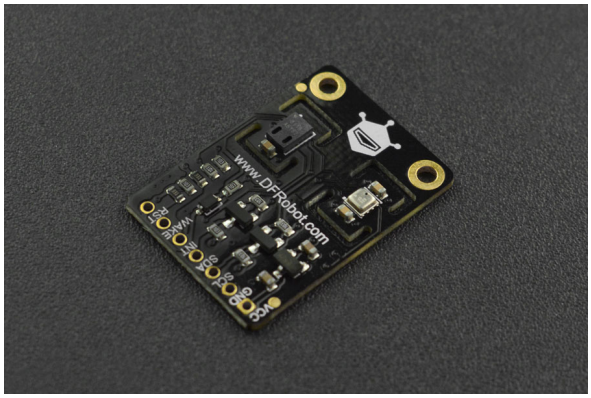
SKU:SEN0335 (https://www.dfrobot.com/product-2064.html)



(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 the ENS160+BME280 chip, this module features high accuracy, I2C interface, and fast Measurement. The BME280 can provide temperature and humidity compensation for ENS160 to improve the whole accuracy to a certain extent. It can be used to detect temperature, humidity, barometric pressure, altitude, TVOC and eCO2.

ENS160 is an air quality sensor based on ScioSense's new ENS160 sensor chip, which is specifically designed for indoor air quality monitoring and offers detection of multiple IAQ data (TVOC, eCO2, AQI). That the innovative TrueVOC™ technology combines the metal oxide (MOX) technology brings this sensor superior accuracy, fast response, anti-interference, etc.

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°C 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.

Air Quality Index

AQI Reference

Level	Description	Suggestion	Recommended

Level	Description	Suggestion	Recommended Stay Time
5	Extremely bad	In unavoidable circumstances, please strengthen ventilation	Not recommended to stay
4	Bad	Strengthen ventilation, find sources of pollution	Less than one month
3	Generally	Strengthen ventilation, close to the water source	Less than 12 months
2	Good	Maintain adequate ventilation	Suitable for long-term living
1	Excellent	No suggestion	Suitable for long-term living

eCO2/CO2 Concentration Reference

eCO2/CO2	Level	Suggestion
21500	Terrible	Indoor air pollution is serious and requires ventilation
1000-1500	Bad	Indoor air is polluted, ventilation is recommended
800-1000	Generally	Can be ventilated
600-800	Good	Keep it normal
400-600	Eexcellent	No suggestion

