# TITLE

# HOME AUTOMATION USING IR COMMUNICATION


# GROUP MEMBERS

| NAME | ROLL NUMBER |
|---|---|
| TANMOY SENGUPTA | 10901617033 |
| SWAGATAM KAR | 10901617036 |
| SUMAN SAURABH | 10901617040 |
| SUBHASIS SAMANTA | 10901617043 |
| VIDYA KUMARI | 10901617031 |

Signature of Trainer

Signature of HOD

Signature of course coordinator

# Objective

- Perform ON/OFF operation of LED Bulb , based on button 9 of IR remote.

- Perform ON/OFF operation of LAMP, based on button 7 of IR remote.

# PRE-REQUISITES
## (KEY COMPONENTS USED)

- **ADRUINO UNO**

The **Arduino Uno** is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc .The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The board has 14 digital I/O pins (six capable of PWM output), 6 analog input pins, and is programmable with the Arduino IDE (Integrated Development Environment), via a type B - USB cable. It can be powered by the USB cable or by an external 9-volt battery, though it accepts voltages between 7 and 20 volts. It is the most commercially used product, and also is similar to the **Arduino Nano** and **Leonardo**.

**TECHNICAL SPECIFICATIONS :-**

*Operating Voltage:5 Volts*

*Input Voltage:7 to 20 Volts*

*Digital I/O Pins:14 (of which 6 can provide PWM output)*

*Analog Input Pins: 6*

*DC Current per I/O Pin: 20 mA*

*DC Current for 3.3V Pin:50 mA*

*Flash Memory:32 KB of which 0.5 KB used by boot loader*

*SRAM:2 KB*

*EEPROM:1 KB*

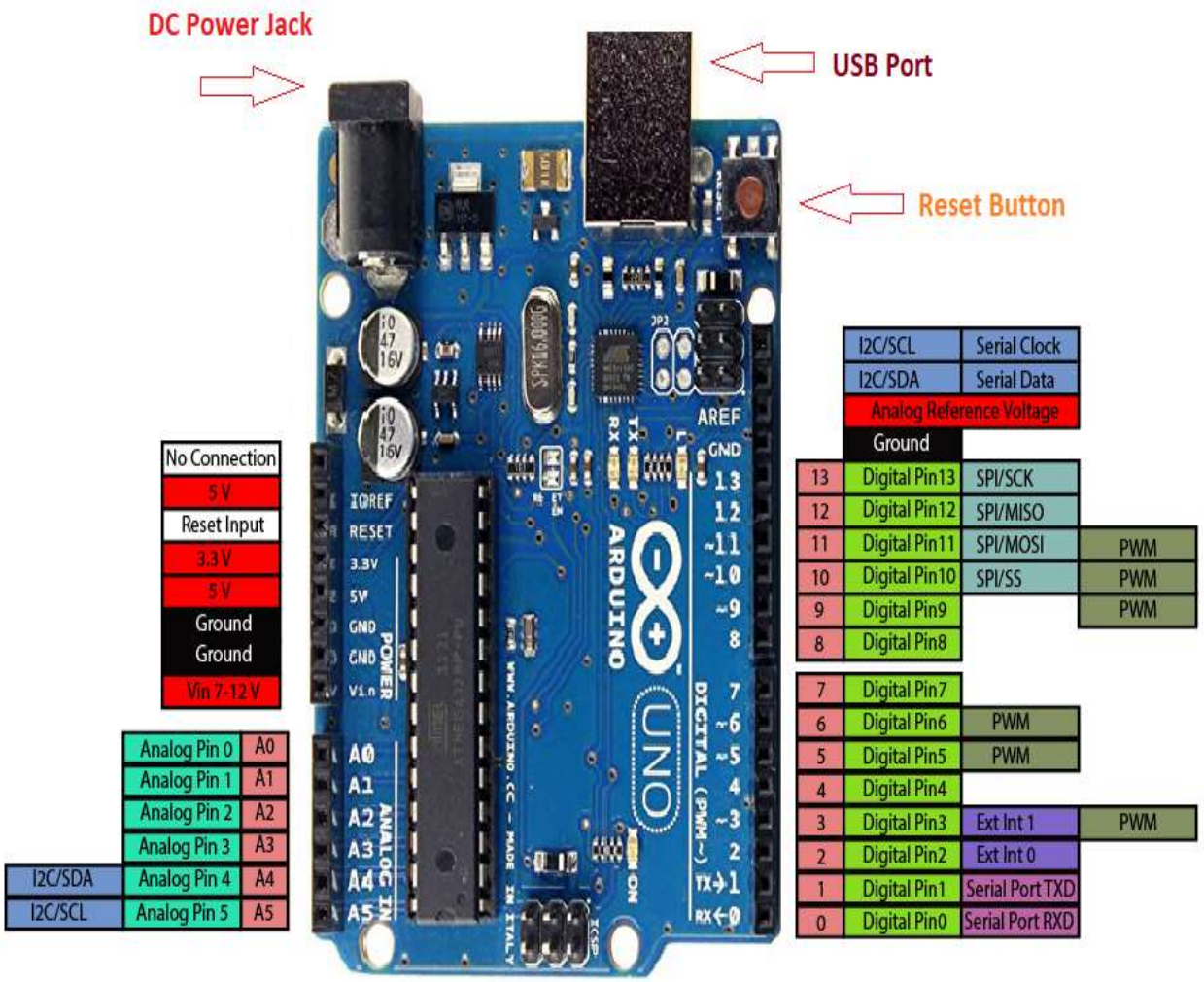
*Clock Speed:16 MHz*

*Length: 68.6mm*

*Width:53.4 mm*

*Weight:25g*

# ARDUINO UNO PIN-OUT DETAILS



## Special pin functions

Each of the 14 digital pins and 6 analog pins on the Uno can be used as an input or output, under software control (using pinMode(), digitalWrite(), and digitalRead() functions). They operate at 5 volts. Each pin can provide or receive 20 mA as the recommended operating condition and has an internal pull-up resistor (disconnected by default) of 20-50K ohm. A maximum of 40mA must not be exceeded on any I/O pin to avoid permanent damage to the microcontroller. The Uno has 6 analog inputs, labeled A0 through A5; each provides 10 bits of resolution (i.e. 1024 different values).

By default, they measure from ground to 5 volts, though it is possible to change the upper end of the range using the AREF pin and the analogReference() function. In addition, some pins have specialized functions:

