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 demandcontrolled 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 VOCbased 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 realworld 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.

新課 Curl+N 打开した。 現意文は未 予約 Curl+W 行用では、 予約 Curl+S 用力、 定日というた。 日本ののです。 の2.0 jajkal 3 01.Basics 3 02.0 jajkal 3 02.0 jajkal 3 02.0 jajkal 3 02.0 jajkal 3 02.0 jajkal 3 02.0 jajkal 3 02.0 curl-D 3 03.0 curl-D 3 05.0 curl-D 3 04.0 curl-D 2007 50 5 50 5	件 编辑项目 工具 帮助			
FTR-Ch1-O TTR-Sc20 TTR-Sc20 TTR-Sc20 TTR-Sc20 P型型の例 D1Batics P型型の例 D1Batics P型型の例 D1Batics P型型の例 D1Batics PUBO例 D1Batics PUBO/D D1Batics PUBO/D D1Batics PUBO/D D1Batics PUBO/D D1Batics PUBO/D D2Diptal PUBO/D	新建 Ctrl+N			
PST 2000 - 2000	打开 Ctrl+O			
 ・ ・・・・・・・・・・・・・・・・・・・・・・・・・・・	打开最近的			S
大田 Chrl+W 02.Dighal 0 保存 Chrl+W 03.Analog 0 男方力 Chrl+Shift+S 0.4.Communication 0 万田位 Chrl+W 03.Encors 0 市田市 Chrl+B 0.5.Control 0 市田市 Chrl+B 0.5.Control 0 英田 Chrl+B 0.0.Strings 0 0.0.Strings 0 0.0.Strings 0 Serial_printle(root) TheTpEMShOR(0 0 Adafruit Circuit Playground > Bridge Epplora 0 Bridge Epplora 0 0 Serial_printle(root) Robet Motor 0 0 Serve 0 0 0 0 Serve 0 <t< td=""><td>项目文件夹</td><td>内體示例</td><td></td><td></td></t<>	项目文件夹	内體示例		
	示例			
第方法:Ctrl+Shift+S 万面位置 Ctrl+Shift+P 万面 Ctrl+型 方面 Ctrl+型 の S.Strings 0 0 0.Strings 0 0	关闭 Ctrl+W			
05.Control 05.Co	保存 Ctrl+S			
Curl - Shift+P TD Curl + P D6.Sensors O Display O O Display O O O O Display O	另存为 Ctrl+Shift+S	04.Communication		
11日 Ctrl+P 第三次 Ctrl+B 07.Display 0 90.USB 0 10.StarterKit, BasicKit 0 Bridge 0 Bridge 0 Bridge 0 Bridge 0 Bridge 0 Serial.printle("voo") Frientet 10.StarterKit, BasicKit 0 Bridge 0 Bridge 0 Serial.printle("voo") Frientet Bridge 0 Soot 0	THEOR CHICKING	Concernence of the second s		
第連項 Cvrl+運号 08.5trings > 実項 Cvrl+運号 09.USB > 09.USB 10.5tarterKit BasicKit > 11.ArduinoISP > > Serial.print("voc); ////////////////////////////////////				
発想 Ctrl+Q 09.USB 3 Yoo = SGP40_MeasureVCC 10.StarterKit_BasicKit 1 Serial.println(voc); 11.ArduinoISP 3 Maffruit Circuit Playground 3 3 Bridge 3 5 Ethernet 3 3 GSM 3 1 UquidCrystal 3 3 Robot Control 3 3 SpacebrewYun 3 3 Stepper 3 5 Temboo 5 5 Temboo 5 5 Arduino Uno BhH? 5 5 EtPROM 5 5 Arduino Uno BhH? 5 5 EtPROM 5 5 Bridge 5 5 Bridge<	fibi Chi+P	07.Display		
Kit Christian (Voc) = SigP40_MeasureVoc(SerFal.print(Voc) = T): SerFal.print(Voc) = T): SerFal.p	首选项 Ctrl+逗号			
10.5 Aarter KG, Basic Kit > 10.5 Aarter KG, Basic Kit > 11.ArduinolSP > 所相开发版的示例。 Adafruit Circuit Playground > Bridge > Ethernet > Firmata > GSM 2 GSM 2 Ethernet > Robot Control > SD 3 Specifie WYun 3 Stepper 3 Temboo 2 Waveshare <u>SGP40</u> 日東用 > KGP40 日東用 > KGP40 日東用 > KGP40 日東用 > KGP40 日本uino Uno 的明子 EEPROM >	半月 (1/1+0)			
Serial.print("voc): Serial.print(voc): Adafruit Circuit Playground > Bridge > Explora > Explora > Ethernet > GSM > LiquidCrystal > Robot Control > SD > Servo > Servo > Servo > SpacebrewYun > Stepper 2 Temboo > Waveshare_SGP40 * CGM- EFR > Arduino Uno 的例子 EEPROM > ***	CY DOCAT _ REMORE COMP (EV)			
Serial.println(voc): ////////////////////////////////////		11.ArduinoISP		
Madafruit Circuit Playground > Bridge > Bridge > Esplora > Ethernet > GSM > LiquidCrystal > Robot Control > SD > SD > SpacebrewYun > Stepper > Temboo > VMN* CEAR Waveshare_SGP40 * Arduino Uno 的例子 SGP40 **** ************************************		所有开发板的示例。		
Bridge > Explora > Explora > Ethernet > Firmata > GSM > LiquidCrystal > Robot Control > SD > SP > SpacebrewYun > Stepper > Temboo > Waveshare_SGP40 SGP40 ERR > Arduino Uno 的例子 > EEPROM >				
Explora > Ethernet > Firmata > GSM > LiquidCrystal > Robot Control > SD > Servo > SpacebrewYun > Stepper > Temboo > Waveshare_SGP40 SGP40 EARH > Waveshare_SGP40 \$GP40 EARH > Arduino Uno 的例子 \$GP40 EEPROM >				
Ethernet > Firmata > GSM > LiquidCrystal > Robot Control > Robot Motor > SD > Servo > SpacebrewYun > Stepper > Temboo > Waveshare_SGP40 SGP40 BR#U* SGP40 Fire- *** #WUT Doolb/ff: EEPROM				
GSM > LiquidCrystal > Robot Control > Robot Motor > SD > Servo > SpacebrewYun > Stepper > Temboo > Waveshare_SGP40 SGP40 已异用 > Arduino Uno 的例子 EEPROM >				
UquidCrystal > Robot Control > Robot Motor > SD > SD > Servo > SpacebrewYun > Stepper > Temboo >		Firmata		
Robot Control 3 Robot Motor 3 SD 3 Servo 3 SpacebrewYun 3 Stepper 3 Temboo 3 VWAveshare_SGP40 5GP40 使用了 6522		GSM		
Robot Control 3 Robot Motor 3 SD 3 Servo 3 SpacebrewYun 3 Stepper 3 Temboo 3 VWAveshare_SGP40 5GP40 使用了 6522		LiquidCrystal		
SD >> Servo >> SpacebrewYun >> Stepper >> Temboo >> Tem		Robot Control		
Servo > SpacebrewYun > Stepper > Temboo > Temboo > Waveshare_SGP40 SGP40 已用用 > 在用和 > 在和uino Uno 的例子 EEPROM >		Robot Motor		
SpacebrewYun > Stepper > Temboo > Vwwshare_SGP40 1 使用了 6522		SD		
Stepper > Temboo > Waveshare_SGP40 SGP40 Waveshare_SGP40 SGP40 使用了 6522 字节、占用了 Arduino Uno 的例子 EEPROM >		Servo		
Temboo > Waveshare_SGP40 SGP40 使用了 6522 字节,占用了 發展使用了568字节,(270)5 Arduino Uno 的例子 EEPROM 字节。 大为2046字节。		SpacebrewYun		
Waveshare_SGP40 SGP40 日共用 日共用 > 日共工 > 日共工 >		Stepper		
化成化 已弃用 > 1使用了 6522 字节,占用了 Arduino Uno 的例子 字节。 該理量使用了568字节,(27%) EEPROM >		Temboo		
a使用了 8522 字节,占用了 設置使用了568字节,(271)/f Arduino Uno 的例子 EEPROM		Waveshare_SGP40	SGP40	
#2里使用了568字节,(271)E Arduino Uno 即例子 EEPROM →	(C)(EAD.	已弃用		
EEPROM >	目使用了 8522 字节,占用了	Andreas Line Milling		
	設量使用了568字节,(27 1)自	Leave the second se	载大为2045字节。	

