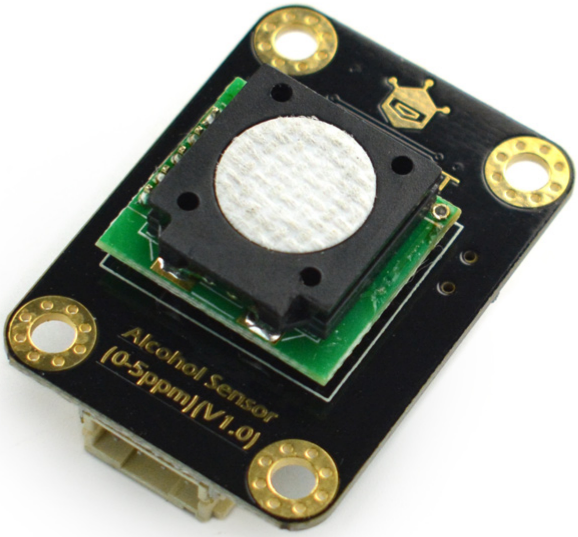
SKU:SEN0376 (https://www.dfrobot.com/product-2186.html)

(https://www.dfrobot.com/product-2186.html)

Introduction



This is an Arduino compatible alcohol sensor module from DFRobot with 0-5ppm measuring range, supporting I2C and UART outputs. Aiming at fast and accurate detection of alcohol gas concentration in the air, the sensor is well applicable to the small alcohol concentration measurement in breathalyzer, smart vehicle, and so on.

Based on the electrochemistry principle, this alcohol sensor has been calibrated before factory. It is characterized with strong anti-interference, high stability and sensitivity and up to 2 years service life. The module supports 0.01ppm(10ppb) resolution and 3.3~5.5V wide range voltage input, and works well with mainstream controllers like Arduino, ESP32 and Raspberry Pi. With the easy-to-use Gravity interface and the sample program provided by us, you can conveniently build your alcohol concentrastion detector using this sensor.

