

Eclipse+OpenJTAG +OpenOCD で ARM シリ ーズ開発環境構築



copyright@2011

※ この文書の情報は、事前の通知なく変更される可能性があります。
 ※ (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	2009/07/31
2	Ver1.1	誤った環境構築用、ワーク用のパスを訂正	2010/07/17
3	Ver1.2	P58~P63 の説明画面の訂正	2010/07/20
4	Ver1.3	フォルダ名誤記訂正	2010/11/22
		ARM EABI「arm-none-eabi-gcc」確認コマンド誤記訂正	
		5.2章 Zylin Embedded CDT のインストール手順変更	
5	Ver1.4	Linux 環境での開発手順を追加	2011/09/28



第一章	背景	. 5
第二章	ARM シリーズ開発の仕組み・イメージ	. 6
第三章	用意するもの	7
第四章 イン	/ストール手順	8
4.1 Oper	nJTAG のドライバをインストールする	8
4.2 ソフト	·ウェアをインストールする	11
4.2.1	「arm-none-eabi.exe」をインストール	12
4.2.2	「openocd」をインストール	16
4.2.3	「yagarto」をインストール	20
4.2.4	「yagarto-tools」をインストール	24
4.2.5	Cygwin のダウンロード&インストール	27
4.2.6	「jre」をインストール	30
4.2.7 I	「Eclipse」をインストール	30
4.3 ソフト	>ウエアインストール後の動作確認	31
4.3.1 (OpenOCD の確認	31
4.3.2	コンパイラ確認	32
第五章 Ecl	ipse の設定	36
5.1 Eclips	se を起動する	36
5.2 Eclips	se プラグイン(Zylin Embedded CDT)インストール	38
5.3 ビル	[©] の設定	39
5.4 Open	OCD の設定	40
5.5 Cygw	in のソース・ルックアップ・パスの設定	44
5.6 Eclips	se デバッグ用のコマンド及びショットカットキー一覧	47
第六章 AR	Μ シリーズデバッグ手順	48
6.1 ARM	I7 の LPC2148	48
6.1.1 l	_PC2148 ボード購入 URL	48
6.1.2	ハードウエア動作確認	48
6.1.3 I	_PC2148 用のサンプル「LED」をデバッグ	49
6.2 ARM	7 の LPC2388	70
6.2.1 l	_PC2388 ボード購入 URL	70
6.2.2	ハードウエア動作確認	70
6.2.3 l	_PC2388 用のサンプル「LED」をデバッグ	72
6.3 ARM	Cotex-M3 の STM32F103	86
6.3.1	STM32F103 ボード購入 URL	86
6.3.2	ハードウエア動作確認	86
6.3.3 l	_ED サンプルデバッグ	87
6.4 ARM	9 の MINI24401	02
6.4.1 I	MINI2440 を購入1	02
6.4.2	ハードウエア動作確認	02
6.4.3 l	_ED サンプルデバッグ 1	03
6.4.4 ι	រ–boot サンプルデバッグ 1	29
第七章 Lin	ux 環境上の OpenJTAG の使用手順1	36
7.1 ハー	・ドウエア、ソフトウエアインストール1	36



7.1.1 Linux で OpenJTAG の自動認識	
7.1.2 OpenOCD、GDB、クロスコンパイルチェイン、Eclipse のインストール	
7.2 OpenJTAGの使用	



第一章 背景

近年, ARM プロセッサが急速に広まっています. さらに、ARM9, ARM11, ARM-M, Cortex といった新たなアーキテクチャが 次々と発表されています. これらの新しいプロセッサでは、従来から存在したARMが拡張され、さらにARM9とARM11が追加され ました.

本書では、従来から広く利用されているGNU クロス開発ツールを、これらの最新アーキテクチャ/命令セットへ対応させます. 最 新版のGNU ツールを実際に使って、対応したクロス開発環境を作ります.

組込アプリを開発する際に、2点は非常に重要です。一つは統合開発環境(IDE)となり、もう一つはデバッグ環境です。いろんな 統合開発環境ツールがありますが、無料で使用できるオープンソースEclipseプラットフォームはJava アプリを含めて、C/C++ア プリを開発用の全世界共通のIDEです、今後、ますます多くの使用者が増えていきます。なお、デバッグツールはオープンソース Open-On-Chip Debugger(OpenOCD)という便利ツールがあります。

Open-On-Chip Debugger(OpenOCD)は、システムプログラミングにおけるデバッグと埋め込まれた対象装置がないかどうかテストされる境界走査を提供することを目指します。

OpenOCDのGDB と接続するためにOpen JTAG コネクトを使う必要です。ただし、パラレルJTAGの転送スピードが遅いので、 本書はカスタマイズ化のOpenJTAGを利用し、Eclipse、OpenOCDなどオープンソースとあわせて、ARMシリーズ開発を行います、 例えば、ARM基板からダウンロード、書き込み、デバッグを行います。

GDBは以下の基板をデバッグできます。

- 1 ARM7(ARM7TDMIŁARM720t)
- ② ARM9、(ARM920t、ARM922t、ARM926ej-s、ARM966e-s)
- ③ XScale(PXA25x、IXP42x)
- ④ Cortex-M3(LM3とST STM32)



第二章 ARM シリーズ開発の仕組み・イメージ

組込開発のよいプラットフォームは基本的に4つモジュールを含まれます。

- ① 統合開発環境:IDE(Integrated Development Environment)、例えばEclipse
- ② クロスコンパイラツールチェイン(Cross Complier tools chain)
- ③ リモートデバッグツール(例: OpenOCD、 Open On-Chip Debugger)
- ④ 開発用のパソコンとARM基板のコネクタ(JTAG)

全体のイメージは下記の図です。





第三章 用意するもの

以下は Windows 用です。

3.1 ソフトウエアダウンロード URL:

弊社のサーバーから全て必要のものを一括ダウンロードできます。 http://www.dragonwake.com/download/open-jtag/openJTAGIDE.zip 中身に、下記内容を含まれます。 ソフトウエアリスト(名前をクリックしてからツールもそれぞれダウンロードできます) * 解凍先:お好きな場所を選べられても OK(例:D:¥tmp)

01.arm-2009q1-161-arm-none-eabi.exe 02.openocd-0.1.0.msi 03.yagarto-bu-2.19.1_gcc-4.3.3-c-c++_nl-1.17.0_gi-6.8.50_20090311.exe 04.yagarto-tools-20070303-setup.exe 05.Cygwin 06.jre-6u7-windows-i586-p.exe 07.eclipse-cpp-galileo-win32.zip

3.2 OpenJTAG 用のデバイス(jtag デバッガ、弊社で販売しているもの)

3.2.1 Open-JTAG の購入 URL:

http://www.csun.co.jp/SHOP/200905191.html

3.2.2 デバイスダウンロード URL:

http://www.dragonwake.com/download/open-jtag/open-jtag-driver.zip

①usb-driver(OpenJTAG 用の USB ドライバ)

* 解凍後:D:\tmp\topen-jtag-driver

3.3 本書使用サンプルソースコード

http://www.dragonwake.com/download/open-jtag/gcc_openjtag_eclipse_arm_example.zip

*解凍後:(D:¥OpenJTAG¥ gcc_openjtag_eclipse_arm_example)

- 1. 「config」: OpenOCD 用の設定ファイル
- 2. 「lpc2148」: LPC2148 ボード用のサンプル
- 3. 「lpc2388」: LPC2388 ボード用のサンプル
- 4. 「stm32」: stm32FX103 ボード用のサンプル
- 5. 「mini2440」: mini2440 ボード用のサンプル



第四章 インストール手順

4.1 OpenJTAG のドライバをインストールする

OpenJTAG をパソコンの USB ポートに挿入して、下の通りにドライバをインストールしてください。





新しいハードウェアの検出ウィザード		
検索とインストールのオブションを選んでください。		
 次の場所で最適のドライバを検索する(S) 下のチェック ボックスを使って、リムーバブル メディアやローカル パスから検索できます。検索された最適のドライバがインストールされます。 リムーバブル メディア (フロッピー、CD-ROM など)を検索(M) 次の場所を含める(D): D*tmp¥open-jtag-driver¥usb-drivers 検索しないで、インストールするドライバを選択する(D) 予約らドライバを選択するには、このオブションを選びます。選択されたドライバは、ハードウェアに最適のものとは限りません。 		
く戻る(B) (次へ(N)) キャンセル		
新しいハードウェアの検出ウィザード		
ソフトウェアをインストールしています。お待ちください		
100ASK JTAG		
< 戻る(B) (次へ(N)> キャンセル		

USBドライバのインストールは3回があります。インストール完了すると、デバイスマネージャで三つのデバイスが見えます。



ファイルビ 抹作仏 表示公 ヘルブ(山) ・ 回 ● ② ③ 回 ● ● UISE (Universal Serial Bue) コントローラ ● 100ASK Serial Port ● 100ASK Serial Port <th>鳥 デバイス マネージャ</th> <th></th>	鳥 デバイス マネージャ	
 ・ ● 回 ● ● 回 ● ■ ● ■ ● ■ ● ● ● ● ● ● ● ● ●	ファイル(E) 操作(A) 表示(V) ヘルブ(H)	
ISB (!hiversal Serial Bus) コントローラ 100ASK JTAG 100ASK Serial Port 100ASK Serial Port 100ASK Serial Port 100ASK Serial Port Intelfや 82801 BA/BAM USB Universal Host Controller - 2442 Litelf(%) 82801 BA/BAM USB Universal Host Controller - 2444 NEC POI to USB Den Host Controller (B1) KEC POI to USB Open Host Controller USB ルート ハブ USB 北ート ハブ USB 北ート ハブ USB オード ハブ USB オード ハブ USB 水ート ハブ USB ホート ハブ USB 水ート ハブ WOLL 14578710 5118.DeviceDesc% (COM4) Y ECP ブリンタ 米ート (UPT)		
	ISB (Universal Serial Bus) 100ASK JTAG 100ASK Serial Port 100ASK Serial Port 100	
- プ %VID_1457&PID_5118.DeviceDesc% (COM4) - プ ECP プリンタ ポート (LPT1)	日 🐨 プロセッサ 日 🗃 フロッピー ディスク コントローラ 日 🤳 フロッピー ディスク ドライブ 日 🚽 ポート (COM と LPT)	
	- ダ %VID_1457&PID_5118.DeviceDesc% (COM4) - ダ ECP プリンタ ポート (LPT1)	

※ OpenJTAG は USB シリアルポートとして使えます。



4.2 ソフトウェアをインストールする

順番で以下のソフトウェアをインストールしてください 01.arm-2009q1-161-arm-none-eabi.exe(EABIを持ってコンパイラ) 02.openocd-0.1.0.msi(jtag デバッガ用ソフト) 03.yagarto-bu-2.19.1_gcc-4.3.3-c-c++_nl-1.17.0_gi-6.8.50_20090311.exe(gcc) 04.yagarto-tools-20070303-setup.exe(各種ユーティリティ) 05.Cygwin 06.jre-6u7-windows-i586-p.exe(Java) 07.eclipse-cpp-galileo-win32.zip(Eclipse) メモ:①もしバージョン 1.4.2 以上の JREを既にパソコンにインストールされたら、 06 番の JRE のインストールが不要です。 ②05 番の eclipse を解凍して OK、インストール不要

ダウンロードしたインストールファイルは「D:¥tmp¥openJTAGIDE」に格納されます、本書に全てソフトウエアは「D:¥embeded」にイ ンストールされます。



4.2.1 「arm-none-eabi.exe」をインストール

📲 Sourcery G++ Lite for ARM EABI		
	Sourcery G++ Lite for ARM EABI Wizard	
Welcome! Important Information	InstallAnywhere will guide you through the installation of Sourcery G++ Lite for ARM EABI.	
O Choose Install Set O Choose Install Folder	It is strongly recommended that you quit all programs before continuing with this installation.	
Add to PATH? Choose Shortcut Folder Pre-Installation Summary	Click the 'Next' button to proceed to the next screen. If you want to change something on a previous screen, click the 'Previous' button.	
O Installing Install Complete	You may cancel this installation at any time by clicking the 'Cancel' button.	
CODESOURCERY		
InstallAnywhere by Macrovision – Cancel	Previous Next	



🔄 Sourcery G++ Lite for ARM	EABI
	License Agreement
Welcome! Important Information	Please read the Software License Agreement below, by scrolling if necessary, and accept by selecting "I accept the terms of the License Agreement " at the bottom of this page.
 Choose Inistall Set Choose Install Folder Add to PATH? Choose Shortcut Folder Pre-Installation Summary Installing Install Complete 	CODESOURCERY, INC. ("CODESOURCERY") IS WILLING TO LICENSE THE SOFTWARE ONLY UPON THE CONDITION THAT YOU ACCEPT ALL OF THE TERMS CONTAINED IN THIS SOFTWARE LICENSE AGREEMENT. PLEASE READ THE TERMS CAREFULLY. BY CLICKING ON "I accept the terms of the License Agreement", YOU WILL INDICATE YOUR AGREEMENT WITH THEM. IF YOU ARE ENTERING INTO THIS AGREEMENT ON BEHALF OF A COMPANY OR OTHER LEGAL ENTITY, YOUR ACCEPTANCE REPRESENTS THAT YOU HAVE THE AUTHORITY TO BIND SUCH ENTITY TO THESE TERMS, IN WHICH CASE "YOU" OR "YOUR" SHALL REFER TO YOUR
~	I do NOT accept the terms of the License Agreement
InstallAnywhere by Macrovision — <u>C</u> ancel	Previous Next







🖳 Sourcery G++ Lite for AR	M EABI 📃 🗆 🔀
	Choose Install Folder
 Welcome! Important Information Choose Install Set Choose Install Folder Add to PATH? Choose Shortcut Folder Pre-Installation Summary Installing 	Where Would You Like to Install? D:¥embeded¥arm-eabi Restore Default Folder Choose
InstallAnywhere by Macrovision	Previous Next

「Next」をクリックすると、インストールが始まります、インストール完了までに何分がかかります、完了までにお待 ちください。



4.2.2 「openocd」をインストール





🚽 OpenOCD 0.1.0 Setup		
End-User License Agreement Please read the following license agreement carefu		
GNU GENERAL PUBLI Version 2, June 1991 Convright (C) 1989, 1991 Free Software Fr	C LICENSE	
59 Temple Place, Suite 33 USA Everyone is permitted to copy and distribu of this license document, but changing it is), Boston, MA 02111-1307 te verbatim copies s not allowed.	
Preamble		
Print Ba	ick Next Cancel	
Belect the way you want features to be installed.		
Click the icons in the tree below to change the way	features will be installed.	
OpenOCD 0.1.0 Manual Board config files Interface config files Target config files OpenOCD source code Modify PATH variable	Main OpenOCD executable, license and build info This feature requires 944KB on your hard drive. It has 5 of 5 subfeatures selected. The	
Drivers ftd2xx.dll and libusb0.dll Location: D:¥embeded¥openocd¥	subfeatures require 2864KB on your hard drive.	
Reset Disk Usage	Back Next Cancel	



閔OpenOCD 0.1.0 Setup	_ 🗆 💌
Ready to install OpenOCD 0.1.0	\odot
Click Install to begin the installation. Click Back to review or change any of your installation settings. Click Cancel to exit the wizard.	
<u>B</u> ack Install	Cancel

「Install」をクリックすると、インストールが始まります、インストール完了までに何分がかかります、完了までにお待ちください。

「Finish」をクリックし、OpenOCD のインストールが完了です。







株式会社日昇テクノロジー

4.2.3 「yagarto」をインストール





🕞 YAGARTO 4.3.3 Setup		
Ple	ense Agreement ease review the license terms before installing YAGARTO 4.3.3.	
Press Page Down to see the res	t of the agreement.	
GNU GENERAL PUB Version 3, 29 Jun Copyright (C) 2007 Free Softw Everyone is permitted to copy of this license document, but d	ILIC LICENSE the 2007 vare Foundation, Inc. < <u>http://fsf.org/</u> > and distribute verbatim copies hanging it is not allowed.	
Preamble	~	
If you accept the terms of the agreement, select the first option below. You must accept the agreement to install YAGARTO 4.3.3. Click Next to continue.		
YAGARTO 4,3,3 - 20090311	< <u>B</u> ack <u>N</u> ext > Cancel	
🕞 YAGARTO 4.3.3 Setup		
-1		
Cha Ch	oose Components noose which features of YAGARTO 4.3.3 you want to install.	
Check the components you wan install. Click Next to continue.	t to install and uncheck the components you don't want to	
Check the components you wan install. Click Next to continue. Select components to install:	Cose Components hoose which features of YAGARTO 4.3.3 you want to install. t to install and uncheck the components you don't want to Binutils-2.19.1 Newlib-1.17.0 GCC-4.3.3 GDB-6.8.50 Add YAGARTO to the PATH variable	
Check the components you wan install. Click Next to continue. Select components to install: Space required: 288.4MB	Dose Components noose which features of YAGARTO 4.3.3 you want to install. t to install and uncheck the components you don't want to Binutils-2.19.1 Newlib-1.17.0 GCC-4.3.3 GDB-6.8.50 Add YAGARTO to the PATH variable	



