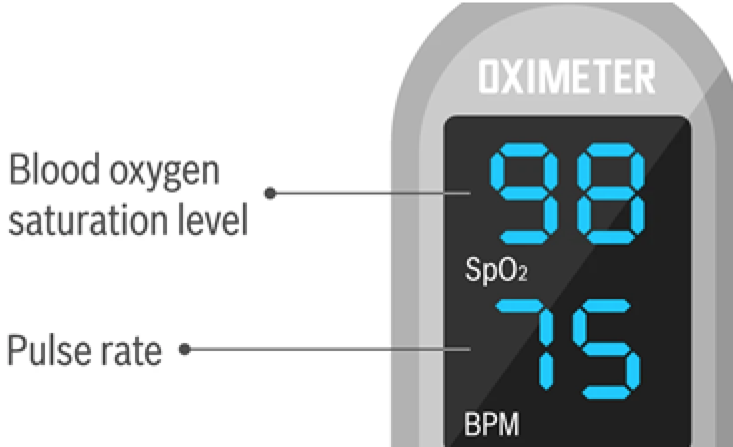
https://wiki.dfrobot.com/Heart_Rate_and_Oximeter_Sensor_V2_SKU_SEN0344#
https://github.com/DFRobot/DFRobot_BloodOxygen_S

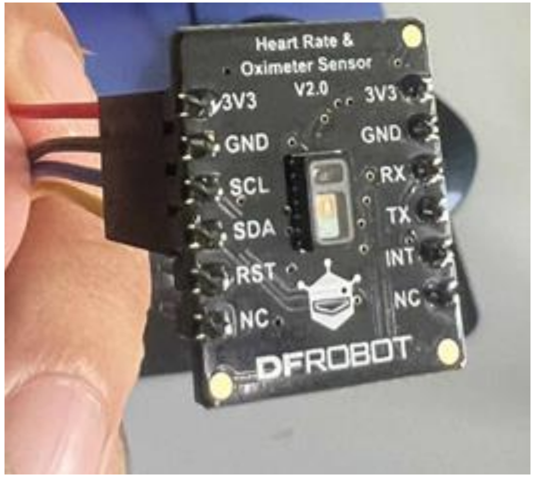
# **Heart Rate** and **Oximeter** Sensor V2 (SKU SEN0344) by DFRobot a **PPG-based sensor**

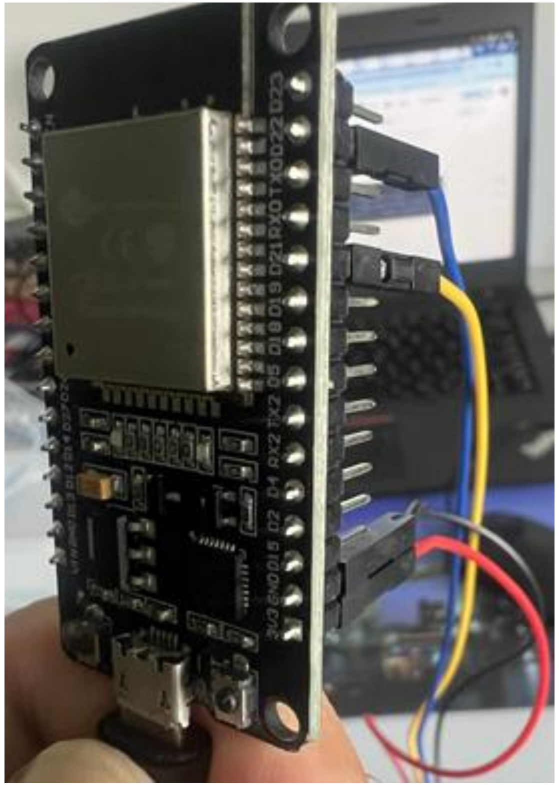


ESP32 MC
C++
Arduino IDE
Telegram (wifi)

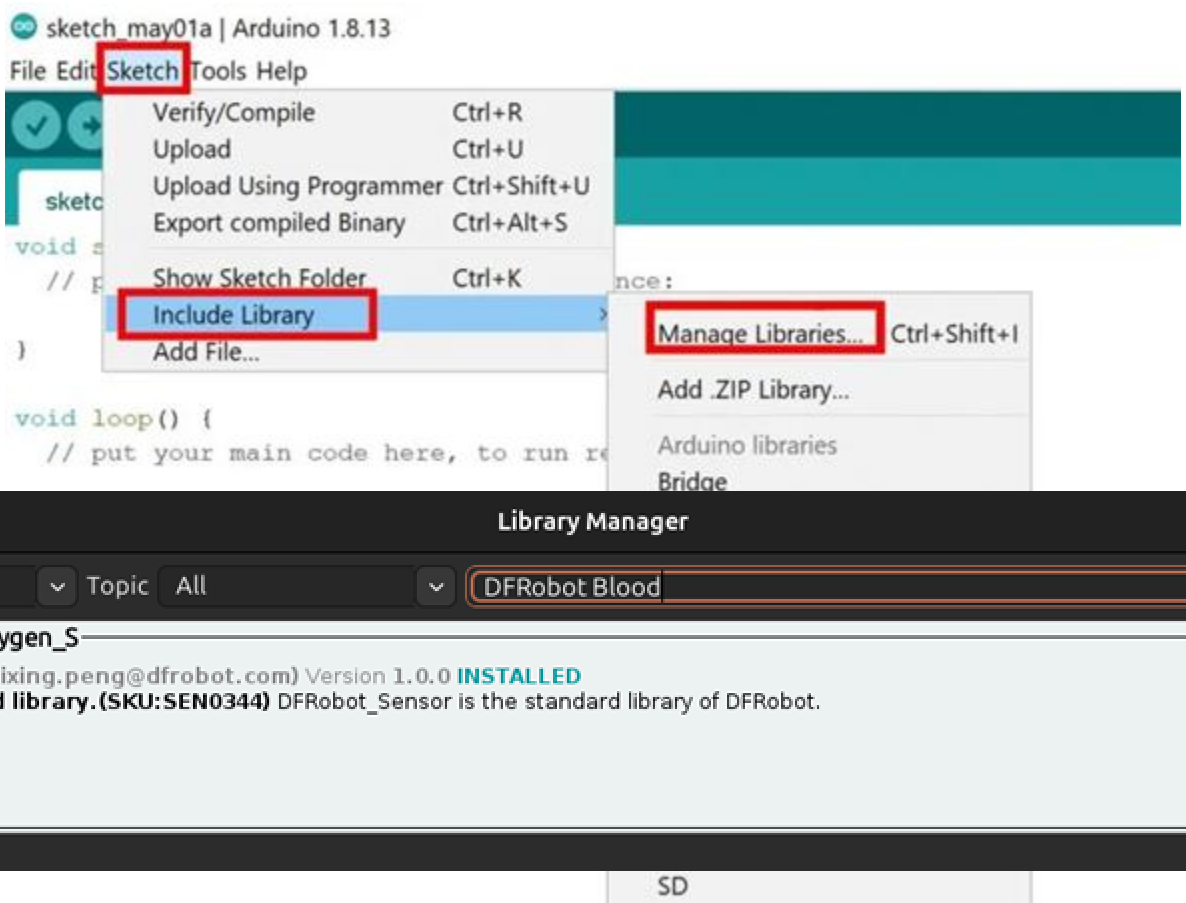
# Hardware Requirements & Wiring

| Component | Connection to ESP32 |
|---|---|
| ESP32 DevKit v1 | The microcontroller |
| MAX30102 Sensor (DFRobot v2) | 3V3 → 3V3<br>GND → GND<br>SDA → GPIO21<br>SCL → GPIO22 |

