MA35D1 OTA Update by Linux Stage

NuMicro® 64/32位系列微控制器范例程序代码介绍

文件信息

应用简述	本范例程序说明如何于 MA35D1 系列开发平台下在 Linux 进行 OTA 更新。
BSP 版本	Linux-5.10.x
开发平台	NuMaker-IoT-MA35D1

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 microcontroller and 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

1. 概述

此范例程序说明文件提供使用「swupdate」应用程序在 MA35D1 的多个储存装置上更新 Linux Kernel、设备树以及根文件系统。储存装置包括 NAND、SPI NAND、SD 卡和 eMMC。更新的过程涉及生成包含固件更新的打包档以及描述固件更新的储存地址信息的头文件档。

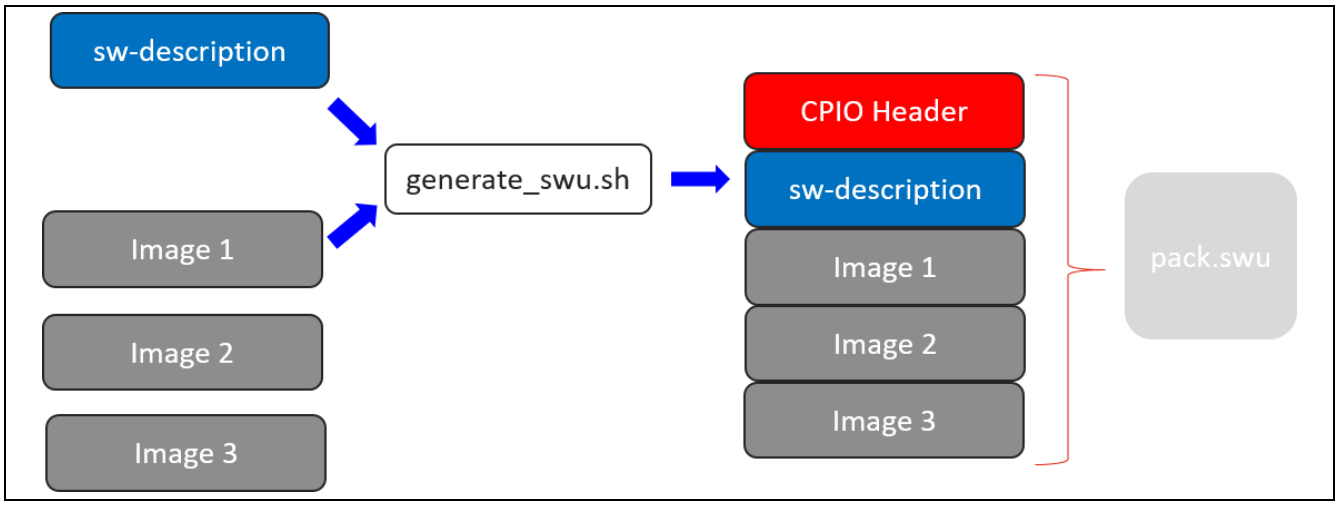


图 1-1 更新包生成流程

为了进行固件更新，首先需要生成包含已更新的 Kernel、设备树和根文件系统的打包档。可以经由脚本 （generate_swu.sh）将更新的档（Image1、Image2、Image3...）合并为一个档来完成。生成打包档后，下一步是创建一个标头文件，该档包含每个固件更新的储存地址信息。这个标头文件通常被命名为「sw-description」，并包含以下信息：

- 固件的版本
- 硬件兼容性信息
- 每个固件更新的文件信息，包括文件名、路径和储存位置

主机端生成 pack.swu 档后，打开网页浏览器，连接到运行在 MA35D1 开发板上的 swupdate 网页服务器。然后将 pack.swu 档拖放到 swupdate 网页上，swupdate 将开始接收该档并解析 sw-description 以将相应的档更新到对应的储存位置。

