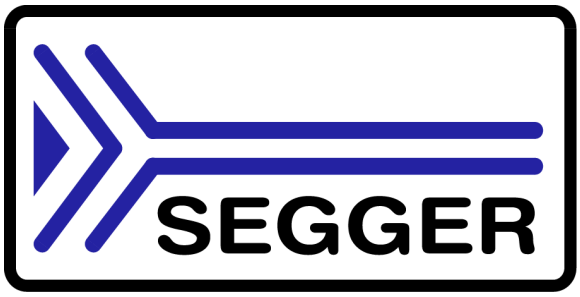
J-Flash ARM

User guide of the stand-alone flash programming software

Software Version 4.10
Manual Rev. 3

Date: September 18, 2009

Document: UM08003



A product of SEGGER Microcontroller GmbH & Co. KG

www.segger.com

Disclaimer

Specifications written in this document are believed to be accurate, but are not guaranteed to be entirely free of error. The information in this manual is subject to change for functional or performance improvements without notice. Please make sure your manual is the latest edition. While the information herein is assumed to be accurate, SEGGER Microcontroller GmbH & Co. KG (the manufacturer) assumes no responsibility for any errors or omissions. The manufacturer makes and you receive no warranties or conditions, express, implied, statutory or in any communication with you. The manufacturer specifically disclaims any implied warranty of merchantability or fitness for a particular purpose.

Copyright notice

You may not extract portions of this manual or modify the PDF file in any way without the prior written permission of the manufacturer. The software described in this document is furnished under a license and may only be used or copied in accordance with the terms of such a license.

© 2002 - 2008 SEGGER Microcontroller GmbH & Co. KG, Hilden / Germany

Trademarks

Names mentioned in this manual may be trademarks of their respective companies.

Brand and product names are trademarks or registered trademarks of their respective holders.

Contact address

SEGGER Microcontroller GmbH & Co. KG

In den Weiden 11
D-40721 Hilden

Germany

Tel.+49 2103-2878-0
Fax.+49 2103-2878-28
Email: support@segger.com
Internet: http://www.segger.com

Manual versions

This manual describes the latest software version. If any error occurs, please inform us and we will try to assist you as soon as possible.

For further information on topics or routines not yet specified, please contact us.

Manual version	Date	By	Explanation
4.10 Rev. 2	090918	AG	Chapter "Command Line interface" * Section "Command line options" updated.
4.10 Rev. 1	090902	AG	Chapter "Device specifics" * Section "ST Microelectronics" updated.
4.10 Rev. 0	090825	AG	Chapter "Device specifics" * Section "ST Microelectronics" updated.
4.04 Rev. 1	090414	AG	Chapter "Introduction" * Section "What is J-Flash?" updated.
4.04 Rev. 0	090204	AG	Chapter "Command Line Interface" * Section "Overview" updated. * Section "Command Line Options" updated.
3.97e Rev. 0	081204	KN	Chapter "Target systems" * Section "Supported Flash Devices" updated Chapter "Settings" * Section "Init sequence" corrected

Table: List of manual versions

Manual version	Date	By	Explanation
3.91n Rev. 0	080923	AG	Chapter "Working with J-Flash" renamed to "Create a new J-Flash project." Chapter "Create a new J-Flash project" Chapter "Settings" * Section "Init sequence" updated. Chapter "Command Line Interface" updated. * Section "Create a new J-FLash project" updated.
3.90 Rev. 0	080811	AG	Chapter "Targets" * Section "Supported Microcontrollers" updated.
3.80 Rev. 2	080408	AG	Chapter "Licensing" * Section "Introduction" added. * Section "License types" added.
3.80 Rev. 1	080311	AG	Chapter "Target systems" * Section "Supported Microcontrollers" updated. Chapter "Working with J-Flash" * Section "Create a new J-Flash project" updated.
3.80 Rev. 0	080206	SK	Chapter "Device specifics" added. Chapter "Target systems" * Section supported MCUs updated.
3.68 Rev. 1	070508	SK	Chapter "Installation" updated. Chapter "Command Line Interface": * Section "Batch processing" added. Various improvements.
3.66 Rev. 1	070322	SK	Chapter "Target systems" updated. Chapter "Getting started" updated.
3.46 Rev. 4	061222	SK	Sektion "About" and company description added.
3.46 Rev. 3	061124	OO	Chapter "Performance" updated.
3.46 Rev. 2	061121	OO	Chapter "Performance" updated.
3.46 Rev. 1	060929	TQ	Update supported target devices.
3.42 Rev. 1	060912	TQ	Update supported target devices.
3.36 Rev. 1	060801	TQ	Update supported target devices.
3.24 Rev. 1	060530	TQ	Update supported target devices.
3.00 Rev. 2	060116	OO	Screenshots updated.
3.00 Rev. 1	060112	TQ	Nothing changed. Just a new software version.
2.14	051025	TQ	Update supported target devices.
2.10	050926	TW	Added troubleshooting section.
2.04	050819	TQ	Nothing changed. Just a new software version.
2.02	050808	TW	Command line added.
2.00	050707	TW	Initial Version

Table: List of manual versions

Software versions

Refers to Release.html for information about the changes of the software versions.

© 2004-2009 SEGGER Microcontroller GmbH & Co. KG

 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

About this document

Assumptions

This document assumes that you already have a solid knowledge of the following:

- The software tools used for building your application (assembler, linker, C compiler)
- The C programming language
- The target processor
- DOS command line.

If you feel that your knowledge of C is not sufficient, we recommend The C Programming Language by Kernighan and Richie (ISBN 0-13-1103628), which describes the standard in C-programming and, in newer editions, also covers the ANSI C standard.

How to use this manual

This manual explains all the functions that J-Flash offers. It assumes you have a working knowledge of the C language. Knowledge of assembly programming is not required.

Typographic conventions for syntax

This manual uses the following typographic conventions:

Style	Used for
Body	Body text.
`Keyword`	Text that you enter at the command-prompt or that appears on the display (that is system functions, file- or pathnames).
`Parameter`	Parameters in API functions.
`Sample`	Sample code in program examples.
Reference	Reference to chapters, tables and figures or other documents.
GUIElement	Buttons, dialog boxes, menu names, menu commands.
Emphasis	Very important sections

Table 1.1: Typographic conventions

© 2004-2009 SEGGER Microcontroller GmbH & Co. KG

SEGGER Microcontroller GmbH & Co. KG develops and distributes software development tools and ANSI C software components (middleware) for embedded systems in several industries such as telecom, medical technology, consumer electronics, automotive industry and industrial automation.

SEGGER's intention is to cut software development-time for embedded applications by offering compact flexible and easy to use middleware, allowing developers to concentrate on their application.

Our most popular products are emWin, a universal graphic software package for embedded applications, and embOS, a small yet efficent real-time kernel. emWin, written entirely in ANSI C, can easily be used on any CPU and most any display. It is complemented by the available PC tools: Bitmap Converter, Font Converter, Simulator and Viewer. embOS supports most 8/16/32-bit CPUs. Its small memory footprint makes it suitable for single-chip applications.

Apart from its main focus on software tools, SEGGER developes and produces programming tools for flash microcontrollers, as well as J-Link, a JTAG emulator to assist in development, debugging and production, which has rapidly become the industry standard for debug access to ARM cores.

Corporate Office:
http://www.segger.com

United States Office:
http://www.segger-us.com

EMBEDDED SOFTWARE (Middleware)

emWin
Graphics software and GUI

emWin is designed to provide an efficient, processor- and display controller-independent graphical user interface (GUI) for any application that operates with a graphical display. Starterkits, eval- and trial-versions are available.

embOS
Real Time Operating System

embOS is an RTOS designed to offer the benefits of a complete multitasking system for hard real time applications with minimal resources. The profiling PC tool embOSView is included.

emFile
File system

emFile is an embedded file system with FAT12, FAT16 and FAT32 support. emFile has been optimized for minimum memory consumption in RAM and ROM while maintaining high speed. Various Device drivers, e.g. for NAND and NOR flashes, SD/MMC and CompactFlash cards, are available.

emUSB
USB device stack

A USB stack designed to work on any embedded system with a USB client controller. Bulk communication and most standard device classes are supported.

SEGGER TOOLS

Flasher
Flash programmer

Flash Programming tool primarily for microcontrollers.

J-Link
JTAG emulator for ARM cores

USB driven JTAG interface for ARM cores.

J-Trace
JTAG emulator with trace

USB driven JTAG interface for ARM cores with Trace memory. supporting the ARM ETM (Embedded Trace Macrocell).

J-Link / J-Trace Related Software

Add-on software to be used with SEGGER's industry standard JTAG emulator, this includes flash programming software and flash breakpoints.

Table 1.1:

© 2004-2009 SEGGER Microcontroller GmbH & Co. KG

Table of Contents

1 Introduction 9

1.1 What is J-Flash? 10
1.1.1 Features 10
1.2 Assumptions 11
1.3 Requirements 12
1.3.1 Host 12
1.3.2 Target 12

2 Licensing 13

2.1 Introduction 14
2.2 License types 15
2.2.1 Built-in license 15
2.2.2 Key-based license 16
2.2.2.1 The serial number 16
2.2.2.2 License management 16

3 Getting Started 19

3.1 Setup 20
3.1.1 What is included? 20
3.2 Using J-Flash for the first time 21
3.2.1 Sample Projects 22
3.3 Menu structure 24

4 Settings 27

4.1 Project Settings 28
4.1.1 General Settings 28
4.1.1.1 TCP/IP 29
4.1.2 JTAG Settings 30
4.1.2.1 JTAG Speed 30
4.1.2.2 JTAG scan chain with multiple devices 31
4.1.3 CPU Settings 31
4.1.3.1 Core 32
4.1.3.2 Device 32
4.1.3.3 Clock 32
4.1.3.4 Endianess 32
4.1.3.5 Check core ID 32
4.1.3.6 Use target RAM 32
4.1.3.7 Init sequence 33
4.1.4 Flash Settings 35
4.1.4.1 Base Address 35
4.1.4.2 Organization 35
4.1.4.3 Select flash device 36
4.1.4.4 ID checking 36
4.1.4.5 Sector selection 36
4.1.5 Production settings 37
4.2 Global Settings 38
4.2.1 Operation 38
4.2.1.1 Disconnect after each operation 38
4.2.1.2 Automatically unlock sectors 38
4.2.1.3 Perform blank check 38

 © 2005 - 2007 SEGGER Microcontroller GmbH & Co. KG

4.2.1.4 Skip blank areas on read 38
4.2.2 Logging 38
4.2.2.1 General log level 38
4.2.2.2 Enable J-Link logfile 39

5 Command Line Interface 41

5.1 Overview 42
5.2 Command line options 43
5.3 Batch processing 45

6 Create a new J-Flash project 47

6.1 Creating a new J-Flash project 48
6.2 Creating a new init sequence 52
6.2.1 Example init sequence 52

7 Device specifics 53

7.1 Analog Devices 54
7.1.1 ADuC7xxx 54
7.2 ATMEL 55
7.2.1 AT91SAM7 55
7.2.2 AT91SAM9 55
7.3 NXP 56
7.3.1 LPC2xxx 56
7.4 OKI 57
7.4.1 ML67Q40x 57
7.5 ST Microelectronics 58
7.5.1 STM32F10x 58
7.5.1.1 Securing/Unsecuring the chip 58
7.5.1.2 Option byte programming 58
7.5.2 STR 71x 59
7.5.3 STR 73x 59
7.5.4 STR 75x 59
7.5.5 STR91x 59
7.6 Texas Instruments 60
7.6.1 TMS470 60

8 Target systems 61

8.1 Which devices can be programmed by J-Flash? 62
8.2 Supported Microcontrollers 63
8.3 Supported Flash Devices 64

9 Performance 71

9.1 Performance of MCUs with internal flash memory 72
9.2 Performance of MCUs with external flash memory 73

10 Support 75

10.1 Troubleshooting 76
10.1.1 General procedure 76
10.1.2 Typical problems 76
10.2 Contacting support 78

 © 2005 - 2007 SEGGER Microcontroller GmbH & Co. KG

Chapter 1

Introduction

The following chapter introduces J-Flash, highlights some of its features, and lists its requirements on host and target systems.

 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

1.1 What is J-Flash?

J-Flash is a stand-alone flash programming software for PCs running Microsoft Windows. The following Microsoft Windows versions are supported:

- Microsoft Windows 2000
- Microsoft Windows XP
- Microsoft Windows XP x64
- Microsoft Windows 2003
- Microsoft Windows 2003 x64
- Microsoft Windows Vista
- Microsoft Windows Vista x64

J-Flash has an intuitive user interface and makes programming flash devices convenient. J-Flash requires a J-Link, JTAG emulator for ARM cores, to interface to the hardware. It is able to program internal and external flash at very high speeds, upwards of 200 kB/sec depending on the chip. J-Flash has an approximate blank check speed of 16 MB/sec. Another notable feature is smart read back, which only transfers non-blank portions of the flash, increasing the speed of read back greatly. These features along with its ability to work with any ARM7 or ARM9 chip makes it a great solution for most projects.

