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

株式会社日昇テクノロジー

http://www.csun.co.jp

info@csun.co.jp

2012/4/25

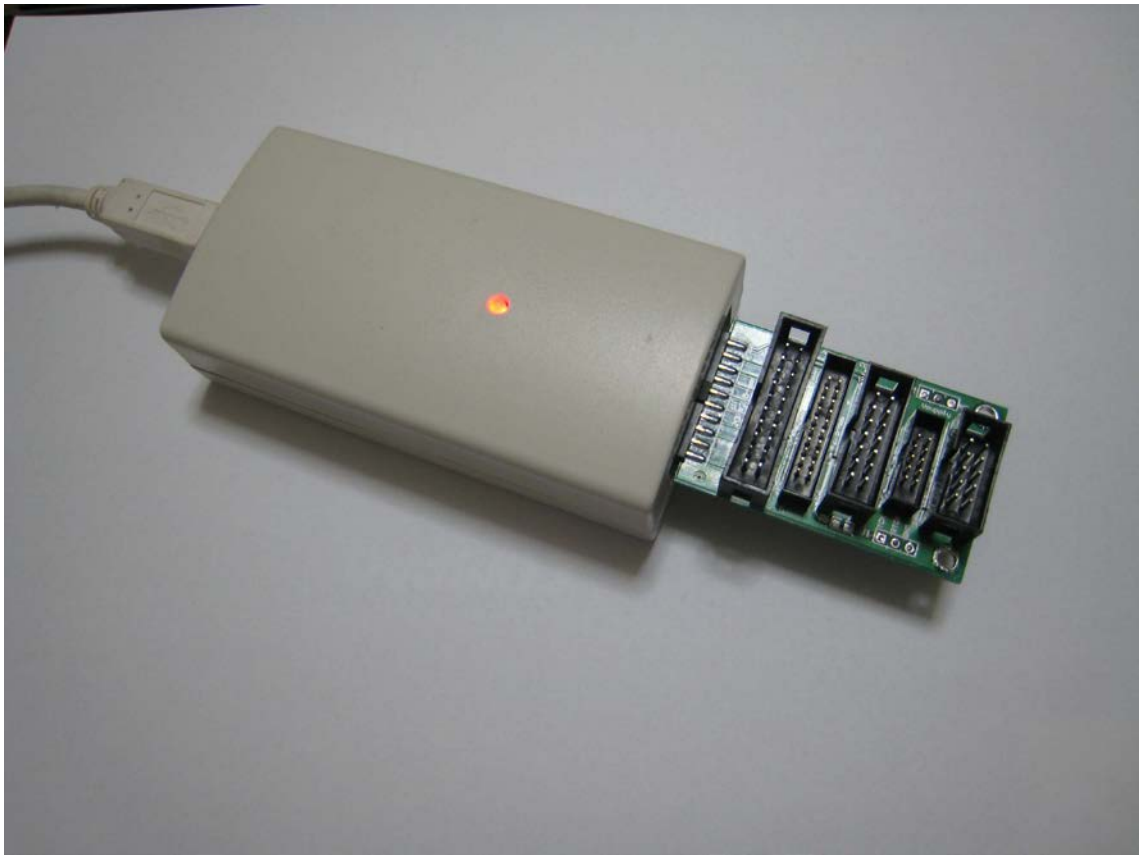
• 修正履歴

•

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

目次

一、USB Open Link の概要



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

1.1 USB Open Link の主な特徴

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

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

1.2 USB Open Link がサーポート ARM コア

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

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

1.3 USB Open Link が利用できるソフトウエア

1)J-Mem:メモリを参照及び修正

2)J-Link Server:(TCP/IP を介し Open Link と接続)

3) J-Flash：独自の Flash プログラミングをサーポート、量産ソリューションとして扱われます。

4) RDI Flash BP：RDI に基づき、Flash にブレークポイントを無制限設定できます。

5) RDI Flash DLL：RDI に基づき、独自の Flash プログラミングをできます。

6) GDB server：GDB 環境でデバッグできます。

二、USB Open Link の USB ドライバインストール

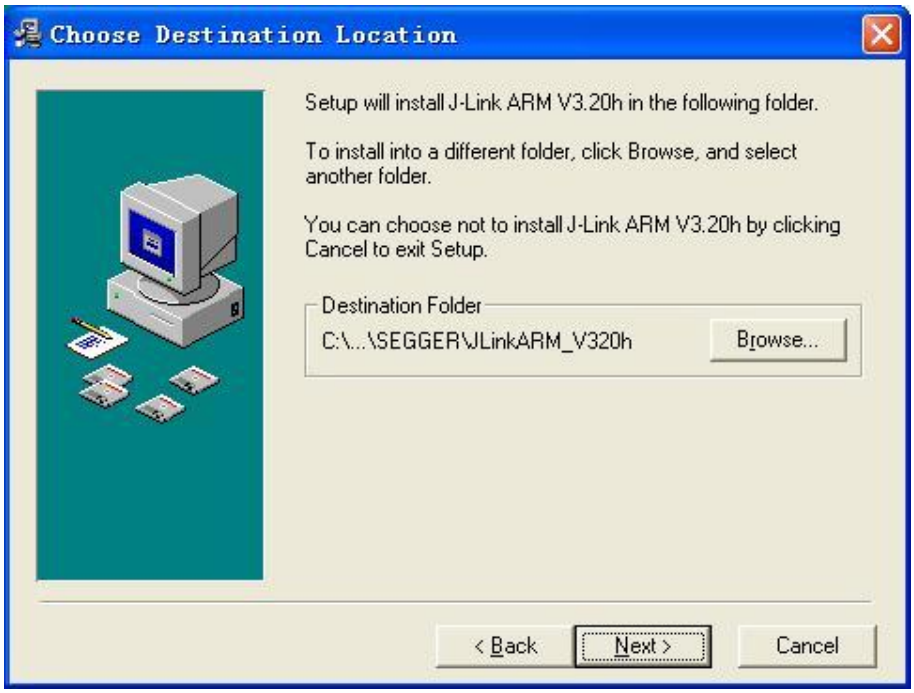
ドライバインストール用のファイルは弊社ホーム下記 URL からダウンロードできます。

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

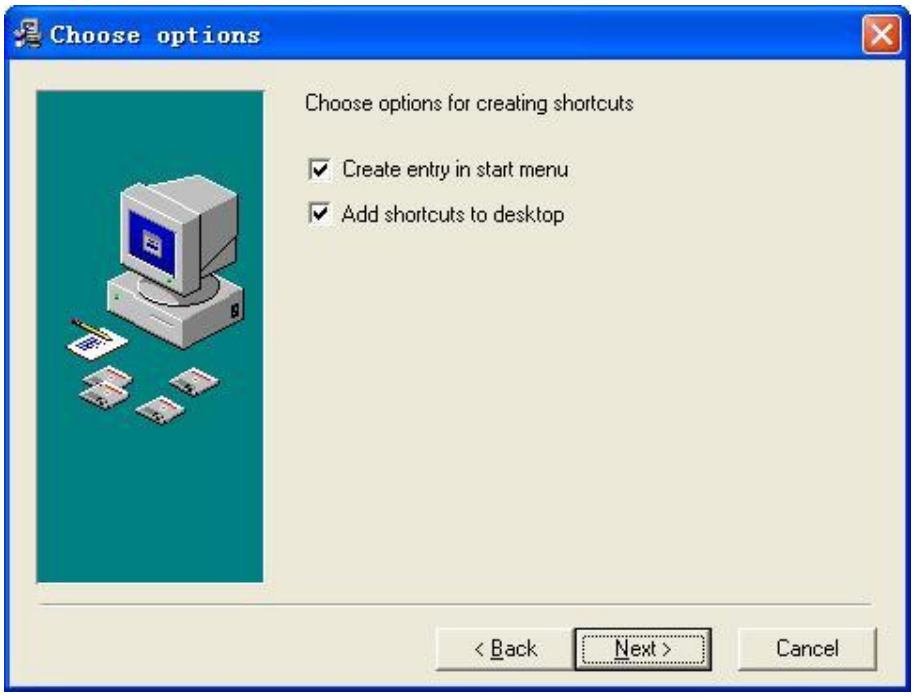
SEGGER 社様のソフトウエアを利用しておりますので、直接 SEGGER 社様ホームページから最新の USB ドライバもダウンロードできます。

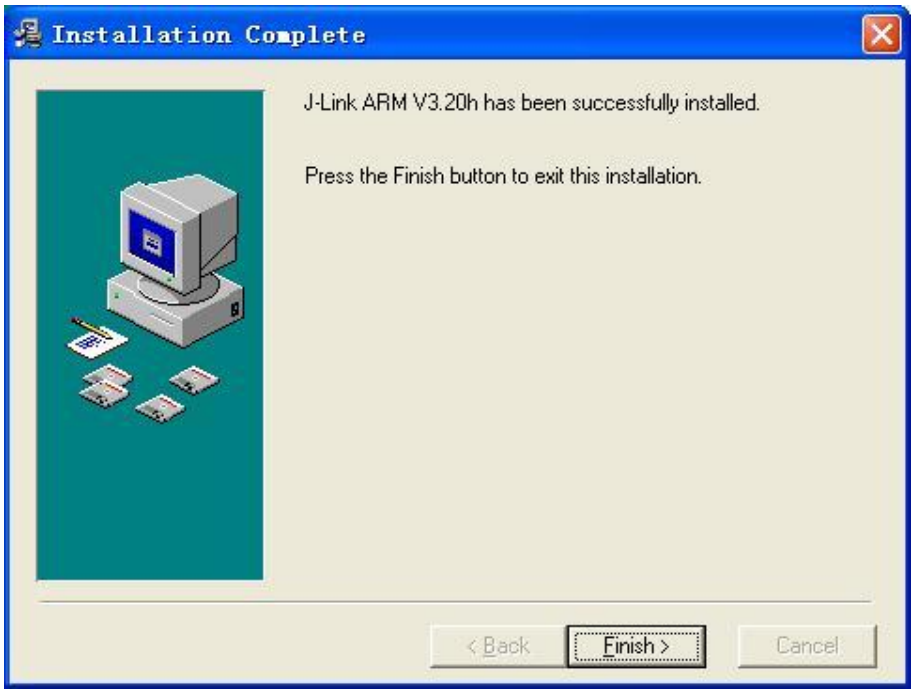
http://www.segger.com/cms/jlink-software.html

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




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

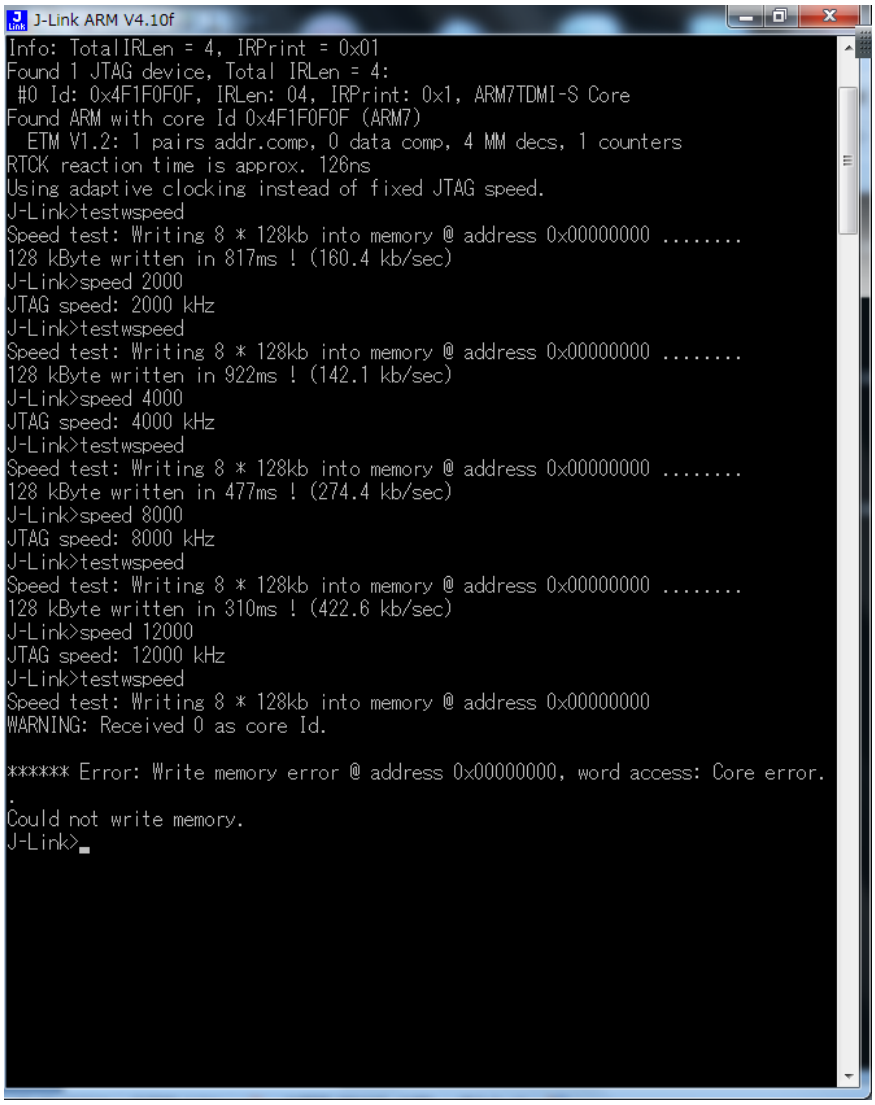

Choose options
Choose options for creating shortcuts
Create entry in start menu
Add shortcuts to desktop
< Back
Next >
Cancel



