

# Kendryte K210 开发板 使用说明书

V0.1.0

## 修改历史

日期	修改原因	版本

Kendryte.com

## 目录

1. 引言.....	4
1.1. 编写目的.....	4
1.2. 缩略语.....	4
1.3. 参考文档.....	4
2. 开发板资源.....	5
3. SDK 布局.....	6
3.1. Standalone SDK.....	6
3.2. FreeRTOS SDK.....	7
4. 建立开发环境.....	7
4.1. 命令行开发.....	7
4.1.1. openocd and toolchain for ubuntu.....	7
4.1.2. openocd and toolchain for windows.....	11
4.1.3. openocd for windows and toolchain for ubuntu.....	12
4.2. IDE 开发.....	12
5. 执行例程.....	12
5.1. 命令行方式.....	12
5.1.1. ubuntu 版.....	12
5.1.2. windows 版.....	14
5.2. IDE 方式.....	16
5.2.1. 界面简介.....	16
5.2.2. 编译、下载、调试.....	16
6. ISP 下载.....	17

# 1. 引言

## 1.1. 编写目的

本文档是 Kendryte K210 开发板使用说明书，型号是 KD233，描述了开发板资源、建立软件开发环境、使用 SDK 在开发板上执行例程。

本文档适用于使用 Kendryte K210 芯片的开发者。

## 1.2. 缩略语

SDK: Software Development Kit

IDE: Integrated Development Environment

## 1.3. 参考文档

《Kendryte K210 开发板电路原理图 V0.2》

## 2. 开发板资源

如图 1 所示，是 KD233 开发板的标示图，其中整板采用 USB Type-C 接口供电，以及 ISP 下载，采用 JTAG 进行 Debug，主控芯片为 Kendryte K210。开发板可以外接 WIFI 模块、麦克风阵列。

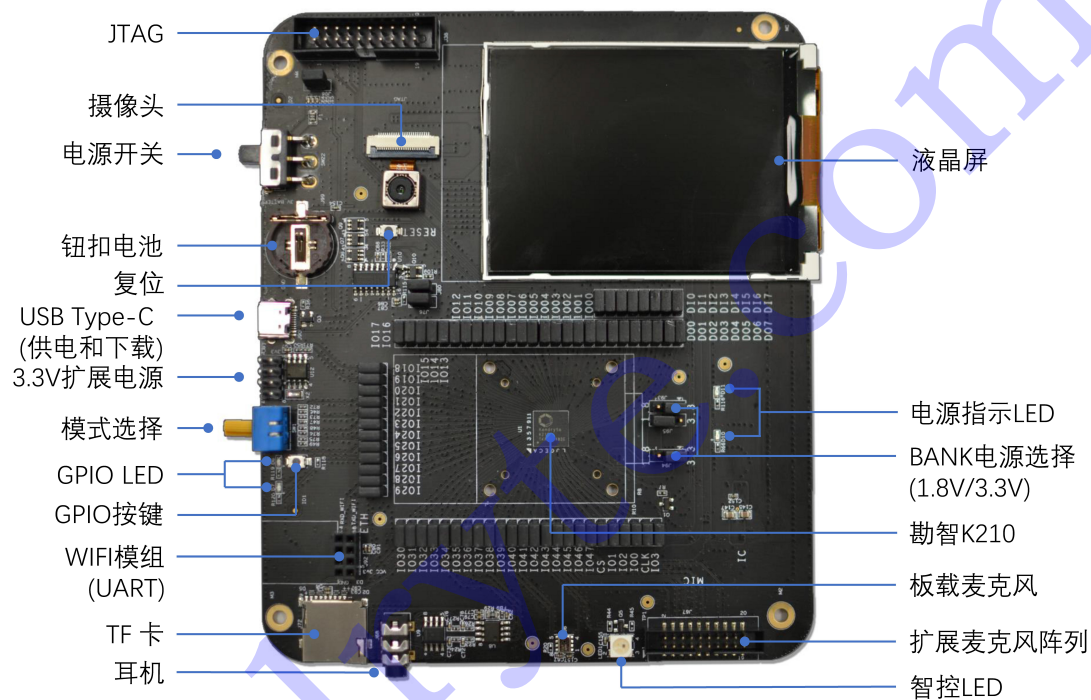


图 1 KD233 开发板图

### 3. SDK 布局

登陆 <https://kendryte.com/downloads/> 站点，下载 SDK 包，并解压，如图 2 所示。

SDK			
文档名称	版本	修改时间	下载
Kendryte K210 FreeRTOS SDK	V0.2.0	2018-09-20	
Kendryte K210 Standalone SDK	V0.2.1	2018-09-22	

HDK			
文档名称	版本	修改时间	下载

Documentations			
文档名称	版本	修改时间	下载
Kendryte K210 datasheet	V0.1.5	2018-09-19	

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	V7.2.0	2018-09-13	
RISC-V 64bit toolchain for Kendryte K210_ubuntu_amd64	V7.2.0	2018-09-13	

图 2 SDK 下载页面

#### 3.1. Standalone SDK

名称	修改日期	类型	大小
cmake	2018/9/22 13:51	文件夹	
lds	2018/9/22 13:51	文件夹	
lib	2018/9/22 13:51	文件夹	
src	2018/9/22 13:51	文件夹	
CHANGELOG.md	2018/9/22 13:51	文本文档	1 KB
CMakeLists	2018/9/22 13:51	MD 文件	1 KB
LICENSE	2018/9/22 13:51	文本文档	1 KB
README.md	2018/9/22 13:51	文件	12 KB
	2018/9/22 13:51	MD 文件	1 KB

图 3 Standalone SDK 目录

如图 3 所示，是 Standalone SDK 的目录，其中

- cmake ----- 编译工具。
- lds ----- 链接文件。
- lib ----- 包含 bsp、drivers、firmware 等。
- src ----- 测试例程及 Demo。

lib 目录如下图 4 所示：

名称	修改日期	类型	大小
bsp	2018/9/22 13:51	文件夹	
drivers	2018/9/22 13:51	文件夹	
firmware	2018/9/22 13:51	文件夹	
math	2018/9/22 13:51	文件夹	
utils	2018/9/22 13:51	文件夹	
CMakeLists	2018/9/22 13:51	文本文档	1 KB

图 4 Standalone SDK\lib 目录

bsp ----- 芯片启动初始化相关部分  
 drivers ----- Soc drivers  
 firmware ----- 外设 drivers

## 3.2. FreeRTOS SDK

FreeRTOS SDK 解压后与 Standalone SDK 的目录结构相同，如图 3 所示。其中 FreeRTOS SDK\lib 目录下略有不同，详情如下图 5 所示。

名称	修改日期	类型	大小
bsp	2018/9/20 20:16	文件夹	
drivers	2018/9/20 20:16	文件夹	
freertos	2018/9/20 20:16	文件夹	
math	2018/9/20 20:16	文件夹	
utils	2018/9/20 20:16	文件夹	
CMakeLists	2018/9/20 20:16	文本文档	2 KB

图 5 FreeRTOS SDK\lib 目录

bsp ----- 芯片启动初始化相关部分，以及 FreeRTOS 调用对 drivers 的 API 封装。  
 drivers ----- Soc drivers  
 freertos ----- FreeRTOS 源文件

## 4. 建立开发环境

开发环境既可以采用命令行，也可以采用 IDE，推荐 IDE。命令行提供 3 种方式，任选其一。

### 4.1. 命令行开发

#### 4.1.1. openocd and toolchain for ubuntu

##### ➤ 准备文件

首先在 <https://kendryte.com/downloads/> 下载如下图 6 所示，四个文件。

SDK			
文档名称	版本	修改时间	下载
Kendryte K210 FreeRTOS SDK	V0.2.0	2018-09-20	<a href="#">下载</a>
Kendryte K210 Standalone SDK	V0.2.1	2018-09-22	<a href="#">下载</a>

HDK			
文档名称	版本	修改时间	下载

Documentations			
文档名称	版本	修改时间	下载
Kendryte K210 datasheet	V0.1.5	2018-09-19	<a href="#">下载</a>

Toolchain			
文档名称	版本	修改时间	下载
Kendryte OpenOCD for win32	V0.1.3	2018-09-13	<a href="#">下载</a>
Kendryte OpenOCD for Ubuntu x86_64	V0.1.3	2018-09-13	<a href="#">下载</a>
RISC-V 64bit toolchain for Kendryte K210_win32	V7.2.0	2018-09-13	<a href="#">下载</a>
RISC-V 64bit toolchain for Kendryte K210_ubuntu_amd64	V7.2.0	2018-09-13	<a href="#">下载</a>

