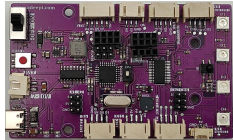




Lesson 8 How to use the Light Tracking Module

In this lesson, we will learn how to use the Light Tracking Module.

8.1 Components used in this course

Components	Quantity	Picture
Adeept Robot Control Board	1	
Type-C USB Cable	1	
Light Tracking Module	1	

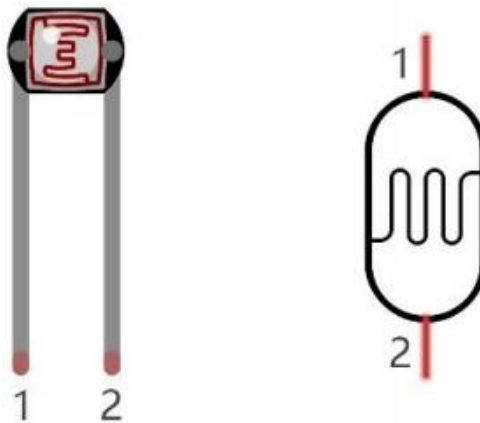
8.2 The introduction of the Light Tracking Module

Light Tracking Module is composed of two photoresistors. The photoresistor is very sensitive to the amount of light present. We can use this feature to make a light tracing car. The car is controlled to turn toward the light source by reading the ADC values of the two photoresistors at the head of the car.

ADC is an electronic integrated circuit used to convert analog signals such as voltages to digital or binary form consisting of 1s and 0s. The range of our ADC on Arduino Uno is 10 bits, that means the resolution is $2^{10}=1024$, and it represents a range (at 5V) will be divided equally to

1024 parts. The range of analog values corresponds to ADC values. So the more bits the ADC has, the denser the partition of analog will be and the greater the precision of the resulting conversion.

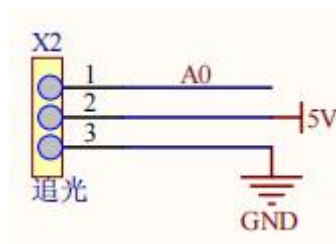
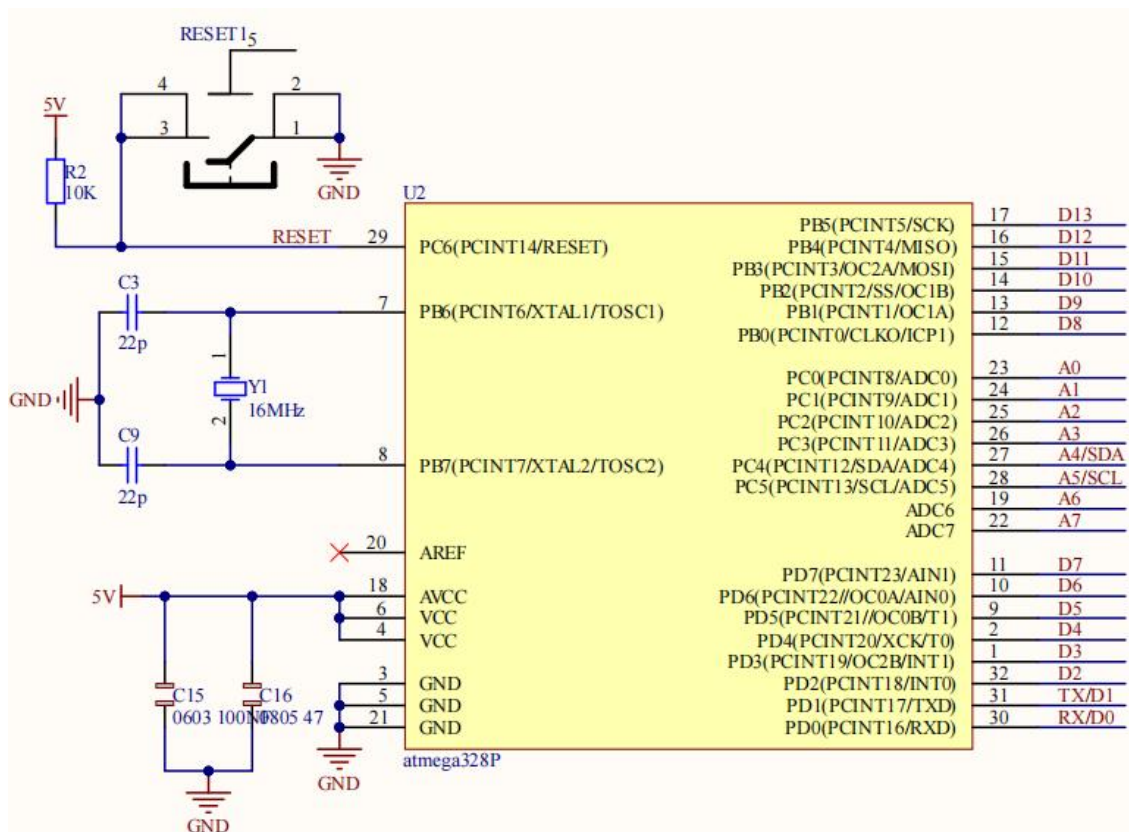
A photoresistor is simply a light sensitive resistor. It is an active component that decreases resistance with respect to receiving luminosity (light) on the component's light sensitive surface. A photoresistor's resistance value will change in proportion to the ambient light detected. With this characteristic, we can use a photoresistor to detect light intensity. The photoresistor and its electronic symbol are as follows.



When a photoresistor's resistance value changes due to a change in light intensity, the voltage between the photoresistor and resistor R1 will also change. Therefore, the intensity of the light can be obtained by measuring this voltage.

When the brightness of the light received by the photoresistor R1 and R2 is the same, the voltage at SIG is 5/2, which is 2.5V. Therefore, when the brightness of the light received by the two photoresistors is different, the voltage at SIG will be greater or less than 2.5V, and the range is 0-5V. The ADC value range corresponding to the SIG voltage range is 0-1024.

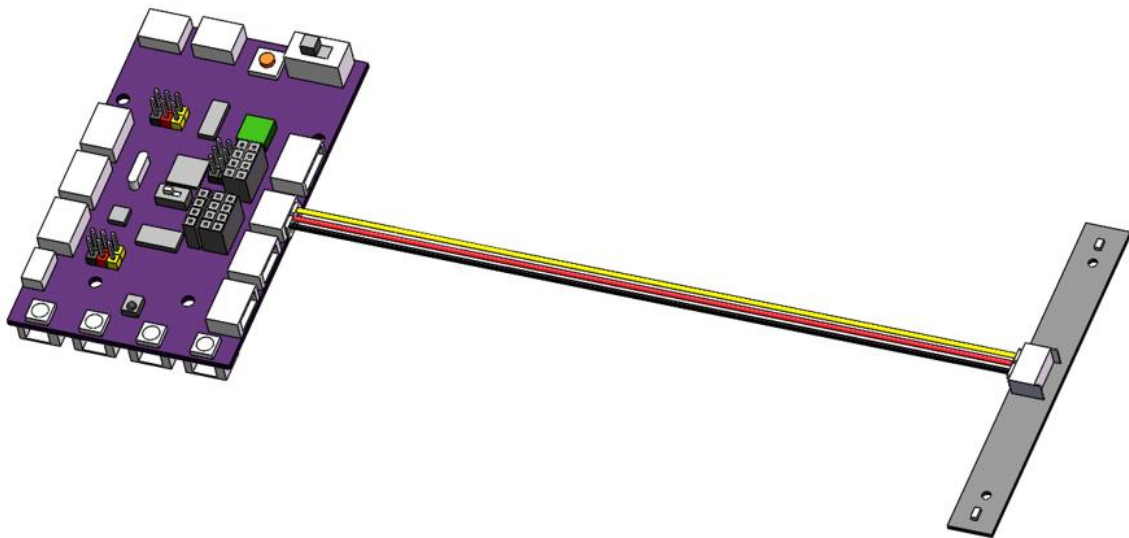
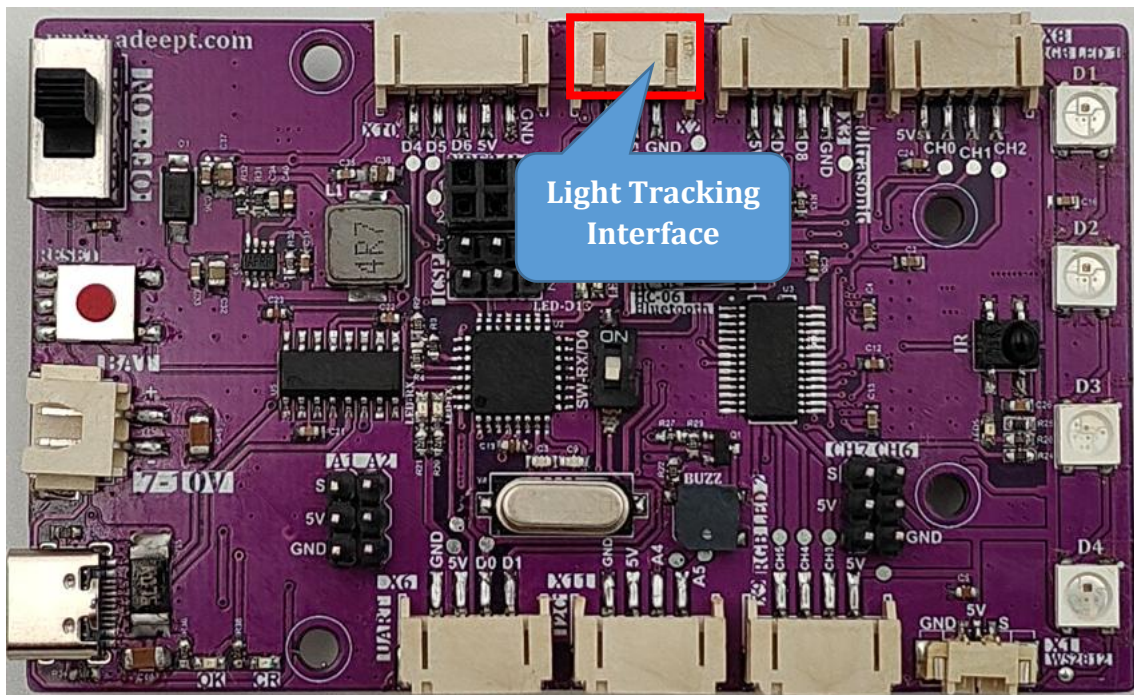
Due to the problem of resistance accuracy, when the module is under the same light intensity, the detected value may deviate, but it does not affect the basic functions of the module.



