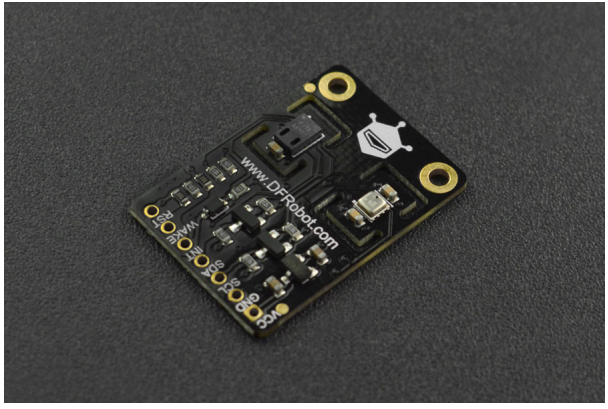
SKU:SEN0335 (https://www.dfrobot.com/product-2064.html)



(https://www.dfrobot.com/product-2064.html)

Introduction

Build up a simple environmental monitor station with this multi-function environment sensor!

Based on the combination of CCS811+BME280 chip, this module features high accuracy, IIC interface and fast Measurement. The BME280 can provide temperature and humidity compensation for CCS811 to improve the whole accuracy to a certain extent. It can be used to detect temperature, humidity, barometric pressure, altitude, TVOC and eCO2.
CCS811 air quality sensor uses AMS's unique micro-hot plate technology. Compared with conventional gas sensors, it has lower power consumption, shorter preheating time, and smaller size. The internally integrated ADC and MCU allow it to collect and process data, and return via I2C.

BME280 is an environmental sensor that combines temperature sensor, humidity sensor and barometer in one board. It has high precision, multiple functions, small size, etc. The sensor offers ±0.5℃ temperature error and ±2%RH humidity error. It provides very stable performance within the detection temperature range. Besides, the offset temperature coefficient is ±1.5 Pa/K, equiv. to ±12.6 cm at 1 °C temperature change.

⚠ **NOTE**: The chip has stretched the clock in I2C. So, it may be not compatible with some controllers, such as Raspberry Pi.

The following table shows the effects of carbon dioxide and TVOC on the human body.

			TVOC	

Carbon Dioxide (PPM)	Effect on Human		TVOC Concentration (PPB)	Effect on Human

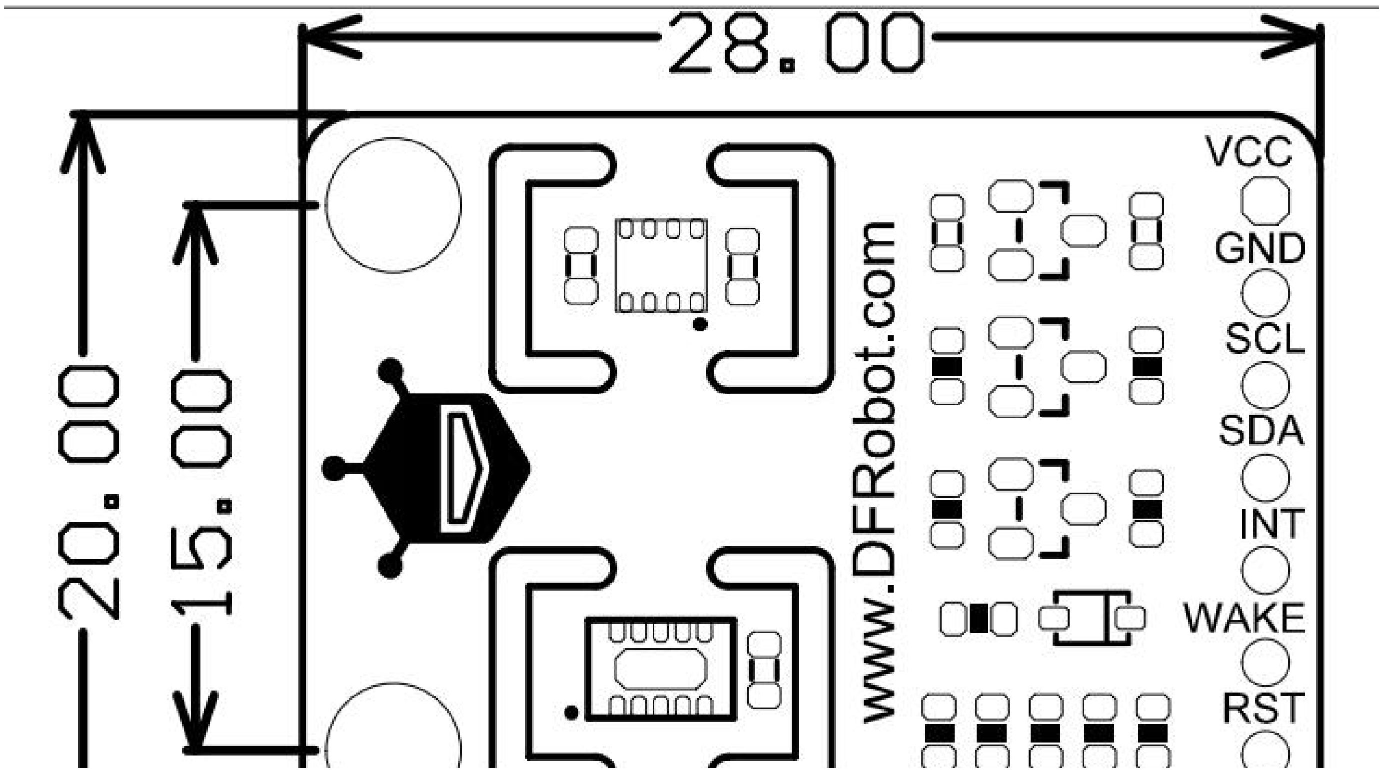
<500	Normal		<50	Normal
500-1000	A little uncomfortable		50-750	Anxious,uncomfortable
1000-2500	Tired		750-6000	depressive, headache
2500-5000	Unhealthy		>6000	headache and other nerve problems

