Lierda IC610 EVK 软件应用开发指导

产品名称：ST-A35-IC610 工业核心板

产品型号：L-IDMIM0-AA185

版本：Rev1.0

日期：25/03/15

状态：受控版本

公司地址：杭州市文一西路 1326 号利尔达物联网科技园　Tel：0571-88800000

法律声明

若接收利尔达科技集团股份有限公司(以下称为“利尔达”)的此份文档，即表示您已经同意以下条款。若不同意以下条款，请停止使用本文档。

本文档版权归利尔达科技集团股份有限公司所有，保留任何未在本文档中明示授予的权利。文档中涉及利尔达的专有信息。未经利尔达事先书面许可，任何单位和个人不得复制、传递、分发、使用和泄漏该文档以及该文档包含的任何图片、表格、数据及其他信息。

本产品符合有关环境保护和人身安全方面的设计要求，产品的存放、使用和弃置应遵照产品手册、相关合同或者相关法律、法规的要求进行。

本公司保留在不预先通知的情况下，对此手册中描述的产品进行修改和改进的权利；同时保留随时修订或收回本手册的权利。

Lierda 利尔达

文件修订历史

文档版本	变更日期	修订人	审核人	变更内容
Rev1.0	25-03-15	YQA		初始版本
Rev1.0	25-06-5	YQA		优化源码编译步骤，增加 tf-a 编译错误提示

Lierda 利尔达

目录

法律声明 1
文件修订历史 2
目录 3
1 引言 4
2 A35 开发环境搭建 4
2.1 交叉编译器安装 4
2.2 源码准备及自动编译 5
2.3 tf-a 6
2.4 optee 7
2.5 u-boot 7
2.6 Kernel 编译 8
2.7 ic610-images 简介 10
2.7.1 烧录镜像更新到 windows 11
2.7.2 自定义文件系统 13
2.7.3 sd 卡启动卡镜像 raw 制作 13
3 系统镜像更新及系统工具安装更新 15
3.1 OTG 烧录 15
3.1.1 STM32CubeProgrammer 安装 15
3.1.2 STM32CubeProgrammer 烧录镜像 21
3.2 SD 卡启动卡烧录 26
3.2.1 SD 卡启动卡制作 26
3.2.2 SD 启动卡烧写 emmc 30
4 cubemx 安装 32
4.1.1 Cubemx gpio 使用 38
4.2 Cubemx dts 开发 39

1 引言

本文档依托 IC610 evk，旨在搭建 A35 linux SDK，包括 Ubuntu 环境搭建、sdk 源码准备、tf-a、optee、u-boot、linux 源码编译及烧录镜像制作、烧录工具安装、镜像烧录等。

2 A35 开发环境搭建

Ubuntu 交叉编译环境安装，本开发环境已经在 ubuntu1804、Ubuntu2004 验证。

```
sudo apt-get update
sudo apt-get install gawk wget git diffstat unzip texinfo gcc-multilib chrpath socat cpio python3 python3-pip python3-pexpect libssl-dev libgmp-dev libmpc-dev lz4 zstd build-essential libncurses-dev libyaml-dev libssl-dev
sudo apt-get install coreutils bsdmainutils sed curl bc lrzsz corkscrew cvs subversion mercurial nfs-common nfs-kernel-server libarchive-zip-perl dos2unix texi2html libxml2-utils mtd-utils vim net-tools
sudo apt-get install git-lfs libsdl1.2-dev pylint python3-git xterm
```

gith 环境配置,具体账号、名称可根据实际修改，未设置源码构建中无法提交 git 。

```
git config --global user.email "lsd@lierda.com"
git config --global user.name "lsd"
```

2.1 交叉编译器安装

ic610-sdk\tools\en.SDK-x86_64-stm32mp2-openstlinux-6.6-yocto-scarthgap-mpu-v24.11.06.tar.gz 为交叉编译器，将该文件复制到 ubuntu 下。

```
~/st/stm32mp2$
$: tar -zxvf en.SDK-x86_64-stm32mp2-openstlinux-6.6-yocto-scarthgap-mpu-v24.11.06.tar.gz
$: cd stm32mp2-openstlinux-6.6-yocto-scarthgap-mpu-v24.11.06/sdk/
$: sudo chown lsd:lsd /opt
```

$: ./st-image-weston-openstlinux-weston-stm32mp2.rootfs-x86_64-toolchain-5.0.3-openstlinux-6.6-yocto-scarthgap-mpu-v24.11.06.sh

注意：lsd 为 ubuntu 用户，配置时以实际用户名为准。

根据示信息输入 y

```
yqa@yqa1804:~/st/stm32mp2/stm32mp2-6.1-v24.06.26/sdk$ ./st-image-weston-openstlinux-weston-stm32mp2-x86_64-toolchain-4.2.4-openstlinux-6.1-yocto-mickledore-mpu-v24.06.26.sh
ST OpenSTLinux - Weston - (A Yocto Project Based Distro) SDK installer version 4.2.4-openstlinux-6.1-yocto-mickledore-mpu-v24.06.26
==========================================================================================================================
Enter target directory for SDK (default: /opt/st/stm32mp2/4.2.4-openstlinux-6.1-yocto-mickledore-mpu-v24.06.26):
You are about to install the SDK to "/opt/st/stm32mp2/4.2.4-openstlinux-6.1-yocto-mickledore-mpu-v24.06.26". Proceed [Y/n]? y
Extracting SDK.....
```

使能交叉编译器

$: source /opt/st/stm32mp2/5.0.3-openstlinux-6.6-yocto-scarthgap-mpu-v24.11.06/environment-setup-cortexa35-ostl-linux

交叉编译安装测试：

yqa@yqa1804:~$ aarch64-ostl-linux-gcc -v

如下，表示交叉编译器安装完成。

```
yqa@yqa1804:~/st/stm32mp2/sdk_v24.11.06$ aarch64-ostl-linux-gcc -v
Using built-in specs.
COLLECT_GCC=aarch64-ostl-linux-gcc
COLLECT_LTO_WRAPPER=/opt/st/stm32mp2/5.0.3-openstlinux-6.6-yocto-scarthgap-mpu-v24.11.06/sysro
Target: aarch64-ostl-linux
Configured with: ../../../../../../work-shared/gcc-13.3.0-r0/gcc-13.3.0/configure --build=x86_
d-buildpath/sysroots/x86_64-ostl_sdk-linux/usr --exec_prefix=/usr/local/oe-sdk-hardcoded-build
4-ostl_sdk-linux/usr/bin/aarch64-ostl-linux --sbindir=/usr/local/oe-sdk-hardcoded-buildpath/sy
dpath/sysroots/x86_64-ostl_sdk-linux/usr/libexec/aarch64-ostl-linux --datadir=/usr/local/oe-sd
ed-buildpath/sysroots/x86_64-ostl_sdk-linux/etc --sharedstatedir=/usr/local/oe-sdk-hardcoded-b
roots/x86_64-ostl_sdk-linux/var --libdir=/usr/local/oe-sdk-hardcoded-buildpath/sysroots/x86_64
s/x86_64-ostl_sdk-linux/usr/include --oldincludedir=/usr/local/oe-sdk-hardcoded-buildpath/sysr
4-ostl_sdk-linux/usr/share/info --mandir=/usr/local/oe-sdk-hardcoded-buildpath/sysroots/x86_64
root=/local/mpuostlqa/gnbsx51818/workspace/ostl-build-cache/workdir/ostl/build/tmp-glibc/work/
le-shared --enable-languages=c,c++ --enable-threads=posix --enable-multilib --enable-c99 --ena
out-local-prefix --disable-install-libiberty --disable-libssp --enable-libitm --enable-lto --d
no --with-cloog=no --enable-checking=release --enable-cheaders=c_global --without-isl --with-g
workspace/ostl-build-cache/workdir/ostl/build/tmp-glibc/work/x86_64-nativesdk-ostl_sdk-linux/g
/exist --with-build-sysroot=/local/mpuostlqa/gnbsx51818/workspace/ostl-build-cache/workdir/ost
ot --with-plugin-ld=ld --enable-poison-system-directories --disable-static --enable-nls --with
Thread model: posix
Supported LTO compression algorithms: zlib zstd
gcc version 13.3.0 (GCC)
yqa@yqa1804:~/st/stm32mp2/sdk_v24.11.06$
yqa@yqa1804:~/st/stm32mp2/sdk_v24.11.06$
yqa@yqa1804:~/st/stm32mp2/sdk_v24.11.06$
```

2.2 源码准备

ic610-sdk\src\src-ic610 为 sdk 源码

将 src-ic610 复制到 Ubuntu 下解压

$cd src-ic610

src-ic610$ chmod 775 *.sh

```
yqa@yqa1804:~/st/stm32mp2/sources-tm32mp2-6.1-v24.06.26/src-ic610$
yqa@yqa1804:~/st/stm32mp2/sources-tm32mp2-6.1-v24.06.26/src-ic610$ ls
build.sh                                          optee-os-stm32mp-3.19.0-stm32mp-r2-r0
gcnano-driver-stm32mp-6.4.15-stm32mp2-r1-r0       prepara_src.sh
get_src.sh                                        stm32mp-ddr-phy-A2022.11-r0
ic610-images                                      tf-a-stm32mp-v2.8.15-stm32mp-r2-r0
linux-stm32mp-6.1.82-stm32mp-r2-r0                u-boot-stm32mp-v2022.10-stm32mp-r2-r0
yqa@yqa1804:~/st/stm32mp2/sources-tm32mp2-6.1-v24.06.26/src-ic610$
```

修改 sudo 密码

```
$vim   passwd
```

yqa12345

上述 yqa12345 为 Ubuntu sudo 密码，修改为当前开发环境下实际密码。

运行 prepara_src.sh

```
src-ic610$   ./prepara_src.sh
```

自动解压相关源码。

注意：

tf-a、optee 及 u-boot 编译会存在镜像依赖，故首次编译需要对 tf-a optee u-boot 单步编译，且首次编译会报错退出，无需关注，编译退出后继续全部首次编译即可，首次编译后可通过 2.7 自动编译则不再报错，后续单步编译或自动编译不再报错。

2.3 tf-a 编译

```
src-ic610$ cd tf-a-stm32mp-v2.10.5-stm32mp-r1-r0/tf-a-stm32mp-v2.10.5-stm32mp-r1/
```

编译：

```
$: ./build.sh
```

注意：

首次编译报错退出，是正常现象，继续编译 optee、u-boot 即可。

```
   --search-devicetree stm32mp257f-ev1 --search-soc-name stm32mp25\
   --sign --signature-key key/stm32mp25/edmk-fip.bin \
   --output deploy-dir/fip
STM32MP257F-EV1 optee sdcard signed encrypted
/opt/st/stm32mp2/5.0.3-openstlinux-6.6-yocto-scarthgap-mpu-v24.11.06/sysroots/x86_64-ostl_sdk-linux/usr/bin/create_st_fip_binary.sh --use-ddr --generate-only-ddr --use-bl31 \
   --search-configuration optee-sdcard --search-storage optee-sdcard \
   --search-devicetree stm32mp257f-ev1 --search-soc-name stm32mp25 \
   --sign --signature-key key/stm32mp25/edmk-fip.bin -E key/stm32mp25/edmk.bin \
   --output deploy-dir/fip
/opt/st/stm32mp2/5.0.3-openstlinux-6.6-yocto-scarthgap-mpu-v24.11.06/sysroots/x86_64-ostl_sdk-linux/usr/bin/create_st_fip_binary.sh --use-ddr --use-bl31 \
   --search-configuration optee-sdcard --search-secondary-config default:optee --search-storage optee-sdcard \
   --search-devicetree stm32mp257f-ev1 --search-soc-name stm32mp25 \
   --sign --signature-key key/stm32mp25/edmk-fip.bin -E key/stm32mp25/edmk.bin\
   --output deploy-dir/fip
STM32MP257F-EV1 fastboot sdcard
/opt/st/stm32mp2/5.0.3-openstlinux-6.6-yocto-scarthgap-mpu-v24.11.06/sysroots/x86_64-ostl_sdk-linux/usr/bin/create_st_fip_binary.sh --use-ddr --generate-only-ddr --use-bl31 \
   --search-configuration fastboot-sdcard --search-storage optee-sdcard \
   --search-devicetree stm32mp257f-ev1 --search-soc-name stm32mp25 \
   --output deploy-dir/fip
/opt/st/stm32mp2/5.0.3-openstlinux-6.6-yocto-scarthgap-mpu-v24.11.06/sysroots/x86_64-ostl_sdk-linux/usr/bin/create_st_fip_binary.sh --use-ddr --use-bl31 \
   --search-configuration fastboot-sdcard --search-secondary-config fastboot-sdcard:optee --search-storage optee-sdcard \
   --search-devicetree stm32mp257f-ev1 --search-soc-name stm32mp25 \
   --output deploy-dir/fip
/home/yqa/st/stm32mp2/sdk_v24.11.06/tf-a-stm32mp-v2.10.5-stm32mp-r1-r0/tf-a-stm32mp-v2.10.5-stm32mp-r1/../Makefile.sdk:295: recipe for target 'fip' failed
make: *** [fip] Error 1
```

编译完成后

```
$: ls ../../ic610-images/arm-trusted-firmware/
```

metadata.bin tf-a-stm32mp255d-ic610-mx-optee-emmc.stm32 tf-a-stm32mp255d-ic610-mx-optee-sdcard.stm32 tf-a-stm32mp255d-ic610-mx-usb.stm32

2.4 Optee 编译

```
src-ic610$ cd optee-os-stm32mp-4.0.0-stm32mp-r1-r0/optee-os-stm32mp-4.0.0-stm32mp-r1/
```

编译脚本：

```
$: ./build.sh
```

编译完成后生成

```
$: ls ../../ic610-images/optee/
```

tee-header_v2-stm32mp255d-ic610-mx.bin tee-pageable_v2-sstm32mp255d-ic610-mx.bin tee-pager_v2-stm32mp255d-ic610-mx.bin

tee-header_v2-stm32mp255d-ic610-mx.bin tee-pageable_v2-stm32mp255d-ic610-mx.bin tee-pager_v2-stm32mp255d-ic610-mx.bin

2.5 u-boot 编译

```
src-ic610$ cd u-boot-stm32mp-v2023.10-stm32mp-r1-r0/u-boot-stm32mp-v2023.10-stm32mp-r1
```

编译：

```
$: ./build.sh
```

编译完成后生成

```
$: ls ../../ic610-images/fip/
```

fip-stm32mp255d-ic610-mx-ddr-optee-emmc.bin fip-stm32mp255d-ic610-mx-ddr-optee-sdcard.bin fip-stm32mp255d-ic610-mx-optee-emmc.bin fip-stm32mp255d-ic610-mx-optee-sdcard.bin

注意：stm32mp255d-ic610-mx 为要编译的 dts 名称，编译其他 dts 修改该参数即可；同时需要在 arch/arm/dts/Makefile　dtb-$(CONFIG_STM32MP25X) += \，添加对应 dts。

如新 dts 为 stm32mp255d-lsd-mx

vim　arch/arm/dts/Makefile

```
dtb-$(CONFIG_STM32MP25X) +=
        stm32mp257f-dk.dtb
        stm32mp257f-ic610.dtb
        stm32mp255d-lsd-mx.dtb
        stm32mp257f-ev1.dtb

dtb-$(CONFIG_SOC_K3_AM654) +=
```

2.6 Kernel 编译

```
src-ic610$:   cd  linux-6.6.48/linux-6.6.48/
```

编译命令

```
$:  ./build.sh
```

编译完成后镜像为

```
$:  ../../ic610-images/bootfs/
```

Image.gz　stm32mp255d-ic610-mx.dtb　stm32mp257f-dk.dtb　stm32mp257f-ev1.dtb

镜像自动更新到烧录镜像 st-image-bootfs-openstlinux-weston-stm32mp2.ext4

该 build.sh 脚本自动编译 kernel、wifi 模组驱动、gpu 驱动及更新文件系统等，用户可自行修改简化该脚本。

注意：build.sh 默认编译 dts 为 arch/arm64/boot/dts/st/stm32mp255d-ic610-mx.dts，编译其他 dts 修改该参数即可；同时需要在 arch/arm64/boot/dts/st/Makefile　dtb-$(CONFIG_ARCH_STM32) += \，添加对应 dts。

如新 dts 为 stm32mp255d-lsd-mx

vim　arch/arm/dts/Makefile

```
yqa@yqa1804:~/st/stm32mp2/sources-tm32mp2-6.1
# SPDX-License-Identifier: GPL-2.0-only
dtb-$(CONFIG_ARCH_STM32) += \
        stm32mp257f-dk.dtb \
        stm32mp257f-ic610.dtb \
        stm32mp255d-lsd-mx.dtb\
        stm32mp257f-ev1.dtb
```

Kernel 配置：

```
source /opt/st/stm32mp2/5.0.3-openstlinux-6.6-yocto-scarthgap-mpu-v24.11.06/environment-setup-cortexa35-ostl-linux
make menuconfig
```

Kernel 单独编译 dtb：

```
make dtbs
```

编译完成后 log 输出重新编译的 dtb 镜像，

如 arch/arm64/boot/dts/st/stm32mp255d-ic610-mx.dtb 。

Kernel 单独编译 Image.gz：

```
make Image.gz vmlinux  LOADADDR=0xC2000040
```

编译完成后生成：arch/arm64/boot/Image.gz

注意：

单独编译后镜像未自动打包到 ic610-images/st-image-bootfs-openstlinux-weston-stm32mp2.ext4 下，需要手动打包镜像。

source /opt/st/stm32mp2/5.0.3-openstlinux-6.6-yocto-scarthgap-mpu-v24.11.06/environment-setup-cortexa35-ostl-linux

为使能交叉编译器环境，在同一个终端下执行一次即可，若打开新的终端需要重新执行。

2.7 自动编译

src-ic610 下 build.sh 为自动编译全部镜像脚本，运行 build.sh 自动编译全部镜像，单步编译可参考软件用户手册。

```
src-ic610$ ./build.sh
```

注意：build.sh 默认编译的 dts 为 stm32mp255d-ic610-mx，编译其他 dts 需要修改 build.sh 内变量 dts 或增加 dts 参数，以下无需还行，如

~~src-ic610$ ./build.sh stm32mp255d-xxx~~

其他设备树需要参考 3.7.3 修改 tsv 文件生成 raw 镜像。

编译完成后生成全部镜像：

```
ic610-images/FlashLayout_sdcard_stm32mp255d-ic610-mx-optee.raw
ic610-images/fip/
fip-stm32mp255d-ic610-mx-ddr-optee-emmc.bin    fip-stm32mp255d-ic610-mx-optee-emmc.bin
fip-stm32mp255d-ic610-mx-ddr-optee-sdcard.bin  fip-stm32mp255d-ic610-mx-optee-sdcard.bin

ic610-images/arm-trusted-firmware/
metadata.bin  tf-a-stm32mp255d-ic610-mx-optee-emmc.stm32  tf-a-stm32mp255d-ic610-mx-optee-sdcard.stm32  tf-a-stm32mp255d-ic610-mx-usb.stm32

st-image-bootfs-openstlinux-weston-stm32mp2.ext4
st-image-vendorfs-openstlinux-weston-stm32mp2.ext4
st-image-weston-openstlinux-weston-stm32mp2.ext4
```