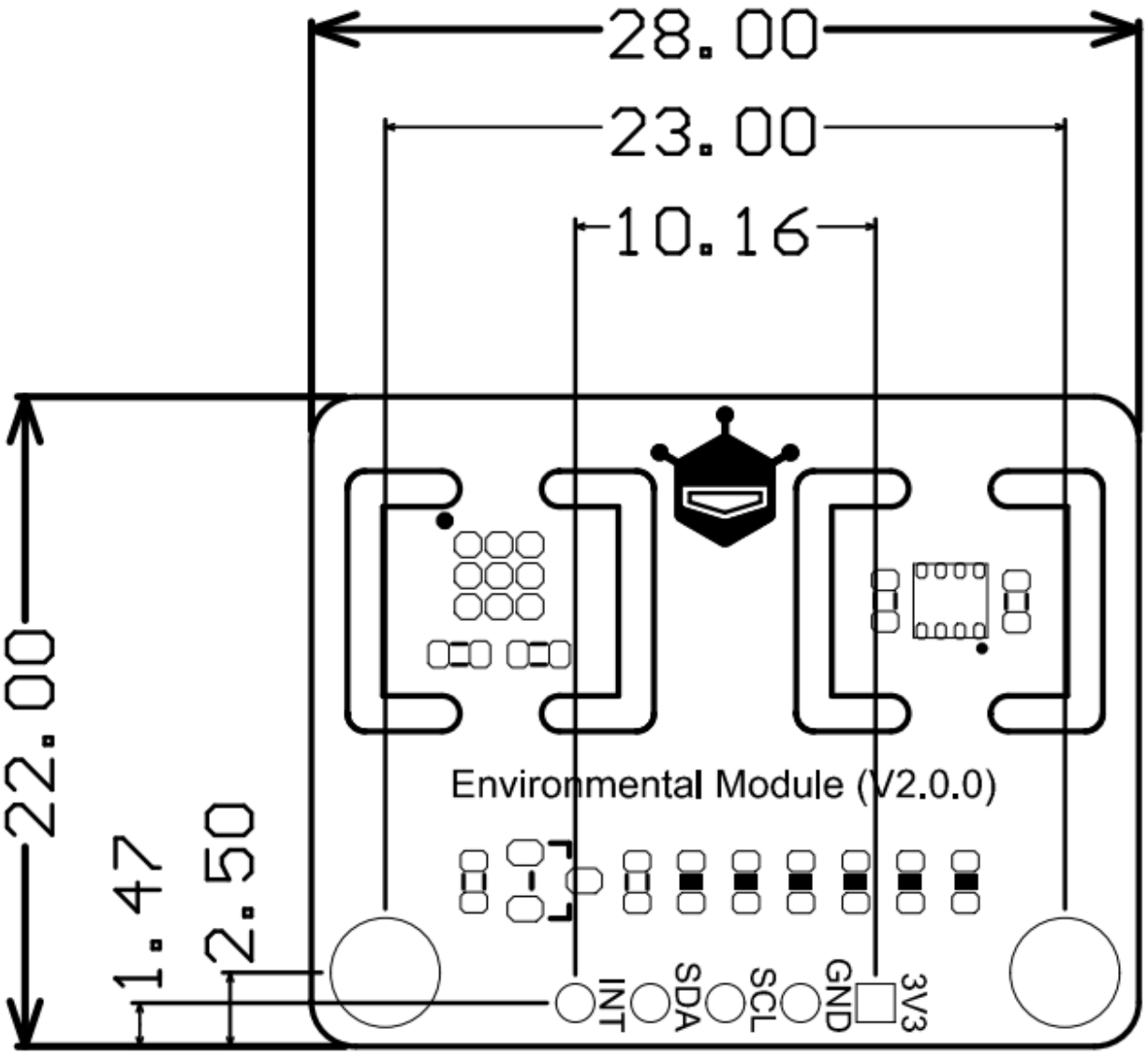
TVOC Concentration Reference

TOVC(ppb)	Effects on Human Health
>6000	Headaches and nerve problem
750-6000	Restless and headache
50-750	Restless and uncomfortable
<50	No effect

Applications

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

Specification



- Operating Voltage: 3.3V~5.5 V
- Working Current: <20mA
- Product Size: 22*28mm/0.87*1.10″

ENS160 Parameter:

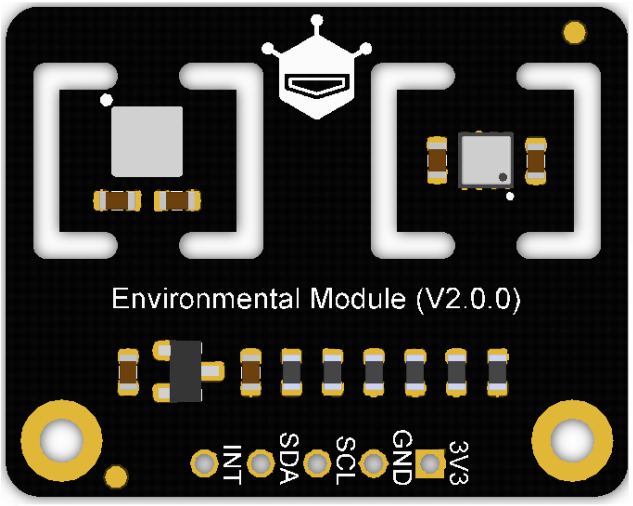
- Preheat Time: <3 minutes
- I2C Address: 0x53(default), 0X52
- Operating Temperature Range: -40°C~85°C

- Operating Humidity Range: 5%RH~95%RH
- eCO2 Measuring Range: 400ppm~65000ppm
- TVOC Measuring Range: 0ppb~65000ppb

