# X-Link 用户指南

版本: Rev2.0

www.mcuzone.com

2006-10

# 版本更新说明

Rev2.0:

添加了 GDBserver 的使用指南作为附录一。

2006年10月。

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

Rev 2.0

- 2 -

## X-LINK 用户手册

#### 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)

Rev 2.0

- 3 -

- \* 带 USB 连接线和 20 芯扁平电缆
- \* 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、X-LINK 驱动安装

X-LINK 是本站开发的 JLINK 的兼容产品,具有一样的性能,但 是却只有十分之一的价格!

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:

🔏 License Agreeme	nt	×
License Agreeme	nt Please read the following license agreement. Use the scrol to view the rest of this agreement. Important - Read carefully: This license is a legal agreement between YOU (either an individual or a single entity) and SEGGER Microcontroller Systeme GmbH. By downloading and/or using J-Link ARM software you agree to be bound by the terms of this agreement. 1. LICENSE AGREEMENT In this agreement "Licensor" shall mean SEGGER Microcontroller Systeme GmbH (called SEGGER) except under the following circumstances: Do you accept all the terms of the preceding license agreement is the Microcontrol of the preceding license agreement.	I bar
	If so, click on the Yes push button. If you select No, Setup close.	WIII
	Yes No	

🔏 Choose Destinat	ion Location	×
	Setup will install J-Link ARM V3.20h in the following folder. To install into a different folder, click Browse, and select another folder. You can choose not to install J-Link ARM V3.20h by clicking Cancel to exit Setup. Destination Folder C:\\SEGGER\JLinkARM_V320h <u>Browse</u>	
	< <u>B</u> ack <u>Next&gt;</u> Cancel	







🔏 Installation Co	mplete	
	J-Link ARM V3.20h has been successfully installed. Press the Finish button to exit this installation.	
	< <u>Back</u> Einish≻ Cance	el 🔤



安装完成后, 请插入 JLINK/XLINK 硬件, 然后系统提示发现新硬件, 请选择手动指定驱动程序位置, 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 X-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: U5.30		
S/N :		
OEM : IAR		
Feature(s) :		
UTarget = 3.332U		
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>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 $\star$ 128kb into memory C 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 C 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 C 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 C 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:



可以看到 XLINK 和 JLINK 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 XLINK 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。

## 三、J-LINK (XLINK) 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:

Team	MCU	zone

levice   Targe	t   Output   Listing   C   As	m   LA Locate   LA	Misc Debug Vtilities
⊂ Use <u>S</u> imul □ Limit Spe	ator Set ed to Real-Time	tings 💽 💽 RDI	Interface Driver 💌 Settings
▼ Load Appl Initializati	ication at Sta 🔽 Run to ma on	n () 🔽 Load Appl Initializati	ication at Sta 🔽 Run to main() on
	Ed:	t	Edit
-Restore Deb	ug Session Settings	Restore Deb	oug Session Settings
🔽 Breakp	oints 🔽 Toolbox	🔽 Breakpo	oints 🔽 Toolbox
🔽 Watchp	oints & Pi	🔽 Watchpo	oints
🔽 Memory	Display	🔽 Memory	Display
CPU DLL:	Parameter:	Driver DLL:	Parameter:
SARM. DLL	-cAT91SAM7S	SARM. DLL	
Dialog DLL:	Parameter:	Dialog DLL:	Parameter:
DADMATS DIT	-p91SAM7S64	TARMATS. DLL	-p91SAM7S64

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

Browse for RDI Driver DLL		
C:\Program Files\SEGGER\JLinkA	RM_V320e\JLinkRDI. dll	
Debug		
Cache Options		
V Lache Lode	Carfimure BDT Designer	

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 RDI Configuration			? 🛛
General       Init       JTAG       Flash         Image: Second stress of the second stress o	Breakpoints CPU Log J-Link-RDI is an RDI compliant sof ARM. It requires a license (RDI), w obtained from SEGGER (www.seg This software is also capable of pro- flash memory of several ARM micro used to download your program to the add. license "FlashDL") and to number of software breakpoints in the add. license "FlashBP"). Connection to J-Link © USB Device 0 ♥ COP/IP	tware for J-Link which can be iger.com). ogramming the os, which can be flash (Requires set an unlimited flash (Requires <u>About</u> <u>License</u> <u>Reset Config</u>	
		确定	取消 应用 (&)

