# STEPS FOR EXCUTION

## Implementation:

Here 1st we need to load the code into NodeMCU and then we need to connet the NodeMCU with power and WIFI connection so that it works fine.

After the code gets compiled successfully we need to open the telegram bot.

After opening the Telegram Bot we need to click on START. Now the bot starts replying the user. Bot keeps the commands which the user can use. The user needs to click on the command so that the command will work.

If it is Light On then the light which we connected will ON.

If it is Light Off then the light which we connected will go OFF.

If it is Byee then it Sends off the user with a message.

If it is Help me the loop will be continued.

## Algorithm:

**Step 1:** Start

**Step 2:** Check whether WIFI is connected to NODEMCU , if WIFI connected goto Step3. Else goto Step 9.

**Step 3:** Code in NodeMCU is processed and now telegram bot will be active.

**Step 4:** If the Telegram access token is true then goto Step 5.Else goto Step 9.

**Step 5:** /start

**Step 6:** /helpme

**Step 7:** Now 4 commands will be displayed. /LIGHT_ON,/LIGHT_OFF,/helpme,/bye

**Step 8:** If /LIGHT_ON:

Light will on and bot replies as LIGHT will be ON.

Else If /LIGHT_OFF:

Light off and bot replies ad LIGHT will OFF.

Else If /helpme :

Goto Step 6.

Else If /bye:

Nothing happens and bot replies as Byee See you soon.

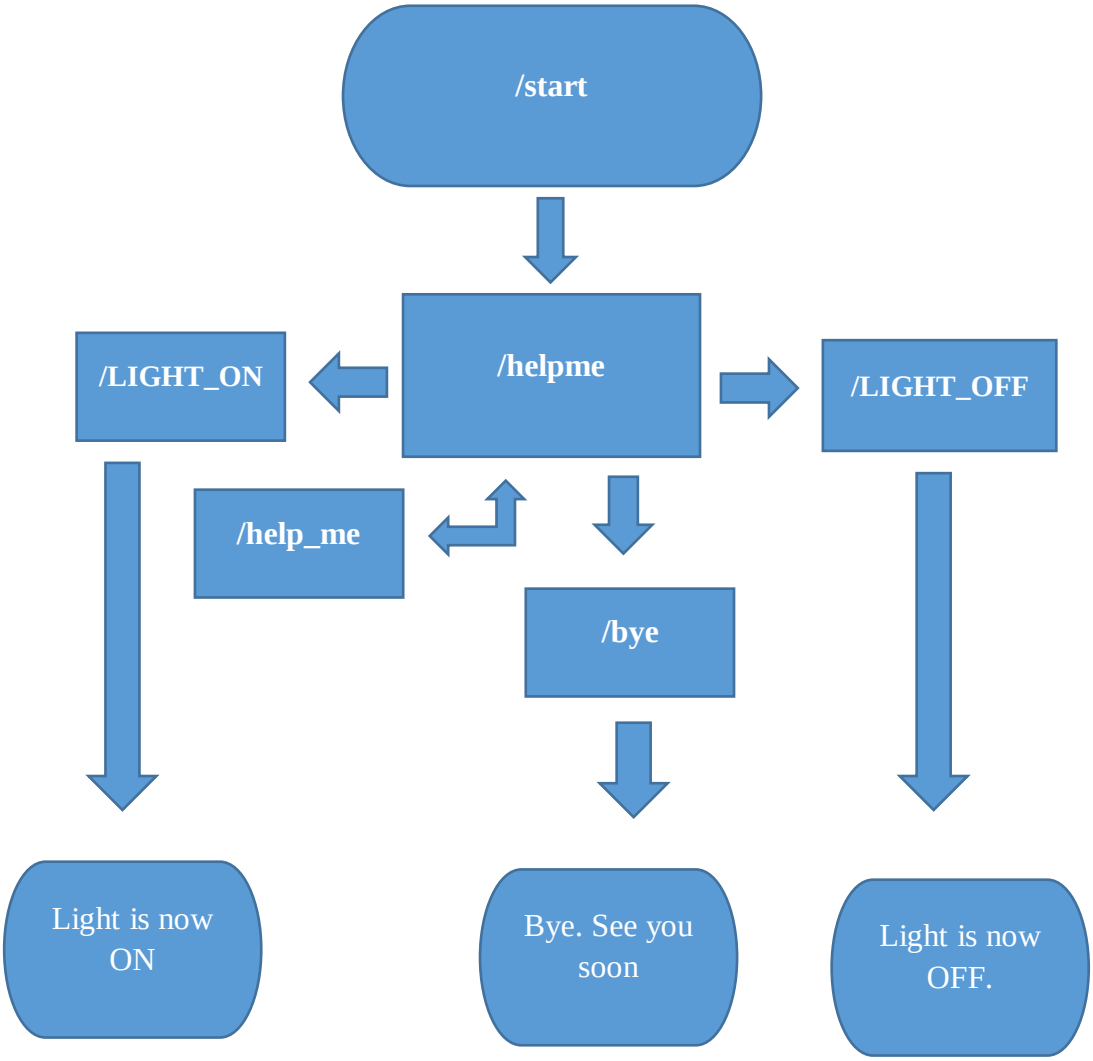
Else:

