



Grafana you can touch



Richard “RichiH” Hartmann

- Munich, Germany
- Community work & Office of the CTO
- Grafanista since 2018
- 3D printing, reading, woodworking



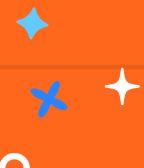
Willie Engelbrecht

- London, UK
- Solutions Engineering at Grafana
- Grafanista since 2021
- Factorio addict, amateur astrophotography



Antonio Calero

- Cadiz, Spain
- Developer advocate
- Almost 3 years at Grafana Labs
- Dad to a 4 year-old, Crossfit, metalhead.



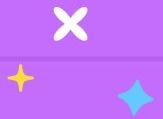
Overview of our IoT project

 Setting up the lab environment

 Download, compile and run the code

 Explore and visualize data in a
dashboard

What's on display



What's on display



M5StickC-Plus

A microcontroller to interact with the hardware



Expansion unit

Enables the connection of multiple devices



EARTH

Measures the moisture content in soil



TVOC/eCO2

Measures various VOC (volatile organic compounds) and hydrogen gas



DLIGHT

Detects light intensity



Heart rate unit

Measures pulse and heart rate



Ultrasonic Distance

Detects objects within 2cm to 450cm

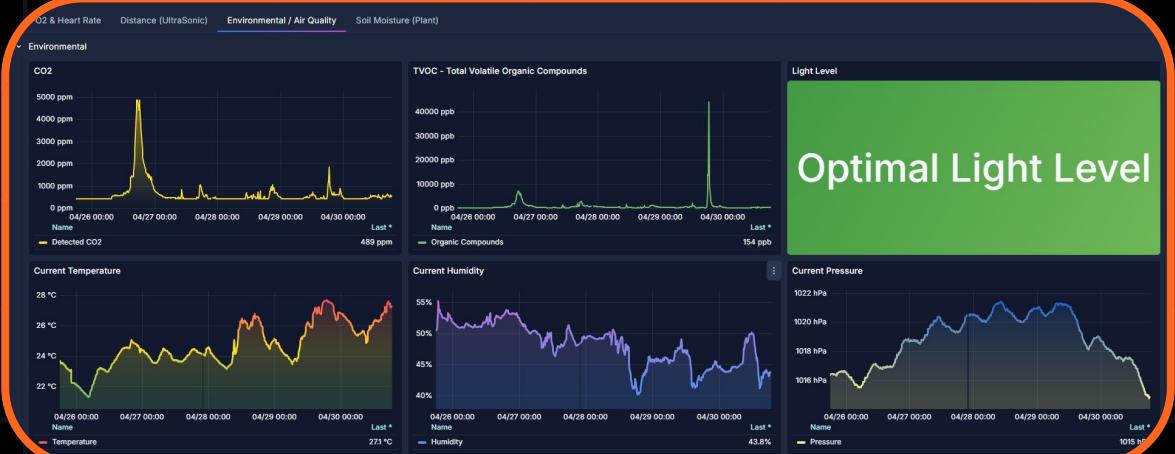


ENV III

Provides temperature, humidity and pressure metrics

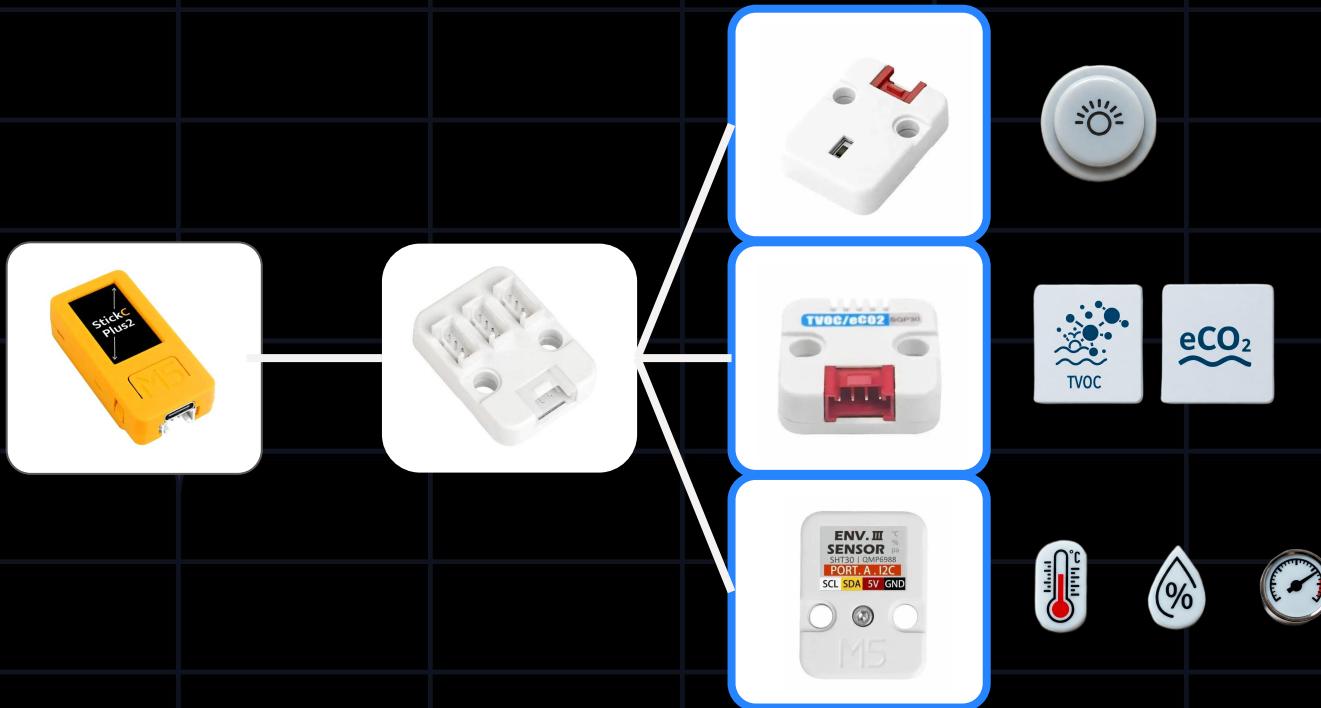
What's running at our booth

- An M5Stick microcontroller and different sensors
- Uploaded our example code to the M5Stick
- Monitor sensor data



Assembly instructions

Connect the **DLIGHT**, **TVOC/eCO₂**, and **ENV III** sensors to the **expansion unit** in any order, then connect it to the **M5Stick**



Assembly instructions

Connect the EARTH sensor to the M5Stick



Assembly instructions

Connect the **Sonic distance sensor** to the **M5Stick**



Assembly instructions

Connect the **HEART** sensor to the **M5Stick**



Setting up the environment



Install Arduino IDE

- The original IDE for Arduino, but very popular for ESP32 too
- Free!

[Download Arduino IDE](#)



Arduino IDE 2.3.1

The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger.

For more details, please refer to the [Arduino IDE 2.0 documentation](#).

Nightly builds with the latest bugfixes are available through the section below.

SOURCE CODE

The Arduino IDE 2.0 is open source and its source code is hosted on [GitHub](#).