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

完了後、デスクトップ上に、下記二つアイコンがあります。

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

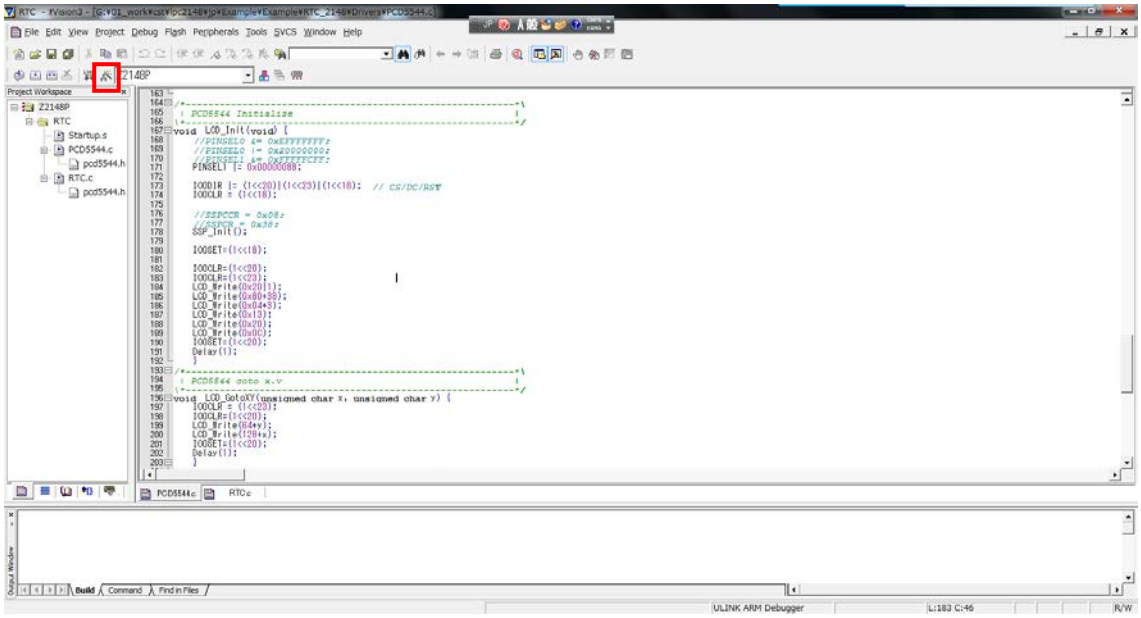


メモ：*ARM7TDMI-S*コアの特徴のため、*LPC2000*シリーズのJTAG速度は最大までクロックの1/6に達しています。一番早い速度は4.8Mとなり、*JTAG*速度は*4.8M*を超える場合、*Open-LINK ARM*は*LPC2000*ボードが見つかれませんというメッセージが出てくるかもしれません。これは*LPC2000*コアの制限です、Open Linkと関係がありません。

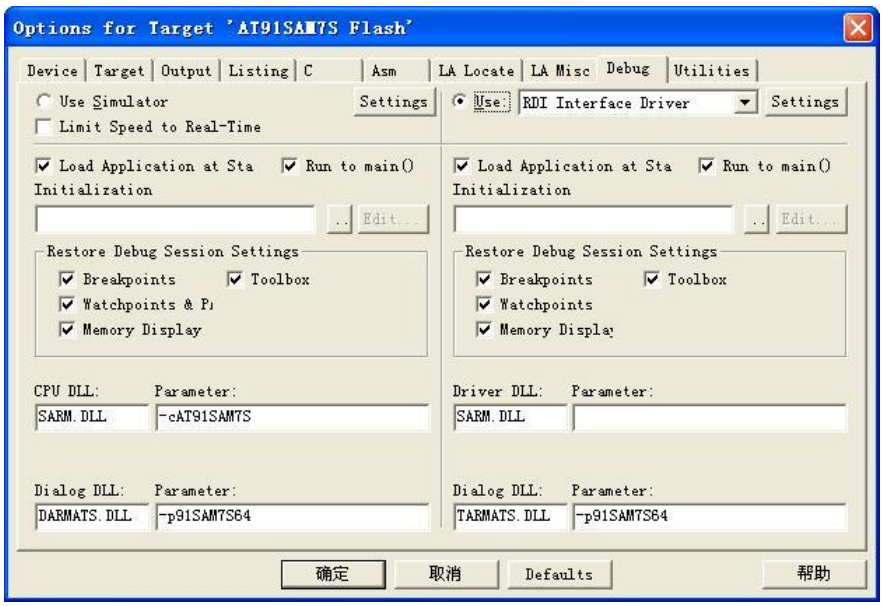
三、各開発統合環境に USB Open Link の設定

3.1 Keil での設定

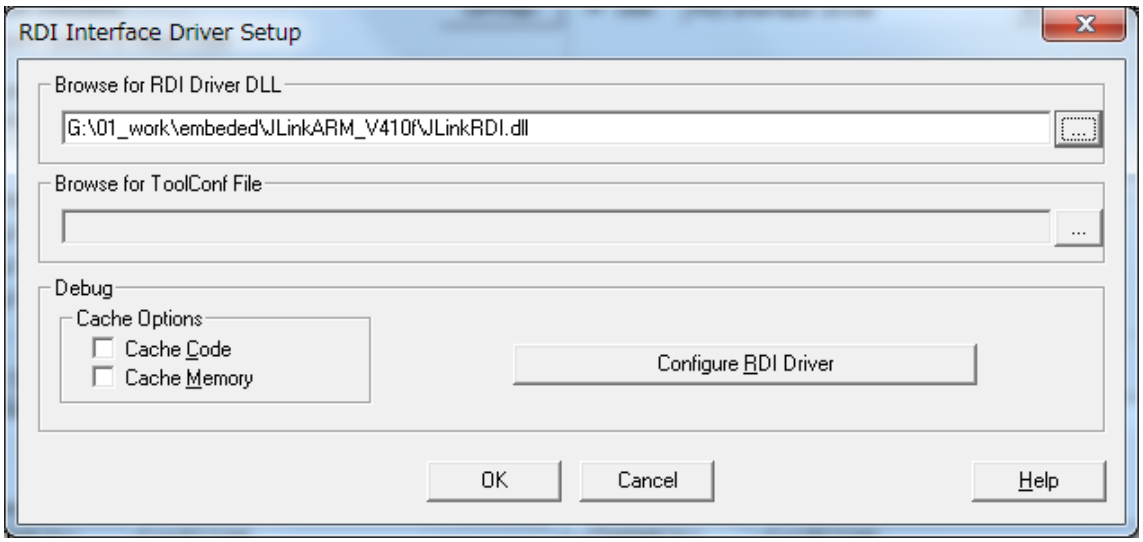
1)Keil 起動後、あるプロジェクトを開く



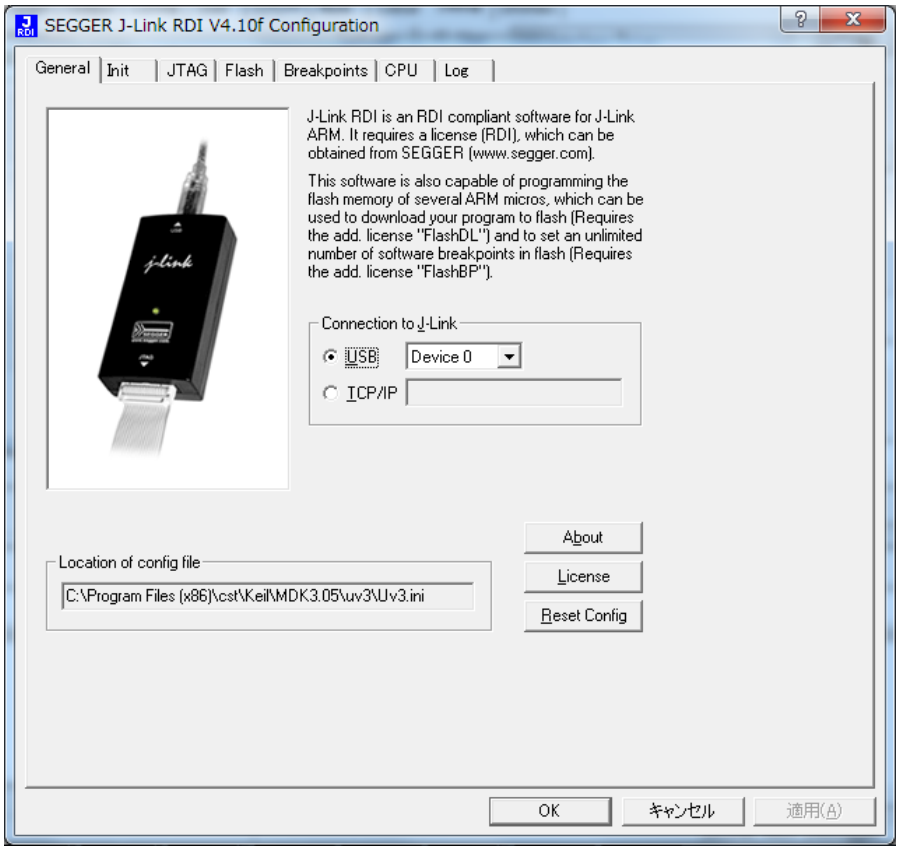
2)「Debug」タブで「RDI Interface Driver」を選択し、「Settings」をクリックする



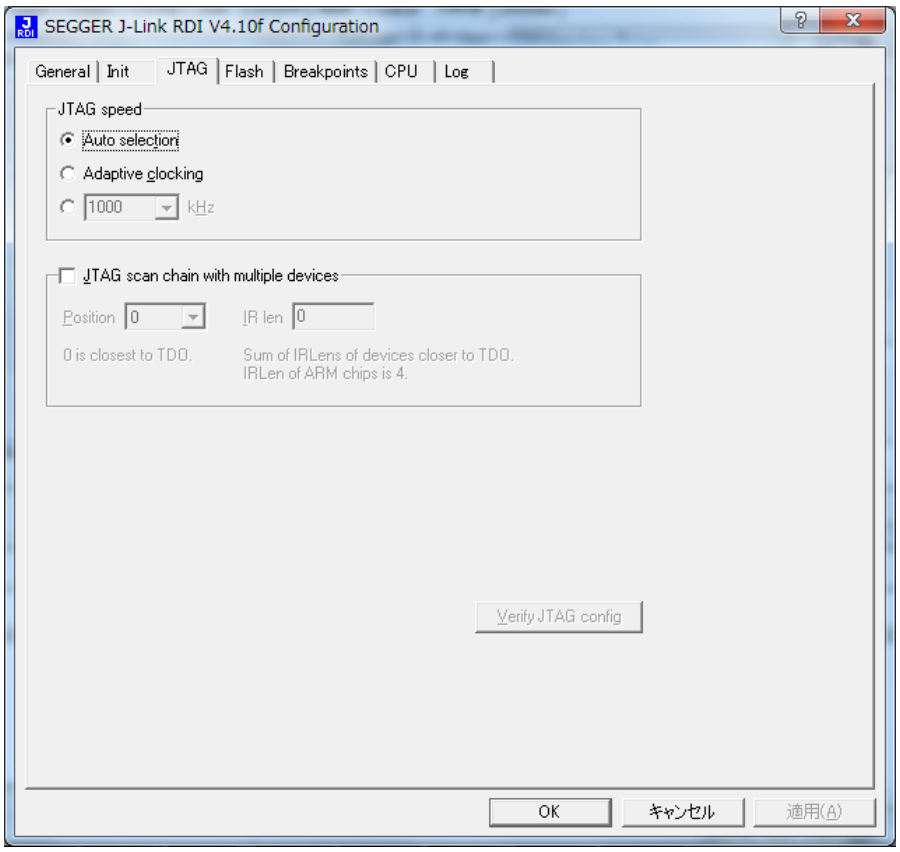
3)「…」をクリックして、Open LINK のインストールディレクトリに行く。



