D.R. B.R AMBEDKAR NATIONAL INSTITUTE OF TECHNOLOGY
JALANDHAR-144011, PUNJAB (INDIA)

PROJECT REPORT

[Summer Internship (May 2019-July 2019)]

Temperature Monitoring System

Submitted by :
Name: Anmol Manik Singh
Roll No: 16102012
Department: ECE(Final year)

ABSTRACT

This project work entitled **'Temperature Monitoring System'** has been designed to keep a check on temperature and whenever the temperature lies outside the permissible limits then an alert message will be send through SMS and EMAIL.

The project aims in developing a computer application through which the information fetched by the temperature sensors will be processed and analysed to determine whether the concerned components are working properly or is there any problem in any of the component. If an anomaly is detected that is either temperature is less than the lower bound or greater than the upper bound then immediately the message regarding the anomaly will be delivered to the concerned authorities.

The hardware components used in the project are Bolt Wi-Fi module, LM35 temperature sensor. Bolt cloud was used to gather the information fetched by sensor and process the information to visualize the data. A python script was written to monitor the temperature value and send the alert message whenever an anomaly is detected. Twilio API's were used to send SMS and Mailgun API's were used to send EMAILS.

Use Cases :

- Temperature can be remotely controlled to avoid issues with refrigerators that might not be working at all times but whose temperature is critical to preserve the food quality over time.

- Temperature of coaches can be remotely controlled to optimise the working of air conditioners.

ACKNOWLEDGMENTS

The training opportunity I had with **RAIL COACH FACTORY** was a great chance for me to learn and develop logical thinking. Therefore, I consider myself as a very lucky individual as I was provided with an opportunity to be a part of it. I am also grateful for having a chance to meet so many wonderful people and professionals who led me though this training period.

I take this opportunity to express a deep sense of gratitude to **Mr S.K Gulati**, for his cordial support, valuable information and guidance, which helped me in completing this task through various stages, his unfailing cooperation and sparing his valuable time to assist me in various training related issues throughout the training period.

Bearing in mind previous I am using this opportunity to express my deepest gratitude to **.Varun Nayyar (Director), Itronix Solution, Jalandhar** for his unfailing cooperation and guidance throughout the training period.

It is my radiant sentiment to place on record my best regards, deepest sense of gratitude to **Mr. B.S Saini**, Electronics and Communication Department (HOD), and all teachers of the department for their careful and precious guidance which were extremely valuable for my study both theoretically and practically.

Last but not the least I would like to mention here that I am greatly indebted to each and everybody who has been associated with my project at any stage but whose name does not find a place in this acknowledgement.

TABLE OF CONTENTS

1.INTRODUCTION

1.1 OVERVIEW:

Temperature Monitoring System refers to analysing the data being collected by temperature sensor and deciding whether the systems are working normally or is there any fault in any part of the system. The data being collected by the sensors is used to do the Z score analysis which determines the permissible lower and upper bounds of temperature value and continuously comparing the current reading with the bounds and determining whether it is in the allowed range or not if due to any reason an anomaly is detected then the alert message will be send to the concerned authorities.

This type of system can be used to remotely control the temperature in AC coaches for better passenger experience and optimising the power consumption by ACs.

Similarly this can also be used to avoid issues with refrigerators that might not be working at all times but whose temperature is critical to preserve the food quality over time.

KEY FEATURES:

- Continuous monitoring of the temperature.
- User friendly interface, easy to operate the application.
- Accurate analysis of data being collected by the sensors.
- Pictorial representations, graphs make it really easy to analyze the various trends in the temperature values being retrieved.
- Instant notification to the concerned authorities whenever an anomaly is detected.

1.2 Internet of Things:

1.2.1 What is IoT?

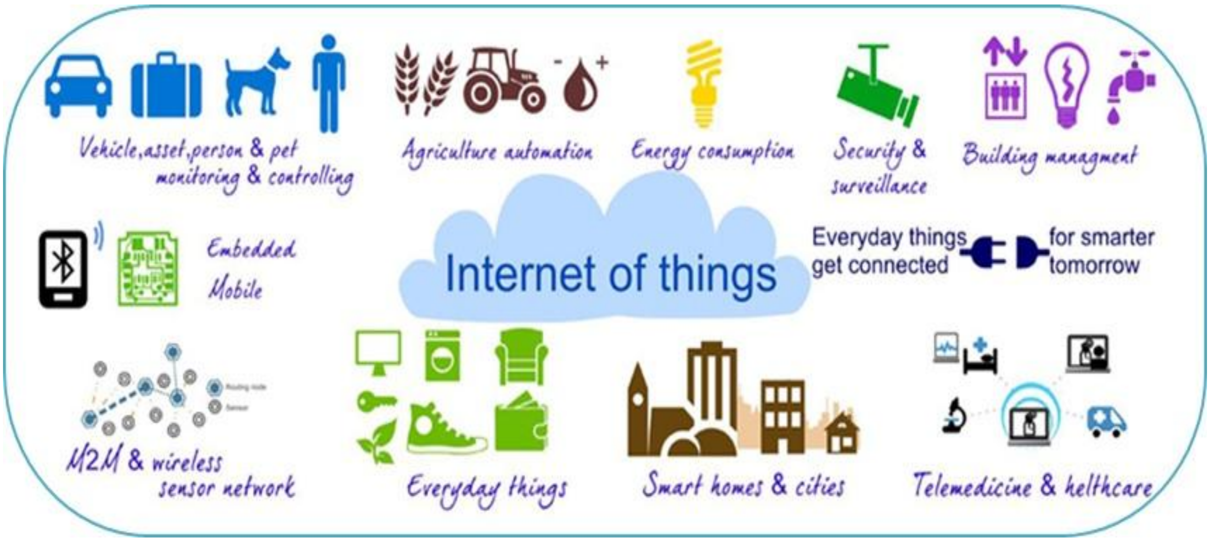
The **Internet of things** (**IoT**) is the extension of Internet connectivity into physical devices and everyday objects. Embedded with electronics, Internet connectivity, and other forms of hardware (such as sensors), these devices can communicate and interact with others over the Internet, and they can be remotely monitored and controlled.

The definition of the Internet of things has evolved due to the convergence of multiple technologies, real-time analytics, machine learning, commodity sensors, and embedded systems. Traditional fields of embedded systems, wireless sensor networks, control systems, automation (including home and building automation), and others all contribute to enabling the Internet of things. In the consumer market, IoT technology is most synonymous with products pertaining to the concept of the "smart home", covering devices and appliances (such as lighting fixtures, thermostats, home security systems and cameras, and other home appliances) that support one or more common ecosystems, and can be controlled via devices associated with that ecosystem, such as smart phones and smart speakers.

The term Internet of Things is 16 years old. But the actual idea of connected devices had been around longer, at least since the 70s. Back then, the idea was often called "embedded internet" or "pervasive computing". But the actual term "Internet of Things" was coined by Kevin Ashton in 1999 during his work at Procter&Gamble. Ashton who was working in supply chain optimization, wanted to attract senior management's attention to a new exciting technology called RFID. Because the internet was the hottest new trend in 1999 and because it somehow made sense, he called his presentation "Internet of Things".

1.2.2 Applications of IoT:

The extensive set of applications for IoT devices is often divided into consumer, commercial, industrial, and infrastructure spaces.



1 - Consumer applications

A growing portion of IoT devices are created for consumer use, including connected vehicles, home automation, wearable technology (as part of Internet of Wearable Things (IoWT), connected health, and appliances with remote monitoring capabilities.

2 - Smart homes

IoT devices are a part of the larger concept of home automation, which can include lighting, heating and air conditioning, media and security systems. Long-term benefits could include energy savings by automatically ensuring lights and electronics are turned off.

3 - Medical and healthcare

The Internet of Medical Things (also called the internet of health things) is an application of the IoT for medical and health related purposes, data collection and analysis for research, and monitoring. This 'Smart Healthcare' as it is also called, led to the creation of a digitized healthcare system, connecting available medical resources and healthcare services.

IoT devices can be used to enable remote health monitoring and emergency notification systems. These health monitoring devices can range from blood pressure and heart rate monitors to advanced devices capable of monitoring specialized implants, such as pacemakers, Fitbit electronic wristbands, or advanced hearing aids. Some hospitals have begun implementing "smart beds" that can detect when they are occupied and when a patient is attempting to get up. It can also adjust itself to ensure appropriate pressure and support is applied to the patient without the manual interaction of nurses.

4 - Transportation

The IoT can assist in the integration of communications, control, and information processing across various transportation systems. Application of the IoT extends to all aspects of transportation systems (i.e. the vehicle, the infrastructure, and the driver or user). Dynamic interaction between these components of a transport system enables inter- and intra-vehicular communication, smart traffic control, smart parking, electronic toll collection systems, logistics and fleet management, vehicle control, safety, and road assistance.

5 - Building and home automation

IoT devices can be used to monitor and control the mechanical, electrical and electronic systems used in various types of buildings (e.g., public and private, industrial, institutions, or residential) in home automation and building automation systems. In this context, three main areas are being covered in literature :

- The integration of the Internet with building energy management systems in order to create energy efficient and IOT-driven "smart buildings".
- The possible means of real-time monitoring for reducing energy consumption and monitoring occupant behaviours.

- The integration of smart devices in the built environment and how they might to know how to be used in future applications.

6 - Agriculture

There are numerous IoT applications in farming such as collecting data on temperature, rainfall, humidity, wind speed, pest infestation, and soil content. This data can be used to automate farming techniques, take informed decisions to improve quality and quantity, minimize risk and waste, and reduce effort required to manage crops. For example, farmers can now monitor soil temperature and moisture and even apply IoT-acquired data to precision fertilization programs.

7 - Infrastructure applications

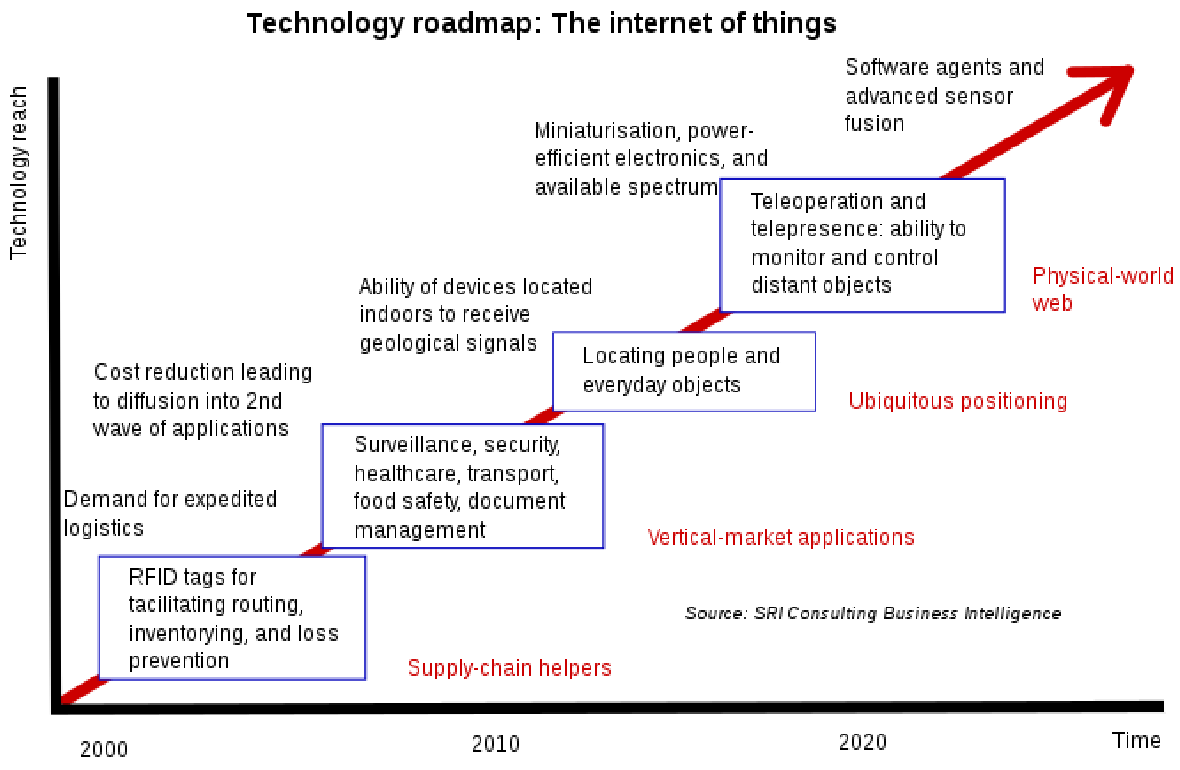
Monitoring and controlling operations of sustainable urban and rural infrastructures like bridges, railway tracks and on- and offshore wind-farms is a key application of the IoT. The IoT infrastructure can be used for monitoring any events or changes in structural conditions that can compromise safety and increase risk. The IoT can benefit the construction industry by cost saving, time reduction, better quality workday, paperless workflow and increase in productivity. It can help in taking faster decisions and save money with Real-Time Data Analytics. It can also be used for scheduling repair and maintenance activities in an efficient manner, by coordinating tasks between different service providers and users of these facilities. IoT devices can also be used to control critical infrastructure like bridges to provide access to ships. Usage of IoT devices for monitoring and operating infrastructure is likely to improve incident management and emergency response coordination, and quality of service, up-times and reduce costs of operation in all infrastructure related areas. Even areas such as waste management can benefit from automation and optimization that could be brought in by the IoT.

