User Guide

Porting Guide for NXP Based Modules

How to Download and Build NXP Wi-Fi Drivers - Rev. 1.0

Table of Contents

Figures

Tables

About This Document

This document describes the steps to download and build NXP Wi-Fi driver for Murata NXP-based modules.

Audience & Purpose

This document is targeted towards system developers of Wi-Fi/Bluetooth solutions on Linux systems, using Murata modules based on NXP-chipset.

Document Conventions

Table 1 describes the document conventions.

Table 1: Document Conventions

Conventions	Description
✋	**Warning Note** Indicates very important note. Users are strongly recommended to review.
ℹ	**Info Note** Intended for informational purposes. Users should review.
▙	**Menu Reference** Indicates menu navigation instructions. **Example**: Insert→Tables→Quick Tables→Save Selection to Gallery ▙
↗	**External Hyperlink** This symbol indicates a hyperlink to an external document or website. **Example**: Murata ↗ Click on the text to open the external link.
↙	**Internal Hyperlink** This symbol indicates a hyperlink within the document. **Example**: Overview ↙ Click on the text to open the link.
`Console input/output or code snippet`	**Console I/O or Code Snippet** This text ***Style*** denotes console input/output or a code snippet.

`# Console I/O comment`
`// Code snippet comment`
	Console I/O or Code Snippet Comment
	This text ***Style*** denotes a console input/output or code snippet comment.
	• Console I/O comment (preceded by “#”) is for informational purposes only and does not denote actual console input/output.
	• Code Snippet comment (preceded by “//”) may exist in the original code.

1 Overview

This document describes how to download NXP Wi-Fi driver source and firmware binaries from GitHub.

> The URL to download the Wi-Fi driver source has changed from CodeAurora to GitHub in September 2022.

2 Installing Git

To download the NXP Wi-Fi driver source and firmware binaries, the user must first install the Git tool on the development host (Linux or Windows PCs).

This section describes the steps for Git installation. If Git is already installed, make sure to use the latest version.

2.1 Installing Git on Linux

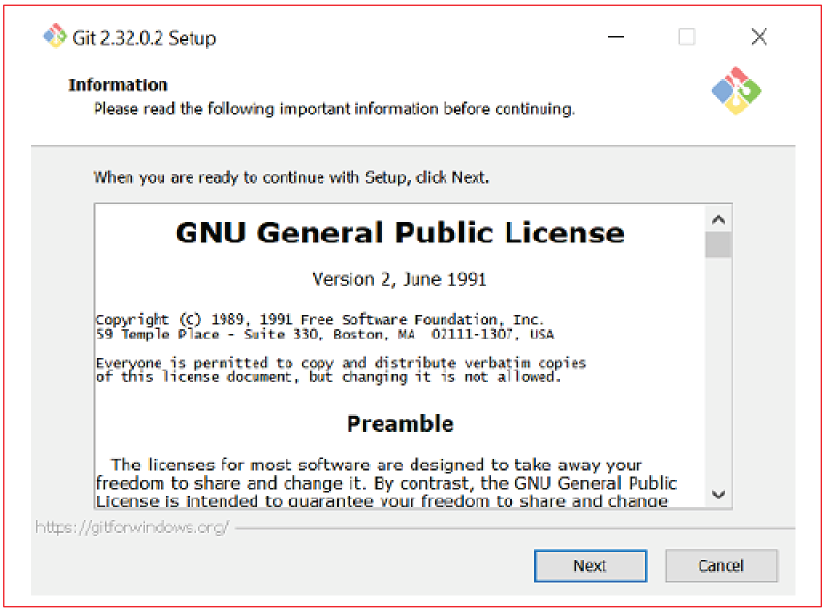
Use following command to install Git on the Debian-based distribution.

```
# sudo apt-get install git
```

2.2 Installing Git on Windows

1. Download Git application from: Git Download for Windows.
2. Double-click the executable file (.exe) .
3. Keep the default options and click **Next** to initiate the installation (**Figure 1**).

Figure 1: Git Installation on Windows



Information

Please read the following important information before continuing.

When you are ready to continue with Setup, click Next.

