An-Najah National University

Faculty of Engineering & Information Technology

Presented in partial fulfillment of the requirements for

Bachelor degree in Computer Engineering

Graduation Project 2

Flipper Zero

Students:

Mohammad Ahmad

Salah Tanbour

.

.

Supervisor:

Dr. Aladdin Masri

Dr. Muhannad Al-Jabi

September 8, 2024

Acknowledgment

We would like to express our gratitude to everyone who helped making this project possible. Special thanks to our two supervisors: Dr. Alaadin Masri for his support and guidance throughout the project period, and for Dr. Muhanad Al-Jabi.

Abbreviations

- RFID: Radio Frequency Identification
- GPIO: Gerneral Purpose Input Output
- NFC: Near Field Communication
- IR: Infrared

Abstract

Flipper Zero is a versatile and hand-held multi-tool used for interacting with digital devices using various communication schemes. it includes wide range of protocols for communication, including 125Khz, NFC, Sub-Ghz, Bluetooth. It is also capable of interfacing with external hardware components which helps in expanding the functionalities of the Flipper Zero. The goal of this project is to design and build a proof-of-concept hardware device that functions similarly to Flipper Zero using common modules to serve as an educational experience. The Flipper Core includes a user-friendly LCD interface to select and activate the various functionalities of the device, along with a keypad or a web app for user input.

Key features include storing and emulating infrared, 125khz RFID and 433mhz signals. Bluetooth used as emulation for mouse and keyboard. and basic external hardware interface using GPIO pins. The project's core aim is to replicate essential Flipper Zero functionalities while offering a solid foundation for future enhancements and applications.

Despite challenges, the project successfully delivers a somewhat functional prototype, a good base for further development in the coming years, with the capability to serve in practical and educational fields. Future work will focus on enhancing the base to handle more real-world application.

Contents

List of Figures

Chapter 1

Introduction

1.1 Background and Literature Review

Hand-held and portable multi-functional devices have been on the trend for a while. They have evolved into powerful tools that can interact with many electrical devices using various communication schemes. This feature made them valuable to ethical hackers and professionals across fields such as cyber-security.

One of the most notable examples of such devices is Flipper Zero. "Flipper Zero is a portable multi-tool for pentesters and geeks in a toy-like body. It loves hacking digital stuff, such as radio protocols, access control systems, hardware, and more. It's fully open-source and customizable, so you can extend it in whatever way you like." [1]

Flipper zero has a wide range of use cases, including RFID, NFC, infrared (IR), and GPIO control. This versatility enables interaction with various systems, including access control devices, smart home systems, televisions, industrial equipment, and IoT devices.

Flipper zero has become widely used in security research and penetration testing [2] [3] [4]. In [2], the flipper zero was used to discover vulnerabilities and then rank each one according to its severity and potential impact. In [4], the flipper zero was used to clone RFID tags to different ones, and it succeeded in authorizing access to the clones as if they were the real ones. Lastly, in [3], the flipper zero was used as a hacking tool to test the maximum distance an IoT device can receive eavesdropper's signal.

1.2 Project Overview

The Flipper Core project works as a proof of concept for re-implementing the necessary features of Flipper Zero. This project focuses on the most crucial features of a Flipper zero.

By concentrating on the core features, Flipper Core ensures the feasibility of building a simplified yet functional version of the original device. Flipper Core sets the groundwork for future

enhancement and improvements, in addition, provides a functional prototype and proof of concept of the Flipper Zero.

1.3 Inspiration

This idea was inspired by the uniqueness of it based on NNU graduation projects in the last semesters. We've wanted to stay away from the traditional projects consisting of mechanical and moving parts such as motors and wheels, and we centered our research on portable and hand-held devices that don't demand a very high budget.
In addition, videos online went viral about the use of flipper zero, a hacking tool for geeks, and we'd thought it would be both educational and beneficial for to try to make a DIY version of it, we'd learn more about the different schemes that can be used to transfer data, and the corresponding devices that function with these schemes.

1.4 Target Audience and Usefulness

The audience of such a project would include people interested with technology and want to try cool and unique devices, in addition, people wanting to switch from using multiple remotes for each different device they have to a device that works as all-in-one. Our device for example can combine multiple Infrared remotes into one. Moreover, such device can be used for testing the security of various systems in terms of access control, it can detect vulnerabilities in old security systems and shine the light on the need to develop newer and safer systems.

