

关于本手册

本文介绍了如何为PaddlePi-K210搭建命令行开发环境。

发布说明

| 日期 | 版本 | 发布说明 |
|-----------|------|--------|
| 2020.1.20 | V0.1 | 首次发布 |
| 2020.5.7 | V0.2 | 修正部分描述 |

免责声明

本文中的信息，包括参考的 URL 地址，如有变更，恕不另行通知。文档“按现状”提供，不负任何担保责任，包括对适销性、适用于特定用途或非侵权性的任何担保和任何提案、规格或样品在他处提到的任何担保。

本文档不负任何责任，包括使用本文档内信息产生的侵犯任何专利权行为的责任。本文档在此未以禁止反言或其他方式授予任何知识产权使用许可，不管是明示许可还是暗示许可。

文中提到的所有商标名称、商标和注册商标均属其各自所有者的财产，特此声明。

版权公告

版权归 © 2020 嘉楠科技所有。保留所有权利。

1、概述

本文介绍如何在Windows和Ubuntu下为PaddlePi-K210搭建开发环境，用户可根据个人喜好选择开发环境。

2、Windows 命令行开发环境搭建

2.1 安装工具链

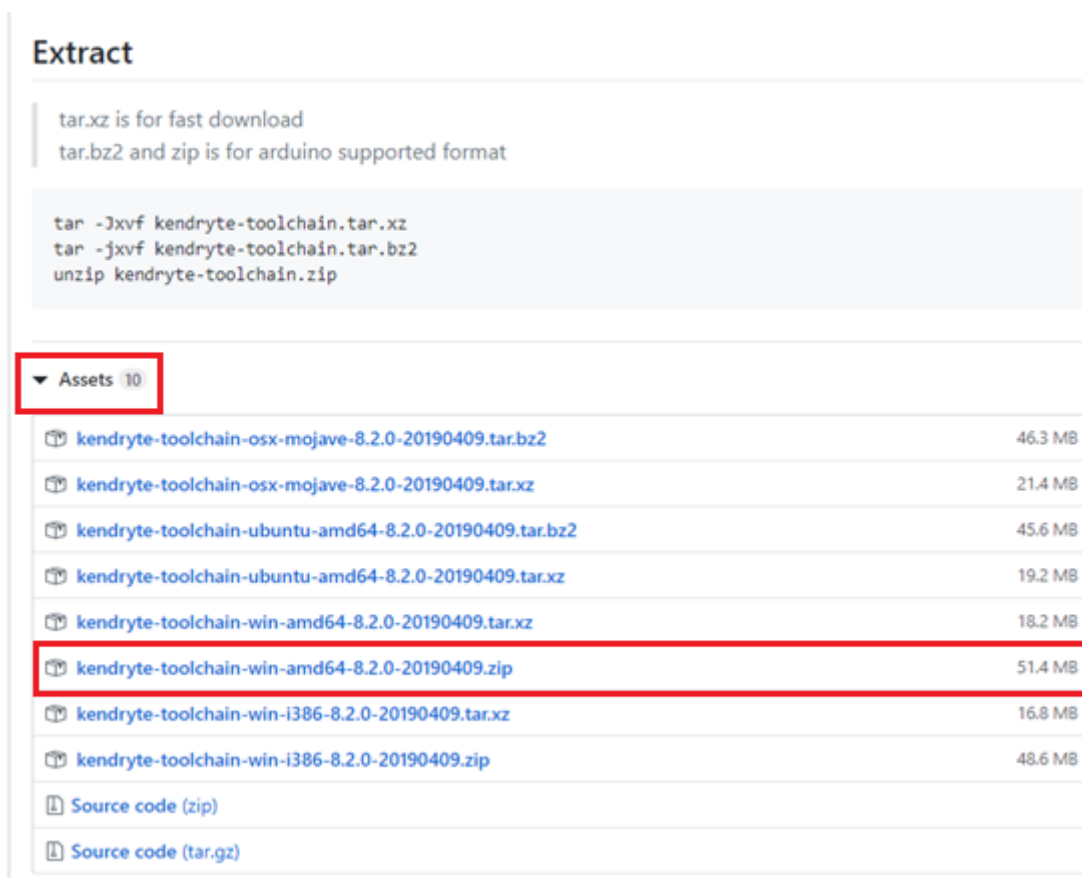
1、下载[cmake](#) V3.0之后的Windows版本。这里以V3.12.4版本为例。把cmake安装到D:\cmake目录，并把D:\cmake\bin目录添加到PATH环境变量。

2、打开一个新的cmd窗口，输入cmake --version命令，若看到如下信息说明设置正确。

```
C:\Users\mbed2>cmake -version
cmake version 3.12.4

CMake suite maintained and supported by Kitware (kitware.com/cmake).
C:\Users\mbed2>
```

3、从[Kendryte Github](#)下载Windows版本工具链。打开网页后展开Assets可看到下载链接。













Extract

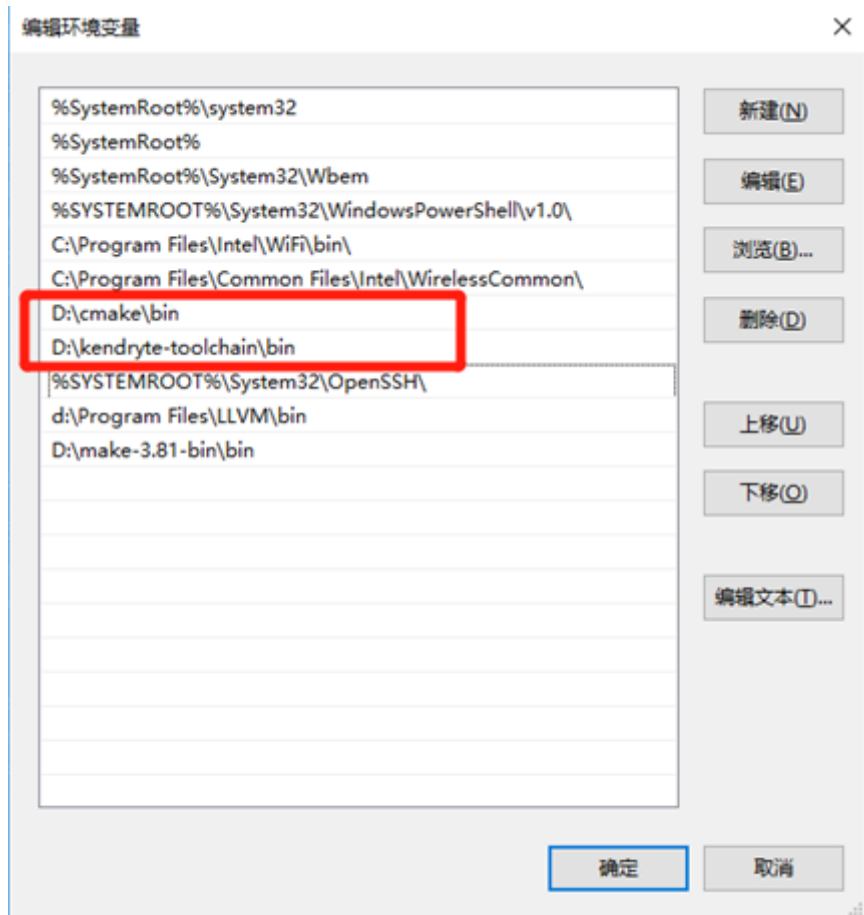
tar.xz is for fast download
tar.bz2 and zip is for arduino supported format

```
tar -Jxvf kendryte-toolchain.tar.xz
tar -jxvf kendryte-toolchain.tar.bz2
unzip kendryte-toolchain.zip
```

▼ Assets 10

| | |
|--|---------|
|  kendryte-toolchain-osx-mojave-8.2.0-20190409.tar.bz2 | 46.3 MB |
|  kendryte-toolchain-osx-mojave-8.2.0-20190409.tar.xz | 21.4 MB |
|  kendryte-toolchain-ubuntu-amd64-8.2.0-20190409.tar.bz2 | 45.6 MB |
|  kendryte-toolchain-ubuntu-amd64-8.2.0-20190409.tar.xz | 19.2 MB |
|  kendryte-toolchain-win-amd64-8.2.0-20190409.tar.xz | 18.2 MB |
|  kendryte-toolchain-win-amd64-8.2.0-20190409.zip | 51.4 MB |
|  kendryte-toolchain-win-i386-8.2.0-20190409.tar.xz | 16.8 MB |
|  kendryte-toolchain-win-i386-8.2.0-20190409.zip | 48.6 MB |
|  Source code (zip) | |
|  Source code (tar.gz) | |

