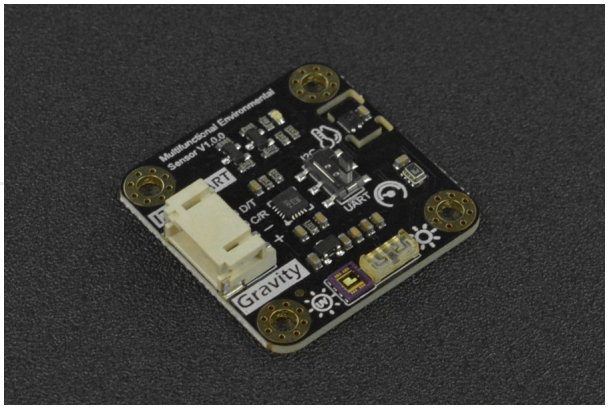
SKU:SEN0501 (https://www.dfrobot.com/product-2528.html)

(https://www.dfrobot.com/product-2528.html)



Introduction

This multifunctional environmental sensor comprises SHTC3 temperature & humidity sensor, BMP280 atmospheric pressure sensor, VEML7700 light sensor, and ultraviolet sensor(V1.0: ML8511, V2.0: LTR390-UV-01) into one and offers 5 kinds of environmental parameters. Professional sensor chips are selected for each kind of parameter measurement. The reasonable layout and heat conduction of the main chip are carefully considered in the circuit design, which effectively guarantees the accuracy of sensor data.

The product has an MCU processing chip onboard that converts the raw sensor data into values with the standard unit so you can directly use them. For example, °C and °F for temperature, % for humidity, Kpa for atmospheric pressure, lx for light illuminance, and mw/cm² for ultraviolet.

The environmental sensor supports two communication methods, UART and I2C. There are two versions: Gravity and Fermion(breakout), also complete Arduino and Python libraries are provided.

Exquisite and small, you can use it to make indoor and outdoor environmental monitoring systems, or for your environmental monitoring topics. This DFRobot environmental sensor can greatly simplify the wirings and codes of your project.

Please kindly note that the multifunctional environmental sensor V2.0 utilizes LTR390-UV-01 ultraviolet sensor, while the V1.0 uses the ML8511 sensor.

Features

- Gravity interface, no need to solder, plug and play.
- Switchable I2C and UART two output modes
- Highly integrated module, can test a variety of data at the same time (temperature, humidity, atmospheric pressure, altitude, ultraviolet intensity, ambient light intensity)
- Reasonable layout and high precision.

Application

- Home indoor and outdoor environment detection system
- Environmental monitoring work

register table

Reg	regAddr	name	R/W	description
inputReg	0x0002	deviceAddr	R	0x0022
inputReg	0x0005	VERSION	R	Firmware version number: 0x1000 represents V1.0.0.0
inputReg	0x0008	UV_INTENSITY	R	Ultraviolet Index (mW/m²)
inputReg	0x0009	AMBIENT_LIGHT	R	original light intensity value
inputReg	0x000A	TEMP	R	original temperature value
inputReg	0x000B	HUMIDITY	R	original humidity value
inputReg	0x000C	ATMOSPHERICPRESSURE	R	atmospheric pressure (hPa)
	baud rate: 9600			

V1.0

Specification

- Working voltage: 3.3~5V DC
- Working current: 45mA
- Output signal: I2C, UART
- Working temperature: -20~70°C
- Product size: 32mm×32mm / 1.26×1.26 inch
- Mounting hole size: M3 (diameter 3mm / 0.12 inch), the distance between the centers of the mounting holes is 25mm / 0.98 inch

Atmospheric pressure sensor

Atmospheric pressure unit relationship: 1000pa = 10hpa = 1kpa

- Sensor chip model: BMP280
- Atmospheric pressure measurement range: 3000~1100 hPa
- Relative accuracy of atmospheric pressure: ±0.12 hPa
- Absolute atmospheric pressure accuracy: ±1 hPa

Temperature & Humidity Sensor

- Sensor chip model: SHT-C3
- Relative humidity accuracy: ±2%RH
- Relative humidity resolution: 0.01%RH
- Relative humidity measurement range: 0~100
- Temperature accuracy: ±0.2°C