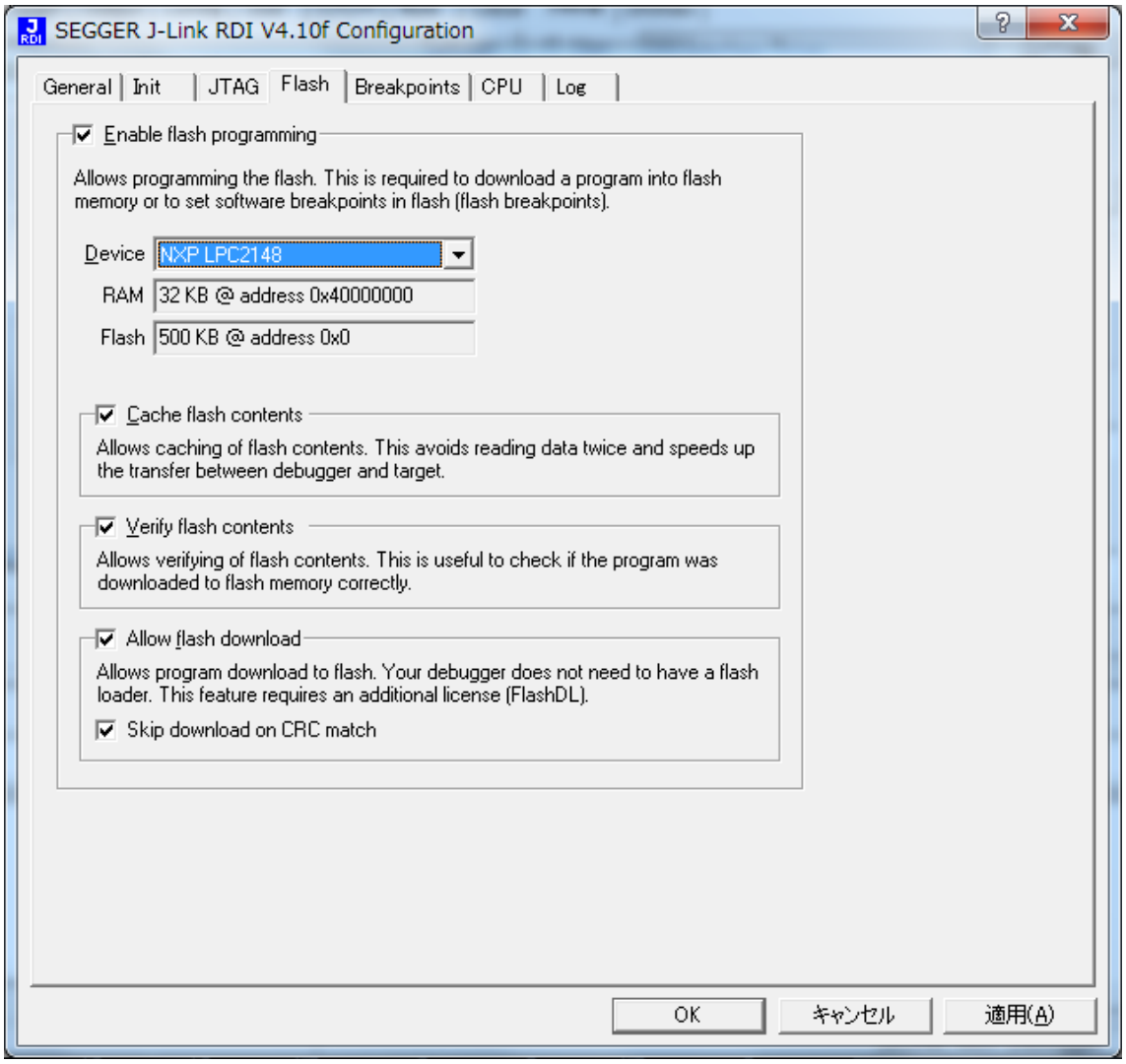
4)「Configure RDI Driver」をクリックして、次のタブが表示されます、PCでデバッグの場合、USBが使えます。ローカルエリアネットワークのデバッグの場合、TCP/ IPを選択し、Open LINKをリンクしているPCのIPアドレスを指定します。



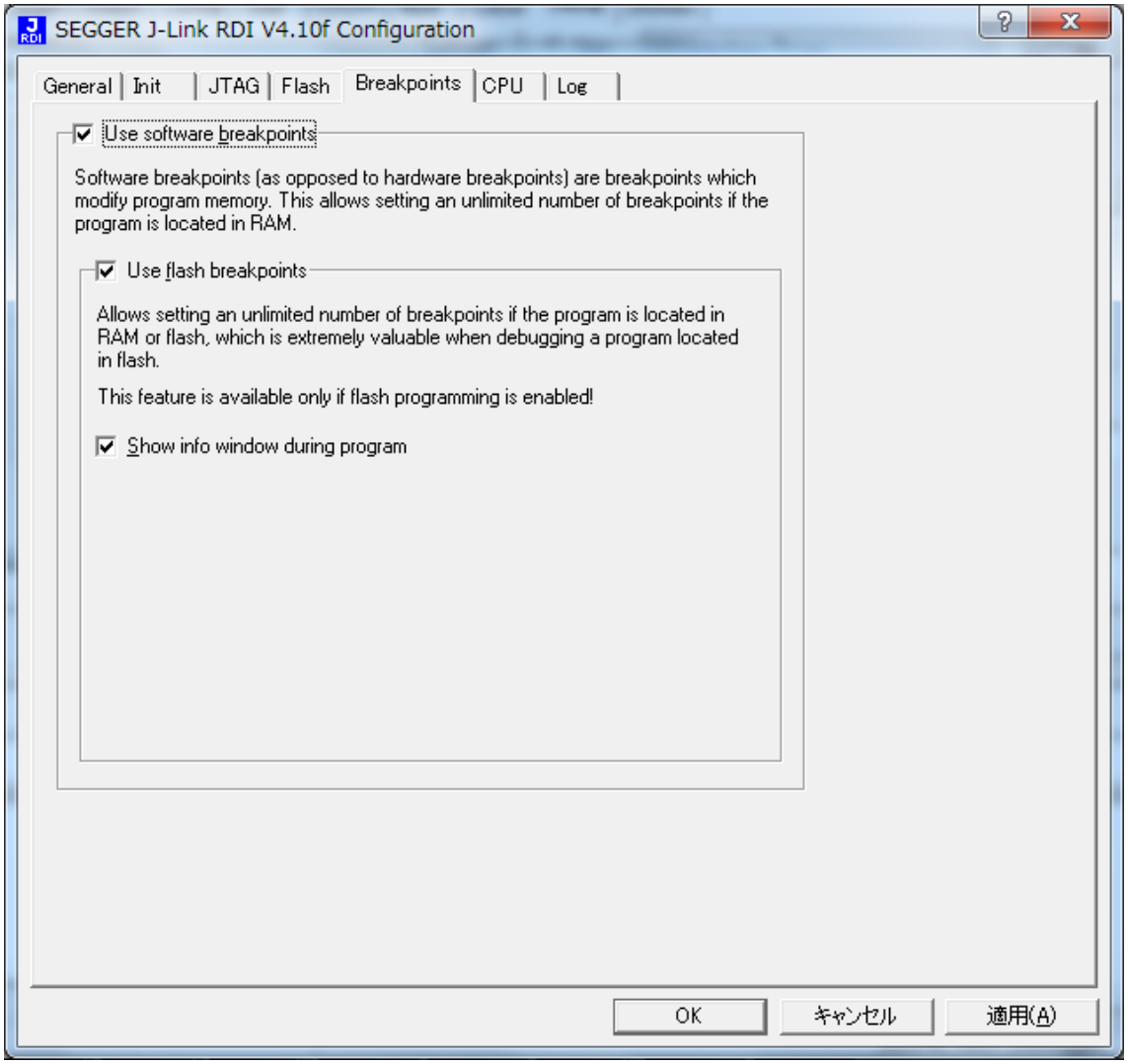
5）JTAGのスピードを設定する。- Sコアの場合はAutoをを使用することをお勧めする。非-Sコアの場合は、直接12Mの最大速度を使用することができます。使用中不安定な現象が発生した場合は、JTAGのクロック速度を適切に低値に調整をお勧めする。



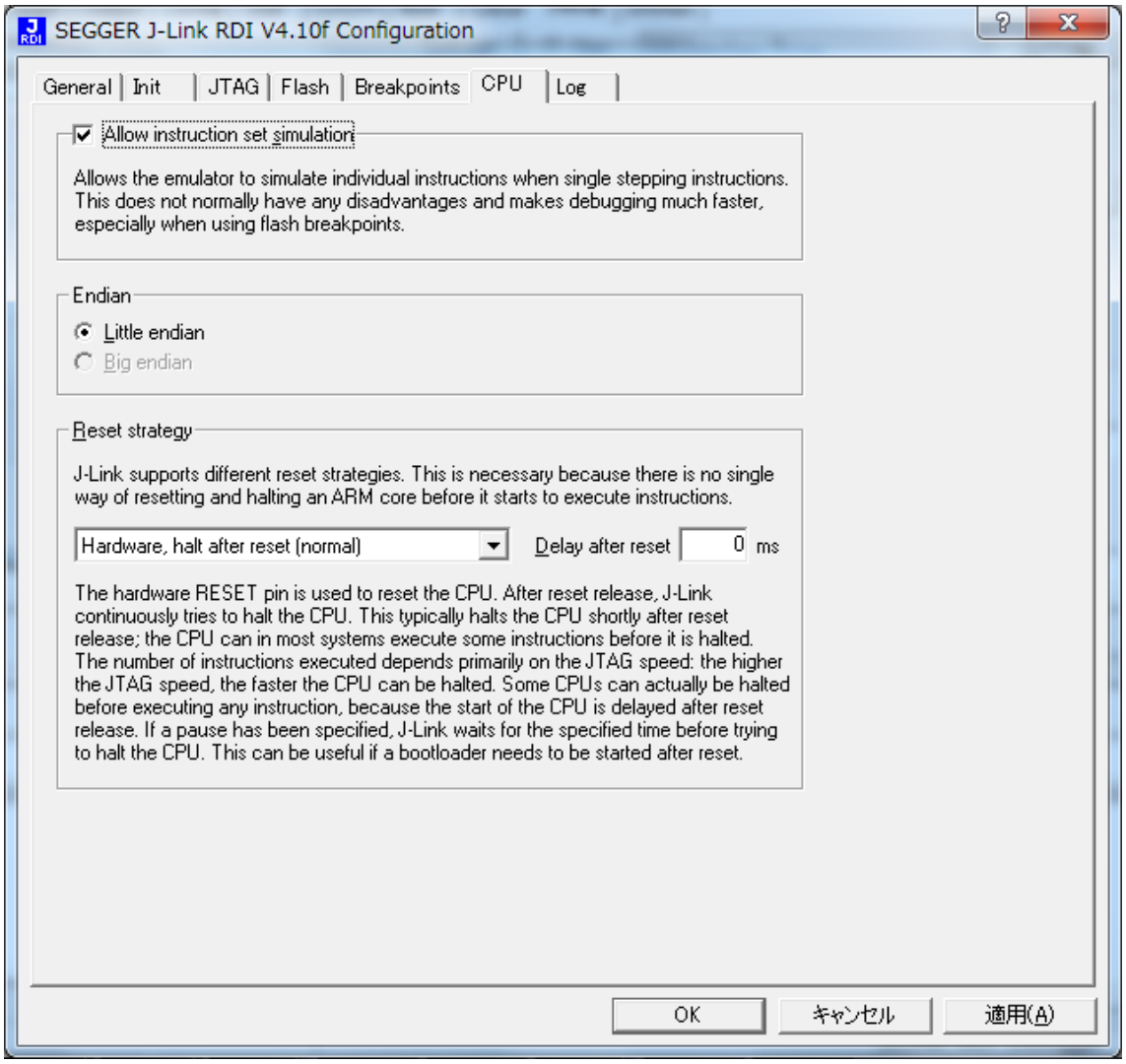
6）フラッシュプログラミング機能：オンチップフラッシュの ARM チップの場合は、この機能が使える。そうすると、デバッグする前に Open Link はフラッシュをプログラミングする。



