
实验二 开发环境搭建和使用

RT-AK 教育套件实验手册

上海睿赛德电子科技有限公司 版权所有 @2021



WWW.RT-THREAD.ORG

Monday 29th November, 2021

目录

目录	i
1 实验目的	1
2 实验环境	2
3 实验原理	3
4 实验步骤	4
4.1 本机 AI 开发环境搭建	4
4.1.1 安装 RT-Thread-Studio	4
4.1.2 安装 Miniconda	4
4.1.3 安装 Jupyter notebook	6
4.1.4 安装 Tensorflow	7
4.1.5 Netron 软件安装	7
4.2 端侧部署环境	8
4.2.1 Studio 配置	9
4.2.1 从模板创建工程	13
4.3 编译与下载	18
4.3.1 编译	18
5 附件	23

第 1 章

实验目的

1. 建立本机 AI 开发环境
2. 搭建基于 RT-AK - K210 插件的 AI 端侧部署工具环境

第 2 章

实验环境

- 硬件环境：PC 机 Pentium 处理器双核 2GHz 以上，内存 4GB 以上
- 操作系统：Windows10 64 位及以上操作系统
- AI 模型编程环境：Python3.7 以上，Miniconda，Jupyter notebook，Tensorflow 等
- 开发软件：RT-Thread-Studio
- 烧录软件：KFlash
- 实验器材：EgdeAI 实验板

第3章

实验原理

开发版电路原理图: [RTT-Draco-Schematic-v0.2.0.1.pdf](#)

开发板模块展示:

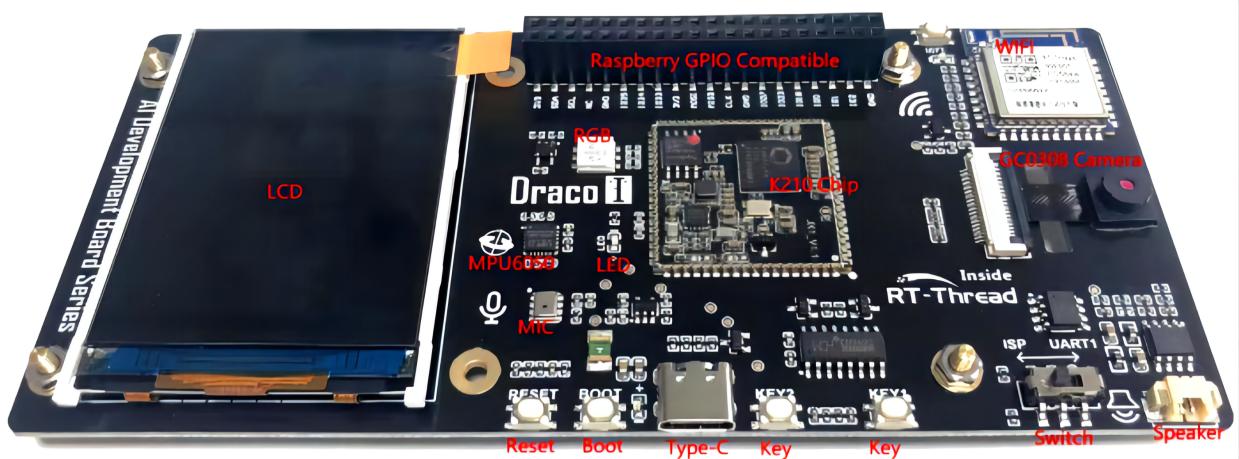


图 3.1: module

第 4 章

实验步骤

本机 AI 开发环境主要用于模型的训练、评估。端侧部署环境包括使用 RT-AK 一键部署模型到工程中，然后使用原厂工具链进行编译、链接，最后用 K-Flash 工具烧录。

4.1 本机 AI 开发环境搭建

本机 AI 开发环境包含 1)Studio、2)Python 3.7.9、3)Python 环境下的 OpenCV/Tensorflow/Numpy 等必备软件包。

4.1.1 安装 RT-Thread-Studio

下载

下载地址: <https://www.rt-thread.org/page/download.html>

选择相应的版本下载。

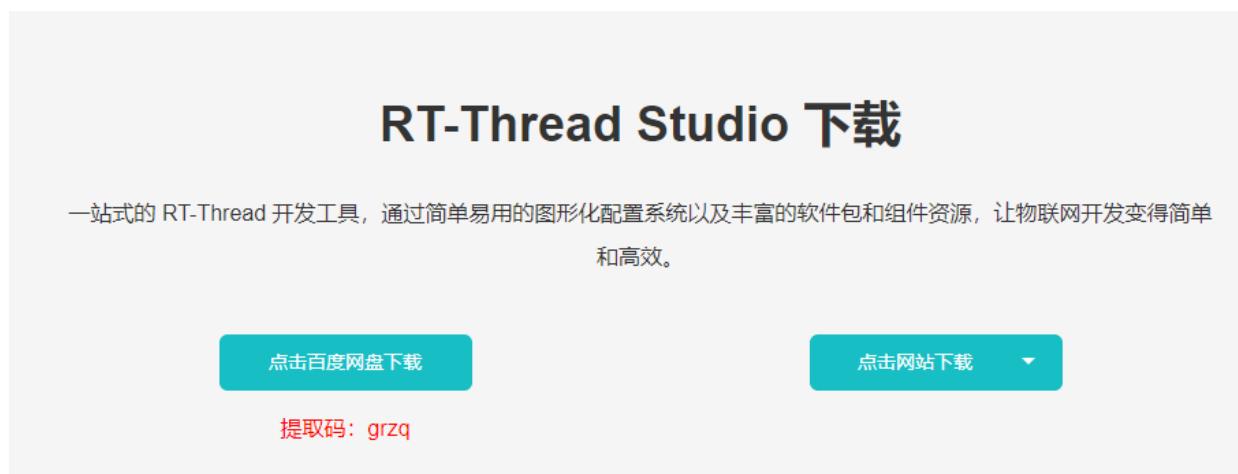


图 4.1: VSCode 下载

4.1.2 安装 Miniconda

说明：这一部分内容只是一种实现方式，如果读者熟悉其他的 AI 开发环境，请选择自己熟悉的

下载

清华源网址: <https://mirrors.tuna.tsinghua.edu.cn/anaconda/miniconda/>, 下载最新的 Miniconda 软件

File Name ↓	File Size ↓	Date ↓
Parent directory/	-	-
Miniconda3-latest-MacOSX-arm64.sh	44.9 MiB	2021-11-08 23:03
Miniconda3-py38_4.10.1-MacOSX-arm64.sh	44.9 MiB	2021-11-08 23:03
Miniconda3-py39_4.10.3-Windows-x86_64.exe	58.1 MiB	2021-07-22 00:14
Miniconda3-py39_4.10.3-Windows-x86.exe	55.3 MiB	2021-07-22 00:14

图 4.2: Miniconda 版本

下载完成后，双击源文件进行安装

安装过程中，遇到下图，请勾选

然后，安装后可以获得下图，即表明安装成功

为了进一步确认安装成功，请随意打开一个命令窗口，输入命令：‘conda’，如果输出选项，说明安装成功

```
C:\Users\rtt_ai>conda
usage: conda-script.py [-h] [-V] command ...

conda is a tool for managing and deploying applications, environments and packages.

Options:
  --help            Show this help message and exit.
  --version         Show the conda version number and exit.

Positional Arguments:
  command
    clean      Remove unused packages and caches.
    compare   Compare packages between conda environments.
    config    Modify configuration values in .condarc. This is modeled after the git config command. Writes to the user .condarc file (C:\Users\rtt_ai\.condarc) by default.
    create    Create a new conda environment from a list of specified packages.
    help     Displays a list of available conda commands and their help strings.
    info      Display information about current conda install.
    init     Initialize conda for shell interaction. [Experimental]
    install  Installs a list of packages into a specified conda environment.
    list     List linked packages in a conda environment.
    package  Low-level conda package utility. (EXPERIMENTAL)
    remove   Remove a list of packages from a specified conda environment.
    uninstall Alias for conda remove.
    run      Run an executable in a conda environment. [Experimental]
    search   Search for packages and display associated information. The input is a MatchSpec, a query language for conda packages. See examples below.
    update   Updates conda packages to the latest compatible version.
    upgrade  Alias for conda update.

Optional Arguments:
  -h, --help            Show this help message and exit.
  -V, --version         Show the conda version number and exit.

conda commands available from other packages:
  env

C:\Users\rtt_ai>
```

图 4.3: 验证 conda 命令

4.1.3 安装 Jupyter notebook

在 conda 环境中执行: `conda install jupyter notebook`, 就会安装 jupyter notebook。如下图所示:

```

Anaconda Prompt (Miniconda3) - conda install jupyter notebook

pyparsing      pkgs/main/noarch::pyparsing-3.0.4-pyhd3eb1b0_0
pyqt          pkgs/main/win-64::pyqt-5.9.2-py39hd77b12b_6
pyrsistent    pkgs/main/win-64::pyrsistent-0.18.0-py39h196d8e1_0
python-dateutil pkgs/main/noarch::python-dateutil-2.8.2-pyhd3eb1b0_0
pywinpty      pkgs/main/win-64::pywinpty-0.5.7-py39haa95532_0
pyzmq          pkgs/main/win-64::pyzmq-22.2.1-py39hd77b12b_1
qt              pkgs/main/win-64::qt-5.9.7-vc14h73c81de_0
qtconsole     pkgs/main/noarch::qtconsole-5.1.1-pyhd3eb1b0_0
qtpy          pkgs/main/noarch::qtpy-1.10.0-pyhd3eb1b0_0
send2trash    pkgs/main/noarch::send2trash-1.8.0-pyhd3eb1b0_1
sip              pkgs/main/win-64::sip-4.19.13-py39hd77b12b_0
terminado     pkgs/main/win-64::terminado-0.9.4-py39haa95532_0
testpath       pkgs/main/noarch::testpath-0.5.0-pyhd3eb1b0_0
tornado        pkgs/main/win-64::tornado-6.1-py39h2bbff1b_0
traitlets     pkgs/main/noarch::traitlets-5.1.1-pyhd3eb1b0_0
wcwidth        pkgs/main/noarch::wcwidth-0.2.5-pyhd3eb1b0_0
webencodings   pkgs/main/win-64::webencodings-0.5.1-py39haa95532_1
widgetsnbextension pkgs/main/win-64::widgetsnbextension-3.5.1-py39haa95532_0
winpty          pkgs/main/win-64::winpty-0.4.3-4
zipp            pkgs/main/noarch::zipp-3.6.0-pyhd3eb1b0_0
zlib            pkgs/main/win-64::zlib-1.2.11-h62dc97_4

The following packages will be UPDATED:

ca-certificates      2021.7.5-haa95532_1 --> 2021.10.26-haa95532_2
certifi               2021.5.30-py39haa95532_0 --> 2021.10.8-py39haa95532_0
openssl               1.1.1k-h2bbff1b_0 --> 1.1.11-h2bbff1b_0

Proceed ([y]/n)? a

