# Enable ubuntu on i.MX8MP

**Yocto BSP: L5.4.3_2.0.0**
**Rootfs: ubuntu19.10-base**
**Board: i.MX8MP-EVK**

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

## 1. Build Yocto **L5.4.3_2.0.0_GA**

## 2. (optional) **Compile the kernel separately to replace yocto kernel**

Kernel 5.4 has build GPU driver(galcore) as default. If you want to use USB camera, should set **CONFIG_USB_VIDEO_CLASS=y**.

## 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

5.1 Create a script to setup simulate aarch64 ubuntu 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
```

### 5.2 setup simulate aarch64 ubuntu environment

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

## 6. Install package and configuration

### 6.1 Update software list

```
$ chmod 777 /tmp
$ chmod 777 /dev/null
$ apt-get update
```

### 6.2 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 udhcpc udhcpd git v4l-utils alsa-utils git gcc less
  autoconf autopoint libtool bison flex gtk-doc-tools libglib2.0-dev
  libpango1.0-dev libatk1.0-dev kmod pciutils libjpeg-dev
```

### 6.3 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
```

#### 6.3.1 Set host name

```
$ echo 'ubuntu.yourusername' > /etc/hostname
```

#### 6.3.2 Set DNS auto update

```
$ dpkg-reconfigure resolvconf
```

## 7. Configure the serial port

```
$ sudo cp -Pra   ~/L5.4.3-2.0.0/bld-xwayland/tmp/work/imx8mpevk-poky-linux/systemd-serialgetty/1.0-r5/image /*   ubuntu-rootfs/
```

**Note: In simulate aarch64 ubuntu rootfs, should confirm blow.**

```
root@ubuntu:/etc/systemd/system/getty.target.wants# ls -al
total 8
drwxr-xr-x 2 root    root       4096 Apr   7   2020 .
drwxr-xr-x 9 imx8mp imx8mp 4096 Mar 31    2020 ..
lrwxrwxrwx 1 root     root         34 Mar 31    2020 getty@tty1.service -> /lib/systemd/system/getty@.service
lrwxrwxrwx 1 root     root         41 Apr   7    2020 serial-getty@ttymxc1.service -> /lib/systemd/system/serial-getty@.service
```

if found lost soft link, should create link.

```
#ln -s  /lib/systemd/system/serial-getty@.service        serial-getty@ttymxc1.service
```

## 8. build Weston

At present, we use the Weston as ubuntu desktop environment.

### 8.1 remove mesa GPU driver and install vivante GPU driver

#### 8.1.1 remove mesa GPU driver

```
$sudo rm -rf ubuntu-rootfs/usr/lib/aarch64-linux-gnu/libdrm*
$sudo rm -rf ubuntu-rootfs/usr/lib/aarch64-linux-gnu/mesa-egl
```

```
$sudo rm -rf ubuntu-rootfs/usr/lib/aarch64-linux-gnu/libglapi.so.0*
$sudo rm -rf /usr/lib/aarch64-linux-gnu/libwayland-*
```

**8.1.2 install vivante GPU driver**

```
$sudo cp -Pra libdrm/ 2.4.99.imx-r0 /image/* ubuntu-rootfs
$sudo cp -Pra imx-gpu-viv/ 1_6.4.0.p2.2-aarch64-r0 /image/* ubuntu-rootfs
$sudo cp -Pra imx-gpu-g2d/ 6.4.0.p2.2-r0 /image/*  ubuntu-rootfs
```

**8.1.3 install imx header**

```
$sudo cp -Pra linux-imx-headers/5.4-r0/image/*  ubuntu-rootfs
$sudo cp -Pra imx-parser/4.5.4-r0/image/*  ubuntu-rootfs
```

**8.1.4 install dependencies**

```
# 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 libsystemd-dev
libpixman-1-dev libcairo2-dev
```

## 8.2 Build wayland

```
#wget https://wayland.freedesktop.org/releases/wayland-1.18.0.tar.xz
#tar -zJvf wayland-1.18.0.tar.xz
#apt-get  install  libffi-dev  libxml2-dev  kbd  libexpat1-dev  autoconf
automake libtool meson cmake
# cd wayland-1.18.0
#./configure --disable-documentation prefix=/usr
#make -j8
#make install
#ldconfig
```

## 8.3 Build wayland protocol

```
#git clone    https://source.codeaurora.org/external/imx/wayland-protocols-imx.git
# cd wayland-protocols-imx
#git checkout wayland-protocols-imx-1.18
#./autogen.sh --prefix=/usr
#make install
#ldconfig
```

### 8.4 Build Weston

```
#git clone https://source.codeaurora.org/external/imx/weston-imx.git
#cd weston-imx
# git checkout weston-imx-8.0
meson build/ --prefix=/usr  -Dbackend-default=auto -Dbackend-rdp=false
-Dpipewire=false  -Dsimple-clients=all -Ddemo-clients=true
-Dcolor-management-colord=false -Drenderer-gl=true -Dbackend-fbdev=true
-Drenderer-g2d=true -Dbackend-headless=false -Dimxgpu=true
-Dbackend-drm=true -Dweston-launch=true -Dcolor-management-lcms=false
-Dopengl=true -Dpam=true -Dremoting=false -Dsystemd=true
-Dlauncher-logind=true -Dbackend-drm-screencast-vaapi=false
-Dbackend-wayland=false -Dimage-webp=false -Dbackend-x11=false
-Dxwayland=true
#cd build
#ninja -v -j 4 install
```

## 9. Build xwayland

### 9.1 build libepoxy

```
$ sudo cp -Pra   ~/L5.4.3-2.0.0/bld-xwayland/tmp/work/ aarch64-mx8mp-poky-linux/
mesa/2_19.1.6-r0/image /*   ubuntu-rootfs/
#wget   https://github.com/anholt/libepoxy/releases/download/1.5.3/libepoxy-1.5.3.tar.xz
#tar -zJvf libepoxy-1.5.3.tar.xz
# meson -Dtests=false -Degl=yes -Dglx=yes build/ --prefix=/usr
#ninja
#ninja install
```

### 9.2 build xserver

```
#apt-get install libxaw7-dev  libtinfo5 libxinerama-dev libxfont2
libxshmfence1  libxdamage-dev libx11-xcb-dev libxcb-glx0-dev
libxshmfence-dev  libxcb-dri2-0-dev libncurses5-dev libncursesw5-dev
libxxf86vm-dev  libxkbfile-dev  libxfont2  libssl-dev  libxfont-dev
xutils-dev  x11proto-xcmisc-dev x11proto-bigreqs-dev x11proto-randr-dev
x11proto-render-dev x11proto-fonts-dev x11proto-video-dev libpciaccess-
dev x11proto-composite-dev x11proto-record-dev x11proto-scrnsaver-dev
x11proto-resource-dev x11proto-xinerama-dev libxkbfile-dev libxfont-dev
libwayland-bin x11-xkb-utils

$ sudo cp -r ~/L5.4.3-2.0.0/bld-xwayland/tmp/work/ aarch64-mx8mp-poky-
linux/xserver-xorg/2_1.20.5-r0/xorg-server-1.20.5  ubuntu-rootfs/home/
```

```
# ./configure   --build=aarch64-poky-linux   --host=aarch64-poky-linux
--target=aarch64-poky-linux --prefix=/usr --disable-silent-rules    --disable-dependency-tracking
--with-libtool-sysroot=/    --with-fop=no       --with-pic    --disable-static      --disable-record
--disable-dmx              --disable-xnest       --enable-xvfb       --enable-composite
--without-dtrace      --with-int10=x86emu    --sysconfdir=/etc/X11     --localstatedir=/var
--with-xkb-output=/var/lib/xkb    --disable-static          --enable-dga --enable-dri --enable-dri2
--enable-dri3 --disable-glamor --enable-glx             --with-sha1=libcrypto --with-systemd-daemon
--enable-systemd-logind=yes --enable-config-udev --disable-libunwind --disable-xinerama
--without-xmlto --enable-xshmfence --enable-xwayland
#make -j8
#make install
```

## 10.(optional) build xterm

```
#wget https://invisible-island.net/datafiles/release/xterm.tar.gz
#tar -zxvf xterm.tar.gz
# cd xterm-353
# ./configure  --prefix=/usr  --disable-static    --disable-freetype  --
disable-silent-rules            --disable-dependency-tracking --disable-
imake --disable-rpath-hack  --disable-setuid
# make -j8
# make install
```

## 11.config sdma-firmware

11.1 modify dtb

```
arch/arm64/boot/dts/freescale/imx8mp.dtsi
            sdma1: dma-controller@30bd0000 {
                compatible = "fsl,imx8mq-sdma", "fsl,imx7d-sdma";
                reg = <0x30bd0000 0x10000>;
                interrupts = <GIC_SPI 2 IRQ_TYPE_LEVEL_HIGH>;
                clocks = <&clk IMX8MP_CLK_SDMA1_ROOT>,
                    <&clk IMX8MP_CLK_AHB>;
                clock-names = "ipg", "ahb";
                #dma-cells = <3>;
  -        fsl,sdma-ram-script-name = "imx/sdma/sdma-imx7d.bin";
  +        fsl,sdma-ram-script-name = "imx/sdma/sdma-imx7d_0.bin";
            };
```

```
Build linux-imx and replace imx8mp-evk.dtb
```

11.2 config systemctl

```
$ sudo cp -Pra  ~/L5.4.3-2.0.0/bld-xwayland/tmp/work/ all-poky-linux$ ls
firmware-imx/1_8.7-r0/image/*  ubuntu-rootfs/
```

In ubuntu filesystem

```
#ln -sf  /lib/systemd/system/sdma-firmware.service
/etc/systemd/system/multi-user.target.wants/sdma-firmware.service
#cp /lib/firmware/imx/sdma/sdma-imx7d.bin   /lib/firmware/imx/sdma/sdma-
imx7d_0.bin
```

## 12.Build gstreamer and plugins

12.0 Remove existing gstreamer packages

```
 #rm -rf /usr/lib/aarch64-linux-gnu/gstreamer-1.0
 #rm -rf /usr/lib/aarch64-linux-gnu/gstreamer1.0
 #rm -rf /usr/lib/aarch64-linux-gnu/libgst*
```

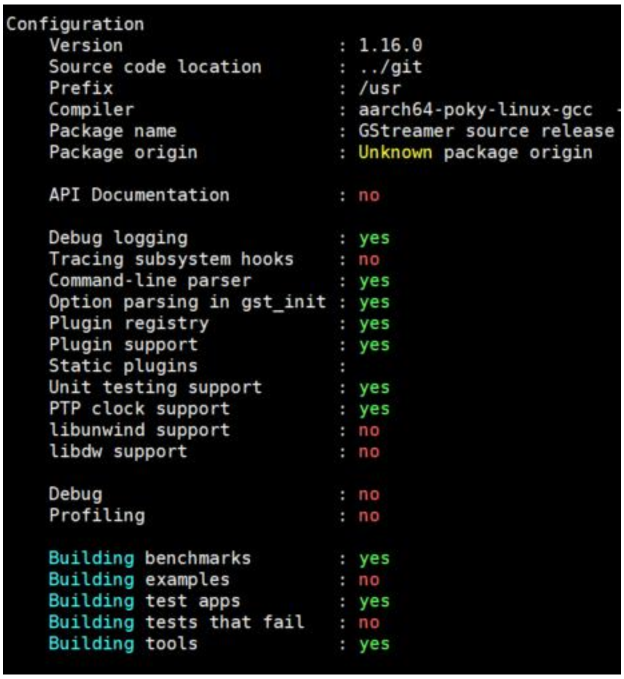
12.1 copy VPU driver

```
$sudo cp -Pra  ~/L5.4.3-2.0.0/bld-xwayland/tmp/work/ aarch64-mx8mp-poky-
linux/imx-vpu-hantro/1.17.0-r0/image /*     ubuntu-rootfs/
$ sudo cp -Pra  ~/L5.4.3-2.0.0/bld-xwayland/tmp/work/ aarch64-mx8mp-
poky-linux/imx-vpuwrap/4.5.4-r0/image /*    ubuntu-rootfs/
$ sudo cp -Pra  ~/L5.4.3-2.0.0/bld-xwayland/tmp/work/ aarch64-poky-
linux/imx-vpu-hantro-vc/1.0.0-r0/image /*   ubuntu-rootfs/
```

12.2 build Build gstreamer

```
#cd /home/user1/workplace
#mkdir gstreamer && cd gstreamer
#git clone https://source.codeaurora.org/external/imx/gstreamer.git
#cd gstreamer/
#git checkout MM_04.05.04_2002_L5.4.3
#./autogen.sh --disable-silent-rules --disable-dependency-tracking  --
disable-gtk-doc          --disable-dependency-tracking --disable-
docbook --disable-examples
--disable-debug --disable-gst-tracer-hooks --disable-tests --disable-
trace --disable-valgrind      --enable-nls --enable-introspection=no --
prefix=/usr/
#make && make install
#ldconfig
```

**Note**: Config log



```
Configuration
    Version                      : 1.16.0
    Source code location         : ../git
    Prefix                       : /usr
    Compiler                     : aarch64-poky-linux-gcc
    Package name                 : GStreamer source release
    Package origin               : Unknown package origin

    API Documentation            : no

    Debug logging                : yes
    Tracing subsystem hooks      : no
    Command-line parser          : yes
    Option parsing in gst_init   : yes
    Plugin registry              : yes
    Plugin support               : yes
    Static plugins               :
    Unit testing support         : yes
    PTP clock support            : yes
    libunwind support            : no
    libdw support                : no

    Debug                        : no
    Profiling                    : no

    Building benchmarks          : yes
    Building examples            : no
    Building test apps           : yes
    Building tests that fail     : no
    Building tools               : yes
```

## 12.3 Build gst-plugins-base

```
#cd /home/user1/workplace/gstreamer
#git clone https://source.codeaurora.org/external/imx/gst-plugins-
base.git
#cd gst-plugins-base/
#git checkout MM_04.05.04_2002_L5.4.3
#./autogen.sh --disable-silent-rules --disable-dependency-tracking
 --disable-gtk-doc --disable-examples --enable-zlib --enable-alsa
 --disable-cdparanoia --disable-debug --enable-ivorbis --enable-ogg
 --disable-opus --enable-orc --enable-pango --enable-theora
 --disable-valgrind --disable-libvisual --enable-vorbis --enable-x
 --enable-xvideo --enable-xshm --enable-nls --enable-introspection=no
 --enable-wayland --disable-opengl --prefix=/usr/
 CPPFLAGS="-I/usr/include/imx"
#make && make install
#ldconfig
```

## 12.4 Build gst-plugins-good

```
#cd /home/user1/workplace/gstreamer
#git clone https://source.codeaurora.org/external/imx/gst-plugins-
good.git
#cd gst-plugins-good
#git checkout MM_04.05.04_2002_L5.4.3
#./autogen.sh --disable-silent-rules  --disable-dependency-tracking
--disable-gtk-doc --disable-examples --enable-bz2 --enable-oss
--enable-zlib --disable-aalib --disable-aalibtest --disable-directsound
```

```
--disable-libcaca --disable-libdv --disable-oss4 --disable-osx_audio
--disable-osx_video --disable-shout2 --disable-sunaudio
--disable-waveform --enable-cairo --disable-debug --disable-dv1394
--enable-flac --enable-gdk_pixbuf --with-gudev --disable-jack
--enable-jpeg --enable-libpng --without-libv4l2 --enable-orc
--enable-pulse --enable-soup --enable-speex --enable-taglib
--enable-gst_v4l2 --enable-v4l2-probe --disable-valgrind --disable-vpx
--disable-wavpack --enable-x --enable-nls --disable-qt --prefix=/usr/
CPPFLAGS="-I/usr/include/imx"
#make && make install
#ldconfig
```

## 12.5 Build gst-plugins-bad

```
#cd /home/user1/workplace/gstreamer
#git clone https://source.codeaurora.org/external/imx/gst-plugins-
bad.git
#cd gst-plugins-bad
#git checkout MM_04.05.04_2002_L5.4.3
#./autogen.sh --disable-silent-rules --disable-dependency-tracking
--disable-gtk-doc --disable-examples --enable-decklink --enable-dvb
--enable-fbdev --enable-netsim --enable-shm --enable-vcd --disable-acm
--disable-android_media --disable-apexsink --disable-apple_media
--disable-avc --disable-bs2b --disable-chromaprint --disable-cocoa
--disable-daala --disable-dc1394 --disable-direct3d
--disable-directsound --disable-dts --disable-gme --disable-gsm
--disable-kate --disable-ladspa --disable-libde265 --disable-libvisual
--disable-linsys --disable-lv2 --disable-mimic --disable-mpeg2enc
--disable-mplex --disable-musepack --disable-nas --disable-nvenc
--disable-ofa --disable-openexr --disable-openh264 --disable-openjpeg
--disable-openni2 --disable-opensles --disable-pvr --disable-sdl
--disable-qt --disable-sdltest --disable-sndio --disable-soundtouch
--disable-spandsp --disable-spc --disable-teletextdec
--disable-timidity --disable-tinyalsa --disable-vdpau --enable-vulkan
--disable-wasapi --disable-wildmidi --disable-wininet --disable-winks
--disable-winscreencap --disable-x265 --disable-xvid --disable-zbar
--disable-assrender --enable-bluez --enable-bz2 --enable-curl
--enable-dash --disable-debug --disable-directfb --enable-dtls
--disable-egl --disable-faac --disable-faad --disable-flite
--disable-fluidsynth --enable-wayland --enable-gles2 --enable-egl
--disable-gtk3 --enable-hls --with-hls-crypto=openssl --disable-libmms
--disable-libssh2 --disable-modplug --enable-neon --disable-openal
--disable-opencv --disable-opengl --disable-opus --enable-orc
--disable-resindvd --enable-rsvg --disable-rtmp --enable-sbc
```

```
--disable-schro --enable-smoothstreaming --enable-sndfile
--disable-srtp --enable-uvch264 --disable-valgrind --disable-voaacenc
--disable-voamrwbenc --enable-webp --enable-nls
--enable-introspection=no --prefix=/usr/ CPPFLAGS="-I/usr/include/imx"
# make && make install
#ldconfig
```

## 12.6 Build imx-gst1.0-plugin

```
#cd /home/user1/workplace/gstreamer
#git clone https://source.codeaurora.org/external/imx/imx-gst1.0-
plugin.git
#cd imx-gst1.0-plugin
#git checkout MM_04.05.04_2002_L5.4.3
#./autogen.sh --disable-x11 --disable-static   --enable-overlaysink --
disable-wma8enc  --prefix=/usr/  PLATFORM=MX8 CPPFLAGS="-
I/usr/include/imx"
```

**Note:** config.log

```
369 Configure result:
370     Enabled features:
371         plugin: vpu_wrap
372         plugin: aiur
373         plugin: beep
374         plugin: v4lsink
375         plugin: overlaysink
376         imx2ddevice: g2d
377         libs: v4l2core
378     Disabled features:
379         plugin: imxmp3enc
380         plugin: wma8_enc
381         library: libX11
382         imx2ddevice: ipu
383         imx2ddevice: pxp
```

### 13.Exit simulate environment

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

### 14.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
```

### 15. 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
# weston --tty=1 --modules=xwayland.so
```

**Comment en_US.UTF-8 UTF-8**

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

## 16. some feature vertify

16.1 Ethernet

```
#apt-get install udhcpc udhcpd
#enable Etherent and get IP address.
#ifconfig eth0 up
#dhcpc -i eth0
```

16.2 xwayland

```
#export DISPLAY=:0
#xterm
```

16.3 camera

```
# gst-launch-1.0 v4l2src device=/dev/videoX ! video/x-raw,width=1280,height=720 ! autovideosink
```

16.4 BT-WIFI

Reference

https://confluence.sw.nxp.com/pages/viewpage.action?spaceKey=MADLinux&title=i.MX8MP+EVK+with+pre-Marvell+88w8997+test+steps+on+KRL-L5.4