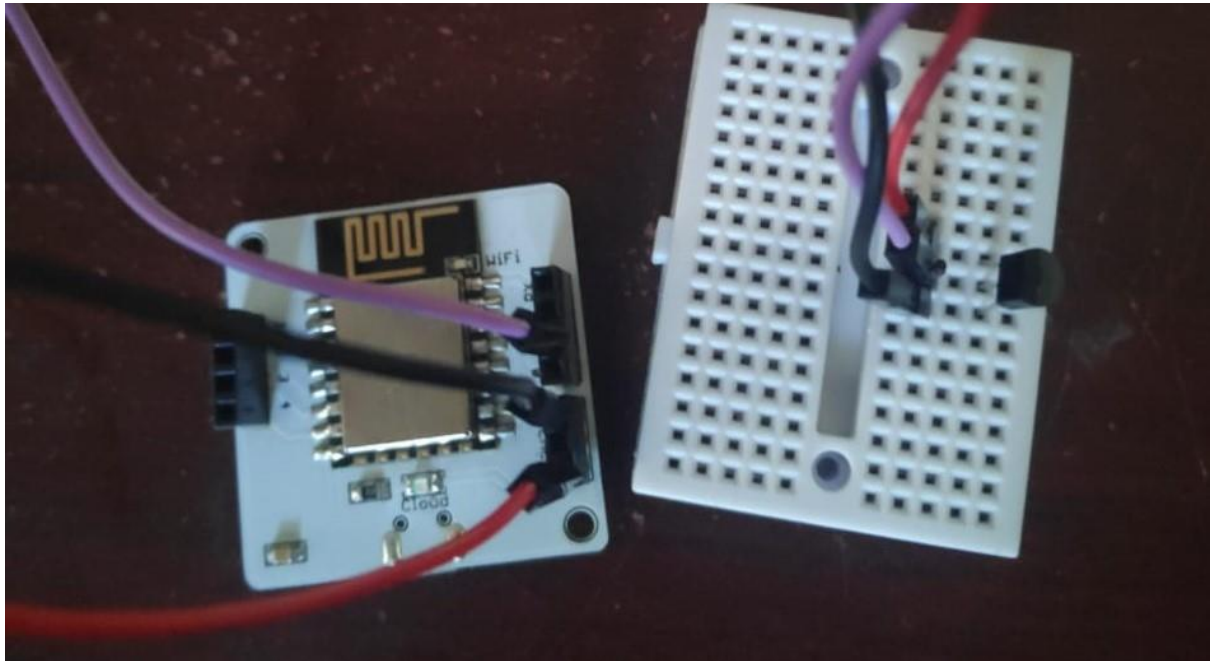


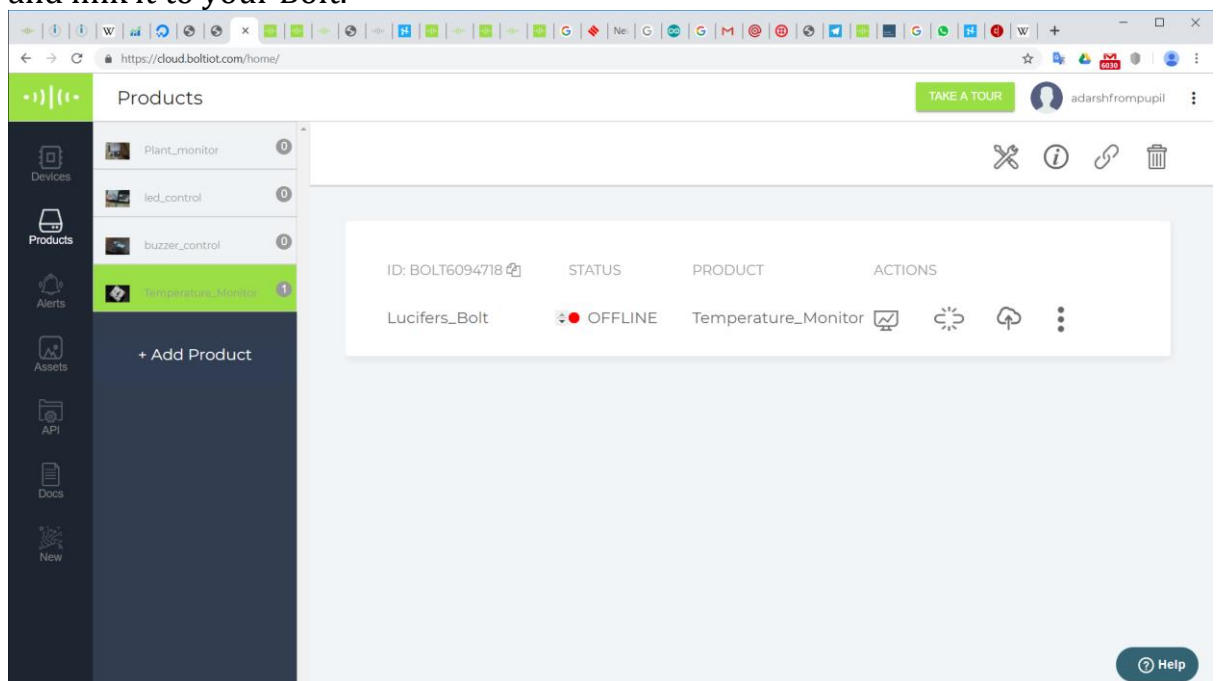
# CAPSTONE PROJECT

## OBJECTIVES:

1. Build the circuit for temperature monitoring system, using the Bolt and LM35 sensor.



2. Create a product on the Bolt Cloud, to monitor the data from the LM35, and link it to your Bolt.

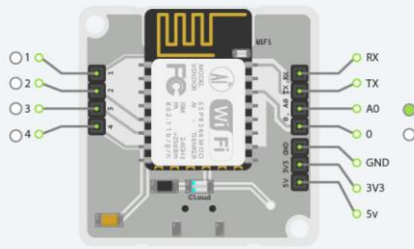


Products: Setup

Temperature\_Monitor  
Output device • Connected with GPIO

Hardware Code

Step 1: Assemble the circuit using Bolt hardware module as per your requirement.  
Step 2: Select the pins as per circuit designed and assign a unique variable name to them.  
Step 3: Data collection rate: 5 Minutes



