



Bachelor Thesis

Examensarbete i elektroteknik 15 hp

Investigate options with spectrum scanning applications

Wireless communication

Halmstad University, June 2, 2024
Edwin Bergström



HÖGSKOLAN
I HALMSTAD

Title Investigate options with spectrum scanning applications
Author Edwin Bergström
School Halmstad University
Supervisor Mark Dougherty
Examiner Pererik Andreasson
Period 15-01-2024/02-06-2024
Pages 54
Keywords Bluetooth, SDR, k-means, Bluetooth-sniffer

Sammanfattning

Detta examensarbete undersöker möjligheterna med mjukvarudefinierade radioapparater som ett övervakningssystem genom att övervaka det elektromagnetiska spektrumet. Övervakningssystemet övervakar Bluetooth-bandbredden med en flerkansalmottagare som passivt lyssnar på Bluetooth-paket. Vidare undersöker detta examensarbete möjligheten att implementera en automatisk k-means klustringsalgoritm för att räkna unika enheter i närheten. Bakgrunden förklarar de grundläggande teknikerna som används i Bluetooth och förklarar hur enheter kommunicerar med varandra. Bakgrunden förklarar också den föreslagna mottagararkitekturen och dess teknologier. Avsnitt Relaterat arbete och liknande produkter undersöker olika metoder för att upptäcka mobila enheter och effektiviteten hos k-means-algoritmen. Metoden förklarar hur mottagaren är modellerad, hur Python-skriptet identifierar Bluetooth-paket i bitströmmen och hur de fysiska defekterna samlas in för den automatiska k-means-algoritmen. Slutligen förklarar metoden hur laborationerna genomfördes. I avsnittet Resultat visas mottagarens och k-means-algoritmen prestanda. Diskussionssektionen analyserar resultaten och diskuterar några designfel och hur man eventuellt kan åtgärda dem. Slutligen jämförs slutsatsen målen med de givna resultaten och presenterar framtida arbete för vidareutveckling.

Abstract

This thesis investigates the possibilities of Software-defined radios as a surveillance system by monitoring the electromagnetic spectrum. The surveillance system monitors the Bluetooth bandwidth with a multi-channel receiver that passively listens to Bluetooth packets. Furthermore, this thesis investigates the possibility of implementing an automatic k-means clustering algorithm to count unique devices in the vicinity. The Background explains the fundamental technologies used in Bluetooth and explains how devices communicate with each other. The Background also explains the proposed receiver architecture and its technologies. Section Related work and similar products investigate different approaches to detecting mobile devices and the effectiveness of the k-means algorithm. The Method explains how the receiver is modeled, how the Python script identifies Bluetooth packets in the bit stream, and how the physical imperfections are collected for the automatic k-means algorithm. Lastly, the Method explains how the labs were conducted. The Result section shows the performance of the receiver and the k-means algorithm. The Discussion section analyzes the results and discusses some design flaws and how to fix them potentially. Lastly, the Conclusion section compares the goals with the results and presents future work for further development.

Contents

1	Introduction	1
1.1	Goals and limitations	1
1.1.1	Goals.....	2
1.1.2	Limitations.....	2
2	Background.....	3
2.1	Bluetooth.....	3
2.1.1	Protocol	3
2.1.2	GFSK modulation	4
2.1.3	GFSK demodulation.....	6
2.1.4	Data packets	7
2.2	Software defined radio.....	7
2.2.1	Proposed architecture	8
2.2.2	Polyphase channelizer	9
2.2.3	Power squelch	9
2.2.4	HackRF one.....	9
2.3	Automatic clustering algorithm	11
2.3.1	Automatic k estimation	12
3	Related work and similar products	13
3.1	Electronic circuit designs.....	13
3.2	Bluetooth physical imperfection.....	13
3.3	Side channel attacks.....	14
4	Method.....	15
4.1	Testing and verification of goals.....	15
4.1.1	Requirements and choices	15
4.2	Preparations	17
4.3	Methodology.....	17
4.3.1	Tools.....	17
4.3.2	Modeled GFSK receivers	17
4.3.3	Packet finder.....	20
4.3.4	Frequency demodulator.....	20
4.3.5	Median method.....	21
4.3.6	K-means code.....	22

4.3.7 Testing set-up	22
4.3.8 EMC test set-up.....	24
5 Results	25
5.1 Captured Bluetooth signal	25
5.2 Captured Bluetooth packets in RF chamber	26
5.2.2 Single channel performance	27
5.2.3 Multi-channel performance	27
5.3 Automatic k-means algorithm.....	28
5.3.1 Elbow and Silhouette method	29
5.3.2 EMC results.....	30
6 Discussion.....	33
6.1 Receiver	33
6.2 Data collection and k-means algorithm	33
6.3 Legal boundaries of wireless sniffing	34
7 Conclusion	37
Reference list	I
Appendices	V

List of Figures

Figure 1: FSK modulation

Figure 2: Gaussian filter processing

Figure 3: GFSK modulation architecture

Figure 4: I/Q data

Figure 5: SDR general receiver architecture

Figure 6: Architecture

Figure 7: Signal in complex plane

Figure 8: Positive and negative frequency

Figure 9: Block structure in GNU Radio

Figure 10: Polyphase channelization illustration

Figure 11: LP filter bank parameters

Figure 12: Filtering and demodulation step

Figure 13: Repackage

Figure 14: Single-channel receiver

Figure 15: Phase demodulator

Figure 16: Lab set-up

Figure 17: EMC set-up

Figure 18: RF activity

Figure 19: Captured I/Q data

Figure 20: Demodulated signal

Figure 21: Elbow method results

Figure 22: Optimal K in k-means algorithm

Figure 23: EMC Elbow method

Figure 24: K-means performance in EMC

List of Tables

Table 1: Tools

Table 2: CMW500 single channel test settings

Table 3: CMW500 multi-channel test settings

Table 4: HackRF one settings

Table 5: EMC devices

Table 6: Captured packets in the multi-channel receiver

1 Introduction

Most of today's security cameras operate around the clock and consume unnecessary energy when an event is not occurring. Axis Communications wants to find a reliable way of identifying a person who is present close to the surveillance camera and start filming. One of the most popular solutions to this problem is using a passive infrared (PIR) motion sensor attached to the surveillance camera, which triggers the camera when motion is detected. The issue with this system is that PIR sensors react to changes in infrared radiation where both objects and people can trigger the device; this means that animals, fluttering leaves, and other non-human objects trigger the PIR sensor and are rarely used outside for motion detection. With the implementation of software-defined radios (SDR) to analyze the electromagnetic spectrum, Axis Communications aims to have its security cameras in sleep mode to minimize energy consumption. SDR's purpose is to be a complimentary peripheral and alert the camera when a person with a mobile device is close to the surveillance camera. The necessity to use SDR stems from the company's goal to innovate and explore new methods for efficient surveillance systems.

Many wireless communications are used in the electromagnetic spectrum, such as Wi-Fi, Bluetooth, 4G, and 5G. This thesis will investigate one of the most popular wireless protocols, Bluetooth, which is used in wireless headphones and smartwatches. The Bluetooth protocol is often used on the move for listening to music on headphones and smartwatches sending biometric data to the phone; this makes it a suitable protocol for analyzing and detecting activity in the electromagnetic spectrum. Therefore, the company created this project to investigate the possibilities of how effective such a system would be. Additionally, this thesis aims to use automatic clustering algorithms to categorize devices from their physical imperfections found in the transmitted signal and count the formed clusters as unique devices.

Section 1.1 explains the goals and limitations of this thesis. Section 2 presents the background of this thesis, such as the Bluetooth technology, SDR receiver architecture, and clustering algorithm. Section 3 investigates related work and products on the market. Section 4 presents how the receiver was modeled, the code used in the project, and how the labs were conducted. Section 5 presents the performance of the Bluetooth receiver and the automatic k-means algorithm. Section 6 discusses the project, lifts some issues in the current implementation, and briefly investigates the GDPR act. Section 7 compares the goals set in Section 1.1 with the results achieved and mentions what work can be done for the future of this project.

1.1 Goals and limitations

This thesis aims to create a Bluetooth receiver on an SDR, where a captured Bluetooth packet signals that a person is in the area, and investigate if it's possible to count how many devices are close to the receiver by collecting the physical imperfection of the transmitted signal. As this thesis is a new topic for Axis Communications, this work

will be a proof of concept, where this technology will be tested in a controlled environment to confirm that the implementations work and are worth developing further.

1.1.1 Goals

- Create a multi-channel Bluetooth receiver to capture packets.
- Program a script that analyses the captured bits and finds Bluetooth packets.
- Collect the transmitted signal's center frequency offset (CFO) and frequency deviation data.
- Use the collected data and investigate if an automatic k-means algorithm can identify how many unique devices are transmitting.

1.1.2 Limitations

- This thesis will only investigate Bluetooth Classic Basic Rate (BR) due to the time constraint of modeling all the different receivers required to cover all the modulation used in Bluetooth.
- Experiments are made in laboratory environments where outside radio frequency (RF) interference is blocked; no experiments are made in real-life environments due to the k-means algorithm being an unsupervised learning model. The number of devices needs to be known to draw clear conclusions.
- Only five Bluetooth devices are used in the data set for the automatic k-means algorithm, as it was hard to source more devices.

2 Background

This section describes how Bluetooth technology works, from how the binary data is modulated in the transmitter, how the signal behaves in the spectrum, and how the signal demodulates and becomes binary data. Furthermore, this section describes the proposed receiver architecture for the SDR and explains the essential techniques used to receive Bluetooth signals. Lastly, this section describes how K-mean clustering works and what method to use for automatic clustering; the reason for choosing this algorithm and receiver architecture is explained in section 3.

2.1 Bluetooth

Bluetooth technology has revolutionized the way devices communicate wirelessly, making it an essential component in modern electronics. Understanding its fundamentals and operational principles is crucial for developing effective surveillance systems using SDRs.

Bluetooth Classic is mainly used for wireless audio streaming and is the standard radio protocol for wireless devices such as wireless speakers, headphones, and in-car entertainment systems. The frequency band used for Bluetooth Classic is the Industrial, Scientific, and Medical (ISM) band, which is between 2.402 – 2.480 GHz. It's divided into 79 channels with 1 MHz spacing and communicates with a point-to-point communication topology [1].

Similarly, Bluetooth Low Energy (LE) uses the same ISM band but has 40 channels instead with 2 MHz spacing, where 3 are advertising channels and 37 are data channels. It uses multiple communication topologies as a mesh [1]. The mesh communication is implemented where various devices need to communicate with each other [2].

2.1.1 Protocol

The Bluetooth MAC protocol is based on a master (paging unit) slave (recipient unit) mechanism where a Bluetooth piconet consists of one master and up to seven slaves [3]. When the devices are not in a piconet, they enter standby mode and listen to page messages where the spectrum is 79 hop carriers, and 32 of them are wake-up hop carriers. Then, a wake-up sequence visits each hop carrier once. Units in standby mode move their wake-up hop carrier forward one step every 1.28 seconds in their wake-up sequence. The listening interval is 11.25 ms, where the unit listens on a single wake-up hop carrier and correlates the signals with its access code. If this correlation matches, the units enter a connection-setup procedure; otherwise, the units return to sleep until the next wake-up event. A master trying to connect to a slave in standby mode needs to know the slave's identity and its native clock to generate the access code, derive the wake-up sequence, and predict the phase of the sequence. This is done by sending out paging messages to the wake-up carriers, and when the master and slave choose the same wake-up carrier, the slave receives the access code and sends an acknowledgment.

Then, the master sends a package containing its identity and current clock; with this information, the master and slave use the parameters for hop selection and establishing a piconet. After that, to establish a connection, the master sends an inquiry access code to the wake-up carrier, and when the slave receives it, the slave sends back its identity and clock and creates a connection [4].

During communication, Bluetooth transmits packets through the frequency hop spread spectrum (FHSS) in a 625 μ s interval, and this is called the frequency hop / time-division-duplex (FH/TDD) scheme; only one packet can be transmitted per interval. This technique achieves a nominal hop rate of 1,600 hops per second, and each packet consists of an Access code (72 bits), a Packet header (54 bits), and a Payload (0-2745 bits). Bluetooth supports multi-slot packets for higher data rates where the packet will be transmitted over the same carrier and hop to the carrier frequency in the sequence, depending on how many slots the packet occupies [4].

2.1.2 GFSK modulation

Modulation techniques play a critical role in wireless communication by determining how data is transmitted over the airwaves. Gaussian Frequency Shift Keying (GFSK), the modulation scheme used by Bluetooth Classic, offers several advantages in terms of bandwidth efficiency and signal robustness. This section will explore the principles of GFSK modulation and its specific application in Bluetooth technology.

To achieve a 1 Mb/s transfer rate, Bluetooth uses GFSK modulation [1]. Frequency Shift Key (FSK) works by shifting the carrier frequency, which is the center frequency, in a binary manner to symbolize zeros and ones; higher frequency is categorized as a binary one, and lower frequency is a binary zero. The maximum deviation for a 1 Mb/s data rate is 185 kHz [5][6]. Figure 1 shows a simulation of how binary data is modulated with FSK; the carrier frequency is set to 2.402 GHz, and the deviation frequency is larger than the maximum 185 kHz to enhance the visualization.

