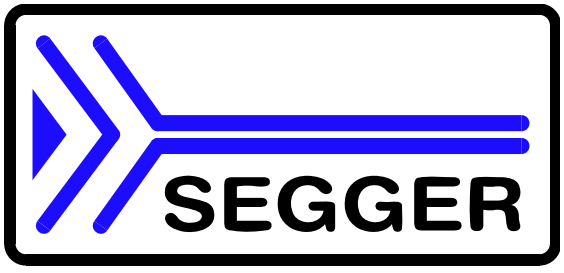
J-Flash ARM

Stand-alone Flash programming software

Version 3.68
Manual Rev. 1



A product of SEGGER Microcontroller Systeme GmbH

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 SYSTEME GmbH (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

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 Systeme GmbH

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

Manual versions

This manual describes the latest software version. The version number of the software can be found in the table 'Software versions' later in this chapter. If any error occurs, please inform us and we will assist you.

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

Manual version	Date	By	Explanation
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	O O	Chapter "Performance" updated.
3.46 Rev. 2	061121	O O	Chapter "Performance" updated.
3.46 Rev. 1	060929	TQ	Update supported target devices.
3.42 Rev. 1	060912	TQ	Update supported target devices.

Manual version	Date	By	Explanation
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

Software versions

Changes in the software are listed in the file "Release.html" shipped with the software.

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 and macros that emFile 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

SEGGER Microcontroller Systeme GmbH 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 of Contents

Chapter 1

Introduction

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

1.1 What is J-Flash?

J-Flash is a stand-alone flash programming software for PCs running Microsoft Windows. It 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 core supported, including thumb mode.
- ARM microcontroller (internal flash) support.
- Support for most external flash chips (see chapter “Target systems” on page 45 for a list of supported devices).
- High speed programming: up to 200 kB/sec* (depending on flash device).
- Very high speed blank check: approximately 16 MB/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

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.

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 Server from a remote system.

Chapter 2

Installation of J-Flash

The following chapter describes how to successfully install J-Flash on your host system.

2.1 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.

2.1.1 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 17 for further details).

