

Lab Submission

ESE 3005: EMBEDDED SYSTEMS ARCHITECTURE II

Lambton College in Toronto

Instructor: Takis Zourntos

STUDENT NAME & ID:

VISHAL HASRAJANI(C0761544)

Parth Patel(C0764929)

Goutham Reddy Alugubelly(C0747981)

Ratnajahnavi rebbapragada(C0762196)

Introduction:

In our embedded linux platform, to automatically mount a micro-SD external device, we need udev rule, service file and a shell script. Udev rules determine how to identify devices that is constant through reboots. Udev receives device event and matches the rules with device attributes in sysfs to identify a particular device.

Description :

In order to mount a micro-Sd card on the beaglebone, we created udev rule, service file and shell script which are linking to each other. It can be seen as follows.

In order to copy udev rule (i.e **10-usb-drive-vish.rules**) to /etc/udev/rules.d/10-usb-drive-vish.rules, we used the command

```
sudo cp 10-usb-drive-vish.rules  
/etc/udev/rules.d/10-usb-drive-vish.rules
```

```
debian@beaglebone: /etc/udev/rules.d
File Edit View Search Terminal Help
Support: http://elinux.org/Beagleboard:BeagleBoneBlack_Debian
default username:password is [debian:temppwd]
debian@192.168.7.2's password:
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Apr  6 22:08:48 2020 from 192.168.7.1
debian@beaglebone:~$ cd /etc/system
-bash: cd: /etc/system: No such file or directory
debian@beaglebone:~$ cd /etc/udev/rules.d/
debian@beaglebone:/etc/udev/rules.d$ ls
10-of-symlink.rules      80-gpio-noroot.rules      86-rpmsg-noroot.rules
10-usb-drive-vish.rules  81-pwm-noroot.rules       87-iio-noroot.rules
50-hidraw.rules          82-gpio-config-pin.rules  tisdk.rules
50-spi.rules             83-eqep-noroot.rules      uio.rules
60-omap-tty.rules        85-gpio-noroot.rules
80-EEPROM-noroot.rules   86-remoteproc-noroot.rules
debian@beaglebone:/etc/udev/rules.d$
```

10-usb-drive-vish.rules :

```
debian@beaglebone: /etc/udev/rules.d
File Edit View Search Terminal Help
GNU nano 3.2                  10-usb-drive-vish.rules
SUBSYSTEMS=="mmc", ATTRS{serial}=="0x597be4b7", ACTION=="add", ENV{SYSTEMD_WANTS}=="vish_mmc_drive_mount.service"
SUBSYSTEMS=="mmc", ATTRS{serial}=="0x597be4b7", ACTION=="remove", ENV{SYSTEMD_WANTS}=="vish_mmc_drive_unmount.service"

^G Get Help      ^O Write Out    ^W Where Is     ^K Cut Text     ^J Justify      ^C Cur Pos      M-U Undo        M-A Mark Text
^X Exit          ^R Read File    ^\ Replace      ^U Uncut Text   ^T To Spell     ^_ Go To Line    M-F Redo        M-G Copy Text
```

All the attributes are written with the help of **udevadm info -a -n /dev/mmcblk0p1 | more** command

[illegible]

