VIET NAM NATIONAL UNIVERSITY HO CHI MINH CITY
HO CHI MINH UNIVERSITY OF TECHNOLOGY
FACULTY OF APPLIED SCIENCE

----------***----------

EMBEDDED SYSTEM (EE3427)

SMART-KEY SYSTEM FOR MOTORBIKE

(mini-project)

Supervisor: Bùi Quốc Bảo

Group: 3

Semester: 242

No.	Name	Student ID
1	Nguyễn Cường Quốc	2251045
2	Lương Thành Vỹ	2151280
3	Phạm Tấn Quang	2151248

TABLE OF CONTENT

I. Introduction

Motorbikes are the primary mean of transportation in Vietnam, with millions on the road and the number continuing to rise each year. They are a crucial part of daily life in Vietnam, offering affordability and flexibility in the country's densely populated cities. However, most affordable motorbikes still operate primarily using mechanical systems, with minimal electronic features, which stypically limited to basic lighting or ignition circuitry. As a result, these vehicles lack the embedded systems that could significantly improve user safety, convenience, and overall experience.

This presents an ideal opportunity for embedded systems to make a real impact. For our project, we identified a particularly common and frustrating issue: the difficulty of locating a motorbike in large, crowded parking lots. This is not a rare scenario. In many schools, workplaces, and public venues, parking lots are packed tightly with motorbikes, often of the same brand and color. During the day, parking staff may move bikes to make room for others, further complicating their original positions. After long hours of study or work, it's easy to forget exactly where the bike was parked. Without any location markers or terminal labels, finding a specific vehicle becomes a tedious, time-consuming task, especially when dozens of bikes look nearly identical.

Our team has personally experienced this situation on many occasions, and it served as a strong motivation for the development of this project to tackle this issue in our life. We aim to design a smart-key system that enables users to locate their motorbike quickly and conveniently in such crowded environments. Beyond this core functionality, we also plan to integrate additional features, such as electronic authentication and remote locking/unlocking, to enhance overall user security and experience.

The proposed system consists of two main hardware components: a transmitter circuit integrated into the motorbike and a compact, low-power receiver (key fob) carried by the user. A key challenge of this project does not lie primarily in the complexity of the software algorithms, but rather in optimizing power consumption across both hardware modules. If the circuit integrated into the motorbike draws excessive power, it risks draining the vehicle's battery over time. Conversely, if the key fob is not energy-efficient, it may require frequent battery replacements, reducing its practicality and convenience for daily use. Additionally, the key fob must be compact, lightweight, and ergonomically designed to ensure it is easy to carry and operate.

Through the development of this system, we aim to deliver a practical and applicable embedded solution for motorbikes, while also gaining hands-on experience in hardware design, software development, and system integration. The end goal is to create a product that is not only technically sound, but also functional, reliable, and user-friendly in real-world scenarios.

II. Smart-key system specifications

1. System Overview

This smart key system is developed to provide enhanced usability and added security for motorbikes, particularly those that do not come equipped with advanced electronic features. The core functionality of the system is to help users remotely locate their motorbike in crowded parking areas by activating either the horn or headlights, making the bike easier to spot among similar-looking vehicles. Since the horn and headlights are high-power devices, the system uses electromechanical relays to switch them safely, as they cannot be directly controlled by a low-power microcontroller.

The system consists of two main parts: a key fob, carried by the user, and a receiver module, installed on the motorcycle. The key fob is a small, battery-powered wireless transmitter. When the user presses a button on it, the device transmits a signal using RF communication. This signal is received by the onboard receiver module, which processes the data and activates the appropriate response—such as flashing the lights or sounding the horn.

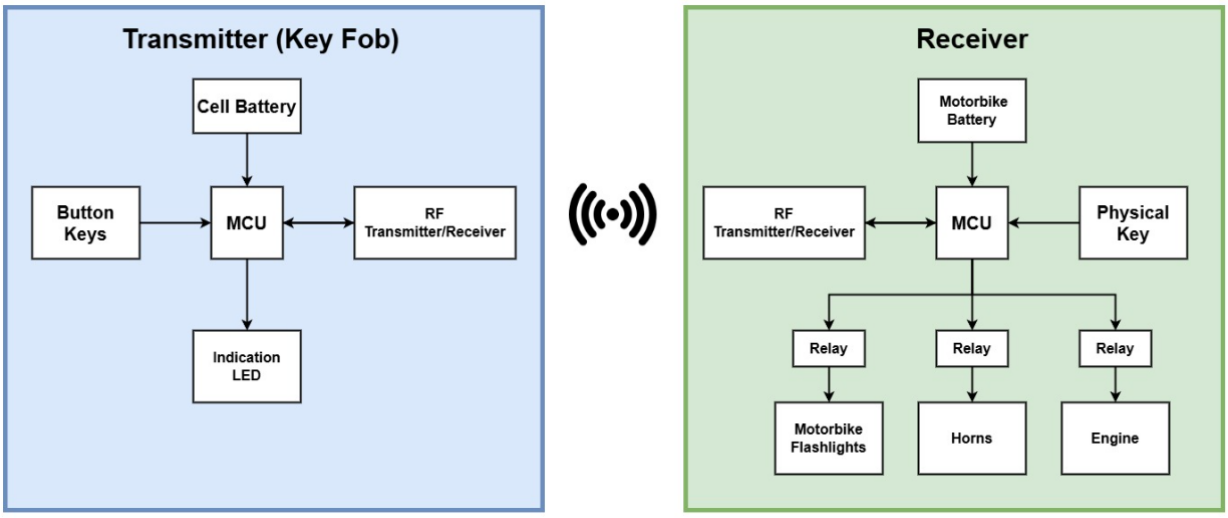

Figure 1: System Architecture Overview

Beyond the basic location feature, the system can be extended to support remote locking and unlocking. This is accomplished by interrupting the power supply from the motorcycle's battery to the engine circuit using a relay. When locked, the power connection to the engine is cut, preventing the vehicle from being started. When unlocked, the circuit is restored, allowing normal operation. Though simple in implementation, this feature adds a layer of protection against unauthorized use.

The entire system is designed with low power consumption, compactness, and ease of use in mind. Special consideration is given to the portability and efficiency of the key fob, which operates on a small coin-cell battery and must remain active for extended periods without frequent replacements. The receiver is built to integrate cleanly with a typical motorcycle electrical system, drawing power from the vehicle's main battery and activating only when necessary. The smart key system offers a cost-effective solution that improves both convenience and security for motorbike users, particularly in densely populated regions like Vietnam where motorbike ownership is widespread.

2. System Requirements and Analysis

The system is made up of two separate circuit boards, one for the Receiver and one for the Transmitter (Key Fob). Each board has its own set of requirements, but one critical goal shared by both is low power consumption to ensure long-term usability. The biggest challenge in this project is building a power-efficient system that can reliably read and transmit signals at any moment. Achieving that requires careful optimization on both the hardware and software side, getting the most out of the components while keeping the energy footprint as low as possible.

a) Hardware requirement for transmitter (Key Fob)

- *Battery Life and Power Consumption Requirement:*

In most smart key designs, the transmitter (key fob) must be compact, lightweight, and portable. This imposes strict constraints on the circuit's size and power consumption. Since it's usually powered by a small 3V coin cell battery which can only supply a limited current (typically just a few milliamps), this means the entire system needs to be highly energy-efficient. Minimizing power draw is essential to extend battery life and avoid frequent replacements

Commercial smart key fobs are usually hardware-optimized with custom low-power circuits, allowing their batteries to last 2–3 years[1]. In our case, we're using pre-built modules designed for general-purpose use, not specifically tailored for low-power applications like a smart key system. As a result, our design is naturally less optimized in terms of power efficiency. Based on typical current consumption and some simple estimations, we've set a realistic target battery life of around 6 months.

Here's the rough math behind it: a typical button cell battery has a capacity of about 200 mAh. The nRF24L01+ draws about 12 mA when transmitting, and combined with the Arduino's overhead, we estimate the system can transmit continuously for at least 8 hours straight before the battery is drained. There's no official data on how often people actually use their key fobs each day, but it's fair to assume average usage is under 2 minutes of active transmission per day (unlocking, locking, or locating the vehicle). With that assumption:

$$8 \text{ hours} = 480 \text{ minutes of usable transmission time}$$

$$480 \text{ minutes} \div 2 \text{ minutes/day} = 240 \text{ days} \approx 8 \text{ months}$$

So realistically, we're looking at somewhere between 6 to 8 months of battery life, which makes our 6-month design goal conservative but reasonable

[1] "When Is It Time to Change Your Key Fob Battery?" *Microbattery*, accessed April 8, 2025, https://www.microbattery.com/blog/post/when-is-it-time-to-change-your-key-fob-battery-2/.

