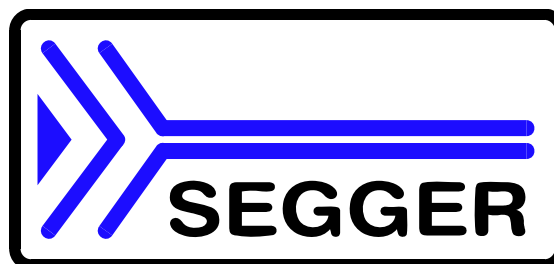


# ***2-Link User Guide***

**Revision 7**



**Emulator for  
MRK-2 Cores**



**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

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.

© 2007-2007 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 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>

## 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
7	070623	AW	Chapter 7.1.4, Restrictions for debugging in field updated
6	070621	AW	Chapter 4, new "enter" command added
5	070608	AW	Chapter 7, Restrictions, added
4	070301	AW	Modified syntax for download command described
3	070215	AW	Default option for fill byte and verify described.
2	070207	AW	Command description added
1	060908	OO	Initial version.

# Table of Contents

1	Introduction .....	5
1.1	About this document .....	6
1.2	Typographic conventions .....	6
1.3	2-Link overview .....	6
1.3.1	Features of 2-Link.....	6
1.4	Specifications .....	7
1.4.1	Specifications for 2-Link .....	7
1.5	Requirements.....	8
2	Setup.....	9
2.1	Installing the 2-Link USB driver .....	10
2.1.1	Verifying correct driver installation .....	12
2.2	Uninstalling the 2-Link USB driver.....	13
2.3	Connecting the target system.....	15
2.3.1	Power-on Sequence .....	15
2.3.2	Verifying target device connection .....	15
2.3.3	Target supply options.....	15
2.3.4	Problems .....	15
3	2-Link related software.....	17
3.1	Free software.....	18
3.1.1	2Link.dll (2-Link use with third party programs).....	18
3.1.2	2Link.exe (Command line tool) .....	20
3.2	Additional software packages .....	21
3.2.1	2-Link Software Developer Kit (SDK).....	21
4	2Link.exe reference .....	23
4.1	2Link.exe commands .....	24
4.1.1	Command overview .....	24
4.1.2	Detailed command description.....	25
5	Hardware .....	33
5.1	Target interface connector .....	34
5.1.1	Pinout .....	34
5.1.2	Target interface cable.....	34
5.2	2-Link hardware versions.....	35
5.2.1	How to determine the hardware version.....	35
6	Background information .....	37
6.1	2-Link firmware .....	38
6.1.1	Firmware update .....	38
7	Specific problems and restrictions .....	1
7.1	Problems and restrictions in general .....	2
7.1.1	Cyclic re-boot.....	2
7.1.2	Protected state.....	2
7.1.3	Entering Power OFF mode.....	2
7.1.4	Debugging in Field .....	2
7.2	Restrictions for PCF7952 running with external supply .....	3

7.2.1	Entering Monitor mode with external supply .....	3
7.2.2	Protected devices can not always be controlled. ....	3
7.2.3	Protected devices can not always be erased. ....	3
7.2.4	Devices which permanently trigger the watchdog can not allways be controlled. 3	
7.3	Restrictions for PCF7953 running with external supply .....	4
7.3.1	Entering Monitor mode with external supply .....	4
7.3.2	Protected devices can not always be controlled. ....	4
7.3.3	Protected devices can not always be erased. ....	4
7.3.4	Devices which permanently trigger the watchdog can not always be controlled. . 4	
8	Support and FAQs .....	5
8.1	Functionality test of 2-Link .....	6
8.1.1	Complete procedure .....	6
8.2	Troubleshooting .....	8
8.2.1	General procedure.....	8
8.2.2	Typical problem scenarios .....	8
8.3	Contacting support.....	9
8.4	Frequently Asked Questions.....	10
9	Glossary.....	11
10	Literature and references.....	13

# Chapter 1

## Introduction

---

This chapter gives a short overview about 2-Link.

## 1.1 About this document

This document describes 2-Link. It provides an overview over the major features of 2-Link, gives you some background information about NXP MRK-2 and describes 2-Link related software packages available from Segger. Finally, the chapter "Support and FAQs" on page 5 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 designed for NXP MRK-2 cores. It connects via USB to a PC running Microsoft Windows 2000 or XP. 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

- Any MRK-2 core supported
- No power supply required, powered through USB
- No power supply for target required, powered through 2-Link
- 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 - 3.6 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

## 1.5 Requirements

### Host System

In order to use 2-Link you need a host system running Windows 2000 or Windows XP.

### Target System (Hardware)

An MRK-2 target system is required. The system should have 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 individual pins are described in section "Target interface connector" on page 34. The individual pins of the target interface cable are described in section "Target interface cable" on page 34.



# Chapter 2

## Setup

---

This chapter describes the setup procedure required in order to work with 2-Link. This includes primarily the installation of a kernel mode 2-Link USB driver in your host system.

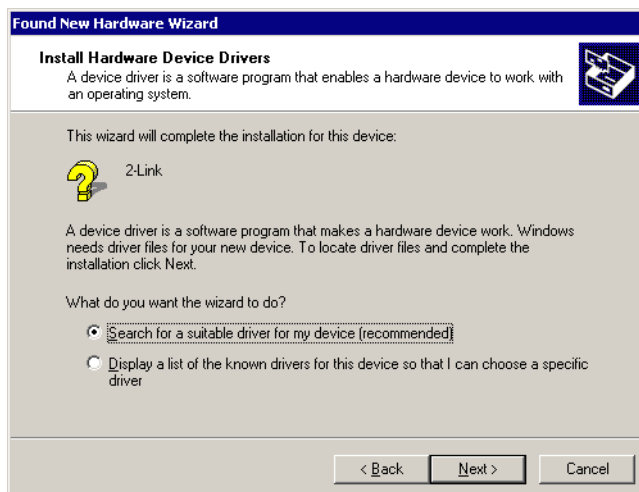
## 2.1 Installing the 2-Link USB driver

To install the 2-Link USB driver, follow this procedure.

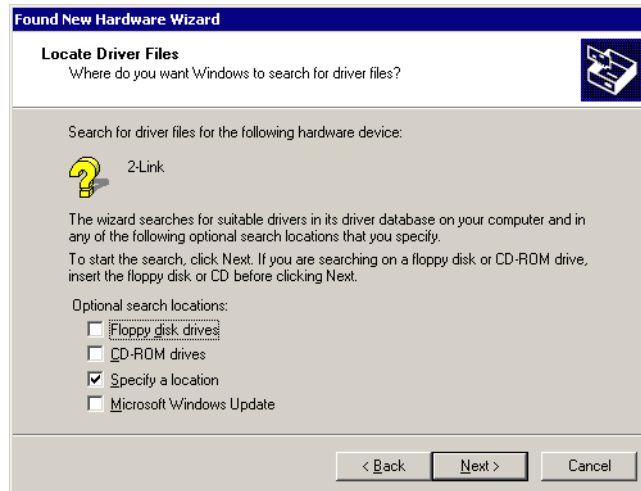
1. When your 2-Link is plugged into your computer's USB port, or when the computer is first powered on after connecting 2-Link, Windows will detect the new hardware. The wizard starts the installation of the driver.



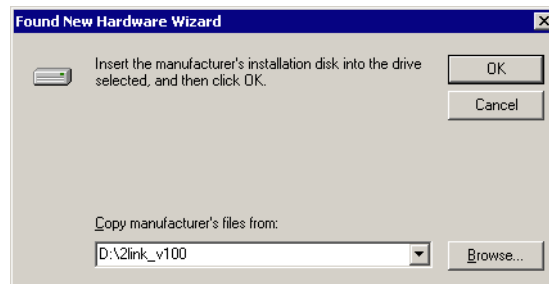
2. First select the Search for a suitable driver for my device (recommended) option, then click the Next > button.



