#include <WiFi.h>

#include <HTTPClient.h>

#include <Wire.h>

#include <SparkFunBME280.h>

#include <MAX30105.h>

#include <spo2\_algorithm.h>

// Replace with your network credentials

const char\* ssid = "your\_SSID";

const char\* password = "your\_PASSWORD";

// Replace with your ThingSpeak API key

String apiKey = "your\_API\_KEY";

// Replace with your ThingSpeak channel ID

String channelID = "your\_CHANNEL\_ID";

// Define BME280 and MAX30105 objects

BME280 bme;

MAX30105 particleSensor;

// Define variables for health sensor data

float temperature;

float humidity;

float pressure;

int heartRate;

int spo2;

// Define variables for ThingSpeak field IDs

const int fieldTemperature = 1;

const int fieldHumidity = 2;

const int fieldPressure = 3;

const int fieldHeartRate = 4;

const int fieldSPO2 = 5;

void setup() {

Serial.begin(115200);

// Connect to Wi-Fi

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) {

delay(1000);

Serial.println("Connecting to WiFi...");

}

Serial.println("Connected to WiFi");

// Initialize BME280 and MAX30105

bme.begin(0x76);

particleSensor.begin(Wire, I2C\_SPEED\_FAST);

// Configure MAX30105

particleSensor.setup();

particleSensor.setPulseAmplitudeRed(0x0A);

particleSensor.setPulseAmplitudeGreen(0);

particleSensor.setMode(MODE\_SPO2);

// Send data to ThingSpeak

sendToThingSpeak();

}

void loop() {

// Read health sensor data

temperature = bme.readTempC();

humidity = bme.readFloatHumidity();

pressure = bme.readFloatPressure() / 100.0;

// Read heart rate and SPO2

particleSensor.check();

if (particleSensor.getHR() > 50 && particleSensor.getHR() < 200) {

heartRate = particleSensor.getHR();

spo2 = getSpO2();

}

// Send data to ThingSpeak every 15 seconds

delay(15000);

sendToThingSpeak();

}

void sendToThingSpeak() {

HTTPClient http;

String url = "http://api.thingspeak.com/update?api\_key=" + apiKey;

url += "&field" + String(fieldTemperature) + "=" + String(temperature);

url += "&field" + String(fieldHumidity) + "=" + String(humidity);

url += "&field" + String(fieldPressure) + "=" + String(pressure);

url += "&field" + String(fieldHeartRate) + "=" + String(heartRate);

url += "&field" + String(fieldSPO2) + "=" + String(spo2);

Serial.println("Sending data to ThingSpeak...");

Serial.println(url);

http.begin(url);

int httpCode = http.GET();

if (httpCode > 0) {

String response = http.getString();

Serial.println("HTTP response code: " + String(httpCode));

Serial.println("HTTP response body: " + response);

} else {

Serial.println("HTTP request failed");

}

http.end();

}

int getSpO2() {

int irValue = particleSensor.getIR();

int redValue = particleSensor.getRed();

// Calculate SpO2 using