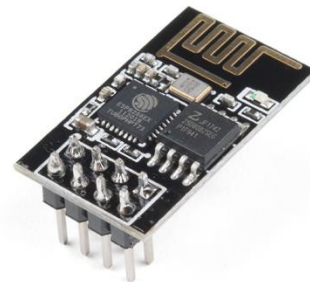


IoT CONTROLLED SMART HOME

USING ARDUINO AND ESP8266

GOOGLE ASSISTANT CONTROLLED

MAJOR PROJECT INTERNET OF THINGS



GIRINATH R

CLASS: IOT05B10

IoT/ROBOTICS
MAY - BATCH2

Problem Statement:

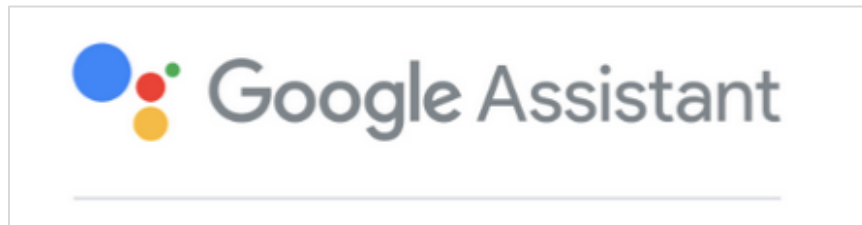
IoT controlled smart home

Setup a Google assistant applet from IFTTT to trigger a web request which will update a field in ThingSpeak cloud. Then read the data from ThingSpeak cloud using arduino to control the appliances in your home from anywhere in the world.

Ex: From your mobile phone, if you say- “ok google” “Turn on the light” from anywhere, it should turn on the light in your home.

Required Components:

- 1) Arduino Uno with cable
- 2) Esp8266 (Wi-Fi module)
- 3) 5v Relay
- 4) 230v bulb along with holder
- 5) Wires
- 6) Jumper cables



INTEGRATING THINGSPEAK CHANNEL WITH GOOGLE ASSISTANT USING IFTTT:

A channel **GOOGLE HOME AUTOMATION** was created in **THINGSPEAK** cloud platform. **Field1** was created to store the data sent from the device.

In **IFTTT** platform, '**this**' is set as '**Google Assistant**' and '**then**' is set as '**Webhooks**'. Two applets are created, one for turning on the light and other is for turning off. When the user says a simple phrase '**Turn on the light**', value **1** is stored in the ThingSpeak cloud and when the user says '**Turn off the light**', value **0** is stored in the ThingSpeak cloud, which is assured by response of the google assistant. The data is sent to the cloud using the URL of write channel feed.

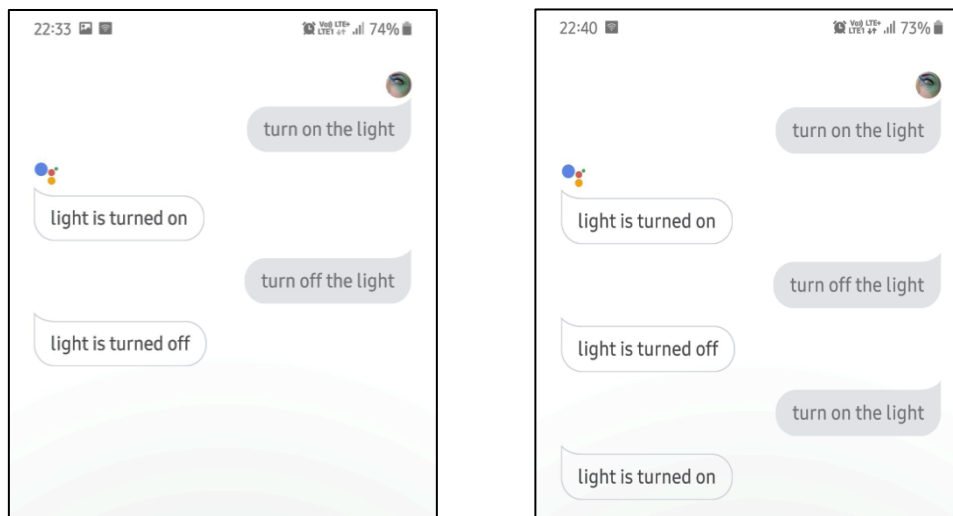
If the phrase is 'Turn off the light' the field1 value is set as '0'.

