1. Learning Objectives

In this course, we mainly learn to use arduino to drive the TT encoder motor to measure the speed.

2. Preparation before class

In this course, we use the TB6612 motor driver module, and other motor driver modules can also be used, which are only used for driving the motor function.

- About wiring:

TB6612module AIN1 ----> arduino D6

TB6612module AIN2 ----> arduino D7

TB6612module PWMA ----> arduino D9

TB6612module STBY ----> arduino D8

TB6612module VCC ----> arduino 5V

TB6612module VM ----> 6-12Vpower

TB6612module GND ----> arduino GND

TB6612module AO1 ----> motor M+

TB6612module AO2 ---->motor M-

TT motor encoder A ----> arduino D2

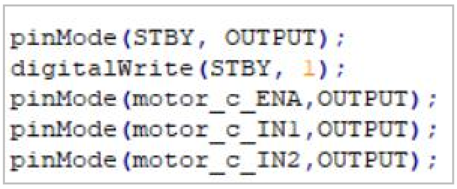
TT motor encoder B----> arduino D3

TT motor encoder VCC----> arduino 5V

TT motor encoder GND----> arduino GND

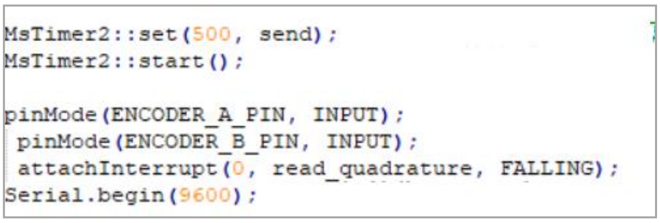
- About code:

Initialize each port



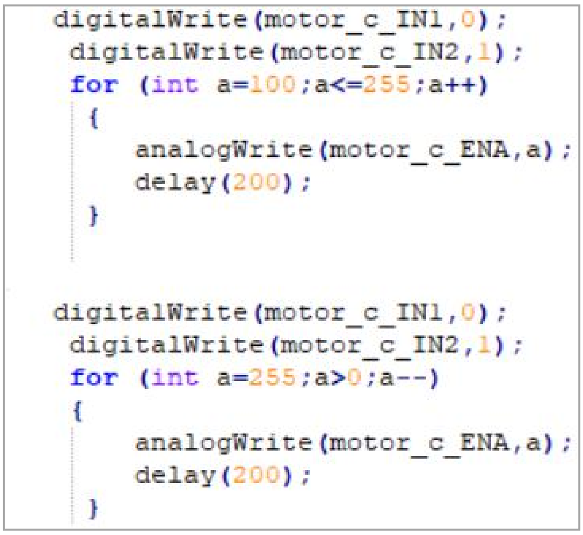
```
pinMode(STBY, OUTPUT);
digitalWrite(STBY, 1);
pinMode(motor_c_ENA,OUTPUT);
pinMode(motor_c_IN1,OUTPUT);
pinMode(motor_c_IN2,OUTPUT);
```

Use timer 2 for motor speed measurement, and send the result every 500ms



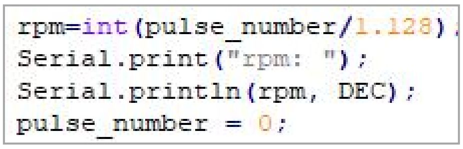
```
MsTimer2::set(500, send);
MsTimer2::start();

pinMode(ENCODER_A_PIN, INPUT);
 pinMode(ENCODER_B_PIN, INPUT);
 attachInterrupt(0, read_quadrature, FALLING);
Serial.begin(9600);
```

The motor speed slowly increases and then slowly decreases



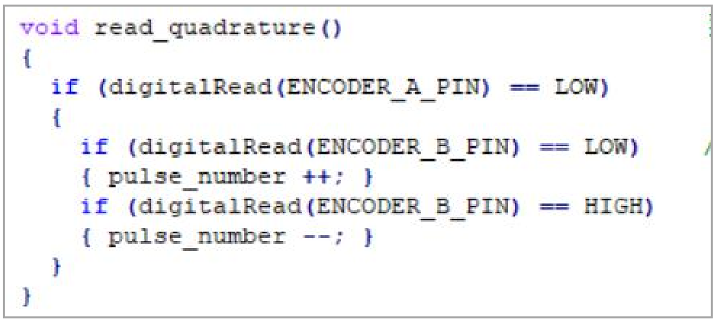
```
  digitalWrite(motor_c_IN1,0);
   digitalWrite(motor_c_IN2,1);
   for (int a=100;a<=255;a++)
    {
        analogWrite(motor_c_ENA,a);
        delay(200);
    }


  digitalWrite(motor_c_IN1,0);
   digitalWrite(motor_c_IN2,1);
   for (int a=255;a>0;a--)
   {
        analogWrite(motor_c_ENA,a);
        delay(200);
    }
```

