

# Sipeed RV Debugger 调试 K210 (Linux, OpenOCD)

## 准备

硬件准备：

- Sipeed RV Debugger
- Sipeed Maix Bit
- USB Type-C 数据线、杜邦线

软件准备：

- Kendryte Standalone SDK/Kendryte FreeRTOS SDK
- kendryte-toolchain
- kendryte-openocd

## K210 引脚描述

IO\_16 用于 **boot** 模式选择，上电时，拉高进入 **Flash** 启动，拉低进入 **ISP** 模式

- Boot 选择

Boot 选择引脚	拉高	拉低
IO_16	Flash 启动	ISP 模式

- ISP 引脚

引脚序号	ISP 功能
IO_4	TX
IO_5	RX

- JTAG 调试引脚

Vcc	1	2	Vcc
nTRST	3	4	GND
TDI	5	6	GND
TMS	7	8	GND
TCK	9	10	GND
RTCK	11	12	GND
TDO	13	14	GND
nSRST	15	16	GND
NC	17	18	GND
NC	19	20	GND

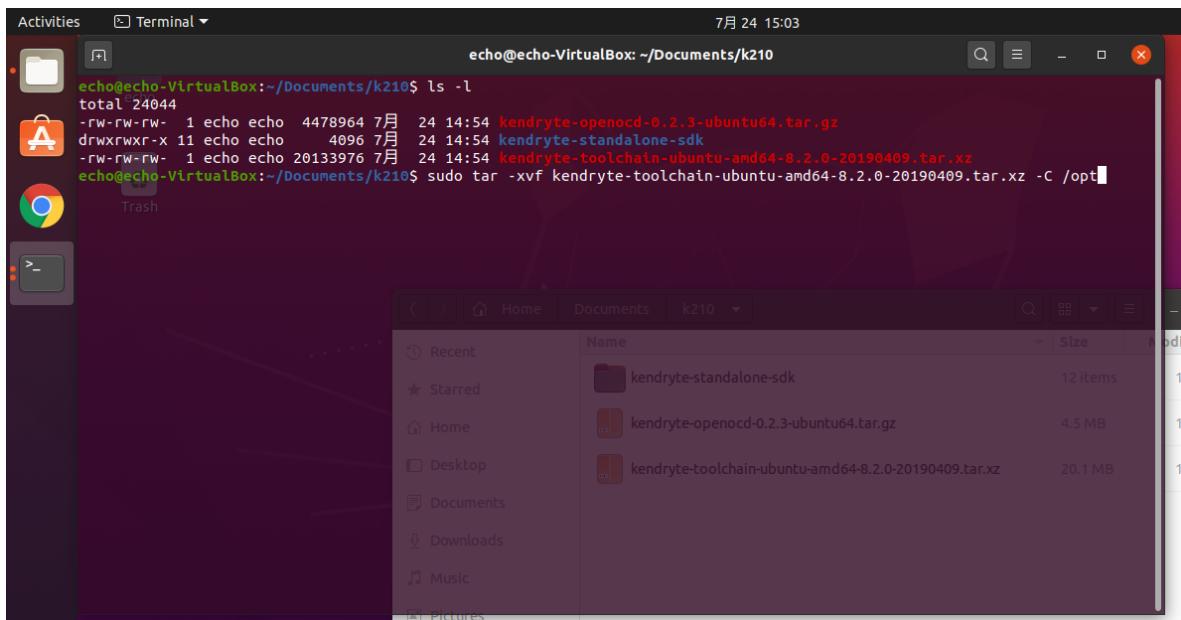
引脚序号	功能管脚	JTAG 功能	JATG 序号
IO_0	FUNC_JTAG_TCLK	JTAG 时钟接口	9
IO_1	FUNC_JTAG_TDI	JTAG 数据输入接口	5
IO_2	FUNC_JTAG_TMS	JTAG 控制 TAP 状态机的转换	7
IO_3	FUNC_JTAG_TDO	JTAG 数据输出接口	13

## 搭建 K210 开发环境()

下载并安装工具链

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

```
1 | sudo tar -xvf kendryte-toolchain-ubuntu-amd64-8.2.0-20190409.tar.xz -C /opt
```