Note: Variable name can only contain lowercase alphanumeric characters and underscore and should start with an alphabet.

Pin	Variable Name
A0 Analog	temp

DELETE CONFIGURATION

Help

- Write the product code, required to run the polynomial regression algorithm on the data sent by the Bolt.

Products: Setup

Temperature\_Monitor  
Output device • Connected with GPIO

Hardware Code

Write your code in the code window below.

predict JS

```

1 setChartLibrary('google-chart');
2 setChartTitle('Capstone Project');
3 setCrosshair(true);
4 setChartType('PredictionGraph');
5 setAxisName('time_stamp', 'temp');
6 mul(0.0977);
7 plotChart('time_stamp', 'temp');

```

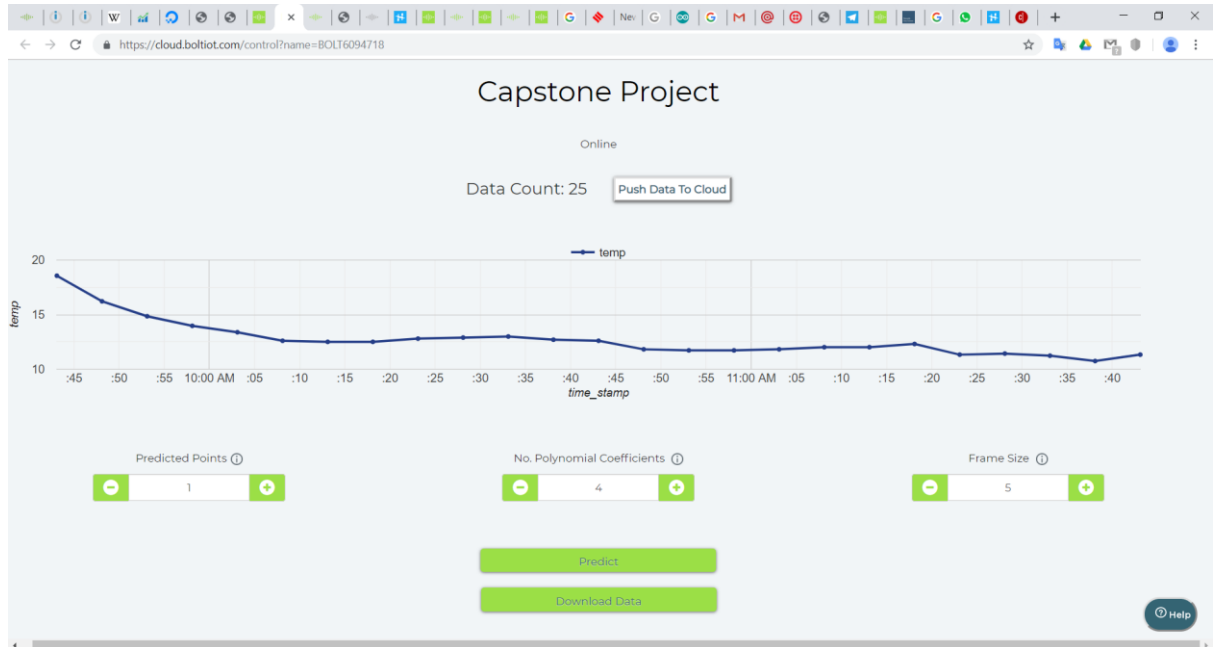
[Click here](#) to access the Documentation

