

tcl で Vivado HW Manager

試した環境は、ちょっと古くて、Vivado 2020.1 .
ターゲット デバイスっていう関係に注意してアクセスするといいい .
USB で接続されているターゲットの情報は

```
get_hw_targets # 接続されてる FPGA ユニットのリストがとれる  
get_hw_targets {*/500202A503FAA} # で名前でフィルタリングできる
```

で取れる . open_hw_target で

```
open_hw_target [get_hw_targets {*/500202A503FAA}]
```

などとしてターゲットを開いたら、

```
get_hw_devices
```

で FPGA にアクセスできるようになる .

get_hw_devices は、同じ種類の FPGA がたくさん USB 接続下にいる場合、
連番のサフィックスがつくので注意 .

一つしかなければ、get_hw_devices の返り値が一つだから問題ないけど複数だとちょっと厄介 .
複数でも名前でわかりやすく区別できればサフィックス以外でフィルタリングして

```
get_hw_devices {*xcu50*}
```

とかして選別できる .

そうじゃない場合、順番気にしないといけなそうで、ちょっと面倒そう .

例 0: 接続されているアダプタの識別子リストを得る

```
open_hw_manager  
connect_hw_server -allow_non_jtag  
puts [get_hw_targets]  
disconnect_hw_server  
close_hw_manager  
quit
```

例 1: 狙ったプログラミングアダプタに接続された FPGA(Alveo U50) を ROM から起動する

プログラミングアダプタの識別子が環境変数 ADAPTER で定義されていて、
その先には Alveo U50 が接続されていることを想定 .

```
open_hw_manager  
connect_hw_server -allow_non_jtag  
set adapter ""  
append adapter "*/" $::env(ADAPTER)  
set target [get_hw_targets -quiet $adapter]; # 以降 $target で参照できる  
if {[string equal $target ""]} {  
    disconnect_hw_server  
    close_hw_manager  
    quit  
}
```

```

}
open_hw_target $target; # これでターゲットに接続
set dev [current_hw_device [get_hw_devices -quiet {*50*}]]; # 念の為に Alveo U50 を選別できるように
; # 以降 $dev で参照できる

if {[string equal $dev ""]} {
  close_hw_target $target
  disconnect_hw_server
  close_hw_manager
  quit
}
boot_hw_device [lindex $dev 0]; # ROM から起動
# 後片付け
close_hw_target $target
disconnect_hw_server
close_hw_manager
quit

```

例 2: 狙ったプログラミングアダプタに接続された FPGA(Alveo U50) の ROM に MCS を書き込む

プログラミングアダプタの識別子が環境変数 ADAPTER で定義されていて

その先には Alveo U50 が接続されていること、および、

書き込む MCS ファイルは環境変数 MCSFILE で定義されていること、を想定。

Alveo U50 想定ということで、ROM は mt25qu01g-spi-x1_x2_x4 のはず、という想定。

```

open_hw_manager
connect_hw_server -allow_non_jtag
set adapter ""
append adapter "*/" $::env(ADAPTER)
set target [get_hw_targets -quiet $adapter]; # 以降 $target で参照できる
if {[string equal $target ""]} {
  disconnect_hw_server
  close_hw_manager
  quit
}
open_hw_target $target; # これでターゲットに接続
set dev [current_hw_device [get_hw_devices -quiet {*50*}]]; # 念の為に Alveo U50 を選別できるように
; # 以降 $dev で参照できる

if {[string equal $dev ""]} {
  close_hw_target $target
  disconnect_hw_server
  close_hw_manager
  quit
}
# ここから ROM の設定
create_hw_cfgmem -hw_device $dev [lindex [get_cfgmem_parts {mt25qu01g-spi-x1_x2_x4}] 0]
set_property PROGRAM.ADDRESS_RANGE {use_file} [get_property PROGRAM.HW_CFGMEM $dev]
set_property PROGRAM.FILES $::env(MCSFILE) [get_property PROGRAM.HW_CFGMEM $dev]
set_property PROGRAM.PRM_FILE {} [get_property PROGRAM.HW_CFGMEM $dev]
set_property PROGRAM.UNUSED_PIN_TERMINATION {pull-none} [get_property PROGRAM.HW_CFGMEM $dev]
set_property PROGRAM.BLANK_CHECK 0 [get_property PROGRAM.HW_CFGMEM $dev]
set_property PROGRAM.ERASE 1 [get_property PROGRAM.HW_CFGMEM $dev]
set_property PROGRAM.CFG_PROGRAM 1 [get_property PROGRAM.HW_CFGMEM $dev]
set_property PROGRAM.VERIFY 1 [get_property PROGRAM.HW_CFGMEM $dev]
set_property PROGRAM.CHECKSUM 0 [get_property PROGRAM.HW_CFGMEM $dev]
# ROM 書き込み用 bit ファイルの書き込み
create_hw_bitstream -hw_device $dev [get_property PROGRAM.HW_CFGMEM_BITFILE $dev]
program_hw_devices $dev
# ROM 書き込み
program_hw_cfgmem -hw_cfgmem [get_property PROGRAM.HW_CFGMEM $dev]
# 後片付け
close_hw_target $target
disconnect_hw_server
close_hw_manager
quit

```