1.5 Key Features and Functionalities

As mentioned above, our Flipper Core tries to implement a subset of the functionalities Flipper zero have, this subset is as follows:

1.5.1 Bluetooth

- Flipper Core can emulate a mouse and control the mouse on a PC with it.
- Emulate a keyboard by sending words to the PC to display on the screen.

1.5.2 Low Frequency RFID (125Khz)

- read 125Khz rfid tags and display its details (version, tagId, checksum)
- store an rfid tag for later use
- emulate a tag stored in the list
- delete an item from the list

1.5.3 High Frequency RFID (NFC 13.56Mhz)

- read NFC cards and/or tags and display its details
- Clone NFC tags

1.5.4 InfraRed

- read infrared signals from IR remotes and display its details
- store infrared signals for later use
- re-transmit stored IR signals
- delete stored IR signals from the list

1.5.5 Sub-Ghz

This feature allows flipper core to use sub-ghz frequency signals.

- Radio interference: used to interfere with radio signals and make noise (old radio box signals)
- read 433mhz signals
- store 433mhz signals
- playback stored 433mhz signal
- delete stored signals from the stored list

1.5.6 GPIO

This feature allows the flipper core to interface with external hardware components by exposing 6 GPIO pins for use. In the flipper core, this is used in two different modes:

Pre-defined Scripts

Flipper Core comes with pre-defined scripts for various external modules that enable the flipper core to be commercialized and encourage customers to buy add-ons to the device. We've implemented two pre-defined scripts:

- Ultrasonic sensor
- Temperature sensor

Controlling individual pins

with this mode, we can set or clear individual pins on the GPIO ports.

1.5.7 Supporting Features

- Persistent Storage: This allows the Flipper Core to remember stored signals even after a reboot, done by using non-volatile storage element.

- Web app: Control the flipper core without the use of keypad.

1.6 Challenges and Constraints

A handful of challenges were faced when building the Flipper Core, which lead to falling short on time, these challenges being:

- limited expertise in such a domain (RFID, NFC, Sub-Ghz, ...etc)
- limited resources online on such usage of hardware components
- shortage of required components to implement some functionalities, so a workaround had to happen.
- Increased delivery time as required components were ordered from non-local shops.
- Trying to stay in the budget range and not exceed it.

Chapter 2

Methodology

We started gathering information, figuring out how far we can reach with this idea, collecting metrics, looking for solutions, technologies that can help us building the project.

2.1 Teamwork

Using Git & Github was an excellent choice to achieve asynchronous working on the coding part of the project meanwhile we conducted meetings synchronous on Discord for online and on-site at NNU for sharing daily progress and desgining/working on the hardware part of the project.

Leveraging Git & Github to accelerate the development process and ensure the reliability of the coding part of the project by branching from the main branch that represent the fully-tested version of the code and sub-branches represent the feature that is not reliable, fully-implemented yet being developed.

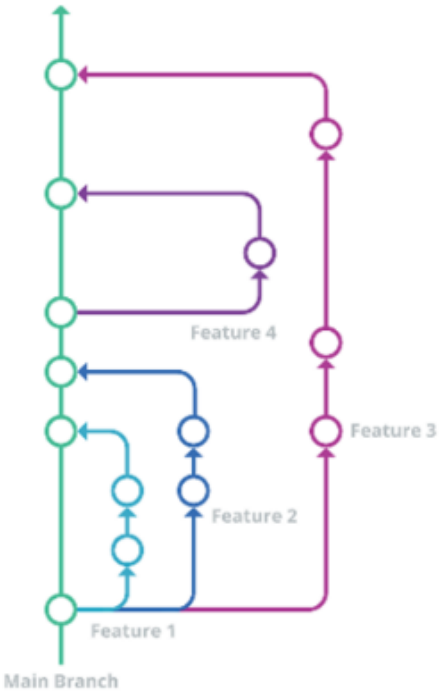

Figure 2.1: Development flow.

After finishing a feature and being fully tested then merge it to the main branch to insure the reliability of the project and re-test it again with the present of the other features.

2.2 Design

The hardware components is connected and top of two-layered 10x22CM.

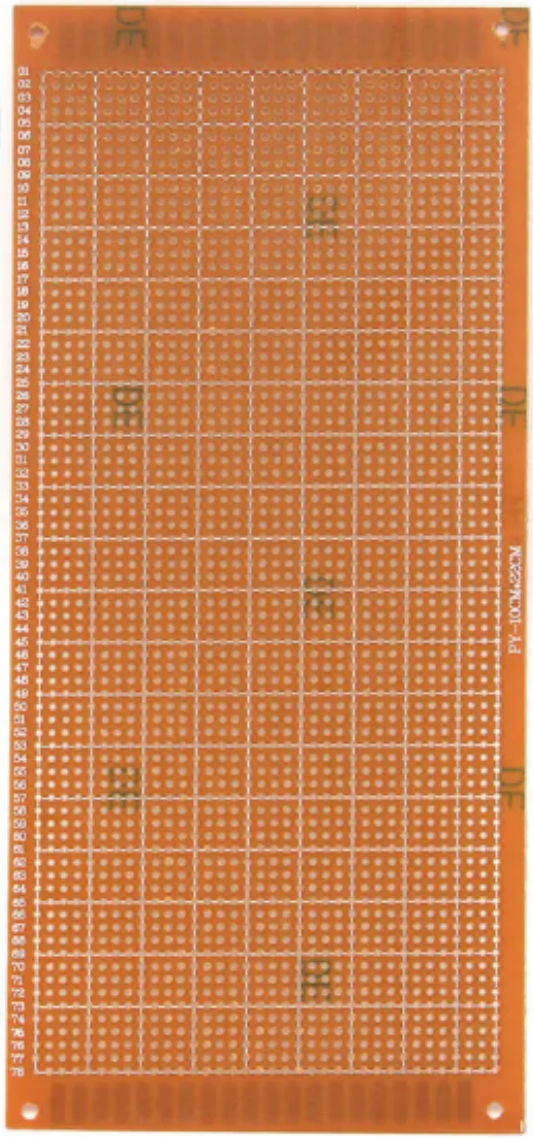

Figure 2.2: 10x22 CM Board

The top layer represent the user interface containing a 4x20 LCD and a keypad with part of the hardware components as in the shown figures.

Figure 2.3: Keypad

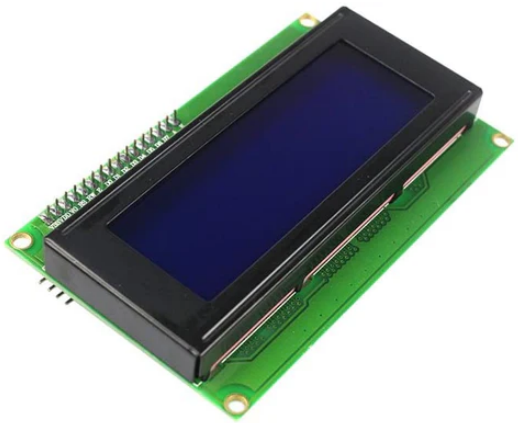

Figure 2.4: 4x20 LCD

The keypad gives the user the ability to navigate through the menus in side the system and choose the desired feature.

The bottom layer is used for extension to provide more space to put components on.

2.3 User Interactions

Mainly the user will use the keypad with buttons that we mention it above to navigate through the menu options and select the desired option that lead to sub-menus or immediately working feature, the SEL button will use it to select a specific option within the current displayed menu, we built a hierarchy to easily insert menus and sub-menus, with UI components across the system such as the keyboard which is widely used to get input from the user text/numbers.

2.4 Electronics Components

2.4.1 ESP32

Represent the core of our project containing the most valuable logic and represent the orchestrator that ON/OFF modules, process, store and CRUD data.

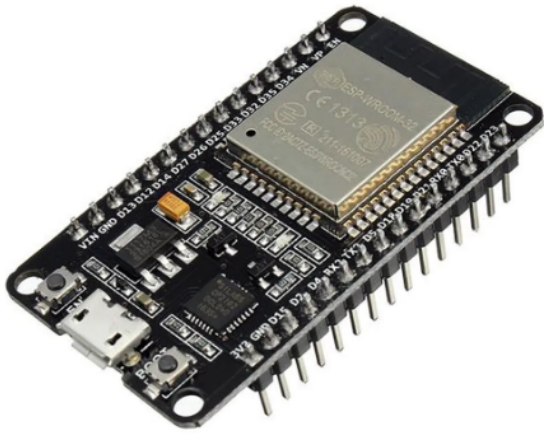

Figure 2.5: ESP32

2.4.2 Arduino UNO

Due to the massive numbers of modules, ESP32 alone can not handle all of them, we found lack of pins, therefore, we used Arduin UNO and connect it with the esp32 that sends commands through UART to the Arduino UNO to do jobs such as GPIO feature and CC1101 module.

