PROJECT

TEMPERATURE SENSING OVER BOLT WIFI MODULE WITH Z-SCORE MACHINELEARNING ALGORITHM

By:

GAURAV MITTAL

Introduction

In this project we are measuring the temperature of the environment using temperature Sensor (LM375) . We are sending this data to the cloud. We are analysing this data on the cloud in form of line graph and if any anomalous peak is shown in the graph we will be alerted.

But always looking at graph and see any anomalous behaviour manually quite difficult. So, I also apply a machine learning algorithm which will tell the anomalous behaviour of the temperature. For this I have use a algorithm called Z- score analysis.

Above can be apply in the real time Scenario like in pharmaceutical companies (medicine manufacturers). In these type of industries temperature play a very important role as per government standards.

1. Temperature should be maintain between -40 to -30 degree Celsius.
2. The temperature of the tablets should never be remain between -33 to 30 degree Celsius for longer than 20 minutes
3. The logs of opening of cooling chamber should be maintained.

For this we have implemented a code in python to solve this problem.

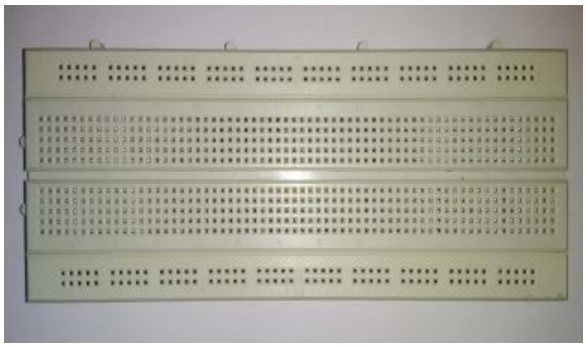
Components required

Hardware

Bolt Wifi Module

LM375 Sensor

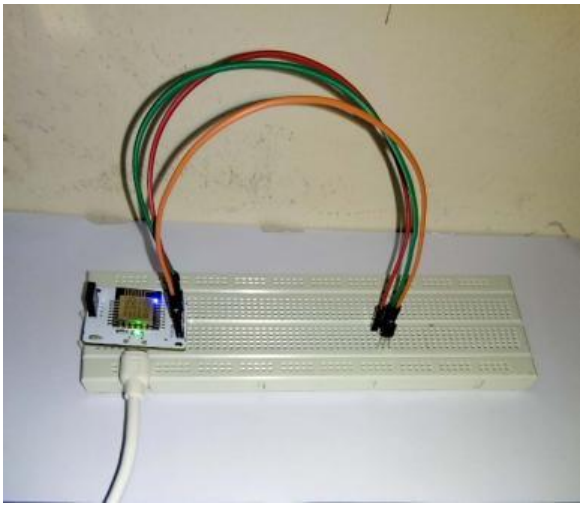
Micro USB Cabel



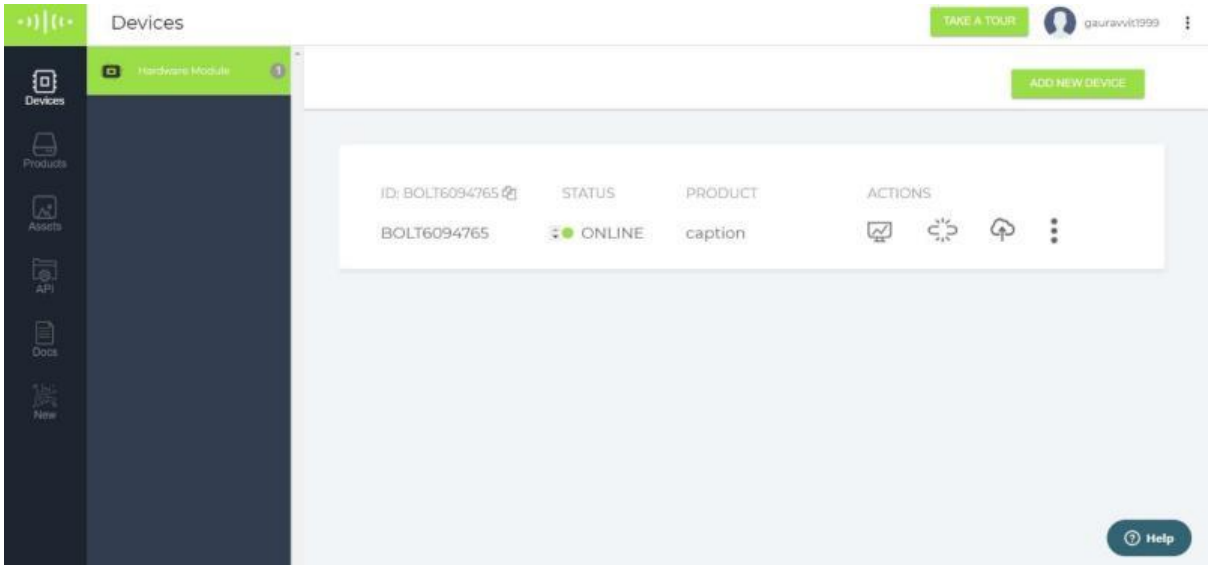
Bread Board

Objectives:

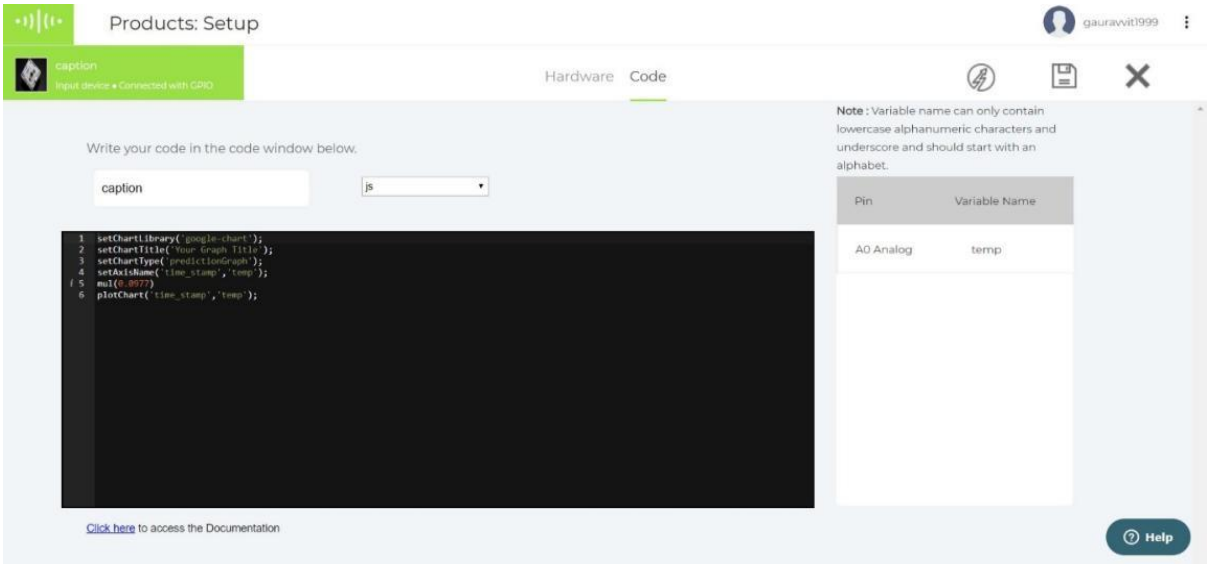
1. Circuit for temperature monitoring System



