11.Host computer control

1.Experimental Purpose

This course mainly studies how to generate upper computer controls through JypyterLab web pages to control a hexapod robot.

2.Experimental Preparation

Due to the need for a camera for upper computer control, to avoid conflicts, please close the APP control program or other programs that occupy the camera before running this program.

3.Experimental process

Open the jupyterLab client and find the code path:

```
muto/Samples/Control/11.Robot_Control.ipynb
```

By default, g_ ENABLE_ CHINESE=False, if Chinese needs to be displayed, 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")
}
```

Turn on the camera, and the default configuration device is/dev/video0.

The display resolution is 640 * 480, and the frame rate is 30 frames.

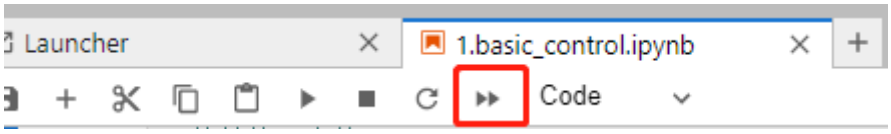
If the device number of the camera in the system is not/dev/video0, please modify the configuration information based on the actual device number.

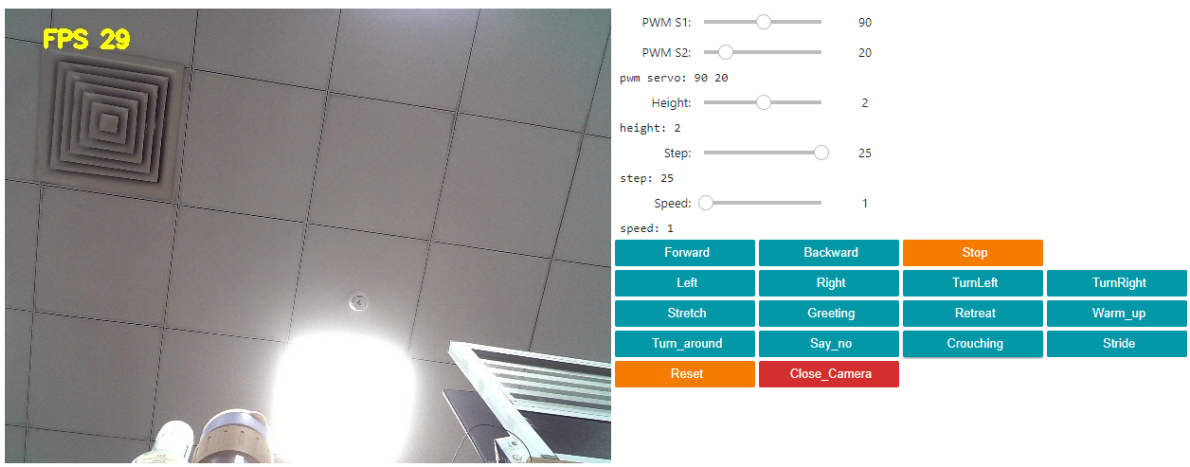
```
image_widget = widgets.Image(format='jpeg', width=640, height=480)
image = cv2.VideoCapture(0)
image.set(3, 640)
image.set(4, 480)
image.set(5, 30)
```

Camera display screen function, reads the camera's screen, and transfers the image to the image display control image_ Widget to 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, then drag to the bottom to see the generated controls.





On the left is the camera display screen, and on the right is the control for the hexapod robot.

The red button at the bottom closes the camera process. To end the program, please click to close the camera, otherwise it may continue to occupy the camera and cause other programs to be unable to use the camera.

8.4 Experimental Summary

This time, JupyterLab controls are used to control the movement and actions of the hexapod robot. You can control the movement of the hexapod robot or the camera pan tilt to make the camera move the image.
When ending the program, you need to click the 'Close Camera' button, otherwise other programs using the camera will report an error.