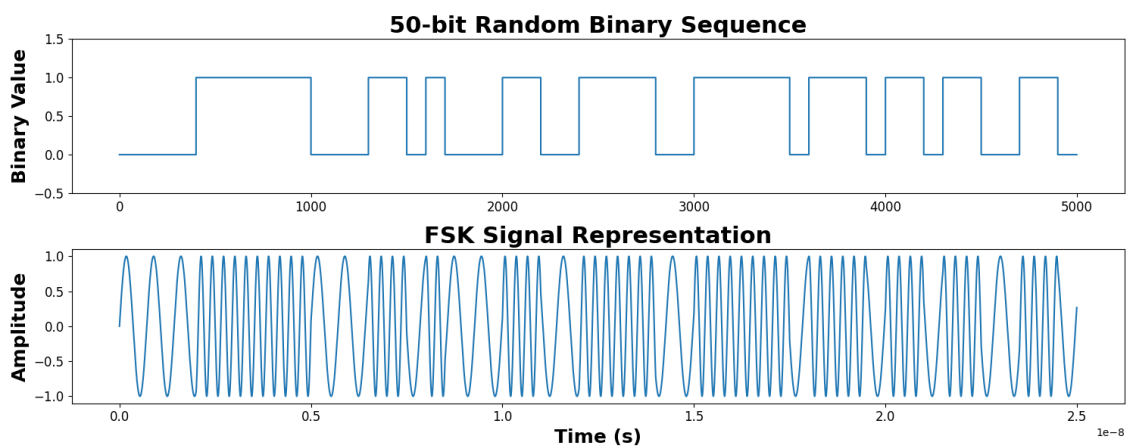


Figure 1: FSK modulation

The GFSK modulation uses a Gaussian filter before the frequency generator, and the impulse response of a Gaussian filter is:

$$h(t) = \frac{1}{\sqrt{2\pi\sigma^2 T}} e^{-\frac{t^2}{2\sigma^2 T^2}} \text{ where } \sigma = \sqrt{\ln(2)} / 2\pi BT \quad (1)$$

The bandwidth time product (BT) in Bluetooth is 0.5 [7]. In Figure 2, the bit sequence was processed by a Gaussian filter; after this step, the GFSK modulation follows the same modulation as FSK.

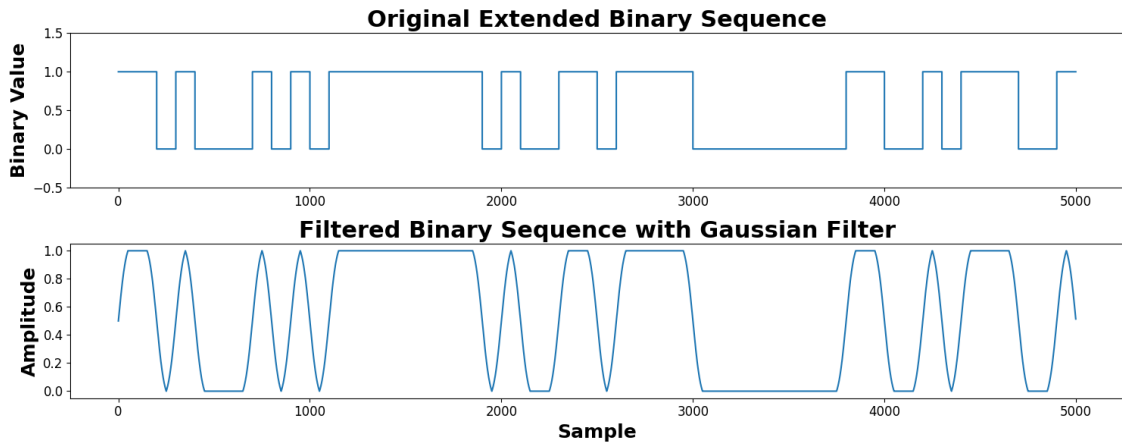


Figure 2: Gaussian filter processing

Bluetooth uses GFSK modulation to smooth out the frequency transmission, which reduces the noise and narrows the spectral width; this reduces interference with other frequencies [7].

The complete GFSK modulation architecture is shown in Figure 3, where the bitstream gets processed by the Gaussian Filter, and then the signal is integrated. Afterward, the signal gets multiplied with a zero-degree phase signal from the local oscillator (LO), which is denoted as the in-phase signal (I). The same signal is multiplied by a 90-degree phase signal, denoted as the quadrature signal (Q). Then, the I and Q signals are added together and transmitted from the antenna [7]. The antenna output will look like the FSK signal representation in Figure 1.

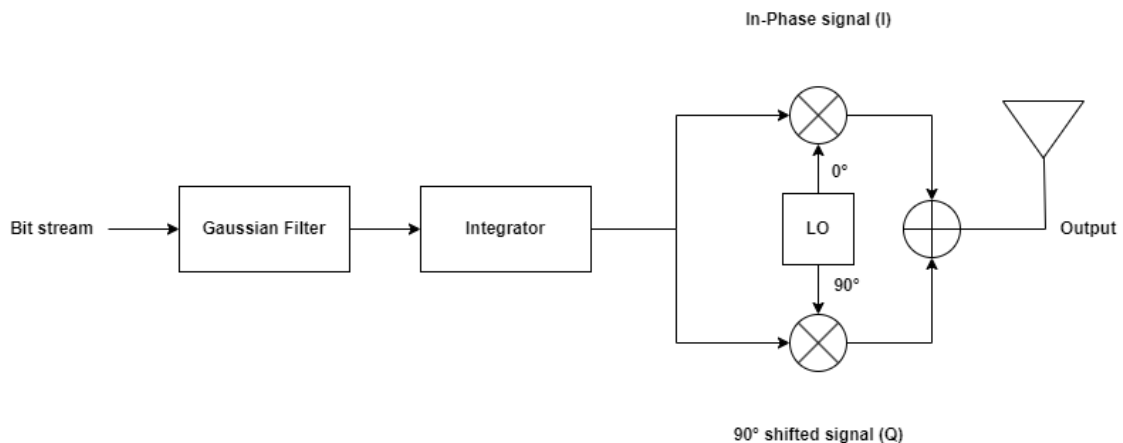


Figure 3: GFSK modulation architecture

Then, the transmitted I/Q signal can be sampled, and the real part I and complex part Q can be extracted; this is done by using the Pythagoras theorem to retrieve the signal's amplitude at that time. Let's say that the signal's real part is $I = 0.69$ and $Q = 0.4$. Then, the amplitude of the signal will be 0.8, and by using trigonometry, the angle will be 30 degrees, so the final signal is $0.8 \cos(30)$. Figure 4 shows how I/Q data is plotted down the time axel [8].

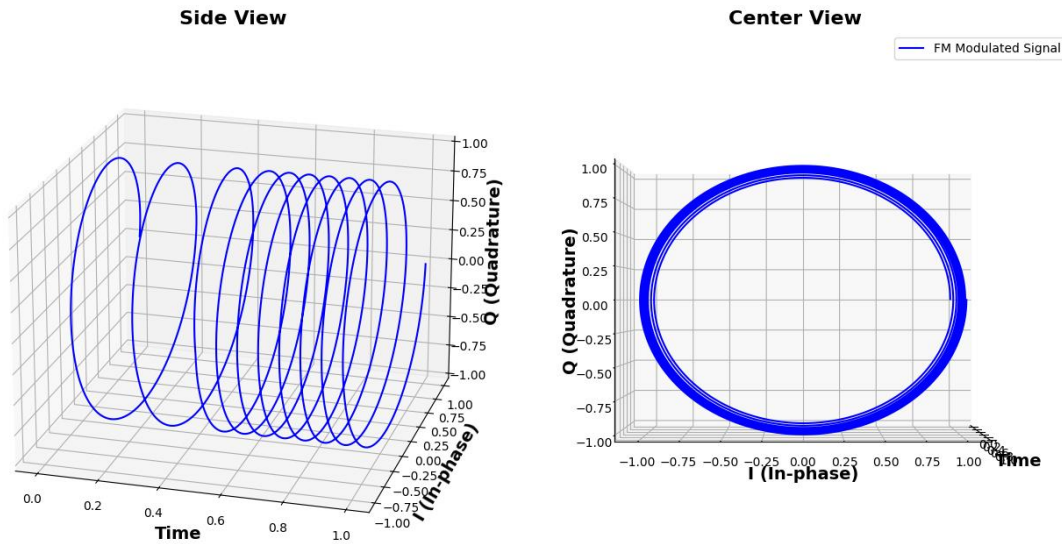


Figure 4: I/Q data

2.1.3 GFSK demodulation

Demodulation is the process of extracting the original data from a modulated carrier signal. In Bluetooth Classic, which uses GFSK, demodulation is crucial for decoding the transmitted information. This section explains how GFSK demodulation works.

There are several different GFSK demodulators for Bluetooth Classic, and the standard designs are analog-digital hybrids [9]. To obtain the Intermediate Frequency (IF) before modulation, the RF signal is amplified by a Low Noise Amplifier (LNA) and then down-converted by an Image Rejection Mixer (IRM) to the desired IF. A gain-compensated bias scheme is used to tune the LNA/mixer gain accurately, and then a cascaded second+fifth-order Butterworth Band Pass Filter (BPF) chooses the desired channel [10].

To eliminate the need for an external clock pulse for sampling, a discrete-time GFSK demodulator can be used; it uses the transmitted signal as its sampling clock. This architecture consists of limiting amplifiers, lowpass filters, discrete-time Quadri correlators, and detection circuits. The limiting amplifiers are implemented by multistage amplifiers with servo loop dc-offset cancelation, and the lowpass filter is a Sallen-Key filter, which is designed with 4-MHz corner frequency [11].

The time-domain differentiation technique works by passing the signal through exclusive-OR after the limiting amplifiers and adding a delay, the demodulated signal follows this equation:

$$v \approx \pi k_d (m(t) - m(t - T_d)) \quad (2)$$

Where the gain πk_d is about 0.5, the $m(t)$ is the digital GFSK baseband data, and the T_d is the delay. This means that the difference of the digital data at time t and $t - T_d$ is equal to the output of the demodulator, and thus, the delayed IF can be used to sample the IF signal in the mixed quadrature demodulator. Lastly, the output of the demodulator is connected to a comparator with hysteresis that sets/resets the data according to the polarity of the pulses [11].

2.1.4 Data packets

This section will explain the structure of Bluetooth BR packets and what patterns there are in the packet that can be used to identify if a Bluetooth packet is present in the bit stream.

The most reliable way to identify unique devices is to save the MAC address, but this address is only available when devices are pairing. To work around this issue and be able to count unique devices during the connection phase, the software needs to know the difference between an empty data signal and a Bluetooth signal. When a signal with Bluetooth data is captured, the software can use the physical characteristics and Machine Learning (ML) algorithm to determine the number of devices; the physical characteristics and ML algorithm are explained in section 3.

The first step to detecting a Bluetooth signal is to match the received signal to the known Bluetooth frame format. The sync word contains a 4-bit preamble, a 64-bit sync word, and a 4-bit trailer at the end. The preamble is a fixed sequence of either 1010 or 0101, depending on whether the following access code starts with 1 or 0. The trailer is configured similarly and depends on the most significant bit (MSB) of the sync word. Although the sync word is unique for the piconet, it cannot be used as a fingerprint for devices because a piconet can have up to 7 slaves and one master. After the access code comes the header that contains Logical Transport-Address (LT_ADDR) 3-bit, Type 4-bit, Flow 1-bit, ARQN 1-bit, SEQN 1-bit, and HEC 8-bit, and each bit is repeated three times so the header comes out to 54-bit, lastly comes the Payload [12].

By knowing the Bluetooth frame format, software can be created that searches for these characteristics in the bit stream capture and classifies whether a unique Bluetooth signal is present.

2.2 Software defined radio

SDR is a flexible and powerful technology that allows for the implementation of radio communication systems using software rather than traditional hardware components. This section explores the key aspects of SDR relevant to the project. First, this section introduces the concept of SDR and the advantages it offers in terms of adaptability and

reconfigurability. Following this, the proposed architecture is presented for the Bluetooth receiver, which utilizes multiple stages to ensure efficient signal processing and accurate data extraction. Additionally, this section explains the techniques of power squelch and polyphase channelizer, which are essential for improving signal quality and managing bandwidth. Finally, this section discusses HackRF One, the hardware platform used for our SDR implementation, detailing its features and role in the project.

SDR's flexibility comes from the digital signal processing and digital back end, which all operate in the digital domain. The architecture of an SDR contains the following components: a radio frequency (RF) front end, a digital back end, and a signal processing step. All these components can be programmed to receive and transmit different RF signals [13]. In this thesis, the SDR mimics a GFSK receiver that captures multiple channels at the same time and demodulates incoming Bluetooth signals. Figure 5 shows a simple overview of the SDR architecture as a receiver [14].

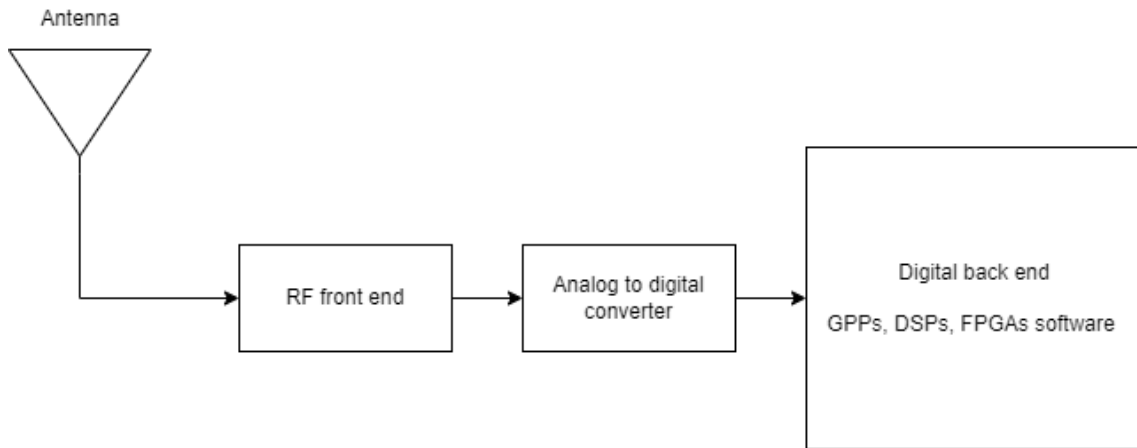


Figure 5: SDR general receiver architecture

2.2.1 Proposed architecture

The SDR captures the spectrum, and a Polyphase channelizer processes the data and splits the spectrum into multiple channels with a channel width of 1 MHz; here the signal will be filtered to remove noise and set the channel parameters. Before the GFSK demodulation, there is a power squelch that only lets through a signal if it goes over the power threshold; this is done to not demodulate the noise floor and only the desired signal. Then, a bit slicer takes the demodulated signal, slices it into 1 and 0, and stores the data stream into a file. Figure 6 shows a visualization of the architecture.

