

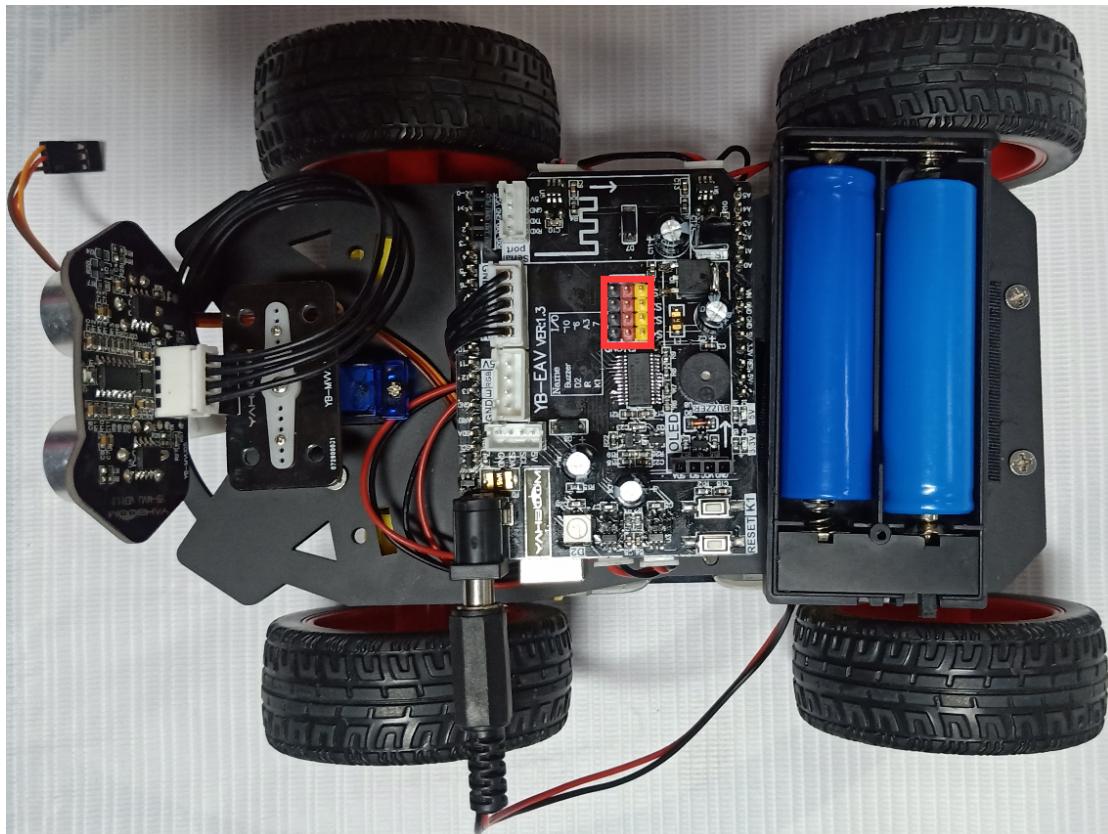
Basic course ---6.Control Servo

1. Learning goal

In this lesson, we will learn how to control servo(180°).

2. Preparation

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



2.2 The pin of UNO board is connected the servo 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° |

From the hardware interface manual, we can know that three servos are driven by PWM by PCA9685 chip.

| Classification | Function | The number of Drive chip PCA9685 | Drive Method | Connection with CPU | Uno board |
|---------------------|------------------------------|----------------------------------|--------------------------|---------------------|-----------|
| Left Motor | Left front motor forward | LINB(13) | PCA9685 | I2C_SDA/I2C_SCL | A4/A5 |
| | Left front motor reverse | LINA(12) | | | |
| | Left rear motor forward | RINB(15) | | | |
| | Left rear motor reverse | RINA(14) | | | |
| Right Motor | Right front motor forward | LED10 | | | A4/A5 |
| | Right front motor reverse | LED11 | | | |
| | Right rear motor forward | LED8 | | | |
| | Right rear motor reverse | LED9 | | | |
| Servo | Control S1 | LED0 | Uno board drive directly | A0 | A0 |
| | Control S2 | LED1 | | | |
| | Control S3 | LED2 | | | |
| | Control S4 | S1 (3) | | | |
| LOGO light | Control bluelight | LED7 | Uno board drive directly | A1 | A1 |
| Tracking sensor | Left tracking sensor | | | | |
| | Middle tracking sensor | | | | |
| | Right tracking sensor | | | | |
| Ultrasonic sensor | Ultrasonic Echo | | Uno board drive directly | A2 | A2 |
| | Ultrasonic RGB light | | | | |
| Key | K1 | | Uno board drive directly | 12 | 12 |
| IR | IR control | | | | |
| Bluetooth interface | RX | | | | |
| | TX | | | | |
| On board RGB Light | RGB Light on expansion board | | Uno board drive directly | 11 | 11 |
| Buzzer | Control buzzer | | | | |
| | | | | | |
| | | | | | |