- *Key Fob Size Requirement:*

The key fob casing must be compact enough to comfortably fit in the palm of a hand, with a maximum size of approximately 5 cm x 5 cm x 2 cm. Since the key fob will be used primarily outdoors, it needs to be built for durability and real-world abuse. This includes withstanding physical impacts, such as accidental drops from about 1.5 meters, which is a typical height from hand to ground.

- *Environmental Durability Requirement:*

The fob must be resistant to water and dust exposure, as it may get wet or dirty during regular motorcycle use, especially in rain or on rough roads. As a result, achieving a reliable IP (Ingress Protection) rating is essential, along with ensuring the overall mechanical design can handle daily wear and tear.

- *User Interface Requirements:*

The key fob must includes two buttons, each assigned to a specific function: triggering the motorbike horn, activating the flashlight, and unlocking the vehicle (connecting the battery to the engine). Beside that the Key Fob must have an LED to indicate a button is being press, or system indication such as the bike has been locked or unlocked.

- *Communication Range Requirement:*

For our smart key system, we set a target communication range of 10 meters, based on typical real-world usage. In most cases, a rider will activate the system while approaching the motorcycle or from a nearby pocket. A range longer than 10 meters is unnecessary and could increase the risk of unauthorized access. On the other hand, a shorter range might be inconvenient, especially when locking the motorcycle from a distance. We also factored in the technical limitations of the NRF24L01 module, which, when properly powered and configured, can maintain a stable connection within this range under normal conditions. Overall, 10 meters offers a practical balance between usability, security, and reliable wireless performance.

- *Casing Requirement:*

The casing for the key fob must meet two key requirements. First, it needs to be lightweight yet durable, capable of withstanding impacts from drops or accidental throws. Most commercial key fobs achieve this by using high-toughness plastics such as ABS or polycarbonate, which offer a good balance between strength and weight. Additionally, metal enclosures are not suitable, as they can interfere with RF signals due to the Faraday cage effect, which would severely degrade wireless communication performance.

Second, the casing must allow for easy battery replacement while still maintaining a certain level of protection against dust and moisture, as mentioned in our environmental durability goals. Therefore, the key fob should be constructed using a durable plastic material and designed with a user-accessible battery compartment.

Based on the analysis above, the key fob hardware requirements can be summarized as follows:

Table 1. Hardware Requirements for Key Fob (transmitter) of the smart-key system

Number	Requirement
Hardware requirement	
1	*The key fob shall have a communication range of at least 10 meters for proximity detection.*
2	*The key fob shall use a low-power design to ensure a battery life of at least one year under normal usage.*
3	*The key fob shall be powered by a replaceable 3V battery.*
4	*The power consumption shall be less than 1mW in sleep mode and less than 50mW during active transmission.*
5	*The key fob shall have an IP65 protection rating to ensure resistance against dust and water splashes.*
6	*The key fob should withstand drops from at least 1 meter without loss of functionality.*
7	*The key fob must include a two buttons to send a signal to the motorbike for location identification.*
8	*The key fob should include a LED indicator*
9	*The dimensions of key fob shall be 5cm × 5cm × 2cm*
10	*The key fob casing must be designed to be easily opened to allow for convenient battery replacement.*
11	*The key fob must weigth less than*
12	*The key fob should weigh no more than 50 grams*
13	*The key fob should have an anthentic look*

b) Receiver

- *Power consumption reuirement:*

Unlike the transmitter (key fob), the receiver unit is installed on the motorbike, where it benefits from being directly powered by the motorbike's 12V battery, eliminating the need for compact battery storage. However, low power consumption is still a critical requirement, if the receiver continuously draws too much current, it risks draining the bike's battery, potentially preventing the engine from starting. This becomes a major inconvenience, especially since recharging or replacing a drained motorcycle battery is time-consuming and not always easy on the go.

- *Size requirement:*

Motorbike models vary significantly in terms of available space for modifications, some bikes provide plenty of room for additional electronics, while others are much more constrained. To ensure compatibility across a wide range of motorbikes, we must define a size limitation for the receiver unit. Since the receiver integrates not only the communication module and microcontroller but also circuitry for power regulation and relay-based power control, it requires a moderate amount of space. Taking all these factors into account, the receiver should not exceed 10 cm × 10 cm × 5 cm in size, allowing it to fit within most motorcycle compartments while still accommodating all necessary components.

- *Environmental Durability and Casing Requirement:*

Because motorbikes are often used in outdoor environments, the receiver unit is likely to be exposed to rain, dust, mud, and road debris. To ensure reliable operation in these conditions, the receiver must be rugged and well-protected. This includes using a sealed enclosure, applying conformal coating to the PCB for moisture resistance, and implementing cable glands to prevent water and dirt ingress through wiring entry points. Achieving an IP65 or higher ingress protection rating is recommended to maintain system reliability in harsh environmental conditions.

- *Communication Range Requirement:*

As with the key fob, the receiver is expected to support wireless communication over a distance of up to 10 meters, ensuring consistent performance within the intended operating range.

Table 2.Hardware requirements for receiver of the smart-key system

Number	Requirement
Hardware requirement	
1	*The receiver shall have a communication range of at least 10 meters for proximity detection.*
2	*The receiver shall use a low-power design to ensure a battery life of at least one year under normal usage.*
3	*The key fob shall be powered by a 12V motorbike battery.*
4	*The power consumption of receiver shall be less than 1mW in sleep mode and less than 50mW during active transmission.*
5	*The receiver shall have an IP65 or higher protection rating to ensure resistance against dust and water splashes.*
6	*The dimensions of key fob shall be 10cm × 10cm × 5cm*

c) Software requirement

Table 3. Software requirements for the smart-key system

Number	Requirement
Software requirement	
1	The range for auto lock must be large enough to not trigger durring ussage of vehicles.
2	Lock/unlock communication must include handshake for confirmation.
3	Lock/unlock range must be limited that similar to range of auto lock
4	The smart key system shall use rolling code for lock/unlock operation for security
5	The code for lock/unlock must change every time a lock/unlock command is issued.
6	The change of code must be in sync between keyfob and vehicle to ensure seemless operation.

3. Component Selection

a) Microcontoller and RF module

Since our Microprocessor course has already introduced us to both AVR architecture and Arduino programming, we've chosen an AVR-based chip for our project. Among the various Arduino boards available, the Arduino Pro Mini stands out as one of the most power-efficient options on the market, making it a great fit for low-power embedded applications.

The Arduino Pro Mini is a compact, low-power microcontroller board built around the ATmega328P 8-bit AVR chip. It includes a crystal oscillator for timing, a voltage regulator for stepping down input voltage (if needed), and a reset circuit with a small push-button.

