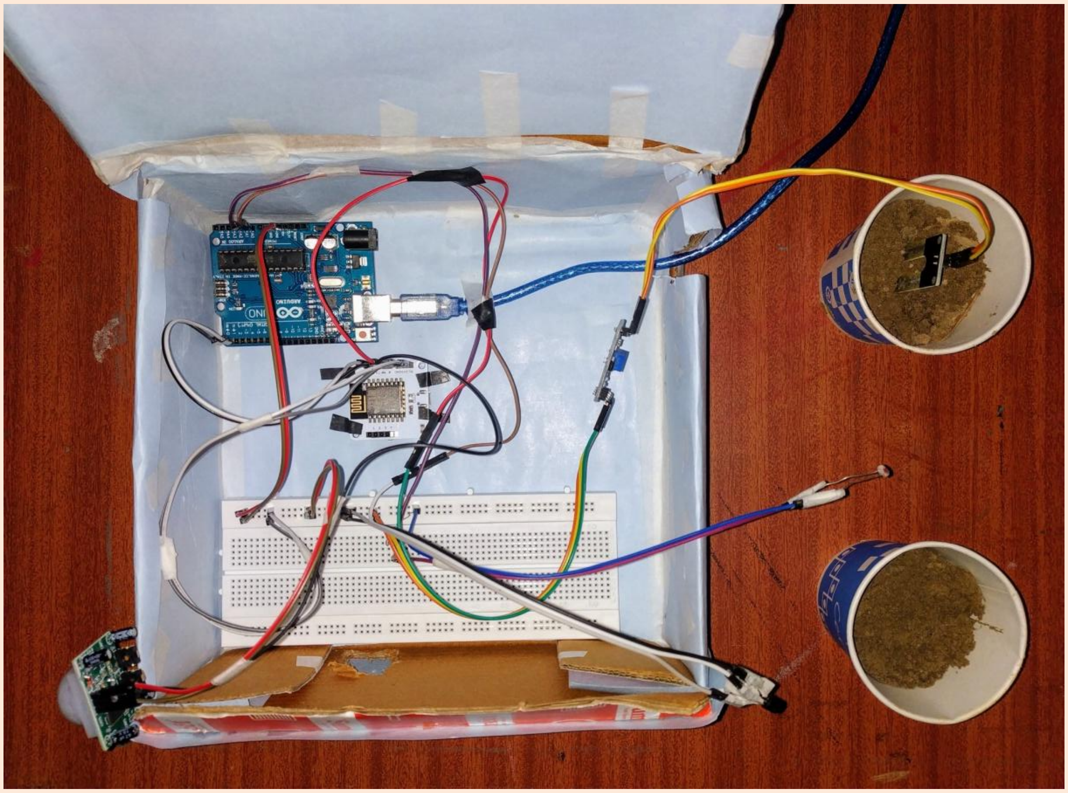
Farm Monitoring Sysytem

Overview:

A device that would help farmers to improve crop productivity by safeguarding the crops from intruder or fire by sending an alert to the farmer if someone tries to destroy the crops, also this will give real time feedbacks about the moisture in the soil along with temperature and amount of light which are essential for healthy crops.



Things Used In This Project:

Hardware Components

I. Bolt IOT Bolt Wi-Fi Module
II. USB-A to Mini-USB Cable
III. Resistor 10k Ohm
IV. Jumping Wires
V. Arduino Uno
VI. USB-A to B Cable for Arduino
VII. Moisture Sensor
VIII. Temperature Sensor
IX. Light Sensor
X. Motion Sensor

Software App and Online Components

1. Bolt IOT android app
2. Arduino IDE
3. Python 3
4. Integromat
5. Bolt Cloud
6. Twilio
7. Mailgun
8. Webhook

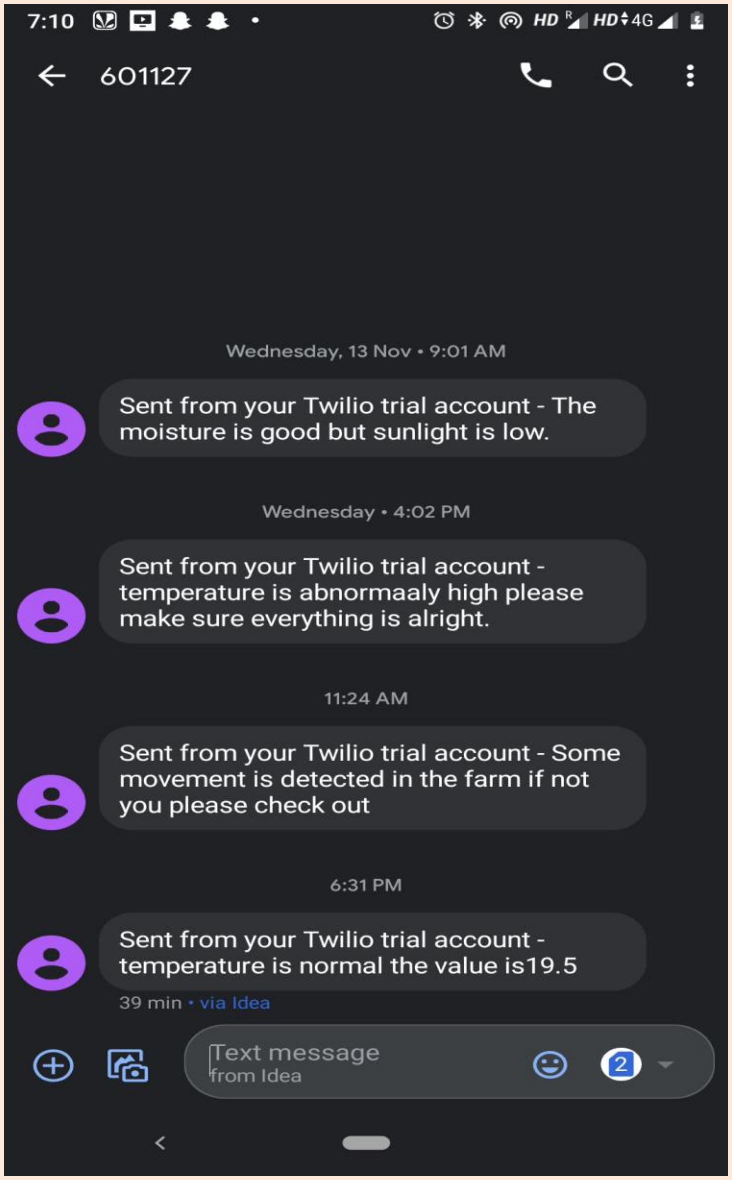
Story:

Introduction

Project has two functionality i.e. to give real time information to a farmer about moisture in soil, the amount of light and temperature also this project can alert farmer against a fire or a intruder trying to destroy crops. It is not easy for a farmer to manage farm all by himself also the factors like temperature

and moisture cannot be measured easily and many others factors like animals, fire can destroy the crops this project will solve all these problems, we can detect sudden change in temperature and light(z-score analysis) and notify the farmer using message or e-mail, also in case of fire or an unwanted intruder we can send an alert to farmer if anomaly detected in temperature or an intruder tries to destroy crop.