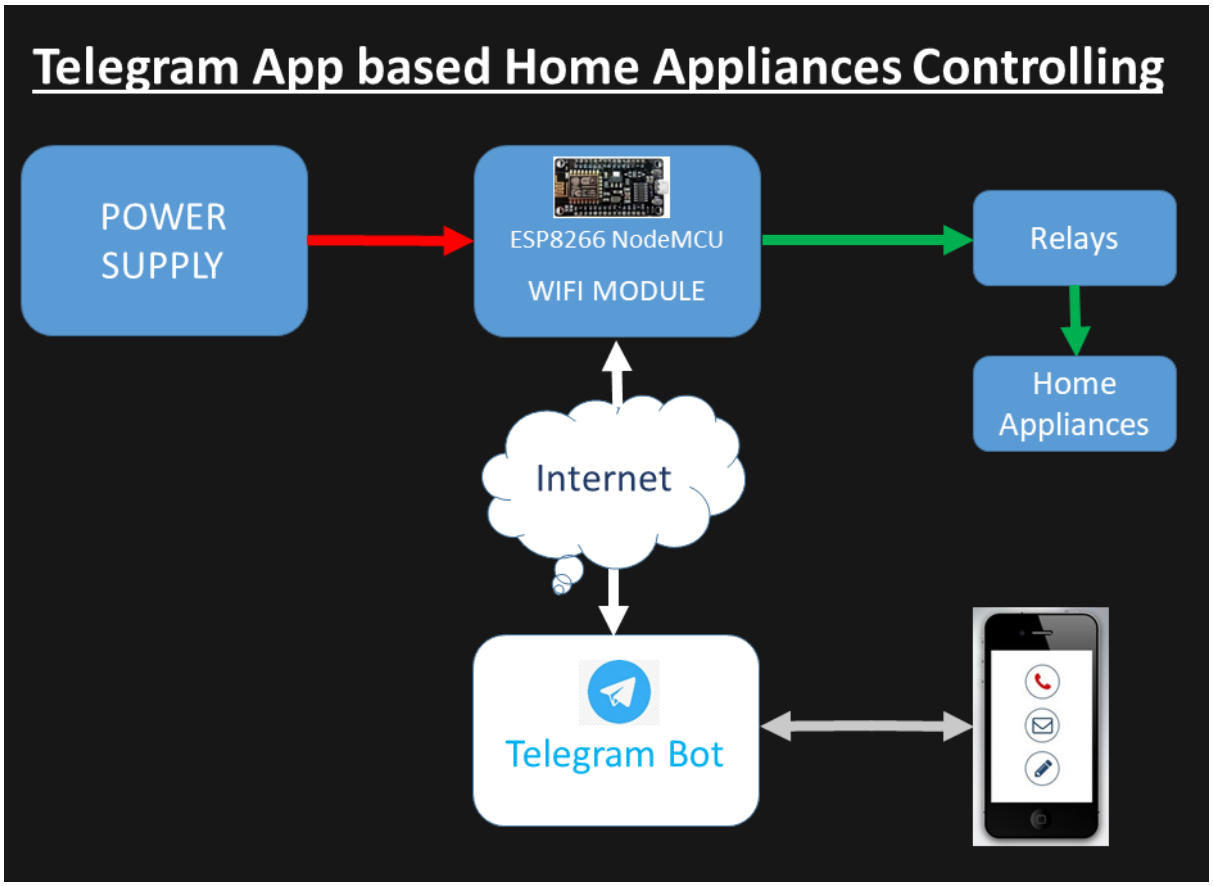
Goto Step 4.

**Step 9:** End.

## Flow Chart:

The below is the flow chart designed for our project.




Telegram App based Home Appliances Controlling
POWER SUPPLY
ESP8266 NodeMCU
WIFI MODULE
Relays
Home Appliances
Internet
Telegram Bot

## Code Explanation

**Complete code with a working video** for this IoT controlled Home Automation is given at the end of this tutorial, here we are explaining the complete program to understand the working of the project.

Replace SSID and password with your Wi-Fi credentials. String token stores the unique token number you got after creating your bot. D0 is the digital output pin for changing the state of the relay.

```
String ssid = "**********";     // Replace with your ssid

String pass = "************"; // Replace with your password

String token = "66****885:A*G-X**dTYdSCt******aCQPCk***SL**b4";   // token number
of your bot

int led = D0;    // digital pin on NodeMCU
```

*myBot.wifiConnect()* function takes the SSID and password to connect ESP8266 with the Wi-Fi and function *myBot.setTelegramToken()* takes the unique token number which further establishes the connection between telegram bot and NodeMCU. On successful connection you get "*testConnection OK*" on the serial monitor.

```
myBot.wifiConnect(ssid, pass);

myBot.setTelegramToken(token);

if (myBot.testConnection())


                    Serial.println("\ntestConnection OK");

                Else

                    Serial.println("\ntestConnection NOK");
```

*TBMessage* stores the message received from the telegram bot. *msg.text.equalsIgnoreCase()* checks if the text received matches with string "light on" or "light off". If the message received is "Light on" it changes the relay state to turn on the lamp and if the message received is "Light off" then it changes the relay state to turn off the lamp.

If the text received is different from “light on” and “light off” then it sends welcome message back to the telegram.

```
TBMessage msg;

if (myBot.getNewMessage(msg)) {

   if (msg.text.equalsIgnoreCase("LIGHT ON")) {

       digitalWrite(led, HIGH);

          myBot.sendMessage(msg.sender.id, "Light is now ON");

            }

            else if (msg.text.equalsIgnoreCase("LIGHT OFF")) {

              digitalWrite(led, LOW);

                 myBot.sendMessage(msg.sender.id, "Light is now OFF");

                    }

         else {

           String reply;

            reply = (String)"Welcome " + msg.sender.username + (String)". Try LIGH
T ON or LIGHT OFF.";

          myBot.sendMessage(msg.sender.id, reply);

       }


  }
```

Now after completing the code its time to test the system.