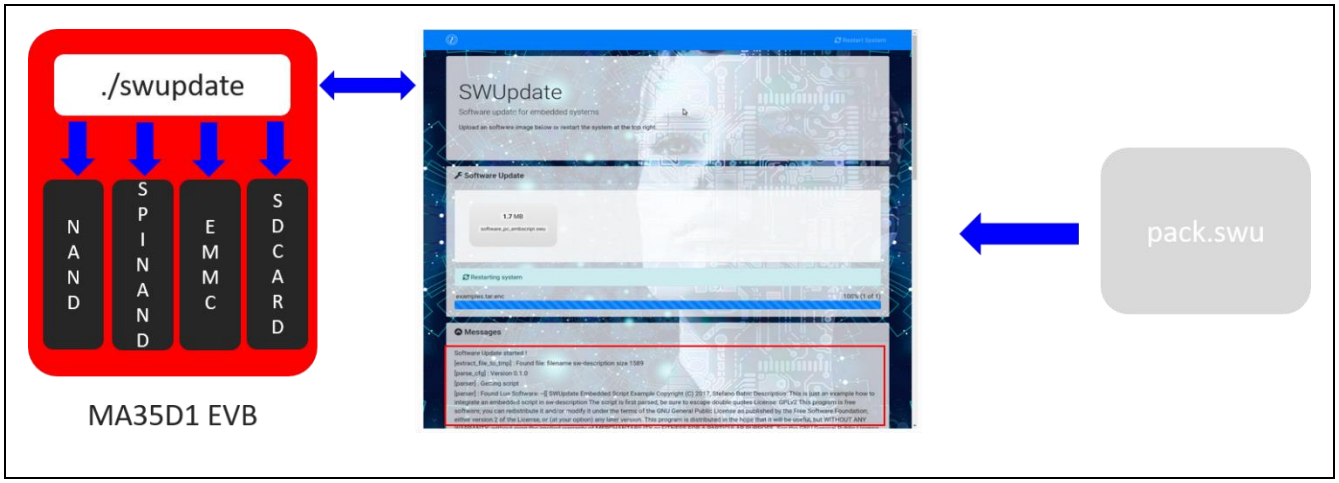


图 1-2 swupdate Web Server 更新框架

总体而言，使用「swupdate」应用程序在多个储存设备上进行固件更新涉及生成固件更新的打包档，并创建一个带有每个更新的储存地址信息的标头文件。拥有这些档后，可以使用「swupdate」应用程序对目标设备进行固件更新。

1.1 环境设定

MA35D1环境架设，产生映射档可参考新唐 MA35D1 Evaluation Board Quick Start 。

- **Buildroot**

于 Buildroot 的 menuconfig 中启用「swupdate」所需的套件，请按照以下步骤进行:

1. 透过执行 make menuconfig 命令来打开 Buildroot 的配置菜单。
2. 在套件选择菜单中，选择相应的选项并按下空格键来启用每个套件。可以使用 / 键按名称搜索套件。
   a. 启用 **BR2_PACKAGE_UBOOT_TOOLS_FWPRINTENV** 以包含 U-Boot Tools fw_printenv 套件。
   b. 启用 **BR2_PACKAGE_MTD** 以包含 MTD 套件。
   c. 启用 **BR2_PACKAGE_SWUPDATE** 以包含 swupdate 套件。
   d. 启用 **BR2_PACKAGE_MMC_UTILS** 以包含 mmc-utils 套件。
3. 启用所有所需的套件后，退出 menuconfig 并保存配置。
4. 通过 make 命令使用更新的配置来构建 Buildroot 系统。

编译完成后，使用 NuWriter 将映射烧录到 MA35D1 开发板

1.2 使用 sw-description 为不同储存类型创建固件更新包

sw-description 告知 swupdate 更新文件的位置。考虑到特定的存储设备（NAND/SPI NAND/SD 卡/eMMC），此章节将说明如何生成 sw-description 文件的过程。此外，也提供将 sw-description 档与固件映像打包在一起的范例，实现完整的固件更新包。