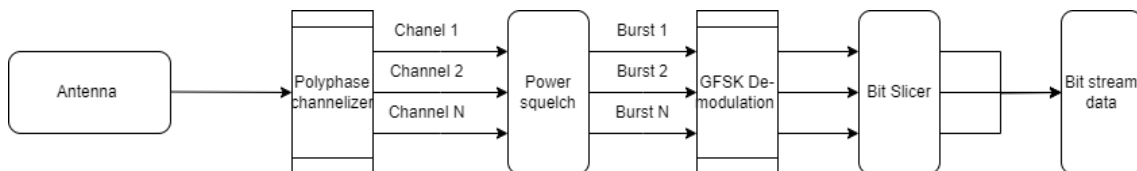


Figure 6: Architecture

2.2.2 Polyphase channelizer

To demodulate a channel, the wideband signal needs to be split into channels, and an effective method is to use a polyphase digital channelizer (PDC). The wideband signal splits into M equally spaced channels and downsamples them to $\frac{1}{M}$ of the original sampling frequency of the wideband signal; after that, the signal goes through a low pass filter (LPF). The last step is the signal processed by the Discrete Fourier transform technique, and after that, the wideband signal is channelized and can be further processed [15].

2.2.3 Power squelch

The purpose of the power squelch is to attenuate the output noise when there is no input signal. There are two ways that a power squelch can be implemented; the first is a simple approach in which the threshold is manually set, usually above the noise floor, and when the RF signal is picked up, the signal's power goes over the threshold. The squelch lets through the signal so it can be demodulated. The second approach uses automatic gain control (AGC) with a squelch.

The AGC estimates the gain required to force a signal to have a unity target energy, and by sampling the signal, it estimates the \hat{e} of the signal energy and updates the gain g . The notation $\hat{\cdot}$ above the variables symbolizes an estimation of the variable. This is done by the following formula.

$$E\{\|\vec{x}\|\} = \left[\sum_{k=0}^{N-1} \|x_k^2\| \right]^{\frac{1}{2}} \quad (3)$$

where the input signal is $\vec{x} = \{x_0, x_1, x_2, \dots, x_{N-1}\}$ and to control the gain, a loop filter is used and has the following transfer function:

$$H_g(z) = \frac{\alpha}{1-(1-\alpha)z^{-1}}, \quad \alpha = \sqrt{\omega} \quad (4)$$

To get the instantaneous ideal gain is to inverse the estimated signal level:

$$\hat{g}_k = \sqrt{\frac{1}{\hat{e}_k}} \quad (5)$$

and before applying the gain, the signal is first filtered as:

$$g_k = \alpha \hat{g}_k + (1 - \alpha)g_{k-1} \quad (6)$$

After that, a squelch is used and operates the same way as the first method [16].

2.2.4 HackRF one

The SDR used for the thesis is the HackRF one created by Great Scott Gadgets; it is relatively affordable and can receive RF between 1 MHz and 6 GHz, which makes it able to listen to Bluetooth signals. With the HackRF one having a maximum sampling frequency of 20 MHz [17], the maximum number of channels possible is 20 due to the Nyquist theorem that the sampling frequency needs to be twice as large as the signal frequency [18]. This is due to down-converting the baseband to zero Hz; the maximum

signal frequency will be at -10 and 10 MHz, where the maximum signal frequency will be sampled at 20 MHz, which is twice the frequency, thus satisfying the Nyquist theorem.

To understand how negative frequencies work, a signal can be represented in the complex plane with Euler's identity using the following formulas.

$$e^{i\omega t} = \cos \omega t + i \sin \omega t \quad (7)$$

$$e^{-i\omega t} = \cos \omega t - i \sin \omega t \quad (8)$$

Where equation 7 represents a positive frequency and equation 8 represents a negative frequency. Thus, the difference between negative and positive frequencies is the exponent $\pm i\omega t$ that determines the angular change depending on time [19]. Figure 7 shows a representation of how these signals change over time.

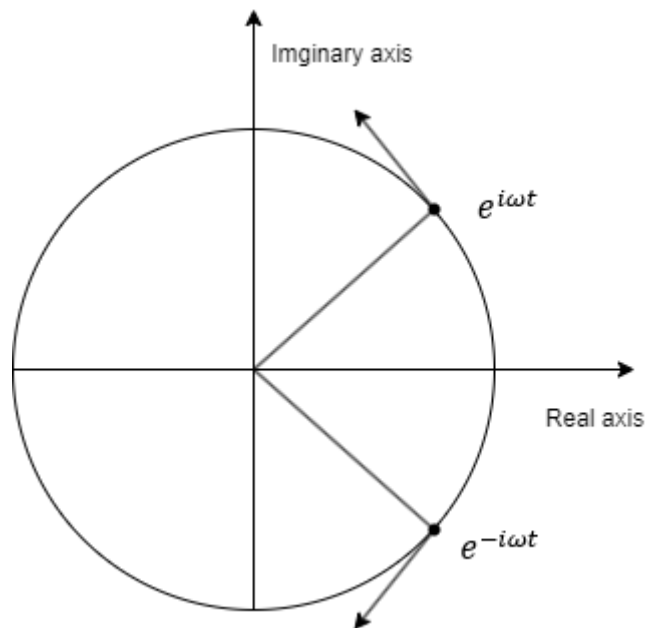


Figure 7: Signal in complex plane

A cosine wave is represented with the following formula [19]:

$$x(t) = \cos(\omega_0 t) = \frac{1}{2}(e^{i\omega t} + e^{-i\omega t}) \quad (9)$$

That means a clean cosine wave will have no imaginary part, as can be observed by adding equations 7 and 8, where the imaginary part cancels out. This means there is only a real part in the signal, and plotting Figure 7 over time creates two identical signals, as seen in Figure 8.

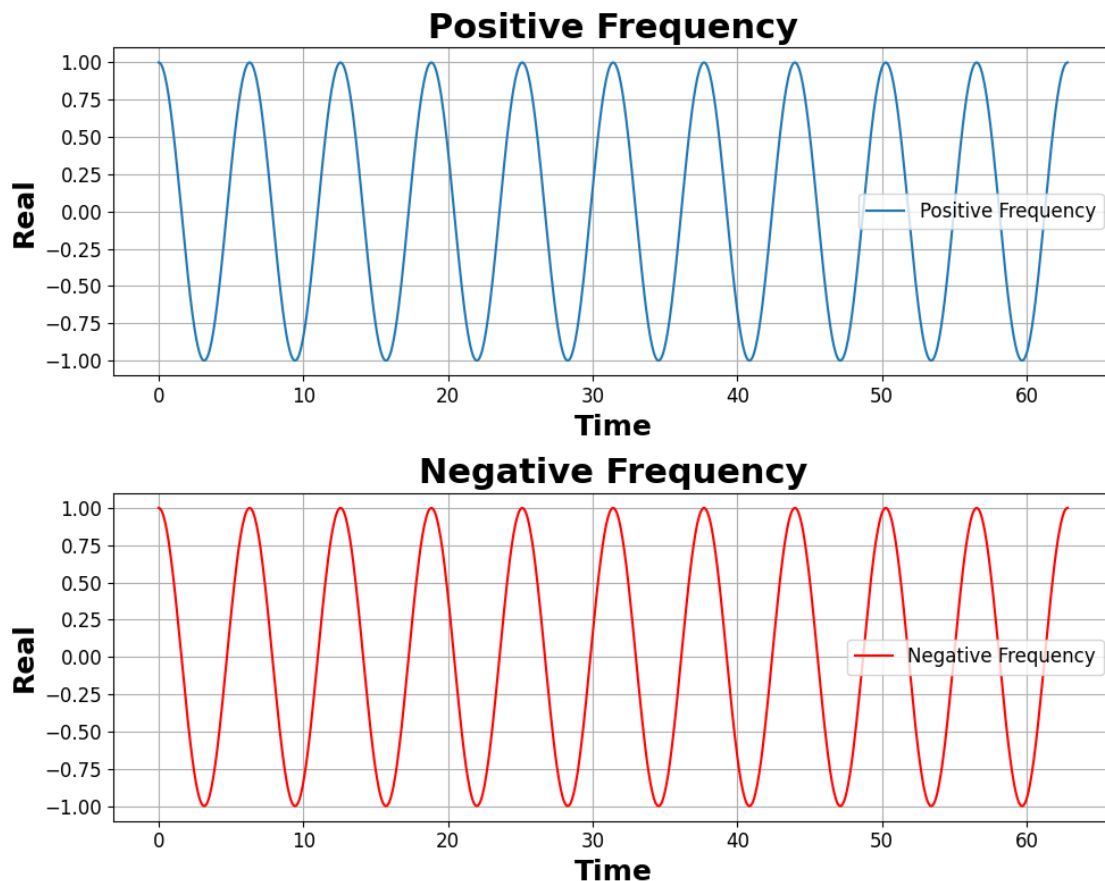


Figure 8: Positive and negative frequency

This means that the SDR can down-convert the captured signal to a negative frequency without losing information about the signal.

As Bluetooth uses 80 MHz bandwidth, the HackRF one will not be able to capture the whole spectrum at once, but this will not hinder the process of detecting Bluetooth signals due to the FHSS technique utilized; instead of capturing 1600 hops per second, the SDR will capture 400 hops per second which are enough to detect Bluetooth devices nearby. A straightforward way to listen to the whole spectrum is to use multiple HackRF ones.

2.3 Automatic clustering algorithm

This section explains the mathematics behind the k-means algorithm and introduces methods on how to estimate the optimal number K with the Elbow and Silhouette method.

The k-means algorithm is an unsupervised learning algorithm that groups data points into K number of disjoint sets by calculating the distance between the data points and the data centers called centroids. A centroid is the center of gravity where data points sharing similar characteristics gather, and one of the most common aggregation functions used is the arithmetic mean. The function is as follows: Let E be a

set of data points i of size n , and each of them have a p value that is $x = \{x_{i1}, \dots, x_{ip}\}$ and the centroid of this set of data points is $c = \{c_{i1}, \dots, c_{ip}\}$ where:

$$c_{ik} = \frac{1}{n} \sum_{i=1}^n x_{ki} \quad k \in (1, \dots, p) \quad (10)$$

To determine the distance between data points and a centroid, a subgroup of E is called R , where it is necessary to measure the resemblance between the data point i and a centroid c in the group of data points R . The centroid c , when compared within the group R , and a point j belonging to E , are considered most similar when their distance is the minimum [20]. In traditional k-mean algorithms, the K needs to be specified before classifying objects, and the function is:

$$\sum_{k=1}^k \sum_{i \in k} \|x_{ik} - C_{ik}\|^2 \quad (11)$$

Where the $\|x_{ik} - C_{ik}\|^2$ is the Euclidean distance between the data points and the centroid [20]. In this thesis, the k-mean algorithm must independently determine the K number cluster, which will determine the number of devices in the area. One of the ways to estimate the number of k is to use the Elbow method or Silhouette.

2.3.1 Automatic k estimation

The Elbow method is the oldest method for determining the optimal cluster numbers in a dataset, where it calculates the within-cluster sum of squares (WSS) for each k starting from 1 and then increases the number of clusters by 1. The elbow method is a visual method where the WSS is plotted against the number of clusters, and the optimal number of clusters is located at the elbow point, where the graph stops decreasing rapidly and flattens out [17].

The Silhouette method can estimate K automatically by calculating the silhouette width coefficient with the following formula.

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \quad (12)$$

Where $a(i)$ is the average distance between the i and all other entities of the cluster to which i belongs. The $b(i)$ is the minimum average between i and all the entities in each other cluster. The silhouette width value range is between 1 and -1; if the value is close to -1, the entity is misclassified, and if the value is close to 1, the entity is well clustered [21].

For this thesis, both methods will be used for analysis to compare if they give out similar results and compare them with the number of Bluetooth connections, which will be a known variable in the experiments.

3 Related work and similar products

This section investigates if it is possible to detect handheld devices with an electronic circuit and how effective it is. Moreover, it goes through related work in the field, such as fingerprinting BLE signals by their physical imperfection and distinguishing if the captured RF activity is a Bluetooth packet. This section also discusses the effectiveness of the k-means algorithm in Side Channel attacks (SCA) to retrieve data from power traces emitted from hardware.

3.1 Electronic circuit designs

The project "Design and Testing of Mobile-Phone Detectors" by Edwin Ataro and Diana Starovoytova (2016) presents designs of two different detectors: a mobile phone detection system with a capacitor and resistor circuit and a reed switch circuit scanner. The mobile phone detection scanner has a capacitor that stores energy in its electromagnetic field, and when a mobile phone radiates, the capacitor's field oscillates and creates an alternative current (AC) that signals detection. Furthermore, the reed switch circuit scanner has a sensing element that consists of two ferromagnetic nickel-iron wires and a glass capsule, and when a magnetic field is introduced, the circuit will close and signal a detection. The detection radius for the mobile phone detection system was about one meter and the reed switch circuit scanner had only a detection radius of one to five centimeters [22].

The paper explains how a detection circuit works and the limitations of the design. This approach is designed to be close to the mobile phone and would not be beneficial for a surveillance camera that would be needed to wake up if an event occurred several meters away.

3.2 Bluetooth physical imperfection

The paper "Evaluating Physical layer BLE Location Tracking Attacks on Mobile Devices" investigates the possibility of fingerprinting BLE devices from their unique physical-layer characteristics CFO and I/Q Offset. The CFO imperfection is the offset generated on the carrier frequency from the RF frontend's local oscillator and yields a unique CFO on every transmission. The I/Q imperfections are caused by two different reasons, the first is the I/Q offset where the carrier frequency signal leaking into the transmitted signal or the DC offset in the baseband signal. This results in a fixed complex term and I/Q sample. The second imperfection is due to the mismatch of the analog components in the RF chain, changing the phase and amplitude of the received I/Q signal. Then the paper estimates these offsets for specific device and calculates the Mahalanobis distance of the device and compares them to successfully identify the device. Figure 6 in the paper shows diagram where the I/Q offset Magnitude is plotted against CFO, this graph shows that the data point with the same chipset groups with similar physical properties [23].

Mike Ryan from ICE 9 consulting similarly found the same concept where he plotted the FSK Deviation against the CFO and showed that unique devices physical

parameters groups together [24]. As both of these works shows that the physical imperfections tend to create cluster of data points this thesis will investigate the implementation of automatic K in k-means algorithm to count the unique devices. Reason that this thesis will not use the same approach as the paper “Evaluating Physical layer BLE Location Tracking Attacks on Mobile Devices” to calculate the Mahalanobis distance for fingerprinting is because Bluetooth BR does not have that feature [1]. Moreover, the aim of this thesis is not to track the devices and only to identify how many devices are in the area of interest.

Mike Ryan also goes in detail on how on every step of the process is done to capture the signal and detecting BLE signal which is the inspiration of the proposed receiver in section 2. Mike Ryan also mentioned a method of detecting Bluetooth packets in a bit stream where the software searcher for the 0101 or 1010 preamble of the package and compares the package data with the burst time [24]. For this thesis this method will be altered to work for BR packets.

3.3 Side channel attacks

The paper” Advantages of unsupervised learning analysis methods in single-trace SCA attacks” investigate the effectiveness of comparison to the mean statistical analysis method and the unsupervised machine learning algorithm K-mean to uncover vulnerabilities. They focused on Elliptic Curve Cryptography (ECC) point scalar multiplication (kP) operation, which is vulnerable to SCA due to its power consumption patterns. The paper measured the power traces with current probe on the hardware that simulated the kP operations. In the results of this paper, they compared the simulated scalar used in the ECC point multiplication operation with the predicted bits of both methods. This paper classified balanced keys as containing 40% to 60% one bits and both methods achieved 100% correctness. However, the k-means algorithm showed superior performance on unbalanced keys but had degraded performance when the data contained mostly zeros or ones. The paper also state that k-means is more sensitive to outliers than the comparison to the mean method [25].

This paper demonstrates the capabilities of k-means algorithm and how effective it is at classifying physical parameters emitted from devices, although this paper used the regular k-means without automatic k estimation because the aim of the paper were to classify zeros and ones. This thesis will investigate the possibility of using automatic k estimation for k-means algorithm to classify unique devices from the physical imperfections in the Bluetooth signal mentioned earlier.

4 Method

This section explains how the implementation will be tested and verifies whether the methods succeed. Furthermore, this section describes how the receiver is modeled on GNU radio, how the Python scripts identify Bluetooth packets, collect physical imperfection data, and how the automatic k-means algorithm is implemented. Lastly, this section introduces how the laboratory tests were set up. The implementation of the receiver and code can be found in Appendix A.

4.1 Testing and verification of goals

One of the metrics used to evaluate the performance of a receiver is to calculate the packet loss ratio, which is the ratio between the number of packets lost and the total number of packets sent [26]. This thesis will alter this formula to calculate the ratio between the number of captured and sent packets. This result will later be used to calculate how fast a packet detection will take. The performance is calculated using the following equation.

$$performance = \frac{captured\ packets}{t * frequency\ hops\ per\ second} \quad (13)$$

Where t is time in seconds and where frequency hops per second is 1600.

The software will count each captured Bluetooth BR packet. Furthermore, the R&S@CMW500 wideband radio communication tester with an RF chamber at Axis Communications will be used to verify that a single device emits a Bluetooth BR packet. The CMW500 will create a piconet with an Axis W101 body-worn camera and send BR packets with known packet types and payload content.

The automatic k-means clustering algorithm will be tested in the EMC chamber to minimize outside interference and evaluate performance reliably. The EMC chamber is the standard for emission testing due to the chamber eliminating signals and ambient noise from the outside [27], which is crucial for accurate measurements inside the EMC chamber; multiple Bluetooth connections will be set up, streaming packets continuously between devices to monitor if the automatic k-means algorithm groups the signals effectively. This will be measured by counting the number of clusters generated and comparing it with the number of connections active. Furthermore, this thesis will investigate if Bluetooth transmitters manufactured by the same manufacturer have a large enough physical signal imperfection difference for the k-means algorithm to be effective. For example, streaming audio from two devices made by the same manufacturer and monitoring if the algorithm can differentiate them.

4.1.1 Requirements and choices

This section explains the choices and requirements for each part of the project and motivates the reasoning behind each choice.

The requirement for the receiver is that it should be able to demodulate the incoming Bluetooth packet and confirm that the signal is a Bluetooth signal. The

performance is calculated so the detection time can be calculated and then evaluated if the expected detection time is fast enough to detect a person.

The requirement for the automatic k-means algorithm is to investigate if this ML algorithm can be used for counting unique devices and to evaluate if the data collected can be used for connection counting.

These are the choices made in the methodology.

Multi-channel receiver:

- Made in GNU radio, as it has a large selection of pre-programmed SDR components and a feature where custom blocks can be added to the program [28]. The custom blocks will be used to implement the median method to collect data for the automatic k means algorithm.
- Using polyphase channelizer to effectively channelize the captured bandwidth.
- Power squelch is used to limit demodulation of noise, and Mike Ryan also uses it in his project.
- The HackRF one SDR is used in this project because that is the recourse provided by Axis Communication.
- To test the performance of the receiver, EMC and RF chambers are used to limit the outside RF interference and know exactly how many packets are being transmitted to be able to calculate the performance.
- The CMW500 machine is used to communicate with an Axis W101 camera; this is done to be able to set the modulation used in the communication, set packet type, and switch on and off frequency hopping. This will enhance the results as all the parameters for calculating the performance of the receiver are known.
- A single receiver is also used in this thesis as the CMW500 machine has the option to turn off frequency hopping and is used primarily to confirm that the receiver architecture works.

Packet finder code:

- Searching for the preamble and trailer pattern in the bit stream as these patterns are always the same in the access code.
- This code does not take bit error into consideration due to time constraints.

Collecting data for the automatic k-means algorithm:

- The median method is used to collect data as it is the same method Mike Ryan uses in his project.
- To use the median method, the first step of demodulation needs to be created using GNU radio. The demodulation technique used is to calculate the phase difference of the signal.
- Then a custom Python block is used to save samples from the demodulated data and use the median method. Then that data is saved for the automatic k-means algorithm.

Automatic k-means algorithm:

- Devices are placed in an RF and EMC chamber so that when the data is collected, the number of devices is known. This is necessary as the k-means algorithm is an unsupervised ML method, and to confirm the performance, the number of devices needs to be known.
- Elbow method is used to compare it to the Silhouette method to see if the automatic method works properly.

4.2 Preparations

Before modeling the receiver, the firmware for the HackRF one needs to be downloaded on the host computer, and that is done by following the instructions given in the installation guide [29]. Afterward, to program the HackRF one, the open-source program GNU radio is used.

4.3 Methodology

This chapter outlines the methodologies employed in this thesis to model and test the GFSK receivers using SDR technology. The section details the tools utilized, the design and implementation of the modeled GFSK receivers, the packet finder, the frequency demodulator, the median method, and the k-means clustering code. It also describes the testing setup and procedures conducted within an EMC chamber to ensure minimal interference and reliable performance evaluation.

4.3.1 Tools

The tools used in this thesis are shown in Table 1.

Tools
HackRF one (SDR)
RTL-SDR Blog V3 R860 Dipole Antenna Kit
Laptop with Ubuntu operating system
Samsung S21
Samsung S10
OnePlus Nord 3
iPhone 15 Pro
Bose quietcomfort 45 (Headphones)
GNU radio
Axis W101
R&S®CMW500 wideband radio communication tester
RF chamber

Table 1: Tools

4.3.2 Modeled GFSK receivers

The first step is to process the bandwidth signal captured from the HackRF block by connecting it to the Remove DC spike block; this is done to remove the frequency spike in the center of the bandwidth that is caused by the local oscillator in the SDR down-converting the frequency to baseband and causing a DC spike in the center frequency

[18]. Then, the baseband is processed by the polyphase channelizer, which splits the baseband into six channels and filters the channel signal for the first time. Figure 9 shows the structure in GNU radio.

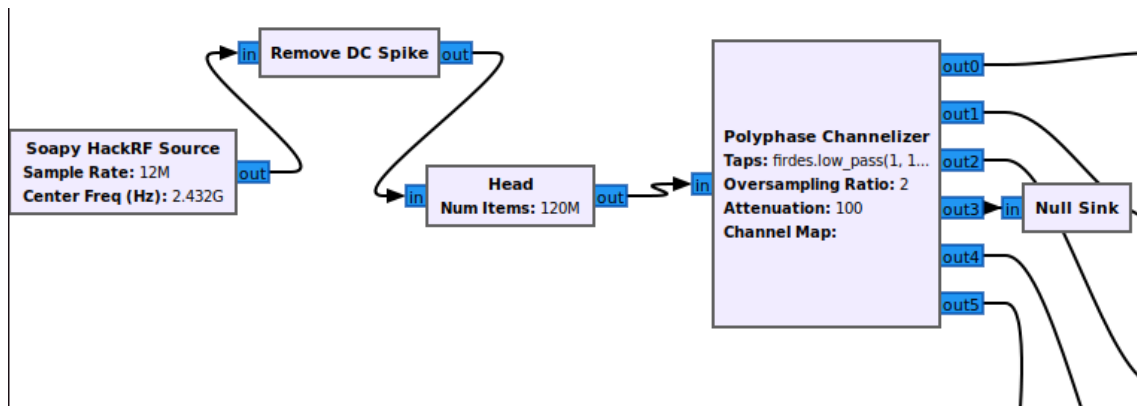


Figure 9: Block structure in GNU Radio

The HackRF sampling rate is 12 MHz, and the baseband is 6 MHz. In this block diagram, we channelize the baseband to 6 channels, each having a 1 MHz baseband, and down sample the signal equally and an oversampling ratio of two to receive a 4 MHz sample rate per channel. The center frequency can be set to anything in the ISM bandwidth where Bluetooth communicates. The polyphase channelizer uses an LP filter bank with a cutoff frequency of 500 KHz and a transition width of 100 KHz. The baseband center frequency is centered at zero Hz after channelization and filtering, so it will pass all low frequencies and limit the high frequencies. This is why a second BP filter is needed to tighten this channel and minimize interference with neighboring channels. A null sink is connected to output three due to how the polyphase channelizer is programmed in GNU radio, where a six-split configuration splits the baseband like Figure 10; this means that channel 3 wraps around from the end to the beginning of the baseband and makes it combine two different channels and thus contains useless data [30].

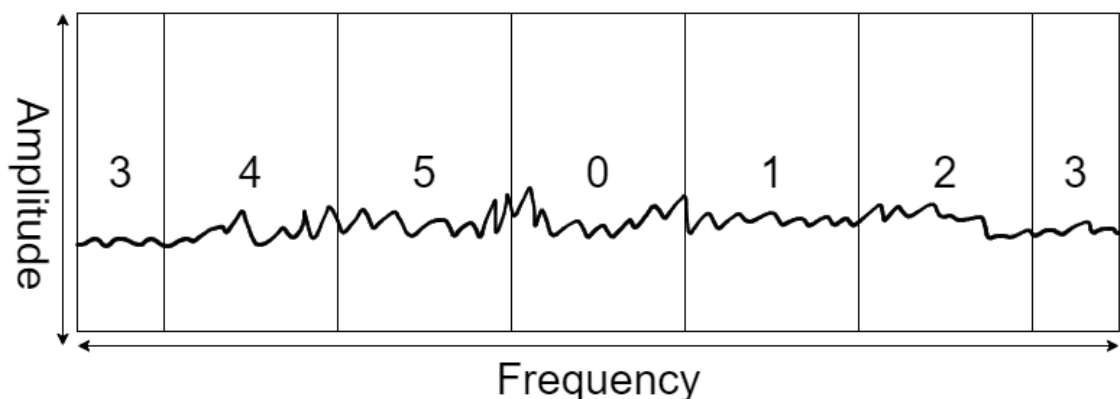


Figure 10: Polyphase channelization illustration

Figure 11 shows the settings for the LP filter bank used in the polyphase channelizer.

Taps

`firdes.low_pass(1, 12e6, 100e3, 500e3, window.WIN_HANN)`

Figure 11: LP filter bank parameters

Then, the Power Squelch block limits the incoming signal depending on the threshold, which is set manually. Setting the threshold around -35 dB will remove unwanted signals and only pass signals with RF activity. When the Power Squelch lets a signal through, it gets further filtered by a BP filter with -500 and 500 KHz cutoff frequencies and a transition bandwidth of 100 KHz. This creates a clean channel for the GFSK Demod block to demodulate. Settings changed in the GFSK Demod block is the Samples/Symbol block, which is set to 4 because the sampling frequency is 4 MHz and Bluetooth BR symbol rate is 1 Msymbol/s (1 Mb/s) and to calculate the sensitivity, the following formula is used:

$$k = 2\pi \frac{f\Delta}{f_s} \quad (14)$$

Where the $f\Delta$ is the frequency deviation of the Bluetooth BR and f_s is the sampling frequency [31]. This will give a sensitivity of 0.29, and the other settings are set to default. Figure 12 shows the diagram for this step in the process.

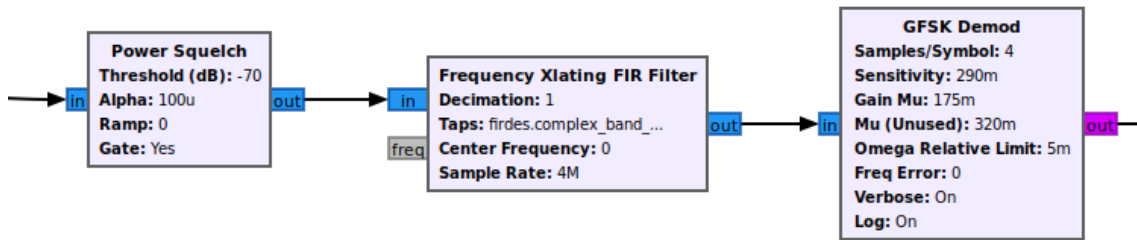


Figure 12: Filtering and demodulation step

The GFSK Demod block contains the M&M clock recovery block, based on the Mueller and Mueller (M&M) algorithm, designed for digital communication systems to align the receiver's timing with the transmitter's symbol timing. It is essential for accurately decoding symbols from a received signal with potential frequency and phase discrepancies. After that, the signal is processed by a binary slicer inside the GFSK Demod block and turns the demodulated signal into 0x00 or 0x01 bytes [32][33]. Lastly, the data stream is repackaged into correct representing byte configurations and saved to a file. Figure 13 shows the last step of the process.

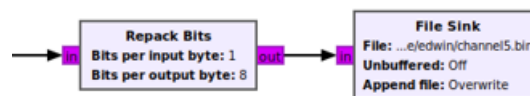


Figure 13: Repackage

The steps in Figures 12 and 13 are repeated for each channel.