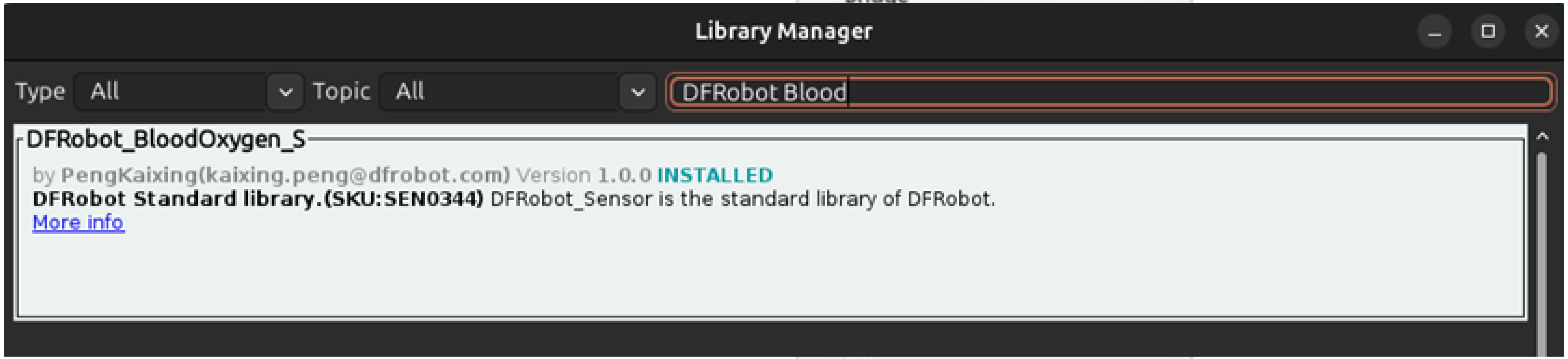




## Installing MAX30102 Library in Arduio IDE


sketch_may01a | Arduino 1.8.13
File Edit Sketch Tools Help
Verify/Compile Ctrl+R
Upload Ctrl+U
Upload Using Programmer Ctrl+Shift+U
Export compiled Binary Ctrl+Alt+S
Show Sketch Folder Ctrl+K
Include Library
Add File...
Manage Libraries... Ctrl+Shift+I
Add .ZIP Library...
Arduino libraries
Bridge
SD

```
void loop() {
  // put your main code here, to run re
```

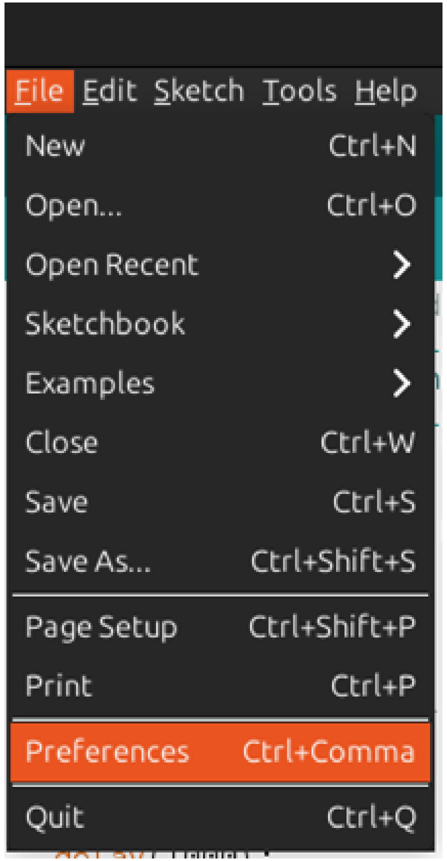


DFRobot Blood

DFRobot_BloodOxygen_S

by PengKaixing(kaixing.peng@dfrobot.com) Version 1.0.0 INSTALLED
**DFRobot Standard library.(SKU:SEN0344)** DFRobot_Sensor is the standard library of DFRobot.
More info

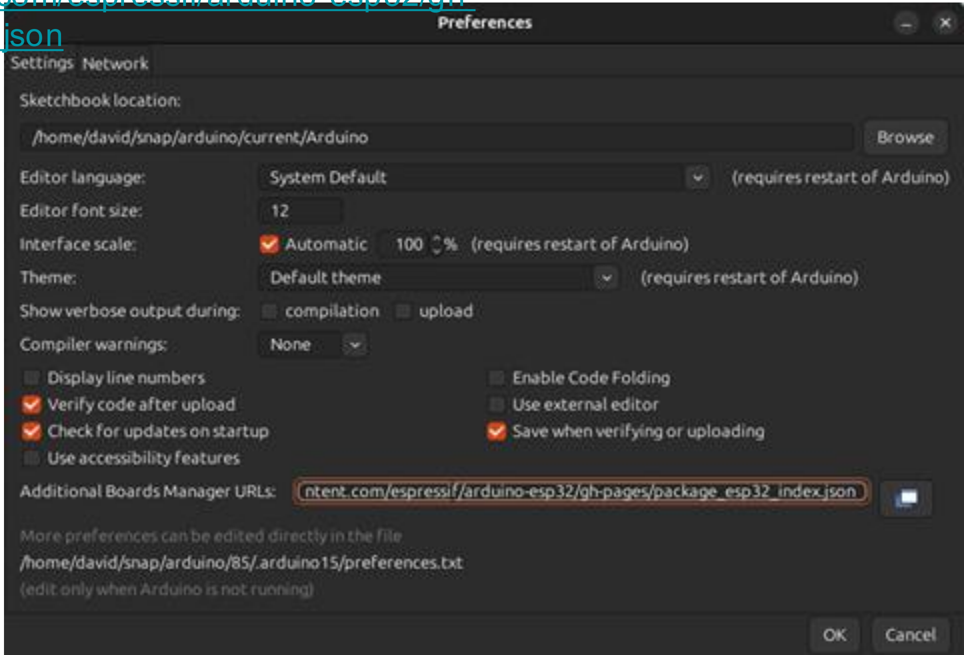
# Install ESP32 Board Package



**How to install:**

1. Open Arduino IDE → **Preferences**
2. In "Additional Board Manager URLs", add:

https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json



Sketchbook location:
/home/david/snap/arduino/current/Arduino

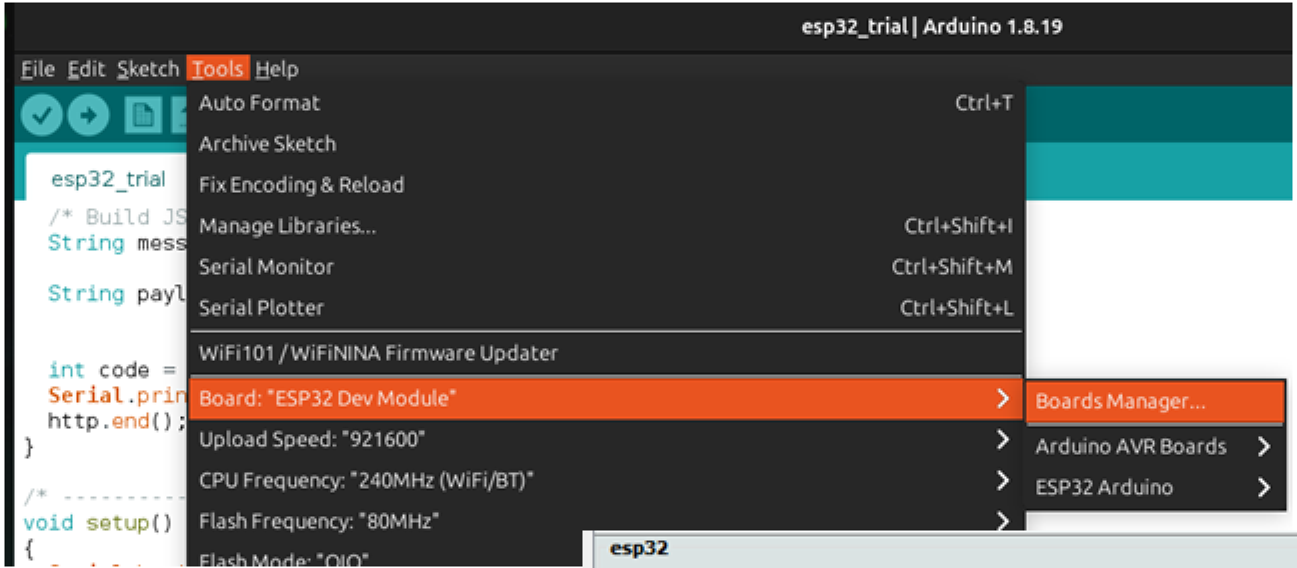
Editor language: System Default (requires restart of Arduino)
Editor font size: 12
Interface scale: ✓ Automatic 100 % (requires restart of Arduino)
Theme: Default theme (requires restart of Arduino)
Show verbose output during: ☐ compilation ☐ upload
Compiler warnings: None

