Product Learning Center About us [SIGN IN](https://cloud.boltiot.com/login)

[Siddharth Bhandarkar](https://www.boltiot.com/projects/author/siddharth-bhandarkar)

Siddharth is a student of the Bolt IoT and ML training, and...

Dec 11, 2020 10 minute read

Real time Noise Pollution Alert

**NOISE POLLUTION MONITORING**

**Story:**

A calculative amalgamation of available sound sensor and knowledge of

Python programming with regard to have a real time surveillance on the

random and disturbing sound intensities around the **No Noise and/or**

**Low Noise Areas** in a region e.g. Hospitals, Schools, Sensitive areas in a

region, Research labs, etc. and sending a real time alert to the concerned

Police o몭 cials about any disturbing noise created in any of the places or

areas above mentioned via Telegram Messaging App.

**Things used in this project:**

**Hardware Components:**

1. 

2. [Sound Detection Sensor Module.](https://www.electronicscomp.com/sound-detection-sensor-module-for-intelligent-vehicle-arduino-compatible?gclid=CjwKCAiAq8f-BRBtEiwAGr3DgRBHrSWEoo0GqSgV-zb4DzxsLVHAWmIuMyDo5DG1KMSAe7XruVcpehoCAngQAvD_BwE)

**Software Apps and Online Services:**

1. Telegram Messaging App.

**Hardware setup:**

Step 1]Connect the sound sensor with the Bolt WiFi module as per the following:

1. **A0 Pin** (Bolt IoT Wi-Fi Module- GPIO) is connected to **A0 Pin** (Sound Sensor).
2. **GND Pin** (Bolt IoT Wi-Fi Module) is connected to **GND Pin** (Sound Sensor).
3. **5V Pin** (Bolt IoT Wi-Fi Module) is connected to **“+” Pin** (Sound Sensor).

Step 2] Power up the Bolt WiFi module using the USB cable.

**Software setup:**

**Creating a Telegram Channel:**

Open your Telegram App and swipe left open the Main Menu and tap on the



feature **New Channel.**

Give a suitable name to the channel and provide description to it.



Set the channel as Public Channel and provide a permanent link to it by giving



a suitable name. **Note:** Space between words is not allowed.

Now we need to create a Telegram Bot.



**Creating a Telegram Bot:**

After creating the Telegram Channel come back to the home page of the



Telegram and search for the **BotFather** in the search menu.

After opening the BotFather enter the command **/start** to create the bot.



To create a new bot enter the command **/newbot** and give a suitable name to



the bot.

Secondly we need to give the username to the bot created it should be ending



with the bot e.g. **usernamebot**

Now a API Key will get generated store that API Key as it have to be used in the



coding.

**Linking the Telegram Channel and Telegram Bot:**

Now that we have created both Telegram Channel and Telegram Bot next we



need to link them both together.

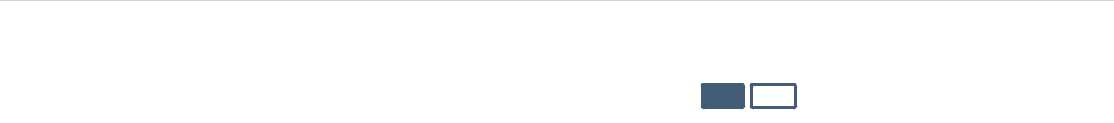
To do that open your channel and tap on the channel name and then click on



the Administrator button and search for the bot name you have created.

**Coding:**

To make and send alerting message facility via Telegram, we need to understand



This website stores cookies on your computer. These cookies are used to collect information about how you interact with our website and allow us to

the skeleton of the coding. The whole program has two parts namely:

remember you. We use this information in order to improve and customize your browsing experience and for analytics and metrics about our visitors

both on this website and other media. To find out more about the cookies we use, see our Privacy Policy.

|  |  |  |  |
| --- | --- | --- | --- |
| If you decline,1. | **Con**yourinformation**몭 guration**won’tbe**code:**tracked whenItconsistyouvisitthisallwebsitethe. Abackendsinglecookiedetailswillbeusedofin yourBoltbrowserIoT toWiremember-FiModuleyourpreference | | |
| not to be tracked. | |  |  |
|  | and the Telegram. |  |  |
| 2. | **Main code:** It consists of the core coding of the facility. | Accept | Decline |

To create above two mention 몭 les, executive the following steps:

Open the Digital Ocean Ubuntu server (For Windows /Mac Operating System) or

open the terminal in the Ubuntu Operating System.