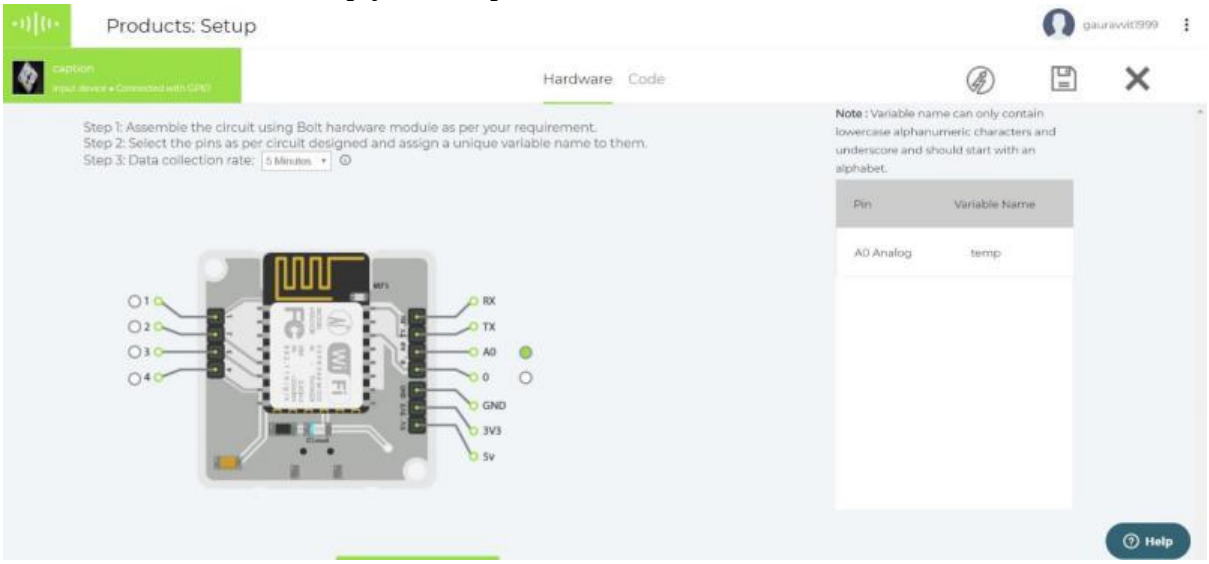
2. Creating new product and linking it to the bolt



3. Polynomial regression code for the product

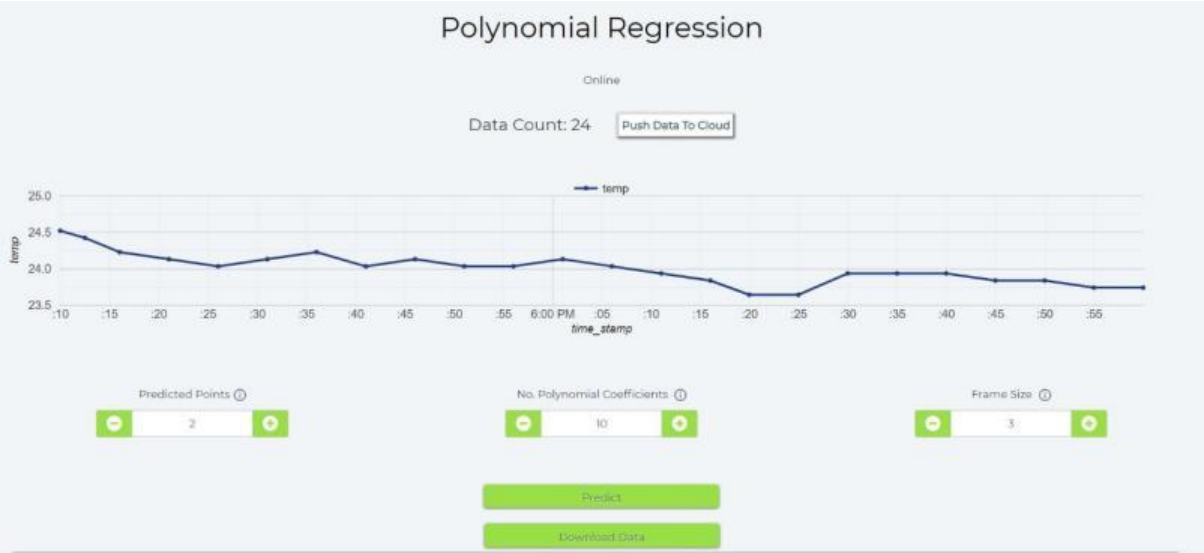


Product code using “google chart” for prediction chart.
In this mul is used to multiply the temp variable with “0.097”



Hardware configuration of bolt wifi.

4. Taking temperature reading for 2 hours with the help of temperature Sensor(lM325)
   Note:- here I have no access to refrigerator to take the reading so I am taking Air conditioner for the readings. So readings will be between 20 to 25 degree Celsius.



Temperature Data Points plotted on the Bolt Cloud.

