag. This is then compared to all previous measurements and the maximal angular difference forms the evaluated measurement. If the evaluated metric is bigger then allowed by the set parameters, the measurement is marked as problematic After evaluation the original measurement is added to the list of previous measurements.
- The distance measurement has a unique evaluation process, which is describbed in section 3.1.2.

Once the data evaluation is done, the evaluated measurement is presented to the user over the GUI, together with the adress of the tag it belongs to. If the evaluated measurement is problematic, the driver it alarmed.

Distance evaluation

The goal of the distance evaluation is to build a working model of where every tag is. To achieve this, a quadratic program is solved, to get the coordinates of all tags. The steps to do this are as follows:

1. Get a list of all current tags, $T := \{t_1, t_2, ...\}$.
2. For each tag, get the last known distance measurements and put it into a set $S_D := \{(t_i, t_j, d_{ij})\}$, where t_i is the tag which measured, t_2 the tag that was measured to and d_{ij} the distance measured.
3. If a tag has no distance measurements, remove it from the list.
4. Assign each tag t_i a position in a 3D coordinate system, (x_i, y_i, z_i)
5. Pick one random tag t_o.
6. Set the values x_o, y_o, z_o all to 0.

7. Create the objective function: $L(X,Y,Z) = \sum_{t_i,t_j,d_{ij} \in S_D} |(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2 - d_{ij}^2|$. For x,y and z use variables for all but the six values set in step (6).

8. Solve the quadratic program consting of the Objective function L and no constrains.

Quadratic Programs in general are NP-Hard, but Quadratic Programs with a convex function can be solved efficiently. $(a-b)^2$ and c are convex functions. The sum uf a convex function is always a convex function. The objective function in (7) only sums up convex functions and is therefore convex itself. The quadratic program can therefore be solved efficiently.
By setting the values of tag t_o to zero, the results of the quadratc function becomes grounded. It is not strictily nessecary, but without it the returned solution could have values anywhere in the Euclidean space. by setting one tag to the coordinates at theo origin, the solution will place the other tags near that region. There are still an infinte amount of solutions to this quadratic function, since all solutions can be roatated around any axis and still return the same objective function.

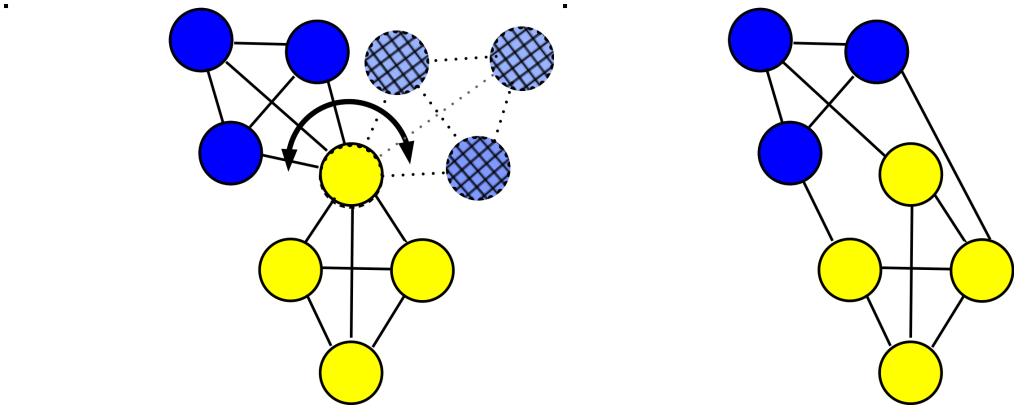


Figure 3.2: Left: Five dots, all having at least two connections, still blue can move independantly. Right: minimal 2-connected graph, no movement possible.

For a point to be clearly placed in Euclidean space, three distances to other points have to known. This alone is not sufficient to insure a unique results. The left of Figure 3.2 illustrates this point in two-dimensional space. Every circle is connected to two others, still the blue circles can move without the whole figure moving. What is needed to keep every point static is for known distances and tags to build a four-connected graph (three in two dimensions). A four-connected graph is a graph, were it takes at least four removal of vertices, to create two isolated subgraphs. The left of Figure 3.2 shows a sollution of the problem on the right by creating a three-connected subgraph.

Once the coordinates for all tags are found, they are compared to previous results. For each tag the phone calculate by how much it has moved. The evaluated measurment is the distance of the tag that has moved the most. If the evaluated measurement is larger then the maximal allowed displacement, the measurement is problematic.

3.2 Network

LR-WPAN Topologies

Star-topologies or PtP where one device is a FFD that acts as a PAN coordinator. ”The main goals of LR-WPANs are easy installation of networks, reliable short-range data transfer and meeting the need for relaxed throughput requirements.”

3.2.1 Ranging

Chapter 4

Implementation

In this section, the implementation that was used for the experiment is discussed. In section 4.1 the implementation of the tags is presented. Section 4.3 is about the implementation of the App.

4.1 Tag

The software of the tags consits of n modules:

1. Temperature and humidity sensor
2. Gyroscope
3. Two way ranging
4. UWB network
5. BLE communication
6. Job handler

The following subsections will discuss the first five modules, followed by how they interact using the job handler module. The section 4.2 discusses chalanges from combining these modules and how they were solved.

4.1.1 Temperature and humidity module

This module is responsible for managing the DHT22 humidity and temperature sensor. It is responsible to setup the sensor during initial startup and to provide the sensors measurements when queried. The DHT22 sensor communicates using only one data pin, pin 13, which will be referd to as the data pin in this section. Dmitry Sysoletin created an

implementation **??** for the DHT11 sensor together with the nRF52840 board that build the basis for this implementation, by addapting it to the DHT22 and adding functionalities needed by the job handler module.

Since the DHT22 is a very simple sensor, using single bus comunication, not much setup is needed. The evaluation of the sensor data requires that the voltage of the pin is read out in pre-defined intervals, when reading the sensor data. To do this, a clock is required. This resource has to be reserved an initiated at startup. This is the only setup that is required for the DHT22 sensor.

To initiate a sensor-read the voltage of the data pin is set to 0. When the sensor is in standby mode, the data pin is on *logic high*, and when set to *logic low*, the sensor will respond with a read of its current value. A schematc view of a sensor read of the DHT22 can be seen in Figure 4.1. The temperature and humidity module will then check the Pin State in intervals of 5ms, until a *logic low* is registered, signalling that the sensor has registered the request. The module will now monitor the pin state, waiting for *logic low* followed by a *logic high*, this beeing the start condition of the data transfer.
The data is transfered in five chunks of eight bits. Each bit is preceeded by a prolonged *logic low* state, that is detected by the module The module then proceeds to write the state of the data pin into a 8-bit buffer, *logic high* corresponding to a 1 and *logic low* to 0.
Once all five chunks are read, the communication has ended and the module can verify the data. The first two bytes correspond are combined to form the temperature information in celcius, the second and third form the humidity. Both values are multiplied by 100 and stored in a 16-bit integer. This doesn't loose data, since the sensor only measures up to a precision of 1 after the decimal point. The data beeing stored in an integer help with data transfer. It will be converted back on the phone. The fith chunk contains the parity and is used to accept or reject the humidity and temperature values. If the process fails at any state, -100Â°C is returned for the temperature and -100These form both impossible values, since humidity can't be negative and the DHT22 sensor can only detect temperatures as low as -20Â°C.

