

HJ-LINK/USB

ARM コア CPU のデバック方法

1版 2010年06月07日

1. 概要

1.1 概要

HJ-LINK/USB は、オープンソースの OpenOCD (Open On-Chip Debugger) を利用して、ARM コア CPU をデバックすることができます。OpenOCD は、「JTAGKey」という JTAG ケーブルに対応しています。JTAGKey は Amontec 社の製品ですが、回路図が公開されているため、各社から互換品が発売され、OpenOCD を使った ARM のデバック環境もさまざまなサイトで公開されています。HJ-LINK/USB も、JTAGKey 互換のハードウェアとなっており、JTAGKey に対応したデバック環境であれば、基本的に利用できます。

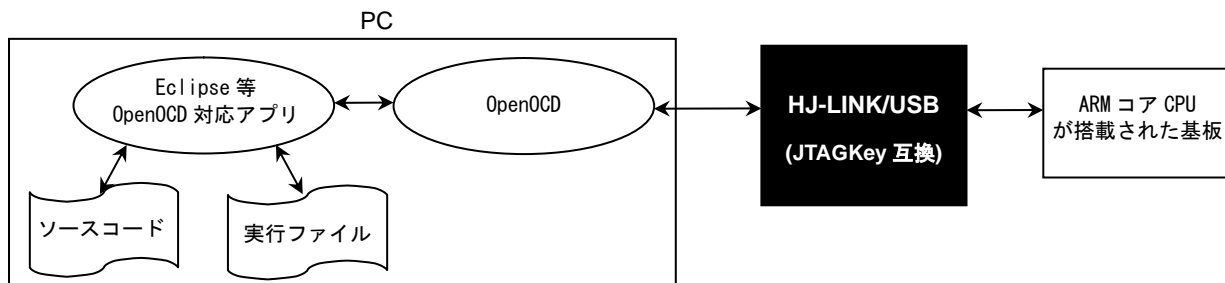
本書では OpenOCD と eclipse を使ったデバック環境の構築方法について解説していますが、「OpenOCD」や「JTAGKey」に関連する情報は、インターネットで多数の情報が公開されておりますので、それらも参考に自分にあつた環境を構築してみてください。また、コンパイラ環境については本書では特に解説しておりませんが、こちら、技術雑誌のほか、インターネットでも簡単に情報を得られますので、それらを参考にしてください。

【OpenOCD】 オープンソースの組み込みシステムのインシステムプログラミングとバウンダリスキャンテストを実現するためのソフトウェア。eclipse 等と組み合わせて、JTAG デバックやフラッシュメモリの書き込み等が可能となる。

参考文献 CQ 出版社 インタフェース 2009 年 4 月号 「組み込み C プログラミングを基本から攻略する！」
CQ 出版社 インタフェース 2009 年 5 月号 「付属 ARM 基板で学ぶ実践マイコン活用入門」
CQ 出版社 インタフェース 2009 年 6 月号 「ARM マイコン基板をとことん使いこなそう！」
CQ 出版社 インタフェース 2010 年 5 月号 「リアルタイム OS を使おう！ビルドで学ぶソフト開発」

* 本書、参考文献の内容に基づいて使用し、損害が生じても参考文献先および(株)アルファプロジェクトは一切の責任は負いませんのでご了承ください。

OpenOCD を使ったデバック環境



1.2 対応デバイス

OpenOCD は、以下の ARM コア CPU に対応しています。

対応デバイス*1
ARM7、ARM9、ARM11、Cortex-A8、Cortex-M3

*1 2010 年 4 月時点

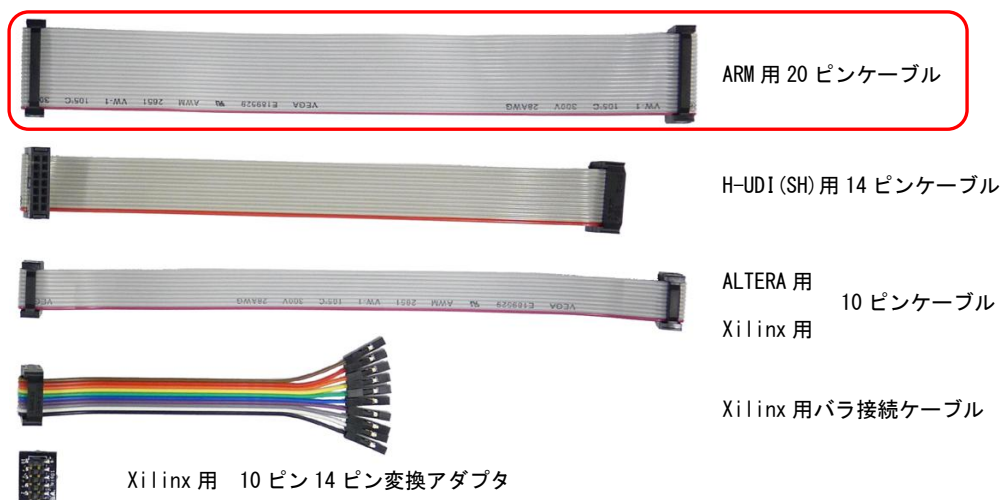
HJ-LINK/USB では、以下のデバイスにて動作確認を行いました。

弊社確認済みデバイス
NXP Semiconductors 社製 ARM7TDMI コア CPU : LPC2388*2

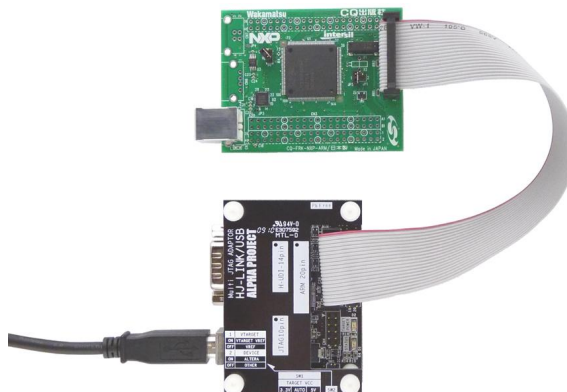
*2 本書は、Interface 誌 2009 年 5 月号付属 ARM7TDMI コア LPC2388 基板を使用して説明します。

1.3 準備物

- ・ HJ-LINK/USB
- ・ USB ケーブル
- ・ ARM コア CPU が搭載された基板（以降ターゲットボードと記載）
- ・ JTAG ケーブル（ARM 用 20 ピンケーブル）



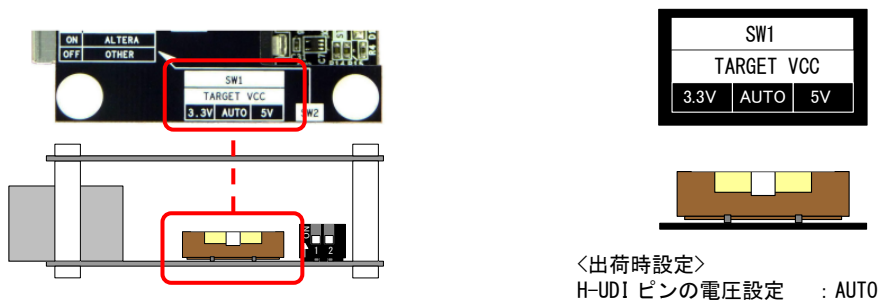
下図は、HJ-LINK/USB とターゲットボードを JTAG ケーブル（ARM 用 20 ピンケーブル）で接続した図です。



1.4 スイッチの設定

1.4.1 JTAG インタフェース電圧の設定

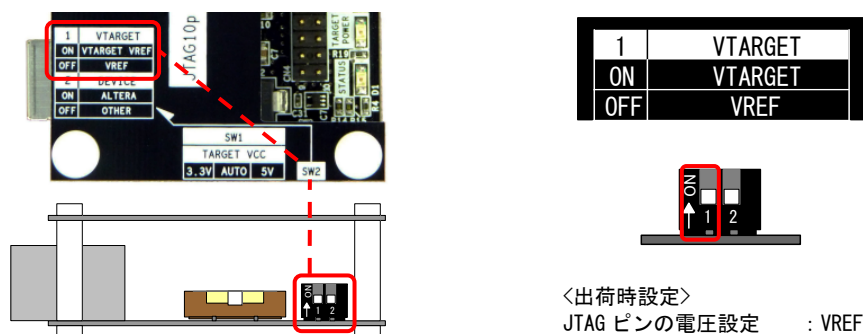
HJ-LINK/USB は、使用するターゲット CPU またはターゲットデバイスの JTAG インタフェース電圧の設定を SW1 で行います。JTAG インタフェース電圧は、5V と 3.3V の設定の他に、ターゲットボードより電源を供給する AUTO があります。AUTO に設定する場合には、必ず使用するインタフェースの JTAG コネクタの 1 ピンをターゲット電圧に接続してください。



SW1	H-UDI インタフェース電圧	備考
3.3V	3.3V 固定	
AUTO	H-UDI 端子の 1 ピンより電源を供給	出荷時設定
5V	5V 固定	

1.4.2 VTARGET/VREF の設定

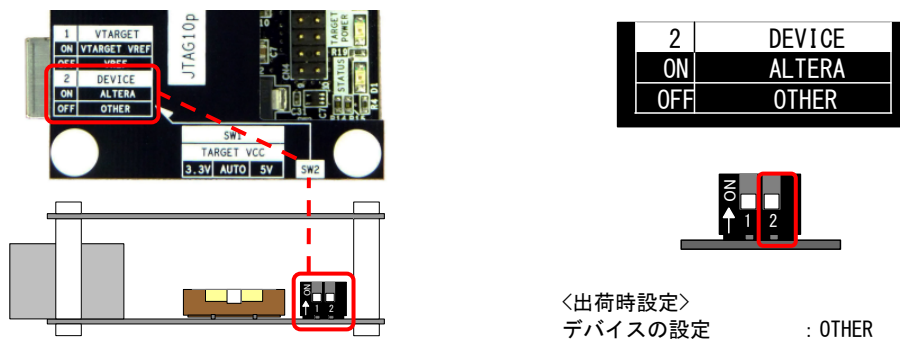
本設定は ARM の JTAG インタフェースを使用する場合にのみ必要な設定です。ARM の JTAG インタフェースのうち、VTARGET 端子 (2 ピン) と VREF 端子 (1 ピン) があります。これらの端子の短絡/未短絡の設定を SW2-1 で行います。



SW2-1	VTARGET/VREF	UVCC との関係	備考
VTARGET	VTARGET 端子と VREF 端子を短絡	UVCC = VTARGET = VREF	
VREF	VTARGET 端子と VREF 端子を未短絡	UVCC = VREF	出荷時設定

1.4.3 デバイスの設定

HJ-LINK/USB では、接続する CPU またはデバイスに応じて SW2-2 を設定する必要があります。ARM コア CPU と接続する場合は、「OTHER」に設定してください。



SW2-2	接続する CPU またはデバイス	備考
ALTERA	ALTERA 製のデバイス	
OTHER	ALTERA 製以外のデバイスまたは CPU	出荷時設定

2. ARM コア CPU のデバック方法

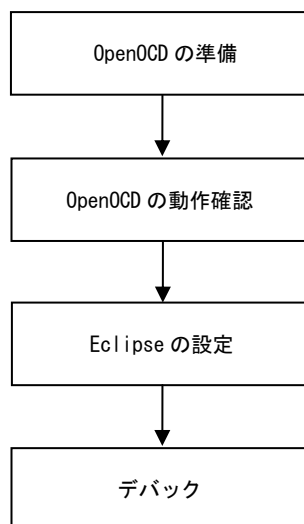
2.1 ソフトウェアの準備

2.1.1 ソフトウェアのインストール

SH-2 のデバックをするには以下のソフトウェアをインストールしておいてください。

- Java (Sun Microsystems 社)
(Eclipse を動作させるのに必要となります。)
- HJ-LINK/USB の USB デバイスドライバ (FTDI 社)
(USB ドライバのインストール方法については、HJ-LINK/USB の取扱説明書を参照してください。)

ARM コア CPU のデバック手順は下図のようになります。



2.2 OpenOCD の準備

HJ-LINK/USB は USB⇔JTAG 変換に FTDI 社の FT2232H デバイスを使用しています。

OpenOCD から入手できる最新のバイナリパッケージは、FTDI 社のドライバで動作させることができません。

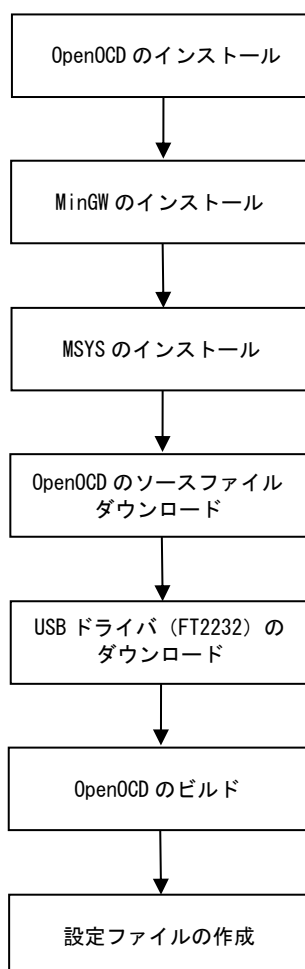
これは FTDI 社の配布しているライブラリが GPLv2 に準拠していないことに由来します。

このためバージョン 0.2.0 以降の OpenOCD では、FTDI 社のライブラリを使用せず、GPLv2 に準拠したフリーの独自ライブラリを使用しています。フリーの独自ライブラリは、OpenOCD 付属のドライバで動作します。

HJ-LINK/USB では、ARM 以外の環境では FTDI 社のドライバが必要なため、OpenOCD 付属のドライバを使うことはできません。

従って FTDI 社のドライバで動作するように、OpenOCD のソースを再ビルドする必要があります。

OpenOCD のビルド手順は以下のようになります。



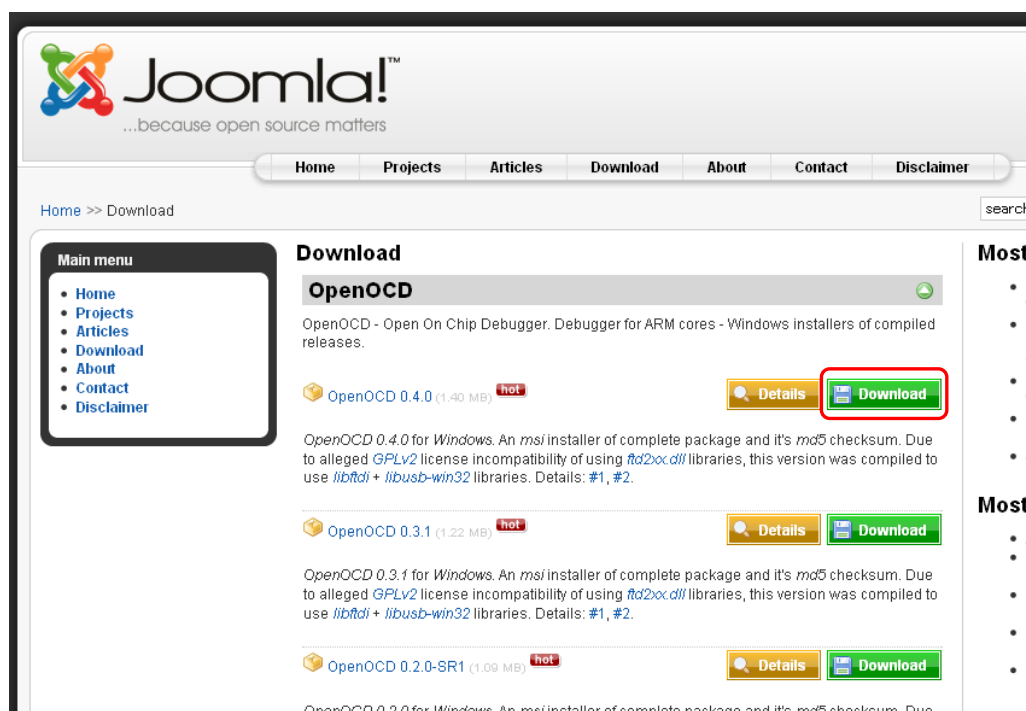
2.2.1 OpenOCD のインストール

OpenOCD にてバイナリパッケージをダウンロードし、インストールします。

- ① OpenOCD^{*1} をダウンロードしてください。OpenOCD のバージョンは 0.4.0 を使用して説明します。

OpenOCD 0.4.0 ダウンロードページ

<http://www.freddiechopin.info/index.php/en/download/category/4-openocd>



The screenshot shows the Joomla! website interface. At the top, the Joomla! logo and tagline "...because open source matters" are visible. Below the logo is a navigation menu with links for Home, Projects, Articles, Download, About, Contact, and Disclaimer. The "Download" link is highlighted. On the left side, there is a "Main menu" with a list of links: Home, Projects, Articles, Download, About, Contact, and Disclaimer. The "Download" link is selected. The main content area is titled "Download" and "OpenOCD". It lists three versions of OpenOCD for Windows: 0.4.0 (1.40 MB), 0.3.1 (1.22 MB), and 0.2.0-SR1 (1.09 MB). Each version has a "Details" button and a "Download" button. The "Download" button for the 0.4.0 version is highlighted with a red box. On the right side, there are two "Most" sections with lists of links.

