Chipsmall
深圳市津利帝科技有限公司
Geelead Technology Limited
采购部

Programmers and Debuggers

SAM-ICE

USER GUIDE

Introduction

SAM-ICE is a JTAG emulator designed for Atmel® AT91 ARM® cores. It connects via USB to a PC running Microsoft® Windows® 2000 or higher. SAM-ICE™ has a built-in 20-pin JTAG connector, which is compatible with the standard 20-pin connector defined by ARM.

Table of Contents

1. Overview

1.1. Features of SAM-ICE

- Any Atmel AT91 core supported, including ARM® Thumb® mode
- Seamless integration into IAR Embedded Workbench®
- No power supply required, powered through USB
- Maximum JTAG speed 12MHz
- Auto speed recognition
- All JTAG signals can be monitored
- Support for multiple devices
- Fully plug and play compatible
- Standard 20-pin JTAG connector
- Wide target voltage range: 1.2V to 3.3V
- USB and 20-pin ribbon cable included
- Memory viewer (J-Mem) included
- SAM-ICE TCP/IP server included, can use SAM-ICE via TCP/IP networks
- RDI server available, can use SAM-ICE with RDI compliant software
- Support for adaptive clocking

1.2. Specification

Table 1-1. SAM-ICE Specification

Parameter	Value
Power supply	USB powered <50mA
USB Interface	USB 2.0, full speed
Target interface	JTAG 20-pin
Serial transfer rate between SAM-ICE and target	Up to 12MHz
Supported Target Voltage	1.2V to 3.3V
Operating Temperature	+5°C to +60°C
Storage Temperature	-20°C to +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 and newer

1.3. Requirements

1.3.1. Host System

In order to use SAM-ICE, a host system running Windows 2000 or newer with the SAM-ICE custom USB driver is required.

1.3.2. Target System

An Atmel AT91 target system is required.

2. Hardware

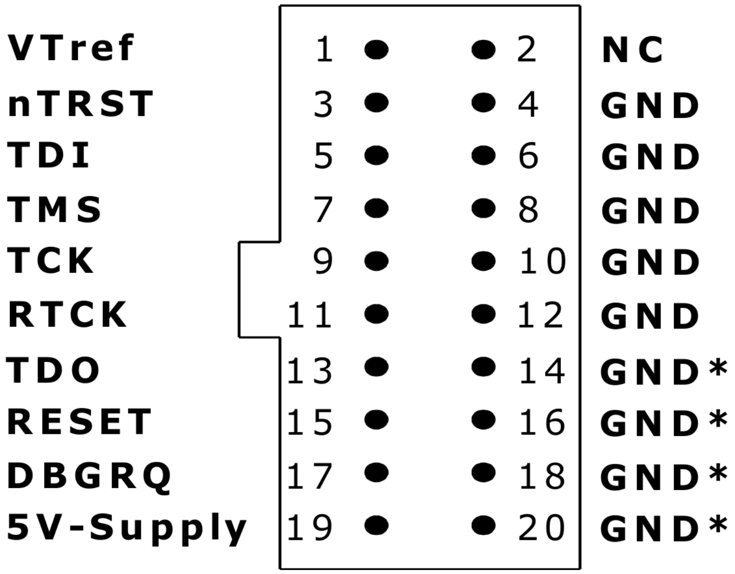
2.1. JTAG Interface

By default, only one device is assumed to be in the JTAG scan chain. If you have multiple devices in the scan chain, you must properly configure it. To do so, you have to specify the exact position of the CPU that should be addressed. Configuration of the scan is done by the target application. A target application can be a debugger such as Atmel Studio, the IAR™ C-SPY® debugger, ARM's AXD using RDI, a flash programming application such as SEGGER's J-Flash, or any other application using J-Link/J-Trace. It is the application's responsibility to supply a way to configure the scan chain. Most applications offer a dialog box for this purpose.

2.1.1. JTAG Pinout

SAM-ICE has a JTAG connector compatible to ARM's Multi-ICE. The JTAG connector is a 20-way Insulation Displacement Connector (IDC) keyed box header (2.54mm male) that mates with IDC sockets mounted on a ribbon cable.

Figure 2-1. JTAG Pinout



The table below lists the SAM-ICE JTAG pinouts.

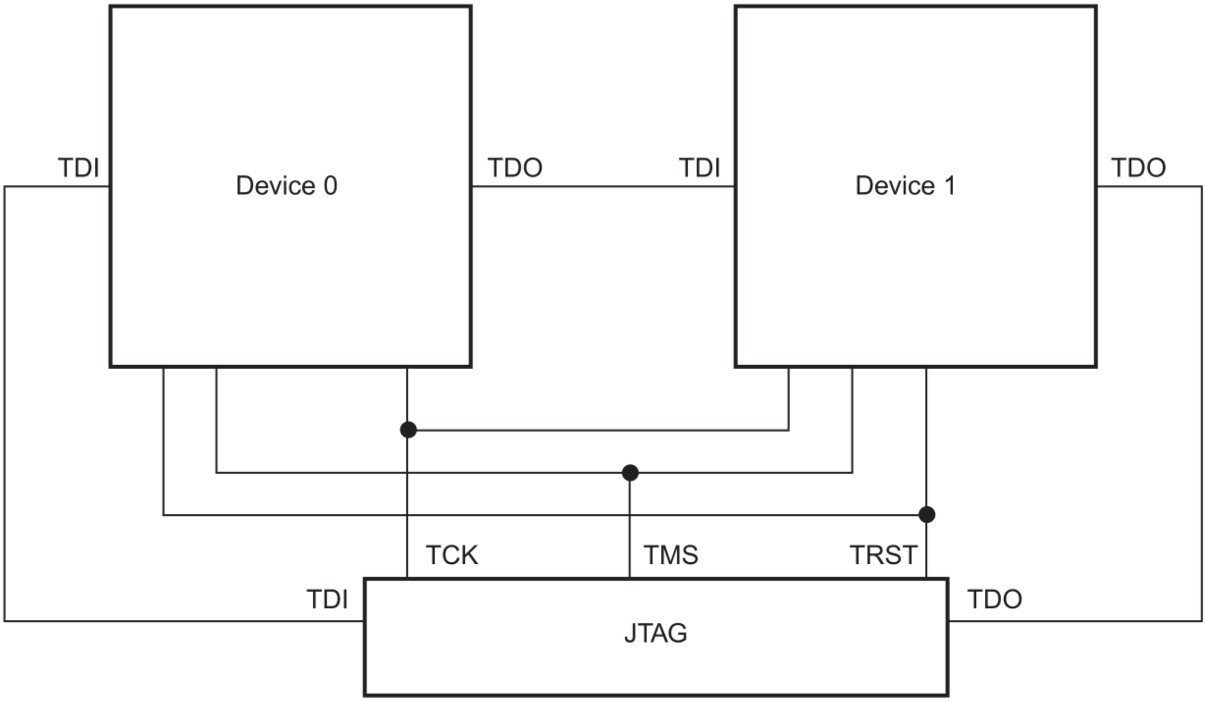
Table 2-1. JTAG Pinout

Pin	Signal	Type	Description
1	VTref	Input	This is the target reference voltage. It is used to check if the target has power, to create the logic-level reference for the input comparators and controls the output logic levels to the target. It is normally fed from V_{DD} on the target board and must not have a series resistor.
2	Vsupply	NC	This pin is not connected in SAM-ICE. It is reserved for compatibility with other equipment. Connect to V_{DD} or leave open in target system.

