

ARM-based Smart Home Application using Tiva C [TM4C123x]

Mohamed Yasser
20P3280@eng.asu.edu.eg

Tasneem Abbas
19P4152@eng.asu.edu.eg

Mariam Hesham
197777@eng.asu.edu.eg

Youssef Mahmoud
18P8814@eng.asu.edu.eg

Doaa Hamada
17P3020@eng.asu.edu.eg

August 22, 2023

1 Introduction

In this era of smart devices and automation, making our living spaces smarter and more convenient has become a necessity. As a group, we decided to take a step in this direction by designing a smart home application that allows users to control and monitor various devices in their home. This project aims to control a 220V lamp and a plug, monitor the door status, and display the room temperature. In addition, the system will trigger an alarm if the temperature exceeds a set value and will automatically turn on a fan for cooling.

The core of this project is the Tiva C series microcontroller (TM4C123GH6PM), which will be responsible for interfacing with the various components and handling the control logic. We will also use the HC-05 Bluetooth module to enable communication with a mobile application, providing users with remote control and monitoring capabilities. The use of a relay module allows us to safely control high-voltage devices, while a limit switch will detect the door status. Finally, a buzzer will serve as a physical alarm to alert users when the temperature exceeds the set threshold.

This document provides a detailed description of the hardware setup, software implementation, and testing procedures for this project.

2 Hardware Setup

2.1 Components List

The components used in this project are:

- **Tiva C series microcontroller (TM4C123GH6PM):** A 32-bit ARM Cortex-M4F microcontroller with up to 256KB of Flash memory, 32KB of SRAM, and a wide range of peripherals such as UART, SPI, I2C, and GPIO. It operates at a frequency of up to 50 MHz and is suitable for a wide range of applications, including real-time control systems.
- **HC-05 Bluetooth module:** A wireless communication module that operates on the Bluetooth 2.0 standard. It supports SPP (Serial Port Profile) and can be configured as a master or slave device. The module has a range of up to 10 meters and can be interfaced with microcontrollers using UART. It allows for wireless communication between the microcontroller and a mobile application.
- **2-channel relay module:** A module that contains two electromagnetic relays, allowing for the control of high-voltage devices (up to 250V AC or 30V DC) using low-voltage control signals. The module provides optical isolation between the control and load circuits, ensuring safe operation. It is suitable for controlling devices such as lamps, fans, and plugs.
- **Limit switch:** A type of switch that is activated when it comes into contact with an object. In this project, it is used to detect the door status (open or closed). Limit switches are commonly used in applications where precise positioning or movement detection is required.
- **Fan:** An electric fan used for cooling purposes. In this project, it is activated when the room temperature exceeds a set threshold. It helps to reduce the temperature and provide a more comfortable environment.
- **Buzzer:** An audio signaling device that produces a sound when a voltage is applied. In this project, it serves as a physical alarm to alert users when the room temperature exceeds the set threshold.
- **DHT11–Temperature and Humidity Sensor:** A digital sensor that measures both temperature and humidity. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air and provides a digital signal output. The DHT11 can measure temperature from 0 to 50°C with an accuracy of $\pm 2^\circ\text{C}$, and humidity from 20% to 80% with an accuracy of $\pm 5\%$. It is suitable for monitoring environmental conditions in a room.

2.2 Connections

- **Tiva C to HC-05 Bluetooth module:** Connect the RX pin of the HC-05 module to the PA1 pin (UART0 TX) of the Tiva C board, and the TX pin of the HC-05 module to the PA0 pin (UART0 RX) of the Tiva C board. Also, connect the VCC and GND pins of the HC-05 module to an external 5V supply.
- **Tiva C to the first 2-channel relay module:** Connect the IN1 pin of the relay module to the PA5 pin (GPIO) of the Tiva C board, and the IN2 pin of the relay module to the PA6 pin (GPIO) of the Tiva C board. Also, connect the VCC and GND pins of the relay module to to an external 5V supply. Connect one channel of the relay to both fan and buzzer and the other channel to the plug.
- **Tiva C to Limit switch:** Connect the Normal Closed terminal of the limit switch to the PB2 pin (GPIO) of the Tiva C board, and the Common terminal to the GND pin of the Tiva C board. Enable the internal pull-up resistor on the PB2 pin to ensure a stable input signal.
- **First 2 channel relay module to Fan and buzzer:** Connect the Common terminal to 220V source, then connect the fan and buzzer to the normal open terminal of the first channel of the relay module. The Tiva C board will control the fan by activating the corresponding relay channel using IN1 on pin PA5.
- **First 2 channel relay module to plug:** Connect the Common terminal to 220V source, then connect the plug to the normal open terminal of the second channel of the relay module. The Tiva C board will control the plug by activating the corresponding relay channel using IN2 on pin PA6.
- **Second 2 channel relay module to two-way switch:** Connect the Common terminal of the first channel to L1 from the two-way switch, likewise connect the Common terminal of the second channel to L2 from the two-way switch. Connect the input of the two-way switch to the 220V supply.
- **Second 2 channel relay module to Lamp:** Parallely connect the Normally Closed terminal of the first channel to the Normally Open terminal of the second channel, then connect it to the positive terminal of the lamp. Don't forget to connect the negative terminal to the shared ground of the 220V supply.
- **Tiva C to DHT11 sensor:** Connect the VCC and GND pins of the DHT11 sensor to an external 5V supply, and the DATA pin of the DHT11 sensor to the PC5 pin (GPIO) of the Tiva C board. Connect a 10k Ω pull-up resistor between the DATA pin and the VCC pin of the DHT11 sensor [as recommended in its datasheet].

2.3 Wiring Diagram

Below are the wiring diagrams for the various components used in this project, along with detailed explanations of the connections depicted in each diagram.

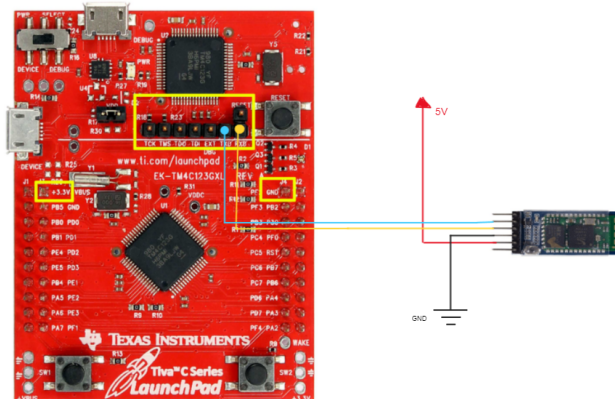


Figure 1: Microcontroller to Bluetooth Module Wiring Diagram

In Figure 1, the connections between the Tiva C microcontroller and the HC-05 Bluetooth module are shown. The module communicates with the microcontroller via UART, with the module's RX pin connected to the microcontroller's UART0 TX (PA1) pin, and the module's TX pin connected to the microcontroller's UART0 RX (PA0) pin. The module is powered by connecting its VCC and GND pins to an external 5V supply.

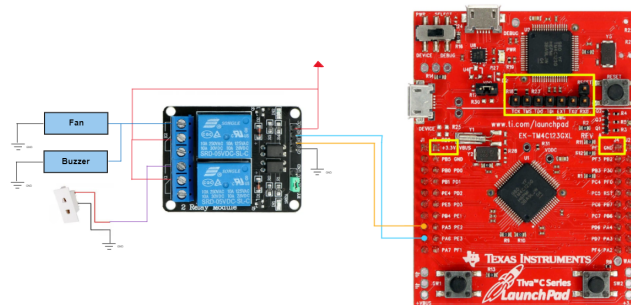


Figure 2: Microcontroller to First Relay Module Wiring Diagram

Figure 2 illustrates the connections between the Tiva C microcontroller and

a 2-channel relay module, to control plug, buzzer, and fan. The relay module's IN1 and IN2 pins are connected to the microcontroller's GPIO pins (PA5 and PA6), allowing the microcontroller to control the relay's channels. The module is powered by connecting its VCC and GND pins to an external 5V supply.

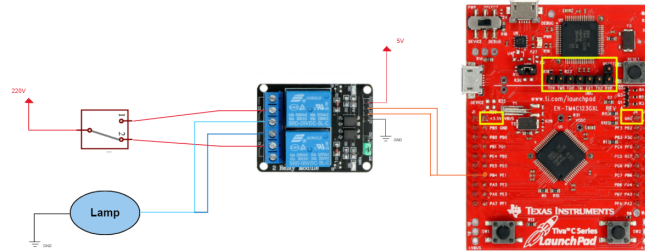


Figure 3: Microcontroller to Second Relay Module Wiring Diagram

Figure 3 illustrates the connections between the Tiva C microcontroller and a 2-channel relay module, to control the lamp from both two-way switch and mobile application. The relay module's IN1 and IN2 pins are connected to the same PIN of the microcontroller's GPIO which is PE1, allowing the microcontroller to control the relay's channels. The module is powered by connecting its VCC and GND pins to an external 5V supply.

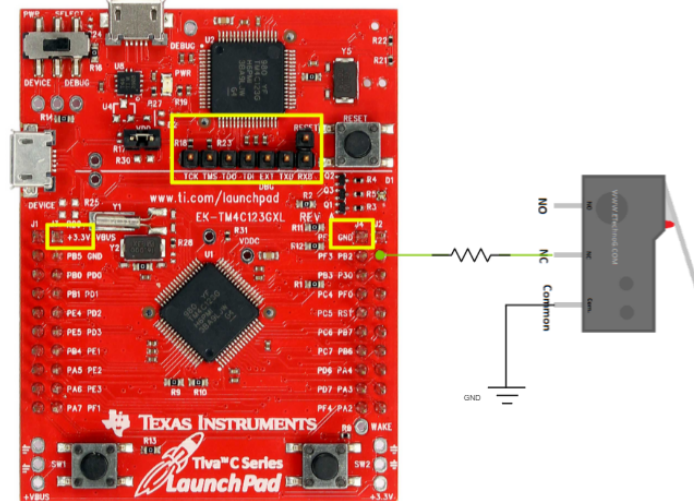


Figure 4: Microcontroller to Limit Switch Wiring Diagram

In Figure 4, the connections between the Tiva C microcontroller and the limit switch are shown. The Normal Closed terminal of the limit switch is connected to the microcontroller's GPIO pin (PB2) with an internal pull-up resistor enabled, while the Common terminal is connected to GND. The placement of the limit switch on the door is also depicted, which allows for the detection of the door's open or closed status.

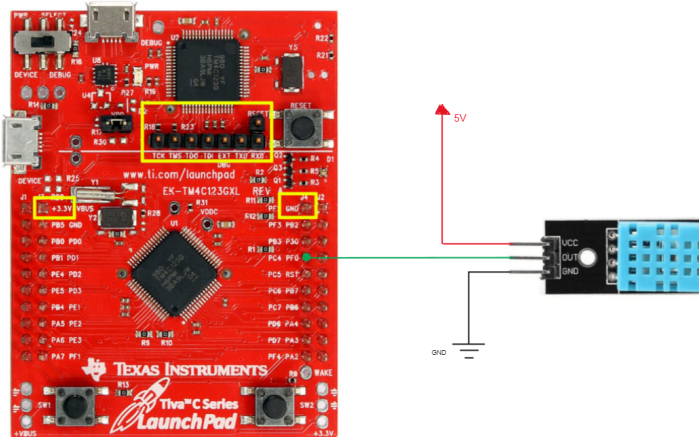


Figure 5: Microcontroller to DHT11 Sensor Wiring Diagram

In Figure 5, the connections between the Tiva C microcontroller and the DHT11 sensor are shown. The sensor's VCC and GND pins are connected to the 5V and GND pins of an external DC supply, respectively. The sensor's DATA pin is connected to the microcontroller's GPIO pin (PC4), with a 10kΩ pull-up resistor connected between the DATA pin and VCC [as recommended in its datasheet]. The sensor communicates using a custom one-wire protocol, so we had to make sure that the microcontroller handle the data format and timing requirements properly.

3 Simulation

3.1 Proteus Simulation for Lamp Control

In this subsection, we present the simulation of the lamp control part of the project using Proteus. Proteus is a powerful electronic design automation (EDA) tool that allows for the simulation and verification of circuits before implementing them in hardware. By simulating the lamp control circuit in Proteus, we can validate the design and ensure that the components are correctly connected and configured. The simulation allows us to test the functionality of the lamp control, including the proper operation of the relay module and the response to control commands from the microcontroller. This step is crucial for identifying and resolving any potential issues before moving on to the hardware implementation.

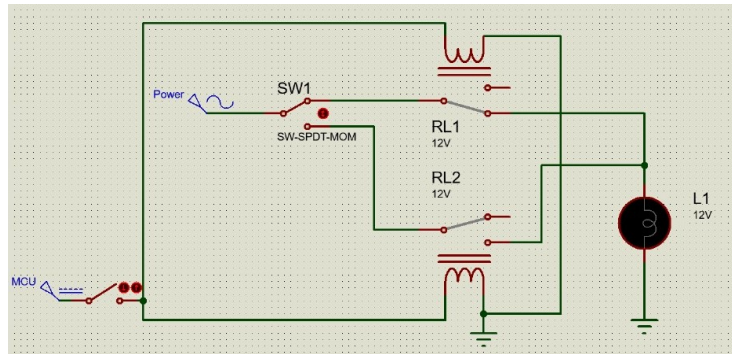


Figure 6: Proteus Simulation of Lamp Control

Figure 6 shows the Proteus simulation of the lamp control circuit. The diagram depicts the connections between the Tiva C microcontroller, the relay module, two-way switch, and the lamp. The simulation demonstrates how the lamp is turned on and off based on both the control signals from the microcontroller and the switch. The results of the simulation confirm the correct operation of the lamp control circuit.

4 Analysis

In this section, we will analyze the operation and key features of the components used in this project. We will provide an in-depth explanation of how each component works and how it contributes to the overall functionality of the smart home application.

4.1 DHT11 Sensor

The DHT11 sensor is a digital temperature and humidity sensor that provides accurate and reliable measurements. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air and provides a digital signal output. The sensor communicates using a custom single-wire two-way serial interface.

4.1.1 Communication Process

The communication process of the DHT11 sensor involves the following steps:

1. **Start Signal and Response:** The microcontroller initiates communication by pulling the data line low for at least 18 ms and then releasing it (pulling it high) and wait 20-40us for DHT's response. The DHT11 sensor detects the start signal and responds by pulling the data line low for 80 μ s (acknowledgment signal) and then pulling it high for 80 μ s.
2. **Data Transmission:** The DHT11 sensor transmits 40 bits of data, consisting of the integer and decimal parts of the humidity and temperature readings, and a checksum value. Each bit is transmitted by pulling the data line low for 50 μ s and then pulling it high for a variable amount of time. A high level of 26-28 μ s represents a 0, while a high level of 70 μ s represents a 1.
3. **Data Reception:** The microcontroller reads the data bits and assembles them into bytes. The first two bytes represent the humidity reading, the next two bytes represent the temperature reading, and the last byte is the checksum. The checksum is calculated by adding the first four bytes and taking the lower 8 bits of the result. The microcontroller compares the received checksum with the calculated checksum to verify data integrity.
4. **End of Communication:** After receiving the data, the microcontroller pulls the data line low for 50 μ s to signal the end of communication. The DHT11 sensor then enters a low-power mode until the next start signal is received.

This communication process allows the DHT11 sensor to transmit temperature and humidity readings to the microcontroller in a reliable and efficient manner. Proper timing and data validation are essential for successful communication.

4.2 HC-05 Bluetooth Module

The HC-05 Bluetooth module is a wireless communication module based on the Bluetooth 2.0 standard. It supports Serial Port Profile (SPP) and can be configured as a master or slave device.

4.2.1 Pairing and Communication

1. **Pairing:** The Bluetooth module enters discovery mode, allowing it to be paired with a mobile device. Once paired, it establishes a wireless connection with the mobile device.
2. **Serial Communication:** The module communicates with the microcontroller via UART. Data received from the mobile device is sent to the microcontroller over UART, and vice versa. This allows for two-way communication between the mobile application and the microcontroller.

4.3 2-Channel Relay Module

The 2-channel relay module contains two electromagnetic relays for controlling high-voltage devices using low-voltage control signals. It provides optical isolation between the control and load circuits.

4.3.1 Relay Operation

1. **Activation:** When a low voltage is applied to the relay's control pin, it activates the relay, closing the contacts and allowing current to flow through the connected device (e.g., fan, buzzer, lamp or plug).
2. **Deactivation:** When the control voltage is removed, the relay deactivates, opening the contacts and stopping the current flow through the connected device.

4.4 Limit Switch

The limit switch is a type of switch that is activated when it comes into contact with an object. In this project, it is used to detect the door status.

4.4.1 Switch Operation

1. **Door Closed:** When the door is closed, the limit switch is pressed, and the connected GPIO pin reads a low level.
2. **Door Open:** When the door is open, the limit switch is released, and the connected GPIO pin reads a high level.

4.5 Fan

The fan is an electric device used for cooling purposes. It is controlled by the microcontroller via the relay module.

4.5.1 Fan Operation

1. **Activation:** When the room temperature exceeds the set threshold, the microcontroller activates the relay connected to the fan, turning it on.
2. **Deactivation:** When the temperature falls below the threshold, the microcontroller deactivates the relay, turning the fan off.

4.6 Buzzer

The buzzer is an audio signaling device that produces a sound when a voltage is applied.

4.6.1 Buzzer Operation

1. **Activation:** When the room temperature exceeds the set threshold, the microcontroller drives the buzzer's GPIO pin high, activating the buzzer and producing an audible alarm.
2. **Deactivation:** When the temperature falls below the threshold or the alarm is acknowledged, the microcontroller drives the buzzer's GPIO pin low, deactivating the buzzer.

5 Software Implementation

5.1 Main Application - main.c

The `main.c` file serves as the central hub for the smart home application. It initializes and integrates all the components, manages the control loop, and oversees the communication with external devices. The core functionality includes reading data from the DHT11 sensor and limit switch, controlling the fan and buzzer based on temperature readings, and interfacing with the mobile application through the Bluetooth module.

```
int main(void)
{
    // Initialize the components
    Relay_Init();
    LimitSwitch_Init();
    HCO5_Init();
    DHT11_Init();

    while (1)
    {
        // Main Block
    }

    return 0;
}
```

In the initialization phase, each component's corresponding initialization function is invoked. The main control loop then continuously monitors the temperature and door status, adjusting the system's behavior accordingly. Additionally, data is sent and received from the mobile application to provide remote control capabilities.

5.2 Relay Module - relay.c and relay.h

The relay module is pivotal for controlling high-power devices like lamps and plugs in the smart home system. The module's code facilitates the initialization of relay control pins and offers a function to adjust the relays according to commands from the mobile application.

```
void Relay_Control(int command)
{
    // Control the relays based on the received command
    switch (command)
    {
        case 0: // Turn off relay 1 channel 1
            GPIOA->DATA &= ~RELAY1_PIN1;
    }
}
```

```

        break;
    case 1: // Turn on realy 1 channel 1
        GPIOA->DATA |= RELAY1_PIN1;
        break;
    case 2: // Turn off realy 1 channel 2
        GPIOA->DATA &= ~RELAY1_PIN2;
        break;
    case 3: // Turn on realy 1 channel 2
        GPIOA->DATA |= RELAY1_PIN2;
        break;
    case 4: // Turn off relay 2
        GPIOE->DATA &= ~RELAY2_PIN;
        break;
    case 5: // Turn on relay 2
        GPIOE->DATA |= RELAY2_PIN;
        break;
    }
}

```

The relay module differentiates between multiple commands, such as turning off both relays, activating relay 1, activating relay 2, or enabling both relays simultaneously. This granularity in control ensures flexible device management in the smart home environment.

5.3 Limit Switch - limitswitch.c and limitswitch.h

The limit switch module is crucial for monitoring the door's status in the smart home system. Through its code, the module efficiently initializes the limit switch pin and provides a function to retrieve the door's current status.

```

int isDoorClosed(void)
{
    // Read and return the status of the limit switch
    return (GPIOB->DATA & LIMIT_SWITCH_PIN) ? 1 : 0;
}

```

Upon initialization, the pin designated for reading the limit switch is set as an input pin with an attached pull-up resistor. The module offers a function to read and relay the status of the switch, allowing the system to determine whether the door is open or closed.

5.4 HC-05 Bluetooth Module - hc05.c and hc05.h

The HC-05 Bluetooth module plays a pivotal role in facilitating remote control and monitoring capabilities for the smart home system. Its code initializes the UART interface necessary for communication with the Bluetooth module and provides robust functions for both sending and receiving data.

```

void HCO5_SendData(int temperature, int door_status)
{
    // Send the temperature and door status to the mobile application
    HCO5_UART->DR = temperature;
    while (HCO5_UART->FR & 0x20);
    HCO5_UART->DR = door_status;
    while (HCO5_UART->FR & 0x20);
}

int HCO5_ReceiveData(void)
{
    // Receive and return the control command from the mobile application
    while (HCO5_UART->FR & 0x10);
    return HCO5_UART->DR;
}

```

Once initialized, the Bluetooth module is ready to communicate with paired devices. The provided functions allow the system to send temperature and door status data to the mobile application. Conversely, control commands received from the mobile application are processed to control the connected devices in the smart home system.

5.5 DHT11 Temperature Sensor - dht11.c and dht11.h

The DHT11 temperature sensor module provides the smart home system with real-time temperature readings. Its code ensures the proper initialization of the sensor's data line and decodes the specific communication protocol of the DHT11 sensor.

```

/* Send Start Signal */
GPIOC->DATA &= ~0x10; // make the pin LOW
delay_us(18000); // wait 18ms
GPIOC->DATA |= 0x10; // make the pin HIGH

delay_us(20); // wait 30us
GPIOC->DIR &= ~0x10; // convert to input

/* Wait for Acknowledgement */
while(GPIOC->DATA >> 4 == 1); // wait until it's LOW
while(GPIOC->DATA >> 4 == 0); // wait until it's HIGH
while(GPIOC->DATA >> 4 == 1); // wait until it's LOW ,then it will start sending data

```

The initialization phase prepares the data line pin for communication. When the system requires a temperature reading, the module initiates communication with the sensor, adhering to the DHT11's unique protocol, and returns the retrieved temperature value.

5.5.1 Conclusion

The modular design of the software implementation ensures clarity, maintainability, and scalability. Each component of the smart home system has its dedicated module, with clear interfaces defined for communication between modules. This approach not only simplifies debugging and testing but also allows for easy expansion of the system in the future. By segregating functionalities into distinct modules, the system remains organized, and the integration of additional features or hardware becomes straightforward.

6 Testing

In this section, we describe the unit tests and integration tests performed on the components and the overall system to ensure that it operates correctly and meets the project requirements.

6.1 Unit Tests

6.1.1 First Relay

We tested the connections of the first relay by simulating user input from the mobile application using the two push buttons on the Tiva board (PF0 and PF4). When Input 1 (PF0) was pressed, we verified that the buzzer and fan, which were connected to the normally opened terminal of channel 1, were both activated. When Input 2 (PF4) was pressed, we confirmed that the plug, connected to the normally opened terminal of channel 2, was turned on.

6.1.2 Second Relay

We tested the second relay by setting up a two-way switch connected to a 220V supply. We connected the common terminal of the first channel to L1 and the common terminal of the second channel to L2. The normally closed terminal of the first channel was connected in parallel with the normally opened terminal of the second channel, and then connected to the positive terminal of the lamp. We simulated the user input from the mobile application using the push button at pin PF4 on the Tiva board and verified that the relay switched the lamp on and off as expected from both user input and the two-way switch.

6.1.3 Limit Switch

We tested the limit switch by connecting the normally closed terminal to pin PB2 on the Tiva board. We configured the red LED to turn on when pin PB2 received a GND signal. We verified that the LED turned on when the limit switch was pressed down (indicating the door was closed) and turned off when the switch was released (door open).

6.1.4 HC-05 Bluetooth Module

We tested the HC-05 Bluetooth module by pairing it with a mobile device and establishing a wireless connection. We connected the module to the Tiva board and used a Bluetooth terminal on the mobile device to send commands. We verified that the three LEDs on the Tiva board could be controlled using the commands sent from the Bluetooth terminal.

6.1.5 DHT11 Temperature Sensor

We tested the DHT11 temperature sensor by connecting the DATA pin to PC4 on the Tiva board. We configured the red LED to turn on when the temperature

reading from the sensor exceeded 30°C. We brought a lighter closer to the sensor to increase the temperature and verified that the LED turned on when the temperature exceeded the threshold.

6.2 Integration Testing

After performing the unit tests on the individual components, we integrated all the components together to test the overall system. We simulated user input from the mobile application using the push buttons on the Tiva board and verified that the fan, buzzer, and plug were controlled as expected. We also tested the door status detection using the limit switch and observed the red LED behavior. We used the Bluetooth terminal to send commands to the system and verified the the control of lamp and plug and the reading of temperature and door status. Finally, we tested the temperature monitoring and alarm functionality by bringing a lighter closer to the DHT11 sensor and observing the activation of the fan, and buzzer. All tests were successful, and the system operated as intended, demonstrating that the components were correctly integrated and that the system met the project requirements.

7 Conclusion

In this project, we successfully developed a smart home application using a Tiva C series microcontroller (TM4C123GH6PM) and various peripheral components. We designed and implemented a system capable of controlling a 220V lamp, plug, fan, and buzzer, as well as monitoring the door status and room temperature. The system can be controlled remotely via a mobile application using a Bluetooth connection.

We conducted a series of unit tests to verify the functionality of each component, including relays, limit switch, Bluetooth module, and temperature sensor. These tests confirmed that the individual components operated as expected. We then performed integration testing to ensure that the entire system worked correctly when all components were integrated. The results of the integration tests demonstrated that the components were correctly connected and that the system met the project requirements.

The successful completion of this project has provided us with valuable insights into the design and implementation of smart home applications. We have gained hands-on experience in interfacing microcontrollers with various peripheral components, communicating with sensors, and developing embedded software. We believe that the knowledge and skills acquired in this project will be beneficial for our future endeavors in the field of embedded systems and smart home technologies.

8 Future Work

This project lays the foundation for various improvements and extensions that could be pursued in the future. Some recommendations for future work include:

- Incorporating additional sensors, such as motion detectors, gas sensors, or smoke detectors, to enhance the system's capabilities.
- Implementing a more sophisticated user interface for the mobile application, allowing for better control and monitoring of the system.
- Integrating the system with a cloud platform to enable remote access and data logging for long-term analysis.
- Improving the robustness and reliability of the system by implementing error handling and fault tolerance mechanisms.

9 Acknowledgements

We would like to express our gratitude to our project supervisors, Eng. Mina Maher, Eng. Nouran Hamdy, and Eng. Mahmoud Ahmed, for their guidance and support throughout the project. We also thank our colleagues who provided valuable feedback and assistance. Finally, we are grateful to the technical staff and our families who helped us with the equipment and facilities required for our work.

10 References

1. DHT11 Datasheet, Aosong Electronics Co., Ltd.
2. TM4C123GH6PM Datasheet, Texas Instruments.
3. HC-05 Bluetooth Module User Manual, HC Corporation.
4. Relay Module User Manual, Relay Corporation.
5. "Embedded Systems: Introduction to ARM Cortex-M Microcontrollers," J. Valvano, 2014.