- ② ダウンロードした圧縮ファイルを解凍します。
解凍したフォルダにある「openocd.msi」を実行してください。



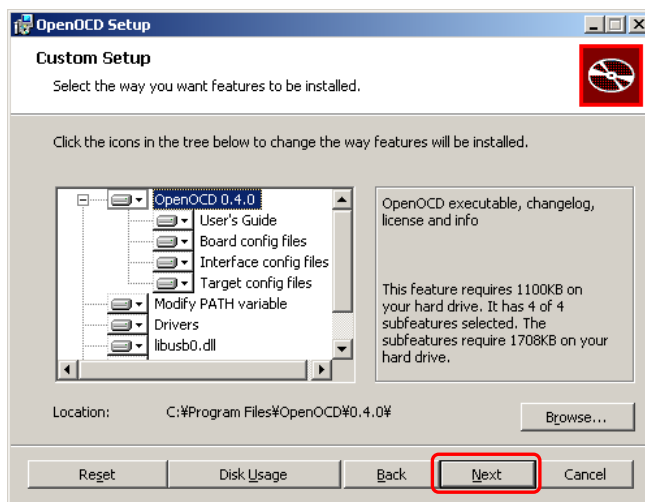
- ③ インストーラが起動するので、「Next」を選択します。



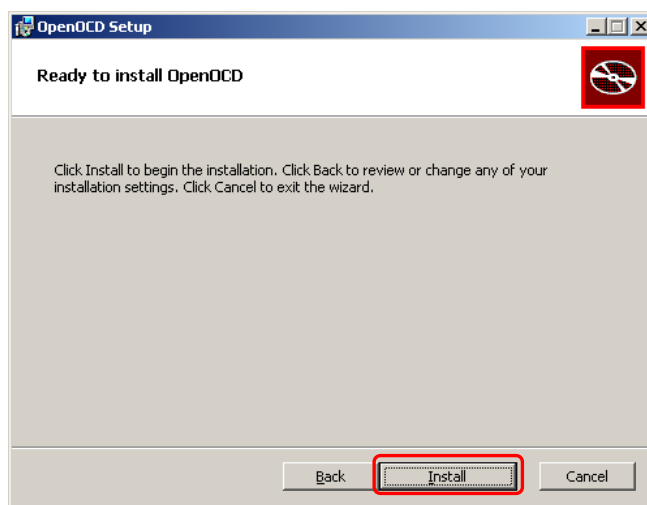
- ④ 使用条件の同意画面が表示されるので、最後まで読み同意できたら「I accept the terms in the License Agreement」にチェックを入れて、「Next」を選択します。



- ⑤ Custom Setup 画面が表示されるので、「Next」を選択します。

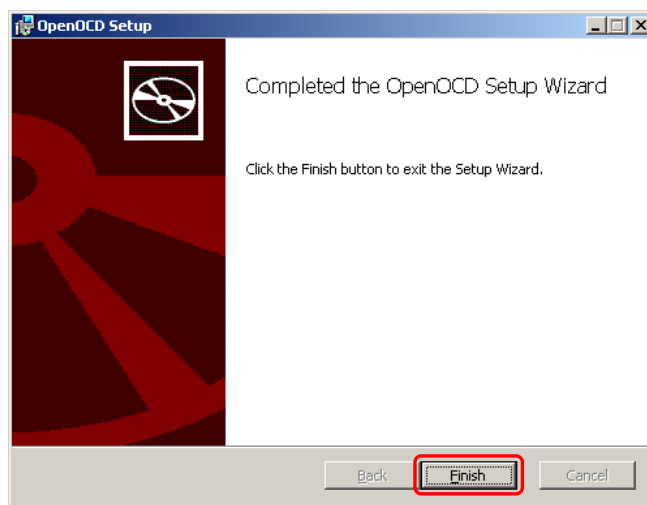


- ⑥ インストール確認画面が表示されるので、「Install」を選択しインストールを開始します。
OpenOCDは、プログラムファイルパス以下にインストールされます。*1



*1 Windows の標準状態であれば C:\Program Files\OpenOCD\0.4.0\bin

- ⑦ インストールの完了通知が表示されるので、「Finish」を選択してインストーラを終了します。



- ⑧ OpenOCD のインストールは終了です。

2.2.2 MinGW のインストール

C コンパイラを使用するために MinGW をダウンロードし、インストールします。

- ① MinGW をダウンロードしてください。MinGW のバージョンは MinGW 5.1.6 を使用して説明します。
 メニューから「Navigation」→「About」→「Downloads」をクリックしてください。
 Downloads ページにある「Automated MinGW Installer」→「MinGW 5.1.6」をクリックし「MInGW-5.1.6.exe」をダウンロードします。

MinGW 5.1.6 ダウンロードページ

<http://www.mingw.org/>

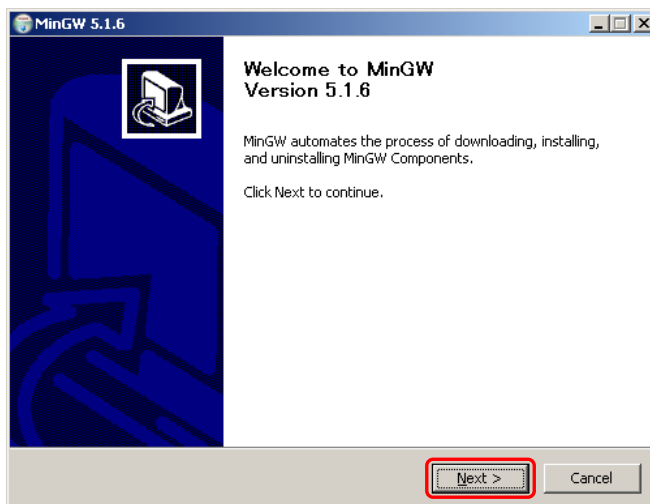
▶ GNU Source-Level Debugger	269.9 MB	2010-03-21	726,453		
▶ MSYS Base System	130.0 MB	2010-03-18	1,437,330		
▶ Cross-Hosted MinGW Build Tool	89.7 KB	2010-03-10	47,400		
▶ GNU Binutils	132.2 MB	2010-03-10	757,150		
▶ MinGW Runtime	7.3 MB	2010-03-07	2,107,340		
▼ Automated MinGW Installer	1.3 MB	2010-02-16	1,623,930		
▶ mingw-get	570.4 KB	2010-02-16	3,691		
▼ MinGW 5.1.6	176.1 KB	2009-10-02	460,548		
<input checked="" type="checkbox"/> MinGW-5.1.6.exe	158.8 KB	2009-10-02	457,401		
<input type="checkbox"/> MinGW-5.1.6.exe-src.tar.lzma	17.2 KB	2009-09-25	3,147		
▶ MinGW 5.1.4	508.6 KB	2008-04-27	1,159,691		
▶ MSYS binutils	17.2 MB	2010-01-16	4,761		
▶ MSYS gcc	20.1 MB	2010-01-16	2,809		
▶ MSYS w32api	2.4 MB	2010-01-15	5,267		
▶ MSYS xz	20.6 MB	2010-01-13	10,483		

- ② ダウンロードした実行ファイル「MinGW-5.1.6.exe」を実行してください。
 MinGW のインストールには、PC がネットワークに接続されている必要があります。



MinGW-5.1.6.exe

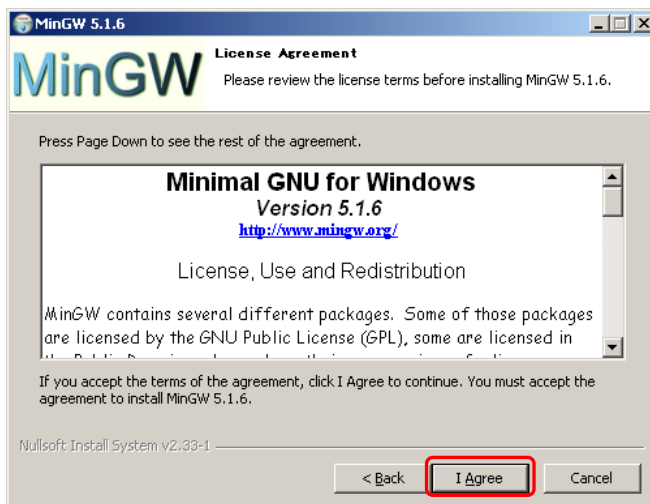
- ③ インストーラが起動するので、「Next」を選択します。



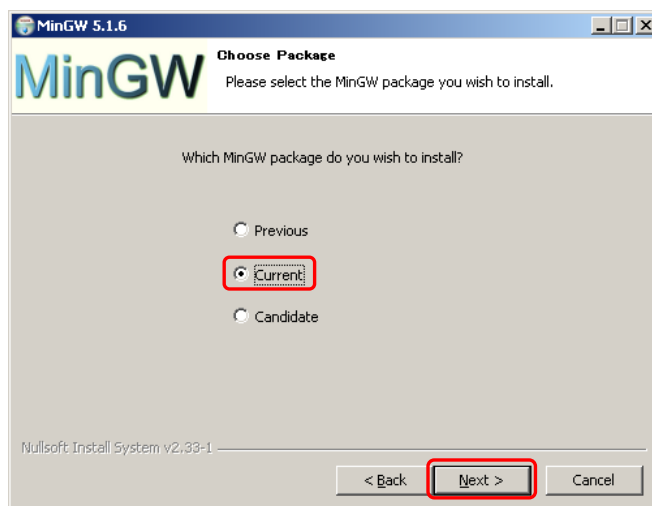
- ④ ダウンロードとインストールの設定画面が表示されるので、「Download and install」を選択し「Next」を選択します。



- ⑤ 使用条件の同意画面が表示されるので、最後まで読み同意できたら「I Agree」を選択します。

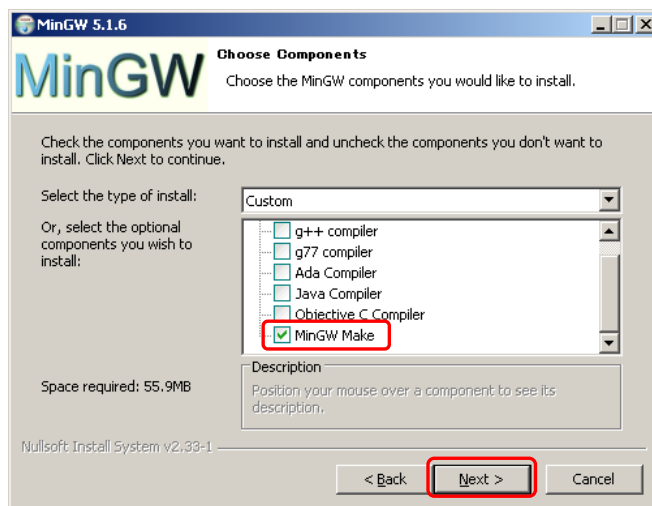


- ⑥ パッケージ選択画面が表示されるので、「Current」を選択し「Next」を選択します。

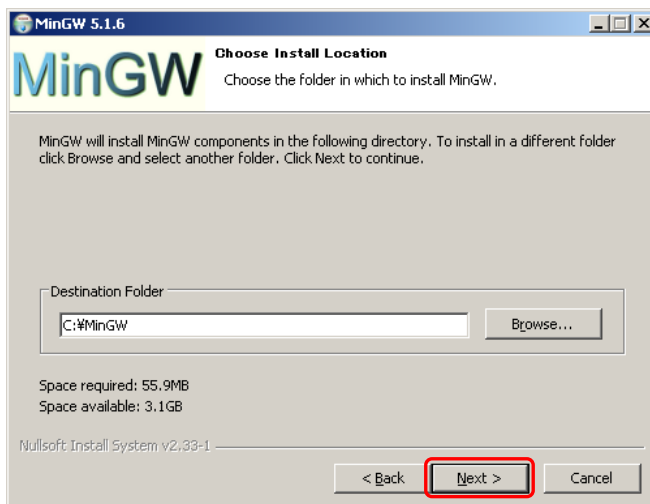


- ⑦ コンポーネントの選択画面が表示されるので、「MinGW base tools」と「MinGW Make」にチェックを入れ「Next」を選択します。

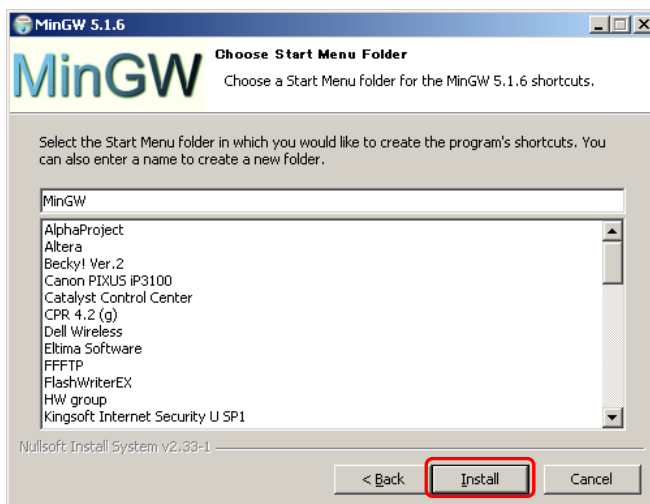
画面には表示されておませんが、「MinGW base tools」にチェックがデフォルトで入っています。



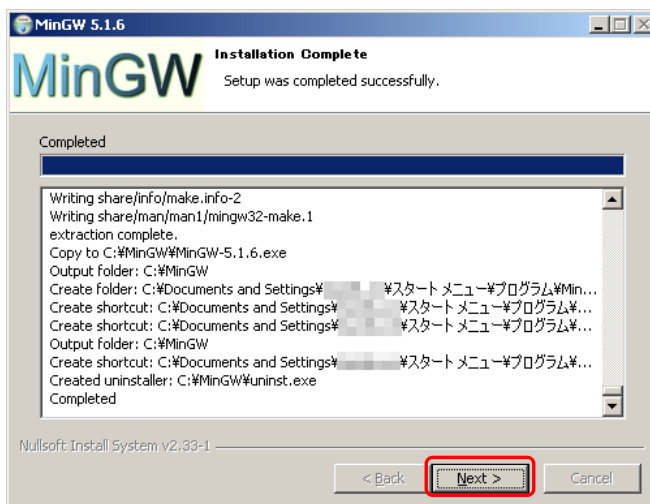
- ⑧ インストール先のフォルダを指定し、「Next」を選択します。



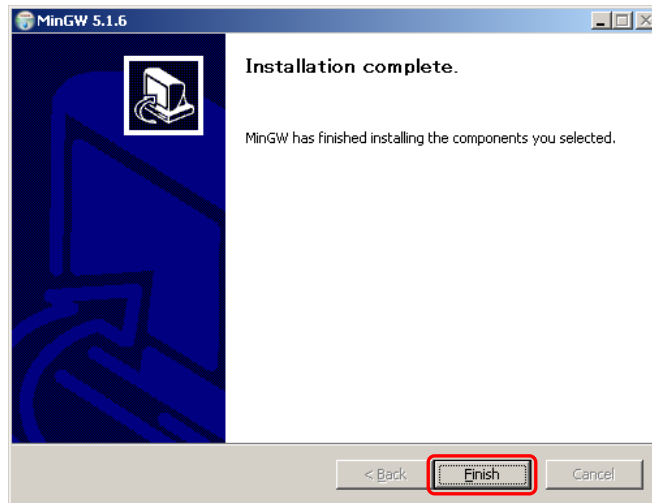
- ⑨ スタートメニューフォルダの選択をし、「Install」を選択します。



- ⑩ インストールしたファイル一覧が表示さるので、「Next」を選択します。



- ⑪ インストールの完了通知が表示されるので、「Finish」を選択してインストーラを終了します。



- ⑫ MinGW のインストールは終了です。

2.2.3 MSYS のインストール

「2.1.2 MinGW のインストール」でインストールした MinGW を使用するために MSYS をダウンロードし、インストールします。

- ① MSYS をダウンロードしてください。MSYS のバージョンは MSYS 1.0.11 を使用して説明します。
ページにある「Install MSYS 1.0.11.」をクリックし「MSYS-1.0.11.exe」をダウンロードしてください。

MSYS-1.0.11 ダウンロードページ

<http://www.mingw.org/wiki/MSYS>

For convenience, you can follow the instructions below to install 1.0.11. It currently requires less work, but the MSYS components installed are not all up-to-date.

Previous MSYS versions (up to 1.0.11)

These instructions were based on the [Enlightenment Wiki](#). Thanks to Vincent Torri for pointing them out.

The total size of the installation of MSYS/MinGW is around 110 MB. Be sure to have enough space on your hard disk.

- If you haven't already installed MinGW on your system, install [MinGW](#) in C:\MinGW. It is better to not install it in the same directory than MSYS, though there should be no problem since MSYS 1.0.11. In the installer, choose "Download and install", then "Current" (it will install gcc 4.4.0).
- Install **MSYS 1.0.11**. I usually install it in C:\msys\1.0, but you can use any directory that you prefer.

Check <http://sourceforge.net/projects/mingw/files/> for more recent versions of all these files.

- Next, the post install process will ask for the directory where MinGW was installed to. Enter "c:/mingw". If you make a mistake, you can change it by editing the "C:\msys\1.0\etc\fstab" file, make sure to use `\r\n` line endings. An example fstab may contain:

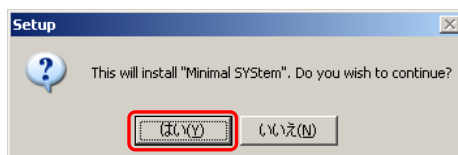
```
c:/mingw /mingw
c:/java /java
```

- Install [MSYS DTK 1.0](#) in C:\msys\1.0.
- Install [MSYS Core 1.0.11](#). It is an archive. Untar it in C:\msys\1.0.

