

第 11 课 驱动直流电机

本章节主要实现扩展板上的驱动直流电机以不同速度进行转动。

1. 实现原理

通过程序对电机的参数进行设置，当数值为正时前进，数值为负时则后退。该程序的源代码位于：`/home/pi/board_demo/motor_control_demo.py`

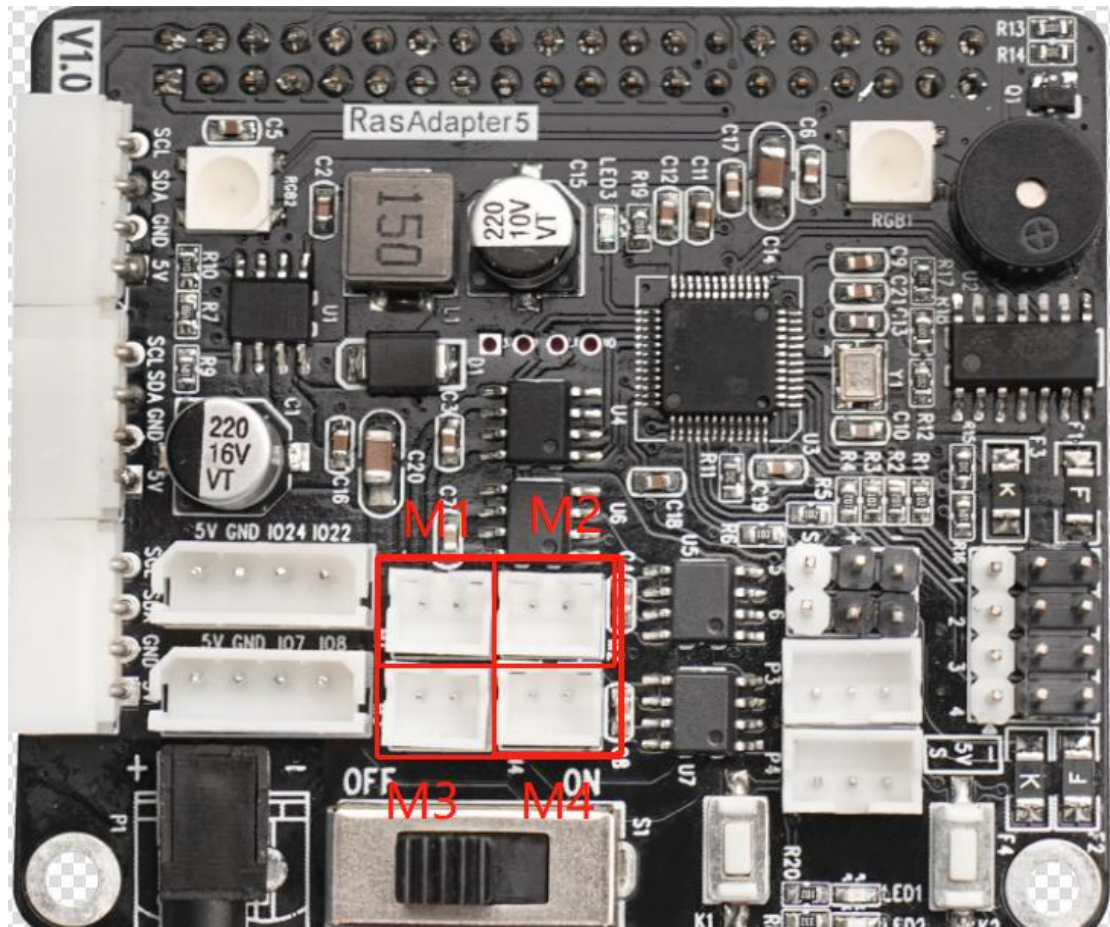
```
24 board = rrc.Board()
25
26 start = True
27 #关闭前处理
28 def Stop(signum, frame):
29     global start
30
31     start = False
32     print('关闭中...')
33     board.set_motor_duty([[1, 0], [2, 0], [3, 0], [4, 0]]) # 关闭所有电机
34
35 signal.signal(signal.SIGINT, Stop)
36
37 if __name__ == '__main__':
38
39     while True:
40         board.set_motor_duty([[1, 35]]) #设置1号电机速度35
41         time.sleep(0.2)
42         board.set_motor_duty([[1, 90]]) #设置1号电机速度90
43         time.sleep(0.2)
44
45         if not start:
46             board.set_motor_duty([[1, 0], [2, 0], [3, 0], [4, 0]]) # 关闭所有电机
47             print('已关闭')
48             break
```