5. Setting the threshold with help of Z-score to detect anomaly in temperature.

```
if not bound:
 req_size=10-len(old_data)
 print("not enough data")
 old_data.append(sensor_value)
 time.sleep(10)
 continue
```

Checking if enough data is available to calculate lower and higher threshold using Z-score method.

output

```
ubuntu@ubuntu:~$ sudo python3 devicestaus.py
the temperature is 23.8388
not enough data
the temperature is 23.9365
not enough data
the temperature is 23.8388
not enough data
the temperature is 23.9365
not enough data
the temperature is 24.0342
not enough data
the temperature is 23.9365
not enough data
the temperature is 23.9365
not enough data
the temperature is 23.9365
not enough data
the temperature is 23.9365
not enough data
the temperature is 23.8388
not enough data
the temperature is 24.0342
temp out of bound
The door is opened
```

Not enough data is present to compute lower and upper bound of the threshold

After enough (10 data points) are available then it will Compute the high and low bound using the function “cbounds()” mention below.

```
def cbounds(old_data,size,factor):
 if len(old_data)<size:
  return None
 if len(old_data)>size:
  del old_data[0:len(old_data)-size]
 mn=statistics.mean(old_data)
 var=0
 for d in old_data:
  var=var+math.pow((d-mn),2)
 zn=factor*math.sqrt(var/size)
 hbound=old_data[size-1]+zn
 lbound=old_data[size-1]-zn
 return [hbound,lbound]
```

To set the bounds for threshold we call the function “cbounds()” which contain three arguments:.

Old_data: it is the data collected by sensor previously in time. With the help of this data we will z score for setting threshold

Size: it is the size of the frame or the number of data points we will use to calculate the bounds.

factor : Multiplication Factor

6. Python code to fetch data every 10 seconds and sending out the email and SMS if temperature crosses the threshold.

THIS IS THE ARGUMENTS TO PASS WHEN FORMING THE BOLTIOT OBJECT

API_KEY = 'This is Bolt Cloud accout API key'
DEVICE_ID = 'This is the ID of Bolt device'

```
mybolt=Bolt(apikey,deviceid)
```

```
response=mybolt.analogRead('A0')
data=json.loads(response)
sensor_value=(int(data['value']))*0.0977
print("the temperature is"+str(sensor_value))
```

Taking data from bolt cloud by making request through python. Storing that data in Jason “data” object and printing the sensor value.

```
try:
 if sensor_value>bound[0] or sensor_value<bound[1]:
  print("temp out of bound")
  print("The door is opened")
  message()
 if c>12:
  print("temperature is between 25 and 26 degree celcius for more than 20 minutes")
  message()
  c=0
except Exception as e:
 print("Error",e)
time.sleep(10)
```

This is used to check the whether the current temperature measured crosses the threshold.

ARGUMENTS PASS WHEN DIFINING THE OBJECT OF SMS LIBRARY

SSID = 'find SSID in Twilio Dashboard'

AUTH_TOKEN = 'find on Twilio Dashboard'

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

TO_NUMBER = 'This is sender number. Make sure to +91 in beginning'

ARGUMENTS PASS WHEN DIFINING THE OBJECT OF MAIL LIBRARY

MAILGUN_API_KEY = 'This is the private API key which can be find on Mailgun Dashboa rd'

SANDBOX_URL= 'find this on Mailgun Dashboard'

SENDER_EMAIL = 'This would be test@your SANDBOX_URL'

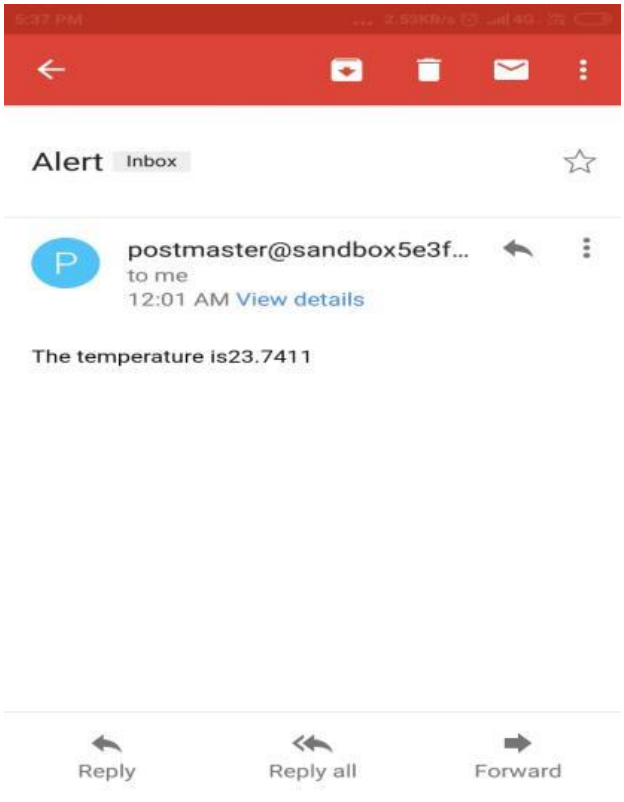
RECIPIENT_EMAIL = 'Enter senders Email ID Here'