Pin	Signal	Type	Description
3	nTRST	Output	JTAG Reset. Output from SAM-ICE to the Reset signal on the target JTAG port. Typically connected to nTRST on the target CPU. This pin is normally pulled HIGH on the target to avoid unintentional resets when there is no connection.
4	GND	-	Common ground
5	TDI	Output	JTAG data input of the target CPU. It is recommended that this pin is pulled to a defined state on the target board. Typically connected to TDI on target CPU.
6	GND	-	Common ground
7	TMS	Output	JTAG mode set input of target CPU. This pin should be pulled up on the target. Typically connected to TMS on target CPU.
8	GND	-	Common ground
9	TCK	Output	JTAG clock signal to target CPU. It is recommended that this pin is pulled to a defined state on the target board. Typically connected to TCK on target CPU.
10	GND	-	Common ground
11	RTCK	NC	Input Return test clock signal from the target. Some targets must synchronize the JTAG inputs to internal clocks. To assist in meeting this requirement, a returned and re-timed TCK can be used to dynamically control the TCK rate. SAM-ICE supports adaptive clocking, which waits for TCK changes to be echoed correctly before making further changes. Connect to RTCK if available, otherwise to GND.
12	GND	-	Common ground
13	TDO	Input	JTAG data output from target CPU. Typically connected to TDO on target CPU.
14	GND	-	Common ground
15	RESET	I/O	Target CPU reset signal
16	GND	-	Common ground
17	-	NC	This pin is not connected in SAM-ICE
18	GND	-	Common ground
19	-	NC	This pin is not connected in SAM-ICE
20	GND	-	Common ground

2.1.2. Multiple Devices in the Scan Chain

SAM-ICE can handle multiple devices in the scan chain. This applies to hardware where multiple chips are connected to the same JTAG connector. As can be seen in the following figure, the TCK and TMS lines of all JTAG device are connected, while the TDI and TDO lines form a bus.



Currently, up to eight devices in the scan chain are supported. One or more of these devices can be CPU cores; the other devices can be of any other type, but need to comply with the JTAG standard.

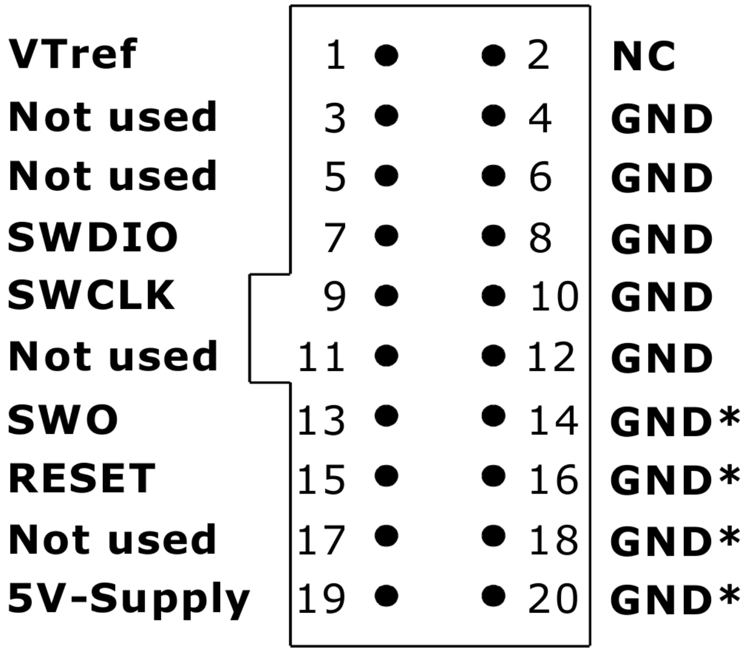
2.2. SWD Interface

The SAM-ICE support ARM's Serial Wire Debug (SWD). SWD replaces the 5-pin JTAG port with a clock (SWDCLK) and a single bi-directional data pin (SWDIO), providing all the normal JTAG debug and test functionality. SWDIO and SWCLK are overlaid on the TMS and TCK pins. In order to communicate with a SWD device, J-Link sends out data on SWDIO, synchronous to the SWCLK. With every rising edge of SWCLK, one bit of data is transmitted or received on the SWDIO.

2.2.1. SWD Pinout

The SAM-ICE JTAG connector is also compatible to ARM’s Serial Wire Debug (SWD).

Figure 2-2. SWD Pinout



The table below lists the SAM-ICE SWD pinout.

Table 2-2. SWD Pinout

Pin	Signal	Type	Description
1	VTref	Input	This is the target reference voltage. It is used to check if the target has power, to create the logic-level reference for the input comparators and controls the output logic levels to the target. It is normally fed from V_{DD} on the target board and must not have a series resistor.
2		NC	This pin is not connected
3		NC	This pin is not used. If the device may also be accessed via JTAG, this pin may be connected to nTRST, otherwise leave open.
4	GND		Common ground
5		NC	This pin is not used. If the device may also be accessed via JTAG, this pin may be connected to TDI, otherwise leave open.
6	GND		Common ground
7	SWDIO	Input/ Output	Single bi-directional data pin. A pull-up resistor is required. ARM recommends 100kΩ.
8	GND		Common ground
9	SWCLK	Output	Clock signal to target CPU. It is recommended that this pin is pulled to a defined state on the target board. Typically connected to TCK of target CPU.

Pin	Signal	Type	Description
10	GND		Common ground
11		NC	This pin is not used by the SAM-ICE when operating in SWD mode. If the device may also be accessed via JTAG, this pin may be connected to RTCK, otherwise leave open.
12	GND		Common ground
13	SWO	Input	Serial Wire Output trace port. (Optional, not required for SWD communication.)
14	GND		Common ground
15	RESET	Input/ Output	Target CPU reset signal. Typically connected to the RESET pin of the target CPU, which is typically called "nRST", "nRESET", or "RESET".
16	GND		Common ground
17		NC	This pin is not connected in SAM-ICE
18	GND		Common ground
19		NC	This pin is not connected in SAM-ICE
20	GND		Common ground

2.2.2. SWO

Serial Wire Output (SWO) support means support for a single pin output signal from the core. The Instrumentation Trace Macrocell (ITM) and Serial Wire Output (SWO) can be used to form a Serial Wire Viewer (SWV). The Serial Wire Viewer provides a low-cost method of obtaining information from inside the MCU.

Usually it should not be necessary to configure the SWO speed because this is usually done by the debugger.

