

Using the Diligent JTAG-HS2 Programming Cable with OpenOCD

2019-10-11

1 About

The JTAG-HS2 Programming Cable is an USB JTAG probe. It is available here: <https://store.digilentinc.com/jtag-hs2-programming-cable/>



The cable is fully compatible with all Xilinx tools, but it can also be used with OpenOCD.

2 Software configuration

2.1 Libusb

You need libusb to use the cable.

On Ubuntu, you can use apt:

```
$ sudo apt install libusb-1.0-0-dev
```

You can also build it yourself using the sources available on <https://libusb.info>

2.2 Libftdi

The cable contains an FTDI chip, so you need libftdi.

Ubuntu:

```
$ sudo apt install libftdi1-dev
```

Or from sources: <https://www.intra2net.com/en/developer/libftdi/download.php>

2.3 OpenOCD Compilation

The OpenOCD interface used with the cable is *ftdi*.

To enable this interface, use the `--enable-ftdi` option when compiling OpenOCD.

```
$ ./configure --enable-ftdi <other options>
```

If you are not sure if your existing instance of OpenOCD supports the ftdi interface, you can list the debug adapter drivers that have been built into the running copy of OpenOCD with:

```
$ openocd -c interface_list
```

2.4 udev rules

It is necessary to add a udev rule to use the cable.

OpenOCD provides a file containing the rule we need. Copy it into `/etc/udev/rules.d/`

```
$ sudo cp <openocd>/contrib/60-openocd.rules /etc/udev/rules.d
```

The file is also available here: <https://github.com/riscv/riscv-openocd/blob/riscv/contrib/60-openocd.rules>

The particular entry about the HS2 cable is :

```
ATTRS{idVendor}=="0403", ATTRS{idProduct}=="6014", MODE="660", GROUP="
plugdev", TAG+="uaccess"
```

Then either reboot your system or reload the udev configuration with :

```
$ sudo udevadm control --reload
```

2.5 Checking

To check if the cable is recognized, run `lsusb`. There should be a line like this:

```
$ lsusb
[...]  
Bus 005 Device 003: ID 0403:6014 Future Technology Devices International,  
Ltd FT232H Single HS USB-UART/FIFO IC  
[...]
```

For more details about the device, use more options:

```
$ lsusb -v -d 0403:6014
```

3 OpenOCD Config File

Examples of config files using the HS2 cable can be found on the Internet:

- <https://github.com/pulp-platform/pulpiissimo/blob/master/fpga/pulpiissimo-zedboard/openocd-zedboard-hs2.cfg>
- <https://github.com/arduino/OpenOCD/blob/master/tcl/interface/ftdi/digilent-hs2.cfg>

Generally, for the interface part of the config file, you can use:

```
interface ftdi
ftdi_device_desc "Digilent USB Device"
ftdi_vid_pid 0x0403 0x6014
ftdi_serial 210249A85F9B
ftdi_channel 0
ftdi_layout_init 0x00e8 0x60eb

reset_config none
```

Use `lsusb -v -d 0403:6014` to find the necessary values :

- **ftdi_device_desc**: this command should use the same value as the **iProduct** field
- **ftdi_vid_pid**: **idVendor** and **idProduct**
- **ftdi_serial**: this command is useful if there are more than one adapter connected to the host and can be omitted otherwise. The needed value is in the **iSerial** field.
- **ftdi_channel**: the default (channel 0) is fine
- **ftdi_layout_init**: initial values of the FTDI GPIO data and direction registers. Refer to the adapter schematics to find them. 0x00e8 and 0x60eb work for the HS2 cable

For more information about the OpenOCD commands, see <http://www.openocd.org/doc/html/Debug-Adapter-Configuration.html>

4 Hardware configuration

The HS2 cable does not use the TRST signal.

Do not leave the pin floating and tie it to 1 if the target TAP reset is active low, or 0 if it is active high.

5 Vivado Hardware Manager

The Digilent JTAG-HS2 cable is supported by the Xilinx Hardware Server. For some reason, it is impossible to use the cable with OpenOCD while the Hardware Server is active.

This makes using Vivado and OpenOCD at the same time a problem.

A possible fix is the following protocol:

- Open the hardware server in Vivado and connect to your FPGA board
- Upload the bitstream and program the device
- **Close** the hardware server
- Launch OpenOCD

Of course, This closes the hardware server and removes access to Vivado features.

To properly fix the issue, you would need to find a way of preventing the Vivado Hardware Server from occupying the USB device of the cable.