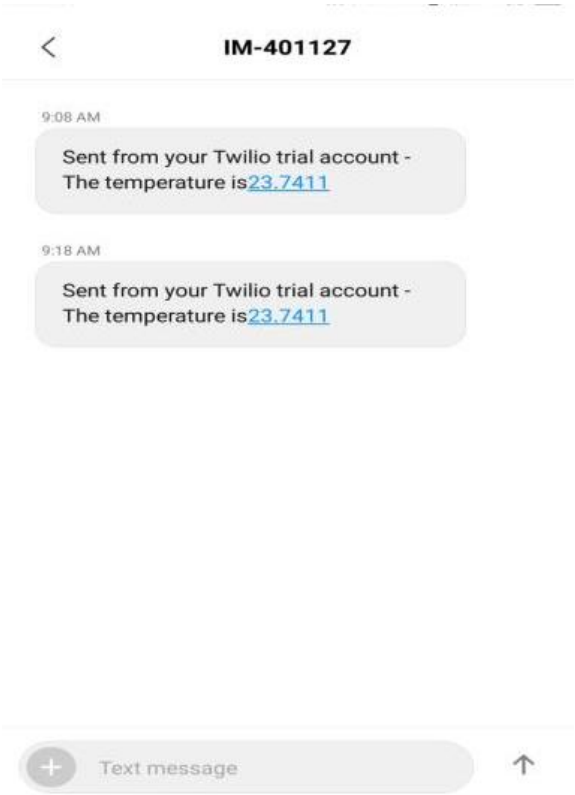
```
def message():
sms = Sms('ACdba7ec1aef98d1338f83d71396dbc4bb', '379ccfa958619f20e7093acfb0286d8e', '+919080085348$
res1=sms.send_sms("The temperature is"+str(sensor_value))
mail=Email('c69016a4c298d6353d2ff83a8f915630-060550c6-f4107ed0','sandbox5e3f5fef9a7a4557838276a7a4$
res2=mail.send_email("Alert","The temperature is"+str(sensor_value))
time.sleep(300);
return 0;
```

Whenever temperature crosses threshold this message function is called.

In this function we sending the message and email to the concerned people about the temperature value. After this we will wat for 5 min and then again check for the temperature value.

Here we are waiting for 5 min because giving alert every 10 second in real world does not make any sense. So after delivering one message we will wait for 5 min.





