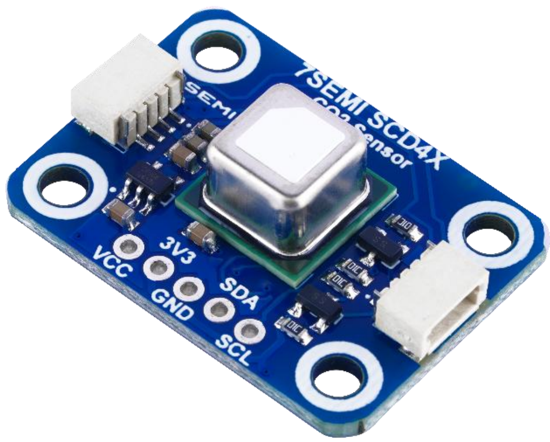
7SEMI

SCD40 CO_2, Temperature and Humidity Sensor Breakout Qwiic I^2C

Version 1.0

Table of Contents

1.Introduction



The **7Semi SCD40 CO_2 Sensor Breakout Qwiic I²C** integrates the **Sensirion SCD40 CO_2 sensor**, offering a compact and efficient solution for real-time environmental monitoring. The sensor provides precise **CO_2, temperature, and humidity** measurements with a small footprint and energy-efficient design.

1.1 Features

- Accurate CO_2 sensor with temperature & humidity compensation.
- Measures CO_2 from 400 to 2000 ppm with fast response.
- Accuracy: ±(50 ppm + 5% of reading).
- Auto & manual calibration for long-term stability.
- Works with 3.3V - 5V power supply.
- I²C interface for easy system integration.
- Compatible with Arduino, ESP32, Raspberry Pi, STM32.
- Qwiic I²C support for plug-and-play use.

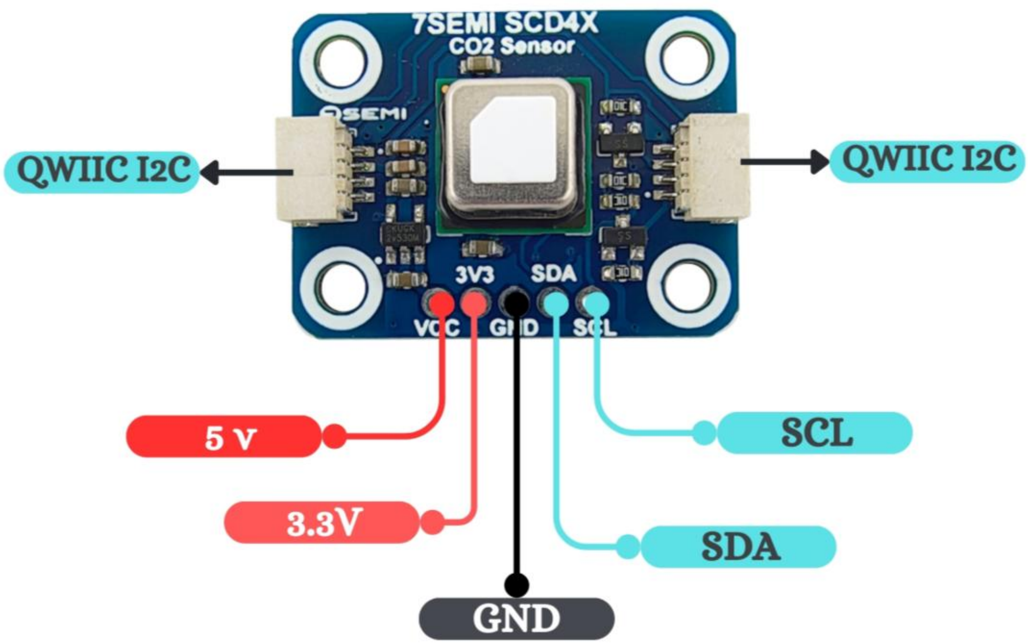
2.Technical Specification

The **Technical Specification** table provides detailed information about the **7Semi SCD40 CO_2 Sensor Breakout Qwiic I²C**, including its operating voltage, current consumption, and electrical characteristics. This data helps users understand the power requirements, communication parameters, and performance capabilities of the sensor. It ensures compatibility with different microcontrollers and embedded systems while providing guidelines for efficient integration into various applications.

