

SGP40 VOC Sensor

The SGP40 VOC Sensor is a digital VOC (Volatile Organic Compounds) sensor module that incorporates SGP40, designed for air quality monitoring. It can be used to measure fire coal, natural gas emissions, cigarette/cooking smoke, and emissions from constructing activity, decoration material, furniture, home appliances, then generate digital signals for triggering air treatment devices. The SGP40 VOC Sensor is small in size for easy integration into air treatment devices and air quality monitors, definitely an ideal choice for learning air quality monitoring.

Feature

- Incorporates SGP40, for measuring 0 ~ 1000 ppm of ethanol equivalents
- Features humidity-compensation and low-heat-dissipation, providing outstanding performance on sorts of environments
- Digital I2C output, less interference
- Onboard voltage translator, compatible with 3.3V/5V operating voltage

Specification

- Sensor: SGP40
- Measurement range: 0~1000 ppm of ethanol equivalents
- Communication bus: I2C (constant address: 0x59)
- Operating voltage: 3.3V/5V
- Response time: <10s (tau 63%)
- Switch-on time: <60s
- Dimensions: 27 x 20mm
- Mounting hole size: 2.0mm

Pinout

- Pins

PIN	SYMBOL	Description
1	VCC	3.3V/5V Power on
2	GND	Ground
3	SDA	I2C data pin
4	SCL	I2C clock pin

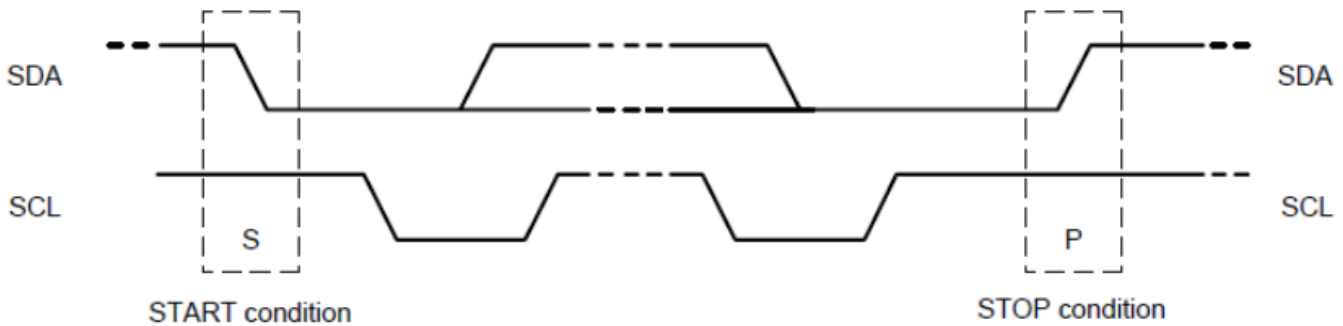
Hardware

Controller

This product uses Sensirion's SGP40 gas sensor and is designed for easy integration into air purifiers or demand-controlled ventilation systems. Sensirion's CMOSens® technology provides a complete, easy-to-use sensor system on a single chip with a digital I2C interface and temperature-controlled micro-heating plates that provide VOC-based humidity compensated indoor air quality signals. Sensirion's powerful VOC algorithm processes the output signal directly, converting the original signal into a VOC index that can be used as a reliable measure of indoor air quality. The VOC algorithm automatically ADAPTS to the environment in which the sensor is located. Both the sensing element and the VOC algorithm are unmatched in robustness against the polluting gases present in real-world applications, resulting in unique long-term stability and low drift and device-to-device differences.

Communication protocol

SGP40 uses I2C communication, I2C communication, one data line, one clock line. Normal I2C bus has three types of signals in the data transmission process: start signal, end signal, and response signal, as shown in the figure below:

