



## SAMC-403. Загрузчик IBL

Руководство пользователя

Версия 1.2



Код документа: UG-SAMC-403-IBL  
Дата сборки: 27 мая 2015 г.  
Листов в документе: 21

© 2015, ООО «Скан Инжиниринг Телеком - СПб»  
<http://www.setdsp.ru>

## История ревизий

Ревизия	Дата	Изменения
1.2	29.10.2014	Исправлено имя файла конфигурации в разделе 3. Внесены исправления, связанные с изменениями в системе сборки загрузчика. Заменен рисунок 4-4.
1.1	12.06.2014	Внесены правки в таблицу 4-1
1.0	13.12.2012	Начальная версия

## Содержание

Список рисунков .....	4
Список таблиц .....	4
Список листингов .....	4
Список процедур .....	4
Перечень сокращений и условных обозначений .....	5
<b>1 Загрузчик IBL .....</b>	<b>6</b>
<b>2 Сборка IBL .....</b>	<b>7</b>
2.1 Сборка в Windows системе .....	7
2.2 Сборка в Linux системе .....	11
<b>3 Запись образа IBL в EEPROM .....</b>	<b>13</b>
<b>4 Конфигурация IBL .....</b>	<b>15</b>
<b>Приложение А: Аппаратная конфигурация модуля SAMC-403 .....</b>	<b>21</b>

## Список рисунков

2-1	Установка MinGW. Выбор каталога репозитория	7
2-2	Установка MinGW. Окно лицензии	8
2-3	Установка MinGW. Выбор папки для установки	8
2-4	Установка MinGW. Выбор устанавливаемых компонентов	9
2-5	Приглашение командной строки MinGW Shell	9
2-6	Редактирование файла «setupenvMsys.sh» в редакторе vim	10
2-7	Редактирование файла «setupenvMsys.sh» в редакторе WordPad	10
2-8	Редактирование файла «setupenvMsys.sh» в редакторе nano	12
4-1	Запуск программы «i2cparam_0x51_samc403_le_0x500.out»	15
4-2	Пункт меню управления GEL файлами	15
4-3	Пункт меню загрузки GEL файла	16
4-4	Меню скриптов для модуля SAMC-403	16

## Список таблиц

4-1	Основные конфигурационные параметры в файле «i2cConfig.gel»	19
A-1	Положение переключателей для программирования модуля SAMC-403	21
A-2	Положение переключателей для загрузки модуля SAMC-403 с NAND-флеш памяти	21
A-3	Положение переключателей для загрузки модуля SAMC-403 с TFTP-сервера	21

## Список листингов

4-1	Функция setConfig_c6678_main() файла «i2cConfig.gel»	16
-----	--	----

## Список процедур

3-1	Обновление GEL файла конфигурации CCS	13
3-2	Запись загрузчика на модуль SAMC-403 в Windows системе	13

## Перечень сокращений и условных обозначений

<b>BOOTP</b>	Bootstrap Protocol	19, 20
<b>CCS</b>	Code Composer Studio	7, 13
<b>EEPROM</b>	Electrically Erasable Programmable Read-Only Memory	6, 7, 11, 13–16
<b>GEL</b>	General Extension Language	11, 13, 15, 16
<b>I<sup>2</sup>C</b>	Inter-Integrated Circuit	7, 14
<b>IBL</b>	Intermediate Boot Loader	6, 7, 9–11, 13–16, 19–21
<b>IP</b>	Internet Protocol	20
<b>JTAG</b>	Joint Test Action Group	21
<b>LTS</b>	Long Term Support	11
<b>MAC</b>	Media Access Control	20
<b>NAND</b>	Not AND	6, 21
<b>NOR</b>	Not OR	6
<b>TFTP</b>	Trivial File Transfer Protocol	6, 21
<b>USB</b>	Universal Serial Bus	13

## 1 Загрузчик IBL

Загрузчик IBL (Intermediate Boot Loader) позволяет выполнять загрузку приложений на модуль SAMC-403 с NAND-флеш памяти, NOR-флеш памяти или по Ethernet с TFTP-сервера. В данном разделе дается краткое описание возможностей IBL, описан алгоритм загрузки модуля SAMC-403 с загрузчиком IBL, описывается процесс сборки IBL из исходных кодов, конфигурация загрузчика для работы на модуле SAMC-403, даются инструкции по правильной подготовке и записи образа IBL в EEPROM.

## 2 Сборка IBL

Для сборки IBL потребуется установка компилятора для процессоров Texas Instruments серии C6000. Данный компилятор входит в состав системы разработки CCS (Code Composer Studio).

Дистрибутив CCS можно найти на сопроводительном диске к модулю SAMC-403 в папке «software/ccs». Исходные коды IBL можно найти на сопроводительном диске к модулю SAMC-403 в папке «/ibl/src».

Результатом сборки IBL из исходных кодов является готовый к записи в I<sup>2</sup>C EEPROM образ загрузчика.

### Внимание



Перед выполнением сборки загрузчика, описанной в данном разделе, перепишите с сопроводительного диска к модулю SAMC-403 папку «ibl» на жесткий диск компьютера. Далее, предполагается, что папка «ibl» переписана в папку «D:/ibl».

### 2.1 Сборка в Windows системе

Для успешной сборки IBL в Windows системе, кроме компилятора C6000 процессоров, необходима GNU система сборки MinGW. Скачать последнюю версию MinGW можно на официальном сайте <http://www.mingw.org>. При написании данного руководства была использована и проверена MinGW версии 20120426. Установочный дистрибутив MinGW версии 20120426 можно найти на сопроводительном диске к модулю SAMC-403 в папке «software/mingw».

### Внимание



Для установки MinGW, указанным в данном документе способом, требуется подключение к сети интернет.

При установке MinGW в окне выбора каталога репозитория (рисунок 2-1) рекомендуется выбрать пункт «Use pre-packaged repository catalogues» (использовать каталог репозитория с заранее собранными пакетами).

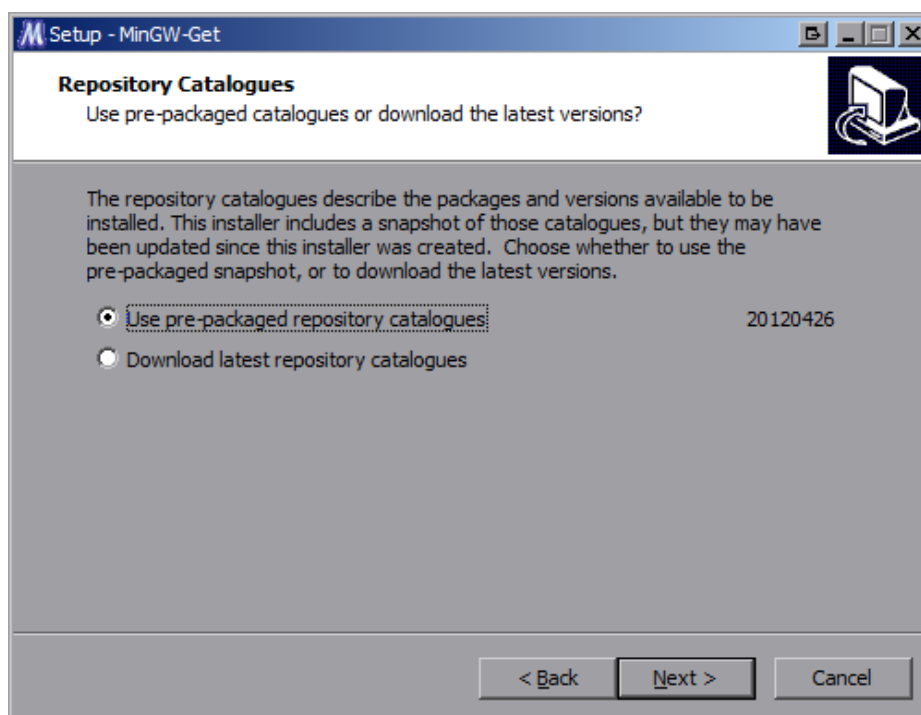


Рисунок 2-1: Установка MinGW. Выбор каталога репозитория

В окне лицензии (рисунок 2-2), прочитайте лицензию, и если вы согласны с ней, выберите пункт «I accept the agreement» и нажмите кнопку «Next».

