

# M33 核开发指导手册

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

## 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。

## 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

# 目录

法律声明 .....	1
文件修订历史 .....	2
目录 .....	3
1 引言 .....	4
2 M33 核环境准备 .....	5
2.1 Cmake 和 Python .....	5
2.2 CubeIDE 安装 .....	5
2.3 M33 固件包 1.0.0 下载 .....	6
2.4 导入工作区的固件工程 .....	7
3 M33 demo 编译及运行 .....	12
3.1 编译 .....	12
3.2 A35 下运行 OpenAMP_TTY_echo .....	14
4 CubeIDE 下调试运行 M33 .....	17
4.1 搭建运行环境 .....	17
4.1.1 连接串口线，设置串口号，使 CubeIDE 连接 A35 的调试串口。 .....	17
4.1.2 连接网线，ping 通开发板和 PC 网络，确保同一个内网。 .....	18
4.2 使用 CubeIDE、ST-Link 烧录运行 M33 核（运行之前要编译，参考 2.1） .....	19
4.2.1 连接 ST-link，确认 CubeIDE 识别到 ST-link .....	19
4.2.2 配置运行参数，串口号，开发板 IP 地址等信息。 .....	19
4.2.3 确保网络连接后，烧录并运行程序 .....	22
4.3 使用 CubeIDE、ST-Link 调试 M33 核（在运行 M33 核之后进行调试） .....	23
4.3.1 进行调试配置 .....	23
4.3.2 开始调试程序 .....	26

# 1 引言

本文档依托 IC610 evk，旨在介绍使用 STM32CubeIDE 如何调试 M33 核应用程序。

## 2 M33 核环境准备

### 2.1 Cmake 和 Python

CubeIDE 依赖 python 和 CMAKE 故需要先按照 cmake 和 python。

CMake 下载链接: [Download CMake](#)

python 下载链接: <https://www.python.org/downloads/>

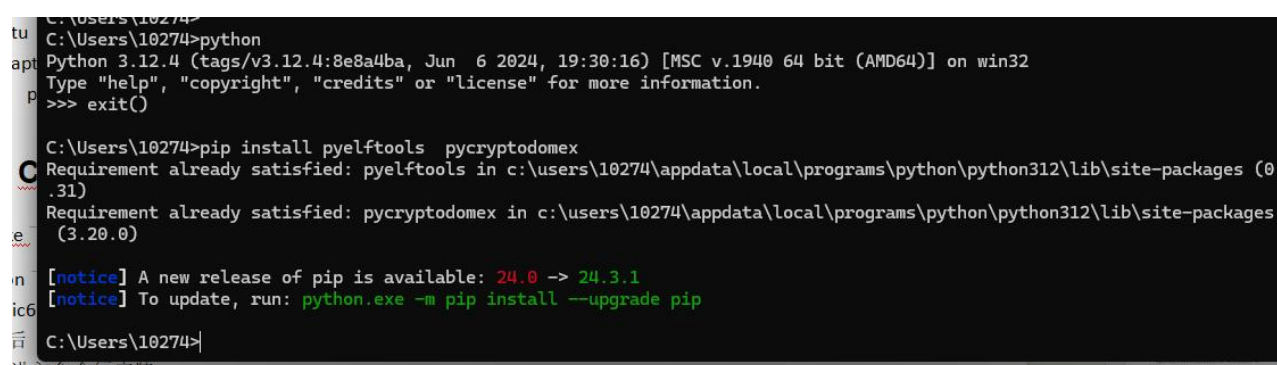
安装 ic610-sdk\tools\下 cmake-3.29.5-windows-x86\_64.msi 和

python-3.12.4-amd64.exe

下载后

Cmd 进入命令行安装

pip install pyelftools pycryptodomex



```
C:\Users\10274>python
Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun 6 2024, 19:30:16) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> exit()

C:\Users\10274>pip install pyelftools pycryptodomex
Requirement already satisfied: pyelftools in c:\users\10274\appdata\local\programs\python\python312\lib\site-packages (0.31)
Requirement already satisfied: pycryptodomex in c:\users\10274\appdata\local\programs\python\python312\lib\site-packages (3.20.0)

[notice] A new release of pip is available: 24.0 -> 24.3.1
[notice] To update, run: python.exe -m pip install --upgrade pip

C:\Users\10274>
```

Ubuntu

sudo apt-get install cmake

sudo pip install pyelftools pycryptodomex

### 2.2 CubeIDE 安装

Windows:

安装 ic610-sdk\tools\st-stm32cubeide\_1.16.0\_21983\_20240628\_1741\_x86\_64.exe 或

下载更新版本

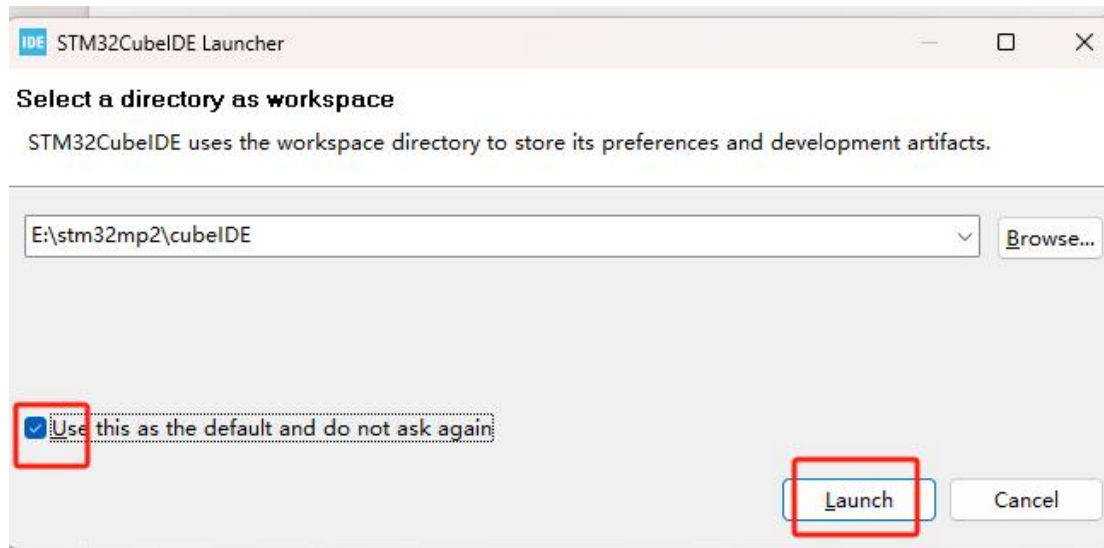
CubeIDE 下载:

[STM32CubeIDE - Integrated Development Environment for STM32 -](#)

## STMicroelectronics

【用管理员权限安装和使用 CubeIDE,并将工作区放在无中文位置】

设置默认工作目录为 E:\stm32mp2\cubeIDE



Ubuntu:

```
cp st-stm32cubeide_1.16.0_21983_20240628_1741_amd64.deb_bundle.sh
/home/hjl/STCubeIDE
sudo chmod 777 st-stm32cubeide_1.16.0_21983_20240628_1741_amd64.deb_bundle.sh
sudo ./st-stm32cubeide_1.16.0_21983_20240628_1741_amd64.deb_bundle.sh
```

```
I ACCEPT (y) / I DO NOT ACCEPT (N) [N/y] y
License accepted.
Do you want to install Segger J-Link udev rules? [Y/n] n
Reading package lists... Done
```

## 2.3 M33 固件包 1.0.0 下载

STM32Cube\_FW\_MP2\_V1.0.0 下载:

[STM32CubeMP2 - STM32CubeMP2 Package for STM32MP2 series \(HAL, Low-Layer APIs and CMSIS, OpenAMP for Inter processor communication, FreeRTOS, STM32 USBPD core/device library and examples running on ST boards - STMicroelectronics\)](#)

Hardware abstraction layer (HAL), enabling portability between different STM32 devices via standardized API calls

Low-layer API (LL), a light-weight, optimized, expert oriented set of APIs designed for runtime efficiency and so enhanced performance

Board support package drivers (BSP), based on HAL drivers, an API set for the evaluation board and third-party components

Collection of middleware components (such as FreeRTOS)

Collection of examples, running on STM32MP25xx-EVx evaluation boards and STM32MP25xx-DK discovery board and allowing demonstration of a basic implementation of features from a set of peripherals.

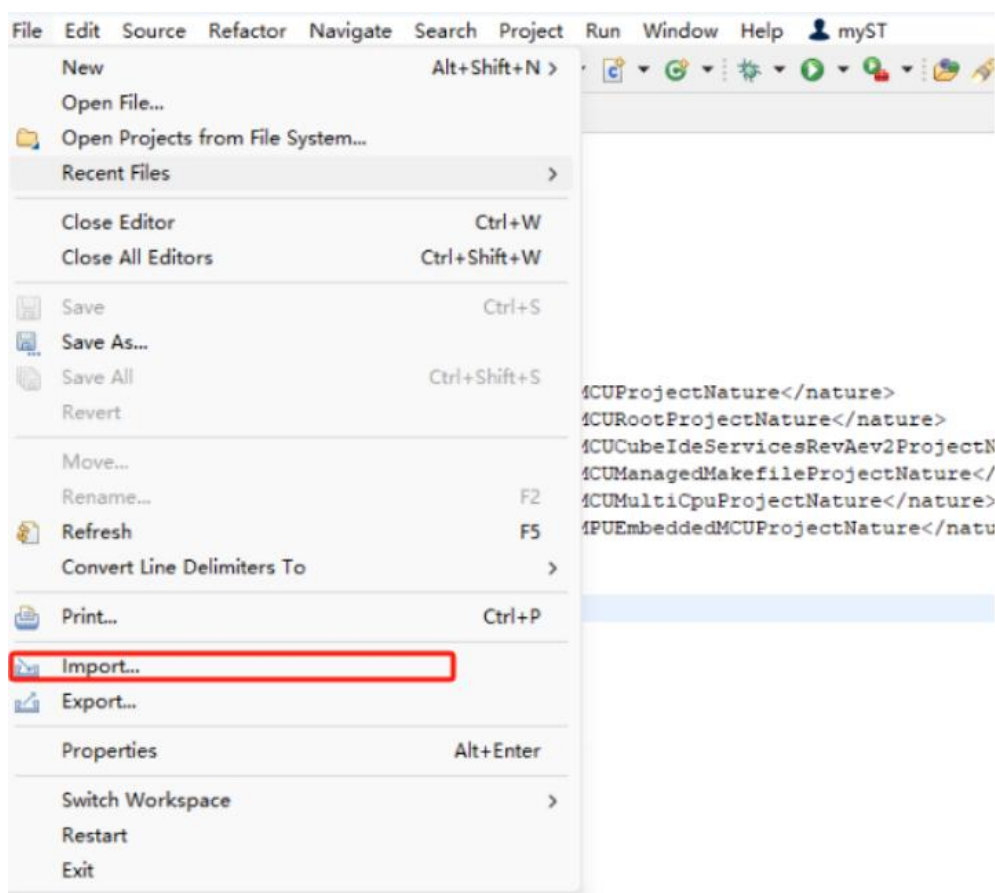
或用 cubemx 自动下载到 C:\Users\lsd\STM32Cube\Repository



