∆VNET®

MaaXBoards

(iMX8M, iMX8mini and iMX8nano)

Yocto Kirkstone 5.15 User Manual

Contents

1 Introduction

This document demonstrate how to get started with a successful Yocto Kirkstone 5.15 build for the Avnet MaaXBoards(MaaXboard, MaaXboard mini and MaaXboard nano).

The layers to be use in this document will be all the layers described in the NXP documentation (IMX_YOCTO_PROJECT_USERS_GUIDE) and the meta-maaxboard layer which contains MaaXBoards specific recipes, such as firmware, connectivity for Bluetooth and Wi-Fi. You can find the meta-maaxboard layer on the next link:

https://github.com/HinoAM/meta-maaxboard

2 User configuration

User configuration helps define the build. Through user configuration, you can tell BitBake the target architecture for which you are building the image, where to store downloaded source, and other build properties.

MaaXborard	Description	Path
maaxboard-ddr4-2g-sdcard.conf	Machine configuration for maaxboard	meta-maaxboard/conf/machine at kirkstone · HinoAM/meta-maaxboard (github.com)
MaaXboard mini	**Description**	**Path**
maaxboard-mini-ddr4-2g-sdcard.conf	Machine configuration for maaxboard mini	meta-maaxboard/conf/machine at kirkstone · HinoAM/meta-maaxboard (github.com)
MaaXboard nano	**Description**	**Path**
maaxboard-nano-ddr4-2g-sdcard.conf	Machine configuration for maaxboard nano	meta-maaxboard/conf/machine at kirkstone · HinoAM/meta-maaxboard (github.com)
Common	**Description**	**Path**
local.conf	contains all the local user configurations for your build environment	meta-maaxboard/conf at kirkstone · HinoAM/meta-maaxboard (github.com)
bblayers.conf	defines layers, which are directory trees, traversed (or walked) by BitBake	meta-maaxboard/conf at kirkstone · HinoAM/meta-maaxboard (github.com)
layer.conf	holds configuration files for the layer	meta-maaxboard/conf at kirkstone · HinoAM/meta-maaxboard (github.com)

3 Booting Linux Operating System

To boot a Linux image on maaxboard, maaxboard mini and maaxboard nano the next files are needed to build the bootloader, kernel image and device tree file(.dtb):

3.1 u-boot-imx and device tree:

Enable hardware specific bootloader and device tree to maaxboard. Different from NXP evk.

MaaXborard	Description	Path
maaxboard.dts	device tree source for maaxboard	u-boot-imx/arch/arm/dts/
maaxboard.h	maaxbaord configuration board	u-boot-imx/include/configs/
maaxboard_defconfig	to retain a known set of bootloader configurations	u-boot-imx/configs/
uEnv-mq.txt	to make the kernel configurable	N/A
MaaXboard mini	**Description**	**Path**
maaxboard-mini.dts	device tree source for maaxboard mini	u-boot-imx/arch/arm/dts/
maaxboard_mini.h	maaxbaord mini configuration board	u-boot-imx/include/configs/
maaxboard_mini_defconfig	to retain a known set of bootloader configurations	u-boot-imx/configs/
uEnv-mini.txt	to make the kernel configurable	N/A
MaaXboard nano	**Description**	**Path**
maaxboard-nano.dts	device tree source for maaxboard mini	u-boot-imx/arch/arm/dts/
maaxboard-nano-extended-gpio.dtsi	to configure the gpios	u-boot-imx/arch/arm/dts/
maaxboard_nano.h	maaxbaord mini configuration board	u-boot-imx/include/configs/
maaxboard_nano_defconfig	to retain a known set of bootloader configurations	u-boot-imx/configs/
uEnv-nano.txt	to make the kernel configurable	N/A
Common	**Description**	**Path**
spl.c	Secondary program loader	u-boot-imx/board/embest/
ddr4_timing.c	DDR4 configuration	u-boot-imx/board/embest/
embest_env.h	to make the kernel configurable	u-boot-imx/include/configs/
embest_fdt_overlay.h	to make the kernel configurable	u-boot-imx/include/configs/
imx8m	add config board definitions	u-boot-imx/arch/arm/mach-imx/imx8m/

You can find all the mentioned files above in github:

https://github.com/HinoAM/meta-maaxboard/tree/kirkstone/recipes-bsp/u-boot/files

3.2 linux-imx and device tree:

Enable hardware specific Linux kernel and device tree to maaxboard. Different from NXP evk.