Demonstration



Wednesday, 13 Nov • 9:01 AM

Sent from your Twilio trial account - The moisture is good but sunlight is low.

Wednesday • 4:02 PM

Sent from your Twilio trial account - temperature is abnormaaly high please make sure everything is alright.

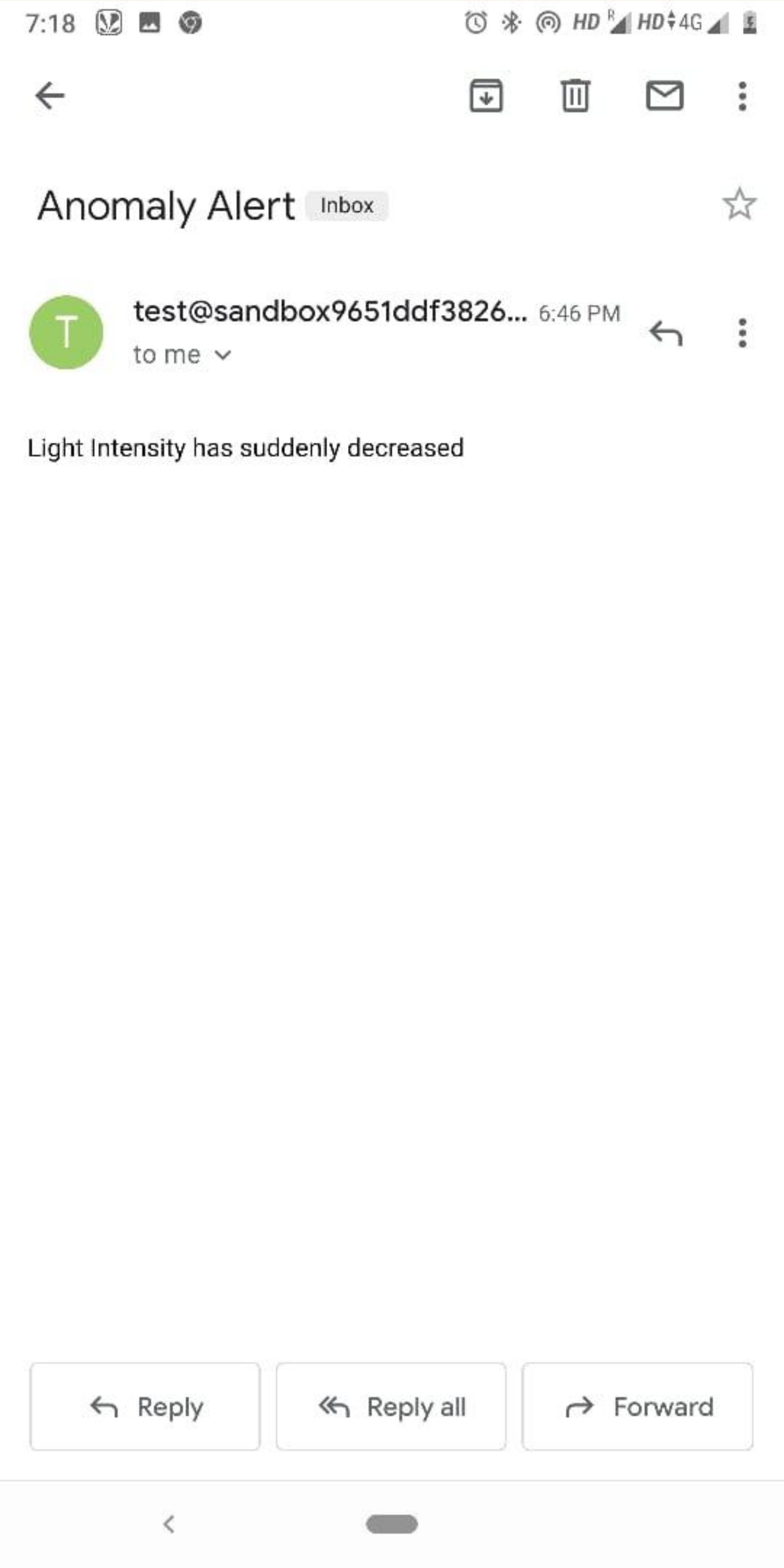
11:24 AM

Sent from your Twilio trial account - Some movement is detected in the farm if not you please check out

