

# UNIT CO2

SKU:U103



## Description

**UNIT CO2** is a photoacoustic Carbon Dioxide (CO<sub>2</sub>) Unit that will tell you the CO<sub>2</sub> PPM (parts-per-million) composition of ambient air. With a built-in Sensirion **SCD40** sensor, power by Buck converter circuitry, and I2C communication. It has a range of 400~2000 ppm with an accuracy of  $\pm(50 \text{ ppm} + 5\% \text{ of reading})$ . What's more, it supports with temperature and humidity measurement too. Perfect for environmental sensing, scientific experiments, air quality and ventilation studies, and more.

## Product Features

- CO<sub>2</sub> concentration detection
- Temperature and humidity measurement
- I2C communication (Add: 0x62)
- Integrated 5V -> 3.3V Buck converter
- 2x Brick holes
- HY2.0-4P interface

## Include

- 1x UNIT-CO2
- 1x HY2.0-4P cable (20CM)

# Applications

- Environmental sensing
- Air quality
- Ventilation studies

# Specification

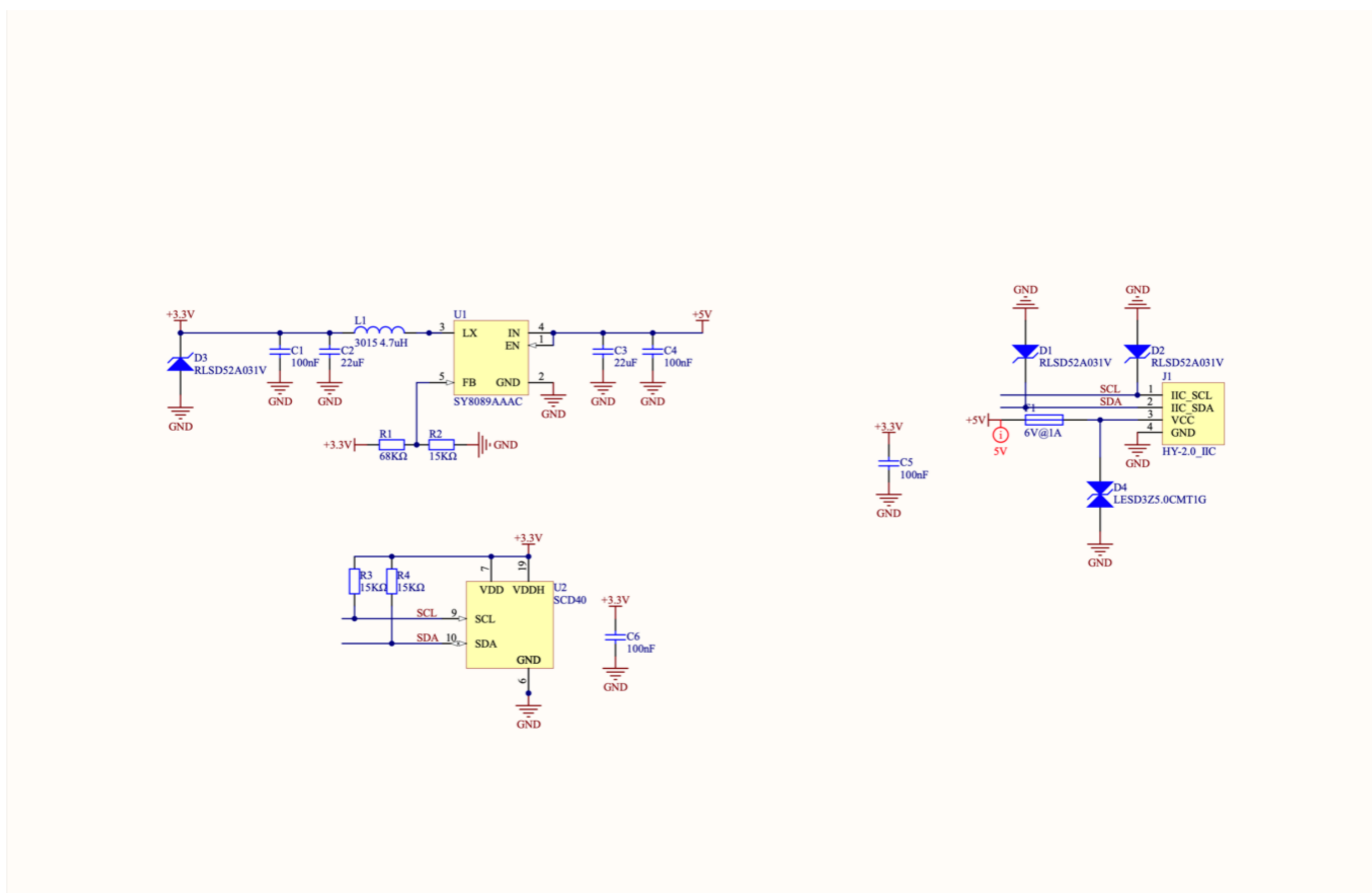
Resources	Parameter
CO2 Measurement range	400 ~ 2000 ppm
CO2 Sampling accuracy	±(50 ppm + 5% of reading)
Temperature range	-10 - 60 °C
Humidity range	0 - 95 %RH
Communication protocol	I2C: 0x62
Net weight	7.54g
Gross weight	13.13g
Product dimensions	48mm x 24mm x 16mm
Package size	134mm x 61mm x 16.3mm
Housing material	Plastic ( PC )