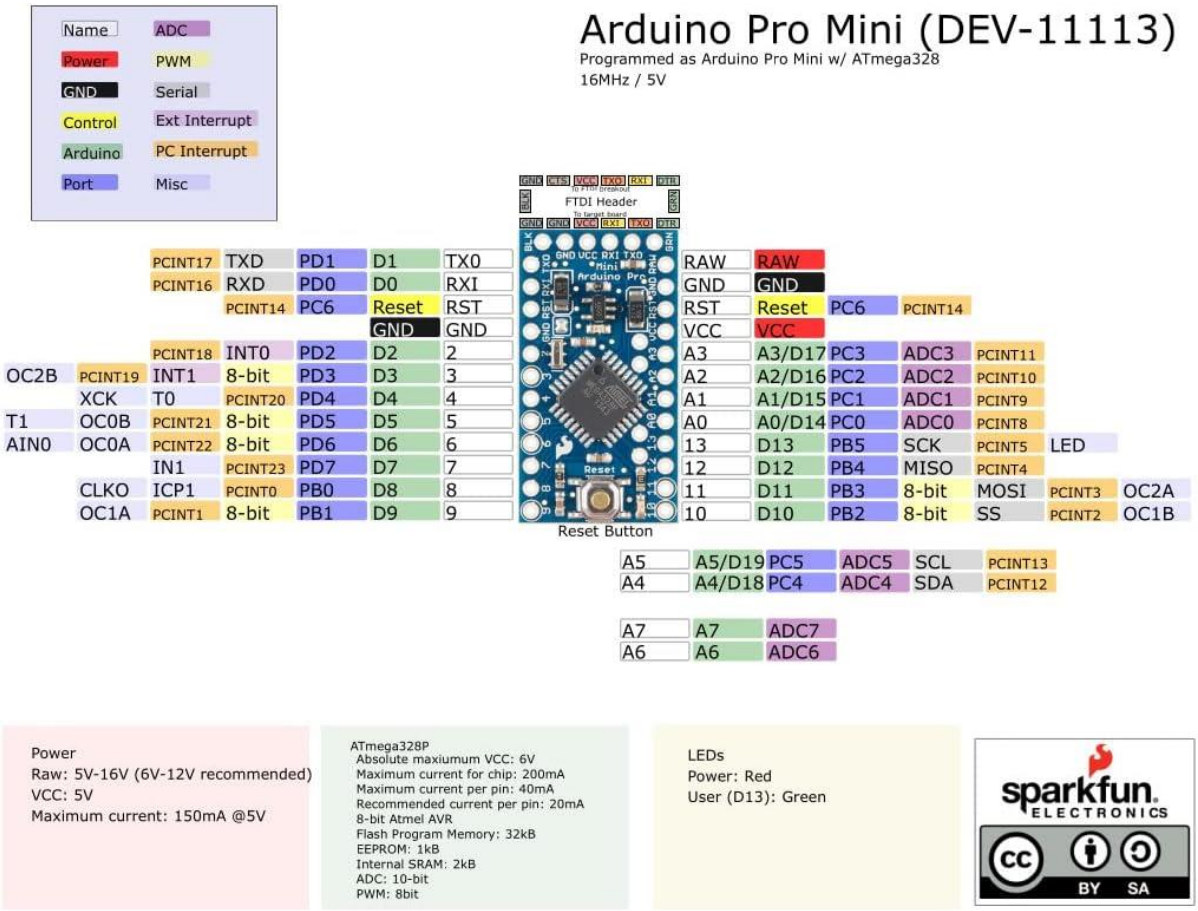

Figure 2: Arduino Pro Mini Pinout Diagram.

Minimal passive components like capacitors and resistors provide stability and support for the core circuit. The board skips fancy extras like USB interfaces or onboard debugging, making it a favorite for low-power and embedded applications. It communicates and is programmed via UART using an external USB-to-serial adapter, and it comes preloaded with a lightweight bootloader like Optiboot. With no unnecessary components drawing power, it's easy to mod for deep sleep and ideal for bare-metal programming or long-lasting battery-powered projects.

This compact board supports all the essential communication protocols we need, like SPI, I2C, and UART, along with built-in ADC and internal EEPROM. These features make it versatile enough to interface with a wide range of modules and handle most embedded application requirements.

b) RF module

There are many RF modules specifically designed for basic switching applications. These modules commonly operate at frequencies such as 433 MHz or 315 MHz and are already optimized for low power consumption and compact size. They are widely used in commercial RF-controlled switch products, but the signals they transmit are limited to simple on/off control and are not suitable for more advanced data communication. In this project, as part of both learning and practical application, we aim to implement a rolling code system for the smart key. This requires a programmable RF module that supports more complex functionality and allows for faster data communication between the transmitter and receiver.

To meet these requirements, we selected the nRF24L01+ module. It is low-cost, widely available, and operates at 2.4 GHz. This module generally have lots of documentations, simple to configure and program, ultra low power consumption, wide operation voltage. Its size is compact enough to be integrated into the key fob without issue. However, because this module is designed for general-purpose wireless communication, it is not specifically optimized for ultra-low power applications like dedicated switching modules.

To address this limitation, we applied several methods to reduce the module's active operation time. This includes implementing sleep modes, scheduled wake-ups, and minimizing unnecessary transmissions to improve power efficiency and extend battery life within our design.

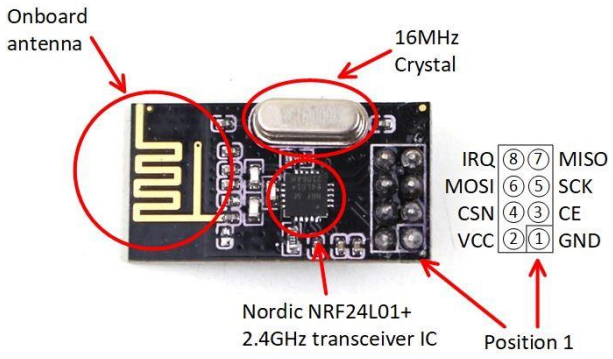

Figure 3: NRF24L01+ 2.4GHz RF Transceiver Module Pinout and Key Components.

c) Battery

In this system, the RF receiver is powered directly by the vehicle's 12V battery, so we only need to carefully select a suitable battery for the RF transmitter located in the key fob.

Most commercial key fobs use button cell batteries as their power source. These batteries typically offer a few hundred milliamp-hours (mAh) of capacity, and they are widely used due to their compact size, low cost, and broad availability. To maintain compatibility with industry norms and make use of easily accessible components, our design also adopts this approach.

However, powering both a general-purpose microcontroller and an RF module from a single button cell presents a major challenge. These components can require short bursts of relatively high current, which is difficult for small batteries to handle efficiently. To help solve this issue, we selected the CR2450 button cell, which provides a 3V output and offers a high capacity of up to 620 mAh, which is one of the highest available in the markety for this battery class.

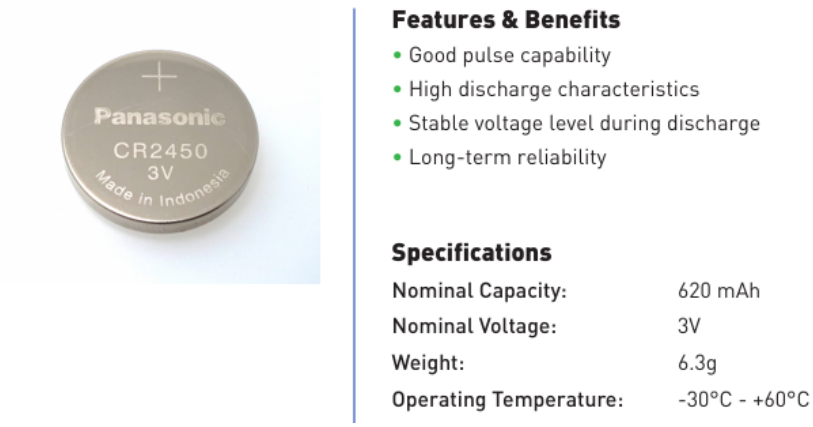

Figure 4: Panasonic CR2450 Coin Cell Battery – Features and Specifications.

Another important benefit of the CR2450, as shown in the internal resistance graph, is its relatively low internal resistance during most of its discharge cycle. This is a critical advantage for our project. Low internal resistance allows the battery to deliver higher current pulses without a significant drop in output voltage. For an application like ours, where the key fob periodically transmits data using the nRF24L01+, this means more stable performance, fewer transmission failures, and longer functional life before the battery becomes unusable under load, even if some capacity remains.

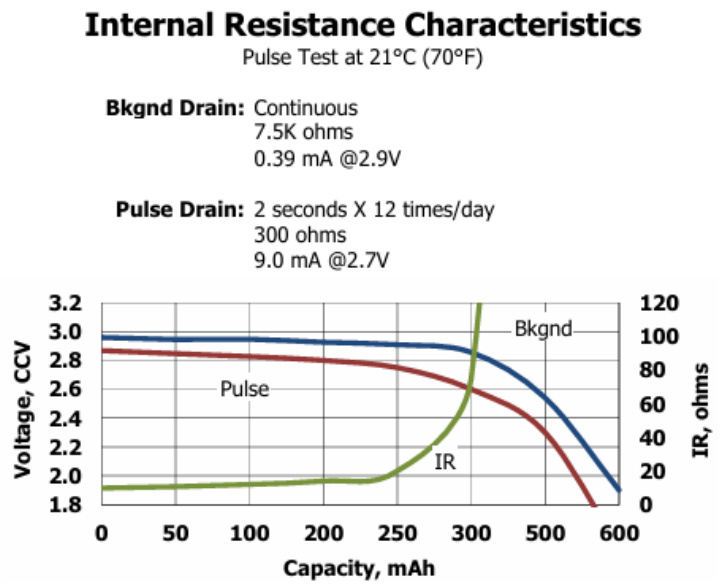

Figure 5: Internal Resistance and Discharge Characteristics of the Panasonic CR2450 Battery.

