

# 56F8xx

Serial Bootloader

*User Manual*

**56F800**  
**16-bit Digital Signal Controllers**

DSP56F8xxBLUM  
Rev. 2  
09/2005

[freescale.com](http://freescale.com)



# TABLE OF CONTENTS

## About This Document

Audience .....	vii
Organization .....	vii
Suggested Reading .....	vii
Conventions .....	viii
Definitions, Acronyms, and Abbreviations .....	ix
References.....	ix

## Chapter 1

### 56F8xx Serial Bootloader

1.1.1	Start-up Sequence with Bootloader .....	1-2
1.1.2	Bootloader version number.....	1-3
1.1.3	Files.....	1-3
1.1.4	Bootloader Hardware Set-up .....	1-4
1.1.5	Target Consideration.....	1-6
1.1.6	Build .....	1-8
1.1.7	Execute .....	1-9
1.1.8	Requirements for a Loaded Program .....	1-11
1.1.9	S-Record Generation.....	1-12
1.1.10	Peripheral Usage.....	1-12

## Chapter 2

### License

2.1	Limited Use License Agreement .....	2-1
-----	-------------------------------------	-----



# LIST OF TABLES

1-1	Error Codes for the Serial Bootloader Application .....	1-10
-----	---	------



# LIST OF FIGURES

1-1	Programming Flash on a 56F8xx Device .....	1-1
1-2	DSP56F80x Boot Sequence with Bootloader .....	1-2
1-3	Loading a User's Application via Serial Bootloader .....	1-4
1-4	56F827 Boot Memory .....	1-7
1-5	Programming the 56F827 Bootloader .....	1-8
1-6	S-Record Settings in the User Application's Project Settings .....	1-12





## About This Document

This manual describes the 56F8xx Serial Bootloaders application.

## Audience

This manual targets software developers utilizing the DSP568xx Bootloaders applications.

## Organization

This User's Manual consists of the following sections:

- [Chapter 1, 56F8xx Serial Bootloader](#) -- describes the Serial Bootloader, applications provided with the FlexCAN driver.

## Suggested Reading

We recommend that you have a copy of the following references:

- *DSP56800 Family Manual*, DSP56800FM
- *56F800 User Manual*, DSP56F801-7UM
- *Inside CodeWarrior: Core Tools*, Metrowerks Corp.

# Conventions

This document uses the following notational conventions:

Typeface, Symbol or Term	Meaning	Examples
Courier Monospaced Type	Code examples	<code>//Process command for line flash</code>
<i>Italic</i>	Directory names, project names, calls, functions, statements, procedures, routines, arguments, file names, applications, variables, directives, code snippets in text	...and contains these core directories: <i>applications</i> contains applications software... ...CodeWarrior project, <i>3des.mcp</i> is... ...the <i>pConfig</i> argument... ...defined in the C header file, <i>aec.h</i> ...
<b>Bold</b>	Reference sources, paths, emphasis	...refer to the <b>Targeting DSP56F80x Platform</b> manual.... ...see: <b>C:\Program Files\help\tutorials</b>
Blue Text	Linkable on-line	...refer to <a href="#">Chapter 7</a> , License....
Number	Any number is considered a positive value, unless preceded by a minus symbol to signify a negative value	3V -10 DES <sup>-1</sup>
ALL CAPITAL LETTERS	# defines/ defined constants	# define INCLUDE_STACK_CHECK
Brackets [...]	Function keys	...by pressing function key [F7]
Quotation marks, "..."	Returned messages	...the message, "Test Passed" is displayed.... ...if unsuccessful for any reason, it will return "NULL"...

## Definitions, Acronyms, and Abbreviations

The following list defines the acronyms and abbreviations used in this document. As this template develops, this list will be generated from the document. As we develop more group resources, these acronyms will be easily defined from a common acronym dictionary. Please note that while the acronyms are in solid caps, terms in the definition should be initial capped ONLY IF they are trademarked names or proper nouns.

<b>EVM</b>	Evaluation module
<b>PC</b>	Personal Computer
<b>PLL</b>	Phase Locked Loop
<b>RAM</b>	Random Access (read/write) Memory
<b>SCI</b>	Serial Communications Interface

## References

The following sources were used to produce this book:

- *DSP56800 Family Manual*, DSP56800FM
- *56F800 User Manual*, DSP56F801-7UM



# Chapter 1

## 56F8xx Serial Bootloader

### 1.1 Serial Bootloader

The Serial Bootloader has been developed to load and run a proprietary user application presented as an S-Record file into the Program and Data memory. The Serial Bootloader is located in the dedicated Program Memory region, called Boot Flash (See [Section 1.1.5](#)). The Serial Bootloader supports the simplest serial protocol, so any standard serial terminal program can be used on the host PC.

The Serial Bootloader application reads the S-Record file of a user application (for example, generated by CodeWarrior) via serial interface, parses this S-Record file, and stores needed data in Program and Data Flash memory. When the processing of the S-Record file is finished, the Bootloader launches the loaded application. If any error occurs while loading the S-Record file, the Bootloader outputs an error message with an error number via the serial line and resets the processor. See [Figure 1-1](#).

