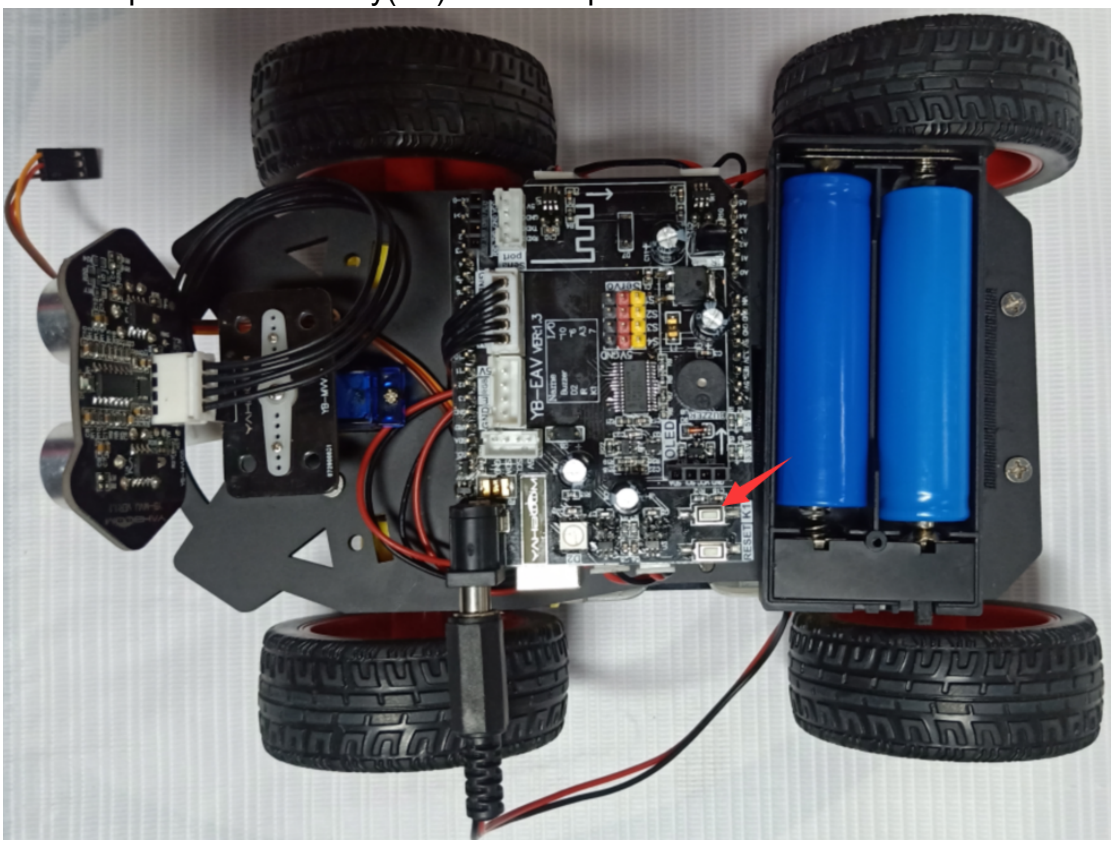
Basic course ---5.Key control RGB

1. Learning goal

In this lesson, we will learn how to use the key on the expansion board.

2. Preparation

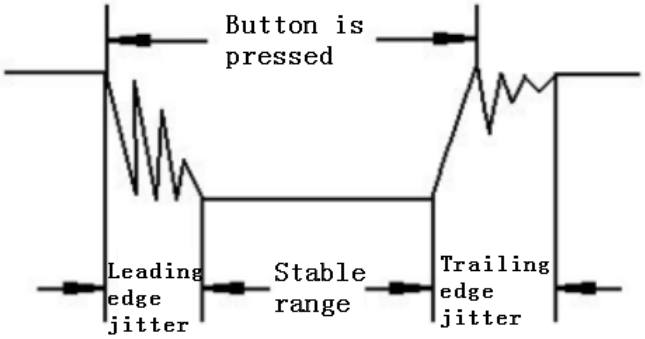
2.1 The position of the key(K1) on the expansion board. As shown below.