GNU General Public License

Version 2, June 1991

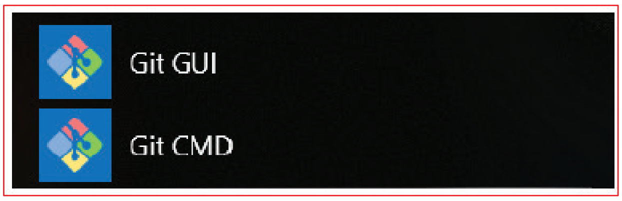
```
Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place - Suite 330, Boston, MA  02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.
```

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change

4. When the installation is complete, Git CMD and Git GUI are available in Windows Explorer as shown in **Figure 2**.

Figure 2: Git CMD and Git GUI in Windows Explorer



3 Downloading NXP Wi-Fi Driver Source

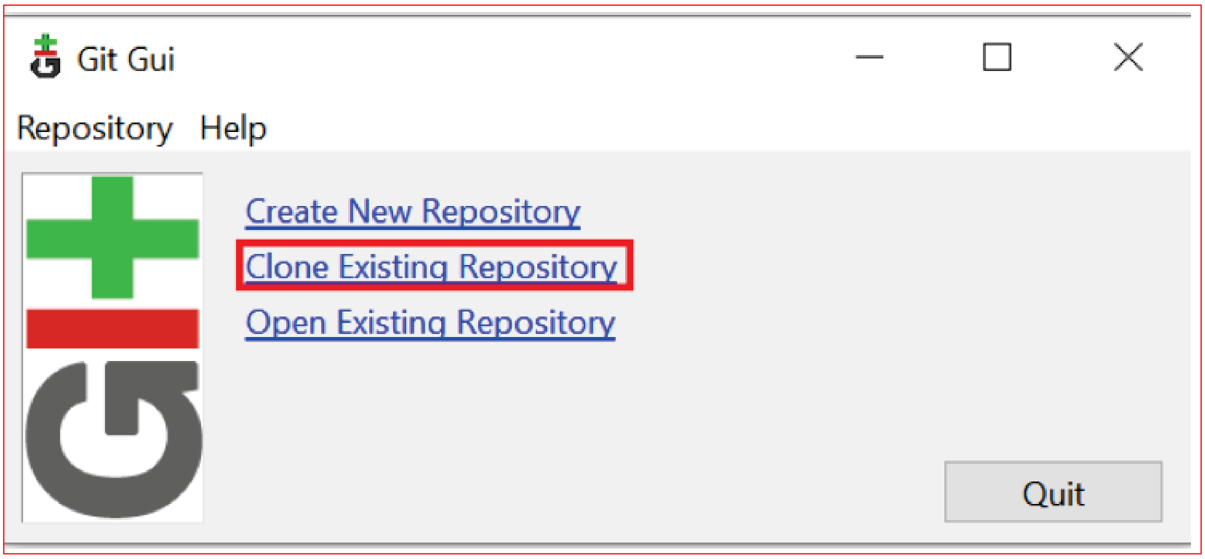
This section shows how to use the Git tool to download NXP Wi-Fi driver source on your Windows or Linux Ubuntu system.

HCI Bluetooth UART driver is open source and available in the Linux kernel source. The Bluetooth features are validated with the BlueZ version 5.56.

3.1 Using Git GUI on Windows

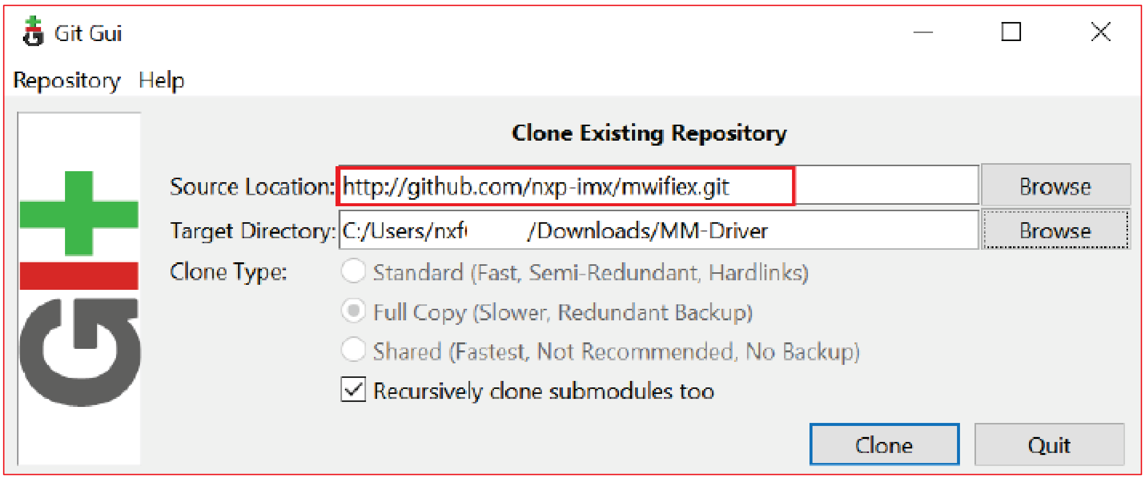
1. Launch Git GUI.
2. Click **Clone Existing Repository**.

