

# Отладочная плата HawkBoard на базе процессора OMAP-L138. Часть 2

Игорь ГУК  
gook\_igor@mail.ru

В первой части статьи [1] были рассмотрены основные характеристики отладочной платы HawkBoard [2]. Кроме того, было показано, как загрузить в память HawkBoard готовые образы ядра и файловой системы, а затем запустить операционную систему (ОС) Linux на отладочной плате. Во второй части статьи рассматривается создание инструментария, требуемого для самостоятельного компилирования основных программных компонентов (загрузчика, ядра и файловой системы).

Это более сложный этап, так как все действия необходимо выполнять на базовом ПК с установленной операционной системой Linux. При работе с операционной системой Windows, что наиболее часто встречается при разработке софта для цифровых сигнальных процессоров компании Texas Instruments, можно рекомендовать использовать виртуальную машину. При написании статьи использовалась виртуальная машина VirtualBox [3], куда была установлена операционная система Ubuntu 10.4 [4].

Описания инсталляции программы виртуализации VirtualBox и создания гостевой ОС выходит за рамки статьи. Всю необходимую информацию можно найти в Интернете, например здесь [5].

В процессе формирования требуемого инструментария для создания ядра и файловой системы Linux требуется выполнить два этапа:

1. Установить необходимый набор программных средств от компании Texas Instruments (TI) [6], так называемый SDK.
2. Установить кросс-компилятор для ARM-ядра компании Code Sourcery [9].

На первом этапе прежде всего нужно загрузить следующие файлы SDK [7]:

- `OMAP-L138_setuplinux_#.#.#.bin`,
- `bios_setuplinux_#.#.#.bin`,
- `xdctools_setuplinux_#.#.#.bin`,
- `ti_cgt_c6000_#.#.#_setup_linux_x86.bin`,
- `cs1omapl138_#.#.#_setup_linux.bin`.

На рис. 1 представлена страница загрузки необходимых компонентов [7] (после загрузки страницы необходимо прокрутить ее вниз).

Перед скачиванием файлов требуется зарегистрироваться на сайте TI, так как при загрузке запрашиваются логин и пароль (рис. 2).

Пакет SDK включает такие компоненты, как Codec Engines, DSP/BIOS Link, xDAIS, а также заголовочные файлы xDM, Framework Components и т.д. Инсталляторы различных компонентов имеют по умолчанию различ-

