

A Telegram bot on the ESP32: your home from your phone



Online guide,
always up to date

You create the bot with BotFather, the ESP32 reads the commands and replies: temperature and LED from anywhere, without touching the router.

By **Ada Intelisano** · informaticait.it technical team

IN SHORT

- You create the bot with @BotFather; keep the token secret.
- The board asks for messages every second: no ports to open.
- The sketch only answers your chat number.
- Libraries: UniversalTelegramBot, ArduinoJson 6, Adafruit BME280.

Time: 1 hour

Level: next steps

Before: guide no. 56

Send **/status** to your bot on Telegram and it replies with the temperature at home; send **/led_on** and a light turns on. It works from anywhere, even away from home, without opening ports on the router: it is the board that goes and asks Telegram for messages.

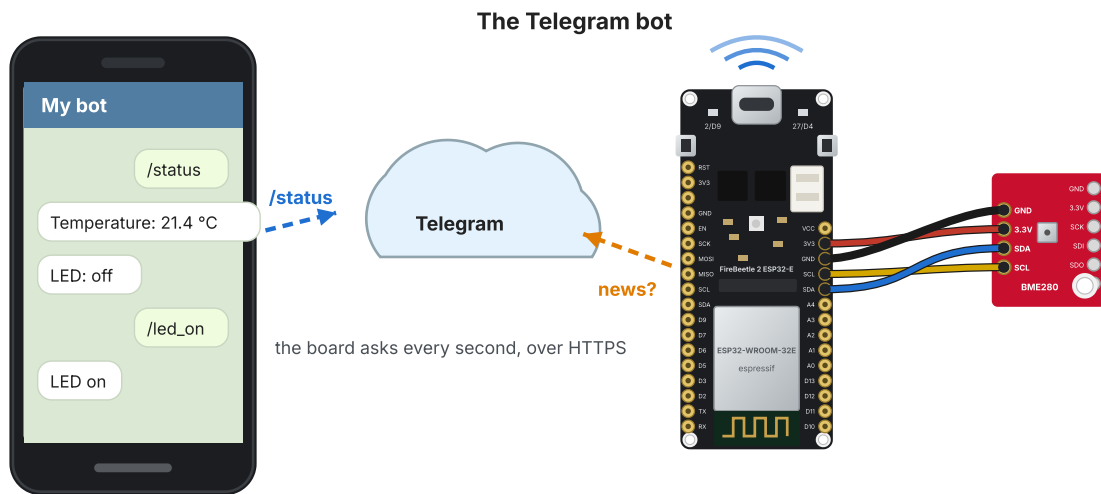
1 Create the bot

1. On Telegram search for **@BotFather** (the one with the blue check mark) and send it **/newbot**.
2. Choose a name and a username that ends in "bot".
3. BotFather gives you the **token**, a line like 110201543:AAHdqTcvCH1v.... Copy it.
4. With **/setcommands** you can give BotFather the list of commands: they will show up as suggestions when you type **"/**.

Warning: the token

- The token is like your house key: whoever has it controls the bot. Do not publish it, do not send it in a chat, do not leave it in a sketch you share.

2 How it works



informaticait.it

The phone sends the message to the Telegram servers. Every second the board asks "any new messages?" (this is called **polling**); if there are, it reads them, runs the command and replies. Everything goes over HTTPS, which is why the sketch contains the Telegram certificate.

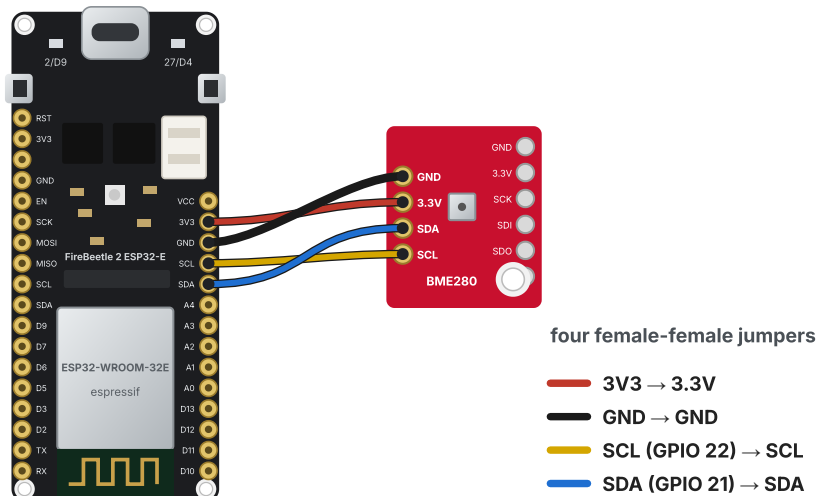
3 Find your chat number

Anyone can find a bot by searching for its name and write to it. That is why the sketch only obeys **one** chat, yours. To find its number: send any message to your bot, then open in the browser <https://api.telegram.org/botTOKEN/getUpdates> (with your token in place of TOKEN). In the response look for "chat":{"id"::: the number that follows is the one to put in MY_CHAT.