SCD-40 Specifications

- Photoacoustic CO_2 sensor technology
- Integrated temperature and humidity sensor
- Measurement range: 400 ppm – 2000 ppm
- Accuracy: ±(50 ppm + 5% of reading)
- Supply Voltage DC : 3.3 or 5 v
- Current consumption:
    - Peak Supply Current in (3.3V is 175mA / 5V is 115 mA)
    - Average Supply Current in (3.3V is 15mA)
- Fully calibrated and linearized
- I²C digital interface address: 0x62
- Sensor Breakout Size: 29.82 x 21.86 mm

3.Pinouts

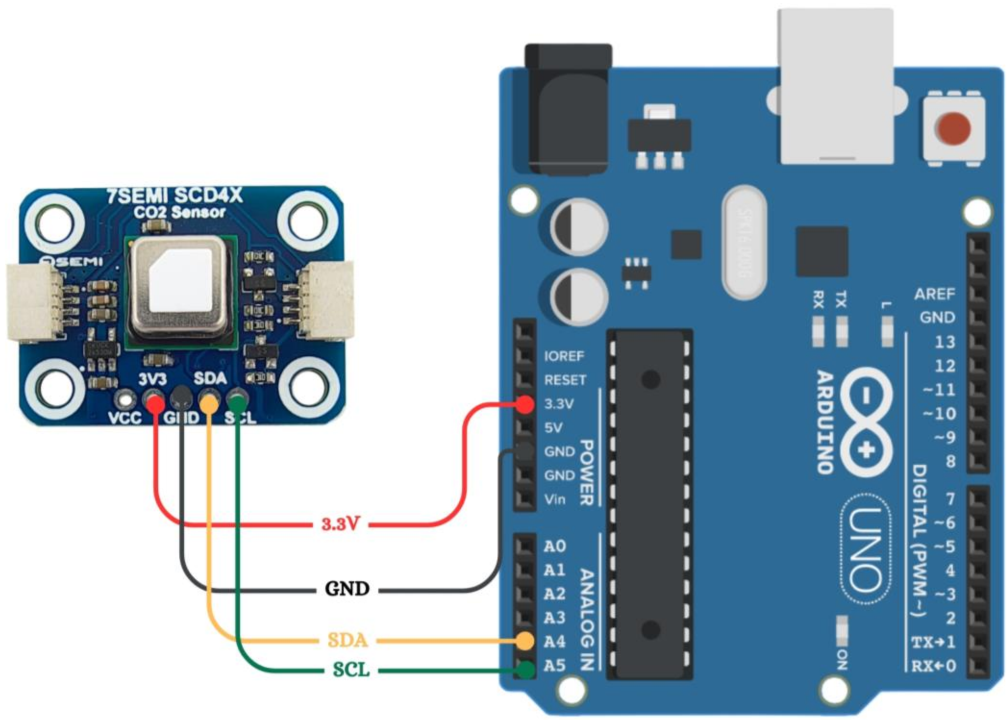


Pin	Name	Description
1	**VCC**	Power Input (5V DC)
2	**3V3**	Power Input (3.3V DC)
3	**SCL**	I^2C Clock Line
4	**SDA**	I^2C Data Line
5	**QWIIC I2C**	I2C communication bus
6	**GND**	Ground Connection

Connection Guidelines

- Ensure a stable 2.3V - 5V DC power source.
- Connect SCL to SCL and SDA to SDA for proper communication.

4.Hardware Interface



Connection Explanation :

Power Connection:

- The **VCC** pin of the sensor is connected to **5V** (or **3.3V**) on the Arduino UNO.
- The **GND** pin of the sensor is connected to the **GND** pin of the Arduino UNO to complete the circuit.

I²C Communication:

- The **SCL (Clock Line)** of the sensor is connected to **A5 (SCL)** on the Arduino UNO.
- The **SDA (Data Line)** of the sensor is connected to **A4 (SDA)** on the Arduino UNO.
- These connections allow the **Arduino UNO** to communicate with the **SCD41 sensor** using the I²C protocol.