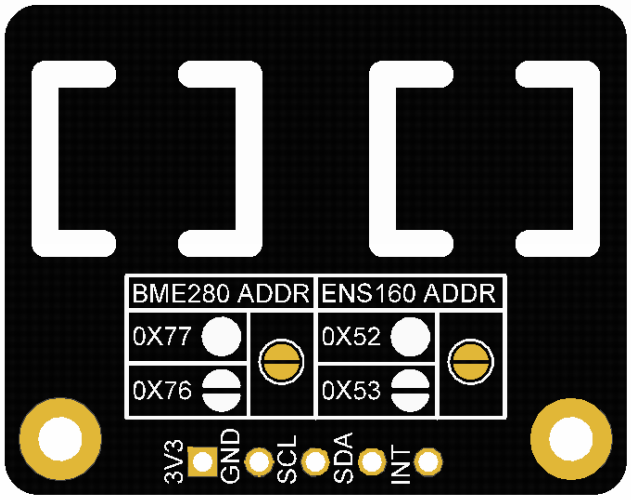
BME280 Parameter:

- I2CAddress： 0x76(default), 0X77
- Operating Temperature: -40°C~85°C
- Temperature Measuring Range: -40°C~+85°C, resolution of 0.1°C, deviation of ±0.5°C
- Humidity Measuring Range: 0~100%RH, resolution of 0.1%RH, deviation of ±2%RH
- Pressure Measuring Range: 300~1100hPa

Board Overview



TOP



BOTTOM

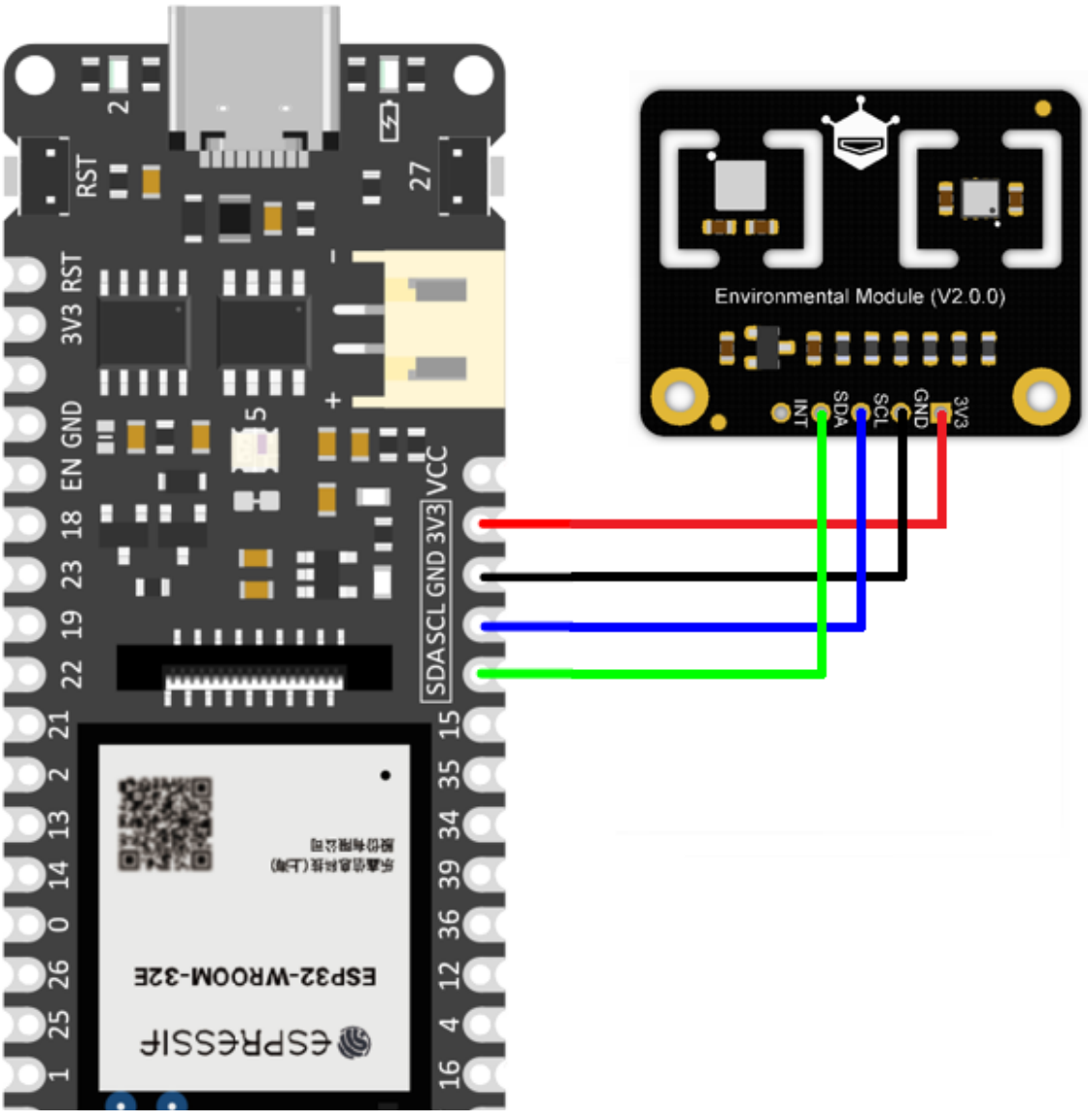
Num	Label	Description
1	3V3	+
2	GND	-
3	SCL	I2C clock line
4	SDA	I2C data line
5	INT	ENS160 Interrupt pin: interrupt in low level

