

## 2. 4Linux 系统下搭建开发环境

### 一、编译和烧录 K210 程序需要的工具

本次我们是在 Ubuntu 18.04 系统下，通过 VSCode 编辑器搭建 K210 的开发环境，以下是我们需要用到的工具及简介。

1. Cmake: CMake 是众多 Make 工具的一种，具有跨平台，跨编译器等特性，是一种非常实用的工具。
2. Toolchain: 交叉编译器 Toolchain 是用于调试开发板的工具和相关库组合起来的集合，也称为工具链。一般内附有程式编译器 (compiler)，如 GCC，可以进行程式编译、或跨平台编译工作。
3. VSCode: VSCode 是 Microsoft 出品的一款开源代码编辑器，支持多种插件，可扩展性强，具有代码高亮，自动补全等功能。
4. K210-SDK: K210 官方出的 SDK，是开发 K210 的基础，里面自带有芯片的各种驱动。
5. flash: K210 芯片烧录程序的工具。

### 一、安装 CMake

#### 1. 验证是否已经安装了 CMake

由于 CMake 应用广泛，可能系统已经安装过 CMake，如果已经安装过，并且 CMake 版本大于 3.0，则不需要重新安装，可以跳过安装 CMake 的步骤。

打开 Ubuntu 的终端，输入以下命令，出现以下界面则表示没有安装 CMake。

```
cmake -version
```

```
gengyue@Ubuntu0:~$ cmake -version  
Command 'cmake' not found, but can be installed with:  
sudo apt install cmake
```

这里以 Ubuntu 18.04 位系统安装 CMake 为例。

## 2. 下载和安装 Cmake 和 make

输入以下命令安装

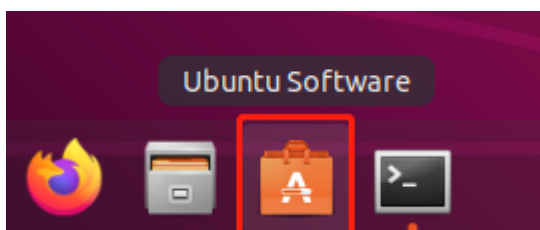
```
sudo apt-get install cmake -y
```

```
sudo apt-get install make -y
```