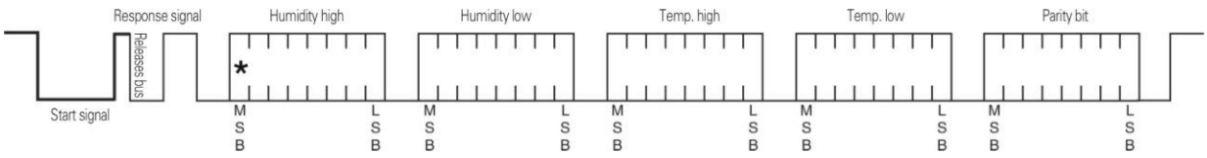


Figure 4.1: Signal of a DHT22 sensor-read as presented in the manual [17].

4.1.2 Gyroscope

4.2 Combining modules

4.3 App

Nordic Semi Conductors, the maker of the used microcontrolers, published the code to a simple app that allows for BLE communication with their devices. It is called nRF Toolbox. It is intended to pair with the ble_ app_ uart example, published in the nRF52 SDK. Since this example code was used as the basis for the ble communication used in this project, it was addapted to work with this project.

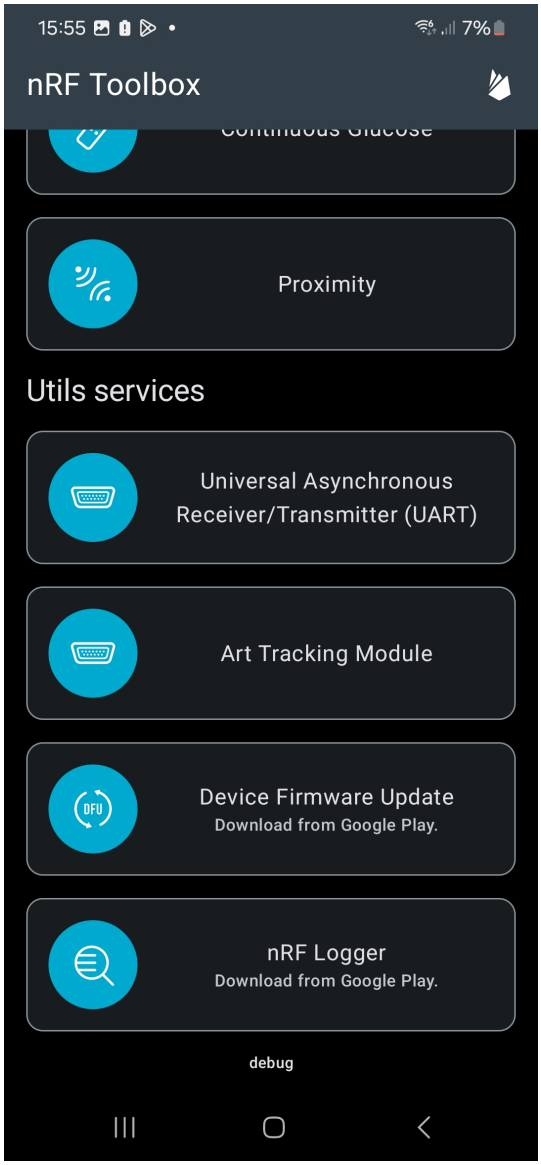
The App contains different modules, intended for different examples, among them the Universal Asynchronus Receiver/Transmitter (UART) module (see 4.2). It is intended to be used with the ble_ app_ uart example. When opem it shows the ble services that are currently beeing addvertised and allows the user to connect to one of them 4.4. It then opens a window similar to phone messangers, were the keyboard can be used to tpye messages, that are sent to the connected devices.

Since the development of an application was not the primary focus of this thesis, it was decided to take the nRF Toolbox app and add a new module for art-traking to it. The UART module searved as the basis for this new module, since it had a lot of usefull services already implemented. As with the UART module the art-tracking module opens up the same connection page 4.4, that allows the user to select the art-tracking and connect to it.

Once connected, the observation screen is shown (figure 4.5). At the bottom seven parameters can be set: *time*, *max Temp*, *min Temp*, *max Hum*, *min Hum*, *max Angle*, *max Dist*. The parameters *max/min Temp/Hum* represent the expected range of humidity and temperature. Any measurement outside these parameter will be considered a dangerous value by the app. The tollerated difference in angle compared to the previous measurement is set by *max Angle*, larger differences are considered dangerous values. Distance measurement work analogously with *max Dist* in meters. The *time* set defines the time that passes enbetween measurements in seconds. The default is set to 350 seconds. This means that the time that passes between, for example, the temperature measurements on tag 2 are 350 seconds.

When the user presses the *Start Service* button, a services starts that poeriodically queries the tags for the Measurements. Figure 4.6 shows the measurement loop. Each sensor is assigned a character. T for temperature and humidity, G for gyro and D for distance. Each tag has a number, here from one to four since four tags were used in the experiments. The loop concatenates these two characters and sends the resulting query to the connected tag. Then the next tag-number is prepared for the next query. Once all tags have been queried for a sensor, the tag-number starts with the first again and the next sensor is queried. In between cals the app waits. The call time for distance-measurement is fixed at 80 seconds. Distance measurement takes longer than the other sensors, since for every

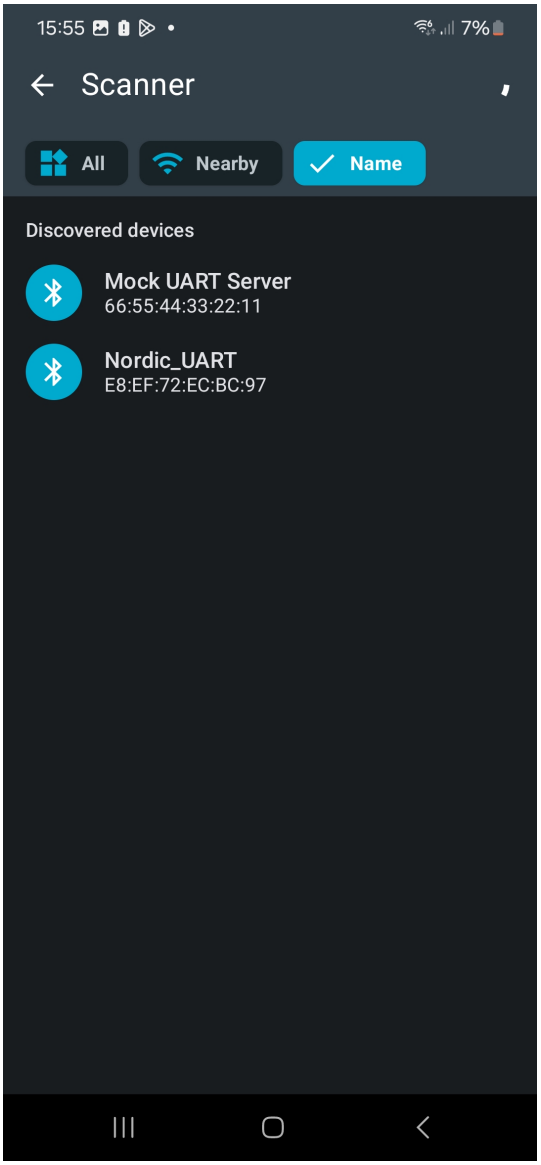
Figure 4.2: nRF Toolbox module menue, with the added Art Tracking Module



devices three measurements need tobe conducted. Additionaly the sensors that do not participate in a ranging session are sleeping for a quite generous amount of time, to ensure they don't distrub the ranging session. 80 seconds has been chosen, since it allows enough time for all the ranging to happen, plus two repeats per sensor in case the ranging session fails. For the other sensors the waiting time in between queries is calculated from the remaining set time, after the ranging time is deducted.

