DHIRAJ ZEN B K 276425@student.annauniv.edu PH : 7338885805

TASK-4

Write a code to read MPU (MPU6050 sensor) data at 200 readings per second from ESP32 and then create a JSON/CSV file of that data and push that file to the cloud endpoint using FTP/MQTT every 5 seconds.

In this task I have used the Adafruit IO as the MQTT server that receives JSON data sent using MQTT and stores the data in its feed.
In order to convert the sensor reading to JSON format and store it in a local JSON file, I have used the ArduinoJson library.
This JSON file is written at a rate of 200 reading per second. The file is read every 5 seconds and deleted after the data is sent to the Adafruit IO feed.
SPIFFS library is used for file handling.

CODE-

```
#include <Adafruit_MQTT.h>
#include <Adafruit_MQTT_Client.h>
#include <SPIFFS.h>
#include <WiFi.h>
#include <Wire.h>
#include <ArduinoJson.h>


//WiFi constants
const char* ssid = "Zen";
const char* pass = "dhirajzen2002";

//MPU constants,variables.
const uint16_t MPU = 0x68; //I2C address
float accX, accY, accZ;
float gyroX, gyroY, gyroZ;
size_t size = 6;

//MQTT broker
#define AIO_SERVER "io.adafruit.com"
#define AIO_SERVERPORT 1883
#define AIO_USERNAME "dhirajzen"
#define AIO_KEY "aio_JcoS35dZWUhFdaeijTlwmNyda9IZ"

//MQTT client
WiFiClient client;
Adafruit_MQTT_Client mqtt(&client, AIO_SERVER, AIO_SERVERPORT, AIO_USERNAME,
AIO_KEY );
Adafruit_MQTT_Publish mpu6050 = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME"/ Feeds /
mpu6050" );
//timing variables
int previousReadingMillis = 0;
int previousUploadMillis = 0;


void connectToWiFi(){
  Serial.print("Connecting to WiFi");
  WiFi.begin(ssid, pass);

  while(WiFi.status() != WL_CONNECTED){
    Serial.print(".");
    delay(100);
  }
  Serial.println("Connected to WiFi.");
}

void connectToMQTT(){
  while(!mqtt.connected()){
    Serial.println("Connecting to the MQTT broker...");
    if(mqtt.connect() == 0){
      Serial.println("Connected to the MQTT broker");
    }else{
      Serial.println("Trying to reconnect in 5 seconds...");
      mqtt.disconnect();
      delay(5000);
    }
  }
}

void readSensorData(){
  //get accelerometer reading
  Wire.beginTransmission(MPU);
  Wire.write(0x3B);
  Wire.endTransmission(false);
  //each axis value is stored in 2 registers.
  Wire.requestFrom(MPU,size, true);
  //divide the raw value with the LSB sensitivity given in the data sheet.
  accX = (Wire.read() << 8 | Wire.read())/16384.0;
  accY = (Wire.read() << 8 | Wire.read())/16384.0;
  accZ = (Wire.read() << 8 | Wire.read())/16384.0;

  //get gyroscope reading
  Wire.beginTransmission(MPU);
  Wire.write(0x43);
  Wire.endTransmission(false);
  //each axis value is stored in 2 registers.
  Wire.requestFrom(MPU,size, true);
  //divide the raw value with the LSB sensitivity given in the data sheet.
  gyroX = (Wire.read() << 8 | Wire.read())/131.0;
  gyroY = (Wire.read() << 8 | Wire.read())/131.0;
  gyroZ = (Wire.read() << 8 | Wire.read())/131.0;

  //create a string to store json data
  String jsonData;
  StaticJsonDocument<200> doc;

  doc["Accelerometer (X,Y,Z)"][0] = accX;
  doc["Accelerometer (X,Y,Z)"][1] = accX;
  doc["Accelerometer (X,Y,Z)"][2] = accX;

  doc["Gyroscope (X,Y,Z)"][0] = accX;
  doc["Gyroscope (X,Y,Z)"][1] = accX;
  doc["Gyroscope (X,Y,Z)"][2] = accX;

  serializeJson(doc, jsonData);

  //we will write this json data into a json file
  File jsonFile = SPIFFS.open("/mpu6050.json", FILE_WRITE);
  if(jsonFile){
    jsonFile.println(jsonData);
    jsonFile.close();
  }else{
    Serial.println("Failed to create json file.");
  }

}

void sendSensorData(){
  File jsonFile = SPIFFS.open("/mpu6050.json", FILE_READ);
  if(jsonFile){
    size_t fileSize = jsonFile.size();
    char* fileData = new char[fileSize];
    jsonFile.readBytes(fileData, fileSize);
    mpu6050.publish(fileData);
    delete [] fileData;
    jsonFile.close();
    SPIFFS.remove("/mpu6050.json");
  }else{
    Serial.println("Failed to open json file.");
  }
}

void setup(){
  Serial.begin(9600);
  connectToWiFi();
  connectToMQTT();
  SPIFFS.begin();

  //reset the mpu6050 registers
  Wire.begin();
  Wire.beginTransmission(MPU);
  Wire.write(0x6B);
  Wire.write(0x00);
  Wire.endTransmission(true);
```

```
}


void loop() {
  unsigned long currentMillis = millis();

  // Read sensor data every 1000/200 milliseconds
  if (currentMillis - previousReadingMillis >= 5) {
    previousReadingMillis = currentMillis;
    readSensorData();
  }

  // Upload JSON file to MQTT every 5 seconds
  if (currentMillis - previousUploadMillis >= 5000) {
    previousUploadMillis = currentMillis;
    sendSensorData();
  }

  // Maintain MQTT connection
  if (!mqtt.connected()) {
    Serial.println("Reconnecting to MQTT...");
    connectToMQTT();
  }

}
```