🕞 YAGARTO 4.3.3 Setup 📃 🗆 🔀	
	Choose Install Location Choose the folder in which to install YAGARTO 4.3.3.
Setup will install YAGARTC Browse and select anothe	94.3.3 in the following folder. To install in a different folder, click r folder. Click Next to continue.
Destination Folder	¥ Browse
Space required: 288.4MB Space available: 1.7GB	
YAGARTO 4,3,3 - 20090311 -	< <u>B</u> ack <u>N</u> ext > Cancel



「Install」を押下、インストール完了まで何分がかかります。

🕞 YAGARTO 4.3.3 Se	tup	_ 🗆 🛛
6	Choose Start Menu Folder Choose a Start Menu folder for the YAGARTO 4.3.3 sho	rtcuts.
Select the Start Menu fol can also enter a name to	der in which you would like to create the program's shortcuts create a new folder.	:. You
YAGARTO		
@niftyでインターネット 91/13サ暗擦引 Adult PDF Password Rec Alive Video Converter CodeSourcery Cygwin DivX EPSON EPSON EPSON Scan FXOnline Toolbar ICBC Personal Internet B	overy Banking Certificate Tools	
🔲 Do not create shortcu	its	
YAGARTO 4.3.3 - 20090311		
	< <u>B</u> ack Install	Cancel



4.2.4 「yagarto-tools」をインストール

🕞 YAGARTO Tools 2007)303 Setup 📃 🗆 🔀					
	Welcome to the YAGARTO Tools 20070303 Setup Wizard This wizard will guide you through the installation of YAGARTO Tools 20070303. It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer. Click Next to continue.					
	Next > Cancel					
🕞 YAGARTO Tools 20070303 Setup						
	cense Agreement Please review the license terms before installing YAGARTO Tools 20070303.					
Press Page Down to see the re	cense Agreement Please review the license terms before installing YAGARTO Tools 20070303. est of the agreement.					
Press Page Down to see the re GNU GR Version	cense Agreement Please review the license terms before installing YAGARTO Tools 20070303. est of the agreement. ENERAL PUBLIC LICENSE on 2, June 1991					
Press Page Down to see the re GNU Ge Versi Copyright (C) 1989, 1991 Fre 51 Franklin Street, Fifth File Everyone is permitted to cop of this license document, but	cense Agreement Please review the license terms before installing YAGARTO Tools 20070303. est of the agreement. ENERAL PUBLIC LICENSE on 2, June 1991 ese Software Foundation, Inc. bor, Boston, MA 02110-1301 USA y and distribute verbatim copies changing it is not allowed.					
Copyright (C) 1989, 1991 Fro 51 Franklin Street, Fifth File Everyone is permitted to cop of this license document, but	cense Agreement Please review the license terms before installing YAGARTO Tools 20070303. est of the agreement. ENERAL PUBLIC LICENSE on 2, June 1991 ese Software Foundation, Inc. bor, Boston, MA 02110-1301 USA y and distribute verbatim copies changing it is not allowed. Preamble					
Press Page Down to see the re GNU Ge Versi Copyright (C) 1989, 1991 Fre 51 Franklin Street, Fifth Fil Everyone is permitted to cop of this license document, but	cense Agreement Please review the license terms before installing YAGARTO Tools 20070303. est of the agreement. ENERAL PUBLIC LICENSE on 2, June 1991 ese Software Foundation, Inc. bor, Boston, MA 02110-1301 USA y and distribute verbatim copies changing it is not allowed. Preamble agreement, select the first option below. You must accept the Tools 20070303. Click Next to continue.					
Li Press Page Down to see the read GNU GE Versid Copyright (C) 1989, 1991 Free S1 Franklin Street, Fifth Fli Everyone is permitted to cop of this license document, but If you accept the terms of the agreement to install YAGARTCO I accept the terms in the L These accent the terms	cense Agreement Please review the license terms before installing YAGARTO Tools 20070303. est of the agreement. ENERAL PUBLIC LICENSE on 2, June 1991 ese Software Foundation, Inc. oor, Boston, MA 02110-1301 USA y and distribute verbatim copies changing it is not allowed. Preamble agreement, select the first option below. You must accept the Tools 20070303. Click Next to continue.					
Link Press Page Down to see the ready Press Page Down to see the ready GNU GE Version Copyright (C) 1989, 1991 Free S1 Franklin Street, Fifth Flu Everyone is permitted to cop of this license document, but If you accept the terms of the agreement to install YAGARTO I accept the terms in the Link YAGARTO Tools 20070303	cense Agreement Please review the license terms before installing YAGARTO Tools 20070303. est of the agreement. ENERAL PUBLIC LICENSE on 2, June 1991 ese Software Foundation, Inc. bor, Boston, MA 02110-1301 USA y and distribute verbatim copies changing it is not allowed. Preamble agreement, select the first option below. You must accept the Tools 20070303. Click Next to continue. cense Agreement n the License Agreement					



🕞 YAGARTO Tools 20070	303 Setup 📃 🗌 🔀				
	oose Components hoose which features of YAGARTO Tools 20070303 you want to stall.				
Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.					
Select components to install:	Make utils Add the make utils to the PATH variable				
Space required: 1.6MB	Description Position your mouse over a component to see its description.				
VAGARTO Tools 20070303	< Back Next > Cancel				
🕞 YAGARTO Tools 20070	303 Setup 📃 🗆 🔀				
	oose Install Location hoose the folder in which to install YAGARTO Tools 20070303.				
Setup will install YAGARTO Tools 20070303 in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.					
Destination Folder					
Destination Folder	Browse				
Destination Folder D:¥embeded¥yagarto-tools Space required: 1.6MB Space available: 1.7GB	Browse				
Destination Folder D:¥embeded¥yagarto-tools Space required: 1.6MB Space available: 1.7GB YAGARTO Tools 20070303	Browse				



「Install」を押下、インストール完了まで何分がかかります。

🐨 YAGARTO Tools 20070303 Setup 📃 🗆 🔀				
6	Choose Start Menu Folder Choose a Start Menu folder for the YAGARTO Tools 2 shortcuts.	20070303		
Select the Start Menu fol can also enter a name to	der in which you would like to create the program's shortco create a new folder.	uts. You		
YAGARTO Tools				
@niftyでインターネット 91/1日サ暗摠日 Adult PDF Password Red Alive Video Converter CodeSourcery Cygwin DivX EPSON EPSON Scan FXOnline Toolbar FXOnline Toolbar	overy Banking Certificate Tools			
🔲 Do not create shortci	its			
YAGARTO Tools 20070303 -	< <u>B</u> ack Install	Cancel		



4.2.5 Cygwin のダウンロード&インストール

1 インストール

a) setup.exe の DL&起動 <u>http://cygwin.com/</u>より <u>setpu.exe</u>をダウンロードし起動てください。 直接ダウンロードしたファイルを起動しても OK。 以下に注意すべき画面を中心に解説します(より詳しい手順は Cygwin のページを参照したください)。

b) テキストファイル形式の選択

以下の「Default Text File Type」に「<u>Unix</u>」を選びます(テキストファイルが UNIX 形式(改行が LF)となりま す)。また Cygwin をインストールするマシンをサーバーとして利用したい場合には、「Install For」で「<u>All Users</u>」 を選びます。





c)パッケージ選択

パッケージ選択の画面にて Category の「All」を「Install」に変更する。この変更によって全パッケージがインストールさ れることになる

		C Keep	C <u>P</u> rev	Θĝ	unj C	Е⊻р	View Ca	ategory
Category	Curre	New			Bi	Sr	Package	^
• All 🚯 Install								
+ Admin 🚯 Ir	nstall							
+ Archive 😯	Install							-
+ Base 🚯 Ins	stall							
+ Database 🌢) Install							
+ Devel 🚯 In	stall							
+ Doc 🚯 Inst	all							
+ Editors 🚯 I	nstall							~
								>

上記の設定が終われば「次へ」をクリックすれば、ダウンロード&インストールが始まる。



2 インストール後の設定

a)ホームディレクトリの作成

通常、デスクトップにできた Cygwin アイコンをクリックすると、/usr/username がホームディレクトリになるが、Windows の環境変数 HOME が設定されていれば、Cygwin においてもそこがホームディレクトリとなる。この問題を避けるために、コントロールパネルの「システム」を起動し、「詳細設定」タブの下にある「環境変数」をクリックする。そしてユーザ環境変数に HOME 変数が設定 されていれば削除する。

ここでデスクトップの Cygwin アイコンをクリックすればホームディレクトリが自動的に作成される。

次にホームディレクトリの下に各種設定ファイルを編集していく。利用する漢字コードによって設定方法が異なるので、以下では SJIS 環 境を説明する。設定するファイルは、.bash_profile、.bashrc、.inputrc、.vimrc の4つで(全てのファイルはピリ オドから始 まるファイルで隠しファイルとなっており、"Is -a" にて確認できる)、以下に示した通りに設定する。既に存在するファイルについ ては追加し、存在しないファイルについては新規作成する。またファイルの編集 は <u>TeraPad</u>などのエディタを利用すれば便利。 注意点として、新規ファイルを保存する際には、「ファイル」→「漢字・改行コードを指定保存」を選び、改行コードに「LF」を選んで 保存してください。

b)SJIS 環境に関わる4つファイルの設定

i).bash_profile

export TERM=vt100 export LANG=ja_JP.SJIS export LESSCHARSET=japanese-sjis

ii).bashrc

alias ls='ls --show-control-chars --color=auto'

alias ct='cygterm &'

function ie {

echo /cygdrive/c/Program¥ Files/Internet¥ Explorer/IEXPLORE "\$(cygpath -w \$PWD)¥¥\$1" /cygdrive/c/Program¥ Files/Internet¥ Explorer/IEXPLORE "\$(cygpath -w \$PWD)¥¥\$1" & 1

上記の ie 関数を設定しておくことにより、XML や XMLtable ファイルをコマンド上から起動可能になる。 例) ie dat.xt #dat.xt を InternetExplore を利用して表示する。

iii).inputrc

set convert-meta off

set input-meta on

set output-meta on

iv).vimrc

set encoding=japan set fileencodings=sjis



4.2.6 「jre」をインストール

一般のソフトウエアと同じようにインストールします。もしバージョン 1.4.2 以上の JRE を既にパソコンにインストールされたら、 インストールが不要です。

「06.jre-6u7-windows-i586-p.exe」あるいはダウンロードした JRE インストールファイルをクリックし、「D:¥Embeded¥JRE」にインストールする

4.2.7 「Eclipse」をインストール

「07.eclipse-cpp-galileo-win32.zip」あるいはダウンロードしたファイルを「D:¥Embeded」解凍



4.3 ソフトウエアインストール後の動作確認

確認方法:「スタート」→「ファイル名を指定して実行」→「cmd」を入力

4.3.1 OpenOCD の確認

確認コマンド: openocd -v C:¥WINDOWS¥system32¥cmd.exe - 🗆 🗙 Microsoft Windows XP [Version 5.1.2600] (C) Copyright 1985-2001 Microsoft Corp. C:¥Documents and Settings¥dragonwake>openocd -v pen On-Chip Debugger 0.1.0 (2009-01-21-21:15) Kelease BUGS? Read http://svn.berlios.de/svnroot/repos/openocd/trunk/BUGS \$URL: https://kc8apf@svn.berlios.de/svnroot/repos/openocd/tags/openocd-0.1.0/src /openocd.c \$ C:¥Documents and Settings¥dragonwake>_



4.3.2 コンパイラ確認

🖾 C:¥WINDOWS¥system32¥cmd.exe	- 🗆 🗙
Microsoft Windows XP [Version 5.1.2600] (C) Copyright 1985-2001 Microsoft Corp.	
C:¥Documents and Settings¥dragonwake>java -version java version "1.5.0_17" Java(TM) 2 Runtime Environment, Standard Edition (build 1.5.0_17-b04) Java HotSpot(TM) Client VM (build 1.5.0_17-b04, mixed mode)	
C: YDocuments and Settings ¥dragonwake>makeversion GNU Make 3.81 Copyright (C) 2006 Free Software Foundation, Inc. This is free software; see the source for copying conditions. There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.	
This program built for i686-pc-mingw32	
C:¥Documents and Settings¥dragonwake>	
	Microsoft Windows XP [Version 5.1.2600] (C) Copyright 1985-2001 Microsoft Corp. C:¥Documents and Settings¥dragonwake>java -version java version "1.5.0_17" Java(TM) 2 Runtime Environment, Standard Edition (build 1.5.0_17-b04) Java HotSpot(TM) Client VM (build 1.5.0_17-b04, mixed mode) C:¥Documents and Settings¥dragonwake>makeversion GNU Make 3.81 Copyright (C) 2006 Free Software Foundation, Inc. This is free software; see the source for copying conditions. There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. This program built for i686-pc-mingw32 C:¥Documents and Settings¥dragonwake>_





	2 : ARM EABI [「] arm=none=eabi=gcc」確認 確認コマンド: arm=none=eabi=gcc ==version		
	C:¥WINDOWS¥system32¥cmd.exe	- 🗆	×
	Microsoft Windows XP [Version 5.1.2600] (C) Copyright 1985-2001 Microsoft Corp.		•
<	C:¥Documents and Settings¥dragonwake>arm-none-eabi-gccversion arm-none-eabi-gcc (Sourcery G++ Lite 2009q1-161) 4.3.3 Copyright (C) 2008 Free Software Foundation, Inc. This is free software; see the source for copying conditions. There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.		
	C:¥Documents and Settings¥dragonwake>		



3:ARM コンパイラ「arm-elf-gcc」確認

確認コマンド: arm-elf-gcc --version

C:#WINDOWS¥system32¥cmd.exe	- 🗆 🗙
Microsoft Windows XP [Version 5.1.2600] (C) Copyright 1985-2001 Microsoft Corp.	<u>^</u>
C:¥Documents and Settings¥dragonwake>openocd -v Open On-Chip Debugger 0.1.0 (2009-01-21-21:15) Release	
BUGS? Read http://svn.berlios.de/svnroot/repos/openocd/trunk/BUGS	
\$URL: https://kc8apf@svn.berlios.de/svnroot/repos/openocd/tags/openocd-0.1.0 /openocd.c \$)/src
C:¥Documents and Settings¥dragonwake>arm-elf-gccversion arm-elf-gcc (GCC) 4.3.3 Conversitet (C) 2002 Free Settinger Regulation - Inc.	
This is free software; see the source for copying conditions. There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.	
C:¥Documents and Settings¥dragonwake>	
	-





4.JRE バージョン確認:

確認コマンド



第五章 Eclipse の設定

5.1 Eclipse を起動する



最初に Workspace の場所を聞いてきます。デフォルトは名前が長いので、"D:¥embeded¥eclipse¥workspace"に変更しました。

Workspace Launcher	
Select a workspace	
Eclipse stores your projects in a folder called a workspace. Choose a workspace folder to use for this session.	
Workspace: D:¥embeded¥eclpise¥workspace	Browse
▶ <u>C</u> opy Settings	
5	
?	OK Cancel


画面の Workbench をクリックします。



注意:サンプルを簡単に導入できるように、ダウンロードして前章で解凍してできた

「D:¥OpenJTAG¥gcc_openjtag_eclipse_arm_example」の中の全てのフォルダーを「D:¥embeded¥eclipse¥workspace」へコピーして ください。



5.2 Eclipse プラグイン(Zylin Embedded CDT)インストール

- * Zylin Embedded CDT の 4.81 バージョンは Zylin のホームページから取り下げますので、 弊社から圧縮ファイルをダウンロードしてから Eclipse にコピーしてください。
- * Zylin Embedded CDT の最新版では検証していないため、インストールしないでください。

Zylin Embedded CDT ダウンロード URL:

http://www.dragonwake.com/download/open-jtag/com.zylin.embeddedcdt_4.8.1.zip

上記ファイルを解凍し「Eclispe インストール先」の「plugins」というフォルダーにコピーしてください。 コピーが完了したら、Eclipse を再起動させます。



5.3 ビルドの設定

Eclipse の"Window"→"Preferences"を選択し、

Preferences の"C/C++ Build"→"Settings"を選択し"Binary Parsers"タブの"GNU Elf Parser"にチェックを入れて OK ボタンを押 します





5.4 OpenOCD の設定

1. Eclipse の"Run"→"External Tools."→"External Tools Configrurations..."を選択します。





2. External Tools Configrurations の"Program"を右クリックし、"New"を選択します。

🚝 External Tools Configuratio	ons	×
Create, manage, and ru Run a program	in configurations	
Filter matched 1 of 1 items	Configure launch settings from this dialog:	
0	Eun	Close



- 3. Main タブの "Name" に適当な名前を入力してください、例えば、"OpenOCD"。
- "Location:"IC"D:¥embeded¥openocd¥bin¥openocd.exe"、
- "Working Directory:"IC" D:¥embeded¥eclipse¥workspace "
- "Arguments:"IC-f"D:¥embeded¥elipse¥worksapce¥config¥interface¥open-jtag.cfg"
 - ーf ″D:¥embeded¥elipse¥worksapce¥config¥target¥lpc2388.cfg″と入力します。
- * 基板種類(lpc2148,lpc2388,mini2440 など)により、Arguments の設定も異なる。

