System Design Document

For

Bluetooth Protocol Analytical Research

Gianna Scarangelli
Matthew Irvin
Carolina Terre
Jonah Rowell
Connal Grace

Version/Author	Date
v1.0/Gianna Scarangelli, Matthew Irvin Caroline Terre, Jonah Rowell, Connal Grace	09/29/2023
v2.0/Gianna Scarangelli, Matthew Irvin Caroline Terre, Jonah Rowell, Connal Grace	10/31/2023
v3.0/Gianna Scarangelli, Matthew Irvin Caroline Terre, Jonah Rowell, Connal Grace	11/21/2023
v4.0/Gianna Scarangelli, Matthew Irvin Caroline Terre, Jonah Rowell, Connal Grace	02/05/2024
v5.0/Gianna Scarangelli, Matthew Irvin Caroline Terre, Jonah Rowell, Connal Grace	03/03/2024
v6.0/Gianna Scarangelli, Jonah Rowell, Caroline Terre	04/14/2024

TABLE OF CONTENT

SYSTEM DESIGN DOCUMENT

1 INTRODUCTION

1.1 Purpose and Scope

This document details the architectural and design elements for the project focused on investigating vulnerabilities in Bluetooth 5, Bluetooth Low Energy (BLE), and Zigbee protocols within a server-client configuration using Raspberry Pis or similar hardware. The purpose of this document is to provide a clear overview of the system design for investigating vulnerabilities in wireless IoT technologies utilizing a server-client setup.

1.2 Project Executive Summary

This section provides an overview of the wireless IoT vulnerabilities investigation project from a management perspective, showcasing the framework within which the system design was conceived.

1.2.1 System Overview

The system aims to explore, identify, and document vulnerabilities inherent to wireless IoT technologies, focusing primarily on the Bluetooth 5, BLE, and Zigbee protocols. Through a server-client configuration, orchestrated using Raspberry Pis (or similar hardware), the project delves into the security implications tied to the use of advertising IDs in wireless communication. A high-level system overview can be seen in Figure 1, showing how the hardware devices communicate with each other and which wireless communication protocol is being used.

Figure 1 is a simplified system architecture diagram, breaking down the system into discernable subsystems. Each box represents a hardware device, and each arrow represents the specified form of wireless communication. The server subsystem is hosted on a Raspberry Pi (or similar hardware), and will house the central server facilitating communication with client devices. The client subsystem, comprising various devices like Raspberry Pis, Zigbee Dev Boards, and ESP32s, will mimic IoT devices connecting to the server via Bluetooth 5 or Zigbee protocols. The analysis subsystem, utilizing Software Radios (SDRs) like HackRF, Flipper Zero, and laptops/servers, will conduct vulnerability analysis, capturing and analyzing the wireless traffic.

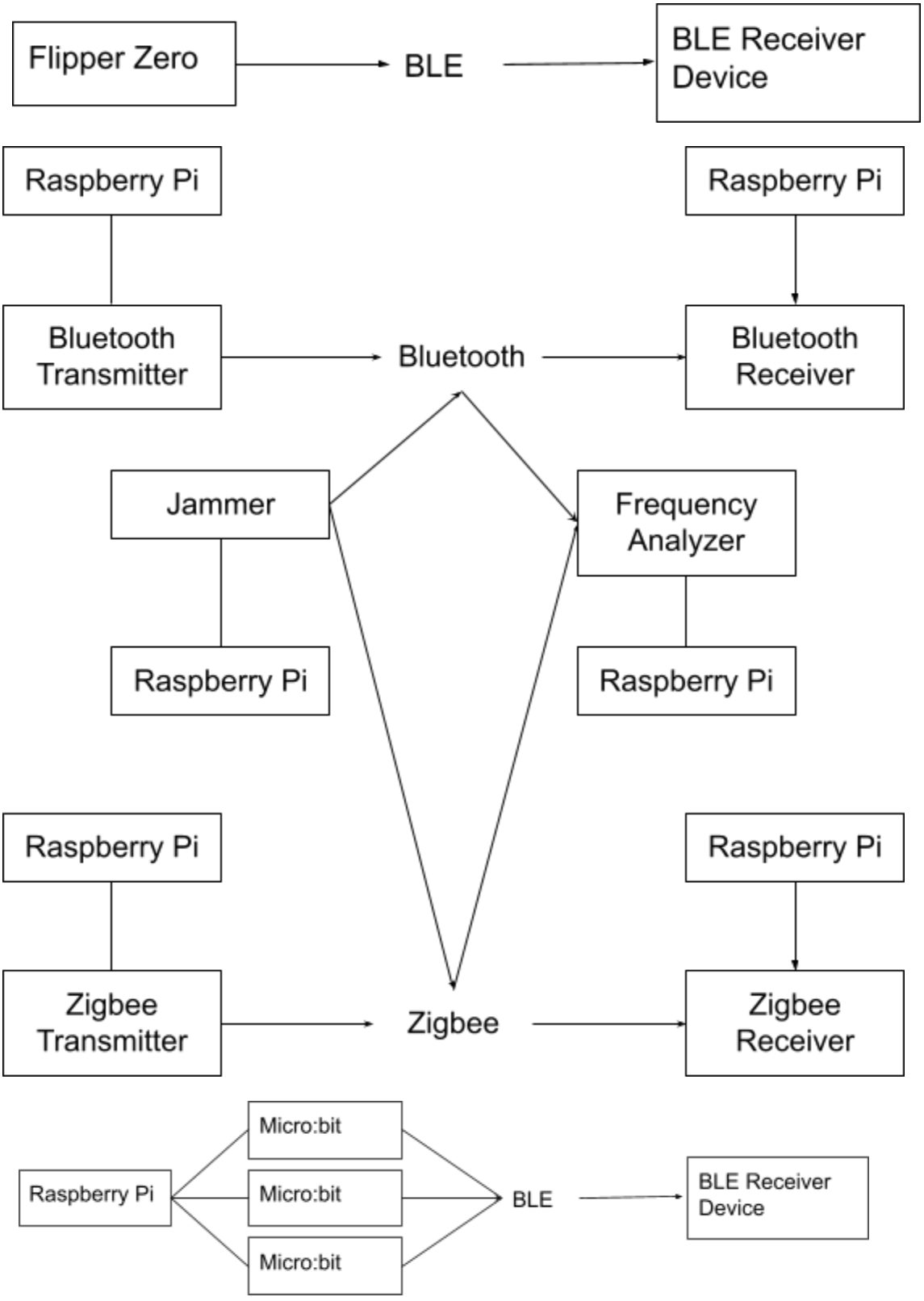

Figure 1: High-Level System Architecture Diagram

1.2.1.1 Reference to Requirements Traceability Matrix (RTM)

A thorough reference to the Requirements Traceability Matrix (RTM) in the Functional Requirements Document (FRD) is provided to map the allocation of functional requirements into this design document, ensuring a systematic transition from requirements to design.