☐ Display line numbers
✓ Verify code after upload
✓ Check for updates on startup
☐ Use accessibility features
☐ Enable Code Folding
☐ Use external editor
✓ Save when verifying or uploading

Additional Boards Manager URLs: ntent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json

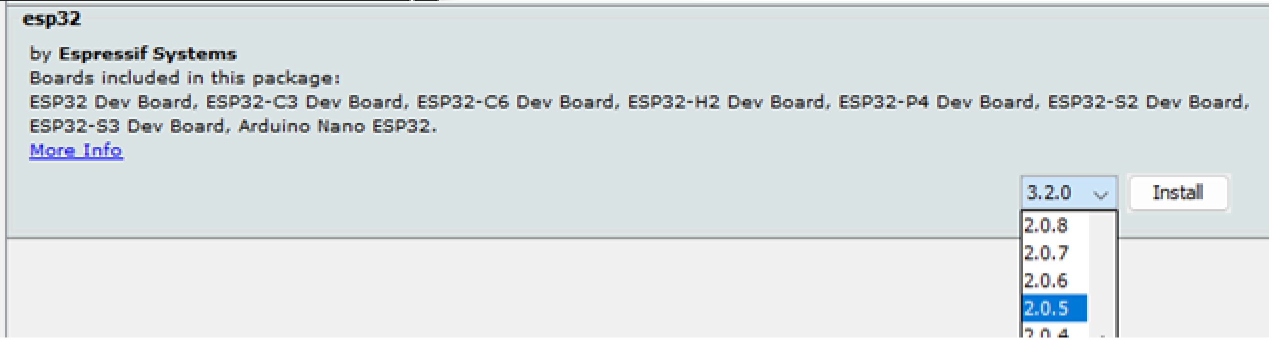
More preferences can be edited directly in the file
/home/david/snap/arduino/85/.arduino15/preferences.txt
(edit only when Arduino is not running)

# Install ESP32 Board Package


esp32_trial | Arduino 1.8.19

File Edit Sketch Tools Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Board: "ESP32 Dev Module"
Upload Speed: "921600"
CPU Frequency: "240MHz (WiFi/BT)"
Flash Frequency: "80MHz"
Flash Mode: "QIO"
Boards Manager...
Arduino AVR Boards
ESP32 Arduino

```
esp32_trial
  /* Build JS
  String mess

  String payl


  int code =
  Serial.prin
  http.end();
}

/* ---------
void setup()
{
```

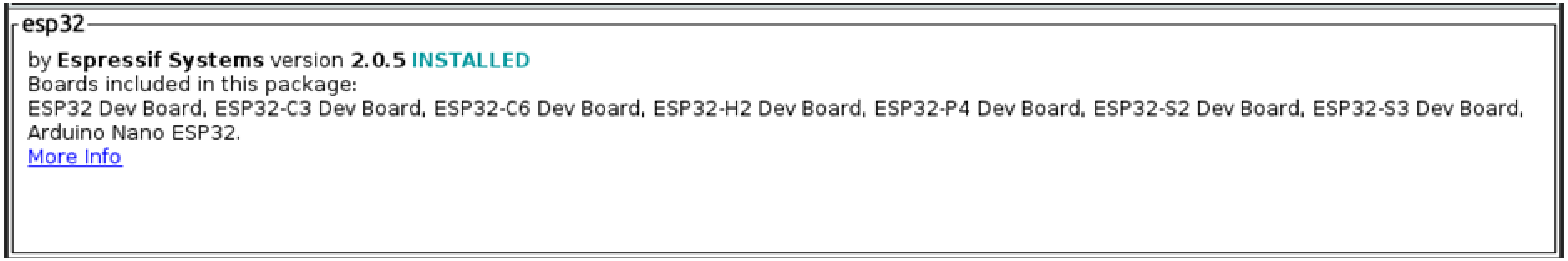

**esp32**

by **Espressif Systems**
Boards included in this package:
ESP32 Dev Board, ESP32-C3 Dev Board, ESP32-C6 Dev Board, ESP32-H2 Dev Board, ESP32-P4 Dev Board, ESP32-S2 Dev Board, ESP32-S3 Dev Board, Arduino Nano ESP32.
More Info

3. Then go to **Tools → Board → Board Manager**

4. Search for “ESP32” and install version **2.0.5** of **Espressif ESP32 platform**.

## Install ESP32 Board Package


esp32

by **Espressif Systems** version **2.0.5** **INSTALLED**
Boards included in this package:
ESP32 Dev Board, ESP32-C3 Dev Board, ESP32-C6 Dev Board, ESP32-H2 Dev Board, ESP32-P4 Dev Board, ESP32-S2 Dev Board, ESP32-S3 Dev Board, Arduino Nano ESP32.
More Info

**WiFi.h and HTTPClient.h**

- **Already included** with **ESP32 board package**.
- **WiFi.h**: Connects ESP32 to your Wi-Fi network.
- **HTTPClient.h**: Makes HTTPS requests to external servers (e.g., Telegram).

