

1、 Environmental construction

1、 Install Speech_Lib library

First, set the name to 'py'_ Install_ Copy the V0.0.1 folder to the root directory of your own system, and then enter the folder,

```
cd py_install  
sudo python3 setup.py install
```

```
.running install  
running bdist_egg  
running egg_info  
writing Speech_Lib.egg-info/PKG-INFO  
writing dependency_links to Speech_Lib.egg-info/dependency_links.txt  
writing top-level names to Speech_Lib.egg-info/top_level.txt  
reading manifest file 'Speech_Lib.egg-info/SOURCES.txt'  
writing manifest file 'Speech_Lib.egg-info/SOURCES.txt'  
installing library code to build/bdist.linux-x86_64/egg  
running install_lib  
running build_py  
creating build/bdist.linux-x86_64/egg  
creating build/bdist.linux-x86_64/egg/Speech_Lib  
copying build/lib/Speech_Lib/Speech_Lib.py -> build/bdist.linux-x86_64/egg/Speech_Lib  
copying build/lib/Speech_Lib/__init__.py -> build/bdist.linux-x86_64/egg/Speech_Lib  
byte-compiling build/bdist.linux-x86_64/egg/Speech_Lib/Speech_Lib.py to Speech_Lib.cpython-38.py  
byte-compiling build/bdist.linux-x86_64/egg/Speech_Lib/__init__.py to __init__.cpython-38.pyc  
creating build/bdist.linux-x86_64/egg/EGG-INFO  
copying Speech_Lib.egg-info/PKG-INFO -> build/bdist.linux-x86_64/egg/EGG-INFO  
copying Speech_Lib.egg-info/SOURCES.txt -> build/bdist.linux-x86_64/egg/EGG-INFO  
copying Speech_Lib.egg-info/dependency_links.txt -> build/bdist.linux-x86_64/egg/EGG-INFO  
copying Speech_Lib.egg-info/top_level.txt -> build/bdist.linux-x86_64/egg/EGG-INFO  
zip_safe flag not set; analyzing archive contents...  
creating 'dist/Speech_Lib-0.0.1-py3.8.egg' and adding 'build/bdist.linux-x86_64/egg' to it  
removing 'build/bdist.linux-x86_64/egg' (and everything under it)  
Processing Speech_Lib-0.0.1-py3.8.egg  
Removing /usr/local/lib/python3.8/dist-packages/Speech_Lib-0.0.1-py3.8.egg  
Copying Speech_Lib-0.0.1-py3.8.egg to /usr/local/lib/python3.8/dist-packages  
Speech-Lib 0.0.1 is already the active version in easy-install.pth  
  
Installed /usr/local/lib/python3.8/dist-packages/Speech_Lib-0.0.1-py3.8.egg  
Processing dependencies for Speech-Lib==0.0.1  
Finished processing dependencies for Speech-Lib==0.0.1
```

Use the following command to check if the installation was successful,

```
pip list
```

```

service-pkg          0.0.0
setuptools            45.2.0
simplejson             3.16.0
sip                   4.19.21
six                   1.14.0
snowballstemmer       2.0.0
soupsieve             1.9.5
Speech-Lib            0.0.1
srus2                 0.9.5
systemd-python        234
teleop-twist-keyboard 2.3.2
topic-monitor         0.9.4
topic-pkg             0.0.0
Twisted               18.9.0
txaio                 2.10.0

```

2、Bind Port

Query USB device ID,

```
lsusb
```

```

yahboom@yahboom-virtual-machine:~$ lsusb
Bus 004 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 003 Device 004: ID 0e0f:0002 VMware, Inc. Virtual USB Hub
Bus 003 Device 003: ID 0e0f:0002 VMware, Inc. Virtual USB Hub
Bus 003 Device 007: ID 1a86:7523 QinHeng Electronics HL-340 USB-Serial adapter
Bus 003 Device 002: ID 0e0f:0003 VMware, Inc. Virtual Mouse
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 002: ID 0e0f:0002 VMware, Inc. Virtual USB Hub
Bus 002 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub

```

Create a new my_Speech.rules file, terminal input,

```
sudo gedit /etc/udev/rules.d/my_speech.rules
```

Copy the following content into this file,

```

KERNEL=="ttyUSB*",ATTRS{idVendor}=="1a86",ATTRS{idProduct}=="7523",MODE:="0777",SYMLINK+="myspeech"

```

After saving, exit and enter the following command to refresh the port rule file,

```

sudo udevadm trigger
sudo service udev reload
sudo service udev restart

```

```

yahboom@yahboom-virtual-machine:~$ ll /dev/myspeech
lrwxrwxrwx 1 root root 7 May 30 17:14 /dev/myspeech -> ttyUSB0

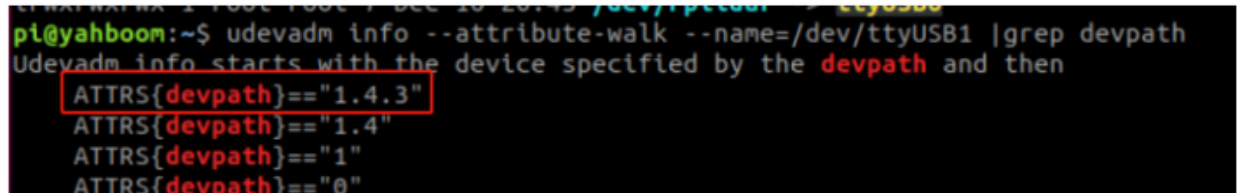
```

The appearance of the above image indicates successful binding.

The above binding method is only applicable to a unique USB device ID. Sometimes, there may be the same device ID. For example, if there are two 1a86:7523 devices, then the ATTRS {devpath} parameter and ATTRS {devpath} parameter need to be added to the query method. For example, if I confirm that ttyUSB1 is a voice board, then input,

```
udevadm info --attribute-walk --name=/dev/ttyUSB1 |grep devpath
```

The terminal will print the following content, based on the actual situation ,



A terminal window showing the command `udevadm info --attribute-walk --name=/dev/ttyUSB1 |grep devpath`. The output lists several ATTRS{devpath} values: "1.4.3", "1.4", "1", and "0". The first value, "1.4.3", is enclosed in a red rectangular box. Above the output, a text overlay reads: "Udevadm info starts with the device specified by the devpath and then".

The red box represents the value of ATTRS {devpath}, so my_ The speech.rules file needs to be modified to the following content,

```
KERNEL=="ttyUSB*",ATTRS{devpath}=="1.4.3",ATTRS{idVendor}=="1a86",ATTRS{idProduct}=="7523",MODE:="0777",SYMLINK+="myspeech"
```

After saving, exit and enter the following command to refresh the port rule file ,

```
sudo udevadm trigger
sudo service udev reload
sudo service udev restart
```

Note that after binding, it cannot be inserted into the driver port, otherwise it will not be recognized.