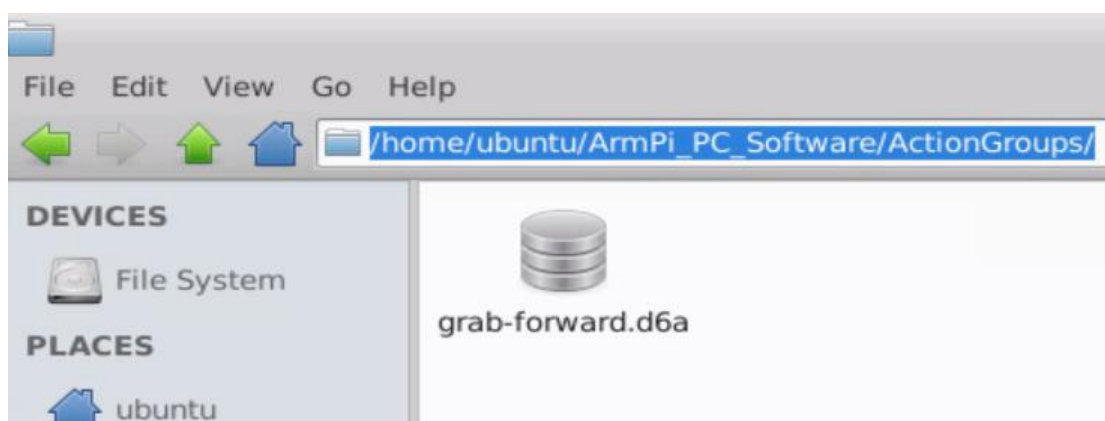


第3课 动作执行教学

本节适用于采用命令行形式启动上位机。

1. 实现目标

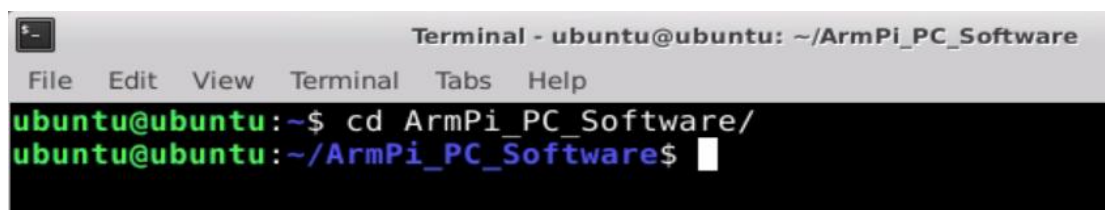
将“第2课 动作编辑教学”所编辑的动作，通过终端命令行的形式进行调用执行。（动作文件必须保存在“/home/ubuntu/ArmPi_PC_Software/ActionGroups/”路径下才能被调用）



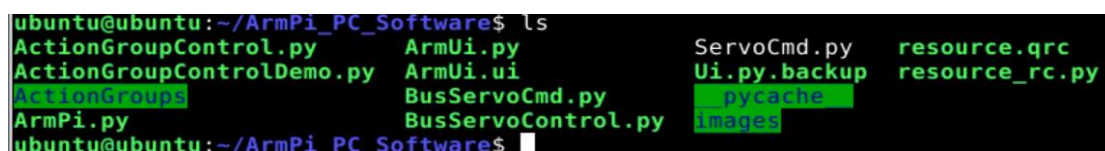
2. 动手实现

2.1 执行方法

1) 单击左下角的“Applications”，输入“`cd ArmPi_PC_Software/`”指令，切换到上位机目录内。



2) 输入“`ls`”指令来查看该目录下的所有文件。



2.1 单个动作执行方法

如需执行单个动作，需要前往调用动作的程序文件中进行修改。

1) 输入指令“`sudo vim ActionGroupControlDemo.py`”，按下回车，通过vi编辑器打开该程序。

2) 在这里我们并未使用Board模块，所以按下“i”键，删除该程序中的“import Board”行。

```
1 import time
2 import Board
3 import ActionGroupControl as AGC
4
```

3) 接下来，可以看到程序中默认执行1号和2号这两个动作。我们需要将“1”和“2”改为执行的动作组名。这里以上一节编辑好的“grab-forward”动作组为例进行，按下“i”键，将“grab-forward”替换括号内的“1”。

```
25 AGC.runAction('grab-forward') #
    入
26 AGC.runAction('2')
-- INSERT --
```

注意：动作名称要与实际存储的名称一致，否则将调用失败！

4) 这里我们执行单个动作，那么将光标移动到第二行前输入“#”注释即可。

```
23
24 # 动作组需要保存在路径/home/ubuntu/Arm
25 AGC.runAction('grab-forward') # 参数为
    入
26 #AGC.runAction('2')
-- INSERT --
```

5) 按下Esc键，然后再输入“:”进入末行命令模式，输入“wq”按下回车，将修改保存并退出。

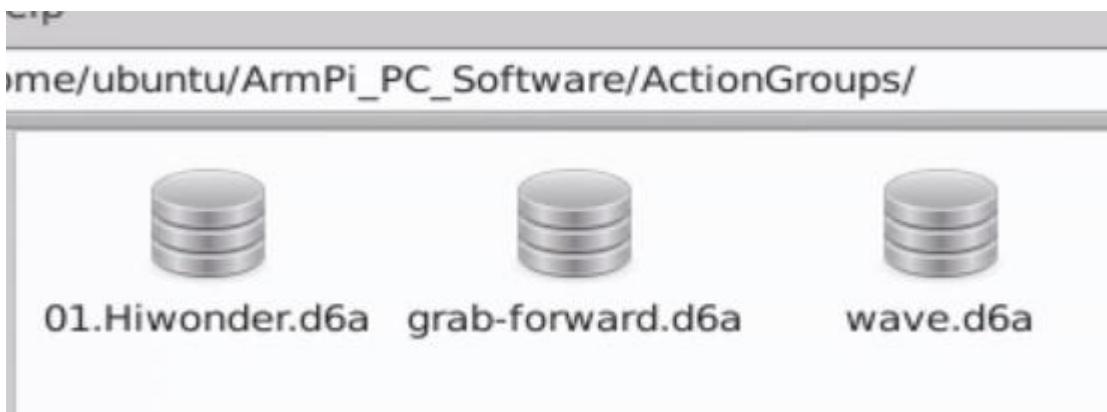
```
24 # 动作组需要保存在路径 /home/ub  
25 AGC.runAction('grab-forward')  
    入  
26 #AGC.runAction('2')  
:wq
```

6) 再次输入指令“**sudo python3 ActionGroupControlDemo.py**”，按下回车，此时机械臂将执行“grab-forward”动作1次。

注意：当路径下不存在动作组文件时，LX终端会显示“未能找到动作组文件”的提示。

2.3 调用更多执行动作

动作编辑及保存教学可参考上一课“第2课 动作编辑教学”内容。这里以已编辑存储好的三个动作为例进行说明：



1) 先按照前面的操作输入“**sudo vim ActionGroupControlDemo.py**”指令进入程序文件。

2) 然后将AGC.runAction(‘’)函数进行复制，函数内参数要与保存的动作文件名称相同，否则无法执行。

3) 比如在执行完上面两个动作文件后，需要执行一下3号动作文件。鼠标选中26行的函数，按“Y”两下，然后按下“P”即可将上面的函数复制下来。

```
25 AGC.runAction('grab-forward') # 参数为动作名称
26 AGC.runAction('01.Hiwonder')
27 AGC.runAction('wave')
28
```

4) 将函数内的参数修改为“**wave**”，保存退出，再次输入指令“**sudo python3 ActionGroupControlDemo.py**”即可。