https://api.thingspeak.com/update?api_key=303RT48UQZ6F2VWE&field1=0

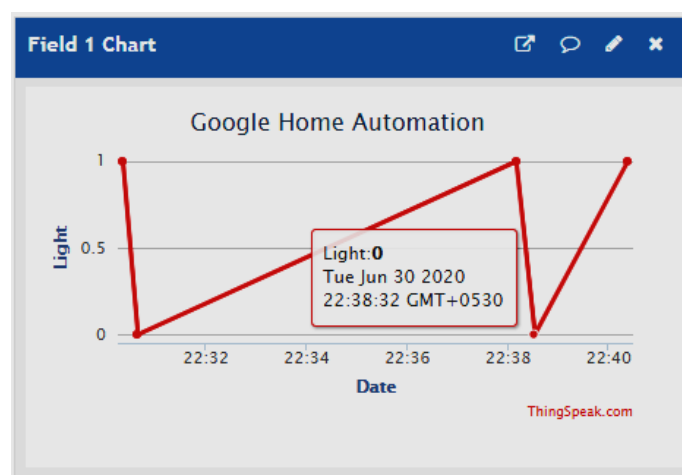
If the phrase is 'Turn on the light' the field1 value is set as '1'.

https://api.thingspeak.com/update?api_key=303RT48UQZ6F2VWE&field1=1

Google assistant tested with phrases at different time.




Field1 of Google Home Automation Channel.



APPLETS:

**If You say "turn off
the light", then
Make a web
request**


by girinath01662

 Connected



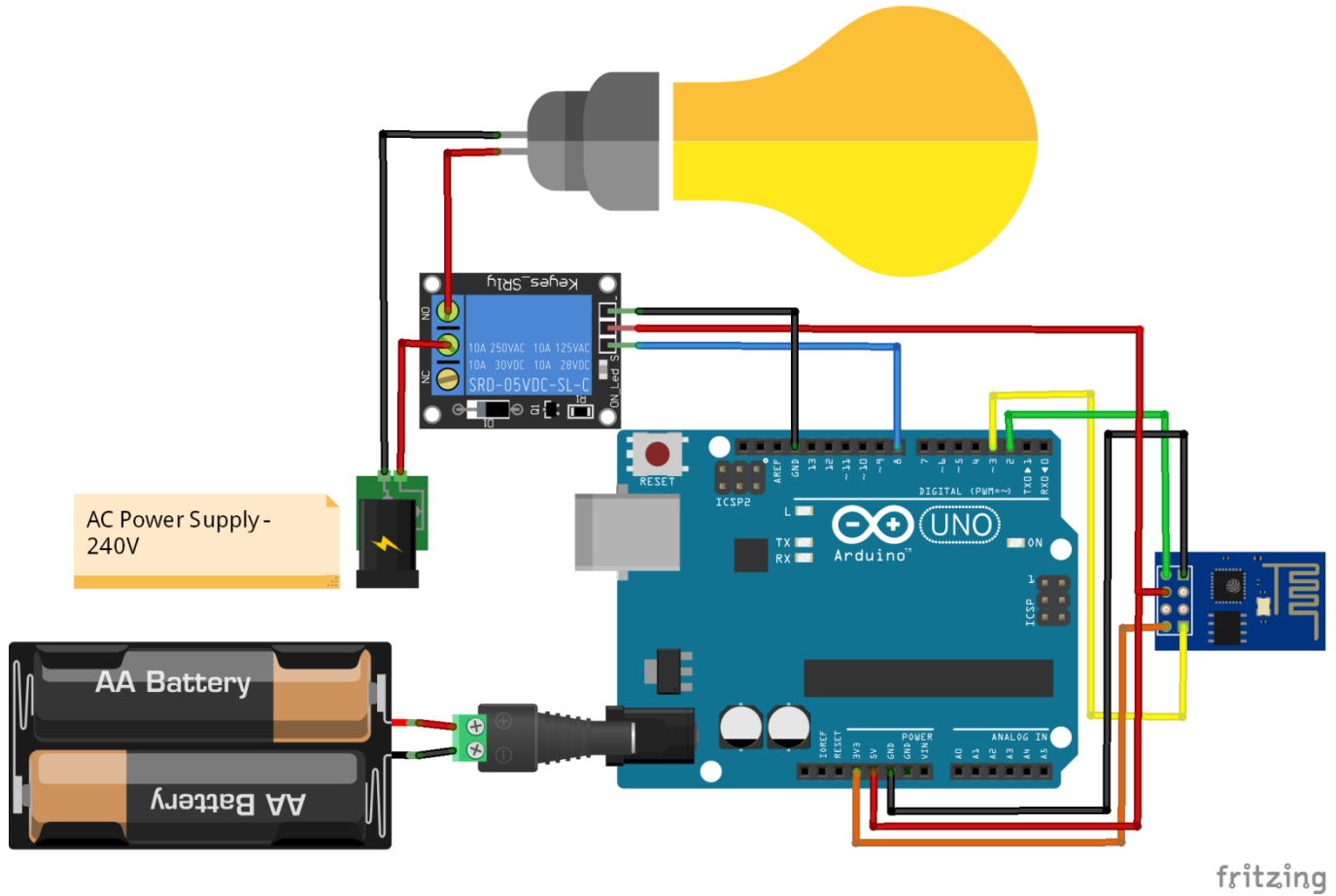
**If You say "turn on
the light", then
Make a web
request**

