

BSP&Buildroot iMX6

Руководство пользователя виртуальной машины для сборки ядра Linux и корневой ФС для изделий starterkit.ru на основе Freescale i.MX6

Виртуальная
машина
запускается как
приложение
Windows

Основана на
Libuntu 14.10

Buildroot – основа
для сборки ядра и
КФС

Основные положения

Виртуальная машина предназначена для сборки ядра Linux и корневой файловой системы (КФС) для изделий starterkit.ru на основе процессора Freescale iMX6.

- **SK-iMX6Q**
- **SK-iMX6S-OEM**
- **SK-IMX6S-OEM-Ind**
- **SK-iMX6S-SODIMM**
- **SK-iMX6S-SODIMM-Ind**
- **SK-iMX6D-SODIMM**
- **Hola-PC**
- **HolaDuo-PC**

Виртуальная машина основана на Libuntu 14.10.

Сборка ядра Linux и КФС интегрирована и осуществляется в одном пакете **Buildroot**. **Buildroot** (www.buildroot.org) это инструмент для сборки КФС, содержит более тысячи приложений и библиотек, состав выбирает сам пользователь.

Для удобства, в виртуальной машине установлены и настроены сервисы для взаимодействия с внешним окружением

- **FTP сервер**
- **TFTP сервер**
- **SSH сервер**
- **Samba** – взаимодействие с сетями Microsoft

Установлен **QtCreator** – среда разработки и отладки Qt приложений, имея Ethernet соединение (TCP/IP канал связи можно организовать практически через любой интерфейс процессора) с платой можно одним щелчком мыши запускать приложение на плате или вести пошаговую отладку.

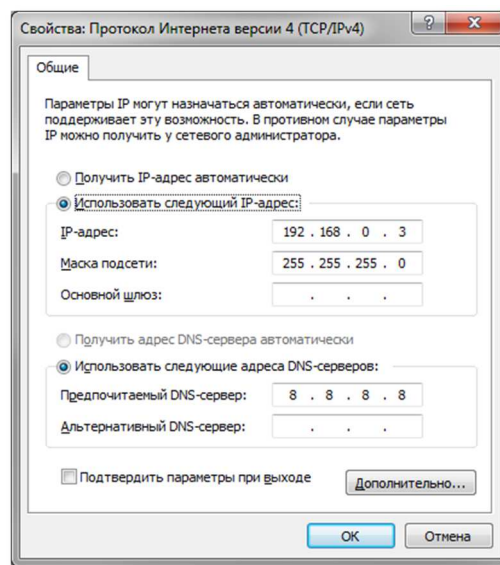
Qt наиболее предпочтительный пакет для написания приложений, т.к. в нем существует огромное количество библиотек и примеров, имеет широкую распространенность, графические приложения Qt работают значительно быстрее и потребляют гораздо меньше ресурсов (особенно памяти) системы в сравнении с приложениями X11.

Настройка виртуальной машины

Перед началом работы необходимо скачать плеер виртуальной машины VMware, бесплатно распространяемый на сайте www.vmware.com.

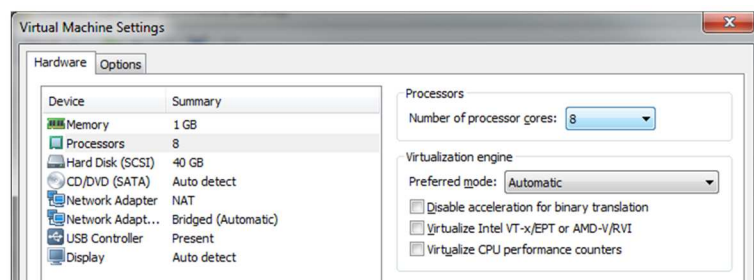
Виртуальная машина имеет 2 сетевых интерфейса:

- 1) NAT – для доступа в Internet
- 2) Bridget – для взаимодействия по локальной сети, необходимо настроить VMware network adapter



Так же необходимо настроить сетевой адаптер PC (или DHCP роутера), так, чтобы присваивался сетевой адрес в группе 192.168.0.XXX (любой кроме 1-3 и 136).

Перед запуском виртуальной машины рекомендуем зайти в ее свойства и выделить количество используемых процессорных ядер (по умолчанию, указан 1), это позволит ощутимо сократить время сборки.



[Скачать плеер VMware](#)

[Настроить виртуальный сетевой адаптер](#)

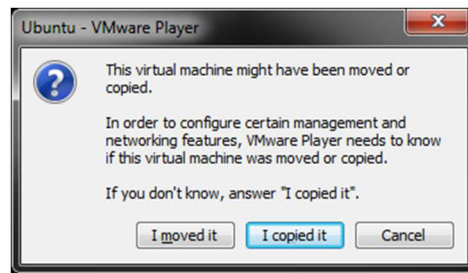
[Настроить сетевой адаптер](#)

[Настроить количество используемых ядер процессора](#)

Первый запуск виртуальной машины

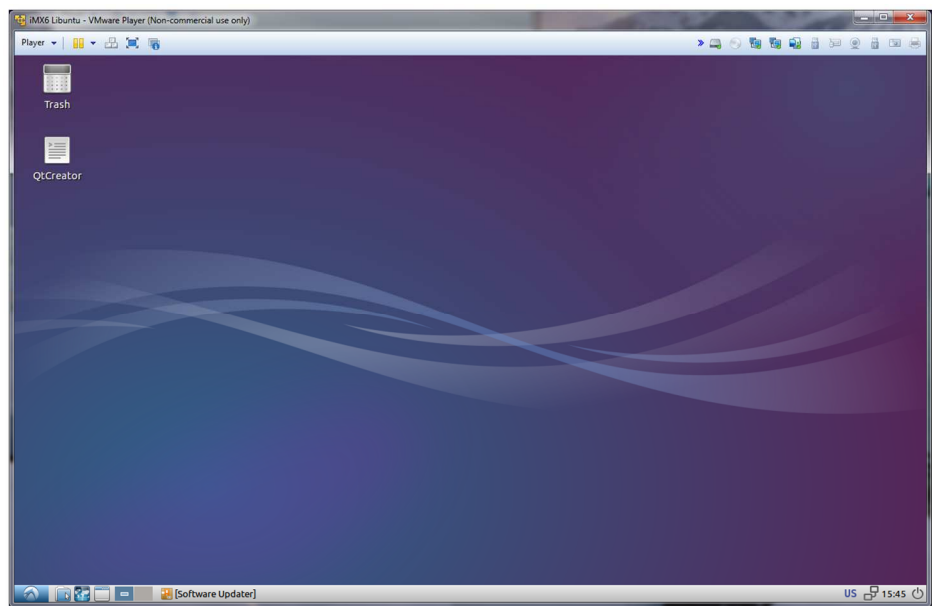
Отвечать:
"I moved it"

При первом запуске виртуальной машины (или после копирования-перемещения), VMware Player спросит:



Нужно ответить «I moved it», это позволит сохранить сетевые настройки.