6:31 PM

Sent from your Twilio trial account - temperature is normal the value is19.5

39 min • via Idea



Anomaly Alert

test@sandbox9651ddf3826... 6:46 PM
to me

Light Intensity has suddenly decreased

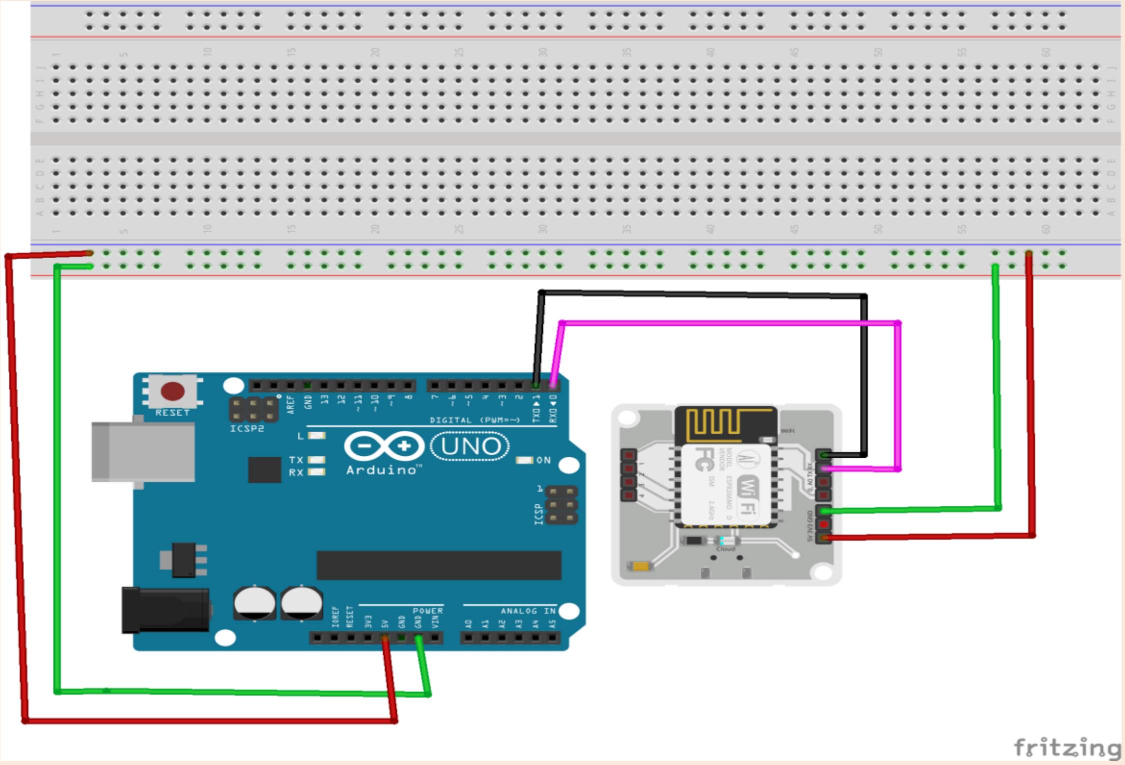
Steps For Building The Project

- **STEP 1: Connect your Arduino to your computer and upload the plantproject.ino program to the Arduino using Arduino IDE(this is done because the cables for communication between bolt WI-FI module and Arduino should not be connected before uploading this code ,cable used for communication can be identified easily it is labeled as Rx and Tx in both Arduino and BOLT WI-FI module)**
- **STEP 2: #Code for Sensors on Arduino IDE(plantproject.ino): There is a simple code in the folder where are calling Serial to run on 9600 baud rate. And then taking the input using analogRead() for pin A0 of Arduino.**
- **STEP 3: now unplug the Arduino and make the connection.**

CONNECTIONS:

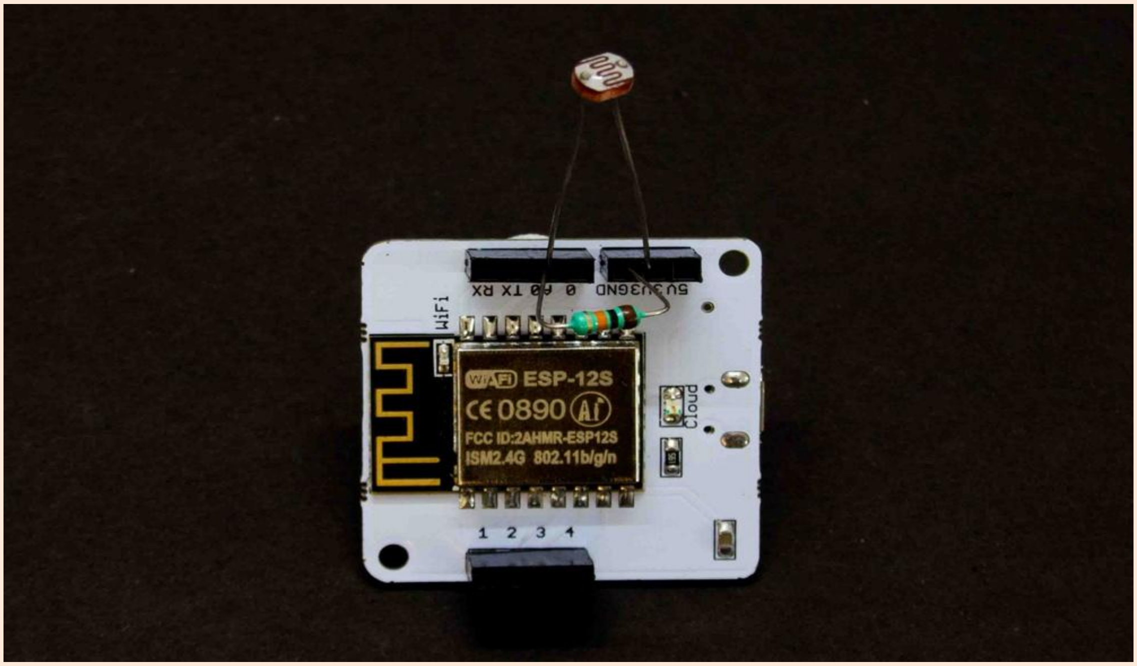
#NOTE: we can connect 5V and ground pin of Arduino to 5V and ground of BOLT WI-FI module to power BOLT WI-FI module using Arduino in this way we need only one power supply to power both Arduino and BOLT WI-FI module now we don't have to give supply to both, only one supply given to Arduino is enough through which our WI-FI module will be powered.

for example:

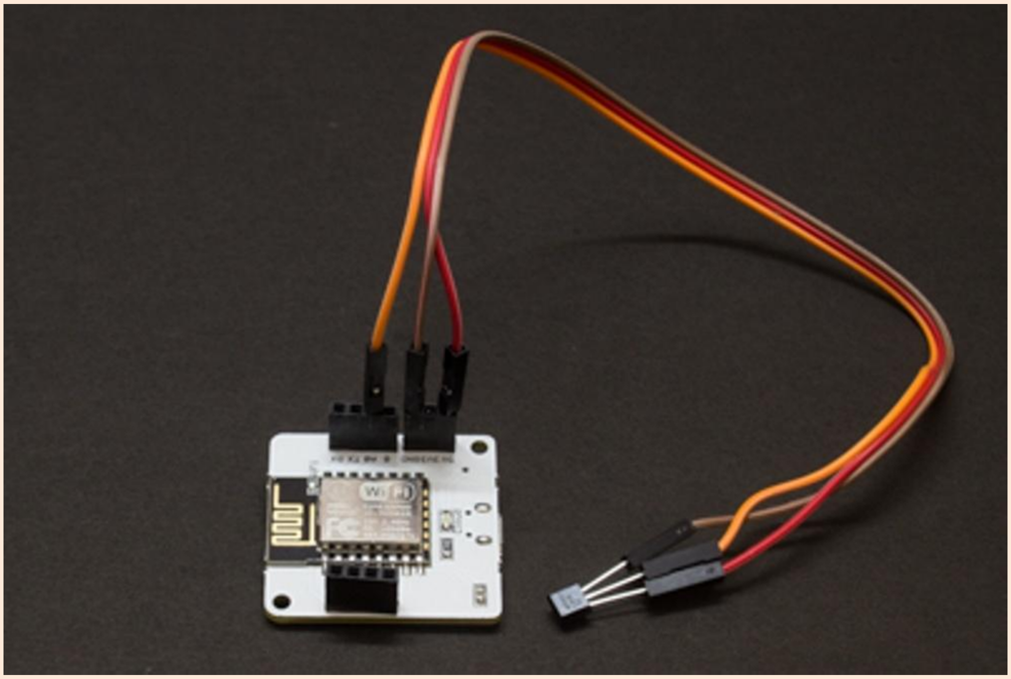


In the above diagram give the power supply only to Arduino through which our WI-FI module will be powered.