2.8 ic610-images 简介

目录	功能	备注
arm-trusted-firmware	tf-a 镜像目录	源码编译镜像自动更新至此
optee	optee 镜像目录	源码编译镜像自动更新至此

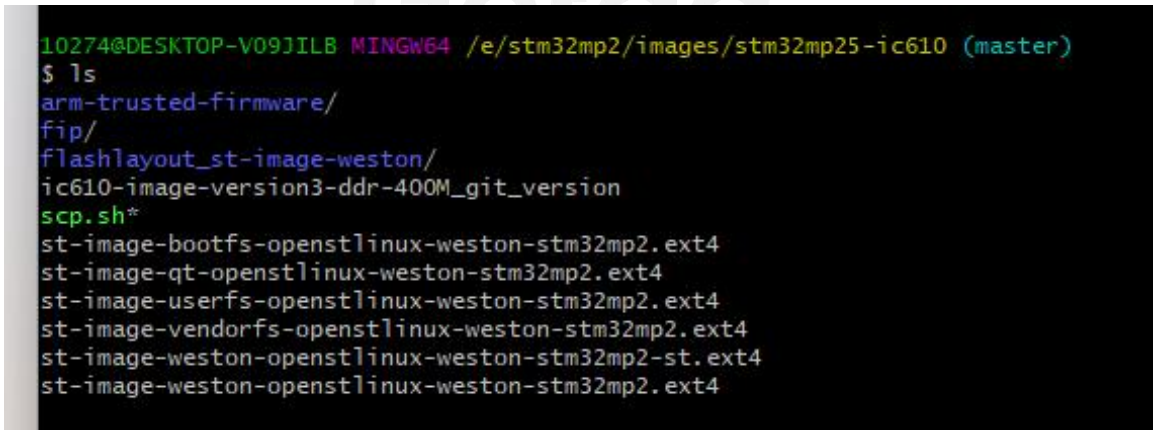
fip	u-boot 镜像目录	源码编译镜像自动更新至此
bootfs	Kernel 镜像目录	源码编译镜像自动更新至此
flashlayout_st-image-weston	系统工具，不可修改及删除	
scripts	系统工具，不可修改及删除	
make_raw.sh	制作 sd 卡烧录镜像工具	脚本内参数需要根据实际修改：sd_dev=/dev/sdb dts=stm32mp255d-ic610-mx
emmc_burn	Sd 卡烧录 emmc 镜像包	emmc_install_command.sh 根据实际问修改
st-image-bootfs-openstlinux-weston-stm32mp2.ext4	烧录 bootfs 镜像	bootfs 目录镜像更新到 ext4 内
st-image-userfs-openstlinux-weston-stm32mp2.ext4	烧录 userfs 镜像	
st-image-vendorfs-openstlinux-weston-stm32mp2.ext4	烧录 wendorfs 镜像	
st-image-weston-openstlinux-weston-stm32mp2.ext4	烧录 weston 镜像	
rootfs	Weston和userfs镜像自定义	可对 weston 镜像和 userfs 镜像修改及添加应用层等

2.8.1 烧录镜像更新到 windows

Windows 下下载 git 并安装，下载路径 https://gitforwindows.org/ 。

ic610-sdk\images\stm32mp25-ic610-rev1.rar 为镜像烧录包，解压后进入。

安装 git 后在镜像烧录文件夹内击桌面选择 Git Bash Here，创建 shell 窗口。



修改 scp.sh 内变量

```
path=/home/yqa/st/stm32mp2/sdk_v24.11.06
ubuntu_ip=192.168.2.103
user=yqa
```

path 为 sdk 源码路径，ubuntu_ip 为 ubuntu 开发环境 ip（开发环境和 windows 在同一个局域网内）

user 为 ubuntu 的用户名。

修改完成后运行./scp.sh 将镜像更新到烧录包内。

```
$ vi scp.sh

10274@DESKTOP-V09JILB MINGW64 /e/stm32mp2/images/stm32mp25-ic610 (master)
$ ./scp.sh
tf-a-stm32mp255d-ic610-mx-optee-emmc.stm32     100%  203KB   3.3MB/s   00:00
tf-a-stm32mp255d-ic610-mx-optee-sdcard.stm32   100%  203KB   3.5MB/s   00:00
tf-a-stm32mp255d-ic610-mx-usb.stm32            100%  199KB   3.7MB/s   00:00
tf-a-stm32mp257f-ic610-optee-emmc.stm32        100%  203KB   4.1MB/s   00:00
tf-a-stm32mp257f-ic610-optee-sdcard.stm32      100%  203KB   2.4MB/s   00:00
tf-a-stm32mp257f-ic610-usb.stm32               100%  199KB   3.1MB/s   00:00
metadata.bin                                   100%  120    27.9KB/s   00:00
fip-stm32mp255d-ic610-mx-ddr-optee-emmc.bin    100%   29KB 692.9KB/s   00:00
fip-stm32mp255d-ic610-mx-ddr-optee-sdcard.bin  100%   29KB 537.2KB/s   00:00
fip-stm32mp255d-ic610-mx-optee-emmc.bin          0%    0     0.0KB/s   --:-- ETA
```

镜像烧录参考 4.2 镜像烧录。

2.8.2 自定义文件系统

SDK 下 ic610-images/rootfs 内为文件系统 st-image-weston-openstlinux-weston-stm32mp2.rootfs.ext4 和 st-image-userfs-openstlinux-weston-stm32mp2.userfs.ext4 完善脚本，用户可修改 perfect.sh 添加相关内容。

开机脚本为 ic610-images/rootfs 下 etc/rc.local

文件系统若在 yocto 下重新编译完善后，可直接替换 ic610-images 下的 st-image-weston-openstlinux-weston-stm32mp2.ext4，然后删除 ic610-images/rootfs/fs 目录，

ic610-images/rootfs$ sudo ./perfect-fs.sh

则将更新后的文件系统添加用户修改后重新打包至 ic610-images 下。

2.8.3 sd 卡启动卡镜像 raw 制作

注意：此镜像用于 sd 卡启动及 sd 卡启动后烧录 emmc，raw 镜像制作过程中自动将 emmc 烧录镜像打包至镜像中。

make_raw.sh 为制作 raw 镜像脚本

```
ic610-images$ sudo ./make_raw.sh
```

raw 镜像 FlashLayout_sdcard_stm32mp255d-ic610-optee.raw ，该镜像需要自 Ubuntu 下手动烧录，或复制到 windows 下参考 4.3 raw 镜像烧写。

make_raw.sh 参数介绍：

dts=stm32mp255d-ic610-mx 为设备树名称，需要根据实际 dts 进行修改，目前支持 s

tm32mp257f-ic610 和 stm32mp255d-ic610-mx 其他 dts 用户可自行修改并增加适配的 tsv 文件。

脚本依赖的 tsv 文件如下：

flashlayout_st-image-weston/optee/FlashLayout_sdcard_stm32mp255d-ic610-mx-optee.tsv

flashlayout_st-image-weston/optee/FlashLayout_sdcard_stm32mp255d-ic610-mx-optee.tsv

烧录 raw 镜像到 sd 卡种，/dev/sdb 为 sd 卡节点，具体以实际节点为准，防止破坏虚拟机环境：

```
ic610-images$ sudo dd if=./FlashLayout_sdcard_stm32mp255d-ic610-optee.raw of=/dev/sdb bs=8M conv=fdatasync status=progress
```