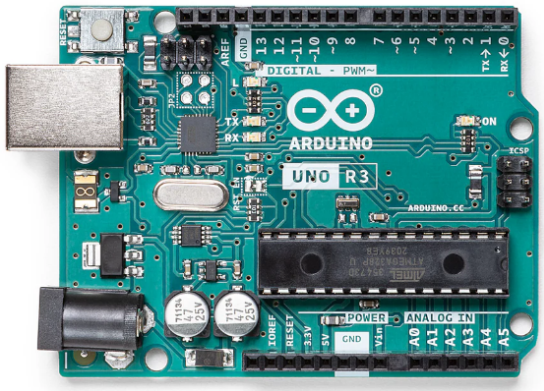

Figure 2.6: Arduino UNO

2.4.3 IR Receiver

This module is responsible of receiving IR signals, then the ESP32 process it for further usages.

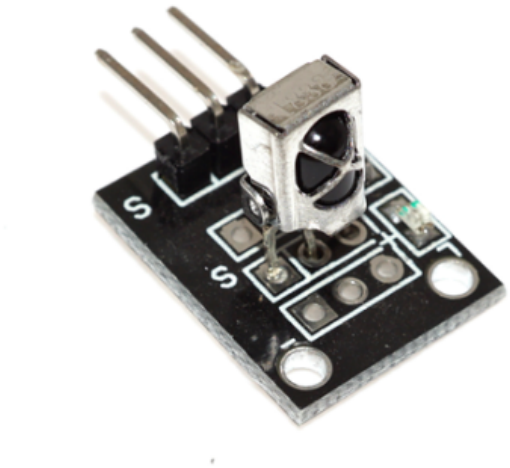

Figure 2.7: IR Receiver

2.4.4 IR Transmitter

This module is responsible of transmit the selected IR signal that stored in the Program memory.

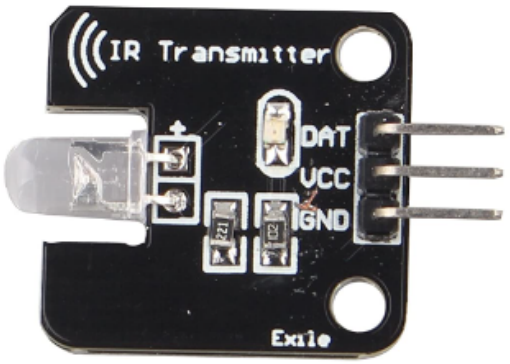

Figure 2.8: IR Transmitter

2.4.5 Radio Signal

Our project represent a universal remote so it is a must to deal with radio signals so we used multiple modules to achive this feature.

CC1101

This module connected directly with the Ardiuno UNO and controlled via UART the ESP32 depends on the user input send commands to the Arduino to activate it, this module main purpose is for do interference with other radio signals by entering the desired frequency you want to interference with.

Figure 2.9: CC1101

RX470-4

RX470-4 is a specific purpose device unlike the CC1101 that only can read from 433Mhz devices such as the cars or garages remotes and then store the signal for further usages this whole operation done by the ESP32 microcontroller from reading/storing to emitting.

Figure 2.10: RX470-4

WL102-341

Like the RX470-4, WL102-341 is also specific purpose device that can only transmit radio signals in our case it will transmit the data we recieved previously from the RX470-4 module that stored in the program memory, therefore, emulate a garage/car or any remote that support 433Mhz radio signals.

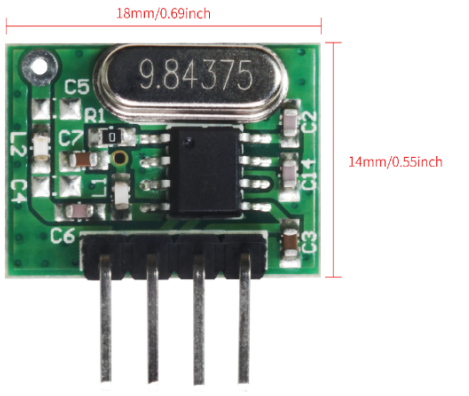

Figure 2.11: WL102

2.4.6 ESP32 Integrated EEPROM

Leveraging EEPROM to provide a better experience to the user that the IR signals the user store will be stored permentaly in the flash memory inside the ESP32 microcontroller this enable the user to save and re-use IR signals even after shutting down the FlipperCore.