将编译链路径添加到系统环境变量中：

- 永久配置环境变量

```
4 # see /usr/share/doc/bash/examples/startup-files for examples.
5 # the files are located in the bash-doc package.
6
7 # the default umask is set in /etc/profile; for setting the umask
8 # for ssh logins, install and configure the libpam-umask package.
9 #umask 022
10
11 # if running bash
12 if [ -n "$BASH_VERSION" ]; then
13     # include .bashrc if it exists
14     if [ -f "$HOME/.bashrc" ]; then
15         . "$HOME/.bashrc"
16     fi
17 fi
18
19 # set PATH so it includes user's private bin if it exists
20 if [ -d "$HOME/bin" ] ; then
21     PATH="$HOME/bin:$PATH"
22 fi
23
24 # set PATH so it includes user's private bin if it exists
25 if [ -d "$HOME/.local/bin" ] ; then
26     PATH="$HOME/.local/bin:$PATH"
27 fi
28
29 export PATH="/opt/kendryte-toolchain/bin/:$PATH"
30
31
-- INSERT --
```

编辑 `vim ~/.profile`，将编译链工具路径添加到最后一行

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

- 临时配置环境变量

```
1 | alias K210="export PATH='/opt/kendryte-toolchain/bin/:$PATH'"
```

- 下载 sdk

```
1 | git clone --recursive https://github.com/kendryte/kendryte-  
|   standalone-sdk.git
```

具体构建编译请查看 K210 官方说明

The screenshot shows the Visual Studio Code interface with the following details:

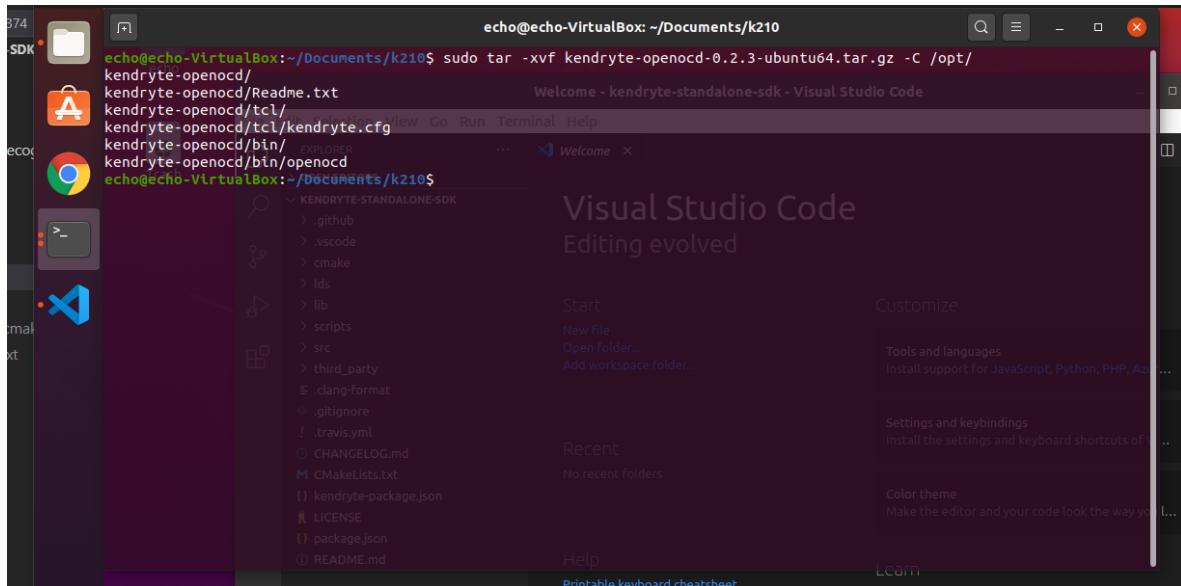
- File Explorer:** Shows files like `README.md`, `CMakeLists.txt`, `kendryte-package.json`, `LICENSE`, `package.json`, and `README.md`.
- Terminal:** Displays the command-line output of a CMake build for a "hello\_world" project. The terminal shows the configuration command: `echo echo-VirtualBox$ cmake ..../ -DPROJ=hello_world -DTOOLCHAIN=/opt/kendryte-toolchain/bin && make`. It then lists various compiler detection steps:
  - Checking for RISCV toolchain.
  - Using `/opt/kendryte-toolchain/bin` RISCV toolchain.
  - The C compiler identification is GNU 8.2.0.
  - The CXX compiler identification is GNU 8.2.0.
  - The AS compiler identification is GNU 8.2.0.
  - Found assembler `/opt/kendryte-toolchain/bin/riscv64-unknown-elf-gcc`.
  - 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.
- Status Bar:** Shows the current file as `README.md` and the status `1:make`.