底盘电机的控制主要引用 Board 库下的 `set_motor_duty()` 函数。以代码“`board.set_motor_duty([[1, 35]])`”为例，其中：

`[[1, 35]]` 设置电机的占空比，嵌套了二元组，二元组包含电机编号1和占空比之35，即速度为35。

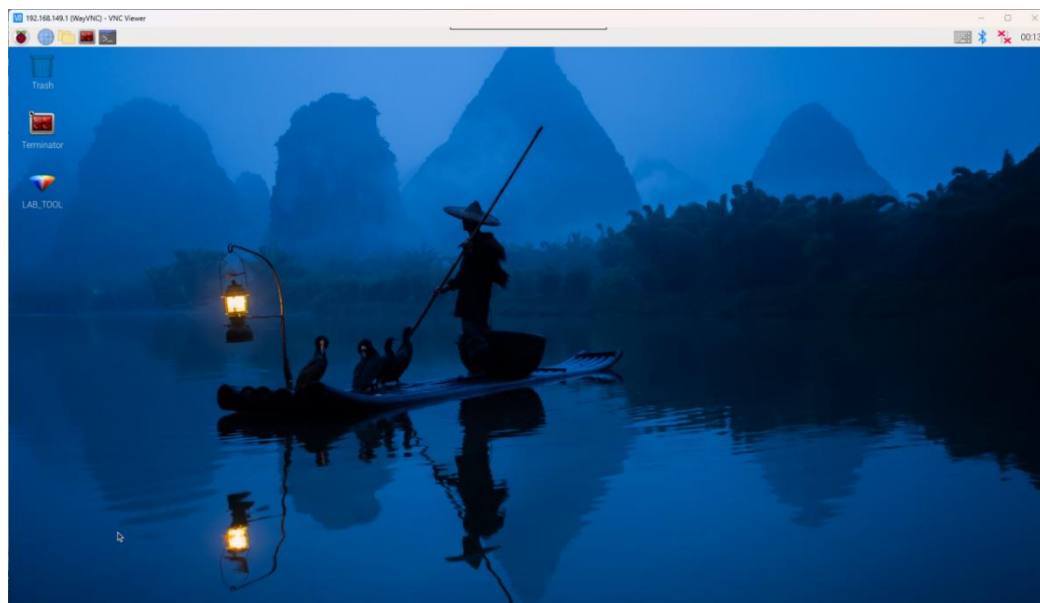
2. 准备工作


树莓派扩展板上板载了 4 个电机接口（如下图），我们接 1 号电机接口进行测试

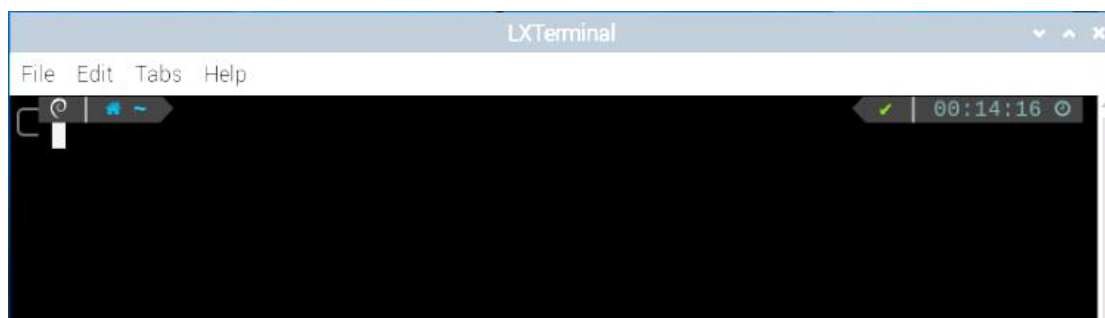


3. 实验步骤

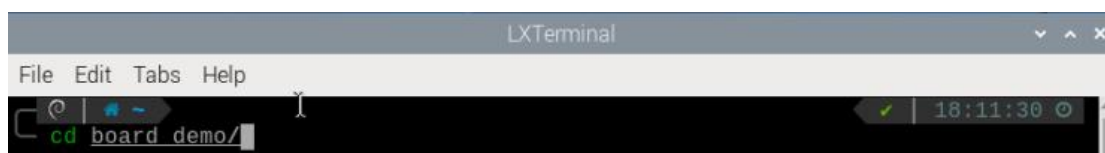
- 1) 将设备开机，将机器人通过 VNC 远程连接工具进行连接。



- 2) 点击桌面左上角的图标，或按下快捷键“Ctrl+Alt+T”打开命令行终端。



- 3) 输入玩法程序所在目录的指令“cd board_demo/”，按下回车键。