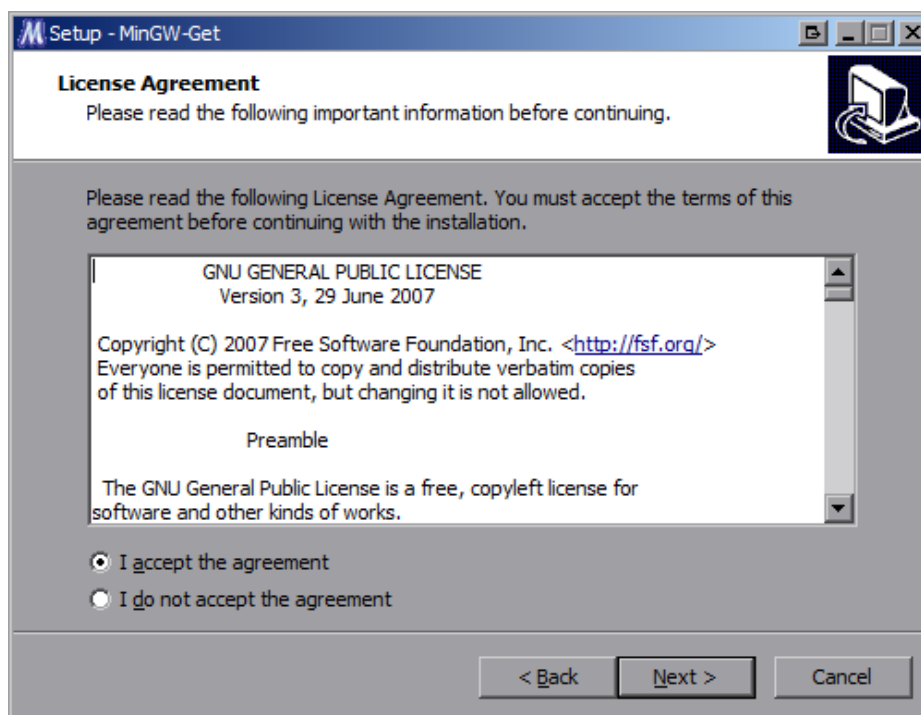


Рисунок 2-2: Установка MinGW. Окно лицензии

Путь установки MinGW (рисунок 2-3) рекомендуется оставить по умолчанию — «C:/MinGW».

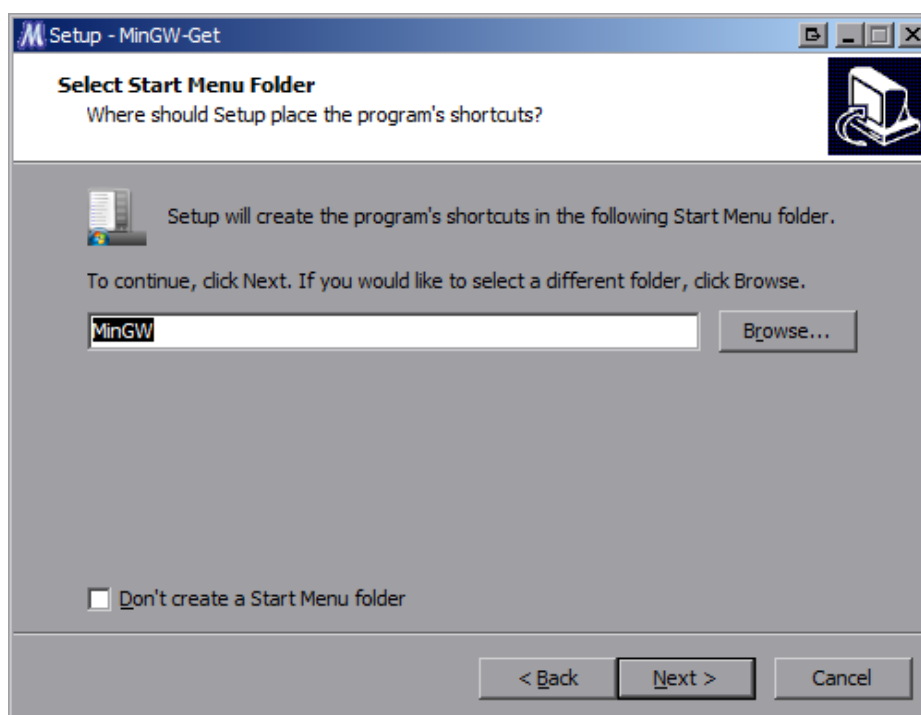


Рисунок 2-3: Установка MinGW. Выбор папки для установки

В окне выбора устанавливаемых компонентов (рисунок 2-4) необходимо отметить следующие компоненты:

- «C Compiler»;
- «MSYS Basic System»;
- «MinGW Developer ToolKit».



Остальные компоненты можно отметить на собственное усмотрение. На дальнейший процесс сборки IBL они никак не повлияют.

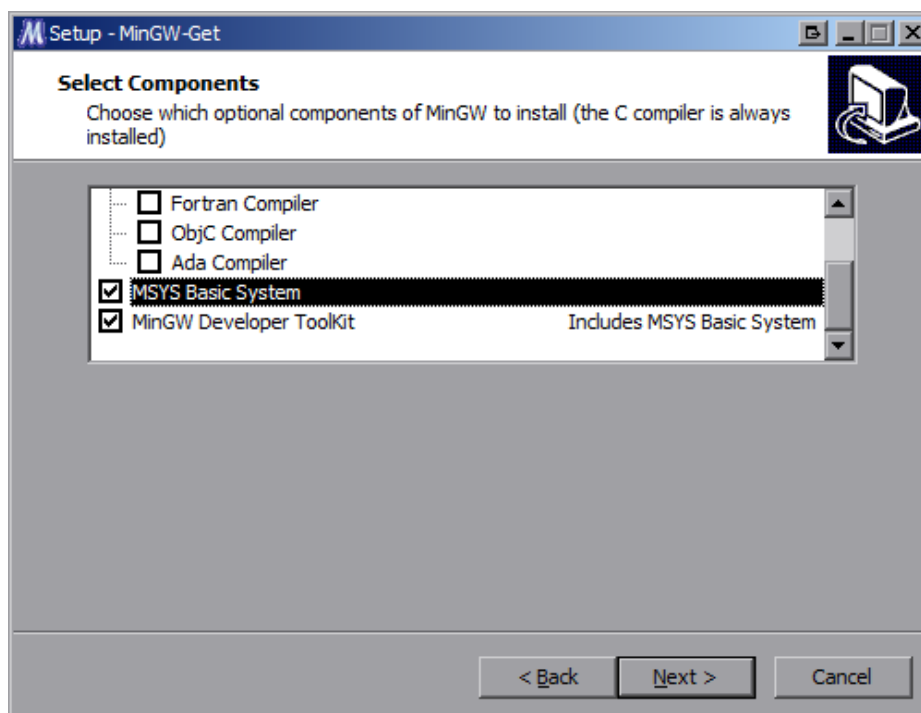


Рисунок 2-4: Установка MinGW. Выбор устанавливаемых компонентов

Остальные параметры установки, которые не описаны в данном руководстве, можно оставить в виде, предлагаемом установщиком по умолчанию.

После установки MinGW, через меню «Пуск», запустите «MinGW Shell» (рисунок 2-5). Все последующие действия по сборке IBL будут производиться путем ввода команд в MinGW Shell.