1.2.3 Trends and characteristics

The IoT's major significant trend in recent years is the explosive growth of devices connected and controlled by the Internet. The wide range of applications for IoT technology mean that the specifics can be very different from one device to the next but there are basic characteristics shared by most.

The IoT creates opportunities for more direct integration of the physical world into computer-based systems, resulting in efficiency improvements, economic benefits, and reduced human exertions.

The number of IoT devices increased 31% year-over-year to 8.4 billion in the year 2017 and it is estimated that there will be 30 billion devices by 2020. The global market value of IoT is projected to reach $7.1 trillion by 2020.



1.2.4 Importance of IoT

The **"Internet of Things"** (IoT) has the power to change our world. IoT will play an important role in the future and there is expected to be a significant amount of cash flowing through the market in the up-coming years. Over half of major new business processes and systems will incorporate IoT elements by 2020. The impact on consumers' lives and corporate business models is rapidly increasing as the cost of instrumenting physical things with sensors and connecting them to other things devices, systems and people continues to drop.

Internet of Things offers some interesting applications to our lives easier like in Healthcare, Transportation, and Agriculture. However, various factors like security, privacy and data storage also need to be considered.

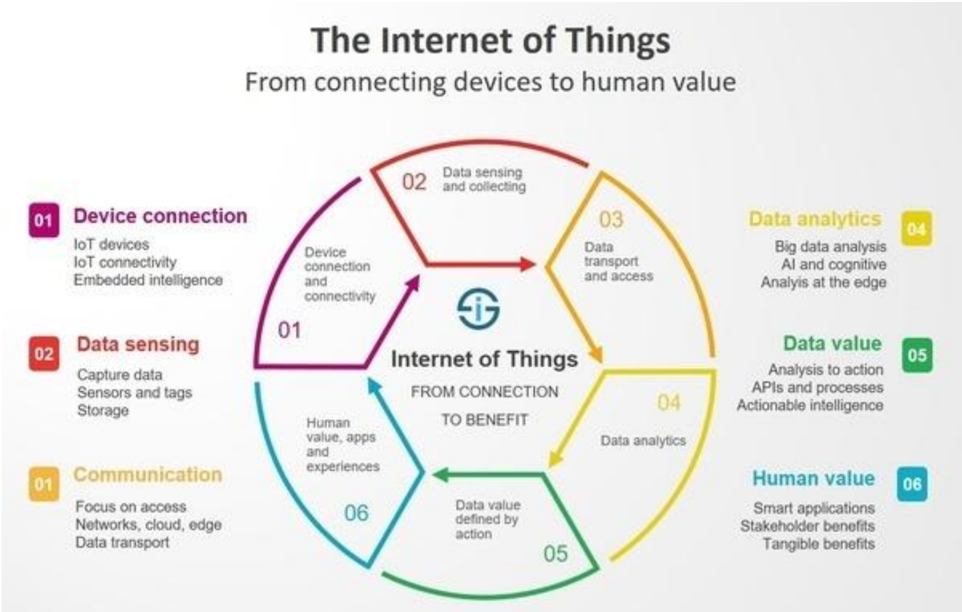
It is also worth noting that things have been connected to networks for ages without the guise of "Internet of Things".

IoT is the connection of objects (things) to the Internet and these objects can collect and exchange data over the Internet and with each other.

The internet of things (IoT) has made the world more efficient, convenient and enjoyable, with the dramatic surge of internet-connected devices transforming how individuals, households, and businesses interact on a daily basis. A recent study released that 43% of all companies are using or planning to implement an IoT application in 2019.

Internet of Things applications is available in every industry for smart homes, construction, travel and transportation, health care and personal care, retail, agriculture, etc. And in industrial area Internet of Things revolves around automation and logistics and creating smarter solutions, programmatically adjusting to human behavior.

There are still some barriers to adoption is there, for example, battery life of devices and cost of devices. GSM and 4G networks are used more and more for IoT applications and new advancements recently made in the network software and device stack will greatly improve on these aspects. IoT helps in finding the right ideas and business model, how to go to the market and secure ROI for the Internet of Things applications are the most critical issues to solve for most companies in the coming years.From digital assistants to baby monitors to sensors monitoring traffic on major highways, a plethora of IoT devices connected to networks are helping drive many benefits for users.



1.2.5 Overcoming Threats to IoT

While the benefits of IoT devices is infinite, the internet of things is vulnerable to security threats.

Here is some technique to secure your IoT devices.

1. Secure Device Hardware Accessibility.

2. Secure Port Communication.

3. Use Encryptions.

4. Use Authentication.

5. Ensure Protection Against Phishing and Malware.

6. Employ Middleware Hardware Physical Security.

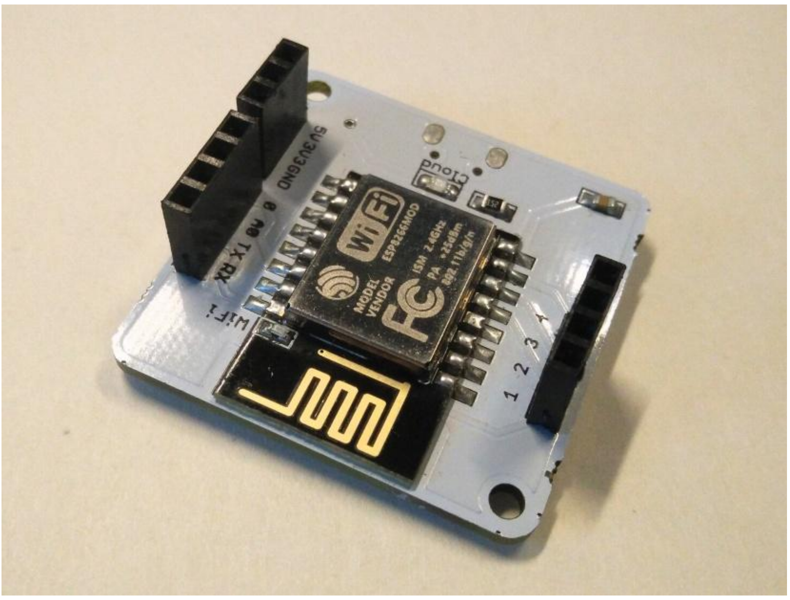
2- HARDWARE REQUIREMENTS

This chapter will include the details regarding the hardware components used to develop the project.

Bolt IoT :

Bolt is an IoT platform to easily and quickly build products and services. Bolt comes with a WiFi/GSM chip and a cloud platform which helps you connect your devices and sensors to the Internet. With Bolt Cloud you can control and monitor them over the internet, create personalised dashboards to visualise the data, monitor the device health, run machine learning algorithms and lot more.

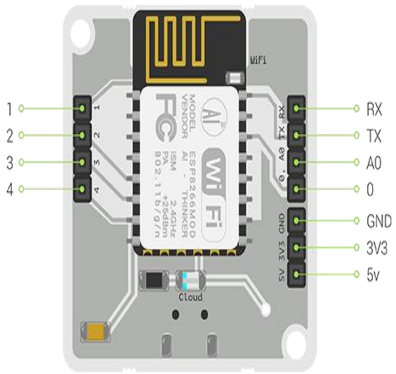
2.1 ESP8266

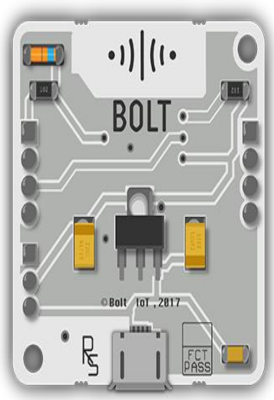


The Internet of Things (IoT) is nothing new, it has been with us for over a decade, but in this time we have seen the price of devices fall from hundreds of dollars to less than $10! The **ESP8266** (and the newer ESP32) have really shaken the world of IoT from being pretty much a novelty industry (we all remember fridges that tweet and ovens that force us to accept an end user licence agreement) into a citizen science revolution where sensors across the world are monitoring climate change, animal migration patterns and much more.

However, all of these good ideas need a power source and the **ESP8266** has become the go-to board largely down to price, and not for its ease of use. But there are boards out there, based on the ESP8266, that offer an easy to use interface that we can use to program the board to do what we want. The Bolt IoT platform is one such board.

2.1.1 Why ESP8266?





ESP8266 is a low-cost Wi-Fi chip with full TCP/IP stack and microcontroller unit capability. It offers a complete and self-contained Wi-Fi networking solution, allowing it to either host the application or to offload all Wi-Fi networking functions from another application processor.

It also has a power LED, a status LED to indicate hardware operation status and a Wi-Fi LED to indicate if the hardware is connected to Wi-Fi or not.

The Cloud is built using a Python based flask server and a message queuing telemetry transport (MQTT) backend to manage devices connected to the Cloud. The developed Cloud stack provides you with a dynamic dashboard using which you can control and monitor your devices over the Internet.

Unlike most other platforms, Bolt offers a hardware chip, Cloud, storage, analytics and visualisation in one integrated package. It also has an excellent partner developer network that provides you with full end-to-end IoT product or service development, which is often preferred by bigger companies.
The hardware unit starts a Wi-Fi hotspot when it is powered on, which is indicated by the blinking of a blue LED. You can connect to this hotspot using your mobile phone

by entering the Wi-Fi credentials for the hardware unit you want to connect to. The device restarts and attempts to connect to the specific Wi-Fi network and, if successful, the blue LED light becomes stable. If not connected, the hardware continues to give out its own Wi-Fi hotspot for setup.

You can register the device to your account using the mobile phone app, after which the device is displayed on your Bolt Cloud dashboard. This gives you access to Control Panel, from where you can setup the device, Control/

Visualise page and the device icon, and so on. You can then proceed to the dashboard and click on the device icon and Control/Visualise it based on the type of the linked page.

