HOW TO Compile Linux kernel module

assumption

- the Linux is compiled according to HOW\_TO Getting Angstrom on a SD card

- the folder structure is as follows

**<home>/bin/**

repo

**oe\_core/**

**bitbake/**...

**build/**...

**meta/**...

...

**angstrom-manifest/**

**bitbake/**...

**build/**

**tmp-angstrom-glibc/work/cyclone5\_angstrom/**

**linux\_gnueabi/linux-altera/**

**4.7+gitAUTOINC+e18ed57863-r0/linux-cyclone5-standard-build**

... (a)

**deploy/**

**glibc/images/cyclone5/**

zImage

zImage-4.7+git0+e18ed57863-r0.2-socfpga\_cyclone5...

...\_de0\_sockit-20170410202547.dtb

Angstrom-console-image-glibc-ipk-v2016.12...

...-cyclone5.rootfs.tar.xz

...

**sources/**

**tmp-angstrom-glibc/work-shared/cyclone5/kernel-source**

...

1) Linux compilation output

- the output of compilation process done by bitbake is in **deploy/glibc/images/cyclone5/**

- important files:

zImage

- the kernel

\*.rootfs.tar.xz

- root file system, which can be extracted to the SD card (see the HOW\_TO\_Linux\_for\_DE0\_Nano\_SoC\_on\_SD\_Card\_with\_Angstrom\_File\_System.docx for instructions on how to delete the SD card partition and how to extract the rootfs)

\*.dtb

- compiled device tree blob

- in order to use the one mentioned above,

it needs to be renamed (to socfpga.dtb)

2) Linux kernel source files

- bitbake stores the kernel source files (already configured, no need to create kernel configuration file again by running *make menuconfig*)

- the files are stored in (a) (see above folder structure)

- the module is then compiled against these kernel source files by changing the KERNEL\_LOCATION in the module Makefile to the (a)

- compilation of the module is then simple *make*