J-Link RDI Configu	ration	?
General Init JTAG JTAG speed Auto selection Adaptive glocking T000	Flash   Breakpoints   CPU   Log	
JTAG scan chain with	n multiple devices	
Position 0	JR len 0	
0 is closest to TDI.	Sum of IRLens of devices closer to TDI. IRLen of ARM chips is 4.	
	<u>V</u> erify JTAG config	
	<b>确定 取消</b> 应	用()

J-Link RD	DI Configuration 🛜	×
General I	Init   JTAG Flash   Breakpoints   CPU   Log	
Enab	ole flash programming	
Allows pr memory c	rogramming the flash. This is required to download a program into flash or to set software breakpoints in flash (flash breakpoints).	
<u>D</u> evice	Atmel AT91SAM7S64 Clock speed 48000000 Hz	
BAM	16 KB @ address 0x200000	
Flash	64 KB @ address 0x100000	
	I Flash is mirrored @ address 0x0	
E Ca	ache flash contents	
	s caching of flash contents. This avoids reading data twice and speeds up	
the tra	ansfer between debugger and target.	
	low flash <u>d</u> ownload	
Allows loader	s program download to flash. Your debugger does not need to have a flash r. This feature requires an additional license (FlashDL).	
E Sh		
1. 20		
	确定 取消 应用 (A)	

J-Link RDI Configuration	? 🔀
General   Init   JTAG   Flash Breakpoints   CPV   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!	
	应用(4)
WHILE AK (FI	

J-Link RDI Configuration			? 🔀
J-Link RDI Configuration         General Init JTAG Flash Breakpoints CPU Log         ✓ Allow instruction set simulation         Allows the emulator to simulate individual instructions when single steppin         This does not normally have any disadvantages and makes debugging mespecially when using flash breakpoints.         Reset strategy         J-Link supports different reset strategies. This is necessary because there way of resetting and halting an ARM core before it starts to execute instru         Hardware, halt after reset (normal) <ul> <li>Delay after reset</li> <li>The bardware BESET pin is used to reset the CPU After reset release.</li> </ul>	ing instructions. nuch faster, e is no single uctions.		? 🔀
The hardware RESET pin is used to reset the CPU. After reset release, J continuously tries to halt the CPU. This typically halts the CPU shortly after release; the CPU can in most systems execute some instructions before i The number of instructions executed depends primarily on the JTAG speehigher the JTAG speed, the faster the CPU can be halted. Some CPUs or be halted before executing any instruction, because the start of the CPU after reset release. If a pause has been specified, J-Link waits for the specified time before to the CPU. This can be useful if a bootloader which resides in flash or ROM be started after reset.	-Link erreset tis halted. ed: the an actually is delayed ying to halt 1 needs to		
	确定	取消	应用 ( <u>A</u> )

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

Team	MCU	Jzone

Options for	Target 'MCB2130 Flash'	×
Device   Target	:   Output   Listing   C   Asm   LA Locate   LA Misc   Debug Utilities	
Configure F	lash Menu Command	1
(• Use Targ	ret Driver for Flash Programming	
	ULINK ARM Debugger 💽 Settings 🔽 Update Target before Debuggi	
Init File:	ULLNK ARM Debugger RDI Interface Driver ULINK Cortex-M3 Debugger	
C Use Exte	rnal Tool for Flash Programming	
Command:	LPC210x_ISP.EXE	
Arguments:	"#H" ^X \$D COM1: 38400 1	
	🔽 Run Independent	
	确定 取消 Defaults 帮助	

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

Link Fl	ash Program	mer	-
1	Link Fla	-Link Flash Program	-Link Flash Programmer

选择"J-Link Flash Programmer"

Browse for RDI Driver DLL		
C:\Program Files\SEGGER\JLinkA	RM_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 <sup>第</sup>按钮进行直接下载。注意,该 功能只支持具备片内 FLASH 的 ARM7/9 芯片。

```
Rev 2.0
```

# 在 ADS 下使用设置:

## 点击"Add",选择 JLINKRDI.DLL:

ADP	1	C:\PROGRA~1\\Bin\Remote_A.dll C:\PROGRA~1\\Bin\ARMulata_dll	1.2.0.805	Add
J-Link	1	C:\\JLinkEDI.dll	3.20e	Kemove
Multi-ICE	1	C:\\Multi-ICE.dll	2.2.6.1346	Re <u>n</u> ame
				Save As
				Configur
Please target has to	select enviror be conf	a target environment from the above ument to the list. Note that a target figured at least once before it can b	list or add a environment e used.	
- nas co	De Com	ingured at reast once before it can b	e useu.	