- Temperature resolution: 0.01°C
- Temperature measurement range: -40~125°C

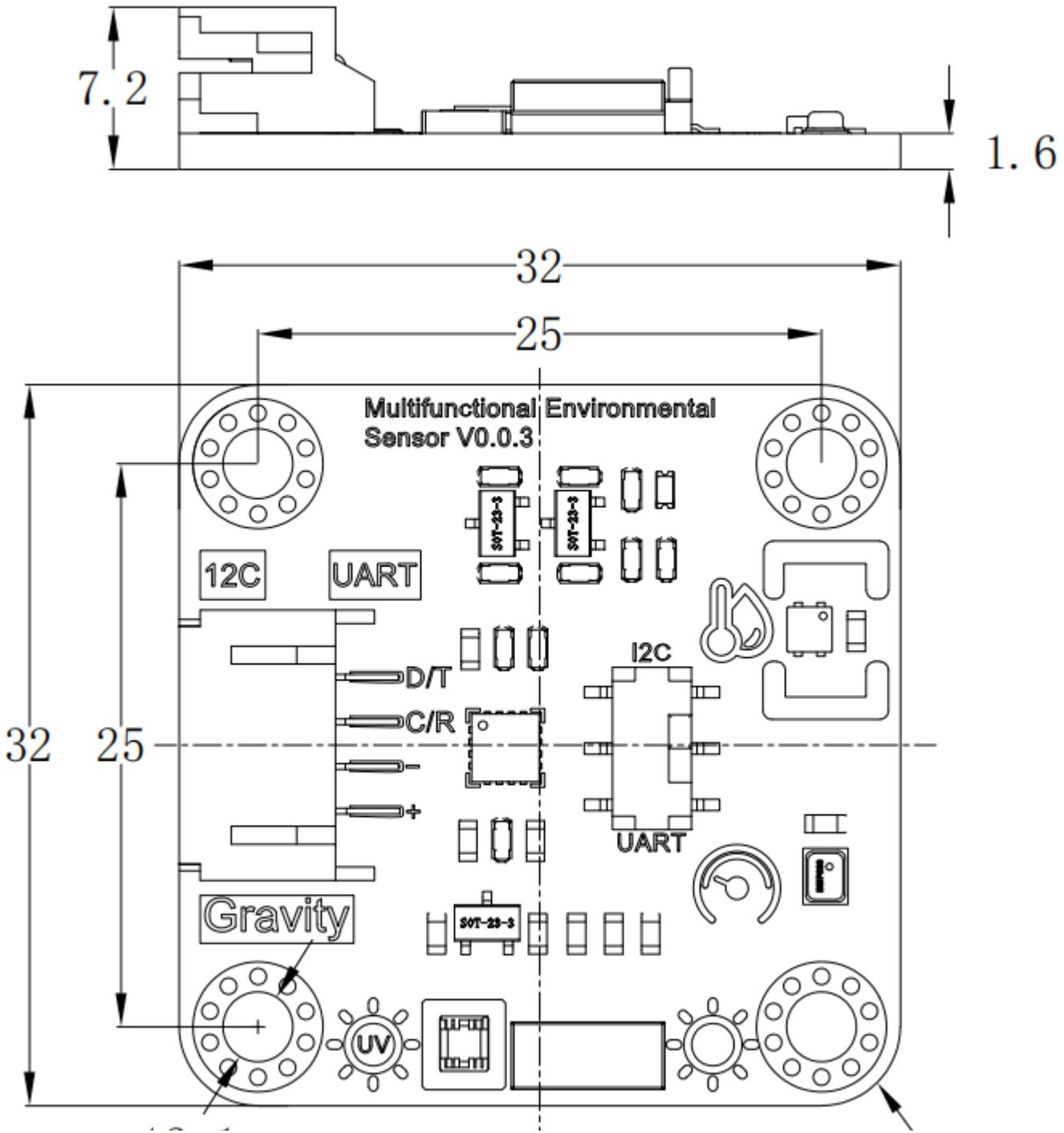
Ultraviolet Sensor

- Sensor chip model: ML8511
- Ultraviolet sensitive wavelength UV-A (320-400nm), UV-B (280-320nm)
- Ultraviolet output unit: mW/cm²

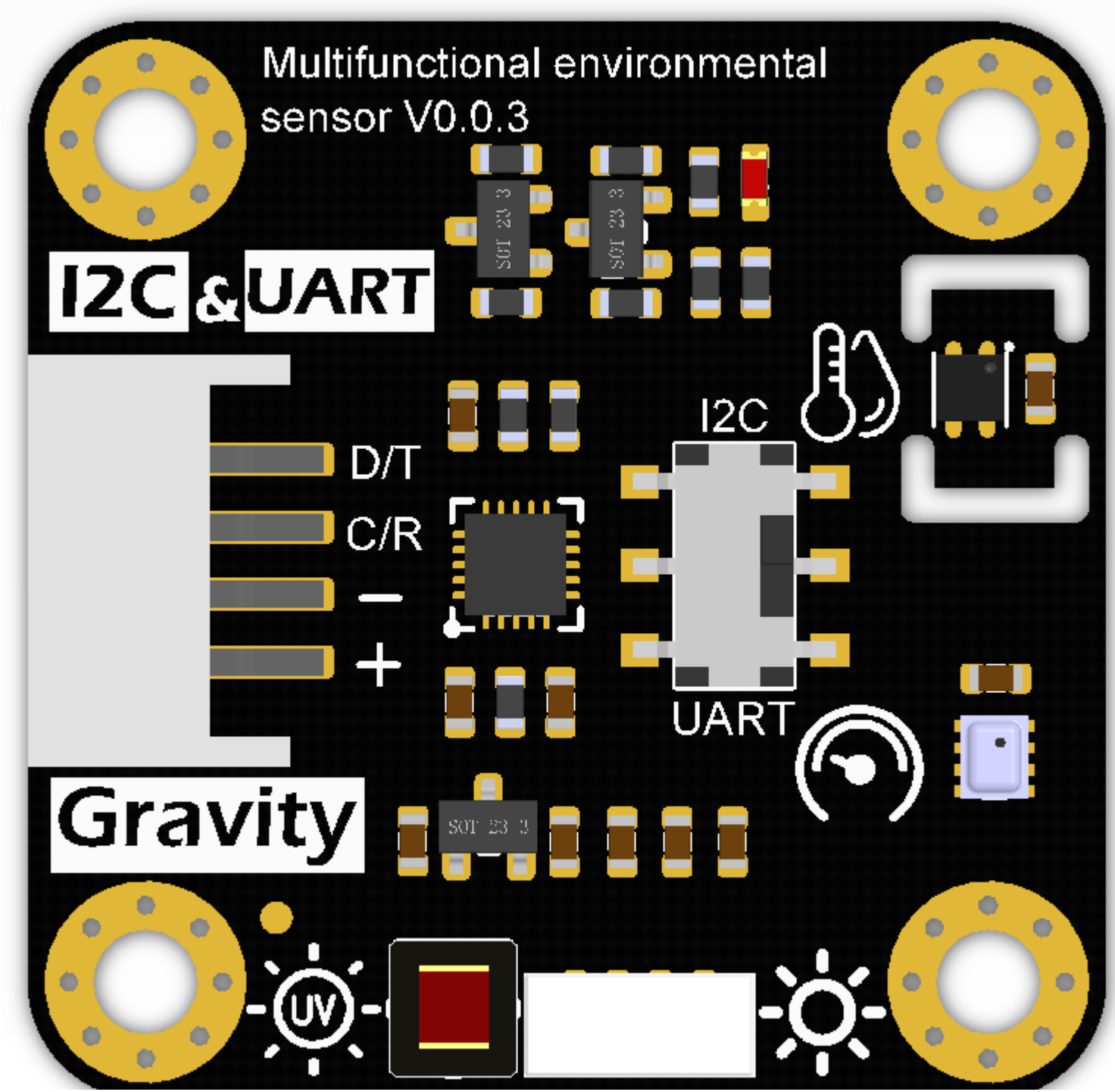
Ambient light sensor

- Sensor chip model: VEML7700
- Ambient light accuracy: 0.0036 lx/ct
- Ambient light range: 0~120 klx

Dimension



Overview



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

Tutorial

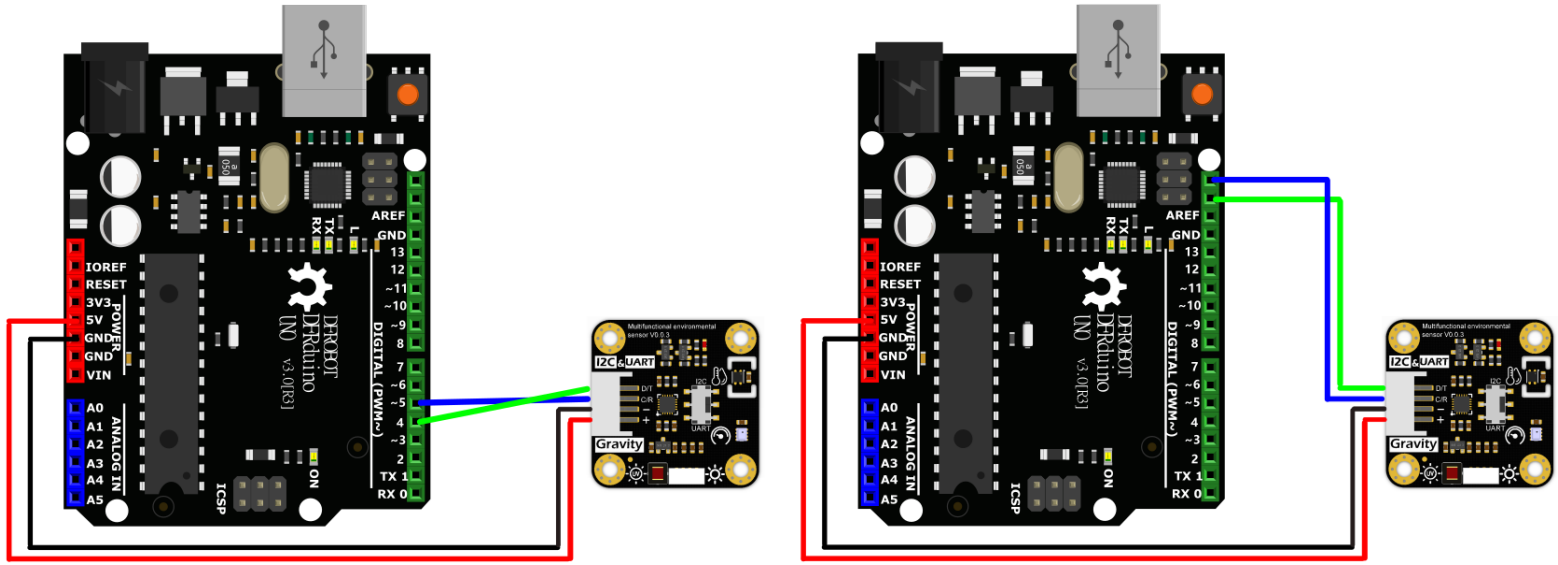
Download the program to DFRuino UNO, open the serial monitor to check various environmental parameters.

Requirements

- **Hardware**
  - DFRduino UNO R3 (https://www.dfrobot.com/product-838.html) (or similar) x 1
  - SEN0501 Multifunctional environmental sensor x1
  - Wires
- **Software**
  - Arduino IDE (https://www.arduino.cc/en/Main/Software)
  - Download and install DFRobot RTU library (https://github.com/DFRobot/DFRobot_RTU)
  - Download and install the **DFRobot_EnvironmentalSensor Library** (https://github.com/cdjq/DFRobot_EnvironmentalSensor) (About how to install the library? (https://www.arduino.cc/en/Guide/Libraries#.UxU8mdzF9H0))

Read Sensor Data via I2C/UART

Connection for UART and I2C



UART on the left, I2C on the right.

- Click to check more wiring diagrams (https://dfimg.dfrobot.com/nobody/wiki/f35bc83177b9866250cc8c04703f6493.png)

Switch Communication Mode

About UART/I2C mode switching:

1. The default mode in the code is UART. Dial the switch to UART side to use it.
2. For using I2C, dial the switch to I2C side and replace the 1 at the beginning of the code with 0, as shown below:

```
#define MODESWITCH        /*UART:*/1 /*I2C: 0*/   to
#define MODESWITCH        /*UART:*/0 /*I2C: 0*/
```

Steps:

Note: Different motherboards correspond to different wiring pins (refer to “more wiring diagrams” above). In the Arduino IDE, they also correspond to different motherboard options and different ports (COM). The other steps are the same as DFRuino UNO.

- Connect the module and DFRuino UNO according to the wiring method above.
- Open the Arduino IDE and upload the following code to DFRuino UNO.
- Open the serial port monitor of Arduino IDE, adjust the baud rate to 115200, and observe the serial port printing result.

