**Sending an SMS when Temperature Crosses Threshold**

we will write a code which will fetch the temperature data collected by Bolt and send SMS if the temperature value crosses a certain threshold.

**Step 1:** Connect the temperature monitoring circuit as we have done in the previous lesson -Hardware connections for temperature monitor.

**Step 2:** Login into the putty by entering the IP address of your digital ocean droplet.

**Step 3:** After successful login, create a file named conf.py which will store all the credentials related to Twilio. To create a new file type sudo nano conf.py in the putty. After that write below code to save all the credentials in a single file.

SSID = 'You can find SSID 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'  
API\_KEY = 'This is your Bolt Cloud accout API key'  
DEVICE\_ID = 'This is the ID of your Bolt device'

**Note:** You have to replace all the above value with your credentials. You can find the first four value in Twilio dashboard and the last two in Bolt Cloud dashboard.

We store all the credentials in a separate file since it is sensitive data which should not be shared with anyone. Hence it is a good practice to avoid using credentials in code directly. After replacing all the values, save the file using CTRL+X.

**Step 4:** Now create one more file named temp\_sms.py. To do so you have to type sudo nano temp\_sms.py in the terminal. Now we will write main code to collect the data from the Bolt and send SMS if it crosses the threshold.

* We have to import our conf file which has all the credentials, json and time.

import conf, json, time

* Now we will import our Bolt python library which will let us fetch the data stored in Bolt Cloud and then based on value send SMS. To do so write

from boltiot import Sms, Bolt

In the above code, we are importing 2 things. First one is SMS which will be used to send alerts and the other one is Bolt which will be used to fetch the temp. data.

* Now we will initialize two variables which will store min. and max. threshold value. You can initialize any min. and max. limits to them.

minimum\_limit = 300

maximum\_limit = 600

* Now to fetch the data from Bolt Cloud, we will create an object of the same.

mybolt = Bolt(conf.API\_KEY, conf.DEVICE\_ID)

The above code will automatically fetch your API key and Device ID that you have initialized in conf.py file.

* Now to send SMS, we will create an object of the same.

sms = Sms(conf.SSID, conf.AUTH\_TOKEN, conf.TO\_NUMBER, conf.FROM\_NUMBER)

The above code will automatically fetch your SSID, AUTH\_TOKEN, TO\_NUMBER and FROM\_NUMBER that you have initialized in conf.py file. Make sure you have passed correct value in conf.py file.

* Now we will continuosly fetch the temperature value using `analogRead`. Then we will compare the value with our threshold, if it didn't fall in the range then SMS will be sent.

while True:

response = mybolt.analogRead('A0')

data = json.loads(response)

print (data['value'])

try:

sensor\_value = int(data['value'])

print (sensor\_value)

if sensor\_value > maximum\_limit or sensor\_value < minimum\_limit:

response = sms.send\_sms("The Current temperature sensor value is " +str(sensor\_value))

except Exception as e:

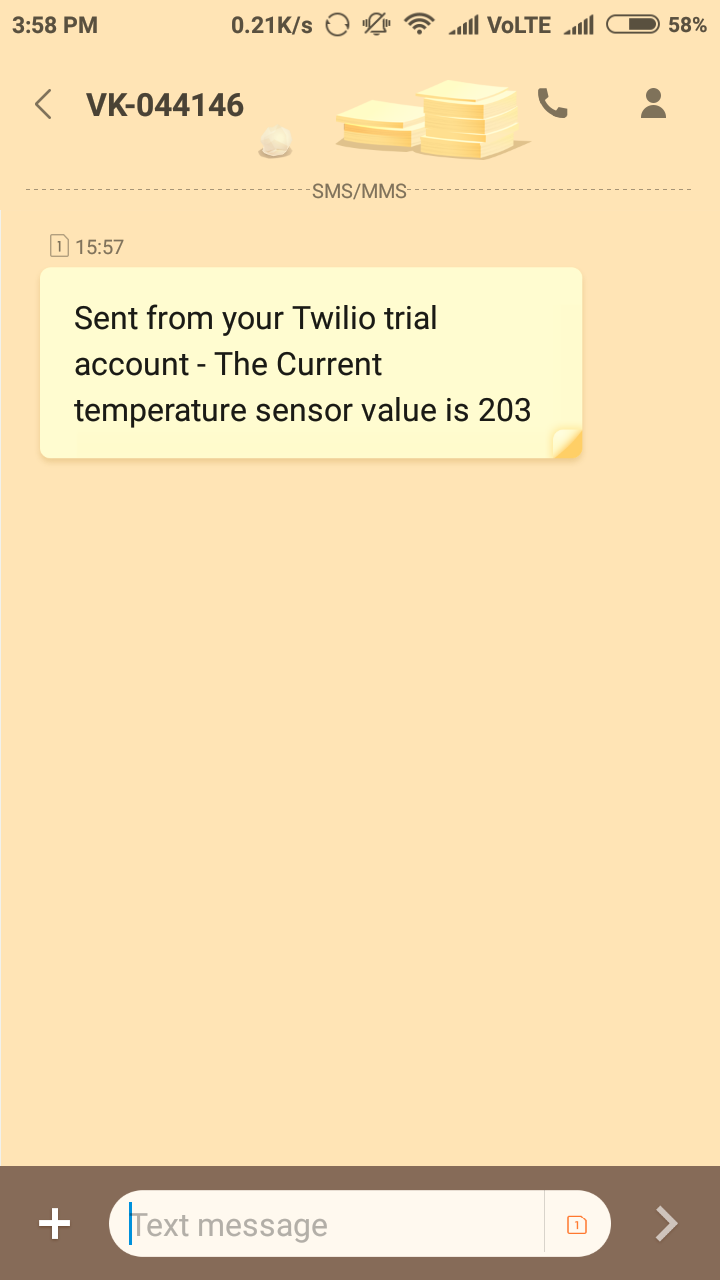
print ("Error",e)

time.sleep(10)

In the above code, we are fetching the data every 10sec. You can change the value but ideally, it should be good if the time interval between 2 data points is more than 10sec.

Below is the complete code:

The screenshot for the SMS sent is given below:



Thanks to Bolt Library which makes this lesson very easy to collect the data and send SMS