**Serial** / UART: pins 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL serial chip.

**External interrupts**: 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** (pulse-width modulation): pins 3, 5, 6, 9, 10, and 11. Can provide 8-bit PWM output with the analogWrite() function.

**SPI** (Serial Peripheral Interface): pins 10 (SS), 11 (MOSI), 12 (MISO), and 13 (SCK). These pins support SPI communication using the SPI library.

**TWI** (two-wire interface) / I²C: pin SDA (A4) and pin SCL (A5). Support TWI communication using the Wire library.

**AREF** (analog reference): Reference voltage for the analog inputs.

## Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino board, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The 16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a (.inf ) file is required. Arduino Software (IDE) includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1). A Software Serial library allows serial communication on any of the Uno's digital pins.

# IR RECEIVER (TSOP1738)

**The TSOP sensor has the ability to read the output signals from home remotes like TV remote, Home theatre remote, AC remote etc.. All these remotes will work with a frequency of 38kHz and this IC can pick up any IR signals process them and provide the output on pin 3. So if you are looking for a sensor to analyze , re-create or duplicate the functions of a remote then this IC will be the perfect choice for you.**
**Also keep in mind that this series TSOP-1738 will receive only 38Khz IR signals. All remotes in India will operate in 38Khz, kindly ensure if it is the same in your country.**

## PINS

| Pin Number | Pin Name | Description |
|---|---|---|
| 1 | Ground | Connected to the Ground of circuit |
| 2 | VCC | Typically connect to +5V, maximum of 6V can be given |
| 3 | Signal | The signal pin gives out the sequence based on the IR signal detected |


FIG- PINOUT DETAILS OF IR-RECEIVER

**TSOP-1738 Characteristics :--------**

✓Minimum and Maximum Input Voltage is -0.3 and 5V respectively. Typically +5V is used.
✓Can detect IR signals from Remotes (38kHz)
✓Operating current: 5mA
✓High Range and wide coverage area.
✓Will respond only to IR signals, due to high immunity against ambient light
✓Low power consumption
✓Has in-built pre amplifier
✓TTL and CMOS compatible

