

USB Open Link(多インタフェース搭載、高速 JTAG)マニュアル

株式会社日昇テクノロジー

http://www.csun.co.jp

info@csun.co.jp

2012/4/25



copyright@2012-2013

- ※ この文書の情報は、文書を改善するため、事前の通知なく変更され ることがあります。
- ※(株)日昇テクノロジーの書面による許可のない複製は、いかなる形 態においても厳重に禁じられています。



• 修正履歴

•

NO	バージョン	修正内容	修正日
1	Ver1.0	新規作成	2009/12/20
2	Ver1.1	OpenLink フォームウエア更新手順を追加	2010/11/15
3	Ver1.3	ARM9 ボードの書き込み手順を追加	2012/4/25



目次

一、USB Open Linkの概要	4
1.1 USB Open Linkの主な特徴	4
1.2 USB Open Link がサーポート ARM コア	5
1.3 USB Open Link が利用できるソフトウエア	6
二、USB Open LinkのUSB ドライバインストール	7
三、各開発統合環境に USB Open Linkの設定	12
3.1 Keil での設定	12
3.2 ADS での設定	21
3.3 IAR での設定	23
3.4 OpenOCD での設定	32
四、Open Linkで J-Flash ARM 使用方法	35
五、ARM9 ボード(Mini2440 シリーズ)書込み手順	41
六、OpenLink フォームウエア更新手順	44



一、USB Open Link の概要



USB Open Link はエミュレーション ARM コアチップをサポートするための高速 JTAG エミュレートです。SEGGER 社様の J-Link と似ています、サーポートする開発環 境は IAR EWARM、ADS、Keil、WINARM、RealView など主な統合開発環境です、すべ ての ARM7/ARM9 コアチップをサーポートしています。なお、RDI のインターフェイスを 介して、各開発環境とスムーズに統合できます。

- 1.1 USB Open Link の主な特徴
- 1) IAR EWARM 統合開発環境にスムーズに接続できる JTAG エミュレータ
- 2) すべての ARM7/ARM9 コアチップ及び Cortex-M3、Thumb モードをサポートしま す。
- 3) ADS、IAR、KEIL、WINARM、REALVIEW などのほとんど開発環境をすべ てサポートします。
- 4) 最大ダウンロード速度 ARM7:600KB/S、ARM9:550KB /S、DCC:800KB/S
- 5) 最大 JTAG 速度 12MHz
- 6) ターゲットボードの電圧範囲は 1.2V-5V
- 7) ダウンロードの速度を自動認識



- 8) すべての信号とターゲットボードの電圧を監視
- 9) 完全なプラグアンドプレイ
- 10) USB から給電(ターゲットボードに給電しない)
- 11) 数多くの JTAG インタフェース搭載(2mm、2.54mm 10 ピン、2.54mm 14 ピン、2mm、
 2.54mm 20 ピン)、異なるピンのインタフェースの間は直接変換可能
- 12) 五つの JTAG ケーブル付け(2.0mm、2.54mm 10 ピン、2.0mm、2.54mm 20 ピン,、 2.54mm 14 ピン)
- 13) マルチデバイスのシリアルと接続の JTAG をサーポート
- 13) TCP/IP サーバーを搭載、TCP/IP ネットワークで使用することができます

1.2 USB Open Link がサーポート ARM コア

Open Link has been tested with the following cores, but should work with any ARM7/ARM9 and Cortex-M3 core.

- * ARM7TDMI (Rev 1)
- * ARM7TDMI (Rev 3)
- * ARM7TDMI-S (Rev 4)
- * ARM720T
- * ARM920T
- * ARM922T
- * ARM926EJ-S
- * ARM946E-S
- * ARM966E-S
- * Cortex-M3
 - a) ARM7TDMI (Atmel AT91M40800 , AT91M55800, AT91M40162, AT91SAM7S64, AT91SAM7S256, Sumsung S3C4510B, S3C44B0X, TMS320VC5470, MSM5100, MSM5105)
 - b) ARM7TDMI-S (LPC2104, LPC2114, LPC2131, LPC2294)
 - c) ARM720T (Hynix HMS30C7202)
 - d) ARM920T (Motorola MC9328MX1, AT91RM9200, S3C2410)
 - e) ARM922T (KS8695)
 - f) ARM926E (Motorola MX21, S3C24A0, MSM6275)
 - g) ARM940T (Conexant CX82100, S3C2510)
 - h) ARM946E (Marvell 88E62)
 - i) XScale (PXA255, PXA262, PXA263, IXP425, IXP465)



- 1.3 USB Open Link が利用できるソフトウエア 1) J-Mem:メモリを参照及び修正
 - 2) J-Link Server:(TCP/IPを介しOpen Linkと接続)
 - 3) J-Flash: 独自の Flash プログラミングをサーポート、量産ソリューションとして扱われます。
 - 4) RDI Flash BP: RDI に基づき、Flash にブレークポイントを無制限設定できます。
 - 5) RDI Flash DLL: RDI に基づき、独自の Flash プログラミングをできます。
 - 6) GDB server:GDB環境でデバッグできます。



二、USB Open Link の USB ドライバインストール

ドライバインストール用のファイルは弊社ホーム下記 URL からダウンロードできます。

<u>http://www.dragonwake.com/download/open-link/Setup_OpenLinkARM.zip</u> SEGGER 社様のソフトウエアを利用しておりますので、直接 SEGGER 社様ホームページ から最新の USB ドライバもダウンロードできます。

http://www.segger.com/cms/jlink-software.html

インストールの際に、ダウンロードの ZIP ファイルを解凍し、ディフォルトのままで 行ってください。

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>Br</u> owse
< <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
	< Back Next > Cancel

🔏 Installation Cor	aplete	
	J-Link ARM V3.20h has been successfully installed. Press the Finish button to exit this installation.	
	< <u>B</u> ack Einish > Canc	el 🔤

インストールが完了後、Open LinkをパソコンとUSBで接続し、新しいハードウエアが 自動的に認識され、ドライバを自動的にインストールされます。自動にインストールされ ない場合、手動にドライバの場所(Open Linkソフトウエアインストール場所のDriverフォ ルダー)を指定しインストールしてください。

完了後、デスクトップ上に、下記二つアイコンがあります。





J-Link ARM は開発ボードに設定できます。例として、LPC2148モジュールを挙げます。





メモ: ARM7TDMI-Sコアの特徴のため、LPC2000シリーズのJTAG速度は最大まで クロックの1/6に達しています。一番早い速度は4.8Mとなり、JTAG速度は4.8Mを超 える場合、Open-LINK ARM/はLPC2000ボードが見つかれませんというメッセージが出 てくるかもしれません。これはLPC2000コアの制限です、Open Linkと関係がありま せん。



