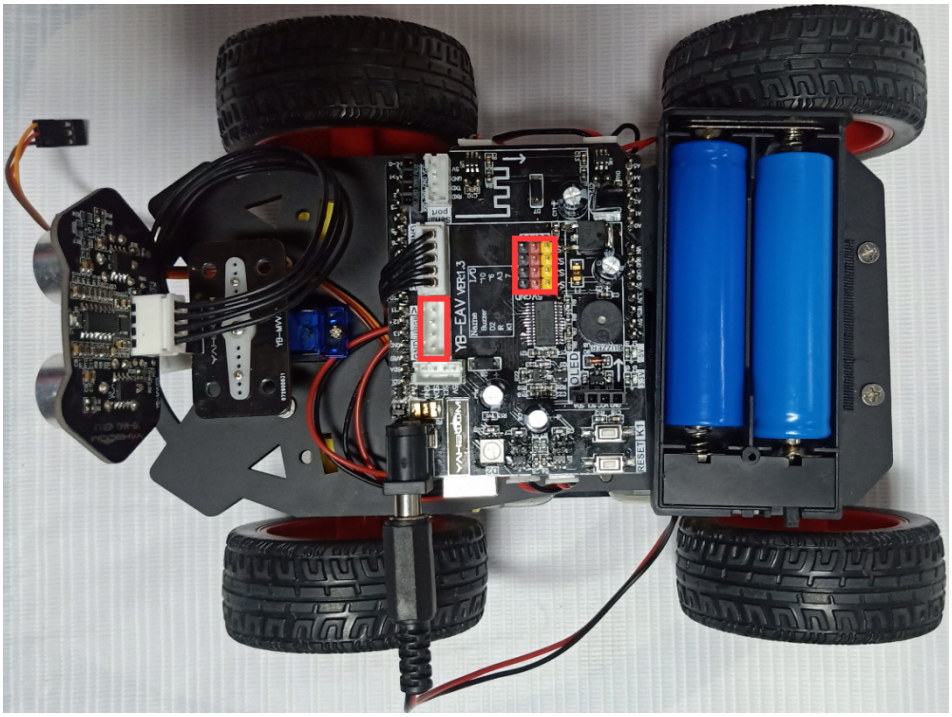
Expand course ---5.Servo Color RGB

1. Learning goal

In this lesson, we will learn how to control servo(180°) and Ultrasonic RGB light at the same time.

2. Preparation

2.1 The position of the servo port and Ultrasonic module port on the expansion board. As shown below.



2.2 The pin of UNO board is connected the servo on the expansion board .
2.3 The pin of UNO board is connected the Ultrasonic RGB light on the expansion board .

3. Principle of experimental

The working principle of the servo:

the control signal enters the signal modulation chip from the channel of the receiver to obtain the bias voltage of the DC. It has a reference circuit inside, which generates a reference signal with a period of 20ms and a width of 1.5ms. It will compares the DC bias voltage with the voltage of the potentiometer to obtain a voltage difference and output. The positive and negative of the voltage difference is outputted to the motor drive chip to determine the forward and reverse of the motor.

Servo rotation angle is by adjusting the duty ratios of PWM (pulse width modulation) signal. The standard PWM (pulse width modulation) signal has a fixed period of 20ms (50Hz). Theoretically, pulse width distribution should be between 1 ms to 2 ms, but in fact between pulse width can be 0.5 ms and 2.5

ms. Pulse width and the servo rotation angle 0°～180° corresponds, as shown below.

0.5ms----------0°
1.0ms----------45°
1.5ms----------90°
2.0ms----------135°
2.5ms----------180°

About Ultrsonic RGB light:
RGB light (red, green, blue) are packaged in the LED module. We can mix different colors(256*256*256) by controlling the brightness of the three LEDs.

From the hardware interface manual, we can know that three servos are driven by PWM by PCA9685 chip.
PCA9685 chip and UNO board adopt I2C communication method. I2C address is 0x40.
connection as shown below.

PCA9685 chip	Servo
0	S1
1	S2
2	S3
3	S4

In this lesson, we will control No.1 servo. Other servo can be controlled by the same way.
From the hardware interface manual, we can know that Ultrasonic RGB light is driven by Pin 11 of UNO board.