A single channel receiver will also be used in this thesis for testing purposes, and the change is that the polyphase channelizer is replaced by a second BP filter with the same parameters, as shown in Figure 14. The rest of the receiver is unchanged.

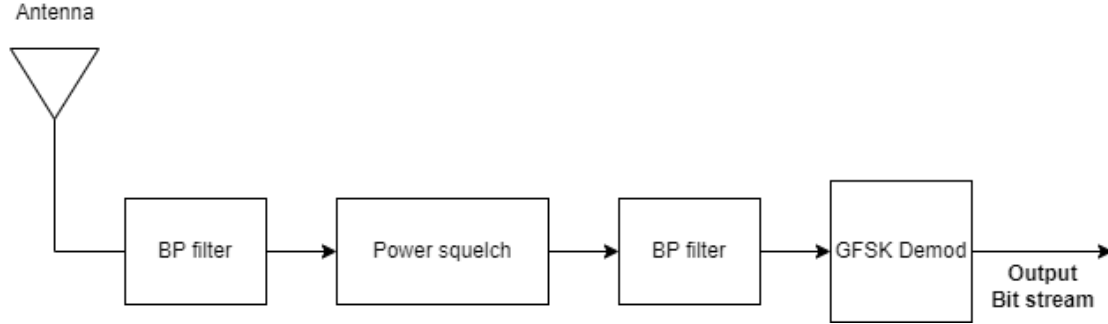


Figure 14: Single-channel receiver

4.3.3 Packet finder

When the data is saved to a file, a Python script will open the saved bin file, read the content, and convert the data into bits for further analysis. The goal of the packet finder is to look through the bit stream and search after access codes because they have a predetermined structure where the start of access code starts with either a 10101 or 01010 preamble, and after 66 bits, it ends with a 10101 or 01010 trailer.

The bits are saved in a string and later looped through to find the access code pattern. It starts by looking at the first-bit index and comparing the five sequential bits to the pattern 10101 and 01010; if the pattern does not match, the script goes to the following index and starts the process again. When a match is found, the program skips the following 66 bits and starts comparing the trailer with the patterns 10101 and 01010; if there are no matches, the script goes back to the preamble start index and moves the index by one bit to start over the process. If the trailer matches the pattern, it saves the access code and increments the total access code counter by one, then starts the process again by moving the index to the end of the trailer.

4.3.4 Frequency demodulator

To collect the CFO and frequency deviation, the first step of the frequency demodulation is created in gnu radio, as shown in Figure 15, where it takes the signal sample z_n and subtracts it from the previous signal sample z_{n-1} [34]. This is done using the following formula.

$$\Delta\theta = \arg(z_n) - \arg(z_{n-1}) \quad (15)$$

Where \arg denotes the angle between the positive real axis and the line joining the origin and z_n . Then the $\Delta\theta$ is the difference in the phase of the two samples.

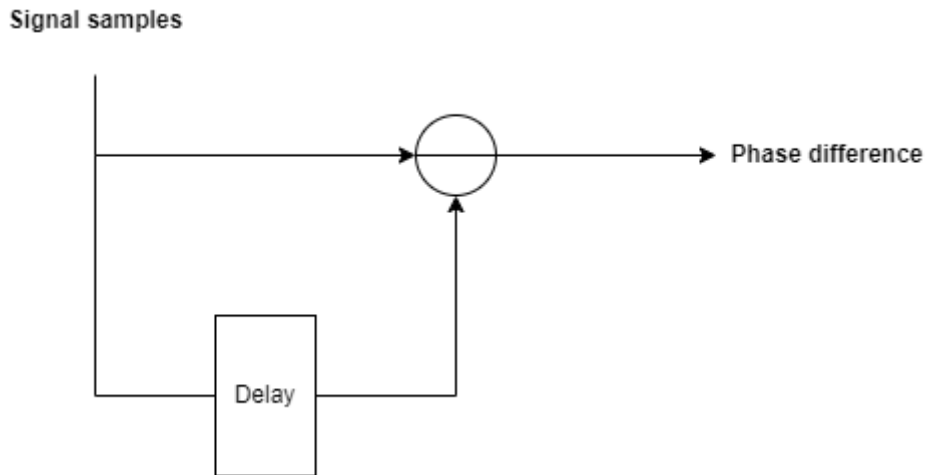


Figure 15: Phase demodulator

4.3.5 Median method

The median method is used to correct the CFO, so it is centered around zero because the transmitter and receiver have different center frequencies during communication. To correct this offset, all the positive samples are saved in one list and the negative in another, and then the median of each list is retrieved. To calculate the CFO, the median positive sample is subtracted from the median negative samples and then divided by two. After that, the positive median sample is subtracted by the calculated CFO and divided by two, resulting in the FSK deviation of the Bluetooth connection [24][35].

To implement these calculations in GNU radio, an embedded Python block is used that allows custom Python code to be implemented and used as a block in the GNU radio software. The first step is to initialize the variables used in the embedded Python block.

Then, the next step in the GNU radio block is to create a work function that calculates during run time. First, we need to reduce data noise before saving the data; the program only starts saving data when the `squelch_sob` tag appears in the signal stream from the power squelch tagging at the beginning of the burst. The power squelch also has a `squelch_eob` tag that represents the end of the burst, but it does not work correctly in GNU radio and tags the stream a sample after the `squelch_sob` tag; that is the reason why the code initializes a variable called `duration_sampels` that will be used to save samples 100 us after the `squelch_sob` tag to get around the end of burst tag issue. Then, the code further filters data, where it only saves data between -0.5 and 0.5 because the `squelch_sob` tag appears in samples before the demodulated signal and contains unnecessary noise data for the k-means algorithm. The reasoning behind these implementations is explained in the results section of this thesis.

Then, the code calculates the median method with the saved data points inside the positive and negative phase lists. The if statement guarantees enough samples are inside the lists to do meaningful computation and get a good representation of the

median phase for both the negative and the positive. Lastly, the data is saved into a CSV file that is used later for the k-mean algorithm.

4.3.6 K-means code

To implement the automatic k-means algorithm, we can use the library sklearn, which is a powerful ML library in Python. First, the code loads the captured data files and transforms them with the StandardScaler method; then, the code runs the k-mean algorithm with 1 cluster, as each data set represents one unique Bluetooth connection. When the cluster is formed, the code removes 20% of the further points from the cluster center to make the data cleaner so the outliers do not interfere with the result.

Then, the code has a method that calculates the WSS of the merged data set for clusters ranging between 1 and 10 and a method that later plots the WSS against the number of clusters for visual analysis.

Then, the preform_k_means method calculates the WSS, plots the elbow method, and performs the Silhouette method by performing k-means on the given number of k and then calculating the Silhouette score of each iteration. Then, it retrieves the optimal cluster score by taking the maximum Silhouette score, forms the last k-means clustering, and plots the output of the k-means algorithm.

4.3.7 Testing set-up

Axis Communications has provided the R&S®CMW500 wideband radio communication tester with an RF chamber where an antenna is placed inside the chamber and connected to the CMW500 tester. The chamber blocks all outside RF activity and is used by Axis to conduct Bluetooth testing on their body-worn W101 camera. Table 2 shows the essential settings for the single channel test used for the CMW500 tester to conduct a performance test on the Bluetooth receiver.

Modulation	GFSK
Symbol rate	Basic rate / 1mbs
Channel	RX 30
Packet Type	DH1 27 bytes, 00000000
Frequency hopping	OFF

Table 2: CMW500 single channel test settings

The frequency hopping setting is enabled to test the multi-channel receiver, as shown in Table 3.

Modulation	GFSK
Symbol rate	Basic rate / 1mbs
Packet Type	DH1 27 bytes, 00000000
Frequency hopping	ON

Table 3: CMW500 multi-channel test settings

Packet type DH1 is used to calculate the receiver's performance quickly because it occupies only a single slot [36]. To calculate the performance of the receiver, take the

number of captured Bluetooth packets per second and divide it by the number of frequency hops per second, which is 1600.

For the receiver test, the Axis W101 is placed inside the RF chamber and switched to test mode, allowing the CMW500 to connect to the camera and send predetermined data, set channel, change modulation, and change packet type. Then, the HackRF one is placed inside the RF chamber with the antenna, and the USB cable is connected to a USB port inside the chamber, which is connected to a USB cable outside so the computer can communicate with the HackRF one without having the chamber door open. Figure 16 shows the lab set-up.

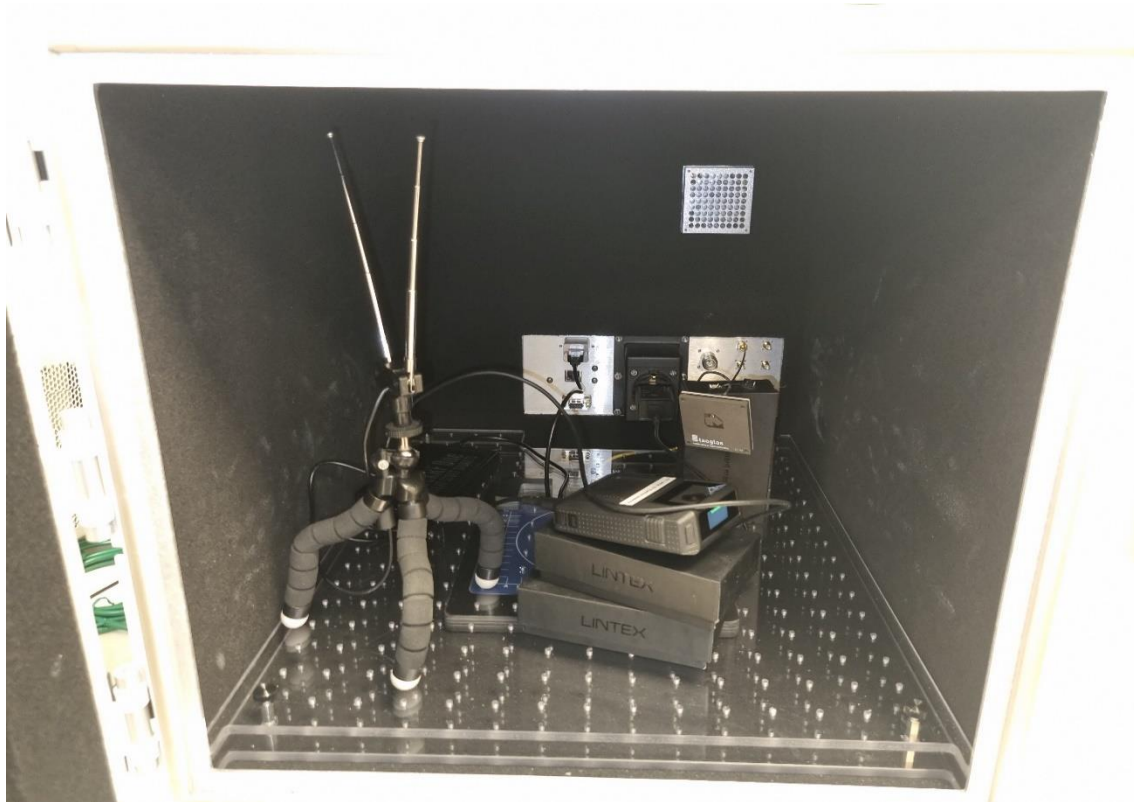


Figure 16: Lab set-up

Then, the single and multichannel receivers will capture packets for 10 seconds, configured by the head block shown in Figure 9 that counts the number of samples taken per second multiplied by the number of seconds wanted.

Table 4 shows the following settings used in HackRF for the experiment.

Single channel	Multi-channel
Bandwidth: 1 MHz	Bandwidth: 6 MHz
Center frequency: 2.432 GHz	Center frequency: 2.432 GHz
LNA: ON	LNA: ON
IF gain: 20 dB	IF gain: 20 dB
VGA: 20 dB	VGA: 20 dB
Sample rate: 4 MHz	Sample rate: 12 MHz

Table 4: HackRF one settings

4.3.8 EMC test set-up

The EMC chamber at Axis makes it possible to have a distance between the monitored device and the receiver; for this experiment, they were placed 3 meters apart. Each device's data was captured for 10 minutes, and they streamed an audio file to the headphones. Table 5 shows the devices in which the data was collected. Figure 17 shows how the experiment was conducted.

Devices
Samsung s21
Samsung s10 plus
Iphone 15 Pro
OnePlus Nord 3
Lenovo Legion 5 Pro 16

Table 5: EMC devices

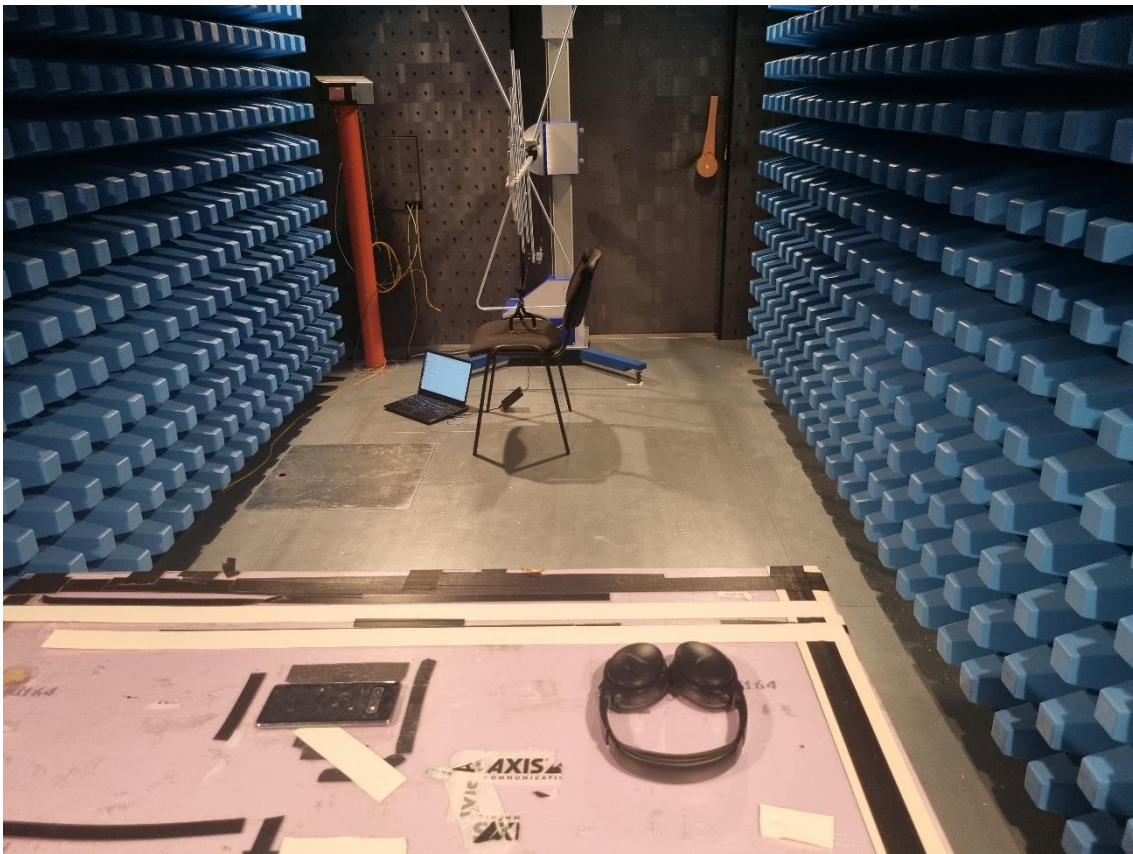


Figure 17: EMC set-up

5 Results

This section will present the results of the Bluetooth receiver performance, such as how many packets are detected per second, and present the data saved from the physical imperfections of the signal for the k-means algorithm. This section will also show the results of the performance of the k-means algorithm inside the EMC chamber for detecting the number of unique devices near the receiver.