图 6 下载开发环境文件

➤ 准备开发板

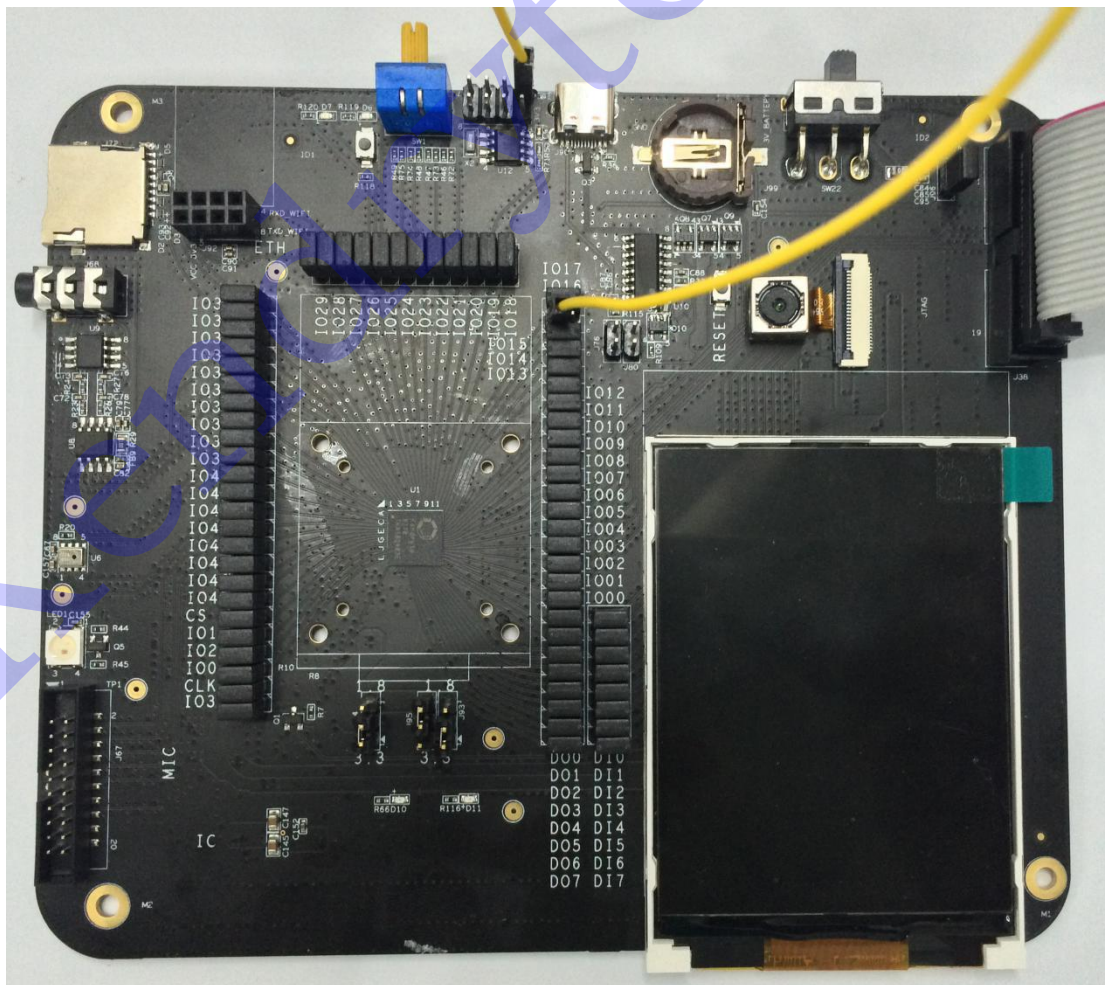


图 7 开发板连线图



如图 7 所示，使用 JLink 调试时，要把靠近 CPU 端的 IO16 与地 GND 连接，否则无法使用 JLink 调试。连接 USB 线和 JLink 线，打开电源开关，在命令行下输入：

```
lsusb
```

```
ls /dev |grep USB
```

查看串口设备号，如下图：

```

zzx@canaan-zzx:~/kendryte-openocd$ ls /dev |grep USB
ttyUSB0
zzx@canaan-zzx:~/kendryte-openocd$ lsusb
Bus 001 Device 037: ID 1366:0101 SEGGER J-Link PLUS
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 023: ID 1a86:7523 QinHeng Electronics HL-340 USB-Serial adapter
Bus 002 Device 022: ID 0e0f:0008 VMware, Inc.
Bus 002 Device 003: ID 0e0f:0002 VMware, Inc. Virtual USB Hub
Bus 002 Device 002: ID 0e0f:0003 VMware, Inc. Virtual Mouse
Bus 002 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
zzx@canaan-zzx:~/kendryte-openocd$

