

openocd -- 目前在ubuntu 16.04系统上测试成功

openocd使用

```
jtag_rest 1 0          -- 复位jtag和TMS组合而成，如果有trstn的话，则产生trstn
irscan zynq.dap 0x5    -- 往tap寄存器写IR指令为0x5， TAP在target.cfg文件中定义jtag newtap
drscan zynq.dap 16 0x4321 -- 向指定TAP输入DR数据，长度为16，数据为0x4321
```

--- 注意，一定要先指定TAP IR, 然后才可以向 DR进行输入数据，不然openocd会报错退出 --- openocd使用tcl语法进行配置，可以使用telnet进行连接

1. config file

1.1 ft2232h.cfg

[ft2232h.cfg](#)

```
#####
#
# Layout: FTDI FT2232H example
# ADBUS0 TCK
# ADBUS1 TDI
# ADBUS2 TDO (input)
# ADBUS3 TMS
# ADBUS4 nTRST
# ADBUS5 nSRST
# ADBUS6 OE (active high) for TRST, TDI, TMS, TCK
# ADBUS7 DBGREQ_L

interface ftdi
ftdi_vid_pid 0x0403 0x6010
ftdi_channel 0

##### interface config 0
# just TCK TDI TDO TMS
# ftdi_layout_init 0x0008 0x000b

##### interface config 1
# ADBUS6 no used
ftdi_layout_init 0x0098 0x00bb; # initial data, port direction (1-out, 0-in)
ftdi_layout_signal nTRST -data 0x0010 -input 0x0010
ftdi_layout_signal nSRST -data 0x0020 -input 0x0020
ftdi_layout_signal DBGREQ_L -data 0x0080 -input 0x0080
```

```
# -data    can drive to 1 & 0
# -oe      can only drive to 0 / high-z, can't drive 1
# -input   can use cmd ftdi_get_signal to get signal value
```

1.2 target.cfg

target.cfg

```
set _CHIPNAME zynq
jtag newtap $_CHIPNAME dap -irlen 8 -ircapture 0x01 -irmask 0x03

adapter_khz 5000

# gdb_port      3333
# tcl_port       6666
# telnet_port    4444
```

2. 启动服务

```
sudo openocd -f ft2232h.cfg -f target.cfg
```

3. RPC server

OpenOCD provides a simple RPC server that allows to run arbitrary Tcl commands and receive the results. To access it, your application needs to connect to a configured TCP port (see `tcl_port`). Then it can pass any string to the interpreter terminating it with `0x1a` and wait for the return value (it will be terminated with `0x1a` as well). This can be repeated as many times as desired without reopening the connection. Remember that most of the OpenOCD commands need to be prefixed with `ocd_` to get the results back. Sometimes you might also need the capture command. See `contrib/rpc_examples/` for specific client implementations.

可以参考openocd包目录contrib/rpc_examples/下的例子。