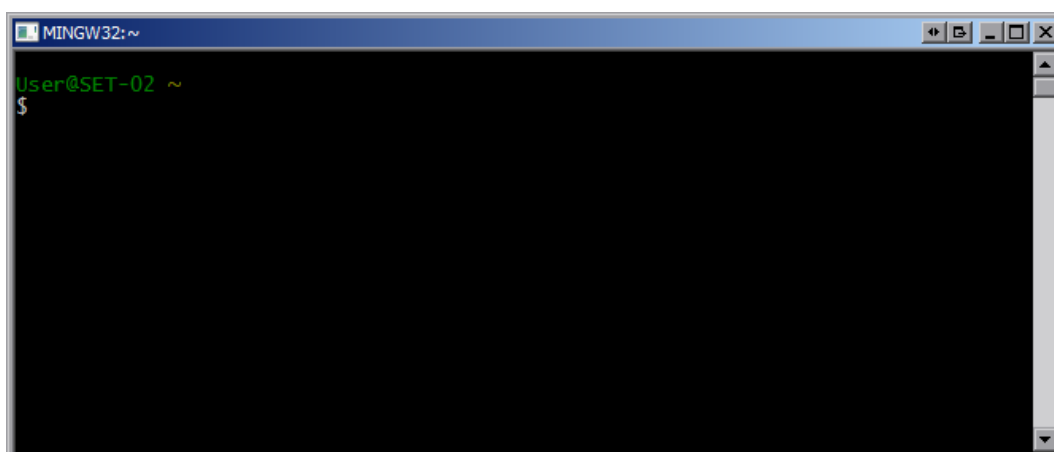
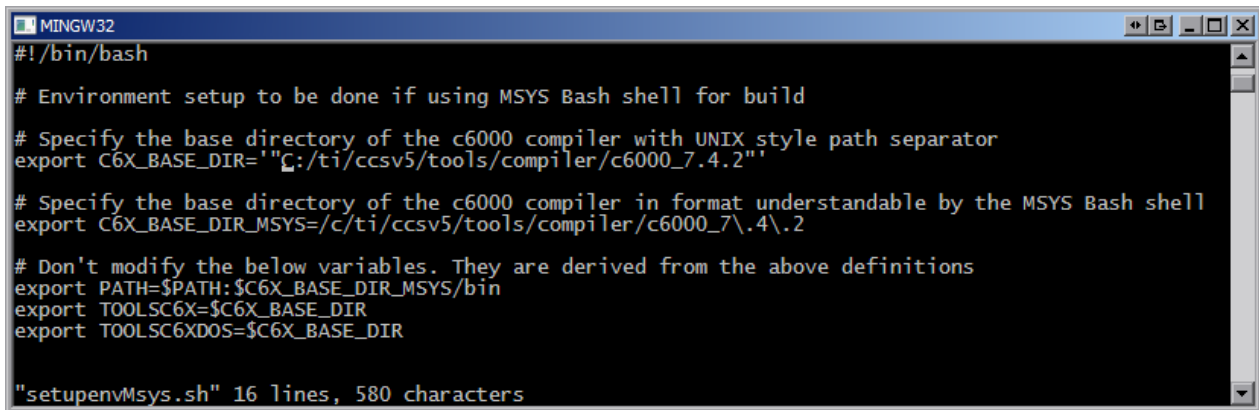


Рисунок 2-5: Приглашение командной строки MinGW Shell

Перед сборкой IBL перепишите исходные коды на жесткий диск компьютера с установленным MinGW и компилятором Texas Instruments для процессоров C6000 серии. Далее, предполагается, что исходные коды IBL помещены на компьютер в папку «D:/ibl», компилятор для процессоров C6000 серии находится в папке «C:/Program Files/Texas Instruments/ccsv5/tools/compiler/c6000\_7.3.4».

Перед сборкой IBL, необходимо выполнить изменение в скрипте конфигурации окружения сборки «D:/ibl/src/make/setupenvMsys.sh». В данном файле необходимо правильно указать путь с компилятору для процессоров C6000 серии. На рисунке 2-6 приведен снимок экрана MinGW Shell с открытым файлом «setupenvMsys.sh» в редакторе vim. На данном рисунке пути установлены в соответствии с путями указанными выше.

Для редактирования файла в редакторе vim выполните в MinGW Shell команду:



```

MINGW32
#!/bin/bash

# Environment setup to be done if using MSYS Bash shell for build

# Specify the base directory of the c6000 compiler with UNIX style path separator
export C6X_BASE_DIR="C:/ti/ccsv5/tools/compiler/c6000_7.4.2"

# Specify the base directory of the c6000 compiler in format understandable by the MSYS Bash shell
export C6X_BASE_DIR_MSYS=/c/ti/ccsv5/tools/compiler/c6000_7\4\2

# Don't modify the below variables. They are derived from the above definitions
export PATH=$PATH:$C6X_BASE_DIR_MSYS/bin
export TOOLSC6X=$C6X_BASE_DIR
export TOOLSC6XDOS=$C6X_BASE_DIR

"setupenvMsys.sh" 16 lines, 580 characters

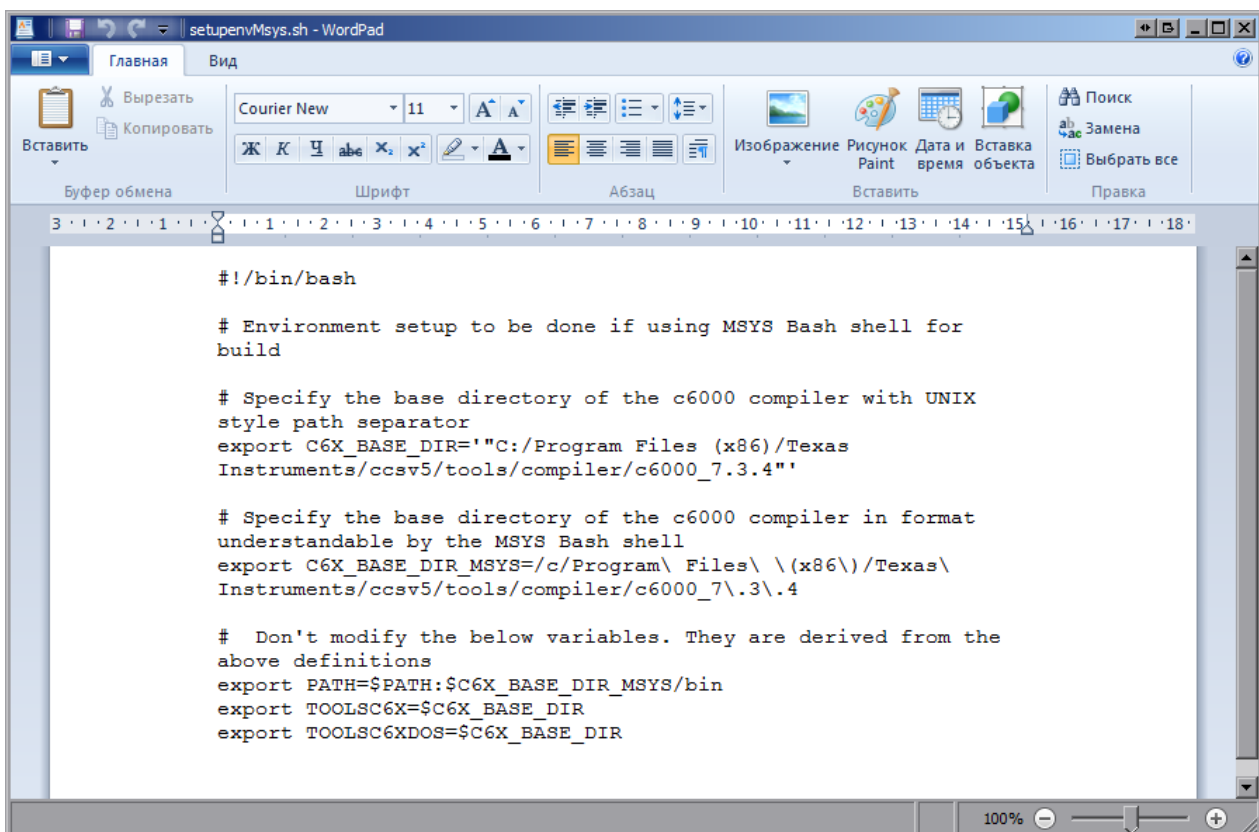
```

Рисунок 2-6: Редактирование файла «setupenvMsys.sh» в редакторе vim

```
vim /d/ibl/src/make/setupenvMsys.sh
```

Описание использования редактора vim для редактирования файлов выходит за рамки данного руководства. Использование редактора vim у не подготовленного пользователя может вызвать определенные проблемы. Поэтому, если пользователь не знаком с редактором vim, рекомендуется воспользоваться стандартным редактором WordPad для редактирования файла «setupenvMsys.sh». Редактор Notepad для редактирования данного файла использовать не рекомендуется.