Applications

- Environment Monitor

- Air Purifiers
- Smart Home
- Ventilation System
- Weather Forecast

Specification


28.00
20.00
15.00
www.DFRobot.com
VCC
GND
SCL
SDA
INT
WAKE
RST

- Operating Voltage: 3.3V~5.5 V
- Working Current: <20mA

CCS811 Parameter:

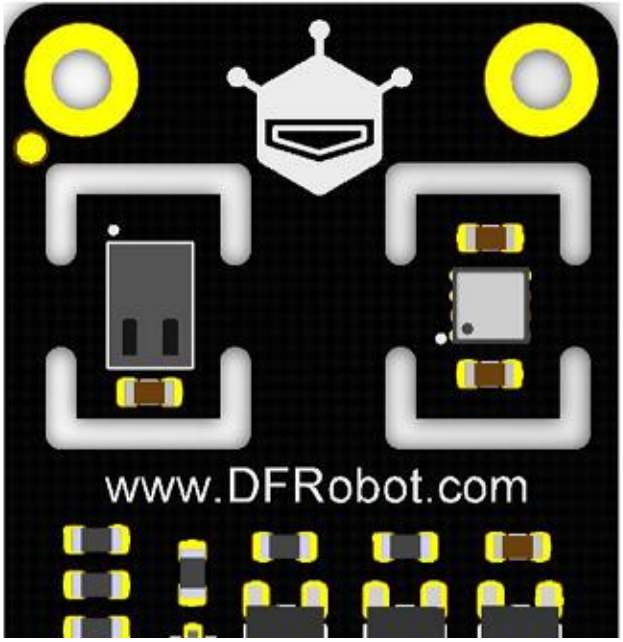
- Preheat Time: <15s
- I2C Address: 0x5A(in default)/0X5B
- Operating Temperature Range: -40℃~85℃
- Operating Humidity Range: 10%RH~95%RH
- eCO2 Measuring Range: 400ppm~8000ppm
- TVOC Measuring Range: 0ppb~1100ppb

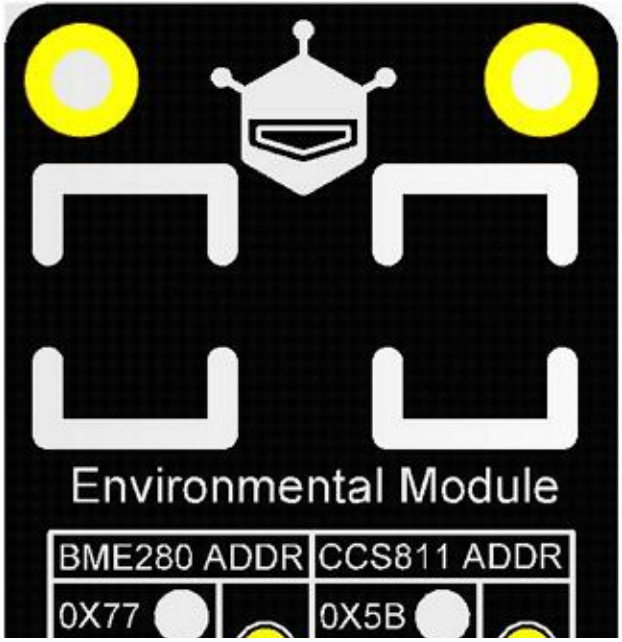
BME280 Parameter:

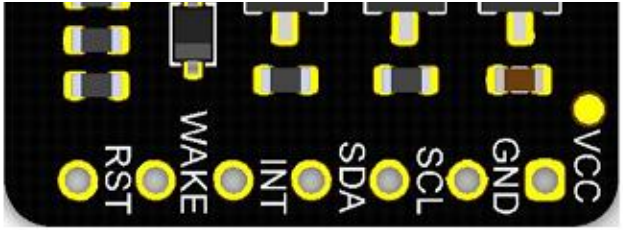
- I2CAddress：0x76(in default)/0X77
- Operating Temperature: -40℃~85℃
- Temperature Measuring Range: -40℃~+85℃, resolution of 0.1℃, deviation of ±0.5℃

- Humidity Measuring Range: 0~100%RH, resolution of 0.1%RH, deviation of ±2%RH
- Pressure Measuring Range: 300~1100hPa

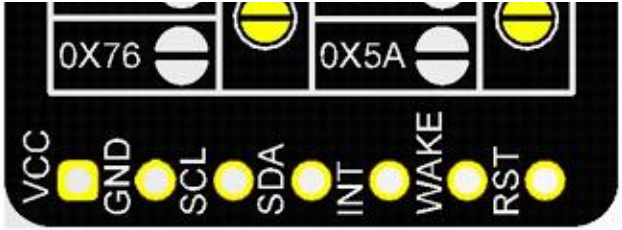
Board Overview







TOP



BOTTOM

Num	Label	Description
1	VCC	+
2	GND	-
3	SCL	IIC clock line
4	SDA	IIC data line
5	INT	Interrupt pin: interrupt in low level
6	WAKE	Switch pin: awake in low level / sleep in high level
7	RST	Reset pin: reset in low level

Tutorial

The product warm-up time is short, accurate readings can be made quickly once powered up. The read time can be shortened by setting the environmental baseline (the baseline acquisition and setup method are explained below; for more details about baseline, go to the end of the wikipage to find the related document).

⚠ **NOTE**: Please run the sensor for 48hours when using it for the first time.

Requirements

- **Hardware**
  - DFRduino UNO R3 (https://www.dfrobot.com/product-838.html) (or similar) x 1
  - Multi-function environmental Module - CCS811+BME280 x 1
  - Jumper wires
- **Software**

    - Arduino IDE (https://www.arduino.cc/en/Main/Software)
    - Download and install the **CCS811 Library and examples** (https://github.com/DFRobot/DFRobot_CCS811).

    - Download and install the **BME280 Library and examples** (https://github.com/DFRobot/DFRobot_BME280) (About how to install the library? (https://www.arduino.cc/en/Guide/Libraries#.UxU8mdzF9H0))

- **About API Function List**

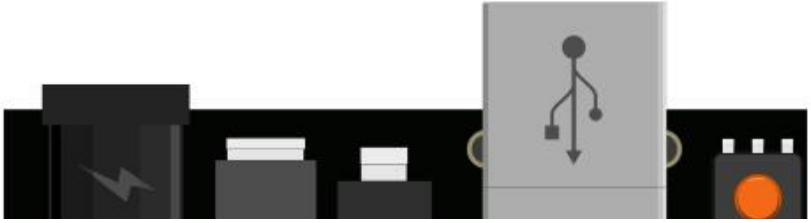
```
/****************CCS811*********************/
/**
  * @brief Judge if the data can be read
  * @return true when the reading is successful, false means it fails to read.
  */
 bool checkDataReady();

   /**
  * @brief Set environment parameter
  * @param temperature Input temperature value, unit: centigrade, range (-40~85℃)
  * @param humidity    Input humidity value, unit: RH, range (0~100)
  */
 void setInTemHum(float temperature, float humidity);

 /**
  * @brief Measurement parameter configuration
  * @param mode:in typedef enum{
  *              eClosed,      //Idle (Measurements are disabled in this mode)
  *              eCycle_1s     //Constant power mode, IAQ measurement every second
```