点击"Configure",出现以下内容:

J-Link RD	DI Configuration	? 🛛
J-Link RD General I Allows pr memory of Device RAM Flash	DI Configuration         Enit       JTAG       Flash       Breakpoints       CPU       Log         De [lash programming]         rogramming the flash. This is required to download a program into flash or to set software breakpoints in flash (flash breakpoints).         Atmel AT91SAM7S64       Clock speed       48000000 Hz         16 KB @ address 0x200000       Clock speed       48000000 Hz         64 KB @ address 0x100000       Flash is mirrored @ address 0x0         ache flash contents       Clock speed       Clock speed	
Allows the tra Allows loader V Sh	s caching of flash contents. This avoids reading data twice and speeds up ansfer between debugger and target. low flash gownload s program download r. This feature requires an additional license (FlashDL). how info window during download	
		应用 (4)

进入 AXD 后的信息(注意 LOG FILE 的内容):

Ø AXD	
Zile Search Processor Views System Views Execute Options Mindow Help	
rre s i l' l' MM reve in q re	
Target Inage Fid F	
System Dutput Homitor	
RDI Log Debug Log	
Log file:	
J-Link RDI DLL V3.20e, compiled May 8 2006 09:35:21	
Firmware: J-Link compiled Apr 27 2006 12:55:19 ARM Rev.5	
Hardware: V5.00	
S/N ; DEM : IAB	
Feature(s) : RDI,FlashDL,FlashBP,JFlash	
VTarget = 3.326V Found 1 ITAG double Tabel/PLan = 4	
Id of device #1: 0x3F0F0F0F	
Found ARM with core Id 0x3F0F0F0F (ARM7)	10004
AHM RDI 1.5.1 -> ASYNC RDI Protocol Converter ADS v1.2 (Build number 805), Copyright (c) ARM Limite	a 2001.
<u> </u>	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	<u>Project Tools W</u> indow <u>H</u>	elp	
Vorkspace	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 Create <u>N</u> ew Project Add <u>E</u> xisting Project	Alt+F7	rs  an be read at requires to id jump. For
L	Source Code Control	•	exeption occu
	<u>M</u> ake <u>C</u> ompile	₽7 Ctrl+F7	In:
	Re <u>b</u> uild All C <u>l</u> ean		una
	Batch build	F8	SW:
	Debug	Ctrl+D	pał
	Make & <u>R</u> estart Debugger	•	dał
	irgvec:	В	rs
	figues:	В	IR

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 Libs Processor variant C Core ARM711 C Device Atmel	cary Configuration	Library options MI
---	--	--------------------	--------------------

Options for node 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         Output file       Output file       Secondary output file:         Output file       Override default       Secondary output file:         Output file       Override default       Secondary output file:         Output file       Override default       Secondary output file:         Output       Format       Override default       Secondary output file:         Format       Image: Control mod       Image: Control mod       Image: Control mod         Image: With I/O emulation modu       Image: Differed terminal output       Image: Control mod         Image: With I/O emulation modu       Image: Differed terminal output       Image: Control mod         Image: Output       Output       Image: Differed terminal output       Image: Differed terminal output         Image: Output       Image: Differed terminal output       Image: Differed terminal output       Image: Differed terminal output         Image: Output       Image: Differed terminal output       Image: Differed terminal output       Image: Differed terminal output         Image: Output       Image: Differed terminal output       Image: Differed terminal output       Image: Differed terminal output       Image: Differed terminal output         Image: Differed terminal output<
	Format variant: Arm compatible (-yas)

Options for node 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	Basic <sup>*</sup> Factory Settings Output Extra Output #define Diagnostics List Confite Generate extra output file Output file Output file Format Output format: simple-code Fogmat variant: None
	OK Cancel

