Version 3

Cloud, APIs and Alerts > Interfacing Sensor over VPS

Sending an SMS when Temperature Crosses Threshold

In the previous lesson, we learned about Twilio and how to create an account on Twilio. Now 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 Twili
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
    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:

```
import conf
from boltiot import Sms, Bolt
import json, time

minimum_limit = 300
maximum_limit = 600


mybolt = Bolt(conf.API_KEY, conf.DEVICE_ID)
sms = Sms(conf.SSID, conf.AUTH_TOKEN, conf.TO_NUMBER, conf.FROM_NUMBER)


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

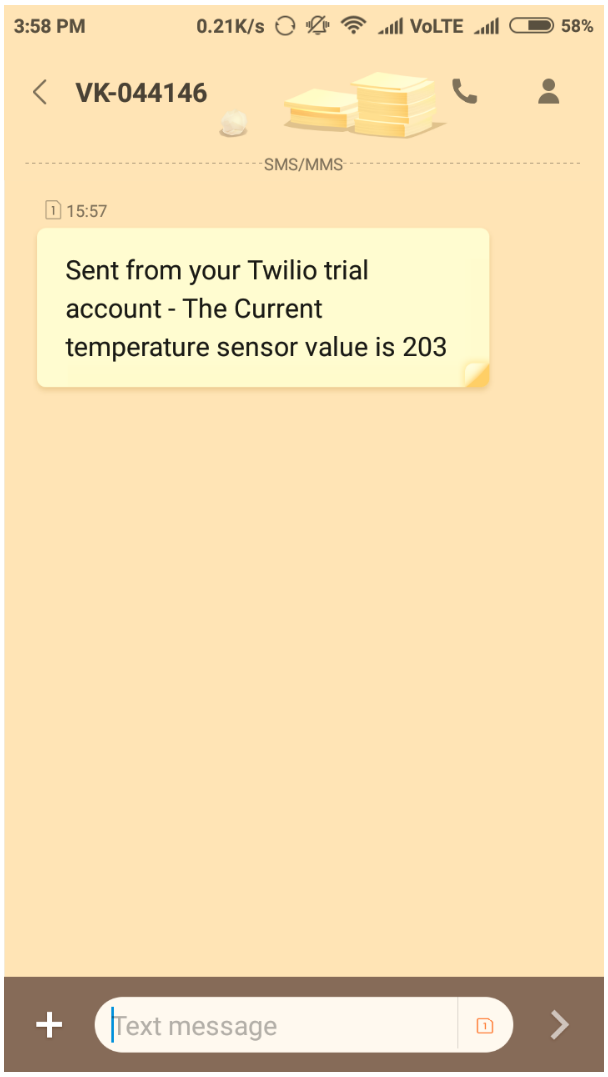
Note: The above "sensor_value" is the raw temperature reading, obtained from the LM35 sensor. In case you want to convert this value to the temperature in degree Celsius, use the formula:

```
Temperature=(100*sensor_value)/1024
```

Where sensor_value = Data obtained from the LM35 sensor. The accuracy of this value obtained can be improved using an advanced calibrated technique.

- Save the file. Time to run the code. To do so type `sudo python temp_sms.py` in terminal

Since we have written couple of print statement in the code. So the temperature data will get printed on the terminal. If that value falls outside the threshold range then SMS will be sent to your registered number. The screenshot for the SMS sent is given below:



Sent from your Twilio trial account - The Current temperature sensor value is 203

How was the lesson? Isn't it easy? Thanks to Bolt Library which makes this lesson very easy to collect the data and send SMS. In the next lesson, we will learn how we can do the same thing by sending an email instead of SMS.

PREV NEXT

Forum

Type your query here SEARCH

Q. I am not able to sign into the Twilio account, it is like I have to solve the CAPTCHA in order to sign in? What is meant by CAPTCHA?

Here is a video which will explain to you what is a CAPTCHA: https://www.youtube.com/watch?v=MWu2UiLLJI8

Q. How long does the temp_sms.py code send an sms? Does it send sms only once, or does it send sms everytime the temperature goes beyond bounds?

The code will send you an sms everytime the temperature goes beyond its bounds.
However, if you execute the command
sudo python temp_sms.py
You will need to stay logged into the digital Ocean droplet for the code to keep monitoring the temperature and send you an sms.

see more ()

Q. When I execute temp_email it is showing unexpected indent error. How do I fix this?

In python, the leading whitespace (spaces and tabs) at the beginning of a logical line is used to compute the indentation level of the line, which in turn is used to determine the grouping of statements.
The levels number of white spaces is known as an indentation level.
If you are facing unexpected indent error, it probably means that the white spaces before and after a logical grouping of instructions is different without having any function declaration or any kind of

see more ()

Q. I am getting an indentation error. How do I solve it?

This means that you have not given proper indentation i.e. spaces before the start of each line of code. Do check the code given as part of the tutorial carefully and make suitable changes. I suggest you read the course content again to be sure that there are no errors.

Q. If I have multiple input-output devices how are they to be arranged? in series or parallel?

The question of using multiple input-output devices in series or in parallel only comes in if you want to sense or control these devices using a single pin of the Bolt module.
It is not recommended that you try to sense or control multiple devices using a single pin of the Bolt unit.
Using multiple input-output devices with a single pin in the right manner is a skill-intensive task, where the method depends on the devices used and desired system behaviour.

see more ()

View more

Ask your doubts to the instructor personally if they are not already answered on forum.

Type what you want to ask the instructor

ASK THE INSTRUCTOR