Connectivity

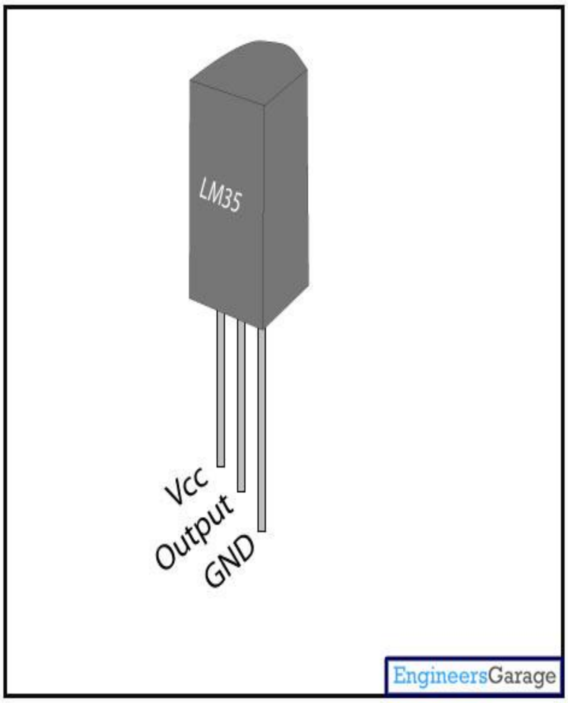
Parameter	Details
WiFi	802.11 b/g/n Automatic AP mode if not connected to WiFi WEP/WPA/WPA2 authentication Only works with 2.4 GHz WiFi
UART	8-N-1 3.3V TTL UART [using TX, RX, GND pins] [2400,4800, 9600,19200 baudrate]
Cloud	Default: Bolt Cloud (https://cloud.boltiot.com) Optional: Custom cloud using Bolt APIs

Parameters	
Connectivity and Processing Module	ESP8266 with custom firmware
MCU	32-bit RISC CPU: Tensilica Xtensa LX106
Power	5V/1A DC via Micro-USB port or 5V and GND pins
Operating Voltage	3.3V
CPU Clock Frequency	80 MHz
MCU Internal Memory	64 KB of instruction RAM; 96KB of data RAM
MCU External Memory	4 MB Flash memory [QSPI]
GPIO pins	5 Digital pins [3.3V logic]
ADC	1 pin 10 bit ADC [0-1V input]
PWM	All 5 Digital pins capable of PWM [Software PWM]
Connectivity	
WiFi	802.11 b/g/n

LED Indicators

Parameter	Details
WiFi LED - WiFi connectivity	1) Slow blinking: Trying to find and connect to WiFi network 2) Fast blinking: User has connected via Bolt IoT app for setup 3) Stable: Connected to WiFi
Cloud LED – Bolt Cloud connectivity	1) Stable: Connected to Bolt Cloud 2) Off: Not connected to Bolt Cloud 3) Dim: Insufficient power/ incorrect boot

2.2 LM35 IC



LM35 is a precision IC temperature sensor with its output proportional to the temperature (in ℃). The sensor circuitry is sealed and therefore it is not subjected to oxidation and other processes. With LM35, temperature can be measured more accurately than with a thermistor. It also possess low self heating and does not cause more than 0.1 ℃ temperature rise in still air.

The operating temperature range is from -55°C to 150°C. The output voltage varies by 10mV in response to every ℃ rise/fall in ambient temperature, i.e., its scale factor is 0.01V/ ℃.

Features of LM35

Calibrated Directly in Celsius (Centigrade)
Linear + 10-mV/°C Scale Factor
0.5°C Ensured Accuracy (at 25°C)
Rated for Full −55°C to 150°C Range
Suitable for Remote Applications
Low-Cost Due to Wafer-Level Trimming
Operates From 4 V to 30 V
Less Than 60-μA Current Drain
Low Self-Heating, 0.08°C in Still Air
Non-Linearity Only ±¼°C Typical
Low-Impedance Output, 0.1 Ω for 1-mA Load

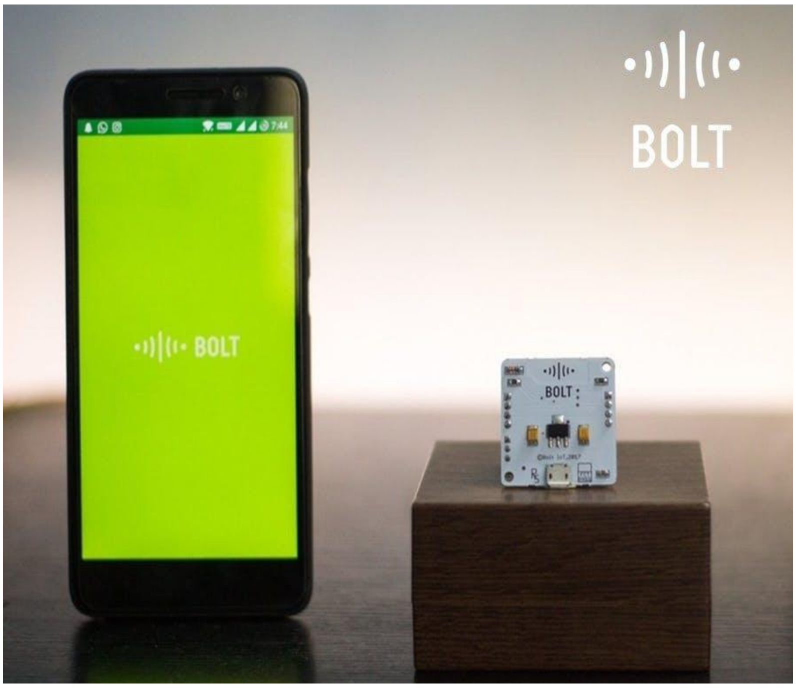
2.3 Jumper Wires :

Jumper wires are simply wires that have connector pins at each end, allowing them to be used to connect two points to each other without soldering. Jumper wires are typically used with breadboards and other prototyping tools in order to make it easy to change a circuit as needed. Fairly simple. In fact, it doesn't get much more basic than jumper wires.

3- SOFTWARE REQUIREMENTS

This chapter will include the details regarding softwares used to develop the project.

3.1 Bolt IoT App:



The App is used to create user account on Bolt cloud. Once the account is created then new devices can be added and linked to the Bolt cloud.

Power ON the Bolt device by inserting the Micro USB cable into the USB port provided on the Bolt and connecting the other end of the cable to the Android charger or to your laptop. Once you have powered ON the Bolt device, blue LED on the Bolt will begin to blink slowly and the green LED will be OFF.

When the blue LED blinks slowly, it means that the Bolt is now transmitting its own WiFi hotspot network and is ready to be setup via the Bolt IoT App on your phone.

The password for the WiFi hotspot transmitted by the Bolt WiFi Module is: **bolt1234** While setting up with the Android mobile app, your mobile phone will automatically connect to the hotspot when you click next.

NOTE: The Bolt WiFi module cannot detect 5 GHz based WiFi networks and will not be able to connect to it.

Select the WiFi network to which Bolt has to be connected. You can click on the WiFi name to choose the WiFi network to connect to.

3.2 Python Library

The Bolt python library provides an easy to use interface of the Bolt Cloud API's .

How to use the Bolt Python library in your code?

Follow the steps given below to use the Bolt python library in your code.

- Install boltiot library first by running the below command on terminal/command line.

Shell

```
sudo pip install boltiot
```

- Create your python file which will use the Bolt API library.

Shell

```
touch file_name.py
```

- Import the library in your python code.

Python

```
from boltiot import Bolt
api_key = "ACXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
device_id  = "BOLT1234"
mybolt = Bolt(api_key, device_id)
```

GPIO Functions :

1- **(digitalWrite Command)-** Write digital output.

Python

```
response = mybolt.digitalWrite('0', 'HIGH')
```

2- **(digitalRead Command) -** Read digital input

Python

```python
response = mybolt.digitalRead('0')
```

3- **(analogWrite Command) - Write analog output**

Python

```python
response = mybolt.analogWrite('0', '100')
```

4- **(analogRead Command) - Read analog output**

Python

```python
response = mybolt.analogRead('A0')
```

Alert Notifications

The Bolt python library provides an easy interface to send Email and SMS alerts using Mailgun and Twilio third-party services respectively.

a) Sending SMS

Twilio is a popular third-party SMS functionality provider. Twilio allows software developers to programmatically make and send and receive text messages using its web service APIs**.**

b) Sending Email

Mailgun is an Email automation service. It has a very powerful set of inbuilt API functions for sending emails. Developers can send emails programmatically with the help of Mailgun API.

CHAPTER 4: PROJECT WORK UNDERTAKEN

4.1 INSTALLED THE REQUIRED SOFTWARE.

4.1.1 Installed Bolt IoT App.

The App can be downloaded from either Google play store or App store. The app is used to link the bolt iot device to the preferred Wi-Fi network.

4.1.2 Linking the device with the cloud.

1. Click on the 'ADD DEVICE' button to setup your Bolt device with your WiFi network.
2. Once you have powered ON the Bolt device, blue LED on the Bolt will begin to blink slowly and the green LED will be OFF.
3. When the blue LED blinks slowly, it means that the Bolt is now transmitting its own WiFi hotspot network and is ready to be setup via the Bolt IoT App on your phone.
4. Click on the '>' symbol on the app to progress further.
5. The password for the WiFi hotspot transmitted by the Bolt WiFi Module is: **bolt1234.**
6. Now, click on 'DONE' and in the next screen, select your Country and click on 'SAVE' to complete the final step of the setup.

7. If the Bolt was able to connect to the WiFi network and is connected to the cloud, a green dot will appear beside the Bolt's device ID as shown below.

Figure 1

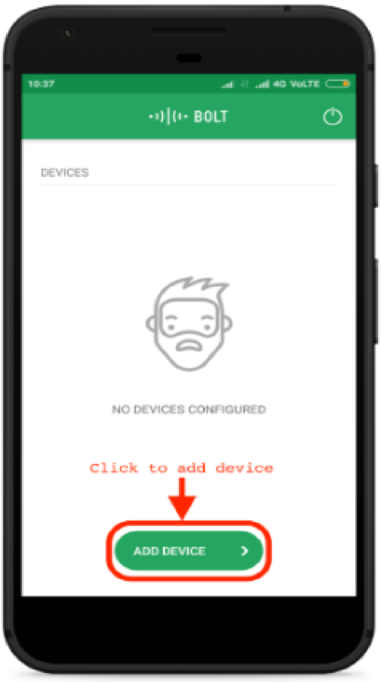


Figure 2

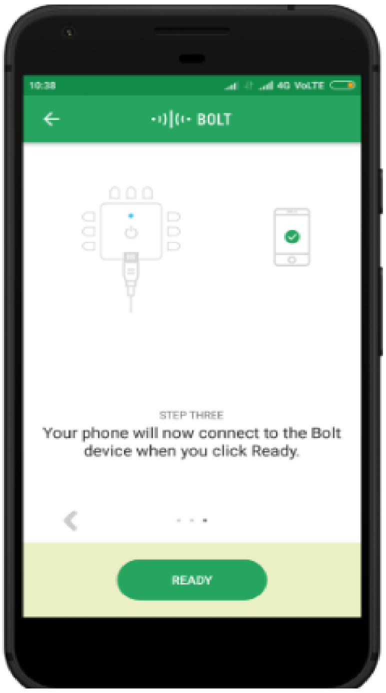


Figure 3

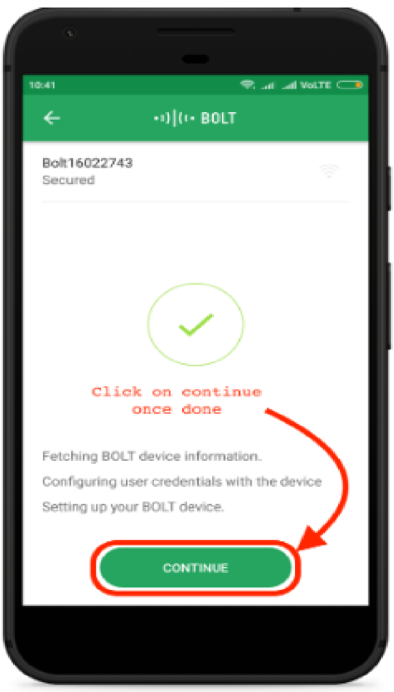


Figure 4

Figure 5

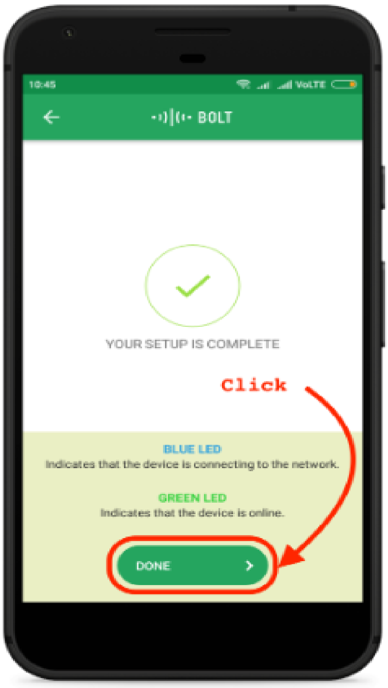
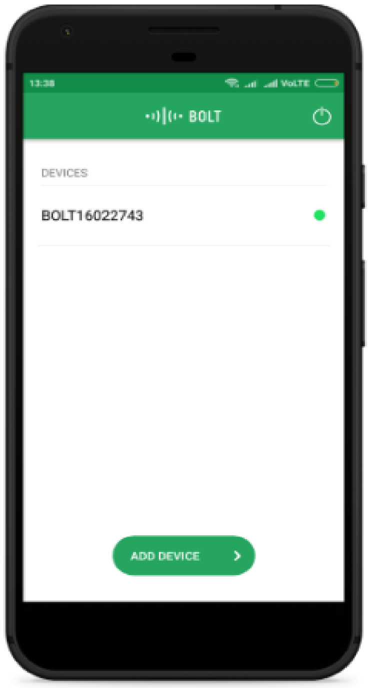
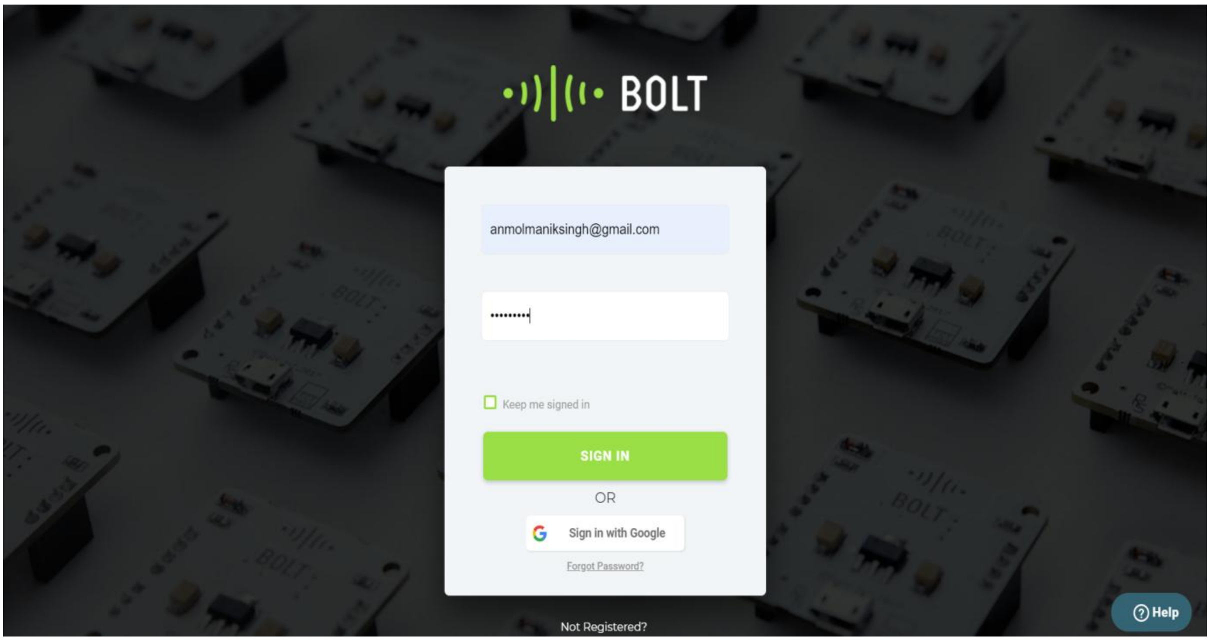


Figure 6

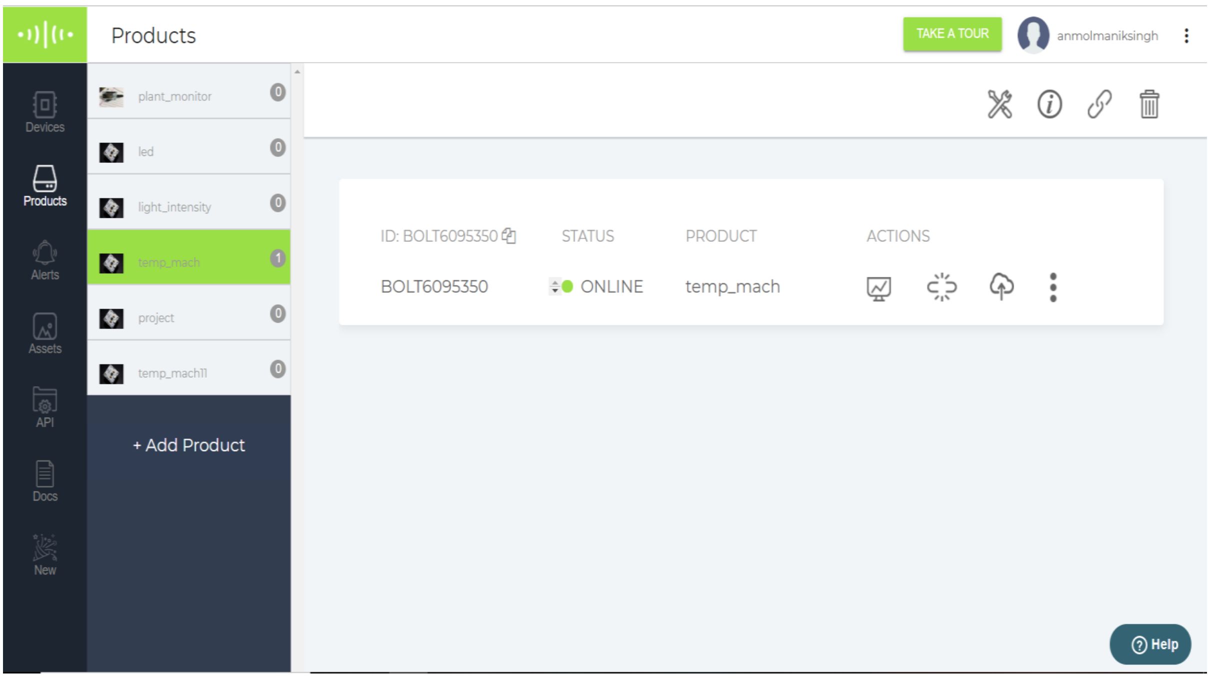


4.2 ACCESSING THE BOLT CLOUD TO BUILD IOT PROJECTS.

- Visit Bolt Cloud (cloud.boltiot.com) and login into your registered account using the same email ID and password that you have used on the Bolt IoT mobile App.



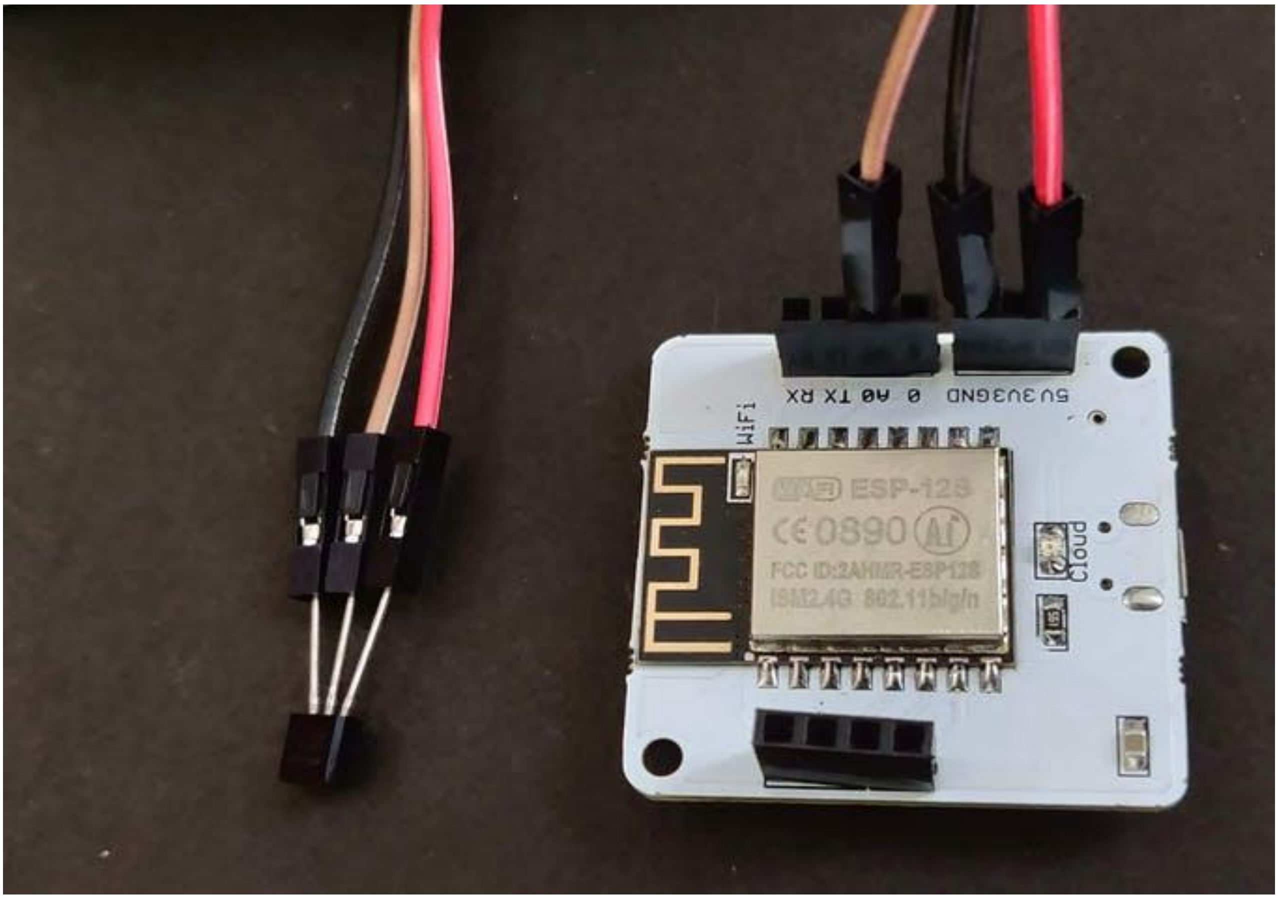
- You will see your Bolt device with status as 'ONLINE' on your account on the dashboard.



