2-Link MRK3 User Guide

Revision 2.5

Emulator for
MRK-III(e) Cores

A product of SEGGER Microcontroller GmbH & Co KG

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.

© 2007-2016 SEGGER Microcontroller Systeme GmbH, 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

Revisions

This manual describes the 2-Link device.
For further information on topics or routines not yet specified, please contact us.

Revision	Date	By	Explanation
2.5	161104	GKf	added new command calcSignDyn added new command openEx fixed bugs and compiler warnings improved resource deployment and documentation
2.4	151105	GKf	fixed isProtected command new rwUlpMasked command
2.3	150330	GKf	added detailed description for usbaddr command
2.2	150306	GKf	erase feature enabled at external power supply
2.1	141106	GKf	new copyULP command for dedicated ICs
2	140714	GKf	new PowerOnReset command new source command for script files in interactive mode new unprotect command for dedicated ICs new MantraC specific commands new powersel command for 5V support Win7 driver installation other new commands and updates
1	090908	AW	With version 1.30 of the software, some function parameter have been changed. The following functions and descriptions were updated: rEROM wEROM rULP wULP
0	090610	AW	Initial version for 2-Link MRK3.

Table of Contents

Chapter 1

Introduction

This chapter gives a short overview about 2-Link for MRK-III(e).

1.1   About this document

This document describes the 2-Link for MRK-III(e). It provides an overview over the major features of 2-Link for MRK-III(e), gives you some background information about NXP MRK-III(e) and describes 2-Link for MRK-III(e) related software packages available from Segger. Finally, the chapter Support helps to troubleshoot common problems.

1.2   Typographic conventions

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 (i.e. system functions, file- or pathnames).

1.3   2-Link overview

2-Link is a small USB to 10-pin conversion box, initially designed for NXP MRK-III(e) cores. It can be used for MRK-III(e) cores when the MRK3LINK software is used. It connects via USB to a PC running Microsoft Windows 2000, XP, Vista, Win7. 2-Link has a built-in 10-pin connector, from which 5 pins are used. The 10-pin ribbon cable connects to the hardware thru an adaptor.

1.3.1   Features of 2-Link for MRK-III(e)

- Any MRK-III(e) core supported e.g. NCF2950 (KEyLink Lite), NCF2951 (KEyLink 3D), NCF2960 (Smart2), NCK2983 (MantraC), NJJ29C0 (Joker)
- Target can be powered via USB through 2-Link or external power supply, support for 3 and 5V devices.
- Fully plug and play compatible
- USB and 10-pin ribbon cable included

1.4   Specifications

1.4.1   Specifications for 2-Link

Power Supply	USB powered <50mA
USB Interface	USB 2.0, full speed
Target Interface	5-pin
Supported Target Voltage	2.1 - 5.0 V
Operating Temperature	+5°C … +60°C
Storage Temperature	-20°C … +65 °C
Relative Humidity (non-condensing)	<90% rH
Size (without cables)	100mm x 53mm x 27mm
Weight (without cables)	70g
Electromagnetic Compatibility (EMC)	EN 55022, EN 55024
Supported OS	Microsoft Windows 2000/XP/Vista/Win7

1.5   Requirements

Host System

In order to use 2-Link for MRK-III(e), you need a host system running Windows 2000, Windows XP, Vista, Win7.

Target System (Hardware)

An MRK-III(e) target system is required. The system should have a 5-pin single in-line female connector with 2.54mm spacing. For proper operation, the interface cable must not exceed a total length of 300 mm. The individual pins are described in section Target interface connector. The individual pins of the target interface cable are described in section Target interface cable.

Chapter 2

Setup

This chapter contains the installation procedure of the 2-Link USB driver, the FAQ and the troubleshooting with common solutions to problems which might occur when using 2-Link for MRK-III(e). These problems can be associated with the tool chain, your target hardware, the use of functions or performance or the 2-Link hardware itself. There are several steps you can take before contacting support. Performing these steps can solve many problems and often eliminates the need for assistance.

2.1   Installing the driver

2.1.1   Win XP installation

When your target device is plugged into your computer's USB port, or when the computer is first powered up after connecting the 2-Link, Windows will detect the new hardware. The wizard starts the installation of the driver.

First select the "Search for a suitable driver for my device (recommended)" option, then click on the "Next >" button.

In the next step, you need to select the "Specify a location" option, and click on the "Next >" button.

The wizard will ask you to help it find the correct driver files for the new device. Use the directory navigator to select "D:\2link_v100\" (or your chosen location) and confirm with a click on the "Next >" button.
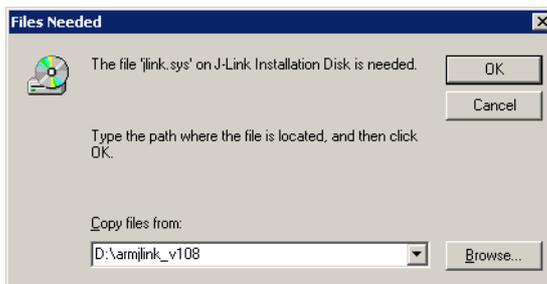
The wizard confirms your choice and starts to copy, when you click on the "Next >" button.

At this point, the installation is complete. Click on the "Finish" button to dismiss the installation.

2.1.2   Win 7 Installation

After Plugging the device into the USB port, the green LED starts to flash which indicates that a valid driver is missing. After successfully walking through the next steps the LED should be constantly ON which indicates the "READY" state of the device. Afterwards control and debug software e.g. MRK3Link.exe can be used.

Before one can install the driver it is necessary to download the driver le from the vault. This compressed archive has to be unpacked rst into a folder. In our example it is "c:\temp". The Installation starts with opening the Device Manager which can be found in the windows control center.
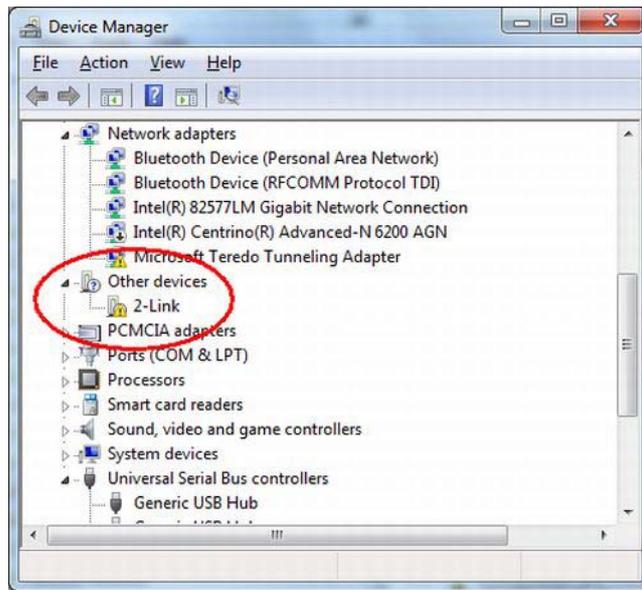
Windows 7 Device Manager

One has to select new item found and right click on it.
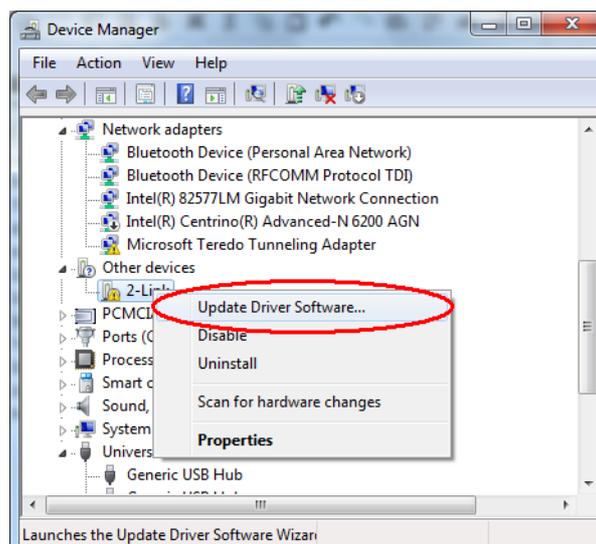
Device Manager - Update Driver

Please choose a driver by hand and go to the location where the driver package was extracted before.
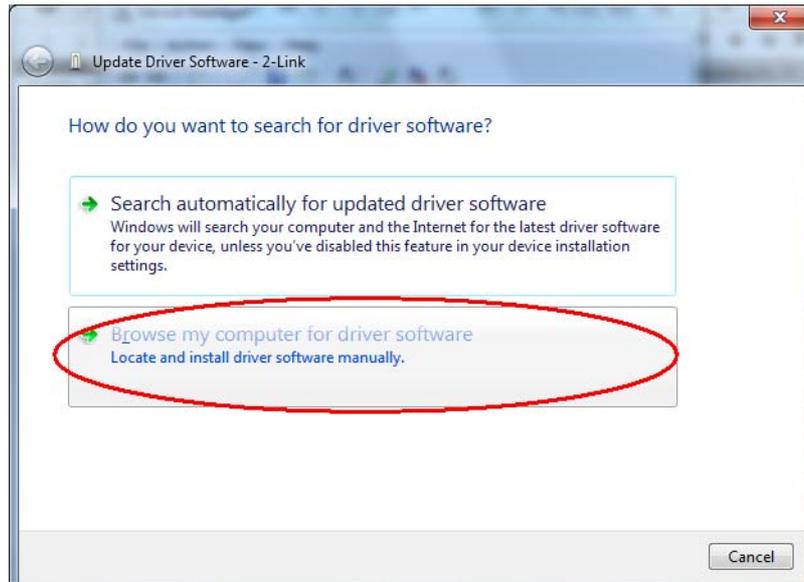
Select 2-Link Driver Step 1
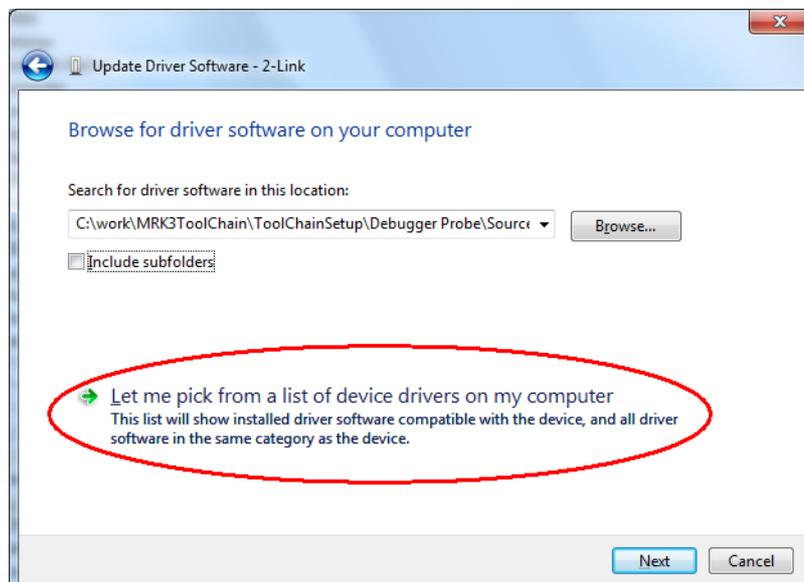
Select 2-Link Driver Step 2
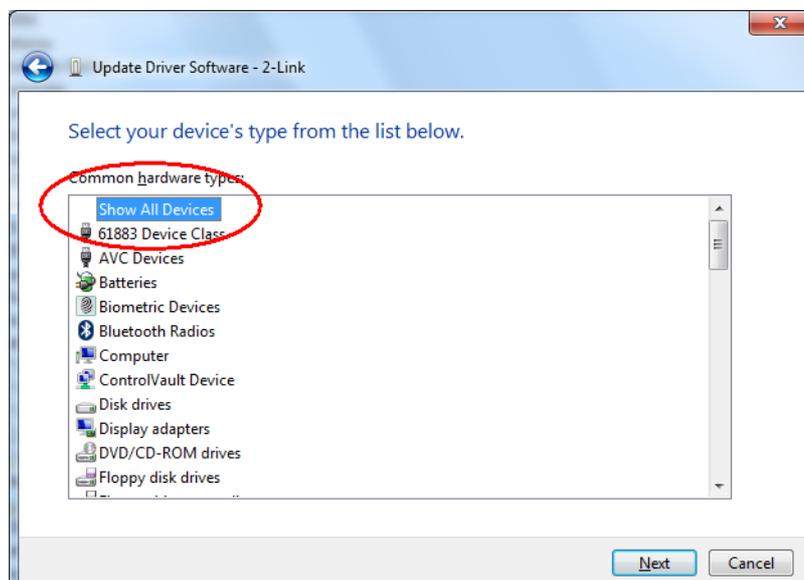
Select 2-Link Driver Step 3
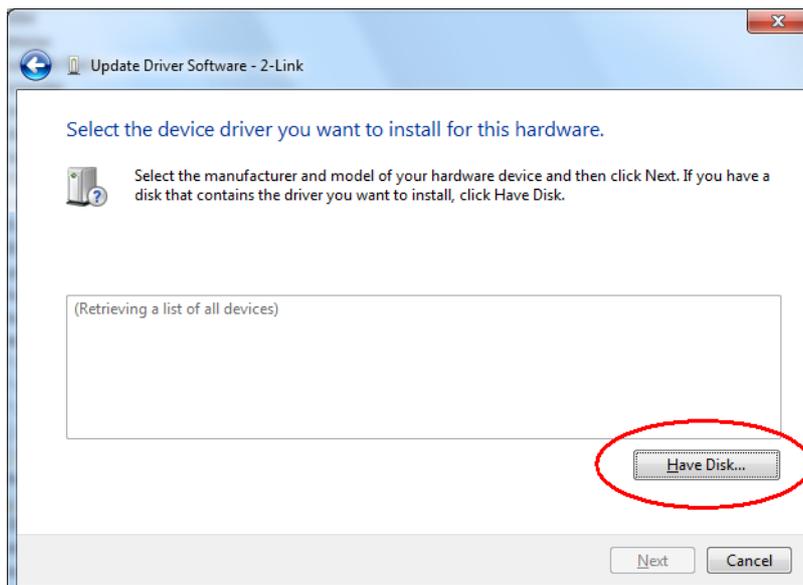
Select 2-Link Driver Step 4
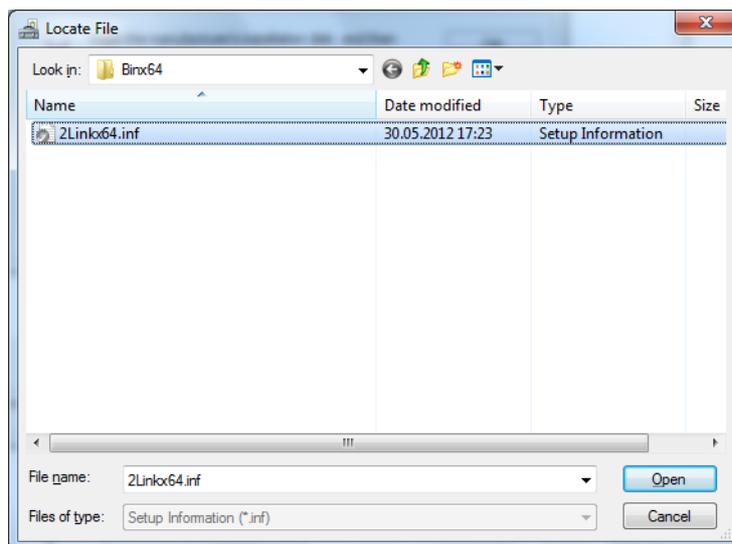
Select 2-Link Driver Step 5
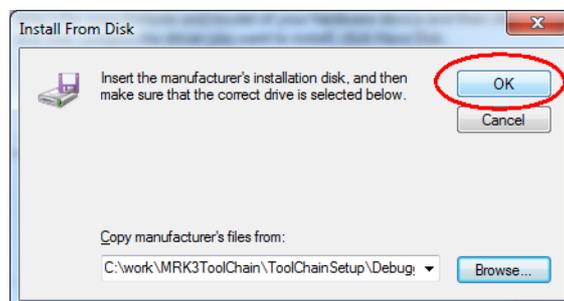
Select 2-Link Driver Step 6
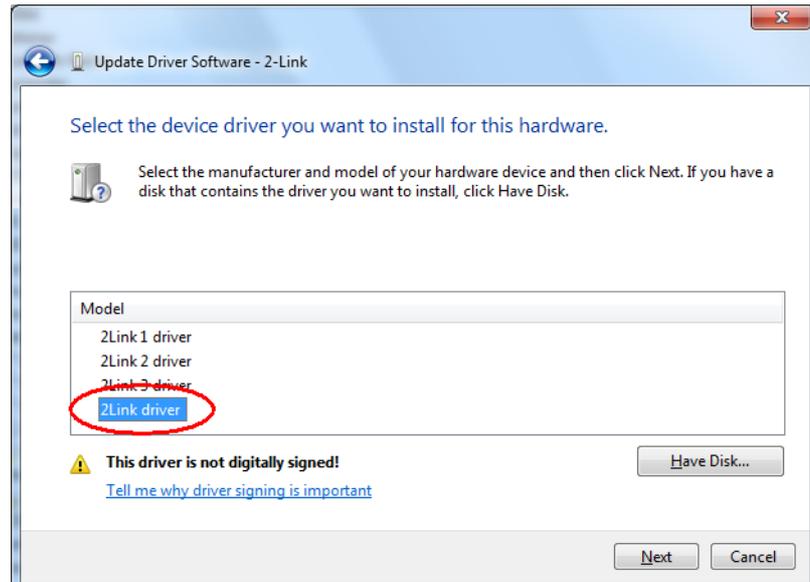
Select 2-Link Driver Step 7
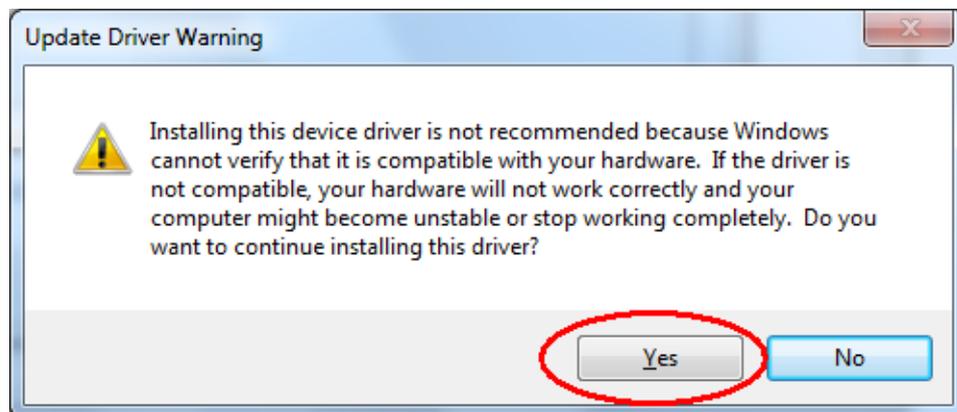
Select 2-Link Driver Step 8

The operating system will now install the driver modules for the 2-Link device.
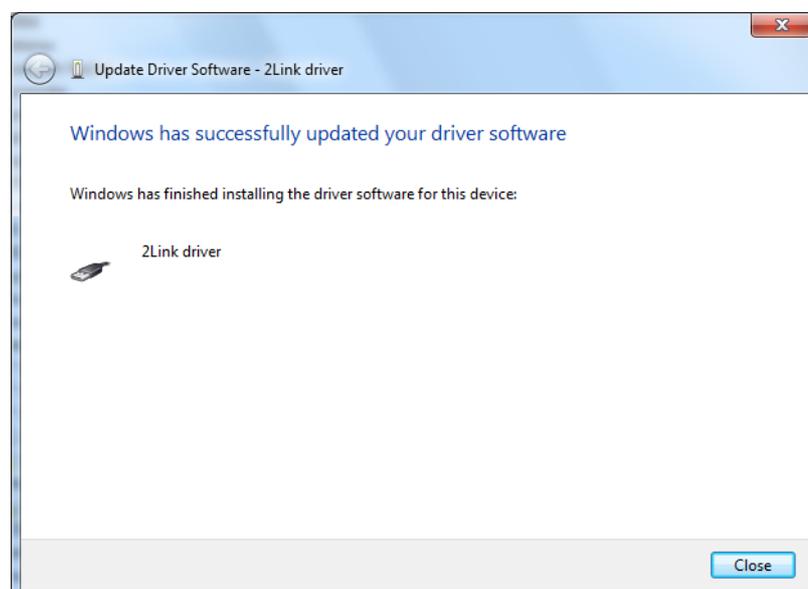
Successful driver installation

2.2 Connecting the target system

2.2.1 Power-on Sequence

In general, 2-Link for MRK-III(e) should be powered on before connecting it to the target device. That means you should first connect 2-Link MRK-III(e) with the host system via USB and then connect 2-Link MRK-III(e) to the target device via the target connector.

2.2.2 Verifying driver installation and operation

To verify the correct driver installation call the Windows Device Manager. If the driver is installed and your 2-Link is connected to your computer the Device Manager should list the 2-Link USB driver as a node below "Universal Serial Bus controllers" as shown in the following screenshot:
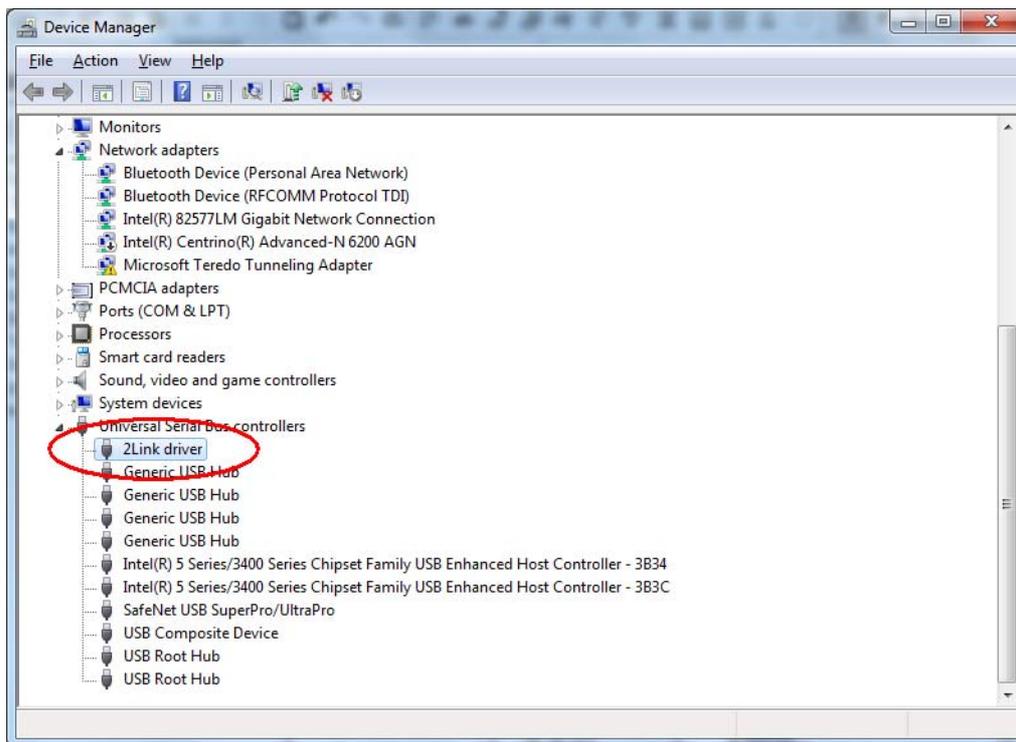
Right-click on the driver to open a context menu which contains the command `Proper-ties`. If you select this command, a new dialog box is opened and should report:
`This device is working properly`. Since version 1.20 of the software, up to 4 2-Links can be connected to one PC. They identify as "2-Link driver", "2-Link 1 driver", "2-Link 2 driver" and "2-Link 3 driver". When a 2 Link is programmed to use any channel from 1 to 3, a new driver installation may be requested, when the 2-Link is connected for the first time.
To verify the correct operation of the USB driver, disconnect and reconnect the 2-Link to the USB port. During the enumeration process which takes about 2 seconds, the LED on the 2-Link is flashing and afterwards glows permanently. Connect your target hardware with the 2-Link via the 10-pin ribbon cable. Start MRK3Link.exe You will be asked to enter a USB channel number used for the connection. As long as your 2Link MRK-III(e) is not configured to use a different channel, type 0. The PC connects to the 2-Link and reads out the software and hardware version numbers and serial number.
It should display a screenshot like shown below:

```
C:\jLinkarmdll_v222a\JLink.exe                                    _ □ ×
SEGGER J-Link Commander V2.22. '?' for help.
Compiled 18:31:06 on Nov 17 2004.

Firmware info: J-Link compiled Nov 15 2004 10:55:44 -- Update --
Speed set to 30 kHz
VTarget = 3.361V
Found 1 JTAG device. (ARM7)
JTAG Id: 0x1F0F0F0F  Version: 0x1 Part no: 0xf0f0 Man. Id: 0787
J-Link>_
```

type "enter"
2-Link switches on power supply to target and switches target CPU into MONITOR mode.

2.2.3   Target supply options

The power supply for the target may be delivered by 2-Link, or the target may be supplied externally by battery or any other supply.
The supply voltage can either be 3.3 Volts or 5 Volts. This can be siwtch with the powersel command and its options.

2.2.4   Target supply delivered by 2-Link

When 2Link accesses the target the first time, the supply voltage of the target is automatically measured. If no supply is detected, the target device is supplied by 2-Link with a voltage of approximately 3.3 Volts.

2.2.5   External target supply

When 2Link accesses the target the first time and measures a target supply voltage above 1.8 Volts, the target is assumed to be supplied externally and 2-Link does not activate its own target supply output.
does not function correctly on a different PC or with a different target hardware, then the other system may have similar problems to the original system or the problem might be with the 2-Link.

2.2.6   Problems

If you experience problems with any of the steps described above, it is recommended you read the chapter Troubleshooting. If you still do not find appropriate help there and your 2-Link is an original Segger product, you may contact Segger support via e-mail. Provide the necessary information about your target processor, board etc. and we will try to solve your problem. A checklist of the required information together with the contact information can be found in the same chapter.

Chapter 3

2-Link related software

This chapter describes Segger's 2-Link MRK-III(e) related software portfolio. The table below lists the available software packages.

Software	Description
MRK3Link.dll	DLL to use 2-Link MRK-III(e) with third-party programs.
MRK3Link.exe	Free command-line tool with basic functionality for target analysis.
MRK3-Link Software Developer Kit (SDK)	The 2-Link Software Developer Kit is needed if you want to write your own program with 2-Link MRK3-III(e).

3.1   Free software

Any kind of free 2-Link for MRK-III(e) related software is shipped together with 2-Link for MRK-III(e) and may also be downloaded from www.segger.com. No additional license is required to use this software.

3.1.1   MRK3Link.dll (2-Link use with third party programs)

The MRK3Link DLL is a standard Windows DLL typically used from "C" (or any other programming environment capable of addressing DLLs with C standard interface like Visual-Basic, Delphi, C# or Java). It makes the entire functionality of the 2-Link for MRK-III(e) available thru the exported functions.
The functionality includes things such as halting/stepping the core and reading/writing CPU registers and memory. Therefore it can be used in any kind of application accessing an MRK-III(e) core.

Updating the DLL in third-party programs

The MRK3Link.dll can be used by any debugger that is designed to work with it. Although some debuggers usually ship with the MRK3Link.dll already installed, it may make sense to replace the included DLL with the latest one available, to take advantage of improvements in the newer version.

Determining the version of MRK3Link.dll

To determine which version of the MRK3Link.dll you are facing, the DLL version can be viewed by right clicking the DLL in the explorer, and choosing "Properties" from the context menu. Click the Version tab or Details tab to display information about the Product Version.
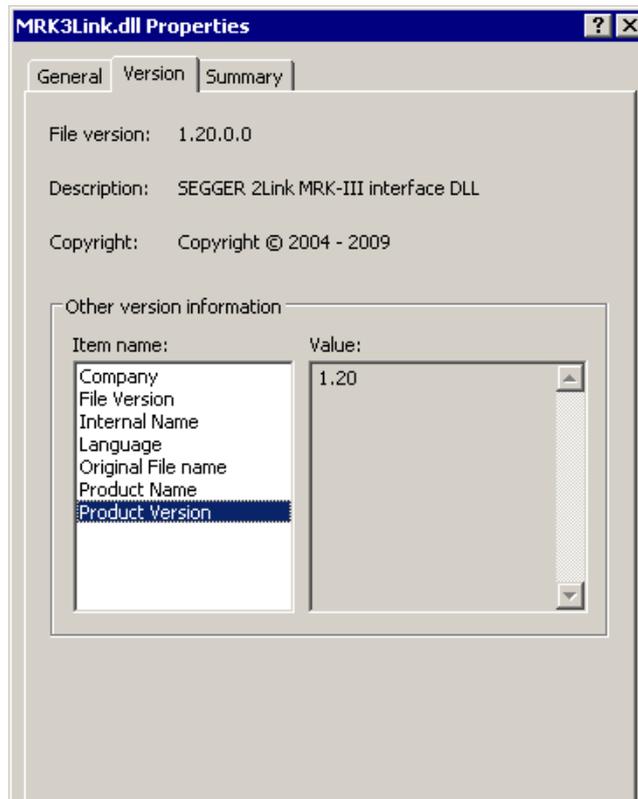
Determining which DLL is used by a program

To verify that the program you are working with is using the DLL you expect it to use, you can investigate which DLLs are loaded by your program with tools like Sysinternals' Process Explorer. It shows details about the DLLs, used by your program, such as manufacturer and version.

Process Explorer is - at the time of writing - a free utility which can be downloaded from www.sysinternals.com.
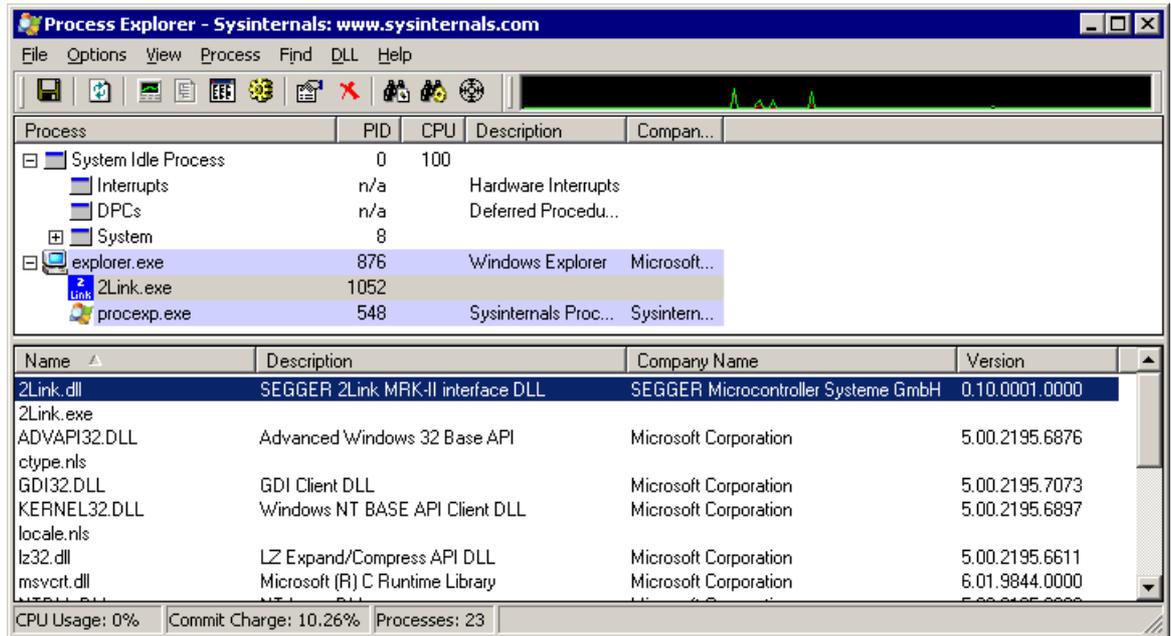
3.1.2 MRK3Link.exe (Command line tool)

MRK3Link.exe is a tool that can be used to verify proper installation of the USB driver and to verify the connection to the MRK-III(e) chip, as well as for simple analysis of the target system. It permits commands, such as halt, step, go, as well as commands to read and write all memory areas of the target device.
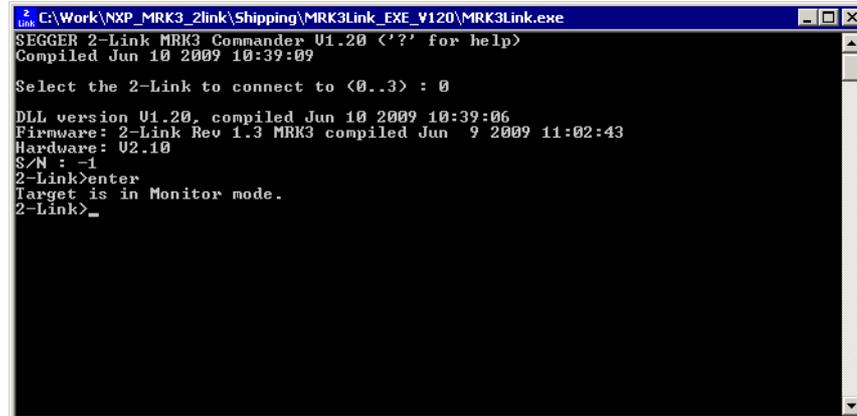
3.2   Additional software packages

The packages described in this section are not available for download. If you wish to use one of them, contact SEGGER Microcontroller GmbH & Co KG.

3.2.1   MRK3Link Software Developer Kit (SDK)

The MRK3Link Software Developer Kit is needed if you want to write your own program with 2-Link for MRK-III(e).

The MRK3Link DLL is a standard Windows DLL typically used from "C" (or any other programming environment capable of addressing DLLs with C standard interface like Visual-Basic, Delphi, C# or Java). It makes the entire functionality of the 2-Link for MRK-III(e) available thru the exported functions. The functionality includes things such as halting/stepping the core and reading/writing CPU registers and memory. Therefore it can be used in any kind of application accessing an MRK-III(e) core.

The table below lists some of the included files and their respective purpose.

Files	Contents
GLOBAL.h MRK3LinkDLL.h	Header files that must be included to use the DLL functions. These files contain the defines, typedef names and function declarations.
MRK3Link.lib	Library contains the exports of the MRK3Link DLL.
MRK3Link.dll	The DLL itself.
Main.c	Sample application, which calls some MRK3Link DLL functions.
2Link.vsproj 2Link.sln	Project file of the sample application. Double click for open. Solution file of the sample application. Double click for open.
2-Link MRK3 API.pdf	Complete documentation for the API and sample projects.

Chapter 4

MRK3Link.exe reference

The MRK3Link.exe is a sample application to demonstrate usage of 2-Link for MRK-III(e) with an MRK-III(e) target. This chapter lists the available commands.

4.1 MRK3Link.exe commands

MRK3Link.exe has several commands to access the connected target device. MRK3Link.exe may be used in three different modes which support all of the commands.
- interactive mode - all commands are just typed on the console
- script mode - all commands are read from a text file
- command line mode - commands are passed to MRK3Link.exe as parameter.

4.2 Command overview

The table below lists the available commands of MRK3Link.exe. All commands are listed in alphabetical order.

Routine	Explanation
Abort	Send an abort commad to the target.
baud	Select communication speed.
ClrBP	Clears breakpoint.
copyPage|cp	Copies one EROM page.
copyULP	Copies ULPEE pages
DevStat|ds	Get and show the CPU status word.
download	Download a file to target
Echo	Output text message for use in scripts.
Enter|e	Unconditionally enter monitor mode
Erase	Unprotect target and erase target EROM memory
filesig	Calculates the 24 bit signature of a given file.
fillpattern|fp	Sets a 32 bit fill pattern for a given memory.
fillrange|fr	Sets a fill range for a given memory.
g	Executes the FlashLoader feature.
g	Starts the MRK-III(e) core.
h	Halts the MRK-III(e) core.
HwStat|hs	Shows the status information of the interface to target.
iOff	Unconditionally switches the interface to target into High-Z mode.
ioTest|io	Enables/disables the NAND tree test mode.
IsHalted	Checks if the MRK-III(e) core is halted.
IsProtected	Checks if the MRK-III(e) core is in protected mode.
log	Activates a log file for 2-Link
noverify	Disables verify for following downloads.
powerdown|pd	Switches the power supply to target on
powersel|ps	Select power supply for target device.
powerup|pu	Switches the power supply to target down
poweronreset|pr	Enter monitor mode after power down.
progPfBits|ppb	Writes bits of an EROM page.
Protect	Sets the MRK-III(e) core into protected mode.
q	Disconnects target connection and closes 2Link.exe.
r	Resets the target CPU
rDat	Reads data area.
rEEPROM	Reads EEPROM area.
regs	Reads register status.
rEROM	Reads EROM area.
rStatusCode|sc	Shows the status code of the last command.
rULP	Reads the ultra low power EEPROM.
rwUlpMasked	Reads ULP data and writes back masked data

Routine	Explanation
s	Single steps the MRK-III(e) core.
SendC	Send a command directly to the target.
SendD	Send a data word directly to the target.
setBitField|sbf	Writes a bit field in the data area.
SetBP	Sets breakpoint.
SetPC	Sets PC register.
SetReg	Sets a register value.
sig	Displays all target signatures
sigDyn	Display a signature of a user defined memory area.
sleep	Sleep specified time in milliseconds.
source	Run script file in interactive mode
unprotect	Unprotects dedicated ICs by a given key
upload	Uploads data from target and stores it to a file.
usb	Selects the USB channel communication to the 2-Link for MRK-III(e)
usbaddr	Sets the USB communication channel of the connected 2Link MRK-III(e)
USER_Clean|uc	Clears the USER data buffer in the 2-Link MRK-III(e)
USER_Disp|ud	Continuous display of user data.
USER_GetCnt|ug	Delivers the number of samples in the user data buffer
USER_Read|ur	Reads out the user data buffer.
USER_Start|us	Starts sampling of user data from the target.
USER_Stop|up	Staops sampling of user data.
verify	Enables verify for following download
version|v	Shows the CPU version info of the connected target
wDat	Writes data area.
wEEPROM	Writes EEPROM area.
wEROM	Writes EROM area.
wULP	Writes to ultra low power EEPROM area.
wULPBytes	Write bytes to ultra low power EEPROM area.

4.3   Command detailed description

All available commands are described in detail in this chapter. All commands are case insensitive. All commands may be used in interactive mode, written in a script file, or passed as parameter when starting MRK3Link.exe (command line mode). When used as command line parameter, all commands have to be preceded by two hyphens "--". The commands are processed as they appear, one after the other. If a command requires parameters, the parameters has to be separated from the command by a space character. Some commands require more parameters, which have to be separated by commas.

4.3.1   Abort

Sending an abort to the target stops a running command in the target immediately.
The command takes no parameter.

Syntax: abort

4.3.2   baud

This command selects the communication speed between the host and the target device.

Syntax: baud <speed>

Parameter: <speed>

<speed>

- 0: 125 kHz
- 1: 250 kHz

4.3.3   ClrBP

Clears a breakpoint which was set by a call of SetBP. The command takes two parameter.

Syntax: ClrBP <No>, <Addr>

Parameter: <No>, <Addr>

<No>

The No parameter is a number between 0 to 3 to identify the breakpoint which should be modified.

<Addr>

The Addr parameter is the address of the breakpoint which should be cleared.

4.3.4   copyPage|cp

This is a MantraC/ATIC134M specific command and copies one EROM page from source address to destination address. The address range is 0-1FF.

Syntax: copyPage|cp <srcAddr>,<desAddr>

Parameter: <srcAddr>,<desAddr>

<scrAddr>

The source page address from where the data should be copied.

<desAddr>

The destination page address to where the data should be copied to.

4.3.5   copyULP

This command copies a given number of pages from a source address to a destination address.

Syntax: copyULP <srcAddr>,<destAddr>,<numPages>

Parameter: <srcAddr>,<destAddr>,<numPages>

<srcAddr>

The source start address from which the pages should be copied from.

<destAddr>

The destination start address to which the pages should be copied to.

<numPages>

The number of pages to be copied.

4.3.6   DevStat|ds

This command delivers the status word of the target device.
This includes informations about the Monitor mode, the exec mode, the CPU idle state, watchpoints and the device mode.
The command takes no parameter.

Syntax: DevStat|ds

output example:

MONITOR mode: YES
exec mode: SYSTEM
CPU idle: NO
watchpoints: 0h
device mode: INIT

4.3.7   download

The download command is used to write data into a target device memory area. The data is read from an INTEL HEX file. The download command expects following (required) parameters:

Syntax: download <memory>,<filename>

Parameter: <memory>,<filename>

<memory>

specifies the target memory area which should be used by download. Valid values for memory are: DATA, EEPROM, EROM and ULP

<filename>

specifies the file to download. The file must be an INTEL-Hex file.
If the file name is entered without extension, the extension ".hex" will be appended.

If a fill range has been set for <memory> by a previous 'fillrange' command, the full address range is written to the device. Any memory address not covered by data from the hex file will be overwritten by the fill pattern in little-endian byte order (LSB to MSB).
In addition the command displays the signature (24 bit CRC) calculated (on the host side) within the fill range. If the fill range covers the full memory, this signature should be identical with the corresponding output of the 'sig' command.

4.3.8   Echo

This command outputs a text message. This command is used for script mode to tell the user specific events.

Syntax: Echo <message>

Parameter: <message>

<message>

specifies the output message

4.3.9   Enter|e

This command is used to send an "Enter monitor" sequence to the connected device. This command is not required to access the connected device, because 2Link for MRK-III(e) issues the enter monitor sequence automatically whenever this is required. It's primary use is for hardware analysis and debugging purposes.

Syntax: enter

4.3.10   Erase

This command is used to erase the connected device and bring it to INIT (initial) mode. If the device was protected, the protection is removed after all memories are erased. This applies to the EROM, ULP EEPROM and Parallel EEPROM (if existent). The command takes no parameters. For a device in protected mode, the erase command must be executed immediately after the 'enter' to avoid that the on-chip watchdog terminates the monitor mode (and forces a device reset) after 0.5 s. Therefore the enter command is automatically send before YES erase command. To enable the enter sequence when the chip is powered externally the WAKE pin of the 2-link box has to be connected with the RESET pin of the target chip.

Syntax: erase

4.3.11   filesig

Use this command to obtain a signature (checksum) for a certain memory by reading code from a hex file. It is required to define a fill range beforehand, to specify the address range for signature calculation. After download, the result should be identical to the output of the 'sig' command for the same memory (assuming the whole memory is loaded).
The calculated signature is also displayed by the download command.

Syntax: filesig <memory>,<filename>

Parameter: <memory>,<filename>

<memory>

The name of the memory (DATA, EROM, ULP, EEPROM).

<filename>

Name of the hex file containing the code, as used in the 'download' command.

4.3.12   fillpattern|fp

This command sets a 32 bit fill pattern for the given memory type.

Syntax: fillpattern|fp <memory>,<Data32>

Parameter: <memory>,<Data32>

<memory>

The memory to be filled with the given pattern.

<Data32>

The pattern with which the memory should be filled.

4.3.13   fillrange|fr

Sets the address range that will be filled up (padded) with the fill pattern, in addition to the code from the hex file, at the next "download" command for the given memory. The same fill range is considered by the signature calculation on host side ("filesig" command). If the hex file contains addresses outside the defined range, an error occurs.

Syntax: fillrange|fr <memory>,<startAddr>,<endAddr>

Parameter to define: <memory>,<startAddr>,<endAddr>

Parameter to undefine: <memory>,off

<memory>

The name of the memory (DATA, EROM, ULP, EEPROM).

<startAddr>

Start address of fill range.

<endAddr>

End address of fill range.

4.3.14   g

Starts the target device. The command takes no parameter.

Syntax: g

If the target is stopped on a breakpoint, the target device will halt again immediately wihtout executing any instruction. To continue execution from a breakpoint, issue a single step command first before sending a go command.

4.3.15 h

Halts a running target device, or switches the device into MONITOR mode, when the halt command is called the first time. Therefore the halt command may be used as firs command to acces a target CPU.
The command takes no parameter.

Syntax: h

4.3.16 HwStat|hs

The status of the interface to the target board is shown. This includes the supply voltage of the target and the state of the interface signals.
The command takes no parameter.

Syntax: HwStat

4.3.17 iOff

This command switches all communication lines from 2Link to the target device into high impedance state.
The command takes no parameter.

Syntax: iOff

The target interface remains disconnected until the next access to the target is required by any other command. The target supply is not affected by this command.

4.3.18 ioTest|io

Enables or disables the NAND tree test mode in the target device.

Syntax: ioTest|io <control>

Parameter: <control>

<control>

• 1: starts the NAND tree test mode
• 0: stops the NAND tree test mode

4.3.19 IsHalted

This command may be used to check whether a connected target device is running or halted. The command takes no parameter.

Syntax: IsHalted

The state of the target is displayed on the screen

4.3.20 IsProtected

This command may be used to check whether a connected target device is protected. The command takes no parameter.

Syntax: IsProtected

The state of the target is displayed on the screen.

4.3.21 log

This command activates a log file which logs communication between 2Link.exe and 2Link. The command may take one parameter.

Syntax: log <filename>

Parameter: <filename>

<filename> specifies the file which shall be used as log file. If the parameter is omitted, the log file is closed and logging is switched off.

4.3.22   noverify

This command disables verification after download. It is valid for all subsequent downloads until a "verify" command is given.
"Noverify" is the default mode for downloads.
The command takes no parameter.

Syntax: noverify

4.3.23   powerdown|pd

This command unconditionally switches the power supply off, if the target is supplied by 2Link and disconnects the interface to the target by switching all communication lines into High-Z mode. The command takes no parameter.

Syntax: powerdown|pd

This command may also be used to control the behavior on exit. After issuing a power-down command, the target is switched off when MRK3Link.exe is closed.

4.3.24   powersel|ps

This command selects the 2-Link output voltage for the target device. The selected output voltage can alternatively stored and locked in the configuration memory of the 2-Link for future use. If the target device is powered by an external supply the value of external voltage is taken as it is. Thus this command is only for power supply by 2-Link.

Syntax: powersel|ps <voltage> [mode]

Parameter: <voltage>, [mode]

<voltage>

- 3: select 3V output voltage
- 5: select 5V output voltage

[mode]

- no mode parameter: voltage selected and switched after pd/pu - value not stored in 2-link non volatile memory
- [weak]: voltage selected and switched after pd/pu - value stored in 2-link non volatile memory but is not locked against new selection - new value can be changed without weak or perm parameter
- [perm]: voltage selected and switched after pd/pu - value stored in 2-link non volatile memory and is locked against new selection - new value can only be changed with weak or perm parameter but not without the mode parameter.

4.3.25   powerup|pu

This command unconditionally switches on the power supply, if the target is not supplied externally. The command takes no parameter.

Syntax: powerup|pu

This command may also be used to control the behavior on exit. After issuing a powerup command, the target remains supplied by 2Link when MRK3Link.exe is closed. The communication interface is always put in High-Z mode on exit, even when a powerup command was given.

4.3.26 poweronreset|pr

This command switches off the power by sending a power down command (see above) then waits approx. 100ms and then sends a enter monitor sequence to switch on the target power and puts the CPU into monitor mode.

Syntax: poweronreset|pr

This command takes no parameters.

4.3.27 progPfBits|ppb

This is a MantraC/ATIC134M specific command and writes specific number of bits starting from an EROM page and its bit offset.

Syntax: progPfBits|ppb <wAddr>,<Offset>,<Number>,<Data16> [<Data16> ...]

Parameter: <wAddr>,<Offset>,<Number>,<Data16> [<Data16> ...]

<wAddr>

specifies the start address of the EROM that should be written. A hex value has to be typed. No range check is performed.

<Offset>

The bit offset of the start address. This value has to be given as decimal number.

<Number>

The number of bits to be write. This value has to be given as decimal number.

<Data16> [<Data16>, ...]

The data words that has to be written to the EROM memory.

4.3.28 Protect

This command is used to protect a connected target. After protection, any access to the target's memory is locked, until the target is erased and unprotected. The command takes no parameter.

Syntax: protect

Some MRK-3 devices may not implement the protected mode. In this case, the command does nothing.

4.3.29 q

This commands ends MRK3Link.exe and switches the communication lines from 2Link to target into High-Z mode.

Syntax: q

If a powerup command was given before calling quit, the target remains supplied if it was supplied by 2Link.

4.3.30 r

This command is used to reset the target. After resetting the target, it is automatically switched into MONITOR mode again and halted at 0.
The command takes no parameter.

Syntax: r

4.3.31 rDat

This command is used to read the data area of a connected target device. The data area also includes the CPU registers and special function registers.
The command requires two parameter.

Syntax: rDat <Addr>,<Size>

Parameter: <Addr>, <Size>

<Addr>

specifies the start address of the data area that should be read. A hex value has to be typed. No range check is performed.

<Size>

specifies the number of bytes which should be read from the data area, starting at the start address given before. No range check is performed.

4.3.32   rEEPROM

This command is used to read the EEPROM area of a connected target device. The command requires two parameter.

Syntax: EEPROM <Addr>,<Size>

Parameter: <Addr>, <Size>

<Addr>

specifies the start address of the EEPROM area that should be read. A hex value has to be typed. No range check is performed.

<Size>

specifies the number of bytes which should be read from the EEPROM area, starting at the start address given before. No range check is performed.

4.3.33   regs

This command is used to display the current register contents of the connected device. The device has to be halted, before the command can be executed.
The command takes no parameter.

Syntax: regs

4.3.34   rEROM

This command is used to read the EROM area of a connected target device. The command requires two parameter.

Syntax: rEROM <wAddr>,<NumWords>

Parameter: <wAddr>, <NumWords>

<wAddr>

specifies the start address of the EROM area that should be read.
A hex value has to be typed which is the "word-address". One 16bit word can be written to every EROM address. No further range check is performed.

<Size>

specifies the number of bytes which should be read from the EROM area, starting at the start address given before.
No range check is performed.

4.3.35   rStatusCode|sc

This command delivers the status code of the last command send and received from the target. The command takes no parameter.

Syntax: rStatuscode|sc

Parameter: no parameter

4.3.36   rULP

This command is used to read the ULP area of a connected target device. The command requires two parameter.

Syntax: rULP <pAddr>,<NumPages>

Parameter: <pAddr>, <NumPages>

<pAddr>

specifies the address of the first page in the ULP area that should be read. A hex value has to be typed. No range check is performed. Every page holds one 32bit data item.

<NumPages>

specifies the number of pages which should be read from the ULP area, starting at the page address given before.
The ULP can not be read byte-wise, therefore, the command reads pages with one page = 4 Bytes.
No range check is performed.

4.3.37   rwUlpMasked

This command reads a given ULPEE page, performs bit-wise AND and OR operations, and writes the result back to the same page address. It can be used to change bit fields within a page, without changing other bits in the same page. (Note that this supersedes the wUlpBytes command.)

Syntax: rwUlpMasked <pAddr>,<Mask32>,<Data32>

<pAddr>

The source and destination ULPEE page address.

<Mask32>

Pattern used in the bitwise AND operation.

<Data32>

Data used in the bitwise OR operation.

Calculation:
<pAddr Data32> = (<pAddr Data32> & <Mask32>) | <Data32>

4.3.38   s

This command is used to execute a single instruction on the connected target device.
The target CPU has to be halted, before a single step can be executed. The command takes no parameter.

Syntax: s

4.3.39   SendC

This command sends an MDI command directly to the target.

Syntax: SendC <Data>,<Mode>

Parameter: <Data>,<Mode>

<Data>

specifies the MDI command data that should be send directly to the target. A hex value has to be typed.

\<Mode\>

The communication mode - only transmit or transmit and receive.

4.3.40   SendD

This command sends a data word directly to the target.

Syntax: SendD \<Data\>

Parameter: \<Data\>

\<Data\>

specifies the data word that should be send directly to the target. A hex value has to be typed.

4.3.41   setBitField|sbf

This is a MantraC/ATIC134M specific command and sets a bit field in the data area.

Syntax: setBitField|sbf \<Addr\>,\<Position(dec)\>,\<Width(dec)\>,\<Data\>

Parameter: \<Addr\>,\<Position\>,\<Width\>,\<Data\>

\<Addr\>

specifies the address of the bit field that should be written. A hex value has to be typed. No range check is performed.

\<Offset\>

The start bit position of the bit field. This value has to be given as decimal number.

\<Number\>

The width of the bit field. This value has to be given as decimal number.

\<Data\>

The bit field data that has to be written.

4.3.42   SetBP

This command is used to set an instruction breakpoint. The target CPU has to be halted, before a breakpoint can be set. The command takes one parameter.

Syntax: SetBP \<No\>, \<Addr\>, [\<Flags\>]

Parameter: \<No\>, \<Addr\>, [Flags]

\<No\>

The breakpoint unit of the MRK-3 core can handle up to 4 breakpoints. The "No" parameter has to address the breakpoint which should be set. This number has to be in the range between 0 to 3 and is used to assemble a cSetBrkp command as described in the MDI documents.

\<Addr\>

The address parameter is the instruction address where the breakpoint should be set. It has to be a Hex number in the target CPU EROM address space.

[Flags]

The "Flags" parametr is optional. If no flags paramter is given, a hardware breakpoint at the given address is set.
Otherwise, the flags are used to assemble a cSetBrkp command as described in the MDI documentation.

A breakpoint which was set by a call of SetBP may be cleared by a call of ClrBP with the same No and Addr parameter.

4.3.43 SetPC

This command is used to set the program counter (PC) of the connected target CPU to a specified address. The target CPU has to be halted, before the PC can be set. The command takes one parameter.

Syntax: SetPC <Addr>

Parameter: <Addr>

The address parameter is the instruction address which should be written into the PC. It has to be a Hex number in the target CPU EROM address space.
A range check is not performed.

4.3.44 SetReg

This command is used to set any register to any value. The target CPU has to be halted, before the Register can be set. The command takes two parameter.

Syntax: SetReg <Index><Value>

Parameter: <Index><Value>

<Index>

The index parameter is used to address the register which should be set:
0: PC
1: PSW
2: R0
3: R1
4: R2
5: R3
6: R4
7: R5
8: R6
9: R7 (SP).

<Value>

The value parameter ist the value which should be written into the addressed register.
It has to be a 16bit hex-number.

4.3.45 sig

This command is used to display all signatures of the connected device. The command takes a number (0 to 10) as an optional parameter.

Syntax: sig [<memory instance>]

If the parameter is given, only the signature of the corresponding memory instance is requested from the device. If the device is not in protected mode and no parameter is given, the tool attempts to obtain signatures for memory instances 0 to 10 from the connected device and lists all available signatures. Please consult the datasheet or MDI documentation for the connected device about which instances are available. In protected mode, only the EROM signature can be retrieved. In this case, the command must be executed immediately after the 'enter' to avoid that the on-chip watchdog terminates the monitor mode (and forces a device reset) after 0.5 s.

4.3.46 sigDyn

This command is used to display a signature over a user defined memory area of the connected device.

Syntax: sigDyn <StartAddress>,<NumWords>

Parameter: <StartAddress>,<NumWords>

All parameters are required.

<StartAddress>

The memory start address of the signature calculation.

<NumWords>

The number of words of the signature calculation.

4.3.47   sleep

This command forces a delay for the specified time in milliseconds. This command is used for script mode to wait for specific time between execution of commands.

Syntax: Sleep <ms>

Parameter: <ms>

<ms>

specifies the time to wait in milliseconds.

4.3.48   source

Run a given script file in interactive mode.

Syntax: source <script command file>

Parameter: <script command file>

<script command file>

The given script command file is a simple text file with a list of command in each line. This file will be parsed by the program and executed command by command.

4.3.49   unprotect

This command unprotects dedicated ICs with a given 128 bit key.

Syntax: unprotect <128 bit key>

Parameter: <128 bit key>

<128 bit key>

A 128 bit key in hexadecimal format - e.g. 0123456789AbCdEf

4.3.50   upload

The upload command is used to read data from a target device memory area and write the data into a specified file. The upload command expects following parameters:

Syntax: upload <memory>,<filename>,<address>,<size>

Parameter: <memory>,<filename>,<address>,<size>

All parameters are required.

<memory>

specifies the target memory area which should be read by upload. Valid values for memory are: DATA, EEPROM, EROM, ULP

<filename>

specifies the INTEL HEX file where the data should be written to.
If the file name is entered without extension, the extension ".hex" will be appended.

<address>

is the start address of the target memory area.

<size>

is the number of ITEMS which should be read from target and written to file.

4.3.51 usb

This command selects the USB communication channel to the 2-Link.

Syntax: usb <addr>

Parameter: <addr>

<addr>

specifies the USB address 0-3.

4.3.52 usbaddr

This command sets the USB communication channel of the connected 2-Link.

Syntax: usbaddr = <addr>

Parameter: <addr>

<addr>

specifies the USB address 0-3.

Remark:
In order to use the new USB channel the following steps have to be done after executing this command:
1. terminate the console program
2. unplug the segger box
3. reconnect the segger box
4. if the LED of the segger box is now flashing fast install the appropriate USB driver for the new channel (see table below and chapter Win 7 Installation) - if the LED shines stable continue with step 5.
5. open the console program and start by typing the new USB channel number.

USB channel	USB driver name
0	2Link driver
1	2Link 1 driver
2	2Link 2 driver
3	2Link 3 driver

4.3.53 USER_Clean|uc

This command is used clean the user data buffer in the 2Link MRK3 which is used to sample the user data sent from the target.
The command takes no parameter.

Syntax: USER_Clean

4.3.54 USER_Disp|ud

This command is used to continously display the user data received from the device.

Syntax: USER_Disp [text]

Parameter (optional): [text]

Without any parameter, user data is displayed as 16 bit hex numbers. If the parameter 'text' is given, the tool displays the user data as ASCII text. In either case, the command is aborted by pressing the escape key on the keyboard. Hence this command is not intended for use in script files.

4.3.55 USER_GetCnt|ug

This command is used to retrieve the number of user data words (16bit item) which are currently sampled in the user data buffer of the 2Link MRK3.

The command does not take a parameter.

Syntax: USER_GetCnt

The command calls a DLL function which delivers the user buffer state. The result is translated and displayed on the screen.

4.3.56   USER_Read|ur

This command is used to read data from the user data buffer sampled in the 2Link MRK3. The command requires one parameter.

Syntax: USER_Read <NumItems>

Parameter: <NumItems>

specifies the number of 16bit data items which should be read from the user data buffer. All available items, or the number of requested items, whichever is least, will be listed on the screen.

4.3.57   USER_Start|us

This command is used to start sampling of user data.
After receiving the start command, the 2-Link MRK3 allows sampling of the user data sent from the target asyncronously to the system data used by the debugger.
As long as the USER_Start command was not sent, the user data are ignored by the 2Link MRK3.
The command does not take a paramerter.

Syntax: USER_Start

4.3.58   USER_Stop|up

This command is used to stop sampling of user data.
After receiving the stop command, the 2Link MRK3 will ignore all user data sent by the target CPU.
The command takes no parameter.

Syntax: USER_Stop

4.3.59   verify

This command is used to enable verification for subsequent download commands. This setting remains valid for all subsequent downloads, until a "noverify" command is called.
Noverify is the default mode for downloads.
The command takes no parameter.

Syntax: verify

4.3.60   version|v

This command is used to get the version informations of all software parts of the target device. Each version information has a specific ID, a mjor number and a minor number.
The command takes no parameter.

Syntax: version

4.3.61   wDat

This command is used to write one or more 8bit data bytes(s) to the DATA memory of a connected device. The command requires at least two parameters.

Syntax: wDat <Addr>,<DataByte> [<DataByte> ...]

Parameter: <Addr>, <DataByte>

<Addr>

specifies the start address of the DATA area that shall be written. This must be a byte address in hexdecimal notation.

<DataByte>

A list of 8bit data bytes (hex) which will be written starting from the specified DATA byte address. No further range check is performed.

4.3.62   wEEPROM

This command is used to write two or more 8bit data bytes(s) to the EEPROM memory of a connected device. The command requires at least two parameters.

Syntax: wEEPROM <Addr>,<DataByte> <DataByte> [<DataByte> <DataByte>...]

Parameter: <Addr>, <DataByte>

<Addr>

specifies the start address of the EEPROM area that shall be written. This must be a byte address in hexdecimal notation.

<DataByte>

A list of 8bit data bytes (hex) which will be written starting from the specified EEPROM byte address. No further range check is performed.

4.3.63   wEROM

This command is used to write one or more 16bit data word(s) to the EROM memory of a connected device. The command requires at least two parameters.

Syntax: wEROM <wAddr>,<DataWord> [<DataWord> …]

Parameter: <wAddr>,<DataWord>

<wAddr>

specifies the start address of the EROM area that shall be written. This must be a word address in hexdecimal notation.

<DataWord>

A list of 16bit data words (hex) which will be written starting from the specified EROM word address. No further range check is performed.

4.3.64   wULP

This command is used to write 32bit data words into the ULP memory of a connected target device. The command requires two parameters.

Syntax: wULP <pAddr>,<Data32> [Data32, ...]

Parameter: <pAddr>, <Data32>

<pAddr>

specifies the page start address of the ULP area that should be written. The page address can tale one or more pages of data to be written. No further range check is performed.

<Data32> [Data32, ...]

The 32bit data words which should be written to the specified address into the ULP memory area.

4.3.65   wULPBytes

This command is used to write 1-4 bytes to one 32bit data word into the ULP memory of a connected target device. The command requires two parameter.

Syntax: wUlpBytes <pAddr>,<Mask4>,<Data32>

Parameter: <pAddr>,<Mask4>,<Data32>

<pAddr>

specifies the page address of the ULP area that should be written. Every page address can tale one page of data to be written. One 32bit data word can be written to every page address. No further range check is performed.

<Mask4>

The bit mask for the bytes that should be written. One bit represents one byte of the 32 bit data page.

<Data32>

The 32bit data word which should be written to the specified address into the ULP memory area.

Chapter 5

Hardware

This chapter gives an overview about 2-Link MRK-III(e) specific hardware details, such as the pinouts.

5.1   Target interface connector

The target interface connector is a 10 way Insulation Displacement Connector (IDC) keyed box header (2.54mm male) that mates with IDC sockets mounted on a ribbon cable. The following table shows the definition of the 10-pin target interface connector.

5.1.1   Pinout

The following table lists the 2-Link target interface connector pinout.

PIN	SIGNAL	TYPE	Description
1	GND	-	Common ground
2	GND	-	Common ground
3	MSDA	Output	Monitor Interface Serial Data, Push-Pull or open drain
4	GND	-	Common ground
5	VBAT	Bidirec-tional	Supply Voltage (positive). Either battery from target, or supply delivered by 2-Link
6	GND	-	Common ground
7	MSCL	Input	Monitor Interface Serial Clock
8	GND	-	Common ground
9	WakeUp	Output	Target WakeUp (P10), open drain output
10	GND	-	Common ground

5.1.2   Target interface cable

The target interface connects the 2-Link and the target.
It plugs into the 10-pin IDC connector in the 2-Link on one side and into a 5-pin single in-line female connector with 2.54mm spacing. The target connector of the target interface cable is compatible to the one used with U-DDB. For proper operation, the interface cable must not exceed a total length of 300 mm. The following table shows the definition of the 10-pin target cable on the target side.

PIN	SIGNAL	TYPE	Description
1	GND	-	Common ground
2	MSDA	NC	Monitor Interface Serial Data
3	VBAT	NC	Battery Supply Voltage (positive)
4	MSCL	NC	Monitor Interface Serial Clock
5	WakeUp	NC	Target WakeUp (P10)

5.2   2-Link hardware versions

2-Link hardware is continually improved.

5.2.1   How to determine the hardware version

In order to determine the hardware version of your 2-Link, the first step should be to look at the label at the bottom side of the unit. 2-Links have the hardware version printed on the back label. If this is not the case with your 2-Link, start MRK3Link.exe. As part of the initial message it displays the hardware version.
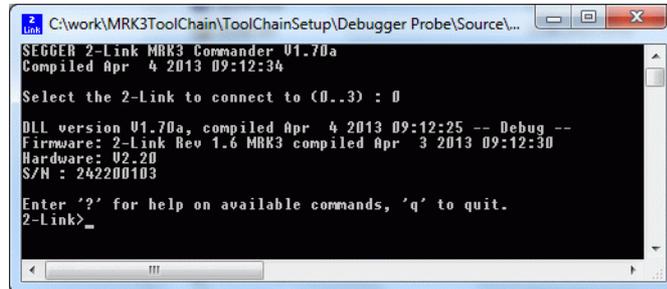
Chapter 6

Background information

This chapter provides background information about the 2-Link emulator.

6.1   2-Link firmware

The heart of 2-Link is a microcontroller. The firmware is the software executed by the microcontroller inside of the 2-Link. The firmware of 2-Link may be updated. A firmware update may be required to support new features or new devices in the future. This firmware update is performed automatically as necessary by the `MRK3Link.dll`.

6.1.1   Firmware update

Every time you connect to 2-Link, the `dll` checks if its embedded firmware is different (newer or older) than the one used in the 2-Link. It will then update/synchronize the firmware automatically. This process takes about 5 seconds and does not require a reboot. It is recommended that you always use the latest version of the `MRK3Link.dll`.

steps for a correct download of the firmware:
- start MRK3Link.exe
- select USB (e.g. 0 for USB 0)
- green LED off about 1 sec.
- green LED on about 1 sec.
- green LED flashes for about 0,5 sec.
- green LED on stable
- wait 2 sec.
- 2-Link promt in the console program
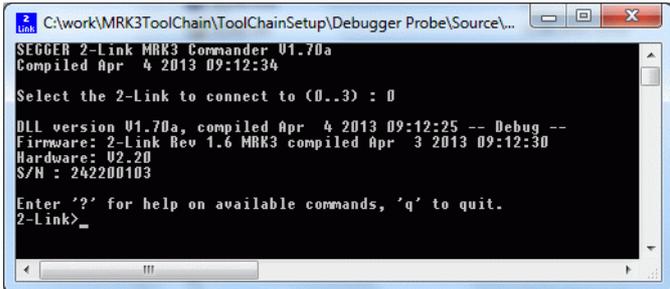
If no update is required the 2-Link prompt appears immediately after typing the USB selection number (0-3).

Chapter 7

Troubleshooting

7.1 Troubleshooting

7.1.1 General procedure

If you experience problems with a 2-Link, you should follow the steps below to solve these problems:

1. Close all running applications on your host system.
2. Disconnect the 2-Link device from USB.
3. Re-connect 2-Link with host system (attach USB cable).
4. Try your target application again. If the problem vanished, you are done; otherwise continue.
5. Close all running applications on your host system again.
6. Disconnect the 2-Link device from USB.
7. Re-connect 2-Link with host system (attach USB cable).
8. Connect your target board to 2-Link.
9. Start `2Link.exe`. The startup screen shows 2-Link version and state.
10. Select a device by command device.
11. Enter `h` to halt the connected target CPU. The `PC` is displayed and should be set to address `0000`.

7.1.2 Typical problem scenarios

2-Link LED is off

Meaning:
The USB connection does not work.
Remedy:
Check the USB connection. Try to reinitialize 2-Link by disconnecting and reconnecting it. Make sure that the connectors are firmly attached. Check the cable connections on your 2-Link and the computer. If this does not solve the problem, please check if your cable is defective. If the USB cable is ok, try a different PC.

2-Link LED is flashing at a high frequency

Meaning:
2-Link could not be enumerated by the USB controller.
Most likely reasons:
a.) Another program is already using 2-Link.
b.) The 2-Link USB driver does not work correctly.
Remedy:
a.) Close all running applications and try to reinitialize 2-Link by disconnecting and reconnecting it.
b.) If the LED blinks permanently, check the correct installation of the 2-Link USB driver. Deinstall and reinstall the driver as shown in chapter Installing the driver.

2-Link does not get any connection to the target

Most likely reasons:
Most likely reasons:
a.) The target connection cable is defective
b.) The target hardware is defective
Remedy:

First make sure that the target interface cable connectors are firmly attached and close all other running applications which may connect to the 2-Link. Check the cable connections on your 2-Link and your target hardware. If this does not solve the problem, please check the target interface connection. In most cases this indicates a problem of the target hardware. The 2-Link is a very robust hardware and works with all MRK-III(e) targets.

Before contacting the support:

Connect your 2-Link to another PC and if possible to another target system to see if it works ok. If the device functions correctly, the USB setup on the original machine or your target hardware is the source of the problem, not the 2-Link. If the 2-Link does not function correctly on a different PC or with a different target hardware, then the other system may have similar problems to the original system or the problem might be with the 2-Link.

7.2 FAQs

Q: Which CPUs are supported?
A: 2-Link can be used with any MRK-III(e) core. e.g. NCF2950 (KEyLink Lite), NCF2951 (KEyLink 3D), NCF2960 (Smart2), NCK2983 (MantraC)

Q: I want to write my own application and use 2-Link. Is this possible?
A: Yes. We offer a dedicated Software Development Kit (SDK).

Q: Can 2-Link read back the status of the connected pins?
A: Yes, the status of all pins can be read. This includes the outputs of the 2-Link as well as the supply voltage and can be useful to detect hardware problems on the target system.

Q: 2-Link is quite inexpensive. What is the advantage of some more expensive MRK-III(e) probes?
A: Some of the more expensive MRK-III(e) probes offered by other manufacturers support higher download speeds or an ethernet interface. The functionality is similar, there is no real advantage of using more expensive probes. 2-Link is a suitable solution for the majority of development tasks as well as for production purposes. Some features that are available for 2-Link, such as a DLL, exposing the full functionality of the emulator are not available for most of these emulators.

7.3 Contacting support

If you need to contact support, please send the following information to
support@segger.com
- a detailed description of the problem
- 2-Link serial number
- screen shot of console if available
- your findings of the signal analysis
- information about your target hardware (processor, board, etc)

2-Link is sold directly by SEGGER or as an OEM-product by other vendors. We can support only official SEGGER products. You can recognize a SEGGER 2-Link by the SEGGER logo on the top side of the housing.

Chapter 8

Glossary

This chapter explains important terms used throughout this manual.

Application Programmer Interface

A specification of a set of procedures, functions, data structures, and constants that are used to interface two or more software components together.

Dynamic Linked Library (DLL)

A collection of programs, any of which can be called when needed by an executing

program. A small program that helps a larger program communicate with a device such as a printer or keyboard is often packaged as a DLL.

Host

A computer which provides data and other services to another computer. Especially, a computer providing debugging services to a target being debugged.

Image

An executable file that has been loaded onto a processor for execution.

Memory coherency

A memory is coherent if the value read by a data read or instruction fetch is the value that was most recently written to that location. Memory coherency is made difficult when there are multiple possible physical locations that are involved, such as a system that has main memory, a write buffer and a cache.

Open collector

A signal that may be actively driven LOW by one or more drivers, and is otherwise passively pulled HIGH. Also known as a "wired AND" signal.

Processor Core

The part of a microprocessor that reads instructions from memory and executes them, including the instruction fetch unit, arithmetic and logic unit and the register bank. It excludes optional coprocessors, caches, and the memory management unit.

Program Status Word (PSW)

Contains some information about the current program and some information about the current processor. Often, therefore, also referred to as Processor Status Register.

Remapping

Changing the address of physical memory or devices after the application has started executing. This is typically done to allow RAM to replace ROM once the initialization has been done.

TRAP

Also known as Software Interrupt. An instruction that causes the processor to call a programmer-specified subroutine.

Target

The actual processor (real silicon or simulated) on which the application program is running.

Transistor-transistor logic (TTL)

A type of logic design in which two bipolar transistors drive the logic output to one or zero. LSI and VLSI logic often used TTL with HIGH logic level approaching +5V and LOW approaching 0V.

Watchpoint

A location within the image that will be monitored and that will cause execution to stop when it changes.

Chapter 9

Literature and references

This chapter lists documents, which we think may be useful to gain deeper understanding of technical details.

Reference	Title	Comments
1	MRK_III_e_Instruction_Set.pdf	MRK-III(e) family - architecture and instruction set

Index

Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

[NXP](#):
  [OM6718/5JP](#)