

# Программирование встроенной Flash-памяти микроконтроллеров TMS320xx24xx

an@aep.mpei.ac.ru

Алексей Сергеевич Анучин

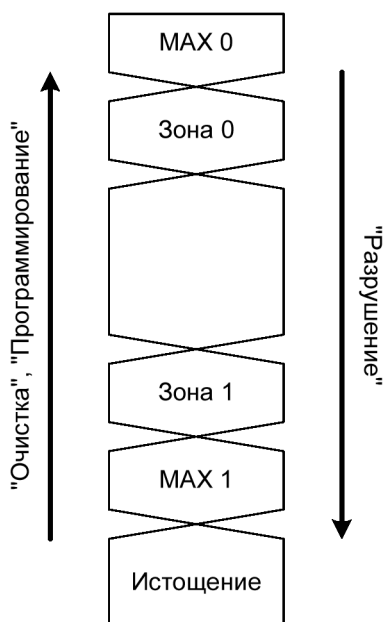
Владимир Филиппович Козаченко

В учебно-научно-консультационный центр “Texas Instruments – МЭИ” регулярно обращаются разработчики стран бывшего СНГ по многим вопросам, но, пожалуй, самое большое число задаваемых вопросов касаются программирования встроенной Flash-памяти. В данной статье мы постараемся описать технологию использования доступных в настоящий момент средств для программирования и по возможности опишем подводные камни, на которые может наткнуться разработчик микропроцессорной системы.

## Технология программирования Flash-памяти

Авторы не знакомы с физическими принципами хранения и записи информации во Flash-память и не ставят задачу в этом разобраться. Мы лишь хотим рассказать про процесс программирования в терминах пользователя и заранее просим простить нас, если мы искажем техническую сторону вопроса.

Flash-память микроконтроллеров имеет страничную организацию. Размер страницы может быть разным в зависимости от микроконтроллера. Если Вы хотите изменить одно слово данных, Вам придется перезаписать всю страницу целиком. Питание модуля программирования Flash-памяти производится во всей серии контроллеров напряжением +5 Вольт, которое подается на ножку  $V_{CCP}/WDDIS$ . Ток потребляемый по этой цепи составляет от 5 до 15 мА, поэтому не следует подключать ножку через резистор. Напряжение питания не должно быть завышенным, так как в этом случае происходят необратимые процессы и память начинает прописываться нерегулярно с выпадениями отдельных бит.



Программирование памяти разделено на три этапа:

- ✓ CLEAR – «очистка», процесс записи «нулей» во все биты секции;
- ✓ ERASE – «разрушение», процесс записи «единиц» во все биты секции;
- ✓ PROGRAM – «программирование», процесс записи «нулей» в определенные биты секции.

Операция «очистки» записывает во все биты «нули» и подготавливает память для произведения операции «разрушения». С этой операции всегда необходимо начинать программирование, даже если Вы уверены, что эта операция уже совершена. Затем производится операция «разрушения». Никогда не производите операцию разрушения два раза подряд, например, в случае ее неудачного завершения — **всегда повторяйте операцию очистки!!!**

Дело в том, что при двойном «разрушении» можно разрядить ячейку до состояния «истощения», из которого зарядить ее будет уже невозможно или затруднительно. Следующей, наконец, производится операция программирования, которая состоит в заряде отдельных ячеек до логической единицы.

## TMS320F241, TMS320F243

Микроконтроллеры этих двух серий абсолютно одинаковы с точки зрения программирования Flash-памяти. Они поставляются с прописанным во Flash-память загрузчиком, который может загружать в оперативную память контроллера исполняемый код по последовательному коммуникационному интерфейсу SCI (обычно в сочетании с микросхемой MAX232 – это интерфейс RS-232). Для загрузки программ пользователя Texas Instruments предлагает воспользоваться набором утилит <http://www.ti.com/sc/docs/tools/dsp/c2000/files/f243fl.zip>, содержащих программы для персонального компьютера под Windows и алгоритмы прожига Flash-памяти для семейства 241/243. Программа пользователя, загружаемая в микроконтроллер должна содержать код, обеспечивающий ее совместную работу с последовательным загрузчиком. То есть собственный код программы должен начинаться с адреса 0040h, а RESET-вектор передавать управление последовательному загрузчику:

<b>0000h:</b>	<b>b</b>	<b>1F00h</b>
---------------	----------	--------------

Программа последовательного загрузчика получит управление после запуска процессора и по состоянию ножки процессора /BIO перейдет либо на адрес 0040h (ножка привязана к «нулю» во время RESET), либо запустит алгоритм загрузки по коммуникационному интерфейсу (ножка привязана к «единице»).