1.2.1 MA35D1 Storages Partition Layout

1.2.1.1 NAND

MA35D1 的 NAND partition layout可以在 ma35d1.dtsi 找到

```
nand:nand@401A0000 {
    …
    partitions {
        compatible = "fixed-partitions";
        #address-cells = <1>;
        #size-cells = <1>;

        uboot@0 {
                label = "nand-uboot";
            reg = <0x0000000 0x300000>;
            read-only;
        };
        uboot-env@300000 {
            label = "nand-uboot-env";
            reg = <0x300000 0xC0000>;
        };
        device-tree@3c0000 {
            label = "nand-device-tree";
            reg = <0x3C0000 0x40000>;
        };
        kernel@400000 {
            label = "nand-kernel";
            reg = <0x400000 0x1800000>;
        };
        rootfs@1c00000 {
            label = "nand-rootfs";
            reg = <0x1C00000 0x6400000>;
        };
    };
};
```

若要在 sw-description 描述更新 MA35D1 NAND 分区的内核，可以参考以下内容：

```
software =
{
        version = "0.1.0";
      description = "Linux kernel update in NAND";
        images: (
        {
                version ="5.10";
             name = "nand-kernel";
             filename = "Image";
             device = "/dev/mtdblock3";
             type = "raw";
        }
        );
}
```

此范例中，"Image" 是内核映像档案的文件名，"/dev/mtdblock3" 是实际内核映像档案的 mtdblock

1.2.1.2 SPI NAND

MA35D1 的 SPI NAND partition layout 可以在 ma35d1.dtsi 找到

```
qspi0: spi@40680000 {
    …
    flash: flash@0 {
         compatible = "spi-nand";
         reg = <0 0x0 0x2000000>;
         spi-ax-frequency = <50000000>;
         #address-cells = <1>;
         #size-cells = <1>;

         uboot@0 {
              label = "spinand-uboot";
              reg = <0x0 0x300000>;
         };
         uboot-env@300000 {
              label = "spinand-uboot-env";
              reg = <0x300000 0xc0000>;
         };
         device-tree@3c0000 {
              label = "spinand-device-tree";
              reg = <0x3c0000 0x40000>;
         };
         kernel@400000 {
              label = "spinand-kernel";
              reg = <0x400000 0x1800000>;
         };
         rootfs@1c00000 {
              label = "spinand-rootfs";
              reg = <0x1c00000 0x6400000>;
         };
    };
     …
};
```

若要在 sw-description 描述更新 MA35D1 SPI NAND 分区的内核，可以参考以下内容：

```
software =
{
        version = "0.1.0";
        description = " Linux kernel update in SPI NAND";
        images: (
          {
                  version ="5.10";
               name = "nand-kernel";
               filename = "Image";
               device = "/dev/mtdblock8";
               type = "raw";
          }
         );
}
```

此范例中，"Image" 是内核映射档的档名，"/dev/mtdblock8" 是指向实际内核映像档的 MTD 区块。

请注意，MTD（Memory Technology Device）区块编号可能会根据设备树的配置而有所不同。对于具有 NAND 和 SPI NAND 连接的 NuMaker-IoT-MA35D1，MTD 区块的编号如下：

NAND：MTD 区块的编号为 mtdblock0 到 mtdblock4。

SPI NAND：MTD 区块的编号为 mtdblock5 到 mtdblock9。

需参考设备树配置或目标设备来确定每个存储类型的正确 MTD 区块编号。这些信息将用于 sw-description 档案中，以指定用于固件更新的正确分区

1.2.1.3 eMMC/SDcard

MA35D1 的 eMMC/SDcard partition layout 可以在 pack-sdcard.json 找到

```
{
  "image": [
    …
    {
      "offset": "0x2c0000",
      "file": "Image.dtb",
      "type": 0
    },
    {
      "offset": "0x300000",
      "file": "Image",
      "type": 0
    },
    {
      "offset": "37748736",
      "file": "rootfs.ext4",
      "type": 0
    }
  ]
}
```

若要在 sw-description 描述更新 MA35D1 eMMC/SDcard 分区的内核，可以参考以下内容:

```
software =
{
        version = "0.1.0";
       description = " Linux kernel update in eMMC/SDcard";
        images: (
         {
              filename = "Image";
              device = "/dev/mmcblk0";
              offset = "0x300000";

         }
        );
}
```

1.2.2 固件映射的封装

为了将 sw-description 档和其他固件更新档封装成单一的包，请按照以下步骤进行操作:

1. 准备必要的文件：确保将 sw-description 档以及其他固件（如内核映像、设备树二进制文件和根文件系统映射）放在同一个文件夹中。

2. 创建封装脚本：开发一个脚本来处理封装过程。该脚本从指定的文件夹中收集档并将它们封装成单一的档。以下是一个示例:

```
#!/bin/sh
CONTAINER_VER="1.0"
PRODUCT_NAME="pack"
FILES="sw-description Image"
for i in $FILES;do
        echo $i;done | cpio -ov -H crc >  ${PRODUCT_NAME}_${CONTAINER_VER}.swu
```

此脚本会将 sw-description 和 Image 档打包成一个名为 pack.swu 的 cpio 格式档

3. 执行打包脚本：运行打包脚本来创建 pack.swu 档，该档包含了压缩的固件更新档和 sw-description 档。

将 sw-description 档和其他固件更新档放入同一个文件夹中，运行打包脚本，就可以生成 pack.swu 文件。该档是一个完整的固件更新包，可以通过 swupdate 应用程序轻松传输和应用。

2. 代码介绍

2.1 更新 Rootfs 中的文件/用户应用程序

```
/* sw-description for updating files/upser applicaitons */

/*denotes the beginning of the software section of the sw-description file*/
software =
{
    /*specifies the version number of the software to be updated*/
        version = "0.1.0";
/*specifies the hardware versions that are compatible with this software version*/
        hardware-compatibility = [ "1.0", "1.2", "1.3" ];

        files: (
        /* single copy mode */
         {
/*the name of the file/user application to be updated*/
                filename = "target.txt";
/*specifies the path where the file should be installed on the target system*/
                path = "/home/target.txt“
         },
        );
}
```

2.2 更新内核和设备树

2.2.1 NAND

```
/* sw-description for updating Kernel and device tree in NAND */

/* Denotes the beginning of the software section of the sw-description file */
software =
{
    /* Specifies the version number of the software to be updated */
    version = "0.1.0";
    /* Provides a brief description of the software update */
    description = "Linux kernel and dtb update for MA35 NAND";
    /* Specifies the hardware versions that are compatible with this software version */
    hardware-compatibility = [ "1.0", "1.2", "1.3" ];
    /* Begins the block for specifying the firmware update images */
    images: (
        {
            /* Specifies the version of the image. */
            version ="5.10";
            /* Provides a name or description for the image */
            name = "nand-kernel device tree";
            /* Specifies the filename of the image */
            filename = "Image.dtb";
                /* Specifies the device or storage location where the image will be
updated */
            device = "/dev/mtdblock2";
            /* Defines the type of the image file */
            type = "raw";
        },
        {
            /* Specifies the version of the image. */
            version ="5.10";
```

```
            /* Provides a name or description for the image */
            name = "nand-kernel";
            /* Specifies the filename of the image */
            filename = "Image";
            /* Specifies the device or storage location where the image will be updated
*/
            device = "/dev/mtdblock3";
            /* Defines the type of the image file */
            type = "raw";
        }
    );
}
```

2.2.2 SPI NAND

```
/* sw-description for updating Kernel and device tree in SPI NAND */

/* Denotes the beginning of the software section of the sw-description file */
software =
{
    /* Specifies the version number of the software to be updated */
    version = "0.1.0";
    /* Provides a brief description of the software update */
    description = "Linux kernel and dtb update for MA35 SPI NAND";
    /* Specifies the hardware versions that are compatible with this software version */
    hardware-compatibility = [ "1.0", "1.2", "1.3" ];
    /* Begins the block for specifying the firmware update images */
    images: (
        {
            /* Specifies the version of the image. */
            version ="5.10";
            /* Provides a name or description for the image */
            name = "spinand-kernel device tree";
            /* Specifies the filename of the image */
            filename = "Image.dtb";
            /* Specifies the device or storage location where the image will be updated
*/
            device = "/dev/mtdblock7";
            /* Defines the type of the image file */
            type = "raw";
        },
        {
            /* Specifies the version of the image. */
            version ="5.10";
            /* Provides a name or description for the image */
            name = "spinand-kernel";
            /* Specifies the filename of the image */
            filename = "Image";
            /* Specifies the device or storage location where the image will be updated
*/
            device = "/dev/mtdblock8";
            /* Defines the type of the image file */
            type = "raw";
        }
    );
}
```