- ② ダウンロードした実行ファイル「MSYS-1.0.11.exe」を実行します。



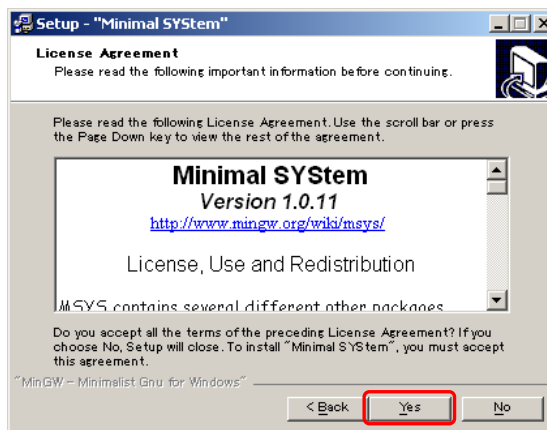
- ③ インストール確認のダイアログが表示されるので、「はい」を選択します。



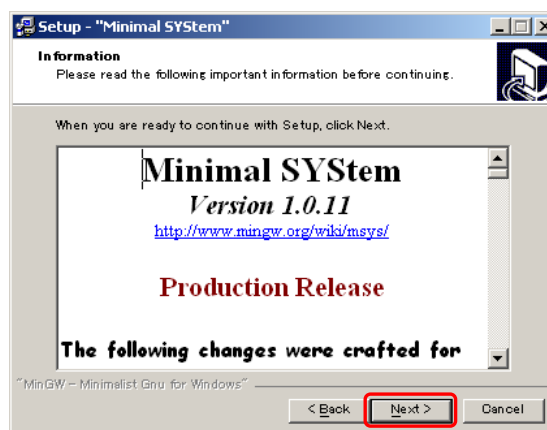
- ④ インストーラが起動するので、「Next」を選択します。



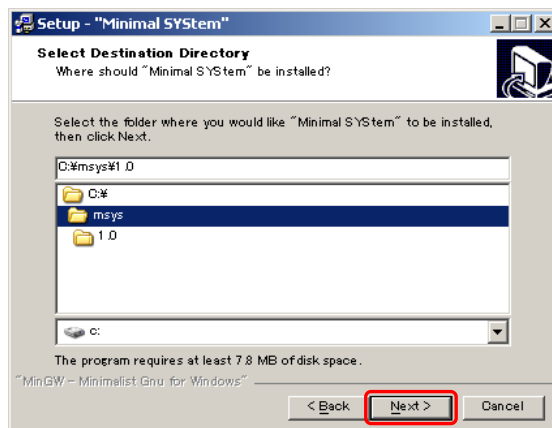
- ⑤ 使用条件の同意画面が表示されるので、最後まで読み同意できたら「Yes」を選択します。



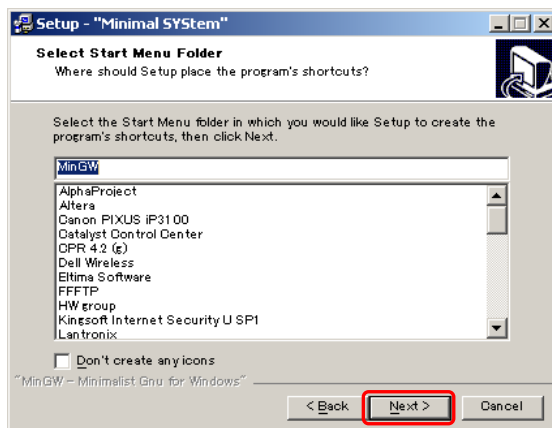
- ⑥ インフォメーション画面が表示されるので、最後まで読み「Next」を選択します。



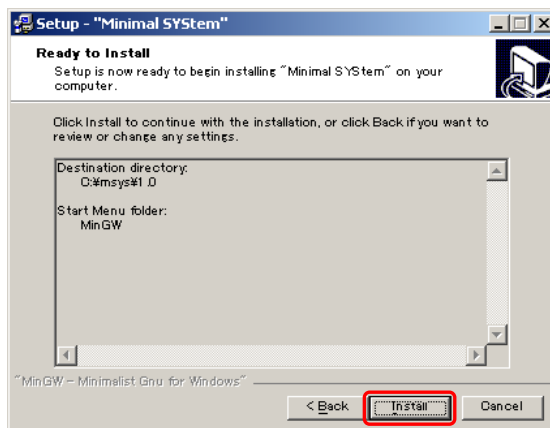
- ⑦ インストール先のフォルダを指定し、「Next」を選択します。



- ⑧ スタートメニューフォルダの選択をし、「Next」を選択します。



- ⑨ インストール確認画面が表示されるので、「Install」を選択します。



Command プロンプトが出て MinGW をインストールしたか聞いてきます。

「2.1.2 MinGW のインストール」を行っている場合は、「y」→「y」→「C:\MinGW」と入力してください。入力が終わると「続行するには何かキーを押してください...」と表示されますので何かキーを押して終了します。

```
This is a post install process that will try to normalize between
your MinGW install if any as well as your previous MSYS installs
if any. I don't have any traps as aborts will not hurt anything.
Do you wish to continue with the post install? [yn ] y

Do you have MinGW installed? [yn ] y

Please answer the following in the form of c:/foo/bar.
Where is your MinGW installation? C:\MinGW

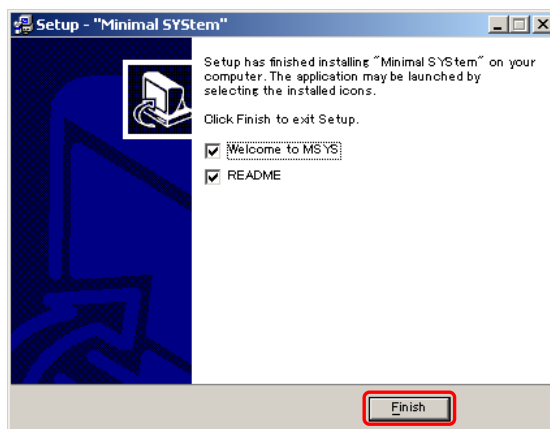
Creating /etc/fstab with mingw mount bindings.
Normalizing your MSYS environment.

You have script /bin/awk
You have script /bin/cmd
You have script /bin/echo
You have script /bin/egrep
You have script /bin/fgrep
You have script /bin/printf
You have script /bin/pwd

Oh joy, you do not have C:\MinGW\bin\make.exe. Keep it that way.

C:\msys\1.0\postinstall>pause
続行するには何かキーを押してください . . .
```

- ⑩ インストールの完了通知が表示されるので、「Finish」を選択してインストーラを終了します。



- ⑪ profile の確認を行います。C:\msys\1.0\etc にある「profile」ファイルをテキストエディタで開き 19 行目が下記のようにになっているか確認します。なっていない場合は下記のように修正します。

```
export PATH=".:usr/local/bin:/c/mingw/bin:/bin:$PATH"
```

- ⑫ デスクトップに作成された MSYS のショートカットを実行し、MSYS を起動します。
実行するとプロンプトが出てきますので gcc が動くか確認します。「gcc --version」と入力し gcc のバージョンが表示されているか確認してください。



```
user@PC
$ gcc --version
gcc.exe (GCC) 3.4.5 (mingw-vista special r3)
Copyright (C) 2004 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.
```

- ⑬ OpenOCD をビルドするためのディレクトリを作成します。
C:\msys\1.0\home の下に openocd ディレクトリ、その下に trunk と ftd2xx ディレクトリを作成してください。

```
user@PC
$ cd /c/msys/1.0/home/

user@PC /c/msys/1.0/home
$ mkdir openocd

user@PC /c/msys/1.0/home
$ cd openocd/

user@PC /c/msys/1.0/home/openocd
$ mkdir trunk

user@PC /c/msys/1.0/home/openocd
$ mkdir ftd2xx

user@PC /c/msys/1.0/home/openocd
$ ls
ftd2xx trunk
```

- ⑭ MSYS のインストールは終了です。

2.2.4 MSYS で使用するコマンドのインストール
















MSYS をインストールしただけでは、Makefile を作成する configure が動作しませんので必要なコマンドをインストールします。必要なコマンドを以下に示します。

```
autoconf
automake
libtool
perl
libtool_dll
crypt_dll
```

MSYS で使用するコマンドのダウンロードページ

<http://www.mingw.org/wiki/MSYS>

- ① メニューから「Navigation」→「About」→「Downloads」をクリックします。リンク先のページにある「MSYS autoconf」→「autoconf-2.63-1」→「autoconf-2.63-1-msys-1.0.11-bin.tar.lzma」をクリックしてダウンロードしてください。以降、保存先フォルダを「C:\msys\1.0」として説明します。

▶ MSYS termcap	71.6 KB	2009-08-01	5,173		
▶ MSYS automake	1.8 MB	2009-08-01	9,551		
▼ MSYS autoconf	2.0 MB	2009-08-01	10,819		
▼ autoconf-2.63-1	2.0 MB	2009-08-01	10,819		
autoconf-2.63-1-msys.RELEASE_NOTES	4.4 KB	2009-08-01	1,250		
autoconf-2.63-1-msys-1.0.11-src.tar.lzma	1.2 MB	2009-08-01	898		
autoconf-2.63-1-msys-1.0.11-lic.tar.lzma	14.9 KB	2009-08-01	656		
autoconf-2.63-1-msys-1.0.11-doc.tar.lzma	523.0 KB	2009-08-01	880		
autoconf-2.63-1-msys-1.0.11-bin.tar.lzma	242.0 KB	2009-08-01	7,135		
▶ MSYS perl	10.8 MB	2009-08-01	11,463		
▶ MSYS crypt	35.1 KB	2009-08-01	14,247		

- ② リンク先のページにある「MSYS automake」→「automake-1.11-1」→「automake-1.11-1-msys-1.0.11-bin.tar.lzma」をクリックし「C:\msys\1.0」にダウンロードします。

▶ MSYS libtool	2.3 MB	2009-08-01	12,050		
▶ MSYS termcap	71.6 KB	2009-08-01	5,173		
▼ MSYS automake	1.8 MB	2009-08-01	9,551		
▼ automake-1.11-1	1.8 MB	2009-08-01	9,551		
automake-1.11-1-msys.RELEASE_NOTES	4.7 KB	2009-08-01	979		
automake-1.11-1-msys-1.0.11-src.tar.lzma	1.1 MB	2009-08-01	781		
automake-1.11-1-msys-1.0.11-lic.tar.lzma	6.6 KB	2009-08-01	598		
automake-1.11-1-msys-1.0.11-doc.tar.lzma	423.4 KB	2009-08-01	823		
automake-1.11-1-msys-1.0.11-bin.tar.lzma	297.2 KB	2009-08-01	6,370		
▶ MSYS autoconf	2.0 MB	2009-08-01	10,819		
▶ MSYS perl	10.8 MB	2009-08-01	11,463		

- ③ リンク先のページにある「MSYS libtool」→「libtool-2.2.7a-1」→「libtool-2.2.7a-1-msys-1.0.11-bin.tar.lzma」をクリックし「C:\msys\1.0」にダウンロードします。

▶ MSYS gettext	15.1 MB	2009-08-01	8,341		
▶ MSYS libiconv	5.3 MB	2009-08-01	8,588		
▼ MSYS libtool	2.3 MB	2009-08-01	12,050		
▼ libtool-2.2.7a-1	2.3 MB	2009-08-01	12,050		
libtool-2.2.7a-1-msys.RELEASE_NOTES	5.6 KB	2009-08-01	1,066		
libtool-2.2.7a-1-msys-1.0.11-src.tar.lzma	1.7 MB	2009-08-01	637		
libtool-2.2.7a-1-msys-1.0.11-lic.tar.lzma	6.6 KB	2009-08-01	748		
libtool-2.2.7a-1-msys-1.0.11-doc.tar.lzma	291.4 KB	2009-08-01	682		
libtool-2.2.7a-1-msys-1.0.11-bin.tar.lzma	250.7 KB	2009-08-01	4,926		
libltdl-2.2.7a-1-msys-1.0.11-dll-7.tar.lzma	19.5 KB	2009-08-01	2,387		
libltdl-2.2.7a-1-msys-1.0.11-dev.tar.lzma	20.7 KB	2009-08-01	1,604		
▶ MSYS termcap	71.6 KB	2009-08-01	5,173		
▶ MSYS automake	1.8 MB	2009-08-01	9,551		

- ④ リンク先のページにある「MSYS perl」→「perl-5.6.1_2-1」→「perl-5.6.1_2-1-msys-1.0.11-bin.tar.lzma」をクリックし「C:\¥msys¥1.0」にダウンロードします。

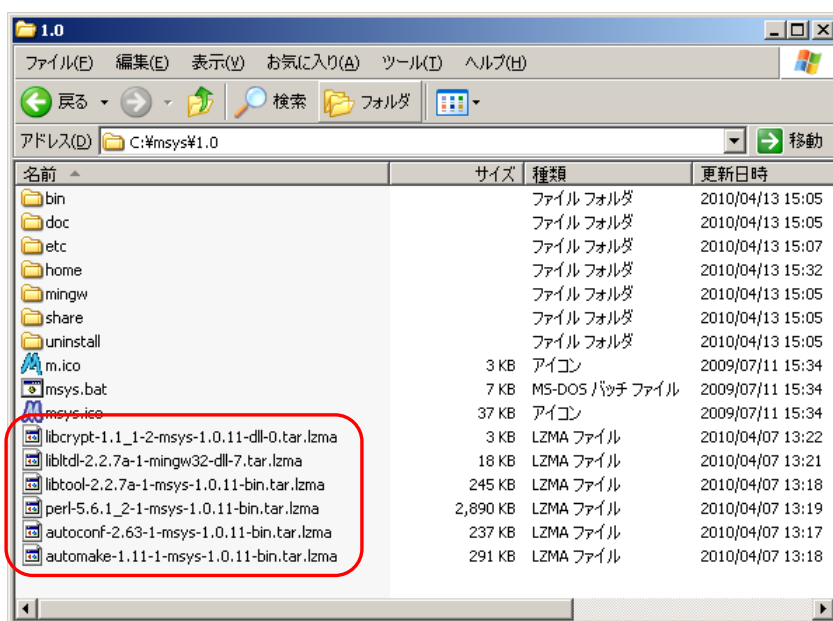
▶ MSYS automake	1.8 MB	2009-08-01	9,551		
▶ MSYS autoconf	2.0 MB	2009-08-01	10,819		
▼ MSYS perl	10.8 MB	2009-08-01	11,463		
▼ perl-5.6.1_2-1	10.8 MB	2009-08-01	11,463		
perl-5.6.1_2-1-msys.RELEASE_NOTES	4.4 KB	2009-08-01	1,254		
perl-5.6.1_2-1-msys-1.0.11-src.tar.lzma	4.7 MB	2009-08-01	755		
perl-5.6.1_2-1-msys-1.0.11-man.tar.lzma	1.3 MB	2009-08-01	1,068		
perl-5.6.1_2-1-msys-1.0.11-lic.tar.lzma	8.2 KB	2009-08-01	818		
perl-5.6.1_2-1-msys-1.0.11-html.tar.lzma	1.4 MB	2009-08-01	980		
perl-5.6.1_2-1-msys-1.0.11-doc.tar.lzma	536.5 KB	2009-08-01	1,003		
perl-5.6.1_2-1-msys-1.0.11-bin.tar.lzma	3.0 MB	2009-08-01	5,585		
▶ MSYS crypt	35.1 KB	2009-08-01	14,247		
▶ MinGW zlib	672.5 KB	2009-07-27	11,047		

- ⑤ リンク先のページにある「MinGW libtool」→「libtool-2.2.7a-1」→「libltdl-2.2.7a-1-mingw32-dll-7.tar.lzma」をクリックし「C:\¥msys¥1.0」にダウンロードします。

▶ MinGW libiconv	11.1 MB	2009-07-26	36,375		
▶ MinGW bzip2	3.0 MB	2009-07-25	10,381		
▼ MinGW libtool	2.3 MB	2009-07-25	10,256		
▼ libtool-2.2.7a-1	2.3 MB	2009-07-25	10,256		
libtool-2.2.7a-1-mingw32.RELEASE_NOTES	5.5 KB	2009-07-25	1,093		
libltdl-2.2.7a-1-mingw32-dll-7.tar.lzma	17.7 KB	2009-07-25	2,229		
libltdl-2.2.7a-1-mingw32-dev.tar.lzma	21.4 KB	2009-07-25	1,726		
libtool-2.2.7a-1-mingw32-src.tar.lzma	1.7 MB	2009-07-25	535		
libtool-2.2.7a-1-mingw32-lic.tar.lzma	6.6 KB	2009-07-25	748		
libtool-2.2.7a-1-mingw32-doc.tar.lzma	291.2 KB	2009-07-25	951		
libtool-2.2.7a-1-mingw32-bin.tar.lzma	248.8 KB	2009-07-25	2,974		
▶ MinGW automake	9.8 MB	2009-07-25	19,243		
▶ GCC Version 4	778.3 MB	2009-06-24	446,426		

- ⑥ リンク先のページにある「MSYS crypt」 → 「crypt-1.1_1-2」 → 「libcrypt-1.1_1-2-msys-1.0.11-dll-0.tar.lzma」をクリックし「C:\msys\1.0」にダウンロードします。

▶ MSYS autoconf	2.0 MB	2009-08-01	10,819		
▶ MSYS perl	10.8 MB	2009-08-01	11,463		
▼ MSYS crypt	35.1 KB	2009-08-01	14,247		
▼ crypt-1.1_1-2	35.1 KB	2009-08-01	14,247		
crypt-1.1_1-2-msys.RELEASE_NOTES	2.1 KB	2009-08-01	1,128		
crypt-1.1_1-2-msys-1.0.11-src.tar.lzma	14.0 KB	2009-08-01	554		
libcrypt-1.1_1-2-msys-1.0.11-dll-0.tar.lzma	2.4 KB	2009-08-01	5,548		
crypt-1.1_1-2-msys-1.0.11-doc.tar.lzma	1.7 KB	2009-08-01	709		
libcrypt-1.1_1-2-msys-1.0.11-dev.tar.lzma	5.9 KB	2009-08-01	1,587		
crypt-1.1_1-2-msys-1.0.11-lic.tar.lzma	6.6 KB	2009-08-01	797		
crypt-1.1_1-2-msys-1.0.11-bin.tar.lzma	2.4 KB	2009-08-01	3,924		
▶ MinGW zlib	672.5 KB	2009-07-27	11,047		
▶ MSYS gdbm	487.6 KB	2009-07-27	7,652		



- ⑦ MSYS を起動し、②から⑥でダウンロードしたファイルを解凍コマンドで解凍します。
解凍コマンドは「tar xvf ダウンロードしたファイル名 --lzma」を使用し、それぞれのファイルを解凍してください。
以下に解凍コマンドを示します。

```
user@PC /c/msys/1.0
$ tar xvf autoconf-2.63-1-msys-1.0.11-bin.tar.lzma --lzma
bin/
bin/autoconf
  —中略—
share/autoconf/m4sugar/m4sugar.m4f
share/autoconf/m4sugar/version.m4

user@PC /c/msys/1.0
$ tar xvf automake-1.11-1-msys-1.0.11-bin.tar.lzma --lzma
bin/
bin/aclocal
  —中略—
share/automake-1.11/ylwrap
share/aclocal/

user@PC /c/msys/1.0
$ tar xvf libcrypt-1.1_1-2-msys-1.0.11-dll-0.tar.lzma --lzma
bin/msys-crypt-0.dll

user@PC /c/msys/1.0
$ tar xvf libltdl-2.2.7a-1-mingw32-dll-7.tar.lzma --lzma
bin/libltdl-7.dll

user@PC /c/msys/1.0
$ tar xvf libtool-2.2.7a-1-msys-1.0.11-bin.tar.lzma --lzma
bin/
bin/libtool
  —中略—
share/aclocal/ltversion.m4
share/aclocal/lt-obsolete.m4

user@PC /c/msys/1.0
$ tar xvf perl-5.6.1_2-1-msys-1.0.11-bin.tar.lzma --lzma
bin/a2p.exe
bin/c2ph
  —中略—
lib/perl5/site_perl/5.6/msys/
perllocal.pod

user@PC /c/msys/1.0
```

- ⑧ MSYS で使用するコマンドのインストールは終了です。

2.2.5 OpenOCD のソースファイルのダウンロード

OpenOCD を再ビルドするために、OpenOCD のソースファイル入手します。

- ① OpenOCD のソースファイルをダウンロードしてください。OpenOCD ソースファイルのバージョンは OpenOCD 0.4.0 を使用して説明します。

Platform が Windows になっている「openocd-0.4.0.zip」をクリックし「openocd-0.4.0.zip」をダウンロードしてください。

OpenOCD ソースファイルのダウンロードページ

<http://sourceforge.net/projects/openocd/files/>

sourceforge FIND AND DEVELOP OPEN SOURCE SOFTWARE

Find Software | Develop | Create Project | Blog | Site Support | About

SourceForge.net > Find Software > OpenOCD - Open On-Chip Debugger (JTAG) > Browse Files

OpenOCD - Open On-Chip Debugger (JTAG)

by dbrownell, gowinex, zwelch

Summary | Files | Support | Develop

The "Open On-Chip Debugger" provides JTAG access from GDB (or directly with TCL scripts) to processors with ARM, MIPS, and other cores. SF.net hosts the master OpenOCD GIT tree, while openocd.berlios.de hosts the project's web site and mailing lists.