## Create a Telegram Bot

**a. Open Telegram App**

- Search for **@BotFather**
- Type `/start` then `/newbot`

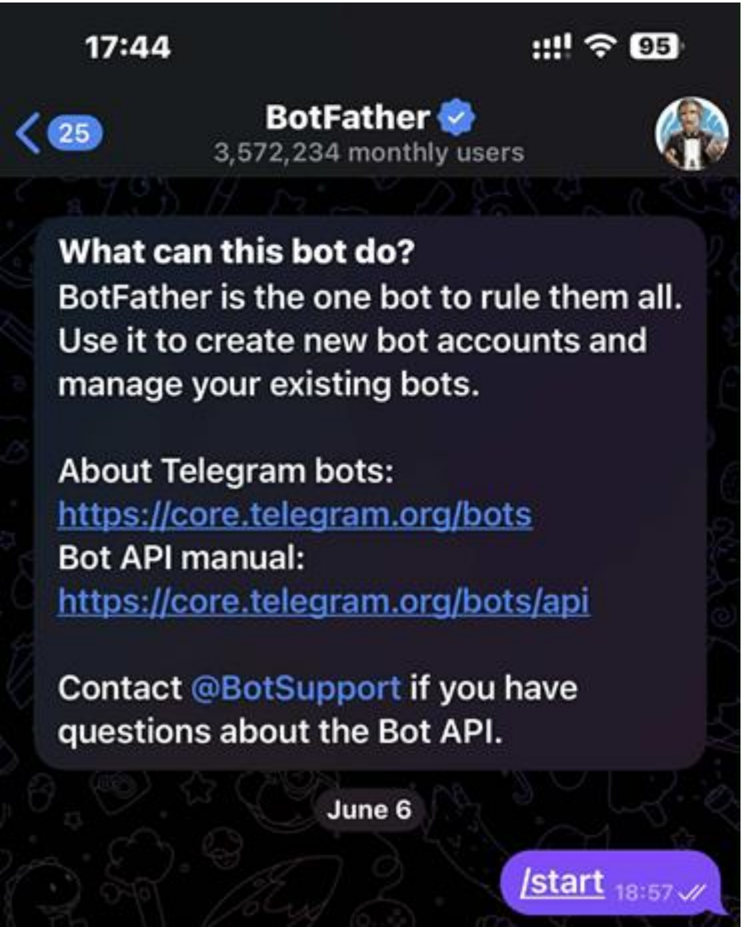
**b. Choose bot name and username**

- Name: `HRrate02%`
- Username: `Heart02Bot`

**c. BotFather replies with a token**

7867129252:AAGifceoByQW4zDy2N9PrnISSPTi12mnxhQ

# Create a Telegram Bot



17:44

BotFather
3,572,234 monthly users

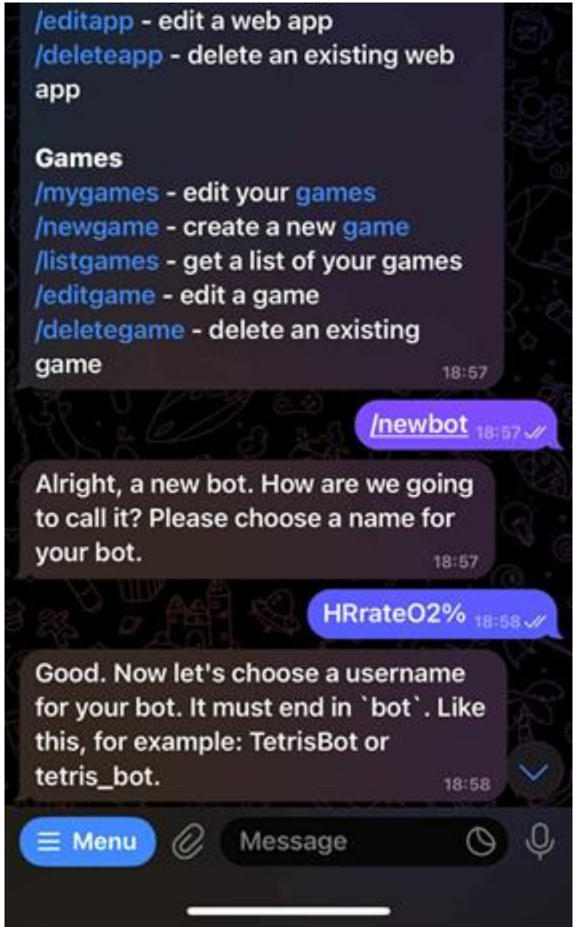
**What can this bot do?**
BotFather is the one bot to rule them all. Use it to create new bot accounts and manage your existing bots.

About Telegram bots:
https://core.telegram.org/bots
Bot API manual:
https://core.telegram.org/bots/api

Contact @BotSupport if you have questions about the Bot API.

June 6

/start 18:57



/editapp - edit a web app
/deleteapp - delete an existing web app

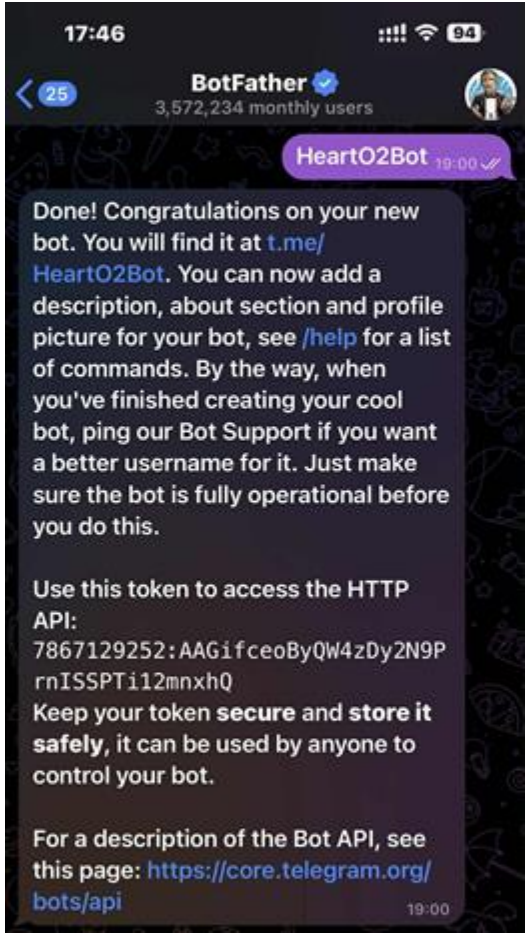
**Games**
/mygames - edit your games
/newgame - create a new game
/listgames - get a list of your games
/editgame - edit a game
/deletegame - delete an existing game 18:57

/newbot 18:57

Alright, a new bot. How are we going to call it? Please choose a name for your bot. 18:57

HRrateO2% 18:58

Good. Now let's choose a username for your bot. It must end in `bot`. Like this, for example: TetrisBot or tetris_bot. 18:58



17:46

BotFather
3,572,234 monthly users

HeartO2Bot 19:00

Done! Congratulations on your new bot. You will find it at t.me/HeartO2Bot. You can now add a description, about section and profile picture for your bot, see /help for a list of commands. By the way, when you've finished creating your cool bot, ping our Bot Support if you want a better username for it. Just make sure the bot is fully operational before you do this.

Use this token to access the HTTP API:
`7867129252:AAGifceoByQW4zDy2N9PrnISSPTi12mnxhQ`
Keep your token **secure** and **store it safely**, it can be used by anyone to control your bot.

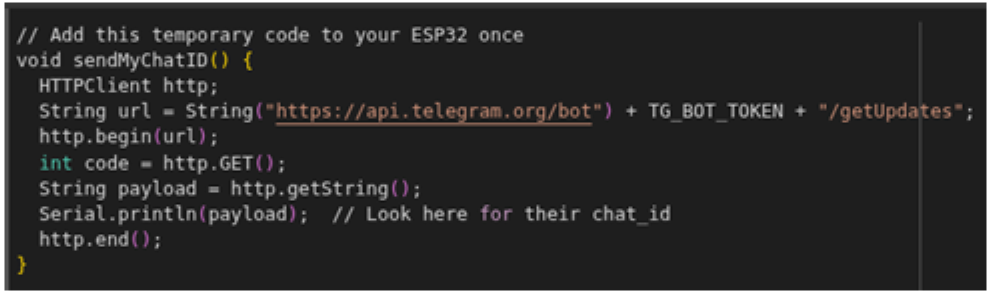
For a description of the Bot API, see this page: https://core.telegram.org/bots/api 19:00

# Get Your Telegram Chat ID

**a. Start a conversation with your bot in Telegram:**

1. Search for your bot's username.

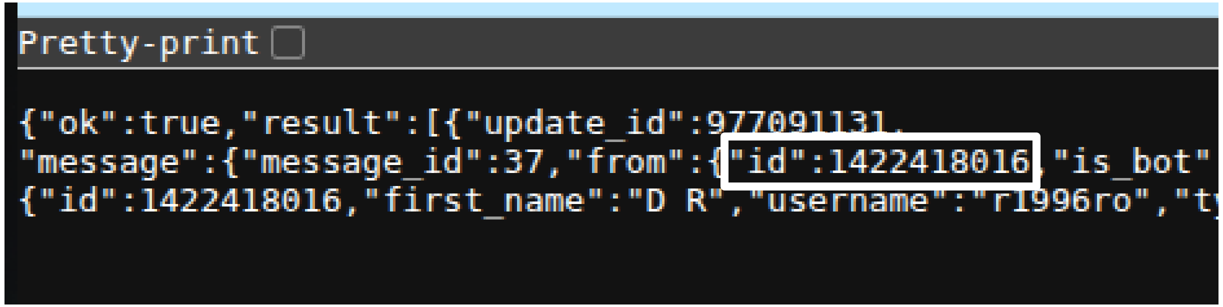
2. Click "Start" to activate it.



```
// Add this temporary code to your ESP32 once
void sendMyChatID() {
  HTTPClient http;
  String url = String("https://api.telegram.org/bot") + TG_BOT_TOKEN + "/getUpdates";
  http.begin(url);
  int code = http.GET();
  String payload = http.getString();
  Serial.println(payload);  // Look here for their chat_id
  http.end();
}
```