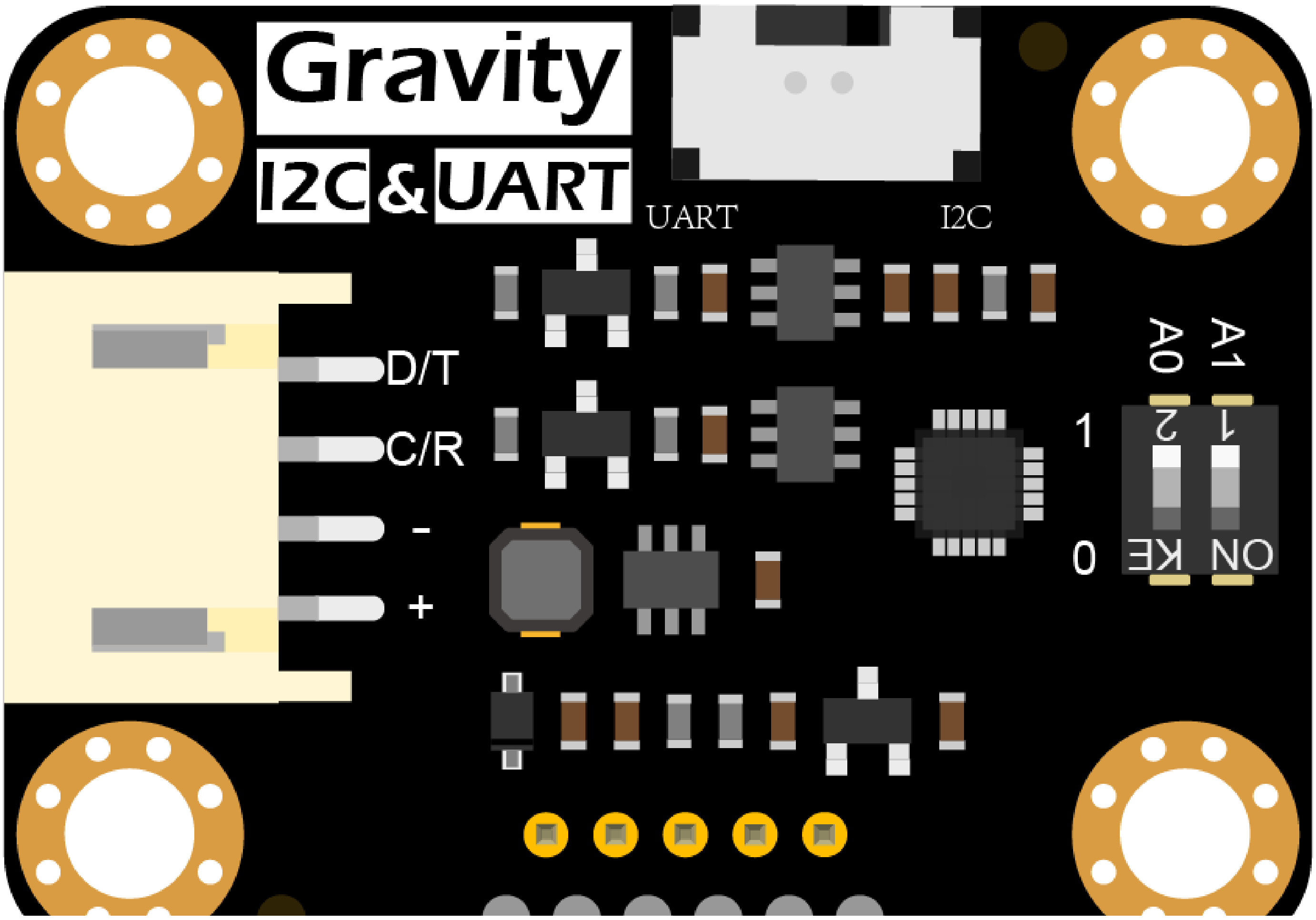
Features

- Factory calibrated, accurate data
- High sensitvity, low power
- Stable performance, anti-interference
- I2C and UART outputs
- Excellent linear output
- Long service life
- Compatible with 3.3~5V controllers
- Reverse-connection protection

Specification

- Detected Gas: alcohol
- Interfering Gas: benzene, toluene, acetic acid, hydrogen sulfide, carbon monoxide
- Operating Voltage: 3.3 ~ 5.5V DC
- Power: 0.05W （5V）
- Signal Output: I2C, UART(0~3V)
- Measuring Range: 0~5ppm
- Resolution: ≤0.01ppm （10ppb）
- Preheat Time: ≤3min
- Response Time: ≤60s
- Recover Time: ≤60s
- Operating Temperature: -20 ~ 50℃
- Operating Humidity: 15 ~ 90%RH (No condense)
- Storage Temperature: 0~25℃
- Service Life: >2 years(In air)
- Board Dimension: 27mm×37mm/1.06×1.46"

Board Overview



Num	Label	Description
1	D/T	I2C data line SDA/UART data transmit-TX
2	C/R	I2C clock line SCL/UART data transmit-RX
3	-	GND
4	+	Positive

Tutorial

Download the codes into the UNO board and open the serial monitor to check the results.

NOTE:

- It is recommended to preheat for 24~48 hours when powering up the module for the first time.
- The alcohol concentration during preheat will become stable 3 minutes later so please ignore the data generated within that time.
- Organic solvents like essential balm(tiger balm) or Florida Water will damage the sensor, so please do not use them for testing.

Requirements

- **Hardware**
  - DFRduino UNO R3 (https://www.dfrobot.com/product-838.html) (or similar) x 1
  - Alcohol sensor x1
  - Jumper wires
- **Software**
  - Arduino IDE (https://www.arduino.cc/en/Main/Software)
  - Download and install the **DFRobot Alcohol Sensor library** (https://github.com/DFRobot/DFRobot_Alcohol/archive/master.zip) (About how to install the library? (https://www.arduino.cc/en/Guide/Libraries#.UxU8mdzF9H0))