5.Example code link

We provide example codes to help you get started with the **7Semi SCD40 CO_2 Sensor Breakout Qwiic I^2C**. These examples demonstrate how to communicate with the sensor and retrieve CO_2, temperature, and humidity data using the I^2C protocol. The code is available for two popular platforms: Arduino and ESP32.

Code Explanation

1. Macro Definitions and Global Variables

```
#ifdef NO_ERROR
#undef NO_ERROR
#endif
#define NO_ERROR 0
SensirionI2cScd4x sensor;
static char errorMessage[64];
static int16_t error;
```

- **Macro Definition:**
  The macro NO_ERROR is defined as 0, ensuring that error checking later in the code is standardized.
- **Global Variables:**
  - sensor: An instance of the sensor driver class.
  - errorMessage: A character array used to store error messages.
  - error: A variable to hold error codes returned by sensor functions.

2. Helper Function: PrintUint64

```
void PrintUint64(uint64_t& value) {
    Serial.print("0x");
    Serial.print((uint32_t)(value >> 32), HEX);
    Serial.print((uint32_t)(value & 0xFFFFFFFF), HEX);
}
```

This function prints a 64-bit unsigned integer in hexadecimal format by splitting it into two 32-bit halves. It's mainly used for displaying the sensor's serial number in a human-readable hex format.

3. The loop() Function

```
void loop() {
    error = sensor.wakeUp();
    if (error != NO_ERROR) { /* handle error */ return; }
```

- **Sensor Wake-Up:**
  Before each measurement, the sensor is woken up. This is important if the sensor was put to sleep to save power between measurements.

```
    Serial.print("CO2 concentration [ppm]: ");
    Serial.print(co2Concentration);
    Serial.println();
    Serial.print("Temperature [°C]: ");
    Serial.print(temperature);
    Serial.println();
    Serial.print("Relative Humidity [RH]: ");
    Serial.print(relativeHumidity);
    Serial.println();
    Serial.print("sleep for 5 minutes until next measurement is due");
    Serial.println();
    delay(300000);}
```

- **Output Results:**
  The measured values are printed to the serial monitor with appropriate labels.
- **Delay:**
  The program then waits for 5 minutes (300,000 milliseconds) before performing the next measurement cycle.

4. Error Handling

Throughout the code, after each sensor operation (e.g., waking up, stopping measurements, reinitializing, reading values), the return value is checked:

```
if (error != NO_ERROR) {
    Serial.print("Error trying to execute <function>(): ");
    errorToString(error, errorMessage, sizeof errorMessage);
    Serial.println(errorMessage);
    return;
}
```

5. Arduino Example Code

This example is designed for Arduino-compatible boards and demonstrates:

- Initializing the I^2C communication with the SCD4x sensor Board.
- Reading CO_2 concentration, temperature, and humidity data.
- Printing the sensor data to the Serial Monitor.

6. ESP32 Example Code

This example targets ESP32 boards and showcases:

- Configuring the ESP32 I^2C interface to communicate with the SCD4x sensor Board.
- Reading and processing CO_2, temperature, and humidity data from the sensor.
- Printing the sensor data to the Serial Monitor.

How to Access the Code

- **Download Link for Arduino and ESP32 Example**: Click Here

5.1 Sample Serial Output Arduino

Sample output Image of Arduino:

```
CO2: 1120 ppm, Temp: 24.25 °C, Humidity: 61.96 %RH
CO2: 1163 ppm, Temp: 24.26 °C, Humidity: 61.79 %RH
CO2: 1136 ppm, Temp: 24.27 °C, Humidity: 61.78 %RH
CO2: 1167 ppm, Temp: 24.24 °C, Humidity: 61.81 %RH
CO2: 1137 ppm, Temp: 24.29 °C, Humidity: 61.67 %RH
CO2: 1143 ppm, Temp: 24.32 °C, Humidity: 61.60 %RH
CO2: 1167 ppm, Temp: 24.35 °C, Humidity: 61.50 %RH
CO2: 1147 ppm, Temp: 24.34 °C, Humidity: 61.39 %RH
CO2: 1135 ppm, Temp: 24.33 °C, Humidity: 61.29 %RH
CO2: 1121 ppm, Temp: 24.22 °C, Humidity: 61.40 %RH
CO2: 1150 ppm, Temp: 24.22 °C, Humidity: 61.50 %RH
```

6.Mechanical Specification