# Sipeed RV Debugger 调试 K210

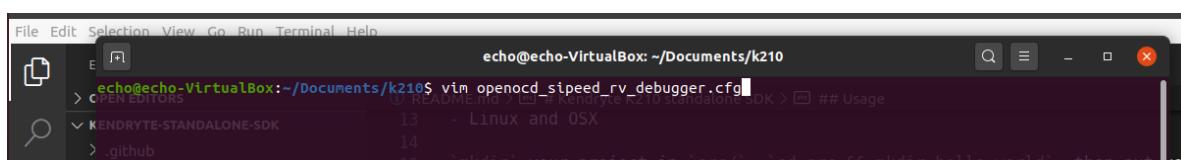
## 下载并安装 OpenOCD

- 工具链下载地址: <https://github.com/kendryte/openocd-kendryte/releases>

```
1 | sudo tar -xvf kendryte-openocd-0.2.3-ubuntu64.tar.gz -C /opt/
```



创建 openocd sipeed rv debugger.cfg 文件并加入下面内容



```

Edit Selection View Go Run Terminal Help
echo@echo-VirtualBox: ~/Documents/k210
E J+ ① README.md > [?] # Kendryte K210 standalone SDK > [?] ## Usage
> OPEN [?] ftdi
> # for canaan's ftdi
> ftdi_vid_pid 0x0403 0x6010 ② - Linux and OSX
> ftdi_layout_init 0x00e8 0x00eb ③
> 5
> 6 transport select jtag
> 7 ftdi_tdo_sample_edge falling
> 8 adapter_khz 64000
> 9
> 10 gdb_port 3333
> 11 telnet_port 4444
> 12
> 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 ignore
> 20 init
> 21 halt
C 22 ANGELOG.md
M ~CMakeLists.txt
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
[ 60%] Building C object lib/CMakeFiles/kendryte.dir/drivers/
[ 62%] Building C object lib/CMakeFiles/kendryte.dir/drivers/
[ 65%] Building C object lib/CMakeFiles/kendryte.dir/drivers/
[ 67%] Building C object lib/CMakeFiles/kendryte.dir/drivers/
[ 69%] Building C object lib/CMakeFiles/kendryte.dir/drivers/
[ 72%] Building C object lib/CMakeFiles/kendryte.dir/drivers/
[ 74%] Building C object lib/CMakeFiles/kendryte.dir/drivers/

```

```

1 interface ftdi
2 # for canaan's ftdi
3 ftdi_vid_pid 0x0403 0x6010
4 ftdi_layout_init 0x00e8 0x00eb
5
6 transport select jtag
7 ftdi_tdo_sample_edge falling
8 adapter_khz 64000
9
10 gdb_port 3333
11 telnet_port 4444
12
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 init
21 halt

```

## Linux 还需要安装 OpenOCD 依赖与配置规则

- 安装 openocd 需要的依赖

```
1 | sudo apt install libusb-dev libftdi-dev libhidapi-dev
```

- 添加 udev 规则

```

1 # Recommended
2 curl -fsSL https://raw.githubusercontent.com/platformio/platformio-
core/master/scripts/99-platformio-udev.rules | sudo tee /etc/udev/rules.d/99-
platformio-udev.rules
3
4 # OR, manually download and copy this file to destination folder
5 sudo cp 99-platformio-udev.rules /etc/udev/rules.d/99-platformio-udev.rules

