HOME AUTOMATION AND SECURITY USING MICROCONTROLLERS

A Project Report

submitted in partial fulfillment of the requirements for the degree

of

Bachelor of Technology

in

Electrical Engineering

by

- Topojeet Chakraborty 12001616026
- Sayak Das 12001616053
- Rupam Mondal 12001616061
- Nishant 12001616086
- Kollol Dhar 12001616104
- Akash Kumar 12001616135

Under the Guidance of

Prof. MOU DAS MAHAPATRA

Department of Electrical Engineering

Dr. B. C. Roy Engineering College, Durgapur

HOME AUTOMATION AND SECURITY USING MICROCONTROLLERS

Completed

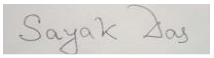
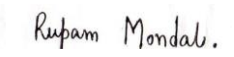
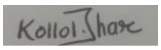
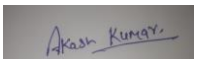
Under the Guidance of

Prof. MOU DAS MAHAPATRA

Department of Electrical Engineering

Dr. B C Roy Engineering College, Durgapur

Project Report Submitted By:

Name	**Roll No.**	**Signature**
Topojeet Chakraborty	12001616026	
Sayak Das	12001616053	
Rupam Mondal	12001616061	
Nishant	12001616086	
Kollol Dhar	12001616104	
Akash Kumar	12001616135	

DR. B. C. ROY ENGINEERING COLLEGE, DURGAPUR

CERTIFICATION

This is to certify that the project work entitled "**Home Automation and Security Using Microcontrollers**" being submitted by Topojeet Chakraborty (12001616026), Sayak Das (12001616053), Rupam Mondal (12001616061), Nishant (12001616086), Kollol Dhar (12001616104), Akash Kumar (12001616135) to Dr. B.C. Roy Engineering College for the award of the degree of Bachelor of Technology in Electrical Engineering in fulfillment of final year project under my supervision and guidance during the academic session of 2019-20.

Mou Das Mahapatra

07.07.2020

…….

Prof. Mou Das Mahapatra

(AP, EE Department)

07.07.2020

…….

Dr. Susanta Dutta

(HOD, EE Department)

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to several individuals and organizations for supporting me throughout my Graduate study. First, I wish to express my sincere gratitude to my supervisor, Prof. Mou Das Mahapatra, for her enthusiasm, patience, insightful comments, helpful information, practical advice and unceasing ideas that have always helped me tremendously in my project and writing of this thesis. Her immense knowledge, profound experience and professional expertise in Home Automation and Security Using Microcontrollers has enabled me to complete this project successfully. Without her support and guidance, this project would not have been possible. I could not have imagined having a better supervisor in our project.

I also wish to express my sincere thanks to Dr. B.C. Roy Engineering College for accepting me into the graduate program. In addition, this has enabled me to complete my Graduate studies successfully. Also, I and grateful the Faculty of Electrical Engineering.

Topojeet Chakraborty (12001616026)
Sayak Das (12001616053)
Rupam Mondal (12001616061)
Nishant (12001616086)
Kollol Dhar (12001616104)
Akash Kumar (12001616135)

ABSTRACT

Digitization has become the part of human lives. With the advancement of technology, the lives of people have become easier. One such example can be cited as, in the year 2015, 199 million users in India used smart phones but with years progressing the number has risen to 401 million users in the year 2020. Not only smart phones various devices such as Amazon Echo and Google Home which are being incorporated to our lives. And these numbers are expected to keep on rising. Hence, why not try to digitize the way we control our houses. An idea of having control of all our appliances right at the tip of our fingers can become a big topic in the near future because of its potential. Switches would not have to be externally fitted to the wall. We would not leave the security of our houses to a mere lock and key rather we could install an entire system along with these methods to take care our security.

At present many technologies are there which have automated houses such as temperature control units, leak detection, smoke detection, etc. But integrating all these into one single unit would help in making an easily accessible automation system. Put together a security system with it such as motion detectors and heat sensors, houses can be transformed into smart homes. Different type of home automation systems offers a wide range of functions and services, some of the common features are appliance control, thermostat control, remote control lighting, live video surveillance, monitor security camera, and real time text alerts.

INDEX

	Title	Page No.
1.	**Introduction**	**1**
2.	**Home automation and security alarm circuit using Arduino**	**2-4**
	2.1 Bluetooth Module	2
	2.2 Relay Module	3
	2.3 Arduino Uno R3	3-4
3.	**Components list**	**4-11**
	3.1 Specification of the components	4-7
	3.1.1 Arduino Uno R3 Development Board	4-5
	3.1.2 Bluetooth Module (HC-05)	6
	3.1.2a Hardware features	6
	3.1.2b Software features	6
	3.1.2c Pin Description	7
	3.2 Relays	7-8
	3.2.1 Principle of relay operation	8-9
	3.2.2 Pin Description	10
	3.2.3 Features	10
	3.2.4 Experiment Principle	10-11
4.	**How to connect HC-05 Bluetooth module**	**11-12**
	4.1 Uno: Hardware and software required	11
	4.2 Hardware Connection	11
	4.3 Block Diagram	12
5.	**Connection, scheme diagram and source code**	**12-20**
	5.1 Connection	12
	5.2 Schematic diagram	13
	5.3 Source code	13-16
	5.4 Simulations, Results and Discussions	17-20
6.	**Security alarm circuit using PIR motion sensor**	**21**
7.	**Components list**	**21-26**
	7.1 PIR sensor	21
	7.1.1 Features	21
	7.1.2 Key Specifications	22

7.1.3 Working of PIR sensor 22-23

7.2 Buzzer 23
7.3 GSM Module 23
7.3.1 SIM900A GSM module features 24
7.3.2 How to use SIM900A 24-26
7.4 Switch 26
8. Connection, circuit diagram and source code 26-29
8.1 Connection 26
8.2 Objectives of the PIR sensor 27
8.3 Circuit diagram 27
8.4 Source code 28-29
8.5 Simulations, Results and Discussions 30-32
9. Conclusion 33
10. Future reference 34
11. Reference 35

LIST OF FIGURES

Figures	Page No.
Figure 1: Bluetooth Module Switch	**2**
Figure 2: Relay	**3**
Figure 3: Arduino Development Board	**7**
Figure 4: Bluetooth Module	**8**
Figure 5: Relay Module	**8**
Figure 6: Overall view of the relay modules	**8**
Figure 7: Equivalent circuit showing the relay connections	**9**
Figure 8: Connection between the Bluetooth module and Arduino Board	**11**
Figure 9: Block diagram of the Automation System	**12**
Figure 10: Complete circuit connection of the Automation system	**13**
Figure 11: Successful compilation of code	**16**
Figure 12: The switch is off at this moment	**17**
Figure 13: When Voltage is LOW	**17**
Figure 14: Relay is not activated	**18**
Figure 15: When switch is on	**18**
Figure 16: When voltage is High	**19**
Figure 17: Relay is activated	**19**
Figure 18: Now the appliance is turned off	**20**
Figure 19: Working of a PIR sensor	**22**
Figure 20: PIR sensor	**23**
Figure 21: SIM900A Module	**24**
Figure 22: SIM900A Module Top view	**25**
Figure 23: Complete circuit diagram of the Security System	**27**
Figure 24: Successful compilation of code	**29**
Figure 25: When the object is out of PIR's range	**30**
Figure 26: When the object is in PIR's range	**30**
Figure 27: When the signal is low no message is displayed on the LCD showing there is no intrusion	**31**
Figure 28: When signal is high the message is displayed as "Security alert!"	**31**

LIST OF TABLES

Tables **Page No.**

Table 1: Specification of Arduino Development Board **4-5**
Table 2: Connection between the relay and development board **10-11**
Table 3: Relay Voltage level **20**
Table 4: Voltage level at buzzer and message status **32**

LIST OF ABBREVIATIONS

DCS: Distributed Control System

EEPROM: Electrically Erasable Programmable Read-Only Memory

EGSM: Extended Global System for Mobile communications

GND: Ground

GPRS: General Packet Radio Service

GSM: Global System for Mobile Communications

ICSP: In Circuit Serial Programming

IDE: Integrated Development Environment

LED: Light Emitting Diode

MCU: Microcontroller

PIR: Passive Infrared Sensor

RXD: Receive Data

SPP: Serial Port Protocol

SRAM: Static Random-Access Memory

TXD: Transmit Data

UART: Universal Asynchronous Receiver-Transmitter

USB: Universal Serial Bus

VCC: Voltage Common Collector

1. INTRODUCTION

Home automation system achieved great popularity in the last decades and it increases the comfort and quality of life. In this paper an overview of current and emerging home automation systems is discussed. Nowadays most home automation systems consist of a smartphone and microcontroller. A smart phone application is used to control and monitor the home appliances using different type of communication techniques. home automation more accurately describes homes in which nearly everything such as lights, appliances, electrical outlets, heating and cooling systems are hooked up to a remotely controllable network. A remotely accessible environment is an environment in which each appliance can be remotely accessed and controlled using software as an interface, which includes an Android application and a Web application. Such remotely accessible systems are already available in the market, but have a number of drawbacks as well. Some systems are described as an enabling system that can be used to provide a common framework for home Automation. It provides a system for a smart home that includes facilities such as a system controller, house-wide wiring and a common interface. This enables using the existing system for home automation

From a home security perspective, this also includes your alarm system, and all of the doors, windows, locks, smoke detectors, surveillance cameras and any other sensors that are linked to it.

2. HOME AUTOMATION AND SECURITY ALARM CIRCUIT USING A PIR MOTION SENSOR: -

The home automation circuit is built around an Arduino Uno board, Bluetooth module HC-05 and a 3-channel relay board. The number of channels depends on the number of appliances you wish to control. Arduino Uno is powered with a 12V DC adaptor/power source. The relay module and Bluetooth module can be, in turn, powered using a board power supply of Arduino Uno.

2.1 BLUETOOTH MODULE: -

Bluetooth module used in this project is HC-05, which supports master and slave mode serial communication (9600-115200 bps) SPP and UART interface. Using these features, it can communicate with other Bluetooth-enabled devices like mobile phones, tablets and laptops. The module runs on 3.3V to 5V power supply.

The HC-05 has two operating modes, one is the Data mode in which it can send and receive data from other Bluetooth devices and the other is the AT Command mode where the default device settings can be changed. We can operate the device in either of these two modes by using the key pin as explained in the pin description.

It is very easy to pair the HC-05 module with microcontrollers because it operates using the Serial Port Protocol (SPP). Simply power the module with +5V and connect the Rx pin of the module to the Tx of MCU and Tx pin of module to Rx of MCU as shown in the figure below

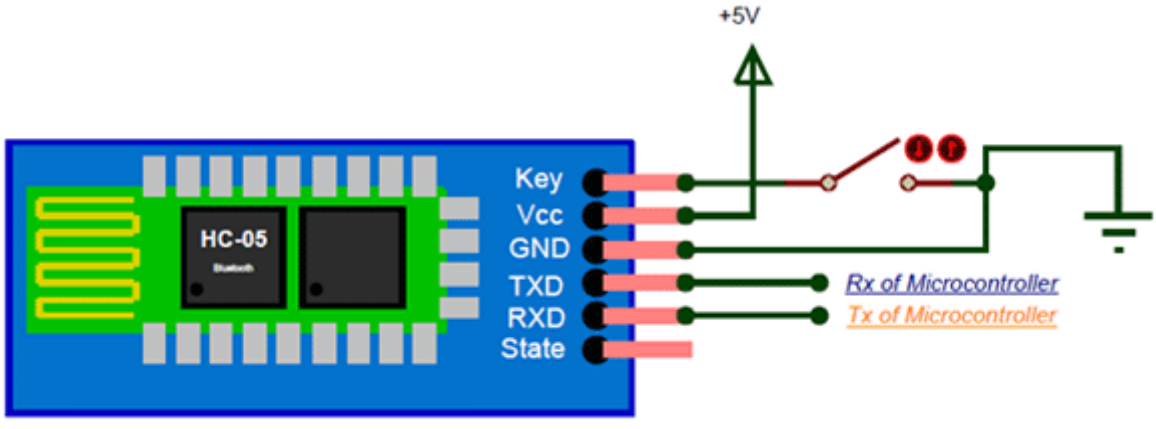


Figure 1: Bluetooth module switch

2.2 RELAY MODULE: -

A relay allows you to turn on or turn off a circuit using voltage and/or current much higher than what Arduino could handle. Relay provides complete isolation between the low-voltage circuit on Arduino side and the high-voltage side controlling the load. It gets activated using 5V from Arduino, which, in turn, controls electrical appliances like fans, lights and air-conditioners.

The relay's switch connections are usually labelled COM(POLE), NC and NO:

COM/POLE= Common, NC and NO always connect to this, it is the moving part of the switch.

NC = Normally Closed, COM/POLE is connected to this when the relay coil is not magnetized.

NO = Normally Open, COM/POLE is connected to this when the relay coil is MAGNETIZED and vice versa.

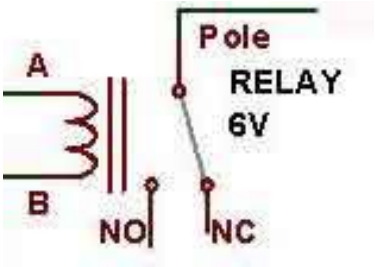


Figure 2: Relay

There are 5 Pins in a relay. Two pins A and B are two ends of a coil that are kept inside the relay. The coil is wound on a small rod that gets magnetized whenever current passes through it.

COM/POLE is always connected to NC (Normally connected) pin. As current is passed through the coil A, B, the pole gets connected to NO (Normally Open) pin of the relay.