三、各開発統合環境に USB Open Link の設定

3.1 Keil での設定

1)Keil 起動後、あるプロジェクトを開く

RTC - IVision3 - [G:V01_worki	<pre>KcstWipt2146WjpWExampleWExampleWRTC_2146</pre>	WDrivers#PCD5544.c]	CHAT IN	
Ble Edit View Project Debi	ug Flash Peripherals Jools SVCS Window	Help	5265 V	X
8 # # Ø 1 % % S	C 使使为没没有啊		金同園	
● □ □ ≤ ¥ 茶 2148P	- 🛃 📾			
Project Workspace	163 -			-
🖂 Z2148P	164 D /* 165 PCD5544 Initialize			
B Startup s	167 void LCO_Init(void) [***************************************		
PCD5544.c	<pre>168 //PINSELO 4= 0xEFFFFFFF: 163 //PINSELO 1= 0x20000000;</pre>			
pcd5544.h	PINSELI I= 0x00000088;			
D pcd5544.b	173 100018 1= (1<20) (1<23) (1<4	8): // cs/bc/RsT		
	175 175			
	177 //SSPCR = 0x38: 178 SSP Init 0:			
	179 180 100SET=(1<<18);			
	181 182 1000LR=(1<<20);			
	183 100CLR=(1<<23); 184 LCD_Write(0x20[1);	1		
	105 LCO_Write(0x80+38); 186 LCO_Write(0x04+8);			
	187 LCD_Write(0x13): 188 LCD_Write(0x20);			
	199 LCD_Brite(0x0C); 190100SET=(1<<20);			
	191 Delay(1); 192 }			
	193 / / PCD5544 goto x.v	1		
	196 Void LCD_GoloXY(unsigned char x	, unsigned char y) (_
	197 1000LR = (1<<23); 198 1000LR=(1<<20);			
	200 LCD_Write(128+x);			
	202 Defay(1);			50
1.	2030 1			
	PCD5544c RTCc			
×[
Per la				
T West				
Build Command	Find in Files /		(c)	
			ULINK ARM Debugger	L:183 C:46

2)「Debug」タブで「RDI Interface Driver」を選択し、「Settings」をクリックする

evice∣large ⊂Use Simul ⊏Limit Spe	ator Setting ed to Real-Time	SS (* <u>Nse</u> :] RDI Interface Driver ▼ <u>Settin</u>
✔ Load Appl Initializati	ication at Sta 🔽 Run to main() on	✓ Load Application at Sta ✓ Run to main Initialization
-Restore De Breakp Watchp Memory	bug Session Settings oints 🔽 Toolbox oints & Pi Display	Restore Debug Session Settings Breakpoints IV Toolbox Watchpoints Memory Display
CPU DLL:	Parameter:	Driver DLL: Parameter:
SARM. DLL	-cAT91SAM7S	SARM. DLL
Dialog DLL:	Parameter:	Dialog DLL: Parameter:
DARMATS DLL.	-p91SAM7S64	TARMATS. DLL -p91SAM7S64



3)「…」をクリックして、Open LINK のインストールディレクトリに行く。

RDI Interface Driver Setup	and a part of the second secon	×
Browse for RDI Driver DLL		
G:\01_work\embeded\JLinkARM_V	/410f\JLinkRDI.dll	
Browse for ToolConf File		
Debug-		
Cache Uptions	Cariferina BDI Deirar	
Cache <u>M</u> emory		
	OK Cancel	<u>H</u> elp

4)「Configure RDI Driver」をクリックして、次のタブが表示されます、PCでデバッグの場合、USBが使えます。ローカルエリアネットワークのデバッグの場合、TCP/ IPを選択し、Open LINKをリンクしているPCのIPアドレスを指定します。

SEGGER J-Link RDI V4.10f C	onfiguration	-		? X
General Init JTAG Flash	Breakpoints CPU Log J-Link RDI is an RDI compliant soft ARM. It requires a license (RDI), wi obtained from SEGGER (www.seg) This software is also capable of pr flash memory of several ARM micro used to download your program to the add. license "FlashDL") and to number of software breakpoints in I the add. license "FlashDL") and to number of software breakpoints in I the add. license "FlashDL") Connection to J-Link © USB Device 0 ▼ © ICP/IP	tware for J-Link hich can be ger.com). agramming the ss, which can be flash (Requires set an unlimiter flash (Requires	e 1	
Location of config file	DK3.05\uv3\Uv3.ini	License Reset Config]	
		ОК	キャンセル	適用(A)



5) JTAGのスピードを設定する。- Sコアの場合はAutoをを使用することをお勧めする。非-Sコアの場合は、直接12Mの最大速度を使用することができます。使用中不安定な現象が 発生した場合は、JTAGのクロック速度を適切に低値に調整をお勧めする。

SEGGER J-Link RDI V4.	10f Configuration	? ×
General Init JTAG F JTAG speed C Auto selection C Adaptive glocking C 1000 Y kHz	lash Breakpoints CPU Log	
JTAG scan chain with Position O oris closest to TDD.	IR len 0 Sum of IRLens of devices closer to TDD. IRLen of ARM chips is 4.	
	⊻erifyJTAG config	
	OK キャンセル	



 フラッシュプログラミング機能:オンチップフラッシュの ARM チップの場合は、この機 能が使える。そうすると、デバッグする前に Open Link はフラッシュをプログラミング する。

SEGGER J-Link RDI V4.10f Configuration	8 ×
General Init JTAG Flash Breakpoints CPU Log Image: Construction of the state of the	
OK *	*シセル 適用(<u>A</u>)



 ソフトウェアブレークポイント機能:オンチップフラッシュのAMR チップの場合、この 機能の使用をお勧めする。デバッグを容易にするようにこの機能の利用をお勧めします。

SEGGER J-Link RDI V4.10f Configuration	8 ×
General Init JTAG Flash Breakpoints CPU Log	
Use software breakpoints	
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.	
Use <u>flash</u> breakpoints	
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	
OK キャンセル :	適用(<u>A</u>)



8)次はリセットの方針を設定できます。いくつかの方針の選択肢があって、また、リセット後の同じ時間をリセットする、いくつかの方針の選択肢をする場合は遅延時間の設定ができる。この設定はリセットするのに時間かかる時は利用できる。例えば AT91RM9200。

