A PROJECT ON HOME AUTOMATION USING BLUIETOOTH

A COURSE PROJECT REPORT

By

NITHYA SRE S S (RA2011003010542)
AKSHITA GUPTA (RA2011003010554)
J.CHETAN (RA2011003010556)

Under the guidance of

DR.P.MADHAVAN

In partial fulfilment for the Course

of

18CSC302J - COMPUTER NETWORKS

in <DEPARTMENT OF COMPUTING TECHNOLOGIES

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

Kattankulathur, Chenpalpattu District

NOVEMBER 2022

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this mini project report "**Home Automation using Bluetooth** " is the bonafide work of **Nithya Sre S S (RA2011003010542) , Akshita Gupta (RA2011003010554) and J Chetan (RA2011003010556)** who carried out the project work under my supervision.

SIGNATURE
Associate Professor
Department of Computing Technologies
SRM Institute of Science and Technology

ACKNOWLEDGEMENT

We express our heartfelt thanks to our honorable **Vice Chancellor Dr. C. MUTHAMIZHCHELVAN**, for being the beacon in all our endeavors.

We would like to express my warmth of gratitude to our **Registrar Dr. S. Ponnusamy,** for his encouragement.

We express our profound gratitude to our **Dean (College of Engineering and Technology) Dr. T. V.Gopal,** for bringing out novelty in all executions.

We would like to express my heartfelt thanks to Chairperson, School of Computing **Dr. Revathi Venkataraman,** for imparting confidence to complete my course project

We wish to express my sincere thanks to **Course Audit Professor Dr.Annapurani Panaiyappan, Professor and Head, Department of Networking and Communications** and **Course Coordinators** for their constant encouragement and support.

We are highly thankful to our my Course project Faculty **Dr.P Madhavan, Associate Proffesor , Department of Computing Technologies,** for his/her assistance, timely suggestion and guidance throughout the duration of this course project.

We extend my gratitude to our **HoD DR. M Pushplatha ,Department of Computing Technologies** and my Departmental colleagues for their support.

Finally, we thank our parents and friends near and dear ones who directly and indirectly contributed to the successful completion of our project. Above all, I thank the almighty for showering his blessings on me to complete my course project.

ABSTRACT

In this project, a home automation system is designed which can be controlled by any smartphone. The automation system connects with the smartphone through Bluetooth. The smart phone sends control signals to switch home appliances ON or OFF by an android app through Bluetooth interface.

The project is built on Arduino UNO and is used to control LEDs and four home appliances connected to the Arduino through relays. The Arduino board is interfaced to an HC-05 Bluetooth module to pair with the smart phone.

An app named "Bluetooth Terminal" is used on the smart phone which is capable of sending text strings to a paired device. Another app named "BT Voice Control for Android" can also be used on the smart phone. The BT Voice app takes voice commands in US English and transfers them as text strings to a paired device. Either of the app will pair with the home automation system through HC-05 Bluetooth Module. Every module has a unique MAC address and a password for pairing with other devices. Like the Bluetooth module used in this project had a MAC address – 98:D3:31:F4:18:22 and had a password "1234" for pairing with other Bluetooth devices.

The Arduino board receives the user commands in the form of numbers from the smart phone through Bluetooth interface. These numbers are assigned to the home appliances and the appliances are toggled either ON or OFF on receiving the numeric command. The Arduino sketch looks for the numeric commands from the Bluetooth module and operates relays to switch appliances.

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Scenario Description

Home automation involves mixing different electrical components to automate actions or devices around your house, including:

Sensors: This is a component which monitors different parts of the environment (such as temperature, light levels and video/cameras). They collect valuable data which allows the home automation device to know how they need to act.

Controllers: A centralized device which receives data and other inputs, and 'works out' what needs to be do – i.e. perform an action. This is a bit like the motherboard (and associated hardware) within a computer that receives inputs – from the keyboard and mouse- and performs 'outputs'/actions – such as performing a calculation and showing something on the screen.

Actuators: This is a fancy word for something which allows access, movement or control of a device, such as a light switch, motor and valve.

A home automated device can then be made smart by hooking it up to devices like smartphones via Bluetooth or most commonly the internet…

Internet connected: an automated device can contain computer components which allow it to make internet calls to a central location, via TCP/IP and probably HTTP (similar to what happens when you access a website with your web browser).

This will involve the device transmitting data about itself to 'the cloud', but also allows it to accept actions to perform – such as 'turn up the heating', 'close the garage door' or 'turn on power' in a smart plug.

User interface: now that the device is 'available' on the internet, it needs some way of displaying its data – and allowing it to be controlled.

This is usually done via a smartphone app (for example to control smart thermostats or smart lightbulbs), although some smart devices also have a website-based control system (such as indoor and outdoor security cameras).

2. LITERATURE SURVEY

In Bluetooth based home automation system the home appliances are connected to the Arduino BT board at input output ports using relay. The program of Arduino BT board is based on high level interactive C language of microcontrollers; the connection is made via Bluetooth. The password protection is provided so only authorized user is allowed to access the appliances. The Bluetooth connection is established between Arduino BT board and phone for wireless communication. In this system the python script is used and it can install on any of the Symbian OS environment, it is portable. One circuit is designed and implemented for receiving the feedback from the phone, which indicate the status of the device. This system uses mobiles or computers to control basic home control and function automatically through internet from anywhere around the world globally, an automated home is sometimes called a smart home. It is meant to save the electric power and human energy. The proposed system is a distributed home automation system, consists of server i.e. Bluetooth module, sensors. Server controls and monitors the various sensors, and can be easily configured to handle more hardware interface module (sensors). The Arduino board, with built in Wi-Fi module acts as web server. Automation System can be accessed from the web browser of any local PC using server IP, or remotely from any PC or mobile handheld device connected to the internet with appropriate web browser through server real IP (internet IP). Bluetooth technology is selected to be the network infrastructure that connects server and the sensors. Bluetooth is chosen to improve system security (by using secure Bluetooth connection), and to increase system mobility and scalability.