Lierda 利尔达

3 系统镜像更新及系统工具安装更新

镜像烧录有 2 种方式：

1、STM32CubeProgrammer 使用 otg 烧录

2、 Sd 卡启动烧录 emmc 镜像

3.1 OTG 烧录

3.1.1 STM32CubeProgrammer 安装

图 2.1 烧录软件安装

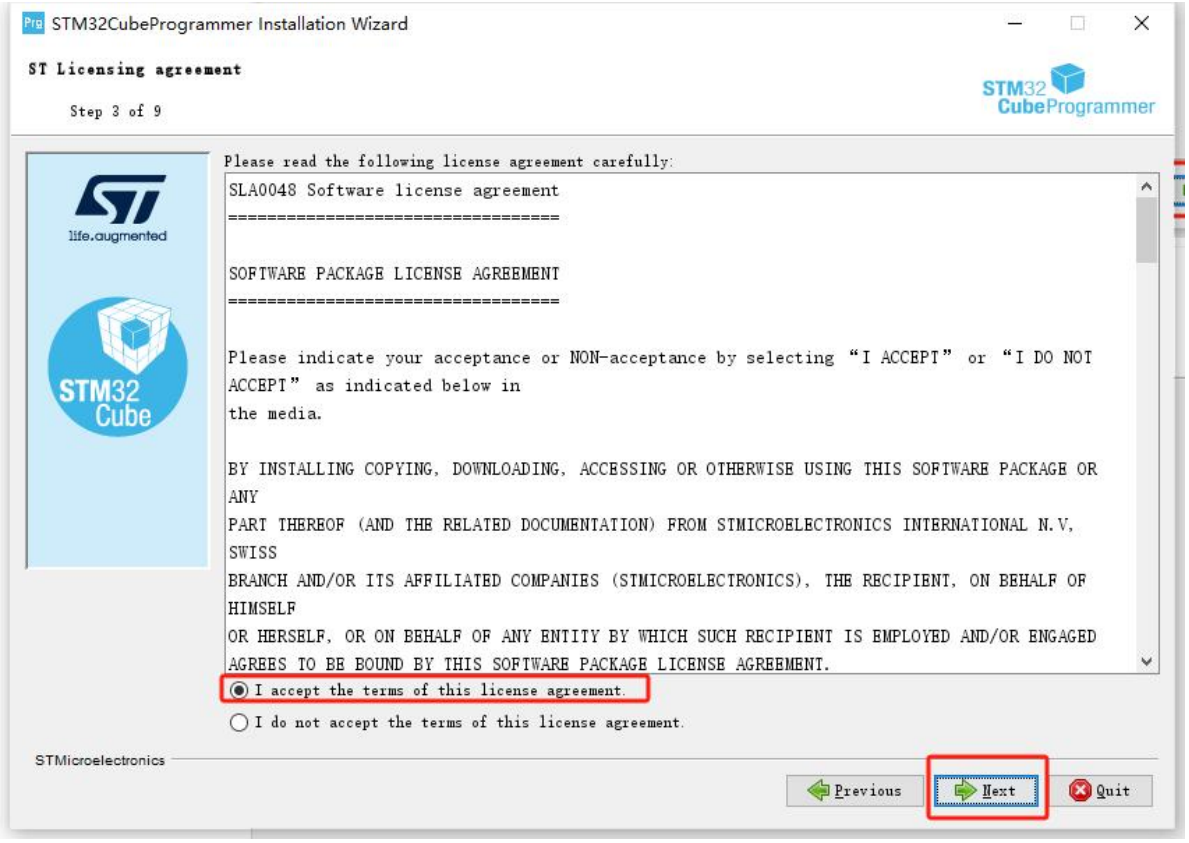


图 2.2 接受协议

图 2.3 烧录软件安装

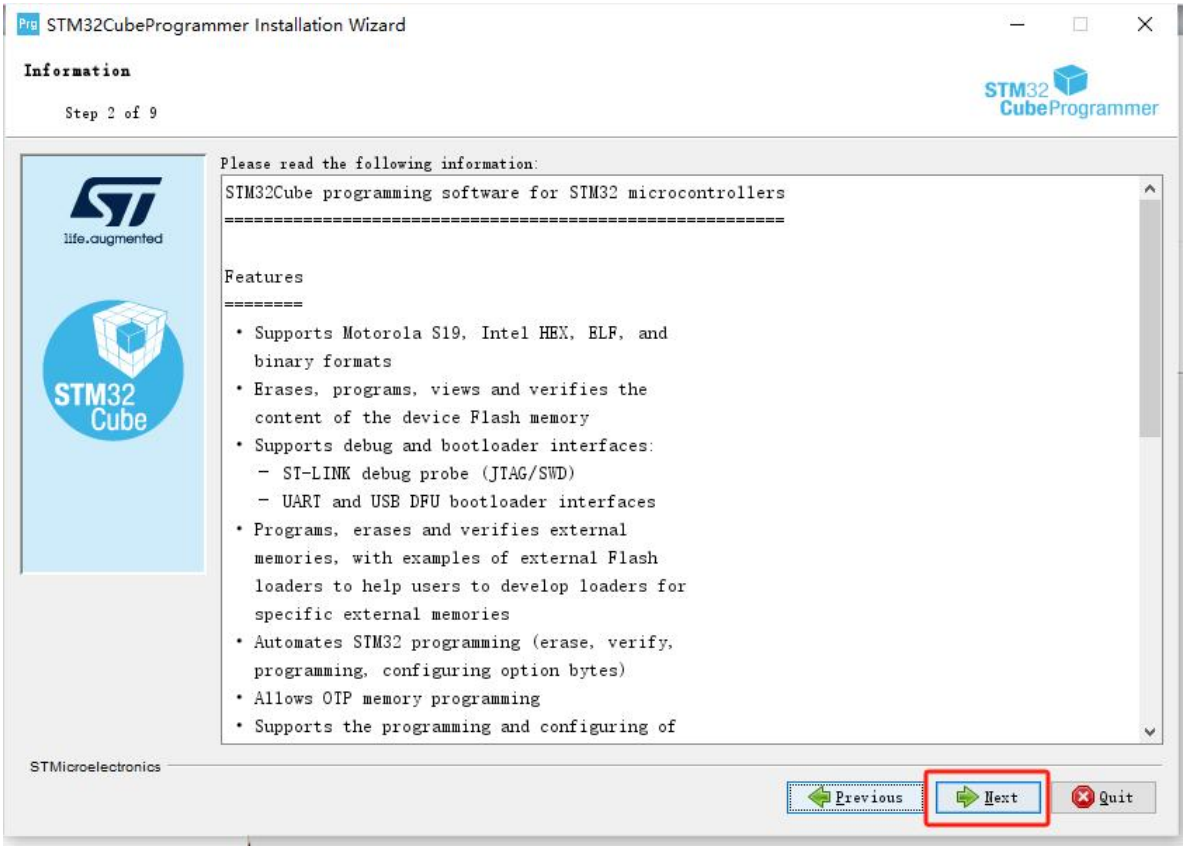


图 2.4 烧录软件安装

图 2.5 烧录软件安装

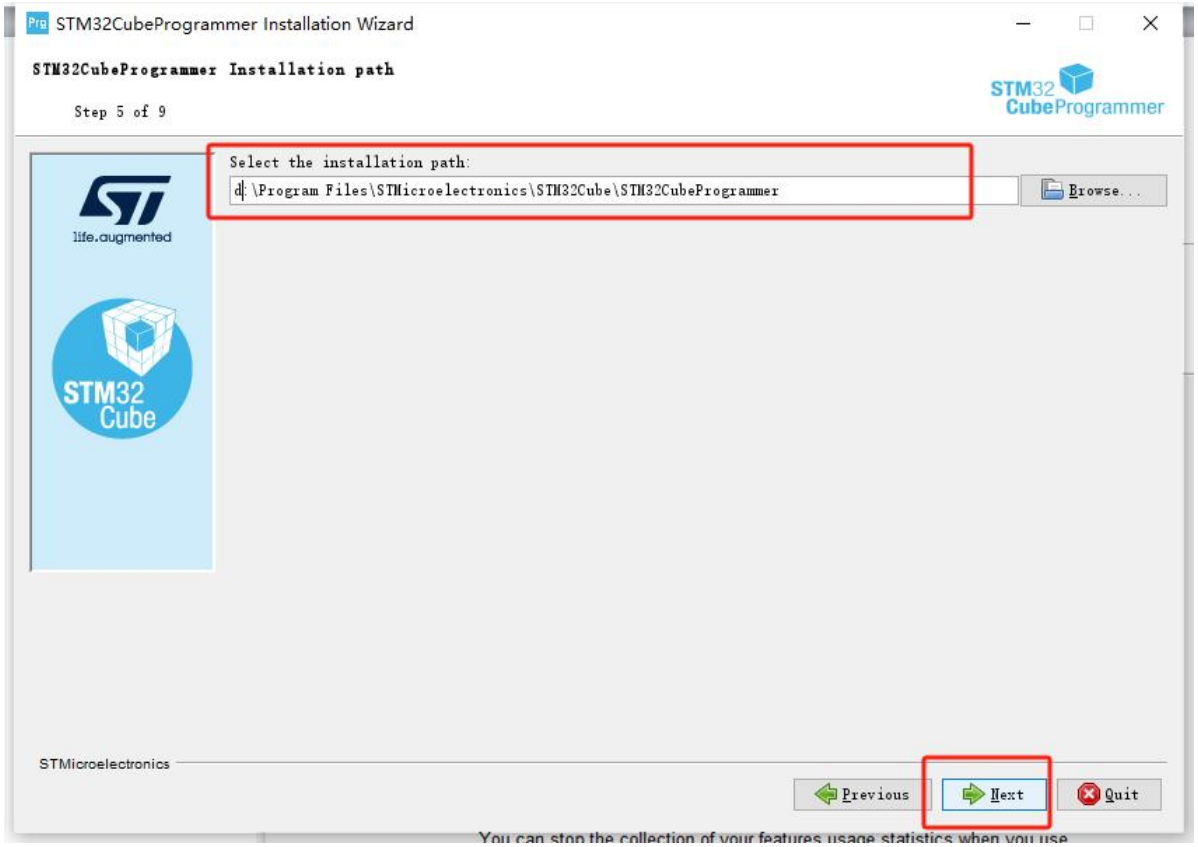


图 2.6 烧录软件安装

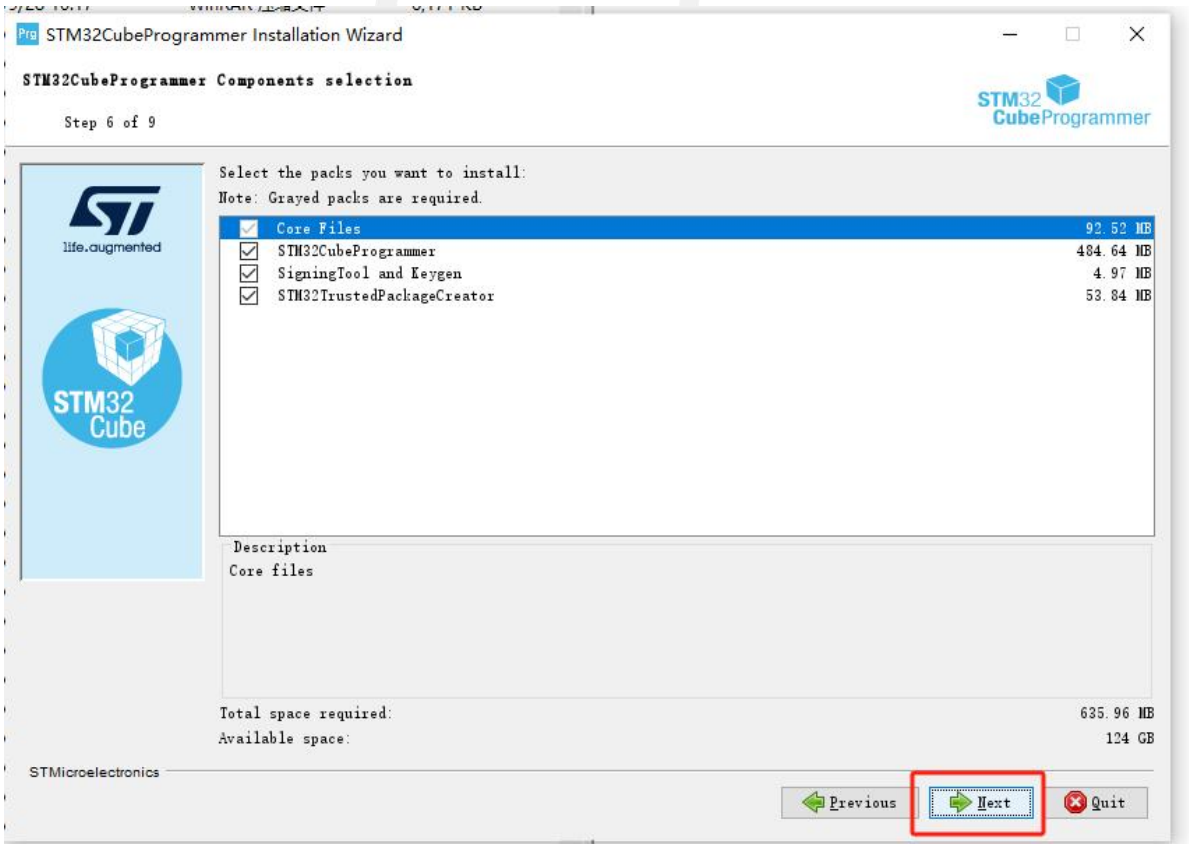


图 2.7 烧录软件安装

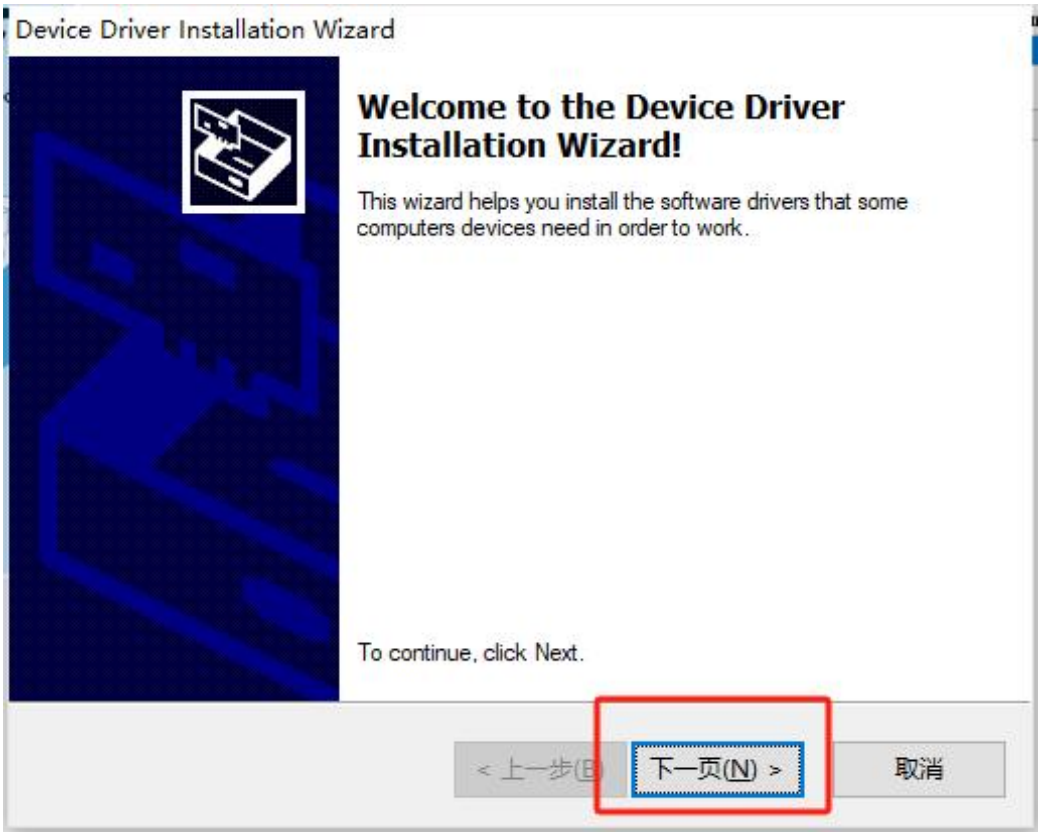


图 2.8 驱动安装

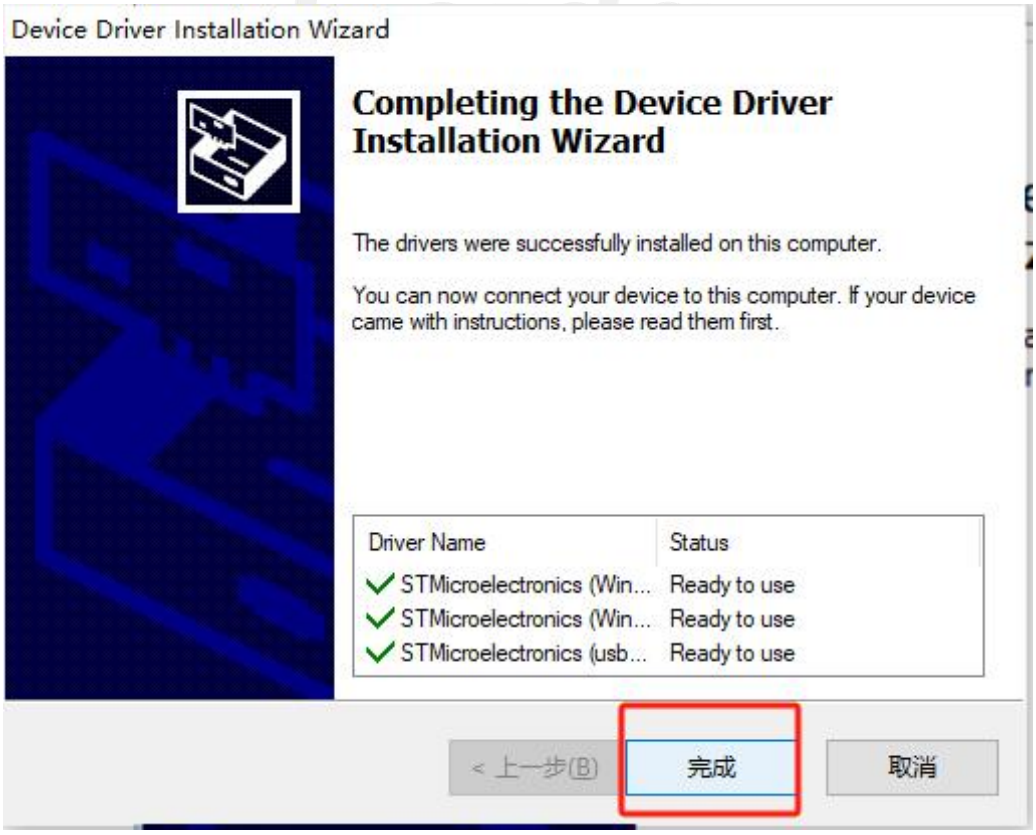


图 2.9 驱动安装完成

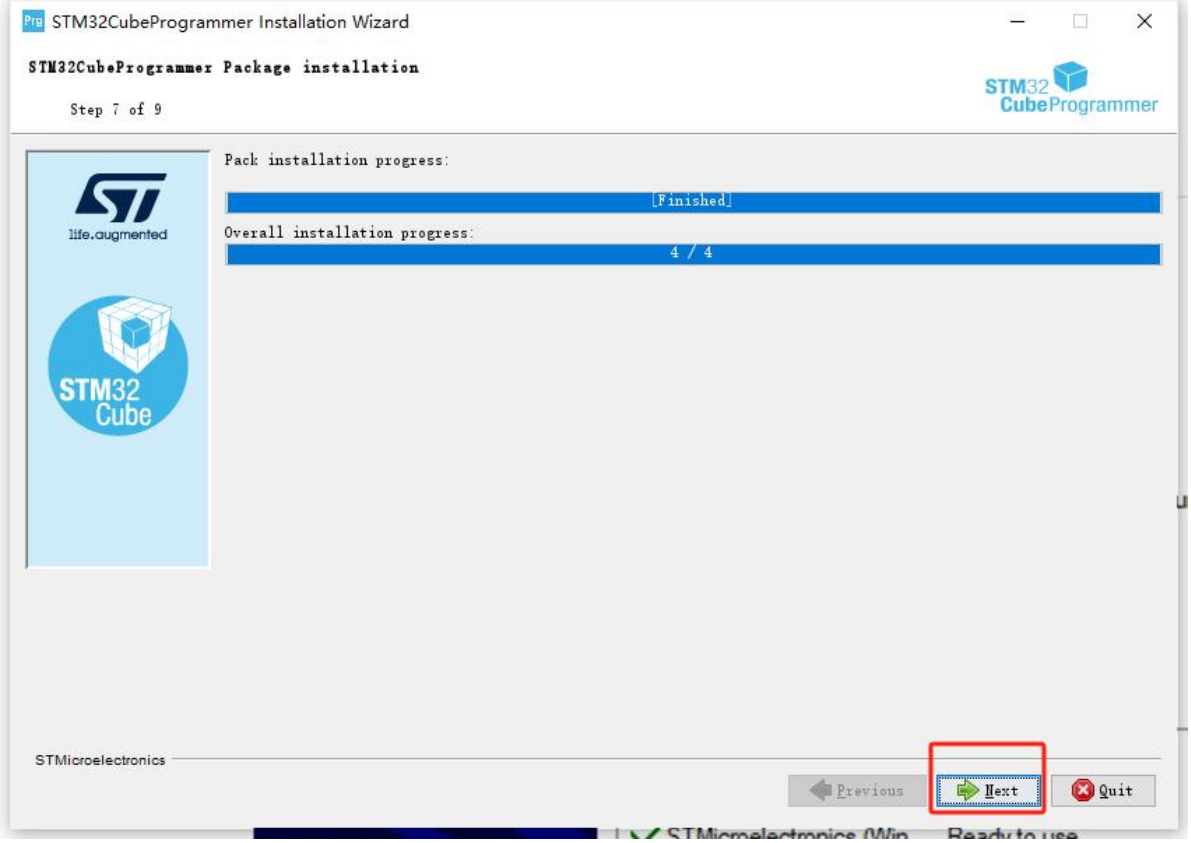


图 2.10 烧录软件安装

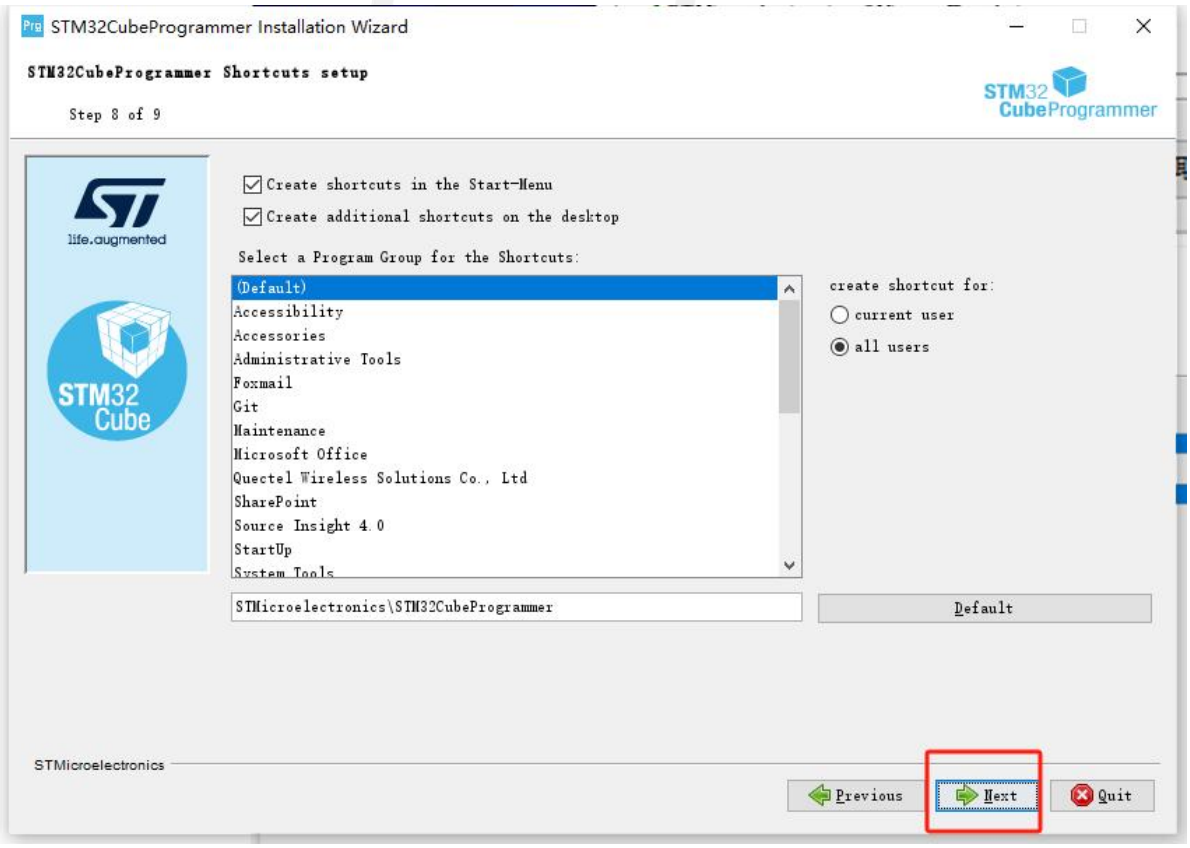


图 2.11 烧录软件安装

安装完成后桌面生成快捷启动方式

，双击运行。

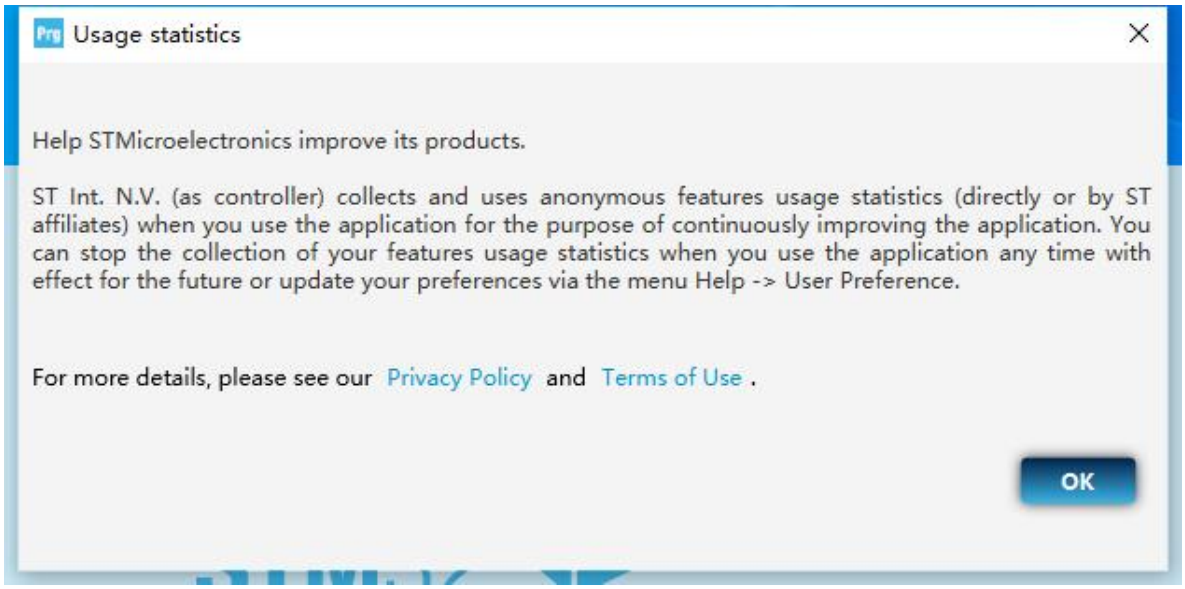


图 2.12 运行 STM32CubeProgrammer

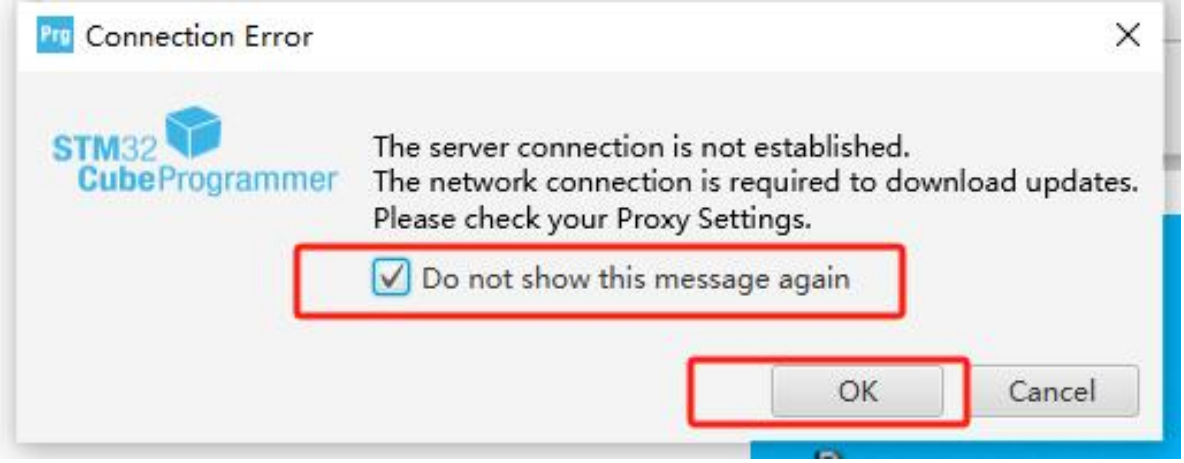


图 2.13 运行 STM32CubeProgrammer

如下 STM32CubeProgrammer 安装完成。

3.1.2 STM32CubeProgrammer 烧录镜像

参考《Lierda_IC610_EVK 快速上手应用指导》2.1Boot 将拨码设置为 OTG 烧录模式，然后对开发板重新上电或按复位按键，开发板重新上电后 OTG 座子连接 pc，选择升级方式为 usb，点击刷新按键出现 USB 设备。点击 connect 连接 ic610。如下图：

