

# BLIND'S COMPANION

-Kumaresh PV

## Introduction:

In this advanced era, World is running in its maximum speed, we must change our pace to move with flow. To move with the flow, we must be working 24 hours, right? Nope not at all. “*Smart working trounces hard working*”. Thus, make your moves smarter and smarter such that you control the world's speed.

If you think, this is in the case of capable person, what about the gifted ones? Why not! Even they can conquer the world, with the help of cutting edges of technology.

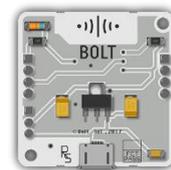
Thus, I here present you, the “*BLIND's COMPANION*”. This is a Bolt-IoT project connected with Arduino and Ultra-sonic sensor.

I show my gratitude towards Bolt-IoT for providing such knowledge and platform to share my project.

## Materials Required:

### Hardware Components:

1. Bolt Wi-Fi Module with ESP8266-12S



2. Arduino UNO



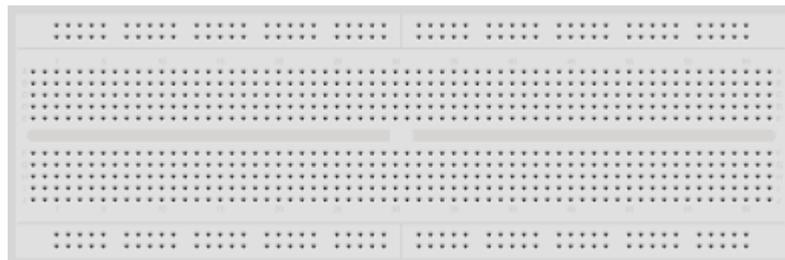
### 3. Ultrasonic Sensor - HC-SR04 (Generic)



### 4. LED (generic)



### 5. Breadboard (generic)



### 6. Jumper wires (generic)



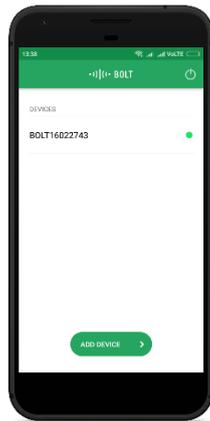
7. Resistor 220 ohm



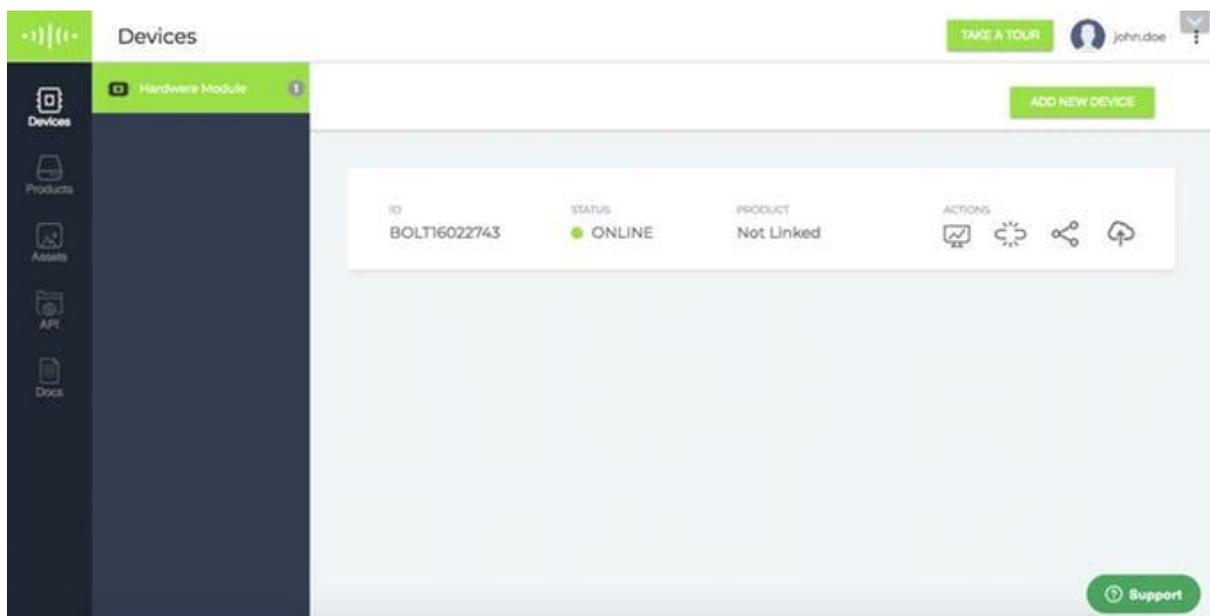
## Software Components:

