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 BOLTXXXXX where XXXXX are numbers.

Step 1:

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

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 pyasnl

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.

Step 5:

You can now run the program by the command,

sudo python3 crypto\_alert.py