Therefore, the CR2450 is a well-suited choice for our key fob because it combines high capacity, wide availability, compact size, and good voltage stability under pulsed loads, which directly benefits the performance and reliability of our design.

d) DC-DC Converter for Powering the Receiver from Motorcycle Battery

In the receiver circuit, we require a 3.3V power supply to operate both the microcontroller and the RF module. Fortunately, the Arduino Pro Mini board we're using includes an onboard 5V to 3.3V low dropout (LDO) regulator. This regulator is powered through the RAW input pin, which supplies the input voltage to the onboard power regulation circuit, as illustrated in the figure below:

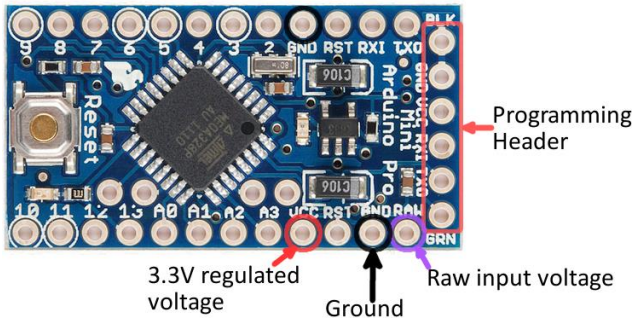

Figure 6. Three different power-related pins: GND, VCC, and RAW pins

To supply the necessary 3.3V for the receiver, we need to step down the 12V from the motorcycle battery. However, finding compact, efficient regulators that directly convert 12V to 3.3V is difficult, as most standard solutions are optimized for 5V or higher intermediate steps. To simplify the design, we took advantage of the onboard 5V-to-3.3V LDO regulator integrated into

the Arduino Pro Mini. This allowed us to reframe the problem as a 12V-to-5V conversion instead. For this purpose, we used a Hi-Link HLK-B_S-1WR3 series isolated DC-DC converter, which reliably steps down the 12V battery voltage to 5V. The resulting 5V is then fed into the Pro Mini's RAW input, which the onboard LDO regulates down to 3.3V for the microcontroller and RF module.

Figure 7. B_S-1WR3 DC-DC converter, a small size and high-efficiency DC/DC power supply module

This module offers built-in short-circuit protection and delivers low output ripple and noise within a 20 MHz bandwidth, ensuring clean and stable power for sensitive components. It achieves efficiency levels of up to 88%, making it highly suitable for low-power embedded systems. Most importantly, the module provides a constant output voltage, input-output isolation, and a single unregulated DC-DC output, these features are particularly useful for low-frequency analog circuits, relay drivers, and data interface applications where electrical isolation is beneficial.

The main reason we chose this converter is its compact form factor. It is small, efficient, and easy to integrate, which simplifies both the PCB layout and the overall hardware design. In contrast, building a discrete DC-DC converter from individual components would have significantly increased the PCB size and complexity, as well as introduced more opportunities for error during prototyping and fabrication, which lead to potentially costly repairs and rework.

III. Firmware design

a) Transmitter (Key Fob)

The key fob in our system has a simple function, it continuously monitors the state of two buttons and sends data to the receiver when a press is detected. However, the main design challenge is to achieve this continuous button monitoring while consuming as little power as possible. Since the key fob is powered by a small coin cell battery and is expected to last for many months, power efficiency is critical. To address this, we applied several key techniques in firmware:

- *Disable all the peripherals that we don't use or not yet to use*

When the microcontroller is powered on, all of its internal peripherals, such as the ADC, SPI, TWI, timers, and serial interfaces, are enabled by default. Even if these peripherals are not used in the program, they still draw current in the background. To avoid this unnecessary power drain, we disable them explicitly using the Power Reduction Register (PRR). This ensures that only the essential modules remain active, reducing power consumption by cutting off clock signals to unused subsystems.

- *Disable all the GPIO ports*

GPIO pins can also contribute to unwanted power loss if not configured properly. Even if we do not interact with a pin in software, the physical I/O circuitry remains powered. Specifically, if a GPIO pin is configured as an input and left floating , it may oscillate due to ambient electrical noise. This causes the internal input buffer to toggle rapidly, which can lead to current consumption in the range of hundreds of microamps. Additionally, enabling internal pull-up resistors on unused input pins can create a continuous current path to ground, leading to additional power loss.

To avoid this, we configure all unused GPIO pins as outputs and drive them low. This ensures the pins are in a known state and immune to external noise or leakage paths, further minimizing power consumption.

- *Sleep the CPU of the AVR and wake it up periodically to read the button*

Although disabling unused peripherals and GPIOs helps reduce power consumption, the microcontroller itself remains the primary source of current draw. In active mode, the CPU can consume several milliamps, which would quickly drain a small battery. To address this, we utilize the AVR's built-in sleep modes. These modes allow the CPU to halt execution and power down internal components, while still enabling selected interrupts such as those from the watchdog timer or pin changes and wake the system when needed.

In our implementation, we use the Power-down sleep mode, which is the most energy-efficient state available. In this mode, the microcontroller draws less than 1 μA of current. A watchdog timer is configured to periodically wake the CPU at fixed intervals. Each time the system wakes up, it briefly re-enables the necessary components to check the state of the buttons. If a button press is

detected, it transmits a signal to the receiver. Once the task is complete, the system disables everything again and returns to sleep. This approach keeps the key fob in an ultra-low-power state for the majority of the time, waking only for brief moments to perform essential operations. As a result, average current consumption remains extremely low, often under 1 µA that greatly extends battery life without sacrificing responsiveness.

The nRF24L01+ module in our design also support a built-in power-down mode, which we take full advantage of. It is activated only when data needs to be transmitted and is immediately powered down once the transmission is complete. Therefore, the RF module only contributes minimal power consumption for the majority of the time while the AVR microcontroller is in sleep mode.

The following flowchart illustrates the main steps executed by the key fob firmware:

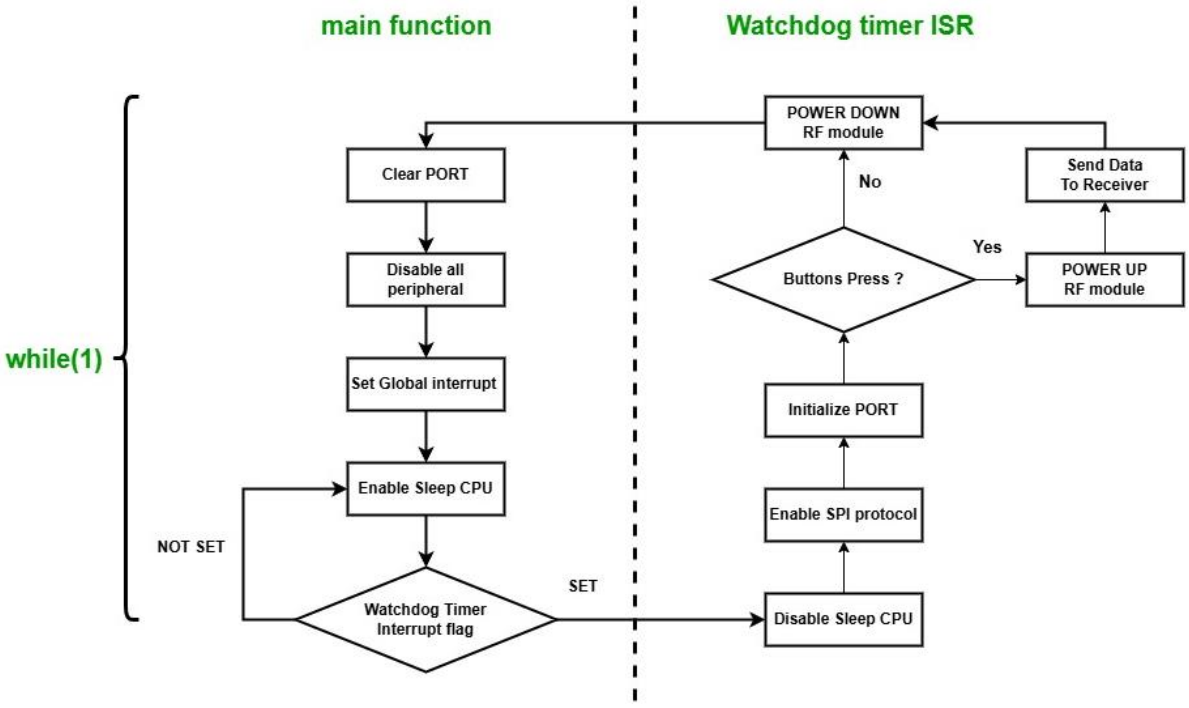

Figure 8. Table 3. Flowchart of the firmware for Key Fob

When the microcontroller powers on, the firmware begins by configuring the Watchdog Timer to generate an interrupt every 0.5 seconds. This timer serves as the system's periodic wake-up source. The watchdog interrupt is enabled, but the global interrupt flag is not set yet to ensure no interrupts are triggered during the initial setup. We also configure the AVR to use the POWER_DOWN sleep mode, which is the lowest power state available. In this mode, the CPU halts execution, all I/O clocks are disabled, and only specific interrupts can wake the device (watchdog timer, external interrupts).