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.

4. About code

For the code of this course, please refer to: [Control_servo.ino](#) in the **Control_servo** 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)

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);
}

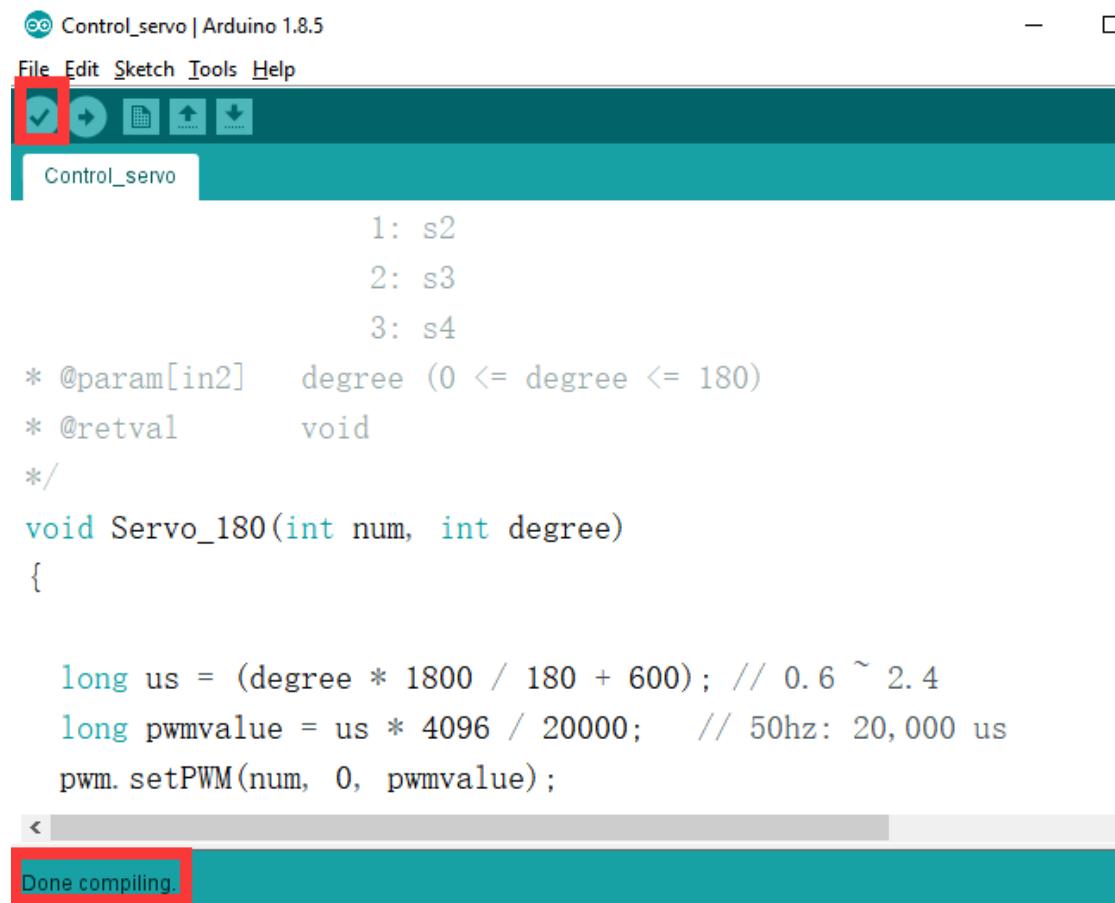
void LOGO_breathing_light(int brightness, int time, int increament)
{
    if (brightness < 0)
    {
        brightness = 0;
    }
    if (brightness > 255)
    {
        brightness = 255;
    }
    for (int b = 0; b < brightness; b += increament)
    {
        int newb = map(b, 0, 255, 0, 4095);
   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