1. Arduino IDE
2. Bolt IoT Bolt Cloud
3. Twilio SMS Messaging API
4. MailGun
5. Integromat
6. Twitter
7. Telegram





#### Step 4: Accessing the Bolt Cloud to Build IoT Projects



Step 5: Create a new product

Step 6: Linking device to a product

Step 7: Get the Bolt Cloud API Key

## Setting up the Arduino IDE

Step 1: Downloading Arduino IDE

Step 2: Enter and Executing the code

## Setting up the Linux Terminal

Step 1: Use Digital Ocean or Virtual Box or VM Ware to set Linux server

Step 2: Downloading Virtual Box and Ubuntu Server

The link to the software is given below:

- VirtualBox - <https://www.virtualbox.org/wiki/Downloads>
- Ubuntu Server ISO link - <http://releases.ubuntu.com/16.04/ubuntu-16.04.6-server-i386.iso>

Step 3: Installing the server through Virtual Box

Step 4: Open the terminal and install the following commands

```
sudo apt-get -y update
```

```
sudo apt install python3-pip
```

```
sudo pip3 install boltiot
```

## Setting up the Twilio SMS Messaging API

Step 1: Creating an account on Twilio

Step 2: Open <https://www.twilio.com/> in browser.

Step 3: Click on [Get a Free API Key](#) button to sign up.

Step 4: Fill all the necessary details in SIGN UP form. Below is the screenshot of filled sign up form.

Step 5: To verify they will ask for your phone number. Choose India as an option in the dropdown and then enter your phone number.

Step 6: Click on "Products"

Step 7: Now enable the SMS services by clicking on two checkboxes for Programmable SMS and Phone Numbers as shown below.

Step 8: Now, you will need to give a name for your project. I have given the name as My Project. Click on "Continue" once you have entered the project name.

Step 9: Click on "Skip this step" when it asks you to Invite a Teammate.

Step 10: Your project should be created at this point. Click on "Project Info" to view the account credentials which is required for your projects.

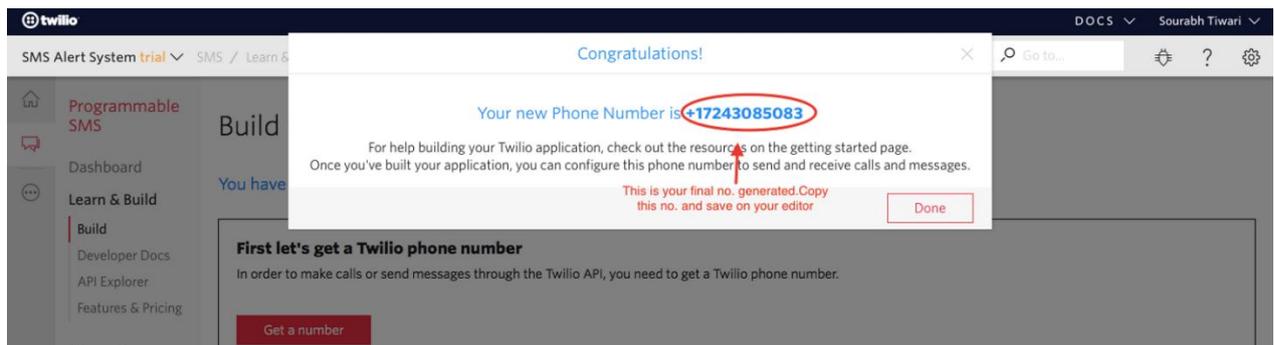
Step 11: You can view the Account SID and Auth token on this page. The Auth token is not visible by default, you can click on "view" button to make the Auth token visible as shown below. Copy both and save them somewhere securely.

