

Конфигурирование Linux для процессорных систем Xilinx с помощью среды **Petalinux**

Дмитрий Шадрин,
Инженер по применению Xilinx

Макро Групп – официальный партнёр Xilinx в России

План вебинара

- ◆ Что такое Petalinux, его системные требования и тонкости установки
- ◆ Варианты сборки Linux в PetaLinux
- ◆ Маршрут проектирования системы с использованием PetaLinux
- ◆ Практический пример – конфигурирование Linux для проекта на платформе ZCU104
 - Инициализация проекта
 - Конфигурирование
 - Подключение собственных драйверов и программ
 - Запись на носитель
 - Тестовый запуск Linux на отладочной плате ZCU104.
- ◆ Итоги и выводы, ответы на вопросы

Что такое Petalinux

PetaLinux - это набор программных инструментов от компании Xilinx, позволяющих оптимизировать разработку встроенного Linux для процессорных систем, производимых компанией.

PetaLinux хоть и является отдельным программным пакетом, но это не отдельное решение для разработки встроенного Linux. Рабочий процесс для PetaLinux состоит из нескольких этапов, на которых он опирается на другое программное обеспечение Xilinx, такое как Vivado и Xilinx SDK.

Системные требования Petalinux

Минимальные системные требования для Petalinux 2019.1:

- 8 ГБ ОЗУ (рекомендуемый минимум для инструментов Xilinx)
- Частота ЦПУ 2 ГГц (минимум 8 ядер)
- 100 ГБ свободного места на накопителе
- Поддерживаемые ОС:
 - Red Hat Enterprise Workstation/Server 7.4, 7.5, 7.6 (64-bit)
 - CentOS 7.4, 7.5, 7.6 (64-bit)
 - Ubuntu Linux 16.04.5, 18.04.1 (64-bit)

Тонкости установки Petalinux

Объём дистрибутива: ≈ 8 ГБ

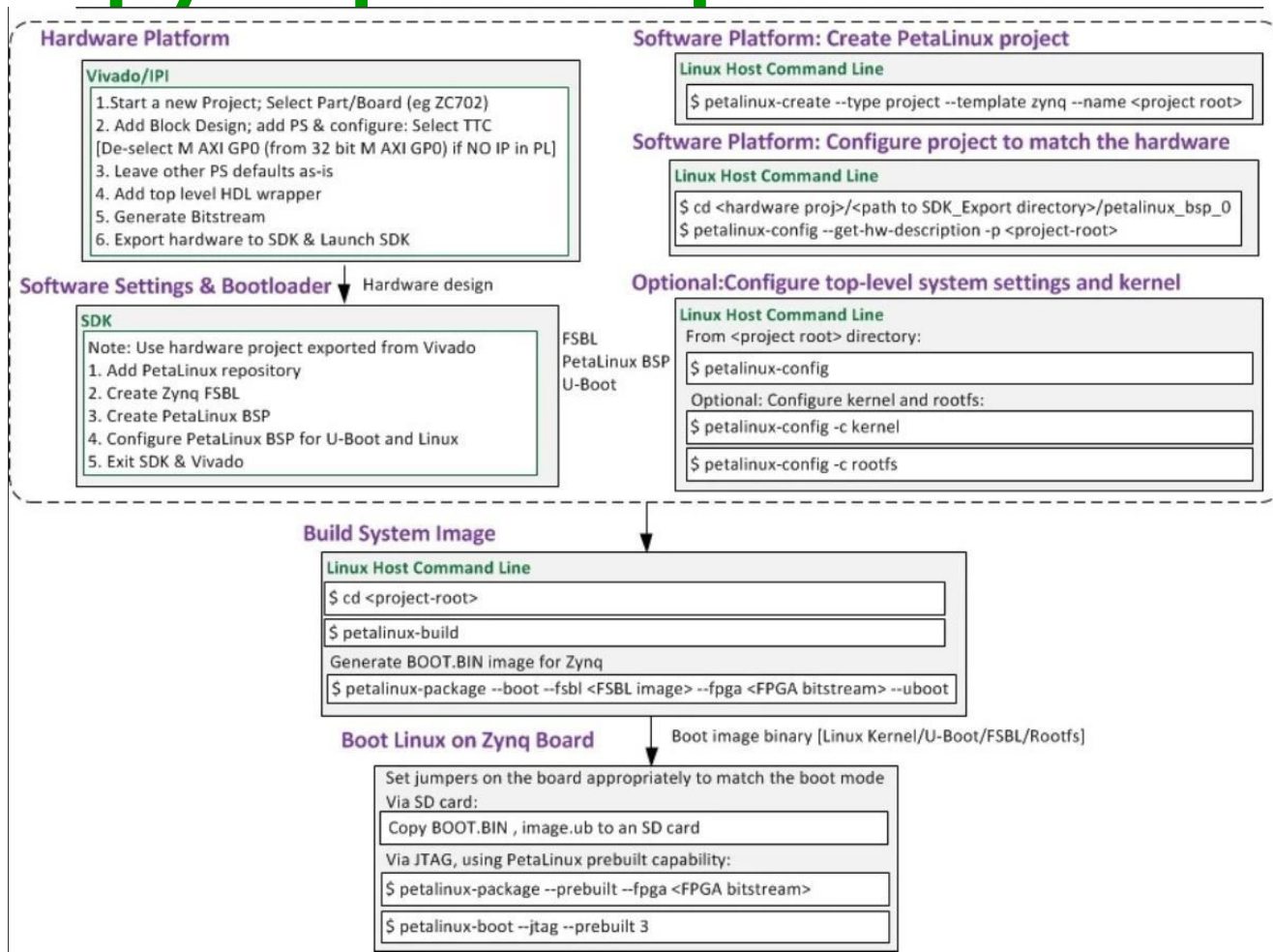
Перед установкой Petalinux необходимо установить дополнительные пакеты следующей командой:

```
sudo apt-get install -y gcc git make net-tools libncurses5-dev tftpd zlib1g-dev libssl-dev flex bison libselinux1 gnupg wget diffstat chrpath socat xterm autoconf libtool tar unzip texinfo zlib1g-dev gcc-multilib build-essential libsdl1.2-dev libglib2.0-dev zlib1g:i386 screen pax gzip gawk
```

Далее выполнить команду:

```
./petalinux-v2019.2-final-installer.run
```

Маршрут проектирования системы



Создание проекта

Создание проекта и компонентов осуществляется с помощью команды `petalinux-create`:

`petalinux-create -t project` - создание проекта

`petalinux-create -t COMPONENT` – создание пользовательских компонентов

Создание проекта при использовании BSP-файлов:

- Перейти в папку, в которой нужно создать проект:
`$ cd /home/user`
- Ввести команду для создания:
`petalinux-create -t project -s <path-to-bsp>`
где `<path-to-bsp>` - путь, по которому лежит BSP-файл

Создание проекта при использовании файлов аппаратной платформы:

- Экспортировать hdf-файл из Vivado;
- Перейти в папку, в которой нужно создать проект:
`$ cd /home/user`
- Ввести команду для создания:
`$ petalinux-create --type project --template <PLATFORM> --name <PROJECT_NAME>`
где `--template <PLATFORM>` - параметр, отвечающий за выбор платформы:
 - `zynqMP` (для UltraScale+™ MPSoC)
 - `zynq` (для серии Zynq-7000)
 - `microblaze` (для MicroBlaze™ CPU)

Создание проекта

Команды:

```
source /opt/Xilinx/2019.2/Petalinux/settings.sh
```

```
cd /home/dmitriy/petalinux
```

Создание проекта:

```
petalinux-create --type project --template zynqMP --name zcu104
```

Создание проекта из bsp-файла:

```
petalinux-create -t project -s /home/dmitriy/Загрузки/xilinx-zcu104-v2019.1-final.bsp
```

Конфигурирование

Конфигурирование сборки осуществляется с помощью команды `petalinux-config`:

`petalinux-config` – настройка параметров системного уровня

`petalinux-config --get-hw-description` – обновление конфигурации оборудования

`petalinux-config -c kernel` – настройка ядра Linux

`petalinux-config -c rootfs` – настройка корневой файловой системы

Последовательность действий:

1) Перейти в папку проекта

```
cd /home/dmitriy/petalinux/xilinx-zcu104-2019.1
```

2) Добавить файл аппаратной платформы

```
petalinux-config --get-hw-description=/home/dmitriy/petalinux
```

3) Настроить меню

Добавление собственной программы и модуля/драйвера

Для добавления собственной программы необходимы следующие действия:

1) Создать файл программы

```
petalinux-create -t apps --name mpsochello --enable
```

2) Найти файл mpsochello.c по адресу

```
<plnx-proj-root>/project-spec/meta-user/recipes-apps/mpsochello
```

3) Заменяем его на собственный файл

4) Редактируем файл mpsochello.bb

Для добавления собственного модуля/драйвера необходимы следующие действия:

1) Создать файл модуля

```
petalinux-create --type modules --name haddriver
```

2) Найти файлы haddriver.c haddriver.bb по адресу

```
<plnx-proj-root>/project-spec/meta-user/recipes-modules/haddriver
```

3) Отредактировать/заменить файлы на собственные

Активировать файлы в меню по команде:

```
petalinux-config -c rootfs
```

Файл mpsochello.bb

```
#
# This file is the mpsochello recipe.
#
SUMMARY = "Simple mpsochello application"
SECTION = "PETALINUX/apps"
LICENSE = "MIT"
LIC_FILES_CHKSUM = "file://${COMMON_LICENSE_DIR}/MIT;md5=0835ade698e0bcf8506ecda2f7b4f302"
SRC_URI = "file://mpsochello.c \
           file://Makefile \
           "
S = "${WORKDIR}"
FILESEXTRAPATHS_prepend := "${THISDIR}/files:"
inherit update-rc.d
INITSCRIPT_NAME = "mpsochello"
INITSCRIPT_PARAMS = "start 99 S ."
do_compile() {
    oe_runmake
}
do_install() {
    install -d ${D}${sysconfdir}/init.d
    install -m 0755 ${S}/mpsochello ${D}${sysconfdir}/init.d/mpsochello
}
FILES_${PN} += "${sysconfdir}/*"
```

Сборка системы

Сборка осуществляется командой:

```
petalinux-build
```

После сборки необходимо сформировать файл BOOT.bin

```
petalinux-package --boot --fsbl images/linux/zynqmp_fsbl.elf --u-boot  
images/linux/u-boot.elf --pmufw images/linux/pmufw.elf --fpga images/linux/*.bit --force
```

Прочие команды:

`petalinux-package` – упаковка для дальнейшего развёртывания системы

`petalinux-boot` – загрузка системы для тестирования

Запись образа на диск

В случае, если файловая система была сконфигурирована как INITRAMFS (монтируется в RAM), то для записи необходимо:

- 1) Отформатировать SD-карту в FAT32
- 2) Записать на неё файлы BOOT.bin и image.ub

В случае, если файловая система была сконфигурирована как INITSD (монтируется в SD-карте), то для записи необходимо:

- 1) Разделить SD-карту на 2 раздела и отформатировать в FAT32 (около 500 МБайт) и ext4.
- 2) Записать на раздел FAT32 файлы BOOT.bin и image.ub, а на раздел ext4 – файл rootfs.tar.gz

Материалы

[ug1144-petalinux-tools-reference-guide.pdf](#)

[ug1157-petalinux-tools-command-line-guide.pdf](#)

[MicroZed Chronicles: PetaLinux Edition — Build Elements & Configuration](#)

[MicroZed Chronicles: PetaLinux Build](#)

Спасибо за внимание!

Компания Макро Групп:

- ◆ официальный партнер Xilinx
- ◆ комплексная поставка электронных компонентов
- ◆ техническая поддержка по всем вопросам применения продукции и ПО Xilinx
- ◆ контрактное производство электроники

Обращайтесь:

- ◆ Dmitriy.Shadrin@macrogroup.ru
- ◆ Dmitry.Khorkov@macrogroup.ru
- ◆ fpga@macrogroup.ru