1.1.1 Features

- Any ARM7/ARM9 and Cortex-M3 core supported, including thumb mode.
- ARM microcontroller (internal flash) support.
- Support for most external flash chips (see chapter *Target systems* on page 61 for a list of supported devices).
- High speed programming: up to 200 KBytes/sec* (depending on flash device).
- Very high speed blank check: approximately 16 MBytes/sec (depending on the chip).
- Smart read back: only non-blank portions of flash are transferred and saved.
- Free evaluation licenses available.
- Verbose logging of all communication.
- .hex, .mot, .srec, and .bin support.
- Intuitive user interface.

* = Measured with J-Link ARM Rev.5 in DCC mode

 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

1.2 Assumptions

This user manual assumes that you already possess working knowledge of the J-Link device. If you feel that your knowledge of J-Link is not sufficient, we recommend the J-Link manual, which describes the device and its use in detail.

 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

1.3 Requirements

1.3.1 Host

J-Flash requires a PC running Microsoft Windows 2000 or Windows XP with a free USB port dedicated for a J-Link. A network connection is required only if you want to use J-Flash together with a remote J-Link server.

1.3.2 Target

A JTAG interface must be available on the target device to establish the connection with the host system. A network connection must be available if and only if it is desired to connect to the J-Link through the J-Link TCP/IP Server from a remote system.

 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

Chapter 2

Licensing

The following chapter provides an overview of J-Flash related licensing options.

 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

2.1 Introduction

J-Flash may be installed on as many host machines as you want. Without a license key you can still use J-Flash to open project files, read from connected devices, blank check target memory, verify data files and so on. However to actually program devices via J-Flash and J-link you are required to obtain a license key from us. A J-Flash license is bound to the serial number of a J-Link. Evaluation licenses which allow you to unlock the full potential of J-Flash for a limited period of time are available upon request. If you need an evaluation license key you only have to tell us the serial number of your J-Link which allows us to send you a proper key. In any case you need to have a license key for each J-Link you want to work with via J-Flash. The following sections describe common operations with reference to handling license keys.

 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

2.2 License types

For J-Flash there are two different types of licenses which are explained below:

Built-in License

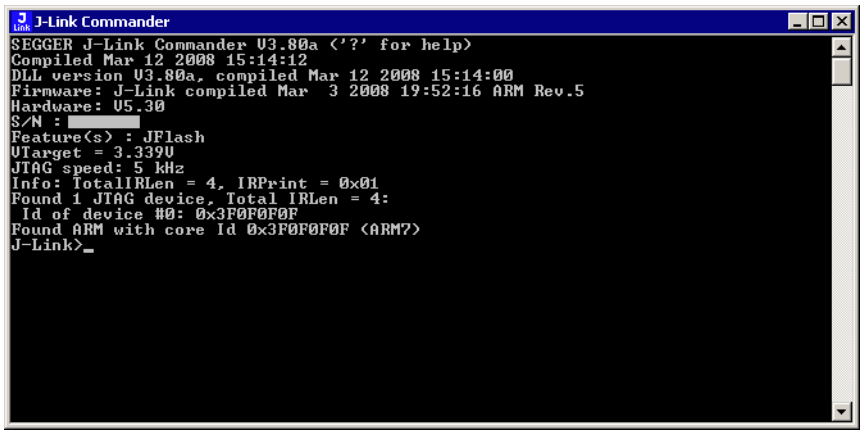
This type of license is easiest to use. The customer does not need to deal with a license key. The software automatically finds out that the connected J-Link contains the built-in license(s). This is the type of license you get if you order J-Link and the license at the same time, typically in a bundle.

Key-based license

This type of license is used if you already have a J-Link, but want to enhance its functionality by using J-Flash. In addition to that, the key-based license is used for trial licenses. To enable this type of license you need to obtain a license key from SEGGER. Free trial licenses are available upon request from *www.segger.com*. This license key has to be added to the J-Flash license management. How to enter a license key is described in detail in section *Key-based license* on page 16. Every license can be used on different PCs, but only with the J-Link the license is for. This means that if you want to use J-Flash with other J-Links, every J-Link needs a license.

2.2.1 Built-in license

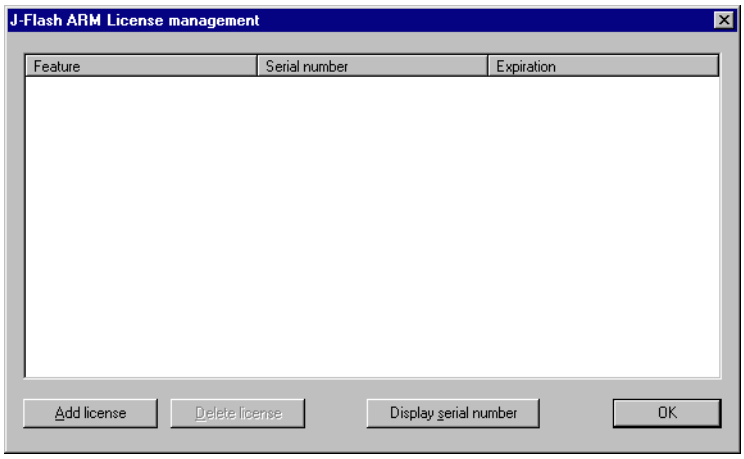
This type of license is easiest to use. The customer does not need to deal with a license key. The software automatically finds out that the connected J-Link contains the built-in license(s). To check what licenses the used J-Link have, simply open the J-Link commander (`JLink.exe`). The J-Link commander finds and lists all of the J-Link's licenses automatically, as can be seen in the screenshot below.



 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

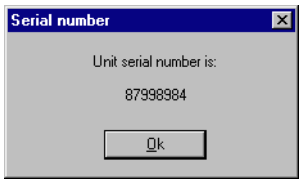
2.2.2 Key-based license

When using a key-based license, a license key is required in order to unlock the full potential of J-Flash. License keys can be added via the J-Flash license management. To get to the J-Flash license management just select Licenses... from the Help menu of the main window. Like the built-in license, the key-based license is only valid for one J-Link, so if another J-Link is used it needs a separate license.



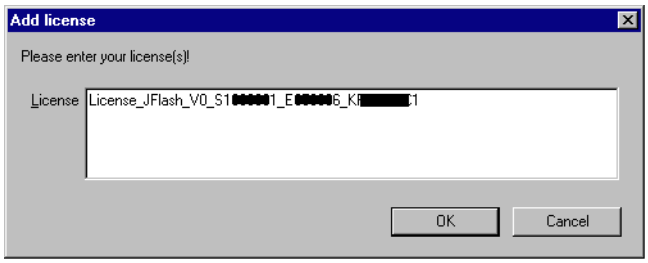
2.2.2.1 The serial number

The licensing dialog contains a button Display serial number. J-Flash tries to read the serial number of a connected J-Link if you press this button.



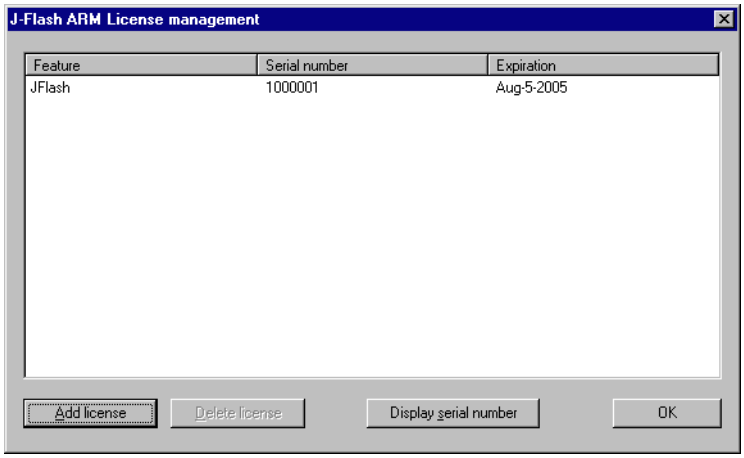
2.2.2.2 License management

The licensing dialog contains buttons to add and remove license keys. After you received a key from us, click on Add license to unlock J-Flash. Depending on the license you requested you are free to use J-Flash either for an unlimited or limited period of time. Enter the key into the Add license dialog and click OK to submit.



 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

The licensing dialog will show the licenses together with their expiration date, the serial number they are bound to and the feature that is licensed by the respective key.



You may select individual license keys for removal. Click the Delete license button after selecting the key you want to remove. The key is deleted immediately without asking for confirmation and the licensed features become unavailable.

© 2004-2009 SEGGER Microcontroller GmbH & Co. KG

 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

Chapter 3

Getting Started

This chapter presents an introduction to J-Flash. It provides an overview of the included sample projects and describes J-Flash's menu structure in detail.

 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

3.1 Setup

The J-Link setup procedure required in order to work with the J-Flash is described in chapter 2 of the *J-Link / J-Trace User Guide*. The J-Link User Guide is part of the J-Link software package which is available for download under *www.segger.com*.

3.1.1 What is included?

The following table shows the contents of all subdirectories of the J-Link ARM software and documentation pack with regard to J-Flash:

Directory	Contents
.	The J-Flash application. Please refer to the J-Link manual for more information about the other J-Link related tools.
.\Doc	Contains the J-Flash documentation and the other J-Link related manuals.
.\ETC\JFlash\	Two *.csv files for the J-Flash internal management of supported MCU's und flash chips.
.\Sample\JFlash\ProjectFiles\	Contains sample projects with good default settings (see section *Sample Projects* on page 22 for further details).

Table 3.1: J-Flash directory structure

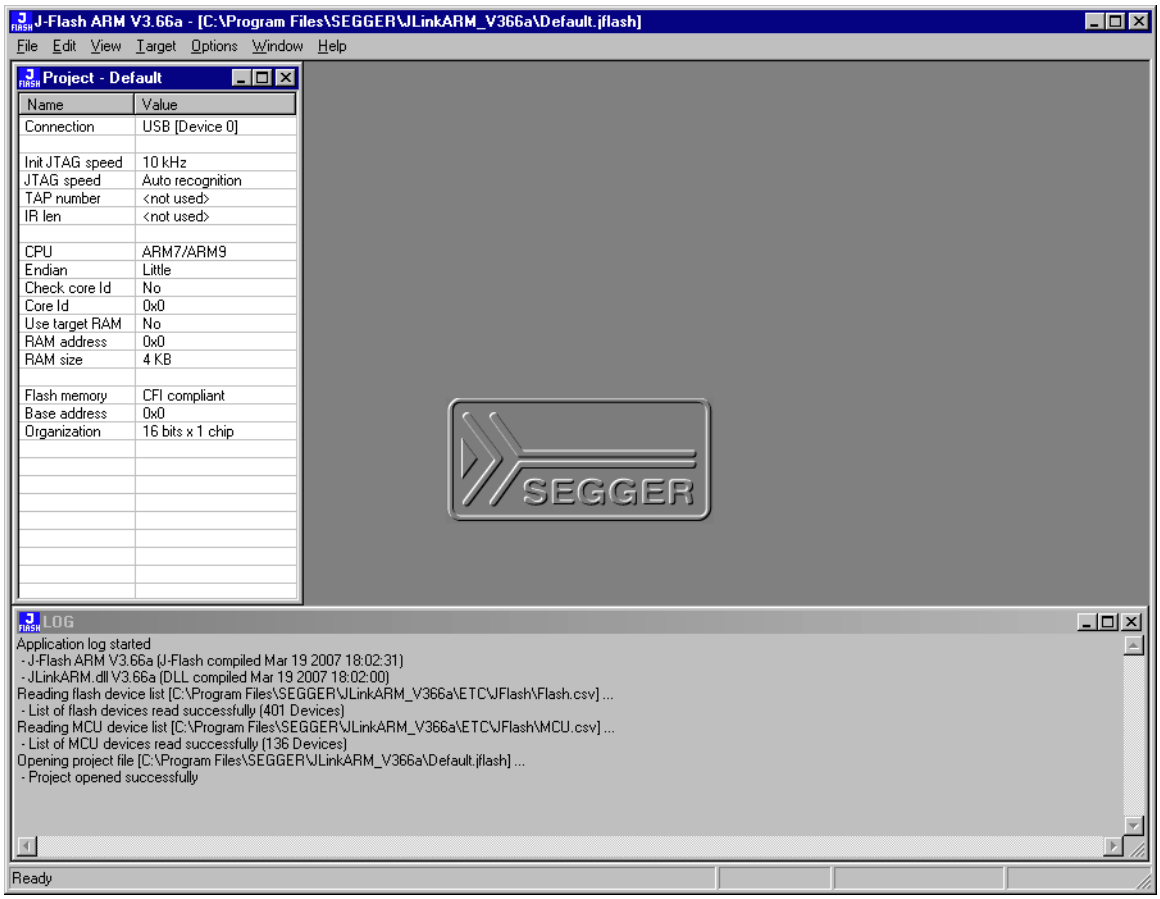
 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

3.2 Using J-Flash for the first time

Start J-Flash from the Windows Start menu. J-Flash's main window will appear, which contains a log window at the bottom and the **Project window** of a default project on the left. The application log will initially display:

- The version and time of compilation for the J-Flash application.
- The version and time of compilation for the J-Link DLL.
- The number of supported flash devices.
- The number of supported MCU devices.
- The location of the default project.

The Project window contains an overview of the current project settings (initially J-Flash opens a default project).



 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

3.2.1 Sample Projects

If you are new to J-Flash, it might be a good idea to open one of our sample projects to familiarize yourself with the application. You find those project files in the Projects subdirectory of J-Flash's installation directory. Once you have opened a project file, the project window contains the relevant project settings, e.g. chip type, clock speed, RAM size etc. The settings are known to be good defaults for the respective devices. You may then continue to open your own data files to actually program your device. The table below contains the included project files together with a short description.

Project	Description
ADuC7020.jflash	Analog Devices ADuC7020 with internal flash memory
ADuC7030.jflash	Analog Devices ADuC7030 with internal flash memory
ADuC7032.jflash	Analog Devices ADuC7032 with internal flash memory
ADuC7229.jflash	Analog Devices ADuC7229 with internal flash memory
AT91FR40162.jflash	AT91FR40162 with internal AT49BV1614A flash memory
AT91M42800A.jflash	AT91M42800A with internal M29W200BB flash memory
AT91M55800A.jflash	AT91M55800 with Am29LV320DT flash memory
AT91R40008_AT91EB40A.jflash	AT91R40008 with external AT91EB40A flash memory
AT91RM9200_CSB337.jflash	Cogent CSB337 eval board with AT91RM9200
AT91RM9200_CSB637.jflash	Cogent CSB637 eval board with AT91RM9200
AT91RM9200_EK.jflash	Atmel AT91RM9200 eval board
AT91SAM7A1_EK.jflash	Atmel AT91SAM7A1 eval board with CFI compliant flash memory
AT91SAM7A3.jflash	Atmel AT91SAM7A3 with internal flash memory
AT91SAM7S32.jflash	AT91SAM7S-EK eval board with SAM7S32
AT91SAM7S64.jflash	AT91SAM7S-EK eval board with SAM7S64
AT91SAM7S128.jflash	AT91SAM7S-EK eval board with SAM7S128
AT91SAM7S256.jflash	AT91SAM7S-EK eval board with SAM7S256
AT91SAM7SE512.jflash	AT91SAM7SE-EK eval board with SAM7SE512
AT91SAM7X128.jflash	AT91SAM7X-EK eval board with SAM7X128
AT91SAM7X256.jflash	AT91SAM7X-EK eval board with SAM7X256
DragonballMX1.jflash	DragonballMX1 eval board with ST M29W400BB
Evaluator7T.jflash	Evaluator7T eval board with SST39LF/VF400A flash memory
LH75411.jflash	Sharp LH75411 with Macronix MX29LV320AB flash memory
LH79520_LogicPD.jflash	Sharp LH79520 with Intel 28F640J3 flash memory
LH79524_LogicPD.jflash	Sharp LH79524 with Sharp LH28F128SPHTD flash memory
LH7A40x_LogicPD.jflash	Sharp LH7A40x with Intel 28F640J3 flash memory (2 chips)
LPC2103.jflash	NXP LPC2103 with internal flash memory
LPC2106.jflash	NXP LPC2106 with internal flash memory
LPC2129_MCB2100.jflash	Keil MCB2100 eval board with NXP LPC2129
LPC2138.jflash	NXP LPC2138 with internal flash memory

Table 3.2: List of sample J-Flash projects

 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

Project	Description
LPC2148.jflash	NXP LPC2148 with internal flash memory
LPC2290.jflash	NXP LPC2290 with internal flash memory
LPC2294.jflash	NXP LPC2294 with internal flash memory
LPC2294_PhyCORE.jflash	NXP LPC2294 with external Am29DL800BT flash memory
LPC2366.jflash	NXP LPC2366 with internal flash memory
LPC2378.jflash	NXP LPC2378 with internal flash memory
MAC7111.jflash	Freescale MAC7111LC eval board with internal flash
ML67Q4050.jflash	OKI ML67Q4050 with internal flash memory
ML67Q4051.jflash	OKI ML67Q4051 with internal flash memory
ML67Q4060.jflash	OKI ML67Q4060 with internal flash memory
ML67Q4061.jflash	OKI ML67Q4061 with internal flash memory
NS7520_CC7U_352.jflash	Digi ConnectCore7U with NetSilicon NS7520 and external Fujitsu MBM29LV650U flash
NS7520_CC7U_355.jflash	Digi ConnectCore7U with NetSilicon NS7520 and external AMD Am29LV160BB flash
NS9360.jflash	NetSilicon NS9360 with external AM29LV160DB flash (2 chips)
NS9750.jflash	NetSilicon NS9750 with Atmel AT49BV322A flash memory
PCF87750.jflash	NXP PCF87750 with internal flash memory
PXA255_CSB625.jflash	Intel XScale PXA255 with external flash memory
S3F445HX.jflash	Samsung S3F445HX with internal flash memory
SJA2010HL.jflash	NXP SJA2010 with internal flash memory
SJA2510HL.jflash	NXP SJA2510 with internal flash memory
SocLitePlus.jflash	NEC System-On-Chip Lite+ with internal memory
STR710.jflash	ST STR710FZ2T6 with internal flash memory
STR711.jflash	ST STR711FR2T6 with internal flash memory
STR712.jflash	ST STR712FR2T6 with internal flash memory
STR730.jflash	ST STR730FZ2 with internal flash memory
STR750.jflash	ST STR750FV2 with internal flash memory
STR912.jflash	ST STR912FM44 with internal flash memory
TMS470R1A64.jflash	TI TMS470R1A64 with internal flash memory
TMS470R1A128.jflash	TI TMS470R1A128 with internal flash memory
TMS470R1A256.jflash	TI TMS470R1A256 with internal flash memory
TMS470R1A288.jflash	TI TMS470R1A288 with internal flash memory
TMS470R1A384.jflash	TI TMS470R1A384 with internal flash memory
TMS470R1B1M.jflash	TI TMS470R1B1M with internal flash memory
TMS470R1B512.jflash	TI TMS470R1B512 with internal flash memory
TMS470R1VF689.jflash	TI TMS470R1VF689 with internal flash memory

Table 3.2: List of sample J-Flash projects

 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

3.3 Menu structure

The main window of J-Flash contains seven drop-down menus (**File**, **Edit**, **View**, **Target**, **Options**, **Window**, **Help**). Any option within these drop-down menus that is followed by a three period ellipsis (...), is an option that requires more information before proceeding.

File menu elements

Command	Description
Open...	Opens a data file that may be used to flash the target device. The data file must be an Intel HEX file, a Motorola S file, or a Binary file (`.hex`, `.mot`, `.srec`, or `.bin`).
Merge	Merges two data files (`.hex`, `.mot`, `.srec`, or `.bin`).
Save	Saves the data file that currently has focus.
Save As...	Saves the data file that currently has focus using the name and location given.
New Project	Creates a new project using the default settings.
Open Project...	Opens a J-Flash project file. Note that only one project file may be open at a time. Opening a project will close any other project currently open.
Save Project	Saves a J-Flash project file.
Save Project As...	Saves a J-Flash project file using the name and location given.
Close Project	Closes a J-Flash project file.
Export Setup File...	Exports a file that can be used to setup the J-Link. Please refer to the J-Link documentation for more information regarding J-Link setup files.
Recent Files >	Contains a list of the most recently open data files.
Recent Projects >	Contains a list of the most recently open project files.
Exit	Exits the J-Flash application.

Table 3.3: File menu elements

Edit menu elements

Command	Description
Relocate...	Relocates the start of the data file to the supplied hex offset from the current start location.
Delete range...	Deletes a range of values from the data file, starting and ending at given addresses. The End address must be greater than the Start address otherwise nothing will be done.
Eliminate blank areas...	Eliminates blank regions within the data file.

Table 3.4: Edit menu elements

View menu elements

Command	Description
Log	Opens and/or brings the log window to the active window.
Project	Opens and/or brings the project window to the active window.

Table 3.5: View menu elements

 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

Target menu elements

Command	Description
Connect	Creates a connection through the J-Link using the configuration options set in the Project settings... of the Options drop-down menu.
Disconnect	Disconnects a current connection that has been made through the J-Link.
Show CFI info...	Reads the CFI query information of a CFI compliant flash device.
Test >	Two test functions are implemented "Generates test data" generates data which can be used to test if the flash can be programmed correctly. The size of the generated data file can be defined. "Tests up/download speed" writes data of an specified size to an defined address, reads the written data back and measures the up- and download speed.
Lock/Unlock sectors >	Sectors may be locked and unlocked. The soft lock and soft unlock work on a software only basis for those sectors that have been selected on the Flash tab of the Project Settings... found in the Options drop-down menu. If the software locks a sector with soft lock, it can easily be unlocked using the soft unlock feature. The hard lock and hard unlock work on a hardware only basis. If a sector is locked using the hard lock command, it can only be unlocked through hardware support. For example, some flash devices have a special PIN that must be set high or low to allow an unlock command.
Secure chip	Secures the MCU.
Unsecure chip	Unsecures the MCU.
Check blank	Checks flash to see if it is empty.
Fill with zero	Fills all selected flash sectors with zero. Some flash chips need this before erasing them.
Erase sectors	Erases all selected flash sectors.
Erase chip	Erases the entire chip.
Program	Programs the chip using the currently active data file.
Program & Verify	Programs the chip using the currently active data file and then verifies that it was written successfully.
Auto	The Auto command performs a sequence of steps. It connects to the device, erases sectors and programs the chip using the currently active data file before the written data is finally verified. The range of sectors to be erased can be configured through the Flash tab of the Project settings dialog and through the Global settings dialog. See chapter *Settings* on page 27 for further details.
Verify	Verifies the data found on the chip with the data file.

Table 3.6: Target menu elements

© 2004-2009 SEGGER Microcontroller GmbH & Co. KG

Command	Description
VerifyCRC >	Verifies the CRC. There are three ways in which the CRC can be verified. "Affected sectors" verifies the CRC of the affected sectors. "Selected sectors" verifies the CRC of the selected sectors. "Entire chip" verifies the CRC of the entire chip.
Read back >	Reads back the data found on the chip and creates a new data file to store this information. There are three ways in which the data can be read back. The Selected sectors identified on the Flash tab of the Project Settings... found in the Options drop-down menu may be read back. The Entire chip may be read back. A specified Range... may be read back.
Start Application	Starts the application found on the chip.

Table 3.6: Target menu elements

Options menu elements

Command	Description
Project settings...	Location of the project settings that are displayed in the snapshot view found in the Project window of the J-Flash application as well as various settings needed to locate the J-Link and pass specified commands needed for chip initialization.
Global settings...	Settings that influence the general operation of J-Flash.

Table 3.7: Options menu elements

Window menu elements

Command	Description
Cascade	Arranges all open windows, one above the other, with the active window at the top.
Tile Horizontal	Tiles the windows horizontally with the active window at the top.
Tile Vertical	Tiles the windows vertically with the active window at the left.

Table 3.8: Window menu elements

Help menu elements

Command	Description
J-Flash ARM User's Guide	Shows this help file in a PDF viewer such as Adobe Reader.
J-Link ARM User's Guide	Shows the J-Link ARM User's Guide in a PDF viewer such as Adobe Reader.
Licenses...	Shows a dialog with licensing information. The serial number of a connected J-Link may be read and licenses added or removed.
About...	J-Flash and company information.

Table 3.9: Help menu elements

 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

Chapter 4

Settings

The following chapter provides an overview of the program settings. Both, general and per project settings are considered.

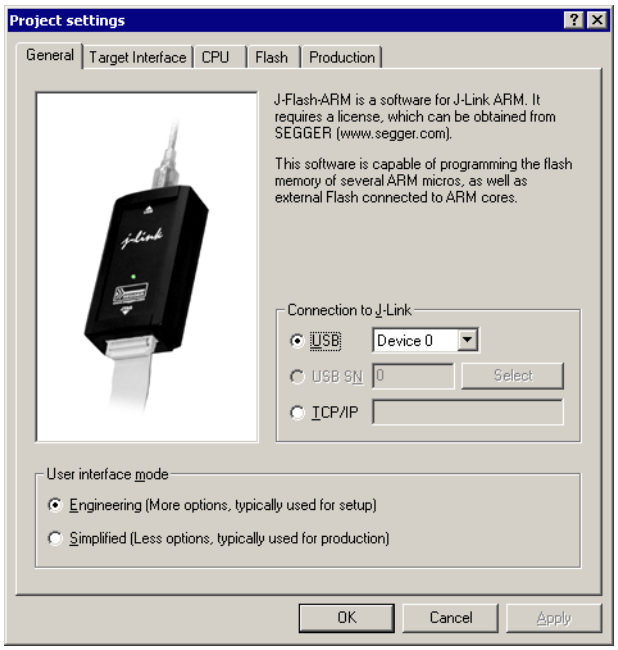
 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

4.1 Project Settings

Project settings are available from the Options menu in the main window or by using the ALT-F7 keyboard shortcut.

4.1.1 General Settings

This dialog is used to choose the connection to J-Link. The J-Link can either be connected directly over USB to the host system of J-Flash, or it can be connected through the J-Link TCP/IP Server running on a remote system. Refer to the J-Link manual for more information regarding the operation of J-Link and J-Link TCP/IP Server.



Since J-Flash version 3.74 can the complexity of user interface be selected. Select the **Engineering** checkbox if you want to setup your project or the **Simplified** checkbox if you use J-Flash in production environments. In the simplified user interface are some options disabled to decrease possible error sources in the production phase.

 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

4.1.1.1 USB

If this option is checked, J-Flash will connect to J-Link over the USB port. You may change the device number if you want to connect more than one J-Link to your PC. The default device number is 0. For more information about how to use multiple J-Links on one PC, please see also the chapter "Working with J-Link" of the J-Link ARM User's Guide.

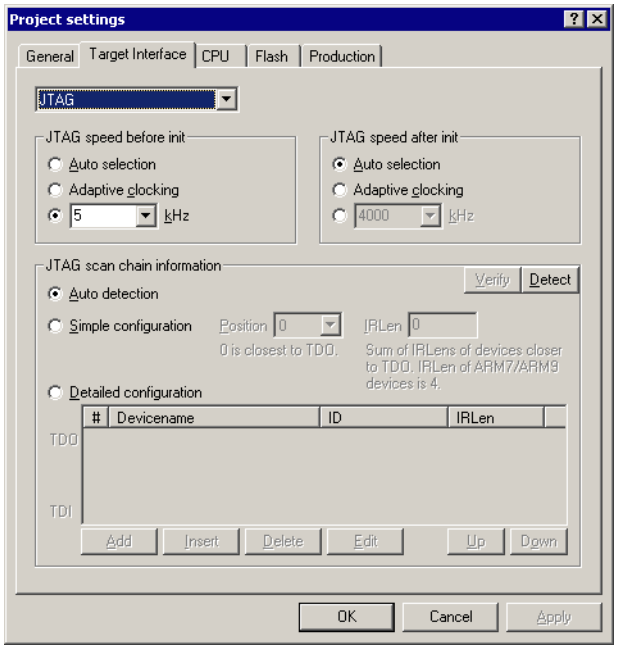
4.1.1.2 TCP/IP

If this option is checked, J-Flash will connect to J-Link via J-Link TCP/IP Server. You have to specify the hostname of the remote system running the J-Link TCP/IP Server.

 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

4.1.2 JTAG Settings

This dialog is used to configure the JTAG connection. You may change the JTAG speed or configure a JTAG scan chain with multiple devices.



4.1.2.1 JTAG Speed

You can configure the JTAG speed used before and after initialization. The JTAG speed before init is used to communicate with the target before and during execution of the custom initialization sequence (described in section *CPU Settings* on page 31). The JTAG speed after init is used to communicate after executing the custom initialization sequence. This is useful if you have a target running at slow speed and you want to set up a PLL in the initialization sequence.

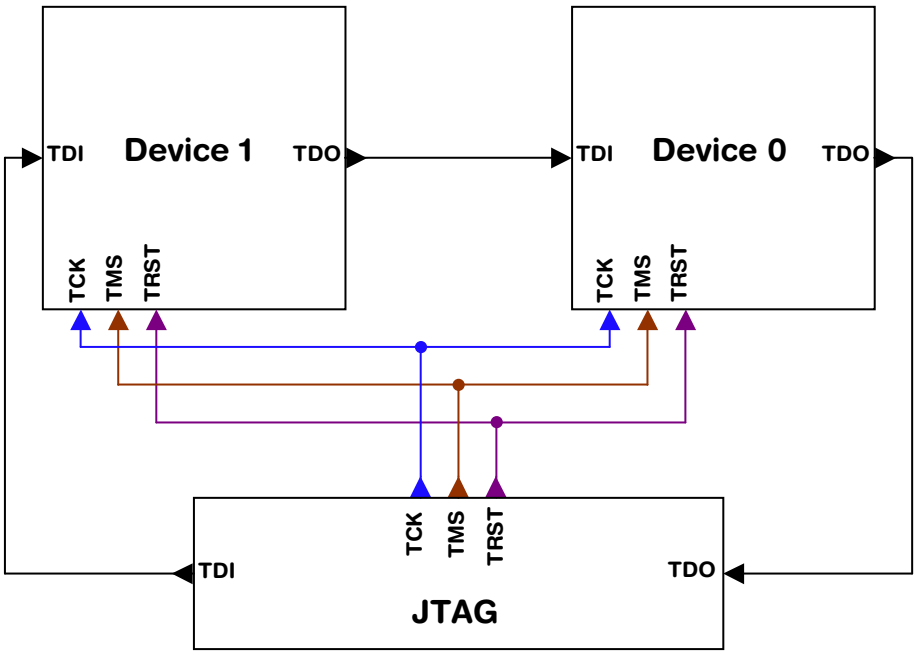
You can choose between automatic speed recognition, adaptive clocking or fixed JTAG speed. If you choose fixed JTAG speed you can select any value between 1kHz and 12MHz.

For more information about the different types of JTAG speed please see the chapter "Setup" of the J-Link ARM User’s Guide.

 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

4.1.2.2 JTAG scan chain with multiple devices

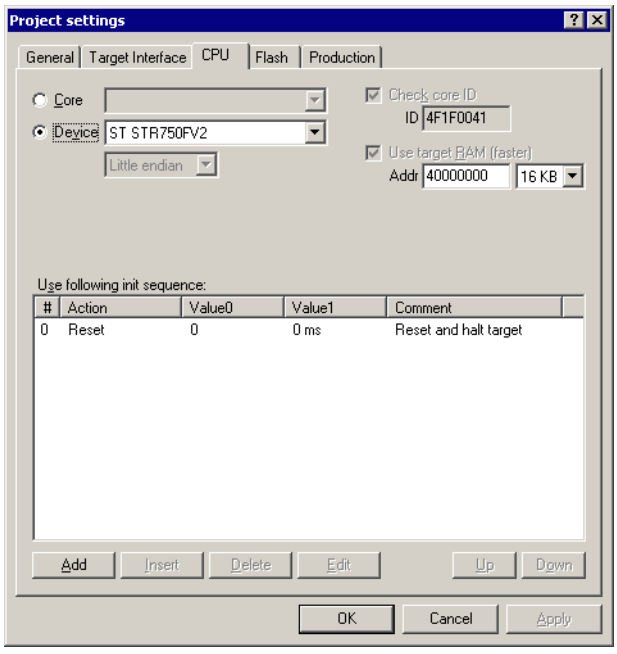
This checkbox allows you to configure a JTAG scan chain with multiple devices on it. In a scan chain configuration with multiple devices, the TCK and TMS lines of all JTAG device are connected, while the TDI and TDO lines form a ring.



The position of the device to connect with J-Flash is selected from the Position drop-down menu. The Instruction Register length (IRLen) of a device is defined by its manufacturer. For ARM cores, the IRLen is always four, which is why the value of IRLen is by default set to four times the position indicated. This works fine for ARM only scan chains. However, if any non-ARM devices are introduced to the scan chain the IRLen must be modified accordingly.

4.1.3 CPU Settings

This dialog allows the selection of microcontroller dependent settings.



© 2004-2009 SEGGER Microcontroller GmbH & Co. KG

J-Flash can be used to program both external or internal flash memory. In order to use J-Flash with an external flash device, the proper **Core** must be selected.

To program internal flash devices choose the respective microcontroller in the **Device** list. If your microcontroller is not found on this list, contact SEGGER as new microcontrollers are continuously being added.

4.1.3.1 Core

Select **Generic ARM7/ARM9** or **XSCALE** depend on the used MCU architecture version to program external flash.

4.1.3.2 Device

Select the respective microcontroller from the list to program internal flash devices.

4.1.3.3 Clock

The correct clock frequency in Hz of your MCU is required to guarantee accurate operation of J-Flash. J-Flash uses default the Auto detection feature. We recommend to change this default behavior only if you know what you do. If you deactivate the Auto detection feature take into account, that you have to modify the value in this dialog if you set up a PLL or otherwise change the clock frequency in the init sequence.

4.1.3.4 Endianess

The compatible endianness of the selected device is set automatically. The endianess must be only explicit defined, if you select the **Core** family to program external flash. Select **Little endian** or **Big endian** from drop-down menu accordant to your core.

4.1.3.5 Check core ID

If the core ID is known for the device to be programmed, it can be used to verify that the device in communication via the J-Link is the intended device. The core ID for all listed devices is known, so is this option selected automatically if you select a device from the **Device** drop-down menu and can not be modified. If you select the core family from the **Core** drop-down menu, you can modify the default core ID.

4.1.3.6 Use target RAM

You may enable the use of target RAM to speed up flash operations. To use the target RAM, a start location in RAM and the amount of RAM to be used must be entered.

 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

4.1.3.7 Init sequence

Many microcontrollers require an initialization sequence for different reasons: When powered on, the PLL may not be initialized, which means the chip is very slow or a watchdog must be disabled manually. To use these chips you must first perform the required initialization.

This dialog allows the user to enter a custom initialization sequence using a predefined list of operations. After choosing an operation and corresponding values to be associated with the operation, a comment may be added to make it easier for others to determine its effect. The following list shows all valid commands which can be used in an init sequence:

Command	Value0	Value1	Description
Delay	--	Length of the delay	Sets a delay.
DisableMMU	--	--	Disables the MMU.
Disable Checks	--	--	Disables JTAG checks. Some CPUs (e.g. TMS470R1B1M) report JTAG communication errors while initializing, so that they can not be programmed if the JTAG communication checks are enabled.
Enable Checks	--	--	Enables JTAG checks. This option is activated by default.
Go	--	--	Starts the CPU
Halt	--	--	Halts the CPU
Reset	J-Link reset type	Length of the delay	Resets the CPU. Refer to the J-Link / J-Trace user guide for an detailed explanation of the different reset types.
Read 8bit	Address (Hex)	--	Reads 8bit from a given address and stores the value in an internal variable.
Read 16bit	Address (Hex)	--	Reads 16bit from a given address and stores the value in an internal variable.
Read 32bit	Address (Hex)	--	Reads 32bit from a given address and stores the value in an internal variable.
SetAllowRemoteRead	--	On/Off	This option defines if the emulator (remote) or the host handles the read access to target. This option is activated by default to enhance the performance.
SetAllowRemoteWrite	--	On/Off	This option defines if the emulator (remote) or the host handles the write access to target. This option is activated by default to enhance the performance.
Verify 8bit	Address (Hex)	Data	Verifies whether 8bit data on a declared address is identical to the declared 8bit data.
Verify 16bit	Address (Hex)	Data (Hex)	Verifies whether 16bit data on a declared address is identical to the declared 16bit data.

Table 4.1: J-Flash init sequence commands

 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

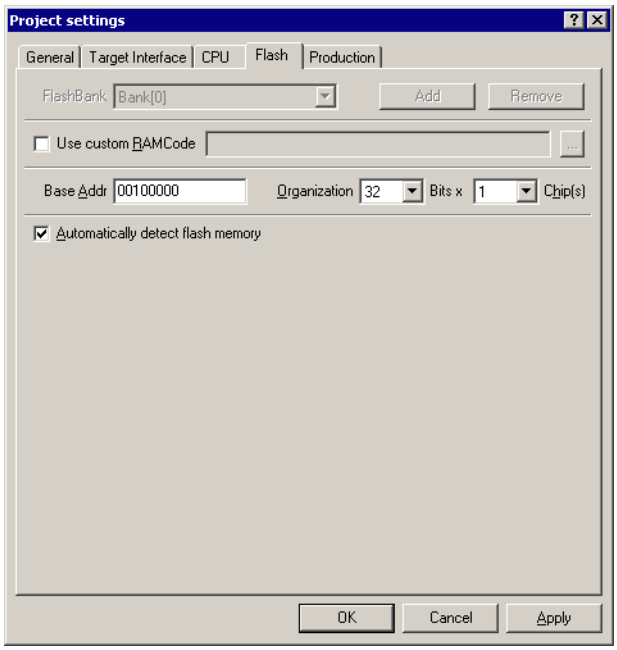
Command	Value0	Value1	Description
Verify 32bit	Address (Hex)	Data (Hex)	Verifies whether 32bit data on a declared address is identical to the declared 32bit data.
Write 8bit	Address (Hex)	Data (Hex)	Writes 8bit data to a given address.
Write 16bit	Address (Hex)	Data (Hex)	Writes 16bit data to a given address.
Write 32bit	Address (Hex)	Data (Hex)	Writes 32bit data to a given address.
Write Register	Register	Value	Writes data into a register.
Write JTAG IR	Command	--	Writes a command in the JTAG instruction register.
Write JTAG DR	NumBits	Data (Hex)	Writes a declared number of bits into the JTAG data register.
Var AND	--	Value (Hex)	Logical AND combination of the internal variable with a given value.
Var OR	--	Value (Hex)	Logical OR combination of the internal variable with a given value.
VAR XOR	--	Value (Hex)	Logical XOR combination of the internal variable with a given value.
VAR BEQ	Index	--	Checks if the internal variable is equal to 0. Performs jump to index on match.
VAR BNE	Index	--	Checks if the internal variable is not equal to 0. Performs jump to index on match.
Var Write 8bit	Address (Hex)	Data (Hex)	Writes 8bit into the internal variable.
Var Write 16bit	Address (Hex)	Data (Hex)	Writes 16bit into the internal variable.
Var Write 32bit	Address (Hex)	Data (Hex)	Writes 32bit into the internal variable.

Table 4.1: J-Flash init sequence commands

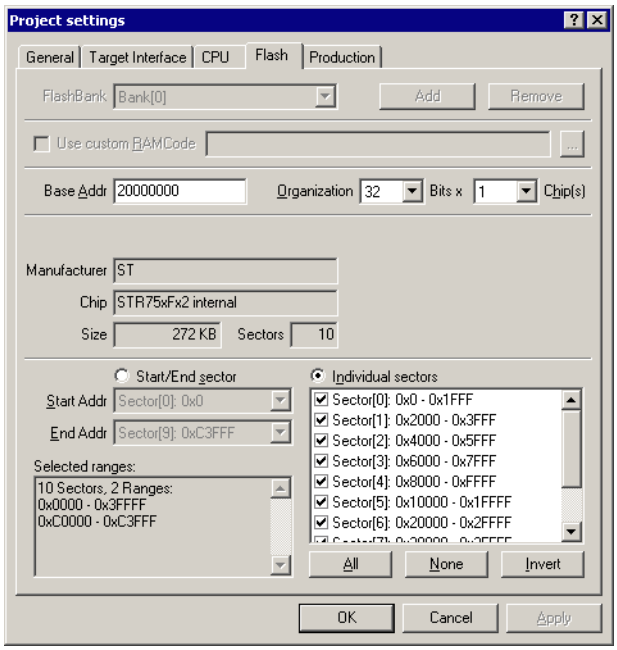
 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

4.1.4 Flash Settings

This dialog is used to select and configure the flash device to operate with. The listed options of the Flash settings menu are dependent on the selection in the **CPU** settings dialog. If you have selected a core family to program external flash memory, the menu should look similar to the screenshot below.



If you have selected a specific device to program the flash of these device, the menu should look similar to the screenshot below.



4.1.4.1 Base Address

You may enter the base address of the selected flash memory. The default value is 0.

4.1.4.2 Organization

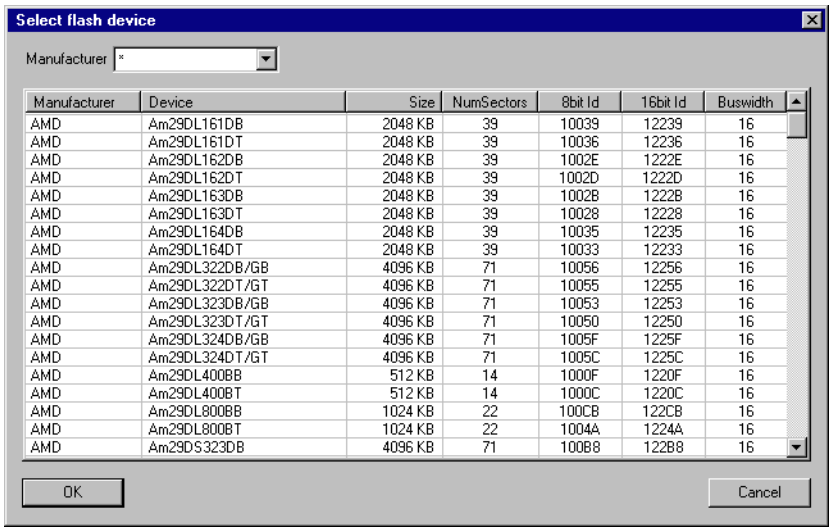
You should select the buswidth and the number of flash chips connected to the address and data bus of the MCU.

© 2004-2009 SEGGER Microcontroller GmbH & Co. KG

4.1.4.3 Select flash device

You can select a device manually or use the J-Flash **Auto Detection** feature. The auto detection feature is select by default. It supports both CFI compliant flash memory chips and non CFI compliant chips. You can select a device manually, if you deselect the Auto Detection checkbox and click on the **Select flash device** button.

After invoking this button a table will be presented. The table may be filtered using the manufacturer name. The chip and its attributes (manufacturer name, device name, size, number of sectors, eight bit identifier, sixteen bit identifier, bus width) must be selected from this table. If the flash chip is not found please contact SEGGER, as devices are continuously being added to this list.



4.1.4.4 ID checking

There are two other check boxes that are of interest in this subsection which are "Check manufacturer flash Id" and "Check product flash Id". These check boxes should be selected to confirm the type of device that is in communication with J-Flash.

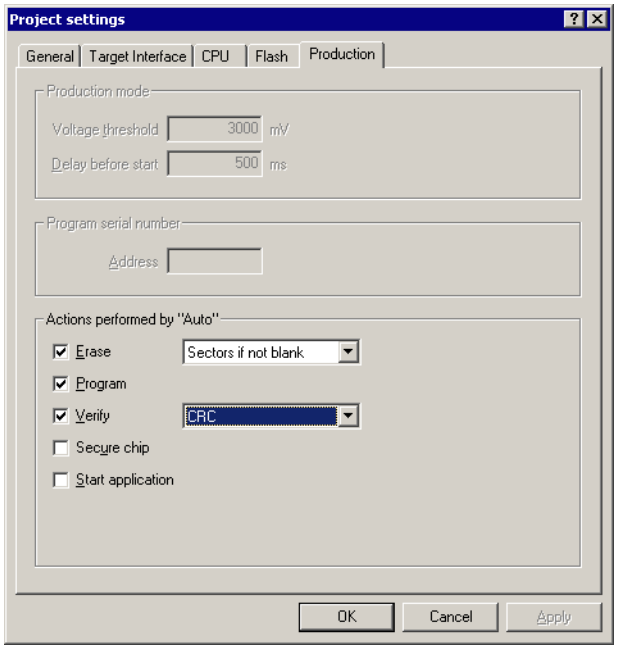
4.1.4.5 Sector selection

The final section of this dialog indicates the sectors to be acted upon, whether they are to be cleared, read back, or written. An individual or series of sectors may be selected from the predetermined valid range.

 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

4.1.5 Production settings

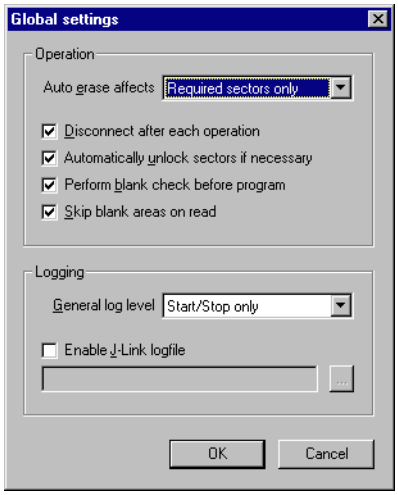
The performed actions of the auto programming feature (**Target** -> **Auto**, shortcut: F7) can be defined in the production settings dialog. The default behaviour is **Erase sectors if not blank**, **Program** and **Verify CRC**. You can optional include **Verify**, **Secure chip** and **Start application**.



© 2004-2009 SEGGER Microcontroller GmbH & Co. KG

4.2 Global Settings

Global settings are available from the Options menu in the main window.



4.2.1 Operation

You may define the behavior of some operations such as "Auto" or "Program & Verify".

4.2.1.1 Disconnect after each operation

If this option is checked, connection to the target will be closed at the end of each operation.

4.2.1.2 Automatically unlock sectors

If this option is checked, all sectors affected by an erase or program operation will be automatically unlocked if necessary.

4.2.1.3 Perform blank check

If this option is checked, a blank check is performed before any program operation to check if the affected flash sectors are completely empty. The user will be asked to erase the affected sectors if they are not empty.

4.2.1.4 Skip blank areas on read

If this option is checked, a blank check is performed before any read back operation to check which flash areas need to be read back from target. This improves performance of read back operations since it minimizes the amount of data to be transferred via JTAG and USB.

4.2.2 Logging

You may set some logging options to customize the log output of J-Flash.

4.2.2.1 General log level

This specifies the log level of J-Flash. Increasing log levels result in more information logged in the log window.

 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

4.2.2.2 Enable J-Link logfile

If this option is checked, you can specify a file name of the J-Link logfile. The J-Link logfile differs from the log window output of J-Flash. It does not log J-Flash operations performed. Instead of that, it logs the J-Link ARM DLL API functions called from within J-Flash.

 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

Chapter 5

Command Line Interface

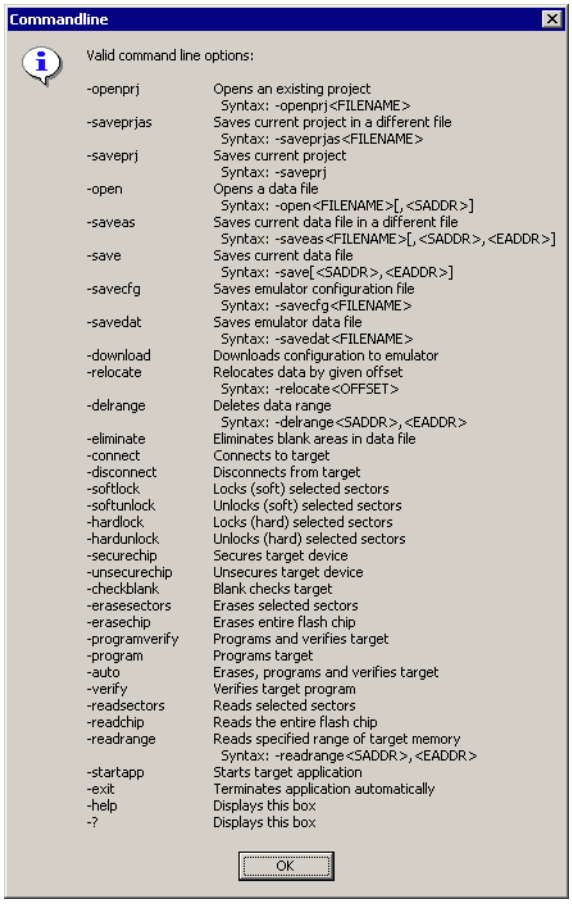
This chapter describes the J-Flash command line interface. The command line allows using J-Flash in batch processing mode and other advanced uses.

 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

5.1 Overview

In addition to its traditional Windows graphical user interface (GUI), J-Flash supports a command line mode as well. This makes it possible to use J-Flash for batch processing purposes. All important options accessible from the menus are available in command line mode as well. If you provide command line options, J-Flash will still start its GUI, but processing will start immediately.

The screenshot below shows the command line help dialog, which is displayed if you start J-Flash in a console window with `JFlashARM.exe -help` or `JFlashARM.exe -?`



 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

5.2 Command line options

This section lists and describes all available command line options. Some options accept additional parameters which are enclosed in angle brackets, e.g. <FILE-NAME>. If these parameters are optional they are enclosed in square brackets too, e.g. [<SADDR>]. Neither the angel nor the square brackets must be typed on the command line, they are used here only to denote (optional) parameters. Also, note that a parameter must follow immediately after the option, e.g. `JFlashARM.exe -openprjC:\Projects\Default.jflash`.

All command line options return 0 if the processing was successfully. An return value unequal 0 means that an error occured.

Option	Description
-openprj<FILENAME>	Open an existing project file.
-saveprjas<FILENAME>	Save the current project in the specified file.
-saveprj	Save the current project.
-open<FILENAME>[,<SADDR>]	Open a data file. Please note that the <SADDR> parameter applies only if the data file is a *.bin file.
-saveas<FILE-NAME>[,<SADDR>,<EADDR>]	Save the current data file into the specified file. Please note that the parameters <SADDR>, <EADDR> apply only if the data file is a *.bin file or *.c file.
-save[<SADDR>,<EADDR>]	Save the current data file. Please note that the parameters <SADDR>,<EADDR> apply only if the data file is a *.bin file or *.c file.
-savecfg<FILENAME>	Saves emulator config file.
-savedat<FILENAME>	Saves emulator data file.
-download	Downloads configuration to emulator.
-relocate<OFFSET>	Relocate data by the given offset.
-delrange<SADDR>,<EADDR>	Delete data in the given range.
-eliminate	Eliminate blank areas in data file.
-connect	Connect to target.
-disconnect	Disconnect from target.
-softlock	Lock (soft) selected sectors.
-softunlock	Unlock (soft) selected sectors.
-hardlock	Locks (hard) selected sectors.
-hardunlock	Unlocks (hard) selected sectors.
-checkblank	Blank check target.
-secure	Secures target device.
-unsecure	Unsecures target device.
-erasesectors	Erase selected sectors.
-erasechip	Erase the entire flash chip.
-programverify	Program and verify target.
-program	Program target.
-auto	Erase, program and verify target.
-verify	Verify target memory.
-readsectors	Read selected sectors.
-readchip	Read entire flash chip.
-readrange<SADDR>,<EADDR>	Read specified range of target memory.
-startapp	Start target application.

Table 5.1: J-Flash command line options

 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

Option	Description
-exit	Exit J-Flash.
-help	Display help dialog.
-?	Display help dialog.

Table 5.1: J-Flash command line options

 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

5.3 Batch processing

J-Flash can be used for batch processing purposes. All important options are available in command line mode as well. If you provide command line options, J-Flash will still start its GUI, but processing will start immediately.

The example batchfile displays a message, opens a project and a data file, starts auto processing and closes J-Flash. The return value will be checked and in case of an error an error message displayed.

Adapt the example according to the requirements of your project.

```
@ECHO OFF

ECHO Open a project and data file, start auto processing and exit
JFlashARM.exe -openprjC:\Projects\Default.jflash -openC:\Data\data.bin,0x100000 -
auto -exit
IF ERRORLEVEL 1 goto ERROR

goto END

:ERROR
ECHO J-Flash ARM:  Error!
pause

:END
```

Note, that every call of `JFlashARM.exe` has to completed with the `-exit` option, otherwise stops the execution of the batch file and the following commands will not be processed.

 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

Chapter 6

Create a new J-Flash project

This chapter contains information about the required steps how to setup a new J-Flash project.

 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

6.1 Creating a new J-Flash project

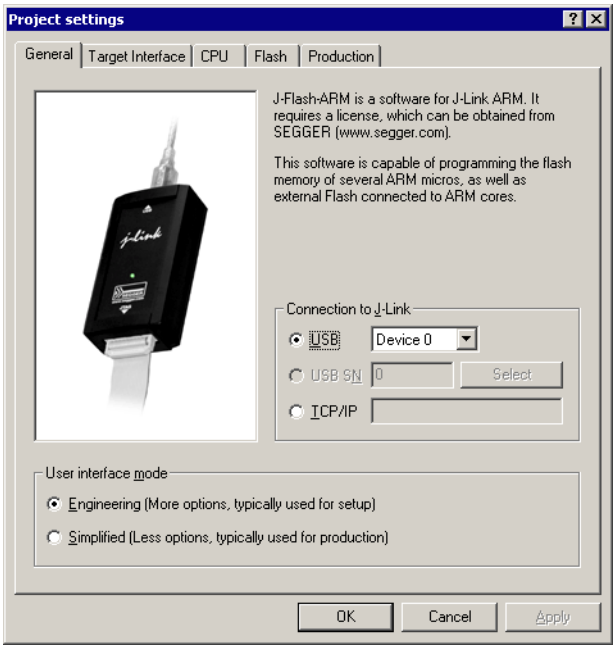
Before creating a new J-Flash project, you should have an understanding of your target system:

- Take a look at the schematic and the documentation of your CPU / SOC.
- Make sure the CPU runs at a decent speed (at least a few MHz, not just 32kHz)
- If necessary, enable & select a PLL as clock source.
- Locate RAM (Ideally on-chip RAM, even if it is just a 4KB) in the chip documentation.
- If necessary, use external RAM (usually SDRAM). You may have to setup the external bus interface.
- If necessary, use external FLASH. You may also have to setup the external bus interface to program the external flash since in a lot of cases, it allows per default just reading of flash memory.
- Last but not least, you should make sure the JTAG speed is as high as possible (on ARM-S cores with RTCK, Adaptive is usually a good choice; if not, 8Mhz or 12MHz is ideal. However, this should be the last step)

Note: Initialization of the PLL and the external bus interface has to be done in the init sequence of the project.

In the following all the necessary steps to create a project file, are explained.

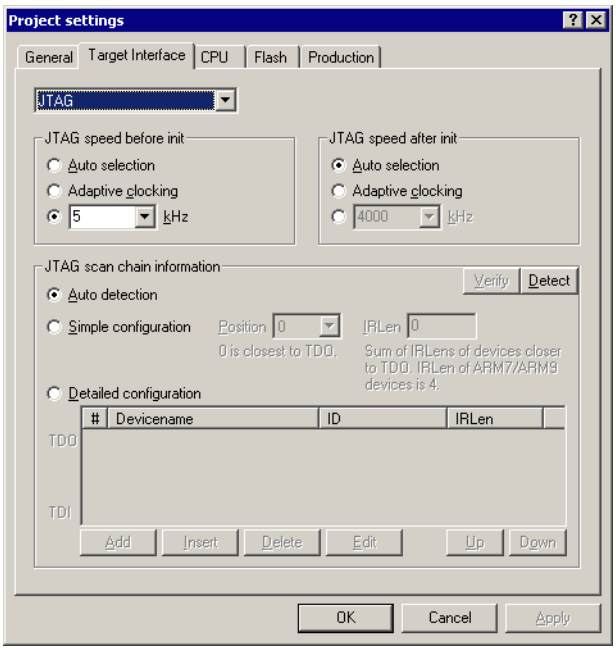
1. Select **File** -> **New Project** to open a new project.
2. Open the **Project Settings** context menu. Select **Options** -> **Project Settings** or press ALT-F7 to open the **Project settings** dialog and select the type of connection to J-Link.



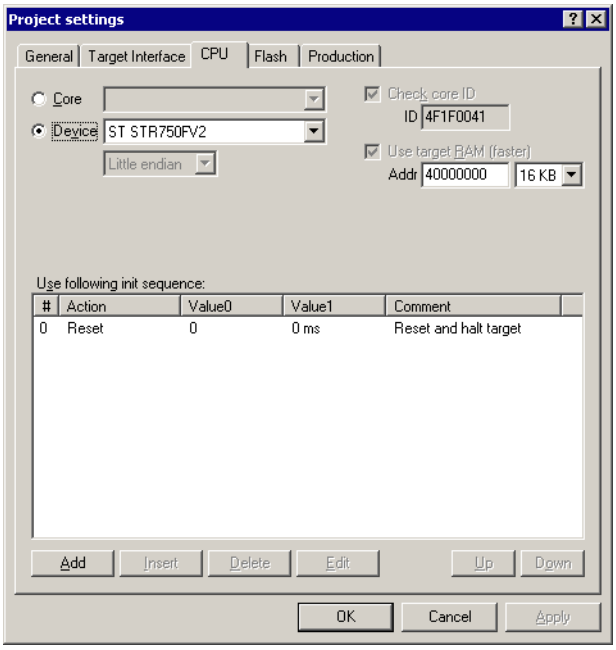
Select **Engineering (More options, typically used for setup)**.

 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

3. Define the **JTAG speed before init** and the **JTAG speed after init**. The default settings work without any problem for the most targets.
   Since software version 3.80 J-Flash supports SWD. To select SWD as target interface, simply select **SWD** from the dropdown box and define the **SWD speed before init** and the **SWD speed after init.**

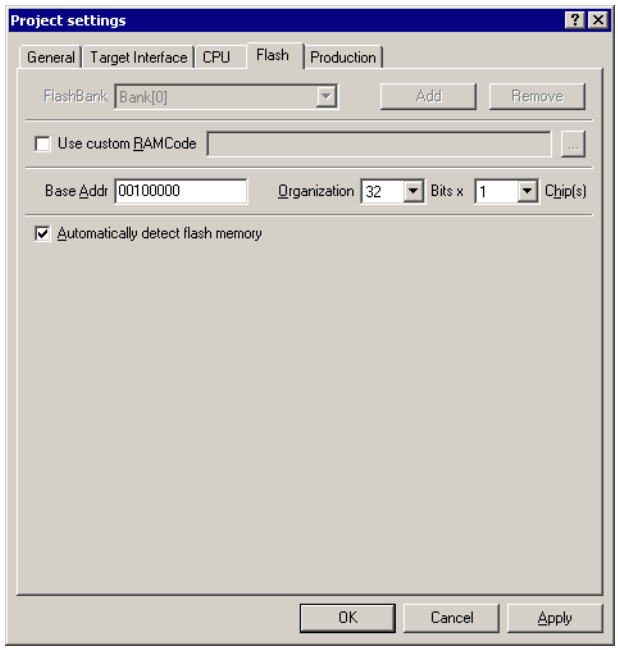


4. Open the **CPU** dialog and select the core architecture in the **Core** choice-list to use J-Flash with an external flash chip. Set the endianess, core ID, RAM address and RAM size of the used MCU.
   To program the internal flash of the chip choose a device from **Device** choice-list. J-Flash uses correct default values (endianess, core ID, RAM address and size) for this device. This is the part where initialization of the external bus interface (if necessary) has to be done. For more information about the valid commands which can be used in an initialization sequence, please refer to *Init sequence* on page 33.

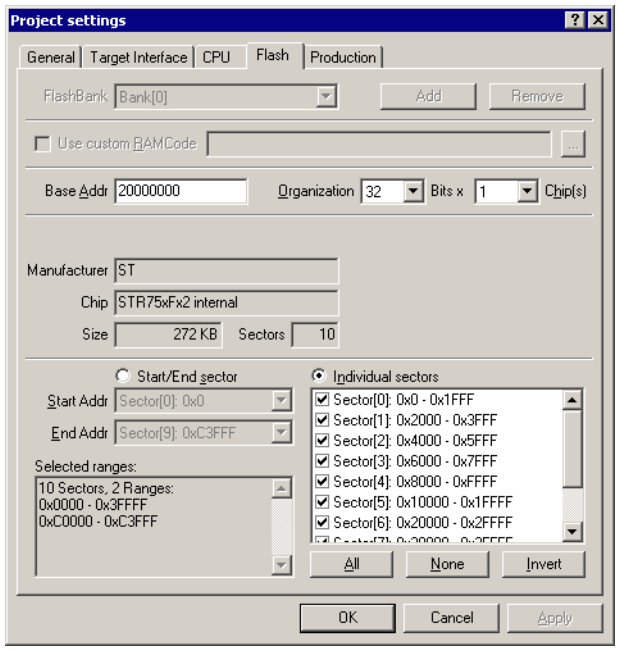


© 2004-2009 SEGGER Microcontroller GmbH & Co. KG

5. The Flash dialog is dependent on selection in the CPU dialog. If you want to program external flash the dialog should look similar to the screenshot below. The used flash chip can be automatically detected or chosen from a list if you disable the Automatic detection checkbox. If you choose the **Automatic detection** feature, the only required settings are the **Base Addr**, **Organization** and **Chip(s)** fields.

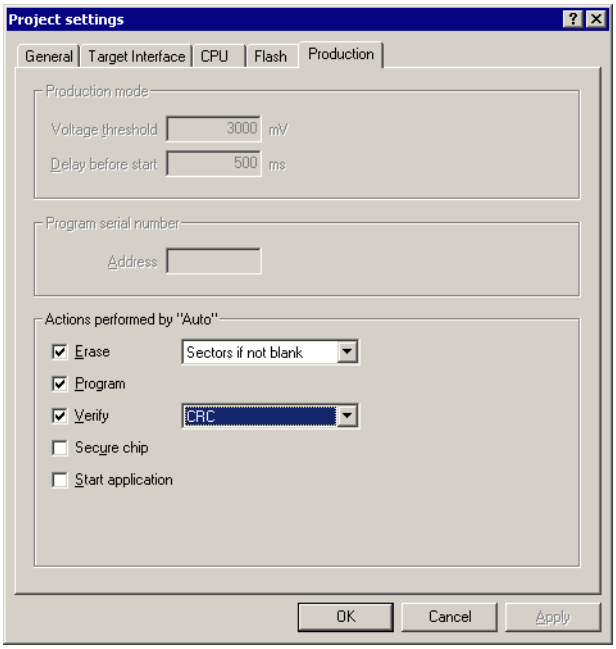


If you want to program the internal flash of an MCU the dialog should look similar to the screenshot below. Normally, all default settings can be used without modifications.



 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

6. In the **Production** dialog is secondary for a setup. You can define the behaviour of the Auto option (**Target** -> **Auto** or shortcut: F7).



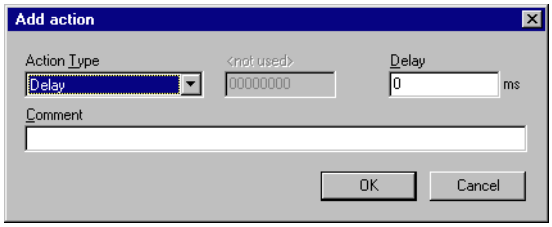
7. Save your project (**File** -> **Save Project**) and test it.

© 2004-2009 SEGGER Microcontroller GmbH & Co. KG

6.2 Creating a new init sequence

Many microcontrollers require a custom init sequence to initialize the target hardware, for example the initialize the PLL, disable the watchdog or define the wait states of the flash. This means that you have to build an compatible init sequence for the microcontroller, if you create a new project or modify one of the existing projects.

You can build or update a custom init sequence in the **CPU** dialog in the **Project settings** menu. Click the **Add** button to open the **Add action** dialog.



In the **Action Type** choice-list all possible commands are listed. The following two textboxes are dependent on the chosen command. They are grayed out or used to enter the required parameter. The **Comment** textbox should be used to enter a short description of the action. For a list of all valid commands which can be used in an init sequence, please refer to *Init sequence* on page 33.

6.2.1 Example init sequence

A good example of a typical init sequence is the init sequence of an AT91SAM7 CPU. The following example is excerpted from the J-Flash project for the AT91SAM7S256.

The example init sequence step by step

0. Reset the target with J-Link reset strategy 0 and 0 delay.
1. Disable the watchdog by writing to the Watchdog Timer Mode Register.
2. Set flash wait states by writing to the MC Flash Mode Register.
3. Set the PLL by writing to power management controller.
4. Set a delay of 200ms.
5. Set the PLL and the divider by writing to PLL Register of the power management controller.
6. Set a delay of 200ms.
7. Set the master and processor clock by writing to the Master Clock Register of the power management controller.

The steps implemented in J-Flash:

 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

Chapter 7

Device specifics

This chapter gives some additional information about specific devices.

 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

7.1 Analog Devices

J-Flash supports flash programming for the Analog Devices ADuC7xxx core family.

7.1.1 ADuC7xxx

J-Flash includes "ready-to-use" projects for all supported devices. For a complete list of supported devices, open J-Flash and check the available projects. If you miss the support of a particular device, do not hesitate to contact Segger.

Refer to *J-Link / J-Trace User Guide* for device specifics which are not related to flash programming.

 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

7.2 ATMEL

J-Flash supports flash programming for the ATMEL AT91SAM7 core family.

7.2.1 AT91SAM7

J-Flash includes "ready-to-use" projects for all supported devices. For a complete list of supported devices, open J-Flash and check the available projects. If you miss the support of a particular device, do not hesitate to contact Segger.

Refer to *J-Link / J-Trace User Guide* for device specifics which are not related to flash programming.

7.2.2 AT91SAM9

J-Flash includes "ready-to-use" projects for all supported devices. For a complete list of supported devices, open J-Flash and check the available projects. If you miss the support of a particular device, do not hesitate to contact Segger.

Refer to *J-Link / J-Trace User Guide* for device specifics which are not related to flash programming.

 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

7.3 NXP

J-Flash supports flash programming for the NXP LPC core family.

7.3.1 LPC2xxx

J-Flash includes "ready-to-use" projects for all supported devices. For a complete list of supported devices, open J-Flash and check the available projects. If you miss the support of a particular device, do not hesitate to contact Segger.

Refer to *J-Link / J-Trace User Guide* for device specifics which are not related to flash programming.

 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

7.4 OKI

J-Flash supports flash programming for the OKI ML67Q40x core family.

7.4.1 ML67Q40x

J-Flash includes "ready-to-use" projects for all supported devices. For a complete list of supported devices, open J-Flash and check the available projects. If you miss the support of a particular device, do not hesitate to contact Segger.

Refer to *J-Link / J-Trace User Guide* for device specifics which are not related to flash programming.

 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

7.5 ST Microelectronics

J-Flash supports flash programming for the ST Microelectronics STR71x, STR73x, STR75x, STR91x and the Cortex-M3 core families.

7.5.1 STM32F10x

7.5.1.1 Securing/Unsecuring the chip

The "Secure Chip" option is available for the STM32 devices. It will read-protect the internal flash memory of the STM32. J-Flash is also able to unsecure a read-protected STM32 device.

Note: Unsecuring a read protected device will cause a mass erase of the flash memory.

7.5.1.2 Option byte programming

J-Flash supports programming of the option bytes for STM32 devices. In order to program the option bytes simply choose the appropriate Device, which allows option byte programming, in the CPU settings tab (e.g. **`STM32F103ZE (allow opt. bytes)`**). J-Flash will allow programming a virtual 16-byte sector at address 0x06000000 which represents the 8 option bytes and their complements. You do not have to care about the option bytes' complements since they are computated automatically. The following table describes the structure of the option bytes sector

Address	[31:24]	[23:16]	[15:8]	[7:0]
0x06000000	complement	Option byte 1	complement	Option byte 0
0x06000004	complement	Option byte 3	complement	Option byte 2
0x06000008	complement	Option byte 5	complement	Option byte 4
0x0600000C	complement	Option byte 7	complement	Option byte 6

Table 7.1: Option bytes sector description

Note: Writing a value of 0xFF inside option byte 0 will read-protect the STM32. In order to keep the device unprotected you have to write the key value 0xA5 into option byte 0.

Note: The address 0x06000000 is a virtual address only. The option bytes are originally located at address 0x1FFFF800. The remap from 0x06000000 to 0x1FFFF800 is done automatically by J-Flash.

Example

To program the option bytes 2 and 3 with the values 0xAA and 0xBB but leave the device unprotected your option byte sector (at addr 0x06000000) should look like as follows:

Address	[31:24]	[23:16]	[15:8]	[7:0]
0x06000000	0x00	0xFF	0x5A	0xA5
0x06000004	0x44	0xBB	0x55	0xAA
0x06000008	0x00	0xFF	0x00	0xFF
0x0600000C	0x00	0xFF	0x00	0xFF

Table 7.2: Option bytes programming example

For a detailed description of each option byte, please refer to *ST programming manual PM0042, section "Option byte description".*

 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

7.5.2 STR 71x

J-Flash includes "ready-to-use" projects for all supported devices. For a complete list of supported devices, open J-Flash and check the available projects. If you miss the support of a particular device, do not hesitate to contact Segger.

Refer to *J-Link / J-Trace User Guide* for device specifics which are not related to flash programming.

7.5.3 STR 73x

J-Flash includes "ready-to-use" projects for all supported devices. For a complete list of supported devices, open J-Flash and check the available projects. If you miss the support of a particular device, do not hesitate to contact Segger.

Refer to *J-Link / J-Trace User Guide* for device specifics which are not related to flash programming.

7.5.4 STR 75x

J-Flash includes "ready-to-use" projects for all supported devices. For a complete list of supported devices, open J-Flash and check the available projects. If you miss the support of a particular device, do not hesitate to contact Segger.

Refer to *J-Link / J-Trace User Guide* for device specifics which are not related to flash programming.

7.5.5 STR91x

J-Flash includes "ready-to-use" projects for all supported devices. For a complete list of supported devices, open J-Flash and check the available projects. If you miss the support of a particular device, do not hesitate to contact Segger.

Refer to *J-Link / J-Trace User Guide* for device specifics which are not related to flash programming.

 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

7.6 Texas Instruments

J-Flash supports flash programming for the TI TMS470 core family.

7.6.1 TMS470

J-Flash includes "ready-to-use" projects for all supported devices. For a complete list of supported devices, open J-Link RDI configuration dialog and check the device list of the Flash programming tab (refer to Flash configuration on page 38 for detailed information). If you miss the support of a particular device, do not hesitate to contact Segger.

Refer to J-Link / J-Trace User Guide for device specifics which are not related to flash programming.

 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

Chapter 8

Target systems

The following chapter lists all supported flash devices.

 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

8.1 Which devices can be programmed by J-Flash?

J-Flash can program external as well as internal flash. Any combination of ARM CPU and external flash is supported if the flash chip is listed in section *Supported Flash Devices* on page 64. Beside the listed flash chips is every CFI compliant chip supported. In addition, all types of flash interfacing are supported: 1x8bit, 2x8bit, 4x8bit, 1x16bit, 2x16bit, 1x32bit.

Regarding internal flash, J-Flash supports a wide range of microcontrollers. The next section lists all supported micros.

If you need support for a chip or flash not listed in the tables below, do not hesitate to contact us. Segger is constantly adding support for new devices. You may want to request an updated list or have a look at *http://www.segger.com* for more up to date information.

 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

8.2 Supported Microcontrollers

For a list of all microcontrollers which are currently supported by J-Flash, please refer to the J-Link / J-Trace User Manual (UM08001). Chapter *Flash download and flash breakpoints*, section *Supported devices*.

 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

8.3 Supported Flash Devices

Manufacturer	Name
AMD	Am29DL161DB
AMD	Am29DL161DT
AMD	Am29DL162DB
AMD	Am29DL162DT
AMD	Am29DL163DB
AMD	Am29DL163DT
AMD	Am29DL164DB
AMD	Am29DL164DT
AMD	Am29DL322DB/GB
AMD	Am29DL322DT/GT
AMD	Am29DL323DB/GB
AMD	Am29DL323DT/GT
AMD	Am29DL324DB/GB
AMD	Am29DL324DT/GT
AMD	Am29DL400BB
AMD	Am29DL400BT
AMD	Am29DL800BB
AMD	Am29DL800BT
AMD	Am29DS323DB
AMD	Am29DS323DT
AMD	Am29F100B
AMD	Am29F100T
AMD	Am29F400BB
AMD	Am29F400BT
AMD	Am29F800BB
AMD	Am29F800BT
AMD	Am29LV001BB
AMD	Am29LV001BT
AMD	Am29LV002BB
AMD	Am29LV002BT
AMD	Am29LV004BB
AMD	Am29LV004BT
AMD	Am29LV033C
AMD	Am29LV033MU
AMD	Am29LV116DB
AMD	Am29LV116DT
AMD	Am29LV160BB
AMD	Am29LV160BT
AMD	Am29LV160DB
AMD	Am29LV160DT
AMD	Am29LV200BB
AMD	Am29LV200BT
AMD	Am29LV320DB
AMD	Am29LV320DT
AMD	Am29LV400BB
AMD	Am29LV400BT
AMD	Am29LV640D
AMD	Am29LV641D

Table 8.1: List of supported flash devices

 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

Manufacturer	Name
AMD	Am29LV800BB
AMD	Am29LV800BT
AMD	Am29SL800DB
AMD	Am29SL800DT
AMIC	A29L400B
AMIC	A29L400T
Atmel	AT29BV010A
Atmel	AT29BV020
Atmel	AT29BV040
Atmel	AT29BV040A
Atmel	AT29C010A
Atmel	AT29C020
Atmel	AT29C040
Atmel	AT29C040A
Atmel	AT29C1024
Atmel	AT29C256
Atmel	AT29C257
Atmel	AT29C512
Atmel	AT29LV010A
Atmel	AT29LV020
Atmel	AT29LV040
Atmel	AT29LV040A
Atmel	AT29LV1024
Atmel	AT29LV256
Atmel	AT29LV512
Atmel	AT49BN6416
Atmel	AT49BN6416T
Atmel	AT49BV001A
Atmel	AT49BV001AN
Atmel	AT49BV001ANT
Atmel	AT49BV001AT
Atmel	AT49BV002
Atmel	AT49BV002A
Atmel	AT49BV002AN
Atmel	AT49BV002ANT
Atmel	AT49BV002AT
Atmel	AT49BV002N
Atmel	AT49BV002NT
Atmel	AT49BV002T
Atmel	AT49BV040A
Atmel	AT49BV1024A
Atmel	AT49BV1604
Atmel	AT49BV1604A
Atmel	AT49BV1604AT
Atmel	AT49BV1604T
Atmel	AT49BV160C
Atmel	AT49BV160CT
Atmel	AT49BV160D
Atmel	AT49BV160DT
Atmel	AT49BV1614

Table 8.1: List of supported flash devices

 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

Manufacturer	Name
Atmel	AT49BV1614A
Atmel	AT49BV1614AT
Atmel	AT49BV1614T
Atmel	AT49BV162A
Atmel	AT49BV162AT
Atmel	AT49BV163D
Atmel	AT49BV163DT
Atmel	AT49BV2048A
Atmel	AT49BV320C
Atmel	AT49BV320CT
Atmel	AT49BV320D
Atmel	AT49BV320DT
Atmel	AT49BV322A
Atmel	AT49BV322AT
Atmel	AT49BV322D
Atmel	AT49BV322DT
Atmel	AT49BV4096A
Atmel	AT49BV512
Atmel	AT49BV640
Atmel	AT49BV640D
Atmel	AT49BV640DT
Atmel	AT49BV640T
Atmel	AT49BV6416
Atmel	AT49BV6416T
Atmel	AT49BV642D
Atmel	AT49BV642DT
Atmel	AT49BV802A
Atmel	AT49BV802AT
Atmel	AT49F001A
Atmel	AT49F001AN
Atmel	AT49F001ANT
Atmel	AT49F001AT
Atmel	AT49F002A
Atmel	AT49F002AN
Atmel	AT49F002ANT
Atmel	AT49F002AT
Atmel	AT49F040A
Atmel	AT49F1024
Atmel	AT49F1024A
Atmel	AT49F1025
Atmel	AT49F2048A
Atmel	AT49F4096A
Atmel	AT49F512
Atmel	AT49LV002
Atmel	AT49LV002N
Atmel	AT49LV002NT
Atmel	AT49LV002T
Atmel	AT49LV1024
Atmel	AT49LV1024A
Atmel	AT49LV1614A

Table 8.1: List of supported flash devices

 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

Manufacturer	Name
Atmel	AT49LV1614AT
Atmel	AT49LV2048A
Atmel	AT49LV4096A
Atmel	AT49SN3208
Atmel	AT49SN3208T
Atmel	AT49SN6416
Atmel	AT49SN6416T
Atmel	AT49SV322A
Atmel	AT49SV322AT
Atmel	AT49SV802A
Eon	EN29LV800BT
Atmel	AT49SV802AT
Cutera	LH28F640BFHE-PBTL (x2)
Cutera	LH28F128BFHED
Fujitsu	MBM29DL322BE/BD
Fujitsu	MBM29DL322TE/TD
Fujitsu	MBM29LV650U
Fujitsu	MBM29LV160B
Fujitsu	MBM29LV160T
Intel	28F004B3B
Intel	28F004B3T
Intel	28F008B3B
Intel	28F008B3T
Intel	28F016B3B
Intel	28F016B3T
Intel	28F128J3
Intel	28F128K18
Intel	28F128K3
Intel	28F128P30B
Intel	28F128P30T
Intel	28F128P33B
Intel	28F128P33T
Intel	28F128W18B
Intel	28F128W18T
Intel	28F160B3B
Intel	28F160B3T
Intel	28F160C3B
Intel	28F160C3T
Intel	28F256J3
Intel	28F256K18
Intel	28F256K3
Intel	28F256P30B
Intel	28F256P30T
Intel	28F256P33B
Intel	28F256P33T
Intel	28F320B3B
Intel	28F320B3T
Intel	28F320C3B
Intel	28F320C3T
Intel	28F320J3

Table 8.1: List of supported flash devices

 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

Manufacturer	Name
Intel	28F320W18B
Intel	28F320W18T
Intel	28F400B3B
Intel	28F400B3T
Intel	28F640B3B
Intel	28F640B3T
Intel	28F640C3B
Intel	28F640C3T
Intel	28F640J3
Intel	28F640K18
Intel	28F640K3
Intel	28F640P30B
Intel	28F640P30T
Intel	28F640P33B
Intel	28F640P33T
Intel	28F640W18B
Intel	28F640W18T
Intel	28F800B3B
Intel	28F800B3T
Intel	28F800C3B
Intel	28F800C3T
Macronix	MX29LV160CB
Macronix	MX29LV160CT
Macronix	MX29LV320AB
Macronix	MX29LV320AT
Macronix	MX29LV400CB
Macronix	MX29LV400CT
Macronix	MX29LV800CB
Macronix	MX29LV800CT
Sharp	LH28F128BFHED
Sharp	LH28F128BFHT
Sharp	LH28F128SPHTD
Sharp	LH28F640BFHE-PBTL
Sharp	LH28F640BFHE-PTTL
Sharp	LH28F640BFHG-PBTL
Sharp	LH28F640BFHG-PTTL
Sharp	LHF00L29
Sirona	Sirona 2x16
Spansion	S29AL008Dxxxxx01
Spansion	S29AL008Dxxxxx02
Spansion	S29AL016Dxxxxx01
Spansion	S29AL016Dxxxxx02
Spansion	S29AL032Dxxxxx00
Spansion	S29AL032Dxxxxx03
Spansion	S29AL032Dxxxxx04
Spansion	S29JL032Hxxxxxx1
Spansion	S29JL032Hxxxxxx2
Spansion	S29GL032AxR1
Spansion	S29GL032AxR2
Spansion	S29GL032AxR3

Table 8.1: List of supported flash devices

 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

Manufacturer	Name
Spansion	S29GL032AxR4
Spansion	S29GL032AxW3
Spansion	S29GL032AxW4
Spansion	S29GL032MxR0
Spansion	S29GL032MxR1
Spansion	S29GL032MxR2
Spansion	S29GL032MxR3
Spansion	S29GL032MxR4
Spansion	S29GL032MxR5
Spansion	S29GL032MxR6
Spansion	S29GL064MxR0
Spansion	S29GL064MxR1
Spansion	S29GL064MxR2
Spansion	S29GL064MxR3
Spansion	S29GL064MxR4
Spansion	S29GL064MxR5
Spansion	S29GL064MxR6
Spansion	S29GL064MxR7
Spansion	S29GL064MxR8
Spansion	S29GL064MxR9
Spansion	S29GL128M
Spansion	S29GL128N
Spansion	S29GL128P
Spansion	S29GL256M
Spansion	S29GL256N
Spansion	S29GL256P
Spansion	S29GL512N
Spansion	S29GL512P
Spansion	S29JL032Hxxxxxx1
Spansion	S29JL032Hxxxxxx2
Spansion	S29WS064J
Spansion	S29WS128J
Spansion	S29WS128N
Spansion	S29WS256N
Spansion	S71PL032J
Spansion	S71PL064J
Spansion	S71PL127J
SST	SST39LF200A
SST	SST39LF400A
SST	SST39LF800A
SST	SST39LF160
SST	SST39VF160
SST	SST39VF1601
SST	SST39VF1602
SST	SST39VF200A
SST	SST39VF3201
SST	SST39VF3202
SST	SST39VF400A
SST	SST39VF6401
SST	SST39VF6401B

Table 8.1: List of supported flash devices

 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

Manufacturer	Name
SST	SST39VF6402
SST	SST39VF6402B
SST	SST39VF800A
ST	M28W320FCB
ST	M28W320FCT
ST	M28W320FSB
ST	M28W320FST
ST	M28W640ECB
ST	M28W640ECT
ST	M28W640FCB
ST	M28W640FCT
ST	M28W640FSB
ST	M28W640FST
ST	M29DW128F
ST	M29DW323DB
ST	M29DW323DT
ST	M29DW324DB
ST	M29DW324DT
ST	M29DW640D
ST	M29DW641F
ST	M29W160DB
ST	M29W160DT
ST	M29W160EB
ST	M29W160ET
ST	M29W200BB
ST	M29W200BT
ST	M29W320DB
ST	M29W320DT
ST	M29W400BB
ST	M29W400BT
ST	M29W400DB
ST	M29W400DT
ST	M29W640DB
ST	M29W640DT
ST	M29W640FB
ST	M29W640FT
ST	M29W800DB
ST	M29W800DT
ST	M58LW064D
Toshiba	TC58FVB160
Toshiba	TC58FVT160

Table 8.1: List of supported flash devices

 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

Chapter 9

Performance

The following chapter lists programming performance of common flash devices and microcontrollers.

 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

9.1 Performance of MCUs with internal flash memory

The following table lists program and erase performance values for different controllers.

Microcontroller	Size [kByte]	Program time [sec]	Program speed [kB/sec]	Erase Time [sec]	Erase speed [kB/sec]
Analog Devices ADuC7020	62	2.234	27.752	3.031	20.455
Atmel AT91SAM7S64	64	3.235	19.783	- Not required	
Atmel AT91SAM7S256	256	6.734	38.016	- Not required	
NXP LPC2148	500	3.953	126.486	12.312	40.610
NXP LPC2138	500	3.906	128.008	12.312	40.610
NXP LPC2129 V1	248	1.828	135.667	7.812	31.746
NXP LPC2106	120	0.948	126.582	6.875	17.454
NXP LPC2129 V2	248	1.797	138.007	7.750	32.000
NXP LPC2294	248	1.875	132.266	7.812	31.746
ST STR711	272	4.890	55.623	9.703	28.032
ST STR912	512	7.000	73.142	9.375	54.613
TI TMS470R1B1M	1024	10.953	93.490	18.359	55.776

Table 9.1: List of performance values of MCUs with internal flash

 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

9.2 Performance of MCUs with external flash memory

Hardware	Flash device	Organization	Speed
Atmel AT91EB40	Atmel AT49BV162A	1*16 Bits	105.025 kB/s
Cogent CSB337	Intel 28F640J3	1*16 Bits	93.058 kB/s
NetSilicon NS9360	AMD Am29LV160DB	2*16 Bits	185.171 kB/s
Logic LH7A400	Intel 28F640J3A120	2*16 Bits	154.978 kB/s

Table 9.2: List of performance values of MCUs with external flash

© 2004-2009 SEGGER Microcontroller GmbH & Co. KG

 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

Chapter 10

Support

The following chapter provides information about how to contact our support.

 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

10.1 Troubleshooting

10.1.1 General procedure

- Make sure your J-Link is working as expected. See the troubleshooting section in the J-Link manual.
- Ensure that the target hardware matches the project file settings. Pay special attention to the following aspects:
  - Init sequence
  - Clock speed
  - RAM address
  - Flash base address
  - MCU / Flash chip
  - Flash organization
- Try to program your target device using a sample project file if available. J-Flash ships with an extensive number of project files for many target boards. See section *Sample Projects* on page 22 for a complete list of project files.
- The JTAG clock frequency depends on several factors, e.g. cable length, target board etc. Try setting the frequency to lower or higher values accordingly.
- Make sure the flash memory is unlocked before programming or erasing.

10.1.2 Typical problems

Failed to connect

Meaning:

This error message is shown if any error occurs during the connection process.

Remedy:

First of all, make sure the target is actually connected to J-Link. Verify the correctness of the init sequence, check the JTAG speed, and ensure the correct flash type is selected.

Programming / Erasing failed

Meaning:

The flash memory sector may be locked and programming or erasing the respective memory section fails therefore.

Remedy:

Make sure the memory sector is unlocked before programming or erasing. J-Flash provides a dedicated menu item for unlocking flash memory.

Timeout errors during programming

Meaning:

A timeout occurs if the target is too slow during DCC communication or the target flash memory is too slow during programming.

Remedy:

Using smaller RAM block sizes may fix this problem.

Blank check failed

Meaning:

The target memory was not empty during blank check.

Remedy:

Erase target memory.

 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

RAM check failed

Meaning:

No RAM found at the specified RAM location.

Remedy:

Make sure a correct RAM address is specified in the project settings. See section *CPU Settings* on page 31.

Unexpected core ID

Meaning:

The specified CPU core ID does not match with the one read from the target CPU.

Remedy:

Ensure the specified core ID is correct for the used target CPU. See section *CPU Settings* on page 31 for information about setting the core ID.

Unsupported flash type / bus width

Meaning:

The target flash memory or the bus organization is not yet supported.

Remedy:

Inform us about the flash type you want to use. SEGGER is constantly adding support for new flash memory devices.

No matching RAMCode

Meaning:

There is no programming algorithm available for the selected target memory type.

Remedy:

Inform us about the flash type you want to use. SEGGER is constantly adding support for new flash memory devices.

 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

10.2 Contacting support

If you experience a J-Flash related problem and the advices from the sections above do not help you to solve it, you may contact our J-Flash support. In this case, please provide us with the following information:

- A detailed description of the problem.
- The relevant log file and project file. In order to generate an expressive log file, set the log level to "All messages" (see section *Global Settings* on page 38 for information about changing the log level in J-Flash).
- The relevant data file as a .hex or .mot file (if possible)
- The processor and flash types used

Once we received this information we will try our best to solve the problem for you. Our contact address is as follows:

SEGGER Microcontroller GmbH & Co. KG

Heinrich-Hertz-Str. 5
D-40721 Hilden

Germany

Tel.+49 2103-2878-0
Fax.+49 2103-2878-28
Email: support@segger.com
Internet: http://www.segger.com

 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG

Index

F

Flash, supported interfacing types 62

J

J-Link 12
JTAG 12, 30

M

Menu structure 24
Microcontrollers 63, 72–73

P

Performance 71
Projects 22

S

Supported flash devices
- AMD 64
- AMIC 65
- ATMEL 65
- Fujitsu 67
- Intel 67
- Macronix 68
- Sharp 68
- Sirona 68
- Spansion 68
- SST 69
- ST 70
- Toshiba 70

Supported Microcontrollers 63
Syntax, conventions used 5

T

TCP/IP 29

U

USB 29

© 2004-2009 SEGGER Microcontroller GmbH & Co. KG

 © 2004-2009 SEGGER Microcontroller GmbH & Co. KG