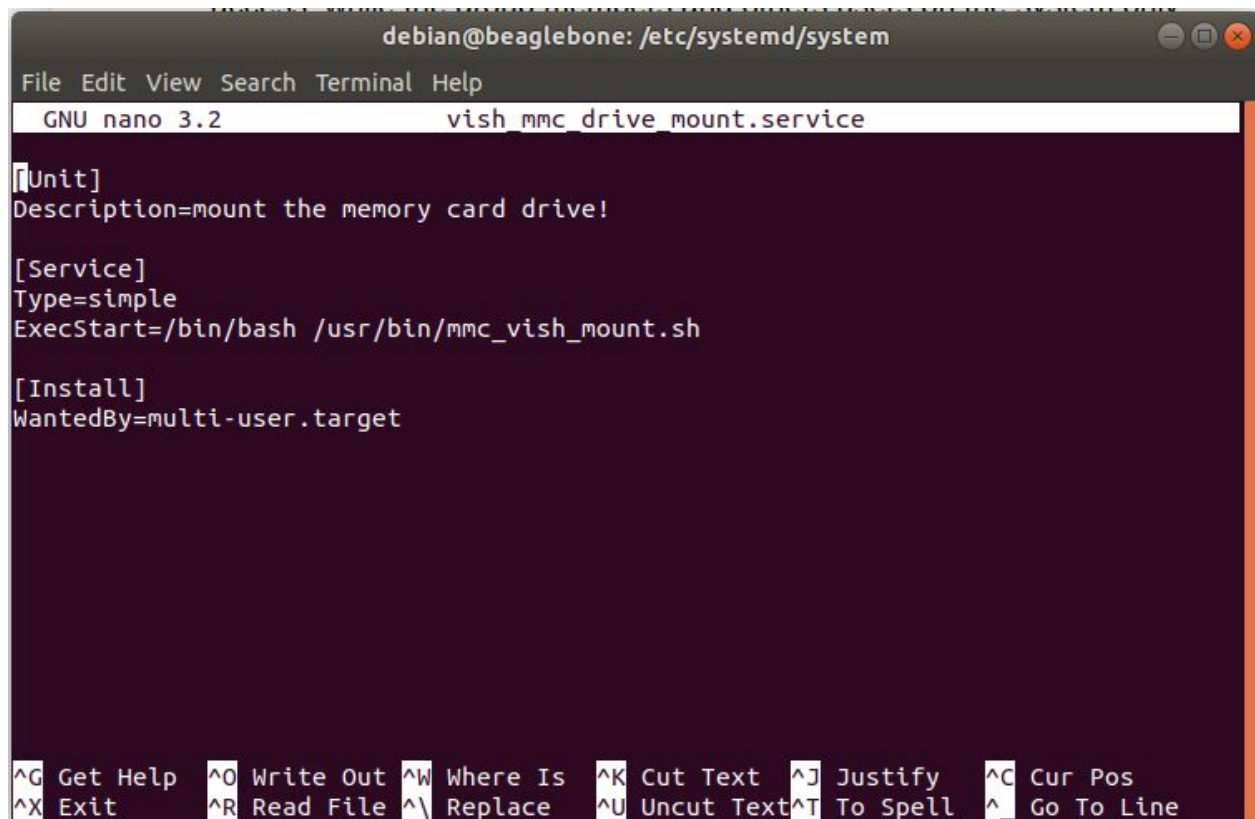
Now coming to the systemd service file, we copied **vish_mmc_drive_mount.service** to **/etc/systemd/system/vish_mmc_drive_mount.service** using the command **sudo cp vish_mmc_drive_mount.service /etc/systemd/system/vish_mmc_drive_mount.service**

Then we made that file executable and gave 644 permissions to that file.

```
sudo chmod 644 /etc/systemd/system/vish_mmc_drive_mount.service
```

Here 644 permission means the owner of the file has read and write access, while the group members and others users on the system only have read access.