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

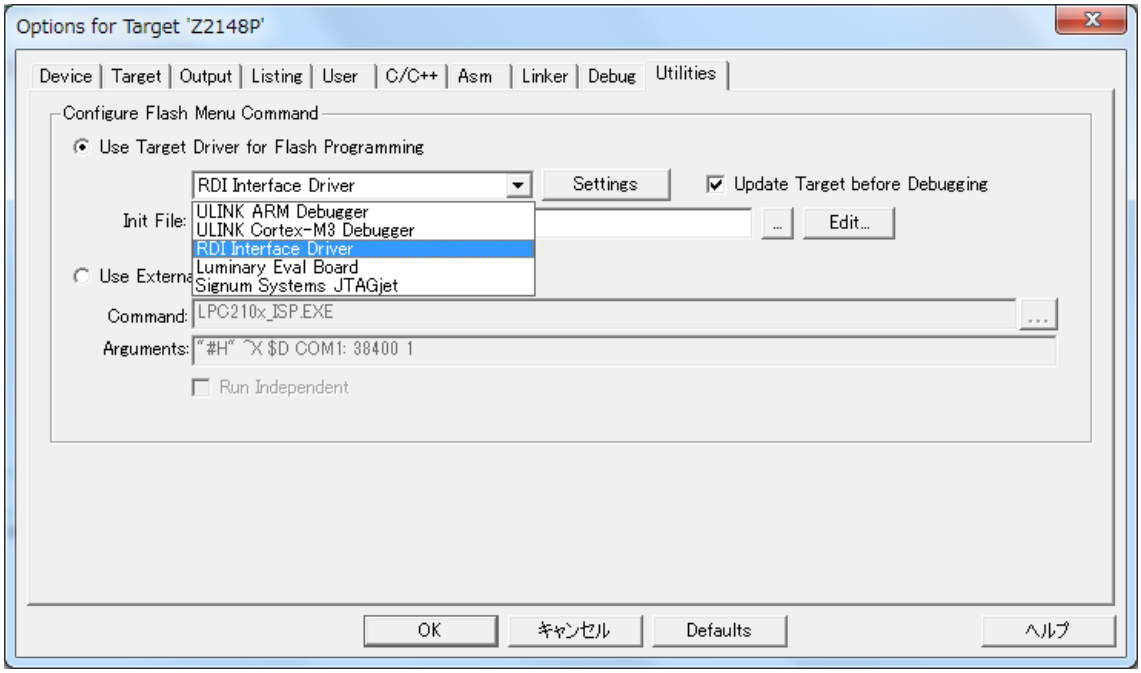


8）次はリセットの方針を設定できます。いくつかの方針の選択肢があって、また、リセット後の同じ時間をリセットする、いくつかの方針の選択肢をする場合は遅延時間の設定ができる。この設定はリセットするのに時間かかる時は利用できる。例えば AT91RM9200。

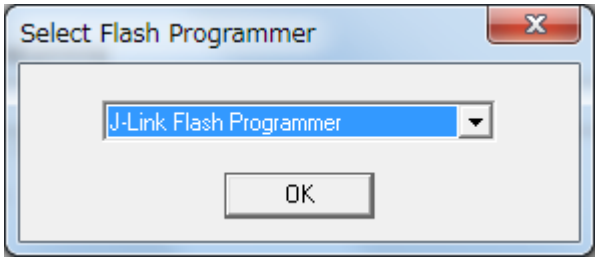


以上の設定は Open Link を使ってデバッグする時に設定内容となります。

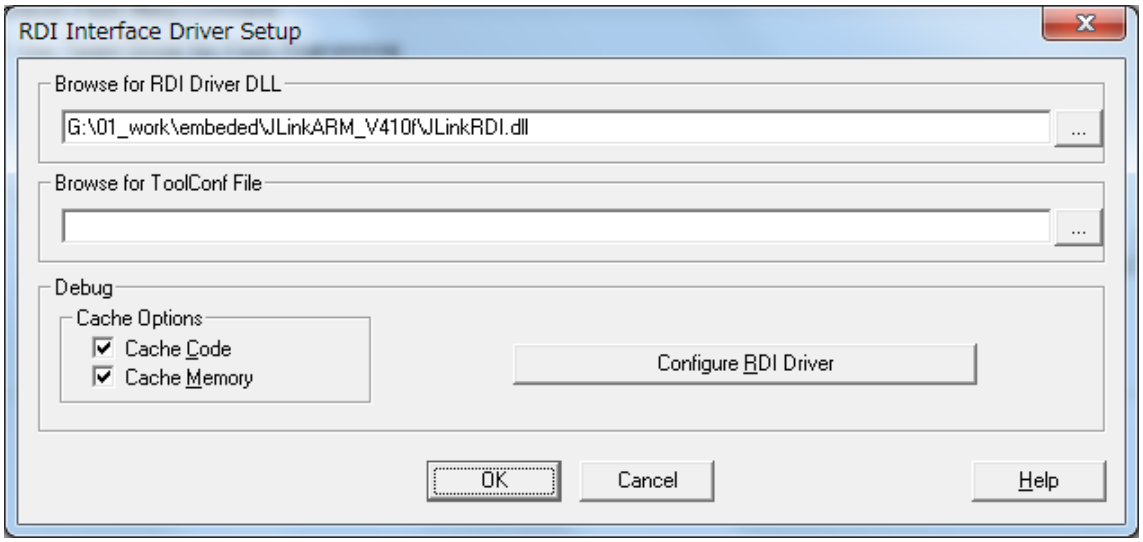
9) KEIL の「DOWNLOAD」機能を使う場合は「Utilities」のメニューで「Debug」上と同じ設定が必要である。



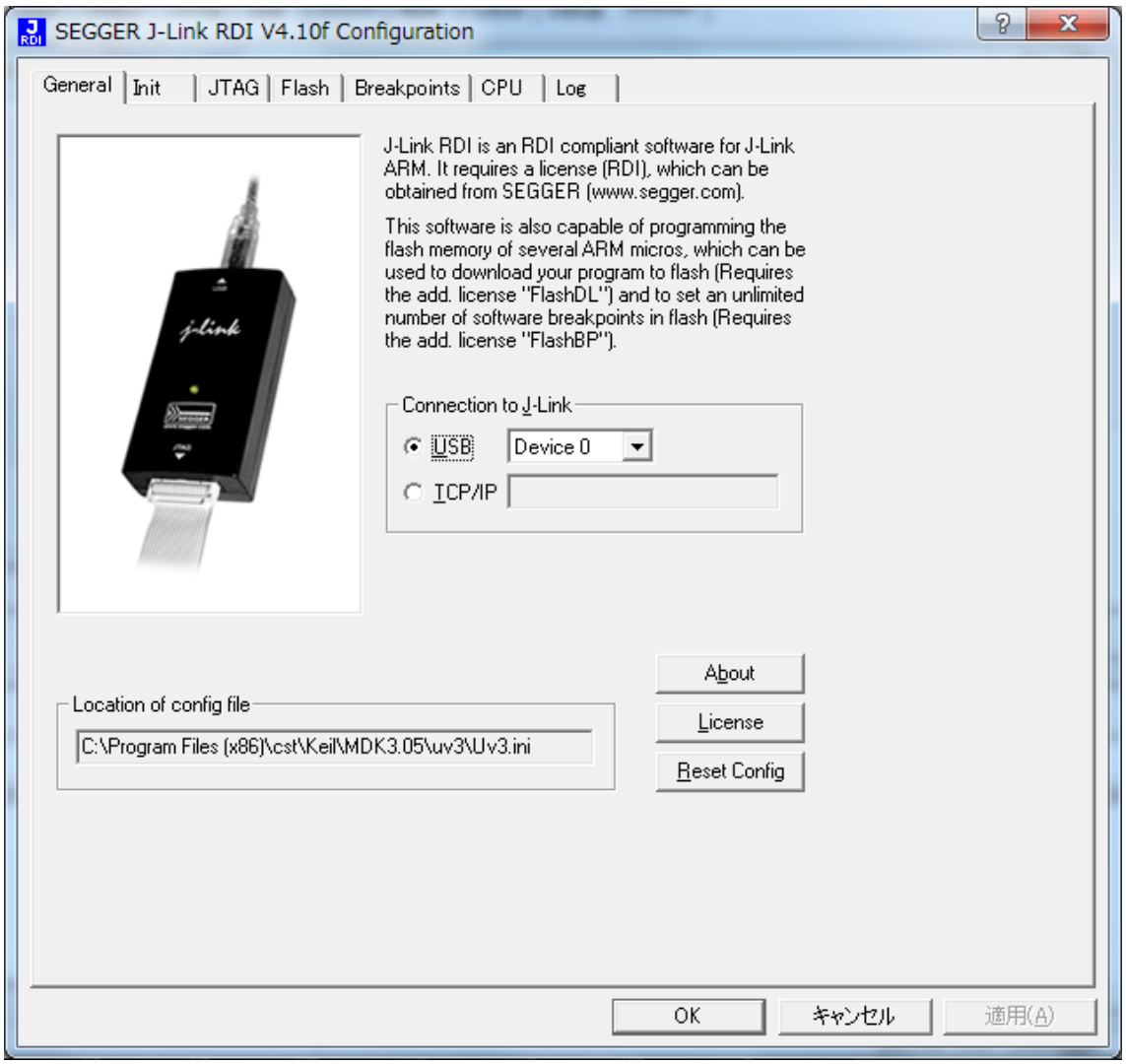
10)「RDI Interface Driver」を選択し，「Settings」をクリックする。