Note: Variable name can only contain lowercase alphanumeric characters and underscore and should start with an alphabet.

Pin	Variable Name
A0 Analog	temp

Help

1. Keep the temperature monitoring circuit inside your fridge with the door of the fridge closed, and let the system record the temperature readings for about 2 hours.



2. Using the reading that you received in the 2 hours, set boundaries for the temperature within the fridge.

```
root@ubuntu-s-4vcpu-8gb-blr1-01: ~/Capstone_project
GNU nano 2.9.3 conf.py
API_KEY='41cb[REDACTED]'
DEVICE_ID='BOLT6094718'
MAILGUN_API_KEY='fff309[REDACTED]'
SANDBOX_URL='sandbox50b3e4eebf347fe813578fad1d262ab.mailgun.org'
FROM_EMAIL='test@sandbox50b3e4eebf347fe813578fad1d262ab.mailgun.org'
TO_EMAIL='adarshfrompupil@gmail.com'

THRESHOLD_MIN=10
THRESHOLD_MAX=15

[ Read 11 lines ]
```

- Write a python code which will fetch the temperature data, every 10 seconds, and send out an email alert, if the temperature goes beyond the temperature thresholds you decided on in objective 'e'.

```
root@ubuntu-s-4vcpu-8gb-blr1-01: ~/Capstone_project
GNU nano 2.9.3 Anomaly_detection.py

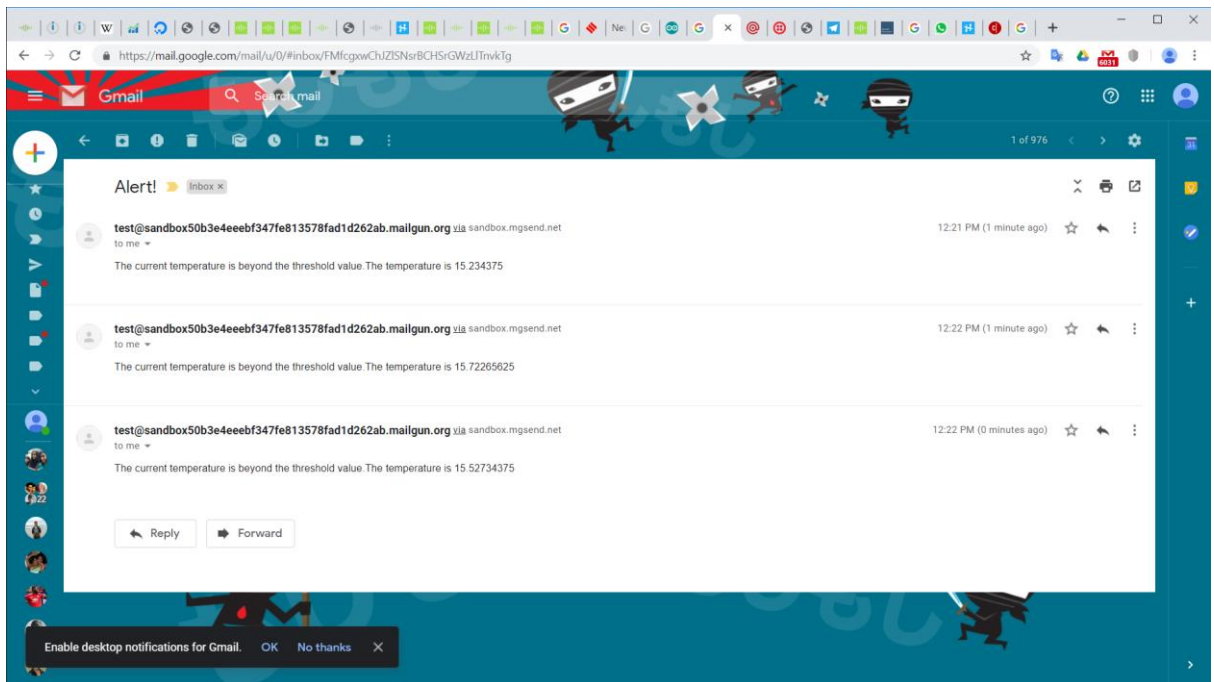
import json, sys, time
from boltiot import Email, Bolt
from conf import *

dev=Bolt(API_KEY, DEVICE_ID)
mailer=Email(MAILGUN_API_KEY, SANDBOX_URL, FROM_EMAIL, TO_EMAIL)

while 1:
    response=dev.analogRead('A0')
    response=json.loads(response)
    if response['success']!=1:
        print('Unsuccessful Request')
    else:
        sensor_val=float(response['value'])
        print('The sensor value is', sensor_val)
        temp=100*sensor_val/1024
        print('The temperature is ', temp)
        if temp>THRESHOLD_MAX or temp<THRESHOLD_MIN:
            print('The current temperature is beyond the threshold value.Sending email.')