DOWNLOAD OPTIONS

Windows Win 10 and newer, 64 bits
Windows MSI installer
Windows ZIP file

Linux AppImage 64 bits (X86-64)
Linux ZIP file 64 bits (X86-64)

macOS Intel, 10.14: "Catalina" or newer, 64 bits
macOS Apple Silicon, 11: "Big Sur" or newer, 64 bits

[Release Notes](#)



Linux drivers

Most modern Linux distributions already have built-in support for the devices. However, if using Linux you may need to:

- Install the following Python library:
`pip install pyserial`
- Add your user to the dialout group (you may need to reboot)
`sudo usermod -aG dialout $USER`

[Troubleshooting](#)



macOS drivers

If you are using MacOS, install the developer tools:

- Open a terminal window and type the following:
 - If you have an Apple Silicon: `softwareupdate --install-rosetta`
 - For all macOS computers: `xcode-select --install`
 - Reboot

[Troubleshooting](#)



Windows drivers

When you first connect an Arduino device, Windows may automatically attempt to install the necessary drivers. It might also prompt you for permission to complete the installation.

- If it doesn't, [download](#) and install the *CP210x* driver.
- You may need to restart your computer

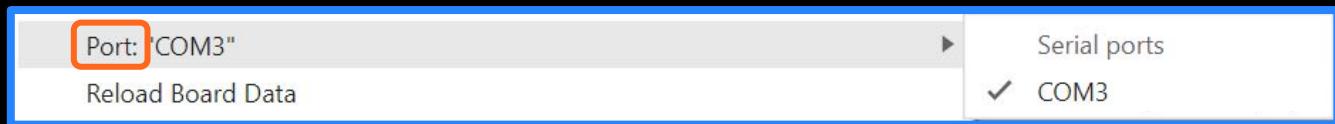
[Troubleshooting](#)

```
Output
Ethernet@2.0.2
Installing Ethernet@2.0.2
Installed Ethernet@2.0.2
Downloading Keyboard@1.0.6
Keyboard@1.0.6
Installing Keyboard@1.0.6
Installed Keyboard@1.0.6
```

Test the connection

- Now that you've installed Arduino, go ahead and plug in your M5Stick to your computer.
- Check if it's recognized and appears as a serial device in the Arduino IDE's **Tools > Port** menu.

Note: each OS has its own virtual serial port name (COM, /dev/tty/, /dev/cu.usbserial, etc)



Install dependencies: boards

A **board** is the software profile that matches your physical hardware, defining how code should be compiled and uploaded.

Some boards (like ESP32) are not included by default in Arduino IDE

Open the Preferences window:

Linux: File > Preferences

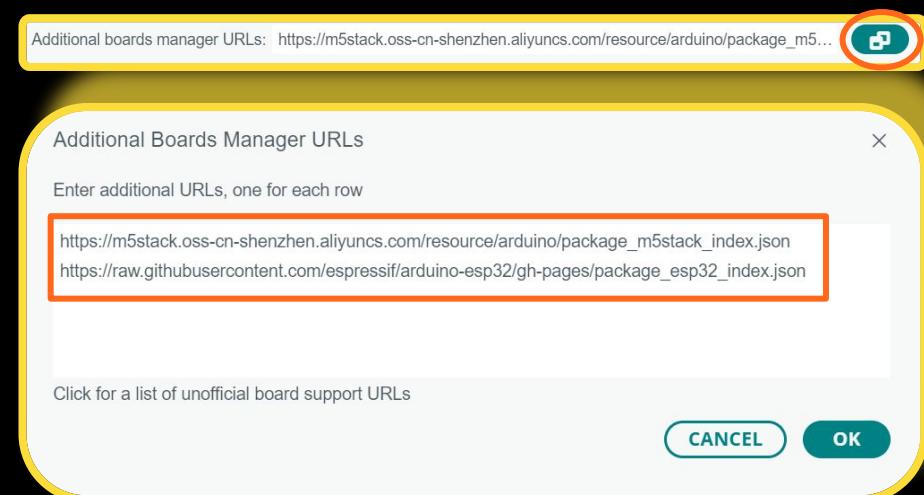
MacOS: Arduino IDE > Settings / Settings

Windows: File > Preferences

Add the following **Additional boards manager URLs:**

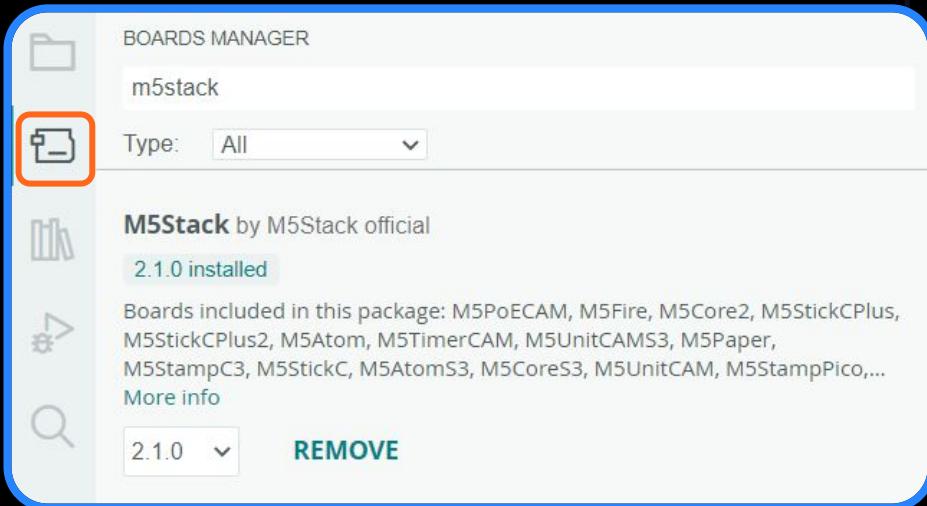
https://m5stack.oss-cn-shenzhen.aliyuncs.com/resource/arduino/package_m5stack_index.json

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



Install dependencies: boards

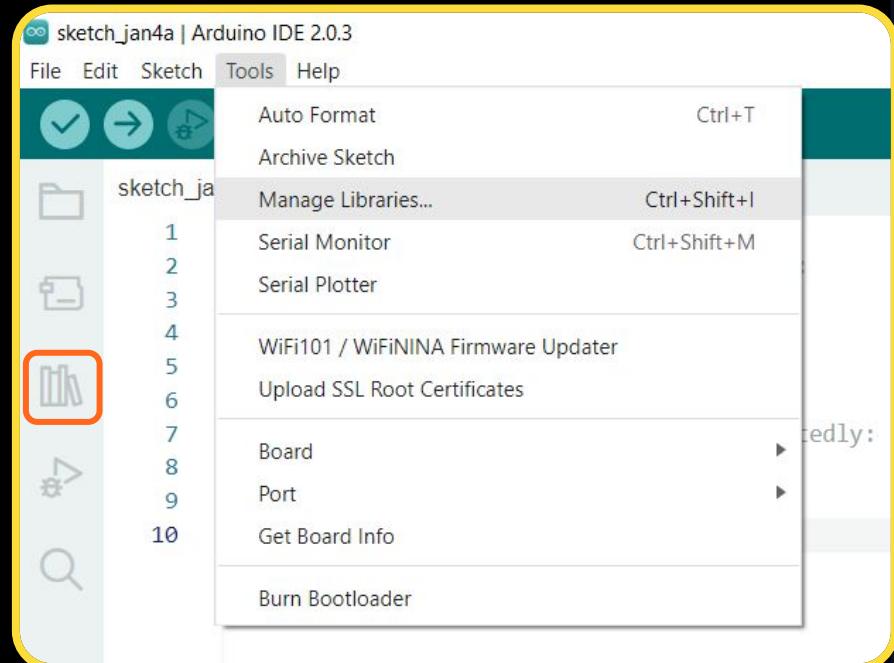
- Open Board Manager
 - Tools > Board > Boards Manager, or click on the board icon
- Search for: M5Stack
- When you find the board, click **INSTALL** (it might take a while)