2.2.3 eMMC/SDcard

```
/* sw-description for updating Kernel and device tree in eMMC/SDcard */

/* Denotes the beginning of the software section of the sw-description file */
software =
{
    /* Specifies the version number of the software to be updated */
    version = "0.1.0";
    /* Provides a brief description of the software update */
    description = "Linux kernel and dtb update for MA35 eMMC/SDcard";
    /* Specifies the hardware versions that are compatible with this software version */
    hardware-compatibility = [ "1.0", "1.2", "1.3" ];
    /* Begins the block for specifying the firmware update images */
    images: (
        {
            /* Specifies the filename of the image */
                filename = "Image.dtb";
            /* Specifies the device or storage location where the image will be updated
*/
                device = "/dev/mmcblk0";
            /* Specifies the offset at which the image will be written in the storage
device */
                offset = "0x2c0000";
        },
        {
                /* Specifies the filename of the image */
                filename = "Image";
                /* Specifies the device or storage location where the image will be
updated */
                device = "/dev/mmcblk0";
                /* Specifies the offset at which the image will be written in the storage
device */
                offset = "0x300000"
        }
    );
}
```

2.3 固件打包脚本

该脚本从指定的文件夹中收集档并将它们打包成一个单独的档.

```
# generate_swu.sh
# Indicating that the script should be executed using the shell interpreter specified
#!/bin/sh
# Assigns the value "1.0" to the variable CONTAINER_VER, which represents the version of
the container being created
CONTAINER_VER="1.0"
# Assigns the value "pack" to the variable PRODUCT_NAME, which represents the name of the
product or container being created
PRODUCT_NAME="pack"
# Assigns the value "sw-description Image" to the variable FILES, which contains a space-
separated list of filenames to be included in the container
FILES="sw-description Image Image.dtb"
# Starts a loop that iterates over each element in the FILES variable
# | cpio -ov -H crc: Pipes the output of the loop to the cpio command, which creates a
cpio archive. The -ov options specify that the output should be in verbose mode and
overwrite existing files. The -H crc option specifies the header format to use, in this
case, CRC.
for i in $FILES;do
        echo $i;done | cpio -ov -H crc >  ${PRODUCT_NAME}_${CONTAINER_VER}.swu
```