**b. Use this URL in your browser:**

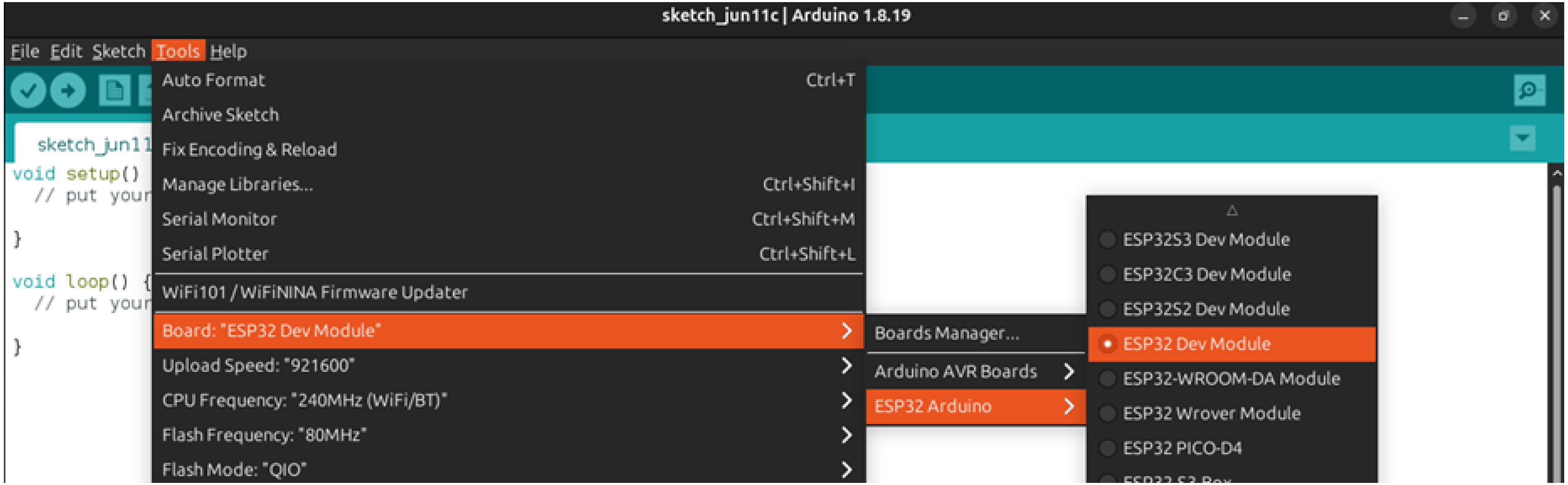
https://api.telegram.org/bot**7867129252:AAGifceoByQW4zDy2N9PrnISSPTi12mnxhQ**/getUpdates



```
{"ok":true,"result":[{"update_id":977091131,
"message":{"message_id":37,"from":{"id":1422418016,"is_bot"
{"id":1422418016,"first_name":"D R","username":"r1996ro","t
```

**Chat ID is 1422418016**

# Select the ESP32 board and Port



sketch_jun11c | Arduino 1.8.19

File Edit Sketch Tools Help

```
void setup() {
  // put your
}

void loop() {
  // put your
}
```

| | |
|---|---|
| Auto Format | Ctrl+T |
| Archive Sketch | |
| Fix Encoding & Reload | |
| Manage Libraries... | Ctrl+Shift+I |
| Serial Monitor | Ctrl+Shift+M |
| Serial Plotter | Ctrl+Shift+L |
| WiFi101 / WiFiNINA Firmware Updater | |
| Board: "ESP32 Dev Module" | |
| Upload Speed: "921600" | |
| CPU Frequency: "240MHz (WiFi/BT)" | |
| Flash Frequency: "80MHz" | |
| Flash Mode: "QIO" | |

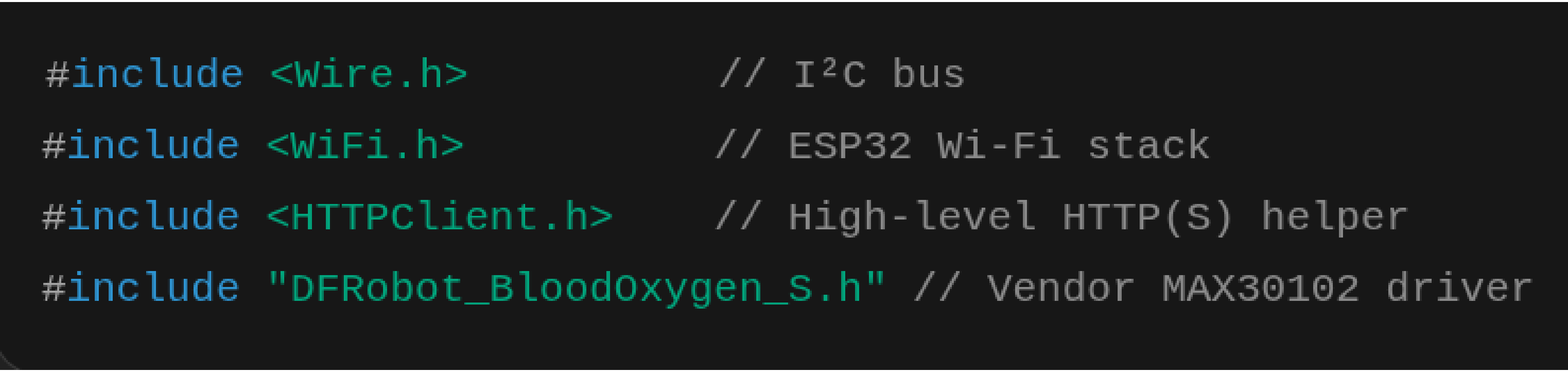
Boards Manager...
Arduino AVR Boards
ESP32 Arduino

ESP32S3 Dev Module
ESP32C3 Dev Module
ESP32S2 Dev Module
● ESP32 Dev Module
ESP32-WROOM-DA Module
ESP32 Wrover Module
ESP32 PICO-D4

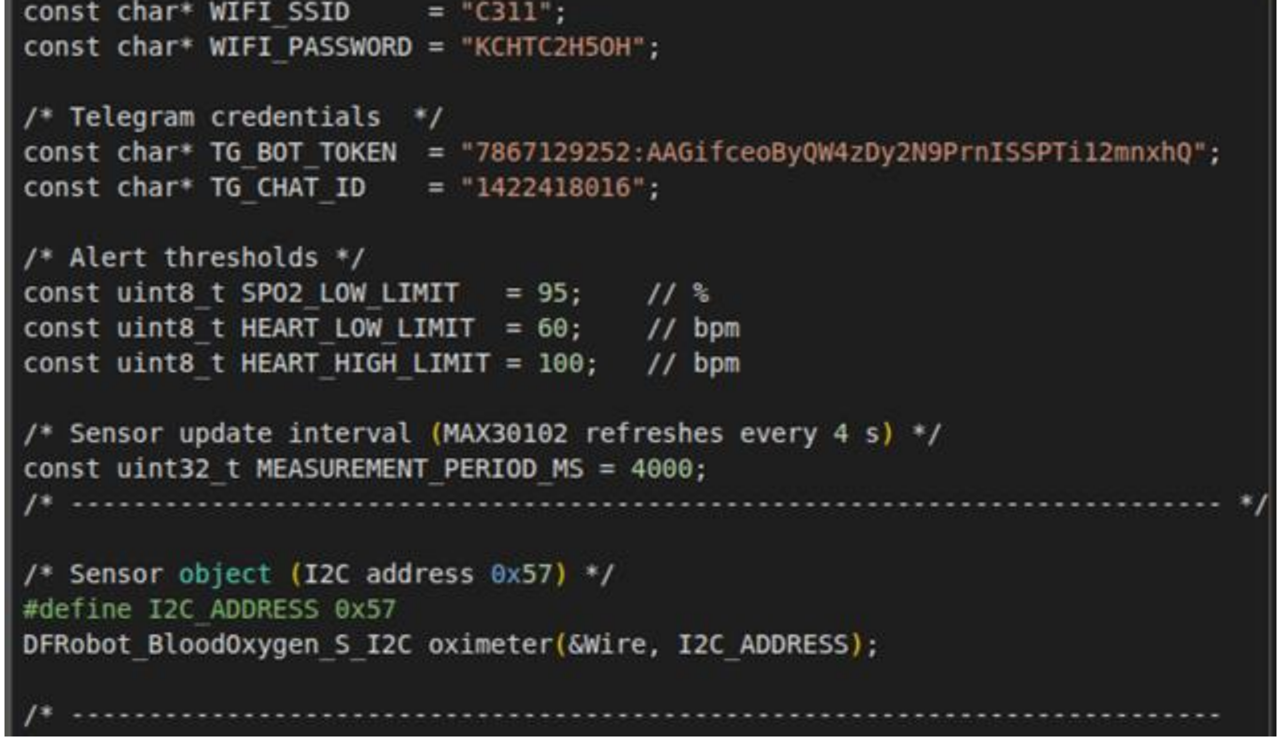
# ESP 32 - Arduino

```
/*******************************************************************************
 *  ESP32 – MAX30102 Heart-Rate & SpO₂ monitor with Telegram alert
 *  Hardware:
 *    • ESP32 DevKit v1 (or similar)
 *    • DFRobot Gravity MAX30102 Heart-Rate & Oximeter Sensor v2.0
 *  Connections (default I²C):
 *    ESP32 3V3  ->  VCC   |  ESP32 GND  ->  GND
 *    ESP32 GPIO21 -> SDA  |  ESP32 GPIO22 -> SCL
 *******************************************************************************/
```