Install dependencies: libraries

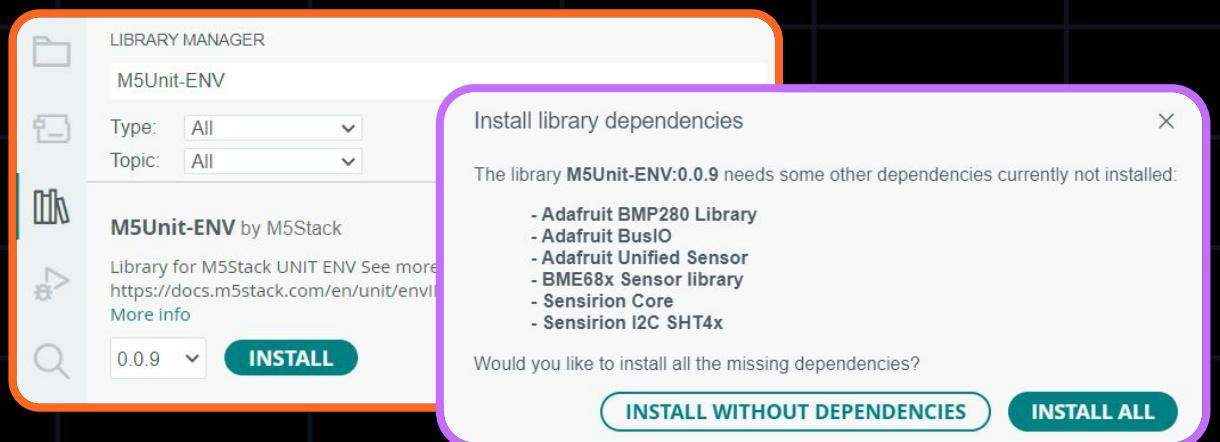
Next we'll install libraries (reusable collections of code) that are needed for the remainder of the workshop

- Open the Library Manager: **Tools** → **Manage Libraries** or click on the library icon



Install dependencies: libraries

- In Arduino Library Manager, for each of the following libraries:
M5Unified, M5UnitENV, M5StackCplus, Adafruit_SGP30, M5_DLIGHT, PromLokiTransport, PrometheusArduino, Unit_Sonic, MAX30105 (Heart Rate)
 - Search by name
 - Click **Install**
 - Then click **Install All** to include dependencies
- Repeat for **each library**.



Grafana Cloud



Create Grafana Cloud account

Go to <https://grafana.com>
and sign up for a free account (if you do not have one already)



The (actually useful) free forever plan

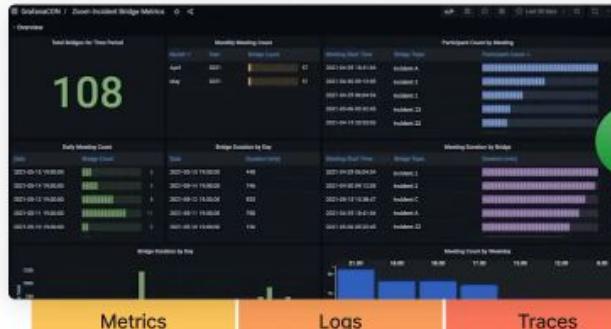
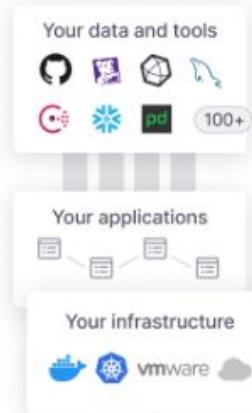
Grafana, of course + 10K series

Prometheus metrics + 50GB logs +

50GB traces + 500VUh k6 testing

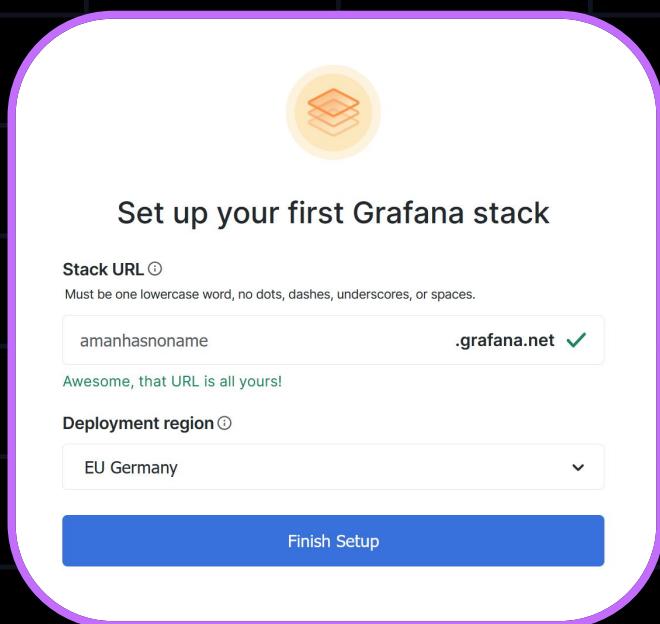
[Create free account](#)

(No credit card required)



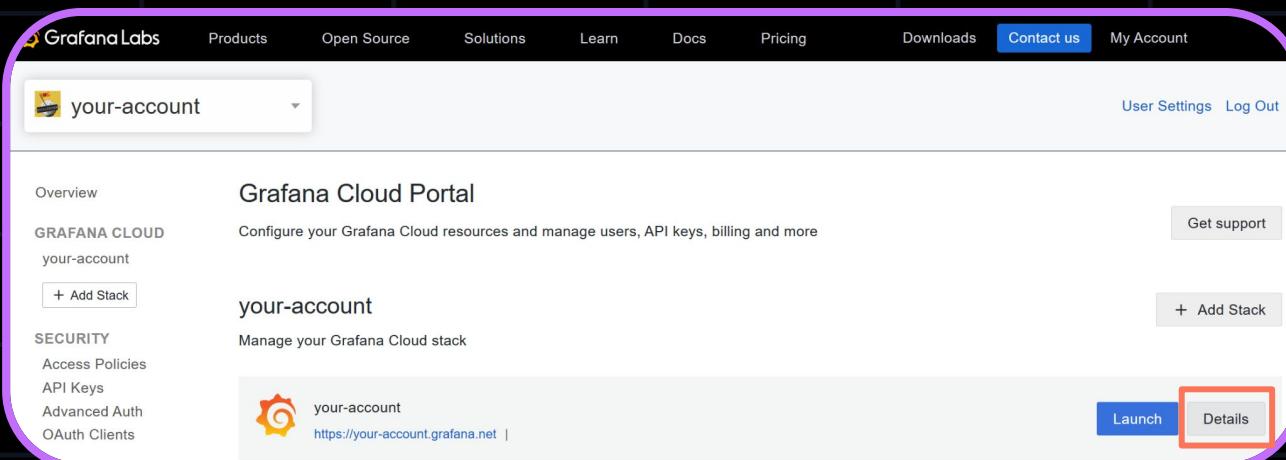
Create Grafana Cloud account

- Create a unique url for this project, easiest is to have your name in it
- Select a region. Any will do!