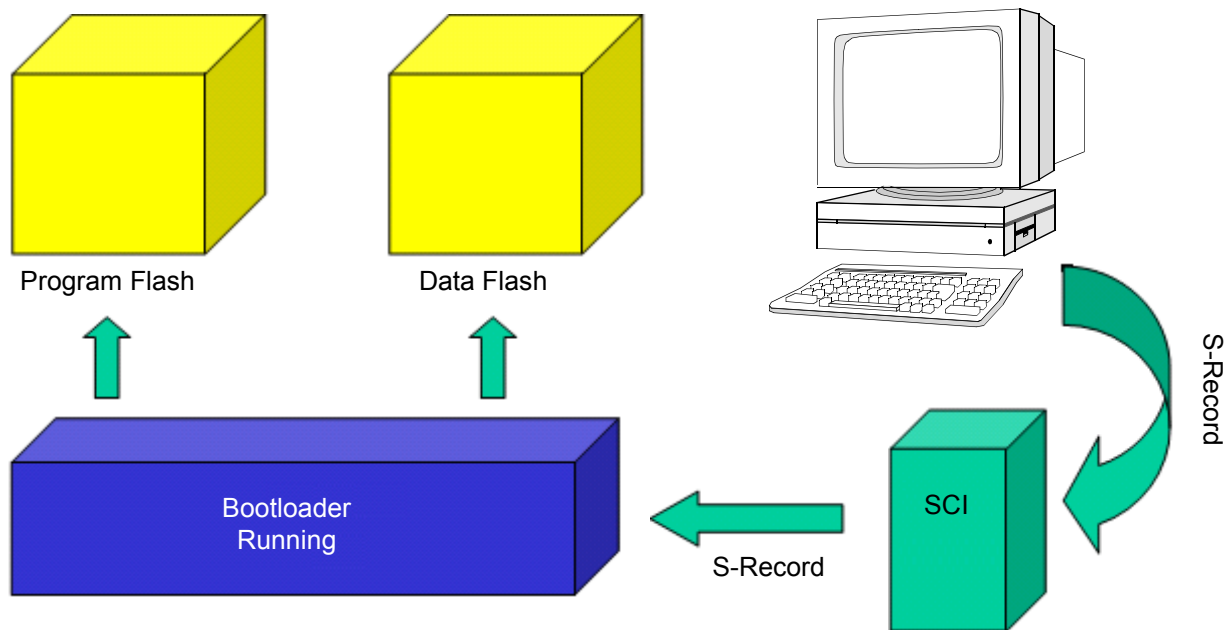


Figure 1-1. Programming Flash on a 56F8xx Device

### 1.1.1 Start-up Sequence with Bootloader

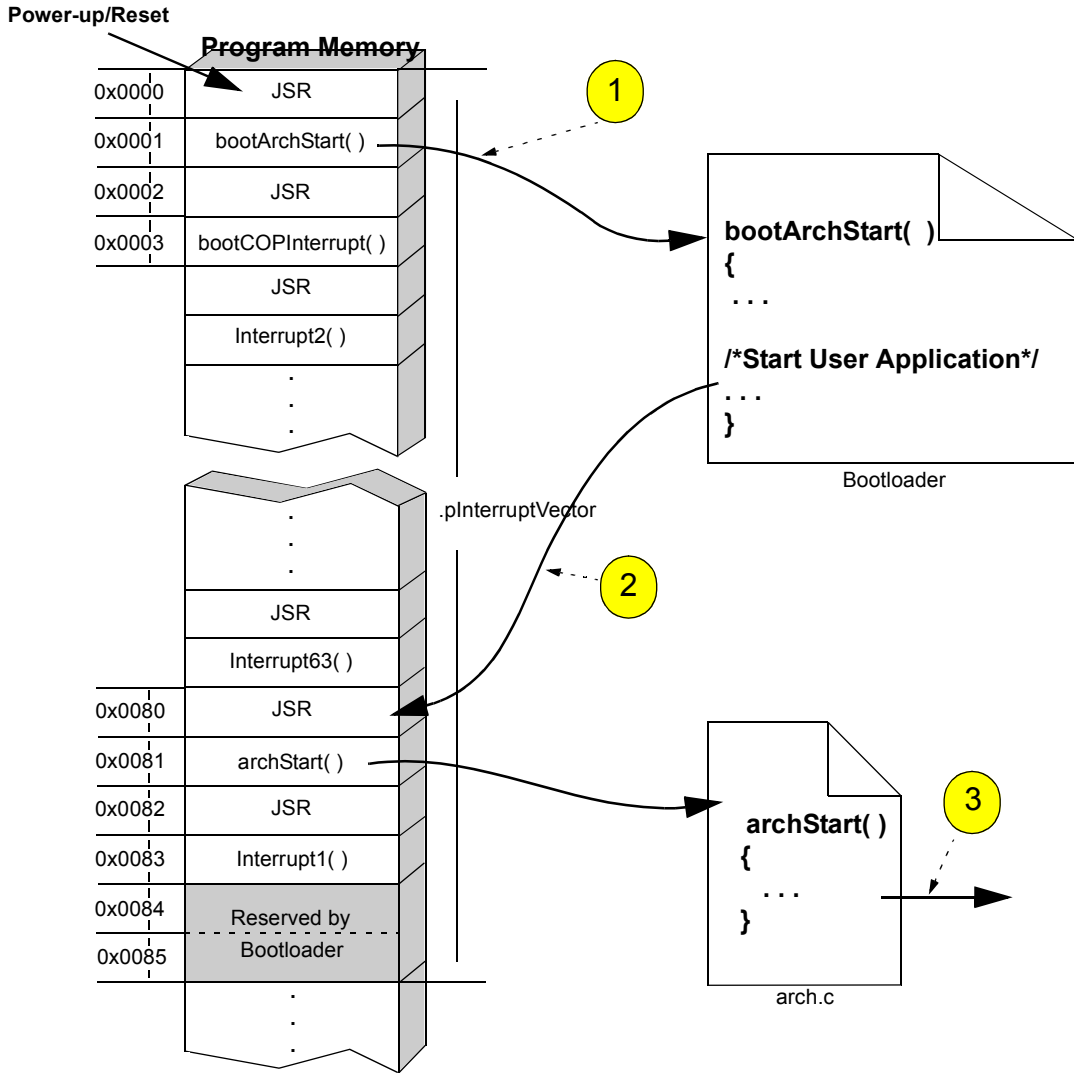


Figure 1-2. DSP56F80x Boot Sequence with Bootloader

### 1.1.1.1 Step 1: Power-up/Reset

The DSP56800 core specifies two reset vectors: Hardware Reset and COP Reset (normally located at addresses 0x0 and 0x2, respectively). The Hardware Reset vector identifies the address that the processor accesses once it recognizes a power-up or power reset. When the DSP56F80x bootloader is present, it controls this address by placing a bootloader entry point into the Hardware Reset vector location.

### 1.1.1.2 Step 2: Transition to Application's Entry Point

When the DSP56F80x bootloader completes its execution, it transfers control to the user's application by performing a *JSR* instruction to an address specified in memory location 0x0081. The address of this routine is placed into the 0x0081 location via the Interrupt Vector table located in the *vector.c* configuration file.

### 1.1.1.3 Step 3: Start-up Code

Execution of *archStart()* begins.

## 1.1.2 Bootloader version number

The most current version of the bootloader is 1.3.

## 1.1.3 Files

The Serial Bootloader application includes the following files:

- `..\applications\serial_bootloader\bootloader.mcp`, project file
- `..\applications\serial_bootloader\bootloader.c`, main program
- `..\applications\serial_bootloader\bootloader.h`, header file with common parameters
- `..\applications\serial_bootloader\com.c`, communication module
- `..\applications\serial_bootloader\com.h`, header for communication module
- `..\applications\serial_bootloader\sparser.c`, S-Record format parser module
- `..\applications\serial_bootloader\sparser.h`, header for S-Record parser module
- `..\applications\serial_bootloader\prog.c`, flash programming module
- `..\applications\serial_bootloader\prog.h`, header for flash programming module
- `..\applications\serial_bootloader\bootstart.c`, startup module
- `..\applications\serial_bootloader\constdata.asm`, description of strings data
- `..\applications\serial_bootloader\resetvector.asm`, Reset and COP Reset interrupt vectors description
- `..\applications\serial_bootloader\flashprog.c`, flash programming subsystem
- `..\applications\serial_bootloader\flashprog.h`, flash programming subsystem
- `..\applications\serial_bootloader\TargetDirectives`, Board name definition
- `..\applications\serial_bootloader\config\linker.cmd`, linker command file used for Boot Flash
- `..\applications\serial_bootloader\config\flash.cfg`, Metrowerks CodeWarrior configuration file to work with Flash

- `..\config\`, configuration files
- `..\include\`, include files

The Bootloader application also uses the files `port.h`, `arch.h` and `periph.h`, found in the `\include` directory.

**Note:** The Bootloader application relies on the `TargetDirectives` and `linker.cmd` files to define the target for which the application will be built.

## 1.1.4 Bootloader Hardware Set-up

To use the Bootloader, these requirements must be met:

- It must be programmed into Boot Flash (See [Section 1.1.5](#))
- The RS-232 socket must be connected by serial cable with the host PC's COM serial port (See [Figure 1-3](#))
- The jumpers on the EVM board must be set to the use of internal memory without debug interface

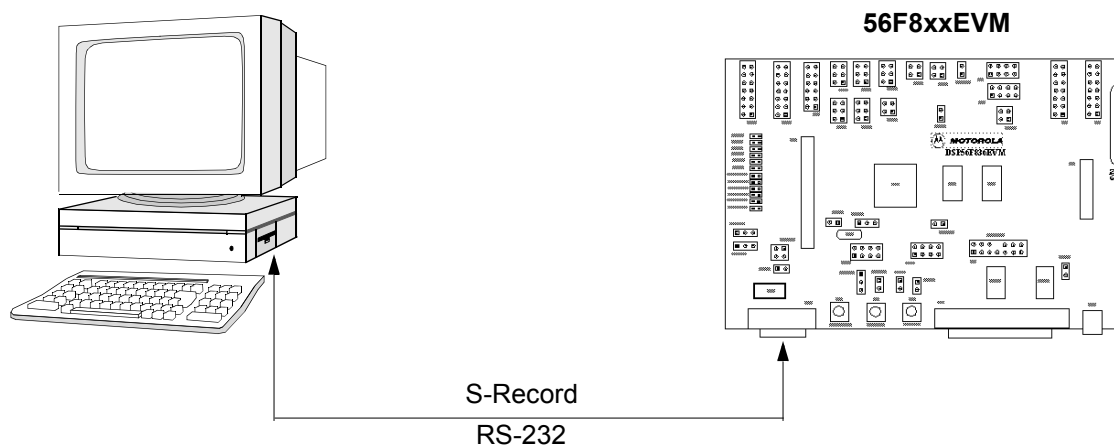


Figure 1-3. Loading a User's Application via Serial Bootloader

### 1.1.4.1 EVM Jumper Settings

#### 1.1.4.1.1 56F801

To load the Bootloader into the **56F801** board, the following jumper settings are required:

- Do not set jumper JG5, "Enable on-board Parallel JTAG Host Target Interface"

To start a previously loaded Bootloader on the **56F801** board, the following jumper settings are needed:

- Set jumper JG5, "Enable on-board Parallel JTAG Host Target Interface"

#### 1.1.4.1.2 56F803

To load the Bootloader into the **56F803** board, the following jumper settings are required:

- Remove jumper JG2, "Enable on-board Parallel JTAG Host Target Interface"
- Set jumper JG4, "Select DSP's Mode 0 operation upon exit from reset"



To start a previously loaded Bootloader on the **56F803** board, the following jumper settings are required:

- Set jumper JG2, “Enable on-board Parallel JTAG Host Target Interface”
- Set jumper JG4, “Select DSP’s Mode 0 operation upon exit from reset”

#### **1.1.4.1.3 56F805**

To load the Bootloader into the **56F805** board, the following jumper settings are required:

- Set jumper JG7, “Select DSP’s Mode 0 operation upon exit from reset”
- Set jumper JG9, “Enable RS-232 output”
- Remove jumper JG5, “Enable on-board Parallel JTAG Host Target Interface”

To start a previously loaded Bootloader on the **56F805** board, the following jumper settings are needed:

- Set jumper JG5, “Enable on-board Parallel JTAG Host Target Interface”
- Set jumper JG7, “Select DSP’s Mode 0 operation upon exit from reset”
- Set jumper JG9, “Enable RS-232 output”

#### **1.1.4.1.4 56F807**

To load the Bootloader into the **56F807** board, the following jumper settings are needed:

- Set jumper JG7, “Select DSP’s Mode 0 operation upon exit from reset”
- Leave jumper JG9, “Enable RS-232 output”, in its default setting (NC)
- Remove jumper JG4, “Enable on-board Parallel JTAG Host Target Interface”

To start a previously loaded Bootloader on the **56F807** board, the following jumper settings are needed:

- Set jumper JG4, “Enable on-board Parallel JTAG Host Target Interface”
- Set jumper JG7, “Select DSP’s Mode 0 operation upon exit from reset”
- Leave jumper JG9, “Enable RS-232 output”, in its default setting (NC)

#### **1.1.4.1.5 56F826**

To load the Bootloader into the **56F826** board, the following jumper settings are needed:

- Set jumper JG6, “Select DSP’s Mode 0 operation upon exit from reset”
- Remove jumper JG2, “Enable RS-232 output”
- Remove jumper JG1, “Enable on-board Parallel JTAG Host Target Interface”

To start a previously loaded Bootloader on the **56F826** board, the following jumper settings are needed:

- Set jumper JG1, “Enable on-board Parallel JTAG Host Target Interface”
- Set jumper JG6, “Select DSP’s Mode 0 operation upon exit from reset”
- Remove jumper JG7, “Enable RS-232 output”

#### **1.1.4.1.6 56F827**

To load the Bootloader into the **56F827** board, the following jumper settings are needed:

- Set jumper JG6, “Select DSP’s Mode 0 operation upon exit from reset”
- Remove jumper JG2, “Enable RS-232 output”
- Remove jumper JG1, “Enable on-board Parallel JTAG Host Target Interface”

To start a previously loaded Bootloader on the **56F827** board, the following jumper settings are needed:

- Set jumper JG1, “Enable on-board Parallel JTAG Host Target Interface”
- Set jumper JG6, “Select DSP’s Mode 0 operation upon exit from reset”
- Remove jumper JG2, “Enable RS-232 output”

## 1.1.5 Target Consideration

### 1.1.5.1 56F801

When the *TargetDirectives* file contains **#define DSP56801EVM**, this application is built assuming the existence of an internal relaxation oscillator and corresponding calibration value. At run-time, the application automatically calibrates the internal relaxation oscillator to run at 8MHz.

**Note:** For the **56F801**, only processors with a date code **GREATER THAN 0152** marked in the lower right corner of the part contain the internal oscillator calibration value. For parts built prior to this date, the *TargetDirectives* file should contain the following for correct operation:

```
#define DSP56801EVM
#define PLL_USE_EXTERNAL_OSC
```

**Note:** The 56F802 does **NOT** have an external oscillator and must therefore have the following in the *TargetDirectives* file for correct operation:

```
#define DSP56801EVM
```

### 1.1.5.2 56F827

#### 1.1.5.2.1 56F827 Boot Flash

The 56F827 has 63KB of on-chip Program Flash memory and 1KB of on-chip Program RAM. Unlike the 56F826, which has dedicated Boot Flash hardware, the 56F827 has no dedicated Boot Flash hardware. As shown in [Figure 1-4](#), Boot Flash is carved from Program Flash 2 and Program Flash 1. The Bootloader is mapped to Program Flash 2 at location 0xF800 to 0xFFFF.

#### 1.1.5.2.2 56F827 Hardware Reset and COP

The hardware and COP reset vectors are mapped to Program Flash 1 at locations 0x0000 and 0x0002. Unlike the 56F826, which has dedicated Boot Flash hardware, the 56F827 hardware and COP reset vectors can be erased when programming the first page in Program Flash 1 through the Bootloader. This can occur if power to the 56F827 is lost when the Bootloader is saving and erasing the page that contains the reset and COP vectors. If this happens, the Bootloader will not be available on the next 56F827 reset.