4.3 SETTING UP THE CIRCUIT.

Step 1: Connecting LM35 to Bolt :

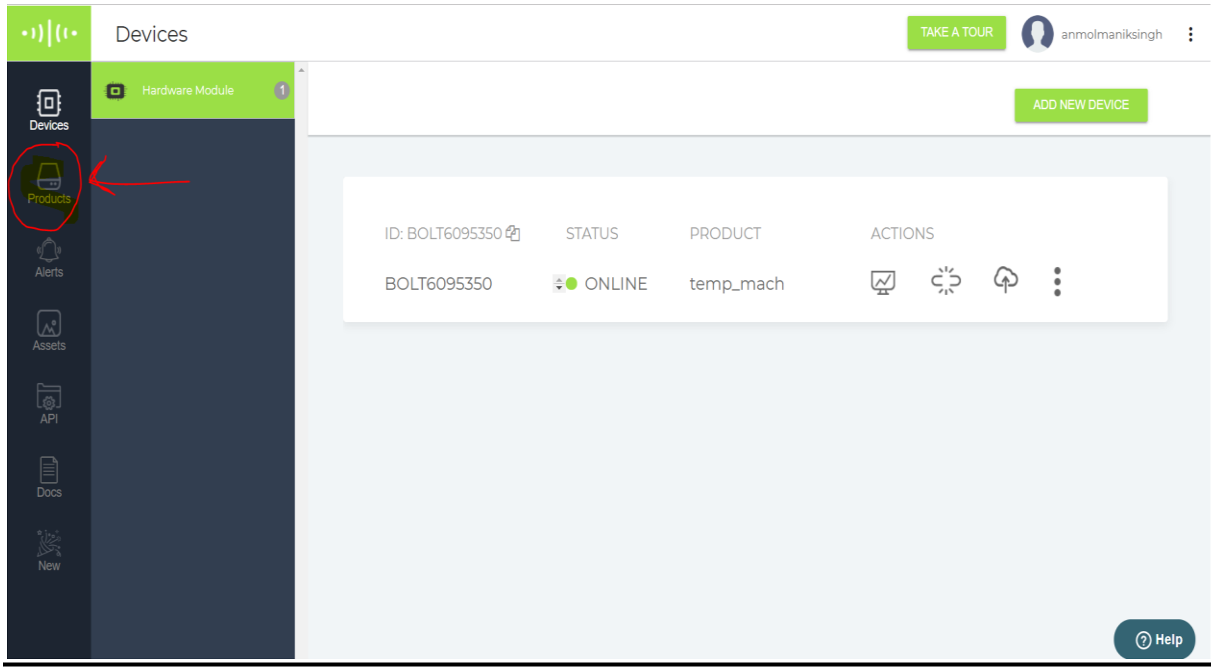
1. Make sure you have not powered on your Bolt Module white connecting the circuit. This will ensure that in case we make any mistake, it will not short circuit your device. Switch off the power if it is connected.
2. Connect the VCC pin of LM35 to 5V pin of the Bolt device.
3. Connect the GND pin of LM35 to GND pin of the Bolt device.
4. Connect the analog output pin of LM35 to A0 (analog input) pin of the Bolt device.



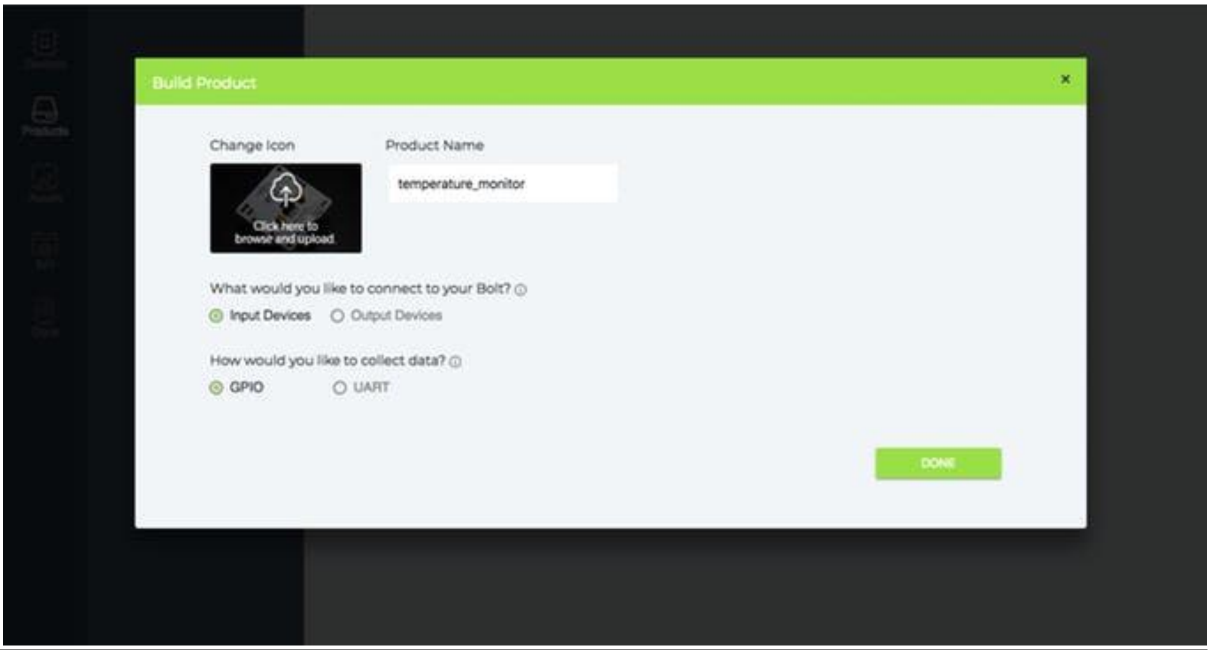
The final circuit setup.

4.4 CREATING NEW PRODUCT.

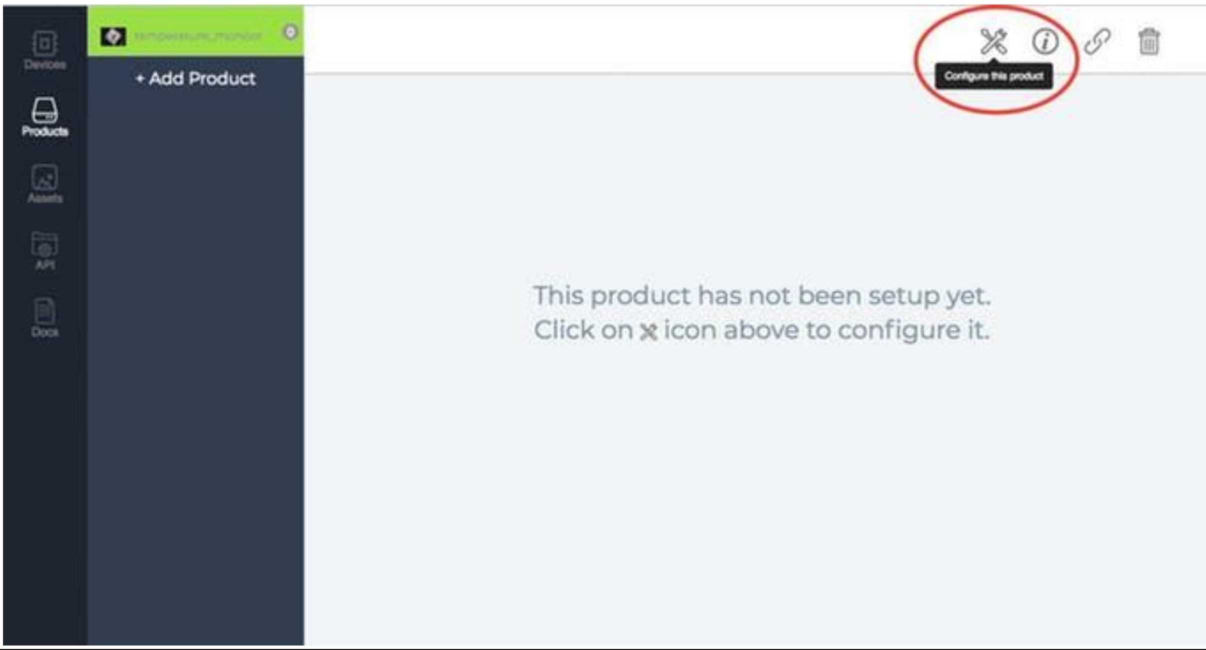
1. Login into https://cloud.boltiot.com and click on the 'Product' tab.



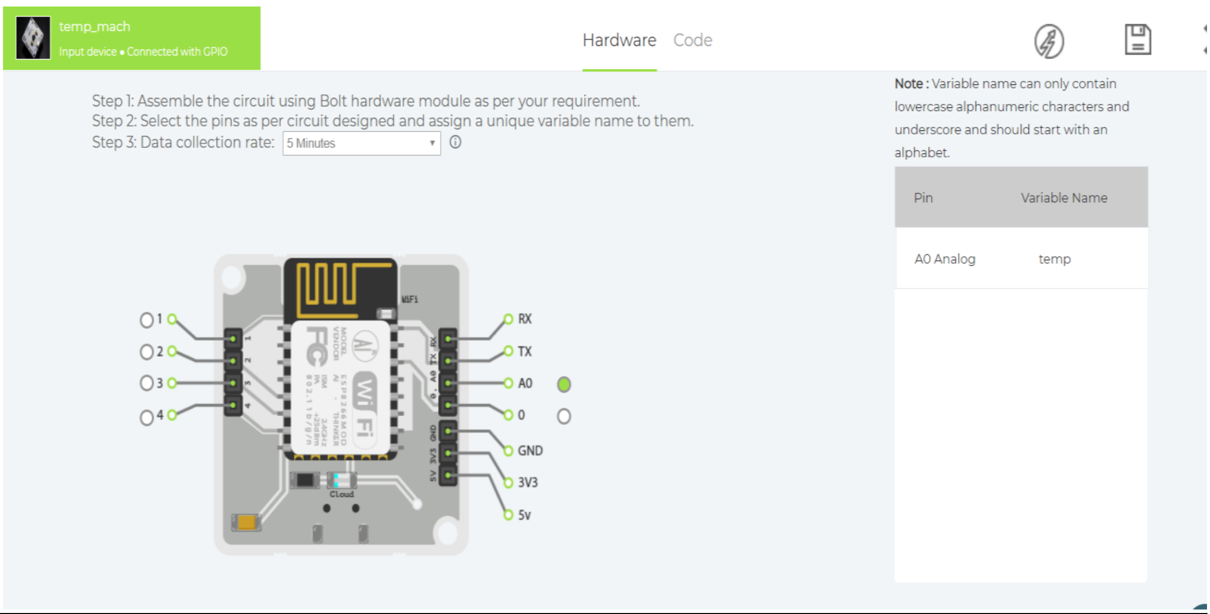
2. Create a new product for your temperature monitoring system. Products are created once and can be used for multiple Bolt devices. This ensures scalability for your IoT products you build on Bolt.