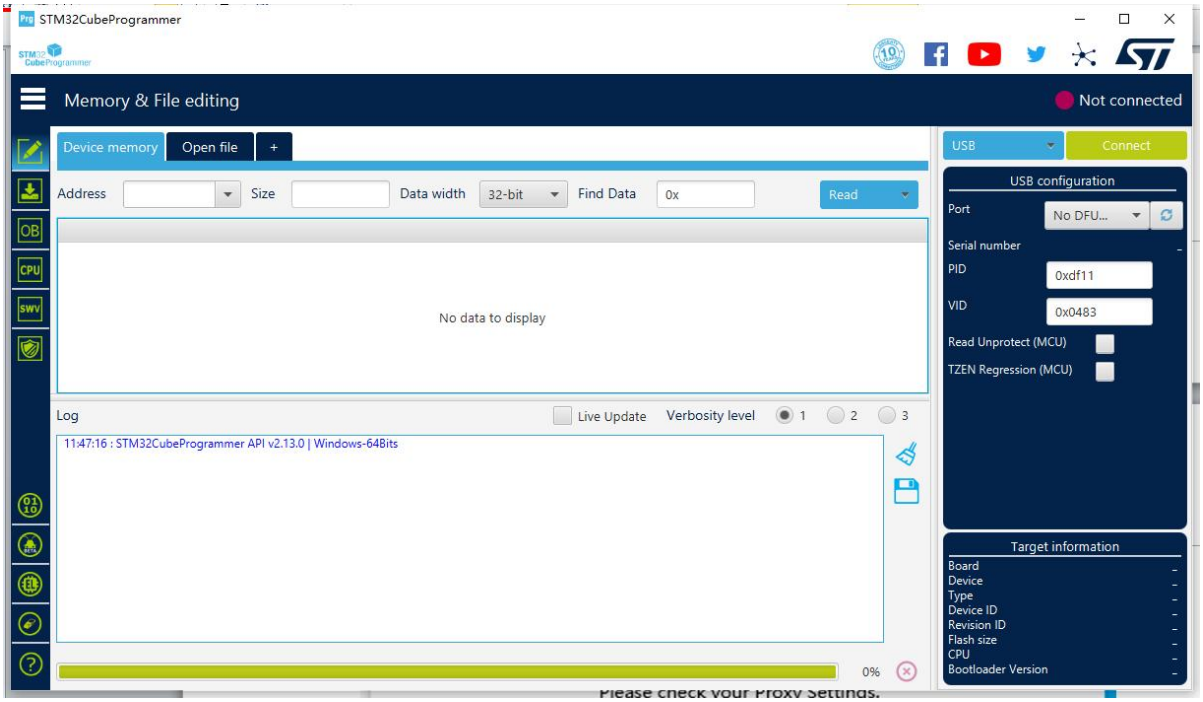
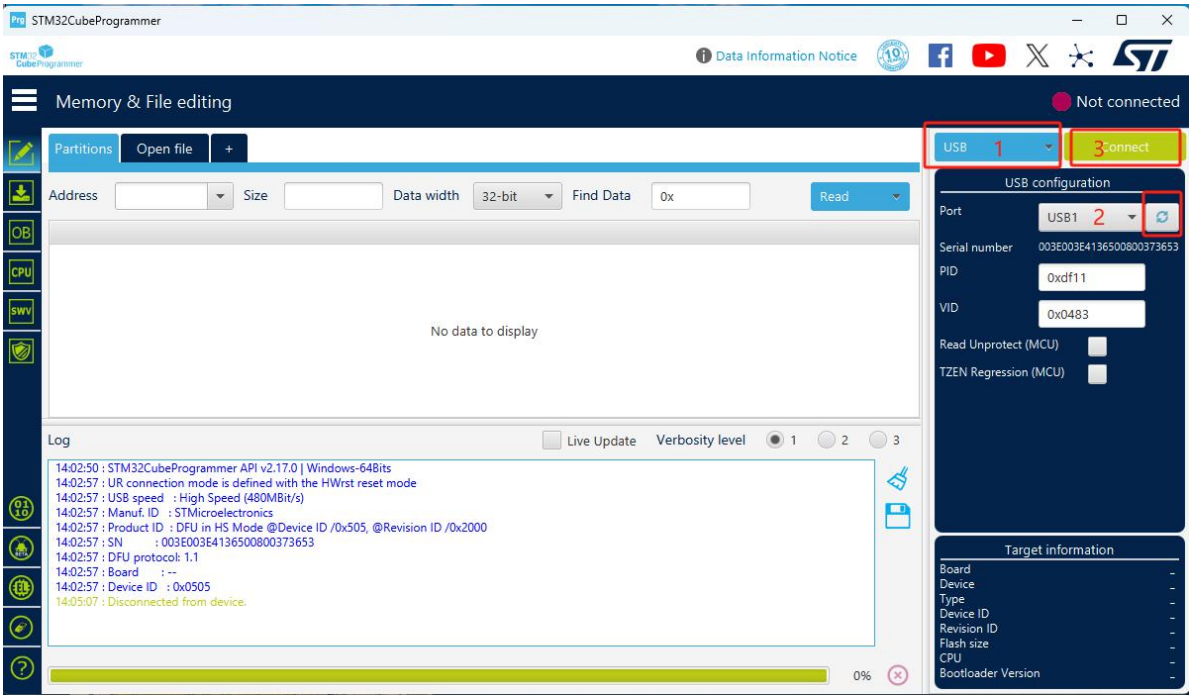


图 2.14 STM32CubeProgrammer 烧录界面

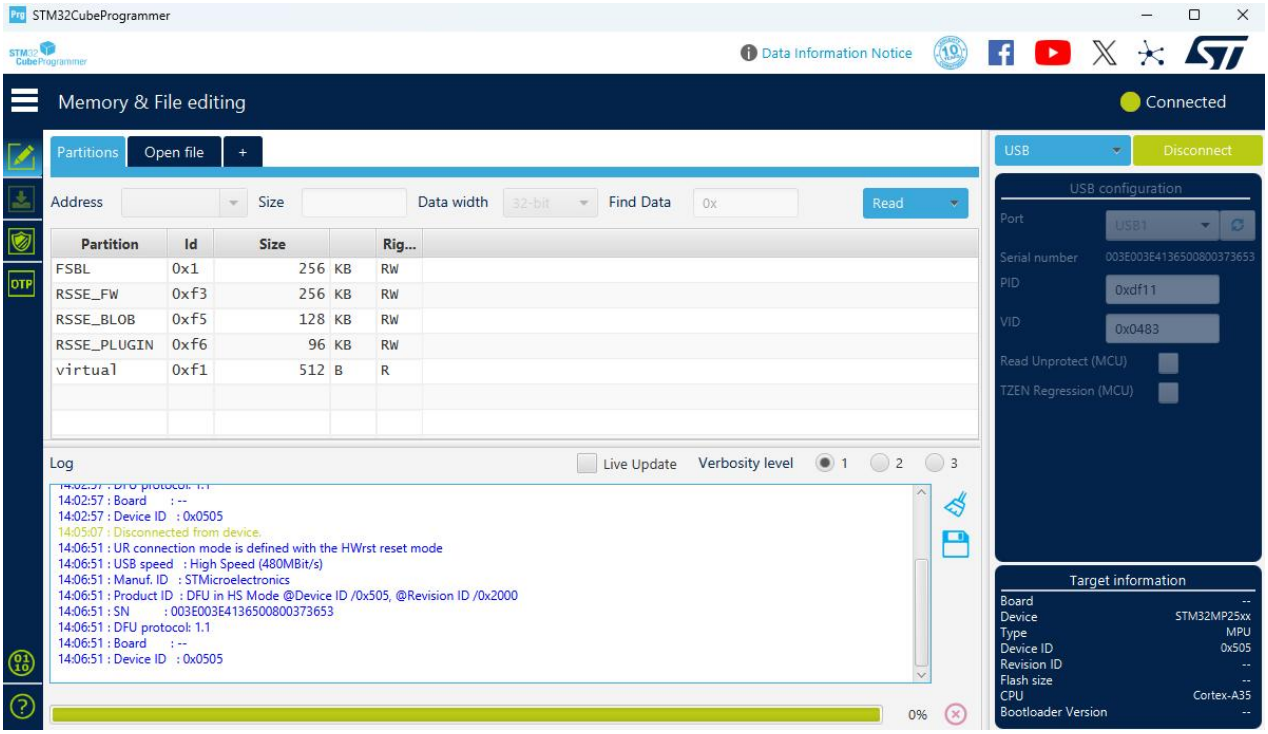


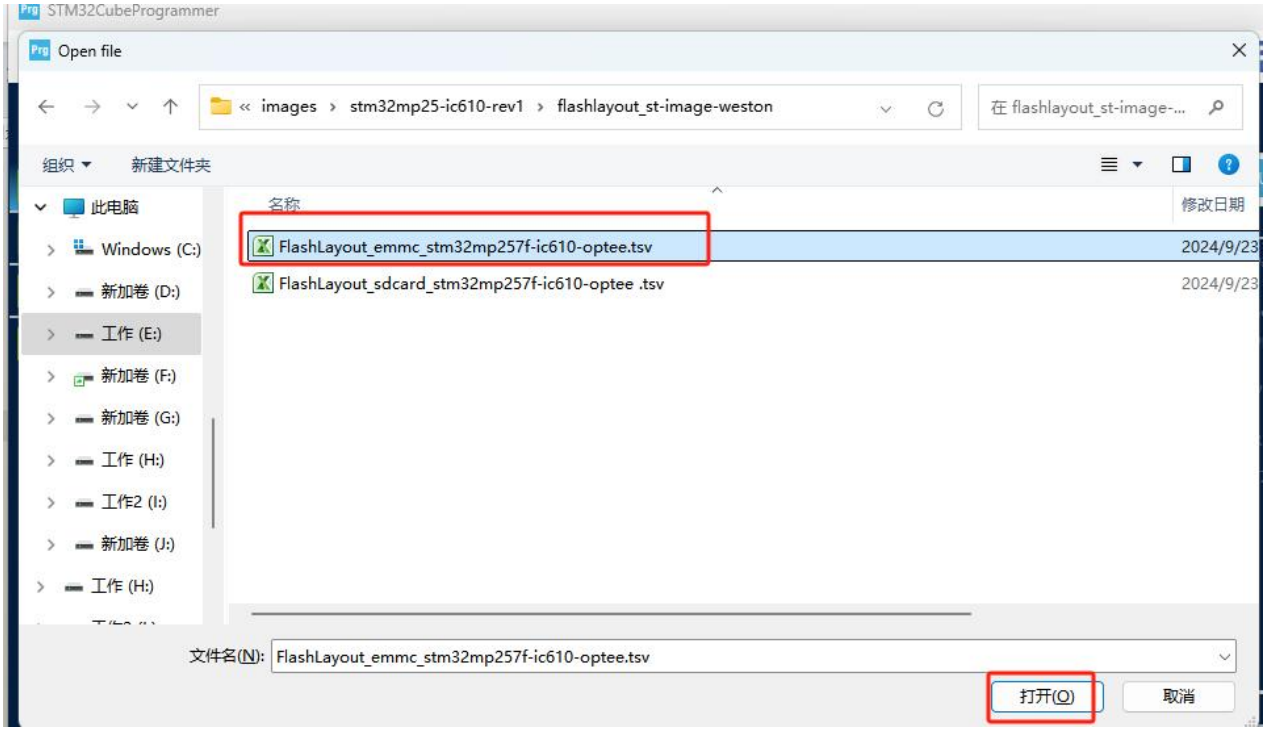
点击“Open file”选择镜像烧录包下的 stm32mp25-ic610-rev1\flashlayout_st-image-weston\FlashLayout_emmc_stm32mp255d-ic610-mx-optee.tsv

注意：本板子支持 emmc 启动和 sd 卡启动，选择相应的 tsv 即可烧录到对应存储设备中。

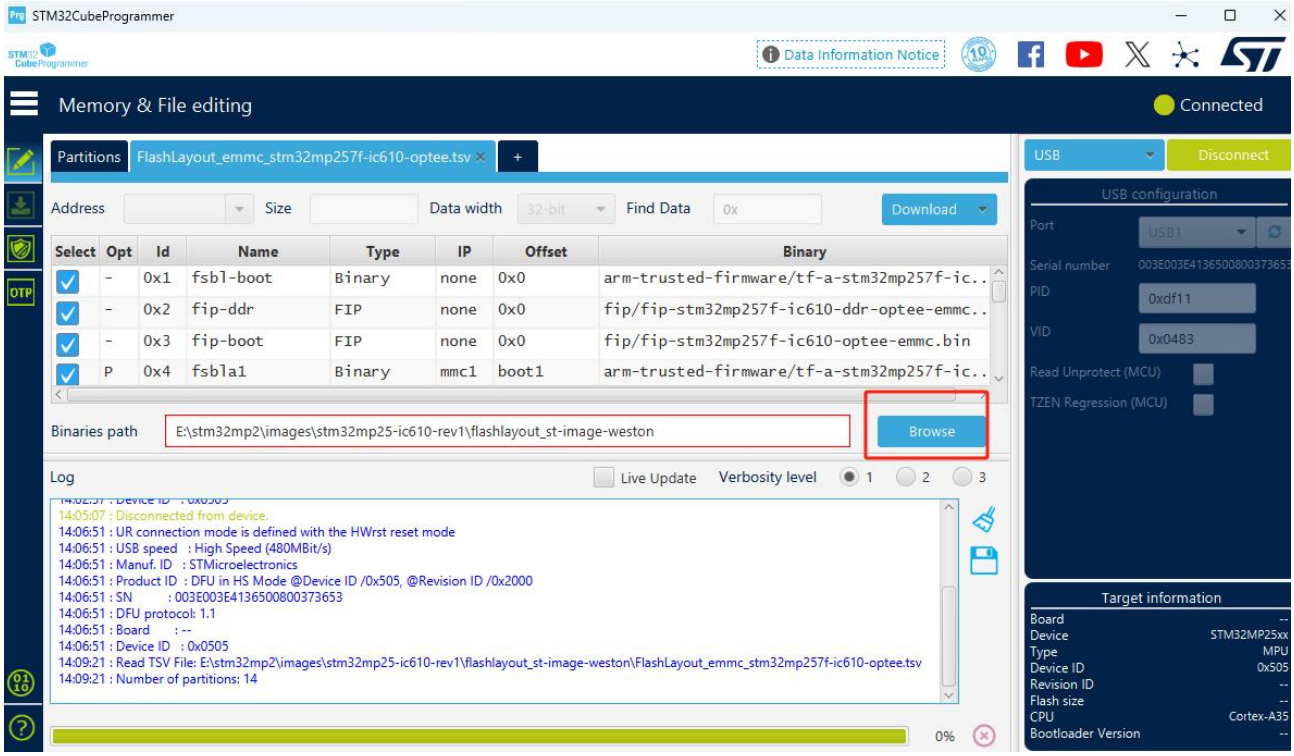
FlashLayout_emmc_stm32mp255d-ic610-mx-optee.tsv emmc 烧录配置文件。

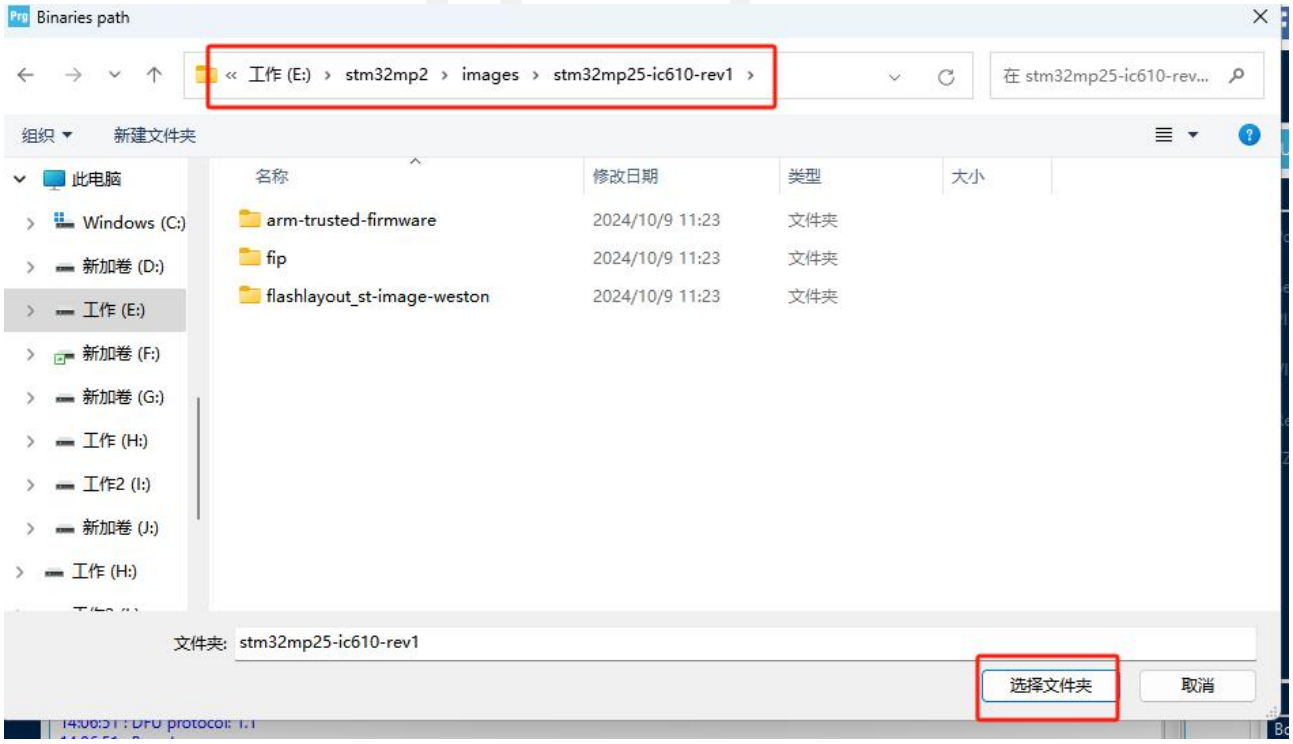
FlashLayout_sdcard_stm32mp255d-ic610-mx-optee.tsv sd 卡烧录配置文件。


STM32CubeProgrammer
Data Information Notice
Memory & File editing
Connected
Partitions
Open file
Address
Size
Data width
Find Data
Read
Partition Id Size Rig...
FSBL 0x1 256 KB RW
RSSE_FW 0xf3 256 KB RW
RSSE_BLOB 0xf5 128 KB RW
RSSE_PLUGIN 0xf6 96 KB RW
virtual 0xf1 512 B R
Log
Live Update
Verbosity level
14:02:57 : Board : --
14:02:57 : Device ID : 0x0505
14:05:07 : Disconnected from device.
14:06:51 : UR connection mode is defined with the HWrst reset mode
14:06:51 : USB speed : High Speed (480MBit/s)
14:06:51 : Manuf. ID : STMicroelectronics
14:06:51 : Product ID : DFU in HS Mode @Device ID /0x505, @Revision ID /0x2000
14:06:51 : SN : 003E003E4136500800373653
14:06:51 : DFU protocol: 1.1
14:06:51 : Board : --
14:06:51 : Device ID : 0x0505
0%
USB
Disconnect
USB configuration
Port
Serial number
PID 0xdf11
VID 0x0483
Read Unprotect (MCU)
TZEN Regression (MCU)
Target information
Board --
Device STM32MP25xx
Type MPU
Device ID 0x505
Revision ID --
Flash size --
CPU Cortex-A35
Bootloader Version --