2.4.7 ESP32 Integrated Bluetooth

Using the integrated bluetooth module to emulate Keyboard/Mouse, the user navigate to bluetooth option in the main menu and a sub-menu will open to further select which device wants to emulate.

2.4.8 125Khz RDM6300

This module is used to read 125Khz RFID tag/card related data such as ID, checksum and other information, the user can choose ethier just read option it to read related data or store, after storing it, the card/tag details will be stored in the for further emulating usages.

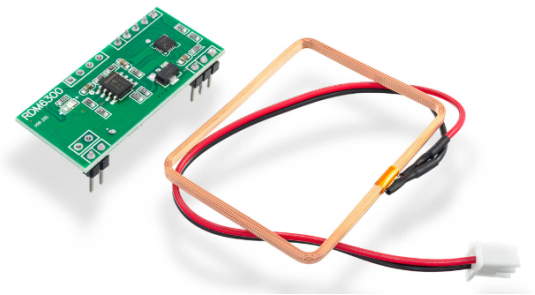

Figure 2.12: RDM6300

2.4.9 1uH Inductor

Used to eumlate the previously stored data that captured by RM6300 Module and stored in a list that the user can navigate through when go to (125Khz rfid ¿ Emulate) a menu will pop up with the saved 125Khz cards/tags that the user can choose one and emulate.

Figure 2.13: 1uH Inductor

2.4.10 NPX PN532

This module used for reading and writing on 13.52Mhz tags/cards, so the user will be able to read the data and the tag Id, store it in the system that act like history, and write/save on tags data for further usages.

Figure 2.14: NPX PN532

2.4.11 RC522

This module used for emulating 13.52Mhz tags/cards that like the 125Khz the user can read and store it to and use it later to emulate a tag/card without the need for the card/tag presence.

Figure 2.15: RC522

2.4.12 125Khz RFID Lock

This lock specifically we bought to test the 125Khz tags/cards emulation process that the components responsible of (RDM6300 and 1uH Inductor) by storing and re-using it, first the user should select from the main menu the option 125Khz RFID then select store the LCD will prompt a message "Scanning" the user will put the tag in the coil then read it and choose the name using the general purpose keyboard UI Element, then it will be saved in the program memory, to use it go again to the 125Khz RFID then select "Emulate" option then the ESP32 will process the request and emulate this tag in the 1uH Inductor.

Figure 2.16: RFID Lock

2.4.13 Lithium Ion Batteries

In order to power up the FlipperCore, we used Dual Lithium Ion Batteries.

Figure 2.17: Lithium Ion Batteries

With battery indicator to indicates how much left in the batteries in order to re-charge or switch to fully-charged batteries.

Figure 2.18: Battery Indicator

Chapter 3

Results

in 3.3 3.1 3.2, the final draft of the project, a working prototype with the features described above.

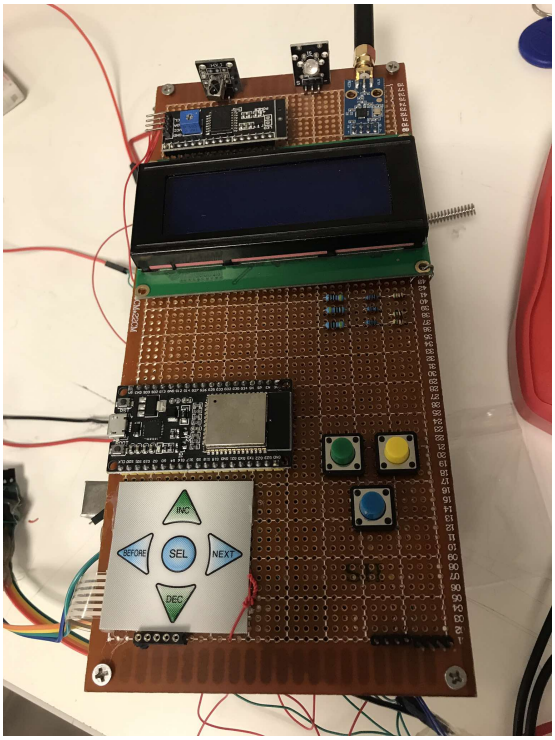

Figure 3.1: Front view of the Flipper Core

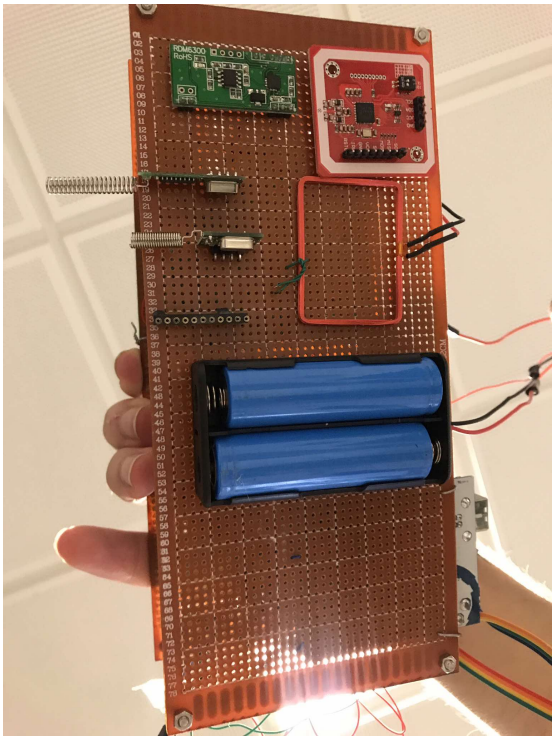

Figure 3.2: Back view of the Flipper Core

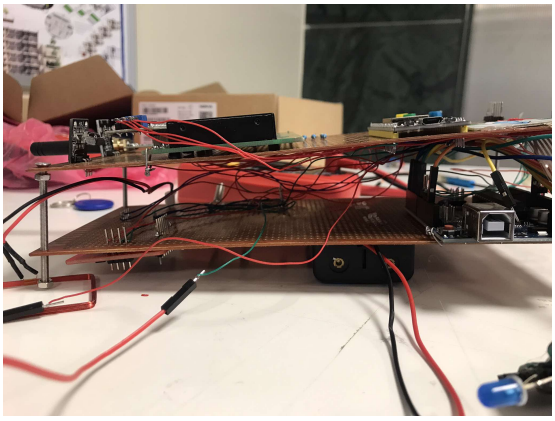

Figure 3.3: Side view of the Flipper Core

Chapter 4

Conclusion and Future Work

4.1 Conclusion

In this project, we created a hand-held device that offers a proof of concept on Flipper Zero mini clone, focusing on portability, functionality, and accessibility, by implementing IR Transceiver, Radio Transceiver and RFID (Low and High frequency) we successfully explained the potential of our device Flipper Core.

4.2 Feature Work

- Use removable expandable storage such as an SD card
- Implement more complex penetration schemes, such as, rolling codes in car keys.
- Increase the distance of the IR transmitter
- Support more well-known Sub-Ghz frequencies like 315 MHz, 868 MHz, 915 MHz
- Implement a repository web application that the user can upload any types of signals and set it as public or private on demand, in this way users from worldwide can access it download signals into SD Cards and upload it to Flipper Core.
- Provide a way to the users to make their own extensions for GPIO with custom code that will upload it to Flipper Core, then connect the extension with the uploaded custom code via SD Card to run it and use it on demand.

Bibliographic

[1] Flipper Devices Inc. *Flipper Zero - Multi-tool Device for Geeks*. Accessed: 2024-09-07. 2024. URL: https://flipperzero.one/.

[2] Joy Winston. "Evaluating Iot Device Security: Penetration Testing and Vulnerability Assessment with Flipper Zero". In: *Available at SSRN 4658141* (2023).

[3] Roxana Mata-Hernandez et al. "Exploring the Path Loss of a Hacking Tool for Security Matters in the Internet of Things". In: *2023 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC)*. Vol. 7. 2023, pp. 1–6. DOI: 10.1109/ROPEC58757.2023.10409407.

[4] Kolin Nielson and Sayeed Sajal. "The Art of RFID Hacking". In: *2023 Intermountain Engineering, Technology and Computing (IETC)*. 2023, pp. 328–333. DOI: 10.1109/IETC57902.2023.10152251.

Appendix

A GitHub Repository

The full source code and additional resources for this project can be found at the following link: SALAHT4N/Flippercore