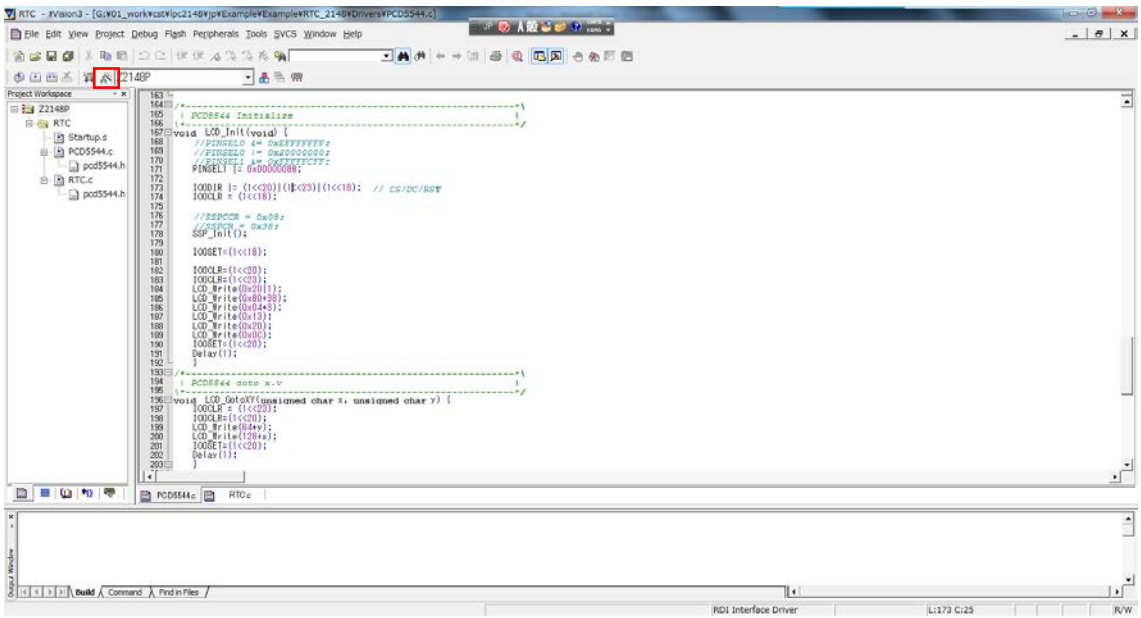
11)「J-Link Flash Programmer」を選択する。



12)次の設定は“Debug”での設定と同じである。



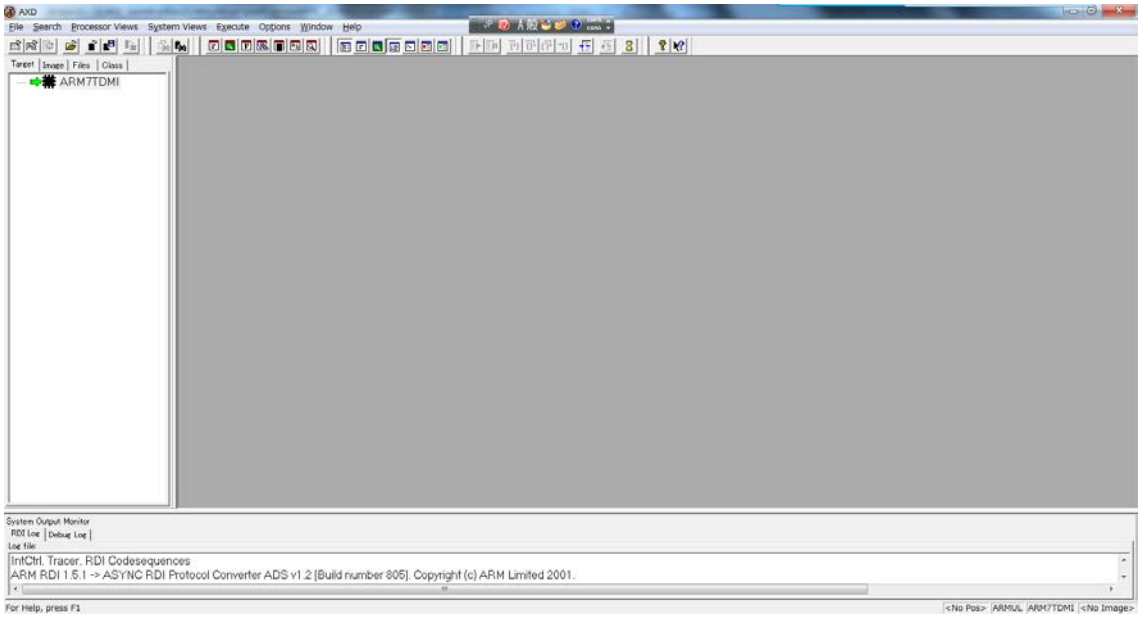
上記設定した後、下記ボタンをクリックしてダウンロードができる。



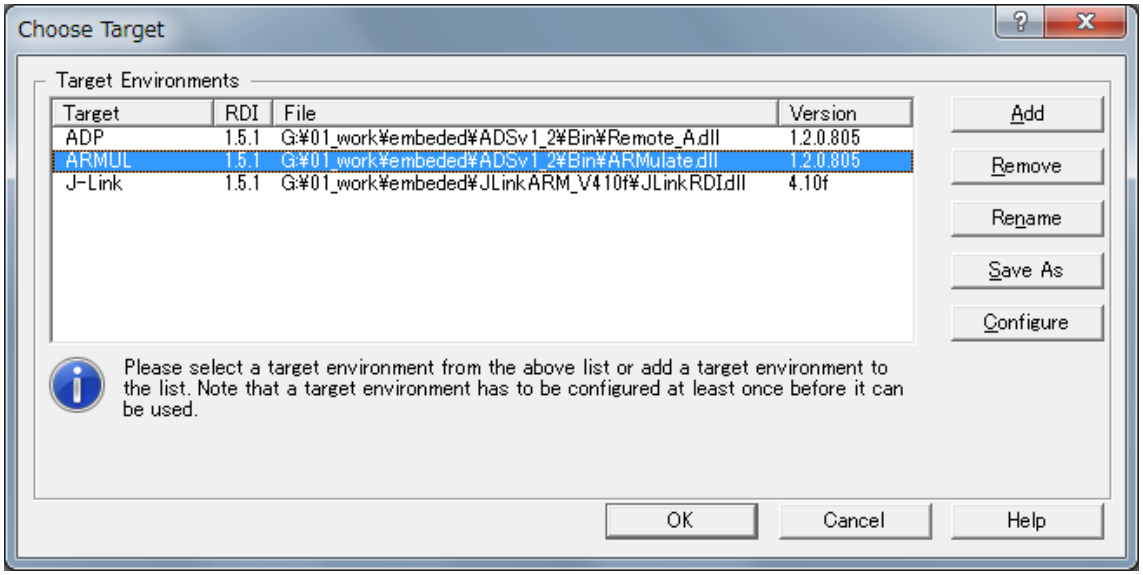
注意事項：この機能はオンチップフラッシュの AMR7/9 チップしかサーポートできません。