5.1 Captured Bluetooth signal

To see if the receiver works appropriately is done by analyzing the FFT plot after the BP filter, as shown in Figure 18. In the first FFT plot, the signal level is around -80 dB, and when the receiver detects RF activity, the signal level rises to around -29.87 dB. This does not mean precisely that this is a Bluetooth signal because Wi-Fi uses the same ISM band as Bluetooth, but it indicates that a signal is captured.

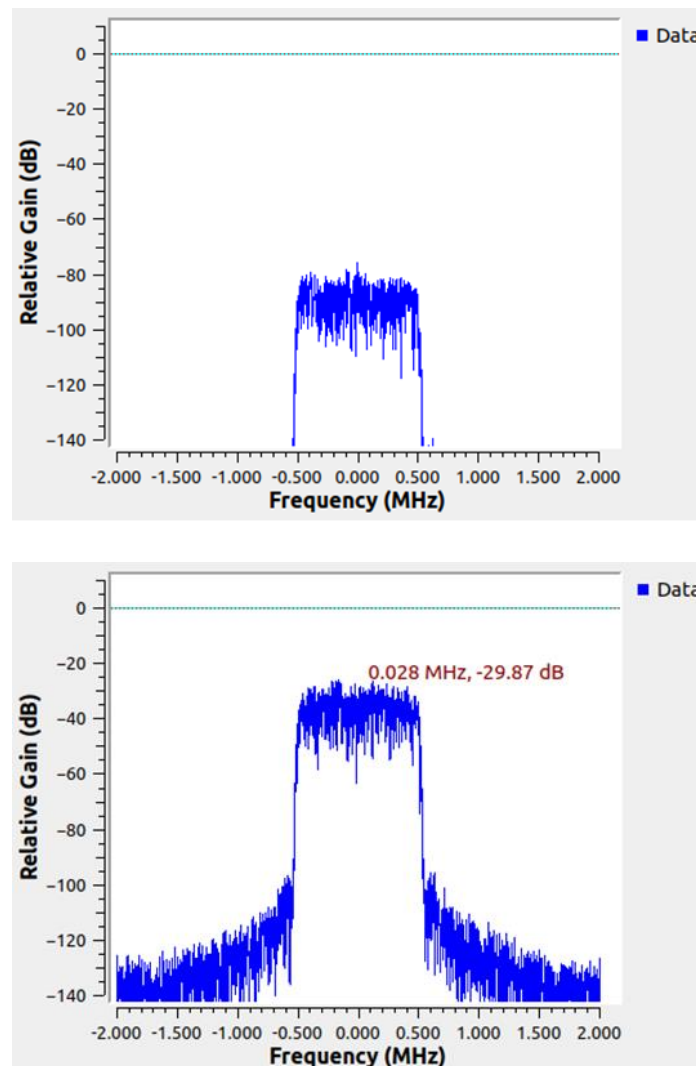


Figure 18: RF activity

By plotting the I/Q data, we can further analyze if the signal has the attributes needed for a Bluetooth signal. The first plot in Figure 19 shows a scatter plot in the shape of a dense cluster; this implies that the signal did not have any valid I/Q data. The second scatter plot shows a circle with a structured pattern, and compared to the center view of Figure 4, it shows that the captured signal has valid I/Q data. The last step to identify if the captured I/Q data is a Bluetooth signal is to use the access code finder script.

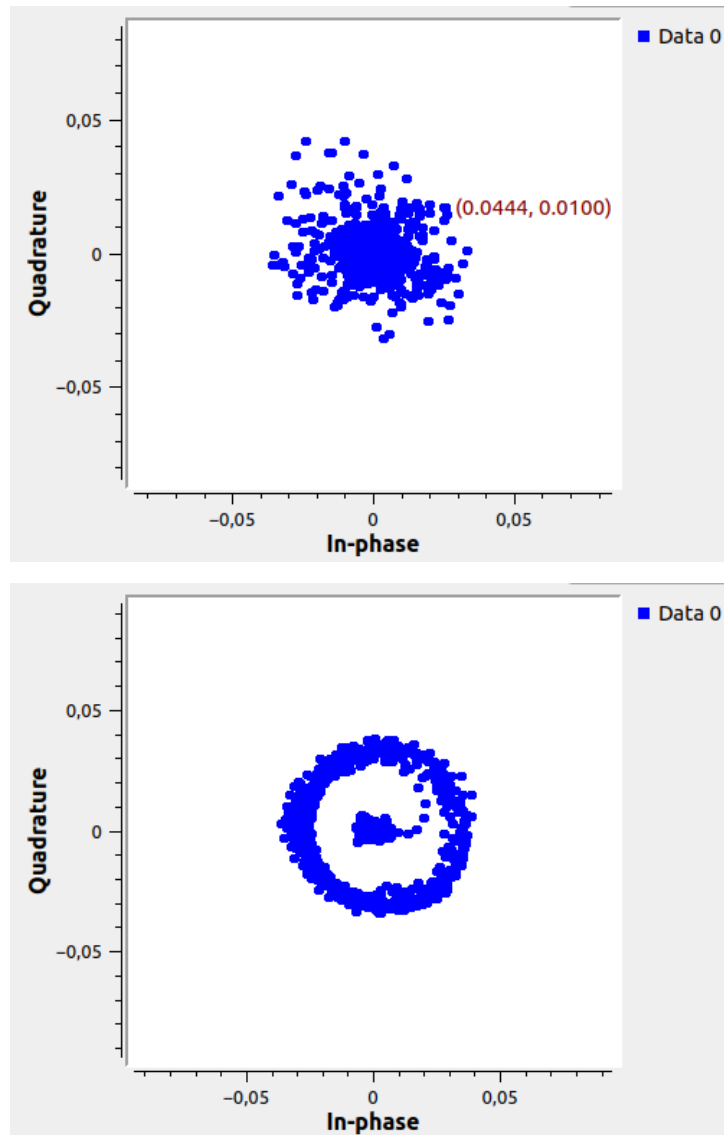


Figure 19: Captured I/Q data

5.2 Captured Bluetooth packets in RF chamber

This section presents the performance of the single and multi-channel receivers by analyzing the captured bit stream and calculating the performance of each receiver. This section also discusses why the receivers don't have perfect capture rates and raises potential factors that affects the results.

Each receiver captured packets for 10 seconds each, and the head block was configured to match the required sample limit to ensure the program was

terminated after 10 seconds. The first step to confirm that the receiver works as intended is to capture Bluetooth packets and then search for the predetermined DH1 packet type set on the CMW500 machine; in this test, the packet contained 27 bytes of zeros.

An important aspect is that the receiver uses the Muller algorithm to recover the clock and later determine what part of the signal is zeros and ones. This means that the captured packets are not always perfect and can contain some bit errors; this issue is also present in regular Bluetooth connections where the clocks are synchronized but are limited to only have a raw bit error of 0.1 % for it to follow the Bluetooth core specification [12].

Sometimes, the payload of a demodulated packet contains a byte 0x01, meaning that the receiver failed to determine one bit and contained the wrong information. This means the access code finder script will not have a 100 percent detection rate as it does not consider possible bit errors in the bit stream. An example of a bit error can be found in Appendix B, where the red square shows the start of the payload.

5.2.2 Single channel performance

For the single channel, the head block is set to contain 40 M samples because the sample rate is set to 4 MHz, and after the packet collection is finished, the Python script is executed on the saved bin folder. The total captured access codes were 11713 during the test, and to calculate the performance equation, 13 is used; this results in the single channel receiver having a performance of around 73%. This is due to the limitations mentioned before, where the clock recovery step can be misestimated, and the script does not consider the bit error. Even with these limitations, it has a 73% packet capture rate. As Bluetooth hops 1600 times per second, a detection will occur in less than a second, fast enough to start surveillance with security cameras.

5.2.3 Multi-channel performance

The head block was set to 120 M samples for the multi-channel test, as the sample rate was 12 MHz. This receiver also captured packets for 10 seconds, but the performance is calculated differently; when frequency hopping is enabled, the packets will hop between all the 79 channels in the ISM band, meaning every channel will have around 20 packets per second. The 5-channel receiver can capture a maximum of 100 packets per second. Table 6 shows the number of packets captured during the 10-second test.

Channel number	Captured packets
1	254
2	210
3	124
4	92
5	188

Table 6: Captured packets in the multi-channel receiver

The total number of packets captured is 868 by using 100 hops per second in equation 13, which means that this receiver has a capture rate of 86.8%; as the hopping sequence is determined by the master and slave when pairing up, some channels will receive more packets depending on the generated sequence.

The multi-channel receiver has the same limitations as the single-channel receiver, where the clock recovery algorithm makes mistakes, and some bits are misinterpreted. If this bit error is assumed to be random, each packet detection will be a separate statistical event, and thus, to calculate the number of trials to first success, we can use the following equation.

$$E = \frac{1}{p} \quad (16)$$

Where E is the expected value of the number of trials until success and p is the probability of the event occurring, which in this case is 0.868. This will result in the E being 1.152, so the program is expected to detect a Bluetooth device when the transmitter has sent out 1.152 Bluetooth packets. In this laboratory environment, the expected detection time is 720 us, because each packet is 625 us long.

5.3 Automatic k-means algorithm

This section presents the results from the automatic k-means algorithm. First, this section presents how the data is collected and what devices were used in the experiments. The results from the optimal k estimation methods are presented with graphs, and the performance is based on whether the methods can correctly estimate the number of devices near the SDR.

Gnu radio plots the signal during run time and to verify that the demodulator works. We can plot the demodulation, as shown in Figure 21. Here, we can see something that resembles zeros and ones. To better represent the zeros and ones, the signal must be processed by a clock recovery algorithm like the M&M clock recovery algorithm. As the data the k-means algorithm needs is the CFO and frequency deviation, the clock recovery step is not required for the gnu radio block diagram. In Figure 20, we can see the issues mentioned in section 4.3.5, where the tags are not working as intended, and some demodulated noise is present at the beginning of the Bluetooth packet. With the implementation of those precautions, the data recovered is as clean as possible for the k-means algorithm.

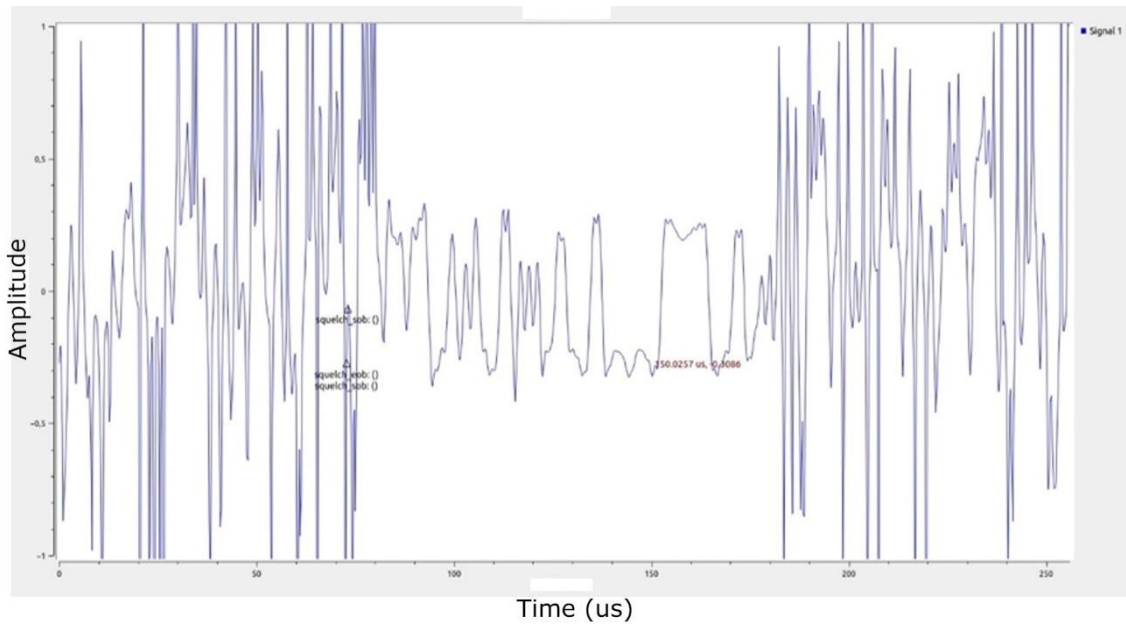


Figure 20: Demodulated signal

5.3.1 Elbow and Silhouette method

The same lab setup was used to collect the data for the k-means algorithm, but the data was collected for 10 minutes for each connection; the first connection was between the W101 camera and the CMW500 machine, and the second connection was between the Samsung s21 and the wireless bose headphones. In Figure 21, we can see the output of the Elbow method.