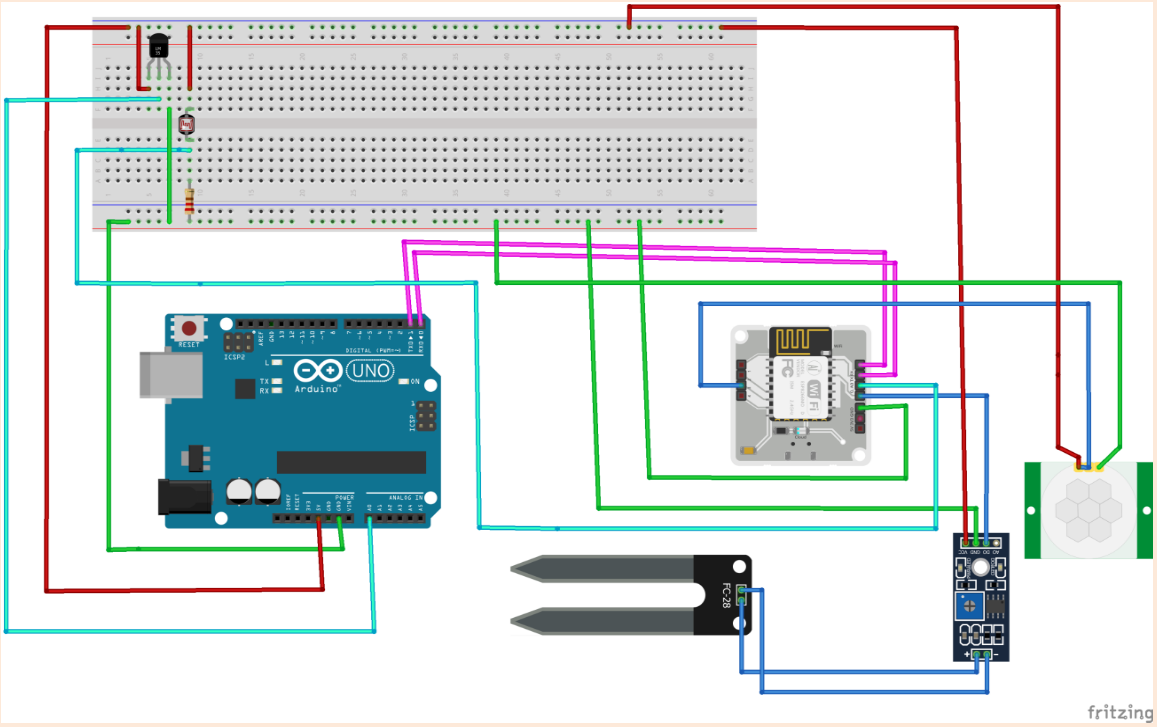
1. connect v output of PIR motion sensor to digital pin 4 of BOLT WI-FI module and Vcc pin to 5V and ground pin to ground
2. connect digital pin of moisture sensor to digital 0 pin of BOLT WI-FI module ,ground pin to ground and voltage pin to 5v supply refer diagram below.
3. connect light sensor and temperature sensor to A0 and A0 pin of BOLT WI-FI module and Arduino remember that you need resistor for connection of light sensor refer the below diagram for better understanding.



source: https://trainings.boltiot.com/

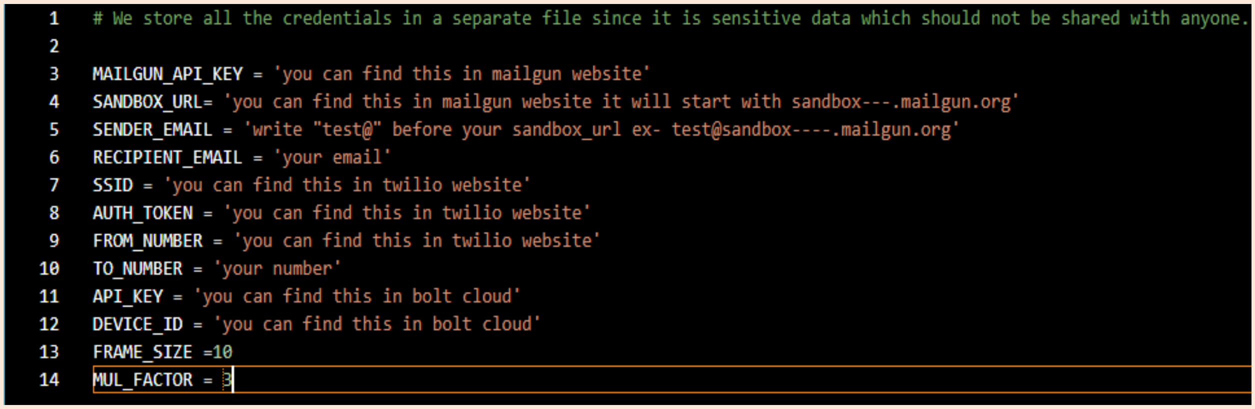


The resistance across LDR changes with change in light intensity falling on it. Since the Bolt Module cannot read resistance values, but it can read voltage values, hence a voltage divider circuit is made and the input to the Bolt module is the voltage across 10k resistance (since it is connected between an LDR terminal and ground) which depends on the resistance across LDR