4、将下载后的文件解压缩，将解压后的D:\kendryte-toolchain\bin目录添加到PATH环境变量，如下图。



5、重新打开一个cmd窗口，输入riscv64-unknown-elf-gcc -v命令，看到如下信息说明编译器设置正确。

```
C:\Users\mbd2>riscv64-unknown-elf-gcc -v
Using built-in specs.
COLLECT_GCC=riscv64-unknown-elf-gcc
COLLECT_LTO_WRAPPER=../kendryte-toolchain/bin/./libexec/gcc/riscv64-unknown-elf/8.2.0/lto-wrapper.exe
Target: riscv64-unknown-elf
Configured with: /workdir/riscv-gcc/configure --target=riscv64-unknown-elf --host=x86_64-w64-mingw32 --prefix=/opt/kendryte-toolchain --disable-shared --enable-threads=posix --enable-languages=c,c++ --enable-libatomic --without-system-zlib --enable-tls --with-newlib --with-sysroot=/opt/kendryte-toolchain/riscv64-unknown-elf --with-native-system-header-dir=/include --disable-libmudflap --disable-libssp --disable-libquadmath --disable-libgomp --disable-nls --src=../riscv-gcc --enable-checking=yes --disable-multilib --with-abi=lp64f --with-arch=rv64imafc 'CFLAGS_FOR_TARGET=-Os -ffunction-sections -fdata-sections -mmodel=medany' 'CXXFLAGS_FOR_TARGET=-Os -ffunction-sections -fdata-sections -mmodel=medany'
Thread model: posix
gcc version 8.2.0 (GCC)
```

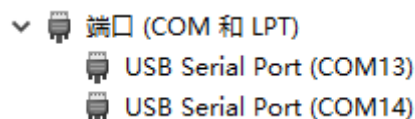
2.2、安装Openocd

若不进行在线仿真调试，可跳过这个步骤，参考2.3节开始编译程序。

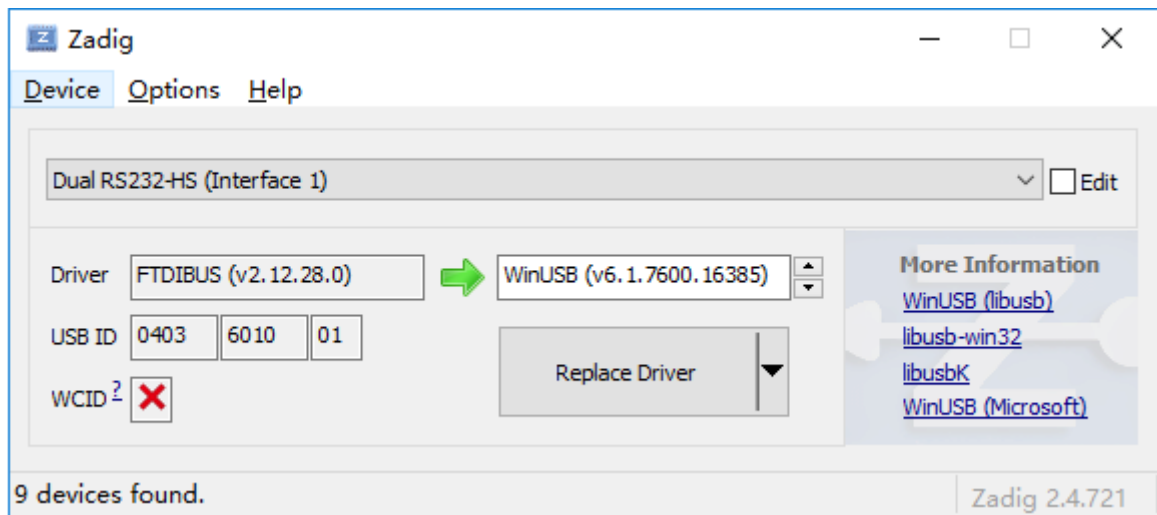
1、从[Kendryte Github](#)下载Windows版本的Openocd并解压缩。

2、安装FT2232驱动。

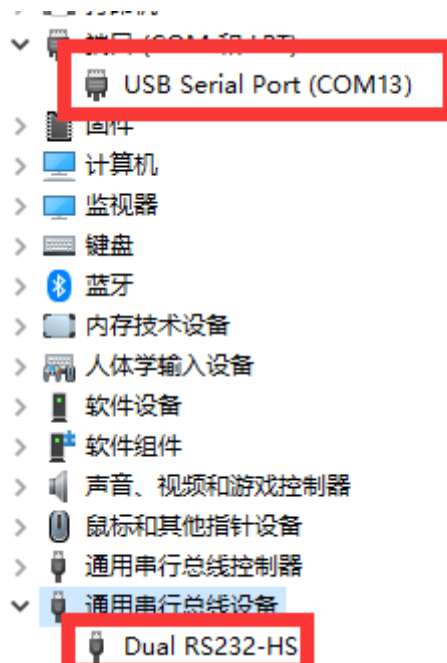
将FT2232 Debugger通过USB连接后，默认设备管理器出现两个串口设备，如下图所示。



运行 XXX\kendryte-openocd\tool下的 zadig-2.4.exe (XXX为kendryte-openocd的保存路径)，点击 Options->List All Devices，在下拉列表选中Dual RS232-HS (Interface 1)，点击【Replace Driver】即可将驱动替换为WinUSB驱动。



注意，替换**Dual RS232-HS (Interface 1)**，**不要将Dual RS232-HS (Interface 0) 转为WinUSB**。替换过驱动后建议断开FT232连接后重新连接。替换完成后，驱动将呈现如下状态：



3、配置 OpenOCD

下载kendryte-openocd-0.X.X-win32，在tcl文件夹中，添加ft232.cfg文件，ft232.cfg填写如下内容：

```
interface ftdi
ftdi_vid_pid 0x0403 0x6010
ftdi_channel 1
ftdi_layout_init 0x00e8 0x00eb

transport select jtag
ftdi_tdo_sample_edge falling
adapter_khz 10000

gdb_port 3333
telnet_port 4444

set _CHIPNAME riscv
```

```
jtag newtap $_CHIPNAME cpu -irlen 5 -expected-id 0x04e4796b

set _TARGETNAME $_CHIPNAME.cpu
target create $_TARGETNAME riscv -chain-position $_TARGETNAME

init
halt
```

2.3 编译

- 1、从[Kendryte Github](#)下载Kendryte-standalone-sdk（请使用develop分支的代码）。
- 2、在Kendryte-standalone-sdk目录下创建build目录。
- 3、打开cmd，进入build目录，在build目录下运行cmake。

```
cmake .. -DPROJ=hello_world -G "Unix Makefiles"
```

当输出如下结果表示cmake 正常。

```
Makefile created.

-- Configuring done
-- Generating done
-- Build files have been written to: C:/Users/mbed2/Desktop/kendryte-standalone-sdk/build
```

- 4、在build 目录下编译。

```
make -j
```

当编译输出以下结果表示make 成功。

```
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
```

2.4 调试

若不进行在线仿真调试，可跳过这个步骤，参考2.5节下载方法验证程序运行结果。

- 1、运行openocd

从cmd进入kendryte-openocd/bin目录, 运行：

```
.\openocd.exe -f ..\tc1\ft232.cfg
```

- 2、gdb调试

从另外一个cmd窗口进入kendryte-standalone-sdk\build目录。

```
riscv64-unknown-elf-gdb hello_world -eval-command="target remote:3333"
```

其中：

- hello_world是编译过程生成的elf文件。

- 3333是openocd监听的端口号。

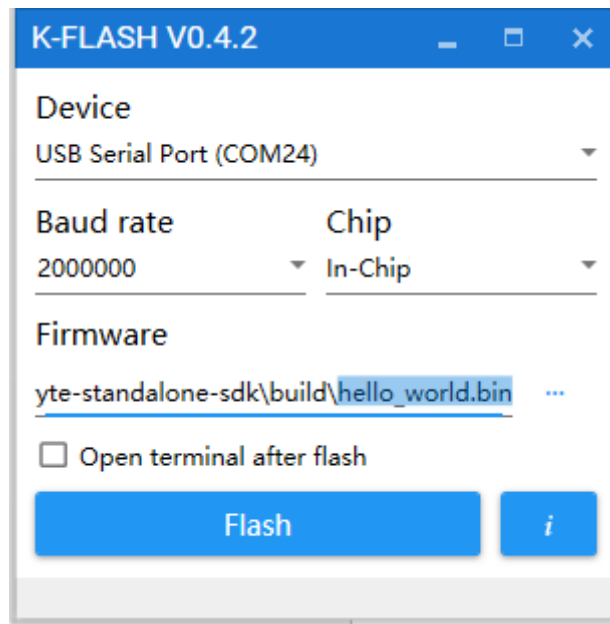
成功连接openocd后，就可以使用`load`、`break`、`continue`等gdb命令来调试了。

注: gdb手册参考[gdb website](#)。

2.5 下载

- 1、从[Kendryte Github](#)下载Windows版本K-flash工具。

2、开发板插上USB Type-C，上电进入ISP 模式。运行K-Flash.exe，如下图所示：自动识别 COM 号及设置波特率，选择Device和Firmware，例如hello_world.bin，点击Flash开始下载。