4. About code
For the code of this course, please refer to: **Servo_Color_RGB.ino** in the **Servo_Color_RGB** folder.

```
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>
Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver(0x40);
#define SERVOMIN 150 // this is the 'minimum' pulse length count (out of 4096)
#define SERVOMAX 600 // this is the 'maximum' pulse length count (out of 4096)

#include "RGBLed.h"
#define RGB_GREEN      0xFF0000      //Define different color(green,red,blue)
#define RGB_RED     0x00FF00
```

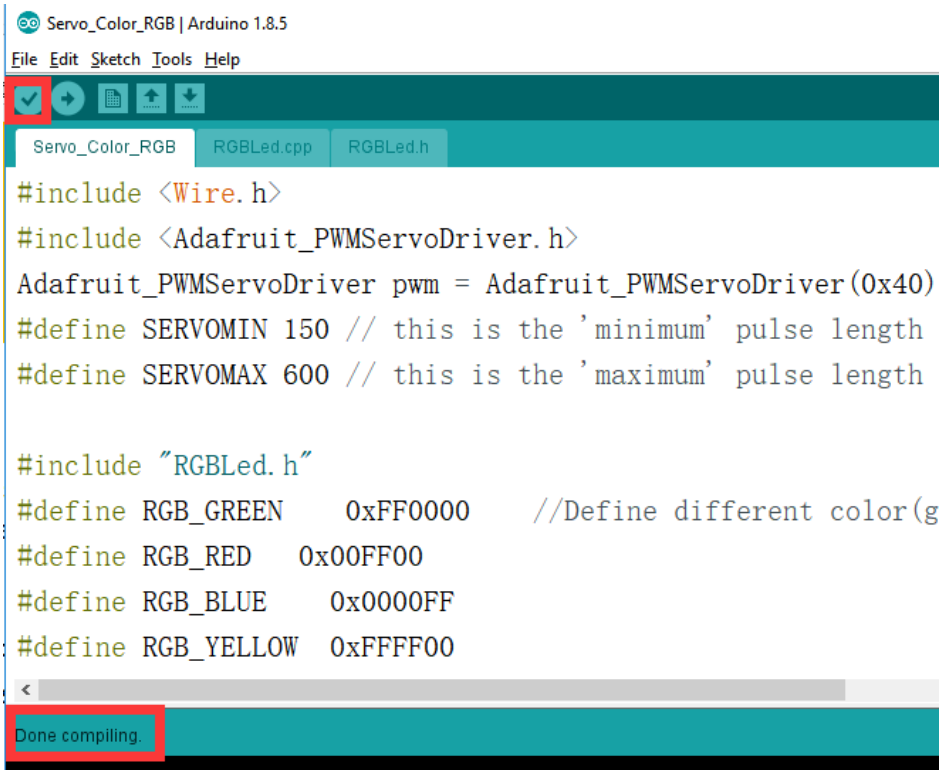
```
#define RGB_BLUE      0x0000FF
#define RGB_YELLOW    0xFFFF00
#define RGB_PURPLE    0x00FFFF
#define RGB_CYAN    0xFF00FF
#define RGB_WHITE     0xFFFFFF
#define RGB_OFF    0

const int RgbPin = 11;       //Define pin of Ultrasonic RGB light
RGBLed mRgb(RgbPin,2);

void Servo_180(int num, int degree)
{

   long us = (degree * 1800 / 180 + 600); // 0.6 ~ 2.4
   long pwmvalue = us * 4096 / 20000;         // 50hz: 20,000 us
   pwm.setPWM(num, 0, pwmvalue);
}

void LOGO_breathing_light(int brightness, int time, int increament)
{
   if (brightness < 0)
   {
      brightness = 0;
   }
   if (brightness > 255)
   {
      brightness = 255;
   }
   for (int b = 0; b < brightness; b += increament)
   {
      int newb = map(b, 0, 255, 0, 4095);
      pwm.setPWM(7, 0, newb);
      delay(time);
   }
}

void color_light(int pos)
{
   //When turning at 150-180 degrees, light white color
   if (pos > 150)
   {
      mRgb.setColor(0,RGB_WHITE);
      mRgb.show();
   }
```

```
  //When turning at 125-150 degrees, light red color
  else if (pos > 125)
  {
    mRgb.setColor(0,RGB_RED);
    mRgb.show();
  }
  //When turning at 100-125 degrees, light green color
  else if (pos > 100)
  {
    mRgb.setColor(0,RGB_GREEN);
    mRgb.show();
  }
  //When turning at 75-100 degrees, light blue color
  else if (pos > 75)
  {
    mRgb.setColor(0,RGB_BLUE);
    mRgb.show();
  }
  //When turning at 50-75 degrees, light yellow color
  else if (pos > 50)
  {
    mRgb.setColor(0,RGB_YELLOW);
    mRgb.show();
  }
  //When turning at 25-50 degrees, light purple color
  else if (pos > 25)
  {
    mRgb.setColor(0,RGB_PURPLE);
    mRgb.show();
  }
  //When turning at 0-25 degrees, light cyan color
  else if (pos > 0)
  {
    mRgb.setColor(0,RGB_CYAN);
    mRgb.show();
  }
  else
  {
    mRgb.setColor(0,RGB_OFF);
    mRgb.show();
  }
}

void Servo_Color_RGB()
```

