

Rome Appliances Control Using Bluetooth for the Patients Temporary Got Disabled

Dr A V Senthil Kumar , K Kiran

Professor, Student, Student PG & Research, Department of Computer Application,
Hindusthan College of Arts & Science
Coimbatore-641028

Abstract: Automation or smart Room is a wireless smart controlling or operating of all room appliances like lights, fans, doors, alarms etc ... where all the devices connected to Bluetooth, are part of the Internet of Things. This thing helps people who can't able to walk to the switch to turn on things. We can monitor our homes from different places or different countries too. It is a basic microcontroller Arduino UNO connected to the internet via a Bluetooth module. Monitoring their Room and automation helps ease human efforts and maintain a comfortable life. Room automation in IOT is gaining more popularity nowadays, because of its many advantages. Everyone can achieve Room automation by simply connecting their electrical devices to the internet and it is simplicity and comparable affordability. It is used in a user-friendly manner using custom-defined portals. we can easily automate our home by using phones from anywhere

Keywords: Automation, Arduino uno, relays, Bluetooth Module.

1. Introduction

Home automation refers to the integration of technology into various aspects of household functions to automate and simplify tasks, enhance security, increase energy efficiency, and improve overall comfort and convenience for homeowners. This concept involves connecting devices and systems within a home to a central hub or network, allowing for remote monitoring and control via smartphones, tablets, or computers. Home automation allows homeowners to control various devices and systems with ease, whether they are at home or away. From adjusting the thermostat and lighting to locking doors and monitoring security cameras, automation provides convenience and peace of mind.

Arduino Uno is a microcontroller board that serves as the brain of the home automation system. It is highly versatile and can be programmed to interface with various sensors, actuators, and devices. Arduino Uno provides the processing power and flexibility needed to control different aspects of home automation. Bluetooth technology enables wireless communication between the Arduino Uno and other Bluetooth-enabled devices, such as smartphones, tablets, or computers. By integrating Bluetooth connectivity, homeowners can remotely monitor and control the home automation system from anywhere within the Bluetooth range.

One of the primary benefits of home automation is its ability to optimize energy usage. Smart thermostats can learn household patterns and adjust temperatures accordingly, while smart lighting systems can automatically turn off lights in unoccupied rooms. This not only reduces energy waste but also helps homeowners save money on utility bills. Automation technology can make homes more accessible for individuals with disabilities or mobility issues. As home automation becomes more prevalent, it is increasingly viewed as a desirable feature in real estate. Homes equipped with automation technology may command higher resale values and attract more potential buyers.

Home automation offers a myriad of benefits that contribute to a more efficient, secure, and comfortable living environment. As technology continues to advance and become more accessible, the adoption of home automation is expected to continue growing, reshaping the way people interact with and manage their homes.

1. What is Home Automation?

Home/office automation is the control of any or all Electrical devices in our home or office, whether we are there or away. Home/office automation is one of the most exciting developments in technology for the home that has come along in decades. There are hundreds of products available today that allow us control over the devices automatically, either by remote control or even by voice command.

Other Types of Home Automations

Apart from the home appliance control using computer and microcontroller which will be further discussed in this paper later, there are many other different ways by which we can control the home appliances and each and every system will have its own pros and cons.

2.1 Home Appliances Control Using Remote Control

The lights, fans can be automatically turned on/off with the help of a remote where there will be a sensor instead of going near to a switch board and putting on/off the switch. Companies like Legrand and Gold Medal already started these kinds of control system and they are at present available in the market.

2.2 Home Appliances Control Using DTMF

In this method, the control of home appliances can be done even though when we are elsewhere just by using the DTMF tone generated when the user pushes mobile phone

keypad buttons or when connected to a remote mobile.