3. Multi-core Debugging

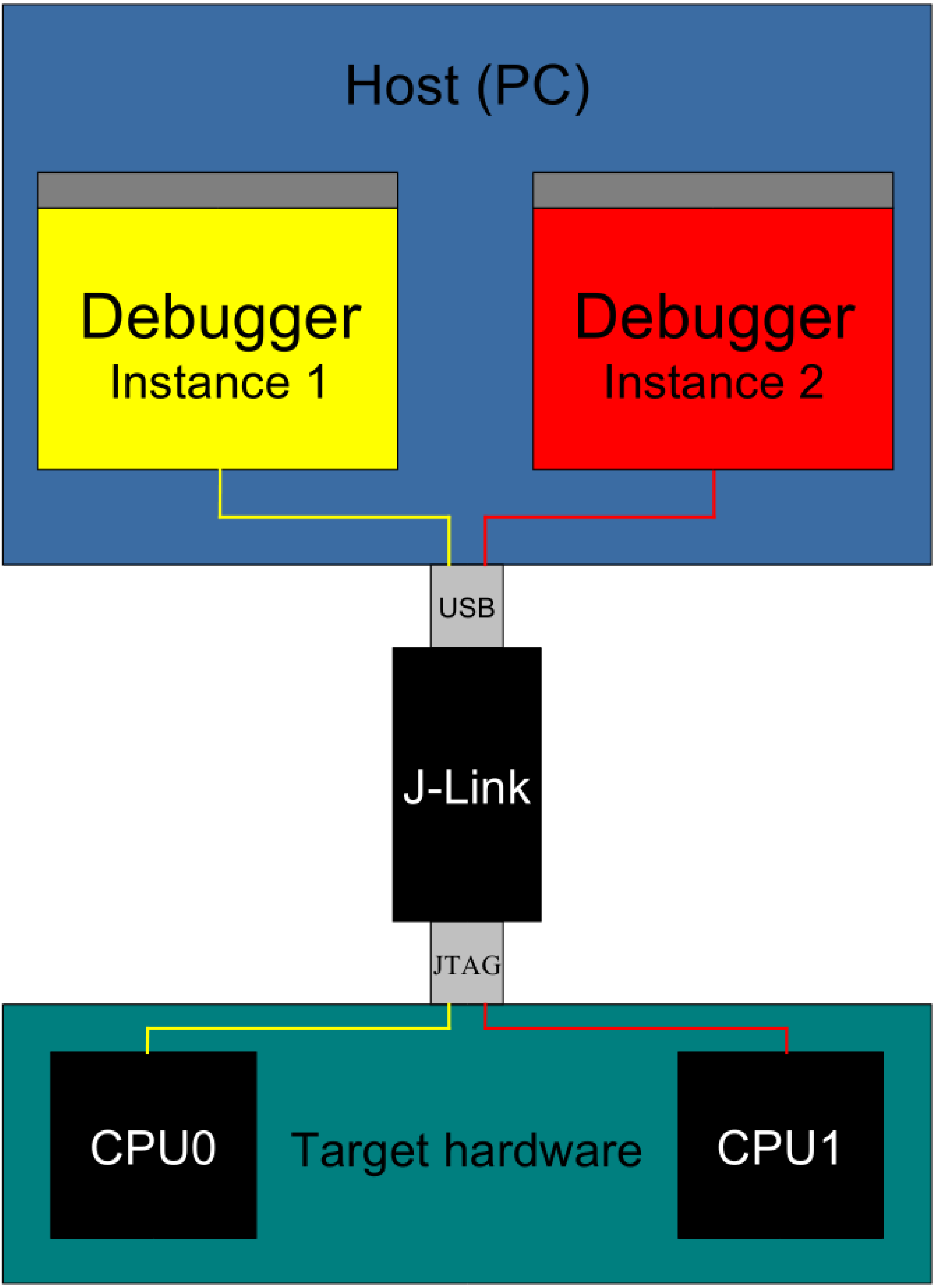
SAM-ICE is able to debug multiple cores on one target system connected to the same scan chain.

3.1. How Multi-core Debugging Works

Multi-core debugging requires multiple debuggers or multiple instances of the same debugger. Two or more debuggers can use the same SAM-ICE simultaneously. Configuring a debugger to work with a core in a multi-core environment does not require special settings. All that is required is proper setup of the scan chain for each debugger. This enables SAM-ICE to debug more than one core on a target at the same time.

The following figure shows a host, debugging two CPU cores with two instances of the same debugger.

Figure 3-1. Multi-core Debugging Setup. Figure from Segger J-Link/J-Trace User Guide UM08001.

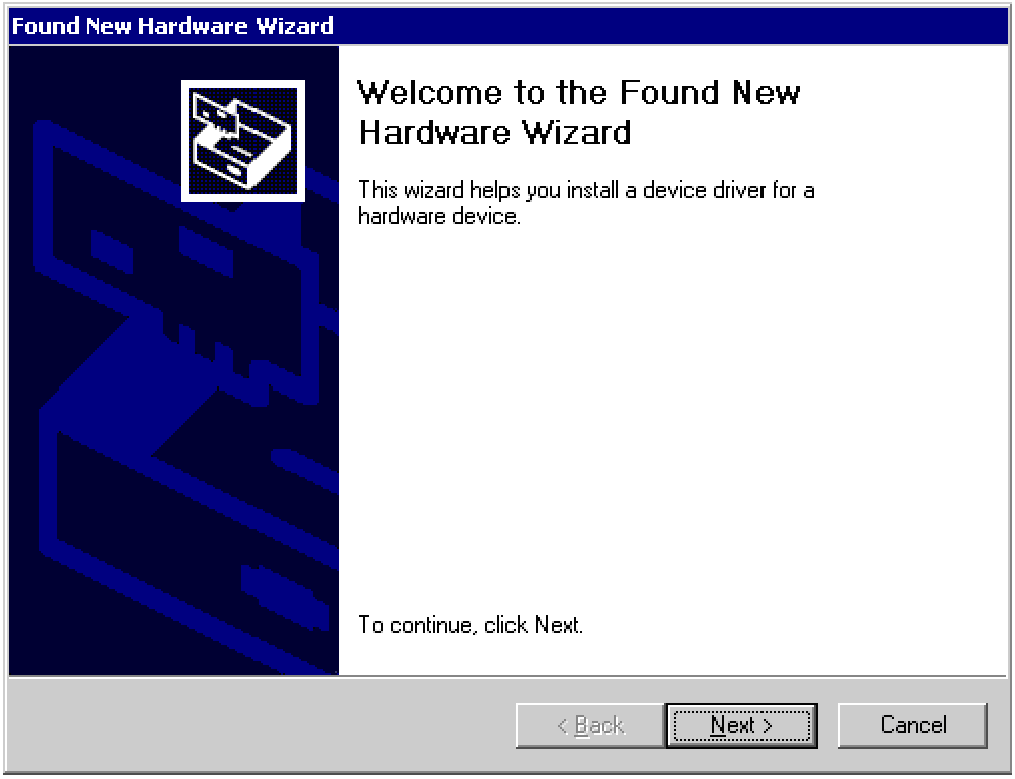


Both debuggers share the same physical connection. The core to debug is selected through the JTAG settings as described below.

4. Setup

4.1. Installing the USB Driver

When your SAM-ICE is plugged into the computer's USB port, or when the computer is first powered on after connecting SAM-ICE, Windows® detects the new hardware.



