

Developing Kernel Module on Yocto

-- *How is a driver integrated into the Yocto Linux?*

Dexuan Cui
Intel Corporation

Agenda

- Introduction to BSP in Yocto
- Kernel Customization in Yocto
- Develop a recipe to integrate our driver
- Build and test the driver

Board Support Package

- BSPs are layers to enable support for specific hardware platforms, like the ESDC-2012 platform
- Defines machine configuration for the board
- Adds machine-specific recipes and customizations
 - Kernel config
 - Graphics drivers (e.g., Xorg)
 - Additional recipes to support hardware features
- <https://wiki.yoctoproject.org/wiki/BSPs>
 - Documentation Overview, Starting a new BSP, FAQs, etc.

Kernel Customization

- We can define a full kernel configuration set (defconfig) or use kernel configuration “fragments”
- Add a kernel configuration fragment (.cfg) to our layer
 - These include standard Linux Kconfig values and are inserted into the generated defconfig
- Add a linux-yocto.bbappend recipe to our layer which includes our config file(s)

Example: linux-yocto_3.0.bbappend

meta-esdc2012/recipes-kernel/linux/linux-yocto_3.0.bbappend:

```
FILESEXTRAPATHS_prepend := "${THISDIR}/${PN}:"  
  
COMPATIBLE_MACHINE_esdc2012 = "esdc2012"  
KMACHINE_esdc2012 = "yocto/standard/crownbay"  
KERNEL_FEATURES_append_esdc2012 += " cfg/smp.scc"  
  
SRCREV_machine_pn-linux-yocto_esdc2012 ?= "63c65842a3a74e4b"  
SRCREV_meta_pn-linux-yocto_esdc2012 ?= "59314a3523e3607"  
  
SRC_URI += "file://enable_various_options.cfg \  
             file://fix_drivers_media_video_Kconfig.patch"
```

Example: enable_various_options.cfg

meta-esdc2012/recipes-kernel/linux/linux-yocto/enable_various_options.cfg

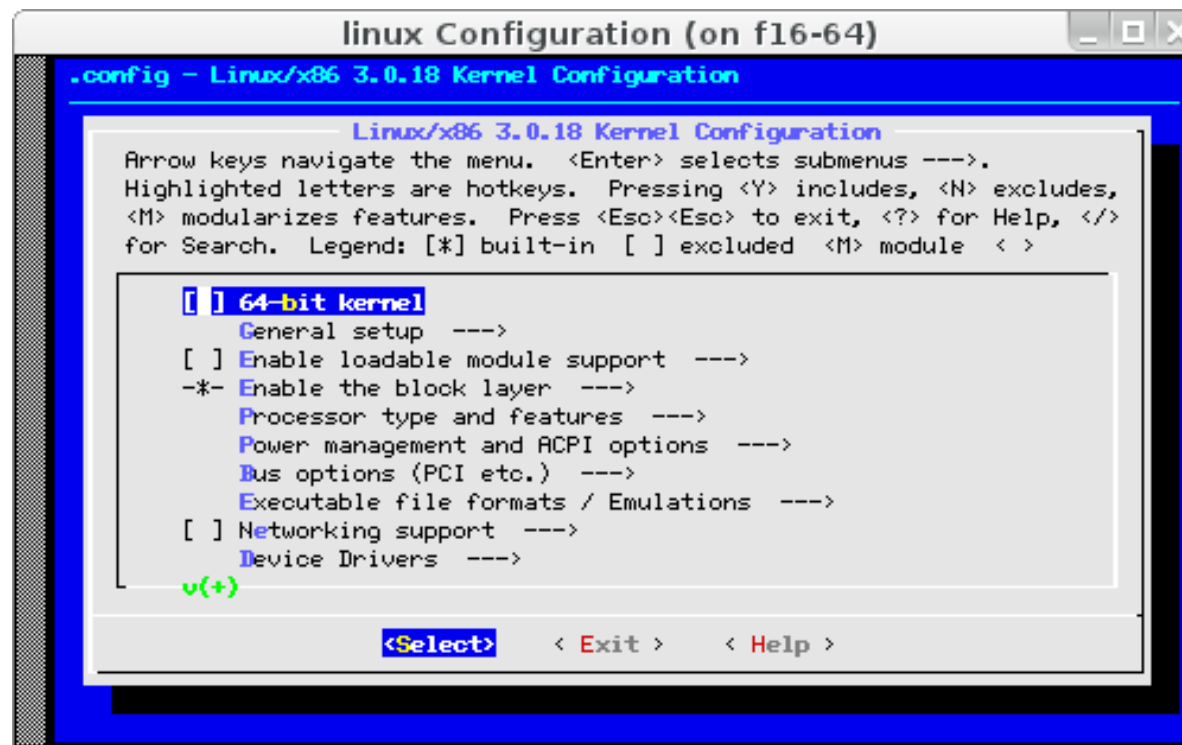
```
CONFIG_NEW_LEDS=y
CONFIG_LEDS_CLASS=y
...
CONFIG_PCI_MSI=y
CONFIG_MFD_SUPPORT=y
CONFIG_MFD_CORE=y
...
```

