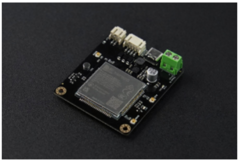
search
DFRobot Products (DFR) Index
Sensor (SEN) Index
Telecom Module (TEL) Index
Motor Driver (DRI) Index
Fitting (FIT) Index
Robot (ROB) Index
micro:bit (MBT) Index
Kit (KIT) Index
Toy (TOY) Index
Servos (SER) Index
What is XX series
How to select series
How it works series
PinPong Library

Introduction
Features
Application
Specifications
Functional Diagram
Diagram
UART Communication Description
Arduino Tutorial
API Functions
More Material Download
Frequently Asked Questions (FAQ)

SKU:SEN0611

Introduction



Gravity: The Color Temperature Illuminance Sensor is a high-performance light source sensing sensor, specifically designed for precise measurement and control of the lighting environment. It directly outputs two key optical physical quantities, color temperature and illuminance intensity, making your understanding of ambient light more accurate.

This sensor supports two communication methods, I2C and UART. After factory calibration, it can quickly and accurately measure various light source data in the environment. With these precise data, you can implement various applications such as home light color temperature synchronization and color temperature illuminance art interaction projects, making your lighting environment more in line with your needs.

Color Temperature

Color temperature is a physical quantity used to describe the color characteristics of a light source, in Kelvin (K). Its definition comes from the theory of blackbody radiation in physics, that is, when a blackbody is heated to a certain temperature, it will emit light of a specific color, and this temperature is defined as the color temperature of this color.

In our daily life, the concept of color temperature is everywhere. For example, what we often say about warm light and cold light actually refers to the different color temperatures of light sources. Warm light has a lower color temperature, usually between 2000K-3000K, and the light emitted is red or warm; while cold light has a higher color temperature, usually above 5000K, and the light emitted is blue or cold. The light temperature of different times of the day and weather conditions also varies. This color temperature illuminance sensor can convert these environmental lights into standard physical quantities (Kelvin, K).

Illuminance Intensity

Illuminance intensity, also known as illuminance, is a physical quantity that describes the brightness intensity of a light source at a specific distance or area, in lux (Lux). It reflects the light flux irradiated on the unit area, that is, what we usually call the "brightness level". For example, in a sunny outdoor environment, the illuminance intensity may be as high as 10,000 Lux, while in a dim indoor environment, the illuminance intensity may only be a few dozen or a few hundred Lux. In lighting design, photography, plant photosynthesis research and other fields, illuminance intensity is an important parameter.

Features

- Outputs various light source data (color temperature, illuminance, xy color coordinates)

- Adopts I2C and UART as two modes of communication, Gravity interface
- Compact size, only 26mm×24mm
- Compatible with 3.3~5V main controller

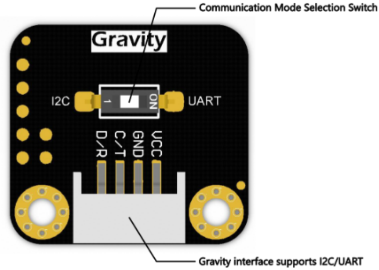
Application

- Indoor lighting systems
- Exhibition hall lighting systems
- Light quality detection equipment
- Supermarket lighting systems
- Screen display adjustment

Specifications

- Operating voltage: 3.3-5V DC
- Operating current: 2.8mA
- Communication mode: I2C, UART
- Illuminance measurement range: 10-10000lux
- Illuminance measurement error: (±5) + (±10%) lux
- Color temperature measurement range: 2700-6500K
- Color temperature measurement error: ±10% K
- Light source angle measurement: 90°
- Operating temperature: -10~60°C
- Size: 26mm×24mm
- Partial environmental colortemperature illuminance data reference:
    - Home living area: 2700K-5000K, 100-500 Lux
    - Office workspace: 3000K-5000K, 300-500 Lux
    - Store lighting: 3500K-4100K, 200-1000 Lux
    - Gallery or museum: 3500K-4100K, 100-500 Lux
    - School or library: 4100K-5000K, 300-500 Lux

Functional Diagram



Communication Mode Selection Switch

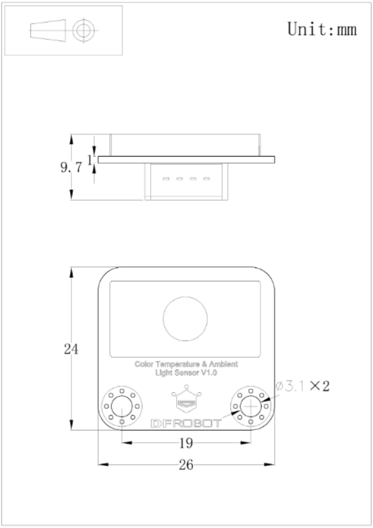
Gravity interface supports I2C/UART

Num	Label	Description
1	D/T	I2C data line SDA or UART TXD
2	C/R	I2C clock line SCL or UART RXD
3	GND	Power negative
4	VCC	Power positive (3.3~5V)

5	Communication Mode Switch	Select I2C/UART communication mode

Note: Power off when switching communication mode

Diagram



UART Communication Description

Factory Parameters

- Device address: 0x0032
- Baud rate: 9600bps

Command Table:

Device Address	Input Register Address	Baud Rate Address	Illuminance Reading Address	Color Temperature Data Reading Address
0x0032	0x0004	0x0003	0x0006	0x0007

The format for sending commands is: Device address + function code + starting address of the register + number of registers + CRC check.

Reading UART Baud Rate

The master reads the baud rate of the module through the UART,

the master sends: 32 04 00 03 00 01 C4 09 //Read the baud rate as: 32 04 02 00 03 FD 35

```
Device Address-------------------------------32
Function Code--------------------------------04//Read Input Reg
Starting Address of Register----------------00 03//Read Baud R
Number of Registers-------------------------00 01
CRC Check-----------------------------------C4 09
```

The baud rate data returned by the module is: 32 04 02 00 03 FD 35.

```
Device Address----------------------32
Function Code-----------------------04//Read Input Register
Number of Bytes---------------------02//Number of Returned Da
Data Read---------------------------00 03//9600bps
CRC Check---------------------------FD 35
```

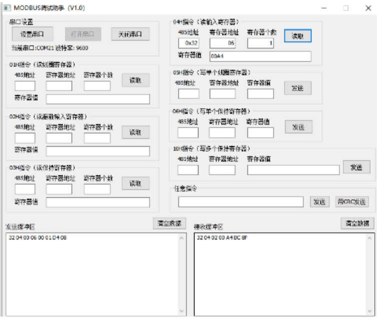
Baud Rate Table:

2400bps	4800bps	9600bps	14400bps	19200bps	3
0x0001	0x0002	0x0003	0x0004	0x0005	0

**Note:* *The CRC check code is calculated through a method called Cyclic Redundancy Check. This algorithm will perform a series of bit operations on the data, including XOR, shift, etc., to finally obtain a check code. The check code can be used to check if there are errors in the data during transmission.

Reading Illuminance

The master reads the illuminance measured by the module through the UART.



当前串口:COM21 波特率: 9600

04H指令（读输入寄存器）
485地址: 0x32
寄存器地址: 06
寄存器个数: 1
寄存器值: 00A4

发送缓冲区: 32 04 00 06 00 01 D4 08

接收缓冲区: 32 04 02 00 A4 BC 8F

The host sends 32 04 00 06 00 01 D4 08 //Illuminance read is 32 04 02 00 A4 BC 8F

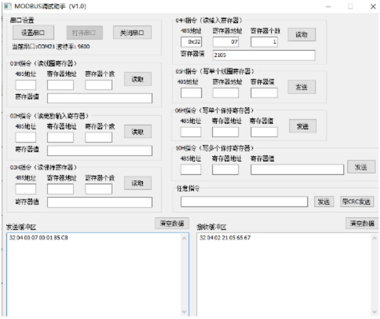
```
Device Address-------------------------32
Function Code--------------------------04//Read Input Registe
Starting Address of Register-----------00 06//Read Illuminanc
Number of Registers--------------------00 01
CRC Check------------------------------D4 08
```

The module reads the illuminance as: 32 04 02 00 A4 BC 8F

```
Device Address----------------------32
Function Code-----------------------04//Read Input Register
Number of Bytes---------------------02//Number of Returned Da
Data Read---------------------------00 A4//Converted to Decim
CRC Check---------------------------BC 8F
```

Reading Color Temperature Data

The master reads the color temperature data measured by the module through the UART.



当前串口:COM21 波特率: 9600

04H指令（读输入寄存器） 485地址 0x32 寄存器地址 07 寄存器个数 1 寄存器值 2105

发送缓冲区

```
32 04 00 07 00 01 85 C8
```

接收缓冲区

```
32 04 02 21 05 65 67
```

The host sends 32 04 00 07 00 01 85 C8 //Color temperature data read is 32 04 02 21 05 65 67

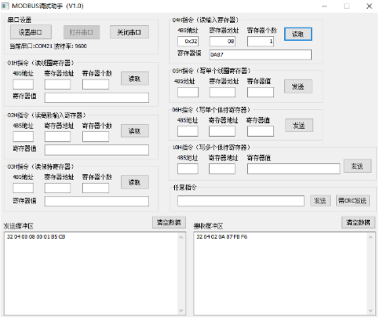
```
Device Address------------------------32
Function Code-------------------------04//Read Input Register
Starting Address of Register----------00 07//Read Color Tempe
Number of Registers-------------------00 01
CRC Check-----------------------------85 C8
```

The module reads the color temperature data as: 32 04 02 21 05 65 67

```
Device Address---------------------32
Function Code----------------------04//Read Input Register
Number of Bytes--------------------02//Number of Returned Dat
Data Read--------------------------21 05//Converted to Decima
CRC Check--------------------------65 67
```

Reading Color Coordinate x

The master reads the color coordinate x measured by the module through the UART.



当前串口:COM21 波特率: 9600

04H指令（读输入寄存器） 485地址 0x32 寄存器地址 08 寄存器个数 1 寄存器值 0A87

发送缓冲区

```
32 04 00 08 00 01 B5 CB
```

接收缓冲区

```
32 04 02 0A 87 FB F6
```

The host sends 32 04 00 08 00 01 B5 CB //Color coordinate x read is 32 04 02 0A 87 FB F6

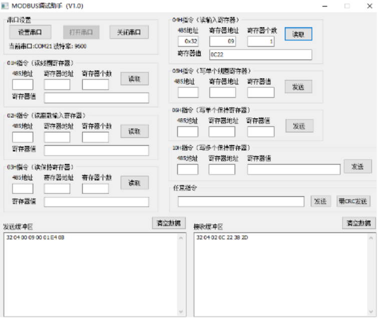
```
Device Address-----------------------32
Function Code------------------------04//Read Input Register
Starting Address of Register----------00 08//Read Color Coordi
Number of Registers------------------00 01
CRC Check----------------------------B5 CB
```

The module reads the color coordinate x as: 32 04 02 0A 87 FB F6

```
Device Address---------------------32
Function Code----------------------04//Read Input Register
Number of Bytes--------------------02//Number of Returned Dat
Data Read--------------------------0A 87//0x0a87/1000
CRC Check--------------------------FB F6
```

Reading Color Coordinate y

The master reads the color coordinate y measured by the module through the UART.



当前串口:COM21 波特率: 9600

04H指令（读输入寄存器）: 485地址 0x32, 寄存器地址 09, 寄存器个数 1, 寄存器值 0C22

发送缓冲区

```
32 04 00 09 00 01 E4 0B
```

接收缓冲区

```
32 04 02 0C 22 38 2D
```

The host sends 32 04 00 09 00 01 E4 0B //Color coordinate y read is 32 04 02 0C 22 38 2D

```
Device Address----------------------32
Function Code-----------------------04//Read Input Register
Starting Address of Register---------00 09//Read Color Coordin
Number of Registers-----------------00 01
CRC Check---------------------------E4 0B
```

The module reads the color coordinate y as: 32 04 02 0C 22 38 2D

```
Device Address--------------------32
Function Code---------------------04//Read Input Register
Number of Bytes-------------------02//Number of Returned Data
Data Read-------------------------0C 22//0x0c22/1000
CRC Check-------------------------38 2D
```

Arduino Tutorial

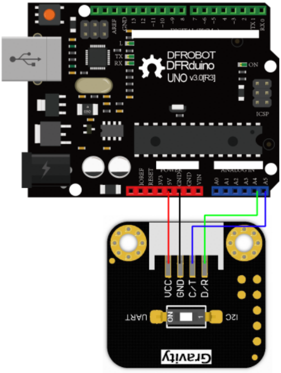
Software and Hardware Preparation

- Hardware
    - DFRduino UNO Controller x1

  - SEN0611 Gravity: Color Temperature Illuminance Sensor x1
- Software
  - Arduino IDE Click to Download Arduino IDE
  - Download and install DFRobot_ColorTemperature library. How to Install Library?

Use I2C to Read Data

Wiring Diagram



Example Code

- Connect the module to the UNO controller according to the above wiring diagram, of course, you can also use the expansion board, which can be more convenient and faster to complete the project prototype.
- Switch the selector switch on the sensor to the I2C side.
- Download and install DFRobot_ColorTemperature library. How to Install Library?
- Open Arduino IDE, upload the following code to the UNO controller.
- Open the serial monitor of Arduino IDE, adjust the baud rate to 115200, and observe the serial print result.

Note: Power off when switching the switch

```
#include "DFRobot_ColorTemperature.h"
DFRobot_ColorTemperature CT(&Wire);

void setup()
{
  Serial.begin(115200);
  while(CT.begin() != 0){
    Serial.println(" Sensor initialize failed!!");
    delay(1000);
  }
  Serial.println(" Sensor  initialize success!!");
}

void loop()
{
  Serial.println("--------------------------------");
  Serial.print("LUX: ");
  Serial.println(CT.readLUX());
  Serial.print("CCT: ");
```

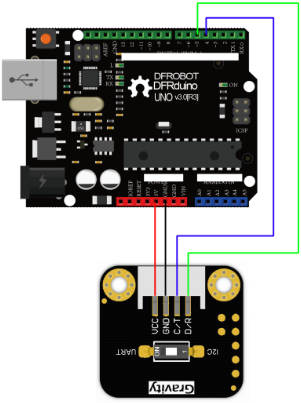
```
  Serial.println(CT.readCCT());
  Serial.print("X: ");
  Serial.println(CT.readX());
  Serial.print("Y: ");
  Serial.println(CT.readY());
  Serial.println("------------------------------");
  delay(500);
}
```

Result

Open the serial monitor to obtain the final color temperature, illuminance, and X, Y color coordinates data.

Use UART to Read Data

Wiring Diagram



Example Code

- Connect the module to the UNO controller according to the above wiring diagram, of course, you can also use the expansion board, which can be more convenient and faster to complete the project prototype.
- Switch the selector switch on the sensor to the UART side.
- Download and install DFRobot_ColorTemperature library. How to Install Library?
- Open Arduino IDE, upload the following code to the UNO controller.
- Open the serial monitor of Arduino IDE, adjust the baud rate to 115200, and observe the serial print result.

Note: Power off when switching the switch

```
#include "DFRobot_ColorTemperature.h"
#include <SoftwareSerial.h>

SoftwareSerial mySerial(/*rx =*/4, /*tx =*/5);
DFRobot_ColorTemperature CT(/*s =*/&mySerial);

void setup()
{
  mySerial.begin(9600);
```

```
  Serial.begin(115200);
  while(CT.begin() != 0){
    Serial.println(" Sensor initialize failed!!");
    delay(1000);
  }
  Serial.println(" Sensor  initialize success!!");
}

void loop()
{
  Serial.println("-------------------------------");
  Serial.print("LUX: ");
  Serial.println(CT.readLUX());
  Serial.print("CCT: ");
  Serial.println(CT.readCCT());
  Serial.print("X: ");
  Serial.println(CT.readX());
  Serial.print("Y: ");
  Serial.println(CT.readY());
  Serial.println("-------------------------------");
  delay(500);
}
```

Result

Open the serial monitor to obtain the final color temperature, illuminance, and X, Y color coordinates data.

API Functions

```
class DFRobot_ColorTemperature: public DFRobot_RTU{
public:
    #define COLORTEMPERATURE_ADDR     0x32 ///<Device Address
    #define COLORTEMPERATURE_ADDR_REG 0X02 ///< Device Address
    #define COLORTEMPERATURE_LUX_REG  0X06 ///< Illuminance Re
    #define COLORTEMPERATURE_CCT_REG  0X07 ///< Color Temperat
    #define COLORTEMPERATURE_X_REG    0X08 ///< Color Coordina
    #define COLORTEMPERATURE_Y_REG    0X09 ///< Color Coordina



  /**
   * @fn DFRobot_ColorTemperature
   * @brief DFRobot_ColorTemperature constructor
   * @param pWire I2C pointer to the TowWire stream, which req
   * @param addr  I2C communication address of SEN0611 device
   */
  DFRobot_ColorTemperature(TwoWire *pWire);

  /**
   * @fn DFRobot_ColorTemperature
   * @brief DFRobot_ColorTemperature constructor
   * @param addr The device address of the communication betwe
   * @param s  The serial port pointer to the Stream, which re
   * @n SEN0611 serial port config: 9600 baud rate, 8-bit data
   */
  DFRobot_ColorTemperature(Stream *s);
  ~DFRobot_ColorTemperature(){};
  /**
   * @fn begin
   * @brief Initialize SEN0593
   * @return Initialization status
   * @retval 0  Success
   * @retval -1 Failure
   */
  int8_t begin(void);

  /**
   * @fn readLUX
   * @brief Get sensor illuminance
   * @return Illuminance obtained
  */
  uint16_t readLUX(void);
```

```
/**
 * @fn readCCT
 * @brief Get color temperature data
 * @return Color temperature obtained
*/
uint16_t readCCT(void);

/**
 * @fn readX
 * @brief Get color coordinate X
 * @return Color coordinate X obtained
*/
float readX(void);

/**
 * @fn readY
 * @brief Get color coordinate Y
 * @return Color coordinate Y obtained
*/
float readY(void);
```

More Material Download

- SEN0611_Schematic
- SEN0611_2D_CAD.pdf
- SEN0611_2D_DXF.rar
- SEN0611_3D_STP.rar Error test record sheet

Frequently Asked Questions (FAQ)

No customer has any questions about this product yet, feel free to contact us via qq or forum!

For more questions and interesting applications, you can **visit the forum** for reference or posting.

Downloaded from Arrow.com.