将固件包 STM32Cube\_FW\_MP2\_V1.0.0 复制到 CubeIDE 工作目录

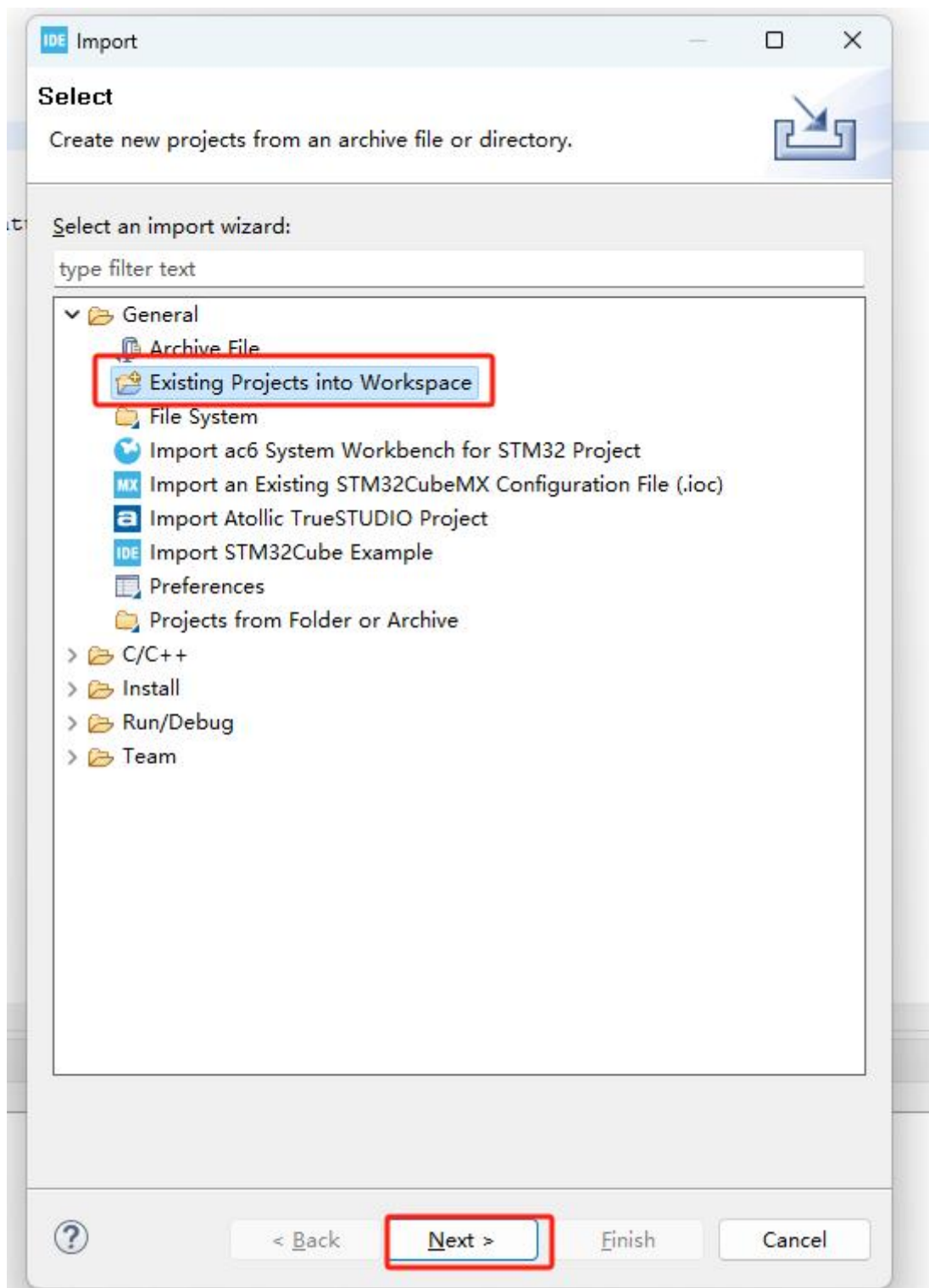
E:\stm32mp2\cubeIDE

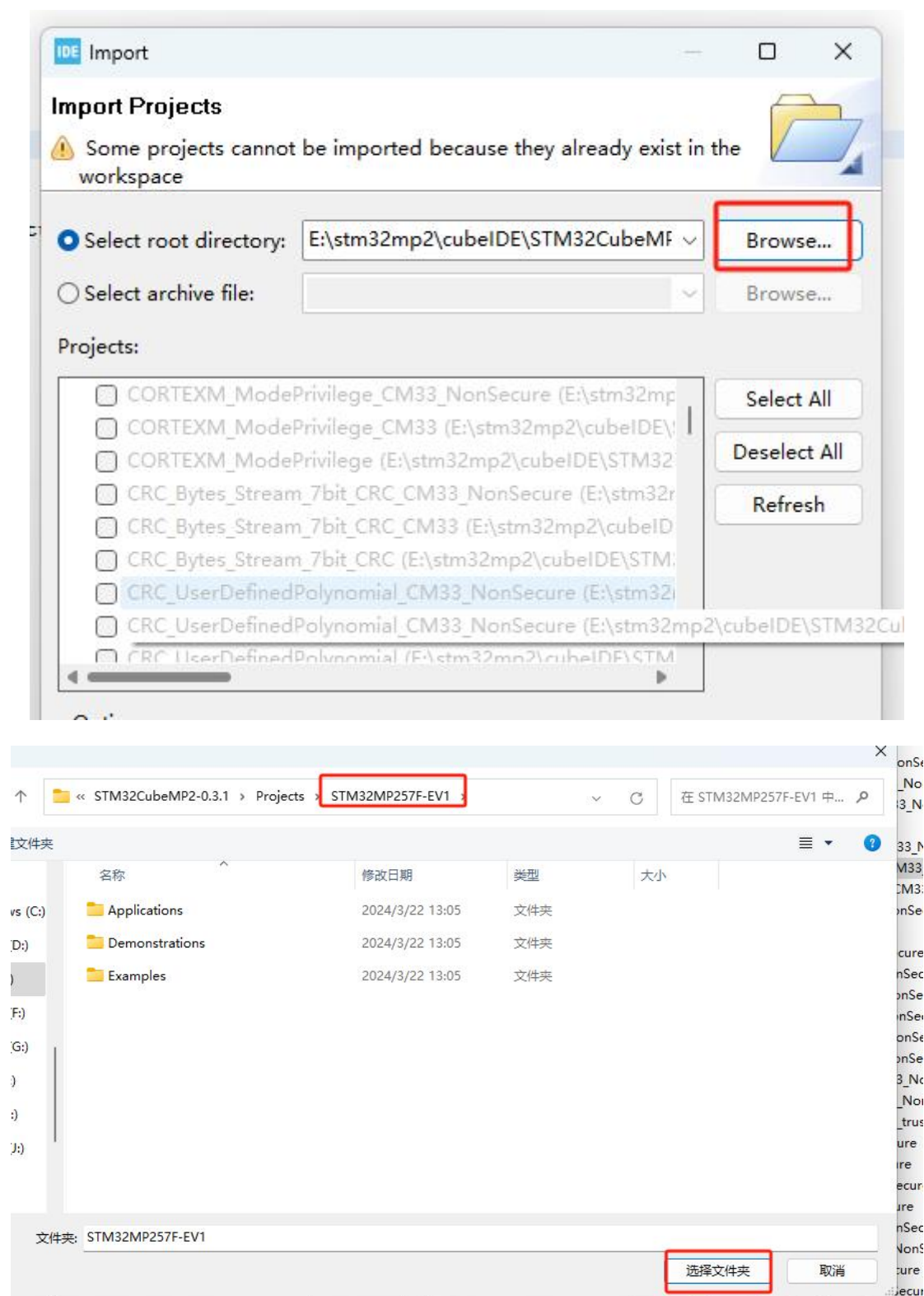
## 2.4 导入工作区的固件工程

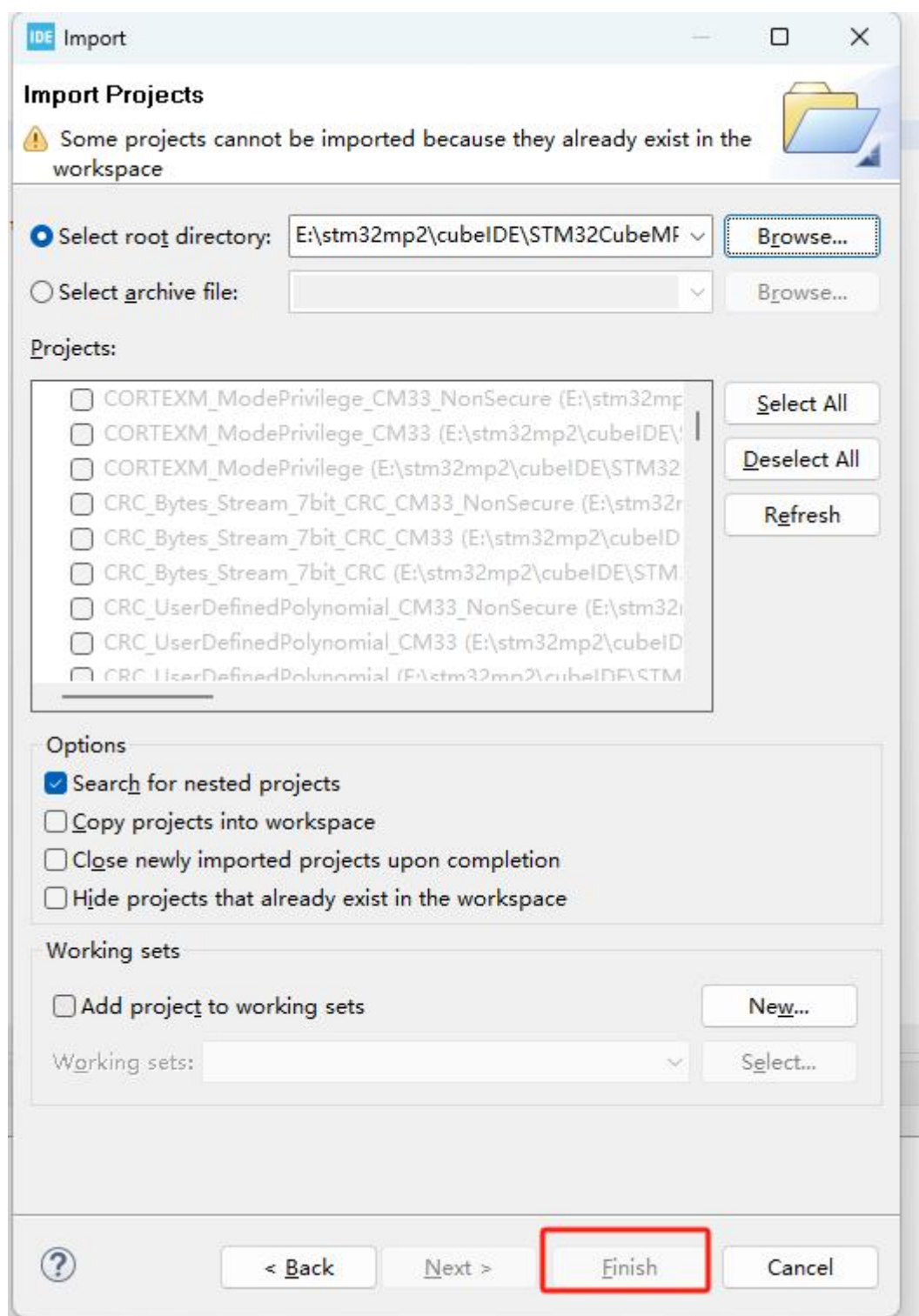


导入工作目录内所有 project

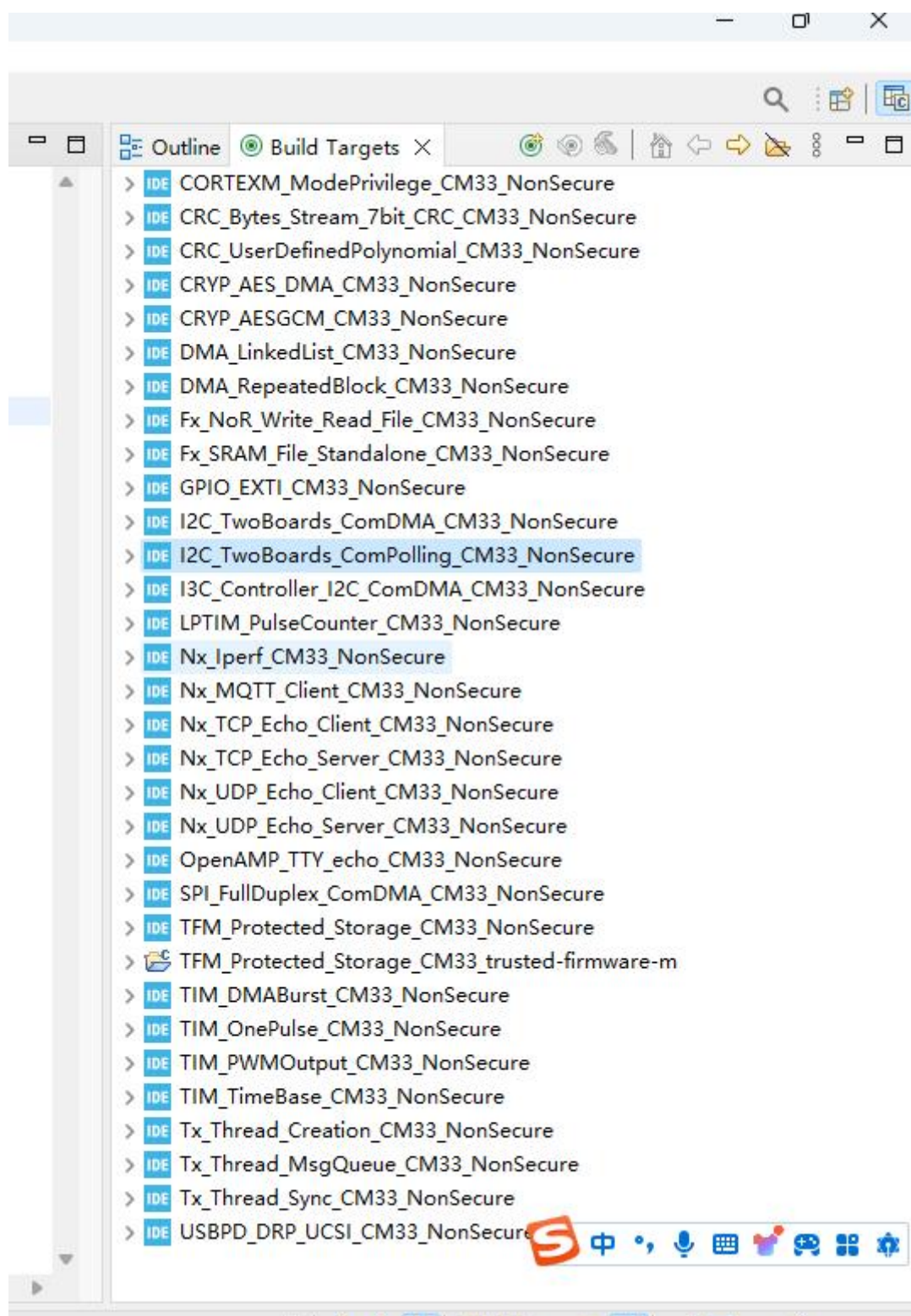








