



K210

命令行开发环境搭建指南



KENDRYTE

勘智

嘉楠科技 版权©2020
KENDRYTE.COM



关于本手册

本文介绍了如何在Ubuntu和Windows下搭建[Kendryte K210](#)的命令行开发环境。

发布说明

日期	版本	发布说明
2020.1.20	V0.1	首次发布

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. Ubuntu

1.1 安装工具链

1. 安装 `build-essential` 以获取 `make` 工具

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

2. 安装 `cmake`

```
$ sudo apt install cmake
```

3. 下载 [Kendryte GNU Toolchain](#) 到 `/opt` 下并解压缩，这里以 `kendryte-toolchain-ubuntu-amd64-8.2.0-20190213.tar.gz` 为例。

```
$ sudo mv kendryte-toolchain-ubuntu-amd64-8.2.0-20190213.tar.gz /opt
$ cd /opt
$ sudo tar -zxvf kendryte-toolchain-ubuntu-amd64-8.2.0-20190213.tar.gz
```

打开 `~/.bashrc`，在末尾添加如下语句，将 `/opt/kendryte-toolchain/bin` 添加到环境变量 `PATH`。

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

然后使该修改生效。

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

1.2 编译

1. 下载 [kendryte-standalone-sdk](#)

```
$ git clone git@github.com:kendryte/kendryte-standalone-sdk.git
```

2. 将源文件目录，例如 `hello_world`，放在 `kendryte-standalone-sdk/src` 下。
3. 进入 `kendryte-standalone-sdk` 目录，新建 `build` 目录并进入。
4. 运行 `cmake`

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

5. 运行 `make`

```
$ make
```

1.3 调试

K210支持的调试器包括JLink、FT232或FT2232。

1. 安装openocd

下载[Kendryte openocd](#)到/opt下并解压缩，这里以 `kendryte-openocd-0.2.3-ubuntu64.tar.gz` 为例。

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

注：在使用openocd之前，可能需要安装一些依赖。

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

2. 运行openocd

调试器连接到PC和目标板后，执行如下命令：

```
$ cd /opt/kendryte-openocd
$ ./bin/openocd -f ./tcl/kendryte.cfg
Open On-Chip Debugger 0.10.0+dev-ga938338 (2018-09-12-11:27)
Licensed under GNU GPL v2
For bug reports, read
    http://openocd.org/doc/doxygen/bugs.html
adapter speed: 3000 kHz
Info : J-Link Ultra V4 compiled Apr 20 2018 16:46:34
Info : Hardware version: 4.00
Info : VTarget = 1.811 V
Info : clock speed 3000 kHz
Info : JTAG tap: riscv.cpu tap/device found: 0x04e4796b (mfg: 0x4b5
(<unknown>), part: 0x4e47, ver: 0x0)
Info : Examined RISCv core; XLEN=64, misa=0x800000000014112d
Info : Listening on port 3333 for gdb connections
Info : [0] Found 4 triggers
halted at 0x80008b74 due to debug interrupt
Info : Listening on port 6666 for tcl connections
Info : Listening on port 4444 for telnet connections
```

注：`./tcl/kendryte.cfg` 是JLink的配置文件，如果使用FT232或者FT2232，配置文件参考文末附录。

3. 运行gdb

打开另一个terminal窗口，进入 `kendryte-standalone-sdk/build` 目录，然后执行：

```
$ riscv64-unknown-elf-gdb hello_world --eval-command="target remote
127.0.0.1:3333"
```

其中：

- o `hello_world` 是编译生成的elf文件。

- 127.0.0.1 表示本地IP地址，如果openocd运行在远程机器上，这里可以指定远程机器的IP地址。
- 3333 为openocd监听的端口号。

在成功连接到openocd server后，就可以使用如load、break、continue等的命令来开始调试了。

注：gdb的使用方法参考[gdb手册](#)。

1.4 程序下载

1. 下载安装脚本工具[kflash.py](#)
2. 下载二进制代码到K210

程序下载使用K210的UARTHS接口，详见《K210硬件设计指南》的“启动模式”章节。下载前需要K210进入ISP模式，即BOOT拉低时复位K210。官方开发板已经对此进行了硬件实现，如果没有硬件实现，则需要手动执行这个操作。

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

注：-p 选项可以用来选择下载端口，-b 选项可以用来设置下载波特率。

2. Windows

2.1 安装工具链

1. 下载Windows版本[cmake](#)，选择不低于V3.0的版本。这里以V3.12.2为例。安装cmake到 `cmake-3.12.2-win64-x64` 目录，并将 `cmake-3.12.2-win64-x64\bin` 目录添加到环境变量 `PATH`。
2. 下载[Kendryte GNU Toolchain](#)并解压缩，将 `kendryte-toolchain\bin` 目录添加到环境变量 `PATH`。

2.2 编译

1. 下载[kendryte-standalone-sdk](#)
2. 将源文件目录，例如 `hello_world`，放在 `kendryte-standalone-sdk/src` 下。
3. 进入 `kendryte-standalone-sdk` 目录，新建 `build` 目录并进入。
4. 运行cmake

```
> cmake .. -DPROJ=hello_world -DTOOLCHAIN="xxxx/kendryte-toolchain/bin" -G
"Unix Makefiles"
```

其中 `xxxx` 是工具链的解压目录。

5. 运行make

```
> make
```

2.3 调试

K210支持的调试器包括JLink、FT232或FT2232。

1. 安装openocd

- 1) 下载[Kendryte openocd](#)并解压缩。

2) 插上调试器后，运行 `kendryte-openocd\tool` 下的 `zadig-2.4.exe` 工具，Options->List all devices，将调试器驱动更换为WinUSB驱动。

2. 运行openocd

调试器连接到PC和目标板后，使用cmd或powershell进入 `kendryte-openocd` 目录，然后执行：

```
> .\bin\openocd.exe -f .\tc1\kendryte.cfg
```

注：`.\tc1\kendryte.cfg` 是JLink的配置文件，如果使用FT232或者FT2232，配置文件参考文末附录。

3. 运行gdb

打开另外一个cmd或者powershell窗口，进入 `kendryte-standalone-sdk\build` 目录，然后执行：

```
> riscv64-unknown-elf-gdb hello_world --eval-command="target remote 127.0.0.1:3333"
```

其中：

- `hello_world` 是编译生成的elf文件。
- `127.0.0.1` 表示本地IP地址，如果openocd运行在远程机器上，这里可以指定远程机器的IP地址。
- `3333` 为openocd监听的端口号。

在成功连接到openocd server后，就可以使用如load、break、continue等的命令来开始调试了。

注：gdb的使用方法参考[gdb手册](#)。

2.4 程序下载

1. 下载Windows版本[K-Flash](#)工具。
2. 运行K-Flash.exe，选择端口和固件，点击“Flash”下载。

程序下载使用K210的UARTHS接口，详见《K210硬件设计指南》的“启动模式”章节。下载前需要K210进入ISP模式，即BOOT拉低时复位K210。官方开发板已经对此进行了硬件实现，如果没有硬件实现，则需要手动执行这个操作。

附录

```
# FT232或FT2232的openocd配置文件
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
```