Email Alert **Message Alert**

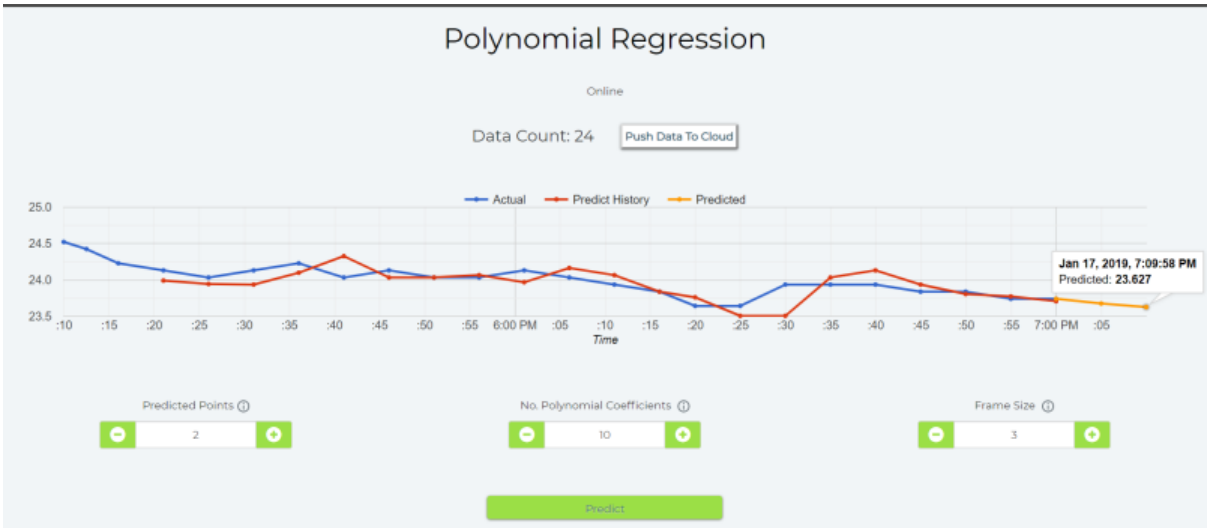
7. With the help of z-Score analysis, tuning the python code such that whenever door of cooling chamber is opened anomaly is detected.

```
try:
 if sensor_value>bound[0] or sensor_value<bound[1]:
  print("temp out of bound")
  print("The door is opened")
  message()
 if c>12:
  print("temperature is between 25 and 26 degree celcius for more than 20 minutes")
  message()
  c=0
except Exception as e:
 print("Error",e)
time.sleep(10)
```

output

```
the temperature is 24.0342
temp out of bound
The door is opened
```

Whenever anomaly is detected with the help of z-Score algorithm. It indicates either door has been opened or something wrong with cooling chamber.



This Graph depicts the actual and the predicted temperature in form of line graph.

Whole python of the code:

```
 GNU nano 2.5.3                File: devicestaus.py

from boltiot import Bolt, Sms, Email
import json, time, math, statistics
min=200*0.0977
max=600*0.0977
apikey="4939eb00-fbd2-472c-98f8-87e820f20e7a"
deviceid="BOLT6094765"
mybolt=Bolt(apikey,deviceid)
def cbounds(old_data,size,factor):
 if len(old_data)<size:
  return None
 if len(old_data)>size:
  del old_data[0:len(old_data)-size]
 mn=statistics.mean(old_data)
 var=0
 for d in old_data:
  var=var+math.pow((d-mn),2)
 zn=factor*math.sqrt(var/size)
```

```
 hbound=old_data[size-1]+zn
 lbound=old_data[size-1]-zn
 return [hbound,lbound]
def message():
 sms = Sms('ACdba7ec1aef98d1338f83d71396dbc4bb', '379ccfa958619f20e7093acfb0286d8e', '+919080085348$
 res1=sms.send_sms("The temperature is"+str(sensor_value))
 mail=Email('c69016a4c298d6353d2ff83a8f915630-060550c6-f4107ed0','sandbox5e3f5fef9a7a4557838276a7a4$
 res2=mail.send_email("Alert","The temperature is"+str(sensor_value))
 time.sleep(300);
 return 0;
old_data=[]
c=0
while True:
 response=mybolt.analogRead('A0')
 data=json.loads(response)
 sensor_value=(int(data['value']))*0.0977
 print("the temperature is"+str(sensor_value))
```

```
if sensor_value>25 and sensor_value<26:
 c=c+1
bound=cbounds(old_data,10,3)
if not bound:
 req_size=10-len(old_data)
 print("not enough data")
 old_data.append(sensor_value)
 time.sleep(10)
 continue
try:
 if sensor_value>bound[0] or sensor_value<bound[1]:
  print("temp out of bound")
  print("The door is opened")
  message()
 if c>12:
  print("temperature is between 25 and 26 degree celcius for more than 20 minutes")
  message()
  c=0
except Exception as e:
 print("Error",e)
time.sleep(10)
```

Conclusion

This project is very useful where temperature play a very vital role. Like in pharmaceutical industries, food industry etc. this temperature measuring and predicting the data from the past analysis of the data provide people power to work and take decisions effectively and efficiently. In this project I have use some basic APIs so that the concerned people can be informed instantly when an anomaly in the temperature occurred. So through this project many temperature concerning problem can be solved.