Step 12: From the drop-down menu, choose "Programmable SMS". Now click on Get Started button to generate phone number.

Step 13: Click on Get a number button.

Step 14: Then a popup will appear. Click on Choose this number button.

Step 14: Then a popup will appear which will have the final number. Copy this number and save to notepad for future references.



That's it. You have successfully created the account on Twilio. In the next lesson, we will use Bolt Python library to create our own SMS Alert system.

## Setting up the MailGun Email Services

Step 1: Creating an account on MailGun

Step 2: Open <https://www.mailgun.com/> in browser.

Step 3: Click on [Sign Up](#) button.

Step 4: Fill all the necessary details in SIGN UP form. Make sure you are unchecking the payment option. Below is the screenshot of filled sign up form.

Step 5: You will get a verification mail having a link. Click on that link to verify your mail.

Step 6: To verify they will ask for phone number. Choose India as an option in the dropdown and then enter your phone number.

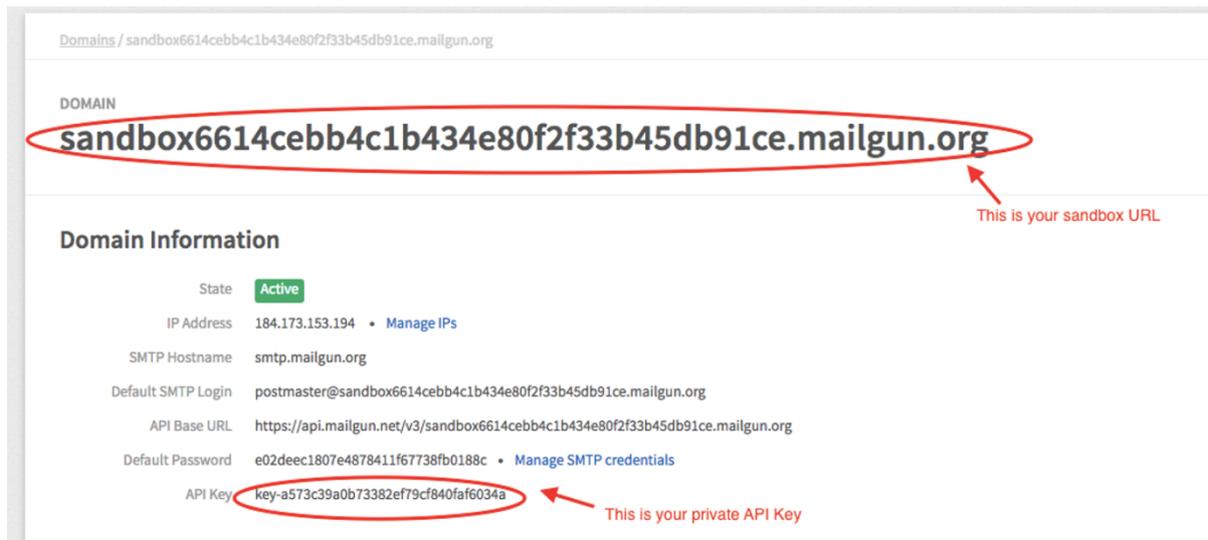
Step 7: After verification, scroll down to Sandbox Domain section. Click on Add Recipient button.

Step 8: Click on Invite New Recipient button.

Step 9: Enter the Recipient Email ID. In this case enter your Email ID.

Step 10: After adding Email ID a new sandbox will be generated. Click on the ID of the newly generated sandbox. Refer below image for the same.

Step 11: The new screen will have all the necessary credentials that you want for sending an email. Copy all this credentials and save in the notepad.



That's it. You have successfully created the account on MailGun. In the next lesson, we will use Bolt Python library to create our own Email Alert system.

## Setting up the Telegram Messaging Services

Step 1: Download and Install the latest version of Telegram app.

Step 2: Signup for a new account or sign into Telegram by providing your mobile number.

Step 3: Telegram will call you OR send you a SMS to verify your mobile number.