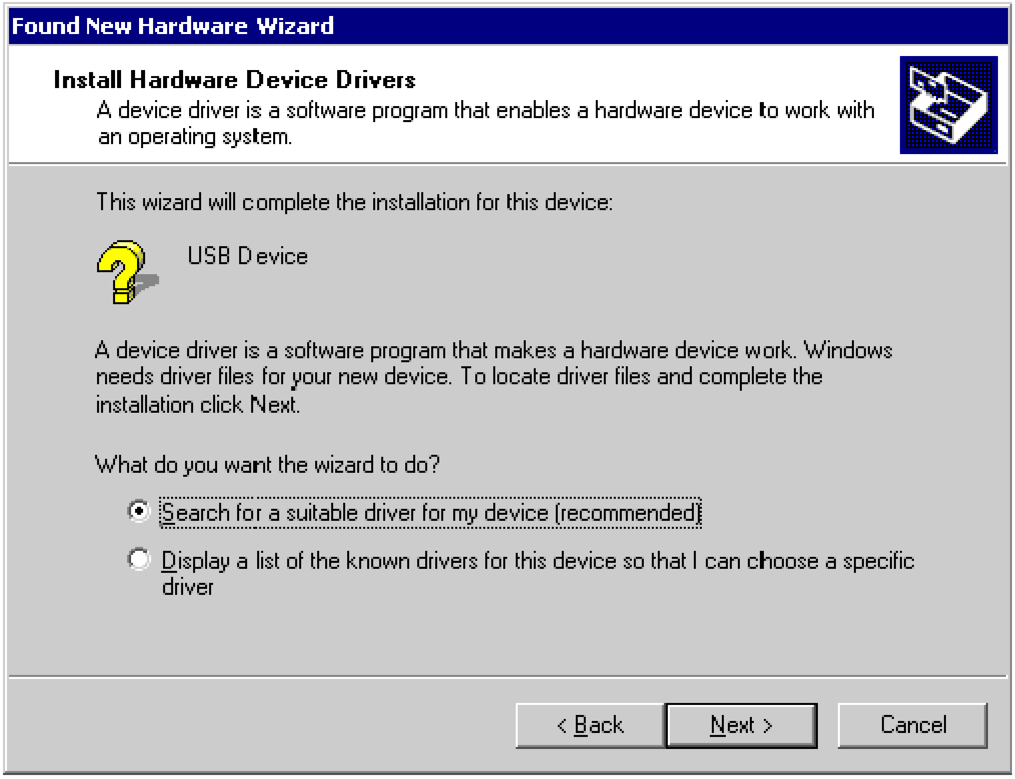
Found New Hardware Wizard

Welcome to the Found New Hardware Wizard

This wizard helps you install a device driver for a hardware device.

To continue, click Next.

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.



Found New Hardware Wizard

Install Hardware Device Drivers

A device driver is a software program that enables a hardware device to work with an operating system.

This wizard will complete the installation for this device:

USB Device

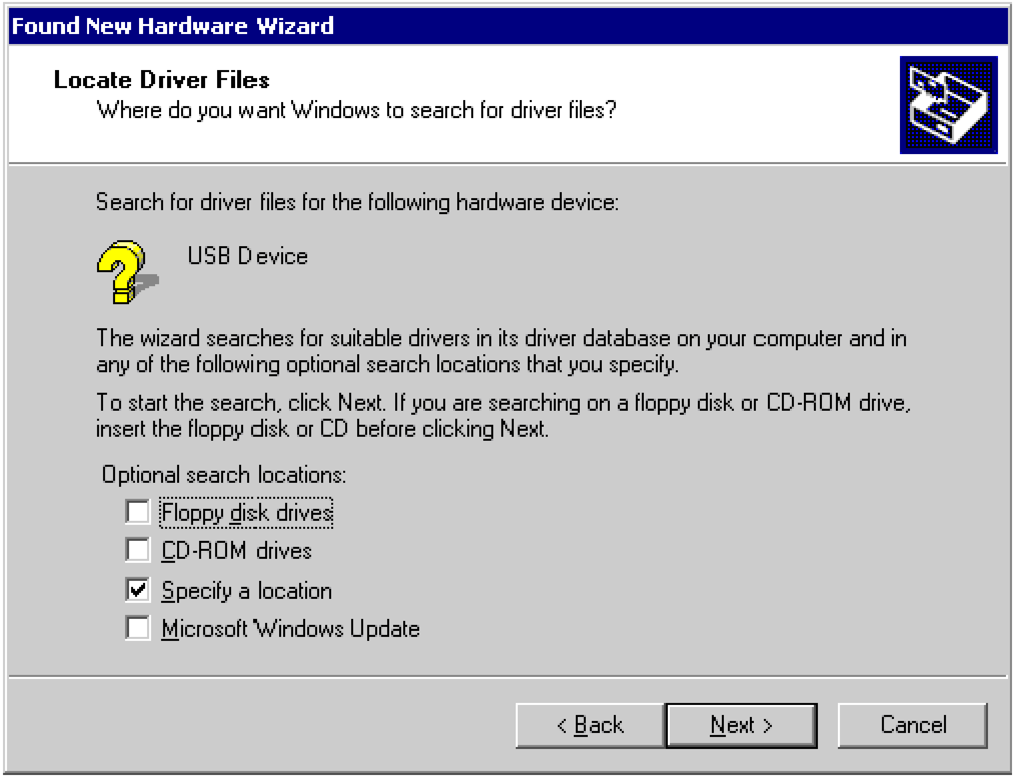
A device driver is a software program that makes a hardware device work. Windows needs driver files for your new device. To locate driver files and complete the installation click Next.

What do you want the wizard to do?

● Search for a suitable driver for my device (recommended)

○ Display a list of the known drivers for this device so that I can choose a specific driver

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



Found New Hardware Wizard

Locate Driver Files

Where do you want Windows to search for driver files?

Search for driver files for the following hardware device:

USB Device

The wizard searches for suitable drivers in its driver database on your computer and in any of the following optional search locations that you specify.

To start the search, click Next. If you are searching on a floppy disk or CD-ROM drive, insert the floppy disk or CD before clicking Next.

Optional search locations:

□ Floppy disk drives

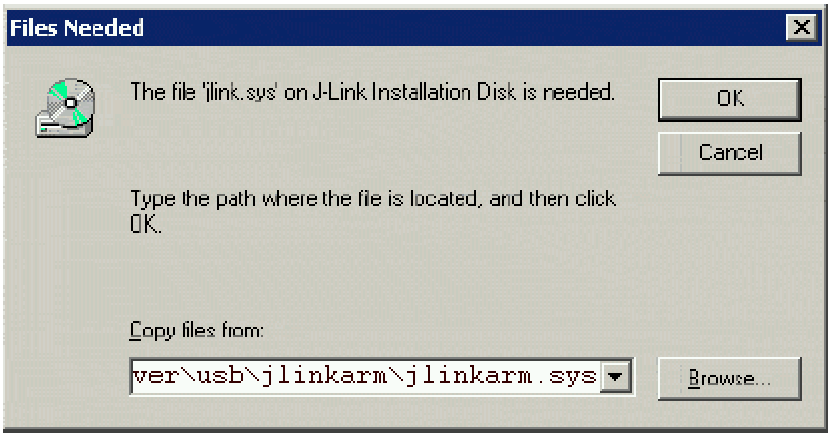
□ CD-ROM drives

☑ Specify a location

□ Microsoft Windows Update

Copy the driver available on the DVD-ROM to your location.

The wizard asks you to specify the location of the correct driver files for the new device. Use the directory navigator to select `D:\tools\driver\usb\jlinkarm` (or your chosen location) and confirm with a click on the "Next >" button.