# PinMap

CO2 Unit	SCL	SDA	5V	GND
M5Core(PORT A)	GPIO22	GPIO21	5V	GND
M5Core2(PORT A)	GPIO22	GPIO21	5V	GND
M5Atom(PORT A)	GPIO32	GPIO26	5V	GND
M5StickC/Plus(PORT A)	GPIO33	GPIO32	5V	GND
M5Station(PORT A1,A2)	GPIO33	GPIO32	5V	GND

# Schematic



# Related Link

- **Datasheet**

- [SCD40](#)
- [SY8089AAAC](#)

# Example

---

```
#include <Arduino.h>
#include <Wire.h>

// SCD4x
const int16_t SCD_ADDRESS = 0x62;

void setup() {
  // check in your settings that the right speed is selected
  Serial.begin(115200);
  // wait for serial connection from PC
  // comment the following line if you'd like the output
  // without waiting for the interface being ready
  while (!Serial)
    ;

  // output format

  Serial.println("CO2(ppm)\tTemperature(degC)\tRelativeHumidity(percent)");

  // init I2C
  Wire.begin();

  // wait until sensors are ready, > 1000 ms according to datasheet
  delay(1000);

  // start scd measurement in periodic mode, will update every 5 s
  Wire.beginTransmission(SCD_ADDRESS);
  Wire.write(0x21);
  Wire.write(0xb1);
  Wire.endTransmission();

  // wait for first measurement to be finished
  delay(5000);
}

void loop() {
  float co2, temperature, humidity;
  uint8_t data[12], counter;

  // send read data command
  Wire.beginTransmission(SCD_ADDRESS);
  Wire.write(0xec);
  Wire.write(0x05);
  Wire.endTransmission();
```

```

// read measurement data: 2 bytes co2, 1 byte CRC,
// 2 bytes T, 1 byte CRC, 2 bytes RH, 1 byte CRC,
// 2 bytes sensor status, 1 byte CRC
// stop reading after 12 bytes (not used)
// other data like ASC not included
Wire.requestFrom(SCD_ADDRESS, 12);
counter = 0;
while (Wire.available()) {
    data[counter++] = Wire.read();
}

// floating point conversion according to datasheet
co2 = (float)((uint16_t)data[0] << 8 | data[1]);
// convert T in degC
temperature = -45 + 175 * (float)((uint16_t)data[3] << 8 | data[4]) /
65536;
// convert RH in %
humidity = 100 * (float)((uint16_t)data[6] << 8 | data[7]) / 65536;

Serial.print(co2);
Serial.print("\t");
Serial.print(temperature);
Serial.print("\t");
Serial.print(humidity);
Serial.println();

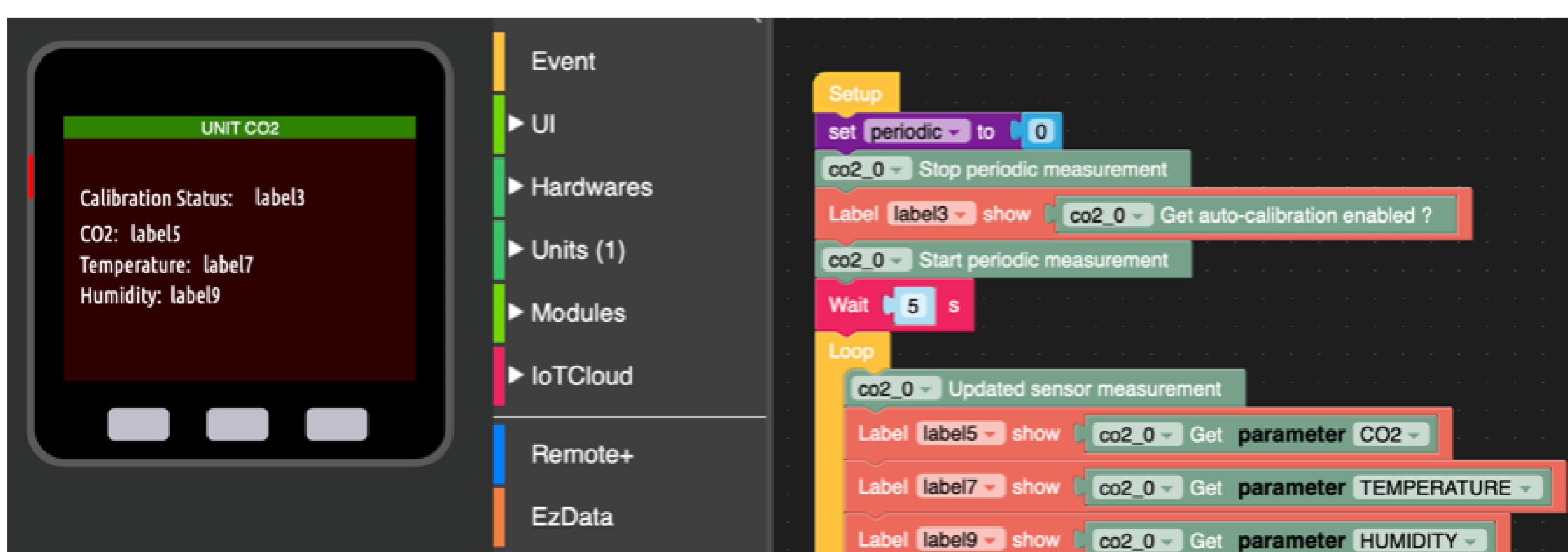
// wait 5 s for next measurement
delay(5000);
}

```

## Arduino

- [M5Unit-CO2 with M5Core](#)
- [M5Unit-CO2 with M5Atom](#)
- [M5Unit-CO2 with M5StickC](#)
- [M5Unit-CO2 with M5StickCPlus](#)

## UIFlow



Units



▶ MediaTrans

Variables

Math