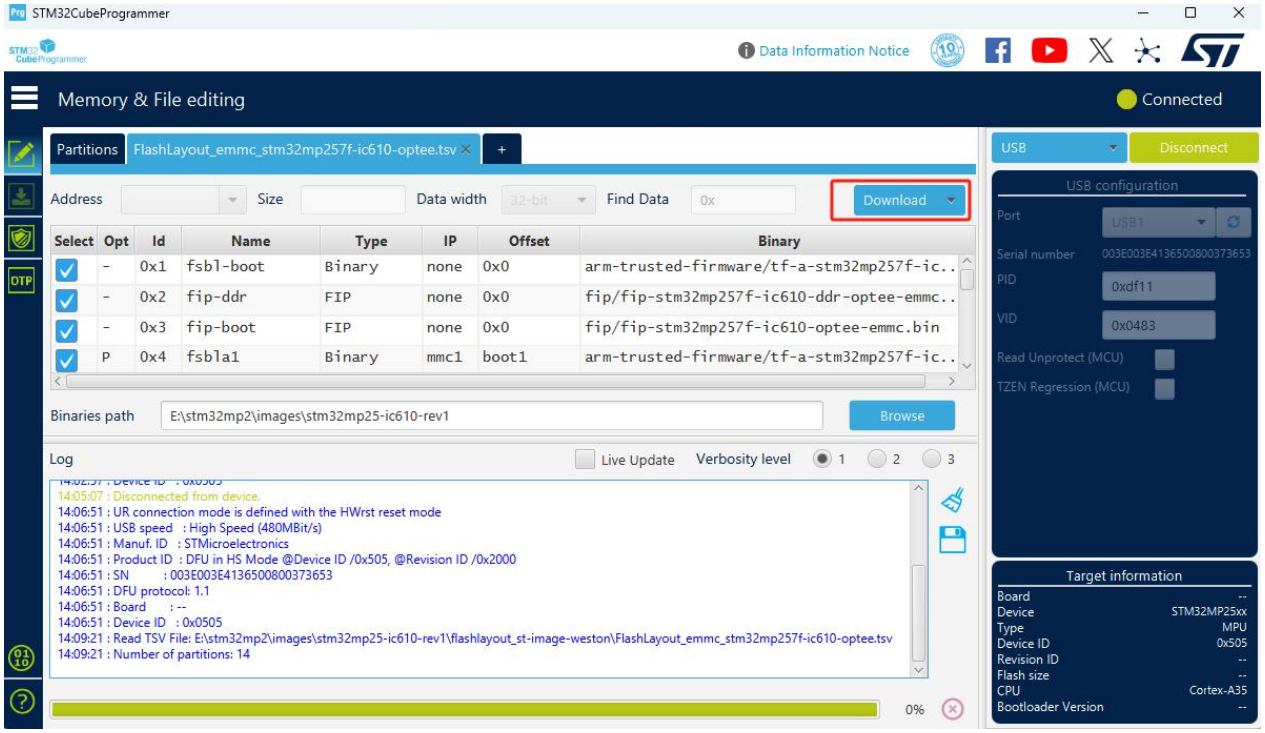

STM32CubeProgrammer
Open file
« images › stm32mp25-ic610-rev1 › flashlayout_st-image-weston
在 flashlayout_st-image-...
组织
新建文件夹
此电脑
Windows (C:)
新加卷 (D:)
工作 (E:)
新加卷 (F:)
新加卷 (G:)
工作 (H:)
工作2 (I:)
新加卷 (J:)
工作 (H:)
名称
修改日期
FlashLayout_emmc_stm32mp257f-ic610-optee.tsv 2024/9/23
FlashLayout_sdcard_stm32mp257f-ic610-optee .tsv 2024/9/23
文件名(N): FlashLayout_emmc_stm32mp257f-ic610-optee.tsv
打开(O)
取消

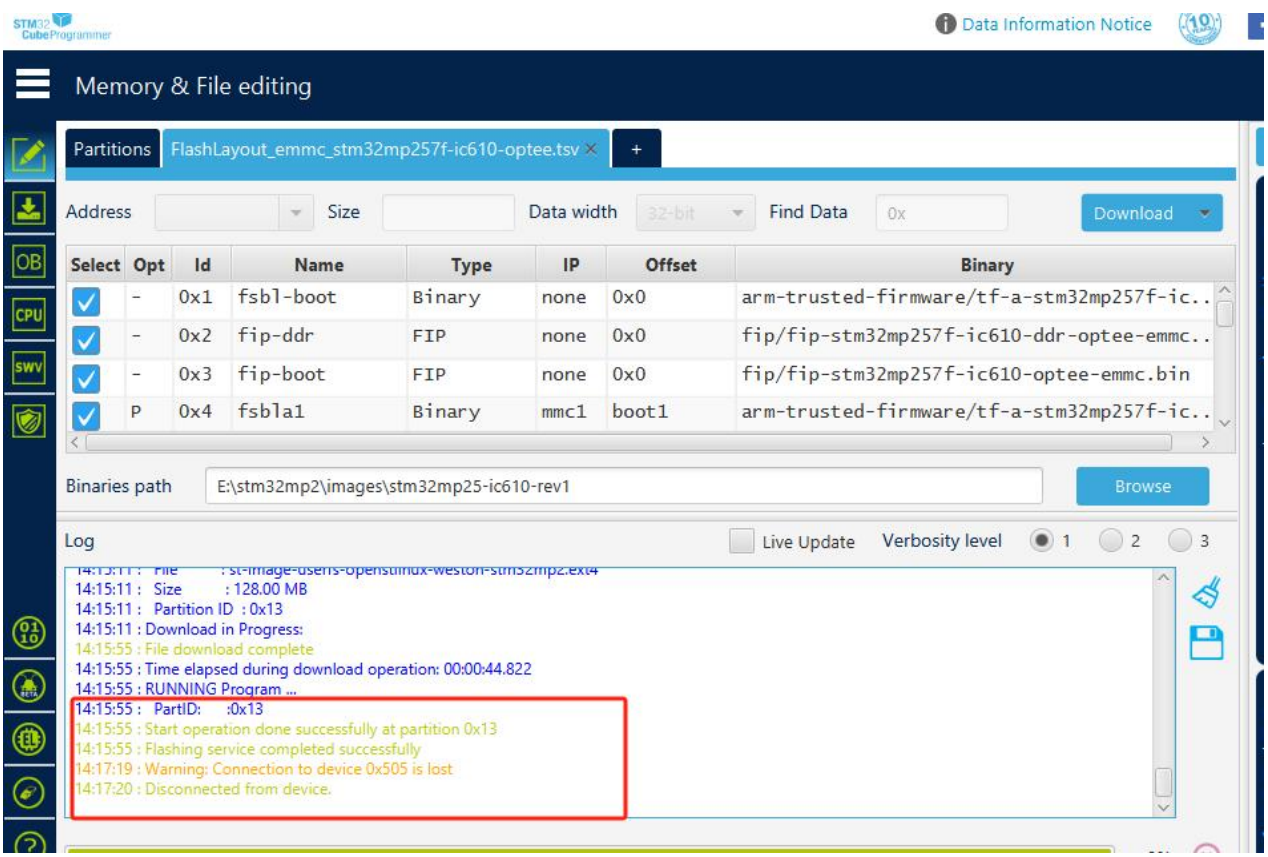
点击“Browse”选择 stm32mp25-ic610-rev1 所在路径。





如下配置完成，点击 Dowload 进行下载。





如上图表示下载完成，将拨码拨到 emmc 启动方式，重新上电或按复位按键进行启动，若烧录 sd 卡镜像，拨码拨至 sd 卡启动方式。

3.2 SD 卡启动卡烧录

3.2.1 SD 卡启动卡制作

Ubuntu 下烧录 raw 镜像：

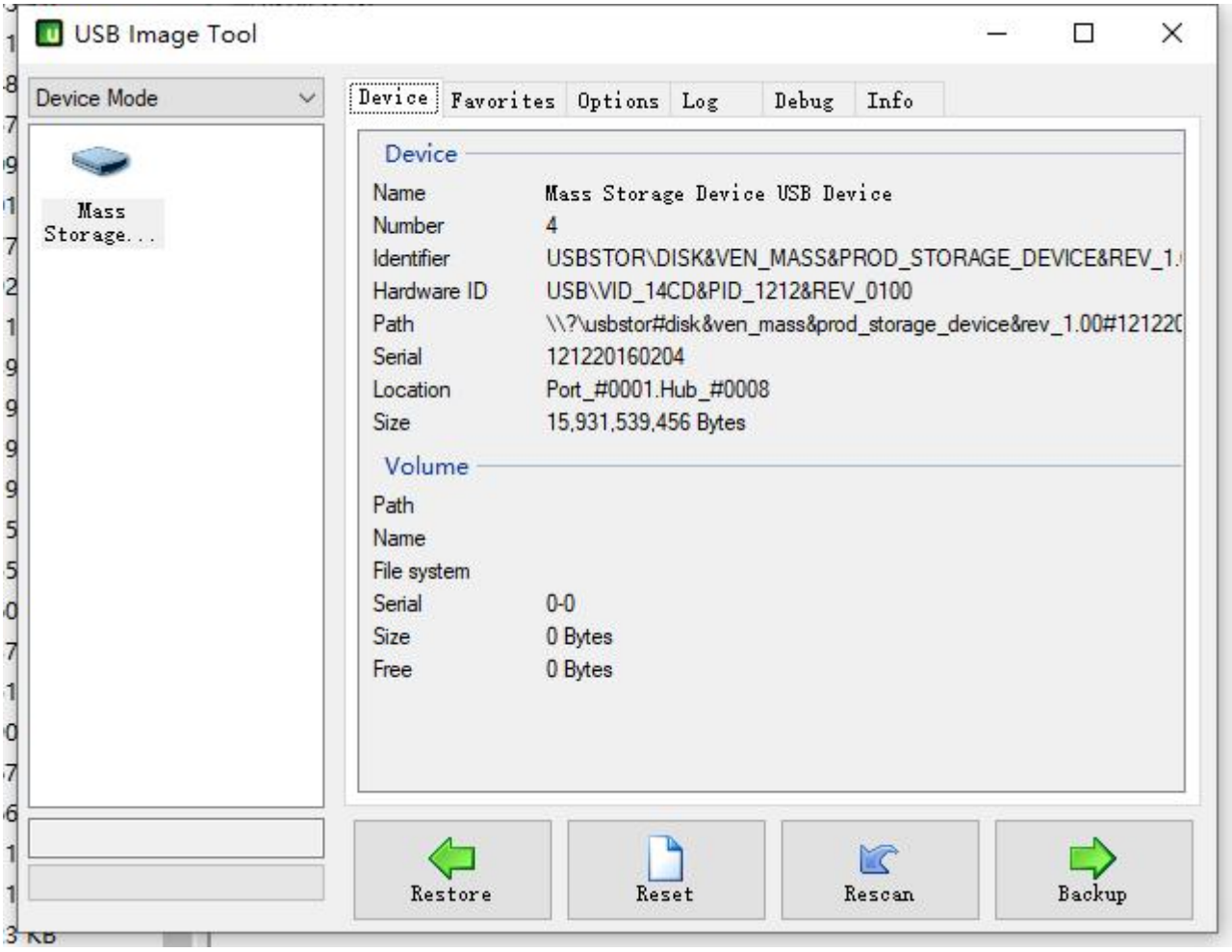
将 raw 镜像拷贝至 Ubuntu 下，将 tf 卡通过读卡器连接至 Ubuntu 下。

```
ic610-images$: sudo dd if=./FlashLayout_sdcard_stm32mp255d-ic610-mx-optee.raw of=/dev/sdb bs=8M conv=fdatasync status=progress
```

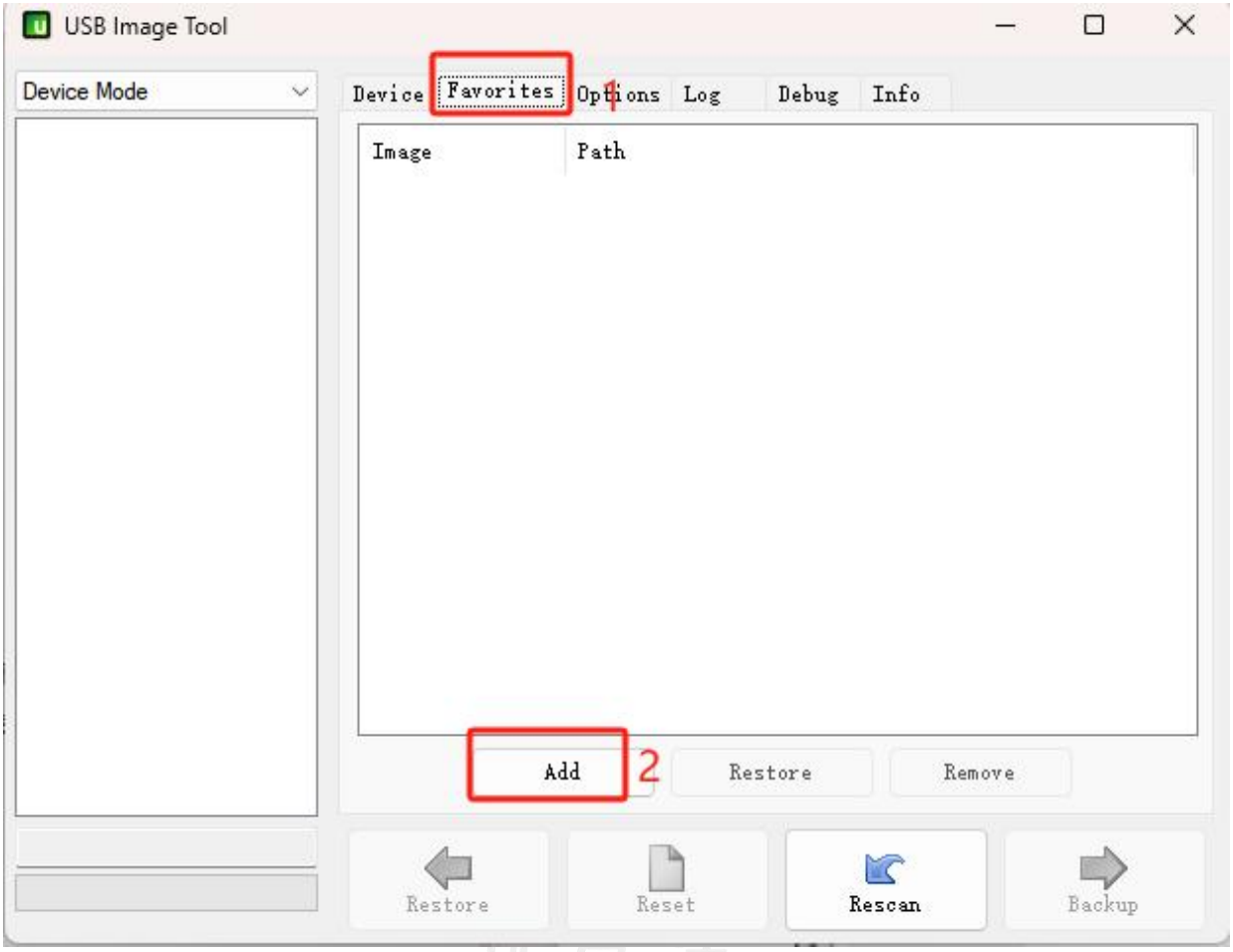
/dev/sdb 为 tf 卡节点，以实际节点为准。

Windows 下：

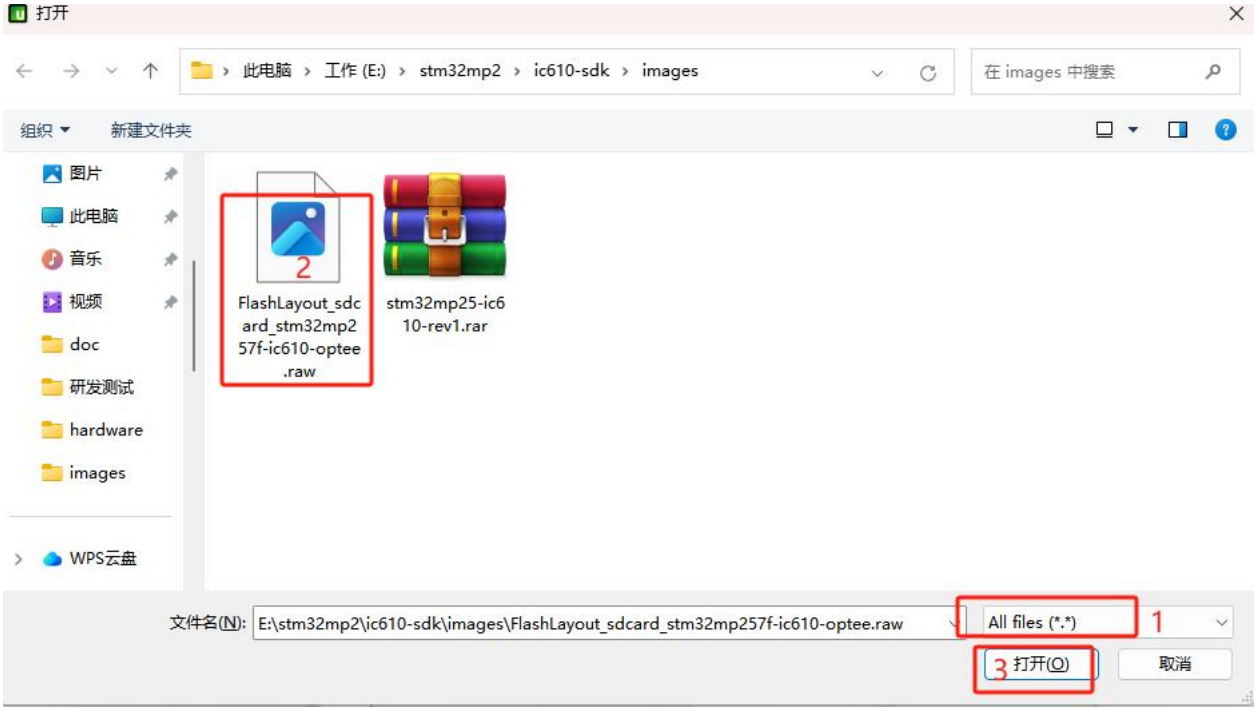
使用 USB Image Tool.exe 烧录 raw，解压 tools 下 usbit.rar， 运行 USB Image Tool.exe 读卡器插入 pc 端



