2.4 Linux development environment(optional)

1. Tools needed to compile K210

We are building the K210 development environment through the VSCode editor with the Win10 system. The following are the tools we need.

1-CMake.
2-Toolchain.
3-VSCode.
4-K210-SDK.
5-flash.

2. Install CMake

2.1 Verify whether CMake be installed

If you have already installed it and the CMake version is greater than 3.0, you do not need to re-install it, you can skip the step of installing CMake.

Open the Ubuntu terminal and enter the following command. If the following interface appears, it means that CMake is not installed.

cmake -version

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

Command 'cmake' not found, but can be installed with:

sudo apt install cmake
```

Tip, we use Ubuntu 18.04 bit system as a example.

2.2 Download and install Cmake and make

Method-1:

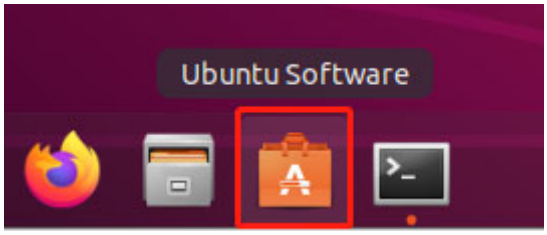
Input following command to install cmake:

sudo apt-get install cmake -y
sudo apt-get install make -y

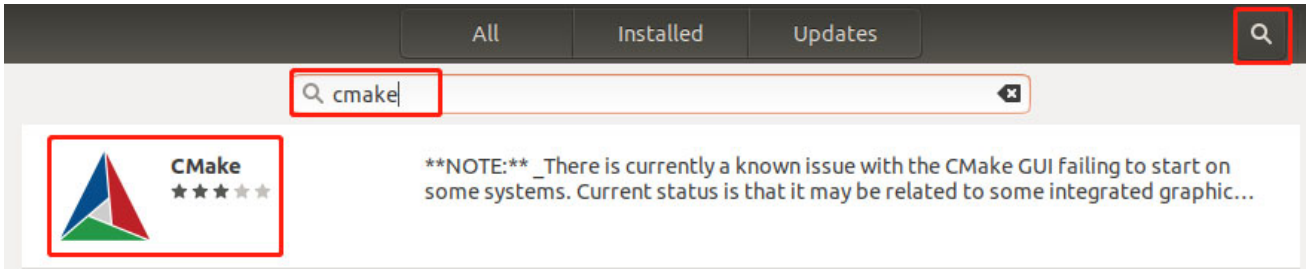
Due to network problems, downloading takes a long time, please be patient.

Method-2:

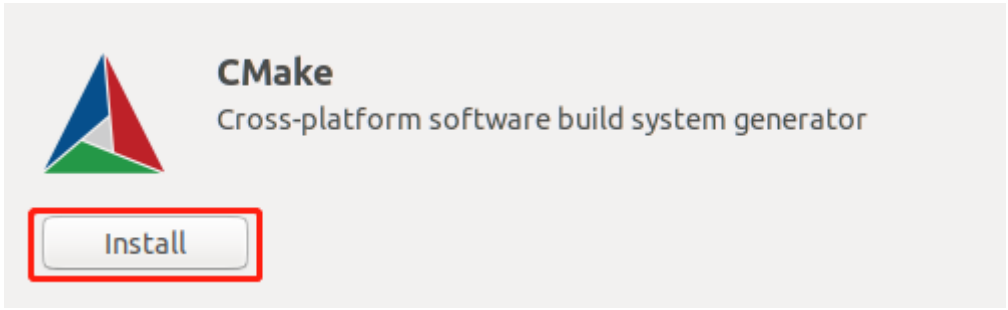
Download cmake by Ubuntu's own software manager.



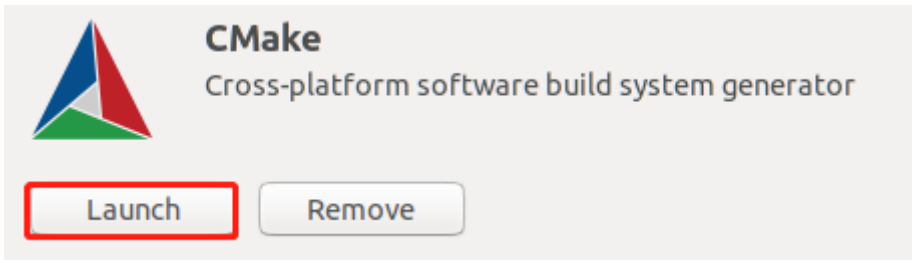
Search “cmake”.



Click “Install”, start install.



After installation is complete. Click “Launch” to run CMake.
At the same time, there is also a CMake icon in the Ubuntu application.



2.3 Check and verify CMake
Open the terminal, enter **cmake -version.**
If you can see the CMake version number you installed, it means the installation is successful.

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

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

3. Install the cross compiler Toolchain
3.1 We have provided this toolchain file**(kendryte-toolchain-ubuntu-amd64-8.2.0-20190213.tar.gz),** please check [Tools] to get this tool.
3.2 Extract toolchain file
Transfer **kendryte-toolchain-ubuntu-amd64-8.2.0-20190213.tar.gz** file to **Downloads** director.
Enter Downloads director. Input following command:
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
```

After extract is completed, we will get a kendryte-toolchain folder.

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

3.3 Input following command to move all the contents of the kendryte-toolchain folder to the /opt directory.

sudo mv kendryte-toolchain /opt

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

4. **Add toolchain to the environment variable and make it effective.**

Input following command to edit/etc/profile

sudo nano /etc/profile

Enter the user password correctly and press Enter to confirm.

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

Add the following at the bottom.

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
```

/opt/kendryte-toolchain/bin is toolchain install path, you need to modify it according to the actual installation path.

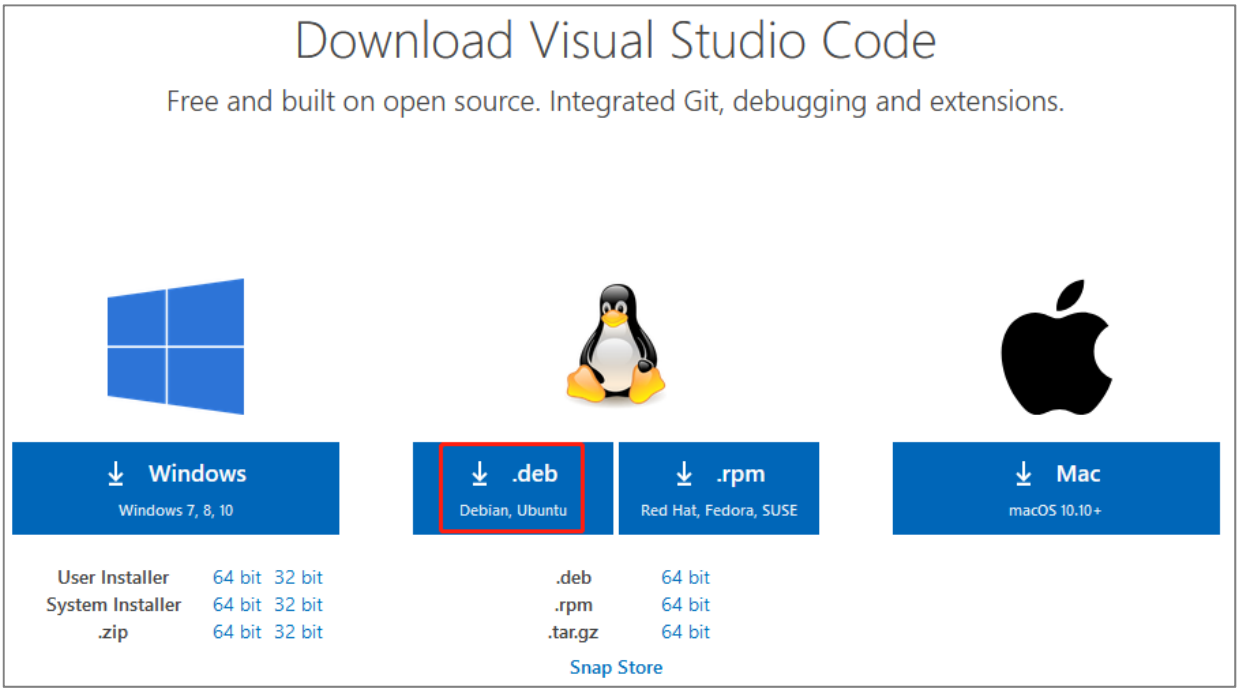
3. Install VSCode editor

3.1 Download VSCode editor

VSCode official download address: https://code.visualstudio.com/Download

According to your own system version, you can choose version. I choose [System Installer 64bit], and downloaded .exe file.

Then, we can install it directly, it can be used by all users.



3.2 Install VSCode

Method-1:

Double-click to run to download the VSCode installation package .deb file, then the Ubuntu software manager interface will pop up, just click Install.

Method-2:

Ensure that the downloaded code installation package .deb file is in the current directory, and input the following command to install:

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

Enter the user password correctly and press Enter to confirm.

```
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) ...
```

3.3 Testing VSCode

input the following command to open VSCode.

code .

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

4.Download K210 software SDK

K210 official provide two SDK.

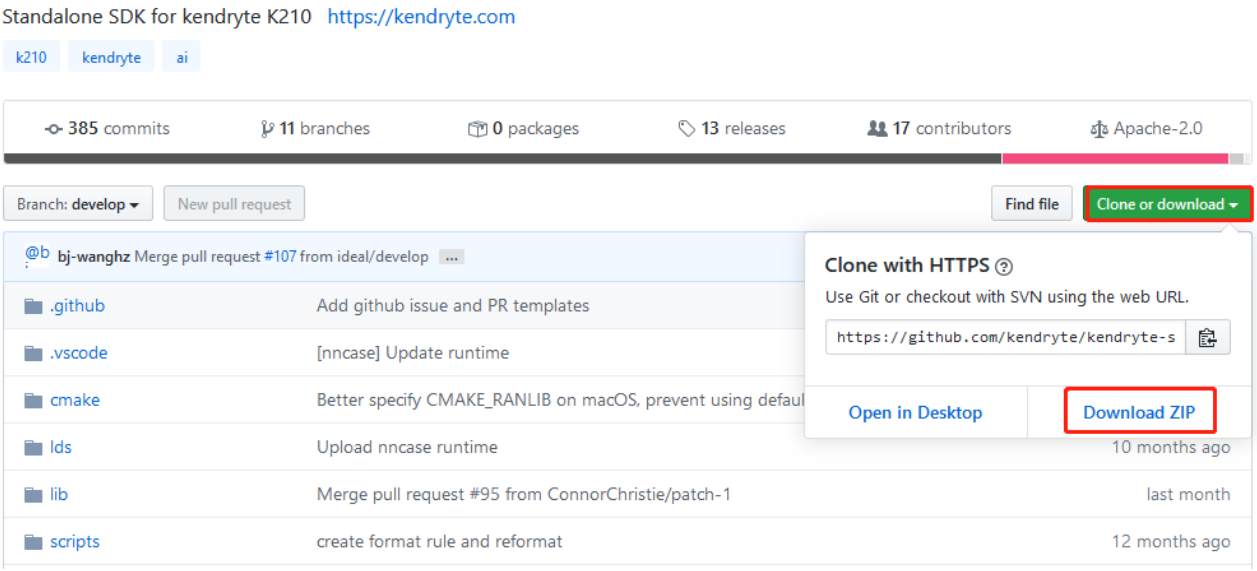
Bare machine version SDK and **freertos SDK**

Eg: we use a **Bare machine version SDK**

4.1 Download K210 **Bare machine version SDK**

Click “Clone or download”---> click “Download ZIP” download SDK.

We have provide this file, please click [Tools] to download this file.



4.2 After download is complete. Input following command to extract .zip file.

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/
```

4.3 Open SDK by VSCode.

Enter SDK directory and input following command.

cd kendryte-standalone-sdk-develop

code .

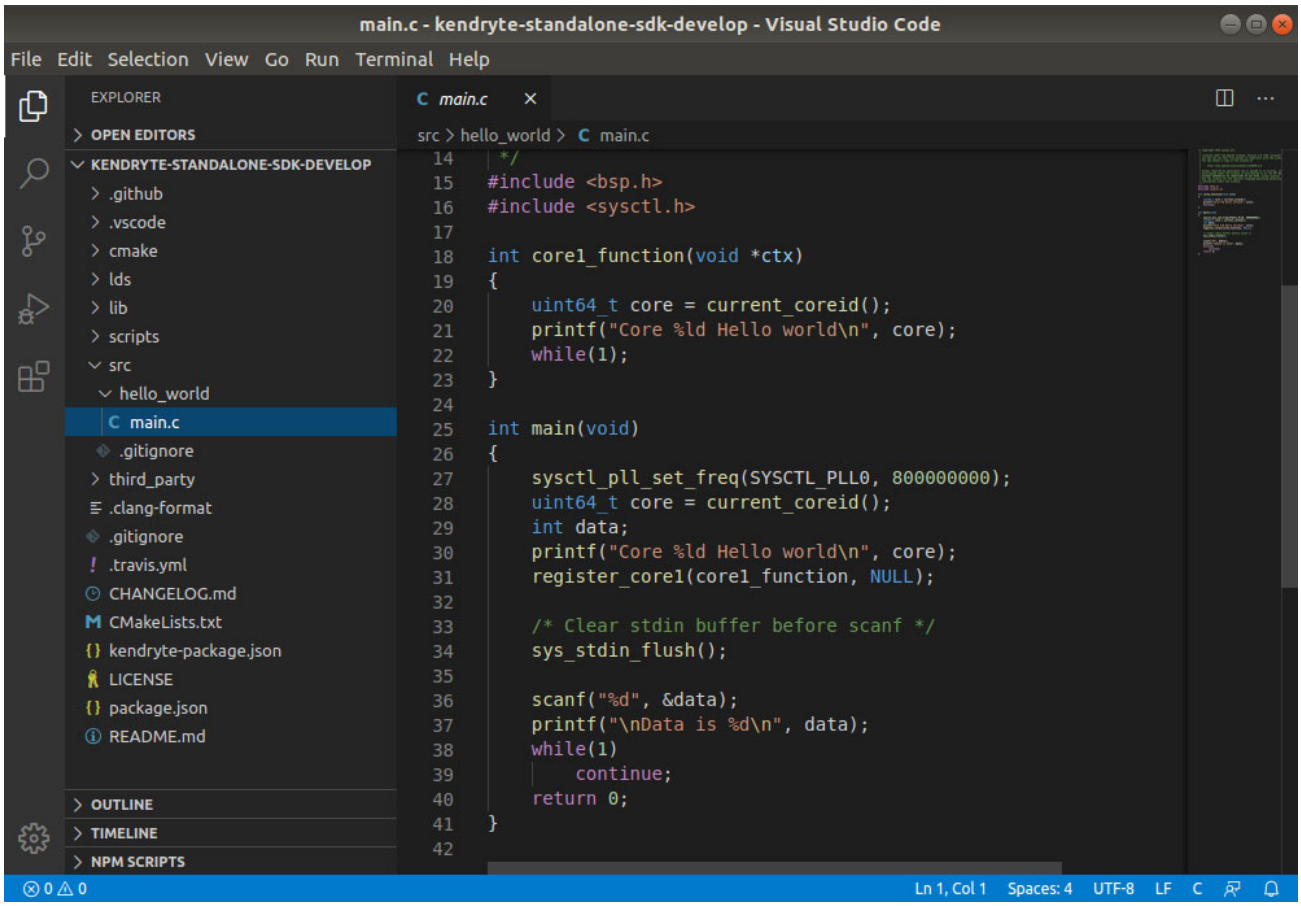
```
gengyue@Ubuntu0:~/Downloads$ cd kendryte-standalone-sdk-develop/
gengyue@Ubuntu0:~/Downloads/kendryte-standalone-sdk-develop$ code .
gengyue@Ubuntu0:~/Downloads/kendryte-standalone-sdk-develop$
```

5.Compile program

5.1 Open SDK by VSCode.

5.2 View the **main.c** file of the hello_world project in the src folder. When we run the modified program, it will print out the data from the USB serial port.

As shown below.



5.3 Create build folder

Enter the following command in the Ubuntu terminal to create the build folder.

Enter the build. **The build folder is used to save the files generated by cmake compilation, and it is also the save path of the write firmware.**

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$
```

5.4 CMake compile program

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

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

5.5 make compile program

```
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
ning: '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$
```

5.6 Input command ls to View the generated file.

```
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$
```

6. Write program into K210 board

6.1 Install pip3

Ubuntu18.04 comes with Python3, but does not come with pip3, you can enter the following command to install pip3.

If pip3 is already installed, please ignore this step.

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
```

6.2 Install kflash

witch to root user, enter the following command to install kflash.

sudo pip3 install kflash

```
root@Ubuntu0:/home/gengyue/Downloads/kendryte-standalone-sdk-develop/build# sudo pip3 i
nstall 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
```

6.3 Input following command to add the current user name to the dial group.

(You need tp replace $(whoami) with your username.)

sudo usermod -a -G dialout $(whoami)

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

6.4 Enter the following command to check 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$
```

6.5 The source code of kflash.py can be viewed at the following URL.

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

6.6 View K210 board port.

connect the computer to the K210 development board through the Type-C data cable. **And open the power switch of K210 board.**

Input following command to view K210 board port.

ls /dev/ttyUSB*

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

We can port is /dev/ttyUSB0

6.7 Write firmware

Switch to the K210-SDK directory and find the **hello_world.bin** file just compiled.

As shown below.

```
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$
```

Enter the following commands to burn to the K210 development board.

-p means the port,

-t means the terminal will be opened automatically after the burning is completed,

hello_world.bin is the name of the firmware to be burned, you can modify it according to the actual situation.

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
```

After the firmware is write, the serial terminal of the K210 will be opened automatically, and it will print the information.
Press **CTRL+]** to exit the K210 serial terminal.