# J-Link 用户手册

版本: Rev3.0

www.mcuzone.com

2007-09

J-Link 用户手册 Rev 3.0 - 1 -

目录

一、J-Link ARM JTAG 仿真器简介

- 二、J-LINK 驱动安装
- 三、J-LINK 在各个主流开发环境下的设置
- 四、J-FLASH ARM 使用设置
- 五、F.A.Q
- 附录一、使用 jlink 的 GDBserver
- 附录二、J-Flash ARM 命令行使用说明
- 附录三、V6版本 J-Link 的特点和 SWD 使用说明

### 版本更新说明

### Rev3.0:

增加了附录三,主要介绍 V6版本的特点和 SWD 使用说明, 2007-09 更新

#### Rev2.1:

增加了命令行使用说明做为附录二, 2007-05 更新。

### Rev2.0:

添加了 GDBserver 的使用指南作为附录一, 2006 年 10 月更新。

### Rev1.6a:

修改了下载速度的错误,更新了器件支持,改动了一些地方的排版格式

### Rev1.6:

添加 IAR 下使用 XLINK 进行 FLASH 下载的注意事项

### Rev1.5:

添加 KEIL 下 Utilities 下的设置截图, 2006-08 升级

### Rev1.4:

修正部分文字错误,添加速度测试截图,添加 J-FLASH ARM 新特性说明, 2006-07 升级

### Rev1.3a:

增加 F.A.Q, 2006-06-07 升级

### Rev1.3:

增加 F.A.Q, 2006-06-04 升级

### Rev1.2:

增加 J-FLASH ARM 的操作说明, 2006-06-02 升级

### Rev1.0:

原始版本, 2006-06-01 完成

### 一、J-Link ARM JTAG 仿真器简介

J-Link 是 SEGGER 公司为支持仿真 ARM 内核芯片推出的 JTAG 仿 真器。配合 IAR EWARM, ADS, KEIL, WINARM, RealView 等集成开发 环境支持所有 ARM7/ARM9 内核芯片的仿真,通过 RDI 接口和各集成开 发环境无缝连接,操作方便、连接方便、简单易学,是学习开发 ARM 最好最实用的开发工具。

J-Link ARM 主要特点

\* IAR EWARM 集成开发环境无缝连接的 JTAG 仿真器

\* 支持所有 ARM7/ARM9 内核的芯片,以及 cortex M3,包括 Thumb 模式

\* 支持 ADS, IAR, KEIL, WINARM, REALVIEW 等几乎所有的开发环境

\* 下载速度高达 ARM7:600kB/s, ARM9:550kB/s, 通过 DCC 最高可 达 800 kB/s

\* 最高 JTAG 速度 12 MHz

\* 目标板电压范围 1.2V - 3.3V

\* 自动速度识别功能

\* 监测所有 JTAG 信号和目标板电压

\* 完全即插即用

\* 使用 USB 电源(但不对目标板供电)

\* 带 USB 连接线和 20 芯扁平电缆

J-Link 用户手册 Rev 3.0

- 4 -

- \* 支持多 JTAG 器件串行连接
- \* 标准 20 芯 JTAG 仿真插头
- \* 选配 14 芯 JTAG 仿真插头
- \* 选配用于 5V 目标板的适配器
- \*带 J-Link TCP/IP server, 允许通过 TCP/ IP 网络使用 J-Link

J-Link 支持 ARM 内核

- \* ARM7TDMI (Rev 1)
- \* ARM7TDMI (Rev 3)
- \* ARM7TDMI-S (Rev 4)
- \* ARM720T
- \* ARM920T
- \* ARM926EJ-S
- \* ARM946E-S
- \* ARM966E-S

速度信息:

	Memory	ARM7	ARM9
Revision	download	Memory	Memory
	via DCC	download	download
J-Link Rev.	185.0 kB/s	150.0 kB/s	75.0 kB/s
1-4	(4MHz JTAG)	(4MHz JTAG)	(4MHz JTAG)
J-Link Rev.	800.0 kB/s	600.0 kB/s	550.0 kB/s
5	(12MHz JTAG)	(12MHz JTAG)	(12MHz JTAG)

### 二、J-LINK 驱动安装

注意: J-LINK 有非常多的授权文件(License),请根据实际需要选择合适的授权版本! IAR 版本的 J-LINK 功能有限,但是价格较为便宜,推荐购买全功能版本的 J-LINK,可以获得更多性能!

首先到 <u>http://www.segger.com/download\_jlink.html</u>下载最新的 J-LINK 驱动软件, <u>J-Link ARM software and documentation pack</u>, 内含 USB driver, J-Mem, J-Link.exe and DLL for ARM, J-Flash and J-Link RDI。

注意: SEGGER 公司升级比较频繁,请密切留意 SEGGER 公司 网站,下载最新驱动,以支持更多器件!

安装驱动很简单,只要将下载的 ZIP 包解压,然后直接安装即可, 默认安装,一路点击"NEXT"即可:







J-Link 用户手册 Rev 3.0 - 8 -

😼 Start Installat	ion	×
	You are now ready to install J-Link ARM V3.20h. Press the Next button to begin the installation or the Back button to reenter the installation information.	
	< <u>B</u> ack ( <u>Next</u> > Cancel	

Installing	
	Current File Copying file: C:\\SEGGER\JLinkARM_V320h\JLinkTCPIPServer.exe All Files Time Remaining 7 minutes 50 seconds
	< <u>B</u> ack <u>N</u> ext > Cancel

J-Link 用户手册 Rev 3.0 - 9 -

🔏 Installation Co	mplete 🔀
	J-Link ARM V3.20h has been successfully installed.
	Press the Finish button to exit this installation.
	< <u>B</u> ack <b>Finish</b> Cancel



安装完成后, 请插入 JLINK 硬件, 然后系统提示发现新硬件, 一般情况下 会自动安装驱动, 如果没有自动安装, 请选择手动指定驱动程序位置(安装目录), 然后将驱动程序位置指向到 JLINK 驱动软件的安装目录下的 Driver 文件夹, 驱 动程序就在改文件夹下。

安装完成可以桌面出现两个快捷图标, J-Link ARM 可以用来进行 设置和测试,下面我们看一下 J-LINK 的测试数据(在7X256 EK 上 测试):

J-Link ARE V3.30g - 🗆 🗙 SEGGER J-Link Commander V3.30g ('?' for help) Compiled Jul 1 2006 12:31:51 DLL version V3.30g, compiled Jul 1 2006 12:31:29 Firmware: J-Link compiled Jun 30 2006 08:34:29 ARM Rev.5 Hardware: V5.30 S∕N : OEM : IAR Feature(s) : VTarget = 3.332VSpeed set to 30 kHz Found 1 JTAG device, Total IRLen = 4: Id of device #1: 0x3F0F0F0F Found ARM with core Id 0x3F0F0F0F (ARM7) J-Link>testwspeed Speed test: Writing 5 \* 8kb into memory @ address 0x00000000 ..... 8 kByte written in 4009ms ! (2.0 kb/sec) J-Link≻speed 1000 Speed: 1000kHz J-Link>testwspeed Speed test: Writing 5 \* 128kb into memory @ address 0x00000000 ..... 128 kByte written in 1847ms ! (71.0 kb/sec) J-Link>speed 4000 Speed: 4000kHz J-Link>testwspeed Speed test: Writing 5 \* 128kb into memory @ address 0x00000000 ..... 128 kByte written in 493ms ! (265.5 kb/sec) J-Link≻speed 8000 Speed: 8000kHz J-Link>testwspeed Speed test: Writing 5 \* 128kb into memory @ address 0x00000000 ..... 128 kByte written in 284ms ! (460.9 kb/sec) J-Link≻speed 12000 Speed: 12000kHz J-Link>testwspeed Speed test: Writing 5 \* 128kb into memory @ address 0x00000000 ..... 128 kByte written in 215ms ! (607.9 kb/sec) J-Link>testwspeed Speed test: Writing 5 \* 128kb into memory @ address 0x00000000 ..... 128 kByte written in 212ms ! (616.5 kb/sec) J-Link>testwspeed Speed test: Writing 5 \* 128kb into memory @ address 0x00000000 ..... 128 kByte written in 212ms ! (617.1 kb/sec) J-Link>

再看看 J-LINK 的原始测试数据:

J-Link 用户手册 Rev 3.0 - 11 -

SEGGER J-Link Commander U3.00h ('?' for help) Compiled Feb 2 2006 15:51:24 DLL version U3.00h, compiled Feb 2 2006 15:50:57 Firmware: J-Link compiled Jan 30 2006 11:25:41 ARM Rev.5 Hardware: U5.20 S/N : 1100001 UTarget = 3.290U Speed set to 30 kHz Found 1 JTAG device, Total IRLen = 4: Id of device #1: 0x3F0F0F0F Found ARM with core Id 0x3F0F0F0F (ARM7) J-Link>speed 12000 Speed: 12000kHz J-Link>testwspeed Speed test: Writing 5 \* 128kb into memory @ address 0x00000000 ..... 128 kByte written in 212ms ! (617.1 kb/sec) J-Link>\_

可以看到 SEGGER 提供的数据和实测性能几乎完全一样!

注意:由于 ARM7TDMI-S 内核的特殊性,LPC2000 系列的 JTAG 速度最高只能达到 1/6 系统时钟,一般最高是 4.8M,如果 JTAG 速度超过 4.8M, J-LINK ARM 将提示找不到 LPC2000。这 是 LPC2000 内核的局限,与 J-LINK 无关!同时在开发环境下调试 LPC2000 的时候,也注意 JTAG 时钟不能设置超过 4.8M,不然将 工作不稳定,甚至无法连接到目标芯片。其他芯片,如 ATMEL 的 SAM7 系列无此问题。

注意:由于 J-LINK 版本不断更新, 该测试数据也会有相应变化, 这里提供的测试数据主要是为了验证我们的产品和原装产品的性能 差异。

### 三、J-LINK 在各个主流开发环境下的设置

下面简单叙述一下在 Keil 下如何使用 J-Link:

Options for Target 'AT91SAM7S Flash'			
Device       Target       Output       Listing       C       Asm         C       Use       Simulator       Settings         Image: Limit Speed to Real-Time       Settings	LA Locate LA Misc Debug Utilities		
▼ Load Application at Sta ▼ Run to main() Initialization	✓ Load Application at Sta ✓ Run to main() Initialization		
Restore Debug Session Settings Breakpoints I Toolbox Watchpoints & Pi Memory Display	Restore Debug Session Settings Breakpoints V Toolbox Watchpoints Memory Display		
CPU DLL: Parameter: SARM. DLL -cAT91SAM7S	Driver DLL: Parameter:		
Dialog DIL: Parameter: DARMATS.DLL -p91SAM7S64	Dialog DLL: Parameter: TARMATS.DLL -p91SAM7S64		
确定 取消 Defaults 帮助			

选择 "RDI Interface Driver", 然后点击 "Settings":