После загрузки перед вами появится рабочий стол.



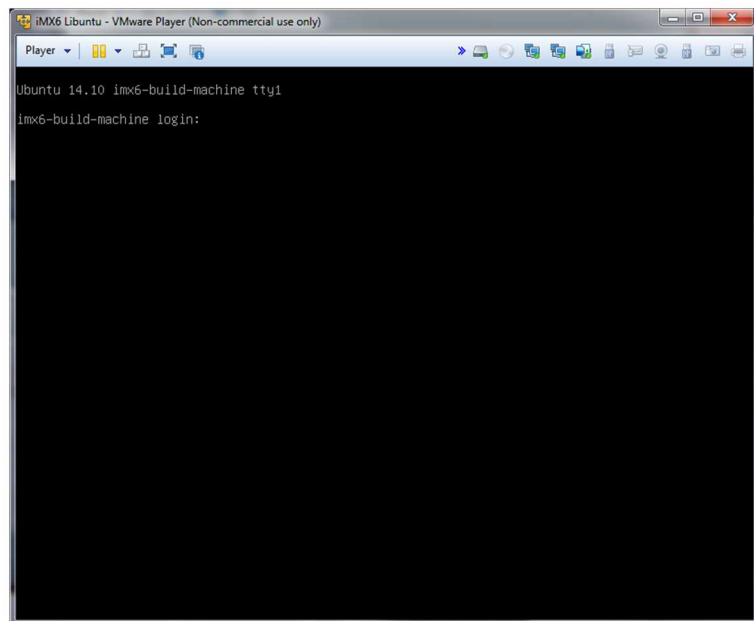
Разрешение экрана можно изменить в разделе **Perfences->Monitor Settings**

Login: user
Pasw: 123456

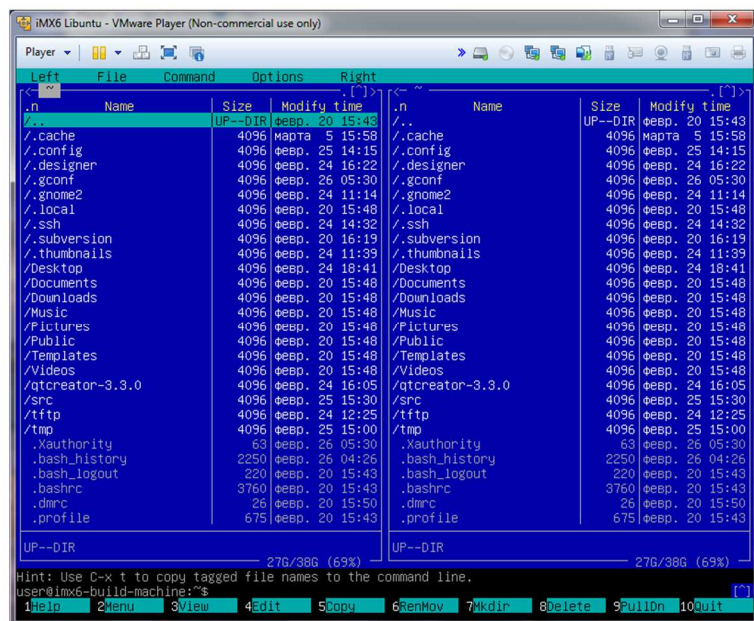
В системе присутствует один пользователь, **логин: user, пароль: 123456**
Суперпользователя в системе нет, для запуска приложений с его привилегиями необходимо использовать **sudo**.

Login: user
Pasw: 123456

Работать с файлами и текстами не всегда удобно через рабочий стол, для переключения в консольный режим необходимо нажать Ctrl+Alt+F(1-6) (Ctrl+Alt+F7 – переключение на графический рабочий стол).

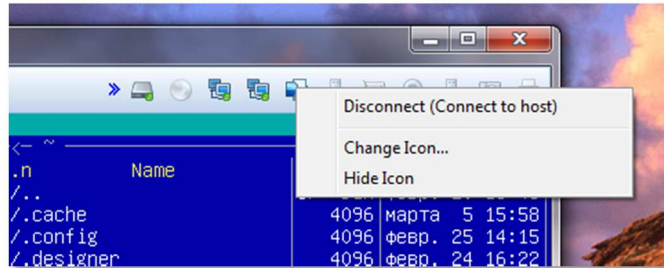


mc – Midnight Commander, файловый менеджер



Карты памяти можно записывать в виртуальной машине

Через раздел **Player>Removable devices** можно подключать-отключать к виртуальной машине различные системные устройства, например, USB устройства, картридеры и т.п. Эта же функция дублируется через графическую панель:



Функция очень полезна, потому как позволяет подключить картридер непосредственно в виртуальную машину и напрямую оперировать с картами памяти.

Во время паузы останавливается системное время!

Нет необходимости выключать виртуальную машину после завершения работы, можно перевести машину в режим паузы, а в следующий раз продолжить работу с момента паузы. **Внимание!** В режиме паузы останавливается системное время, что может негативно сказаться на сборке вновь скачиваемых архивов, во избежание этого следует подстраивать системное время, либо проводить перезагрузку виртуальной машины.

IP виртуальной машины: 192.168.0.2

При правильной настройке сетевых интерфейсов, виртуальная машина должна иметь доступ в Internet, PC должен иметь успешный ping по адресу 192.168.0.2 (адрес Bridget сетевого адаптера в виртуальной машине) и при подключенной плате должен быть успешным ping адреса 192.168.0.136.