Tutorial

Requirements

- Hardware
  - DFRduino UNO R3 (https://www.dfrobot.com/product-838.html) (or similar) x 1
  - ENS160 + BME280 Multi-function Environmental Sensor x 1
  - Jumper wires
- Software
  - Arduino IDE (https://www.arduino.cc/en/Main/Software)
  - Download and install the **ENS160 Library and examples** (https://github.com/DFRobot/DFRobot_ENS160).
  - 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))

Connection Diagram



1. Get Baseline

Burn the program to FireBeetle Board ESP32-E through Arduino IDE, open the serial port, we can see the printed.

Sensor operating status

Status	Description
0	Operate normally
1	Preheat, the first 3 minutes when powered on(No more inital startup status for the sensor)
2	Initial startup, the first 1 hours when powered on(the sensor will no longer be in this state after 24 hours of continuous operation, if there is a power failure during this period, the sensor will still enter the initial startup state after power-on again)

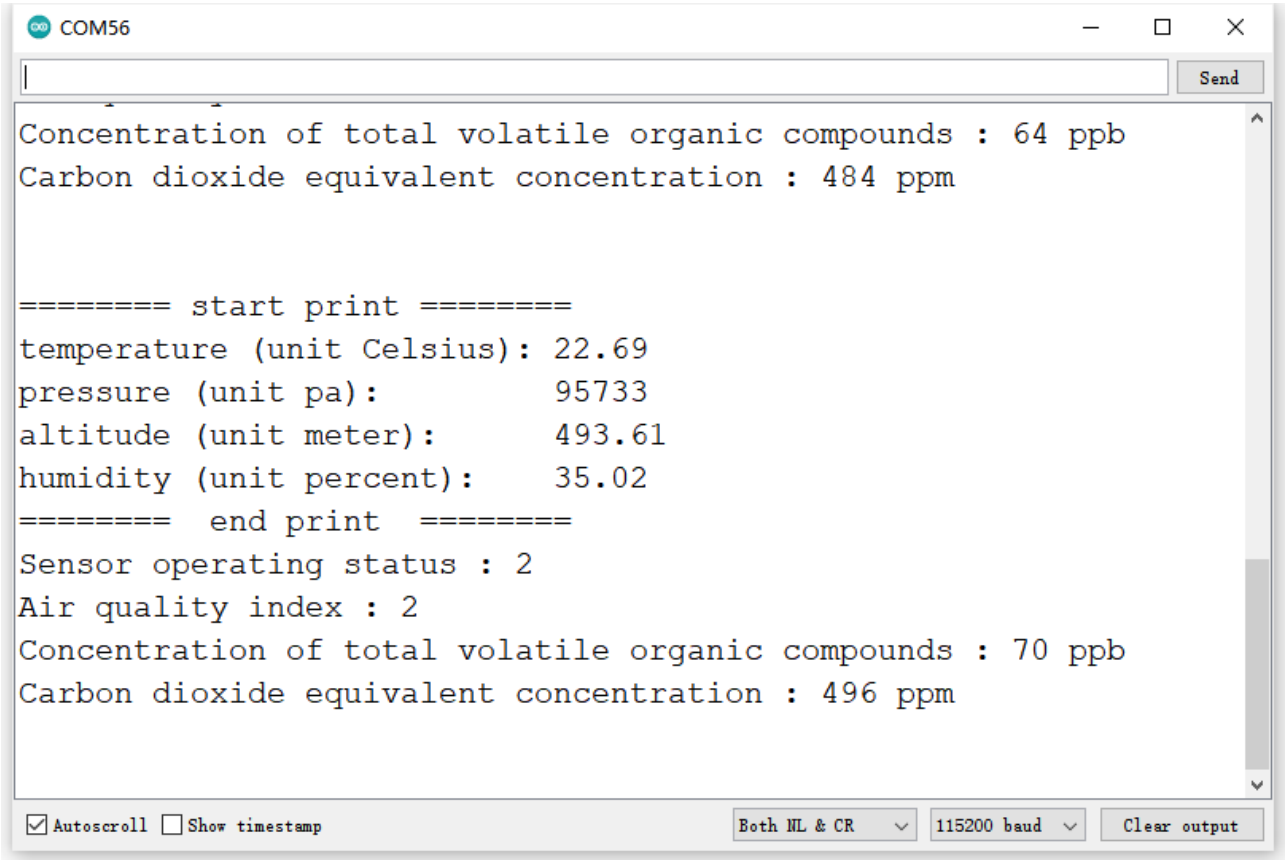
For more details, please refer to Chapter 10 in the Chip Manual (https://dfimg.dfrobot.com/nobody/wiki/0fc3080c820be7734ad56bbbbea32564.pdf).

```
#include <DFRobot_ENS160.h>
#include "DFRobot_BME280.h"

#define SEA_LEVEL_PRESSURE    1015.0f

DFRobot_ENS160_I2C ENS160(&Wire, /*i2cAddr*/ 0x53);
typedef DFRobot_BME280_IIC    BME;    // ******** use abbreviations instead of
BME   bme(&Wire, 0x76);   // select TwoWire peripheral and set sensor address

// 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 detecte
  case BME::eStatusErrParameter:    Serial.println("parameter error"); break;
  default: Serial.println("unknow status"); break;
  }
}

void setup(void)
{
  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);
  // Init the sensor
  while( NO_ERR != ENS160.begin() ){
    Serial.println("Communication with device failed, please check connection")
    delay(3000);
  }

  /**
   * Set power mode
   * mode Configurable power mode:
   *   ENS160_SLEEP_MODE: DEEP SLEEP mode (low power standby)
