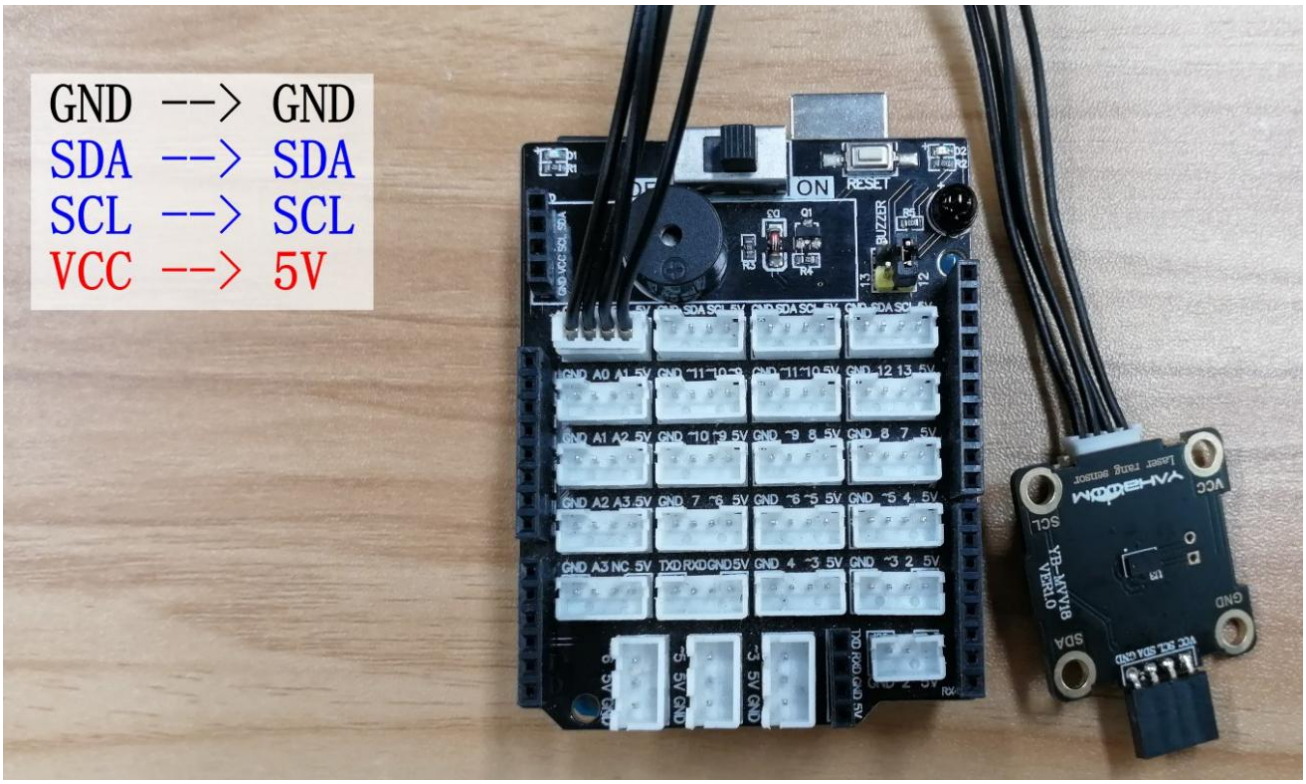
1. Learning target

In this course, we will earn how to use Arduino and laser ranging module to achieve ranging.

2. Preparation

Connect the module to Arduino board as shown below.



3. Program analysis

3.1 Initialize the serial port, initialize I2C, initialize the mode.

Then start the ranging.

During initialization, three ranging modes are realized by setting the initialization mode:

VL53L0X_high_accuracy_ranging_init() is a high-precision mode. In this mode, the accuracy will be higher, but the reading speed will be slower.

VL53L0X_high_speed_ranging_init () is a high-speed mode. In this mode, the accuracy is lower, but the reading speed is faster.

VL53L0X_long_distance_ranging_init () is the long-distance mode, in which the long-distance accuracy is higher.