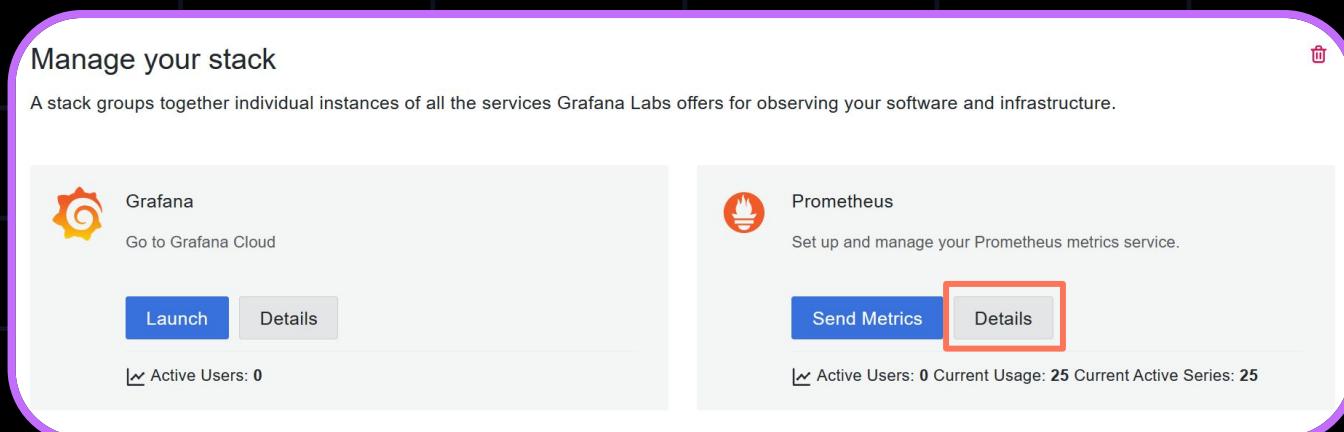
Get your credentials from Grafana Cloud

- Navigate to grafana.com > My Account
- In your Grafana Cloud Portal, open up the **details** of your Grafana Instance



Get your credentials from Grafana Cloud

- Open up the details of your Prometheus instance



Get your credentials from Grafana Cloud

- Make a note of your username, you will put this in your `config.h` file later on

Username / Instance ID

Your Grafana Cloud Prometheus username.

283982

[Copy to Clipboard](#)

Get your credentials from Grafana Cloud

- Get your remote write URL

Remote Write Endpoint

Use this URL to **send** Prometheus metrics to Grafana Cloud.

<https://prometheus-prod-01-eu-west-0.grafana.net/api/prom/push>

 [Copy to Clipboard](#)

- Copy just the main part of the URL: prometheus-prod-01-eu-west-0.grafana.net

Get your credentials from Grafana Cloud

- Generate a new API key
- Give it a name
- Click on Create Token

Password / API Key

Your Grafana Cloud API Key. Be sure to grant the key a role with metrics push privileges. [Generate now](#)

Create an API token X

Access policy name
Autogenerated access policy name.

Token name
Enter a descriptive name to identify this token.

Scopes
Predefined scopes for this token.

This token is generated using [Cloud Access Policies](#). You can [create a new token](#) manually in case you need to customize predefined scopes.

Get your username and token from Grafana Cloud

- Copy the API token and keep it somewhere safe, after you close this window, you won't be able to access it anymore.

Create an API Key

API Key created

Your token has been successfully issued.

```
eyJrIjoiMjgwMzRiMWFlY2MwI  
iLCJpZCI6NDQyMzQxfQ==
```

ya3Nob3A

[Copy to Clipboard](#)

Note: Once closed, you won't see this token again. However, you will see a list of issued tokens and you can revoke them at any time.

[Close](#)

Get your credentials from Grafana Cloud

- Repeat the process to obtain the details of your other InfluxDB instance (remote write URL, instance ID, and token).

Manage your stack

A stack groups together individual instances of all the services Grafana Labs offers for observing your software and infrastructure.

 **Grafana**
Go to Grafana Cloud

[Launch](#) [Details](#)

Active Users: 0

 **Prometheus**
Set up and manage your Prometheus metrics service.

[Send Metrics](#) [Details](#)

 **InfluxDB connectivity**
Connect using InfluxDB protocol.

[Configure](#)

Importing the project



Clone the IOT Workshop repository

- Find your Arduino IDE sketchbook location:
 - Linux: File > Preferences > Sketchbook location
 - Default location: `~/Arduino`
 - MacOS: Arduino IDE > Settings (or Preferences) > Sketchbook location
 - Default location: `/Users/USERNAME/Documents/Arduino`
 - Windows: File > Preferences > Sketchbook location
 - Default location: `C:\Users\USERNAME\Documents\Arduino`
- Open a terminal window (Linux, MacOS) or a Git CMD window (Windows) and clone the IOT workshop repository:
`cd SKETCHBOOK_LOCATION`
- Clone the workshop repository:
`git clone https://github.com/grafana/GrafanaCon2025-ScienceFair-IoT.git`
 - As an alternative, you can also download the .zip file from the Git repo

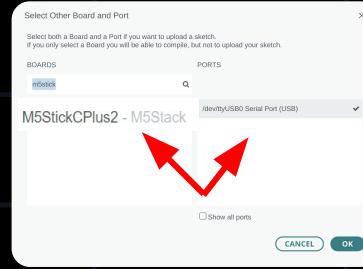
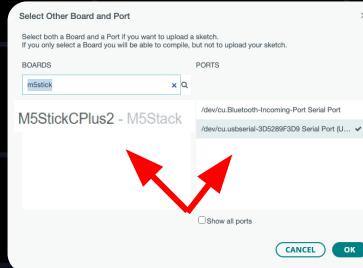
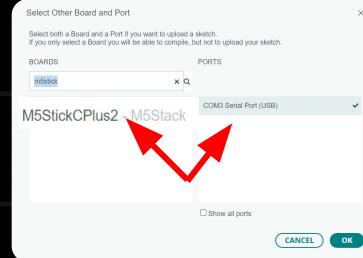
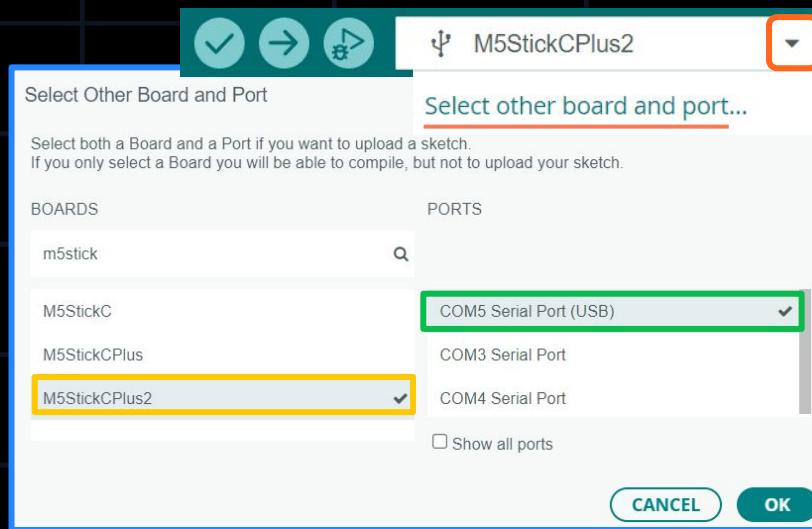
Open the workbook in the Arduino IDE