- Note: If you want to use the graphs in your project you can go through these links: https://github.com/Inventrom/boltiot-arduino-helper
- STEP 4: #Code for Detection or Alert on Python IDE: Once you can run the Arduino code then we want a python IDE to write the code. You can write either in a virtual machine or your raspberry pi or in windows. I'm using it in windows Python 3.6 IDE. You can download and read documentation to install on your system https://www.python.org/
- About libraries we are using external boltiot and requests library only. you can install both of them using pip command on terminal or cmd. For windows you can learn how to do this https://www.liquidweb.com/kb/install-pip-windows/
- STEP 5: PYTHON CODE(conf.py) we need to make a configuration file which will have the specific keys for each user/device. We will import this file in our main code and use the various attributes. The advantage of this is that each user

will only have to change the contents the configuration file to use the product.



```
# We store all the credentials in a separate file since it is sensitive data which should not be shared with anyone.

MAILGUN_API_KEY = 'you can find this in mailgun website'
SANDBOX_URL= 'you can find this in mailgun website it will start with sandbox---.mailgun.org'
SENDER_EMAIL = 'write "test@" before your sandbox_url ex- test@sandbox----.mailgun.org'
RECIPIENT_EMAIL = 'your email'
SSID = 'you can find this in twilio website'
AUTH_TOKEN = 'you can find this in twilio website'
FROM_NUMBER = 'you can find this in twilio website'
TO_NUMBER = 'your number'
API_KEY = 'you can find this in bolt cloud'
DEVICE_ID = 'you can find this in bolt cloud'
FRAME_SIZE =10
MUL_FACTOR = 3
```

- **in the above configuration file we are mentioning the bolt API key and the device id also we mention the frame rate and multiplication factor for z- analysis.**
- **STEP 6: PYTHON CODE(temp.py)**

For the above code, the logic is we have to fetch data from bolt serial so we are using the functioning bolt.serialRead('1') but going through the documentation I found that value is 10 instead of 1. Actually while using the value 10 the fetching data is not instant. While using 1 it’s an instant data fetched but with a drawback i.e. in the first iteration it will fetch a bunch of combined data. After getting the data from serial we have to make it in the correct form in my case I’m getting data as {‘value’:’546\n45\n’, ’ success’:’1’} so I need to extract it. So I first converted it into JSON data then using list(data['value'].split("\n")) I get the list so at index 0 I have sensor value.

- **STEP 7: PYTHON CODE(light_anomaly.py)**

In this program we use z-score anomaly detection for sudden change in the data values. We used this concept in detecting any sudden change in the light value, and temperature value too so this method is also used in the above program.

Anomaly Detection Alert:

The assessment of the Z-score is used to detect anomalies. The anomaly here implies that the value of a variable (temperature or light value) exceeds a certain value called bounds. The value range is referred to as boundaries (upper and lower bound). We use the upper limit only to detect the fire warning. The input values, frame size, and multiplication factor are used to calculate these boundaries. The frame size is the minimum number of input values needed for the Z-score analysis and the multiplication factor determines the proximity of the bounds to the input values curve.

$$Mn = \frac{\sum_{i=1}^{r} Vi}{r}$$

$$Zn = C * \sqrt{\frac{\sum_{i=1}^{r} (Vi - Mn)^2}{r}}$$

$$Tn = Vi \pm Zn$$

r = Frame Size

C = Multiplication Factor

Given above is the formula to calculate the bounds. Here the input is represented as 'Vi', 'r' denotes the frame size and 'C' is the multiplication factor. Firstly we calculate the mean (Mn) of the input values (for every new input, the mean is calculated again). The variation of input value (from the mean) is given as : (Vi -

Mn)^2. The Z-score (Zn) is calculated as shown above (square root of the mean of the variation of each input value multiplied by the multiplication factor). The bounds are represented as 'Tn' and the upper bound is calculated as (Vi + Zn) and the lower bound is calculated as (Vi - Zn).The frame size and multiplication factor are determined using the trial-and-error method.

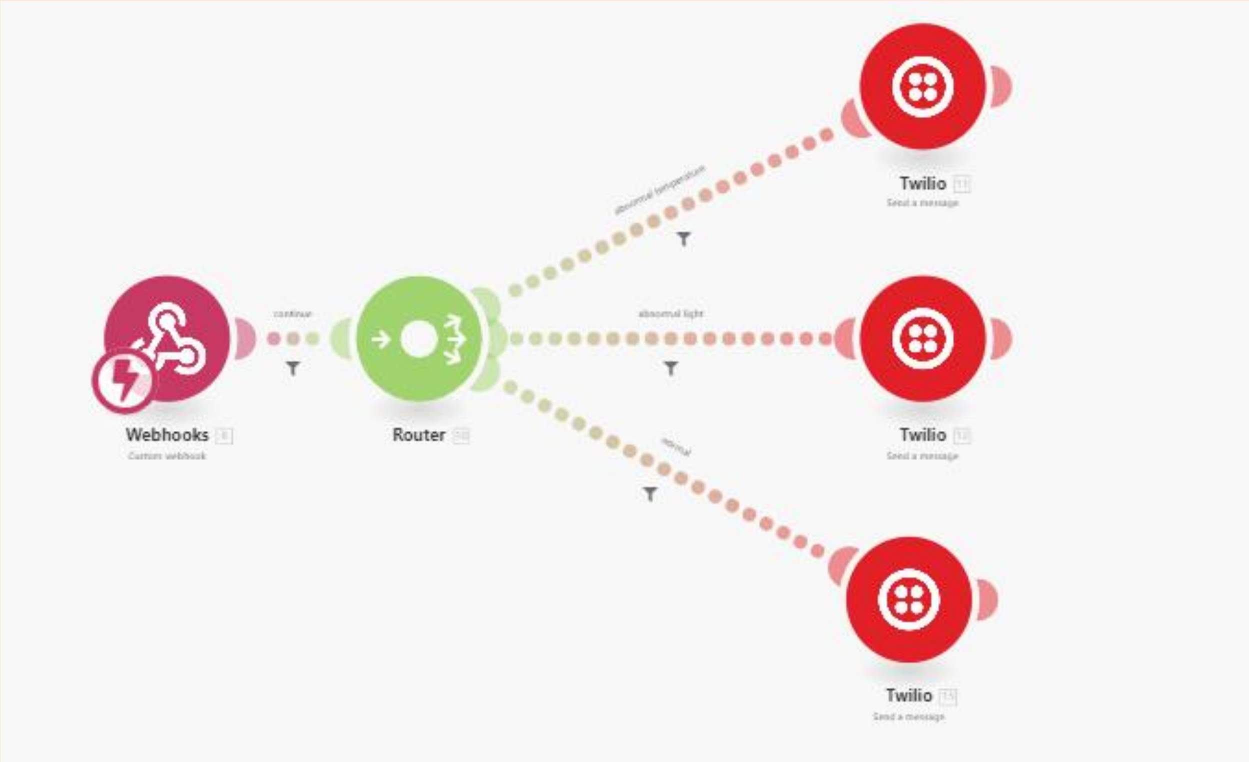
- **STEP 8: PYTHON CODE(intruder.py)**

This is a very simple code in which we read the sensor value using mybolt.digitalRead('pin no') if this value is equal to 1 we send an alert