	Factory Settin	gs
ieneral Options	Setup   Download   Extra Ontions   Plugins	
Assembler	· [bounded] and a operand [ reaging ]	
Custom Build	Driver $\square$ Run to	
Build Actions	J-Link/J-Trace	
Linker	Angel	
Debugger	IAR ROM-monitor	
Simulator	Macraigor	
Angel IAB BOM-monitor	RDI Third-Party Driver	
J-Link/J-Trace	\$FRUJ_DIR\$\resource\SAM7.mac	
Macraigor	Device depariation file	-34
RDI		-
Third-Party Driver		
A	\$TUULKIT_DIR\$\CUNFIG\ioat91sam7s64.ddf	
1		
	A <del></del>	

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

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 Download Extra Option Driver RDI Angel IAR ROM-monitor J-Link/J-Trace Macraigor RDI Third-Party Driver \$PRUJ_UIR\$\resource\SAM7 Devige description file Qverride default \$TOOLKIT_DIR\$\COMPIG\ioa	ons   Plugins   Run to main .mac t91sam7s64.ddf	Factory Settings
--	---	---	------------------

# 如果选择 "J-LINK/J-TRACE",则无需额外设置:

ategory: ieneral Options C/C++ Compiler	Setup Connection	Factory Settings
Assembler Custom Build	Reset	
Build Actions	C Hardware, halt at 0 (if sup	
Linker Debugger	© Software	
Simulator	🔽 Init sequence	
Angel IAB BOM-monitor		
J-Link/J-Trace	G Auto	eset <b>Г</b> <u>I</u> RQ
Macraigor RDI	Initia 32 kHz U	ndef 🔽 <u>F</u> IQ
Third-Party Driver	C Eixed 30 kHz 5	íL ata
***	C Adaptive	refetch
Ľ		

# 如果选择 "RDI",则还需要指定 JLINKRDI.DLL 的位置:

Category: General Options C/C++ Compiler Assembler	RDI	Factory Settings
Custom Build	C:\Program Files\SEGGER\TLi	PLARM V3206\TT iptRDT dll
Build Actions Linker Debugger Simulator Angel IAR ROM-monitor J-Link/J-Trace Macraigor RDI Third-Party Driver	☐ <u>A</u> llow hardware reset ☐ <u>E</u> TM trace ☐ Log RDI <u>c</u> ommunication	Note Use the RDI menu to specify additional driver settings. (This menu is available after the RDI driver to the RDI drite to the RDI driver to the RDI drit
	\$TOOLKIT_DIR\$\cspycomm.l.	og

# 设置完成后将多出一个 RDI 菜单,如下图:

🄏 IAR Embedded Vork	bench IDE	
<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>P</u> roject	<u>RDI T</u> ools <u>W</u> indow <u>H</u> elp	<u></u>
0 🛩 🖬 🕼 🐇 🛛	<u>C</u> onfigure	
Workspace	ETM Trace Window	12.C
Debug	Trace <u>S</u> etup	_task_id;
Files 8	Trace Sa <u>v</u> e	(* (RTOS_U.
	Breakpoint Usage /* If the ma following */ #define MAIN #define MAIN #endif	D == RTOS in_task li; defines a; TASK_ROM_: TASK_RAM_: /* (RTOS U,
H-⊞ III WEB2.C	/* int main(voi { #if (RTOS_US main task	d) ED == RTOS

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 XLINK 的时候, 注意不要再使用 IAR 自带的 FLASHLOADER 软件进行 FLASH 下载:

Category:		Factory Settings
C/C++ Compiler	Setup Download Extra Options Plug	zins
Assembler Custom Build Build Actions Linker Debugger Simulator Angel IAR ROM-monitor J-Link/J-Trace Macraigor RDI	<ul> <li>▲ttach to progr</li> <li>✓ Yerify download</li> <li>□ Suppress download</li> <li>□ Use flash loades</li> <li>,,,, (default), ATMEL</li> </ul>	Edit
Third-Party Driver	3	2

请将"Use flash loader"前的勾去掉,使用 XLINK 的 FLASH 编程算法,FLASH 编程速度可以达到 50K 以上,而如果使用 IAR 的 FLASHLOADER,则速度只能达到 10K 多一点!

