MaaXBoards

(iMX8M, iMX8mini and iMX8nano)

Yocto Honister 5.15

User Manual iMX

Desktop PoC

Revision History

Rev.	Description	Author	Date
V0.0	Initial version	Mitsuki	12142022

Contents

Images

Chapter 1 Introduction

1.1 Brief Introduction

The i.MX desktop build uses a Yocto-based meta layer to generate a desktop Proof of Concept (PoC) image. It works together with i.MX release layer (meta-imx). It reuses the Linux BSP release framework to manage and generate the U-Boot bootloader, Linux kernel image, and i.MX root file system in desktop image build.

Information:

- Built using Yocto Project
- Updated quarterly
- All source code is located on codearuror.org
- This is a Linux distribution that uses the Canonical Ubuntu aarch64 repositories
- The bootloader and kernel are identical to the standard Yocto project builds
- The root filesystem and userspace toolchain are based on Ubuntu

You can download the documentation from NXP webpage:

https://www.nxp.com/webapp/Download?colCode=L5.15.32_2.0.0_Desktop_POC_Docs

Chapter 2 Setup project

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

2.1 Host setup

2.1.1 Operating system

These instructions assume that you are using one of these two versions of standard desktop Ubuntu:

- Ubuntu 20.04 LTS "Bionic Beaver" is the officially supported version for Yocto honister.

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

2.1.2 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 coreuti
ls texi2html docbook-utils python-pysqlite2 help2man make gcc g++ desktop-file-utils libg
l1-mesa-dev libglu1-mesa-dev mercurial autoconf automake groff curl lzop asciidoc u-boot-
tools cpio sudo locales
```

2.1.3 Install the repo utility

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)

chmod a+x ~/bin/repo
curl http://commondatastorage.googleapis.com/git-repo-downloads/repo > /usr/bin/repo

export PATH="${HOME}/bin:${PATH}"
```

2.2 Yocto setup

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

2.2.1 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
```

2.2.2 Download meta layers from NXP desktop

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

```
mkdir imx-yocto-bsp-desktop
sudo chown -R [USERNAME] imx-yocto-bsp-desktop
cd imx-yocto-bsp
```

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

```
$ repo init -u https://source.codeaurora.org/external/imx/imx-manifest -b imx-linux-honist
er -m imx-5.15.5-1.0.0_desktop.xml
repo sync
```

You should now see the following folders / files

```
imx-setup-release.sh
README
README-IMXBSP
setup-environment
sources

imx-setup-desktop.sh
```

2.2.3 Create 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)

Go to sources folder

```
$ cd sources
```

Clone the meta-maaxboard layer

```
$ git clone https://github.com/HinoAM/meta-maaxboard.git

$ cd meta-maaxboard

$ git checkout honister
```

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

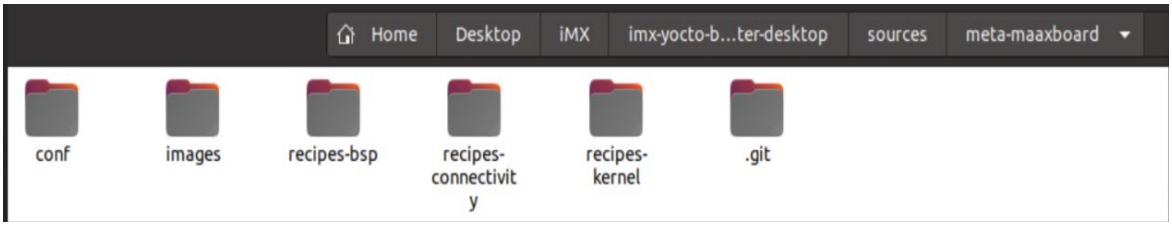

Image 1.1 Final meta-maaxboard layer

2.2.4 Setup the yocto build enviroment

First some Yocto definitions

MACHINE=<machine>

Use EVK names for <machine> listed in Yocto Project Users Guide, section 5.1 "Build configurations"

DISTRO=fsl-imx-<backend> where <backend> refers to the graphics type:

xwayland = Wayland with X11 support - default distro

wayland = Wayland only

fb = Framebuffer (not supported for imx8)

Note: Each build folder can only support a single DISTRO

Setup the Yocto build environment

```
mkdir maaxboard-desktop
MACHINE=imx8mqevk DISTRO=imx-desktop-xwayland source imx-setup-desktop.sh -b maaxboard-desktop
```

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

- local.conf
- bblayers.conf

You will need to modify both conf files according to your settings. Under the 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 desktop you want, for example:

```
./replace-conf.sh maaxboard-desktop maaxboard-desktop
bblayers.conf for maaxboard-desktop copied … [OK]
local.conf maaxboard desktop copied … [OK]
Done
```

Options are:

Maaxboard = maaxboard-desktop

Maaxboard mini = maaxboard-mini-desktop

Maaxboard nano = maaxboard-nano-desktop

2.2.5 Build Yocto

From maaxboard-desktop directory, run bitbake:

```
bitbake imx-image-desktop
```

2.2.6 Build output

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

imx-yocto-bsp-desktop/maaxboard-desktop/build/tmp/deploy/images/maaxboard-ddr4-2g-sdcard

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

2.2.7 Returning to this project at a 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:

```
$ source setup-environment maaxboard-desktop
```