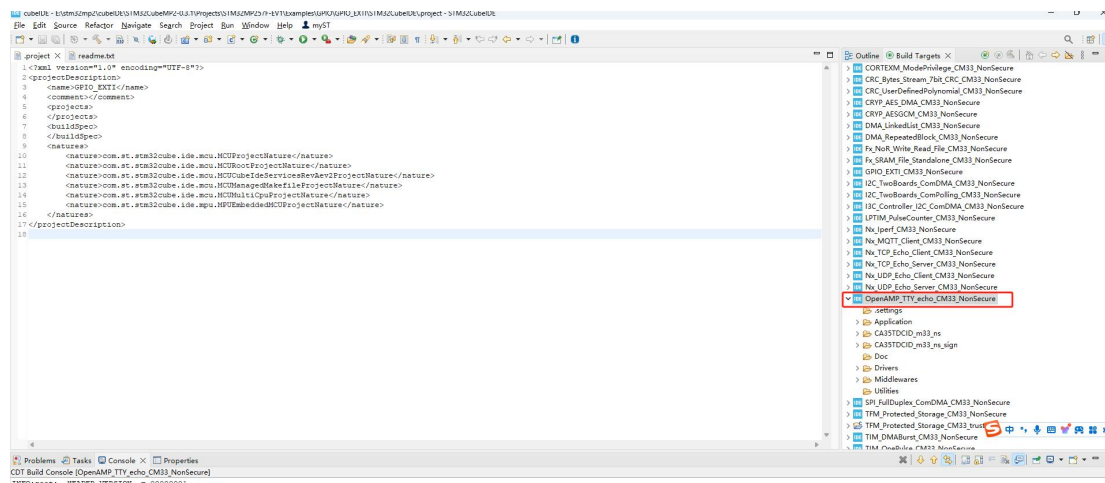
导入完成后



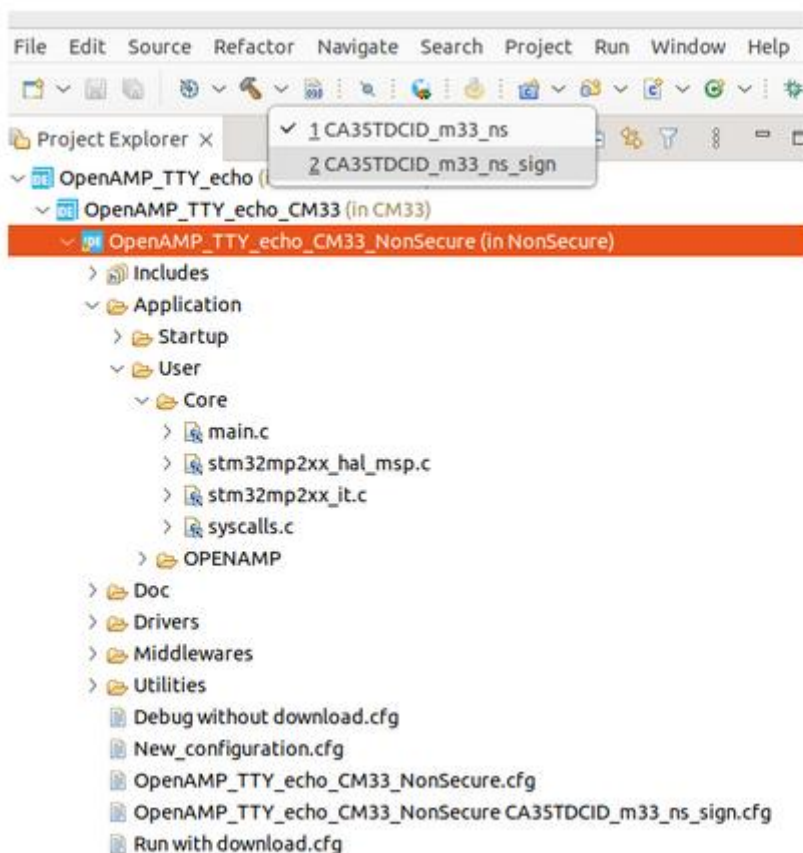
## 3 M33 demo 编译及运行

### 3.1 编译

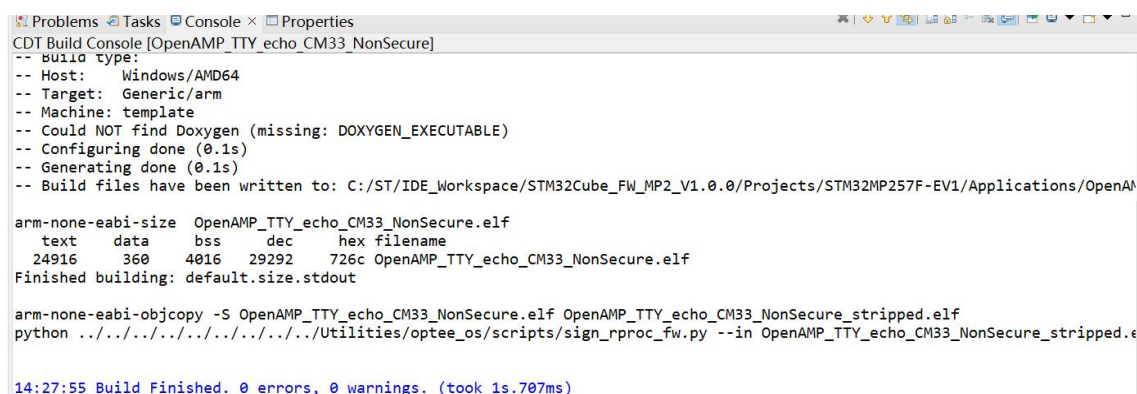
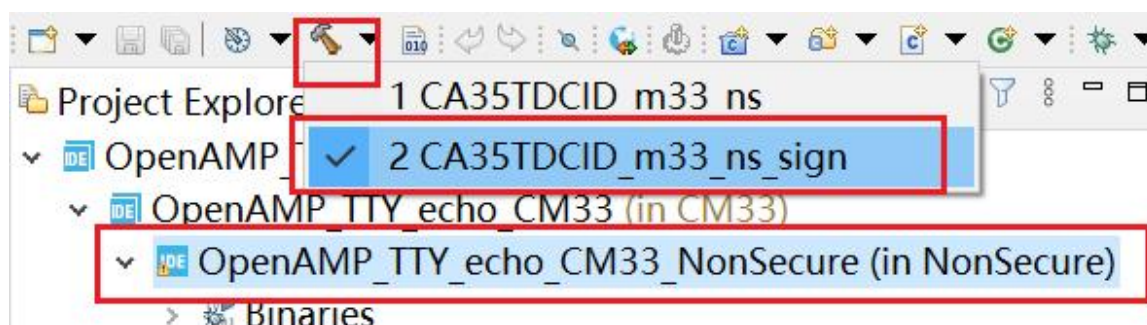
选中项目编译，选择 OpenAMP\_TTY\_Echo\_CM33\_NonSecure