External Tools Configurations	
Create, manage, and run config	urations
Run a program	
	Name OpenOCD
type filter text	The Main Refresh Build To Environment Common
Ring Build	Location: D:¥embeded¥openocd¥bin¥openocd.exe
	Browse Workspace Browse File System Variables
	Working Directory:
	D:¥embeded¥eclipse¥workspace
	Browse Workspace Browse File System Variables
	Arguments:
C	-f "D:¥embedded¥elipse¥worksapce¥config¥interface¥open-jtag.cfg" -f "D:¥embedded¥elipse¥worksapce¥config¥target¥lpc2388.cfg"
	Variables
	Note: Enclose an argument containing spaces using double-quotes (").
Filter matched 3 of 3 items	Apply Re <u>v</u> ert
?	Run Close



🖨 External Tools Configurat	ions 🔀
Create, manage, and run con Run a program	figurations Organizations
type filter text	Mare: OpenOCD Main Refrest Build Environment Qommon Eulid before launch The gntire workspace The groject containing the selected resource Specific projects Projects. Include referenced projects Projects Projects
Filter matched 2 of 2 items	nppiy reger
•	<u>R</u> un Close

Build タブをクリックし"Build before launch"にチェックを外れます。全てを入力し終えたら"Apply"ボタンを押し、"Close"ボタンを 押します。



5.5 Cygwin のソース・ルックアップ・パスの設定

GDB でデバッグを開始すると「Can't find a source file at "/cygdrive/c/project/hoge/hoge.cpp"」なんて怒られる。Cygwin のパス を eclipse が認識できていない様 子。この場合は、ルックアップ・パスの設定が必要。

eclipse のメニュー[ウィンドウ]→[設定]で設定ダイアログを開き、[C/C++]→[Debug]→[Common Source Lookup path]にルッ ク アップ設定を追加する。

🤿 Preferences		_ 🗆 🔀
type filter text	Common Source Lookup Path	← - ⇒ -
General C/C++	Common S <u>o</u> urce Lookup Path:	
Appearance Build Console Build Variables Code Style Debug Breakpoint Actions Common Source Lookup Path Debugger Types GDB MI Editor Environment File Types Indexer Language Mappings Make New CDT project wizard Property Pages Settings Task Tags Template Default Values Help Install/Update Run/Debug Tasks		Add <u>E</u> dit Remove Up Down
⊕- Usage Data Collector	Restore Defa	ults <u>A</u> pply
?	ОК	Cancel



「Path Mapping」を選択、「OK」を押下

😂 Add Source		
Select the type of source to add to	the source lookup path	
A path mapping.		
🕆 Absolute File Path		
E File System Directory		
🤁 Workspace		
🗁 Workspace Folder		
?	OK Cancel	
مطرية م السرم [Compilation path الت[/	averdaine /a / L El agal fila avertam path L	に「。」な指字する
Rath Mannings	cygunve/c/j, i Locar me system patrij	[]を指定する
Chocify the manning nathe		
specity the mapping paths		11
Name: New Mapping		
Compilation path:	Local file system path:	Add
#cygdrive¥c¥	C:¥	Remove
		Up
		Down
?	ОК	Cancel



「Add」をクリック、他のローカルドライブを同じように追加する。最後、「OK」ボタンを押下

🖉 Path Mappings			
Specify the mapping paths			11
Name: New Mapping			
Compilation path:	Local file system path:		Add
K K K K K K K K K K K K K K K K K K K	C:¥		Remove
Kouzdrine¥d¥			
E state stat	L.Ŧ		
			Down
l			
<		>	
?		ОК	Cancel



5.6 Eclipse デバッグ用のコマンド及びショットカットキー一覧

詳しくは Eclipse に関わるドキュメントを参照



再開(F8) 終了 ステップイン(F5) ステップ・リターン(F7) ステップ実行において良く使われる操作の一覧を以下に示します。

操作名	ショートカットキー
再開	F8
ステップイン	F5
ステップオーバー	F6
ステップ・リターン	F7

ステップ実行とは関係ありませんが、前回起動したクラスを再度実行したりデバッグする場合 は、以下のショートカットキーが便利です。

操作名	ショートカットキー
前回の起動を実行	Ctrl + F11
前回の起動をデバッグ	F11



第六章 ARM シリーズデバッグ手順

*本書に使われる ARM シリーズは全て弊社の製品となります。

詳しくは弊社の通販サイト:<u>http://www.csun.co.jp</u>までにご参照ください。

第四章「インストール手順」、第五章「Eclipse の設定」に従って行っていれば、ARM シリーズ、例えば、ARM7(LPC2148、 LPC2388)、Contex-M3(STM32F103)、ARM9(MINI2440)のサンプルを導入のうえ、デバッグ手順を説明させて頂きます。

6.1 ARM7 の LPC2148

6.1.1 LPC2148 ボード購入 URL

http://www.csun.co.jp/SHOP/2009071209.html

ボード URL から LPC2148 マニュアルを参照し、Open-JTAG に接続してください。

LPC2148 マニュアルダウンロード URL :

http://www.dragonwake.com/download/LPC2148/LPC2148_manual.pdf

6.1.2 ハードウエア動作確認



- 1. OpenJTAG をパソコンの USB ポートに挿入する
- 2. JTAG ケーブルで OpenJTAG と LPC2148 ボードを繋ぐ
- 3. LPC2148 ボードに電源を入れる
- 4. 下記のコマンドを入力します。

openocd -f ["]D:¥embeded¥eclipse¥workspace¥config¥interface¥open-jtag.cfg["] ["]D:¥embeded¥eclipse¥workspace¥config¥target¥lpc2148.cfg["] はじめのーfは OpenJTAG のコンフィグファイルを使います。二番目のーf は lpc2148 のコンフィグファイルを使います。 -f



6.1.3 LPC2148 用のサンプル「LED」をデバッグ

サンプルダウンロード URL:

http://www.dragonwake.com/download/LPC2148/Example-2148GCC.rar

*サンプルの導入について、二つ方法があります、一つは新規プロジェクトを作成してからソースを導入、もう一つは ダウンロードプロジェクトをインポートします。

*ここに新規プロジェクトの作成として説明

1. プロジェクト「LPC2148_LED」を作る

a)新規プロジェクトを作成するため"File"→"New"→"C Project"を選択します

⇒C/C++ - LPC2388_LED/makefile - Eclipse				
<u>File Edit Source Refactor I</u>	<u>N</u> avigate Se <u>a</u> r	rch <u>P</u> roject <u>R</u> un <u>W</u> indow <u>H</u> elp		
<u>N</u> ew Al	lt+Shift+N 🔸	A Project	👟 🚓 🛷 • 🗄 🖬 🖷 🗄 🖓 • 🖓 • 🕅	
Open File _e .		😭 C Project		
Close Ct	:trl+\\/	Project	<u> </u>	
Close All Ct	trl+Shift+W	Convert to a C/C++ Project	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	
	6	Source Folder	is makefile with an editor wh	
Save Ot	trl+S	Solder	in maneric wrom an earoor wh	
Save <u>A</u> s		r Source File		
Sav <u>e</u> All Ot	trl+Shift+S	🔐 Header File		
Kever <u>t</u>		✤ File from Template		
Mo <u>v</u> e	Č	😚 Class		
Rena <u>m</u> e F2	2	• Other Ctvl+N		
🐑 Re <u>f</u> resh 🛛 🛛 🖓	5 L		t files.	
Con <u>v</u> ert Line Delimiters To		# # To robuild project do "mo	to alcont and their allt	
Print Ot	itrl+P	# TO FEDULIA PROJECT AD ~ma	ake clean" and "make all".	
Suitet Westerree				
Switch workspace	· · · · · · · · · · · · · · · · · · ·	******	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	
		# Start of default section #		
🚵 Import		7		
Export		TRGT = arm-elf-		
P <u>r</u> operties Al	lt+Enter	CC = \$ (TRGT) gcc		
1 makafila [LBC2200 LED]		CP = \$ (TRGT) objcopy	ler_with_con	
2 Makefile [Keil GCC/ADC]		BIN = $(CP) - O$ ihex	Jier-wich-opp	
3 adoc [ADC]				
4 Makefile [ADC]		MCU = arm7tdmi	1	
E <u>x</u> it		🖹 Problems 🖉 Tasks 📮 Console 🖾 🔪	Properties	
		C-Build [LPC2388_LED]		
		**** Build of configuration D	efault for project LPC2388 LEI	
			···· •································	
	:	make all		
		make: Nothing to be done for	`all'.	
: L 💓 LP02388_LED				



b) 適当なプロジェク	ト名前(LPC2148	_LED)を入力し	Finish ボタン	~を押します。
-------------	-------------	-----------	------------	---------

ar C Project	
C Project Create C project of selected type	
Project name LPC2148_LED	
Use <u>d</u> efault location	
Location: D:¥embeded¥eclipse¥w	vorkspace¥lpc2148¥LPC2148_LED Browse
Project type:	Toolchains:
⊟	Other Toolchain
Show project types and toolch	nains only if they are supported on the platform
?	< <u>Back</u> <u>N</u> ext > <u>F</u> inish Cancel



c)サンプルソースを作成されたプロジェクトにインポート Project Explorer の「LPC2148_LED」を選べ、右クリックし、「Import」を押下





d)「File System」を選択し、「Next」をクリック 1

🖨 Import	
Select Import resources from the local file system into an existing project.	Ľ
Select an import source:	
General Cristing Projects into Workspace File System Professore Professore C/C++ C/S Run/Debug Tasks Team	
? < <u>Back</u> <u>Next</u> > <u>Finish</u>	Cancel



e) ダウンロードしたサンプル LPC2148_LED を選択





f) ソースファイルと Makefile を導入

Import	
File system Import resources from the local file system.	
From directory: D:¥embeded¥eclipse¥workspace¥lpc2148¥Example-2148GCC¥1. LED	
■ 🧁 1. LED	 depend LED.c LED.dmp LED.elf LED.hex LED.map LED.o Makefile
Filter Types Select All	
Into folder: LPC2148_LED Options Overwrite existing resources without warning Ocreate complete folder structure Ocreate selected folders only	
ext > <u>Back</u>	Cancel



g) 共通のソースも導入(他の操作は上記 c)、d)と同じ)

Import	
File system Import resources from the local file system.	
From directory: D:¥embeded¥eclipse¥workspace¥lpc2148¥Example-2148GCC > >	 LPC2141_42_44_46_48_4.pdf LPC2148_manual.pdf Z213x4xSch.pdf
Filter Types Select All Into folder: LPC2148_LED	
Options Overwrite existing resources without warning Oreate complete folder structure Create selected folders only	
Image: Second	Finish Cancel



h)サンプル LPC2148_LED をビルド





株式会社日昇テクノロジー

i)ビルドエラーメッセージ(エラーメッセージがなければ、i)を飛ばしてi)を直接参照)





株式会社日昇テクノロジー

j)「common」こういうソースのパスを「Makefile」に修正 $\lceil Edit \rfloor \rightarrow \lceil Find/Replace \rfloor$ 「Find」:「../common」を入力、「Replace with」:「./common」を入力、「Replace All」をクリック 置き換えの後、「File」→「Save」をクリック - 0 23 C/C++ - LPC2148_LED/Makefile - Eclipse File Edit Source Refactor Navigate Search Run Project Window Help 📸 = 🛍 = 🔂 = 🧭 = 🐔 = 🗞 = 🔅 = 🗘 = 🖓 = 📑 🔻 🔚 🖻 🖬 😭 🏇 Debug 🔂 C/C- 💥 🔳 🛐 👘 🖢 ▾ 🏳 ▾ ⇔ ▾ ⇔ ▾ 🗖 🗖 🚺 🗈 LED.c 🔊 *Makefile 🛛 💦 - 8) 陷 Project Explorer 🐒 🗄 Outli 🖾 Mak [🚡 Makefile §9 ▽ \bigtriangledown 日 🕼 # Tool definitions °a ↓a cc = arm-elf-gcc 🞏 22.nRF24L01 CC 🔄 = arm-elf-ld -v LDLPC2148_IIC LD AR х Find/Replace EPC2148_LED AS CP 🔊 Includes OD Find: ../common • 👝 common RM = LED.c Replace with: ./common Ŧ LED.dmp # Tool fl 🗟 LED.elf CFLAGS Direction Scope FLAGS ASFLAGS SFLAGS LED.hex Forward All LFLAGS FLAGS LED.map Backward Selected lines CPFLAGS LED.0 PFLAGS ODFLAGS Options 🚡 Makefile DFLAGS Case sensitive Wrap search LINKFILE INKFILE Whole word Incremental SRCS CSRCS Regular expressions BJS OBJS XAMPLES EXAMPLES Find Replace/Find • ь Replace Replace All 🚼 Problems rch 🕸 Debug 3 matches replaced Close C-Build [LPC2: 🗽 🛃 🖻 👻 📑 🕶 **** Build of configuration Default for project LPC2148 LED **** make all



k)修正後、再度ビルド(赤丸に囲まれるメッセージが出てきれば、ビルドが成功ということを判明)

C/C++ - LPC2148_LED/Makefile - Ec	lipse
<u>File Edit Source Refactor Navigat</u>	re Se <u>a</u> rch <u>R</u> un <u>P</u> roject <u>W</u> indow <u>H</u> elp
□	r 🖸 ▼ 🞯 ▼ 🦓 ▼ 🛞 ▼ 🕸 ▼ 🕼 ▼ 💁 ▼ 🖉 🖉 🖉 ▼ 👔 🕸 Debug 🛅 C/C- [≫] ⇔ ▼
Project Explorer 🕱 🛛 🗖	🗆 🔂 LED.c 🕞 Makefile 🕞 Makefile 🛛 🍡 3 🛛 🗖 🗄 Outli 🛛 🛞 Mak 🖓 🗖
Project Explorer ⋈ Project Explorer ⋈ E ↓ LPC2148_IIC LPC2148_LED Includes common LED.c LED.dmp LED.elf LED.hex LED.map LED.o Makefile	<pre> LED.c Makehle Makehle 3 "3 C Makehle 3 "3 C Makehle 3 Makehle 3 "3 C Makehle 3 Makehle</pre>
	<pre>common/src/crt.o common/src/startup.o LED.o -o LED.elf -Wl,-Map,LED.map arm-elf-objcopy -O ihex LED.elf LED.hex arm-elf-objdump -xsyms LED.elf > LED.dmp</pre>
₩ritable	



- 2. openocd 設定の修正
 - 一般の設定は「5.4 OpenOCD の設定」を参照
 - a)「Working Directory」を「LPC2148_LED」に変更
 - b)「Arguments」を下記のように変更

-f "D:¥embeded¥eclipse¥workspace¥config¥interface¥open-jtag.cfg"

 $"D: \verb"Yembedded" eclipse \verb"Yworkspace" years a config" target \verb"Yembedded" provide the target the target the target target the target target$

🥃 External Tools Configurat	ions 🛛 🛛 📉
Create, manage, and run cor Run a program	nfigurations Organizations
type filter text	Name: openocd Main Refresh Image: Second
Filter matched 2 of 2 items	Apply Revert
?	Close



3. LPC2148_LED 用の GDB を追加

a)アイコン「Debug」をクリック、「Debug Configuration」を押下





b) 「Zylin Embedded debug(Cygwin)」を右クリック、「New」を選択





c)「Name」」を入力、「Project」、「C/C++ Application」(elf ファイル指定)を指定

Debug Configurations	$\overline{\mathbf{X}}$
Create, manage, and run con	igurations 🔅
Image: Second system Image: Second system Image: Secon	Name: LPC2148_LED
Filter matched 11 of 11 items	Apply Revert
?	Debug Close



d) 「GDB debugger」を「arm -elf-gdb.exe」の場所に指定

🌲 Debug Configurations	
Create, manage, and run con	figurations 🗸
Image: Second system Image: Second system Image: Secon	Name: LPC2148_LED Main Debugger Commands Common Debugger: Embedded GDB Advanced Debugger: D*embeded#yagarto#bin¥arm-elf-gdb.exe Browse GDB debugger: D*embeded#yagarto#bin¥arm-elf-gdb.exe Browse GDB command Estimate Browse Warning: Some commands in this file may interfere with the startup operation of the debugger, for example "run". GDB command set: Standard Protocol: mi Imit is full file path to set breakpoints
Filter matched 11 of 11 items	Apply Revert
?	Debug Close



株式会社日昇テクノロジー

e)「Commands」を入力(//の右はコメントです、Eclipeにコピーしないでください。)
 target remote localhost:3333 // ローカルポート「3333」と接続(OpenOCDと接続)
 monitor halt//ボードの実行を中断させる
 monitor step//ステップで実行するように
 load // leds_elfをロード,「elf」というフォーマットのファイルにアドレスを含まれる
 break main // 「main」 関数にブレークポイントを設定
 continue // プログラムを実行させて、「main」にとまってステップでデバッグ可能

$\overline{\mathbf{X}}$ Debug Configurations Create, manage, and run configurations 🖹 📄 🗶 🕞 🍰 -Name: LPC2148_LED type filter text 📄 Main (🏇 Debugger (📶 Commands 🔪 🧤 Source) 🔲 Common 🗋 C/C++ Application or tips on how to setup GDB init s C/C++ Attach to Applica. 'Initialize' commands ---- C/C++ Postmortem Debu target remote localhost:3333 - 🐌 Launch Group monitor halt 🖮 🗊 Zylin Embedded debug 🚺 monitor step load break main continue 'Run' commands < 1111 Apply Revert Filter matched 11 of 11 items ? <u>D</u>ebug Close



