

Course 27 ----Dual axis XYrocker

The purpose of the experiment:

In this course we mainly study the use of RGB module.

Introduction of Dual axis XY rocker module:

The actual object is shown below.

**List of components required for the experiment:**

Arduino UNO board *1

USB cable *1

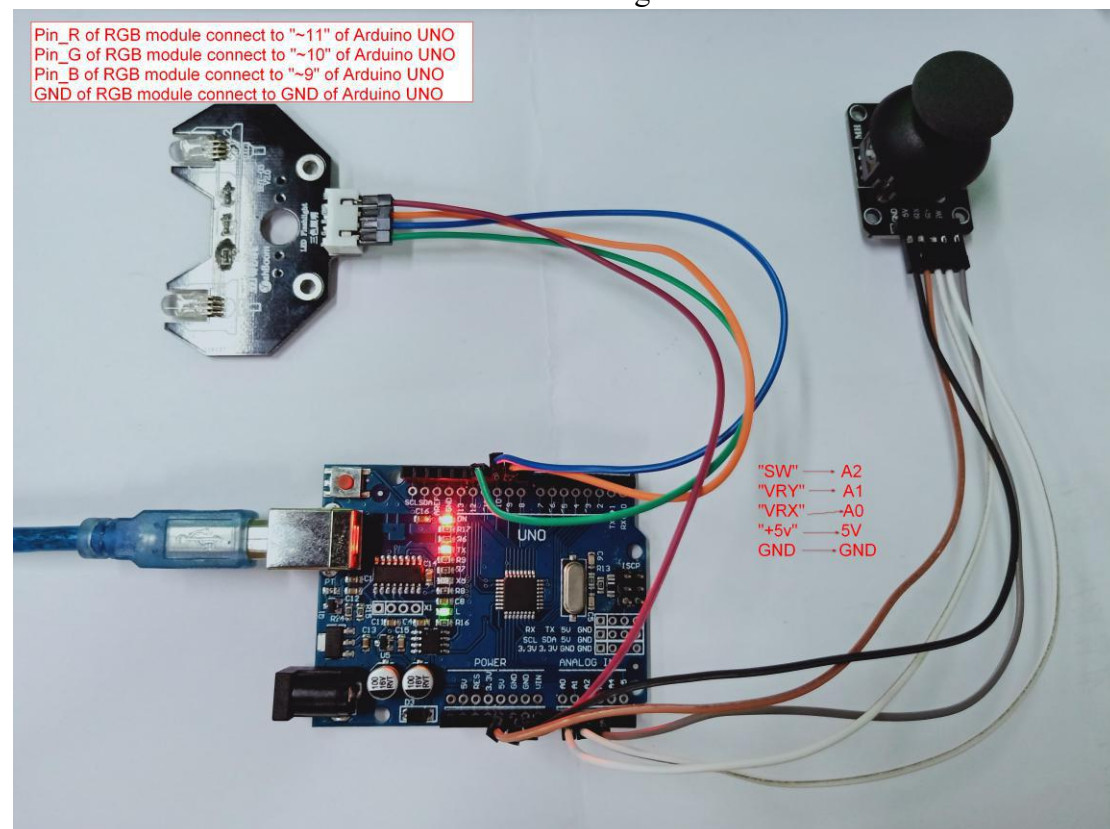
Dual axis XY rocker module *1

RGB*1

Dupont line *1 bunch

Actual object connection diagram:

We need to connect the circuit as shown in the figure below.



Experimental code analysis:

```
int VRX = A0; //The rocker X axis is connected to the analog port A0 of Arduino
UNO
int VRY = A1; //The rocker Y axis is connected to the analog port A1 of Arduino
UNO
int SW = A2; //The rocker button is connected to the analog port A2 of Arduino
UNO
int red = 11; //The red pin of 3 color RGB is connected to the port 11 of Arduino
UNO
int green = 10; //The green pin of 3 color RGB is connected to the port 10 of
Arduino UNO
int blue = 9; //The blue pin of 3 color RGB is connected to the port 9 of Arduino
UNO
int x = 0, y = 0, z;
int val_red = 0, val_green = 0, val_blue = 0;
void setup() //The rocker is set as input port and the RGB lamp is output port
{
    pinMode(VRX, INPUT);
    pinMode(VRY, INPUT);
    pinMode(SW, INPUT);
    pinMode(red, OUTPUT);
    pinMode(green, OUTPUT);
    pinMode(blue, OUTPUT);
    Serial.begin(9600);
}

void loop()
{
    x = analogRead(VRX); //Read the analog voltage value of A0 and assign it to x
    y = analogRead(VRY); //Read the analog voltage value of A1 and assign it to y
    z = analogRead(SW); //Read the analog voltage value of A2 and assign it to z
    Serial.print("x: ");
    Serial.println(x);
    Serial.print("y: ");
    Serial.println(y);
    Serial.print("z: ");
    Serial.println(z);
    if (z == 0) //When the rocker is pressed,white light is on
    {
        val_red = 255;
        val_green = 255;
        val_blue = 255;
    }
    else
```

```

{
  if (y < 5)
  {
    val_red = (1364 - x) * 255 / 1364;
    val_green = x * 255 / 1364;
    val_blue = 0;
  }
  else if (x > 1020)
  {
    if (y < 341)
    {
      val_red = (341 - y) * 255 / 1364;
      val_green = (y + 1023) * 255 / 1364;
      val_blue = 0;
    }
    else
    {
      val_red = 0;
      val_green = (682 + (1023 - y)) * 255 / 1364;
      val_blue = (y - 341) * 255 / 1364;
    }
  }
  else if (y > 1020)
  {
    if (x > 341)
    {
      val_red = 0;
      val_green = (x - 341) * 255 / 1364;
      val_blue = (682 + (1023 - x)) * 255 / 1364;
    }
    else
    {
      val_red = (341 - x) * 255 / 1364;
      val_green = 0;
      val_blue = (1023 + x) * 255 / 1364;
    }
  }
  else if (x < 5)
  {
    val_red = (682 + (1023 - y)) * 255 / 1364;
    val_green = 0;
    val_blue = y * 255 / 1364;
  }
  else

