HOW TO : Cross-compile kernel module

!! EVERYTHING IS DONE IN VM VirtualBox !!

host PC : Windows 7 (Intel processor)

guest system : Ubuntu 12.04

/NOTE/ The following is done on guest only (Terminal = Ubuntu Terminal, etc)

1) Download kernel source files

- mind the kernel version, as modules are very sensitive to differences between kernel version compilation and kernel version under which they run (simply said : make sure the version of kernel source files are matching with target linux kernel)

- the embedded platform DE0\_Nano\_SoC runs Linux Console (http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&No=941&PartNo=4) built on kernel version 3.13

- download the kernel source files (http://packages.ubuntu.com/trusty/linux-source-3.13.0)

[linux\_3.13.orig.tar.gz]

- unzip the <file>.tar.gz by command

tar xvzf <file>.tar.gz

2) before compilation of kernel

- install compiler for ARM architecture

sudo apt-get install gcc-arm-linux-gnueabihf

- if compiling 32bit kernel on 64bit machine

sudo apt-get install ia32-libs

- install curses to use **config menuconfig** to create configuration file .config for kernel

sudo apt-get apt-get install libncurses5-dev

- this might not be required

sudo apt-get install build-essentials

3) make menuconfig (SUPER NICE SUPER EASY compared to make config)

- navigate to untared folder (folder linux-3.13)

- run make menuconfig (or make config)

- important settings

- 32 bit system (this ARM is 32bit)

- arm-linux-gnueabihf-

- embedded platform **Y**

4) make modules\_prepare CROSS\_COMPILE=arm-linuxgnueabihf- ARCH=arm

- !!!!! again questions regarding kernel configuration

- !!!!!!!!! MARK Y FOR ALTERA PROCESSOR (approx 10th/20th question from the beginning, everything else default)

5) adjust Makefile of custom module

KERNEL\_LOCATION=<path>/linux-3.13

ARMMAKE=make ARCH=arm SUBARCH=arm CROSS\_COMPILE=arm-linux-gnueabihf-

obj-m := fpga\_uinput.o

fpga\_uinput.ko: fpga\_uinput.c

$(ARMMAKE) -C $(KERNEL\_LOCATION) M=$(PWD) modules

clean:

rm -f \*.ko \*.o \*.mod.c \*.symvers \*.order

- put above Makefile and fpga\_uinput.c into the same folder

6) make

- navigate to folder with custom module source files (Makefile and fpga\_uinput.c)

and run *make*

7) load the kernel module

To load a module by filename (i.e. one that is not installed in /lib/modules/$(uname -r)/):

# insmod filename [args]

To unload a module:

# modprobe -r *module\_name*

Or, alternatively:

# rmmod *module\_name*