



SAMC-404. Загрузчик IBL

Руководство пользователя

Версия 1.1



Код документа: UG-SAMC-404-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
1.1	—	Внесены правки в таблицу 5-1
1.0	—	Начальная версия

Содержание

Список рисунков	4
Список таблиц	4
Список листингов	4
Перечень сокращений и условных обозначений	5
1 Загрузчик IBL	6
2 Процесс сетевой загрузки модуля SAMC-404 с IBL	7
3 Сборка IBL из исходных кодов	8
3.1 Сборка в Windows системе	8
3.2 Сборка в Linux системе	12
4 Запись образа IBL в EEPROM	14
5 Конфигурация IBL	17

Список рисунков

3-1	Установка MinGW. Выбор каталога репозитория	8
3-2	Установка MinGW. Окно лицензии	9
3-3	Установка MinGW. Выбор папки для установки	9
3-4	Установка MinGW. Выбор устанавливаемых компонентов	10
3-5	Приглашение командной строки MinGW Shell	10
3-6	Редактирование файла «setupenvMsys.sh» в редакторе vim	11
3-7	Редактирование файла «setupenvMsys.sh» в редакторе WordPad	11
3-8	Редактирование файла «setupenvMsys.sh» в редакторе nano	13
4-1	Запуск целевой конфигурации оборудования	15
4-2	Подключение к процессору	15
4-3	Окно выбора файла для загрузки в DSP	16
5-1	Запуск программы «i2cram_0x51_c6678_le_0x500.out»	17
5-2	Меню загрузки GEL файлов	17
5-3	Меню загрузки GEL файлов	18
5-4	Меню скриптов GEL файлов для модуля SAMC-404	18

Список таблиц

5-1	Основные конфигурационные параметры в файле «i2cConfig.gel»	20
-----	---	----

Список листингов

5-1	Функция setConfig_samc404() файла «i2cConfig.gel»	18
-----	---	----

Перечень сокращений и условных обозначений

BOOTP	Bootstrap Protocol	7, 20, 21
CCS	Code Composer Studio	8, 14
DSP	Digital Signal Processor	4, 7, 8, 16
EEPROM	Electrically Erasable Programmable Read-Only Memory	6–8, 12, 14, 15, 17, 18
EMAC	Ethernet Media Access Controller	7
GEL	General Extension Language	12, 17, 18
I²C	Inter-Integrated Circuit	8, 14
IBL	Intermediate Boot Loader	6–8, 10, 12–14, 17, 18, 20
IP	Internet Protocol	20, 21
LTS	Long Term Support	12
MAC	Media Access Control	20
PLL	Phase Locked Loop controller	7
SGMII	Serial Gigabit Media Independent Interface	7
TFTP	Trivial File Transfer Protocol	6, 7

1 Загрузчик IBL

Загрузчик IBL (Intermediate Boot Loader) позволяет выполнять загрузку приложений на модуль SAMC-404 по Ethernet TFTP сервера. В данном разделе дается краткое описание возможностей IBL, описан алгоритм загрузки модуля SAMC-404 с загрузчиком IBL, описывается процесс сборки IBL из исходных кодов, конфигурация загрузчика для работы на модуле SAMC-404, даются инструкции по правильной подготовке и записи образа IBL в EEPROM.

2 Процесс сетевой загрузки модуля SAMC-404 с IBL

Алгоритм сетевой загрузки модуля SAMC-404 с записанным в EEPROM загрузчиком IBL после включения питания выглядит следующим образом:

- 1) встроенный в чип C6457 загрузчик выполняет чтение из I²C EEPROM блока со смещением 0x400 байт относительно начала. В данном блоке содержится таблица параметров загрузки (boot parameters table);
- 2) после того как будет прочитана таблица параметров загрузки, она запоминается в памяти DSP. Далее, выполняется перезапуск загрузчика C6457. Происходит считывание содержимого EEPROM в память DSP и управление передается по адресу, записанному в таблице параметров загрузки, то есть управление передается программе первого этапа загрузки IBL;
- 3) на первом этапе загрузки IBL происходит считывание структуры карты адресного пространства EEPROM. В этой структуре содержатся адреса структуры конфигурации и точки входа в программу второго этапа загрузки IBL. Управление передается программе второго этапа загрузки IBL;
- 4) программа второго этапа загрузки IBL выполняет инициализацию всех периферийных устройств, необходимых для выполнения Ethernet загрузки (PLL (Phase Locked Loop controller), EMAC, SGMII (Serial Gigabit Media Independent Interface), и т. п.);
- 5) IBL выполняет широковещательный запрос сетевых параметров и адрес файла для загрузки используя BOOTP протокол. BOOTP запросы на получение сетевой конфигурации посылаются IBL бесконечно, до тех пор, пока не будет получен ответ от BOOTP-сервера;
- 6) если получен ответ от BOOTP-сервера, то выполняется попытка загрузки указанного в BOOTP ответе файла образа с удаленного TFTP-сервера в оперативную память модуля SAMC-404. После успешной загрузки образа с TFTP-сервера, выполняется запуск загруженного образа.

3 Сборка IBL из исходных кодов

Для сборки IBL потребуется установленный компилятор для DSP Texas Instruments серии C6000. Данный компилятор входит в состав системы разработки CCS (Code Composer Studio) компании Texas Instruments.

Дистрибутив CCS можно найти на сопроводительном диске к модулю SAMC-404 в папке «software/ccs». Исходные коды IBL можно найти на сопроводительном диске к модулю SAMC-404 в папке «/ibl/src».

Результатом сборки IBL из исходных кодов является готовый к записи в I²C EEPROM образ загрузчика.

Внимание



Перед выполнением сборки загрузчика, описанной в данном разделе, перепишите с сопроводительного диска к модулю SAMC-404 папку «ibl» на жесткий диск компьютера. Далее, предполагается, что папка «ibl» переписана в папку «D:/ibl».