Files Needed

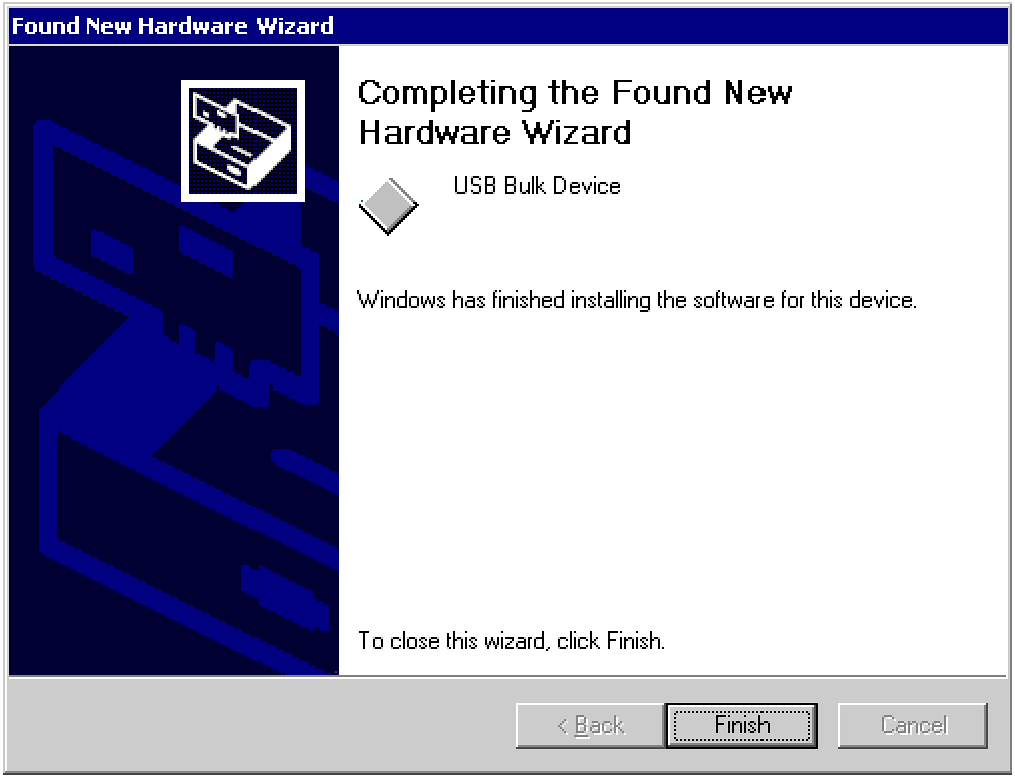
The file 'jlink.sys' on J-Link Installation Disk is needed.

Type the path where the file is located, and then click OK.

Copy files from:

ver\usb\jlinkarm\jlinkarm.sys

The wizard confirms your choice and starts to copy, when you click on the "OK" button.



Found New Hardware Wizard

Completing the Found New Hardware Wizard

USB Bulk Device

Windows has finished installing the software for this device.

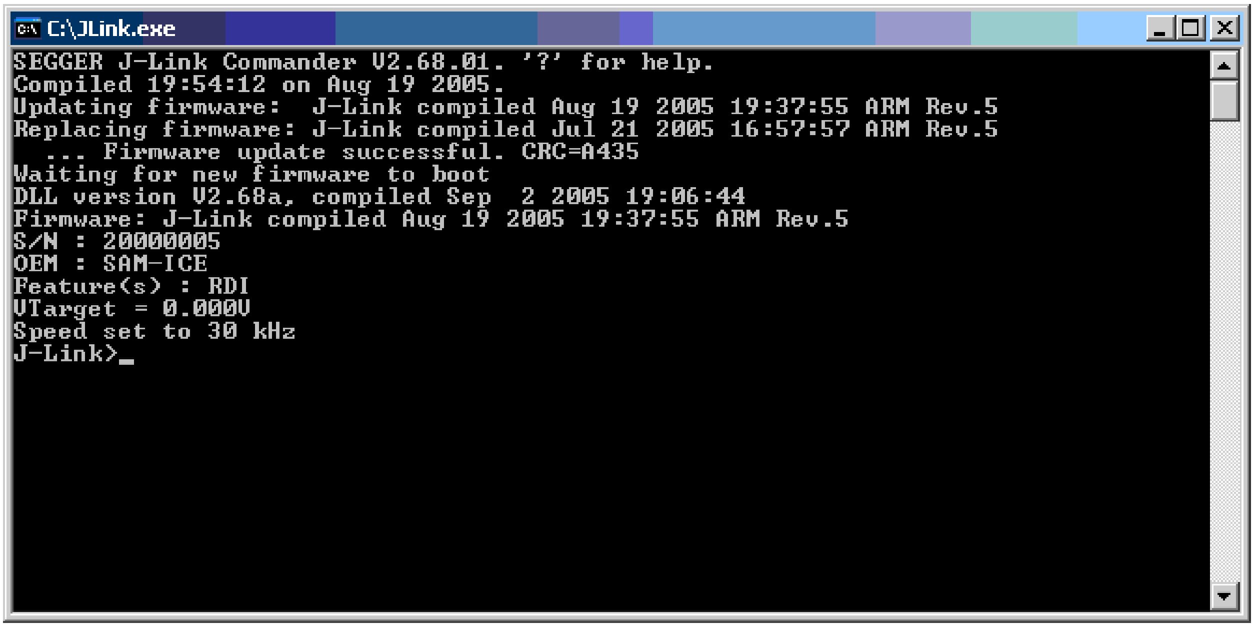
To close this wizard, click Finish.

