

Enable ubuntu on i.MX8MQ

Yocto BSP: L4.14.98_2.0.0_GA

Rootfs: ubuntu19.10-base

Board: i.MX8MQ-EVK

This is just simple guide for enable ubuntu on i.MX8M series. The Document will be continuously updated.

1. Build Yocto L4.14.98_2.0.0_GA

2. **Compile the kernel separately to replace yocto kernel**

The step is build GPU driver into Image, yocto is build GPU driver as module(galcore.ko) by default.

3. **Download ubuntu rootfs**

```
$ mkdir ~/rootfs && cd ~/rootfs
```

```
$ wget http://cdimage.ubuntu.com/ubuntu-base/releases/19.10/release/ubuntu-base-19.10-base-arm64.tar.gz
```

```
$ mkdir ubuntu-rootfs
```

```
$ tar -xvf ubuntu-base-19.10-base-arm64.tar.gz -C ubuntu-rootfs
```

```
$ cd ubuntu-rootfs
```

```
$ ls
```

4. **Install qemu-user-static on PC to simulate arm64 environment**

```
$ apt-get install qemu-user-static
```

```
$ cp /usr/bin/qemu-aarch64-static usr/bin
```

```
$ cp -b /etc/resolv.conf etc/
```

5. **Simulate arm64 environment**

```
$ cd ../
```

```
$ vi ch-mount.sh
```

(copy the content to ch-mount.sh)

```
#!/bin/bash
```

```
#
```

```
function mnt() {
```

```
    echo "MOUNTING"
```

```
    sudo mount -t proc /proc ${2}proc
```

```
    sudo mount -t sysfs /sys ${2}sys
```

```
    sudo mount -o bind /dev ${2}dev
```

```
    sudo mount -o bind /dev/pts ${2}dev/pts
```

```
    sudo chroot ${2}
```

```
}
```

```
function umnt() {
```

```
    echo "UNMOUNTING"
```

```

sudo umount ${2}proc
sudo umount ${2}sys
sudo umount ${2}dev/pts
sudo umount ${2}dev
}
if [ "$1" == "-m" ] && [ -n "$2" ];
then
    mnt $1 $2
elif [ "$1" == "-u" ] && [ -n "$2" ];
then
    umnt $1 $2
else
    echo ""
    echo "Either 1'st, 2'nd or both parameters were missing"
    echo ""
    echo "1'st parameter can be one of these: -m(mount) OR -u(umount)"
    echo "2'nd parameter is the full path of rootfs directory(with trailing '/)"
    echo ""
    echo "For example: ch-mount -m /media/sdcard/"
    echo ""
    echo 1st parameter : ${1}
    echo 2nd parameter : ${2}
fi

```

(chroot to arm64 filesystem)

```
$ ./ch-mount.sh -m ubuntu-rootfs
```

6. Install package and configuration

0) Update software list

```
$ apt-get update
```

For the previous errors: “apt-get update” failed with the below message.

```
>Could not execute 'apt-key' to verify signature (is gnupg installed?)
```

Referring to the below post,

<https://askubuntu.com/questions/759071/cant-update-upgrade-du-to-could-not-execute-apt-key-to-verify-signature>

① I changed access permissions as below and it worked.

```
#chmod 777 tmp
```

② Set DNS auto update failed, with error log “/usr/sbin/dpkg-reconfigure: resolvconf is not installed”

This is solved by installing resolvconf.

Or `echo nameserver 8.8.8.8 > /etc/resolv.conf`

Install package you need

```
$ apt-get install \  
  language-pack-en-base \  
  sudo \  
  ssh \  
  net-tools \  
  network-manager \  
  iputils-ping \  
  rsyslog \  
  bash-completion \  
  htop \  
  resolvconf \  
  dialog \  
  vim
```

```
#apt-get install v4l-utils alsa-utils git gcc less autoconf autopoint libtool bison flex gtk-doc-tools  
glib-2.0 libglib2.0-dev libpango1.0-dev libatk1.0-dev kmod pciutils libjpeg-dev
```

Add user

```
$ useradd -s '/bin/bash' -m -G adm,sudo yourusername  
$ echo "Set password for yourusername:"  
$ passwd yourusername  
$ echo "Set password for root:"  
$ passwd root  
1) Set host name  
$ echo 'ubuntu.yourusername' > /etc/hostname  
2) Set DNS auto update  
$ dpkg-reconfigure resolvconf  
3) Configure the serial port  
sudo cp -ra /yocto/L4.14.98/imx-yocto-bsp-ft/build-8qxpmeke-xwayland/tmp/work/imx8mqevk-  
poky-linux/systemd-serialgetty/1.0-r5/image/* ubuntu-roots/
```

7.build Weston

```
apt-get install libudev-dev libinput-dev libxkbcommon-dev libpam0g-dev libx11-xcb-  
dev libxcb-xfixes0-dev libxcb-composite0-dev libxcursor-dev libxcb-shape0-dev libdbus-1-  
dev libdbus-glib-1-dev
```

1)Build wayland (to remove wayland-egl dependency on mesa)

```
#wget https://wayland.freedesktop.org/releases/wayland-1.16.0.tar.xz  
#apt-get install libffi-dev libxml2-dev  
#tar -xvf ./wayland-1.16.0.tar.xz  
#./configure --disable-documentation prefix=/usr
```

```
#make
#make install
#ldconfig
```

2)Build wayland protocol (because wayland protocol version mismatch)

```
#git clone https://source.codeaurora.org/external/imx/wayland-protocols-imx.git
#cd wayland-protocols-imx/
#git checkout e05c19d9520f0b1289cf0844d6e2f877114f39d5
#./autogen.sh --prefix=/usr
#make install
#ldconfig
```

2)Build Weston(8M have not g2d)

```
#cd ~
#git clone https://source.codeaurora.org/external/imx/weston-imx.git
#cd weston-imx
#git checkout fb563901657b296c7c7c86d26602a622429e334f
./autogen.sh --prefix=/usr --disable-silent-rules --disable-dependency-tracking --enable-
setuid-install --disable-rdp-compositor --enable-clients --enable-simple-clients --enable-demo-
clients-install --disable-colord --enable-egl --enable-simple-egl-clients --enable-fbdev-compositor -
-disable-headless-compositor --enable-drm-compositor --enable-weston-launch --disable-lcms --
disable-libunwind --with-pam --disable-vaapi-recorder --enable-wayland-compositor --without-
webp --disable-x11-compositor --disable-xwayland --disable-imxg2d

#make -j4 COMPOSITOR_LIBS="-lGLESv2 -lEGL -lGAL -lwayland-server -lxxkbcommon -
lpixman-1"
#make install
```

8.Exit simulate environment

```
$ exit
$ ./ch-mount -u ubuntu-rootfs
```

9.replace filesystem

```
$ mkdir mountpoint
$ umount /dev/sdb2
$ mkfs.ext4 /dev/sdb2
$ mount /dev/sdb2 /home/gnar/mountpoint
$ cp -a /home/gnar/rootfs/ubuntu-rootfs /* /home/gnar/mountpoint/
$umount /dev/sdb2
$umount /home/gnar/mountpoint
$sync
```

10. start Weston

Confirm had GPU driver, like this

```
root@ubuntu:~# ls /dev/galcore  
/dev/galcore
```

Then run command start weston

```
# export XDG_RUNTIME_DIR=/run/user/1000  
# sudo -E weston --tty=1
```

Comment en_US.UTF-8 UTF-8

```
$ localectl set-locale LANG=en_US.UTF-8  
$ reboot
```