Read Sensor Data via I2C

Connection Diagram

Connection Diagram-I2C

Connection Diagram-I2C

Sample Code

- Connect the sensor with Arduino as shown above. A Gravity I/O expansion board will make it more easier.
- Dial the select switch on the sensor to I2C.
- The default I2C address is 0x75, ADDRESS_3 in code. If you want to change the address, first configure the hardware I2C address via the DIP switch on the module, and then revise I2C address define ADDRESS_X in the sample code. The relation between DIP switch and I2C address is shown below:
  - ADDRESS_0：0x72, A0=0, A1=0
  - ADDRESS_1：0x73, A0=1, A1=0
  - ADDRESS_2：0x74, A0=0, A1=1
  - ADDRESS_3：0x75, A0=1, A1=1
- Download and install DFRobot Alcohol Sensor library (https://github.com/DFRobot/DFRobot_Alcohol/archive/master.zip)
- Open Arduino IDE, upload the following codes into Arduino UNO.
- Open Arduino serial monitor, set baud rate to 9600, then check the result.

```
``c
 /*!
  * @file  readAlcohol.ino
  * @brief Reading Alcohol concentration, A concentration of one part per million (PPM).
  * @n step: we must first determine the iic device address, will dial the code switch A0,
  *
  * @copyright   Copyright (c) 2010 DFRobot Co.Ltd (http://www.dfrobot.com)
  * @licence     The MIT License (MIT)
  * @author      ZhixinLiu(zhixin.liu@dfrobot.com)
  * @version     V1.0
  * @date        2020-09-09
  * @get         from https://www.dfrobot.com
  * @url         https://github.com/dfrobot/DFRobot_Alcohol
  */

#include "DFRobot_Alcohol.h"

#define COLLECT_NUMBER   5                      // collect number, the collection range is
#define ALCOHOL_I2C_ADDRESS ADDRESS_3
/*   iic slave Address, The default is ADDRESS_3
       ADDRESS_0               0x72             // i2c device address
       ADDRESS_1               0x73
       ADDRESS_2               0x74
       ADDRESS_3               0x75
*/
DFRobot_Alcohol_I2C Alcohol(&Wire ,ALCOHOL_I2C_ADDRESS);

/*
#ifdef ESP_PLATFORM
  // ESP32 user hardware uart
  // RX io16
  // TX io17
  DFRobot_Alcohol_UART Alcohol(&Serial2 ,9600);
#else
  // Arduino user software uart
  // RX io10
  // TX io11
  SoftwareSerial             mySerial(10 ,11);
  DFRobot_Alcohol_UART Alcohol(&mySerial ,9600);
#endif
*/

void setup()
{
  Serial.begin(9600);
  while(!Alcohol.begin())
  {
    Serial.println("NO Deivces !");
    delay(1000);
  }
  Serial.println("Device connected successfully !");

/*  Set iic mode, active mode or passive mode
      MEASURE_MODE_AUTOMATIC            // active  mode
      MEASURE_MODE_PASSIVE              // passive mode
*/
  Alcohol.SetModes(MEASURE_MODE_AUTOMATIC);
}


void loop()
{
/*  Smooth data collection
      COLLECT_NUMBER                    // The collection range is 1-100
*/
  float alcoholConcentration = Alcohol.ReadAlcoholData(COLLECT_NUMBER);
  if(alcoholConcentration == ERROR)
  {
    Serial.println("Please check the connection !");
  }else{
    Serial.print("Alcohol concentration is ");
    Serial.print(alcoholConcentration);
    Serial.println(" PPM.");
  }
  delay(1000);
}
```

Expected Result 1

Open the serial monitor, preheat 3 minutes and check the final data.

Note:

1. There may be data drift appearing.
2. Electrochemical type sensors need to be preheated for 24~48 hours when powering up for the first time.

Result 1

Read Sensor Data via UART

Connection Diagram

Connection Diagram-UART

Sample Code

- Connect the sensor with Arduino as shown above. A Gravity I/O expansion board will make it more easier.

- Dial the select switch on the sensor to UART.