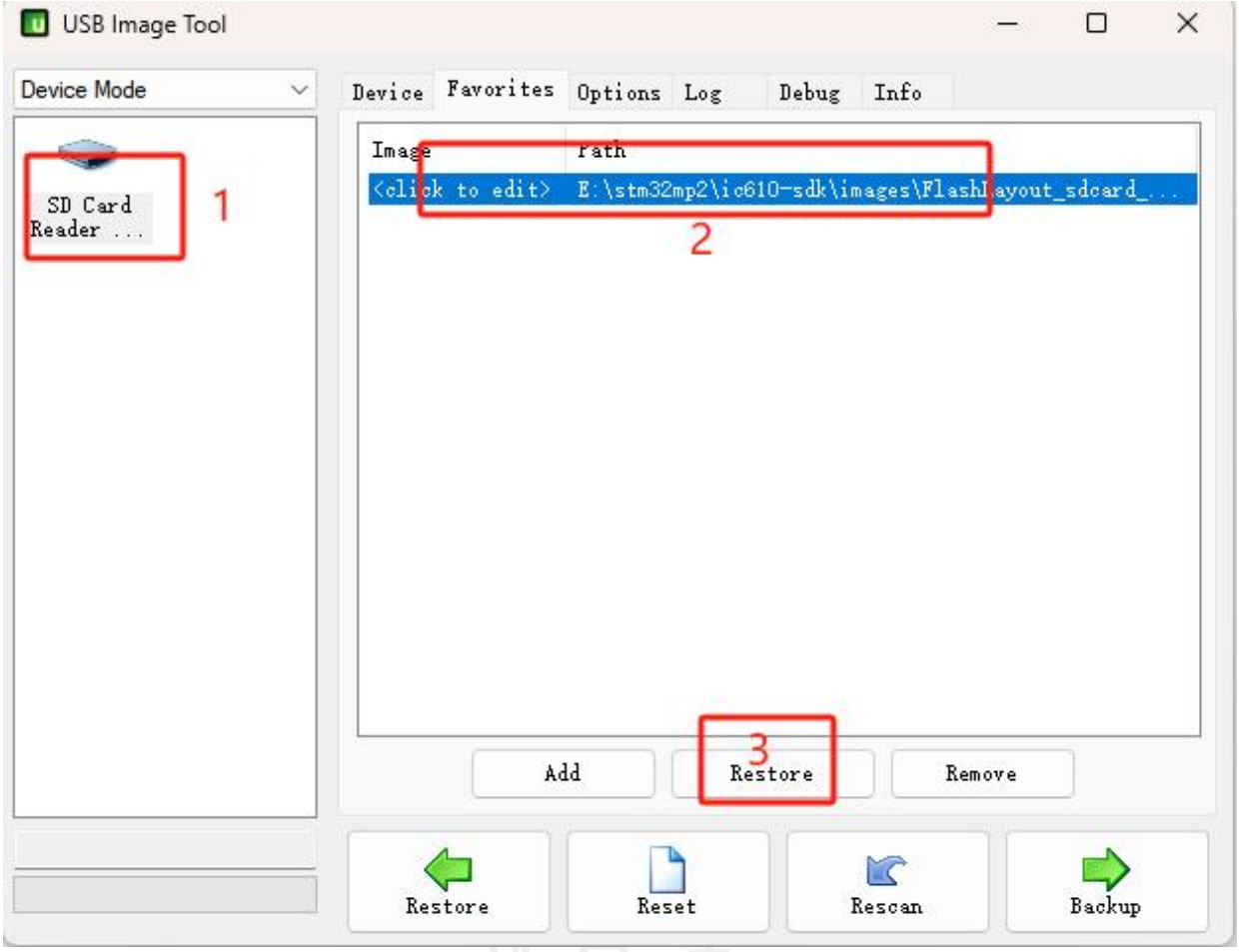
点击“Favorites”后选择“Add”



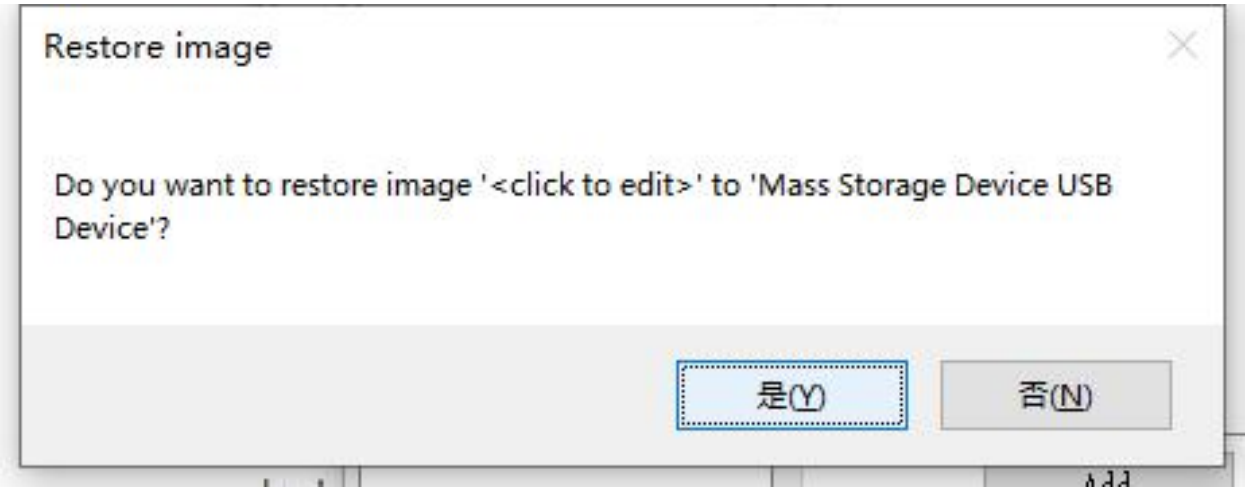
选择 All files(*.*) 选择要烧录的 raw 文件，点击打开，添加后如下：



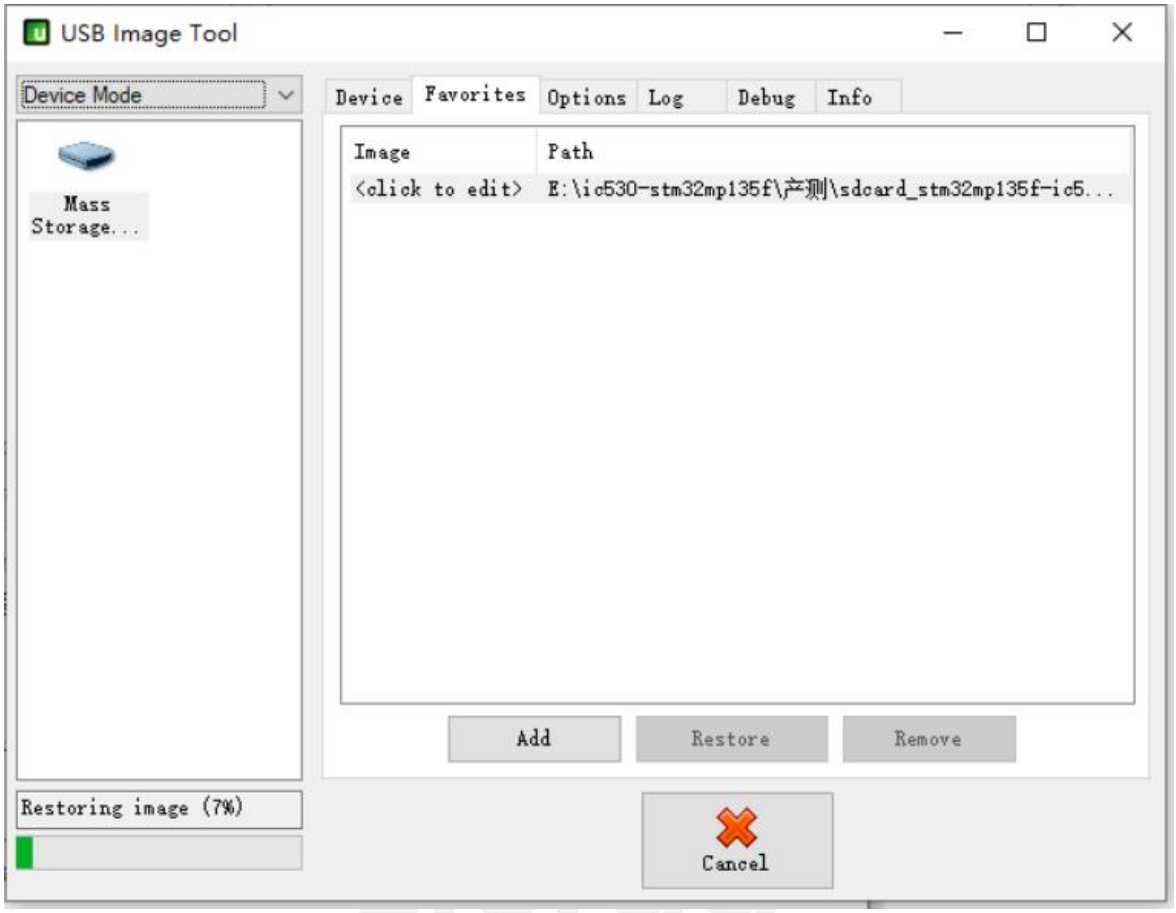
选中要烧录的镜像，点击 Restore。



点击“Restore”按钮后会弹出对话框询问是否要继续操作，我们选择“是”。



可以在界面的左下方看到当前的进度。



3.2.2 SD 启动卡烧写 emmc

拨码到 sd 卡启动，启动后烧录 emmc 镜像：

```
root@stm32mp2:~# cd /usr/local/emmc_burn/
root@stm32mp2:/usr/local/emmc_burn# ./emmc_install_command.sh
```

烧录 log 如下：

```
root@stm32mp2:/usr/local/emmc_burn# ./emmc_install_command.sh
umount: /dev/mmcblk1: not mounted.
umount: /dev/mmcblk1boot0: not mounted.
umount: /dev/mmcblk1boot1: not mounted.
umount: /dev/mmcblk1p1: not mounted.
umount: /dev/mmcblk1p2: not mounted.
umount: /dev/mmcblk1p3: not mounted.
umount: /dev/mmcblk1p4: not mounted.
umount: /dev/mmcblk1p5: not mounted.
[   81.836667] EXT4-fs (mmcblk1p6): unmounting filesystem.
[   81.842190] EXT4-fs (mmcblk1p7): unmounting filesystem.
[   81.886855] EXT4-fs (mmcblk1p8): unmounting filesystem.
[   81.896882] EXT4-fs (mmcblk1p9): unmounting filesystem.
umount: /dev/mmcblk1rpmb: not mounted.
GPT fdisk (gdisk) version 1.0.9.1

[   81.955399]  mmcblk1: p1 p2 p3 p4 p5 p6 p7 p8 p9
Partition table scan:
  MBR: protective
  BSD: not present
  APM: not present
  GPT: present

Found valid GPT with protective MBR; using GPT.
Disk /dev/mmcblk1: 17317888 sectors, 8.3 GiB
Sector size (logical/physical): 512/512 bytes
Disk identifier (GUID): FB231B46-E90B-4E6E-A0F8-FC5DDCB2EB9D
Partition table holds up to 128 entries
Main partition table begins at sector 2 and ends at sector 33
First usable sector is 34, last usable sector is 17317854
Partitions will be aligned on 1024-sector boundaries
Total free space is 1981 sectors (990.5 KiB)

Number  Start (sector)    End (sector)  Size       Code  Name
   1            1024            2047   512.0 KiB   FFFF  metadata1
   2            2048            3071   512.0 KiB   FFFF  metadata2
   3            3072           11263   4.0 MiB     FFFF  fip-a
   4           11264           19455   4.0 MiB     FFFF  fip-b
   5           19456           20479   512.0 KiB   FFFF  u-boot-env
   6           20480          151551   64.0 MiB    8300  bootfs
   7          151552          526335   183.0 MiB   8300  vendorfs
   8          526336         6817791   3.0 GiB     8300  rootfs
   9         6817792        17316863   5.0 GiB     8300  userfs
/dev/mmcblk1: 8 bytes were erased at offset 0x00000200 (gpt): 45 46 49 20 50 41 52 54
/dev/mmcblk1: 8 bytes were erased at offset 0x2107ffe00 (gpt): 45 46 49 20 50 41 52 54
/dev/mmcblk1: 2 bytes were erased at offset 0x000001fe (PMBR): 55 aa
/dev/mmcblk1: calling ioctl to re-read partition table: Success
Checking that no-one is using this disk right now ... OK

Disk /dev/mmcblk1: 8.26 GiB, 8866758656 bytes, 17317888 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

>>> Script header accepted.
>>> Script header accepted.
```

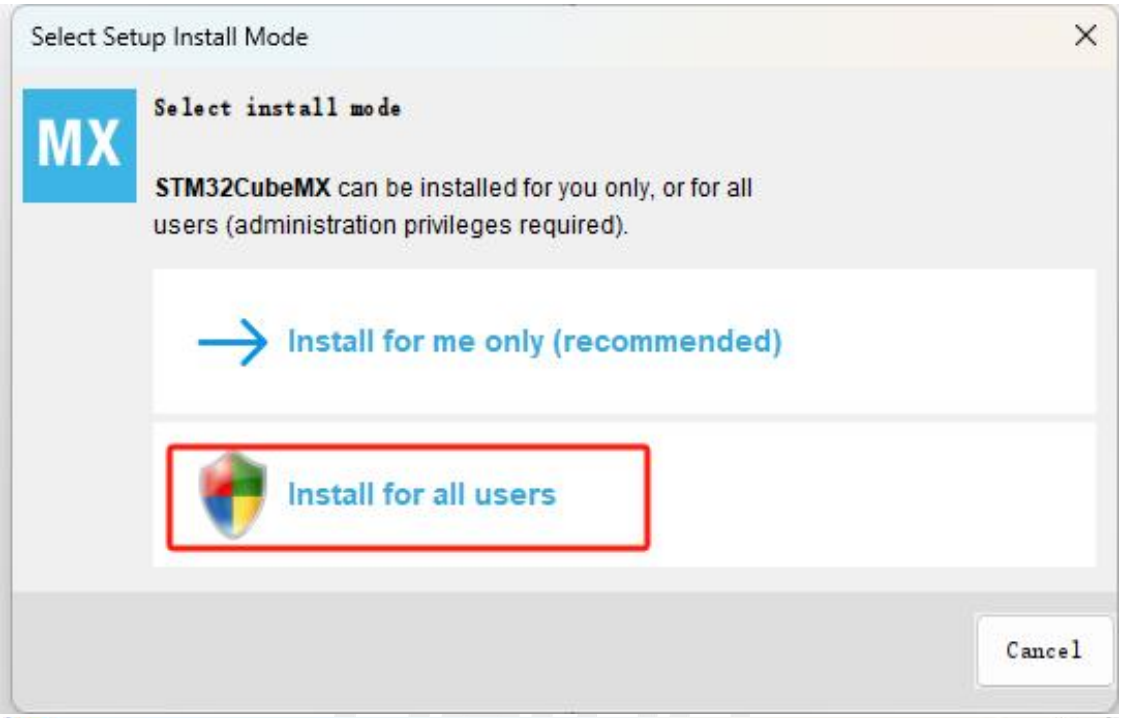
如下，镜像烧录完成，心跳灯由闪烁变为常亮，boot 拨码到 emmc 后对 evk 重新上电即可。

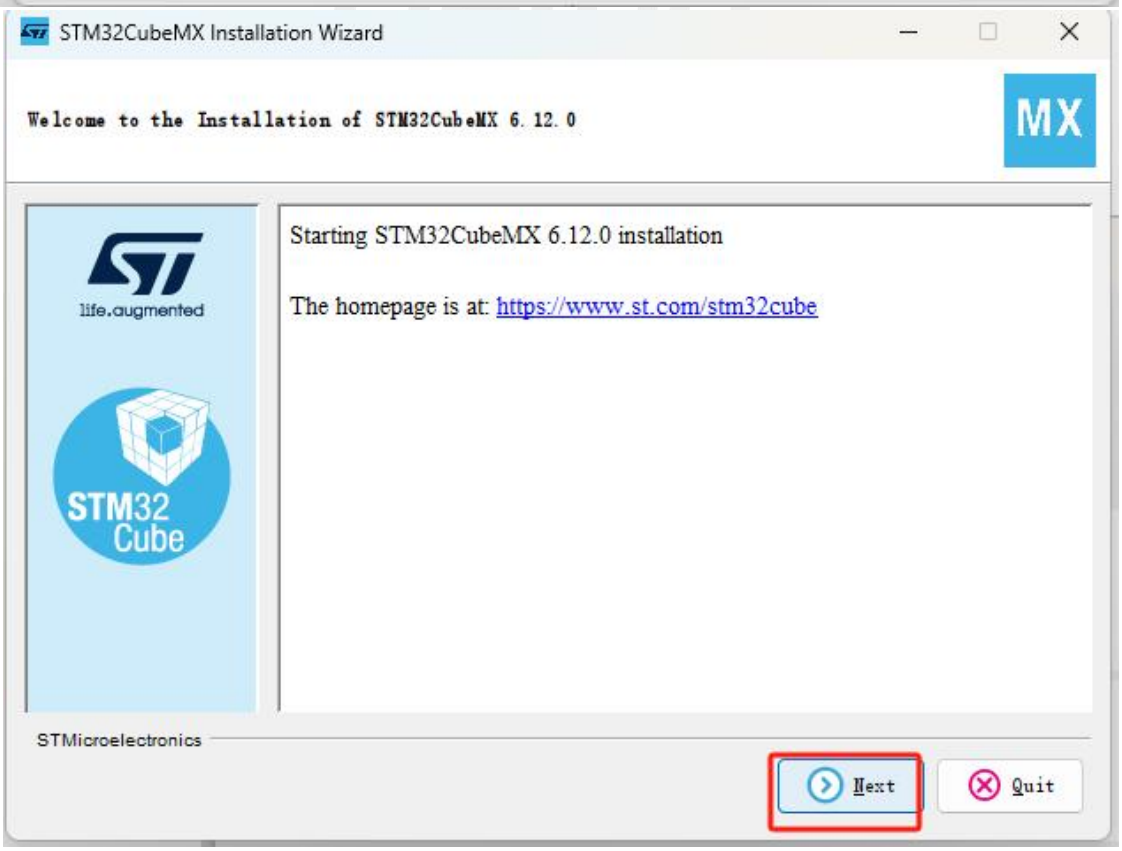
```
3113407 bytes (3.1 MB, 3.0 MiB) copied, 0.185797 s, 16.8 MB/s
2+1 records in
2+1 records out
3113407 bytes (3.1 MB, 3.0 MiB) copied, 0.131814 s, 23.6 MB/s
64+0 records in
64+0 records out
67108864 bytes (67 MB, 64 MiB) copied, 4.37545 s, 15.3 MB/s
48+0 records in
48+0 records out
50331648 bytes (50 MB, 48 MiB) copied, 2.11897 s, 23.8 MB/s
834+0 records in
834+0 records out
874512384 bytes (875 MB, 834 MiB) copied, 36.7013 s, 23.8 MB/s
128+0 records in
128+0 records out
134217728 bytes (134 MB, 128 MiB) copied, 3.3978 s, 39.5 MB/s
ic610 emmc 烧录完成，请移除sd卡修改启动方式到emmc并重新上电
ic610 emmc 烧录完成，请移除sd卡修改启动方式到emmc并重新上电
ic610 emmc 烧录完成，请移除sd卡修改启动方式到emmc并重新上电
ic610 emmc 烧录完成，请移除sd卡修改启动方式到emmc并重新上电
ic610 emmc 烧录完成，请移除sd卡修改启动方式到emmc并重新上电
ic610 emmc 烧录完成，请移除sd卡修改启动方式到emmc并重新上电
ic610 emmc 烧录完成，请移除sd卡修改启动方式到emmc并重新上电
ic610 emmc 烧录完成，请移除sd卡修改启动方式到emmc并重新上电
ic610 emmc 烧录完成，请移除sd卡修改启动方式到emmc并重新上电
ic610 emmc 烧录完成，请移除sd卡修改启动方式到emmc并重新上电
```

