# OHWR Generic setup tutorial

20 Jun. 2012

**Benoit RAT**

# Contents

# 1 Introduction

This guide has been written to obtain a quick development environment for an ohwr.org user.

We have used ***Ubuntu 12.04 (LTS) 32bits*** to perform the installation of the different components.

# 2 Hardware Tools

The common tools for all the hardware project in ohwr.org are:

- Xilinx ISE
- Hdlmake (Python)
- Git
- Subversion

To install git and subversion just look at Global apt-get

## 2.1 Xilinx ISE[1]

You can follow this http://www.george-smart.co.uk/wiki/Xilinx_JTAG_Linux to install Xilinx ISE on Linux. The following steps try to briefly resume the steps after the Xilinx ISE instalation for a 32 bits platform

### 2.1.1 Xilinx USB JTAG cable

First we download the required libraries

```
sudo apt-get install gitk git-gui libusb-dev build-essential libc6-dev fxload
```

Then we download the driver source

```
cd /opt/Xilinx
sudo git clone git://git.zerfleddert.de/usb-driver
```

Then we compile the driver:

```
cd usb-driver/
sudo make
```

And finally we set it up

```
./setup_pcusb /opt/Xilinx/13.2/ISE_DS/ISE/
```

### 2.1.2 Xilinx Environment

Then we add Xilinx ISE to the **PATH** and create a **XILINX** variable by editing `${HOME}/.bashrc` and adding:

[1] We have installed Xilinx ISE 13.2 on a 32bit linux platform in this tutorial, you might changes some parameters to fit your instalation

```
export XILINX=/opt/Xilinx/13.2/ISE_DS/
PATH=$PATH:$XILINX/ISE/bin/lin
```

### 2.2 HDLMake

Download from git by executing the following command.

```
git clone git://ohwr.org/misc/hdl-make.git
```

And then install by putting hdlmake executable somewhere in the PATH

- If you have root access we suggest you do copy or link hdlmake to the `/usr/local/bin`, this way hdlmake will be installed for all users:

```
sudo ln -s /opt/hdlmake/hdlmake /usr/local/bin/hdlmake
```

- If you don't have root access the best way is to modify the path variable by editing `${HOME}/.bashrc` and adding:

```
PATH=$PATH:~/hdlmake/
```

Finally you should check that it work `hdlmake --help`

### 2.3 Fake modelsim error

This operation might be not usefull on all computer

You need to perform the following steps

```
sudo mkdir /opt/modelsim
cd /opt/modelsim
touch modelsim.ini
mkdir linux
touch linux/vsim
chmod +x linux
```

Setting the environment (editing `${HOME}/.bashrc`) you must also add modelsim (vsim executable to the path), because it needs to find modelsim.ini

```
export MODELTECH=/opt/modelsim
PATH=$PATH:${MODELTECH}/linux
```

## 3 Software Tools

The common tools used for the sofware project in ohwr are

- Git
- Subversion
- build-essentials: *Contains various binaries to build source code*
- Kernel sources: *Might be usefull to compile drivers & kernel modules*
- minicom: *Hyperterminal for linux*
- LM32 cross compiler

### 3.1 Global apt-get

Just enter the following command

```
sudo apt-get install git subversion build-essential build-dep linux minicom
```

### 3.2 LM32 Cross-compiler:

You can install soc-lm32 cross compiler by downloading & extracting it to /opt/

```
wget http://www.das-labor.org/files/madex/lm32_linux_i386.tar.bz2
sudo tar -xvjf lm32_linux_i386.tar.bz2 -C /opt/
export PATH=/opt/lm32/bin/:$PATH
```

To use it you just need to export the **CROSS_COMPILER** variable:

```
export CROSS_COMPILER="/opt/lm32/bin/lm32-elf-"
```

### 3.3 Texinfo & Markdown (Pandoc)

#### 3.3.1 Texinfo

Some documentation are written in texinfo format `.texi`

To compile them you just need to install it

```
sudo apt-get install texinfo
```

And then you should run `make` in the documentation folder. You can also try

```
texi2pdf --batch <filename>.texi
```

#### 3.3.2 Markdown

Markdown is the syntax used to easily generate pretty-formated documentation using plane text. This document is written using the markdown syntax (And all `.mkd` documents)

The syntax is described http://daringfireball.net/projects/markdown/

However we use a special markdown syntax and we generate the tools using pandoc:

- Setup[2] : `sudo apt-get install pandoc`
- Syntax: http://johnmacfarlane.net/pandoc/README.html#pandocs-markdown
- Simple call: `pandoc --toc -o output.pdf input.mkd`

## 4 Installing gnurabbit PCIe driver

This driver (rawrabbit kernel module) must be installed in all the projects that use **SPEC** card (FMC DIO, FMC ADC, PTS, Starting Kit, ...) As this tutorial is created for ubuntu distribution we propose the "cleanest" way to install it.

[2] If you use pandoc version older than `1.9`, you might have problem to generate pdf directly from markdown.

1. First, compile the module
2. Add the module file `rawrabbit.ko` in `/lib/modules/$(uname -r)/kernel/drivers/pci.`
3. Edit `/etc/modules` file and add a new line containing `rawrabbit`
4. Run `sudo update-initramfs -u`
5. and finally (after rebooting) the new module must be loaded as expected

# 5 OHW Repositories Structure

As one can be easily lost inside the OHWR, we have tried to quickly resume the structure of the repositories

- HDLCore lib: *Sharing generic core for all OHWR projects*
  - White Rabbit Core Collection: *Module specific to WR boards (WRS, SPEC, …)*
    * wr_softpll
    * wr_lm32
    * wr_endpoint
    * …
  - DDR3 & QDRII
  - LM32
  - Whishbone Crossbar
  - Whishbone serializer
  - Whishbone Slave Generator: *Should be use when we want to create a new WB slave*
  - …
- White Rabbit: *Contains specific project for WR protocols and WR boards*
  - WR Switch hardware
  - WR Switch Software
  - WR Switch HDL
  - WR NIC
  - PPSi
  - …
- FMC Projects: *Contains all FMC mezzanines & the carriers*
  - SPEC, SVEC, SPEXI, …
  - DIO, ADC, TDC, …
  - …
- Miscellaneous Projets
  - Production Test Suite: *Test of specific boards, such as SPEC, DIO, …*
  - HDLMake: *Software to compile hdl core for OHWR project (modules from various repository)*
  - …