Once the process has started, the queries will appear in the chat window on the right side of the screen. The responses are on the right side, see figure 4.6. If the response is inside the set parameters, the message bubble will appear blue. If the measured value is considered a dangerous value, the text bubble will appear red (see figure 4.7). Since the message display is programmed in a asynchronus way, it can happen, that the answer to a query appears before the querry itself, if the querried tag is the same as the connected tag. The service can be stopped by pressing the *start service* button again or by exiting

Figure 4.3: nRF Toolbox shows avaliable devices to connect to



this screen in any way.

The query-answers are appended to a file that is safed in the app-storage. The information appendded consits of: the queried tag, the returned values, a timestamp and if the value was unproblematic. This functionality is intended for experimental evaluation. In a real word application, this data should be periodically backed up on a server in a compressed manner. When pressing the share-button on the top right of the message-box 4.7. It will open the Android naitive share functionality, to share the file over mail, an installed messanger, save it to onedrive or send it over Bluetooth. In this project all files were sent with email. Pressing the trashcan next to it will delete the chat and empty the file. This allows the user to distinguish between different testing session.

Figure 4.4: nRF Toolbox UART module screen

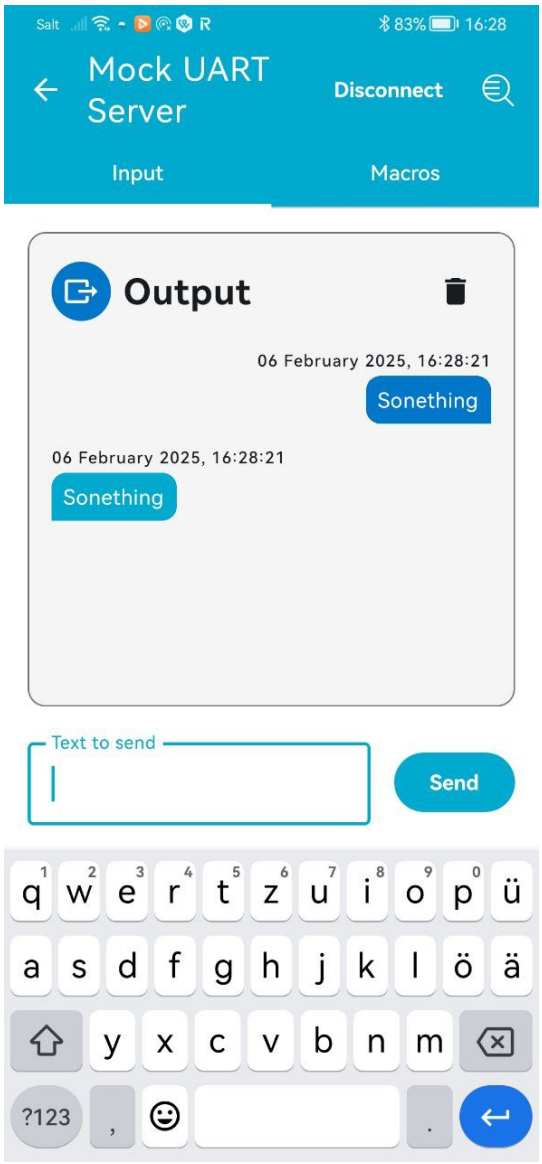
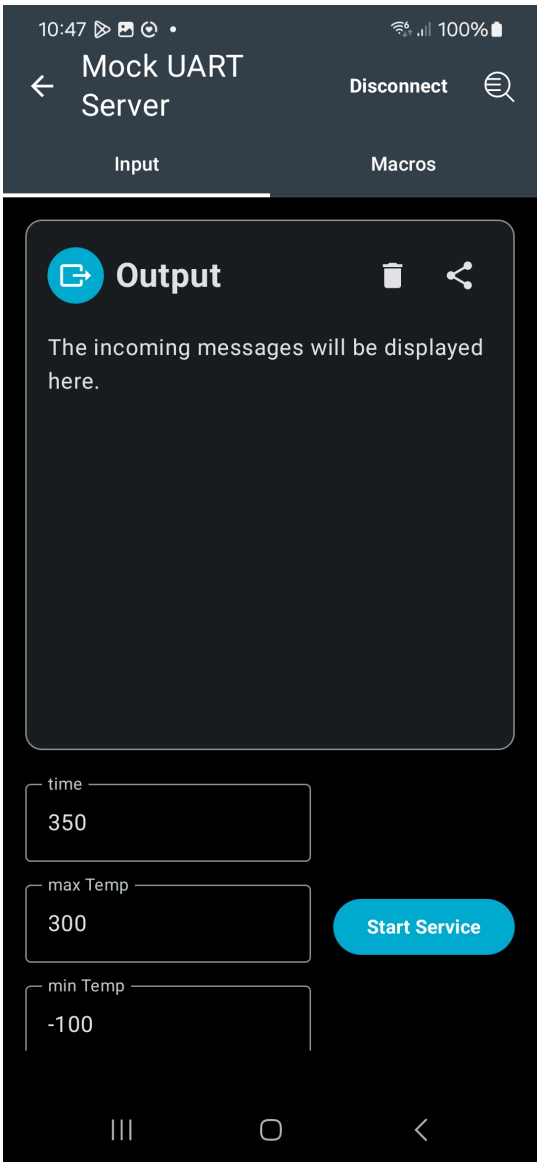


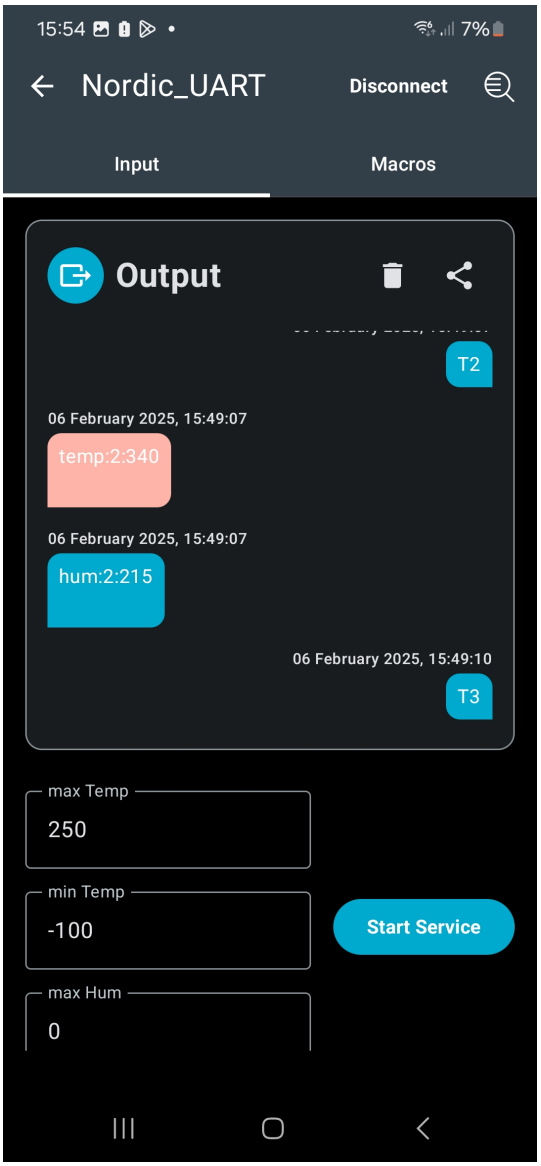
Figure 4.5: Art Tracking module oberservation screen before measurements



```
1     private val sensors = listOf("T", "G", "D")
2     private val devices = listOf("1", "2", "3", "4")
3     private var measurement_type = 0
4     private var tag = 0
5     private var timeBetweenCals: Long = 3750
6
7     private val runnable = object : Runnable {
8         override fun run() {
9             if (tag >= list2.size) {
10                tag = 0
11                measurement_type += 1
12            }
13            if (measurement_type >= list1.size) {
14                measurement_type = 0
15            }
16            val textToSend = "${list1[measurement_type]}${list2[tag]}"
17            artRepository.sendText(textToSend, MacroEol.LF)
18            tag += 1
19            if(list1[measurement_type] == "D"){
20                handler.postDelayed(this, 80000)
21            } else {
22                handler.postDelayed(this, timeBetweenCals)
23            }
24        }
25    }
```