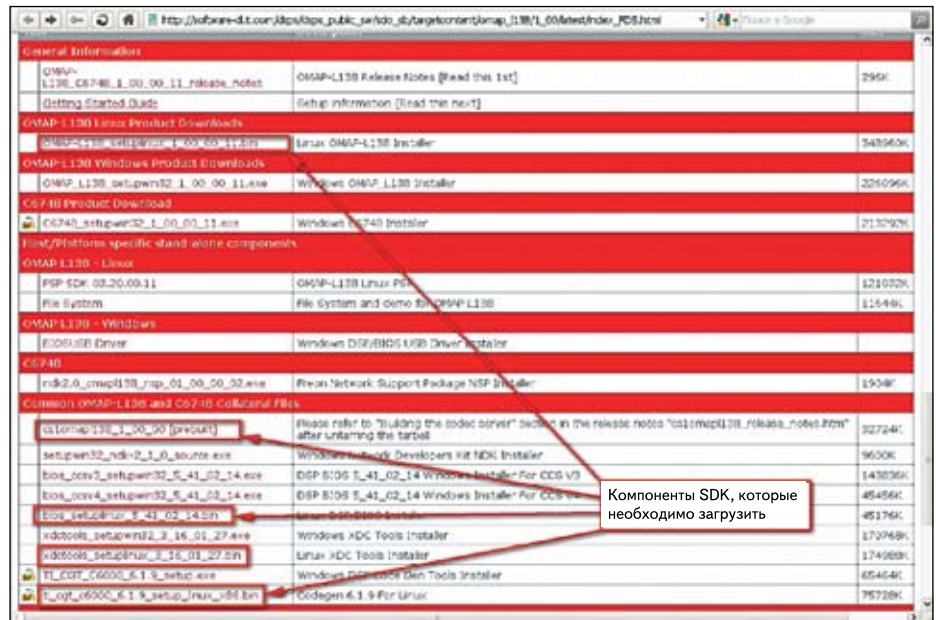


Рис. 1. Страница загрузки компонентов SDK

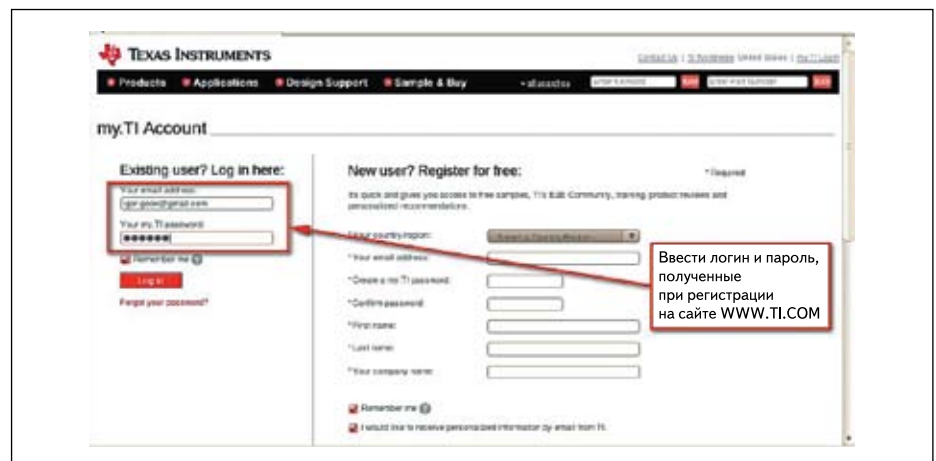


Рис. 2. Запрос логина и пароля при загрузке компонентов SDK

ные директории установки. Рекомендуется не использовать стандартные (по умолчанию) установки, а выбрать одно и то же место для всех компонентов. Это упростит дальнейшую работу с устанавливаемым программным обеспечением (ПО).

Для установки требуемого ПО необходимо выполнить следующие шаги:

1. Войти в операционную систему (ОС) Linux, используя учетную запись обычного пользователя. На рис. 3 показан пример авторизации в момент загрузки ОС Ubuntu 10.4 для пользователя с именем *ubuntu*. Затем загрузить указанные выше компоненты SDK в папку */home/ubuntu/Загрузки/* (это стандартная папка загрузки русифицированной ОС Ubuntu 10.4 для пользователя с именем *ubuntu*).



Рис. 3. Авторизация пользователя

2. Запустить терминальную программу (рис. 4).
3. Переключиться в режим *root* — команда *sudo -i* (может потребоваться введение пароля учетной записи пользователя, причем он не отображается при вводе) и перейти в каталог — команда *cd*, куда были загружены бинарные файлы (в нашем случае это папка */home/ubuntu/Загрузки/*). Затем сделать все файлы с расширением *bin* исполняемыми — команда *chmod +x \*.bin*. Выйти из режима *root* — команда *exit*. Все команды вводятся в окне терминала (рис. 5).



Рис. 5. Определение исполняемых файлов

4. Распаковать архив, запустив файл *OMAP-L138\_setuplinux\_#.#.#.#.bin*, используя команду *./OMAP-L138\_setuplinux\_#.#.#.#.bin*, как это показано на рис. 6, сноска 1. (Обратите внимание, что вначале осуществляется переход в папку, где располагаются загруженные файлы, — команда *cd*.)

После введения команды появится окно запуска инсталлятора SDK (рис. 6, сноска 2), необходимо подтвердить начало установки, нажав клавишу *Yes* (рис. 6, сноска 3). Процесс установки SDK показан на рис. 7.