Printing speed rpm, 1.128 here is the most accurate ratio obtained after many actual speed measurements, and the ratio of 3 lines and 13 lines is different.



```
rpm=int(pulse_number/1.128);
Serial.print("rpm: ");
Serial.println(rpm, DEC);
pulse_number = 0;
```

Judging motor direction by quadrature level.



```
void read_quadrature()
{
  if (digitalRead(ENCODER_A_PIN) == LOW)
  {
    if (digitalRead(ENCODER_B_PIN) == LOW)
    { pulse_number ++; }
    if (digitalRead(ENCODER_B_PIN) == HIGH)
    { pulse_number --; }
  }
}
```

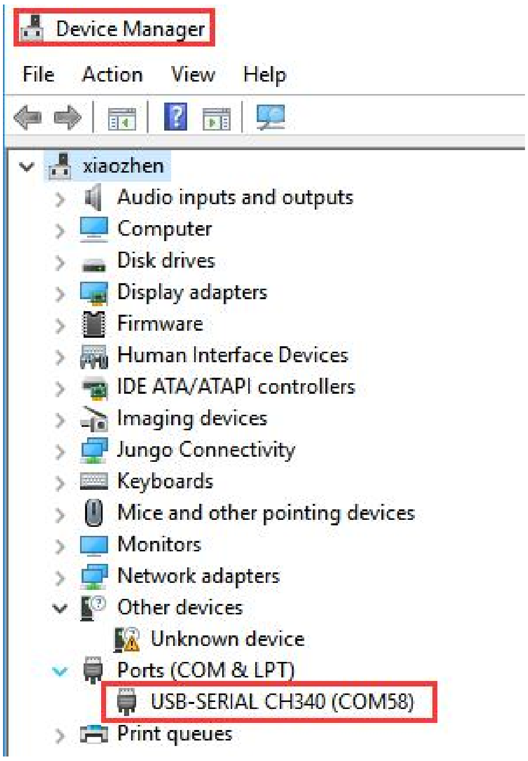
3. Compile and upload code

3.1You need to double-click to open the Arduino software that is installed on the desktop.

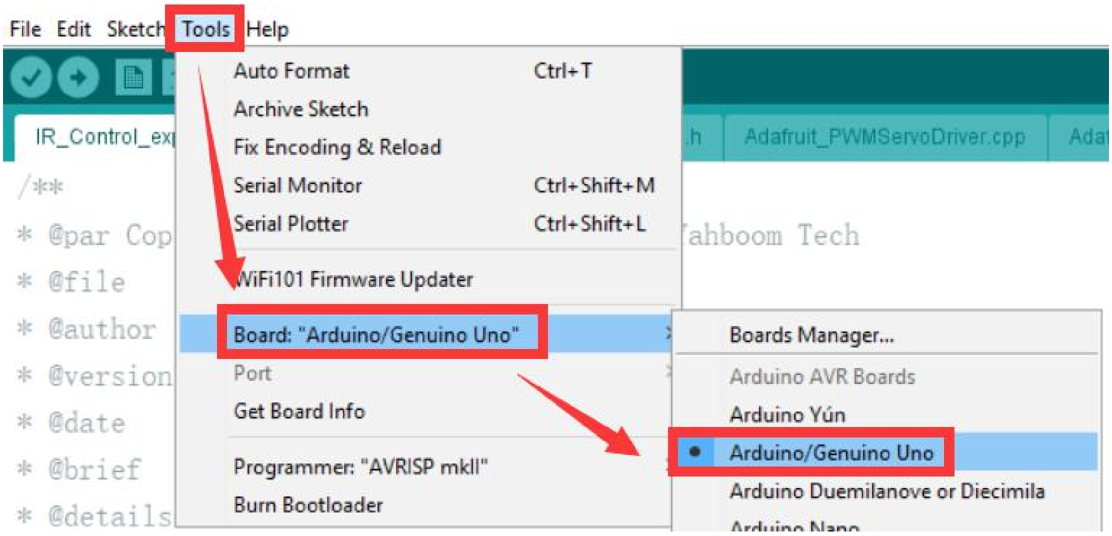
3.2You need to connect the Arduino UNO board to the computer with the data cable.