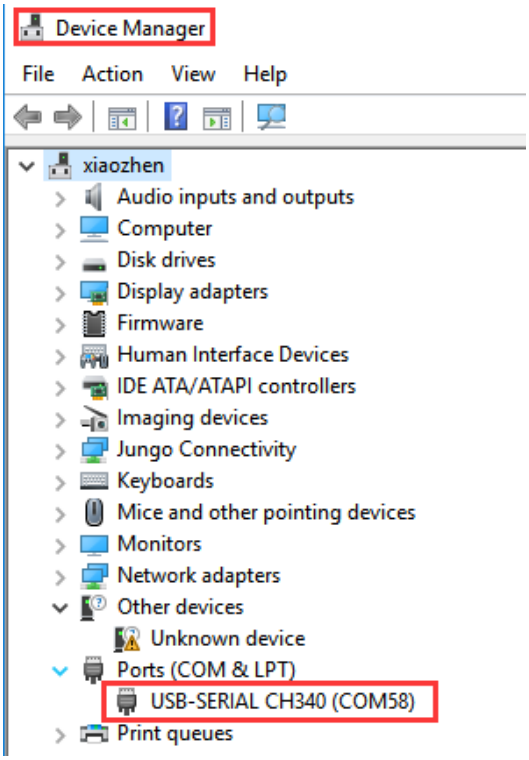
```
{
   int pos = 0;
   for (pos = 0; pos < 180; pos += 10)
   {
      Servo_180(0, pos);
      color_light(pos);
      delay(500);
   }
   for (pos = 180;pos > 0;pos -= 10)
   {
      Servo_180(0, pos);
      color_light(pos);
      delay(500);
   }
}

void setup()
{
    pwm.begin();
    pwm.setPWMFreq(60);    // Analog servos run at ~60 Hz updates
    LOGO_breathing_light(255, 40, 5);     //Gradually light the blue light of the
Yhaboom_LOGO
    Servo_180(0, 90);
}

void loop()
{
      delay(1000);
      while(1)
      {
         Servo_Color_RGB();
      }
}
```