- 1) 输入控制电机转动的指令“python3 motor_control_demo.py”，按下回车。



- 2) 如需关闭此程序，可按下“Ctrl+c”。若关闭失败，请反复尝试。

3. 功能实现

玩法开启后，直流电机 M1 先以 35 的速度正转 0.2 秒，然后以 90 的速度正转 0.2 秒，以此不断循环转动。

4. 功能延伸

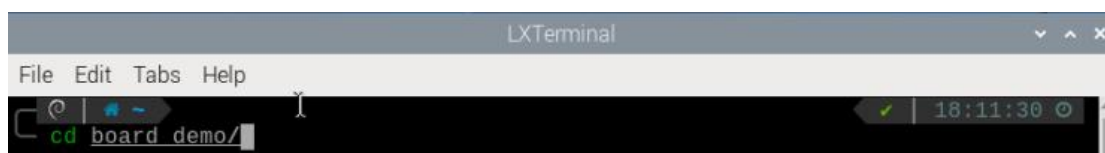
4.1 速度和方向调节

注意：电机速度范围为-100~100，请勿设置超出范围的数值，以免对电机造成损坏。

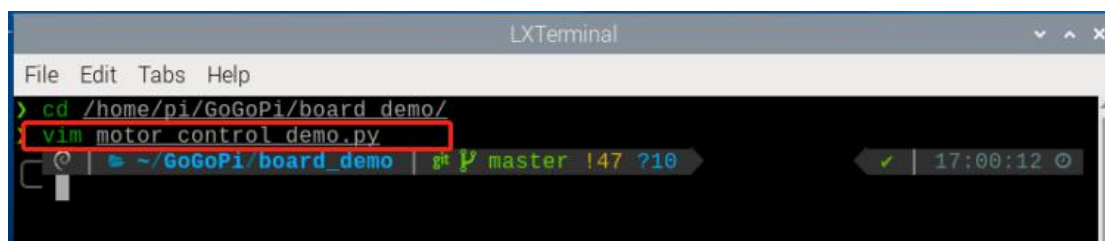
若想修改电机转动的速度和方向，可参照以下步骤进行。

本节以控制 1 号接口电机先以 80 的速度正转 2 秒，再以 60 的速度反转 2 秒，为例进行修改。

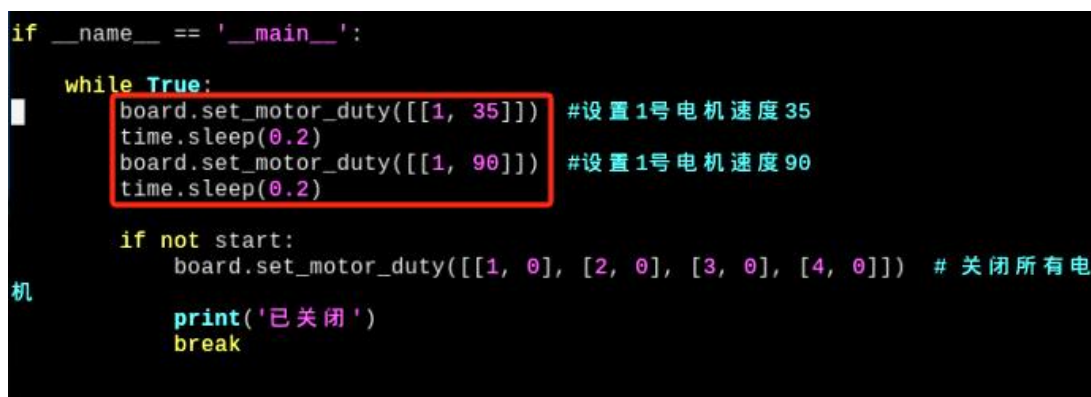
- 1) 输入进入玩法程序所在目录的指令“`cd /home/pi/board_demo/`”，并按下回车。



- 2) 输入打开程序文件的指令“`vim motor_control_demo.py`”，并按下回车。



- 3) 在打开的页面找到如下图框出代码。



- 4) 按一下键盘上的“i”键，进入编辑模式。

```
if __name__ == '__main__':  
    while True:  
        board.set_motor_duty([[1, 35]]) #设置1号电机速度35  
        time.sleep(0.2)  
        board.set_motor_duty([[1, 90]]) #设置1号电机速度90  
        time.sleep(0.2)  
  
        if not start:  
            board.set_motor_duty([[1, 0], [2, 0], [3, 0], [4, 0]]) # 关闭所有电机  
            print('已关闭')  
            break  
  
-- INSERT --
```

5) 如下图框出部分所示进行修改, 需要注意, 当速度值为正时, 电机为正转, 值越大, 电机正转速度越快; 当速度值为负值时, 电机反转, 值越小, 电机反转速度越快。

```
if __name__ == '__main__':  
    while True:  
        board.set_motor_duty([[1, 80]]) #设置1号电机速度35  
        time.sleep(2)  
        board.set_motor_duty([[1, -60]]) #设置1号电机速度90  
        time.sleep(2)  
  
        if not start:  
            board.set_motor_duty([[1, 0], [2, 0], [3, 0], [4, 0]]) # 关闭所有电机  
            print('已关闭')  
            break  
  
-- INSERT --
```

6) 修改完成后, 按下键盘上的“Esc”键, 然后依次输入“:wq” (注意 wq 前为冒号:), 输入完成后按下回车, 即可保存并退出程序。

```
if not start:  
    MotorStop() # 关闭所有电机 Turn off all motors  
    print('已关闭')  
    break  
  
:wq
```

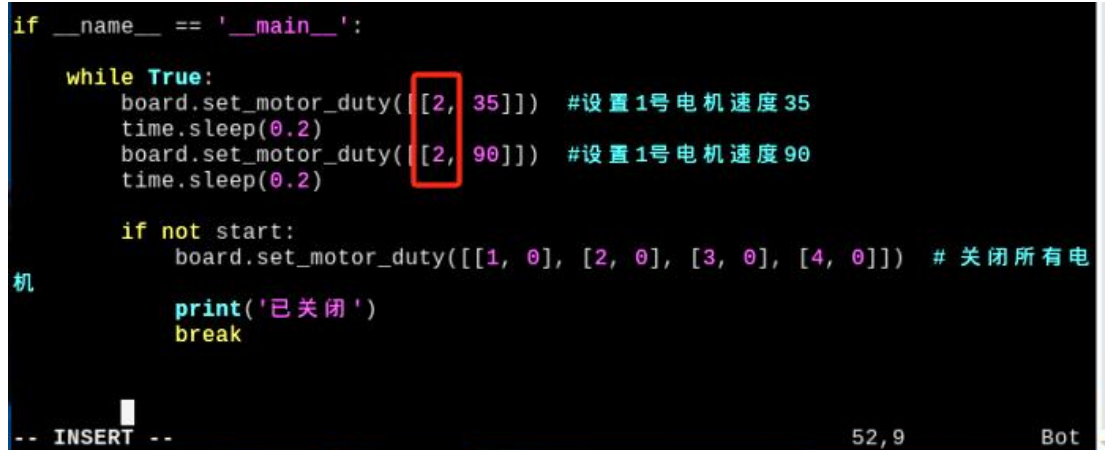
7) 程序保存之后, 按照 [2.实验步骤](#) 内容操作即可查看程序运行效果。

4.2 电机接口调节

1) 程序默认的电机接口为 1 号, 我们这里以修改为 2 号电机为例。参照 [4.1 速度和](#)


[方向调节](#)找到电机参数的代码并进入编辑模式。

2) 将电机 1 号接口修改为 2 号。如下图所示：



```
if __name__ == '__main__':  
    while True:  
        board.set_motor_duty([2, 35]) #设置1号电机速度35  
        time.sleep(0.2)  
        board.set_motor_duty([2, 90]) #设置1号电机速度90  
        time.sleep(0.2)  
  
        if not start:  
            board.set_motor_duty([[1, 0], [2, 0], [3, 0], [4, 0]]) # 关闭所有电机  
            print('已关闭')  
            break  
  
-- INSERT --
```

3) 修改完成后按下键盘上的“Esc”键，然后依次输入“:wq”（注意 wq 前为冒号:），输入完成后回车，即可保存并退出程序。



```
if not start:  
    MotorStop() # 关闭所有电机 Turn off all motors  
    print('已关闭')  
    break  
  
:wq
```

4) 程序保存之后按照 [2.实验步骤](#) 内容操作即可查看程序运行效果。