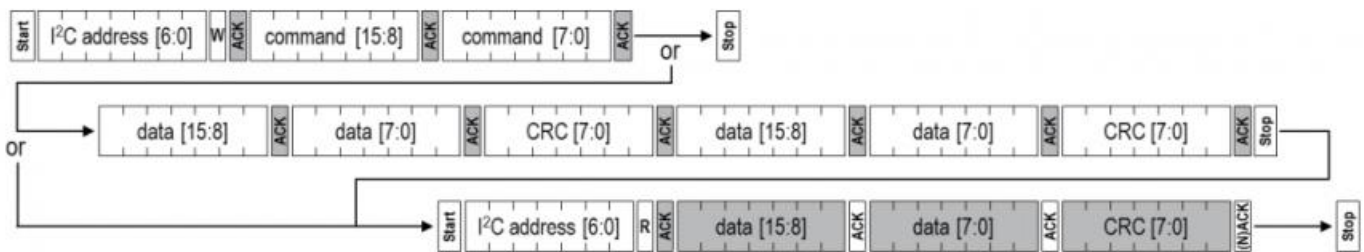


Start signal: When SCL is high, SDA jumps from high level to low level, and data transmission begins.

End signal: When SCL is high, SDA jumps from low level to high level, and data transmission ends.

Reply signal: After receiving 8bit data, the receiving IC sends a specific low-level pulse to the sending IC to indicate that the data has been received..

- I2C data read and write operations for SGP40

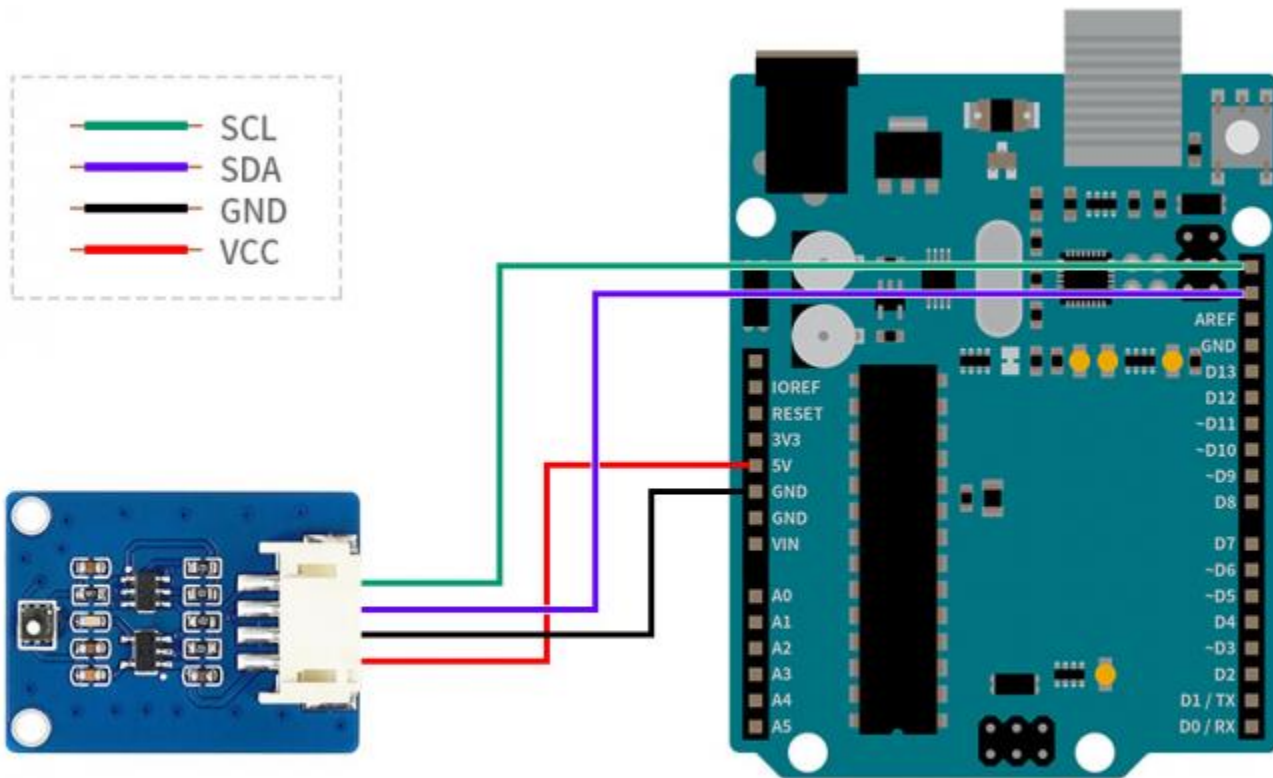


As can be seen from the figure above, the data of SGP40 is sent with 16-bit addresses, and the 16-bit data sent needs a CRC-8/MAXIM verification code.