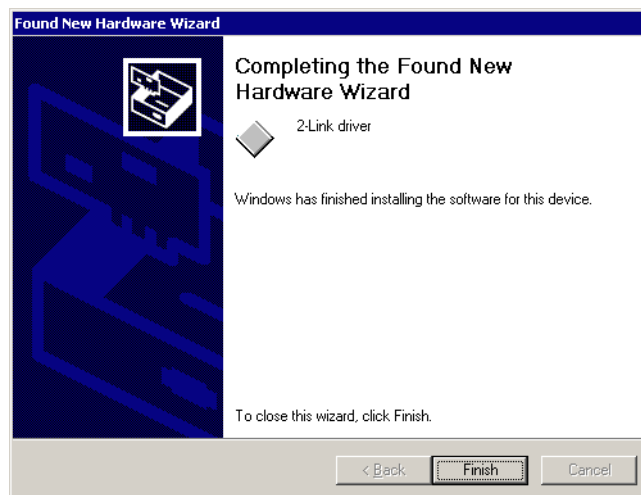
3. In the next step, you need to select the `Specify a location` option, and click the `Next >` button.



4. The wizard will ask you to specify the location of 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 `OK` button.



The wizard confirms your choice and starts to copy, when you click the `Next >` button.



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

## 2.1.1 Verifying correct driver installation

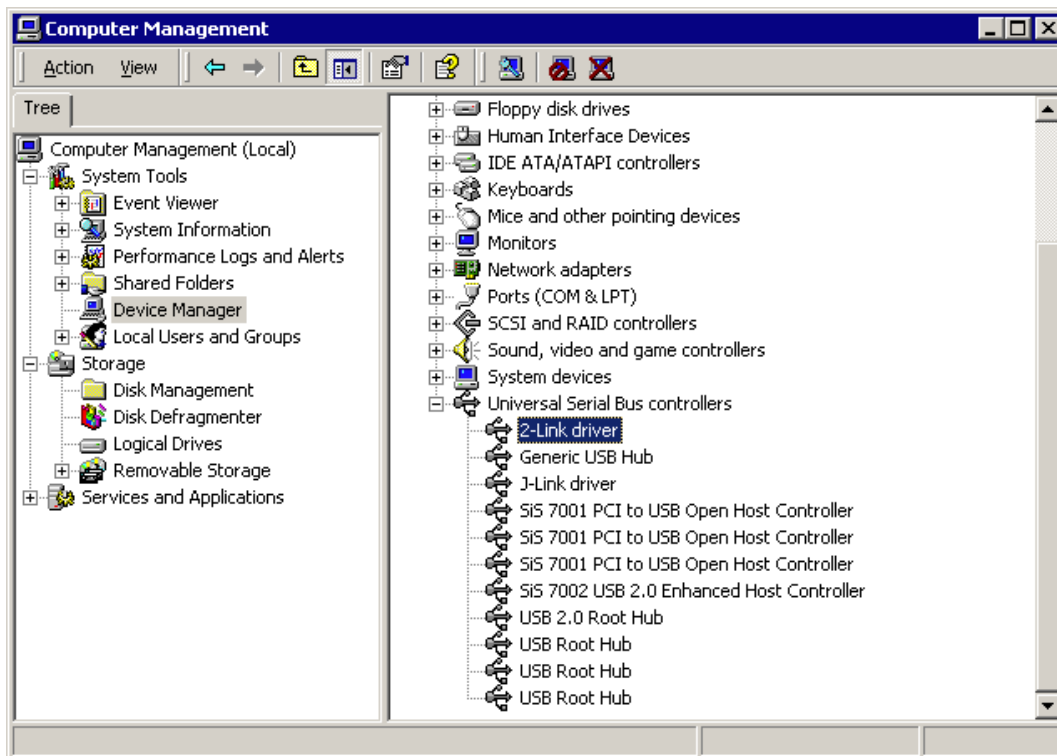
To verify the correct installation of the driver, disconnect and reconnect 2-Link to the USB port. During the enumeration process which takes about 2 seconds, the LED on 2-Link is flashing. After successful enumeration, the LED stays on permanently.

Start the provided sample application `2Link.exe`, which should display the compilation time of the 2-Link firmware and the serial number. The screenshot below should give you an idea.

```

C:\Work\2link\Shipping\2Link\2Link.exe
SEGGER 2-Link Commander V0.10c <'?' for help>
Compiled Sep 7 2006 17:47:45
DLL version V0.10c, compiled Sep 7 2006 17:47:42
Firmware: 2-Link Rev.0 compiled Aug 30 2006 17:47:33
Hardware: V0.00
S/N : -1
2-Link>_
  
```

In addition to this you may verify the driver installation by consulting 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 `Properties`. If you select this command, a new dialog box is opened and should report: `This device is working properly`. If you experience problems, have a look at chapter "Support and FAQs" on page 5 for help.

## 2.2 Uninstalling the 2-Link USB driver

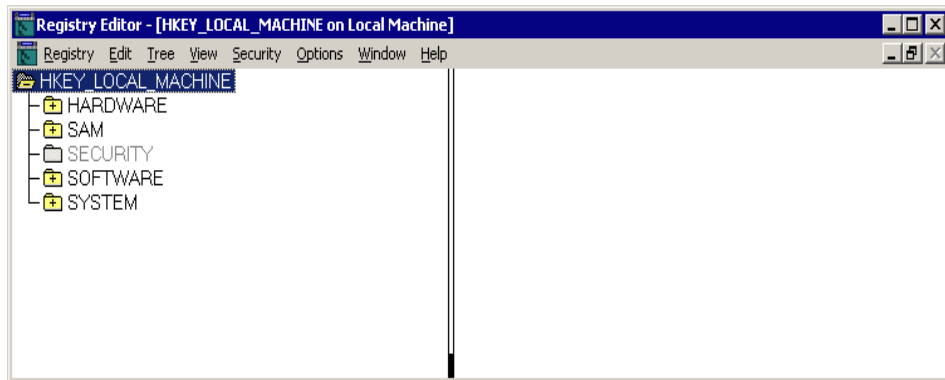
If 2-Link is not properly recognized by windows and therefore does not enumerate, it make sense to uninstall the 2-Link USB driver.

Indicators for this behavior are:

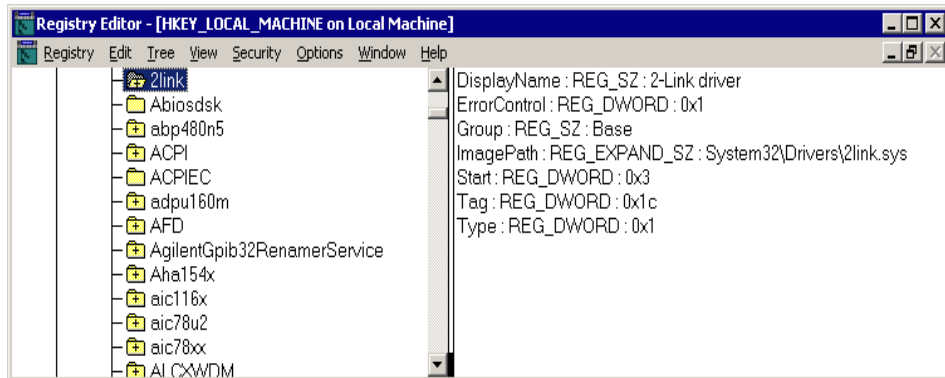
- The LED on the 2-Link is rapidly flashing.
- The 2-Link is recognized as `Unknown Device` by Windows.