3.1 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-404 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 3-1) рекомендуется выбрать пункт «Use pre-packaged repository catalogues» (использовать каталог репозитория с заранее собранными пакетами).

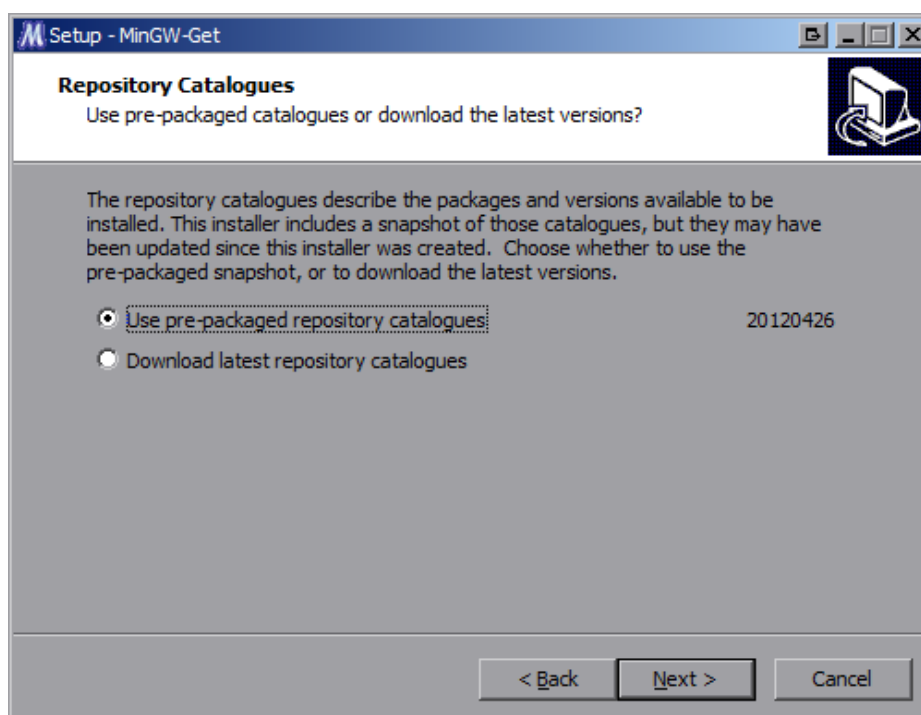


Рисунок 3-1: Установка MinGW. Выбор каталога репозитория

В окне лицензии (рисунок 3-2), прочитайте лицензию, и если вы согласны с ней, выберите пункт «I accept the agreement» и нажмите кнопку «Next».

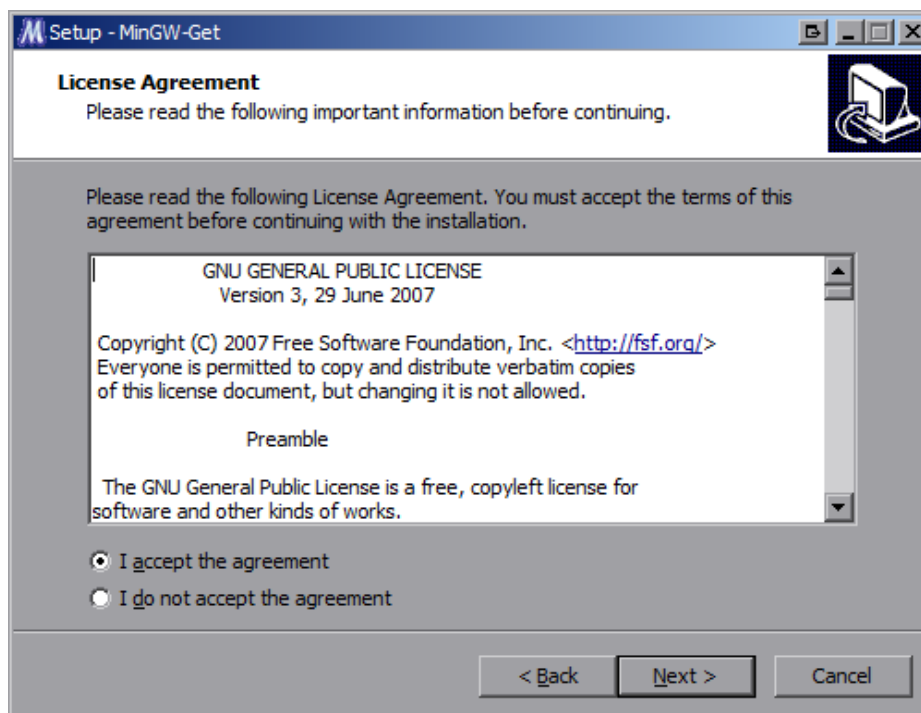


Рисунок 3-2: Установка MinGW. Окно лицензии

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

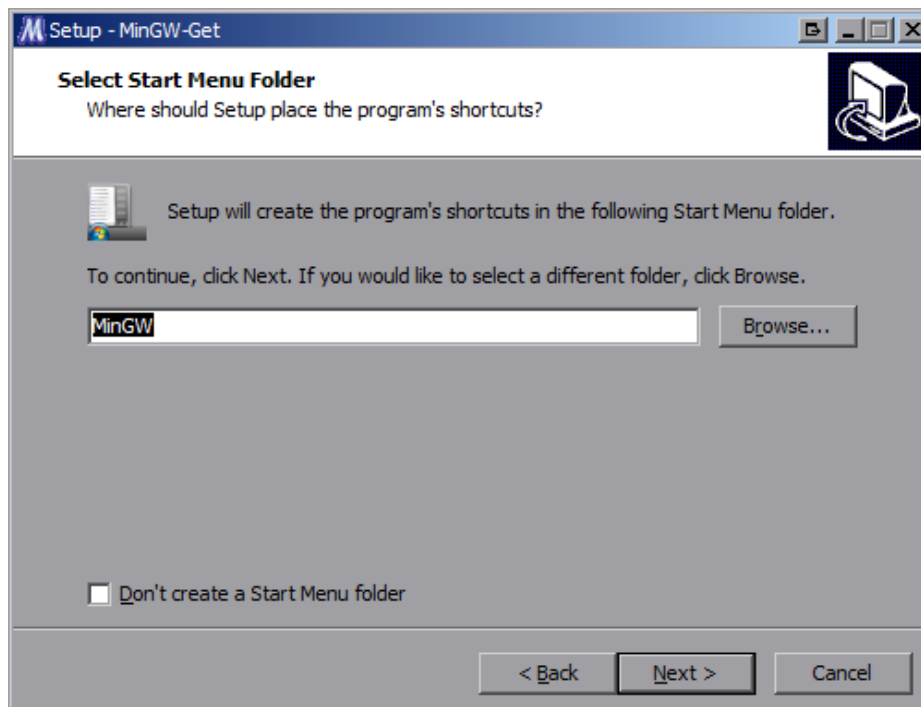


Рисунок 3-3: Установка MinGW. Выбор папки для установки

В окне выбора устанавливаемых компонентов (см. рисунок 3-4) 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.

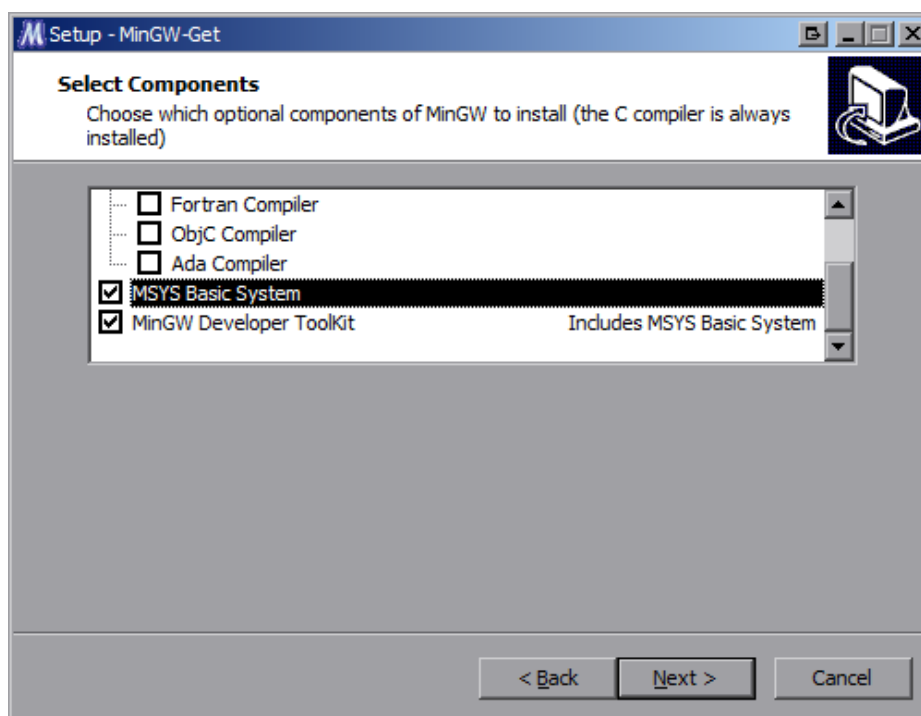


Рисунок 3-4: Установка MinGW. Выбор устанавливаемых компонентов

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 3-5). Все последующие действия по сборке IBL будут производиться путем ввода команд в MinGW Shell.

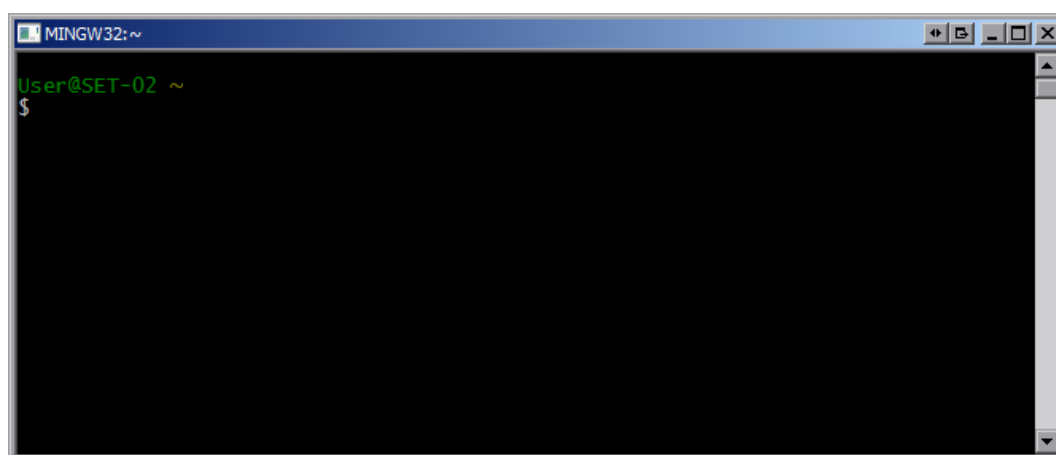
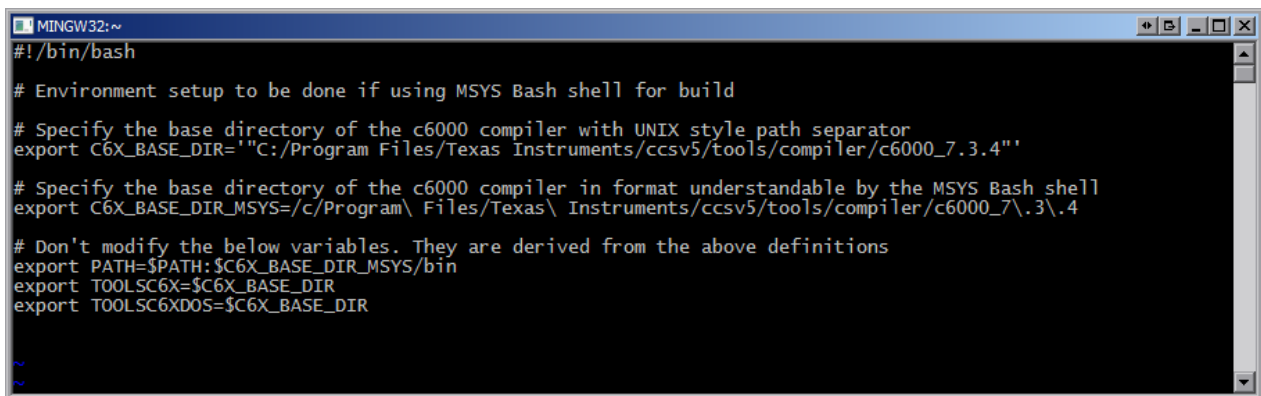