Buildroot

`/home/user/src`

Buildroot располагается в папке `/home/user/src/buildroot`

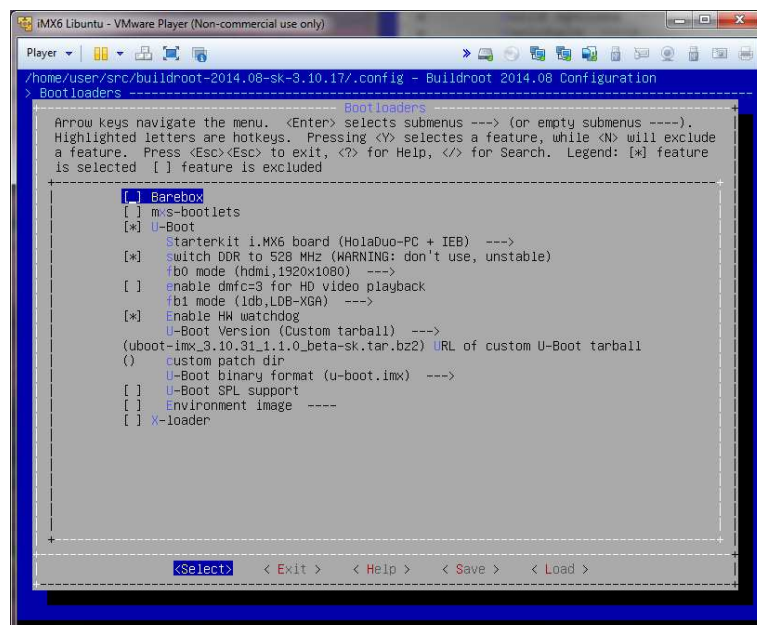
Перед началом сборки необходимо сконфигурировать Buildroot, имеется четыре конфигурации:

1. **imx6sk_min_defconfig**
Содержит минимальный набор включаемых пакетов, как следствие, небольшой размер КФС и малое требуемое время для сборки
2. **imx6sk_xbmc_defconfig**
Конфигурация для сборки медиа-центра XBMC
3. **imx6sk_gst_defconfig**
Конфигурация для сборки Gstreamer
4. **imxx6sk_qt5gst_defconfig**
Основная конфигурация, кроме основных сервисов и утилит, включает в себя Gstreamer и Qt5.

- > `cd /home/user/src/buildroot-2014.08-sk-3.10.17`
- > `make clean`
- > `make imx6sk_qt5gst_defconfig`
- > `make menuconfig`

Зайдите в подменю «Bootloadres».

`make menuconfig`



В ниспадающем меню «**Starterkit i.MX6 board**» выберите необходимый вариант.

Переключатель «**switch DDR to 528 MHz (WARNING: don't use, unstable)**» на данный момент (03.2015) актуален только для платы SK-iMX6Q.

Выберите режимы работы фреймбуфера «**fb0/1 mode**», выбрать требуемый графический модуль расширения.

watchdog важен

Переключатель «**Enable HW watchdog**» настоятельно рекомендован для включения, особенно для изделий на iMX6Q/D, на данный момент имеется баг – при программном сбросе (команда reboot) система может не загрузиться, watchdog исключит эту проблему.

Далее можно выйти из меню с сохранением изменений и приступить к сборке.

```
> make
```

Первая сборка не быстрый процесс

Процесс сборки зависит от производительности вашего PC, может занять несколько часов, типичное время сборки с imx6sk_qt5_gst_defconfig на 4-х ядерном PC составляет 2 часа (при выделении 4 ядер для виртуальной машины). При повторных запусках будут собираться только вновь добавленные пакеты, что не требует много времени. **Важно!** При исключении пакета из сборки он не удаляется, остается в сборке КФС до чистки.

В результате сборки в папке **output** появится несколько новых папок:

- **build** – содержит рабочие папки собираемых пакетов, а также ядро и загрузчик
- **target** – результат сборки, скопировав сюда файл, он появится в образах КФС после выполнения make
- **images** – ядро, загрузчик, архив КФС ...

В корне buildroot папки присутствует три вспомогательных скрипта:

Изменили исходный код или состав ядра – выполните kernel_rebuild.sh

- **kernel_rebuild.sh** – пересобирает ядро Linux
При редактировании исходников ядра Linux в папке output/build/linux-custom команда make не пересобирает ядро, скрипт устраняет данный недостаток. **make linux-menuconfig** позволяет конфигурировать состав ядра Linux.
- **uboot_rebuild.sh** – пересобирает загрузчик uboot
При изменении типа платы или прочих опций в подменю «**Bootloadres**», для применения изменений в исполняемых файлах необходимо запускать этот скрипт.

Изменили тип платы и т.п. – выполните uboot_rebuild.sh

Busybox

Большинство системных утилит реализованы не отдельными программами, а специальным многофункциональным средством Busybox, в папке /bin находятся не программы, а ссылки на Busybox с указанием требуемого вызова.

Изменить состав Busybox можно командой **make busybox-menuconfig**, а перед сборкой удалить папку output/build/busybox...

make clean (чистка) – удаляет все результаты сборки

make clean удалит все изменения и настройки в ядре и пакетах

Важно!!! **make clean** удалит все в папке output, т.е. все изменения в ядре Linux и прочих пакетах, поэтому стоит с внимательно относиться к применению этой команды и заблаговременно сохранять ваши изменения. Для того, чтобы изменения в ядре Linux не были безвозвратно утеряны после чистки, необходимо перед этой командой создать архив ядра и скопировать его в папку downloads:

```
> cd /home/user/src/buildroot-2014.08-sk-3.10.17/output/build/linux-custom
> make clean
> tar cjf /home/user/src/buildroot-2014.08-sk-3.10.17/downloads/linux-
imx_3.10.17_1.0.2_ga-sk.tar.bz2 ./*
```

Подготовка загрузочной карты памяти

SK-iMX6Q

Hola-PC

HolaDuo-PC

Данный раздел относится к платам способным загружаться с micro SD карт памяти:

- **SK-iMX6Q**, предварительно необходимо сконфигурировать плату на загрузку с uSD карты памяти – замкнуть J6 и разомкнуть J7,J8
- **Hola-PC**
- **HolaDuo-PC**

Внимание! Если изменили тип платы или настройки подраздела «**Bootloadres**», необходимо выполнить скрипт **uboot_rebuild.sh**.