To have a clean system and help Windows to reinstall the 2-Link driver, follow this procedure:

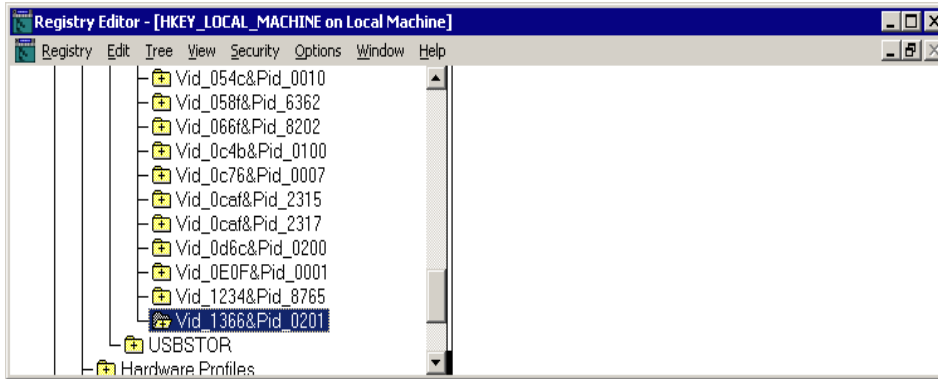
1. Disconnect 2-Link from PC.
2. Delete `2Link.inf` and `2Link.PNF` from the `\Windows\Inf` or `\Winnt\Inf` directory.
3. Delete `2Link.Sys` from the `\Windows\System32\Drivers` or `\Winnt\System32\Drivers` directory.
4. Click the `Start` button in the taskbar and select `Run...` to run a command.
5. Enter `regedt32` in the open edit field. The Registry Editor will open.
6. Select `HKEY_LOCAL_MACHINE` window.



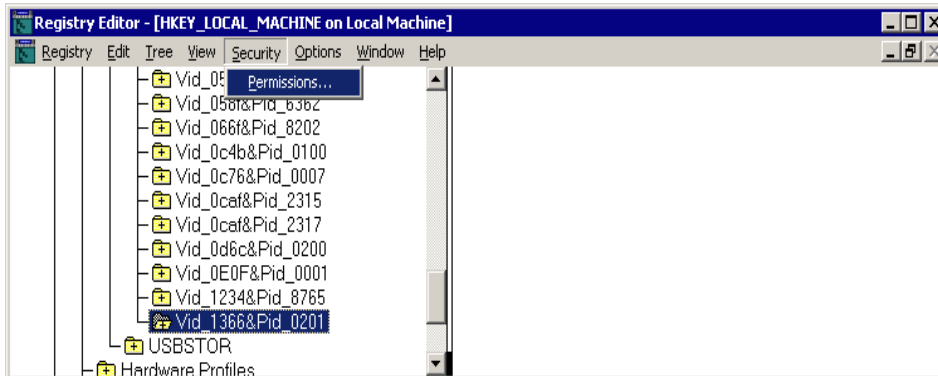
7. Select `System\CurrentControlSet\Services\2link` and delete the `2link` branch.



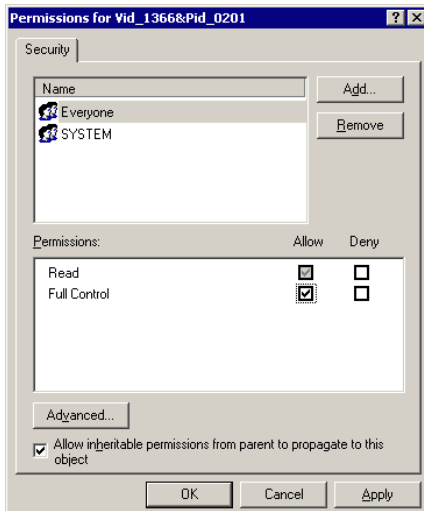
8. Select `System\CurrentControlSet\Enum\USB\Vid_1366&Pid_0201`.



9. Open the Security menu and choose Permission



10. Give group Everyone full control by clicking the checkbox Allow on Full Control.



11. Save the changes by clicking OK.  
 12. Now delete the `Vid_1366&Pid_0201` branch.  
 13. If the following branch  
`System\CurrentControlSet\Enum\USB\Vid_0000&Pid_0000`  
 is also available. Repeat the step that was made for  
`System\CurrentControlSet\Enum\USB\Vid_1366&Pid_0201`.  
 14. Close the Registry Editor window and restart the computer.

## 2.3 Connecting the target system

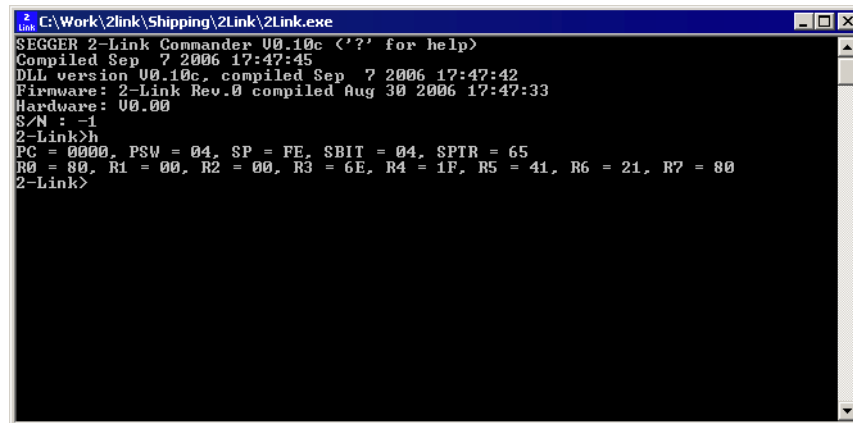
### 2.3.1 Power-on Sequence

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

### 2.3.2 Verifying target device connection

If the USB driver is working properly and your 2-Link is connected with the host system, you may connect 2-Link to your target hardware.

- Start 2Link.exe  
Connection to 2Link is identified and 2Link version and state is displayed.
- Enter device followed by the device name of the target which is connected to 2Link
- Enter h  
2-Link switches on power supply to target and switches target CPU into MONITOR mode.  
The content of the MRK-2 program counter (PC) is displayed.



```

C:\Work\2link\Shipping\2Link\2Link.exe
SEGGER 2-Link Commander V0.10c ('?' for help)
Compiled Sep  7 2006 17:47:45
DLL version V0.10c, compiled Sep  7 2006 17:47:42
Firmware: 2-Link Rev.0 compiled Aug 30 2006 17:47:33
Hardware: V0.00
S/N : -1
2-Link>h
PC = 0000, PSW = 04, SP = FE, SBIT = 04, SPTR = 65
R0 = 80, R1 = 00, R2 = 00, R3 = 6E, R4 = 1F, R5 = 41, R6 = 21, R7 = 80
2-Link>

```

### 2.3.3 Target supply options

The power supply for the target may be delivered by 2Link, or the target may be supplied externally by battery or any other supply.

#### 2.3.3.1 Target supply delivered by 2Link

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 2Link with a voltage of approximately 3.2 Volts.

#### 2.3.3.2 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 2Link does not activate its own target supply output.

### 2.3.4 Problems

If you experience problems with any of the steps described above, it is recommended you read the chapter "Support and FAQs" on page 5 for troubleshooting tips. 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 chapter "Support and FAQs" on page 5 as well.





# Chapter 3

## 2-Link related software

---

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

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

## 3.1 Free software

Any kind of free 2-Link related software is shipped together with 2-Link and may also be downloaded from [www.segger.com](http://www.segger.com). No additional license is required to use this software.

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

The 2-Link DLL is a standard Windows DLL typically used from C or C++, but also Visual Basic or Delphi projects. It makes the entire functionality of the 2-Link available through 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-2 core.

#### 3.1.1.1 Updating the DLL in third-party programs

The 2Link.dll can be used by any debugger that is designed to work with it. Although some debuggers, like the IAR C-SPY® debugger, usually ship with the 2Link.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.