void setup()
{
    pwm.begin();
    pwm.setPWMDutyCycle(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_180(0, 10);      //This speed can be adjusted by yourslef
        delay(500);
        Servo_180(0, 90);      //This speed can be adjusted by yourslef
        delay(500);
        Servo_180(0, 170);      //This speed can be adjusted by yourslef
        delay(500);
        Servo_180(0, 90);      //This speed can be adjusted by yourslef
        delay(500);
    }
}
```

5. Compiling and downloading code

5.1 We need to open the **Control_servo .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 right corner, as shown in the figure below.



Control_servo | Arduino 1.8.5

File Edit Sketch Tools Help

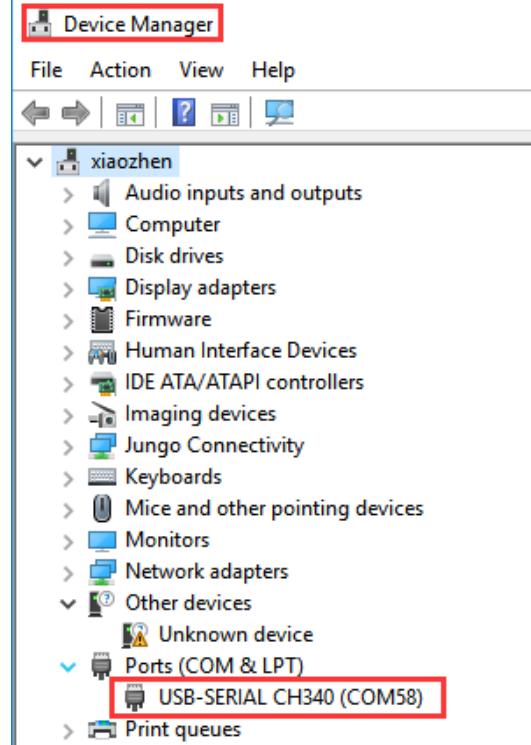
```

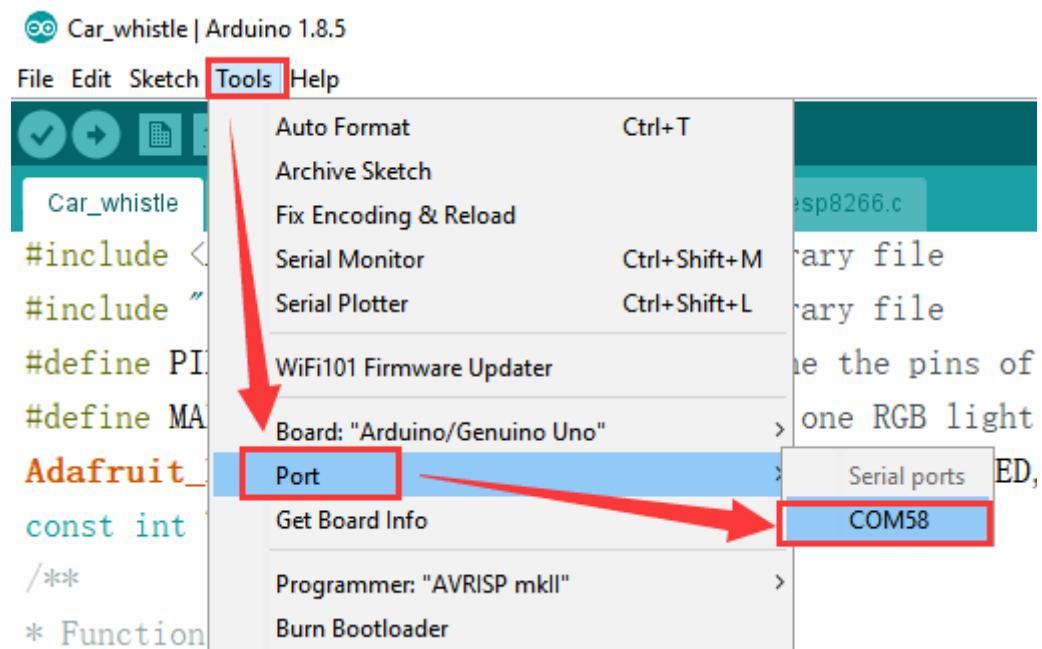
1: s2
2: s3
3: s4
* @param[in] degree (0 <= degree <= 180)
* @retval void
*/
void Servo_180(int num, int degree)
{
    long us = (degree * 1800 / 180 + 600); // 0.6 ~ 2.4
    long pwmvalue = us * 4096 / 20000; // 50hz: 20,000 us
    pwm.setPWM(num, 0, pwmvalue);
}

Done compiling.