3、ubuntu18.04 命令行开发环境搭建

3.1 安装工具链

1、安装build-essential以获取make工具。

```
$ sudo apt install build-essential
```

若如上命令执行不成功，执行如下命令更新一下软件源。








```
$ sudo apt update
```

2、安装cmake。

```
$ sudo apt install cmake
```

3、从[Kendryte Github](#)下载如下图所示的Ubuntu版本工具链，放到/opt目录并解压缩。

▼ Assets 10

| | |
|--|---------|
|  kendryte-toolchain-osx-mojave-8.2.0-20190409.tar.bz2 | 46.3 MB |
|  kendryte-toolchain-osx-mojave-8.2.0-20190409.tar.xz | 21.4 MB |
|  kendryte-toolchain-ubuntu-amd64-8.2.0-20190409.tar.bz2 | 45.6 MB |
|  kendryte-toolchain-ubuntu-amd64-8.2.0-20190409.tar.xz | 19.2 MB |
|  kendryte-toolchain-win-amd64-8.2.0-20190409.tar.xz | 18.2 MB |
|  kendryte-toolchain-win-amd64-8.2.0-20190409.zip | 51.4 MB |
|  kendryte-toolchain-win-i386-8.2.0-20190409.tar.xz | 16.8 MB |
|  kendryte-toolchain-win-i386-8.2.0-20190409.zip | 48.6 MB |
|  Source code (zip) | |
|  Source code (tar.gz) | |

操作命令如下：

```
$ sudo mv kendryte-toolchain-ubuntu-amd64-8.2.0-20190409.tar.bz2 /opt
$ cd /opt
$ sudo tar -jxvf kendryte-toolchain-ubuntu-amd64-20190409.tar.bz2
```

打开~/.bashrc文件，在文件末尾添加如下一行，将/opt/kendryte-toolchain/bin目录添加到PATH环境变量。

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

最后使修改生效。

```
$ source ~/.bashrc
```

3.2 安装openocd

若不进行在线仿真调试，可跳过这节内容，参考3.3节开始编译。

1、从[Kendryte Github](#)下载Ubuntu版本的Openocd。这里以kendryte-openocd-0.2.3-ubuntu64.tar.gz 为例。

```
$sudo mv ken-openocd-0.2.3-ubuntu64.tar.gz /opt
$cd opt/
$ sudo tar -zxvf kendryte-openocd-0.2.3-unbutu64.tar.gz
```

注: 可能需要安装如下依赖包。

```
$ sudo apt install libusb-dev libftdi-dev libhidapi-dev
```

2、确认kendryte-openocd/tcl/ft232.cfg文件是否存在，如果ft232.cfg不存在，需要创建ft232.cfg文件。

```
$ touch /opt/kendryte-openocd/tcl/ft232.cfg
$ vi /opt/kendryte-openocd/tcl/ft232.cfg
```

添加如下内容：

```
interface ftdi
ftdi_vid_pid 0x0403 0x6010
ftdi_channel 1
ftdi_layout_init 0x00e8 0x00eb

transport select jtag
ftdi_tdo_sample_edge falling
adapter_khz 10000

gdb_port 3333
telnet_port 4444

set _CHIPNAME riscv

jtag newtap $_CHIPNAME cpu -irlen 5 -expected-id 0x04e4796b

set _TARGETNAME $_CHIPNAME.cpu
target create $_TARGETNAME riscv -chain-position $_TARGETNAME

init
halt
```

3、确认/etc/udev/rules.d/ftdi-usb.rules文件是否存在，如果ftdi-usb.rules不存在，需要创建ftdi-usb.rules文件。

```
$ sudo touch /etc/udev/rules.d/ftdi-usb.rules
$ sudo vi /etc/udev/rules.d/ftdi-usb.rules
```

在ftdi-usb.rules中添加如下内容：


```
ACTION=="add", ATTR{idVendor}=="0403", MODE:="666"
```

3.3 编译

- 1、从[Kendryte Github](#)下载kendryte-standalone-sdk（请使用develop分支的代码）。
- 2、在kendryte-standalone-sdk目录下创建build目录。
- 3、进入build目录后运行cmake。

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

- 4、编译。

```
$ make -j
```

3.4 调试

若不进行在线仿真调试，可跳过这个步骤，参考3.5节下载以验证程序运行结果。

- 1、运行openocd

```
$ cd /opt/kendry-openocd
$ sudo ./bin/openocd -f ./tcl/ft232.cfg
```

正常将显示如下内容：

```
Kendryte Open On-Chip Debugger For RISC-V v0.1.3 (20180912)
Licensed under GNU GPL v2
ftdi samples TDO on falling edge of TCK
adapter speed: 10000 kHz
Info : clock speed 10000 kHz
Info : JTAG tap: riscv.cpu tap/device found: 0x04e4796b (mfg: 0x4b5 (<unknown>),
part: 0x4e47, ver: 0x0)
Info : [0] Found 4 triggers
Info : [1] Found 4 triggers
[1] halted at 0x8000af8a due to debug interrupt
Info : Examined RISC-V core; found 2 harts, XLEN=64, misa=0x800000000014112d
Info : Listening on port 3333 for gdb connections
[1] halted at 0x8000af8a due to debug interrupt
[0] halted at 0x800015f0 due to debug interrupt
Info : Listening on port 6666 for tcl connections
Info : Listening on port 4444 for telnet connections
```

- 2、运行gdb。

进入目录，打开另一个terminal，输入：

```
$ riscv64-unknown-elf-gdb hello_world -eval-command="target remote:3333"
```

其中：

- **hello_world**是编译过程生成的elf文件。
- 3333是openocd监听的端口号。

成功连接openocd后，就可以使用`load`、`break`、`continue`等gdb命令来调试了。

注: gdb手册参考 [gdb website](#)。

3.5 下载

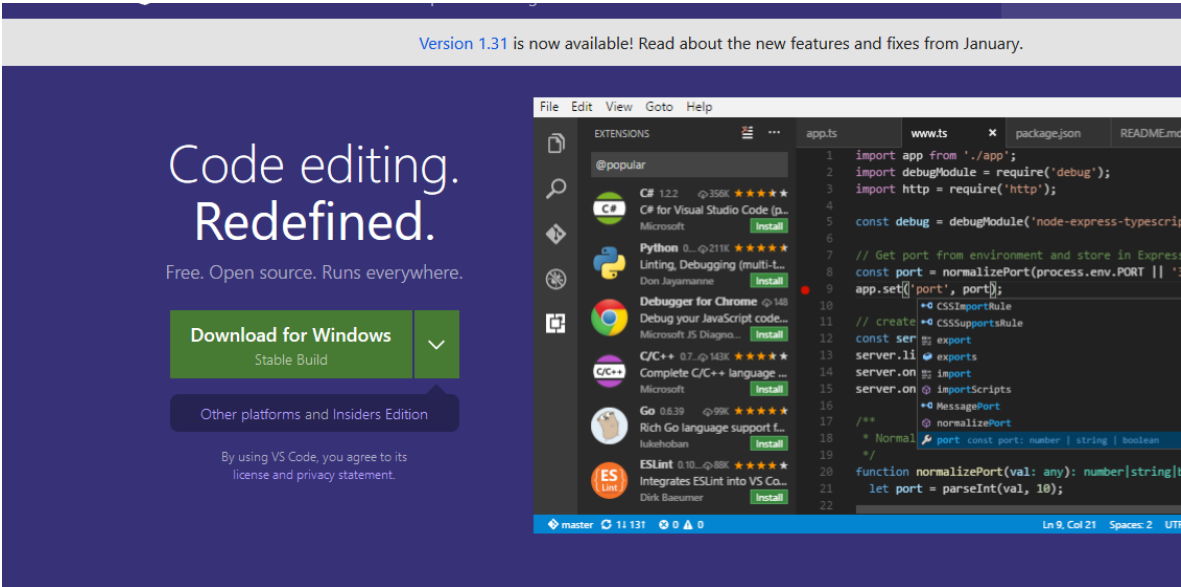
- 1、从[Kendryte Github](#)下载kflash.py脚本。
- 2、下载bin文件到K210.

```
$ sudo python3 kflash.py kendryte-standalone-sdk/build/hello_world.bin
```

4、Visual studio Code 开发环境搭建

4.1 安装Vscode

下载地址：<https://code.visualstudio.com/>



4.2 安装cmake

下载地址：<https://cmake.org/download/>

The release was packaged with CPack which is included as part of the release. The .sh files are self extracting gzipped tar files. To install a .sh file, run it with /bin/sh and follow the directions. The OS-machine.tar.gz files are gzipped tar files of the install tree. The OS-machine.tar.Z files are compressed tar files of the install tree. The tar file distributions can be untared in any directory. They are prefixed by the version of CMake. For example, the Linux-x86_64 tar file is all under the directory cmake-Linux-x86_64. This prefix can be removed as long as the share, bin, man and doc directories are moved relative to each other. To build the source distributions, unpack them with zip or tar and follow the instructions in Readme.txt at the top of the source tree. See also the [CMake 3.14 Release Notes](#). Source distributions:

| Platform | Files |
|---------------------------------------|---|
| Unix/Linux Source (has \n line feeds) | cmake-3.14.0-rc2.tar.gz cmake-3.14.0-rc2.tar.Z |
| Windows Source (has \r\n line feeds) | cmake-3.14.0-rc2.zip |