## Library Imports

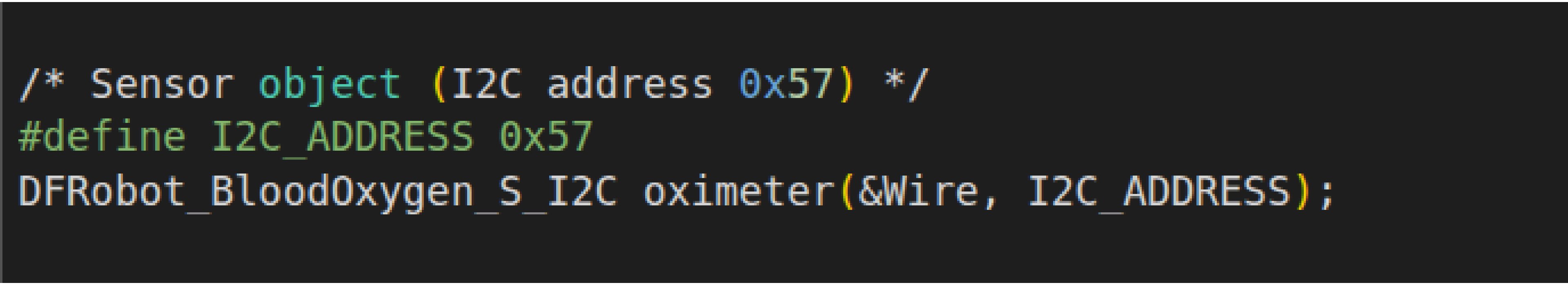


```
#include <Wire.h>          // I²C bus
#include <WiFi.h>          // ESP32 Wi-Fi stack
#include <HTTPClient.h>    // High-level HTTP(S) helper
#include "DFRobot_BloodOxygen_S.h" // Vendor MAX30102 driver
```



```
const char* WIFI_SSID     = "C311";
const char* WIFI_PASSWORD = "KCHTC2H5OH";

/* Telegram credentials  */
const char* TG_BOT_TOKEN  = "7867129252:AAGifceoByQW4zDy2N9PrnISSPTi12mnxhQ";
const char* TG_CHAT_ID    = "1422418016";

/* Alert thresholds */
const uint8_t SPO2_LOW_LIMIT   = 95;    // %
const uint8_t HEART_LOW_LIMIT  = 60;    // bpm
const uint8_t HEART_HIGH_LIMIT = 100;   // bpm

/* Sensor update interval (MAX30102 refreshes every 4 s) */
const uint32_t MEASUREMENT_PERIOD_MS = 4000;
/* ---------------------------------------------------------------------------- */

/* Sensor object (I2C address 0x57) */
#define I2C_ADDRESS 0x57
DFRobot_BloodOxygen_S_I2C oximeter(&Wire, I2C_ADDRESS);

/* -------------------------------------------------------------------------
```

What 95 means?
Uint8? Unified integer 8-bit



```
/* Sensor object (I2C address 0x57) */
#define I2C_ADDRESS 0x57
DFRobot_BloodOxygen_S_I2C oximeter(&Wire, I2C_ADDRESS);
```

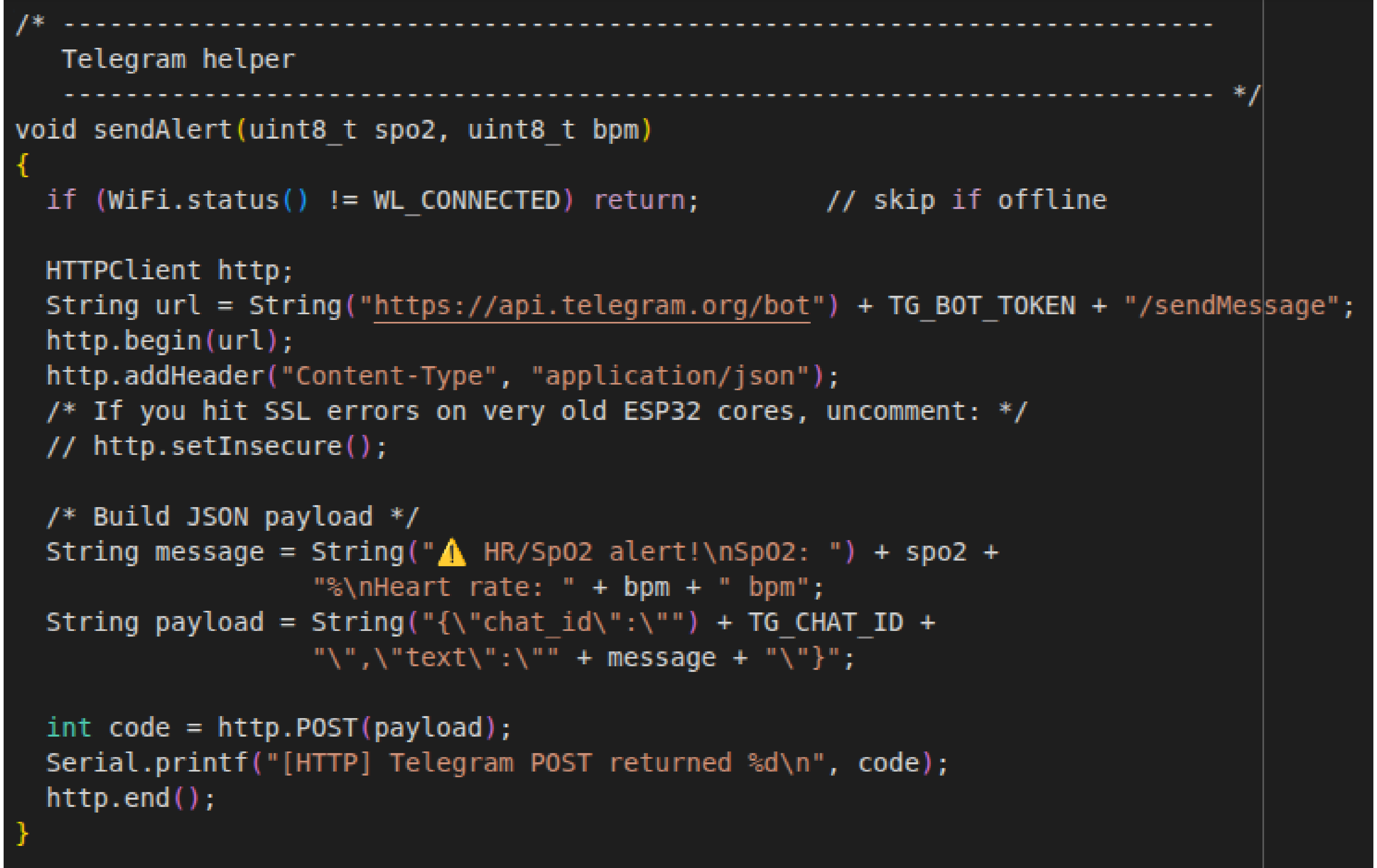
`Sensor object (I2C_ADDRESS 0x57)`

→ **Think of this as setting the house number of the sensor on the shared I²C "street".**