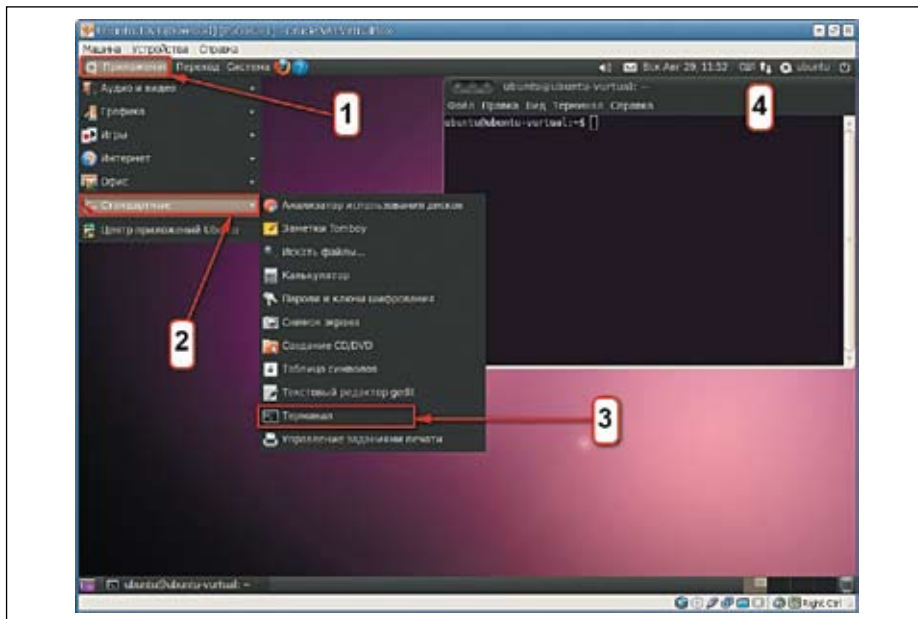


Рис. 4. Запуск терминальной программы

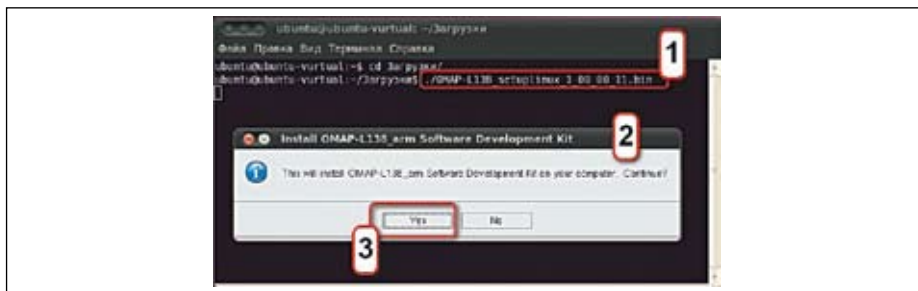


Рис. 6. Распаковка архива OMAPL138\_arm\_#.#.#.#\_#

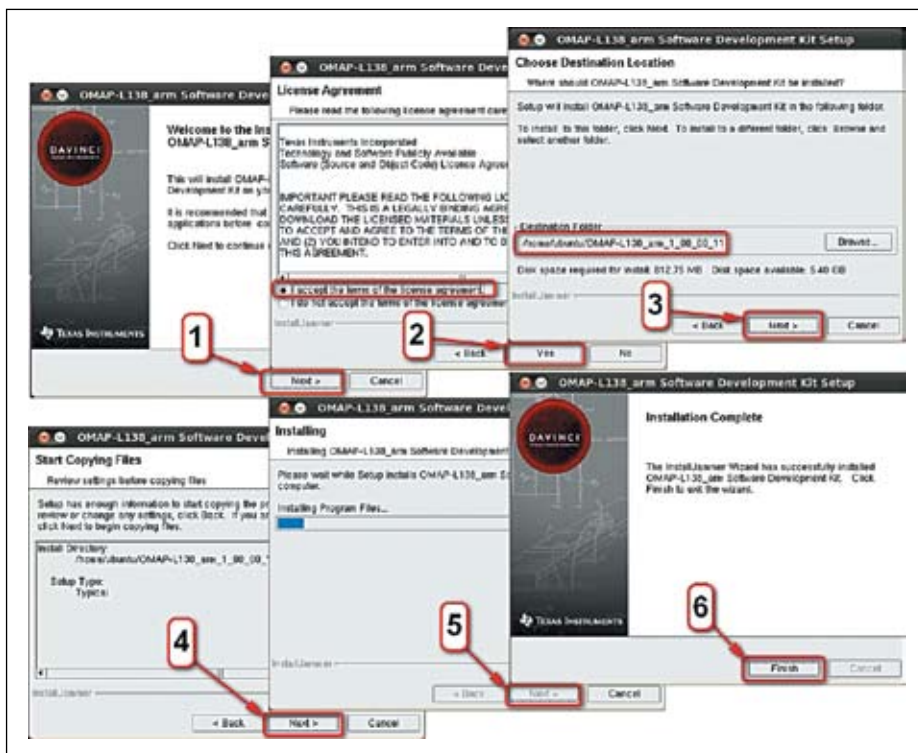


Рис. 7. Инсталляция SDK

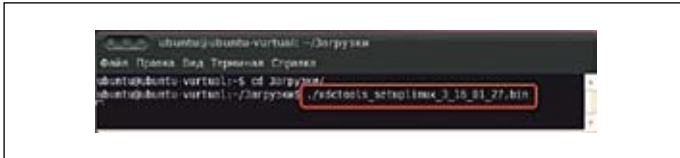


Рис. 8. Запуск файла установки XDCtools

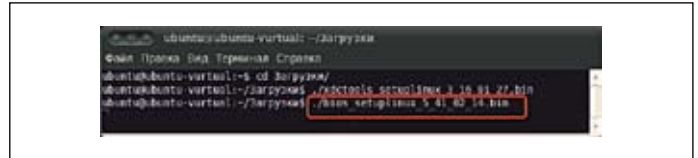


Рис. 11. Запуск файла установки DSP/BIOS

Для размещения распакованных файлов используется каталог `/home/<useract>/OMAPL138_arm_#_#_#_#` (в рассматриваемом примере — это `/home/ubuntu/OMAPL138_arm_1_00_00_11`). Подробную информацию о содержании каталога можно получить в [8].

5. Инсталлировать XDCtools, выполнив команду `./xdttools_setuplinux_#_#_#_#.bin`, как это показано на рис. 8.

На рис. 8 первой командой осуществляется переход в папку, где размещены все загруженные файлы, однако, если после установки пакета SDK терминал не закрывался, то команда перехода не нужна.

Установка XDCtools аналогична предыдущему пункту. Стоит отметить необходимость выбора английского языка в качестве языка установки, в противном случае может некорректно отображаться выводимая информация. Окно с выбором языка появляется первым при запуске инсталлятора (рис. 9).



Рис. 9. Выбор языка инсталляции

Второе замечание по процессу установки XDCtools — не использовать размещение по умолчанию. Вместо этого устанавливайте ПО в директорию, созданную на предыдущем шаге. Выбор директории установки показан на рис. 10.

6. Установить DSP/BIOS, для чего введите команду `./bios_setuptools_#_#_#_#.bin`, как это показано на рис. 11. Обратите внимание, что нет команды перехода в папку загрузки, так как окно терминала не закрывалось после предыдущего шага.

Несколько замечаний. Первое: выберите английский язык установки. Второе: здесь также не используйте размещение по умолчанию, а устанавливайте ПО в директорию `/home/<useract>/OMAPL138_`