by girinath01662

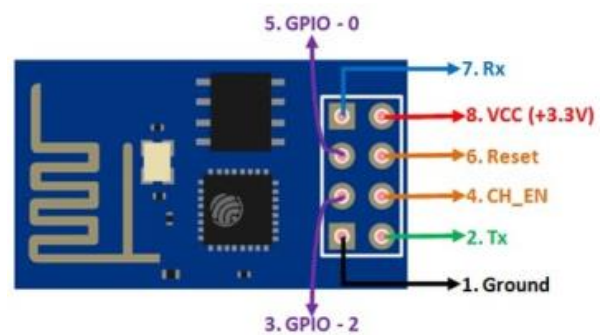
 Connected



CIRCUIT DIAGRAM:



Esp8266	Arduino Uno
GND	GND
VCC	3v3
CH_EN	5V
Rx	3
Tx	2



CODE:

```
#include <SoftwareSerial.h>

SoftwareSerial espSerial(2,3); // Rx, Tx

String wifiname = "JioFi2_0A6EBE"; // userID of the network to
connect

String pass = "abcd123"; // passcode of the wifi

String recvdata = "GET
/channels/1091619/feeds.json?api_key=CMSV35528WSBBGOE&results=2"
; // URL of read channel feed

String output ;

#define relay 8

void setup()
{
    pinMode(relay, OUTPUT);
    digitalWrite(relay, LOW);
    Serial.begin(9600);
    espSerial.begin(9600);
    espSerial.println("AT+RST");
    espSerial.println("AT+CWMODE = 1");
    espSerial.println("AT+CWJAP = \"" + wifiname + "\", \"" + pass
+ "\"");
    while(!espSerial.find("OK"))
    {
        //wait for connection
    }
    Serial.println("Connected...");
```

```
    delay(500);
}