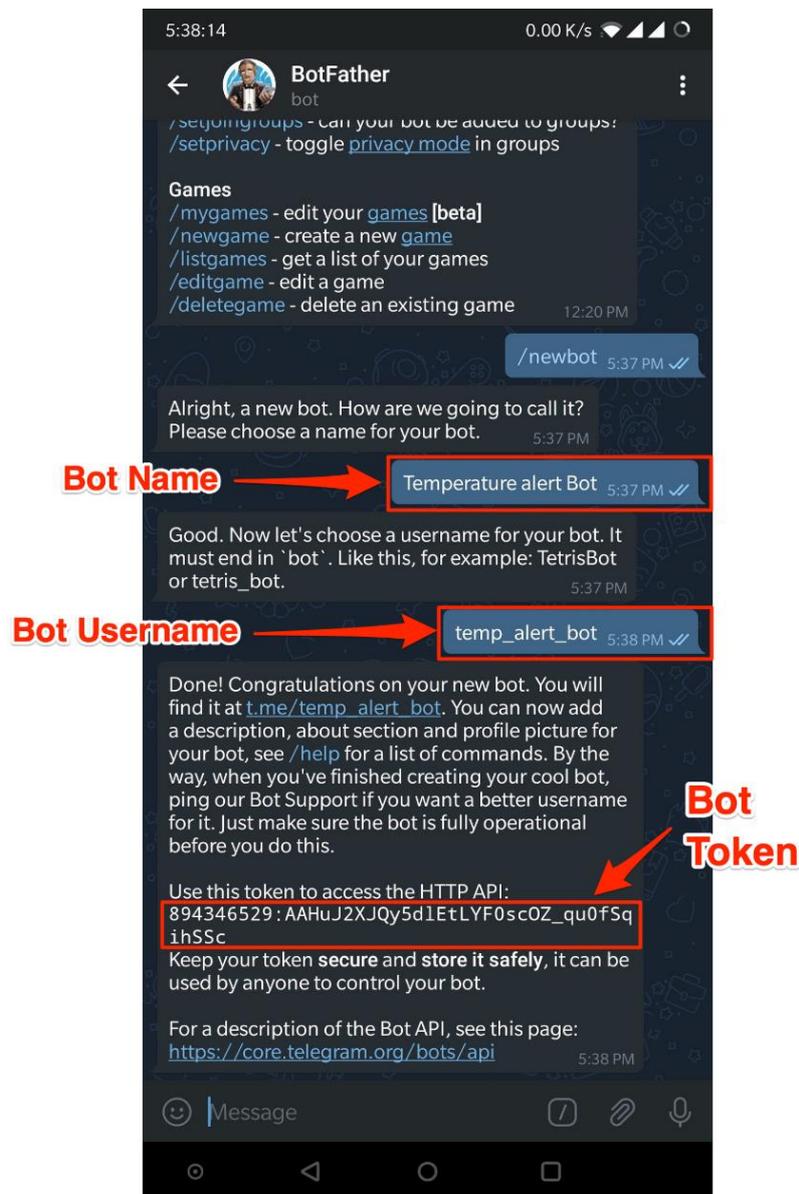
Step 4: Get your account verified and approve any permissions if required.

Step 5: You will be logged in to Telegram.

Step 6: Create a new Channel and unique Channel link

Step 7: Create a new Bot with the help of a Bot called as "BotFather"

Step 8: Create a new Bot, go ahead and type in "/newbot" in the window.



Step 9: When your bot is created successfully, you will be shown which contains the Bot Token. This token is used to control your Bot as well as send messages on behalf of it. **Please keep this token secure as it will allow anyone to access your Bot.**

Step 10: Add the bot to channel

Step 11: Click on the Administrators button so that we can add the newly created bot to the channel.

Step 12: You will be asked to confirm the rights for the bot. Press on the tick mark on the top right to continue adding the bot to the channel. Make sure that the bot has the rights to "Post Messages".

Now we will code for sending messages to the channel via the Bot.

## Setting up the Integromat Scenario

Step 1: Create your account on Integromat by clicking on this link: <https://www.integromat.com/en/?promo=boltiot>.

