

Модульная Сетевая Операционная Система Телекоммуникационного Оборудования –
SDK CITOS

Руководство программиста. Часть 1

643.ТЛМБ.13001-01 33 01-1

Листов 7

Содержание

1	Введение	3
2	Назначение и условия применения программы	3
3	Характеристики программы	4
3.1	Модули загрузки BSP	4
3.2	Библиотека API	4
3.3	Уровень пользовательских приложений	5
3.4	Первоначальный загрузчик	5
4	Установка SDK CITOS на станцию сборки	5
4.1	Установка компилятора и бинарных компонентов BSP	5
4.2	Установка исходных кодов ПО прикладного уровня	5
4.3	Внесение изменений в исходные коды прикладного уровня	6
4.4	Настройка и сборка ПО	6

1 Введение

Руководство программиста состоит из двух частей.

Часть 1 - содержит инструкции по установке, настройке SDK CITOS, а также базовый набор действий по сборке двоичного образа под целевую платформу.

Часть 2 - содержит описание функциональных характеристик SDK CITOS, решение задач конфигурирования под целевую платформу, требования к ресурсам.

2 Назначение и условия применения программы

В документе представлено введение в средство разработки (SDK) с точки зрения разработчика программного обеспечения (ПО).

Документ включает общее описание архитектуры, параметры интеграции, среду разработки, настройку и сборку ПО.

Начиная с данной версии 1.0.0, стек ПО работает в составе операционной системы Linux.

Разработчику необходимо обладать следующими навыками и инструментами:

- Языки программирования C/C++ и система сборки на основе файлов Makefile. Языки разметки и программирования HTML/CSS/JS для изменения веб-интерфейса.
- Средой разработки Linux и достаточный опыт работы с оболочкой Linux.
- Для создания новых образов из исходных текстов (включая загрузчик, BSP и приложение) требуется 64-разрядная машина Linux с по меньшей мере 8 ГБ оперативной памяти, 50 ГБ дискового пространства и 4-8 ядрами процессора. В этом документе в качестве станции разработки используется Ubuntu 18.04 LTS (OC).
- Необходим доступ к серверу TFTP или HTTP, который можно использовать для обновления ПО.
- Терминал RS232 для доступа к целевому оборудованию построенного на основе специализированного набора микросхем (ASIC) (необходим для отладки без подключения по локальной вычислительной сети ЛВС).

Дополнительное ПО разработки.

Кроме операционной системы (OC) в станцию разработки должно быть установлено следующее дополнительное программное обеспечение:

- bc - консольный калькулятор;
- build-essential - набор компиляторов;
- bzip2 - архиватор ZIP;
- coreutils - набор утилит ядра;
- cpio - архиватор СPIO;
- findutils - набор утилит для поиска;
- gawk - потоковый редактор;
- git - система контроля версий;
- grep - утилита поиска внутри файлов;
- gzip - архиватор ZIP;
- libc6-i386 - библиотека обработки системных вызовов;
- libcrypt-openssl-rsa-perl - библиотека шифрования для системы perl;
- libncurses5-dev - библиотека консольной графики;
- patch - утилита наложения изменений;
- perl - интерпретатор языка PERL;
- python - интерпретатор языка Python;

- ruby - интерпретатор языка Ruby;
- sed - потоковый редактор Sed;
- squashfs-tools - утилиты настройки файловой системы Squashfs;
- tcl - интерпретатор языка Tcl;
- tar - архиватор TAR;
- wget - утилита загрузки файлов;
- libyaml-tiny-perl - библиотека поддержки файлов конфигурации YAML для интерпретатора Perl;
- libcgi-fast-perl - библиотека поддержки CGI для интерпретатора Perl;
- ruby-parslet - библиотека парсинга файлов конфигурации Ruby;
- libstdc++6 - стандартная библиотека языка C++.

Установить приведенный набор утилит и библиотек возможно с помощью следующей команды (для систем на базе Debian):

```
$ sudo apt-get install bc build-essential bzip2 coreutils cpio findutils gawk git grep gzip libc6-i386 libcrypt-openssl-rsa-perl libncurses5-dev patch perl python ruby sed squashfs-tools tcl tar wget libyaml-tiny-perl libcgi-fast-perl ruby-parslet libstdc++6
```

3 Характеристики программы

SDK состоит из различных компонентов. Все компоненты необходимы для создания работающего образа встраиваемого программного обеспечения для оборудования на основе специализированного набора микросхем. В этом разделе будет дан обзор различных компонентов и объяснено, какую роль они выполняют. Разработчику может потребоваться изменить один или несколько компонентов для поддержки нового функционала и настройки различных моделей интеграции.