3.2 ADS での設定

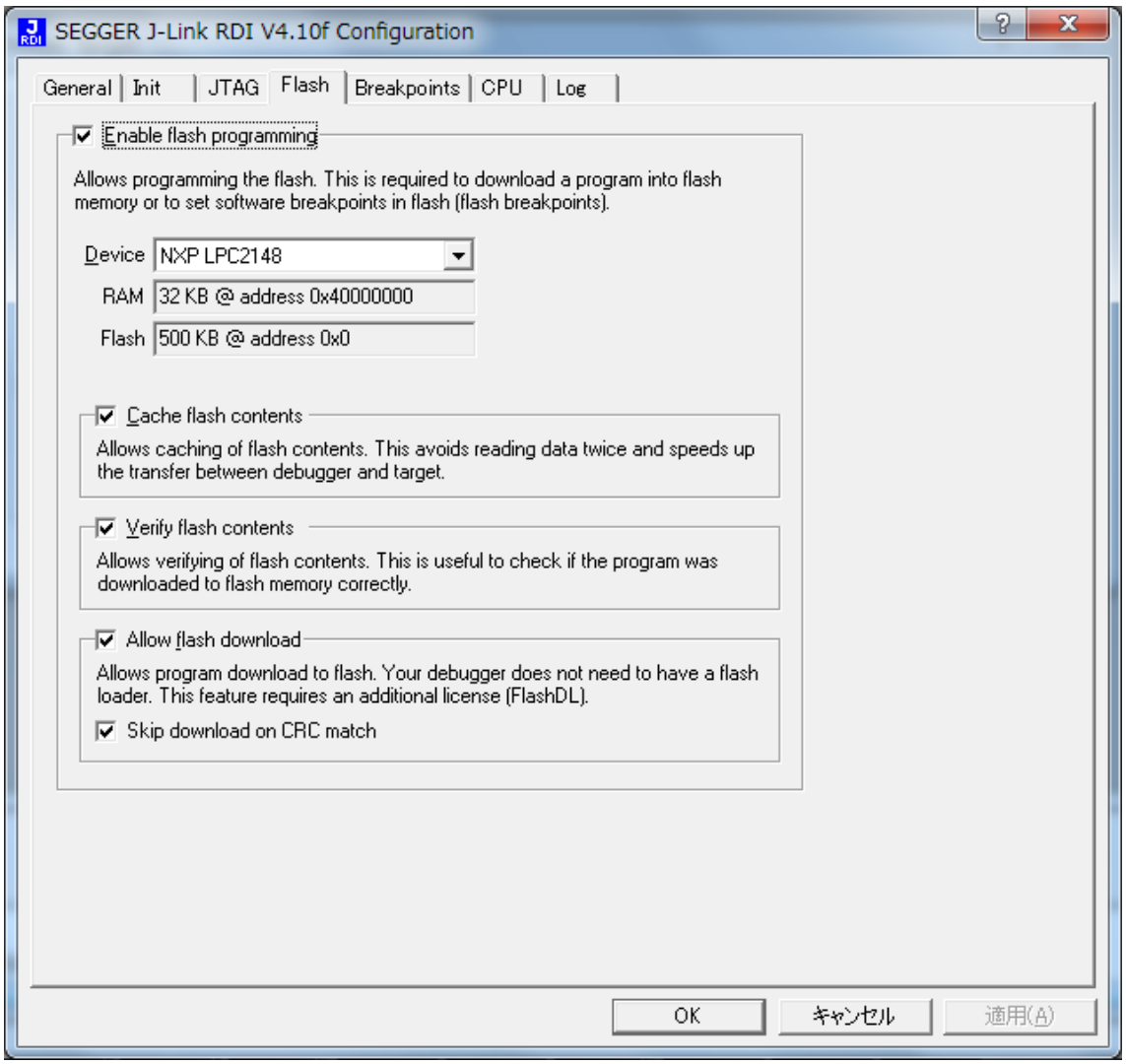
1)ADX を起動



2) 「Add」をクリックし，JLINKRDI.DLL を選択する

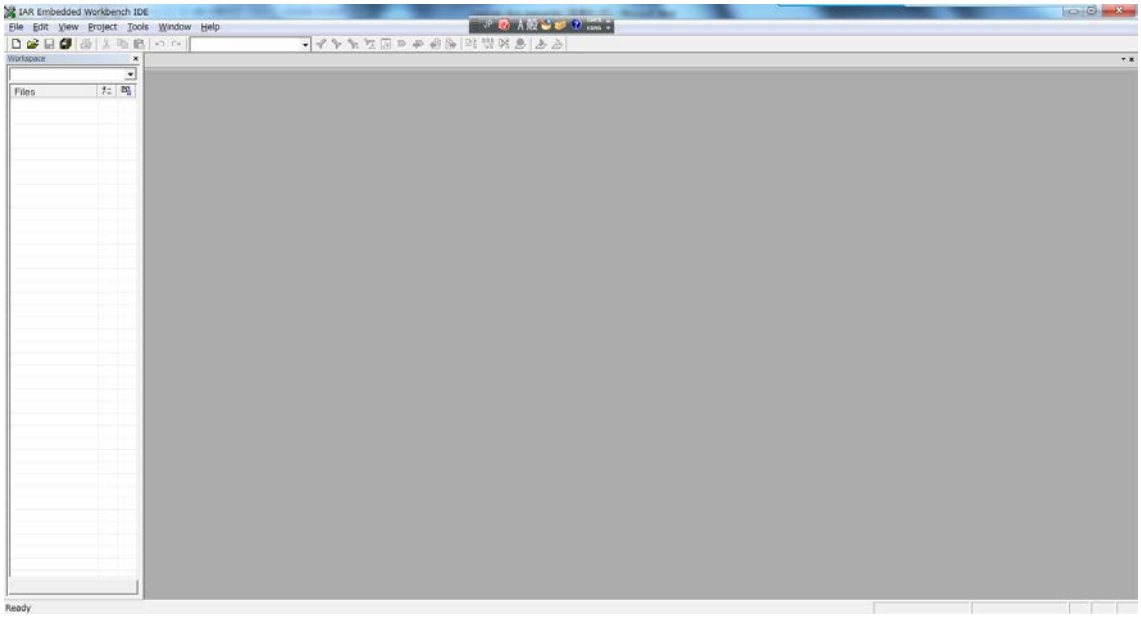


3) 「Configure」をクリックすると，下記内容を出る



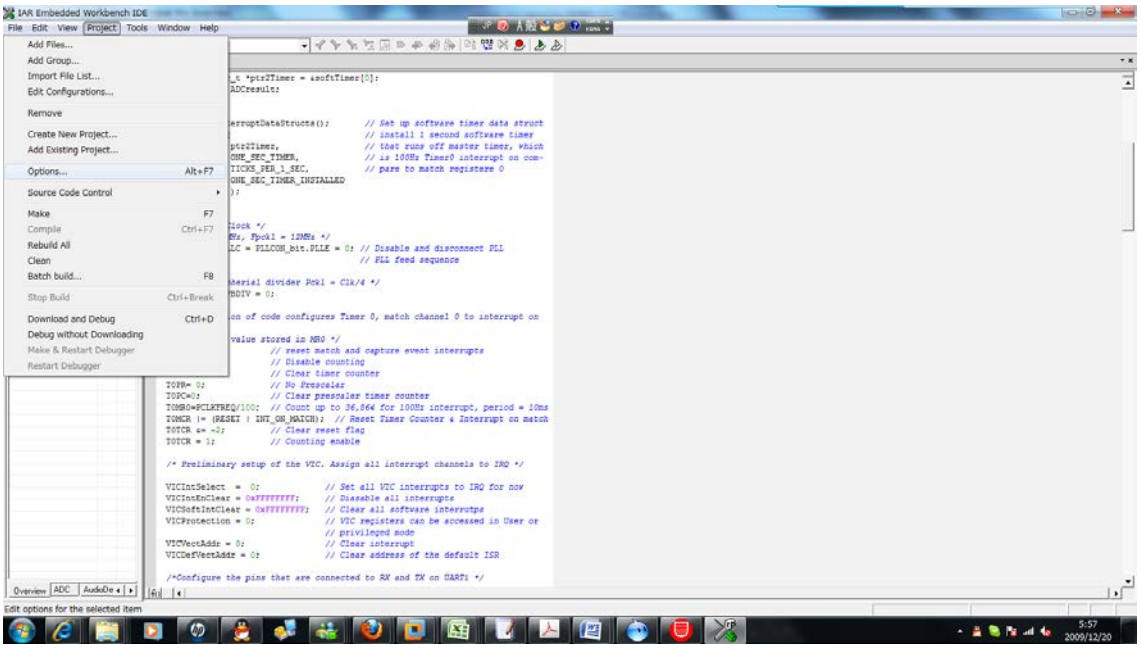
3.3 IAR での設定

1)IAR を起動

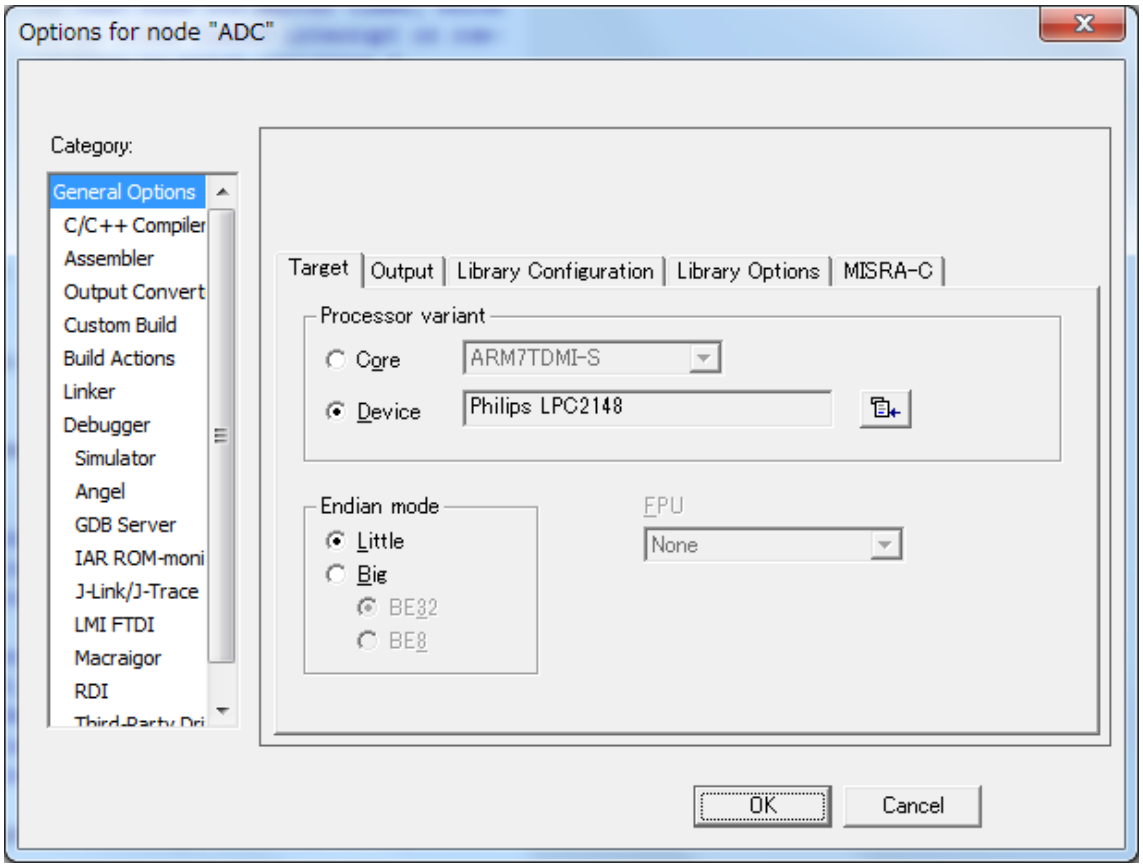


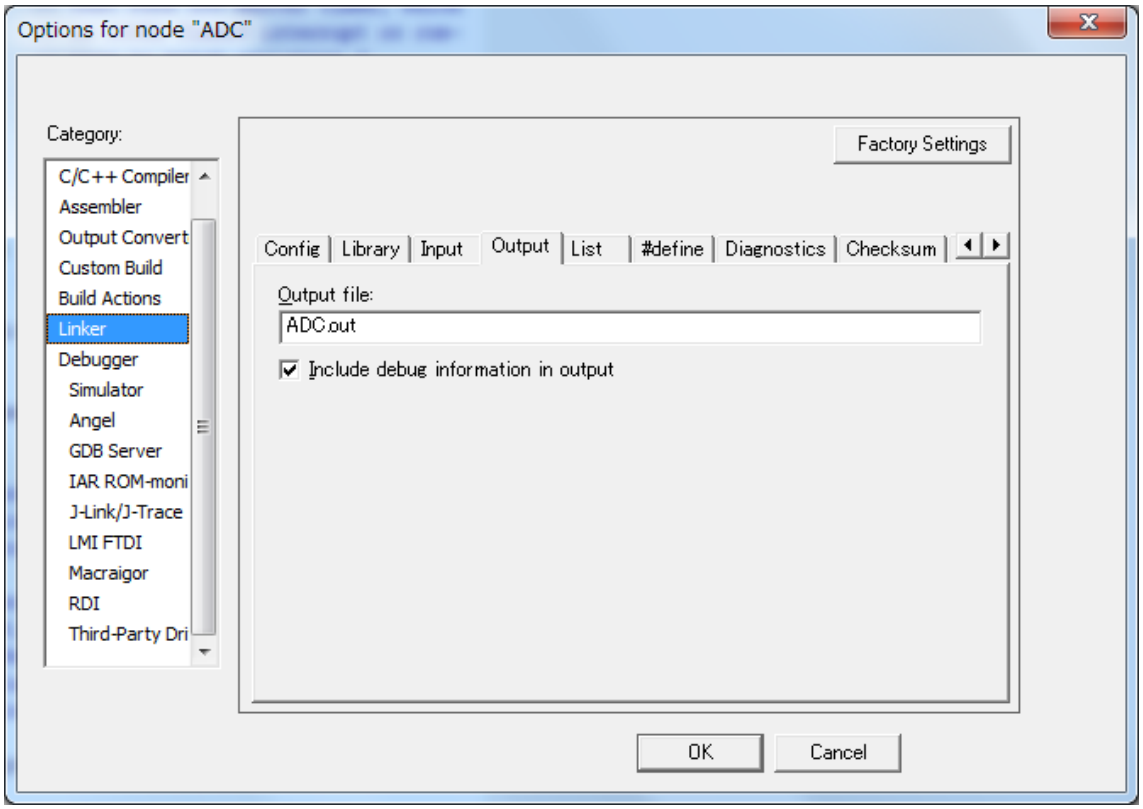
2)IAR の設定

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

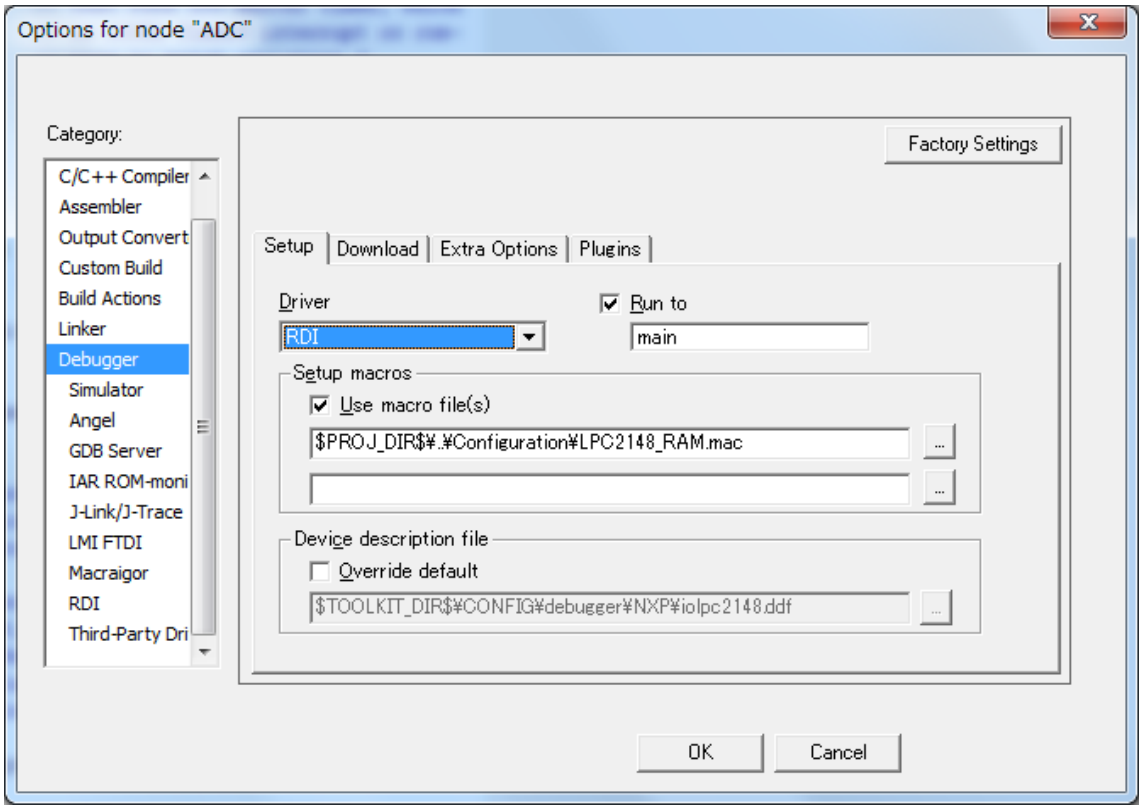


3)オプションを設定

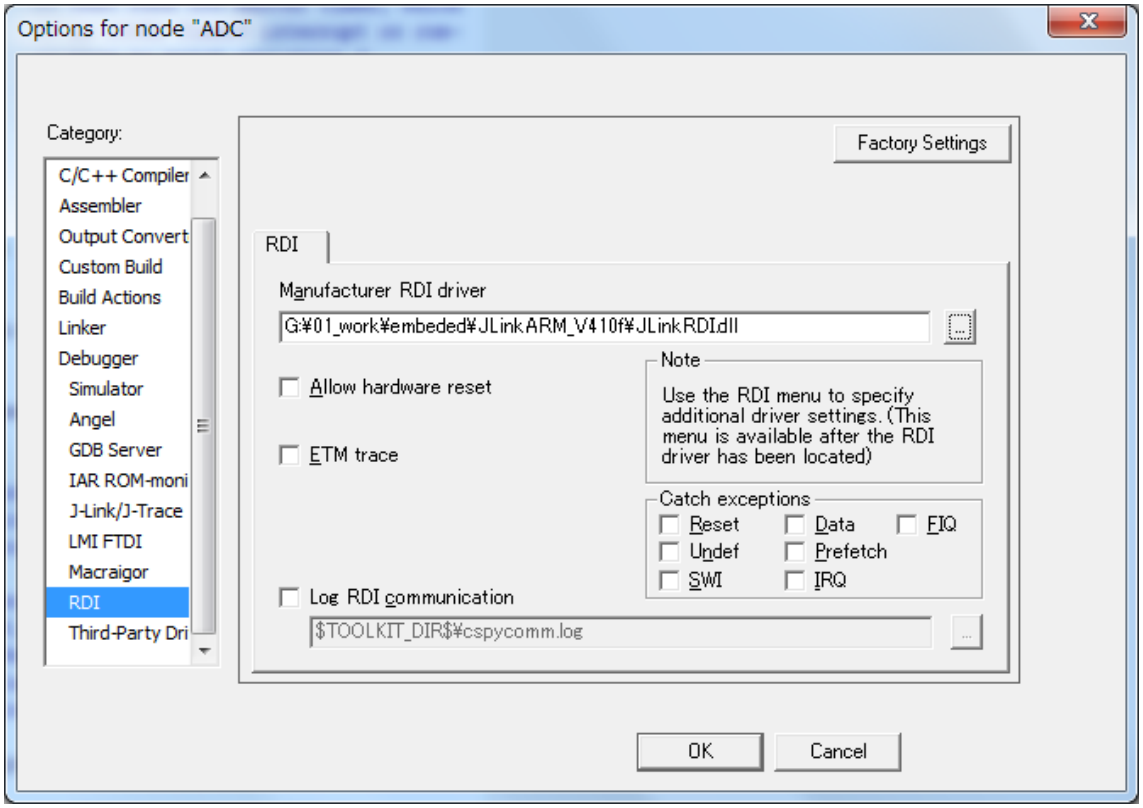

Options for node "ADC"
Category:
General Options
C/C++ Compiler
Assembler
Output Convert
Custom Build
Build Actions
Linker
Debugger
Simulator
Angel
GDB Server
IAR ROM-moni
J-Link/J-Trace
LMI FTDI
Macraigor
RDI
Target
Output
Library Configuration
Library Options
MISRA-C
Processor variant
Core
ARM7TDMI-S
Device
Philips LPC2148
Endian mode
Little
Big
BE32
BE8
FPU
None
OK
Cancel