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

4.1.1. Verifying Correct Driver Installation

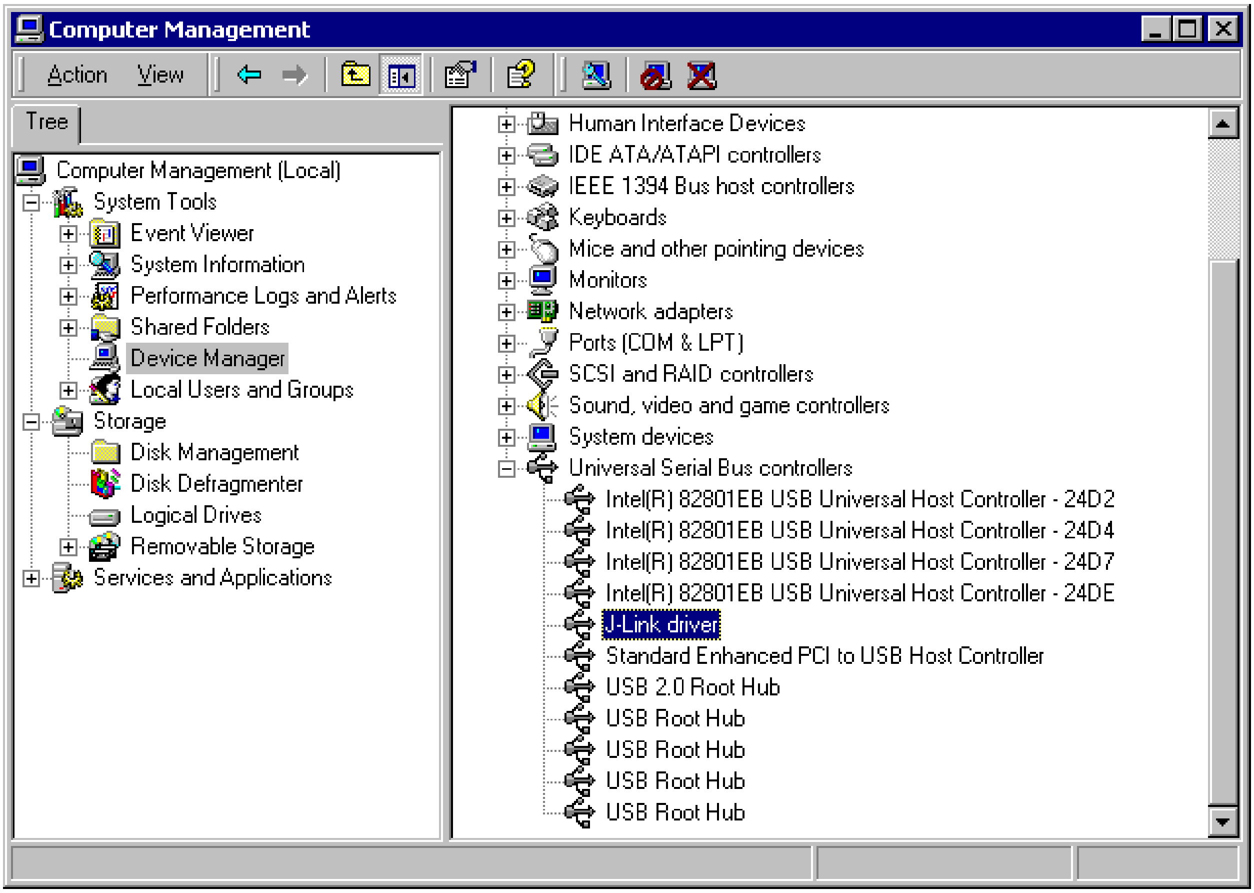
To verify the correct installation of the driver, disconnect and reconnect SAM-ICE to the USB port. During the enumeration process, which takes about 2 seconds, the LED on SAM-ICE is flashing. After successful enumeration, the LED stays on permanently.

Start the provided sample application JLink.exe. JLink.exe should display the build date of the SAM-ICE firmware, the serial number, a target voltage of 0.000V if a target is not connected to SAM-ICE, and the speed selection. See the screen-shot below.



```
SEGGER J-Link Commander V2.68.01. '?' for help.
Compiled 19:54:12 on Aug 19 2005.
Updating firmware:  J-Link compiled Aug 19 2005 19:37:55 ARM Rev.5
Replacing firmware: J-Link compiled Jul 21 2005 16:57:57 ARM Rev.5
  ... Firmware update successful. CRC=A435
Waiting for new firmware to boot
DLL version V2.68a, compiled Sep  2 2005 19:06:44
Firmware: J-Link compiled Aug 19 2005 19:37:55 ARM Rev.5
S/N : 20000005
OEM : SAM-ICE
Feature(s) : RDI
VTarget = 0.000V
Speed set to 30 kHz
J-Link>_
```

In addition to this you may verify the driver installation by consulting the Windows device manager. If the driver is installed and your SAM-ICE is connected to your computer, the device manager should list the J-Link driver as a node below "Universal Serial Bus controllers", as shown in the following screen-shot:



Computer Management (Local)
- System Tools
  - Event Viewer
  - System Information
  - Performance Logs and Alerts
  - Shared Folders
  - Device Manager
  - Local Users and Groups
- Storage
  - Disk Management
  - Disk Defragmenter
  - Logical Drives
  - Removable Storage
- Services and Applications

- Human Interface Devices
- IDE ATA/ATAPI controllers
- IEEE 1394 Bus host controllers
- Keyboards
- Mice and other pointing devices
- Monitors
- Network adapters
- Ports (COM & LPT)
- SCSI and RAID controllers
- Sound, video and game controllers
- System devices
- Universal Serial Bus controllers
  - Intel(R) 82801EB USB Universal Host Controller - 24D2
  - Intel(R) 82801EB USB Universal Host Controller - 24D4
  - Intel(R) 82801EB USB Universal Host Controller - 24D7
  - Intel(R) 82801EB USB Universal Host Controller - 24DE
  - J-Link driver
  - Standard Enhanced PCI to USB Host Controller
  - USB 2.0 Root Hub
  - USB Root Hub
  - USB Root Hub
  - USB Root Hub
  - USB Root Hub

A right-click on the driver opens a context menu, which contains the item "Properties". If you select this item, a new dialog is opened and should report: "This device is working properly".

4.2. Debug Installation

The SAM-ICE RDI software is an RDI interface for SAM-ICE. It makes it possible to use SAM-ICE with any RDI compliant debugger.

The main part of the software is an RDI compliant DLL, which needs to be selected in the debugger.

Supported configurations are described on the AT91 DVD-ROM. For additional information on debug, refer to the SAM-ICE "Getting started" bar menu.

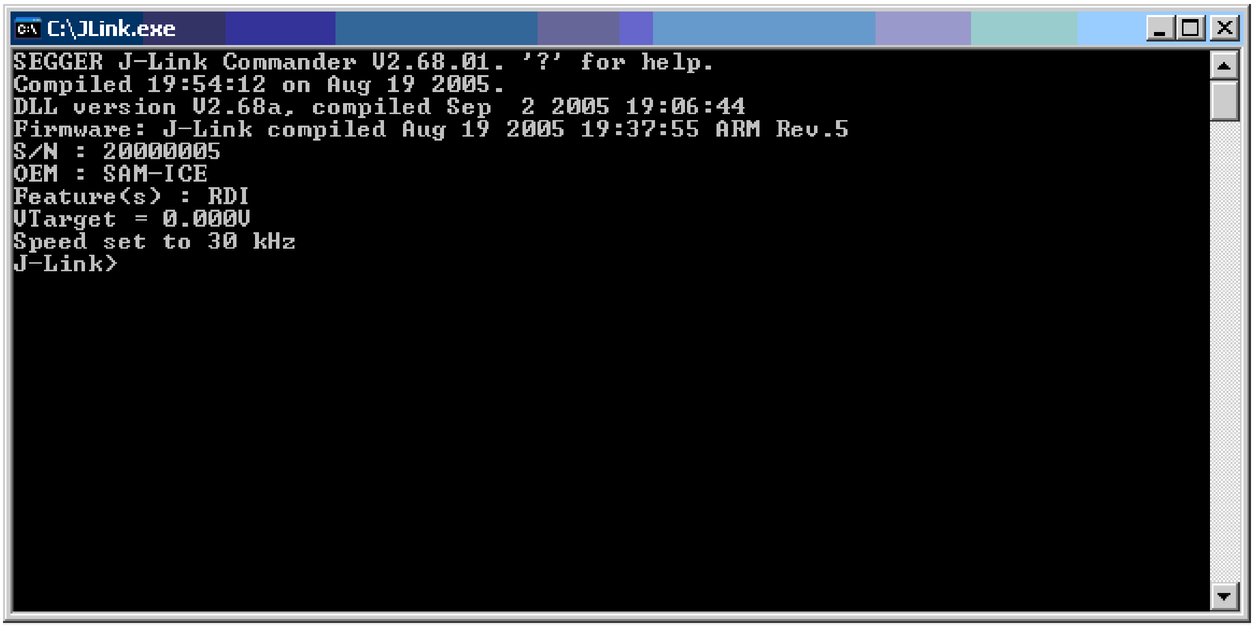
4.3. Connecting the Target System

4.3.1. Power-on Sequence

SAM-ICE must be powered on before connecting it to the target device. First, connect SAM-ICE to the host system via the USB and then connect SAM-ICE to the target device via JTAG. Power-on the device after you have connected SAM-ICE to it.