```

图 4.4: 安装 Jupyter notebook

注意: 遇到类似 Proceed ([y]/n), 请选择 y, 然后回车即可

安装完成后, 即可运行 `jupyter notebook`, 此时的 jupyter notebook 就安装完成了, 用于编写 AI 模型

```

Anaconda Prompt (Miniconda3) - jupyter notebook

(base) C:\Users\rtt_ai>jupyter notebook
[I 10:57:25.054 NotebookApp] [nb_conda_kernels] enabled, 1 kernels found
[I 10:57:25.078 NotebookApp] Writing notebook server cookie secret to C:\Users\rtt_ai\AppData\Roaming\jupyter\runtime\notebook_cookie_secret
[I 10:57:25.632 NotebookApp] Serving notebooks from local directory: C:\Users\rtt_ai
[I 10:57:25.632 NotebookApp] Jupyter Notebook 6.4.6 is running at:
[I 10:57:25.632 NotebookApp] http://localhost:8888/?token=357c9e843e0c28d30accb6e09f37fc4d74066428bb29e506
[I 10:57:25.632 NotebookApp] or http://127.0.0.1:8888/?token=357c9e843e0c28d30accb6e09f37fc4d74066428bb29e506
[I 10:57:25.632 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).

To access the notebook, open this file in a browser:
  file:///C:/Users/rtt_ai/AppData/Roaming/jupyter/runtime/nbserver-10832-open.html
Or copy and paste one of these URLs:
  http://localhost:8888/?token=357c9e843e0c28d30accb6e09f37fc4d74066428bb29e506
  or http://127.0.0.1:8888/?token=357c9e843e0c28d30accb6e09f37fc4d74066428bb29e506

```

图 4.5: Jupyter 测试

为了测试是否成功安装 Jupyter, 可以按照教程 3 (mnist 实验来测试)

4.1.4 安装 Tensorflow

在 conda 环境中，在 base 环境下：使用命令 `conda install tensorflow==2.5.0`, 注意需要使用 `conda` 命令安装，才能够自动搜索依赖。

注意：这一部分要注意，需要确认自己电脑是否带有 NVIDIA 的 GPU，如果没有 NVIDIA 的 GPU，请使用 `tensorflow` 的 `cpu` 版本。

```
C:\Users\rtt_ai>conda install tensorflow==2.5.0
get metadata (current_repodata.json).done
ent: failed with initial frozen solve. Retrying with flexible solve.
ent: failed with repodata from current_repodata.json, will retry with next repodata source.
get metadata (repodata.json): done
ent: \
```

图 4.6: 安装 `Tensorflow`

这一部分，可能由于网络原因导致安装失败，请多次尝试，知道得到下图，说明安装成功。

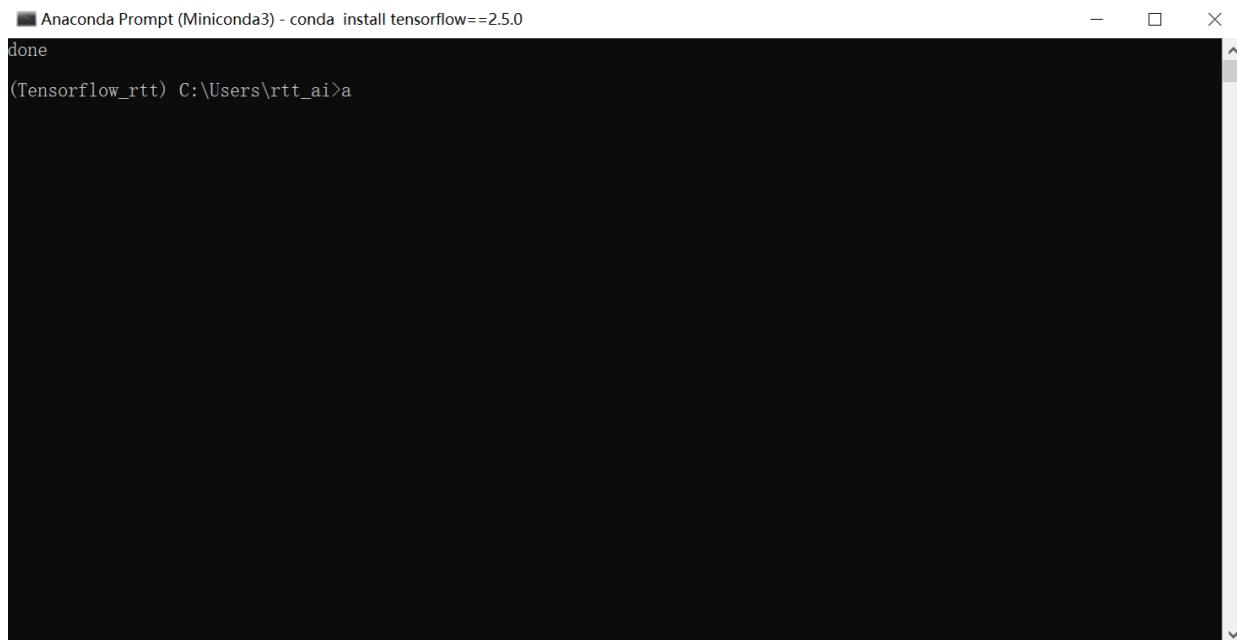


图 4.7: `Tensorflow` 安装完成

至此，AI 模型训练所需要的软件环境我们就配置好了。

注意：在运行不同的例程时，可能需要安装不同的依赖库，比如 `opencv`, `matplotlib` 等，此时需要在命令行窗口执行 `pip` 命令安装即可。例如 `pip install opencv-python`、`pip install matplotlib` 等

4.1.5 Netron 软件安装

使用 Netron 软件查看人工智能模型的内部网络结构。

[下载地址](#)

