04. Servo control (PTZ control)

04. Servo control (PTZ control)

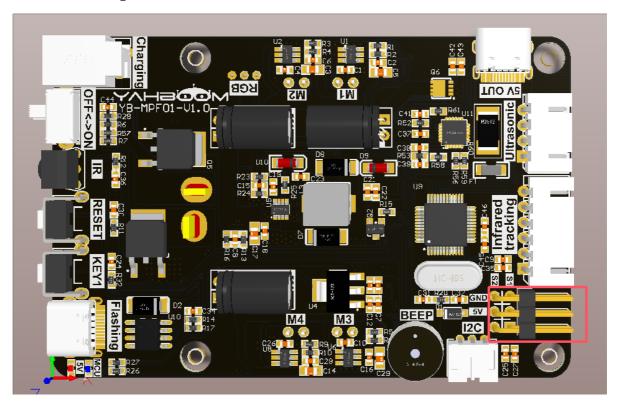
- 1. Learning objectives
- 2. Experimental preparation
- 3. Core code analysis
- 4. Experimental phenomenon

1. Learning objectives

Control the servo connected to the expansion board.

2. Experimental preparation

As shown in the figure below, connect the servo to the interface.



3. Core code analysis

Raspbot_Lib library functions needed to control the servo:

Ctrl_Servo(id, angle):

Parameter explanation: Control connected to expansion board S1, S2.

id=1: Control S1 servo, id=2: Control S1 servo.

angle=[0,180], control the rotation angle of the servo

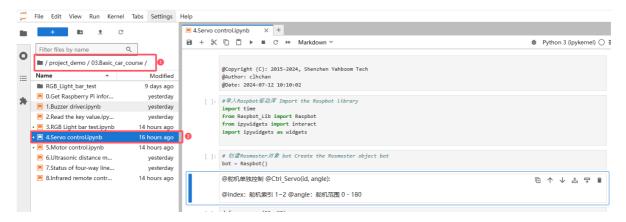
Return value: None.

Source code path: project_demo\03.Basic_car_course

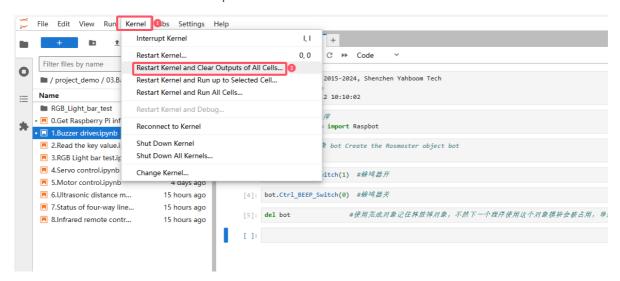
4. Experimental phenomenon

Turn on the robot, open the computer browser to enter the Jupyter lab editor

Enter the source code path and double-click the code to be run



Restart the kernel and clear all outputs



Click the first code block, then click the run button to start running one by one

```
4.Servo control.ipynb
                     × +
1 + % □ □ 1 • C → Code

    ₱ Python 3 (ipykernel) ○ ■

          @Copyright (C): 2015-2024, Shenzhen Yahboom Tech
          @Author: clhchan
          @Date: 2024-07-12 10:10:02
     []: #导入Raspbot驱动库 Import the Raspbot library
                                                                                                      □↑↓占♀ⅰ
          from Raspbot_Lib import Raspbot
          from ipywidgets import interact
          import ipywidgets as widgets
     []: # 创建Rosmaster对象 bot Create the Rosmaster object bot
          bot = Raspbot()
          @舵机单独控制 @Ctrl Servo(id, angle):
          @index: 舵机索引 1~2 @angle: 舵机范围 0 - 180
     ef pwm_servo(S1, S2):
              bot.Ctrl_Servo(1, S1)
             bot.Ctrl_Servo(2, S2)
```

After the program runs, as the code block runs, we can use the slider to control the angle of the servo