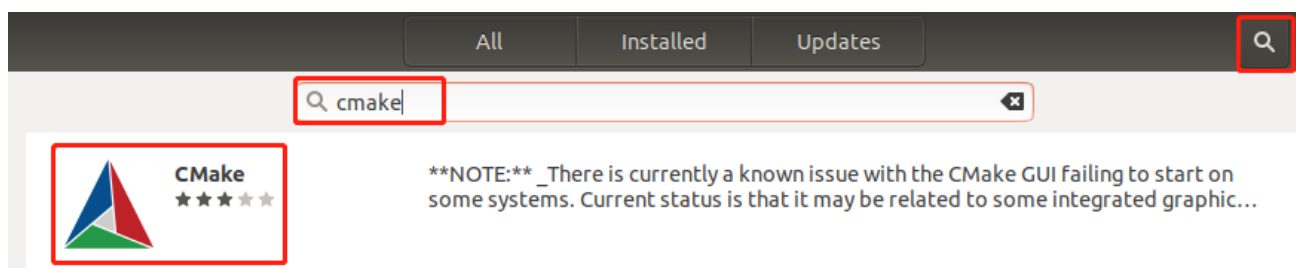
等待下载安装完成即可。

由于网络问题，可能会出现下载比较慢的情况，也可以通过 Ubuntu18.04 自带的软件包管理器来搜索下载。

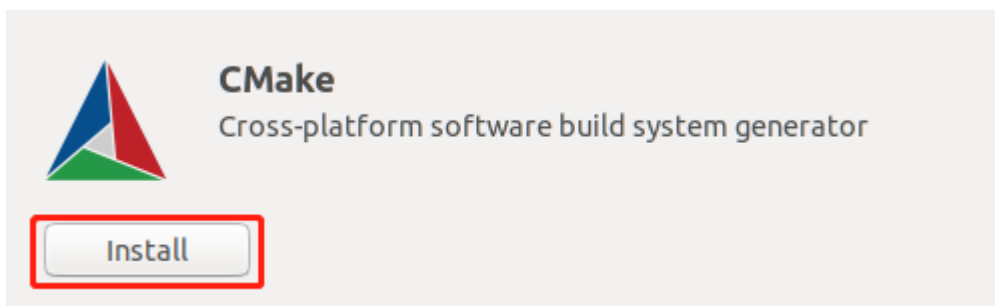
## 3. 通过 Ubuntu 自带的软件管理器来下载（如果第二步已经安装好了 CMake 则忽略此步骤）。



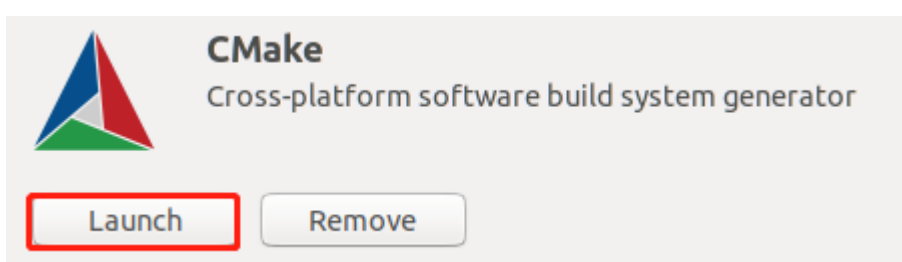
搜索 cmake



点击 Install，开始安装



安装完成。点击 Launch 运行 CMake。同时在 Ubuntu 的应用里也有 CMake 的图标。



#### 4. 检查验证 CMake

打开终端，输入 `cmake -version`，可以看到自己安装的 CMake 版本号则表示安装成功。

```
gengyue@Ubuntu0:~$ cmake -version
cmake version 3.17.2

CMake suite maintained and supported by Kitware (kitware.com/cmake).
gengyue@Ubuntu0:~$
```

## 二、安装交叉编译器 Toolchain

### 1. 下载勘智版本的 toolchain

勘智开发者资源下载网站: <https://kendryte.com/downloads/>

点击下载 RISC-V 64bit toolchain for Kendryte K210\_ubuntu\_amd64

## Toolchain

文档名称	版本	修改时间	下载
Kendryte OpenOCD for win32	V0.1.3	2018-09-13	
Kendryte OpenOCD for Ubuntu x86_64	V0.1.3	2018-09-13	
RISC-V 64bit toolchain for Kendryte K210_win32	V8.2.0	2019-03-15	
RISC-V 64bit toolchain for Kendryte K210_ubuntu_amd64	V8.2.0	2019-03-15	
RISC-V 64bit toolchain for Kendryte K210_osx_mojave	V8.2.0	2019-03-26	

## 2. 解压 toolchain 文件

把下载到的 kendryte-toolchain-ubuntu-amd64-8.2.0-20190213.tar.gz 文件放在 Downloads 目录下，所以先进入 Downloads 目录再解压：

```
cd Downloads
```

```
tar zxvf kendryte-toolchain-ubuntu-amd64-8.2.0-20190213.tar.gz
```

```
gengyue@Ubuntu0:~$ cd Downloads/
gengyue@Ubuntu0:~/Downloads$ ls
kendryte-toolchain-ubuntu-amd64-8.2.0-20190213.tar.gz
gengyue@Ubuntu0:~/Downloads$ tar zxvf kendryte-toolchain-ubuntu-amd64-8.2.0-20190213.tar.gz
kendryte-toolchain/
kendryte-toolchain/bin/
kendryte-toolchain/bin/libgmp.so.10
kendryte-toolchain/bin/libisl.so.19
```

解压后得到 kendryte-toolchain 文件夹。

```
gengyue@Ubuntu0:~/Downloads$ ls
kendryte-toolchain  kendryte-toolchain-ubuntu-amd64-8.2.0-20190213.tar.gz
gengyue@Ubuntu0:~/Downloads$
```