2.3 ARDUINO UNO R3: -

Arduino is an open source electronics prototyping platform based on flexible, easy- to- use hardware and software. It is intended for artists, designers, hobbyists and anyone interested in creating interactive objects or environments. Arduino Uno is based on ATmega328 microcontroller (MCU). It consists of 14 digital input/output pins, six analogue inputs, a USB connection for programming the onboard MCU, a power jack, an ICSP header and a reset button. It is operated with a 16MHz crystal oscillator and contains everything needed to support the MCU. It is very easy to use as you simply need to connect it to a computer using a USB cable, or power it with an AC-to-DC adapter or battery to get started. The MCU onboard is programmed in Arduino programming language using Arduino IDE.

The 14 digital input/output pins can be used as input or output pins by using pinMode(), digitalRead() and digital Write() functions in Arduino programming. Each pin operates at 5V and can provide or receive a maximum of 40mA current and has an internal pull-up resistor of 20-50 KOhms which are disconnected by default. Out of these 14 pins, some pins have specific functions as listed below:

- Serial Pins 0 (Rx) and 1 (Tx): Rx and Tx pins are used to receive and transmit TTL serial data. They relate to the corresponding ATmega328P USB to TTL serial chip.
- External Interrupt Pins 2 and 3: These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.
- PWM Pins 3, 5, 6, 9 and 11: These pins provide an 8-bit PWM output by using analogWrite() function.
- SPI Pins 10 (SS), 11 (MOSI), 12 (MISO) and 13 (SCK): These pins are used for SPI communication.
- In-built LED Pin 13: This pin is connected with an built-in LED, when pin 13 is HIGH – LED is on and when pin 13 is LOW, its off.

Along with 14 Digital pins, there are 6 analog input pins, each of which provide 10 bits of resolution, i.e. 1024 different values. They measure from 0 to 5 volts, but this limit can be increased by using AREF pin with analog Reference () function.

- Analog pin 4 (SDA) and pin 5 (SCA) also used for TWI communication using Wire library.
- AREF: Used to provide reference voltage for analog inputs with analogReference() function.
- Reset Pin: Making this pin LOW, resets the microcontroller.

3. COMPONENTS LIST: -

- Arduino Uno R3 Development Board
- Bluetooth Module (HC-05)
- Android Device (Preferably one running v4.0 or above)
- Relays (R1-R4 in Schematic diagram)
- Connectors
- USB cable for Arduino
- Bread Board

3.1 SPECIFICATION OF THE COMPONENTS: -

3.1.1. ARDUINO UNO R3 DEVELOPMENT BOARD:

Table 1: Specification of Arduino Development Board

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6

Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by boot loader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
LED_BUILTIN	13
Length	68.6 mm
Width	53.4 mm
Weight	25 gm

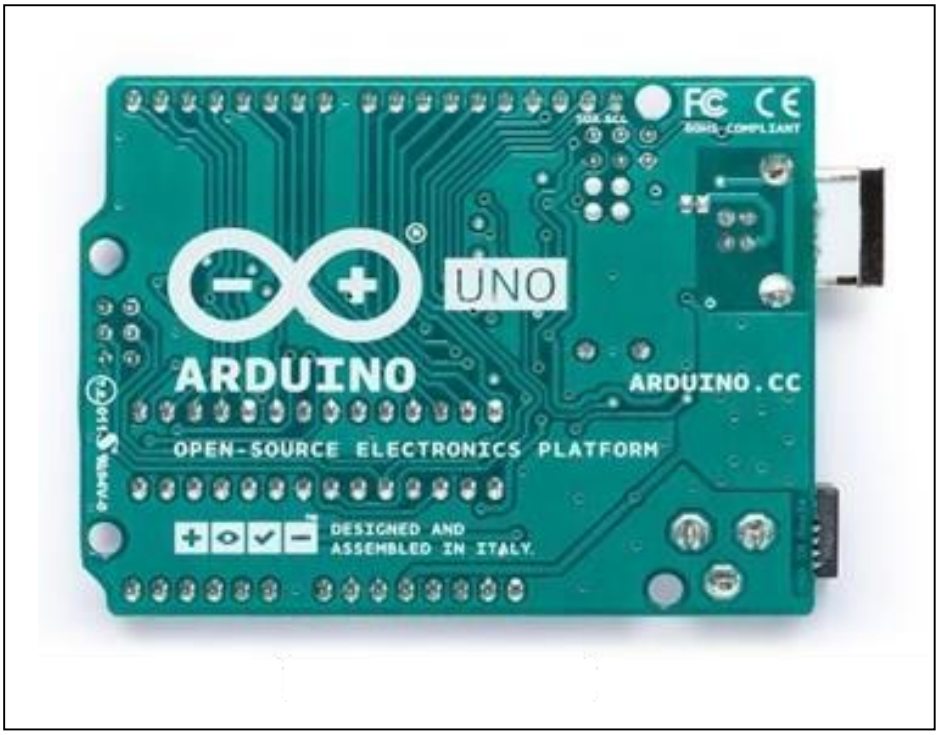


Figure 3: Arduino Development Board

3.1.2 BLUETOOTH MODULE (HC-05): -

The Bluetooth module HC-05 is a MASTER/SLAVE module. By default, the factory setting is SLAVE. The Role of the module (Master or Slave) can be configured only by AT COMMANDS. The slave modules cannot initiate a connection to another Bluetooth device, but can accept connections. Master module can initiate a connection to other devices. The user can use it simply for a serial port replacement to establish connection between MCU and GPS, PC to our embedded project.

3.1.2 a. Hardware Features: -

- Typical -80dBm sensitivity.
- Up to +4dBm RF transmits power.
- 3.3 to 5 V I/O.
- PIO (Programmable Input/Output) control.
- UART interface with programmable baud rate.
- With integrated antenna.
- With edge connector.

3.1.2 b. Software Features: -

- Slave default Baud rate: 9600, Data bits: 8, Stop bit: 1, Parity: No parity.
- Auto-connect to the last device on power as default.
- Permit pairing device to connect as default.
- Auto-pairing PINCODE:"1234" as default.

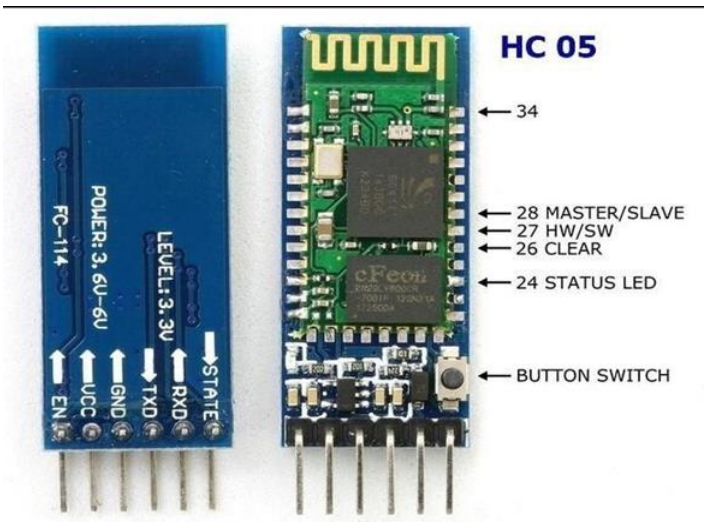


Figure 4: Bluetooth Module

3.1.2 c. Pin Description: -

The HC-05 Bluetooth Module has 6 pins. They are as follows:

Enable: -

When enable is pulled LOW, the module is disabled which means the module will not turn on and it fails to communicate. When enable is left open or connected to 3.3V, the module is enabled i.e. the module remains on and communication also takes place.

V_{CC}: - Supply Voltage 3.3V to 5V

G_{ND}: - Ground pin

T_{XD} & **R_{XD}**: - These two pins acts as an UART interface for communication

State: -

It acts as a status indicator. When the module is not connected to / paired with any other Bluetooth device, signal goes Low. At this low state, the led flashes continuously which denotes that the module is not paired with other device. When this module is connected to/paired with any other Bluetooth device, the signal goes high. At this high state, the led blinks with a constant delay say for example 2s delay which indicates that the module is paired.

Button switch: -

This is used to switch the module into AT command mode. To enable AT command mode, press the button switch for a second. With the help of AT commands, the user can change the parameters of this module but only when the module is not paired with any other BT device. If the module is connected to any other Bluetooth device, it starts to communicate with that device and fails to work in AT command mode

3.2 Relays: -

This is a 5V 4-channel relay interface board, and each channel needs a 15-20mA driver current. It can be used to control various appliances and equipment with large current. It is equipped with high-current relays that work under AC 250V/10A or DC 30V/10A. It has a standard interface that can be controlled directly by microcontroller.

On the other side of the module we have these 2 sets of pins. The first one has 4 pins, a Ground and a V_{CC} pin for powering the module and 2 input pins In1 and In2. The second set of pins has 3 pins with a jumper between the JD, Vcc and the Vcc pin. With a configuration like this the electromagnet of the relay is directly powered from the Arduino Board and if something goes wrong with the relay the microcontroller could get damaged.

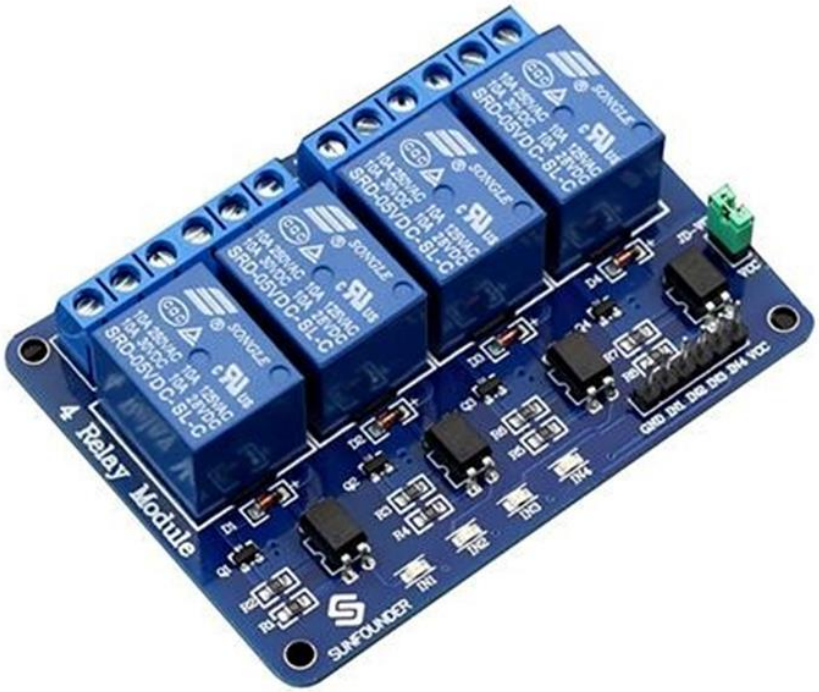


Figure 5: Relay Module

3.2.1. PRINCIPLE OF RELAY OPERATION: -

From the picture next page, you can see that when the signal port is at low level, the signal light will light up and the Optocoupler 817c (it transforms electrical signals by light and can isolate input and output electrical signals) will conduct, and then the transistor will conduct, the relay coil will be electrified, and the normally open contact of the relay will be closed. When the signal port is at high level, the normally closed contact of the relay will be closed. So you can connect and disconnect the load by controlling the level of the control signal port.