Options for node "ADC"
Category:
C/C++ Compiler
Assembler
Output Convert
Custom Build
Build Actions
Linker
Debugger
Simulator
Angel
GDB Server
IAR ROM-moni
J-Link/J-Trace
LMI FTDI
Macraigor
RDI
Third-Party Dri
Factory Settings
Config
Library
Input
Output
List
#define
Diagnostics
Checksum
Output file:
ADC.out
Include debug information in output
OK
Cancel

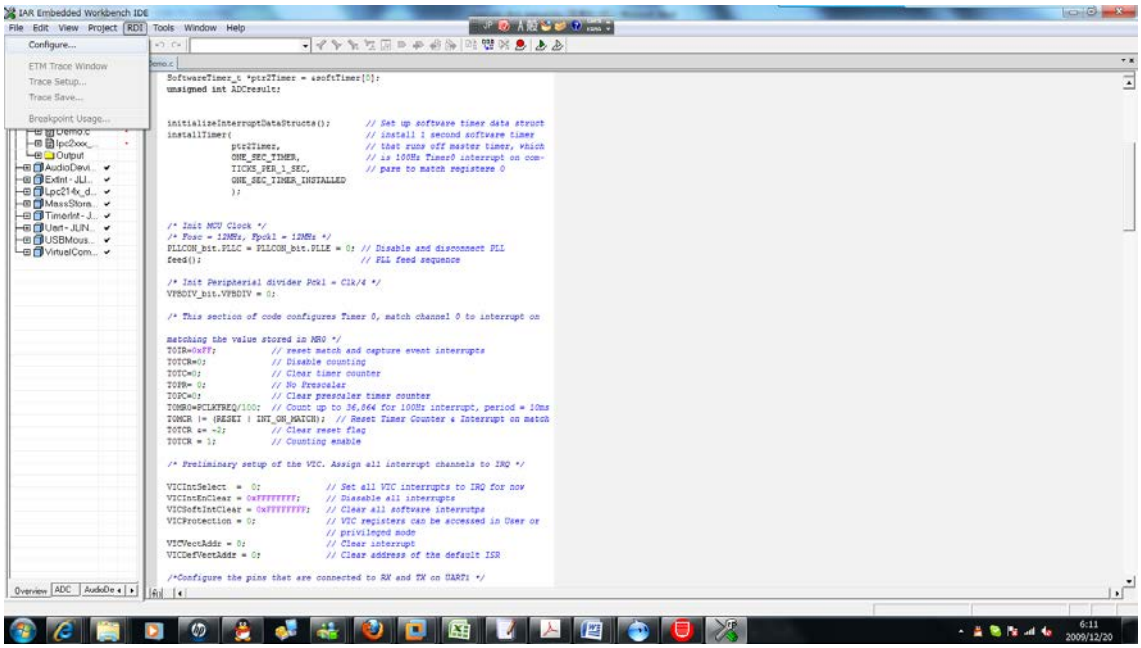
a)IAR 社の JLINK の場合は，“J-LINK/J-TRACE”を選択してください。全機能バージョンの JLINK の場合は“J-LINK/J-TRACE”或いは“RDI”が選択できる。Open Link は全機能バージョンの JTAG なので、性能向上する為、“RDI”の選択をお勧めする。



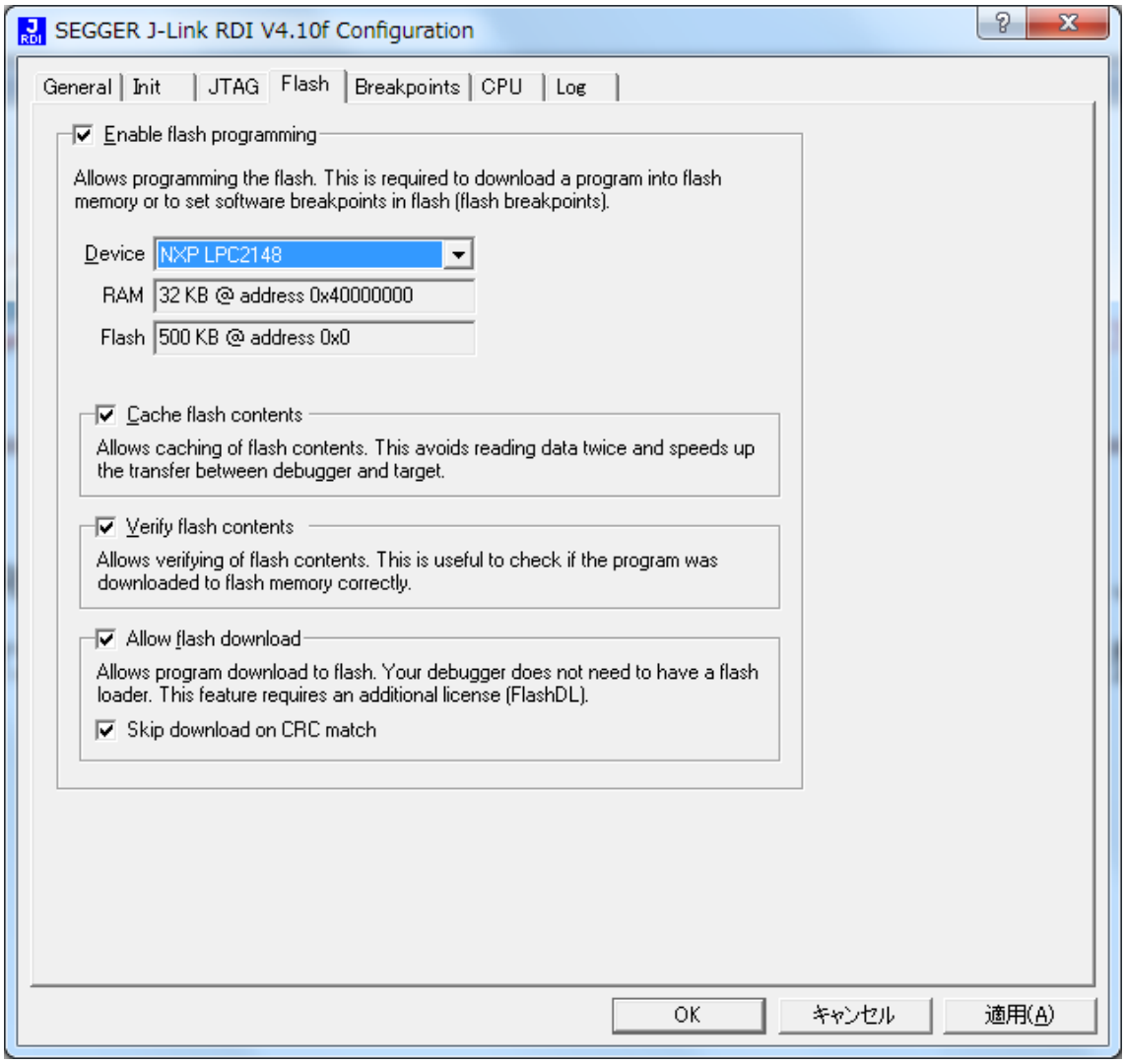
b) 「J-LINK/J-TRACE」を選択した場合は、他の設定は必要ない。「RDI」を選択した場合、JLINKRDI.DLL の場所を指定してください。



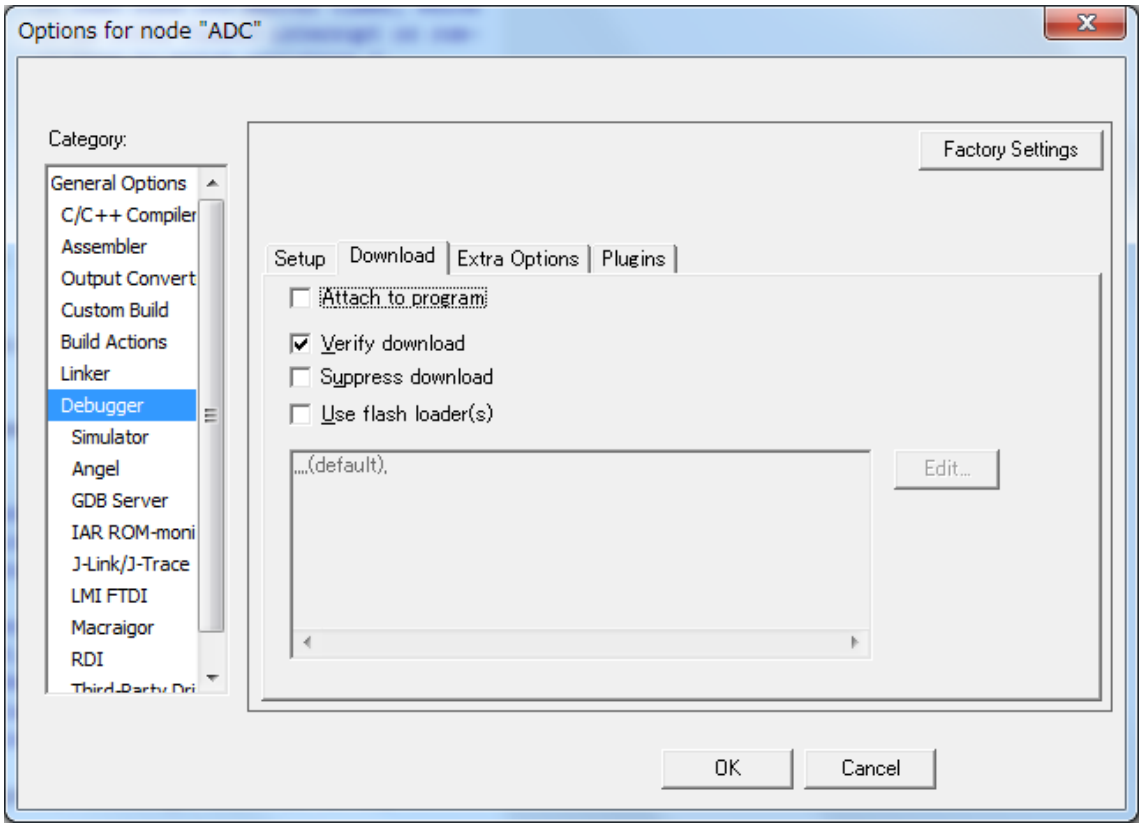
c）設定完了した後、RDI メニューが増える、下記の図の通りです。



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



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



3.4 OpenOCD での設定

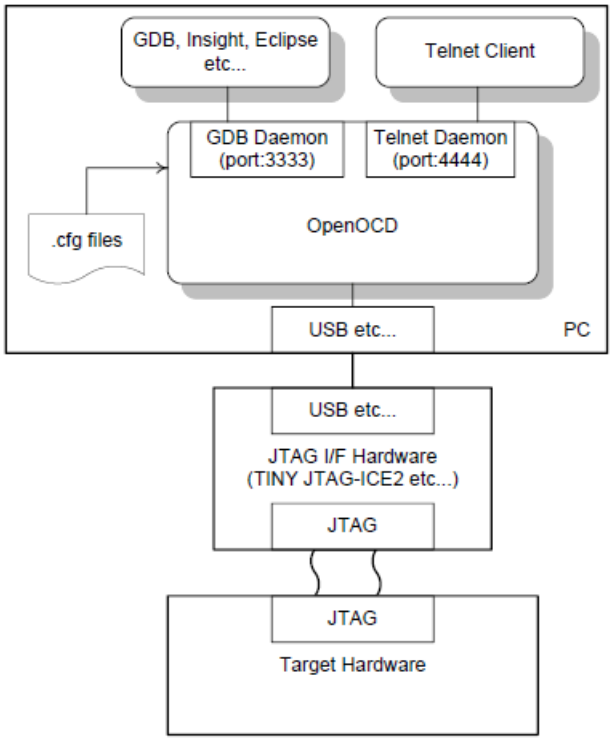
1) OpenOCD 概要

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

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

2) OpenOCD の構成イメージ

OpenOCD によって構成できるシステムの概略イメージです。



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

3）Open Link の使用手順

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

http://www.dragonwake.com/download/open-jtag/OpenJtag-Arm-All_manual.pdf

OpenOCD が動かすまでの大まかなネタは、

・OpenOCD.exe 単体では実行しても何事もなかったように終了してしまう。実行時に諸々記述したオプションを指定し、使用する USB 接続の JTAG I/F を接続した状態でしか使用できない。