And select “Port” in the device manager of your computer, marked with CH340 port. For example:USB-SERIAL CH340(COM58), as shown in the following picture.

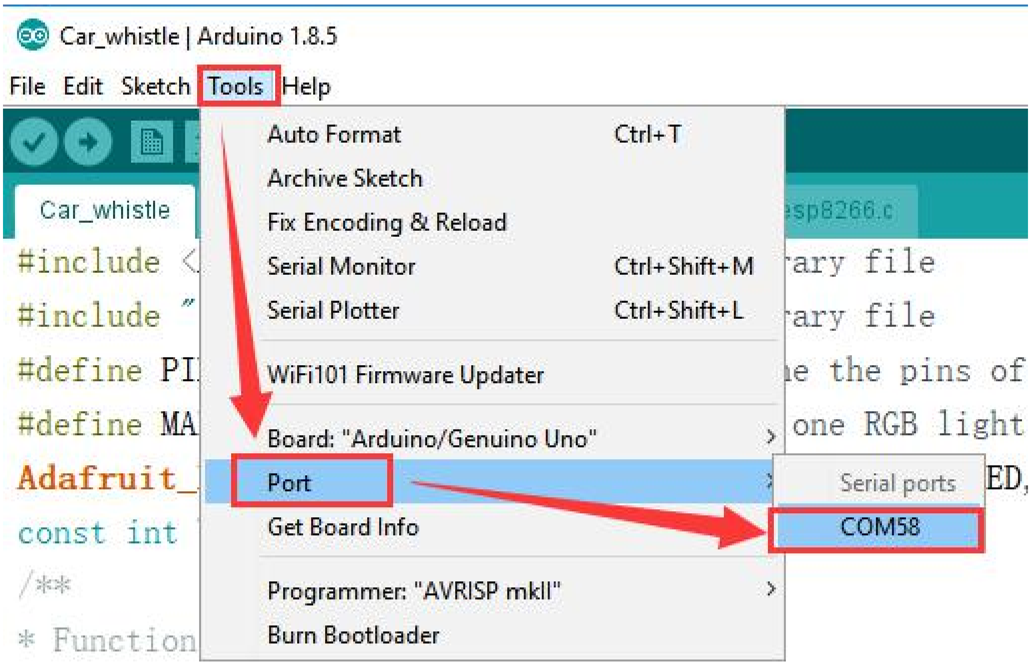
Before this step, please refer to the "Installation of CH340 Driver" file to complete the installation of the CH340 driver. Otherwise, the port will not be recognized.



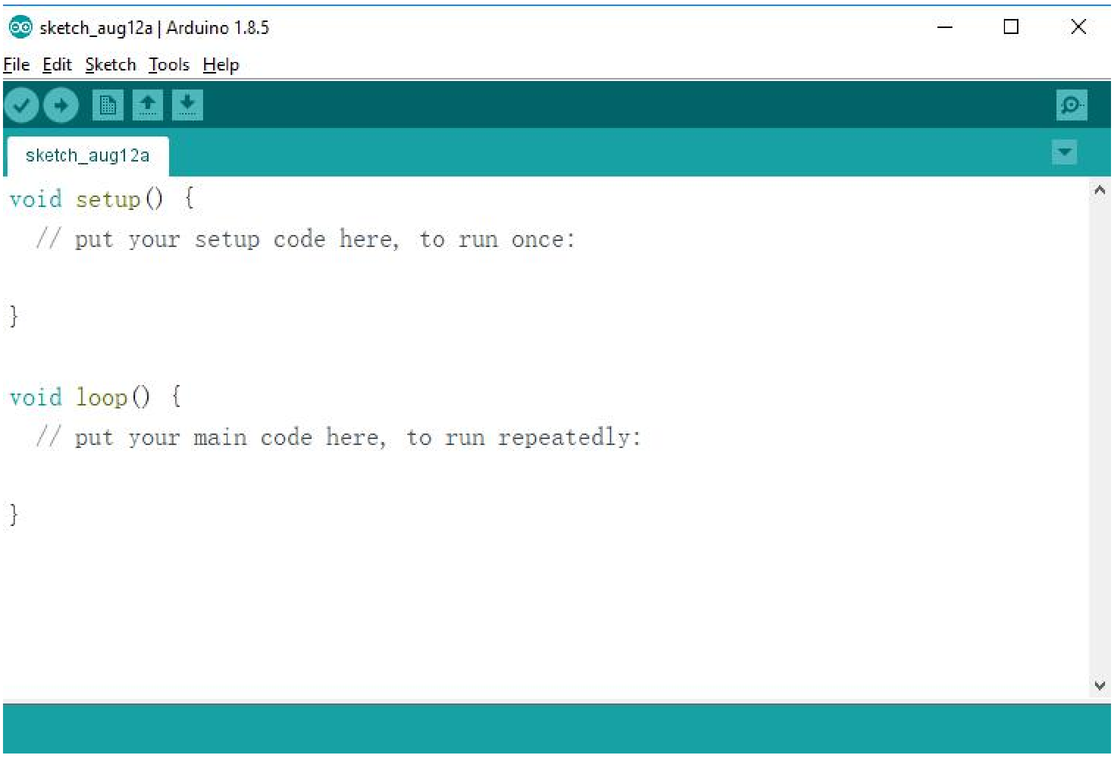
3.3 You need to select 【Tool】--【Board】--【Arduino Uno】 in the Arduino IDE menu bar, as shown in the following picture.