1.2.2 Design Constraints

Investigating vulnerabilities within wireless IoT technologies, notably within Bluetooth 5 and Zigbee protocols, is restrained by several notable constraints: legality, budget, technology, time, and IT department approvals. The budget constraint of $25K necessitates prudent fiscal management for procuring vital equipment such as Software Defined Radios (SDRs), Zigbee Dev Boards, and Raspberry Pis. The technological boundary is drawn around Bluetooth 5 and Zigbee protocols, with a further obstacle of limited access to existing information on Bluetooth vulnerabilities, requiring a more thorough and potentially creative investigative approach. The time constraint is across a division of two semesters, enforcing a strict timeline for each critical phase: research, planning, and testing/findings. Additionally, the requisite approvals from the IT department add a procedural hurdle, as not all desired resources or methods may receive approval, possibly leading to adjustments or deviations in the planned approach. Another task is compliance with the standards of testing Bluetooth security legally. To test Bluetooth security, the project must adhere to Bluetooth SIG (Special Interest Group) specifications, while also not interfering with other signals. Each phase hence demands a meticulous orchestration of time, technological resources, budget, and adherence to organizational protocols to ensure a thorough investigation and insightful documentation of findings. The outlined constraints underscore the imperative for a meticulously structured, well-coordinated, and flexible approach to surmount these challenges, ensuring the project's objectives are duly met within the delineated framework.

1.2.3 Future Contingencies

This project, aimed at investigating vulnerabilities in wireless IoT technologies using a server-client configuration with Raspberry Pis and exploring vulnerabilities through Zigbee or Bluetooth 5 advertising IDs, may face several contingencies. First, equipment procurement delays might arise due to budget approval or supply chain disruptions, mitigated by initiating the procurement process early and identifying alternative suppliers. Second, the IT department might not approve some tools or methodologies, potentially hindering progress. Engaging with the IT department from the project's inception and preparing alternative methodologies can mitigate this risk. Third, the opaque nature of information regarding Bluetooth vulnerabilities might lead to undiscovered vulnerabilities; engaging with external experts or reviewing recent publications could provide deeper insights. Fourth, technical limitations with Raspberry Pis or other chosen hardware might inhibit the full exploration of vulnerabilities; having backup hardware options or considering upgrades can alleviate this issue. Fifth, project expenses may exceed the allocated budget due to unforeseen costs, which can be mitigated by meticulous budget tracking, allocating a contingency fund, and prioritizing essential expenditures. Lastly, project phases might extend beyond stipulated timelines due to unforeseen challenges; implementing a well-structured project management approach, utilizing monitoring tools, and allocating time buffers in the schedule can help in navigating through these challenges.

1.3 Document Organization

The document is arranged to analyze the system's framework in a fashion to help understand the system's design for a clear and detailed understanding. An introduction and executive summary of the project are presented at the start, and then sections on the high-level system architecture, design limitations, contingencies, and the precise hardware and software design follow. It also covers system interactions, user interfaces, and security issues, particularly as they relate to Bluetooth and Zigbee technologies. The document also discusses the testing phases, suggested equipment purchases, a breakdown of tasks by semester, and a glossary to clarify technical terms at the end, ensuring a thorough understanding of the project's design framework and the justification for various design decisions.

1.4 Project References

There are not any project references at this time.

1.5 Glossary

Flipper Zero - A tool to read, store, and emulate remote signals
Kali Linux - An open-source operating system aimed at Penetration Testing and Security Auditing.
Bluetooth - Short-range wireless data transmission standard
Zigbee - Wireless personal area network standard
RF - Radio Frequency
IoT - Internet of things
SDR - Software defined radio
USB - Universal serial bus
BBC - British Broadcasting Company

2 SYSTEM ARCHITECTURE

This section describes and gives an overview of the systems and subsystems used in this project.

2.1 System Hardware Architecture

This section describes the overall hardware of the system, a list of hardware devices used, and connectivity diagrams.

- **Zigbee Development Boards (IOTZTB-DK0006) -** Development boards for testing Zigbee signals and devices
- **HackRF One -** Software-defined radio used for transmitting signals
- **ESP32 2.4GHz Dual-Core -** Microcontroller with onboard Wi-Fi and Bluetooth
- **Raspberry Pi 4 8GB RAM -** Microcomputer used to transmit and receive Bluetooth signals as well as to control various tasks within the system
- **SD Cards** - Small storage devices for Raspberry Pi
- **SAMSUNG SSD T7 Portable External Solid State Drive 1TB -** Large portable storage devices used for file storage on Raspberry Pis.
- **Peripherals -** Displays, mouse, and keyboard to control Raspberry Pis as well as a case to protect the Raspberry Pi. An SD card reader with a USB adapter will be needed to connect

the SD card to computers

- **Flipper Zero -** A device that can transmit and receive Bluetooth signals and has computational power.
- **BBC Micro:bit -** An embedded system with capabilities such as Bluetooth and RF signal transmission

The Raspberry Pi system, shown below in Figure 2, will contain all of the components needed to operate the Raspberry Pi as a standalone computer consisting of the power source, display, keyboard, and mouse. An SD card will be used to run the applications and operating system of the Raspberry Pi. Four Raspberry Pis will be used to run the ESP32 and Zigbee modules as well as one more Raspberry Pi for the HackRF Module.

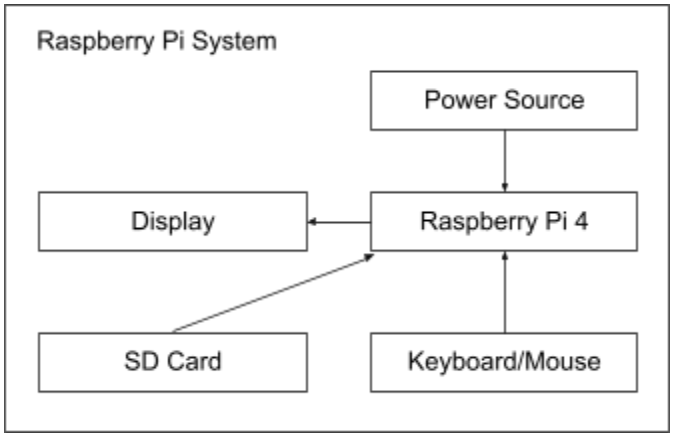

Figure 2: Raspberry Pi System Hardware Overview

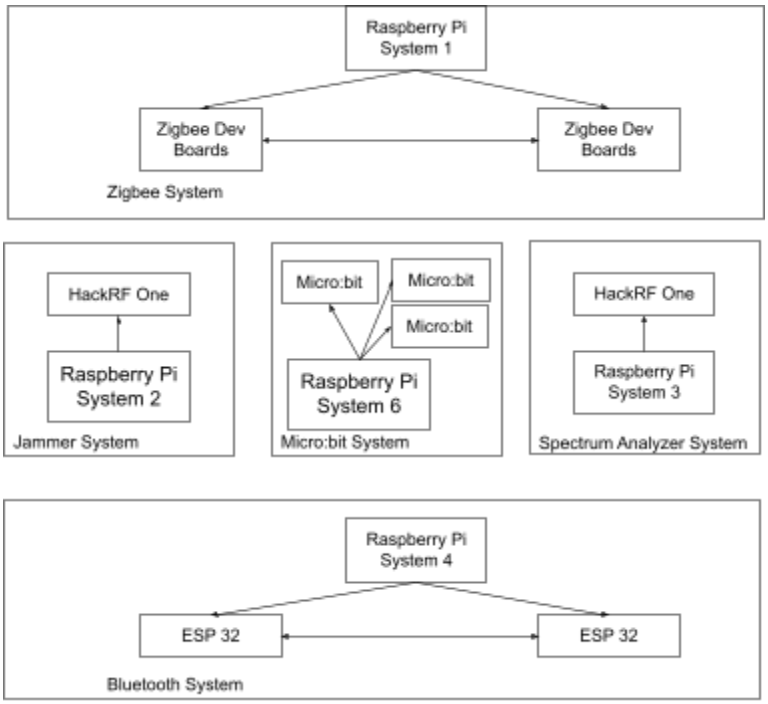

Figure 3: Diagrams of the Bluetooth, Zigbee, Micro:bit, and Software-Designed Radio Systems

The remaining additional systems - Zigbee, jammer, spectrum analyzer, and Bluetooth - are composed of the Raspberry Pi system combined with other hardware, as seen above in Figure 3. The Zigbee system consists of the Raspberry Pi system connected to two Zigbee development boards in a circular communication form. Both the jammer and spectrum analyzer systems consist of its over Raspberry Pi system connected to a HackRF One. The Bluetooth system consists of another Raspberry Pi system connected to two ESP 32's in a circular communication form.