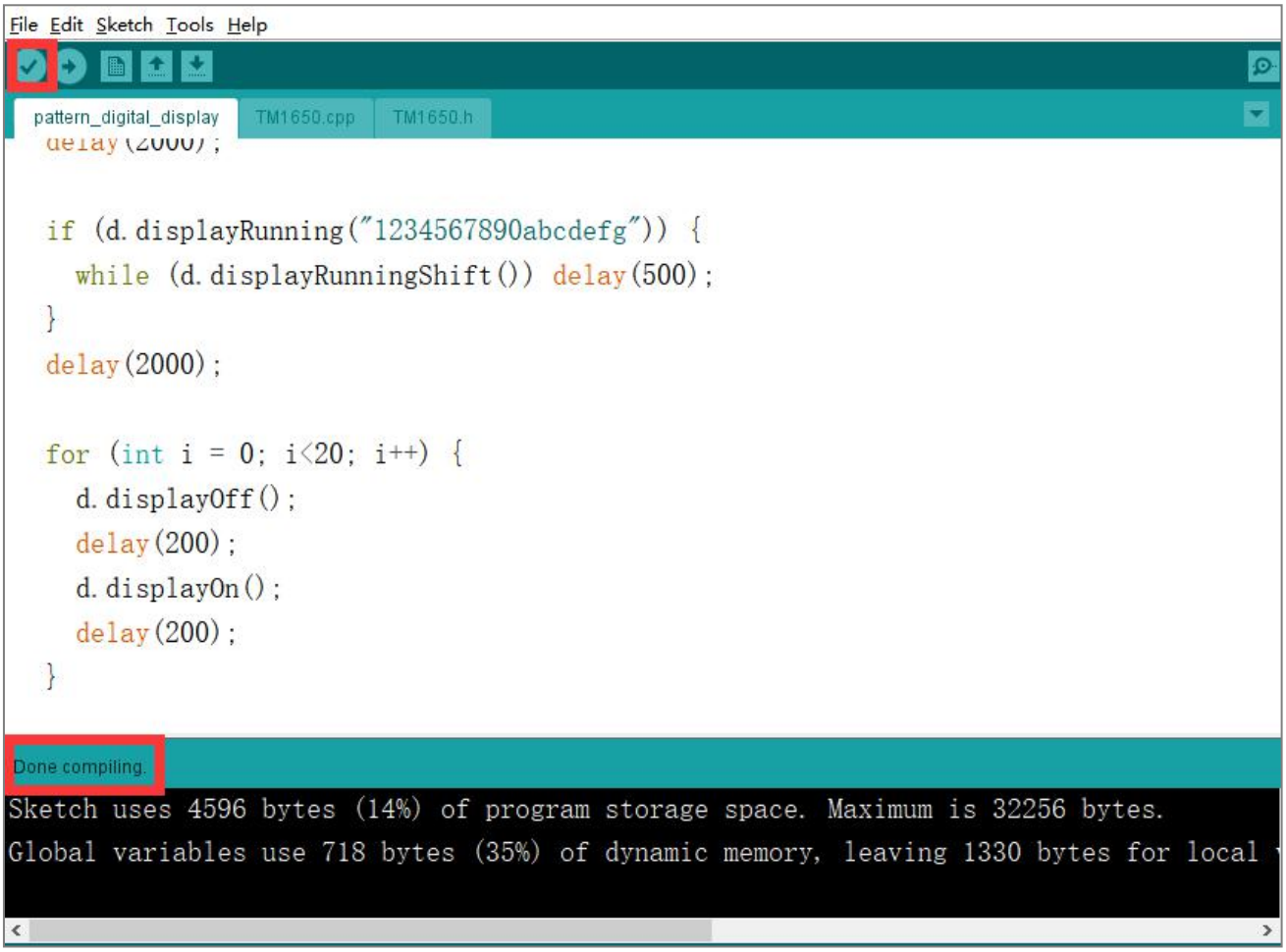
```
void setup()
{
    VL53L0X_Error Status = VL53L0X_ERROR_NONE;
    Serial.begin(115200);
    Status=VL53L0X.VL53L0X_common_init();
    if(VL53L0X_ERROR_NONE!=Status)
    {
        Serial.println("start vl53l0x mesurement failed!");
        VL53L0X.print_pal_error(Status);
        while(1);
    }

    VL53L0X.VL53L0X_high_accuracy_ranging_init();

    if(VL53L0X_ERROR_NONE!=Status)
    {
        Serial.println("start vl53l0x mesurement failed!");
        VL53L0X.print_pal_error(Status);
        while(1);
    }
}
```