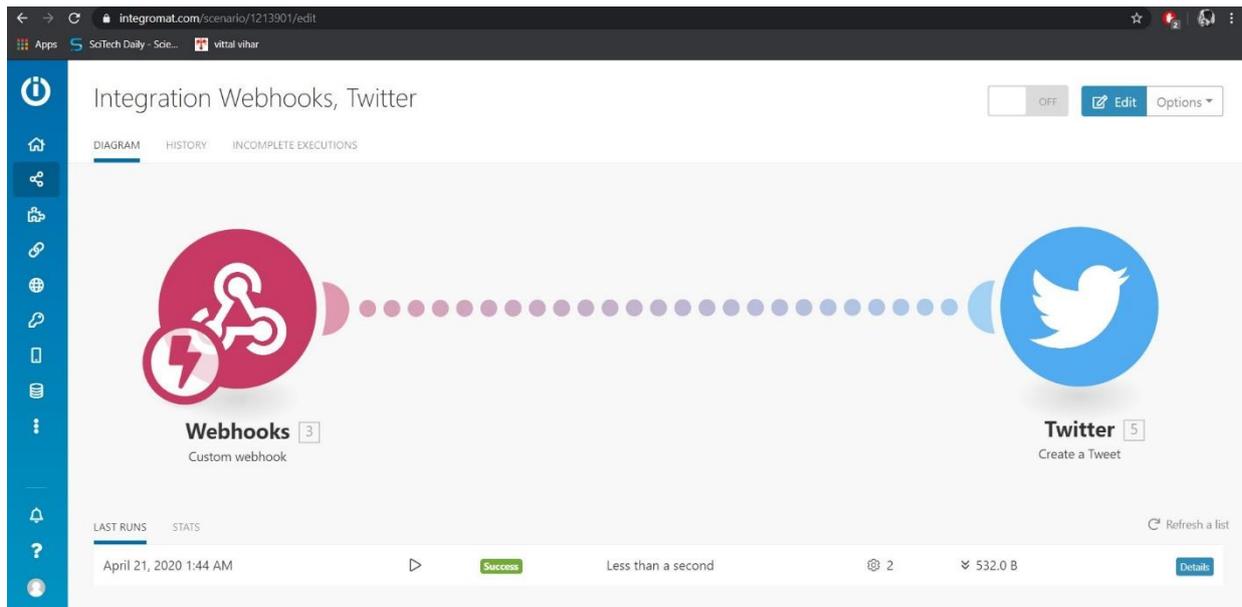
Step 2: Create a scenario on Integromat

Step 3: Create new scenario from your Integromat dashboard. It will ask you choose which services you want. Select Twitter and Webhooks.

Step 4: Configure your Webhooks module. Select the trigger as Custom Webhook. This will provide you with a unique URL which you will use in your Python program in the next task. Copy this URL and save it for later.

Step 5: Link the Twitter module to the webhooks module. Configure the Twitter module trigger as create tweet.

Step 6: The tweet posted on twitter should be "I am not getting enough light - sent by your Plant" The integromat scenario will look as shown in the image below.



## Working:

- To Know the working of the project more practically, let us see two scenarios regarding to our project.
- Scenario 1:
  - Mr. Martin, who is visually impaired, employs this project in his home. He has completed the setup by entering his neighbour's phone number.
  - Every day, he turns this on when he goes to sleep.
  - One day, the same routine happens and his in fine sleep. That night some intruder tries to rob his house.
  - And as the sensor detects people who cross the threshold line, it started sending message to the neighbour's phone number & Telegram, mails were sent to his known circle, and his tweets were recognised by Cops and Mr. Martin was saved from this incident.
- Scenario 2:

- Mr. Jacob, who is a software engineer, implements the same setup in his home. Now he gives his own number and finishes the setup.
  - When same incident takes place on his house, he is notified with all messages and mails.
  - He is also saved from this kind of burglary.
- 
- Thus, for both categories of people, this serves at the right time.
  - As the Ultra-sonic Sensor detects the change in the value, when someone enters.
  - It denotes Arduino as the threshold levels are crossed.
  - And the link between Arduino and Bolt IoT helps in sending messages.
  - Thus, Victims are saved from intruders. And safety is ensured.
  - The alerts are sent to Social media thus it gets easier for cops to save the victims.