#### 3.1.1.2 Updating the 2Link.dll in the IAR Embedded Workbench (EWMRKII)

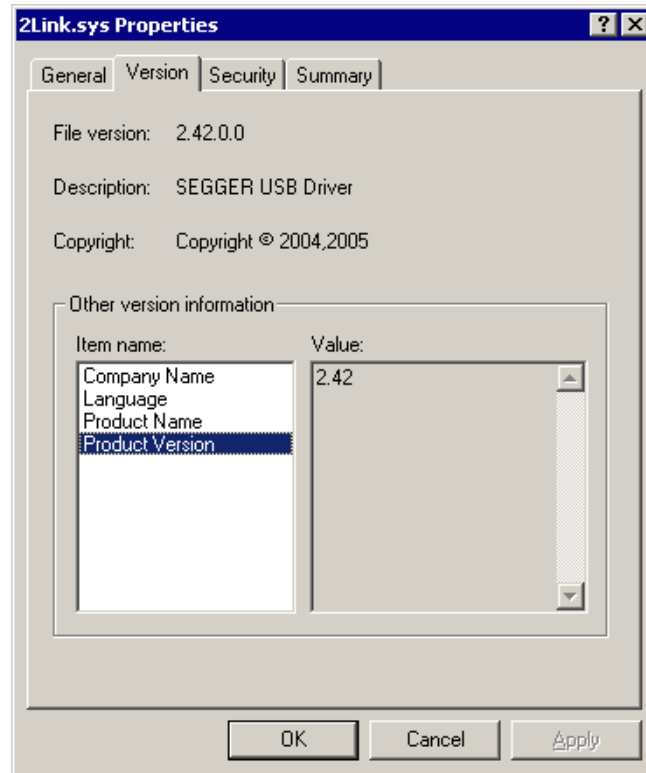
IAR Embedded Workbench is a high performance integrated development environment with editor, compiler, linker, debugger. The compiler generates very efficient code and is widely used. It comes with the 2Link.dll in the `mrkii\bin` subdirectory of the installation directory. To update this DLL, you should backup your original DLL and then replace it with the new one.

Typically, the DLL is located in `C:\Program Files\IAR Systems\Embedded Workbench 4.0\mrkii\bin\`.

After updating the DLL, it is advisable to verify that the new DLL is loaded as described in "Determining which DLL is used by a program" on page 20.

### 3.1.1.3 Determining the version of 2Link.dll

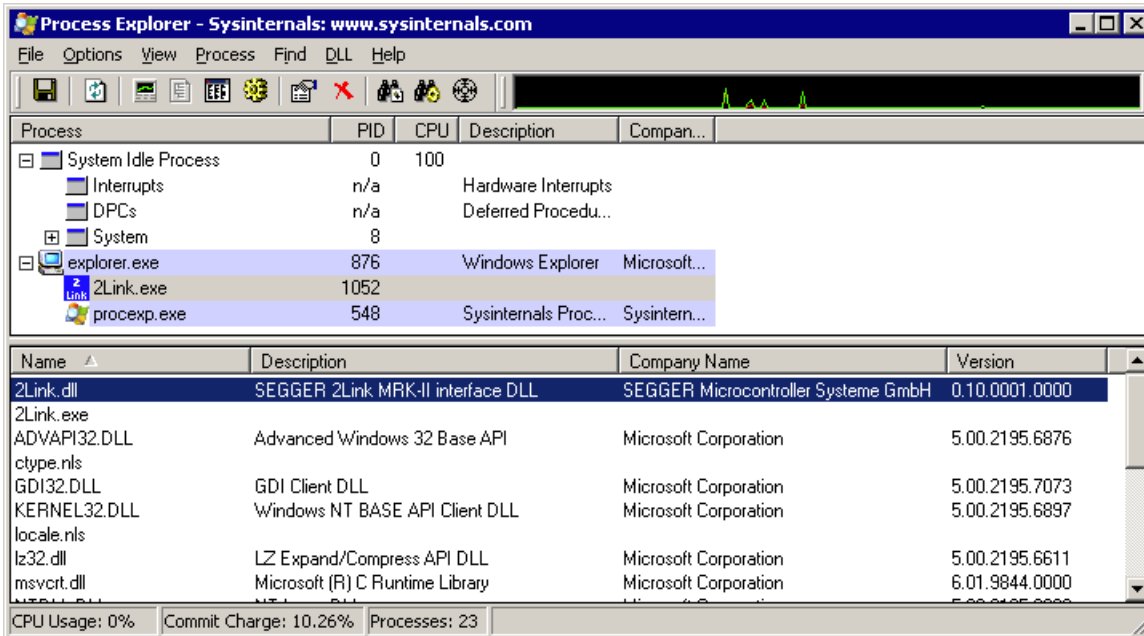
To determine which version of the 2Link.dll you are facing, the DLL version can be viewed by right clicking the DLL in explorer, and choosing Properties from the context menu. Click the Version tab to display information about the Product Version.



### 3.1.1.4 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 you 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](http://www.sysinternals.com).



## 3.1.2 2Link.exe (Command line tool)

2Link.exe is a tool that can be used to verify proper installation of the USB driver and to verify the connection to the MRK-2 chip, as well as for simple analysis of the target system. It permits some simple commands, such as halt, step, go, as well as commands to read and write all memory areas of the device.

```

C:\Work\2link\Shipping\2Link\2Link.exe
SEGGER 2-Link Commander V0.10c <'?' for help>
Compiled Sep  7 2006 17:47:45
DLL version V0.10c, compiled Sep  7 2006 17:47:42
Firmware: 2-Link Rev.0 compiled Aug 30 2006 17:47:33
Hardware: V0.00
S/N : -1
2-Link>h
PC = 0000, PSW = 04, SP = FE, SBIT = 04, SPTR = 65
R0 = 80, R1 = 00, R2 = 00, R3 = 6E, R4 = 1F, R5 = 41, R6 = 21, R7 = 80
2-Link>

```

## 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 Systeme.

### 3.2.1 2-Link Software Developer Kit (SDK)

The 2-Link Software Developer Kit is needed if you want to write your own program with 2-Link. The 2-Link DLL is a standard Windows DLL typically used from C programs (Visual Basic or Delphi projects are also possible). It makes the entire functionality of 2-Link available through its exported functions, such as halting/stepping the MRK-2 core and reading/writing CPU and TRAP registers. Therefore it can be used in any kind of application accessing an MRK-2 core. The table below lists some of the included files and their respective purpose.

Files	Contents
GLOBAL.h 2LinkDLL.h	Header files that must be included to use the DLL functions. These files contain the defines, typedef names and function declarations.
2Link.lib	Library contains the exports of the 2-Link DLL.
2Link.dll	The DLL itself.
Main.c	Sample application, which calls some 2-Link DLL functions.
2Link.dsp 2Link.dsw	Project files of the sample application. Double click 2Link.dsw to open the project.
2LinkDLL.pdf	Extensive documentation (API, sample projects etc.).



# Chapter 4

## 2Link.exe reference

---

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

## 4.1 2Link.exe commands

2Link.exe has several commands to access the connected target device. 2Link.exe may be used in three different modes which support all of the commands. In interactive mode, all commands are just type on the console. In script mode, all commands are read from a text file. In command line mode, commands are passed to 2Link.exe as parameter.

### 4.1.1 Command overview

The table below lists the available commands of 2Link.exe. All commands are listed in alphabetical order.