3. REQUIREMENTS

3.1 Requirement Analysis

A smart home is defined by four key aspects: a communication network, which connects sensors (and devices); intelligent controls for system management, and smart features that respond to user-, sensor- or system (data) input. Devices in such a smart home can be controlled, accessed and monitored remotely.

3.2 Hardware Requirement

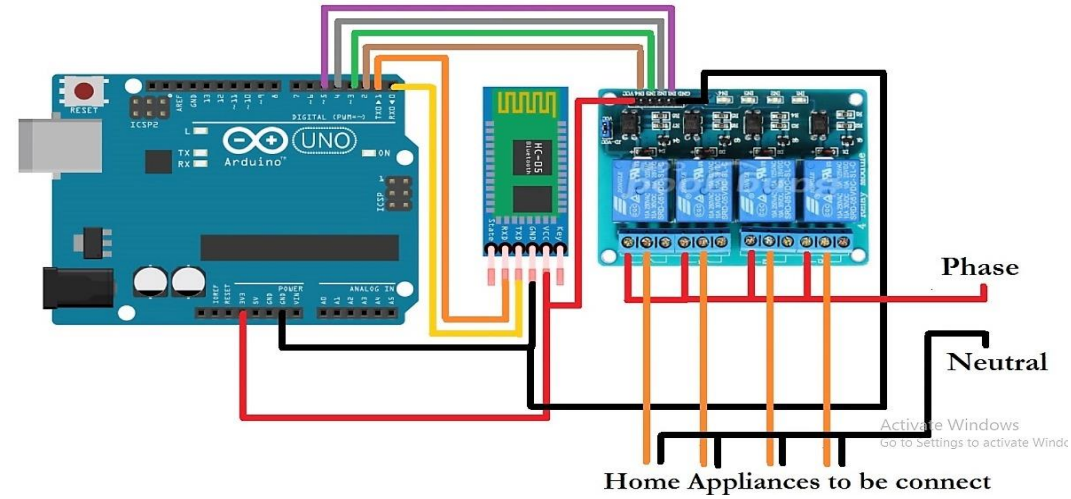
- Arduino Uno R3
- HC_05 Bluetooth Module
- LED
-
- Resistor
- LDR
-
- Breadboard
- Relay Module

