Readings on a Google sheet, with deep sleep

Every ten minutes the ESP32 wakes up, reads the BME280, adds a row to a Google sheet with an Apps Script and goes back to sleep.

Online guide, always up to date

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

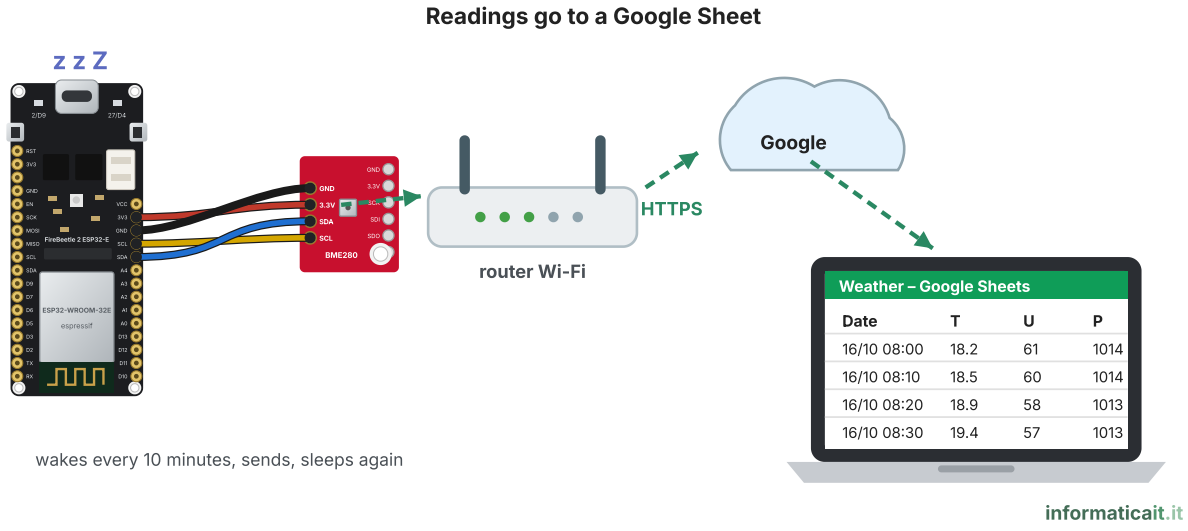
IN SHORT

- An Apps Script deployed as a web app writes the rows into the sheet.
- The board calls the /exec address with the values and follows Google's redirect.
- Deep sleep: on waking it starts again from setup(); RTC_DATA_ATTR keeps the counters.
- The /exec address is a key: do not publish it.

Time: 1 hour | Level: next steps | Before: guide no. 56

Every ten minutes the board wakes up, reads temperature, humidity and pressure, adds a row to a **Google sheet** and goes back to sleep. From your phone you see the history and make charts with the sheet's own tools. It is free, and the board sleeps almost all the time.

1 How it works



New word: Apps Script
Small JavaScript programs that run on Google's servers, attached to your sheet. Ours waits for the values from the board and writes them into a new row.

2 Prepare the sheet and the script

1. Create a new Google sheet; in the first row write **Date, Temperature, Humidity, Pressure**.
2. From the **Extensions** → **Apps Script** menu open the editor, delete what is there and paste the code below. Save.

```
// Google Apps Script: receives the readings from the board and appends them as a row to the sheet.
// Paste it in Extensions > Apps Script of the Google sheet, then deploy it as a web app.
function doGet(e) {
  const sheet = SpreadsheetApp.getActiveSpreadsheet().getSheets()[0];
  const t = e.parameter.t;              // temperature
  const u = e.parameter.u;              // humidity
  const p = e.parameter.p;              // pressure
  sheet.appendRow([new Date(), t, u, p]);    // date and time, then the three values
  return ContentService.createTextOutput('ok');
}
```

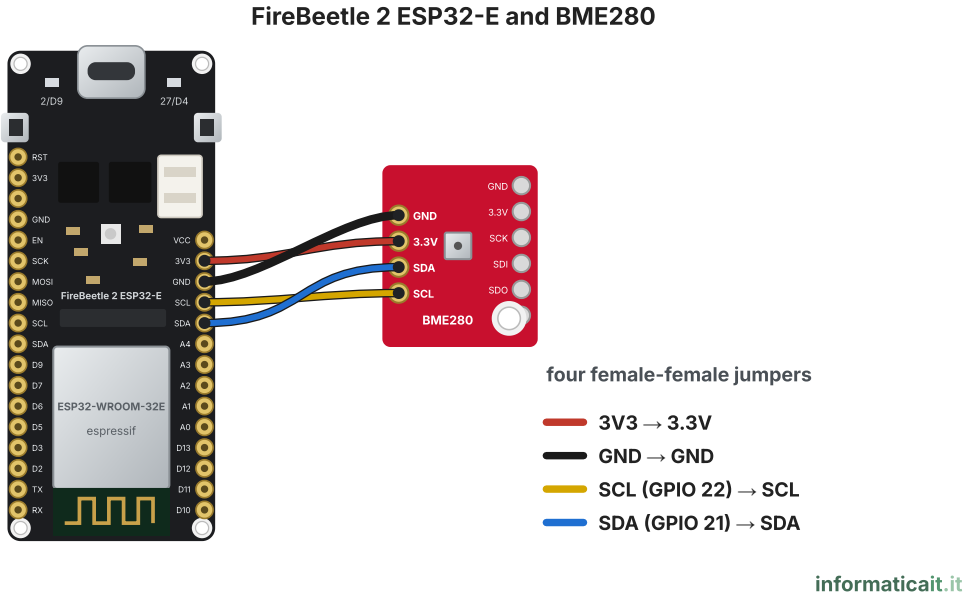
3 Deploy it as a web app

1. Top right: **Deploy** → **New deployment**, type **Web app**.
2. Execute as: **Me**; access: open to anyone with the link, so the board can call it without logging in.
3. Authorize access to your sheet when Google asks, then copy the address ending in `/exec`.