Предварительно подключите картридер к виртуальной машине, см. раздел «Первый запуск виртуальной машины», убедитесь, что карта появилась в устройствах виртуальной машины:

```
> ls /dev/sd*
```

```
/dev/sda /dev/sda1 /dev/sda2 /dev/sda5 /dev/sdb /dev/sdb1
```

Если к виртуальной машине не подключены другие накопители, устройство **sdb** и есть карта памяти.

Для подготовки карты памяти необходимо выполнить скрипт **mk-sd.sh**, а в качестве аргумента указать имя устройства карты памяти:

```
> cd /home/user/src/buildroot-2014.08-sk-3.10.17
```

```
> sudo ./mk-sd.sh /dev/sdb
```

Команда `sudo` потребует ввести пароль суперпользователя – 123456, в результате должен получиться следующий лог операции:

```
create the bootable SD card for hola_dual_ieb on device /dev/sdb
```

```
delete partition table          [OK]
create primary partition        [OK]
prepare u-boot.imx              [OK]
make ext4 fs                    [OK]
mount partition                  [OK]
extract rootfs.tar              [OK]
umount partition                [OK]
remove tmp mount point          [OK]
```

```
all commands are completed without errors, SD is ready
```

Карта памяти готова к использованию.

Подготовка SK-iMX6-SODIMM

Данный раздел относится к модулям, загрузка которых осуществляется со встроенной eMMC flash:

SK-iMX6S-SODIMM

SK-iMX6S-SODIMM-Ind

SK-iMX6D-SODIMM

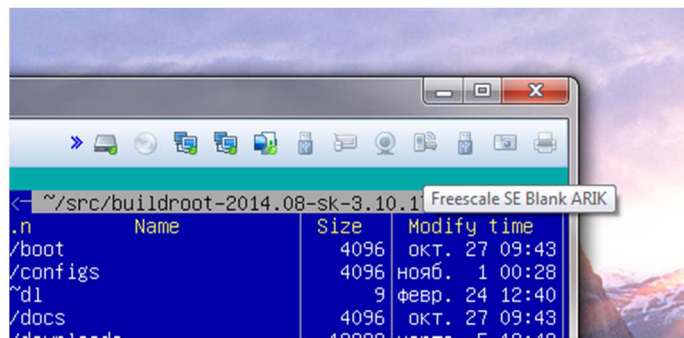
- SK-iMX6S-SODIMM
- SK-iMX6S-SODIMM-Ind
- SK-iMX6D-SODIMM

Дальнейшие инструкции относятся к модулям совместно используемым с материнской платой **SK-iMX6-MB-SODIMM**, от которой лучше отключить всю периферию.

Внимание! Если изменили тип платы или настройки подраздела «**Bootloadres**», необходимо выполнить скрипт **uboot_rebuild.sh**.

Замкнуть J1

- Необходимо замкнуть джампер J1, что укажет модулю перейти в режим загрузки по USB.
- Подключить USB кабель к разъему X19, включить питание. При первом включении, в системе установится новое HID устройство.



- Подключить **Freescale SE Blank ARIK** устройство к виртуальной машине. **Важно!!!** После этого нужно обязательно переключиться в саму виртуальную машину, т.к. процесс загрузки происходит в несколько этапов и первое запущенное по USB приложение изменит тип устройства, если в этот момент вы находитесь не в виртуальной машине – требуемое устройство не подключится.

Успешностью установления USB соединения с модулем является сообщение появляющееся через несколько секунд в консоли:

```
user@imx6-build-machine:~/src/buildroot-2014.08-sk-3.10.17$
user@imx6-build-machine:~/src/buildroot-2014.08-sk-3.10.17$
user@imx6-build-machine:~/src/buildroot-2014.08-sk-3.10.17$
user@imx6-build-machine:~/src/buildroot-2014.08-sk-3.10.17$ [21848.825616] cdc_acm 1-2:1.0: This device cannot do calls on its own. It is not a modem.
```

После подключения устройства, необходимо переключится в виртуальную машину!

После этого необходимо выяснить имя устройства, которое назначено подключенной карте модуля:

```
> ls /dev/sd*
```

```
/dev/sda /dev/sda1 /dev/sda2 /dev/sda5 /dev/sdb /dev/sdb1
```

Если к виртуальной машине не подключены другие накопители или картридер, устройство **sdb** и есть карта памяти модуля.

Для подготовки карты памяти необходимо выполнить скрипт **mk-sd.sh**, а в качестве аргумента указать имя устройства карты памяти:

```
> cd /home/user/src/buildroot-2014.08-sk-3.10.17
```

```
> sudo ./mk-sd.sh /dev/sdb
```

Команда `sudo` потребует ввести пароль суперпользователя – 123456, в результате должен получиться следующий лог операции:

```
create the bootable SD card for hola_dual_ieb on device /dev/sdb
```

```
delete partition table      [OK]
create primary partition    [OK]
prepare u-boot.imx         [OK]
make ext4 fs                [OK]
mount partition             [OK]
extract rootfs.tar          [OK]
umount partition           [OK]
remove tmp mount point      [OK]
```

```
all commands are completed without errors, SD is ready
```

[Разомкнуть J1](#)

Модуль запрограммирован и готов к работе, разомкните J1.

Подготовка SK-iMX6-OEM, SK-iMX6Q

Данный раздел относится к платам и модулям, загрузка которых может осуществляться со встроенной NAND flash:

- SK-iMX6S-OEM
- SK-iMX6S-OEM-Ind
- SK-iMX6S-OEM-WiFi
- SK-iMX6Q

Предварительно необходимо подготовить загрузочную microSD карту памяти, см. раздел «Подготовка загрузочной карты памяти».

Внимание! Если изменили тип платы или настройки подраздела «**Bootloadres**», необходимо выполнить скрипт **uboot_rebuild.sh**.

Замкнуть J1

Login: root
Pasw: root

- На модуле необходимо замкнуть J1 – переключает загрузку на micro SD карту памяти, для платы Sk-iMX6Q необходимо замкнуть J6.
- Вставить подготовленную карту памяти в разъем модуля X1, включить питание.
- Дождаться завершения загрузки, зайти на модуль через консольный порт RS232 или по сети (SSH – по адресу 192.168.0.136), логин:root, пароль:root
- Выполнить скрипт /root/root2nand.sh, результат работы:

/root/root2nand.sh

```
# /root/root2nand.sh
format root partition mtd1      [OK]
make volume <rootfs>           [OK]
mount ubifs                     [OK]
copy root fs to NAND           [OK]
make directories               [OK]
umount ubifs                   [OK]
detach mtd1                    [OK]
erase boot partition           [OK]
prepare u-boot-nand.imx        [OK]
```

all commands are completed without errors, NAND is ready

Разомкнуть J1

Модуль запрограммирован и готов к работе, разомкните J1.