Sample Code

Function: Print all data obtained by the module

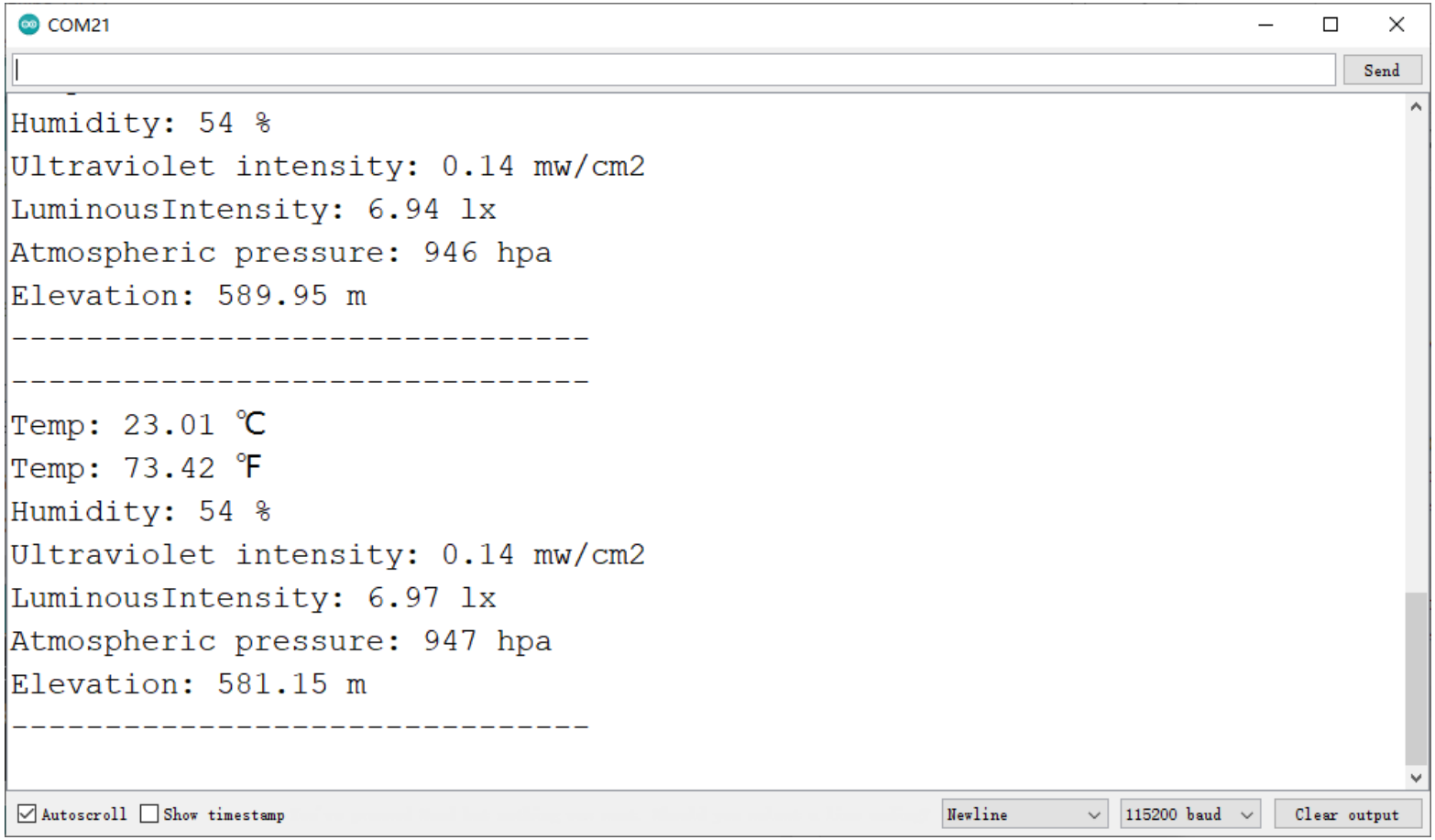
```
/*!
 *@file  read_data.ino
 *@brief This demo shows how to get data of the SEN0500/SEN0501 sensor and outputs data throu
 *@n Print the data returned by SEN0500/SEN0501 in the serial port monitor.
 * @n connected table
 * ---------------------------------------------------------------------------------------
 *    board   |             MCU                | Leonardo/Mega2560/M0 |    UNO    | ESP8266 |
 *     VCC    |            3.3V/5V             |        VCC           |    VCC    |   VCC   |
 *     GND    |              GND               |        GND           |    GND    |   GND   |
 *     RX     |              TX                |     Serial1 TX1      |     5     |   5/D6  |
 *     TX     |              RX                |     Serial1 RX1      |     4     |   4/D7  |
 * ---------------------------------------------------------------------------------------
 *
 * @copyright   Copyright (c) 2021 DFRobot Co.Ltd (http://www.dfrobot.com)
 * @license     The MIT License (MIT)
 * @author      [TangJie](jie.tang@dfrobot.com)
 * @version     V1.0
 * @date        2021-08-31
 * @url         https://github.com/DFRobot/DFRobot_EnvironmentalSensor
 */
#include "DFRobot_EnvironmentalSensor.h"
#if defined(ARDUINO_AVR_UNO)||defined(ESP8266)
#include <SoftwareSerial.h>
#endif

#define MODESWITCH        /*UART:*/1 /*I2C: 0*/

#if MODESWITCH
#if defined(ARDUINO_AVR_UNO)||defined(ESP8266)
  SoftwareSerial mySerial(/*rx =*/4, /*tx =*/5);
  DFRobot_EnvironmentalSensor environment(/*addr =*/SEN0500/SEN0501_DEFAULT_DEVICE_ADDRESS, /
#else
  DFRobot_EnvironmentalSensor environment(/*addr =*/SEN0500/SEN0501_DEFAULT_DEVICE_ADDRESS, /
#endif
#else
```

```
DFRobot_EnvironmentalSensor environment(/*addr = */SEN0500/SEN0501_DEFAULT_DEVICE_ADDRESS, /*
#endif
void setup()
{
#if MODESWITCH
  //Init MCU communication serial port
#if defined(ARDUINO_AVR_UNO)||defined(ESP8266)
  mySerial.begin(9600);
#elif defined(ESP32)
  Serial1.begin(9600, SERIAL_8N1, /*rx =*/D3, /*tx =*/D2);
#else
  Serial1.begin(9600);
#endif
#endif
  Serial.begin(115200);
  while(environment.begin() != 0){
    Serial.println(" Sensor initialize failed!!");
    delay(1000);
  }
  Serial.println(" Sensor  initialize success!!");
}

void loop()
{
  //Print the data obtained from sensor
  Serial.println("-----------------------------");
  Serial.print("Temp: ");
  Serial.print(environment.getTemperature(TEMP_C));
  Serial.println(" ℃");
  Serial.print("Temp: ");
  Serial.print(environment.getTemperature(TEMP_F));
  Serial.println(" ℉");
  Serial.print("Humidity: ");
  Serial.print(environment.getHumidity());
  Serial.println(" %");
  Serial.print("Ultraviolet intensity: ");
```

```
  Serial.print(environment.getUltravioletIntensity());
  Serial.println(" mw/cm2");
  Serial.print("LuminousIntensity: ");
  Serial.print(environment.getLuminousIntensity());
  Serial.println(" lx");
  Serial.print("Atmospheric pressure: ");
  Serial.print(environment.getAtmospherePressure(HPA));
  Serial.println(" hpa");
  Serial.print("Altitude: ");
  Serial.print(environment.getElevation());
  Serial.println(" m");
  Serial.println("------------------------------");
  delay(500);
}
```

Expected Results

The read data will be dipalyed on the serial monitor.



Function Library Name Definition

```
/**
   * @fn begin
   * @brief Init SEN0500/SEN0501 sensor
   * @return Return init status
   * @retval 0  Succeed
   * @retval -1 failed
   */
  int8_t begin(void);

  /**
   * @fn getTemperature
   * @brief Get SEN0500/SEN0501 temperature data
   * @param units Temperature data unit select
   * @n     TEMP_C ℃
   * @n     TEMP_F ℉
   * @return Return the obtained temperature data
   */
  float getTemperature(uint8_t unist);

  /**
   * @fn getHumidity
   * @brief Get SEN0500/SEN0501 humidity data
   * @return Return the obtained humidity data
   */
  float getHumidity(void);

  /**
   * @fn getUltravioletIntensity
   * @brief Get SEN0500/SEN0501 UV intensity index data
   * @return Return the obtained UV intensity index data
   */
  float getUltravioletIntensity(void);
```

```
/**
 * @fn getLuminousIntensity
 * @brief Get SEN0500/SEN0501 luminous intensity data
 * @return Return the obtained luminous intensity data
 */
float getLuminousIntensity(void);

/**
 * @fn getAtmospherePressure
 * @brief Get SEN0500/SEN0501 atmosphere pressure data
 * @param units Atmosphere pressure data unit select
 * @n            HPA: Hectopascal
 * @n            KPA: Kilopascal
 * @return Return the obtained atmosphere pressure data
 */
uint16_t getAtmospherePressure(uint8_t units);

/**
 * @fn getElevation
 * @brief Get SEN0500/SEN0501 altitude data
 * @return Return the obtained altitude data
 */
float getElevation(void);
```

V2.0

Specification

- Working voltage: 3.3-5V DC
- Working current: about 4mA
- Output signal: I2C, UART
- Working temperature: -20 to 70°C

- Product size: 32mm×32mm / 1.26×1.26 inch
- Mounting hole size: M3 (diameter 3mm / 0.12 inch), the distance between the centers of the mounting holes is 25mm / 0.98 inch

Atmospheric pressure sensor

Atmospheric pressure unit relationship: 1000pa = 10hpa = 1kpa

- Sensor chip model: BMP280
- Atmospheric pressure measurement range: 3000~1100 hPa
- Relative accuracy of atmospheric pressure: ±0.12 hPa
- Absolute atmospheric pressure accuracy: ±1 hPa

Temperature & Humidity Sensor

- Sensor chip model: SHT-C3
- Relative humidity accuracy: ±2%RH
- Relative humidity resolution: 0.01%RH
- Relative humidity measurement range: 0-100
- Temperature accuracy: ±0.2°C
- Temperature resolution: 0.01°C
- Temperature measurement range: -40-125°C

Ultraviolet Sensor

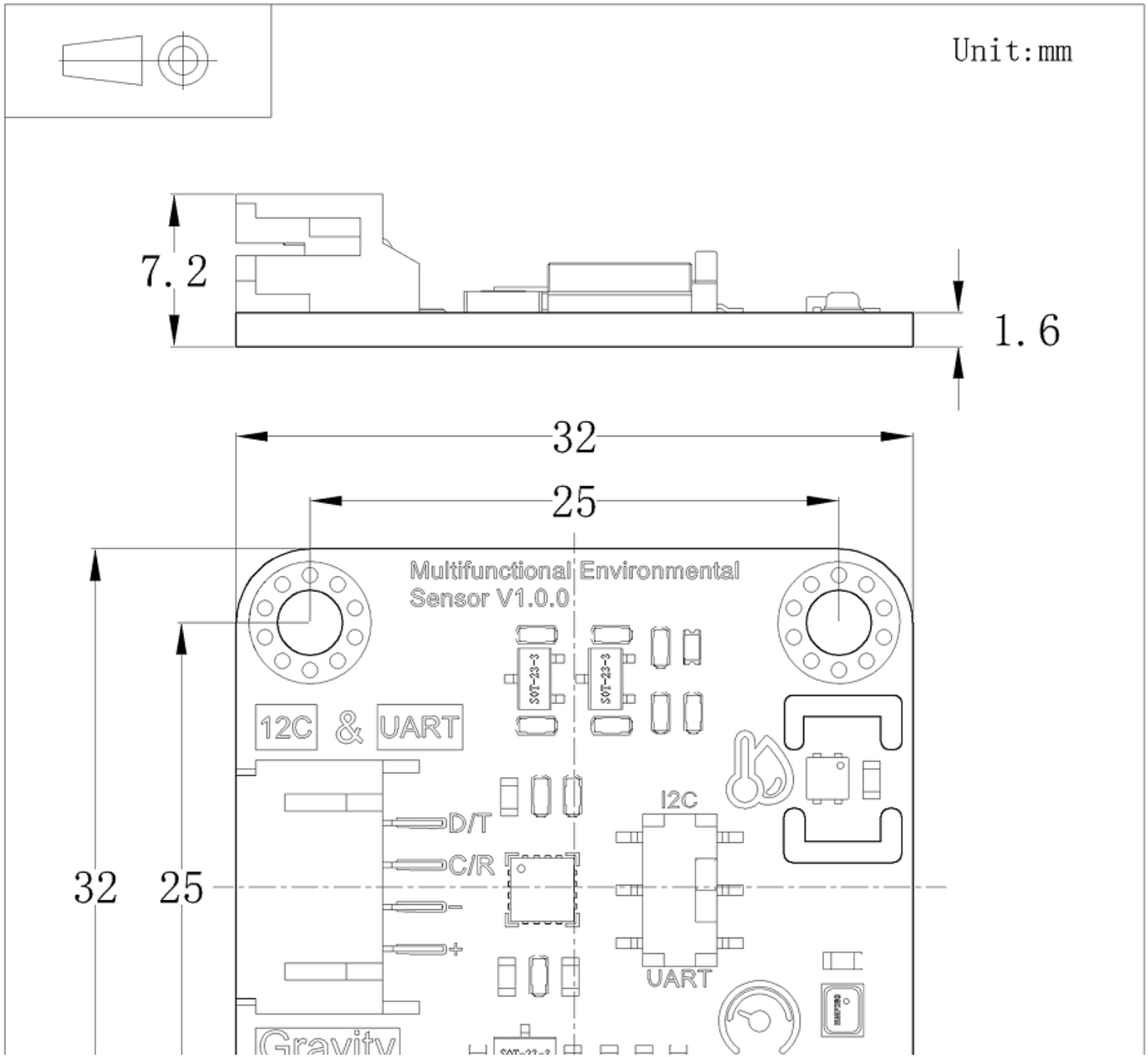
- Sensor chip model: LTR390-UV-01
- Ultraviolet sensitive wavelength: 280-430nm
- Measurement data Range: 13 Bit, 16 Bit, 17 Bit, 18 Bit, 19 Bit, 20 Bit
- Measurement rate: 25ms, 50ms, 100ms, 200ms, 500ms, 1000ms

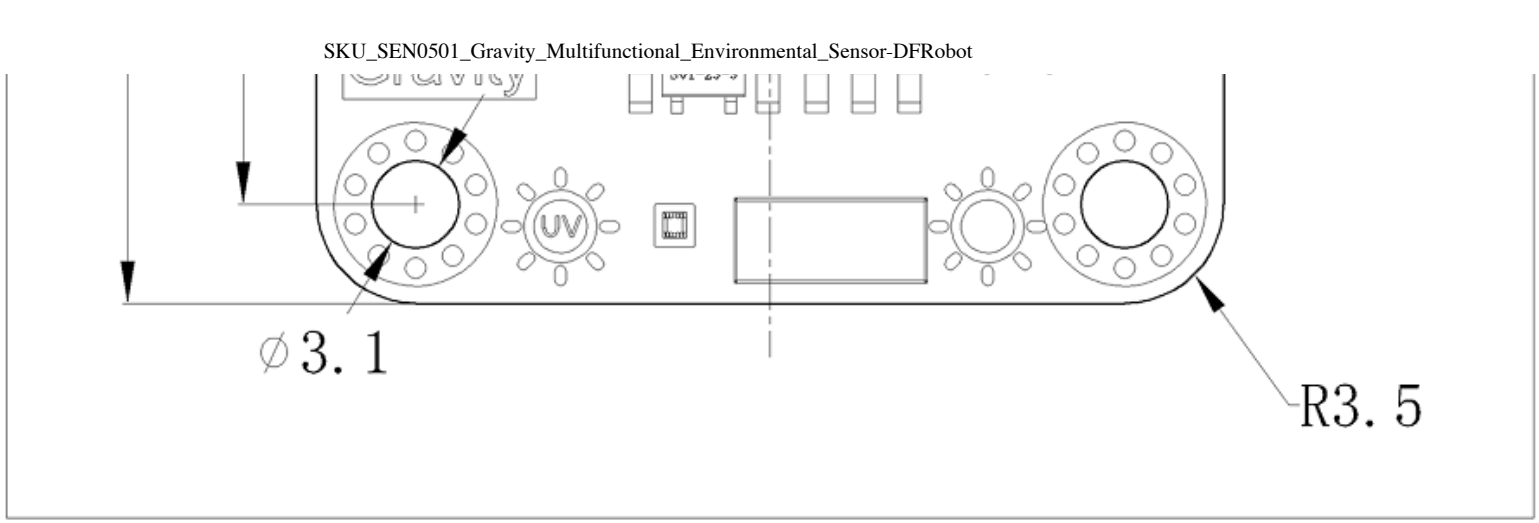
- Adjustable gain: X1, X3, X6, X9, X18

Ambient light sensor

- Sensor chip model: VEML7700
- Ambient light accuracy: 0.0036 lx/ct
- Ambient light range: 0-120 klx

Dimension





Overview

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

Tutorial

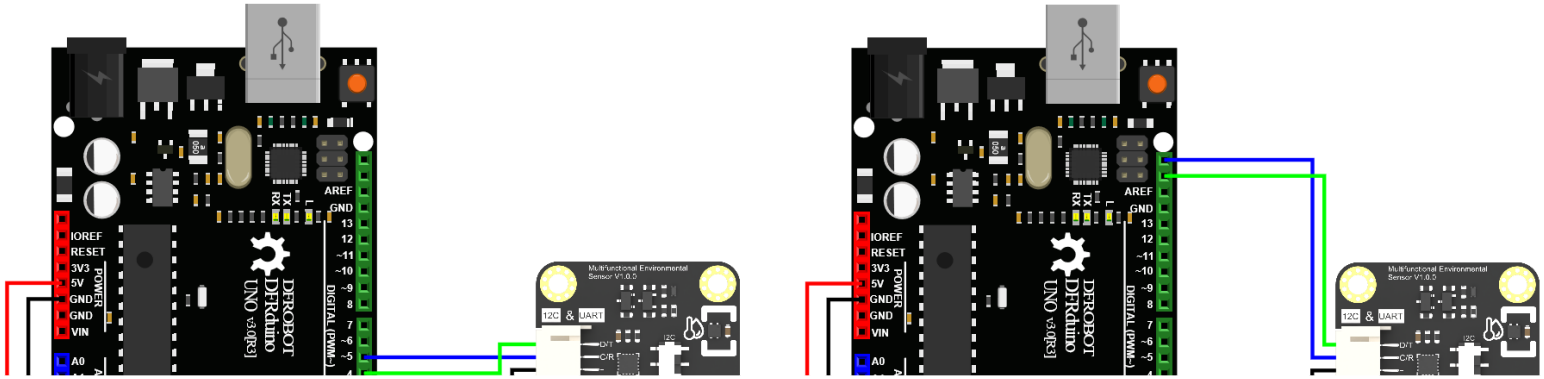
Download the program to DFRuino UNO, open the serial monitor to check various environmental parameters.

Requirements

- **Hardware**
  - DFRduino UNO R3 (https://www.dfrobot.com/product-838.html) (or similar) x 1
  - SEN0501 Multifunctional environmental sensor x1
  - Wires
- **Software**
  - Arduino IDE (https://www.arduino.cc/en/Main/Software)
  - Download and install DFRobot RTU library (https://github.com/DFRobot/DFRobot_RTU)
  - Download and install the **DFRobot_EnvironmentalSensor Library** (https://github.com/cdjq/DFRobot_EnvironmentalSensor) (About how to install the library? (https://www.arduino.cc/en/Guide/Libraries#.UxU8mdzF9H0))

Read Sensor Data via I2C/UART

Connection for UART and I2C



Switch Communication Mode

About UART/I2C mode switching:

1. The default mode in the code is UART. Dial the switch to UART side to use it.
2. For using I2C, dial the switch to I2C side and replace the 1 at the beginning of the code with 0, as shown below:

```
#define MODESWITCH        /*UART:*/1 /*I2C: 0*/
```

to

```
#define MODESWITCH        /*UART:*/0 /*I2C: 0*/
```

Steps:

Note: Different motherboards correspond to different wiring pins (refer to “more wiring diagrams” above). In the Arduino IDE, they also correspond to different motherboard options and different ports (COM). The other steps are the same as DFRuino UNO.

- Connect the module and DFRuino UNO according to the wiring method above.
- Open the Arduino IDE and upload the following code to DFRuino UNO.
- Open the serial port monitor of Arduino IDE, adjust the baud rate to 115200, and observe the serial port printing result.

Sample Code

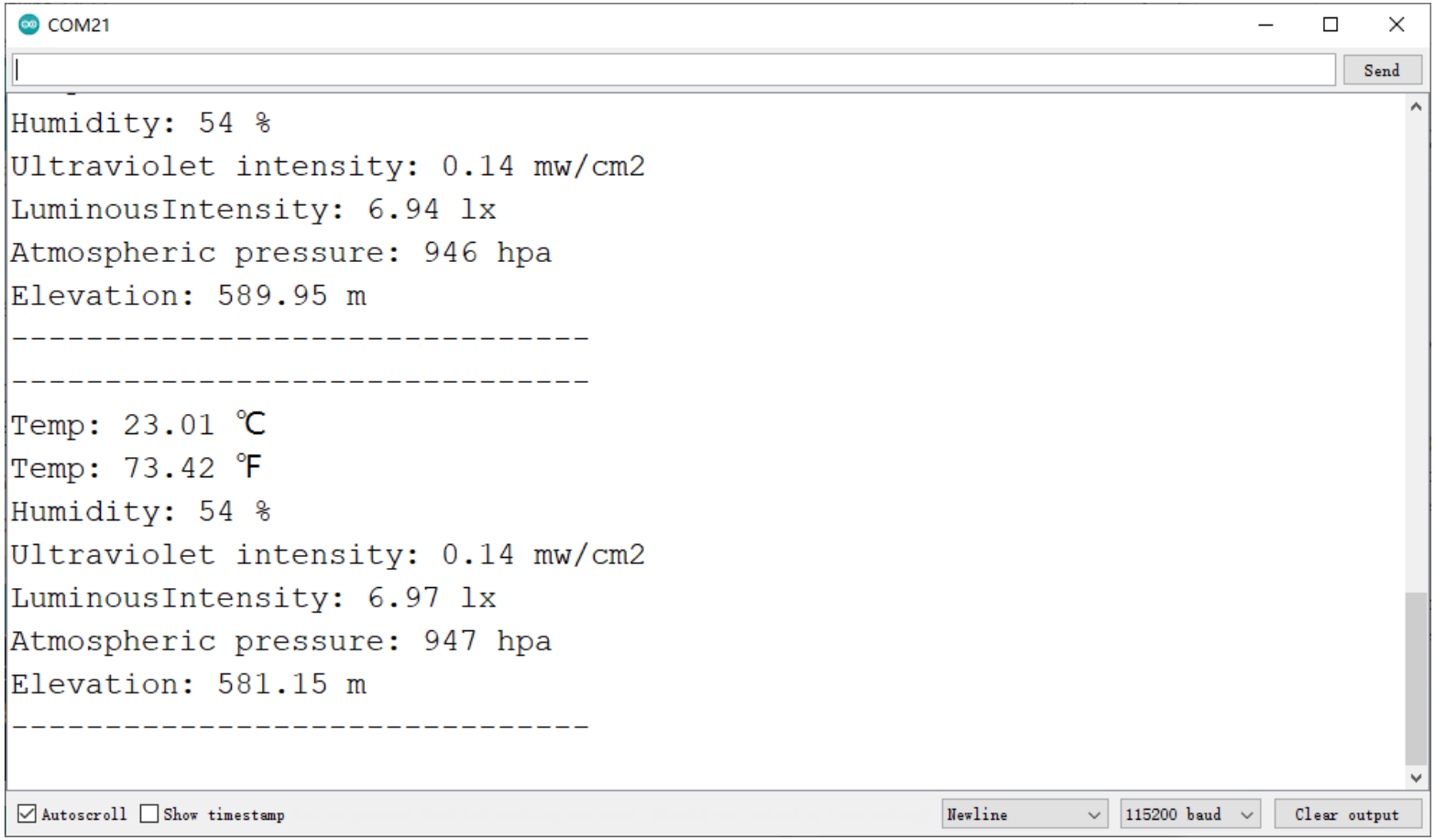
Function: Print all data obtained by the module

```
#include "DFRobot_EnvironmentalSensor.h"
#if defined(ARDUINO_AVR_UNO)||defined(ESP8266)
#include <SoftwareSerial.h>
#endif

#define MODESWITCH        /*UART:*/1 /*I2C: 0*/

#if MODESWITCH
#if defined(ARDUINO_AVR_UNO)||defined(ESP8266)
  SoftwareSerial mySerial(/*rx =*/4, /*tx =*/5);
  DFRobot_EnvironmentalSensor environment(/*addr =*/SEN0500/SEN0501_DEFAULT_DEVICE_ADDRESS, /
#else
  DFRobot_EnvironmentalSensor environment(/*addr =*/SEN0500/SEN0501_DEFAULT_DEVICE_ADDRESS, /
#endif
#else
DFRobot_EnvironmentalSensor environment(/*addr = */SEN0500/SEN0501_DEFAULT_DEVICE_ADDRESS, /*
#endif
void setup()
{
#if MODESWITCH
  //Init MCU communication serial port
#if defined(ARDUINO_AVR_UNO)||defined(ESP8266)
  mySerial.begin(9600);
#elif defined(ESP32)
  Serial1.begin(9600, SERIAL_8N1, /*rx =*/D3, /*tx =*/D2);
#else
  Serial1.begin(9600);
#endif
#endif
  Serial.begin(115200);
  while(environment.begin() != 0){
    Serial.println(" Sensor initialize failed!!");
    delay(1000);
  }
  Serial.println(" Sensor  initialize success!!");
```

```
}

void loop()
{
  //Print the data obtained from sensor
  Serial.println("------------------------------");
  Serial.print("Temp: ");
  Serial.print(environment.getTemperature(TEMP_C));
  Serial.println(" ℃");
  Serial.print("Temp: ");
  Serial.print(environment.getTemperature(TEMP_F));
  Serial.println(" ℉");
  Serial.print("Humidity: ");
  Serial.print(environment.getHumidity());
  Serial.println(" %");
  Serial.print("Ultraviolet intensity: ");
  Serial.print(environment.getUltravioletIntensity());
  Serial.println(" mw/cm2");
  Serial.print("LuminousIntensity: ");
  Serial.print(environment.getLuminousIntensity());
  Serial.println(" lx");
  Serial.print("Atmospheric pressure: ");
  Serial.print(environment.getAtmospherePressure(HPA));
  Serial.println(" hpa");
  Serial.print("Altitude: ");
  Serial.print(environment.getElevation());
  Serial.println(" m");
  Serial.println("------------------------------");
  delay(500);
}
```

Expected Results

The read data will be dipalyed on the serial monitor.



Function Library Name Definition

```
/**
   * @fn begin
   * @brief Init SEN0500/SEN0501 sensor
   * @return Return init status
   * @retval 0  Succeed
   * @retval -1 failed
   */
  int8_t begin(void);

  /**
   * @fn getTemperature
   * @brief Get SEN0500/SEN0501 temperature data
   * @param units Temperature data unit select
   * @n     TEMP_C ℃
   * @n     TEMP_F ℉
   * @return Return the obtained temperature data
   */
  float getTemperature(uint8_t unist);

  /**
   * @fn getHumidity
   * @brief Get SEN0500/SEN0501 humidity data
   * @return Return the obtained humidity data
   */
  float getHumidity(void);

  /**
   * @fn getUltravioletIntensity
   * @brief Get SEN0500/SEN0501 UV intensity index data
   * @return Return the obtained UV intensity index data
   */
  float getUltravioletIntensity(void);
```

```
/**
 * @fn getLuminousIntensity
 * @brief Get SEN0500/SEN0501 luminous intensity data
 * @return Return the obtained luminous intensity data
 */
float getLuminousIntensity(void);

/**
 * @fn getAtmospherePressure
 * @brief Get SEN0500/SEN0501 atmosphere pressure data
 * @param units Atmosphere pressure data unit select
 * @n            HPA: Hectopascal
 * @n            KPA: Kilopascal
 * @return Return the obtained atmosphere pressure data
 */
uint16_t getAtmospherePressure(uint8_t units);

/**
 * @fn getElevation
 * @brief Get SEN0500/SEN0501 altitude data
 * @return Return the obtained altitude data
 */
float getElevation(void);
```

FAQ

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

More Documents

- SEN0501_V1.0_SCH_Schematics (https://dfimg.dfrobot.com/nobody/wiki/627c78f8e1fb081b16840367dd803555.PDF)

- SEN0501_V1.0_2D_PDF.pdf (https://dfimg.dfrobot.com/nobody/wiki/3d0739003e69fdaaf5c7046421e7ade8.pdf)
- SEN0501_V1.0_2D_DXF.rar (https://dfimg.dfrobot.com/nobody/wiki/c85979fabc698981f3705f7dab79aba2.rar)
- SEN0501_3D_V1.0_STP.rar (https://dfimg.dfrobot.com/nobody/wiki/928c3290f5d362327bdfa7ef156e2333.rar)
- SEN0501_V2.0_SCH_Schematics.PDF (https://dfimg.dfrobot.com/nobody/wiki/47b088c2f7034de0eb9feda7240529fe.PDF)
- SEN0501_V2.0_2D_PDF.pdf (https://dfimg.dfrobot.com/nobody/wiki/bc6ad86c7e93d7baa4ac97ba8e17e225.pdf)
- SEN0501_V2.0_2D_DXF.rar (https://dfimg.dfrobot.com/nobody/wiki/9cc8cacce48c00d4e60530a0edb1f6a9.rar)
- SEN0501_V2.0_3D_STP.rar (https://dfimg.dfrobot.com/nobody/wiki/9bb3e6e497098446b65c550e5d58758d.rar)

 Get **Gravity Multifunctional Environmental Sensor** (https://www.dfrobot.com/product-2528.html) from DFRobot Store or **DFRobot Distributor**. (https://www.dfrobot.com/distributor)

Turn to the Top