Test it right away from the browser: paste the address and add `?t=20&u=50&p=1013`. It must answer “ok” and a row appears in the sheet.

> **Warning**
> - The /exec address is the key to your sheet: whoever knows it can write to it. Do not publish it.

4 The circuit and the sketch



The sensor is connected as in guide no. 56. In the sketch, write your network name, password and the `/exec` address.

```
// Every 10 minutes the board wakes up, reads the BME280, sends the values to a Google sheet and goes back to sleep.
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include <HTTPClient.h>
#include <Adafruit_BME280.h>

const char* NETWORK = "your-network-name";
const char* PASSWORD = "your-network-password";
const String SCRIPT_URL = "https://script.google.com/macros/s/YOUR-ID/exec";   // from the web app (step 3)
const uint64_t MINUTES = 10;                                                   // how often to measure

RTC_DATA_ATTR int wakeups = 0;     // stays in memory even during deep sleep
Adafruit_BME280 bme;

void sleepNow() {
  Serial.printf("Going to sleep for %llu minutes\n", MINUTES);
  Serial.flush();
  esp_sleep_enable_timer_wakeup(MINUTES * 60ULL * 1000000ULL);   // the time is in microseconds
  esp_deep_sleep_start();                                         // there is no coming back from here: on waking, setup() starts again
}

void setup() {
  Serial.begin(115200);
  wakeups++;
  Serial.printf("Wake-up no. %d\n", wakeups);
  if (!bme.begin(0x77)) {
    Serial.println("BME280 not found");
```

```
    sleepNow();
  }
  float t = bme.readTemperature();
  float u = bme.readHumidity();
  float p = bme.readPressure() / 100.0;

  WiFi.mode(WIFI_STA);
  WiFi.begin(NETWORK, PASSWORD);
  unsigned long start = millis();
  while (WiFi.status() != WL_CONNECTED && millis() - start < 15000) delay(250);
  if (WiFi.status() != WL_CONNECTED) {        // no network: try again at the next wake-up
    Serial.println("Wi-Fi not available");
    sleepNow();
  }

  WiFiClientSecure client;
  client.setInsecure();                       // no certificate check: see the guide
  HTTPClient http;
  http.setFollowRedirects(HTTPC_STRICT_FOLLOW_REDIRECTS);   // Google answers with a redirect to
follow
  String url = SCRIPT_URL + "?t=" + String(t, 1) + "&u=" + String(u, 0) + "&p=" + String(p, 0);
  http.begin(client, url);
  int code = http.GET();
  Serial.printf("Response: %d %s\n", code, http.getString().c_str());
  http.end();
  sleepNow();
}

void loop() {}   // never reached: after setup() the board sleeps
```

New word: deep sleep
The processor and almost all of the memory switch off; only a clock stays on to wake the board up. On waking, the sketch **starts over from the beginning**, from `setup()`: that is why `loop()` is empty. Variables marked `RTC_DATA_ATTR` survive the sleep: `wakeups` counts the measurements.

New word: redirect
Google does not answer directly: it sends the board on to another address. `setFollowRedirects` tells it to follow it.

5 Try it

✓ What you should see

In the Serial Monitor: “Wake-up no. 1”, “Response: 200 ok”, “Going to sleep for 10 minutes”. A new row appears in the sheet; ten minutes later, another one.

If it doesn't work

- Response -1 or nothing: the network does not reach that far, or the /exec address is wrong.
- Response 302 or an HTML page instead of “ok”: `setFollowRedirects` is missing, or the deployment is not open to anyone with the link.
- You change the script and nothing changes: every change must be deployed again (Manage deployments → new version).

6 Two things to know

- **Security**: with `setInsecure()` the board does not check Google's certificate. For home readings this is an acceptable compromise; for important data you load the root certificate with `setCACert()`.
- **Power consumption**: for the ESP32 chip alone Espressif states 10 µA in deep sleep, but the whole board draws more and DFRobot gives inconsistent figures: on batteries, measure it (guide no. 24).

7 Challenge yourself

Try it yourself

- Add a column with `wakeups`, so you notice if a measurement gets lost.
- Create a temperature chart in the sheet: it updates by itself with every row.

✓ What you learned

- an Apps Script deployed as a web app receives the data and writes it into the sheet;
- the board sends an HTTPS request and follows Google's redirect;
- with deep sleep the board sleeps between measurements and starts again from `setup()`.

Sources: Google, Apps Script: web apps, Content Service, SpreadsheetApp, quotas · Espressif: sleep modes and ESP32 datasheet v5.3 · Espressif, arduino-esp32 2.0.17: HTTPClient and WiFiClientSecure · DFRobot, FireBeetle 2 ESP32-E wiki (deep sleep, jumper) · informaticait guides no. 23, 24 and 56