Рисунок 3-5: Приглашение командной строки MinGW Shell

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 3-6 приведен снимок экрана MinGW Shell с открытым файлом

«setupenvMsys.sh» в редакторе vim. На данном рисунке пути установлены в соответствии с путями указанными выше.



```

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:/Program Files/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/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

```

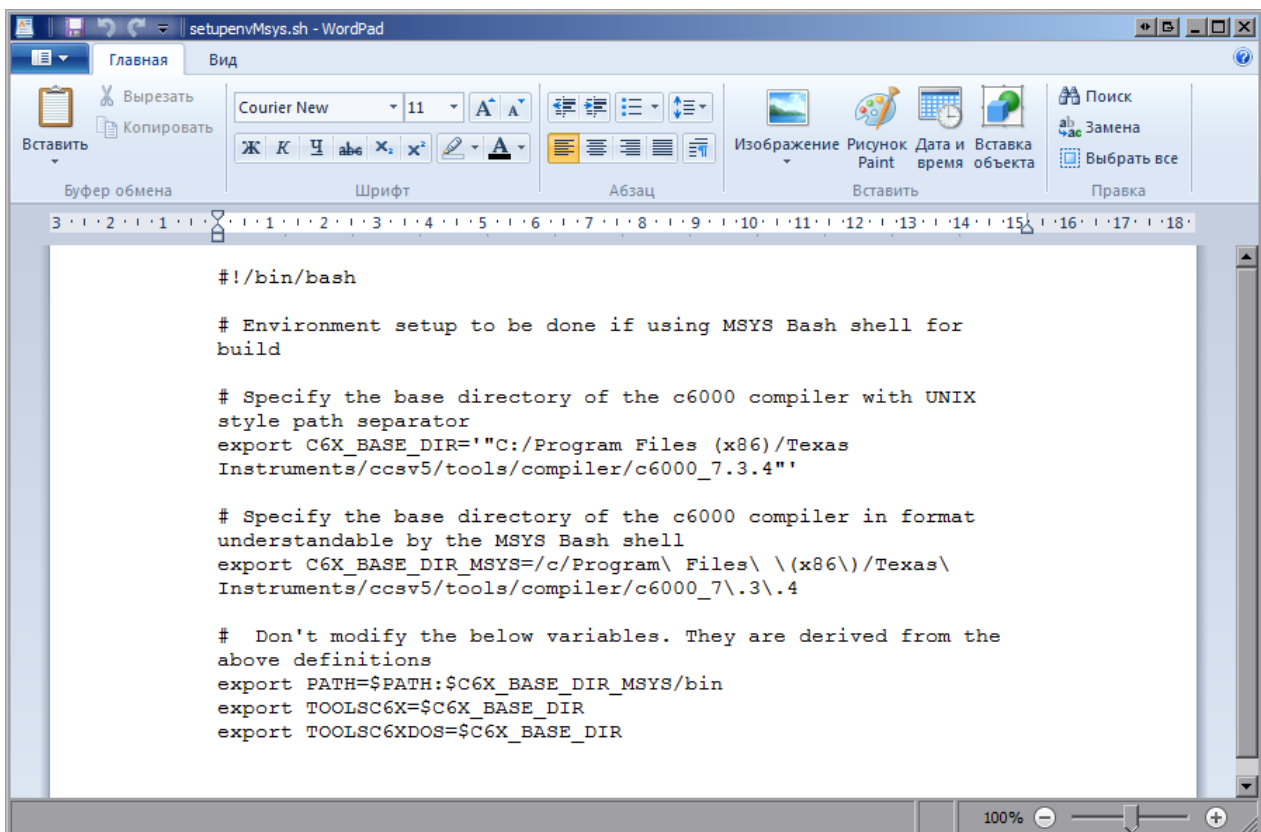
Рисунок 3-6: Редактирование файла «setupenvMsys.sh» в редакторе vim

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

```
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 3-7).



```

setupenvMsys.sh - WordPad
Главная Вид
Вставить Вырезать Копировать
Буфер обмена Шрифт Абзац Вставить Пуска
Courier New 11
Изображение Рисунок Дата и Вставка
Paint время объекта
Поиск Замена Выбрать все
3 2 1 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18

#!/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 3-7: 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 samc404 ENDIAN=little
```

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 5);
- «i2cparam_0x51_samc404_le_0x0.out» — программа конфигурации IBL для Code Composer Studio (см. раздел 5);
- «i2crom_0x51_samc40_le.bin» — бинарный образ IBL для загрузки в EEPROM (см. раздел 4);
- «i2crom_0x51_samc40_le.dat» — образ IBL для загрузки в EEPROM в формате Code Composer Studio (см. раздел 4).

3.2 Сборка в Linux системе

Сборка IBL на Linux системе мало чем отличается от сборки для Windows систем, которая описана в разделе 3.1.

В данном разделе, в качестве Linux дистрибутива, предполагается использование 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 с компилятору для процессоров C6000 серии. На рисунке 3-8 приведен снимок экрана с открытым файлом «setupenvLnx.sh» в редакторе nano. На данном рисунке пути установлены в соответствии с путями указанными выше.