3.1 Модули загрузки BSP

Пакет BSP содержит исходные коды программ для первоначальной загрузки системы. Также включает в себя утилиты и инструменты необходимые для создания исполняемого кода и компоненты необходимые для целевой установки на конечное оборудование. В состав набора инструментов входят:

- кросс-компиляторы - gcc, g++;
- система сценариев сборки - make, automake;

SDK содержит несколько вариантов BSP, оптимизированных для платформ, собранных на различных специализированных наборах микросхем (ASIC). BSP требует изменения или дополнения при отличии аппаратной части целевой платформы от референсного дизайна. В случае отсутствия изменений могут быть использованы бинарные сборки BSP непосредственно от референсного дизайна. Следует иметь ввиду, что сборка BSP из исходных кодов может занимать значительное время, поэтому рекомендуется с осторожностью относиться к необходимости изменения аппаратной части целевой платформы.

3.2 Библиотека API

API - набор программных файлов необходимых для предоставления взаимодействия программного обеспечения прикладного уровня с ядром (BSP) основной системы.

3.3 Уровень пользовательских приложений

Уровень пользовательских приложений включает в себя набор компонентов пользовательского уровня, необходимых для функционирования целевой платформы в требуемом сетевом окружении.

3.4 Первоначальный загрузчик

Первоначальный загрузчик предоставляет собой часть BSP, которое запускается на целевой платформе при подаче питания. Загрузчик отвечает за настройку процессора, контроллера памяти, загрузку ядра Linux в память. Загрузчик основан на базовой версии свободного программного обеспечения Das Uboot с внесенными платформозависимыми изменениями. Несмотря на то, что загрузчик является универсальным для множества целевых платформ, в некоторых случаях может потребоваться его изменение. Эти изменения должны быть сделаны в исходных текстах модулей Uboot и скомпилированы соответствующим кросс-компилятором.

4 Установка SDK CITOS на станцию сборки

В данном разделе описывается процесс развертывания SDK CITOS на станции сборки, внесение изменений в исходные коды и сборка (компиляция) под целевую платформу.

4.1 Установка компилятора и бинарных компонентов BSP

Для установки предсобранных бинарных образов и исходных кодов BSP референсных платформ необходимо распаковать архив в каталог */opt* на станции сборки:

```
$ cd /opt
$ mkdir -p opt/mscc
$ cd /opt/mscc
$ cp /media/CITOS-BSP_1.0..tar.bz2 /opt/mscc/
$ tar -xvjf CITOS-BSP_1.0..tar.bz2
```

4.2 Установка исходных кодов ПО прикладного уровня

Для установки исходных кодов ПО прикладного уровня на станцию сборки, распакуйте архив в каталог */home/user*. Здесь и далее по тексту текущий пользователь системы обозначен как *user*.

```
$ cd /home/user
$ mkdir -p /home/user/citos
$ cp /media/CITOS-APP_1.0.tar.bz2 /home/user/citos/
$ cd /home/user/citos/
$ tar -xvjf CITOS-APP_1.0.tar.bz2
```

4.3 Внесение изменений в исходные коды прикладного уровня

Для внесения изменений перейдите в каталог с установленным ПО:

```
$ cd /home/user/citos/CITOS-APP_1.0/src
```

Выберите компонент и внесите необходимые изменения пользуясь редактором исходных текстов.

4.4 Настройка и сборка ПО

SDK содержит подготовленные конфигурационные файлы описания референсных платформ с расширением **.mk*, которые находятся в каталоге *configs*.

Таблица 1 — Сценарии сборки

Сценарий	Платформа	Команда
<i>Reference_equipment_var1.mk</i>	RFE_VAR1	<pre>\$ tools/set_build_config.rb -c configs/Reference_equipment_var1.mk</pre>
<i>Reference_equipment_var2.mk</i>	RFE_VAR2	<pre>\$ tools/set_build_config.rb -c configs/Reference_equipment_var2.mk</pre>
<i>Reference_equipment_var3.mk</i>	RFE_VAR3	<pre>\$ tools/set_build_config.rb -c configs/Reference_equipment_var3.mk</pre>

Таблица 2 — Наименования образов для загрузки

Платформа	Наименование
RFE_VAR1	<i>obj/Reference_equipment_var1.mfi</i>
RFE_VAR2	<i>obj/Reference_equipment_var2.mfi</i>
RFE_VAR3	<i>obj/Reference_equipment_var3.mfi</i>

Для настройки системы сборки под требуемую платформу выполните следующие шаги:

1. Перейдите в каталог ПО *src/build* :

```
$ cd src/build
```

2. Выполните очистку от артефактов предыдущих сборок:

```
$ make mrproper
```

3. Установите требуемый сценарий (см. таблицу 1). Например, для оборудования RFE_VAR1 необходимо выполнить следующую команду:

```
$ tools/set_build_config.rb -c configs/Reference_equipment_var1.mk
```

Запустите сборку выполнив команду *make*:

```
$ make
```

643.ТЛМБ.13001-01 33 01-1

Собранный образ, для выбранной платформы (см. табл. 2) будет находится в каталоге:

obj/Reference_equipmnet_var1.mfi

Данный образ готов для загрузки в платформу.