```

安装 screen，串口接收窗口。执行：

```
apt-get install screen
```

设置波特率。

```
sudo screen /dev/ttyUSB0 115200
```

#### ➤ 安装编译工具

将 kendryte-toolchain.tar.gz 移至/opt 目录下并解压，输入：

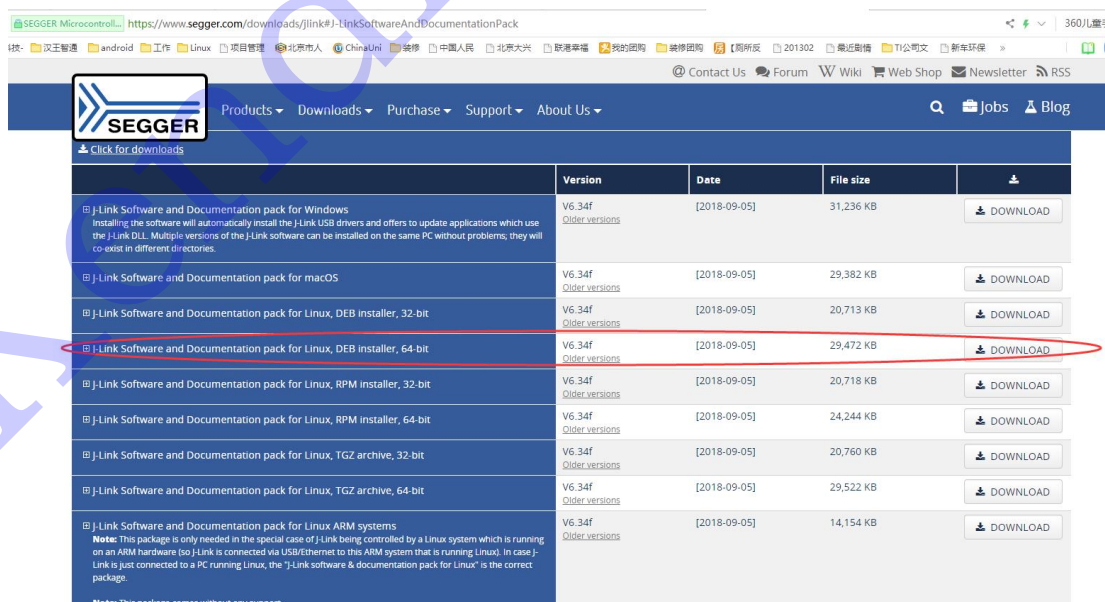
```
sudo mv xvf kendryte-toolchain.tar.gz /opt/
```

```
cd /opt/
```

```
sudo tar xvf kendryte-toolchain.tar.gz
```

#### ➤ 安装 JLink 驱动

在 SEGGER 官网 <https://www.segger.com/downloads/jlink> 下载 JLink 驱动，如下图：



安装 deb 软件包，输入命令：

```
sudo dpkg -i JLink_Linux_V634f_x86_64.deb
```

安装驱动后需要重新插拔一下 JLink 接口，再执行：

```
/opt/SEGGER/JLink/JLinkExe
```

查找 JLink 序列号，如下图所示：

```

zzx@canaan-zzx:~$ /opt/SEGGER/JLink/JLinkExe
SEGGER J-Link Commander V6.32d (Compiled May 28 2018 16:59:04)
DLL version V6.32d, compiled May 28 2018 16:58:57

Connecting to J-Link via USB...O.K.
Firmware: J-Link Ultra V4 compiled Apr 20 2018 16:46:34
Hardware version: V4.00
S/N: 504404512
License(s): RDI, FlashBP, FlashDL, JFlash, GDB
VTref=3.293V

Type "connect" to establish a target connection, '?' for help
J-Link>q
zzx@canaan-zzx:~$

```

➤ 安装 openocd

sudo apt install libusb-dev -y

tar xvf kendryte-openocd-0.1.3-ubuntu64.tar.gz

修改 tcl/openocd.cfg 中 JLink 的序列号。具体操作可详见 readme.txt，如下图所示：

```

终端 文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
jlink serial 000504401724

transport select jtag
adapter_khz 3000

# server port
gdb_port 3333
telnet_port 4444

# add cpu target
set _CHIPNAME riscv

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

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

# command
init
halt
zzx@canaan-zzx:~/kendryte-openocd$

```

在 kendryte-openocd 目录下执行：

./bin/openocd -f tcl/openocd.cfg

如下图所示：

```

zzx@canaan-zzx:~/kendryte-openocd$ ./bin/openocd -f tcl/openocd.cfg
Kendryte Open On-Chip Debugger For RISC-V v0.1.3 (20180912)
Licensed under GNU GPL v2
adapter speed: 3000 kHz
Info : J-Link Ultra V4 compiled Apr 20 2018 16:46:34
Info : Hardware version: 4.00
Info : VTarget = 3.293 V
Info : clock speed 3000 kHz
Info : JTAG tap: riscv.cpu tap/device found: 0x04e4796b (mfg: 0x4b5 (<unknown>), part: 0x4e47, ve
r: 0x0)
Info : [0] Found 4 triggers
Info : [1] Found 4 triggers
[1] halted at 0x800002aa 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 0x800002aa due to debug interrupt
[0] halted at 0x800002aa due to debug interrupt
Info : Listening on port 6666 for tcl connections
Info : Listening on port 4444 for telnet connections