Figure 4.6: Section from the ArtMetricService.kt, main measurement loop

Figure 4.7: Art Tracking module, queries and responses



Chapter 5

Evaluation

5.1 Experiments

Five experiments were performed to validate the functionality of the tags. The first two are non specific and ment to test the setup in a stable environment. Experiments three to five are intended to test the detection of unwanted circumstances. For all experiments the query-frequency was set to 330s, so measurment was evaluated once every 330s. The measurements queries are spread across this timeframe. Each experiment lasted between 40 minutes and one hour. The results were stored on the phone and then exported using email. The analysis of the data and creation of graphs was then performed using a Jupiter Notebook, using Pandas and Pyplot for datamanagment and the creation of graphs.

5.1.1 Experiment 1: Static

The four tags where placed on the corners of a 80 cm by 50 cm rectangle on a wooden table. Each tag was turned on sequentially and given enough time to establish the network. The phone then was connected to one tag. The parameters in the app were left unchanged. The default parameters are large enough, that no measurement should be large enough to trigger a warning. The setup was then left untouched for 35 min. The goal of this experiment was, to gauge by how much the measurements can vary in a static environment.

5.1.2 Experiment 2: Coordinated movement

5.1.3 Experiment 3: Temperature

The four tags were placed in the same 80 cm by 50 cm rectangle as in experiment one. One tag placed on a elevated surface, 4 cm above the table. Under the tag seven candles were placed (see figure TODO, Bild einfÃ¼gen). Next to the tag two thermometers detectors were placed. Each tag was turned on sequentially and given enough time to establish the

network. The phone then was connected to one tag. The max Temperature parameter in the app was changed to 35Â°C. After 20 minutes the candles were lit. The experiment was then left alone for another 30 minutes. The independent thermometers were filmed during the process, to allow for later review and comparesment. The goal of experiment 3 was to test the temperature detection capabilities of the system.

5.1.4 Experiment 4: Gyroscope

Again all for tags were placed on a 80 cm by 50 cm rectangle. Each tag was turned on sequentially and given enough time to establish the network. The phone then was connected to one tag. The maximal allowed angular difference was set to 30Â°. After 20 minutes one tag was turned by 90Â° counterclockwise. The experiment then ran for another 30 minutes. The goal of experiment number 4 was to test the detection of unwanted rotations. Experiment four was repeated with, with the gyro sending the angular velocity for all axes instead of the current position.

5.1.5 Experiment 5: Distance

The same 80 cm by 50 cm rectangle setup was used. The tags were turned on sequentilay, giving them enough time to build the network. The phone was connected to one tag. The max distance parameter was set to 20 centimeter. After 20 minutes, one tag was moved parallel to the shorter rectangle line about 20 cm towards the tag on the next corner. The system was then left resting for another 30 minutes. The goal of experiment number 5 was to test the detection of unwanted movement.

5.2 Experiment Results

In this section the results of the experiments are presented. All eperiments were performed two to three times. In each section only the data-set from the first experiment run is presented fully. The other experiments will be mentioned only, if they have differing data or to confrim an unexpected datapoint.

5.2.1 Experiment 1: Static

In eperiment one, all measurements are expected to be unchanging. Table 5.1 shows the mean values for temperature, humidity and angle during the experiment by tag. Figures 5.1, 5.2, 5.3, 5.4 shows the change of these values over time.

All four tags have a similar mean temperature and are all less than 0.2 Â°C apart from each other. The varaince are also small, tag two having the highest one with 0.05 Â°C variance. The graph shows that all tags have a rising temperature. The increase is

Tag	Temp Mean	Hum Mean
1	22.06	32.56
2	21.90	33.93
3	22.06	32.94
4	21.87	32.80

Table 5.1: Mean and Variances for Temperature, Humidity, and Gyroscope Data by Tag during experiment 1

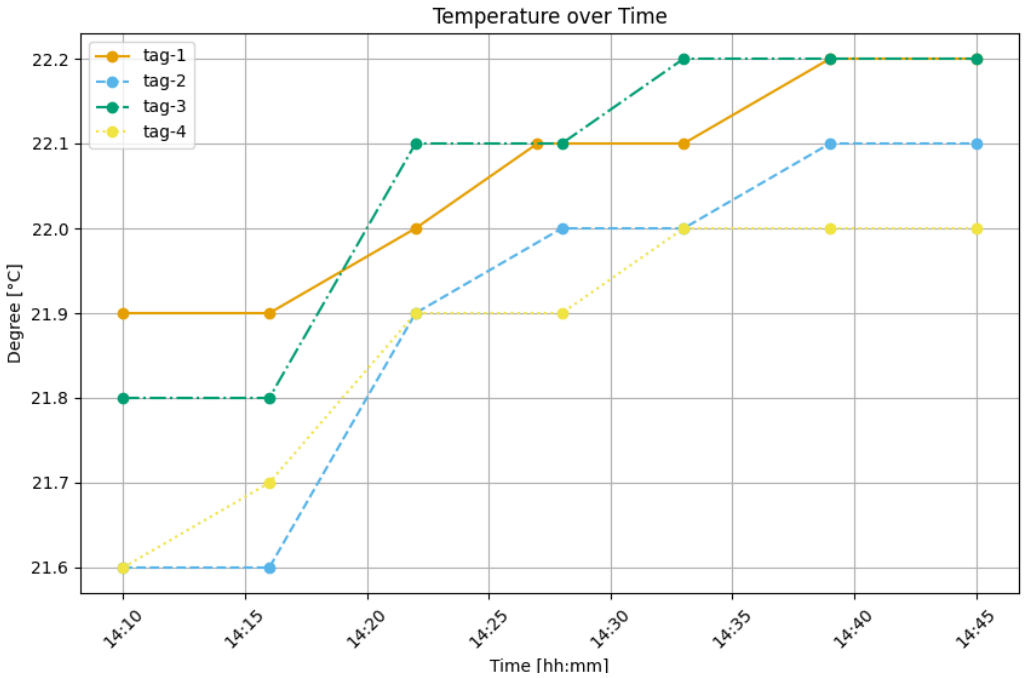


Figure 5.1: Experiment 1, temperature over time.

quite small with tag two having the biggest increase of 0.5 Â°C over 20 minutes. When the experiment was repeated, the means stayed similar and the variance small, but the temperature changed course. A downward trend was visible, instead of the upward trend seen during the first experiment.

Humidity follows a similar trajectory as. The means only vary by 1.5 % pt. The variance is small, with tag three having the biggest variance with 0.06% pt. During the first experiment, humidity increased by a small amount. When the experiment was repeated, the humidity dropped during the experiment.