双击安装即可。

验证：

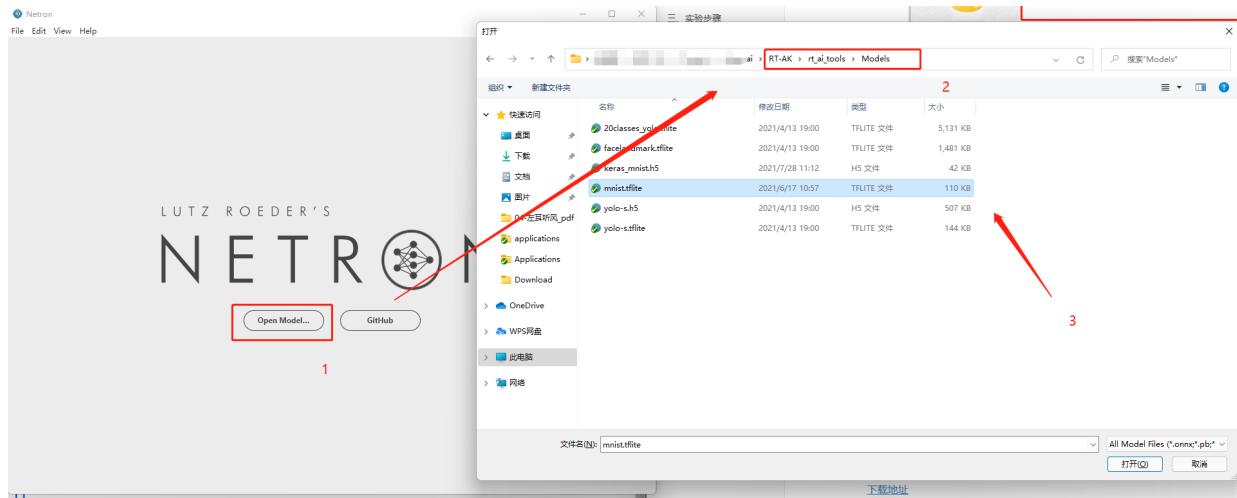


图 4.8: 如何使用 Netron

结果：

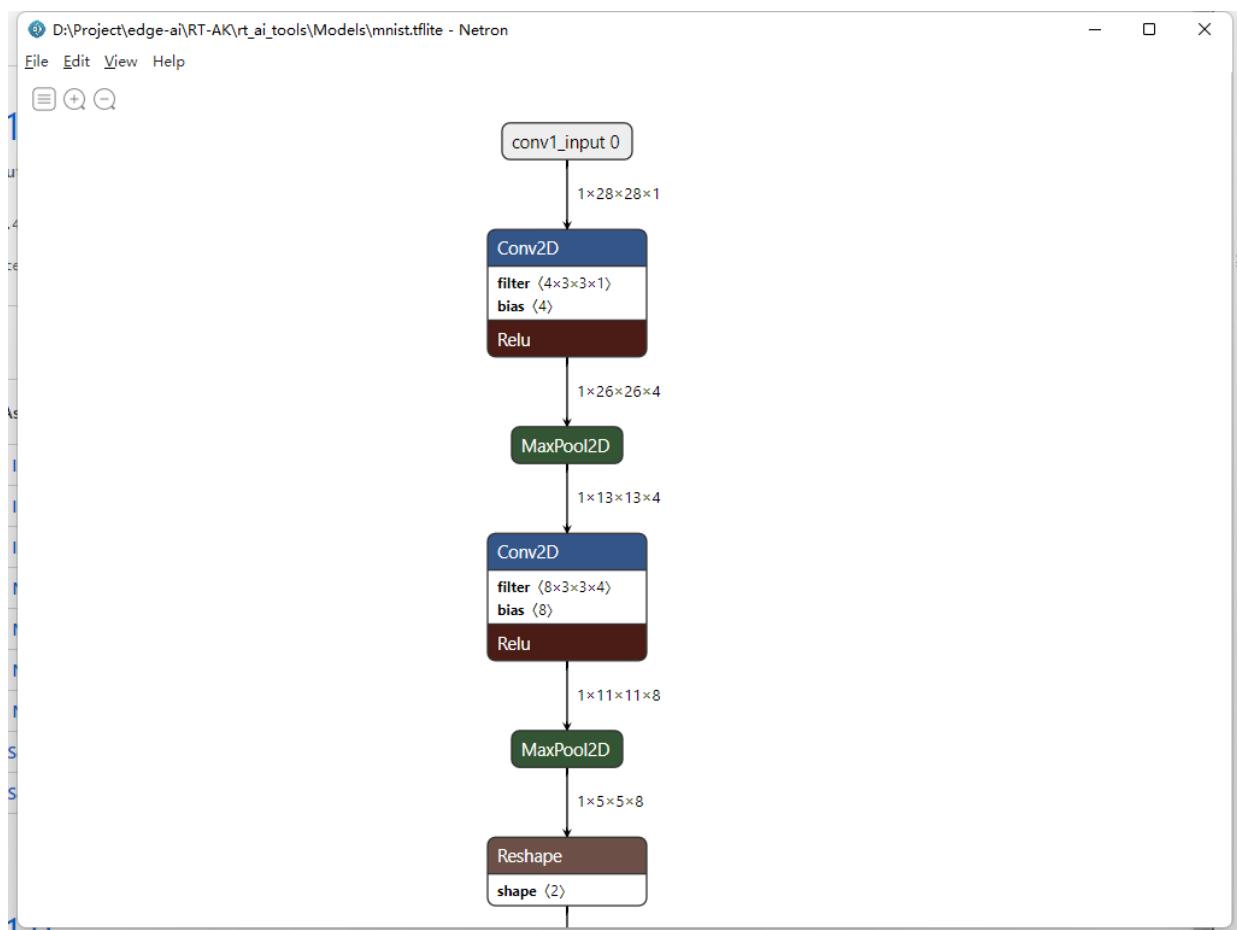


图 4.9: 模型网络结构

4.2 端侧部署环境

端侧部署分为工程准备、编译链接、系统烧录三步。工程准备基于 RT-AK 的 K210 插件，详见实验一。

下载并安装 RT-Thread Studio 完成后，下面实现一个简单的工程创建的流程：

4.2.1 Studio 配置

此章节为更新 Studio 到最新版本，并配置工程模板教程，若用户当前 Studio 环境中没有 K210-RT-DRACO sdk 的模板工程，可参考此章节进行模板导入。若此前已经进行过导入，则可过此节直接进行工程创建。

1. 打开桌面上的 RT-Thread-Studio。



图 4.10: Studio_icon.png

2. 若 Studio 弹框提示 可用的更新，一直点击下一步。

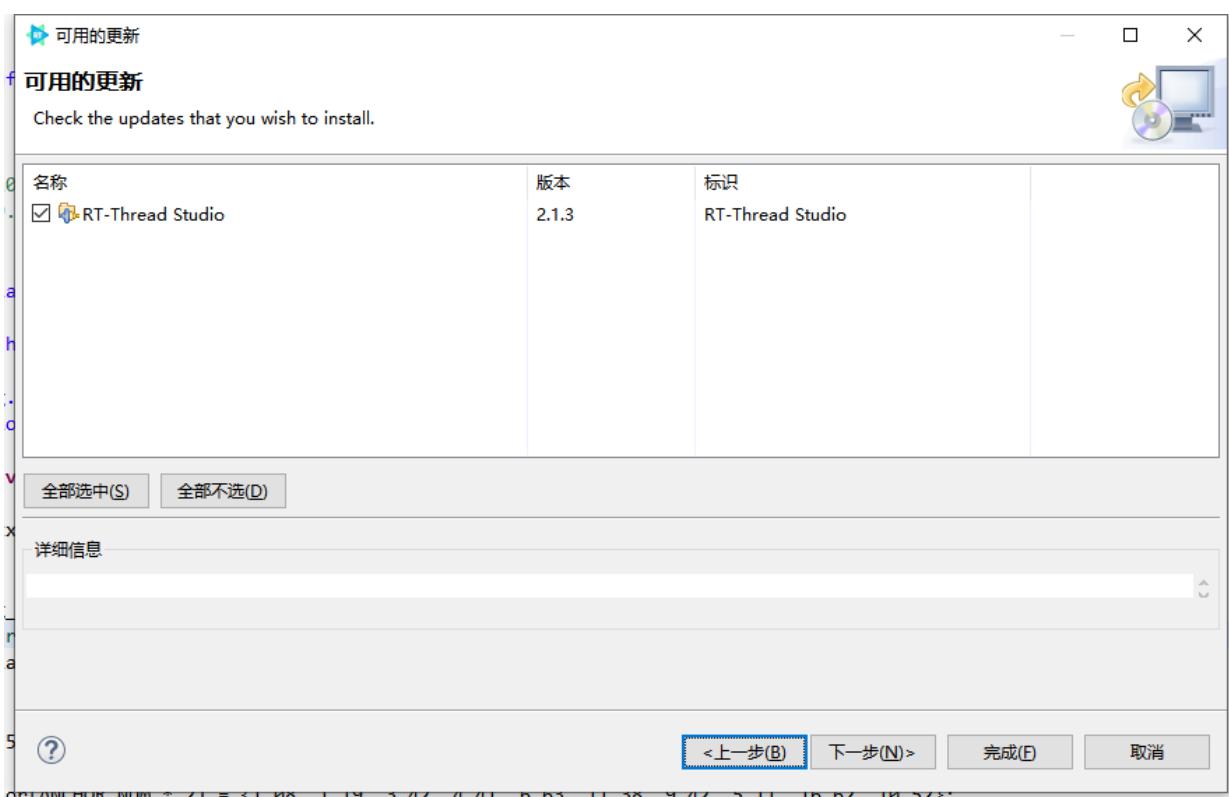


图 4.11: update-1

直到勾选 接收许可协议，然后点击完成：

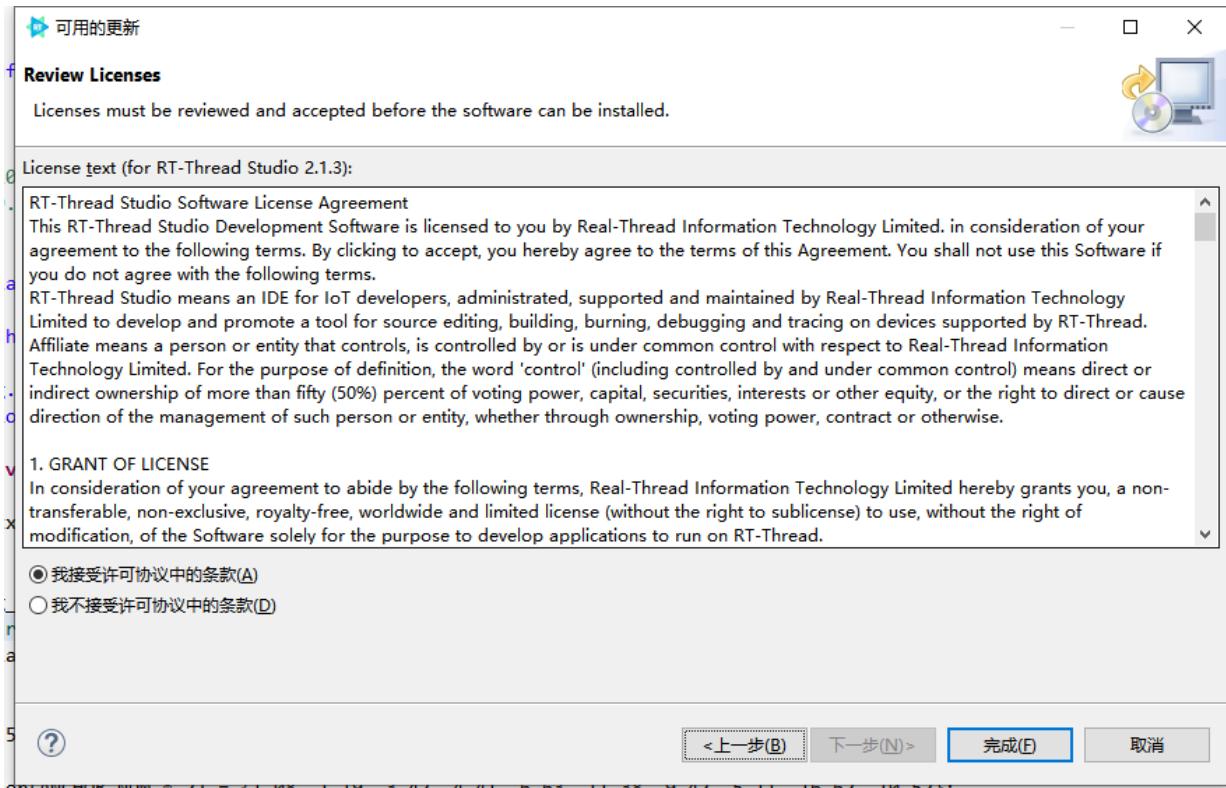


图 4.12: update-2

3. 若安装过程中提示如下图所示，点击 install anyway:

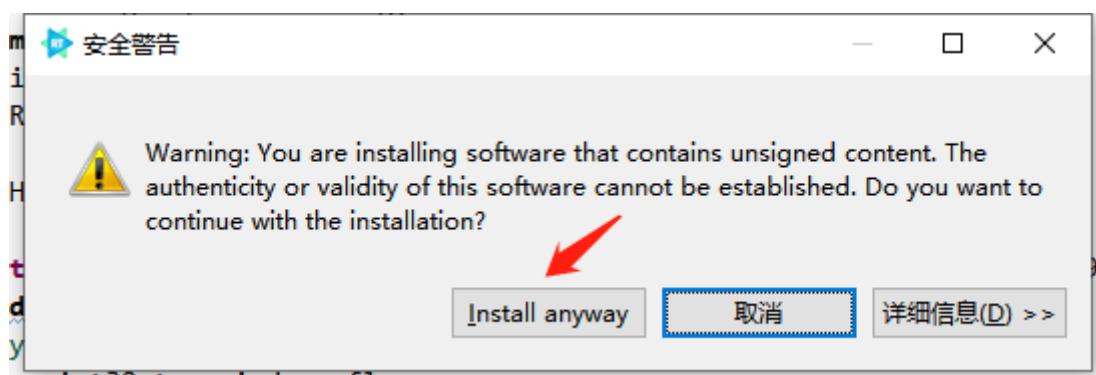


图 4.13: install-anyway

可在右下角查看安装正在进行:



图 4.14: updating

4. 安装完成后提示 Restart Now点击进行重启:

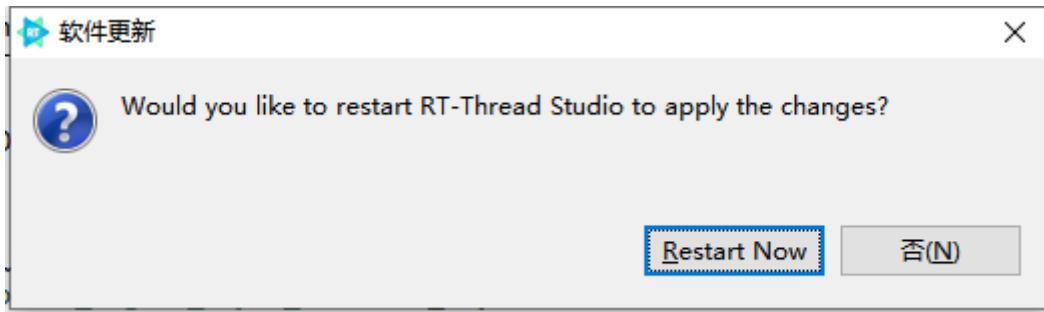


图 4.15: update-restart

5. 点击 Studio 界面菜单栏下方的 SDK Manager->勾选K210-RT-DRACO:

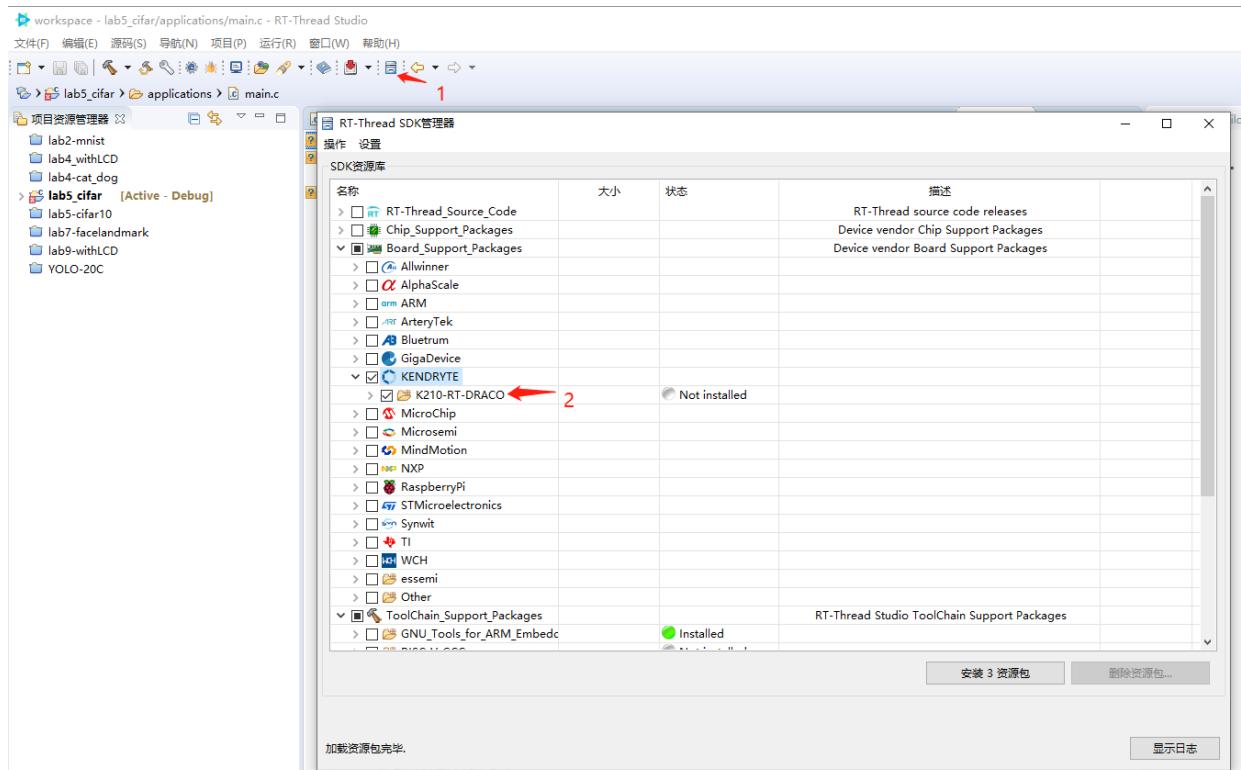


图 4.16: sdk-manager

然后继续勾选编译器和调试器如下图:

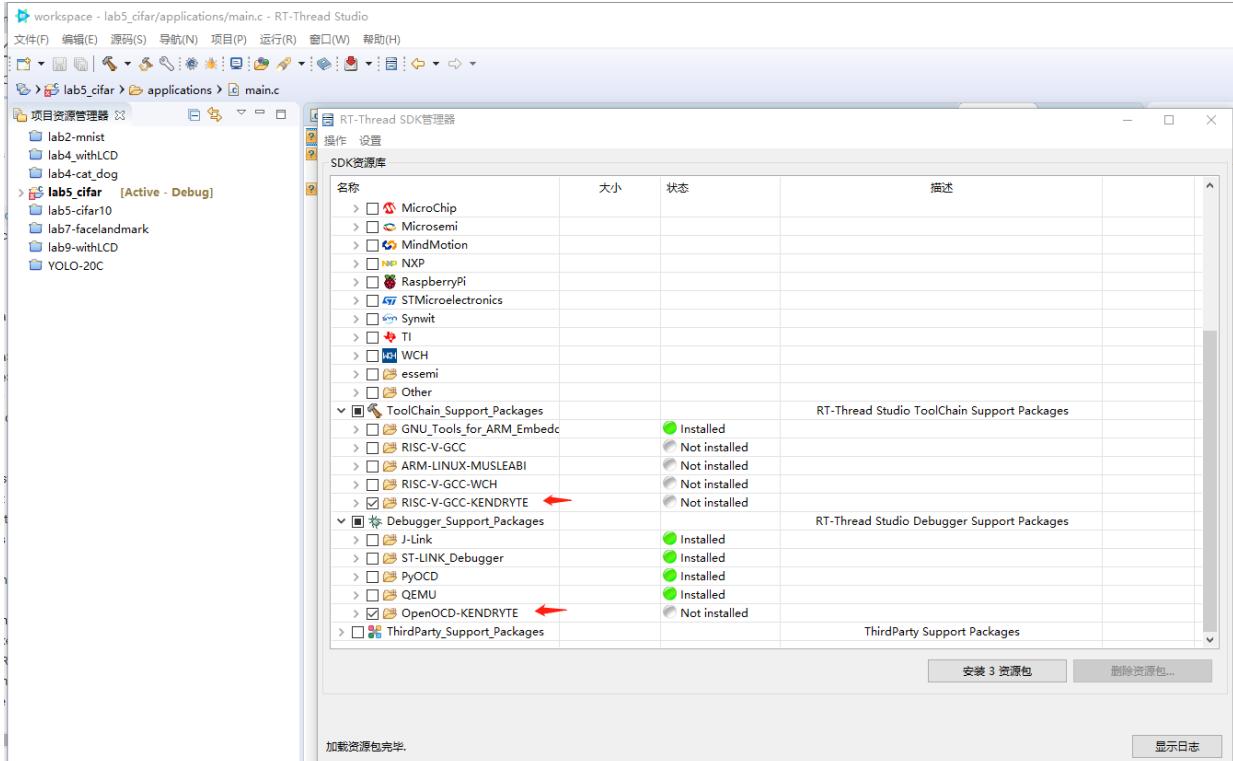


图 4.17: SDK-GCC

最后点击 安装资源包 等待安装完成:

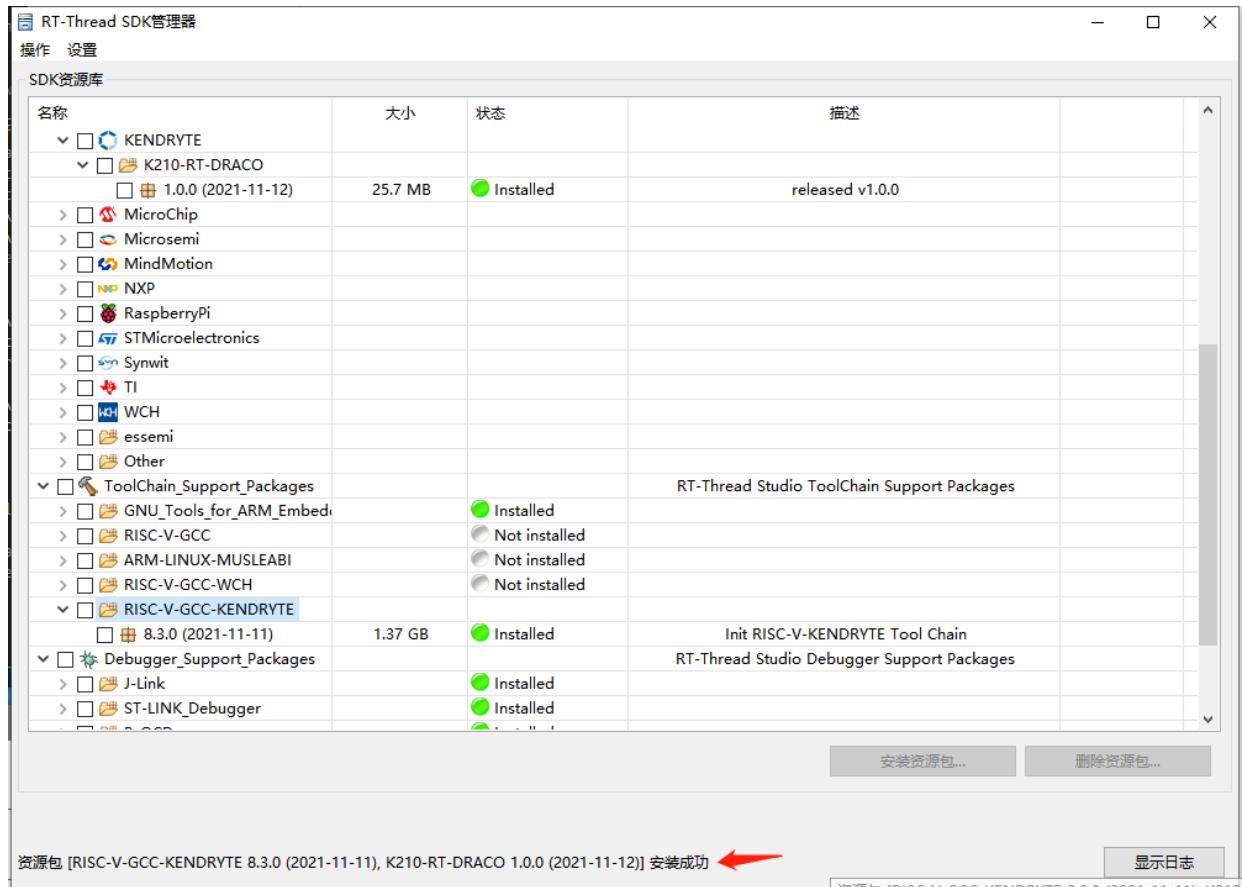


图 4.18: SDK-Complete

4.2.1 从模板创建工程

1. 打开桌面上的 RT-Thread-Studio。



图 4.19: Studio_icon.png

2. 打开 Studio 界面后，点击左上角 文件->新建->RT-Thread 项目。

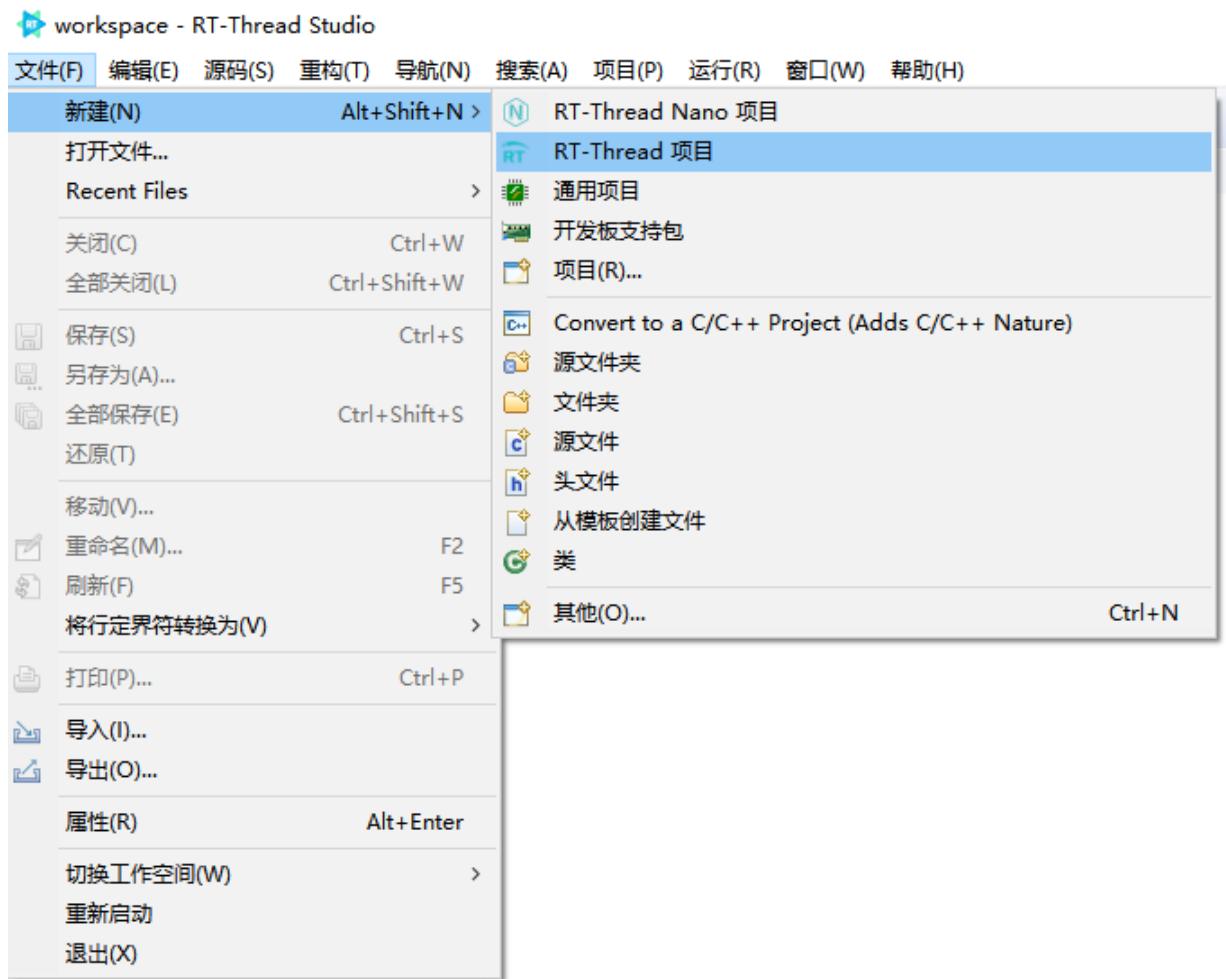


图 4.20: 1-create_prj.png

3. 在新建项目界面选择 基于开发板，输入 Project name 和要创建工程的位置路径，然后点击 完成。

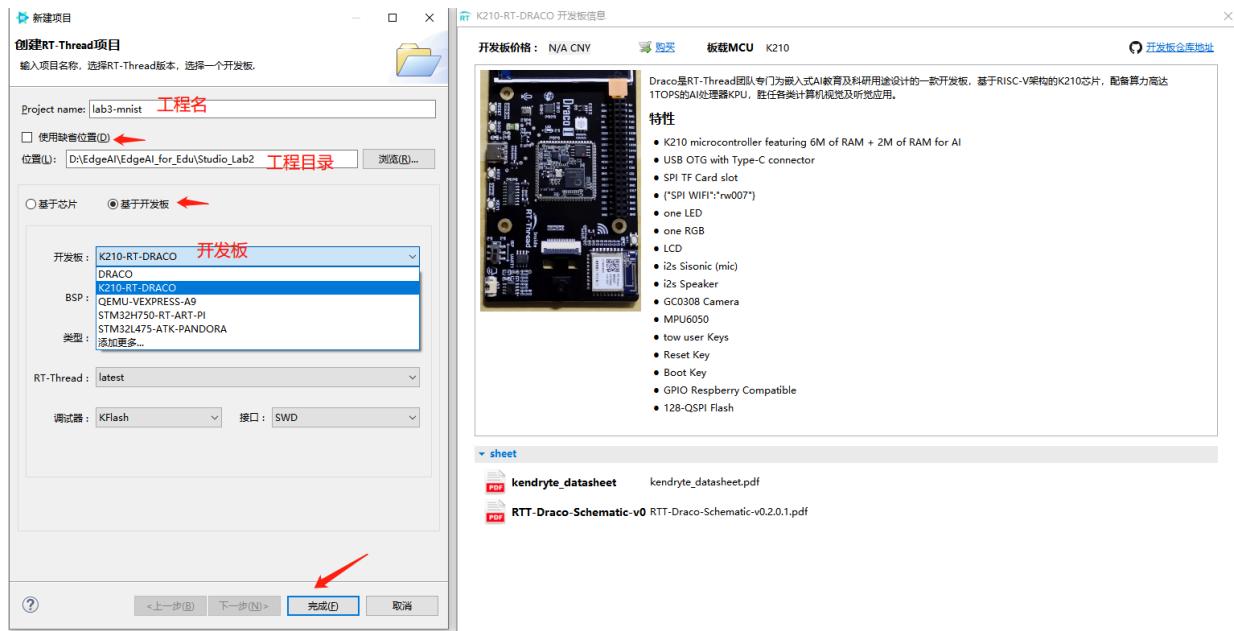


图 4.21: 2-lab2-create.png

4. 每次创建完新的工程或添加新的代码文件或软件包后，进行刷新工程可将文件添加到工程目录中。操作如下所示，先右键点击项目，点击更新软件包，然后再次鼠标点击项目，点击刷新：