- Download and install DFRobot Alcohol Sensor library (https://github.com/DFRobot/DFRobot_Alcohol/archive/master.zip)
- Open Arduino IDE, upload the following codes into Arduino UNO.
- Open Arduino serial monitor, set baud rate to 9600, then check the result.

```
/*!
  * @file  readAlcohol.ino
  * @brief Reading Alcohol concentration, A concentration of one part per million (PPM).
  * @n step: we must first determine the iic device address, will dial the code switch A0,
  *
  * @copyright   Copyright (c) 2010 DFRobot Co.Ltd (http://www.dfrobot.com)
  * @licence     The MIT License (MIT)
  * @author      ZhixinLiu(zhixin.liu@dfrobot.com)
  * @version     V1.0
  * @date        2020-09-09
  * @get         from https://www.dfrobot.com
  * @url         https://github.com/dfrobot/DFRobot_Alcohol
  */

#include "DFRobot_Alcohol.h"

#define COLLECT_NUMBER   5                      // collect number, the collection range is
#define ALCOHOL_I2C_ADDRESS ADDRESS_3
/*   iic slave Address, The default is ADDRESS_3
       ADDRESS_0               0x72             // i2c device address
       ADDRESS_1               0x73
       ADDRESS_2               0x74
       ADDRESS_3               0x75
*/
//DFRobot_Alcohol_I2C Alcohol(&Wire ,ALCOHOL_I2C_ADDRESS);

#ifdef ESP_PLATFORM
  // ESP32 user hardware uart
  // RX io16
  // TX io17
  DFRobot_Alcohol_UART Alcohol(&Serial2 ,9600);
#else
  // Arduino user software uart
  // RX io10
  // TX io11
  SoftwareSerial             mySerial(10 ,11);
  DFRobot_Alcohol_UART Alcohol(&mySerial ,9600);
#endif

void setup()
{
  Serial.begin(9600);
  while(!Alcohol.begin())
  {
    Serial.println("NO Deivces !");
    delay(1000);
  }
  Serial.println("Device connected successfully !");
/*  Set iic mode, active mode or passive mode
      MEASURE_MODE_AUTOMATIC            // active  mode
      MEASURE_MODE_PASSIVE              // passive mode
*/
  Alcohol.SetModes(MEASURE_MODE_AUTOMATIC);
}


void loop()
{
/*  Smooth data collection
      COLLECT_NUMBER                    // The collection range is 1-100
*/
  float alcoholConcentration = Alcohol.ReadAlcoholData(COLLECT_NUMBER);
  if(alcoholConcentration == ERROR)
  {
    Serial.println("Please check the connection !");
  }else{
    Serial.print("Alcohol concentration is ");
    Serial.print(alcoholConcentration);
    Serial.println(" PPM.");
  }
  delay(1000);
}
```

Expected Result 1

Open the serial monitor, preheat 3 minutes and check the final data.

Note:

1. There may be data drift appearing.
2. Electrochemical type sensors need to be preheated for 24~48 hours when powering up for the first time.

Result 2

Data Reading Mode

There two data reading modes for this Alcohol Sensor: aoto-upload and passive reply. It is default to auto-upload, and users can alter it in Codes.

Mode Select Functions

Revise the parameter in the parentheses to adjust the data sending mode.

“**MEASURE_MODE_AUTOMATIC**” is auto-upload mode. The sensor automatically uploads parameter every 1 second in this mode.

“**MEASURE_MODE_PASSIVE**” is passive reply mode. The sensor feedbacks parameter only when the function is called in this mode.

```
Alcohol.SetModes(MEASURE_MODE_AUTOMATIC);
/*
      MEASURE_MODE_AUTOMATIC            // Auto-upload
      MEASURE_MODE_PASSIVE              // Passive Reply
*/
```

Data Reading Functions

The data of alcohol sensor can be read by the function “ReadAlcoholData()”. Before outputing, this function will take the average of the data in data collect array “COLLECT_NUMBER”. The default size of that array is 5, and it can be adjusted within 1~100 in the code “#define COLLET_NUMBER 5".

```
#define COLLECT_NUMBER   5

Alcohol.ReadAlcoholData(COLLECT_NUMBER);
/*
      COLLECT_NUMBER                    // The collection range is 1-100
*/
```

Communication Protocol

By UART communication protocol, the SEN0376 Alcohol sensor can be connected to any controllers with UART for data reading and sensor configuration.

Parameter Setup

Baud Rate	**9600**
Data bit	**8**
Check	**1**

Communication Commands

The sensor has two communication modes: auto-upload and passive reply. The default one is auto-upload, and the sensor sends the alcohol concentration every 1 second in this mode.