MaaXborard	Description	Path
maaxboard-dcss-hdmi.dts	enable dcss hdmi maaxboard	linux-imx/arch/arm64/boot/dts/freescale/
maaxboard-dcss-mipi.dts	enable dcss mipi maaxboard	linux-imx/arch/arm64/boot/dts/freescale/
maaxboard-lcdif-mipi.dts	enable lcd mipi maaxboard	linux-imx/arch/arm64/boot/dts/freescale/
maaxboard-dual-display.dts	enable dual display maaxboard	linux-imx/arch/arm64/boot/dts/freescale/
maaxboard-usb0-device.dts	enable usb maaxboard	linux-imx/arch/arm64/boot/dts/freescale/
maaxboard-dsi-common.dtsi	display serial interface maaxboard	linux-imx/arch/arm64/boot/dts/freescale/
maaxboard-extended-gpio.dtsi	gpios functionality maaxboard	linux-imx/arch/arm64/boot/dts/freescale/
maaxboard.dtsi	device tree source include, SoC level definitions maaxboard	linux-imx/arch/arm64/boot/dts/freescale/
maaxboard_defconfig	to retain a known set of kernel configurations	linux-imx/arch/arm64/configs/
MaaXboard mini	**Description**	**Path**
maaxboard-mini-device.dts	enable usb maaxboard mini	linux-imx/arch/arm64/boot/dts/freescale/
maaxboard-mini-mipi.dts	enable mipi dsi maaxboard mini	linux-imx/arch/arm64/boot/dts/freescale/
maaxboard-mini-extended-gpio.dtsi	gpios functionality maaxboard mini	linux-imx/arch/arm64/boot/dts/freescale/
maaxboard-mini.dtsi	device tree source include, SoC level definitions maaxboard mini	linux-imx/arch/arm64/boot/dts/freescale/
maaxboard_mini_defconfig	to retain a known set of kernel configurations	linux-imx/arch/arm64/configs/
MaaXboard nano	**Description**	**Path**
maaxboard-nano-mipi.dts	enable mipi dsi maaxboard nano	linux-imx/arch/arm64/boot/dts/freescale/
maaxboard-nano-extended-gpio.dtsi	gpios functionality maaxboard nano	linux-imx/arch/arm64/boot/dts/freescale/
maaxboard-nano.dtsi	device tree source include, SoC level definitions maaxboard nano	linux-imx/arch/arm64/boot/dts/freescale/
maaxboard_nano_defconfig	to retain a known set of kernel configurations	linux-imx/arch/arm64/configs/

You can find all the mentioned files above in github:

https://github.com/HinoAM/meta-maaxboard/tree/kirkstone/recipes-kernel/linux/files

4 Host setup

This chapter will introduce the setup host and Yocto project build as described in NXP documentation.

4.1 Operating system

Yocto project supports very specific Linux distributions and version:

https://docs.yoctoproject.org/ref-manual/system-requirements.html#supported-linux-distributions

But for these instructions assume that you are using one of these versions of standard desktop Ubuntu:

- Ubuntu 18.04 LTS
- Ubuntu 20.04 LTS
- Ubuntu 22.04 LTS

These instructions also assume that you are using the default Bash shell that comes with Ubuntu.

4.2 Host packages

Install the following packages

```
~$ sudo apt-get update && sudo apt-get install -y gawk wget git-core diffstat unzip texinfo gcc-multilib
build-essential chrpath socat libsdl1.2-dev xterm sed cvs subversion coreutils texi2html docbook-utils
python-pysqlite2 help2man make gcc g++ desktop-file-utils libgl1-mesa-dev libglu1-mesa-dev mercurial
autoconf automake groff curl lzop asciidoc u-boot-tools cpio sudo locales
```

Repo is a tool built on top of Git that makes it easier to manage projects that contain multiple repositories, which do not need to be on the same server

```
~$ mkdir ~/bin (this step may not be needed if the bin folder already exists)
~$ curl http://commondatastorage.googleapis.com/git-repo-downloads/repo > /usr/bin/repo
~$ chmod a+x ~/bin/repo
~$ export PATH="${HOME}/bin:${PATH}"
```

5 Yocto setup

This section will introduce all the needed setup we will need in order to build all the MaaXboard versions recipes we choose.

5.1 Configure git

List present configuration

```
~$ git config --list
```

If the above command demonstrates that you already have a username and email configured, then you can skip the remainder of this section and continue with configuring Repo.

Configure Git

```
~$ git config --global user.name "Firstname Lastname"
~$ git config --global user.email "EmailAddress@Domain.com"
```

Confirm that Git is configured properly

```
~$ git config --list
```

You should see at least these two lines with your name and email address

```
user.name=Firstname Lastname
user.email=EmailAddress@Domain.com
```

5.2 Download meta layers from NXP

The very first thing you need to do is to create your work directory:

Workdirectory/maaxboard/

Create a new directory called imx-yocto-bsp. We'll be downloading the board support package (BSP) and other meta layers here:

```
Workdirectory/maaxboard$ mkdir imx-yocto-bsp
Workdirectory/maaxboard$ cd imx-yocto-bsp
Workdirectory/maaxboard/imx-yocto-bsp$
```

Install the i.MX BSP repo and download the Yocto Project Layers. I'll be using kirkstone here:

```
Workdirectory/maaxboard/imx-yocto-bsp$ repo init -u https://github.com/nxp-imx/imx-manifest -b imx-linux-
kirkstone -m imx-5.15.71-2.2.0.xml
Workdirectory/maaxboard/imx-yocto-bsp$ repo sync
```

You should now see the following folders / files

```
Workdirectory/maaxboard/imx-yocto-bsp$ ls
imx-setup-release.sh
README
README-IMXBSP
setup-environment
sources
```

5.3 Setup the Yocto build environment

i.MX provides a script, imx-setup-release.sh, that simplifies the setup for i.MX machines. To use the script, the name of the specific machine to be built for needs to be specified as well as the desired graphical backend. The script sets up a directory and the configuration files for the specified machine and backend.

Note: Each build folder can only support a single DISTRO

Setup the Yocto build environment

```
Workdirectory/maaxboard/imx-yocto-bsp$ MACHINE=imx8mqevk DISTRO=fsl-imx-wayland source imx-setup-
release.sh -b maaxboard-wayland
```

This operation will generate two conf files under the path maaxboard-wayland

```
Workdirectory/maaxboard/imx-yocto-bsp/maaxboard-wayland/conf$ ls
local.conf
bblayer.conf
```

5.4 Clone meta-maaxboard git repository

All the needed files to build the kernel and universal bootloader can be downloaded from github:

[HinoAM/meta-maaxboard: Yocto meta-layer for MaaXBoard/Mini/Nano (github.com)](github.com)

Go to sources folder

```
Workdirectory/maaxboard/imx-yocto-bsp$ cd sources
```

Clone the meta-maaxboard layer

```
Workdirectory/maaxboard/imx-yocto-bsp/sources$ git clone https://github.com/HinoAM/meta-maaxboard.git
Workdirectory/maaxboard/imx-yocto-bsp/sources$ cd meta-maaxboard
Workdirectory/maaxboard/imx-yocto-bsp/sources/meta-maaxboard$ git checkout kirkstone
```

At the end of this steps, you will have some repository like below:

```
Workdirectory/maaxboard/imx-yocto-bsp/sources/meta-maaxboard$ ls
conf    README.md    recipes-connectivity    replace-conf.sh
docs    recipes-bsp    recipes-kernel
```

You will need to modify local.conf and bblayer.conf (created before by the imx-setup-release.sh) files according to your settings. Inside meta-maaxboard folder you will find a script that you can run and replace the files. You need to put the folder destination and the type of maaxboard you want, for example:

```
Workdirectory/maaxboard/imx-yocto-bsp/sources/meta-maaxboard$ ./replace-conf.sh maaxboard-wayland
maaxboard
bblayers.conf maaxboards copied ... [OK]
local.conf maaxboard copied ... [OK]
Done
```

5.5 Soruce files and bbappend files

Inside of the meta-maaxboard folder, you will find the folder structure and files needed to replace the files in kernel and universal bootloader, this is possible with the bbappend files, after bitbake downloaded the source code from the repositories the bbappend files will replace the needed source files in order to build the kernel and universal bootloader for MaaXboards.

u-boot path:

```
Workdirectory/maaxboard/imx-yocto-bsp/sources/meta-maaxboard/recipes-bsp/u-boot$ ls
files
u-boot-imx_2022.04.bbappend
uenv_1.1.bb
```

linux path:

```
Workdirectory/maaxboard/imx-yocto-bsp/sources/meta-maaxboard/recipes-kernel/linux$ ls
files
linux-imx_5.15.bbappend
```

5.6 Build Yocto

The Yocto Project provides some images that are available on different layers. Poky provides some images, meta-freescale and meta-freescale-distro provide others, and additional image recipes are provided in the meta-imx layer. The following table lists various key images, their contents, and the layers that provide the image.

For all maaxboard versions we will be using the imx-image-full, is highly recommend fetching the sources first:

```
$ bitbake imx-image-full --runall fetch
```

Then run bitbake:

```
$ bitbake imx-image-full
```

5.7 Build output

Once it's done building, the build output is located under path:

```
Workdirectory/maaxboard/imx-yocto-bsp/maaxboard-wayland/tmp/deploy/images/maaxboard-ddr4-2g-sdcard$
```

You will find a wic file which is the needed file to flash into the SD card.

5.8 Returning to this project at later date

Bitbake will not run if the environment is not configured. If you close the present shell (terminal) then you will lose the environment set up by imx-setup-release.sh. To set up our environment again:

```
Workdirectory/maaxboard/imx-yocto-bsp$ source setup-environment maaxboard-wayland
```

5.9 Wi-Fi and Bluetooth

Maaxboard include an Azure Wave which include the wi-fi and ble, the embedded zip already has the needed files the enable the features, the firmware comes in binary and blob we just need to tell bitbake where to put them in order to use them.

linux firmware path:

```
Workdirectory/maaxboard/imx-yocto-bsp/sources/meta-maaxboard/recipes-kernel/linux-firmware$ ls
files
linux-firmware_%.bbappend
```