General Init JTAG Flash Breakpoints CPU Log Image: Construction set simulation Allows instruction set simulation Allows the emulator to simulate individual instructions when single stepping instructions. This does not normally have any disadvantages and makes debugging much faster, especially when using flash breakpoints. Endian It is the endian It is endian Image: Beset strategy J-Link supports different reset strategies. This is necessary because there is no single way of resetting and halting an ARM core before it starts to execute instructions. Hardware, hait after reset (normal) Image: Delay after reset Image: Delay after reset Image: the CPU can in most systems execute some instructions before it is halted. The number of instructions executed depends primarily on the JTAG speed, the lated the CPU. This typically halts the CPU is delayed after reset release, if a pause has been specified, J-Link waits for the specified time before reset.	SEGGER J-Link RDI V4.10f Configuration	x
Endian Little endian Big endian Beset strategy J-Link supports different reset strategies. This is necessary because there is no single way of resetting and halting an ARM core before it starts to execute instructions. Hardware, halt after reset (normal) Link continuously tries to halt the CPU. This typically halts the CPU shortly after reset release; the CPU can in most systems execute some instructions before it is halted. The number of instructions, executed depends primarily on the JTAG speed, the faster the CPU can be halted. Some CPUs can actually be halted before executing any instruction, because the start of the CPU is delayed after reset release. If a pause has been specified, J-Link waits for the specified time before trying to halt the CPU. This can be useful if a bootloader needs to be started after reset.	General Init JTAG Flash Breakpoints CPU Log Image: Allow instruction set simulation Allows the emulator to simulate individual instructions when single stepping instructions. This does not normally have any disadvantages and makes debugging much faster, especially when using flash breakpoints.	
 Beset strategy J-Link supports different reset strategies. This is necessary because there is no single way of resetting and halting an ARM core before it starts to execute instructions. Hardware, halt after reset (normal) Delay after reset ms The hardware RESET pin is used to reset the CPU. After reset release, J-Link continuously tries to halt the CPU. This typically halts the CPU shortly after reset release; the CPU can in most systems execute some instructions before it is halted. The number of instructions executed depends primarily on the JTAG speed: the higher the JTAG speed, the faster the CPU can be halted. Some CPUs can actually be halted before executing any instruction, because the start of the CPU is delayed after reset release. If a pause has been specified, J-Link waits for the specified time before trying to halt the CPU. This can be useful if a bootloader needs to be started after reset. 	Endian C Big endian	
	Beset strategy J-Link supports different reset strategies. This is necessary because there is no single way of resetting and halting an ARM core before it starts to execute instructions. Hardware, halt after reset (normal)	
		B(A)

以上の設定は Open Link を使ってデバッグする時に設定内容となります。



9) KEIL の「DOWNLOAD」機能を使う場合は「Utilities」のメニューで「Debug」上と同じ設 定が必要である。

Options for Target '	Z2148P'	x
Device Target C	utput Listing User C/C++ Asm Linker Debug Utilities	
Configure Flash	Menu Command	
O Use Target	Driver for Flash Programming	
	RDI Interface Driver 💽 Settings 🔽 Update Target before Debugging	
Init File:	ULINK ARM Debugger Edit	
0 H F .	RDI Interface Driver	
O Use Externa	Signum Systems JTAGjet	
Command:	ILPO210x_ISP.ExE	<u></u>
Arguments:]"#H" `X \$D COM1: 38400 1	
	🗖 Run Independent	
	OK キャンセル Defaults	~117

10) 「RDI Interface Driver」を選択し、「Settings」をクリックする。

Select Flash Programmer
J-Link Flash Programmer
ОК

11)「J-Link Flash Programmer」を選択する。

RDI Interface Driver Setup	-	×
Browse for RDI Driver DLL		
G:\01_work\embeded\JLinkARM_V	/410f/JLinkRDI.dll	
Browse for ToolConf File		
Debug		
Cache Uptions ✓ Cache <u>C</u> ode ✓ Cache <u>M</u> emory	Configure <u>R</u> DI Driver	
	Cancel	<u>H</u> elp



12)次の設定は"Debug"での設定と同じである。

SEGGER J-Link RDI V4.10f Co	onfiguration	8 ×
General Init JTAG Flash	Breakpoints CPU Log J-Link RDI is an RDI compliant software for J- ARM. It requires a license (RDI), which can b obtained from SEGGER (www.segger.com). This software is also capable of programming flash memory of several ARM micros, which c used to download your program to flash (Requ the add. license "FlashDL") and to set an unlin number of software breakpoints in flash (Requ the add. license "FlashDL")	Link e the an be uires imited uires
	Connection to <u>J</u> -Link CUSB Device 0 CICP/IP	
Location of config file C:\Program Files (x86)\cst\Keil\M	About License IDK3.05\uv3\Uv3.ini <u>R</u> eset Cor	e nfig
	ОК	キャンセル 適用(<u>A</u>)



上記設定した後、下記ボタンをクリックしてダウンロードができる。





3.2 ADS での設定

1)ADX を起動

Ele Search Brocessor Views System Views Execute Options Window Help	
na 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	
Teer bage File Class	
System Output Monitor R01 log [Debug Log]	
tee file TotOtti Tracer, DDI Codecentrances	
ARM RDI 15.1 -> ASYNC RDI Protocol Converter ADS v1.2 (Build number 805), Copyright (c) ARM Limited 2001.	
Ear Main mode E1	AND BOSS ARMIN ARMITTING AND IMAGES

2) 「Add」をクリックし、JLINKRDI.DLLを選択する

Ch	oose Target			? <mark>×</mark>	
	Target Environments	s			
	Target R	RDI File	Ver	rsion <u>A</u> dd	
	ARMUL 1. J-Link 1.	 5.1 G:¥01 work¥embeded¥ADSV1 2¥Bin 5.1 G:¥01 work¥embeded¥ADSv1 2¥Bin 5.1 G:¥01 work¥embeded¥JLinkARM V4 	¥ARMulatedii 1.2. ¥ARMulatedii 1.2. ¥10f¥JLinkRDIdii 4.10	0.805 0.805 <u>R</u> emove	
				Re <u>n</u> ame	
				<u>S</u> ave As	
				Configure	
	Please selec the list. Note be used.	t a target environment from the above lis that a target environment has to be con	st or add a target enviror figured at least once bef	nment to fore it can	
			ОК	Cancel Help	



3) 「Configure」をクリックすると、下記内容を出る

SEGGER J-Link RDI V4.10f Configuration	? ×
General Init JTAG Flash Breakpoints CPU Log	
Enable 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 NXP LPC2148	
RAM 32 KB @ address 0x40000000	
Flash 500 KB @ address 0x0	
✓ <u>Cache flash contents</u> Allows caching of flash contents. This avoids reading data twice and speeds up.	
the transfer between debugger and target.	
Allows verifying of flash contents. This is useful to check if the program was downloaded to flash memory correctly.	
Allow <u>flash</u> download	
Allows program download to flash. Your debugger does not need to have a flash loader. This feature requires an additional license (FlashDL).	
Skip download on CRC match	
OK キャンセル	適用(<u>A</u>)



