NuMicro® Family
Arm® -based Microprocessor

Nuvoton OpenWrt 22.03 Project
User Manual

For additional information or questions, please contact: Nuvoton Technology Corporation.

www.nuvoton.com

Table of Contents

1 OVERVIEW

The OpenWrt Project is a Linux operating system targeting embedded devices. Instead of trying to create a single, static firmware, OpenWrt provides a fully writable filesystem with package management. This Nuvoton OpenWrt is based on OpenWrt 22.03.0. The OpenWrt official website https://openwrt.org/ has some documents that OpenWrt base concepts.

1.1 Feature List

The released Nuvoton OpenWrt 22.03 has the following functions.

- Linux
- LuCI
- U-boot
- Arm-Trusted-Firmware (MA35D1/MA35D0/MA35H0 only)
- Optee-OS (MA35D1/MA35D0/MA35H0 only)
- Python3-Nuwriter (MA35D1/MA35D0/MA35H0 only)

2 DEVELOPMENT ENVIRONMENT SETUP

You need these things to develop projects in the OpenWrt Project environment. A host system with a minimum of 15 Gbytes of free disk space that is running a supported Linux distribution (i.e. recent releases of Fedora, CentOS, Debian, or Ubuntu), and appropriate packages installed on the system you are using for builds.

Nuvoton provides two environments of building image, one is Docker and the other is Linux. Docker is a virtual machine based on host Linux OS, so the setting in the Docker won't affect the host OS and the Docker can create an environment only for building image. Linux distribution will be updated and may result in building image error, so Docker provided by Nuvoton is a better way than Linux.

2.1 Docker

Docker is an open-source project based on Linux contains. They are similar to virtual machines, but containers are more portable, more resource-friendly, and more dependent on the host operating system. Docker provides a quick and easy way to get up and running with OpenWrt. Install docker, example for Ubuntu 20.04:

First, update your existing list of packages:

```
$ sudo apt-get update
```

Next, install a few prerequisite packages which let apt use packages over HTTPS:

```
$ sudo apt install apt-transport-https ca-certificates curl software-properties-
common
```

Then add Docker's official GPG key for the official Docker repository to your system:

```
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

Use the following command to set up the stable repository, add the Docker repository to APT sources:

```
$ sudo add-apt-repository "deb [arch=amd64]
https://download.docker.com/linux/ubuntu focal stable"
```

Next, update the package database with the Docker packages from the newly added repo:

```
$ sudo apt-get update
```

Finally, install Docker:

```
$ sudo apt-get install docker-ce docker-ce-cli containerd.io
```

Then use the Dockerfile to generate the docker image environment. after completion, use the repo utility to download the OpenWrt project after enter the docker image.

Dockefile source at (https://github.com/OpenNuvoton/MA35D1_Docker_Script.git)

You can use the docker script we provide.

```
build.sh  Dockerfile  join.sh  README.md
```

Setup docker image, and select folder to be share.

```
$ ./build.sh
Please enter absolute path for shared folders(eg:/home/<user name>) :
```

Enter docker image, you will see "[user name]&[container id]:$"

```
$ ./join.sh
ma35d1_test
test@575f27a6d251:~$
```

Create a shared/openwrt folder and enter

```
test@575f27a6d251:~$ mkdir shared/openwrt
test@575f27a6d251:~$ cd shared/openwrt
```

The first time you use repo, you need to set up the GIT environment.

```
test@575f27a6d251:~$ git config --global user.email "test@test.test.test"
test@575f27a6d251:~$ git config --global user.name "test"
```

Using git to download OpenWrt project

```
test@575f27a6d251:~$ git clone https://github.com/OpenNuvoton/Nuvoton-OpenWrt-22.03.git
```

You can check Docker documentation:

https://docker-curriculum.com/

https://docs.docker.com/get-started/

https://github.com/OpenNuvoton/docker

2.2 Linux

If you choose Linux environment for image building, there are some necessary packages must be installed before using OpenWrt project. For different Linux distributions, please refer below link to know the package list.

https://openwrt.org/docs/guide-developer/toolchain/install-buildsystem

For example, in Ubuntu 22.04, user can run below command to install necessary packages of OpenWrt project.

```
$ sudo apt install build-essential gawk gcc-multilib flex git gettext libncurses5-dev libssl-dev python3-distutils rsync unzip zlib1g-dev
```

In Ubuntu 20.04 and older, user can run below command to install necessary packages.

```
$ sudo apt install build-essential ccache ecj fastjar file g++ gawk \
gettext git java-propose-classpath libelf-dev libncurses5-dev \
libncursesw5-dev libssl-dev python python2.7-dev python3 unzip wget \
python-distutils-extra python3-setuptools python3-dev rsync subversion \
swig time xsltproc zlib1g-dev
```

Except of the OpenWrt project, there are also necessary packages must be installed for other projects. For building arm-trusted-firmware project, user can run below command to install necessary packages.

```
$ sudo apt install libssl-dev device-tree-compiler
```

For building op-tee project, user can run below command to install necessary packages.

```
$ pip3 install pycryptodomex pyelftools
```

For using python3-nuwriter command line tool, user can run below command to install necessary packages.

```
$ pip3 install pyusb usb crypto ecdsa crcmod tqdm pycryptodome
```

3 BUILD IMAGE

This section provides the detailed information along with the process for building an image.

3.1 Update Feeds Scripts

After download OpenWrt source is completed, execute following commands to update the OpenWrt feeds script.

```
$ ./scripts/feeds update -a
$ ./scripts/feeds install -a
```

If encounter following error, it means that your machine does not have the required certificate path to the CA of OpenWrt.

```
fatal: unable to access 'https://git.openwrt.org/feed/routing.git/':
gnutls_handshake() failed: The TLS connection was non-properly terminated.
```

To avoid this issue, you can run following command to tell git to ignore the cert procedure.

```
$ export GIT_SSL_NO_VERIFY=1
```

3.2 OpenWrt Configurations

3.2.1 NUC980 Platform

The NUC980 platform supports IoT board, IoT-G2 board, IoT-G2D board, Chili board, Server board and EVB board. Please use the correct setting for the target board. For example, to use the NUC980 IoT board, user needs to follow the steps below:

In folder Nuvoton_OpenWrt_22.03, use the file Nuvoton/config/config_nuc980_iot as the OpenWrt configuration file.

```
$ cp Nuvoton/config/config_nuc980_iot .config
```

Run “make menuconfig” to configure OpenWrt. Confirm the Target System is “Nuvoton NUC980”, and the Subtarget is the “NUC980 IoT”, as shown in Figure 3-1.

```
$ make menuconfig
```

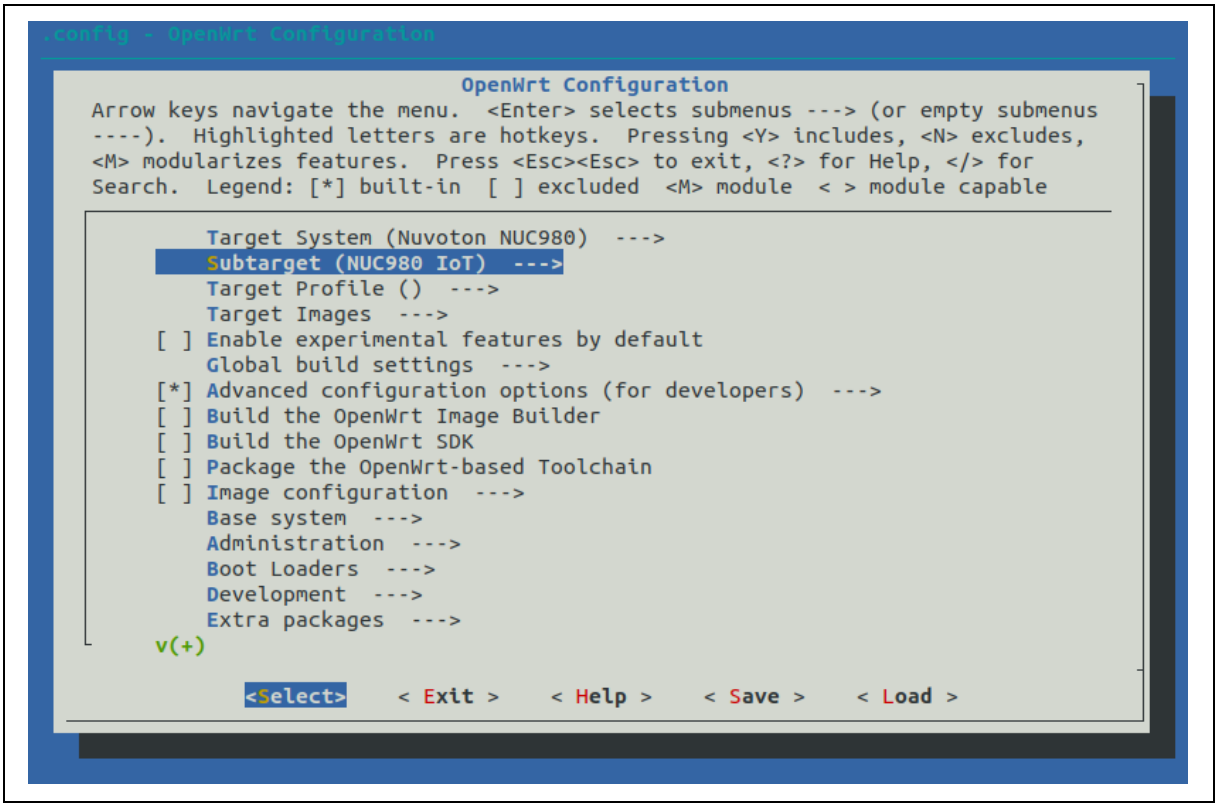

Figure 3-1 Select Target and Subtarget for NUC980

In Advanced configuration options page, specify the git repository and a branch/commit to clone Linux kernel source.

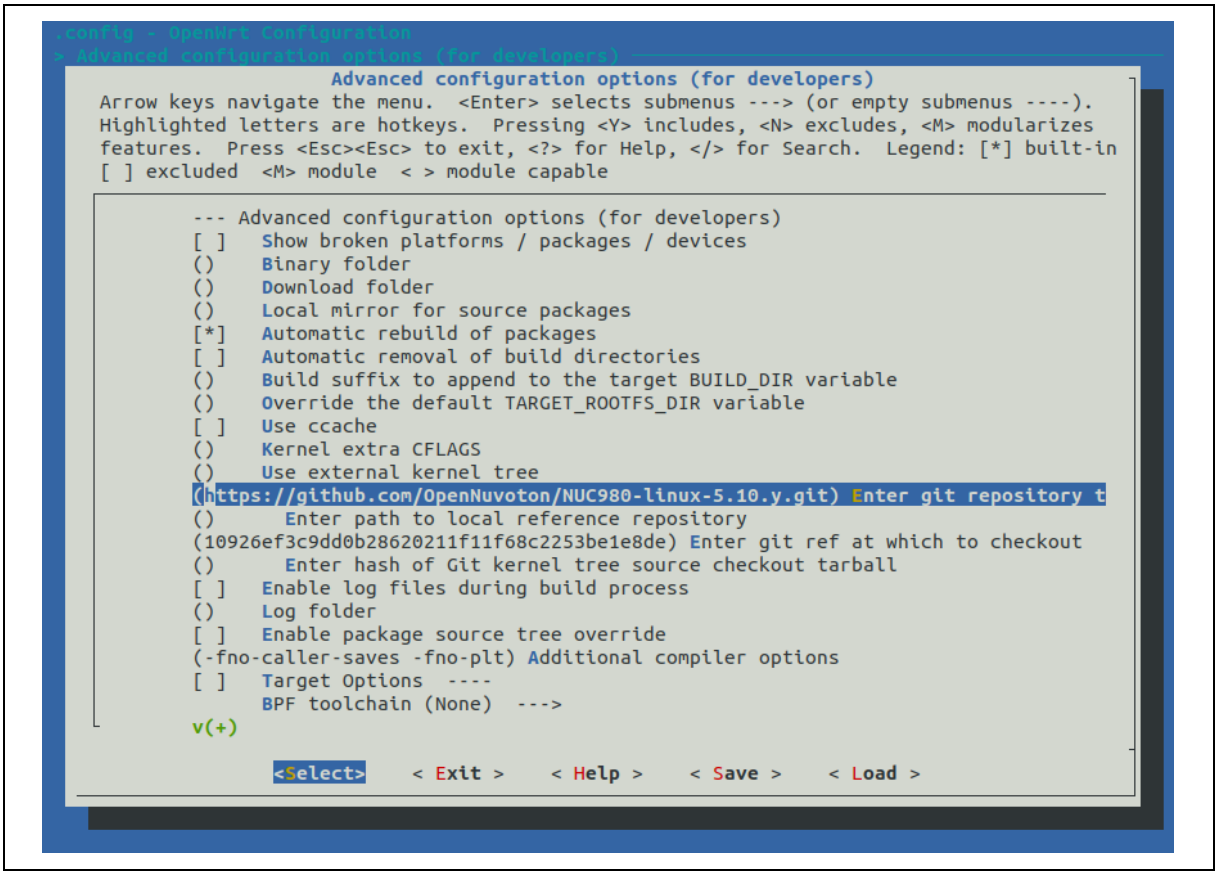

Figure 3-2 Configure Kernel Repository for NUC980

The default setting uses the HTTPS method to clone git repository from Github, user can change to use other repository managers. Table 3-1 lists available NUC980 Linux kernel repositories.

Repository Manger	URL
Github	https://github.com/OpenNuvoton/NUC980-linux-5.10.y.git
Gitlab	https://gitlab.com/OpenNuvoton/NuMicro-ARM7-ARM9-Family/NUC980-linux-5-10-y.git
Gitee	https://gitee.com/OpenNuvoton/NUC980-linux-5.10.y.git

Table 3-1 Linux Kernel Repositories for NUC980

In Boot Loaders page, specify the correct U-boot options. As shown in Figure 3-3.

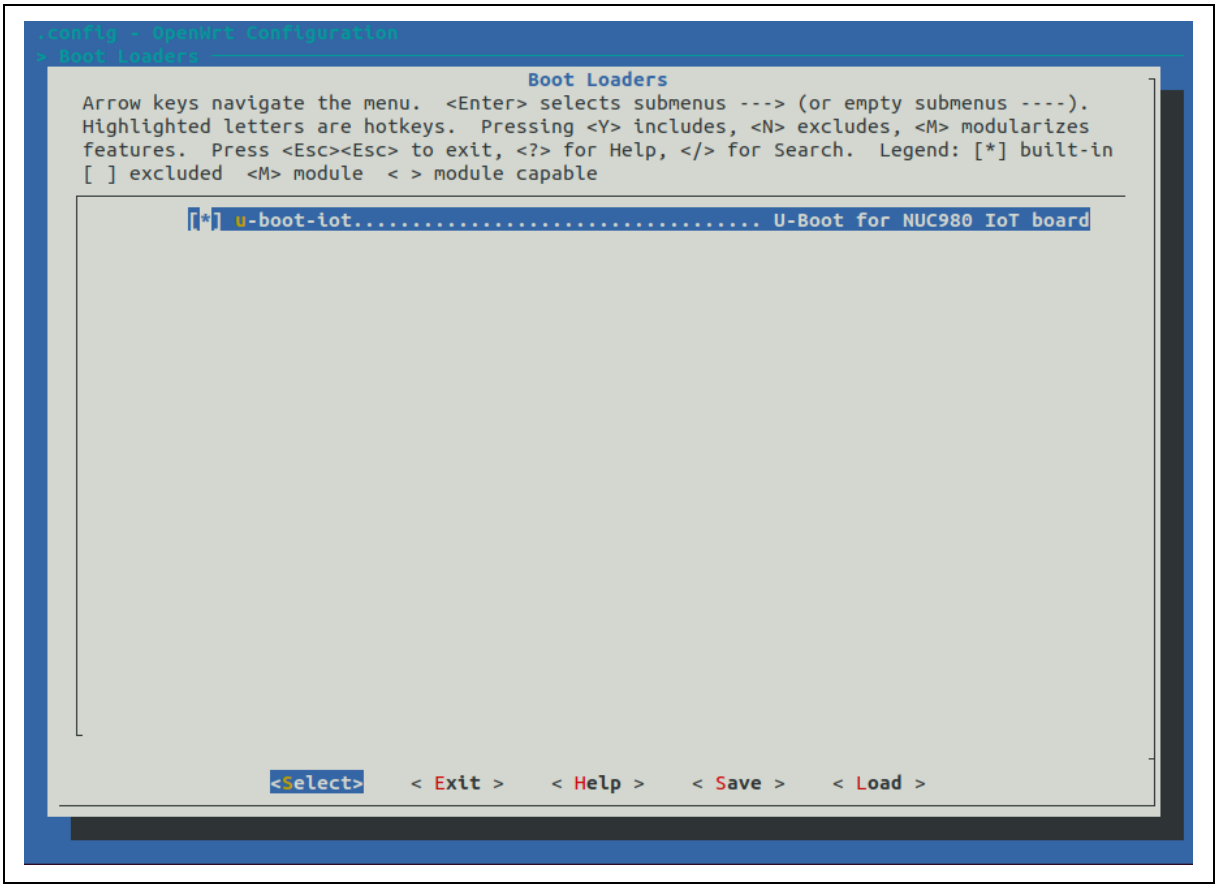

Figure 3-3 Boot Loaders Configurations for NUC980

3.2.2 MA35D1/MA35D0/MA35H0 Platform

The MA35D1 platform supports both of IoT board and SOM board, the MA35D0 platform supports IoT board, and the MA35H0 platform supports HMI board. Please use the correct setting for the target board. For example, to use the MA35D1 IoT board, user needs to follow the steps below:

In folder Nuvoton_OpenWrt_22.03, use the file Nuvoton/config/config_ma35d1_iot as the OpenWrt configuration file.

```
$ cp Nuvoton/config/config_ma35d1_iot .config
```

Run “make menuconfig” to configure OpenWrt. Confirm the Target System is “Nuvoton MA35D1”, the Subtarget is the “MA35D1 IoT”, and the Profile with the correct memory size and booting storage, as shown in Figure 3-4.

```
$ make menuconfig
```

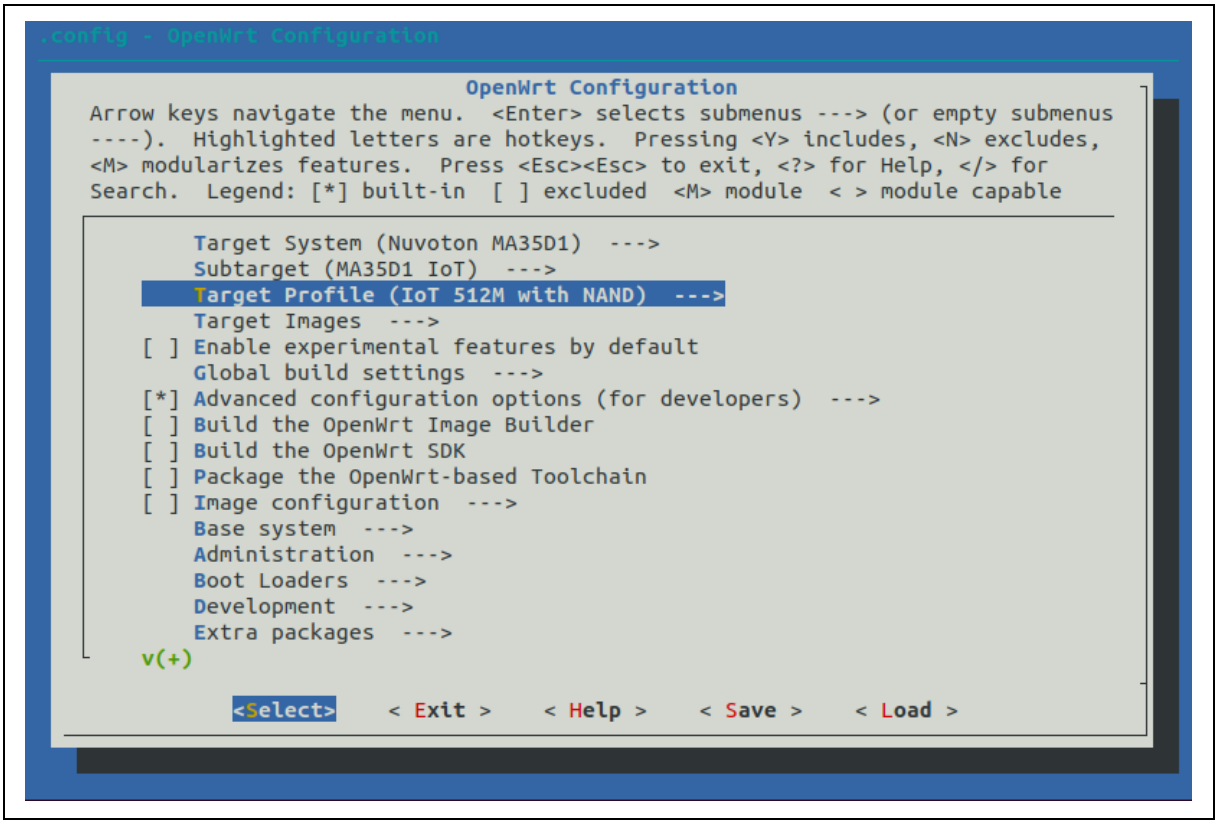

Figure 3-4 Select Target, Subtarget and Profile for MA35D1

In Advanced configuration options page, specify the git repository and a branch/commit to clone Linux kernel source.

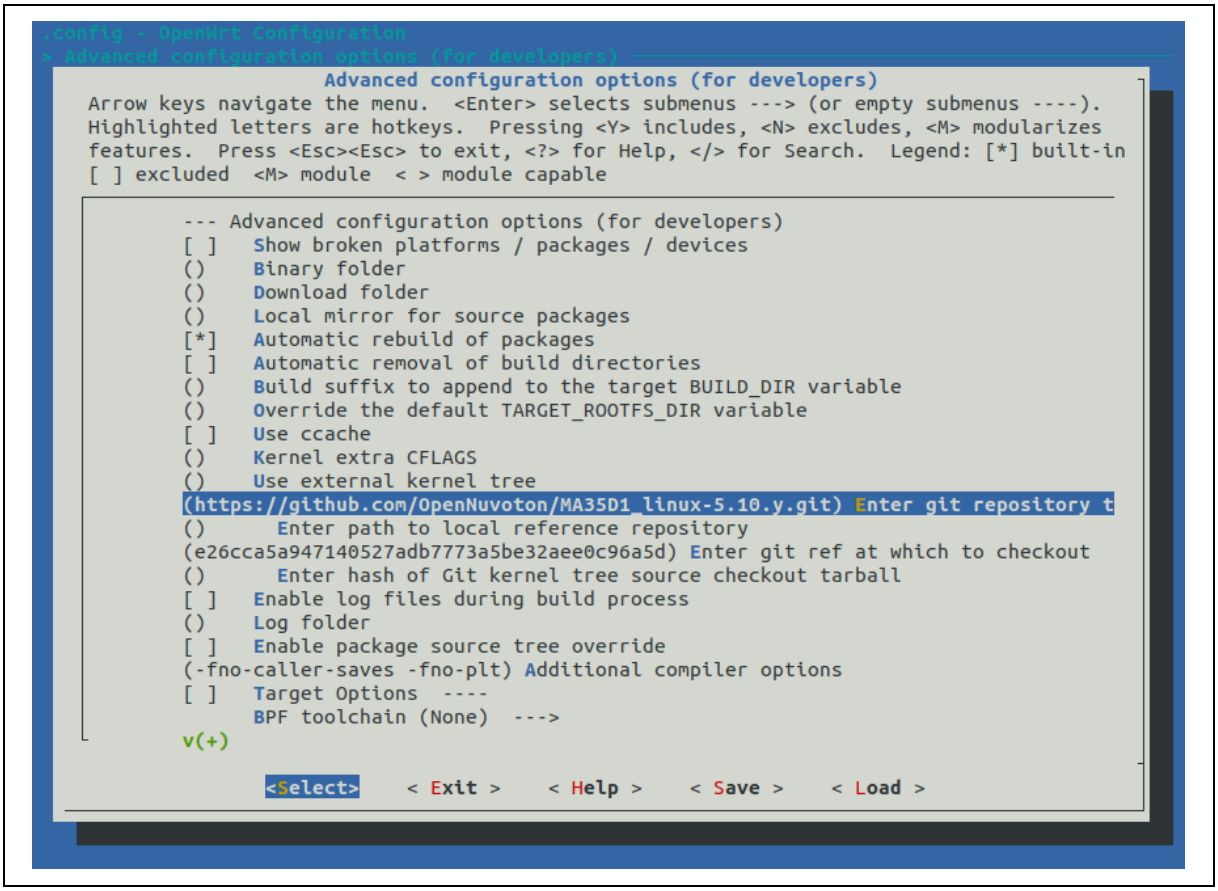

Figure 3-5 Configure Kernel Repository for MA35D1

The default setting uses the HTTPS method to clone git repository from Github, user can change to use other repository managers. Table 3-2 lists available MA35D1 Linux kernel repositories.

Repository Manger	URL
Github	https://github.com/OpenNuvoton/MA35D1_linux-5.10.y.git
Gitlab	https://gitlab.com/OpenNuvoton/Cortex-A-Family/MA35D1_linux-5-10-y.git
Gitee	https://gitee.com/OpenNuvoton/MA35D1_linux-5.10.y.git

Table 3-2 Linux Kernel Repositories for MA35D1

In Boot Loaders page, specify the correct Optee-OS, TF-A and U-boot options. As shown in Figure 3-6. Please notice you can select only one of Optee-OS, one of TF-A and one of U-boot.

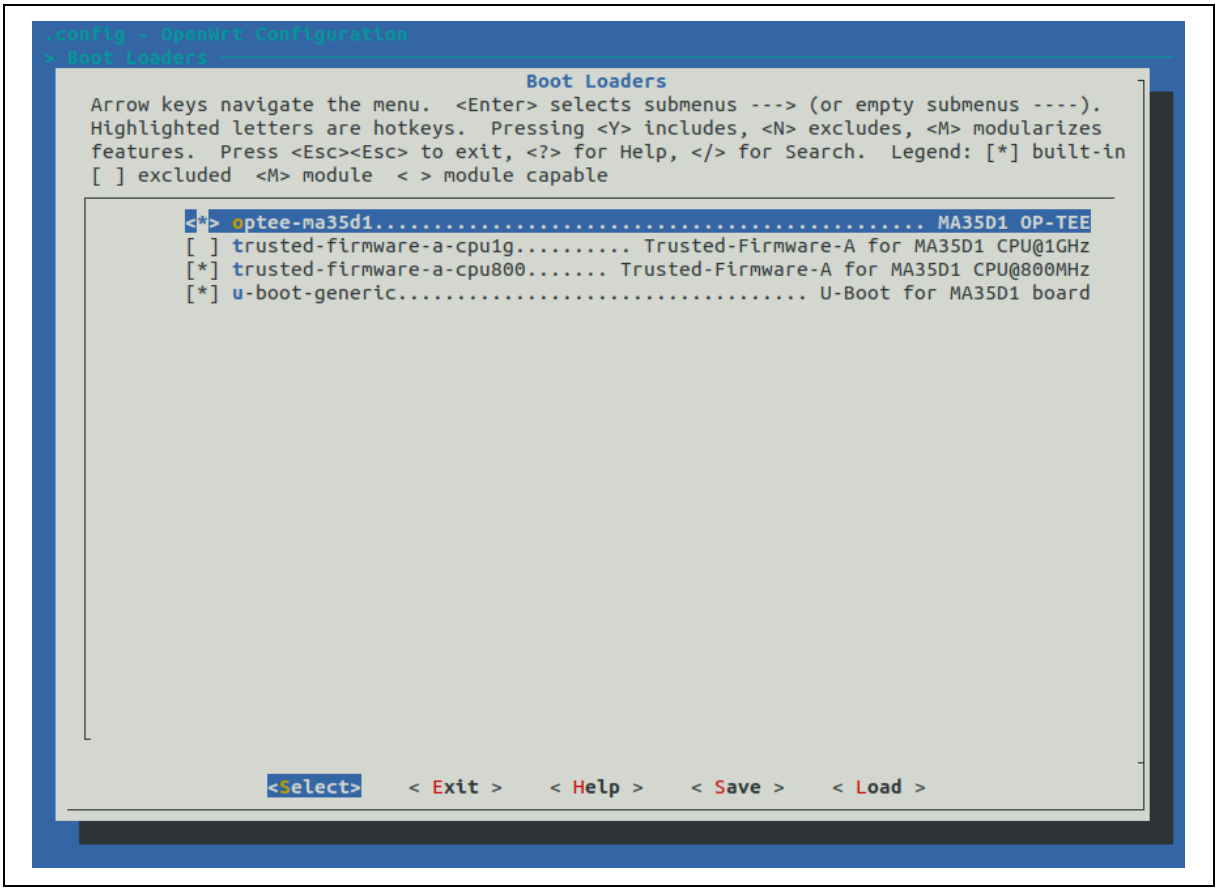

Figure 3-6 Boot Loaders Configurations for MA35D1

3.2.3 NUC970/N9H30 Platform

The NUC970 platform supports EVB board, and the N9H30 platform supports HMI board. Please use the correct setting for the target board. For example, to use the N9H30 HMI board, user needs to follow the steps below:

In folder Nuvoton_OpenWrt_22.03, use the file Nuvoton/config/config_n9h30_hmi as the OpenWrt configuration file.

```
$ cp Nuvoton/config/config_n9h30_hmi .config
```

Run “make menuconfig” to configure OpenWrt. Confirm the Target System is “Nuvoton NUC970/N9H30”, and the Subtarget is the “N9H30 HMI”, as shown in Figure 3-7.

```
$ make menuconfig
```

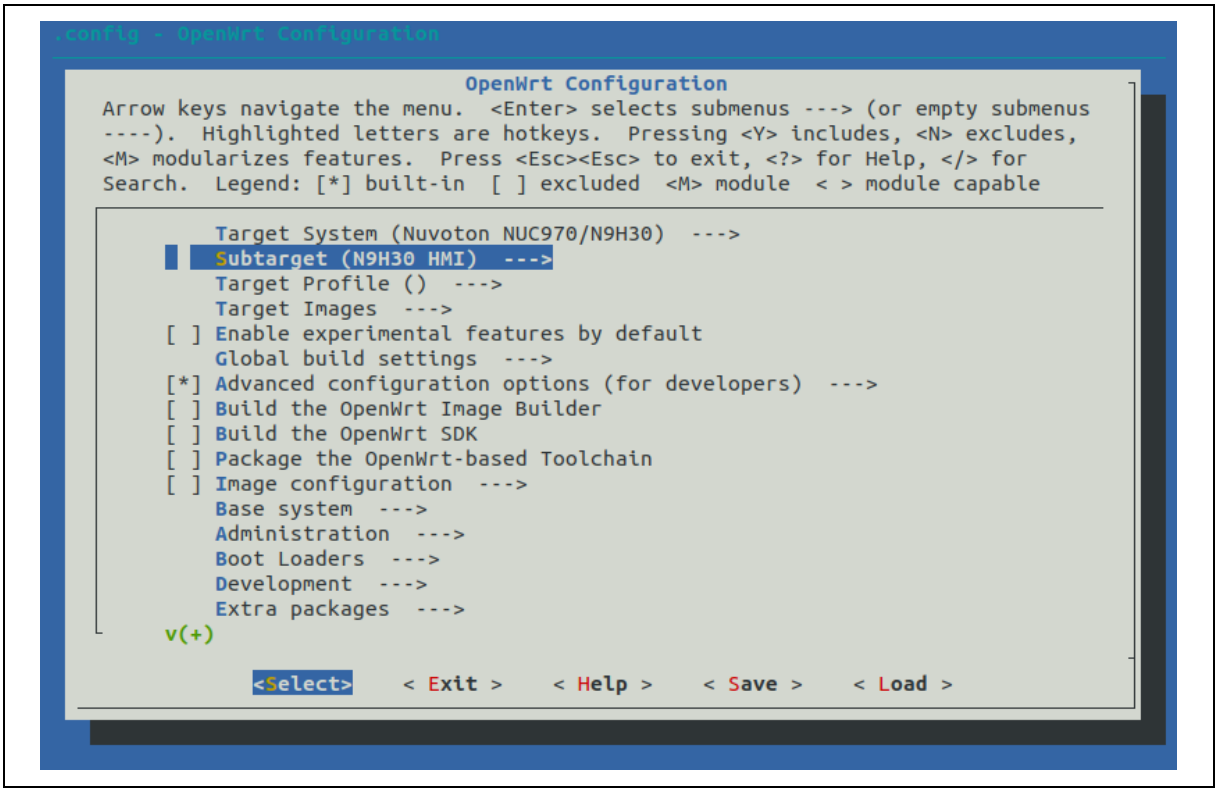

Figure 3-7 Select Target and Subtarget for NUC970/N9H30

In Advanced configuration options page, specify the git repository and a branch/commit to clone Linux kernel source.

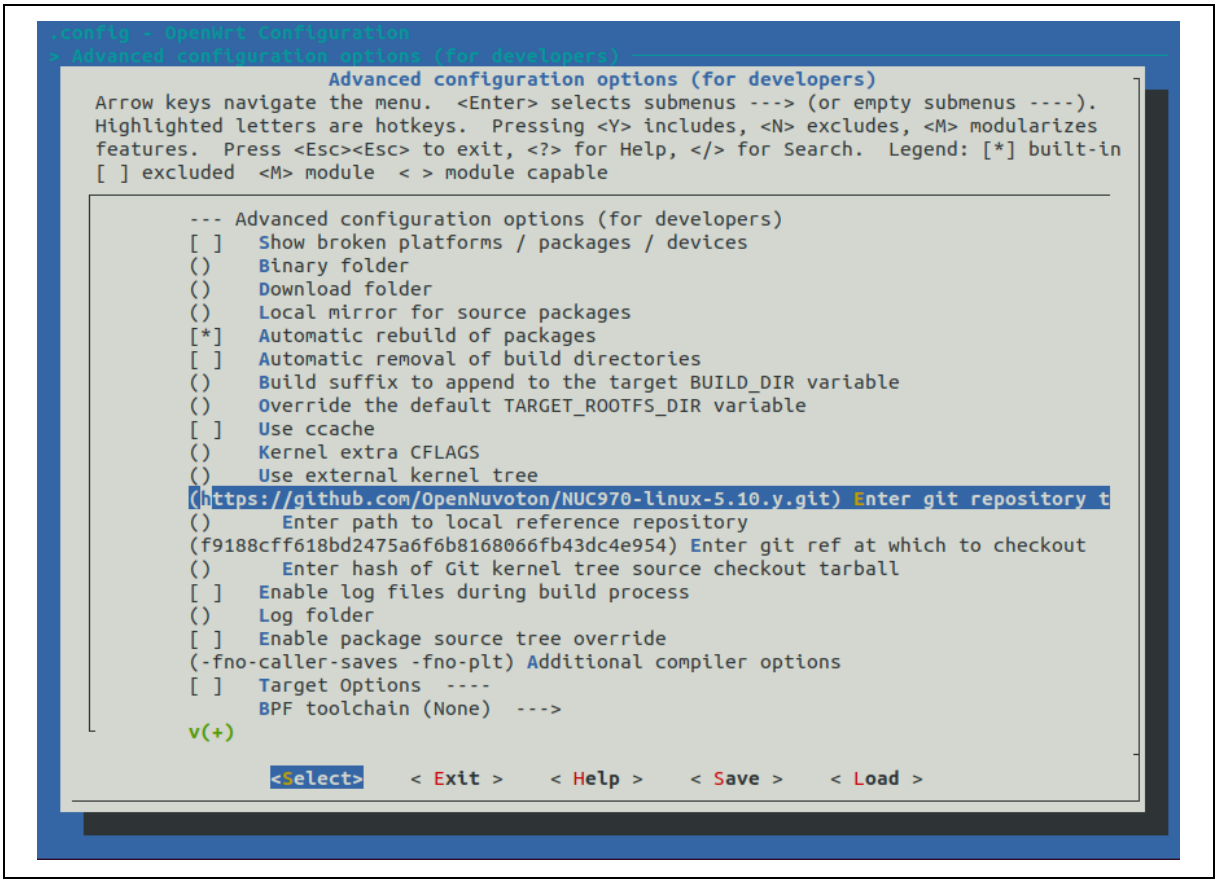

Figure 3-8 Configure Kernel Repository for NUC970/N9H30

The default setting uses the HTTPS method to clone git repository from Github, user can change to use other repository managers. Table 3-3 lists available NUC970/N9H30 Linux kernel repositories.

Repository Manger	URL
Github	https://github.com/OpenNuvoton/NUC970-linux-5.10.y.git
Gitlab	https://gitlab.com/OpenNuvoton/NuMicro-ARM7-ARM9-Family/NUC970-linux-5-10-y.git
Gitee	https://gitee.com/OpenNuvoton/NUC970-linux-5.10.y.git

Table 3-3 Linux Kernel Repositories for NUC970/N9H30

In Boot Loaders page, specify the correct U-boot options. As shown in Figure 3-9.

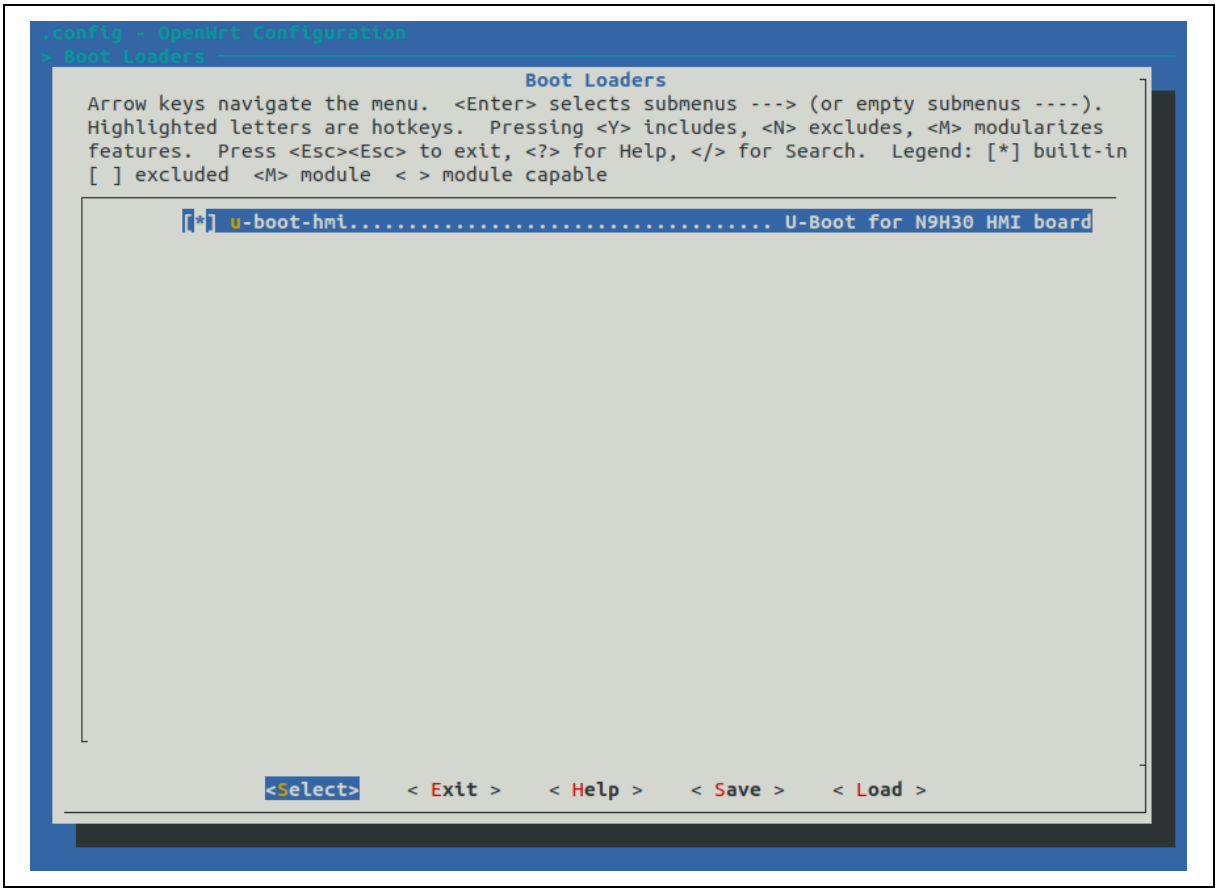

Figure 3-9 Boot Loaders Configurations for NUC970/N9H30

3.3 Linux Kernel Configurations

After setting OpenWrt configurations is completed, execute the command "make kernel_menuconfig" to configure Linux kernel.

```
$ make kernel_menuconfig
```

Before enter to Linux kernel configurations, it will build related toolchain packages and download NUC980 Linux BSP. This step may take 90 minutes, then user can exit and save the config directly.

If user modifies any Linux kernel configuration, for the NUC980 IoT board case, the config file will be stored as file Nuvoton_OpenWrt_v22.03/target/linux/nuc980/iot/config-5.10.

3.4 Build Image

3.4.1 NUC980 Platform

After configuration is completed, run “make” command to build the OpenWrt. The building may take around 30 minutes.

```
$ make
```

For the IoT board, the generated images are located in the bin/targets/nuc980/iot/ directory. The output files are shown as Table 3-5. Please note for the OverlayFS Root file system, the IoT / IoT-G2 / Server / EVB board (SPI-NAND) uses the Squashfs+UBIFS, but Chili board (SPI-NOR) uses the Squashfs+JFFS2 by default.

Image Name	Description
openwrt-nuc980-${Subtarget}-${Profile}-squashfs-firmware.bin	OpenWrt firmware image, includes Linux kernel Image, and OverlayFS root filesystem.
openwrt-nuc980-${Subtarget}-${Profile}-squashfs-sysupgrade.bin	OpenWrt system upgrade image.
openwrt-nuc980-${Subtarget}.dtb	Device tree image.
openwrt-nuc980-${Subtarget}-u-boot.bin	Main U-Boot loader.
openwrt-nuc980-${Subtarget}-u-boot-spl.bin	Load main U-Boot from NAND Flash to DDR (IoT / IoT-G2 / EVB board only).
openwrt-nuc980-${Subtarget}-u-boot-env.txt	U-Boot environment file.

Table 3-4 NUC980 OpenWrt Generated Images in Output Directory

3.4.2 MA35D1/MA35D0/MA35H0 Platform

After configuration is completed, run “make” command to build the OpenWrt. The building may take around 30 minutes.

```
$ make
```

For the MA35D1 IoT board, the generated images are located in the bin/targets/ma35d1/iot/ directory. The output files are shown as Table 3-5.

Image Name	Description
openwrt-ma35d1-${Subtarget}-${Profile}-pack.bin	OpenWrt pack image using NAND / SPINAND / SDCARD / eMMC booting, can be used by MA35D1 nuwriter tool.
openwrt-ma35d1-${Subtarget}-${Profile}-squashfs-firmware.bin	OpenWrt firmware image using NAND / SPINAND booting, includes Linux kernel uImage, and OverlayFS root filesystem.
openwrt-ma35d1-${Subtarget}-${Profile}-squashfs-sysupgrade.bin	OpenWrt system upgrade image using NAND / SPINAND booting.
openwrt-ma35d1-${Subtarget}-${Profile}-ext4-sysupgrade.gz	OpenWrt system upgrade image using SDCARD / eMMC booting.

Table 3-5 MA35D1 OpenWrt Generated Images in Output Directory

When user changes the Target System, Subtarget or Target Profile setting in configuration, it needs to run below commands to clean the TF-A, Optee-OS and U-boot” packages for re-building, or the loaders will still keep the original setting.

```
$ make package/boot/arm-trusted-firmware-ma35d1/clean
$ make package/boot/optee-ma35d1/clean
$ make package/boot/uboot-ma35d1/clean
```

3.4.3 NUC970/N9H30 Platform

After configuration is completed, run “make” command to build the OpenWrt. The building may take around 30 minutes.

```
$ make
```

For the N9H30 HMI board, the generated images are located in the bin/targets/nuc970/hmi/ directory. The output files are shown as Table 3-6.

Image Name	Description
openwrt-n9h30-${Subtarget}-${Profile}-squashfs-firmware.bin	OpenWrt firmware image, includes Linux kernel Image, and OverlayFS root filesystem.
openwrt-n9h30-${Subtarget}-${Profile}-squashfs-sysupgrade.bin	OpenWrt system upgrade image.
openwrt-n9h30-${Subtarget}.dtb	Device tree image.
openwrt-n9h30-${Subtarget}-u-boot.bin	Main U-Boot loader.
openwrt-n9h30-${Subtarget}-u-boot-spl.bin	Load main U-Boot from NAND Flash to DDR.
openwrt-n9h30-${Subtarget}-u-boot-env.txt	U-Boot environment file.

Table 3-6 NUC970/N9H30 OpenWrt Generated Images in Output Directory

3.4.4 Error Fix

If any error occurs during compilation, use the following command to collect error log for further check.

```
$ make -j1 V=sc
```

3.5 Deploy Image

3.5.1 NUC980 Platform

When the target board uses the NAND related device as the storage media, such as IoT / IoT-G2 / Server / EVB board, user should do the “erase all” action first before programming the firmware.

Refer to *NUC980 NuWriter User Manual* to program U-Boot, U-Boot environment, device tree and firmware images to the target storage media, as shown in Figure 3-10 (for IoT board) and Figure 3-11 (for Chili board).

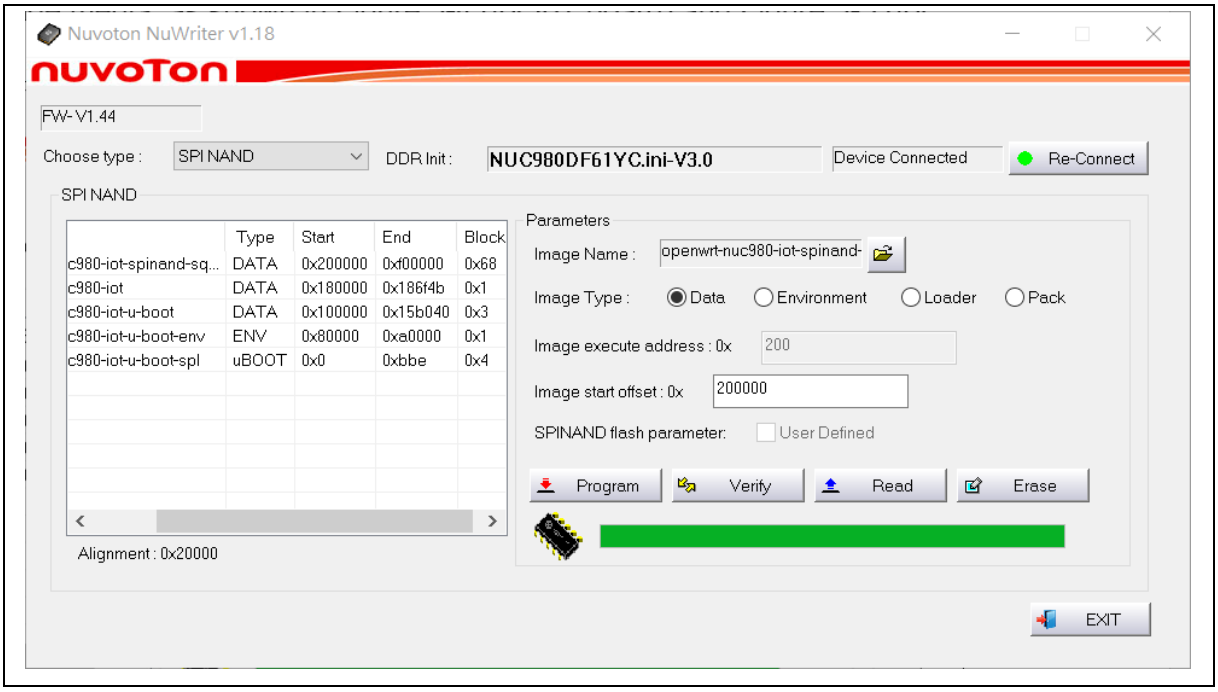

Figure 3-10 Use NuWriter to Program Images for NUC980 IoT Board

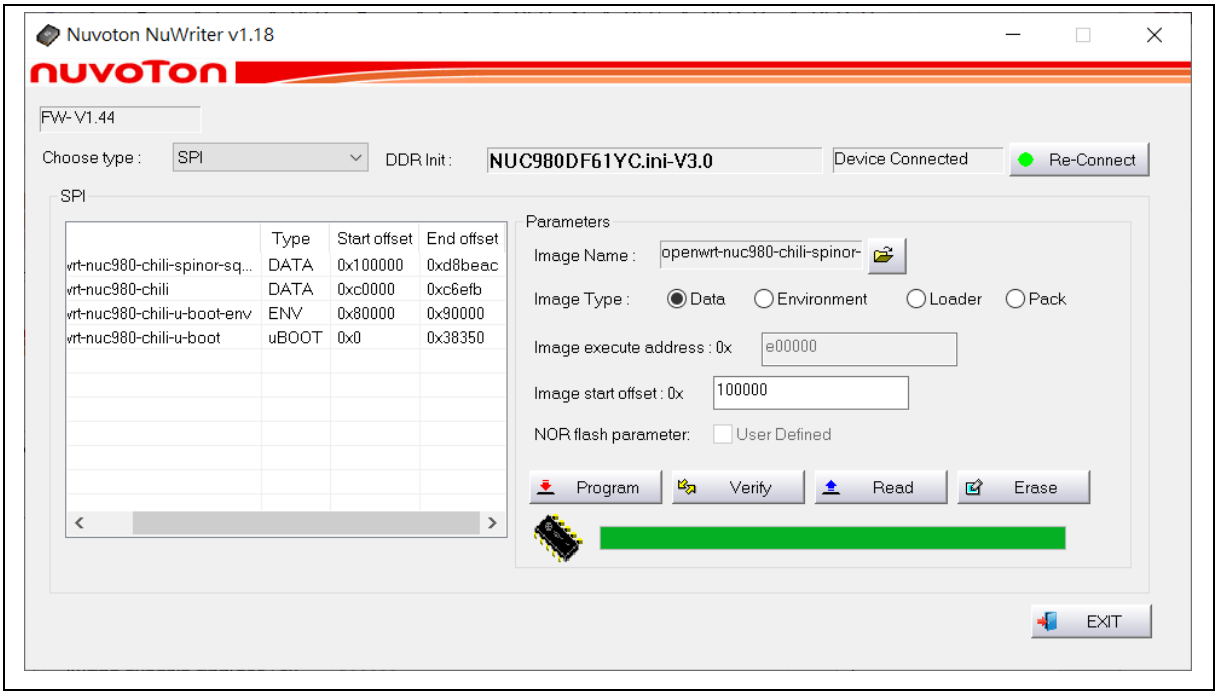

Figure 3-11 Use NuWriter to Program Images for NUC980 Chili Board

3.5.2 MA35D1/MA35D0/MA35H0 Platform

We provide pack image that includes TF-A, Optee-OS, U-boot, Linux kernel, and file system for NAND / SPINAND / SDCARD / eMMC booting storages. You can use the NuWriter GUI or command to write a pack image into MA35D1 demo board after computer connected that board.

3.5.2.1 NuWriter GUI Tool

For the NuWriter GUI tool, you need to choice the correct DDR image, and then press “Attach” button to connect the MA35D1 demo board.

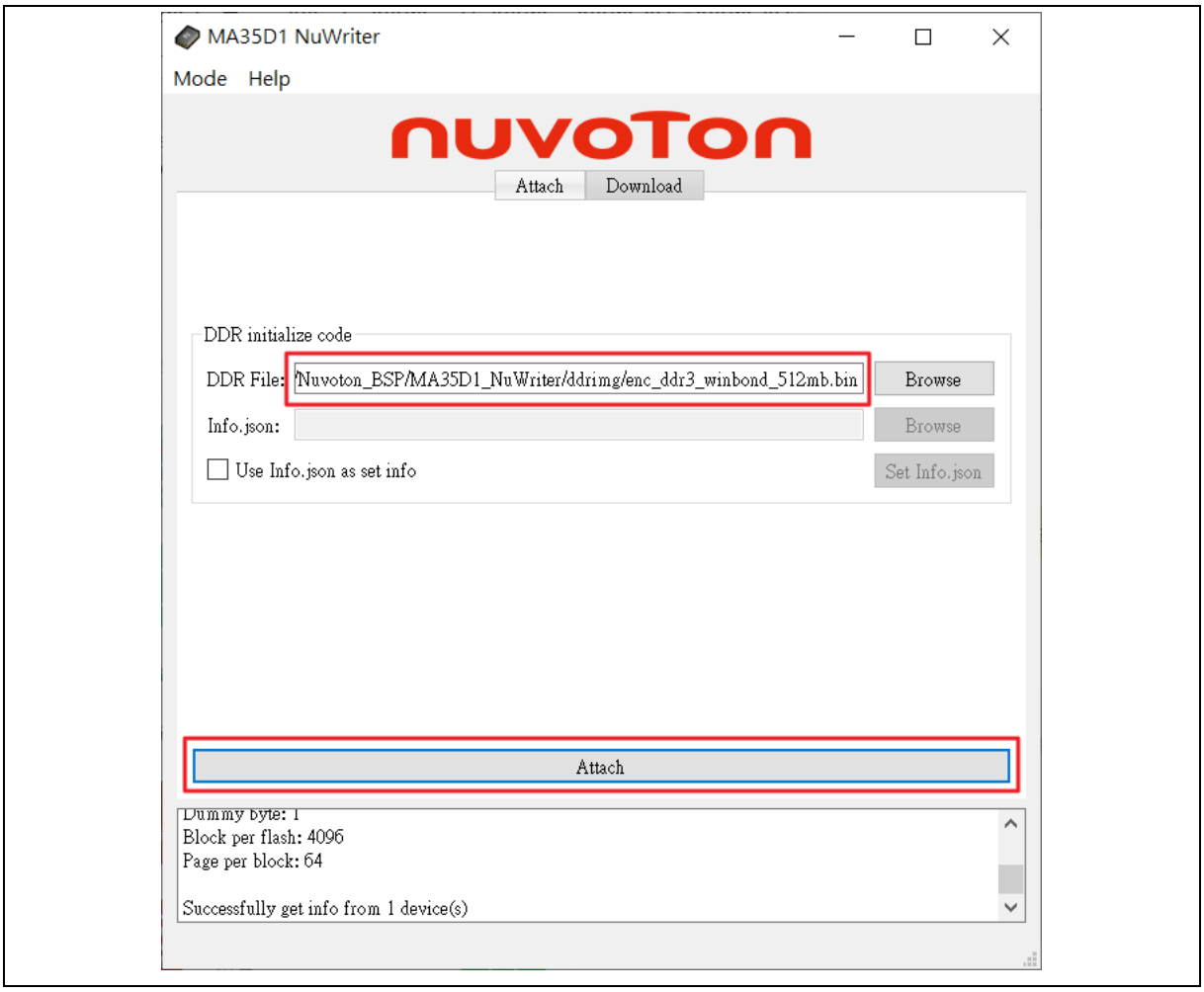

Figure 3-12 Use NuWriter to Attach MA35D1 IoT Board

Select the Download tab and the booting storage. Before to program NAND or SPINAND, you need to erase the storage at first.

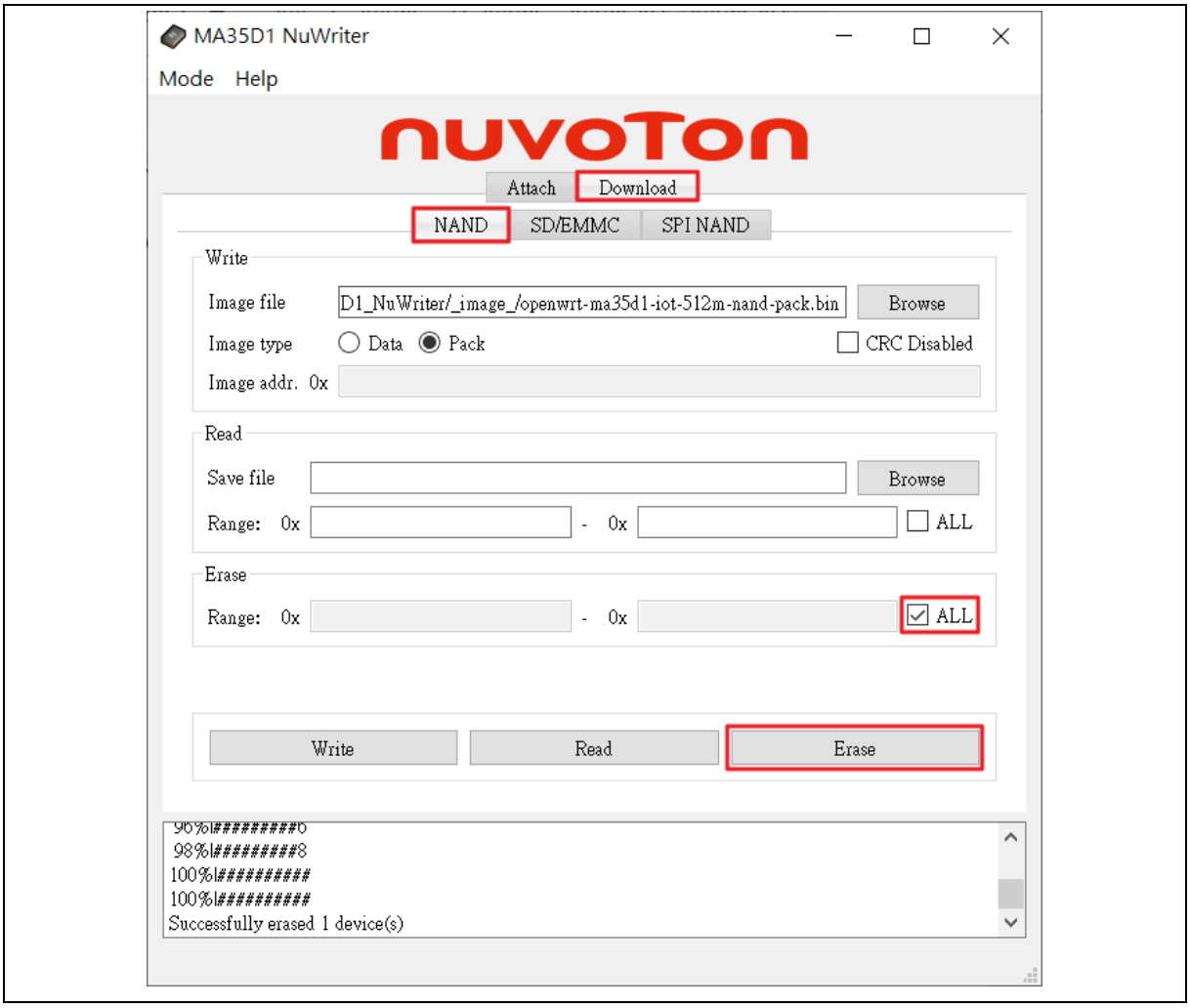

Figure 3-13 Use NuWriter to Erase MA35D1 IoT Board

Choice the correct pack image and select the image type as “Pack”, then click the “Write” button to program the pack image.

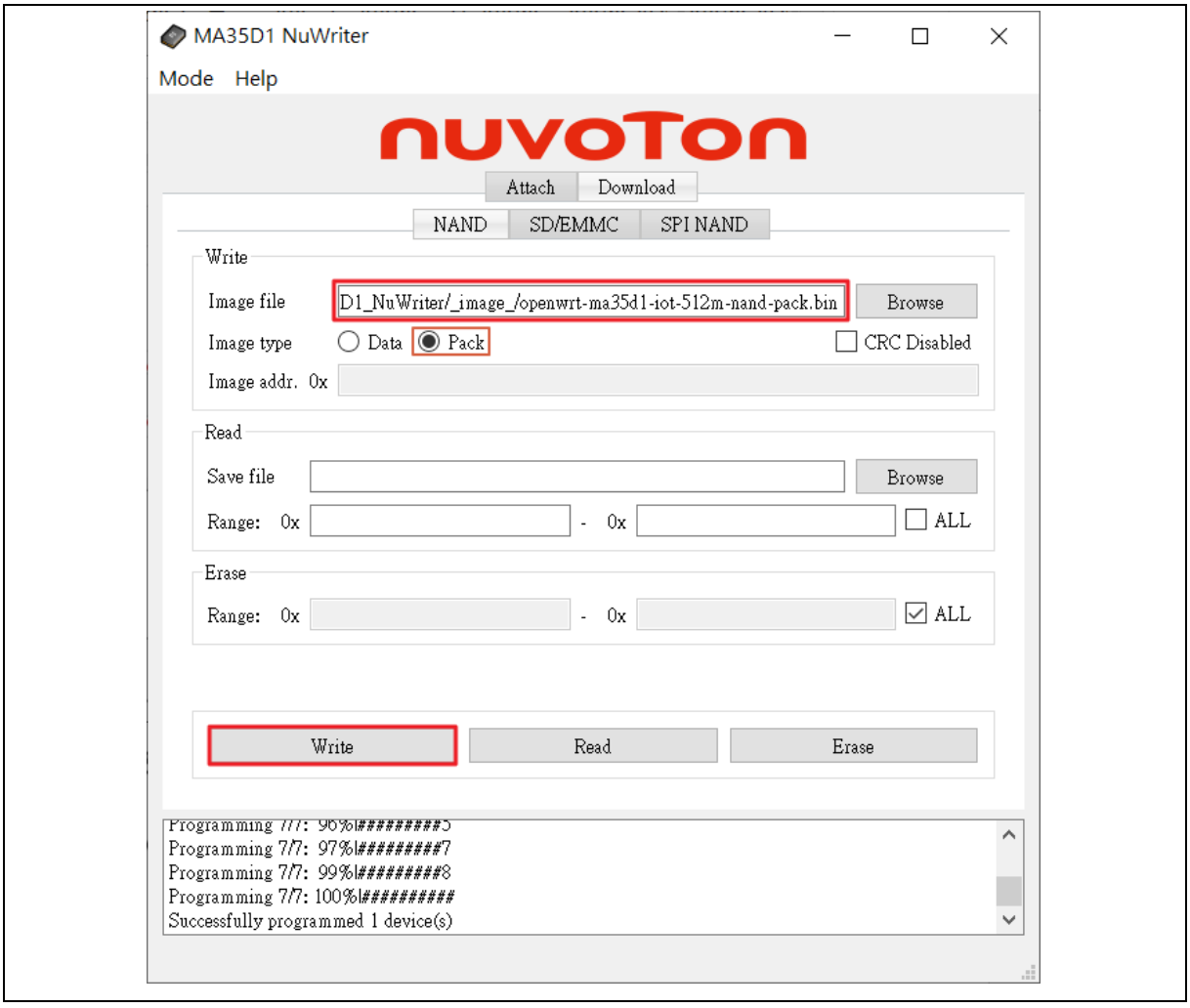

Figure 3-14 Use NuWriter to Program MA35D1 IoT Board

3.5.2.2 NuWriter Command

For the NuWriter command, you can use “lsusb” command to check whether the connection board is working. ***Using “nuwriter” command does not work properly in Docker environment.***

```
$ sudo lsusb
Bus 001 Device 002: ID 80ee:0021 VirtualBox USB Tablet
Bus 001 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
/OpenWrt/bin/targets/ma35d1/iot$
```

Burn openwrt-ma35d1-iot-512m-nand-pack.bin image to NAND.

```
/OpenWrt/bin/targets/ma35d1/iot$ nuwriter.py -a enc_ddr3_winbond_512mb.bin
/OpenWrt/bin/targets/ma35d1/iot$ nuwriter.py -e nand all
/OpenWrt/bin/targets/ma35d1/iot$ nuwriter.py -w nand openwrt-ma35d1-iot-512m-nand-pack.bin
```

Burn openwrt-ma35d1-iot-512m-spinand-pack.bin image to SPINAND.

```
/OpenWrt/bin/targets/ma35d1/iot$ nuwriter.py -a enc_ddr3_winbond_512mb.bin
/OpenWrt/bin/targets/ma35d1/iot$ nuwriter.py -e spinand all
```

```
/OpenWrt/bin/targets/ma35d1/iot$ nuwriter.py –w spinand openwrt-ma35d1-iot-512m-spinand-pack.bin
```

Burn openwrt-ma35d1-iot-512m-sdcard1-pack.bin image to SDCARD.

```
/OpenWrt/bin/targets/ma35d1/iot$ nuwriter.py –a enc_ddr3_winbond_512mb.bin
/OpenWrt/bin/targets/ma35d1/iot$ nuwriter.py –w sd openwrt-ma35d1-iot-512m-sdcard1-pack.bin
```

3.5.3 NUC970/N9H30 Platform

When the target board uses the NAND related device as the storage media, such as HMI / EVB board, user should do the “erase all” action first before programming the firmware.

Refer to *NUC970 NuWriter User Manual* to program U-Boot, U-Boot environment, device tree and firmware images to the target storage media, as shown in Figure 3-15 (for HMI board).

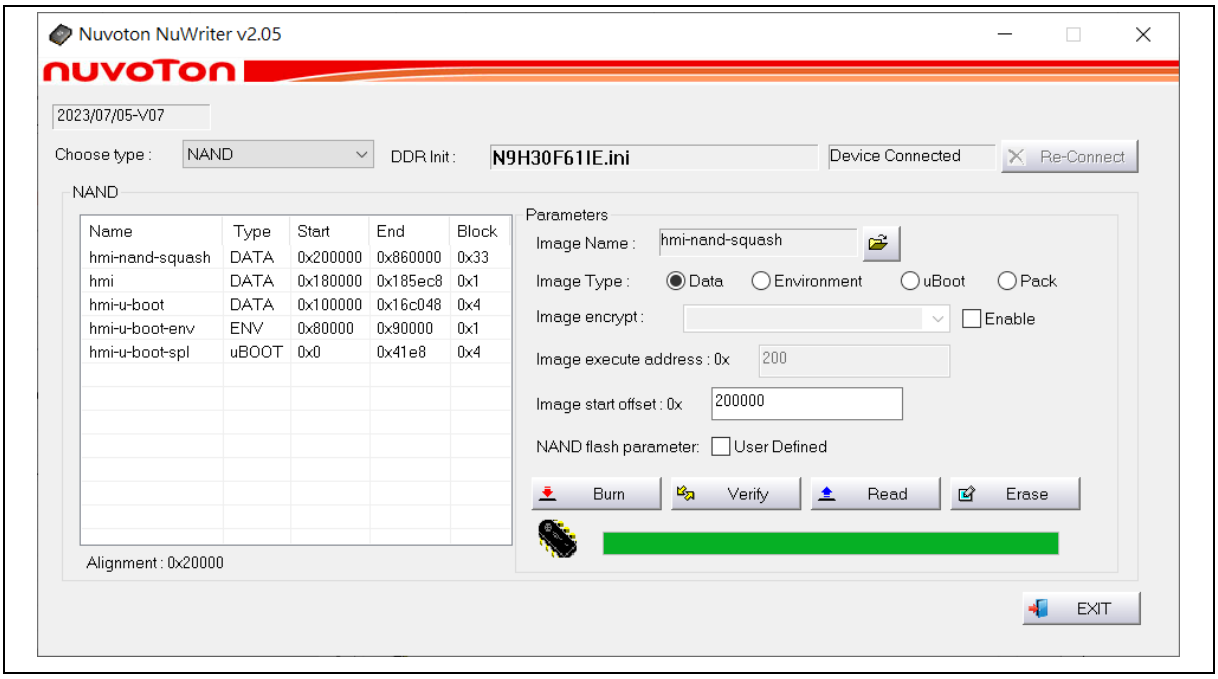

Figure 3-15 Use NuWriter to Program Images for N9H30 HMI Board

3.6 Secure Boot in MA35D1/MA35D0/MA35H0 Platforms

This section introduces how to enable secure boot feature in MA35D1/MA35D0/MA35H0 platform.

3.6.1 Configuration

In the “Target Images” menu, user can enable the Secure Boot feature, and change the AES key and ECDSA key.

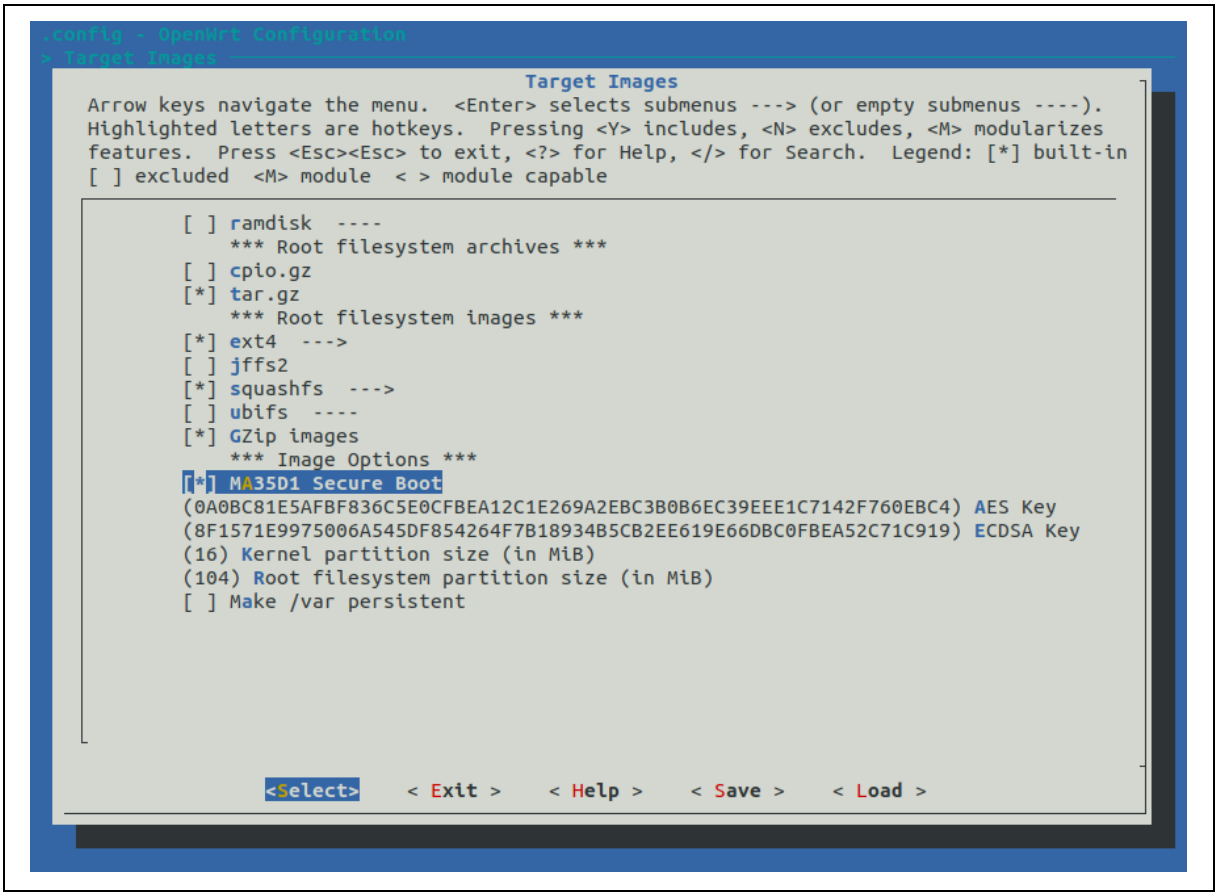

Figure 3-16 Enable the Secure Boot

3.6.2 Build Image

When user changes the Secure Boot setting in configuration, it needs to run below commands to clean the TF-A, Optee-OS and U-boot" packages for re-building, or the loaders will still keep the original setting.

```
$ make package/boot/arm-trusted-firmware-ma35d1/clean
$ make package/boot/optee-ma35d1/clean
$ make package/boot/uboot-ma35d1/clean
```

After the image building is done, user can find the output images with the "-enc" ending, which is different to the general output images. Except the original images, the OTP key file will also be generated. Following is the case of IoT board with NAND booting.

```
openwrt-ma35d1-iot-512m-nand-pack-enc.bin
openwrt-ma35d1-iot-512m-nand-squashfs-sysupgrade-enc.bin
openwrt-ma35d1-otp-key-enc.json
```

The content of the OTP key file is as following.

```
$ cat openwrt-ma35d1-otp-key-enc.json
{
  "publicx": "72F84F681092E3A05C1437E3E40534962A5C70556025D348FF9DB97D6AF83EB5",
  "publicy": "8D32DAC7AB6F90332E8E0060E159E0B31502BB4FB2D78369F02D1F5B0C335AD3",
```

```
  "aeskey": "0A0BC81E5AFBF836C5E0CFBEA12C1E269A2EBC3B0B6EC39EEE1C7142F760EBC4"
}
```

3.6.3 Deploy OTP Key File

For secure boot mode, except the pack image, the OTP key file also needs to be programmed to the key store of MA35D1 through NuWriter tools. For more details, please refer to *MA35D1 Secure Boot Application Note*.

3.6.3.1 NuWriter GUI Tool

For the NuWriter GUI tool, you need to change to OTP mode, and choice the OTP key file, then click the “Write” button to program the pack image.

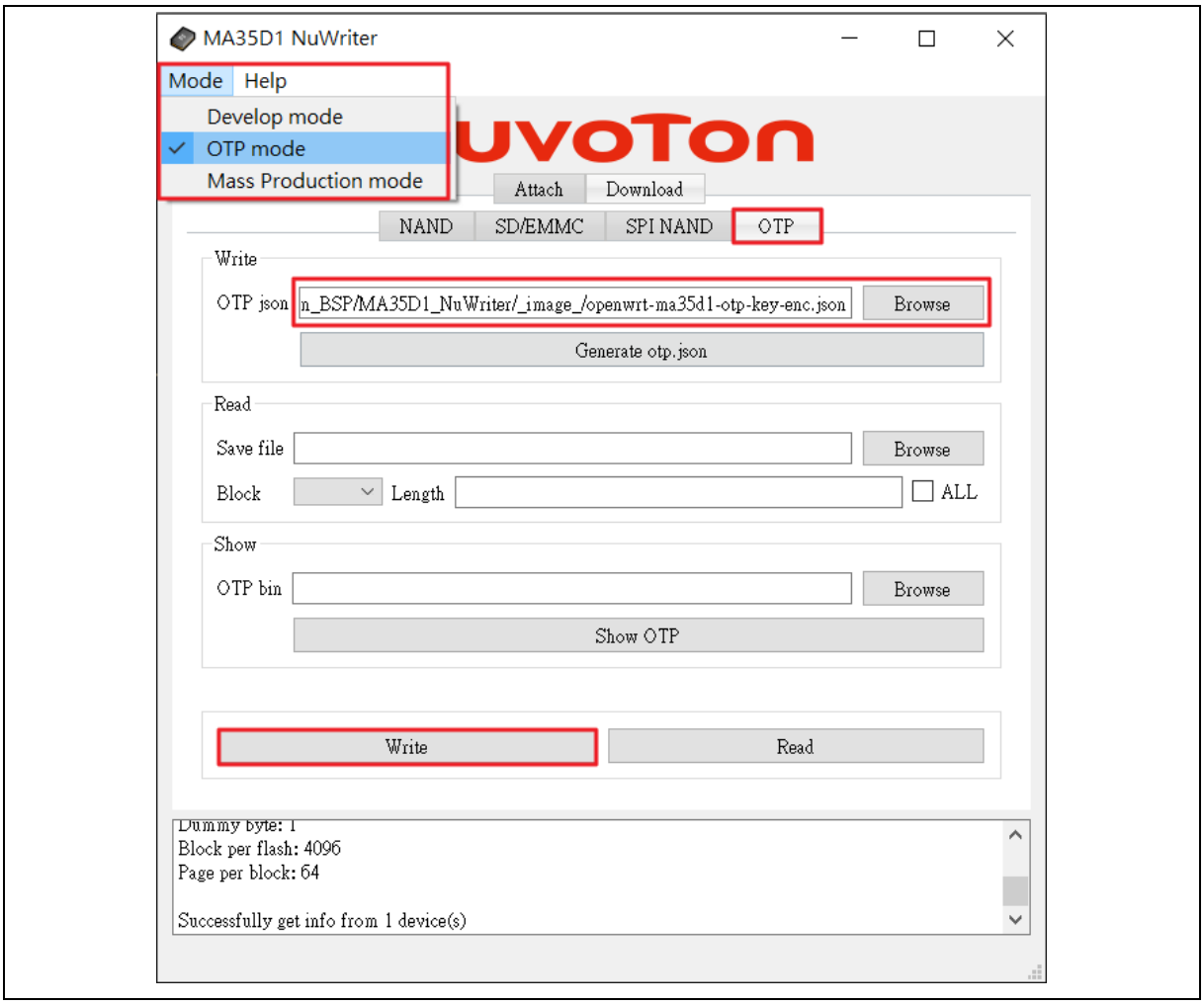

Figure 3-17 Use NuWriter to Program MA35D1 OTP key

3.6.3.2 NuWriter Command

For the NuWriter command, you can add follow command to burn OTP key file to Key Store.

```
/OpenWrt/bin/targets/ma35d1/iot$ nuwriter.py -w otp openwrt-ma35d1-otp-key-
enc.json
```

3.7 emWin

This section introduces how to enable the emWin feature on supported Nuvoton boards. Currently, the following development boards equipped with a display panel can support emWin.

NuMaker-HMI-N9H30

NuMaker-HMI-MA35D1

NuMaker-HMI-MA35H0

NuMaker-NUC980-IOTG2D

3.7.1 Copy the emWin Source to OpenWrt

After downloading the emWin package from the Nuvoton website, the user can extract the Linux emWin package tarball. For example, in the N9H30-HMI case, the user can execute the following commands.

```
$ cd package/nuvoton/emwin-nuc970/
$ tar zxvf N9H30_Linux_emWin_Package_20240110.tar.gz
```

The directory names after decompression contain a date suffix, and the user needs to remove it.

```
mv N9H30_Linux_emWin_Package_20240110 N9H30_Linux_emWin_Package
```

Please be aware that OpenWrt only supports the emWin package version released after the year 2024.

3.7.2 Configuration

In the “Nuvoton” menu, user can enable the emWin and tslib features.

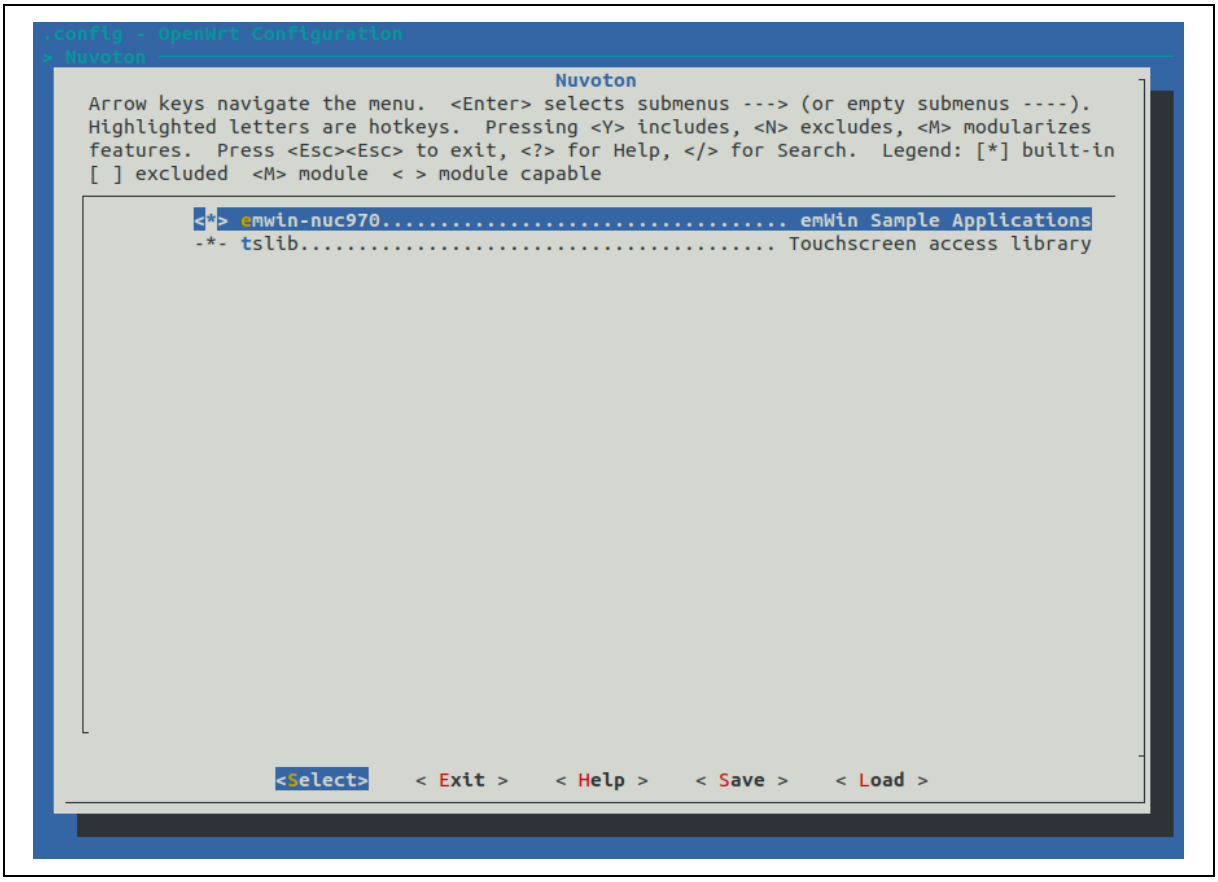

Figure 3-18 Enable the Secure Boot

3.7.3 Samples

The emWin source package contains several samples under “Sample” folder.

```
$ ls package/nuvoton/emwin-nuc970/N9H30_Linux_emWin_Package/Sample/
GUIDemo  SimpleDemo  SimpleDemoAppWizard
```

In the image-building procedure, these emWin samples will be compiled and copied to the rootfs. After OpenWrt boots up, users can run the emWin demos as follows.

```
root@OpenWrt:/# ls usr/bin/*Demo*
usr/bin/GUIDemo
usr/bin/SimpleDemoAppWizard
usr/bin/SimpleDemo
```

4 TEST OPENWRT

4.1 Booting Messages

After starting the device, user will see the OpenWrt booting messages, as shown in Figure 4-1.

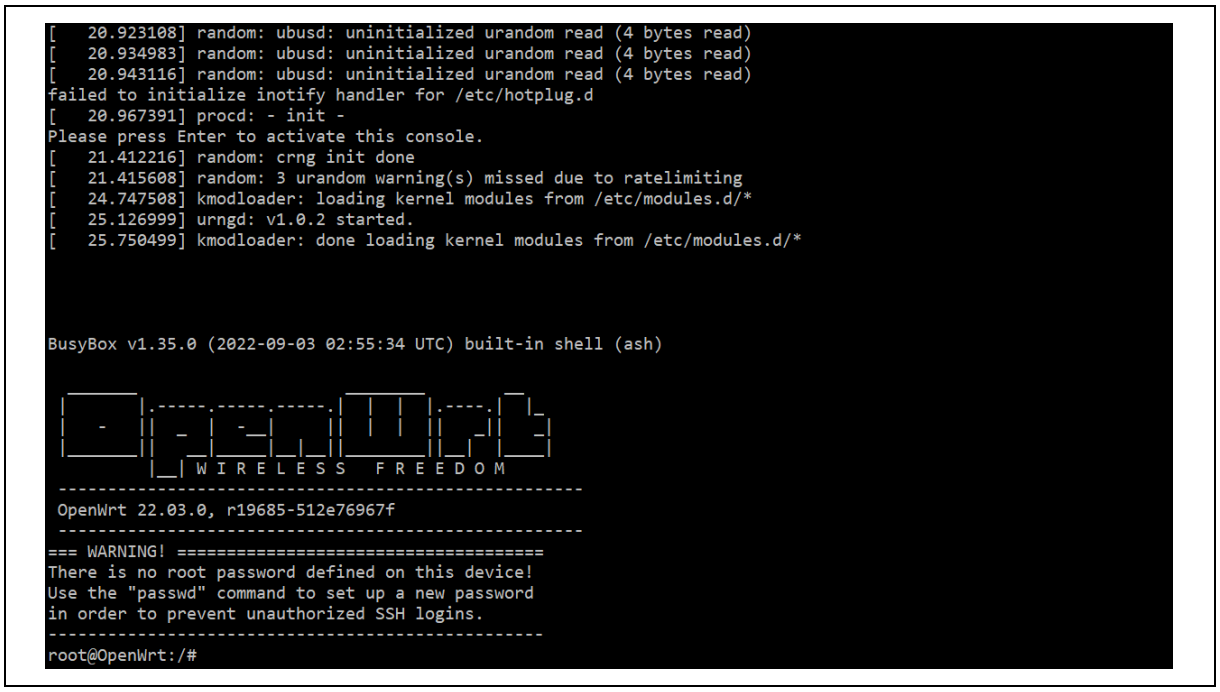

Figure 4-1 OpenWrt Booting Messages

4.2 Network Settings

To get the network setting information, user can run the following command.

```
uci show network
```

The default network setting is using the DHCP address, as shown in Figure 4-2 Default Network Settings

.

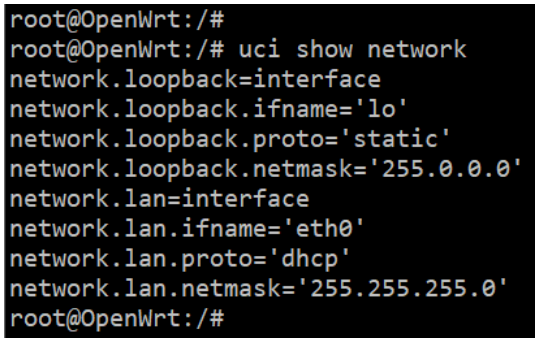

Figure 4-2 Default Network Settings

To change the network settings, user can modify the file path /etc/config/network directly, or run the “uci set” command. For example, to change to a static address, user can run following commands to modify and reset the network settings.

```
uci set network.lan.proto=static
uci set network.lan.ipaddr=192.168.1.100
uci set network.lan.netmask=255.255.255.0
/etc/init.d/network restart
```

4.3 LuCI Web Interface

To login the LuCI Web interface, user can connect to https://YOUR_IP_ADDRESS through a web browser such as Chrome. In the first time login, user may encounter a security warning message, as shown in Figure 4-3 First Time to Login the LuCI Web Interface

.

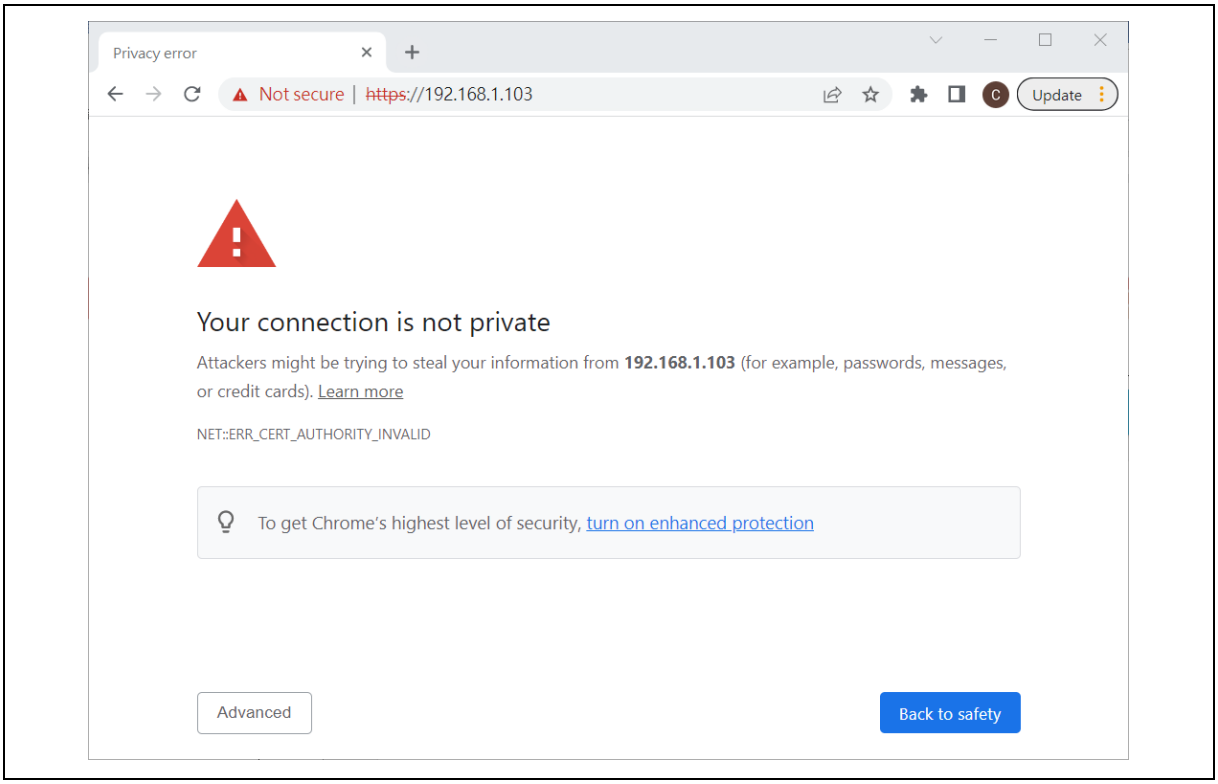

Figure 4-3 First Time to Login the LuCI Web Interface

After clicking the “Advanced” button, user will see a new screen, as shown in Figure 4-4 Proceed with Connection to LuCI

. Please proceed with the connection.

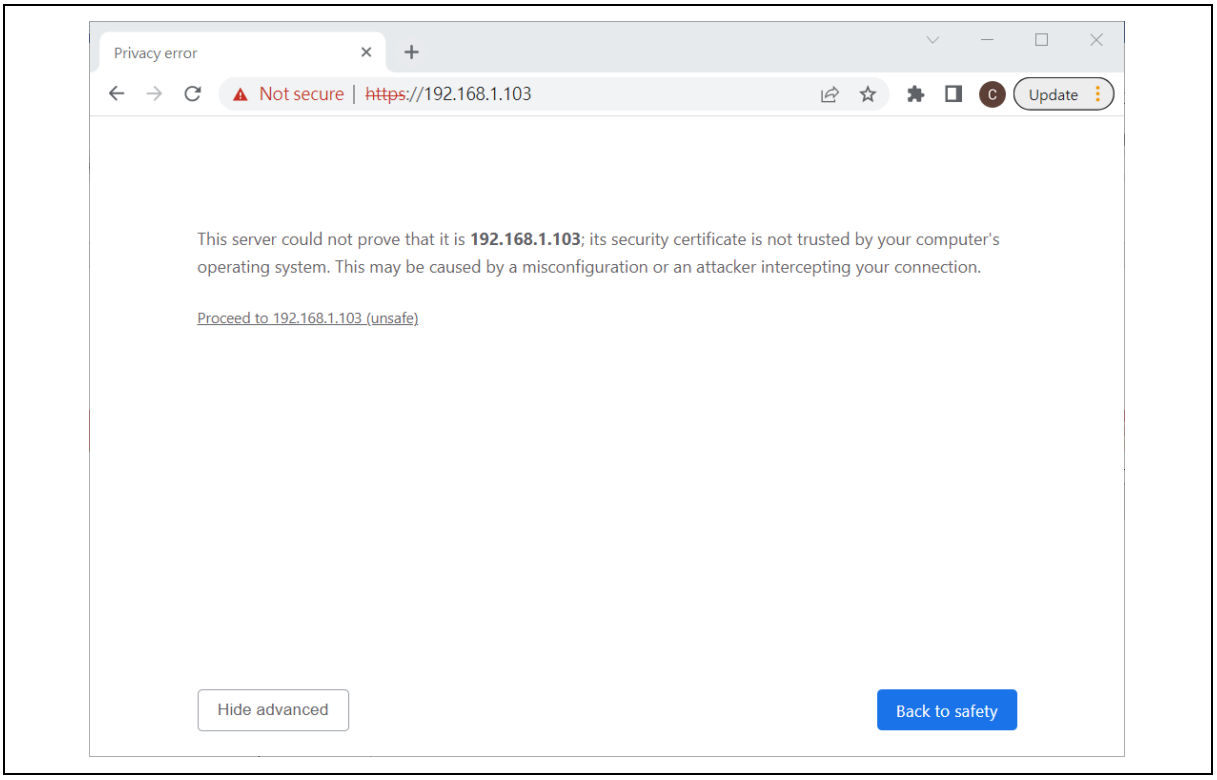

Figure 4-4 Proceed with Connection to LuCI

Then user can see the login screen. Since there is no password by default, user can login directly, or set a new password, as shown in Figure 4-5 Login Screen of LuCI

.

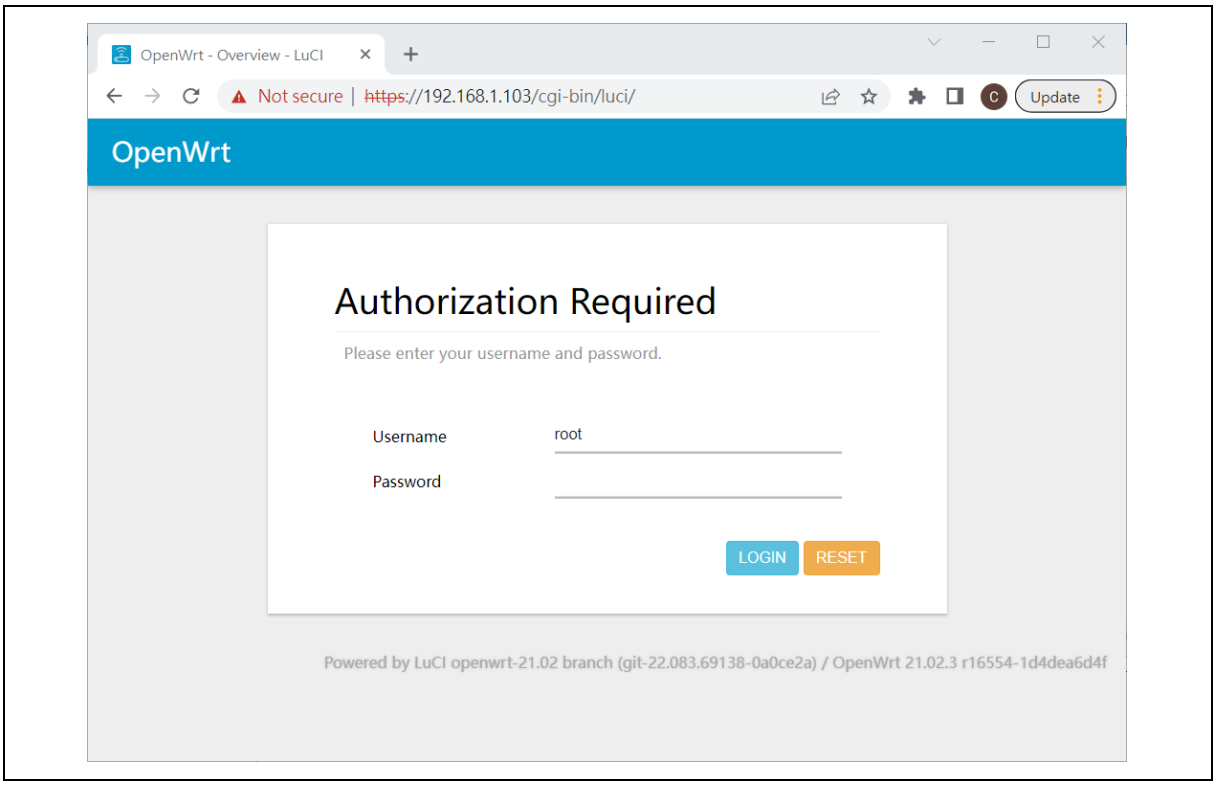

Figure 4-5 Login Screen of LuCI

4.4 Firmware Update

With the LuCI interface, user can upgrade the image which includes Linux kernel and OverlayFS root filesystem.

For the NUC980 IoT board, the image is as follows.

```
openwrt-nuc980-iot-spinand-squashfs-sysupgrade.bin
```

For the MA35D1 IoT board, the images are as follows.

```
openwrt-ma35d1-iot-512m-nand-squashfs-sysupgrade.bin
openwrt-ma35d1-iot-512m-spinand-squashfs-sysupgrade.bin
openwrt-ma35d1-iot-512m-sdcard1-ext4-sysupgrade.gz
```

To do the OpenWrt firmware upgrade, enter "System -> Backup/Flash Firmware". Choose the new sysupgrade image file, and then click "FLASH IMAGE" button, as shown in Figure 4-6 Firmware Upgrade in LuCI

.

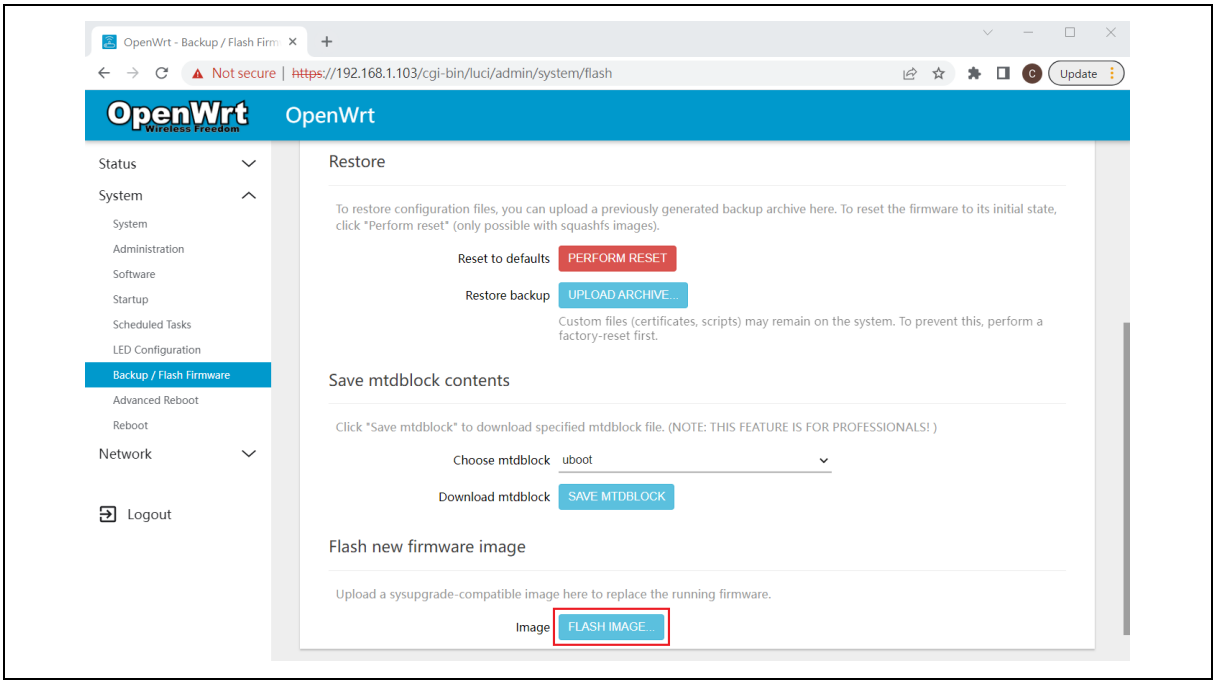

Figure 4-6 Firmware Upgrade in LuCI

5 REVISION HISTORY

Date	Revision	Description
2022.10.07	1.00	● Initial version. ● Support NUC980 platform.
2022.12.27	1.10	● Support MA35D1 platform.
2023.01.16	1.11	● MA35D1 platform supports eMMC booting.
2023.02.10	1.20	● MA35D1 platform supports secure boot.
2023.07.10	1.21	● NUC980 platform supports IoT-G2 board and EVB board.
2023.11.23	1.30	● Support NUC970/N9H30 platform.
2024.01.18	1.31	● Support emWin and samples.
2024.04.02	1.32	● Support MA35D0/MA35H0 platform.

Important Notice

Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, "Insecure Usage".

Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.

All Insecure Usage shall be made at customer's risk, and in the event that third parties lay claims to Nuvoton as a result of customer's Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.