选择 CA35TDCID\_m33\_ns\_sign 并编译

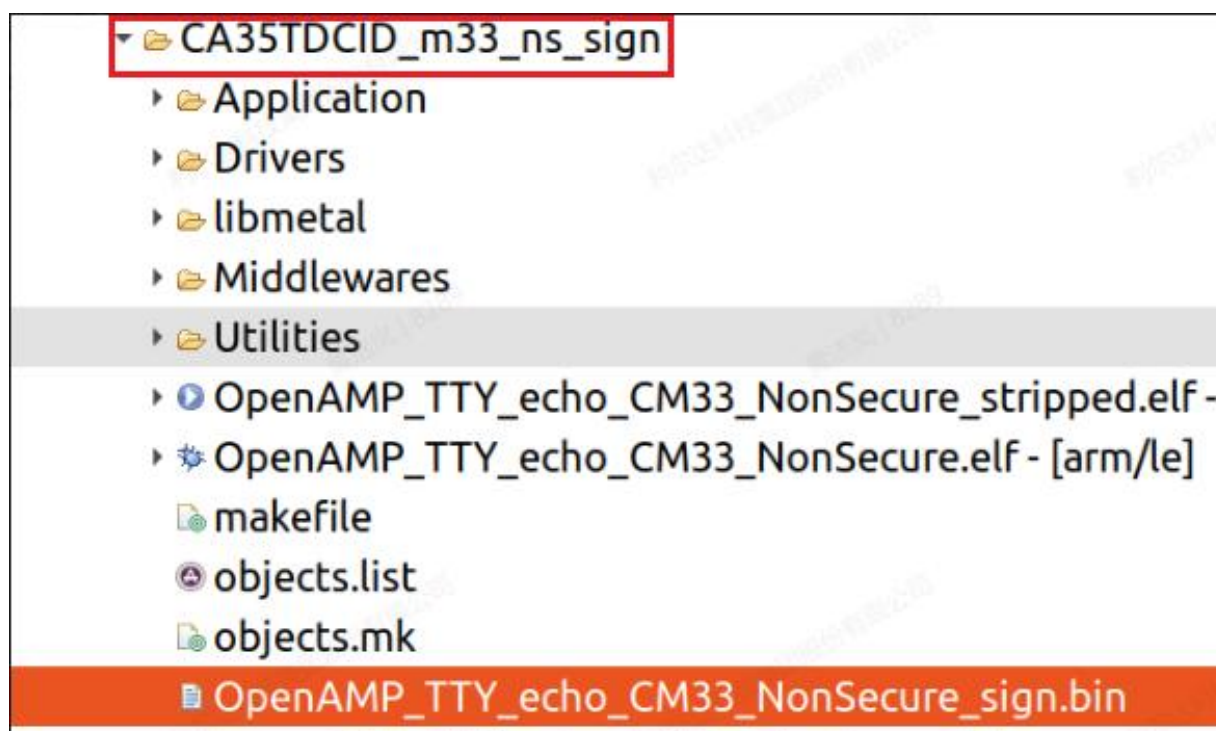






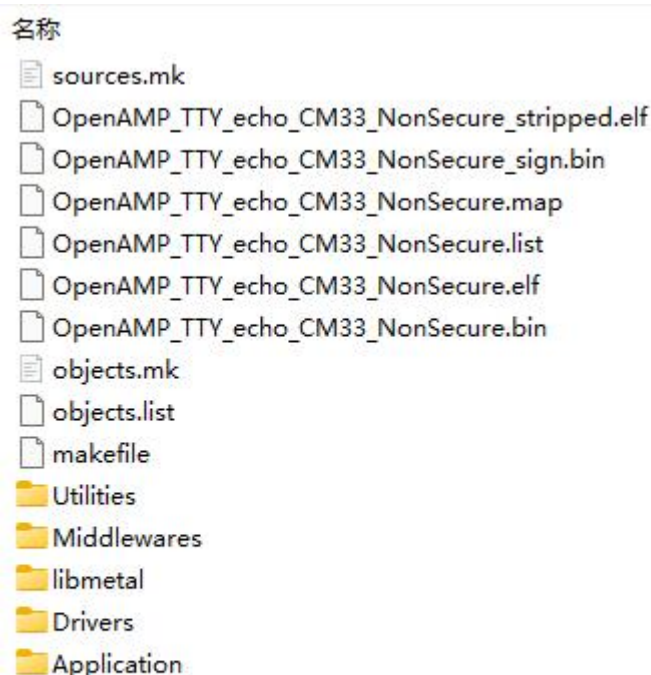
## 编译结果

产生 OpenAMP\_TTY\_echo\_CM33\_NonSecure\_sign.bin 文件、两个 elf 文件、\_sign.list 文件



编译完成后

E:\stm32mp2\cubeIDE\STM32Cube\_FW\_MP2\_V1.0.0\Projects\STM32MP257F-EV1\Applications\OpenAMP\OpenAMP\_TTY\_echo\STM32CubeIDE\CM33\NonSecure\CA35TDCID\_m33\_ns\下生成



## 3.2 A35 下运行 OpenAMP\_TTY\_echo

OpenAMP\_TTY\_echo\_CM33\_NonSecure.elf 为 A35 端运行镜像，OpenAMP\_TTY\_echo\_CM33\_NonSecure.bin 为 IDE 端下载运行镜像。

创建目录 OpenAMP\_TTY\_echo，该名称必须与 M33 工程名称相同，并在该目录下创建 lib/firmware/，串口工具连接 M33 debug 接口，波特率为 115200

```
root@stm32mp2:~#mkdir -p OpenAMP_TTY_echo/lib/firmware/
```

将 fw\_cortex\_m33.sh（该脚本在板子端 /usr/local/Cube-M33-examples/STM32MP215F-DK/Demonstrations/OpenAMP/OpenAMP\_TTY\_echo/fw\_cortex\_m33.sh）复制到 OpenAMP\_TTY\_echo 目录下并赋予可执行权限，将 OpenAMP\_TTY\_echo\_CM33\_NonSecure.elf 复制到 OpenAMP\_TTY\_echo/lib/firmware/

```
root@stm32mp2:~/OpenAMP_TTY_echo# ls
```

```
fw_cortex_m33.sh  lib
```

```
root@stm32mp2:~/OpenAMP_TTY_echo# chmod 775 fw_cortex_m33.sh
```

```
root@stm32mp2:~/OpenAMP_TTY_echo# ls lib/firmware/
```

```
OpenAMP_TTY_echo_CM33_NonSecure.elf
```

```
root@stm32mp2:~/OpenAMP_TTY_echo# ./fw_cortex_m33.sh start
```

```
fw_cortex_m33.sh: fmw_name=OpenAMP_TTY_echo_CM33_NonSecure.elf
```

```
[319790.672182] remoteproc remoteproc0: powering up m33
```

```
[319790.678740] remoteproc remoteproc0: Booting fw image
```

```
OpenAMP_TTY_echo_CM33_NonSecure.elf, size 3413720
```

```
[319790.689617] rproc-virtio rproc-virtio.1.auto: assigned reserved memory node
vdev0buffer@812fa000
```

```
[319790.695864] virtio_rpmsg_bus virtio0: rpmsg host is online
```

```
[319790.699606] rproc-virtio rproc-virtio.1.auto: registered virtio0 (type 7)
```

```
[319790.705170] virtio_rpmsg_bus virtio0: creating channel rpmsg-tty addr 0x400
```

```
[319790.708541] remoteproc remoteproc0: remote processor m33 is now up
```

```
[319790.720598] virtio_rpmsg_bus virtio0: creating channel rpmsg-tty addr 0x401
```

```
root@stm32mp2:~/OpenAMP_TTY_echo# ls /dev/ttyRPMSG*
```

```
/dev/ttyRPMSG0 /dev/ttyRPMSG1
```

驱动加载后生成节点/dev/ttyRPMSG0 和 /dev/ttyRPMSG1

```
root@stm32mp2:~/OpenAMP_TTY_echo# echo "Hello Virtual
```

```
UART0" >/dev/ttyRPMSG0
```

```
root@stm32mp2:~/OpenAMP_TTY_echo# echo "Hello Virtual UART1" >/dev/ttyRPMSG1
```

信息输出到 M33 核。



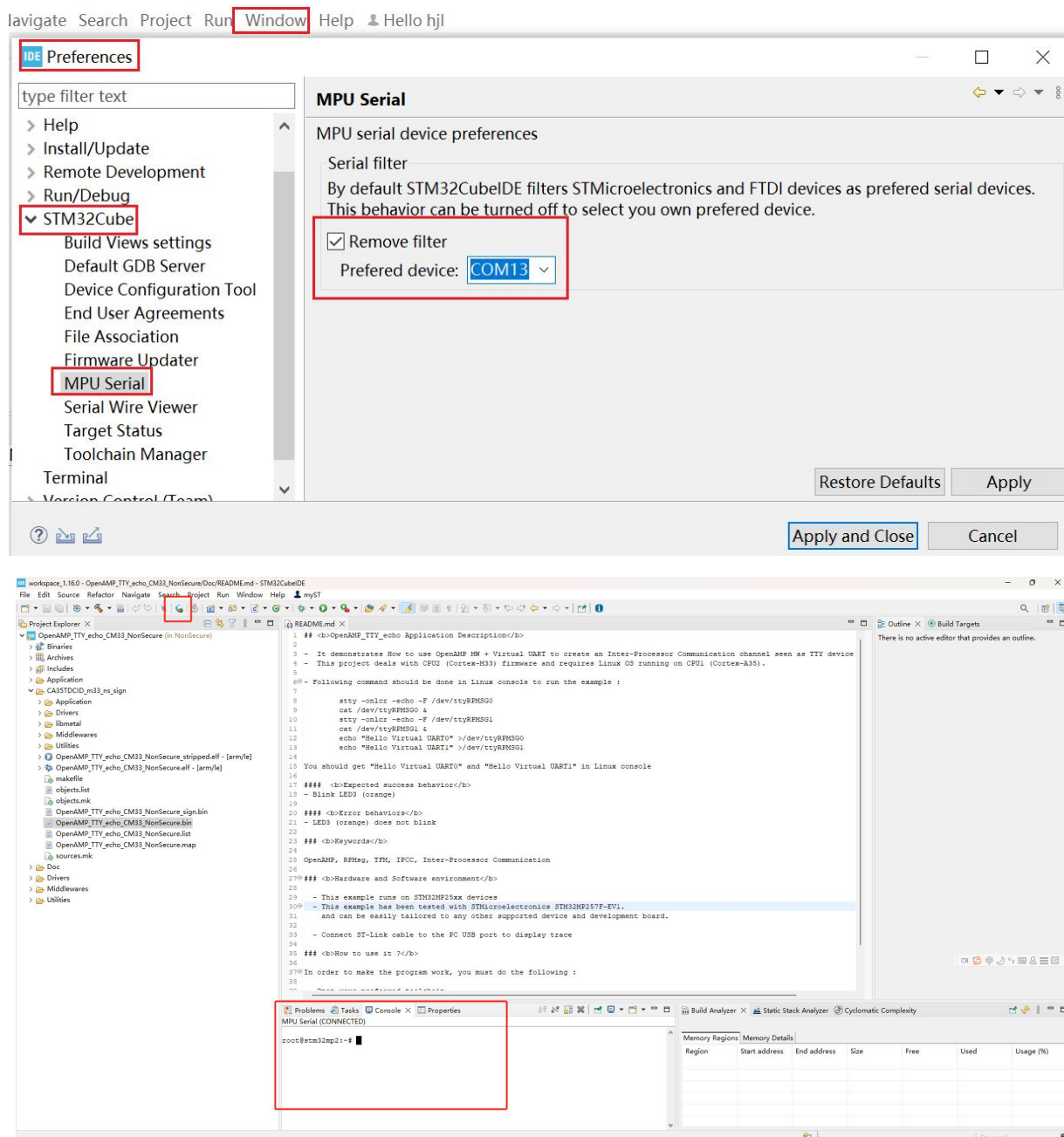
```
ubuntu1804 [1] serial-com11 x serial-com9 x
root@stm32mp2:~/OpenAMP_TTY_echo#
root@stm32mp2:~/OpenAMP_TTY_echo#
root@stm32mp2:~/OpenAMP_TTY_echo# ls
fw_cortex_m33.sh lib
root@stm32mp2:~/OpenAMP_TTY_echo# ls lib/
firmware
root@stm32mp2:~/OpenAMP_TTY_echo# ls lib/firmware/
OpenAMP_TTY_echo_CM33_NonSecure.elf
root@stm32mp2:~/OpenAMP_TTY_echo# ls /dev/ttyRPMMSG*
ls: cannot access '/dev/ttyRPMMSG*': No such file or directory
root@stm32mp2:~/OpenAMP_TTY_echo# ./fw_cortex_m33.sh start
fw_cortex_m33.sh: fwm_name=OpenAMP_TTY_echo_CM33_NonSecure.elf
[319790.672182] remoteproc remoteproc0: powering up m33
[319790.678740] remoteproc remoteproc0: Booting fw image OpenAMP_TTY_echo_CM33_NonSecure.elf, size 3413720
[319790.689617] rproc-virtio rproc-virtio.1.auto: assigned reserved memory node vdev0buffer@812fa000
[319790.693864] virtio_rpmsg_bus virtio0: rpmsg host is online
[319790.699606] rproc-virtio rproc-virtio.1.auto: registered virtio0 (type 7)
[319790.705170] virtio_rpmsg_bus virtio0: creating channel rpmsg-tyt add r 0x400
[319790.708541] remoteproc remoteproc0: remote processor m33 is now up
[319790.720598] virtio_rpmsg_bus virtio0: creating channel rpmsg-tyt add r 0x401
root@stm32mp2:~/OpenAMP_TTY_echo# ls /dev/ttyRPMMSG*
/dev/ttyRPMMSG0 /dev/ttyRPMMSG1
root@stm32mp2:~/OpenAMP_TTY_echo# echo "Hello Virtual UART0" >/dev/ttyRPMMSG0
root@stm32mp2:~/OpenAMP_TTY_echo# echo "Hello Virtual UART1" >/dev/ttyRPMMSG1
root@stm32mp2:~/OpenAMP_TTY_echo#
```