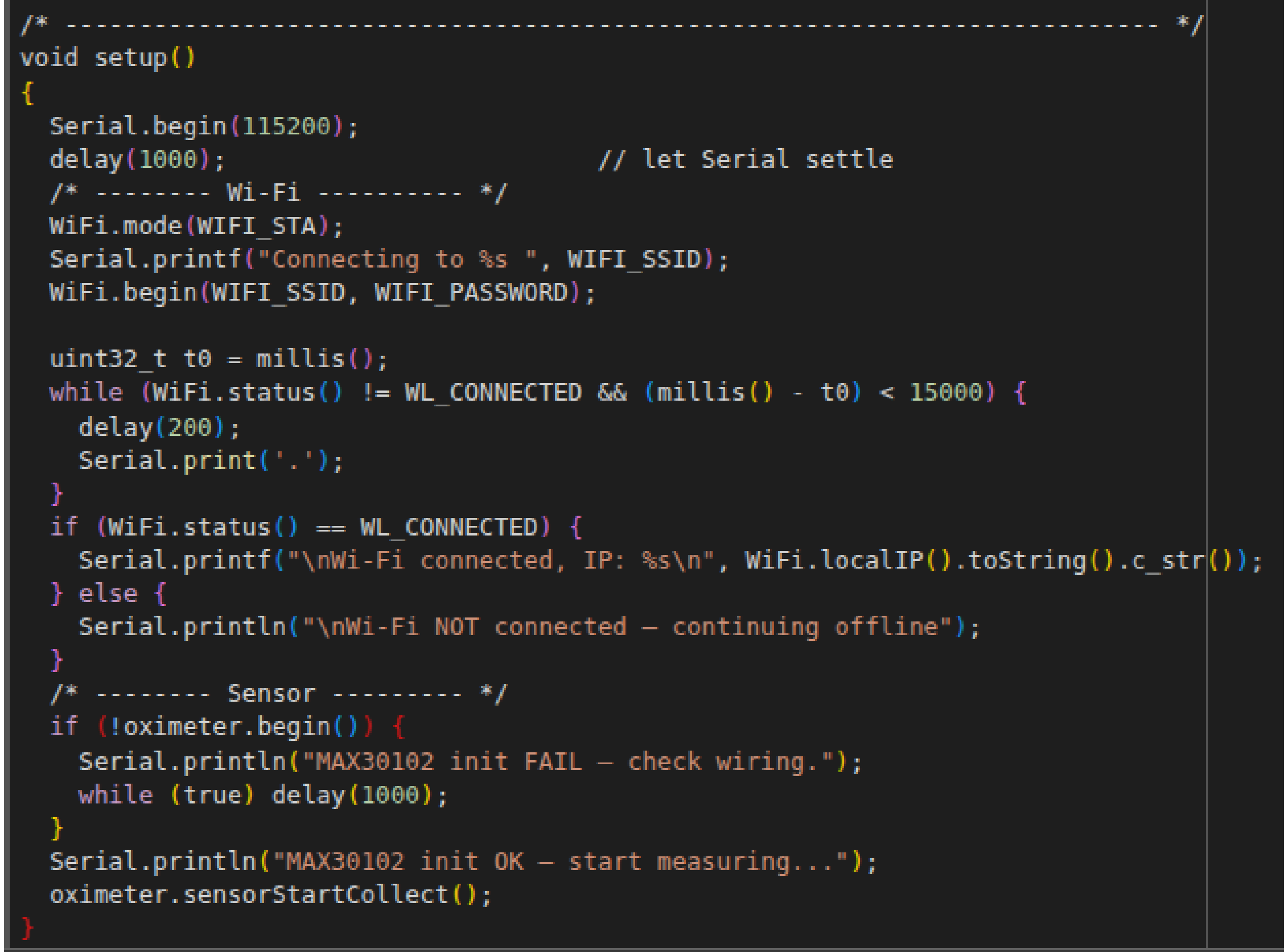
`DFRobot_BloodOxygen_S_I2C oximeter(&Wire, I2C_ADDRESS);`

→ **Like assigning a smart assistant (the object) who knows how to talk to the sensor and fetch meaningful health data for you.**

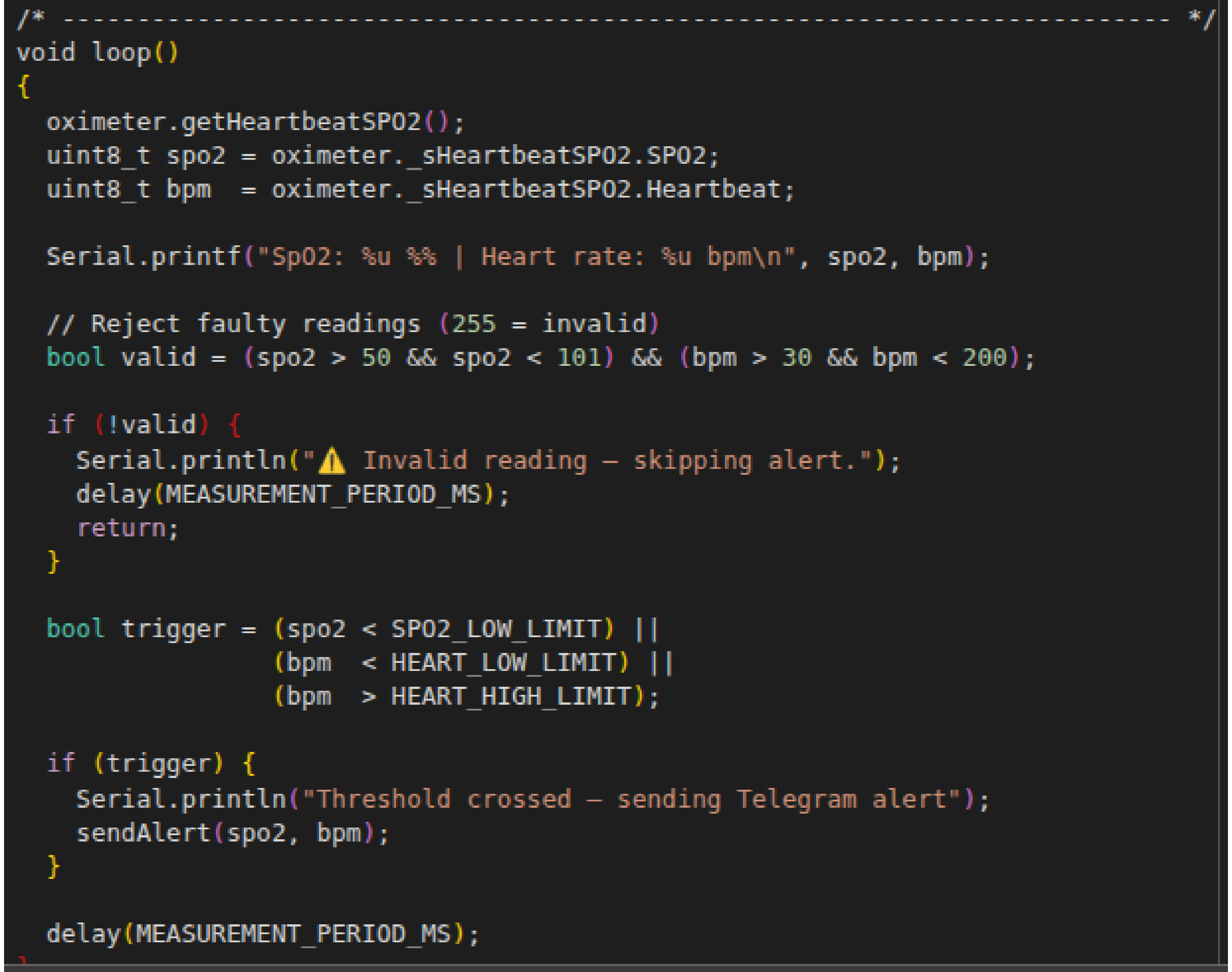
What 0x57 mean?



```
/* ----------------------------------------------------------------------------
   Telegram helper
   ---------------------------------------------------------------------------- */
void sendAlert(uint8_t spo2, uint8_t bpm)
{
  if (WiFi.status() != WL_CONNECTED) return;        // skip if offline

  HTTPClient http;
  String url = String("https://api.telegram.org/bot") + TG_BOT_TOKEN + "/sendMessage";
  http.begin(url);
  http.addHeader("Content-Type", "application/json");
  /* If you hit SSL errors on very old ESP32 cores, uncomment: */
  // http.setInsecure();

  /* Build JSON payload */
  String message = String("⚠ HR/SpO2 alert!\nSpO2: ") + spo2 +
                   "%\nHeart rate: " + bpm + " bpm";
  String payload = String("{\"chat_id\":\"") + TG_CHAT_ID +
                   "\",\"text\":\"" + message + "\"}";

  int code = http.POST(payload);
  Serial.printf("[HTTP] Telegram POST returned %d\n", code);
  http.end();
}
```