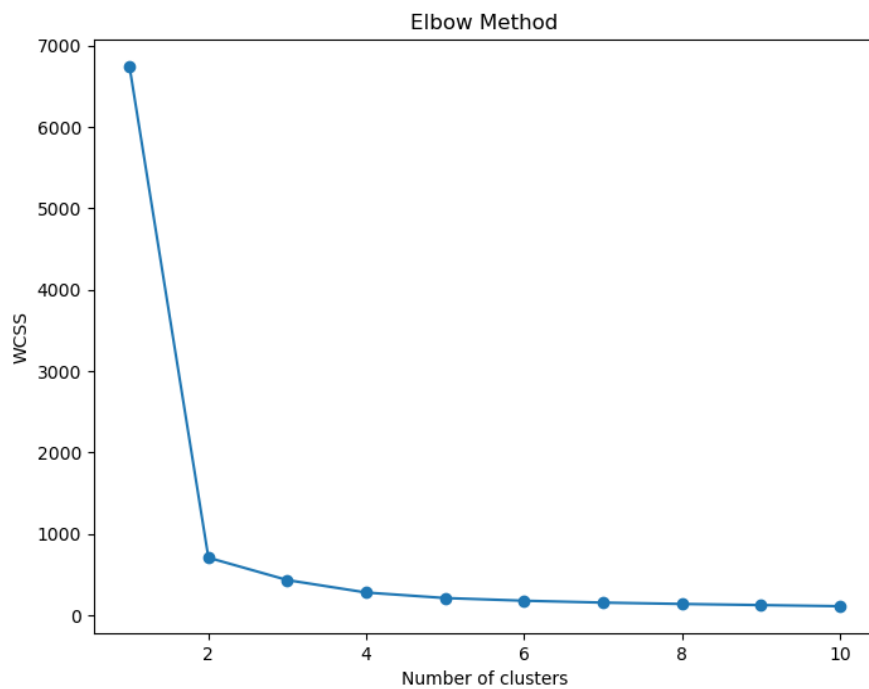


Figure 21: Elbow method results

In this plot, we can see where the elbow point is as the graph decreases drastically from 1 to 2, and later, it levels out the more clusters are added. This means that the optimal cluster number for this data set is two and correct, as in the experiment, only has 2 Bluetooth connections were captured. After the Elbow method, the code runs the Silhouette method. The highest Silhouette score was 0.84 at two clusters, the same as the elbow method, and the number of connections in the dataset is correctly determined. This means that the code will use 2 clusters for the k-means algorithm, and in Figure 22, we can see the results of how the algorithm clusters the data.

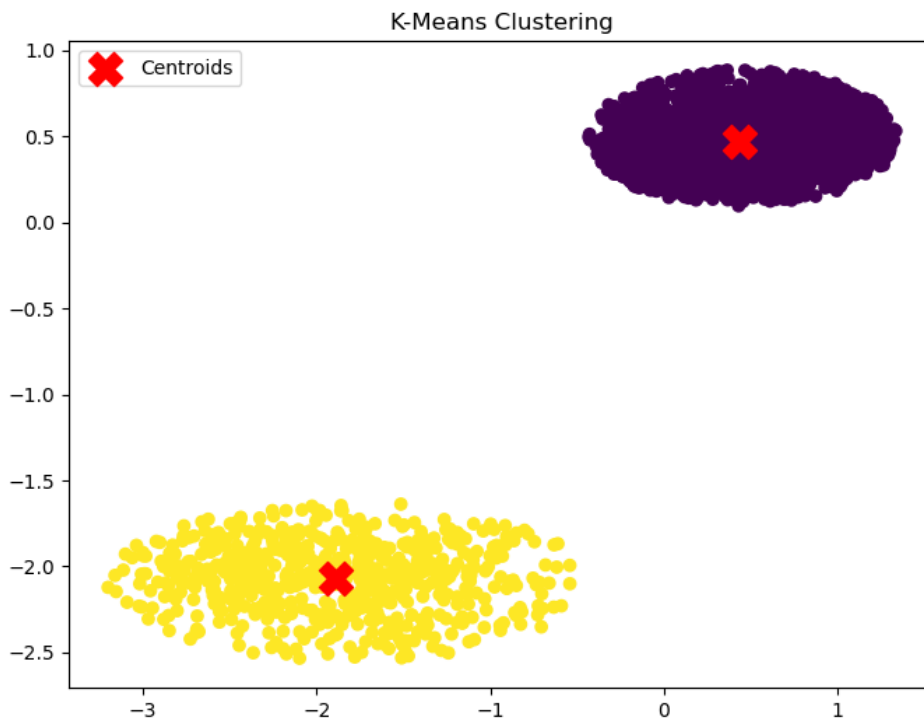


Figure 22: Optimal k in k-means algorithm

Here, we can see that the CFO and the frequency deviation can be used as a metric to distinguish whether a unique Bluetooth connection is present around the SDR. This graph also shows that the data cleaning that the code made before performing the k-means algorithm created optimal clusters.

5.3.2 EMC results

The data was processed the same way as in the experiment with the CMW500 machine, where the furthest 20% of data points were removed. The Elbow method predicted that three connections were present during this time, as shown in Figure 23.

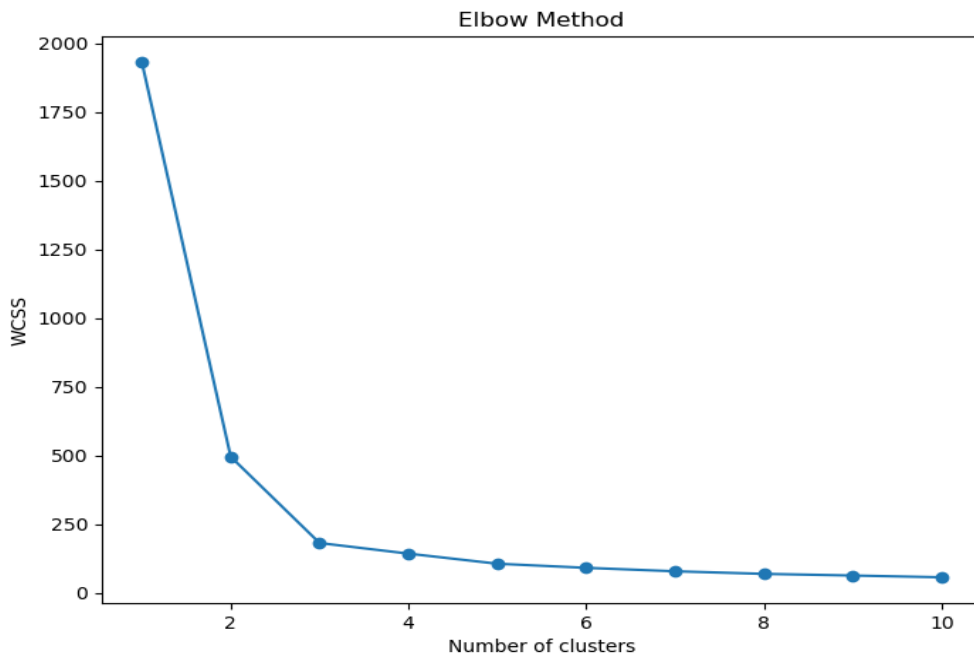


Figure 23: EMC Elbow method

This means that some data sets are too close to each other for the algorithm to distinguish between them. Figure 24 shows the clustering formation from the optimal number of K from the Silhouette score; in this graph, each data set has been labeled to see which data sets cluster too close to each other.

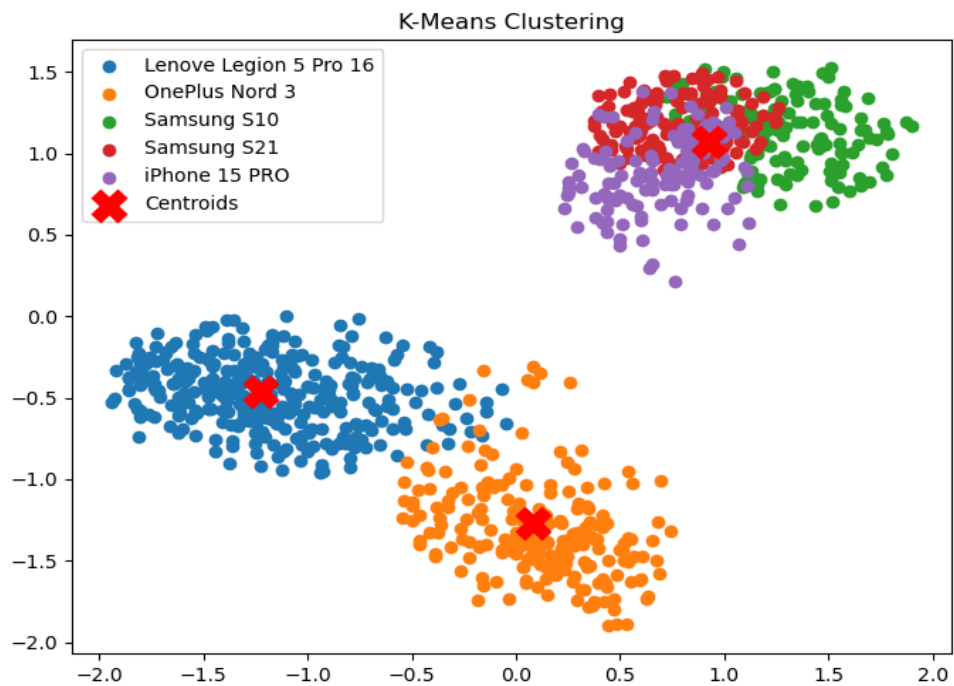


Figure 24: K-means performance in EMC

Here, we can see that even if the clusters are close to each other for the Samsung and iPhone phones, they still cluster slightly differently. However, these points are too close to each other to estimate the k correctly.

6 Discussion

This section discusses the results of the thesis. It compares it with similar projects, identifies what sections could be further analyzed, and discusses what improvements can be made to improve the project's overall performance. This section also presents the legal boundaries of the project and discuss how to handle personal data.

6.1 Receiver

The Bluetooth receiver achieved good results, where the single-channel receiver had a capture rate of 73%, and the multi-channel receiver had a capture rate of 86.8%. This works perfectly in an environment like an RF chamber where all outside interference is blocked, but the access code finder code searches after preambles and trailers, which are susceptible to noise, as Mike Ryan mentioned in his BLE sniffer project. The method he uses to determine if a Bluetooth packet is present is to search the bit stream valid candidates by finding the BLE packet structure and comparing it to the burst length. The candidate with the closest burst length to the recorded burst length is chosen as the correct Bluetooth packet [24]. To make a similar burst length comparison for this thesis implementation, we need to replace the power squelch block in GNU radio because it cannot correctly tag the beginning and end of the burst. This issue can be resolved by using out-of-the-tree modules in GNU radio and importing Liquid's AGC burst capture, which is the power squelch method Mike Ryan uses in his project [35].

Compared to Mike Ryan's project, this thesis uses the M&M clock recovery algorithm for better symbol recovery compared to using every other sample as a logical one or zero [24]. This yielded promising results, but not perfect, as there were still some bit errors in the captured bit stream, as mentioned in the result section. This can be due to the RF chamber not being perfect and still letting through some RF signals that interfere with the receiver; another reason is that the M&M algorithm was first proposed in 1976 which makes the clock recovery algorithm quite old [37]. The issue with GNU radios GFSK demod block is that the algorithm is embedded in the source code and cannot be changed, which means that to investigate other clock recovery algorithms the symbol sync block can be used with the combination with the quadrature demod block [38]. The symbol sync block can change algorithms simply by changing the settings in GNU radio, and an investigation into the best algorithm can be made to enhance the receiver's performance.

6.2 Data collection and k-means algorithm

Collecting clean data is essential for ML development because insufficient data always results in wrong outputs. In this regard, the code collecting the data implemented measures to reduce data containing noise by removing samples with too much phase difference, as seen in Figure 20. Then, further filtering out 20% of the outliers in each data set resulted in clean-formed clusters that the automatic k-means algorithm could successfully determine the number of connections in the RF chamber data set. However,

the biggest issue with this approach is again the power squelch block that tags the stream incorrectly, right now the code takes in 100 us of samples after the sod tag, and this approach can still save samples if the sample is set before it goes over the 0.5 threshold set in the code. By having a squelch that correctly sets the tags, many of the precautions in the code can be removed, like the 100 us saving duration. This will lead to cleaner data and a better data set for the algorithm.

Furthermore, the automatic k-means method used in this thesis showed promising results. Both the Elbow and Silhouette methods predicted the correct number of connections in the data set collected in the RF chamber. For the EMC test, the automatic K estimations predicted the wrong number of connections due to the difference in cluster formation being too close together. However, by analyzing the result where each data set has been labeled in Figure 24, we can see that even if the clusters for Samsung and iPhone are close, the data points are concentrated in different places. The methods used in this thesis do not consider the distance between clusters, but a recent method has been developed to calculate the distance between the clusters. This method is called the weighted barycenter method and can be used to determine the distance between clusters. To implement this method, the distance between two clusters must be calculated, which is achieved by computing the distances between their weighted centroids. The centroids are calculated by weighted average, where each data point's contribution is adjusted by its weight. By implementing a weighted barycenter in the elbow method, the results are better and enhance the cluster forming where clusters are close to each other [20]. The reason for not implementing this method in the thesis is that the paper was written in 2024, and there are yet no Python libraries supporting these calculations during the time of this thesis.

The reason why Samsung and iPhone clustered so close to each other might be that they used the same manufacturer for their transceivers. The company Ifixit publishes teardowns of mobile devices to help consumers repair their broken devices and list the components used in each device. The devices that had a detailed description were the Samsung s10 lineup and iPhone 14 Pro Max, where the Samsung s10 uses the Qualcomm SDR8150 RF Transceiver and the iPhone 14 Pro Max uses the Qualcomm SDR735 RF Transceiver[39][40]. This likely means that the devices used in the EMC test that clustered together had transceivers made by Qualcomm. They still had slightly different cluster formations because they used different models from the same company.

This thesis has only worked with a small number of devices. A more extensive study needs to be conducted to determine if the automatic k-means algorithm is effective on a bigger sample size or if the number of increasing devices creates too many clusters close to each other. This concern was already spotted in Figure 24 for Samsung and iPhone.