```

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 TOOLSBIOOSC6XDOS=$C6X_BASE_DIR

^G Get Help      ^C 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 3-8: Редактирование файла «setupenvMsys.sh» в редакторе nano

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 evm_c6678_i2c ENDIAN=little
```

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_0x50_samc404_le_0x0.out»;
- «i2crom_0x50_samc404_le.bin»;
- «i2crom_0x50_samc404_le.dat».

Краткое описание назначения каждого из файлов дано в конце раздела 3.1, описывающего процесс сборки IBL в Windows системе.

4 Запись образа IBL в EEPROM

Запись образа IBL в EEPROM производится с помощью специальной программы, которая имеется на сопроводительном диске к модулю SAMC-404. Для записи образа в EEPROM потребуется установленная система разработки CCS. Рекомендуется использовать CCS 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 3, или использовать заранее собранный образ «i2crom_0x50_samc404_le.bin», который имеется на сопроводительном диске к модулю SAMC-404 в папке «ibl/bin».

Для записи образов в EEPROM модуля, используется специальная программа eepromwriter, запускаемая посредством среды разработки CCS. Исходный код и готовый к запуску бинарный файл программы eepromwriter можно найти на сопроводительном диске к модулю SAMC-404 в папке «ibl/eepromwriter». Перед началом процесса записи, необходимо переписать с сопроводительного диска файлы из папки «ibl/eepromwriter/bin» на жесткий диск (предположим, в папку «D:/eepromwriter»). Файл «eepromwriter_samc404_le.out» является скомпилированной программой, готовой для загрузки в память процессора модуля. В файле «eepromwriter_input.txt» содержится конфигурация программы eepromwriter, в частности, в этом файле задается имя файла образа для записи в EEPROM.

Файл «eepromwriter_input.txt» имеет следующий вид:

```
file_name   = i2crom_0x50_samc404_le.bin
bus_addr    = 80
start_addr  = 0
swap_data   = 0
```

В параметре «file_name» задается имя файла образа для записи в I²C EEPROM (в данном случае — «i2crom_0x50_samc404_le.bin»). Файл образа для записи в EEPROM должен находиться в той же папке, что и файлы «eepromwriter_samc404.out» и «eepromwriter_input.txt».

В параметре «bus_addr» задается адрес I²C шины целевой EEPROM. Для модуля SAMC-404 значение этого параметра должно быть равно 80 (0x50).

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-404 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).

Рассмотрим процедуру записи образа в EEPROM модуля SAMC-404. В первую очередь, необходимо запустить в CCS требуемую целевую конфигурацию оборудования, в зависимости от используемого отладчика (см. рисунок 4-1). Файлы целевых конфигураций имеются на сопроводительном диске к модулю SAMC-404 в папке «ccs_config».

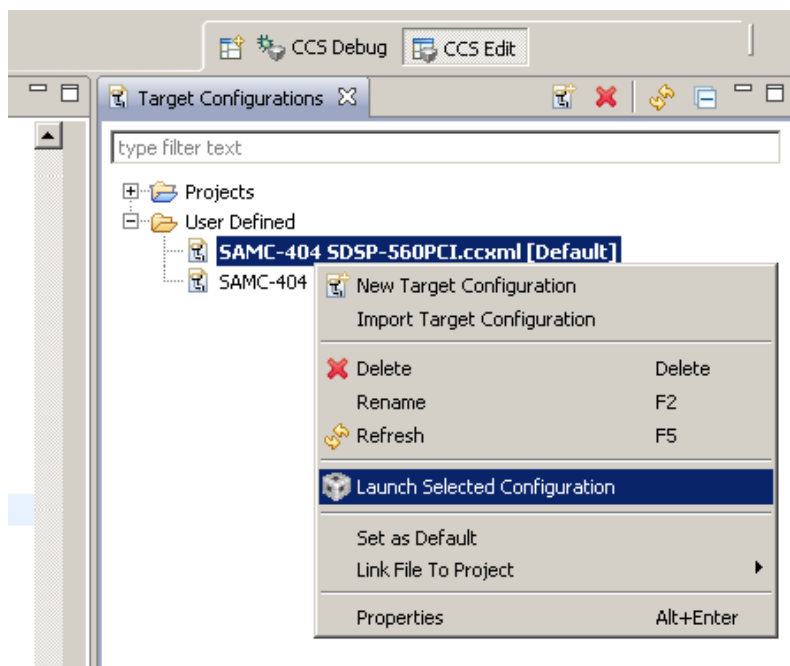


Рисунок 4-1: Запуск целевой конфигурации оборудования

После запуска конфигурации, в окне «Debug» выбирается нужный процессор из списка, путем нажатия правой кнопкой мыши на нем, для открытия контекстного меню, и выбора пункта меню «Connect Target» (см. рисунок 4-2).

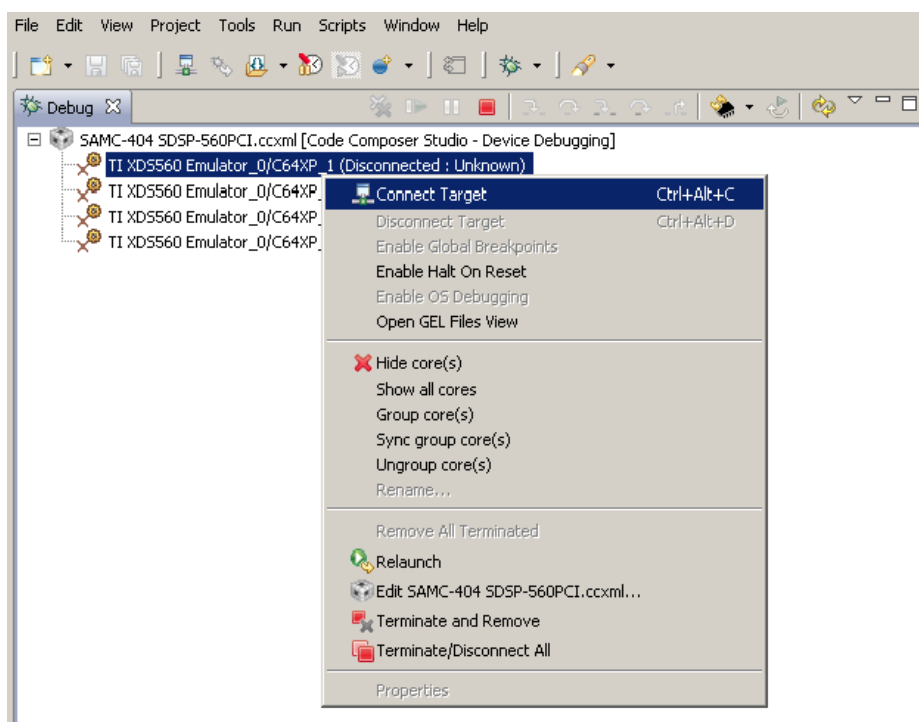


Рисунок 4-2: Подключение к процессору

Далее, в процессор загружается программа eepromwriter для записи EEPROM. Для этого нужно выбрать из главного меню пункт «Run > Load > Load Program...». Для выбора файла, в появившемся окне (рисунок 4-3), необходимо нажать на кнопку «Browse...» и выбрать файл «eepromwriter_samc404_le.out».

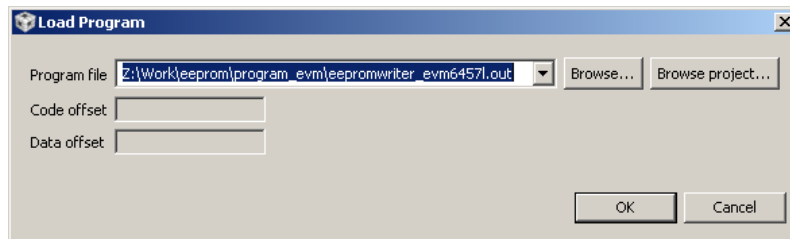


Рисунок 4-3: Окно выбора файла для загрузки в DSP.

После загрузки программы в процессор, её необходимо запустить, нажав на клавиатуре клавишу F8 или выбрав пункт меню «Run > Resume».

5 Конфигурация IBL

Конфигурация IBL выполняется путем записи блока с конфигурационными параметрами в EEPROM с определенным смещением. Для облегчения этого процесса, используется специальная программа «i2cparam_0x50_samc404_le_0x0.out», которая собирается вместе с IBL (см. раздел 3). Эта программа предназначена для загрузки в память процессора модуля SAMC-404, аналогично программе eepromwriter, которая описана в разделе 4.

После загрузки в память процессора программы «i2cparam_0x51_c6678_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 5-1).

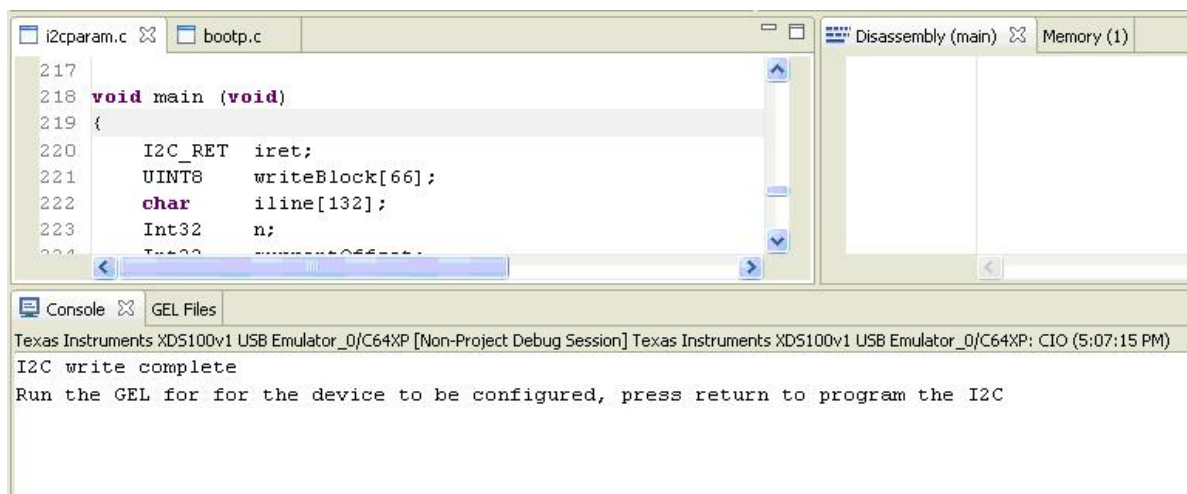
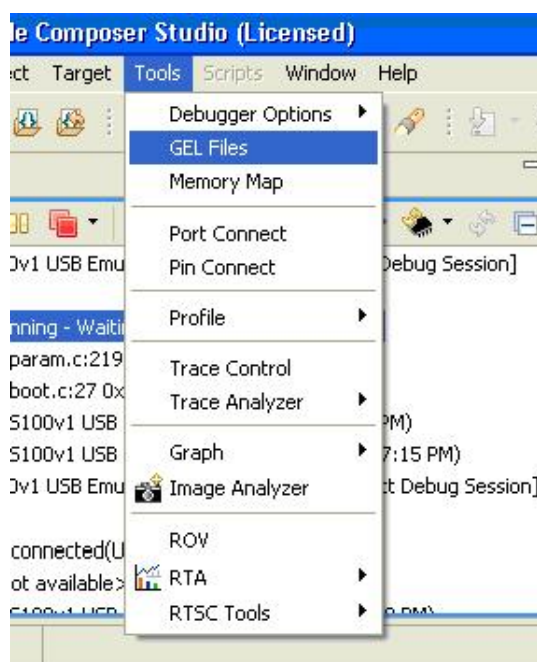


Рисунок 5-1: Запуск программы «i2cparam_0x51_c6678_le_0x500.out»

Для загрузки GEL файла, нужно выбрать пункт главного меню «Tools > GEL Files» (рисунок 5-2), в результате чего откроется окно (рисунок 5-3), в котором необходимо загрузить GEL файл, путем нажатия правой кнопки мыши и выбора пункта меню «Load GEL...».



11

Рисунок 5-2: Меню загрузки GEL файлов

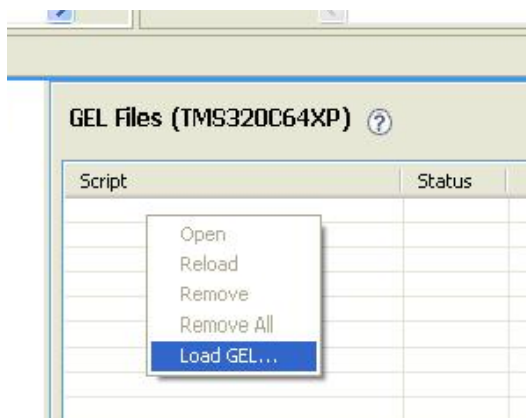


Рисунок 5-3: Меню загрузки GEL файлов

В открывшемся окне выбора файла, необходимо выбрать файл «i2cConfig.gel», который расположен в той же папке, куда записывается образ IBL после сборки (см. раздел 3). Также, файл «i2cConfig.gel» можно найти на сопроводительном диске к модулю SAMC-404 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.

Для применения конфигурации, описанной в файле «i2cConfig.gel», необходимо выбрать в главном меню пункт «Scripts > SET SAMC-404 IBL > setConfig_samc404», как показано на рисунке 5-4.

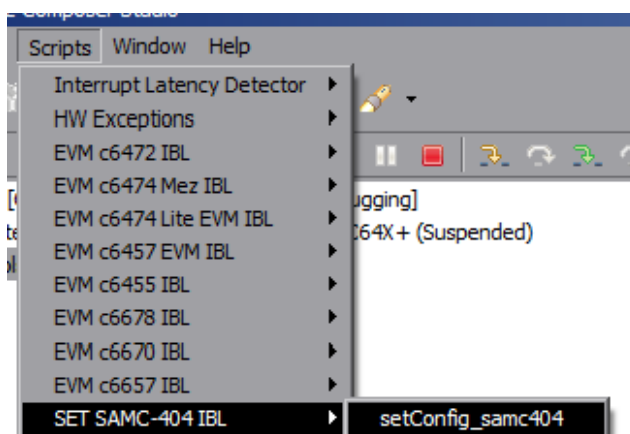


Рисунок 5-4: Меню скриптов GEL файлов для модуля SAMC-404

Для записи конфигурации в EEPROM, необходимо переключиться в окно «Console» и нажать клавишу Enter на клавиатуре (см. рисунок 5-1).

В конфигурационном файле «i2cConfig.gel» содержатся конфигурационные параметры IBL, которые можно задавать по своему усмотрению путем редактирования этого файла. При сборке образа (раздел 3), 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-404, расположены в этом файле в функции setConfig_samc404().

Ниже приведен оригинальный листинг функции setConfig_samc404() из файла «i2cConfig.gel» с параметрами по умолчанию:

Листинг 5-1: Функция setConfig_samc404() файла «i2cConfig.gel»

```
hotmenu setConfig_samc404()
{
    ibl.iblMagic = ibl_MAGIC_VALUE;
    ibl.iblEvmType = ibl_EVM_C6457L;

    ibl.pllConfig[ibl_MAIN_PLL].doEnable      = TRUE;
    ibl.pllConfig[ibl_MAIN_PLL].prediv       = 1;
```

```

ibl.pllConfig[ibl_MAIN_PLL].mult      = 20;
ibl.pllConfig[ibl_MAIN_PLL].postdiv   = 1;
ibl.pllConfig[ibl_MAIN_PLL].pllOutFreqMhz = 1000;

/* The DDR PLL. The multipliers/dividers are fixed, so are really dont cares */
ibl.pllConfig[ibl_DDR_PLL].doEnable = TRUE;

/* The network PLL. The multipliers/dividers are fixed */
ibl.pllConfig[ibl_NET_PLL].doEnable = TRUE;

/* EMIF configuration */
ibl.ldrConfig.configDdr = TRUE;

ibl.ldrConfig.uEmif.emif3p1.sdcfg = 0x00d38a32; /* cas5, 8 banks, 10 bit column */
ibl.ldrConfig.uEmif.emif3p1.sdrfc = 0x00000a0e; /* Refresh 333Mhz */
ibl.ldrConfig.uEmif.emif3p1.sdtim1 = 0x832474da; /* Timing 1 */
ibl.ldrConfig.uEmif.emif3p1.sdtim2 = 0x0144c742; /* Timing 2 */
ibl.ldrConfig.uEmif.emif3p1.dmcctl = 0x001800C6;

/* SGMII 0 is present */
ibl.sgmiiConfig[0].configure = TRUE;
ibl.sgmiiConfig[0].adviseAbility = 0x9801;
ibl.sgmiiConfig[0].control      = 0x20;
ibl.sgmiiConfig[0].txConfig     = 0x00000e21;
ibl.sgmiiConfig[0].rxConfig     = 0x00081021;
ibl.sgmiiConfig[0].auxConfig    = 0x00000009;

/* There is no port 1 on the 6457 */
ibl.sgmiiConfig[1].configure = FALSE;

/* MDIO configuration */
ibl.mdioConfig.nMdioOps = 5;
ibl.mdioConfig.mdioClkDiv = 0xa5;
ibl.mdioConfig.interDelay = 3000; /* ~2ms at 1000 MHz */

ibl.mdioConfig.mdio[0] = (1 << 30) | ( 4 << 21) | (27 << 16) | 0x0081;
ibl.mdioConfig.mdio[1] = (1 << 30) | (26 << 21) | (14 << 16) | 0x0047;
ibl.mdioConfig.mdio[2] = (1 << 30) | ( 0 << 21) | (14 << 16) | 0x8140;
ibl.mdioConfig.mdio[3] = (1 << 30) | ( 1 << 21) | (22 << 16) | 0x043e;
ibl.mdioConfig.mdio[4] = (1 << 30) | ( 0 << 21) | ( 1 << 16) | 0x9140;

/* spiConfig and emifConfig not needed */

/* Ethernet configuration for Boot mode 0 */
ibl.bootModes[0].bootMode = ibl_BOOT_MODE_TFTP;
ibl.bootModes[0].priority = ibl_HIGHEST_PRIORITY;
ibl.bootModes[0].port = 0;

/* Bootp is disabled. The server and file name are provided here */
ibl.bootModes[0].u.ethBoot.doBootp = TRUE;
ibl.bootModes[0].u.ethBoot.useBootpServerIp = TRUE;
ibl.bootModes[0].u.ethBoot.useBootpFileName = TRUE;
ibl.bootModes[0].u.ethBoot.bootFormat = ibl_BOOT_FORMAT_COFF;

SETIP(ibl.bootModes[0].u.ethBoot.ethInfo.ipAddr, 158,218,100,115);
SETIP(ibl.bootModes[0].u.ethBoot.ethInfo.serverIp, 158,218,100,25);
SETIP(ibl.bootModes[0].u.ethBoot.ethInfo.gatewayIp, 158,218,100,2);
SETIP(ibl.bootModes[0].u.ethBoot.ethInfo.netmask, 255,255,255,0);

/* Set the hardware address as 0 so the e-fuse value will be used */
ibl.bootModes[0].u.ethBoot.ethInfo.hwAddress[0] = 0;
ibl.bootModes[0].u.ethBoot.ethInfo.hwAddress[1] = 0;
ibl.bootModes[0].u.ethBoot.ethInfo.hwAddress[2] = 0;
ibl.bootModes[0].u.ethBoot.ethInfo.hwAddress[3] = 0;
ibl.bootModes[0].u.ethBoot.ethInfo.hwAddress[4] = 0;
ibl.bootModes[0].u.ethBoot.ethInfo.hwAddress[5] = 0;
ibl.bootModes[0].u.ethBoot.ethInfo.fileName[0] = 's';
ibl.bootModes[0].u.ethBoot.ethInfo.fileName[1] = 'a';
ibl.bootModes[0].u.ethBoot.ethInfo.fileName[2] = 'm';
ibl.bootModes[0].u.ethBoot.ethInfo.fileName[3] = 'c';
ibl.bootModes[0].u.ethBoot.ethInfo.fileName[4] = '4';

```

```

ibl.bootModes[0].u.ethBoot.ethInfo.fileName[5] = '0';
ibl.bootModes[0].u.ethBoot.ethInfo.fileName[6] = '4';
ibl.bootModes[0].u.ethBoot.ethInfo.fileName[7] = '-';
ibl.bootModes[0].u.ethBoot.ethInfo.fileName[8] = '1';
ibl.bootModes[0].u.ethBoot.ethInfo.fileName[9] = 'e';
ibl.bootModes[0].u.ethBoot.ethInfo.fileName[10] = '.';
ibl.bootModes[0].u.ethBoot.ethInfo.fileName[11] = 'b';
ibl.bootModes[0].u.ethBoot.ethInfo.fileName[12] = 'i';
ibl.bootModes[0].u.ethBoot.ethInfo.fileName[13] = 'n';
ibl.bootModes[0].u.ethBoot.ethInfo.fileName[14] = '\0';

/* Even though the entire range of DDR2 is chosen, the load will
 * stop when the ftp reaches the end of the file */
ibl.bootModes[0].u.ethBoot.blob.startAddress = 0xe0000000;
ibl.bootModes[0].u.ethBoot.blob.sizeBytes = 0x20000000;
ibl.bootModes[0].u.ethBoot.blob.branchAddress = 0xe0000000;

/* bootMode[1] not configured */
ibl.bootModes[1].bootMode = ibl_BOOT_MODE_NONE;

/* bootMode[2] not configured */
ibl.bootModes[2].bootMode = ibl_BOOT_MODE_NONE;

ibl.chkSum = 0;
}

```

В таблице 5-1 дано краткое описание основных конфигурационных параметров и их возможных значений.

Параметры, не описанные в таблице 5-1, связаны с аппаратной конфигурацией периферии и прочего оборудования модуля SAMC-404 и их изменение не рекомендуется.

Таблица 5-1: 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.
ibl.bootModes[2].u.ethBoot.bootFormat	<p>Задаёт формат загружаемого образа. Может принимать одно из следующих значений:</p> <ul style="list-style-type: none"> ibl_BOOT_FORMAT_COFF — объектный формат COFF. Загружается через встроенный в IBL загрузчик COFF файлов; ibl_BOOT_FORMAT_ELF — объектный формат ELF. Загружается через встроенный в IBL загрузчик ELF файлов; ibl_BOOT_FORMAT_BBLOB — бинарный формат готовый к загрузке на модуле (не требующий соответствующего загрузчика); ibl_BOOT_FORMAT_AUTO — автоматическое определение формата по сигнатуре файла; ibl_BOOT_FORMAT_NAME — автоматическое определение формата по расширению загружаемого файла («.out» — COFF, «.elf» — ELF, «.bin» — BBLOB).
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.

Продолжение таблицы на следующей странице

Продолжение таблицы 5-1

Параметр	Описание
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.