Data format in auto-upload mode：

Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7

Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
Start	Gas Name C2H5OH	Unit (ppb)	Decimal digit None	Gas Concentration High	Gas Concentration Low	Full-scale High	Full-scale Low
0xFF	0x17	0x04	0x00	0x00	0x25	0x13	0x88

Note: **Gas Concentration(PPB)=Gas Concentration High*256+Gas Concentration Low**. When converted in PPM: PPM= PPB/1000.

In passive reply mode, the commands to re-switch to auto-upload mode are as follows:

Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
Start	None	Swtich command	Auto-upload	none	none	none	none	Check
0xFF	0x01	0x78	0x40	0x00	0x00	0x00	0x00	0x47

The commands to switch to passive-reply mode and read concentration in this mode:

Switch to passive-reply mode:

Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
Start	None	Switch command	passive-reply	None	None	None	None	Check
0xFF	0x01	0x78	0x41	0x00	0x00	0x00	0x00	0x46

Read gas concentration:

Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
Start	None	Command	None	None	None	None	None	Check
0xFF	0x01	0x86	0x00	0x00	0x00	0x00	0x20	0x79

Data format in passive-reply mode:

Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
Start	Command	None	None	None	None	Gas concentration High(ppb)	Gas concentration Low(ppb)
0xFF	0x86	0x00	0x00	0x00	0x00	0x00	0x20

0xFF	0x86	0x00	0x00	0x00	0x00	0x00	0x20
Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7

Gas concentration(PPB)=Gas concetration High*256+Gas concentration Low

Checksum Caculation

Checksum= （Reverse（Byte1 + Byte2 + ⋯ ⋯ +Byte7） + 1

Reference Sample Program:

```
/*****************************************************************
* Name: unsigned char FucCheckSum(uchar *i,ucharln)
* Function:checksum（Reverse the sum of 1\2\3\4\5\6\7 of sending and receiveing, then plus
* Description:Sum up the elements from the first to the second to last in the array, rever
*****************************************************************/
unsigned char FucCheckSum(unsigned char *i,unsigned char ln)
{
    unsigned char j,tempq=0;
    i+=1;
    for(j=0;j<(ln-2);j++)
    {
        tempq+=*i;
        i++;
    }
    tempq=(~tempq)+1;
    return(tempq);
}
```

Note

Cross Interference Characteristics

Due to the electrochemical principle, the sensor performance may be influenced by other gases, shown as below:

Gas	Concentration/PPM	Equivalent concentration of ethanol
Benzene	10	0.1
Toluene	10	0.46
Acetic acid	200	0.52
Hydrogen sulfide	50	3
Carbon monoxide	200	0.64

Operating Principles

Operating Principles

- Do not contact the module with substances like silica gel, adhesives, coatings, chemicals, oil and high concentration gas.

- The sensor cannot be completely packaged with resin material or immersed in an oxygen free environment, otherwise its performance will be affected.
- The module cannot be used in environment with corrosive gas for a long time since it will cause damage to the sensor.
- Excessive impact and vibration should not be applied to the sensor.
- Preheat the sensor for 24-48 hours when powering up for the first time to make it perform stably.
- Organic solvents will affect the accuracy of the sensor, such as air freshener, cosmetics, Florida Water, etc.
- Do not install the module in environment with Strong convective air.
- Do not place the product in high concentration organic gas for a long time since it will lead to sensor zero drift and slow recovery.
- It is forbidden to use hot melt adhesive or sealant with curing temperature higher than 80℃ to fix the module.
- Do not place and use the sensor in alkaline gas of high concentration for a long time.

FAQ

For any questions, advice or cool ideas to share, please visit the **DFRobot Forum** (https://www.dfrobot.com/forum/).

More Documents

- Schematics Diagram (https://github.com/DFRobot/Wiki/raw/master/SEN0376/%5BSEN0376%5DSchematic%20V1.0.pdf)
- Dimension and Component Layout (https://github.com/DFRobot/Wiki/raw/master/SEN0376/%5BSEN0376%5DDimension%20V1.0.pdf)

Get **Alcohol Sensor** (https://www.dfrobot.com/product-2186.html) from DFRobot Store or **DFRobot Distributor**. (https://www.dfrobot.com/index.php?route=information/distributorslogo)

Turn to the Top