3.3 IAR での設定

1) IAR を起動

IAR Emt	edded Workbench IDE	the same with the second little to be the	and the second se	
Elle Edit	View Project Tools Window Help	ा प ⁹ 🤨 🔥 🔂 😂 😂 😯 📷 🖓		
0 📽 🖬	日日に日間のの	• 《》》之国》》《帝国》《梁武帝》》		
Workspace	*			**
Files	#= B]			
Ready				



2) IAR の設定

IAR で IAR から提供する JLINK のドライブを使えるし、RDI インタフェースのドライバも 使える。RDI インターフェイスのドライバを使うのがお勧めする。理由は IAR バージョン に JLINK が速度と機能に制限がある。

X IAR Embedded Workbench IDE	A CONTRACTOR OF	
File Edit View Project Tools Window Help	v ^p ⊗0 A 20 🛶 🐳 V sans +	
Add Files	 ・インドゴ目ッキ会を回帰来る 	
Add Group		•
Import File List	ptr2Timer = &softTimer[0];	
Edit Configurations	ALCOULD .	
Remove		
Create New Project	erroptDataStructs(); // Set up software timer data struct	
Add Existing Project	ptrilimer, // that runs off master timer, which	
A-1	ONE_SEC_TIMER, // is 100H: Timer0 interrupt on com-	
Options Alt+F7	OHE_SEC_TIMER_INSTALLED	
Source Code Control +	23	
Make F7		
Comple Ctrl+F7	Lock */	-
Rebuild All	<pre>prs. sport = 12Mes */ LC = PILOS bit.PILE = 0: // Disable and disconnect PIL</pre>	
Clean	// FLL feed sequence	
Batch build F8	Destial divider Dobl = Clb/d +/	
Stop Build Ctrl+Break	BOIV = D:	
Download and Debug Ctrl+D	on of code configures Timer 0, match channel 0 to interrupt on	
Debug without Downloading	the shared in Min a/	
Make & Restart Debugger	Value system and nov -/	
Restart Debugger	// Disable counting	
7079- 0:	// Clear Liser counter // So Preselar	
TOPC=0:	// Clear prescaler timer counter	
TOMOR IN ART	EQ/100: // Count up to 36,864 for 1002: interrupt, period = 10ms	
TOTCR == -2;	// Clear recet flag	
TOTCR = 12	// Counting enable	
/* Prelimine	ary setup of the VIC. Assign all interrupt channels to IRQ */	
VICIntSelect	 Or // Set all VTC interrupts to IRO for now 	
VICIntEnCles	st = Cuffffffff; // Disseble all interrupts	
VICSoftIntCl	leat = OxFITTITITI // Clear all software interrutps	
	// privileged mode	
VICVectAddr VICPectAddr	 0: // Clear interrupt 0: // Clear interrupt 	
VICUEIVECUE	ar - or // clear entry of the detect low	
/*Configure	the pins that are connected to RX and TX on DART1 */	10 B
Uverview [HDC] AudioDe 4 [+] [Ai] 4		
Edit options for the selected item		
😭 💪 🚞 🖸 🚳	🤌 🐢 🐳 🤨 🛄 🛤 📝 📥 🕮 🕘 📵 💥	▲ Solution >:57
		2003) ad 60



3)オプションを設定

Options for node "ADC		×
Category: General Options C/C++ Compiler Assembler Output Convert Custom Build Build Actions Linker Debugger Simulator Angel GDB Server IAR ROM-moni J-Link/J-Trace LMI FTDI Macraigor RDI Third-Party Dri 	Target Output Library Configuration Library Options MISRA-C Processor variant Ogre ARM7TDMI-S Device Philips LPC2148 Endian mode EPU Little Big BE32 BE32 BE32 EPU	
	Cancel	



低価格、高品質が不可能? 日昇テクノロジーなら可能にする

Options for node "ADC		x
Category: C/C++ Compiler ^ Assembler Output Convert Custom Build Build Actions Linker Debugger Simulator Angel GDB Server IAR ROM-moni J-Link/J-Trace LMI FTDI Macraigor RDI Third-Party Dri	Factory Settings Config Library Output List #define Output file:	



a) IAR 社の JLINK の場合は, "J-LINK/J-TRACE"を選択してください。全機能バージョンの JLINK の場合は "J-LINK/J-TRACE"或いは "RDI"が選択できる。Open Link は全機能バー ジョンの JTAG なので、性能向上する為、"RDI"の選択をお勧めする。

Options for node "ADC	
Category:	Setup Download Extra Options Plugins Driver Imain Imain Setup macros Imain Setup macro file(s) Imain Setup Macros Imain Devige macros Imain Imain Devige description file Imain Imain Devige description file Imain Imain Imain Imain Imain Devige description file Imain Imain Imain I
	OK Cancel



b) 「J-LINK/J-TRACE」を選択した場合は、他の設定は必要ない。「RDI」を選択した場合、 JLINKRDI. DLL の場所を指定してください。

Options for node "ADO	C"	X
Options for node "ADO Category: C/C++ Compiler ^ Assembler Output Convert Custom Build Build Actions Linker Debugger Simulator Angel GDB Server IAR ROM-moni J-Link/J-Trace LMI FTDI	RDI Manufacturer RDI driver G:¥01_work¥embeded¥JLinkARM_V ☐ <u>A</u> llow hardware reset ☐ <u>E</u> TM trace	Factory Settings /410f¥JLinkRDIdII Note Use the RDI menu to specify additional driver settings. (This menu is available after the RDI driver has been located) Catch exceptions Beset Data Beset Data Undef Prefetch
Macraigor RDI Third-Party Dri	Log RDI <u>communication</u> STOOLKIT_DIR\$¥cspycomm.log	
	L	OK Cancel



c) 設定完了した後、RDI メニューが増える、下記の図の通りです。





d) RDI メニューに"CONFIGURE"のオプションがある。ここで、JTAG クロック、FLASH、ブレークポイント、CPU などの設定ができる。ご注意:FLASH と CPU 型はターゲットと一致する必要。

SEGGER J-Link RDI V4.10f Configuration	? ×
General Init JTAG Flash Breakpoints CPU Log Image: Enable 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 Image: Enable flash contents RAM 32 KB @ address 0x40000000 Flash Flash Soft KB @ address 0x0 Image: Provide flash contents Allows caching of flash contents. This avoids reading data twice and speeds up the transfer between debugger and target. Image: Provide flash contents Allows verifying of flash contents. This is useful to check if the program was downloaded to flash memory correctly. Image: Allow flash download Allows program download to flash. Your debugger does not need to have a flash loader. This feature requires an additional license (FlashDL). Image: Skip download on CRC match Skip download on CRC match	
OK ++>セル	適用(<u>A</u>)



e) IAR 環境で JLINK を使用する時、IAR の FLASHLOADER で FLASH をダウンロードはしない でください。「Use flash loader」の前のチックをなくして、JLINK の FLASH プログラミ ングアルゴリズムを使うのと、IAR 的 FLASHLOADER を使うのに、スピードが何倍の差が ある可能性がある!

Options for node "ADC'	" X
Options for node "ADC" Category: General Options C/C++ Compiler Assembler Output Convert Custom Build Build Actions Linker Debugger Simulator Angel GDB Server IAR ROM-moni J-Link/J-Trace LMI FTDI	Factory Settings Setup Download Extra Options Plugins Attach to program Yerify download Syppress download Use flash loader(s) (default),
Macraigor RDI	۲. () () () () () () () () () (
I Dird Party Der	OK Cancel



3.4 OpenOCD での設定

1) OpenOCD 概要

各種 JTAG-ICE に対応したオープンソースとして公開されているデバッガです。マイコ ン上のプログラムをデバッグする事だけが目的というわけで はなく、マイコンに接続な いし内蔵された様々なメモリへのアクセスも出来ますので、単体でマイコン内蔵の FLASH 書き込みツールとしても便利に使えま す。

ここでは ARM7 に主眼を置いていますが、OpenOCD としては ARM7 (ARM7TDMI, ARM720t), ARM9 (ARM920T, ARM922T, ARM926EJ-S, ARM966E-S) · XScale (PXA25x, IXP42x) · Cortex-M3 (Stellaris LM3, ST STM32)のデバッグ、CFI compatible NOR フラッシュ (Intel, AMD/Spansion) · 各種マイコン内蔵フラッシュ (LPC2000, LPC1700, AT91SAM7, STR7x, STR9x, LM3, and STM32x)の書き込み等をサポートしますので、興味があれば本家のサイ ト「http://openocd.sourceforge.net/」を参考に使ってみてはいかがでしょう。

2) OpenOCD の構成イメージ

OpenOCD によって構成できるシステムの概略イメージです。



OpenOCD は PC 内でネットワークのデーモンとして実行され、割り当てられたポートに 対して種々のアプリケーションからアクセスし、OpenOCD 及び JTAG I/F Hardware を介して ターゲットとコンタクトする形を取るのが一般的の様です。



3) Open Link の使用手順

EclipseでのOpenOCD環境構築は「Eclipse+OpenJTAG +OpenOCDでARMシリーズ開発環境 構築」をご参照ください。

<u>http://www.dragonwake.com/download/open-jtag/OpenJtag-Arm-All_manual.pdf</u> OpenOCD が動かすまでの大まかなネタは、

- ・OpenOCD. exe 単体では実行しても何事もなかったように終了してしまう。実行時 に諸々記述したオプションを指定し、使用する USB 接続の JTAG I/F を接続した状 態でしか使用できない。
- ·設定ファイルのある場所を検索させるには、-sオプションを付加して起動。
- ・何が起こっているのか詳細に知りたい時は-d3を指定すると、内部状況をつぶさ にコンソールに吐き出す。そのログをファイルに保存するオプションもあり。
- ・shutdown コマンドを記述しない設定ファイルを指定して起動すると常駐する。また、設定ファイル内に telnet_port の指定があれば telnet のデーモンとなるので、指定されたポートへ telnet クライアントを使ってログインできる。使用できるコマンドは help で参照可。
- ・Windows 環境で動くと言っても、自らウィンドウを持つプログラムではない。コマンドプロンプト内でひっそり動いている。強制終了するならそのコマンドプロンプト内で[CTRL]+[C]を入力するか、コマンドプロンプトのタイトルバーの×を押すが程度。
- ・設定ファイルに gdb_port の設定があれば、gdb からそのポートへアクセスし gdb のコマンドを使用してターゲットの制御が可能。
- ・FLASH WRITER の CPU TYPE に OpenOCD が指定されていると、FLASH WRITER は OpenOCD の telnet デーモンにログインし、FLASH WRITER の Write Script File に従ってコマンドを送信できる。先の telnet クライアントにて手動でコマンドを叩くといった操作を自動化できるので、複数のコマンドを併用す る必要があるターゲットのフラッシュ ROM の書き込み処理に使用してるだけ。
- ・日本語は理解してくれないので、ファイル名等は半角英数字でなくてはならない。
- ・全く同じ回路であっても、ターゲット間をつなぐケーブルの長さ等の条件によっ て両者で同じ設定で同様に動くとは限らない。JTAG ラインのプルアップ/ダウン・ コネクタの接触抵抗・ノイズの影響等を考慮しながら JTAG のスピードを調整する。 速ければ良い物でもない。
- ·予め用意された設定ファイルはよく見かける主要なマイコン用の物しか用意され ていない。似たような型番だからといって同じ設定ファイルが使えるわけでもな いので、新しいデバイスに対応させるにはそれ相応のスキルが必要。
- ・チップ内蔵のフラッシュ ROMに書き込むだけなら OpenOCD を使うメリットは無く、



サポートされていないチップの方が遙かに多い。メーカの提供するツールを適用 する方が良い。

・本家に協力者として参加するのも良し。自力で解決したいのであればメーリング リストに参加すべき。英語は必須。

「OpenOCD」というプロジェクトのホームページは下記 URL であります。

http://openocd.sourceforge.net/



四、Open Link で J-Flash ARM 使用方法

4.1 Open Link のドライバのインストール完成後、二つのショートカットアイコンが出てくる。一つが J-FLASH ARM である。このアプリケーションは FLASH のプログラミングで使う (J- FLASH ARM License 必要)。