```

- 重启“udev”管理工具

```

1 sudo service udev restart
2
3 # or
4
5 sudo udevadm control --reload-rules
6 sudo udevadm trigger

```

- Ubuntu / Debian用户可能需要在“dialout”组中添加自己的“用户名”，如果不是“root”用户，执行此命令

```

1 sudo usermod -a -G dialout $USER
2 sudo usermod -a -G plugdev $USER

```

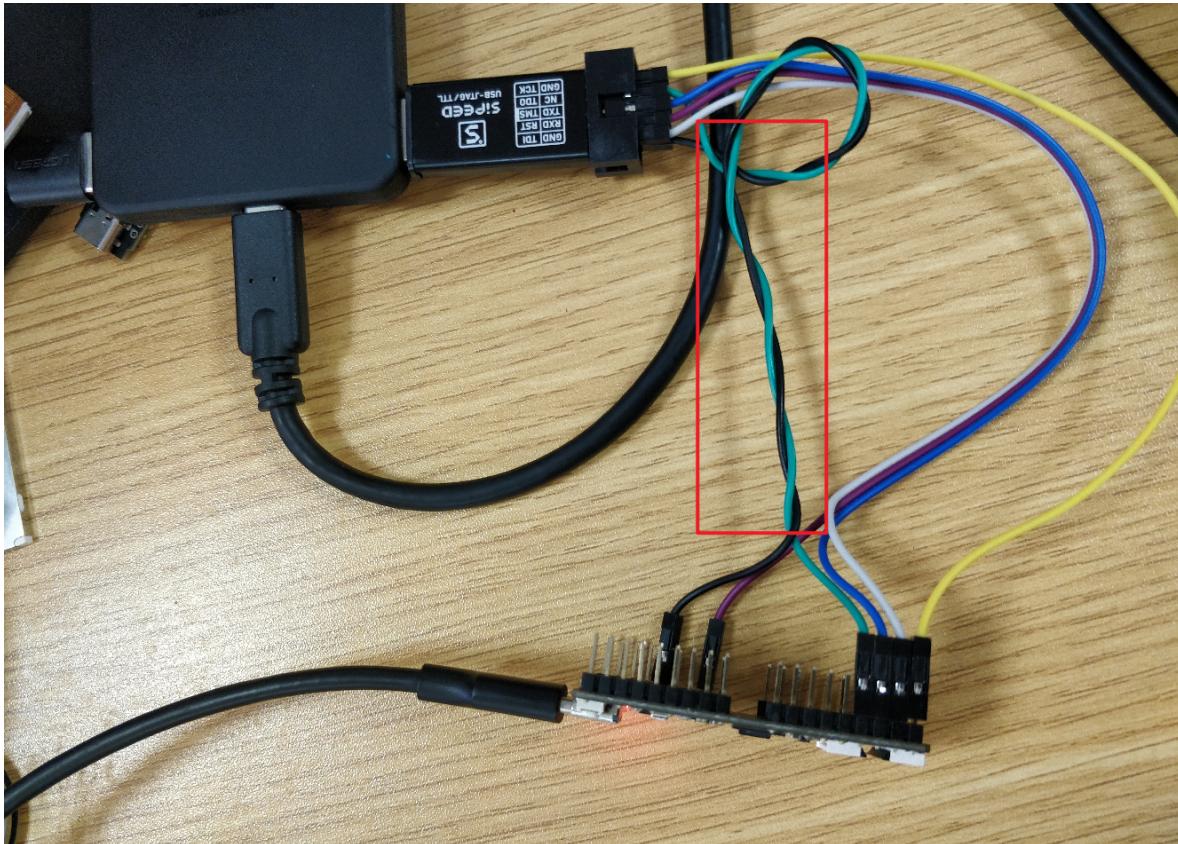
- 重新插拔调试器使修改生效

按照下表连接好 Sipeed RV Debugger 与 K210 开发板

K210 引脚序号	JTAG 功能管脚	Sipeed RV Debug 引脚	功能	说明
IO_0	FUNC_JTAG_TCLK	TCK	JTAG 时钟接口	
IO_1	FUNC_JTAG_TDI	TDI	JTAG 数据输入接口	
IO_2	FUNC_JTAG_TMS	TMS	JTAG 控制 TAP 状态机的转换	
IO_3	FUNC_JTAG_TDO	TDO	JTAG 数据输出接口	使用双绞线方式与 GND 缠绕，降低干扰
RST		RST	复位引脚	
GND		GND	信号地	