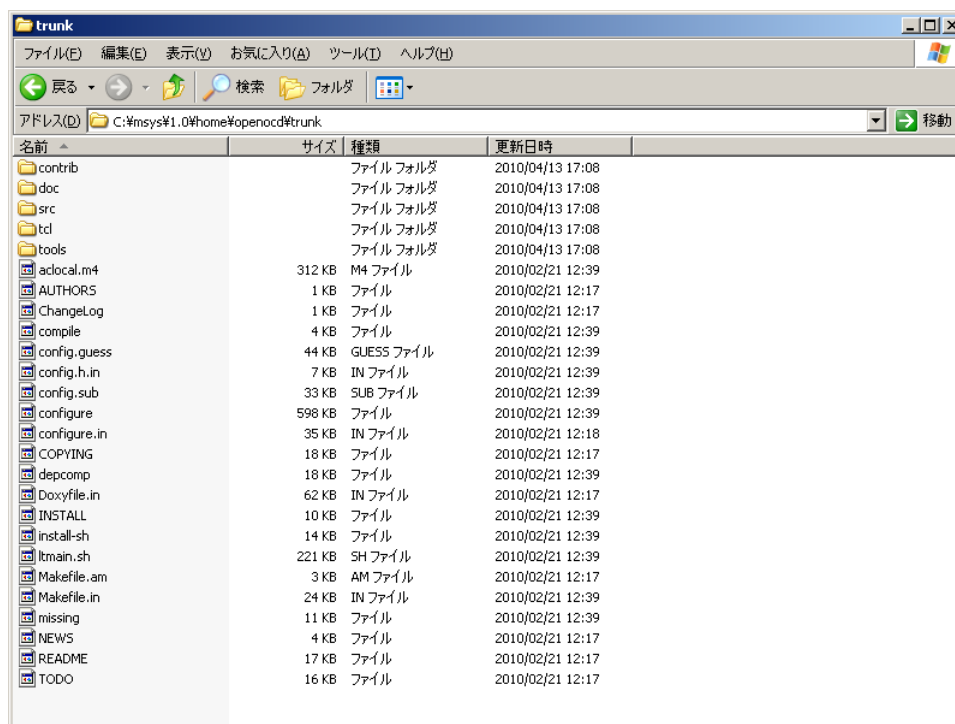
Download Now!
openocd-0.4.0.zip (2.1 MB) OR [View all files](#)

Browse Files for OpenOCD - Open On-Chip Debugger (JTAG)

File/Folder Name	Platform	Size	Date	Downloads	Notes/Subscribe
Newest Files					
openocd.pdf	OpenOCD Users Guide	757.5 KB	2010-02-21	255	
openocd-0.4.0.zip		2.1 MB	2010-02-21	1,029	
openocd-0.4.0.tar.gz	others	1.7 MB	2010-02-21	80	
openocd-0.4.0.tar.bz2		1.3 MB	2010-02-21	566	

All Files

- ② 「openocd-0.4.0.zip」を「C:\%msys%1.0\home\openocd\trunk」に解凍します。



- ③ OpenOCD のソースファイルのダウンロードは終了です。

2.1.6 USB ドライバ (FT2232) のダウンロード

OpenOCD に組み込む USB ドライバ (FT2232) をダウンロードします。

- ① USB ドライバを FTDI 社のホームページからダウンロードします。
Windows2000 以上に対応しているドライバ「2.06.00」をクリックし「CDM 2.06.00 WHQL Certified.zip」をダウンロードしてください。
なお、取扱説明書「3. USB ドライバ」を参照して USB ドライバをダウンロードして保存している場合②から行います。

OpenOCD ソースファイルのダウンロードページ

<http://www.ftdichip.com/Drivers/D2XX.htm>

Future Technology Devices International Ltd.
USB Device Solutions ASIC Design Product Design

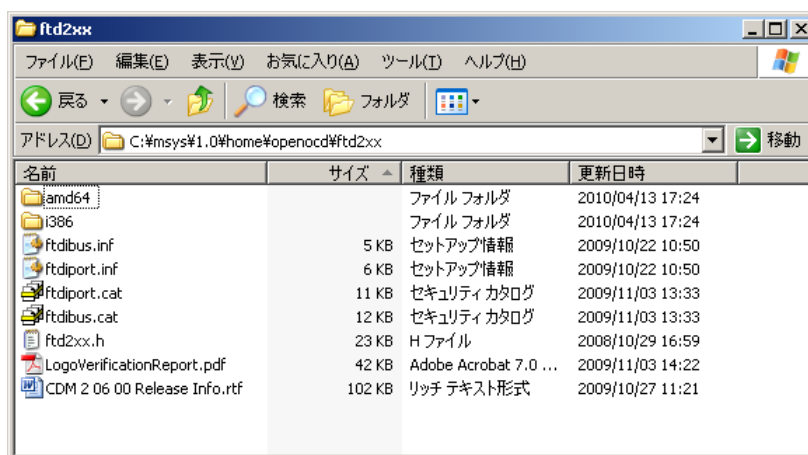
Home
Products
Drivers
VCP
D2XX
3rd Party Drivers
Documents
Resources
Projects
Support
Knowledgebase
Sales Network
Web Shop
Design Services
Corporate
Press
FTDI Newsletter
Contact

D2XX Direct Drivers
This page contains the D2XX drivers currently available for FTDI devices.
For Virtual COM Port (VCP) drivers, please click [here](#).
Installation guides are available from the [Installation Guides](#) page of the [Documents](#) section of this site for selected operating systems.

D2XX Drivers
D2XX drivers allow direct access to the USB device through a DLL. Application software can access the USB device through a series of DLL function calls. The functions available are listed in the [D2XX Programmer's Guide](#) document which is available from the [Documents](#) section of this site.
Programming examples using the D2XX drivers and DLL can be found in the [Projects](#) section of this site.
FTDI device drivers may be used only in conjunction with products based on FTDI parts.
The driver may be distributed in any form as long as our license information is not modified.
If a custom Vendor ID and/or Product ID, or description string are used, it is the responsibility of the product manufacturer to maintain any changes and subsequent WHQL re-certification as a result of using these changes.

Operating System	Devices Supported	Driver Version	Release Date	Comments
Windows Server 2008 R2				
Windows 7				
Windows 7 x64				
Windows Server 2008				Microsoft WHQL certified.
Windows Server 2008 x64	FT2232H, FT4232H,			Also available as a setup executable
Windows Vista	FT232R, FT245R, FT2232,	2.06.00	3rd November 2009	For custom VID and PID combinations see AN232R-03
Windows Vista x64	FT232B, FT245B,			Combined driver model (D2XX and VCP).
Windows XP	FT8U232AM, FT8U245AM			Devices programmed as VCP will expose a COM port, as will AM and BM devices.
Windows XP x64				Release Notes
Windows 2000				
Windows Server 2003				
Windows Server 2003				

- ② 「CDM 2.06.00 WHQL Certified.zip」を「C:\%msys%1.0\home\openocd\ftd2xx」に解凍します。



- ③ USB ドライバ (FT232) のダウンロードは終了です。

2.1.7 OpenOCD のビルド

OpenOCD のビルドを行い、USB ドライバ (FT2232) を組み込みます。

- ① MSYS を起動し、「C:\msys\1.0\home\openocd\trunk」に移動します。
フォルダ移動には「cd /c/msys/1.0/home/openocd/trunk/」コマンドを使用してください。

```
user@PC
$ cd /c/msys/1.0/home/openocd/trunk/
user@PC /c/msys/1.0/home/openocd/trunk
$
```

- ② configure スクリプトを実行します。
configure スクリプトの実行には「./configure --enable-maintainer-mode --disable-werror --disable-shared --enable-ft2232_ftd2xx --with-ftd2xx-win32-zipdir=/home/openocd/ftd2xx CC="gcc -mno-cygwin" CFLAGS="-O0 -g -Wall"」コマンドを使用してください。コマンドは一行で入れます。

```
user@PC /c/msys/1.0/home/openocd/trunk
$ ./configure --enable-maintainer-mode --disable-werror --disable-shared
--enable-ft2232_ftd2xx --with-ftd2xx-win32-zipdir=/home/openocd/ftd2xx
CC="gcc -mno-cygwin" CFLAGS="-O0 -g -Wall"
checking for a BSD-compatible install... /bin/install -c
checking whether build environment is sane... yes
—中略—
config.status: executing depfiles commands
config.status: executing libtool commands
user@PC /c/msys/1.0/home/openocd/trunk
$
```

- ③ Makefile が出来たことを確認後、make を行います
Makefile の確認には、「ls Makefile」コマンドを使用し、make の実行には「make」コマンドを使用してください。
make には時間がかかります。

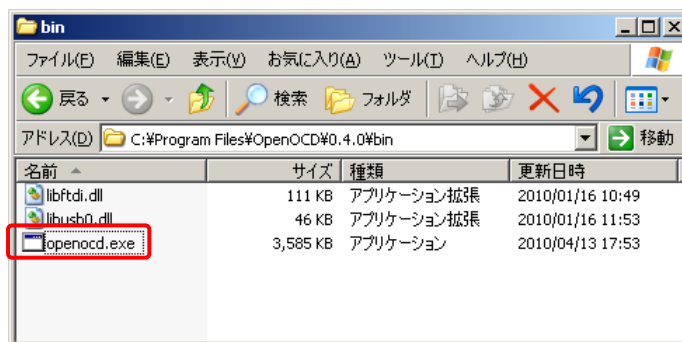
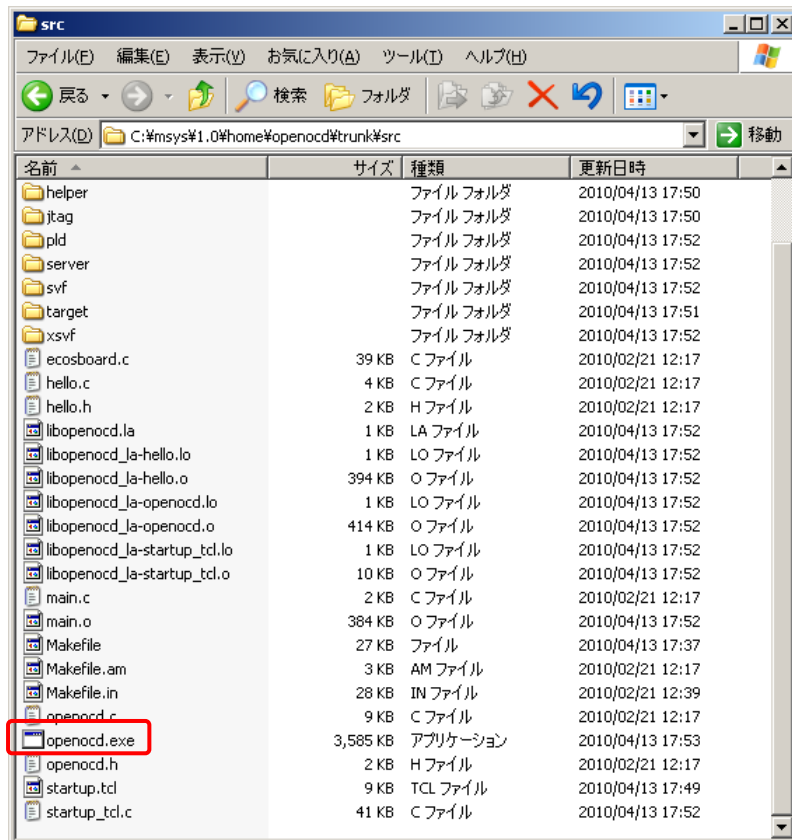
```
user@PC /c/msys/1.0/home/openocd/trunk
$ ls Makefile
Makefile

user@PC /c/msys/1.0/home/openocd/trunk
$ make
make all-recursive
make[1]: Entering directory `/c/msys/1.0/home/openocd/trunk'
—中略—
make[2]: Leaving directory `/c/msys/1.0/home/openocd/trunk'
make[1]: Leaving directory `/c/msys/1.0/home/openocd/trunk'

user@PC /c/msys/1.0/home/openocd/trunk
$
```

- ④ make 終了後、実行ファイルができていることを確認します。
「ls src/openocd.exe」コマンドを実行し src/openocd.exe が表示されれば実行ファイルが正常に作成されたことを確認できます。

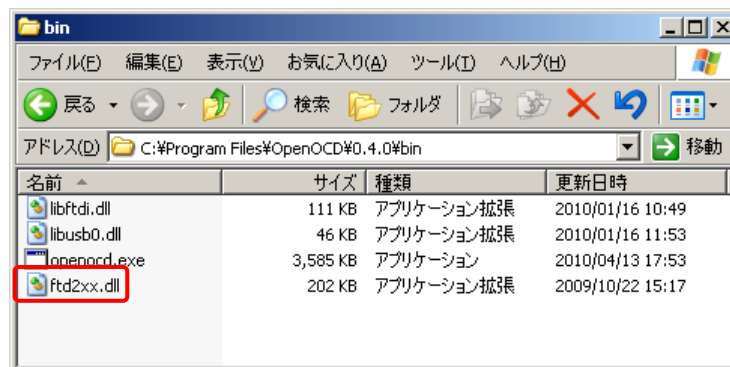
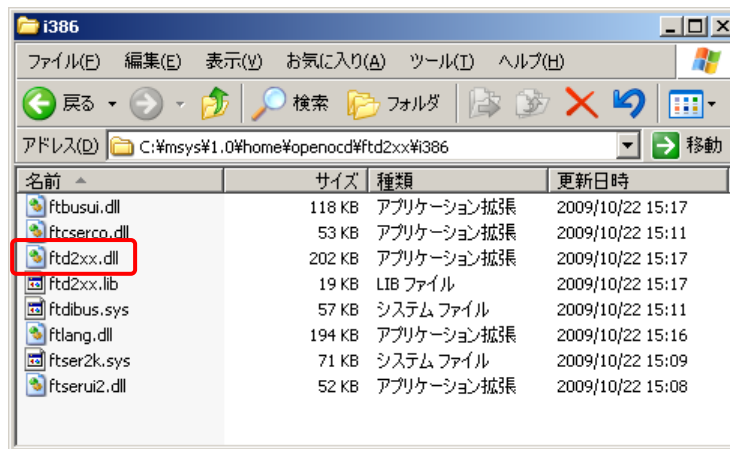
- ⑤ ビルドして作成された「openocd.exe」を 2.1 章で OpenOCD をインストールしたフォルダにコピーします。
- コピー元 : 「C:\msys\1.0\home\openocd\trunk\src\openocd.exe」
- コピー先 : 「C:\Program Files\OpenOCD\0.4.0\bin」



- ⑥ 2.5章でダウンロードしたUSBドライバ「ftd2xx.dll」を2.1章でOpenOCDをインストールしたフォルダにコピーします。

コピー元：「C:\msys\1.0\home\openocd\ftd2xx\i386」

コピー先：「C:\Program Files\OpenOCD\0.4.0\bin」



- ⑦ OpenOCD のビルドは終了です。

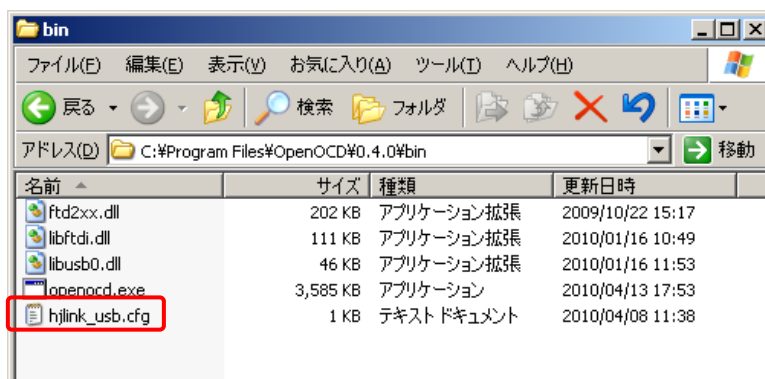
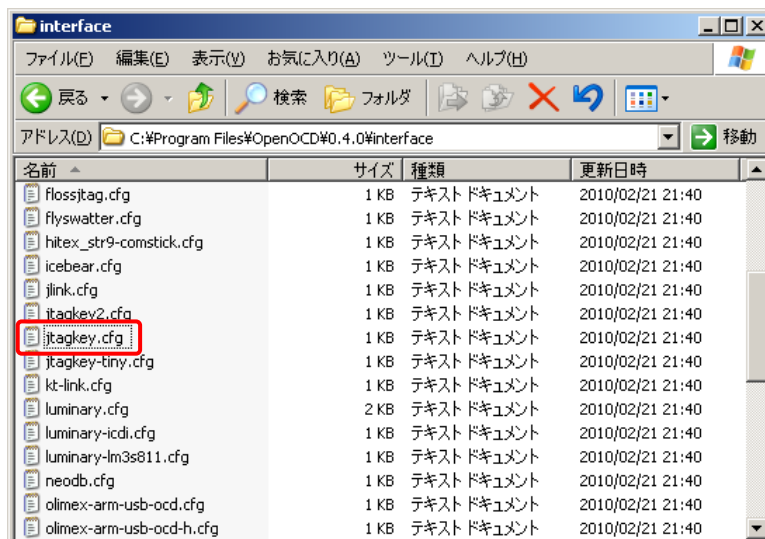
2.1.7 設定ファイルの作成

HJ-LINK/USB とターゲットボードを OpenOCD に対応させるための設定ファイルをそれぞれ作成します。

- ① HJ-LINK/USB を OpenOCD に対応させる設定ファイルをコピーします。

「C:\Program Files\OpenOCD\0.4.0\interface」にある設定ファイル「jtagkey.cfg」のファイル名を変更して「C:\Program Files\OpenOCD\0.4.0\bin」にコピーしてください。