Starting OpenAMP application (Nov 29 2024: 16:50:11)  
Virtual UART0 OpenAMP-rpmsg channel creation  
Virtual UART1 OpenAMP-rpmsg channel creation  
Msg received on VIRTUAL UART0 channel: Hello Virtual UART0  
Msg received on VIRTUAL UART1 channel: Hello Virtual UART1

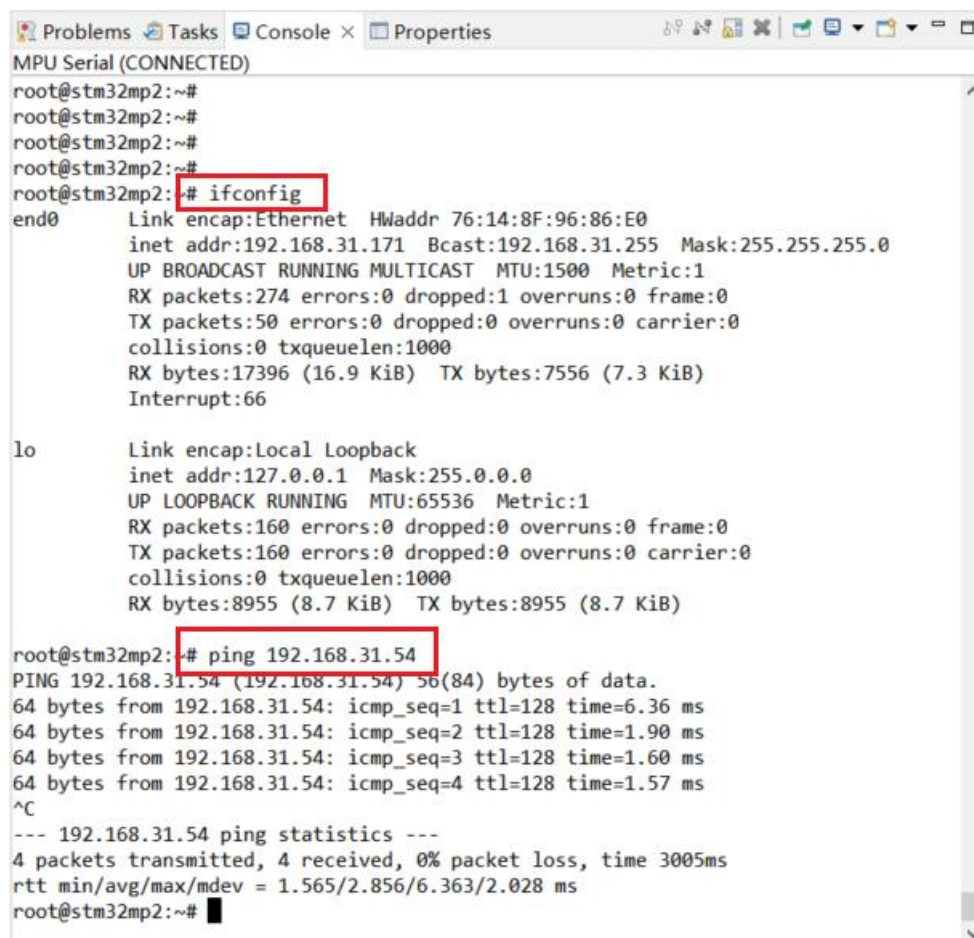
## 4 CubeIDE 下调试运行 M33

### 4.1 搭建运行环境

#### 4.1.1 连接串口线，设置串口号，使 CubeIDE 连接 A35 的调试串口。



## 4.1.2 连接网线，ping 通开发板和 PC 网络，确保同一个内网。



```

MPU Serial (CONNECTED)
root@stm32mp2:~#
root@stm32mp2:~#
root@stm32mp2:~#
root@stm32mp2:~#
root@stm32mp2:~# ifconfig
end0      Link encap:Ethernet  HWaddr 76:14:8F:96:86:E0
          inet addr:192.168.31.171  Bcast:192.168.31.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:274  errors:0  dropped:1  overruns:0  frame:0
          TX packets:50  errors:0  dropped:0  overruns:0  carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:17396 (16.9 KiB)  TX bytes:7556 (7.3 KiB)
          Interrupt:66

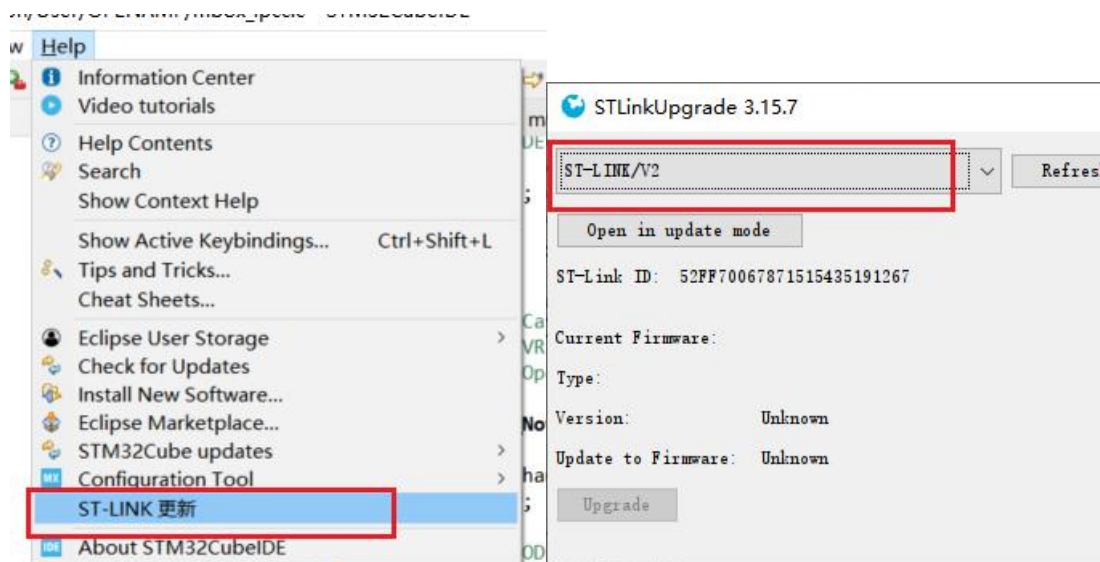
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:160  errors:0  dropped:0  overruns:0  frame:0
          TX packets:160  errors:0  dropped:0  overruns:0  carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:8955 (8.7 KiB)  TX bytes:8955 (8.7 KiB)

root@stm32mp2:~# ping 192.168.31.54
PING 192.168.31.54 (192.168.31.54) 56(84) bytes of data.
64 bytes from 192.168.31.54: icmp_seq=1 ttl=128 time=6.36 ms
64 bytes from 192.168.31.54: icmp_seq=2 ttl=128 time=1.90 ms
64 bytes from 192.168.31.54: icmp_seq=3 ttl=128 time=1.60 ms
64 bytes from 192.168.31.54: icmp_seq=4 ttl=128 time=1.57 ms
^C
--- 192.168.31.54 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 1.565/2.856/6.363/2.028 ms
root@stm32mp2:~#

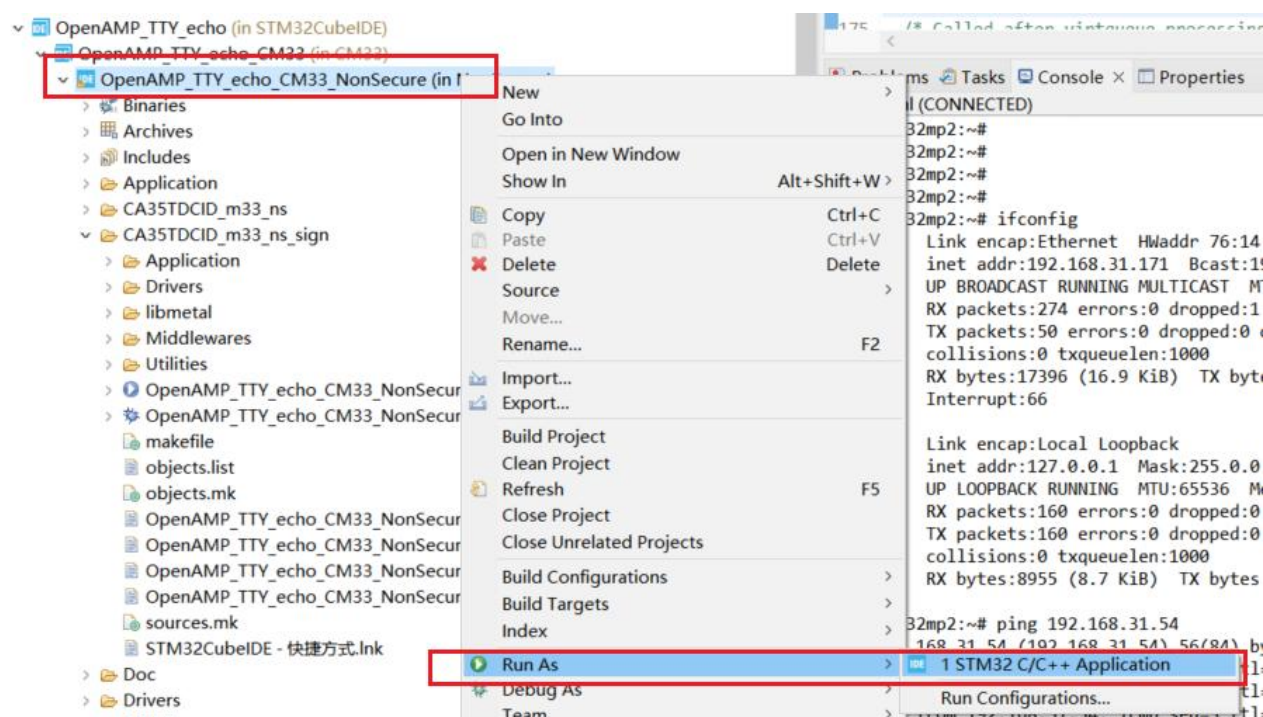
```