4. デバッグ

a) OpenOCD を起動

External Tools の▼ボタンをクリックし、OpenOCD を選択



b)Console ウインドに下記のメッセージが出力

 Image: Second second

c)Debug の▼ボタンをクリックし、"LPC2148_LED"を選択。





d)「main」関数に止まる

🖨 Debug - LED/LED.c - Eclipse					
<u>File E</u> dit <u>S</u> ource Refac <u>t</u> or <u>N</u> avigate Se <u>a</u> rch <u>R</u> un <u>P</u> roject <u>W</u> indow <u>H</u> elp					
i 📬 • 🔜 🖆 📾 i 🏇 • 🔕 • 9 💁 🥔 🖉 • 1 🍠 i 🖉 - 💱 +	* 🗘 *	😭 🐝 Debug 🔤 C/C++			
🏇 Debug 🛛 🗖 🗖	🖓 🗖 🕬= Variables 🕱 💊 Breakpoints 👯 Registers 📑 Modules 🔗 🖓 🗖				
🎽 🤹 🔅 🖬 🚸 🖿 🗉 🔳 🖓 🔍 🖘 👘 😵		約 🐗 🕞 🍫 💕 💥 💥 🏹			
- 🎯 Embedded GDB (09/07/12 21:22) (Suspended)	Name	Value			
terminated, exit value: −1>C:¥Program Files¥yagarto¥bin¥arm-elf-gdb.exe (09/07	(x)= i	0			
Green COD (Program)					
a citritigiani nastoponocotto nobintoponocotto					
🖨 🔐 Embedded GDB (09/07/12 21:23) (Suspended)					
😑 🝿 Thread [0] (Suspended)					
= 1 main0 C:¥zero¥Example-2148GCC¥LED¥LED.c:9 0x00000168		<u>^</u>			
C¥Program Files¥yagarto¥bin¥arm-elf-gdb.exe (09/07/12 21:23)	2	<u> </u>			
LED.c 🕱		🗄 Outline 🚮 Disassembly 🛛 🗌			
<pre>#include <lpc214x.h></lpc214x.h></pre>		▽			
// LED P0.31		• 0x00000168 <main+12>: ldi</main+12>			
#define LED 0x8000000		0x0000016c <main+16>: su</main+16>			
int main(woid) (0x00000170 <main+20>: mov</main+20>			
volatile int i:		0x00000174 <main+24>: mov</main+24>			
IODIRO = LED:	0x00000178 <main+28>: sti</main+28>				
		0x0000017c <main+32>: b</main+32>			
<pre>for(i = 0; i < 100000; i ++);</pre>		0x00000180 <main+36>: 1d)</main+36>			
IOSETO = LED;		0x00000104 (main+40): au			
101(1 0, 1 < 100000; 1 ++);		0x0000018c <main+48>: 1di</main+48>			
IOCLRO = LED;	24	0x00000190 <main+52>: cmj</main+52>			
Concele C Taska R Brahlama C Every dahlar R Mamanu					
adb [Zulin Embedded debug (Natiue)] C#Program Files#usgarts#bin#arm-alf-adb eve (09/07/12)	1-22)				
Note: automatically using hardware breakpoints for read-only a	dresses.	~			
······································		_			
Breakpoint 1, main () at LED.c:9					
9 for(i = 0; i < 100000; i ++);					
mi_cmd_disassemble: Invalid filename.					
i □ [◆]		🐴 🥥 🗷 💖 🔶			



e) ブレークポイントでプログラムが中断した状態から、次のブレークポイントまで実行させたり、1行ずつ実行させたりできます。コードを繰り返して実行することにより、LED ランプが点滅します。





- 5. デバッグ終了
- a) gdb の停止 Debug ウインドウの gdb の Thread を選択し、停止ボタンと押します 🚝 Debug - Source not found. - Eclipse Platform <u>File Edit Navigate Search Project Run Window Help</u> 📑 • 🖫 👜 🏇 • 🕥 • 💁 • 😥 🖉 • 🗦 🏄 - 🎘 • 🎋 Debug 🙁 🗋 -0-_i⇒ 5 3 A C Ė∽**≣c** gdb [Zylin Embedded debug (Native)] 🗄 🍻 Embedded GDB (09/03/25 17:31) (Suspended) 🗄 🌈 Thread [1] (Supponded) 1 <symbol is not available> 0x0000000 📕 arm-none-eabi edb (09/03/25 17:01
 - b) OpenOCD の停止

Debug ウインドウの OpenOCD の Thread を選択し、停止ボタンと押します

🛑 De	bug – C	Source not :	found. – E	olipse Pla	atform					
<u>F</u> ile	<u>E</u> dit	<u>N</u> avigate	Se <u>a</u> rch	<u>P</u> roject	<u>R</u> un	<u>W</u> indow	<u>H</u> elp			
] 📬	• 🖁	≞ ≣] 参・	0 • 9	-]	o s	•] ½	- 12 - 9	- 🔶 -	⇒ ~
🕸 D	ebug	x	7%	\$~ IE	(р ф	- to =5 🛛	i⇒ 🕏	▽□□
Ę.	Ъ Оре	enOCD [Pro	graml							
		C:¥Program	i Files¥Op	enOCD¥0	.1.0¥bii	n¥openoco	.exe >			
÷	C edb	[Zylin Emb	edded deb	oug (Nativ	/e)]					
	ം	<terminated< td=""><td>d>Embedd</td><td>ed GDB (</td><td>09/03/</td><td>(25 17:31)</td><td>(Exited.</td><td>Exit code =</td><td>0.)</td><td></td></terminated<>	d>Embedd	ed GDB (09/03/	(25 17:31)	(Exited.	Exit code =	0.)	
		<terminated< td=""><td>d, exit valu</td><td>ıe: O≻arm∙</td><td>-none-</td><td>eabi-gdb</td><td>(09/03/2</td><td>25 17:31)</td><td></td><td></td></terminated<>	d, exit valu	ıe: O≻arm∙	-none-	eabi-gdb	(09/03/2	25 17:31)		

c) 電源停止

ターゲットの電源を停止

- d) OpenJTAG をターゲットから取り外す
- e) 上記が面倒であれば Eclipse を終了しターゲットの電源停止、open-JTAG を取り外しでも OK です。



6.2 ARM7 Ø LPC2388

6.2.1 LPC2388 ボード購入 URL

<u>http://www.csun.co.jp/SHOP/200903021.html</u> ボード URL から LPC2388 マニュアルを参照し、Open-JTAG に接続してください。 LPC2338 マニュアルダウンロード URL : <u>http://www.dragonwake.com/download/LPC2388/LPC2388 manual.pdf</u> **6.2.2 ハードウエア動作確認**



(1). OpenJTAG をパソコンの USB ポートに挿入する

(2). JTAG ケーブルで OpenJTAG と LPC2388 ボードを繋ぐ

(3). LPC2388 ボードに電源を入れる

(4). 下記のコマンドを入力します。

openocd -f " D:¥embeded¥eclipse¥workspace¥config¥interface¥open-jtag.cfg" -f " D:¥embeded¥eclipse¥workspace¥config¥target¥lpc2388.cfg"

※ はじめのfは OpenJTAG のコンフィグファイルを使います。二番目のfは lpc2388 のコンフィ グファイルを使います。



🖾 C:¥WINDOWS¥system32¥cmd.exe – openocd –f D:¥embeded¥eclipse¥workspace¥config 💶 🗙
BUGS? Read http://svn.berlios.de/svnroot/repos/openocd/trunk/BUGS
\$URL: https://kc8apf@svn.berlios.de/svnroot/repos/openocd/tags/openocd-0.1.0/src /openocd.c \$ Can't find D:¥embeded¥eclipse¥workspace¥config¥interface¥open-jtag.cfg-f
C:¥Documents and Settings¥dragonwake>openocd -f D:¥embeded¥eclipse¥workspace¥con fig¥interface¥open-jtag.cfg -f D:¥embeded¥eclipse¥workspace¥config¥target¥lpc238
8.ctg Open On-Chip Debugger 0.1.0 (2009-01-21-21:15) Release
BUGS? Read http://svn.berlios.de/svnroot/repos/openocd/trunk/BUGS
\$URL: https://kc8apf@svn.berlios.de/svnroot/repos/openocd/tags/openocd-0.1.0/src /openocd.c \$ jtag_speed: 0
500 kHz Info : JTAG tap: lpc2388.cpu tap/device found: 0x4f1f0f0f (Manufacturer: 0x787, Part: 0xf1f0, Version: 0x4)
into : JIAG lap/device matched Warn : EmbeddedICE version 7 detected, EmbeddedICE handling might be broken

画面のように"Info: JTAG Tap/device matched"と表示されればOKです (この時点で ARM LPC2388 と通信が出来ています)

(6). その後 CTRL+C を押してデバッグを中止します。

71



6.2.3 LPC2388 用のサンプル「LED」をデバッグ

*サンプルの導入について、二つ方法があります、一つは新規プロジェクトを作成してからソースを導入、もう一つは ダウンロードプロジェクトをインポートします。

*新規プロジェクトの作成方法は「6.1.3 LPC2148用のサンプル「LED」をデバッグ」を参照

*ここに既存プロジェクトのインポートとして説明

1. プロジェクト導入: 「File」 → 「Import」 → 「Existing Projects into workspace」




株式会社日昇テクノロジー





既にダウンロードしたプロジェクトを Workspace に入れていますので、ここに「Copy projects into workspace」を外 します。Workspace に入れていない場合、「Copy projects into workspace」をチェックしてください。

🗦 Import		
Import Projects Select a directory to sear	ch for existing Eclipse projects.	
 Select root directory: Select archive file: Projects: 	D:¥embeded¥eclipse¥workspace¥lpc2388¥Ll	Browse Browse
V LPC2388_LED (D.	¥embeded¥eclipse¥workspace¥lpc2388¥LPC2	Select All
Copy projects into wo	un virkspace	
Working sets	ing sets	S <u>e</u> lect
?	< <u>B</u> ack <u>N</u> ext > <u>F</u> inish	Cancel



2. 導入済みのプロジェクトをコンパイル 「LPC2388_LED」を選択、「Project」→「Build Project」を押下



コンパイルが成功すれば、実行ファイルLPC2388_LED_rom.elfとLPC2388_LED_rom.hexを生成されます。



- 3. openoed 設定の修正
 - 一般の設定は「5.4 OpenOCD の設定」を参照
 - a)「Working Directory」を「LPC2388_LED」に変更
 - b) 「Arguments」を下記のように変更

-f "D:¥embeded¥eclipse¥workspace¥config¥interface¥open-jtag.cfg" "D:¥embeded¥eclipse¥workspace¥config¥target¥lpc2388.cfg"

😂 External Tools Configurat	ions 🛛 🛛 🔀
Create, manage, and run con Run a program	nfigurations Oracle Company of the second seco
type filter text	Name: openocd Image: Main Refresh Refr
Filter matched 2 of 2 items	
?	<u>B</u> un Close



4. LPC2388_LED 用の GDB を追加

①アイコン「Debug」をクリック、「Debug Configuration」を押下





②「Zylin Embedded debug(Cygwin)」を右クリック、「New」を選択





③「Name」」を入力、「Project」、「C/C++ Application」(elf ファイル指定)を指定

😂 Debug Configurations	X
Create, manage, and run cor	nfigurations
Image: Second system Image: Second system <td>Main Image: Commands Common Project (optional): Image: Common Image: Common C/C++ Browse Image: Common Image: Common D*embeded*eclipse*workspace*lpc2388*LPC2388_LED*LPC2388_LED_rol Image: Common Image: Common D*embeded*eclipse*workspace*lpc2388*LPC2388_LED*LPC2388_LED_rol Image: Common Image: Common Image: Common Image: Common Image: Common Image: Common Image: Common Image: Common Image: Common Image: Common Image: Common Image: Common Image: Common Image: Common Image: Common Image: Common<</td>	Main Image: Commands Common Project (optional): Image: Common Image: Common C/C++ Browse Image: Common Image: Common D*embeded*eclipse*workspace*lpc2388*LPC2388_LED*LPC2388_LED_rol Image: Common Image: Common D*embeded*eclipse*workspace*lpc2388*LPC2388_LED*LPC2388_LED_rol Image: Common Image: Common Image: Common Image: Common Image: Common Image: Common Image: Common Image: Common Image: Common Image: Common Image: Common Image: Common Image: Common Image: Common Image: Common Image: Common<
Filter matched 10 of 10 items	Apply Revert
?	Debug Close



④「GDB debugger」を「arm -elf-gdb.exe」の場所に指定

😂 Debug Configurations	×
Create, manage, and run cor	nfigurations
Image: Second state st	Name: LPC2388_LED Main Debugger Commands Debugger: Embedded GDB Advanced Debugger: Debugger Advanced Main GDB debugger: D¥embeded¥yagarto¥bin¥arm-elf-gdb.exe Browse GDB command file Browse Browse Warning: Some commands in this file may interfere with the startup operation of the debugger, for Warning: Some commands in this file may interfere with the startup operation of the debugger, for GDB command set: Standard Protocol: mi Overbose console mode Use full file path to set breakpoints
Filter matched 10 of 10 items	Apply Revert
?	Debug Close



株式会社日昇テクノロジー

⑤「Commands」を入力

target remote localhost:3333 // ローカルポート「3333」と接続(OpenOCDと接続) monitor halt//ボードの実行を中断させる monitor step//ステップで実行するように load // leds_elfをロード, 「elf」というフォーマットのファイルにアドレスを含まれる break main // 「main」関数にブレークポイントを設定 continue // プログラムを実行させて、「main」にとまってステップでデバッグ可能

「LPC2388_LED」プロジェクトの配下、「gdb.cmd」ファイルの内容をそのままコピーしても OK 「gdb.cmd」を右クリック、「Open With」→「Text Editor」 *ほかの設定を保存するため、コピー途中で「Apply」ボタンを押して適用します。





上記のコピー内容を下記「Commands」に貼り付け、「Apply」ボタンを押下

🖨 Debug Configurations 🔀			
Create, manage, and run cor	figurations	Ť	
Image: Second system Image: Second system Image: Secon	Name: LPC2388_LED Main Debugger Commands Help/tips on how to setup GDB init script 'Initialize' commands Target remote localhost:3333 monitor halt monitor arm920t cp15 2 0 monitor arm920t cp15 2 0 load break main continue 'Run' commands		
Filter matched 10 of 10 items	Apply	Re <u>v</u> ert	
?	Debug	Close	



- 5. デバッグ
 - ① openocd 起動







② gdb 起動

アイコン「Debug」をクリックし、「LPC2388_LED」を押下





③ 関数「main」に止まり、ショットカットキー「F6」でステップでデバッグできる

🛢 Debug - D:¥embeded¥eclipse¥workspace¥lpc2388¥LPC2388_LED¥src¥main.c - Eclipse 📃 🗖 🛛				
<u>F</u> ile <u>E</u> dit <u>S</u> ource Refac <u>t</u> or <u>N</u> avigate Se <u>a</u> rch <u>P</u> roject <u>R</u> un <u>W</u> indow <u>H</u> elp				
∃ 📬 ▾ 🔜 🗁 📾 ∃ 🏇 ▾ 🔾 ▾ ધୁ🏧 ▾ ຢ 🎾 😂 🛷 ▾ ຢ 🌽 ▾ 🖏 ▾	*> + +	🖹 🏇 Debug 💀 C/C++		
🏇 Debug 🕱 📃 🗖	🝽= Variables 😫 📲 Registers 📑 Modules	🗄 🍕 🕞 🏟 🗳 💥 🍇 🗸 🗖 🗖		
🍇 🖉 🚸 🕪 🗉 🔳 🕅 🧏 👁 🕫 📑 🚺 🏘 🎽	Name V	alue		
Openocd [Program]	(x)= i 38	5		
Difembeded¥openocd¥bin¥openocd.exe				
erozolog (2000) Erozolog (2000) Embedded debde (0090000) erozolog (2000) Embedded GDB (09/07/26 16:09) (Suspended)				
🖨 👘 Thread [0] (Suspended)				
☐				
Di¥embeded¥yagarto¥bin¥arm-elf-gdb.exe (09/07/26 16:09)		~		
	<	>		
Construction (Double and (Double a))				
Makerile gob.cmo gob.cmo c main.c 23				
int unin/wrid)	Ov000001b0 <main+8>: mov 0v000001b4 <main+12>: atv</main+12></main+8>	vr3, #U; UXU		
	0x000001b4 <main+12>: St.</main+12>	Ovios (LFD Init>		
int i = 0;	0x000001bc <main+20>: mo</main+20>	v r0. #255 : 0xff		
	0x000001c0 <main+24>: b1</main+24>	Ox140 <led on=""></led>		
LED_Init();	0x000001c4 <main+28>: mov</main+28>	vr3, #0 ; 0x0		
	0x000001c8 <main+32>: st</main+32>	r r3, [sp]		
while(1)	Ox000001cc <main+36>: b</main+36>	Ox1dc <main+52></main+52>		
{	<pre>0x000001d0 <main+40>: 1d;</main+40></pre>	r r3, [sp]		
LED_ON(UXFF);	0x000001d4 <main+44>: ad</main+44>	d r3, r3, #1 ; 0x1		
IOT(1 = 0; 1 < 10000; 1++);	0x000001da <main+46>: st:</main+46>	r r3, [sp] r r2 [an]		
for(i = 0; i < 10000; i++);	• 0x0000014C <main+52>. 14</main+52>	v r3, #9984 : 0x2700		
	>	>		
📃 Console 🕱 🛛 🖉 Tasks 🚼 Problems 🚺 Executables 📋 Memory	🔲 🗶 💥 📑 🚮 🚦	🔜 🖳 🗐 🛃 🛃 • 🗂 🗖		
LPC2388_LED [Zylin Embedded debug (Cygwin)] D¥embeded¥yagarto¥bin¥arm-elf-gdb.exe (09/07.	/26 16:09)			
Warning: /cygdrive/D/embeded/eclipse/workspace/lpc2388/LPC2388_	LED/.dep: No such file or director	cy. 🔼		
Warning: /cygdrive/D/embeded/eclipse/workspace/lpc2388/LPC2388_	LED: No such file or directory.			
mi_cmd_disassemble: Invalid filename.				
No line 69 in file "STC/main.C".				
No source file haned feas.c.		~		
3				
÷ ∎◆				