Binary distributions:

| Platform | Files |
|---|---|
| Windows win64-x64 Installer: Installer tool has changed. Uninstall CMake 3.4 or lower first! | cmake-3.14.0-rc2-win64-x64.msi |
| Windows win64-x64 ZIP | cmake-3.14.0-rc2-win64-x64.zip |
| Windows win32-x86 Installer: Installer tool has changed. Uninstall CMake 3.4 or lower first! | cmake-3.14.0-rc2-win32-x86.msi |
| Windows win32-x86 ZIP | cmake-3.14.0-rc2-win32-x86.zip |
| Mac OS X 10.7 or later | cmake-3.14.0-rc2-Darwin-x86_64.dmg cmake-3.14.0-rc2-Darwin-x86_64.tar.gz |
| Linux x86_64 | cmake-3.14.0-rc2-Linux-x86_64.sh cmake-3.14.0-rc2-Linux-x86_64.tar.gz |



Download verification:

| Role | Files |
|------|-------|
|------|-------|

安装cmake，本示例安装到D盘根目录下。

查看

电脑 > 本地磁盘 (D:) > cmake > bin

| 名称 | 修改日期 | 类型 | 大小 |
|---|------------------|------|-----------|
|  cmake.exe | 2018/12/13 20:44 | 应用程序 | 7,606 KB |
|  cmake-gui.exe | 2018/12/13 20:44 | 应用程序 | 18,341 KB |
|  cmcldeps.exe | 2018/12/13 20:44 | 应用程序 | 798 KB |
|  cpack.exe | 2018/12/13 20:44 | 应用程序 | 7,394 KB |
|  ctest.exe | 2018/12/13 20:44 | 应用程序 | 8,133 KB |

4.3 安装工具链

下载地址: <https://github.com/kendryte/kendryte-gnu-toolchain/releases>

[Pull requests](#) [Issues](#) [Marketplace](#) [Explore](#)

[kendryte / kendryte-gnu-toolchain](#) [Watch](#) 17 [Star](#) 16 [Fork](#) 4

[Code](#) [Issues](#) 2 [Pull requests](#) 0 [Projects](#) 0 [Wiki](#) [Security](#) [Insights](#)

[Releases](#) [Tags](#)

Pre-release

v8.2.0-20190409

289ea18

Kendryte GNU Toolchain v8.2.0-20190409

vowstar released this on 10 Apr · 4 commits to develop since this release

Version

- Binutils 2.31.51
- GDB 8.2
- GCC 8.2.0











Build

- Linux

```
./configure --prefix=/opt/kendryte-toolchain --with-cmodel=medany --with-arch=rv64imafc --with-a
make -j8
```
- OSX

```
./configure --prefix=/usr/local/opt/kendryte-toolchain --with-cmodel=medany --with-arch=rv64imaf
make -j8
```
- Windows

Assets 10
























| | |
|--|---------|
|  kendryte-toolchain-osx-mojave-8.2.0-20190409.tar.bz2 | 46.3 MB |
|  kendryte-toolchain-osx-mojave-8.2.0-20190409.tar.xz | 21.4 MB |
|  kendryte-toolchain-ubuntu-amd64-8.2.0-20190409.tar.bz2 | 45.6 MB |
|  kendryte-toolchain-ubuntu-amd64-8.2.0-20190409.tar.xz | 19.2 MB |
|  kendryte-toolchain-win-amd64-8.2.0-20190409.tar.xz | 18.2 MB |
|  kendryte-toolchain-win-amd64-8.2.0-20190409.zip | 51.4 MB |
|  kendryte-toolchain-win-i386-8.2.0-20190409.tar.xz | 16.8 MB |
|  kendryte-toolchain-win-i386-8.2.0-20190409.zip | 48.6 MB |
|  Source code (zip) | |
|  Source code (tar.gz) | |

下载完成后，解压到任意目录，本示例是放到D盘根目录下。

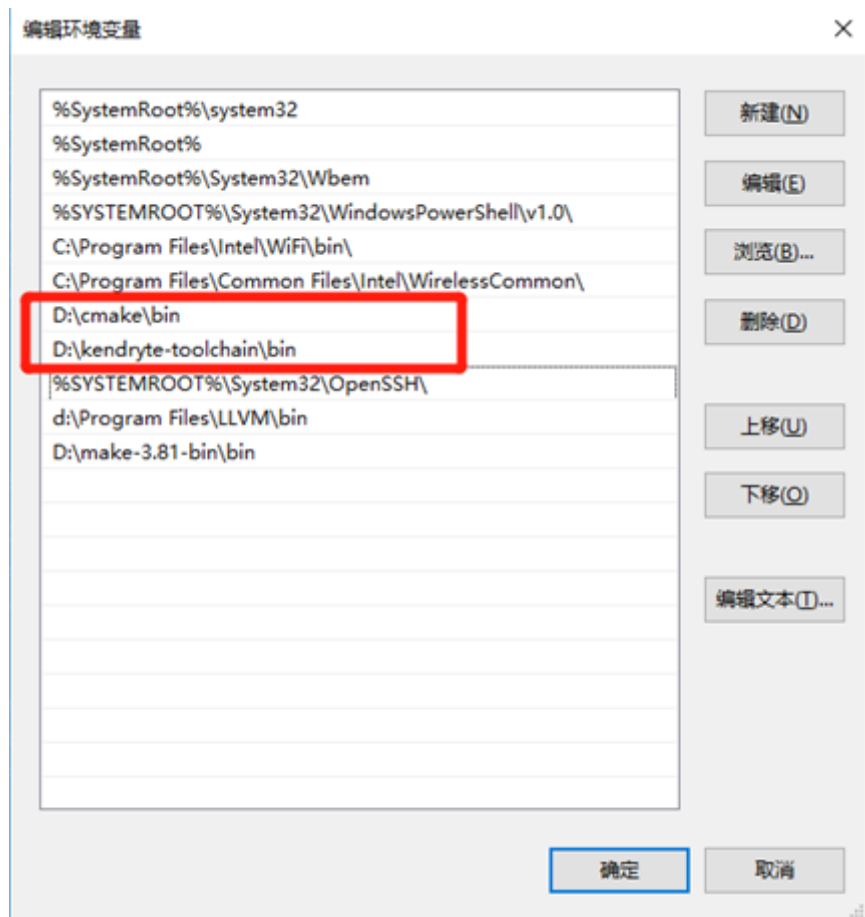
共享 查看

D:\kendryte-toolchain\bin

搜索"bin"