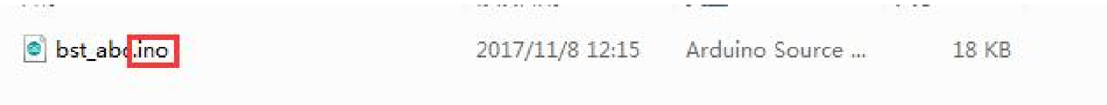
3.4 In the menu bar of Arduino IDE, you can select the 【Tools】---【Port】--- select the port that the serial number displayed by the device manager just now. for example:COM58.


Car_whistle | Arduino 1.8.5
File Edit Sketch Tools Help
Car_whistle
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 Firmware Updater
Board: "Arduino/Genuino Uno"
Port
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
Serial ports
COM58

3.5 You can write codes in the blank area or directly open the file in .ino format, as shown in the following picture.

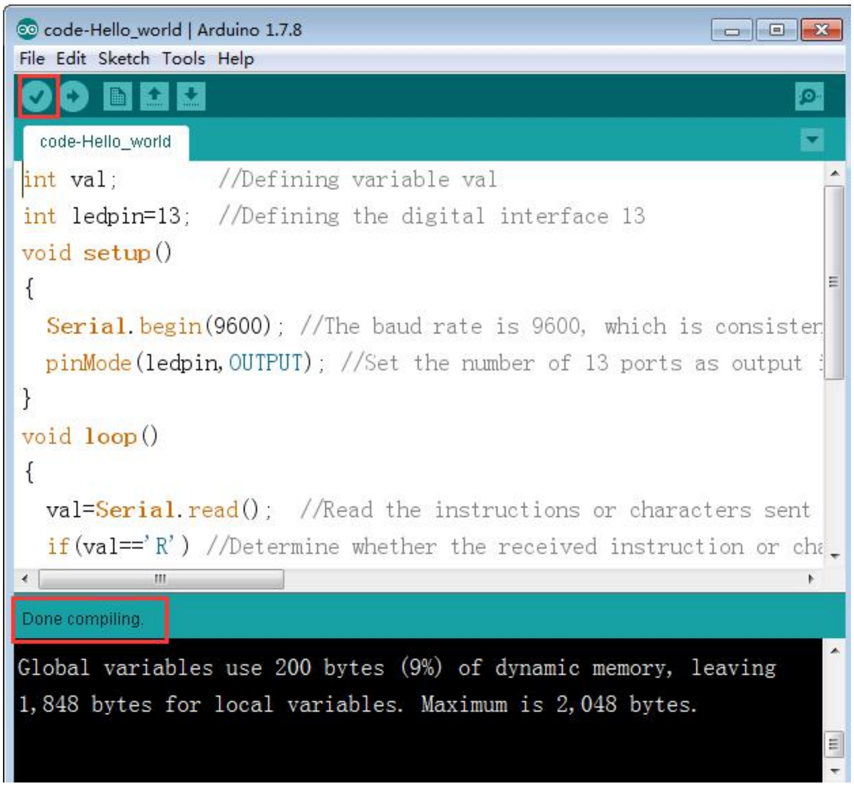


```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```



bst_abc.ino 2017/11/8 12:15 Arduino Source ... 18 KB

3.6 After the codes is written, you need to click"√" in the menu bar to compile the program, and wait for the word "Done compiling " in the lower right corner, as shown in the figure below.

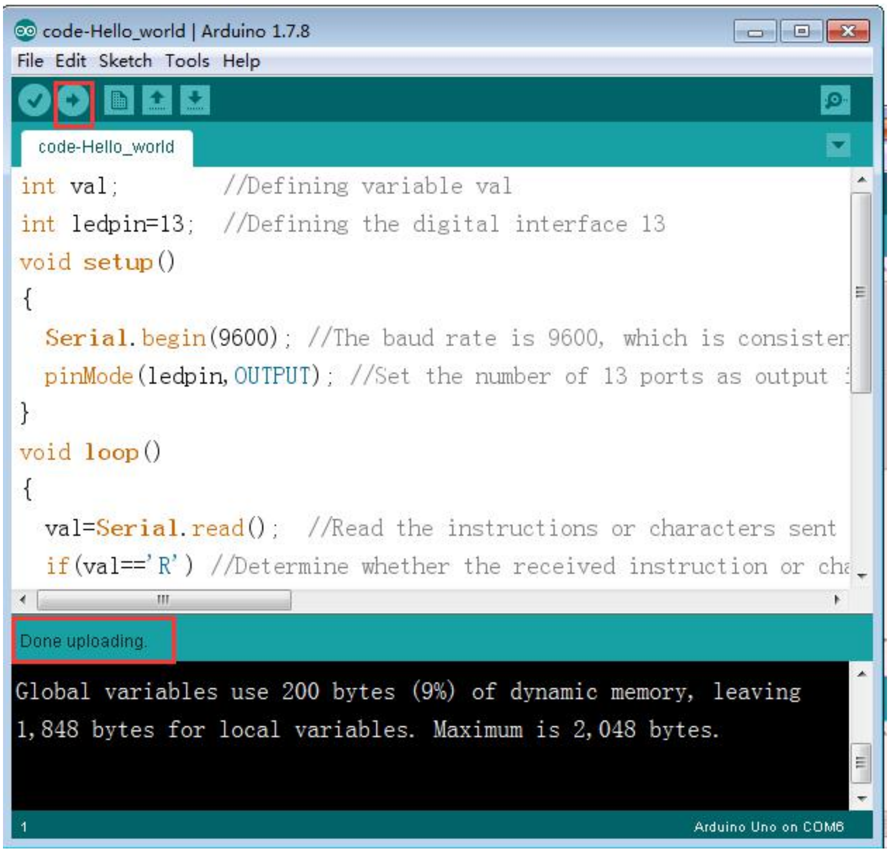


```
int val;        //Defining variable val
int ledpin=13;  //Defining the digital interface 13
void setup()
{
  Serial.begin(9600); //The baud rate is 9600, which is consisten
  pinMode(ledpin,OUTPUT); //Set the number of 13 ports as output i
}
void loop()
{
  val=Serial.read();  //Read the instructions or characters sent
  if(val=='R') //Determine whether the received instruction or cha
```

Done compiling.

```
Global variables use 200 bytes (9%) of dynamic memory, leaving
1,848 bytes for local variables. Maximum is 2,048 bytes.
```

3.7 You can click “→”under the menu bar to upload the program to the UNO board. When the word “Done uploading” appears in the lower left corner, the program has been successfully uploaded to the UNO board, as shown in the figure below.



```
int val;        //Defining variable val
int ledpin=13;  //Defining the digital interface 13
void setup()
{
  Serial.begin(9600); //The baud rate is 9600, which is consisten
  pinMode(ledpin,OUTPUT); //Set the number of 13 ports as output i
}
void loop()
{
  val=Serial.read();  //Read the instructions or characters sent
  if(val=='R') //Determine whether the received instruction or cha
```

Done uploading.

```
Global variables use 200 bytes (9%) of dynamic memory, leaving
1,848 bytes for local variables. Maximum is 2,048 bytes.
```

After these steps, the program is uploaded to the UNO board successfully.

4. Experimental phenomenon

After the program is downloaded successfully, the motor rotates. Click the serial monitor🔍 to see the corresponding speed.