8.3 Wiring diagram

Light Tracking Module	Arduino(X2)
SIG	A0
VCC	5V
GND	GND

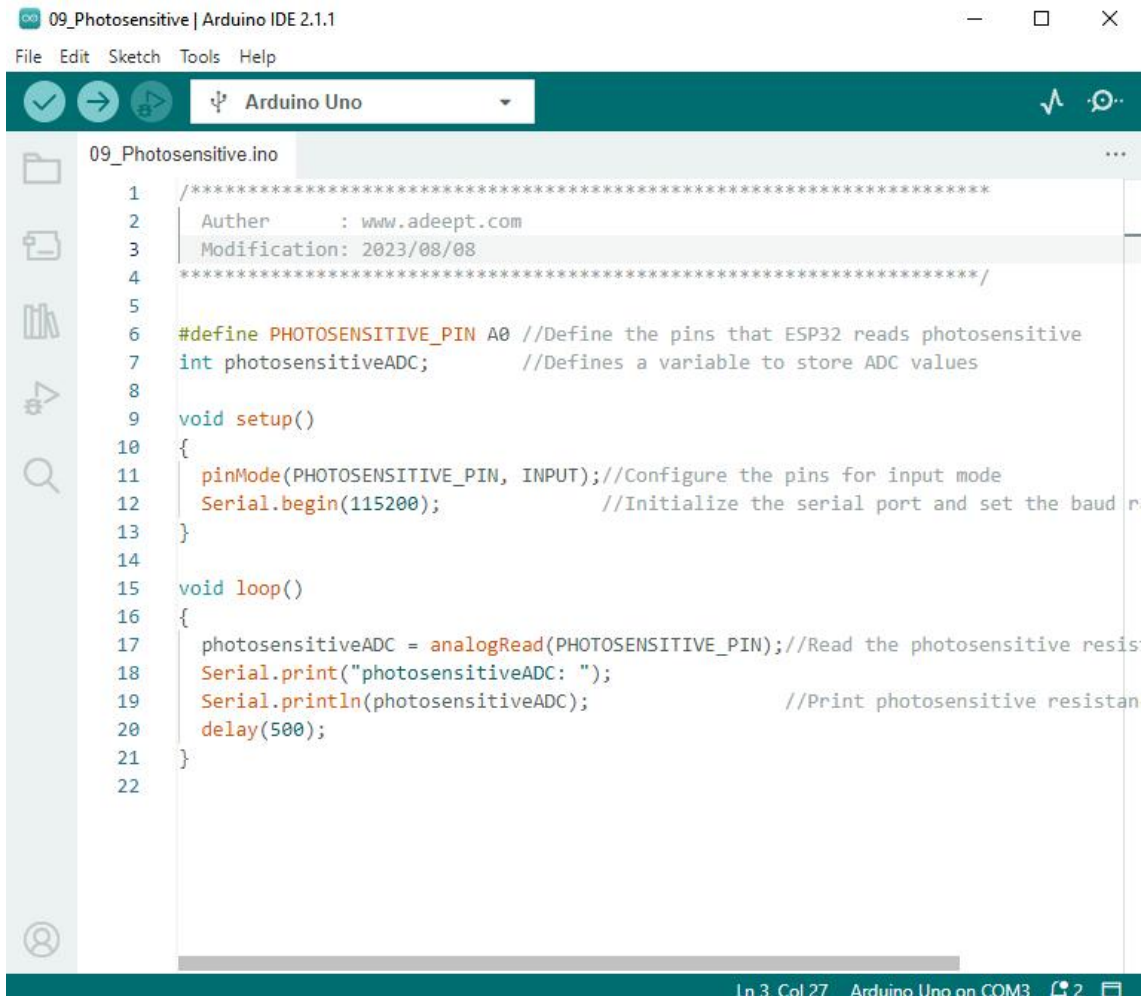
Figure as below:



Light Tracking Module uses 3pin cable, the color is as shown in the picture, and the length is 19CM.

8.4 How to control Light Tracking Module

1. Connect your computer and Adeept Robot Control Board with a USB cable.
2. Open "08_Photosensitive" folder in ["/Code"](#), double-click ["08_Photosensitive.ino"](#).