After this setup, the firmware enters an infinite while(1) loop, which serves as the main execution cycle. Inside the loop, the program first disables all unused peripherals and clears the

I/O ports to save power, even those ports that will be used later are disabled temporarily. Once everything is shut down, the firmware enables sleep mode, reenables global interrupts, and puts the CPU to sleep. From this point on, the CPU remains inactive and consumes minimal power until it is awakened by the next watchdog timer interrupt.

When the Watchdog Timer interrupt occurs, the Interrupt Service Routine (ISR) is executed. Inside the ISR, the system powers up the SPI peripheral and reinitializes the necessary GPIO pins to allow communication with the RF module and to read the button states. If a button press is detected, the RF module is powered on, the signal is transmitted to the receiver, and then the RF module is powered down again to conserve energy. After completing this operation, the ISR disables the sleep flag, allowing the main loop to reinitialize the shutdown process and return the system to sleep. This cycle continues, enabling the key fob to remain in an ultra-low-power state while still periodically checking for user input.

The schamtic of the Key Fob:

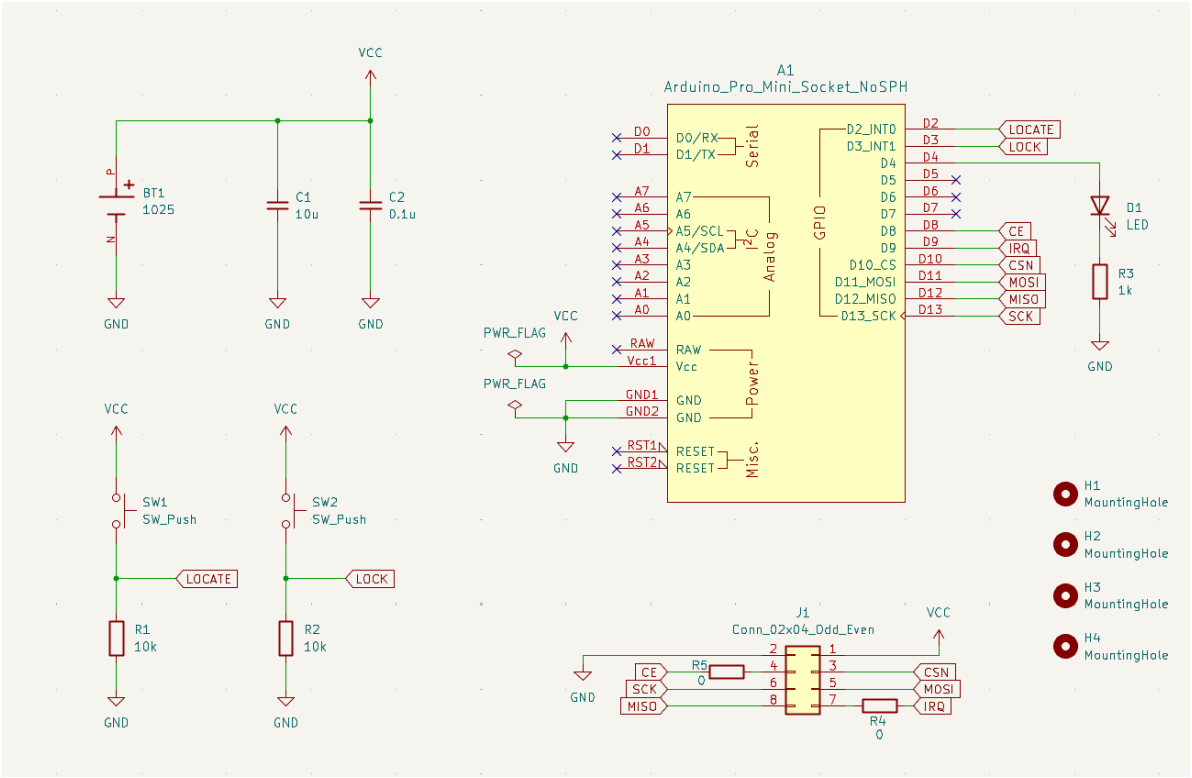

Figure 9. Schematic of the Key Fob

b) Receiver

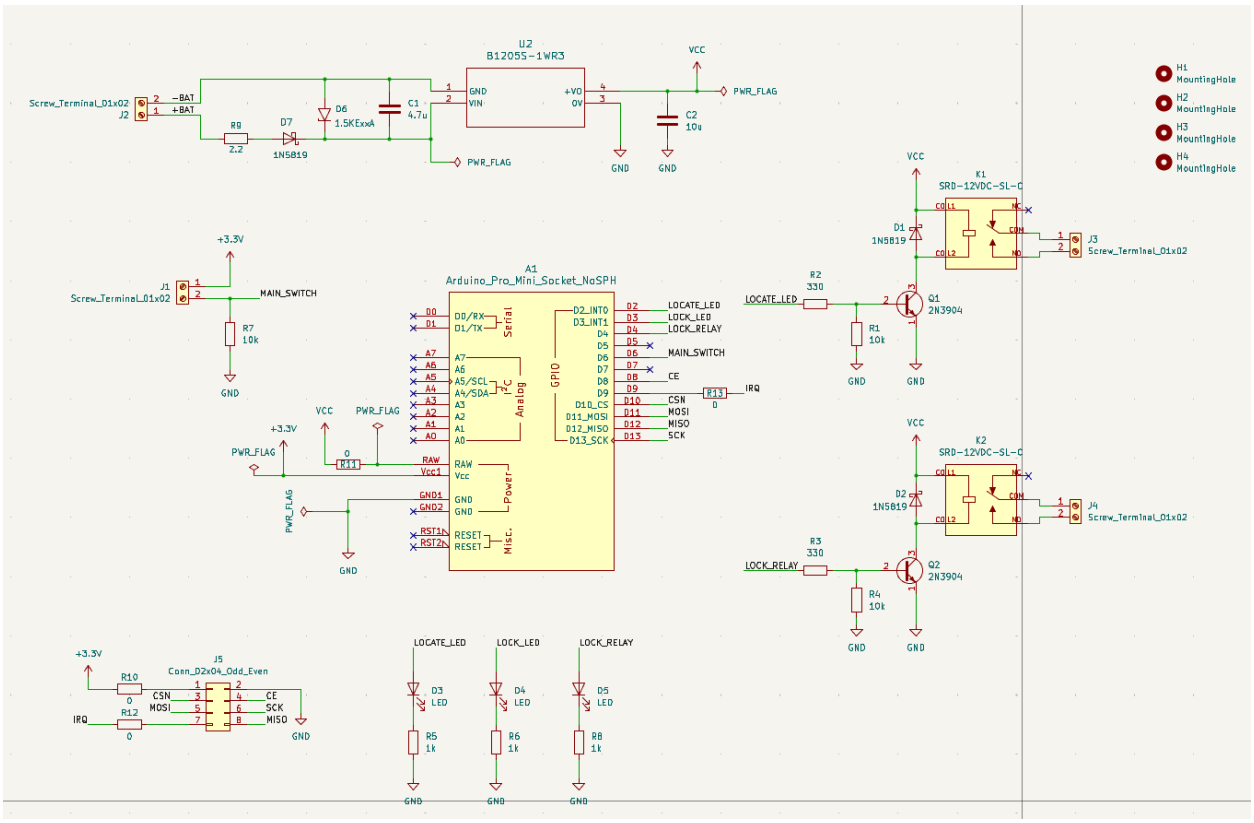

Figure 10. Schematic of the Receiver

IV. Implementation Details

1. PCB Fabrication

Although we were recommended to use Altium Designer for creating the schematic and PCB layout, our team decided to also use KiCad because one of our members already had experience with it. While we did implement the schematic and layout in Altium as required, we also recreated the design in KiCad to streamline the workflow and make use of the tools we're most familiar with.

When it comes to fabricating the PCB, the typical approach is to send the design files to a third-party manufacturing service. However, this method has two major drawbacks: it can be expensive, and it often takes 1–2 weeks to receive the finished board. Additionally, if there's any mistake in the design, we won't know until the boards arrive—wasting both time and money.