In the Arduino IDE, open the `.ino` file located inside each of the separate project folders (e.g., `EarthMoisture.ino`)



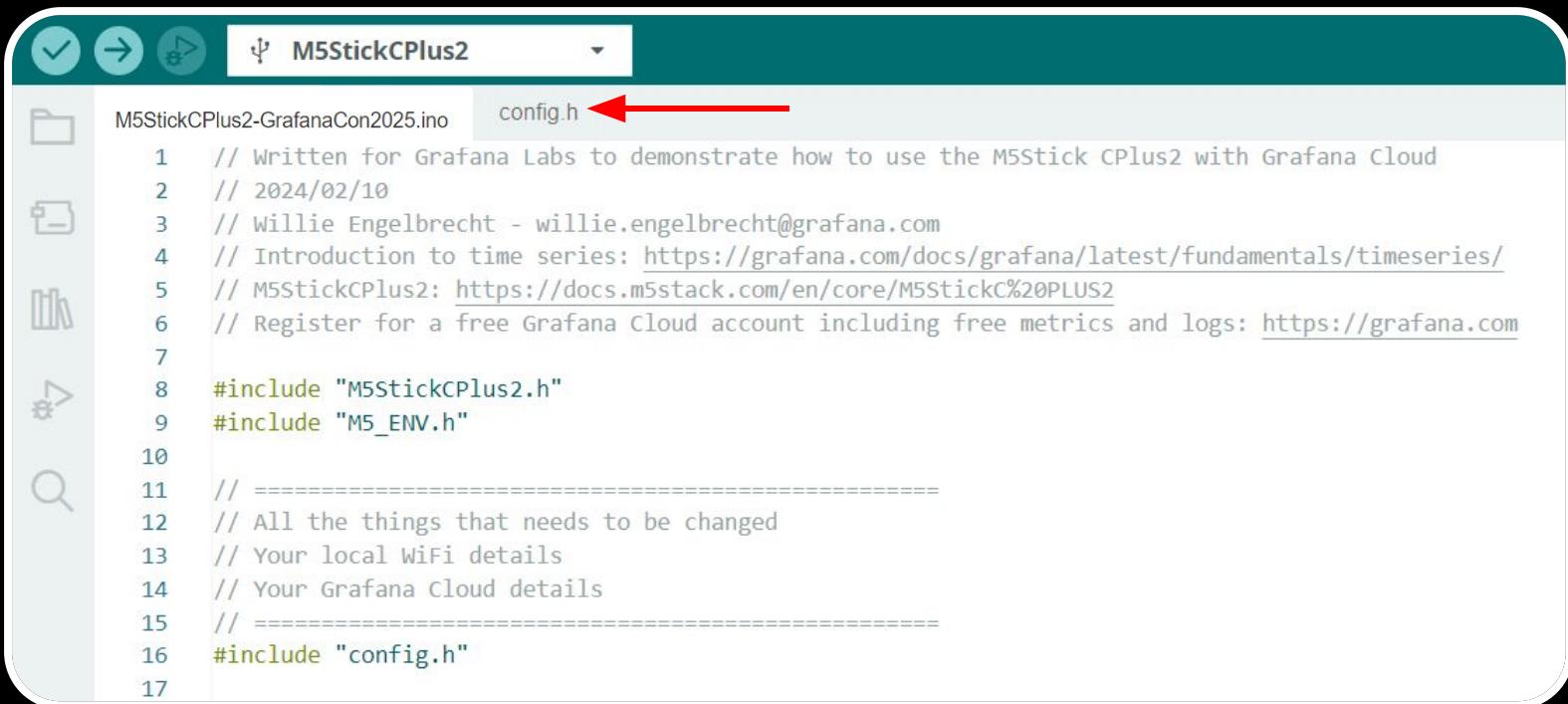
Select the M5Stick Board

- Connect your M5StickCPlus2 (with the sensor) to the USB port on your computer
- Click **Select Board > Select other board and port**
- Under BOARDS, search for “M5StickC”
- Select **M5StickCPlus2**, select the correct port, and click **OK**



Configure the M5Stick application

- Click the config.h tab



```
M5StickCPlus2-GrafanaCon2025.ino config.h
1 // Written for Grafana Labs to demonstrate how to use the M5Stick CPlus2 with Grafana Cloud
2 // 2024/02/10
3 // Willie Engelbrecht - willie.engelbrecht@grafana.com
4 // Introduction to time series: https://grafana.com/docs/grafana/latest/fundamentals/timeseries/
5 // M5StickCPlus2: https://docs.m5stack.com/en/core/M5StickC%20PLUS2
6 // Register for a free Grafana Cloud account including free metrics and logs: https://grafana.com
7
8 #include "M5StickCPlus2.h"
9 #include "M5_ENV.h"
10
11 // =====
12 // All the things that needs to be changed
13 // Your local WiFi details
14 // Your Grafana Cloud details
15 // =====
16 #include "config.h"
17
```

Configure the M5Stick application*

- Enter your WiFi SSID and password
- Enter your Grafana URL, User, and API Key

```
M5StickCPlus2.ino  certificates.h  config.h  config.h.home.h
1 // Set your local WiFi username and password. Please use a 2.4GHz access point
2 #define WIFI_SSID   ""
3 #define WIFI_PASSWORD ""
4
5 // Put you name in below, and the location where you will place the sensor
6 #define YOUR_NAME ""
7
8 // Comma-separated Prometheus labels (key=value) to apply to the metrics from the sensor
9 // Example: site=home,location=study
10 //
11 // Adding labels might require you to increase the size of the WriteRequest buffer in M5StickCPlus.ino
12 #define PROM_LABELS "site=home,location=study"
13
14 // For more information on where to get these values see: https://github.com/grafana/diy-iot/blob/main/README.md
15 #define GC_URL ""
16 #define GC_PATH "/api/prom/push"
17 #define GC_PORT 443
18 #define GC_USER ""
19 #define GC_PASS ""
20
21 // Set to 1 to show debug information on the LCD screen, or 0 to not display
22 // The debug information will show if you are able to connect to the WiFi or not, and whether writing the
23 #define LCD_SHOW_DEBUG_INFO "1"
```

The M5Stick only supports 2.4Ghz WiFi. Make sure you are connecting to the 2.4 GHz band on your WiFi.

Let's walk through the sketch

Environmental/Environmental.ino

1

```
#include "config.h"
```

2

```
// Define all our timeseries
TimeSeries ts_m5stick_temperature(1, "m5stick_temp", PROM_LABELS);
TimeSeries ts_m5stick_humidity(1, "m5stick_hum", PROM_LABELS);
TimeSeries ts_m5stick_pressure(1, "m5stick_pressure", PROM_LABELS);
TimeSeries ts_m5stick_bat_volt(1, "m5stick_bat_volt", PROM_LABELS);
TimeSeries ts_m5stick_bat_current(1, "m5stick_bat_current", PROM_LABELS);
TimeSeries ts_m5stick_bat_level(1, "m5stick_bat_level", PROM_LABELS);
TimeSeries ts_m5stick_voc(1, "m5stick_voc", PROM_LABELS);
TimeSeries ts_m5stick_co2(1, "m5stick_co2", PROM_LABELS);
TimeSeries ts_m5stick_lux(1, "m5stick_lux", PROM_LABELS);
```

3

```
void setup() {
```

4

```
void loop() {
    // Get new updated values from our sensor
    if (qmp6988.update()) {
        pressure = qmp6988.calcPressure();
    }
    if (sht30.update()) {      // Obtain the data
        temp = sht30.cTemp;    // Store the temperature
        hum = sht30.humidity; // store the humidity
    } else {
        temp = 0; hum = 0;
    }
    if (!sgp.IAQmeasure()) { // Comm with SGP
        Serial.println("eCO2/VOC Measure Failed");
        //ESP.restart();
        //return;
    }
    // Now send all of our data to Grafana Cloud!
    PromClient::SendResult res = client.send(req);
```



Let's walk through the sketch

EarthMoisture/EarthMoisture.ino

1

```
#include "config.h"
```

2

```
void loop() {
    int soilMoisture = analogRead(soil_moisture_pin);
    int soilMoisturePCT = map(soilMoisture, 1000, 4095, 0, 100);
```

3

```
// Send to Grafana cloud
String postData = "";
postData = "m5Soil,location=home moisture=" +
int httpResponseCode = http.POST(postData);
```

Let's walk through the sketch

Heartrate/Heartrate.ino

1

```
#include "config.h"
```

2

```
while (1) {  
    red = Sensor.getRed();  
    ir = Sensor.getIR();
```

Let's walk through the sketch

Ultrasonic/Ultrasonic.ino

1

```
#include "config.h"
```

2

```
void loop() {
    // put your main code here, to run repeatedly:
    static float red = 0.0;
```

```
    red = sensor.getDistance();
    Serial.println("Distance measured: " + String(red));
```

3

```
String postData = "";
postData = "m5UltraSonic,location=home distance=" + String(message.distance);
```

4

```
for (int i = 0; i < 240; i++) {
    red_disdata = map(line[0][(red_pos * 240) + i], 0, 255, 0, 255);
    Disbuff.drawLine(i, red_last, i + 1, red_last);
    red_last = red_disdata;
}
```

```
Disbuff.setTextSize(2);
Disbuff.setTextColor(GREEN);
Disbuff.setCursor(5, 5);
Disbuff.printf("Dist:%0.2fmm", red);
```

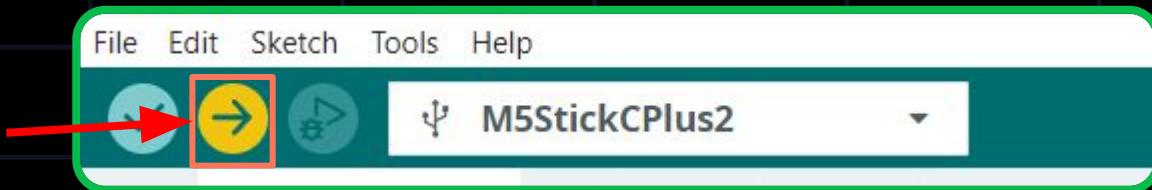
```
int httpResponseCode = http.POST(postData);
```

Compiling the code and uploading the firmware



Compiling and uploading

- On each project: to **compile and upload**, click the “right arrow”



- And on the bottom right of your screen, it should say **Compiling sketch**
This can take a few minutes the first time you run it



Compiling and **uploading**

- After a while, it will start **uploading** to your device

Output

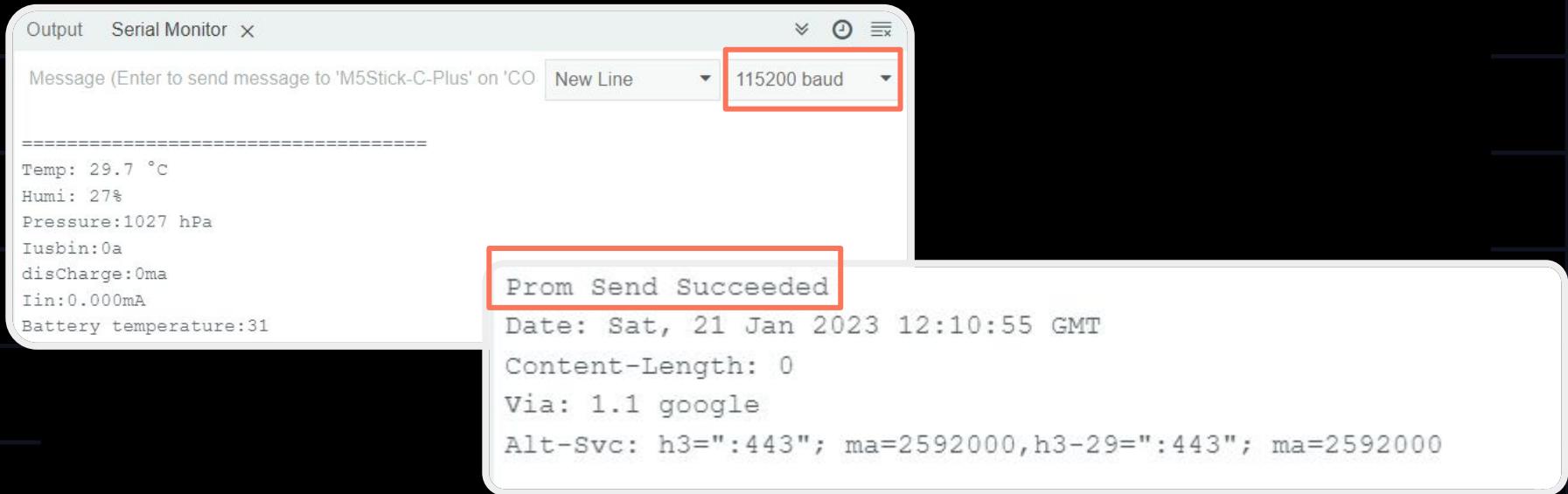
```
Wrote 8192 bytes (47 compressed) at 0x0000e000 in 0.1 seconds (effective 584.1 kbit/s)...
Hash of data verified.
Compressed 943200 bytes to 600927...
Writing at 0x00010000... (2 %)
Writing at 0x0001a4d8... (5 %)
Writing at 0x0002667c... (8 %)
Writing at 0x0003148c... (10 %)
Writing at 0x00040884... (13 %)
Writing at 0x0004648f... (16 %)
Writing at 0x0004b95... (18 %)
Writing at 0x00051430... (21 %)
Writing at 0x00056ab7... (24 %)
Writing at 0x0005c1f8... (27 %)
Writing at 0x000613e7... (29 %)
Writing at 0x000666ff... (32 %)
Writing at 0x0006bab1... (35 %)
Writing at 0x00070c43... (37 %)
Writing at 0x00075f91... (40 %)
Writing at 0x0007b460... (43 %)
Writing at 0x0008080f... (45 %)
Writing at 0x00085917... (48 %)
Writing at 0x0008aa49... (51 %)
Writing at 0x000906f8... (54 %)
Writing at 0x00095cae... (56 %)
Writing at 0x0009b1eb... (59 %)
Writing at 0x000a64ab... (62 %)
Writing at 0x000a57b6... (64 %)
Writing at 0x000aabec... (67 %)
```

- When the upload is complete the Output will show: **Hard resetting via RTS pin**, and the M5StickCPlus will reboot.