Таким образом, программа загрузчика загружает по очереди алгоритмы «очистки», «разрушения» и «программирования» памяти. Последнему алгоритму последовательно передается программа пользователя, а после – код загрузчика. В процессе загрузки загрузчика в памяти нет, поэтому в случае сбоя по каналу связи или источнику питания процессор перестает загружаться. Исправить положение можно восстановив загрузчик с помощью утилиты загрузки процессора по JTAG-интерфейсу (входит в комплект f243fl.zip), но этот метод требует наличия разведенного на плате контроллера JTAG-интерфейса и довольно дорогого внутрисхемного эмулятора.

## TMS320LF240xA (кроме TMS320LF2401A)

Микроконтроллеры TMS320LF240x (без буквы A) уже сняты с производства, поэтому рассматривать их не будем.

В новых микроконтроллерах семейства motor-control TMS320LF240xA и последующих предусмотрен режим защиты программы, прошитой во встроенную Flash-память, 64-разрядным ключом.

CSM или Code Security Module не позволяет просматривать содержимое Flash-памяти лицам, не знающим ключа. Программа, записанная в контроллер, будет функционировать правильно, если пользователь кристалла не нарушает правила безопасности интеллектуальной собственности, хранящейся в устройстве. При нарушении прав доступа к контроллеру Flash-память засекречивается, процессор не может прочитать ее содержимое. Нарушения могут быть следующие:

- ✓ подключен JTAG-интерфейс,
- ✓ процессор работает в режиме последовательной загрузки,
- ✓ контроллер был включен в режиме микропроцессора,
- ✓ значения регистров ключа и памяти ключа не совпадают.

Таким образом исключается возможность любого несанкционированного чтения текста программы. Следует опасаться только ситуации, когда программа, заложенная в контроллер, поддерживает загрузку и выполнение программ в оперативной памяти со стороны системы высокого уровня.

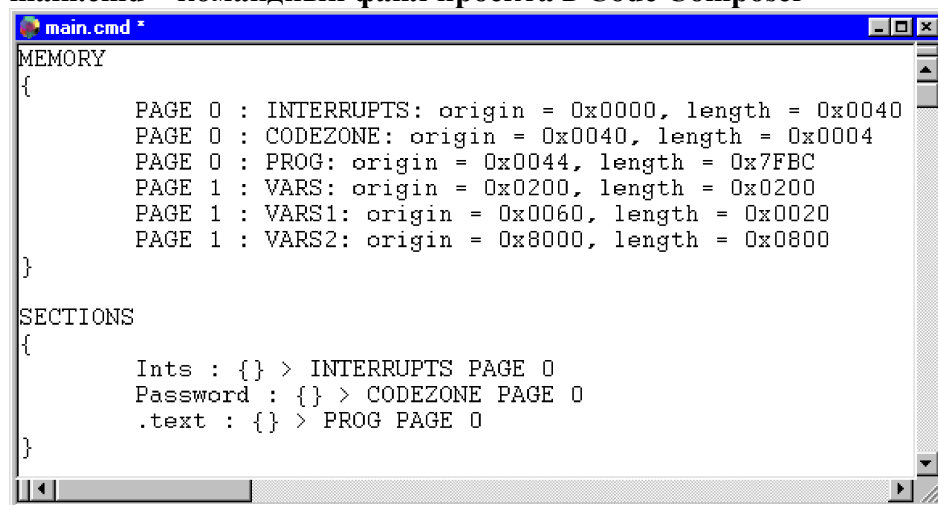
Система защиты имеет две области памяти по 64 разряда или 4 слова. Одна область находится во Flash-памяти с адреса 0040h по адрес 0043h. В ней хранится ключ доступа. Другая область расположена в памяти данных с 77F0h по 77F3h. Эта область служит для записи ключа для разблокировки Flash. Для разблокировки необходимо вначале обратиться к ключу во

Flash. Очевидно, это нужно, чтобы процессор для себя выяснил какой ключ является текущим. Затем прописываем ключ в память данных. Если ключи совпадут Вы получаете доступ к программе.

Когда Вы производите обычную отладку, то, наверняка, устанавливать код доступа Вам не нужно. Существует возможность установить «пустой» пароль 0000 0000 0000 0000h или FFFF FFFF FFFF FFFFh. Следует помнить, что даже в этом случае требуется чтение ключа из Flash-a.

Чтобы избежать проблем с программированием микроконтроллера следует правильно описать области памяти для Вашего проекта, ведь если фрагмент программы случайно попадет на область памяти ключа, этот фрагмент автоматически станет текущим паролем, и узнать его можно только путем анализа ранее загруженного объектного файла (если он сохранился). Поэтому рекомендуем воспользоваться следующими настройками.

#### **main.cmd – командный файл проекта в Code Composer**

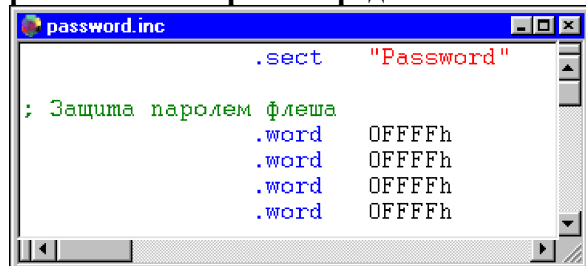


```
main.cmd *
MEMORY
{
    PAGE 0 : INTERRUPTS: origin = 0x0000, length = 0x0040
    PAGE 0 : CODEZONE: origin = 0x0040, length = 0x0004
    PAGE 0 : PROG: origin = 0x0044, length = 0x7FBC
    PAGE 1 : VARS: origin = 0x0200, length = 0x0200
    PAGE 1 : VARS1: origin = 0x0060, length = 0x0020
    PAGE 1 : VARS2: origin = 0x8000, length = 0x0800
}

SECTIONS
{
    Ints : {} > INTERRUPTS PAGE 0
    Password : {} > CODEZONE PAGE 0
    .text : {} > PROG PAGE 0
}
```

Файл определяет отдельные секции для прерываний, ключа и программы. В самое начало главного модуля следует включить ссылку на файл с ключом.

#### **password.inc – файл определения ключа**



```
password.inc
        .sect    "Password"

; Защита паролем флеша
        .word   0FFFFh
        .word   0FFFFh
        .word   0FFFFh
        .word   0FFFFh
```

Для программирования этой серии по последовательному каналу Texas Instruments 12 июня 2002 года выпустила новую версию загрузчика, доступную для свободного скачивания <http://www-s.ti.com/sc/psheets/sprc074/sprc074.zip>.

**ВНИМАНИЕ !!!** Ни в коем случае не используйте для программирования этой серии любые другие программы, выпущенные ей для TMS320LF240x (без буквы A).

Технология записи во Flash-память у этих устройств в корне изменилась, поэтому убедитесь, что в Вашем распоряжении последняя версия программного обеспечения. Программа должна быть сконфигурирована на работу с определенным PLL и кварцем. Зайдите директорию “Загрузчик\Algos\Include\”. В ней находится файл “var.h”, в котором необходимо установить ссылку на файл “Timings.XX”, где XX – тактовая частота процессора, а так же установить необходимый режим работы PLL (для TMS320LF240xA есть возможность производить загрузку на удвоенной и учетверенной частоте кварца). После этого файлы алгоритмов необходимо перекомпилировать запустив “mk\_algos.bat” из директории загрузчика.

Примите во внимание, что режим умножения частоты выбирается последовательным загрузчиком по потенциалу на ножке SCITXD, которая должна быть привязана через резистор к питанию для PLL=4 и к земле для PLL=2. SPISIMO должен быть привязан к земле, для выбора режима загрузки по SCI-интерфейсу.

Если Вы уверены, что программы утилит для загрузки подготовлены к работе с Вашей платой контроллера, то можете приступать к загрузке. Если Вы где-то ошиблись и выставили параметры не правильно, то это приведет к тому, что отдельные разряды Flash-памяти будут прописываться с ошибками. А если эти отдельные разряды окажутся в области хранения пароля, то Вы должны будете угадывать этот новый 64-разрядный пароль.

Загрузка требует привязки ножки /BOOTEN к земле, в этом случае вместо Flash-памяти в область адресов с 0000h по 00FFh попадает ПЗУ с программой загрузки по одному из последовательных интерфейсов аналогичной той, которая хранилась во Flash-памяти TMS241/243. Теперь эту программу испортить нельзя. После завершения загрузки ножка /BOOTEN привязывается к напряжению питания и программа пользователя, записанная во Flash-память начнет выполняться после перезапуска процессора. При написании пользовательской программы учтите, что переадресация RESET-вектора на загрузчик, как это требовалось в TMS320F241/243, уже не требуется.

В заключении хочется высказать благодарность отделу технической поддержки Texas Instruments за оперативную помощь по данному вопросу.

Напоминаем, что Вы можете задавать ваши вопросы по электронной почте по адресу [epic@ti.com](mailto:epic@ti.com) на английском языке. Служба поддержки TI обещает отвечать на вопросы за 48 часов, что, надо сказать, всегда выполняет. Если Вы не владеете английским или не получили удовлетворяющего Вас ответа от Texas Instruments, то обращайтесь на [an@aep.mpei.ac.ru](mailto:an@aep.mpei.ac.ru), и авторы статьи постараются ответить на Ваш вопрос.