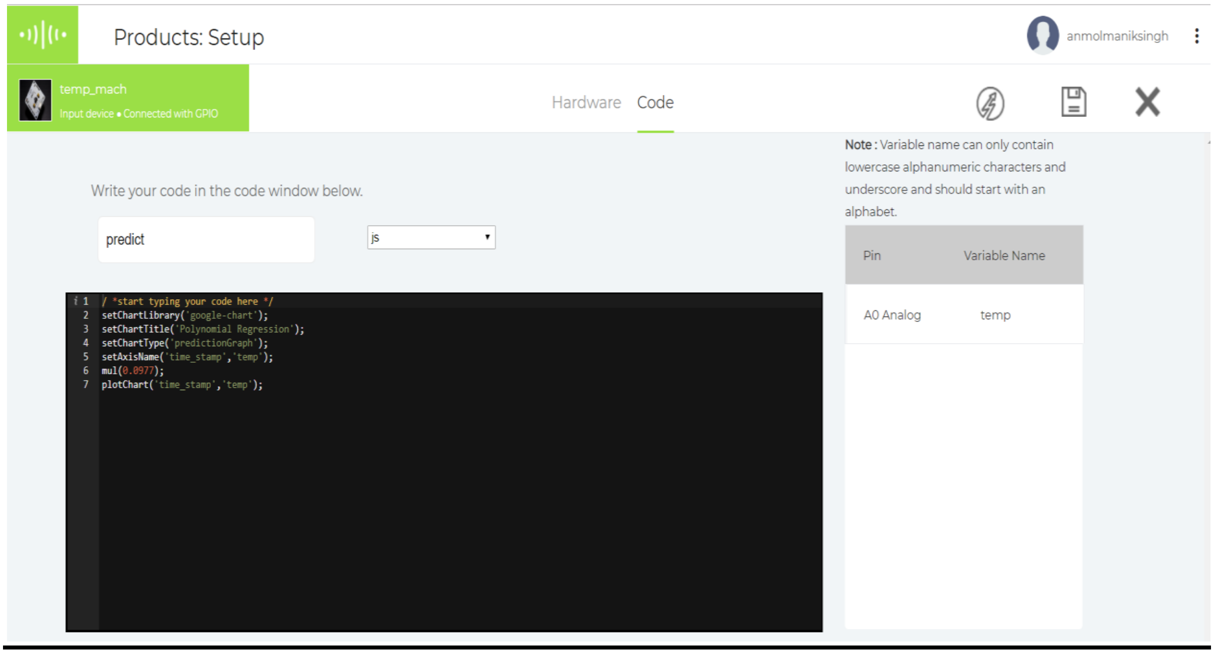
3. Click on Configure this product to configure the product. This will open a popup where you can configure your products hardware setting and write the software code.



4. Click on the "A0" pin of the Bolt and give it a name in the right side naming section. Finally, click on the "Save" icon to save your change and wait for the page to reload.



5. Now you will be presented with a variety of graphs to choose from. Choose the "Line Graph" and then click on the "Import" button.



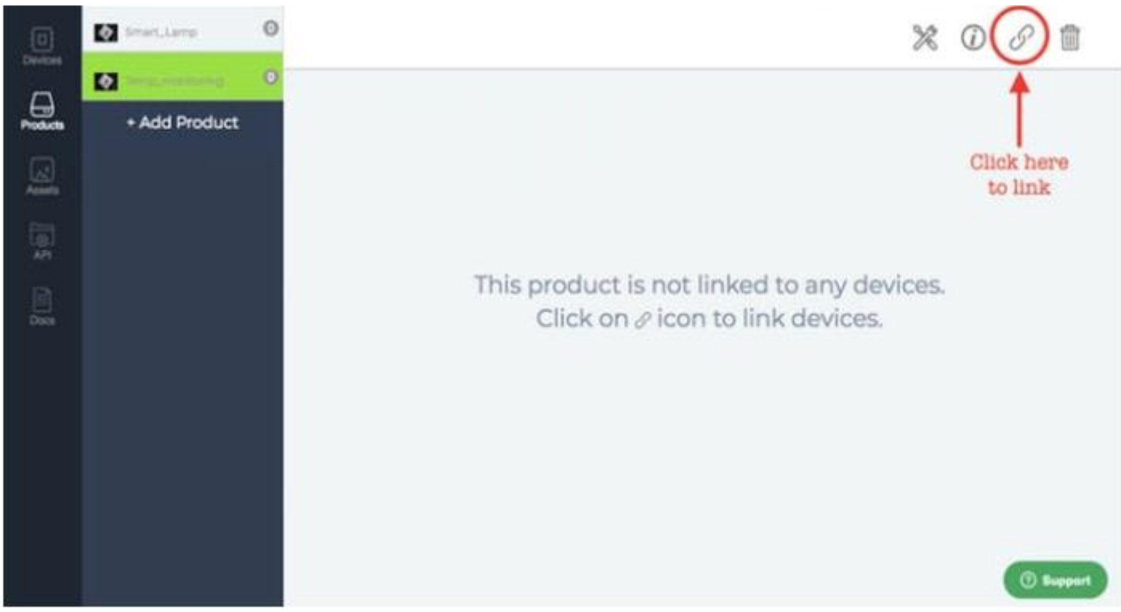
Now let me explain each line of the code so that you could make suitable changes as you wish.

- **setChartLibrary** function sets the Data Visualisation Library you would use. The most commonly used one on Bolt Cloud is the Google Library. However, you could use any other JavaScript or HTML code here to visualise the data.
- **setChartTitle** function sets the Title of the Chart/Graph. Give a suitable name for your graph here which will be shown in the heading of the page. This is different from the name of the code file.
- **setChartType** function is where you choose which type of chart you want i.e. Line Graph, Bar Graph etc.
- **setAxisName** will set the name for the X Axis and Y Axis
- **plotChart** is where you choose which variable you want to choose in your chart.

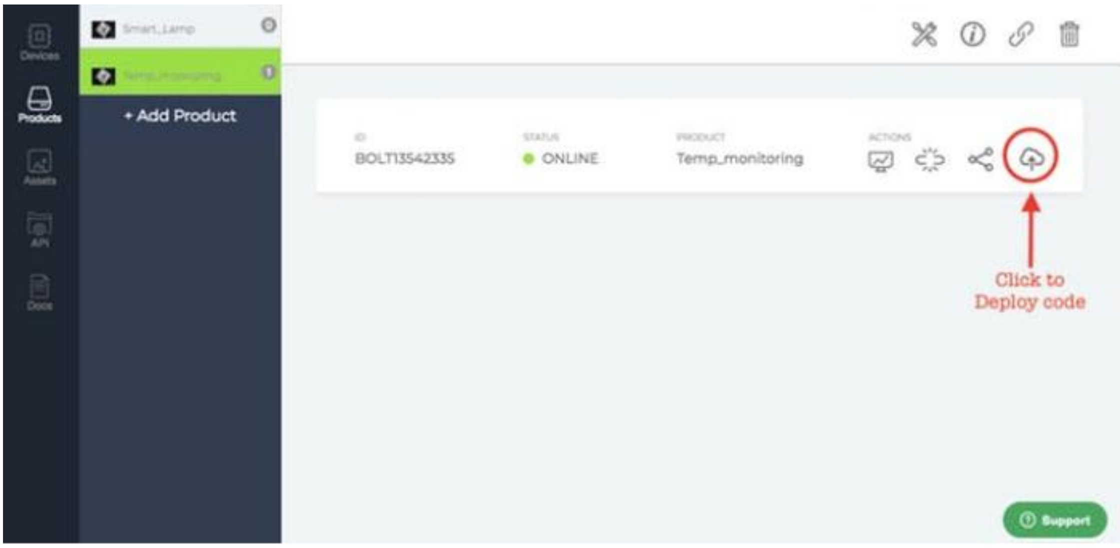
6. Next, we will need to convert the raw sensor value received to degrees. For this, we will need to multiply the raw sensor value with 0.0977. For multiplication, we use the 'mul' function. Just enter the line mul(0.0977) before the plotChart function. This will multiply the sensor value received with the multiplication factor.

```
/ *start typing your code here */
setChartLibrary('google-chart');
setChartTitle('Polynomial Regression');
setChartType('predictionGraph');
setAxisName('time_stamp','temp');
mul(0.0977);
plotChart('time_stamp','temp');
```

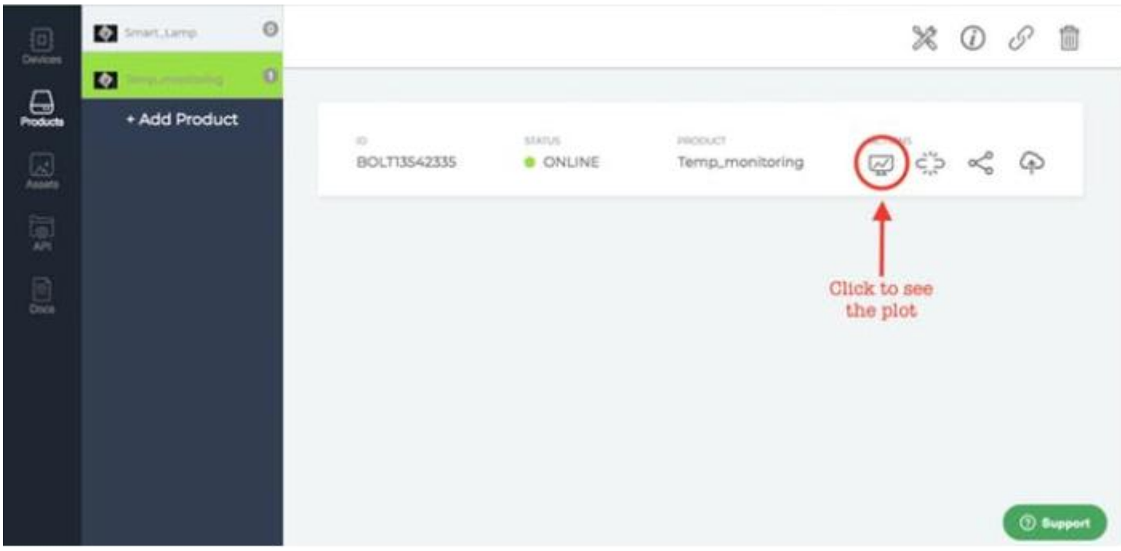
7. Now, our product configuration is ready, but we need to "link" a bolt device to the product so that the Bolt can actually start sending temperature data.



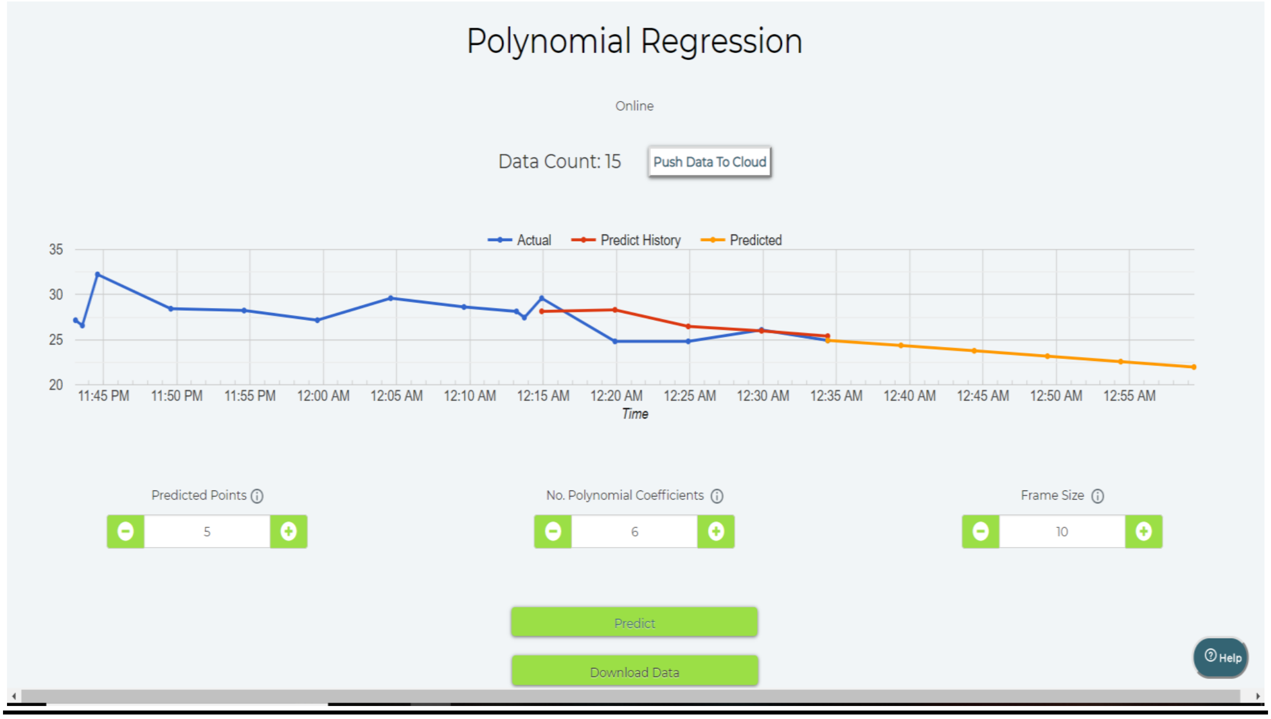
8. Now, we need to deploy the code to the Bolt device. Deploying will transfer the code and configuration to the Bolt device. Its similar to programming an hardware device like an Arduino. Except here the programming happens over the internet.