RDI Interface Driver Setu	Ψ	
Browse for RDI Driver DLL C:\Program Files\SEGGER\JLinks	ARM_V320e\JLinkRDI.dll	
Debug Cache Options V Cache <u>C</u> ode V Cache <u>M</u> emory	Configure <u>R</u> DI Driver	]
	OK Cancel	<u>H</u> elp

请点击"…",指向到 JLINK 安装目录。

点击"Configure RDI Driver"出现以下几个选项卡: J-Link 用户手册 Rev 3.0 - 13 -

J-Link RDI Configuration				? 🗙
General       Init       JTAG       Flash         Image: State of the state o	Breakpoints CPU Log J-Link-RDI is an RDI compliant ARM. It requires a license (RDI) obtained from SEGGER (www.s This software is also capable of flash memory of several ARM m used to download your program the add. license "FlashDL") and number of software breakpoints the add. license "FlashBP"). Connection to J-Link © USB Device 0 © ICP/IP s will not be effective during	software for J-Link , which can be egger.com). programming the cros, which can be to flash (Requires I to set an unlimited in flash (Requires		
		确定	取消	应用(4)

如果是本机调试,直接使用 USB 口即可;如果是在局域网内调试,可以选择 TCP/IP,然后指定一个挂接了 J-LINK 的 PC 的 IP 地址。

J-Link RDI Configu	ration	? 🗙
General Init JTAG	Flash Breakpoints CPU Log	_
JTAG speed Auto selection Adaptive <u>clocking</u> 1000    KHz		
📕 JTAG scan chain with	n multiple devices	
Position 0	IR len 0	
0 is closest to TDI.	Sum of IRLens of devices closer to TDI. IRLen of ARM chips is 4.	
	⊻erify JTAG config	
	确定 取消 应用	( <u>A</u> )

设置 JTAG 速度,如果是-S 内核,建议使用 Auto 方式,如果是 非-S 内核,可以直接使用最高速度 12M。使用过程中如果出现不稳 定情况,可以将 JTAG 时钟速度适当调低。

J-Link RDI Configuration	
J-Link ROI Configuration         General Init JTAG Flash Breakpoints CPU Log         Imable flash programming         Allows programming the flash. This is required to download a program into flash memory or to set software breakpoints in flash (flash breakpoints).         Device AtmelAT91SAM7S64       Image: Click speed 48000000 Hz         RAM 16 KB @ address 0x200000       Image: Flash is girrored @ address 0x0         Image: Flash is girrored @ address 0x0       Image: Flash is girrored @ address 0x0         Image: Click clic	
	A.)

使能 FLASH 编程功能,如果你的目标芯片是带片内 FLASH 的 ARM,就可以使用该功能,这样子在调试前 J-LINK 就会先编程 FLASH。

J-Link RDI Configuration		? 🛛
General Init   JTAG   Flash Breakpoints   CPU   Log		
Software breakpoints (as opposed to hardware breakpoints) are breakpoints which modify program memory. This allows setting an unlimited number of breakpoints if the program is located in RAM.		
Allows setting an unlimited number of breakpoints if the program is located in RAM or flash, which is extremely valuable when debugging a program located in flash. This feature is available only if flash programming is enabled!		
Show info window during program		
	取消	应用 (4)

使用软件断点,如果是带片内 FLASH 的 ARM,建议使用该功能,可以打上n多断点,方便调试。

在这里可以设置 Reset 策略,有好几种 Reset 策略可选,同时可以设置 Reset 后的延迟时间,这个设置对于需要较长复位时间的芯片较为有用,如 AT91RM9200。

以上设置是用 XLINK 进行 Debug 的设置,如果要使用 KEIL 提供的 單即 "DOWNLOAD"功能则还需要在"Utilities"菜单里面进行和"Debug"一样的设置:

微控电子——	一专业的开发工具提供商	专业 ATARM 推广	~商
--------	-------------	-------------	----

Options for Target 'ICB2130 Flash'	×
Device   Target   Output   Listing   C   Asm   LA Locate   LA Misc   Debug   Utilities	
Configure Flash Menu Command	
O Use Target Driver for Flash Programming	
ULINK ARM Debugger 💌 Settings 🔽 Update Target before Debuggi	
Init File: ULINK ARM Debugger RDI Interface Driver Edit	
ULINK Cortex-M3 Debugger C Use External Tool for Flash Programming	
Command: IPC210x_ISP. EXE	
Arguments: "#H" ^X \$D COM1: 38400 1	
Run Independent	

选择"RDI Interface Driver", 然后点击"Settings"

Select Flash Programmer		×
J-Link Flash Programmer	•	
()		

选择"J-Link Flash Programmer"

RDI Interface Driver Setu	p	X
Browse for RDI Driver DLL C:\Program Files\SEGGER\JLinks	ARM_V324e\JLinkRDI. dll	
Debug Cache Options V Cache <u>C</u> ode V Cache <u>M</u> emory	Configure <u>R</u> DI Driver	
	OK Cancel	Help

接下来的设置就同"Debug"下设置一样了

完成以上设置后,就可以通过 🦉 按钮进行直接下载。注意,该

J-Link 用户手册 Rev 3.0 - 19 -

功能只支持具备片内 FLASH 的 ARM7/9 芯片。

### 在 ADS 下使用设置:

点击"Add",选择 JLINKRDI.DLL:

C	loose Targe	et		? 🔀
Г	Target Envir	onments		
	Target	RDI File	Version	Add
	ADP ARMIIT	1 C:\PROGRA~1\\Bin\Remote_A.dll	1.2.0.805	
	J-Link	1 C:\\JLinkRDI.dll	3.20e	<u>K</u> emove
	Multi-ICE	1 C:\\Multi-ICE.dll	2.2.6.1346	Re <u>n</u> ame
				<u>S</u> ave As
				Configure
	Please target has to	select a target environment from the abo environment to the list. Note that a tar be configured at least once before it ca	ve list or add a get environment n be used.	
		ОК	Cancel	Help

点击"Configure",出现以下内容:

J-Link RDI Configuration	? 🛛
J=Link RDI Configuration         General Init       JTAG       Flash       Breakpoints       CPU       Log         Image: Configuration       Image: Configuration       Image: Configuration       Image: Configuration       Image: Configuration         Image: Configuration       Image: Configuration       Image: Configuration       Image: Configuration       Image: Configuration         Image: Configuration       Image: Configuration       Image: Configuration       Image: Configuration       Image: Configuration         Image: Configuration       Image: Configuration       Image: Configuration       Image: Configuration       Image: Configuration         Image: Configuration       Image: Configuration       Image: Configuration       Image: Configuration       Image: Configuration         Image: Configuration       Image: Configuration       Image: Configuration       Image: Configuration       Image: Configuration         Image: Configuration       Image: Configuration       Image: Configuration       Image: Configuration       Image: Configuration         Image: Configuration       Image: Configuration       Image: Configuration       Image: Configuration       Image: Configuration       Image: Configuration       Image: Configuration       Image: Configuration       Image: Configuration       Image: Configuration       Image: Configuration       Image: Config	
<ul> <li>✓ Allow flash <u>d</u>ownload</li> <li>Allows program download to flash. Your debugger does not need to have a flash loader. This feature requires an additional license (FlashDL).</li> <li>✓ Show info window during download</li> </ul>	
确定	<b>取消</b> 应用 (A)

进入 AXD 后的信息(注意 LOG FILE 的内容):

微控电子——专业的开发工具提供商	专业 ATARM 推广商
------------------	--------------

(Ø AXD	
Zile Search Processor Views System Views Execute Options Window Help	
nn 🖻 🖆 🖳 🙀 🐂 🛛 🗖 🖬 🔲	
Target Inage Fid >	
<b>⇒</b> ₩ ARM_1	
System Output Monitor	
RDI Log Debug Log	
Log file:	
J-Link RDI DLL V3.20e, compiled May 8 2006 09:35:21 Lil ink ABM DLL V3.20e, compiled May 8 2006 09:34:53	
Firmware: J-Link compiled Apr 27 2006 12:55:19 ARM Rev.5	
Hardware: V5.00	
DEM : IAR	
Feature(s): RDI,FlashDL,FlashBP,JFlash	
Found 1 JTAG device, Total IRLen = 4:	
Id of device #1: 0x3F0F0F0F Exclud APM with core Id 0x3F0F0F0F (APM 7)	
ARM RDI 1.5.1 -> ASYNC RDI Protocol Converter ADS v1.2 [Build number 805]. Copyright (c) ARM Lim	ited 2001.
	2
For Help, press F1	(No Pos> J-Link ABM_1 (No Image Name> //

### 在 IAR 下使用设置:

在 IAR 既可以使用 IAR 提供的 JLINK 的驱动,也可以使用 RDI 接口的驱动,推荐使用 RDI 接口的驱动,因为 IAR 版本的 JLINK 对 速度和功能做了限制。

首先打开一个工程,然后按照下图开始进入设置页面:

💥 IAR Embedde	d Workbench IDE		
<u>F</u> ile <u>E</u> dit <u>V</u> iew	Project Tools Mindow	<u>H</u> elp	
Workspace Flash_debug Files Basic - Fl	Add <u>F</u> iles Add <u>G</u> roup <u>I</u> mport File List Edi <u>t</u> Configurations Remo <u>v</u> e		rs
│	Create <u>N</u> ew Project Add <u>E</u> xisting Project.		an be read a requires to id jumn. For
H = I = xRessou H = I = Output	Options So <u>u</u> rce Code Control	Alt+F7	exeption occ
	<u>M</u> ake Compile Re <u>b</u> uild All	<b>F7</b> Ctrl+F7	In
	C <u>l</u> ean B <u>a</u> tch build	F8	sw:
	<u>Debug</u>	Ctrl+D	pal
	rsvdve	B	rsv
	firmer	в	IR

Options for node	"Basic"
Category: General Options C/C++ Compiler Assembler Custom Build Build Actions Linker Debugger Simulator Angel IAR ROM-monitor J-Link/J-Trace Macraigor RDI Third-Party Driver	Target Output Library Configuration Library options M] Processor variant C Core ARM7TDMI Device Atmel AT91SAM7S64 PPU None Processor mode C Arm C Arm C Little D Endian mode Stack align C 4 bytes C 8 bytes
	Cancel

Options for node	"Basic"
Category: General Options C/C++ Compiler Assembler Custom Build Build Actions Linker Debugger Simulator Angel IAR ROM-monitor J-Link/J-Trace Macraigor RDI Third-Party Driver	Factory Settings         Output       Extra Output #define       Diagnostics       List       Confi:         Output file       Oyerride       Ogerride       Secondary output file:         Dasic. d79       Wone for the selected for         Format       • Debug information for C-SPY       With runtime control mod         Image: With I/O emulation modu       • Buffered terminal output         Image: With I/O emulation modu       • Buffered terminal output         Image: Other       Output       elf/dwarf         Image: Other       Output       elf/dwarf         Format variant:       Arm compatible (-yas)       Image: Other
	Module-local Include all

Options for node	"Basic"
Category: General Options C/C++ Compiler Assembler Custom Build Build Actions Linker Debugger Simulator Angel IAR ROM-monitor J-Link/J-Trace Macraigor RDI Third-Party Driver	Factory Settings         Output Extra Output #define Diagnostics List Confit         Generate extra output file         Output file         Oyerride default         Basic.sim         Lormat         Output format:         simple=code         Format variant:
	OK Cancel

C/C++ Compiler Assembler Custom Build Build Actions Linker Debugger Simulator Angel IAR ROM-monitor J-Link/J-Trace Macraigor RDI Third Party Driver	Setup Download Extra Options Plugins           Driver       Run to         J-Link/J-Trace       main         Angel       IAR ROM-monitor         J-Link/J-Trace       Main         Macraigor       RDI         Third-Party Driver       SAM7.mac         Devige description file       Override default
J-Link/J-Trace Macraigor RDI Third-Party Driver	Devige description file 

如果购买的是 IAR 版本的 J-LINK, 请选择"J-LINK/J-TRACE"; 如果购买的是全功能版本 J-LINK,则既可以选择"J-LINK/J-TRACE", 也可以选择"RDI", 建议选择"RDI", 以提升性能。

Options for node	"Basic"
Category: General Options C/C++ Compiler Assembler Custom Build Build Actions Linker Debugger Simulator Angel IAR ROM-monitor J-Link/J-Trace Macraigor RDI Third-Party Driver	Factory Settings         Setup       Download       Extra Options       Plugins         Driver       Run to         RDI       main         Angel       main         JAR ROM-monitor       main         J-Link/J-Trace       main         Macraigor       main         SPRUJ_DINS\resource\SAMT.mac          Devige description file          Qverride default
	OK Cancel

如果选择 "J-LINK/J-TRACE",则无需额外设置:

General Options C/C++ Compiler Assembler Custom Build Build Actions Linker Debugger Simulator Angel IAR ROM-monitor J-Link/J-Trace Macraigor RDI Third-Party Driver	Setup       Connection         Reset
--	--------------------------------------

### 如果选择 "RDI",则还需要指定 JLINKRDI.DLL 的位置:

Options for node	"Basic"	
Category: General Options C/C++ Compiler Assembler Custom Build	RDI   M <u>a</u> nufacturer RDI driver	Factory Settings
Build Actions Linker Debugger Simulator Angel IAR ROM-monitor	C:\Program Files\SEGGER\JLinks	RM_V320h\JLinkRDI. dll (.) Note Use the RDI menu to specify additional driver settings. (This menu is available after the RDI
J-Link/J-Trace Macraigor RDI Third-Party Driver	Log RDI <u>communication</u> \$TOOLKIT_DIR\$\cspycomm.log	Catch exceptions <u>Reset</u> <u>Data</u> <u>F</u> IQ <u>Undef</u> <u>Prefetch</u> <u>SWI</u> <u>IRQ</u>
		OK Cancel

设置完成后将多出一个 RDI 菜单,如下图:

🄀 IAR Embedded Workh	ench IDE
<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>P</u> roject	RDI Tools Mindow Help
🗅 🗲 🖬 🖪 😹 🖇	Configure
Workspace	ETM Trace Window 32.C
Debug	Trace Setup 1_task_id;
Files දී	Trace Saye /* (RTOS_U.
□       Image       Image         Image       Image       Image <td><pre>Breakpoint Usage D == RTOS /* If the main_task lix following defines a; */ #define MAIN_TASK_ROM_: #define MAIN_TASK_RAM_: #endif /* (RTOS_U. /* int main(void) { #if (RTOS_USED == RTOS]</pre></td>	<pre>Breakpoint Usage D == RTOS /* If the main_task lix following defines a; */ #define MAIN_TASK_ROM_: #define MAIN_TASK_RAM_: #endif /* (RTOS_U. /* int main(void) { #if (RTOS_USED == RTOS]</pre>

在 RDI 菜单下有"CONFIGURE"选项,这里可以对 JTAG 时钟, FLASH,断点,CPU 等进行设置,请注意里面的 FLASH 和 CPU 型 号与目标板相吻合。

另外, IAR 下使用 J-LINK 的时候, 注意不要再使用 IAR 自带的 FLASHLOADER 软件进行 FLASH 下载:

Options for node	"Teb2"
Category: General Options C/C++ Compiler Assembler	Factory Settings Setup Download Extra Options Plugins
Custom Build Build Actions Linker Debugger Simulator Angel IAR ROM-monitor J-Link/J-Trace Macraigor BDI	✓ Verify download Suppress download ✓ Use flash loaded ,,,, (default), ATMEL Edit
Third-Party Driver	OK Cancel

请将"Use flash loader"前的勾去掉,使用 J-LINK 的 FLASH 编 程算法和使用 IAR 的 FLASHLOADER,速度可能差好几倍!

### 四、J-FLASHARM 使用设置

安装完 JLINK 的驱动后会出现两个快捷图标,其中一个是 J-FLASH ARM,这个应用程序是用来单独编程 FLASH 的(需要 J-FLASH ARM License 支持):

🛃 J-Flash ARM V3.20h - [C:\Program Files\SEGGER\JLinkARM_V320h\Samples\JFlash\ProjectFiles\AT91S 🗐 🗖 🔀																						
<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>T</u> a	rget <u>O</u> ptions <u>)</u>	Mindow <u>H</u> elp																				
Project		BASH C:\Docu	men	rt s	and	Set	tin	lgs\	ncu	zone	€\桌	面∖∶	san 1	′ <b>x−</b> t	c. b	in						$\mathbf{X}$
Name Value		Address: 0x10		0	x1	x2	×4															
Connection USB	Device 0]			-	1.04		<u> </u>															
		Address	0	1	2	3	4	5	6	7	8	9	A	B	С	D	E	F	ASCII			
Init JTAG speed 30 kF	Z	1000000	ØF	00	00	EA	FE	FF	FF	EA	FE	FF	FF	EA	FE	FF	FF	EA				
TAP number < not	ised>	1000010	FE	FF	FF	EA	FE	FF	FF	EA	19	00	00	EA	00	90	AØ	E1				
IR len <not< th=""><th>ised&gt;</th><th>1000020</th><th>04</th><th>01</th><th>98</th><th>E5</th><th>D3</th><th>FØ</th><th>21</th><th>E3</th><th>ØE</th><th>50</th><th>2D</th><th>E9</th><th>ØF</th><th>EØ</th><th>AØ</th><th><b>E1</b></th><th></th><th>• P</th><th></th><th></th></not<>	ised>	1000020	04	01	98	E5	D3	FØ	21	E3	ØE	50	2D	E9	ØF	EØ	AØ	<b>E1</b>		• P		
		1000030	10	FF	2F	E1	ØE	50	BD	E8	D1	FØ	21	E3	09	00	AØ	E1	/1	P	• • • • • • •	
Chip Atmel	AT91SAM7X	1000040	Ø4	FØ	5F	E2	81	D9	AØ.	E3	90	00	9F	E5	ØF.	EØ	AØ	E1	^			
Clock speed 4792	3200 Hz	1000050	10	FF	2F	F1	81	<b>6</b> 9	60	E3	D1	FØ	21	E3	80	20	96	E2			•	
Endian Little		1000050	10	PO	21	E2	00	D07	00	EJ E4	60	1.9	40	E3	40	DO DO	24	E3				
ABM core Id 0x3E0	FOFOF	1000000	00	L D D	21	E3	60	00	не 0 Г	EL	00	50	40	EZ E4	13	ге ге	21 0E	E3			e	
Use target RAM Yes		1000070	99	שע	нø	EI	50	99	44	E5	96	EQ	ню	EI	10	FF	ZF	EL			/-	
RAM address 0x200	000	1000080	FE	FF	FF	EA	64	ЕØ	<b>4</b> E	E2	99	40	2D	EA	90	EЮ	4F	E1		.N@	0.	
RAM size 16 KE		1000090	00	40	2D	E9	01	00	2D	E9	44	EØ	9F	E5	00	01	9E	E5	.e	D.		
Use DCC mode Yes		10000A0	00	E1	8E	E5	13	FØ	21	E3	ØE	50	2D	E9	ØF	EØ	AØ	E1		. • P		
Elash memoru AT91	SAM7V256 int	10000B0	10	FF	2F	E1	ØE	50	BD	E8	92	FØ	21	E3	20	EØ	9F	E5	/I	P	t	
Manufacturer Atmel	JAMTA230 IRC	10000C0	30	E1	8E	E5	01	00	BD	E8	00	40	BD	E8	ØE	FØ	6F	E1	0	e		
Size 256 K	В	10000D0	00	80	FD	E8	FE	FF	FF	EA	FE	FF	FF	EA	FE	FF	FF	EA				
Flash Id 0x0		10000E0	FØ	03	ØØ	00	00	FØ	FF	FF	EC	ØØ	00	ЮЙ	ØØ	ØØ	00	EA				
Check flash Id No		1000070	F4	00	00	FO	28	00	8F	F2	00	ØC.	90	FS	00	00	80	FØ	(			
Base address Ux100	UUU 1. akia	1000100	0-1	70	40	E2	00	DIA	00	TO	60	00	50	E4	ED	00	00	60		•••••	7	
Organization 52 bit	sxicnip	1000100	OF	60		EQ	14	EO	40	E0	01	00	10	ET	62	EO	49	10	. po		۰۰۰۰۰ ۲	
		1000110	бь	99	БН	LO	14	ЕØ	41	EZ	ЮT	99	13	EG	63	гө	47	10		.0		
🔜 LOG																						
Opening project file [C:\Yrogram Files\SEGGER\JLinkARM_V320h\Samples\JFlash\ProjectFiles\AT91FR40162.jflash]																						
(- Project opened successfully Denning data file (')Program Filestar Sucters/Frhedded Workbarch 4314/484/srokeren]arkitms/AfO1FRev/Flacher4r404/Frakat01 about his																						
- Data file opened successfully (10588 bytes, 1 range, CKC = Ox785456D6)																						
Close project																						
- rroject closed Dpening project file [C:\Program Files\SEGGER\JLinkdRM_V320h\Samples\JFlash\ProjectFiles\AT91SAMTX256.jflash]																						
- Project opened successfully																						
Upening data file U: Wocuments and Settings\ncurone\kkklekklekklekklekklekklekklekklekklek																						
																						~
<							1														11	> .:
Ready																						//

首次使用的时候应该在 File 菜单,选择 Open Project,选择你的

目标芯片:

Open project	:		? 🗙
查找范围(I):	🗁 ProjectFiles		
<ul> <li>表最近的文档</li> <li>夏面</li> <li>美面</li> <li>大的文档</li> <li>支約</li> <li>大的电脑</li> <li>一次</li> <li>一次<th>Image: ADuC7020. jflashImage: ADuC7030. jflashImage: ADuC7032. jflashImage: ADuC7032. jflashImage: ADuC7229. jflashImage: ADUC7290. jfla</th><th>m AT91SAMTS256. jflash m AT91SAMTS256. jflash m AT91SAMTX128. jflash m AT91SAMTX256. jflash m DragonballMX1. jflash m EvaluatorTT. jflash m LH7A40x_LogicPD. jflash m LH7540x_LogicPD. jflash m LH79520_Log 修改日期: 2006- 大小: 8.09 KB m LPC2106. jfl m LPC2108. jflash m LPC2148. jflash m LPC2294. jflash m LPC2294_PhyCORE. jflash</th><th>MAC7111. MS9360. j NS9360. j PCF87750 S3F445HX SocLiteP STR710. j 4-27 16:07 j j STR912. j STR912. j STR912. j TMS470R1 TMS470R1 TMS470R1 TMS470R1</th></li></ul>	Image: ADuC7020. jflashImage: ADuC7030. jflashImage: ADuC7032. jflashImage: ADuC7032. jflashImage: ADuC7229. jflashImage: ADUC7290. jfla	m AT91SAMTS256. jflash m AT91SAMTS256. jflash m AT91SAMTX128. jflash m AT91SAMTX256. jflash m DragonballMX1. jflash m EvaluatorTT. jflash m LH7A40x_LogicPD. jflash m LH7540x_LogicPD. jflash m LH79520_Log 修改日期: 2006- 大小: 8.09 KB m LPC2106. jfl m LPC2108. jflash m LPC2148. jflash m LPC2294. jflash m LPC2294_PhyCORE. jflash	MAC7111. MS9360. j NS9360. j PCF87750 S3F445HX SocLiteP STR710. j 4-27 16:07 j j STR912. j STR912. j STR912. j TMS470R1 TMS470R1 TMS470R1 TMS470R1
	<ul> <li></li></ul>		>
	文件名 (M): <b>*. jflash</b>	•	打开 (0)
	文件类型(I): J-Flash project	t files (*.jflash) 💌	取消

然后通过"File"菜单下的"Open..."来打开需要烧写的文件,可以是.bin 格式,也可以是.hex 格式,甚至可以是.mot 格式。注意起始地址。

接下来在"Options"选择"Project settings":

Project	t settings				? 🗙
General Chip Cloc <u>k</u> E <u>n</u> dian	Atmel AT 91 SAM	Flash   F 17X256 ] Hz ]	roduction	Check ARM core ID ID 3F0F0F0F Use target <u>RAM (faster)</u> Addr 200000 16 KB 💌 Enable DCC <u>m</u> ode (faster)	
#         T.           0         R:           1         W           2         W           3         W           4         D:           5         W           6         D:           7         W	ype eset /rite 32bit /rite 32bit elay /rite 32bit elay /rite 32bit	Value0 0xFFFFFD44 0xFFFFF60 0xFFFFFC20 0xFFFFFC2C 0xFFFFFC2C	Value1 0 ms 0x00008000 0x00320300 0x00000601 200 ms 0x00191C05 200 ms 0x00000007	Comment Disable watchdog Set flash wait states Set PLL Set PLL and divider Select master clock and	
<u>A</u> d	ld <u>I</u> nser	t Dejete	e <u>E</u> dit	Up Down	

在 ARM 选项卡可以选择目标芯片,如果不是具备片内 FLASH 的芯片的话请选择 "Generic ARM7/ARM9"。

Project settings	? 🗙
General JTAG ARM Flash Production	
FlashBank Bank[0] Add Remove	
Base Addr 00100000 Organization 32 🖵 Bits x 1 🖵 Chip(s)	
Use custom <u>B</u> AMCode	
Manufacturer Atmel Chip AT91SAM7X256 internal Size 256 KB Sectors 1024 C Start/End sector Start Addr Sector(0): 0x0 V End Addr Sector(1023): 0x3FFFF V Sector(2): 0x20 - 0x2FF	
Selected ranges: 1024 Sectors, 1 Range: 0x0000 · 0x3FFFF ✓ Sector[3]: 0x300 · 0x3FF ✓ Sector[4]: 0x400 · 0x4FF ✓ Sector[5]: 0x500 · 0x5FF ✓ Sector[6]: 0x600 · 0x6FF ✓ Sector[6]: 0x600 · 0x6FF	
确定	<b>取消</b> 应用 (A)

FLASH 选项卡,如果之前是"Opon project"这里就不需要设置, 默认即可,如果是自己新建的 project,则需要小心设置。

如果前面的 ARM 选项卡里选择的是"Generic ARM7/ARM9", 则可以在 FLASH 选项卡里面选择 FLASH 型号:

Project settings	? 🛛
General   JTAG   ARM Flash   Production	
FlashBank Bank[0]  Add Remove	
Base Addr 000000000 Organization 16	
Use custom <u>B</u> AMCode	
Manual flash selection Select flash device	
Manufacturer Check manufacturer flash Id	
Size Sectors Buswidth Id	
C Start/End seator	
Start Addr	
End Addr	
Selected ranges:	
▲I None Invert	

Select flas	h device						X
Manufacturer ×	•						
Manufacturer	Device	Size	NumSectors	8bit Id	16bit Id	Buswidth	
AMD	Am29DL161DB	2048 KB	39	10039	12239	16	
AMD	Am29DL161DT	2048 KB	39	10036	12236	16	
AMD	Am29DL162DB	2048 KB	39	1002E	1222E	16	
AMD	Am29DL162DT	2048 KB	39	1002D	1222D	16	
AMD	Am29DL163DB	2048 KB	39	1002B	1222B	16	
AMD	Am29DL163DT	2048 KB	39	10028	12228	16	
AMD	Am29DL164DB	2048 KB	39	10035	12235	16	
AMD	Am29DL164DT	2048 KB	39	10033	12233	16	
AMD	Am29DL322DB/GB	4096 KB	71	10056	12256	16	
AMD	Am29DL322DT/GT	4096 KB	71	10055	12255	16	
AMD	Am29DL323DB/GB	4096 KB	71	10053	12253	16	
AMD	Am29DL323DT/GT	4096 KB	71	10050	12250	16	
AMD	Am29DL324DB/GB	4096 KB	71	1005F	1225F	16	
AMD	Am29DL324DT/GT	4096 KB	71	1005C	1225C	16	
AMD	Am29DL400BB	512 KB	14	1000F	1220F	16	
AMD	Am29DL400BT	512 KB	14	1000C	1220C	16	
AMD	Am29DL800BB	1024 KB	22	100CB	122CB	16	
AMD	Am29DL800BT	1024 KB	22	1004A	1224A	16	
AMD	Am29DS323DB	4096 KB	71	100B8	122B8	16	~
						_	_
OK						Cancel	

\_\_\_\_\_\_支持非常多的 FLASH 器件,只要是大厂的 FLASH,基本都可以

找到! 而且会不断升级以支持最新器件。

设置好之后,就可以到 Target 里面进行操作,一般步骤是先 "Connect",然后"Erase Chip",然后"Program",可以自己慢慢体 会。大部分芯片还可以加密,主要的操作都在 Target 菜单下完成。

从 3.30g 版本开始, J-FLASH ARM 开始支持 XSCALE:

### 微控电子——专业的开发工具提供商 专业 ATARM 推广商

Project settings		? 🛛
General   JTAG CPU   Flash   Produc	tion	
<ul> <li>C PU XScale</li> <li>✓</li> <li>MCU ARM7/ARM9</li> <li>XScale</li> <li>Endian Little ▼</li> </ul>	☐ C <u>heck core ID</u> ID 00000000 ☐ Use target <u>B</u> AM (faster) Addr 0 4 KB ▼	
Use following init sequence:           # Type         Value0         Value	1 Comment	
Add Insert Delete	Edit Down	
	福完	

如有任何问题,可以到 BBS 发问,同时请多多关注网站,以获得最新说明和最新范例!

### 五、JLINK F.A.Q

2Q: JLINK 和其他 JTAG 调试工具相比有什么优势?

2A: 全功能版本的 JLINK (XLINK) 具有如下主要特点:

 1),支持ADS,KEIL,IAR,WINARM,RV等几乎所有开发环境(RDI License 支持);

2),支持FLASH软件断点,突破一般ARM仿真器2个FLASH断 点的限制,可以设置无穷个FLASH断点,极大的提高调试效率(F1a sh BP License 支持);

3),支持FLASH编程,可以在各个开发环境下轻松编程FLASH(Flash DL License 支持);

4),具备单独烧写 FLASH 的独立软件,提高生产效率(J-FLASH ARM License 和 J-FLASH ARM 软件支持);

5),超快速度,编程速度和调试速度在目前已知调试工具里面 最快(达到 600K,请参考 XLINK 用户手册);

6),支持几乎所有 ARM7, ARM9,暂时不支持 XSCALE(支持器件 列表请参考 XLINK 用户手册),从 3.30g版本开始 J-FLASH ARM 软件
已经可以支持 XSCALE 系统的 FLASH 编程;

> J-Link 用户手册 Rev 3.0 - 40 -

目前,ULINK (SMARTDEBUGGER) 只能在 KEIL 下使用; MULTI-IC E (本站提供并口,USB 两个版本)可以在 ADS、IAR 下使用,在 IAR 下使用的时候可以利用 IAR 的 FLASHLOADER 进行 FLASH 编程,但是在 ADS 下使用的时候缺少编程插件; EASYJTAG 只能在 ADS 下使用; WIG GLER 可以在各个开发环境下使用,但是目前只能在 IAR 下用 MACRAI GOR 的驱动,才能编程 FLASH,而且速度很慢;而 JLINK 可以在各种 开发环境下调试、下载程序!

3Q: JLINK 提供升级以支持新器件么?

3A: 可以到 <u>www.segger.com</u> 网站下载 JLINK (XLINK) 安装程序(驱动), segger 升级较快,请密切关注。如果需要更改 JLINK (XLINK) 的授权,比如将 IAR 版本升级到全功能版本,请直接发回给我们进行升级,最终补版本差价即可。

4Q:为什么我购买的 JLINK 在 KEIL 以下不能使用,出现以下错误:



J-Link 用户手册 Rev 3.0 - 41 -



4A: J-LINK 提示没有相应的 license,亦即没有授权,亦即您购买的J-LINK 并不附带 RDI License,是 IAR 版本的 J-LINK,如果需要 RDI 接口的 License 可以发回给我们进行升级,升级按照功能收费。

5Q: J-Link 的驱动程序和应用程序是否可以免费升级?

