SKU:SEN0562 (https://www.dfrobot.com/product-2664.html)

(https://www.dfrobot.com/product-2664.html)

Introduction

This waterproof digital ambient light sensor is designed to deliver accurate measurements of light intensity within the range of 1-65535lx. It incorporates a 50Hz/60Hz light noise reject function that ensures stable and reliable performance.
With an easy-to-use Gravity interface and I2C communication, the ambient light sensor is incredibly user-friendly for using with main-controllers like Arduino UNO. Plus, the sensor features an IP68 waterproof rating, which means it can be fully immersed in water without suffering any damage. This makes it perfect for outdoor projects that require reliable light measurement in harsh environments, such as vegetable greenhouses, smart car lights, and weather stations.

Brightness Reference:

- Nighttime: 0.001-0.02 lx
- Moonlit night: 0.02-0.3 lx
- Cloudy indoor: 5-50 lx
- Cloudy outdoor: 50-500 lx
- Sunny indoor: 100-1000 lx
- Summer noon sunlight: about 10*6 lx
- Illuminance for reading books: 50-60 lx
- Standard illuminance for home video: 1400 lx

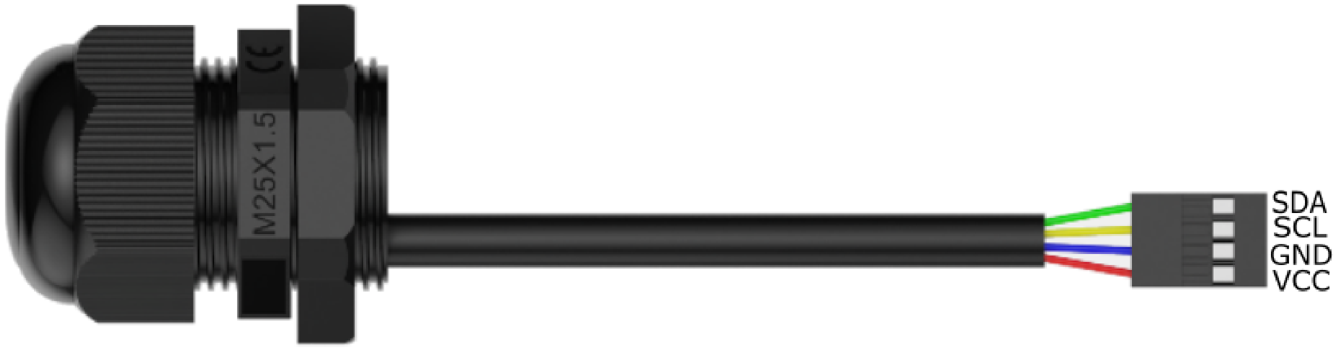
Specification

- Supply Voltage: 5V
- Operating Current: 1μA

- Detection Range: 1 - 65535lx
- Accuracy: 1.2lx
- Communication Mode: I2C

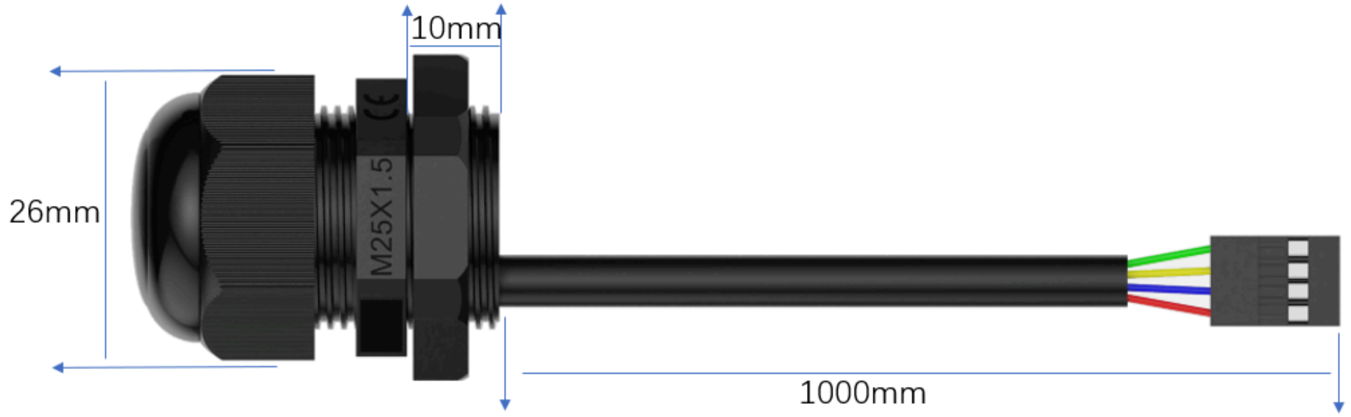
- Operating Temperature: -40 to 85°C/-40 to 185℉
- Waterproof Rating: IP68
- Thread Length: 10mm
- Cutout Size: 26mm
- Wrench Size: 31mm
- Cable Diameter: 3mm
- Wire Length: 1m

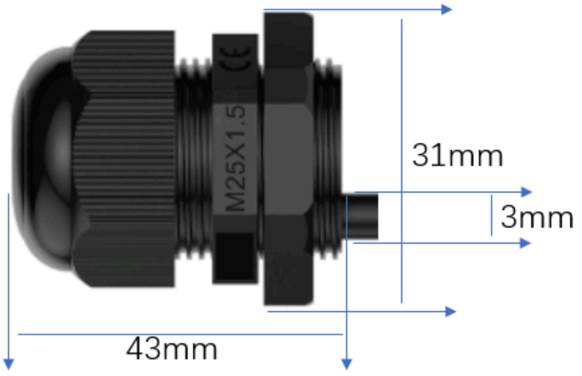
Board Overview



Color	Label	Description
Green	SDA	I2C Data Input
Yellow	SCL	I2C Clock Input
Blue	GND	Power -
Red	VCC	Power +

Dimension



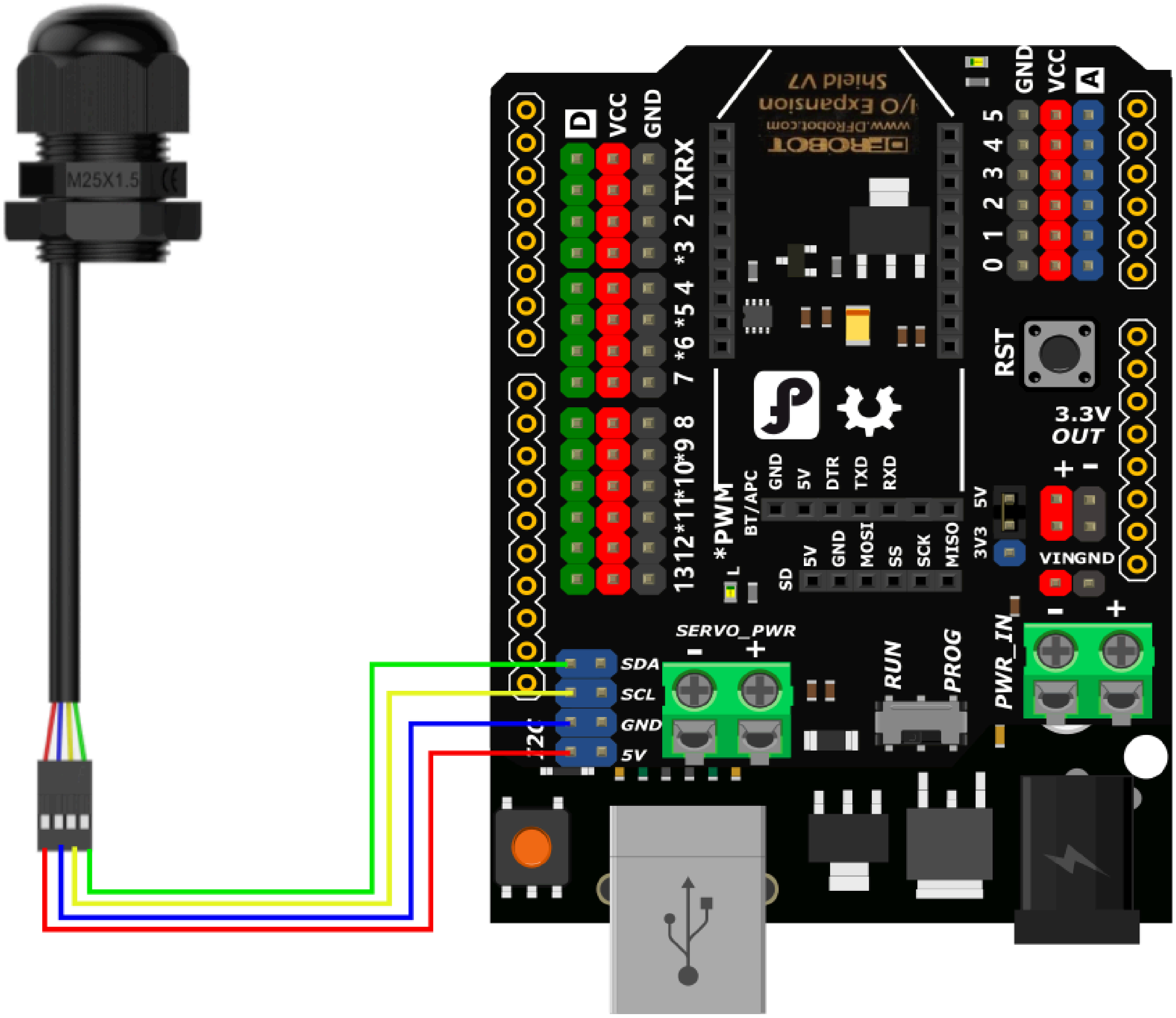


Tutorial

Requirements

- **Hardware**
  - DFRduino UNO R3 (https://www.dfrobot.com/product-838.html) (or similar) x 1
  - Gravity: Waterproof Ambient Light Sensor(1-65535lx) x1
  - M-M/F-M/F-F Jumper wires
- **Software**
  - Arduino IDE (https://www.arduino.cc/en/Main/Software)

Connection Diagram



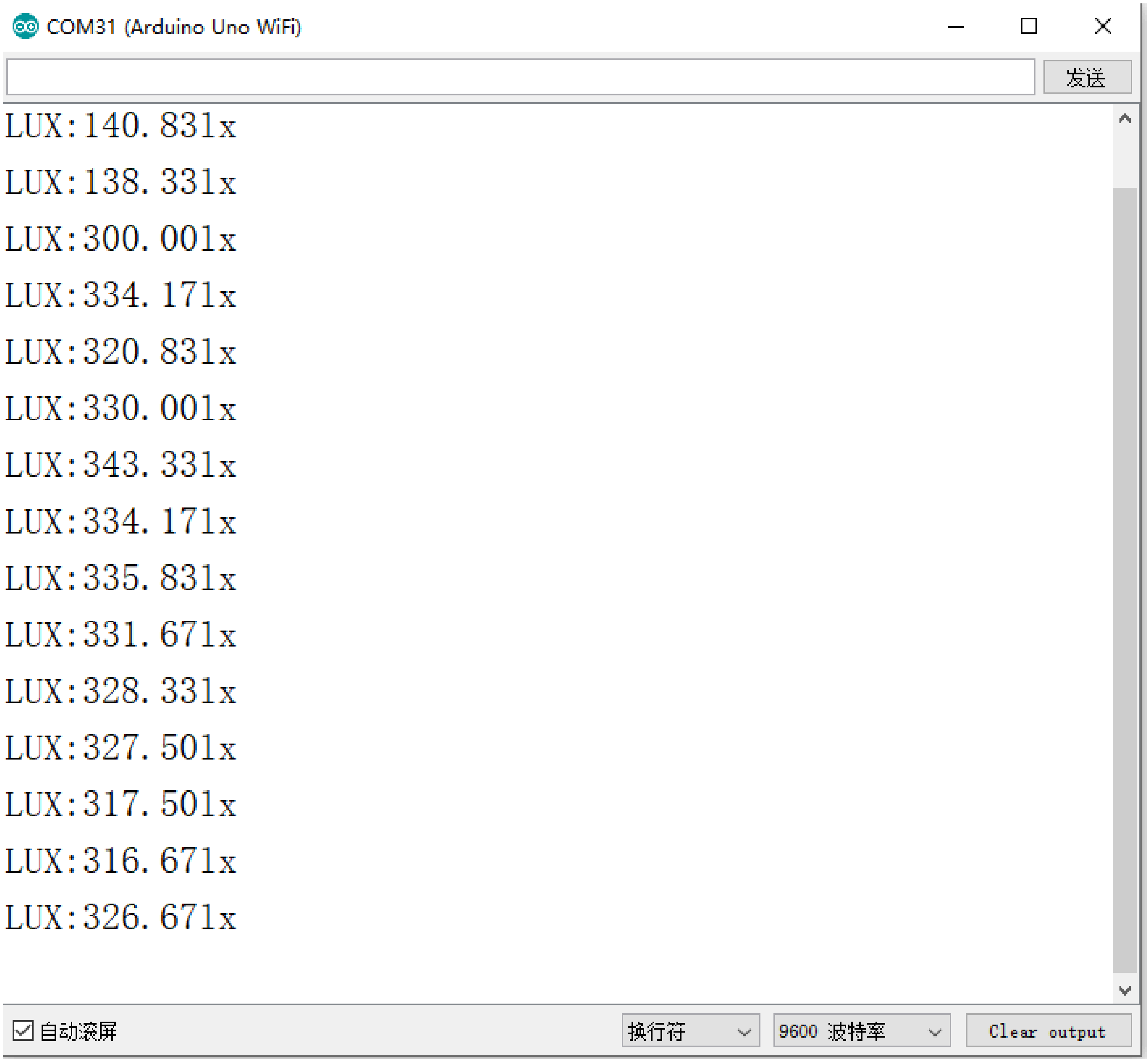
Sample Code

```
#include "Wire.h"
#define address 0x23                 //I2C Address 0x23
void setup()
{
  Serial.begin(9600);
  Wire.begin();
}
uint8_t buf[4] = {0};
uint16_t data, data1;
float Lux;
void loop()
{
  readReg(0x10, buf, 2);              //Register Address 0x10
  data = buf[0] << 8 | buf[1];
  Lux = (((float)data )/1.2);
  Serial.print("LUX:");
  Serial.print(Lux);
  Serial.print("lx");
  Serial.print("\n");
  delay(500);
}
uint8_t readReg(uint8_t reg, const void* pBuf, size_t size)
{
  if (pBuf == NULL) {
    Serial.println("pBuf ERROR!! : null pointer");
  }
  uint8_t * _pBuf = (uint8_t *)pBuf;
  Wire.beginTransmission(address);
  Wire.write(®, 1);
  if ( Wire.endTransmission() != 0) {
    return 0;
  }
  delay(20);
  Wire.requestFrom(address, (uint8_t) size);
  for (uint16_t i = 0; i < size; i++) {
    _pBuf[i] = Wire.read();
  }
  return size;
}
```

Expected Results

Expected Results

The light value collected by the sensor can be viewed on serial monitor.



```
LUX:140.831x
LUX:138.331x
LUX:300.001x
LUX:334.171x
LUX:320.831x
LUX:330.001x
LUX:343.331x
LUX:334.171x
LUX:335.831x
LUX:331.671x
LUX:328.331x
LUX:327.501x
LUX:317.501x
LUX:316.671x
LUX:326.671x
```

FAQ

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

More Documents