Chapter 3

Getting Started

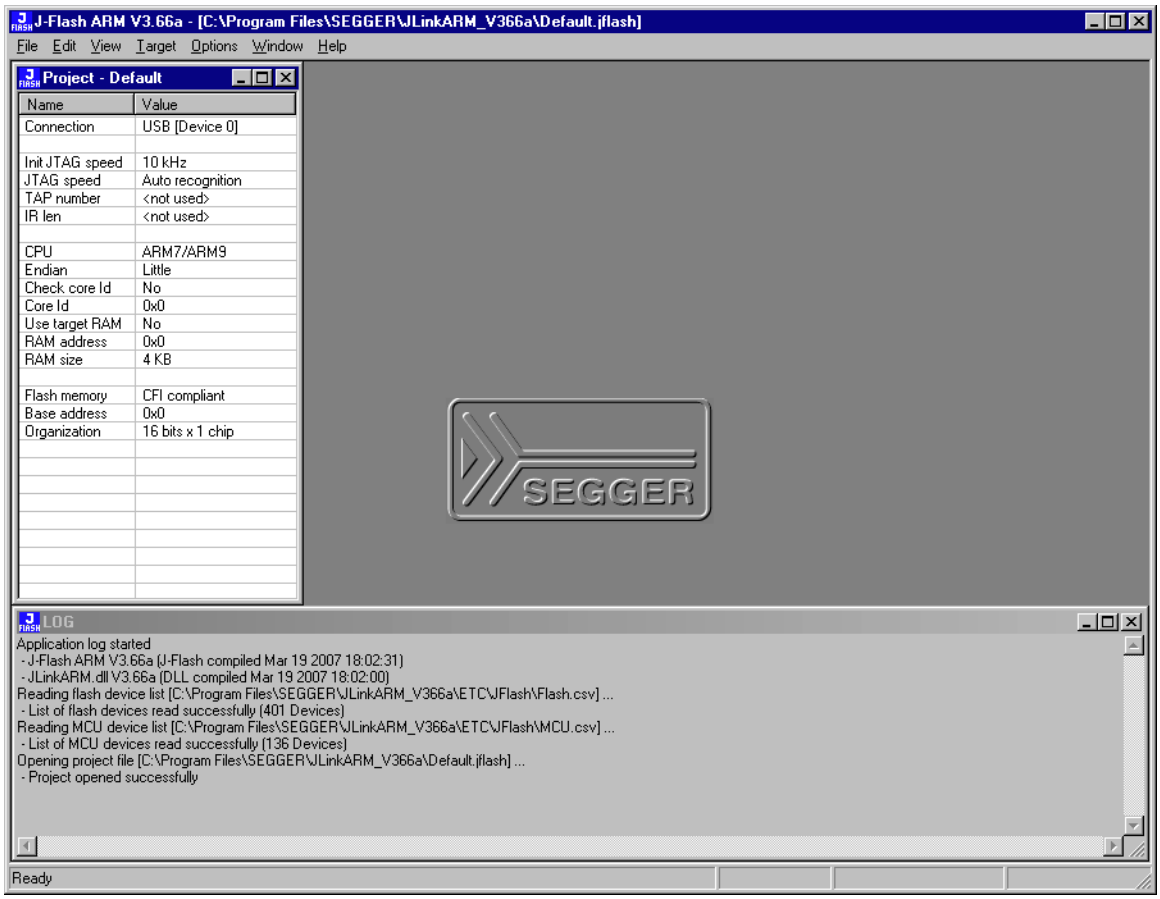
This chapter presents an introduction to J-Flash. It provides an overview of the included sample projects and describes J-Flash's menu structure in detail.

3.1 Using J-Flash for the First Time

Start J-Flash from the Windows Start menu. J-Flash's main window will apear, 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).



J-Flash main window (as of version 2.00).

3.1.1 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
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
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

Project	Description
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	
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
TMS470R1B1M.jflash	TI TMS470R1B1M with internal flash memory
TMS470R1VF689.jflash	TI TMS470R1VF689 with internal flash memory

3.2 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.

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.

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.

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.
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 23 for further details.
Test >	Two test functions are implementet "Generates test data" generates data which can be used to test if the flash can be programmed correctly. The size of the gerated 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.
Verify	Verifies the data found on the chip with the data file.

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.

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.

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.

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.

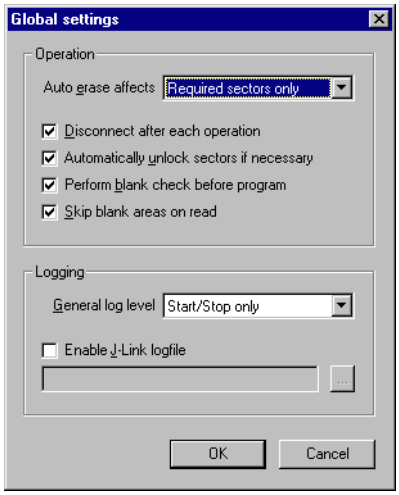
Chapter 4

Settings

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

4.1 Global Settings

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



4.1.1 Operation

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

Auto erase

You can specify if an automatically performed erasure during any program operation is restricted to required sectors, selected sectors or not restricted at all. In the latter case all sectors are erased.

Disconnect after each operation

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

Automatically unlock sectors

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

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.

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.

4.1.2 Logging

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

General log level

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

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.

4.2 Project Settings

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

4.2.1 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. Please refer to the J-Link manual for more information regarding the operation of J-Link and J-Link TCP/IP Server.