QtCreator

QtCreator – среда разработки позволяющая одним щелчком мыши собирать и запускать приложения пользователя на плате, а так же вести пошаговую отладку.

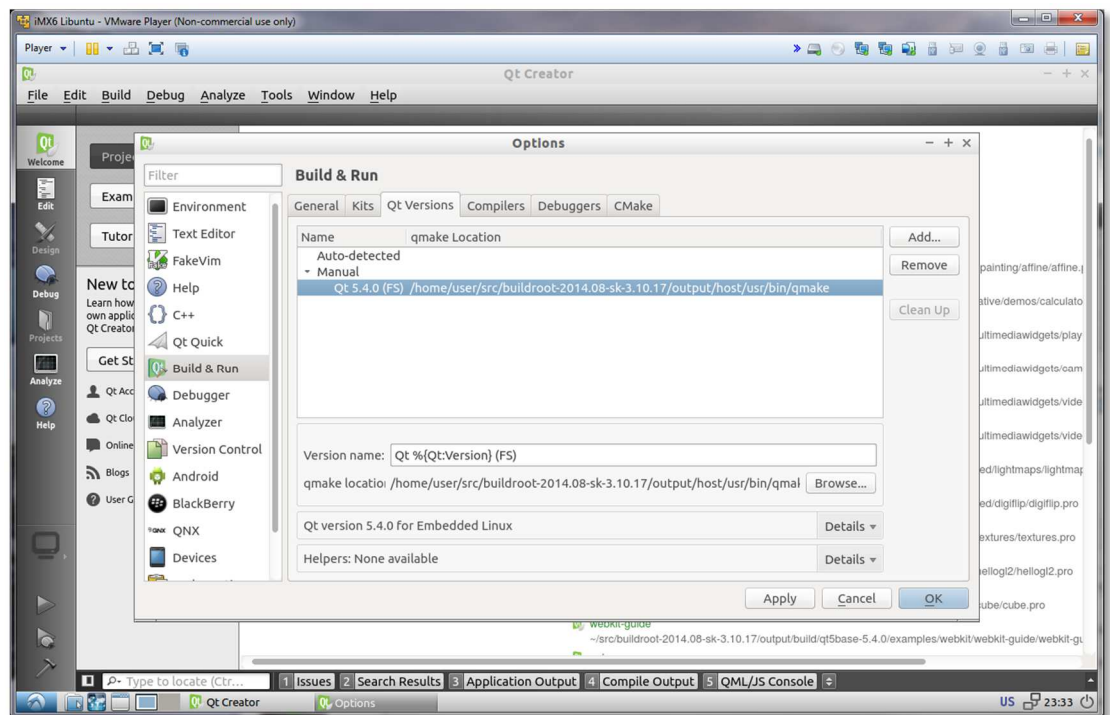
Так сложилось, что настройки Qt проектов взаимосвязаны с рабочими папками output/build, которые очищаются перед подготовкой к публикации виртуальной машины, поэтому необходимо настроить QtCreator перед началом работы.

Примечание. Вполне возможно, когда Вы читаете этот раздел, мы решили не экономить Ваш трафик (примерно 1Г) и публикуем виртуальную машину без предварительной чистки от чего настройки QtCreator остаются не удаляются.

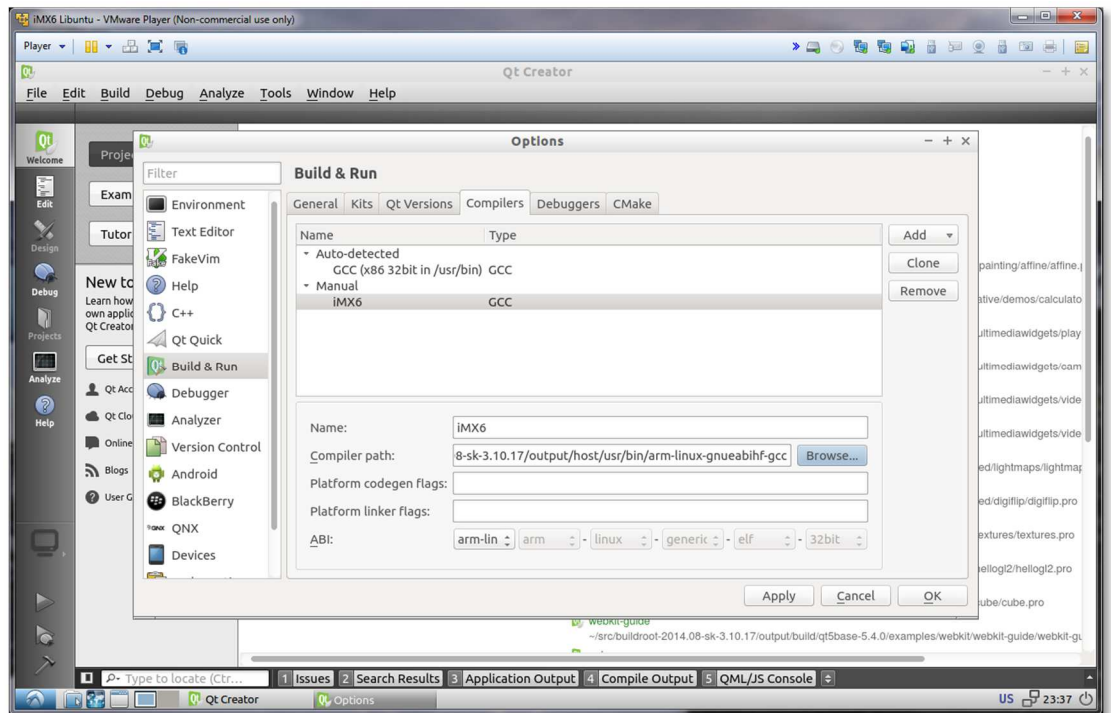
Настроить QtCreator

Предварительно соберите Buildroot с конфигурацией imx6sk_qt5gst_defconfig и запрограммируйте плату/модуль. Подключите Ethernet кабель и проверьте наличие соединения с платой, должен быть успешным ping из виртуальной машины с адресом 192.168.0.136.