Step 1] We need to create a directory (folder) to store the two coding 몭 les

mentioned above. To create a directory named alert, type out the following

command

sudo mkdir alert

Step 2] Next we have to enter the directory that we just created. To enter the

directory named alert type out the following command:

cd alert

Step 3] After entering the folder 몭 rst we need to create the con몭 guration python

몭 le which will hold all the backend details of the Bolt IoT Wi-Fi Module and

Telegram.

To create the con몭 guration python 몭 le in the folder which was created 몭 rst type

out the following command for creating the 몭 le with extension .py;

sudo nano configuration.py

Step 4] After the above mentioned 몭 le enter the following data into the 몭 le. Make

sure that you add the updated Bolt API key, device id and Telegram details:

"""Configurations for Telegram alert message"""

BOLT\_API\_KEY = "XXXXXXXXX" #This is your Bolt cloud API

Key.

DEVICE\_ID = "XXXXXXXXX" #This is the ID number of your

Bolt device.

TELEGRAM\_CHAT\_ID = "@XXXXXXXXX" #This is the channel ID of the

channel created in the

Telegram. Paste after @.

TELEGRAM\_BOT\_ID = "botXXXXXXXXX" #This is the bot ID of the bot

created in the Telegram. Paste

after bot.

THRESHOLD = 80 #Threshold beyond which the

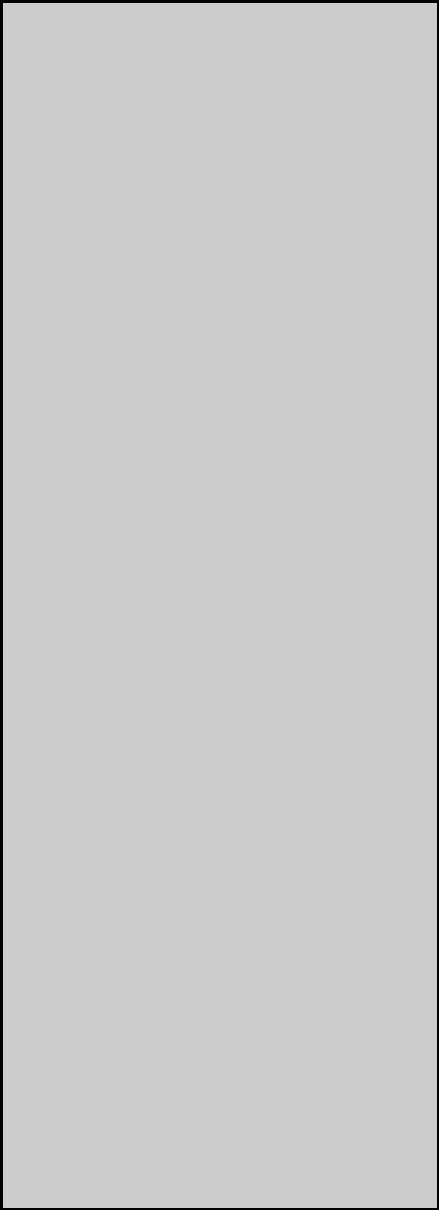
alert should be sent.

Step 5] Save the 몭 le by clicking "ctrl+x" and press enter. Next create another 몭 le

which will include the main coding of the facility.

sudo nano alert.py

Step 6] Enter the following code into the newly created 몭 le:



import requests #for making HTTP requests

import json #library for handling JSON data

import time #module for sleep operation

from boltiot import Bolt #importing Bolt from boltiot

module

import configuration #configuration file

mybolt = Bolt(configuration.BOLT\_API\_KEY,configuration.DEVICE\_ID)

def get\_sound\_sensor\_value\_from\_pin(pin):

try:

response = mybolt.analogRead(pin)

data = json.loads(response)

if data["success"] != 1:

print("Request not successful")

print("This is the response‐>", data)

return ‐999

sound\_sensor\_value = int(data["value"])

return sound\_sensor\_value

except Exception as e:

print("Something went wrong when returning the sensor value")

print(e)

return ‐999

def send\_telegram\_message(message):

url = "https://api.telegram.org/" + configuration.TELEGRAM\_BOT\_ID + "/sendMessage"

data = {

"chat\_id": configuration.TELEGRAM\_CHAT\_ID,

"text": message

}

try:

response = requests.request(

"GET",

url,

params = data

)

print("This is the Telegram response")

print(response.text)

telegram\_data = json.loads(response.text)

return telegram\_data["OK"]

except Exception as e:

print("An error occurred in sending the alert message via Telegram")

print(e)

return False

while True:

