

## Lesson 17 Configuring Auto-run Program

### 17.1 Configure Auto-run Program

- This tutorial only introduces the methods for configuring auto-run for the Adeept Robot; for more details of auto-run on Raspberry Pi, refer to the document [Auto-Run](#) by itechfy: <https://www.itechfy.com/tech/auto-run-python-program-on-raspberry-pi-startup/>
- If you have installed the dependent libraries, the script program is already configured to auto-run. Here we will explain how to configure from scratch.

The following uses the default path **home/pi** of the Raspberry Pi. If you have changed the username of the Raspberry Pi, you need to change pi to the username you use. (**home/<username>**)

Enter "**pwd**" on the Raspberry Pi command line to view the path where the current file is located.

1. First, type in the code to create a file [startup.sh](#):

```
sudo touch //home/<username>/startup.sh
```

```
sudo touch //home/pi/startup.sh
```

2. Edit the startup.sh file:

```
sudo nano startup.sh
```

3. Write the following code into the startup.sh file. Insert next to python3 the program to auto run, and enter the file name of your product for [RobotName]. Note that you should use an absolute path here. We take the webServer.py for example:

```
#!/bin/sh
```

```
sudo python3 //home/<username>/[RobotName]/web/webServer.py
```

For instance, configure the auto-run program for the RaspTank product:

```
#!/bin/sh  
sudo python3 //home/pi/Aadept_RaspTank_Metal/web/webServer.py
```

4. Press Ctrl + X to exit editing, press Y to save your changes, and press Enter to confirm exit.
5. Grant permission for startup.sh, in which **\*\*\*** is the code for Linux permission. We do not suggest using the permission **777**, yet it can help novice with user account and permission problems. You can also set it as 700 so only the owner can read, write, and run startup.sh. To learn more about the Linux permissions, refer to the article Understanding File Permissions by maketecheasier:

<https://www.maketecheasier.com/file-permissions-what-does-chmod-777-means/>

```
sudo chmod 777 //home/pi/startup.sh
```

6. Edit the rc.local file to configure the auto-run script

```
sudo nano /etc/rc.local
```

7. Add the line below to the file of the rc.local file, save and exit:

```
//home/pi/startup.sh start
```

- You can also replace the script path above to any other script to auto-run.

## 17.2 Modify Auto-run Program

If you have purchased several products, you can choose the program to auto-run through this tutorial.

To change the program to auto-run, you just need to edit the *startup.sh* file:

```
sudo nano //home/pi/startup.sh
```

For instance, if you want to replace *webServer.py* with *app.py*, all you need is to edit the following code ([RobotName] is the file name of your product):

Replace

```
sudo python3 [RobotName]/web/webServer.py
```

with

```
sudo python3 [RobotName]/web/app.py
```

E.g:

Replace “`sudo python3 //home/pi/Adeept_RaspTank_Metal/web/webServer.py`” with  
“`sudo python3 //home/pi/Adeept_RaspTank_Metal/web/app.py`”

Press Ctrl + X to exit editing, press Y to save your changes, and press Enter to confirm exit.

When the robot boots up next time, *server.py*, instead of *webServer.py*, will auto-run then.

- *server.py* is the *socket* server when python GUI is used. We do not recommend beginners to use because you need to manually install many dependent libraries in the computer that controls it before communicating with GUI. To avoid difficulties of installation, it's thus recommended for novice to control the Raspberry Pi robot with WEB application instead.

When you don't need the Raspberry Pi to automatically run the program after booting, just delete “`sudo python3 //home/pi/[RobotName]/web/webServer.py`” or add “#” in front.

E.g:   # `sudo python3 //home/pi/Adeept_RaspTank_Metal/web/webServer.py`