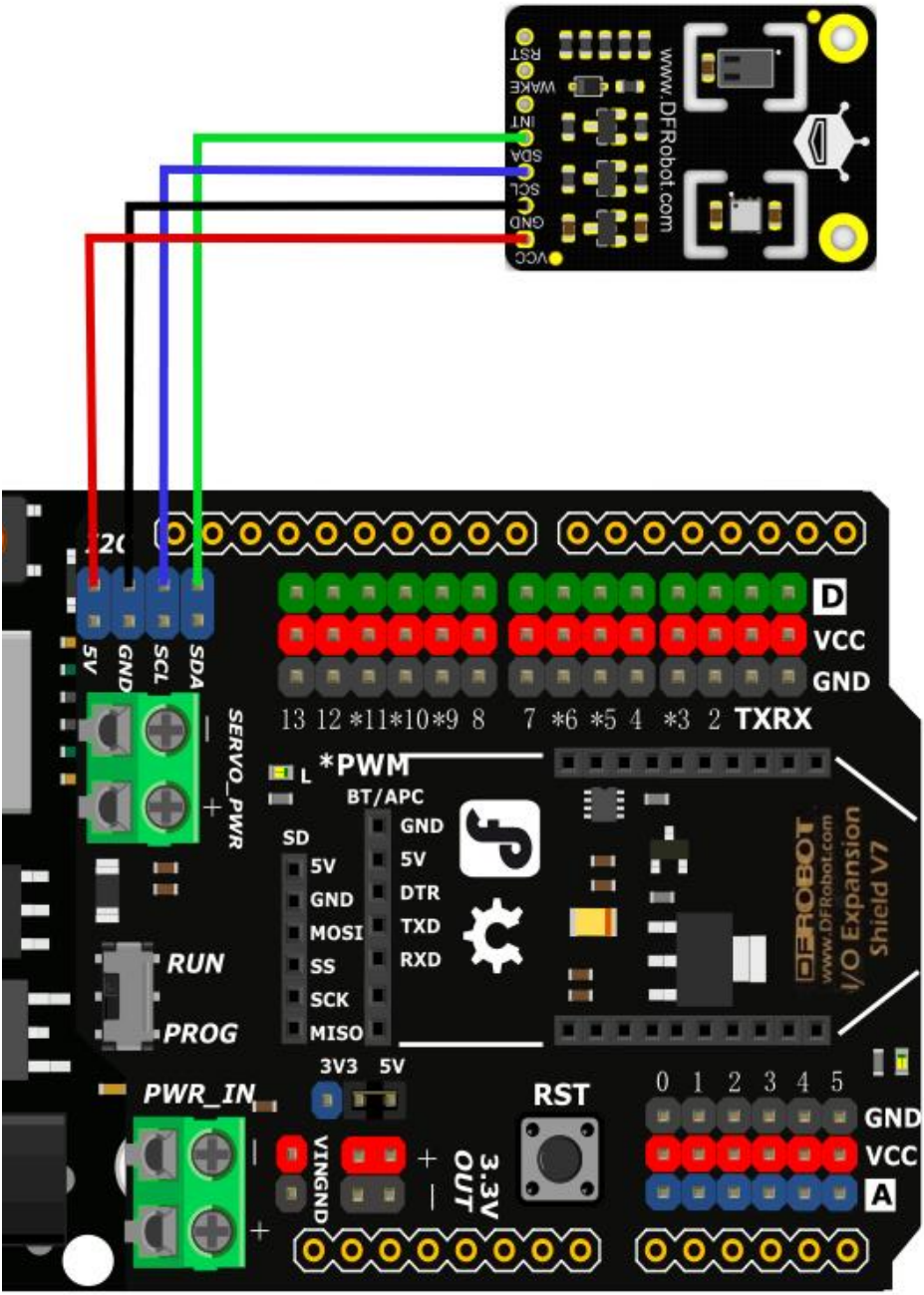
```
 *              eCycle_1s,    //Constant power mode, IAQ measurement every second
 *              eCycle_10s,   //Pulse heating mode IAQ measurement every 10 seconds
 *              eCycle_60s,   //Low power pulse heating mode IAQ measurement every 60 se
 *              eCycle_250ms  //Constant power mode, sensor measurement every 250ms 1xx
 *          }eCycle_t;

 * @param thresh:0 for Interrupt mode operates normally; 1 for interrupt mode only asser
 * @param interrupt:0 for Interrupt generation is disabled; 1 for the nINT signal is ass
 */
setMeasurementMode(eCycle_t mode, uint8_t thresh = 0, uint8_t interrupt = 0),

/**
 * @brief Get the current carbon dioxide concentration
 * @return current carbon dioxide concentration, unit:ppm
 */
uint16_t  getCO2PPM();

/**
 * @brief Get current TVOC concentration
 * @return Return current TVOC concentration, unit: ppb
 */
uint16_t getTVOCPPB();

/**
 *@brief get the current baseline number
```

```
   *@return a Hexadecimal number of the current baseline number
   */
  uint16_t readBaseLine();

  /**
   *@brief write a baseline number into register
   *@param a Hexadecimal number get from getBaseLine.ino
   */
  void writeBaseLine(uint16_t baseLine);


/*****************BME280*********************/
   /**
   * @brief getTemperature Get temperature
   * @return Temprature in Celsius
   */
  float getTemperature();

  /**
   * @brief getPressure Get pressure
   * @return Pressure in pa
   */
  uint32_t getPressure();

  /**
```

```
    /**
     * @brief getHumidity Get humidity
     * @return Humidity in percent
     */
    float getHumidity();


    /**
     * @brief calAltitude Calculate altitude
     * @param seaLevelPressure Sea level pressure
     * @param pressure Pressure in pa
     * @return Altitude in meter
     */
    float calAltitude(float seaLevelPressure, uint32_t pressure);
```

Connection Diagram




RST
WAKE
INT
SDA
SCL
GND
VCC
www.DFRobot.com
5V
GND
SCL
SDA
SERVO_PWR
D
VCC
GND
13 12 *11 *10 *9 8 7 *6 *5 4 *3 2 TXRX
*PWM
L
BT/APC
SD
5V
GND
MOSI
SS
SCK
MISO
GND
5V
DTR
TXD
RXD
RUN
PROG
3V3
5V
PWR_IN
RST
VIN
GND
3.3V OUT
0 1 2 3 4 5
GND
VCC
A
DFROBOT
www.DFRobot.com
I/O Expansion Shield V7

1. Get Baseline

Why should we get the baseline?

Why should we get the baseline? Because once get it, we can show the air quality quickly after the sensor warm-up by inputting the baseline. Otherwise, it will cost a long time to read correctly when startup in polluted air.

During the first week of running the sensor, it is recommended to save a new baseline every 24 hours. After 1 week of operation, it can be saved every 1-28 days

> ⚠ **NOTE**:
>
> - Please place it in a fresh air environment (20 minutes or more) to obtain the baseline.
> - Different sensors, different measurement cycles have different baselines.