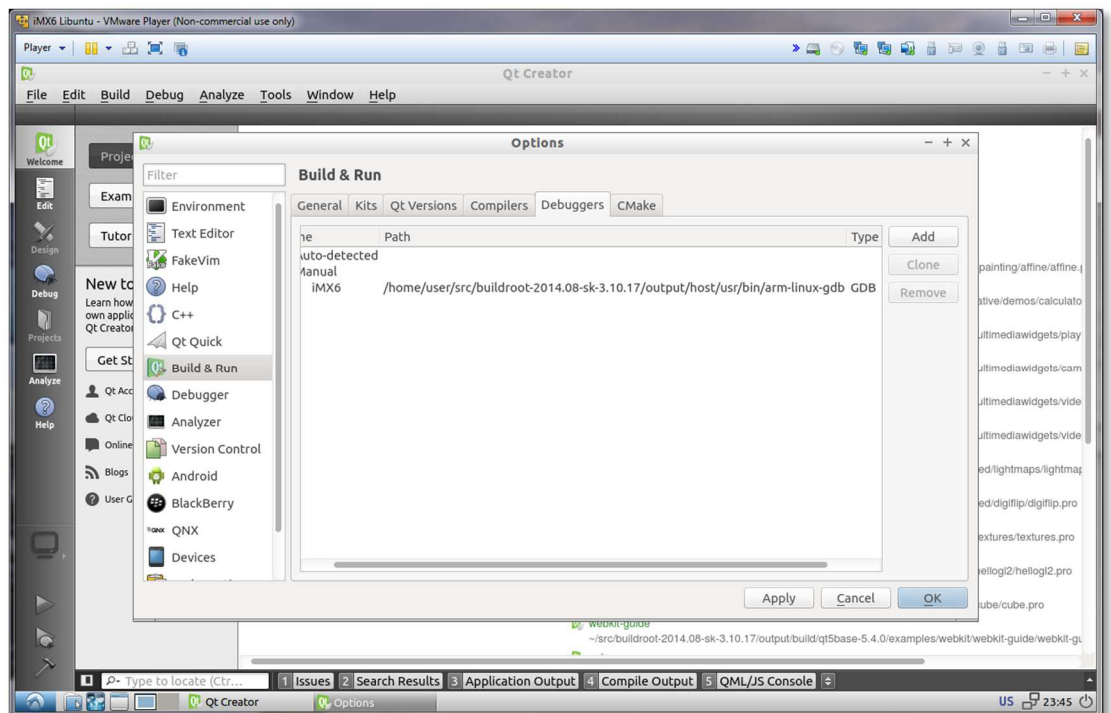
Запустите скрипт **QtCreator** на рабочем столе, зайдите в меню **Tools->Options->Build & Run**, закладка **Qt Version** – добавьте путь к **qmake**, как показано на рисунке:



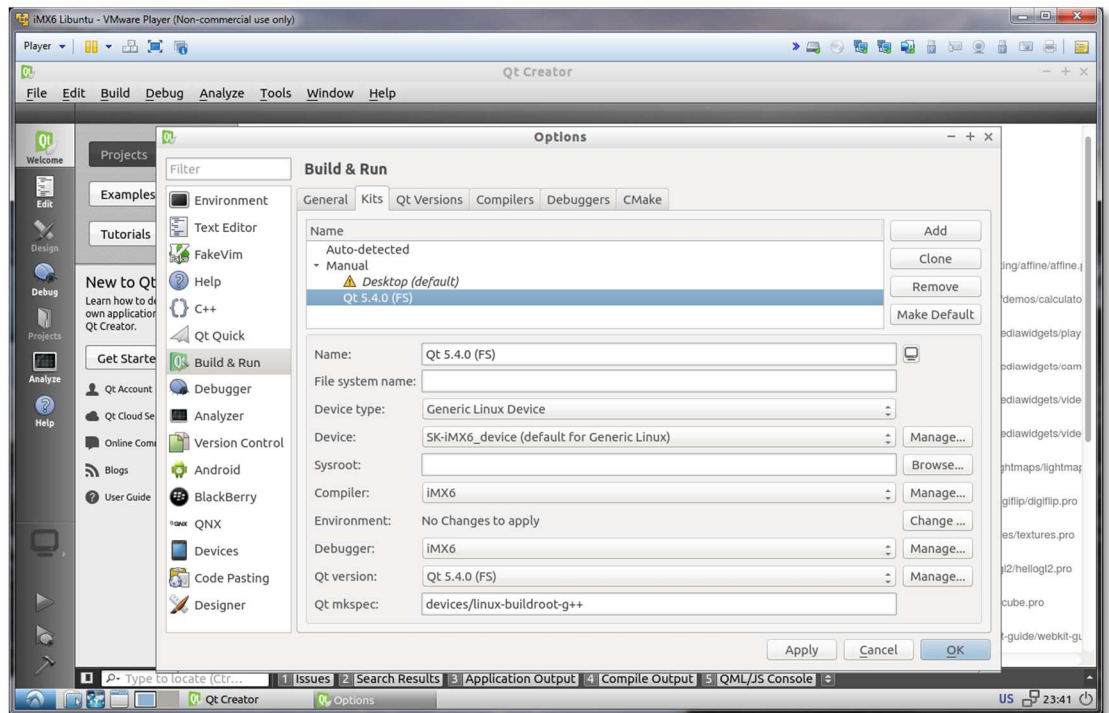
В закладке **Compilers** добавьте компилятор:



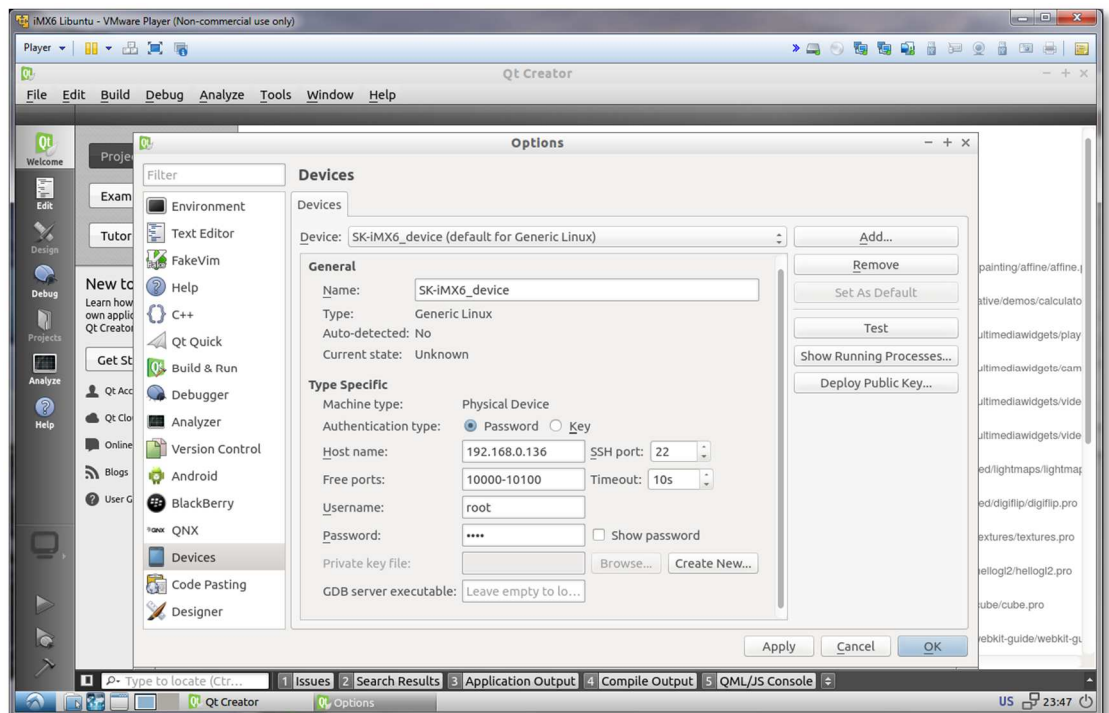
В закладке **Debuggers** добавьте отладчик:



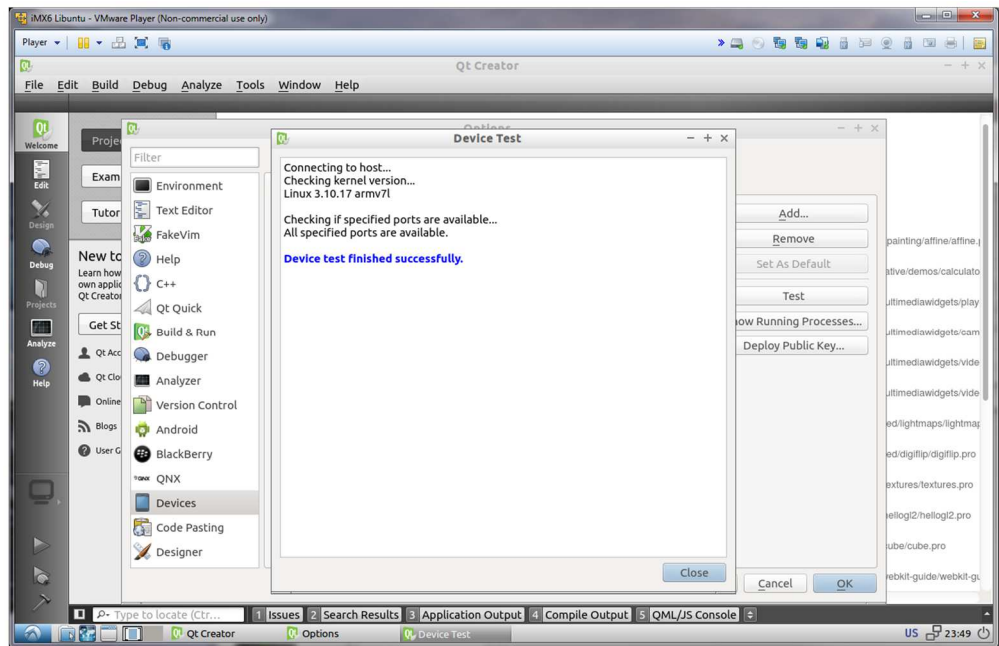
В закладке **Kits** добавить правило, как на рисунке:



Перейти в раздел **Tools->options->Devices** и добавить **Generic Linux Device**, указать IP адрес **192.168.0.136**, login:root, password:root



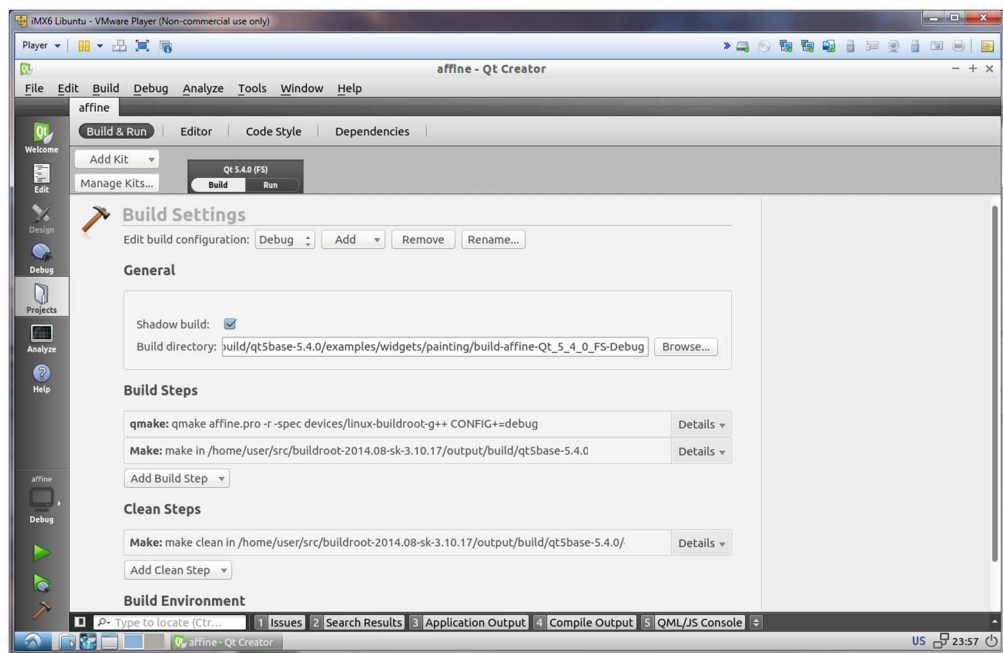
Проверить соединение с платой можно нажав кнопку **Test**.



