🌿 CO_2 Monitoring Device — Full Assembly and Setup Guide

Version 3.1 – 2025

---

1. Overview

This guide explains how to build, program, and set up four versions of a CO_2 monitoring device that uploads live data to a Firebase database and displays it through a web dashboard.

Supported hardware configurations:

1. K30 Sensor + Arduino Uno R4 WiFi
2. SCD30 Sensor + Arduino Uno R4 WiFi
3. K30 Sensor + ESP32 Dev Board
4. SCD30 Sensor + ESP32 Dev Board

---

2. Ordering the Parts

You will need one set of the following for your chosen version:

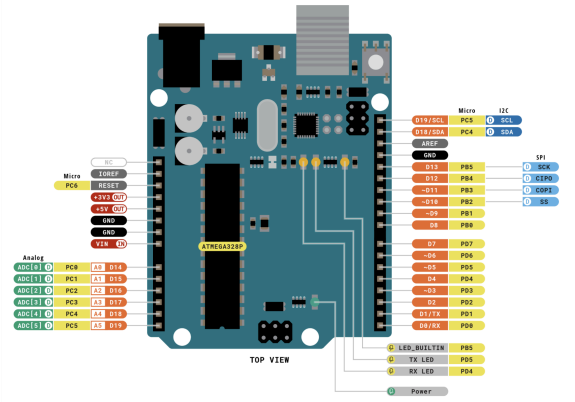
Part	Description	Example Source
Arduino Uno R4 WiFi	Wi-Fi enabled microcontroller	Arduino.cc or Amazon

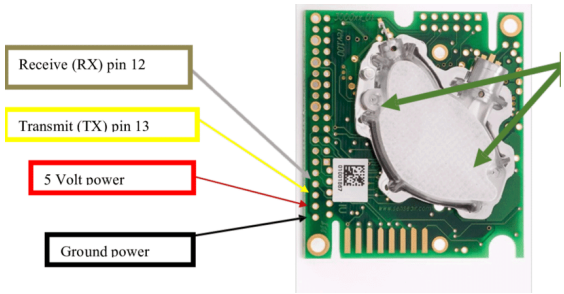
ESP32 Dev Board (WROOM-32)	Wi-Fi microcontroller	Amazon, Adafruit
K30 CO_2 Sensor	NDIR CO_2 sensor (5V logic)	CO2meter.com
SCD30 CO_2 Sensor	I²C CO_2 sensor (3.3–5V logic)	SparkFun or Adafruit
Jumper wires	Male-to-female	Amazon
Breadboard	For temporary connections	Amazon
USB cable	USB-A to Micro USB (ESP32) or USB-C (Uno R4 WiFi)	Included with board

---

3. Connecting the Parts

3.1 K30 Sensor → Arduino Uno R4 WiFi



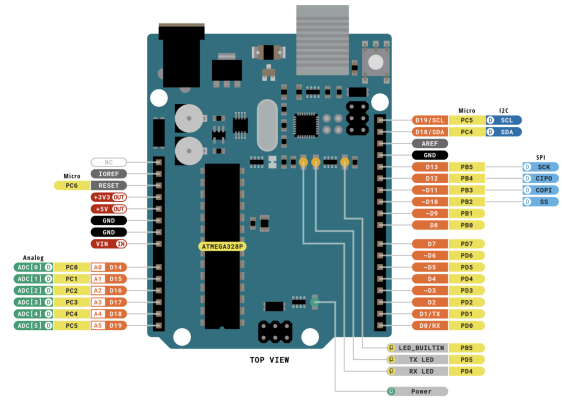


K30 Pin	Uno R4 WiFi Pin
RX	D12
TX	D13
5V	5V

GND	GND

(Use SoftwareSerial on pins 12 and 13)

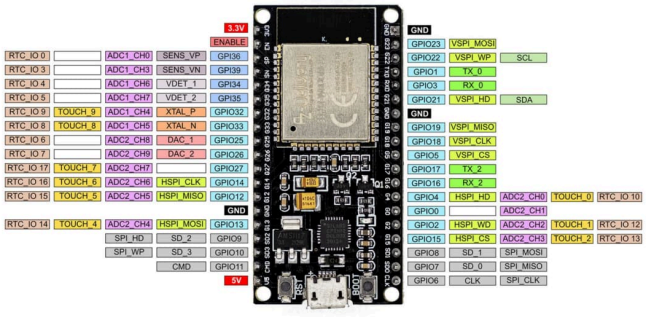
3.2 SCD30 Sensor → Arduino Uno R4 WiFi

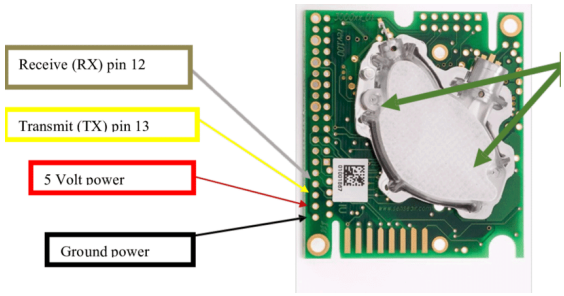


SCD30 Pin	**Uno R4 WiFi Pin**
VIN	5V
GND	GND
SCL	A5
SDA	A4

(I²C protocol – no SoftwareSerial needed)

3.3 K30 Sensor → ESP32 Dev Board

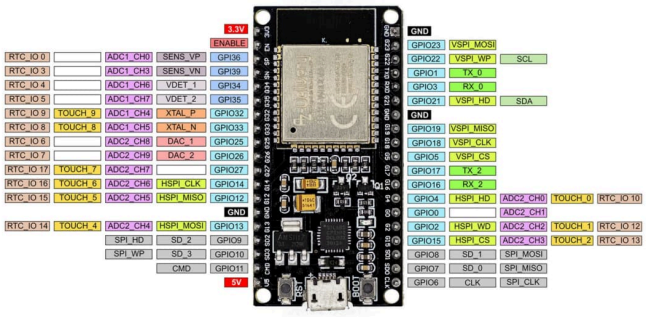




K30 Pin	ESP32 Pin
RX	GPIO12
TX	GPIO13
5V	5V
GND	GND

(Use `SoftwareSerial(12,13)` as shown in code)

3.4 SCD30 Sensor → ESP32 Dev Board



SCD30 Pin	ESP32 Pin
VIN	3.3V or 5V
GND	GND

SCL	GPIO22
SDA	GPIO21

Important Notes

- Double-check voltage:

- ESP32 logic is 3.3V
- K30 and SCD30 tolerate 5V safely.
- Always power through USB when programming.

---

4. Installing the Arduino IDE

1. Go to https://www.arduino.cc/en/software
2. Download **Arduino IDE** for your operating system (Mac, Windows, or Linux).
3. Install and open it.
4. Go to **Tools** → **Board** → **Board Manager**.

   - Install **Arduino Uno R4 Boards** if using Uno R4 WiFi.
   - Install **ESP32 Boards** by searching “ESP32” (by Espressif Systems).
5. Select your board under **Tools** → **Board** and choose the correct COM port under **Tools** → **Port**.

---

5. Installing Required Libraries

In **Arduino IDE** → **Tools** → **Manage Libraries**, search and install the following:

Library	Purpose
`WiFiS3`	Wi-Fi for Uno R4 WiFi
`WiFiSSLClient`	Secure HTTP for Uno R4

`WiFi` or `WiFi.h`	ESP32 Wi-Fi
`ArduinoJson`	JSON encoding for Firebase
`SoftwareSerial`	Serial communication for K30
`NTPClient`	Real-time clock synchronization
`SparkFun_SCD30_Arduino_Library`	For SCD30 sensors

---

6. Uploading the Correct Code

Each device type has its own code file.

1. Download the code below for your hardware setup into a new Arduino sketch.

FINAL-WORKING-V5.ino

2. Update the following fields before uploading:

```
#define WIFI_SSID "YourWiFiName"
#define WIFI_PASSWORD "YourWiFiPassword"

const char* FIREBASE_HOST = "your-firebase-project.firebaseio.com";
const char* FIREBASE_SECRET = "your-database-secret";
const char* DEVICE_ID = "YourDeviceName";
```

3. Click the **checkmark** to verify, then **upload** (right arrow).
4. Watch the Serial Monitor (`Tools → Serial Monitor`) for messages.

---

7. Setting Up Firebase (Secure Mode)

1. Go to https://firebase.google.com → **Go to Console**.
2. Click **Create New Firebase Project**, name it `CO2-Monitor`, and disable Google Analytics.
3. Go to **Build → Realtime Database → Create Database**.

- Choose a region (e.g., `us-central1`).
- Choose **Start in Locked Mode** (NOT test mode).

4. Under **Rules**, replace with:

```
{
  "rules": {
    ".read": true,
    ".write": false,
    "readings": {
      ".read": true,
      ".write": "auth != null"
    },
    "errors": {
      ".read": true,
      ".write": "auth != null"
    }
  }
}
```

5. Go to **Project Settings** → **Service Accounts** → **Database Secrets**.

- Copy your **Database Secret** (used in your Arduino code).

---

8. Setting Up the Web Dashboard

8.1 Requirements

- macOS or Linux preferred.
- Homebrew and Node.js installed.

8.2 Installing Dependencies

Open Terminal and run:

```
/bin/bash -c "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

```
brew install node
```

8.3 Setting Up the Web App

1. Download the provided `co2-dashboard-copy.zip`.

2. Extract it to your `Documents` folder so the path becomes:
   `~/Documents/co2-dashboard-copy/public/index.html`

3. Open the folder in Terminal:

```
cd ~/Documents/co2-dashboard-copy
```

4. Edit `public/index.html` in a text editor.
   Replace the Firebase configuration with your project's details:

```
const firebaseConfig = {
  apiKey: "YOUR_API_KEY",
  authDomain: "YOUR_PROJECT_ID.firebaseapp.com",
  databaseURL: "https://YOUR_PROJECT_ID-default-rtdb.firebaseio.com",
  projectId: "YOUR_PROJECT_ID",
  storageBucket: "YOUR_PROJECT_ID.appspot.com",
  messagingSenderId: "YOUR_SENDER_ID",
  appId: "YOUR_APP_ID"
};
```

5. Save the file.

8.4 Viewing the Dashboard

In Terminal:

```
npm install -g serve
serve public
```

Then open the displayed local URL (e.g., `http://localhost:3000`) in your browser.
Your live CO_2 readings will appear on the dashboard map and chart.

---

9. Maintenance and Tips

- The system auto-reboots every 24 hours to prevent crashes.
- You can monitor Firebase data directly from the Firebase Console under **Realtime Database → readings**.
- To add more devices, copy one of the Arduino sketches and change the `DEVICE_ID` to a unique name.

---

10. Troubleshooting

Problem	Possible Cause	Solution
No Wi-Fi connection	Wrong SSID or password	Double-check credentials
-1 PPM values	Sensor error	Check wiring and sensor power
Dashboard not updating	Cached data	Refresh browser or clear cache
Upload failed	Wrong board or COM port	Re-select board and port in Tools menu
Firebase “Permission denied”	Rules too strict	Ensure `.write` rules allow authenticated writes

---