TSOP17XX receives the modulated Infrared waves and changes its output. TSOP is available in many frequency ranges like TSOP1730, TSOP1738, TSOP1740 etc. Last two digits represent the frequency (in KHz) of modulated IR rays, on which TSOP responds. Like for example TSOP1738 reacts when it receives the IR radiation modulated at 38Khz. Means it detects the IR which is switching On and Off at the rate of 38Khz. TSOP's output is active low, means its output is remains HIGH when there is no IR, and becomes low when it detects IR radiation. TSOP operates on particular frequency so that other IRs in the environment can't interfere, except the modulated IR of particular frequency. It has three pins, Ground, Vs (power), and OUTPUT PIN.

# RELAY MODULE (5V - DC)

The Single Relay Board can be used to turn lights, fans and other devices on/off while keeping them isolated from your microcontroller. The Single Relay Board allows you to control high-power devices (up to 10 A) via the on-board relay. Control of the relay is provided via a 1 x 3 header – friendly to servo cables and convenient to connect to many development boards.

**TECHNICAL SPECIFICATIONS** :-

*Supply Voltage-5V*

*Control high-power devices up to 10 A with a simple high/low signal*

*Provides isolation between the microcontroller and device being controlled*

*Screw terminals for relay connections*

*3-pin servo-style header for power/signal interface*

*Equipped with a high-current relay (10A @ 28VDC)*

*2x LED's that show the current state of the relay*

# COMPONENTS USED

- Single Channel Relay Module * (1 No.s)
- Arduino Uno * (1 No.s)
- IR Receiver (TSOP1738) * (1 No.s)
- LED Bulb * (1 No.s)
- LAMP (230V AC) * (1 No.s)
- Bread board * (1 No.s)
- Jumper cables * (1 Lot)