Routine	Explanation
Clk 1..5	Set target CPU speed
ClrBP	Clears breakpoint.
Device	Select a device
download <memory>, <filename> [,<address>[,<size>]] [,<start>-<end>]	Download a file to target
Enter	Unconditionally enter monitor mode
Erase	Unprotect target and erase target EROM memory
fill [<hex>]	Set a fill byte for following downloads
g	Starts the MRK-2 core.
h	Halts the MRK-2 core.
IsHalted	Checks if the MRK-2 core is halted.
iOff	Unconditionally switches the interface to target into High-Z mode.
IsProtected	Checks if the MRK-2 core is in protected mode.
List	Displays a list of supported devices.
log	Activates a log file for 2Link
nofill	Disables a fill byte for following downloads.
noverify	Disables verify for following downloads.
powerdown	Switches the power supply to target on
powerup	Switches the power supply to target off
Protect	Sets the MRK-2 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.
s	Single steps the MRK-2 core.
SetBP	Sets breakpoint.
SetPC	Sets PC register.
SetWP	Sets watchpoint.
sig	Displays all target signatures
upload <memory>, <filename> [,<address>[,<size>]]	Uploads data from target and stores it to a file.
verify	Enables verify for following download
wDat	Writes data area.
wEEPROM	Writes EEPROM area.
wEROM	Writes EROMS area.



## 4.1.2 Detailed command 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 2Link.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.

Before the target can be accessed, the device selection command has to be called.

If a command requires parameter, the parameter has to be separated from the command by a space character. Some commands require more parameters, which have to be separated by comma.

### 4.1.2.1 Clk (Select the operating frequency of the target device)

The clock command is used to select the operating clock of the connected target CPU. The command takes one parameter.

**Syntax:** `clk <speed>`

**Parameter:** `<speed>`

Valid speed parameter: 1..5

1: selects 125 kHz

2: selects 250 kHz

3: selects 500 kHz

4: selects 1 MHz

5: selects 2 MHz.

If no clk command is used, the target CPU operation speed is automatically set to 1MHz after Monitor mode is entered.

**Example:** `clk 1`

### 4.1.2.2 ClrBP (Clear a breakpoint)

Clears a breakpoint which was set by a call of SetBP or SetWP. The command takes one parameter.

**Syntax:** `ClrBP <BPHandle>`

**Parameter:** `<BPHandle>`

The BPHandle is a positive number which names the breakpoint to be deleted. It is the breakpoint handle which was returned from a SetBP or SetWP command.

**Example:** `ClrBP 1`

### 4.1.2.3 Device (Select a device)

This command is used to select a target device. Before any command which accesses the target device is called, the target device has to be selected by the device command. This is required to inform 2Link about the correct mode to enter the MONITOR mode as well as to perform range checks for other commands.

The command takes one parameter which is the name of the target device. The device name is case insensitive.

**Syntax:** `device <name>`

**Parameter:** `<name>`

The name has to be a valid name of a supported device.

**Example:** `device PCF7953`

A list of available device names can be displayed by the "List" command.

#### 4.1.2.4 download (Download a file to the target device)

The download command is used to write data into a target device memory area. The data is read from a file.

The download command takes a various number of parameters.

**Syntax:** `download <memory>, <filename>[,<address>[,<size>]][, <start>-<end>]`

**Parameter:** `<memory>, <filename>, <address>, <size>, <start>, <end>`

The memory and filename parameters are required, the others may be omitted.

**<memory>:**

specifies the target memory area which should be used by download. Valid values for memory are:

DATA, EEPROM, EROM

**<filename>**

specifies the file to download. The file may be an INTEL-Hex file or a binary file. If the file name is entered without extension, or with extension "hex", an Intel-Hexfile is assumed, otherwise, the file is handled as binary file.

**<address>**

is the start address of the loaded file. If the address parameter is omitted, the start address is taken from the loaded file.

For binary files, the address parameter is required.

For Hex-files, the address parameter is not allowed.

**<size>**

is the number of bytes which should be written. If the parameter is omitted, the size is taken from the file which should be downloaded. If a fill byte was specified, the memory area may be filled up to the last address, if the file contains less data and a size parameter is not specified.

**<start>**

The start parameter may be used together with the "end" parameter to define a window which limits the range of data which is actually downloaded to the target CPU. It has to be used together with the "end" parameter.

The start parameter is the address of the first byte which should be downloaded to the target CPU.

**<end>**

The end parameter may be used together with "start" parameter to define a window which limits the range of data which is actually downloaded to the target CPU. It has to be used together with the "start" parameter and has to be separated from the start parameter by a hyphen "-".

The end parameter is the address of the last byte which should be downloaded to the target device. If the start and end parameter are omitted, the addresses are taken from the address and size parameter, or from the loaded file.

**Example:** `download EROM,c:\mrk2test.bin,0,100,10-100`

#### 4.1.2.5 Enter (Enter monitor mode unconditionally)

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 issues the enter monitor sequence automatically whenever this is required.

It's primary use is for hardware analysis and debugging purposes.

**Syntax:** `enter`

A device has to be selected before the enter command can be used, because the enter monitor sequence to use depends on the device which is connected to 2Link.

#### 4.1.2.6 Erase (Erase the connected target device)

This command is used to erase the connected device. If the device was protected, the protection is automatically removed. The EROM area and EEPROM area are erased. The command takes no parameter.

**Syntax: erase**

For an PCH797x device, the EEPROM area is not erased.

#### 4.1.2.7 fill (Set a fill byte for download)

This command is used to set a fill byte which is used to fill memory areas on subsequent downloads. The fill byte is automatically inserted in those gaps which are not included in the file to download. The fill byte remains valid for all subsequent downloads, until a "nofill" command is given. "nofill" is default for downloads. The command may take one parameter.

**Syntax: fill [<hex>]**

**Parameter: <hex>**

The parameter has to be a hex value from 0x00 to 0xFF.  
If the parameter is omitted, a default fill byte of 0x00 is used.

**Example:fill 0xAA**

#### 4.1.2.8 g (Go, start the target device)

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 without executing any instruction. To continue execution from a breakpoint, issue a single step command first before sending a go command.

#### 4.1.2.9 h (Halt the target device)

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 should be the first command after device selection.

The command takes no parameter.

**Syntax: h**

#### 4.1.2.10 IsHalted (Check whether target device is halted)

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

#### 4.1.2.11 iOff (Switch interface to target device to High-Z mode)

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.

#### 4.1.2.12 IsProtected (Check whether target device is protected)

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

**4.1.2.13 List (List all supported devices)**

This command is used to get a list of the device names of all devices which are supported by 2Link. The command takes no parameter.

**Syntax: List**

The list of supported devices is displayed on the screen. Only device names listed here are valid for the "device" command.

**4.1.2.14 log (Activate logging to a file)**

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.

**4.1.2.15 nofill (Disable fill byte for download)**

This command disables usage of a fill byte on subsequent downloads. The command remains valid until a fill command is given.

Nofill is the default mode for downloads.

The command takes no parameter.

**Syntax: nofill****4.1.2.16 noverify (Disable verify for download)**

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****4.1.2.17 powerdown (Switch off power supply to target)**

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**

This command may also be used to control the behavior on exit. After issuing a powerdown command, the target is switched off when 2Link.exe is closed.

**4.1.2.18 powerup (Switch on power supply to target)**

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

**Syntax: powerup**

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

#### 4.1.2.19 Protect (Protect the target device)

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

**Syntax: protect**

#### 4.1.2.20 q (Quit, exit 2Link.exe)

This commands ends 2Link.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.

#### 4.1.2.21 r (Reset and halt the target device)

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**

#### 4.1.2.22 rDat (Read the data area of the target device)

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. A range check is performed. If the start address is larger than the last RAM address of the selected device, the function reports an error.

**<Size>**

specifies the number of bytes which should be read from the data area, starting at the start address given before. A range check of address and size according to the selected device is automatically performed. If any reserved addresses within the specified data area shall be read, the value 0x00 is sent back for those reserved addresses. Regardless of the size parameter, no read access is performed behind the last RAM address of the selected device.

Even though the CPU registers could be read by the rDat command, the CPU registers should be read by the regs command.

#### 4.1.2.23 rEEPROM (Read the EEPROM area of the target device)

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. A range check is performed. If the start address is larger than the last EEPROM address of the selected device, the function reports an error.

**<Size>**

specifies the number of bytes which should be read from the EEPROM area, starting at the start address given before. A range check of address and size according to the selected device is automatically performed. Regardless of the size parameter, no read access is performed behind the last EEPROM address of the selected device.

**4.1.2.24 regs (Display the registers of the target device)**

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

**4.1.2.25 rEROM (Read the EROM area of the target device)**

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

**Syntax:** rDat <Addr>,<Size>

**Parameter:** <Addr>, <Size>

**<Addr>**

specifies the start address of the EROM area that should be read. A hex value has to be typed. A range check is performed. If the start address is larger than the last EROM address of the selected device, the function reports an error.

**<Size>**

specifies the number of bytes which should be read from the EROM area, starting at the start address given before.

A range check of address and size according to the selected device is automatically performed. Regardless of the size parameter, no read access is performed behind the last RAM address of the selected device.

**4.1.2.26 s (Single step the target device)**

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

**4.1.2.27 SetBP (Set a breakpoint)**

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 <Addr>

**Parameter:** <Addr>

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

The function returns a positive number, if the breakpoint could be set. This number (BPHandle) has to be used as parameter for ClrBP, if the breakpoint should be removed later.

**4.1.2.28 SetPC (Set the program counter of the target device)**

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. 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 an even Hex number in the target CPU EROM address space.

#### 4.1.2.29 SetWP (Set a watchpoint)

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

**Syntax: SetWP <Addr>**

**Parameter: <Addr>**

The address parameter is the address where the watchpoint should be set to. It has to be an even Hex number in the target CPU EROM address space.

The function returns a positive number, if the watchpoint could be set. This number (BPHandle) has to be used as parameter for ClrBP, if the watchpoint should be removed later.

#### 4.1.2.30 sig (Display all signatures of the target device)

This command is used to display all signatures of the connected device. The command takes no parameter.

**Syntax: sig**

2Link.exe automatically determines which signatures can be read from the connected device and lists all those signatures on the screen.

#### 4.1.2.31 upload (Read from the target device and write to a file)

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 takes a various number of parameters.

**Syntax: download <memory>, <filename>[,<address>[,<size>]]**

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

The memory and filename parameters are required, the others may be omitted.

**<memory>:**

specifies the target memory area which should be read by upload. Valid values for memory are:

DATA, EEPROM, EROM

**<filename>**

specifies the file where the data should be written to. The file may be an INTEL-Hex file or a binary file. If the file name is entered without extension, or with extension "hex", an Intel-Hexfile is assumed, otherwise, the file is handled as binary file.

**<address>**

is the start address of the target memory area. If the address parameter is omitted, the start address is assumed to be 0.

**<size>**

is the number of bytes which should be read from target and written to file. If the parameter is omitted, the size is taken from the device definition. The whole memory area starting from the selected start address is read.

**Example: upload EROM,c:\mrk2test.bin,0**

#### 4.1.2.32 verify (Activate verify after download)

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**

#### 4.1.2.33 wDat (Write into the data area of the target device)

This command is used to write one byte of data at the specified address. The data area also includes the CPU registers and special function registers.

The command requires two parameter.

**Syntax: wDat <Addr>,<Data>**

**Parameter: <Addr>, <Data>**

**<Addr>**

specifies the address of the data area where the data should be written to. A hex value has to be typed. A range check is performed. If the address is larger than the last RAM address of the selected device, 2Link.exe reports an error.

**<Data>**

specifies the data byte which should be written to the specified address. A hex value form 0x00 to 0xff has to be typed.

#### 4.1.2.34 wEEPROM (Write into the EEPROM area of the target device)

This command is used to write one byte of data into the EEPROM memory of the connected target device. The command requires two parameter.

**Syntax: wEEPROM <Addr>,<Data>**

**Parameter: <Addr>, <Data>**

**<Addr>**

specifies the address of the EEPROM area where the data byte should be written to. A hex value has to be typed. A range check is performed. If the address is larger than the last EEPROM address of the selected device, or a restricted or protected EEPROM location is addressed, 2Link.exe reports an error.

**<Data>**

The data byte which should be written to the specified address into the EEPROM area.

#### 4.1.2.35 wEROM (Write into the EROM area of the target device)

This command is used to write one data byte into the EROM memory of a connected target device. The command requires two parameter.

**Syntax: wEROM <Addr>,<Data>**

**Parameter: <Addr>, <Data>**

**<Addr>**

specifies the start address of the data area that should be read. A hex value has to be typed. A range check is performed. If the start address is larger then the last EROM address of the selected device, 2Link.exe reports an error.

**<Data>**

The data byte which should be written to the specified address into the EROM area.



# Chapter 5

## Hardware

---

This chapter gives an overview about 2-Link 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	Bidirectional	Supply Voltage (positive). Either battery from target, or supply delivered by 2Link
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)

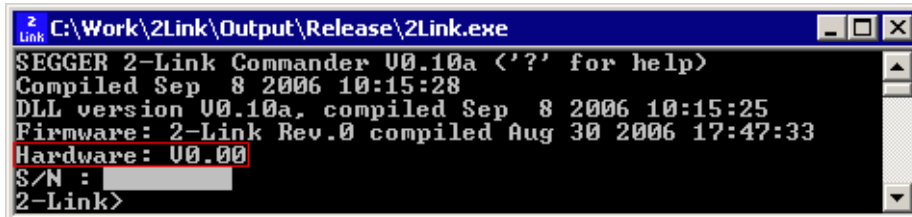
## 5.2 2-Link hardware versions

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 `2Link.exe`. As part of the initial message it displays the hardware version.



```
C:\Work\2Link\Output\Release\2Link.exe
SEGGER 2-Link Commander V0.10a <'?' for help>
Compiled Sep 8 2006 10:15:28
DLL version V0.10a, compiled Sep 8 2006 10:15:25
Firmware: 2-Link Rev.0 compiled Aug 30 2006 17:47:33
Hardware: V0.00
$/N :
2-Link>
```



# Chapter 6

## Background information

---

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

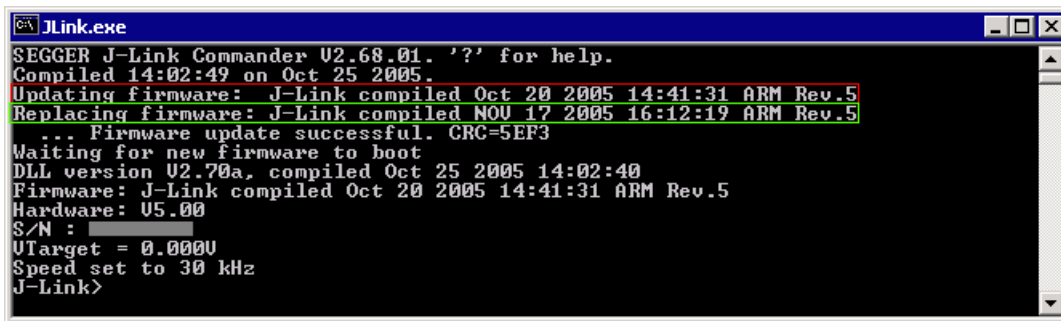
## 6.1 2-Link firmware

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 2Link.dll.

### 6.1.1 Firmware update

Every time you connect to 2-Link, 2Link.dll checks if its embedded firmware is newer than the one used in the 2-Link. It will then update the firmware automatically. This process takes less than 3 seconds and does not require a reboot.

It is recommended that you always use the latest version of 2Link.dll.



```
JLink.exe
SEGGER J-Link Commander V2.68.01. '?' for help.
Compiled 14:02:49 on Oct 25 2005.
Updating firmware: J-Link compiled Oct 20 2005 14:41:31 ARM Rev.5
Replacing firmware: J-Link compiled NOV 17 2005 16:12:19 ARM Rev.5
... Firmware update successful. CRC=5EF3
Waiting for new firmware to boot
DLL version V2.70a, compiled Oct 25 2005 14:02:40
Firmware: J-Link compiled Oct 20 2005 14:41:31 ARM Rev.5
Hardware: 05.00
S/N : 
UTarget = 0.0000
Speed set to 30 kHz
J-Link>
```

The red box identifies the new firmware.

The green box identifies the old firmware which has been replaced.

# Chapter 7

## Specific problems and restrictions

---

This chapter describes problems and restrictions which exist for specific MRKII devices under various operating conditions.

## 7.1 Problems and restrictions in general

In general, the following problems may occur.

### 7.1.1 Cyclic re-boot

Depending on configuration and program memory, some devices perform a periodic re-boot.

To enter debug mode, or to halt the target, it is required to examine the state of the device by reading the state of the SDA and SCL signals.

If the target device re-boots, the device state and state of the signals may change after they were read. Halting the target may fail, because the target is in an unexpected state.

Targets with wake signal pull the SCL line low during cyclic re-boot. When the debugger detects a low signal on SCL, it assumes the target is halted. When the target then re-boots and is accessed by the debugger, the access fails. The debugger then tries to reset the target and re-enter monitor mode.

PCF7952 and PCF7953 devices output a small pulse on the SDA line when they re-boot. This signal can be detected by the debugger and a special enter monitor sequence is used to access the target.

### 7.1.2 Protected state

Targets which are protected can not always be controlled.

In general, the following restrictions exist for all devices:

- A running target can not be halted
- A target executing an endless loop and running from battery can not be controlled.

### 7.1.3 Entering Power OFF mode

Depending on device configuration or application, the target may enter power off mode after the application was started by the debugger.

Examining the state on the signal lines, the debugger assumes, the target is halted, and the debugger tries to read the PC register. This operation fails, because the target left the monitor mode, and the debugger will issue an error message.

The device will be reset by the next access.

### 7.1.4 Debugging in Field

Debugging in field requires special device setup and an application which contains valid code for the NMI routine. If the target application does not contain these settings or interrupt handler, the target may execute internal ROM routines and can not be accessed.

For all devices, the following conditions have to be met to allow debugging in field:

- The device has to be powered internally by 2Link or external battery supply.
- A breakpoint has to be set.
- The device must enter RUN mode (must not be in Monitor mode or in Power-Off mode when LF Field is applied).
- Debugging in Field must be enabled by configuration setting, Transponder Emulation (TEN) must be enabled.

When these conditions are set, the LF field may be applied to access the device for debugging in field.



## **7.2 Restrictions for PCF7952 running with external supply**

### **7.2.1 Entering Monitor mode with external supply**

To control a PCF7952 with external supply, the following conditions are required:

- Debug features have to be enabled by programming the value 0x00 at address 0xFFE of the EROM memory. Older devices may require the value 0xA5.
- The device enters power off mode by application and does not execute an endless loop which triggers the watchdog.

### **7.2.2 Protected devices can not always be controlled.**

- Protected devices can not be halted if the target executes a program. This is a general restriction for all devices.
- If the target executes a program in an endless loop that triggers the watchdog, the target can not be accessed or halted when it is protected. Regardless debug features are enabled or not.

To access a protected device that runs on battery, it is required that the device enters power off mode.

### **7.2.3 Protected devices can not always be erased.**

Protected devices can not be erased if they can not be accessed. The restrictions are the same as described above:

- Protected devices can not be halted if the target executes a program. When the target can not be halted, it can not be erased.
- If the target executes a program in an endless loop that triggers the watchdog, the target can not be accessed or halted and therefore can not be erased.

### **7.2.4 Devices which permanently trigger the watchdog can not always be controlled.**

Devices which permanently trigger the watchdog in an endless loop do not react on any enter monitor sequence and can not be halted, if debug features are not enabled. To enable debug features, the value 0x00 has to be programmed at address 0xFFE (DBG\_CFG). This is default after erasure of the device.

Please note, that older devices may require the value 0xA5 programmed at DBG\_CFG.

## 7.3 Restrictions for PCF7953 running with external supply

### 7.3.1 Entering Monitor mode with external supply

To control a PCF7953 with external supply, the following conditions are required:

- Debug features have to be enabled by programming the value 0x00 at address 0x1FFE of the EROM memory.
- The device enters power off mode by application and does not execute an endless loop which triggers the watchdog.

### 7.3.2 Protected devices can not always be controlled.

- Protected devices can not be halted if the target executes a program. This is a general restriction for all devices.
- If the target executes a program in an endless loop that triggers the watchdog, the target can not be accessed or halted when it is protected. Regardless debug features are enabled or not.
- If the target is erased and protected, it can not be halted or accessed if it is running on battery supply. Do not protect an empty device!

### 7.3.3 Protected devices can not always be erased.

Protected devices can not be erased if they can not be accessed. The restrictions are the same as described above:

- Protected devices can not be halted if the target executes a program. When the target can not be halted, it can not be erased.
- If the target executes a program in an endless loop that triggers the watchdog, the target can not be accessed or halted and therefore can not be erased.
- If the target is erased and protected, it can not be halted or accessed if it is running on battery supply. The target can not be erased and can therefore not be unprotected.

### 7.3.4 Devices which permanently trigger the watchdog can not always be controlled.

Devices which permanently trigger the watchdog in an endless loop do not react on any enter monitor sequence and can not be halted, if debug features are not enabled. To enable debug features, the value 0x00 has to be programmed at address 0x1FFE (DBG\_CFG). This is default after erasure of the device.

# Chapter 8

## Support and FAQs

---

This chapter contains troubleshooting tips together with solutions for common problems which might occur when using 2-Link. There are several steps you can take before contacting support. Performing these steps can solve many problems and often eliminates the need for assistance. This chapter also contains a collection of frequently asked questions (FAQs) with answers.

## 8.1 Functionality test of 2-Link

### 8.1.1 Complete procedure

The functionality of 2-Link hardware and software may be verified by the following steps:

1. Connect an PKE Actic-PRO board with PCF79531 CPU to 2-Link.
2. Connect 2-Link via USB to PC.  
2-Link starts up, the LED flashes and then stays steady after approx. 2 seconds when the enumeration is done.
3. Start `2Link.exe`.  
The Start-up screen shows 2-Link DLL-version and also state and version of 2-Link hard- and software.
4. Enter: `device PCF7953`  
2-Link executes the device selection and responds with a message which shows the index of the currently selected device.
5. Enter: `h`  
2-Link halts the connected target CPU and the program counter register (PC) content is displayed. The PC should be set to 0.
6. Enter: `download EROM,C:\mrk2test.bin,0`  
2-Link programs the File into the EROM memory starting from address 0.
7. Enter: `rerom 0,30`  
The EROM memory is read back, the result is shown on the screen and can be verified.  
The sample application `MRK2Test.txt` contains the following 18 bytes:  
01 80 10 78 1B C4 00 EA 02 80 02 78 A6 70 08 80
8. Enter: `s`  
A single-step is performed, the PC value advances to 0002 and is displayed.
9. Enter: `s`  
Another single-step is performed, the PC advances to 0004
10. Enter: `regs`  
The registers are read from target, R0 was set to 10 by the last single-step, PC shows 0004, all other registers remained unchanged.
11. Enter: `SetBP 8`  
A breakpoint is set at address 8, 2-Link.exe displays  
Breakpoint set at Addr. 0008. Handle: 00000001
12. Enter: `g`
13. Enter: `h`  
Target CPU reached the breakpoint, the registers are displayed. PC = 0008, R0 = 0F, all other registers unchanged.
14. Enter: `ClrBP 1`  
The breakpoint with handle 1 is removed. 2-Link.exe shows:  
Breakpoint removed
15. Enter: `g`
16. Enter: `h`  
The CPU is halted. As the breakpoint was removed before, the PC value reflects the current value when execution was interrupted by a break command from 2-Link. R0 has any value, all other registers remained unchanged.
17. Enter: `rEEPROM 20,10`  
The content of EEPROM from address 20 to 2F is displayed
18. Enter: `wEEPROM 20,20`  
2-Link writes the value 20 into the EEPROM memory at address 20. 2-Link.exe shows the following message:  
Writing 20 -> 00000020
19. Enter: `rEEPROM 20,10`  
The EEPROM memory is read back to verify the write operation. The content of EEPROM from address 20 to 2F is displayed, address 20 now has value 20.
20. Enter: `rDat 40,10`  
The content of Data area from address 40 to 4F is displayed.
21. Enter: `wDat 40,40`  
This writes the value 40 into the Data area (RAM) of the target CPU. 2-Link.exe

shows:

Writing 40 -> 00000040

22. Enter: SetPC 0

23. Enter: s

A single-step is performed. Because the PC was set to 0 before, a PC value of 0002 is displayed.