Откройте файл «D:/ibl/src/make/setupenvMsys.sh» в редакторе WordPad и отредактируйте значения переменных C6X\_BASE\_DIR и C6X\_BASE\_DIR\_MSYS, указав в них путь к компилятору Texas Instruments для процессоров C6000 серий (см. рисунок 2-7).



```

setupenvMsys.sh - WordPad
Главная Вид
Вырезать Вставить
Буфер обмена
Шрифт: Courier New, 11
Абзац
Вставить: Изображение, Рисунок Paint, Дата и время, Вставка объекта
Правка: Поиск, Замена, Выбрать все
100%
#!/bin/bash

# Environment setup to be done if using MSYS Bash shell for
build

# Specify the base directory of the c6000 compiler with UNIX
style path separator
export C6X_BASE_DIR="C:/Program Files (x86)/Texas
Instruments/ccsv5/tools/compiler/c6000_7.3.4"

# Specify the base directory of the c6000 compiler in format
understandable by the MSYS Bash shell
export C6X_BASE_DIR_MSYS=/c/Program\ Files\ \ (x86\)/Texas\
Instruments/ccsv5/tools/compiler/c6000_7\3\4

# Don't modify the below variables. They are derived from the
above definitions
export PATH=$PATH:$C6X_BASE_DIR_MSYS/bin
export TOOLSC6X=$C6X_BASE_DIR
export TOOLSC6XDOS=$C6X_BASE_DIR

```

Рисунок 2-7: Редактирование файла «setupenvMsys.sh» в редакторе WordPad

Следует отметить, что в значении переменной C6X\_BASE\_DIR\_MSYS такие символы как пробел, точка, открывающая и закрывающая круглые скобки должны обязательно предваряться (экранироваться) символом обратной косой черты «\».

Для запуска процесса сборки IBL, выполните в MinGW Shell последовательно следующие команды:

```
cd /d/ibl/src/make/  
source setupenvMsys.sh  
make samc403
```

После выполнения этих команд, будет запущена сборка IBL.

В случае успешного завершения сборки, в папку «D:/ibl/src/make/bin» будут записаны четыре файла:

- «i2cConfig.gel» — GEL скрипт конфигурации IBL для Code Composer Studio (см. раздел 4);
- «i2cparam\_0x51\_samc403\_le\_0x500.out» — программа конфигурации IBL для Code Composer Studio (см. раздел 4);
- «i2crom\_0x51\_samc403\_le.bin» — бинарный образ IBL для загрузки в EEPROM (см. раздел 3);
- «i2crom\_0x51\_samc403\_le.dat» — образ IBL для загрузки в EEPROM в формате Code Composer Studio (см. раздел 3).

## 2.2 Сборка в Linux системе

Сборка IBL на Linux системе мало чем отличается от сборки для Windows систем, которая описана в разделе 2.1.

В данном разделе, в качестве Linux дистрибутива, предполагается использование Linux дистрибутива Ubuntu 10.04.4 LTS. При использовании других Linux дистрибутивов, некоторые моменты могут незначительно отличаться от описанного здесь.

Перед сборкой IBL необходимо переписать исходные коды на жесткий диск компьютера с установленным компилятором для процессоров C6000 серии, на котором будет выполняться сборка. Далее, предполагается, что исходные коды IBL находятся на компьютере в папке «~/ibl», компилятор Texas Instruments для процессоров C6000 серии в папке «/opt/TI\_CGT\_C6000\_7.2.1».

Перед сборкой IBL, необходимо выполнить изменение в скрипте конфигурации окружения сборки «~/ibl/src/make/setupenvLnx.sh». В данном файле необходимо правильно указать путь к компилятору для процессоров C6000 серии. На рисунке 2-8 приведен снимок экрана с открытым файлом «setupenvLnx.sh» в редакторе nano. На данном рисунке пути установлены в соответствии с путями указанными выше.

Для редактирования файла в редакторе nano выполните в терминале команду:

```
nano ~/ibl/src/make/setupenvLnx.sh
```

Отредактируйте значения переменных C6X\_CGT\_VERSION и C6X\_BASE\_DIR, указав в них версию и путь к компилятору Texas Instruments для процессоров C6000 серий.

Для запуска процесса сборки IBL, выполните в терминале следующие команды:

```
cd ~/ibl/src/make/  
source setupenvLnx.sh  
make samc403
```

После выполнения этих команд, будет запущена сборка IBL. В случае успешного завершения сборки, в папку «~/ibl/src/make/bin» будут записаны четыре файла:

- «i2cConfig.gel»;
- «i2cparam\_0x51\_samc403\_le\_0x500.out»;
- «i2crom\_0x51\_samc403\_le.bin»;
- «i2crom\_0x51\_samc403\_le.dat».

Краткое описание назначения каждого из файлов дано в конце раздела 2.1, описывающего процесс сборки IBL в Windows системе.

```
GNU nano 2.2.2 File: setupenvLnx.sh Modified
#!/bin/bash
export OS="Linux"
C6X_CGT_VERSION=7.2.1
C6X_BASE_DIR=/opt/TI_CGT_C6000_${C6X_CGT_VERSION}

if [ -z $C6X_CGT_VERSION ]; then
    C6X_CGT_VERSION=7.2.4
fi

if [ -z "$C6X_BASE_DIR" ]; then
    for dir in {~,}{,/opt}/{ti,TI,texas_instruments}/TI_CGT_C6000_${C6X_CGT_VERSION} {~,}{,/opt}/$
        if [ -x $dir/bin/cl6x ]; then
            C6X_BASE_DIR=$dir
            break
        fi
    done
fi

if [ ! -x $C6X_BASE_DIR/bin/cl6x ]; then
    echo "You must define the C6X_BASE_DIR to point to TI CGT compiler for C6000"
    exit 2
fi

#make sure its exported
export C6X_BASE_DIR

export PATH=$C6X_BASE_DIR/bin:$PATH
export TOOLSC6X=$C6X_BASE_DIR
export TOOLSC6XDOS=$C6X_BASE_DIR
export TOOLSBIOSC6XDOS=$C6X_BASE_DIR

^G Get Help      ^O WriteOut      ^R Read File     ^Y Prev Page     ^K Cut Text      ^C Cur Pos
^X Exit          ^J Justify       ^W Where Is     ^V Next Page     ^U UnCut Text   ^T To Spell
```

Рисунок 2-8: Редактирование файла «setupenvMsys.sh» в редакторе nano

### 3 Запись образа IBL в EEPROM

Запись образа IBL в EEPROM производится с помощью специальной программы, которая имеется на сопроводительном диске к модулю SAMC-403. Для записи образа в EEPROM потребуется установленная система разработки Code Composer Studio. Рекомендуется использовать Code Composer Studio версии 5.2.0.00069, дистрибутив которой также имеется на диске как для Windows, так и для Linux системы.

Перед записью образа IBL в EEPROM, необходимо его собрать, руководствуясь инструкциями данными в разделе 2, или использовать заранее собранный образ «i2crom\_0x51\_samc403\_le.bin», который имеется на сопроводительном диске к модулю SAMC-403 в папке «ibl/bin».

Перед выполнением записи, необходимо обновить GEL файл конфигурации «evmc6678l.gel» модуля SAMC-403 для CCS, который имеется на сопроводительном диске к модулю в папке «ibl/program\_evm/gel». Для обновления GEL файла конфигурации, выполните следующие действия, описанные в процедуре 3-1.