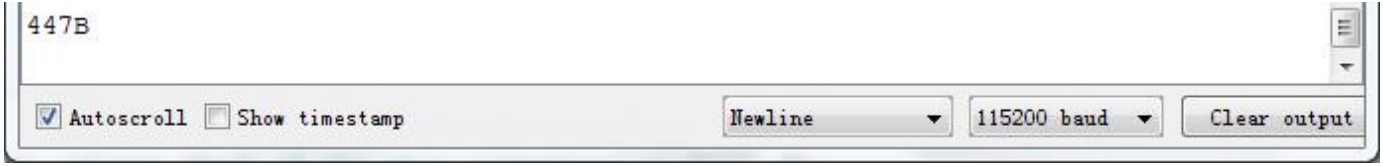
```
/*!
 * @file getBaseLine.ino
 * @brief Put the module in clear air and work a few minutes, wait for baseline doing not
 * @n Experiment phenomenon: get
 *
 * @copyright    Copyright (c) 2010 DFRobot Co.Ltd (http://www.dfrobot.com)
 * @licence     The MIT License (MIT)
 * @author [LuoYufeng](yufeng.luo@dfrobot.com)
 * @version  V0.1
 * @date  2019-07-19
 * @get from https://www.dfrobot.com
 * @url https://github.com/DFRobot/DFRobot_CCS811
 */
#include "DFRobot_CCS811.h"


/*
 * IIC address default 0x5A, the address becomes 0x5B if the ADDR_SEL is soldered.
 */
```

```
 */
//DFRobot_CCS811 CCS811(&Wire, /*IIC_ADDRESS=*/0x5A);
DFRobot_CCS811 CCS811;

void setup(void)

{
    Serial.begin(115200);
    /*Wait for the chip to be initialized completely, and then exit*/
    while(CCS811.begin() != 0){
        Serial.println("failed to init chip, please check if the chip connection is fine")
        delay(1000);
    }
}
void loop() {
    if(CCS811.checkDataReady() == true){
        /*!
         * @brief Set baseline
         * @return baseline in clear air
         */
        Serial.println(CCS811.readBaseLine(), HEX);

    } else {
        Serial.println("Data is not ready!");
    }
```

```
    //delay cannot be less than measurement cycle
    delay(1000);
}
```

Expected Results

After a while, the baseline reaches to a stable value.



2. Get Data

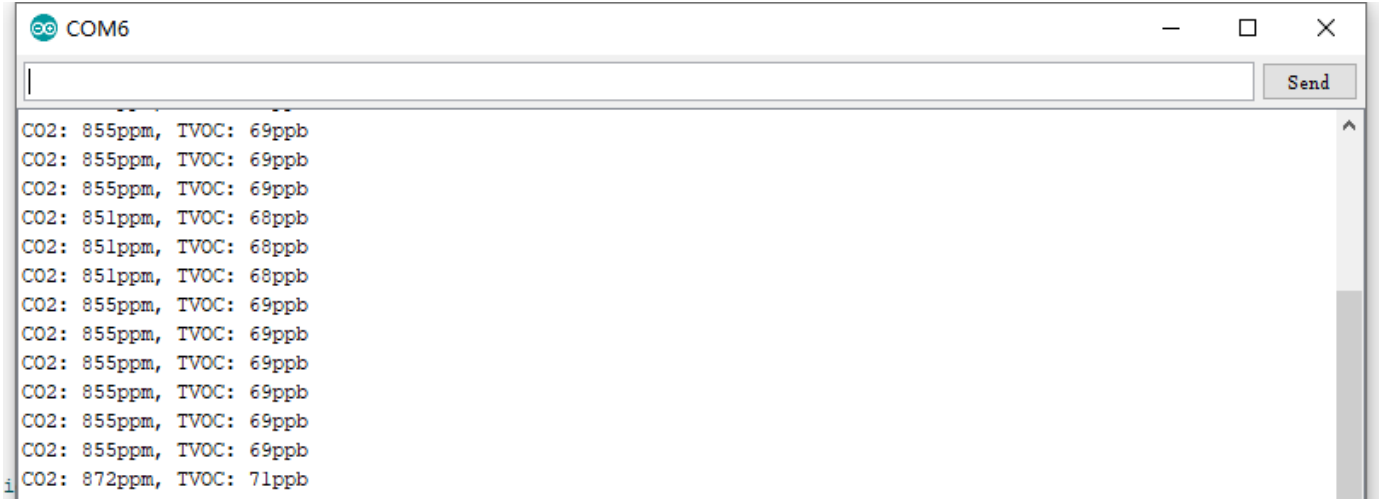
Input the baseline value to the function `sensor.writeBaseLine();` If you don't want to set the baseline, please disable this function in the sample program. The sensor will automatically calibrate the baseline, but it would be a pretty long process. after the function uploaded to UNO, open the serial port monitor to check the carbide dioxide concentration and TVOC concentration.