#### 注意：

由于 调试时使用的数据速率较高，需要使用双绞线方式把 TDO 与 GND 缠绕，降低干扰



运行 openocd

```
echo@echo-VirtualBox:~/Documents/k210$ ls -l
total 24048
drwxrwxr-x 12 echo echo 4478964 7月 24 14:54 kendryte-openocd-0.2.3-ubuntu64.tar.gz
drwxrwxr-x 12 echo echo 4096 7月 24 15:40 kendryte-standalone-sdk
-rw-rw-rw- 1 echo echo 20133976 7月 24 14:54 kendryte-toolchain-ubuntu-amd64-8.2.0-20190409.tar.xz
> -rw-rw-r-- 1 echo echo 389 7月 24 15:46 openocd_sipeed_rv_debugger.cfg
echo@echo-VirtualBox:~/Documents/k210$ sudo service udev restart
echo@echo-VirtualBox:~/Documents/k210$ sudo usermod -a -G dialout echo
echo@echo-VirtualBox:~/Documents/k210$ /opt/kendryte-openocd/bin/openocd -f openocd_sipeed_rv_debugger.cfg
mkdir build && cd build
make .. -DPROJ=<ProjectName> -DT0OLCHAIN=/opt/riscv-toolchain/bin && make
Windows
Kendryte Open On-Chip Debugger For RISC-V v0.2.3 (2019-02-21)
Licensed under GNU GPL v2
ftdi samples TDO on falling edge of TCK Download and install latest CMake.
adapter speed: 64000 kHz
Info : clock speed 64000 kHz
Info : JTAG tap: riscv.cpu tap/device found: 0x04e4796b (mfg: 0x4b5 (<unknown>), part: 0x4e47, ver: 0x0)
Core [0] halted at 0x80000362 due to debug interrupt
Info : Examined RISCV core; found 2 harts
Info : Listening on port 3333 for gdb connections
Core [1] halted at 0x8000071a2 due to debug interrupt
Core [0] halted at 0x80000362 due to debug interrupt
Info : Listening on port 6666 for tcl connections
Info : Listening on port 4444 for telnet connections
[ 67%] Building C object lib/CMakeFiles/kendryte.dir/drivers/i2c.c.obj
[ 69%] Building C object lib/CMakeFiles/kendryte.dir/drivers/i2s.c.obj
[ 72%] Building C object lib/CMakeFiles/kendryte.dir/drivers/plic.c.obj
[ 72%] Building C object lib/CMakeFiles/kendryte.dir/drivers/pwm.c.obj
```