The Flipper Zero is a self-contained system that handles the transmission of Bluetooth packets, reception of Bluetooth packets and computation all by itself. It also serves as a substitute for the Software Defined Radio System in attacks involving the Flipper Zero.

2.2 System Software Architecture

Two Raspberry Pis, which is the basis for each system, will be preloaded with Kali Linux on the SD card as well as the software to run the required module. Two Raspberry Pis will have Arduino IDE to run the ESP32s, two will have the Zigbee Development Software, and one will have the HackRF Tools and GNU Radio installed, all of which can be seen in Figure 4 below. Another Raspberry Pi will have BTLEJack installed.

The software tools used are defined as follows:

- **Kali Linux 2023.3 -** Linux-based operating system can be used for pen testing
- **Raspberry PI OS 2023-10-10 -** Operating system for Raspberry Pi
- **GNU Radio 3.10.8.0 -** Software used to control software-defined radios such as the HackRF One
- **HackRF Tools 2023.01.1 -** Command line to interact with HackRF
- **Arduino IDE 2.2.1 -** Used to program ESP32
- **Zigbee Development Kit Software 3.0 -** Software used to control Zigbee boards
- **Btlejack 2.1.1** - Software used to sniff and hijack Bluetooth connections

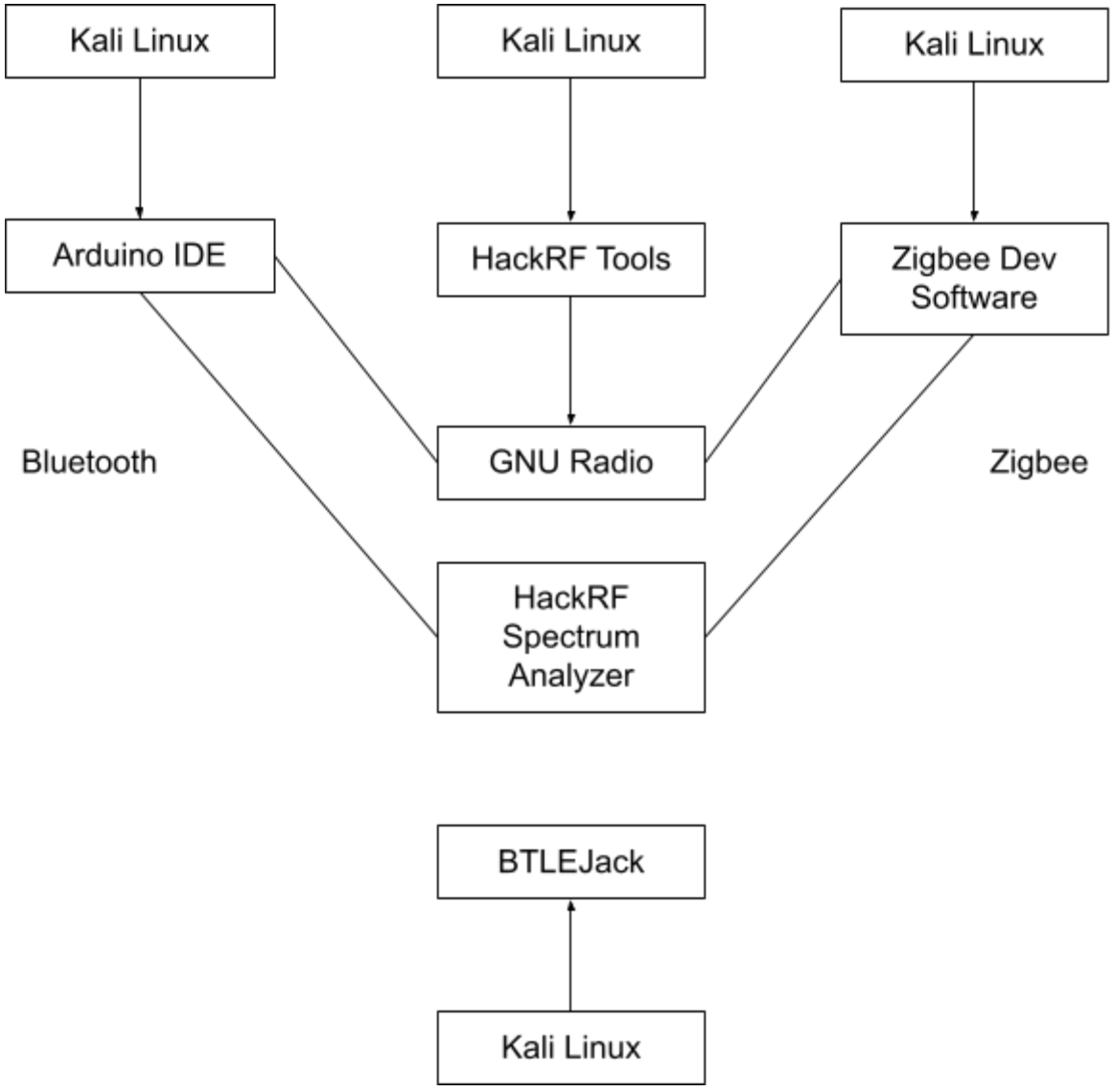

Figure 4: System Software Architecture Diagram

2.3 Internal Communications Architecture

Each Raspberry Pi will operate using an operating system and software through an SD card. The Raspberry Pis will connect to either the ESP32 or Zigbee Development boards through USB type A to USB micro B. The HackRF will connect to the Raspberry Pi via USB A to USB type C. The two ESP32 modules will communicate via Bluetooth and the Zigbee Development boards will communicate via Zigbee. The HackRF will send signals through a set frequency to intercept or block Bluetooth and Zigbee signals. The Flipper Zero will send and receive Bluetooth signals and use Flipper Zero apps to intercept or block the signals. Three Micro:bits will be used to send and receive Bluetooth packets. Figure 5 below shows how each device is connected to each system and communicates between itself. The solid lines indicate a wired connection such as between the Raspberry Pi and ESP32. The lines with a space in the middle indicate a wireless connection such as Zigbee or Bluetooth. The Flipper Zero is a standalone unit and does not communicate with any devices to function.