以降、「jtagkey.cfg」を「hjlink_usb.cfg」にファイル名を変更して説明を行います。



- ② 「hlink_usb.cfg」をテキストエディタ等で開き記述を下記のように変更します。
変更後、保存しテキストエディタを閉じてください。

変更前

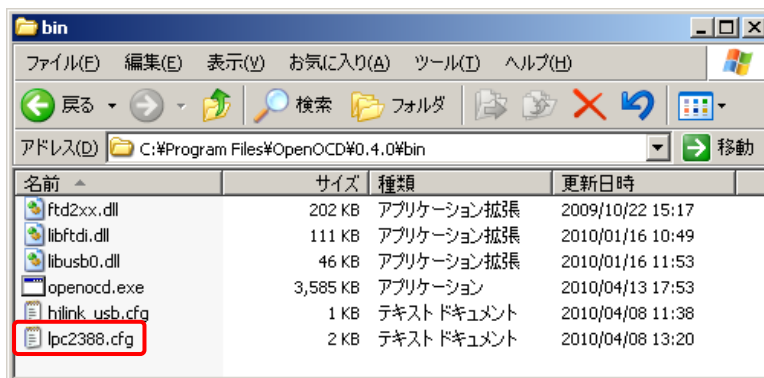
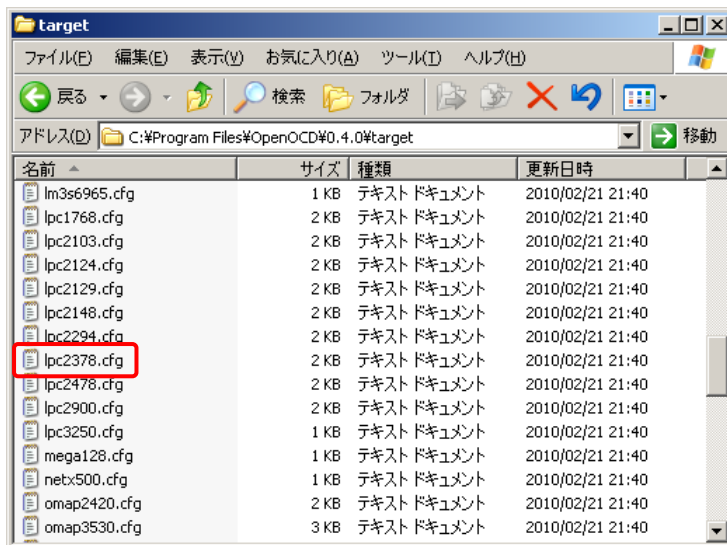
```
#
# Amontec JTAGkey
#
# http://www.amontec.com/jtagkey.shtml
#
interface ft2232
ft2232_device_desc "Amontec JTAGkey"
ft2232_layout jtagkey
ft2232_vid_pid 0x0403 0xcff8
```

変更後

```
#interface
interface ft2232
ft2232_device_desc "USB <-> Serial Cable A"
ft2232_layout jtagkey
ft2232_vid_pid 0x0403 0x6010
```

- ③ ターゲットボードを OpenOCD に対応させる設定ファイルをコピーします。

「C:\Program Files\OpenOCD\0.4.0\target」にある設定ファイル「lpc2378.cfg」のファイル名を変更して「C:\Program Files\OpenOCD\0.4.0\bin」にコピーしてください。以降、「lpc2378.cfg」を「lpc2388.cfg」にファイル名を変更して説明を行います。



- ④ 「lpc2388.cfg」をテキストエディタ等で開き記述を下記のように変更します。
変更後、保存しテキストエディタを閉じてください。

変更前

```
# NXP LPC2378 ARM7TDMI-S with 512kB Flash and 32kB Local On-Chip SRAM (58kB total),
# clocked with 4MHz internal RC oscillator

if { [info exists CHIPNAME] } {
    set _CHIPNAME $CHIPNAME
} else {
    set _CHIPNAME lpc2378
}

if { [info exists ENDIAN] } {
    set _ENDIAN $ENDIAN
} else {
    set _ENDIAN little
}

if { [info exists CPUTAPID ] } {
    set _CPUTAPID $CPUTAPID
} else {
    set _CPUTAPID 0x4f1f0f0f
}

#delays on reset lines
jtag_nsrst_delay 200
jtag_nrst_delay 200

# LPC2000 -> SRST causes TRST
reset_config trst_and_srst srst_pulls_trst

jtag newtap $_CHIPNAME cpu -irlen 4 -ircapture 0x1 -irmask 0xf -expected-id $_CPUTAPID

set _TARGETNAME $_CHIPNAME.cpu
target create $_TARGETNAME arm7tdmi -endian $_ENDIAN -chain-position $_TARGETNAME -variant arm7tdmi-s_r4

# LPC2378 has 32kB of SRAM on its main system bus (so-called Local On-Chip SRAM)
$_TARGETNAME configure -work-area-phys 0x40000000 -work-area-size 0x8000 -work-area-backup 0

$_TARGETNAME configure -event reset-init {
    # Force target into ARM state
    arm core_state arm
    #do not remap 0x0000-0x0020 to anything but the flash
    mwb 0xE01FC040 0x01
}

# LPC2378 has 512kB of FLASH, but upper 8kB are occupied by bootloader.
# After reset the chip uses its internal 4MHz RC oscillator
#flash bank lpc2000 <base> <size> 0 0 <target#> <variant>
set _FLASHNAME $_CHIPNAME.flash
flash bank $_FLASHNAME lpc2000 0x0 0x0007D000 0 0 $_TARGETNAME lpc2000_v2 4000 calc_checksum

# 4MHz / 6 = 666kHz, so use 500
jtag_khz 500
```

変更後

```
# NXP LPC2388 ARM7TDMI-S with 512kB Flash and 32kB Local On-Chip SRAM (58kB total),
#clocked with 4MHz internal RC oscillator

if { [info exists CHIPNAME] } {
    set _CHIPNAME $CHIPNAME
} else {
    set _CHIPNAME lpc2388
}

if { [info exists ENDIAN] } {
    set _ENDIAN $ENDIAN
} else {
    set _ENDIAN little
}

if { [info exists CPUTAPID ] } {
    set _CPUTAPID $CPUTAPID
} else {
    set _CPUTAPID 0x4f1f0f0f
}

#delays on reset lines
jtag_nsrst_delay 200
jtag_ntrst_delay 200

# LPC2000 -> SRST causes TRST
reset_config trst_and_srst srst_pulls_trst

jtag newtap $_CHIPNAME cpu -irlen 4 -ircapture 0x1 -irmask 0xf -expected-id $_CPUTAPID

set _TARGETNAME $_CHIPNAME.cpu
target create $_TARGETNAME arm7tdmi -endian $_ENDIAN -chain-position $_TARGETNAME -variant arm7tdmi-s_r4

# LPC2388 has 64kB of SRAM on its main system bus (so-called Local On-Chip SRAM)
$_TARGETNAME configure -work-area-virt 0 -work-area-phys 0x4000000 -work-area-size 0x8000
-work-area-backup 0

$_TARGETNAME configure -event reset-init {
    # Force target into ARM state
    soft_reset_halt
#
    arm core_state arm
    #do not remap 0x0000-0x0020 to anything but the flash
    mwb 0xE01FC040 0x01
}

# LPC2378 has 512kB of FLASH, but upper 8kB are occupied by bootloader.
# After reset the chip uses its internal 4MHz RC oscillator
#flash bank lpc2000 <base> <size> 0 0 <target#> <variant>
set _FLASHNAME $_CHIPNAME.flash
flash bank $_FLASHNAME lpc2000 0x0 0x0007D000 0 0 $_TARGETNAME lpc2000_v2 4000 calc_checksum

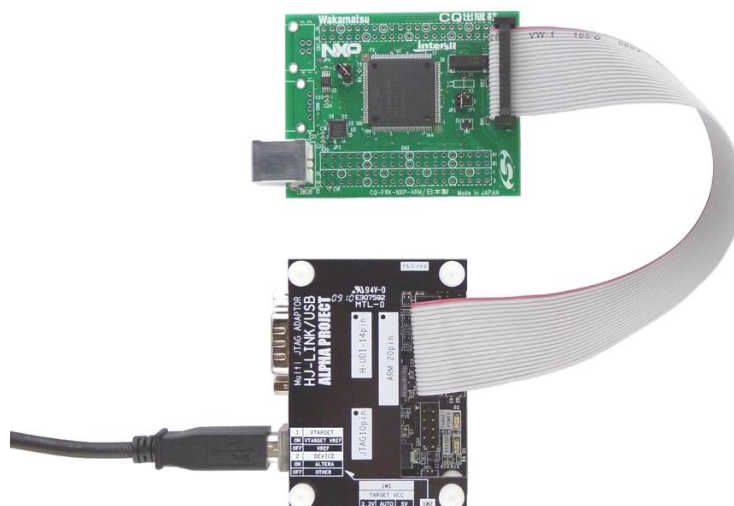
# 4MHz / 6 = 666kHz, so use 500
jtag_khz 500
```

- ⑤ 設定ファイルの作成は終了です。

2.3 OpenOCD の動作確認

OpenOCD の動作確認を行います。

- ⑥ HJ-LINK/USB とターゲットボードを付属の JTAG ケーブルで接続します。
HJ-LINK/USB と PC を付属の USB ケーブルで接続してください。



- ⑦ Windows のコマンド プロンプトを起動し、「openocd.exe」や設定ファイルを保存した場所
「C:\Program Files\OpenOCD\0.4.0\bin」へ移動します。

```
C:>cd c:\Program Files\OpenOCD\0.4.0\bin
C:\Program Files\OpenOCD\0.4.0\bin>
```

- ⑧ ターゲットボードの電源を入れます。

- ⑨ ターゲットボードと接続します。以下のコマンドを入力します。

```
C:¥Program Files¥OpenOCD¥0.4.0¥bin>openocd -f hlink_usb.cfg -f lpc2388.cfg
```

ターゲットボードと正しく通信ができていると以下のように表示されます。

```
Open On-Chip Debugger 0.4.0 (2010-04-13-17:52)
Licensed under GNU GPL v2
For bug reports, read
  http://openocd.berlios.de/doc/doxygen/bugs.html
jtag_nsrst_delay: 200
jtag_ntrst_delay: 200
trst_and_srst srst_pulls_trst srst_gates_jtag trst_push_pull srst_open_drain
500 kHz
Info : device: 6 "2232H"
Info : deviceID: 67330064
Info : SerialNumber: FTT2ZKH9A
Info : Description: USB <-> Serial Cable A
Info : max TCK change to: 30000 kHz
Info : clock speed 500 kHz
Info : JTAG tap: lpc2388.cpu tap/device found: 0x4f1f0f0f (mfg: 0x787, part: 0xf
1f0, ver: 0x4)
Info : Embedded ICE version 7
Error: EmbeddedICE v7 handling might be broken
Info : lpc2388.cpu: hardware has 2 breakpoint/watchpoint units
```

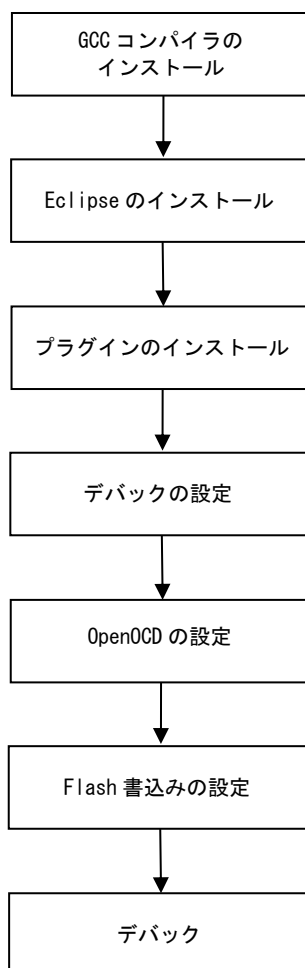
- ⑩ OpenOCD は「Ctrl+C」キーを押すことで終了します。
「Ctrl+C」キーを押して OpenOCD を終了してください。
- ⑪ OpenOCD の動作確認は終了です。

2.4 Eclipse の設定

Eclipse を使用しての開発環境構築方法については、インターネット上のさまざまなサイトで詳しく説明されております。

「1.1 概要」で紹介している参考文献やインターネット上のサイトを参考にするなどして開発環境を構築してみてください。

以下に、簡単な開発環境構築方法とデバックの手順記します。



2.4.1 GCC コンパイラのインストール

GCC コンパイラのインストールは、さまざまなサイトで詳しく説明されております。

「1.1 概要」で紹介している参考文献やインターネット上のサイトを参考にするなどして、GCC コンパイラのインストールを行ってください。

2.4.2 Eclipse のインストール

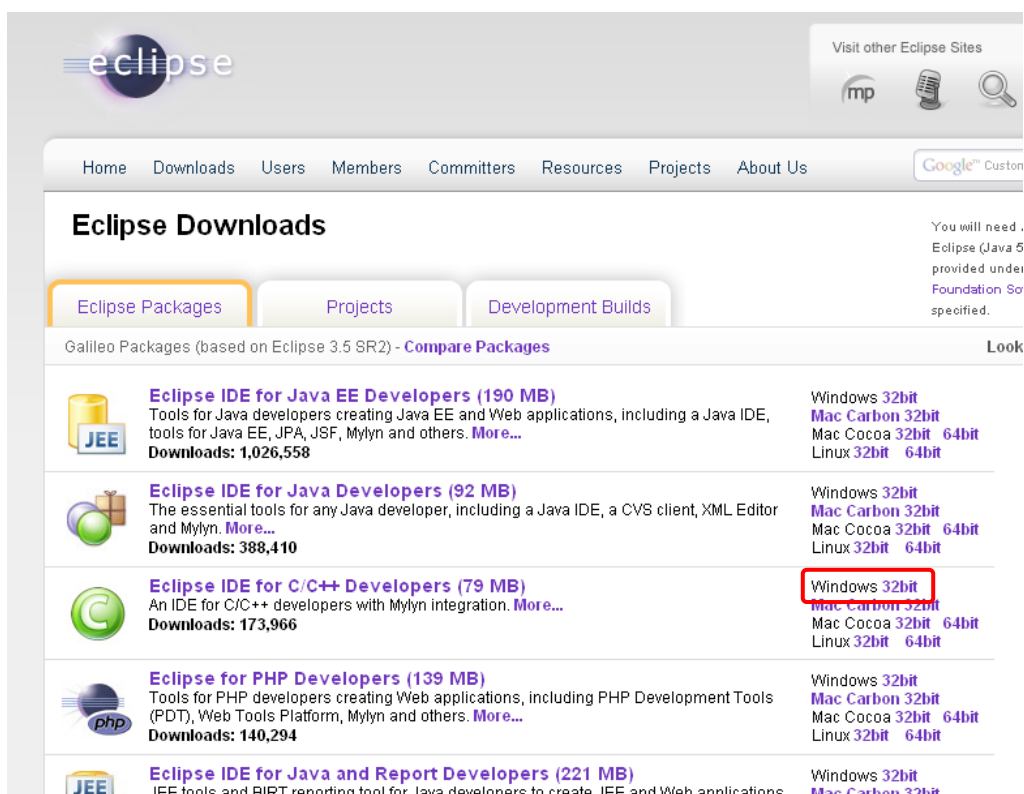
ターゲットボードをデバックするために Eclipse をダウンロードし、インストールします。
なお、Eclipse は、WindowsOS の 32bit のみに対応しています。

- ① Eclipse をダウンロードします。

Eclipse のホームページから「Eclipse IDE for C/C++ Developers(79MB)」の Windows 32bit をクリックしてダウンロードしてください。

Eclipse のダウンロードページ

<http://www.eclipse.org/downloads/index.php>



The screenshot shows the Eclipse Downloads page. The main heading is "Eclipse Downloads". Below it, there are tabs for "Eclipse Packages", "Projects", and "Development Builds". The "Eclipse Packages" tab is selected. The page lists several Eclipse IDE packages. The package "Eclipse IDE for C/C++ Developers (79 MB)" is highlighted with a red box around the "Windows 32bit" link. The package details are as follows:

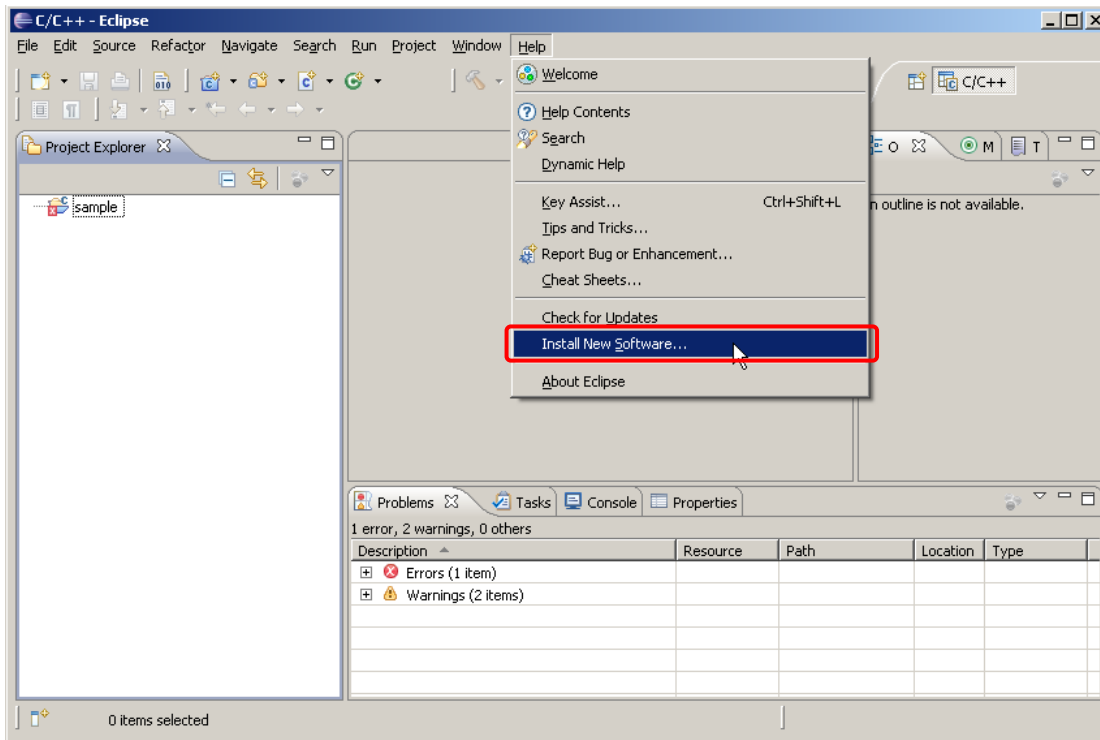
Package Name	Size	Downloads	Operating Systems
Eclipse IDE for Java EE Developers	190 MB	1,026,558	Windows 32bit, Mac Carbon 32bit, Mac Cocoa 32bit 64bit, Linux 32bit 64bit
Eclipse IDE for Java Developers	92 MB	388,410	Windows 32bit, Mac Carbon 32bit, Mac Cocoa 32bit 64bit, Linux 32bit 64bit
Eclipse IDE for C/C++ Developers	79 MB	173,966	Windows 32bit, Mac Carbon 32bit, Mac Cocoa 32bit 64bit, Linux 32bit 64bit
Eclipse for PHP Developers	139 MB	140,294	Windows 32bit, Mac Carbon 32bit, Mac Cocoa 32bit 64bit, Linux 32bit 64bit
Eclipse IDE for Java and Report Developers	221 MB	-	Windows 32bit, Mac Carbon 32bit

- ② ダウンロードした eclipse-cpp-galileo-SR2-win32.zip を解凍してください。以降、解凍先フォルダを C:\eclipse として説明します。
- ③ Eclipse のインストールは終了です。