## Source Code:

```
#include "CTBot.h"

CTBot myBot;


String ssid = "VARSHA";     // REPLACE mySSID WITH YOUR WIFI SSID
```

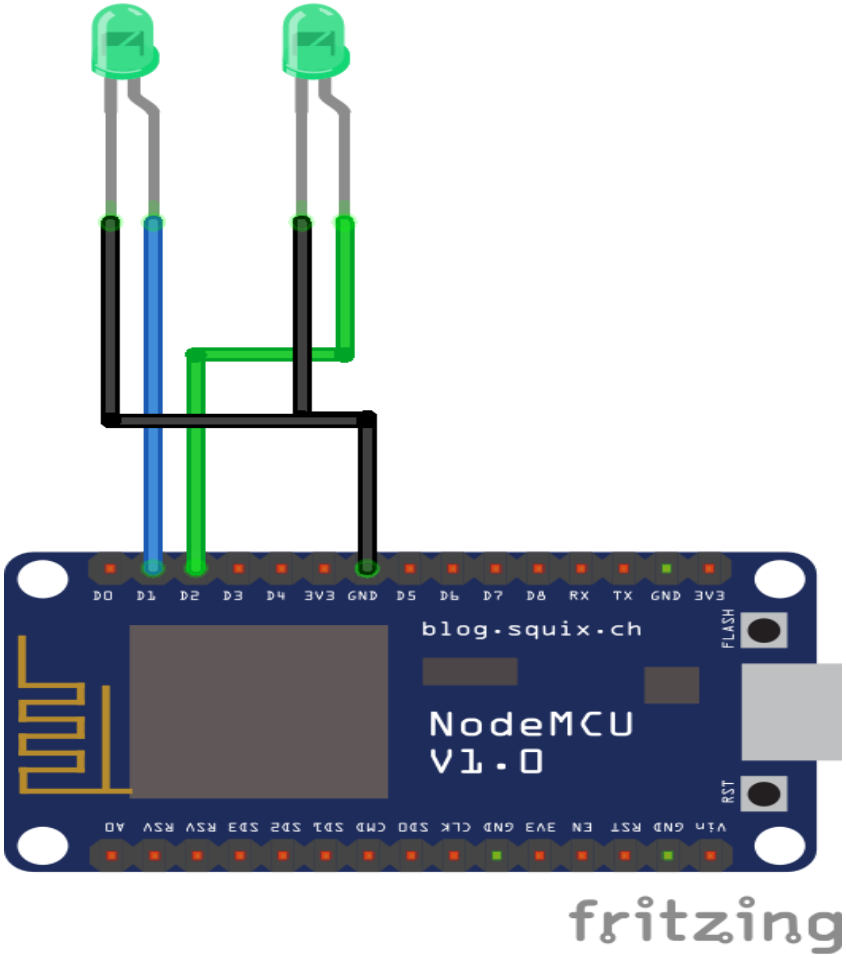
```
String pass = "9866028800";
 // REPLACE myPassword YOUR WIFI PASSWORD, IF ANY
String token = "5482254733:AAFeAVFLEtt5c83Sc6rE9oyiP2sD5E1By5Y";   // REPLACE
myToken WITH YOUR TELEGRAM BOT TOKEN
uint8_t led = D0;           // the onboard ESP8266 LED.
                            // If you have a NodeMCU you can use the BUILTIN_LED pin
                            // (replace 2 with BUILTIN_LED)

void setup() {
    // initialize the Serial
    Serial.begin(115200);
    Serial.println("Starting TelegramBot...");

    // connect the ESP8266 to the desired access point
    myBot.wifiConnect(ssid, pass);

    // set the telegram bot token
    myBot.setTelegramToken(token);

    // check if all things are ok
    if (myBot.testConnection())
        Serial.println("\ntestConnection OK");
    else
        Serial.println("\ntestConnection NOK");

    // set the pin connected to the LED to act as output pin
    pinMode(led, OUTPUT);
    digitalWrite(led, HIGH); // turn off the led (inverted logic!)

}
```

```
void loop() {
  // a variable to store telegram message data
  TBMessage msg;

  // if there is an incoming message...
  if (myBot.getNewMessage(msg)) {

    if (msg.text.equalsIgnoreCase("/LIGHT_ON")) {          // if the received message is
"LIGHT ON"...
      digitalWrite(led, LOW);                        // turn on the LED (inverted logic!)
      myBot.sendMessage(msg.sender.id, "Light is now ON 💡 ");  // notify the sender
    }
    else if (msg.text.equalsIgnoreCase("/LIGHT_OFF")) {       // if the received message is
"LIGHT OFF"...
      digitalWrite(led, HIGH);                       // turn off the led (inverted logic!)
      myBot.sendMessage(msg.sender.id, "Light is now OFF"); // notify the sender
    }
    else if (msg.text.equals("/helpme")) {
   digitalWrite(led, HIGH);
   myBot.sendMessage(msg.sender.id, "Try sending following commands\n");
   myBot.sendMessage(msg.sender.id, "/LIGHT_ON\n");
   myBot.sendMessage(msg.sender.id, "/LIGHT_OFF\n");
   myBot.sendMessage(msg.sender.id, "/helpme\n");
   myBot.sendMessage(msg.sender.id, "/Byee\n");
  }
  else if (msg.text.equals("/Byee")){
        String r;
        r = (String)"Byeeee  " + msg.sender.username + (String)" .See you soon 👋  !!! ";
        myBot.sendMessage(msg.sender.id, r);
  }
    else {                                   // otherwise...
```

