

**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  ^L Replace  ^U Uncut Text  ^T To Spell  ^G 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

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

```
debian@beaglebone: /etc/systemd/system
File Edit View Search Terminal Help
GNU nano 3.2           vish_mmc_drive_mount.service

[Unit]
Description=mount the memory card drive!

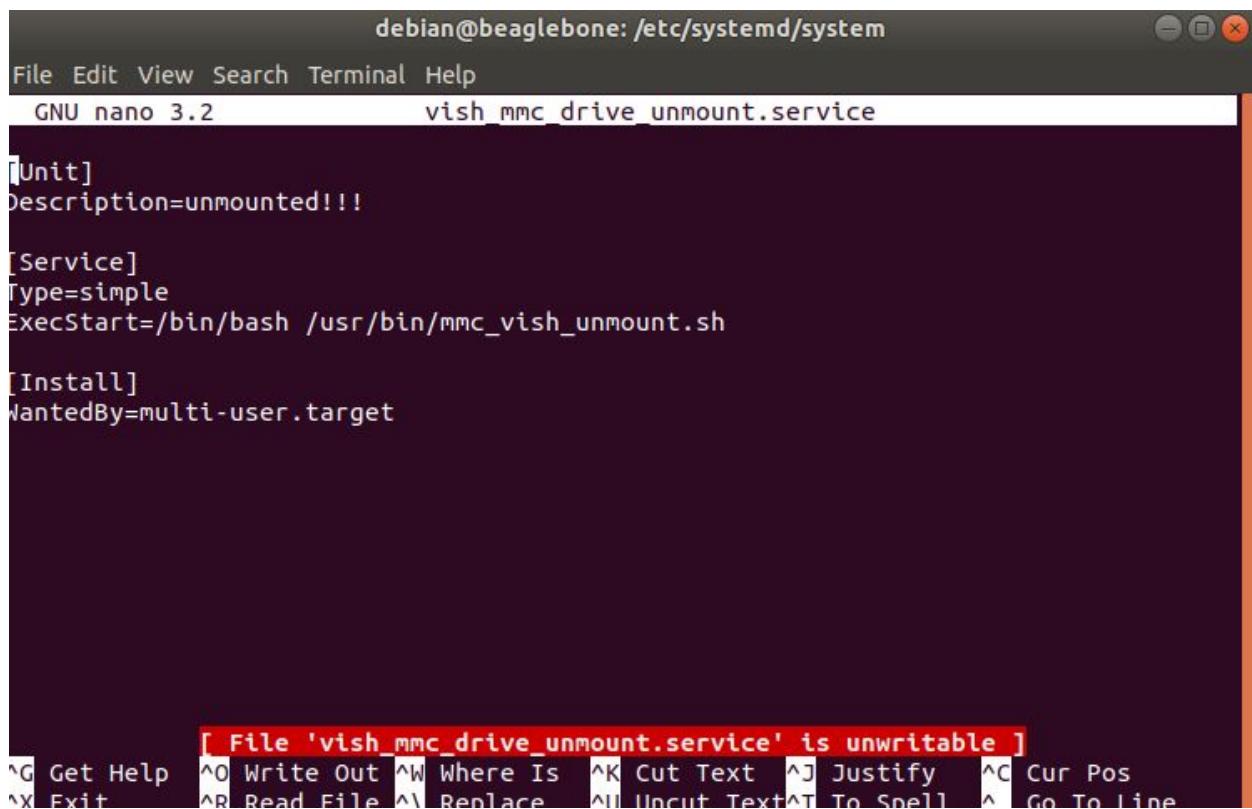
[Service]
Type=simple
ExecStart=/bin/bash /usr/bin/mmc_vish_mount.sh

[Install]