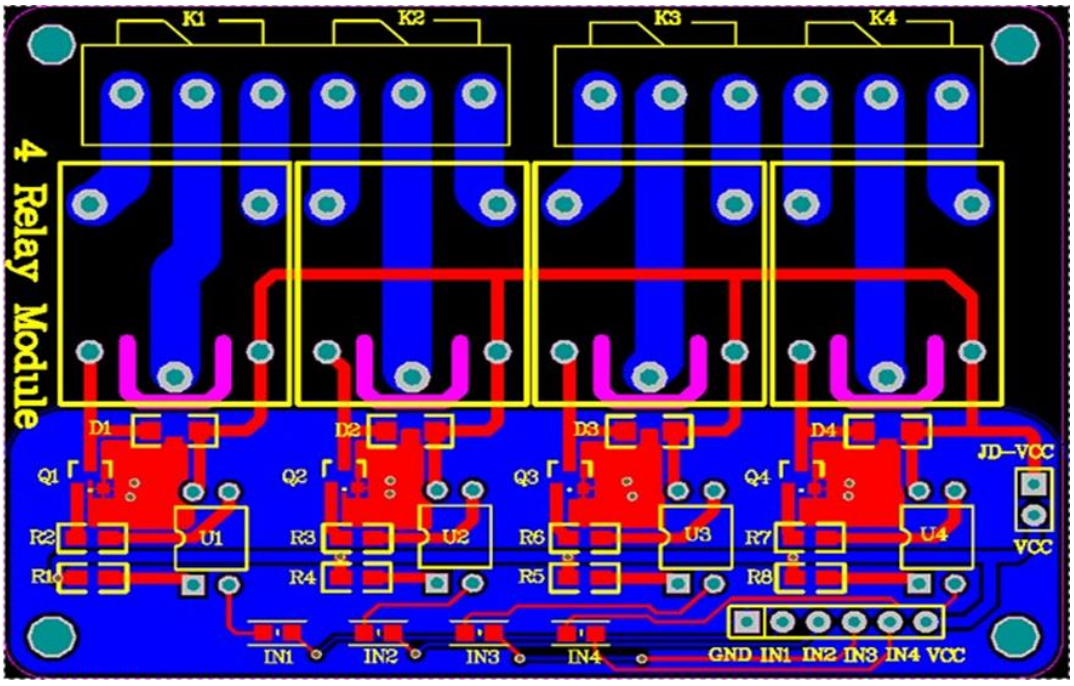


Figure 6: Overall view of the relay modules

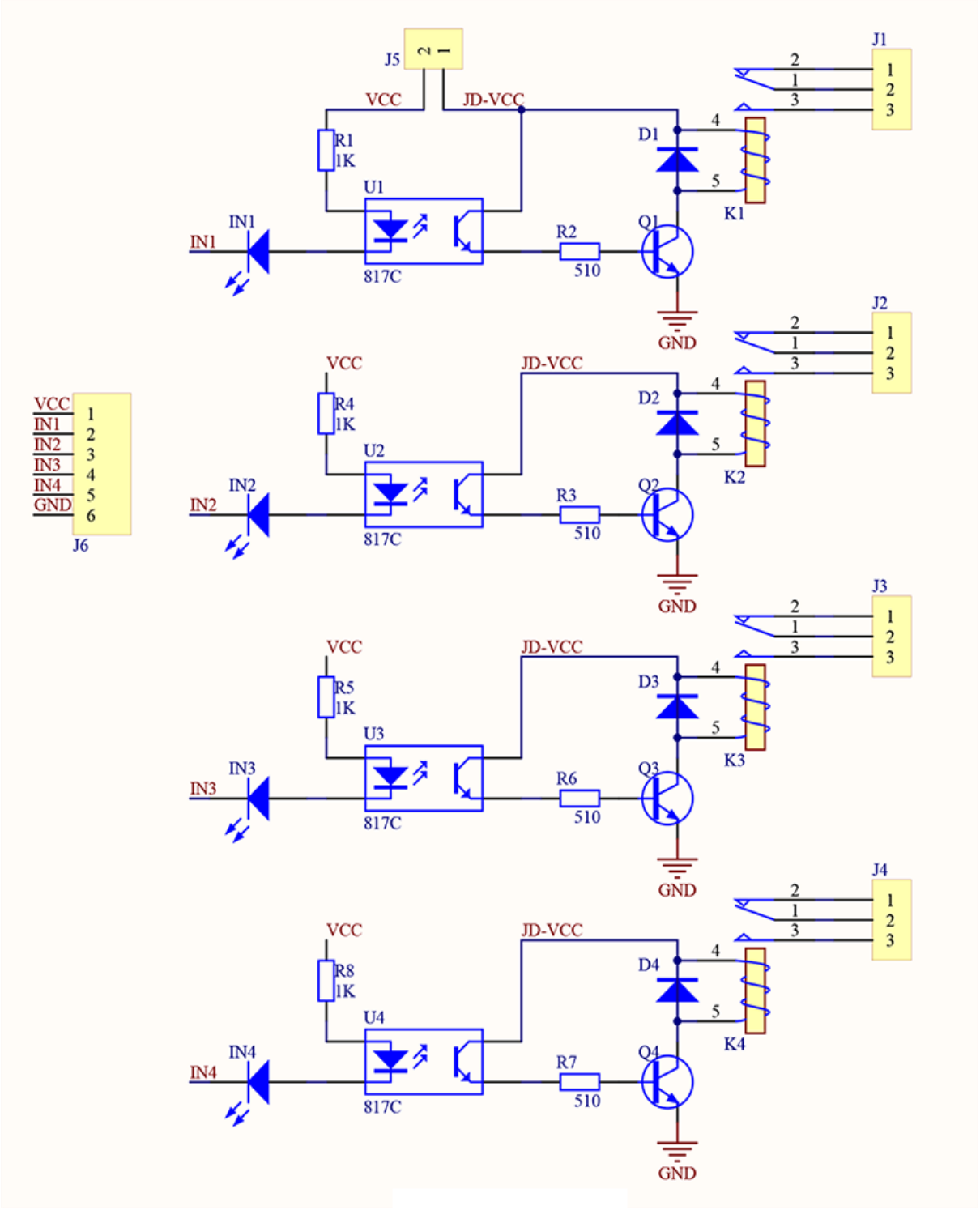


Figure 7: Equivalent circuit showing the relay connections

3.2.2 PIN DESCRIPTION: -

Input

VCC: Positive supply voltage

GND: Ground

IN1--IN4: Relay control port

Output: Connect a load, DC 30V/10A, AC 250V/10A

3.2.3 FEATURES: -

- Size: 75mm (Length) * 55mm (Width) * 19.3mm (Height)
- Weight: 61g
- PCB Color: Blue
- There are four fixed screw holes at each corner of the board, easy for install and fix. The diameter of the hole is 3.1mm
- High quality Single relay is used with single pole double throw, a common terminal, a normally open terminal, and a normally closed terminal
- Optical coupling isolation, good anti-interference.
- Closed at low level with indicator on, released at high level with indicator off
- V_{CC} is system power source, and JD_V_{CC} is relay power source. Ship 5V relay by default. Plug jumper cap to use.

3.2.4 EXPERIMENT PRINCIPLE: -

When the input terminals (IN1, IN2, IN3, and IN4) are supplied with low level signals, you can see relay K1, K2, K3, K4 closed successively and repeat this cycle. In order to show its the ability of driving load more intuitively, two LEDs are connected to relay K1 and K2.

Table 2: Connection between the relay and development board

4 Channel relay shield	Sun Founder UNO R3
GND	GND
IN1	2
IN2	3
IN3	4

IN4	5
VCC	5V

4. HOW TO CONNECT HC05 BLUETOOTH MODULE: -

4.1 UNO: HARDWARE AND SOFTWARE REQUIRED: -

- HC-05 Bluetooth Module
- Arduino Uno
- Arduino IDE(1.0.6V)

4.2 HARDWARE CONNECTIONS: -

As we know that Vcc and G_{ND} of the module goes to Vcc and G_{ND} of Arduino. The T_{XD} pin goes to R_{XD} pin of Arduino and R_{XD} pin goes to T_{XD} pin of Arduino i.e. (digital pin 0 and 1). The user can use the on board Led. But here, LED is connected to digital pin 12 externally for betterment of the process.

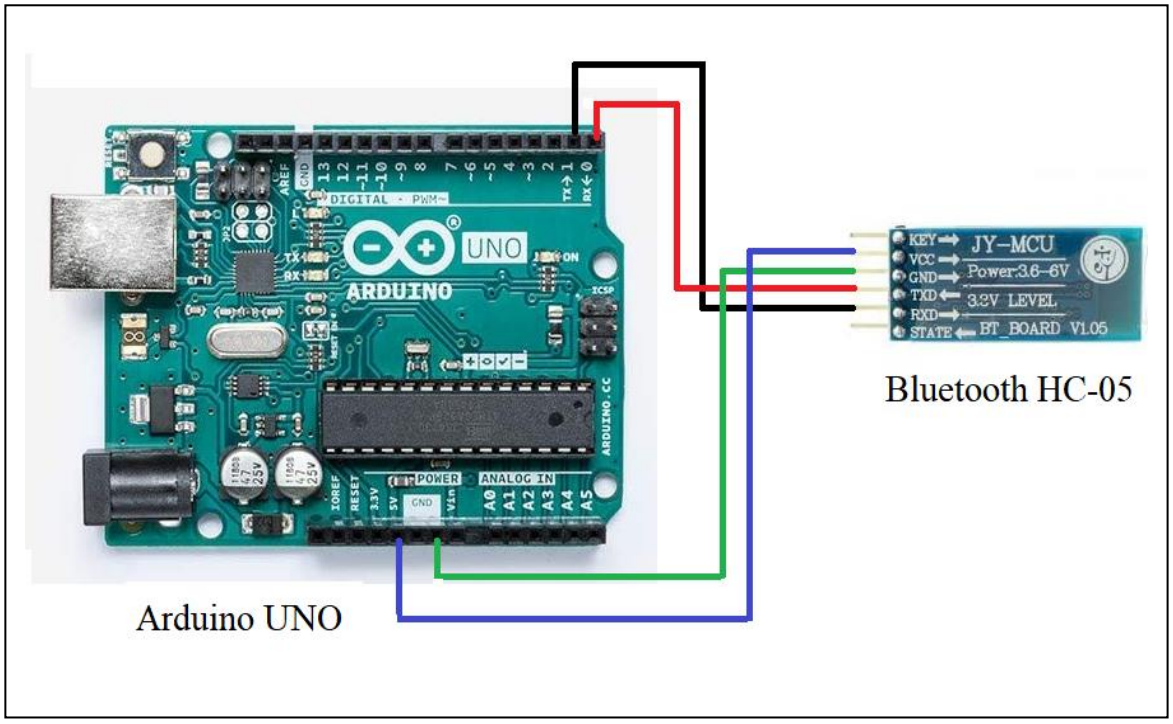


Figure 8: Connection between the Bluetooth module and Arduino Board

4.3 BLOCK DIAGRAM: -

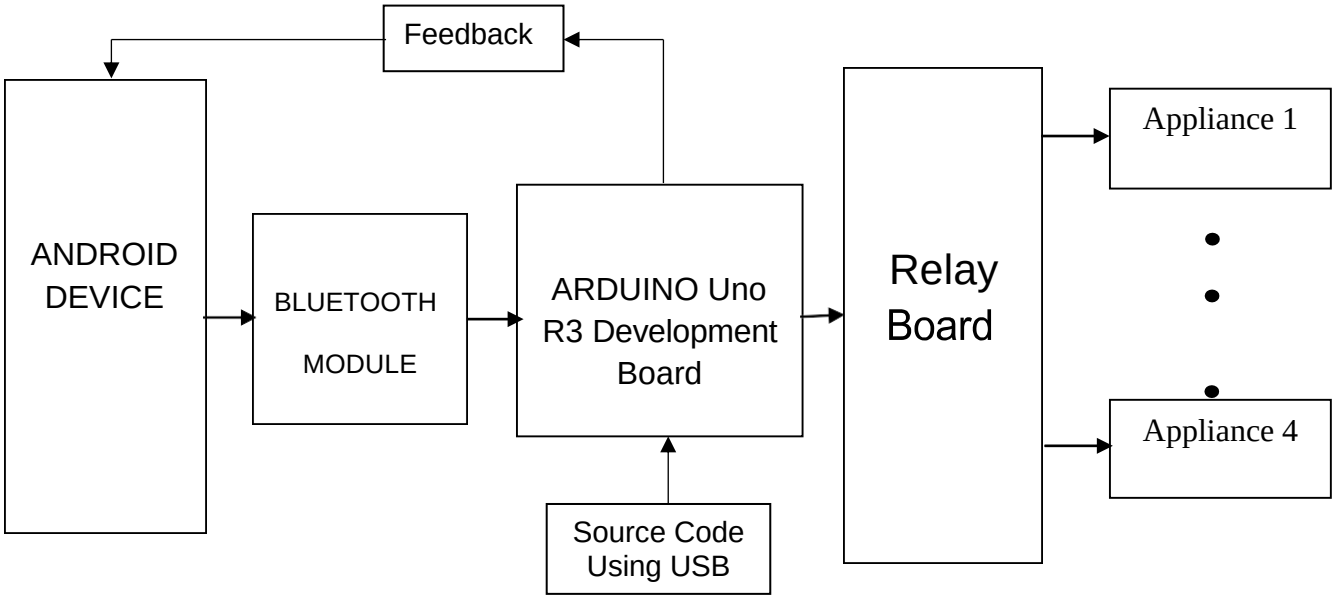


Figure 9: Block Diagram of the Automation System

5. CONNECTION, SCHEME DIAGRAM AND SOURCE CODE: -

5.1 CONNECTION: -

- Connect pins 2-5 of Arduino to Relays R1-R4 at pin "x" of each relay respectively.
- Interconnect all the "y" pins of each relay and connect one of them to the GND pin of Arduino.
- Connect VCC of Bluetooth module to 5v power pin of Arduino and likewise GND pin of Bluetooth module to GND pin of Arduino.
- Interconnect all the "t1" pins of each relay and connect one of them to 220V input of main power supply.
- Connect any one terminal of each of the devices to be controlled to pin "t2" of each of the relays R1-R4 respectively.
- Connect the other remaining terminals (Ground terminals) of all the devices to be controlled to GND of the main power supply.
- Connect T_{XD} of Bluetooth module to R_{XD} of Arduino and R_{XD} of Bluetooth module to T_{XD} of Arduino.