(

株式会社日昇テクノロジー

6.3 ARM Cotex-M3 の STM32F103	
6.3.1 STM32F103 ボード購入 URL	
http://www.csun.co.jp/SHOP/200903019.html	
ボード URL から STM32F103 マニュアルを参照し、Open-JTAG に接続してください。	
ボード URL から LPC2388 マニュアルを参照し、Open-JTAG に接続してください。	
STM32F103 ボードマニュアルダウンロード URL :	
http://www.dragonwake.com/download/download-stm32/STM32_manual.pdf	
6.3.2 ハードウエア動作確認	
(1). OpenJTAG をパソコンの USB ポートに挿入する	
(2). JTAG ケーブルで OpenJTAG と STM32 ボードを繋ぐ	
(3). STM32 ボードに電源を入れる	
(4). 下記のコマンドを入力します。	
openocd -f "D:\embeded\eclipse\workspace\config\interface\open-jtag.cf	g" -f
"D:\embeded\eclipse\workspace\config\target\stm32.cfg"	
※ はじめの-f は OpenJTAG のコンフィグファイルを使います。二番目の-f は	stm32のコンフ
ィグファイルを使います。	
🔤 C:¥WINDOWS¥system32¥cmd.exe - openocd -f "D:¥embeded¥eclipse¥workspace¥confi 🗕 🗖	×
\$URL: https://kc8apf@svn.berlios.de/svnroot/repos/openocd/tags/openocd-0.1.0/src	
/openocd.c \$	
Can't find D:¥embeded¥eclipse¥workspace¥config¥interface¥open-jtag.cfg	
C:¥Documents and Settings¥dragonwake>openocd -f ~D:¥embeded¥eclipse¥workspace¥cc	
ntig¥intertace¥open-jtag.ctg" -t "D:¥embeded¥eclipse¥workspace¥contig¥target¥stm	
Upen Un-Chip Debugger U.I.U (2009-01-21-21:15) Kelease	
DLCC2. Dead, latter: //even logical de /eveneet /rense /energed /trumb//DLCC	
DUGS: Nead Inttp://svn.beriios.de/svnroot/repos/openoca/trunk/dugs	
\$1. RL: https://kc8anf@svp.berligs.de/svproot/repos/opepocd/tags/opepocd-0.1.0/src	
Zopenood o \$	
itag speed: 0	
500 kHz	
Info : JTAG tap: stm32.cpu tap/device found: 0x3ba00477 (Manufacturer: 0x23b, Pa	
rt: Oxba00, Version: 0x3)	
into : JTAG Tap/device matched	
Info : JTAG tap: stm32.bs tap/device found:_0x16410041 (Manufacturer: 0x020, Par	
t: Ox6410, Version: Ox1)	
Info : JTAG Tap/device matched	
	-



6.3.3 LED サンプルデバッグ

*サンプルの導入について、二つ方法があります、一つは新規プロジェクトを作成してからソースを導入、もう一つ はダウンロードプロジェクトをインポートします。

*新規プロジェクトの作成方法は「6.1.3 LPC2148 用のサンプル「LED」をデバッグ」を参照 *ここに既存プロジェクトのインポートとして説明

1. プロジェクト導入: 「File」 \rightarrow 「Import」 \rightarrow 「Existing Projects into workspace」





株式会社日昇テクノロジー





既にダウンロードしたプロジェクトを Wokspace に入れていますので、ここに「Copy projects into workspace」を 外します。

🥏 Import		_ 🗆 🔀
Import Projects Select a directory to sear	ch for existing Eclipse projects.	
 Select root directory: Select archive file: Projects: 	D:¥embeded¥eclipse¥workspace¥stm32¥STI	Browse Browse
⊽ STM32_LED (D:¥e	mbeded¥eclipse¥workspace¥stm32¥STM32_L	<u>S</u> elect All <u>D</u> eselect All R <u>e</u> fresh
Copy projects into we	III III III III III III IIII IIII IIII IIII	
Working sets: Sample	ing sets	S <u>e</u> lect
?	< <u>B</u> ack <u>N</u> ext > <u>F</u> inish	Cancel



2. 導入済みのプロジェクトをコンパイル 「STM32_LED」を選択、「Project」→「Build Project」を押下





- 3. openocd 設定の修正
 - 一般の設定は「5.4 OpenOCD の設定」を参照
 - a)「Working Directory」を「STM32_LED」に変更
 - b) 「Arguments」を下記のように変更

-f "D:¥embeded¥eclipse¥workspace¥config¥interface¥open-jtag.cfg" "D:¥embeded¥eclipse¥workspace¥config¥target¥stm32.cfg"

😂 External Tools Configurat	ions 🛛 🕅 🕅
Create, manage, and run cor Run a program	nfigurations Organizations
type filter text	Name: openocd Main Refresh Difference Browse Browse Browse File System Variables Variables Arguments: Prowse Workspace In Difference Difference In Difference Difference In Difference Difference In Difference Browse Workspace In Differenco <td< th=""></td<>
Filter matched 2 of 2 items	
?	<u>R</u> un Close



4. STM32_LED 用の GDB を追加

①アイコン「Debug」をクリック、「Debug Configuration」を押下





②「Zylin Embedded debug(Cygwin)」を右クリック、「New」を選択





③「Name」」を入力、「Project」、「C/C++ Application」を指定

Debug Configurations			
Create, manage, and run cor	figurations	Ť.	
Image: Strain Strai	Name: STM32_LED Main Source Optional Project (optional): STM32_LED C/C++ Application: Application: D¥embeded¥eclipse¥workspace¥stm32¥STM32_LED¥main elf Application console	Browse Search Project Browse	
Filter matched 9 of 9 items		Apply Revert	
?		Debug Close	



④「GDB debugger」を「arm-none-eabi-gdb.exe」の場所に指定

Debug Configurations	
Create, manage, and run con	figurations
Image: Second system Image: Second system <th>Name: STM32_LED Main Debugger Commands Source Common Debugger: Embedded GDB Advanced Debugger Options Main GDB debugger: Dr#embeded¥arm-eabi¥bin¥arm-none-eabi-gdb.exe Browse GDB command file: Browse Warning: Some commands in this file may interfere with the started operation of the debugger, for example "run") GDB command set: Standard Protocol: mi Protocol: mi Use full file path to set breakpoints</th>	Name: STM32_LED Main Debugger Commands Source Common Debugger: Embedded GDB Advanced Debugger Options Main GDB debugger: Dr#embeded¥arm-eabi¥bin¥arm-none-eabi-gdb.exe Browse GDB command file: Browse Warning: Some commands in this file may interfere with the started operation of the debugger, for example "run") GDB command set: Standard Protocol: mi Protocol: mi Use full file path to set breakpoints
Filter matched 9 of 9 items	Appl <u>v</u> Re <u>v</u> ert
?	Debug Close



株式会社日昇テクノロジー

⑤「Commands」を入力

target remote localhost:3333 // ローカルポート「3333」と接続(OpenOCDと接続) monitor halt//ボードの実行を中断させる monitor step//ステップで実行するように load // leds_elfをロード, 「elf」というフォーマットのファイルにアドレスを含まれる break main // 「main」関数にブレークポイントを設定 continue // プログラムを実行させて、「main」にとまってステップでデバッグ可能 「STM32_LED」プロジェクトの配下、「gdb.cmd」ファイルの内容をそのままコピーしても OK 「gdb.cmd」を右クリック、「Open With」→「Text Editor」

*ほかの設定を保存するため、コピー途中で「Apply」ボタンを押して適用します。





上記のコピー内容を下記「Commands」に貼り付け、「Apply」ボタンを押下

🍃 Debug Configurations		
Create, manage, and run cor	figurations	TO-
Image: Second system Image: Second system <td>Name: STM32_LED Main Main Debugger Commands E Commands Source Common Help/tips on how to setup GDB init script 'Initialize' commands tagest remote localhost:3333 monitor halt monitor step load break main continue 'Run' commands</td> <td></td>	Name: STM32_LED Main Main Debugger Commands E Commands Source Common Help/tips on how to setup GDB init script 'Initialize' commands tagest remote localhost:3333 monitor halt monitor step load break main continue 'Run' commands	
Filter matched 9 of 9 items		Apply Reyert
?		Debug Close



5. デバッグ





株式会社日昇テクノロジー

② gdb 起動





③ 関数「main」に止まり、ショットカットキー「F6」でステップでデバッグできる

😂 Debug - D:¥embeded¥eclipse¥workspace¥stm32¥STM32_LED¥main.c - Ec	Eclipse
<u>File E</u> dit <u>S</u> ource Refactor <u>N</u> avigate Search <u>P</u> roject <u>R</u> un <u>W</u> indow <u>H</u> elp	
i 📬 • 📰 🖮 i 🕸 • 🔾 • 🎥 • 🥬 • i 🖉	
🏇 Debug 🕱 🗖 🗖	' 🗖 (X)= Variables 🕱 🛛 🔐 Registers 🛋 Modules 🛛 😓 🐗 📄 🧔 🌮 🕱 🖾
🍇 🚧 🚸 🗈 🗉 🔳 🖄 😎 📭 🖬 🙀 🎽	Name Value
Comparison of Program D¥embeded¥openocd¥bin¥openocd.exe J D¥embeded4openocd¥bin¥openocd.exe Fc STM32_LED [Zylin Embedded debug (Cygwin)] Je Je Embedded GDB (09/07/23 2020) (Suspended) Je Je RecQ Configuration0 d¥embeded¥eclipse¥workspace¥stm32¥stm32_led¥main.c47 0x08000 J 1 main0 d¥embeded¥eclipse¥workspace¥stm32¥stm32_led¥main.c47 0x08000	¥maii 18000
D:¥embeded¥arm-eabi¥bin¥arm-none-eabi-gdb.exe (09/07/23 2:20)	
(*)	
mainc X	🖓 🗖 🔚 Outline 🔜 Disassembly 🕱 🗸 🗸 🗸
<pre>* Function Name : RCC_Configuration * Description : Configures the different system clocks. * Input : None * Output : None * Return : None * Return : None ************************************</pre>	<pre>(</pre>
📮 Console 🕱 🧔 Tasks 🛃 Problems 💽 Executables 📋 Memory	📒 🗶 💥 📴 🔐 🖳 🖳 🖳 🛃 🖅 😁 🗖
STM32_LED [Zylin Embedded debug (Cygwin)] D¥embeded¥arm-eabi¥bin¥arm-none-eabi-gdb.excontinue Note: automatically using hardware breakpoints for read-only a Breakpoint 1, main () at main.c:47 47 RCC_Configuration();	bexe (09/07/23 220) y addresses.
	Millable Cillat District 00 - 1



- ④ ステップでデバッグ
 - *関数「main」のコーディング「GPIO_SetBits(GPIOB, GPIO_Pin_5);」を実行前、 要は、LED 点灯前に、STM32 ボードの LED が消している



ショットカットキー「F6」を押しながら、関数「main」のコーディング「GPI0_SetBits(GPI0B, GPI0_Pin_5);」 を実行後、SMT32 ボードの LED が点灯している





6.4 ARM9 Ø MINI2440

6.4.1 MINI2440を購入	
MINI2440 ボード購入 URL : (LCD3.5 とのセット)	
http://www.csun.co.jp/SHOP/200812021.html	
LPC2148 マニュアルダウンロード URL :	
http://www.dragonwake.com/download/arm9-download/linux-2.6.29/MINI2440-linux-2.6.29.pdf	
6.4.2 ハードウエア動作確認	
1. OpenJTAGをパソコンの USB ポートに挿入する	
2. JTAG ケーブルで OpenJTAG と MINI2440 ボードを繋ぐ	
3. MINI2440 ボードに電源を入れる	
4. 下記のコマンドを入力します。	
openocd -f "D:¥embeded¥eclipse¥workspace¥config¥interface¥open-jtag.cfg"	_
"D:¥embeded¥eclipse¥workspace¥config¥target¥mini2440.cfg"	
※ はじめの-f は OpenJTAG のコンフィグファイルを使います。二番目の-f は mini244 アイルを使います。	0 のコンフィグフ
🖾 C:¥WINDOWS¥system32¥cmd.exe - openocd -f "D:¥embeded¥eclipse¥workspace¥confi 🗕 🗖 🗙	
500 kHz Info : JTAG tap: stm32.cpu tap/device found: 0x3ba00477 (Manufacturer: 0x23b, Pa rt: 0xba00, Version: 0x3) Info : JTAG Tap/device matched Info : JTAG tap: stm32.bs tap/device found: 0x16410041 (Manufacturer: 0x020, Par t: 0x6410, Version: 0x1) Info : JTAG Tap/device matched	
C:¥Documents and Settings¥dragonwake>openocd -f "D:¥embeded¥eclipse¥workspace¥co nfig¥interface¥open-jtag.cfg" -f "D:¥embeded¥eclipse¥workspace¥config¥target¥min i2440.cfg" Open On-Chip Debugger 0.1.0 (2009-01-21-21:15) Release	
BUGS? Read http://svn.berlios.de/svnroot/repos/openocd/trunk/BUGS	
\$URL: https://kc8apf@svn.berlios.de/svnroot/repos/openocd/tags/openocd-0.1.0/src /openocd.c \$ jtag_speed: 0 fast_memory_access_is_enabled	
Info : JTAG tap: s3c2440.cpu tap/device found: 0x0032409d (Manufacturer: 0x04e, Part: 0x0324, Version: 0x0) Info : JTAG Tap/device matched	



6.4.3 LED サンプルデバッグ

1. サンプルの導入とコンパイル:

サンプルの導入について、二つ方法があります、一つは新規プロジェクトを作成してからソースを導入、

もう一つはダウンロードプロジェクトをインポートします。

*ここに新規プロジェクトの作成方法として説明

①プロジェクト「MINI2440_LED」を作る

a) 新規プロジェクトを作成するため "File"→"New"→"C Project" を選択します

	D/makefile -	Eclipse
<u>File</u> <u>Edit</u> <u>Source</u> Refactor	<u>N</u> avigate Se	earch Project Run Window Help
<u>N</u> ew	Alt+Shift+N	C++ Project 😰 😥 🛷 🔹 🗐 👖 🗄 🖢 🖓 🗸
Open File <u>"</u>		C Project
Close	Ctrl+W	Project
Close All	Ctrl+Shift+W	Convert to a C/C++ Project
Save	Ctrl+S	😚 Source Folder nis makefile with an editor wh
Save As		😭 Folder
Save All	Ctrl+Shift+S	Source File
Revert		h Header File
		📑 🕆 File from Template
Mo <u>v</u> e	50	C Class
Rena <u>m</u> e	F2	<mark>≓∲</mark> Other Ctrl+N
& Retresh	F5	t files.
Convert Line Delimiters To		# To rebuild project do "make clean" and "make all".
<u>P</u> rint	Gtrl+P	#
Switch <u>W</u> orkspace		
Restart		# Start of default section
N. Turnaut		#
Import		
Z Export		TRGT = arm-elf-
P <u>r</u> operties	Alt+Enter	CC = \$ (TRGT) gcc
1 makafila [LPC2288 LED]		$CP = \Im(RGI) \operatorname{op}_{COP}$
2 Makefile [Keil GCC/ADC	1	BIN = (CP) -O ihex
3 adoc [ADC]	1	
4 Makefile [ADC]		MCU = arm7tdmi
T waterie [upo]		
E <u>x</u> it		🚼 Problems 🕢 Tasks 📮 Console 🛛 🔲 Properties
		C-Build [LPC2388_LED]
		**** Build of configuration Default for project LPC2388_LEI
		make all
		make: Nothing to be done for `all'.
☐ [◆] ² ² LPC2388_LEE)	



b)適当なプロジェクト名前(MINI2440_LED)を入力し Finish ボタンを押します。

😂 C Project	🗆 🔀
C Project	
Create C project of selected type	
Project name MINI2440_LED	
Use <u>d</u> efault location	
Location: D:¥embeded¥eclipse¥workspace¥n	nini2440¥MINI2440_LED Browse
Project type:	Toolchains:
 Makefile project Empty Project 	Other Toolchain
Show project types and toolchains only if	they are supported on the platform
(?) < <u>B</u> ack	<u>N</u> ext > <u>Einish</u> Cancel



c)サンプルソースを作成されたプロジェクトにインポート Project Explorer の「MINI2440_LED」を選べ、右クリックし、「Import」を押下

🛢 C/C++ - L	PC2388_LED/makefile -	Eclipse		_ 🗆 🔀
<u>F</u> ile <u>E</u> dit <u>S</u> ou	irce Refac <u>t</u> or <u>N</u> avigate Se	e <u>a</u> rch <u>P</u> roject <u>R</u> un <u>V</u>	<u>M</u> indow <u>H</u> elp	
1 📬 - 🗔 🖻	🗟 : @ - 않 - € -	⊙ - ≪ - ⊗ -	券・ 🖸 ・ 隆 🗁 🖋 ・ 🗄 🕤 🗄 🖢 - 🖏 - 🐤 -	😭 📴 C/C++
Project Explo	orer 🛛 🗖 🗖	🛛 🔓 makefile 🛛	- 8	🗄 Outli 🛛 💿 Make 🗖 🗖
🕀 🎏 LPC2381		#	- L	TRGT
🕀 🔁 MINI244	<u>N</u> ew		• Do NOT edit this makefile with an editor which rep	
	ιο μτο			CP
	Open in <u>N</u> ew Window			BIN
	Cody	Ctrl+C	line:	MCU
	Paste	Ctrl+V	Create project	DDEFS
	💢 <u>D</u> elete	Delete	create project	DADEFS
	Remove from Context	Ctrl+Alt+Shift+Down	= Clean project files.	
	Mo <u>v</u> e			DLIBS
	Rename	F0	project do "make clean" and "make all".	PROJECT
<	No Import			LDSCRIPT_ROM
	A Export			
			fault section	SRC
	Build Project			ASRC
	Clean Project		f-	
	8 Retresh	FD) gee	
	Close Project)objcopy	- OPT
	Close Unrelated Projects)gcc -x assembler-with-cpp	
	Exclude from build		-0 inex	
	Build Configurations	1	mi 🗸	
	Make Targets	I		
	Index	I	S 📮 Console 🕱 🔲 Properties	
	Convert To			
	<u>R</u> un As	1	•	<u>^</u>
	<u>D</u> ebug As	I	•	
	<u>P</u> rofile As	1		
	Team	I	•	
	Comp <u>a</u> re With		•	
	Restore from Local History	<u>V</u>		<u>×</u>
1 D* 💈	Properties	Alt+Enter		



d)「File System」をクリック

Import	_ 🗆 🔼
Select Import resources from the local file system into an existing project.	Ľ
Select an import source: type filter text General Grupp Archive File	
Existing Projects into Workspace Proferences C/C++ CVS Run/Debug	
⊕…	
	Cancel



e) ダウンロードしたサンプル MINI2440_LED を選択





f)選択したソースを導入

🔿 Import 📃 🗔 🔀
File system Source is in the hierarchy of the destination.
From directory: D:¥embeded¥eclipse¥workspace¥mini2440¥MINI2440_LED 💽 Browse
MINI2440_LED □ cproject □ project □ gdb.cmd □ jtag □ leds.c □ leds.let □ leds.lst □ leds.lst
Into folder: MINI2440_LED Browse
Options Overwrite existing resources without warning Oreate complete folder structure Oreate selected folders only
Image: Second


<u>g)サンプル MINI2440_LED をビ</u>ルド

😂 C/C++ - LPC2388_LED/makefile - Ec	lipse	
<u>Eile E</u> dit <u>S</u> ource Refac <u>t</u> or <u>N</u> avigate Se <u>a</u> rc	h <u>Project R</u> un <u>Window H</u> elp	
: 📬 • 🔚 🕒 💼 : 🎯 • 🚳 • 💣 • 🮯	 Opgn Project Cloge Project Deg Project 	😭 🖬 C/C++
Project Explorer 🛛 🗖 🗖	Build All Ctrl+B	🗄 Outli 🛛 💿 Make 🗖 🗖
□ 🔄 😜 ▽		_e j <u>a</u> ⊽
⊕ 🚝 LPC2388_LED	Eulid Project	TRGT
MINI2440_LED		
	Build Automatically	CP
	Mala Taurat	AS DIN
ede Ide		
Makefile	Properties nbler-with-cpp	DDEFS
	BIN = S(CP) - O INEX	DADEFS
	MCU = arm7tdmi	- DINCDIR
		DLIBDIR
	# List all default C defines here, like -D_DEBUG=1	
	DDEFS =	
	A Line all default ACM defines have like D DEDUCAT	UDEFS
	DADEFS =	UADEFS -
		SRC .
	# List all default directories to look for include files here	ASRC
	DINCDIR =	
	# List the default directory to look for the libraries here	- OPT
	BHIDDIK -	INCDIR
	# List all default libraries here	
	DLIBS =	DEFS
(E	🦹 Problems 🕢 Tasks 📃 Console 🙁 🔲 Properties	🔒 🚉 🛃 🗐 - 🗂 🗖
	-Build [MINI2440_LED]	
	**** Clean-only build of configuration Default for project MINI2440_L1	5D ****
n	ake clean	
r	m -f leds.dis leds.bin leds elf *.o	
	-	
		~
É D [♦] 🥵 MINI2440_LED		



h)ビルドが成功かどうかを確認

<pre>Project Explore 2 Project Explore 2 Makefile # Start of default section # Start of default default default section # List all default default default section section section # List all default directories to look for include files here DINCOR # List all default directory to look for the libraries here DINCOR # List all default directory to look for the libraries here DINCOR # List all default libraries here DINCOR # List all default libraries here DINCOR # List all default directory to look for the libraries here DINCOR # List all default directory to look for the libraries here DINCOR # List all default directory to look for the libraries here DINCOR # List all default directory to look for the libraries here DINCOR # List all default directory to look for the libraries here DINCOR # List all default directory to look for the libraries here DINCOR # List all default directory to look for the libraries here DINCOR # List all default directory to look for the libraries here DINCOR # List all default directory to look for the libraries here DINCOR # List all default directory of look for the libraries here DINCOR # Code = The Start - The</pre>		h <u>P</u> roject <u>R</u> un <u>Wi</u> ndow <u>H</u> elp	
<pre>Project Explore S</pre>	🔁 • 🗄 📥 🗄 💣 • 🚳 • 🗳 • 🤤	• ! 🗞 • 🕲 • ! 🏇 • 🖸 • 🗛 • ! 😕 😂 🔗 • ! 🗉 🔳 ! ½ • ? • * • • •	🖹 🛅 C/C++
<pre># Start of default section # Start of default section # TROT = arm-elf- CC = \$(TROT) age CC = \$(TROT) a</pre>	Project Explorer ⋈ □ □	🚡 makefile 🛛 🗋 Makefile 🦳 🧮 🖓	🗆 🗄 Outli 🛛 💿 Make 🕤
<pre>TRCT = arm-elf- CC = \$(TRCT) dpicopy AS = \$(TR</pre>	E 25 LPC2388_LED	######################################	TRGT
CP = \$ (TRGT) objcopy AS = \$ (TRGT) objcopy AS = \$ (TRGT) dojcopy AS = \$ (TRGT) dojcopy	in - ∰ Binaries in crt0.S	TRGT = arm-elf- CC = \$(TRGT)gcc	CP
Image: Second Mathematical Second	€- <u>c</u> leds.c → <u>m</u> cro.o - (arm/le) ⊕ -☆ leds_elf - [arm/le] ⊕ -∰ leds.o - [arm/le]	CP = \$(TRGT)objcopy AS = \$(TRGT)gcc -x assembler-with-cpp BIN = \$(CP) -O ihex	BIN
<pre># List all default C defines here, like -D_DEBUG=1 DDEFS = # List all default ASM defines here, like -D_DEBUG=1 DADEFS = # List all default ASM defines here, like -D_DEBUG=1 UADEFS # List all default directories to look for include files here DINCDIR = # List all default directory to look for the libraries here DIEDIR = # List all default libraries here DLES = # List all default all default libraries here DLES = # List all default all def</pre>	edb.cmd	MCU = arm7tdmi	
<pre># List all default ASH defines here, like -D_DEBUG=1 DADEFS = # List all default directories to look for include files here DINCDIR = # List the default directory to look for the libraries here DIBDIR = # List all default libraries here DLIBD R ULBD R ULB R ULBD R ULB R U</pre>	eus.us leds.lds	<pre># List all default C defines here, like -D_DEBUG=1 DDEFS =</pre>	PROJECT
# List all default directories to look for include files here DINODIR = # List the default directory to look for the libraries here DLIBDIR = # List all default libraries here DLIBS = C-Buid [MDERATIED] Froblems Tasks Consolt 30 Troperties C-Buid [MDERATIED] Frob arm-elf-gcc -Wall -Wstrict-prototypes -g -fomit-frame-pointer -ffreestanding c -o leds.o leds.c arm-elf-ld -Tleds.lds crt0.o leds.o -o leds_elf arm-elf-objcopy -0 binary -S leds_elf leds.bin arm-elf-objdump -D -m arm leds_elf > leds.dis		<pre># List all default ASM defines here, like -D_DEBUG=1 DADEFS =</pre>	UDEFS
<pre># List the default directory to look for the libraries here DLIBDIR = # List all default libraries here DLIBS =</pre>		<pre># List all default directories to look for include files here DINCDIR =</pre>	UINCDIR
<pre># List all default libraries here DLIBS =</pre>		<pre># List the default directory to look for the libraries here DLIBDIR =</pre>	OPT
C-Build [Multiconstant] C-Build [Multiconstant] C-Build [Multiconstant] C-Build [Multiconstant] C-Build [Multiconstant] C-Build [Multiconstant] C-Build [Multiconstant] C-Build [Multiconstant] arm-elf-gcc -Wall -Wistrict-prototypes -g -fomit-frame-pointer -ffreestanding c -o leds.o leds.c arm-elf-old -Tleds.lds crt0.o leds.o -o leds_elf arm-elf-objcopy -0 binary -S leds_elf leds.bin arm-elf-objdump -D -m arm leds_elf > leds.dis		<pre># List all default libraries here DLIBS =</pre>	
C-Build [Multivero[LED] prov.S arm-elf-gcc -Wall -Wstrict-prototypes -g -fomit-frame-pointer -ffreestanding c -o leds.o leds.c arm-elf-ld -Tleds.lds crt0.o leds.o -o leds_elf arm-elf-objcopy -O binary -S leds_elf leds.bin arm-elf-objdump -D -m arm leds_elf > leds.dis	(E	Problems 2 Tasks Concele 2 Troperties	
arm-elf-gcc -Wall -Wstrict-prototypes -g -fomit-frame-pointer -ffreestanding c -o leds.o leds.c arm-elf-ld -Tleds.lds crt0.o leds.o -o leds_elf arm-elf-objcopy -O binary -S leds_elf leds.bin arm-elf-objdump -D -m arm leds_elf > leds.dis	C	-Build [MDM2440_LED]	
arm-elf-objcopy -O binary -S leds_elf leds.bin arm-elf-objdump -D -m arm leds_elf > leds.dis		rm-elf-gcc -Wall -Wstrict-prototypes -g -fomit-frame-pointer -ffree eds.c rm-elf-ld -Tleds.lds crt0.o leds.o -o leds elf	standing c -o leds.o
	e	rm-elf-objcopy -O binary -S leds_elf leds.bin rm-elf-objdump -D -m arm leds_elf > leds.dis)
	₽		



2. サンプル MINI2440_LED をデバッグ		
①MINI2440 用の OpenOCD デバッグパラメタを設定		
* OpenOCD の設定は「5.4 OpenOCD の設定」(P23~P26)を参照		
a) MINI2440 用のコンフィギュレーションファイル内容を確認		
「config/interface」の中:JTAG の種類による コンフィギュレ-	ーションファイル	
Open-JTAG 用のンフィギュレーションファイル:「open-jtag.cfg	[]	0
D:¥embeded¥eclipse¥workspace¥config¥interface¥open-jtag.cfg - EmEditor File Edit Search View Macros Tools Window Help		×
: □ • ♪ ↓ ≥ ∅ X □ © ♥ ♥ ₽ ♥ ♥ ♥ ■ @ @ @ ♥ ♥ ■ ● ● ◆	j Took	»
j Windows 强 Mini2440.cte 📴 open-itae.cfe		×
#Change the default telnet port V		Î
telnet_port 4444		
# GDB connects here↓		
gdb_port 3333↓		
# GDB can also flash my flash!↓		
gdb_memory_map enable↓		Ξ
gdb_flash_program enable↓		
#TCL port setting↓		
tcl_port 6666↓		
interface ft2232		
jtag_speed O ψ		
ft2232_vid_pid_0x1457_0x5118↓		
ft2232_layout "jtagkey_prototype_v1"↓		
ft2232_device_desc ~USB<=>JTAG&RS232 A~~		
		~
344 bytes, 14 lines.	Text Ln 10, Col 17 日本語 (シフト JIS)	







c) OpenOCD の設定画面を呼び出す





- d)「Location」、「Working Directory」、「Arguments」を設定
- *特に「Arguments」に MINI2440 用のコンフィギュレーションファイルを設定必要
- -f D:¥embeded¥eclipse¥workspace¥config¥interface¥open-jtag.cfg
- -f D:¥embeded¥eclipse¥workspace¥config¥target¥Mini2440.cfg





e) OpenOCD の起動が成功かどうかを確認





②MINI2440 用のデバッグを設定a)デバッグ設定画面を呼び出す

➡ C/C++ - LPC2388_LED/makefile -	Eclipse	_ 🗆 🗙
<u>Eile E</u> dit <u>S</u> ource Refac <u>t</u> or <u>N</u> avigate S	earch <u>Project Run Wi</u> ndow <u>H</u> elp	
		E Out Mula
 ► S LPC2388_LED ► S Invies ► S ortLS ► C leds.c ► C leds.c ► C leds.c = (<i>i</i> m / e] ► d leds.d = (<i>i</i> m / e] ► d led	<pre>####################################</pre>	TRGT CC CP AS DDEFS DADEFS DADEFS DINCDIR DLIBDIR DLIBDS PROJECT LDSCRIPT_ROM UDEFS ASRC ASRC UINCDIR UINCDIR ULIBDS OPT
	BIN = \$(CP) -O ihex MCU = arm7tdmi	
	Problems @ Tasks E Console 23 Properties	
	openocd [Program] D/¥embeded¥openocd¥bin¥openocd.exe	
	<pre>\$URL: https://kc8apf@svn.berlios.de/svnroot/repos/openocd/tags/openocd</pre>	-0.1.0/src/openocd.c 📐
	jtag_speea: U fast memory access is enabled	
	Info : JTAG tap: s3c2440.cpu tap/device found: 0x0032409d (Manufacture Info : JTAG Tap/device matched	er: OxO4e, Part: OxO32



b)「Zylin Embedded debug(Cygwin)」を選べ、右にクリックしてから「New」を押下

😂 C/C++ - LPC	Debug Configurations	
Eile Edit Source	Create, manage, and run configurations	Debug 📴 C/C++
	Configure launch settings from this dialog: ype filter text Configure launch settings from this dialog: C/C++ Application C/C++ Attach to Application C/C++ Postmortem Debugger tearch Group C/C++ Opticate Debugger tearch Group C/C++ Opticate Debugger C/C++ Optic	S R R R R R R R R R R R R R R R R R R R
	Filter matched 6 of 6 items	📮 • 📬 • 🗖 🗖
	Debug Ciose	openocd.c
	fast memory access is enabled Info: JTAG tap: s3c2440.cpu tap/device found: 0x0032409d (Manufacturer: 0x04e, Info: JTAG Tap/device matched	Part: 0x032
: 🗆		



c)デバッグ設定画面の中、「Main」タブ項目を設定

Debug Configurations		
Create, manage, and run configuratio	ns 🔅	
Image: Second state st	Name: MINIZ440_LED Main Debugger Commands Common Project Coptional): MINIZ440_LED Browse C/C++ Application: D¥embeded¥eclipse¥workspace¥mini2440¥MINIZ440_LED¥leds Search Project Browse Other and the state of the st	
Filter matched 7 of 7 items		
?	Debug Close	



d)「Commonds」タブ項目を設定(//の後にはコマンドのコメントとなり、設定内容に入れない)

target remote localhost:3333 // ローカルポート「3333」と接続(OpenOCDと接続)
monitor halt//ボードの実行を中断させる
monitor arm920t cp15 2 0 // MMU機能をクローズ
monitor step//ステップで実行するように
load // leds_elfをロード, 「elf」というフォーマットのファイルにアドレスを含まれる
break main // 「main」関数にブレークポイントを設定
continue // プログラムを実行させて、「main」にとまってステップでデバッグ可能

「MINI2440_LED」プロジェクトの配下、「gdb.cmd」ファイルの内容をそのままコピーしても OK 「gdb.cmd」を右クリック、「Open With」→「Text Editor」 *ほかの設定を保存するため、コピー途中で「Apply」ボタンを押して適用します。





③ OpenOCD を起動させる





b)成功に起動したことを確認





④ Debug ツール「gdb」が起動前、ARM9ボードの様子 a) 4LED が全て消している





⑤ Debug ツール「gdb」を起動させる

🔿 Debug – D:¥embeded¥eclipse¥workspace¥mini2440¥MINI2440_LED¥leds.c –	- Eclipse	
<u>File E</u> dit <u>S</u> ource Refactor <u>N</u> avigate Se <u>a</u> rch <u>P</u> roject <u>R</u> un <u>W</u> indow <u>H</u> elp		
i 📬 • 🖫 💩 📾 j 🦀 • 🕖 • 🖓 • 😳 🔗 • i 🌛 i 🖢 - 🖓 • 🌼 🗇	• 🗘 - 🗈 🖹 🏇 Debug	/C++
🏇 Debug 🛛 🦷 🔂 😨 🖓 🖾 🖾 🖓	M= Variables 🛿 り 🔛 Registers 🛋 Modules 🛛 😓 🏘 📄 🏟 🜮 😭 🏀	- 0
<u></u> → 💦 🖓 👘 🗸	Name Value	
🖃 💁 openocd [Program Debug Configurations	(x)⊧i 3	
D:¥embeded¥o Organize Fa <u>v</u> orites		
i∋ ∉c MINI2440Debug [Zyrin Emperated debug (∪ygwin)]		
General Control (1970) General (1970)		
Thread [0] (Suspended)		
I mainV Difembededfecilpsefworkspacefminiz440fMiniz440_LEDfieds.c.22 0 Difembededfiuggartaffinitian_clf_gdb.ovg (00/07/16.0.22)		
		<u>^</u>
		~
		>
€ leds.c 🔲 leds.c 🕱	🖓 🗖 🔡 Outline 📓 Disassembly 🕱 🗸 🗸 🗸	- 0
	Ox00000058 <main+20>: add r3, r3, #16 ; 0x10</main+20>	
int main(void)	0x0000005c <main+24>: mov r2, #87040 ; 0x15400</main+24>	
(Ox00000060 <main+28>: str r2, [r3]</main+28>	
unsigned long i = 0;	Ox00000064 <main+32>: mov r0, #29952 ; 0x7500</main+32>	
	0x00000068 <main+36>: add r0, r0, #48 ; 0x30</main+36>	
GPBCON = GPB5_out GPB6_out GPB7_out GPB8_out; // G	<pre>3PB 0x0000006c <main+40>: bl 0x18 <wait></wait></main+40></pre>	
	0x00000070 <main+44>: mov r2, #1442840576 ; 0x560</main+44>	
While (1) {	UXUUUUUUU74 <main+48>: add r2, r2, #20 ; UX14</main+48>	
GPBD#T = (~(i<<5))・ // iの値に上り LED1-4を占減させる	0x0000007c (main+52): 1al r3, [ap]	
if (++i == 16)	0x00000080 <main+60>: myn r3, r3</main+60>	
i = 0;	Ox00000084 <main+64>: str r3, [r2]</main+64>	
3	Ox00000088 <main+68>: 1dr r3. [sv]</main+68>	~
📮 Console 🕱 🖉 Tasks 🔝 Problems 💽 Executables 📋 Memory	😑 🗶 🔆 🗟 🖓 🖳 🔛 🖳 🖳 🛃 🛃 🚽 😁 😁 "	
MINI2440Debug [Zylin Embedded debug (Cygwin)] D:¥embeded¥yagarto¥bin¥arm-elf-gdb.exe (09/1	/07/16 0:32)	
Breakpoint 1, main () at leds.c:17		
17 unsigned long i = 0;		
warning: /cygarive/D/embeded/eclipse/workspace/mini2440/MINI244	40_LED: No such file or directory.	
mi_cmu_uisassemple: invalid filename.		
3		



⑥ ウインドに関数「main」の一番最初に止まっている

The Tark Source Landate Taylore Op		1.0 P				
🗄 🔁 🖌 🔚 📄 👘 🕈 🏷 🗸 🆓 🕶 🖓 🕶	1 🗁 🗁 🔗 🔹 🎿 🗄 🖄	- 친 - 🏷 🔶	• 🗘 -			😭 🕸 Debug 🔤
🏂 Debug 🛛		- 8	(×)= Varia	ables 🕴 🔐 Registers 🛋 Modul	es) 🖆 👘 🕻	= 🍫 🜮 🗶 🖗 🗉
🎽 🖗 🕼 🎉	🔳 🔊 🔊 🔊 🔊 🔊	⇒ 🕫 🍫 ▽	Name		Value	
🖃 💁 openocd [Program]			(×)=		0	
📕 📕 D:¥embeded¥openocd¥bin¥openocd.e	xe					
⊟-≣c MINI2440Debug [Zylin Embedded debug	(Cygwin)]					
🖨 🎯 Embedded GDB (09/07/16 0:32) (Su	spended)					
😑 🛷 Thread [0] (Suspended: Breakpo	nt hit)					
1 main0 D¥embeded¥eclipse	¥workspace¥mini2440¥MINI244	0_LED¥leds.c:19 0				
D:¥embeded¥yagarto¥bin¥arm-elf-go	lb.exe (09/07/16 0:32)					
			2			
						~
leds.c			- 0	📴 Outline 📷 Disassembly 🖾 🔪	~	
			^	0x0000004c <main+8>:</main+8>	mov r3, #0	; 0x0
int main(void)				0x00000050 <main+12>:</main+12>	str r3, [s	p]
{				≫0x00000054 <main+16>:</main+16>	mov r3, #1	442840576 ; Ox5
unsigned long i = 0;				0x00000058 <main+20>:</main+20>	add r3, r3	,#16 ; 0x10
				0x0000005c <main+24>:</main+24>	mov r2, #8	7040 ; 0x15400
GPBCON = GPB5_out GPB6_o	ut GPB7_out GPB8_out,	; // (PD	0x00000060 <main+28>:</main+28>	str rZ, [r	3]
				0x00000064 <main+32>:</main+32>	mov r0, #2	9952 ; UX7500
while (1) {				0x00000068 <main+36>:</main+36>	add ru, ru	, #40 ; UX3U
ware(30000);	// iの値に上り <u>i pp</u>	ATT2		0x00000000 <main+40>:</main+40>	DI UX10 <	442940576 · Ove
if(++i) == 16)			· 🛄	0x00000074 <main+44>:</main+44>	add r2 r2	#20 · 0x14
i = 0;				0x00000078 <main+52>:</main+52>	ldr r3. [s	, #20 , 0AI1
}			~	0x000007c <main+56>:</main+56>	lsl r3, r3	. #5
<			>	<	ш	
🔲 Console 🕅 🧔 Tasks 💽 Problems 🕻	Executables 1 Memory				8 📖 🔳 🕼	
MINI2440 Debug Culin Embedded debug (Cuguin			07/16.0.99			
MINIZ440Debug [2)III EIIIbedded debug (0)gwill	/] D:+embeded+yagar(0+biri+arm	I EII EUD.EXE (03)	07710-0.02	3		
Breaknei	:17					
17 unsigned long i	= 0;				_	
Warning: /cygdrive/D/embeded/ec	lipse/workspace/mini	2440/MINI24	40 LED:	No such file or directo:		
mi cmd disassemble: Invalid fil	ename.		-)	
<						



- ⑦ ステップ実行して Debug (ショットカットキーF6)
- a) ブレークポイントでプログラムが中断した状態から、次のブレークポイントまで実行させたり、1行ずつ実 行させたりできます。
 - コード「GPBDAT=(~(i<<5))」を繰り返して実行することにより、

LED ランプが1つずつ点滅





HanRun R911103 09/03

b)Debug 途中の ARM9 ボードの様子の1(一番右の LED ランプが点灯)





c)Debug 途中の ARM9 ボードの様子の2(右から2番目の LED ランプが点灯)



⑧ Debug を停止する (gdb、openocd をそれぞれ選択し「■」をクリック)

😂 Debug – D:¥embeded¥eclipse¥workspace¥mini2440¥MINI2440_LED¥leds.c – Ed	c lipse 📃 🗖 🔀
<u>Eile E</u> dit <u>S</u> ource Refactor <u>N</u> avigate Se <u>a</u> rch <u>P</u> roject <u>R</u> un <u>W</u> indow <u>H</u> elp	
i 📬 • 🖫 🖮 📾 i 🏇 • Ø • 🥵 • i 🥭 😂 🖋 • i 🥖 i 🖢 - २ • ♦ ♦ • •	-> - 🖹 🏇 Debug 📴 C/C++
🏇 Debug 🛛 🗧 🗖 🕅	🗏 Variables 🛞 🛛 🔠 Registers 🛋 Modules 🛛 🖑 🏘 📄 🧇 🗳 🗶 🏹 🍟 🗖
🎽 🐼 🔅 🖿 🖉 🛤 🖉 🖬 🖉 🖓 👘 💙	Name Value
🖃 🗛 openocd [Program]	(x)= i 0
D:¥embeded¥openocd¥bin¥openocdex	
⊡ ∰c MINI2440Debug [Zylin Embedded debug (Cygwin)]	
Embedded GDB (09/07/16 0:32) (Suspended)	
Inread [U] (Suspended: Breakpoint nit.)	
T mainto Difembeded Fechose # Workspace # mini2440 # MuNi2440 _ CD # Housic 19 0 T by the model of the sector # bin # arm - elf - arch ever (09/07/16 0.32)	
leds.c ☐ leds.c ⊠	T 🗖 📴 Outline 🗟 Disassembly 🛛 🗸 🗸 🖓 🖓
	Ox0000004c <main+8>: mov r3, #0 ; 0x0</main+8>
int main(void)	0x00000050 <main+12>: str r3, [sp]</main+12>
{	≫0x00000054 <main+16>: mov r3, #1442840576 ; 0x560</main+16>
unsigned long i = 0;	0x00000058 <main+20>: add r3, r3, #16 ; 0x10</main+20>
	0x0000005c <main+24>: mov r2, #87040 ; 0x15400</main+24>
GPBCON = GPB5_out GPB6_out GPB7_out GPB8_out; // GPB	Ox00000060 <main+28>: str r2, [r3]</main+28>
	0x00000064 <main+32>: mov r0, #29952 ; 0x7500</main+32>
while(1)(Ox00000068 <main+36>: add r0, r0, #48 ; 0x30</main+36>
wait(30000); approx = / ////state	UXUUUUUUGc <main+40>: bl UX18 <wait></wait></main+40>
GPBDAI = (~(1<<5)); // 10) 値により、LEDI-4と 点版できる	UXUUUUUUU/U <main+44>: mov r2, #1442840576 ; UX560</main+44>
i = 0	0x00000074 (main+40): add 12, 12, #20 ; 0x14
1 - 0,	▼ 0x0000007c <main+56>: 1s1 r3, r3, #5</main+56>
📮 Console 🕱 🗸 🖉 Tasks 🚼 Problems 🕥 Executables 📋 Memory	
MINI2440Debug [Zylin Embedded debug (Cygwin)] D.¥embeded¥yagarto¥bin¥arm-elf-gdb.exe (09/07/	16 0.32)
· · · · · · · · · · · · · · · · · · ·	^
Breakpoint 1, main () at leds.c:17	
17 unsigned long i = 0;	
Warning: /cygdrive/D/embeded/eclipse/workspace/mini2440/MINI2440_	LED: No such file or directory.
mi_cmd_disassemble: Invalid filename.	
	<u>></u>
i ∎*	



6.4.4 u-boot サンプルデバッグ

1. サンプルの導入

①プロジェクト「MINI2440_u-boot」を作る

作成方法は「4.4.3 LED サンプルデバッグ」の「①プロジェクト「MINI2440_LED」を作る」を参照

メモ:

a)u-boot のソースコードを Windows 上にコンパイルできないので、コンパイル後のファイルを提供 b)自分のソースをコンパイル場合、下記の2点修正が必要 #define CONFIG_SKIP_LOWLEVEL_INIT 1 /* SDRAMの初期化を行わない */ #define CONFIG_SKIP_RELOCATE_UBOOT 1 /*実行アドレスをロードされるため、再配置必要なし*/ 作成されたプロジェクトフォルダー構成:

(ア) コンパイル後のファイル:u-boot は elf ファイルとなる

(イ) デバッグ用の u-boot ソース

(ウ) 「init」初期化用のファイル(u-boot は SDRAM に実行されるため、初期化必要)





2. サンプル MINI2440_u-boot をデバッグ

他の設定は MINI2440_LED と同じですので、違い設定のみを説明 ① openocd 設定 「Working Directory」を「MINI2440 u-boot」に変更

External Tools Configurat	ions
Create, manage, and run cor Run a program	figurations
Image: Second system Image: Second system Image: Second system Image: Second system	Name: openocd Main Refresh & Build Environment Common Location: D*#embeded#openocd#bin#openocd.exe Browse Workspace Browse File System Varjables Working Discourcy Nerospace_loc:/MINI2440_u-boot Browse Workspace Browse File System Variables Arguments: -f D*#embeded#eclipse#workspace#config#interface#open-jtag.cfg -f D*#embeded#eclipse#workspace#config#iarget#Mini2440.cfg Variables Variables Variables Variables
Filter matched 2 of 2 items	Appl <u>y</u> Re <u>v</u> ert
?	<u>B</u> un Close



MINI2440_u-boot 用の GDB を追加 * 追加方法は「MINI2440_LED」の GDB と同じです。

Debug Configurations		
Create, manage, and run con	figurations	To the second se
Image: Second state st	Name: MINI2440_u-boot Main	Browse Search Project Browse
Filter matched 8 of 8 items		Apply Reyert
?		Debug Close



③デバッグ用のコマンドを設定(//はコメント内容となり、実際の設定に入れないでください。) target remote localhost:3333 // ローカルポート「3333」と接続(OpenOCD と接続) monitor halt//ボードの実行を中断させる monitor arm920t cp15 2 0 // MMU機能をクローズ monitor step//ステップで実行するように設定 monitor load_image D:¥¥embeded¥¥eclipse¥¥workspace¥¥mini2440¥¥MINI2440_u-boot/init/init.bin 0x0 //u-boot は SDRAM に実行されるため、初期化実行モジュールをロード(実際の init パスを入力) monitor resume 0x0//初期化モジュールを実施 monitor halt//ボードの実行を中断させる load//u-boot をロード break start_armboot//「start_armboot」関数にブレークポイントを設定 continue// プログラムを実行させて、「start_armboot」にとまってステップでデバッグ可能

Debug Configurations	
Create, manage, and run con	figurations
Image: Second state st	Name: MINI2440_u-boot
Filter matched 8 of 8 items	ApplyRevert
?	Debug Close



④デバッグ





b) gdb 起動

_ 関数「start_armboot」に止まり、ショットカットキー「F6」でステップでデバッグできる

🥃 Debug - MINI2440_u-boot/u-boot-1.1.6_OpenJTAG/lib_arm/board.c - Eclipse 📃 🗖 🗙								
<u>File Edit Source Refactor Navigate Search Project Run Window H</u> elp								
i 📬 • 📰 🗁 📾 i 🏇 • 🔾 • 💁 🖆 🔗 • i 🥖 i 🖢 - 🖓 • 🔄	• 🔿 -				😫 🕏	🏂 Debug	C/C++	
🏇 Debug 🛛 🗖 🗖	🗱 Variables 🖇	3 lili Reg	isters 🛋 Module	s) 🐁 📲	E 🕺 (fr 🗙 💥	~ - 8	
🍇 🖉 🧇 🕪 🗉 🔳 🖉 🐟 💀 🥳 👘 😿 🏟 🏹	Name			Value				
🖃 💁 openocd [Program]		. ● init_fnc_ptr			0x0000000c			
D:¥embeded¥openocd¥bin¥openocd.exe	(x)= size	(x)= size			12			
ia - ‰ Embedded GDB (09/07/22 1:24) (Suspended)								
🗐 🔐 Thread [0] (Suspended)								
⊇ 2 start_armboot0 board.c:248 0x33f81544								
□ = 1 stack_setup0 start.S:193 Ux33f8UUcU □ = D:Yewbeded¥uagarto¥bin¥erw=elf=gdbleve (00/07/22 1/23)							<u>~</u>	
							×	
	<						>	
🔓 leds.c 👔 Makefile.bak2 🚺 crt0.S 🕼 board.c 🔀 🚺 Makefile 🥍		utline 🚮 Dis	assembly 🛛				~	
#endif	► Ox	33£81540	<start_armbo< th=""><th>ot>:</th><th>push {1</th><th>:4, r5,</th><th>r6, 🔺</th></start_armbo<>	ot>:	push {1	:4, r5,	r6, 🔺	
	Ox	33f81554	<start_arms< th=""><th>1205:</th><th>sub sp</th><th>o, sp, ≴</th><th>68</th></start_arms<>	1205:	sub sp	o, sp, ≴	68	
<pre>/* Pointer is writable since we allocated a register for d = (ad t*)(armboot start - CFG MALLOC LEN - sizeof(ac</pre>	ii 🗖	ga = (g 33f81544	a_t*)(_armbo ≺start armbo	ot_start	Idr r	i. Inc.	.N -	
/* compiler optimization barrier needed for GCC >= 3.4	7 0x	33f81548	<start_armbo< th=""><th>ot+8>:</th><th>ldr r3</th><th>3, [r5]</th><th></th></start_armbo<>	ot+8>:	ldr r3	3, [r5]		
asmvolatile_("": : :"memory");	Ox	33 f 8154c	<start_armbo< th=""><th>ot+12>:</th><th>sub r3</th><th>3, r3, #</th><th>1961</th></start_armbo<>	ot+12>:	sub r3	3, r3, #	1961	
memore ((woidt) ad 0 = aireof (ad t))	Ox Ox	33£81550	<start_armbc< th=""><th>ot+16>:</th><th>sub ra</th><th>3, r3, #</th><th>136</th></start_armbc<>	ot+16>:	sub ra	3, r3, #	136	
$da_{t} = (bd t^{*})((char^{*})gd - sizeof(bd t));$	0.	memset	(void*)qd,	O, sizeof				
memset (gd->bd, U, sized (1,1)).	0.4	3340155	armoc	ot+28>:	mov r:	L, #O	;(
	0x	33£81560	<start_armbo< th=""><th>ot+32>:</th><th>mov r2</th><th>2,#36</th><th>;(</th></start_armbo<>	ot+32>:	mov r2	2,#36	;(
monitor_flash_len = _bss_startarmboot_start;		33181564 33f81568	<start_armbc <start_armbc< th=""><th>ot+36>:</th><th>hl Ov</th><th>J, I3 ∕33f970∈</th><th></th></start_armbc<></start_armbc 	ot+36>:	hl Ov	J, I3 ∕33f970∈		
			ш				>	
📮 Console 🛛 🏼 🧟 Tasks 🔝 Problems 💽 Executables 🏮 Memory			🗶 🔆 🖹 🚡	🖳 🖳	e e :	1 💷 - 👩	9 • - D	
MINI2440_u-boot [Zylin Embedded debug (Cygwin)] D.¥embeded¥yagarto¥bin¥arm-elf-gdb.exe (09	/07/22 1:23)							
Warning: /cygdrive/D/embeded/eclipse/workspace/mini2440/MINI2440_u-boot/u-boot/1.1.6_OpenJTAG/board: No such file or direct								
warning: /cygarive/D/embeded/eclipse/workspace/mini2440/MINI2440_u-boot/u-boot-1.1.6_OpenJTAG: No such file or directory. Warning: /cygdrive/D/embeded/eclipse/workspace/mini2440/MINI2440_u-boot/init: No such file or directory.								
Warning: /cygdrive/D/embeded/eclipse/workspace/mini2440/MINI2440_u-boot: No such file or directory.								
No source file named leds.c.								
							>	
	00.4.61.	Count Ing and	949 - 1	:				
	writable	omart insert	248:1					