这里开启的是 3333 端口，使用 K210 编译链提供的 GDB 连接 3333 端口

The screenshot shows a terminal window with several tabs open. The active tab displays the command-line interface for building a RISC-V application. The user has navigated to the directory containing their code and run a series of commands to compile the application. The output of these commands is visible in the terminal window, showing the compilation progress and the final successful execution of the application.

```
echo@echo-VirtualBox:~/Documents/k210$ ls -l
total 24048
drwxrwxr-x 12 echo echo 4478964 7月 24 14:54 kendryte-openocd-0.2.3-ubuntu64.tar.gz
-rw-rw-r-- 1 echo echo 20133976 7月 24 14:54 kendryte-toolchain-ubuntu-ande4-0.2.0-20190409.tar.xz
-rw-rw-r-- 1 echo echo 389 7月 24 14:54 openocd_sipeed_rv_debugger.cfg
echo@echo-VirtualBox:~/Documents/k210$ sudo service udev restart
echo@echo-VirtualBox:~/Documents/k210$ sudo usermod -a -G dialout echo
echo@echo-VirtualBox:~/Documents/k210$ /opt/kendryte-openocd/bin/openocd -f openocd_sipeed_rv_debugger.cfg

mkdir build && cd build
[  0%] [  1%] [  2%] [  3%] [  4%] [  5%] [  6%] [  7%] [  8%] [  9%] [ 10%] [ 11%] [ 12%] [ 13%] [ 14%] [ 15%] [ 16%] [ 17%] [ 18%] [ 19%] [ 20%] [ 21%] [ 22%] [ 23%] [ 24%] [ 25%] [ 26%] [ 27%] [ 28%] [ 29%] [ 30%] [ 31%] [ 32%] [ 33%] [ 34%] [ 35%] [ 36%] [ 37%] [ 38%] [ 39%] [ 40%] [ 41%] [ 42%] [ 43%] [ 44%] [ 45%] [ 46%] [ 47%] [ 48%] [ 49%] [ 50%] [ 51%] [ 52%] [ 53%] [ 54%] [ 55%] [ 56%] [ 57%] [ 58%] [ 59%] [ 60%] [ 61%] [ 62%] [ 63%] [ 64%] [ 65%] [ 66%] [ 67%] [ 68%] [ 69%] [ 70%] [ 71%] [ 72%] [ 73%] [ 74%] [ 75%] [ 76%] [ 77%] [ 78%] [ 79%] [ 80%] [ 81%] [ 82%] [ 83%] [ 84%] [ 85%] [ 86%] [ 87%] [ 88%] [ 89%] [ 90%] [ 91%] [ 92%] [ 93%] [ 94%] [ 95%] [ 96%] [ 97%] [ 98%] [ 99%] [ 100%]
Scanning dependencies of target hello_world
[ 97%] Building C object CMakeFiles/hello_world.dir/src/main.c.o
[100%] Linking C executable hello_world
Generating bin file
[100%] Built target hello_world
echo@echo-VirtualBox:hello_world$ /opt/kendryte-toolchain/bin/riscv64-unknown-elf-gdb hello_world
```

### 远程调试:

```
1 | target remote [remote_ip]:3333
```

本机调试:

```
1 target remote localhost:3333  
2 #or target remote :3333
```

显示已连接

```
echo@echo-VirtualBox:~/Documents/k210$ /opt/kendryte-openocd/bin/openocd -f openocd_steedo_rv_debugger.cfg
RECOMMENDED: If you are using K210 evaluation board, run "make Usage"
> OPEN EDITORS
K210/ESTANDALONE-Linux and OSX
> [ ] mkdir` your project in 'src', `cd src && mkdir hello_world`, then put your codes in it, enter SDK root directory and run "make" to build it.
> Kendryte Open-Chip Debugger For RISC-V v0.2.3 (2019-02-21)
> Licensed under GNU GPL v2
> ftdi samples TDO on falling edge of TCK mkdir build && cd build
adapter speed: 64000 kHz
> Info : clock speed 64000 kHz          19  cmake .. -DPROJ=<ProjectName> -DTOOLCHAIN=/opt/riscv-toolchain/bin && make
> Info : JTAG tap: riscv.cpu tap/device found: 0x04e4796b (mfg: 0x4b5 (<unknown>), part: 0x4e47, ver: 0x0)
> Core [0] halted at 0x00000362 due to debug interrupt
> Info : Examined RISCV core; found 2 harts
> Info : Listening on port 3333 for gdb connections
> Core [1] halted at 0x000071a2 due to debug interrupt
> Core [0] halted at 0x00000362 due to debug interrupt and install latest CMake.
> Info : Listening on port 6666 for tcl connections
> Info : Listening on port 4444 for telnet connections
> Info : accepting 'gdb' connection on tcp/3333
C:\CHANGELOG.md
C:\CHANGELOG.md

M CMakeLists.txt
{} kendryte-package.json
R LICENSE
{:} package.json
{:} README.md

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
echo@echo-VirtualBox:hello_world$ /opt/kendryte-toolchain/bin/riscv64-unknown-elf-gdb hello_world
GNU gdb (GDB) 8.2
Copyright (C) 2018 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 :3333
Remote debugging using :3333
warning: Target-supplied registers are not supported by the current architecture
0x000000000000362 in __do_global_dtors_aux ()
(gdb) load[]

> OUTLINE
> TIMELINE
```

## 加载 bin