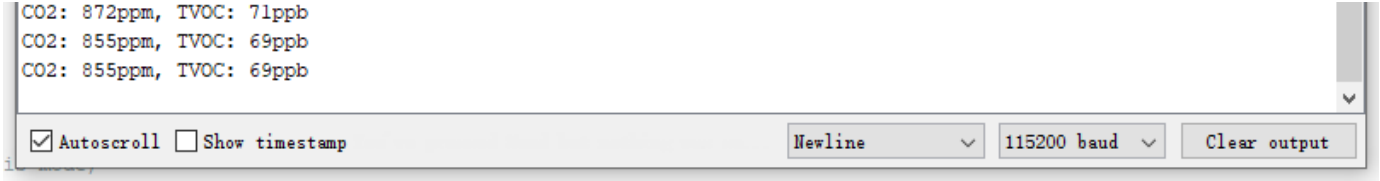
```
/*!
 * @file readData.ino
 * @brief Read the concentration of carbon dioxide and TVOC
 * @n Experiment phenomenon: read data every 0.5s, and print it out on serial port.
 *
 * @copyright    Copyright (c) 2010 DFRobot Co.Ltd (http://www.dfrobot.com)
 * @licence     The MIT License (MIT)
 * @author [LuoYufeng](yufeng.luo@dfrobot.com)
 * @version  V0.1
 * @date  2019-07-19
 * @get from https://www.dfrobot.com
 * @url https://github.com/DFRobot/DFRobot_CCS811
 */
#include "DFRobot_CCS811.h"

/*
 * IIC address default 0x5A, the address becomes 0x5B if the ADDR_SEL is soldered.
 */
//DFRobot_CCS811 CCS811(&Wire, /*IIC_ADDRESS=*/0x5A);
```

```
//DFRobot_CCS811 CCS811(&Wire, /*IIC_ADDRESS=*/0x5A);
DFRobot_CCS811 CCS811;

void setup(void)
{

    Serial.begin(115200);
    /*Wait for the chip to be initialized completely, and then exit*/
    while(CCS811.begin() != 0){
        Serial.println("failed to init chip, please check if the chip connection is fine");
        delay(1000);
    }
}
void loop() {
    if(CCS811.checkDataReady() == true){
        Serial.print("CO2: ");
        Serial.print(CCS811.getCO2PPM());
        Serial.print("ppm, TVOC: ");
        Serial.print(CCS811.getTVOCPPB());
        Serial.println("ppb");

    } else {
        Serial.println("Data is not ready!");
    }
    /*!
```

```
     * @brief Set baseline
     * @param get from getBaseline.ino
     */
    CCS811.writeBaseLine(0x447B);
    //delay cannot be less than measurement cycle

    delay(1000);
}
```

Expected Results





3. Concentration Alarm

Input the baseline value to the function `sensor.writeBaseLine();` Upload it to UNO, when the CO2 concentration moves from the current range (low, medium, high) to another range (more than 50 ppm), an interruption is generated and the current CO2 value will be printed.

NOTE: This example requires the INT pin of the sensor to be connected to the corresponding interrupt pin on the main board (in the sample, D2 of UNO is selected).

AVR Series Interrupt Pin and Interrupt Number

	328 Mainboards: Uno, Nano, Mini...	Mega2560	32u4 Mainboards: Leonardo...
Interrupt Pin	D2, D3	D2, D3, D21, D20, D19, D18	D3, D2, D0, D1, D7
Interrupt			

Number	0, 1 328 Mainboards: Uno,	0, 1, 2, 3, 4, 5 Mega2560	0, 1, 2, 3 ,4 32u4 Mainboards:

```
/*!
 * @file setInterrupt.ino
 * @brief Set interrupt parameter, when CO2 concentration range changes, get an interrupt
 * @n Experiment phenomenon: read data every 1s, and print it out on serial port.
 *
 * @copyright    Copyright (c) 2010 DFRobot Co.Ltd (http://www.dfrobot.com)
 * @licence     The MIT License (MIT)
 * @author [LuoYufeng](yufeng.luo@dfrobot.com)
 * @version  V1.0
 * @date  2019-07-13
 * @get from https://www.dfrobot.com
 * @url https://github.com/DFRobot/DFRobot_Sensor
 */
#include "DFRobot_CCS811.h"

volatile  int8_t GPIO1TRIG = 0;

/*
 * IIC address default 0x5A, the address becomes 0x5B if the ADDR_SEL is soldered
```