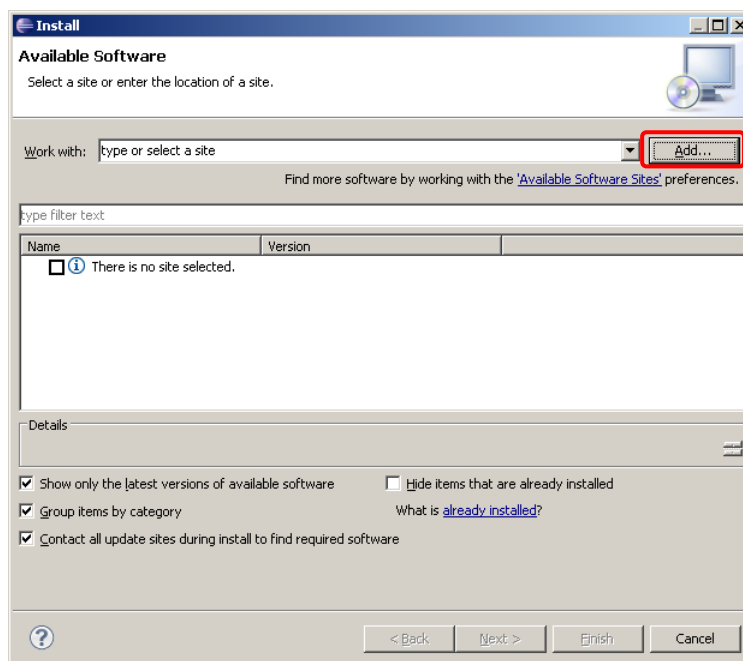
2.4.3 Eclipse のプラグインのインストール

Eclipse に C 言語対応機能を追加するためにプラグインをインストールします。

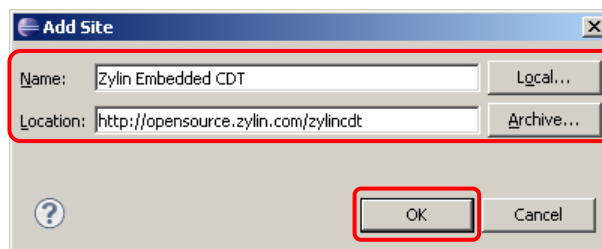
- ① Eclipse メニューバーにある「Help」→「Install New Software...」を選択します。



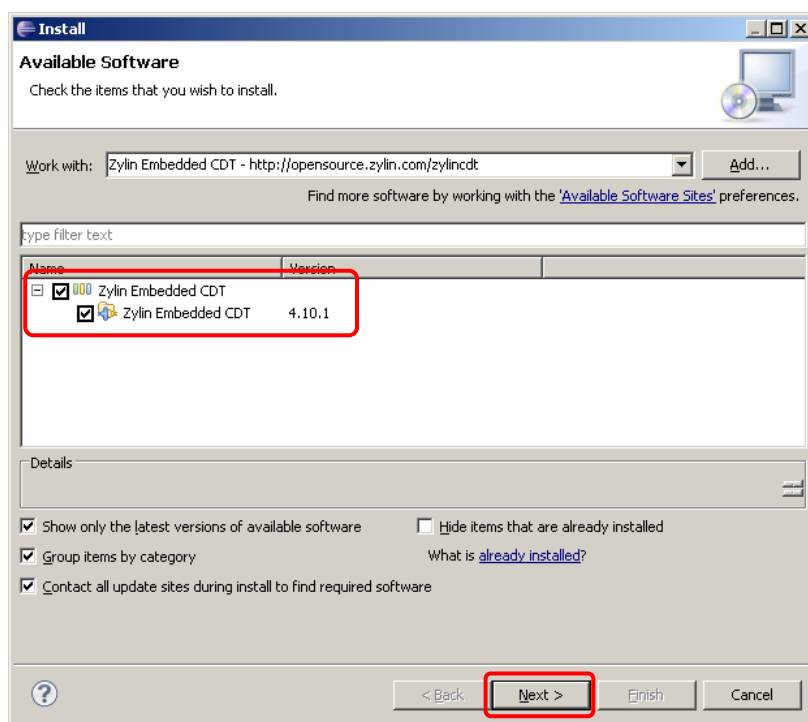
- ② Available Software 画面が表示されるので、「Add...」を選択します。



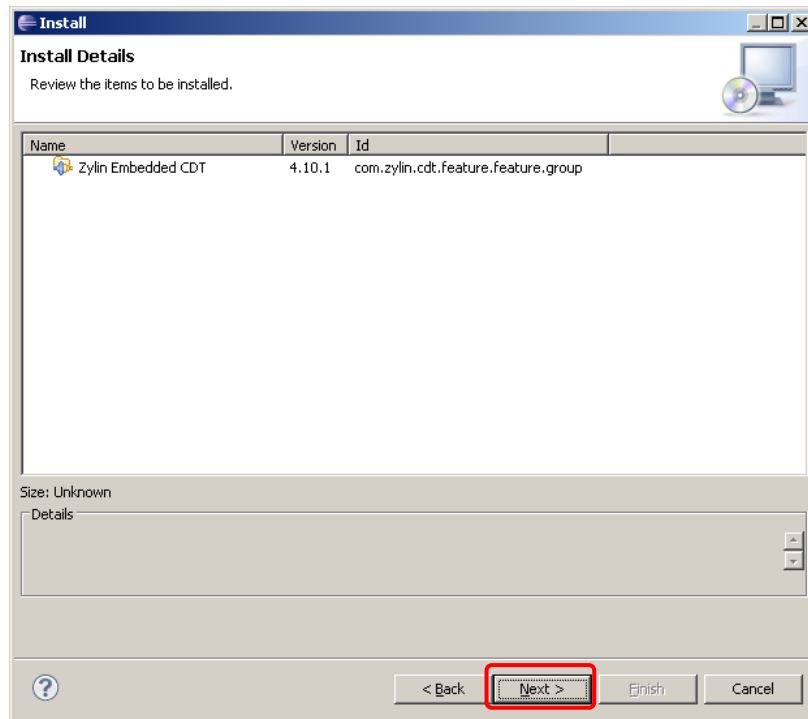
- ③ Add Site 画面が表示されます。「Name」に「Zylin Embedded CDT」を入力し、「Location」に「http://opensource.zylin.com/zylincdt」と入力後「OK」を選択してください。



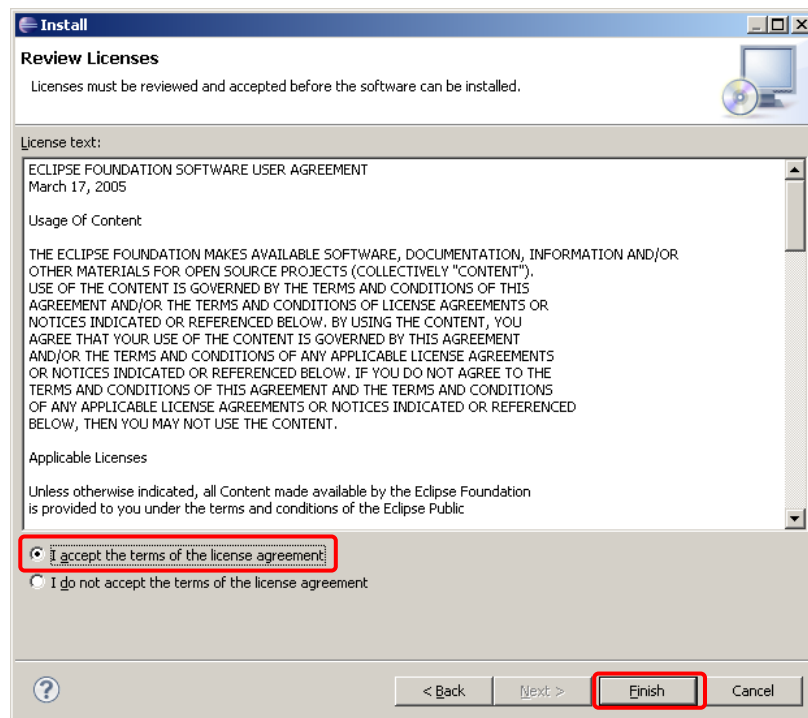
- ④ Available Software 画面に「Zylin Embedded CDT」が追加されます。チェックボックスにチェックを入れ「Next」を選択してください。



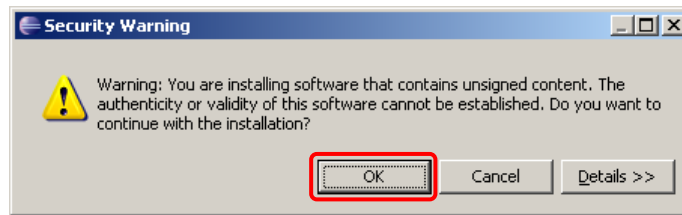
- ⑤ インストールの確認画面が表示されるので、「Next」を選択します。



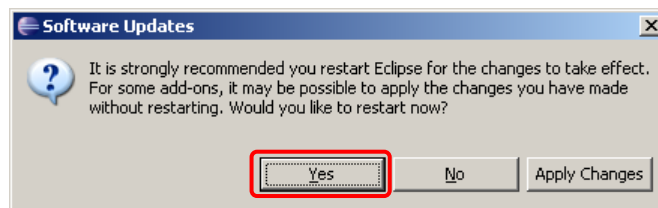
- ⑥ 使用条件の同意画面が表示されるので、最後まで読み同意できたら「I accept the terms in the license agreement」にチェックを入れ、「Finish」を選択します。
インストールが開始されます。



インストール中に以下の警告文が表示されますが、「OK」を選択します。



- ⑦ インストールが終了すると Eclipse の再起動を促すウィンドウが表示されます。Eclipse を再起動させるために「Yes」を選択してください。



- ⑧ Eclipse のプラグインのインストールは終了です。

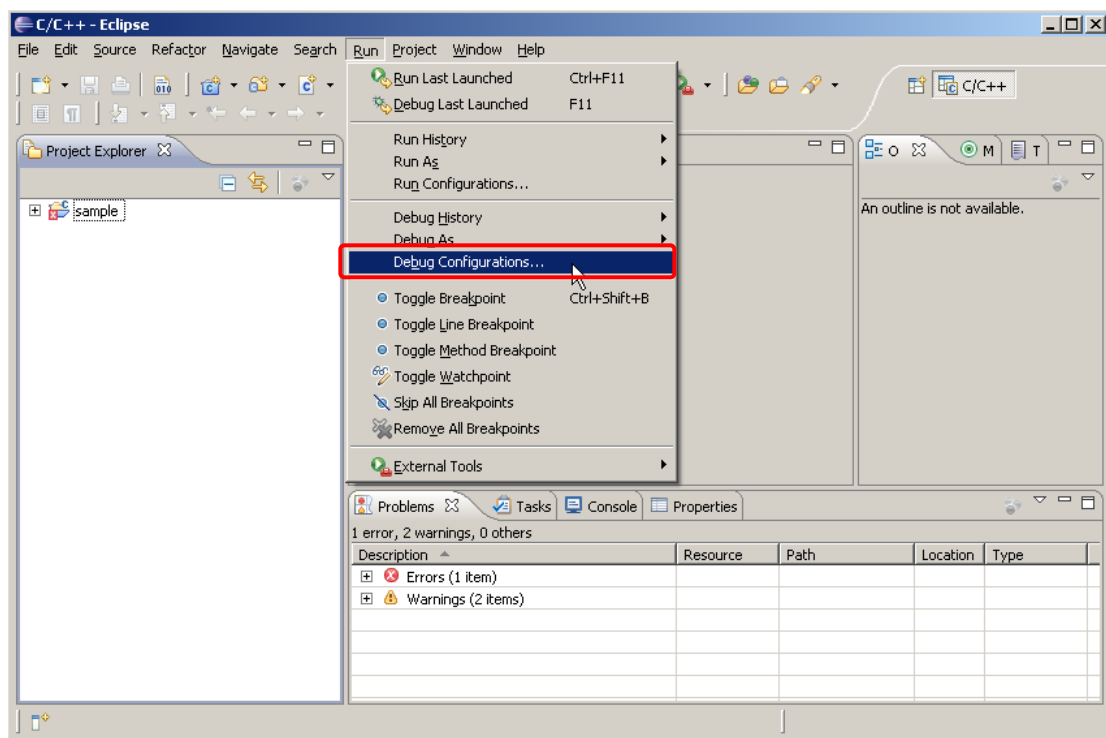
2.4.4 Eclipse (デバック) の設定

Eclipse からターゲットボードのデバックを行うための設定を行います。

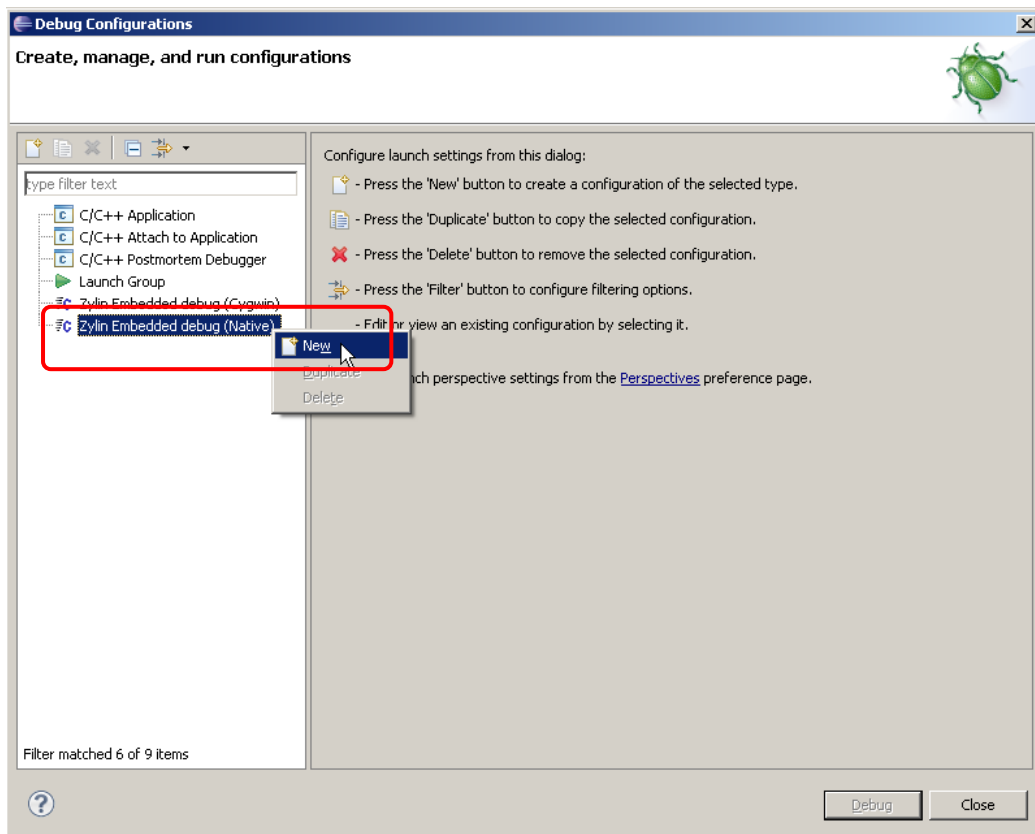
プログラム作成後に以下の作業を行ってください。

ターゲットボードへダウンロードするプログラムはお客様自身でご用意してください。

- ① Eclipse メニューバーにある「Run」→「Debug Configurations...」を選択します。

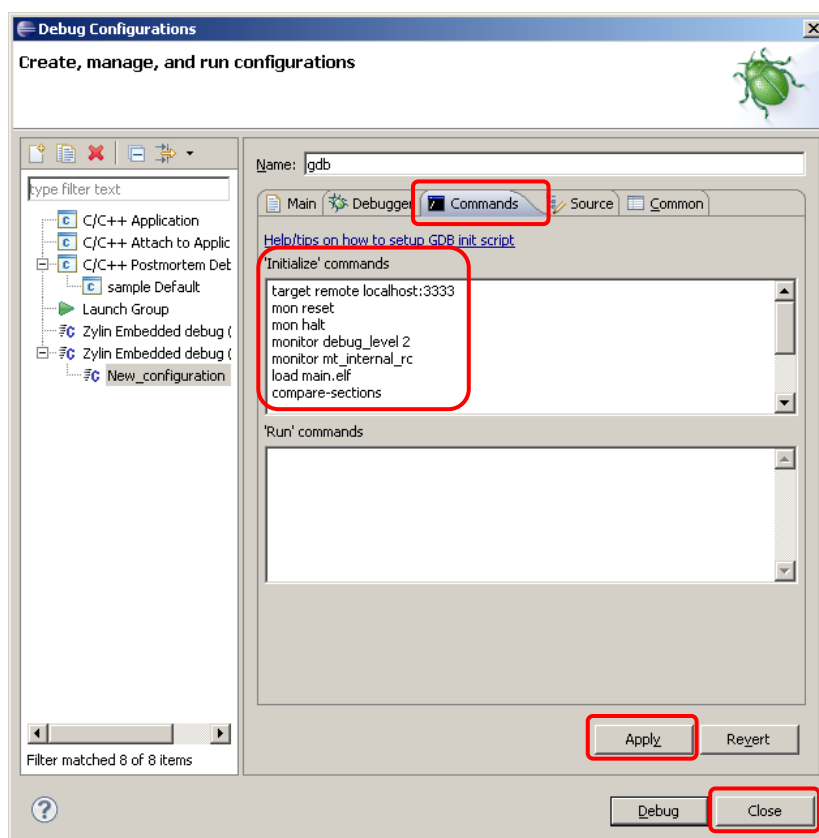


- ② Debug Configurations 画面で「Zylin Embedded debug(Native)」を右クリックし「New」を選択します。



- ③ Commands タブの「Initialize' commands」に以下のコマンドを入力します。
Name や Main、Debugger 等のその他タブはお客様の仕様にあった設定を行ってください。
入力が終わりましたら、「Apply」を選択し、「Close」を選択してください。
なお、Name は「gdb」として説明します。

```
target remote localhost:3333
mon reset
mon halt
monitor debug_level 2
monitor mt_internal_rc
load main.elf
compare-sections
monitor soft_reset_halt
set mem inaccessible-by-default off
monitor debug_level 0
thbreak main
```

