# Color Following

## 4.1 Introduction to Playing

This course mainly adds the function of PID control of the car's movement based on color tracking. After turning on the start switch, the car will adjust its position to keep the color object in the middle of the screen as much as possible.

## 4.2 Core Content Analysis

The simplePID.py file is a simple PID controller, and the incremental function is an incremental PID algorithm function.

```
def incremental(self, current_value, limit=0):
self.err = self.setPoint - current_value
result = self.last_result + self.Kp * (self.err - self.err_next) + self.Ki *
self.err + self.Kd * (self.err - 2 * self.err_next + self.err_last)
self.err_last = self.err_next
self.err_next = self.err
if limit > 0:
if result > limit:
result = limit
if result < -limit:
result = -limit
self.last_result = result
return result
```

Import and configure PID parameters. Since the image pixel is 1920*1080, the midpoint values are 960 and 540 as the target values. The PID parameter values can be adjusted according to the robot's motion effect.

```
from simplePID import PID

PID_X_Kp = 0.8
PID_X_Ki = 0
PID_X_Kd = 0.2

PID_Y_Kp = 0.8
PID_Y_Ki = 0
PID_Y_Kd = 0.2

PID_X = PID(160, PID_X_Kp, PID_X_Ki, PID_X_Kd)
PID_Y = PID(120, PID_Y_Kp, PID_Y_Ki, PID_Y_Kd)
```

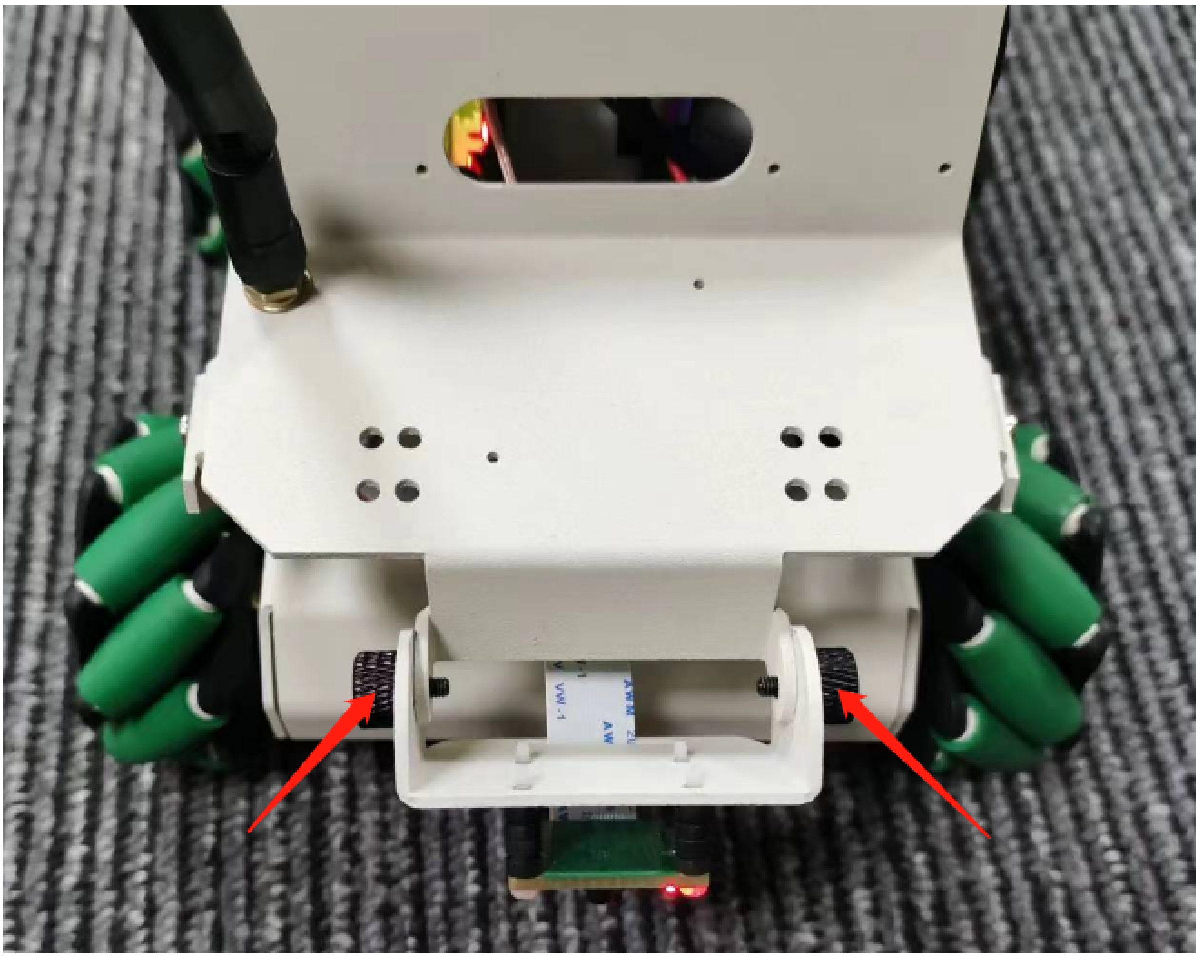
When the start button is clicked, the car_control function is called to control the movement of the car. In order to keep the color object in the middle of the screen, the car moves with the color object.

```
from SunriseRobotLib import SunriseRobot
g_car = SunriseRobot()

def car_control(x, y, radius):
    global last_x, last_y, last_radius
    speed_x = PID_X.incremental(y)/300.0
    speed_y = PID_Y.incremental(x)/300.0
    speed_z = 0
    '''
    if -0.1 < speed_x < 0.1:
        speed_x = 0
    if -0.1 < speed_y < 0.1:
        speed_y = 0
    if speed_x > 0.5:
        speed_x = 0.5
    if speed_x < -0.5:
        speed_x = -0.5
    if speed_y > 0.5:
        speed_y = 0.5
    if speed_y < -0.5:
        speed_y = -0.5
    print("speed:", speed_x, speed_y, speed_z)
    print("x:", x, "y:", y)
    '''
    g_car.set_car_motion(speed_x, speed_y, speed_z)
    last_x, last_y, last_radius = x, y, radius
```

In order for the robot to better follow the color object, the angle of the camera bracket needs to be adjusted.

First run all cells. Without starting the robot, loosen the screws on the left and right sides of the CSI camera bracket, then adjust the angle of the camera bracket, observe the image captured by the camera, and tighten the screws in the appropriate position to fix the angle of the camera bracket.

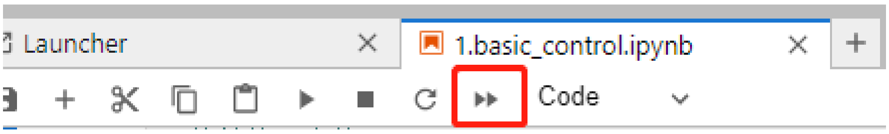


## 4.3 How to play

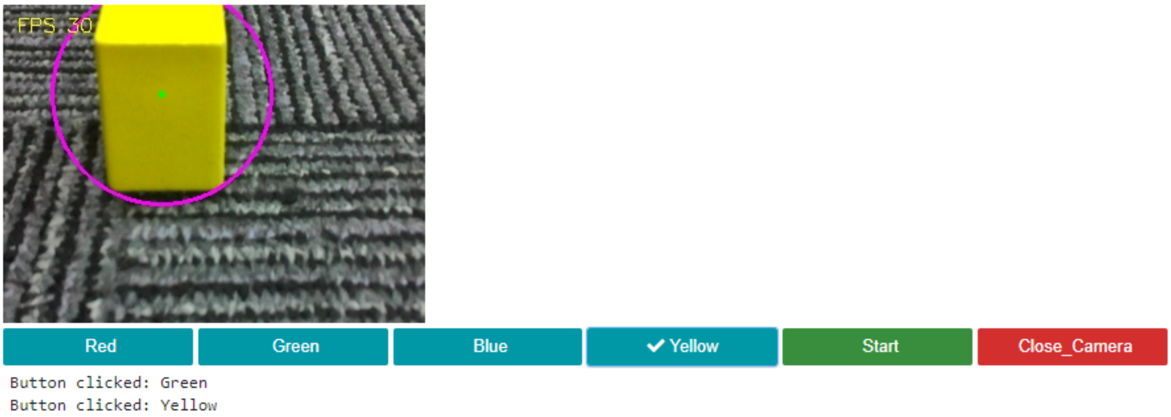
Open the jupyterLab client and find the code path:

```
/home/sunrise/sunriseRobot/Samples/2_AI/05_color_following/color_following.ipynb
```

Click to run all cells, then pull to the bottom to see the generated controls.



The camera will track the red object. If you need to track other objects, click the button below to switch. Click the Start button and the car will start moving. Keep the color object in the center of the camera screen. At this time, move the color object and the car will move with it.



```
Button clicked: Green
Button clicked: Yellow
```

Finally, click the Close_Camera button to close the camera.