

6.Robot rotate left and right

6.Robot rotate left and right

- 1.Experimental purpose
- 2.Experimental preparation
- 3.Experimental process
- 4.Experiment summary

1.Experimental purpose

This course mainly learns how to control the left and right rotation function of the robot.

2.Experimental preparation

The functions of the Muto hexapod robot Python library involved in this course are:

turnleft(step): Left rotation, step is the step width, the larger the effective value, the larger the width of each step. The value range of step is [10, 25].

turnright(step): Right rotation, step is the step width, the larger the effective value, the larger the width of each step. The value range of step is [10, 25].

3.Experimental process

Open the jupyterLab client and find the code path:

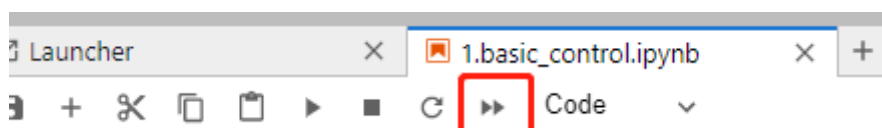
```
muto/Samples/Control/6.turn_left_right.ipynb
```

By default `g_ENABLE_CHINESE=False`, if you need to display Chinese, please set `g_ENABLE_CHINESE=True`.

```
# 中文开关, 默认为英文 Chinese switch. The default value is English
g_ENABLE_CHINESE = False

Name_widgets = {
    'Stop': ("Stop", "停止"),
    'TurnLeft': ("TurnLeft", "向左转"),
    'TurnRight': ("TurnRight", "向右转"),
    'End': ("End", "结束")
}
```

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



Click different buttons to correspond to different functions.

Step: 25
step: 25
Speed: 3

TurnLeft TurnRight Stop End

Every time a button is clicked, the corresponding function will be executed. Key event processing.

```
# 按键按下事件处理    Key press event processing
def on_button_clicked(b):
    ALL_Uncheck()
    b.icon = 'check'
    with output:
        print("Button clicked:", b.description)
    if b.description == Name_widgets['Stop'][g_ENABLE_CHINESE]:
        g_bot.stop()
        b.icon = 'uncheck'
    elif b.description == Name_widgets['TurnLeft'][g_ENABLE_CHINESE]:
        g_bot.turnleft(g_step)
    elif b.description == Name_widgets['TurnRight'][g_ENABLE_CHINESE]:
        g_bot.turnright(g_step)
```

The variable `g_step` is controlled by the Step slider. When the value of the slider changes, it will detect which button has been pressed, and the value of `g_step` will be updated immediately to the robot.

```
def on_slider_step(value):
    global g_step
    g_step = value
    if button_turn_left.icon == 'check':
        g_bot.turnleft(g_step)
    elif button_turn_right.icon == 'check':
        g_bot.turnright(g_step)
    print("step:", value)
```

The robot's movement speed is controlled by the Speed slider. When the slider value changes, the robot's movement speed is immediately updated.

```
def on_slider_speed(value):
    global g_bot
    g_bot.speed(value)
    print("speed:", value)
```

4.Experiment summary

This time, JupyterLab control is used to control the left and right rotation, speed control and other functions of the six-legged robot. For example, set Step to 20, Speed to 2, and then click the TurnLeft button, the six-legged robot will rotate to the left. At this time, the TurnLeft button will be ticked, and the command to be pressed will be prompted below the control. At this time, if you change the value of the Step or Speed slider, you can see the robot's step width and speed change.

When you need to stop, please click the Stop button to stop.

Step:

25

step: 25

Speed:

3

TurnLeft

TurnRight

Stop

End

Press the End button to exit the program.