The devices discussed above are defined as follows:

- **Zigbee Development Boards -** Mesh networking for Zigbee devices
- **HackRF One -** Radiofrequency communication
- **ESP32 -** Wi-Fi and Bluetooth communication
- **Raspberry Pi -** Bluetooth, local network, and internet communication
- **SD Cards and SSDs -** Data storage and retrieval
- **Peripherals -** USB communication
- **SD Card Reader -** USB data transfer to computers
- **Flipper Zero -** Bluetooth communication

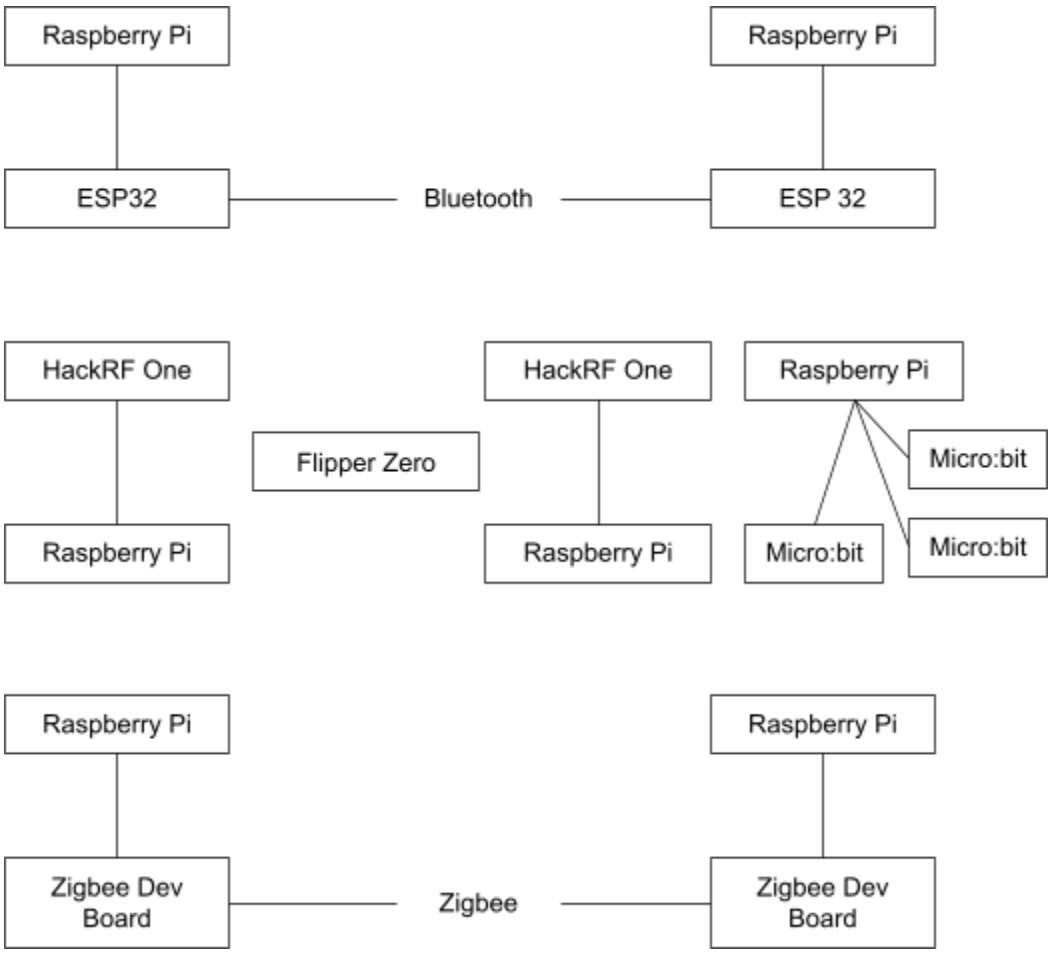

Figure 5: Diagram of Internal Communications Architecture

3 HUMAN-MACHINE INTERFACE

This section provides the detailed design of the system and subsystem inputs and outputs relative to the user/operator.

3.1 Inputs

The user is required to provide the frequency that is being monitored, by using the respective

system-controlling device and the used frequency (2.4 GHz to 2.5 GHz for Bluetooth and Zigbee). The frequency range in HackRF Spectrum Analyzer will need to be set to the desired frequency range, also 2.4 GHz to 2.5 GHz. In GNU Radio, the target frequency will need to be inputted depending on where the data is being transmitted as found in the spectrum analyzer, and the data rate depending on which device is being jammed. As seen in Figure 6 below, the BTLEJack requires that the user inputs a command into the Kali Linux terminal. The commands include scan (btlejack -s), install (btlejack -i), hijack connection (btlejack -f), or read (btlejack read).

Figure 6: User Inputs

3.2 Outputs

When the HackRF Spectrum Analyzer seen in Figure 7 is opened, the user will be presented with a graphical view of the data being transmitted on the selected frequency as well as the power in dB and frequency in MHz. When the GNU Radio jammer is run, the user will be presented with a graph that shows the power level and frequency that is being sent through the HackRF one. In this graph, the user can adjust the zoom and move through past data transmissions.

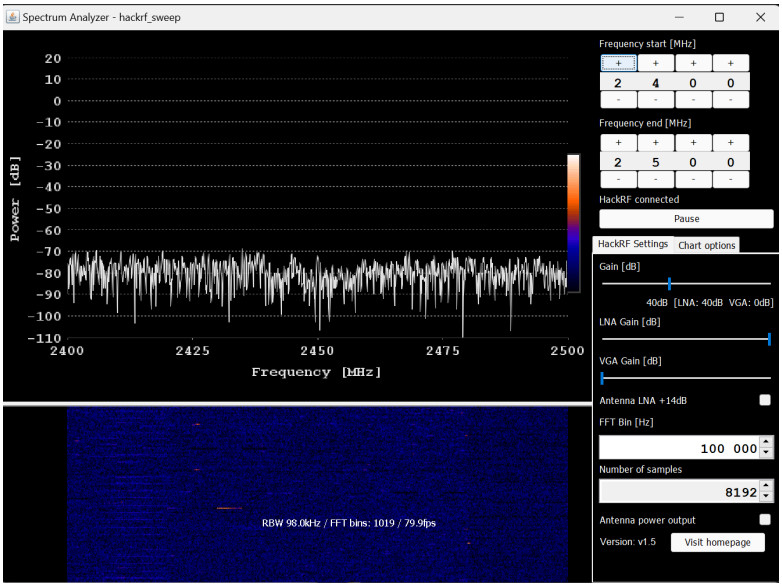

Figure 7: View of HackRF Spectrum Analyzer

The Flipper Zero creates several device outputs when performing a BLE spam attack. Using

Windows devices, the following popup in Figure 8 appears.

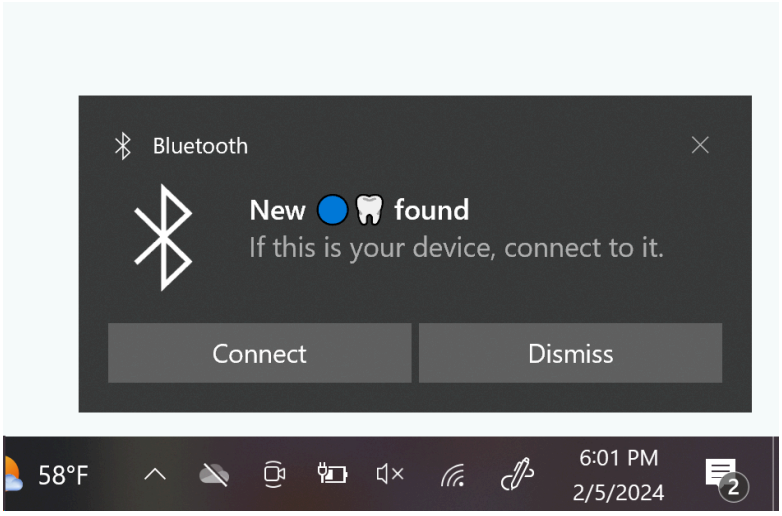

Figure 8: Windows BLE Spam Popup

When another BLE spam attack configured for the iPhone is performed, the popup in Figure 9 appears.

Figure 9: iPhone BLE Spam Popup

When BTLEJack is launched it will display a terminal view where the user can launch different attacks by using the corresponding command such as sniffing, jamming, and decoding data. Figure 10 shows BTLEJack active in sniffing mode.

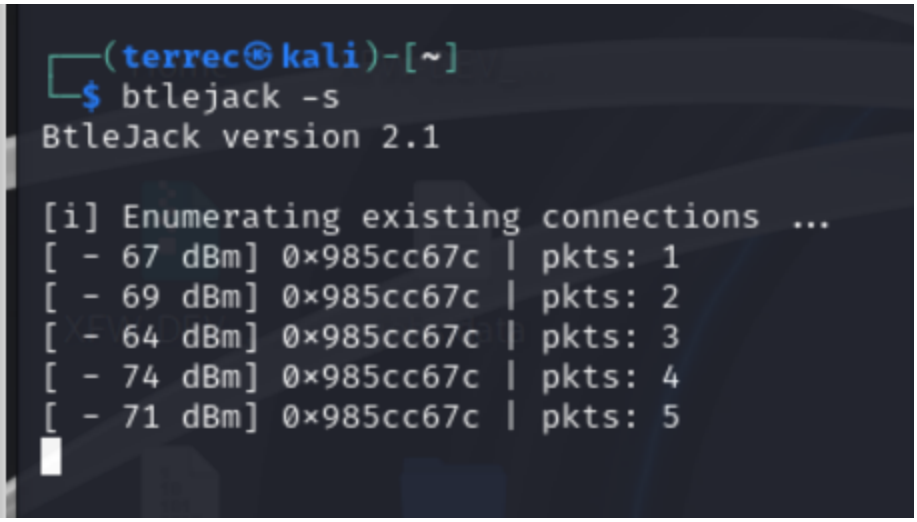

Figure 10: BTLEJack in Sniffing Mode

4 DETAILED DESIGN

This section provides the information needed for a system development team to build and integrate the hardware components, code, and integrate the software modules, and interconnect the hardware and software segments into a functional product.

4.1 Hardware Detailed Design

The system is composed of several identical modules that connect to form either a Bluetooth, BLE, or Zigbee network.

4.1.1 Raspberry Pi System

Each Raspberry Pi system is composed of a Raspberry Pi 4, a power supply, a display, a keyboard, and a mouse.

The mouse is an official Raspberry Pi mouse. The mouse connects to the Raspberry Pi board using a USB connection. The male USB connector is integrated into the mouse and will connect with a female USB port on the Raspberry Pi board. The mouse receives its power through the USB connection.

The keyboard is an official Raspberry Pi keyboard. The mouse connects to the Raspberry Pi board using a USB connection. Both the keyboard and the board have female USB ports and will be connected using a USB cable. The keyboard receives its power through the USB connection.

The display is an ELECROW 7-inch mini monitor. The display connects to the Raspberry Pi board using a HDMI connection. The display contains a female micro USB port and connects to the board using a micro USB to HDMI adapter which connects to the HDMI port on the board. The display receives its power through the HDMI connection. The display has a resolution of 1024x600 pixels and a refresh rate of 60Hz.

The power supply is the CanaKit 3.5A USB-C Raspberry Pi 4 Power Supply. The power supply connects to the Raspberry Pi board using a USB connection. The male USB connector is integrated into the supply and will connect with a female micro USB port on the Raspberry Pi board, as seen in Figure 11 below. Figure 11 shows the Raspberry Pi and its peripherals such as the power supply, mouse, keyboard, and monitor.

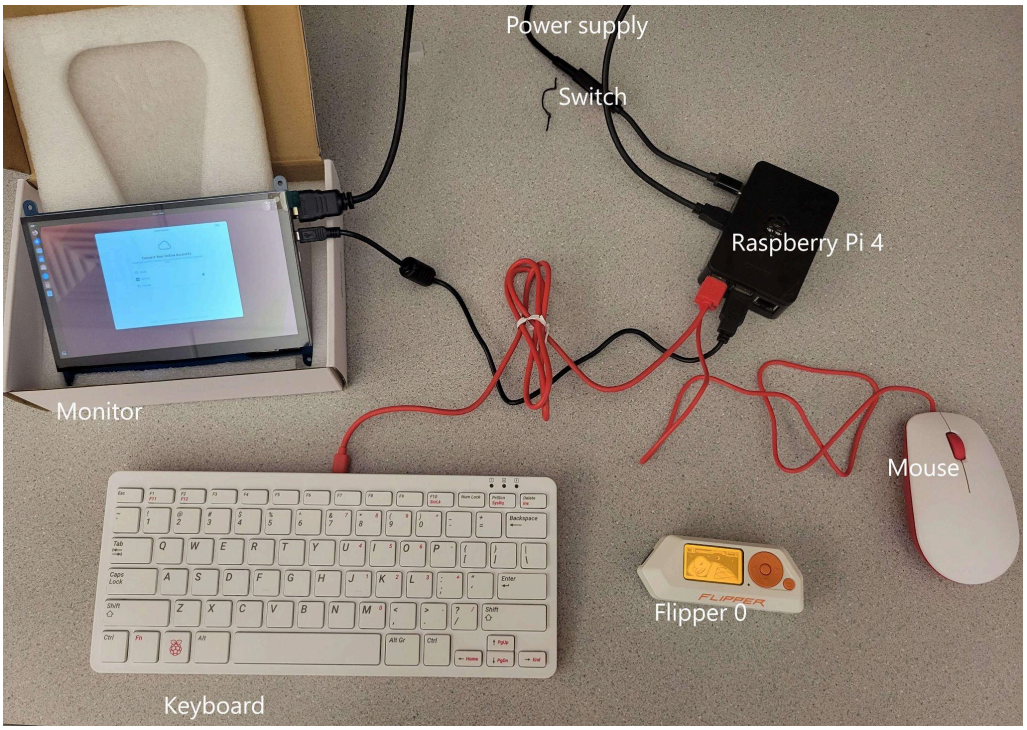

Figure 11: A Visual of the Raspberry Pi System

4.1.2 Bluetooth System

Each module that is part of the Bluetooth network is composed of a Raspberry Pi system and an ESP32 development board. The Raspberry Pi systems connect to the Bluetooth network using ESP32 development boards. The ESP32 development board connects to the Raspberry Pi board using a micro USB cable as shown in Figure 12 below. Figure 12 shows the Bluetooth system with the ESP32 and Raspberry Pi system. Both components have female micro USB ports so a male-to-male micro USB cable will be used. The micro USB cable supplies power to the ESP32 development board. The ESP32 development board is what is used to transmit and receive data from other Raspberry Pi systems. The system remains consistent for both BLE and Advanced Audio Distribution Profile (AD2P).

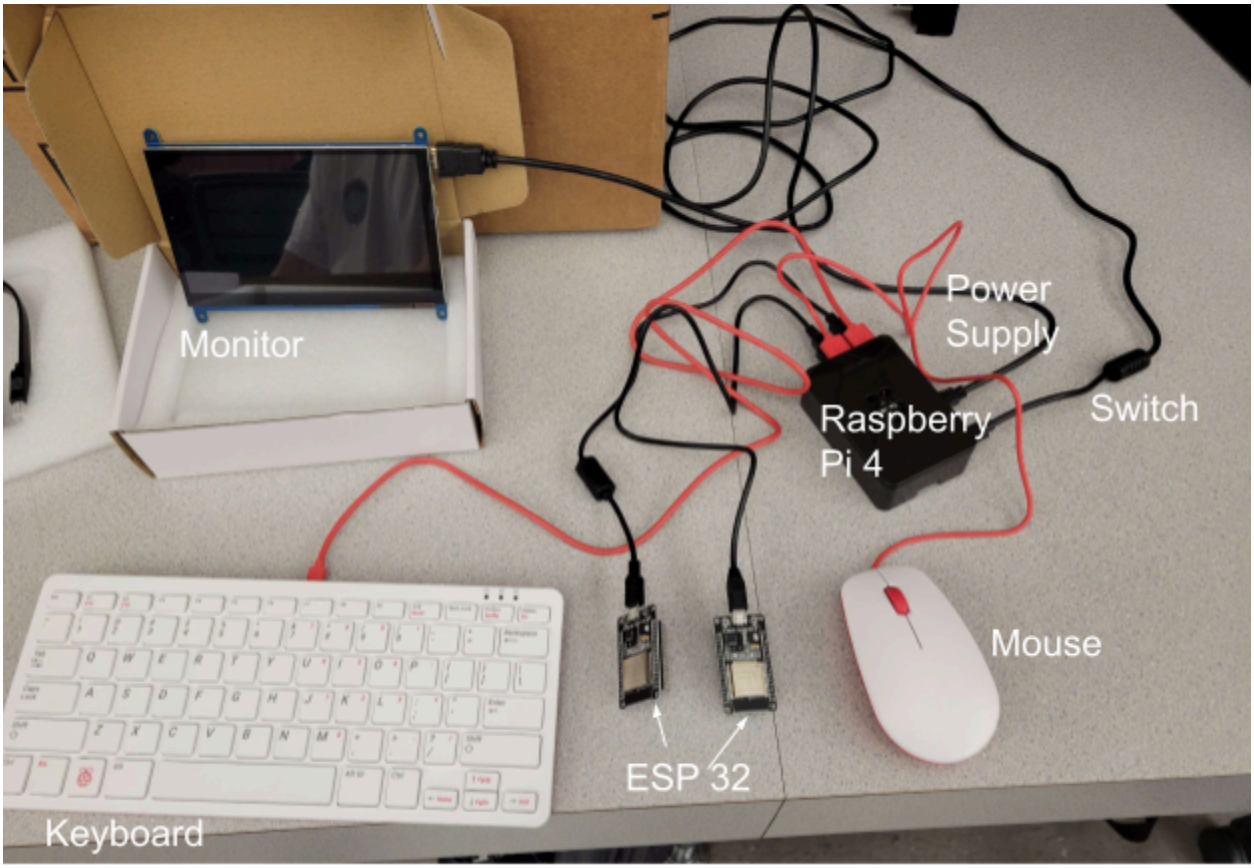

Figure 12: Bluetooth System

4.1.3 Zigbee System

When the Raspberry Pi system is sending and receiving data over a Zigbee network a Zigbee development board is used instead of an ESP32 development board. The Zigbee development board connects to the Raspberry Pi board using a micro USB cable. The Zigbee system is controlled by the Raspberry Pi system, as shown in Figure 13. Both components have female micro USB ports so a male-to-male micro USB cable will be used. The micro USB cable supplies power to the Zigbee development board. The Zigbee development board is what is used to transmit and receive data from other Raspberry Pi systems.

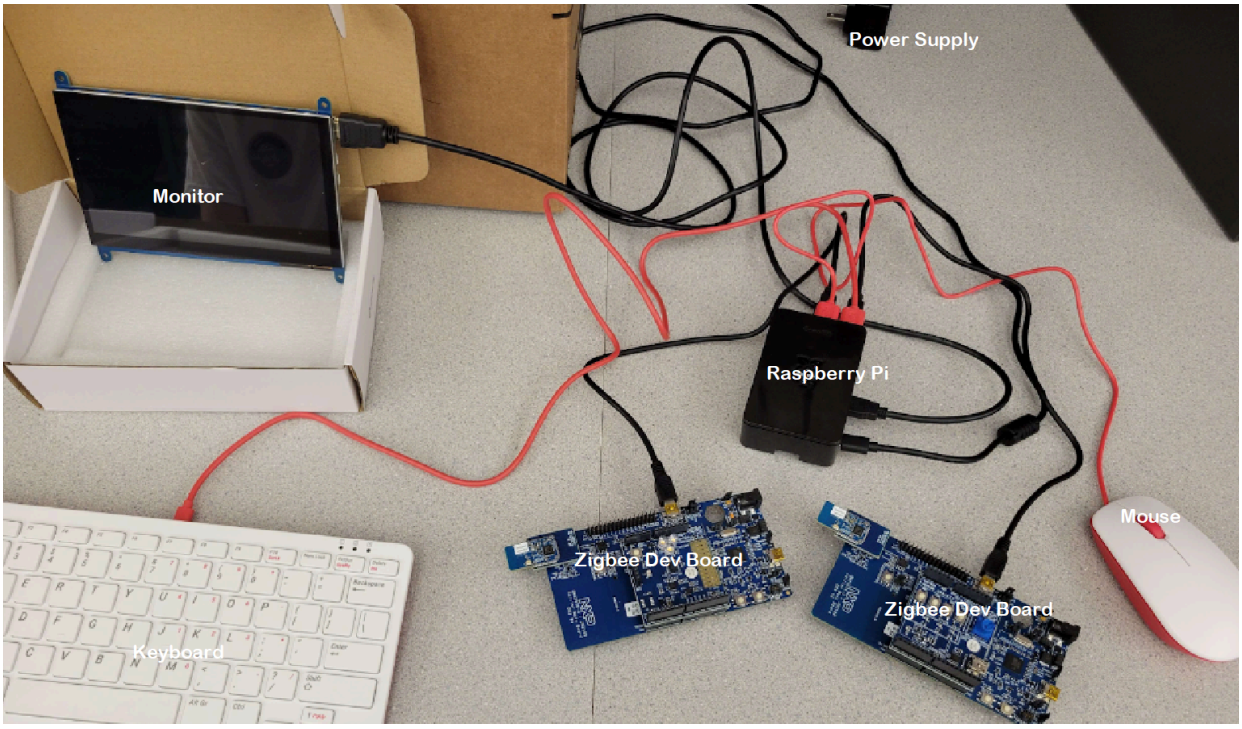

Figure 13: Zigbee System

4.1.4 Software-Defined Radio

In other cases, a Raspberry Pi system is connected to a HackRF One The HackRF connects to the Raspberry Pi board using a micro USB cable. The HackRF system is controlled by a Raspberry Pi system, as shown below in Figure 14. Both components have female micro USB ports so a male-to-male micro USB cable will be used. The micro USB cable supplies power to the HackRF One.