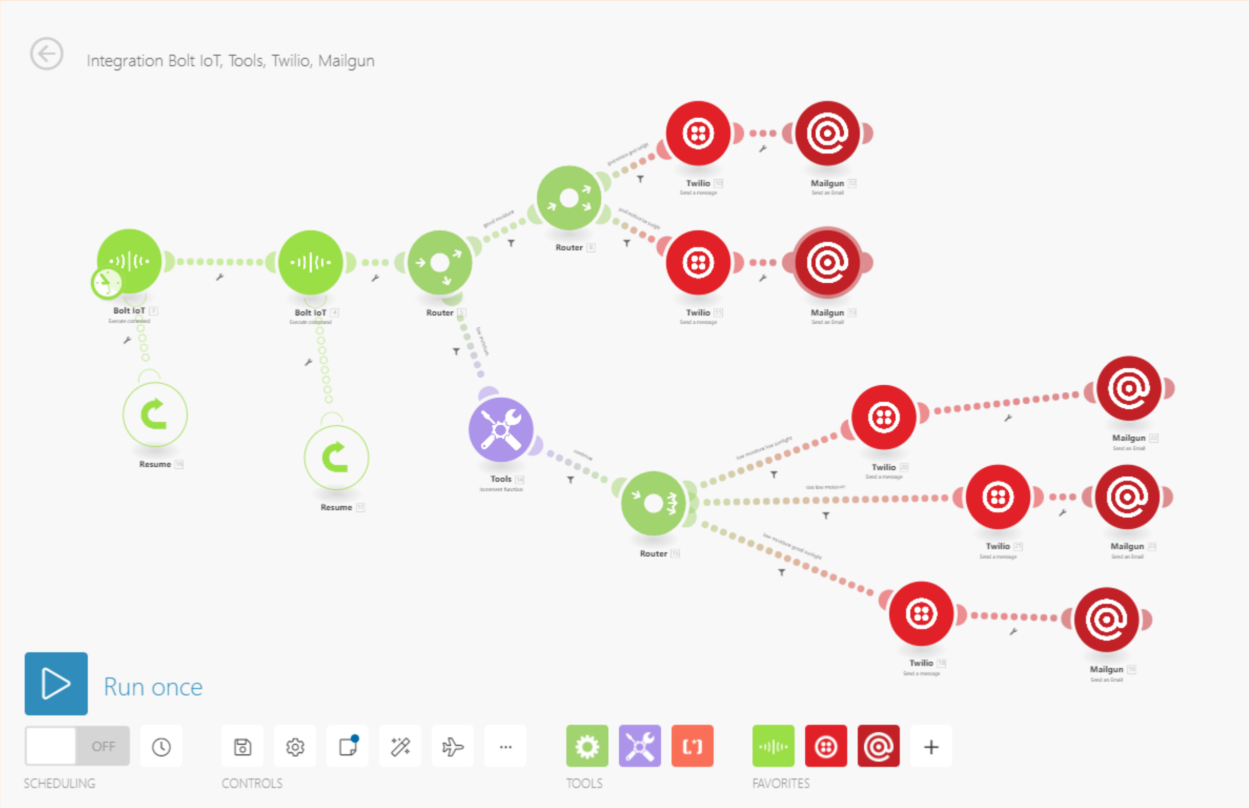
- **STEP 9: Integromat**

For temperature sensor since it is not connected to WI-FI module directly we will make a web request for Integromat scenario. Which will send the notification and text message to our device which generates the current situation according to the condition mentioned in the scenario. In Integromat add webhook and twilio or any other service you want to use. Right click on empty screen -> add module -> webhook -> custom webhook trigger when data recieved -> add -> give any name and save copy the link from there and paste it in your temp.py program

The conditions are if temperature>55 then it’s a abnormal temperature alert, else everything is normal. Make the scenario as shown below. we can easily change the schedule of Integromat and also in python code we can decide how often to take readings we can make the program sleep for certain interval of time and hence take reading for example after every 20 minutes or every hour.


Webhooks
Custom webhook
continue
Router
abnormal temperature
abnormal light
normal
Twilio
Send a message
Twilio
Send a message
Twilio
Send a message

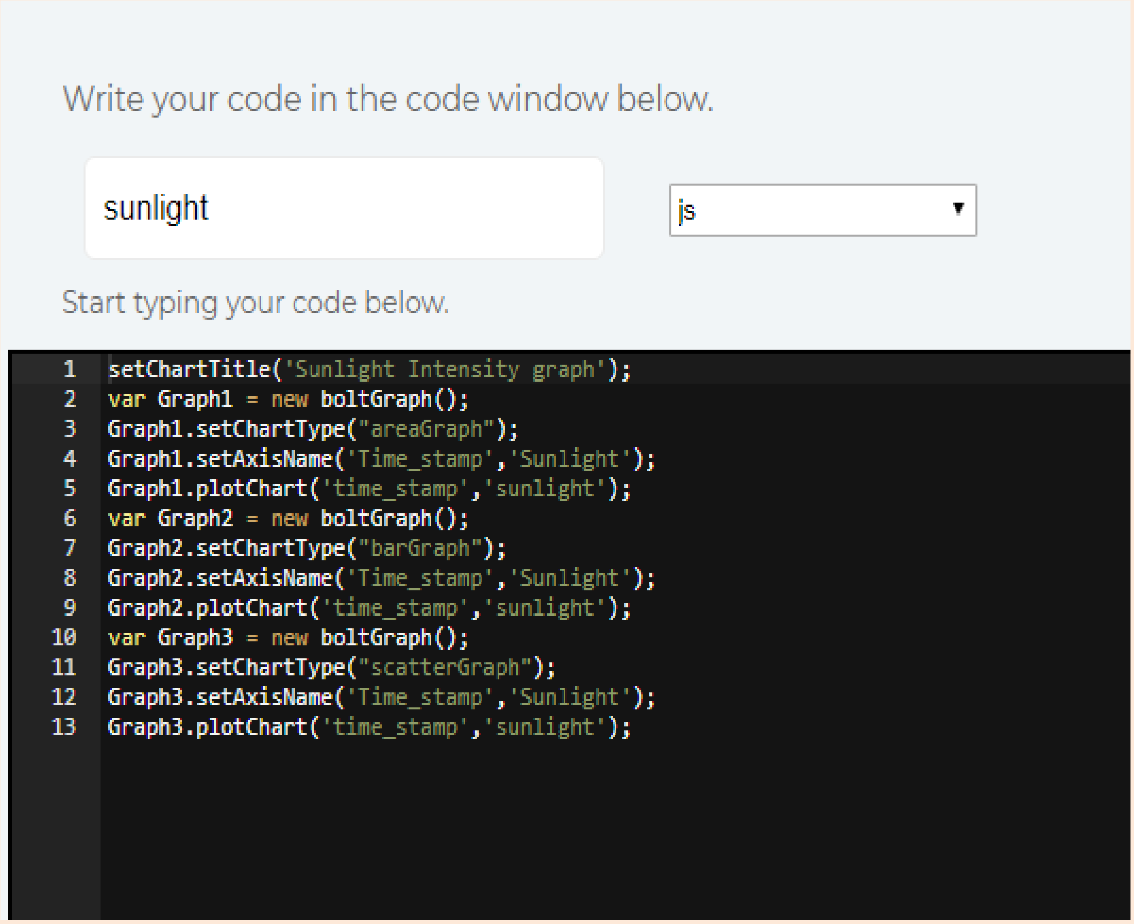
SECOND:


Integration Bolt IoT, Tools, Twilio, Mailgun
Bolt IoT
Execute command
Resume
Bolt IoT
Execute command
Resume
Router
good moisture
Router
Twilio
Send a message
Mailgun
Send an Email
Twilio
Send a message
Mailgun
Send an Email
low moisture
Tools
Increment function
continue
Router
Twilio
Send a message
Mailgun
Send an Email
Twilio
Send a message
Mailgun
Send an Email
Twilio
Send a message
Mailgun
Send an Email
Run once
OFF
SCHEDULING
CONTROLS
TOOLS
FAVORITES

add bolt along with mailgun and Twilio now build your logic as shown above. Two bolt module is added to fetch two values one for moisture other for light value. logic is like this if moisture value 1 then good moisture otherwise less moisture in the good moisture part if sunlight is above 850 then good amount of sun light else it is not now in the low moisture part we increase the value of i by 1 in each scenario run and this integromat is scheduled to run after every 1 hour if the value of 'i' becomes greater than 6 than it has been 6 hours without enough moisture for crops so we will send an alert to irrigate the farm else we will check sunlight values and send the alert acordingly.

Setting up a Product on Bolt Cloud

- Go to https://cloud.boltiot.com/ and then follow the steps below
- Create product https://docs.boltiot.com/docs/adding-a-productWriting code for controlling device https://docs.boltiot.com/docs/using-custom-files-in-your-product?
- Link device to product ?https://docs.boltiot.com/docs/link-device-to-a-product?
- STEP 10: Now for visualizing the amount of light the crops are getting we will set up product on bolt cloud. First click on add product in product section then give your product a name select input device and GPIO. Click on configure this product then select and give variable name 'Sunlight' to A0 pin then go to code section and write the code as below. I am plotting 3 different graphs.



Write your code in the code window below.