Since all tags were stationary during the experiment, the gyro sensor was expected to be unchanging. This is not what happened. looking at the graph 5.3 it is clear, that the measurement shows a wide range of angles for each tag and axis. The only exception is tag 2 around the x axis, which stays at 0 for the whole measurement duration. Since angle measurements fall into modular arithmetic, it ”wrappes around” at 360Â°, means can only meanigfully be taken if the angles are in a small range. Since this is not the case for most tags, the only thing that can be said is, that tag 2 has a mean of 0 with variance

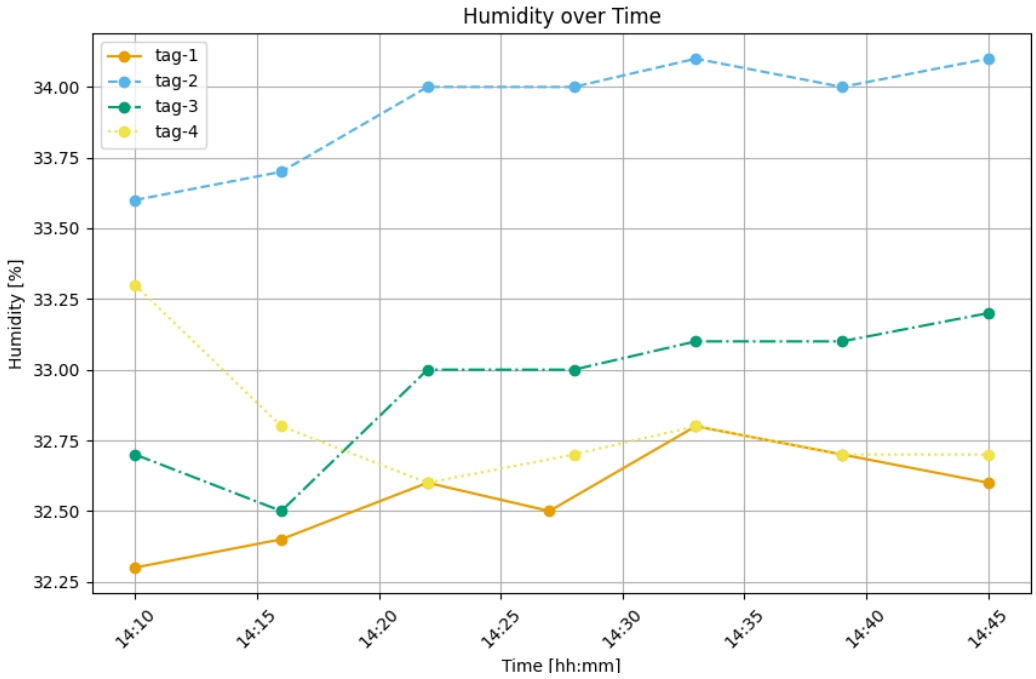


Figure 5.2: Experiment 1, humidty over time.

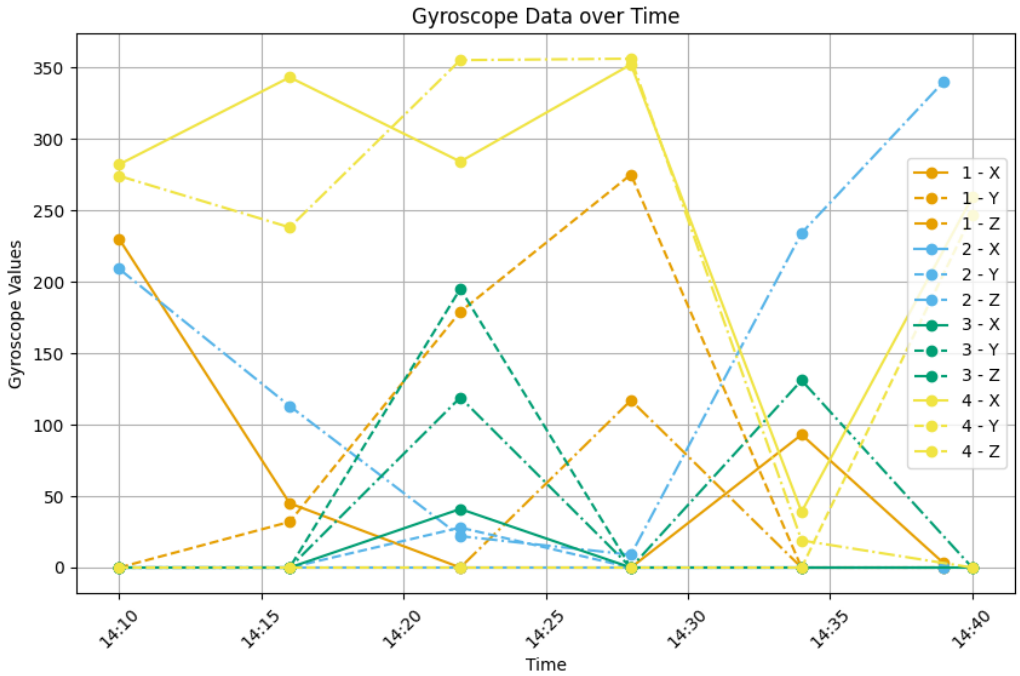


Figure 5.3: Experiment 1, angles over time.

0 around axis x.

Table 5.1, shows the mean of the measured distances, the row entry beeing the queried tag that initiates the distance measurement, and the row corresponding to the responding

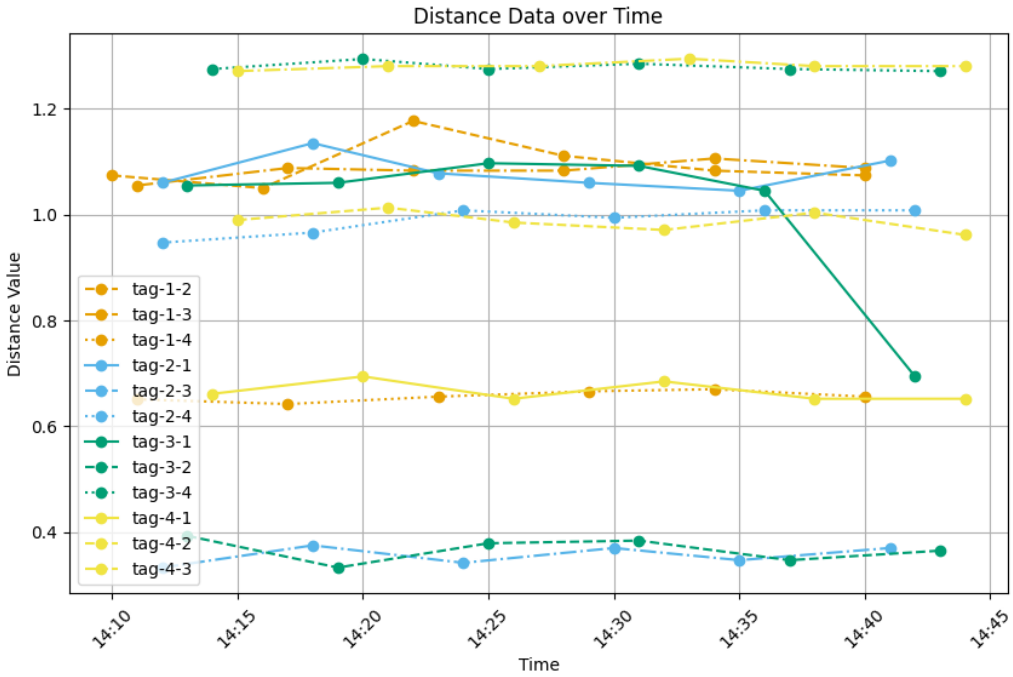


Figure 5.4: Experiment 1, distance over time.

tag. By looking to the measurements diagonaly oposed to each other, one can see that the measured distaances is the the same, indipendent of who initiated the measurement, up to a range of two centimeters. The varince on table 5.3 also show, that these measurements are stable over time and don't change by much. Looking at the graphs on figure 5.4, the pairs of measurements are visible. One outlier happens when tag 3 measures the distance to tag 1 at very end of the measurements. When repearing the measurements these outliers happened again, a bit less frequently then twice per hour. The outliers always affected a measurement involving tag 1. The distances measured do not correspond to the actual distances the tags had to each other, also seen in table 5.2. The measured distances can be as far of as 0.5 meters. The two larger distances, 0.8 and 0.94 meters, correspond to the two larger measured values for each tag, while the smallest measured value always corresponds to the smallest distance, 0.5 meters. The two larger values are not always ordered correctly, 0.94 meters sometimes beeing measured smaller then 0.8 meters. In repeated experiments, all these facts stayed true.

	1	**2**	**3**	**4**
1	0.0	1.094	1.084	0.657
2	1.080	0.0	0.356	0.989
3	1.007	0.367	0.0	1.279
4	0.666	0.987	1.281	0.0

	1	**2**	**3**	**4**
1	0.0	0.8	0.94	0.5
2	0.8	0.0	0.5	0.94
3	0.94	0.5	0.0	0.8
4	0.5	0.94	0.8	0.0

Table 5.2: Left: Mean distances between tags in experiment 1. Right: Expected values.

	1	2	3	4
1	0.0	0.002	0.000	0.000
2	0.001	0.0	0.000	0.001
3	0.024	0.001	0.0	0.000
4	0.000	0.000	0.000	0.0

Table 5.3: Variance of distances between tags in experiment 1. Row corresponds to queried tag.

5.2.2 Experiment 3: Temperature

Experiment 3 introduced heat-sources the system. Since the main setup was the same as experiment 1 5.2.1, many of the findings are the same. In this section, only differences in results are discussed. If a metric is not measioned, one can assume it behaved the same as for experiment 1 (see section 5.2.1).

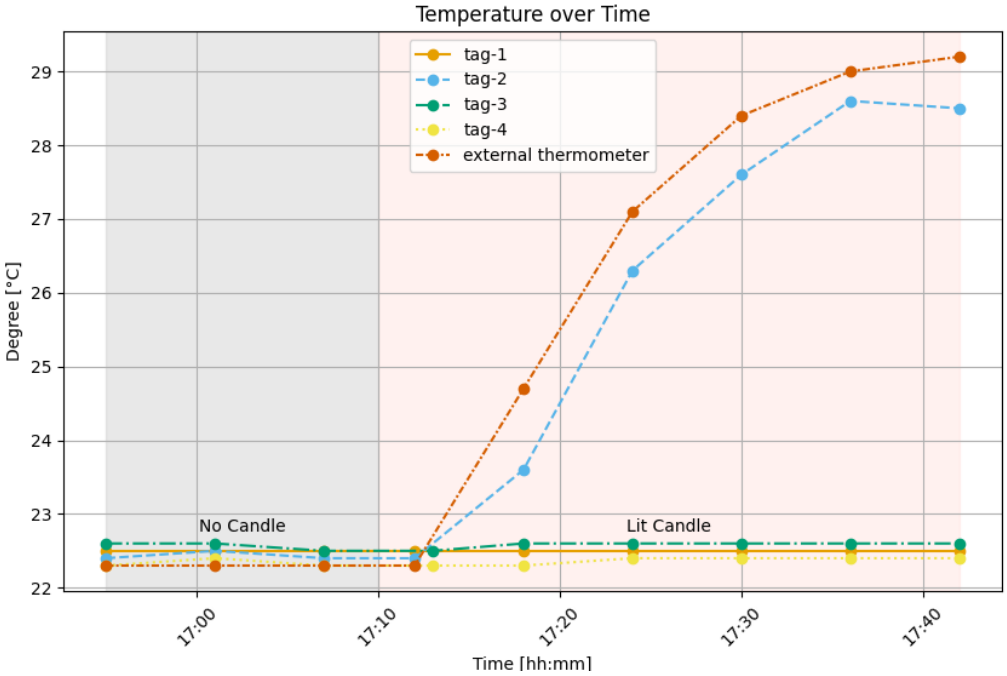


Figure 5.5: Experiment 3, temperature over time, mith external measurement added.

The progression of the external thermoeter and the internal temperature sensor can be seen in figure **??**. The candles, that functioned as the heat source, were lit at 15.10. During the next measurement of tag 2, at 15.12, both the external thermoeter and the temperature sensor on tag 2 had not yet registered any change, remaing at 22.4 Â°C. The extrenal thermometer started rising 1 minutes later, at 15.13. During the next measurement at 15.18, the temperature-sensor registered a slightly increased temperature of 23.6 Â°C, while the external thermometer registered 24.7 Â°C. During the next measurement at 17.24 the tag reported 26.3 Â°C while the thermometer showed 27.1 Â°C. The measured temperature of the external thermometer keeps klimbing faster than the internal

temperature sensor, until the end of the experiment, as seen in Figure 5.5. Their distance nether exeeds 1 Â°C and gets smaller towards the end of the experiment. The other tags do not report any significant change in temperature.

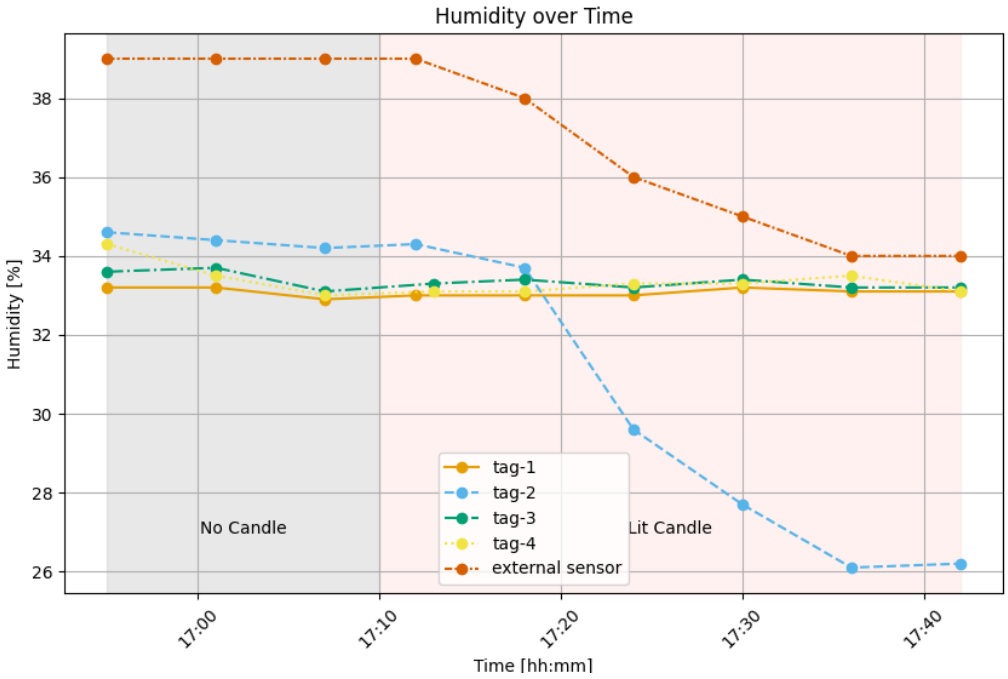


Figure 5.6: Experiment 3, humidity over time, with external measurement added.