3.2 After getting the current ranging result, print out the data through the serial port.

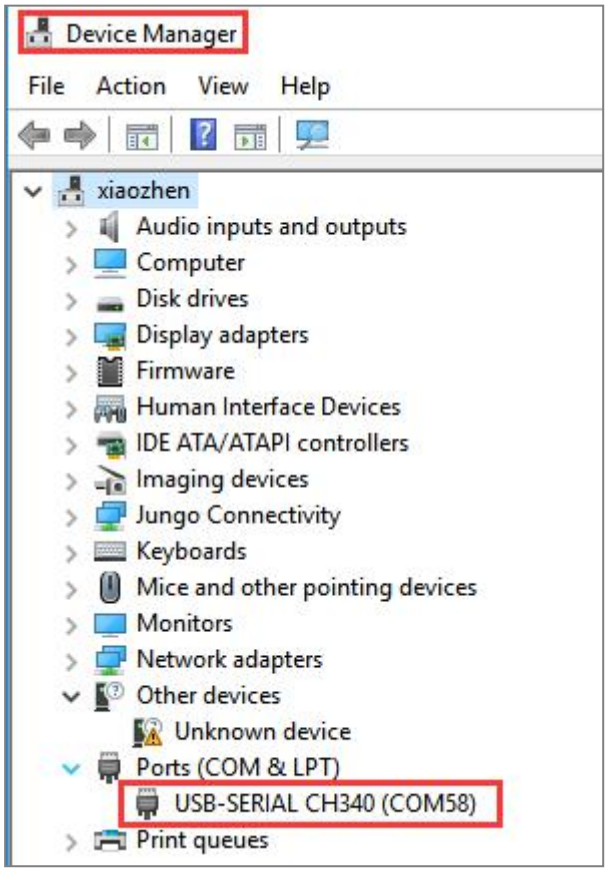
```
void loop()
{
    VL53L0X_RangingMeasurementData_t RangingMeasurementData;
    VL53L0X_Error Status = VL53L0X_ERROR_NONE;

    memset(&RangingMeasurementData,0,sizeof(VL53L0X_RangingMeasurementData_t));
    Status=VL53L0X.PerformSingleRangingMeasurement(&RangingMeasurementData);
    if(VL53L0X_ERROR_NONE==Status)
    {
        if(RangingMeasurementData.RangeMilliMeter>=2000)
        {
            Serial.println("out of range!!");
        }
        else
        {
            Serial.print("Measured distance:");
            Serial.print(RangingMeasurementData.RangeMilliMeter);
            Serial.println(" mm");
        }
    }
    else
    {
        Serial.print("mesurement failed !! Status code =");
        Serial.println(Status);
    }

    delay(300);
}
```

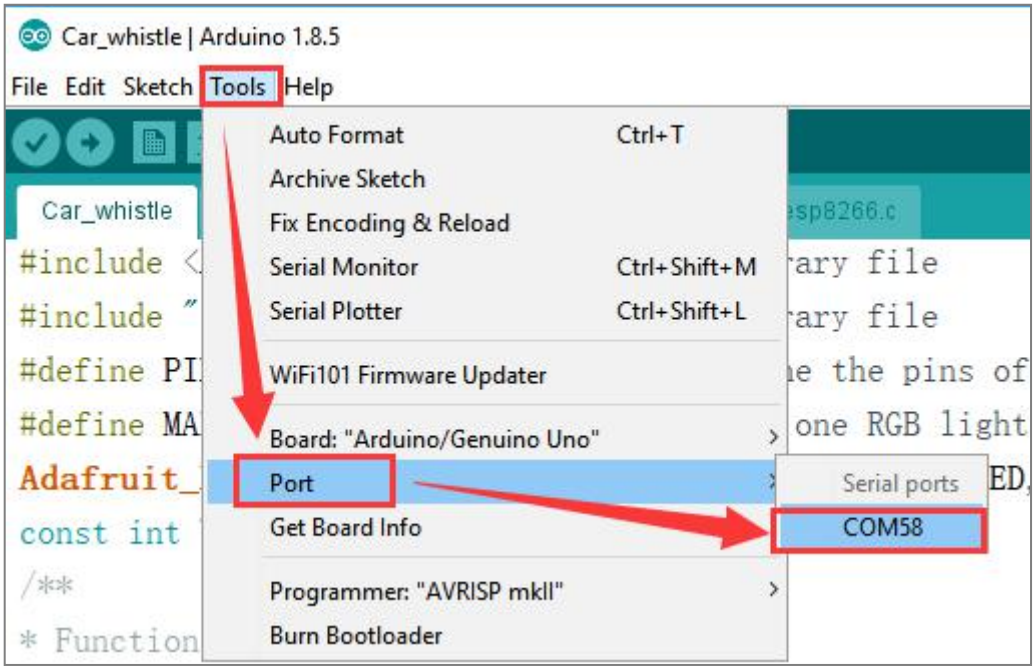
4. Compiling and downloading code

4.1 We need to open the **.ino** file by Arduino IDE software. Then click“√”under the menu bar to compile the code, and wait for the word "Done compiling " in the lower left corner, as shown in the figure below.