4.3.2. Verifying Target Device Connection

If the USB driver is working properly and your SAM-ICE is connected to the host system, you may connect SAM-ICE to your target hardware. Then start JLink.exe again; it should now display the same SAM-ICE related information as above. In addition, it should report that it found a JTAG target and the target core ID. The screen-shot below shows the output of `JLink.exe`. As can be seen, it reports a SAMICE with one JTAG device connected.



```
SEGGER J-Link Commander V2.68.01. '?' for help.
Compiled 19:54:12 on Aug 19 2005.
DLL version V2.68a, compiled Sep  2 2005 19:06:44
Firmware: J-Link compiled Aug 19 2005 19:37:55 ARM Rev.5
S/N : 20000005
OEM : SAM-ICE
Feature(s) : RDI
VTarget = 0.000V
Speed set to 30 kHz
J-Link>
```

4.3.3. Problems

For help with any of the steps described above, refer to Troubleshooting for troubleshooting tips.

5. SAM-ICE Related Software

5.1. Overview

Table 5-1. Available Software Packages

Software	Description
JLink.exe	Free Command line tool with basic functionality for target analysis.
SAM-ICE TCP/IP Server	Free utility which provides the possibility to use SAM-ICE remotely via TCP/IP.
J-Mem memory viewer	Free target memory viewer. Shows the memory content of a running target and allows editing as well.
SAM-ICE ARM Flash DLL	An enhanced version of the JLinkARM.DLL, which contains additional API functions for Flash programming.
RDI support	Provides Remote Debug Interface (RDI) support.

5.2. Free Software

Free software related to SAM-ICE ships with SAM-ICE and may also be downloaded from the web site: http://www.segger.com.

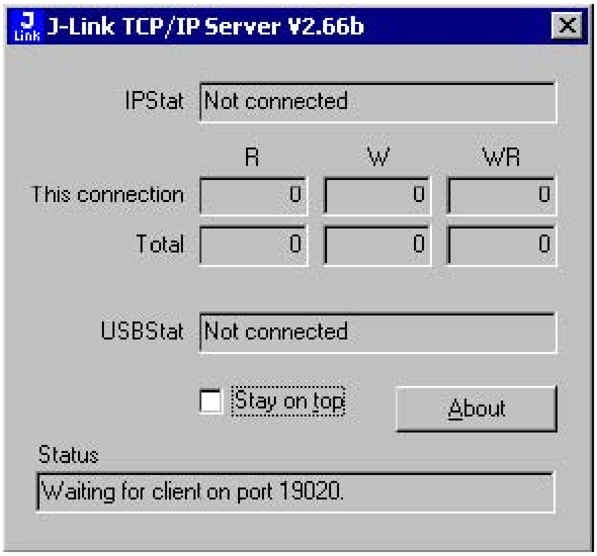
No additional license is required to use this software.

5.2.1. JLink.exe (Command Line Tool)

JLink.exe is a tool, that can be used to verify proper installation of the USB driver and to verify the connection to the ARM chip, as well as for simple analysis of the target system. It permits some simple commands, such as memory dump, halt, step, go, and ID-check, as well as some more in-depths analysis of the state of the ARM core and the ICE breaker module.

5.2.2. SAM-ICE TCP/IP Server (Remote SAM-ICE Use)

The SAM-ICE TCP/IP server allows using SAM-ICE remotely via TCP/IP. This enables you to connect to and fully use a SAM-ICE from another computer. Performance is just slightly (about 10%) lower than with direct USB connection.



IPStat: Not connected

	R	W	WR
This connection	0	0	0
Total	0	0	0

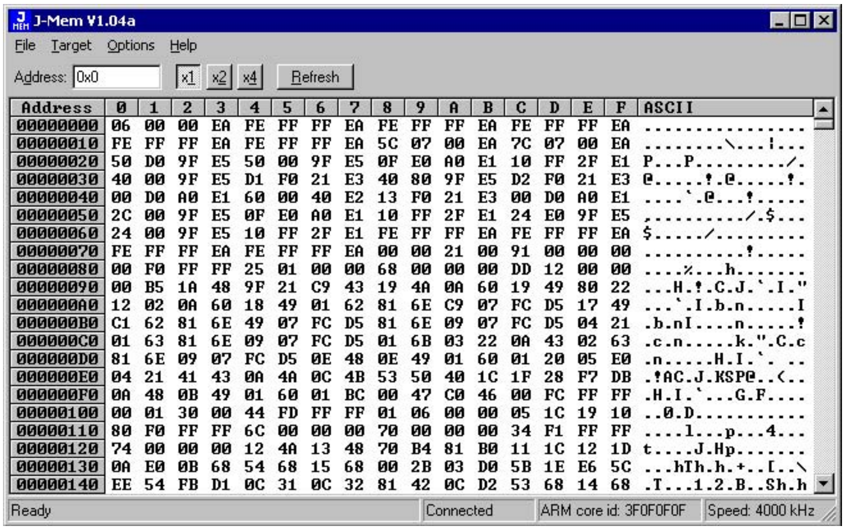
USBStat: Not connected

□ Stay on top

Status: Waiting for client on port 19020.

5.2.3. J-Mem Memory Viewer

J-Mem displays memory contents of ARM systems and allows modifications of RAM and SFRs (Special Function Registers) while the target is running. This makes it possible to look into the memory of an ARM chip at run-time; RAM can be modified and SFRs can be written. The type of access for both read and write access can be selected to be 8/16/32 bits. This is an efficient means for modifying SFRs, in particular because it writes the SFR only after the complete value has been entered.