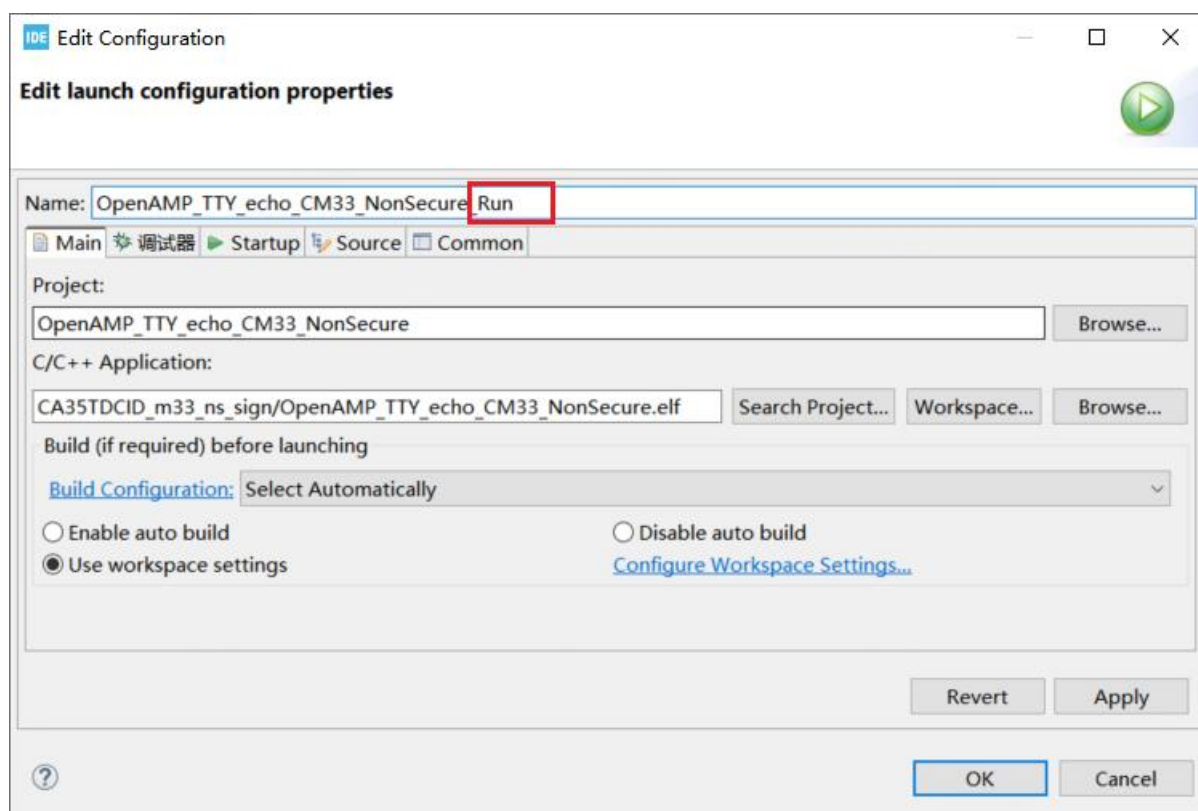
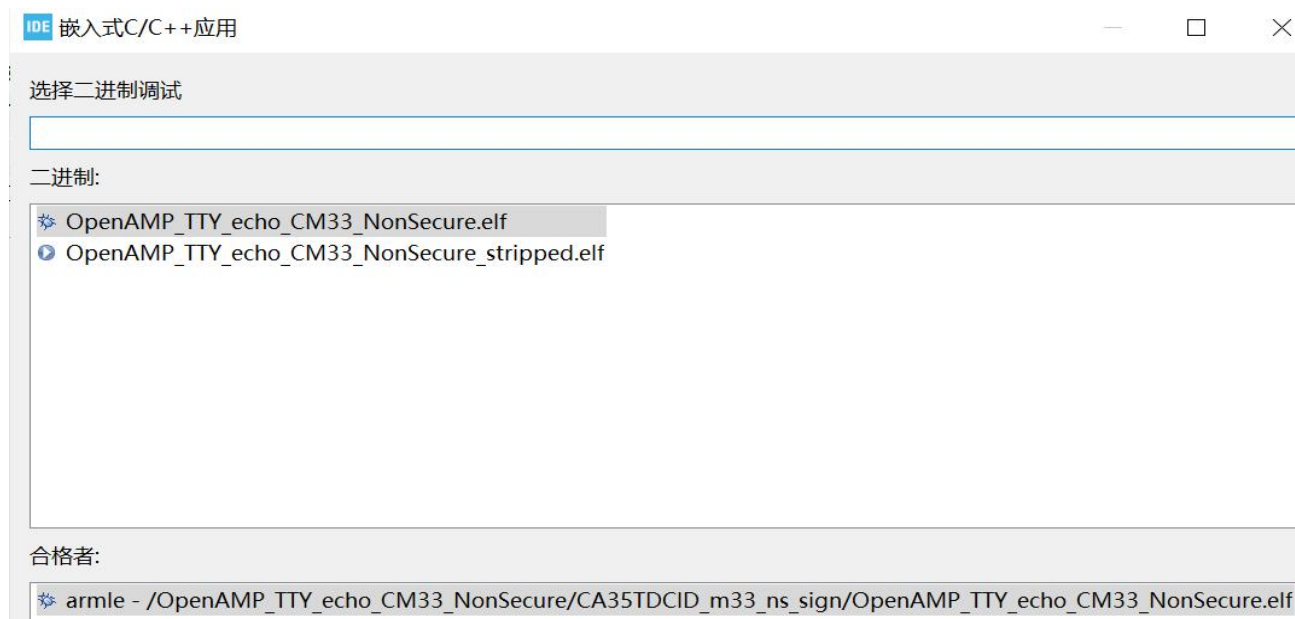
## 4.2 使用 CubeIDE、ST-Link 烧录运行 M33 核（运行之前要编译，参考 2.1）

### 4.2.1 连接 ST-link，确认 CubeIDE 识别到 ST-link

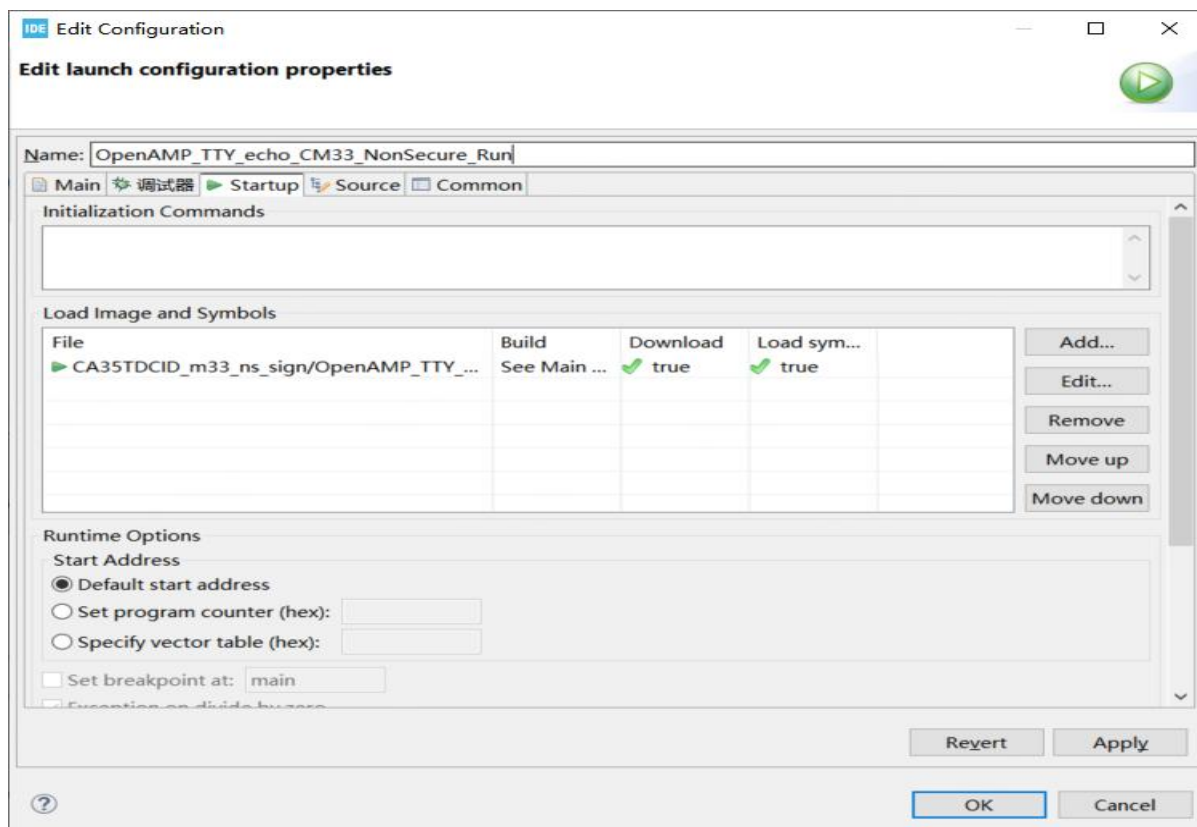
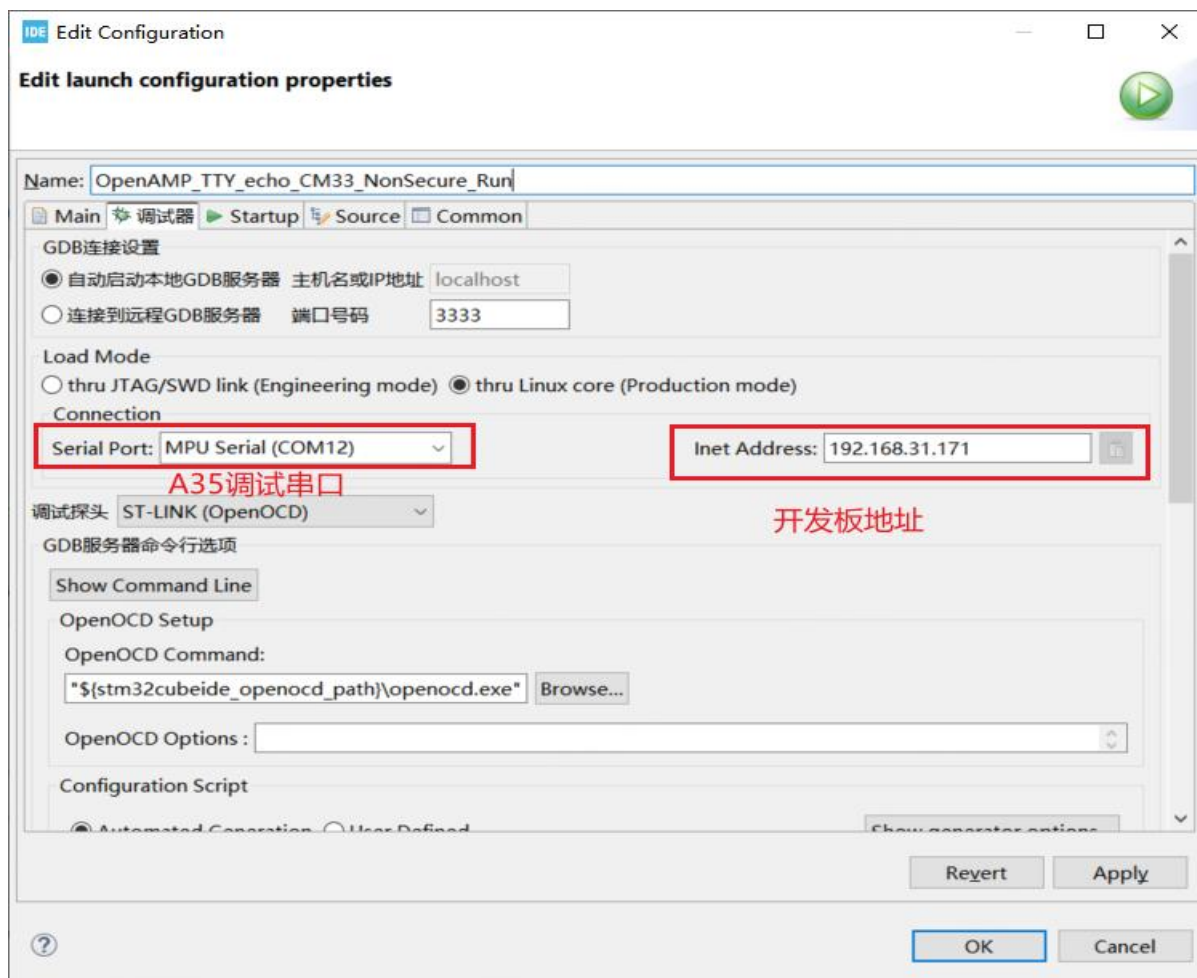