3. 软件与硬件需求

3.1 软件需求

- BSP 版本
  - Linux-5.10.x

3.2 硬件需求

- 电路组件
  - NuMaker-IoT-MA35D1

4. 目录信息

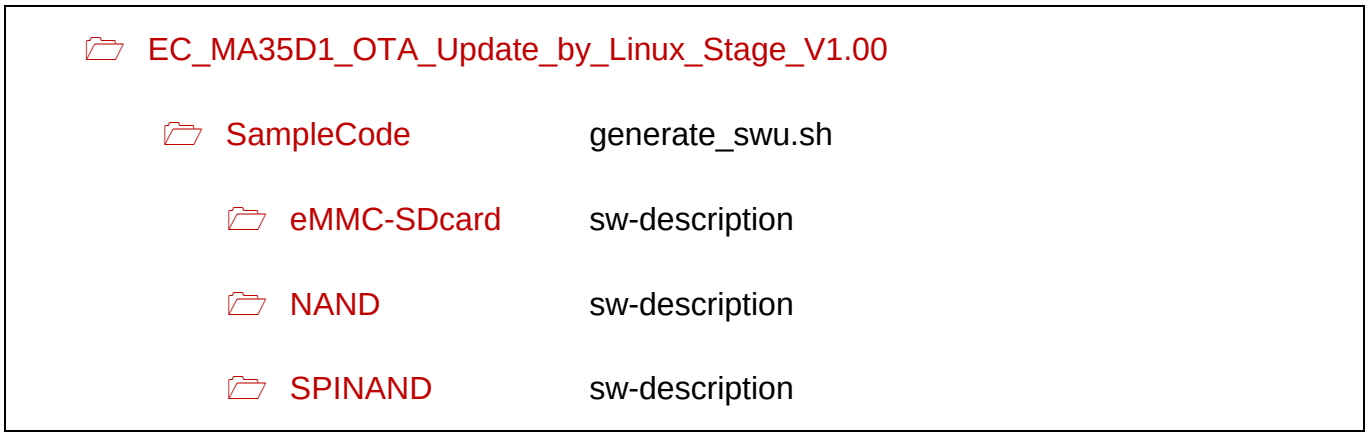


图 4-1 目录信息

5. 范例程序执行

于主机端打包映像:

要在主机端进行映像打包，需要将 sw-description、Image、Image.dtb 和 generate.sh 文件放在同一个文件夹中。这样可以确保在 PC 主机上的打包过程中所有所需的档都可以访问到:

```
      @ubuntu:              swupdate       $ ls -l
total 10044
-rwxrw-rw- 1                     196 May  9 14:18 generate_swu.sh
-rwxrw-rw- 1                10232320 Oct 31  2022 Image
-rwxrw-rw- 1                   37525 Oct 31  2022 Image.dtb
-rwxrw-rw- 1                     607 Oct 31  2022 sw-description
```

图 5-1 映像档的文件夹内容(1)

执行 generate_swu.sh 脚本后，最终的映射套件将在当前文件夹建立。该脚本负责执行必要的步骤以生成套件，这可能包括档操作、压缩或其他操作。当脚本成功执行后，将在当前目录中找到生成的映射套件。

```
ubuntu:~$ ./generate_swu.sh
```

```
      @ubuntu:              swupdate/      $ ./generate_swu.sh
sw-description
Image
Image.dtb
20061 blocks
      @ubuntu:              swupdate/      $ ls -l
total 20076
-rwxrw-rw- 1                     196 May  9 14:18 generate_swu.sh
-rwxrw-rw- 1                10232320 Oct 31  2022 Image
-rwxrw-rw- 1                   37525 Oct 31  2022 Image.dtb
-rw-rw-r-- 1                10271232 May  9 14:22 Linux_kernel_dtb_update_1.0.swu
-rwxrw-rw- 1                     607 Oct 31  2022 sw-description
```

图 5-2 映像档的文件夹内容(2)

在 MA35 装置上，可以通过配置设备使用 DHCP（动态主机配置协议）从路由器获取 IP 地址。

```
MA35D1# ifconfig eth0 up
MA35D1# udhcpc -i eth0
udhcpc: started, v1.33.1
udhcpc: sending discover
udhcpc: sending discover
udhcpc: sending select for 192.168.0.175
udhcpc: lease of 192.168.0.175 obtained, lease time 7200
deleting routers
adding dns 192.168.0.1
```

系统初始化之后，swupdate 进程会在后台运行。要列出正在运行的 swupdate 进程，可以使用 ps 命令，该命令会显示系统上活动进程的相关信息。以下是列出 swupdate 进程的示例命令:

```
MA35D1# ps aux | grep swupdate
  212 root     /usr/bin/swupdate -v -w -r /var/www/swupdate -p 8080
  219 root     /usr/bin/swupdate -v -w -r /var/www/swupdate -p 8080
  258 root     grep swupdate
```

要从计算机访问在 MA35D1 设备上运行的 swupdate 网页服务器，可以打开一个网页浏览器，然后输入 MA35D1 设备的 IP 地址，后面加上 ":8080"。以下是具体步骤:

1. 确保计算机和 MA35D1 设备连接到同一个网络。
2. 在计算机上打开一个网页浏览器（如 Edge、Chrome、Firefox 等）。
3. 在网页浏览器的地址栏中，输入 MA35D1 设备的 IP 地址，后面加上 ":8080"。例如，如果 MA35D1 设备的 IP 地址是 192.168.0.175，则输入 "192.168.0.175:8080"。
4. 网页浏览器将与在 MA35D1 设备上运行的 swupdate 网页服务器建立连接。