4.2 初回使用する時、「File」→「Open Project」からターゲットを選択する必要。

🛃 J-Flash A	RE V5. 206 - 10:1	Program P	ile	#\ 58	IGGE	R\J	Lini	AR	E 78	205	\Sm	gele	us/J	Fla	sh'i	Prop	inci	Pi 1	ex\AT915	
gile gdit gi	es Jurgot Options	findow Help																		
27copect	🗆 🔀	A. CrABoe		t s	and	Set	tis	gie\			R /•	¢١,	FRAME (n-t	o, b	in:				
Name Connection	Velue USB (Device 0)	Address: 0x1	00000	0	×1	8	*4													
1.5.000		Address	-	1	z	3	4	5	6	7	8	9	A	B	C	1	E	F	esct1	*
Inf J I Als speed	30 kHz 4000 kHz	1000000	ØF	88	60	EA	PE	FF	PF	EA	FE	PF	b b	EA	PE	b b	FF	EA		****
TAP number	<not td="" used<=""><td>1000010</td><td>FE</td><td>PP</td><td>FF</td><td>EA</td><td>FE</td><td>FF</td><td>FF</td><td>EA</td><td>19</td><td>66</td><td>88</td><td>EA</td><td>88</td><td>58</td><td>AÐ</td><td>E1</td><td></td><td></td></not>	1000010	FE	PP	FF	EA	FE	FF	FF	EA	19	66	88	EA	88	58	AÐ	E1		
IR len	<not used=""></not>	1000820	84	81	98	85	D3	FØ	21	E3	8E	50	20	E9	RF	EØ	88	E1	tP	=
-		1000030	10	PP	2F	E1	ØE	50	BD	E8	B 1	PØ	21	E3	89	88	88	E1	/Pt.	
Linp Flock speed	47923200 He	1 000240	84	FH	5E	E2	81	19	AR	E3	98	180	9F	E5	MF	EB	88	E1		
Endian	Little	1000850	18	PP	2F	E1	81	89	6B	E3	D1	PØ	21	E3	88	88	9P	ES	·	
Check core Id	Yes	1000860	02	FØ	21	E3	88	10	AB	EI	68	88	48	E2	13	FØ	21	E3		
AFM core id	OK3F0F0F0F	1000870	88	DØ	AB	E1	60	20	9F	ES	ØF	EØ	68	Ei	1.0	FF	2F	Et		/.
Doe target HAM	747	1000080	FE	PP	FF	EA	84	EB	48	E2	88	48	20	E9	88	EB	4F	E1	N	
RAM size	16 KB	1 300350	88	48	20	E9	81	80	20	E9	44	EØ	9F	ES	88	81	9E	85		
Use DCC mode	Yes	1888868	88	E1	8E	EG	13	FØ	21	E3	ØE	58	20	EŸ	ØF	EB	88	Et	tP	
P. C.		1.889889	18	PP	2F	Et	RE	58	BD	E8	92	PB	21	E3	28	EH	98	E5		
Hash memory University	A191SAM/X256 mt	1000800	30	EI	8E	25	81	20	BD	ES	88	40	BD	E8	RE	FØ	6F	E1	Ø	
Size	256 KB	1.898809	88	88	FD	ER.	FE	FF	FF	EA	FE	FF	PP	FA	FE	PP	FF	EA		
Flesh Id	0+0	1 DEMONSTR	FR	513	00	100	DID	FR	FF	FF	FC	100	-	0.0	100	00	0.0	ED		
Check Bash Id	No	1 999920	24	80	-00	50	20	00	OF	10	0.0	80	-	T.C.	80	60	04	-		
Base address	27 bits a Lobin	1999199	01	29	de	89	89	ER	RE	FR	OR	80	50	F1	FB	88		80	nl 7	
Logar Lanch	Se Dies i emp	10001100	MF	DIA	ED	ER	14	FR	dF	E2	81	180	12	E1	113	FR	47	1.0	0	6 -
		Toomero	1.00			-							***		-					
1.105																				
Opening project	Eile (C:)Fregran Fi	149\SEGGER\JL	i adeki	UN VS	205	Sungi	les\J	Flas	hirr	ojact	File	TAI	91.PEA	0162	n.	[dea		_	_	A
- Project open	and successfully									600	0.2									
- Bata file or	and successfully (1)	ULAS Systems' DSE8 bytes 1	Esta	deed re [Werk:	0 and	1 431 85458	16/148 [[][]	a)ar	c./#84	eb14	2744	eet va	dia 18	Base's	tL and	ucre de	-3194	UA/Ese/wt91_ebms.8	bsm.]
Class preject																				100
- Project clo	ted 1. file (f.)Promo Fi	A-USERGERATE	inkl	RIN VS	iomA.	·		Flar	hipm	iari	File	-1471		12299		harb				
- Project oper	and morestfully		22					5000	020	333	2000	0.005	1000		2.42					
Opening data fi - Data file of	ile EC: "Documents and peaked processfully (1)	Settingstwom 984 bytes, 1	1 mg	1. CI	C = 1	1x-1 0x55	o. biz 07A35	d	1											
4																				3
Baudy												T								1



4.3 「File」→「Open…」で書き込みターゲットファイルを選択する。.bin ファイル、.hex ファイルあるいは.mot ファイルどちでもできる。開始アドレスにご注意。

後は $\lceil \text{Options} \rceil \rightarrow \lceil \text{Project settings} \rceil$:

Open project			? 🛛
查找范围(L):	C ProjectFiles	- 🖬 😁 🖃 -	
	📾 ADuC7020. jflash	📾 AT91SAM7S256.jflash	MAC7111.
	📷 ADuC7030.jflash	📷 AT91SAM7X128. jflash	📷 NS9360. j
我最近的文档	🖬 ADuC7032. jflash	📷 AT91SAM7X256.jflash	📷 NS9750. j
	📷 ADuC7229. jflash	🛅 DragonballMX1.jflash	🔂 PCF87750
	📷 AT91FR40162. jflash	📷 Evaluator7T. jflash	🗟 S3F445HX
桌面	📷 AT91M55800A.jflash	📷 LH7A40x_LogicPD. jflash	🖬 SocLiteP
	🖬 AT91R40008_AT91EB40A. jflash	📷 LH75411. jfl عده h	STR710 j
	📷 AT91RM9200_CSB337.jflash	□ LH79520_Log: 类型: JFLASH 文	伴。
我的文档	📷 AT91RM9200_CSB637.jflash	圖 LH79524_Log: 大小: 8 09 KB	4-27 16:07 j
	🖬 AT91RM9200_EK.jflash	ELPC2106.jfl	j
<u> </u>	📷 AT91SAM7A1_EK.jflash	📷 LPC2129. jflash	🖬 STR912. j
我的电脑	📷 AT91SAM7A3.jflash	🐻 LPC2138. jflash	TMS470R1
	aT91SAM7S32.jflash	🛅 LPC2148. jflash	TMS470R1
3	📷 AT91SAM7S64.jflash	🐻 LPC2294. jflash	TMS470R1
网上邻居	🛅 AT91SAM7S128. jflash	📷 LPC2294_PhyCORE. jflash	TMS470R1
	K		Σ
	文件名 (M): Flash	•	打开 (2)
	文件类型(I): J-Flash project	files (*.jflash) 💌	取消



4.4 ARM タブでターゲットチップを選択できる。オンチップフラッシュでない場合は、 「Generic ARM7/ARM9」を選択する。(ここに「NXP LPC2148」を選択)

Project settings	_	S ×
General Target Interface CPU FI	ash Production	Check core ID
Core Little endian ▼ Clock speed Clock speed Auto detection 49152000 Hz Use following init sequence:	<u>v</u>	ID 4F1F0F0F Use target <u>B</u> AM (faster) Addr 40000000 16 KB -
# Action Value0 0 Reset 0	Value1 0 ms	Comment
1 Write 32bit 0xE01FC084 2 Write 32bit 0xE01FC080 3 Write 32bit 0xE01FC08C 4 Write 32bit 0xE01FC08C 5 Delay 0xE01FC080 6 Write 32bit 0xE01FC080 7 Write 32bit 0xE01FC080 8 Write 32bit 0xE01FC08C 9 Write 32bit 0xE01FC08C 9 Write 32bit 0xE01FC040	0x00000022 0x0000001 0x0000004A 0x00000055 20 ms 0x00000003 0x0000004A 0x00000055 0x00000055 0x0000001	Mul = 3 (PLLCFG) Enable PLL (PLLCON) Update PLL configuratio Wait for PLL Select PLL (PLLCON) Update PLL configuratio Map User Flash into Vec
		OK キャンセル 適用(A)



4.5 FLASH タブで, Opon project だったら、特に何も設定する必要はない。新しく作った プロジェクトの場合は、次の設定が必要になる。この前の ARM タブで "Generic ARM7/ARM9" を選択した場合、FLASH タブで FLASH 型を選択できる。

General Target Interface CPU Flash Production FlashBank Bank[0] ✓ Add Remove Use custom BAMCode … … Base Add Occount Occupanization 32 Bits x 1 Chip(s) Manufacturer NXP … … … … Chip LPC2148 internal … … … Size 500 KB Sectors 27 … … Chip Start/End sector … … … Start/End sector … … … …
FlashBank Bank[0] Use custom BAMCode Base Addr 00000000 Organization 32 Bits x 1 Chip LPC2148 internal Size 500 KB Sectors 27 Start/End sector Start/End sector Individual sectors Start Addr Sector[0]: 0x0
□ Use custom BAMCode Base Addr 00000000 Organization 32 Bits x 1 Chip(s) Manufacturer NXP Chip LPC2148 internal Size 500 KB Sectors 27 Start/End sector Start/End sector Individual sectors Start Addr Sector[0]: 0x0 Sector[0]: 0x0 - 0xFFF
Base Addr 00000000 □rganization 32 Bits x 1 Chip(s) Manufacturer NXP Chip LPC2148 internal Size 500 KB Sectors 27 Chip Start/End sector Individual sectors Start Addr Sector[0]: 0x0 Individual sectors
Manufacturer NXP Chip LPC2148 internal Size 500 KB Sectors 27 C Start/End sector Individual sectors Start Addr Sector[0]: 0x0 ✓ Sector[0]: 0x0 - 0xFFF
End Addr Sector[26]: 0x7CFFF ✓ Sector[1]: 0x1000 - 0x1FFF ■ Selected ranges: ✓ Sector[2]: 0x2000 - 0x2FFF ✓ Sector[2]: 0x3000 - 0x3FFF 27 Sectors, 1 Range: ✓ Sector[4]: 0x4000 - 0x4FFF ✓ Sector[5]: 0x5000 - 0x4FFF 0x0000 - 0x7CFFF ✓ Sector[5]: 0x5000 - 0x5FFF ✓ Sector[6]: 0x6000 - 0x6FFF



4.6 主なメーカーの FLASH をサポートする。また最新のデバイスをサポートするためにアッ プグレードを継続する。

設定終了後、ターゲットに対する操作ができる。通常の手順は「Connect」→「Erase Chip」 →「Program」である。

Project settings	2 ×
General Target Interface CPU Fla	Sh Production
FlashBank Bank[0]	▼ Add Remove
Use custom <u>R</u> AMCode	
Base Addr 00000000 0	rganization 32 💌 Bits x 1 💌 Chip(s)
Manufacturer NXP Chip LPC2148 internal Size 500 KB Sectors C Start/End sector Start Addr Sector[0]: 0x0 End Addr Sector[26]: 0x7CFFF Selected ranges: 27 Sectors, 1 Range: 0x0000 - 0x7CFFF	27 ● Individual sectors ● Sector[0]: 0x0 - 0xFFF ● Sector[1]: 0x1000 - 0x1FFF ● Sector[2]: 0x2000 - 0x2FFF ● Sector[3]: 0x3000 - 0x3FFF ● Sector[3]: 0x3000 - 0x3FFF ● Sector[5]: 0x5000 - 0x3FFF ● Sector[5]: 0x5000 - 0x3FFF ● Sector[5]: 0x5000 - 0x5FFF ● Sector[6]: 0x6000 - 0x6FFF ● Sector[6]: 0x6000 - 0x6FFF ● Sector[6]: 0x6000 - 0x6FFF
	OK キャンセル 適用(A)



4.7 3.30g バージョンから、J-FLASH ARM は XSCALE をサポートする。

Project settings	? 🔀
General JTAG CPU Flash Production	
Endian Little	
Addr 0 4KB -	
Use following init sequence:	
# Type Value0 Value1 Comment	
Add Insert Delete Edit Down	
确定	取消 应用())



五、ARM9 ボード(Mini2440 シリーズ)書込み手順

Note:「<u>四、Open Link で J-Flash ARM 使用方法</u>」を参照して行えますが、元々J-FLASH ARM プロジェクトの中、ARM9 を使えるものがありません。それ以外、Open Link では Nand Flash を直接書き込めないため、本章で説明します。

ARM9 ボードを書き込む用のリソースダウンロード URL:

http://www.dragonwake.com/download/open-link/arm9-resource.zip

1. Nor Flash に書き込む

① "File -> Open -> Open Project…"を選べ、「¥open-link¥s3c2440.jflash」を開いて ください。





②"Target -> Connect"を選択



③ "File -> Open"を選べ、書き込み対象バイナリファイル、例えば「¥u-boot. bin」, "Start address"に0を入力

ile <u>E</u> dit <u>V</u> iew	Target Options Window	Help	
Project - s3c	:2440 - 🗆 🗙		
Name	Value		
Connection	USB (Device 0)		
Target interface	JTAG		
Init JTAG speed	12000 kHz		
JTAG speed	12000 kHz		
TAP number	0		
IR len	0		
CPU	A. 4a		
Endian	Little		
Check core ld	No		
Core Id	0x3E0E0E0E		
Use target BAM	Yes		
RAM address	0x30000000		
RAM size	64 KB		
		Enter start address	
Flash memory	Auto detection		
Flash size	2048 KB	Start address R	
Algorithm	AMD		
Base address	0x0	Carried L	
Organization	16 bits x 1 chip		
· · · · · · · · · · · · · · · · · · ·			
,			

- ④ "Target -> Auto"をクリックして書込みを自動に始める
- 2. Nand Flash に書き込む
- Open-Link で Nand Flash に書き込めないため、まず、上記の手順で Nor Flash に 「¥u-boot.bin」を書き込む必要です。

①シリアルケーブルで ARM9 ボードが PC と接続にし、ハイパーターミナルを設定 ハイパーターミナルの設定は「<u>Mini2440 ボードマニュアル</u>」の P41 をご参照ください。 ARM9 ボードに Nor Flash モードで電源を入れ、u-boot を起動します。

