Smart Door Security And Home Automation System Using Arduino And Bluetooth Module

Chavan Yashodeep Laxman[1], More Ganesh Vishwanath[2], Joshi Chinmay Sanjay Rao[3], Birajdar Dhiraj Vijaykumar[4], Prof.KaziA.S.M[5]

[1, 2, 3, 4] Dept of Computer Engineering

[5]GuideLecturer, Dept of Computer Engineering

[1, 2, 3, 4, 5] Vishweshwarayya Abhiyantriki Padvika Mahavidyalaya, Almala, Latur, Maharashtra, India.

Abstract- *Nowadays, we have remote controls for us television sets and other electronic systems, which have made our lives really easy. Have you ever wondered about home automation which would give the facility of controlling tube lights, fans and other electrical appliances at home using a remote control? Off-course, yes! But, are the available options cost-effective? If the answer is No, we have found a solution to it.*

I. LITERATURE REVIEW

Designing an intuitive and user-friendly web-based interface or mobile application for controlling home appliances and devices remotely, considering factors such as ease of use, responsiveness, and security.

Developing a robust and reliable communication protocol between the Bluetooth, Arduino and home appliances/devices, ensuring seamless connectivity and efficient data exchange.

CONSTRUCTION

Designing a complete home automation system using Arduino and Bluetooth module involves several steps. Here is a brief overview of the system design and analysis:

1. System requirements: The first step is to determine the system requirements. This involves identifying the devices and appliances that need to be automated and the desired functionality of the system.

2. Hardware selection: Once the system requirements have been determined, the next step is to select the hardware components. This includes selecting the Arduino and Bluetooth module board, sensors, and other necessary components.

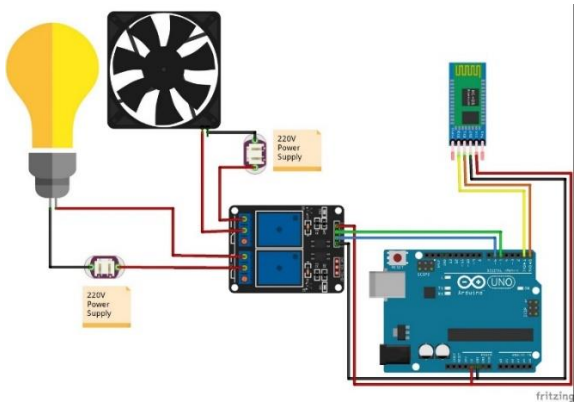
3. Circuit design: After selecting the hardware components, the next step is to design the circuit. This involves connecting the hardware components and designing the necessary interfaces.

4. Software development: Once the hardware and circuit design are complete, the next step is to develop the software. This involves writing the code for the Arduino and Bluetooth module board and other components.

5. Testing: After the software development is complete, the system needs to be tested. This involves verifying that the system functions properly and meets the system requirements. Implementation: Once testing is complete, the system can be implemented. This involves installing the hardware, setting up the software, and integrating the system with the existing infrastructure.

6. Maintenance: Regular maintenance is required to ensure that the system continues to function properly. This involves monitoring the system and making any necessary updates or repairs.

In terms of analysis, it is important to evaluate the system's performance and identify any areas for improvement. This can involve analyzing the system's energy efficiency, user- friendliness, and reliability. It is also important to consider the cost and scalability of the system, as well as any potential security or privacy concerns.



1.1 Arduino Uno

The Arduino UNO is a standard board of Arduino. Here UNO means 'one' in Italian. It was named as UNO to label the first release of Arduino Software. It was also the first USB board released by Arduino. It is considered as the powerful board used in various projects. Arduino.cc developed the Arduino UNO board.

ArduinoUNO is based on an ATmega328P microcontroller. It is easy to use compared to other boards, such as the Arduino Mega board, etc. The board consists of digital and analog Input/Output pins (I/O), shields, and other circuits.

The Arduino UNO includes 6 analog pin inputs, 14 digital pins, a USB connector, a power jack, and an ICSP (In-Circuit Serial Programming) header. It is programmed based on IDE, which stands for Integrated Development Environment. It can run on both online and offline platforms

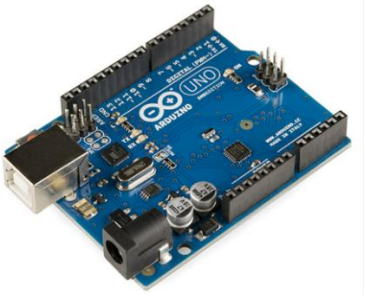


Figure 1.1: Arduino Uno

1.2 Relay Module

A relay is an electrically operated switch that can be turned on or off, letting the current go through or not, and can be controlled with low voltages, like the 5V provided by the Arduino pins. Controlling a relay module with the Arduino is as simple as controlling any other output as we'll see later on. This relay module has two channels (those blue cubes). There are other models with one, four and eight channels. This module should be powered with 5V, which is appropriate to use with an Arduino. There are other relay modules that are powered using 3.3V, which is ideal for ESP32, ESP8266, and other microcontrollers.

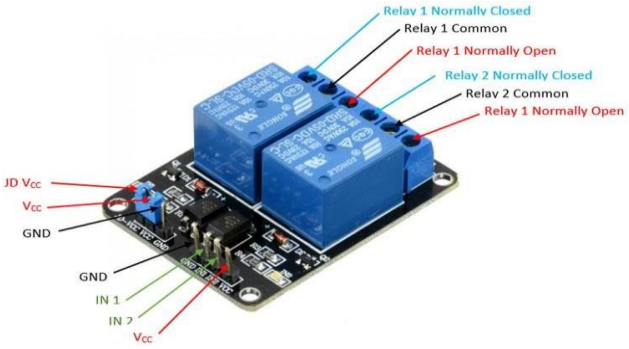


Figure1.2: Relay Module

1.3 Servo Motor

A servo motor is a type of motor that can rotate with great precision. Normally this type of motor consists of a control circuit that provides feedback on the current position of the motor shaft, this feedback allows the servo motors to rotate with great precision. If you want to rotate an object at some specific angles or distance, then you use a servo motor. It is just made up of a simple motor which runs through a servo mechanism.

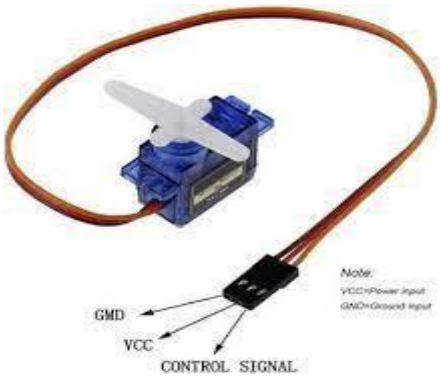


Figure 1.3: Servo Motor

1.4 Bluetooth HC-05

It is used for many applications like wireless headset, game controllers, wireless mouse, wireless keyboard, and many more consumer applications.

It has range up to <100m which depends upon transmitter and receiver, atmosphere, geographic & urban conditions.

It is IEEE 802.15.1 standardized protocol, through which one can build wireless Personal Area Network (PAN). It uses frequency-hopping spread spectrum (FHSS) radio technology to send data over air.

It uses serial communication to communicate with devices. It communicates with microcontroller using serial port (USART).

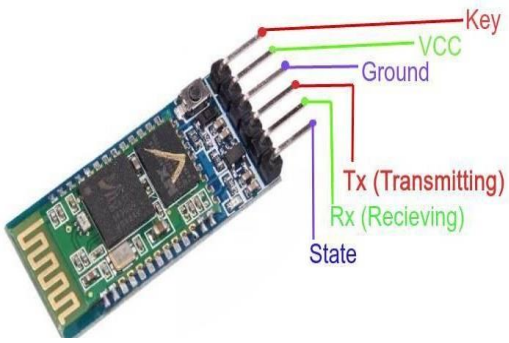


Figure 1.4: Bluetooth HC-05

1.5 Jumper Wires

A jump wire (also known as jumper, jumper wire, DuPont wire) is an electrical wire, or group of them in a cable, with a connector or pin at each end (or sometimes without them simply "tinned"), which is normally used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components, without soldering.

Individual jump wires are fitted by inserting their "end connectors" into the slots provided in a breadboard, the header connector of a circuit board, or a piece of test equipment.

There are different types of jumper wires. Some have the same type of electrical connecter at both ends, while others have different connectors. Some common connectors are:

1.5.1 **Solid tips–** are used to connect on/with a breadboard or female header connector. The arrangement of the elements and ease of insertion on a breadboard allows increasing the mounting density of both components and jump wires without fear of short-circuits. The jump wires vary in size and colour to distinguish the different working signals.

1.5.2 **Crocodile clips –** are used, among other applications, to temporarily bridge sensors, buttons and other elements of prototypes with components or equipment that have arbitrary connectors, wires, screw terminals, etc.

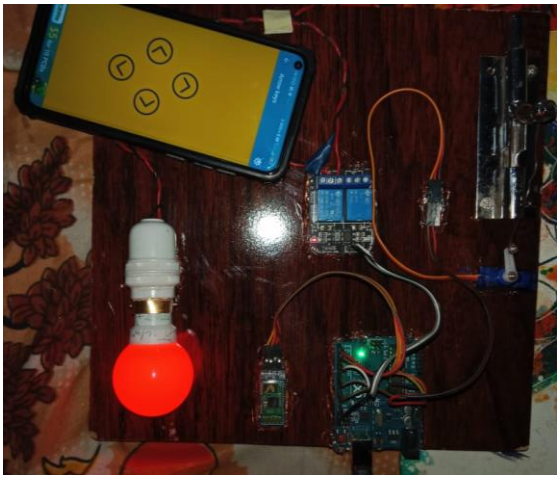
Figure1.5:JumperWire

SOURCE CODE

```
#include <Servo.h>
Servo my_servo;
char incoming_data;
int relay = 4;
String data;
void setup() {
Serial.begin(9600);
my_servo.attach(10);

my_servo.write(0);
pinMode(relay, OUTPUT);
digitalWrite(relay, LOW);
  delay(100);
  // pinMode(12,OUTPUT);
  // digitalWrite(12,HIGH);
pinMode(7, OUTPUT);
digitalWrite(7, HIGH);
}

void loop() {

  while (Serial.available()) {
    delay(10);
    // digitalWrite(relay, LOW);
    delay(10);

    char incoming_data = Serial.read();
    data += incoming_data;

    if (data == "ON") {
my_servo.write(180);
      delay(100);
    }
    if (data == "OFF") {
my_servo.write(0);
      delay(100);
```

```
    }
    if (data == "BULBN") {
digitalWrite(relay, HIGH);
      delay(100);
    }
    if (data == "BULBF") {
digitalWrite(relay, LOW);
      delay(100);
    }
  }
  data = "";
}
```

Program Uploading



II. RESULT AND CONCLUSION

1. Result

1.1 Convenience and Comfort: The system aims to provide convenient and comfortable control of home devices, allowing users to remotely manage and automate tasks such as turning lights on/off, controlling appliances, adjusting thermostat settings, and managing security features.

1.2 **Energy Efficiency:** Another aim of a Arduino and Bluetooth module-based home automation system is to optimize energy consumption and promote energy efficiency. The system may include features such as scheduling, automated sensors, and remote monitoring of energy usage, allowing users to effectively manage and reduce energy waste in their homes.

1.3 **Security and Safety:** Home automation systems using Arduino and Bluetooth module can also incorporate security and safety features, such as remote monitoring and control of security cameras, door locks, and alarms. The system aims to enhance the security of homes by providing real-time monitoring, alerts, and control of security devices, which can improve the safety and peace of mind of the residents.

1.4 **Customization and Flexibility:** The system may aim to provide customization and flexibility to users, allowing them to define their own automation rules, schedules, and preferences. This can enable personalized automation based on individual needs, preferences, and lifestyles, making the system adaptable to different users and home environments.

2. Conclusion

The home automation system using Arduino UNO is a low-cost and customizable solution that provides users with an easy and efficient way to control their home appliances remotely. In this project, we have successfully developed a home automation system that can be controlled through a mobile application and can turn appliances on and off based on the user's preferences.

Through the use of the Arduino UNO board, relay module, and Super Debug USB to Micro USB Cable wire, we were able to create a system that is highly scalable and customizable. The Arduino UNO board allowed us to connect the system to the internet, while the relay module allowed us to control the appliances. The Super Debug USB to Micro USB Cable wire facilitated the programming of the Arduino UNO board.

In addition, the use of the Blynk mobile application allowed for easy and convenient control of the home appliances from a distance. The application provided an intuitive and user- friendly interface that allowed users to monitor and control their appliances with ease.

Overall, this project has demonstrated the potential of low-cost and customizable home automation systems that can be easily deployed and controlled remotely. The system can be easily adapted and modified to fit the specific needs of the user, making it an ideal solution for a wide range of home automation applications.

As a conclusion, the development of this home automation system using Arduino UNO has provided us with valuable insights into the potential of this technology, and has demonstrated the importance of low-cost and customizable solutions in the home automation market. The project has successfully achieved its objectives and has provided a solid foundation for further development and optimization of this technology.

3. Future Scope

1. **Wireless Connectivity:** The system can be enhanced with wireless connectivity such as Bluetooth to allow for remote control.
2. **Multiple Appliances**: The system can be expanded by adding multiple appliances. This would stop the manually on off the Door or Home Appliances.
3. In order to increase the range and to monitor and control the security from anywhere in the world we will try to implement it using a Wi-Fi module (i.e., esp8266) . so that it will be more dependable and helpful also. we can also try to join a camera with it so that we can continuously keep track of our house.

REFERENCES

[1] ArduinoIDE:https://www.arduino.cc/en/software
[2] Servo_Motor:https://circuitdigest.com/article/servo-motor
[3] https://www.electronicwings.com/sensors-modules/bluetooth-module-hc-05-
[4] Relay Module: https://electrical-engineering-portal.com/connections-of-overcurrent-relay-part-2