```

至此，openocd 安装成功。openocd and toolchain for ubuntu 的开发环境建立完毕。

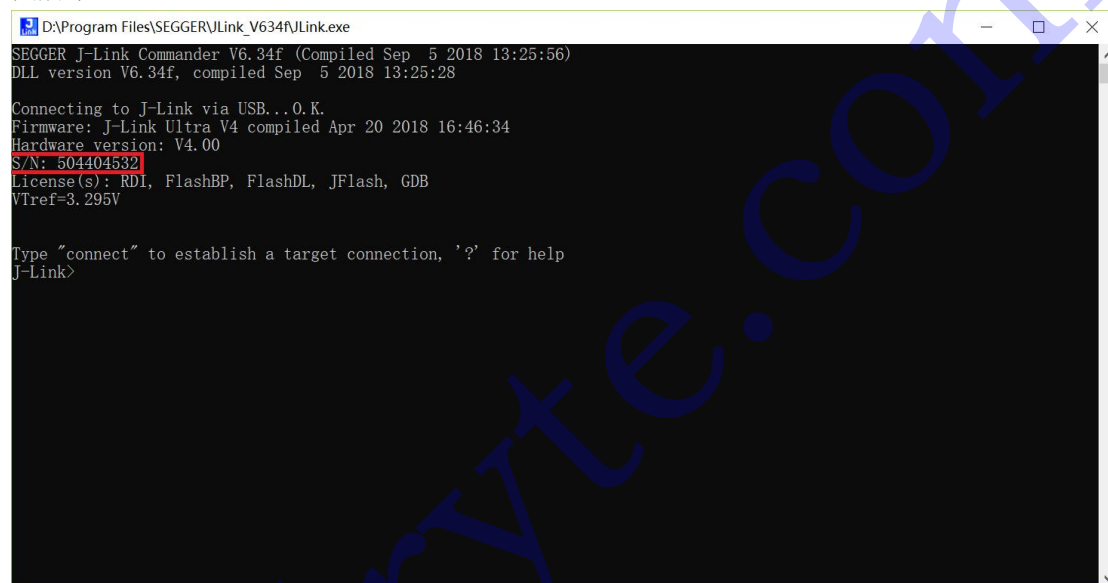
## 4.1.2.openocd and toolchain for windows

### ➤ 安装 OpenOCD

下载安装“J-Link Software and Documentation pack for Windows”

(<https://www.segger.com/downloads/jlink/#J-LinkSoftwareAndDocumentationPack>)

通过 USB 接口连接 J-Link 到计算机，运行上述软件安装目录下的 Jlink.exe，查看 S/N。如下图所示：



下载“Kendryte OpenOCD for win32”(<https://kendryte.com/downloads/>)，解压到指定目录，例如：D:\Program Files。打开 D:\Program Files\kendryte-openocd\tcl 目录下的配置文件 openocd.cfg，将其中的 SN 修改为步骤 2 中查看到的值。如下图所示：

```

1  # debug adapter
2  interface jlink
3  jlink serial 000504404532
4
5  transport select jtag
6  adapter_khz 3000
7
8  # server port
9  gdb_port 3333
10 telnet_port 4444
11
12 # add cpu target
13 set _CHIPNAME riscv
14
15 jtag newtap $_CHIPNAME cpu -irlen 5 -expected-id 0x04e4796b
16
17 set _TARGETNAME $_CHIPNAME.cpu
18 target create $_TARGETNAME riscv -chain-position $_TARGETNAME
19
20 # command
21 init
22 halt

```

运行 D:\Program Files\kendryte-openocd\tool 下的 zadig-2.4.exe，点击 Options->List

All Devices，在下拉列表中选中 J-Link，将其驱动转换为 WinUSB。

➤ 安装交叉编译 Toolchain

下载安装 cmake，版本不低于 V3.0，建议安装最新版本(<https://cmake.org/download/>)。例如 cmake 安装在 D:\Program Files\cmake-3.12.2-win64-x64 下，则将 D:\Program Files\cmake-3.12.2-win64-x64\bin 加入系统环境变量 Path 中。

下载“RISC-V 64bit toolchain for Kendryte K210\_win32”(<https://kendryte.com/downloads/>)，解压到指定目录，例如：D:\Program Files，将 D:\Program Files\kendryte-toolchain\bin 加入系统环境变量 Path 中。

### 4.1.3.openocd for windows and toolchain for ubuntu

按照上述方法，也可以采用 windows 版的 openocd 和 ubuntu 版的 toolchain 进行组合使用。

## 4.2.IDE 开发

（下载，解压，运行）待续。

## 5. 执行例程

执行第一个例程，Hello Word!