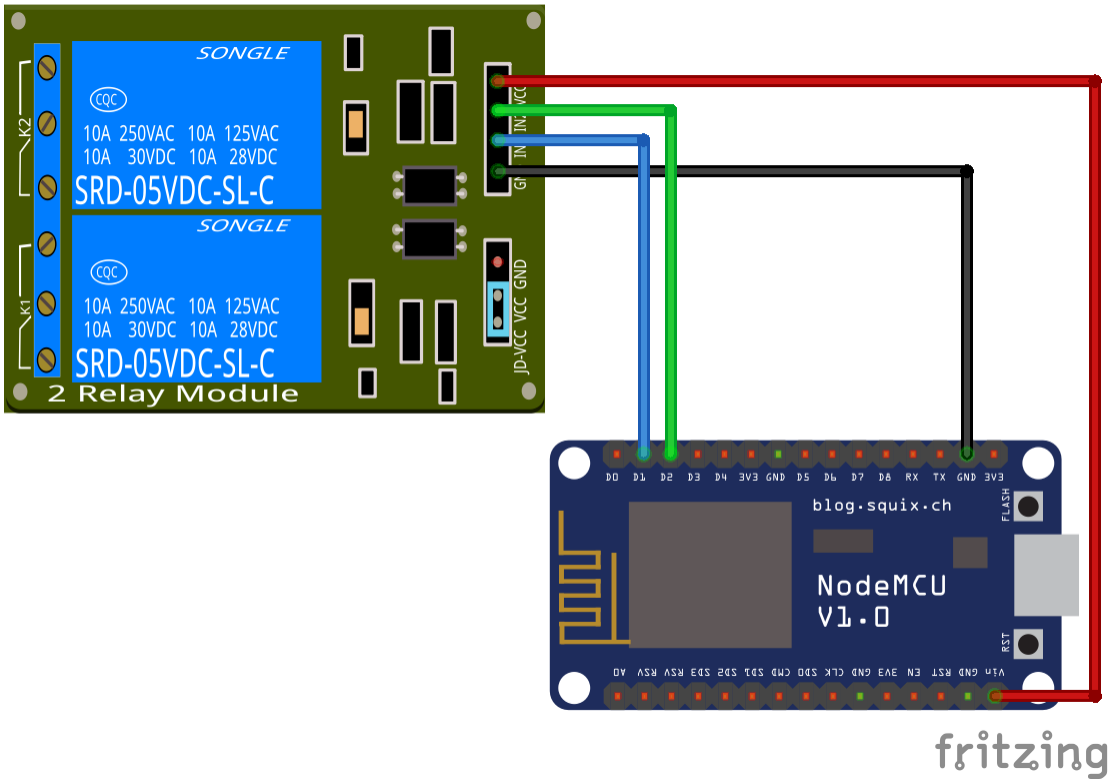
```
        // generate the message for the sender
        String reply;
        reply = (String)"Welcome  😄 " + msg.sender.username + (String)".Try /helpme";
        myBot.sendMessage(msg.sender.id, reply);            // and  send it
      }
    }
    // wait 500 milliseconds
    delay(500);
}
```