www.shutterstock.com • 318842942



shutterstock.com • 777145786

## Coding:

### Arduino Code:

```
#include <Ultrasonic.h>

Ultrasonic ultrasonic(5, 6);

int LED = 2;

int threshold = 100;

void setup() {

  // put your setup code here, to run once:

  Serial.begin(9600);

  pinMode(LED, OUTPUT);

}

void loop() {

  // put your main code here, to run repeatedly:

  int distance = ultrasonic.distanceRead();

  if(distance < threshold)

  {

    digitalWrite(LED, HIGH);

    Serial.println("HIGH");

    delay(10000);

  }

  else{
```

```
digitalWrite(LED,LOW);

}

delay(1000);

}
```

## Python Code:

### *Configuration file (conf.py)*

```
"""Configurations"""

API_KEY = 'This is your Bolt Cloud account API key' This is your Bolt Cloud API Key
y

DEVICE_ID = 'This is the ID of your Bolt device'# This is the device ID

TELEGRAM_CHAT_ID = "@XXXX" #This is the channel ID of the created Telegram channel
.

TELEGRAM_BOT_ID = "botXXXXX" # This is the bot ID of the created Telegram Bot.

SID = 'You can find SID in your Twilio Dashboard'

AUTH_TOKEN = 'You can find on your Twilio Dashboard'

FROM_NUMBER = 'This is the no. generated by Twilio. You can find this on your Twilio Dashboard'

TO_NUMBER = 'This is your number. Make sure you are adding +91 in beginning'

MAILGUN_API_KEY = 'This is the private API key which you can find on your MailGun Dashboard'

SANDBOX_URL = 'You can find this on your MailGun Dashboard'

SENDER_EMAIL = 'This would be test@your SANDBOX_URL'

RECIPIENT_EMAIL = 'Enter your Email ID Here'
```

## *Sensor file (sensor.py)*

```
import requests, json, time, conf

from boltiot import Bolt, Sms, Email

mybolt = Bolt(conf.API_KEY, conf.DEVICE_ID) #Create object to fetch data

sms = Sms(conf.SID, conf.AUTH_TOKEN, conf.TO_NUMBER, conf.FROM_NUMBER)

mailer = Email(conf.MAILGUN_API_KEY, conf.SANDBOX_URL, conf.SENDER_EMAIL, conf.RECIPIENT_EMAIL)

response = mybolt.serialRead('10')

print (response)

def send_telegram_message(message):

    """Sends message via Telegram"""

    url = "https://api.telegram.org/" + conf.TELEGRAM_BOT_ID + "/sendMessage"

    data = {

        "chat_id": conf.TELEGRAM_CHAT_ID,

        "text": message

    }

def trigger_integromat_webhook():

    URL = "https://www.integromat.com/" # REPLACE WITH CORRECT URL

    response = requests.request("GET", URL)

    print(response.text)
```

```

try:

    response = requests.request("POST", url, params=data )

    print("This is the Telegram response")

    print(response.text)

    telegram_data = json.loads(response.text)

    return telegram_data["ok"]

except Exception as e:

    print("Error occured: Below are the details")

    print(e)

    return False

while True:

    response = mybolt.serialRead('10') #Fetching the value from Arduino

    data = json.loads(response)

    status_value = data['value'].rstrip()

    if str(status_value) == 'HIGH':

        trigger_integromat_webhook()

        print ("Status is", status_value)

        message = "Some intruder has entered the room"

        telegram_status = send_telegram_message(message)

        print("Making request to Twilio to send a SMS")

        response = sms.send_sms("Some intruder has entered the room " +str(sensor_v
alue))