Figure 3: Clone Existing Repository



3. Capture the address to the source location: http://github.com/nxp-imx/mwifiex.git.
4. Define the **target directory** for NXP Wi-Fi driver source to download.

Figure 4: Capturing the Source Location and Target Directory



Source Location: http://github.com/nxp-imx/mwifiex.git

Target Directory: C:/Users/nxf /Downloads/MM-Driver

Clone Type:

○ Standard (Fast, Semi-Redundant, Hardlinks)

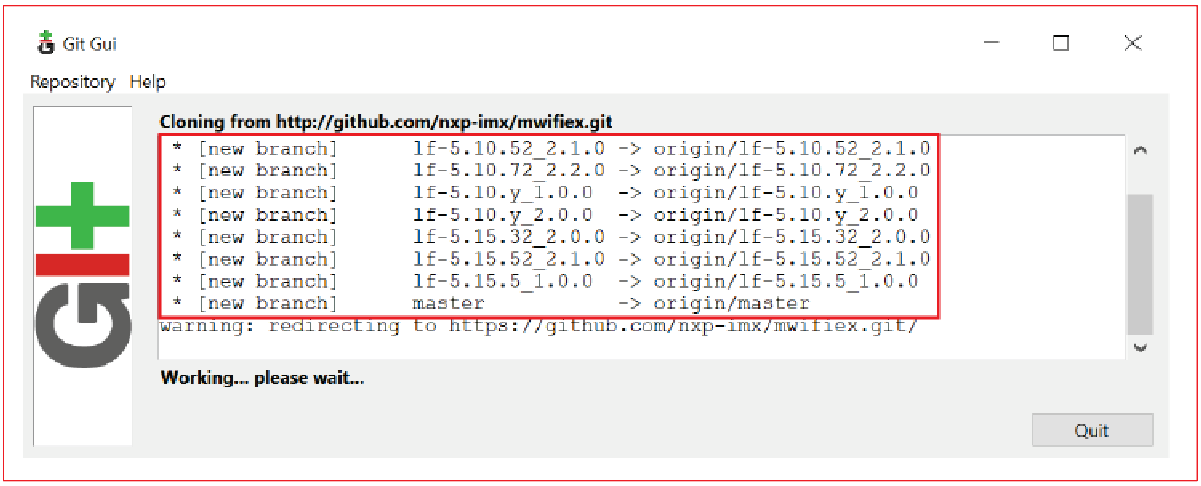
● Full Copy (Slower, Redundant Backup)

○ Shared (Fastest, Not Recommended, No Backup)

☑ Recursively clone submodules too

5. To initiate the download of NXP Wi-Fi driver source, click the Clone button.

Figure 5: NXP Wi-Fi Driver Source Download in Progress



Cloning from http://github.com/nxp-imx/mwifiex.git

```
 * [new branch]      lf-5.10.52_2.1.0 -> origin/lf-5.10.52_2.1.0
 * [new branch]      lf-5.10.72_2.2.0 -> origin/lf-5.10.72_2.2.0
 * [new branch]      lf-5.10.y_1.0.0  -> origin/lf-5.10.y_1.0.0
 * [new branch]      lf-5.10.y_2.0.0  -> origin/lf-5.10.y_2.0.0
 * [new branch]      lf-5.15.32_2.0.0 -> origin/lf-5.15.32_2.0.0
 * [new branch]      lf-5.15.52_2.1.0 -> origin/lf-5.15.52_2.1.0
 * [new branch]      lf-5.15.5_1.0.0  -> origin/lf-5.15.5_1.0.0
 * [new branch]      master           -> origin/master
warning: redirecting to https://github.com/nxp-imx/mwifiex.git/
```

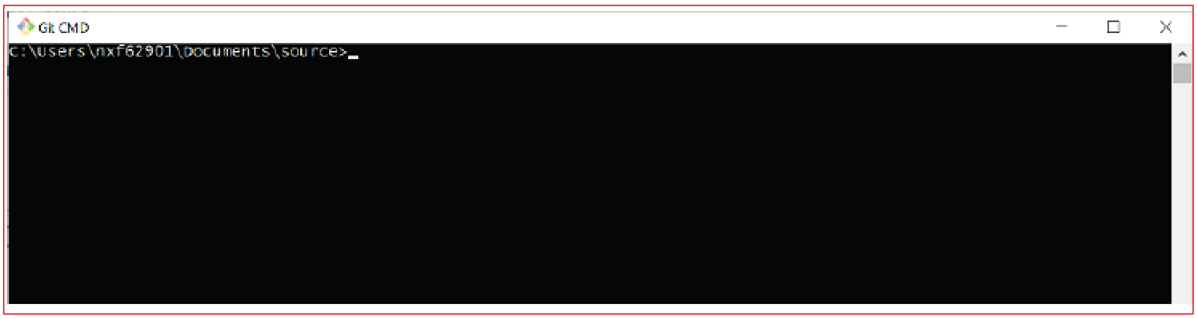
Working... please wait...

6. Once the download is complete, the driver source is available in the target directory.

3.2 Using Git CMD on Windows

1. Launch Git CMD from Windows Explorer.
2. Go to your Windows home directory.
3. Use *mkdir* command to create the source directory to store NXP Wi-Fi driver source.
4. Use *cd* command to go to source directory.

Figure 6: User Home Directory in Git CMD

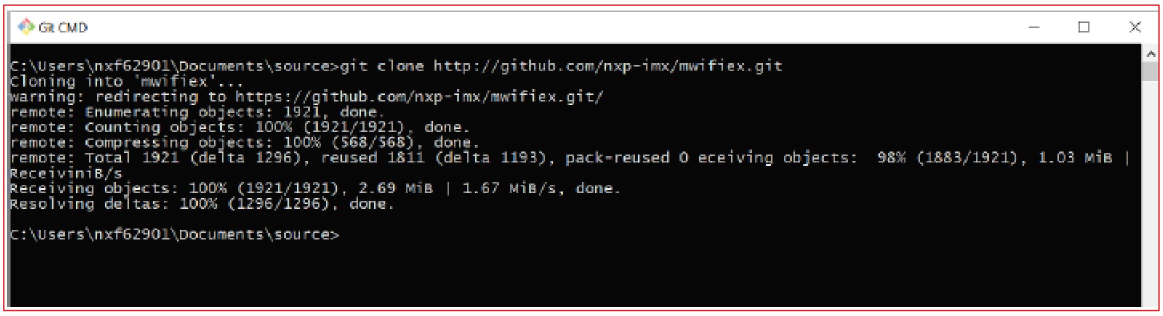


```
C:\Users\nxf62901\Documents\source>_
```

5. Use the “git clone” command to download NXP Wi-Fi driver source.

```
git clone http://github.com/nxp-imx/mwifiex.git
```

Figure 7: git clone Command Output



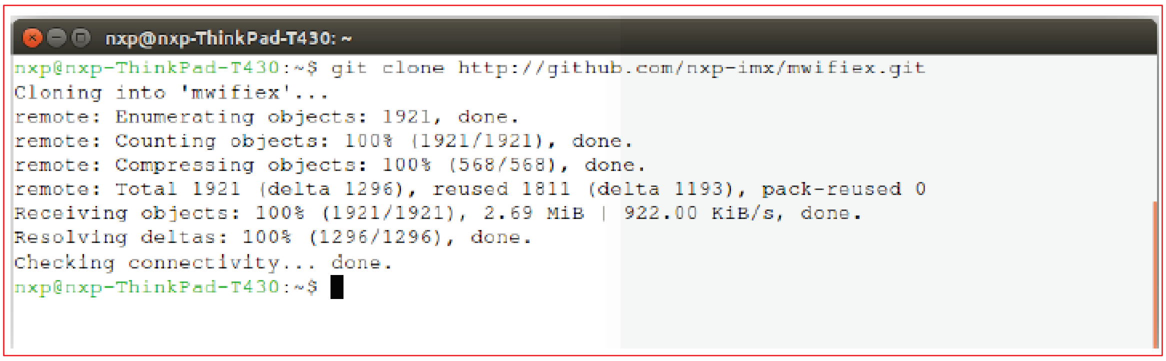
```
C:\Users\nxf62901\Documents\source>git clone http://github.com/nxp-imx/mwifiex.git
Cloning into 'mwifiex'...
warning: redirecting to https://github.com/nxp-imx/mwifiex.git/
remote: Enumerating objects: 1921, done.
remote: Counting objects: 100% (1921/1921), done.
remote: Compressing objects: 100% (568/568), done.
remote: Total 1921 (delta 1296), reused 1811 (delta 1193), pack-reused 0 eceiving objects:  98% (1883/1921), 1.03 MiB |
ReceiviniB/s
Receiving objects: 100% (1921/1921), 2.69 MiB | 1.67 MiB/s, done.
Resolving deltas: 100% (1296/1296), done.

C:\Users\nxf62901\Documents\source>
```

6. Once the download is completed, NXP Wi-Fi driver source is available in the (*source*) directory.

3.3 Using Git CMD on Linux

1. Launch the Linux terminal (CTRL+ALT+T).

Figure 8: Git CMD Prompt on Linux

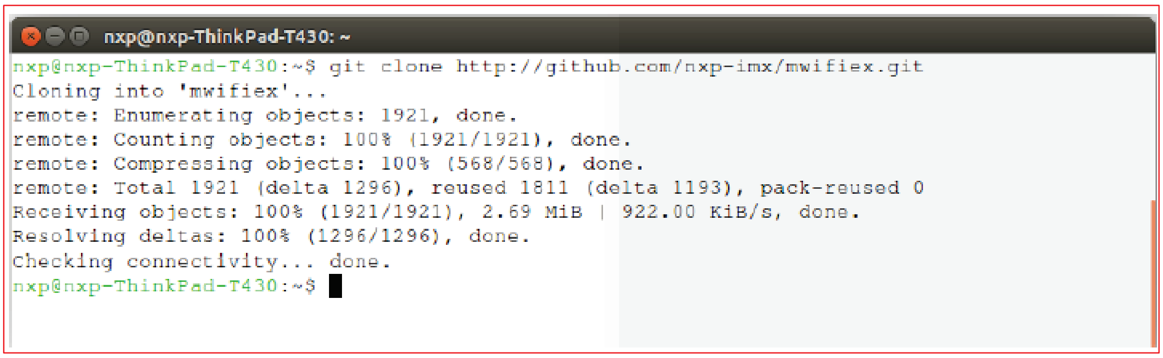


```
nxp@nxp-ThinkPad-T430:~$ git clone http://github.com/nxp-imx/mwifiex.git
Cloning into 'mwifiex'...
remote: Enumerating objects: 1921, done.
remote: Counting objects: 100% (1921/1921), done.
remote: Compressing objects: 100% (568/568), done.
remote: Total 1921 (delta 1296), reused 1811 (delta 1193), pack-reused 0
Receiving objects: 100% (1921/1921), 2.69 MiB | 922.00 KiB/s, done.
Resolving deltas: 100% (1296/1296), done.
Checking connectivity... done.
nxp@nxp-ThinkPad-T430:~$ 
```

2. Go to your Linux home directory.
3. Use *mkdir* command to create the source directory to store NXP Wi-Fi driver source.
4. Use *cd* command to go to source directory.
5. Use *git clone* command to download the Wi-Fi driver source.

```
$ git clone http://github.com/nxp-imx/mwifiex.git
```

Figure 9: git-clone Command Output in Linux



```
nxp@nxp-ThinkPad-T430:~$ git clone http://github.com/nxp-imx/mwifiex.git
Cloning into 'mwifiex'...
remote: Enumerating objects: 1921, done.
remote: Counting objects: 100% (1921/1921), done.
remote: Compressing objects: 100% (568/568), done.
remote: Total 1921 (delta 1296), reused 1811 (delta 1193), pack-reused 0
Receiving objects: 100% (1921/1921), 2.69 MiB | 922.00 KiB/s, done.
Resolving deltas: 100% (1296/1296), done.
Checking connectivity... done.
nxp@nxp-ThinkPad-T430:~$
```

6. Once the download completed, NXP Wi-Fi driver source is available in the source directory.

4 Downloading Firmware Binaries

The firmware binary is available for download in GitHub. You can use either the command line or GitHub website to download the binaries.

4.1 Download Using Command Line

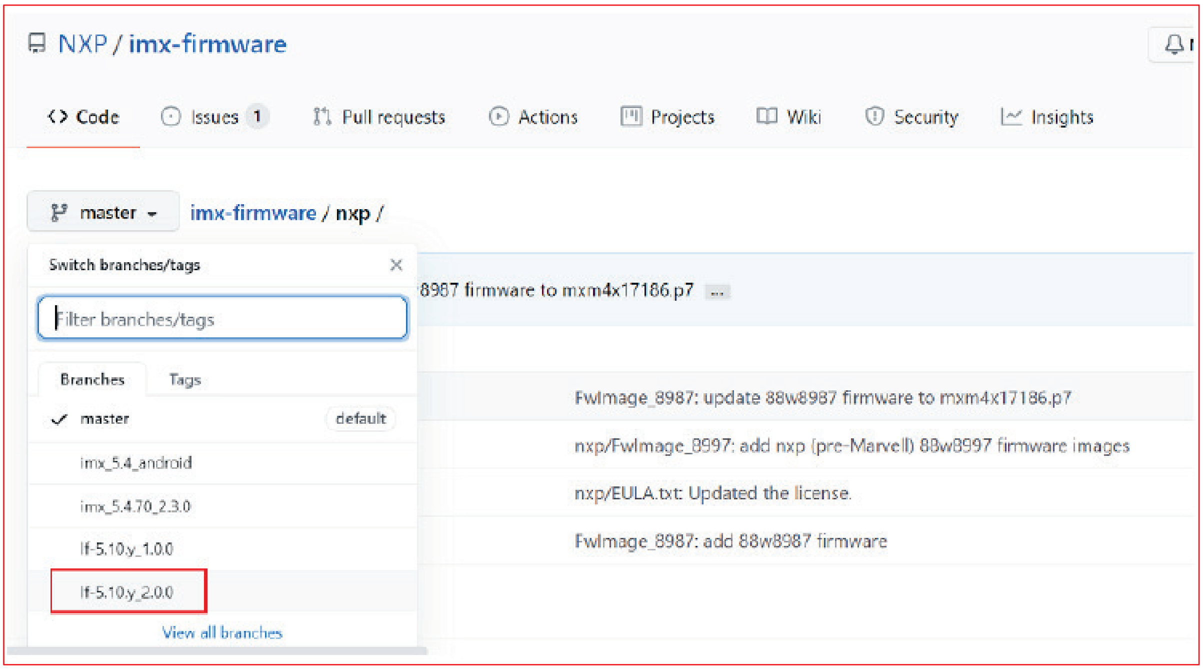
1. Follow the steps in section Using Git GUI on Windows or Using Git CMD on Windows for Windows and Using Git CMD on Linux for Linux to open the Git tool.
2. Issue the *git clone* command to transfer the firmware binaries, and go to the directory with the new content:

```
# git clone https://github.com/NXP/imx-firmware.git
# cd imx-firmware/nxp/
```

4.2 Download Using GitHub Website

1. Open a web browser and access NXP/imx Firmware.

Figure 10: NXP/imx-firmware Page



8987 firmware to mxm4x17186.p7 ...

FwImage_8987: update 88w8987 firmware to mxm4x17186.p7

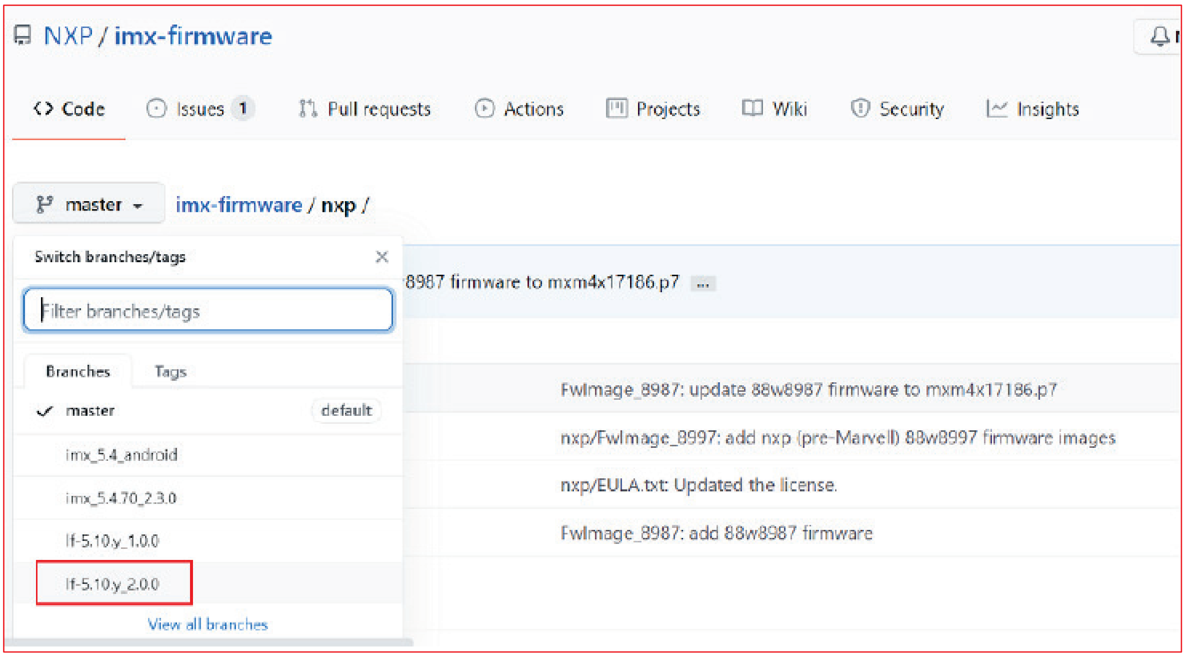
nxp/FwImage_8997: add nxp (pre-Marvell) 88w8997 firmware images

nxp/EULA.txt: Updated the license.

FwImage_8987: add 88w8987 firmware

2. Select the branch.

Figure 11: Selecting the Branch



8987 firmware to mxm4x17186.p7 ...

FwImage_8987: update 88w8987 firmware to mxm4x17186.p7

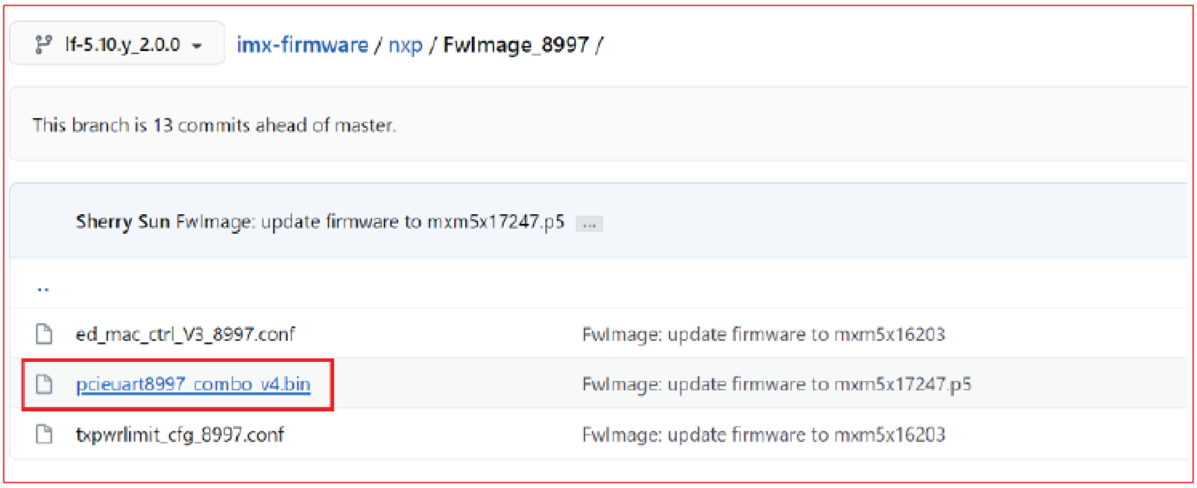
nxp/FwImage_8997: add nxp (pre-Marvell) 88w8997 firmware images

nxp/EULA.txt: Updated the license.

FwImage_8987: add 88w8987 firmware

3. Select the product for the firmware binary. **Figure 12** shows the example of 88W8997, for Murata module Type 1ZM.

Figure 12: Selecting the Product



imx-firmware / nxp / FwImage_8997 /

This branch is 13 commits ahead of master.

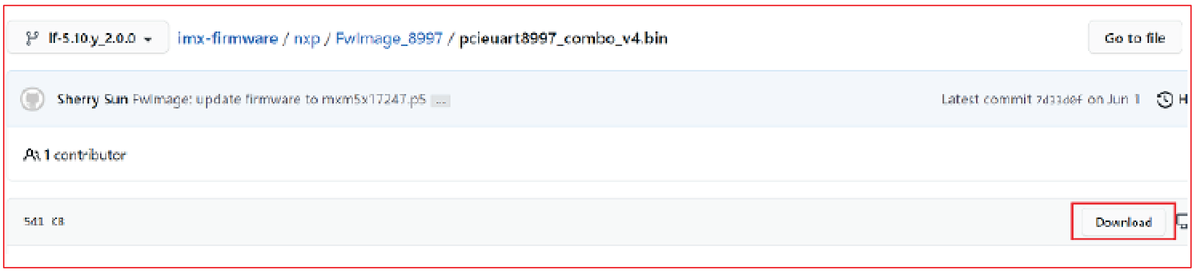
Sherry Sun FwImage: update firmware to mxm5x17247.p5 ...

..

ed_mac_ctrl_V3_8997.conf	FwImage: update firmware to mxm5x16203
pcieuart8997_combo_v4.bin	FwImage: update firmware to mxm5x17247.p5
txpwrlimit_cfg_8997.conf	FwImage: update firmware to mxm5x16203

4. Click the **Download** button to initiate the firmware download.

Figure 13: Initiating the Firmware Download



imx-firmware / nxp / FwImage_8997 / pcieuart8997_combo_v4.bin

Sherry Sun FwImage: update firmware to mxm5x17247.p5 ...

Latest commit 7d33def on Jun 1

1 contributor

541 KB

5 Build the Driver for ARM64 Platforms

This section describes the steps to build the driver package including the utility for ARM64 platforms.

The examples include building the driver for the following platforms:

- i.MX 8M evaluation platform (EVK) with Yocto build system.
- Generic ARM64 platform.

5.1 i.MX 8M Evaluation Platform with Yocto

This section shows how to generate a development SDK, build the toolchain, the driver, and utils binary.

Follow these steps:

1. Generate the development SDK.
   - Issue the command to generate an SDK from the Yocto Project build Yocto Project build environment.

```
$ DISTRO=<distro name> MACHINE=<machine name> source imx-setup-
release.sh -b <build dir>
```

- To set up the Yocto Project build environment, refer to the i.MX Yocto Project User's Guide (IMXLXYOCTOUG).
- For valid DISTRO options, see the section Build configurations in i.MX Yocto Project..; User's Guide (IMXLXYOCTOUG).

2. Build the toolchain.
   - Issue the command to build the toolchain from the build directory

```
$ bitbake meta-toolchain
```

3. Install the tool chain and tools.
   - Stay in the build directory.
   - Execute the script located in tmp/deploy/sdk/ directory.

```
$ sudo tmp/deploy/sdk/fsl-imx-wayland-glibc-x86_64-meta-
toolchaincortexa53- crypto-imx8mqevk-toolchain-5.10-hardknott.sh
```

> The name of the toolchain and/or script depends on your build machine and on the bitbake image.

   - When prompted about the directory to install the toolchain, accept the default options.
   - The following message confirms the installation completion.

```
SDK has been successfully set up and is ready to be used
```

4. Set up the toolchain environment on the host terminal window.
   - Issue the command to set up the toolchain environment on the host terminal window:

```
$ source <toolchain installed directory>/environment-setup-<toolchain
script>
```

Example for i.MX 8M EVK:

```
$ source /opt/fsl-imx-wayland/5-10-hardknott/environment-setupaarch64-
poky-linux
```

   - Check that the new environment variables are correctly set for the target i.MX 8 EVK.

```
$ echo $ARCH
arm64
$ echo $CROSS_COMPILE
aarch64-poky-linux
```

   - Export KERNELDIR directory to build the compatible driver modules.
   - Check that the Linux kernel version in KERNELDIR directory matches the Linux kernel version on the target i.MX 8M EVK.

```
$ export KERNELDIR=/<absolute directory path to yocto build>/tmp/work/
imx8mqevk-poky-linux/linux-imx/5.10.72+gitAUTOINC+a68e31b63f-r0/build/
```

5. Build the driver and utils.
   - Issue the cd command to go to the directory with the driver source file.

```
$ cd <driver source wlan_src directory>
```

- Issue the *make* command to build the driver.

```
$ make clean
$ make build
```

- The *bin_wlan* directory is created outside *wlan_src*. *bin_wlan* includes the driver and utils binary.

5.2 Build for ARM64 Generic Platforms

To build drivers for generic ARM64 platforms, first set up the toolchain for the generic ARM64 platform.

Refer to your platform BSP build environment setup guide for the steps and procedure.

Once the toolchain installation and setup are complete, follow the steps below to build the driver and utils.

1. Export KERNELDIR directory.
    - Export KERNELDIR directory to build the compatible driver modules.
    - Check that the Linux kernel version in KERNELDIR directory path matches the Linux kernel version on your ARM64 platform.

    ```
    $ export KERNELDIR=/absolute path to kernel build directory/
    ```

2. Build the driver and utils binary.
    - Issue the *cd* command to go to the directory with the driver source file.

    ```
    $ cd <driver source wlan_src directory>
    ```

    - Issue the *make* command to build the driver.

    ```
    $ make clean
    $ make build
    ```

    The *bin_wlan* directory is created outside *wlan_src* directory. *bin_wlan* directory includes the driver and utils binary.

Revision History

Revision	Date	Author	Change Description
1.0	Jul 30, 2024		Initial release

INNOVATOR IN ELECTRONICS

Specifications are subject to change without notice.