

Steps to Build the system

In order to build the system, you will need to find and note down your API key and Device ID of your Bolt device.

The API key can be found in the API section on your Cloud Dashboard.

The Device ID can be found from the Cloud Dashboard and will be something like BOLTXXXXXX where XXXXX are numbers.

Step 1:

Login to your Ubuntu Server which is either on the DigitalOcean droplet or on your VirtualMachine.

Step 2:

Create a folder where the code for the Bitcoin alert system will reside. For this, make use of the `mkdir` command to make a directory called `crypto_alert`

```
sudo mkdir crypto_alert
```

Now you will need to move to that directory. For this make use of the `cd` command to switch to that directory.

```
cd crypto_alert
```

Step 3:

Install the python libraries required to execute this project. Type and execute the following commands one after other to install the required libraries.

```
sudo apt-get update
sudo pip3 install boltiot
sudo pip3 install pyOpenSSL ndg-httpsclient pyasn1
sudo pip3 install 'requests[security]'
```

Step 4:

Now we will write a short Python script that will gather the Bitcoin price information from the internet and it will also send commands to our Bolt WiFi module when we want to be notified about the price hike.

Create a new file named `crypto_alert.py`. This is our Python script. All our code will go inside this file.

```
sudo nano crypto_alert.py
```

We will start writing the script by importing the necessary libraries and tools and also defining some constants so that we can use them in our code wherever we need.

```
import json
import time
import requests
from boltiot import Bolt

SELLING_PRICE = 1720.56
API_KEY = "XXXX"
DEVICE_ID = "XXXX"

bolt = Bolt(API_KEY, DEVICE_ID)
```

So now we have imported all the necessary libraries and we have declared a `SELLING_PRICE` variable that holds the desired selling price of bitcoin. When the price of bitcoin increases beyond selling price, we want to alert ourselves. You can change this value to set the trigger for the selling price.

We will write a function in our script to check the current price. This function will use cryptocompare (<https://www.cryptocompare.com/api/>) API to fetch the latest Bitcoin price.

```
def price_check():
    url = "https://min-api.cryptocompare.com/data/price"
    query_string = {"fsym":"BTC", "tsyms":"USD"}
    response = requests.request("GET", url, params=query_string)
    response = json.loads(response.text)
    current_price = response["USD"]
    return current_price
```

This function will visit the specified URL and will receive the current price of Bitcoin from it and then send that value back to the script.

It's time for us to now build our notification system. Now we will regularly track the price of Bitcoin every 5 seconds and if the market price is greater than our desired selling price, then, we want to trigger the buzzer connected to our Bolt WiFi module.

```
while True:
    market_price = price_check()
    print("Market price is :", market_price)
    print("Selling price is :", SELLING_PRICE)
    if market_price > SELLING_PRICE:
        bolt.digitalWrite("0", "HIGH")
        time.sleep(5)
        bolt.digitalWrite("0", "LOW")
        continue
    time.sleep(5)
```

Now save your code by pressing CTRL+X followed by letter Y. Press ENTER key to save the file.

The complete code will look something like this.

```
import json
import time
import requests
from boltiot import Bolt

SELLING_PRICE = 1720.56
API_KEY = "XXXX"      # Replace with your API key
DEVICE_ID = "XXXXX"   # Replace with your Device ID

bolt = Bolt(API_KEY, DEVICE_ID)

def price_check():
    url = "https://min-api.cryptocompare.com/data/price"
    query_string = {"fsym":"BTC", "tsyms":"USD"}
    response = requests.request("GET", url, params=query_string)
    response = json.loads(response.text)
    current_price = response["USD"]
    return current_price

while True:
    market_price = price_check()
    print("Market price is :", market_price)
    print("Selling price is :", SELLING_PRICE)
    if market_price > SELLING_PRICE:
        bolt.digitalWrite("0", "HIGH")
        time.sleep(5)
        bolt.digitalWrite("0", "LOW")
        continue
    time.sleep(5)
```

Step 5:

You can now run the program by the command,

```
sudo python3 crypto_alert.py
```

```
root@boltiot-learning-node:~# sudo python3 crypto_alert.py
Market price is : 7572.22
Selling price is : 1720.56
Market price is : 7572.22
Selling price is : 1720.56
```

[PREV](#)[NEXT](#)[Forum](#)[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=MWu2UiLLJl8>

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
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