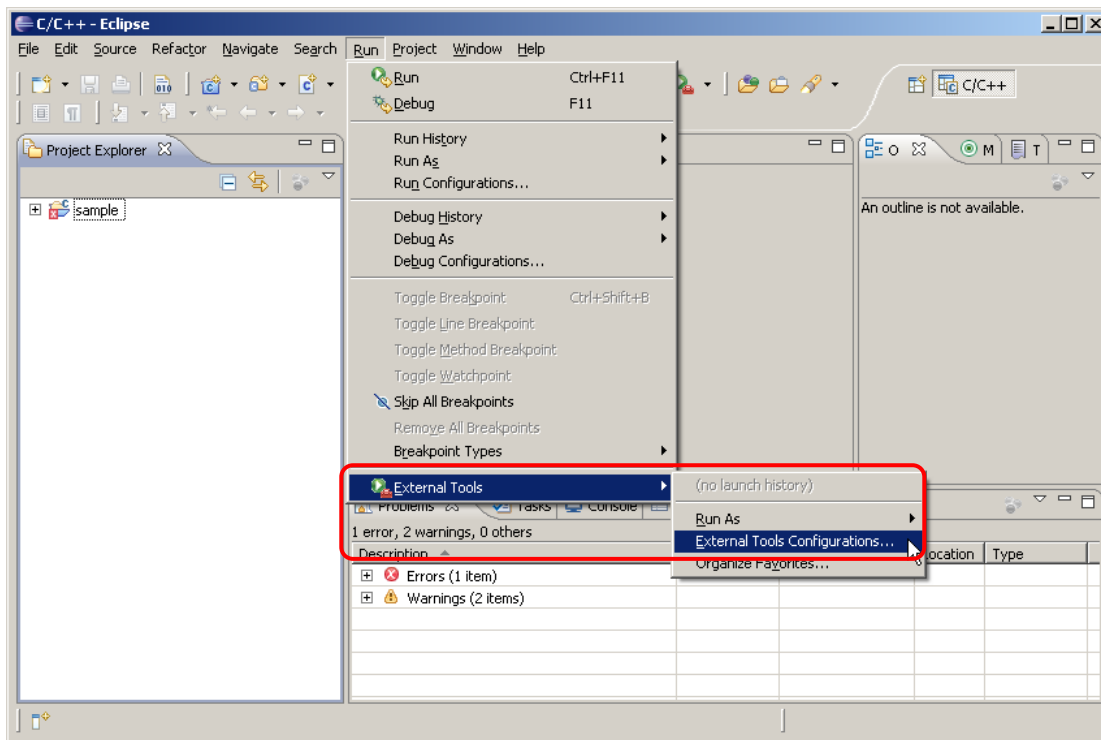


- ④ Eclipse (デバック) の設定は終了です。

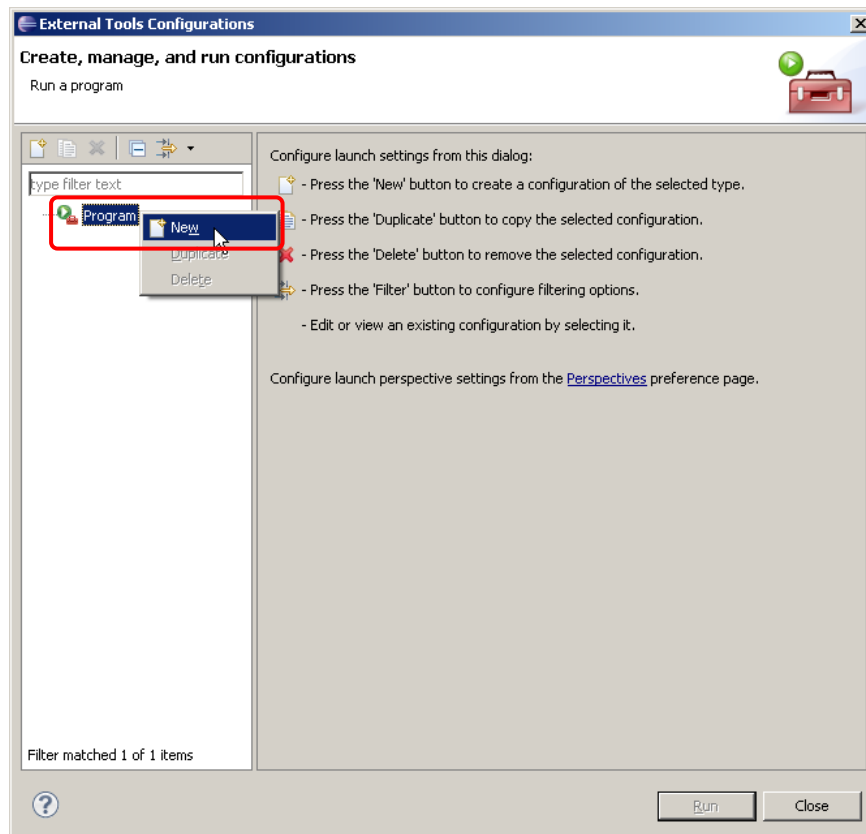
2.4.5 Eclipse (OpenOCD) の設定

Eclipse から OpenOCD へ接続するための設定を行います。

- ① Eclipse メニューバーにある「Run」→「External Tools」→「External Tools Configurations...」を選択します。



- ② External Tools Configurations 画面で「Program」を右クリックし、「New」を選択します。

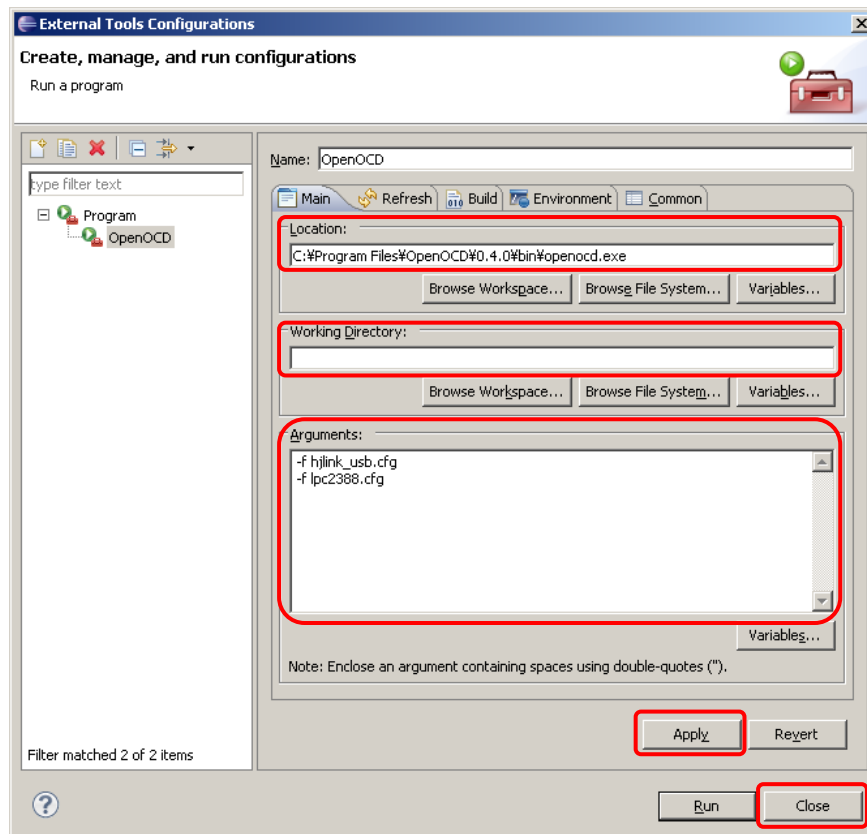


- ③ Main タブの「Location」に OpenOCD.exe をインストールした場所「C:\Program Files\OpenOCD\0.4.0\bin\openocd.exe」を指定します。「Arguments」には、設定ファイル「-f hlink_usb.cfg」と「-f lpc2388.cfg」を入力してください。Main タブにある「Working Directory」項目や、「Name」や「Refresh」、「Build」等の他のタブはお客様の仕様にあった設定を行ってください。

「Working Directory」で指定したフォルダには、「2.1.7 設定ファイル」で作成した設定ファイル「hlink_usb.cfg」と「lpc2388.cfg」をコピーしてください。

入力が終わりましたら、「Apply」を選択し、「Close」を選択します。

なお、Name は「OpenOCD」として説明します。

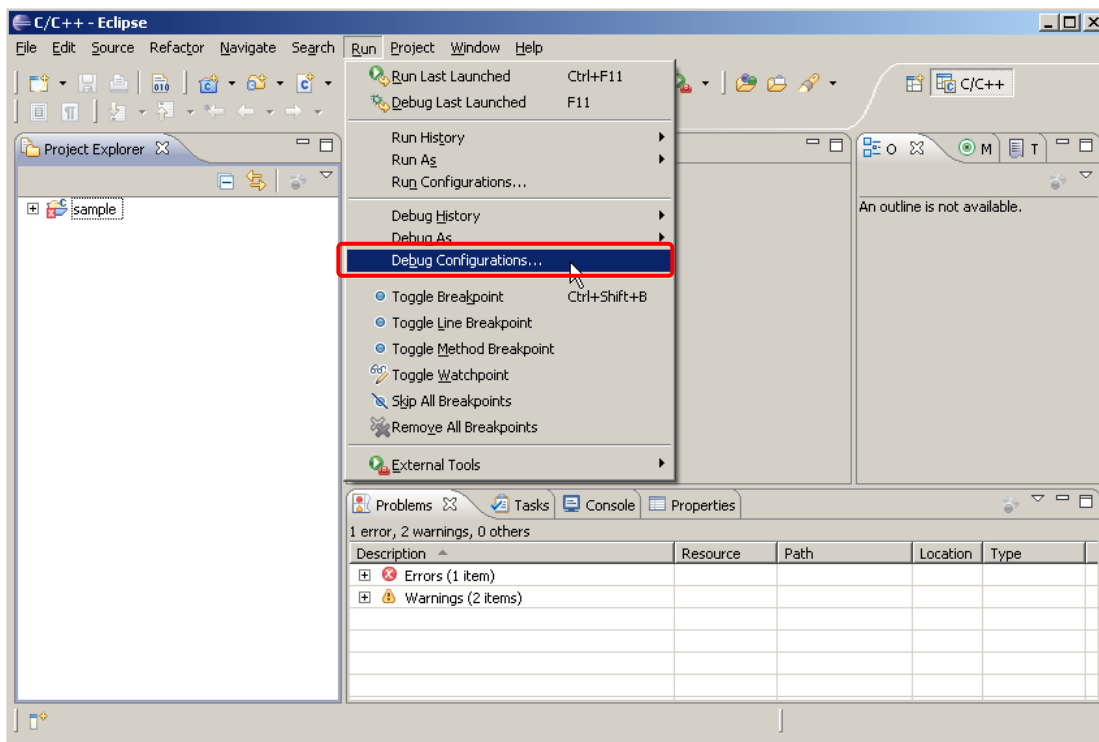


- ④ Eclipse (OpenOCD) の設定は終了です。

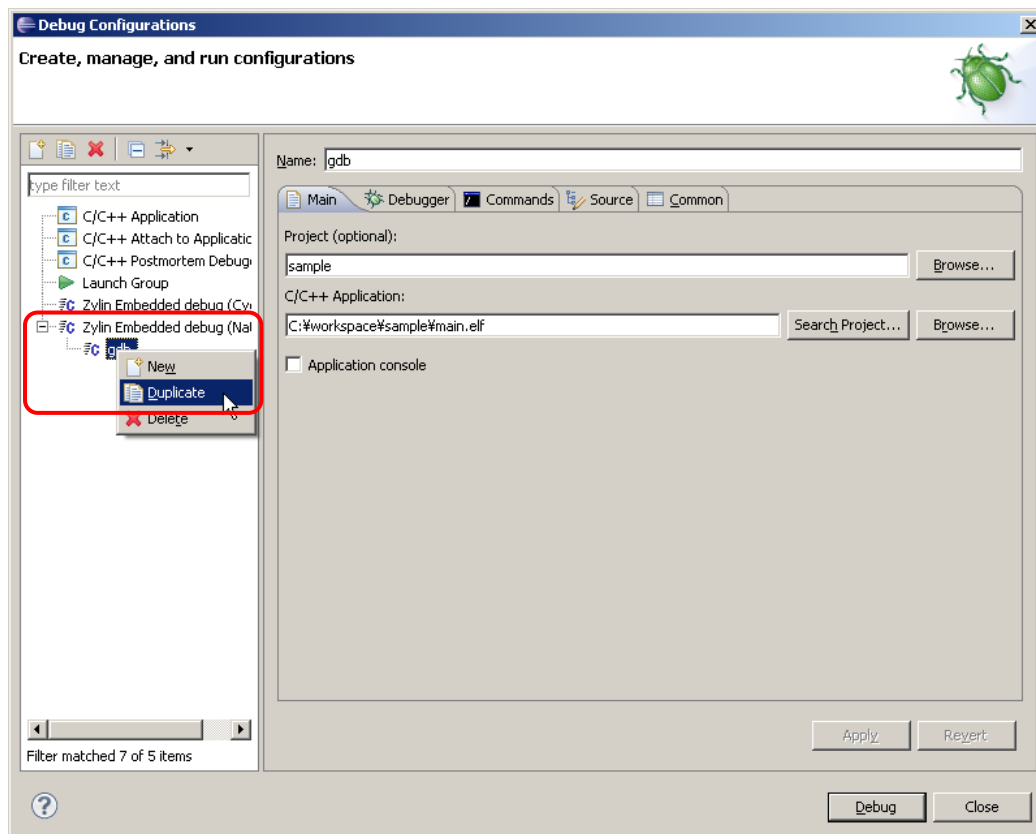
2.4.6 Flash 書き込みの設定

Eclipse からターゲットボードのフラッシュにプログラムを書き込むための設定を行います。

- ① Eclipse メニューバーにある「Run」→「Debug Configurations...」を選択します。

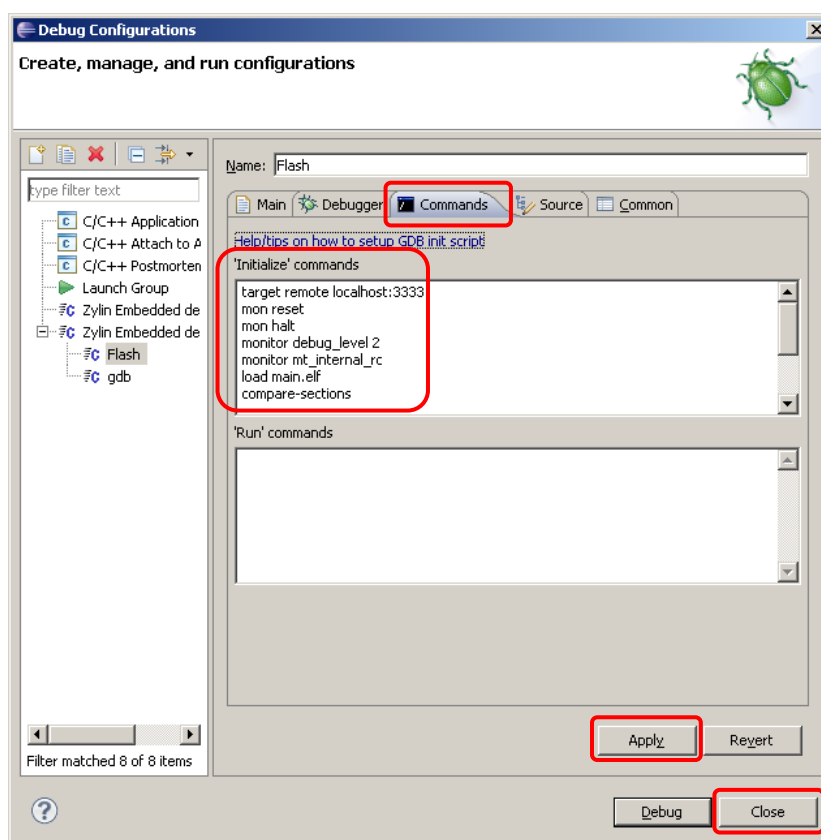


- ② Debug Configurations 画面で「Zylin Embedded debug (Native)」→「gdb」を右クリックし「Duplicate」を選択します。



- ③ Commands タブの「Initialize' commands」に以下のコマンドを入力します。
Name や Main、Debugger 等のその他タブはお客様の仕様にあった設定を行ってください。
入力が終わりましたら、「Apply」を選択し、「Close」を選択してください。
なお、Name は「Flash」として説明します。

```
target remote localhost:3333
mon reset
mon halt
monitor debug_level 2
monitor mt_internal_rc
load main.elf
compare-sections
monitor soft_reset_halt
set mem inaccessible-by-default off
monitor debug_level 0
mon mt_flash_elf main.elf 0
quit
```