## 3. 移动 kendryte-toolchain 文件夹所有内容到/opt 目录下。

```
sudo mv kendryte-toolchain /opt
```

正确输入用户密码后按回车确认。

```
gengyue@Ubuntu0:~/Downloads$ sudo mv kendryte-toolchain /opt
[sudo] password for gengyue:
gengyue@Ubuntu0:~/Downloads$ ls /opt
kendryte-toolchain
gengyue@Ubuntu0:~/Downloads$
```

## 4. 移添加 toolchain 到环境变量并使其生效。

输入以下命令编译/etc/profile

```
sudo nano /etc/profile
```

```
gengyue@Ubuntu0:~/Downloads$ sudo nano /etc/profile  
[sudo] password for gengyue:
```

在最底部添加以下内容

```
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/opt/kendryte-toolchain/bin
```

```
export LD_LIBRARY_PATH
```

```
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/opt/kendryte-toolchain/bin  
export LD_LIBRARY_PATH
```

其中/opt/kendryte-toolchain/bin 为 toolchain 的安装路径，需根据实际替换。

### 三、安装 VSCode 编辑器

#### 1. 下载 VSCode 编辑

VSCode 官方下载地址: <https://code.visualstudio.com/Download>

可以根据自己系统版本选择，Ubuntu 系统可以选择.deb 格式的安装包，这样可以直接双击运行安装，也可以使用命令安装。

## Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extensions.



↓ Windows

Windows 7, 8, 10

User Installer	64 bit	32 bit
System Installer	64 bit	32 bit
.zip	64 bit	32 bit



↓ .deb

Debian, Ubuntu

↓ .rpm

Red Hat, Fedora, SUSE

.deb	64 bit
.rpm	64 bit
.tar.gz	64 bit

Snap Store



↓ Mac

macOS 10.10+

## 2. 安装 VSCode

### 方法一：图形化安装方式

双击运行下载 VSCode 安装包. deb 文件, 然后会弹出 Ubuntu 软件管理器界面, 点击安装即可。

### 方法二：命令行安装方式

确认下载的 code 安装包. deb 文件在当前目录下, 并且运行以下命令安装:

```
sudo dpkg -i code_1.45.1-1589445302_amd64.deb
```

正确输入用户名密码并按回车确认。

```
gengyue@Ubuntu0:~/Downloads$ ls
code_1.45.1-1589445302_amd64.deb  kendryte-toolchain-ubuntu-amd64-8.2.0-20190213.tar.gz
gengyue@Ubuntu0:~/Downloads$ sudo dpkg -i code_1.45.1-1589445302_amd64.deb
[sudo] password for gengyue:
Selecting previously unselected package code.
(Reading database ... 151123 files and directories currently installed.)
Preparing to unpack code_1.45.1-1589445302_amd64.deb ...
Unpacking code (1.45.1-1589445302) ...
Setting up code (1.45.1-1589445302) ...
Processing triggers for gnome-menus (3.13.3-11ubuntu1.1) ...
Processing triggers for desktop-file-utils (0.23-1ubuntu3.18.04.2) ...
Processing triggers for mime-support (3.60ubuntu1) ...
```

## 3. 测试 VSCode

输入以下命令打开 VSCode

code .

```
gengyue@Ubuntu0:~/Downloads$ code .
gengyue@Ubuntu0:~/Downloads$
```

## 四、下载 K210 软件 SDK

K210 官方提供两个 SDK, 一个是裸机版 SDK, 另一个是 freertos 系统 SDK, 我们以裸机版 SDK 为例。

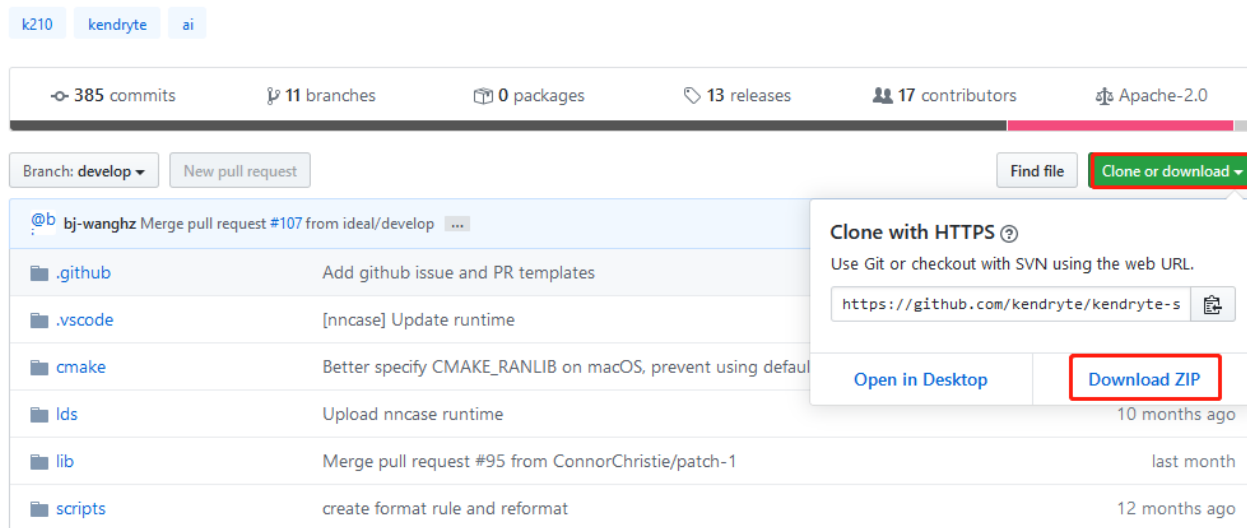
### 1. 下载 K210 裸机版 SDK

K210 裸机版 SDK 下载地址:

<https://github.com/kendryte/kendryte-standalone-sdk>

点击右边的 Clone or download, Download ZIP 下载 SDK。

Standalone SDK for kendryte K210 <https://kendryte.com>



2. 下载完成后, 解压.zip 文件

`unzip kendryte-standalone-sdk-develop.zip`

```
gengyue@Ubuntu0:~/Downloads$ ls
code_1.45.1-1589445302_amd64.deb    kendryte-toolchain-ubuntu-amd64-8.2.0-20190213.tar.gz
kendryte-standalone-sdk-develop.zip
gengyue@Ubuntu0:~/Downloads$ unzip kendryte-standalone-sdk-develop.zip
Archive:  kendryte-standalone-sdk-develop.zip
6c3540600422257029c40e6c77b137d5a2c5d794
  creating: kendryte-standalone-sdk-develop/
  inflating: kendryte-standalone-sdk-develop/.clang-format
  creating: kendryte-standalone-sdk-develop/.github/
```

3. 使用 VSCode 打开 SDK。

先进入 SDK 目录下, 并使用 VSCode 打开

`cd kendryte-standalone-sdk-develop`

`code .`

```
gengyue@Ubuntu0:~/Downloads$ cd kendryte-standalone-sdk-develop/
gengyue@Ubuntu0:~/Downloads/kendryte-standalone-sdk-develop$ code .
gengyue@Ubuntu0:~/Downloads/kendryte-standalone-sdk-develop$
```

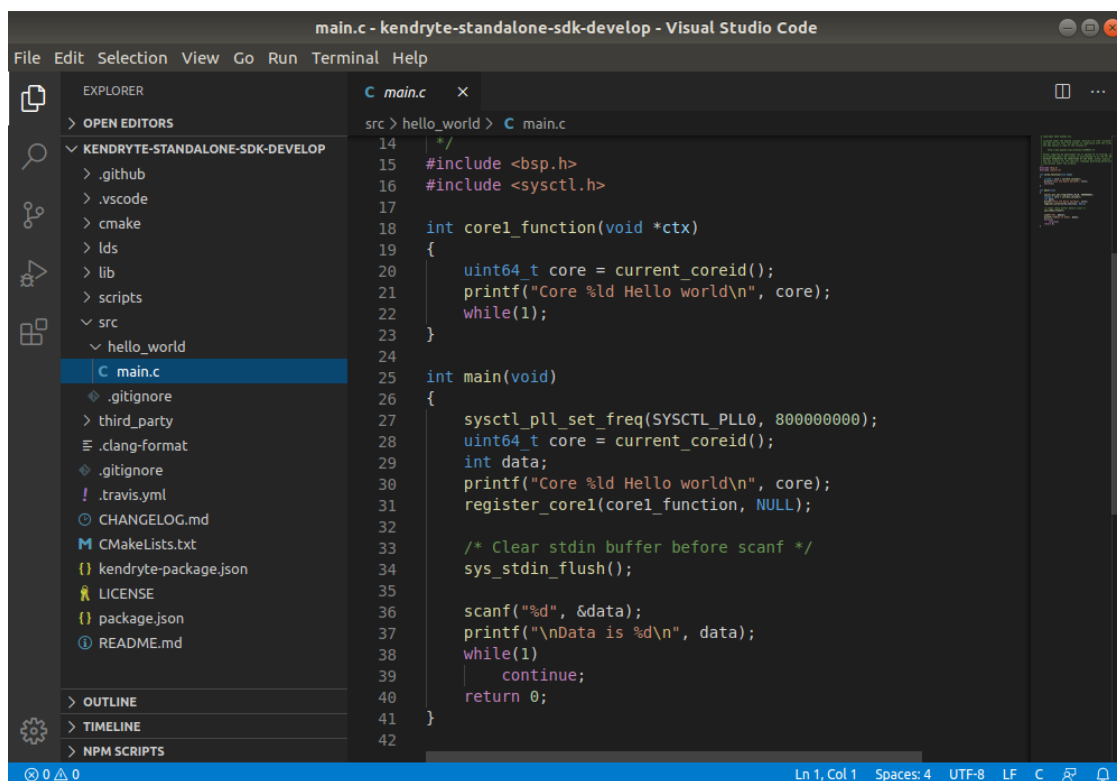
#### 4. SDK 文件架构简介。



### 五、编译程序

#### 1. 用 VSCode 打开 SDK

2. 查看 src 路径的 hello\_world 工程的 main.c 文件。运行时，它会从 USB 串口打印出数据。





### 3. 创建 build 文件夹并进入

返回 Ubuntu 的终端,输入以下命令创建 build 文件夹,并且进入 build。build 文件夹用于保存 cmake 编译产生的文件,也是我们最后要得到的烧录固件的保存路径。

```
mkdir build
```

```
cd build
```

```
gengyue@Ubuntu0:~/Downloads/kendryte-standalone-sdk-develop$ mkdir build
gengyue@Ubuntu0:~/Downloads/kendryte-standalone-sdk-develop$ cd build
gengyue@Ubuntu0:~/Downloads/kendryte-standalone-sdk-develop/build$
```

### 4. CMake 编译程序

```
cmake .. -DPROJ=hello_world -DTOOLCHAIN=/opt/kendryte-toolchain/bin
```

```
gengyue@Ubuntu0:~/Downloads/kendryte-standalone-sdk-develop/build$ cmake .. -DPROJ=hello_world -DTOOLCHAIN=/opt/kendryte-toolchain/bin
PROJ = hello_world
-- Check for RISC-V toolchain ...
-- Using /opt/kendryte-toolchain/bin RISC-V toolchain
-- The C compiler identification is GNU 8.2.0
```

### 5. make 编译程序

```
gengyue@Ubuntu0:~/Downloads/kendryte-standalone-sdk-develop/build$ make
Scanning dependencies of target nncase
[ 2%] Building CXX object lib/nncase/CMakeFiles/nncase.dir/nncase.cpp.obj
/home/gengyue/Downloads/kendryte-standalone-sdk-develop/lib/nncase/nncase.cpp:29:
nning: 'void {anonymous}::kpu_upload_dma(dmac_channel_number_t, const uint8_t*, u
, size_t, plic_irq_callback_t, void*)' defined but not used [-Wunused-function]

Scanning dependencies of target hello_world
[ 97%] Building C object CMakeFiles/hello_world.dir/src/hello_world/main.c.obj
[100%] Linking C executable hello_world
Generating .bin file ...
[100%] Built target hello_world
gengyue@Ubuntu0:~/Downloads/kendryte-standalone-sdk-develop/build$
```

### 6. 查看生成的文件

```
gengyue@Ubuntu0:~/Downloads/kendryte-standalone-sdk-develop/build$ ls
CMakeCache.txt  cmake_install.cmake  hello_world.bin  Makefile
CMakeFiles      hello_world          lib
gengyue@Ubuntu0:~/Downloads/kendryte-standalone-sdk-develop/build$
```

## 六、kflash 烧录程序

### 1. 安装 pip3

Ubuntu18.04 是自带 Python3 的，但是不自带 pip3，可以输入以下命令安装 pip3。如果已安装 pip3 请忽略此步骤。

```
sudo apt-get install python3-pip -y --fix-missing
```

```
gengyue@Ubuntu0:~$ sudo apt-get install python3-pip -y --fix-missing
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  efibootmgr fonts-liberation2 fonts-opensymbol gir1.2-gst-plugins-base-1.0
  gir1.2-gstreamer-1.0 gir1.2-gudev-1.0 gir1.2-udisks-2.0 grilo-plugins-0.3-base
```

### 2. 安装 kflash

切换到 root 用户，输入以下命令安装 kflash

```
sudo pip3 install kflash
```

```
root@Ubuntu0:/home/gengyue/Downloads/kendryte-standalone-sdk-develop/build# sudo pip3 i
ninstall kflash
Collecting kflash
  Downloading https://files.pythonhosted.org/packages/b2/7c/62858234aa44a4d2b3cecc57436
eff4580acca8a1147493acdf91d981e9e/kflash-0.8.5-py3-none-any.whl (77kB)
    100% |████████████████████████████████████████| 81kB 227kB/s
Collecting pyserial>=3.4 (from kflash)
  Downloading https://files.pythonhosted.org/packages/0d/e4/2a744dd9e3be04a0c0907414e2a
01a7c88bb3915cbe3c8cc06e209f59c30/pyserial-3.4-py2.py3-none-any.whl (193kB)
    100% |████████████████████████████████████████| 194kB 411kB/s
Collecting enum34>=1.1.6 (from kflash)
  Downloading https://files.pythonhosted.org/packages/63/f6/ccb1c83687756aeabbf3ca0f213
508fcfb03883ff200d201b3a4c60cedcc/enum34-1.1.10-py3-none-any.whl
Collecting pyelftools>=0.25 (from kflash)
  Downloading https://files.pythonhosted.org/packages/bb/2f/bf41f3c3867d6707fa9b872658b
b23088a64d0e522e8979f54c694b8cbe1/pyelftools-0.26-py2.py3-none-any.whl (136kB)
    100% |████████████████████████████████████████| 143kB 686kB/s
Installing collected packages: pyserial, enum34, pyelftools, kflash
Successfully installed enum34-1.1.10 kflash-0.8.5 pyelftools-0.26 pyserial-3.4
```

### 3. 将当前用户名添加到拨号组里。

把\$(whoami)替换成自己的用户名即可。

```
sudo usermod -a -G dialout $(whoami)
```

```
gengyue@Ubuntu0:~$ sudo usermod -a -G dialout gengyue
gengyue@Ubuntu0:~$
```

#### 4. 输入以下命令检查 kflash

kflash --help

```
gengyue@Ubuntu0:~/Downloads/kendryte-standalone-sdk-develop/build$ kflash --help
usage: kflash [-h] [-p PORT] [-f FLASH] [-b BAUDRATE] [-l BOOTLOADER] [-k KEY]
              [-v] [--verbose] [-t] [-n] [-s]
              [-B {kd233,dan,bit,bit_mic,goE,goD,maixduino,trainer}] [-S]
              firmware

positional arguments:
  firmware              firmware bin path

optional arguments:
  -h, --help            show this help message and exit
  -p PORT, --port PORT  COM Port
  -f FLASH, --flash FLASH
                        SPI Flash type, 0 for SPI3, 1 for SPI0
  -b BAUDRATE, --baudrate BAUDRATE
                        UART baudrate for uploading firmware
  -l BOOTLOADER, --bootloader BOOTLOADER
                        Bootloader bin path
  -k KEY, --key KEY     AES key in hex, if you need encrypt your firmware.
  -v, --version         Print version.
  --verbose            Increase output verbosity
  -t, --terminal        Start a terminal after finish (Python miniterm)
  -n, --noansi          Do not use ANSI colors, recommended in Windows CMD
  -s, --sram            Download firmware to SRAM and boot
  -B {kd233,dan,bit,bit_mic,goE,goD,maixduino,trainer}, --Board {kd233,dan,bit,bit_mic,
                        Select dev board
  -S, --Slow            Slow download mode
gengyue@Ubuntu0:~/Downloads/kendryte-standalone-sdk-develop/build$
```

其中包括 kflash 的参数

- h: 帮助信息
- p: 选择端口
- f: SPI Flash 类型, 0 表示 SPI3, 1 表示 SPI0
- b: 选择波特率
- l: bootloader 的路径
- k: AES 加密
- v: 打印版本
- verbose: 打印烧录固件的详细信息
- t: 烧录固件完成后打开终端
- n: 不要使用 ANSI 颜色, 推荐在 Windows CMD
- s: 下载固件到 SRAM
- B: 预设的开发板型号
- S: 慢速下载模式

5. kflash.py 的源码可以在以下网址查看:

<https://github.com/kendryte/kflash.py>

6. 查看 K210 开发板端口号

将 K210 开发板与电脑连接, 输入以下命令查看 K210 开发板的端口号, 可以看到 K210 开发板的端口号为/dev/ttyUSB0.

```
ls /dev/ttyUSB*
```

```
gengyue@Ubuntu0:~$ ls /dev/ttyUSB*  
/dev/ttyUSB0  
gengyue@Ubuntu0:~$
```

7. 烧录固件

切换到 K210-SDK 的 build 目录下, 找到刚刚编译生成的 hello\_world.bin 文件。

```
gengyue@Ubuntu0:~$ cd Downloads/kendryte-standalone-sdk-develop/build/  
gengyue@Ubuntu0:~/Downloads/kendryte-standalone-sdk-develop/build$ ls  
CMakeCache.txt  cmake_install.cmake  hello_world.bin  Makefile  
CMakeFiles      hello_world          lib  
gengyue@Ubuntu0:~/Downloads/kendryte-standalone-sdk-develop/build$
```

输入以下命令烧录到 K210 开发板上, 其中-p 表示端口, -t 表示烧录完成后自动打开终端, hello\_world.bin 是实际要烧录的固件名称, 根据实际修改。

```
sudo kflash -p /dev/ttyUSB0 -t hello_world.bin
```

```
gengyue@Ubuntu0:~/Downloads/kendryte-standalone-sdk-develop/build$ sudo kflash -p /dev/
ttyUSB0 -t hello_world.bin
[INFO] COM Port Selected Manually: /dev/ttyUSB0
[INFO] Default baudrate is 115200 , later it may be changed to the value you set.
[INFO] Trying to Enter the ISP Mode...
.
[INFO] Automatically detected dan/bit/trainer

[INFO] Greeting Message Detected, Start Downloading ISP
Downloading ISP: |=====| 100.0% 10kiB/s
[INFO] Booting From 0x80000000
[INFO] Wait For 0.1 second for ISP to Boot
[INFO] Boot to Flashmode Successfully
[INFO] Selected Flash: On-Board
[INFO] Initialization flash Successfully
Programming BIN: |=====| 100.0% 10kiB/s
[INFO] Rebooting...
--- forcing DTR inactive
--- forcing RTS inactive
--- Miniterm on /dev/ttyUSB0 115200,8,N,1 ---
--- Quit: Ctrl+] | Menu: Ctrl+T | Help: Ctrl+T followed by Ctrl+H ---
Core 0 Hello world
Core 1 Hello world
█
```

固件烧录完成会自动打开 K210 的串口终端，K210 会打印信息。

按 CTRL+] 退出 K210 串口终端。