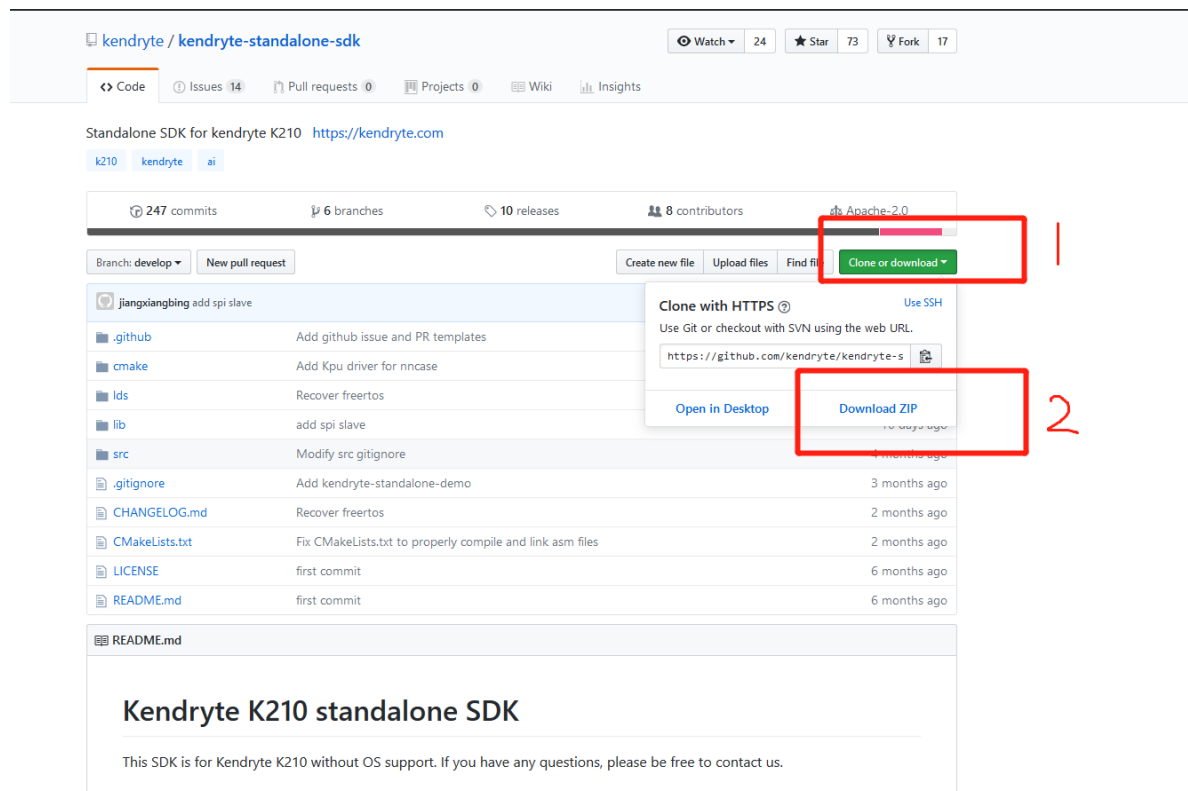
| 名称 | 修改日期 | 类型 | 大小 |
|---|-----------------|------|----------|
|  make.exe | 2019/4/11 17:10 | 应用程序 | 214 KB |
|  riscv64-unknown-elf-addr2line.exe | 2019/4/11 17:10 | 应用程序 | 775 KB |
|  riscv64-unknown-elf-ar.exe | 2019/4/11 17:10 | 应用程序 | 797 KB |
|  riscv64-unknown-elf-as.exe | 2019/4/11 17:10 | 应用程序 | 1,048 KB |
|  riscv64-unknown-elf-c++.exe | 2019/4/11 17:10 | 应用程序 | 1,000 KB |
|  riscv64-unknown-elf-c++filt.exe | 2019/4/11 17:10 | 应用程序 | 772 KB |
|  riscv64-unknown-elf-cpp.exe | 2019/4/11 17:10 | 应用程序 | 998 KB |
|  riscv64-unknown-elf-elfedit.exe | 2019/4/11 17:10 | 应用程序 | 62 KB |
|  riscv64-unknown-elf-g++.exe | 2019/4/11 17:10 | 应用程序 | 1,000 KB |
|  riscv64-unknown-elf-gcc.exe | 2019/4/11 17:10 | 应用程序 | 997 KB |
|  riscv64-unknown-elf-gcc-8.2.0.exe | 2019/4/11 17:10 | 应用程序 | 997 KB |
|  riscv64-unknown-elf-gcc-ar.exe | 2019/4/11 17:10 | 应用程序 | 58 KB |
|  riscv64-unknown-elf-gcc-nm.exe | 2019/4/11 17:10 | 应用程序 | 58 KB |
|  riscv64-unknown-elf-gcc-ranlib.exe | 2019/4/11 17:10 | 应用程序 | 58 KB |
|  riscv64-unknown-elf-gcov.exe | 2019/4/11 17:10 | 应用程序 | 683 KB |
|  riscv64-unknown-elf-gcov-dump.exe | 2019/4/11 17:10 | 应用程序 | 561 KB |
|  riscv64-unknown-elf-gcov-tool.exe | 2019/4/11 17:10 | 应用程序 | 613 KB |
|  riscv64-unknown-elf-gdb.exe | 2019/4/11 17:10 | 应用程序 | 5,322 KB |
|  riscv64-unknown-elf-gdb-add-index.... | 2019/4/11 15:34 | 应用程序 | 4 KB |
|  riscv64-unknown-elf-gprof.exe | 2019/4/11 17:10 | 应用程序 | 837 KB |
|  riscv64-unknown-elf-ld.bfd.exe | 2019/4/11 17:10 | 应用程序 | 1,168 KB |
|  riscv64-unknown-elf-ld.exe | 2019/4/11 17:10 | 应用程序 | 1,168 KB |
|  riscv64-unknown-elf-nm.exe | 2019/4/11 17:10 | 应用程序 | 786 KB |

4.4 设置环境变量



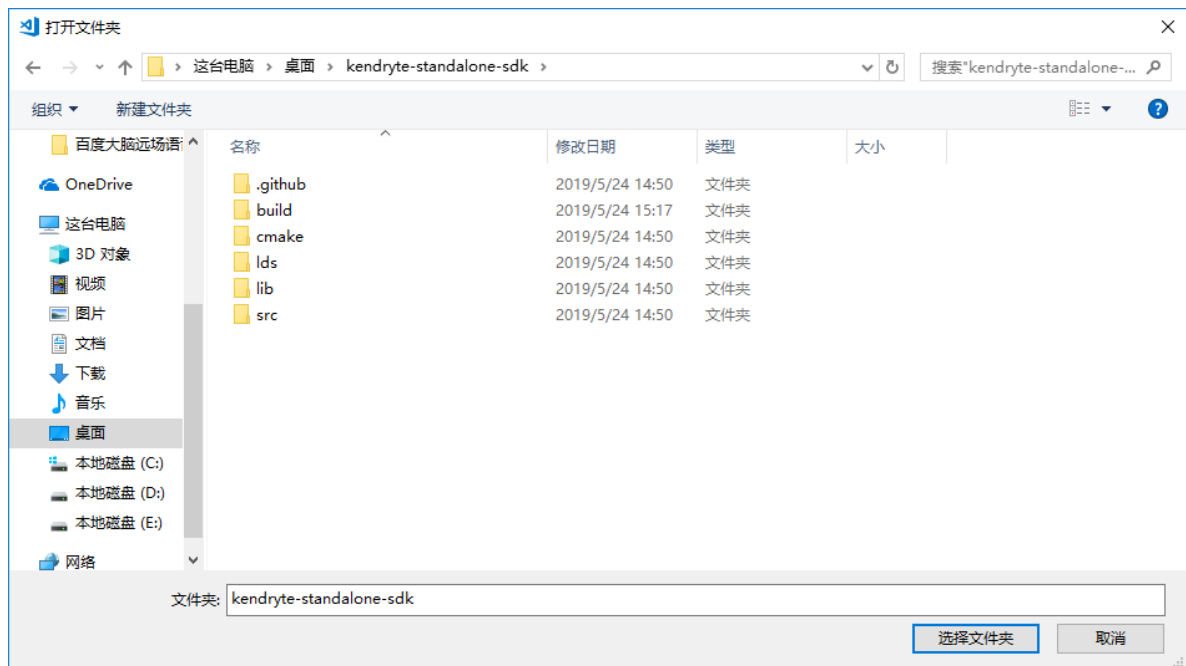
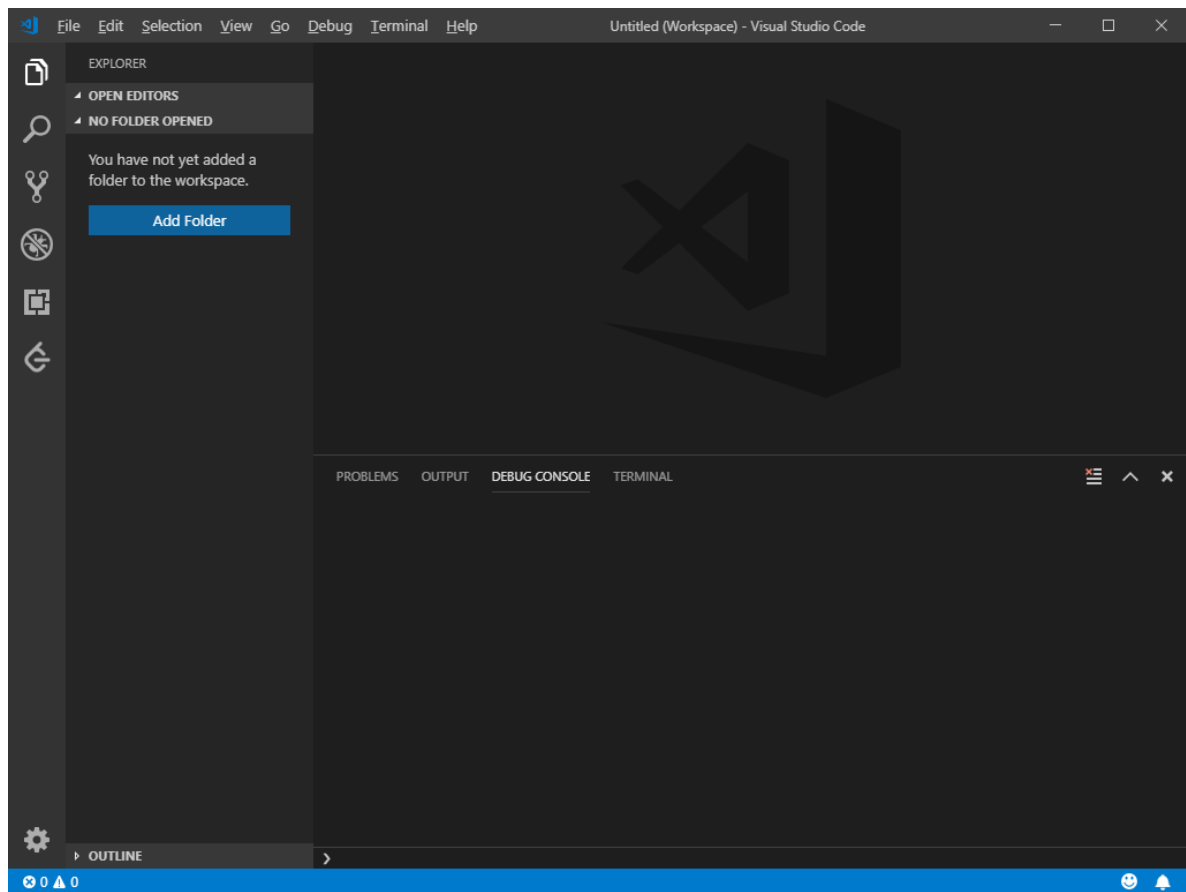
4.5 下载sdk

