code explanation:

As explained previously in the phase_2 project ,In this project we are using MQ135 Sensors. In this phase we are going to explain coding related to our project

Sensors:

System uses air sensors to sense presence of harmful gases/compounds in the air and constantly transmit this data to microcontroller. Also system keeps measuring sound level and reports it to the online server over IOT. The sensors interact with microcontroller which processes this data and transmits it over internet.

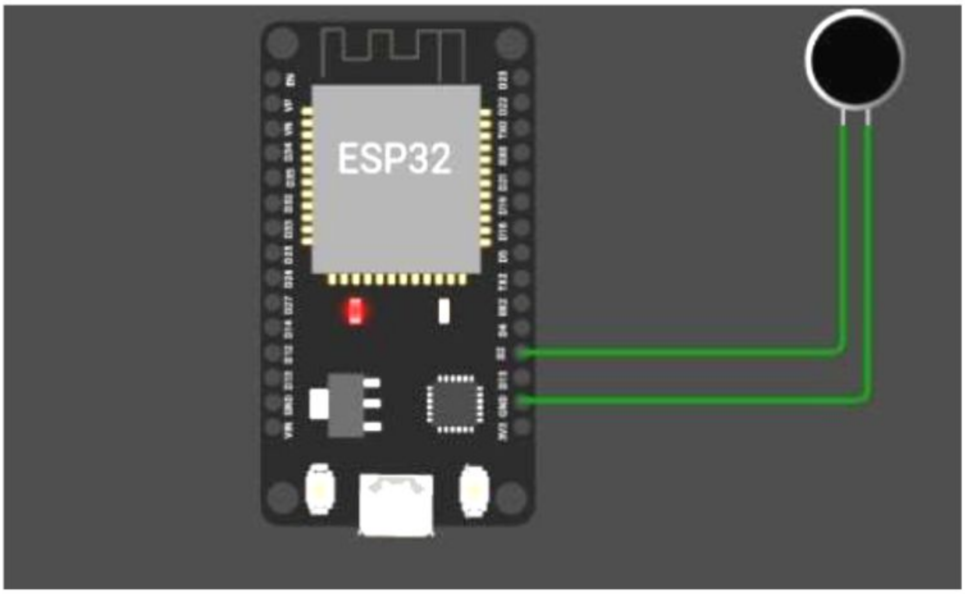
- Atmega Microcontroller
- MQ 135 Sensor
- Mic Sensor or ESP8266 Wifi module

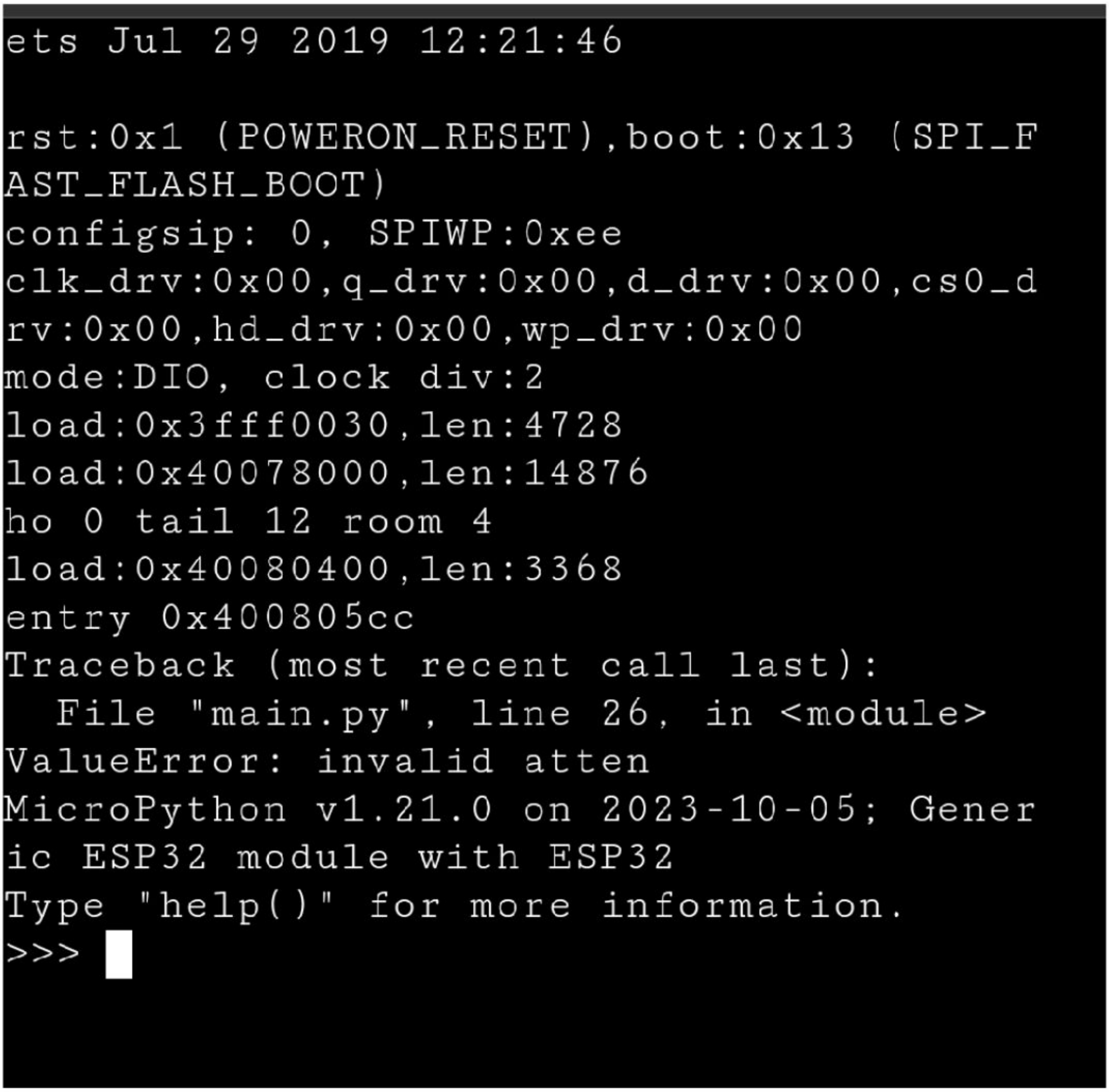
Coding:

```
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.
```

OUTPUT:





```
ets Jul 29 2019 12:21:46

rst:0x1 (POWERON_RESET),boot:0x13 (SPI_F
AST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_d
rv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:2
load:0x3fff0030,len:4728
load:0x40078000,len:14876
ho 0 tail 12 room 4
load:0x40080400,len:3368
entry 0x400805cc
Traceback (most recent call last):
  File "main.py", line 26, in <module>
ValueError: invalid atten
MicroPython v1.21.0 on 2023-10-05; Gener
ic ESP32 module with ESP32
Type "help()" for more information.
>>> 
```