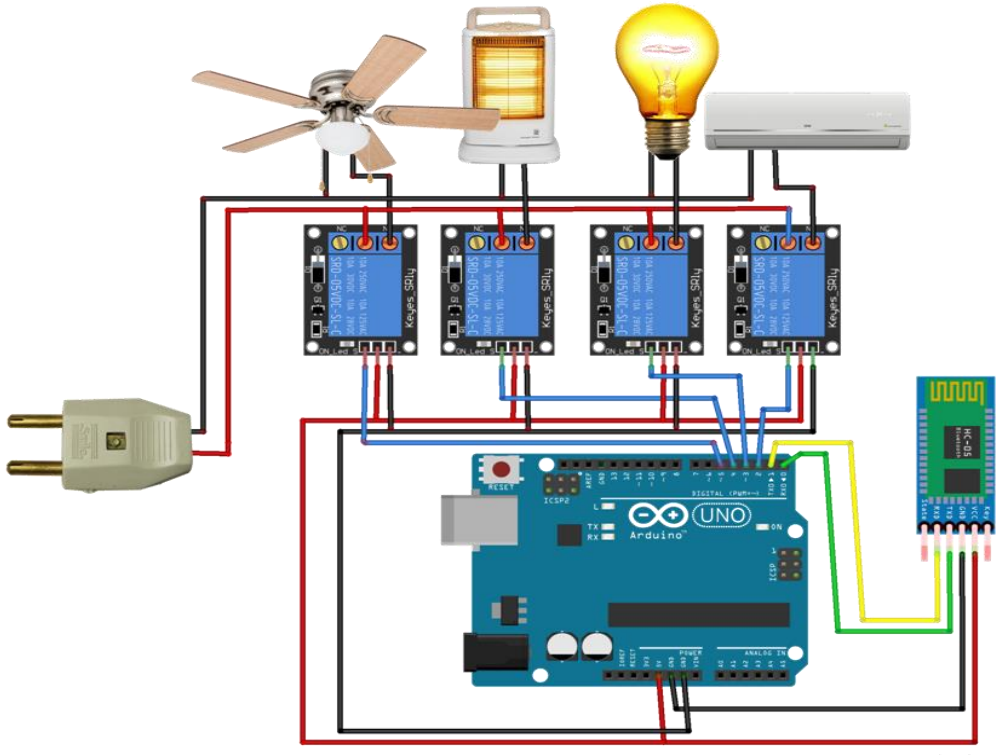
3.3 Software Requirement

- Arduino

4. ARCHITECTURE AND DESIGN

4.1 CIRCUIT DIAGRAM

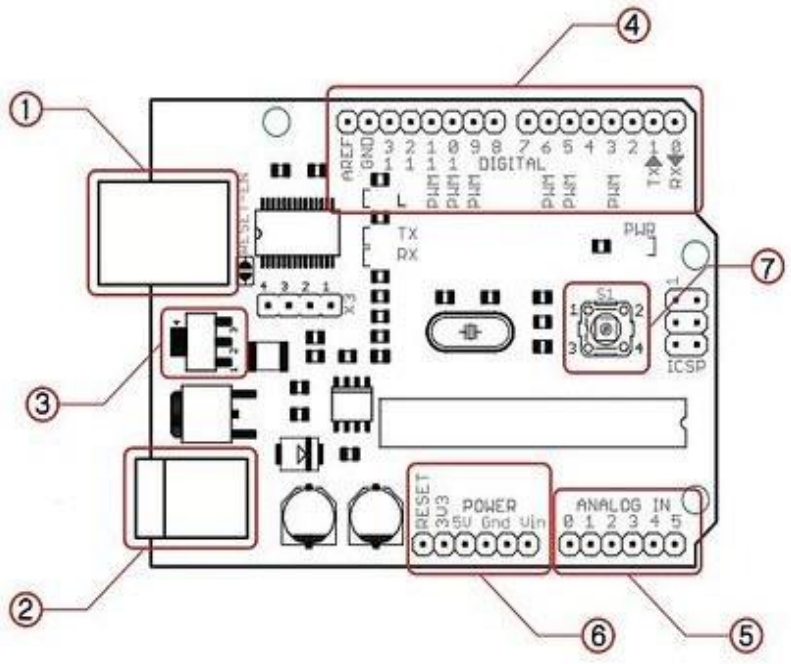




- It is used for many applications like wireless headset, game controllers, wireless mouse, wireless keyboard and many more consumer applications.
- It has range up to <100m which depends upon transmitter and receiver, atmosphere, geographic & urban conditions.
- It is IEEE 802.15.1 standardized protocol, 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).

ARDUINO UNO

Arduino board is a one type of microcontroller based kit. The first Arduino technology was developed in the year 2005 by David Cuartillas and Massimo Banzi. The designers thought to provide easy and low cost board for students, hobbyists and professionals to build devices. Arduino board can be purchased from the seller or directly we can make at home using various basic components. The best examples of Arduino for beginners and hobbyists includes motor detectors and thermostats, and simple robots. Arduino technology is used in many operating devices like communication or controlling.



The most important parts on the Arduino board high lighted in red:

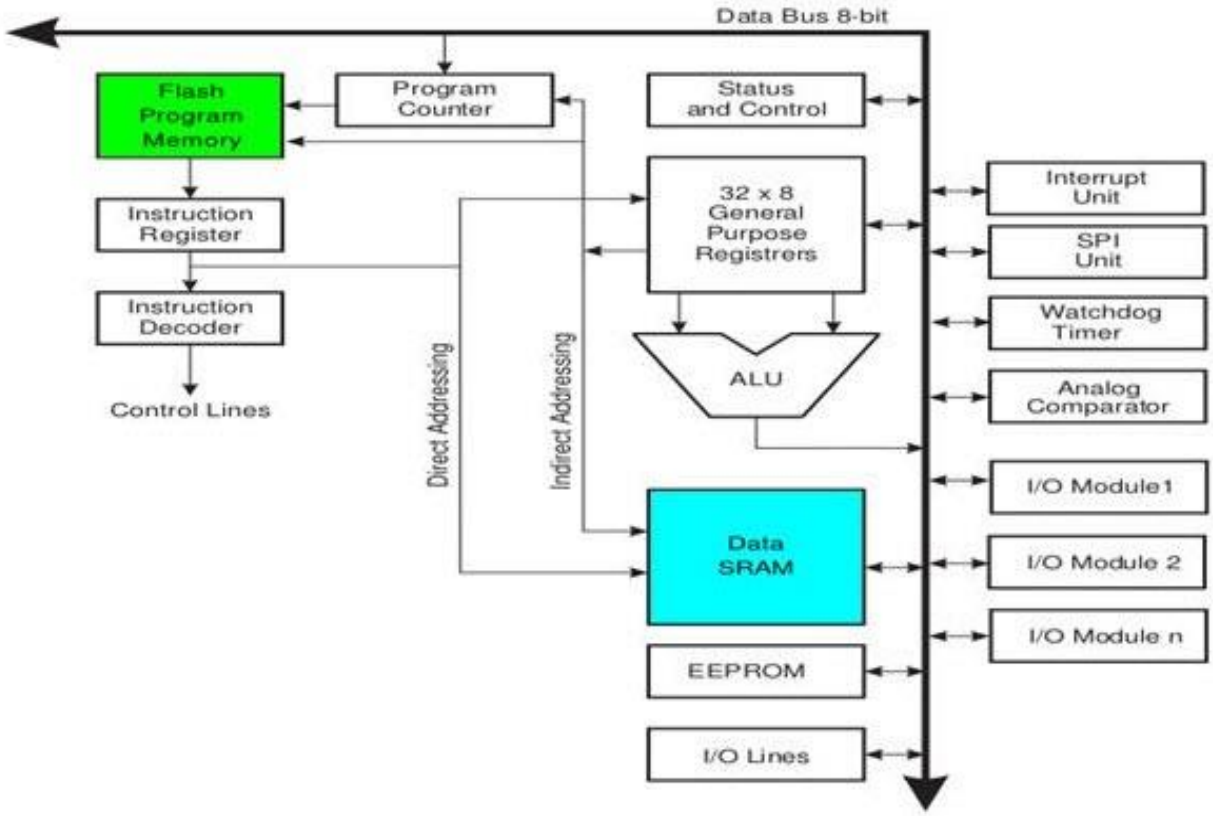
1: USB connector
2: Power connector
3: Automatic power switch
4: Digital pins
5: Analog pins
6: Power pins
7: Reset switch

The pin configuration of the Arduino Uno board is shown in the above. It consists of 14-digital i/o pins. Wherein 6 pins are used as pulse width modulation o/ps and 6 analog i/ps, a USB connection, a power jack, a 16MHz crystal oscillator, a reset button, and an ICSP header. Arduino board can be powered either from the personal computer through a USB or external source like a battery or an adaptor. This board can operate with an external supply of 7-12V by giving voltage reference through the IORef pin or through the pin Vin.