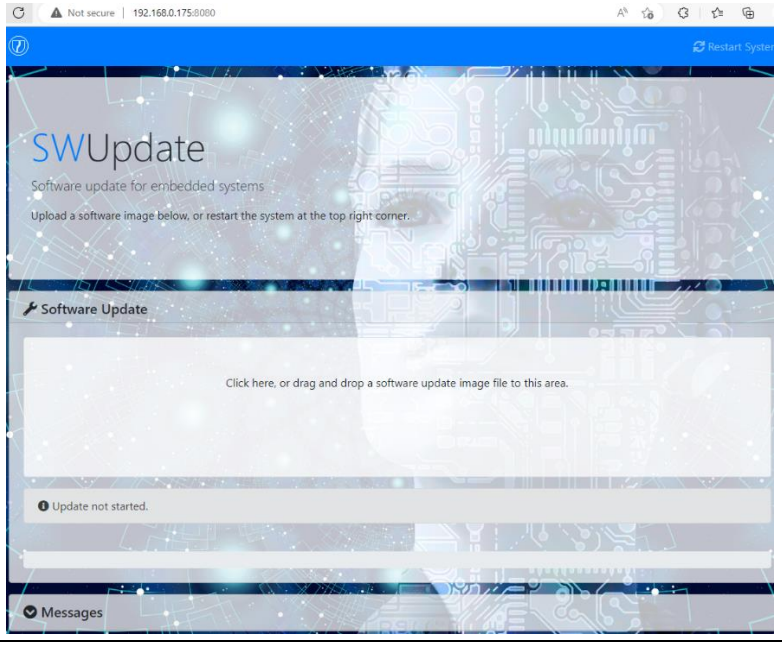


图 5-3 swupdate 网页端

访问 swupdate 网页服务器执行一系列操作，例如上传 pack.swu 固件套件进行更新、启动更新流程以及监控更新的进度或状态。

要将 pack.swu 固件套件传输到 MA35D1 设备上，只需在浏览器中的 swupdate 网页上将 pack.swu 档案拖放即可。

以下为操作实例画面:

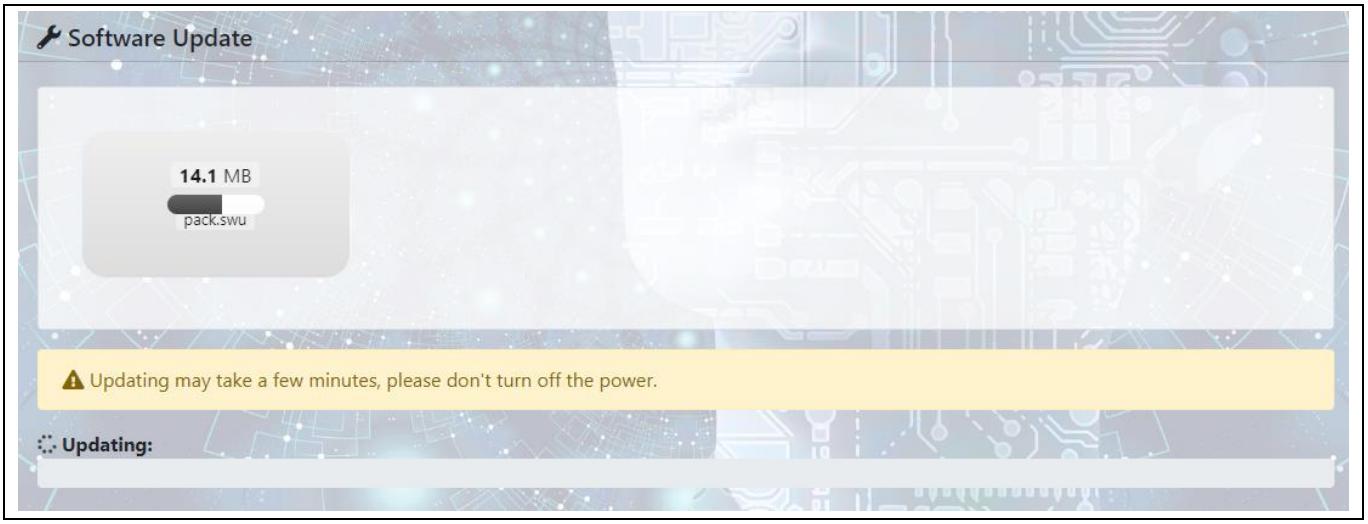
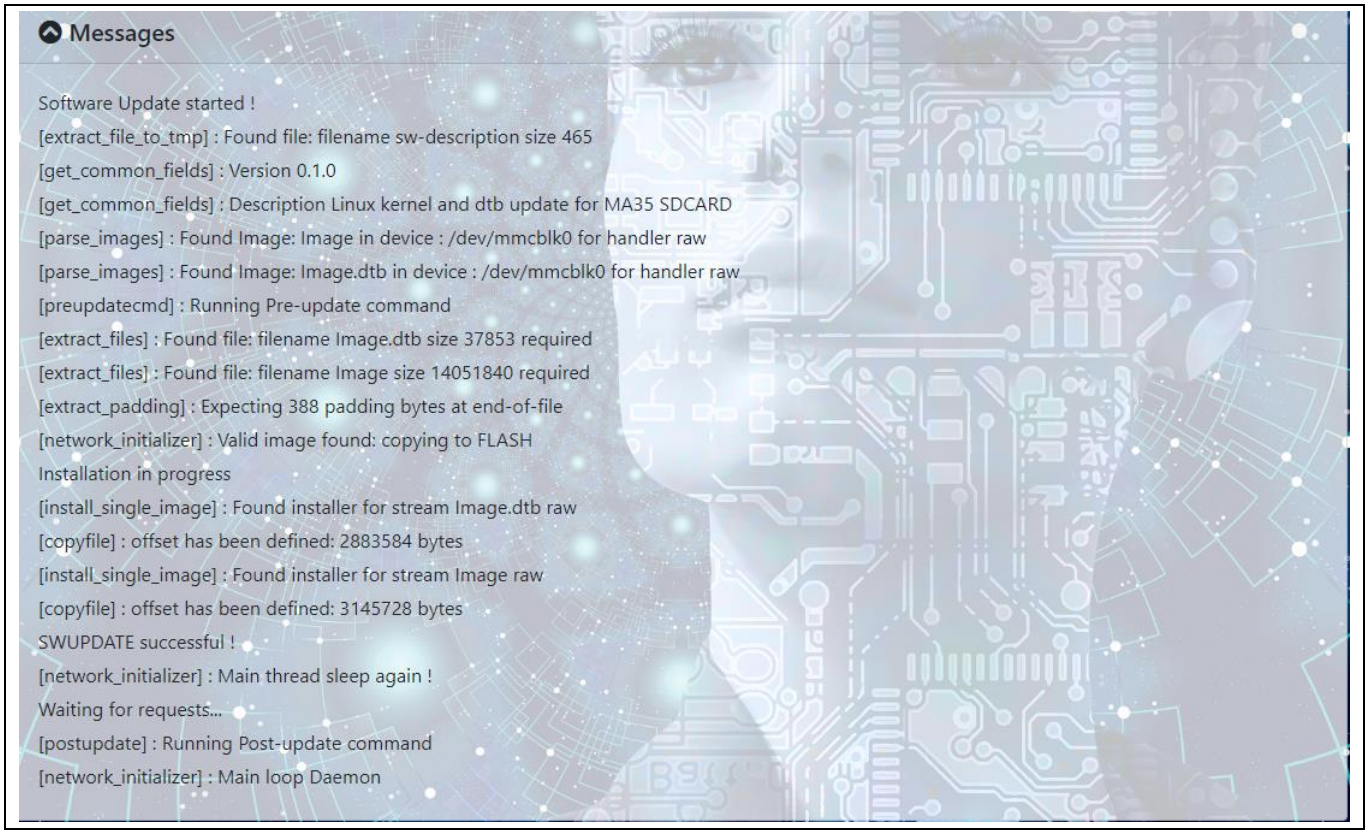


图 5-4 网页端更新中

图 5-5 网页端更新完成



5-6 网页端更新过程记录

6. 修订记录

Date	Revision	Description
2023.05.09	1.00	初始发布。

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.