USB

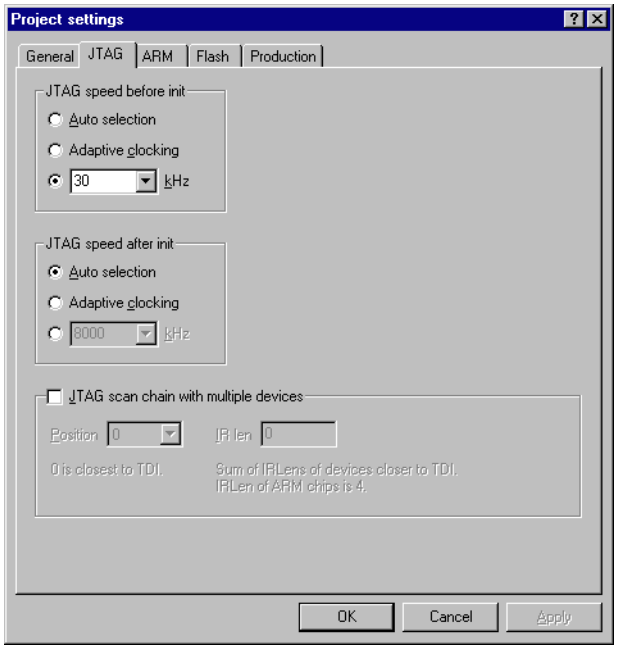
If this option is checked, J-Flash will connect to J-Link over the USB port. You may change the device number if you want to connect more than one J-Link to your PC. The default device number is 0. For more information about how to use multiple J-Links on one PC, please see also the chapter "Working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

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.

4.2.2 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.



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 chapter “ARM Settings” on page 27). 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.

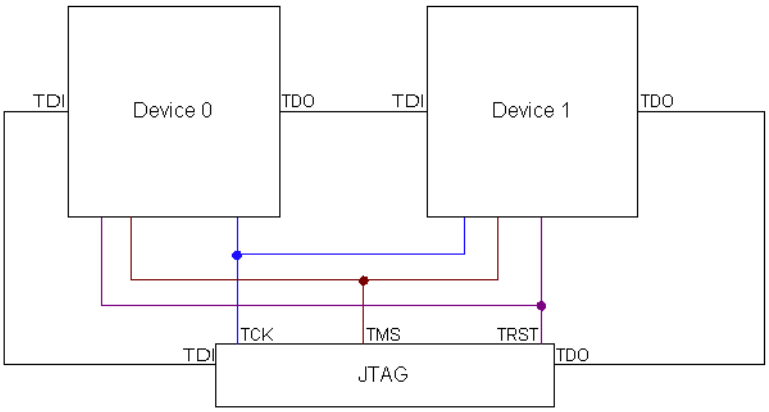
You can choose between automatic speed recognition, adaptive clocking or fixed JTAG speed. If you choose fixed JTAG speed you can select any value between 1kHz and 12MHz.

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

JTAG scan chain with multiple devices

This checkbox allows you to configure a JTAG scan chain with multiple devices on it.

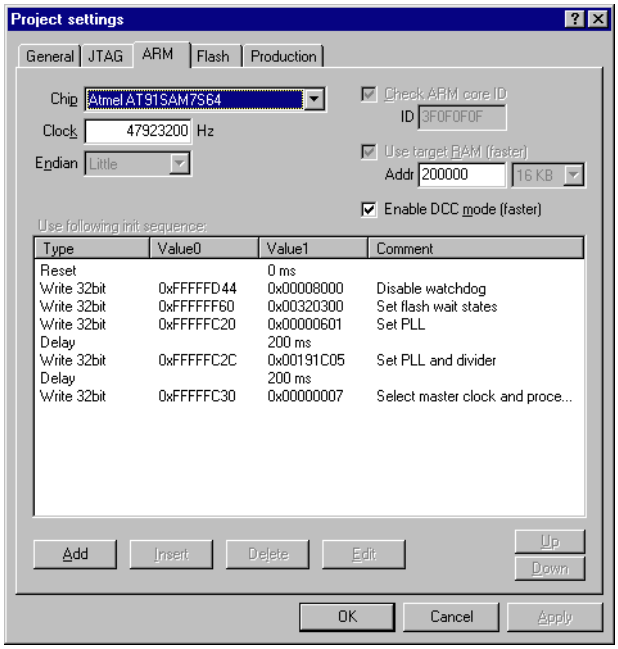
In a scan chain configuration with multiple devices, the TCK and TMS lines of all JTAG device are connected, while the TDI and TDO lines form a ring.