5.2 SCHEMATIC DIAGRAM: -

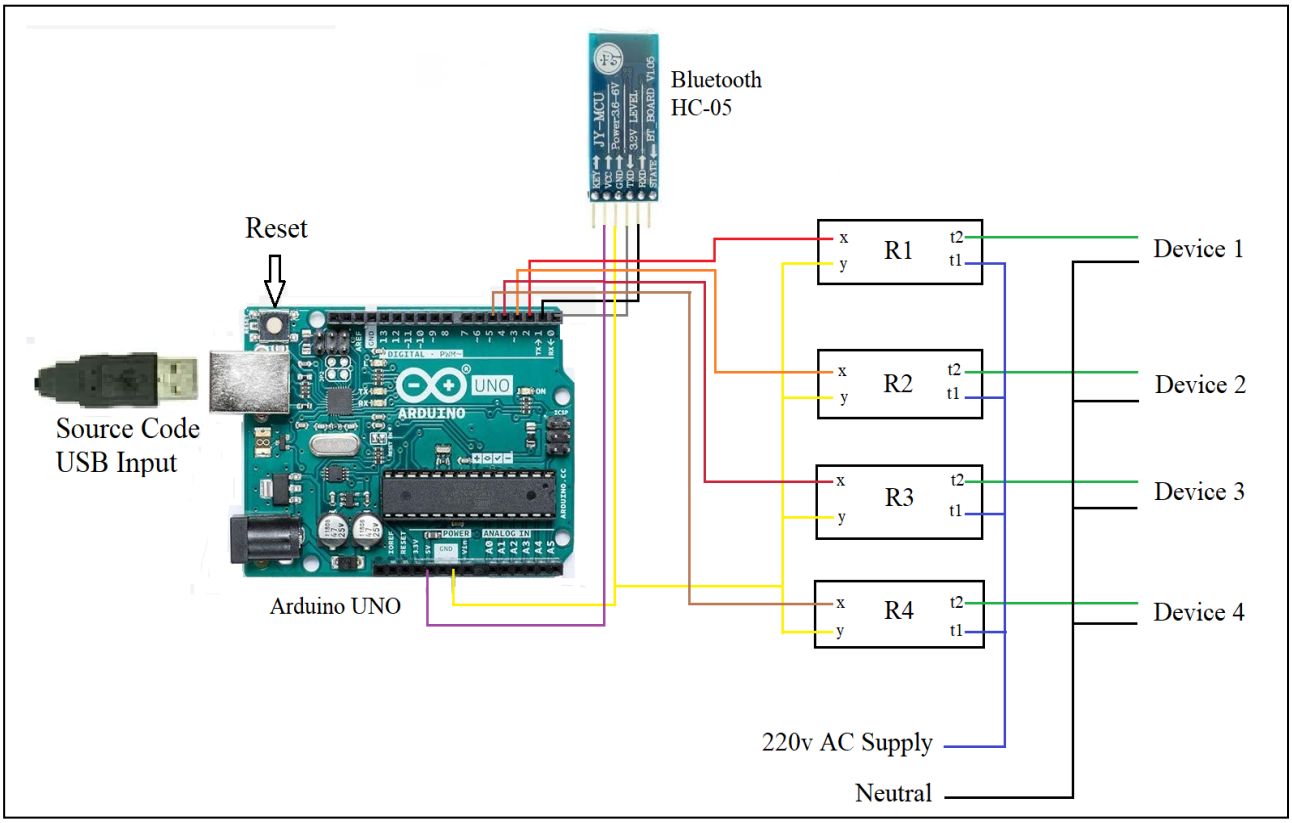


Figure 10: Complete circuit connection of the Automation system

5.3 SOURCE CODE: -

```
String inputs;
#define relay1 2 //Connect relay1 to pin 2
#define relay2 3 //Connect relay2 to pin 3
#define relay3 4 //Connect relay3 to pin 4
#define relay4 5 //Connect relay4 to pin 5 int
val1=0;
int val2=0; int
val3=0; int
val4=0; 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
digitalWrite(relay1, HIGH); //Switch relay1 off digitalWrite(relay2,
HIGH); //Swtich relay2 off digitalWrite(relay3, HIGH); //Switch
relay3 off digitalWrite(relay4, HIGH); //Swtich relay4 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")
{
if(val1==0)
{
digitalWrite(relay1,LOW);
val1=1;
}
else
{
digitalWrite(relay1,HIGH);
val1=0;
}
delay(100);
}
else if(inputs == "b")
```

```
{
if(val2==0)
{
digitalWrite(relay2,LOW);
val2=1;
}
else
{
digitalWrite(relay2,HIGH);
val2=0;
}
delay(100);
}
else if(inputs == "c")
{
if (val3==0)
{
digitalWrite (relay3,LOW);
val3=1;
}
else
{
digitalWrite (relay3,HIGH);
val3=0;
}
delay(100);
```

```
}

else if(inputs == "d")

{

if(val4==0)

{

digitalWrite(relay4,LOW);

val4=1;

}

else

{

digitalWrite(relay4,HIGH);

val4=0;

}

delay(100);

}

inputs="";
}
}
```

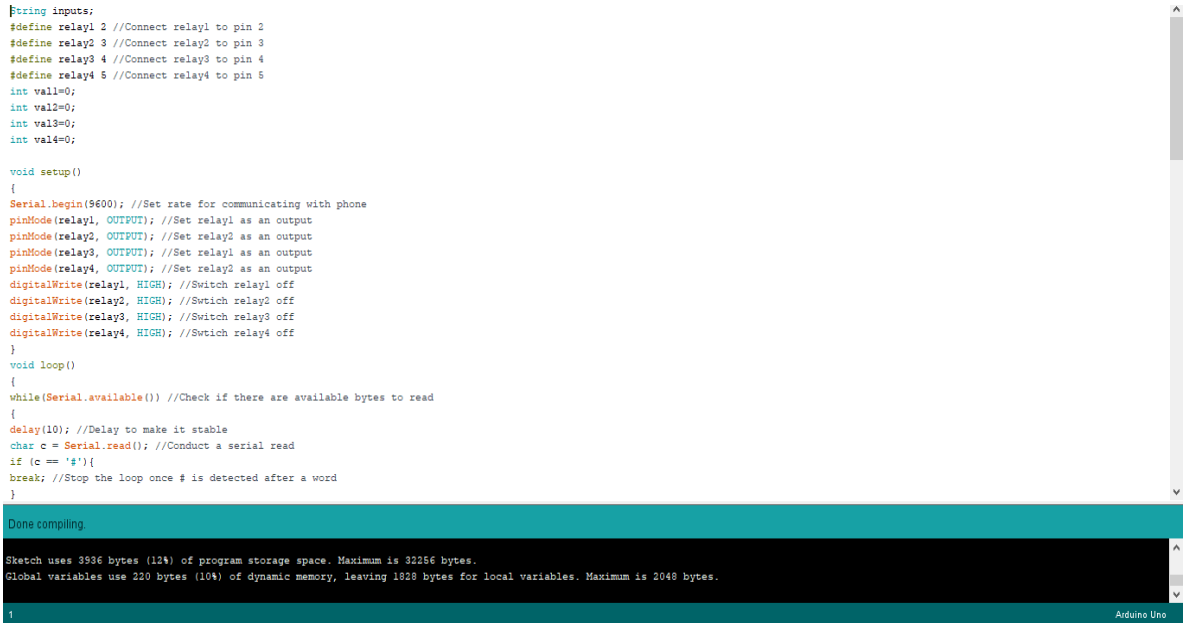


Figure 11: Successful compilation of code

5.4 SIMULATIONS, RESULTS AND DISCUSSION:

To simulate the circuits we have used Proteus 8 Professional, Arduino compiler and Tinkercad.

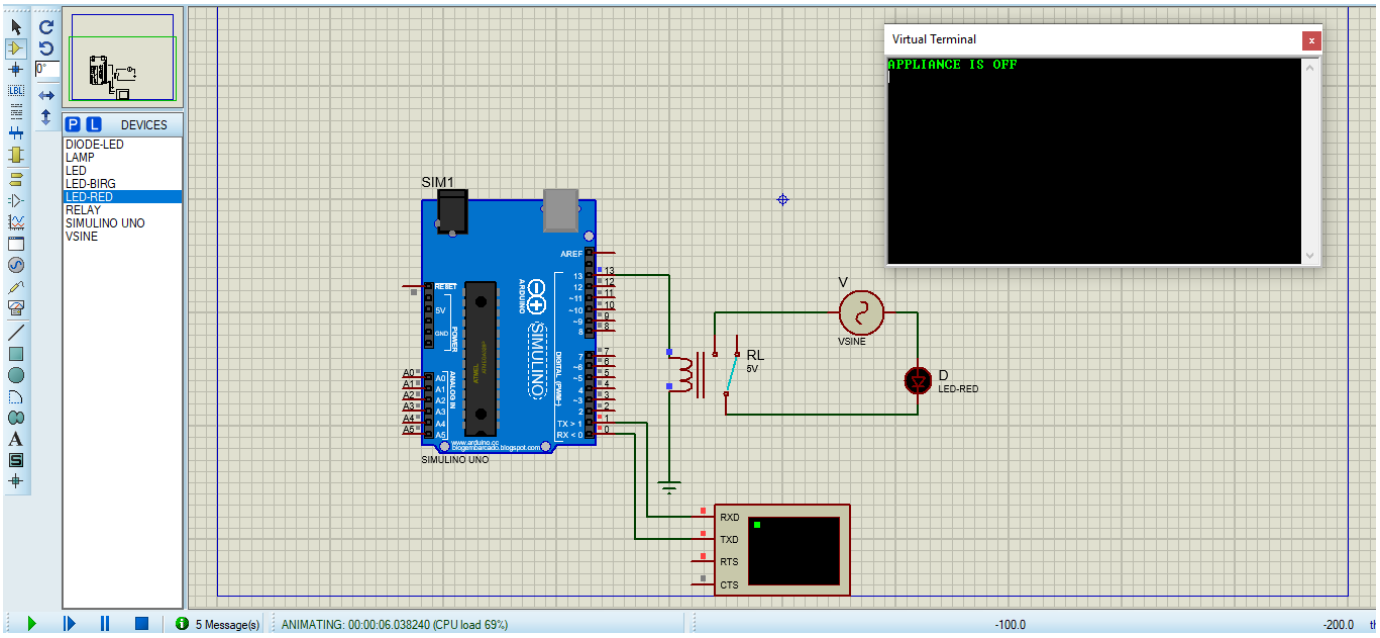


Figure 12: The switch is off at this moment

In figure 12, we take the input through the virtual terminal. Thus the virtual terminal acts as an interface here. It has been coded in such a way that when any single digit number is entered apart from 1 then a LOW signal is sent to the circuit denoting that the “APPLIANCE IS OFF”. Now when we enter 1 into the virtual terminal, a HIGH signal is sent denoting “RELAY IS ACTIVATED, APPLIANCE IS SWITCHED ON”.