## Circuit Diagrams:

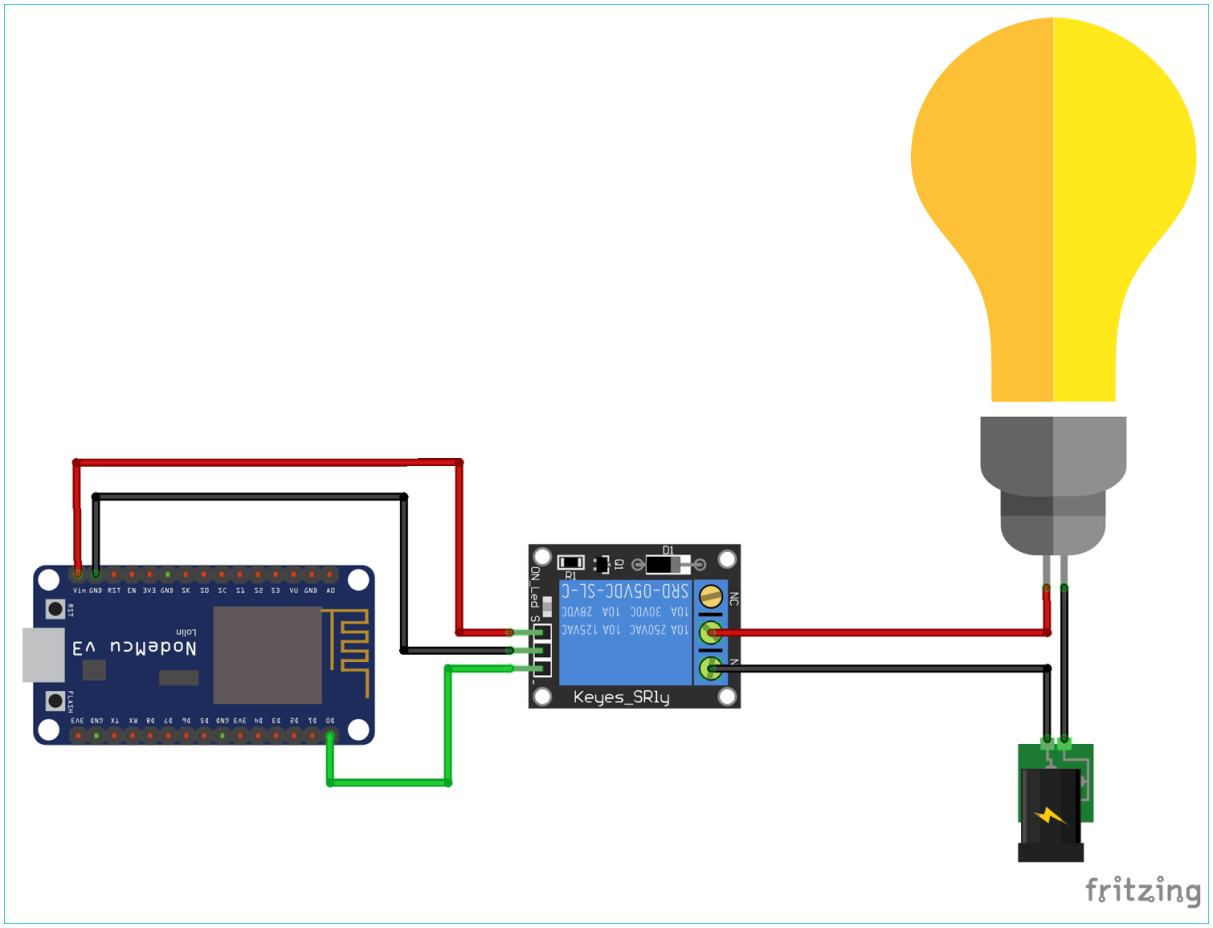
## Node MCU



**Relay Module:**



## Project Circuit Diagram:

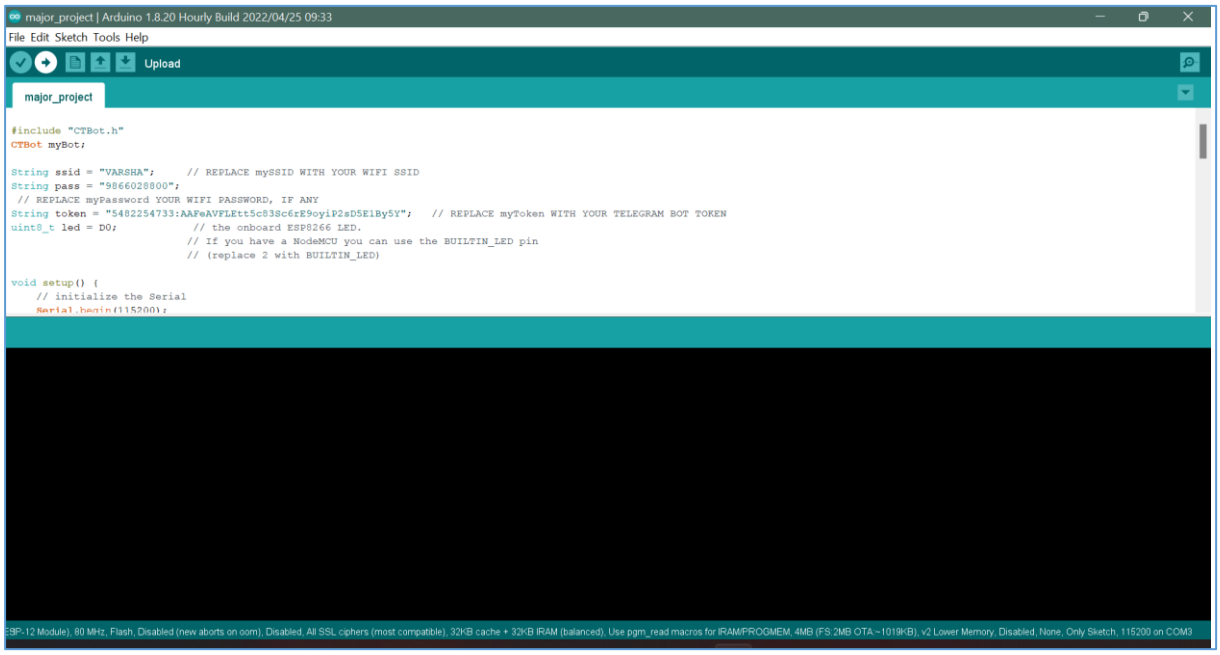


# Output:

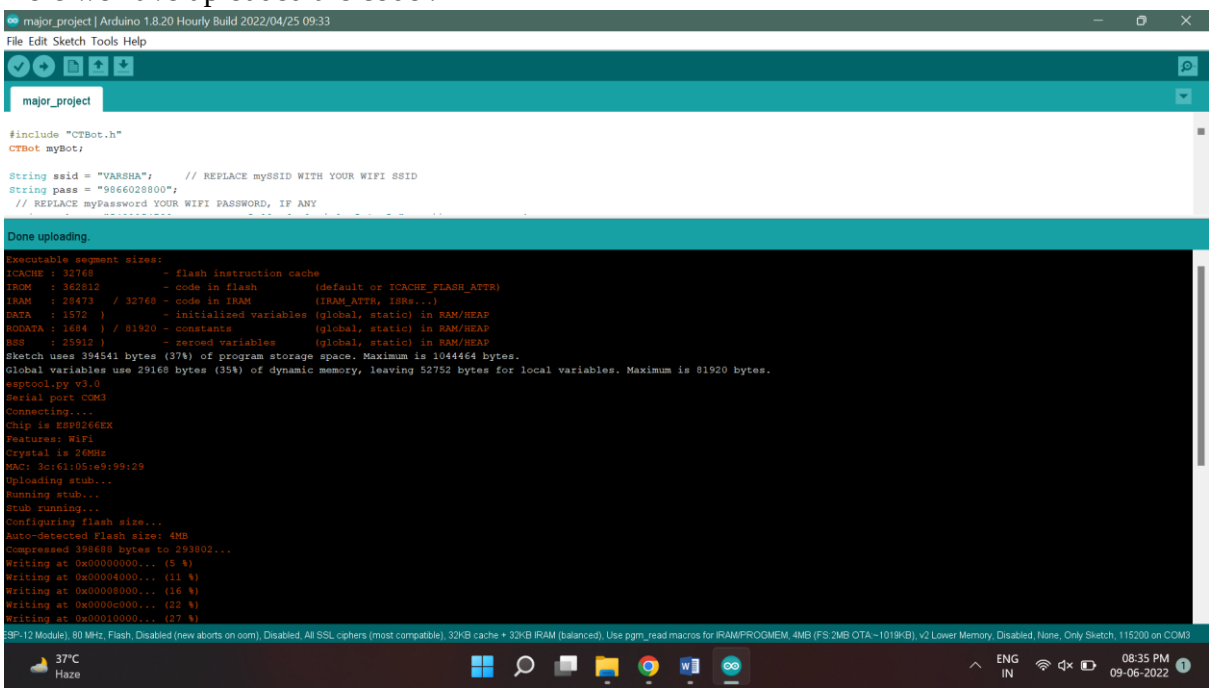