Arduino Architecture

Basically, the processor of the Arduino board uses the Harvard architecture where the program code and program data have separate memory. It consists of two memories such as program memory and

data memory. Wherein the data is stored in data memory and the code is stored in the flash program memory. The Atmega328 microcontroller has 32kb of flash memory, 2kb of SRAM 1kb of EPROM and operates with a 16MHz clock speed.



HC-05 Bluetooth Module

- HC-05 is a Bluetooth module which is designed for wireless communication. This module can be used in a master or slave configuration.
- Bluetooth serial modules allow all serial enabled devices to communicate with each other using Bluetooth.
- It has 6 pins,
- 1. **Key/EN:** It is used to bring Bluetooth module in AT commands mode. If Key/EN pin is set to high, then this module will work in command mode. Otherwise by default it is in data mode. The default baud rate of HC-05 in command mode is 38400bps and 9600 in data mode.
- HC-05 module has two modes,
- 1. **Data mode:** Exchange of data between devices.
- 2. **Command mode:** It uses AT commands which are used to change setting of HC-05. To send these commands to module serial (USART) port is used.

- 2. **VCC:** Connect 5 V or 3.3 V to this Pin.
- 3. **GND:** Ground Pin of module.
- 4. **TXD:** Transmit Serial data (wirelessly received data by Bluetooth module transmitted out serially on TXD pin)
- 5. **RXD:** Receive data serially (received data will be transmitted wirelessly by Bluetooth module).
- 6. **State:** It tells whether module is connected or not.

HC-05 module Information

- HC-05 has red LED which indicates connection status, whether the Bluetooth is connected or not. Before connecting to HC-05 module this red LED blinks continuously in a periodic manner. When it gets connected to any other Bluetooth device, its blinking slows down to two seconds.
- This module works on 3.3 V. We can connect 5V supply voltage as well since the module has on board 5 to 3.3 V regulator.
- As HC-05 Bluetooth module has 3.3 V level for RX/TX and microcontroller can detect 3.3 V level, so, no need to shift transmit level of HC-05 module. But we need to shift the transmit voltage level from microcontroller to RX of HC-05 module.

RELAY MODULE

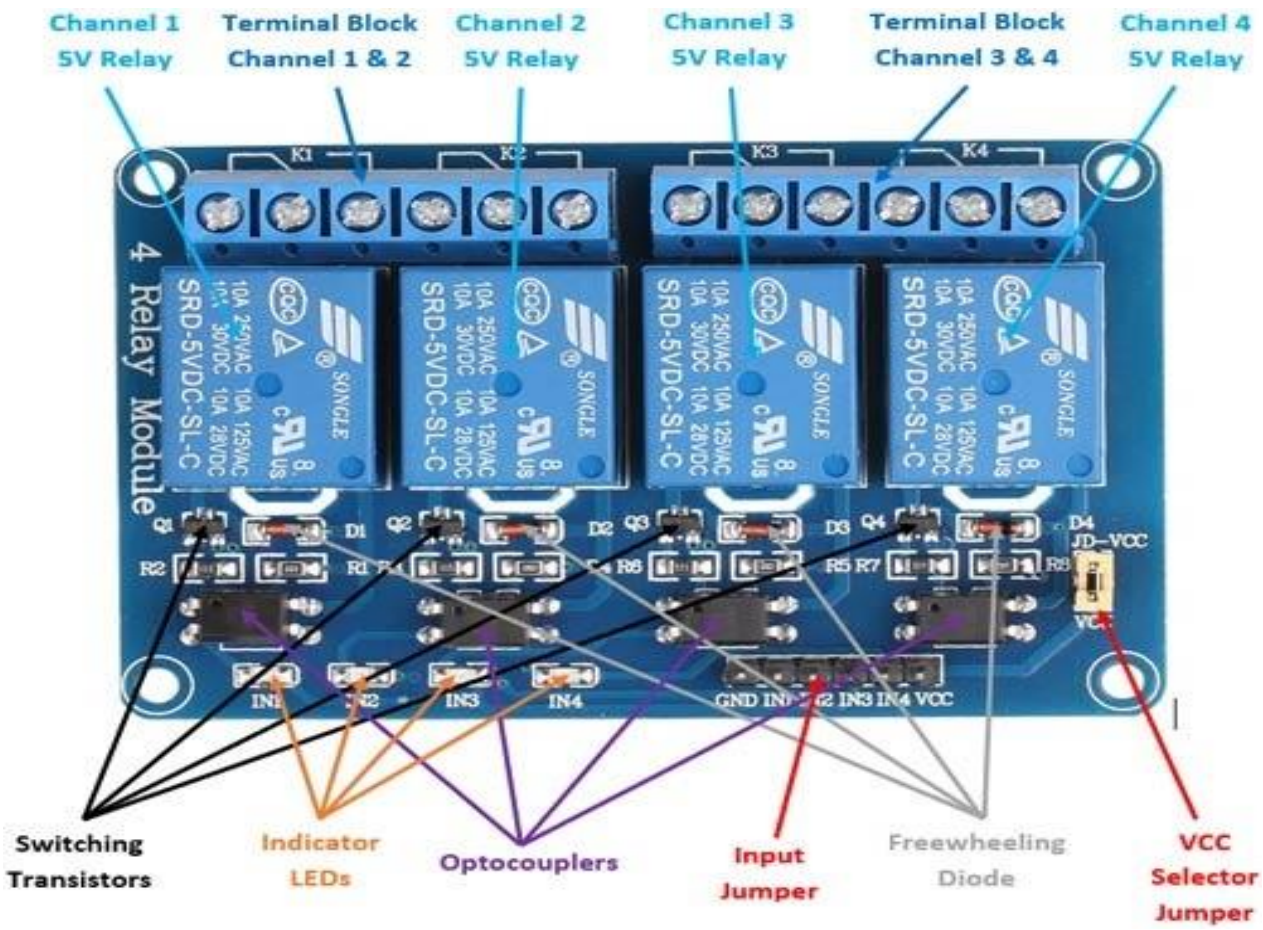
The four-channel relay module contains four 5V relays and the associated switching and isolating components, which makes interfacing with a microcontroller or sensor easy with minimum components and connections. There are two terminal blocks with six terminals each, and each block is shared by two relays. The terminals are screw type, which makes connections to mains wiring easy and changeable.The four relays on the module are rated for 5V, which means the relay is activated when there is approximately 5V across the coil.The indicator LEDs glow when the coil of the respective relay is energized, indicating that the relay is active. The optocouplers form an additional layer of isolation between the load being switched and the inputs.