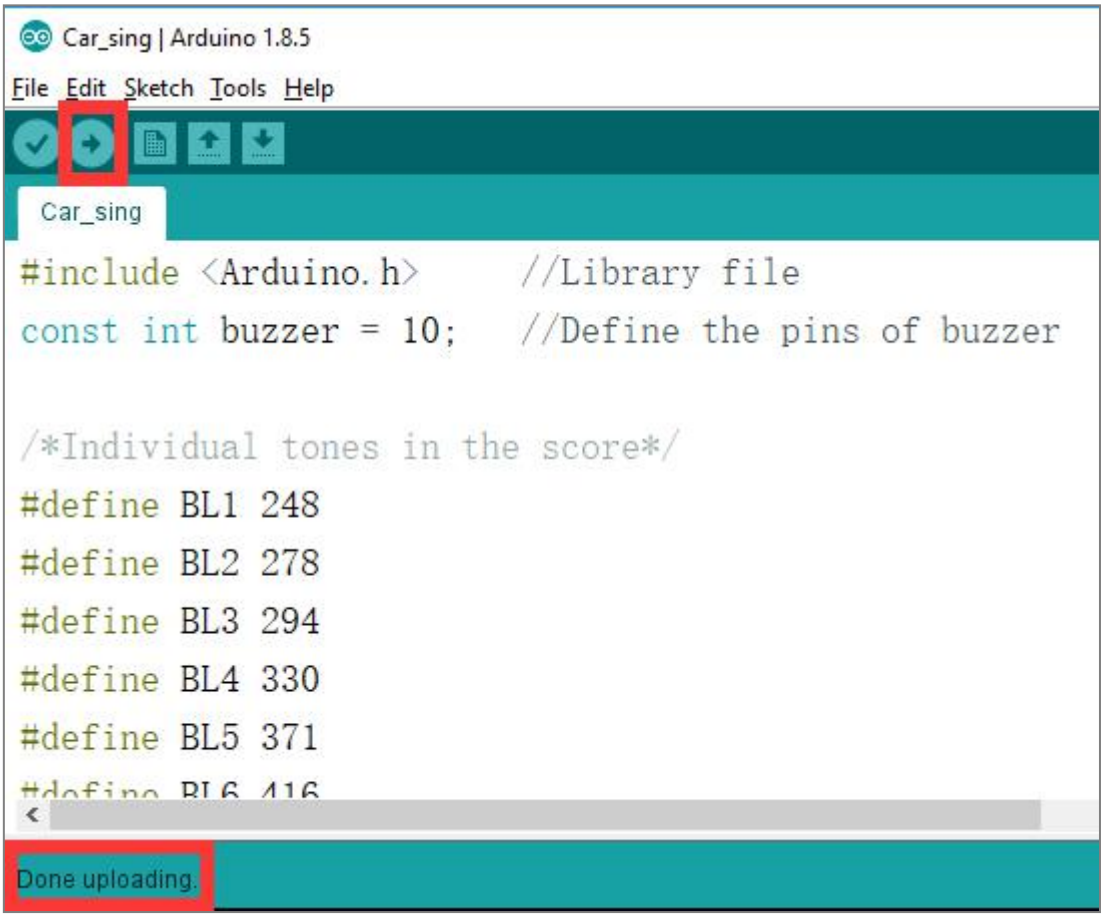


4.2 In the menu bar of Arduino IDE, we need to select【Tools】---【Port】--- selecting the port that the serial number displayed by the device manager just now, as shown in the figure below.





4.3 After the selection is completed, you need to click "→"under the menu bar to upload the code to the UNO board. When the word "Done uploading" appears in the lower left corner, the code has been successfully uploaded to the UNO board, as shown in the figure below.



5. Phenomenon

After the program is downloaded successfully, the program will be automatically initialized and calibrated according to the current environment.

Then, the serial monitor will display the currently detected distance.

Test range is 300mm-2000mm.