## (a) Software:

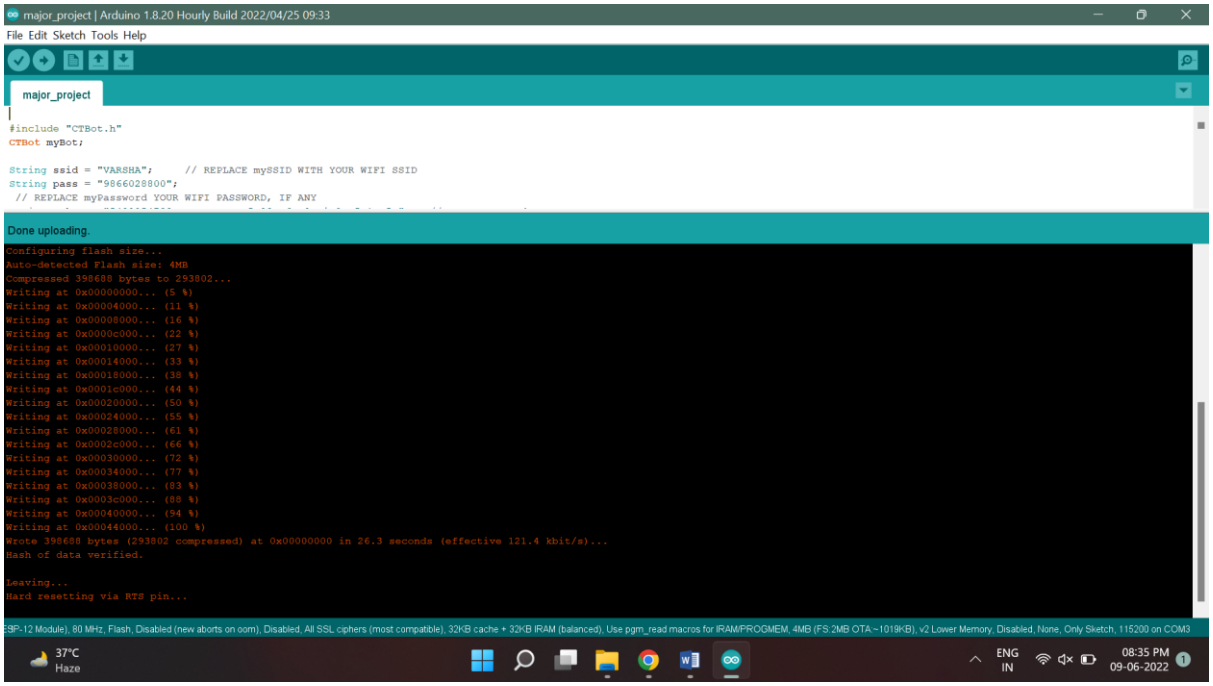
## (i)Arduino IDE:

Firstly we need to select the board and tools in the arduino which we downloaded earlier.



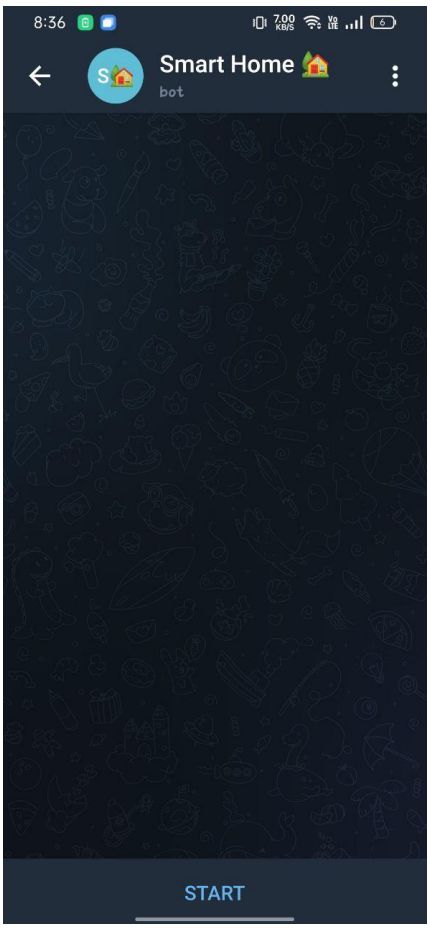

Here we have uploaded the code .





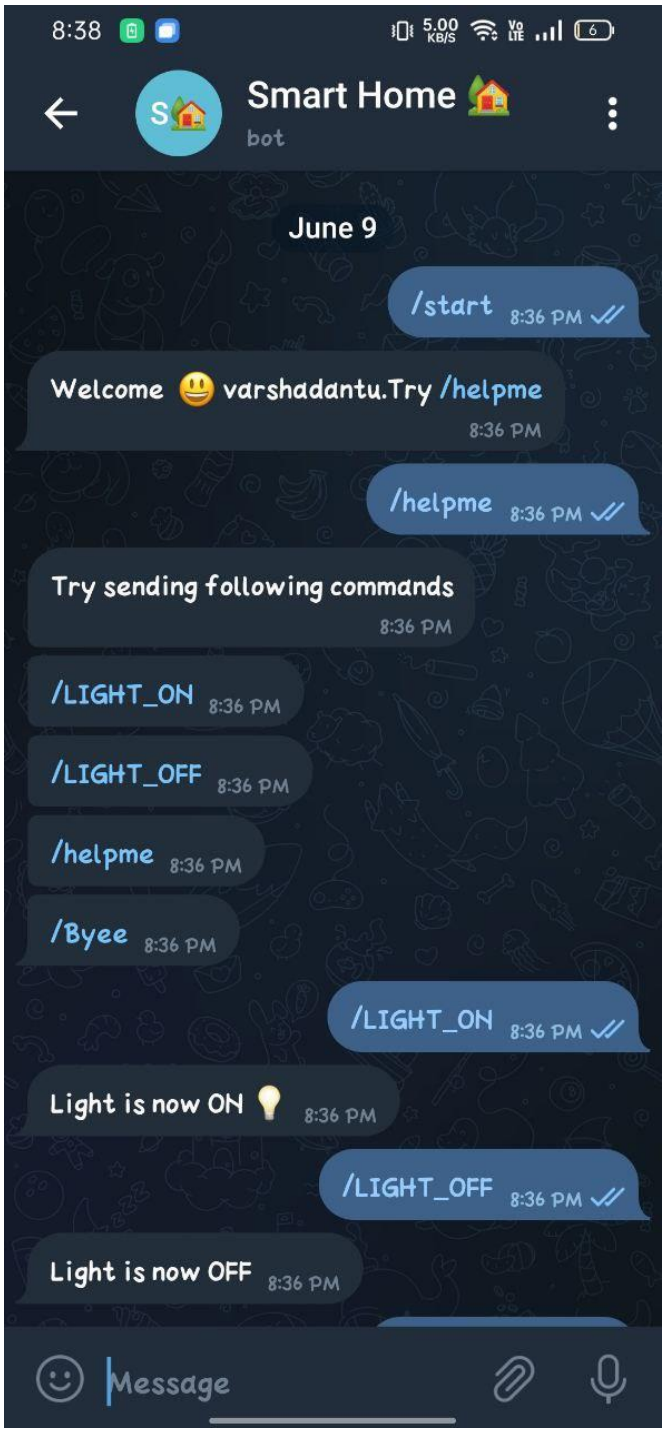

Now the code compiled successfully and done uploading into the NODEMCU.