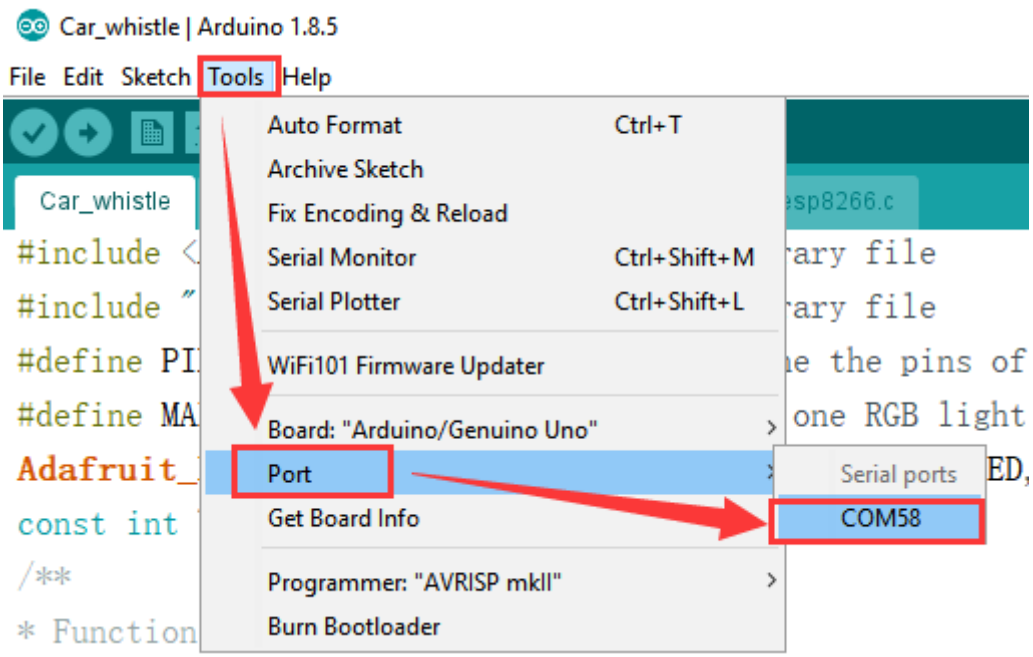
5. Compiling and downloading code

5.1 We need to open the **Servo_Color_RGB.c.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.

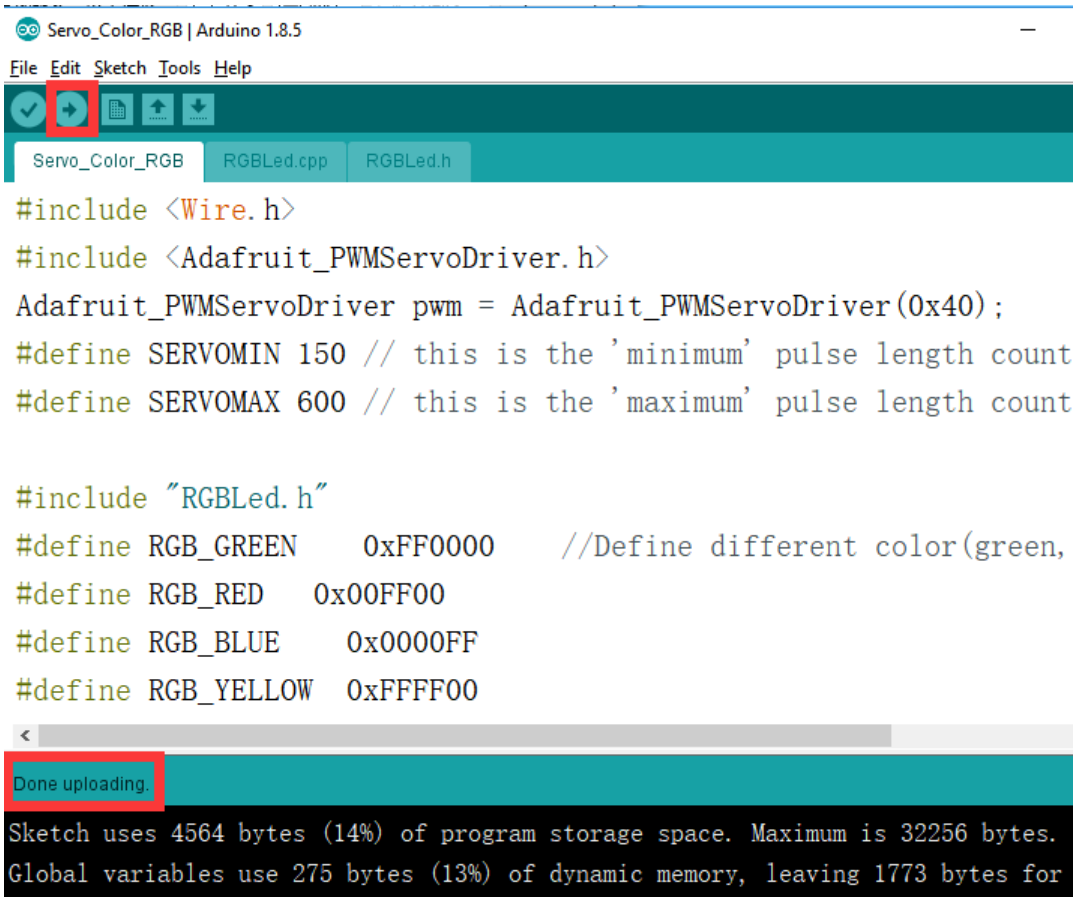


5.2 In the menu bar of Arduino IDE, we need to select 【Tools】---【Port】---selecting the port that the serial number displayed by the device manager just now, as shown in the figure below.





5.3 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.



6. Experimental phenomena

After the program is downloaded. We can see that the RGB lights on the ultrasonic module change different colors as the servo rotates at different angles. As shown below.