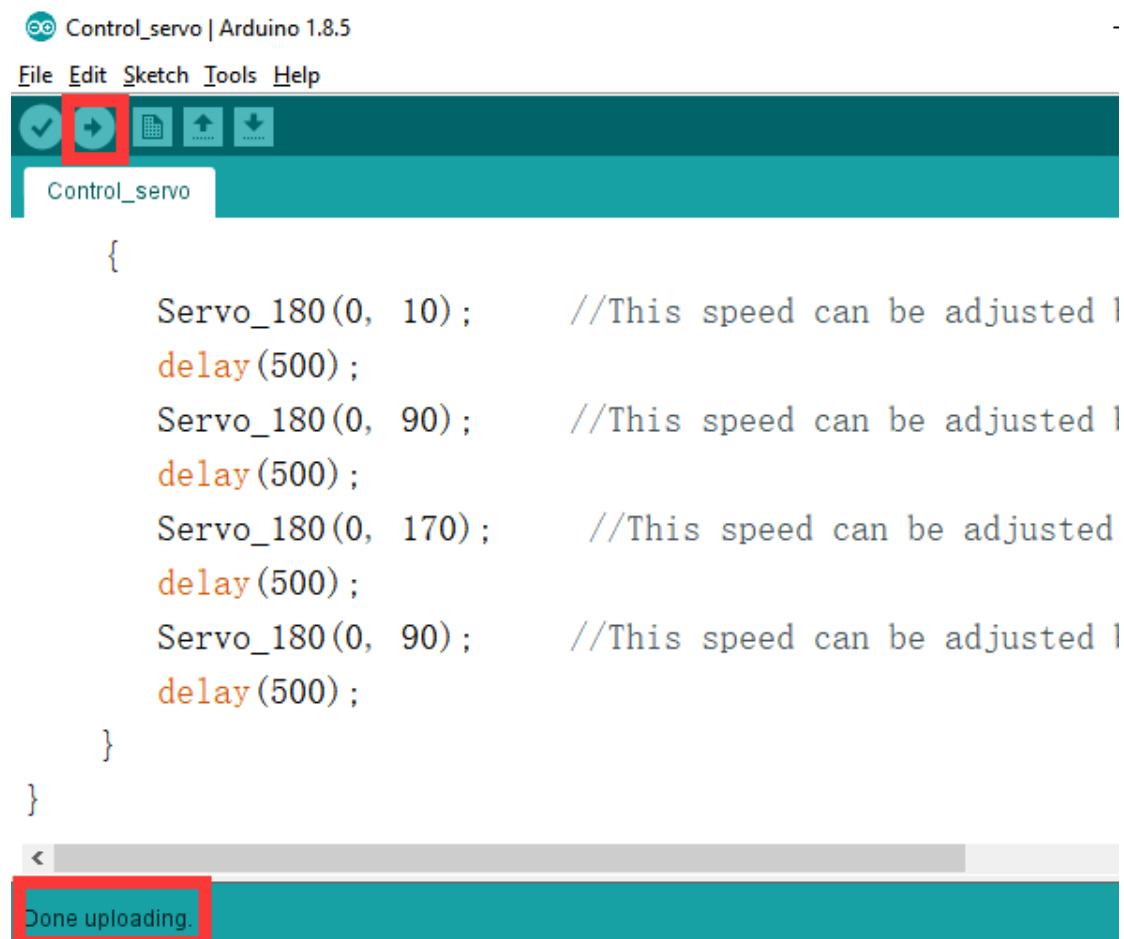
```

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. When we can open the power of car, No.1 servo will rotate to 90°(Initial configuration). Then No.1 servo will rotate to 0°-->90°--> 170°--> 90. And keep looping in this state.