   *   ENS160_IDLE_MODE: IDLE mode (low-power)
   *   ENS160_STANDARD_MODE: STANDARD Gas Sensing Modes
   */
  ENS160.setPWRMode(ENS160_STANDARD_MODE);
```

```
  /**
   * Users write ambient temperature and relative humidity into ENS160 for cali
   * ambientTemp Compensate the current ambient temperature, float type, unit:
   * relativeHumidity Compensate the current ambient temperature, float type, u
   */
  ENS160.setTempAndHum(/*temperature=*/bme.getTemperature(), /*humidity=*/bme.g

}

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  ========");
  /**
   * Get the sensor operating status
   * Return value: 0-Normal operation,
   *         1-Warm-Up phase, first 3 minutes after power-on.
   *         2-Initial Start-Up phase, first full hour of operation after init
   * note: Note that the status will only be stored in the non-volatile memory
   *       operation. If unpowered before conclusion of said period, the ENS160
   *       after re-powering.
   */
  uint8_t Status = ENS160.getENS160Status();
  Serial.print("Sensor operating status : ");
  Serial.println(Status);
  /**
   * Get the air quality index
   * Return value: 1-Excellent, 2-Good, 3-Moderate, 4-Poor, 5-Unhealthy
   */
  uint8_t AQI = ENS160.getAQI();
  Serial.print("Air quality index : ");
  Serial.println(AQI);

  /**
   * Get TVOC concentration
   * Return value range: 0-65000, unit: ppb
   */
  uint16_t TVOC = ENS160.getTVOC();
  Serial.print("Concentration of total volatile organic compounds : ");
  Serial.print(TVOC);
  Serial.println(" ppb");
```

```
  /**
   * Get CO2 equivalent concentration calculated according to the detected data
   * Return value range: 400–65000, unit: ppm
   * Five levels: Excellent(400 - 600), Good(600 - 800), Moderate(800 - 1000),
   *              Poor(1000 - 1500), Unhealthy(> 1500)
   */
  uint16_t ECO2 = ENS160.getECO2();
  Serial.print("Carbon dioxide equivalent concentration : ");
  Serial.print(ECO2);
  Serial.println(" ppm");
  Serial.println();

  delay(1000);
}
```

Expected Results



- About API Function List

```
/**
   * @fn setPWRMode
   * @brief Set power supply mode
   * @param mode Configurable power mode:
   * @n       ENS160_SLEEP_MODE: DEEP SLEEP mode (low power standby)
   * @n       ENS160_IDLE_MODE: IDLE mode (low-power)
   * @n       ENS160_STANDARD_MODE: STANDARD Gas Sensing Modes
   * @return None
   */
  void setPWRMode(uint8_t mode);

    /**
   * @fn setTempAndHum
   * @brief Users write ambient temperature and relative humidity into ENS160 f
   * @param ambientTemp Compensate the current ambient temperature, float type,
   * @param relativeHumidity Compensate the current ambient temperature, float
   * @return None
   */
  void setTempAndHum(float ambientTemp, float relativeHumidity);

  /**
   * @fn getENS160Status
   * @brief This API is used to get the sensor operating status
   * @return Operating status:
   * @n        eNormalOperation: Normal operation;
   * @n        eWarmUpPhase: Warm-Up phase;
   * @n        eInitialStartUpPhase: Initial Start-Up phase;
   * @n        eInvalidOutput: Invalid output
   */
  uint8_t getENS160Status(void);

  /**
   * @fn getAQI
   * @brief Get the air quality index calculated on the basis of UBA
   * @return Return value range: 1-5 (Corresponding to five levels of Excellent
   */
  uint8_t getAQI(void);

  /**
   * @fn getTVOC
   * @brief Get TVOC concentration
   * @return Return value range: 0-65000, unit: ppb
   */
  uint16_t getTVOC(void);

  /**
   * @fn getECO2
```

```
    * @brief Get CO2 equivalent concentration calculated according to the detect
    * @return Return value range: 400–65000, unit: ppm
    * @note Five levels: Excellent(400 - 600), Good(600 - 800), Moderate(800 - 1
    * @n                 Poor(1000 - 1500), Unhealthy(> 1500)

    */
   uint16_t getECO2(void);
```

FAQ

Q: Why the sensor init failed and the serial monitor printed "device not detected"?
A: Check whether the I2C address in the codes is the same as the sensor address.

More

- Schematics (https://dfimg.dfrobot.com/nobody/wiki/21ed8a5cc736ad8e58bcd3562bae6397.pdf)
- Dimension (https://dfimg.dfrobot.com/nobody/wiki/425ac461f1068fd0edd333ec00883f1c.pdf)
- BME280 Datasheet (https://dfimg.dfrobot.com/nobody/wiki/eebf7904aecb84aeebf5af3f6a19533f.pdf)
- ENS160 Manual.pdf (https://dfimg.dfrobot.com/nobody/wiki/cbe10f01b67c3fee6d365039eb54f52c.pdf)
- PCB Package (https://dfimg.dfrobot.com/nobody/wiki/6d6947f0849cbb42b894c59a0b1df010.zip)

Get **Multi-function Environmental Module - ENS160+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