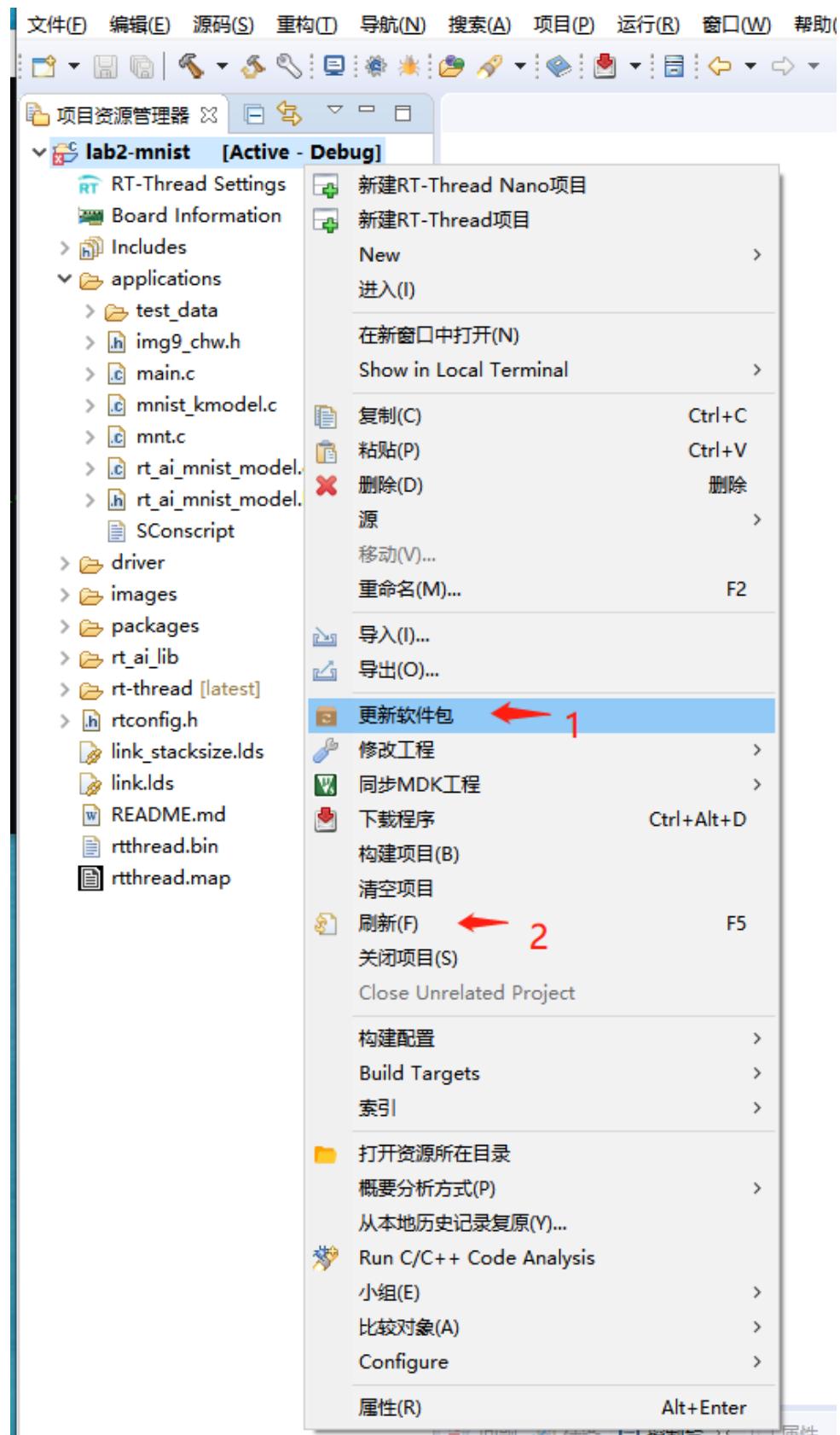


图 4.22: 4-fresh.png

5. 若代码中注释出现乱码现象，可以点击界面左上角 窗口->首选项。如下图所示：

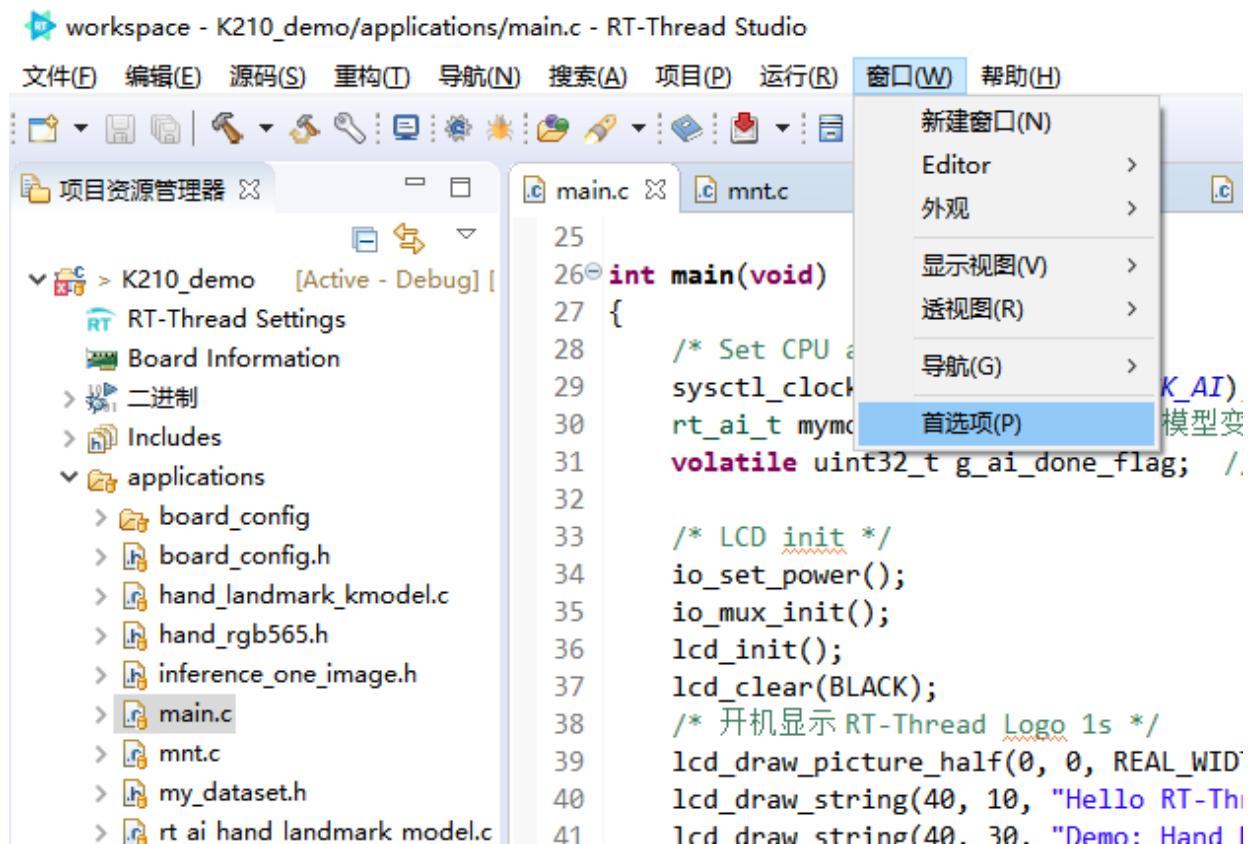


图 4.23: 5.png

点击 工作空间，设置 文本文件编码，编码设置为 UTF-8即可。

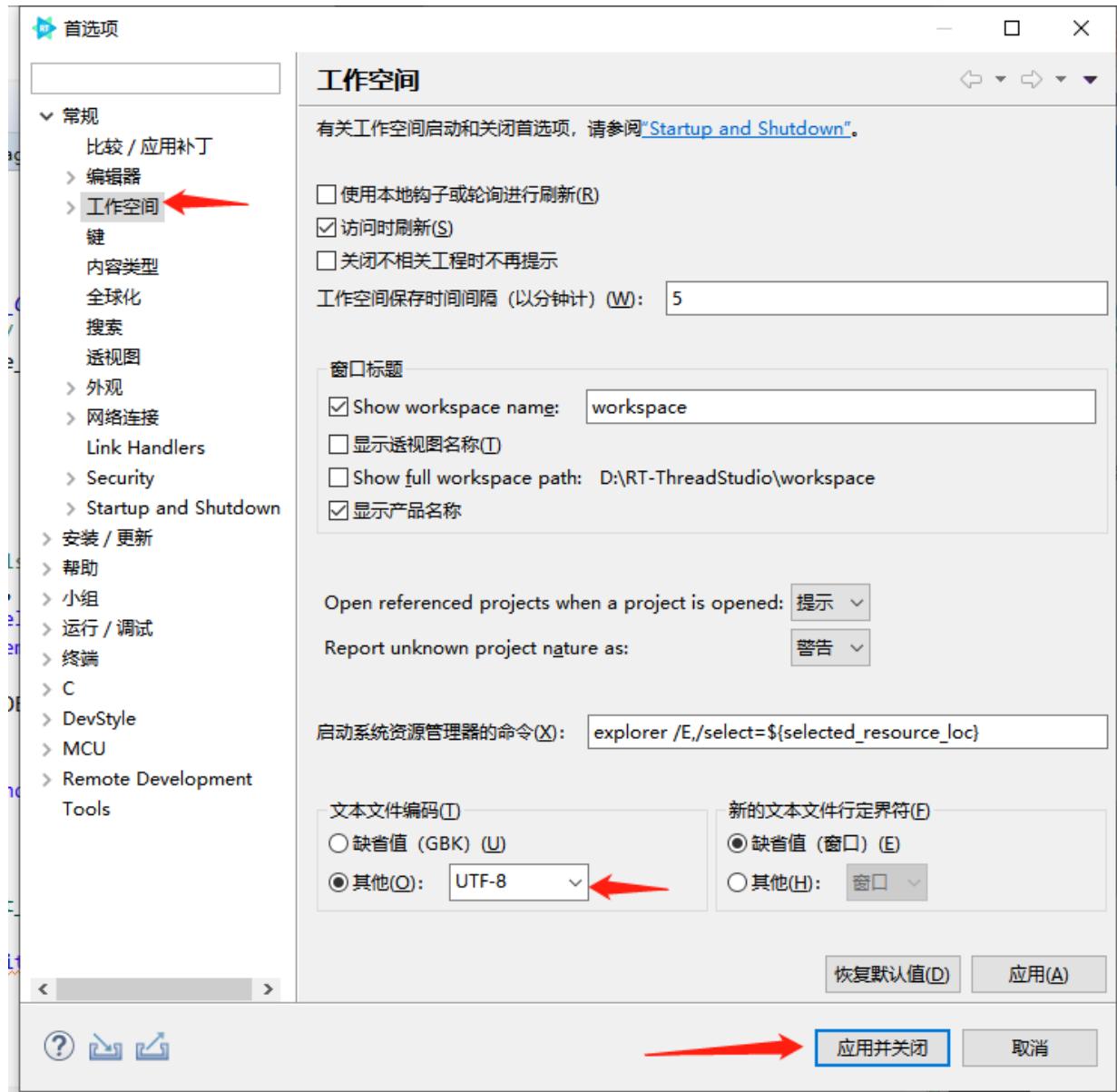


图 4.24: 6.png

6. 工程创建完成，IDE 界面基本功能按钮如下图：

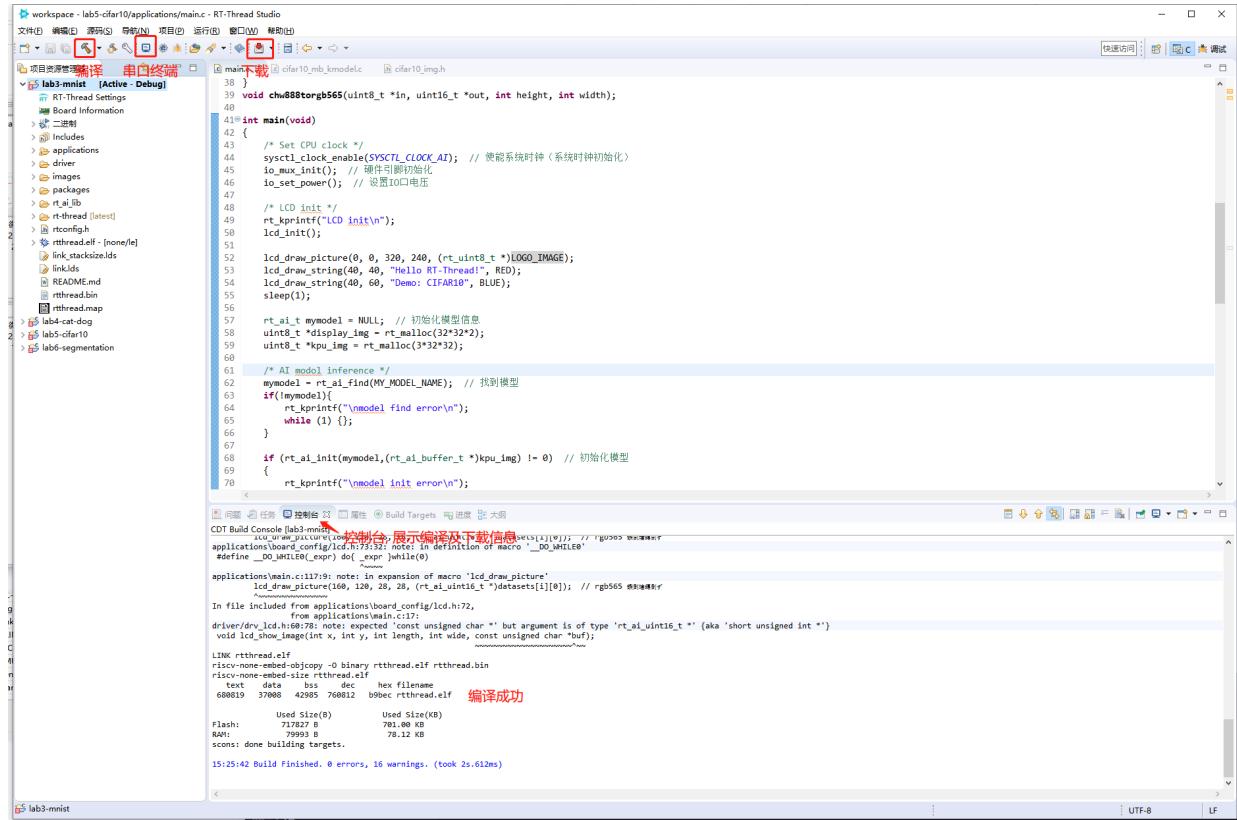


图 4.25: image-20211105135453338

4.3 编译与下载

4.3.1 编译

1. 参考 4.2 节中，界面展示的说明。点击上方编译图标进行编译：

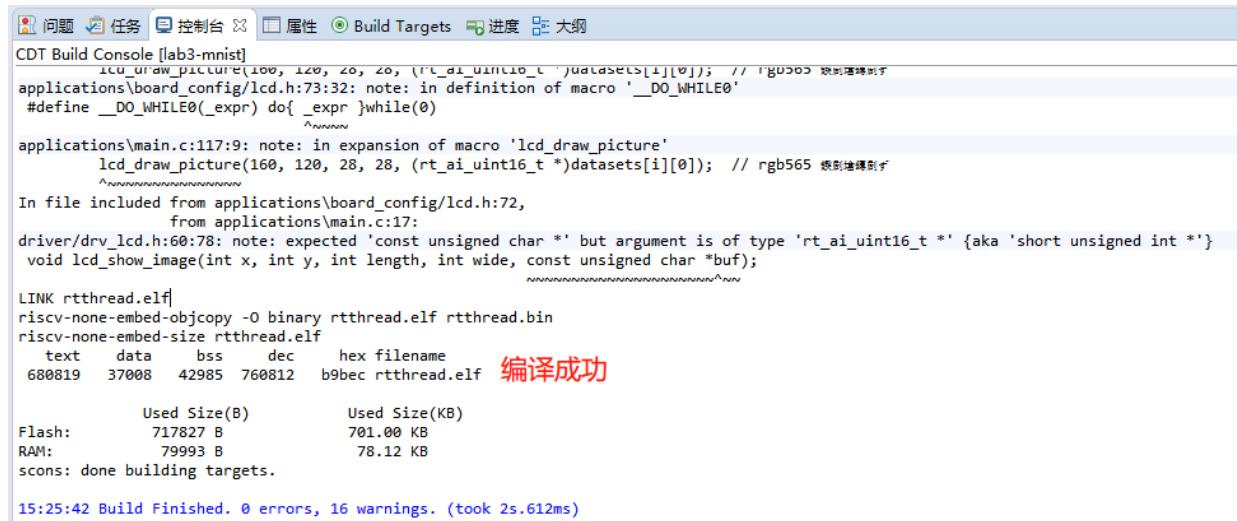


图 4.26: compile_ok

2. 点击下载，第一次下载通常会弹出端口选择框，进行选择即可：

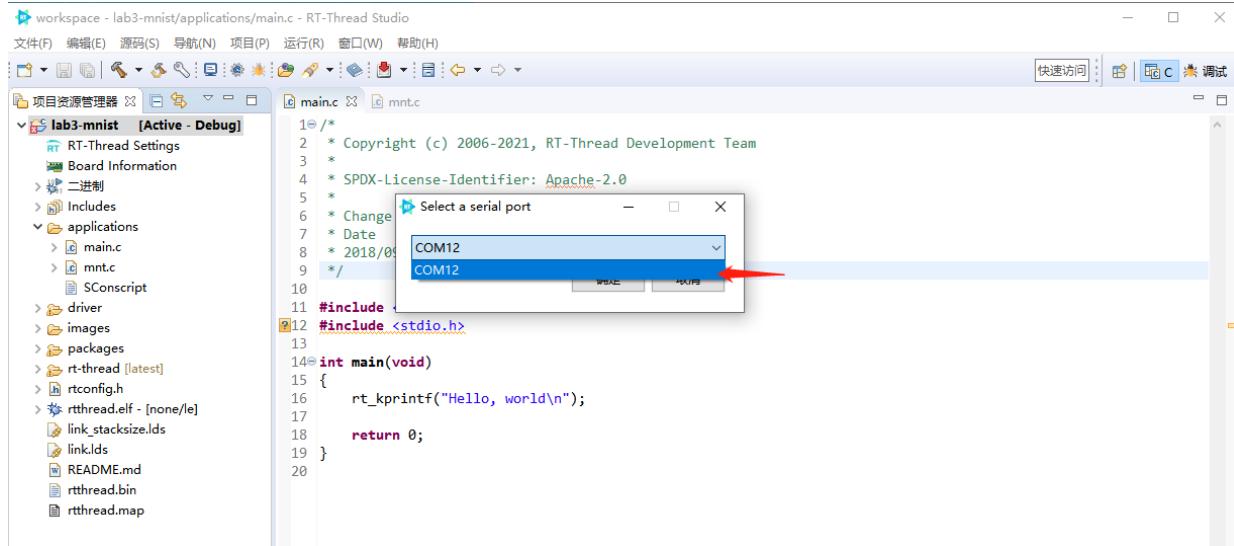


图 4.27: download_com

下载成功如下所示：

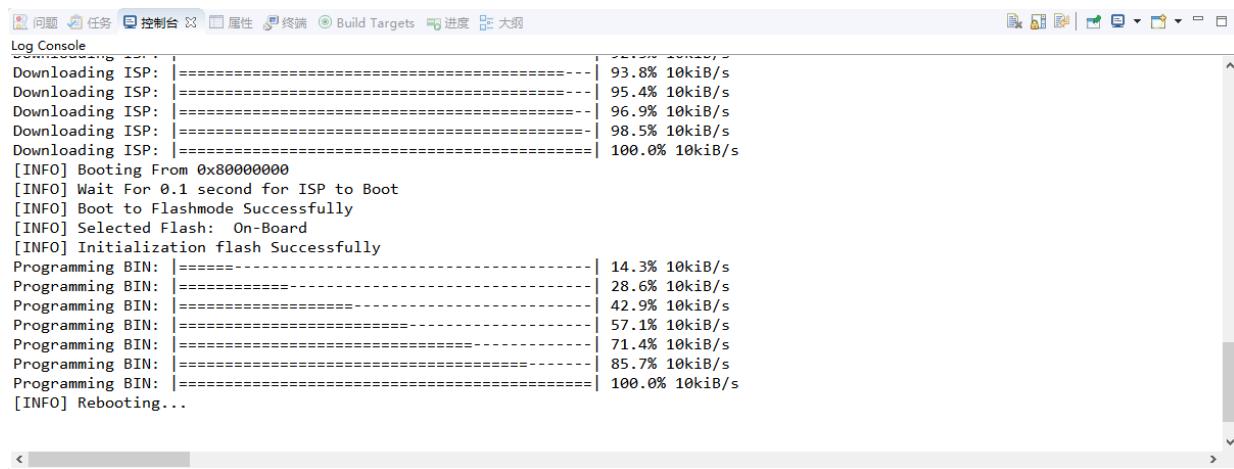


图 4.28: download_ok

注意：若点击下载时控制台已经开始下载时再次弹出串口选择框，则需要点击取消，或者可能出现串口占用。

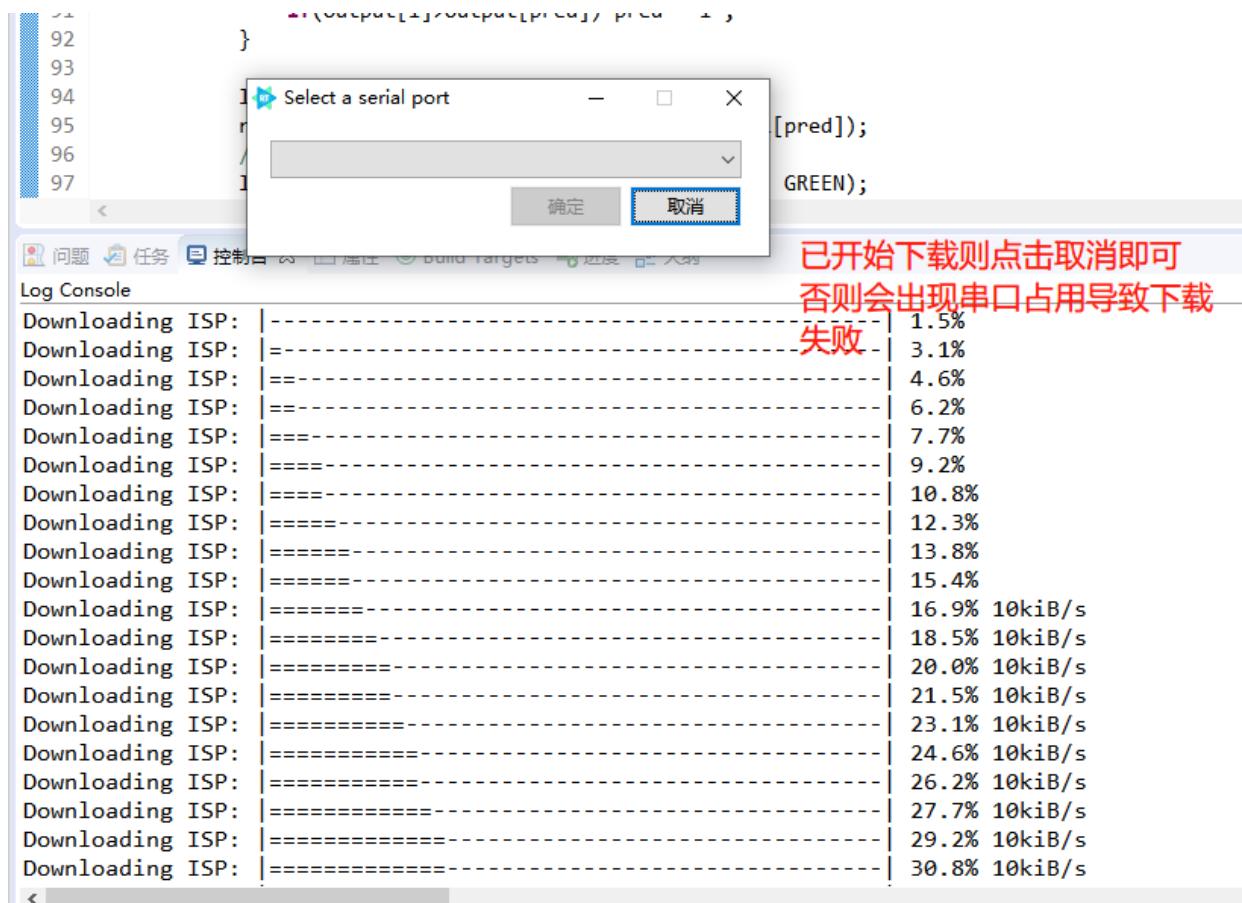


图 4.29: note_com

3. 下载完成后打开串口，可查看交互信息：

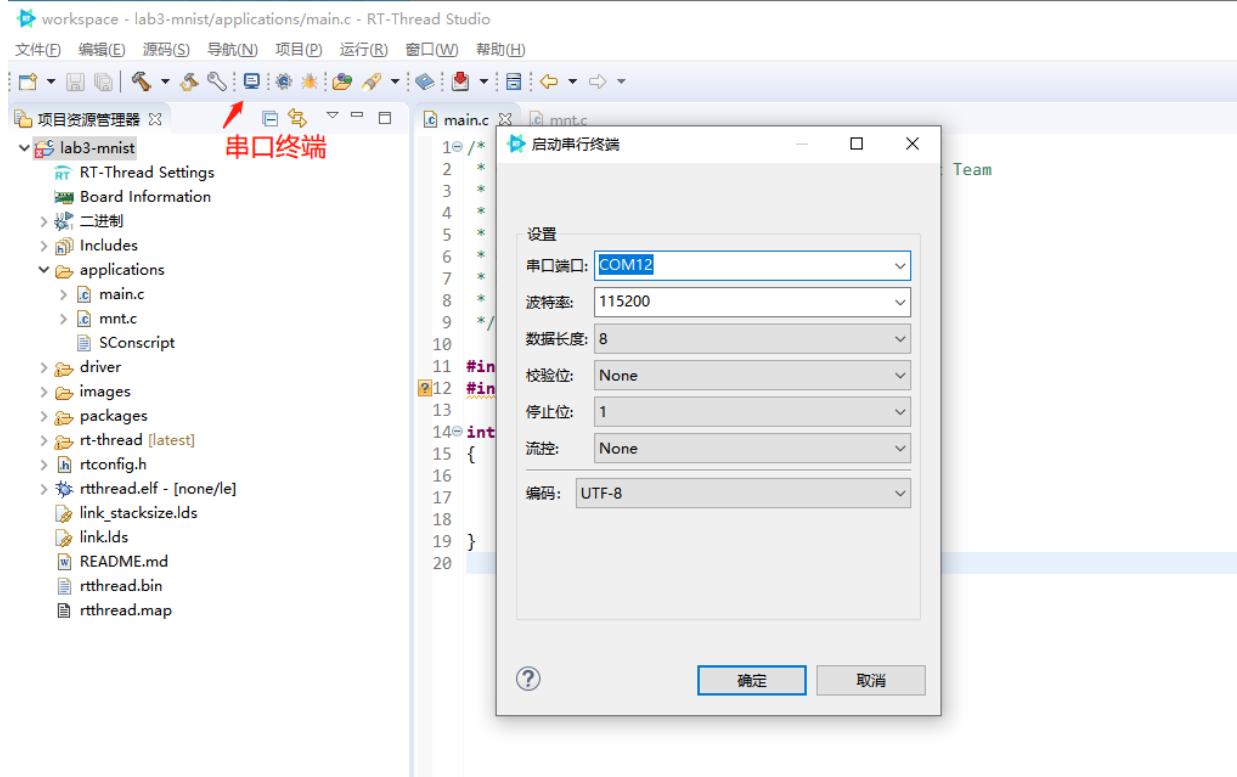


图 4.30: com_open

4. 串口打开成功，打印出 RT-Thread LOGO 和 msh 命令行：

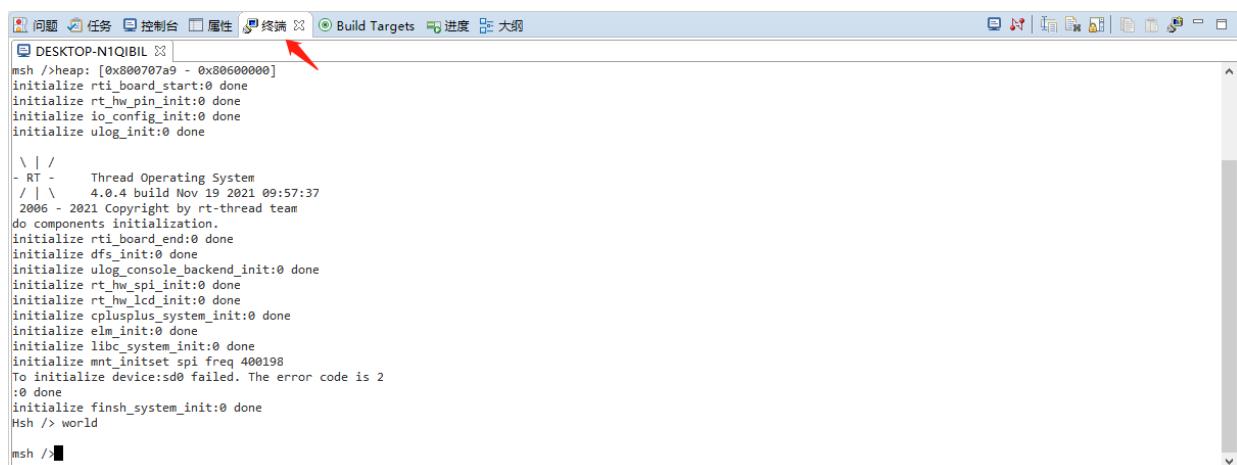
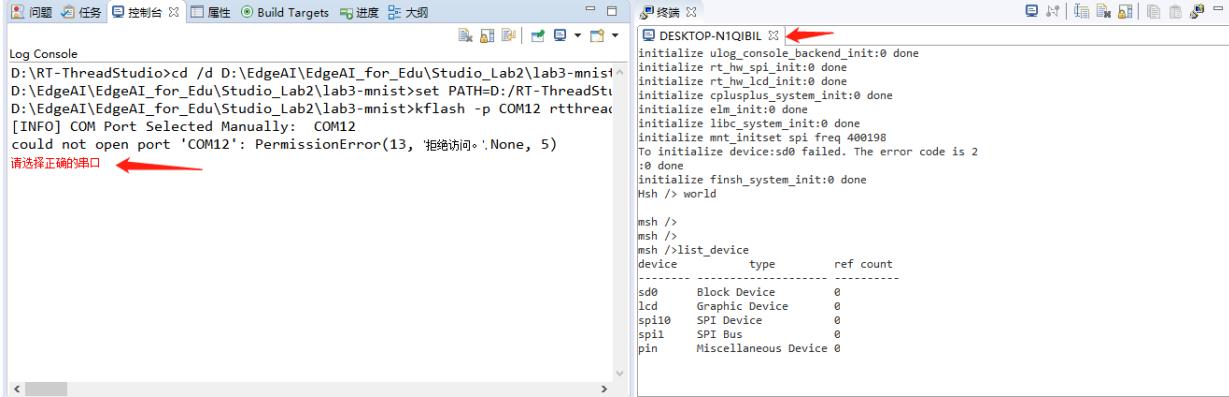


图 4.31: com_ok

注意：若下载时出现 请选择正确串口，可能是由于串口被占用导致，需检查并关闭占用串口的窗口。



```

Log Console
D:\RT-ThreadStudio>cd /d D:\EdgeAI\EdgeAI_for_Edu\Studio_Lab2\lab3-mnist
D:\EdgeAI\EdgeAI_for_Edu\Studio_Lab2\lab3-mnist>set PATH=D:/RT-ThreadStudio
D:\EdgeAI\EdgeAI_for_Edu\Studio_Lab2\lab3-mnist>kflash -p COM12 rtthread
[INFO] COM Port Selected Manually: COM12
could not open port 'COM12': PermissionError(13, '拒绝访问', 5)
请选择正确的串口

```

The screenshot shows the RT-Thread Studio interface with a terminal window. The terminal output indicates a permission error ('PermissionError(13, '拒绝访问', 5)') when trying to open port 'COM12'. A red arrow points to the error message, and another red arrow points to the prompt '请选择正确的串口' (Please select the correct serial port).

图 4.32: note download

其它相关工具参考链接: 交叉编译工具链, 下载地址 <https://github.com/xpack-dev-tools/riscv-none-embed-gcc-xpack/releases/tag/v8.3.0-1.2>

或者: <https://github.com/kendryte/K210-Micropython-OpenMV/tree/master/toolchain>

固件下载工具 下载地址 <https://github.com/kendryte/kendryte-flash-windows/releases/download/v0.4.1/K-Flash.zip>

K-Flash 下载工具使用截图

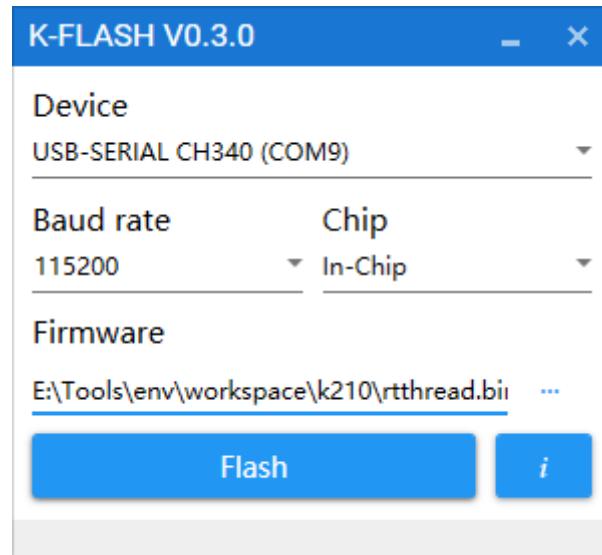


图 4.33: K-Flash

第 5 章

附件

更多 RT-AK 的内容请查阅文档, (查看 markdown 文档推荐使用 Typora 软件):

1. 每个文件夹下面的 `README.md` , 尤其是 `RT-AK\rt_ai_tools\platforms\plugin_k210` 插件的 `README.md` 需要查阅
2. `RT-AK` 提供的嵌入式 api 的文档位于: `rt_ai_lib/readme.md`
3. 其他文档 `RT-AK/documents`