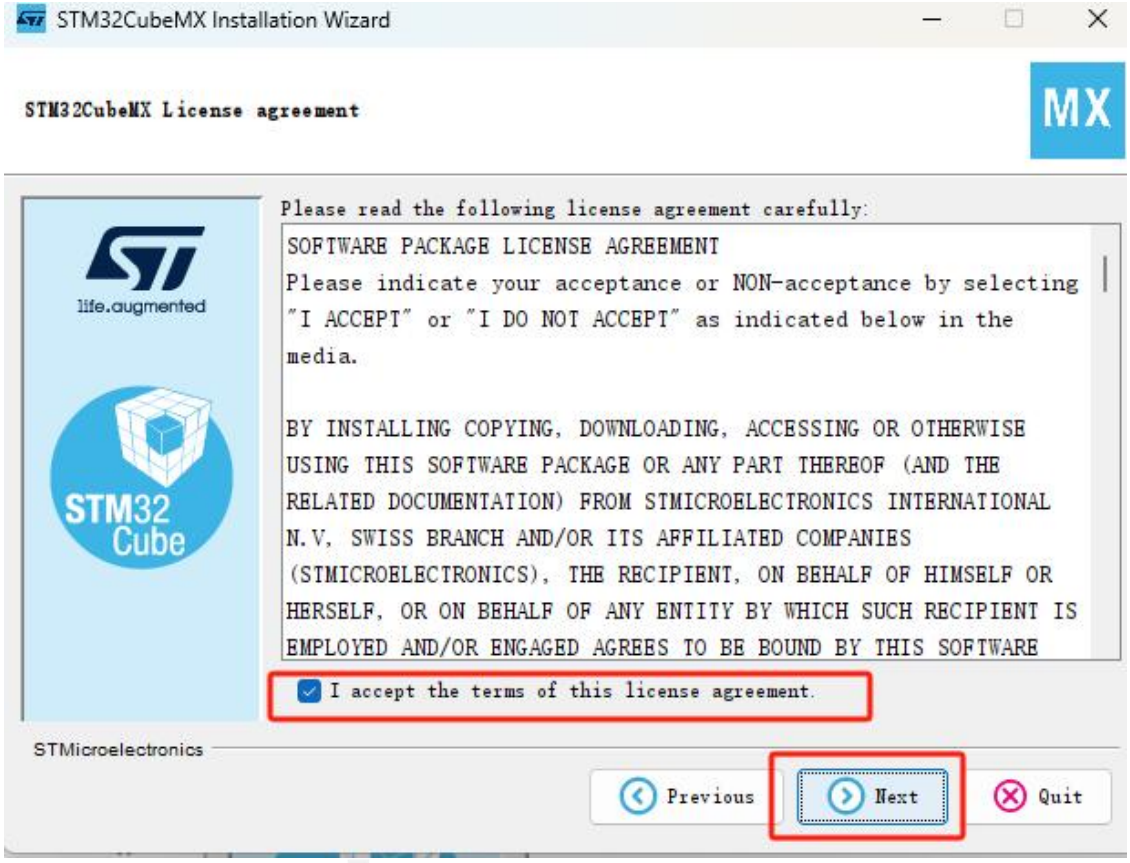
4 cubemx 安装

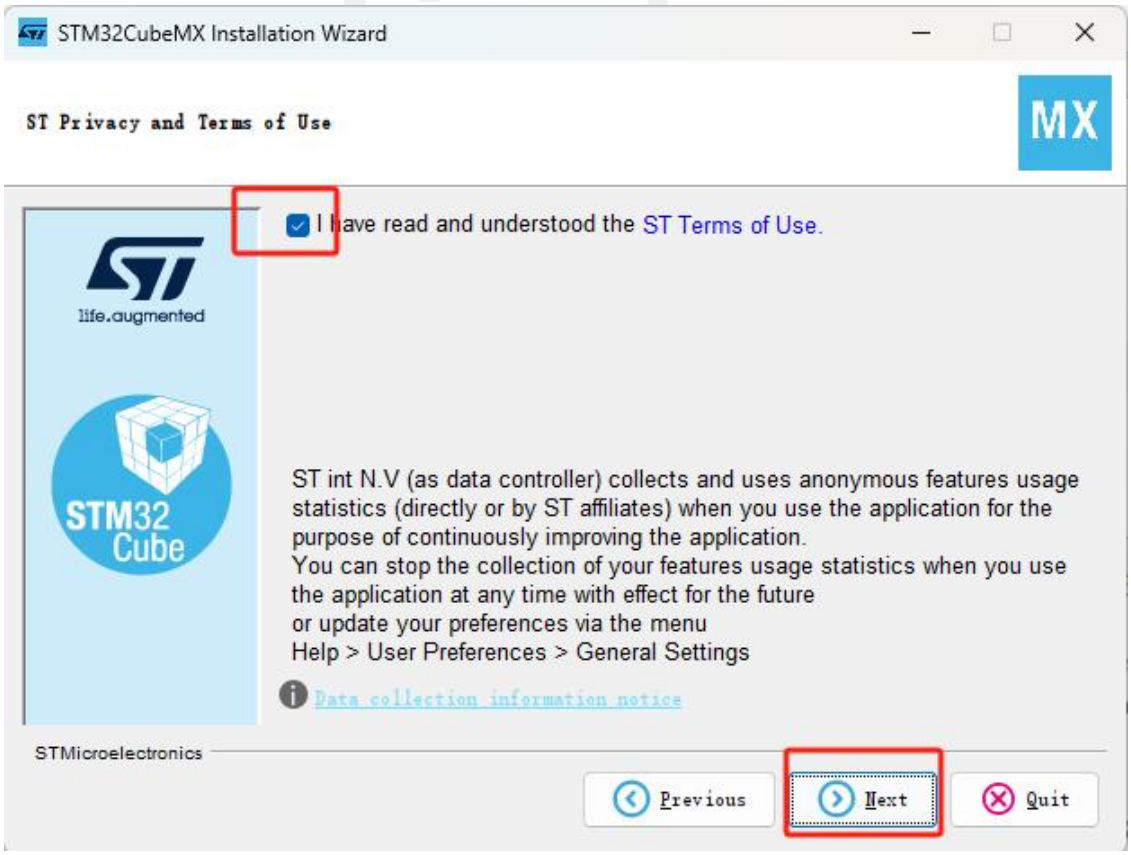
安装 tools 下 SetupSTM32CubeMX-6.14.0-Win.exe

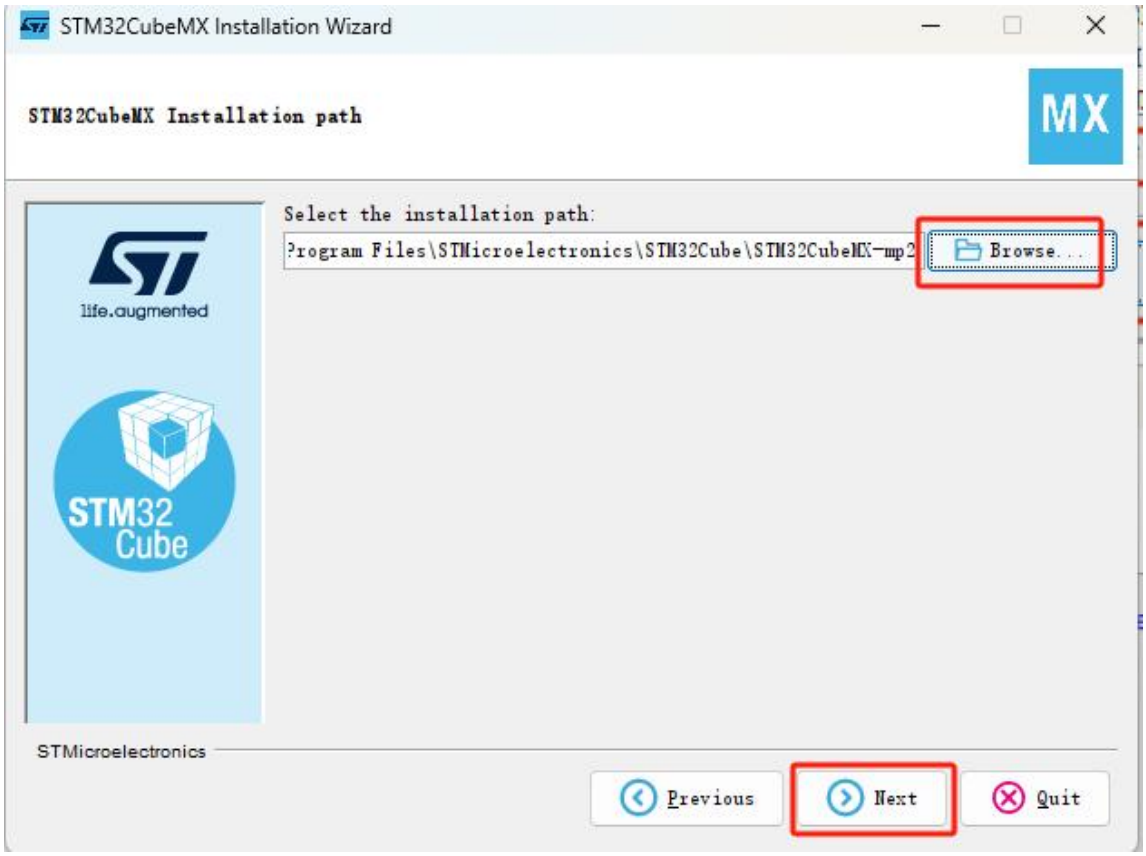
注意：cubemx 版本与 sdk 版本存在绑定关系，请固定使用此版本。

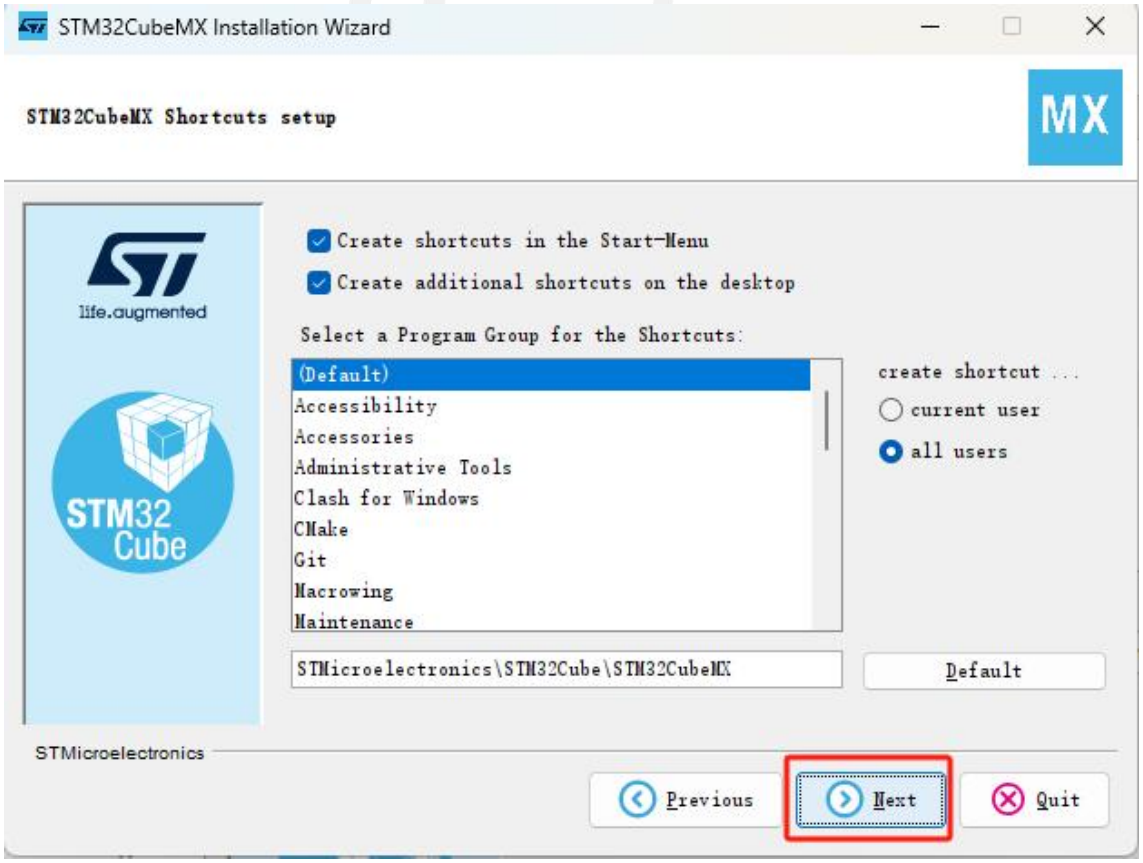


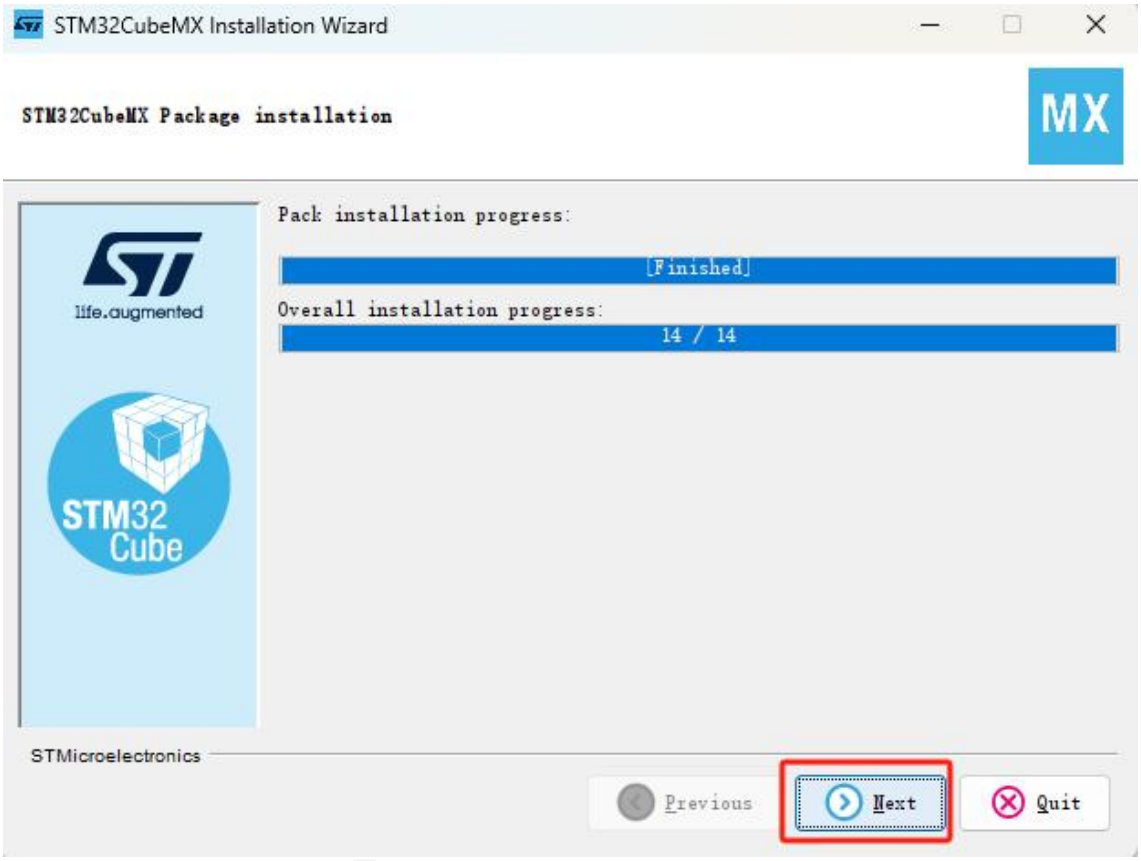


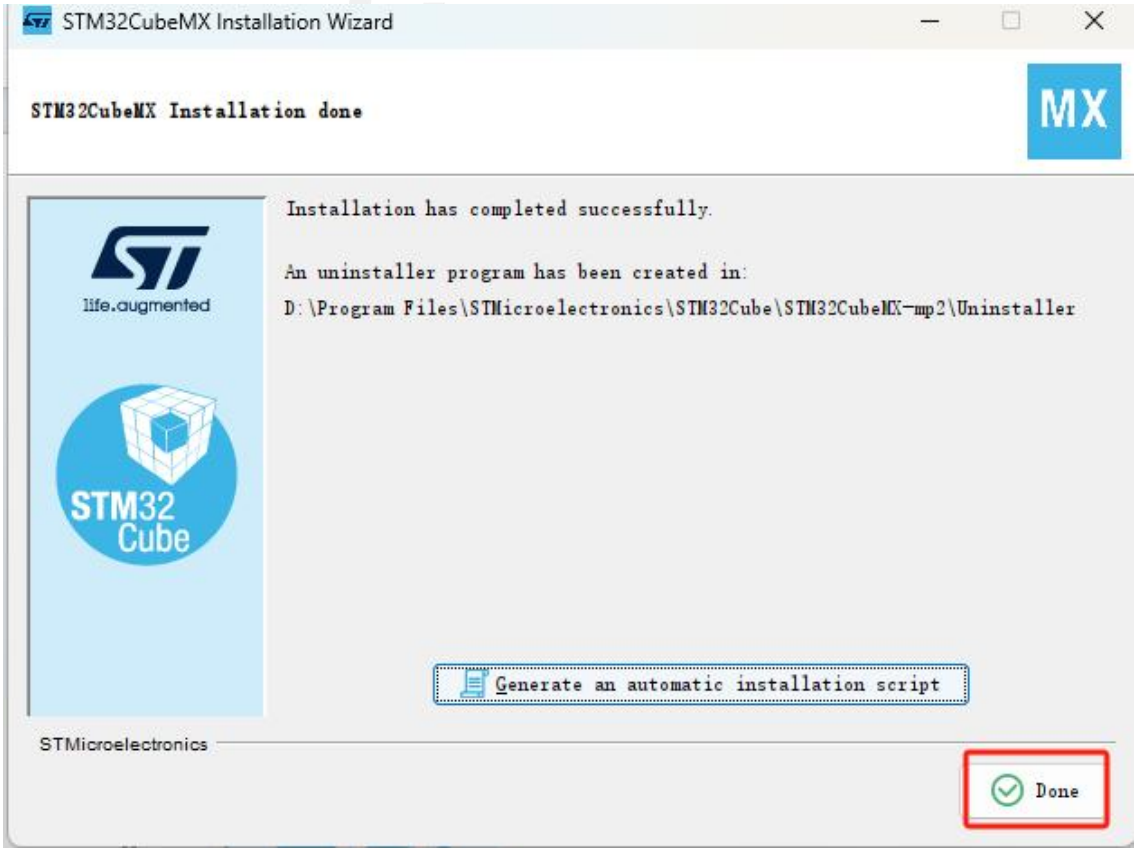

STM32CubeMX Installation Wizard
STM32CubeMX License agreement
MX
Please read the following license agreement carefully:
SOFTWARE PACKAGE LICENSE AGREEMENT
Please indicate your acceptance or NON-acceptance by selecting "I ACCEPT" or "I DO NOT ACCEPT" as indicated below in the media.
BY INSTALLING COPYING, DOWNLOADING, ACCESSING OR OTHERWISE USING THIS SOFTWARE PACKAGE OR ANY PART THEREOF (AND THE RELATED DOCUMENTATION) FROM STMICROELECTRONICS INTERNATIONAL N.V, SWISS BRANCH AND/OR ITS AFFILIATED COMPANIES (STMICROELECTRONICS), THE RECIPIENT, ON BEHALF OF HIMSELF OR HERSELF, OR ON BEHALF OF ANY ENTITY BY WHICH SUCH RECIPIENT IS EMPLOYED AND/OR ENGAGED AGREES TO BE BOUND BY THIS SOFTWARE
I accept the terms of this license agreement.
life.augmented
STM32 Cube
STMicroelectronics
Previous
Next
Quit