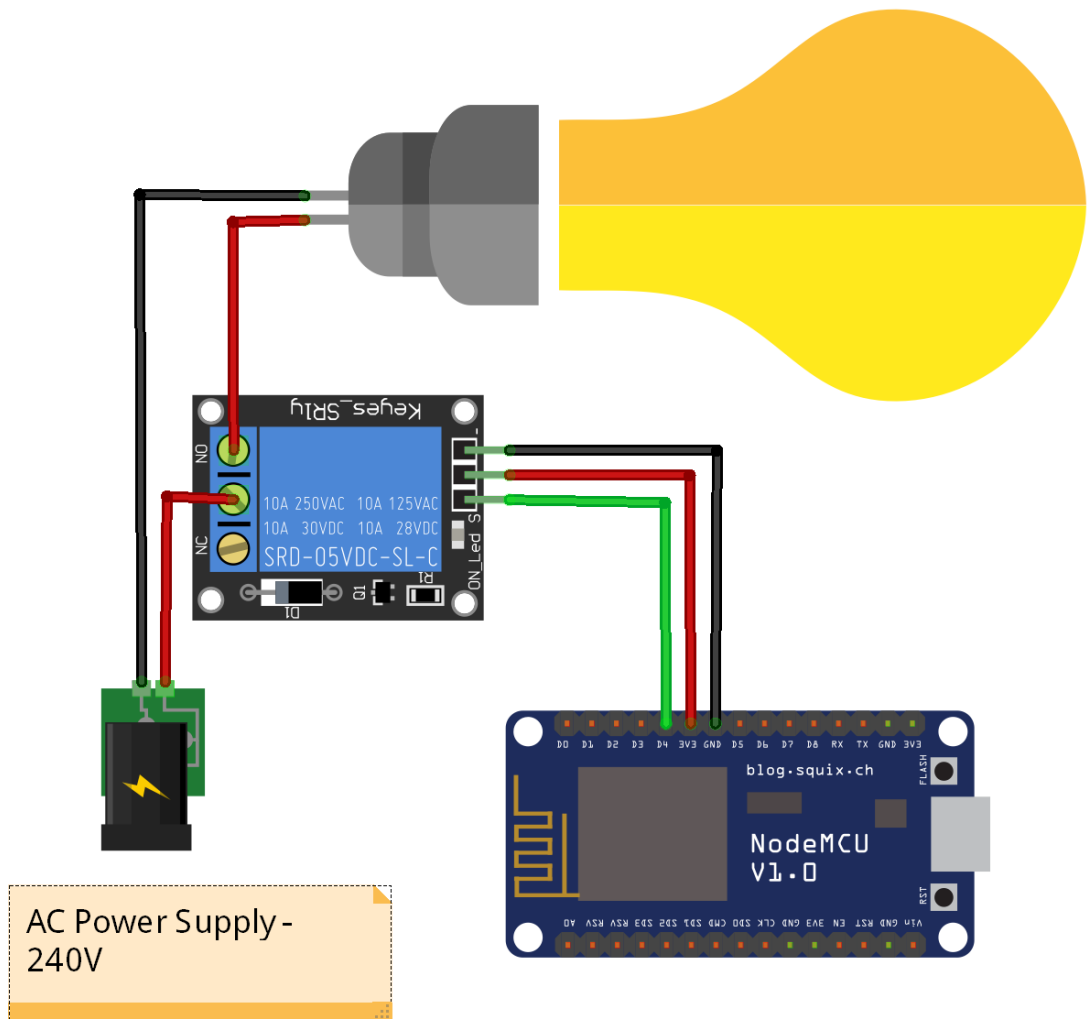
void loop()
{
    espSerial.println("AT+CIPMUX = 0");
    espSerial.println("AT+CIPSTART = \"TCP\",
\"api.thingspeak.com\", 80");
    espSerial.println("AT+CIPSEND = " +
String(recvdata.length()));
    espSerial.find(">");
    espSerial.println(recvdata);
    output=""; //clear the string
    while (espSerial.available()>0)
    {
        output = espSerial.readStringUntil('\n');
    }
    Serial.print("Received: ");
    Serial.println(output);
    char light = output[output.length()-5];
    Serial.println(light);
    Serial.print("The value is...");
    Serial.println(light);
    if (light == '0')
    {
        digitalWrite(relay, LOW);
    }
}
```

```
        Serial.println("Light is turned off!");
    }
    if (light == '1')
    {
        digitalWrite(relay, HIGH);
        Serial.println("Light is turned on!");
    }
    espSerial.println("AT+CIPCLOSE = 0");
    delay(1000);
}
```