### 5.1. 命令行方式

#### 5.1.1.ubuntu 版

将下载好的 SDK 包进行解压，输入：

```
unzip kendryte-freertos-sdk-master.zip
```

进入 kendryte-freertos-sdk-master 下，输入：

```
cd ./kendryte-freertos-sdk-master
```

➤ 创建 build 目录

```
mkdir build
```

```
cd build
```

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

如下图所示：



```

-- Configuring incomplete, errors occurred!
zxx@Canaan-zxx:/mnt/d/kendryte-freertos-sdk-master/build$ cmake .. -DPROJ=hello_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 7.2.0
-- The CXX compiler identification is GNU 7.2.0
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
CMAKE_MAKE_PROGRAM=/usr/bin/make
TC_MAKE=
SOURCE_FILES=/mnt/d/kendryte-freertos-sdk-master/src/hello_world/main.c

Project: hello_world
LIST_FILE=/mnt/d/kendryte-freertos-sdk-master/cmake/executable.cmake
TOOLCHAIN=/opt/kendryte-toolchain/bin
KENDRYTE_IDE=
BUILDING_SDK=yes

CMAKE_BUILD_TYPE=Debug
CMAKE_C_COMPILER=/opt/kendryte-toolchain/bin/riscv64-unknown-elf-gcc
CMAKE_CXX_COMPILER=/opt/kendryte-toolchain/bin/riscv64-unknown-elf-g++
CMAKE_LINKER=/opt/kendryte-toolchain/bin/riscv64-unknown-elf-ld
CMAKE_OBJCOPY=/opt/kendryte-toolchain/bin/riscv64-unknown-elf-objcopy
CMAKE_MAKE_PROGRAM=/usr/bin/make

CMAKE_C_FLAGS= -mcmodel=medany -fno-common -ffunction-sections -fdata-sections -fstrict-volatile-bitfields -O0 -ggdb -std=gnu99 -Wno-error=unused-variable -Wno-error=deprecated-declarations -Wno-error=maybe-uninitialized -Wextra -Werror=frame-larger-than -Wno-sign-compare -Wno-error=missing-braces -Wno-old-style-declaration
CMAKE_CXX_FLAGS= -mcmodel=medany -fno-common -ffunction-sections -fdata-sections -fstrict-volatile-bitfields -O0 -ggdb -std=gnu99 -Wno-error=unused-variable -Wno-error=deprecated-declarations -Wno-error=maybe-uninitialized -Wextra -Werror=frame-larger-than -Wno-sign-compare -Wno-error=missing-braces
LDFLAGS= -nostartfiles -static -Wl,--gc-sections -Wl,-static -Wl,--start-group -Wl,--whole-archive -Wl,--no-whole-archive -Wl,-zrelro -Wl,-znow
CMAKE_BINARY_DIR=/mnt/d/kendryte-freertos-sdk-master/build
Makefile created.

-- Configuring done
-- Generating done
-- Build files have been written to: /mnt/d/kendryte-freertos-sdk-master/build

```

#### ➤ make

```

[ 95%] Linking C static library libkendryte.a
[ 95%] Built target kendryte
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
zxx@Canaan-zxx:/mnt/d/kendryte-freertos-sdk-master/build$

```

#### ➤ riscv64-unknown-elf-gdb hello\_world

然后输入：target remote 192.168.1.17:3333。

如果是本机调试输入：target remote :3333。

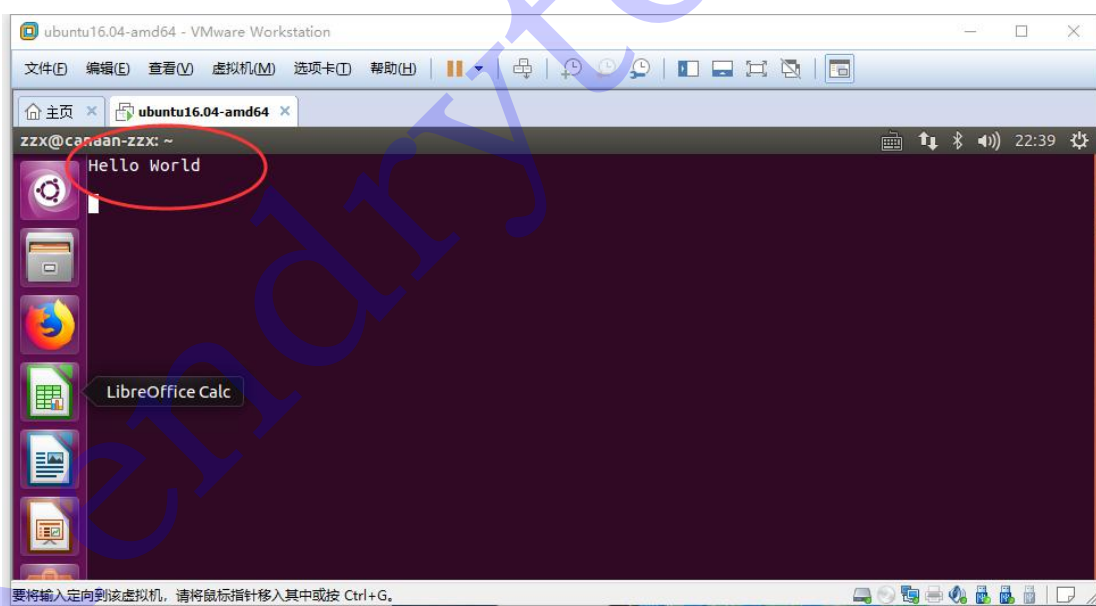
```

zzx@Canaan-zzx: /mnt/d/kendryte-freertos-sdk-master/build$ riscv64-unknown-elf-gdb hello_world
GNU gdb (GDB) 8.0.50.20170724-git
Copyright (C) 2017 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "--host=x86_64-pc-linux-gnu --target=riscv64-unknown-elf".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from hello_world...done
(gdb) target remote 192.168.1.17:3333
Remote debugging using 192.168.1.17:3333
0x000000008008be4 in ?? ()
(gdb) load
Loading section .text, size 0xae68 lma 0x80000000
Loading section .rodata, size 0x3818 lma 0x8000ae68
Loading section .init_array, size 0x8 lma 0x8000e680
Loading section .fini_array, size 0x8 lma 0x8000e688
Loading section .data, size 0x2330 lma 0x8000e690
Start address 0x80000000, load size 68032
Transfer rate: 74 KB/sec, 9718 bytes/write.
(gdb) c
Continuing.
(gdb) ^CThe target is not responding to interrupt requests.
Stop debugging it? (y or n) y
Disconnected from target.
(gdb) q
zzx@Canaan-zzx: /mnt/d/kendryte-freertos-sdk-master/build$ target remote 192.168.1.17:3333

```

输入：load

输入：c，运行程序，此时查看串口，收到 Hello World。



至此，第一个例程“Hello World”在开发板上运行成功，可以使用 gdb 方式调试程序。

## 5.1.2.windows 版

### ➤ 编译调试程序

以下说明以 Standalone SDK 为例，对 FreeRTOS SDK 的操作与此相同。使用 windows 命令行进入 D:\Program Files\kendryte-openocd 目录，启动 OpenOCD Server，如下图所示：

```
.\bin\openocd.exe -f .\tcl\openocd.cfg
```

```

C:\Windows\system32\cmd.exe - .\bin\openocd.exe -f .\tcl\openocd.cfg
Microsoft Windows [版本 10.0.17134.285]
(c) 2018 Microsoft Corporation. 保留所有权利。

C:\Users\wanghongzhu>cd "Program Files\kendryte-openocd"

D:\Program Files\kendryte-openocd>.bin\openocd.exe -f .\tcl\openocd.cfg
Kendryte Open On-Chip Debugger For RISC-V v0.1.3 (20180912)
Licensed under GNU GPL v2
adapter speed: 3000 kHz
Info : J-Link Ultra V4 compiled Apr 20 2018 16:46:34
Info : Hardware version: 4.00
Info : VTarget = 3.296 V
Info : clock speed 3000 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
Info : Examined RISC-V core; found 2 harts, XLEN=64, misa=0x800000000014112d
Info : Listening on port 3333 for gdb connections
[1] halted at 0x80003316 due to debug interrupt
[0] halted at 0x80006ccc due to debug interrupt
Info : Listening on port 6666 for tcl connections
Info : Listening on port 4444 for telnet connections

```

下载“Kendryte K210 Standalone SDK”（<https://kendryte.com/downloads/>），解压到指定目录，例如 E:\project。

打开另一个 windows 命令行窗口进入 E:\project\kendryte-standalone-sdk-0.2.1，创建并进入 build 目录

```
mkdir build
cd build
```

运行 cmake 生成 Makefile，其中 hello\_world 为要编译的工程名，其代码在\src 目录下  
 cmake .. -DPROJ=hello\_world -DTOOLCHAIN="D:/Program Files/kendryte-toolchain/bin" -G "MinGW Makefiles"

运行 mingw32-make 完成编译。

```
mingw32-make
```

➤ 启动 gdb 调试，如下图所示：

riscv64-unknown-elf-gdb hello\_world --eval-command="target remote 127.0.0.1:3333"

```

C:\Windows\system32\cmd.exe - riscv64-unknown-elf-gdb hello_world --eval-command="target remote 127.0.0.1:3333"
Generating .bin file ...
[100%] Built target hello_world

E:\project\kendryte-standalone-sdk-0.2.1\build>riscv64-unknown-elf-gdb hello_world --eval-command="target remote 127.0.0.1:3333"
Python Exception <type 'exceptions.ImportError'> No module named gdb:
D:\Program Files\kendryte-toolchain\bin\riscv64-unknown-elf-gdb.exe: warning:
Could not load the Python gdb module from 'e:\mingw64\share\gdb\python'.
Limited Python support is available from the _gdb module.
Suggest passing --data-directory=/path/to/gdb/data-directory.

GNU gdb (GDB) 8.0.50.20170724-git
Copyright (C) 2017 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "--host=x86_64-w64-mingw32 --target=riscv64-unknown-elf".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from hello_world...done.
Remote debugging using 127.0.0.1:3333
Python Exception <type 'exceptions.NameError'> Installation error: gdb.execute_unwinders function is missing:
0x0000000080006ccc in plic_instance ()
(gdb)

```

其中 hello\_world 为编译生成的 elf 文件；127.0.0.1 表示运行 OpenOCD 的设备的 IP 地址，如果与该示例相同，OpenOCD 与 gdb 运行在同一台设备上，则可以用 127.0.0.1 表示本机 IP，

否则要改为运行 OpenOCD 的设备的实际 IP 地址；3333 表示 OpenOCD 的配置文件中设置的端口号，必须与配置文件中一致。

```
1 # debug adapter
2 interface jlink
3 jlink serial 000504404532
4
5 transport select jtag
6 adapter_khz 3000
7
8 # server port
9 gdb_port 3333
10 telnet_port 4444
11
12 # add cpu target
13 set _CHIPNAME riscv
14
15 jtag newtap $_CHIPNAME cpu -irlen 5 -expected-id 0x04e4796b
16
17 set _TARGETNAME $_CHIPNAME.cpu
18 target create $_TARGETNAME riscv -chain-position $_TARGETNAME
19
20 # command
21 init
22 halt
```

输入：load

输入：c，运行程序，此时查看串口工具，收到 Hello World。

## 5.2.IDE 方式

### 5.2.1.界面简介

### 5.2.2.编译、下载、调试

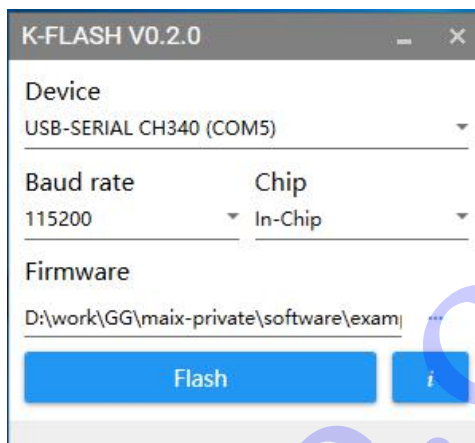
（编译，下载，调试）



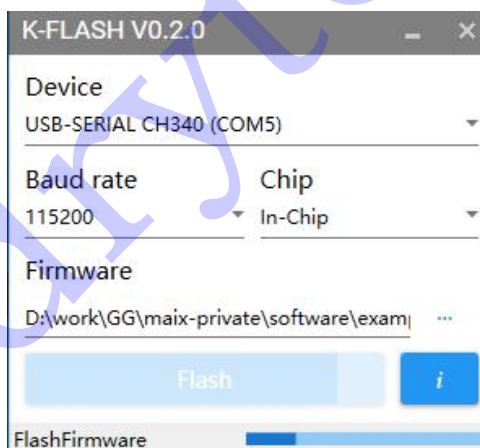
## 6.ISP 下载

下载 K-Flash-V0.2.0 包，解压。开发板插上 Type-C，IO16 连接地（GND），上电进入 ISP 模式。

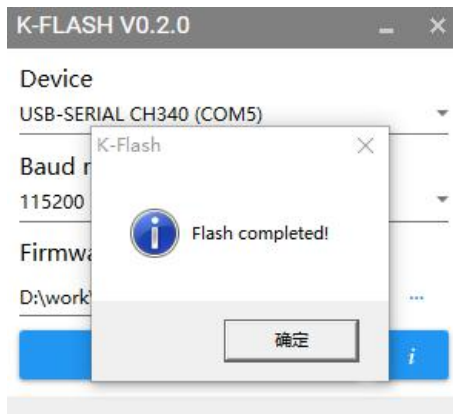
- 运行 K-Flash.exe，如下图所示：自动识别 COM 号及设置波特率。



- 选择下载的 bin，点击 Flash 按钮。下载中，如下图所示：



- 下载结束，如下图：



- 拔下 ISP 短接线，按复位键，运行。