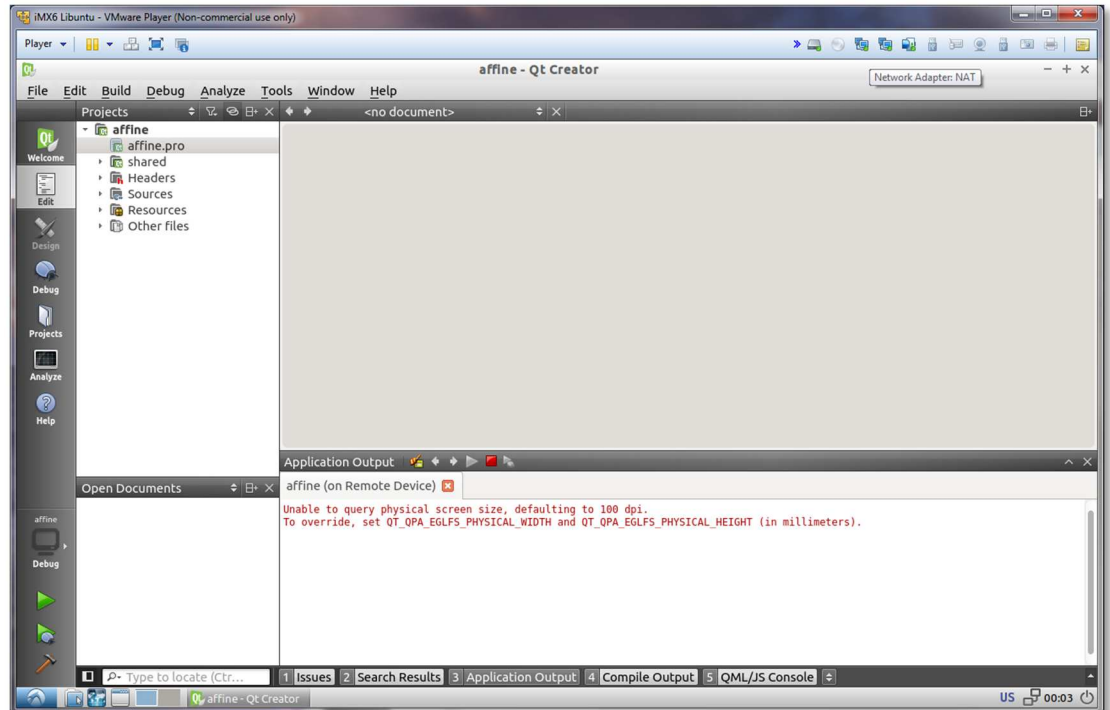
Итак, QtCreator настроен для сборки и соединения с платой, далее необходимо открыть демонстрационный проект Qt и проверить настройки.

File->Open File or Project... выберите `/home/user/src/buildroot-2014.08-sk-3.10.17/output/build/qt5base-5.4.0/examples/widgets/painting/affine/affine.pro`

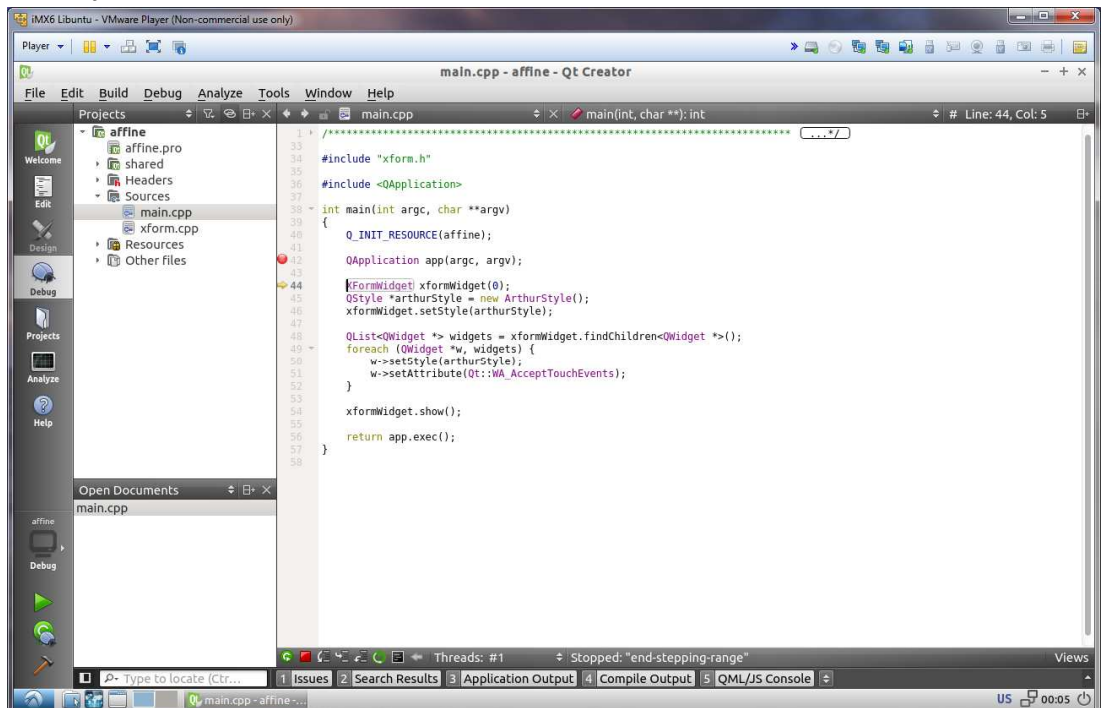
В разделе **Projects** убедитесь, что используется нужный **Kit**:



Далее **Build->Rebuild All**, по завершении, нажать кнопку Run, на плату скопируется и запустится приложение:



Предварительно установив точки останова в исходном коде, можно вести пошаговую отладку:



Сведения об организации

www.starterkit.ru

Россия 426000, г.Ижевск
ул.Телегина д.30

Тел. (+7 3412)478-448