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.

4.2.3 ARM Settings

This dialog allows the selection of microcontroller dependent settings.



Chip

J-Flash can be used to program both external or internal flash memory. In order to use J-Flash with an external flash device, "Generic ARM7/ARM9" must be selected.

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

Clock

In order to guarantee accurate operation of J-Flash you have to enter the correct clock frequency in Hz of your MCU. If you set up a PLL or otherwise change the clock frequency in the init sequence please take into account that you also have to modify the value in this dialog.

Endian

The endianness of the chip is indicated through the Endian drop-down menu.

Check ARM 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.

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.

Enable DCC mode

DCC mode encompasses those features of halt mode and run mode debugging that in most instances facilitate quicker communication. Consequently enabling DCC mode results in improved performance. It is therefore suggested that DCC mode is enabled unless there are communication difficulties.

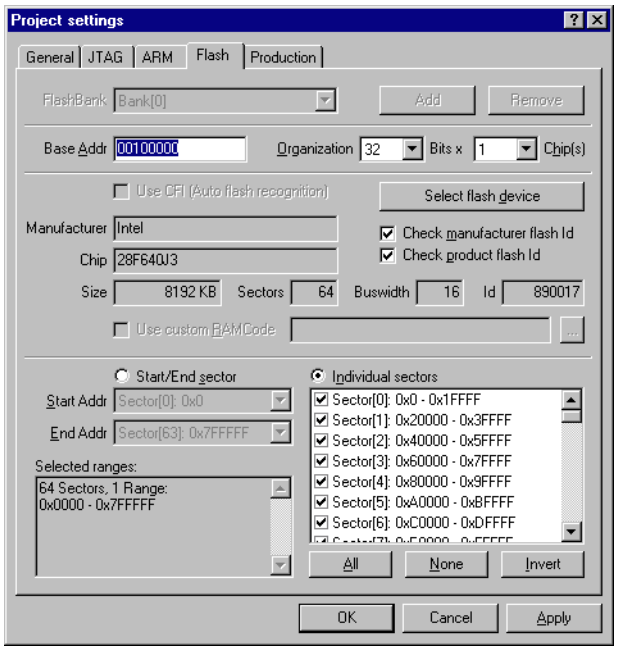
Init sequence

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 lets the user 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.

4.2.4 Flash Settings

This dialog is used to select and configure the flash device to operate with.



Base Address

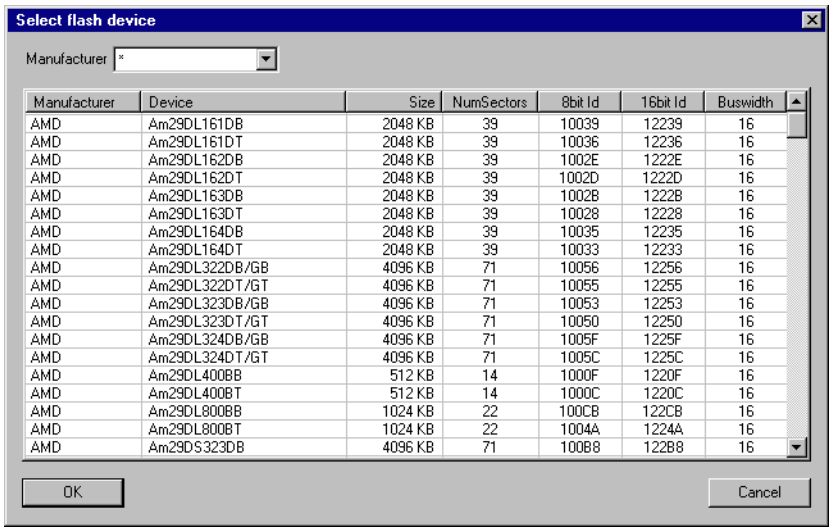
You may enter the base address of the selected flash memory. The default value is 0.

Organization

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

Select flash device

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.



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.

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.

Chapter 5

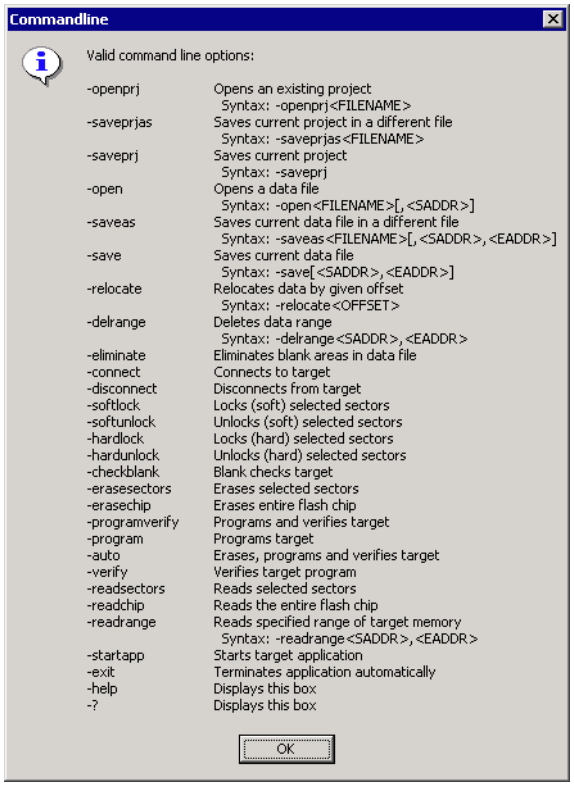
Command Line Interface

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

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 -?`



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.
-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.
-erasesectors	Erase selected sectors.
-erasechip	Erase the entire flash chip.
-programverify	Program and verify target.
-program	Program target.
-auto	Erase, program and verify target.
-readsectors	Read selected sectors.
-readchip	Read entire flash chip.
-readrange<SADDR>,<EADDR>	Read specified range of target memory.
-startapp	Start target application.
-exit	Exit J-Flash.
-help	Display help dialog.
-?	Display help dialog.

5.2.1 Examples

Open a project and data file, start auto processing and exit

```
JFlashARM.exe -openprjC:\Projects\Default.jflash -openC:\Data\data.bin,0x100000
-auto -exit
```

Open a project file, read back selected sectors and write the data to disk

```
JFlashARM.exe -openprjC:\Projects\Default.jflash -readsectors
-saveasC:\Data\data.bin,0x100000,0x10FFFF
```


5.2.2 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.

Chapter 6

Licensing

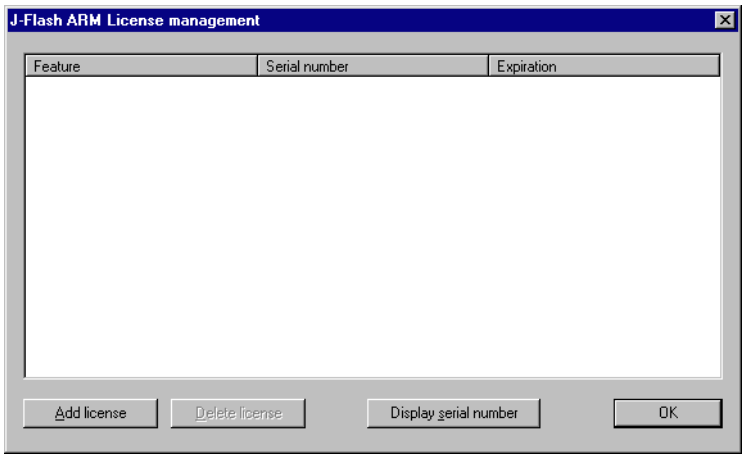
The following chapter provides an overview of J-Flash related licensing options.

6.1 General information about Licensing

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. If you need a license key you only have to tell us the serial number of your J-Link which allows us to send you a proper key. Evaluation licenses which allow you to unlock the full potential of J-Flash for a limited period of time are available. 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.

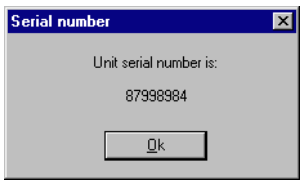
6.2 The licensing dialog

The licensing dialog will be displayed after selecting Licenses... from the Help menu of the main window. It shows the available licenses and allows to add and remove licenses as well.



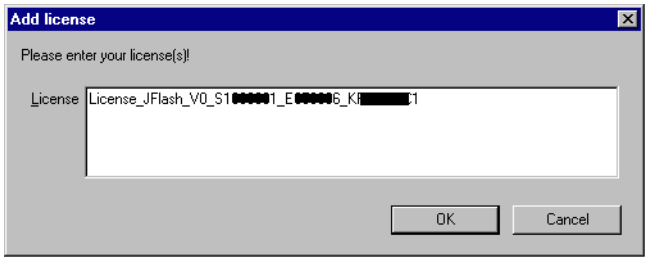
6.2.1 The serial number

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.

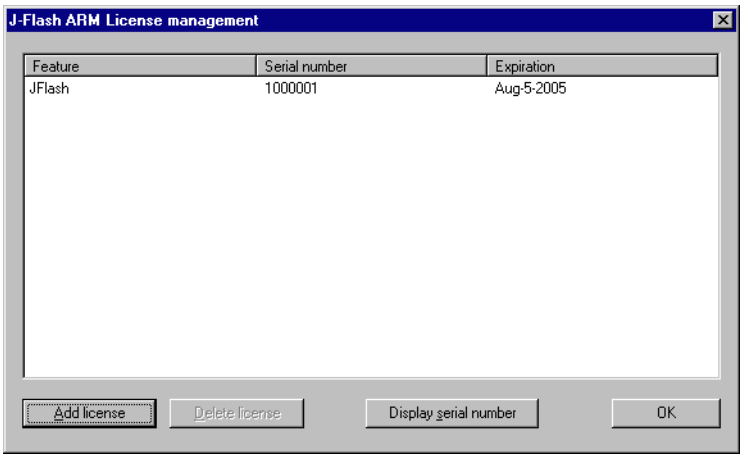


6.2.2 License management

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.



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.

Chapter 7

Support

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

7.1 Troubleshooting

7.1.1 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 17 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.

7.1.2 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.

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 "ARM Settings" on page 27.

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 "ARM Settings" on page 27 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.

7.2 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 24 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 Systeme GmbH

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

Chapter 8

Target systems

The following chapter lists all supported flash devices and microcontrollers.

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 50. 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 *www.segger.com* for more up to date information.

8.2 Supported Microcontrollers

Manufacturer	Name
Analog Devices	ADuC7020x62 (to E)
Analog Devices	ADuC7020x62 (G on)
Analog Devices	ADuC7021x32 (to E)
Analog Devices	ADuC7021x32 (G on)
Analog Devices	ADuC7021x62 (to E)
Analog Devices	ADuC7021x62 (G on)
Analog Devices	ADuC7022x32 (to E)
Analog Devices	ADuC7022x32 (G on)
Analog Devices	ADuC7022x62 (to E)
Analog Devices	ADuC7022x62 (G on)
Analog Devices	ADuC7024x62 (to E)
Analog Devices	ADuC7024x62 (G on)
Analog Devices	ADuC7025x62 (to E)
Analog Devices	ADuC7025x62 (G on)
Analog Devices	ADuC7025x32 (to E)
Analog Devices	ADuC7025x32 (G on)
Analog Devices	ADuC7026x62 (to E)
Analog Devices	ADuC7026x62 (G on)
Analog Devices	ADuC7027x62 (to E)
Analog Devices	ADuC7027x62 (G on)
Analog Devices	ADuC7030
Analog Devices	ADuC7031
Analog Devices	ADuC7032
Analog Devices	ADuC7033
Analog Devices	ADuC7128
Analog Devices	ADuC7129
Analog Devices	ADuC7229x126
Atmel	AT91SAM7A3
Atmel	AT91SAM7S32
Atmel	AT91SAM7S321
Atmel	AT91SAM7S64
Atmel	AT91SAM7S128
Atmel	AT91SAM7S256
Atmel	AT91SAM7S512
Atmel	AT91SAM7SE32
Atmel	AT91SAM7SE256
Atmel	AT91SAM7SE512
Atmel	AT91SAM7X128
Atmel	AT91SAM7X256
Freescale	MAC7101
Freescale	MAC7106
Freescale	MAC7111
Freescale	MAC7112
Freescale	MAC7116
Freescale	MAC7121
Freescale	MAC7122
Freescale	MAC7126
Freescale	MAC7131
Freescale	MAC7136

Manufacturer	Name
Freescale	MAC7141
Freescale	MAC7142
OKI	ML67Q4050
OKI	ML67Q4051
OKI	ML67Q4060
OKI	ML67Q4061
NXP	LPC2101
NXP	LPC2102
NXP	LPC2103
NXP	LPC2104
NXP	LPC2105
NXP	LPC2106
NXP	LPC2114
NXP	LPC2119
NXP	LPC2124
NXP	LPC2129
NXP	LPC2131
NXP	LPC2132
NXP	LPC2134
NXP	LPC2136
NXP	LPC2138
NXP	LPC2141
NXP	LPC2142
NXP	LPC2144
NXP	LPC2146
NXP	LPC2148
NXP	LPC2194
NXP	LPC2212
NXP	LPC2214
NXP	LPC2292
NXP	LPC2294
NXP	LPC2364
NXP	LPC2366
NXP	LPC2368
NXP	LPC2378
NXP	PCF87750
NXP	SJA2010
NXP	SJA2510
Samsung	S3F445HX
ST	STR710FZ1
ST	STR710FZ2
ST	STR711FR0
ST	STR711FR1
ST	STR711FR2
ST	STR712FR0
ST	STR712FR1
ST	STR712FR2
ST	STR715FR0
ST	STR730FZ1
ST	STR730FZ2
ST	STR731FV0

Manufacturer	Name
ST	STR731FV1
ST	STR731FV2
ST	STR735FZ1
ST	STR735FZ2
ST	STR736FV0
ST	STR736FV1
ST	STR736FV2
ST	STR750FV0
ST	STR750FV1
ST	STR750FV2
ST	STR751FR0
ST	STR751FR1
ST	STR751FR2
ST	STR752FR0
ST	STR752FR1
ST	STR752FR2
ST	STR755FR0
ST	STR755FR1
ST	STR755FR2
ST	STR755FV0
ST	STR755FV1
ST	STR755FV2
ST	STR911FM32
ST	STR911FM44
ST	STR912FM32
ST	STR912FM44
TI	TMS470R1A64
TI	TMS470R1A128
TI	TMS470R1A256
TI	TMS470R1A288
TI	TMS470R1A384
TI	TMS470R1B512
TI	TMS470R1B768
TI	TMS470R1B1M
TI	TMS470R1VF288
TI	TMS470R1VF688
TI	TMS470R1VF689

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
AMD	Am29LV800BB

Manufacturer	Name
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	AT49BV163D
Atmel	AT49BV163DT
Atmel	AT49BV1614

Manufacturer	Name
Atmel	AT49BV1614A
Atmel	AT49BV1614AT
Atmel	AT49BV1614T
Atmel	AT49BV162A
Atmel	AT49BV162AT
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	AT49BV640T
Atmel	AT49BV642D
Atmel	AT49BV642DT
Atmel	AT49BV6416
Atmel	AT49BV6416T
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
Atmel	AT49LV1614AT
Atmel	AT49LV2048A
Atmel	AT49LV4096A
Atmel	AT49SN3208
Atmel	AT49SN3208T

Manufacturer	Name
Atmel	AT49SN6416
Atmel	AT49SN6416T
Atmel	AT49SV322A
Atmel	AT49SV322AT
Atmel	AT49SV802A
Atmel	AT49SV802AT
Fujitsu	MBM29DL322BE/BD
Fujitsu	MBM29DL322TE/TD
Fujitsu	MBM29LV650U
Intel	28F004B3B
Intel	28F004B3T
Intel	28F008B3B
Intel	28F008B3T
Intel	28F016B3B
Intel	28F016B3T
Intel	28F128J3
Intel	28F128K18
Intel	28F128K3
Intel	28F128P30B
Intel	28F128P30T
Intel	28F128W18B
Intel	28F128W18T
Intel	28F160B3B
Intel	28F160B3T
Intel	28F160C3B
Intel	28F160C3T
Intel	28F256J3
Intel	28F256K18
Intel	28F256K3
Intel	28F256P30B
Intel	28F256P30T
Intel	28F320B3B
Intel	28F320B3T
Intel	28F320C3B
Intel	28F320C3T
Intel	28F320J3
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	28F640W18B
Intel	28F640W18T

Manufacturer	Name
Intel	28F800B3B
Intel	28F800B3T
Intel	28F800C3B
Intel	28F800C3T
Macronix	MX29LV320AB
Macronix	MX29LV320AT
Sharp	LH28F128BFHED
Sharp	LH28F128BFHT
Sharp	LH28F128SPHTD
Sharp	LH28F640BFHE-PBTL
Sharp	LH28F640BFHE-PTTL
Sharp	LH28F640BFHG-PBTL
Sharp	LH28F640BFHG-PTTL
Sharp	LHF00L29
Spansion	S29AL016Dxxxxx01
Spansion	S29AL016Dxxxxx02
Spansion	S29AL032Dxxxxx00
Spansion	S29AL032Dxxxxx03
Spansion	S29AL032Dxxxxx04
Spansion	S29AL032Hxxxxx01
Spansion	S29AL032Hxxxxx02
Spansion	S29AL032Hxxxxx21
Spansion	S29AL032Hxxxxx22
Spansion	S29AL032Hxxxxx31
Spansion	S29AL032Hxxxxx32
Spansion	S29AL032Hxxxxx41
Spansion	S29AL032Hxxxxx42
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	S29GL256M
Spansion	S29GL256N
Spansion	S29GL512N
Spansion	S71PL032J
Spansion	S71PL064J

Manufacturer	Name
Spansion	S71PL127J
SST	SST39LF200A
SST	SST39LF400A
SST	SST39LF800A
SST	SST39VF1601
SST	SST39VF1602
SST	SST39VF200A
SST	SST39VF3201
SST	SST39VF3202
SST	SST39VF400A
SST	SST39VF6401
SST	SST39VF6401B
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	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	M29W800DB
ST	M29W800DT
ST	M58LW064D
Toshiba	TC58FVB160
Toshiba	TC58FVT160

Chapter 9

Performance

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

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

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

Index

