Cloud, APIs and Alerts > Interfacing Sensor over VPS

Sending an Email when Temperature Crosses Threshold

In the previous lesson, we learned about Mailgun and how to create an account on Mailgun. Now we will write a code which will fetch the temperature data collected by Bolt and send an email 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 `email_conf.py` which will store all the credentials related to Mailgun. To create a new file type `sudo nano email_conf.py` in the putty. After that write below code to save all the credentials in a single file.

```
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'

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 Mailgun 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_email.py`. To do so you have to type `sudo nano temp_email.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 email_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 an email. To do so write

```
from boltiot import Email, Bolt
```

In the above code, we are importing 2 things. First one is Email which will be used to send an email 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(email_conf.API_KEY, email_conf.DEVICE_ID)
```

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

- Now to send email, we will create an object of the same.

```
mailer = Email(email_conf.MAILGUN_API_KEY, email_conf.SANDBOX_URL,
email_conf.SENDER_EMAIL, email_conf.RECIPIENT_EMAIL)
```

The above code will automatically fetch your MAILGUN_API_KEY, SANDBOX_URL, SENDER_EMAIL andRECIPIENT_EMAIL that you have initialized in `email_conf.py` file. Make sure you have passed correct value in `email_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 = mailer.send_email("Alert", "The Current
temperature sensor value is " +str(sensor_value))
    except Exception as e:
        print ("Error",e)
    time.sleep(10)
```

In the above code, the send_email function takes two parameters. First one is `Subject of the Mail` andanother one is `Message content`. Also, 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.

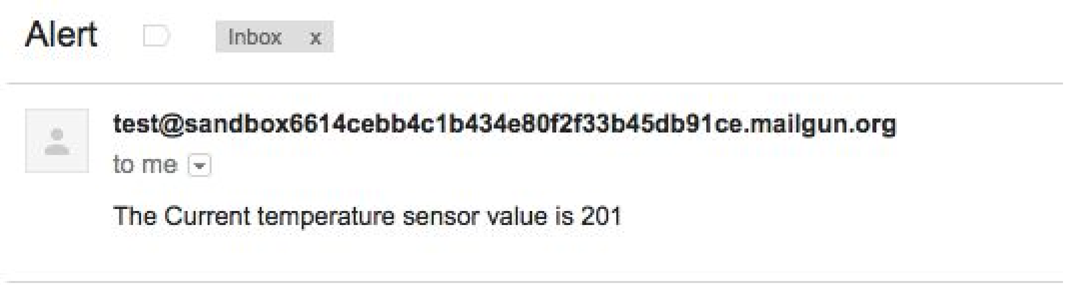
Below is the complete code:

```
import email_conf, json, time
from boltiot import Email, Bolt

minimum_limit = 300 #the minimum threshold of light value
maximum_limit = 600 #the maximum threshold of light value


mybolt = Bolt(email_conf.API_KEY, email_conf.DEVICE_ID)
mailer = Email(email_conf.MAILGUN_API_KEY, email_conf.SANDBOX_URL,
email_conf.SENDER_EMAIL, email_conf.RECIPIENT_EMAIL)


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 = mailer.send_email("Alert", "The Current
temperature sensor value is " +str(sensor_value))
    except Exception as e:
        print ("Error",e)
    time.sleep(10)
```

Note: The above "sensor_value" is the digital value, obtained from the LM35 sensor. In case you want to convert this value to 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_email.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 an email will be sent to your registered Email ID. The screenshot for the email sent is given below:



Alert

test@sandbox6614cebb4c1b434e80f2f33b45db91ce.mailgun.org

to me

The Current temperature sensor value is 201

How was the lesson? Isn't it easy? Thanks to Bolt Library which makes this lesson very easy to collect the data and send an email.