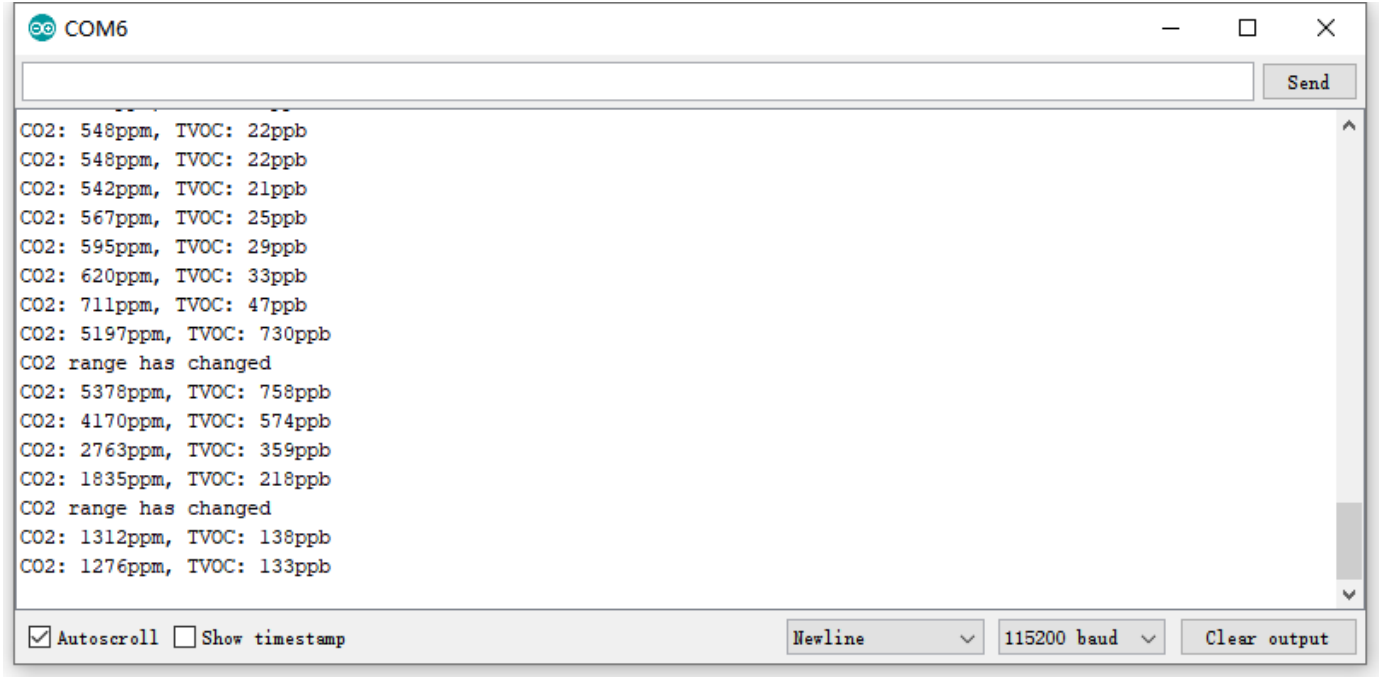
```
 * IIC address default 0x5A, the address becomes 0x5B if the ADDR_SEL is soldered.
 */
//DFRobot_CCS811 CCS811(&Wire, /*IIC_ADDRESS=*/0x5A);
DFRobot_CCS811 CCS811;


void setup(void)
{
    Serial.begin(115200);
    /*wait for the chip to be initialized completely, and then exit*/
    while(CCS811.begin() != 0){
        Serial.println("failed to init chip, please check if the chip connection is fine")
        delay(1000);
    }
    attachInterrupt(0, interrupt, RISING);
    /**
     * @brief Measurement parameter configuration
     * @param mode:in typedef enum{
     *              eClosed,      //Idle (Measurements are disabled in this mode)
     *              eCycle_1s,    //Constant power mode, IAQ measurement every second
     *              eCycle_10s,   //Pulse heating mode IAQ measurement every 10 seconds
     *              eCycle_60s,   //Low power pulse heating mode IAQ measurement every 60
     *              eCycle_250ms  //Constant power mode, sensor measurement every 250ms 1
     *          }eCycle_t;
     * @param thresh:0 for Interrupt mode operates normally; 1 for interrupt mode only ass
```

```
     * @param interrupt:0 for interrupt generation is disabled; 1 for the nINT signal is a
     */
    CCS811.setMeasurementMode(CCS811.eCycle_250ms, 1, 1);
    /**
     * @brief Set interrupt thresholds
     * @param lowToMed: interrupt triggered value in range low to middle
     * @param medToHigh: interrupt triggered value in range middle to high
     */
    CCS811.setThresholds(1500,2500);
}
void loop() {
    if(GPIO1TRIG == 1){
        Serial.println("CO2 range has changed");
        Serial.print("CO2: ");
        Serial.print(CCS811.getCO2PPM());
        Serial.print("ppm, TVOC: ");
        Serial.print(CCS811.getTVOCPPB());
        Serial.println("ppb");
        delay(1000);
    }
    GPIO1TRIG = 0;
    Serial.print("CO2: ");
    Serial.print(CCS811.getCO2PPM());
    Serial.print("ppm, TVOC: ");
```

```
    Serial.print(CCS811.getTVOCPPB());
    Serial.println("ppb");
    CCS811.writeBaseLine(0x447B);
    delay(1000);
}



void interrupt(){
  GPIO1TRIG = 1;
}
```

Expected Results

When you blow air to the sensor, the range of CO2 concentration changes, which results in an interruption; when the gas concentration decreases, there is also an interruption.



4. BME280 Read Data

4. BME280 Read Data

Note：please check the IIC address of BME280 in the program before using. Chip initialization will fail when the hardware IIC address is not the same as the address in the codes.