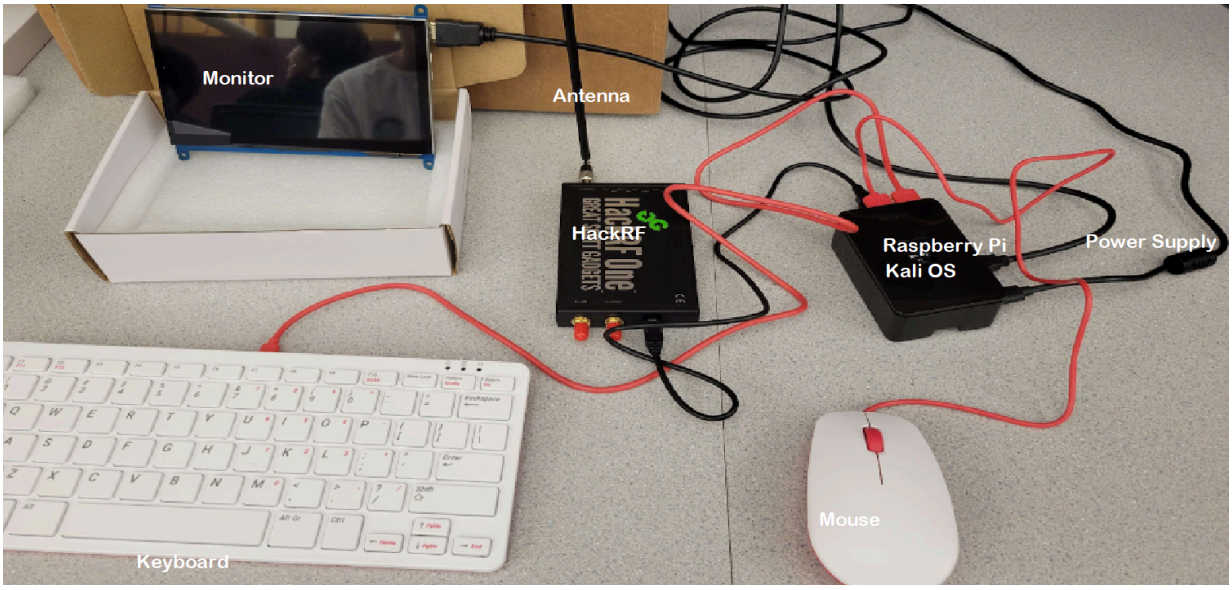

Figure 14: Software Defined Radio System

4.1.5 Flipper Zero

The Flipper Zero acts as its own, independent system. While it presents itself as a “electronic pet dolphin game”, as can be seen in Figure 15 below, the interface of the Flipper Zero allows for a variety of actions by scanning, receiving, and transmitting various signals such as Bluetooth, RFID, NFC, among others. Both the additional desktop and phone applications allow for customization and optimization of its BLE capabilities. The BLE receiving devices from our current testing include iPhones and Microsoft laptops.

Figure 15: Flipper Zero

4.1.6 Micro:bit System

A Raspberry Pi system is connected to three Micro:bits. The Micro:bits connect to the Raspberry Pi board using a micro USB cable, since there are only two available USB ports one Micro:bit is connected via the USB port on the back of the keyboard. The Micro:bit system is controlled by a Raspberry Pi system, as shown below in Figure 16. Both components have female micro USB ports so a male-to-male micro USB cable will be used. The micro USB cable supplies power to each of the Micro:bits.

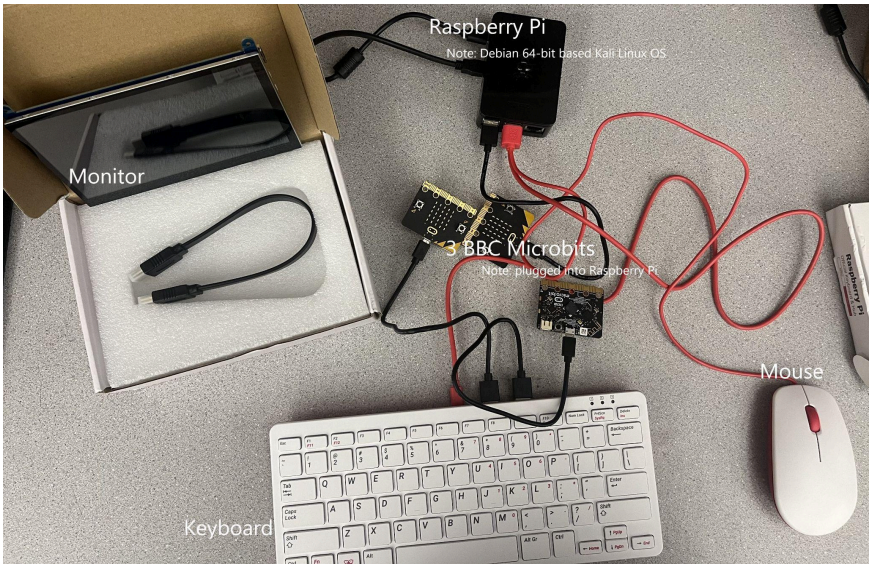

Figure 16: BBC Micro:bit System Design

4.2 Software Detailed Design

The software is broken up into three sections: the jammer, frequency analyzer, and communication system as shown in Figure 17. The jammer system is responsible for transmitting RF signals on the 2.4 GHz to 2.5 GHz frequency. The frequency analyzer system is responsible for analyzing and decoding frequencies on the 2.4 GHz to 2.5 GHz to detect Bluetooth and Zigbee data transmission as well as confirm the jammer is working properly. The communication system is responsible for transmitting and receiving Zigbee and Bluetooth signals. The Hijacking system is responsible for receiving, transmitting, and decoding Bluetooth packets.

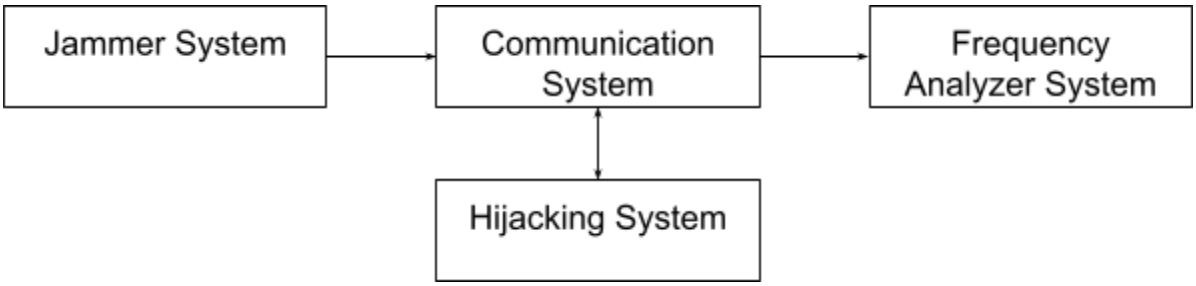

Figure 17: Software Architecture

The Flipper Zero has its own detailed software. Various personal changes or new firmware may be downloaded to the Flipper Zero as part of our research. These firmware changes will aid in utilizing Flipper Zero's BLE capabilities.