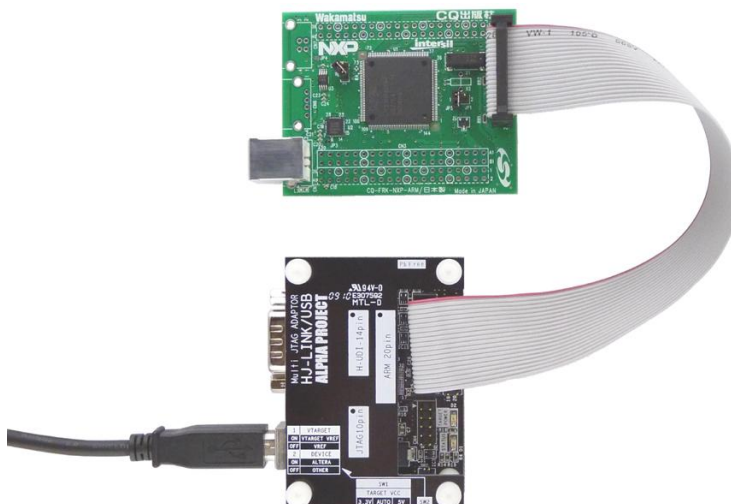
- ④ Flash 書き込みの設定は終了です。

2.5 デバック

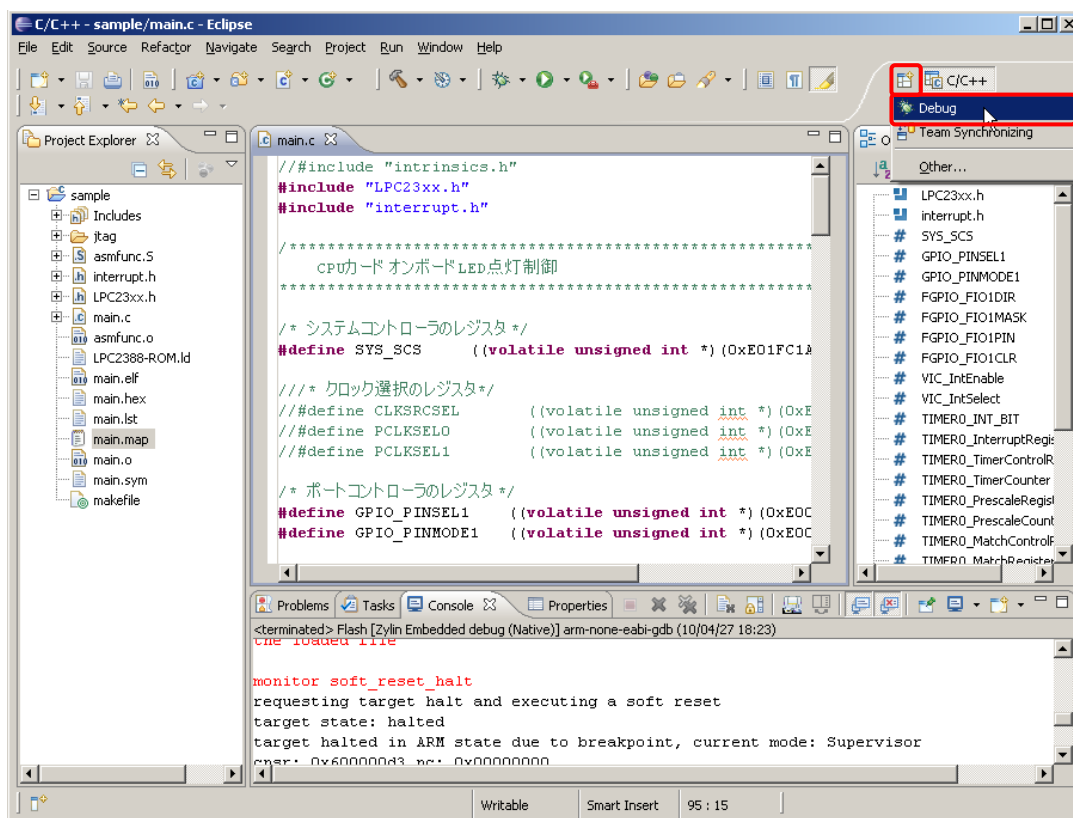
2.5.1 デバック (Flash 書込み) 作業

Eclipse からターゲットボードへFlash 書込みを行います。

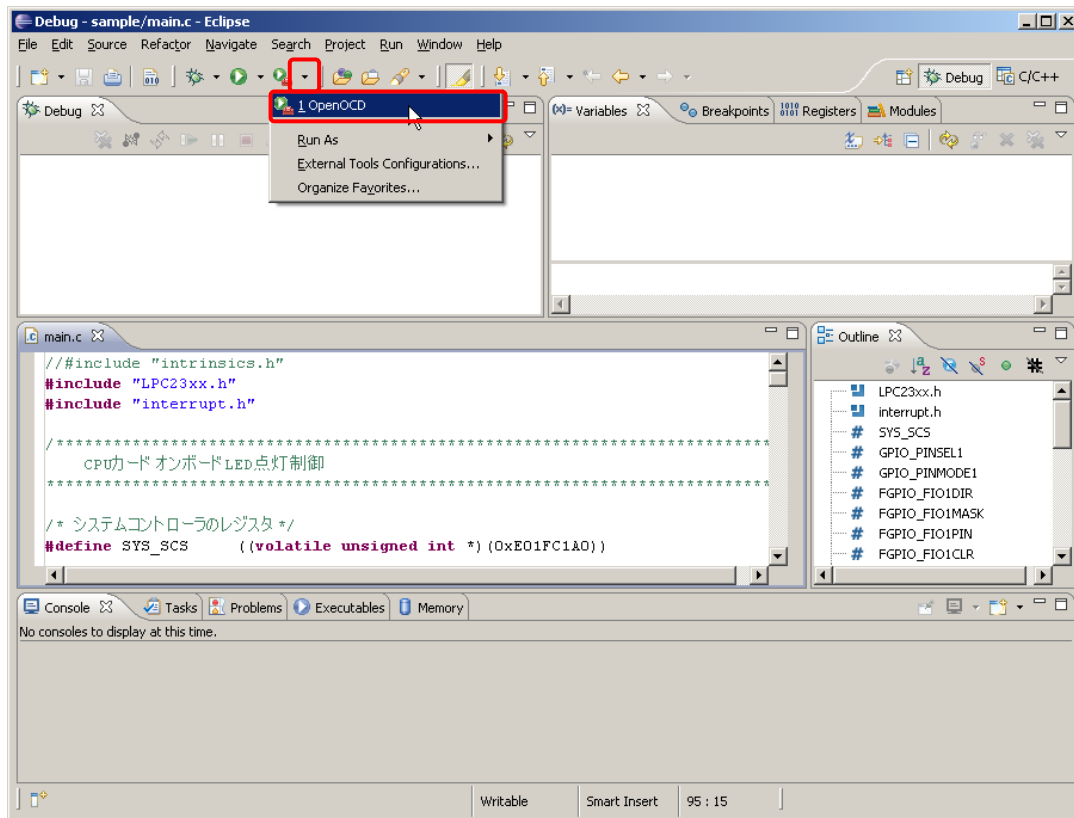
- ① HJ-LINK/USB とターゲットボードを付属の JTAG ケーブルで接続します。
HJ-LINK/USB と PC を付属の USB ケーブルで接続してください。



- ② ターゲットボードに電源を入れてください。
- ③ Eclipse の Open Perspective をクリックし Debug を選択します。

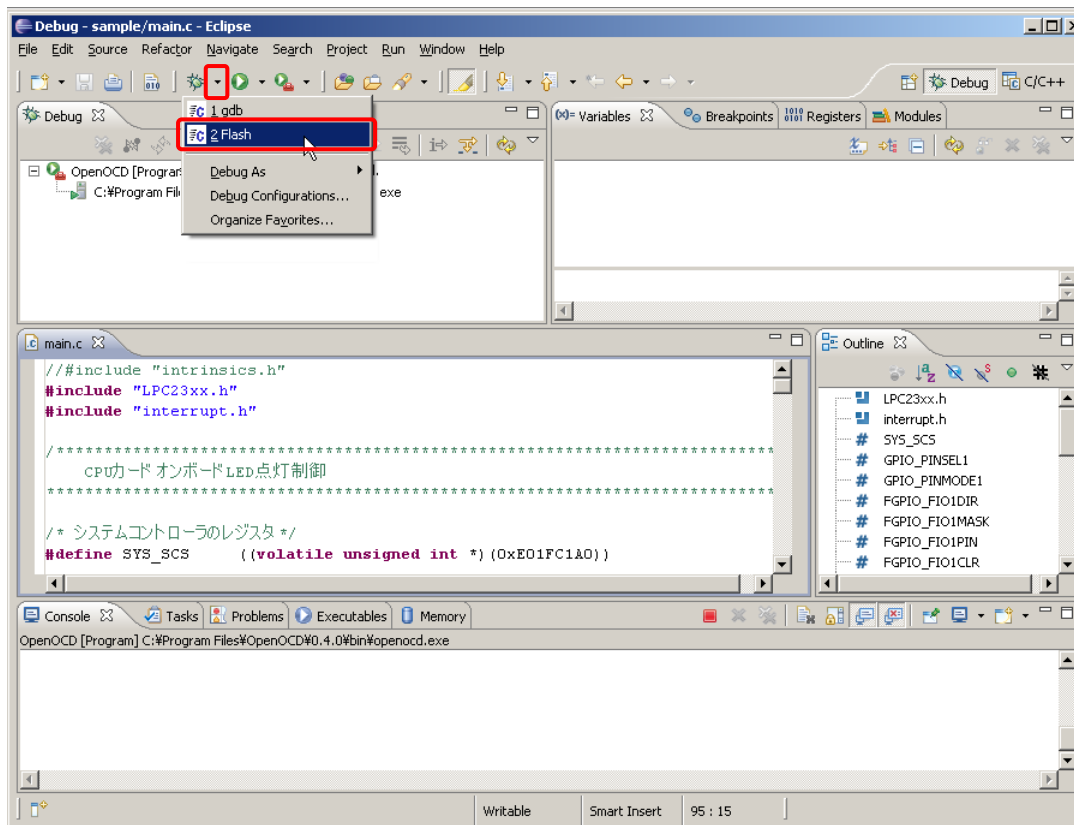


- ④ External Tools の▼ボタンをクリックし、OpenOCD を選択します。



- ⑤ Debug の▼ボタンをクリックし、Flash を選択してください。

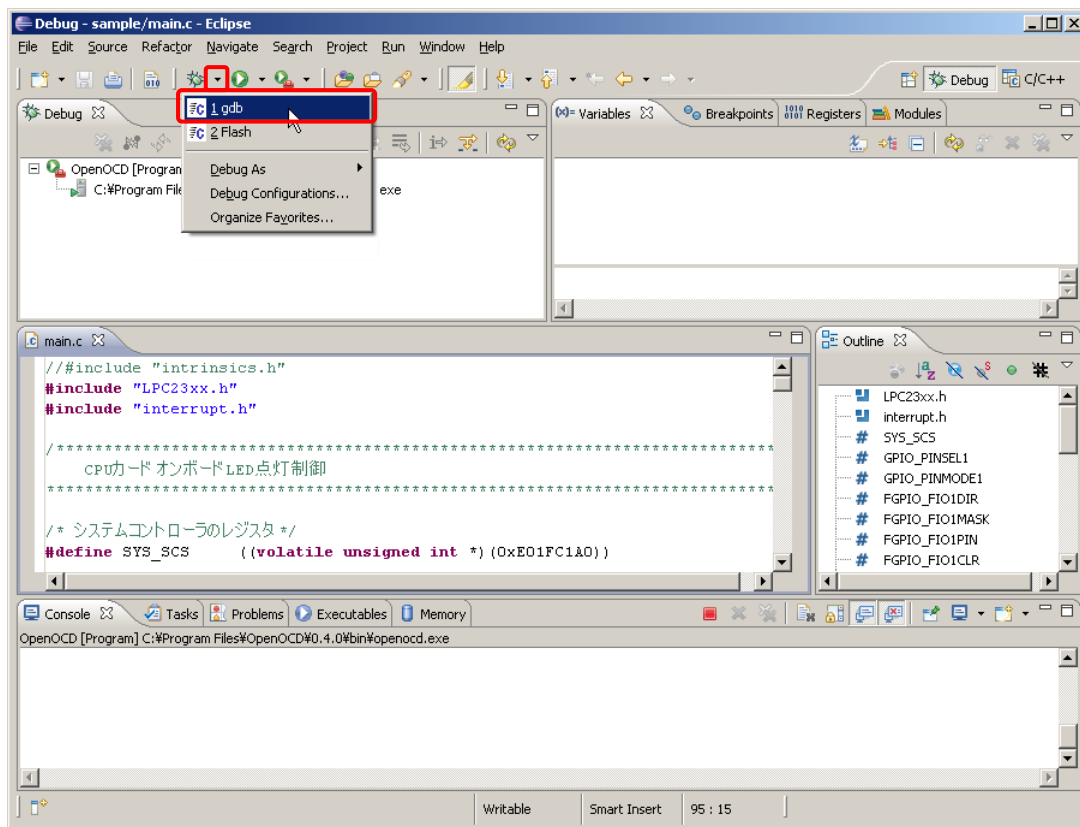
Flash への書込みが開始されます。続けてデバックを行う場合には「2. 4. 7 デバック」へ進み、終了する場合には、「2. 4. 8 デバック終了作業」へ進んでください。



2.5.2 デバック

Eclipse からターゲットボードのデバックを行います。

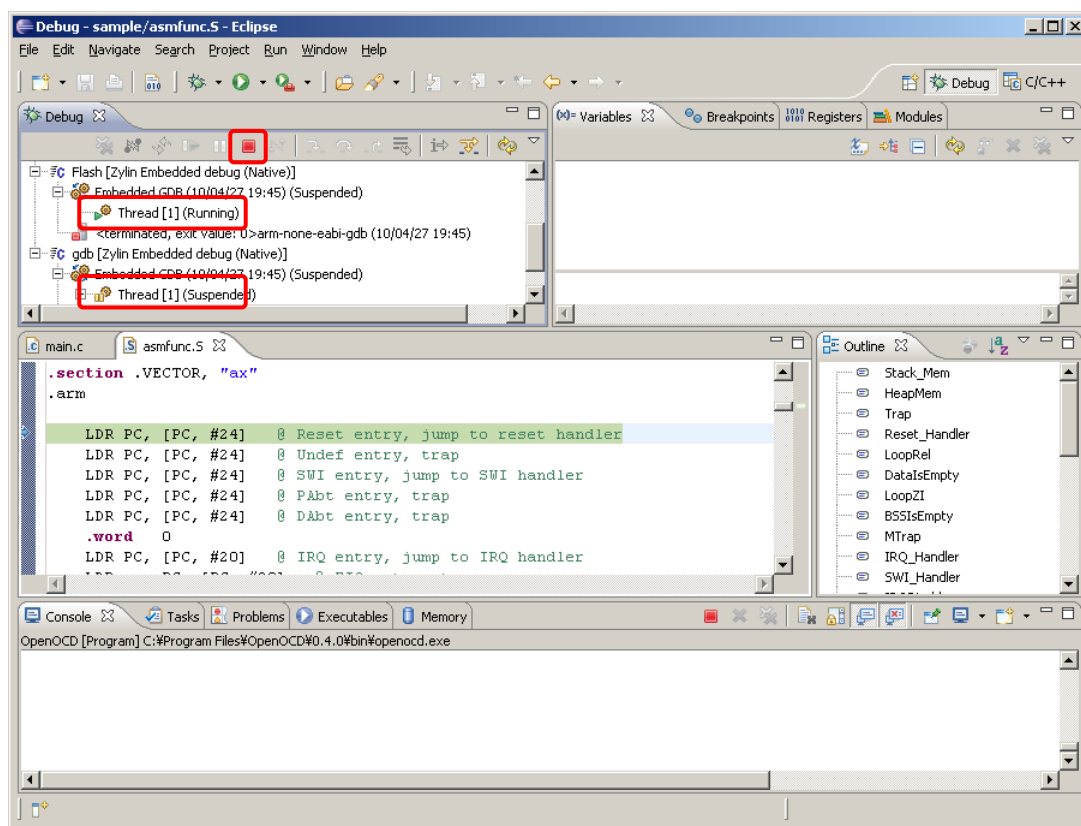
- ① Debug の▼ボタンをクリックし、gdb を選択します。
デバック作業が行えるようになりますので、デバック作業を行ってください。



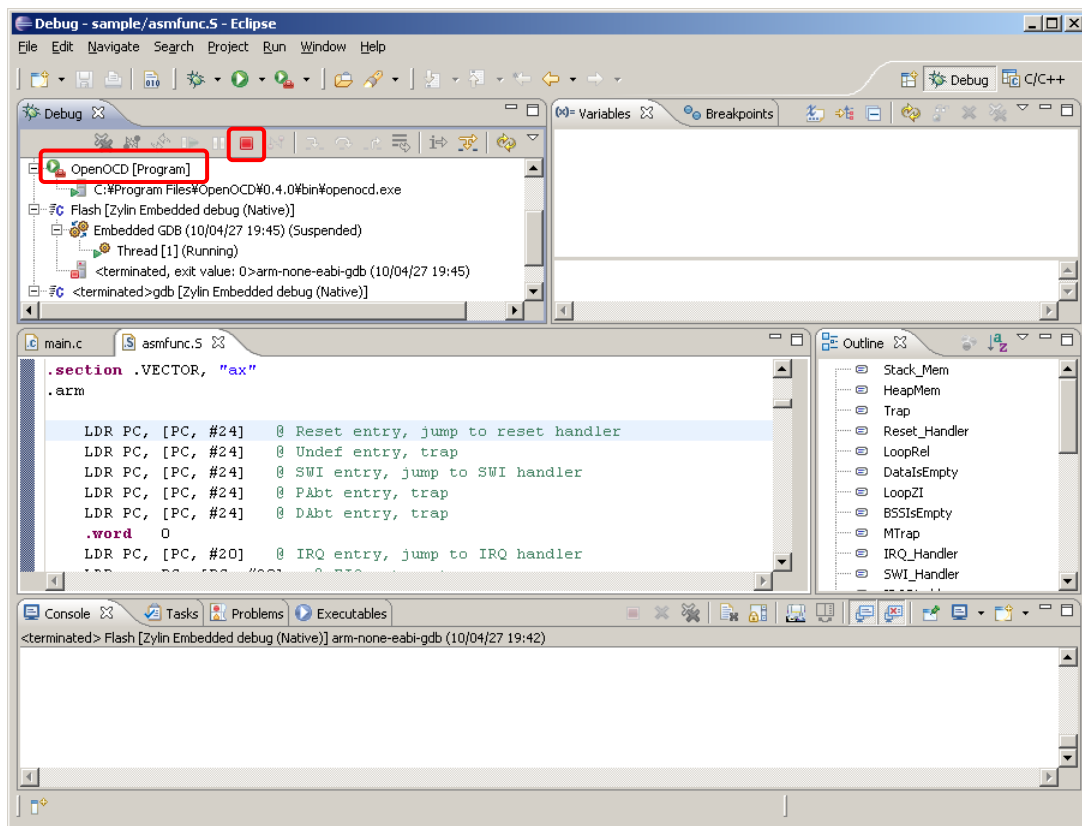
2.5.3 デバック終了作業

デバックの終了作業を行います。

- ① Debug ウィンドウ上の「gdb」と「Flash」の Thread をそれぞれ選択し、停止ボタンをクリックしてください。



- ② Debug ウィンドウ上の「OpenOCD」を選択し、停止ボタンをクリックしてください。



- ③ ターゲットボードの電源を切ってください。

ご注意

本文書の著作権は（株）アルファプロジェクトが保有します。

本文書の内容を無断で転載することは一切禁止します。

本文書の内容は、将来予告なしに変更されることがあります。

本文書に記載されている内容についての質問等のサポートは一切受け付けておりませんのでご了承ください。

本文書の内容については、万全を期して作成いたしました。万が一不審な点、誤りなどお気づきの点がありましたら弊社までご連絡下さい。

本文書の内容に基づき、アプリケーションを運用した結果、万一損害が発生しても、弊社では一切責任を負いませんのでご了承下さい。

商標について

- ・ Windows®の正式名称は Microsoft®Windows®Operating System です。
- ・ Microsoft、Windows は、米国 Microsoft Corporation.の米国およびその他の国における商標または登録商標です。
- ・ Windows®7、Windows®Vista、Windows®XP、Windows®2000 Professional は、米国 Microsoft Corporation.の商品名称です。

本文書では下記のように省略して記載している場合がございます。ご了承下さい。

- ・ Windows®7 は Windows 7 もしくは Win7
 - ・ Windows®Vista は Windows Vista もしくは WinVista
 - ・ Windows®XP は Windows XP もしくは WinXP
 - ・ Windows®2000 Professional は Windows 2000 もしくは Win2000
- ・ その他の会社名、製品名は、各社の登録商標または商標です。



株式会社アルファプロジェクト
〒431-3114
静岡県浜松市東区積志町 8 3 4
<http://www.apnet.co.jp>
E-MAIL : query@apnet.co.jp