WantedBy=multi-user.target

^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**



The screenshot shows a terminal window titled "debian@beaglebone: /etc/systemd/system". The window title bar also displays "File Edit View Search Terminal Help" and "GNU nano 3.2". The main content of the terminal is the configuration for the service file:

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

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

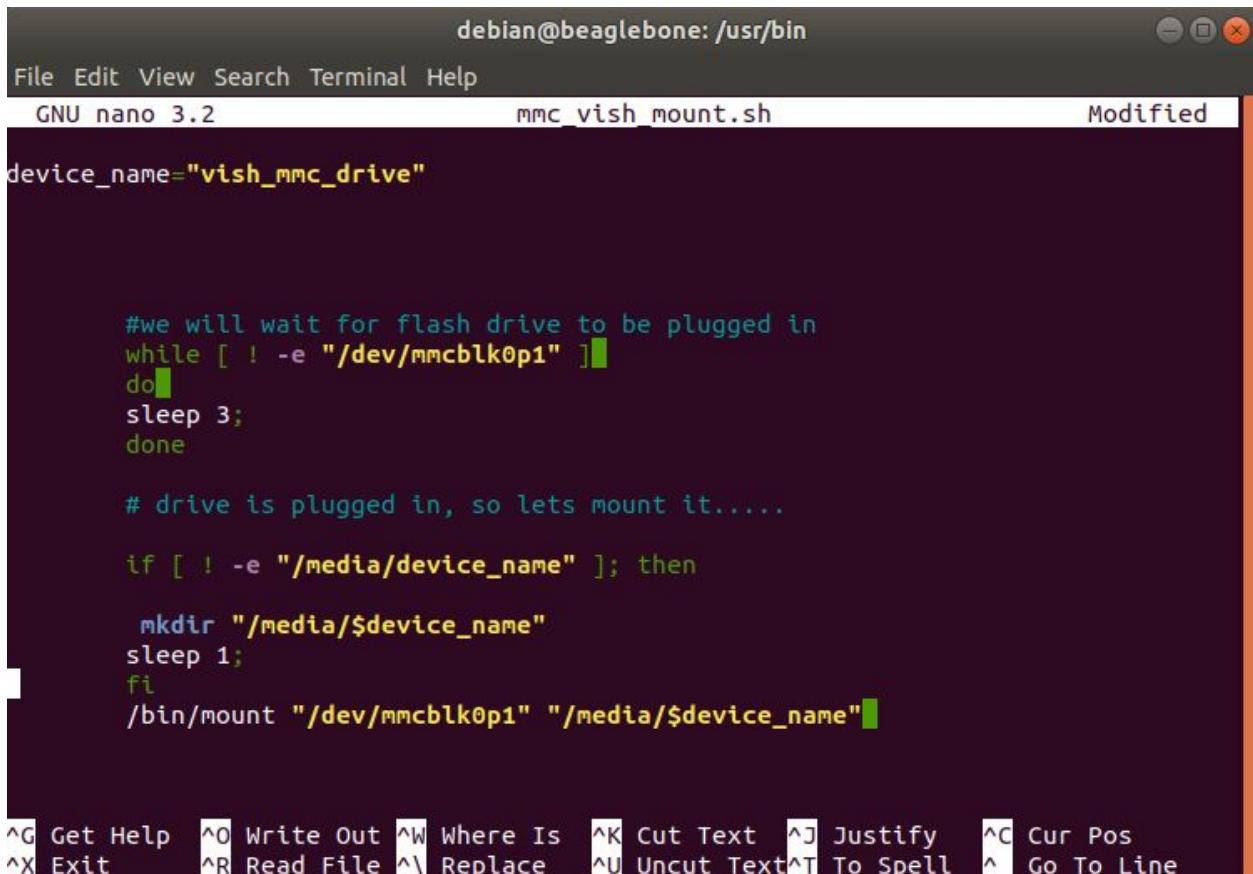
[Install]
WantedBy=multi-user.target
```

At the bottom of the terminal, there is a status message in red brackets: "[ File 'vish\_mmc\_drive\_unmount.service' is unwritable ]". Below the message, there is a menu bar with various keyboard shortcuts for nano editor commands.

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



The screenshot shows a terminal window titled "debian@beaglebone: /usr/bin" running the "nano" text editor. The file being edited is "mmc\_vish\_mount.sh". The code in the file is as follows:

```
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"
```

At the bottom of the terminal window, there is a menu bar with options: File, Edit, View, Search, Terminal, Help. The status bar indicates the file is "Modified". The bottom of the screen shows the nano editor's command-line interface with various keyboard shortcuts.

### **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 :

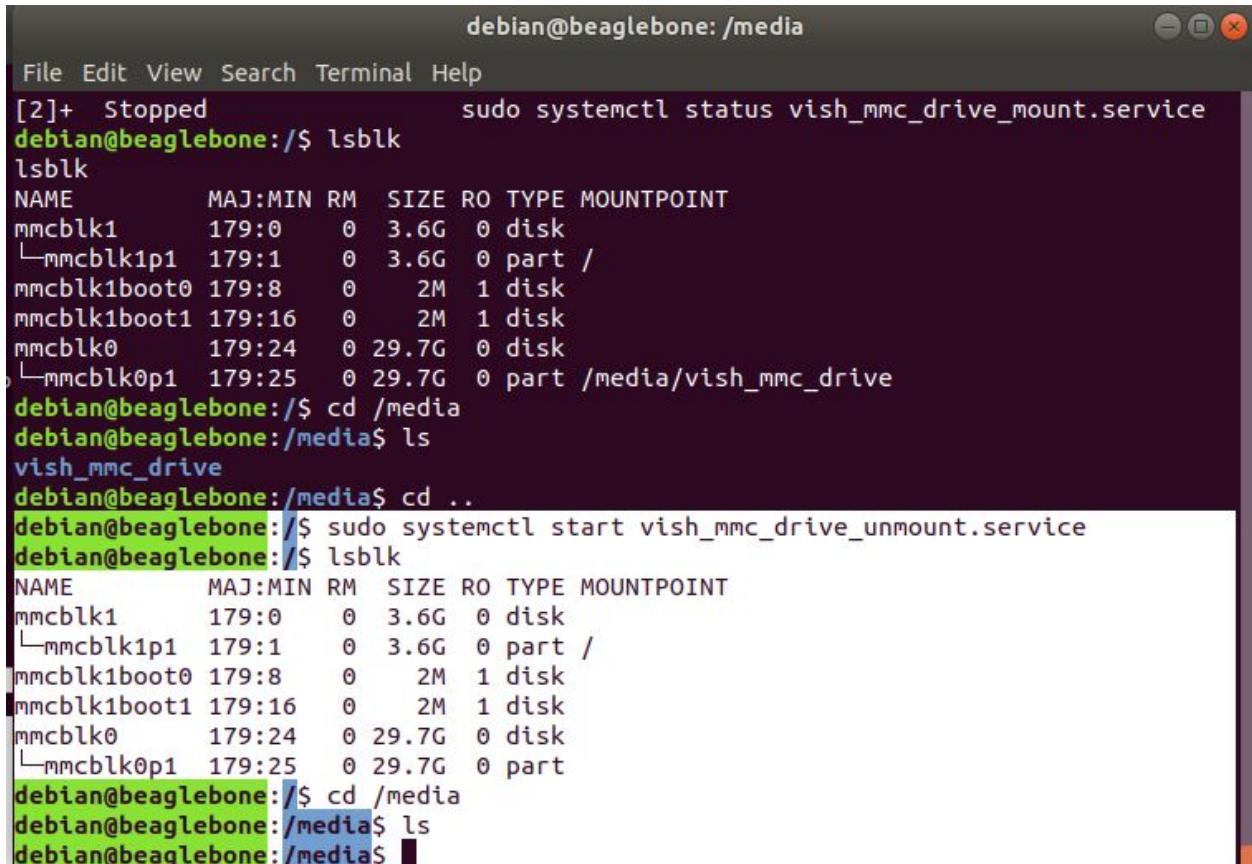
```
sudo systemctl status vish_mmc_drive_mount.service
```

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

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**



The screenshot shows a terminal window titled "debian@beaglebone: /media". The terminal displays the following command sequence:

```
sudo systemctl status vish_mmc_drive_mount.service
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
cd /media
ls
vish_mmc_drive
cd ..
sudo systemctl start vish_mmc_drive_unmount.service
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
cd /media
ls
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"
```