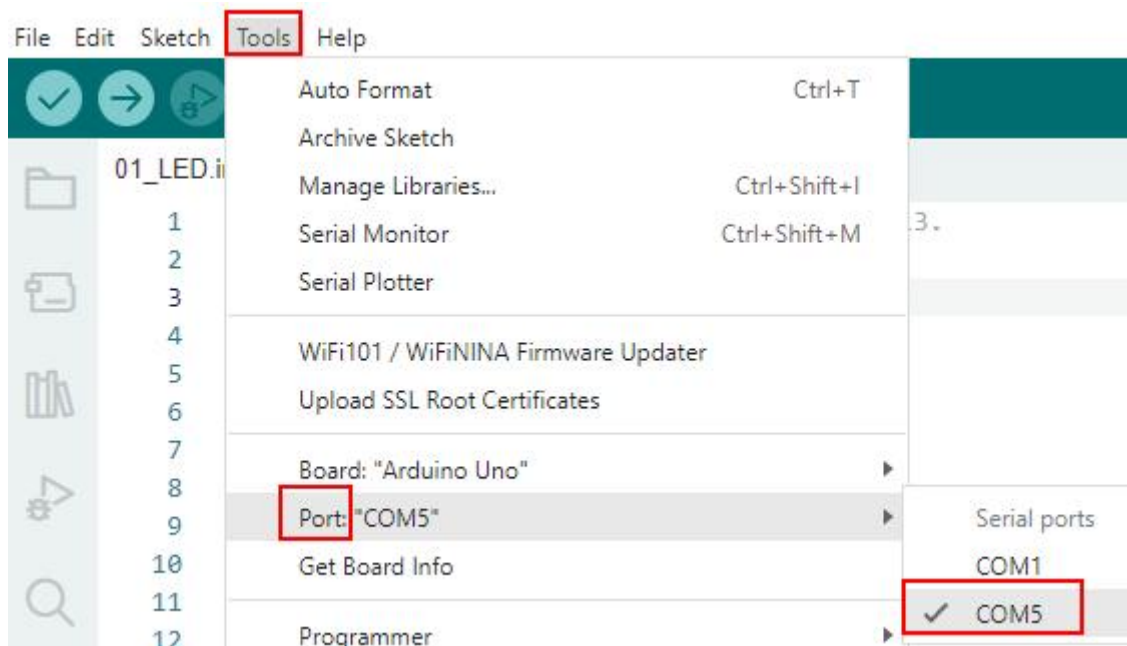
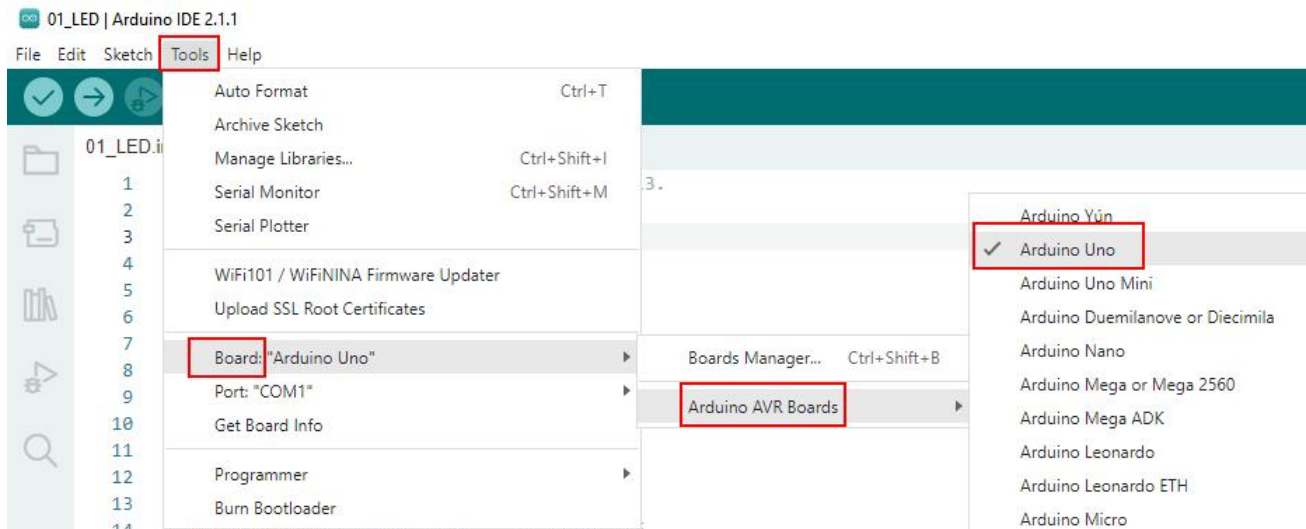
```
09_Photosensitive.ino
1  /*****
2   Author      : www.adeept.com
3   Modification: 2023/08/08
4   *****/
5
6  #define PHOTSENSITIVE_PIN A0 //Define the pins that ESP32 reads photosensitive
7  int photosensitiveADC;        //Defines a variable to store ADC values
8
9  void setup()
10 {
11   pinMode(PHOTSENSITIVE_PIN, INPUT); //Configure the pins for input mode
12   Serial.begin(115200);               //Initialize the serial port and set the baud rate
13 }
14
15 void loop()
16 {
17   photosensitiveADC = analogRead(PHOTSENSITIVE_PIN); //Read the photosensitive resistor value
18   Serial.print("photosensitiveADC: ");
19   Serial.println(photosensitiveADC);                //Print photosensitive resistor value
20   delay(500);
21 }
22
```