2.2 The pin of UNO board is connected the key(K1) on the expansion board .

3. Principle of experimental

Generally, our button switches are mechanical elastic switches. When the mechanical contacts are opened and closed, due to the elastic action of the mechanical contactor, switchs will not be able to be connected immediately when closed and it will not be disconnected at once.



Button jitter state diagram

The jitter time is usually within 10ms. The button must be eliminated jitter to

ensure that the program only responds once after the button is closed once.

In this experiment, we took the software delay eliminated jitter. After detecting that the button is closed, delay codes is executed to generate a delay of 5ms to 10ms, and the state of the button is detected again after the leading edge jitter disappears. When it is detected that the button is released, it also needs to delay 5ms~10ms.

From the hardware interface manual, we can know that On board RGB light is driven by Pin 7 of UNO board.

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			
	Right front motor reverse	LED11			
	Right rear motor forward	LED8			
	Right rear motor reverse	LED9			
Servo	Control S1	LED0			
	Control S2	LED1			
	Control S3	LED2			
	Control S4	S1(3)			
LOGO light	Control bluelight	LED7			
Tarcking sensor	Left tracking sensor	Uno board drive directly			A0
	Middle tracking sensor				A1
	Right tracking sensor				A2
Ultrasonic sensor	Ultrasonic Echo				12
	Ultrasonic RGB Light				11
Key	K1				7
IR	IR control				A3
Bluetooth interface	RX				0
	TX				1
On board RGB Light	RGB Light on expansion board				6
Buzzer	Control buzzer				10

In the lesson, when key is pressed, Pin 7 get LOW level; when key is not pressed, Pin 7 get HIGH level.

4. About code

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

```
#include <Wire.h>
```

```
#include <Adafruit_PWMServoDriver.h>
#include <Adafruit_NeoPixel.h>        //Library file
#define PIN 6                                 //Define the pins of the RGB light
#define MAX_LED 1                           //Just one RGB light on the car
#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_WHITE     0xFFFFFF
#define RGB_OFF    0x000000
Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver(0x40);
Adafruit_NeoPixel strip = Adafruit_NeoPixel( MAX_LED, PIN, NEO_RGB +
NEO_KHZ800 );
const int key = 7;              //Define pin of key
const int RgbPin = 12;       //Define pin of ultrasonic RGB light
const int buzzer = 10;       //Define pin of buzzer
RGBLed mRgb(RgbPin,2);

void setup()
{
    pinMode(key, INPUT);
    pinMode(RgbPin,OUTPUT);
    pinMode(buzzer,OUTPUT);
    mRgb.setColor(1,RGB_OFF);
    mRgb.setColor(2,RGB_OFF);
    mRgb.show();
    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
    strip.begin();
    strip.show();
}

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 whistle()
{
  for (int i = 0; i < 100; i++)
  {
    digitalWrite(buzzer, HIGH); //Sound
    delay(1);
    digitalWrite(buzzer, LOW);   //Nosound
    delay(1);
  }
}

void keyscan()
{
  while (digitalRead(key));          //When the button not press, it while
  while (!digitalRead(key))          //When the button pres
  {
    delay(10);                              //delay 10ms
    if (digitalRead(key)   ==   LOW)//The second time to determine whether
the button is pressed
    {
      delay(100);
      while (!digitalRead(key));   //Test whether the key is released
    }
  }
 }

void Onboard_RGB(int R, int G, int B)
{
  uint8_t i = 0;
  R = map(R, 0, 255, 0, 10);
  G = map(G, 0, 255, 0, 10);
  B = map(B, 0, 255, 0, 10);
  uint32_t color = strip.Color(G, R, B);
```