4

Connect the sensor and upload

FireBeetle 2 ESP32-E and BME280


informaticait.it

The sensor is connected as in guide no. 56. Install the **UniversalTelegramBot** (Brian Lough), **ArduinoJson** (version 6) and **Adafruit BME280** libraries. Write your network, password, token and chat number in the first lines.

```
// Telegram bot: turn the LED on and off and ask for the temperature from your phone. It only
// answers your chat.
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>
#include <Adafruit_BME280.h>

const char* NETWORK = "your-network-name";
const char* PASSWORD = "your-network-password";
#define TOKEN "123456789:the-token-BotFather-gave-you" // secret: never publish it
const String MY_CHAT = "123456789"; // your chat number (step 3)

const int LED = 2;
WiFiClientSecure client;
UniversalTelegramBot bot(TOKEN, client);
Adafruit_BME280 bme;
unsigned long lastCheck = 0;

void reply(int count) {
  for (int i = 0; i < count; i++) {
    String chat = bot.messages[i].chat_id;
    String text = bot.messages[i].text;
    if (chat != MY_CHAT) { // a stranger: do not obey
      bot.sendMessage(chat, "You are not authorized.", "");
      continue;
    }
    if (text == "/led_on") {
      digitalWrite(LED, HIGH);
      bot.sendMessage(chat, "LED on", "");
    } else if (text == "/led_off") {
```

```

        digitalWrite(LED, LOW);
        bot.sendMessage(chat, "LED off", "");
    } else if (text == "/status") {
        String r = "Temperature: " + String(bme.readTemperature(), 1) + " °C\n";
        r += "Humidity: " + String(bme.readHumidity(), 0) + " %\n";
        r += "LED: " + String(digitalRead(LED) ? "on" : "off");
        bot.sendMessage(chat, r, "");
    } else {
        bot.sendMessage(chat, "Commands: /led_on, /led_off, /status", "");
    }
}

void setup() {
    Serial.begin(115200);
    pinMode(LED, OUTPUT);
    if (!bme.begin(0x77)) Serial.println("BME280 not found: check the wires");
    WiFi.mode(WIFI_STA);
    WiFi.begin(NETWORK, PASSWORD);
    client.setCACert(TELEGRAM_CERTIFICATE_ROOT); // the certificate to talk to Telegram securely
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    configTime(0, 0, "pool.ntp.org"); // the exact time, as in the library example
    Serial.println("\nConnected. Write to your bot.");
}

void loop() {
    if (millis() - lastCheck > 1000) { // one check per second, as in the library
example
        int newCount = bot.getUpdates(bot.last_message_received + 1);
        while (newCount) {
            reply(newCount);
            newCount = bot.getUpdates(bot.last_message_received + 1);
        }
        lastCheck = millis();
    }
}

```

5 Try it

✓ What you should see

Send /status to the bot: within a few seconds you get the temperature, humidity and LED state. /led_on and /led_off turn the board's LED on and off.

If it doesn't work

- It never replies: did you copy the whole token, with no spaces? Is the board connected to the network (check the Serial Monitor)?
- It replies "You are not authorized" to you too: the number in MY_CHAT is not the right one (repeat step 3).
- Compile errors with ArduinoJson: you have version 7; the bot library is written for version 6.

6 Two things to know

- Telegram asks you not to send more than one message per second in the same chat: no loops that fire off messages in bursts.
- A bot receives messages either by polling or through a webhook, not both: if one day you set a webhook, this sketch stops receiving.

7 Challenge yourself

Try it yourself

- An alarm: if the temperature goes above 30 °C the bot writes to you on its own (once, not at every reading).
- Control the relay from guide no. 43 instead of the LED.

✓ What you learned

- you create a bot with BotFather and control it with a secret token;
- the board goes and asks for messages (polling), so there is no need to open ports on the router;
- a bot is public: the sketch must only answer your chat.

Sources: Telegram, bot documentation and Bot API (core.telegram.org) · UniversalTelegramBot library by Brian Lough and ESP32 FlashLED example · DFRobot, FireBeetle 2 ESP32-E wiki · [informaticait](https://www.informaticait.it) guides no. 43 and 56