CRC

Cyclic Redundancy Check (CRC) is one of the most commonly used error Check codes in the field of data communication. Its characteristic is that the length of information fields and verification fields can be arbitrarily selected. Cyclic redundancy check (CRC) is a data transmission error detection function that performs polynomial calculations on the data and attaches the results to the back of the frame. The receiving device also performs a similar algorithm to ensure the correctness and integrity of the data transmission. SGP40 uses THE CRC-8/MAXIM check code calculation, that is, the calculation result of polynomial $x^8 + x^5 + x^4 + 1$. The procedure has been simplified, using the lookup table method.

Hardware connection



Install Arduino IDE

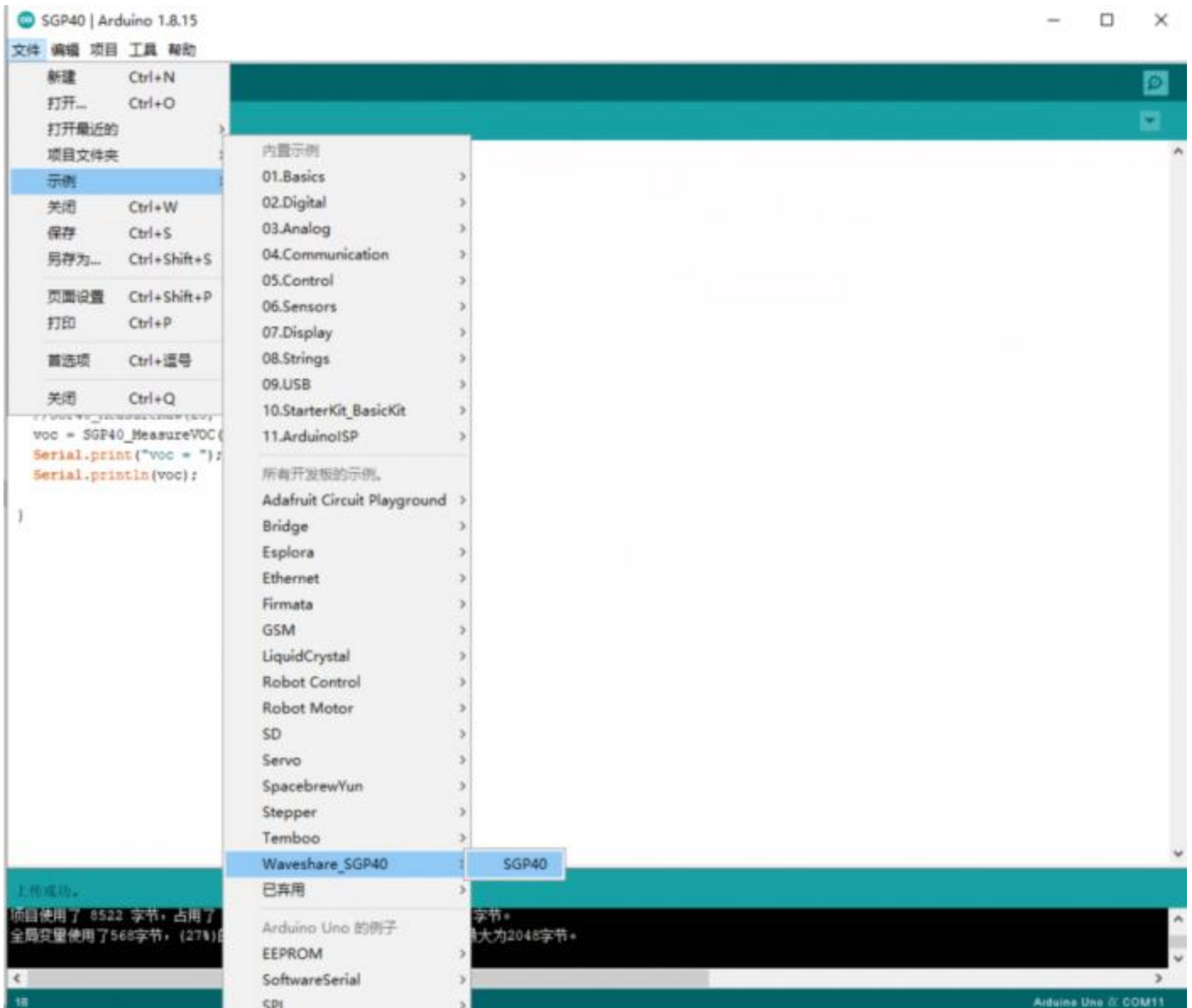
Please download the IDE from the Arduino website and install it.