9. Click on the "Computer Monitor" icon on your Bolt device. This will open up a new page which will have a graph. Every 5 minutes, the Bolt will send a temperature reading to the Bolt Cloud.



❖ **OUTPUT PLOT .**



A. The blue line indicates the plot of actual values recorded by the sensor.

B. The orange line is the output of prediction algorithm running to determine the expected future values of temperature depending upon the past data collected by the sensor.

C. Providing 'predictionGraph' as input to the setChartType function enables the Polynomial Visualizer function for the device view.

4.5 USING THE POLYNOMIAL REGRESSION VISUALIZER.

The fields are explained below:

- Prediction points: This number tells the Visualizer how many future data points need to be predicted. By default, the Visualizer spaces the points with the data collection time in the hardware configuration of the product. So if you set the product to collect data every 5 minutes, and select 6 prediction points, the Visualizer will predict the trend and show 6 points up to 30 minutes into the future.

- No. Polynomial coefficients: Polynomial Visualizer processes the given input time-dependent data, and outputs the coefficients of the function of the form:

$$data(t) = (C_n * t^n) + (C_{n-1} * t^{n-1}) + (C_{n-2} * t^{n-2}) + \ldots\ldots + (C_1 * t^1) + C_0$$

which most closely resembles the trend in the input data. This number tells the Visualizer how many elements should be present in the function i.e. the value of n.

- Frame Size: These are the number of previous data points the Visualizer will use to predict the trend of the data. For example, if you set this value to 5, the Visualizer will use the previous 5 points to predict the trend.

When you click the 'Predict' button, the prediction history (Red line) and the next predicted trend (Yellow line) are added to the graph.

The prediction history is a graph of points the Visualizer would have predicted, at the time with the data before that point in time, using the current settings.

4.6 WRITING THE PYTHON SCRIPT FOT HANDLING EMAILS AND SMS.

Working Principle: The python code written to do the Z Score analysis first collects the data pushed to cloud by Bolt IoT Wi-Fi module. The collected data is used to find the upper threshold and lower bound of the values pushed to cloud by the sensor. Then continuously reading the value and setting the bounds and whenever the door of refrigerator is opened the reading value exceeds the upper bound and an email ALERT is send.

4.6.1 What is Z-Score analysis?

To detect an anomaly we need threshold values so that whenever the sensor reading goes above / below the threshold value then immediately the alert message is sent but how to set this threshold value so answer to this question is Z-score analysis. Z-Score analysis is the method used to find the threshold value from the past readings of the sensor.

Mean of last r values is given by:

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

Z-Score :

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

Threshold Values :

$$Tn = Vi \pm Zn$$

r = Frame Size

C = Multiplication Factor

Upper Bound on threshold: Vi + Zn   Lower Bound on threshold: Vi – Zn

So using the above formulas the upper and lower bounds were calculated from the collected sensor data and after that whenever the sensor value is out of the bounds then the alert message is send.

❖ The python script which detects the anomaly and sends notification.

```
import conf1,conf, json, time, math, statistics
from boltiot import Email, Bolt,Sms
def compute_bounds(history_data,frame_size,factor):
    if len(history_data)<frame_size :
        return None

    if len(history_data)>frame_size :
        del history_data[0:len(history_data)-frame_size]
    Mn=statistics.mean(history_data)
    Variance=0
    for data in history_data :
        Variance += math.pow((data-Mn),2)
    Zn = factor * math.sqrt(Variance / frame_size)
    High_bound = history_data[frame_size-1]+Zn
    Low_Bound = history_data[frame_size-1]-Zn
    hval=int(High_bound)
    lval=int(Low_Bound)
    print("High_bound "+str(100*(hval/1024))+" degree Celsius.")
    print("Low_bound "+str(100*(lval/1024))+" degree Celsius.")
    return [High_bound,Low_Bound]

mybolt = Bolt(conf1.api, conf1.id)
mailer=Email(conf1.mailgun_api,conf1.sandbox_url,conf1.sender,conf1.to)
mysms=Sms(conf.sid,conf.auth_token,conf.to_num,conf.from_num)
print(".........THE TEMPERATURE MONITORING SYSTEM IS RUNNING NOW........")
history_data=[]
while True:
    response = mybolt.analogRead('A0')
    data = json.loads(response)
    if data['success'] != 1:
        print("There was an error while retriving the data. ")
        print("This is the error:"+data['value'])
        time.sleep(10)
        continue
    val=int(data['value'])
    print ("This is the value send by the sensor "+str((100*val)/1024)+" degree Celsius.")
    sensor_value=0
    try:
        sensor_value = int(data['value'])
    except Exception as e:
        print("There was an error while parsing the response: ",e)
        continue

    bound = compute_bounds(history_data,conf1.frame,conf1.mul)
    l=len(history_data)
```

```
l=len(history_data)
if not bound:
    print(l)
    required_data_count=conf1.frame-len(history_data)
    print("Not enough data to compute Z-score. Need ",required_data_count," more data points")
    history_data.append(int(data['value']))
    time.sleep(10)
    continue

try:
    if sensor_value > bound[0] :

        print ("The temperature increased anomaly detected. Sending an email to registered email id.")
        response=mailer.send_email("ALERT","The temperature increased from the expected value,
            the current value is " +str(100*(sensor_value/1024))+" degree Celsius.")
        response_text=json.loads(response.text)
        print("This is the response ",response_text['message'])


except Exception as e:
    print ("Error",e)
try:
    if sensor_value > bound[0] :

        print ("The temperature increased anomaly detected. Sending SMS to registered number.")
        response=mysms.send_sms("the current temperature is "+str(100*(sensor_value/1024))+" degree Celsius.")
        print("status of SMS at twilio is : "+str(response.status))


except Exception as e:
    print ("Error",e)
time.sleep(10)
```

4.6.2 EXPLANATION OF THE CODE :

I. **Compute_bounds ():** This method is used to calculate the lower and upper bounds on the temperature value.

II. **Mn = statistics.mean()** Calculates the value of mean of last frame size number of values.

a) Sending SMS

i. **from boltiot import Sms:** Import the Sms Class from the boltiot library.

ii. **sms = Sms (sid, auth_token, to_number, from_number)** Create a sms object and pass all the credentials as a parameter.

iii. **sms.send_sms("Write your message here.")** Message content to be sent in the SMS.

SAMPLE CODE :

```
from boltiot import Sms # Import SMS class from boltiot
library
# Credentials required to send SMS
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'
sms = Sms(SID, AUTH_TOKEN, TO_NUMBER, FROM_NUMBER) # Create
object to send SMS
sms.send_sms("Write your message here.") # Call function to
send SMS!'
```

b) Sending EMAIL

i. **from boltiot import Email** Import the Email Class from the boltiot library.

ii. **mailer = email(mailgun_api_key, sandbox_url, sender_email, recipient_email)** Create a mailer object and pass all the credentials as a parameter.

iii. **mailer.send_email("Subject", "Email Body Content")** Call the send_email function to send the email.

SAMPLE CODE :

```
from boltiot import Email # Import Email class from boltiot
library
# Credentials required to send Email
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'
mailer = Email(MAILGUN_API_KEY, SANDBOX_URL, SENDER_EMAIL,
RECIPIENT_EMAIL) # Create object to send Email
mailer.send_email("Subject", "Email Body Content") # Call
function to send Email
```

c) Read Analog Input.

Command : analogRead

Parameters :

- *pin*: It is the Bolt GPIO you want to control. It can take value A0.
- *deviceName*: The name of the Bolt device you want to control.
- browser output for 0 volts:
  {"success":"1","value":"0"} or
- browser output for 2.5 volts:
  {"success": "1", "value": "245"}

d) Converting the raw sensor value into degree Celsius.

- By multiplying the sensor value by 100 and dividing the result by 1024 that is multiplying the sensor value by 0.0976 converts the raw sensor value into degree Celsius.

4.6.3 OUTPUT AFTER EXECUTION OF THE PYTHON SCRIPT :

➢ *Runnig the program:*

```
anmol@anmol-VirtualBox:~/Desktop$ python3 project.py
.........THE TEMPERATURE MONITORING SYSTEM IS RUNNING NOW........
This is the value send by the sensor 27.24609375 degree Celsius.
0
Not enough data to compute Z-score. Need  10  more data points
This is the value send by the sensor 27.24609375 degree Celsius.
1
Not enough data to compute Z-score. Need  9  more data points
This is the value send by the sensor 27.34375 degree Celsius.
2
Not enough data to compute Z-score. Need  8  more data points
This is the value send by the sensor 27.44140625 degree Celsius.
3
Not enough data to compute Z-score. Need  7  more data points
This is the value send by the sensor 27.44140625 degree Celsius.
4
Not enough data to compute Z-score. Need  6  more data points
This is the value send by the sensor 27.44140625 degree Celsius.
5
Not enough data to compute Z-score. Need  5  more data points
This is the value send by the sensor 27.5390625 degree Celsius.
6
Not enough data to compute Z-score. Need  4  more data points
This is the value send by the sensor 27.5390625 degree Celsius.
7
Not enough data to compute Z-score. Need  3  more data points
```