24. Enter: rDat 40,10

The content of the Data area from address 40 to 4F is read back from target and displayed, RAM at address 40 contains value 40, other RAM locations are unchanged.

25. Enter: q

Application is closed, power supply to target is switched off

The following screenshot shows the complete test-procedure:

```

SEGGER 2-Link Commander V0.10b ('?' for help)
Compiled Sep  7 2006 15:06:25
DLL version V0.10b, compiled Sep  7 2006 15:06:22
Firmware: 2-Link Rev.0 compiled Aug 30 2006 17:47:33
Hardware: U0.00
S/N : -1
2-Link>h
PC = 0000, PSW = 05, SP = FE, SBIT = 04, SPTR = 65
R0 = 00, R1 = 00, R2 = 00, R3 = 6E, R4 = 1F, R5 = 41, R6 = 21, R7 = 80
2-Link>loadbin C:\mrk2test.txt, 0
Loading binary file... [C:\mrk2test.txt]
Writing bin data into target memory @ 0x00000000.
2-Link>regs 0,30
00000000 = 01 80 10 78 1B C4 00 E0 02 80 02 80 02 78 A6 70
00000010 = 08 80 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000020 = 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
2-Link>s
PC: 0002
2-Link>s
PC: 0004
2-Link>regs
PC = 0004, PSW = 05, SP = FE, SBIT = 04, SPTR = 65
R0 = 10, R1 = 00, R2 = 00, R3 = 6E, R4 = 1F, R5 = 41, R6 = 21, R7 = 80
2-Link>SetBP 8
Breakpoint set at Addr 0008. Handle: 00000001
2-Link>g
2-Link>h
PC = 0008, PSW = 05, SP = FE, SBIT = 04, SPTR = 65
R0 = 0F, R1 = 00, R2 = 00, R3 = 6E, R4 = 1F, R5 = 41, R6 = 21, R7 = 80
2-Link>ClrBP 1
Breakpoint removed
2-Link>g
2-Link>h
PC = 0008, PSW = 05, SP = FE, SBIT = 04, SPTR = 65
R0 = 10, R1 = 00, R2 = 00, R3 = 6E, R4 = 1F, R5 = 41, R6 = 21, R7 = 80
2-Link>*EEPROM 20,10
00000020 = 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
2-Link>wEEPROM 20,20
Writing 20 -> 00000020
2-Link>*EEPROM 20,10
00000020 = 20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
2-Link>rDat 40,10
00000040 = 0C 00 03 20 19 B1 80 20 00 85 14 01 49 02 00 00
2-Link>wDat 40,40
Writing 40 -> 00000040
2-Link>SetPC 0
2-Link>s
PC: 0002
2-Link>rDat 40,10
00000040 = 40 00 03 20 19 B1 80 20 00 85 14 01 49 02 00 00
2-Link>_

```

The file MRK2Test.bin which is loaded into the target CPU was built from the following source file:

```

?PR?boot SEGMENT CODE AT 0000h
ASEG ?PR?boot
0000    jmp main
?PR?main SEGMENT CODE
RSEG ?PR?main
main:
0002    mov R0, #10h
loop:
0004    setb WCLR ;reset watchdog
0006    dsz R0
0008    jmp loop
000A    jmp loop ; may be replaced (patched) by NOP instruction (0xFFFF)
                ; to continue with addr 0x0C
000C    mov R0, #02h
000E    mov PCON, R0 ;power off
0010    jmp $ ;infinite loop until power off
END

```

## 8.2 Troubleshooting

### 8.2.1 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`.

```

C:\Work\2link\Shipping\2Link\2Link.exe
SEGGER 2-Link Commander V0.10c (<'?' for help>)
Compiled Sep  7 2006 17:47:45
DLL version V0.10c, compiled Sep  7 2006 17:47:42
Firmware: 2-Link Rev.0 compiled Aug 30 2006 17:47:33
Hardware: V0.00
SW: -1
2-Link>h
PC = 0000, PSW = 04, SP = FE, SBIT = 04, SPTR = 65
R0 = 80, R1 = 00, R2 = 00, R3 = 6E, R4 = 1F, R5 = 41, R6 = 21, R7 = 80
2-Link>
  
```

### 8.2.2 Typical problem scenarios

#### 2-Link LED is off

*Meaning:*

The USB connection does not work.

*Remedy:*

Check the USB connection. Try to re-initialize 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, check if your cable is defective. If the USB cable is ok, try a different host computer.

#### 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. De-install and reinstall the driver as shown in chapter "Setup" on page 9.

## **2-Link does not get any connection to the target**

*Most likely reasons:*

- a.) The target connection cable is defective
- b.) The target hardware is defective

*Remedy:*

Follow the steps described in section 8.2.1.

## **8.3 Contacting support**

Before contacting support, make sure you tried to solve your problem by following the steps outlined in section "General procedure" on page 8. You may also try your 2-Link with another PC and if possible with another target system to see if it works there. If the device functions correctly, the USB setup on the original machine or your target hardware is the source of the problem, not 2-Link.

If you need to contact support, send the following information to [support@segger.com](mailto:support@segger.com):

- A detailed description of the problem
- 2-Link serial number
- Output of `2Link.exe` if available
- Your findings of the signal analysis
- Information about your target hardware (processor, board, etc.).

## 8.4 Frequently Asked Questions

Q: Which CPUs are supported?

A: 2-Link should work with any MRK-2 core. The device selection accepts the following device: PCF7952, PCF7953, PCF7941, PCF7961, PCH7970.

Q: I want to write my own application and use 2-Link. Is this possible?

A: Yes. We offer a dedicated Software Developer Kit (SDK). See section "2-Link Software Developer Kit (SDK)" on page 21 for further information.

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 2-Link as well as the supply voltage, which 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-2 probes?

A: Some of the more expensive MRK-2 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.



# Chapter 9

## Glossary

---

This chapter explains important terms used throughout this manual.

**Application Program 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 10

## Literature and references

---

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

Reference	Title	Comments

# Index

---

## Numerics

2-Link	
Features .....	6
PINS .....	34
Software Developer Kit .....	21
Specifications .....	7
2Link.exe commands .....	24

## A

Application Program Interface .....	12
-------------------------------------	----

## D

Dynamic Linked Library (DLL) .....	12
------------------------------------	----

## H

Host .....	12
------------	----

## I

Image .....	12
-------------	----

## M

Memory coherency .....	12
------------------------	----

## O

Open collector .....	12
----------------------	----

## P

Processor Core .....	12
Program Status Register (PSR) .....	12

## R

Remapping .....	12
-----------------	----

## S

Support .....	5, 11
---------------	-------

## T

Target .....	12
Transistor-transistor logic (TTL) .....	12

TRAP ..... 12

**W**

Watchpoint ..... 12