#Step 1

sound\_sensor\_value = get\_sound\_sensor\_value\_from\_pin("A0")

print("The current sensor reading is:", sound\_sensor\_value)

#Step 2

if sound\_sensor\_value == ‐999:

print("Request was unsuccessful. Skipping.")

time.sleep(10)

continue

#Step 3

if sound\_sensor\_value >= configuration.THRESHOLD:

print("Sensor value has exceeded threshold")

message = "Alert! Noise disturbance around the XYZ Hospital. Random and unidentified sound intensity has crossed " + str(configuration.THRESHOLD) + str(“dB”) + \

“The current sound sensor reading is ” + str(sound\_sensor\_value) + str(“dB”) + str(“To, The Police Incharge, Immediate action required. Thankyou.”)

telegram\_status = send\_telegram\_message(message)

print("This is the Telegram status:", telegram\_status)

# Step 4

time.sleep(50) #Time interval to get the status update.

Step 7] Exit the code editor using "ctrl+x" and then run the code that you have

written using the following command.

sudo python3 alert.py

Conclusion:

An alert message is expected on your Telegram channel. Possible output as follows

also the alert message can be customised as per the requirement:





**Want to build more such IoT and ML projects? Want to learn**

**IoT and ML from basics?**

Check out the Bolt IoT and ML training. This online video training is

excellent for those who want to start with IoT and ML because it teaches

you to build projects from the basics. Click on the button below to know

more about the training.

alert [noise](https://www.boltiot.com/projects/tag/noise)

[Siddharth](https://www.boltiot.com/projects/author/siddharth-bhandarkar)

[Bhandarkar](https://www.boltiot.com/projects/author/siddharth-bhandarkar)

Siddharth is a student of

the Bolt IoT and ML

training, and...

Subscribe to Bolt IoT to get

latest projects delivered to

your inbox

Stay up to date with the latest technologies and project tutorials

By clicking this button, you agree to our [Terms of Service](https://www.boltiot.com/termsofservice) and [Privacy Policy](https://www.boltiot.com/privacypolicy).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
|  | [IoT](https://www.boltiot.com/projects/topic/iot), [Bolt IoT](https://www.boltiot.com/projects/topic/bolt-iot), [sensor](https://www.boltiot.com/projects/topic/sensor), [alert](https://www.boltiot.com/projects/topic/alert), [diy](https://www.boltiot.com/projects/topic/diy), [Arduino](https://www.boltiot.com/projects/topic/arduino) |  | [IoT](https://www.boltiot.com/projects/topic/iot), [Bolt IoT](https://www.boltiot.com/projects/topic/bolt-iot), [sensor](https://www.boltiot.com/projects/topic/sensor), [alert](https://www.boltiot.com/projects/topic/alert), [diy](https://www.boltiot.com/projects/topic/diy), [Smart](https://www.boltiot.com/projects/topic/smart-parking) |  | [IoT](https://www.boltiot.com/projects/topic/iot), [Bolt IoT](https://www.boltiot.com/projects/topic/bolt-iot), [sensor](https://www.boltiot.com/projects/topic/sensor), [alert](https://www.boltiot.com/projects/topic/alert), [diy](https://www.boltiot.com/projects/topic/diy), [remote](https://www.boltiot.com/projects/topic/remote-control) | |
|  | [Bidirectional Visitor Counter](https://www.boltiot.com/projects/real-time-noise-pollution-alert) |  | [Parking](https://www.boltiot.com/projects/topic/smart-parking) |  | [control](https://www.boltiot.com/projects/topic/remote-control), [smart switch](https://www.boltiot.com/projects/topic/smart-switch) | |
|  | [System](https://www.boltiot.com/projects/real-time-noise-pollution-alert) |  | [Smart Parking System](https://www.boltiot.com/projects/real-time-noise-pollution-alert) |  | [Buzzer Symphony Using Bolt](https://www.boltiot.com/projects/real-time-noise-pollution-alert) | |
|  |  |  |  |  | [IoT Cloud Platform](https://www.boltiot.com/projects/real-time-noise-pollution-alert) | |
|  | [Nipun Jyoti Pathak](https://www.boltiot.com/projects/real-time-noise-pollution-alert) |  | [Sayali Borate](https://www.boltiot.com/projects/real-time-noise-pollution-alert) |  |  |  |
|  | 10 minute read |  | 10 minute read |  | [Apurva Kulpe](https://www.boltiot.com/projects/real-time-noise-pollution-alert) | |
|  |  |  |  |  | 10 minute read | |
|  |  |  |  |  |  |  |