

# Setting up overlayFS on Raspberry Pi

www.domoticz.com

November 8th, 2015

**Warning:** Do this only on a new SD image that can be recreated, **or** make sure you can mount the SD card elsewhere to revert changes.

## Purpose

In this page, we will describe how to implement overlayfs using the normal SD card directory tree as the bottom (persistent) layer, and as many RAM / tmpfs areas you may want as the upper layer to provide the overlay mounts. In most cases this will do:

- */boot* and the *root* filesystem are changed to be normally read only
- */var* and */home* is changed into read/write overlay filesystems
- An init script named *saveoverlays* will take care of syncing the overlay filesystems back to the bottom layer SD card upon system shutdown and restarts

Using this, you can eliminate SD card writes almost entirely. As the read-write parts are then RAM based, this will also speed up Domoticz.

## Preparations

## Requirements

OverlayFS is included in kernel versions 3.18 and later.

In case of errors, it is strongly recommended to have be able to mount and edit the modified SD card contents via a card reader on another system

## Downloads

The easiest way to get the needed scripts into place is this:

## Setup procedure

Stop using swap

In case you also use the SD card for swap, this should be turned off as follows:

```
sudo dphys-swapfile swapoff
sudo dphys-swapfile uninstall
sudo update-rc.d dphys-swapfile disable
```

Move files you need to be able to update with a read-only root

Some files may need to be continuously updated after all changes are done. For example, if you use the *fake-hwclock* service, the data file for that service need to be moved:

```
sudo service fake-hwclock stop
sudo mv /etc/fake-hwclock.data /var/log/fake-
hwclock.data
sudo ln -s /var/log/fake-hwclock.data /etc/fake-
hwclock.data
sudo service fake-hwclock start
sudo update-rc.d fake-hwclock disable
```

The fake-hwclock actions are taken care of in the *saveoverlays* service described later. There is no harm in leaving it as is, but it will complain during start and do nothing.

Set up fuse and mount script

Install the fuse package. We need only */sbin/mount.fuse* from there. *lsuf* is essential to find open files in case you cannot remount a filesystem back to read-only

```
sudo apt-get install fuse lsuf
```

The following mount script is used in */etc/fstab* to do the actual overlay mount. When the filesystem type is *fuse*, you can use a script like this from */usr/local/bin/mount\_overlay* to do the mount via *mount.fuse*

```
#!/bin/sh
DIR="$1"
[ -z "${DIR}" ] && exit 1
if [ ! -d "${DIR}_org" ]
then
    echo "${DIR}_org does not exist" >&2
    exit 1
fi
if [ ! -d "${DIR}_rw" ]
then
    echo "${DIR}_rw does not exist" >&2
    exit 1
fi
#
# ro must be the first mount option for root .....
#
ROOT_MOUNT=$( grep -v "^#" /etc/fstab | awk
'$2=="/" { print substr($4,1,2) }' )
if [ "$ROOT_MOUNT" != "ro" ]; then
    /bin/mount --bind ${DIR}_org ${DIR}
else
    /bin/mount -t tmpfs ramdisk ${DIR}_rw
    /bin/mkdir ${DIR}_rw/upper
    /bin/mkdir ${DIR}_rw/work
    OPTS="-o
lowerdir=${DIR}_org,upperdir=${DIR}_rw/upper,workd
ir=${DIR}_rw/work"
    /bin/mount -t overlay ${OPTS} overlay ${DIR}
fi
```

Please note that if the root filesystem is read-write in /etc/fstab, the mount will be a loopback mount instead.

Set up the saveoverlays service

This *saveoverlays* service (follow the link to download) will take care of saving changes to the overlays back to the underlying SD card storage (now the *\_org* part of the overlay). It also contains some logic to check if the fake clock data should be loaded. If you are not using the *fake-hwclock* service, you should comment that out (two places).

The overlays are automatically detected, so you should not need to change any of that.

After copying the file to */etc/init.d/saveoverlays*, enable the service as follows:

```
sudo update-rc.d saveoverlays start 10 S . stop
05 0 6 .
```

**Note:** If you have remounted root to be read-write, this service assumes that changes are not to be synced. Please remount back to read-only if you want changes to be synced on next reboot.

Change boot command line

Change */boot/cmdline.txt* similar to this, i.e. add *noswap fastboot ro*

```
dwc_otg.lpm_enable=0 console=ttyAMA0,115200
console=tty1 root=/dev/mmcblk0p2 rootfstype=ext4
elevator=deadline rootwait noswap fastboot ro
```

Update */etc/fstab* for read-only SD card

*/etc/fstab* need to be updated similar to this:

```
proc          /proc          proc
defaults      0 0
/dev/mmcblk0p1 /boot          vfat
ro            0 2
/dev/mmcblk0p2 /              ext4
ro,noatime    0 1
mount_overlay /var          fuse
nofail,defaults 0 0
mount_overlay /home        fuse
nofail,defaults 0 0
none          /tmp          tmpfs
defaults      0 0
```

Prepare the overlay directories

Services that have open files should be stopped. The following should do if you use the current Domoticz SD card image

```
sudo service domoticz.sh stop
sudo service nodered stop
sudo service influxdb stop
sudo service mosquitto stop
sudo service ntp stop
sudo service rsyslog stop
```

Then change into layered mount setups as follows:

```
sudo mv /home /home_org
sudo mkdir /home /home_rw
sudo mv /var /var_org
sudo mkdir /var /var_rw
```

Mount /home and /var as overlays

This is to make sure the next reboot does not run into trouble due to moved data:

```
sudo mount /home
sudo mount /var
```

Modify domoticz.sh service

For some reason the domoticz executable is mistaken on startup directory when in an overlay filesystem:

To get around that, add the following after other **DAEMON\_ARGS** lines in */etc/init.d/domoticz.sh*

```
APPDIR=$( dirname $DAEMON )
DAEMON_ARGS="$DAEMON_ARGS -aproot $APPDIR/
-wwroot $APPDIR/www -dbase $APPDIR/domoticz.db"
```

The trailing slash for aproot is needed.

Reboot

Now reboot. Since root is not mounted read-only at this time, the *saveoverlays* service will not attempt to sync anything.

It is recommended to have a console attached so that any error messages can be detected, and problems dealt with.

## Tests

## Check your mounts

After boot, the *mount* command should display similar to this, with relevant changes in red



```
/dev/root on / type ext4
(ro,noatime,data=ordered)
devtmpfs on /dev type devtmpfs
(rw,relatime,size=234492k,nr_inodes=58623,mode=755
)
tmpfs on /run type tmpfs
(rw,nosuid,noexec,relatime,size=47756k,mode=755)
tmpfs on /run/lock type tmpfs
(rw,nosuid,nodev,noexec,relatime,size=5120k)
proc on /proc type proc
(rw,nosuid,nodev,noexec,relatime)
sysfs on /sys type sysfs
(rw,nosuid,nodev,noexec,relatime) tmpfs on
/run/shm type tmpfs
(rw,nosuid,nodev,noexec,relatime,size=95500k)
devpts on /dev/pts type devpts
(rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode
=000)
/dev/mmcblk0p1 on /boot type vfat
(ro,relatime,fmask=0022,dmask=0022,codepage=437,io
charset=ascii,shortname=mixed,errors=remount-ro)
ramdisk on /var_rw type tmpfs (rw,relatime)
overlay on /var type overlay
(rw,relatime,lowerdir=/var_org,upperdir=/var_rw
/upper,workdir=/var_rw/work)
ramdisk on /home_rw type tmpfs (rw,relatime)
overlay on /home type overlay
(rw,relatime,lowerdir=/home_org,upperdir=/home_rw
/upper,workdir=/home_rw/work)
none on /tmp type tmpfs (rw,relatime)
```

## Check that you can update the system

To do system maintenance after root and /boot is changed to read-only, proceed

as follows:

```
sudo mount -o remount,rw /
sudo mount -o remount,rw /boot
sudo apt-get update
sudo apt-get upgrade
sudo mount -o remount,ro /
sudo mount -o remount,ro /boot
sudo reboot
```

## Check that overlays are synced to SD card

These syncs are logged to */var/log/saveoverlays-log*

## References

There are many references for doing similar setups. The most similar one using unionfs is likely this:

Other material: