This project demonstrates creative problem-solving techniques and a thorough investigation into the fields of hardware integration and system architecture. The project highlights the value of thorough study, cooperative effort, and flexible thinking by overcoming difficulties in integrating ultrasonic drivers, flame sensors, and Arduino-Tiva communication.

Final Project Report

Li-Fi Project Section1/SubsectionA/Team4

Submitted to:

Dr. Sherif Hammad

Eng: Dina zakaria

Eng: Mohamed Tarek

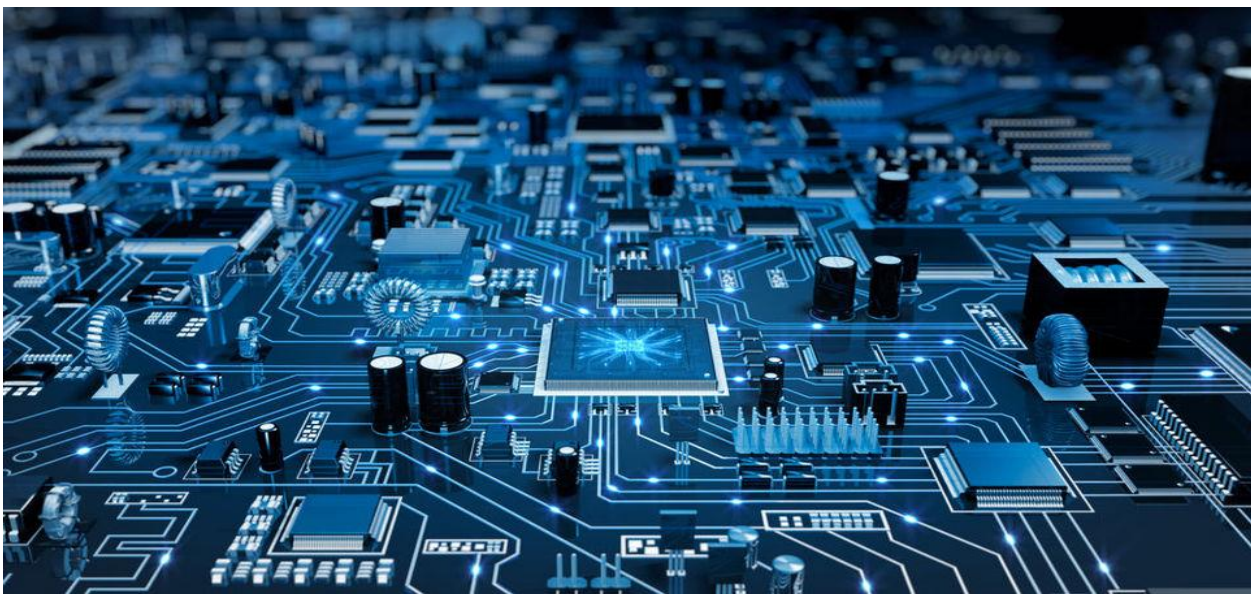
Eng. Hesham Salah

Eng. Mina

Mohamad Waleed

AIN SHAMS UNIVERSITY
FACULTY OF ENGINEERING
International Credit Hours Engineering Program (I-CI

Introduction to Embedded Systems
CSE211
Major Task



Names	Id
Ezz Eldeen Ayman	20P1165
Hossam Eldeen Torky	20P8211
Mohamed Waleed Nassar	20P1297
Youssof Tamer Aly	20P2497
Eyad Khaled Mohamed	20P1418

Contribution Table:

Name	Tiva- C				Arduino Coding	Hardware Implementation	Project Design (CAD)	Report	Logistics
	Flame & ADC	Interrupts	UltraSonic & Timers	Integration					
Mohamad Waleed									
Hossam Torky									
EzzEldeen Ayman									
Youssof Tamer									
Eyad Khaled									

Color Key				
	3	2	1	0

Table of Contents

List Of Figures

1.0 Introduction

In an era where wireless communication plays a pivotal role in connecting our devices, the Li-Fi Hazard Detection System emerges as an innovative solution that leverages the speed and security of light for data transmission. This project explores the integration of Li-Fi technology into a hazard detection system, utilizing the power of light to transmit crucial information between devices.

The Li-Fi Hazard Detection System is designed to enhance home safety by seamlessly integrating a **sender** platform, powered by the ARM Cortex-M4-based TIVA-C microcontroller, and a **receiver** platform, utilizing the Arduino microcontroller. The sender platform is equipped with three key sensors—Fume, Ultrasonic, and Magnetic each responsible for detecting potential hazards within the environment.

Kindly find the link for all of the Project Details Attached below:

2.0 **Project** Overview

2.1 Objective

The primary objectives of the Li-Fi Hazard Detection System are as follows

- Develop a hazard detection system utilizing Li-Fi communication for secure and high-speed data transmission.

- Employ **TIVA-C** as the sender platform with sensors (Fume, Ultrasonic, Magnetic) and **Arduino** as the receiver platform.

- Implement a robust alarm system comprising flashing lamps, buzzers, and LCD displays upon detecting hazards.

- Incorporate user-friendly features such as start/stop operations and a mute button for alarm silencing.

In summary, the project objectives encompass a holistic approach to hazard detection, utilizing advanced technologies such as Li-Fi, integrating multiple sensors, and incorporating user-friendly features for enhanced control and responsiveness. The recognition of Li-Fi's inherent advantages further underscores the project's commitment to leveraging cutting-edge solutions for home safety.

2.2 Key Features

- **Li-Fi Communication**: The project employs Li-Fi communication to transmit data at remarkable speeds of up to 100 Gbit/sec over the visible light spectrum. This ensures secure and interference-free communication, making it an ideal choice for applications such as hazard detection.

- **Secure Signal Transmission**: By utilizing light as the medium for data transmission, the system mitigates the risk of remote signal interception. Light's inability to penetrate walls enhances the security of the transmitted data, ensuring that signals remain confined to the intended space.

- **Multifaceted Alarming System**: Upon detecting hazards, the system activates a comprehensive alarming mechanism. This includes the illumination of a flashing lamp, the activation of a buzzer, and the display of a descriptive message on an LCD screen. Additionally, the system offers the flexibility to communicate hazard alerts to a mobile device via Bluetooth.

- **User-Controlled Operations**: Two pushbuttons facilitate user control, allowing for the initiation and termination of system operations. In the event of an alarm, a mute pushbutton provides a quick means to silence the alarms while keeping the system active.

3.0 System Architecture

3.1 Components Summed up

3.1.1 Sender Platform Components

1) **TIVA-C TM4C123GH6PM microcontroller**
2) **Flame Sensor**
3) **Ultrasonic Sensor**
4) **Magnetic Sensor**
5) **Laser emitter module**
6) **Start/Stop pushbutton**
7) **Bluetooth module**

3.1.2 Receiver Platform Components

1) **Arduino microcontroller (Arduino Uno)**
2) **LDR (Light Dependent Resistor) module**
3) **LED**
4) **Buzzer**
5) **LCD**
6) **Mute pushbutton**

3.2 Sender Platform (TIVA-C)

1) TIVA-C TM4C123GH6PM microcontroller

- The TIVA-C series microcontrollers, developed by Texas Instruments, are part of the broader ARM Cortex-M family, specifically based on the ARM Cortex-M4 core. These microcontrollers are designed for embedded applications that require a balance between high performance and low power consumption.

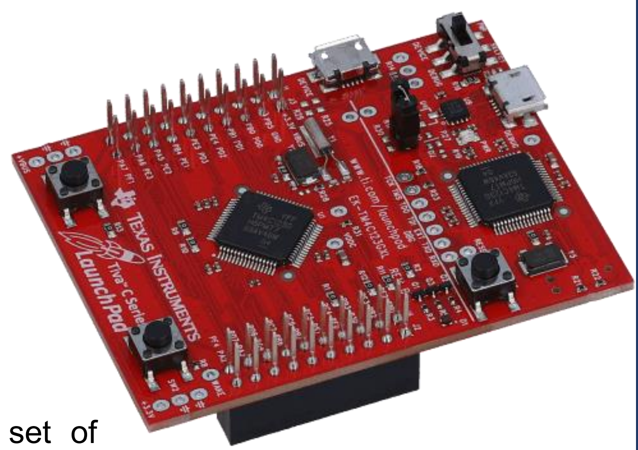


Figure 1 Tiva-C Microcontroller

- TIVA-C microcontrollers come with a rich set of integrated peripherals, including GPIO (General Purpose Input/Output), timers, UART (Universal Asynchronous Receiver-Transmitter), SPI (Serial Peripheral Interface), I2C (Inter-Integrated Circuit), and more. This integration simplifies the design of embedded systems by providing a variety of on-chip resources.

- TIVA-C microcontrollers typically offer a variety of memory options, including Flash memory for program storage and SRAM (Static Random-Access Memory) for data storage. This flexibility allows developers to choose the right balance between program space and data storage based on the requirements of their applications.

2) Flame Sensor

Flame sensors operate on the principle of detecting the specific wavelengths of light emitted by flames. They often use infrared (IR) sensors to identify the characteristic radiation emitted by flames.

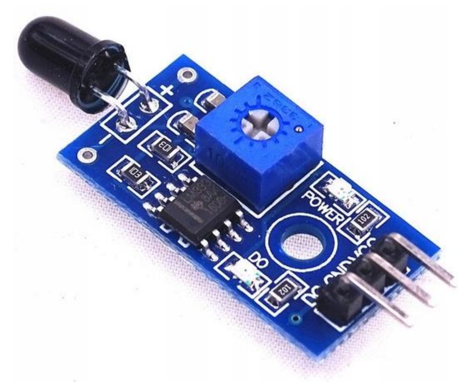


Figure 2 Flame Sensor

Flame sensors are easily integrated into microcontroller-based systems, making them suitable for use in embedded applications. The output of the flame sensor can be processed by the microcontroller to initiate specific actions based on the detected flame.

Ultrasonic sensors are simply integrated into microcontroller-based systems. They provided an analog signal as required but we needed to read specific values so we used ADC Peripheral (Analog to digital converter), allowing the microcontroller to interpret the data and make decisions.

3) Ultrasonic Sensor

Ultrasonic sensors operate on the principle of sending and receiving ultrasonic waves. They emit high-frequency sound waves (ultrasonic pulses) and measure the time it takes for the waves to reflect off an object and return to the sensor.

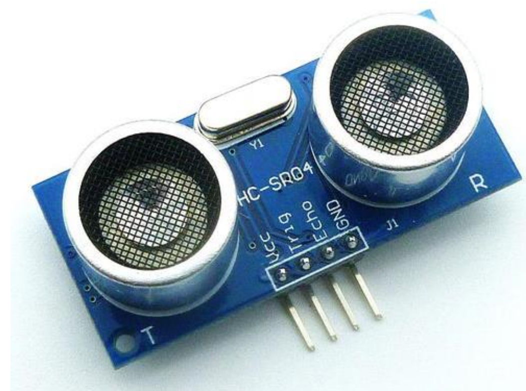


Figure 3 Ultrasonic Sensor

Ultrasonic sensors are commonly used for non-contact distance measurement. By measuring the time delay between the emitted and received ultrasonic waves, the sensor can calculate the distance to the target object.

Ultrasonic sensors are simply integrated into microcontroller-based systems. They provided a digital signal proportional to the measured distance by using timer3 in Tiva-c, allowing the microcontroller to interpret the data and make decisions based on the proximity of objects.

4) Magnetic Sensor

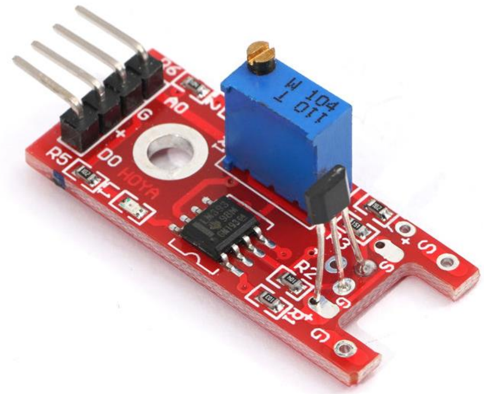


Figure 4 Magnetic Sensor

Magnetic sensors operate based on the Hall effect, where the sensor responds to changes in magnetic fields. The Hall effect refers to the generation of a voltage difference (Hall voltage) across a conductor when subjected to a magnetic field perpendicular to the current flow.

Magnetic sensors are easily integrated into microcontroller-based systems. They can be connected to digital input pins of a microcontroller to detect changes in magnetic fields and trigger appropriate actions.

5) Laser Module

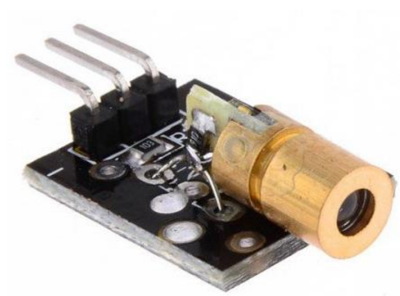


Figure 5 Laser Module

Laser modules operate based on the principle of stimulated emission of radiation. They contain a gain medium (such as a semiconductor material) that, when stimulated by an external energy source, emits coherent light in a specific direction.

the laser module serves as a means of Li-Fi communication. The modulation of the laser light can be used to transmit data between the sender (TIVA-C) and the receiver (Arduino) platforms.

7) Start/Stop Pushbuttons.

Figure 6 Push Bottoms

The start/stop pushbuttons serve as the primary means for users to activate or deactivate the hazard detection system. When the "Start" button is pressed, the system initiates its monitoring operations, and when the "Stop" button is pressed, the system halts its monitoring activities.

8) Bluetooth Module.

The Bluetooth module facilitates wireless communication between the hazard detection system (TIVA-C microcontroller) and a mobile device. This wireless link enables data transfer without the need for physical connections, enhancing the system's flexibility and user convenience.

When the hazard detection system identifies a potential danger, such as a fire or intrusion, it can transmit alert messages via Bluetooth to the connected mobile device. The mobile app can then display these alerts, providing real-time information to the user.

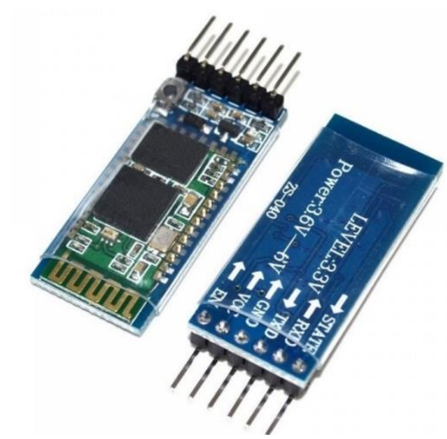


Figure 7 Bluetooth Module

3.3 Receiver Platform (Arduino)

1) Arduino Microcontroller

The Arduino Uno is renowned for its user-friendly environment. The simplicity of the Arduino Integrated Development Environment (IDE) and the straightforward programming model provided an accessible platform for project initiation, especially given the diverse skill set of the development team.

The Arduino microcontroller serves as the receiver platform for the Li-Fi signals transmitted by the TIVA-C microcontroller (sender platform). It is equipped to capture and interpret the modulated light signals sent by the laser emitter module connected to the TIVA-C.

Figure 8 Arduino Uno

2) Light Dependent Resistor (LDR)

The LDR module is employed to capture the light signals transmitted by the laser emitter module connected to the TIVA-C microcontroller.

The varying resistance of the LDR in response to the received light signals is utilized to convert the light intensity into an electrical signal.

Figure 9 Light Dependent Resistor

3) The Light Emitting Diode (LED)

Various sensors, including the LDR, Ultrasonic Sensor, Flame Sensor, and Magnetic Sensor, provide inputs to the Arduino Uno. These sensors detect hazards such as fire or any other hazards, then the Arduino Uno processes the sensor inputs, analyzing the data to determine the type and severity of the detected hazard. Based on this analysis, the Arduino Uno activates the LED for visual alerts.

Figure 10 Light Emitting Diode

4) Buzzer

Various sensors, such as Ultrasonic Sensor, Flame Sensor and Magnetic Sensor provide inputs to the Arduino Uno. These sensors detect hazards like fire, intrusion, or other predefined events within the monitored environment. The Arduino Uno processes the sensor inputs, analyzing the data to identify the type and severity of detected hazards. Based on this analysis, the Arduino Uno activates the Buzzer to generate audible alarms.

Figure 11 Buzzer

5) The Liquid Crystal Display (LCD)

Various sensors, including the LDR, Ultrasonic Sensor, Flame Sensor, and Magnetic Sensor, provide inputs to the Arduino Uno. These sensors detect hazards such as fire or any other hazards, The Arduino Uno processes the sensor inputs, analyzing the data to identify the type and severity of detected hazards. Based on this analysis, the Arduino Uno generates messages to be displayed on the LCD.

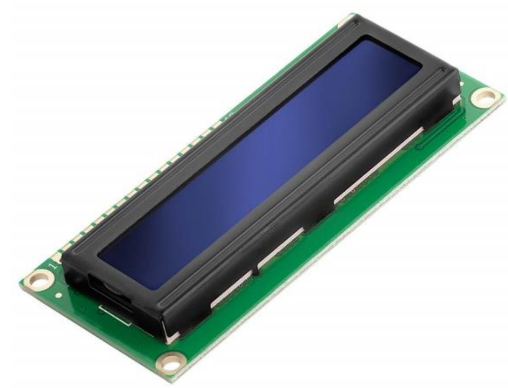


Figure 12 LCD

6) **Muting Push Botton**

The primary role of the mute pushbutton is to si the alarms initiated by the hazard detection system. When a hazard is detected, alarms such as flashing lamps, buzzers, and LCD messages are activated to alert the user. The mute pushbutton allows the user to silence these alarms temporarily.

Figure 13 Push bottoms

4.0 Bluetooth Module (UART)

The Bluetooth module plays a crucial role in facilitating wireless communication between the TIVA-C sender platform and a mobile device, enabling real-time hazard notifications and interaction. In this project, a Serial Bluetooth module is utilized to establish a UART (Universal Asynchronous Receiver-Transmitter) communication link.

1. HC-05 Bluetooth Module Overview:

- The HC-05 is a Bluetooth module that supports UART communication. It operates in Master or Slave mode and is widely used for short-range wireless communication.

2. Hardware Integration:

- **Connections:**
  - Connect the TX pin of the HC-05 to the RX pin of the TIVA-C microcontroller.
  - Connect the RX pin of the HC-05 to the TX pin of the TIVA-C microcontroller.
  - Connect the VCC and GND pins of the HC-05 to the appropriate power supply on the TIVA-C.

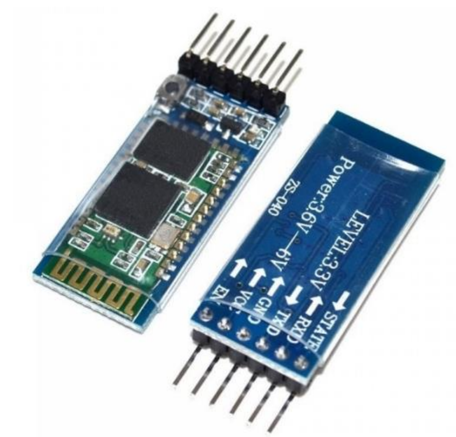


Figure 14: Bluetooth Module

- **Mode Configuration:**
  - Set the HC-05 to the desired mode (Master or Slave) based on the project requirements.
  - Configure the communication parameters (baud rate, data bits, stop bits) as needed.

3. Software Configuration:

- **TIVA-C Firmware:**
  - Initialize the UART module in the TIVA-C firmware code.
  - Implement functions to send hazard detection data over UART to the HC-05.
  - Include error-checking mechanisms to ensure data integrity.
- **HC-05 Configuration:**

Use AT commands to configure the HC-05 settings. Set the Bluetooth module to the desired baud rate using commands like **AT+BAUD**. *Then* Configure the name and PIN of the Bluetooth module using commands like **AT+NAME** and **AT+PIN**.

4. Mobile Application:

- **Serial Bluetooth Terminal App:**
    - Install the "Serial Bluetooth Terminal" app on the mobile device.
    - Pair the mobile device with the HC-05 Bluetooth module.
    - Use the app to establish a virtual serial port connection with the HC-05.
- **Data Reception:**
    - Develop logic in the TIVA-C firmware to send hazard detection data as strings over UART.
    - Interpret and display received data on the "Serial Bluetooth Terminal" app.

5. Integration with Li-Fi Communication:

- **Synchronization:**
    - Ensure synchronization between Li-Fi and Bluetooth communication.
    - Send consistent hazard information to both the mobile app and the Arduino receiver.

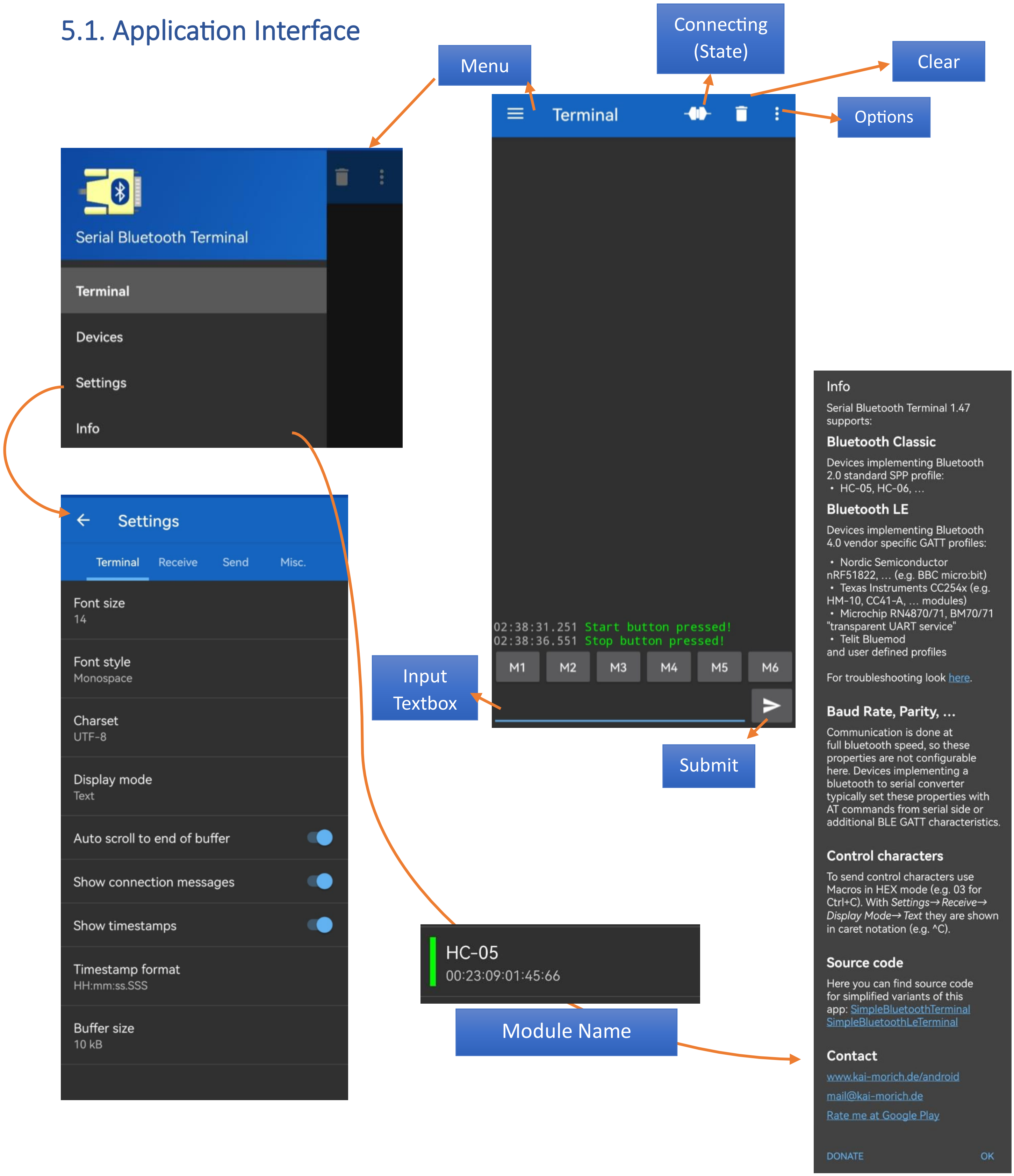
6. User Interaction:

- **Mobile App Features:**
    - Utilize the mobile app to provide user-friendly features such as acknowledgment of alerts, muting alarms, and adjusting system settings.
    - Display hazard-related messages on the mobile app interface.

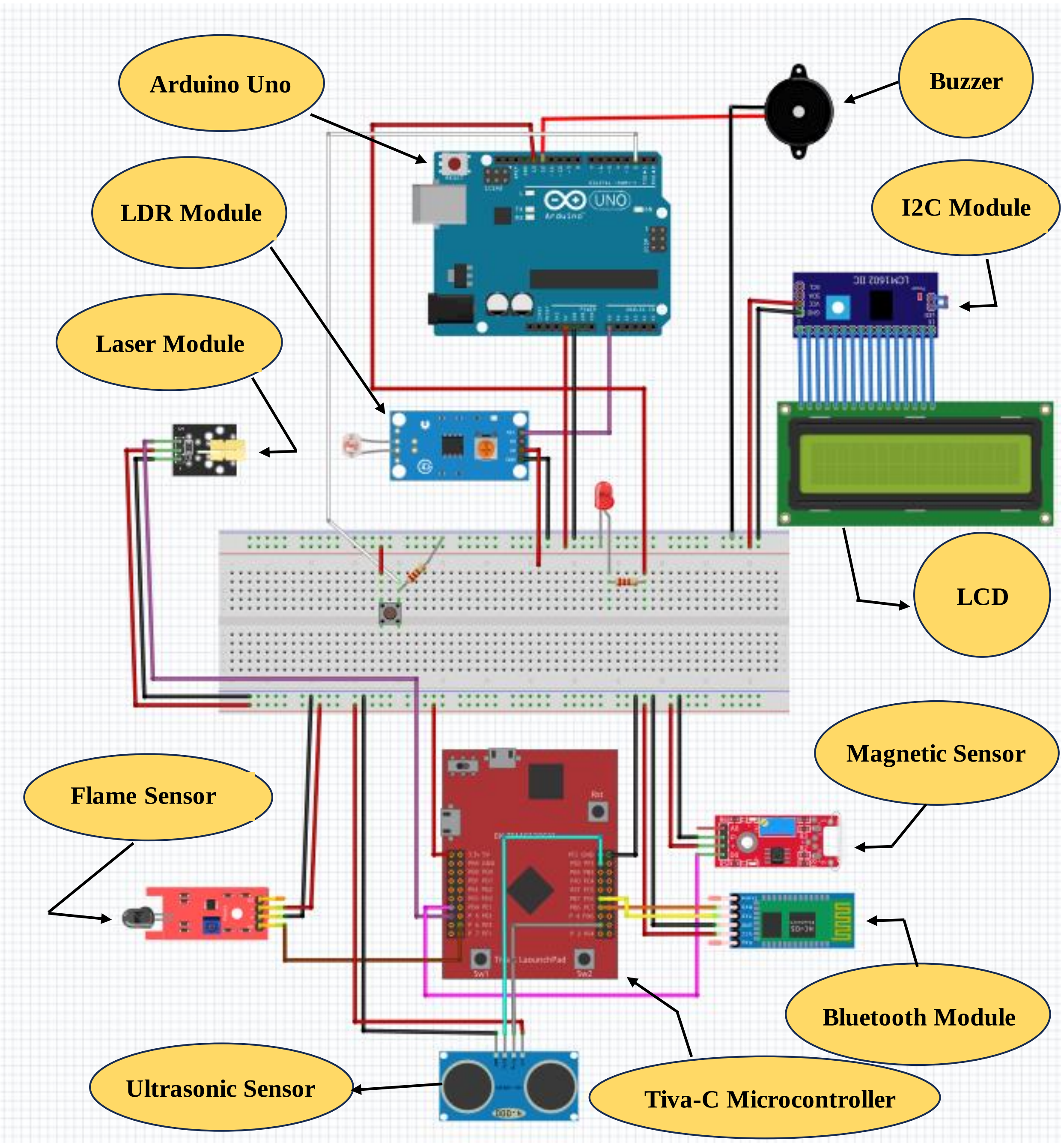
5. Developed Application Screenshots

We used an application Called “Serial Bluetooth Terminal”, Which we used to display all of the required messages that will be displayed on the LCD connected to the Arduino.

5.1. Application Interface



6. System Layout



7. Circuit Wiring

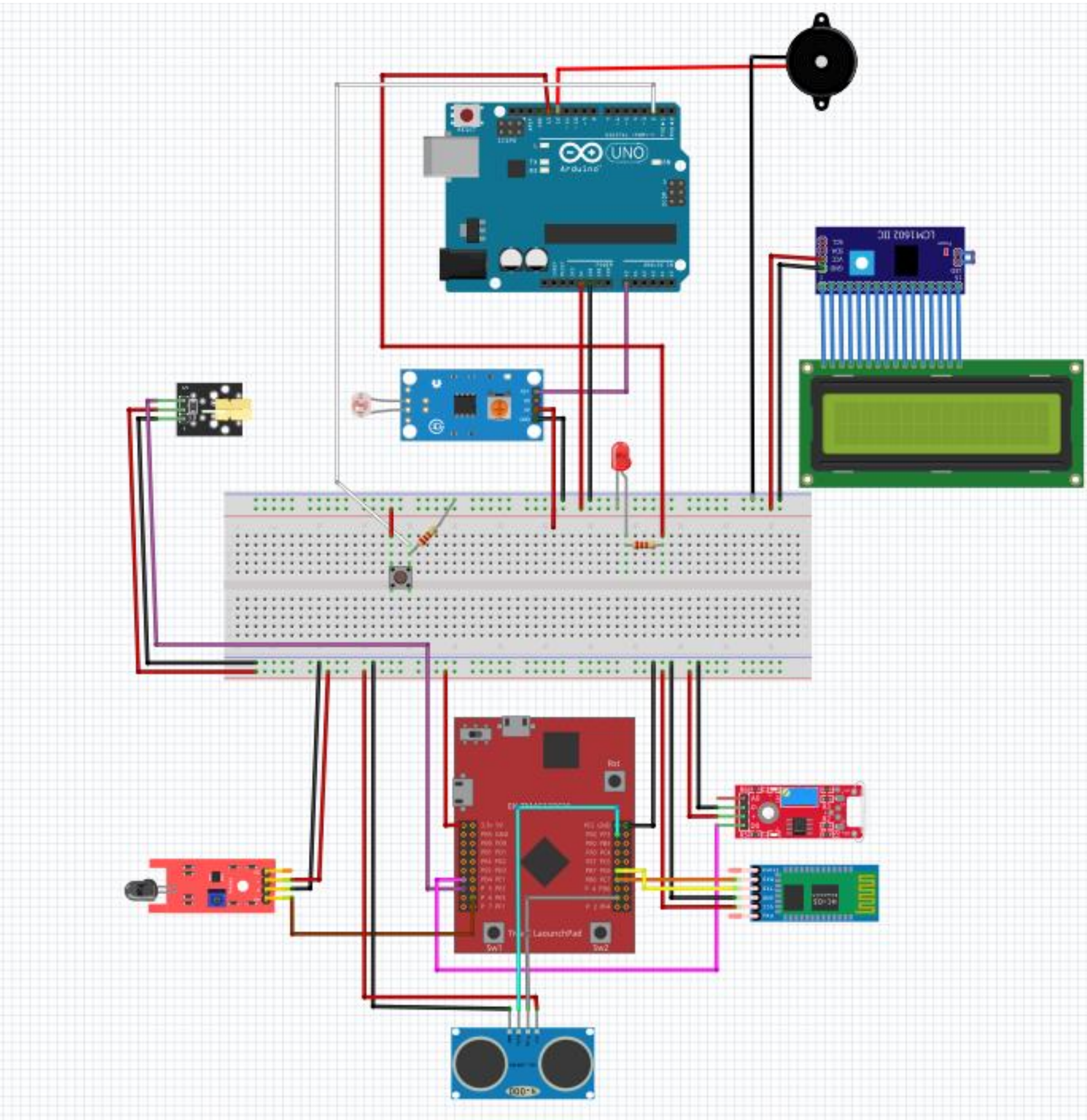


Figure 15: Circuit Wiring on Fritzing

8. Flowcharts

8.1. General Flow Diagram For the project

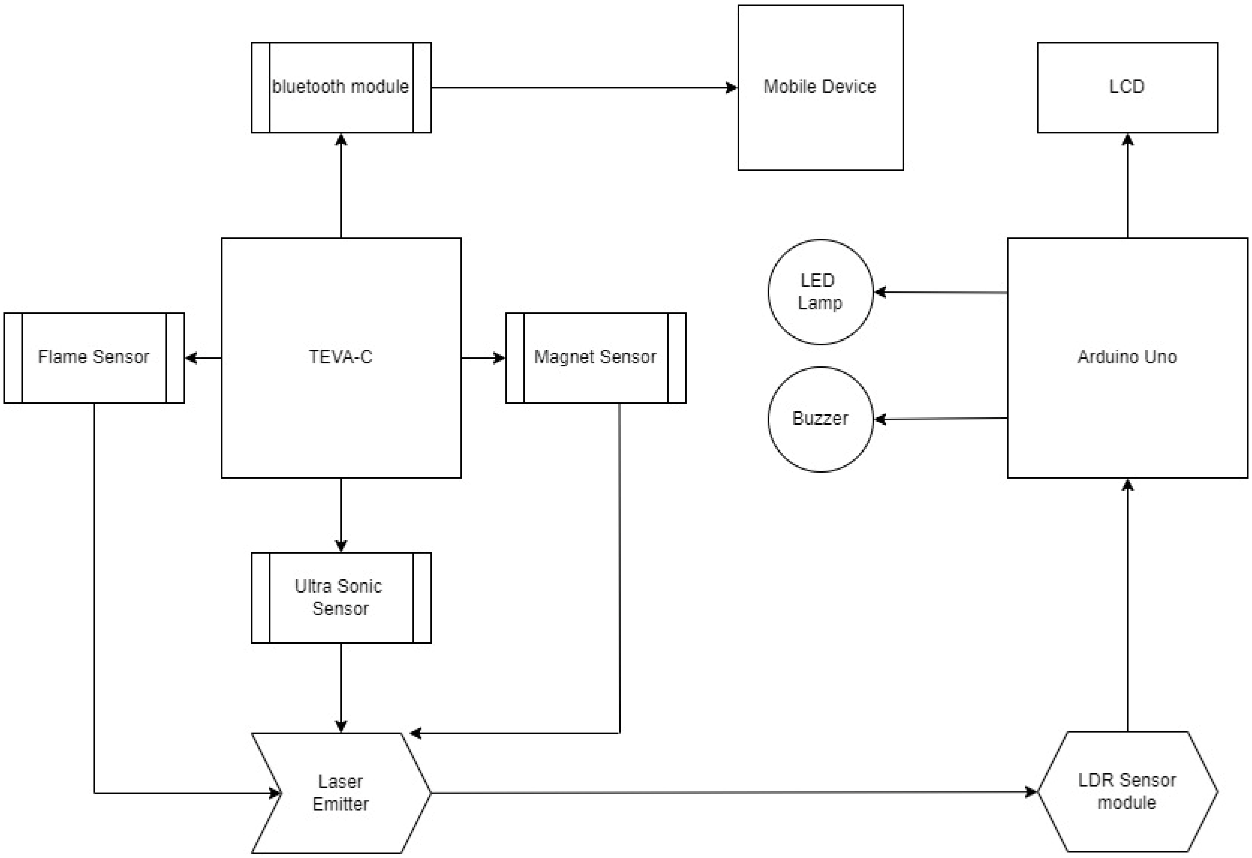
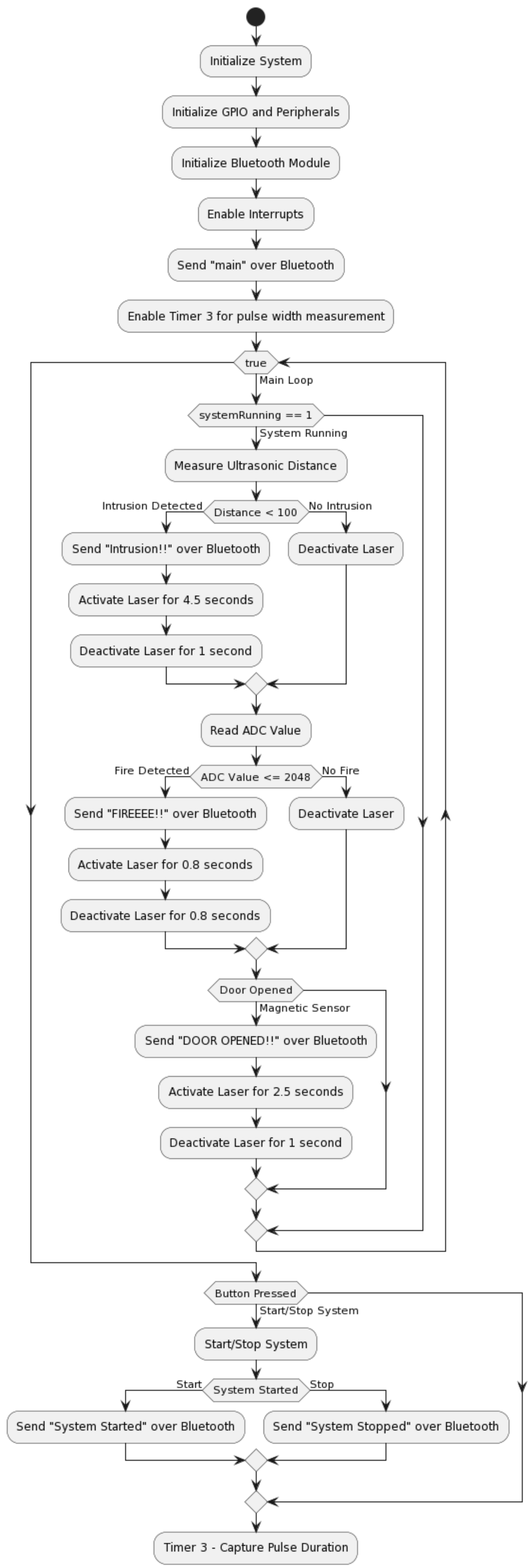


Figure 16: General Flowchart

This is a general flowchart that shows how our project's main components work together. The flow of operations and the connections between different components are shown in the figure, which summarises the system's main functions. It offers a high-level summary of how the primary functions, including ADC readings, interactions with sensors like the Magnetic Sensor, and ultrasonic distance measurement, are coordinated inside the main loop. The system's control flow, in particular the circumstances for identifying intruders, fires, and door openings and setting off appropriate actions like turning on the laser and sending out notifications via Bluetooth, is also depicted in the flowchart.

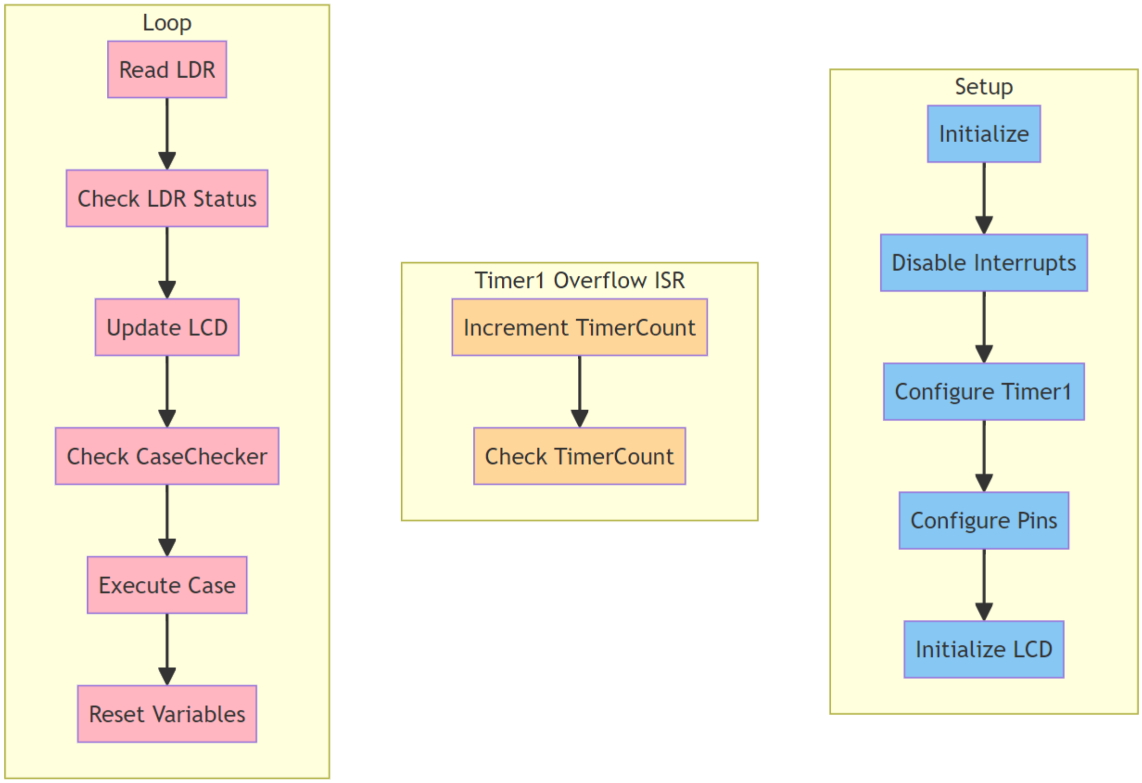
Included is the integration of the Start/Stop system functionality and how it works with external buttons. This flowchart provides a thorough understanding of the project's relationships and logic, but it's vital to remember that certain intricacies and subtleties of the code implementation might need further work to be fully represented.

8.2. Tiva-c



Initialize System
Initialize GPIO and Peripherals
Initialize Bluetooth Module
Enable Interrupts
Send "main" over Bluetooth
Enable Timer 3 for pulse width measurement
true
Main Loop
systemRunning == 1
System Running
Measure Ultrasonic Distance
Intrusion Detected
Distance < 100
No Intrusion
Send "Intrusion!!" over Bluetooth
Deactivate Laser
Activate Laser for 4.5 seconds
Deactivate Laser for 1 second
Read ADC Value
Fire Detected
ADC Value <= 2048
No Fire
Send "FIREEEE!!" over Bluetooth
Deactivate Laser
Activate Laser for 0.8 seconds
Deactivate Laser for 0.8 seconds
Door Opened
Magnetic Sensor
Send "DOOR OPENED!!" over Bluetooth
Activate Laser for 2.5 seconds
Deactivate Laser for 1 second
Button Pressed
Start/Stop System
Start/Stop System
Start
System Started
Stop
Send "System Started" over Bluetooth
Send "System Stopped" over Bluetooth
Timer 3 - Capture Pulse Duration

8.3. Arduino



Loop
Read LDR
Check LDR Status
Update LCD
Check CaseChecker
Execute Case
Reset Variables
Timer1 Overflow ISR
Increment TimerCount
Check TimerCount
Setup
Initialize
Disable Interrupts
Configure Timer1
Configure Pins
Initialize LCD

9. Problems Faced

9.1. Integration of Ultrasonic Driver with Timers:

Problem: Integrating the Ultrasonic sensor with timers posed a challenge. This required synchronizing the timer interrupts with the ultrasonic driver to ensure accurate distance measurements.

Solution: Extensive research was conducted to understand the interaction between timers and the ultrasonic sensor. This involved configuring timer settings to align with the sensor's requirements. Proper synchronization was achieved through precise timer management

9.2. Flame Sensor Integration with ADC:

Problem: Integrating the Flame sensor with the Analog-to-Digital Converter (ADC) presented difficulties. Ensuring accurate readings from the sensor required careful calibration and circuit verification.

Solution: To address this, we initially tested the ADC independently using an external circuit. Once the ADC functionality was validated, the integration with the Flame sensor was executed. Calibrations were fine-tuned to achieve accurate and responsive flame detection, ensuring the reliability of the entire system.

9.3. Integration Process between Tiva and Arduino:

Problem: The integration between Tiva and Arduino platforms presented communication challenges, requiring a seamless data exchange protocol.

Solution: A creative solution was devised to leverage timers and interrupts on the Arduino itself. By employing timers and interrupts, the Arduino independently tracked the duration of the laser signal. The Arduino then communicated this timing information to the Tiva, streamlining the integration process. This innovative approach eliminated the need for complex external communication protocols, simplifying the overall system architecture.

9.4. Interrupt Handling for Case Identification:

Problem: Identifying and handling different cases within the Arduino loop required effective interrupt management to ensure timely responses to sensor inputs.

Solution: Leveraging Timer1 overflow interrupts, we implemented a mechanism to increment a TimerCount variable. This variable was then used to categorize different cases such as Fire, Magnet, and Ultrasonic events. By strategically placing checks in the loop, the Arduino responded appropriately to the sensor inputs based on the TimerCount, enhancing the overall responsiveness of the system.

9.5. LDR Sensor Calibration for Intrusion Detection:

Problem: Calibrating the LDR sensor for intrusion detection posed challenges due to varying ambient light conditions.

Solution: The system implemented a dynamic LDR calibration mechanism. During safe conditions, the LDR sensor readings were calibrated to establish a baseline. This baseline was then used to differentiate between safe and intrusion scenarios, ensuring reliable intrusion detection under changing lighting conditions.

10. Tiva-Ware library Implementation

Incorporating the TIVAWARE library into our project presented both challenges and invaluable contributions. The TIVAWARE library, designed for Tiva C Series microcontrollers, provided a comprehensive set of functions and tools that greatly facilitated the development process. However, navigating and integrating this extensive library posed initial challenges due to its complexity and feature-rich nature.

Once we overcame the learning curve, the TIVAWARE library proved instrumental in enhancing the functionality and efficiency of our project. Its robust set of pre-built functions streamlined various tasks, from low-level hardware control to complex functionalities, enabling us to focus on higher-level aspects of our project.

Despite the initial challenges, the utilization of the TIVAWARE library significantly contributed to the project's success, showcasing its importance in embedded systems development. The experience of working with this library not only expanded our understanding of embedded programming but also underscored the significance of leveraging established libraries to expedite development while maintaining code reliability.

11. Real Life Photos

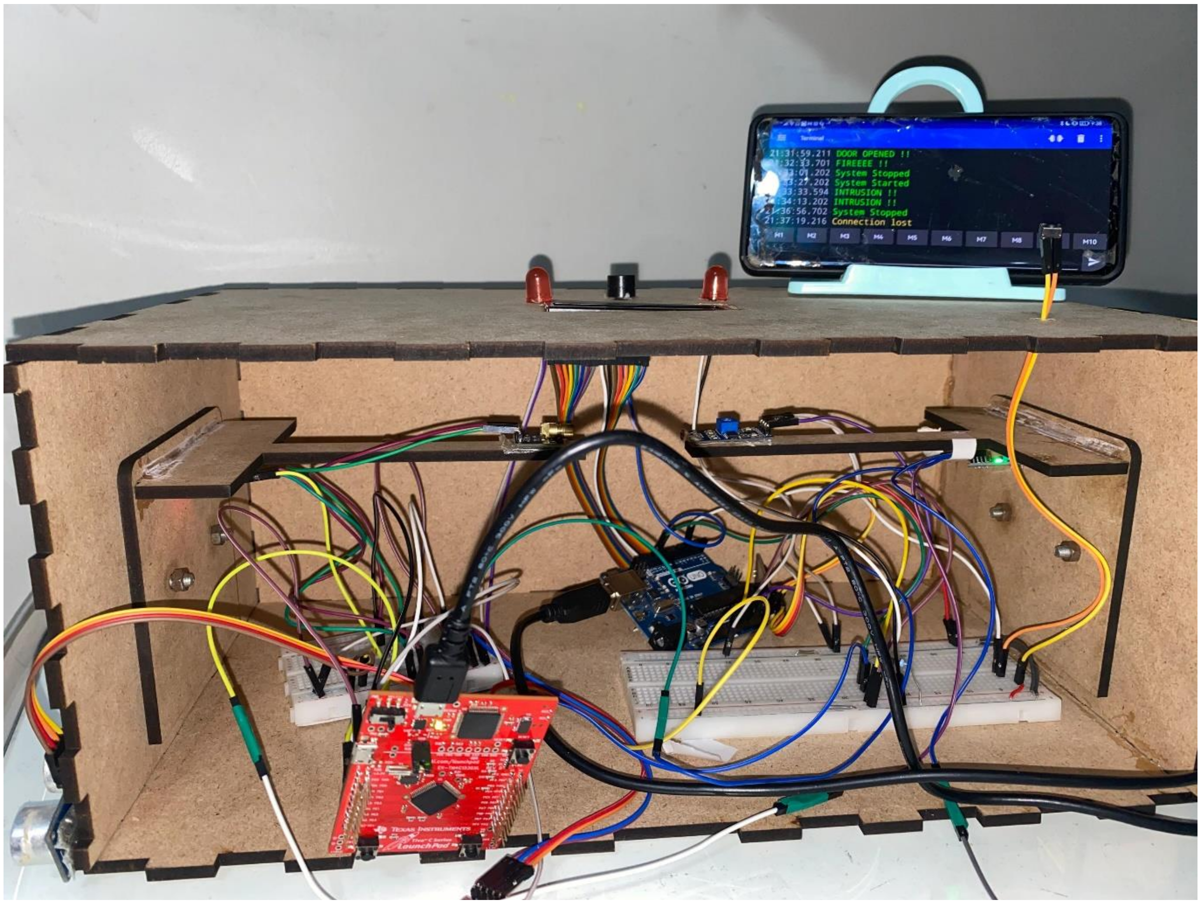


Figure 19: Whole Project

Figure 17: Internal Picture of the project

Figure 18: Alarm System

12. Extra Circuits we Did

12.1. Laser emitter module Control Circuit

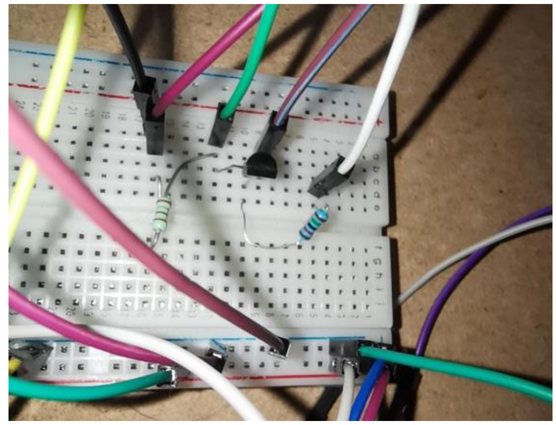


Figure 20: BJT with Laser Module

We designed an amplification and control circuit for the laser module to operate efficiently at 5 volts, contrary to the standard 3.3 volts used by sensors in the Tiva. Our objective was to harness the laser module's full capacity. Upon reviewing the datasheet, we discovered that the laser module required 5 volts instead of 3.3 volts.

To achieve this, we implemented a combination of electrical components to regulate the input voltage to the laser module. This configuration included a Bipolar Junction Transistor (BJT) and several resistors. The tailored arrangement of these components allowed us to seamlessly control the voltage supplied to the laser module, ensuring optimal performance while integrating it into a system predominantly operating at 3.3 volts.

12.2. LCD Screen with I2C module

Using an LCD with an I2C (I2C or IIC) module in Arduino offers several advantages over using a standalone LCD without an I2C interface:

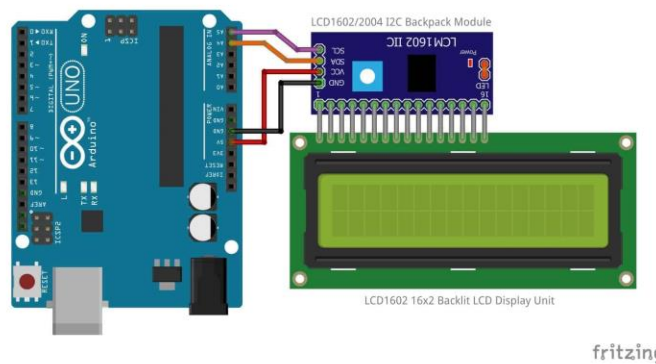


Figure 21: LCD with I2c

- Reduced Pin Usage: LCDs without I2C require a significant number of digital pins to operate, limiting the available pins for other components. I2C allows you to control the LCD using only two pins (SDA and SCL), freeing up other digital pins for additional functionalities.
- Simplified Wiring: With I2C, the wiring becomes more straightforward. You only need to connect the SDA and SCL pins to the corresponding pins on the Arduino, simplifying the connection process.
- Addressing Multiple Devices: I2C supports multiple devices on the same bus. Each I2C device has a unique address, and the Arduino can communicate with different devices using the same set of pins.
- Ease of Use: Libraries such as the "LiquidCrystal_I2C" library make it easier to control an LCD with an I2C module. This simplifies code development and reduces the complexity of managing low-level details.
- Space Saving: I2C modules are often compact and can be integrated directly onto the back of the LCD, saving space and making the overall setup more compact.
- Improved Readability: Some I2C modules come with adjustable contrast and backlight, allowing for better visibility and customization

13. Project Environment Design (CAD)

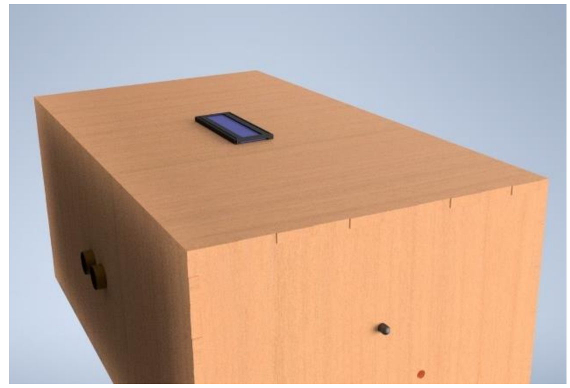


Figure 22: Home View

We achieved a significant milestone in the project by meticulously crafting a dedicated environment—an enclosed **box** that serves as a comprehensive solution ensuring precise signal transduction without errors. This enclosure not only enhances the functionality of the system but also addresses several key considerations:

1. **Signal Integrity:** The carefully designed box shields the internal components from external interference, safeguarding the signal's integrity. This is crucial for reliable and accurate data processing.
2. **Error Minimization:** By enclosing the components within a controlled environment, we effectively minimized the potential for errors that could arise from external factors such as electromagnetic interference or ambient light.
3. **Ease of Handling:** The integrated design of the box makes the entire setup much easier to handle. This is particularly important for practical applications, allowing users to transport and deploy the system with convenience.

Figure 23: Uncovered Box

4. **Functional Integration:** The enclosure serves as a platform for the seamless integration of diverse components, promoting an organized and efficient layout. This not only contributes to the aesthetic appeal but also enhances the overall functionality of the system.
5. **User-Friendly Design:** Our focus on creating a user-friendly environment ensures that the device is not only technologically advanced but also intuitive and easy to use. This is a key aspect in real-world applications where user interaction is a critical consideration.

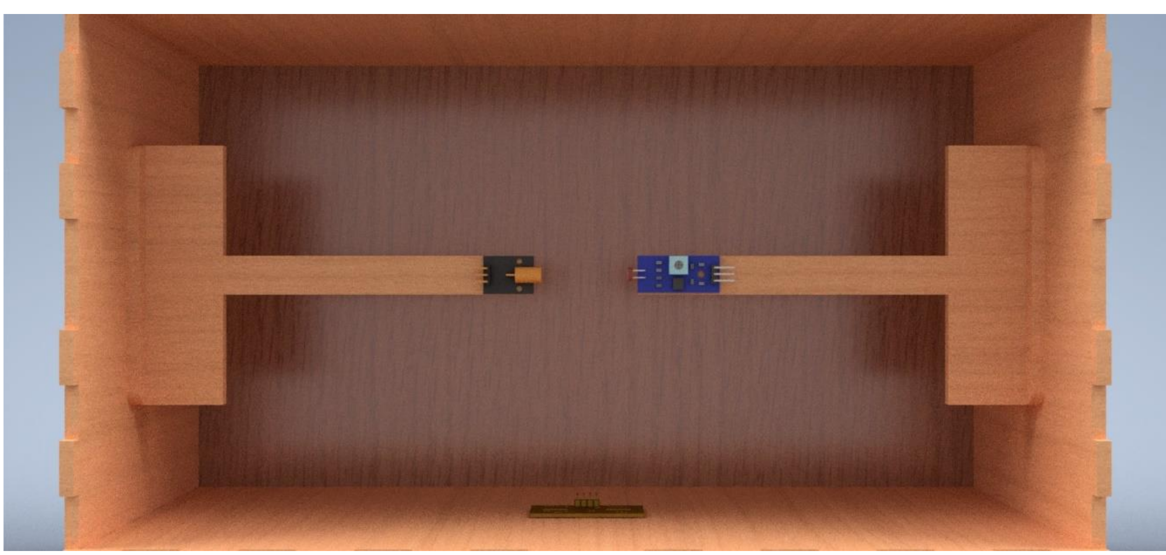


Figure 24: Top View

14. Conclusion

In conclusion, our journey in developing this project has been both challenging and rewarding, providing valuable insights into various aspects of hardware integration, signal processing, and system design. Here are some key takeaways and considerations for improvement:

1. **Integration Expertise:** Through challenges such as integrating the ultrasonic driver with timers and incorporating the flame sensor with ADC, we honed our skills in navigating complex hardware integrations. This experience underscores the importance of thorough research and experimentation in overcoming integration hurdles.

2. **Creative Problem Solving:** The successful resolution of issues, such as using Arduino timers and interrupts for communication between Tiva and Arduino, showcased our ability to think creatively and devise innovative solutions. This adaptive problem-solving approach is a valuable skill in the field of electronics and embedded systems.

3. **Documentation and Research:** The project underscored the significance of in-depth research and comprehensive documentation. Clear documentation not only aids in troubleshooting but also facilitates knowledge transfer and collaboration among team members.

4. **Voltage Compatibility:** Addressing challenges related to voltage differences, such as amplifying the laser module's control circuit for compatibility with the Tiva's 3.3V, demonstrated our meticulous attention to hardware specifications. Future projects can benefit from an early and thorough assessment of voltage compatibility issues.

5. **Collaborative Teamwork:** The collaborative effort in overcoming challenges involving the Arduino and Tiva integration highlighted the importance of effective teamwork. Clear communication and a shared understanding of project goals are crucial for success.

Areas for Improvement:

1. **Testing Protocols:** Implementing more robust testing protocols during the development phase could enhance the reliability and performance of the system. Rigorous testing can help identify and address potential issues early in the process.

2. **Scalability:** Considerations for scalability and future modifications should be integrated into the initial design. This ensures that the system can adapt to evolving requirements without significant redesign.

3. **User Interface Enhancement:** While the system's functionality is paramount, improvements in the user interface can enhance the overall user experience. Intuitive controls, informative displays, and user-friendly interactions contribute to a more accessible and user-centric solution.

4. **Power Efficiency:** Evaluating and optimizing power consumption is crucial for applications where energy efficiency is a concern. Future iterations could explore ways to enhance power efficiency without compromising system performance.