Test the codes

Download the demo codes from the Resources part and unzip it

This routine is written based on the Arduino library version, so you need to copy the program to the Arduino library directory. Copy the Waveshare_SGP40 folder in the Arduino directory from the next week's files to the Libraries folder in the Arduino installation directory (C:\Users\XXX\Documents\Arduino\libraries or C:\Program Files (x86)\Arduino\libraries)

Open the Arduino IDE: Choose file -> example -> Waveshare_SGP40 and open it.



The Development board select the appropriate model, select the appropriate COM port, compile the program, download to UNO, open the serial port monitor

```

COM11
00:49:34.591 -> SGP40 VOC Sensor Init
00:49:36.609 -> voc = 0
00:49:37.080 -> voc = 0
00:49:37.597 -> voc = 0
00:49:38.101 -> voc = 0
00:49:38.620 -> voc = 0
00:49:39.140 -> voc = 0

```

COM11 window controls: 发送, 自动滚屏, Show timestamp, 换行符, 115200 波特率, 清空输出

It is important to note that this value uses a converted value, where VOC is 0 under normal conditions. Unless the environment is harsh, alcohol can be used instead of volatile gas, and a greater sensitive beat can be obtained.

Hardware connection

Connection with XNUCLEO-F103RB:

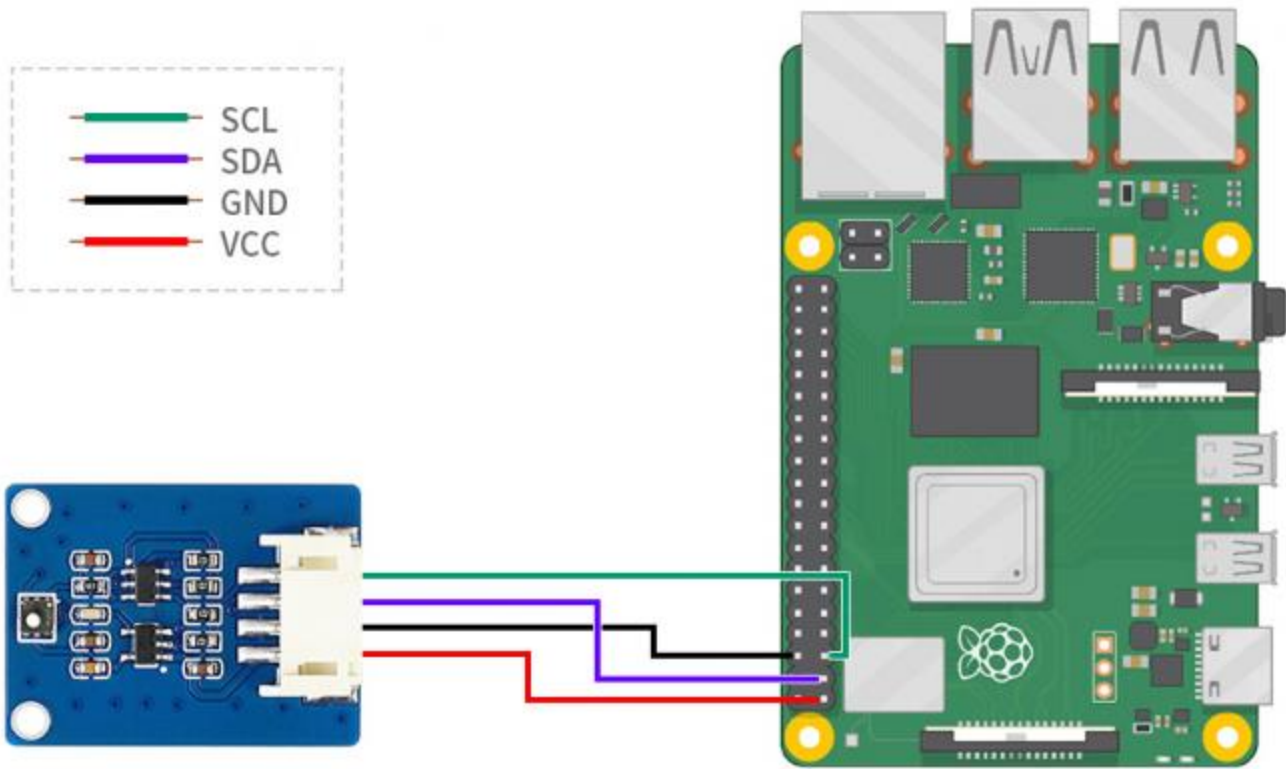
AS7341 Spectral Color Sensor	XNUCLEO-F103RB
VCC	3.3V/5V
GND	GND
SDA	SDA/D14/PB9
SCL	SCL/D15/PB8