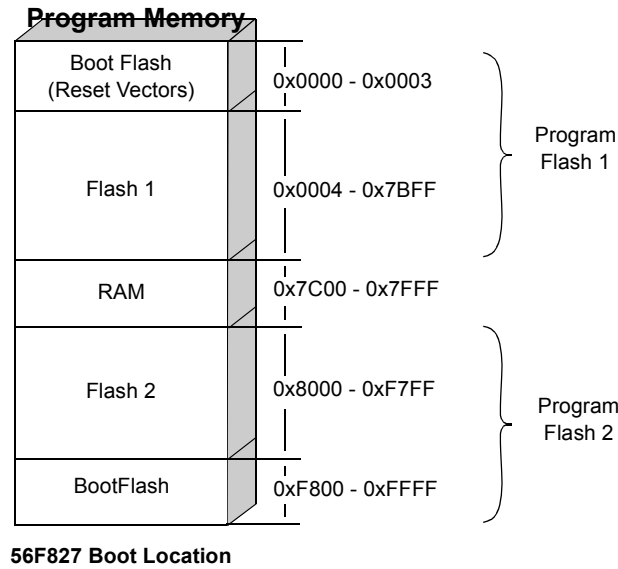


Figure 1-4. 56F827 Boot Memory

### 1.1.5.2.3 56F827 Bootloader Operation

The Bootloader program code which permits programming both Program Flash1 and Program Flash 2 resides in a Boot Flash section which is part of Program Flash 2. In order to write or erase part of Program Flash 2, the Bootloader functions that permit programming Flash 2 are relocated at run time and are executed from Program RAM. This allows the ability to program both Program Flash 1 and Program Flash 2 with user application code. Every time the 56F827 power is recycled or reset, the Serial Bootloader resident in the Boot Flash will re-copy Flash programming functions from the Boot Section to Program RAM. [Figure 1-5](#) demonstrates how the Bootloader program code (which is configured for internal memory operation) will be mapped into physical memory on the 56F827 target using CodeWarrior tools.

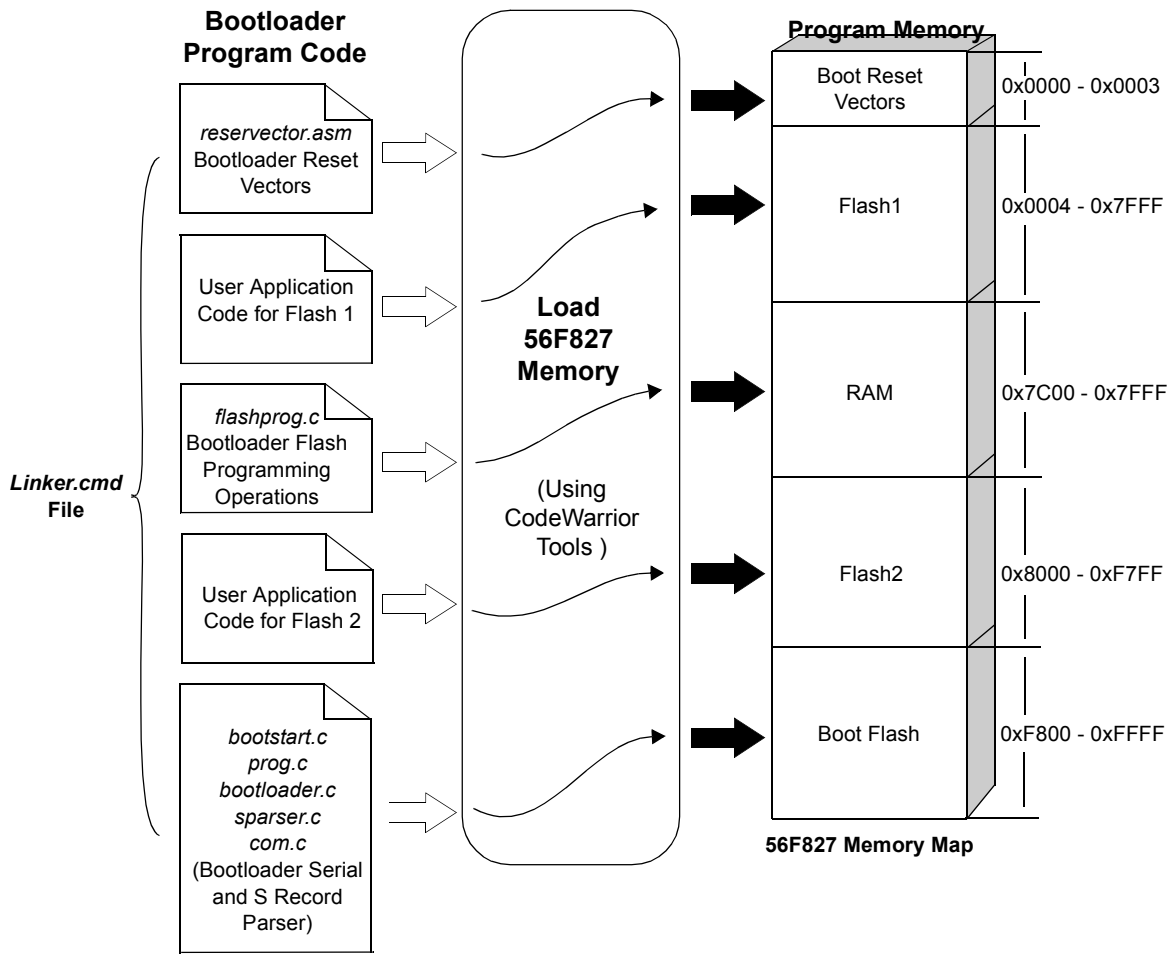


Figure 1-5. Programming the 56F827 Bootloader

## 1.1.6 Build

To build the Serial Bootloader, open the `bootloader.mcp` project file in the CodeWarrior IDE and execute the `Project/Make` command. This will build and link the Serial Bootloader application.

### 1.1.6.1 Download into Boot Flash

To download, build the Bootloader from Codewarrior and load it into the board by choosing the `Project/Debug` command in the CodeWarrior IDE. Make sure that jumpers are set for loading as described in [Section 1.1.4.1](#).

## 1.1.6.2 Host Terminal Program Set-up

A host terminal program is used to communicate with Bootloader. The terminal must be configured to the following mode:

Baud rate	115200 bps
8N1	8 data bits, no parity, 1 stop bit character format
Flow control protocol	Xon / Xoff

The following description assumes that Microsoft Windows HyperTerminal program is used. To configure Microsoft HyperTerminal to communicate with Bootloader, the PC user should start HyperTerminal and create a new connection. Select the COM port connected to the EVM board and set:

Baud rate	115200 bps
8N1	8 data bits, no parity, 1 stop bit character format
Flow control protocol	Xon / Xoff

The HyperTerminal can now display Bootloader messages.

## 1.1.7 Execute

To execute the Serial Bootloader application after loading it into Flash, set jumpers as described in [Section 1.1.4.1](#) and push the RESET button.

If the terminal program is properly set up and the EVM and the Host PC are properly connected, the terminal program will display a Bootloader start-up message:

```
"(c) 2000-2001 Motorola Inc. S-Record loader. Version 1.1"
```

To load the S-Record file, select the *Transfer/Send text file* from the HyperTerminal menu and select a file. When the S-Record file is loaded and the application is started, the terminal windows displays a message similar to this:

```
"(c) 2000-2001 Motorola Inc. S-Record loader. Version 1.1
```

```
Loaded 0x044d Program and 0x000a Data words.  
Application started."
```

The download rate is about 7660 bytes of S-Record file per second loaded from the terminal or about 1740 words of program or data memory stored into Flash per second.

If an error is detected while loading the S-Record file, the Bootloader displays the error message and resets the processor. For example, if an S-record file contains a character that is not permitted for S-Records, the following message is displayed:

```
"(c) 2000-2001 Motorola Inc. S-Record loader. Version 1.1
```

```
Error # 0002  
Resetting the processor."
```

After this message, the Bootloader resets the processor and waits for the S-Record file again. Other loading errors are described in [Table 1-1](#).

**Table 1-1. Error Codes for the Serial Bootloader Application**

Error Code	Error Title	Possible Reasons	What to Do
1	Data Receive Error	•Any SCI error except Noise Error (Overrun, Frame Error, Parity Error)	•Check connections with Host PC and settings for host terminal program
2	Invalid Character	•Received character is not "S" or any hexadecimal digit	•Verify that S-Record file does not contain any invalid characters •Check connections and send mode in the terminal program
3	Invalid S-Record Format	•Invalid record type; permitted type is 0,3,7 •S-Record length is less than address plus checksum length	•Verify S-Record file
4	Wrong S-Record Checksum	•Checksum calculated around received S-Record did not match with received one.	•Check S-Record file •Check connections and send mode in terminal program
5	Buffer Overrun	•Internal data buffer was full	•Terminal program did not stop after receiving Xoff character •Confirm that the terminal program supports Xon / Xoff flow control protocol
6	Flash Programming Error	•After programming word into Flash, the programmed word read back is not equal to expected value	•The Bootloader tries to program Flash only once and performs a read back / verify of the value
7	Internal Error	•Bootloader data corrupt	•Try to reload Bootloader via CodeWarrior
8	Parity Error	•Bootloader data corrupt	•Try to reload application with Bootloader
9	Protected Boot Flash Error	•TS-Record you are trying to program may contain addresses to Boot Flash	•Look at S-record to see if it contains addresses to Boot Flash

If an application previously loaded via the Bootloader uses the variable `BSP_BOOTLOADER_DELAY`, (see [Section 1.1.8](#)), the Bootloader waits for the S-Record file only until the required time-out expires, then launches the application. When this happens, the terminal window contains a message similar to this:

```
"(c) 2000-2001 Motorola Inc. S-Record loader. Version 1.1
Application started."
```

## 1.1.8 Requirements for a Loaded Program

If the application is loaded via the Bootloader, it must meet the following requirements:

- **Particular start address for application** - The entry point for the loaded application must be located at address 0x0080 in Program memory; i.e., immediately after the interrupt table
- **Application COP vector** - To use COP interrupt vector in an application loaded via Bootloader, the entry point for the COP ISR must be located at address 0x0082 in Program memory
- **Application start delay variable must be set at address 0x0085 in Program Flash**
- **Restricted resources use** - The application cannot occupy Reset and COP interrupt vectors, and cannot place code into Boot Flash memory area. There is no way to place any initialized variable from the application into internal data RAM while loading. This memory area is used by Bootloader. All data from the S-Record file that address to the restricted area will be ignored. External Program memory is unavailable while loading the application because the Bootloader is performed in DSP Mode 0A memory map.

The Bootloader provides a configuration variable (`BSP_BOOTLOADER_DELAY`) can be defined in the Bootloader's `appconfig.h` file that determines the application's start delay time-out.

Possible values of `BSP_BOOTLOADER_DELAY`:

- |                |   |
|----------------|---|
| <b>0</b>       | <p>Disable the Bootloader, start the application immediately. After loading the application with this setting, there are only two ways to re-enter Bootloader:</p> <ul style="list-style-type: none"> <li>• Reload the Bootloader from Metrowerks CodeWarrior into Boot Flash</li> <li>• Reprogram the value of the start delay variable located at address 0x0083 in the Program Flash to zero value (0x0000) in the loaded application</li> </ul> |
| <b>0 - 254</b> | <p>Set waiting time for the start of the S-Record file for 0-255 seconds before the start of a previously-loaded application</p>  |
| <b>255</b>     | <p>Set waiting time to infinity. After reset, the Bootloader waits for the S-Record file without counting down any time-out</p>   |

If there is no `BSP_BOOTLOADER_DELAY` set in the `appconfig.h` file, the default value for time-out is 30 seconds.

## 1.1.9 S-Record Generation

Figure 1-6 illustrates how a user’s application within Code Warrior can be set up to generate an S-Record needed by the Bootloaders to program the user’s application into the 56F8xx device’s Flash.

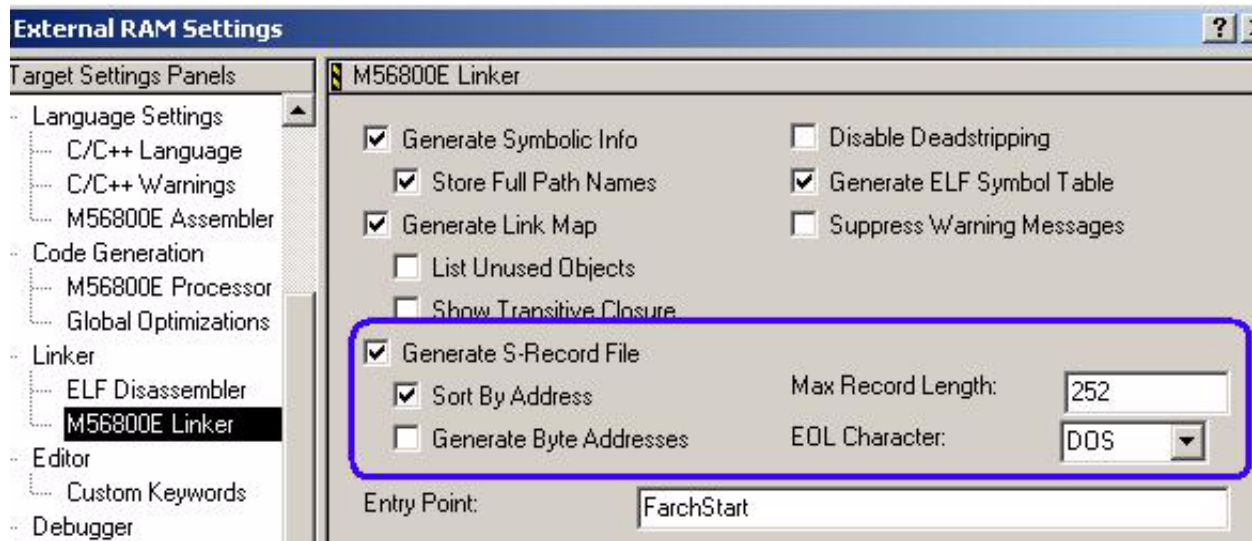


Figure 1-6. S-Record Settings in the User Application’s Project Settings

“Max Record Length” and “EOL Character” are set to the default values shown in Figure 1-6. “Max Record Length” can be modified to up to 252. “EOL Character” can be changed to “Mac” or “UNIX” with no effect.

### 1.1.10 Peripheral Usage

If the hybrid controller is used in a single chip mode, the Bootloader uses only internal Data RAM for data buffering.

The Bootloader does not initialize any device peripheral except SCI0, Port E and PLL. The PLL multiplier is set to 18, which equals the hybrid controller’s operational frequency of 72MHz. Before starting the application, SCI0 is disabled, but the PLL is not reprogrammed to its initial state.

The Bootloader uses a statically-calculated SCI baud rate value. This value was calculated with the assumption that the external Oscillator Frequency is 8MHz.



# Chapter 2

## License

### 2.1 Limited Use License Agreement

#### LIMITED USE LICENSE AGREEMENT

PLEASE READ THIS AGREEMENT CAREFULLY BEFORE USING THIS SOFTWARE. BY USING OR COPYING THE SOFTWARE, YOU AGREE TO THE TERMS OF THIS AGREEMENT.

The software in either source code form ("Source") or object code form ("Object") (cumulatively hereinafter "Software") is provided under a license agreement ("Agreement") as described herein. Any use of the Software including copying, modifying, or installing the Software so that it is usable by or accessible by a central processing unit constitutes acceptance of the terms of the Agreement by the person or persons making such use or, if employed, the employer thereof ("Licensee") and if employed, the person(s) making such use hereby warrants that they have the authority of their employer to enter this license agreement. If Licensee does not agree with and accept the terms of this Agreement, Licensee must return or destroy any media containing the Software or materials related thereto, and destroy all copies of the Software.

The Software is licensed to Licensee by Freescale Semiconductor Inc. ("Freescale") for use under the terms of this Agreement. Freescale retains ownership of the Software. Freescale grants only the rights specifically granted in this Agreement and grants no other rights. Title to the Software, all copies thereof and all rights therein, including all rights in any intellectual property including patents, copyrights, and trade secrets applicable thereto, shall remain vested in Freescale.

For the Source, Freescale grants Licensee a personal, non-exclusive, non-assignable, revocable, royalty-free right to use, copy, and make derivatives of the Source solely in a development system environment in order to produce object code solely for operating on a Freescale semiconductor device having a central processing unit ("Derivative Object").

For the Object and Derivative Object, Freescale grants Licensee a personal, non-exclusive, non-assignable, revocable, royalty-free right to copy, use, and distribute the Object and the Derivative Object solely for operating on a Freescale semiconductor device having a central processing unit.

Licensee agrees to: (a) not use, modify, or copy the Software except as expressly provided herein, (b) not distribute, disclose, transfer, sell, assign, rent, lease, or otherwise make available the

Software, any derivatives thereof, or this license to a third party except as expressly provided herein, (c) not remove, obliterate, or otherwise defeat any copyright, trademark, patent or proprietary notices, related to the Software (d) not in any form export, re-export, resell, ship or divert or cause to be exported, re-exported, resold, shipped, or diverted, directly or indirectly, the Software or a direct product thereof to any country which the United States government or any agency thereof at the time of export or re-export requires an export license or other government approval without first obtaining such license or approval.

THE SOFTWARE IS PROVIDED ON AN "AS IS" BASIS AND WITHOUT WARRANTY OF ANY KIND INCLUDING (WITHOUT LIMITATION) ANY WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL FREESCALE BE LIABLE FOR ANY LIABILITY OR DAMAGES OF ANY KIND INCLUDING, WITHOUT LIMITATION, DIRECT OR INDIRECT OR INCIDENTAL OR CONSEQUENTIAL OR PUNITIVE DAMAGES OR LOST PROFITS OR LOSS OF USE ARISING FROM USE OF THE SOFTWARE OR THE PRODUCT REGARDLESS OF THE FORM OF ACTION OR THEORY OF LIABILITY (INCLUDING WITHOUT LIMITATION, ACTION IN CONTRACT, NEGLIGENCE, OR PRODUCT LIABILITY) EVEN IF FREESCALE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. THIS DISCLAIMER OF WARRANTY EXTENDS TO LICENSEE OR USERS OF PRODUCTS AND IS IN LIEU OF ALL WARRANTIES WHETHER EXPRESS, IMPLIED, OR STATUTORY, INCLUDING IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR PARTICULAR PURPOSE.

Freescale does not represent or warrant that the Software is free of infringement of any third party patents, copyrights, trade secrets, or other intellectual property rights or that Freescale has the right to grant the licenses contained herein. Freescale does not represent or warrant that the Software is free of defect, or that it meets any particular requirements or need of the Licensee, or that it conforms to any documentation, or that it meets any standards.

Freescale shall not be responsible to maintain the Software, provide upgrades to the Software, or provide any field service of the Software. Freescale reserves the right to make changes to the Software without further notice to Licensee.

The Software is not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Software could create a situation where personal injury or death may occur. Should Licensee purchase or use the Software for any such unintended or unauthorized application, Licensee shall indemnify and hold Freescale and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if

such claim alleges that Freescale was negligent regarding the design or manufacture of the Software.

The term of this Agreement is for as long as Licensee uses the Software for its intended purpose and is not in default of any provisions of this Agreement. Freescale may terminate this Agreement if Licensee is in default of any of the terms and conditions of this Agreement.

This Agreement shall be governed by and construed in accordance with the laws of the State of Arizona and can only be modified in a writing signed by both parties. Licensee agrees to jurisdiction and venue in the State of Arizona.

By using, modifying, installing, compiling, or copying the Software, Licensee acknowledges that this Agreement has been read and understood and agrees to be bound by its terms and conditions. Licensee agrees that this Agreement is the complete and exclusive statement of the agreement between Licensee and Freescale and supersedes any earlier proposal or prior arrangement, whether oral or written, and any other communications relative to the subject matter of this Agreement.



# INDEX

## D

DSP56800 Family Manual vii, ix  
DSP56F800 User Manual vii, ix

## E

Evaluation Module  
    EVM ix  
EVM ix

## F

FlexCAN ix  
Flexible Controller Area Network  
    FlexCAN ix

## I

Inside CodeWarrior  
    Core Tools vii

## L

linker.cmd file 1-4

## P

PC ix  
Personal Computer  
    PC ix  
Phase Locked Loop  
    PLL ix  
PLL ix, 1-12  
Port E 1-12

## R

RAM ix  
Random Address Memory  
    RAM ix

## S

SCI ix  
SCI0 1-12  
Serial Communications Interface  
    SCI ix  
S-Record file 1-1

## T

Target Directives file 1-4, 1-6



## **How to Reach Us:**

### **Home Page:**

[www.freescale.com](http://www.freescale.com)

### **E-mail:**

[support@freescale.com](mailto:support@freescale.com)

### **USA/Europe or Locations Not Listed:**

Freescale Semiconductor  
Technical Information Center, CH370  
1300 N. Alma School Road  
Chandler, Arizona 85224  
+1-800-521-6274 or +1-480-768-2130  
[support@freescale.com](mailto:support@freescale.com)

### **Europe, Middle East, and Africa:**

Freescale Halbleiter Deutschland GmbH  
Technical Information Center  
Schatzbogen 7  
81829 Muenchen, Germany  
+44 1296 380 456 (English)  
+46 8 52200080 (English)  
+49 89 92103 559 (German)  
+33 1 69 35 48 48 (French)  
[support@freescale.com](mailto:support@freescale.com)

### **Japan:**

Freescale Semiconductor Japan Ltd.  
Headquarters  
ARCO Tower 15F  
1-8-1, Shimo-Meguro, Meguro-ku,  
Tokyo 153-0064, Japan  
0120 191014 or +81 3 5437 9125  
[support.japan@freescale.com](mailto:support.japan@freescale.com)

### **Asia/Pacific:**

Freescale Semiconductor Hong Kong Ltd.  
Technical Information Center  
2 Dai King Street  
Tai Po Industrial Estate  
Tai Po, N.T., Hong Kong  
+800 2666 8080  
[support.asia@freescale.com](mailto:support.asia@freescale.com)

### **For Literature Requests Only:**

Freescale Semiconductor Literature Distribution Center  
P.O. Box 5405  
Denver, Colorado 80217  
1-800-441-2447 or 303-675-2140  
Fax: 303-675-2150  
[LDCForFreescaleSemiconductor@hibbertgroup.com](mailto:LDCForFreescaleSemiconductor@hibbertgroup.com)

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals", must be validated for each customer application by customer's technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.



Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. This product incorporates SuperFlash® technology licensed from SST.

© Freescale Semiconductor, Inc. 2005. All rights reserved.