3. Basics of Arduino Uno and Bluetooth Module

3.1. Arduino Uno

Arduino Uno is a microcontroller board that serves as the brain of the home automation system. It is highly versatile and can be programmed to interface with various sensors, actuators, and devices. Arduino Uno provides the processing power and flexibility needed to control different aspects of home automation. The Arduino Uno is a microcontroller board based on the ATmega328P microcontroller. It features digital and analog input/output pins that can be programmed to interact with various electronic components and devices.

Arduino Uno is an open-source platform, meaning its hardware and software specifications are freely available for anyone to use and modify. Arduino Uno is programmed using the Arduino IDE, a user-friendly development environment that simplifies the process of writing and uploading code to the board.: Arduino Uno is highly versatile and can be expanded with various shields and modules to add additional functionality, such as Ethernet connectivity, Wi-Fi, sensors, actuators, and more. Arduino Uno is widely used by beginners and hobbyists due to its ease of use, extensive documentation, and a large community of users and developers.



Arduino Uno can be powered through a USB connection or an external power supply. The voltage range is typically 7-12V. Arduino Uno is programmed using the Arduino IDE (Integrated Development Environment), which uses a simplified version of C++.

The board can be programmed using a USB cable, and programs are uploaded via the USB serial interface. Arduino Uno can be expanded using shields, which are additional boards that can be plugged into the top of the Arduino to add functionality like Ethernet, Wi-Fi, motor control, and more.

ATmega328P Microcontroller:

This is the brain of the Arduino Uno. It runs the code you upload to the board and controls all its functions.

Digital I/O Pins:

The Uno has 14 digital input/output pins (of which 6 can be used as PWM outputs), allowing you to interface with digital sensors, actuators, and other devices.

Analog Inputs:

There are 6 analog input pins on the Uno, which can be used to read analog signals from sensors like temperature sensors, light sensors, and potentiometers.

USB Connection:

The Uno can be powered and programmed via a USB connection to a computer. This makes it easy to upload code and power the board without needing an external power source.

Power Jack:

In addition to USB power, the Uno can also be powered using an external power supply connected to its power jack.

Reset Button:

This button allows you to reset the microcontroller, restarting your program from the beginning.

Clock Crystal:

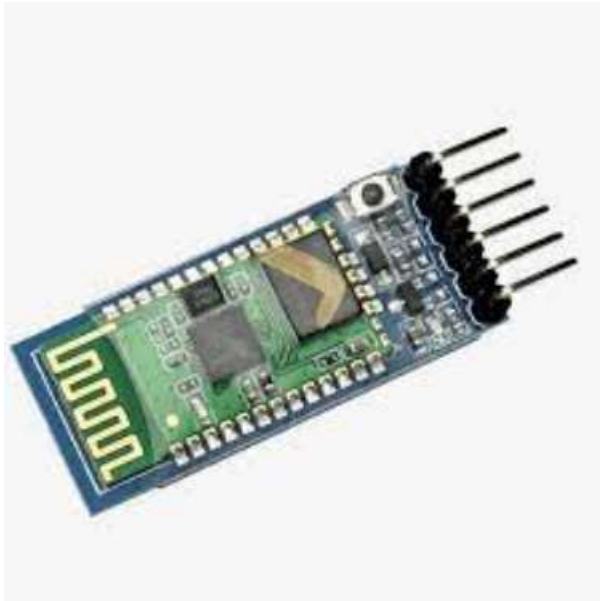
The Uno has a 16MHz crystal oscillator, providing the clock signal for timing operations.

Voltage Regulator:

The board contains a voltage regulator, allowing it to be powered with voltages between 7V and 12V, which are then regulated down to the 5V required by the microcontroller.

3.2. Bluetooth Module

Bluetooth technology enables wireless communication between the Arduino Uno and other Bluetooth-enabled devices, such as smartphones, tablets, or computers. By integrating Bluetooth connectivity, homeowners can remotely monitor and control the home automation system from anywhere within the Bluetooth range. Connect the Bluetooth module to the Arduino Uno using appropriate pins (e.g., UART pins for serial communication).



The HC-05 and HC-06 are common Bluetooth modules used with Arduino. They are capable of establishing wireless communication between the Arduino and other Bluetooth-enabled devices. Bluetooth modules use the Serial Communication Protocol (UART) to communicate with the Arduino. This simplifies the process of sending and receiving data wirelessly. Some Bluetooth modules can be configured using AT commands, allowing users to set parameters such as the device name, baud rate, and pairing code.

HC-05 modules often have a master/slave configuration. In master mode, they can connect to multiple slave devices, while in slave mode, they can connect to a single master. The module typically has a set of pins for VCC (power), GND (ground), TX (transmit), and RX (receive). It can be connected to the Arduino's digital pins to establish a serial communication link. Bluetooth modules can pair with other devices, like smartphones or computers, allowing for bidirectional communication.

4.Connecting Arduino Uno and Bluetooth Module:

Write Arduino code using the SoftwareSerial library to establish communication with the Bluetooth module. This allows the Arduino to send and receive data wirelessly. Connect the VCC and GND pins of the Bluetooth module to the 5V and GND pins on the Arduino, respectively.

Connecting an Arduino Uno to a Bluetooth module is a common task and can be done using various Bluetooth modules such as HC-05 or HC-06. Here's a general guide on how to connect an Arduino Uno to a Bluetooth module:

Connect the TX pin of the Bluetooth module to the RX pin of the Arduino and vice versa. Set the mode of the Bluetooth module (master/slave) based on your application.

4.1.Hardware Setup:

Bluetooth Module Pins:

Identify the pins on your Bluetooth module. Typically, they include VCC, GND, RX, and TX.

Arduino Uno Pins:

Identify the pins on your Arduino Uno board. You'll use digital pins for communication with the Bluetooth module.

Connection:

Connect the VCC pin of the Bluetooth module to the 5V pin on the Arduino Uno.

Connect the GND pin of the Bluetooth module to any GND pin on the Arduino Uno.

Connect the TX pin of the Bluetooth module to a digital pin (e.g., pin 2) on the Arduino Uno. This pin will receive data from the Bluetooth module.

Connect the RX pin of the Bluetooth module to a digital pin (e.g., pin 3) on the Arduino Uno. This pin will transmit data to the Bluetooth module.

4.2.Software Setup:

Arduino IDE:

Install the Arduino IDE on your computer if you haven't already.

Library Installation:

If you're using a specific Bluetooth module, there might be libraries available for it. Install the required libraries through the Arduino Library Manager.

Programming:

Write your Arduino code to communicate with the Bluetooth module. This code will typically involve initializing the serial communication, configuring the pins, and handling the data received/transmitted via Bluetooth.

Make sure to set the baud rate of the serial communication to match the baud rate of your Bluetooth module (usually 9600 or 57600 baud).

Example code might include initializing the serial port (`Serial.begin()`), reading data from the Bluetooth module (`Serial.read()`), and sending data to the Bluetooth module (`Serial.write()`).

Testing:

Upload the code to your Arduino Uno.

Power up your Arduino Uno.

Pair your Bluetooth module with a device (e.g., smartphone, computer).

Use a serial monitor in the Arduino IDE or a serial terminal program on your computer to send/receive data to/from the Arduino Uno via Bluetooth.

Ensure that the voltage levels are compatible between the Arduino Uno and the Bluetooth module. Most modules work with 5V logic, but some might require level shifting if operating at different voltage levels.

Double-check the datasheet or documentation of your specific Bluetooth module for any additional setup requirements or precautions.

Be mindful of the pin configurations and wiring to avoid damaging the components. Test your setup incrementally, starting with simple communication tasks, before moving on to more complex interactions.

Understanding Relays

Relays are electromechanical switches that are commonly used in various applications, including home automation. They play a crucial role in allowing low-power electronic devices, like microcontrollers such as the Arduino Uno, to control high-power electrical devices such as lights, fans, or appliances. Relays consist of a coil and one or more sets of contacts. When current flows through the coil, it generates a magnetic field that attracts or repels the contacts, causing them to open or close.

Relays typically have one set of normally open (NO) contacts and one set of normally closed (NC) contacts. When the relay is not energized, the NC contacts are closed, and the NO contacts are open. Relays can handle both AC (alternating current) and DC (direct current) loads, making them versatile for various applications.



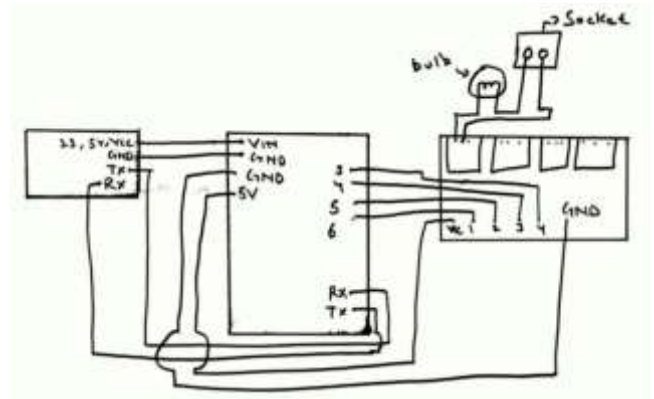
Relays provide electrical isolation between the low-voltage control circuit (such as Arduino Uno) and the high-voltage load circuit (such as lights or appliances). This isolation ensures safety and protects the control circuit from potential damage due to voltage spikes or surges. Many home appliances and devices operate on higher voltages and currents than microcontrollers like Arduino Uno can directly handle. Relays act as intermediaries, allowing these low-power control signals to switch high-power loads safely.

Relays can be used to control a wide range of devices, including lights, fans, heaters, air conditioners, and more. This versatility makes them essential components in home automation systems.

5. Setting Up the Hardware

5.1. Components Needed:

- Arduino Uno board
- Bluetooth module (such as HC-05 or HC-06)
- Relay module
- Jumper wires
- A power source (USB cable or external power supply)



5.2. Connect the Bluetooth Module to Arduino Uno:

- Connect the VCC pin of the Bluetooth module to the 5V pin of the Arduino Uno.
- Connect the GND pin of the Bluetooth module to the GND pin of the Arduino Uno.
- Connect the TX pin of the Bluetooth module to the RX pin (pin 0) of the Arduino Uno.
- Connect the RX pin of the Bluetooth module to the TX pin (pin 1) of the Arduino Uno.

5.3. Connect the Relay Module to Arduino Uno:

- Connect the VCC pin of the relay module to the 5V pin of the Arduino Uno.
- Connect the GND pin of the relay module to the GND pin of the Arduino Uno.
- Connect IN1, IN2, IN3, etc., pins of the relay module to the digital output pins of the Arduino Uno.

5.4. Power Supply:

Power the Arduino Uno either via a USB cable connected to your computer or using an external power supply connected to the DC barrel jack.

Pair the Bluetooth Module with a Device:

Use a smartphone or a computer with Bluetooth capability to pair with the Bluetooth module. Search for nearby Bluetooth devices and pair with the Bluetooth module using the default pairing code (usually '1234' or '0000').

Ensure that the relay module is properly powered and the relay pins are connected to the appropriate digital output pins of the Arduino Uno. Double-check the baud rate and communication settings for the Bluetooth module to ensure compatibility with your Arduino sketch. Test the system thoroughly to verify proper communication and relay control.

6. Programming Basics

Arduino programming involves writing code to control Arduino microcontroller boards. Arduino uses a simplified version of C and C++ programming languages, which makes it accessible to beginners and experienced programmers alike. Arduino programming allows you to interact with sensors, actuators, and other electronic components to create various projects such as robots, home automation systems, and IoT devices.

Setup() Function: This function is called once when the Arduino board starts. It is used for variables, setting pins, and performing necessary setup tasks.

Loop() Function: The loop function runs continuously after the setup function completes. It contains the main logic of your Arduino program, where you read sensors, process data, and control outputs.

Arduino supports various data types including integers, floats, characters, and strings. Variables hold data that can be manipulated within the program. Arduino programming includes control structures such as if statements, for loops, while loops, and switch-case statements to control the flow of the program.

Code

```
String inputs;
#define relay1 2 //Connect relay1 to pin 9
#define relay2 3 //Connect relay2 to pin 8
#define relay3 4 //Connect relay3 to pin 7
#define relay4 5 //Connect relay4 to pin 6
#define relay5 6 //Connect relay5 to pin 5
#define relay6 7 //Connect relay6 to pin 4
#define relay7 8 //Connect relay7 to pin 3
#define relay8 9 //Connect relay8 to pin 2
void setup()
{
  Serial.begin(9600); //Set rate for communicating with
  phone
  pinMode(relay1, OUTPUT); //Set relay1 as an output
  pinMode(relay2, OUTPUT); //Set relay2 as an output
  pinMode(relay3, OUTPUT); //Set relay1 as an output
  pinMode(relay4, OUTPUT); //Set relay2 as an output
  pinMode(relay5, OUTPUT); //Set relay1 as an output
  pinMode(relay6, OUTPUT); //Set relay2 as an output
  pinMode(relay7, OUTPUT); //Set relay1 as an output
  pinMode(relay8, OUTPUT); //Set relay2 as an output
  digitalWrite(relay1, LOW); //Switch relay1 off
  digitalWrite(relay2, LOW); //Switch relay2 off
  digitalWrite(relay3, LOW); //Switch relay1 off
  digitalWrite(relay4, LOW); //Switch relay2 off
  digitalWrite(relay5, LOW); //Switch relay1 off
```

```
digitalWrite(relay6, LOW); //Switch relay2 off
digitalWrite(relay7, LOW); //Switch relay1 off
digitalWrite(relay8, LOW); //Switch relay2 off
}
void loop()
{
  while(Serial.available()) //Check if there are available bytes
  to read
  {
    delay(10); //Delay to make it stable
    char c = Serial.read(); //Conduct a serial read
    if (c == '#'){
      break; //Stop the loop once # is detected after a word
    }
    inputs += c; //Means inputs = inputs + c
  }
  if (inputs.length() >0)
  {
    Serial.println(inputs);
    if(inputs == "A")
    {
      digitalWrite(relay1, LOW);
    }
    else if(inputs == "a")
    {
      digitalWrite(relay1, HIGH);
    }
    else if(inputs == "B")
    {
      digitalWrite(relay2, LOW);
    }
    else if(inputs == "b")
    {
      digitalWrite(relay2, HIGH);
    }
    else if(inputs == "C")
    {
      digitalWrite(relay3, LOW);
    }
    else if(inputs == "c")
    {
      digitalWrite(relay3, HIGH);
    }
    else if(inputs == "D")
    {
      digitalWrite(relay4, LOW);
    }
    else if(inputs == "d")
    {
      digitalWrite(relay4, HIGH);
    }
    else if(inputs == "E")
    {
      digitalWrite(relay5, LOW);
    }
    else if(inputs == "e")
    {
      digitalWrite(relay5, HIGH);
    }
    else if(inputs == "F")
    {
      digitalWrite(relay6, LOW);
    }
  }
}
```

```
else if(inputs == "f")
{
digitalWrite(relay6, HIGH);
}
else if(inputs == "G")
{
digitalWrite(relay7, LOW);
}
else if(inputs == "g")
{
digitalWrite(relay7, HIGH);
}
else if(inputs == "H")
{
digitalWrite(relay8, LOW);
}
else if(inputs == "h")
{
digitalWrite(relay8, HIGH);
}
inputs="";
}
}
```

7. Expanding Functionality

Relays serve as critical components in advanced home automation systems, enabling the control of multiple devices and the creation of complex scenarios. Here are some advanced applications of relays in home automation.

8. Time-Based Automation

Use relays in conjunction with timers or scheduling systems to automate tasks based on specific times of the day. For instance, you can schedule the lights to turn on/off at certain times, adjust the thermostat temperature, or activate security systems when you're away.

9. Environmental Control

Employ relays to integrate environmental sensors, such as temperature sensors, humidity sensors, and motion detectors. Based on sensor inputs, relays can activate or deactivate devices to maintain optimal environmental conditions or trigger security alerts.

10. Remote Monitoring and Control

Enable remote monitoring and control of your home automation system using mobile apps or web interfaces. With relays, you can remotely check the status of devices, adjust settings, and receive notifications on your smartphone or computer. Emergency

11. Response and Safety

Integrate relays with smoke detectors, carbon monoxide detectors, and other safety devices for automated emergency response. In the event of a fire or gas leak, relays can activate alarms, shut off gas supplies, and illuminate escape routes for enhanced safety.

12. Conclusion

home automation using Arduino Uno presents a versatile and accessible solution for enhancing comfort, convenience, and efficiency in modern living spaces. By leveraging the capabilities of Arduino Uno, coupled with a variety of sensors, actuators, and connectivity modules, homeowners can create customized automation systems tailored to their specific needs and preferences.

Throughout this exploration, we've seen how Arduino Uno serves as the central control hub, enabling communication and coordination between various devices and subsystems within the home. Whether it's controlling lighting systems, adjusting room temperature, monitoring security cameras, or managing energy usage, Arduino Uno offers a flexible platform for implementing a wide range of automation tasks. Moreover, the integration of Arduino Uno with advanced technologies such as voice assistants, cloud connectivity, and the Internet of Things (IoT) opens up new possibilities for remote access, data analytics, and intelligent automation. With voice-controlled operation, cloud-based management, and IoT integration, homeowners can enjoy seamless control over their smart home systems from anywhere in the world, using intuitive interfaces and voice commands.

In essence, home automation using Arduino Uno represents a journey towards a more efficient, comfortable, and sustainable way of living. By harnessing the power of technology to automate routine tasks, optimize energy usage, and enhance security, Arduino-based home automation systems offer tangible benefits in terms of convenience, cost savings, and peace of mind. As we continue to explore and innovate in this field, the possibilities for creating smarter, more interconnected homes are virtually limitless.

References

- [1] Hari Charan Tadimetil, Manas Pulipati1, Overview of Automation Systems and Home Appliances Control using PC and Microcontroller at International Journal of Science and Research (IJSR), India Online ISSN: 2319-7064 Volume 2 Issue 4, April 2013.
- [2] R.Piyare, M.Tazil, BLUETOOTH BASED HOME AUTOMATION SYSTEM USING CELL PHONE at 2011 IEEE 15th International Symposium on Consumer Electronics 978-1-61284-842 6/11/\$26.00©2011 IEEE.
- [3] Vinay sagar K, Kusuma S, Home Automation Using Internet of Things at International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 - 0056 Volume: 02 Issue: 03 | June-2015.
- [4] S. H. Husin, A. A. Ngahdiman, N. M. Z. Hashim, Y. Yusop, A.S Ja'afar, Home Electrical Appliances Smart System at International Journal of Computer Science and Mobile Computing, ISSN 2320-088XIJCSMC, Vol. 2, Issue. 9, September 2013

[5] Angel Deborah S, Vidhyalakshimi S, Naren T Kesh, Naveen Hariharan, Satish Palaniappan, Home Automation Systems - A Study at International Journal of Computer Applications (0975 – 8887) Volume 116 – No. 11, April 2015

[6] Arun Cyril Jose and Reza Malekian, Smart Home Automation Security: A Literature Review at Smart Computing Review, vol. 5, no. 4, August 2015 269 DOI: 10.6029/smarter.2015.04.004

[7] Dr Manoj Priyatham M, Home Automation using Bluetooth and IOT Modules, Vol 4, Issue 6, June 2017.