Four-Channel Relay Module Specifications

Supply voltage – 3.75V to 6V

Trigger current – 5mA

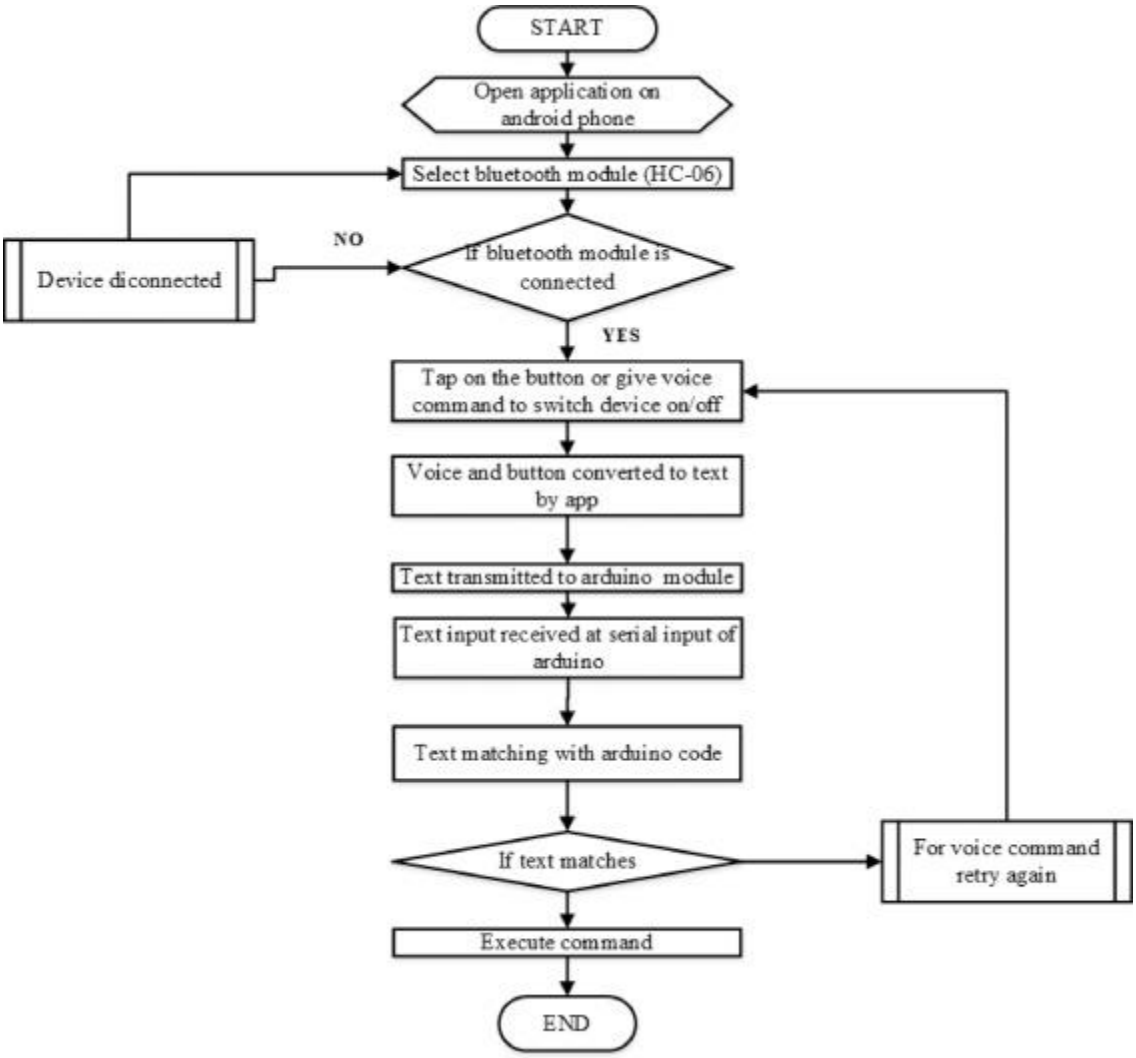
Current when the relay is active - ~70mA (single), ~300mA (all four)



RELAY ARCHITECTURE

The driver circuit for this relay module is slightly different compared to traditional relay driving circuits since there is an optional additional layer of isolation. When the jumper is shorted, the relay and the input share the same VCC, and when it is open, a separate power supply must be provided to the JD-VCC jumper to power the relay coil and optocoupler output. The inputs for this module are active low, meaning that the relay is activated when the signal on the input header is low. This is because the indicator LED and the input of the optocoupler are connected in series to the VCC pin on one end, so the other end must be connected to the ground to enable the current flow.

5. IMPLEMENTATION



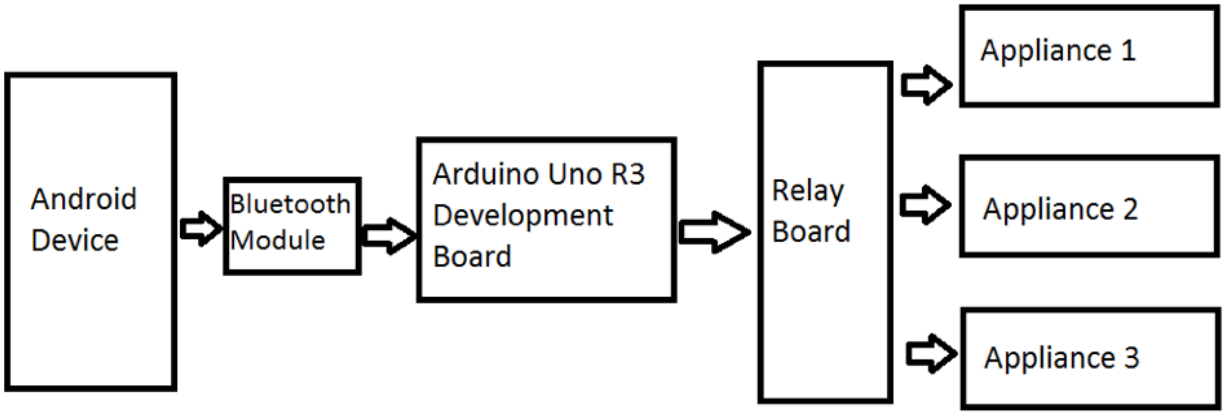
Home automation involves a network of interconnected smart devices communicating via Bluetooth and the internet. Each device uses mechanisms and sensors to automate mundane tasks around your house, allowing you to control them via your smartphone.

Some home automation tasks can be achieved right now for relatively little cost, but some are expensive – whilst others are still a pipe dream. But even automation a few repetitive tasks around your house is worth doing, so read on to learn all that you need to know.

Home automation is a wireless home appliance control system accessed by a remote device such as mobile phone (Android or iOS) to allow a home owner to control, monitor and coordinate home appliances , without changing the home infrastructure. Home automation systems are a technological means of intelligent monitoring, control, feedbacks, and actions of home appliances according to the needs of the home occupants.

Home automation system is use of information technologies and control system to reduce the human labor. The rapid growth of technologies influences us to use smartphones to remotely control the home appliances. An automated devices has ability to work with versatility, diligence and with lowest error rate.

The idea of home automation system is a significant issue for Researchers and home appliances companies. Automation system not only helps to decrease the human labor but it also saves time and energy. Early home automation systems were used in labor saving machines but nowadays its main objective is provided facilities to elderly and handicapped people to perform their daily routine tasks and control the home appliances remotely.



CODE FOR CONNECTING TO AURDINO UNO

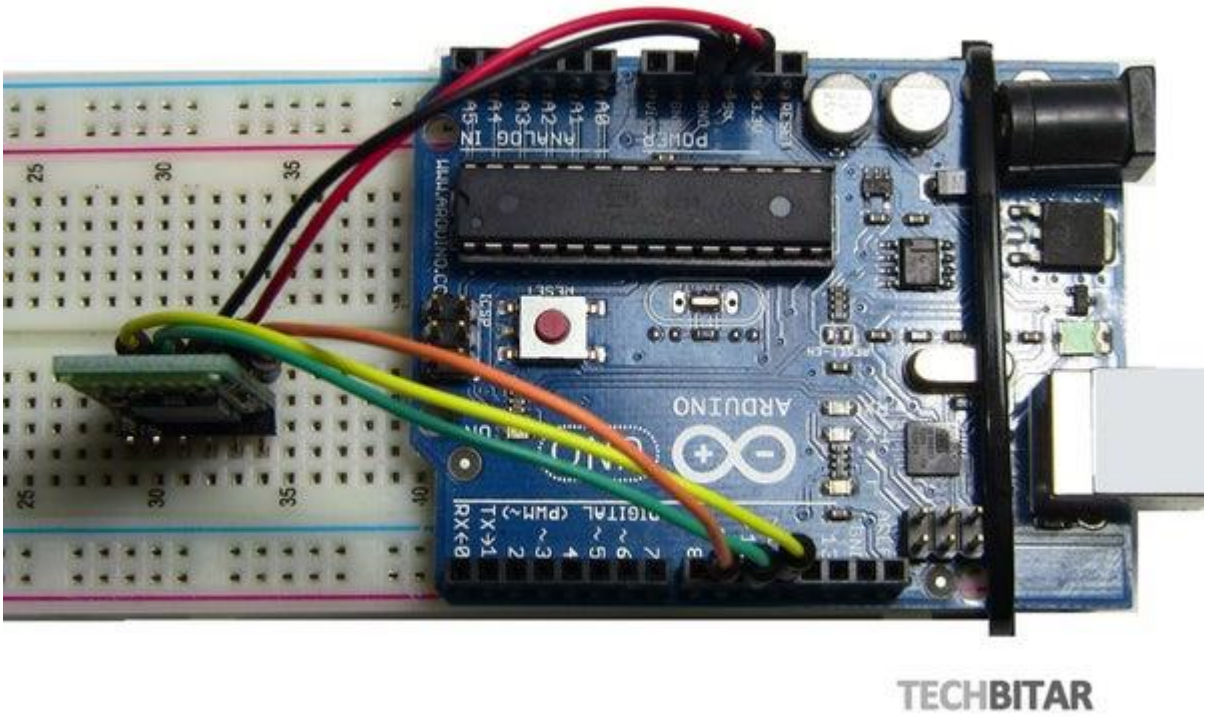
```
#include <EEPROM.h>
#include <IRremote.h>
#include <SoftwareSerial.h>
SoftwareSerial mySerial(2, 3); // RX, TX D2, D3
#define RelayPin1 4  //D4
#define RelayPin2 5  //D5
#define RelayPin3 6  //D6
#define RelayPin4 7  //D7
#define SwitchPin1 10  //D10
#define SwitchPin2 11  //D11
#define SwitchPin3 12  //D12
#define SwitchPin4 13  //D13
#define IR_RECV_PIN A0  //A0
//Update the HEX code of IR Remote 0x<HEX CODE>
#define IR_Button_1   0x80BF49B6
#define IR_Button_2   0x80BFC936
#define IR_Button_3   0x80BF33CC
#define IR_Button_4   0x80BF718E
#define IR_All_Off    0x80BF3BC4
#define IR_All_On     0x80BFF10E
IRrecv irrecv(IR_RECV_PIN);
decode_results results;
// Switch State
bool SwitchState_1 = LOW;
bool SwitchState_2 = LOW;
bool SwitchState_3 = LOW;
bool SwitchState_4 = LOW;
char bt_data; // variable for storing bluetooth data
void relayOnOff(int relay){
 switch(relay){
     case 1:
          digitalWrite(RelayPin1, !digitalRead(RelayPin1)); // change state for relay-1
          EEPROM.update(0,digitalRead(RelayPin1));
          delay(100);
     break;
     case 2:
          digitalWrite(RelayPin2, !digitalRead(RelayPin2)); // change state for relay-2
          EEPROM.update(1,digitalRead(RelayPin2));
          delay(100);
     break;
     case 3:
          digitalWrite(RelayPin3, !digitalRead(RelayPin3)); // change state for relay-3
          EEPROM.update(2,digitalRead(RelayPin3));
          delay(100);
     break;
     case 4:
          digitalWrite(RelayPin4, !digitalRead(RelayPin4)); // change state for relay-4
          EEPROM.update(3,digitalRead(RelayPin4));
          delay(100);
     break;
```

```
     default : break;
     }
}

void eepromState()
{
  digitalWrite(RelayPin1, EEPROM.read(0)); delay(200);
  digitalWrite(RelayPin2, EEPROM.read(1)); delay(200);
  digitalWrite(RelayPin3, EEPROM.read(2)); delay(200);
  digitalWrite(RelayPin4, EEPROM.read(3)); delay(200);
}

void bluetooth_control()
{
  if(mySerial.available()) {
    bt_data = mySerial.read();
    Serial.println(bt_data);
    switch(bt_data)
        {
         case 'A': digitalWrite(RelayPin1, LOW);  EEPROM.update(0,LOW); break; // if 'A'
received Turn on Relay1
         case 'a': digitalWrite(RelayPin1, HIGH); EEPROM.update(0,HIGH); break; // if 'a' received
Turn off Relay1
         case 'B': digitalWrite(RelayPin2, LOW);  EEPROM.update(1,LOW); break; // if 'B'
received Turn on Relay2
         case 'b': digitalWrite(RelayPin2, HIGH); EEPROM.update(1,HIGH); break; // if 'b'
received Turn off Relay2
         case 'C': digitalWrite(RelayPin3, LOW);  EEPROM.update(2,LOW); break; // if 'C'
received Turn on Relay3
         case 'c': digitalWrite(RelayPin3, HIGH); EEPROM.update(2,HIGH); break; // if 'c' received
Turn off Relay3
         case 'D': digitalWrite(RelayPin4, LOW);  EEPROM.update(3,LOW); break; // if 'D'
received Turn on Relay4
```

Steps modify the HC-05 Bluetooth Module Defaults Using AT Commands

The HC-05 Bluetooth module and its siblings are by far the most popular and inexpensive Bluetooth modules used for RF communications by microcontroller hackers. It costs less than $10 on ebay and it's easy to implement. I have published two guides based on the HC-05 Bluetooth module. The first guide explains how to use the HC-05 with the Arduino. The second is an Android app that simplifies controlling Arduino from your smart phone over Bluetooth using the HC-05. In both cases, the default settings for the HC-05 were fine.

Step 1: Components & Wiring



PARTS

- HC-05 Bluetooth module (Bluetooth over serial)
- Arduino Uno R3
- Breadboard & jumper wires
- Arduino IDE

WIRING

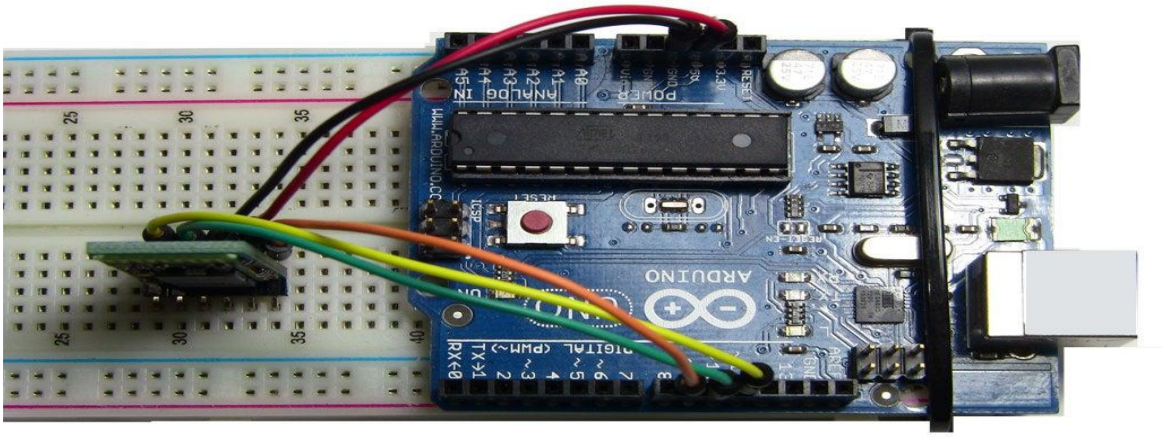
- HC-05 GND --- Arduino GND Pin
- HC-05 VCC (5V) --- Arduino 5V
- HC-05 TX --- Arduino Pin 10 (soft RX)
- HC-05 RX --- Arduino Pin11 (soft TX)
- HC-05 Key (PIN 34) --- Arduino Pin 9

Step 2: The Arduino Code for HC-05 Command Mode

This Arduino program (HC_05.ino) does two things. It takes the AT commands you enter from the Arduino IDE Serial Monitor and sends those commands to the HC-05. The program then reads the output of the HC-05 and displays it on the Arduino IDE Serial Monitor. You can also use a terminal emulator such as Tera Term instead of the Arduino Serial Monitor.

The Arduino communicates with the HC-05 using the Software Serial ports while the Arduino communicates with the user via the Serial Monitor

Steps3: To Switch the HC-05 Into Command Mode



For the HC-05 module to switch to AT command mode, the HC-05 pin 34 (often referred to as the Key pin) needs to pulled HIGH but in a certain order of events explained below. When the HC-05 enters the AT command mode, it will communicate at 38400 baud rate. Follow these steps in the stated order to switch to the HC-05 to AT command mode.

1. Wire the HC-05 and Arduino Uno per instructions.
2. BEFORE YOU CONNECT THE ARDUINO TO THE USB remove the VCC (power) red wire from the HC-05 so it's not getting any power from the Arduino. All other wires are still connected.
3. Now connect the Arduino Uno to the USB cable extended from your PC.
4. Make sure the HC-05 module is NOT PAIRED with any other Bluetooth device.
5. Re-connect the Arduino Uno 5V wire to the HC-05's VCC (5V power) pin.
6. The HC-05 LED will blink on and off at about 2 second intervals. Now the HC-05 is in AT command mode ready to accept commands to change configuration and settings.
7. To test if everything is wired correctly, open the Serial Monitor from the Arduino IDE and type "AT" and click SEND. You should see an "OK"
8. If you don't see an "OK" check your wiring.

CODE FOR CONFIGURING HC-05 TO COMMAND MODE

```
#include <SoftwareSerial.h>

SoftwareSerial BTSerial(10, 11);   // RX | TX

void setup() {
  pinMode(9, OUTPUT);    /* this pin will pull the HC-05 pin 34 (KEY pin) HIGH to switch
module to AT mode */
  digitalWrite(9, HIGH);

  Serial.begin(38400);
  Serial.println("Enter AT Commands:");
  BTSerial.begin(38400);  // HC-05 default speed in AT command mode
}

void loop() {

//The code below allows for commands and messages to be sent from COMPUTER (serial
monitor) -> HC-05
  if (Serial.available())           // Keep reading from Arduino Serial Monitor
    BTSerial.write(Serial.read());  // and send to HC-05

//The code below allows for commands and messages to be sent from HC-05 -> COMPUTER
(serial monitor)
  if (BTSerial.available())         // Keep reading from HC-05 and send to Arduino
    Serial.write(BTSerial.read());  // Serial Monitor
}
```

6.RESULTS AND DISCUSSION

In this project we have introduced design and implementation of a low cost, flexible and wireless solution to the home automation. The system is secured for access from any user or intruder. The users are expected to acquire pairing password for the Arduino BT and the cell phone to access the home appliances. This adds a protection from unauthorized users. This system can be used as a test bed for any appliances that requires on-off switching applications without any internet connection.

7.CONCLUSION AND FUTURE ENHANCEMENT

An Arduino based home automation system using Bluetooth and an android application with voice command has been designed and implemented. The Home automation system used an Android application and a Bluetooth technology in the design; this is because they are easy to use, fast, readily available, and reliable in communications between the remote user and devices. A low cost and highly reliable home automation system that can assist handicapped/old aged people, as well as a user-friendly device was developed. Other features can be added in the future such as biometrics so that unauthorised persons can not have access to the appliances and an also timing schedule can developed for each appliances connected this will effectively conserve energy.

REFERENCES

- https://www.instructables.com/Modify-The-HC-05-Bluetooth-Module-Defaults-Using-A/
- https://www.electronicwings.com/sensors-modules/bluetooth-module-hc-05-
- https://www.sciencedirect.com/science/article/pii/S2468227621000156
- https://www.instructables.com/Modify-The-HC-05-Bluetooth-Module-Defaults-Using-A/
- https://www.researchgate.net/publication/313672507_A_Novel_Home_Automation_System_using_Bluetooth_and_Arduino