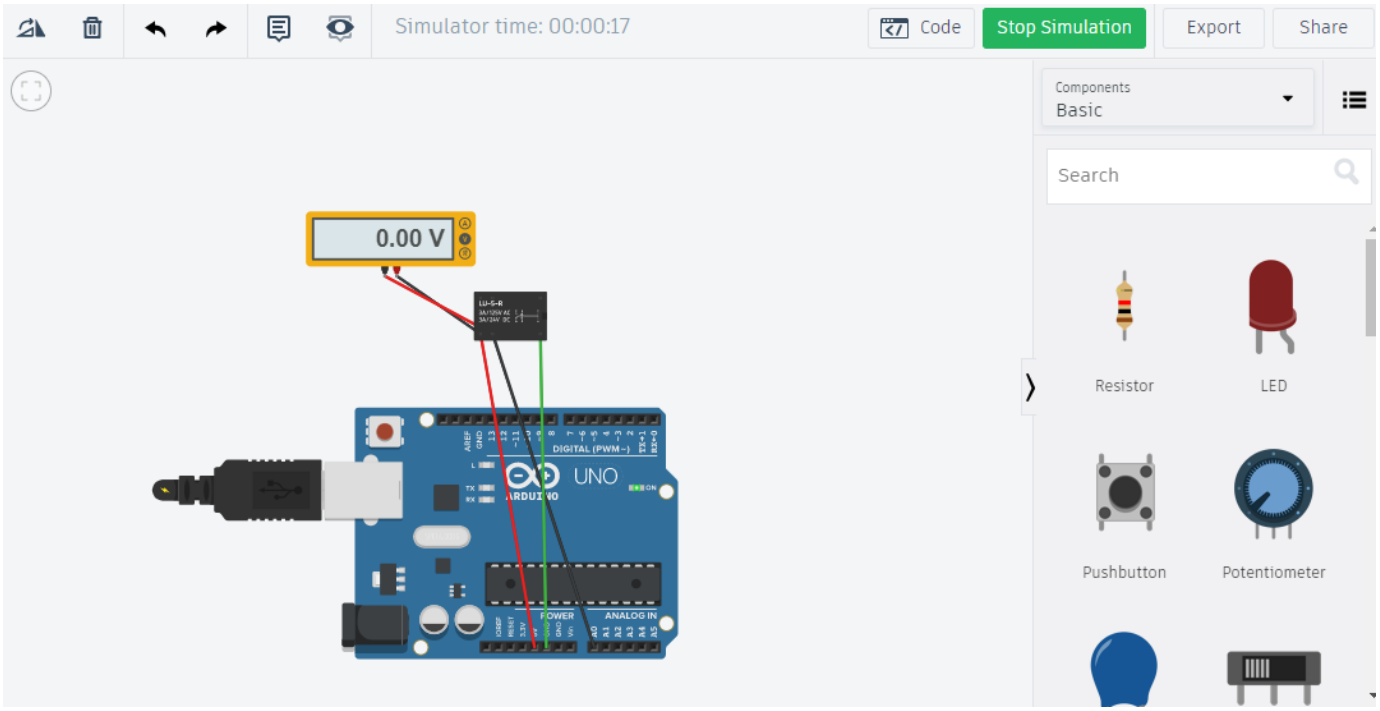


Figure 13: When Voltage is LOW

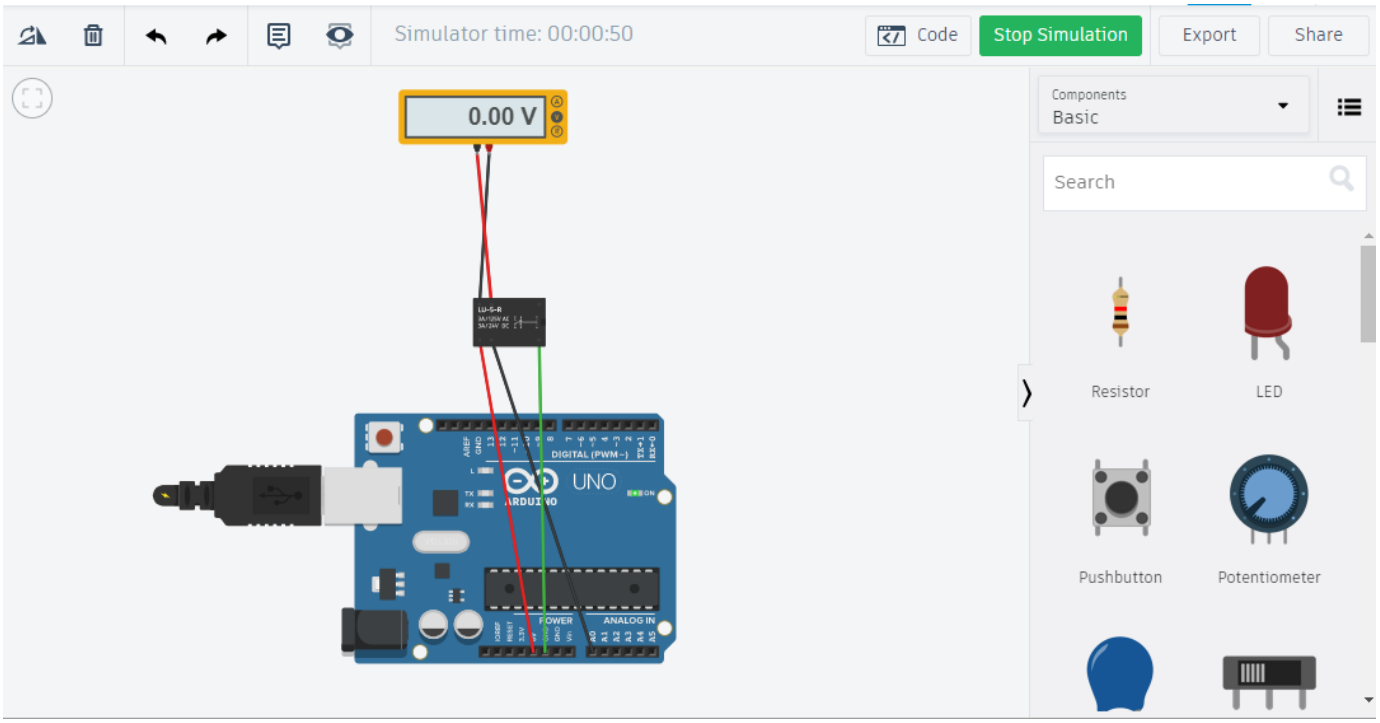


Figure 14: Relay is not activated

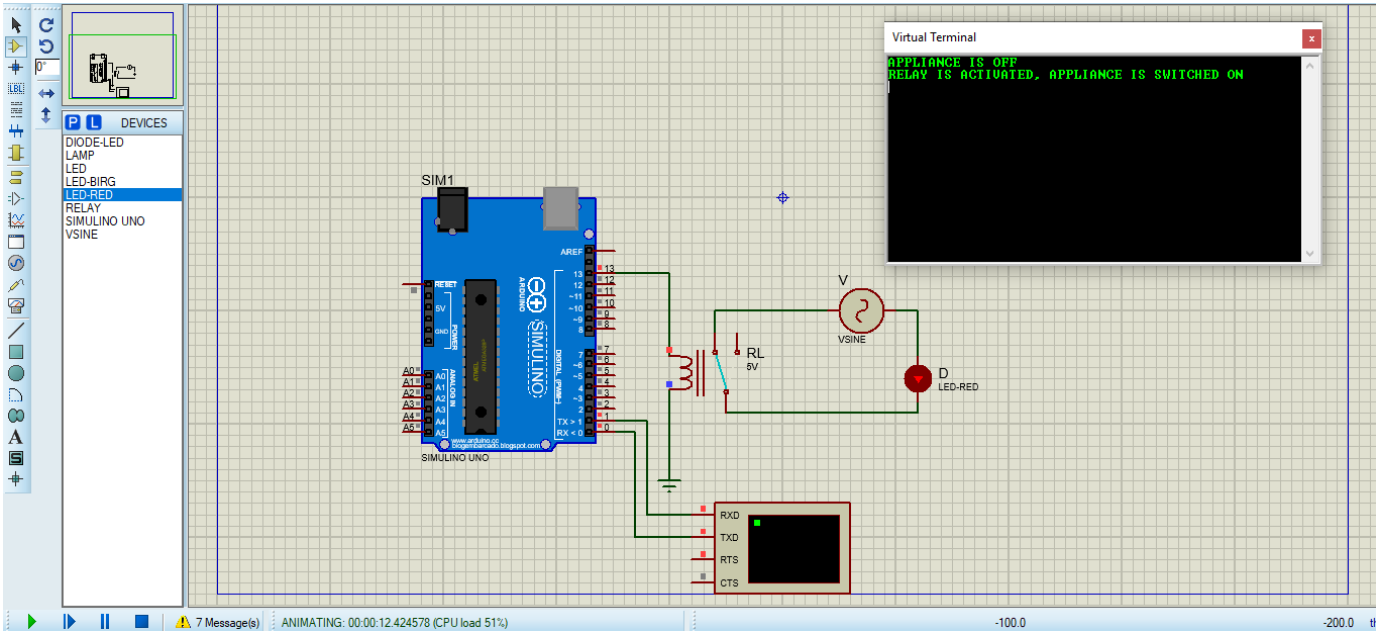


Figure 15: When switch is on

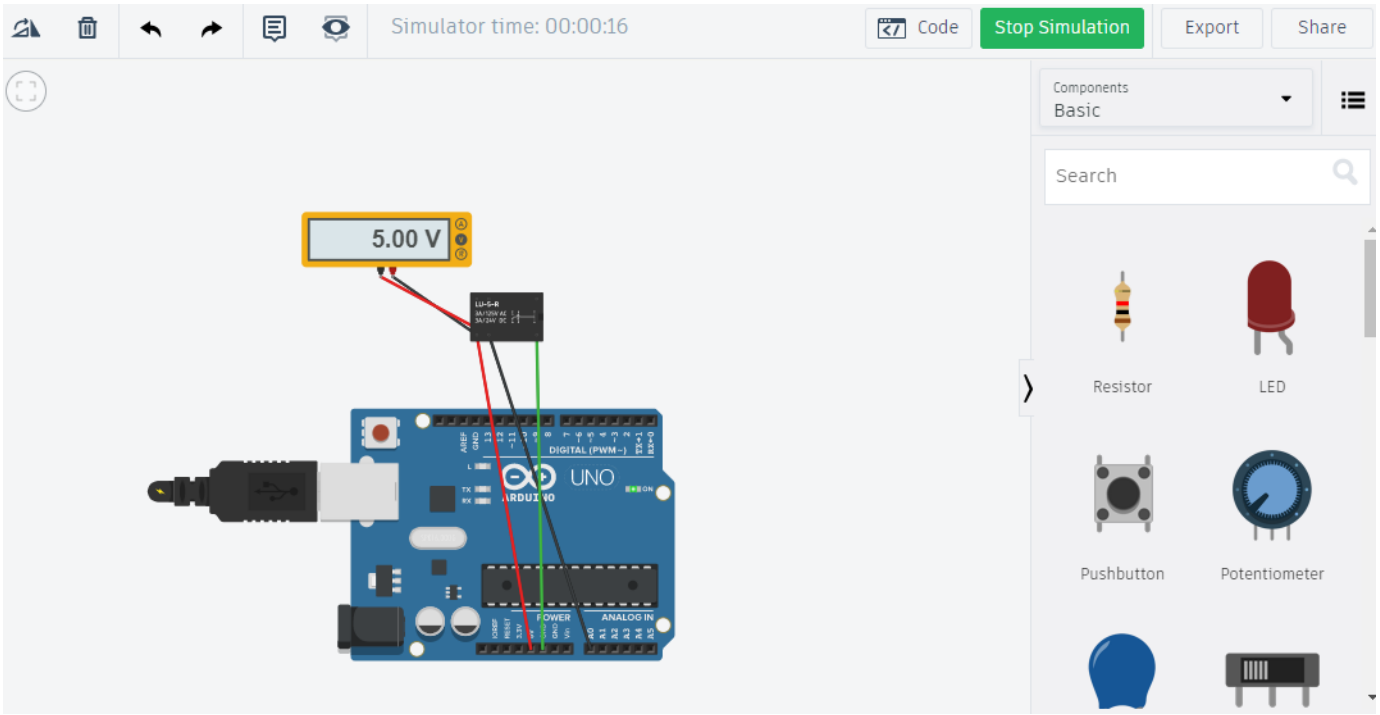


Figure 16: When voltage is High

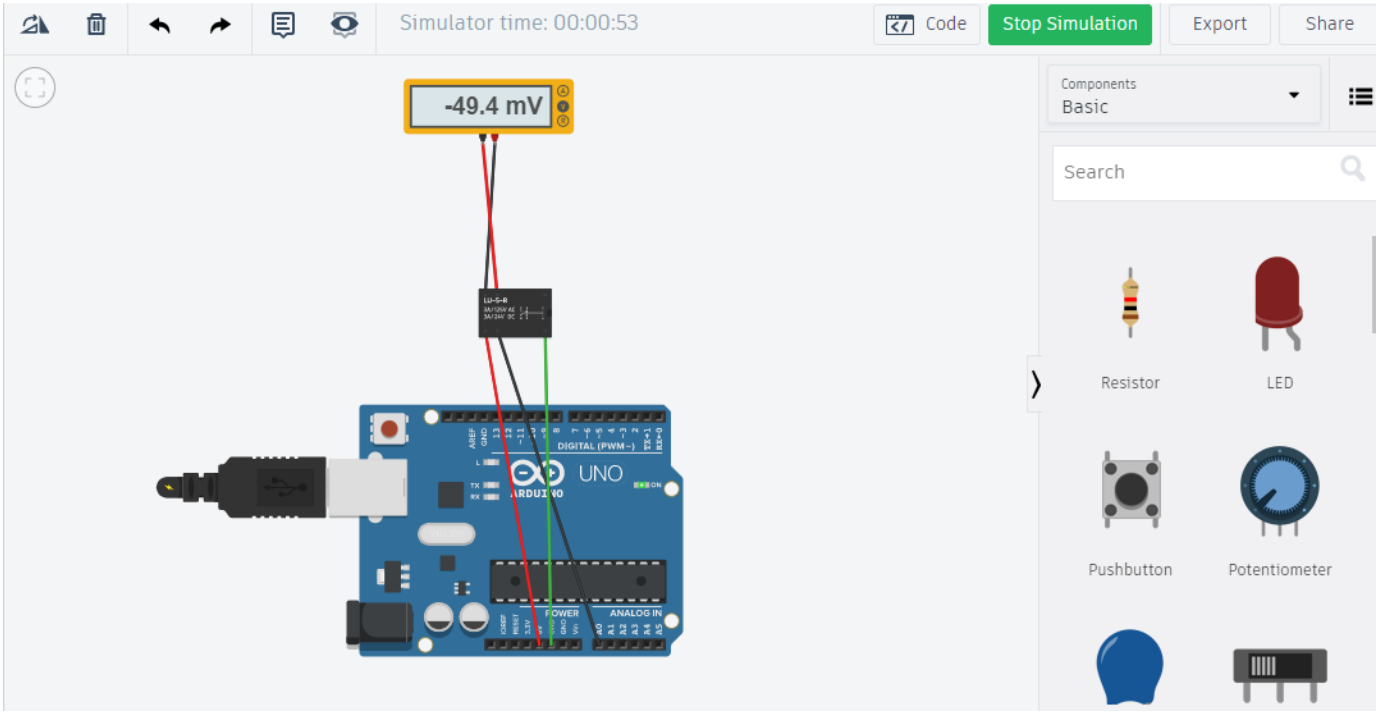


Figure 17: Relay is activated

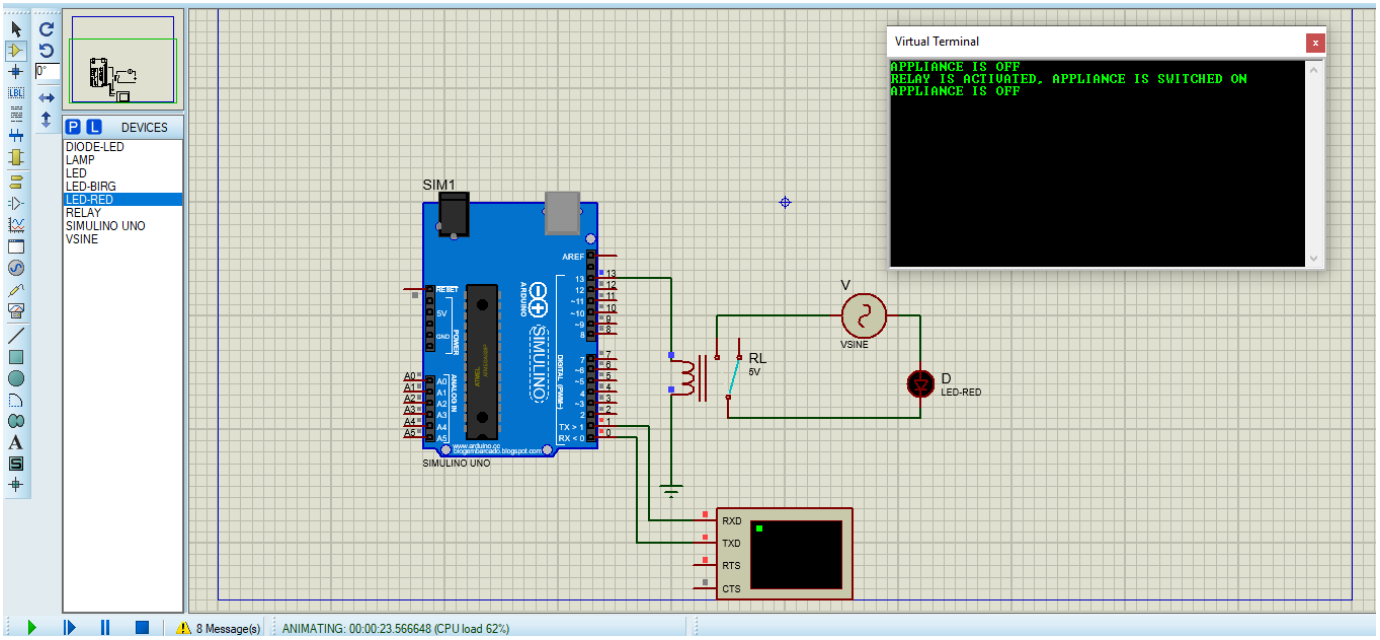


Figure 18: Now the appliance is turned off

Table 3: Relay Voltage level

Voltage level (Volts) at Vcc	0 (LOW)	5 (HIGH)
Status of Relay	OFF	ON

6. SECURITY ALARM CIRCUIT USING PIR MOTION SENSOR: -

PIR sensors allow you to sense motion, almost always used to detect whether a human has moved in or out of the sensors range. They are small, inexpensive, low- power, easy to use and don't wear out. PIRs are basically made of a pyro electric which can detect levels of infrared radiation. Everything emits some low level radiation, and the hotter something is, the more radiation is emitted. The sensor in a motion detector is actually split in two halves.

The reason for that is that we are looking to detect motion (change) not average IR levels. The two halves are wired up so that they cancel each other out. If one half sees more or less IR radiation than the other, the output will swing high or low.

7. COMPONENTS LIST: -

- Arduino Uno board
- PIR Sensor DSN FIR-800
- GSM Module (SIM 900A)
- 10K Resistor
- 10 μF Capacitor
- Buzzer
- Push Bottom Switch
- USB cable for Arduino

7.1 PIR SENSOR: -

The PIR (Passive Infra-Red) Sensor is a pyro electric device that detects motion by sensing changes in the infrared (radiant heat) levels emitted by surrounding objects. This motion can be detected by checking for a sudden change in the surrounding IR pattern. When motion is detected the PIR sensor outputs a high signal on its output pin. This logic signal can be read by a microcontroller or used to drive an external load.

7.1.1 FEATURES: -

i. Detect a person up to approximately 30 ft. away, or up to 15 ft. away in reduced sensitivity mode.
ii. Jumper selects normal operation or reduces sensitivity.
iii. Source current up to 12 mA @ 3 V, 23 mA @ 5 V.
iv. Onboard LEDs light up the lens for fast visual feedback when movement is detected.
v. Mounting holes for #2 sized screws.
vi. 3-pin SIP header ready for breadboard or through-hole projects.
vii. Small size makes it easy to conceal.

7.1.2 KEY SPECIFICATIONS: -

i. Power Requirements: 3 to 6 VDC; 130 μA idle, 3 mA active (no load)

ii. Communication: Single bit high/low output

iii. Operating temperature: 32 to 122 °F (0 to 50 °C)

iv. Dimensions: 1.41 x 1.0 x 0.8 in (35.8 x 25.4 x 20.3 cm)

7.1.3 WORKING OF PIR SENSOR: -

PIR sensors are more complicated than many of the other sensors explained in these tutorials (like photocells, FSRs and tilt switches) because there are multiple variables that affect the sensors input and output.

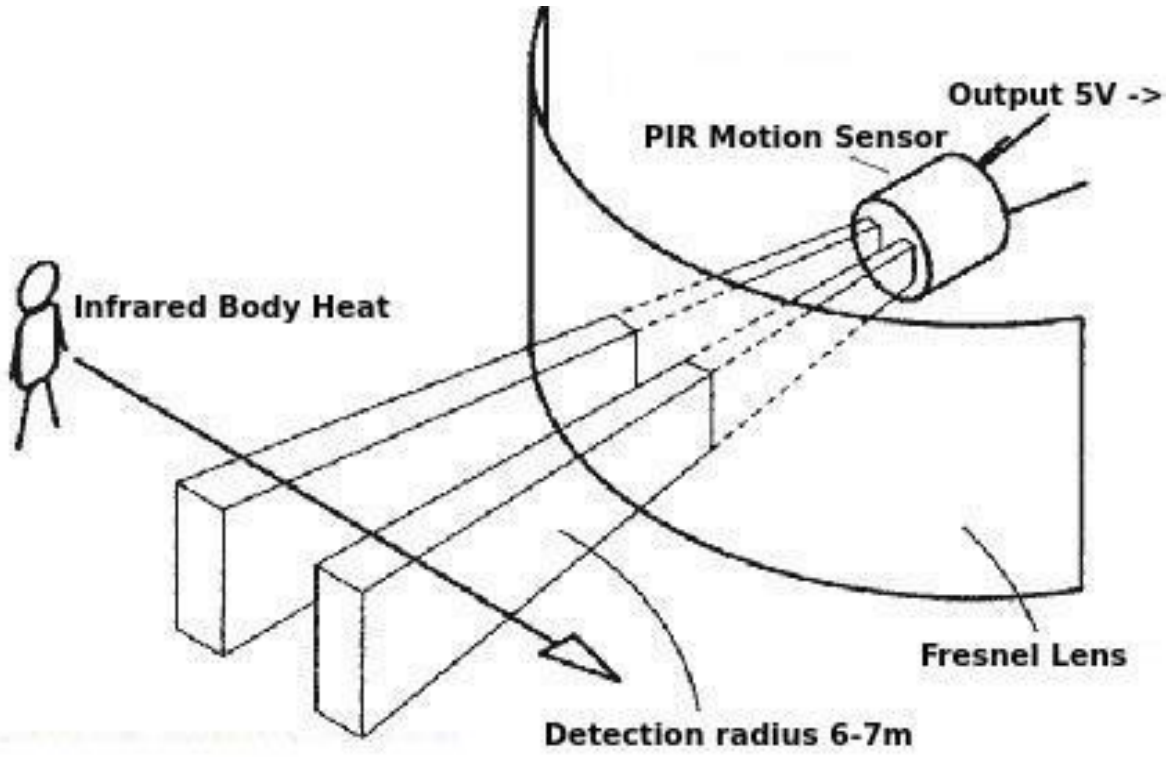


Figure 19: Working of a PIR sensor

The PIR sensor itself has two slots in it; each slot is made of a special material that is sensitive to IR. The lens used here is not really doing much and so we see that the two slots can 'see' out past some distance (basically the sensitivity of the sensor). When the sensor is idle, both slots detect the same amount of IR, the ambient amount radiated from the room or walls or outdoors. When a warm body like a human or animal passes by, it first intercepts one half of the PIR sensor, which causes a positive differential change between the two halves.

When the warm body leaves the sensing area, the reverse happens, whereby the sensor generates a negative differential change. These change pulses are what is detected.

Figure 20: PIR sensor

7.2 BUZZER: -

This is used to create a sound alarm whenever a movement is identified inside the range of PIR sensor. A transistor 2N2222 is used to drive the buzzer. The maximum current that can be sourced or synced from an Arduino pin is 20mA (the total current being 200mA from different pins). But the buzzer will need more than just 20mA for its proper functioning. So how to give the necessary current required fir buzzer? We use switching transistor 2N222 for this purpose. It can act as a switch and at the same time it provides the required current amplification. A 2N2222 transistor with a gain of 100 can give up to 1A current at its output. Another purpose of using a transistor in between Arduino pin and buzzer is isolation. A short circuit of the buzzer will destroy only the collector-emitter junction of transistor. Since there is isolation at the base region of transistor (base is connected to Arduino), the destruction of collector-emitter junction will not affect base and hence our Arduino will be safe from getting burned! The 100 ohms' resistor at base is used to limit base current of transistor.

7.3 GSM MODULE (SIM 900A): -

The SIM900A is a readily available GSM/GPRS module, used in many mobile phones and PDA. The module can also be used for developing IOT (Internet of Things) and Embedded Applications. SIM900A is a dual-band GSM/GPRS engine that works on frequencies EGSM 900MHz and DCS 1800MHz. SIM900A features GPRS multi-slot class 10/ class 8 (optional) and supports the GPRS coding schemes CS-1, CS-2, CS-3 and CS-4.

7.3.1 SIM900A GSM MODULE FEATURES: -

- **Single supply voltage:** 3.4V – 4.5V
- **Power saving mode:** Typical power consumption in SLEEP mode is 1.5mA
- **Frequency bands:** SIM900A Dual-band: EGSM900, DCS1800. The SIM900A can search the two frequency bands automatically. The frequency bands also can be set by AT command.
- **GSM class:** Small MS
- **GPRS connectivity:** PRS multi-slot class 10 (default), GPRS multi-slot class 8 (option)
- **Transmitting power:** Class 4 (2W) at EGSM 900, Class 1 (1W) at DCS 1800
- **Operating Temperature:** -30ºC to +80ºC
- **Storage Temperature:** -5ºC to +90ºC
- **DATA GPRS:** download transfer max is 85.6KBps, Upload transfer max 42.8KBps
- Supports CSD, USSD, SMS, FAX
- Supports MIC and Audio Input
- Speaker Input
- Features keypad interface
- Features display interface
- Features Real Time Clock
- Supports UART interface
- Supports single SIM card
- Firmware upgrade by debug port
- Communication by using AT commands

7.3.2 HOW TO USE SIM900A MODULE: -

The usage of the module let us consider a simple application circuit as shown below.

Figure 21: SIM900A Module

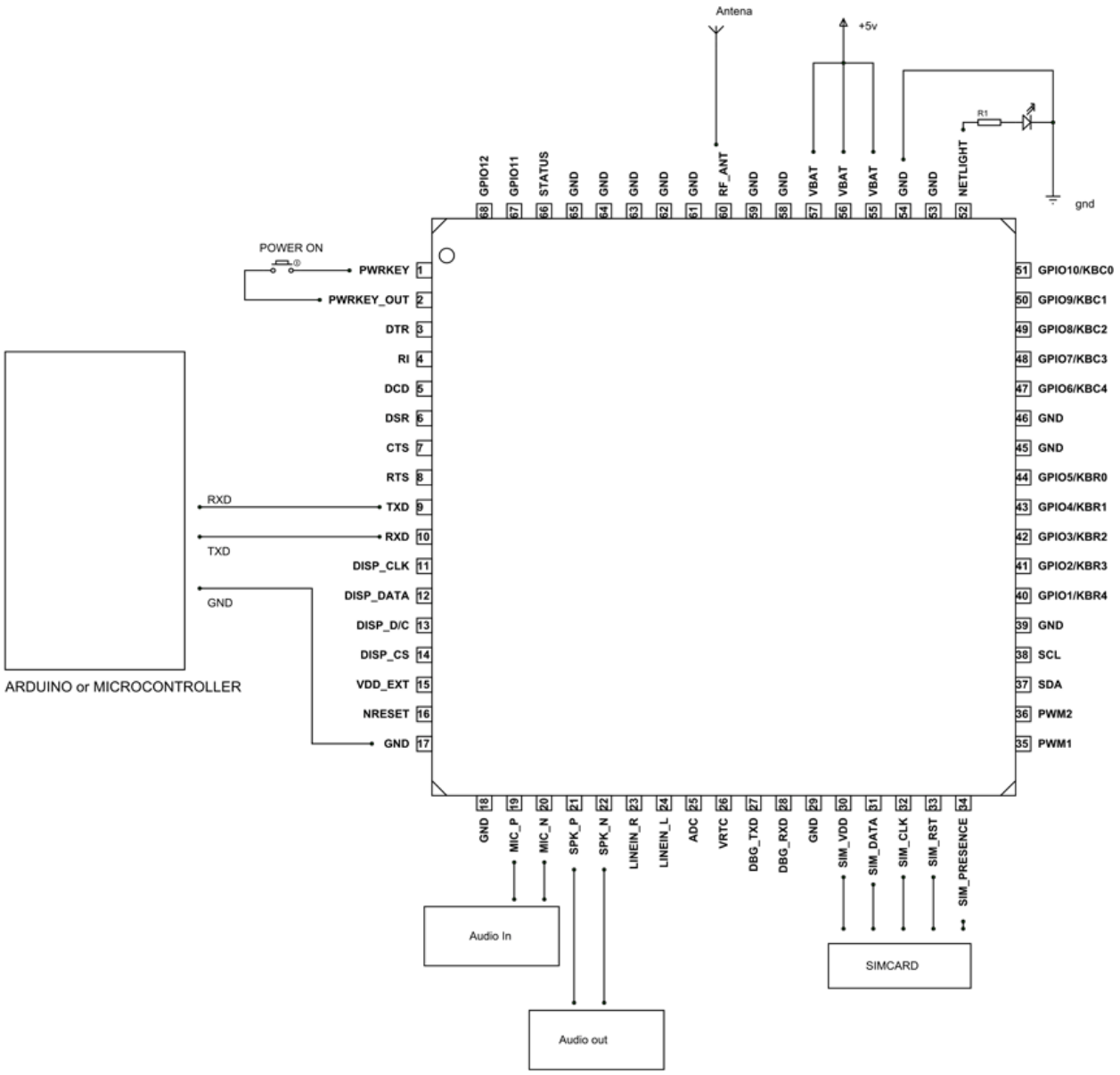


Figure 22: SIM900A Module Top view

As Shown above, the communication with this module is done through UART or RS232 Interface. The data is sent to the module or received from the module though UART interface.

The module is typically connected to +4.0V standard power supply. It can work on +4.5V regulated power and any higher voltage may damage the module. And the power source should be able to deliver a peak current of 2A. The UART interface is established as shown in figure. All you need to do is connect R_{XD} of module to T_{XD} of Arduino and T_{XD} is connected to R_{XD} of ARDUINO. The ground of controller and module must be connected for voltage reference. Here AUDIO IN is connected to MIC and AUDIO OUT is connected to a speaker or headset.

And at last we need to connect a working GSM SIM card to the module.