5A: Segger 网站升级较快,建议大家经常关注一下,J-Link 的驱动程 序和应用程序是免费升级的,只要你购买了某个功能的 License,该 部分功能就可以永远免费升级,当下载了新版本的 J-LINK 程序后, 只要插上 J-LINK, 然后运行 J-LINK ARM.EXE, 就可以实现 J-LINK 的固件升级,如下:



请注意看图中的第3-5行的信息,软件提示升级成功。

6Q: 我在 KEIL 下面调试 LPC2142,为了达到最快的速度,我在 Configure 里面将 JTAG 速度从 AUTO 修改到 12M,但是系统提示:



请问这个是什么问题?如何解决?另外,用AUTO的话就没有问题。 6A:这个是由LPC2000的内核特殊性所决定的。LPC2000的内核是 ARM7TDMI-S,是可综合版本的ARM7TDMI,即PHLIPS有权限来 对ARM7TDMI进行部分改动,主要是调试接口的改动,LPC2000采 用的JTAG接口包含了一个RTCK引脚,这个引脚是用来同步JTAG 调试时钟用的,当TCK发送一个时钟,该时钟经过一定延迟后就由 RTCK返回,如果接收不到返回的时钟,系统就会提示找不到目标芯 片,即调试失败。经过测试,LPC2000系列ARM7TDMI-S最高只能 稳定工作在4800KHz频率下,再高就会出现以上错误提示。由于 ULINK使用的最高JTAG只能达到1M,所以在使用ULINK的时候 根本就不会出现这个问题。从另一个侧面讲,亦即调试 LPC2000 的 时候,JLINK的速度最高可以是ULINK的4.8倍。

7Q:我使用 IAR 开发环境,为什么用 J-LINK 的 FLASH 下载速度和用 MULTI-ICE 的下载速度差不多?

J-Link 用户手册 Rev 3.0

- 43 -

7A:使用 IAR 的时候请注意不要使用 IAR 自带的 FLASHLOADER 进行 FLASH 下载,而应该使用 J-LINK 的 FLASH 编程算法,关键一 点是将"USE FLASH LOADER"前的勾去掉,如下图:

Options for node	Teb2T
Category: General Options C/C++ Compiler Assembler Custom Build Build Actions Linker Debugger Simulator Angel IAR ROM-monitor J-Link/J-Trace Macraigor RDI Third-Party Driver	Factory Settings         Setup       Download         Attach to progr         Yerify download         Suppress download         Use flash loadei         ,,,, (default), ATMEL         Edit
l	
	OK Cancel

使用 FLASH LOADER 进行 FLASH 下载和使用 J-LINK 进行 FLASH 下载的速度要差好几倍!

### 附录一 使用 jlink 的 GDBserver

Team MCUzone

GDB 作为开源的调试器,其使用比较广泛,是使用 gcc 的标配调试器。

在 segger 官方推出 GDBserver 之前,网上也有个人提供的 jlinkgdbserver,但是效果不 是很好。需要注意的是,后者在一般的 jlink 上即可使用,而 segger 官方的还需要一个 GDBfull license 的授权,需要额外的费用。

一,软件安装

请先到 <u>http://www.segger.com/download gdb.html</u>下载包含有 GDBserver 的软件, 然后 安装。

安装完成后,请连接好 jlink 与目标板,在 pc 端运行 jlink GDBserver,正确的显示如图:

508 J-Link GDB Server ▼3.48b									
<u>F</u> ile <u>H</u> elp									
Debugger Waiting for connection J-Link Connected Target ARM7, Core Id: 0x3F0F0F0F	Initial JTAG speed 1000 ▼ kHz Current JTAG speed 1000 kHz 3.31 V Little endian ▼	<ul> <li>Stay on top</li> <li>Log window</li> <li>Log to file</li> <li>Cache reads</li> </ul>							
Log output:		<u>C</u> lear log							
J-Link GDB Server V3.48b		<u></u>							
JLinkARM.dll V3.48b (DLL com	piled Oct 10 2006 11:40:	57)							
J-Link connected Firmware: J-Link compiled Sep 28 2006 10:05:22 ARM Rev.5 Listening on TCP/IP port 2331 J-Link found 1 JTAG device, Total IRLen = 4									
JTAG ID: Ux3FUFUFUF (ARM7)									
2		×							
0 KB downloaded 1 JT	AG device								

可以看到目标器件的类型, ID, 目标板电压等。

说明此时 GDBserver 已经与目标器件建立了联系,等待 GDB 从端口 2331 来连接。 为了使用 GCCARM 来编译软件,还需要安装 GNUARM 或者 WinARM。

二,调试

使用 GCCARM 编译应用,最终会生成一个 elf 文件,注意在编译的时候要打开调试信息的选项,比如使用参数-gdwarf-2。 运行 erm elf insight 这是个图形化的 APM CDP 加下图。

运行 arm-elf-insight,这是个图形化的 ARM GDB,如下图:

J-Link 用户手册 Rev 3.0 - 45 -



使用 file→open, 打开前面所创建的 elf 文件, insight 中的显示将如下

•	74 main. c - Source Vindov		
Γ	<u>F</u> ile <u>R</u> un <u>V</u> iew Control <u>P</u> references <u>H</u> elp		
	🚿 🕐 🕐 🐌 🚯 🗑 👗 🔏 📾 🍕 🖾 🖉 🖾 Find:		<b>*</b>
	main.c 🔻 main 💌	SOURCE	•
	25 //* Function Name       : change_speed         26 //* Object       : Adjust "LedSpeed" value depending on SW1 and SW2 are pressed or         27 //* Input Parameters       : none         28 //* Output Parameters       : Update of LedSpeed value.         29 //*	not	-
	30 static void change_speed ( void ) 31 {//* Begin - 32 if ( (AT91F_PIO_GetInput(AT91C_BASE_PIOA) & SW2_MASK) == 0 )		
	33 { - 34 if ( LedSpeed > SPEED ) LedSpeed -=SPEED ; 35 }		
	- 36 if ( (AT91F_PI0_GetInput(AT91C_BASE_PI0A) & SW3_MASK) == 0 ) 37 { - 38 if ( LedSpeed < MCK ) LedSpeed +=SPEED ; 39 } 40 }//* End 41		
	42 //*43 //*43 //* Function Name : wait 43 //* Function Name : wait 44 //* Object : Software waiting loop 45 //* Input Parameters : none. Waiting time is defined by the global variable LedSpeed. 46 //* Output Parameters : none 47 //*		
	<pre>48 static void wait ( void ) - 49 {//* Begin 50 volatile unsigned int waiting_time ; 51 change_speed () ; - 52 for(waiting_time = 0; waiting_time &lt; LedSpeed; waiting_time++) ; - 53 }//* End 55</pre>		

然后点击 run→run, 在弹出的 target select 中按照如下设置

J-Link 用户手册 Rev 3.0 - 46 -

7% Target Selection	X
Connection	🔽 Set breakpoint at 'main'
Target: GDBserver/TCP	🗖 Set breakpoint at 'exit'
Hostname: localhost	□ Set breakpoint at
Port: 2331	└ Display Download Dialog
▷ More Options	OK Cancel Help

注意端口号一定要与 GDBserver 提供的一致。点击 ok,即可连接。正确连接后如下图:

_												
	49	reset:										
	50	/*										
	51	1 //*- Exception vectors										
	52	2 //*										
	53	3 //*- These vectors can be read at address 0 or at RAM address										
	54	¥ //∗- They ABSOLUTELY requires to be in relative addresssing mode in order to										
	55	//*- guarantee a valid jump. For the moment, all are just looping.										
	56	//*- If an exception occurs before remap, this would result in an infinite loop.										
	57	//*- To ensure if a exeption occurs before start application to infinite loop.										
	58	//**/										
	59											
-	60	В	InitRese	et /* 0x00 Reset handler */	-							
	61	undefvec:										
-	62	В	undefved	c /* 0x04 Undefined Instruction */								
	63	swivec:										
-	64	В	swivec	/* 0x08 Software Interrupt */								
	65	pabtvec:										
-	66	В	pabtvec	T THE COR COMPANY AND								
	67	dabtvec:		GDE J-LINK GDD Server VJ. 40D								
-	68	В	dabtvec	<u>F</u> ile <u>H</u> elp								
	69	rsvdvec:										
-	70	В	rsvdvec	Debugger Connected to 127.0.0.1 Initial JTAG speed 1000 👻 kHz 💻	Stay on top							
	71	irqvec:			Log <u>w</u> indow							
-	72	В	IRQ_Han	J-Link  Connected Current JTAG speed   1000 kHz	Log to file							
	73	fiqvec:		Target ABM7 Halted	Cache reads							
	74	/*										
	75	//*- Function :	FIQ_Han		Clearlog							
	76	//*- Treatments :	FIQ Con	Log output:								
	77	//*- Called Functions :	AIC_FVR	Connected to 127.0.0.1	<u>^</u>							
	78	//*		Reading all registers Read 4 bytes @ address 0x00000000 (Data = 0xF&000010)								
	79											
	80	FIQ_Handler_Entry:										
	81											
	82	/*- Switch in SVC/User Mode	to allo									
	83	/* hecause the FIN is not up	t ackno									

GDBserver 中会显示已连接,同时 insight 中指令也会停在起始位置。此时就可以开始调试了,比如按 s 单步进入。

由此可见,新加入的 GDB full license 对 GDB 的支持更好,使用其来调试也较方便。

### 附录二、J-Flash ARM 命令行使用说明

为了方便扩展使用, J-Flash ARM 还提供了命令行方式。

需要注意的是,默认安装目录是 program files 文件夹下,而这个路径存在一个空格(即 program 和 files 中间的空格),而这在命令行方式下是不允许的,所以,如果使用命令行,需要更改安装路径,或者把工程文件和目标文件放到别的目录下。

J-FLASH ARM 主要有以下命令:

Valid command line options:           -openprj         Opens an existing project Syntax: -openprj           -saveprjas         Saves current project in a different file Syntax: -saveprjas           -saveprj         Saves current project Syntax: -saveprja           -saveprj         Saves current project Syntax: -saveprj           -open         Opens a data file Syntax: -open <filename>[,<saddr>]           -saveas         Saves current data file Syntax: -saveas           -saveas         Saves current data file Syntax: -saves (SADDR&gt;, <eaddr>]           -save         Syntax: -save[<saddr>, <eaddr>]           -save         Syntax: -delrange <saddr>, <eaddr>]           -elerinate         Eliminates blank areas in data file           -connect         Connects from target           -softlock         Locks (hard) selected sectors           -hardunlock         Unlocks (soft) selected sectors           -hardunlock         Unlocks (hard) selected sectors           -hardunlock         Unlocks (hard) selected sectors</eaddr></saddr></eaddr></saddr></eaddr></saddr></eaddr></saddr></eaddr></saddr></eaddr></saddr></eaddr></saddr></eaddr></saddr></filename>	Comman	dline	
<ul> <li>openprj Opens an existing project Syntax: -openprj<filename> </filename></li> <li>-saveprjas Saves current project in a different file Syntax: -saveprja</li> <li>-saveprj Saves current project Syntax: -saveprj</li> <li>-open Opens a data file Syntax: -saveprj</li> <li>-open Saves current data file in a different file Syntax: -saveas</li> <li>-saveas Saves current data file in a different file Syntax: -saveas</li> <li>-savea Saves current data file Syntax: -saveas</li> <li>-save Relocates data by given offset Syntax: -relocate <offset> </offset></li> <li>-delrange Deletes data range Syntax: -relocate <offset> </offset></li> <li>-delrange Deletes data range</li> <li>-softlock Locks (soft) selected sectors</li> <li>-softlock Locks (soft) selected sectors</li> <li>-softlock Locks (soft) selected sectors</li> <li>-hardunlock Unlocks (hard) selected sectors</li> <li>-hardunlock Unlocks (hard) selected sectors</li> <li>-hardunlock Unlocks (hard) selected sectors</li> <li>-hardunlock Blank checks target</li> <li>-erasesectors Erases selected sectors</li> <li>-hardunlock Unlocks (hard) selected sectors</li> <li>-hardunlock Current fals chip</li> <li>-program Programs target</li> <li>-auto Erases, programs and verifies target</li> <li>-yerify Verifies target program</li> <li>-readrange Reads selected sectors</li> <li>-readrange Reads specified range of target memory Syntax: -readra</li></ul>	(j)	Valid command line	options:
-saveprjas Saves current project in a different file Syntax: -saveprjas <filename> -saveprj Saves current project Syntax: -saveprj -open Opens a data file Syntax: -open<filename>[,<saddr>] -saveas Saves current data file in a different file Syntax: -saveas<filename>[,<saddr>,<eaddr>] -saveas Saves current data file Syntax: -save[<saddr>,<eaddr>,<eaddr>] -save Saves current data file Syntax: -save[<saddr>,<eaddr>] -save Caleas data by given offset Syntax: -relocate <offset> -delrange Deletes data range Syntax: -delrange <saddr>,<eaddr> -eliminate Eliminates blank areas in data file -connect Connects to target -disconnect Disconnects from target -softlock Locks (soft) selected sectors -hardlock Unlocks (hard) selected sectors -hardlock Discons target -erasesectors Erases selected sectors -hardlock Unlocks (hard) selected sectors -theckblank Blank checks target -erasesectors Erases selected sectors -theckblank Blank checks target -program Programs target -auto Erases, programs and verifies target -yerify Verifies target program -readsectors Reads selected sectors -readchip Reads the entire flash chip -readrange Reads specified range of target memory Syntax: -readrange <saddr>, <eaddr> -stat arget application -exit Terminates application automatically -help Displays this box</eaddr></saddr></eaddr></saddr></offset></eaddr></saddr></eaddr></eaddr></saddr></eaddr></saddr></filename></saddr></filename></filename>	$\checkmark$	-openprj	Opens an existing project Syntax: -openpri <filename></filename>
-saveprj Saves current project Syntax: -saveprj -open Opens a data file Syntax: -open <filename>[, <saddr>] -saveas Saves current data file in a different file Syntax: -saveas <filename>[, <saddr>, <eaddr>] -save Saves current data file Syntax: -save[<saddr>, <eaddr>] -relocate Relocates data by given offset Syntax: -relocate <offset> -delrange Deletes data range Syntax: -delrange SADDR&gt;, <eaddr> -eliminate Eliminates blank areas in data file -connect Connects to target -disconnect Disconnects from target -softlock Locks (soft) selected sectors -hardlock Locks (hard) selected sectors -hardunlock Unlocks (soft) selected sectors -hardunlock Unlocks (hard) selected sectors -hardunlock Blank checks target -eraseschip Erases elleted sectors -hardunlock Unlocks (hard) selected sectors -reasectors Erases selected sectors -reasectors Reads selected sectors -readentip Reads the entire flash chip -readrange Reads selected sectors -readrange Shot entire flash chip -readrange Shot entire flash chip -readrange Reads specified range of target memory Syntax: -readrange -/exantple Starts target application -exit Terminates application automatically -help Displays this box</eaddr></offset></eaddr></saddr></eaddr></saddr></filename></saddr></filename>		-saveprjas	Saves current project in a different file Syntax: -saveprjas <filename></filename>
-openOpens a data file Syntax: -open <filename>[, <saddr>]-saveasSaves current data file in a different file Syntax: -saveas <filename>[, <saddr>, <eaddr>]-saveSaves current data file Syntax: -save[<saddr>, <eaddr>]-saveSaves current data file Syntax: -save[<saddr>, <eaddr>]-relocateRelocates data by given offset Syntax: -relocate <offset>-delrangeDeletes data range Syntax: -delrange <saddr>, <eaddr>-eliminateEliminates blank areas in data file -connect-connectConnects from target-disconnectDisconnects from target-softlockLocks (soft) selected sectors-hardlockLocks (hard) selected sectors-hardlockUnlocks (hard) selected sectors-checkblankBlank checks target-programPrograms and verifies target-programPrograms and verifies target-programPrograms and verifies target-verifyVerifies target program-readsectorsReads selected sectors-readrangeReads the entire flash chip-readrangeReads specified range of target memory Syntax: -readrange <saddr>, <eaddr>-startappStarts target application-exitTerminates application-exitTerminates application</eaddr></saddr></eaddr></saddr></offset></eaddr></saddr></eaddr></saddr></eaddr></saddr></filename></saddr></filename>		-saveprj	Saves current project Syntax: -saveprj
-saveasSaves current data file in a different file Syntax: -saveaSaves current data file Syntax: -savea-saveSaves current data file Syntax: -save[ <saddr>, <eaddr>]-relocateRelocates data by given offset Syntax: -relocate <offset>-delrangeDeletes data range Syntax: -delrange-eliminateEliminates blank areas in data file-connectConnects to target-disconnectDisconnects from target-softlockLocks (soft) selected sectors-softlockUnlocks (soft) selected sectors-hardlockUnlocks (hard) selected sectors-hardunlockUnlocks target-erasesectorsErases selected sectors-erasesectorsErases selected sectors-eraseschipErases, programs and verifies target-programPrograms target-autoErases, programs and verifies target-verifyVerifies target program-readsectorsReads selected sectors-readrangeReads the entire flash chip-readrangeReads the get range of target memory Syntax: -readrange of target memory Syntax: -readrange of target application-startappStarts target application-readrangeStarts target application-readrangeStarts target application</offset></eaddr></saddr>		-open	Opens a data file Syntax: -open <filename>[,<saddr>]</saddr></filename>
-save Saves current data file Syntax: -save[ <saddr>, <eaddr>] -relocate Relocates data by given offset Syntax: -relocate <offset> -delrange Deletes data range Syntax: -delrange <saddr>, <eaddr> -eliminate Eliminates blank areas in data file -connect Connects to target -disconnect Disconnects from target -softlock Locks (soft) selected sectors -softunlock Unlocks (soft) selected sectors -hardlock Locks (hard) selected sectors -hardunlock Unlocks (hard) selected sectors -hardunlock Unlocks (hard) selected sectors -hardunlock Erases selected sectors -checkblank Blank checks target -erasesectors Erases selected sectors -erasechip Erases entire flash chip -program Programs and verifies target -verify Verifies target program -readsectors Reads selected sectors -readchip Reads the entire flash chip -readrange Reads selected sectors -readrange of target memory Syntax: -readrange of target memory Syntax: -r</eaddr></saddr></offset></eaddr></saddr>		-saveas	Saves current data file in a different file Syntax: -saveas <filename>[,<saddr>,<eaddr>]</eaddr></saddr></filename>
-relocateRelocates data by given offset Syntax: -relocate <offset>-delrangeDeletes data range Syntax: -delrange <saddr>, <eaddr>-eliminateEliminates blank areas in data file-connectConnects to target-disconnectDisconnects from target-softlockLocks (soft) selected sectors-softlockUnlocks (soft) selected sectors-hardunlockUnlocks (hard) selected sectors-hardunlockUnlocks target-erasesectorsErases selected sectors-erasesectorsErases selected sectors-programPrograms and verifies target-programPrograms target-autoErases, program and verifies target-verifyVerifies target program-readsectorsReads selected sectors-readsectorsReads selected sectors-startappStarts target application-readrangeSapcified range of target memory Syntax: -readrange <saddr>, <eaddr>-startappStarts target application automatically -help-biplays this box-</eaddr></saddr></eaddr></saddr></offset>		-save	Saves current data file Syntax: -save[ <saddr>,<eaddr>]</eaddr></saddr>
-delrangeDeletes data range Syntax: -delrange <saddr>, <eaddr>-eliminateEliminates blank areas in data file-connectConnects to target-disconnectDisconnects from target-softlockLocks (soft) selected sectors-softunlockUnlocks (soft) selected sectors-hardlockLocks (hard) selected sectors-hardunlockUnlocks (hard) selected sectors-hardunlockUnlocks (hard) selected sectors-hardunlockErases selected sectors-erasesectorsErases selected sectors-erasesectorsErases selected sectors-erasesectorsErases number of target-programPrograms and verifies target-programPrograms target-autoErases, programs and verifies target-verifyVerifies target program-readsectorsReads the entire flash chip-readrangeReads specified range of target memory Syntax: -readrange <saddr>, <eaddr>-startappStarts target application-exitTerminates application automatically-helpDisplays this box</eaddr></saddr></eaddr></saddr>		-relocate	Relocates data by given offset Syntax: -relocate <offset></offset>
-eliminateEliminates blank areas in data file-connectConnects to target-disconnectDisconnects from target-softlockLocks (soft) selected sectors-softunlockUnlocks (soft) selected sectors-hardlockLocks (hard) selected sectors-hardunlockUnlocks (hard) selected sectors-hardunlockUnlocks (hard) selected sectors-checkblankBlank checks target-erasesectorsErases selected sectors-erasechipErases entire flash chip-programverifyPrograms and verifies target-programPrograms target-autoErases, programs and verifies target-verifyVerifies target program-readsectorsReads the entire flash chip-readrangeReads specified range of target memory Syntax: -readrange <saddr>, <eaddr>-startappStarts target application-exitTerminates application automatically-helpDisplays this box</eaddr></saddr>		-delrange	Deletes data range Syntax: -delrange <saddr>,<eaddr></eaddr></saddr>
-connectConnects to target-disconnectDisconnects from target-softlockLocks (soft) selected sectors-softunlockUnlocks (soft) selected sectors-hardlockLocks (hard) selected sectors-hardunlockUnlocks (hard) selected sectors-hardunlockUnlocks (hard) selected sectors-checkblankBlank checks target-erasesectorsErases selected sectors-erasechipErases entire flash chip-programverifyPrograms and verifies target-programPrograms target-autoErases, program and verifies target-verifyVerifies target program-readsectorsReads selected sectors-readsectorsReads the entire flash chip-readrangeReads specified range of target memory Syntax: -readrange <saddr>, <eaddr>-startappStarts target application-exitTerminates application automatically-helpDisplays this box</eaddr></saddr>		-eliminate	Eliminates blank areas in data file
-disconnectDisconnects from target-softlockLocks (soft) selected sectors-softunlockUnlocks (soft) selected sectors-hardlockLocks (hard) selected sectors-hardunlockUnlocks (hard) selected sectors-hardunlockUnlocks (hard) selected sectors-checkblankBlank checks target-erasesectorsErases selected sectors-erasechipErases entire flash chip-programverifyPrograms and verifies target-programPrograms target-autoErases, programs and verifies target-verifyVerifies target program-readsectorsReads selected sectors-readrangeReads the entire flash chip-readrangeReads specified range of target memory Syntax: -readrange <saddr>, <eaddr>-startappStarts target application-exitTerminates application automatically -help-helpDisplays this box</eaddr></saddr>		-connect	Connects to target
-softlockLocks (soft) selected sectors-softunlockUnlocks (soft) selected sectors-hardlockLocks (hard) selected sectors-hardunlockUnlocks (hard) selected sectors-checkblankBlank checks target-erasesectorsErases selected sectors-erasesectorsErases entire flash chip-programverifyPrograms and verifies target-autoErases, programs and verifies target-verifyVerifies target program-readsectorsReads selected sectors-readsectorsReads selected sectors-readsectorsReads the entire flash chip-readrangeReads specified range of target memory Syntax: -readrange <saddr>, <eaddr>-startappStarts target application-exitTerminates application automatically -help-helpDisplays this box</eaddr></saddr>		-disconnect	Disconnects from target
-softunlockUnlocks (soft) selected sectors-hardlockLocks (hard) selected sectors-hardunlockUnlocks (hard) selected sectors-checkblankBlank checks target-erasesectorsErases selected sectors-erasesectorpErases selected sectors-erasechipErases entire flash chip-programverifyPrograms and verifies target-autoErases, programs and verifies target-verifyVerifies target program-readsectorsReads selected sectors-readsectorsReads the entire flash chip-readrangeReads specified range of target memory Syntax: -readrange <saddr>, <eaddr>-startappStarts target application-exitTerminates application automatically-helpDisplays this box</eaddr></saddr>		-softlock	Locks (soft) selected sectors
-hardlockLocks (hard) selected sectors-hardunlockUnlocks (hard) selected sectors-checkblankBlank checks target-erasesectorsErases selected sectors-erasechipErases entire flash chip-programverifyPrograms and verifies target-autoErases, programs and verifies target-verifyVerifies target program-readsectorsReads selected sectors-readchipReads the entire flash chip-readrangeReads specified range of target memory Syntax: -readrange<-startappStarts target application-exitTerminates application automatically -help-helpDisplays this box		-softunlock	Unlocks (soft) selected sectors
-hardunlockUnlocks (hard) selected sectors-checkblankBlank checks target-erasesectorsErases selected sectors-erasechipErases entire flash chip-programverifyPrograms and verifies target-programPrograms target-autoErases, programs and verifies target-verifyVerifies target program-readsectorsReads selected sectors-readrangeReads the entire flash chip-readrangeStarts target application-exitTerminates application automatically-helpDisplays this box		-hardlock	Locks (hard) selected sectors
-checkblankBlank checks target-erasesectorsErases selected sectors-erasechipErases entire flash chip-programverifyPrograms and verifies target-programPrograms target-autoErases, programs and verifies target-verifyVerifies target program-readsectorsReads selected sectors-readrangeReads the entire flash chip-readrangeStarts target application-exitTerminates application automatically-helpDisplays this box		-hardunlock	Unlocks (hard) selected sectors
-erasesectorsErases selected sectors-erasechipErases entire flash chip-programverifyPrograms and verifies target-programPrograms target-autoErases, programs and verifies target-verifyVerifies target program-readsectorsReads selected sectors-readrangeReads the entire flash chip-readrangeReads specified range of target memory Syntax: -readrange-startappStarts target application-exitTerminates application automatically -help-helpDisplays this box		-checkblank	Blank checks target
-erasechipErases entire flash chip-programverifyPrograms and verifies target-programPrograms target-autoErases, programs and verifies target-verifyVerifies target program-readsectorsReads selected sectors-readrangeReads the entire flash chip-readrangeReads specified range of target memory Syntax: -readrange-startappStarts target application-exitTerminates application automatically -help-helpDisplays this box		-erasesectors	Erases selected sectors
-programverifyPrograms and verifies target-programPrograms target-autoErases, programs and verifies target-verifyVerifies target program-readsectorsReads selected sectors-readchipReads the entire flash chip-readrangeReads specified range of target memory Syntax: -readrange-startappStarts target application-exitTerminates application automatically -help-helpDisplays this box		-erasechip	Erases entire flash chip
-programPrograms target-autoErases, programs and verifies target-verifyVerifies target program-readsectorsReads selected sectors-readchipReads the entire flash chip-readrangeReads specified range of target memory Syntax: -readrange<-startappStarts target application-exitTerminates application automatically -help-helpDisplays this box		-programverify	Programs and verifies target
-auto Erases, programs and verifies target -verify Verifies target program -readsectors Reads selected sectors -readchip Reads the entire flash chip -readrange Reads specified range of target memory Syntax: -readrange <saddr>,<eaddr> -startapp Starts target application -exit Terminates application automatically -help Displays this box</eaddr></saddr>		-program	Programs target
-verify Verifies target program -readsectors Reads selected sectors -readchip Reads the entire flash chip -readrange Reads specified range of target memory Syntax: -readrange <saddr>,<eaddr> -startapp Starts target application -exit Terminates application automatically -help Displays this box</eaddr></saddr>		-auto	Erases, programs and verifies target
-readsectors Reads selected sectors -readchip Reads the entire flash chip -readrange Reads specified range of target memory Syntax: -readrange <saddr>,<eaddr> -startapp Starts target application -exit Terminates application automatically -help Displays this box</eaddr></saddr>		-verify	Verifies target program
-readchip Reads the entire flash chip -readrange Reads specified range of target memory Syntax: -readrange <saddr>,<eaddr> -startapp Starts target application -exit Terminates application automatically -help Displays this box</eaddr></saddr>		-readsectors	Reads selected sectors
-readrange Reads specified range of target memory Syntax: -readrange <saddr>,<eaddr> -startapp Starts target application -exit Terminates application automatically -help Displays this box</eaddr></saddr>		-readchip	Reads the entire flash chip
-startapp Starts target application -exit Terminates application automatically -help Displays this box		-readrange	Reads specified range of target memory Syntax: -readrange <saddr>,<eaddr></eaddr></saddr>
-exit Terminates application automatically -help Displays this box		-startapp	Starts target application
-help Displays this box		-exit	Terminates application automatically
		-help	Displays this box
-? Displays this box			
确定			備定

J-Link 用户手册 Rev 3.0 - 48 - 下面我们以 AT91SAM7S64 为目标芯片,来进行命令行演示。

进入命令行状态前,我们先把 AT91SAM7S64.JFLASH 工程文件 和 KEIL\_MOUSE.BIN 文件放到 C 盘根目录下,方便操作。然后连接 好目标板和 J-LINK。

首先进入到安装目录:



### 然后键入命令,如下图所示:



回车后,J-FLASH ARM 的用户界面会被弹出,然后可以看到 J-FLASH ARM 很快完成操作并退出,如果我们要看整个操作过程的 log 信息,我们可以去掉命令行的-exit 参数,去掉这个参数后, J-FLASH ARM 在完成操作后并不会被关闭,这个时候我们可以通过 log 窗口看到操作信息:

👷 J-Flash ARM V3.70a - [c:\AT91SAM7S64.jflash]																							
File Edit View	ile Edit View Target Options Window Help																						
The Fair New	in En Teil Take Neur Then The																						
Project - A	T9 💶 🗆 🗙		nous	e bi	n																		x
Name	Value	HINSH CTUTCT					- 4																
Connection	USB [Device 0]	Address: 0x1	00000	)	×1	x2	×4																
1.5.1740	00111	Address	Ø	1	2	3	4	5	6	7	8	9	A	в	С	D	E	F	ASC				
Init JTAG speed	30 KHZ 4000 LU-	100000	18	FØ	98	E5	18	FØ	97	E5	18	FØ	97	E2	18	FØ	97	E5					
TAB speed		100000	10	TO	0.0	E	00	00		E4	10	DD.	112	E	10	DD	10	E					_
IB len	<not used=""></not>	100010	10	гө	71	БЭ	99	60	но	EI	20	гг	IF	БЭ	20	FF	11	БЭ					
		100020	40	ИN	10	NN	C4	17	10	ии	сø	17	10	ии	BC	17	10	ИИ	e.,			• • • • •	
MCU	Atmel AT91SAM7S64	100030	B8	17	10	00	00	00	00	00	00	00	00	00	00	00	00	00					
Clock speed	47923200 Hz	100040	B4	00	9F	E5	B4	10	9F	E5	08	10	80	E5	FF	00	EØ	E3					
Endian	Little	100050	AC	10	9F	E5	60	10	80	E5	A8	00	9F	E5	02	19	AØ	E3		· · ·			
Check core Id	Yes	100060	04	10	80	F5	60	ØØ	9F	E2	60	10	9F	E2	20	10	80	E2					
Core Id	0x3F0F0F0F	100000	6	10	00	E	04	200	10	E9	DC DC	DD	DD.	80	00	10	00	E					
Use target HAM	Yes 0-200000	100070	00	20	70	БЭ	10	20	12	EZ	ru at	гг	гг	ы	70	10	71	БЭ	л.				
RAM address	1C / D	100080	20	10	80	E5	68	20	90	E5	04	20	12	EZ	FC	FF	FF	ØА		h .			
NAM SIZE	TO ND	100090	07	10	AØ	E3	30	10	80	E5	78	00	9F	E5	DB	FØ	21	E3		0	.×	† .	
Flash memory	AT91SAM7S64 internal	1000A0	00	DØ	AØ	E1	04	00	40	E2	D7	FØ	21	E3	00	DØ	AØ	E1		e			
Manufacturer	Atmel	1000B0	04	00	40	E2	D1	FØ	21	E3	00	DØ	AØ	<b>E1</b>	04	00	40	E2	0			e.	
Size	64 KB	100000	D2	FØ	21	F3	ØØ	ъø	60	F1	80	ØØ	40	F2	DЗ	FØ	21	E3			e		
Flash Id	0x0	100000	00	Ба	~	E4	04	00	40	E2	10	го	10	E2	00	Ба	~	E4					
Check flash Id	No	1000000	00	שט	нө	EI	94	99	40	E2	10	гө	21	EJ	99	00	нө	EI					
Base address	0x100000	1000E0	34	ии	9F	E5	61	ИN	10	E3	30	EЮ	9F	69	30	EЮ	9F	15	4		.0		
Organization	32 bits x 1 chip	1000F0	10	FF	2F	E1	FE	FF	FF	EA	FE	E7	CØ	46	00	FD	FF	FF	/	′	· · · · l	F	
		100100	01	04	00	A5	00	01	32	00	40	FD	FF	FF	00	FC	FF	FF		2	.e		
		100110	01	06	00	00	05	10	19	10	90	04	20	00	25	17	10	00					_
		100120	F4	00	10	00	F9	00	10	00	02	00	00	00	40	00	00	00				.e	<b>_</b>
J. LOG																					_		×
- Project open	ed successfully																						
Opening data fi	le [c:\keil_mouse.bin]	· · ·																					
- Data file op	- Data file opened successfully (6088 bytes, 1 range, CRC = 0x552C494F)																						
Auto programmin - Connecting	uto programming target (6088 bytes, 1 range)																						
- Connected	- Connected successfully																						
- Programming	target (6088 bytes, 1 r	ange)																					
- Target pr - Verifying CB	ogrammed successfully																						
- CRC of af	fected sectors verified	 successfully	(CR	2 = 0	xE89	8B400	))																=
- Target erase	d, programmed and verif	ied successfu	lly	- Com	plet	ed a:	Eter	1.31	2 sec	2													
																							~
																						3	) - 3
Ready												Сог	necte	d	0	lore I	d: 0x	3F0F0	FOF	Sp	beed: 4	000 kHz	

### Log 窗口内的主要信息是:

Opening project file [c:\AT91SAM7S64.jflash] ...

