

8.PC software control

8.PC software control

- 8.1 Experimental goals
- 8.2 Experiment preparation
- 8.3 Experiment procedure
- 8.4 Experiment summary

8.1 Experimental goals

This course mainly learns how to generate host computer controls through the JupyterLab web page to control the Muto robot.

8.2 Experiment preparation

Since the host computer control requires the use of the camera, in order to avoid conflicts, please close the APP control program or other programs that occupy the camera before running this program.

8.3 Experiment procedure

Open the jupyterLab client and find the code path:

```
muto/Samples/Control/8.Robot_Control.ipynb
```

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

```
g_ENABLE_CHINESE = True
Name_widgets = {
    'Stop': ("Stop", "停止"),
    'Forward': ("Forward", "前进"),
    'Backward': ("Backward", "后退"),
    'Left': ("Left", "左平移"),
    'Right': ("Right", "右平移"),
    'TurnLeft': ("TurnLeft", "向左转"),
    'TurnRight': ("TurnRight", "向右转"),
    "Step": ("Step", "步伐宽度"),
    'Reset': ("Reset", "恢复初始姿态"),
    'Stretch': ("Stretch", "伸懒腰"),
    'Greeting': ("Greeting", "打招呼"),
    'Retreat': ("Retreat", "害怕退缩"),
    'warm_up': ("warm_up", "热身蹲起"),
    'Turn_around': ("Turn_around", "原地转圈"),
    'Say_no': ("Say_no", "挥手说不"),
    'Crouching': ("Crouching", "寄居蜷起"),
```

```

'Stride': ("Stride", "大步向前"),
'Close_Camera': ("Close_Camera", "关闭摄像头")
}

```

Open the camera. The default configuration device is /dev/video0, the display resolution is 640*480, and the frame rate is 30 frames. If the device number of the camera in the system is not /dev/video0, please modify the configuration information according to the actual device number.

```

image_widget = widgets.Image(format='jpeg', width=640, height=480)
image = cv2.VideoCapture(0)
image.set(3, 640)
image.set(4, 480)
image.set(5, 30)

```

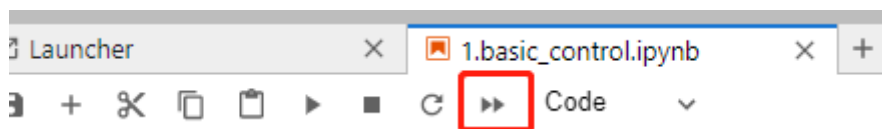
The camera display screen function reads the camera screen and transmits the image to the image display control image_widget for display.

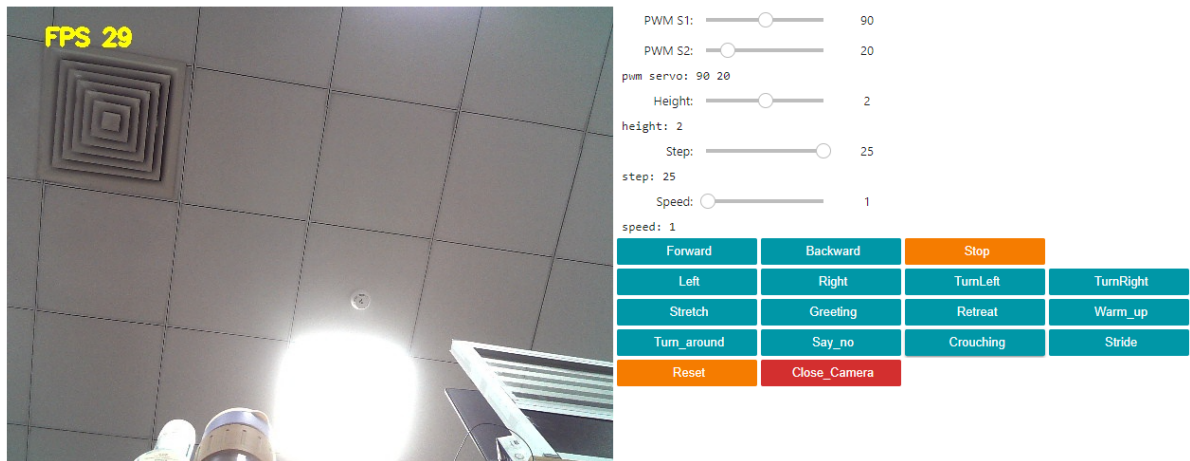
```

def Display_Camera():
    global g_stop, g_bot
    global g_height, g_width, g_roll, g_pitch, g_yaw
    t_start = time.time()
    fps = 0
    while not g_stop:
        ret, frame = image.read()
        fps = fps + 1
        mfps = fps / (time.time() - t_start)
        try:
            cv2.putText(frame, "FPS " + str(int(mfps)), (40,40),
cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0,255,255), 2)
            image_widget.value = bgr8_to_jpeg(frame)
        except:
            pass
    image.release()
    g_bot.reset()
    time.sleep(.1)
del g_bot

```

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





On the left is the camera display, and on the right are the controls for controlling the Muto robot. The function is the same as the previous control content, so no further explanation will be given here.

Close_Camera

The red button at the bottom is for closing the camera process. When you want to end the program, please click to close the camera. Otherwise, the camera may be occupied and other programs cannot use the camera.

8.4 Experiment summary

This time, the JupyterLab control is used to control the movement and actions of the Muto robot. You can control the movement of the Muto robot or the camera pan/tilt to move the camera.

You need to click the Turn off camera button when ending the program, otherwise other programs using the camera will report an error.