Процедура 3-1. Обновление GEL файла конфигурации CCS

1. В последующих шагах предполагается, что CCS установлена в папку «<CCS\_INSTALL\_DIR>» (обычно это «C:/Program Files/Texas Instruments/ccsv5»).
2. Закройте Code Composer Studio, если она открыта.
3. Замените существующий файл «evmc6678l.gel» в папке «<CCS\_INSTALL\_DIR>/ccs\_base/emulation/boards/evmc6678l/gel» на файл «evmc6678l.gel» с сопроводительного диска к модулю SAMC-403.

#### Внимание



Перед выполнением действий, описанных в данном разделе, перепишите с сопроводительного диска к модулю SAMC-403 папку «ibl» на жесткий диск компьютера, если этого не было сделано ранее. Далее, предполагается, что папка переписана в папку «D:/ibl».

Шаги, необходимые для выполнения записи загрузчика IBL на модуль SAMC-403 в Windows системе, описаны в процедуре 3-2.

Процедура 3-2. Запись загрузчика на модуль SAMC-403 в Windows системе

1. На выключенном модуле SAMC-403 установите переключатели в соответствии с таблицей A-1 (см. приложение A).
2. Установите правильное значение переменной окружения DSS\_SCRIPT\_DIR. Если CCS установлена в папку «C:/Program Files (x86)/Texas Instruments/ccsv5», то переменной DSS\_SCRIPT\_DIR должно быть присвоено значение «C:/Program Files (x86)/Texas Instruments/ccsv5/ccs\_base/scripting/bin». Сделать это можно выполнив команду

```
set DSS_SCRIPT_DIR="C:\Program Files (x86)\Texas Instruments\ccsv5\ccs_base\scripting\bin"
```

3. Включите модуль SAMC-403 и дождитесь инициализации эмулятора и завершения USB инициализации (около 10 секунд для XDS100 эмулятора и около 45 секунд для XDS560 эмулятора).
4. Запустите программу program\_evm для записи образа загрузчика IBL в EEPROM:

```
cd <путь к программе program_evm>  
%DSS_SCRIPT_DIR%\dss.bat program_evm.js TMDSEVMC6678L-le "eeprom51"
```

В случае успешной записи загрузчика в EEPROM, в терминал будут выведены следующие сообщения:

```
I2C Bus address is = 81  
I2C writing has started  
Please be Patient
```

```
I2C write complete, reading data
I2C read complete, comparing data
Data compare passed
```

#### Примечание

Процедура 3-2 записывает данные только в EEPROM на I<sup>2</sup>C шине с адресом 0x51. Поэтому, при выполнении процедуры 3-2, все предупреждения о невозможности записи других типов памяти следует игнорировать.

При запуске программы `program_evm` для записи содержимого EEPROM (на это указывает параметр командной строки «`eeeprom51`»), будет записываться файл «`D:/ibl/program_evm/binaries/evm6678l/i2crom_0x51_samc403_le.bin`». Имя данного файла считывается программой `program_evm` из файла «`D:/ibl/program_evm/binaries/evm6678l/eeepromwriter_input51.txt`». По умолчанию, содержимое этого файла имеет следующий вид:

```
file_name = i2crom_0x51_samc403_le.bin
bus_addr = 81
start_addr = 0
swap_data = 0
```

В параметре «`file_name`» задается имя файла образа для записи в I<sup>2</sup>C EEPROM (в данном случае — «`i2crom_0x51_samc403_le.bin`»). Файл образа для записи в EEPROM должен находиться в той же папке, что и файлы «`eeepromwriter_evm6678l.out`» и «`eeepromwriter_input.txt`».

В параметре «`bus_addr`» задается адрес I<sup>2</sup>C шины целевой EEPROM. Для модуля SAMC-403 значение этого параметра должно быть равно 81 (0x51).

В параметре «`start_addr`» задается смещение относительно начала EEPROM, с которого начинать запись. Значение этого параметра зависит от записываемого образа, в общем случае должно быть равно 0.

Параметр «`swap_data`» позволяет включить преобразования порядка записи байтов (endianness) записываемых данных из big-endian в little-endian. Процессоры модуля SAMC-403 работают в режиме little-endian, поэтому использовать этот параметр следует лишь в том случае, если будет записываться образ, подготовленный в формате big-endian. Параметр может принимать значение 0 (преобразование выключено) или 1 (преобразование включено).

Файл «`i2crom_0x51_samc403_le.bin`» является бинарным образом загрузчика IBL, который был получен в результате сборки загрузчика (см. раздел 2).

#### Внимание



После выполнения сборки загрузчика, для записи образа, необходимо переписать собранный образ загрузчика «`i2crom_0x51_samc403_le.bin`» из папки «`D:/ibl/src/make/bin`» в папку «`D:/ibl/program_evm/binaries/evm6678l`».

## 4 Конфигурация IBL

Конфигурация IBL выполняется путем записи блока с конфигурационными параметрами в EEPROM с определенным смещением. Для облегчения этого процесса, используется специальная программа «i2cparam\_0x51\_samc403\_le\_0x500.out», которая собирается вместе с IBL (см. раздел 2). Эта программа предназначена для загрузки в память процессора модуля SAMC-403, аналогично программе eepromwriter, которая описана в разделе 3.

После загрузки в память процессора программы «i2cparam\_0x51\_samc403\_le\_0x500.out», программу необходимо запустить, нажав на клавиатуре клавишу F8 или выбрав пункт меню «Run > Resume». При запуске программы, программа сообщит о необходимости загрузки конфигурационного GEL файла (рисунок 4-1).

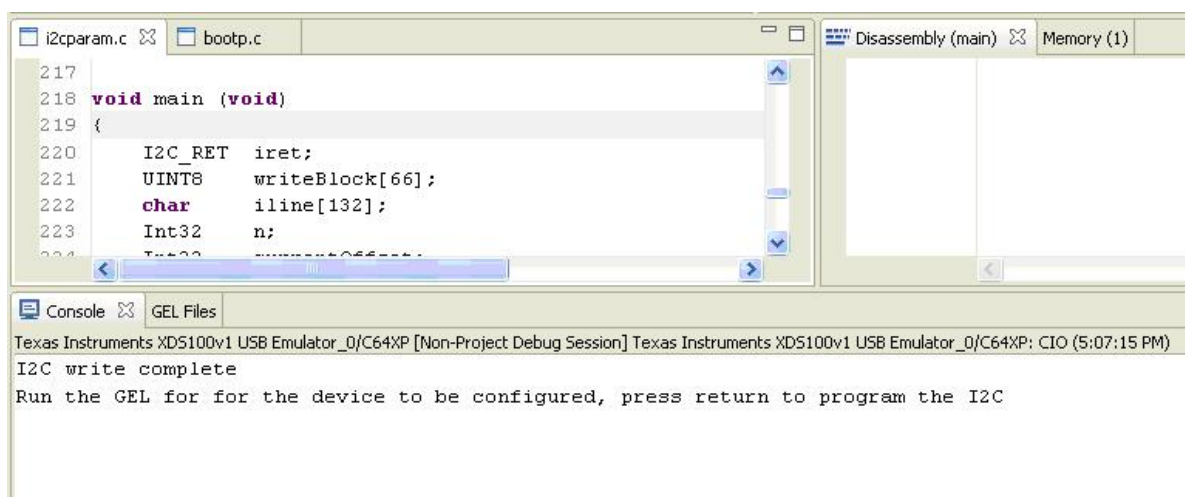


Рисунок 4-1: Запуск программы «i2cparam\_0x51\_samc403\_le\_0x500.out»

Для загрузки GEL файла, нужно выбрать пункт главного меню «Tools > GEL Files» (рисунок 4-2), в результате чего откроется окно (рисунок 4-3), в котором необходимо загрузить GEL файл, путем нажатия правой кнопки мыши и выбора пункта меню «Load GEL...».



Рисунок 4-2: Пункт меню управления GEL файлами

В открывшемся окне выбора файла, необходимо выбрать файл «i2cConfig.gel», который расположен в той же папке, куда записывается образ IBL после сборки (см. раздел 2). Также, файл «i2cConfig.gel» можно найти на сопроводительном диске к модулю SAMC-403 в папке «ibl/bin». Следует отметить, что данное действие необходимо выполнять только после завершения изменения конфигурации в файле «i2cConfig.gel», то есть файл должен быть сохранен.