- Project opened successfully
- Opening data file [c:\keil\_mouse.bin] ...
- Data file opened successfully (6088 bytes, 1 range, CRC = 0x552C494F) Auto programming target (6088 bytes, 1 range) ...

- Connecting ...

- Connected successfully
- Programming target (6088 bytes, 1 range) ...

- Target programmed successfully

- Verifying CRC of affected sectors ...

- CRC of affected sectors verified successfully (CRC = 0xE898B400)

- Target erased, programmed and verified successfully - Completed after 1.312 sec

可以看到-openprj命令就是打开工程文件,即FLASH 编程算法; -open 是打开数据文件,即需要写入的 bin 或者 hex 文件,需要注意 的是-open 参数后面还需要添加烧写地址,即上述命令里面的 ",0x100000",不然会编程失败;-auto 是指自动操作,包含了擦除, 编程,校验几个步骤。如果只需要读,擦除,编程等一个单独的操作, J-FLASH ARM 也提供了相应的命令参数,可以自行尝试。在尝试阶 段,建议不要加-exit 命令,方便查看 log 窗口的信息,以确认操作是 否成功完成。

既然提供了命令行方式,我们就可以使用批处理命令来使得操作 更为简单:

新建一个文本文件,然后键入以下内容:

"cd c:\program files\segger\jlinkarm\_v370a jflasharm.exe-openprjc:\AT91SAM7S64.jflash -openc:\keil\_mouse.bin,0x100000 -auto "

然后另存为 bat 文件, 如 jflash.bat。

然后运行该 bat 文件,可以获得和前面命令行一样的效果。

更进一步的,我们来挖掘一下 bat 的批处理功能,新建一个 bat 文件,键入以下内容:

J-Link 用户手册 Rev 3.0 - 51 -

"cd c:\program files\segger\jlinkarm\_v370a

jflasharm.exe -openprjc:\AT91SAM7S64.jflash -openc:\keil\_mouse.bin,0x100000 -auto -exit

jflasharm.exe -openprjc:\AT91SAM7S64.jflash -openc:\keil\_memory.bin,0x100000 -auto"

然后运行 bat 文件,可以看到 J-FLASH 运行了两次,分别把

keil\_mouse.bin 和 keil\_memory.bin 写入到了 AT91SAM7S64 里面:

📲 J-Flash ARM V3.70a - [c:\AT91SAM7S64.jflash *]													
Elle Edit Yew Target Options Window Help													
(													
<b>•</b>													
'L													
·····e.													
.e													
IØ													
P													
· · · · · · · · · · · · · · · · · · ·													
· · · · · · · 📃 💾													
e 🔟 🔤													
Auto programming target (36864 bytes, 1 range) - Connecting - Connected successfully - Programming target (36864 bytes, 1 range) - Target programmed successfully - Verifying CRC of affected sectors - CRC of affected sectors verified successfully (CRC = 0x7CA17320) - Target reased, programmed and verified successfully - Completed after 2.969 sec WARNING: Supply voltage too low, disconnecting target! Disconnecting - Disconnected													

利用 bat 的特性我们可以用来完成一些特殊用途。比如对于具备 片内 FLASH,同时又开放总线的 ARM 芯片,比如 STR710。我们可 以先新建两个工程,分别针对片内 FLASH 和片外 FLASH,然后建立 bat 文件,分别打开两个工程,编程两段 FLASH。这样可以有效提高 工作效率。

J-Link 用户手册 Rev 3.0 - 52 -

更进一步,我们可以在 keil 下也加入这个功能,由于 keil 开放了 一个外部 FLASH 编程工具接口,使得使用 bat 文件成为可能,打开 Keil 的"Options for Target"选项,选择"Utilities"选项卡,把默认 的"Use Target Driver for Flash Programming"换成"Use External Tool for Flash Programming",然后在"Command"一栏选择之前设定好的 bat 文件,点击确认。

Options for Target 'SAM7S Flash - ULINK'											
Device Target Output Listing User C/C++ Asm Linker Debug Utilities											
Configure Flash Menu Command											
C Use Target Driver for Flash Programming											
ULINK ARM Debugger 💌 Settings 🔽 Update Target before Debuggi											
Init File: Edit											
Use External Tool for Flash Programming											
Command: U: \Documents and Settings\mcuzone\臬团\jflash.bat											
Arguments:											
🔽 Run Independent											
确定 取消 Defaults 帮助											

完成以上设置后,点击 Keil 工具栏上的 Load 按钮:

Y	🛛 ВІ	linky	/ - 1	lit i	sion 3										
	<u>F</u> ile	<u>E</u> dit	<u>V</u> ie	w	Project	De	bug	Fl <u>a</u> sh	Per	ipherals	; <u>T</u> ools	<u>s</u> vcs	<u>W</u> ind	ow į	<u>H</u> elp
	1	2	H	Ø	1 %	Đ	ß		$\square$	重	- <i>A</i>	% %	b 16		
	٩		**	X		*	SA	M7S	Flas	h - U	LINK		- 🐣	5	

马上会调入J-FLASH的编程界面,和直接运行bat文件一样效果。

J-Link 用户手册 Rev 3.0 - 53 -

更多的花样和参数可以按照实际需求进行变化和改进,利用批处 理的优势可以在调试和批量生产过程中极大的提升效率。

## J-Link 用户手册 Rev 3.0 - 54 -

### 附录三、V6版本 J-Link 的特点和 SWD 使用说明

V6 版本 J-Link 在硬件电路和软件方面都做了加强。硬件电路方面增加了 USB 保护,降低了仿真器功耗,拓展了接口电平支持范围;软件方面更有重大改进,主要是支持了最新的 SWD 接口,SWD 是ARM 公司新推出的一种调试接口,它仅需要 2 条线即可进行调试,与传统的 4 线 JTAG 相比可以有效减少调试占用的口线资源,有效提高少引脚芯片的口线利用率。目前 SWD 接口主要存在 Cortex-M3 内核的芯片上,如 ST 公司的 STM32 系列、Luminary 公司的 LM3S 系列。注意,只有 V6 版本的 J-Link 才支持 SWD!

目前 SWD 只有 4.42 版本 IAR 才开始支持, SWD 设置很简单, 如下图所示:

Options for node "A	T91SAM9261_leds_blink"	
Options for node "A Category: General Options C/C++ Compiler Assembler Custom Build Build Actions Linker Debugger Simulator Angel IAR ROM-monitor J-Link/J-Trace LMI FTDI Macraigor RDI Third-Party Driver	T91SAL9261_leds_blink" Factory Se tup Connection Breakpoints Communication • USB Device 0 • • TCP/IP asa. bbb. ccc. ddd Interface • JTAG scan chain • JTAG scan chain with multiple tas TAP number: 0 • SMD • SMD • Communication \$TOOLKIT_DIR\$\cspycomm.log	*ttings
	OK Cancel	

J-Link 用户手册 Rev 3.0 - 55 -

### 下面看看用 SWD 接口调试的 log 信息:

Tue Sep 18 19:48:18 2007: Loaded macro file: C:\Program Files\IAR Systems\Embedded Workbench ARM4.42 Evaluation\arm\config\flashloader\Luminary\ FlashLM3Sxxx2k.mac Tue Sep 18 19:48:18 2007: DLL version: V3.74g, compiled Aug 27 2007 18:51:17 Tue Sep 18 19:48:18 2007: Firmware: J-Link ARM V6 compiled Jun 14 2007 14:33:17 Tue Sep 18 19:48:18 2007: Selecting SWD as current target interface. Tue Sep 18 19:48:18 2007: JTAG speed is initially set to: 32 kHz Tue Sep 18 19:48:18 2007: Found SWD-DP with ID 0ba01477 Tue Sep 18 19:48:19 2007: TPIU fitted. Tue Sep 18 19:48:19 2007: FPUnit: 6 code (BP) slots and 2 literal slots Tue Sep 18 19:48:19 2007: Found SWD-DP with ID 0ba01477 Tue Sep 18 19:48:19 2007: TPIU fitted. Tue Sep 18 19:48:19 2007: FPUnit: 6 code (BP) slots and 2 literal slots Tue Sep 18 19:48:19 2007: Software reset was performed Tue Sep 18 19:48:19 2007: Initial reset was performed Tue Sep 18 19:48:19 2007: Turning off watchdog Tue Sep 18 19:48:19 2007: 1340 bytes downloaded and verified (2.04 Kbytes/sec) Tue Sep 18 19:48:19 2007: Loaded debugee: C:\Program Files\IAR Systems\Embedded Workbench ARM4.42 Evaluation\arm\config\flashloader\Luminary\ FlashLM3Sxxx2k.d79 Tue Sep 18 19:48:19 2007: Found SWD-DP with ID 0ba01477 Tue Sep 18 19:48:19 2007: TPIU fitted. Tue Sep 18 19:48:19 2007: FPUnit: 6 code (BP) slots and 2 literal slots Tue Sep 18 19:48:19 2007: Software reset was performed Tue Sep 18 19:48:19 2007: Target reset Tue Sep 18 19:48:20 2007: Program exit reached. Tue Sep 18 19:48:21 2007: 638 bytes downloaded into FLASH and verified (0.43 Kbytes/sec) Tue Sep 18 19:48:21 2007: Loaded debugee: C:\Program Files\IAR Systems\Embedded Workbench ARM4.42 Evaluation\arm\examples\Luminary\LM3S102-SK\ FlashingLight\Debug\Exe\FlashingLight.d79 Tue Sep 18 19:48:21 2007: Target reset 可以清楚看到 SWD 调试成功!

请多多访问 http://www.mcuzone.com , 以获取最近更新!

J-Link 用户手册 Rev 3.0 - 56 -



Powered by Team Mcuzone

QQ:8204136

Website: <u>www.mcuzone.com</u>

2007



www.mcuzone.com