c) ステップでデバッグ コーディング「hang()」を実行したら、MINI2440 ボードからベルを鳴る

Debug - MINI2440_u-boot/u-boot-1.1.6_Open JTAG/lib_arm/board.c - Ecli	ose						
<u>File E</u> dit <u>S</u> ource Refact <u>or N</u> avigate Se <u>a</u> rch <u>P</u> roject <u>R</u> un <u>W</u> indow <u>H</u> elp							
i 📬 • 🖫 💩 🗟 i 🏇 • 🗨 • i 🥭 😂 🖋 • i 🍠 i 🖗 • 🖗 •	•	😭 🐝 Debug 🔤 C/C++					
🏇 Debug 🛛 🗖 🗖	🗱 Variables 🛿 👫 Registers 🛋 Modules	: 🗄 🐗 📄 🍪 🗳 💥 🏹 🖓 🗖					
👋 🖉 🖉 🖿 🗉 🔳 🖄 😎 🖉 🙀 🏹	Name	Value					
🖃 💁 openocd [Program]	. ● init_fnc_ptr	0x000000c					
D:¥embeded¥openocd¥bin¥openocd.exe	(x)= size	12					
i in MINI244U u-boot [∠yin Embedded debug (Cygwin)]							
indedded CDD (05/07/22 1.24/ Couspended)							
= 2 start_armboot0 board.c:248 0x33f81544							
= 1 stack_setup0 start.S:193 0x33f800c0		~					
D:¥embeded¥yagarto¥bin¥arm-elf-gdb.exe (09/07/22 1:23)							
	3						
尾 leds.c 🕼 Makefile.bak2 🕼 crt0.S 💽 board.c 🛛 🚡 Makefile 🔭	🗖 🗖 🔚 Outline 🚮 Disassembly 🕱						
	Ox33f81540 <start armbo<="" p=""></start>	ot>: push {r4, r5, r6, 🔨					
<pre>for (init_fnc_ptr = init_sequence; *init_fnc_ptr; ++init</pre>	_f Ox33f81554 <start_armbo< td=""><td>ot+20>: sub sp, sp, #68</td></start_armbo<>	ot+20>: sub sp, sp, #68					
if ((1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1	gd = (gd_t*) (_armbo	Dt_start - CFG_MALLOC_LEN -					
hang ();	Ox33f81544 <start_armbo Ox32f81548 <start_armbo< p=""></start_armbo<></start_armbo 	ot+4>: 1dr r5, [pc, #43]					
	0x33f8154c <start armbo<="" td=""><td>ot+12>: sub r3, r3, #196</td></start>	ot+12>: sub r3, r3, #196					
	Ox33f81550 <start armbo<="" td=""><td>ot+16>: sub r3, r3, #36</td></start>	ot+16>: sub r3, r3, #36					
#ifndef CFG_NO_FLASH	0x33f81558 <start_armbo< td=""><td>ot+24>: mov r8, r3</td></start_armbo<>	ot+24>: mov r8, r3					
/* configure available FLASH banks */	memset ((void*)gd,	D, sizeof (gd_t));					
size = flash_init (); dignley flesh config (size);	Ux3318155c <start_armbo< td=""><td>ot+28>: mov r1, #U ; (</td></start_armbo<>	ot+28>: mov r1, #U ; (
#endif /* CFG NO FLASH */	0x33f81564 <start armbo<="" td=""><td>ot+36>: mov r0, r3</td></start>	ot+36>: mov r0, r3					
	0x33f81568 <start_armbo< td=""><td><u>ot.+40</u>>: bl 0x33f970e0_<t⊻< td=""></t⊻<></td></start_armbo<>	<u>ot.+40</u> >: bl 0x33f970e0_ <t⊻< td=""></t⊻<>					
🔄 Console 🛛 🧔 Tasks 🚼 Problems 🜔 Executables 🏮 Memory	🔲 🗶 💥 📄 🔒						
MINI2440_u-boot [Zylin Embedded debug (Cygwin)] D:¥embeded¥yagarto¥bin¥arm-elf-gdb.exe (09	/07/22 1:23)						
Warning: /cygdrive/D/embeded/eclipse/workspace/mini2440/NINI2440_u-boot/u-boot-1.1.6_OpenJTAG/board: No such file or direct							
warning: /cyguire///embeded/eclipse/workspace/mini2440/MINi2440_u-boot/init: No such file or directory. Warning: /cyguire///embeded/eclipse/workspace/mini2440/MINi2440_u-boot/init: No such file or directory							
Warning: /cygdrive/D/embeded/eclipse/workspace/mini2440/MINI2440 u-boot: No such file or directory.							
No source file named leds.c.							
	Writable Smart Insert 261 : 10	1					



第七章 Linux 環境上の OpenJTAG の使用手順

7.1 ハードウエア、ソフトウエアインストール

もし OpenJTAG を USB からシリアルポート変換というツールのみとして使う場合、7.1.1 節のみを参照すれば OK です。

※Linux 関連リソースダウンロード URL:

http://www.dragonwake.com/download/open-jtag/Linux.zip

本マニュアルで使われる Linux ディストリビューションは Ubuntu となります。

仮に上記 URL からダウンロードしたファイルは「/tmp」に解凍します。

7.1.1 Linux で OpenJTAG の自動認識

■udev ルールを修正すれば、Ubuntu で OpenJTAG を認識後ドライバを自動ロードします。 Ubuntu7.10 の場合、下記圧縮ファイルの中「Linux¥install¥ 50-ftdi.rules」を「/etc/udev/rules.d/」にコピ

\$sudo cp /tmp/Linux/install/50-ftdi.rules /etc/udev/rules.d/
udev ルールをすぐに有効するため、下記のコマンドを発行してください。(次回起動後、実行必要ない)
\$sudo udevcontrol reload_rules
Ubuntu8.10 以降のバージョンであれば、上記作業が必要ありません。

OpenJTAG を PC の USB と接続すれば、Ubuntu は自動認識してドライバをロードします。「/dev」フォルダデバイス 「ttyUSBO」を自動生成されます。古い Ubuntu バージョンの場合、ttyUSB1 を生成されたかも知れません。 \$ ls /dev/ttyUSB* -l crw-rw---- 1 root dialout 188, 0 2011-09-28 13:25 /dev/ttyUSB0 crw-rw---- 1 root dialout 188, 1 2011-09-28 13:25 /dev/ttyUSB1

「/dev/ttyUSB1」があれば、「/dev/ttyUSB1」を普通のシリアルポートとして使えます、ttyUSB0 のみがある場合、 /dev/ttyUSB0」を普通のシリアルポートとして使えます。

Ubuntu の中、 kermit あるいは minicom ツールでシリアルポートを操作できます。下記のコマンドでこの二つツ ールをインストールできます。

\$ sudo apt-get install ckermit
\$ sudo apt-get install minicom

kermit ツールを使用前、「/home/csun」(仮にユーザー名が csun)フォルダーに名前が「.kermrc」のファイルを作成します。内容は以下通りです。(/dev/ttyUSB1 がない場合、/dev/ttyUSB0 に変更してください。)

set line /dev/ttyUSB1 set speed 115200 set carrier-watch off set handshake none set flow-control none robust set file type bin set file name lit set rec pack 1000 set send pack 1000 set window 5

"\$ sudo kermit -c" コマンドでシリアルポート操作画面を起動できます。シリアルポートを閉じたい場合、
 "Ctrl"と "¥"を同時押し、その後、"Ctrl"を押さなくて、"exit"を入力して Enter キーを押します。



minicom を使う場合、使用前に"minicom -s"コマンドで設定画面に入ります、"Serial port setup"を選べ、下 図のように設定を行いましょう。その後、"Save setup as dfl"を選択(/dev/ttyUSB1 がない場合、/dev/ttyUSB0 に変更)



minicom 設定画面

上記設定を行った後、直接 minicom コマンドを発行すれば、シリアルポート操作画面を起動できます。操作画面 んを閉じたい場合、"Ctrl"と "A"を同時押し、その後、離して "X"を押します。

7.1.2 OpenOCD、GDB、クロスコンパイルチェイン、Eclipse のインストール

1. OpenOCD、GDB、oflashのインストール

Linux¥install フォルダの直下の openocd. tar. bz2 を/に解凍して、arm-linux-gdb、oflash を/usr/bin にコピーし てください。

そして、所属のユーザーを root に設定、実行可能の属性も追加:

\$ sudo tar xjf /tmp/Linux/install/openocd.tar.bz2 -C /

\$ sudo cp /tmp/Linux/install/arm-linux-gdb /usr/bin/

\$ sudo cp /tmp/Linux/tools/oflash /usr/bin/

\$ sudo chown root:root /usr/local/bin/openocd /usr/bin/arm-linux-gdb /usr/bin/oflash

\$ sudo chmod +xs /usr/local/bin/openocd /usr/bin/arm-linux-gdb /usr/bin/oflash

注意:必ず/usr/bin/arm-linux-gdb を実行権限"x"、スーパーユーザー権限"s"を追加

2. クロスコンパイルチェインのインストール

もし、Ubuntu 上にクロスコンパイルチェインをインストールされない場合("arm-linux-gcc -v"コマンドで確認、出力内容がない場合、インストールされないという事です。)

Linux/install/arm-linux-gcc-3.4.5-glibc-2.3.6.tar.bz2 をあるフォルダに解凍してから PATH 変数を設定。 例 :

\$ cp /tmp/Linux/install/arm-linux-gcc-3.4.5-glibc-2.3.6.tar.bz2 /work/tools

\$ cd /work/tools

\$ tar xjf arm-linux-gcc-3.4.5-glibc-2.3.6.tar.bz2

「/etc/environment」を編集して PATH 環境変数を修正しましょう。(赤字が追加)

PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/work/tools/gcc-3.4.5-gl



ibc-2.3.6/bin"

次回起動後、直接クロスコンパイルチェインを使えます、すぐ使いたい場合、下記コマンドで発行します。 \$ export PATH=\$PATH:/work/tools/gcc-3.4.5-glibc-2.3.6/bin

Eclipse のインストール
 ①JAVA のインストール(以前の SUN、今の Oracle 社)
 Ubuntu 上に既に GNU バージョンの Java をインストールされますが、
 このバージョンの Java は機能上 SUN Java より弱いですので、Eclipse のあるプラグインは必ず SUN Java 上に使えますので、下記のコマンド SUN Java をインストールしましょう。
 \$sudo apt-get install sun-java6-jre
 \$sudo apt-get install sun-java6-jdk
 注意: リモートツールで上記コマンドを実行しないでください。インストール途中で以下の二つ画面が出てきます、
 リモートツールで GUI 画面をサポートしないです。

上記一番のコマンドを実行する時、下記画面が出てきます、Ok、Yes を選択してください。

Package configuration

– Configuring sun-java6-bin 🗕

Operating System Distributor License for Java v1.1 (DLJ)

Operating System Distributor License for Java version 1.1 (DLJ)

SUN MICROSYSTEMS, INC. ("SUN") IS WILLING TO LICENSE THE JAVA PLATFORM STANDARD EDITION DEVELOPER KIT ("JDK" - THE "SOFTWARE") TO YOU ONLY UPON THE CONDITION THAT YOU ACCEPT ALL OF THE TERMS CONTAINED IN THIS LICENSE AGREEMENT (THE "AGREEMENT"). PLEASE READ THE AGREEMENT CAREFULLY. BY INSTALLING, USING, OR DISTRIBUTING THIS SOFTWARE, YOU ACCEPT ALL OF THE TERMS OF THE AGREEMENT.

 DEFINITIONS. "Software" means the code identified above in binary form, any other machine readable materials including, but not limited to, libraries, source files, header files, and data files), any updates or error corrections provided by Sun, and any user manuals, programming guides and other documentation provided to you by Sun under this Agreement, and any subsequent versions that Sun makes available to you hereunder. "Operating System" means any version of the Linux or OpenSolaris operating systems that manages the hardware resources of a general purpose desktop or server

<0k>



株式会社日昇テクノロジー

Configuring sun-java6-bin 🛏 In order to install this package, you must accept the license terms, the "Operating System Distributor License for Java" (DLJ), v1.1. Not accepting will cancel the installation. Do you accept the DLJ license terms? <N0><Yes> 以下のコマンドで上記インストールされた sun-java6 をデフォルトの Java に設定しましょう。 \$sudo update-alternatives --config java book@book-desktop:/var/lib/dpkg/info\$ sudo update-alternatives --config java There are 2 choices for the alternative java (providing /usr/bin/java). Selection Path Priority Status /usr/lib/ivm/iava-6-openidk/ire/bin/iava auto mode * 0 1061 /usr/lib/jvm/java-6-openjdk/jre/bin/java 1061 manual mode 1 2 /usr/lib/jvm/java-6-sun/jre/bin/java 63 manual mode Press enter to keep the current choice[*], or type selection number: 2 注意: Ubuntu 上に Java をインストールした事がない場合、下記の様なメッセージがでてきます。 There is only 1 program which provides java (/usr/lib/jvm/java-6-sun/jre/bin/java). Nothing to configure. ②Eclipse のインストール /tmp/Linux/install/eclipse-cpp-helios-SR1-linux-gtk.tar.gz を/optに解凍します、そして、属性も下記のよう に変更してください。 \$ cd /tmp/Linux/install \$ sudo tar xzf eclipse-cpp-helios-SR1-linux-gtk.tar.gz -C /opt \$ cd /opt \$ sudo chmod 775 eclipse -R スクリプトファイル「/tmp/Linux/install/eclipse」を「/usr/bin」にコピーし、実行可能の属性も追加します。 \$ sudo cp /tmp/Linux/install/eclipse /usr/bin \$ sudo chmod 775 /usr/bin/eclipse 最後、メニューに Eclipse のショットカットも追加しましょう。 \$ sudo cp /tmp/Linux/install/eclipse.desktop /usr/share/applications \$ sudo chmod +r /usr/share/applications/eclipse.desktop インストール結果を確認 \$ openocd -v \$ arm-linux-gdb -v \$ arm-linux-gcc -v \$ iava -version メニューから Eclipse を起動しましょう。



- Applications Pla	aces	System 🕹 🖳 🕢	Workspace Launcher					
🛄 Accessories	•		Select a workspace					
🚯 Games	•		Eclipse SDK stores your projects in a folder called a workspace. Choose a workspace folder to use for this session.					
🔏 Graphics	•		Workspace: /work/eclipse projects					
🅪 Internet	•							
📊 Office	•							
🥖 Programming	•	Eclipse Platform	□ Use this as the default and do not ask again					
🧓 Sound & Video	-							
🍪 System Tools	•		OK R Cancel					
🐻 Add/Remove			related trademarks and logos are trademarks or registered trademarks of som morosystems, Inc. in the U.S., other countries, or both. Eclipse is a trademark of the Eclipse Foundation, Inc.					

7.2 OpenJTAGの使用

環境を構築した後、使用方法は Windows と同じです、このマニュアルの Windows 部分をご参照ください。