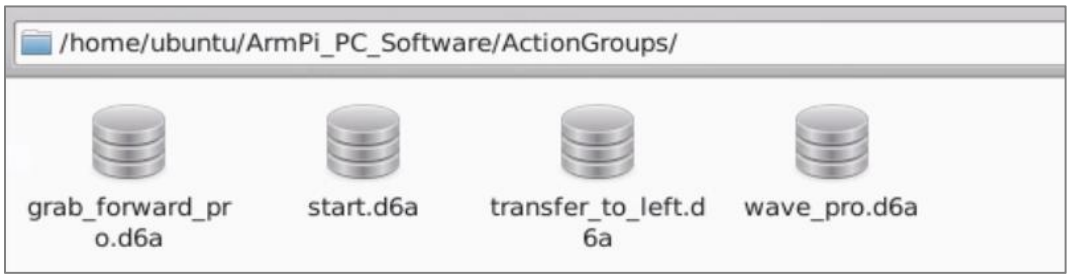
Lesson 2 Call Action Group

1. Realization

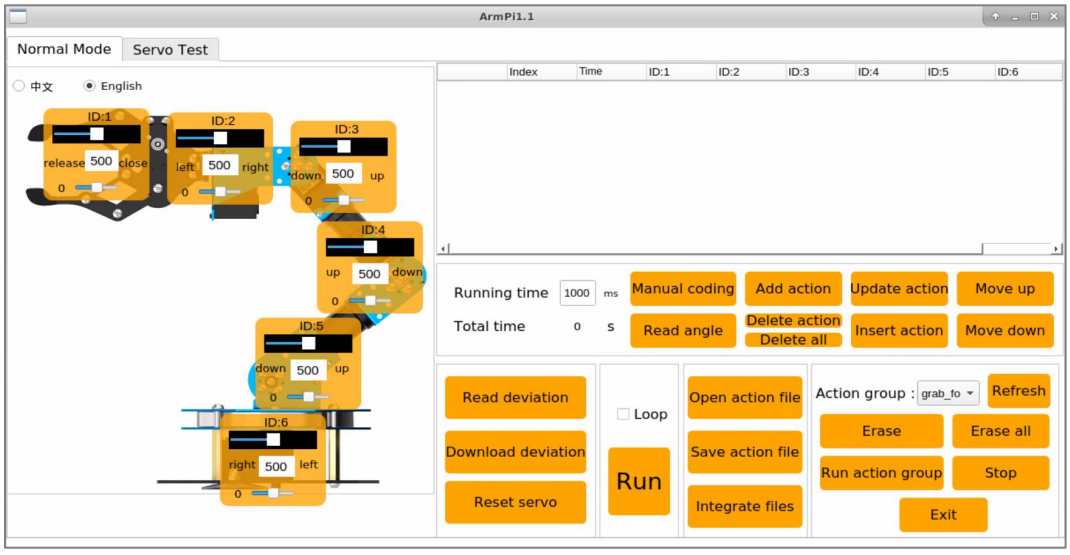
ArmPi Pro has built-in action groups which are saved in "**/home/ubuntu/ArmPi_PC_Software/ActionGroups/**". We can check and call the built-in action group by PC software or command line.



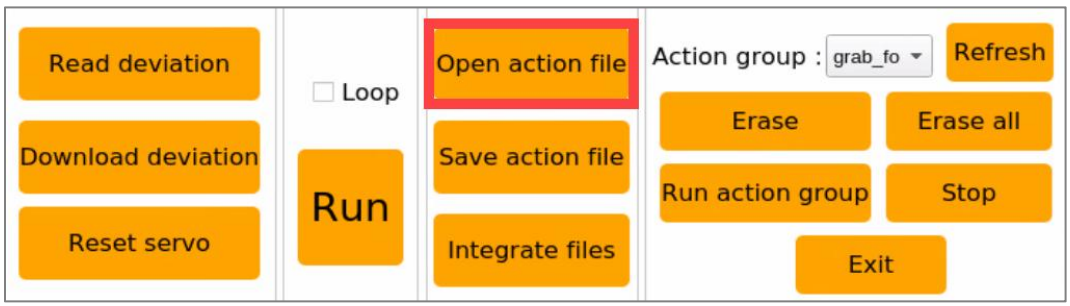
2. Call Action Group

2.1 PC software

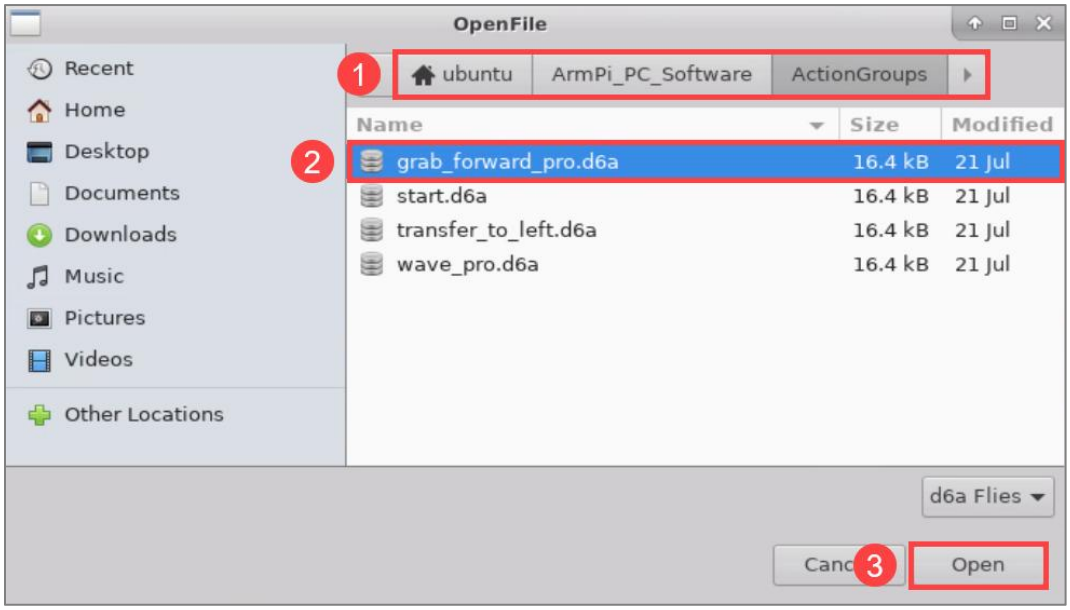
1) Double click  to enter PC software, as the figure shown below:



2) Then click "Open action file".



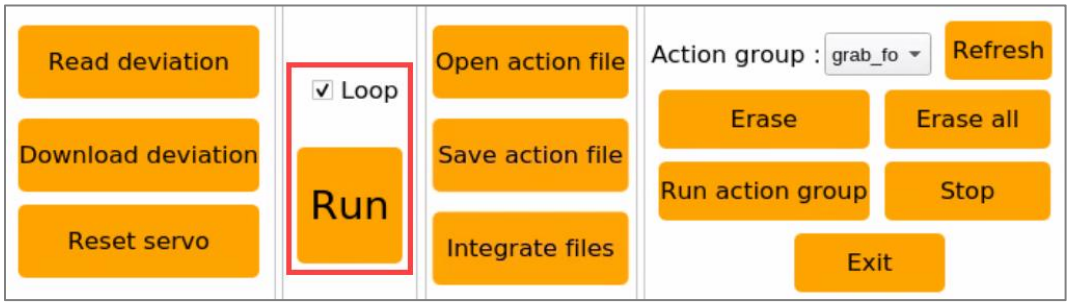
3) Then select the action group to be executed and click “Open”.



4) Now, the running time and servo value of each action will be displayed in action data list.

	Index	Time	ID:1	ID:2	ID:3	ID:4	ID:5	ID:6
	1	800	500	500	300	900	700	500
	2	600	500	500	150	800	450	500
	3	100	500	500	150	800	450	500
	4	200	200	500	150	800	450	500
	5	800	200	500	200	800	290	500
	6	400	600	500	200	800	290	500
	7	800	600	500	155	770	520	500
	8	200	600	500	155	770	520	500
	9	400	600	500	300	770	450	500

5) Click “Run” to run this action group. If want to repeatedly run the action group, you can click “loop”.



2.2 Command line

1) Click 

in the lower left corner and select 

to open terminal.

2) In the opened terminal interface, enter command “cd ArmPi_PC_Software/” to access to the directory of PC software.

```
ubuntu@ubuntu:~$ cd ArmPi_PC_Software/
```

3) Enter command “ls” to check all files under the directory. Action group files are saved in “ActionGroups”, as the figure shown below:

```
ubuntu@ubuntu:~/ArmPi_PC_Software$ ls
ActionGroupControl.py      ArmUi.py          ServoCmd.py    resource.qrc
ActionGroupControlDemo.py  ArmUi.ui          Ui.py.backup   resource_rc.py
ActionGroups               BusServoCmd.py    __pycache__
ArmPi.py                   BusServoControl.py images
```

2.2.1 Execute single action

If want perform single action, you need to modify the program file for calling action.

1) Enter command “**sudo vim ActionGroupControlDemo.py**” and press “Enter” to open the program file for calling action via vi editor.

```
ubuntu@ubuntu:~/ArmPi_PC_Software$ sudo vim ActionGroupControlDemo.py
```

2) Then press “i” to switch to the program editing mode.

```
 25 AGC.runAction('wave') # The parameter is the action group name including
    not suffix passed in as a character
-- INSERT --                                              23,1          Bot
```

3) Next, we can find that the program will perform "**wave_pro**" action by default.

```
25 AGC.runAction( wave ) # The parameter is the action group name including
   not suffix passed in as a character
```

4) Take changing the default action group to "grab-forward_pro" as example. Replace "wave_pro" in parenthesis with "grab-forward_pro".

```
25 AGC.runAction( grab_forward_pro ) # The parameter is the action group
   name including not suffix passed in as a character
```

Note: The action name must be consistent with the file name, otherwise, it will be failed to call!

5) Press "Esc" and enter ":wq". Then press "Enter" to save and exit.

```
25 AGC.runAction('grab_forward_pro') # The parameter is the action group
   name including not suffix passed in as a character
:wq
```

6) Enter command "sudo python3 ActionGroupControlDemo.py" and press "Enter" to execute "grab-forward_Pro" action once.

```
ubuntu@ubuntu:~/ArmPi_PC_Software$ sudo python3 ActionGroupControlDemo.py
```

Note: If no action action file is saved in the path, LX terminal will prompt "no action group file is found." Therefore, please make sure the action group to be called is save in the correct path.

2.2.2 Call several action groups

If want to call several action groups, please refer to the following operation steps. This section will call "wave_Pro" and "grab-forward_Pro" actions as example:

1) According to the operation steps in "2.2.1" Execute single action, enter command "**sudo vim ActionGroupControlDemo.py**" to enter program file and editing mode.

```
ubuntu@ubuntu:~/ArmPi_PC_Software$ sudo vim ActionGroupControlDemo.py
```

```
25 AGC.runAction('wave') # The parameter is the action group name including
   not suffix passed in as a character
-- INSERT --                                              23,1          Bot
```

2) Then copy “AGC.runAction(‘’)” function. Please note that the parameters in function should be consistent with the action group name, otherwise, the action can not be performed .

```
26 AGC.runAction('wave_pro')# 参数为动作组的名称，不包含后缀，
27  AGC.runAction('grab_forward_pro')
-- INSERT --
```

3) There is another tip for you. For example, after performing two actions above, it there are other action files to be performed, you can select the function in 27 line, press “Y” twice and then press “P” to copy the function. (Before this step, you need to press “Esc” to exit the editing mode.)

```
25 AGC.runAction('wave') # The parameter is the action group name including
   not suffix passed in as a character
26 AGC.runAction('wave') # The parameter is the action group name including
   not suffix passed in as a character
```

4) Now, modify the parameter in function with the name of action group to be performed. Then, save and exit. Finally, enter command “sudo python3 ActionGroupControlDemo.py” .

```
ubuntu@ubuntu:~/ArmPi_PC_Software$ sudo python3 ActionGroupControlDemo.py
```