```

```

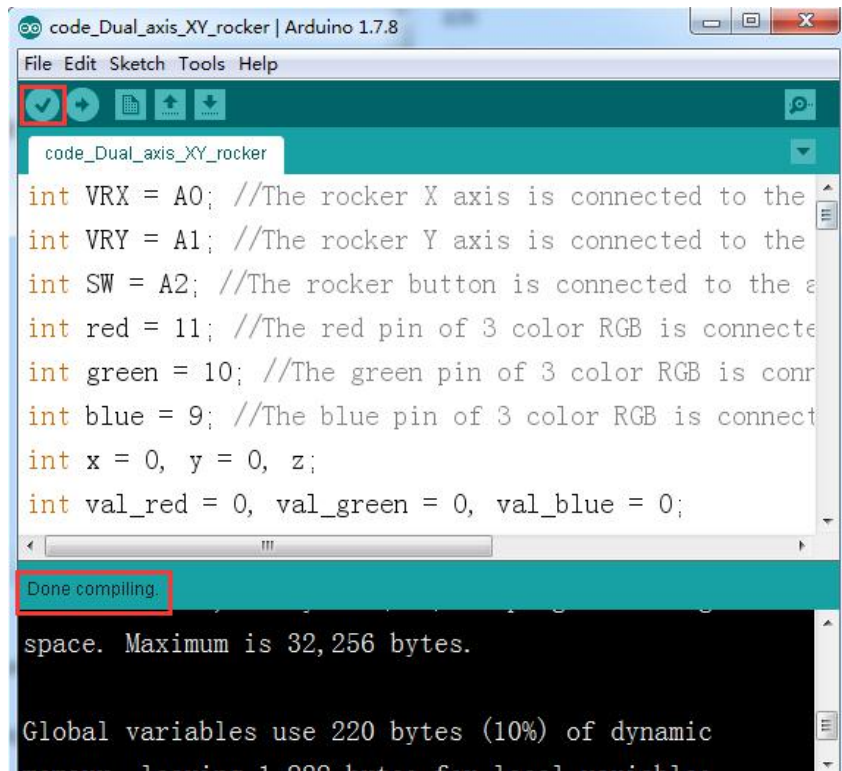
    {
        val_red = 0;
        val_green = 0;
        val_blue = 0;
    }
}
analogWrite(red, val_red);
analogWrite(green, val_green);
analogWrite(blue, val_blue);
}

```

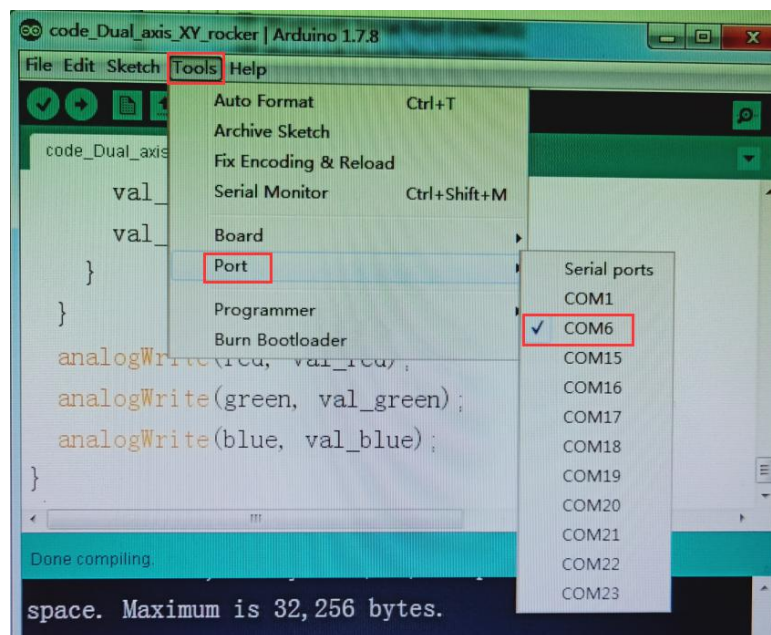
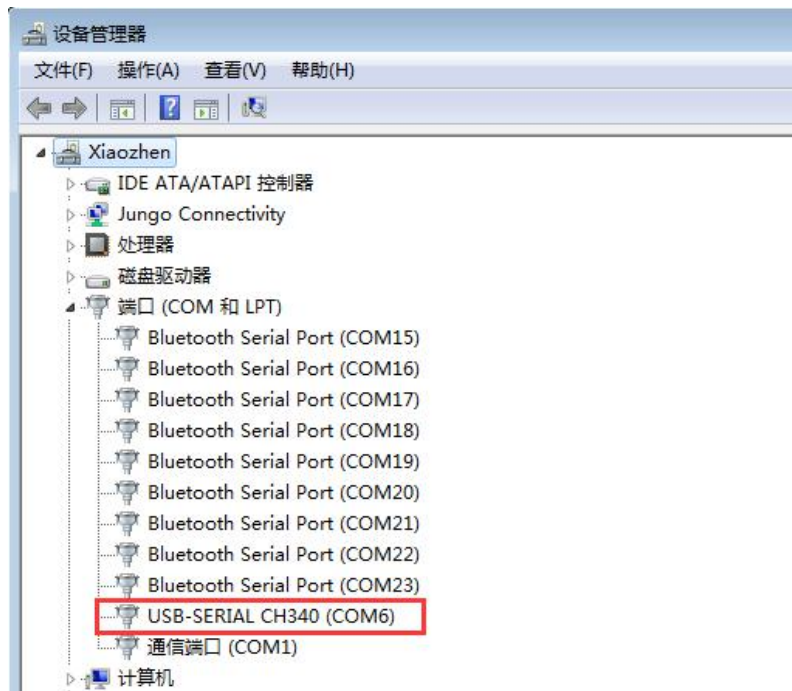
Experimental steps:

1. We need to open the program for this experiment:

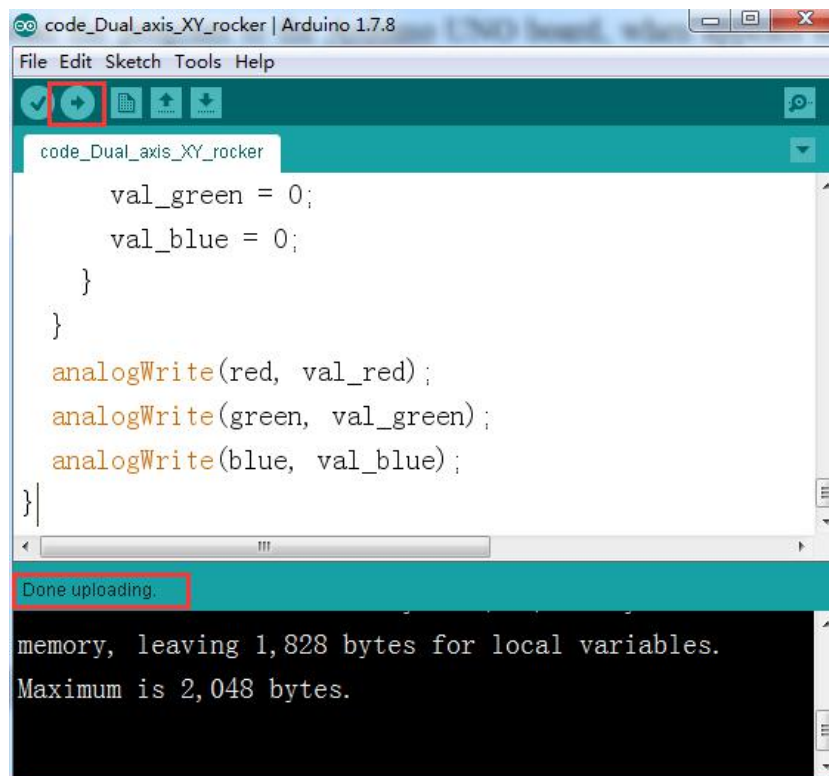
code_Dual_axis_XY_rocker.ino, click “✓” under the menu bar, compile the program, and wait for the words of **Done compiling** in the lower left corner, as shown in the following figure.



2. In the menu bar of Arduino IDE, you need to select the **【Tools】**---**【Port】**--- select the port that the serial number displayed by the device manager just now, for example: COM6, as shown in the following figure.



3. After the selection is completed, you need to click “→” under the menu bar, and upload the program to the Arduino UNO board, when appears to **Done uploading** on the lower left corner , that means that the program has been successfully uploaded to the Arduino UNO board, as shown in the following figure.



```
code_Dual_axis_XY_rocker | Arduino 1.7.8
File Edit Sketch Tools Help
code_Dual_axis_XY_rocker
    val_green = 0;
    val_blue = 0;
  }
}
analogWrite(red, val_red);
analogWrite(green, val_green);
analogWrite(blue, val_blue);
}

Done uploading.
memory, leaving 1,828 bytes for local variables.
Maximum is 2,048 bytes.
```

4. After the program upload is completed, We can control the RGB to emit different colors of light by swinging the rocker back,forth, left, right. As shown in the following figure.