### 4.2.2 配置运行参数，串口号，开发板 IP 地址等信息。

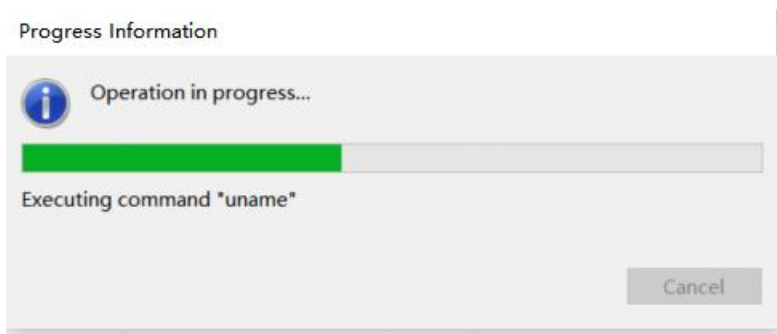
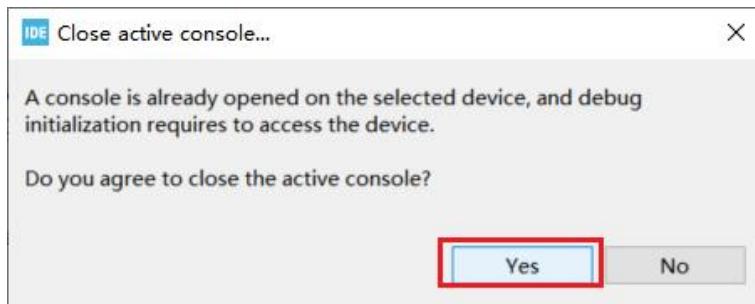
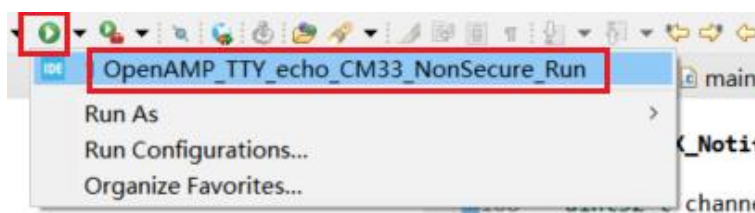
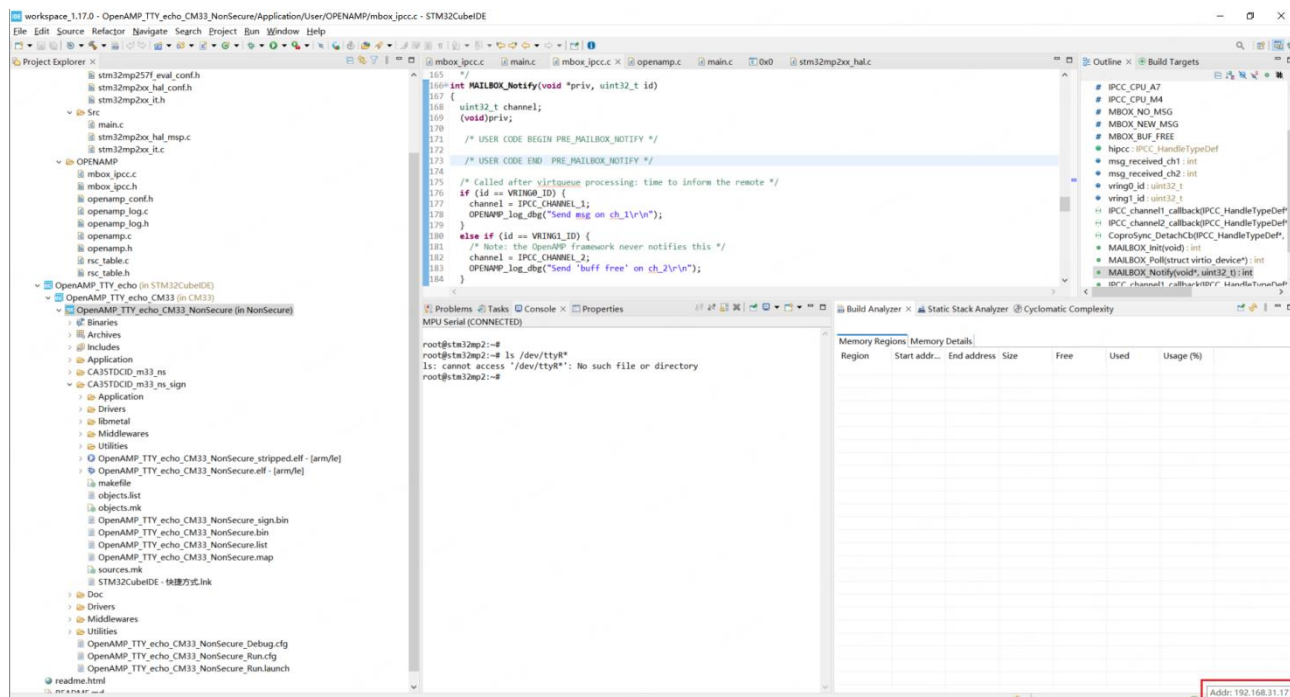