Experiment 3 was intended to test the temperature and not the humidity. Luckily, the external thermoeter also included a humidity sensor, that could retroactivly be used for evaluation. Figure 5.6 shows the humidity over time, with a added humidity sensor added to the graph. Since the external humidity sensor was initialy not intended to be used, it is not perticulalry precise and does not display any digits after the decimal point. The humidity sensor consitently shows a much higher humidity than the one on the tag. Once the experiment starts at 15.10, the humidity behaves inversly to the temperature and starts falling. This happens with the external sensor as well as the internal one in parallel. The registered values plato at 34% for the external and 26% for the internal sensor. The other tags do not report any significant change in humidity.

5.2.3 Experiment 4: Gyroscope

Experiment 4 was intended to check the functionality of the gyroscope. Temperature and humidity behaviour was the same as in the static experiment 5.2.1. As already seen in during the evaluation of experiment 1, the gyroscope does not work as planned. Figure 5.7 shows the values of the gyro over time. Tag 1 was rotated by 90Â° at 22.25 around the Z axis. Their is no disernable change in the output of the gyro during or after this process.

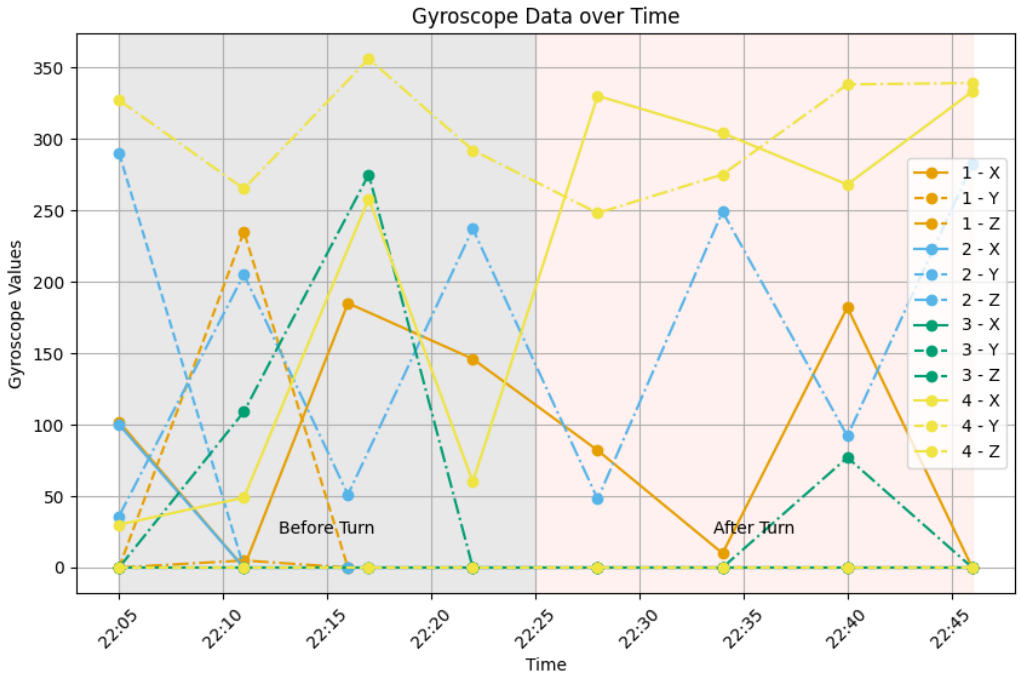


Figure 5.7: Experiment 4, gyroscope over time.

Figure 5.8 shows the distances of the tags during the experiment. Before the event, all tags are in a stable state. As in experiment 1 5.2.1 the distances do not represent what is physically happening. After the tag is turned at 22.26, all measurements involving tag 1 change, and becomming stable again afterwards. This can be bit hard to see, since ”2-1” and ”3-1” have an outlier measurement right before and ”1-2” right after. Distance 1 to 2 and 1 to 3 changes between 0.2 and 0.3 meters and distance 1 to 4 changes by arround 0.4 dm.

5.2.4 Experiment 5: Distance

Experiment 5 was intended to test the distance measurement capabilities of the setup. Temperature and humidity and gyro behave as they do in experiment 1 5.2.1. They will not be discussed for this experiment.

Figure 5.9 shows the measured distances of the 4 tags over time. As in the static experiment, the measured distances of two devices are similar and mostly stable, before any movement is introduced. As in experiment 1 the values reported are not correct. At 14.24 tag 1 is moved by 0.23 meters toward tag 2. The measured distances to tag 3 increases while the distance to tags 2 and 4 dicreases. This represent what is happening in reality, since tag 1 is now closer to tag 2 and 4 and further away from tag 3 as before. The difference in distance is roughly 0.2 meters for tag 2. This is correct, since tag 1 was moved about that distance towards tag 2. The measurements show tag 4 now 0.15 meters closer to tag 1. The effect on tag 4 should be notissable but not as large as it is. Since the tag moves lateraly towards tag 4, the difference should only be 0.11 meters. The same is

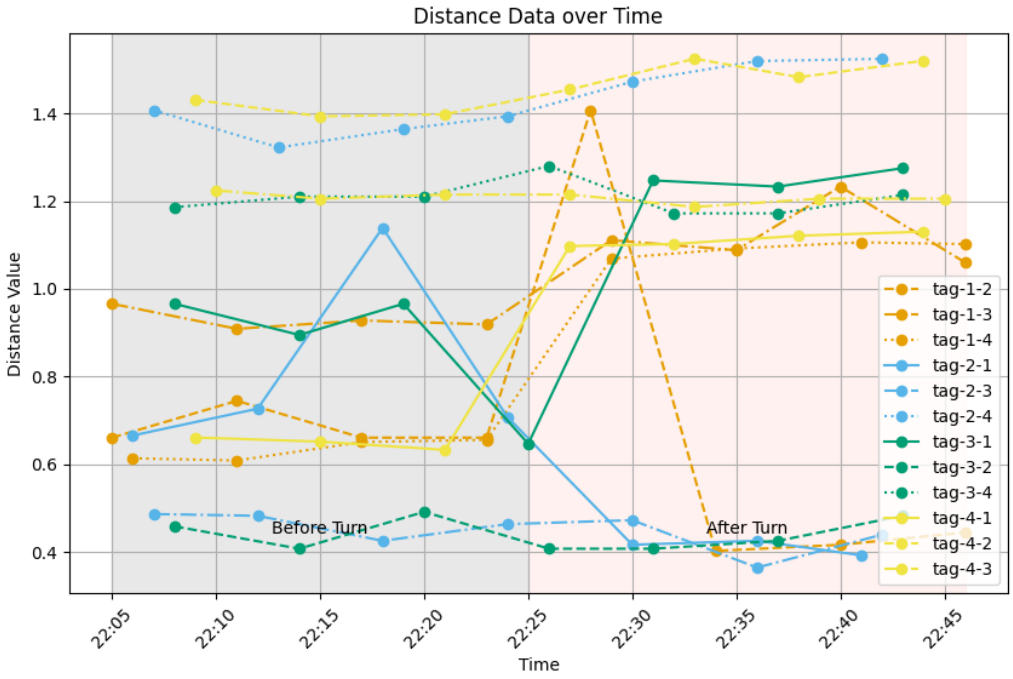


Figure 5.8: Experiment 4, gyroscope over time.

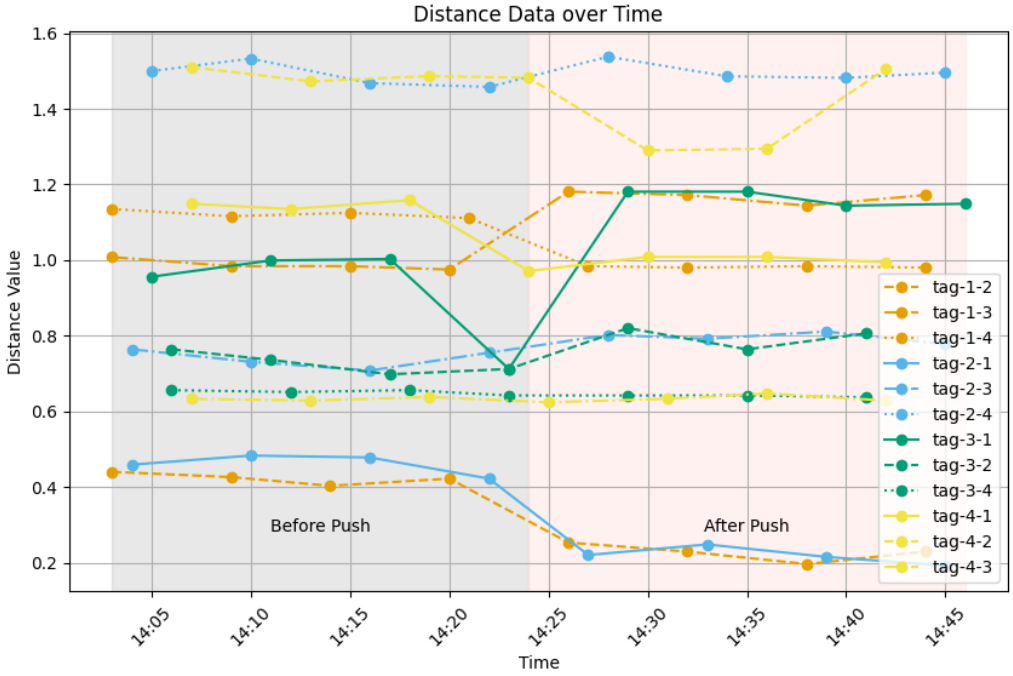


Figure 5.9: Experiment 5, distance over time.

true for tag 3. The difference in measured distance between 1 and 3 is between 0.15 and 0.2 meters. This is too large for the difference a latteral move, it should only be a 0.02 meters difference. Their is also a small increase in the distance between tags 2 and 3, but which starts before tag 1 was moved.

Chapter 6

Final Considerations

6.1 Summary

6.2 Conclusions

6.3 Future Work

Bibliography

[1] E. Hsu, "An overview of the IEEE 802.15.4 HRP UWB standard," https://blogs.keysight.com/blogs/tech/rfmw.entry.html/2021/07/28/an_overview_of_IEEE-J7ac.html, 2021, (Accessed: 2022-12-22).

[2] "IEEE standard for low-rate wireless networks," C/LM - LAN/MAN Standards Committee, IEEE, 2020.

[3] "IEEE standard for low-rate wireless networks–amendment 1: Enhanced ultra wideband (UWB) physical layers (PHYs) and associated ranging techniques," C/LM - LAN/MAN Standards Committee, IEEE, 2020.

[4] "Getting back to basics with ultra-wideband (uwb)," Qorvo US, Inc, Tech. Rep., 2021.

[5] M. F. Mecklenburg and C. S. Tumosa, "Mechanical behavior of paintings subjected to changes in temperature and relative humidity," *Art in transit: studies in the transport of paintings*, 1991.

[6] S. Michalski, *Paintings, their response to temperature, relative humidity, shock and vibration*. National Gallery, 1991.

[7] D. Saunders and J. Kirby, "The effect of relative humidity on artists' pigments," *National Gallery Technical Bulletin*, Vol. 25, pp. 62–72, 2004.

[8] B. Borg, M. Dunn, A. Ang, and C. Villis, "The application of state-of-the-art technologies to support artwork conservation: Literature review," *Journal of Cultural Heritage*, Vol. 44, pp. 239–259, 2020.

[9] E. Schito and D. Testi, "Integrated maps of risk assessment and minimization of multiple risks for artworks in museum environments based on microclimate control," *Building and Environment*, Vol. 123, pp. 585–600, 2017.

[10] J. Shah and B. Mishra, "Customized iot enabled wireless sensing and monitoring platform for preservation of artwork in heritage buildings," *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*. IEEE, 2016, pp. 361–366.

[11] E. Landi, L. Parri, R. Moretti, A. Fort, M. Mugnaini, and V. Vignoli, "An iot sensor node for health monitoring of artwork and ancient wooden structures," *2022 IEEE International Workshop on Metrology for Living Environment (MetroLivEn)*. IEEE, 2022, pp. 110–114.

[12] K. Gulati, R. S. K. Boddu, D. Kapila, S. L. Bangare, N. Chandnani, and G. Saravanan, "A review paper on wireless sensor network techniques in internet of things (iot)," *Materials Today: Proceedings*, Vol. 51, pp. 161–165, 2022.

[13] R. K. Garg, J. Bhola, and S. K. Soni, "Healthcare monitoring of mountaineers by low power wireless sensor networks," *Informatics in Medicine Unlocked*, Vol. 27, p. 100775, 2021.

[14] D. Coppens, E. De Poorter, A. Shahid, S. Lemey, and C. Marshall, "An overview of ultra-wideband (uwb) standards (ieee 802.15. 4, fira, apple): Interoperability aspects and future research directions," *IEEE Access*, Vol. 10, pp. 1–23, 2022.

[15] P. Leu, G. Camurati, A. Heinrich, M. Roeschlin, C. Anliker, M. Hollick, S. Capkun, and J. Classen, "Ghost peak: Practical distance reduction attacks against {HRP}{UWB} ranging," *31st USENIX Security Symposium (USENIX Security 22)*, 2022, pp. 1343–1359.

[16] M. Stocker, H. Brunner, M. Schuh, C. A. Boano, and K. Römer, "On the performance of ieee 802.15. 4z-compliant ultra-wideband devices," *2022 Workshop on Benchmarking Cyber-Physical Systems and Internet of Things (CPS-IoTBench)*. IEEE, 2022, pp. 28–33.

[17] aosong, "Am2302 product manual."

[18] Y. A. Ahmad, T. S. Gunawan, H. Mansor, B. A. Hamida, A. F. Hishamudin, and F. Arifin, "On the evaluation of dht22 temperature sensor for iot application," *2021 8th international conference on computer and communication engineering (ICCCE)*. IEEE, 2021, pp. 131–134.

Abbreviations

ABI	Application Binary Interface
AITF	Active Internet Traffic Filtering
AWS	Amazon Web Service
BloSS	Blockchain Signaling System
CIA	Confidentiality, Integrity and Availability
CSIRT	Computer Security Incident Response Team
DDoS	Distributed Denial of Service
DoS	Denial of Service
DNS	Domain Name System
DOTS	Distributed-Denial-of-Service Open Threat Signaling
ETH	Ether (Cryptocurrency)
EVM	Ethereum Virtual machine
IaaS	Infrastructure as a Service
IETF	Internet Engineering Task Force
IoT	Internet of Things
IPFS	Inter Planetary File System
ISP	Internet Service Provider
NFV	Network Function Virtualization
P2P	Peer to Peer
PoA	Proof-of-Authority
PoW	Proof-of-Work
REST	Representational State Transfer
RTT	Round Trip Time
SDN	Software-Defined Networking
SLA	Service Level Agreement
VNF	Virtualized Network Function

List of Figures

List of Tables

Listings

Appendix A

Contents of the Repository

The code repository contains the following content:

Installation

Operation