Serial Monitor

- You can also inspect console print outs from the serial monitor
- Select Tools -> Serial Monitor, and change the baud to 115200
- Ensure you get a Prom Send Succeeded



LCD Output

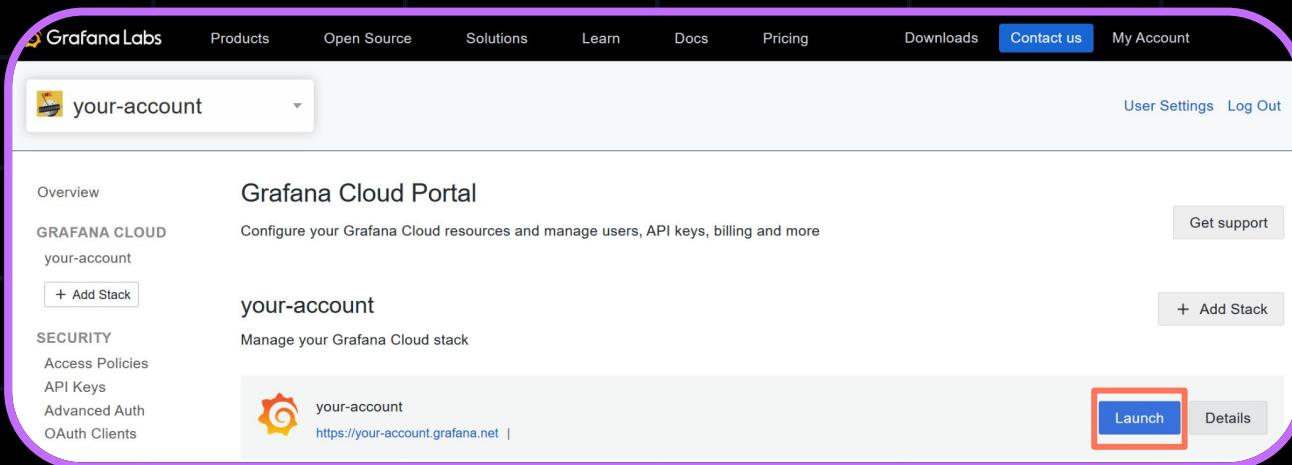
- The LCD Screen should also display the values being measured



Explore your data in Grafana

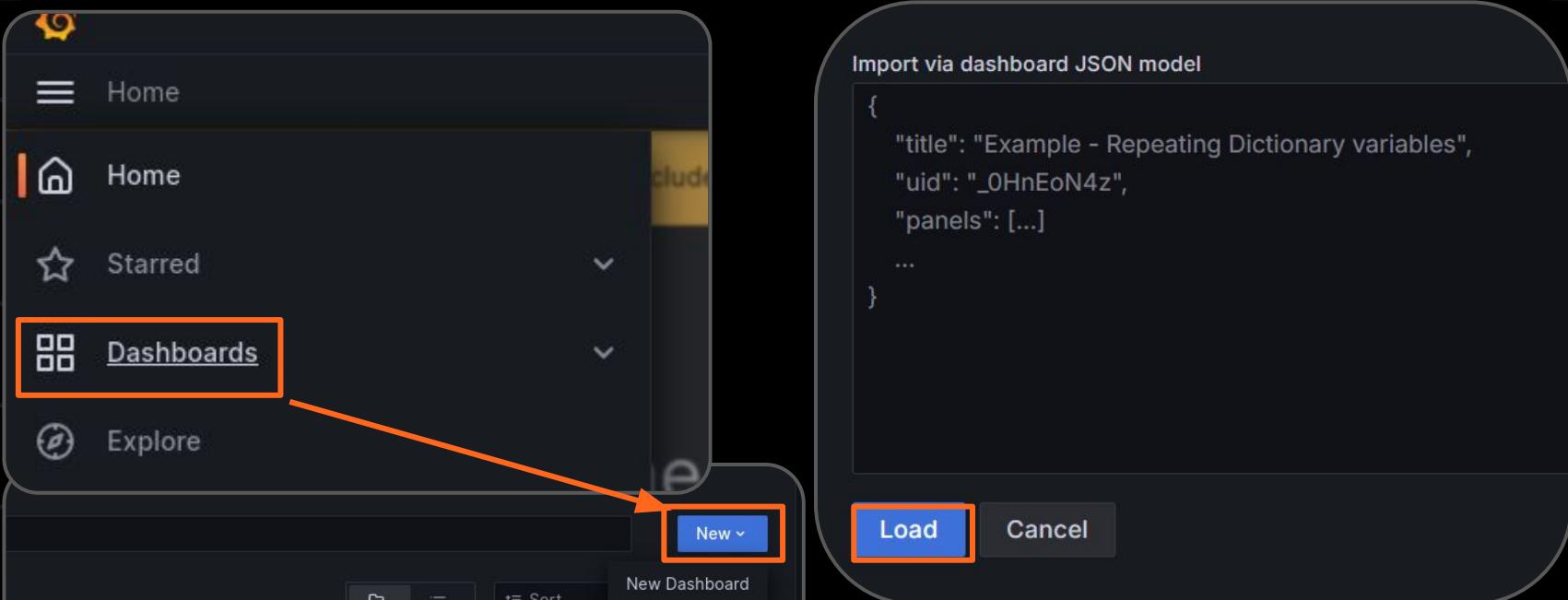
Setting up the **Grafana** Dashboard

- Go to grafana.com > My Account > click **Launch**



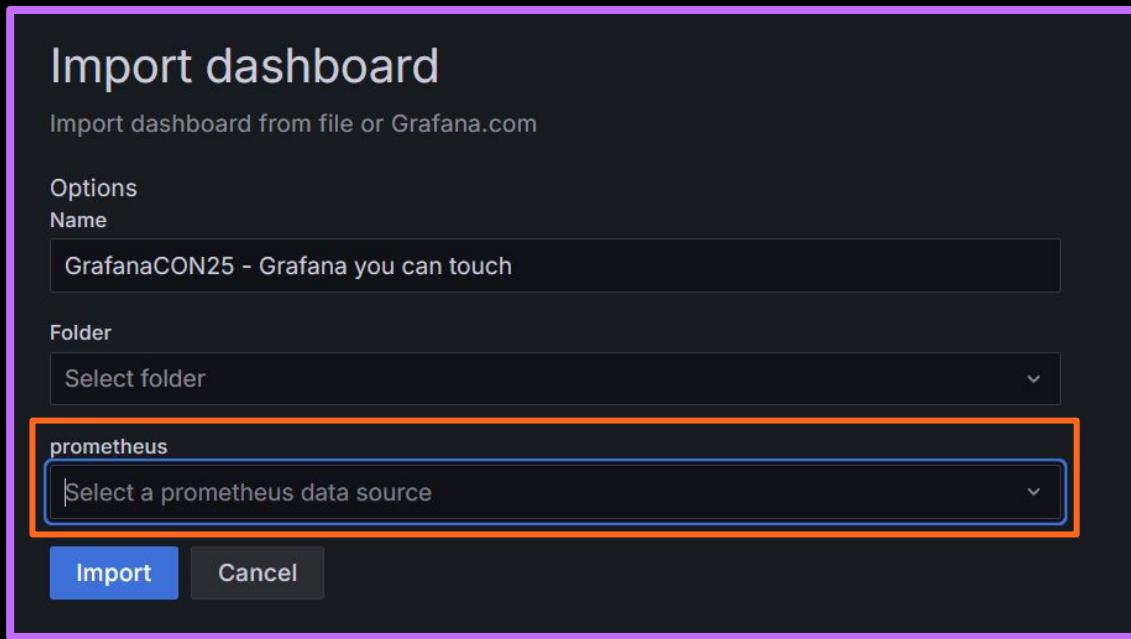
Setting up the **Grafana** Dashboard

- Copy the content of dashboards/**GrafanaCON25-grafana-you-can-touch.json**
- In Grafana, import our dashboard by navigating to Dashboard > Import.
- Paste in the dashboard json
- Click **Load**