四、J-FLASH ARM 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):

		Land and the second				~						-							
12101001		al, C: Abort	and the second	it a	mand	Ster	a te ha	4E # /	- eur	semi	: <b>\.</b> 42	m0 / 3	100.00	x t	c. b:				51
opperion	USB (Davice 0)	Address: Oct 1	10000	0	1×1	*2	×4												
	and the server of	Address	8	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ASCII
at JTAG speed	30 kHz 4000 kHz	1000000	ØF	69	98	ER	FE	FF	FF	EA	FE	PF	FF	EA	FE	FF	FF	EA	*****
AP number	(not used)	1000010	PE	FF	FF	EA	FE	FF	FF	EA	19	ØB	68	EA	86	98	AB	El	
i len	(not used)	1000020	84	81	98	ES	03	FØ	21	E3	ØE	50	20	E9	ØF	EØ	AB	Eİ	!P
	A CONTRACTOR OF A	1000030	18	FF	2F	E1	ØE	50	BD	E8	<b>D1</b>	FB	21	E3	89	88	AB	EI	/P!
hip	Afmel AT91SAM7X	1000040	84	FR	SE	F2	81	89	68	FR	96	RR	9F	FK.	RF	ER	AR	FI	•
ock speed	47923200 Hz	1000010	10	PP	90	174		00	0.0	10	84	THE	-	E.9	00	-	97	-	
ndian	Little	1000050	10	rr.	ar	ni			ne	E.a			41	E.a	00			20	
heck core to	Tes butchenene	1000060	02	FØ	21	E3	69	DH	AR	EI	68	86	48	EZ	13	FØ	21	123	
rank cone ad	Ver	1000070	86	DG	AB	E1	6C	98	9F	E5	ØP	EB	80	E1	10	FF	2F	E	1
oM address	0x200000	1000080	FE	FF	FF	EA	84	EØ	4E	E2	98	40	20	E9	86	EØ	4F	Eİ	N@0
AM size	16 KB	1000070	88	40	2D	E9	81	08	20	E9	44	EB	9F	ES	88	81	9E	ES	.8D
se DCC mode	Yes	1000000	88	E1	8E	ES	13	FH	21	E3	ØE	50	20	E9	BF	EØ	AR	EL	
		1008078	18	P.F	20	124	OF.	-	100		99	1768	21	E.5	28	E.B.			
lach memory	AT915AM7x256 int	1000000	10		ar		-	20	200	10	74			E.o	20	100		-	
anulachurer	Atmel	1996905	36	EI	8E	Eb	61	90	RD	ES	98	48	RD	EB	RF	FR	6F	ы	8
20	256 KB	1000000	86	80	FD	EB	FE	FF	FF	EA	FE	FF	FF	EA	FE	FF	FF	EA	
ash id	-DMO	10000E0	FØ	63	98	88	68	FØ	FF	FF	EC	88	68	00	86	68	618	EA	
neck nashi la	NO 9-400000	10000F0	F4	88	88	EA	28	08	8F	E2	88	BC	98	E8	88	88	BA	EB	
manitation	32 hits w1 chin	1000100	RI	78	40	F2	88	BH	88	FØ	RR	RP	50	F1	ED	88	88	RA	.nl
(Bourdeout	or promit comp	1998118	ar	-	BO	UN	14	ER	AF	F2	-	-	19	E2	87	EB	47	18	0 0
_		1000110	or	68	DI	1.0	14	EO	-11	E.6	81	96	10	Ea	63	ra	30	10	
100																			