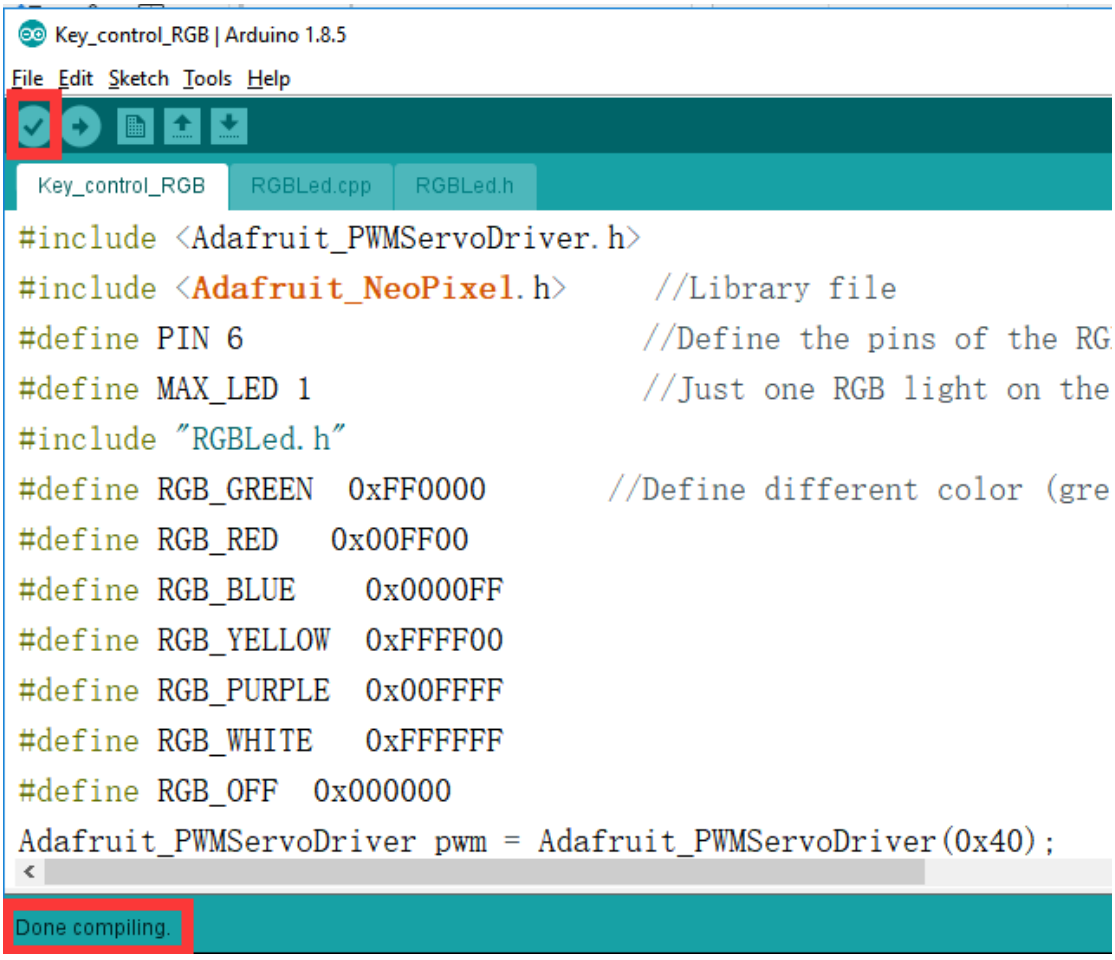
```
    strip.setPixelColor(i, color);
    strip.show();
}

void loop()
{
      keyscan();
      whistle();
      delay(500);
      Onboard_RGB(255,255,255);
      mRgb.setColor(1,RGB_WHITE);
      mRgb.setColor(2,RGB_WHITE);
      mRgb.show();
      delay(500);
      Onboard_RGB(255,0,0);
      mRgb.setColor(1,RGB_RED);
      mRgb.setColor(2,RGB_RED);
      mRgb.show();
      delay(500);
      Onboard_RGB(0,255,0);
      mRgb.setColor(1,RGB_GREEN);
      mRgb.setColor(2,RGB_GREEN);
      mRgb.show();
      delay(500);
      Onboard_RGB(0,0,255);
      mRgb.setColor(1,RGB_BLUE);
      mRgb.setColor(2,RGB_BLUE);
      mRgb.show();
      delay(500);
      Onboard_RGB(255,0,255);
      mRgb.setColor(1,RGB_PURPLE);
      mRgb.setColor(2,RGB_PURPLE);
      mRgb.show();
      delay(500);
      Onboard_RGB(255,0,255);
      mRgb.setColor(1,RGB_PURPLE);
      mRgb.setColor(2,RGB_PURPLE);
      mRgb.show();
      delay(500);
      Onboard_RGB(255,255,0);
      mRgb.setColor(1,RGB_YELLOW);
      mRgb.setColor(2,RGB_YELLOW);
      mRgb.show();
      delay(500);
```