4.2.1 Jammer System

The jammer system consists of two sub-systems: the RF signal generator and the control interface as shown in Figure 18. The RF signal generator is responsible for creating and transmitting RF signals in the 2.4 GHz to 2.5 GHz frequency band. It uses a high-frequency oscillator and a power amplifier to generate and transmit the signals. The control interface is a software component that allows the user to control the operation of the jammer system. It

provides options to start or stop the jamming, adjust the frequency and power of the RF signals, and monitor the status of the jammer system.

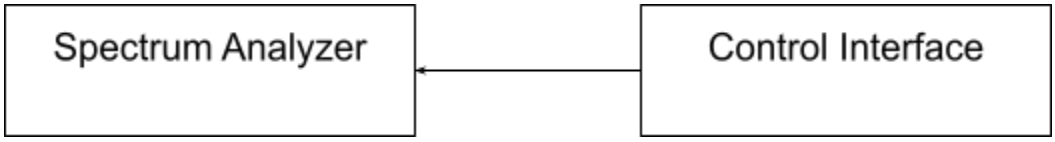

Figure 18: Jammer System

4.2.2 Frequency Analyzer System

The frequency analyzer system is designed to analyze and decode frequencies in the 2.4 GHz to 2.5 GHz band. It consists of a spectrum analyzer and a decoding module as shown in Figure 19. The spectrum analyzer scans the frequency band and captures the signals present in it. The decoding module then processes these signals to identify and decode any Bluetooth and Zigbee data transmissions. The system also verifies the operation of the jammer by detecting the presence of its RF signals in the frequency band.

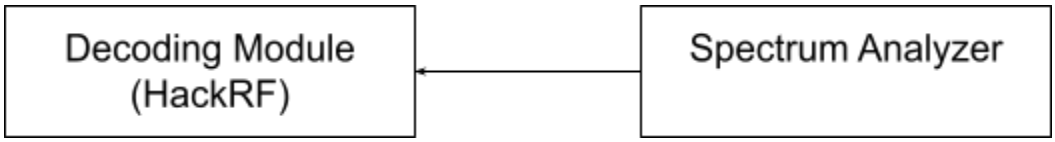

Figure 19: Frequency Analyzer System

4.2.3 Communication System

The communication system is responsible for handling Zigbee and Bluetooth communications. It consists of two sub-systems: the Zigbee communication module and the Bluetooth communication module as shown in Figure 20. The Zigbee communication module handles the transmission and reception of Zigbee signals. It uses a Zigbee transceiver and a microcontroller to encode and decode the Zigbee data. The Bluetooth communication module handles the transmission and reception of Bluetooth signals. It uses a Bluetooth module and a microcontroller to encode and decode the Bluetooth data. Both modules are designed to operate in the presence of the jammer system's RF signals. Both modules track whether Bluetooth and Zigbee packets are received. The communication system sends data that the client system will verify with its own prediction to ensure the correctness of the message, any faults would lead us to determine how successful the attack was. There also exists an alternate version of the Bluetooth Communication Module that is used to simulate use cases in which a Bluetooth audio profile would be used. The alternate communication module uses the advanced audio distribution profile to send audio between the Bluetooth modules. The alternate communication module also makes a prediction and verifies it to determine how successful the attack was.

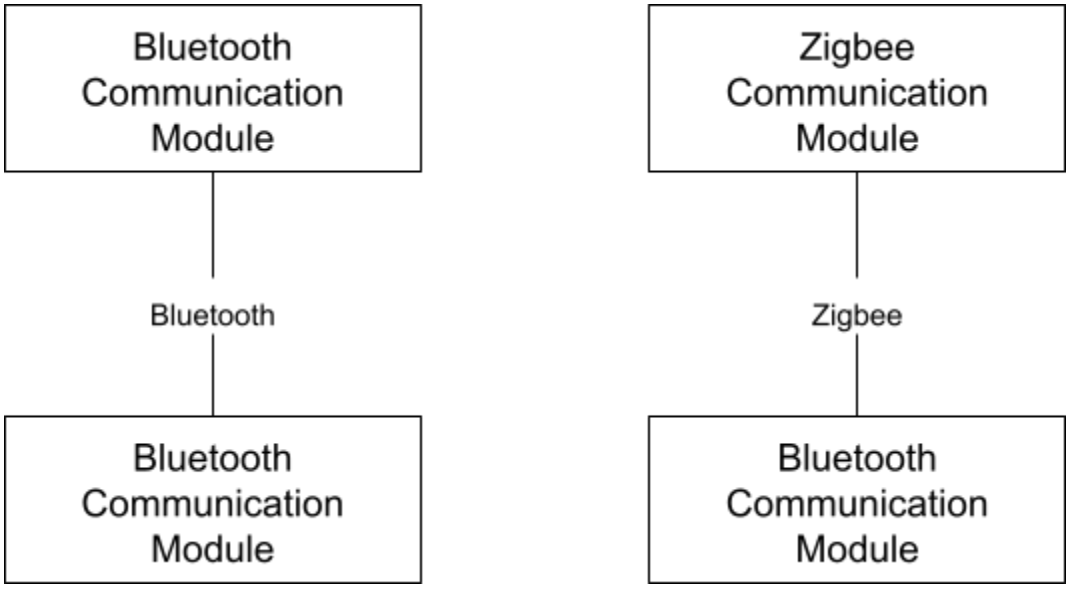

Figure 20: Communication System

4.2.4 Hijacking System

The interception system in Figure 21 below is responsible for intercepting and decoding Bluetooth packets. It consists of three modules: the transmission module, decoding module, and receiver module. The receiver module is responsible for discovering target devices on 2.4 GHz to 2.5 GHz frequencies and acquiring their advertising address. Once a target is connected, the decoding module can analyze the income Bluetooth packets which are presented to the user in hexadecimal strings. The transmission module allows for continued data flow between the Micro:bits and the intercepted devices.

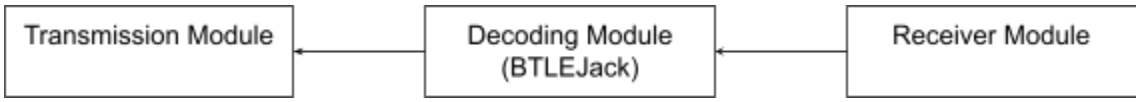

Figure 21: Interception System

4.3 Internal Communications Detailed Design

Modules on the Bluetooth network communicate with each other following the specifications outlined by the Bluetooth protocol. Modules on the Zigbee network communicate with each other following the specifications outlined by the Zigbee protocol.

5 EXTERNAL INTERFACES

This is a closed system that does not communicate with any external systems.

6 SYSTEM INTEGRITY CONTROLS

The system will pick up Bluetooth, BLE, and Zigbee signals which are widely used by most IOT devices. These signals can be picked up by anyone with a frequency analyzer, which can be installed on most mobile devices and computers. The system will be used in a controlled

environment to prevent signal interruption and potential interceptions. This will also ensure that the Bluetooth, BLE, and Zigbee signals can be correctly detected by the HackRF Spectrum Analyzer. The data collected from the HackRF Spectrum Analyzer and Micro:bits will be downloaded and stored locally on the Raspberry Pi for future analysis if needed. The data will be securely stored and deleted once used to prevent frequency data from exposure.

The Raspberry Pis will have a kill switch on the power supply in case of an emergency. If the kill switch is pressed on the power supply the Raspberry Pi will lose power instantly. This will also be used in case of a signal leak to prevent potential exposure of Zigbee, Bluetooth, or jamming signals.