DESCRIPTION:

The ESP8266 Wi-Fi module used gives microcontroller the access to the Wi-Fi network. The library included is SoftwareSerial. An object espSerial is created with digital pins 2 and 3 configured as Rx and Tx of the module. The name and passcode of the Wi-Fi network used is initialized. The URL of read channel feed of the GOOGLE HOME AUTOMATION channel is declared as recvdata. First the microcontroller is connected to the specified network. Using the URL the whole channel feed is read as string. In this string the fifth character from the end of the string stores the value of last entered data. Using this value we control the bulb with a relay circuit. The digital pin 4 is connected to the input pin of the 5V relay. The Vcc pin of the relay is connected to the 5V pin and the GND is connected to the GND of Arduino uno. The live wire of the bulb is connected between COM and NO pins of the relay. If the value is '1' the digital pin 4 is set HIGH. This turns on the bulb and if the value is '0' the digital pin 4 is set LOW, the bulb is turned off.

CIRCUIT DIAGRAM: (If NodeMCU is used)



fritzing

CODE:

```
#include <ESP8266WiFi.h>

#include <Thingspeak.h>

#include <WifiClient.h>

#define LIGHT 4


const char* user_id = "JioFi2_0A6EBE";//userID of the network to
connect

const char* password = "abcd123";//the passcode of the wifi

unsigned long channel = 1091619;// the channel ID

const char* APIKey = "CMSV35528WSBBGOE";// read APIKey of the
Channel used

unsigned int field1 = 1;//the field number in which the value is
stored

WifiClient client;


void setup()

{

    Serial.begin(115200);

    pinMode( LIGHT,OUTPUT);

    digitalWrite(LIGHT, LOW);


    Wifi.begin(user_id , password);

    Serial.println("");

    Serial.print("Connecting");

    while (Wifi.status() != WL_CONNECTED)
```

```
{  
    Serial.print(".");  
    delay(5000);  
}  
Serial.println("");  
Serial.print("Connected to: ");  
Serial.println(user_id);  
Serial.print("IP address: ");  
Serial.println(WiFi.localIP());  
ThingSpeak.begin(client);  
}  
  
void loop()  
{  
    int status = 0;  
    int bulb_state = 0;  
    bulb_state = ThingSpeak.readIntField(channel, field1, APIKey);  
    status = ThingSpeak.getLastReadStatus();  
    Serial.println(bulb_state);  
    delay(1000);  
    if (status == 200)          //200 represents successful request.  
    {  
        if(bulb_state == 1)  
        {  
            digitalWrite(LIGHT, HIGH);  
        }  
    }  
}
```

```

        Serial.println("Light is on.");
    }
    else if (bulb_state ==0)
    {
        digitalWrite(LIGHT, LOW);
        Serial.println("Light is off.");
    }
}
else
{
    Serial.println("Error");
}

delay(1000);
}

```

DESCRIPTION:

NodeMCU Esp8266, a microcontroller with Wi-Fi capability is used for the project instead of Arduino Uno. This calls for installing esp8266 board from boards manager and the board is set to “NodeMCU 1.0(ESP – 12E Module)”. Libraries to be included are ThingSpeak, WifiClient, and ESP8266WiFi. The baud rate of serial monitor and Wi-Fi are both set to 115200. Initially, we assign the user ID and password to which the board is to be connected. Along with it the channel number, field number and API key of ThingSpeak channel is also initialized. The connection of the micro-controller to the defined network is done first. Then with the help of ThingSpeak library, using predefined commands we retrieve the last entered value from the ThingSpeak cloud. Using this value, we control the bulb with a relay circuit, which functions as switch in high voltage circuit. The live wire travels through COM and NO pins of relay module and from there to the light. The input pin of the relay module is connected to the D4 GPIO pin of NodeMCU. If the value read is ‘1’, the D4 pin is set HIGH. The circuit closes and the current flows through the circuit, which makes the bulb glow. If the value read is ‘0’, D4 pin is set LOW. The bulb turns off.