3. Select development board and serial port.

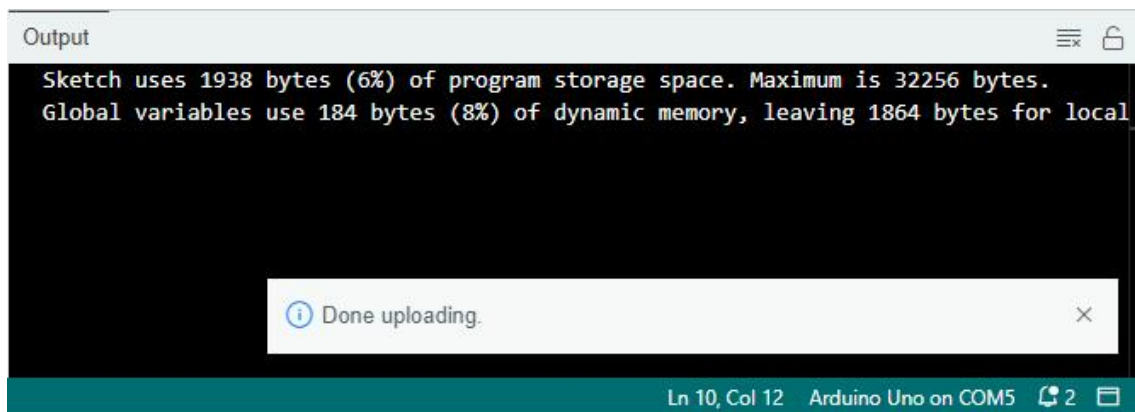
Board: Tools--->Board--->Arduino AVR Boards--->Arduino Uno

Port: Tools --->Port--->COMx

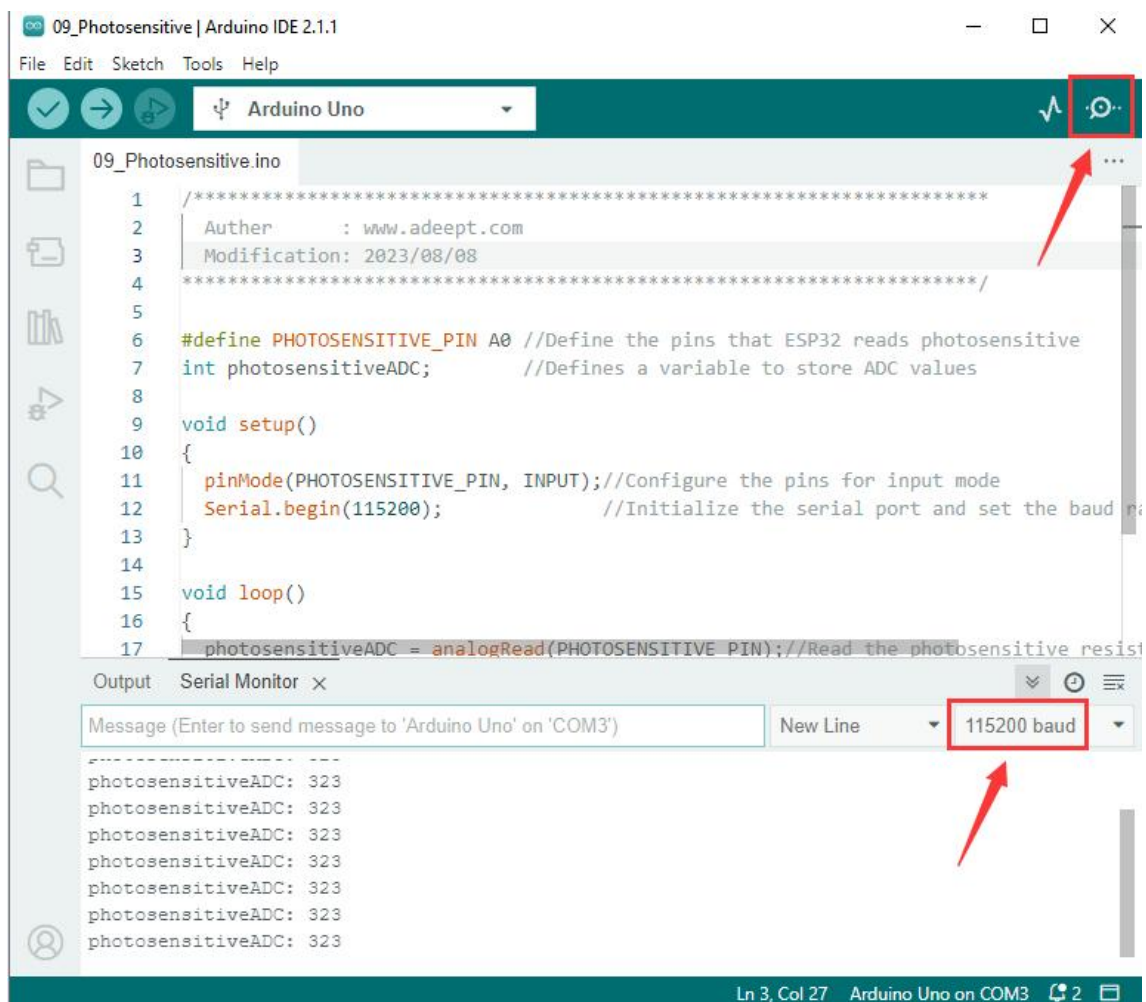
Note: The port number will be different in different computers.



4. After opening, click  to upload the code program to the Arduino. If there is no error warning in the console below, it means that the Upload is successful.



5. Click Serial Monitor, set the baud rate as 115200.



6. You will see the detected ADC value displayed on the screen.

8.5 Code

```
1. #define PHOTOSENSITIVE_PIN A0 //Define the pins that ESP32 reads photosensitive
2. int photosensitiveADC;          //Defines a variable to store ADC values
3.
4. void setup()
5. {
6.   pinMode(PHOTOSENSITIVE_PIN, INPUT); //Configure the pins for input mode
7.   Serial.begin(115200);             //Initialize the serial port and set the
   baud rate to 115200
8. }
9.
10. void loop()
11. {
12.   photosensitiveADC = analogRead(PHOTOSENSITIVE_PIN); //Read the photosensitive
   resistance value
13.   Serial.print("photosensitiveADC: ");
14.   Serial.println(photosensitiveADC);           //Print photosensitive re
   sistance value
15.   delay(500);
16. }
```