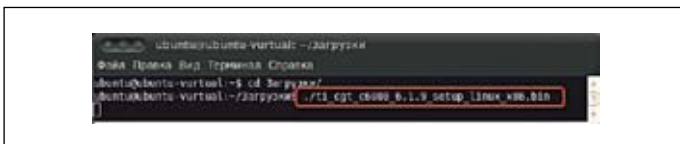


Рис. 12. Запуск файла установки Code Generation Tools

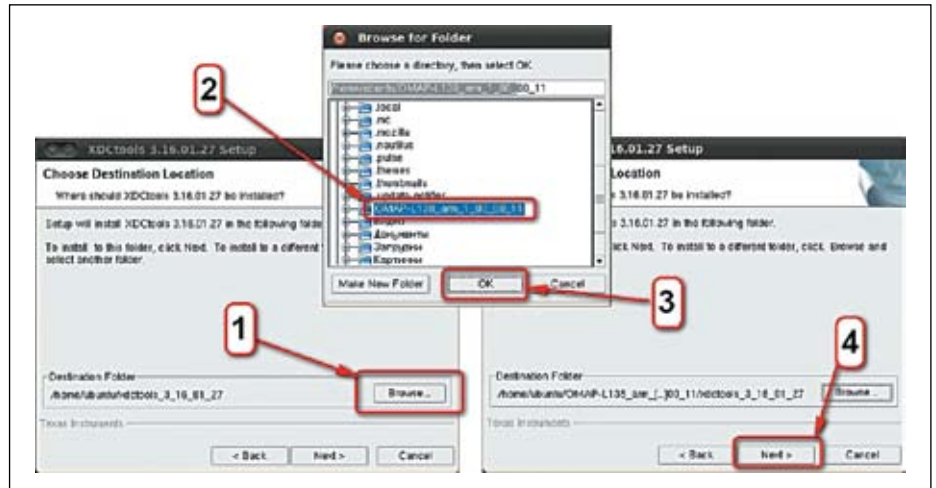


Рис. 10. Выбор папки установки XDCtools

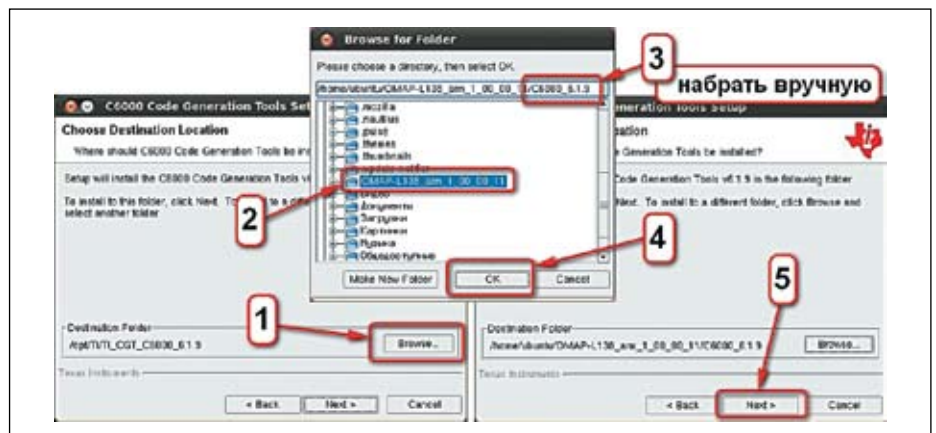


Рис. 13. Создание папки установки Code Generation Tools

`arm_#_#_#_#`, что была создана при установке SDK. В остальном процесс инсталляции аналогичен предыдущим.

7. Выполнить установку Code Generation Tools, введя в терминале команду `./t11_cgt_c6000_#_#_#_#_setup_linux_x86.bin`, как это показано на рис. 12.

В процессе инсталляции необходимо выбрать английский язык и вручную (в отличие от предыдущих шагов) создать папку `C6000_#_#_#_#` в каталоге `/home/<useract>/`

`OMAPL138_arm_#_#_#_#`, куда будет устанавливаться Code Generation Tools. Создание папки показано на рис. 13 (в рассматриваемом примере создается папка `C6000_6.1.9`).

Кроме того, требуется установить переменную среды в соответствии с параметрами установки:

```
export C6X_C_DIR="/home/<useract>/OMAPL138_arm_#_#_#_#/  
cg6x_#_#_#/include;  
/home/<useract>/OMAPL138_arm_#_#_#_#/cg6x_#_#_#/lib"
```



Рис. 14. Установка переменных среды





Рис. 19. Выбор установочного пакета кросс-компилятора CodeSourcery

При использовании 64-разрядной ОС Linux может потребоваться установка пакета *ia32-libs*, что выполняется командой `sudo apt-get install ia32-libs`.