Hardware connection

Connection with XNUCLEO-F103RB:

AS7341 Spectral Color Sensor	XNUCLEO-F103RB
VCC	3.3V/5V
GND	GND
SDA	SDA/D14/PB9
SCL	SCL/D15/PB8

Hardware connection



AS7341 Spectral Color Sensor

Raspberry Pi (BCM)

VCC

3.3V/5V

GND

GND

SDA

SDA(2)

SCL

SCL(3)

Working with Raspberry Pi

Enable I2C interface

- Open a terminal and run the following commands:

```
sudo raspi-config
#Choose Interfacing Options -> I2C ->yes to enable the I2C interface
```

Raspberry Pi Software Configuration Tool (raspi-config)

- 1 Change User Password Change password for the current user
- 2 Network Options Configure network settings
- 3 Boot Options Configure options for start-up
- 4 Localisation Options Set up language and regional settings to match your location
- 5 Interfacing Options Configure connections to peripherals**
- 6 Overclock Configure overclocking for your Pi
- 7 Advanced Options Configure advanced settings
- 8 Update Update this tool to the latest version
- 9 About raspi-config Information about this configuration tool

<Select>

<Finish>

Raspberry Pi Software Configuration Tool (raspi-config)

- P1 Camera Enable/Disable connection to the Raspberry Pi Camera
- P2 SSH Enable/Disable remote command line access to your Pi using SSH
- P3 VNC Enable/Disable graphical remote access to your Pi using RealVNC
- P4 SPI Enable/Disable automatic loading of SPI kernel module
- P5 I2C Enable/Disable automatic loading of I2C kernel module**
- P6 Serial Enable/Disable shell and kernel messages on the serial connection
- P7 1-Wire Enable/Disable one-wire interface
- P8 Remote GPIO Enable/Disable remote access to GPIO pins

<Select>

<Back>

Would you like the ARM I2C interface to be enabled?

<Yes>

<No>

- Restart Raspberry Pi

```
sudo reboot
```

Install libraries

- Install BCM2835, open the terminal and run the following command

```
wget http://www.airspayce.com/mikem/bcm2835/bcm2835-1.60.tar.gz
tar zxvf bcm2835-1.60.tar.gz
cd bcm2835-1.60/
sudo ./configure
sudo make
sudo make check
sudo make install
```

- Install wiringpi

```
sudo apt-get install wiringpi
cd /tmp
wget https://project-downloads.drogon.net/wiringpi-latest.deb
sudo dpkg -i wiringpi-latest.deb
gpio -v
```

Download and test the demo codes

```
sudo apt-get install p7zip-full
wget https://www.waveshare.com/w/upload/6/62/SGP40_Voc_Sensor_code.7z
7z x SGP40_Voc_Sensor_code.7z -r -o./SGP40_Voc_Sensor_code
sudo chmod 777 -R SGP40_Voc_Sensor_code
```

python codes

```
cd
cd SGP40_Voc_Sensor_code/RPI
sudo python SGP40.py
```

- Note: This sensor is extremely sensitive to ethanol
- Note: The sensor starts with a heating process, during which the raw data slowly increases until it stabilizes.

Documents

- [SGP40 Datasheet](#)

- [Schematic](#)

Codes

- [Demo codes](#)

Support

If you require technical support, please go to the [Support](#) page and open a tickets.