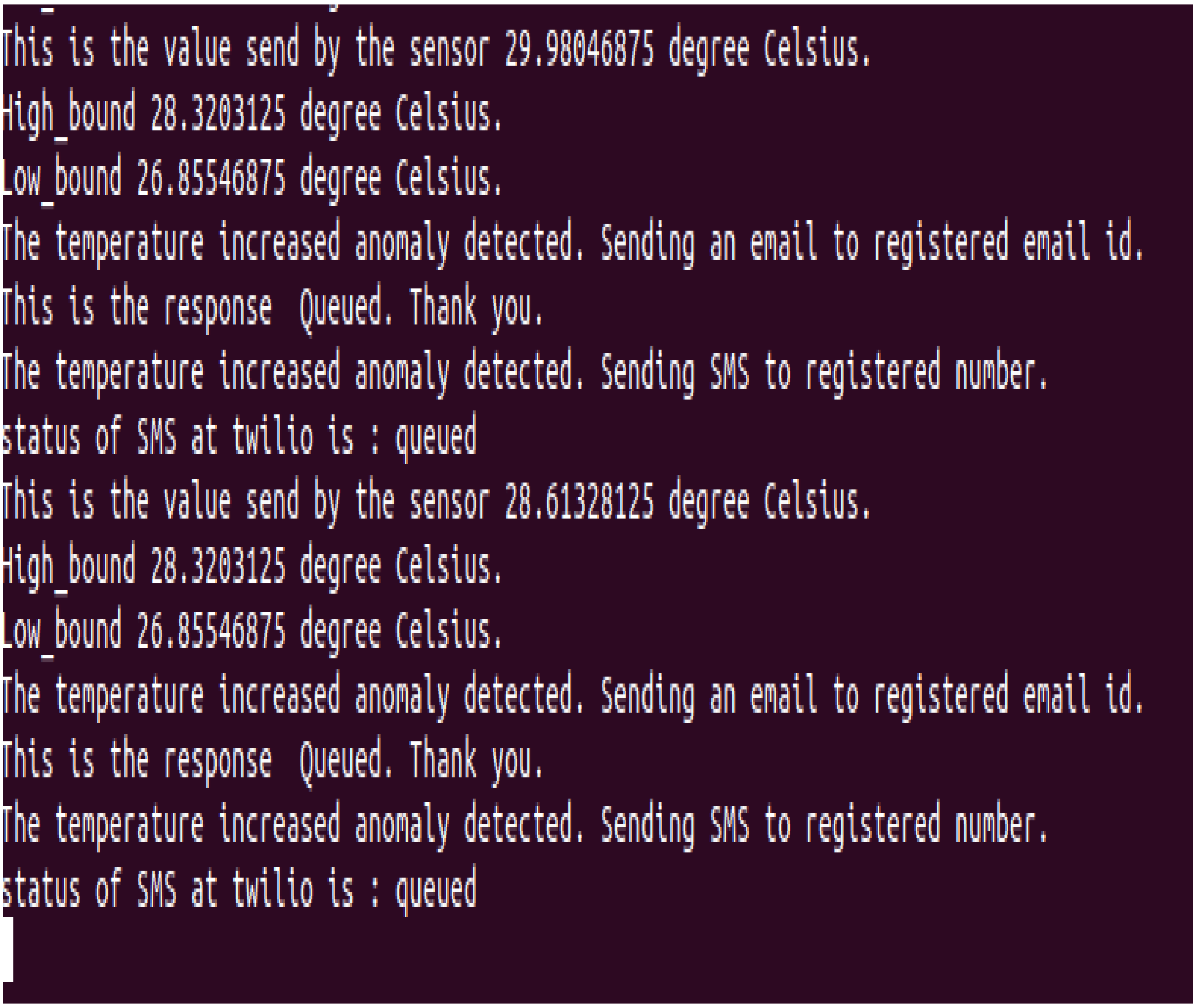
- In this snapshot we can see that the the sensor values are recorded and is seen that the threshold is still not calculated because number of readings is less than the frame size .
- When the number of past readings become equal to frame size then immediately the threshold are calculated from the past frame size number of readings.

➢ *Calculated Thresholds:*

```
Not enough data to compute Z-score. Need  3  more data points
This is the value send by the sensor 27.5390625 degree Celsius.
8
Not enough data to compute Z-score. Need  2  more data points
This is the value send by the sensor 27.63671875 degree Celsius.
9
Not enough data to compute Z-score. Need  1  more data points
This is the value send by the sensor 27.63671875 degree Celsius.
High_bound 28.3203125 degree Celsius.
Low_bound 26.85546875 degree Celsius.
This is the value send by the sensor 27.734375 degree Celsius.
High_bound 28.3203125 degree Celsius.
Low_bound 26.85546875 degree Celsius.
This is the value send by the sensor 27.63671875 degree Celsius.
High_bound 28.3203125 degree Celsius.
Low_bound 26.85546875 degree Celsius.
```

- In this snapshot we can see that the thresholds are calculated from the previous frame size number of values.
- Upper Bound : 28.32℃.
- Lower Bound :   26.85℃..

➢ *Detecting Anomaly:*



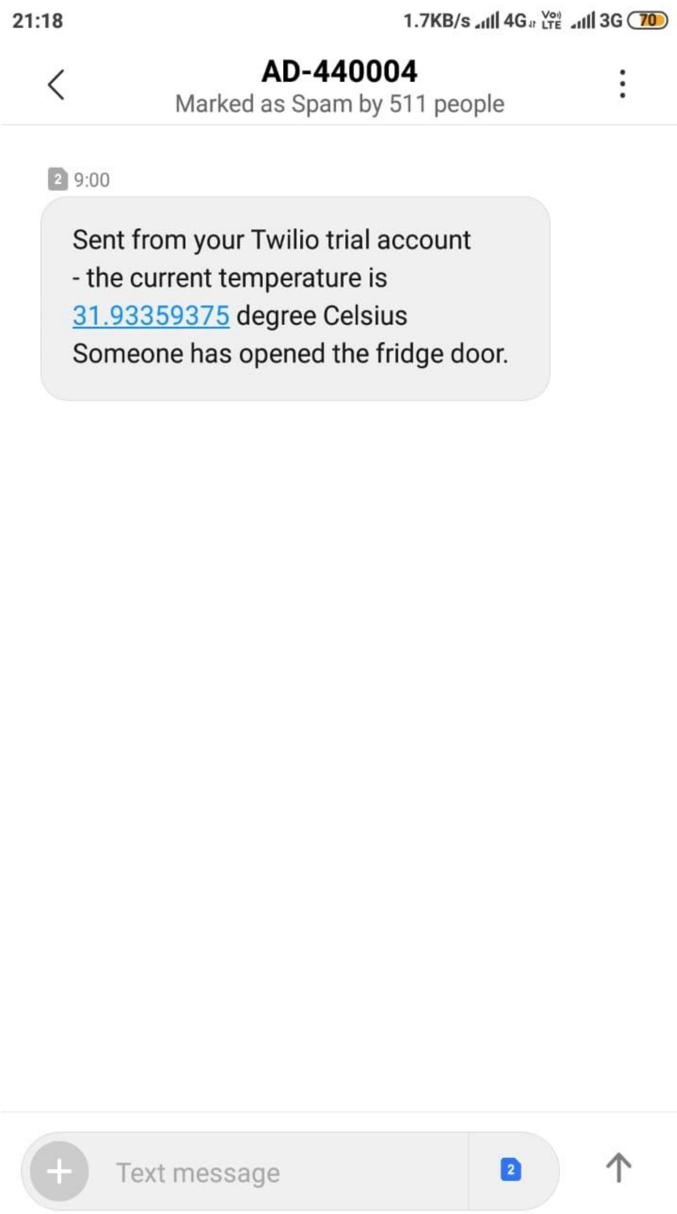
```
This is the value send by the sensor 29.98046875 degree Celsius.
High_bound 28.3203125 degree Celsius.
Low_bound 26.85546875 degree Celsius.
The temperature increased anomaly detected. Sending an email to registered email id.
This is the response  Queued. Thank you.
The temperature increased anomaly detected. Sending SMS to registered number.
status of SMS at twilio is : queued
This is the value send by the sensor 28.61328125 degree Celsius.
High_bound 28.3203125 degree Celsius.
Low_bound 26.85546875 degree Celsius.
The temperature increased anomaly detected. Sending an email to registered email id.
This is the response  Queued. Thank you.
The temperature increased anomaly detected. Sending SMS to registered number.
status of SMS at twilio is : queued
```

- In this snapshot we can see that the ALERT message is send via EMAIL and SMS when the temperature increased from the upper bound or decreases from the lower bound.

➢ *ALERT NOTIFICATIONS.*

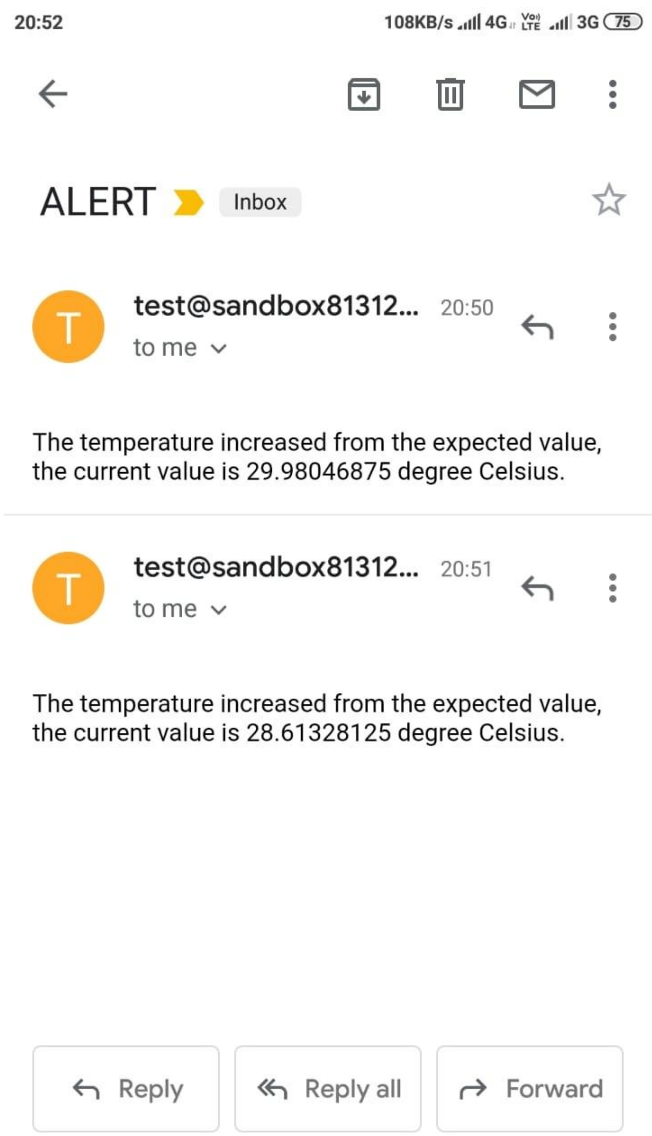
❖ *SMS ALERT*

➢ The notification send by the Twilio account about the increase in the temperature.



❖ *EMAIL ALERT*

➢ The notification send by the Mailgun account about the increase in the temperature.



CHAPTER 5 : SUMMARY

So from the results shown it can be summarized that the ‘Temperature Monitoring System’ can really assist in maintenance of Air Conditioned parts of trains. The benefit of continuously monitoring the temperature through the sensor is that whenever the temperature increases or decreases from allowable values then immediately it can be reported to the concerned authority.

Use Case 1: When any of the refrigerators in the pantry car is not working properly then immediately it can be reported and there will be no compromise with the quality of food. As soon as a refrigerator starts facing any temperature related issue it will be detected which will be reported and preventive actions can be taken as soon as possible.

Use Case 2: The air conditioners in the coaches can be remotely analyzed and controlled as per the current temperature. If the air conditioners in a coach are not working properly then immediately actions can be taken to repair them. If in some coaches there is enough cooling then the air conditioners can be managed to optimize the power consumption.

CHAPTER 6 : CONCLUSION

The ‘Temperature Monitoring System’ project concludes that IoT solutions can prove to be really helpful in various scenarios including transportation, smart homes, maintenance of machines, equipments etc. The IoT solutions continuously monitors the data retrieved from sensors and draws beneficial conclusions from that data and in return transmits control signals to control the anomaly if detected or to optimize the power consumption by the electrical appliances in case of smart homes.

The Bolt IoT device makes it easy to interface hardware and software together. The bolt cloud gathers the data from the sensors and run different algorithms as per user requirement. The machine learning algorithm ‘Linear Regression’ is used to predict the future temperature values from past collected data. The google libraries are used to plot the data collected and the line graph representation is shown in the report.

The Twilio and Mailgun Api’s controlled the Emails and SMS services. These third party Api’s did the job of sending Alert notifications whenever an anomaly is detected.

So from this project it can be concluded that IoT along with sensor networks can be used to improve the present scenario in various departments whether it is railways,
Environment related or smart homes/cities etc. In the coming time with more and more devices connected to the internet various problems will be handled remotely and more precisely by less human efforts and more artificial intelligence applications.

CHAPTER 7 : REFERENCES

WEBSITES :

- www. cloud.boltiot.com
- www.pythonprogramming.net
- www.stackoverflow.com
- www.twilio.com
- www.mailgun.com