```

```
print("Response received from Twilio is: " + str(response))

print("Status of SMS at Twilio is :" + str(response.status))

print("Making request to MailGun to send an email")

response = mailer.send_email("Alert", " Someone intruder has entered the ro
om" +str(sensor_value))

response_text = json.loads(response.text)

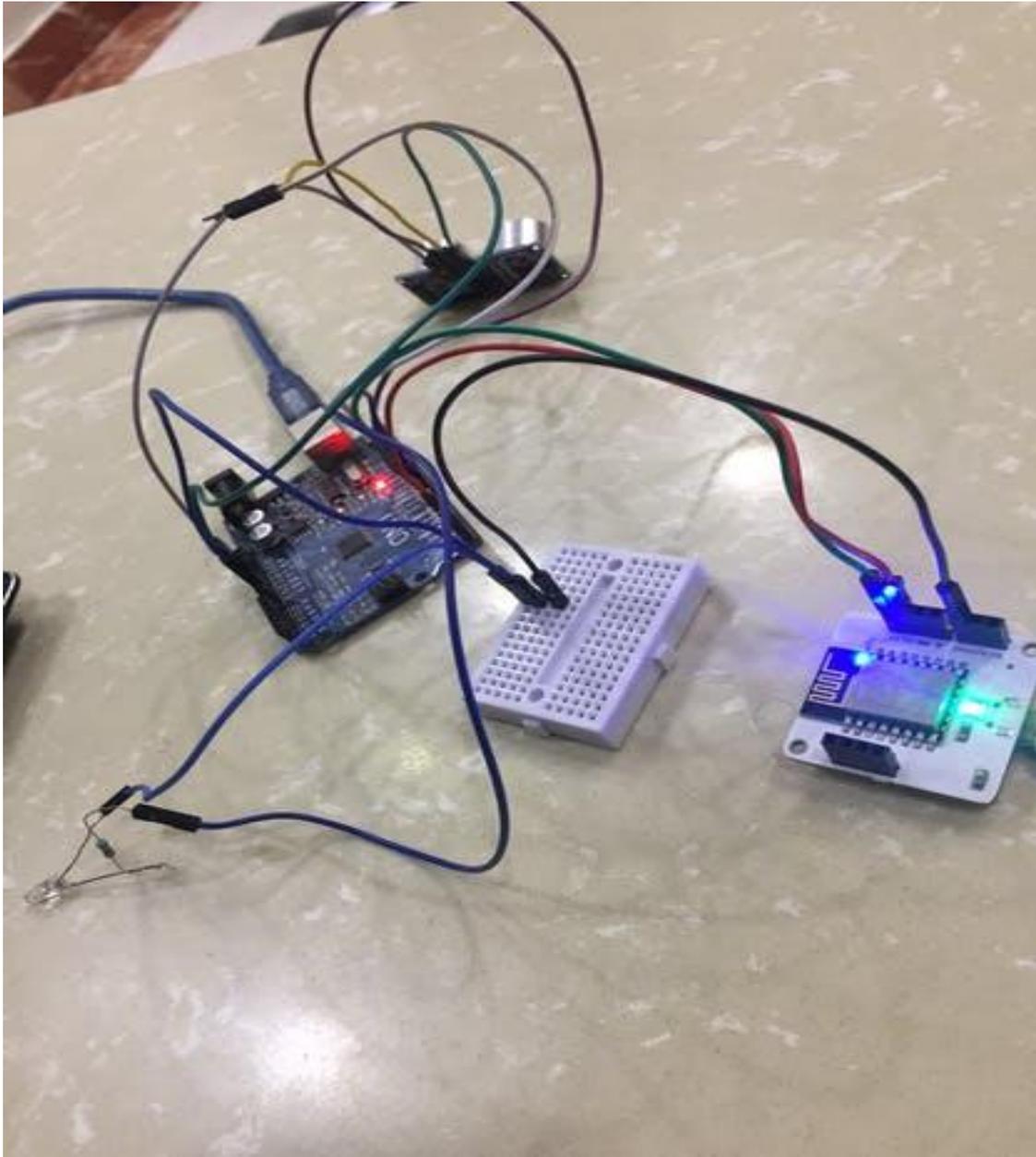
print("Response received from Mailgun is: " + str(response_text['message'])
)

else:

    print ("Status is LOW!",status_value)

time.sleep(10)
```

**OUTPUT:**



**test** 23

15:11

**Alert**

Some intruder has entered the room





Sent from your Twilio trial account -  
Some intruder has entered the room

## Conclusion:

*“The only thing worse than being blind is having sight but no vision”*

*-Hellen Keller*

Perhaps Bolt-IoT has given a great platform to share our ideas and enhance our knowledge. The small contribution from myself is like ‘a drop in sea’. Thus, people’s safety and security are ensured. I further want to develop similar projects and extend my support toward **Internet of Things**. I thank everyone who has supported me to render this project.

***THANK-YOU***