

Introduction

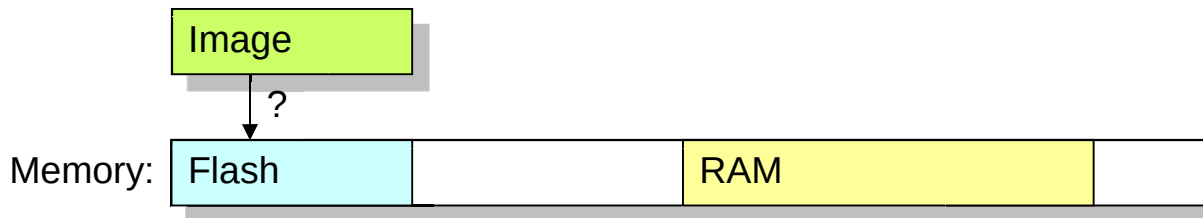
Many development boards use flash memory as the primary code memory. Normally, flash memory cannot be written directly from C-SPY when a program is to be downloaded and debugged, but must instead be *flashed* or *programmed* by a dedicated program—the *flash loader*—that executes on the target system.

Note: The flash loader mechanism is useful for any sort of memory that cannot be directly written to by the debugger, but instead has to be written to by a dedicated program that executes on the target system. Although, the flash loader is mostly used for flash memory, it can for example also be used for various forms of external RAM or disk-like storage devices. However, in the rest of this guide, flash memory is the assumed memory.

The IAR flash loader framework is available as open source under an Apache license. The framework package can be downloaded from http://links.iar.com/cmsis/IAR_flashloader_framework_200.zip

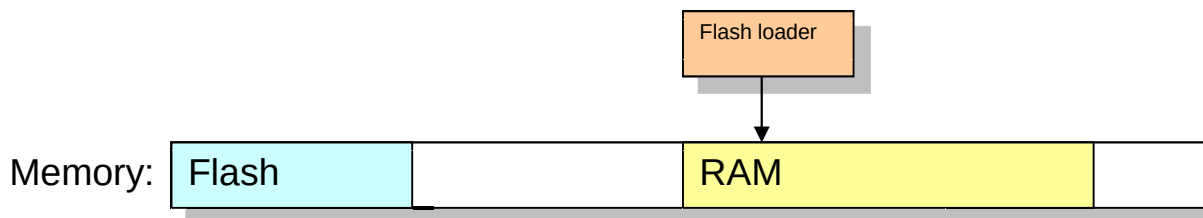
The process in brief

This is the problem: you need to download an application image to flash memory, but C-SPY can only download data directly to RAM.



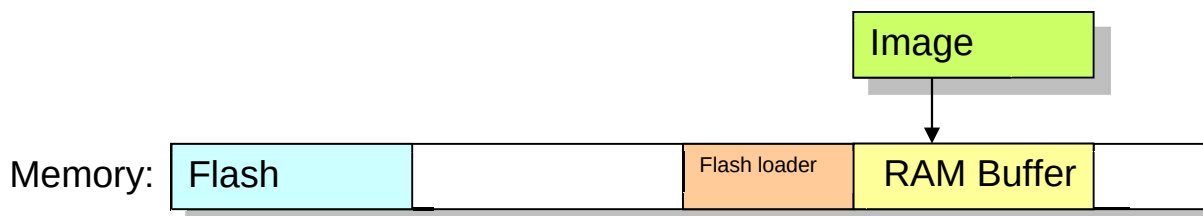
To solve the problem, this procedure is performed:

- 1 Download a dedicated flash loader into RAM.

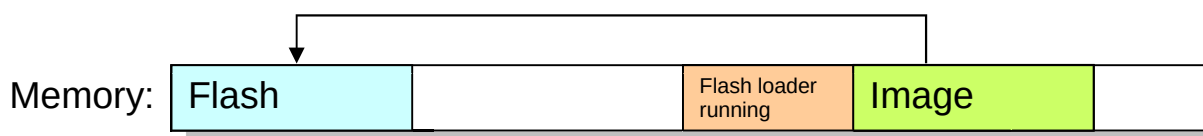


Part of RAM is also reserved for a download buffer.

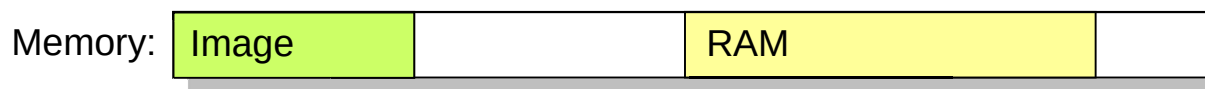
- 2 The image is written to the RAM buffer.



- 3 The flash loader is started by C-SPY. The flash loader reads data from the RAM buffer and programs the flash memory.



- 4 The image now resides in flash memory and can be started. The flash loader and the RAM buffer are no longer needed, so RAM is fully available to the application in the flash memory.



In practice, the process is a bit more complicated. For example, the RAM buffer is usually much smaller than the image to be downloaded, so the flash programming must be performed in several steps.

Flash loading components

When using flash loading, there are two major components involved:

- The flash loader program (in short, *the flash loader*)
- The flash loader configuration file
- The flash memory *system* configuration file

The flash loader

A flash loader is usually a rather small program which can program one or more flash memories. The flash loader consists of a small set of functions, mainly for erasing or writing designated portions of the flash memory. C-SPY downloads this program into RAM (it must be linked to an address in RAM). To run the program, C-SPY sets the PC to one of the functions in the flash loader, writes data and directives for that function into a RAM buffer, and starts execution. When the function returns, execution will hit a breakpoint. C-SPY will then know that the function has finished and can proceed to make further calls to continue the flash loading process. In short, C-SPY calls functions in the flash loader.

The flash memory configuration file

The flash memory configuration file is an XML file (filename extension `flash`) which describes for C-SPY all relevant properties of a certain flash memory. For example, the base address of the flash memory and details such as block and page sizes. The file also specifies which flash loader to use.

The flash memory system configuration file

The flash memory system specification file is an XML file (filename extension `board`) which describes for C-SPY the flash loading properties of the complete development board. This file can sometimes contain references to more than one flash memory (through the appropriate `.flash` files), if the board has more than one type of flash memory that needs to be programmed separately in several *passes*. In this case, the `.board` file also specifies specific address ranges of the image which belongs to different flash memories.

One `.board` file specifies the information needed to perform flash loading for a specific board using IAR Embedded Workbench. Such a file can be prepared in advance for various development boards, and can also be created or modified in the **Project>Options>Debugger>Download** dialog box in the IAR Embedded Workbench IDE.

If a board contains two flash memories, where one is used only for a boot loader and the other for the application, only one of them would be relevant for any given project. There would then be two different `.board` files, one for each kind of project.