6.3 Legal boundaries of wireless sniffing

The legal boundaries are an essential aspect for Axis Communications, which will further develop this thesis: what information can be listened to, and are there any

restrictions regarding monitoring the electromagnetic spectrum? The EU law will be examined for this part because this thesis was made in Sweden.

The General Data Protection Regulation (GDPR) Act applies as this project analyzes and collects Bluetooth data. Personal data is categorized as any data that can directly or indirectly identify a person, such as names, emails, biometric data, and more [41]. For Bluetooth, especially BLE devices that monitor heart rate and other biometric data and send out that information in the Bluetooth packets. The sniffer picks up this signal and demodulates it, and then the company is in the position of personal data, which needs to be handled within the guidelines set by GDPR.

These are the seven data protection principles [41].

1. **Lawfulness, fairness and transparency** — Processing must be lawful, fair, and transparent to the data subject.
2. **Purpose limitation** — You must process data for the legitimate purposes specified explicitly to the data subject when you collected it.
3. **Data minimization** — You should collect and process only as much data as absolutely necessary for the purposes specified.
4. **Accuracy** — You must keep personal data accurate and up to date.
5. **Storage limitation** — You may only store personally identifying data for as long as necessary for the specified purpose.
6. **Integrity and confidentiality** — Processing must be done in such a way as to ensure appropriate security, integrity, and confidentiality (e.g. by using encryption).
7. **Accountability** — The data controller is responsible for being able to demonstrate GDPR compliance with all of these principles.

This means that if the sniffer has a data packet that contains personal data, it needs to process the data to identify if a person is present. However, as soon as that processing is finished, the data needs to be removed, as stated in principle 5. Suppose this thesis becomes a stand-alone product or a peripheral to a surveillance camera. In that case, Axis Communications needs to investigate how to properly handle the data collected from the sniffer and ensure it follows all the guidelines set by the GDPR Act. As this work has already been done for their other surveillance products, there should already be an internal process to ensure that personal data is handled correctly.

7 Conclusion

This thesis developed a Bluetooth receiver with multiple channels that can capture Bluetooth packets without creating a connection between the devices, also known as a Bluetooth sniffer. This thesis's first and second goals were to make a receiver that could capture a Bluetooth packet with multiple channels, demodulate it, and confirm that the signal was a Bluetooth packet. This was achieved as the multi-channel receiver had an identification rate of 86.8%, which gave an expected detection time of 720 us.

The third goal of the thesis was also met, where the receiver did the first demodulation step and calculated the signal's phase difference. From that data, a Python script collected the positive and negative samples of the signal, performed the median method on the samples, and calculated the CFO and frequency deviation. Then, that data was used in an automatic k-means algorithm.

Furthermore, an investigation was conducted to analyze whether different devices had physical imperfections in the transmitted signal that could be identified as a unique connection. The observation that was made by other works suggested that these physical imperfection data clustered together and corresponded to a unique connection. Due to this observation, the thesis investigated if an automatic k-means algorithm could estimate the number of unique devices in the area from these clusters. Data was collected from multiple devices, showing that these physical imperfections clustered together. The first data set was collected in the RF chamber with the CMW500 machine, which showed that the data clustered well and could be correctly estimated using the Elbow and Silhouette methods. The second test in the EMC chamber with more devices showed that the Elbow and Silhouette method could not predict the correct number of connections due to several clusters being too close to each other. This observation showed that unique connections can be distinguished by their physical imperfections. However, the automatic k-means algorithm must implement calculations with weighted barycenters to estimate K effectively in close cluster scenarios.

In future work, the issues mentioned in the discussion should be corrected to optimize the receiver to retrieve better data and identify more packets in the captured signal. Furthermore, the code used in this thesis does not do real-time analysis, as the project now takes in all the data and processes it separately with the scripts developed. This can be achieved by programming embedded Python blocks in GNU radio or programming the SDR directly; for product development, the SDR should be programmed directly as GNU radio is used for prototype development. For this thesis to become a product, Axis needs to develop its own SDR platform and integrate it with its security camera to communicate properly and work with its eco-system.

Lastly, create a reliable testing environment for the automatic k-means algorithm, where noise is added and still know the number of Bluetooth devices that the SDR is sniffing. This is not possible in an Office environment as devices are hidden and can transmit. This can be done in an EMC chamber, where a Wi-Fi router and other

noise sources are placed, but for Axis EMC chamber, there are no ports to connect additional noise sources without having the door open.

This thesis only worked with Bluetooth, and to expand on this concept, other wireless protocols can be studied, such as Wi-Fi, 4G, and 5G, to cover a broader range of wireless protocols and make the SDR able to detect people even if they do not have an active Bluetooth connection active.

Reference list

- [1] Bluetooth SIG, “Bluetooth Technology Overview.” Accessed: Apr. 24, 2024. [Online]. Available: <https://www.bluetooth.com/learn-about-bluetooth/tech-overview/>
- [2] Bluetooth SIG, “Bluetooth® Mesh.” Accessed: Apr. 24, 2024. [Online]. Available: <https://www.bluetooth.com/learn-about-bluetooth/feature-enhancements/mesh/>
- [3] M. Ahmad Sofi, “Bluetooth Protocol in Internet of Things (IoT), Security Challenges and a Comparison with Wi-Fi Protocol: A Review.” [Online]. Available: www.ijert.org
- [4] J. Haartsen HNV, “Bluetooth-the universal radio interface for ad hoc, wireless connectivity.” [Online]. Available: <https://www.researchgate.net/publication/290790635>
- [5] B. Watson, “The Communications Edge™ Tech-note.” [Online]. Available: www.wj.com
- [6] Jason Marcel, “How Bluetooth Technology Creates Reliability From Unreliable Foundations.” Accessed: Apr. 24, 2024. [Online]. Available: <https://www.bluetooth.com/blog/how-bluetooth-technology-creates-reliability-from-unreliable-foundations/>
- [7] D.-C. Chang and T.-H. Shiu, “Digital GFSK Carrier Synchronization.”
- [8] Mikael Q Kuisma, “I/Q Data for Dummies.” Accessed: Apr. 24, 2024. [Online]. Available: <http://whiteboard.ping.se/SDR/IQ>
- [9] M. Silva Pereira, J. Caldinhas Vaz, C. Azeredo Leme, J. T. De Sousa, and J. Costa Freire, “A 170 μ a All-Digital GFSK Demodulator with Rejection of Low SNR Packets for Bluetooth-LE,” *IEEE Microwave and Wireless Components Letters*, vol. 26, no. 6, pp. 452–454, Jun. 2016, doi: 10.1109/LMWC.2016.2562639.
- [10] S. Byun, C. H. Park, Y. Song, S. Wang, C. S. G. Conroy, and B. Kim, “A Low-Power CMOS Bluetooth RF Transceiver with a Digital Offset Canceling DLL-Based GFSK Demodulator,” *IEEE J Solid-State Circuits*, vol. 38, no. 10, pp. 1609–1618, Oct. 2003, doi: 10.1109/JSSC.2003.817265.
- [11] T. C. Lee and C. C. Chen, “A Mixed-Signal GFSK Demodulator for Bluetooth,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 53, no. 3, pp. 197–201, Mar. 2006, doi: 10.1109/TCSII.2005.858320.

- [12] Bluetooth SIG, “Bluetooth Core Specification Bluetooth ® Specification,” 1999. [Online]. Available: <https://www.bluetooth.com/specifications/adopted-specifications>
- [13] R. Akeela and B. Dezfouli, “Software-defined Radios: Architecture, state-of-the-art, and challenges,” *Comput Commun*, vol. 128, pp. 106–125, 2018, doi: <https://doi.org/10.1016/j.comcom.2018.07.012>.
- [14] J. Moskal, “Interfacing a Reasoner with Heterogeneous Self-controlling Software,” 2011.
- [15] Y. Jang, G. Kim, B. Park, and H. Lim, “Generalized Polyphase Digital Channelizer,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 68, no. 10, pp. 3366–3370, Oct. 2021, doi: [10.1109/TCSII.2021.3069887](https://doi.org/10.1109/TCSII.2021.3069887).
- [16] Joseph D. Gaeddert, “Automatic Gain Control (agc).” Accessed: Apr. 24, 2024. [Online]. Available: <https://liquidsdr.org/doc/agc/>
- [17] “GREAT SCOTT GADGETS HackRF One,” 2023. Accessed: Apr. 24, 2024. [Online]. Available: <https://greatscottgadgets.com/hackrf/one/>
- [18] Marc Lichtman, “IQ Sampling.” Accessed: Apr. 24, 2024. [Online]. Available: <https://pysdr.org/content/sampling.html#dc-spike-and-offset-tuning>.
- [19] R. Lyons, “Understanding Digital Signal Processing’s Frequency Domain,” *RF Design Magazine*, May 2001.
- [20] J. Sara and B. Youssef, “A NEW METHOD FOR GETTING THE OPTIMAL NUMBER OF CLUSTERS BY K-MEANS USING THE WEIGHTED BARYCENTER,” *J Theor Appl Inf Technol*, vol. 15, no. 1, 2024, [Online]. Available: www.jatit.org
- [21] T. M. Kodinariya and P. R. Makwana, “Review on determining number of Cluster in K-Means Clustering,” *International Journal of Advance Research in Computer Science and Management Studies*, vol. 1, no. 6, 2013, [Online]. Available: www.ijarcsms.com
- [22] D. Starovoytova, E. Ataro, D. S. Madara, and S. Sitati, “Design and Testing of Mobile-Phone-Detectors,” vol. 7, no. 9, 2016, [Online]. Available: www.iiste.org
- [23] H. Givehchian *et al.*, “Evaluating Physical-Layer BLE Location Tracking Attacks on Mobile Devices,” in *2022 IEEE Symposium on Security and Privacy (SP)*, 2022, pp. 1690–1704. doi: [10.1109/SP46214.2022.9833758](https://doi.org/10.1109/SP46214.2022.9833758).

- [24] Mike Ryan, “Building a Modern Bluetooth Sniffer for SDRs,” DEF CON 30 RF Village . Accessed: Apr. 24, 2024. [Online]. Available: <https://www.youtube.com/watch?v=lpM9rnMfy2w>
- [25] M. Aftowicz, I. Kabin, Z. Dyka, and P. Langendoerfer, “Advantages of unsupervised learning analysis methods in single-trace SCA attacks,” *Microprocess Microsyst*, vol. 105, p. 104994, 2024, doi: <https://doi.org/10.1016/j.micpro.2023.104994>.
- [26] W. Mansouri, K. Ben Ali, F. Zarai, and M. S. Obaidat, “Chapter 27 - Radio resource management for heterogeneous wireless networks: Schemes and simulation analysis,” in *Modeling and Simulation of Computer Networks and Systems*, M. S. Obaidat, P. Nicopolitidis, and F. Zarai, Eds., Boston: Morgan Kaufmann, 2015, pp. 767–792. doi: <https://doi.org/10.1016/B978-0-12-800887-4.00027-4>.
- [27] A. Eadie, “The Anechoic Chamber Guide For EMC and RF (Wireless) Testing.” Accessed: Apr. 24, 2024. [Online]. Available: <https://www.etsolution-asia.com/blog/what-is-an-emc-or-anechoic-chamber-and-how-is-working>
- [28] “ABOUT GNU RADIO.” Accessed: Apr. 24, 2024. [Online]. Available: <https://www.gnuradio.org/about/>
- [29] “Installing HackRF Software.” Accessed: Apr. 24, 2024. [Online]. Available: https://hackrf.readthedocs.io/en/latest/installing_hackrf_software.html
- [30] “Channel Map.” Accessed: Apr. 24, 2024. [Online]. Available: https://wiki.gnuradio.org/index.php?title=Channel_Map
- [31] “Frequency Mod.” Accessed: Apr. 24, 2024. [Online]. Available: https://wiki.gnuradio.org/index.php/Frequency_Mod
- [32] “Clock Recovery MM.” Accessed: Apr. 24, 2024. [Online]. Available: https://wiki.gnuradio.org/index.php/Clock_Recovery_MM
- [33] RedhawkSDR, “integration-gnuhawk.” Accessed: Apr. 24, 2024. [Online]. Available: <https://github.com/RedhawkSDR/integration-gnuhawk/blob/master/gnuradio/gr-digital/python/gfsk.py>
- [34] Gary Schafer, “Gnu Radio & Frequency Demodulation - One More Time.” Accessed: Apr. 24, 2024. [Online]. Available: <http://www.site2241.net/may2020.htm>
- [35] Mike Ryan, “ice9-bluetooth-sniffer,” 2023, Accessed: Apr. 24, 2024. [Online]. Available: <https://github.com/mikeryan/ice9-bluetooth-sniffer>

- [36] Tom Nakase, “Basics of Asynchronous Connection-Less (ACL) Bluetooth Communication Protocol.” Accessed: Apr. 24, 2024. [Online]. Available: <https://www.silextechnology.com/unwired/basics-of-asynchronous-connection-less-acl-bluetooth-communication-protocol>
- [37] K. Mueller and M. Muller, “Timing Recovery in Digital Synchronous Data Receivers,” *IEEE Transactions on Communications*, vol. 24, no. 5, pp. 516–531, 1976, doi: 10.1109/TCOM.1976.1093326.
- [38] “Symbol Sync.” Accessed: Apr. 24, 2024. [Online]. Available: https://wiki.gnuradio.org/index.php/Symbol_Sync
- [39] “iPhone 14 Pro Max Chip ID”, Accessed: May 04, 2024. [Online]. Available: <https://www.ifixit.com/Guide/iPhone+14+Pro+Max+Chip+ID/153224#s318461>
- [40] Adam O’Camb, “Samsung Galaxy S10 and S10e Teardown.” Accessed: May 04, 2024. [Online]. Available: <https://www.ifixit.com/Teardown/Samsung+Galaxy+S10+and+S10e+Teardown/120331>
- [41] Ben Woford, “ What is GDPR, the EU’s new data protection law?” Accessed: Apr. 24, 2024. [Online]. Available: <https://gdpr.eu/what-is-gdpr/?cn-reloaded=1>

Appendices

Appendix A Github

<https://github.com/uncrazy12/bluetoothsniffer>

Appendix B DH1 payload with bit error

```
13 c3 32 ad 68 24 58 3d 00 1c 3f 00 00 00 bc 01
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 98 11 f9 d9 3f 9d
```