# DIAGRAMS


ARDUINO UNO

IR SENSOR

RELAY MODULE

# Circuit Diagram- I

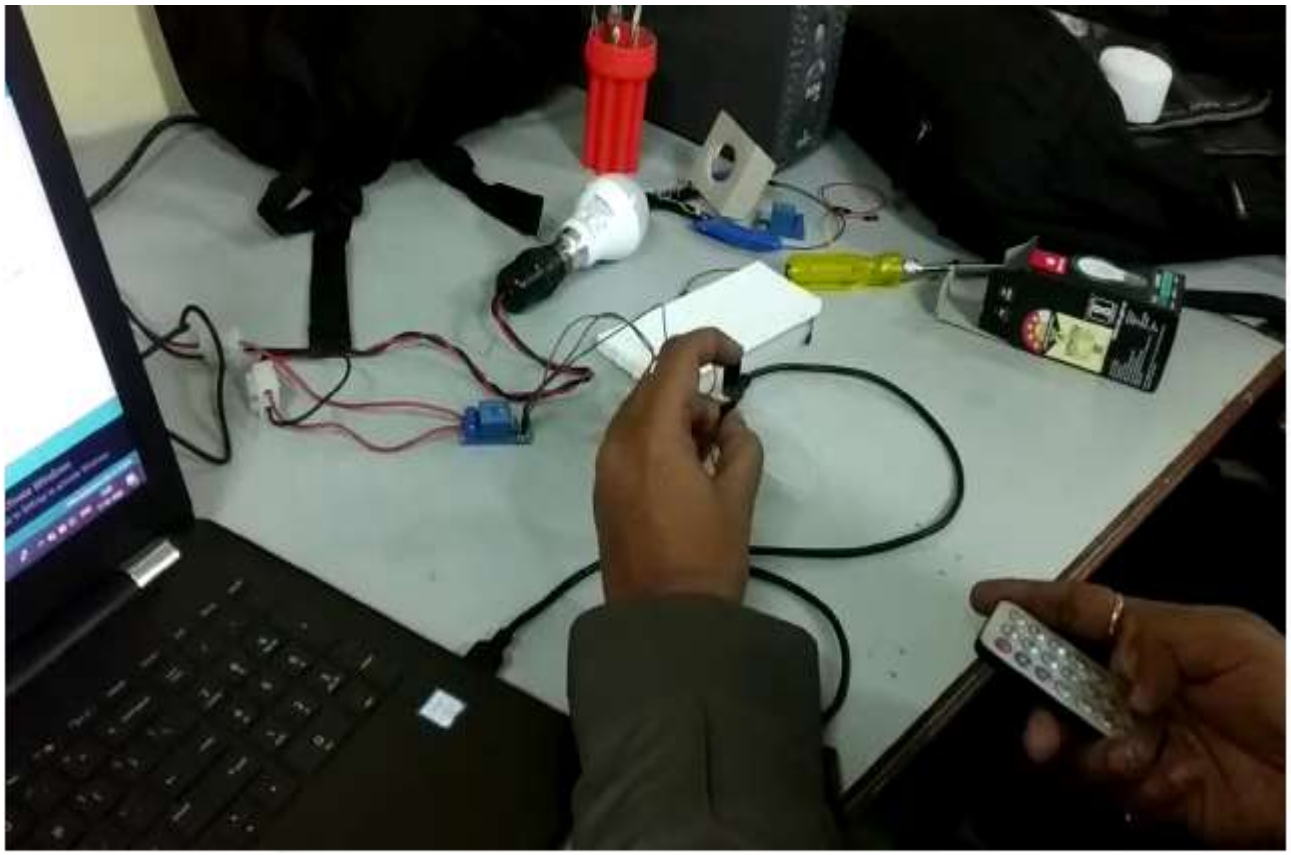


**Fig:- Both Lamp and led at low state.**

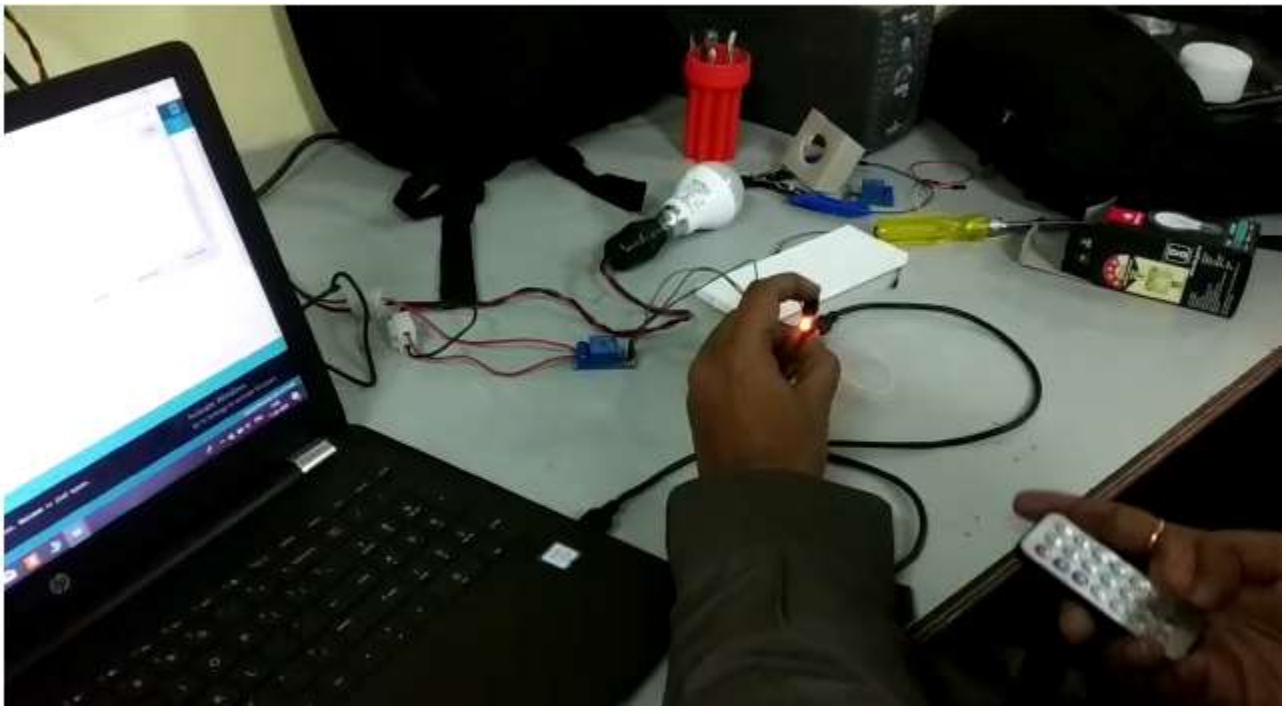


**Fig:- Led at High State and Lamp at low**

# Circuit Diagram- II

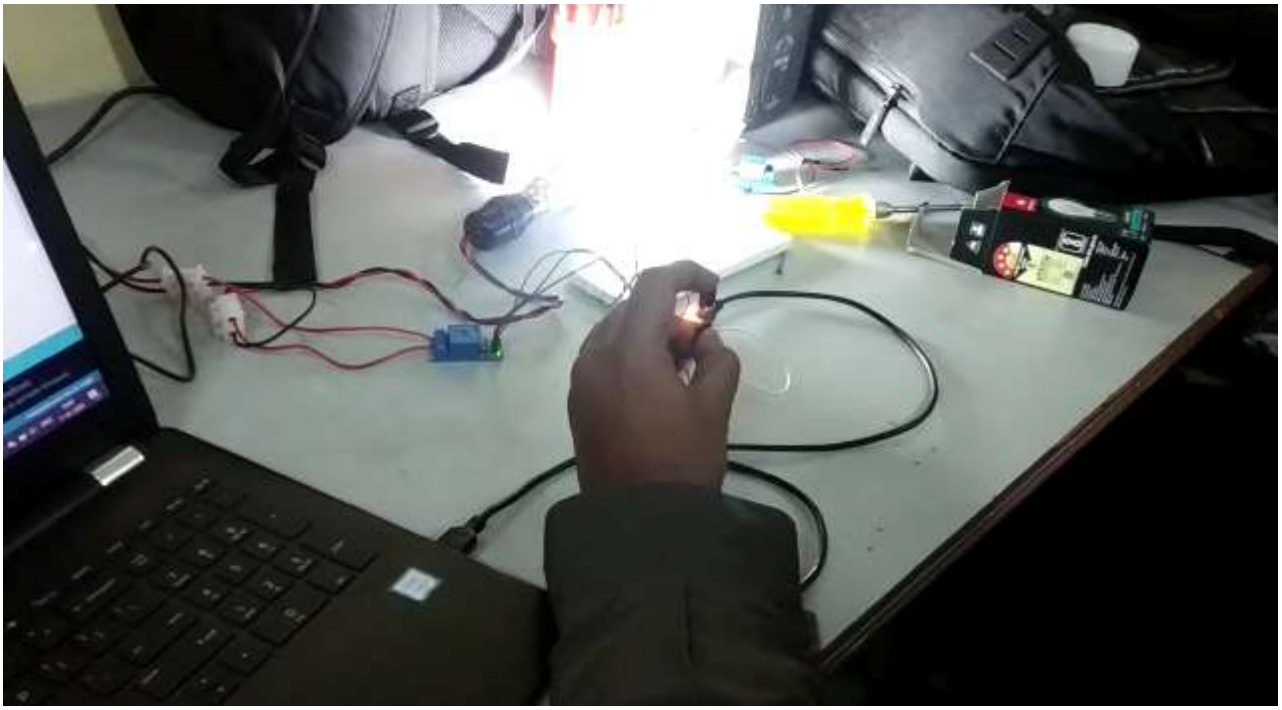


**Fig:- Both Lamp and led at High state.**

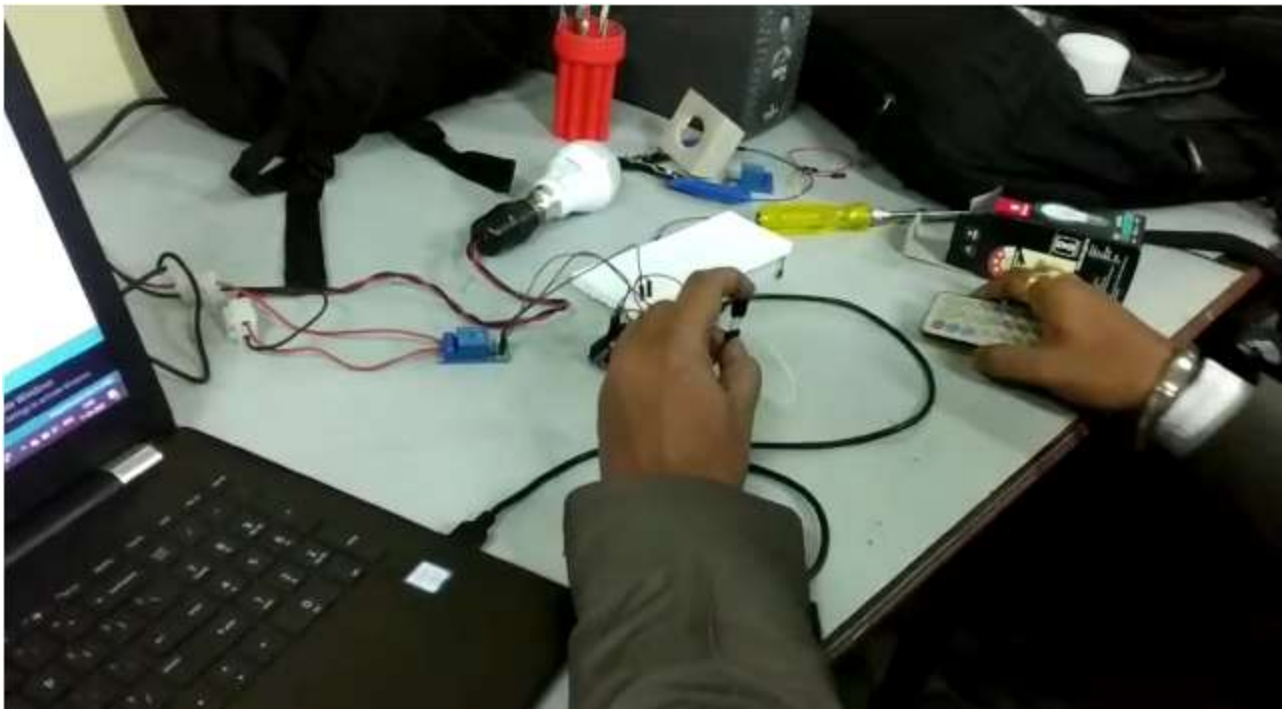


**Fig:- Both Lamp and led at low state.**

# Connection Diagram

| Arduino | IR RECEIVER |
|---|---|
| 5v | Vcc |
| GND | GND |
| 10(Digital) | RECV_PIN |

Fig-Pin Diagram For Arduino UNO and IR RECEIVER