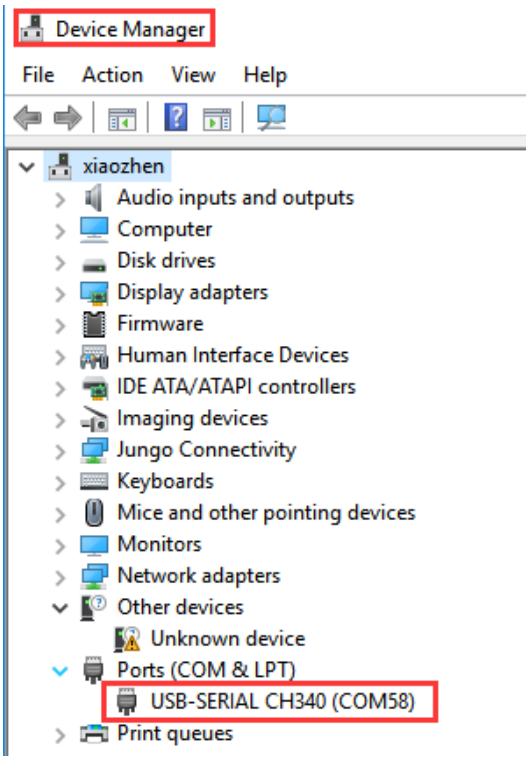
```
    Onboard_RGB(0,0, 0);
    mRgb.setColor(1,RGB_OFF);
    mRgb.setColor(2,RGB_OFF);
    mRgb.show();
    delay(500);
}
```

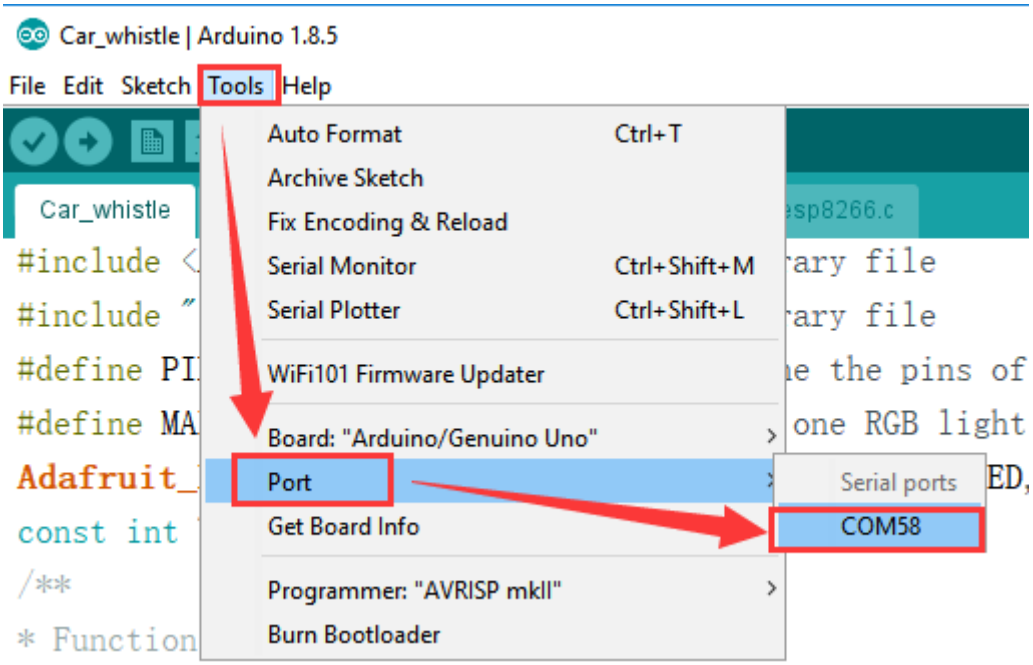
5. Compiling and downloading code

5.1 We need to open the **Key_control_RGB.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.

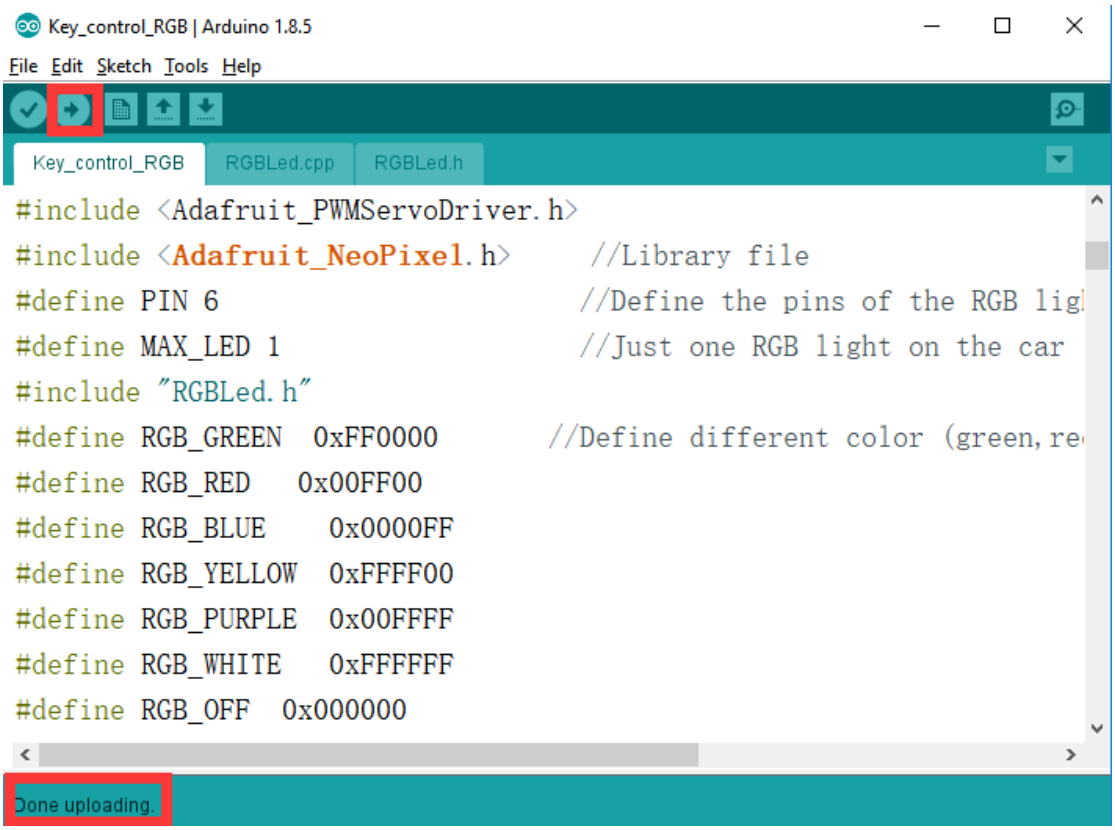


5.2 In the menu bar of Arduino IDE, we need to select 【Tools】---【Port】---selecting the port that the serial number displayed by the device manager just now, as shown in the figure below.





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 press K1 on the expansion board, we can see that On board RGB and Ultrasonic RGB will change color: White --> Red --> Green --> Blue --> Purple --> Yellow --> OFF. Time interval is 500 ms.