NOISE POLLUTION MONITORING

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 officials 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. Bolt WiFi Module
2. Sound Detection Sensor Module.

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 create above two mention files, 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 files 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 first we need to create the configuration python file which will hold all the backend details of the Bolt IoT Wi-Fi Module and Telegram.

To create the configuration python file in the folder which was created first type out the following command for creating the file with extension .py;

```
sudo nano configuration.py
```

Step 4] After the above mentioned file enter the following data into the file. 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 file by clicking "ctrl+x" and press enter. Next create another file which will include the main coding of the facility.

```
sudo nano alert.py
```

Step 6] Enter the following code into the newly created file:

```
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:

ALERT! Noisy disturbance around
the XYZ Hospital. Random and
unindentified sound intensity has
crossed 80dB. The current sound
sensor reading is 292dB. To, The
Police Incharge, Immediate action
required. Thankyou.
👁 2 22:33

Sound_level_intensity
ALERT! Noisy disturbance around
the XYZ Hospital. Random and
unindentified sound intensity has
crossed 80dB. The current sound
sensor reading is 291dB. To, The
Police Incharge, Immediate action
required. Thankyou.
👁 2 22:33

October 17

Sound_level_intensity
ALERT! Noise disturbance around
the XYZ Hospital. Random and
unindentified sound intensity has
crossed 80dB. The current sound
sensor reading is 290dB. To, The
Police Incharge, Immediate action
required. Thankyou.
👁 1 22:31