Рисунок 4-3: Пункт меню загрузки GEL файла

Для применения конфигурации, описанной в файле «i2cConfig.gel», необходимо выбрать в главном меню пункт «Scripts > EVM c6678 IBL > setConfig\_c6678\_main», как показано на рисунке 4-4.

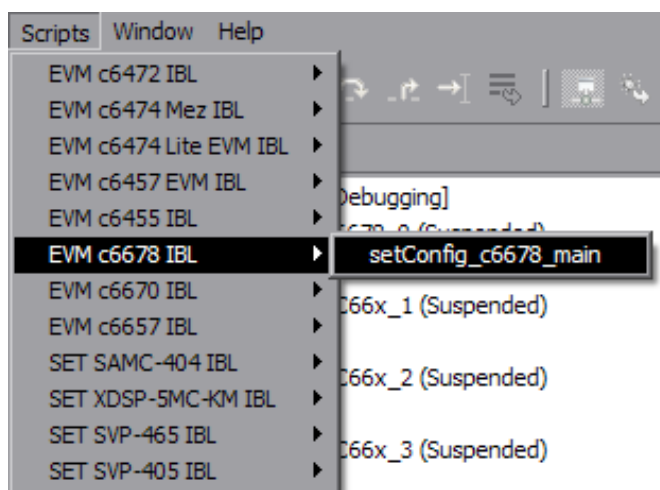


Рисунок 4-4: Меню скриптов для модуля SAMC-403

Для записи конфигурации в EEPROM, необходимо переключиться в окно «Console» и нажать клавишу Enter на клавиатуре (см. рисунок 4-1).

В конфигурационном файле «i2cConfig.gel» содержатся конфигурационные параметры IBL, которые можно задавать по своему усмотрению путем редактирования этого файла. При сборке образа (раздел 2), IBL сконфигурирован с параметрами, которые указаны в этом файле по умолчанию. Данный файл представляет из себя код на специальном скриптовом языке GEL системы разработки Code Composer Studio. Конфигурационные параметры, относящиеся к модулю SAMC-403, расположены в этом файле в функции setConfig\_c6678\_main().

Ниже приведен оригинальный листинг функции setConfig\_c6678\_main() из файла «i2cConfig.gel» с параметрами по умолчанию:

Листинг 4-1: Функция setConfig\_c6678\_main() файла «i2cConfig.gel»

```

1 hotmenu setConfig_c6678_main()
2 {
3     ibl.iblMagic    = ibl_MAGIC_VALUE;
4     ibl.iblEvmType = ibl_EVM_C6678L;
5
6     /* Main PLL: 100 MHz reference, 1GHz output */
7     ibl.pllConfig[ibl_MAIN_PLL].doEnable    = 1;
8     ibl.pllConfig[ibl_MAIN_PLL].prediv     = 1;
9     ibl.pllConfig[ibl_MAIN_PLL].mult      = 20;
10    ibl.pllConfig[ibl_MAIN_PLL].postdiv    = 2;
11    ibl.pllConfig[ibl_MAIN_PLL].pllOutFreqMhz = 1000;
12
13    /* DDR PLL: 66.66 MHz reference, 400 MHz output, for an 800MHz DDR rate */
14    ibl.pllConfig[ibl_DDR_PLL].doEnable     = 1;
15    ibl.pllConfig[ibl_DDR_PLL].prediv      = 1;
16    ibl.pllConfig[ibl_DDR_PLL].mult       = 20;
17    ibl.pllConfig[ibl_DDR_PLL].postdiv    = 2;
18    ibl.pllConfig[ibl_DDR_PLL].pllOutFreqMhz = 1333;
19
20    /* Net PLL: 100 MHz reference, 1050 MHz output */
21    /* (followed by a built in divide by 3 to give 350 MHz to PA) */