Menuconfig

- We can run menuconfig as a bitbake task

```
$ bitbake linux-yocto -c menuconfig
```

- Updates the .config in the build directory



Develop a recipe to integrate our driver

- It is simple to write such a recipe
- Poky has an example
 - meta/recipes-kernel/hello-mod/
- Copy the files and make a few **changes**

meta-esdc2012/recipes-kernel/zlg/**zlge6x5c-leds_1.0.bb**

meta-esdc2012/recipes-kernel/zlg/**zlge6x5c-leds/Makefile**

meta-esdc2012/recipes-kernel/zlg/**zlge6x5c-leds/zlge6x5c-leds.c** (the actual source code of the driver)

meta-esdc2012/recipes-kernel/zlg/**zlg6x5c-leds**_1.0.bb

- We only need to change the lines in bold font

```
DESCRIPTION = "ZLG: altera FPGA LED driver"
```

```
LICENSE = "GPLv2+"
```

```
LIC_FILES_CHKSUM = "file://${BPN}.c;endline=19;md5=60b826fef16a983a9523257165cecc2b"
```

```
inherit module
```

```
PR = "r0"
```

```
SRC_URI = "file://Makefile file://${BPN}.c "
```

```
S = "${WORKDIR}"
```

`${BPN}` is a variable in poky, here it means the base package name: `zlg6x5c-leds`

meta-esdc2012/recipes-kernel/zlg/zlge6x5c-leds/Makefile

- We only need to change the lines in bold font

```
obj-m := zlge6x5c-leds.o
```

```
SRC := $(shell pwd)
```

```
all:
```

```
$(MAKE) -C $(KERNEL_SRC) M=$(SRC)
```

```
modules_install:
```

```
$(MAKE) INSTALL_MOD_DIR=kernel/drivers/leds -C $(KERNEL_SRC) M=$(SRC) modules_install
```

```
clean:
```

```
rm -f *.o *~ core .depend *.cmd *.ko *.mod.c
```

```
rm -f Module.markers Module.symvers modules.order
```

```
rm -rf .tmp_versions Modules.symvers
```

Build and install the driver

- `$ bitbake zlge6x5c-leds`
 - If `do_compile` fails, check the building log
 - `tmp/work/esdc2012-poky-linux/zlge6x5c-leds-1.0-r0/temp/log.do_compile`
 - `bitbake zlge6x5c-leds -c clean`; `bitbake zlge6x5c-leds`
- Install the generated `.rpm` package in the target board
 - `tmp/deploy/rpm/esdc2012/zlge6x5c-leds-1.0-r0.esdc2012.rpm`
 - we can copy the file into the running system by `scp` or USB drive
 - `$ rpm -ivh zlge6x5c-leds-1.0-r0.esdc2012.rpm`
- Or, let poky automatically install the package
 - Add a line into `conf/local.conf`: `IMAGE_INSTALL_append = " zlge6x5c-leds"`
 - `$ bitbake core-image-sato`, and use the new `.hddimg`

Boot the board and test the driver

- **modprobe zlge6x5c-leds**
 - Also need to modprobe zlge6x5c-mfd as a dependency
 - Use “lsmod” to check it’s loaded successfully
- **Test the driver with the interface it supplies**

```
echo 1 > /sys/class/leds/zlge6x5c:red:0/brightness
```



Yocto Resources

- <https://wiki.yoctoproject.org/wiki/BSPs>
 - Documentation Overview, Starting a new BSP, FAQs, etc.
- The source code of the meta-esdc2012 layer
 - esdc2012-poky-src.tar.bz2 is available via DVD and the website.
 - See the directory meta-esdc2012/
- Development is done in the open through public mailing lists:
 - openembedded-core@lists.openembedded.org,
 - poky@yoctoproject.org, yocto@yoctoproject.org
- And public code repositories:
 - <http://git.yoctoproject.org>, <http://git.openembedded.net>
- Documentations:
 - <http://www.yoctoproject.org/documentation>
- Bug reports and feature requests:
 - <http://bugzilla.yoctoproject.org>

Join us! Join the community!



Intel Cup Embedded System Design Contest

Legal Information

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY RELATING TO SALE AND/OR USE OF INTEL PRODUCTS, INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT, OR OTHER INTELLECTUAL PROPERTY RIGHT.

Intel may make changes to specifications, product descriptions, and plans at any time, without notice.

All dates provided are subject to change without notice.

Intel is a trademark of Intel Corporation in the U.S. and other countries.

*Other names and brands may be claimed as the property of others.

Copyright © 2012, Intel Corporation. All rights reserved.