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(L):	C ProjectFiles	- E 🕂 💷 -	
表 近 前 支 前 文 代 代 一 代 一 代 一 代 一 代 一 代 一 代 一 の 文 档 一 の 文 档 一 の 文 档 一 の 文 档 一 の 文 档 一 の 文 档 一 の う 代 の 文 档 の 、 の 文 代 の 、 の 文 代 の 、 の 文 代 の 、 の 、 の 、 の の 、 の 、 の の 、 の の 、 の の 、 の の 、 の の 、 の の 、 の の 、 の の 、 の の 、 の の 、 の の の の 、 の の の 、 の の 、 の の 、 の の 、 の の 、 の の 、 の の 、 の の 、 の の 、 の の 、 の の の の の の の 、 の	ADuC7020. jflash ADuC7030. jflash ADuC7032. jflash ADuC7032. jflash ADuC7229. jflash ADuC7229. jflash AT91FR40162. jflash AT91M55800A. jflash AT91M55800A. jflash AT91RM9200_CSB337. jflash AT91RM9200_CSB337. jflash AT91RM9200_CSB637. jflash AT91SAM7A1_EK. jflash AT91SAM7A3. jflash AT91SAM7S32. jflash AT91SAM7S64. jflash	m AT91SAM7S256.jflash mAC m AT91SAM7X128.jflash mNS9 m AT91SAM7X128.jflash mNS9 m AT91SAM7X256.jflash mNS9 m DragonballMX1.jflash mS7 m LH7A40x_LogicPD.jflash mS7 m LH79520_Log 类型: JFLASH 文件 m LH79520_Log 类型: JFLASH 文件 m LH79524_Log 大小: 8.09 KB m LPC2106.jfl m LPC2138.jflash mTMS m LPC2148.jflash mTMS m LPC2294.jflash mTMS	7111. 360. j 750. j 87750 445HX LiteP 710. j 912. j 912. j 470R1 470R1
1,412 ≤ 10,12	AI9ISAMISI28. jflash	C LFC2294_FhyCORE.jflash CIMS	4 TURI
	文件名 (M): <b>*. jflash</b> 文件类型 (I): J-Flash project	▼ 打开 . 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 settings	? 🗙
General   JTAG ARM   Flash   Production	1
Chig Atmet AT 91 SAM 7X256 Clock 47923200 Hz Endian Little AT 91 SAM 7X256 Clock 47923200 Hz Endian Little AT 91 SAM 7X256 Clock ABM core ID ID 3F0F0F0F Use target BAM (faster) Addr 200000 16 KB Enable DCC mode (faster) Use following init sequence:	
# Type Value0 Value1 Comment	
0Reset0 ms1Write 32bit0xFFFFFD440x00008000Disable watchdog2Write 32bit0xFFFFF600x00320300Set flash wait states3Write 32bit0xFFFFFC200x00000601Set PLL4Delay200 ms5Write 32bit0xFFFFFC2C0x00191C05Set PLL and divider6Delay200 ms7Write 32bit0xFFFFFC300x00000007Select master clock and	
Add Insert Delete Edit Down	
<b>确定 取消</b> 应用	A)

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		2
General   JTAG   ARM Flash   Pro	oduction	
FlashBank Bank[0]	Add Remove	
Base Addr 00100000 0rg	anization 32 💌 Bits x 1 🛒 Chip(s)	
☐ Use custom <u>B</u> AMCode		
Manufacturer Atmel Chip AT91SAM7X256 internal Size 256 KB Sectors	1024	
Start/End sector Start Addr Sector[0]: 0x0	<ul> <li>Individual sectors</li> <li>✓ Sector[0]: 0x0 - 0xFF</li> <li>✓ Sector[1]: 0x100 - 0x1FF</li> <li>✓ Sector[21: 0x200 - 0x2EF</li> </ul>	
Selected ranges: 1024 Sectors, 1 Range: 0x0000 - 0x3FFFF	<ul> <li>✓ Sector[3]: 0x300 · 0x3FF</li> <li>✓ Sector[4]: 0x400 · 0x4FF</li> <li>✓ Sector[5]: 0x500 · 0x5FF</li> <li>✓ Sector[6]: 0x600 · 0x6FF</li> <li>✓ Sector[6]: 0x700 · 0x7FF</li> </ul>	
	<u>All None Invert</u>	
<u>.</u>	确定	取消 应用 ( <u>k</u> )

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   Prod	luction
FlashBank Bank[0]	Add Remove
Base Addr 00000000 Organ	nization 16 💌 Bits x 1 💌 Chip(s)
Use custom <u>B</u> AMCode	
✓ Manual flash selection	Select flash <u>d</u> evice
Manufacturer	Check manufacturer flash Id
Size Sectors	Buswidth Id
C Start/End sector	Individual sectors
Start Addr	
End Addr	
.I	<u>All</u> <u>None</u> <u>Invert</u>
	确定取消应用 (A)