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

COM11	- 0
	发
00:49:34.591 -> SGP40 VOC Sensor Init	
00:49:36.609 -> voc = 0	
0:49:37.080 -> voc = 0	
0:49:37.597 -> voc = 0	
0:49:38.101 -> voc = 0	
0:49:38.620 -> voc = 0	
0:49:39.140 -> voc = 0	
🗹 自动滚屏 🗹 Show timestamp	执行符 ∨ 115200 波特率 ∨ 清空編

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

5 Interfacing 0 6 Overclock 7 Advanced Opti 8 Update 9 About raspi-c	ptions Configure conne Configure overco ons Configure advar Update this too	e and regional settings to match your location actions to peripherals clocking for your Pi nced settings of to the latest version out this configuration tool
<	elect>	<finish></finish>
		al remote access to your Pi using RealVNC
P3 VNC P4 SPI P4 SPI P5 12C P6 Serial P7 1-Wire P5 12C P5 P5 12C P5 Serial P7 1-Wire P5 P5 12C P5	Enable/Disable automati Enable/Disable automati	Ic loading of SPI kernel module ic loading of I2C kernel module id kernel messages on the serial connection interface
P3 VNC E P4 SPI E P5 I2C E P6 Serial E P7 1-Wire E P8 Remote GPIO E	Enable/Disable automati Enable/Disable automati Enable/Disable shell an Enable/Disable one-wire	Ic loading of SPI kernel module ic loading of I2C kernel module id kernel messages on the serial connection interface

<N0>

• Restart Raspberry Pi

<Yes:

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

<u>SGP40 Datasheet</u>

• <u>Schematic</u>

Codes

• <u>Demo codes</u>

Support

If you require technical support, please go to the <u>Support</u> page and open a tickets.