```



```
22  ibl.pllConfig[ibl_NET_PLL].doEnable      = 1;
23  ibl.pllConfig[ibl_NET_PLL].prediv       = 1;
24  ibl.pllConfig[ibl_NET_PLL].mult        = 21;
25  ibl.pllConfig[ibl_NET_PLL].postdiv      = 2;
26  ibl.pllConfig[ibl_NET_PLL].pllOutFreqMhz = 1050;
27
28  ibl.ldrConfig.configDdr = 1;
29  ibl.ldrConfig.uEmif.emif4p0.registerMask = ibl_EMIF4_ENABLE_sdRamConfig |
    < ibl_EMIF4_ENABLE_sdRamRefreshCtl | ibl_EMIF4_ENABLE_sdRamTiming1 |
    < ibl_EMIF4_ENABLE_sdRamTiming2 | ibl_EMIF4_ENABLE_sdRamTiming3 | ibl_EMIF4_ENABLE_ddsPhyCtl1;
30
31  ibl.ldrConfig.uEmif.emif4p0.sdRamConfig      = 0x63C452B2;
32  ibl.ldrConfig.uEmif.emif4p0.sdRamConfig2    = 0;
33  ibl.ldrConfig.uEmif.emif4p0.sdRamRefreshCtl  = 0x000030D4;
34  ibl.ldrConfig.uEmif.emif4p0.sdRamTiming1    = 0x0AAAE51B;
35  ibl.ldrConfig.uEmif.emif4p0.sdRamTiming2    = 0x2A2F7FDA;
36  ibl.ldrConfig.uEmif.emif4p0.sdRamTiming3    = 0x057F82B8;
37  ibl.ldrConfig.uEmif.emif4p0.lpDdrNvmTiming  = 0;
38  ibl.ldrConfig.uEmif.emif4p0.powerManageCtl  = 0;
39  ibl.ldrConfig.uEmif.emif4p0.iODFTTestLogic  = 0;
40  ibl.ldrConfig.uEmif.emif4p0.performCountCfg = 0;
41  ibl.ldrConfig.uEmif.emif4p0.performCountMstRegSel = 0;
42  ibl.ldrConfig.uEmif.emif4p0.readIdleCtl     = 0;
43  ibl.ldrConfig.uEmif.emif4p0.sysVbusmIntEnSet = 0;
44  ibl.ldrConfig.uEmif.emif4p0.sdRamOutImpdedCalCfg = 0;
45  ibl.ldrConfig.uEmif.emif4p0.tempAlterCfg    = 0;
46  ibl.ldrConfig.uEmif.emif4p0.ddsPhyCtl1     = 0x0010010d;
47  ibl.ldrConfig.uEmif.emif4p0.ddsPhyCtl2     = 0;
48  ibl.ldrConfig.uEmif.emif4p0.priClassSvceMap = 0;
49  ibl.ldrConfig.uEmif.emif4p0.mstId2ClsSvce1Map = 0;
50  ibl.ldrConfig.uEmif.emif4p0.mstId2ClsSvce2Map = 0;
51  ibl.ldrConfig.uEmif.emif4p0.eccCtl         = 0;
52  ibl.ldrConfig.uEmif.emif4p0.eccRange1      = 0;
53  ibl.ldrConfig.uEmif.emif4p0.eccRange2      = 0;
54  ibl.ldrConfig.uEmif.emif4p0.rdWrtExcThresh = 0;
55
56  ibl.sgmiiConfig[0].configure      = 1;
57  ibl.sgmiiConfig[0].adviseAbility = 1;
58  ibl.sgmiiConfig[0].control       = 1;
59  ibl.sgmiiConfig[0].txConfig      = 0x108a1;
60  ibl.sgmiiConfig[0].rxConfig      = 0x700621;
61  ibl.sgmiiConfig[0].auxConfig     = 0x41;
62
63  ibl.sgmiiConfig[1].configure      = 1;
64  ibl.sgmiiConfig[1].adviseAbility = 1;
65  ibl.sgmiiConfig[1].control       = 1;
66  ibl.sgmiiConfig[1].txConfig      = 0x108a1;
67  ibl.sgmiiConfig[1].rxConfig      = 0x700621;
68  ibl.sgmiiConfig[1].auxConfig     = 0x41;
69
70  ibl.mdioConfig.nMdioOps = 0;
71
72  ibl.spiConfig.addrWidth = 24;
73  ibl.spiConfig.nPins     = 5;
74  ibl.spiConfig.mode      = 1;
75  ibl.spiConfig.csel      = 2;
76  ibl.spiConfig.c2tdelay  = 1;
77  ibl.spiConfig.busFreqMHz = 20;
78
79  ibl.emifConfig[0].csSpace = 2;
80  ibl.emifConfig[0].busWidth = 8;
81  ibl.emifConfig[0].waitEnable = 0;
82
83  ibl.emifConfig[1].csSpace = 0;
84  ibl.emifConfig[1].busWidth = 0;
85  ibl.emifConfig[1].waitEnable = 0;
86
87  ibl.bootModes[0].bootMode = ibl_BOOT_MODE_NOR;
88  ibl.bootModes[0].priority = ibl_HIGHEST_PRIORITY;
89  ibl.bootModes[0].port     = 0;
90
```

```

91  ibl.bootModes[0].u.norBoot.bootFormat      = ibl_BOOT_FORMAT_ELF;
92  ibl.bootModes[0].u.norBoot.bootAddress[0][0] = 0;
93  ibl.bootModes[0].u.norBoot.bootAddress[0][1] = 0xA00000;
94  ibl.bootModes[0].u.norBoot.bootAddress[1][0] = 0;
95  ibl.bootModes[0].u.norBoot.bootAddress[1][1] = 0xA00000;
96  ibl.bootModes[0].u.norBoot.interface       = ibl_PMEM_IF_SPI;
97  ibl.bootModes[0].u.norBoot.blob[0][0].startAddress = 0x80000000;
98  ibl.bootModes[0].u.norBoot.blob[0][0].sizeBytes  = 0xA00000;
99  ibl.bootModes[0].u.norBoot.blob[0][0].branchAddress = 0x80000000;
100 ibl.bootModes[0].u.norBoot.blob[0][1].startAddress = 0x80000000;
101 ibl.bootModes[0].u.norBoot.blob[0][1].sizeBytes  = 0xA00000;
102 ibl.bootModes[0].u.norBoot.blob[0][1].branchAddress = 0x80000000;
103 ibl.bootModes[0].u.norBoot.blob[1][0].startAddress = 0x80000000;
104 ibl.bootModes[0].u.norBoot.blob[1][0].sizeBytes  = 0xA00000;
105 ibl.bootModes[0].u.norBoot.blob[1][0].branchAddress = 0x80000000;
106 ibl.bootModes[0].u.norBoot.blob[1][1].startAddress = 0x80000000;
107 ibl.bootModes[0].u.norBoot.blob[1][1].sizeBytes  = 0xA00000;
108 ibl.bootModes[0].u.norBoot.blob[1][1].branchAddress = 0x80000000;
109
110 ibl.bootModes[1].bootMode = ibl_BOOT_MODE_NAND;
111 ibl.bootModes[1].priority = ibl_HIGHEST_PRIORITY;
112 ibl.bootModes[1].port     = 0;
113
114 ibl.bootModes[1].u.nandBoot.bootFormat      = ibl_BOOT_FORMAT_BBLOB;
115 ibl.bootModes[1].u.nandBoot.bootAddress[0][0] = 0x4000;
116 ibl.bootModes[1].u.nandBoot.bootAddress[0][1] = 0x2000000;
117 ibl.bootModes[1].u.nandBoot.bootAddress[1][0] = 0x4000;
118 ibl.bootModes[1].u.nandBoot.bootAddress[1][1] = 0x2000000;
119 ibl.bootModes[1].u.nandBoot.interface       = ibl_PMEM_IF_CHIPSEL_2;
120
121 ibl.bootModes[1].u.nandBoot.blob[0][0].startAddress = 0x80000000;
122 ibl.bootModes[1].u.nandBoot.blob[0][0].sizeBytes  = 0xFFC000;
123 ibl.bootModes[1].u.nandBoot.blob[0][0].branchAddress = 0x80000000;
124 ibl.bootModes[1].u.nandBoot.blob[0][1].startAddress = 0x80000000;
125 ibl.bootModes[1].u.nandBoot.blob[0][1].sizeBytes  = 0xFFC000;
126 ibl.bootModes[1].u.nandBoot.blob[0][1].branchAddress = 0x80000000;
127 ibl.bootModes[1].u.nandBoot.blob[1][0].startAddress = 0x80000000;
128 ibl.bootModes[1].u.nandBoot.blob[1][0].sizeBytes  = 0xFFC000;
129 ibl.bootModes[1].u.nandBoot.blob[1][0].branchAddress = 0x80000000;
130 ibl.bootModes[1].u.nandBoot.blob[1][1].startAddress = 0x80000000;
131 ibl.bootModes[1].u.nandBoot.blob[1][1].sizeBytes  = 0xFFC000;
132 ibl.bootModes[1].u.nandBoot.blob[1][1].branchAddress = 0x80000000;
133
134
135 ibl.bootModes[1].u.nandBoot.nandInfo.busWidthBits = 8;
136 ibl.bootModes[1].u.nandBoot.nandInfo.pageSizeBytes = 512;
137 ibl.bootModes[1].u.nandBoot.nandInfo.pageEccBytes = 16;
138 ibl.bootModes[1].u.nandBoot.nandInfo.pagesPerBlock = 32;
139 ibl.bootModes[1].u.nandBoot.nandInfo.totalBlocks = 4096;
140
141 ibl.bootModes[1].u.nandBoot.nandInfo.addressBytes = 4;
142 ibl.bootModes[1].u.nandBoot.nandInfo.lsbFirst    = TRUE;
143 ibl.bootModes[1].u.nandBoot.nandInfo.blockOffset = 14;
144 ibl.bootModes[1].u.nandBoot.nandInfo.pageOffset  = 9;
145 ibl.bootModes[1].u.nandBoot.nandInfo.columnOffset = 0;
146
147 ibl.bootModes[1].u.nandBoot.nandInfo.eccBytesIdx[0] = 0;
148 ibl.bootModes[1].u.nandBoot.nandInfo.eccBytesIdx[1] = 1;
149 ibl.bootModes[1].u.nandBoot.nandInfo.eccBytesIdx[2] = 2;
150 ibl.bootModes[1].u.nandBoot.nandInfo.eccBytesIdx[3] = 3;
151 ibl.bootModes[1].u.nandBoot.nandInfo.eccBytesIdx[4] = 4;
152 ibl.bootModes[1].u.nandBoot.nandInfo.eccBytesIdx[5] = 6;
153 ibl.bootModes[1].u.nandBoot.nandInfo.eccBytesIdx[6] = 7;
154 ibl.bootModes[1].u.nandBoot.nandInfo.eccBytesIdx[7] = 13;
155 ibl.bootModes[1].u.nandBoot.nandInfo.eccBytesIdx[8] = 14;
156 ibl.bootModes[1].u.nandBoot.nandInfo.eccBytesIdx[9] = 15;
157
158 ibl.bootModes[1].u.nandBoot.nandInfo.badBlkMarkIdx[0] = 5;
159 ibl.bootModes[1].u.nandBoot.nandInfo.badBlkMarkIdx[1] = 0xff;
160
161 ibl.bootModes[1].u.nandBoot.nandInfo.resetCommand = 0xff;