            response=mailer.send_email('Alert!!', 'The current temperature is beyond the threshold value.The temperature is '+str(temp))
            print('The response is ', response.text)

        time.sleep(10)
```

```
root@ubuntu-s-4vcpu-8gb-blr1-01: ~/Capstone_project# python3 Anomaly_detection.py
The sensor value is 133.0
The temperature is 12.98828125
The sensor value is 129.0
The temperature is 12.59765625
The sensor value is 147.0
The temperature is 14.35546875
The sensor value is 149.0
The temperature is 14.55078125
The sensor value is 143.0
The temperature is 13.96484375
The sensor value is 156.0
The temperature is 15.2234375
The current temperature is beyond the threshold value.Sending email.
The response is {
  "id": "<20190611065147.1.F2B1D77DAD4A7502@sandbox50b3e4eeebf347fe813578fadid262ab.mailgun.org>",
  "message": "Queued. Thank you."
}
The sensor value is 161.0
The temperature is 15.72265625
The current temperature is beyond the threshold value.Sending email.
The response is {
  "id": "<20190611065200.1.48CBE8007D427A3C@sandbox50b3e4eeebf347fe813578fadid262ab.mailgun.org>",
  "message": "Queued. Thank you."
}
The sensor value is 159.0
The temperature is 15.52734375
The current temperature is beyond the threshold value.Sending email.
The response is {
  "id": "<20190611065212.1.B06F8F1FCE2F4E97@sandbox50b3e4eeebf347fe813578fadid262ab.mailgun.org>",
  "message": "Queued. Thank you."
}
The sensor value is 151.0
The temperature is 14.74609375
The sensor value is 151.0
The temperature is 14.74609375
C:\Traceback (most recent call last):
  File "Anomaly_detection.py", line 24, in <module>
    time.sleep(10)
KeyboardInterrupt
root@ubuntu-s-4vcpu-8gb-blr1-01: ~/Capstone_project#
```