vish_mmc_drive_mount.service



The screenshot shows a terminal window titled "debian@beaglebone: /etc/systemd/system". Inside, the nano 3.2 text editor is open, editing the file "vish_mmc_drive_mount.service". The editor's content is as follows:

```
[Unit]
Description=mount the memory card drive!

[Service]
Type=simple
ExecStart=/bin/bash /usr/bin/mmc_vish_mount.sh

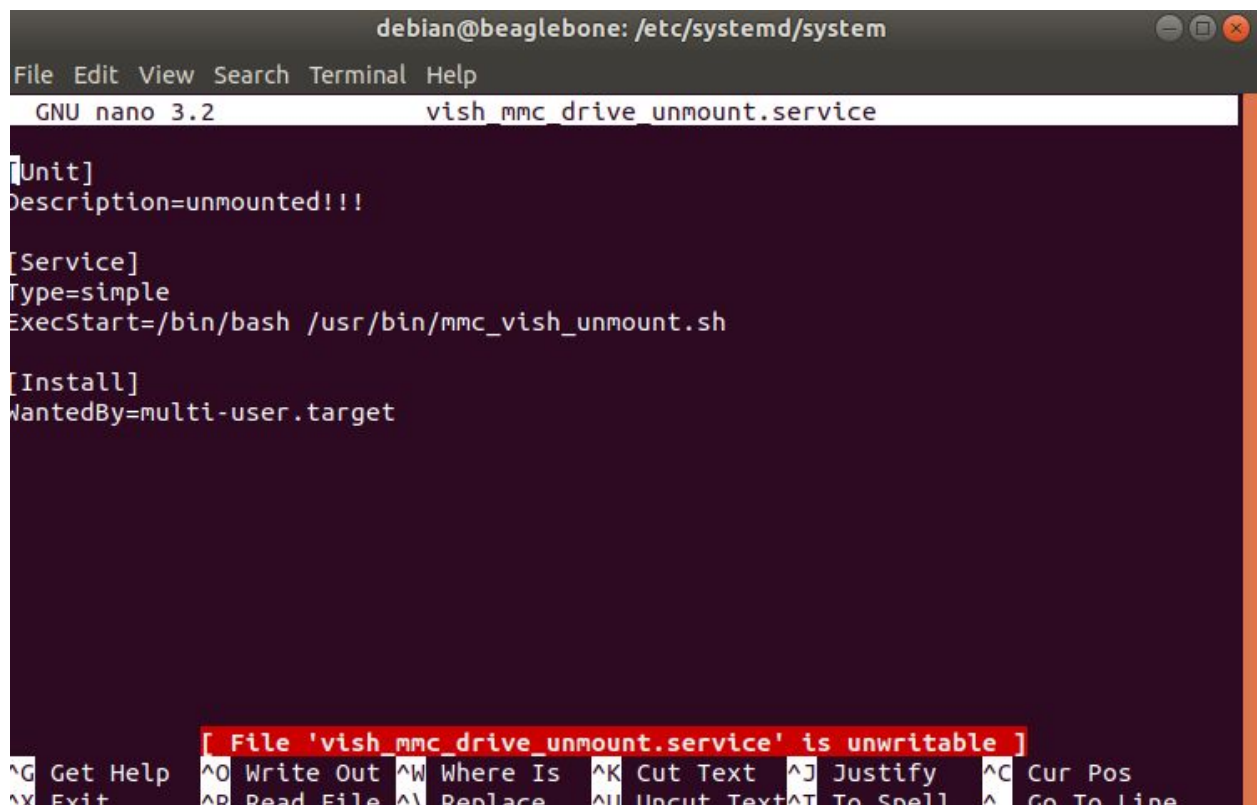
[Install]
WantedBy=multi-user.target
```

The bottom of the terminal window displays a row of keyboard shortcuts for nano:

^G Get Help	^O Write Out	^W Where Is	^K Cut Text	^J Justify	^C Cur Pos
^X Exit	^R Read File	^_ Replace	^U Uncut Text	^T To Spell	^_ Go To Line

Similarly we wrote a service file for unmounting micro-SDcard .

vish_mmc_drive_unmount.service

A screenshot of a terminal window showing the nano text editor editing the file `vish_mmc_drive_unmount.service` in the directory `/etc/systemd/system`. The file content is as follows:

```
[Unit]
Description=unmounted!!!

[Service]
Type=simple
ExecStart=/bin/bash /usr/bin/mmc_vish_unmount.sh

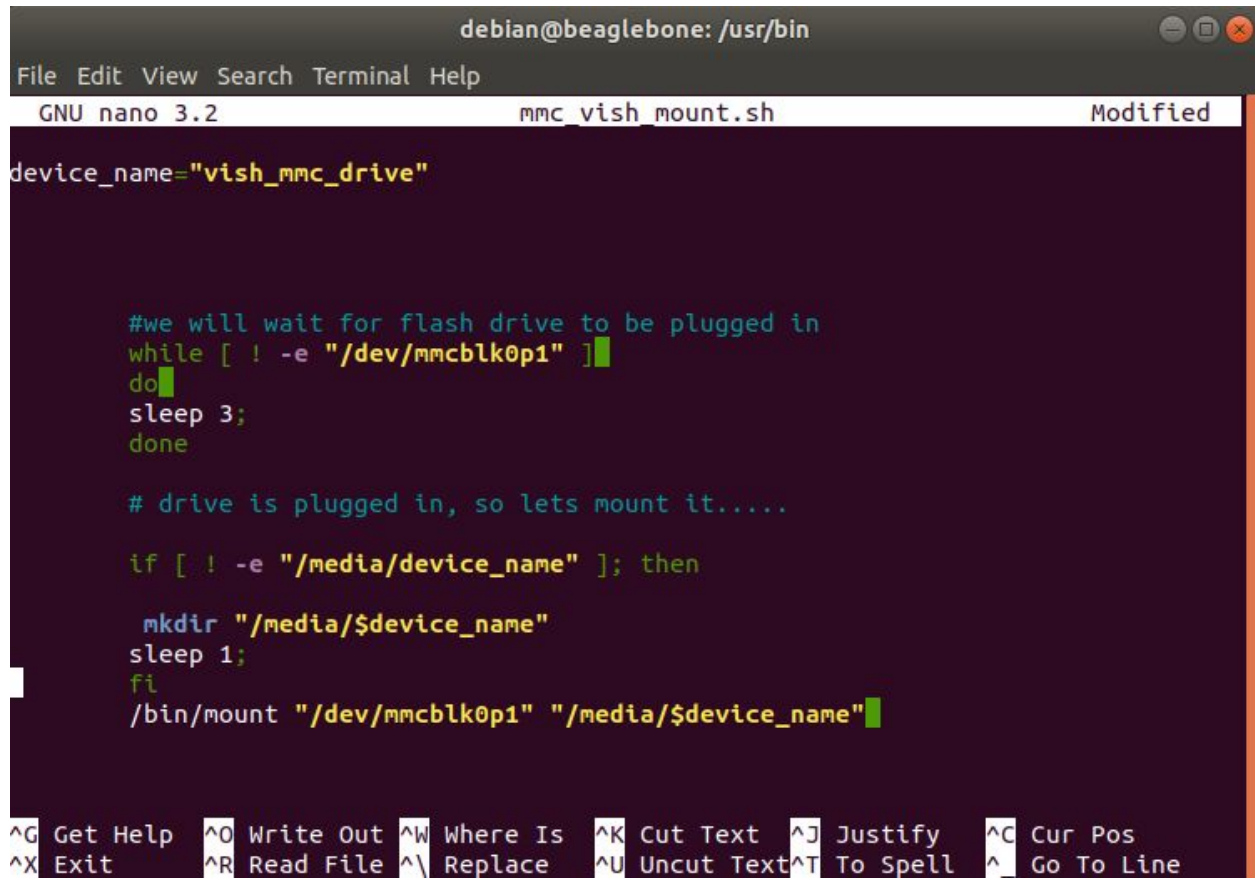
[Install]
WantedBy=multi-user.target
```

The terminal window title is `debian@beaglebone: /etc/systemd/system`. The nano editor's status bar at the bottom shows `GNU nano 3.2` and a red error message: `[File 'vish_mmc_drive_unmount.service' is unwritable]`. The bottom of the screen displays nano editor shortcuts like `^G Get Help`, `^O Write Out`, etc.

Finally, we copied 2 bash scripts (mount and unmount scripts) to `/usr/bin` and made that executable.

```
sudo cp mmc_vish_mount.sh /usr/bin/mmc_vish_mount.sh
sudo chmod +x /usr/bin/mmc_vish_mount.sh
```

mmc_vish_mount.sh



```
debian@beaglebone: /usr/bin
File Edit View Search Terminal Help
GNU nano 3.2      mmc_vish_mount.sh      Modified

device_name="vish_mmc_drive"

#we will wait for flash drive to be plugged in
while [ ! -e "/dev/mmcblk0p1" ]
do
sleep 3;
done

# drive is plugged in, so lets mount it.....

if [ ! -e "/media/device_name" ]; then

    mkdir "/media/$device_name"
    sleep 1;
fi
/bin/mount "/dev/mmcblk0p1" "/media/$device_name"

^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text ^T To Spell ^_ Go To Line
```

Working :

First of all we will wait for the SD card to be plugged in. Here, while loop checks whether `/dev/mmcblk0p1` exists or not. If it doesn't exist then sleep for 3sec.

Then using the if condition, we will make a directory called `vish_mmc_drive` and then sleep for 1sec.

Finally we mount the micro SD-card using `/bin/mount "/dev/mmcblk0p1" "/media/vish_mmc_drive"`.

Similarly **mmc_vish_unmount.sh** is implemented

```
debian@beaglebone: /usr/bin
File Edit View Search Terminal Help
GNU nano 3.2          mmc_vish_unmount.sh

device_name="vish_mmc_drive"

umount "/dev/mmcblk0p1"
rmdir "/media/$device_name"
```

Working :

Unmounting of flash drive using umount “/dev/mmcblk0p1”

Removing the created directory using rmdir.

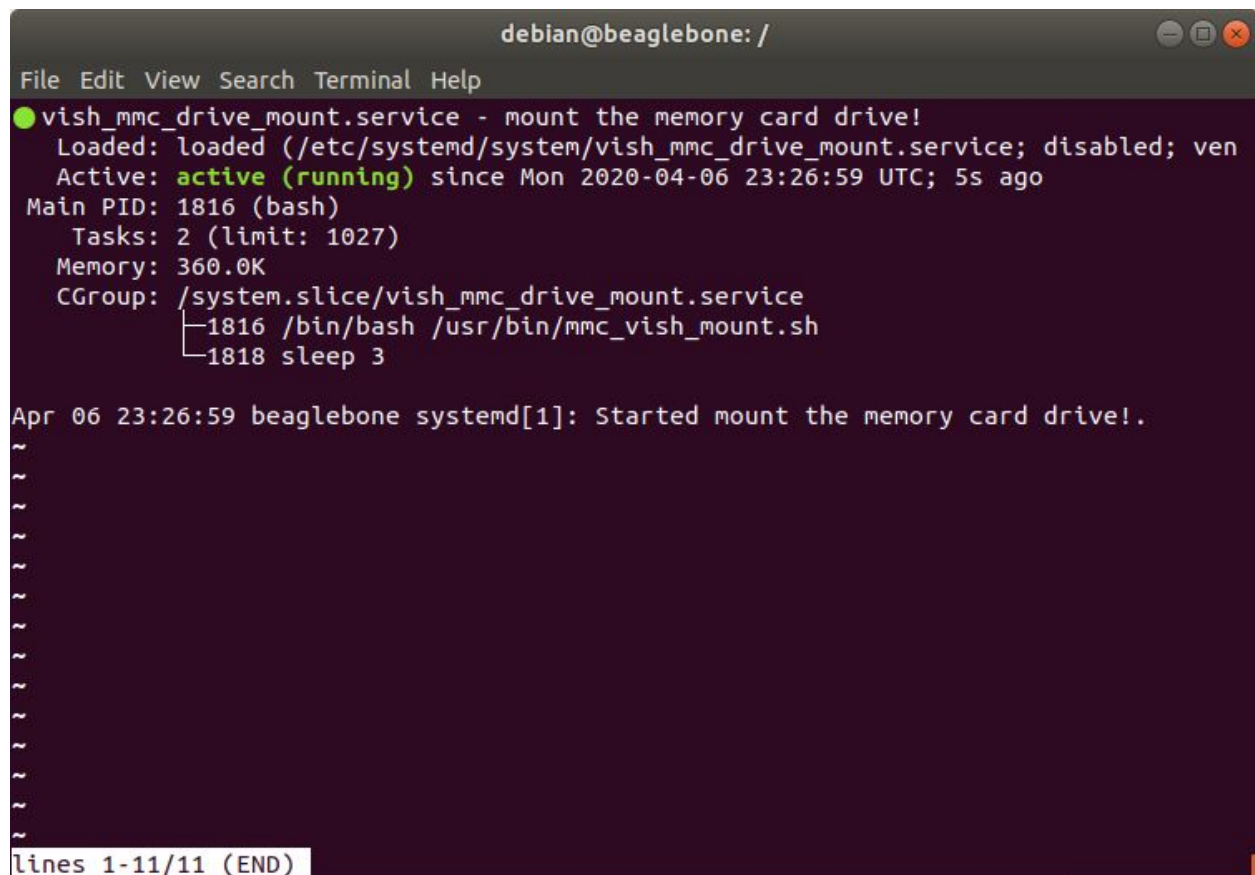
Rmdir “/media/vish_mmc_drive”.

Now let's start the service (vish_mmc_drive_mount.service).

```
sudo systemctl start vish_mmc_drive_mount.service
```

Status :

```
sudo systemctl status vish_mmc_drive_mount.service
```

A terminal window titled 'debian@beaglebone: /' with a menu bar (File, Edit, View, Search, Terminal, Help). The output of 'sudo systemctl status vish_mmc_drive_mount.service' is displayed. It shows the service is active (running) since Mon 2020-04-06 23:26:59 UTC. The main PID is 1816 (bash). Tasks: 2 (limit: 1027). Memory: 360.0K. CGroup: /system.slice/vish_mmc_drive_mount.service, with sub-entries for PID 1816 (/bin/bash /usr/bin/mmc_vish_mount.sh) and PID 1818 (sleep 3). A log message at the bottom states: 'Apr 06 23:26:59 beaglebone systemd[1]: Started mount the memory card drive!.'. The terminal shows several tilde characters (~) representing scrollback history. A status bar at the bottom left indicates 'lines 1-11/11 (END)'.

```
debian@beaglebone: /
File Edit View Search Terminal Help
● vish_mmc_drive_mount.service - mount the memory card drive!
   Loaded: loaded (/etc/systemd/system/vish_mmc_drive_mount.service; disabled; ven
   Active: active (running) since Mon 2020-04-06 23:26:59 UTC; 5s ago
 Main PID: 1816 (bash)
   Tasks: 2 (limit: 1027)
  Memory: 360.0K
   CGroup: /system.slice/vish_mmc_drive_mount.service
           └─1816 /bin/bash /usr/bin/mmc_vish_mount.sh
             └─1818 sleep 3

Apr 06 23:26:59 beaglebone systemd[1]: Started mount the memory card drive!..
~
~
~
~
~
~
~
~
~
~
~
lines 1-11/11 (END)
```

After starting the service ,we have to insert our micro SD-card and check whether it is mounted or not.

```
debian@beaglebone: /
File Edit View Search Terminal Help
~
~
~
~
~
~
~
~
~
~
lines 1-11/11 (END)
[2]+ Stopped sudo systemctl status vish_mmc_drive_mount.service
debian@beaglebone:/$ lsblk
lsblk
NAME          MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
mmcblk1       179:0    0   3.6G  0 disk
└─mmcblk1p1    179:1    0   3.6G  0 part /
mmcblk1boot0  179:8    0    2M  1 disk
mmcblk1boot1  179:16   0    2M  1 disk
mmcblk0       179:24   0  29.7G  0 disk
└─mmcblk0p1    179:25   0  29.7G  0 part /media/vish_mmc_drive
debian@beaglebone:/$
```

Checking whether directory is created in /media or not as follows :

```
debian@beaglebone: /media
File Edit View Search Terminal Help
~
~
~
~
~
~
lines 1-11/11 (END)
[2]+  Stopped                  sudo systemctl status vish_mmc_drive_mount.service
debian@beaglebone:/$ lsblk
lsblk
NAME                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
mmcblk1             179:0    0   3.6G  0 disk
└─mmcblk1p1          179:1    0   3.6G  0 part /
mmcblk1boot0         179:8    0    2M   1 disk
mmcblk1boot1         179:16   0    2M   1 disk
mmcblk0              179:24   0  29.7G  0 disk
└─mmcblk0p1          179:25   0  29.7G  0 part /media/vish_mmc_drive
debian@beaglebone:/$ cd /media
debian@beaglebone:/media$ ls
vish_mmc_drive
debian@beaglebone:/media$
```

To unmount the micro SD-card we will run our service i.e **vish_mmc_drive_unmount.service**

```
debian@beaglebone: /media
File Edit View Search Terminal Help
[2]+  Stopped                                sudo systemctl status vish_mmc_drive_mount.service
debian@beaglebone:/$ lsblk
lsblk
NAME                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
mmcblk1              179:0    0   3.6G  0 disk
└─mmcblk1p1          179:1    0   3.6G  0 part /
mmcblk1boot0         179:8    0    2M  1 disk
mmcblk1boot1         179:16   0    2M  1 disk
mmcblk0              179:24   0  29.7G  0 disk
└─mmcblk0p1          179:25   0  29.7G  0 part /media/vish_mmc_drive
debian@beaglebone:/$ cd /media
debian@beaglebone:/media$ ls
vish_mmc_drive
debian@beaglebone:/media$ cd ..
debian@beaglebone:/$ sudo systemctl start vish_mmc_drive_unmount.service
debian@beaglebone:/$ lsblk
NAME                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
mmcblk1              179:0    0   3.6G  0 disk
└─mmcblk1p1          179:1    0   3.6G  0 part /
mmcblk1boot0         179:8    0    2M  1 disk
mmcblk1boot1         179:16   0    2M  1 disk
mmcblk0              179:24   0  29.7G  0 disk
└─mmcblk0p1          179:25   0  29.7G  0 part
debian@beaglebone:/$ cd /media
debian@beaglebone:/media$ ls
debian@beaglebone:/media$
```

So,unmounting the SD-card is done successfully and the directory is also deleted after unmounting it.

CONCLUSION :

To summarize, we can say that mounting and unmounting of a particular device can be done by adding particular rule in the udev rules.d folder and then linking it to the *.service and *.sh file.

APPENDIX:

10-usb-drive-vish.rules

```
SUBSYSTEMS=="mmc", ATTRS{serial}=="0x597be4b7", ACTION=="add",  
ENV{SYSTEMD_WANTS}=="vish_mmc_drive_mount.service"  
SUBSYSTEMS=="mmc", ATTRS{serial}=="0x597be4b7", ACTION=="remove",  
ENV{SYSTEMD_WANTS}=="vish_mmc_drive_unmount.service"
```

vish_mmc_drive_mount.service

[Unit]

Description=mount the memory card drive!

[Service]

Type=simple

ExecStart=/bin/bash /usr/bin/mmc_vish_mount.sh

[Install]

WantedBy=multi-user.target

vish_mmc_drive_unmount.service

[Unit]

Description=unmounted!!!

[Service]

Type=simple

ExecStart=/bin/bash /usr/bin/mmc_vish_unmount.sh

[Install]

WantedBy=multi-user.target

mmc_vish_mount.sh

```
device_name="vish_mmc_drive"
```

```
#we will wait for flash drive to be plugged in
while [ ! -e "/dev/mmcblk0p1" ]
do
sleep 3;
done
```

```
# drive is plugged in, so lets mount it.....
```

```
if [ ! -e "/media/device_name" ]; then
```

```
    mkdir "/media/$device_name"
```

```
sleep 1;
```

```
fi
```

```
/bin/mount "/dev/mmcblk0p1" "/media/$device_name"
```

mmc_vish_unmount.sh

```
device_name="vish_mmc_drive"
```

```
umount "/dev/mmcblk0p1"
```

```
rmdir "/media/$device_name"
```