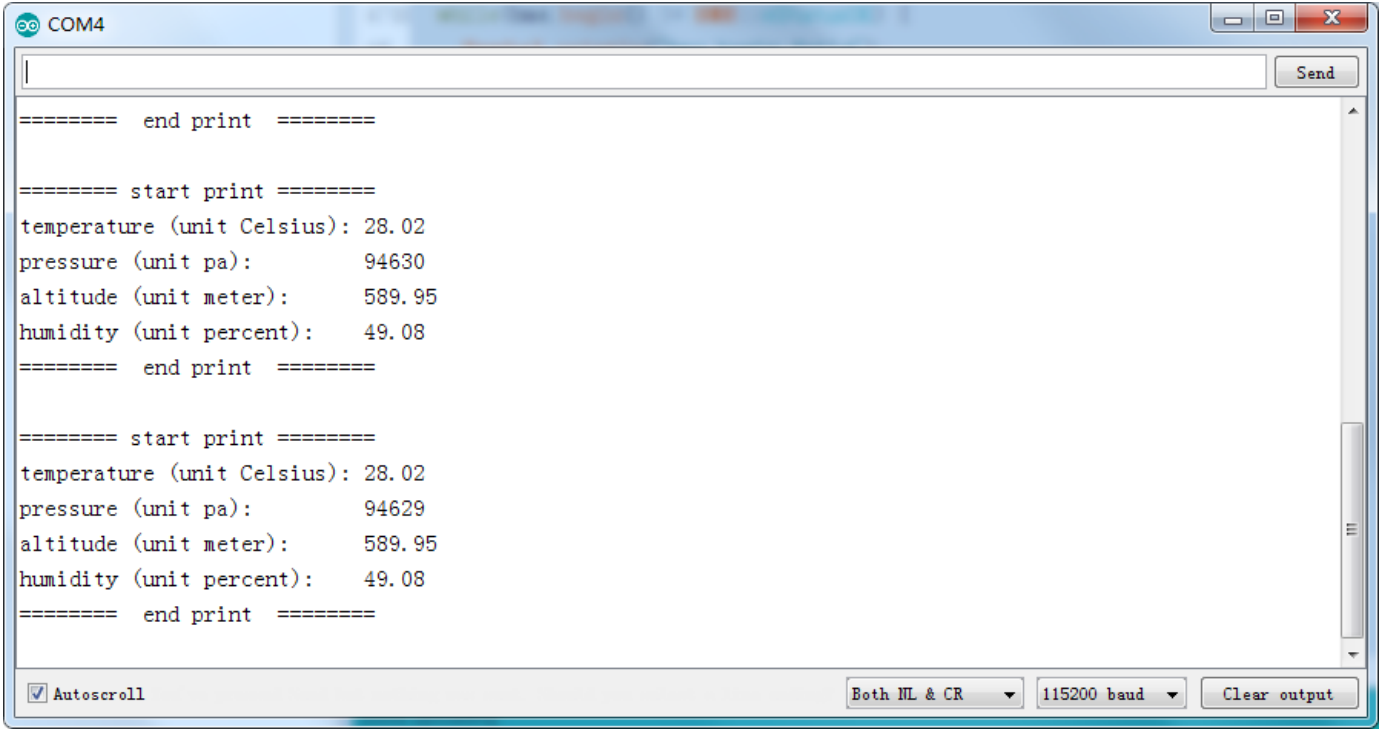
Run the codes, then the related data will be printed.

```
/*!
 * raed_data_i2c.ino
 *
 * Download this demo to test read data from bme280, connect sensor through IIC interface
 * Data will print on your serial monitor
 *
 * Copyright   [DFRobot](http://www.dfrobot.com), 2016
 * Copyright   GNU Lesser General Public License
 *
 * version  V1.0
 * date  12/03/2019
 */

#include "DFRobot_BME280.h"
#include "Wire.h"

typedef DFRobot_BME280_IIC    BME;    // ******** use abbreviations instead of full names

BME   bme(&Wire, 0x76);   // select TwoWire peripheral and set sensor address
```

```
BME   bme(&Wire, 0x76);   // select TwoWire peripheral and set sensor address

#define SEA_LEVEL_PRESSURE    1015.0f

// show last sensor operate status

void printLastOperateStatus(BME::eStatus_t eStatus)
{
  switch(eStatus) {
  case BME::eStatusOK:    Serial.println("everything ok"); break;
  case BME::eStatusErr:   Serial.println("unknow error"); break;
  case BME::eStatusErrDeviceNotDetected:    Serial.println("device not detected"); break;
  case BME::eStatusErrParameter:    Serial.println("parameter error"); break;
  default: Serial.println("unknow status"); break;
  }
}

void setup()
{
  Serial.begin(115200);
  bme.reset();
  Serial.println("bme read data test");
  while(bme.begin() != BME::eStatusOK) {
    Serial.println("bme begin faild");
    printLastOperateStatus(bme.lastOperateStatus);
```

```
    delay(2000);
  }
  Serial.println("bme begin success");
  delay(100);
}


void loop()
{
  float   temp = bme.getTemperature();
  uint32_t    press = bme.getPressure();
  float   alti = bme.calAltitude(SEA_LEVEL_PRESSURE, press);
  float   humi = bme.getHumidity();

  Serial.println();
  Serial.println("======== start print ========");
  Serial.print("temperature (unit Celsius): "); Serial.println(temp);
  Serial.print("pressure (unit pa):         "); Serial.println(press);
  Serial.print("altitude (unit meter):      "); Serial.println(alti);
  Serial.print("humidity (unit percent):    "); Serial.println(humi);
  Serial.println("========  end print  ========");

  delay(1000);
}
```

Expected Results



FAQ

Q: Why the sensor init failed and the serial monitor printed "device not detected"?
A: Check whether the IIC address in the codes is the same as the sensor address.

More

- Schematics (https://dfimg.dfrobot.com/nobody/wiki/65c19ce0e8ca04e6cc662ff062f5bcb5.pdf)
- Dimension (https://dfimg.dfrobot.com/nobody/wiki/5425404295931b6b8d0160deb68310ce.pdf)
- BME280 Datasheet (https://dfimg.dfrobot.com/nobody/wiki/eebf7904aecb84aeebf5af3f6a19533f.pdf)
- Datasheet (https://dfimg.dfrobot.com/nobody/wiki/7334c560756596ba0cf3f1d2102d19dd.pdf)
- CCS811 Baseline (https://dfimg.dfrobot.com/nobody/wiki/ab83d61ed52c66c2dd4067eed25b0c35.pdf)

 Get **Multi-function Environmental Module - CCS811+BME280** (https://www.dfrobot.com/product-2064.html) from DFRobot Store or **DFRobot Distributor**. (https://www.dfrobot.com/index.php?route=information/distributorslogo)

Turn to the Top