4. Modify the python code, to also do a Z-score analysis and print the line “Someone has opened the fridge door” when an anomaly is detected.

```

root@ubuntu-14cpu-8gb-b1r1-01: ~/Capstone_project
GNU nano 2.9.3 Anomaly_detection3.py Modified

import json,sys,time,math,statistics
from boltiot import Email,Bolt
from conf import *

dev=Bolt(API_KEY,DEVICE_ID)
mailer=Email(MAILGUN_API_KEY,SANDBOX_URL,FROM_EMAIL,TO_EMAIL)
data_set=[]
r=10
c=3

def set_bound(data_set,r,c):
    if len(data_set)<r:
        return None
    if len(data_set)>=r:
        data_set=data_set[len(data_set)-r:]
        Mn=statistics.mean(data_set)
        Variance=0
        for data in data_set:
            Variance+=math.pow((data-Mn),2)
        Zn=c*math.sqrt(Variance/r)
        h=data_set[r-1]+Zn
        l=data_set[r-1]-Zn
        return [h,l]

while 1:
    response=dev.analogRead('A0')
    response=json.loads(response)
    if response['success']==1:
        print('Unsuccessful Request')
    else:
        sensor_val=float(response['value'])
        print('The sensor value is',sensor_val)
        temp=100*sensor_val/1024
        print('The temperature is ',temp)
        bound=set_bound(data_set,r,c)
        if not bound:
            print('Not enough value to conduct Z-score analysis.Need atleast '+str(r-len(data_set)-1)+' more points.')
            data_set.append(temp)
        else:
            if temp>bound[0]:
                print('Detecting anomaly.Something has opened the fridge.')
                print('Sending mail.')
                response=mailer.send_email('Alert!','Someone has opened the fridge.')
                data_set.append(temp)
            time.sleep(10)

```

Initially set frame size and multiplication factor values

5. Tune the Z-score analysis code, such that, it detects an anomaly when someone opens the door of the fridge.

```
root@ubuntu-s-4vcpu-8gb-bl1-01: ~/Capstone_project
GNU nano 2.9.3 Anomaly detection2.py

import json,sys,time,math,statistics
from boltiot import Email,Bolt
from conf import *

dev=Bolt(API_KEY,DEVICE_ID)
mailer=Email(MAILGUN_API_KEY,SANDBOX_URL,FROM_EMAIL,TO_EMAIL)
data_set=[]
r=10
c=5

def set_bound(data_set,r,c):
    if len(data_set)<=r:
        return None
    if len(data_set)>r:
        data_set=data_set[len(data_set)-r:]
        Mn=statistics.mean(data_set)
        Variance=0
        for data in data_set:
            Variance+=math.pow((data-Mn),2)
        Sqr=math.sqrt(Variance/r)
        H=data_set[r-1]+Mn
        L=data_set[r-1]-Mn
        return [H,L]

while 1:
    response=dev.analogRead('A0')
    response=json.loads(response)
    if response['success']!=1:
        print('Unsuccessful Request')
    else:
        sensor_val=float(response['value'])
        print('The sensor value is',sensor_val)
        temp=100*sensor_val/1024
        print('The temperature is ',temp)
        bound=set_bound(data_set,r,c)
        if not bound:
            print('Not enough value to conduct Z-score analysis.Need atleast '+str(r-len(data_set)-1)+' more points.')
            data_set.append(temp)
        else:
            if temp>bound[0]:
                print('Detecting anomaly.Someone has opened the fridge.')
                print('Sending mail.')
                response=mailer.send_email('Alert!','Someone has opened the fridge.')
                data_set.append(temp)
            else:
                data_set.append(temp)

    time.sleep(10)

root@ubuntu-s-4vcpu-8gb-bl1-01: ~/Capstone_project
The temperature is 12.3046875
The sensor value is 138.0
The temperature is 13.4765625
Detecting anomaly.Someone has opened the fridge.
Sending mail.
The sensor value is 157.0
The temperature is 15.33203125
Detecting anomaly.Someone has opened the fridge.
Sending mail.
The sensor value is 150.0
The temperature is 14.6484375
The sensor value is 145.0
The temperature is 14.16015625
The sensor value is 143.0
The temperature is 13.96484375
The sensor value is 143.0
The temperature is 13.96484375
The sensor value is 141.0
The temperature is 13.76953125
The sensor value is 144.0
The temperature is 14.0625
The sensor value is 142.0
The temperature is 13.8671875
The sensor value is 143.0
The temperature is 13.96484375
The sensor value is 141.0
The temperature is 13.76953125
The sensor value is 140.0
The temperature is 13.671875
The sensor value is 139.0
The temperature is 13.57421875
The sensor value is 140.0
The temperature is 13.671875
The sensor value is 176.0
The temperature is 17.1875
Detecting anomaly.Someone has opened the fridge.
Sending mail.
The sensor value is 165.0
The temperature is 16.11328125
The sensor value is 160.0
The temperature is 15.625
The sensor value is 157.0
The temperature is 15.33203125
The sensor value is 153.0
The temperature is 14.94140625
The sensor value is 154.0
The temperature is 15.0390625
The sensor value is 153.0
The temperature is 14.84140625
```

Finally set values to detect anomaly