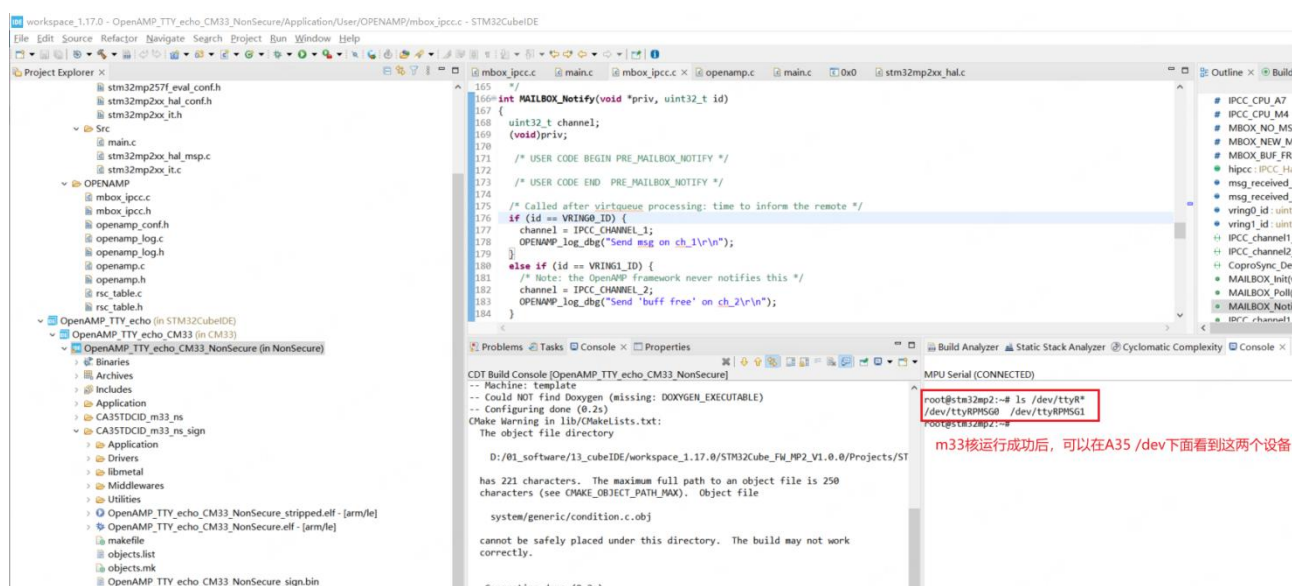






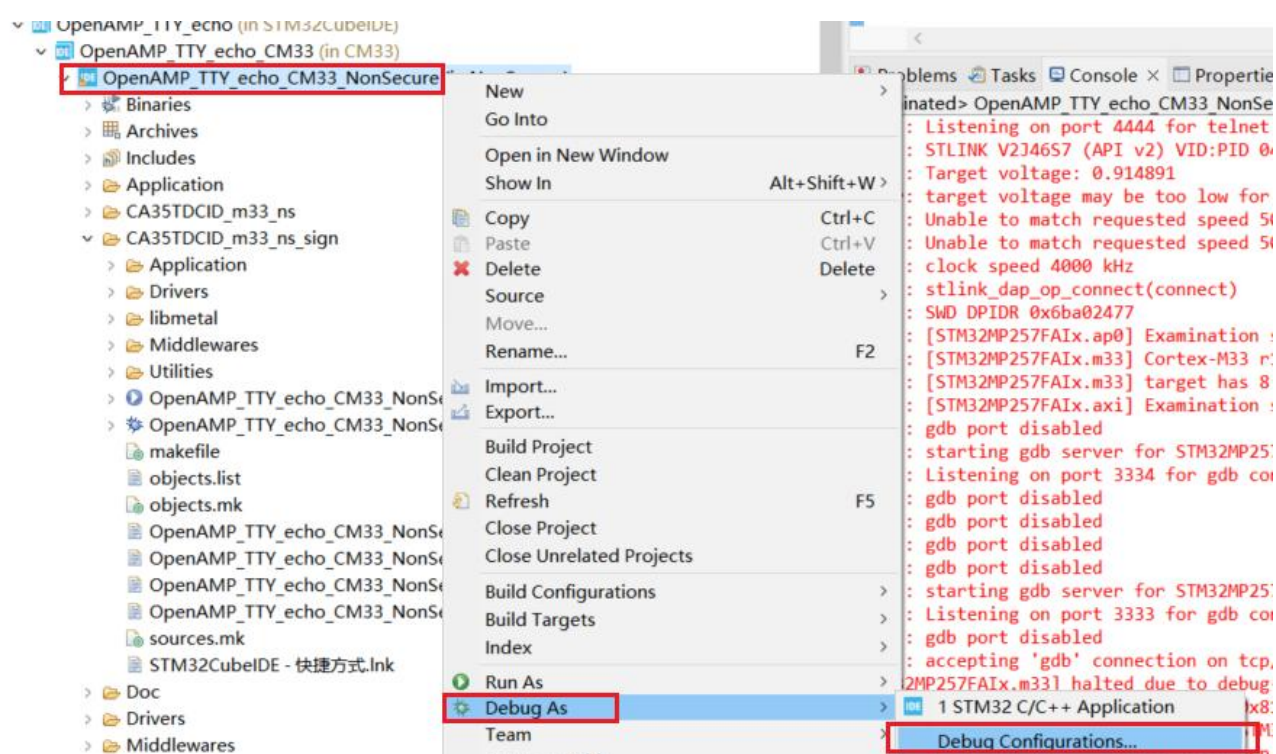
## 4.2.3 确保网络连接后，烧录并运行程序



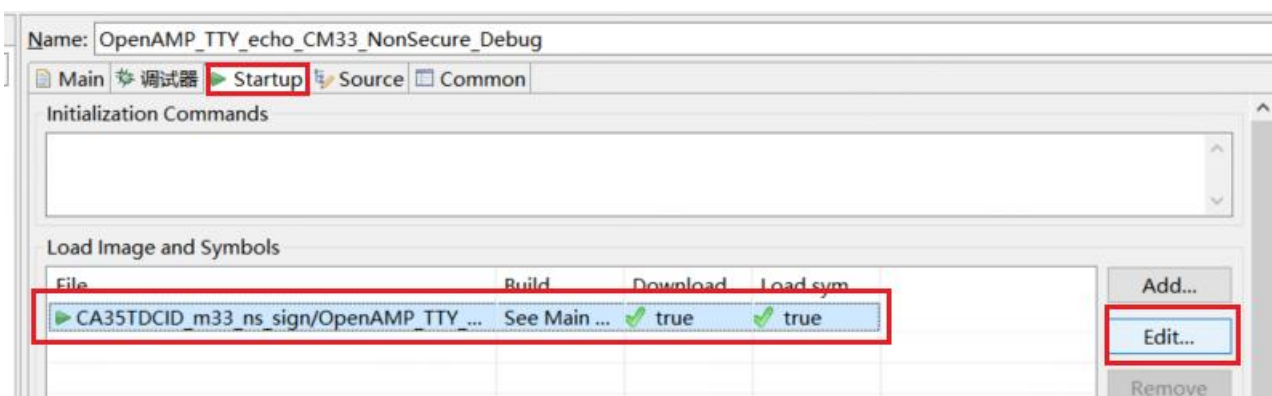
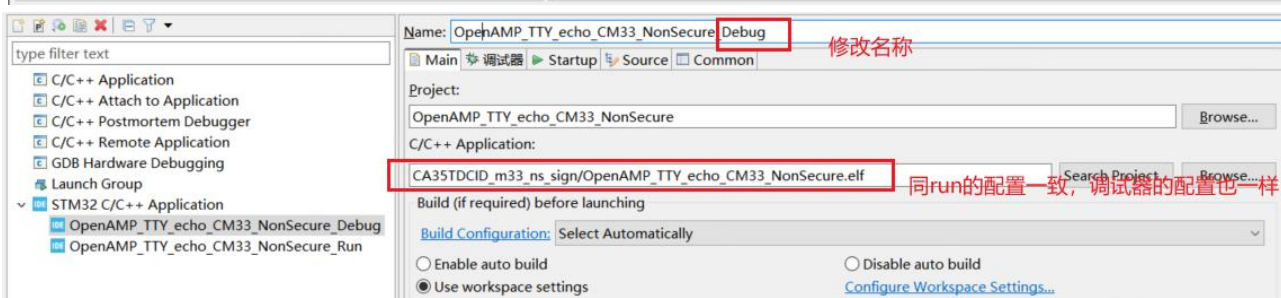
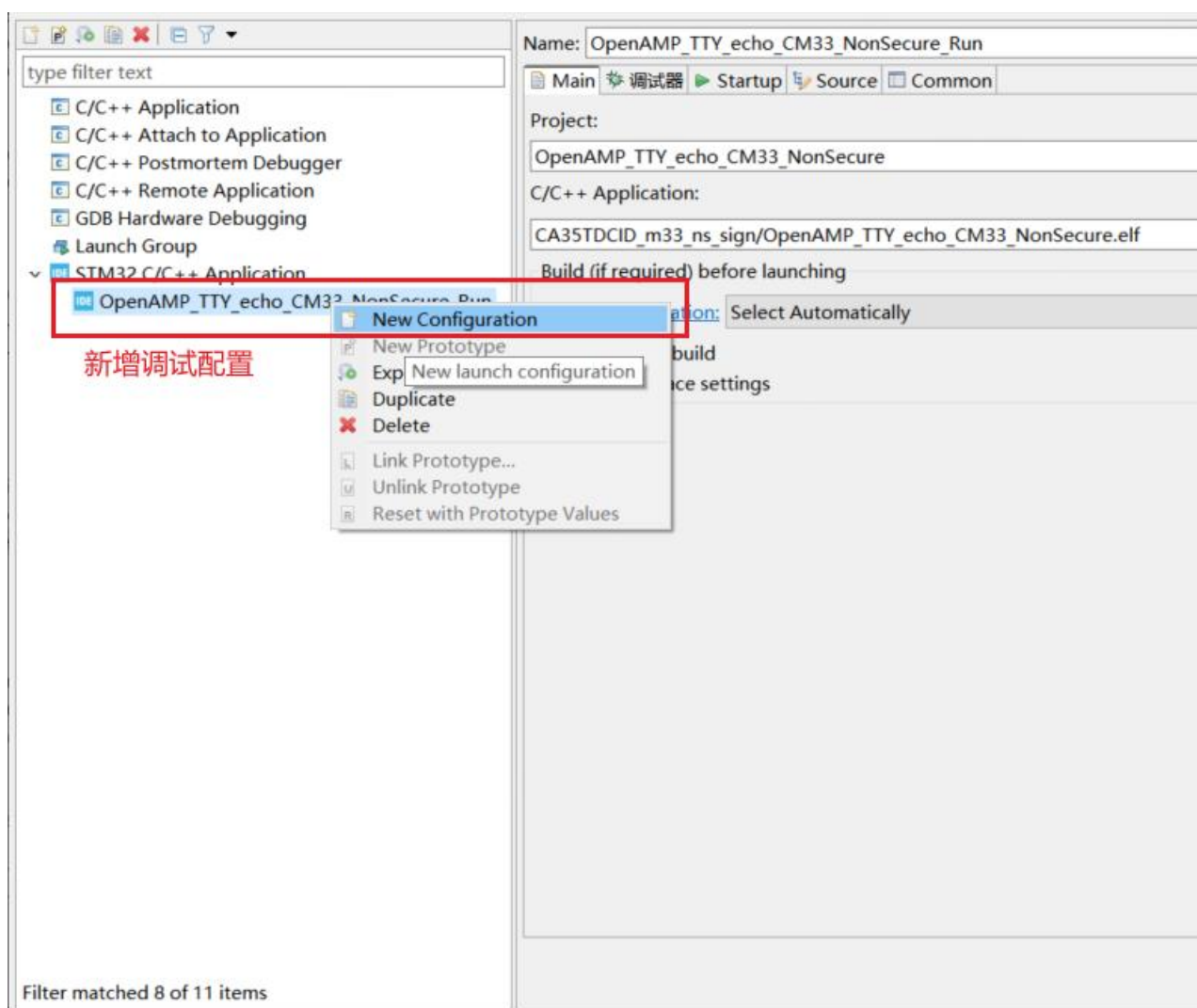


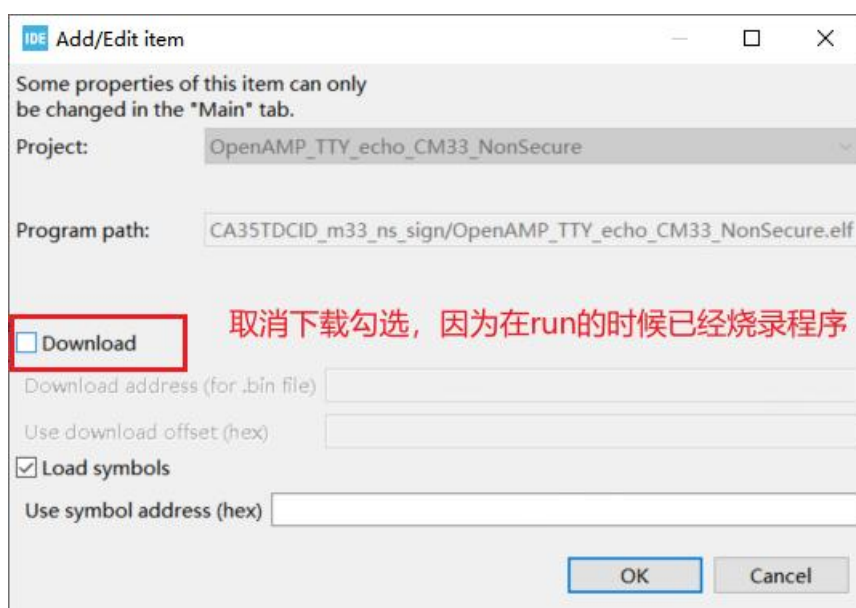
## 4.3 使用 CubeIDE、ST-Link 调试 M33 核（在运行 M33 核之后进行调试）

### 4.3.1 进行调试配置



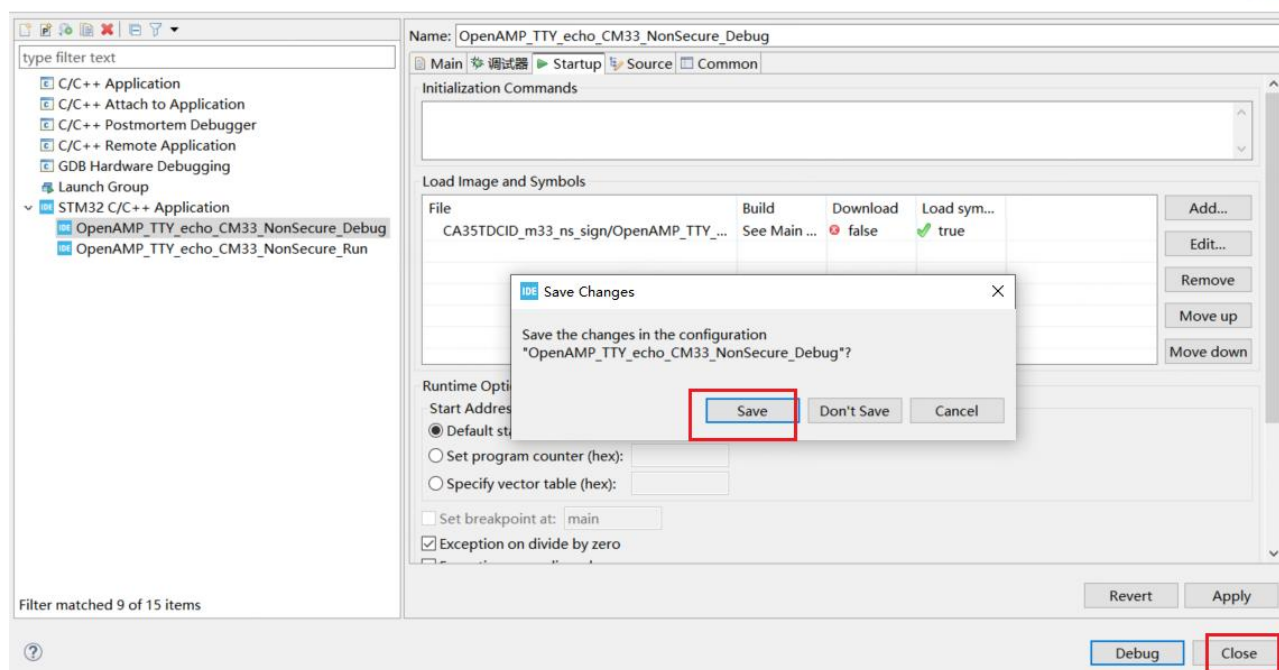






IDE Debug Configurations

Create, manage, and run configurations



### 4.3.2 开始调试程序