Due to budget constraints and the need for quicker feedback, we chose a manual and low-cost PCB fabrication method known as the toner transfer method. This approach allows us to produce the PCB ourselves using basic tools and materials. The process works as follows:

First, we design the PCB layout in KiCad using the correct footprints for the components. It's crucial that the footprint dimensions and pin spacing match the actual components so everything fits properly during assembly. Below is the layout we created for both the key fob (transmitter) and receiver using this method:

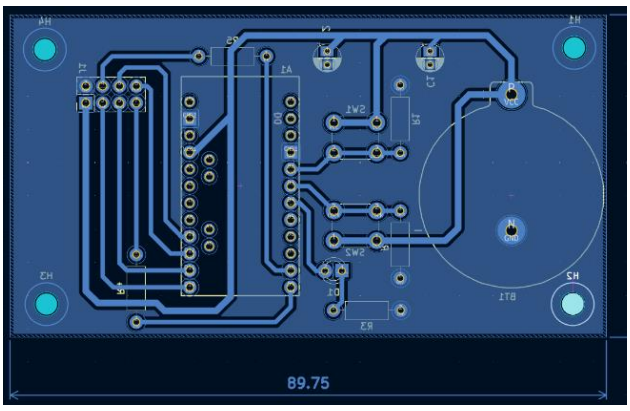

Figure 11. PCB layout of Key Fob on KiCad to print out on toner-transfer paper

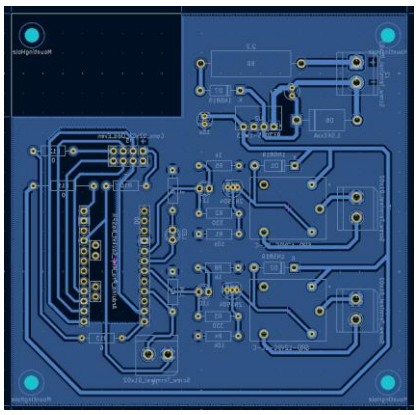

Figure 12. PCB layout of Receiver on KiCad to print out on toner-transfer paper

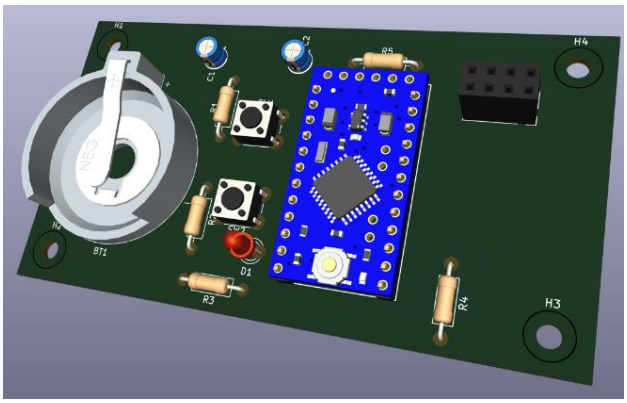

Figure 13. PCB layout of Key Fob on KiCad in 3D

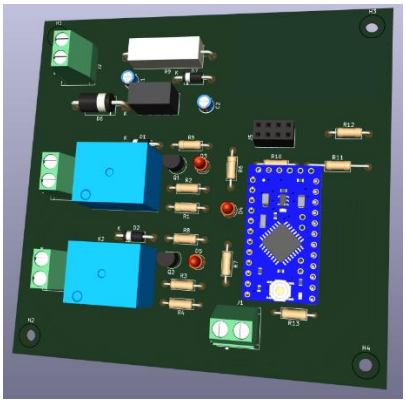

Figure 14. PCB layout of Key Fob on KiCad in 3D

Next, we print the layout onto glossy or toner-transfer paper using a laser printer. This type of paper doesn't absorb the toner ink, allowing it to stay on the surface. Then, we place the printed layout face-down on a copper-clad board (a piece of plastic or fiberglass coated with a thin layer of copper). Using a hot iron or laminator, we apply heat and pressure, which melts the toner and transfers it onto the copper surface.

Figure 15. Copper-clad board after toner transfer for the Key Fob PCB

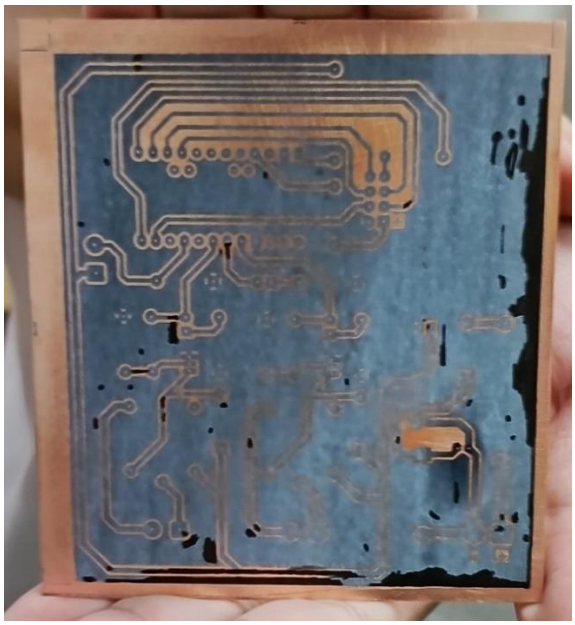

Figure 16. Copper-clad board after toner transfer for the Receiver PCB

Once the toner has adhered, we remove the paper, leaving behind a printed version of our PCB traces on the copper. The toner acts as a resist, protecting the copper underneath it. We then immerse the board in a chemical etchant (like ferric chloride or sodium persulfate), which dissolves the uncovered copper, leaving only the copper traces we designed.

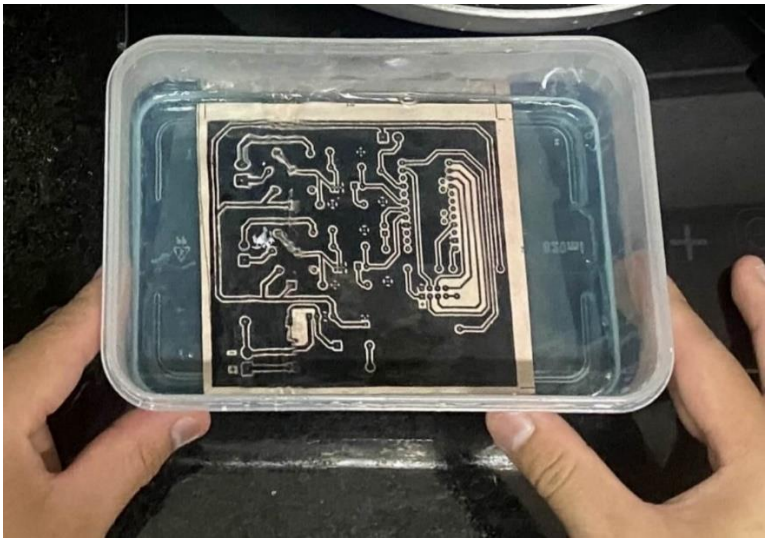

Figure 17. Wet Etching the PCB to remove the uncovered copper

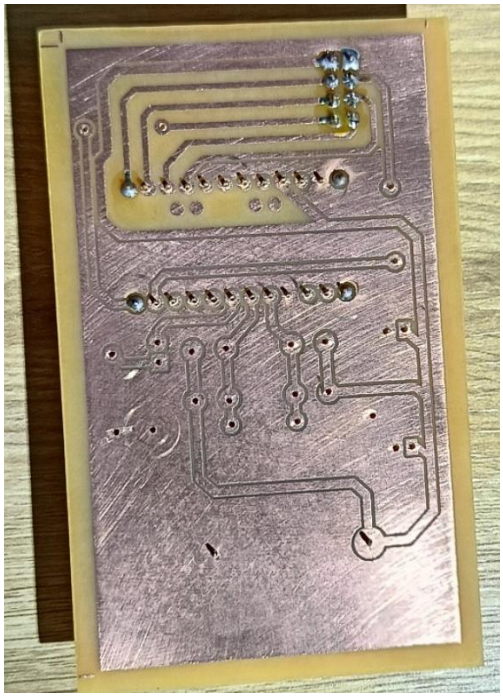

Figure 18. Final PCB of the Key Fob after Wet Etching Process

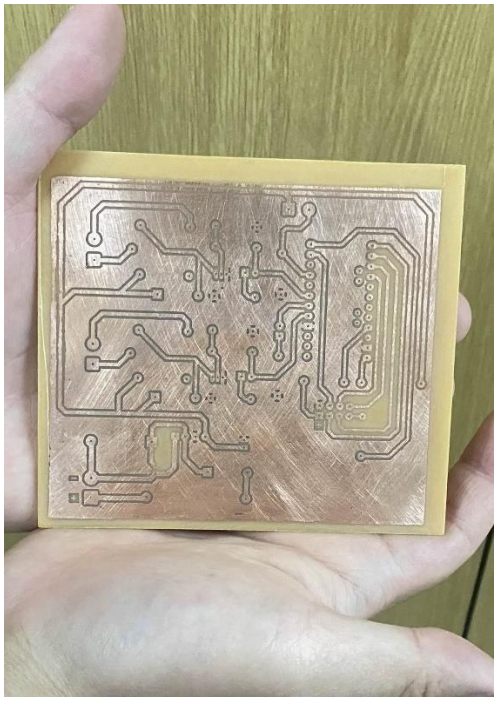

Figure 19. Final PCB of the Receiver after Wet Etching Process

The result of this process is a functional PCB that closely follows our intended layout and is ready for drilling and assembly. One of the key advantages of using the toner transfer method is that it gives us full control over the prototyping process. Unlike sending the design to a third-party PCB manufacturer which can be expensive and time-consuming, we're able to iterate and test much more quickly.