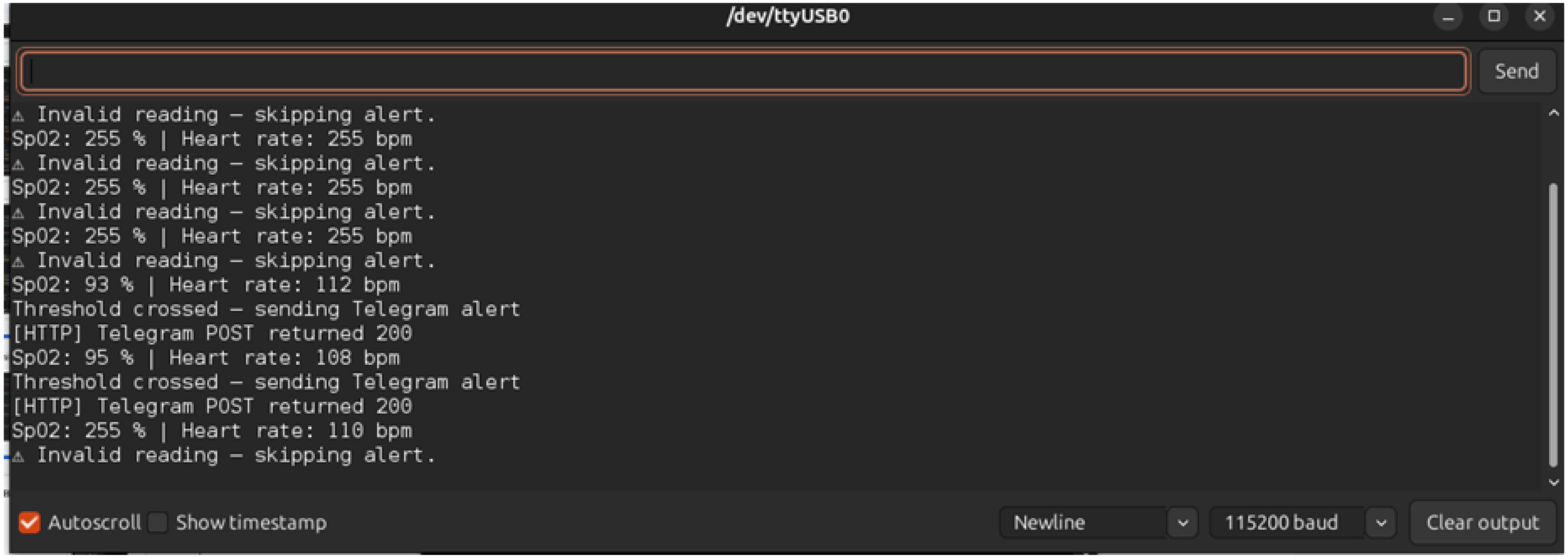


```
/* ------------------------------------------------------------------------- */
void setup()
{
  Serial.begin(115200);
  delay(1000);                         // let Serial settle
  /* -------- Wi-Fi ---------- */
  WiFi.mode(WIFI_STA);
  Serial.printf("Connecting to %s ", WIFI_SSID);
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);

  uint32_t t0 = millis();
  while (WiFi.status() != WL_CONNECTED && (millis() - t0) < 15000) {
    delay(200);
    Serial.print('.');
  }
  if (WiFi.status() == WL_CONNECTED) {
    Serial.printf("\nWi-Fi connected, IP: %s\n", WiFi.localIP().toString().c_str());
  } else {
    Serial.println("\nWi-Fi NOT connected – continuing offline");
  }
  /* -------- Sensor --------- */
  if (!oximeter.begin()) {
    Serial.println("MAX30102 init FAIL – check wiring.");
    while (true) delay(1000);
  }
  Serial.println("MAX30102 init OK – start measuring...");
  oximeter.sensorStartCollect();
}
```



```
/* ------------------------------------------------------------------------ */
void loop()
{
  oximeter.getHeartbeatSPO2();
  uint8_t spo2 = oximeter._sHeartbeatSPO2.SPO2;
  uint8_t bpm  = oximeter._sHeartbeatSPO2.Heartbeat;

  Serial.printf("SpO2: %u %% | Heart rate: %u bpm\n", spo2, bpm);

  // Reject faulty readings (255 = invalid)
  bool valid = (spo2 > 50 && spo2 < 101) && (bpm > 30 && bpm < 200);

  if (!valid) {
    Serial.println("⚠️ Invalid reading – skipping alert.");
    delay(MEASUREMENT_PERIOD_MS);
    return;
  }

  bool trigger = (spo2 < SPO2_LOW_LIMIT) ||
                 (bpm  < HEART_LOW_LIMIT) ||
                 (bpm  > HEART_HIGH_LIMIT);

  if (trigger) {
    Serial.println("Threshold crossed – sending Telegram alert");
    sendAlert(spo2, bpm);
  }

  delay(MEASUREMENT_PERIOD_MS);
```

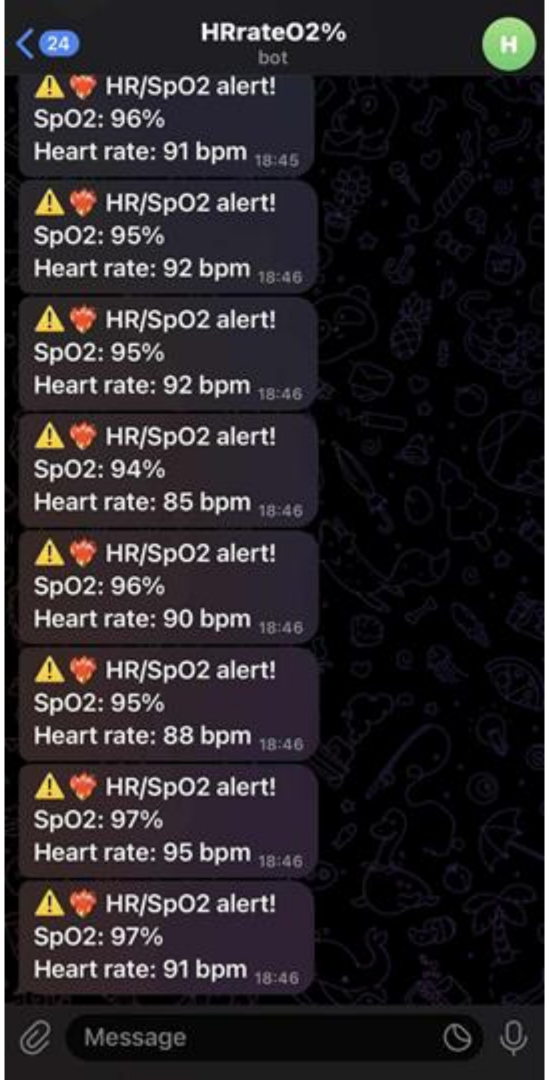
# Serial monitor



```
⚠ Invalid reading – skipping alert.
SpO2: 255 % | Heart rate: 255 bpm
⚠ Invalid reading – skipping alert.
SpO2: 255 % | Heart rate: 255 bpm
⚠ Invalid reading – skipping alert.
SpO2: 255 % | Heart rate: 255 bpm
⚠ Invalid reading – skipping alert.
SpO2: 93 % | Heart rate: 112 bpm
Threshold crossed – sending Telegram alert
[HTTP] Telegram POST returned 200
SpO2: 95 % | Heart rate: 108 bpm
Threshold crossed – sending Telegram alert
[HTTP] Telegram POST returned 200
SpO2: 255 % | Heart rate: 110 bpm
⚠ Invalid reading – skipping alert.
```

[✓ Autoscroll] Show timestamp · Newline · 115200 baud

# Telegram Bot Alert



⚠️❤️ HR/SpO2 alert!
SpO2: 96%
Heart rate: 91 bpm 18:45

⚠️❤️ HR/SpO2 alert!
SpO2: 95%
Heart rate: 92 bpm 18:46

⚠️❤️ HR/SpO2 alert!
SpO2: 95%
Heart rate: 92 bpm 18:46

⚠️❤️ HR/SpO2 alert!
SpO2: 94%
Heart rate: 85 bpm 18:46

⚠️❤️ HR/SpO2 alert!
SpO2: 96%
Heart rate: 90 bpm 18:46

⚠️❤️ HR/SpO2 alert!
SpO2: 95%
Heart rate: 88 bpm 18:46

⚠️❤️ HR/SpO2 alert!
SpO2: 97%
Heart rate: 95 bpm 18:46

⚠️❤️ HR/SpO2 alert!
SpO2: 97%
Heart rate: 91 bpm 18:46