anufacturer 🛛	•						
Manufacturer	Device	Size	NumSectors	8bit Id	16bit Id	Buswidth	1
MD.	Am29DL161DB	2048 KB	39	10039	12239	16	
MD	Am29DL161DT	2048 KB	39	10036	12236	16	-
MD	Am29DL162DB	2048 KB	39	1002E	1222E	16	
MD	Am29DL162DT	2048 KB	39	1002D	1222D	16	
MD	Am29DL163DB	2048 KB	39	1002B	1222B	16	
MD	Am29DL163DT	2048 KB	39	10028	12228	16	
MD	Am29DL164DB	2048 KB	39	10035	12235	16	
MD	Am29DL164DT	2048 KB	39	10033	12233	16	
MD	Am29DL322DB/GB	4096 KB	71	10056	12256	16	
MD	Am29DL322DT/GT	4096 KB	71	10055	12255	16	
MD	Am29DL323DB/GB	4096 KB	71	10053	12253	16	
MD	Am29DL323DT/GT	4096 KB	71	10050	12250	16	
MD	Am29DL324DB/GB	4096 KB	71	1005F	1225F	16	
MD	Am29DL324DT/GT	4096 KB	71	1005C	1225C	16	
MD	Am29DL400BB	512 KB	14	1000F	1220F	16	
MD	Am29DL400BT	512 KB	14	1000C	1220C	16	
MD	Am29DL800BB	1024 KB	22	100CB	122CB	16	
MD	Am29DL800BT	1024 KB	22	1004A	1224A	16	
AMD	Am29DS323DB	4096 KB	71	100B8	122B8	16	-

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:

Team	MC	Uzone

Project settings	
General JTAG CPU Flash Production	
CPU       XScale       ▼         CMCU       ABM7/ARM9       ID         MCU       XScale       Use target <u>B</u> AM (faster)         Endian       Little       Addr	
Use following init sequence:	
# Type Value0 Value1 Comment	
Add Insert Delete Edit Down	
确定 取消 应用 (	v I

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!

### JLINK/XLINK F.A.Q

1Q: 国内代理商卖的 JLINK 价格是 1900,贵站兼容 JLINK 的名为 XL INK 的调试器的价格也在 1000 多,价格优势似乎不明显?

1A: 注意, JLINK 分很多版本, 国内代理商销售的 1900 的 JLINK 全 是 IAR 版本的, 即只能在 IAR 下使用, 而且功能有所限制, 速度也有 限制!本站可以按照客户需求进行定制, 最全版本的 XLINK 的价格不 超过 2000, 但是同样功能的原装 JLINK 的价格是 1000 欧元! 而且目 前国内仅 MCUZONE 一家可以提供全功能版本的 JLINK (XLINK), 市 面上在销售的其他 JLINK 全部都是 IAR 版本的, 即 SEGGER 为 IAR 做 的 0EM 产品! 同样, ATMEL 的 SAM-ICE 也是 SEGGER 做的 0EM 产品, 但是限制更多。本站也可以按照客户需求进行定制, IAR 版本的 XLI NK 的价格在 1000 以内。

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(Fla 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:

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!

- 39 -

3Q: JLINK (XLINK) 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 RDI License	
	Sorry, no valid license for RDI found. J-Link S/N is Please contact SEGGER Microcontroller (www.segger.com, sales@segger.com) to obtain a license.	
	Time limited trial versions are available.	
Ågd	iRDI	×
1	J-Link - RDI-Error 242: RDIError_SoftInitialiseE	rror

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 XLINK 的 FLASH 下载速度和用 MULTI-ICE 的下载速度差不多?

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 XLINK 的 FLASH 编程算法,关键一 点是将"USE FLASH LOADER"前的勾去掉,如下图:

- 42 -

Satur D	beo [awo	Rutus Ontisa	Plusies	Factory Settings
Image: A trace of the second seco	h to pro y downlo ess down lash loa fault), Al	gr ad Loax Lea MEL		Edit
	Setup D Attac Verif Suppr Use f	Setup Download Attach to prov Yerify downlow Suppress downlow Use flash load ,, (default), AT	Setup Download Extra Option: Attach to progr Verify download Suppress download Use flash loader (default), AIMEL	Setup Download Extra Options Plugins Attach to progr Verify download Suppress download Use flash loade; , (default), ATMEL

使用 FLASH LOADER 进行 FLASH 下载和使用 XLINK 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:

J. J-Lin	k GDB Server ¥3.48	b	
File Help	p		
Debugger J-Link Target	Waiting for connection Connected ARM7, Core Id: 0x3F0F0F0F	Initial JTAG speed 1000 v kHz Current JTAG speed 1000 kHz 3.31 V Little endian v	<ul> <li>✓ Stay on top</li> <li>✓ Log window</li> <li>✓ Log to file</li> <li>✓ Cache reads</li> </ul>
Log output	E.		<u>C</u> lear log
J-Link JLinkA J-Link Firmwa Listen	GDB Server V3.48b RM.dll V3.48b (DLL connected re: J-Link compile ing on TCP/IP port	compiled Oct 10 2006 11:40 d Sep 28 2006 10:05:22 ARM H 2331	. 57) Rev. 5
J-Link JTAG I	found 1 JTAG devi D: 0x3F0F0F0F (ARM	ce, Total IRLen = 4 7)	<u>×</u>
O KB downl	oaded	1 JTAG 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 arm-elf-insight,这是个图形化的 ARM GDB,如下图:



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

-	main.c - Source Window	1	
213	e Am Yiss Control Preferences Help		
	- [1] ( ) ( ) 🖗 🔉 🦓 📇 🚓 🛃 式 國 Find:	4	44
-	ain.c - [nain -	SOURCE	•
	<pre>25 //* Function Name : change_speed 26 //* Object : Adjust "LedSpeed" value depending on SW1 and SW2 are pressed en 27 //* Input Parameters : none 28 //* Output Parameters : Update of LedSpeed value. 29 //*</pre>	r net	
	<pre>42 //*- 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 //*- 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 54</pre>	2	-

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

	on	🦳 🔽 Set breakpoint at 'main'
Target:	GDBserver/TCP	▼ ⊑ Set breakpoint at 'exit'
Hostname:	localhost	E Set breakpoint at
Port:	2331	
		🗖 Display Download Dialog

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:

	reset			
50	//a- Exception vectors			
52 //*				
53	//*- These vectors can I	e read at ad	ress 0 or at BAH address	
54	//*- They ABSOLUTELY red	uires to be	n relative addresssing mode in order to	
55	//*- guarantee a valid	jump. For the	noment, all are just looping.	
56	//*- If an exception occ	curs before r	map, this would result in an infinite l	.cep.
57	//*- To ensure if a exe	tion occurs	eFore start application to infinite loo	ф.
58	//*			*/
59				
60	B	InitHese	/* BxDB Reset handler */	
61	undefvec:			
62	B	undefvec	/* 0x04 Undefined Instruction	n */
63	swivec:			
64	B	swivec	/∗ 0x08 Software Interrupt *	4
65	pabtvec:			
66	В	pabtwec		
67	dabtuec:	200000000000000000000000000000000000000	5 J-Link GDB Server V3.486	
68	B	dabtwec	ile Help	
69	rsudvec:			
78	B	rsuduec	Debugger Connected to 122 B.B.1	aned 1000 + kHz II Stay on top
71	iravec:	10.00.0000000		
72		IRO Han	J-Link Connected Current JTAB op	seed 1000 kHz = Lastalla
73	finuec:	and a state of the second		In roĝojie
74	/		Target ARM7, Halted 3.31 V	Min endian 📩 🗉 Çadra reada
75	//s- Function	: FIO Han		
76	//s- Treatments	: FIO Con	Log output:	Lieur log
77	//w- Called Eunctions	. ALC FUR	Connected to 127.0.0.1	
78	//*		Reading all registers	
70			Read 4 bytes @ address 0x00000000 (Data	a = DxEA000010)
88	FIR Handler Entru:			
81	. re_namerer _energy			
82	/- Switch in SUC/User 1	nde te alle		
90	/a because the EIR is no	at nat ackno		
	551233455555678906123345567855555678906123345567777777777777777777777777777777777	50 /*- Exception vectors 51 //*- Exception vectors 52 //*- They MESOLUTELY res 53 //*- They MESOLUTELY res 54 //*- If an exception ocs 56 //*- If an exception ocs 57 //*- To ensure if a exception 58 //*- To ensure if a exception 58 //*- To ensure if a exception 58 //*- To ensure if a exception 50 B 60 B 61 undefvec: 62 B 63 swivec: 64 B 65 pabtvec: 66 B 67 dabtvec: 68 B 69 rsudvec: 70 B 71 irquec: 72 J*- Function 76 //*- Treatments 77 //*- Called Functions 78 //*- 79 80 FIQ_Handler_Entry: 81 82 /*- Switch in SUC/User I 83 /* hecause the FIB is nu	50 /* Exception vectors 51 //*- Exception vectors 52 //*- They ABSOLUTELY requires to be i 53 //*- They ABSOLUTELY requires to be i 55 //*- guarantee a valid jump. For the 56 //*- If an exception occurs before re 57 //*- To ensure if a exeption occurs b 58 //*	<pre>50 /*</pre>

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。

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