2. Casing

One of the design requirements for both the Key Fob and the receiver was to include a protective enclosure with an IP65 or higher rating to ensure resistance to dust and water. However, achieving this level of protection requires specialized tools, materials, and manufacturing techniques such as precision sealing, gaskets, and enclosure-grade plastics, which were beyond our budget, resources and skills for this project.

As a practical alternative, we built temporary protective box casings using Plexiglas, which serve as enclosures primarily for prototyping and demonstration purposes. While these enclosures do not meet the full IP65 standard, they provide a basic level of physical protection for the circuit during testing and presentation. The final prototypes, including the custom casings, are shown in the results section later in this report.

3. Problems during Fabrication process

Although this method involves more manual effort compared to professional fabrication, it significantly reduces cost and shortens the development cycle. Throughout the process, we frequently adjusted the layout and schematic as we discovered practical issues, for example, some component pads were too close together to solder properly, or certain parts were physically too large to fit as expected because we only accounted for the pin layout, not the full body size. These real-world adjustments were only apparent once we began building the board by hand.

This trial-and-error process naturally came with some mistakes, but it also gave us valuable experience in designing and producing our own PCBs. Unlike outsourced manufacturing, which might take a week or more and only reveals design problems after the board arrives, this hands-on method gave us immediate feedback, allowing us to fix problems and test improvements on the spot, with virtually no additional cost. It proved to be a practical and educational solution for our project’s budget and timeline constraints.

One of the most frustrating parts of the project was the soldering process during PCB assembly. In our initial designs, we placed many of the component pads too close together, especially for components with fine pitch. This led to frequent solder bridges forming between adjacent pins during soldering. These shorts were difficult to detect visually and didn't show up in the schematic or layout checks, as KiCad’s Design Rule Check (DRC) only verifies spacing in the digital layout but not what happens during real-world soldering. As a result, we spent several hours debugging faulty boards, even though everything looked correct in the software.

We also encountered issues related to residual toner and incomplete etching. In some cases, leftover toner or small bits of copper remained between traces that should have been isolated, especially in areas with fine gaps. These unintentional copper connections weren’t visible at a glance but caused subtle short circuits that made the circuit malfunction. It wasn’t until several

failed boards that we realized we could have avoided much of this by using an ohmmeter to check for unexpected continuity between traces before soldering any components. This hands-on experience taught us that validating the physical board (with an Ohm-meter) is just as important as verifying the design file, it's something we wouldn't have fully appreciated through software simulation alone.

We also made a critical layout mistake that took us multiple boards and days of troubleshooting to uncover. Specifically, the GND wire to the Arduino Pro Mini was routed using an extremely thin 0.3 mm trace, with multiple sharp corners and a long, winding path. At first, we didn't suspect this trace, as the connection seemed electrically valid in the schematic. However, in practice, the narrow, high-resistance ground path caused power delivery issues and erratic behavior in the circuit. This wasn't something that failed in simulation or layout review and it only became obvious after several unsuccessful attempts and hardware tests.

From this, we learned an essential lesson in PCB design: proper trace width, spacing, and routing are critical, not just for signal lines, but especially for power delivery. Ground traces, in particular, should be short, wide, and direct, avoiding unnecessary corners or long circular paths. These physical considerations are often overlooked in early designs but are crucial for both electrical stability and ease of assembly.

V. Result

1. Final PCB result

Our final PCBs are cased with shield of Plexiglas to support a certain level of dust and impact resitance and tolerance, all the PCBs and cases are completely homemade and used afforable and easy to find materials and tools. The following is our final result of our PCBs for both Key Fob and transmitter:

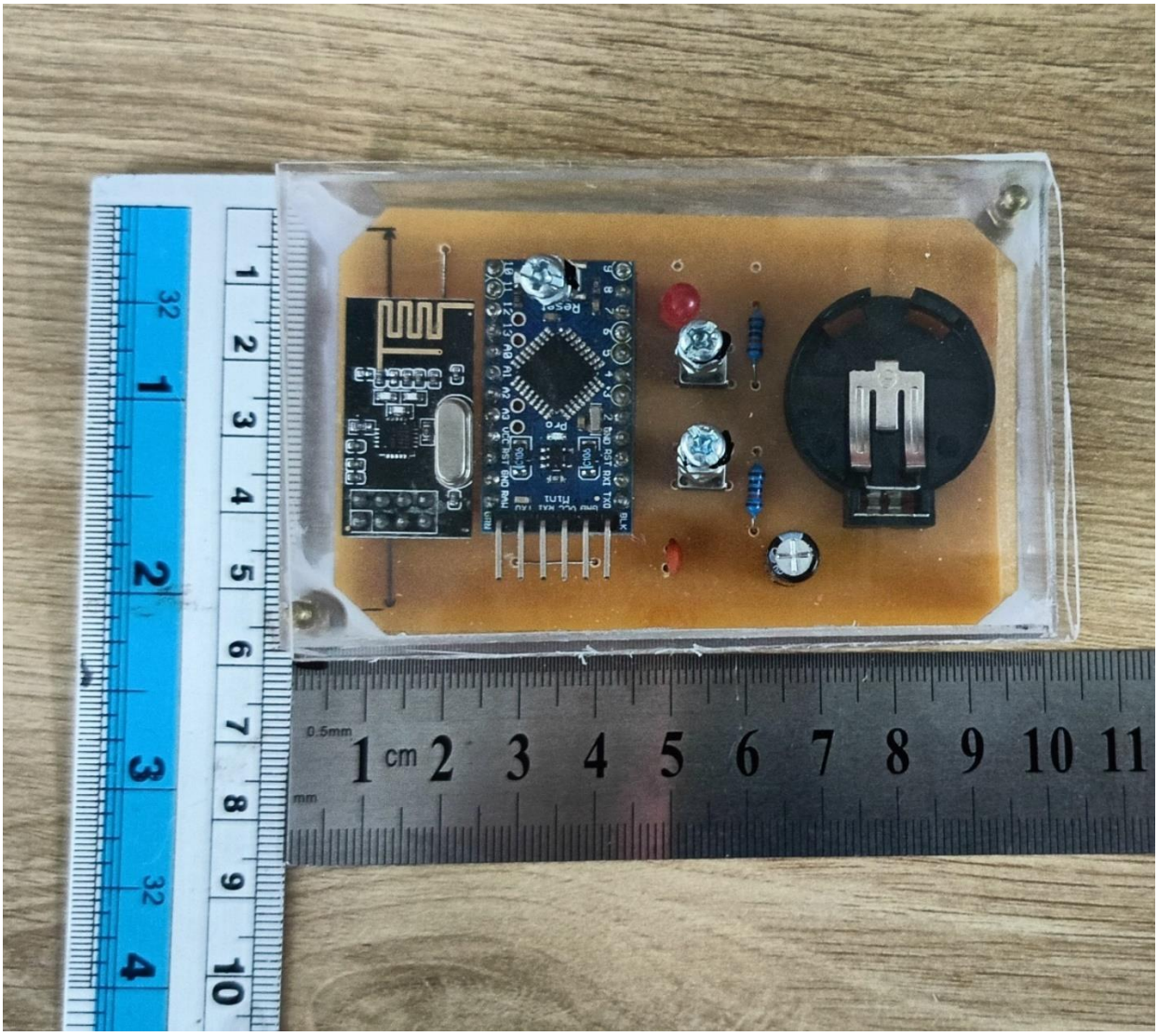

Figure 20. Final result of the PCB for the Key Fob

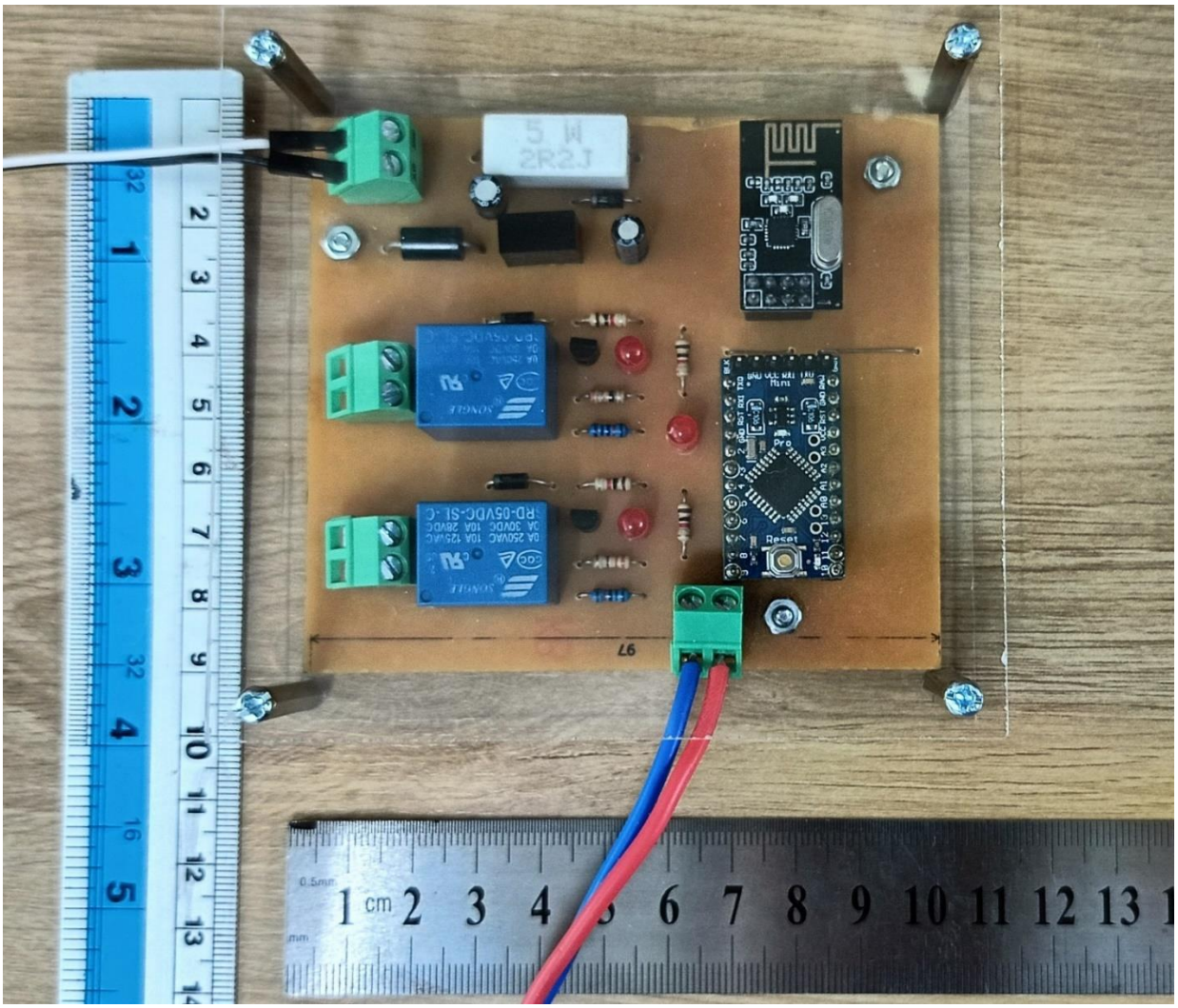

Figure 21. Final result of the PCB for the Receiver

The final dimensions of the receiver circuit are approximately 9 cm × 10 cm × 3 cm, while the Key Fob measures around 6 cm × 10 cm × 2 cm. Based on these results, the receiver meets our design constraint, which required the total size to stay within 10 cm × 10 cm × 5 cm. However, the Key Fob exceeds its intended size specification of 5 cm × 5 cm × 2 cm, making it too large to comfortably carry in a pocket or attach to a keychain, and therefore not ideal for its intended use.

As previously discussed, the oversize and enclosure limitations were primarily due to budget constraints and a lack of access to specialized tools, materials, and manufacturing techniques. As a result, we built the casings using basic household tools and low-cost prototyping materials, which, while functional for demonstration purposes, do not meet the original Environmental Durability requirements such as water and dust resistance. Additionally, the final appearance of the casings may not be visually polished or production-ready. Despite these limitations, the prototypes successfully demonstrate the core functionality of the system.

2. Improvement

Our PCB and layout are one-sided, which results in a large PCB, however the size of our design can be significantly reduced if we fabricate the PCB ultilizing both side of the PCB. For example, our Key Fob can easily matched the size reuirement if we utilize the other size of the PCB and move the battery holder to the back.

Related to the PCB, our wiring and path are surely not optimized in length and spacing. Due to lack of experience on PCB layouting, our layout mostly done manually without any automated tools or features and we violate certain rules and standards in PCB layout design and some other electrical hazards like high-resistance from thin traces, or delay of signals that caused by capacitance and inductance from long, corner and circular path traces. The result of this has caused our many attempts and failed PCB and time of debugging to found out this mistakes.

VI. Conclusion

Our goal was to design an affordable smart-key system for motorbikes, targeting the large number of budget-friendly motorcycles commonly used in Vietnam. The system is intended as a practical add-on to enhance both usability and security. One key feature is the ability to visually locate a motorcycle in crowded parking areas where bikes often look identical by triggering a visual indicator like flashing lights. Additionally, the system provides an extra layer of protection by enabling remote locking and unlocking, effectively disconnecting the engine's power supply when needed. The system operates using RF communication between a compact Key Fob and a receiver module installed on the motorbike. Its core functionality is based on two main control features: activating the horn and lights as visual/audible indicators, and controlling the internal power connection between the motorcycle's battery and the engine to enable or disable ignition. When activated, the Key Fob sends signals to the receiver, which controls two electrical switches: one for activating a visual indicator (e.g., flashing the headlights or signal lights), and another for interrupting power to the engine, serving as a basic immobilizer.

During the project, the most significant challenges we faced were related to prototyping and PCB fabrication. All of our PCBs were entirely homemade using the toner transfer method, as we chose not to rely on third-party fabrication services due to financial limitations. This decision allowed us to maintain full control over the process but also introduced many practical difficulties, especially in achieving precise layouts and reliable etching results. Consequently, while the functional circuits worked, the final PCBs did not fully meet the physical design specification requirements, particularly in terms of size, durability, and enclosure integration.

On the software side, we were more successful. Our firmware achieved most of the intended functionality, including low-power operation using sleep mode, RF transmission, and system wake-up via interrupts. However, due to time and resource constraints, we were unable to implement certain features such as secure pairing, power loss recovery, or improved code structure. Additionally, our source code lacks full standardization and documentation, which we acknowledge as an area for future improvement.

Despite the limitations, our system successfully demonstrates the core functionality of a smart key solution, and more importantly, we gained valuable hands-on experience in embedded system design, low-power firmware development, and DIY PCB fabrication. From learning how to troubleshoot layout errors to understanding the impact of real-world constraints on theoretical designs, this project gave us a deeper appreciation for the complexity of bringing an embedded product from concept to working prototype.

REFERNECE

[1] V. M. Kulasekara, I. Kavalchuk, and A. Smith, Smart Key System Design for Electric Bike for Vietnam Environment, in Proc. 2019 Int. Conf. on System Science and Engineering (ICSSE), Ho Chi Minh City, Vietnam, Jul. 2019, doi: 10.1109/ICSSE.2019.8823367.

[2] J. Blom, "Using the Arduino Pro Mini 3.3V," SparkFun Electronics, [Online]. Available: https://learn.sparkfun.com/tutorials/using-the-arduino-pro-mini-33v/all.

[3] Microchip Technology Inc., ATmega328P – 8-bit AVR Microcontroller with 32KB Flash, Atmel-7810D–AVR–01/15, Jan. 2015. [Online]. Available: https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf