## How to integrate a ssh server

Materials:

- i.MX8M Plus EVK Rev. A
- USB cable type-C
- USB cable type-B
- AC Adapter EA1045CR
- Micro SD (Optional)

Software:

- Yocto Project
- Mobaxterm Personal Edition v20.2 Build 4296

This test was done on an i.MX8M Plus EVK with Linux 5.10. Hardknott, also this was tested on the i.MX6 SabreSD.

So first of all, What is SSH? SSH, also known as Secure Shell or Secure Socket Shell, is a network protocol that gives users, particularly system administrators, a secure way to access a computer over an unsecured network.

SSH also refers to the suite of utilities that implement the SSH protocol. Secure Shell provides strong password authentication and public key authentication, as well as encrypted data communications between two computers connecting over an open network, such as the internet.

In our boards, by default the ssh package is include on the prebuilt images but the firewall will block the port you are trying to connect, so you need to build your own image with Yocto and add the package you will need:

- 1. Setup your build following the Yocto users guide.
- 2. Before you bitbake edit the local.conf file: nxf63675@lsv07091:~/imx-yocto-bsp/imx8mp-ddr/conf\$ ls bblayers.conf bblayers.conf.org local.conf local.conf.org local.conf.sample templateconf.cfg nxf63675@lsv07091:~/imx-yocto-bsp/imx8mp-ddr/conf\$ nano local.conf

Add the following lines to your custom build:

CORE\_IMAGE\_EXTRA\_INSTALL += "openssh" PACKAGE\_EXCLUDE += " packagegroup-core-ssh-dropbear" CORE\_IMAGE\_EXTRA\_INSTALL += "packagegroup-core-ssh-openssh"

```
GNU nano 2.9.3
```

```
ACHINE ??= 'imx8mpevk'
DISTR0 ?= 'fsl-imx-xwavland'
PACKAGE CLASSES ?= 'package rpm'
EXTRA IMAGE FEATURES ?= "debug-tweaks"
USER CLASSES ?= "buildstats image-mklibs image-prelink"
PATCHRESOLVE = "noop"
BB DISKMON DIRS ??= "\
    STOPTASKS, ${TMPDIR}, 1G, 100K \
    STOPTASKS, ${DL DIR}, 1G, 100K \
    STOPTASKS, ${SSTATE DIR}, 1G, 100K \
    STOPTASKS,/tmp,100M,100K \
    ABORT, ${TMPDIR}, 100M, 1K \
    ABORT, ${DL DIR}, 100M, 1K \
    ABORT, ${SSTATE DIR}, 100M, 1K \
    ABORT,/tmp,10M,1K"
PACKAGECONFIG append pn-gemu-system-native = " sdl"
CONF VERSION = "1"
DL DIR ?= "${BSPDIR}/downloads/"
ACCEPT FSL EULA = "1"
# Switch to Debian packaging and include package-management in the image
PACKAGE CLASSES = "package deb"
EXTRA IMAGE FEATURES += "package-management"
CORE_IMAGE_EXTRA INSTALL += "openssh"
PACKAGE_EXCLUDE += " packagegroup-core-ssh-dropbear"
CORE IMAGE EXTRA INSTALL += "packagegroup-core-ssh-openssh"
```

local.conf

- 3. Execute " bitbake imx-image-multimedia " and wait.
- 4. Deploy your image on an SD or eMMC.

These instructions apply to SD and MMC cards although for brevity, and usually only the SD card is listed.

For a Linux image to be able to run, four separate pieces are needed:

- Linux OS kernel image (zlmage/lmage)
- Device tree file (\*.dtb)
- Bootloader image
- Root file system (i.e., EXT4)

The Yocto Project build creates an SD card image that can be flashed directly. This is the simplest way to load everything needed onto the card with one command.

A .wic image contains all four images properly configured for an SD card. The release contains a pre-built .wic image that is built specifically for the one board configuration. It runs the Wayland graphical backend. It does not run on other boards unless U-Boot, the device tree, and rootfs are changed. When more flexibility is desired, the individual components can be loaded separately, and those instructions are included here as well. An SD card can be loaded with the individual components one-by-one or the .wic image can be loaded and the individual

parts can be overwritten with the specific components.

The rootfs on the default .wic image is limited to a bit less than 4 GB, but repartitioning and re-loading the rootfs can increase that to the size of the card. The rootfs can also be changed to specify the graphical backend that is used.

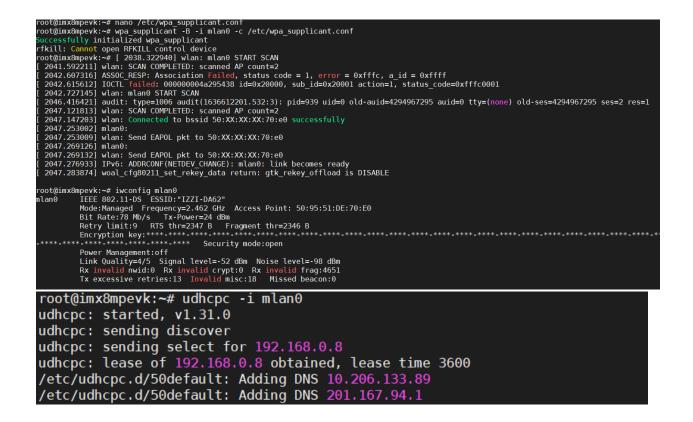
Carry out the following command to copy the SD card image to the SD/MMC card. Change sdx below to match the one used by the SD card.

\$ sudo dd if=<image name>.wic of=/dev/sdx bs=1M && sync

The entire contents of the SD card are replaced. If the SD card is larger than 4 GB, the additional space is not accessible.

5. Once your system booted, on Linux console configure your connection (Ethernet or wireless), in my case I connect the board wireless as the board has an 88W8997-based Wireless Modules.

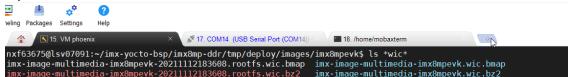
root@imx8mpevk:~# ifconfig mlan0 up	
root@imx8	8mpevk:~# ifconfig
eth0	Link encap:Ethernet HWaddr 00:04:9f:06:f6:72 UP BROADCAST MULTICAST MTU:1500 Metric:1 RX packets:0 errors:0 dropped:0 overruns:0 frame:0
	TX packets:0 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:1000 RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)
eth1	Link encap:Ethernet HWaddr 00:04:9f:06:f6:73 UP BROADCAST MULTICAST MTU:1500 Metric:1 RX packets:0 errors:0 dropped:0 overruns:0 frame:0 TX packets:0 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:1000 RX bytes:0 (0.0 B) TX bytes:0 (0.0 B) Interrupt:47
lo	Link encap:Local Loopback inet addr:127.0.0.1 Mask:255.0.0.0 inet6 addr: ::1/128 Scope:Host UP LOOPBACK RUNNING MTU:65536 Metric:1 RX packets:1306 errors:0 dropped:0 overruns:0 frame:0 TX packets:1306 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:1000 RX bytes:80840 (78.9 KiB) TX bytes:80840 (78.9 KiB)
mlan0	Link encap:Ethernet HWaddr 70:66:55:9b:36:03 UP BROADCAST MULTICAST MTU:1500 Metric:1 RX packets:0 errors:0 dropped:0 overruns:0 frame:0 TX packets:0 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:1000 RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)



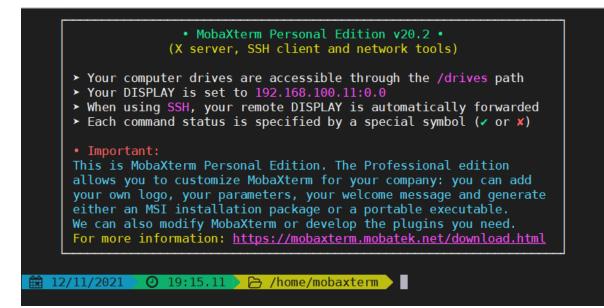
6. When you finish to configure your connection review the address that was assigned to you, in my case is the 192.168.0.8

root@imx8mpevk:~# ifconfig mlan0
mlan0 Link encap:Ethernet HWaddr 70:66:55:9b:36:03
inet addr:192.168.0.8 Bcast:192.168.0.255 Mask:255.255.255.0
inet6 addr: 2806:310:12c:821a:7266:55ff:fe9b:3603/64 Scope:Global
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:1518 errors:0 dropped:0 overruns:0 frame:0
TX packets:218 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:213019 (208.0 KiB) TX bytes:25139 (24.5 KiB)
root@imx8mpevk:~#
TX packets:218 <b>errors</b> :0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:1000 RX bytes:213019 (208.0 KiB) TX bytes:25139 (24.5 KiB)

7. Open a new window on mobaxterm:



8. You do not need to modify nothing on the configuration from mobaxterm, it will open a new window like below:



9. Enter the SSH command to connect the i.MX, the user in this case is root:

11/11/2021 O 20:05.48 ▷ ▷ /home/mobaxterm > ssh root@192.168.0.8 Warning: Permanently added '192.168.0.8' (RSA) to the list of known hosts. X11 forwarding request failed on channel 0 Last login: Thu Nov 11 05:56:13 2021 root@imx8mpevk:~#