| Arduino | Relay |
|---|---|
| 5v | Vcc |
| GND | GND |
| 8(Digital) | IN |

Fig-Pin Diagram For Arduino UNO and Relay.

# Programming Code

```
  #include <IRremote.h>

int RECV_PIN = 10;
int r=8;
int led=13;
int i=1;
int j=1;
++
IRrecv irrecv(RECV_PIN);

decode_results results;

void setup()
{

  Serial.begin(9600);
  irrecv.enableIRIn();
  pinMode(r, OUTPUT);
  pinMode(led, OUTPUT);
}
void loop() {
  if (irrecv.decode(&results)) {
    Serial.println(results.value, HEX);
    if (results.value == 0x1FE00FF)   //button 7
    {
      i++;
      int a =i%2;
      digitalWrite(r,a);
    }
    else if(results.value == 0x1FE9867) //button 9
    {
      j++;
      int b=j%2;
      digitalWrite(led,b);
    }

    irrecv.resume(); // Receive the next value
  }
}
```

# Conclusion

So, at last but not the least we would like say that ,if you want optimization of any task ,go for Automation because it not only reduces human interaction but also Improves the output , save times(since it consumes less time ). And we all observed above advantages during the “Automation of Home using IR sensor” application. Hence, in order to be parallel with the running world, one should prefer automation over manual working.

# Thank You