sunlight

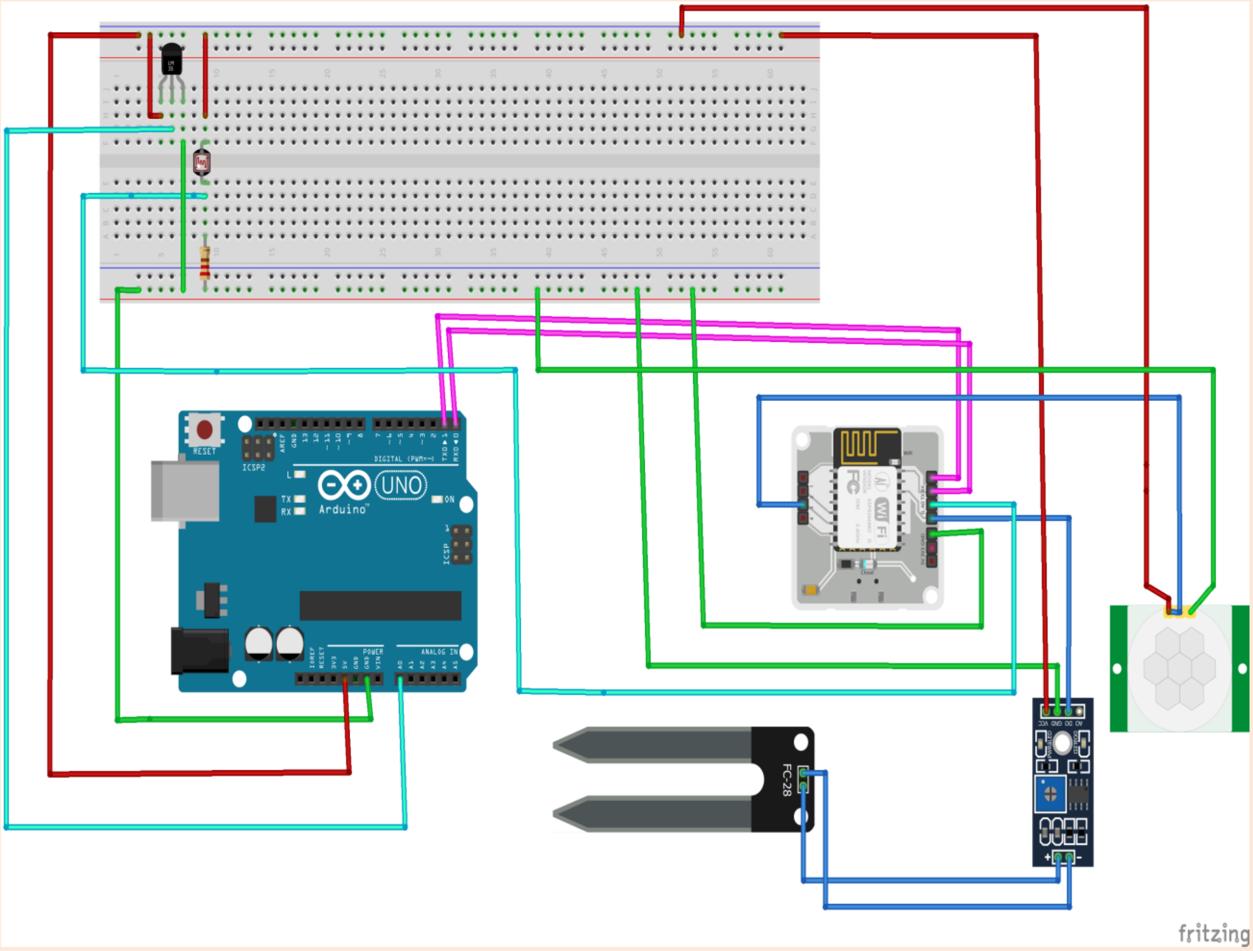
js

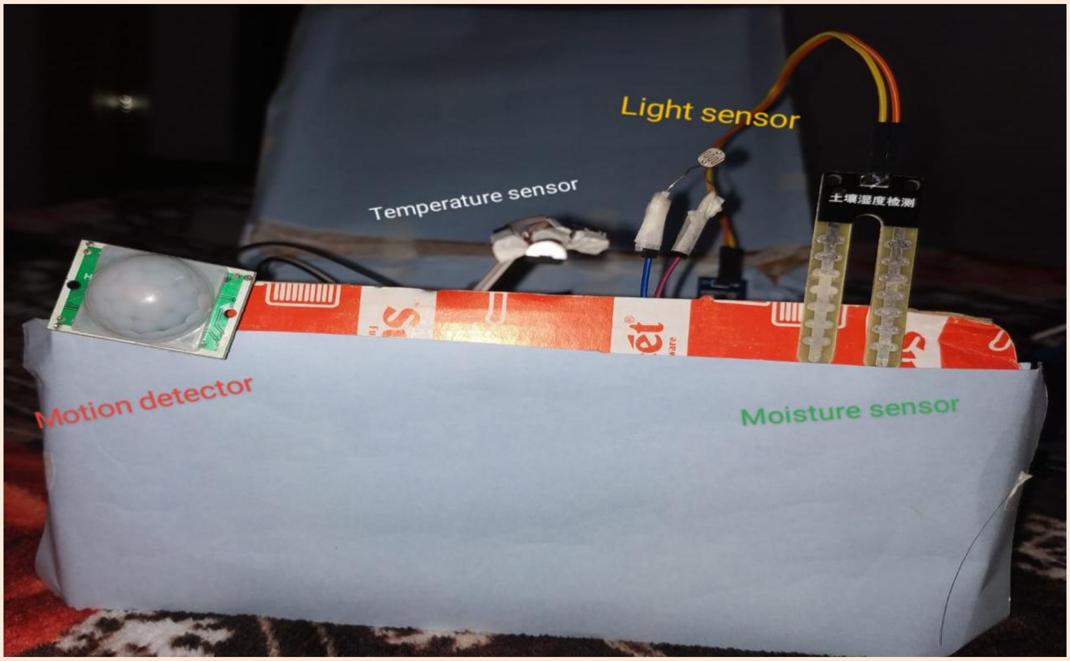
Start typing your code below.

```
setChartTitle('Sunlight Intensity graph');
var Graph1 = new boltGraph();
Graph1.setChartType("areaGraph");
Graph1.setAxisName('Time_stamp','Sunlight');
Graph1.plotChart('time_stamp','sunlight');
var Graph2 = new boltGraph();
Graph2.setChartType("barGraph");
Graph2.setAxisName('Time_stamp','Sunlight');
Graph2.plotChart('time_stamp','sunlight');
var Graph3 = new boltGraph();
Graph3.setChartType("scatterGraph");
Graph3.setAxisName('Time_stamp','Sunlight');
Graph3.plotChart('time_stamp','sunlight');
```

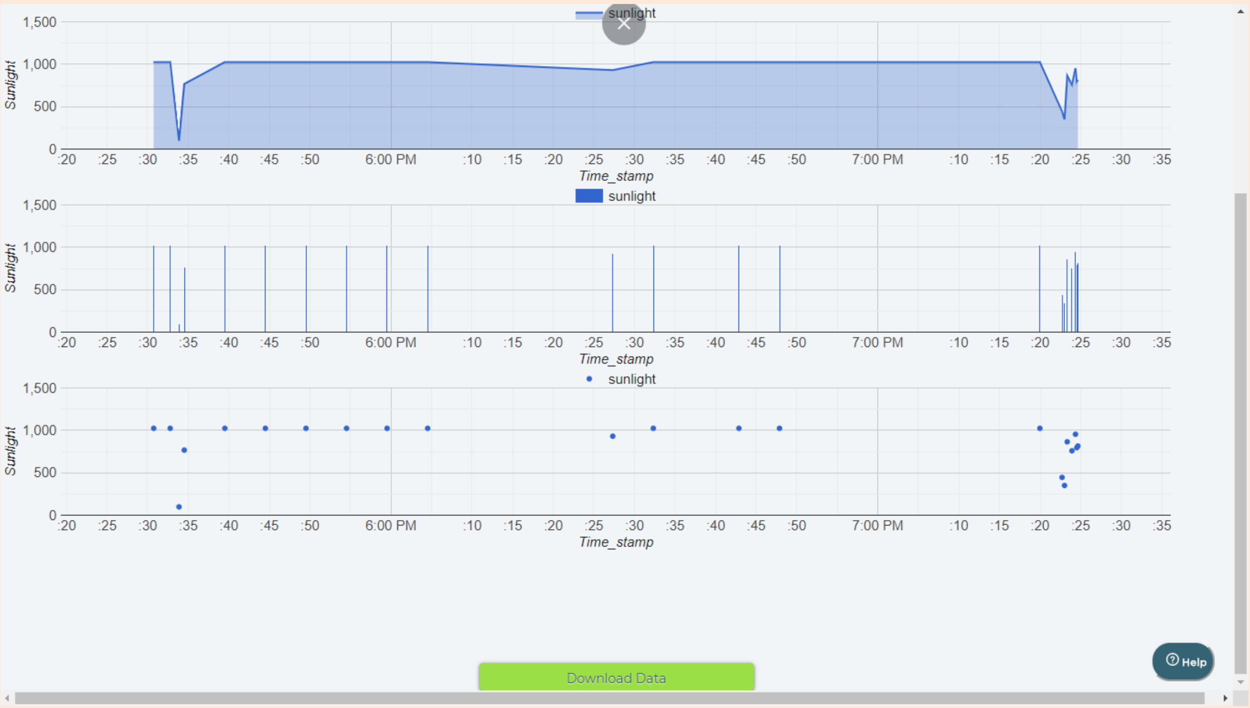
- **STEP 11: Just final check all your connection and connect it to a power source and run the codes.**

Schematics:


RESET
ICSP2
UNO
Arduino
DIGITAL (PWM=~)
ICSP
POWER
ANALOG IN
TX
RX
ON
FC-28
fritzing


Light sensor
Temperature sensor
土壤湿度检测
Motion detector
Moisture sensor

GRAPH BOLT CLOUD:



Thank You