## (ii) Telegram Bot:

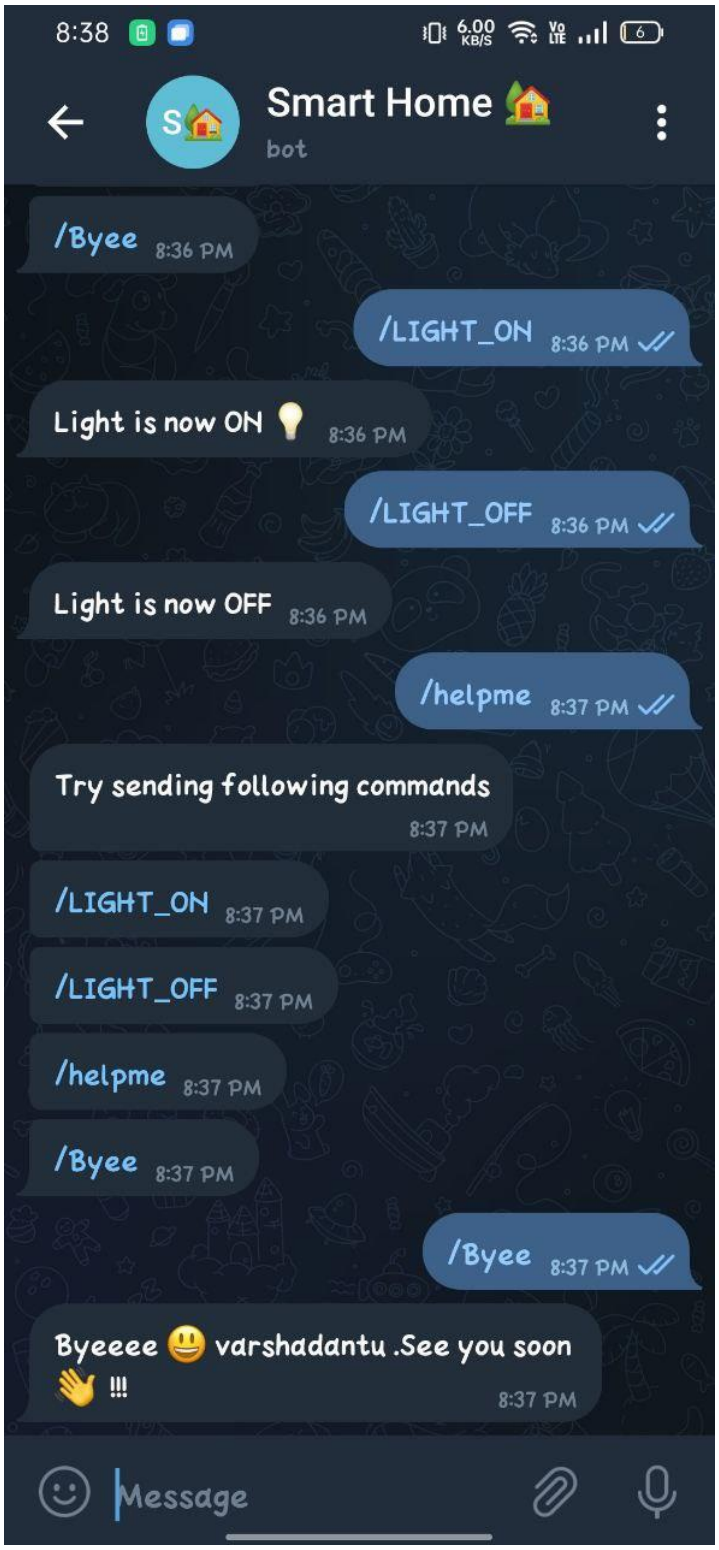


/start → /helpme → /LIGHT_ON → Light is now ON.

/start → /helpme → /LIGHT_OFF → Light is now OFF.



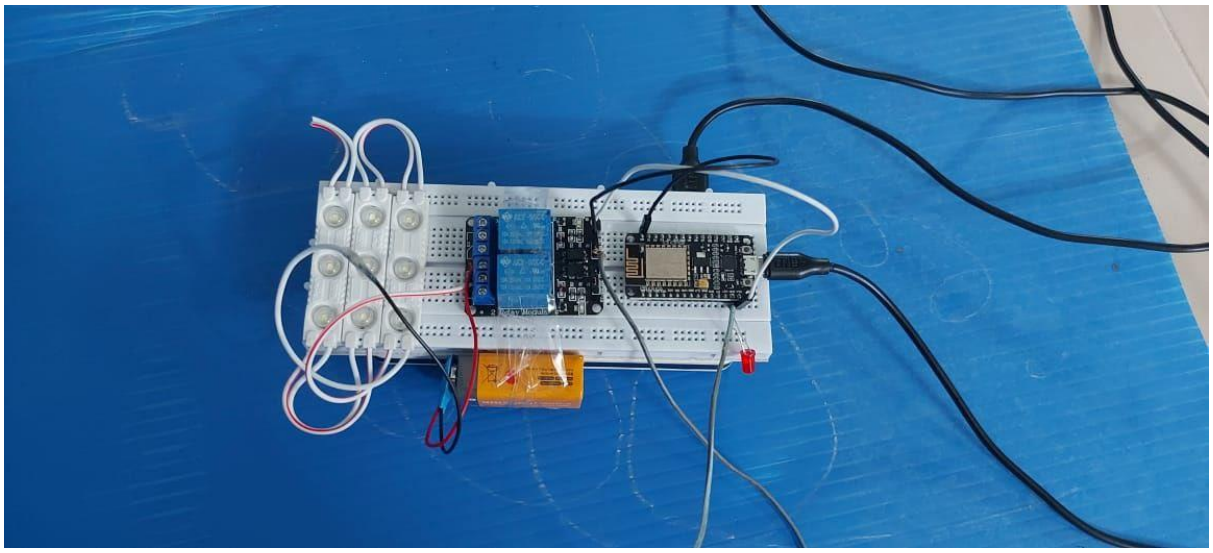

/start→ /helpme→/Byee → (username). See you soon.


8:38
Smart Home
bot
/Byee 8:36 PM
/LIGHT_ON 8:36 PM
Light is now ON 8:36 PM
/LIGHT_OFF 8:36 PM
Light is now OFF 8:36 PM
/helpme 8:37 PM
Try sending following commands 8:37 PM
/LIGHT_ON 8:37 PM
/LIGHT_OFF 8:37 PM
/helpme 8:37 PM
/Byee 8:37 PM
/Byee 8:37 PM
Byeeee varshadantu .See you soon !!! 8:37 PM
Message

## (iii) Hardware:

Whole figure.

Light Off:



Light On: