

Arm[®] Cortex[®]-A35
64/32-bit Microcontroller

NuMicro[®] Family
MA35D1 OpenWrt Project
User Manual

The information described in this document is the exclusive intellectual property of Nuvoton Technology Corporation and shall not be reproduced without permission from Nuvoton.

Nuvoton is providing this document only for reference purposes of NuMicro microprocessor based system design. Nuvoton assumes no responsibility for errors or omissions.

All data and specifications are subject to change without notice.

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

www.nuvoton.com

Table of Contents

1 OVERVIEW 3

1.1 Feature List..... 3

2 DEVELOPMENT ENVIRONMENT SETUP 4

2.1 Docker 4

2.2 Linux 5

3 BUILD IMAGE 6

3.1 Update Feeds Scripts 6

3.2 OpenWrt Configurations..... 6

3.3 Linux Kernel Configurations..... 7

3.4 Build Image..... 8

3.5 Deploy Image 8

4 TEST OPENWRT 9

4.1 Booting Messages..... 9

4.2 Network Settings 9

4.3 LuCI Web Interface 10

4.4 Firmware Update 12

5 REVISION HISTORY 13

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. The MA35D1 OpenWrt is based on OpenWrt 21.02.3. The OpenWrt official website <https://openwrt.org/> has some documents that OpenWrt base concepts.

1.1 Feature List

The release MA35D1 of the OpenWrt has the following functions

- Linux
- LuCI
- Uboot
- Arm-Trusted-Firmware
- Optee-OS
- Python3-Nuwriter

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/MA35D1_OpenWrt.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
```

3.2 OpenWrt Configurations

The MA35D1 OpenWrt only supports SOM board with 256MB DDR and SPINAND booting now. Please use the correct setting for the target board. For example, to use the SOM board, user needs to follow the steps below:

In folder MA35D1_OpenWrt, use the file Nuvoton/config/config_ma35d1_som as the OpenWrt configuration file.

```
$ cp Nuvoton/config/config_ma35d1_som .config
```

Run “make menuconfig” to configure OpenWrt. Confirm if the Target System is “Nuvoton MA35D1”, and the Subtarget is the “MA35D1 SOM”, as shown in Figure 3-1.

```
$ make menuconfig
```

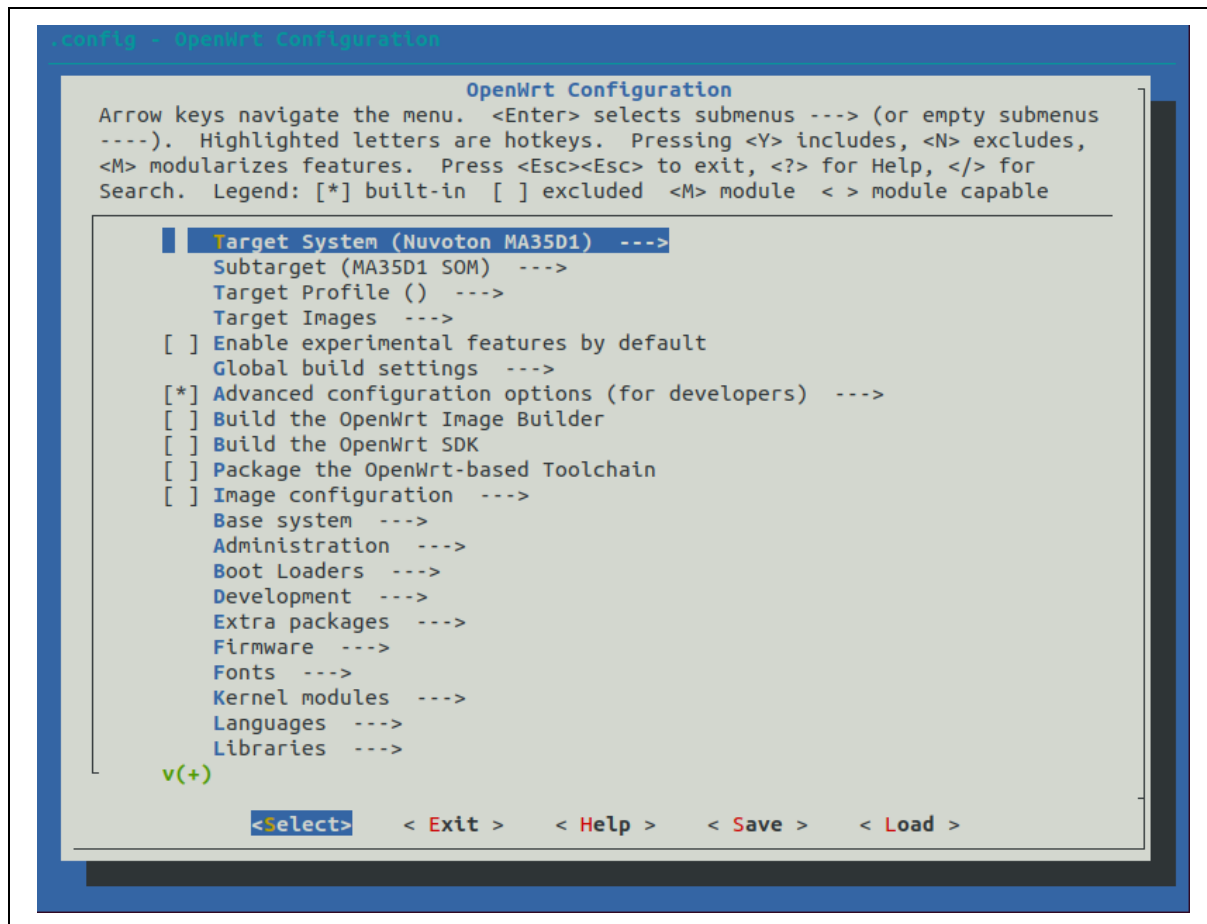


Figure 3-1 Select Target and Subtarget

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

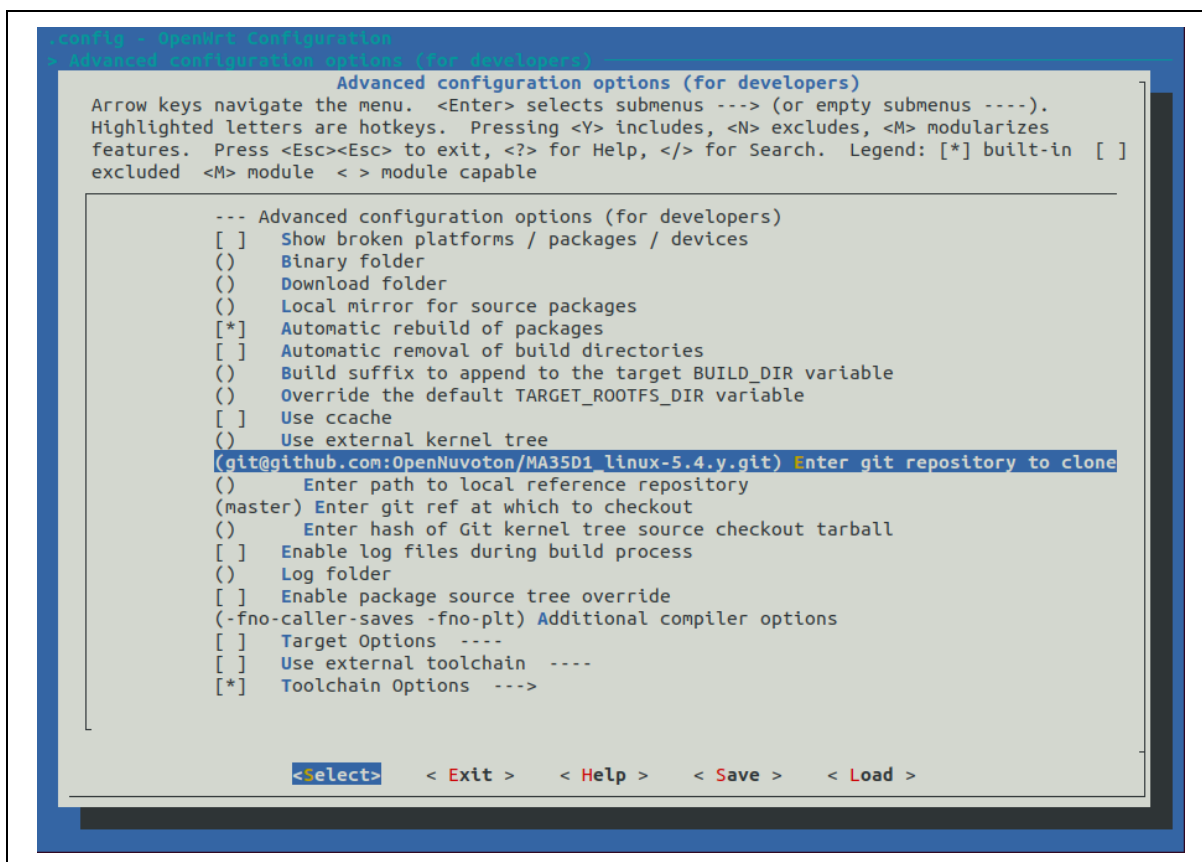


Figure 3-2 Configure Kernel Repository

The default setting uses the SSH method to clone git repository. Please make sure the public key of computer is attached to user's git account. If not, user can change to use the HTTPS method to clone. Table 3-1 lists available MA35D1 Linux kernel repositories of using the HTTPS method.

Repository Manger	URL
Github	https://github.com/OpenNuvoton/MA35D1_linux-5.4.y.git
Gitlab	https://gitlab.com/OpenNuvoton/Cortex-A-Family/ma35d1_linux-5.4.y.git
Gitee	https://gitee.com/OpenNuvoton/MA35D1_linux-5.4.y.git

Table 3-1 MA35D1 Linux Kernel Repositories

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 MA35D1 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 SOM board case, the config file will be stored as

file MA35D1_OpenWrt/target/linux/ma35d1/som/config-5.4.

3.4 Build Image

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

```
$ make
```

The generated images are located in the bin/targets/ma35d1/som/ directory. The output files are shown as Table 3-2.

Image Name	Description
Openwrt-ma35d1- $\{\text{Subtarget}\}$ - $\{\text{Subtarget}\}$ -pack-spinand.bin	OpenWrt pack image, can be used by MA35D1 nuwriter tool.
Openwrt-ma35d1- $\{\text{Subtarget}\}$ - $\{\text{Subtarget}\}$ -squashfs-firmware.bin	OpenWrt firmware image, includes Linux kernel ulmage, and OverlayFS root filesystem.
Openwrt-ma35d1- $\{\text{Subtarget}\}$ - $\{\text{Subtarget}\}$ -squashfs-sysupgrade.bin	OpenWrt system upgrade image.

Table 3-2 OpenWrt Generated Images in Output Directory

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

We provide pack image that includes TF-A, OPTEE-OS, u-boot, Linux kernel, and file system for SPINAND storages. Use the command to write a pack image into MA35D1 demo board after computer connected MA35D1 demo board. You can use “lsusb” command to check whether the connection board is working. **Using “nuwriter” or “dd” 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/som$ ls
openwrt-ma35d1-som-som-pack-spinand.bin
```

Burn openwrt-ma35d1-som-som-pack-spinand.bin image to SPINAND.

```
/OpenWrt/bin/targets/ma35d1/som $ nuwriter.py -a enc_ddr3_winbond_256mb.bin
/OpenWrt/bin/targets/ma35d1/som $ nuwriter.py -e spinand all
/OpenWrt/bin/targets/ma35d1/som $ nuwriter.py -w spinand openwrt-ma35d1-som-som-pack-spinand.bin
```


4 TEST OPENWRT

4.1 Booting Messages

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

```
[ 13.436387] NET: Registered protocol family 24
[ 13.450439] kmodloader: done loading kernel modules from /etc/modules.d/*
[ 17.812489] ma35d1-gmac 40120000.ethernet eth0: PHY [stmmac-0:01] driver [RTL8211F Gigabit Ethernet]
[ 17.839575] ma35d1-gmac 40120000.ethernet eth0: No Safety Features support found
[ 17.847031] ma35d1-gmac 40120000.ethernet eth0: No MAC Management Counters available
[ 17.854754] ma35d1-gmac 40120000.ethernet eth0: IEEE 1588-2008 Advanced Timestamp supported
[ 17.863062] ma35d1-gmac 40120000.ethernet eth0: configuring for phy/rgmii-id link mode
[ 18.930472] ma35d1-gmac 40120000.ethernet eth0: Link is Up - 100Mbps/Full - flow control off
[ 18.938912] IPv6: ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready

BusyBox v1.33.2 (2022-04-16 12:59:34 UTC) built-in shell (ash)

 _   _   _   _   _   _   _   _   _   _   _   _   _   _   _   _   _   _   _   _   _   _   _   _   _   _
| |   | |   | |   | |   | |   | |   | |   | |   | |   | |   | |   | |   | |   | |   | |   | |   | |
|_|  |_|  |_|  |_|  |_|  |_|  |_|  |_|  |_|  |_|  |_|  |_|  |_|  |_|  |_|  |_|  |_|  |_|  |_|  |_|  |_|
----- W I R E L E S S F R E E D O M -----

OpenWrt 21.02.3, r16554-1d4dea6d4f
=====
=== WARNING! =====
There is no root password defined on this device!
Use the "passwd" command to set up a new password
in order to prevent unauthorized SSH logins.
=====
root@OpenWrt:/#
```

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.

```
root@OpenWrt:/#  
root@OpenWrt:/# uci show network  
network.loopback=interface  
network.loopback.ifname='lo'  
network.loopback.proto='static'  
network.loopback.netmask='255.0.0.0'  
network.lan=interface  
network.lan.ifname='eth0'  
network.lan.proto='dhcp'  
network.lan.netmask='255.255.255.0'  
root@OpenWrt:/#
```

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.

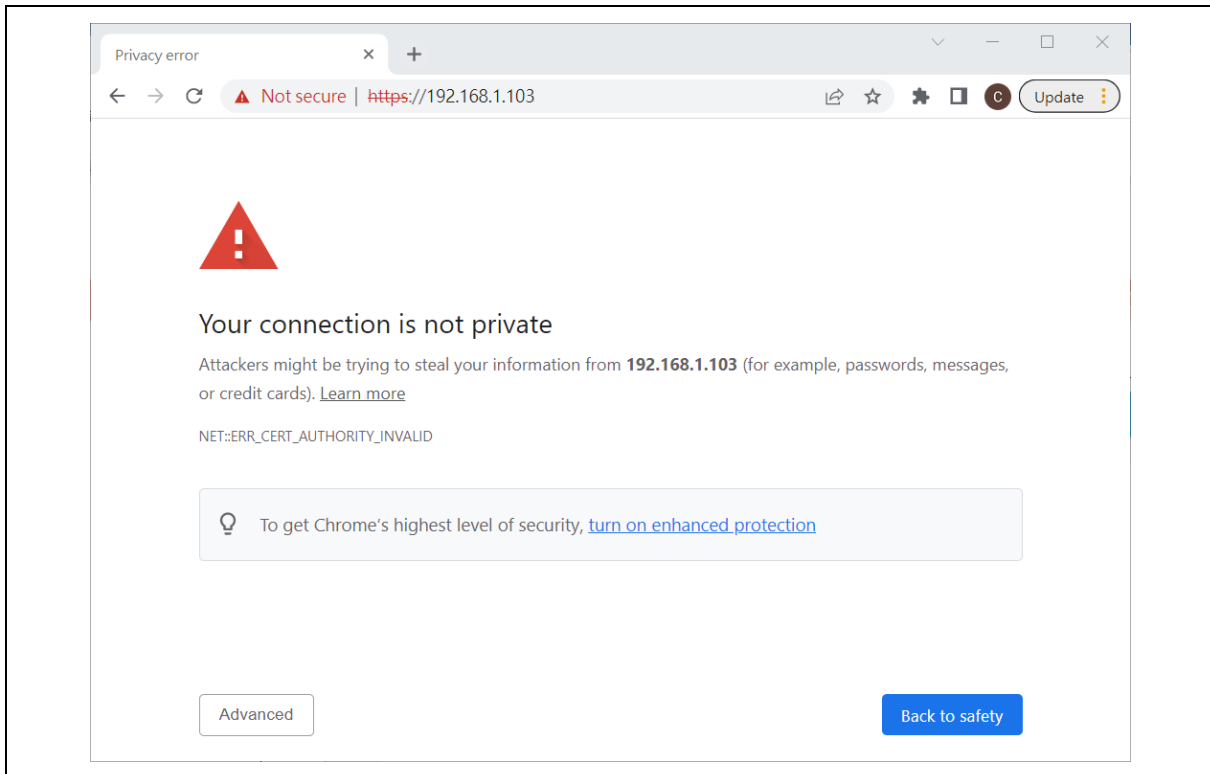


Figure 4-3 First Time to Login the LuCI Web Interface

After clicking the “Advanced” button, user will see a new screen, as shown in Figure 4-4. Please proceed with the connection.

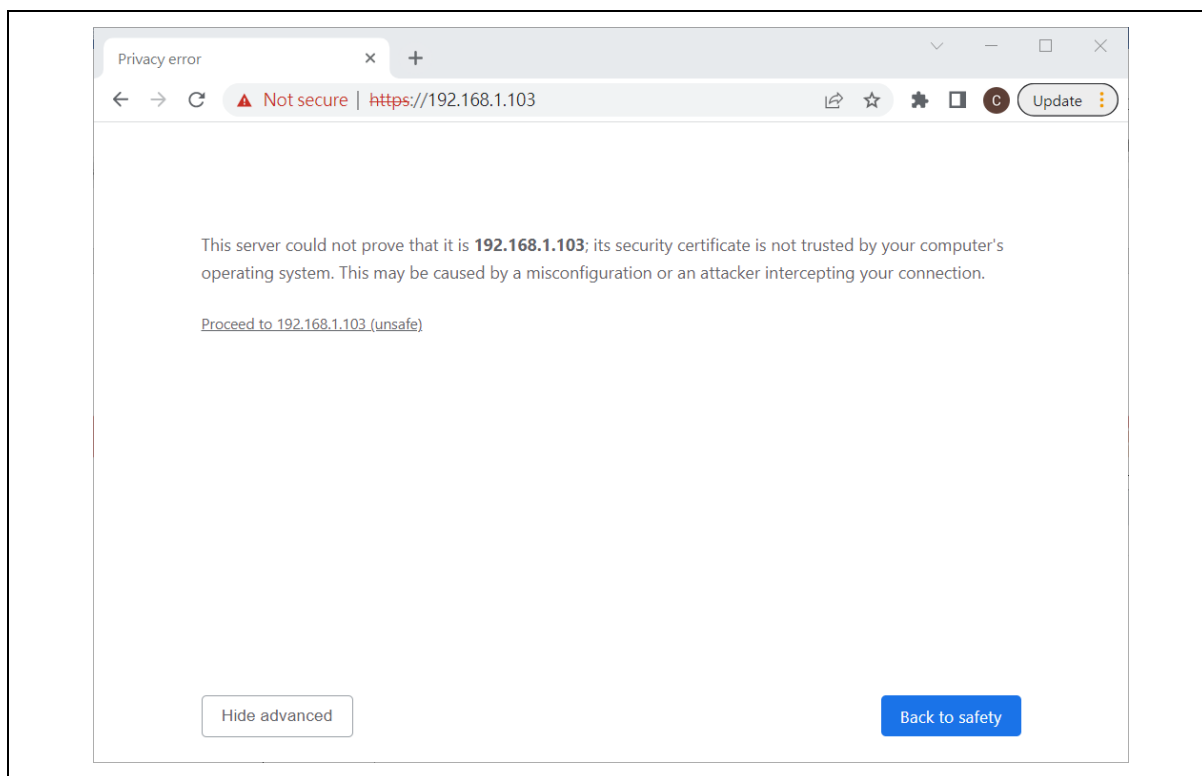


Figure 4-4 Proceed with Connection to LuCI

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.

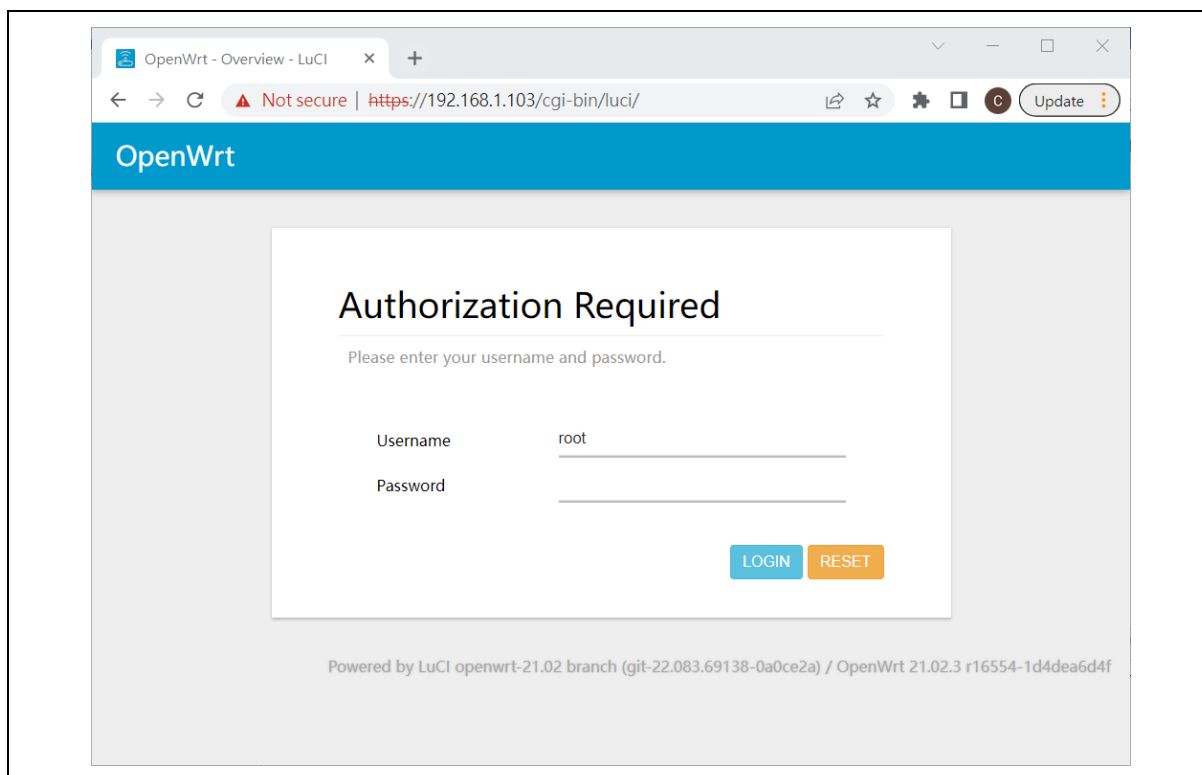


Figure 4-5 Login Screen of LuCI

4.4 Firmware Update

With the LuCI interface, user can upgrade the image which includes Linux kernel and OverlayFS root filesystem. For the MA35D1 SOM board, the image is as follows.

```
openwrt-ma35d1-som-som-squashfs-sysupgrade.bin
```

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.

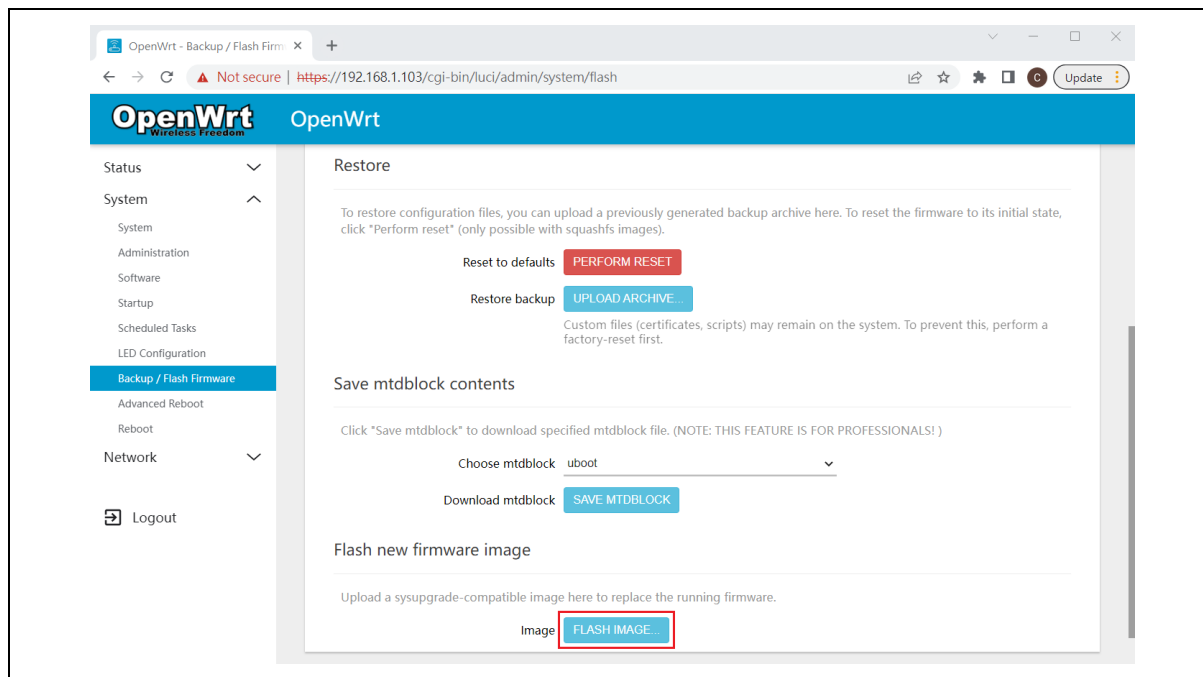


Figure 4-6 Firmware Upgrade in LuCI

5 REVISION HISTORY

Date	Revision	Description
2022.08.16	1.00	1. Preliminary release

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.

*Please note that all data and specifications are subject to change without notice.
All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.*