On powering the module, the NETLIGHT LED will blink periodically to state successful connection After all connections are done, we need to write a program for the microcontroller to exchange data with module. Since data exchange sequence between controller and module is really complex we will use libraries prewritten for the module. You can download libraries for controller or module through their websites. Using these libraries makes the communication easy. All you need to do is download these libraries and call them in programs. Once the header file is included, you can use simple commands in the program to tell the controller to send or receive data. The controller sends the data to the module through UART Interface based on protocol setup in libraries. The module sends this data to another GSM user using cellular network. If the module receives any data from the cellular network (or another GSM user) it will transmit it to controller through UART serial communication. This way we can use GSM900A module to establish cellular connection.

7.4 SWITCH: -

A push button switch is used to reset the burglar alarm once its activated. The capacitor is used for bypassing bouncing effects of a switch (de bouncing capacitor).

8. CONNECTION, CIRCUIT DIAGRAM AND SOURCE CODE

8.1 CONNECTION: -

Arduino – Pin 7 – Output of PIR Sensor | Pin 6 – Push button switch | Pin 8 – Buzzer

Buzzer – + pin to Vcc (5 volts) | other pin to collector side of 2N2222

Transistor – 2N2222 – NPN – Collector to Buzzer | Emitter to Ground | Base to Arduino through 100 Ohm Resistor

Switch – One end of switch to +5V | Other end to Ground through a 10K current limiting resistor

PIR Sensor – has got 3 pins – Vcc to +5 volts | GND to Ground | OUT pin to Arduino pin 7.

GSM Module – Connect its T_{XD} pin to Pin 9 of Arduino | Connect R_{XD} to Pin 10 – Arduino

| Vcc or Power Jack to +12 Volt | Make GND or Ground pin common to all other components and modules

Note: - Wire all grounds together at a common point.

8.2 OBJECTIVES OF THE PIR SENSOR ALARM USING ARDUINO: -

- Detect a motion – an intruder or a burglar using PIR sensor
- Activate the buzzer alarm upon detection of burglar/intruder – Alarm should sound until Reset switch is pressed
- Send 3 SMS to a predefined mobile number set inside the program.
- Stop the alarm when reset switch is pressed. Also reactivate the SMS alert facility upon reset.

8.3 CIRCUIT DIAGRAM: -

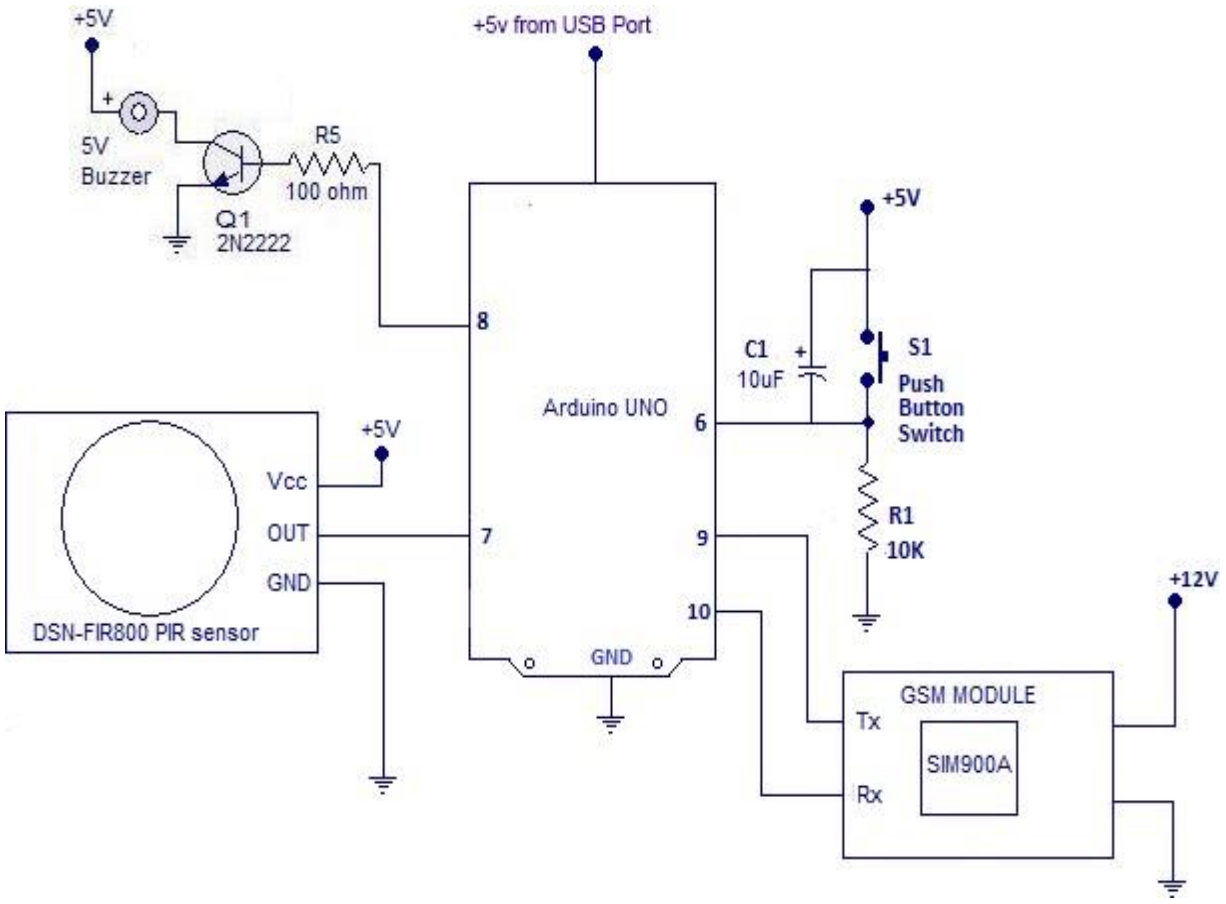


Figure 23: Complete circuit diagram of the Security System

8.4 SOURCE CODE: -

```
#include<SoftwareSerial.h> SoftwareSerial mySerial(9, 10);
int sensor=7; //The output of PIR sensor connected to pin 7 int
push_switch=6; // push button switch connected to pin 6 int
buzzer=8; // buzzer connected at pin 8
int sensor_value; //variable to hold read sensor value int
sms_count=0;
void setup()
{
pinMode(sensor,INPUT); // configuring pin 7 as Input
pinMode(push_switch,INPUT); // configuring pin 6 as Input
pinMode(buzzer,OUTPUT); // configuring pin 8 as OUTPUT
mySerial.begin(9600);
}
void loop()
{
Check_Burglar();// subroutine to check sensor status and activation of outputs Check_Reset(); //
subroutine to check if alarm reset switch pressed or not
}
void Check_Burglar()
{
sensor_value=digitalRead(sensor); // Reading sensor value from pin 7 if(sensor_value==HIGH) //
Checking if PIR sensor sends a HIGH signal to Arduino
{
digitalWrite(buzzer,HIGH); // Activating the buzzer while(sms_count<3)
//Number of SMS Alerts to be sent limited at 3
{
SendTextMessage(); // Function to send AT Commands to GSM module
}
}}
void Check_Reset()
{
if(digitalRead(push_switch==HIGH))// Checking if pushbutton was pressed
{
digitalWrite(buzzer,LOW); // turning OFF the buzzer sms_count=0;
```

```
// Reactivating the SMS Alert Facility
}}
void SendTextMessage()
{
mySerial.println("AT+CMGF=1"); //To send SMS in Text Mode
delay(1000);
mySerial.println("AT+CMGS=\"+919495xxxxxx\"\r"); // change to the user phone number
delay(1000);
mySerial.println("Security Alert!");//the content of the message
delay(200);
mySerial.println((char)26);//the stopping character delay(1000);
sms_count++;
}
```

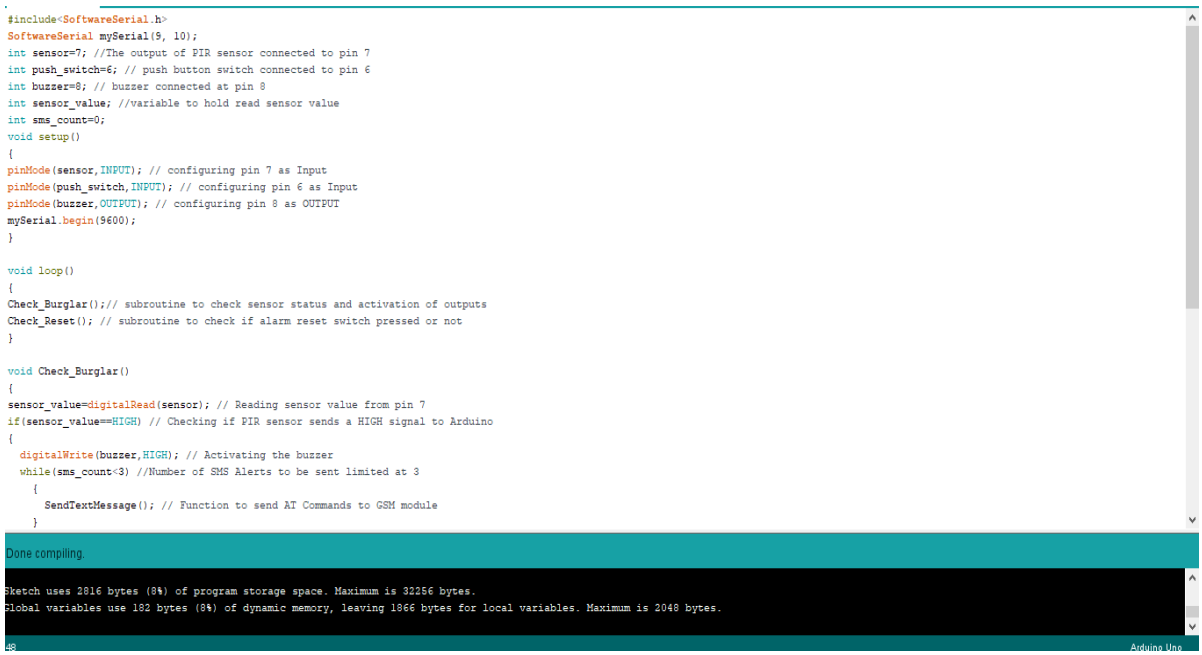


Figure 24: Successful compilation of code

8.5 SIMULATIONS, RESULTS AND DISCUSSIONS: -

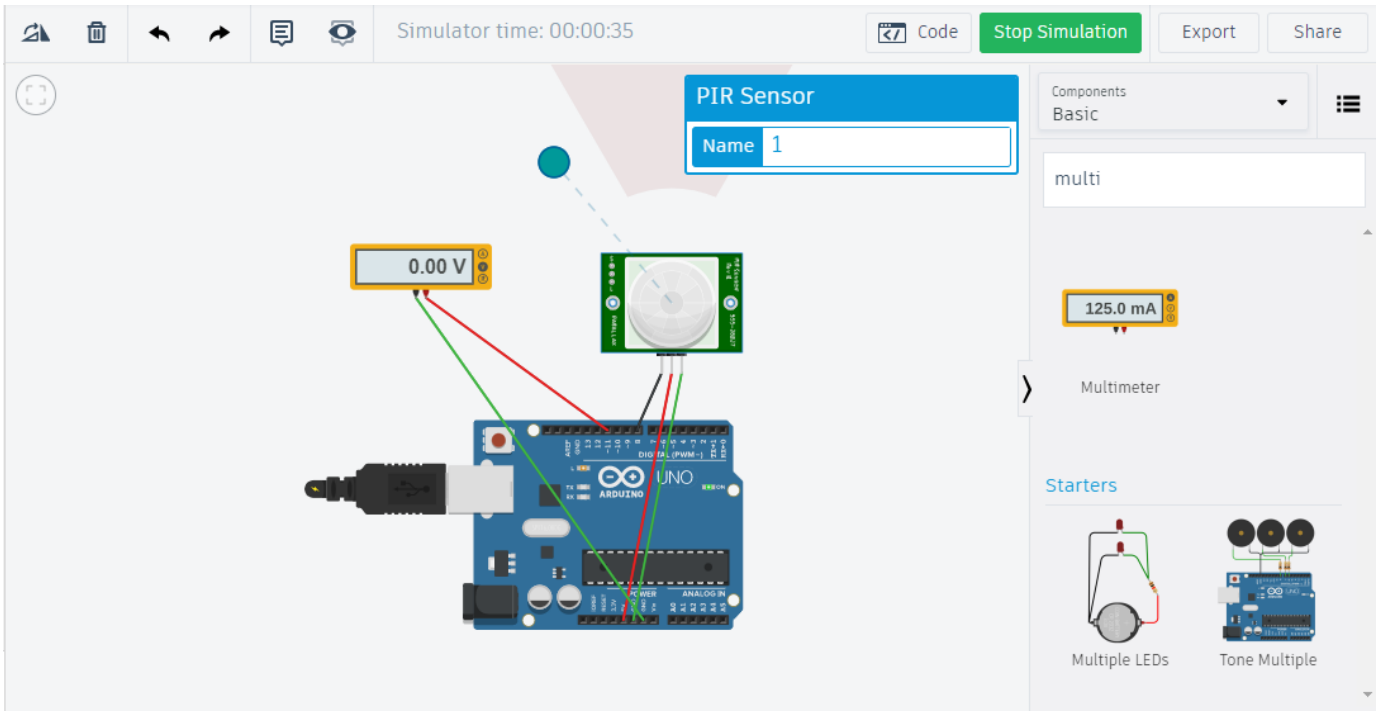


Figure 25: When the object is out of PIR's range

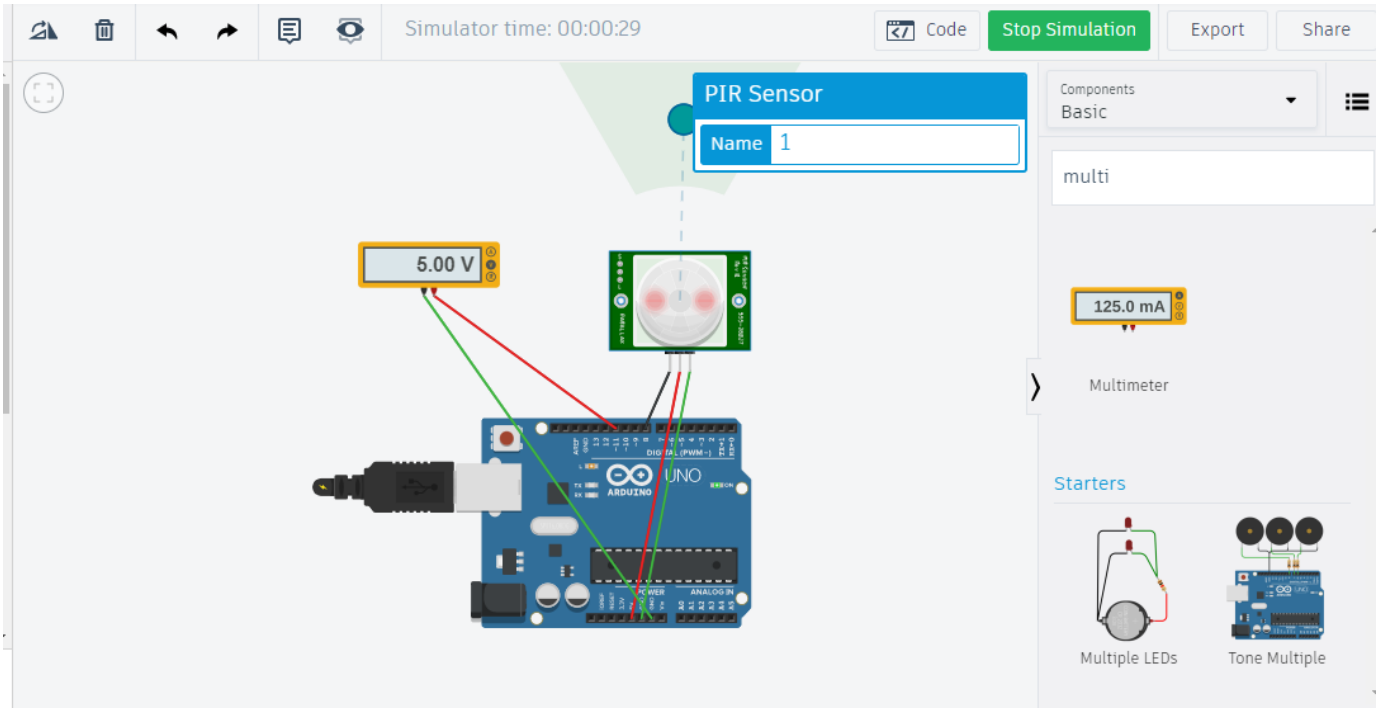


Figure 26: When the object is in PIR's range

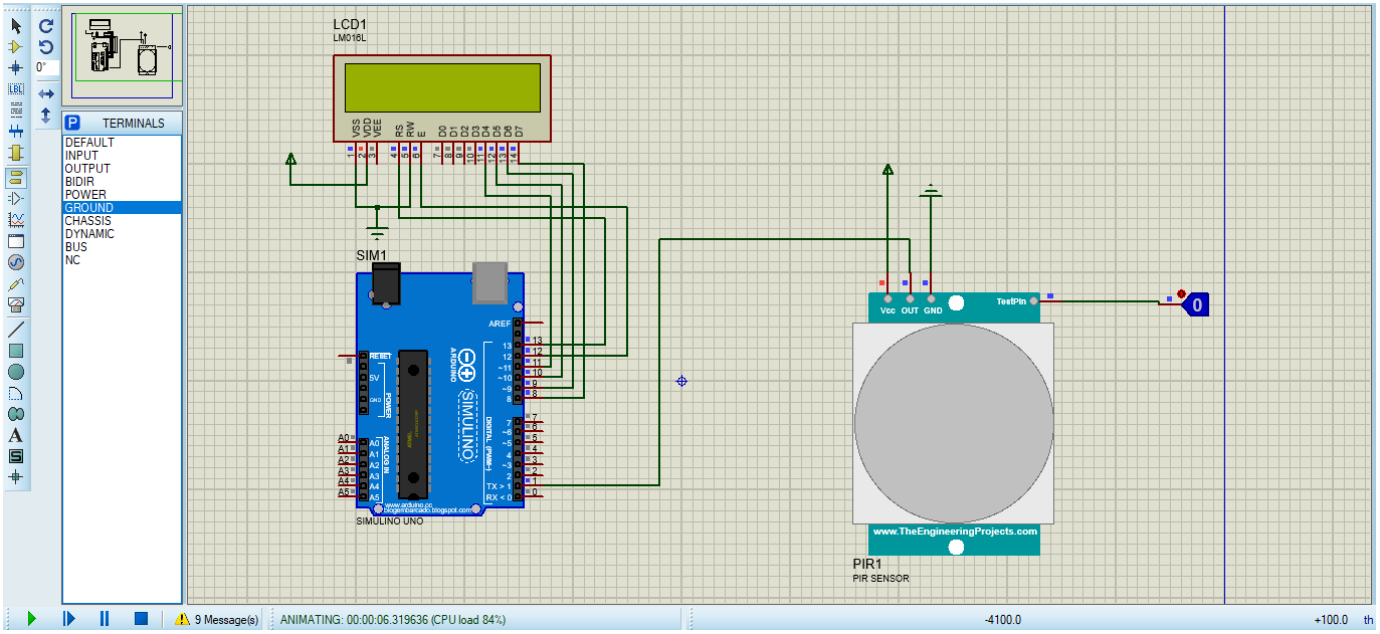


Figure 27: When the signal is low no message is displayed on the LCD showing there is no intrusion

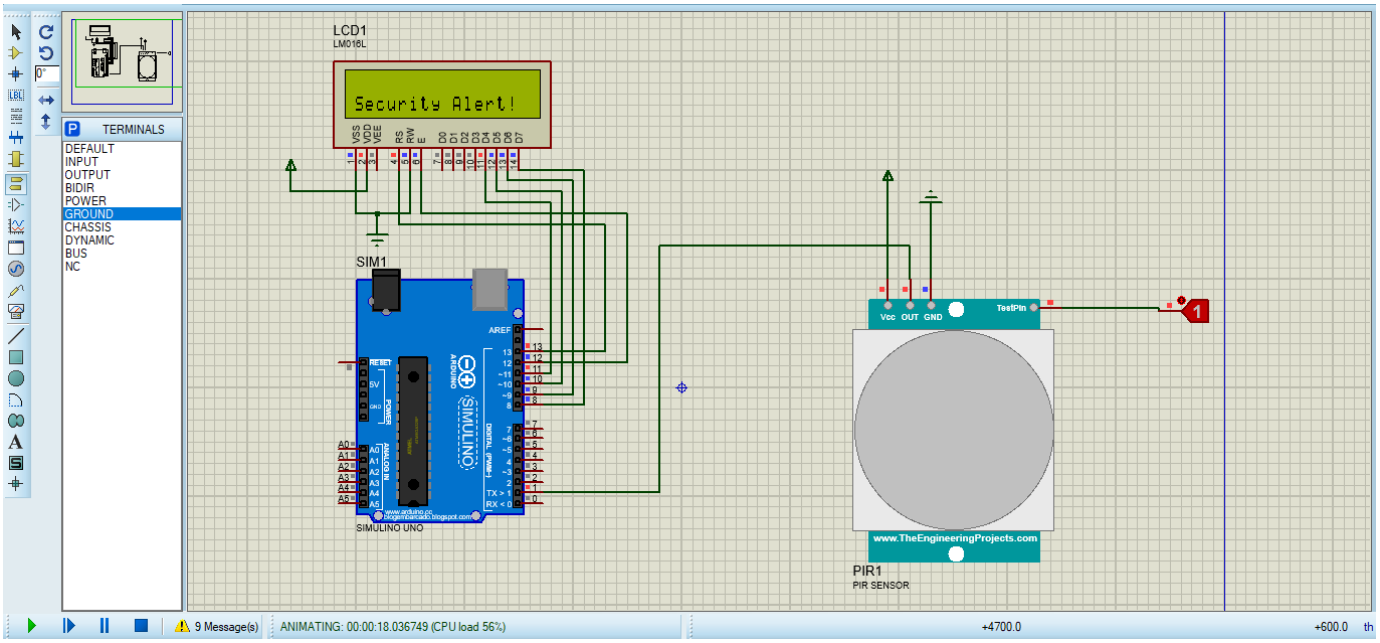


Figure 28: When signal is high the message is displayed as “Security alert!”

In figure 27 and figure 28, an LCD screen is attached to the Arduino board which gives the status. If nothing is in the PIR range then no signal is given, but if the PIR detects any signal, then a message stating "Security Alert!" is displayed. Here a Logic Toggle is used to set the signal from PIR high or LOW.

Table 4: Voltage level at buzzer and message status

Voltage level (Volts) at Vcc	0 (LOW)	5 (HIGH)
Status of Buzzer	OFF	ON
Message Sent to the user's mobile	-	Security Alert!

Vcc of the PIR is connected to the 5V pin of Arduino. The ground pin is grounded and the Output Pin of the PIR is connected to a Digital Pin of the Arduino. Thus when an intrusion occurs then a message is sent by the GSM module to the user's mobile stating that "Security Alert!" as coded.

9. CONCLUSION: -

The main objective of this project is to develop a home automation system using an Arduino board with Bluetooth being remotely controlled by any Android OS smart phone. We divided the project into two parts one being the automation part and the other being the security part.

The automation part as discussed is Bluetooth based and our user interface is the Bluetooth controller app which is installed in the mobile of the user and specified instructions are given by the user from the user interface which is transferred via Bluetooth to the microprocessor which in turn interprets the signal and accordingly directs the relays to perform the task, so that the devices can be controlled in the houses.

And the Security part is GSM based and a PIR sensor is used to detect this. Even if the user is not present in their house, they would be able to know if any security problems have occurred as the GSM module will send the message of a security breach. Also when a breach is detected a buzzer starts buzzing which can inform people nearby about the whereabouts of the house.

This is a prototype of housing technologies. Because as technology is advancing so houses are also getting smarter. Also we can add certain elements such as including a surveillance camera in the security system. Also not only are these type of projects viable for houses but with certain step-ups it can be used for commercial purposes also. Some aspects are even being used today like biometrics scan, motion sensors, smart cards, etc.

With the invention of lots of automation technologies, home automation has become a reality. With the advancement in IOT and Artificial intelligence the lives of people are going to get easier. These technologies can be used to build fully functional home automation system and control smart home devices including smart lights, connected thermostats and appliances.

Finally, we can say that the project has proposed the idea of smart homes that can support a lot of home automation systems. It has discussed the designed modules like sensors circuits and relays. There are a variety of enhancements that could be made to this system to achieve greater accuracy in sensing and detection such as adding other types of modules.

10. FUTURE REFERENCES

2014 to 2017 the sentiment towards smart home technology started to sway, with over 60% of consumers favouring the technology.

So, in the immediate future, we can expect a mass adaptation of smart home products with technology evolving to be more efficient, cost-effective and secured.

On big challenge that still stays on the way of the widespread adaptation of smart home is 'integration.' We can have a voice-controlled smart fan, remote controlled air purifier, a smart mirror, an indoor positioning system. But all the existing products have to be controlled separately, and till the time there's no integrated controlling protocol implemented, there's a little hope for smart homes to be available on a commercial scale.

Security is another issue which IoT, in general, is facing. So, in the future, we can expect more secured solution.

11. REFERENCES

[1] Tutorials point. (n.d.). Arduino tutorial. Retrieved from

https://www.tutorialspoint.com/arduino/arduino_tutorial.pdf

[2] P.Bhagyalakshmi, G.Divya, N.L.Aravinda, "Raspberry PI And Wifi Based Home Automation", International Journal of Engineering Research and Applications, pp. 57-60, January 2015

[3] Electronics Hub. (n.d.). Home automation tutorial. Retrieved from *https://www.electronicshub.org/arduino-based-home-automation/*

[4] Electronic Wings. (n.d.). Bluetooth module HC-05. Retrieved from

https://www.electronicwings.com/sensors-modules/bluetooth-module-hc-05-

[5] Instructables Circuit. (n.d.) How to connect relays to the Arduino board. Retrieved from

https://www.instructables.com/id/Home-Automation-How-to-Add-Relays-to-Arduino/

[6] Anandhavalli D, Noorul S. Mubina, Bharathi P, "Smart Home Automation Control Using Bluetooth and GSM", International Journal of Informative & Futuristic Research, pp. 2620-2624, April 2015

[7] Adafruit. (2014). Infrared proximity sensor. Retrieved from

https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor

[8] Instructables Circuit. (n.d.). Tutorial to Interface GSM SIM900A with Arduino. Retrieved from *https://www.instructables.com/id/GSM-SIM900A-With-Arduino/*

[9] E. Isa, N. Sklavos, "Smart Home Automation: GSM Security System Design & Implementation, JOURNAL OF Engineering Science and Technology Review, pp. 170-174, January 2016

[10] Ma Yuchun, Huang Yinghong, Zhang Kun, Li Zhuang, "General Application Research on GSM Module", 2011 International Conference on Internet Computing and Information Services, pp. 525-528, 2011