Address: 0x0

```
Address   0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F  ASCII
00000000 06 00 00 EA FE FF FF EA FE FF FF EA FE FF FF EA ................
00000010 FE FF FF EA FE FF FF EA 5C 07 00 EA 7C 07 00 EA ........\...|...
00000020 50 D0 9F E5 50 00 9F E5 0F E0 A0 E1 10 FF 2F E1 P...P........./.
00000030 40 00 9F E5 D1 F0 21 E3 40 80 9F E5 D2 F0 21 E3 @.....!.@.....!.
00000040 00 D0 A0 E1 60 00 40 E2 13 F0 21 E3 00 D0 A0 E1 ....`.@...!.....
00000050 2C 00 9F E5 0F E0 A0 E1 10 FF 2F E1 24 E0 9F E5 ,........./.$...
00000060 24 00 9F E5 10 FF 2F E1 FE FF FF EA FE FF FF EA $...../.........
00000070 FE FF FF EA FE FF FF EA 00 00 21 00 91 00 00 00 ..........!.....
00000080 00 F0 FF FF 25 01 00 00 68 00 00 00 DD 12 00 00 ....%...h.......
00000090 00 B5 1A 48 9F 21 C9 43 19 4A 0A 60 19 49 80 22 ...H.!.C.J.`.I."
000000A0 12 02 0A 60 18 49 01 62 81 6E C9 07 FC D5 17 49 ...`.I.b.n.....I
000000B0 C1 62 81 6E 49 07 FC D5 81 6E 09 07 FC D5 04 21 .b.nI....n.....!
000000C0 01 63 81 6E 09 07 FC D5 01 6B 03 22 0A 43 02 63 .c.n.....k.".C.c
000000D0 81 6E 09 07 FC D5 0E 48 0E 49 01 60 01 20 05 E0 .n.....H.I.`. ..
000000E0 04 21 41 43 0A 4A 0C 4B 53 50 40 1C 1F 28 F7 DB .!AC.J.KSP@..(..
000000F0 0A 48 0B 49 01 60 01 BC 00 47 C0 46 00 FC FF FF .H.I.`...G.F....
00000100 00 01 30 00 44 FD FF FF 01 06 00 00 05 1C 19 10 ..0.D...........
00000110 80 F0 FF FF 6C 00 00 00 70 00 00 00 34 F1 FF FF ....l...p...4...
00000120 74 00 00 00 12 4A 13 48 70 B4 81 B0 11 1C 12 1D t....J.Hp.......
00000130 0A E0 0B 68 54 68 15 68 00 2B 03 D0 5B 1E E6 5C ...hTh.h.+..[..\
00000140 EE 54 FB D1 0C 31 0C 32 81 42 0C D2 53 68 14 68 .T...1.2.B..Sh.h
```

Ready | Connected | ARM core id: 3F0F0F0F | Speed: 4000 kHz

6. Frequently Asked Questions

Which CPUs are supported?

SAM-ICE supports all ARM based Atmel microcontrollers.

What is the maximum JTAG speed supported by SAM-ICE?

The maximum supported JTAG speed is 8MHz.

Can I access individual ICE registers via SAM-ICE?

Yes, you can access all individual ICE registers via SAM-ICE.

Can SAM-ICE read back the status of the JTAG pins?

Yes, the status of all pins can be read. This includes the outputs of SAM-ICE as well as the supply voltage and can be useful to detect hardware problems on the target system.

Does SAM-ICE support the embedded trace macro (ETM)?

No. ETM requires another connection to the ARM chip and a CPU with built-in ETM.

7. Support

7.1. Troubleshooting

7.1.1. General Procedure

If you experience problems with a SAM-ICE, follow the steps below to solve these problems:

1. Close all running applications on your host system.
2. Disconnect the SAM-ICE device from USB.
3. Power-off target.
4. Re-connect SAM-ICE to host system (attach USB cable).
5. Power-on target.
6. Try your target application again. If the problem disappears, you are done; otherwise, continue.
7. Close all applications running on your host system again.
8. Disconnect the SAM-ICE device from USB.
9. Power-off target.
10. Re-connect SAM-ICE to host system (attach USB cable).
11. Power-on target.
12. Start `JLink.exe`.
13. If `JLink.exe` reports the SAM-ICE serial number and the target processor's core ID, the SAM-ICE is working properly and cannot be the cause of your problem.
14. If `JLink.exe` is unable to read the target processor's core ID, you should analyze the communication between your target and SAM-ICE with a logic analyzer or oscilloscope.
15. If your problem persists and you own an original SAM-ICE (not an OEM version), see section Typical Problem Scenarios.

7.1.2. Typical Problem Scenarios

7.1.2.1. SAM-ICE LED is OFF

Meaning: The USB connection does not work.

Corrective action: Check the USB connection. Try to re-initialize SAM-ICE by disconnecting and reconnecting it. Make sure that the connectors are firmly attached. Check the cable connections on your SAM-ICE 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 PC.

7.1.2.2. SAM-ICE LED is Flashing at a High Frequency

Meaning: SAM-ICE cannot be enumerated by the USB controller.

Most likely causes:

- Another program is already using SAM-ICE
- The SAM-ICE USB driver does not work correctly

Corrective action:

- Close all running applications and try to reinitialize SAM-ICE by disconnecting and reconnecting it
- If the LED blinks permanently, check the correct installation of the SAM-ICE USB driver. Deinstall and reinstall the driver as shown in chapter Setup.

7.1.2.3. SAM-ICE does not get any Connection to the Target

Most likely causes:	- The JTAG cable is defective - The target hardware is defective
Corrective action:	Follow the steps described in section General Procedure.

7.2. Contacting Support

Before contacting support, assure that you have tried to solve your problem by following the steps outlined in section General Procedure. You may also try your SAM-ICE 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 SAM-ICE.

Make sure that you have the following information available for the support team:

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

8. Glossary

8.1. Terminology

Adaptive clocking A technique in which a clock signal is sent out by Multi-ICE and waits for the returned clock before generating the next clock pulse. The technique allows the Multi-ICE interface unit to adapt to differing signal drive capabilities and differing cable lengths.

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.

Big-endian Memory organization where the least significant byte of a word is at a higher address than the most significant byte. See Little-endian.

Cache cleaning The process of writing dirty data in a cache to main memory.

Coprocessor An additional processor that is used for certain operations, for example, for floating-point math calculations, signal processing, or memory management.

Dirty data When referring to a processor data cache, data that has been written to the cache but has not been written to main memory. Only write-back caches can have dirty data, because a write-through cache writes data to the cache and to main memory simultaneously. The process of writing dirty data to main memory is called cache cleaning.

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.

EmbeddedICE™ The additional hardware provided by debug-able ARM processors to aid debugging.

Halfword A 16-bit unit of information. Contents are taken as being an unsigned integer, unless otherwise stated.

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

ICache Instruction cache.

ICE Extension Unit A hardware extension to the EmbeddedICE logic that provides more breakpoint units.

ID Identifier.

IEEE® 1149.1 The IEEE Standard, which defines TAP. Commonly (but incorrectly) referred to as JTAG.

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

In-Circuit Emulator (ICE)
A device enabling access to and modification of the signals of a circuit while that circuit is operating.

Instruction Register
When referring to a TAP controller, a register that controls the operation of the TAP.

IR
See Instruction Register.

Joint Test Action Group (JTAG)
The name of the standards group which created the IEEE 1149.1 specification.

Little-endian
Memory organization where the least significant byte of a word is at a lower address than the most significant byte. See also Big-endian.

Memory Management Unit (MMU)
Hardware that controls caches and access permissions to blocks of memory, and translates virtual to physical addresses.

Multi-ICE
Multi-processor EmbeddedICE interface.

nSRST
Abbreviation of System Reset. The electronic signal which causes the target system other than the TAP controller to be reset. This signal is known as nSYSRST in some other manuals. See also nTRST.

nTRST
Abbreviation of TAP Reset. The electronic signal that causes the target system TAP controller to be reset. This signal is known as nICERST in some other manuals. See also nSRST.

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

Is also referred to as Current PSR (CPSR), to emphasize the distinction between it and the Saved PSR (SPSR). The SPSR holds the value the PSR had when the current function was called, and which will be restored when control is returned.

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.

Remote Debug Interface (RDI)
RDI is an open ARM standard procedural interface between a debugger and the debug agent. The widest possible adoption of this standard is encouraged.

RTCK
Returned TCK. The signal which enables Adaptive Clocking.

RTOS
Real Time Operating System.