3. Перейти в папку загрузки кросс-компилятора (команда `cd` — рис. 21, сноска 1), сделать установочный файл исполняемым (команда `chmod +x arm-2009q1-203-arm-none-linux-gnueabi.bin` — рис. 21, сноска 2) и запустить инсталляцию (команда `./arm-2009q1-203-arm-none-linux-gnueabi.bin` — рис. 21, сноска 3).

Появится окно старта процесса инсталляции кросс-компилятора CodeSourcery. Необходимо запустить программу установки, нажав кнопку *Next* (рис. 21, сноска 4), и следовать инструкциям. При написании статьи предложенные инсталлятором настройки не менялись.

Если путь, заданный по умолчанию, не изменялся, тогда файлы установленного кросс-компилятора можно найти в директории `~/home/<user>/CodeSourcery`. При этом необходимо указать путь для Code Sourcery tools в переменных окружения, что выполняется командой `export PATH=~/home/<user>/CodeSourcery/Sourcery_G++_Lite/bin:$PATH`. Строку с указанием пути (без кавычек!) можно добавить в файл `~/bashrc`, как это было сделано ранее для переменной `C6X_C_DIR` (рис. 15, 16), и в дальнейшем устанавливать требуемые параметры при помощи команды `source ~/bashrc`.

На этом вторая часть статьи, посвященной отладочной плате HawkBoard, завершена. В третьей части будет сделан следующий шаг освоения отладочной платы: мы рассмотрим вопрос самостоя-

тельного компилирования основных программных компонентов: загрузчика, ядра и файловой системы.

## Литература

1. Гук И. Отладочная плата HawkBoard на базе процессора OMAP-L138. Часть 1 // Компоненты и технологии. 2010. № 7.
2. <http://www.innovatesolutions.net/products/hawkboard>
3. <http://www.virtualbox.org/>
4. <http://www.ubuntu.com/>
5. <http://ru.wikipedia.org/wiki/VirtualBox>
6. <http://www.ti.com>
7. [http://software-dl.ti.com/dsps/dsps\\_public\\_sw/sdo\\_sb/targetcontent/omap-l138/l\\_00/latest/index\\_FDS.html](http://software-dl.ti.com/dsps/dsps_public_sw/sdo_sb/targetcontent/omap-l138/l_00/latest/index_FDS.html)
8. [http://processors.wiki.ti.com/index.php/GSG:\\_Installing\\_the\\_Software\\_for\\_OMAP-L138#About\\_the\\_PSP\\_package](http://processors.wiki.ti.com/index.php/GSG:_Installing_the_Software_for_OMAP-L138#About_the_PSP_package)
9. <http://www.codesourcery.com/>
10. <http://www.codesourcery.com/sgpp/lite/arm/portal/release858>

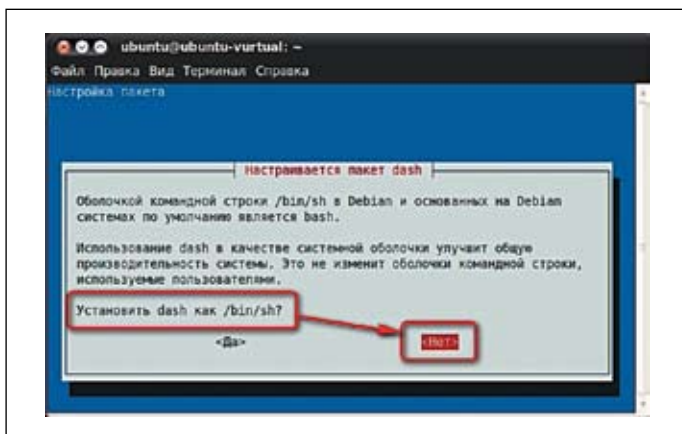


Рис. 20. Выбор bash в качестве основной оболочки



Рис. 21. Запуск инсталляции кросс-компилятора CodeSourcery