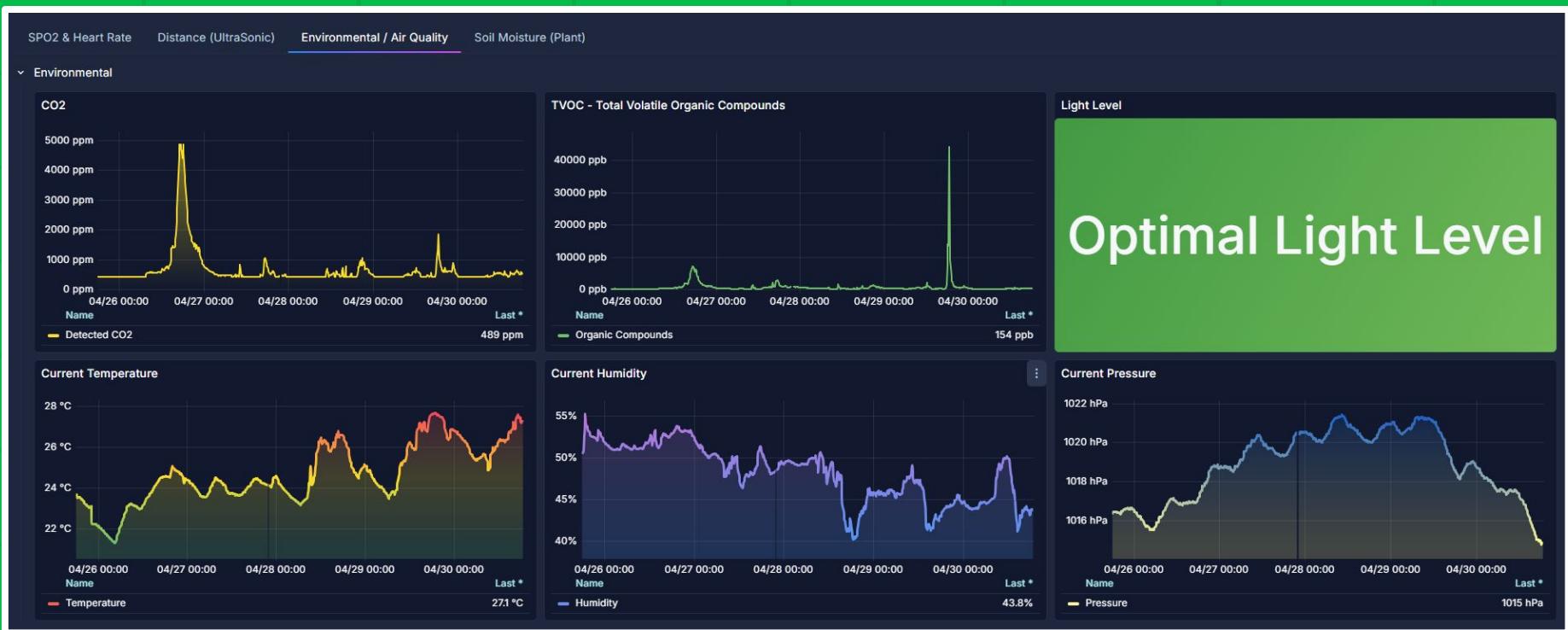


Setting up the **Grafana** Dashboard

- Select your Grafana Cloud **Prometheus** datasource.
- Click Import



You did it!





Troubleshooting

If the device is **not recognized**:

- **Linux** and **MacOS**: see if the driver is listed :

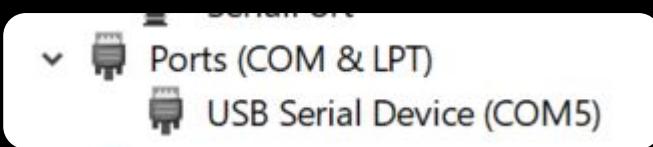
`ls /dev/tty*`

Look for `/dev/ttyUSB0`, `/dev/ttyACM0`, or

`/dev/tty.wchusbserial-XXXX`.

If not listed, [download](#) and install the *CP210x* driver.

- **Windows**: Check Device Manager under Ports (COM & LPT)



✗ Still nothing?

✓ Make sure you're using a data cable – some cables are power-only and won't transmit data.

Try swapping the cable.



Troubleshooting

- On **MAC**, ensure you have devtools installed or else you may not be able to access the COM port (./dev/tty on MacOS/Linux)
If you have an Apple Silicon: `softwareupdate --install-rosetta`
 - For all Mac computers: `xcode-select --install`
 - Reboot
- Still can't see any of the M5 devices in the COM Port list? Change to a different USB-C cable. Don't use an Apple Mac supplied USB-C cable. It should be a non Apple branded cable, but a cable capable of transferring data.
- On **Windows**, if the virtual COM port isn't detected, it probably means that the drivers for CP210x need to be manually installed.

[Back to Test the connection](#)

Troubleshooting

- Uploading on **Linux**
 - "Could not open /dev/ttyACM0, the port doesn't exist"
 - add your user to the dialout group to access serial devices
 - `sudo usermod -aG dialout $USER`
 - Reboot machine and then check if your user is in the group:
`groups`
 - "No serial data received" error.
 - Lower the upload_speed to 115200 or 57600 by adding the following to `platformio.ini`
`upload_speed = 115200`
`monitor_speed = 115200`
 - Hold the BOOT button while flashing.

Troubleshooting

- .ino file requires the folder where it's contained has the same name. E.g. **M5StickCPlus2-GrafanaCon2025**.ino requires to be in a folder named M5StickCPlus2-GrafanaCon2025
- Only ==**Grafana Labs**== shows on the little LCD screen, nothing else: Something is wrong with your WiFi credentials and the M5Stick is not able to connect. Try restarting the M5Stick or check your credentials
- **httpResponseCode 500:** Something is wrong with your Grafana API username/token/URL. For example, ensure that you don't have **https://** or **/api/prom** in the URL. Remove these.
- **{"status":"error","error":"authentication error: invalid scope requested"}**
 - Ensure you have “Write” access enabled on your token for your Prometheus instance.
- Inside the Arduino IDE when you compile, you get an error that says it is not able to find **M5StickCPlus.h** You forgot to install some of the M5 libraries ([slide 20 in this deck](#))