ハイパーターミナルに下記のようなメッセージが表示されます、0を減るまでにスペースキ ーを押します。



U-Boot 1.1.6 (Nov 26 2010 - 08:49:16)
DRAM: 64 MB
Flash: 2 MB
NAND: 256 MiB
In: serial
Out: serial
Err: serial
UPLLVal [M:38h, P:2h, S:2h]
MPLLVal [M:5ch, P:1h, S:1h]
CLKDIVN:5h
++
S3C2440A USB Downloader ver R0.03 2004 Jan
++
USB: IN_ENDPOINT:1 OUT_ENDPOINT:3
FORMAT: <addr(data):4>+<size(n+10):4>+<data:n>+<cs:2></cs:2></data:n></size(n+10):4></addr(data):4>
NOTE: Power off/on or press the reset button for 1 sec
in order to get a valid USB device address.
Hit any key to stop autoboot: 0
100ask Bootloader for OpenJTAG
[n] Download u-boot to Nand Flash
[o] Download u-boot to Nor Flash
[k] Download Linux kernel uImage
[j] Download root_jffs2 image
[y] Download root_yaffs image
[d] Download to SDRAM & Run
[z] Download zImage into RAM
[g] Boot linux from RAM
[f] Format the Nand Flash
[s] Set the boot parameters
[b] Boot the system
[r] Reboot u-boot
[q] Quit from menu
②USB ケーブルで ARM ボードの USB デバイスポートが PC と接続

③初回使う場合、USB ダウンロードツールの DNW ドライバをインストールしてください。



インストール方法は「<u>Mini2440 ボードマニュアル</u>」の P128 をご参照ください。実際のドラ イバファイルは上記解凍フォルダ「¥drivers¥dnw_win7_64bit」(Win7)或いは 「¥drivers¥dnw_xp」を使います。

インストール後、下記 USB 右側に「OK」が表示されれば、ドライバは正常にインストール されたことを明らかにします。



④u-bootのメニューに従い、ファイルを Nand Flash に書き込む事が出来ます。

例えば、「n」: "Download u-boot to Nand Flash"と言うメニューを選択する場合、 USBケーブルでDNWツールを使ってファイルをNand Flashにダウンロードします。 ※DNW ツールが「¥tools」にある

実は、弊社の Mini2440/Micro2440 ボードの場合、ボード自身は Nor Flash に既にブートロ ーダを書き込んでいますので、そのブートロードを利用すれば、上記のようなダウンロー ド操作は必要ありません。弊社以外の ARM9 ボードで Open Link を使う際、上記の手順を参 照してください。

※上記の手順は Mini 2440 ボードで検証済み

六、OpenLink フォームウエア更新手順

Note: 普通の場合、本章の手順は実施必要がありません。 下記の場合のみはフォームウエア更新必要です。 ①バージョンアップ ②既存のフォームウエアが壊れた



1. 更新用リソースダウンロードURL:

http://www.dragonwake.com/download/open-link/firmware-update.zip

- 2. 圧縮ファイル中の sam-ba_2.10. exe をインストールしてから PC を再起動
- 3. OpenLink のカーバーを取り出す
- 4. AT91SAM7S64の既存フォームウエアを消す
 - ① AT91SAM7S64 (U1)の電源を切断 (USB ケーブルを抜く)
 - ② ERASE の二つ PIN をショート



- ③ USB ケーブルで OpenLink を PC と接続
- ④ 60秒を待って USB ケーブルを抜く
- ⑤ ERASE のショートを戻す
- 5. SAM-BA Boot フォームウエア更新の準備
 - USB ケーブルを抜く(AT91SAM7S64 電源を切断)
 - TST の2つPINをショート



- ③ USB ケーブルで PC と接続
- ④ 60秒を待って USB ケーブルを抜く
- ⑤ TST のショートを戻す
- 6. USB ドライバーインストール



① USB ケーブルで PC と接続

「sam-ba_2.10. exe」をインストール時に、一緒に USB ドライバーもインストールされています。

*Windows XP の場合、デフォールトのままインストールできます。

*Windows 7 の場合、OS が自動に別のデバイスドライバ(正しくないドライバー)をインストール します。正しくインストールするため、手動インストール必要です。(32bit の Win7 のみをサポート)

コントロール パネル-->ハードウェアとサウンド-->デバイスとプリンター-->デバイスを右クリック--> ハードウエア-->プロパティ-->ドライバー-->ドライバーの更新-->ドライバーソフトウエアを手動で検索 してインストール-->コンピュータ上のディスクドライバーの一覧から選択します-->ディスク使用

参照パスは sam-ba_2.10. exe のインストール先となります。

例:「G:¥02_tools¥embedded¥SAM-BA v2.10¥drv」

7. SAM-BA ツールを起動してからフォームウエアを更新



💽 SAT-BA 2.9			
Select the connection : Select your board :	\usb\ARM0 at91sam7s64-ek	•	
Connect	,	Exit	

ボードの種類は at91sam7s64-ek です。

Select your board : at91sam7s64-ek	•
③ Connect をクリック	



🔽 SAT-BA 2.9 - at91sam7s64-	-ek		
File Script File Link Help			
at91 sam7s64 Memoru Displau			
Start Address : 0x200000 Refresh Size in byte(s) : 0x100	☐ Display format C ascii C 8-bit C 16-bit	Applet traces on DBGU	
0x00200000 0xEA000013	0xEAFFFFFE 0xEA000054 0xEAFFFFFE		
0x00200010 0xEAFFFFE	OxEAFFFFFE OxEAFFFFFE OxE599820C		
0x00200020 0xE3A0D004	0xE58BD128 0xE59AD04C 0xE59CD004		
0x00200030 0xE21DD001	0x125EF004 0xE59AD03C 0xE21DDF80		
0x00200040 0x01CC80B0	0x11CC80B2 0x13A0D001 0x158CD004	_	
<u>0x00200050</u> 0xF25FF004	0vF10F0000 0vF321F0D1 0vF28F200C		
EEPROM AT24 Flash SRAM		1	
Download / Upload File			
Send File Name :	🚔	Send File	
Receive File Name :	∠	Receive File	
Address : 0x100000 Size	(For Receive File) : 0x1000 byte(s)	Compare sent file with memory	
Scripts			
Disable BrownOut Detector (GPNVM0)			
loading history file 0 events added	2 / TF8 4 13)		
(AT91-ISP v1.13) 1 %	// INU.T.10/		
(AT91-ISP v1.13) 1 %			
		WebVADM0 Reards = 101ep= 7-04 at	
		Nuso verimu i board : atorsam7s64-ek	

設定はデフォールトのまま、下記のようにフォームウエアの場所を 参照しフォーム ウエア「V84.10.bin」ファイルを選択して、(右の)「Send File」ボタンをクリック(更新 時、何にかがあっても「Yes」をクリック)

- Download / Upload File Send File Name :

更新完了後、OpenLink は普通通り使用できます。