```

```

162     ibl.bootModes[1].u.nandBoot.nandInfo.readCommandPre = 0;
163     ibl.bootModes[1].u.nandBoot.nandInfo.readCommandPost = 0;
164     ibl.bootModes[1].u.nandBoot.nandInfo.postCommand    = FALSE;
165
166     ibl.bootModes[2].bootMode = ibl_BOOT_MODE_TFTP;
167     ibl.bootModes[2].priority = ibl_HIGHEST_PRIORITY+1;
168     ibl.bootModes[2].port     = ibl_PORT_SWITCH_ALL;
169
170     ibl.bootModes[2].u.ethBoot.doBootp                = TRUE;
171     ibl.bootModes[2].u.ethBoot.useBootpServerIp       = TRUE;
172     ibl.bootModes[2].u.ethBoot.useBootpFileName       = TRUE;
173     ibl.bootModes[2].u.ethBoot.bootFormat             = ibl_BOOT_FORMAT_NAME;
174
175
176     /* Use the e-fuse value */
177     ibl.bootModes[2].u.ethBoot.ethInfo.hwAddress[0] = 0;
178     ibl.bootModes[2].u.ethBoot.ethInfo.hwAddress[1] = 0;
179     ibl.bootModes[2].u.ethBoot.ethInfo.hwAddress[2] = 0;
180     ibl.bootModes[2].u.ethBoot.ethInfo.hwAddress[3] = 0;
181     ibl.bootModes[2].u.ethBoot.ethInfo.hwAddress[4] = 0;
182     ibl.bootModes[2].u.ethBoot.ethInfo.hwAddress[5] = 0;
183
184
185     ibl.bootModes[2].u.ethBoot.ethInfo.fileName[0] = 'a';
186     ibl.bootModes[2].u.ethBoot.ethInfo.fileName[1] = 'p';
187     ibl.bootModes[2].u.ethBoot.ethInfo.fileName[2] = 'p';
188     ibl.bootModes[2].u.ethBoot.ethInfo.fileName[3] = '.';
189     ibl.bootModes[2].u.ethBoot.ethInfo.fileName[4] = 'o';
190     ibl.bootModes[2].u.ethBoot.ethInfo.fileName[5] = 'u';
191     ibl.bootModes[2].u.ethBoot.ethInfo.fileName[6] = 't';
192     ibl.bootModes[2].u.ethBoot.ethInfo.fileName[7] = '\0';
193     ibl.bootModes[2].u.ethBoot.ethInfo.fileName[8] = '\0';
194     ibl.bootModes[2].u.ethBoot.ethInfo.fileName[9] = '\0';
195     ibl.bootModes[2].u.ethBoot.ethInfo.fileName[10] = '\0';
196     ibl.bootModes[2].u.ethBoot.ethInfo.fileName[11] = '\0';
197     ibl.bootModes[2].u.ethBoot.ethInfo.fileName[12] = '\0';
198     ibl.bootModes[2].u.ethBoot.ethInfo.fileName[13] = '\0';
199     ibl.bootModes[2].u.ethBoot.ethInfo.fileName[14] = '\0';
200
201     ibl.bootModes[2].u.ethBoot.blob.startAddress = 0x80000000;
202     ibl.bootModes[2].u.ethBoot.blob.sizeBytes   = 0x20000000;
203     ibl.bootModes[2].u.ethBoot.blob.branchAddress = 0x80000000;
204
205     ibl.chkSum = 0;
206 }

```

В таблице 4-1 дано краткое описание основных конфигурационных параметров и их возможных значений.

Параметры, не описанные в таблице 4-1, связаны с аппаратной конфигурацией периферии и прочего оборудования модуля SAMC-403 и их изменение не рекомендуется.

Таблица 4-1: Основные конфигурационные параметры в файле «i2cConfig.gel»

Параметр	Описание
ibl.bootModes[2].u.ethBoot.doBootp	Если равен TRUE, IBL будет пытаться получить сетевую конфигурацию по протоколу BOOTP. В противном случае, будут использоваться параметры конфигурации сети, описанные ниже.

*Продолжение таблицы на следующей странице*

Продолжение таблицы 4-1

Параметр	Описание
ibl.bootModes[2].u.ethBoot.bootFormat	Задаёт формат загружаемого образа. Может принимать одно из следующих значений: <ul style="list-style-type: none"> <li>• ibl_BOOT_FORMAT_COFF — объектный формат COFF. Загружается через встроенный в IBL загрузчик COFF файлов;</li> <li>• ibl_BOOT_FORMAT_ELF — объектный формат ELF. Загружается через встроенный в IBL загрузчик ELF файлов;</li> <li>• ibl_BOOT_FORMAT_BBLOB — бинарный формат готовый к загрузке на модуле (не требующий соответствующего загрузчика);</li> <li>• ibl_BOOT_FORMAT_AUTO — автоматическое определение формата по сигнатуре файла;</li> <li>• ibl_BOOT_FORMAT_NAME — автоматическое определение формата по расширению загружаемого файла («.out» — COFF, «.elf» — ELF, «.bin» — BBLOB).</li> </ul>
ibl.bootModes[2].u.ethBoot.ethInfo.hwAddress[0...5]	Задаёт значение аппаратного MAC-адреса сетевого интерфейса. Если все значения равны 0, используется встроенный производителем в процессор MAC-адрес.
ibl.bootModes[2].u.ethBoot.ethInfo.fileName[0...63]	Задаёт имя файла загрузки. Максимальная длина имени файла составляет 64 символа. Последним символом имени файла загрузки обязательно должен быть символ «\0».
ibl.bootModes[2].u.ethBoot.ethInfo.ipAddr	Позволяет задать фиксированный IP-адрес. Октеты IP-адреса в файле «i2cConfig.gel» записываются через запятую.
ibl.bootModes[2].u.ethBoot.ethInfo.serverIp	Задаёт IP-адрес сервера загрузки в случае, если ibl.bootModes[2].u.ethBoot.doBootp = FALSE.
ibl.bootModes[2].u.ethBoot.ethInfo.gatewayIp	Задаёт IP-адрес основного шлюза в случае, если ibl.bootModes[2].u.ethBoot.doBootp = FALSE.
ibl.bootModes[2].u.ethBoot.ethInfo.netmask	Задаёт маску подсети в случае, если ibl.bootModes[2].u.ethBoot.doBootp = FALSE.
ibl.bootModes[2].u.ethBoot.useBootpServerIp	Если равен FALSE, то в качестве IP-адреса сервера загрузки будет использовано значение параметра ibl.bootModes[2].u.ethBoot.ethInfo.serverIp. Если равен TRUE, то будет использован IP-адрес сервера загрузки, указанный в BOOTP-ответе от BOOTP-сервера. Данный параметр имеет значение только в случае, когда ibl.bootModes[2].u.ethBoot.doBootp = TRUE.
ibl.bootModes[2].u.ethBoot.useBootpFileName	Если равен FALSE, то в качестве имени файла загрузки будет использовано значение параметра ibl.bootModes[2].u.ethBoot.ethInfo.fileName[0...63]. Если равен TRUE, то будет использовано имя файла загрузки, указанное в BOOTP-ответе от BOOTP-сервера. Данный параметр имеет значение только в случае, когда ibl.bootModes[2].u.ethBoot.doBootp = TRUE.

## Приложение А Аппаратная конфигурация модуля SAMC-403

При загрузке и отладке программ напрямую через JTAG, без использования загрузчика IBL, положение переключателей на модуле SAMC-403 необходимо установить в соответствии с таблицей А-1.

Таблица А-1: Положение переключателей для программирования модуля SAMC-403

Переключатель	1	2	3	4
SW3	OFF	ON	ON	ON
SW4	ON	ON	ON	ON
SW5	ON	ON	ON	ON
SW6	ON	ON	ON	ON

Для загрузки образа приложения с NAND-флеш памяти, положение переключателей на модуле SAMC-403 необходимо установить в соответствии с таблицей А-2.

Таблица А-2: Положение переключателей для загрузки модуля SAMC-403 с NAND-флеш памяти

Переключатель	1	2	3	4
SW3	OFF	OFF	ON	OFF
SW4	ON	OFF	ON	ON
SW5	ON	ON	ON	OFF
SW6	ON	ON	ON	ON

Для загрузки образа приложения с TFTP-сервера, положение переключателей на модуле SAMC-403 необходимо установить в соответствии с таблицей А-3.

Таблица А-3: Положение переключателей для загрузки модуля SAMC-403 с TFTP-сервера

Переключатель	1	2	3	4
SW3	OFF	OFF	ON	OFF
SW4	ON	ON	OFF	ON
SW5	ON	ON	ON	OFF
SW6	ON	ON	ON	ON