STM32CubeMX Installation Wizard
ST Privacy and Terms of Use
MX
I have read and understood the ST Terms of Use.
ST int N.V (as data controller) collects and uses anonymous features usage statistics (directly or by ST affiliates) when you use the application for the purpose of continuously improving the application.
You can stop the collection of your features usage statistics when you use the application at any time with effect for the future
or update your preferences via the menu
Help > User Preferences > General Settings
Data collection information notice
life.augmented
STM32 Cube
STMicroelectronics
Previous
Next
Quit

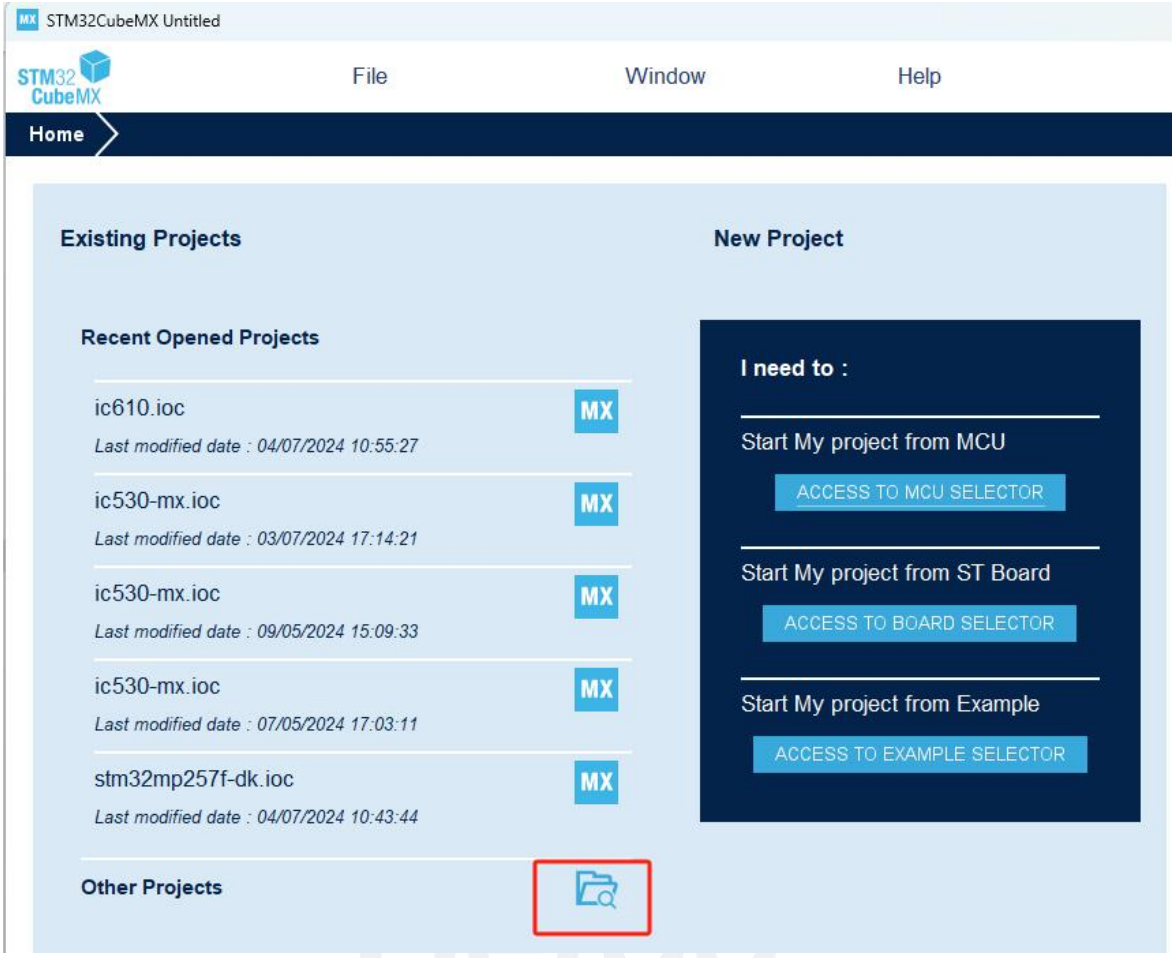

STM32CubeMX Installation Wizard
STM32CubeMX Installation path
MX
Select the installation path:
Program Files\STMicroelectronics\STM32Cube\STM32CubeMX-mp2
Browse...
life.augmented
STM32 Cube
STMicroelectronics
Previous
Next
Quit

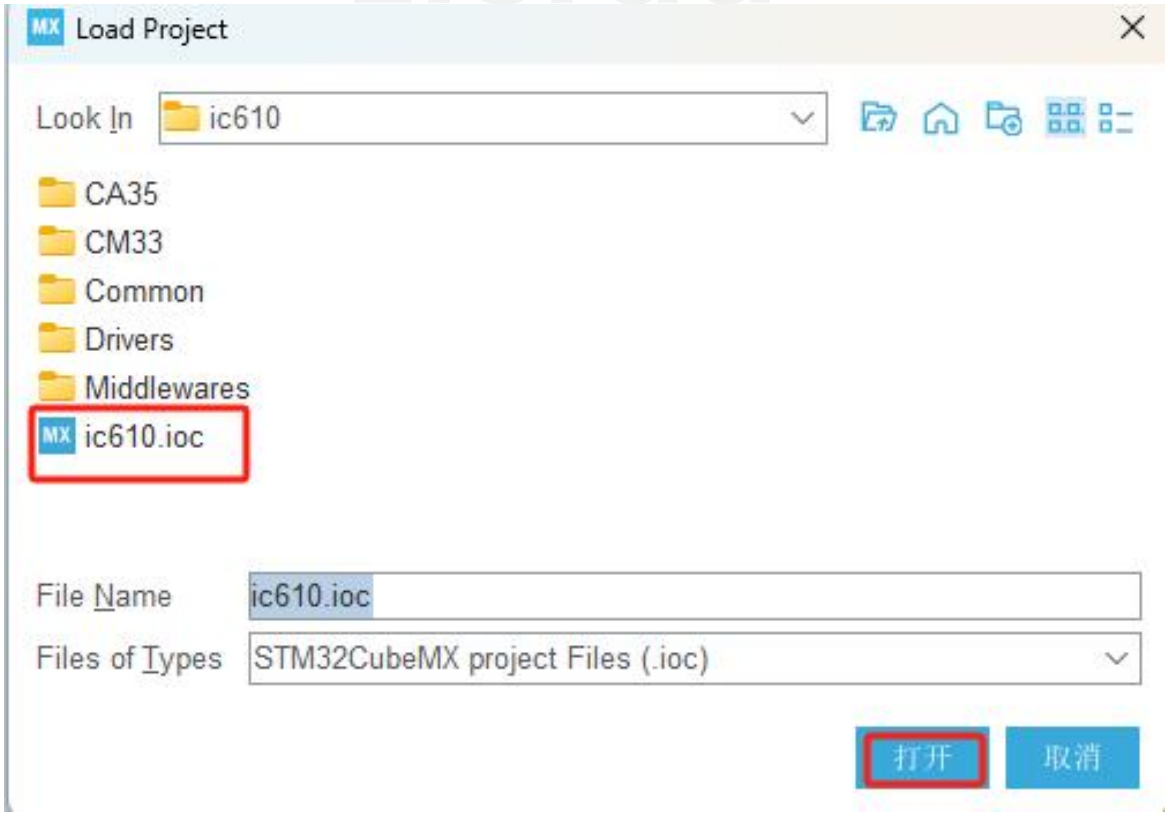

STM32CubeMX Installation Wizard
STM32CubeMX Shortcuts setup
MX
Create shortcuts in the Start-Menu
Create additional shortcuts on the desktop
Select a Program Group for the Shortcuts:
(Default)
Accessibility
Accessories
Administrative Tools
Clash for Windows
CMake
Git
Macrowing
Maintenance
create shortcut ...
current user
all users
STMicroelectronics\STM32Cube\STM32CubeMX
Default
life.augmented
STM32 Cube
STMicroelectronics
Previous
Next
Quit



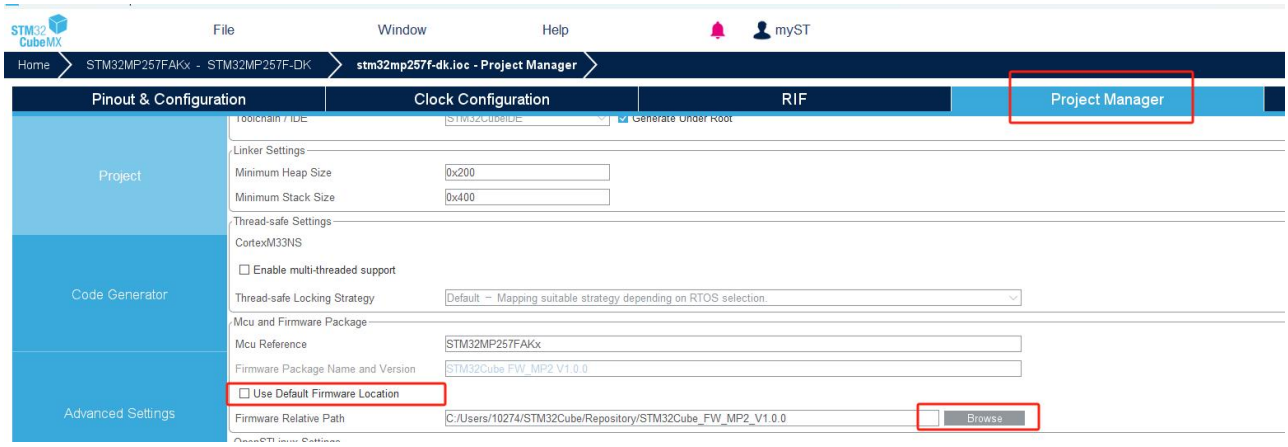


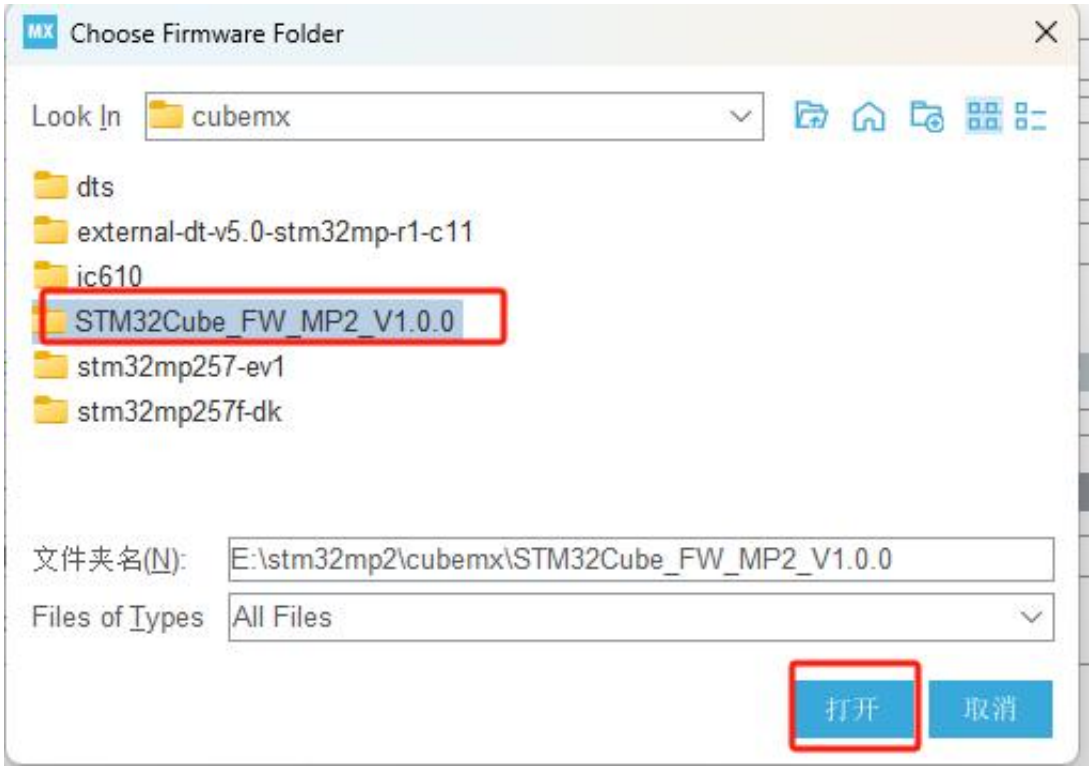
Cubemx 安装后运行桌面快捷方式 STM32CubeMX。


STM32CubeMX Untitled
STM32 CubeMX
File
Window
Help
Home
Existing Projects
New Project
Recent Opened Projects
ic610.ioc
Last modified date : 04/07/2024 10:55:27
ic530-mx.ioc
Last modified date : 03/07/2024 17:14:21
ic530-mx.ioc
Last modified date : 09/05/2024 15:09:33
ic530-mx.ioc
Last modified date : 07/05/2024 17:03:11
stm32mp257f-dk.ioc
Last modified date : 04/07/2024 10:43:44
Other Projects
I need to :
Start My project from MCU
ACCESS TO MCU SELECTOR
Start My project from ST Board
ACCESS TO BOARD SELECTOR
Start My project from Example
ACCESS TO EXAMPLE SELECTOR


Load Project
Look In
ic610
CA35
CM33
Common
Drivers
Middlewares
ic610.ioc
File Name
ic610.ioc
Files of Types
STM32CubeMX project Files (.ioc)
打开
取消

CM33 生成代码需要添加固件，将 tools/stm32cube_fw_mp2_v100.zip，解压至任一路径下，





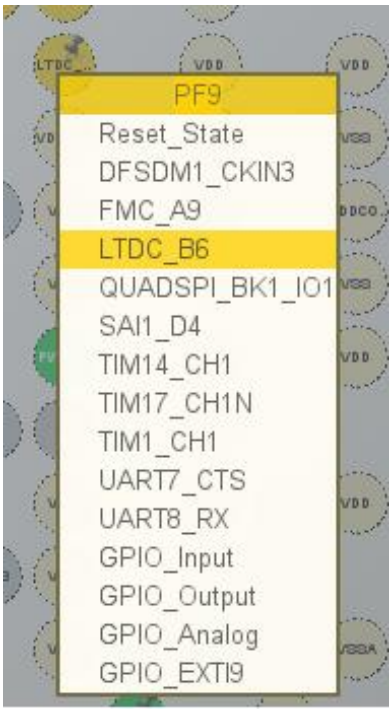
配置完成后，点击 GENERATE CODE 即可生成代码。

4.1.1 Cubemx gpio 使用

黄色：

该 gpio 手动复用为某个功能，该外设功能关闭也不会释放，若需要 cubemx 自动使用该引脚，需要手动释放，右击黄色引脚选择 Reset_State 模式即可。

如下引脚 PF9 手动设置复用为 LTDC_B6，点击 Reset_State 可释放该 gpio。



绿色：

cubemx 自动复用为某个功能，该外设功能关闭，io 自动释放为灰色。

灰色：

cubemx 未使用该 gpio，闲置，配置功能时 cubemx 会根据当前闲置的 gpio，选择复用功能配置。

4.2 Cubemx dts 开发

ic610-sdk\src\cubemx-dts\ic610.rar 为 ic610_evk 对应的 dts，解压后进入 ic610 目录，安装 git 后在 ic610 文件夹内击桌面选择 Git Bash Here，创建 shell 窗口

修改 scp 下配置信息

path=/home/yqa/st/stm32mp2/sdk_v24.11.06

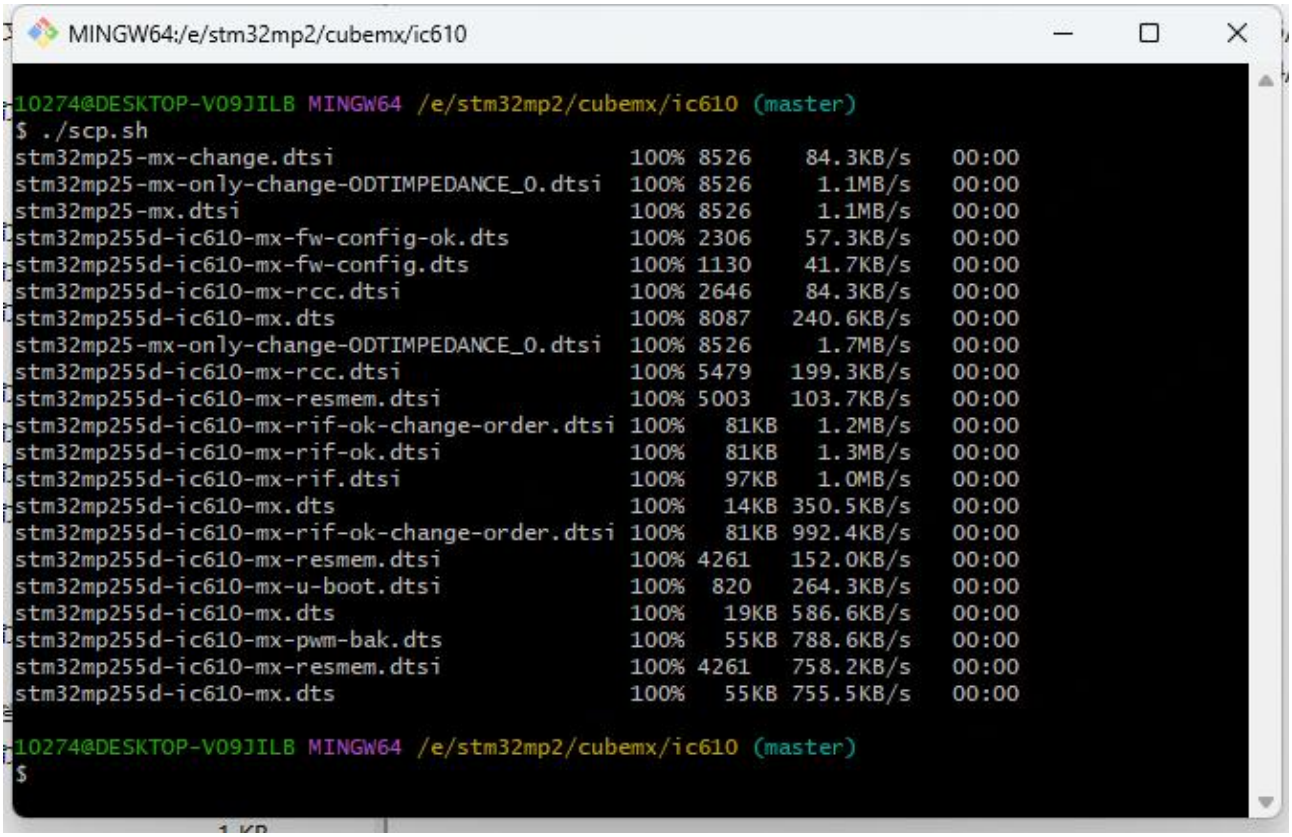
ubuntu_ip=192.168.2.103

user=yqa

path 为 sdk 源码路径，ubuntu_ip 为 ubuntu 开发环境 ip（开发环境和 windows 在同一个局域网内）

user 为 ubuntu 的用户名。

修改完成后运行./scp.sh 将镜像更新到编译源码包内。



Cubemx 端修改 dts 请参考《Lierda_IC530&IC610_ST_cubemx 应用指导_Rev1.0》及相关应用指导。

Lierda 利尔达