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

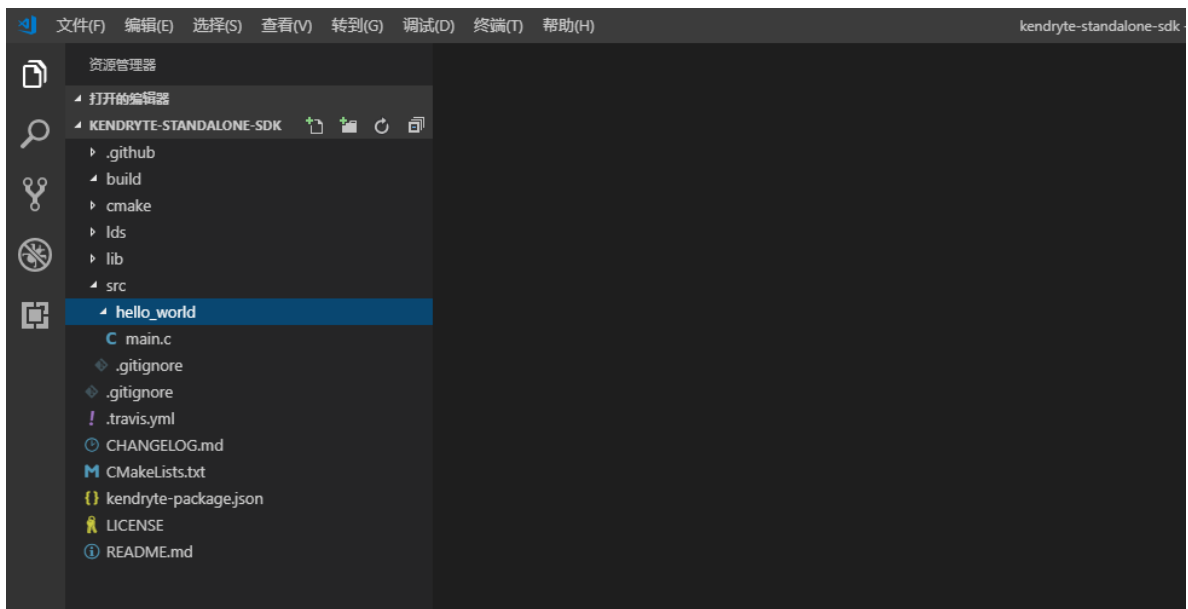


4.6 编译

打开VSCode, 点击左边窗口的“Add Folder”添加kendryte-standalone-sdk。如下图:

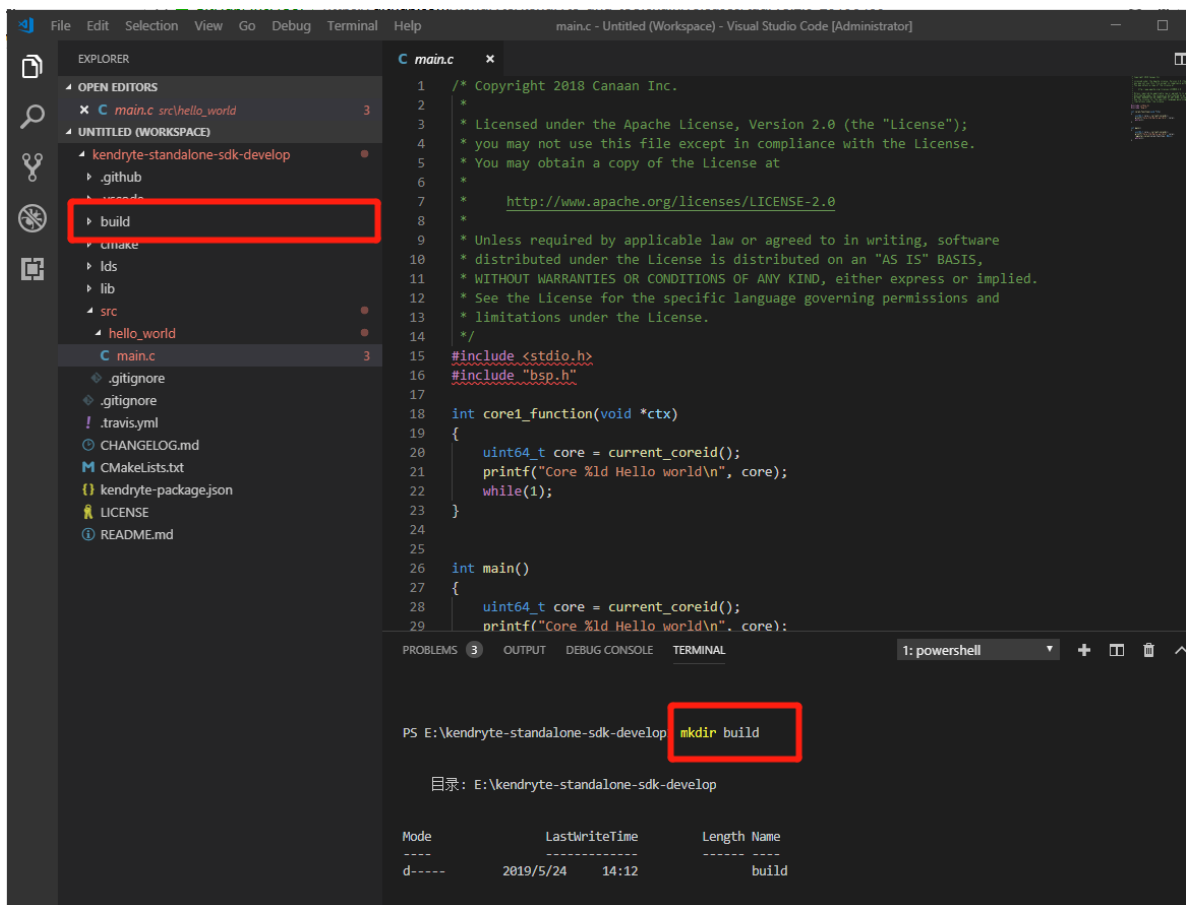


成功添加工程后的界面：

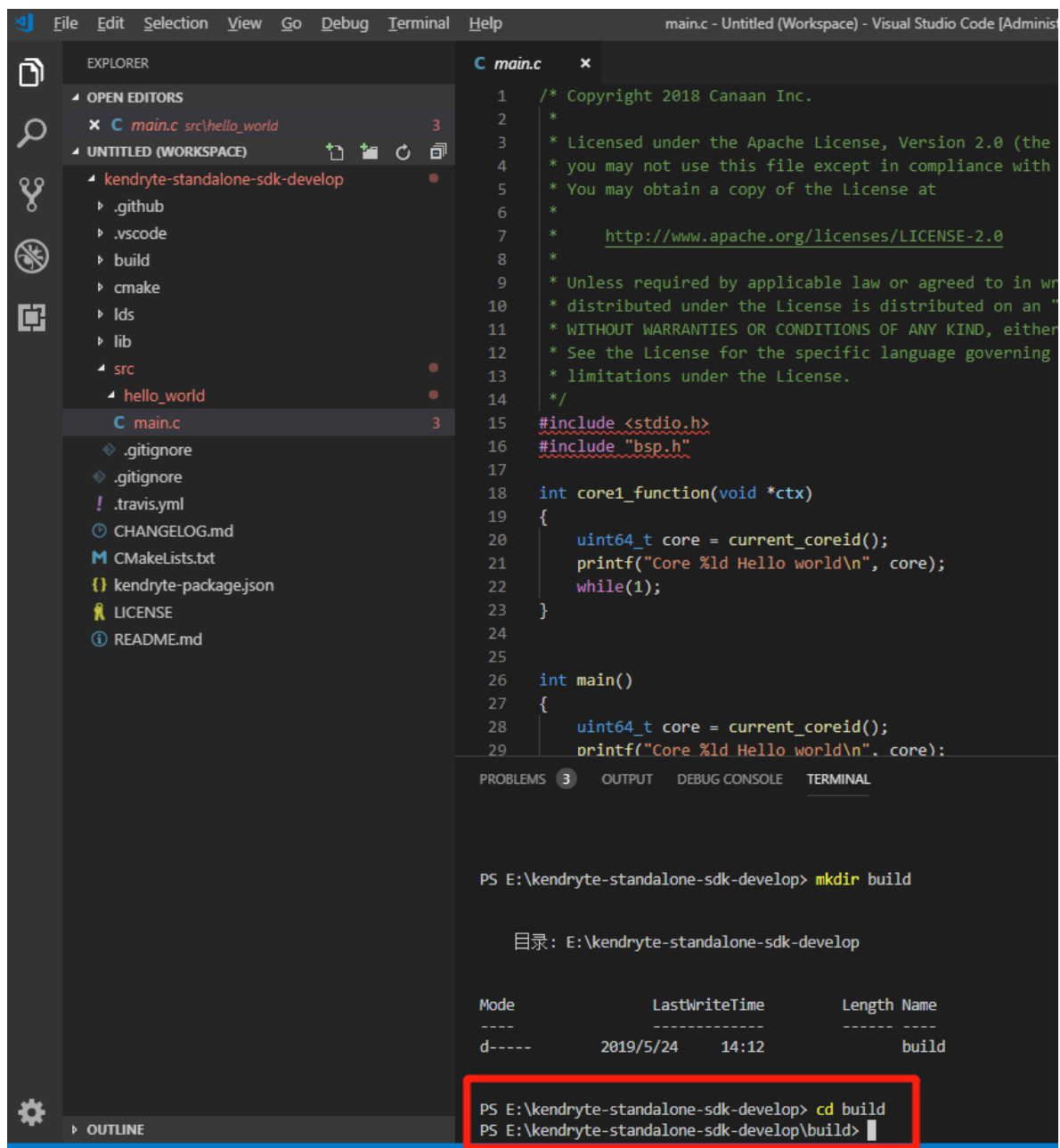


按键盘ctrl+shift + ~，下方出现命令终端。

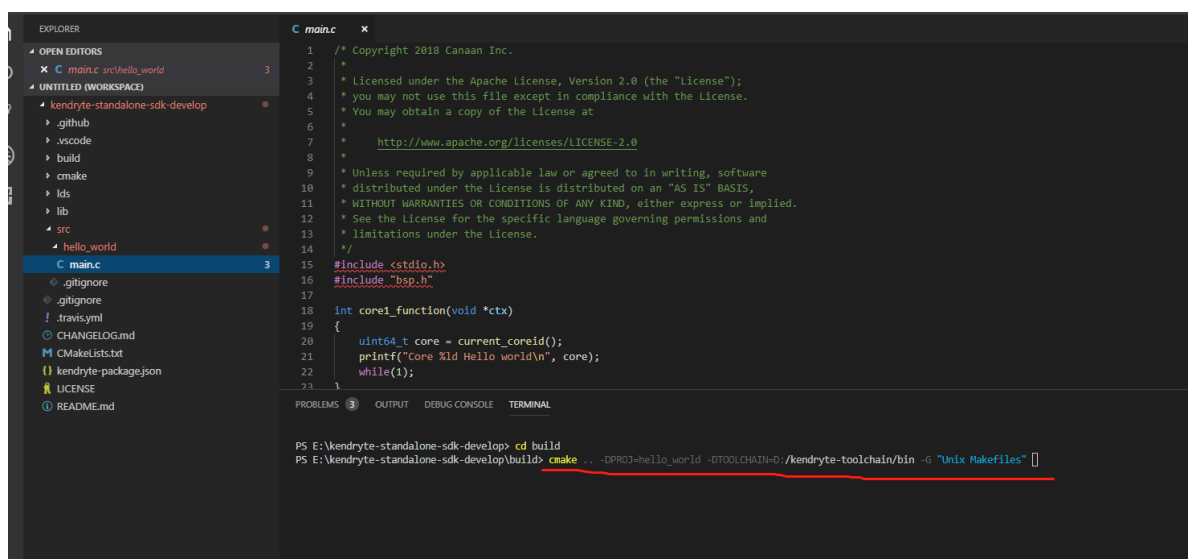
输入命令 mkdir build，回车，创建build文件夹。



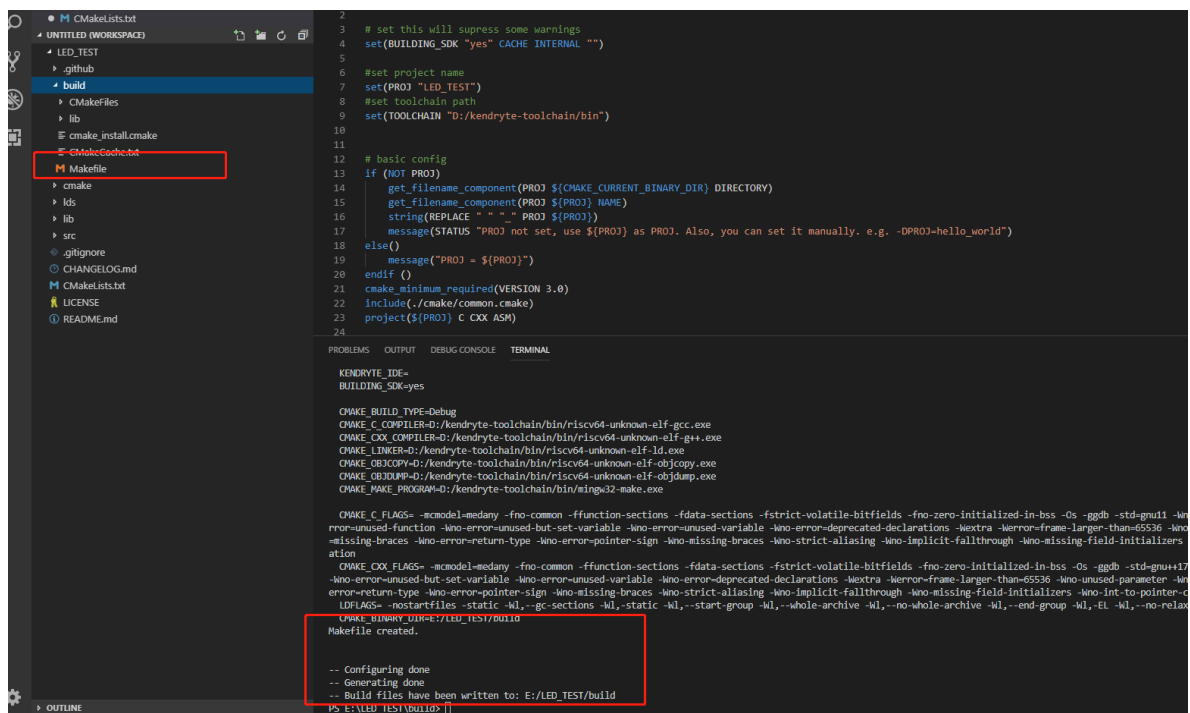
cd build 回车，进入build文件夹内。



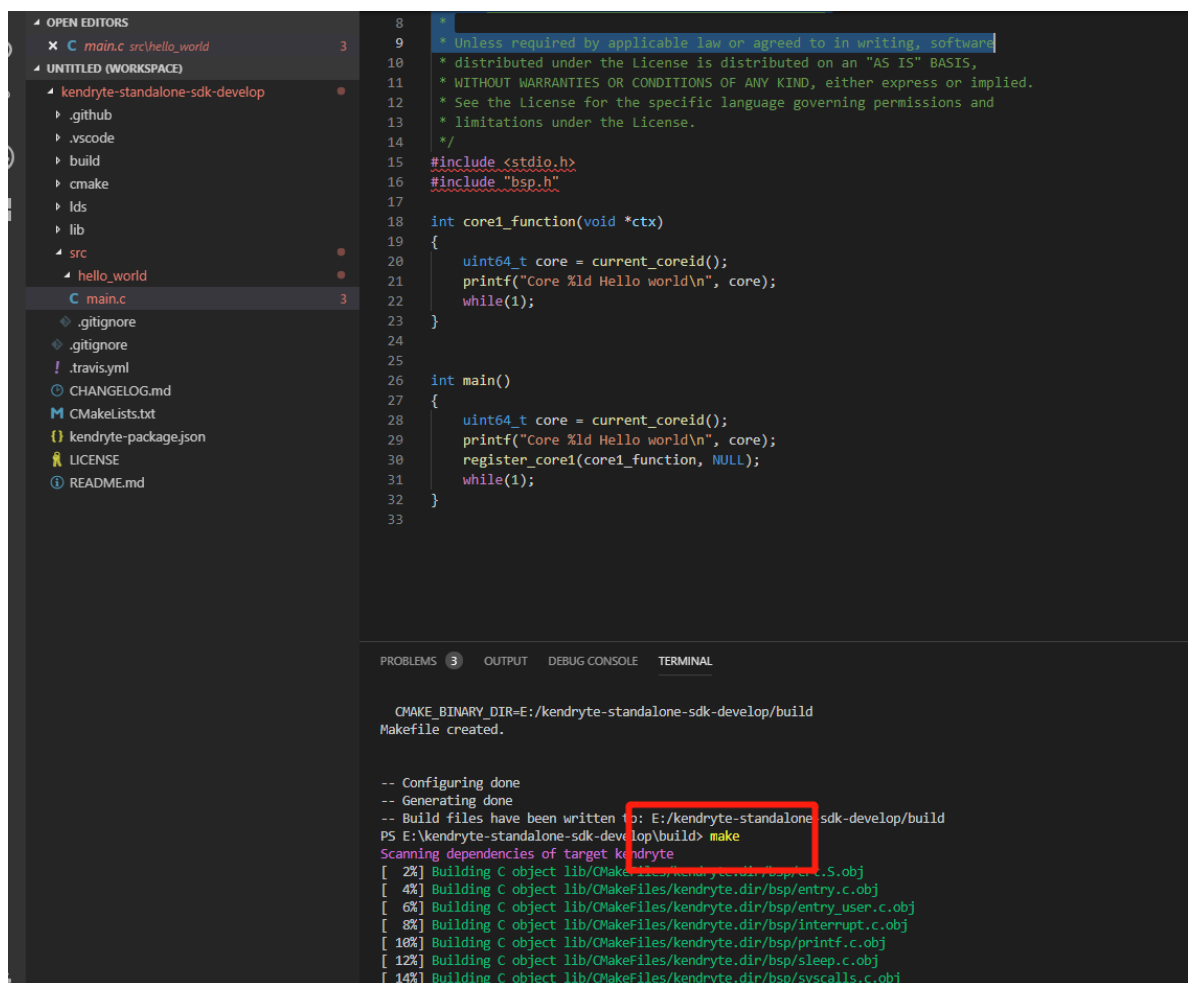
输入 `cmake .. -DPROJ=hello_world -G "Unix Makefiles"` 回车。



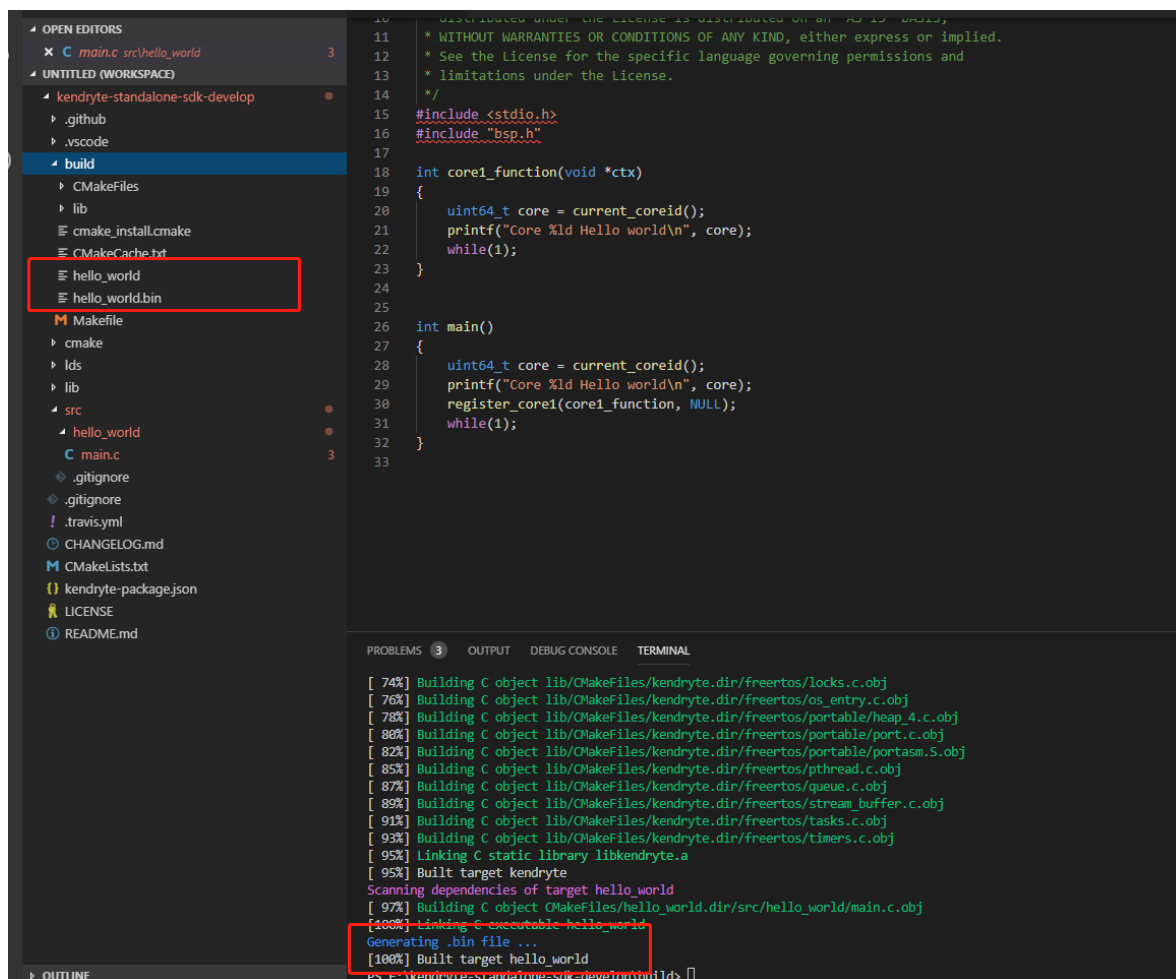
此时，在根目录下build文件中，makefile文件已创建。



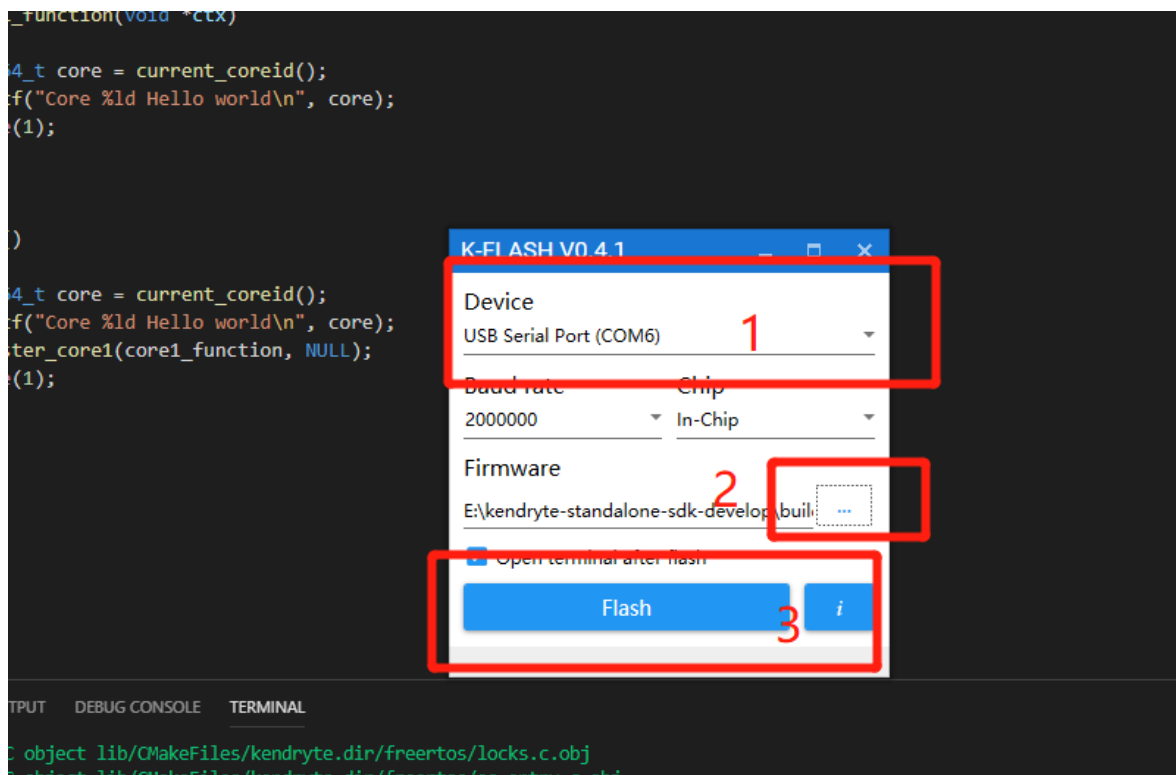
继续输入命令make，然后回车。



完成编译后，将生成hello_world.bin，hello_world.bin为可直接烧写的文件，如下图：



打开K-Flash工具，加载刚才生成的bin文件,点击“Flash”等待完成即可。



下载完成后，自动弹出的命令行窗口会直接打印“Core 0 hello_wolrd”。