・設定ファイルのある場所を検索させるには、-s オプションを付加して起動。

・何が起こっているのか詳細に知りたい時は-d 3 を指定すると、内部状況をつぶさにコンソールに吐き出す。そのログをファイルに保存するオプションもあり。

・shutdown コマンドを記述しない設定ファイルを指定して起動すると常駐する。また、設定ファイル内に telnet_port の指定があれば telnet のデーモンとなるので、指定されたポートへ telnet クライアントを使ってログインできる。使用できるコマンドは help で参照可。

・Windows 環境で動くと言っても、自らウィンドウを持つプログラムではない。コマンドプロンプト内でひっそり動いている。強制終了するならそのコマンドプロンプト内で[CTRL]+[C]を入力するか、コマンドプロンプトのタイトルバーの×を押すが程度。

・設定ファイルに gdb_port の設定があれば、gdb からそのポートへアクセスし gdb のコマンドを使用してターゲットの制御が可能。

・FLASH WRITER の CPU TYPE に OpenOCD が指定されていると、FLASH WRITER は OpenOCD の telnet デーモンにログインし、FLASH WRITER の Write Script File に従ってコマンドを送信できる。先の telnet クライアントにて手動でコマンドを叩くといった操作を自動化できるので、複数のコマンドを併用す る必要があるターゲットのフラッシュ ROM の書き込み処理に使用してるだけ。

・日本語は理解してくれないので、ファイル名等は半角英数字でなくてはならない。

・全く同じ回路であっても、ターゲット間をつなぐケーブルの長さ等の条件によって両者で同じ設定で同様に動くとは限らない。JTAG ラインのプルアップ/ダウン・コネクタの接触抵抗・ノイズの影響等を考慮しながら JTAG のスピードを調整する。速ければ良い物でもない。

・予め用意された設定ファイルはよく見かける主要なマイコン用の物しか用意されていない。似たような型番だからといって同じ設定ファイルが使えるわけでもないので、新しいデバイスに対応させるにはそれ相応のスキルが必要。

・チップ内蔵のフラッシュ ROM に書き込むだけなら OpenOCD を使うメリットは無く、

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

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

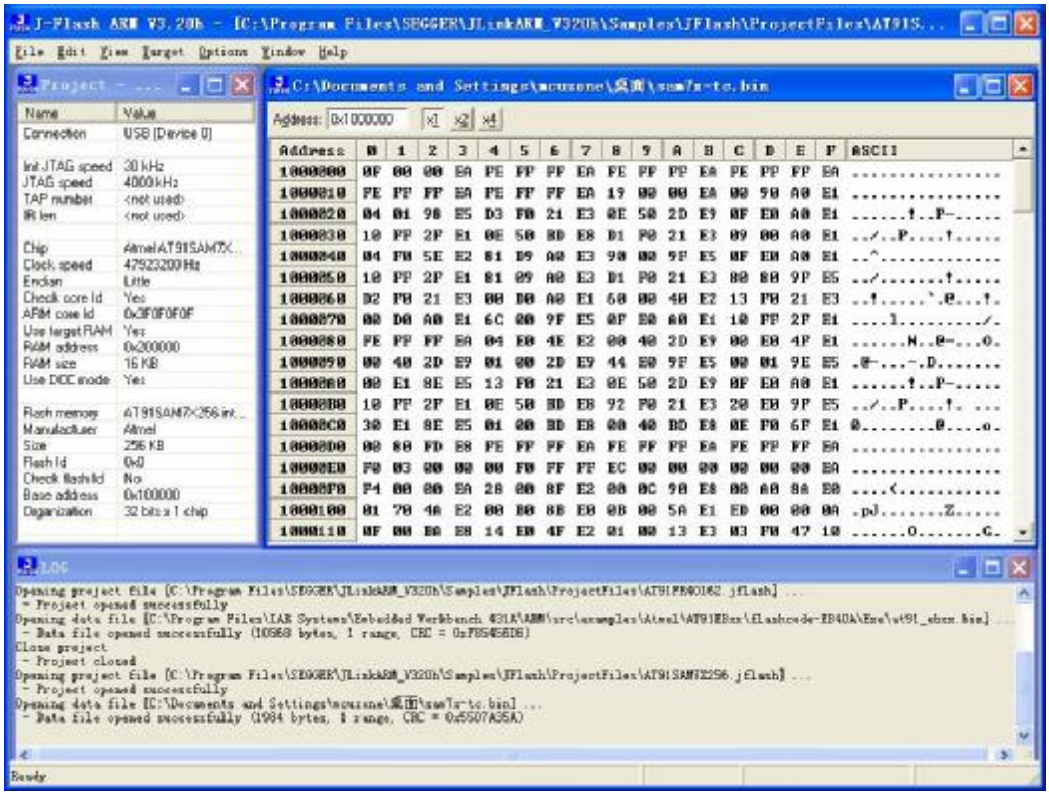
「OpenOCD」というプロジェクトのホームページは下記 URL であります。

http://openocd.sourceforge.net/

四、Open Link で J-Flash ARM 使用方法

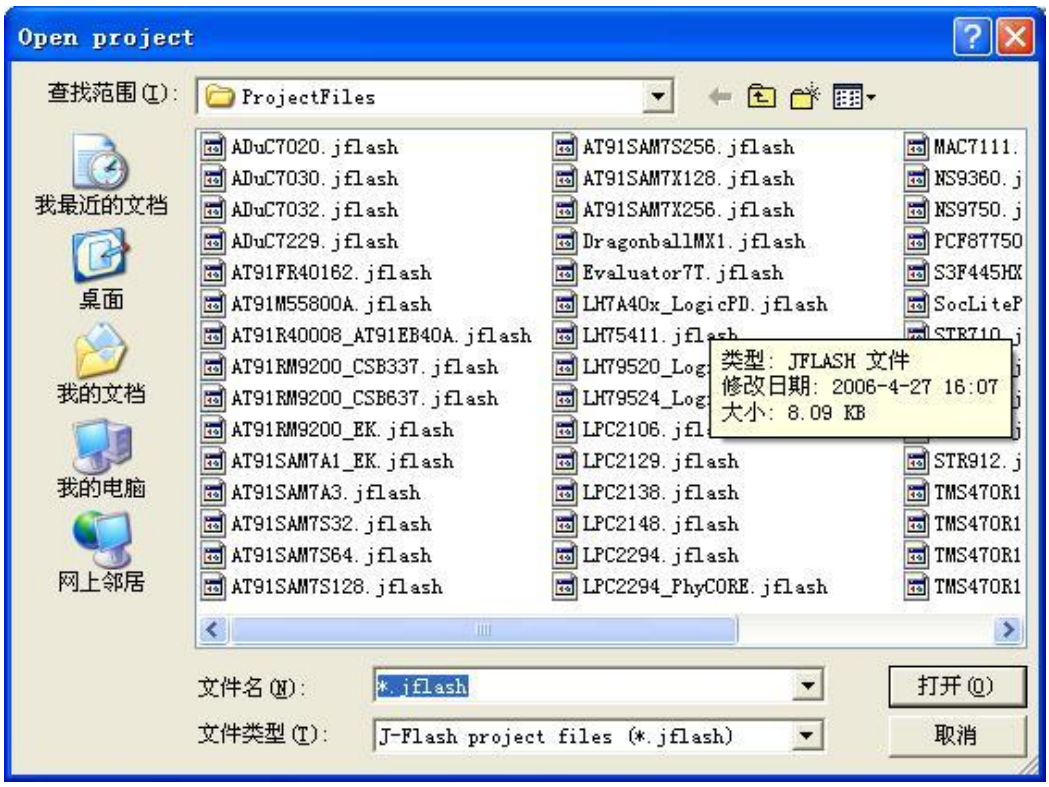
4.1 Open Link のドライバのインストール完成後、二つのショートカットアイコンが出てくる。一つが J-FLASH ARM である。このアプリケーションは FLASH のプログラミングで使う（J- FLASH ARM License 必要)。

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

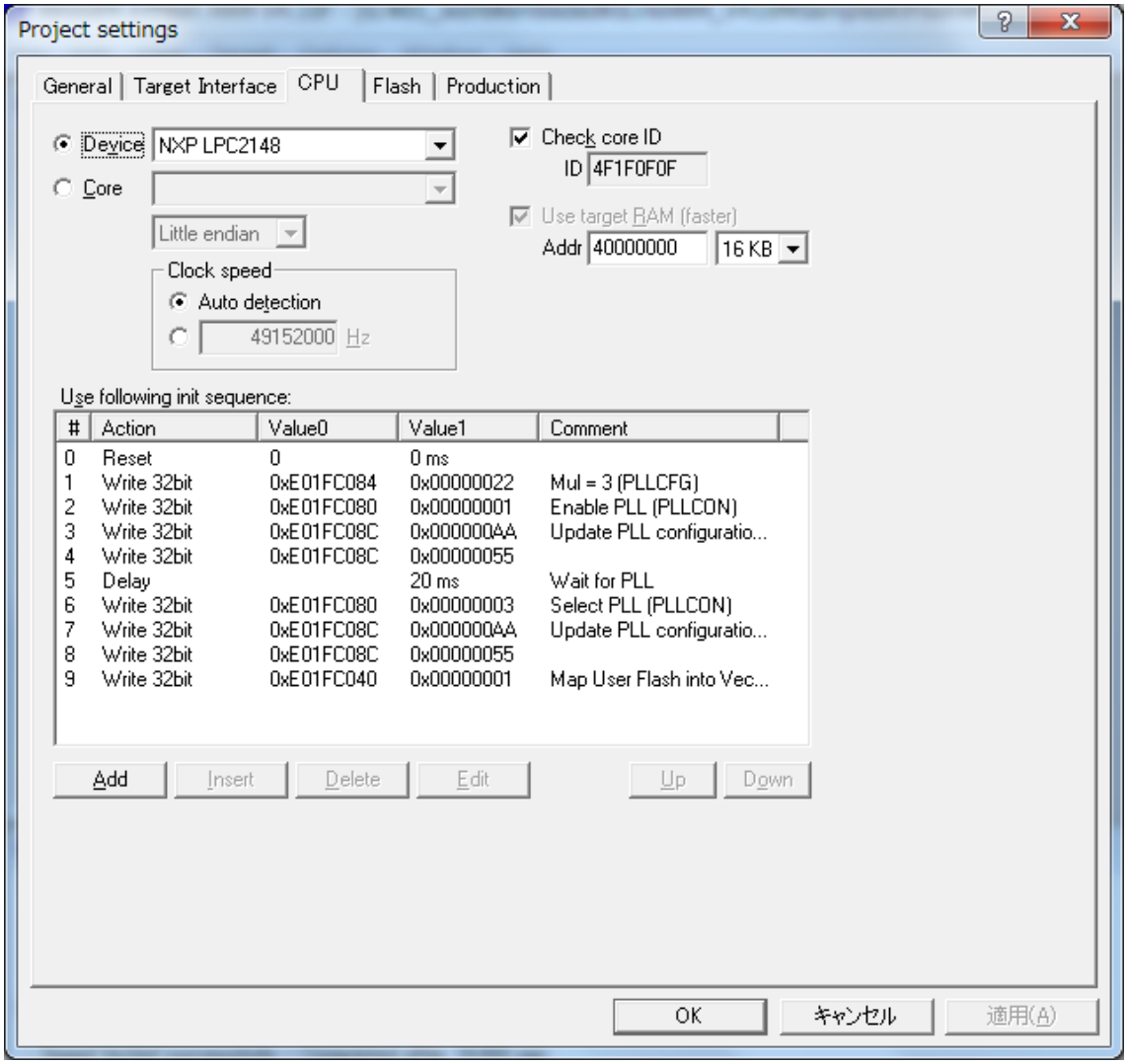


4.3「File」→「Open…」で書き込みターゲットファイルを選択する。.bin ファイル、.hex ファイルあるいは.mot ファイルどちでもできる。開始アドレスにご注意。

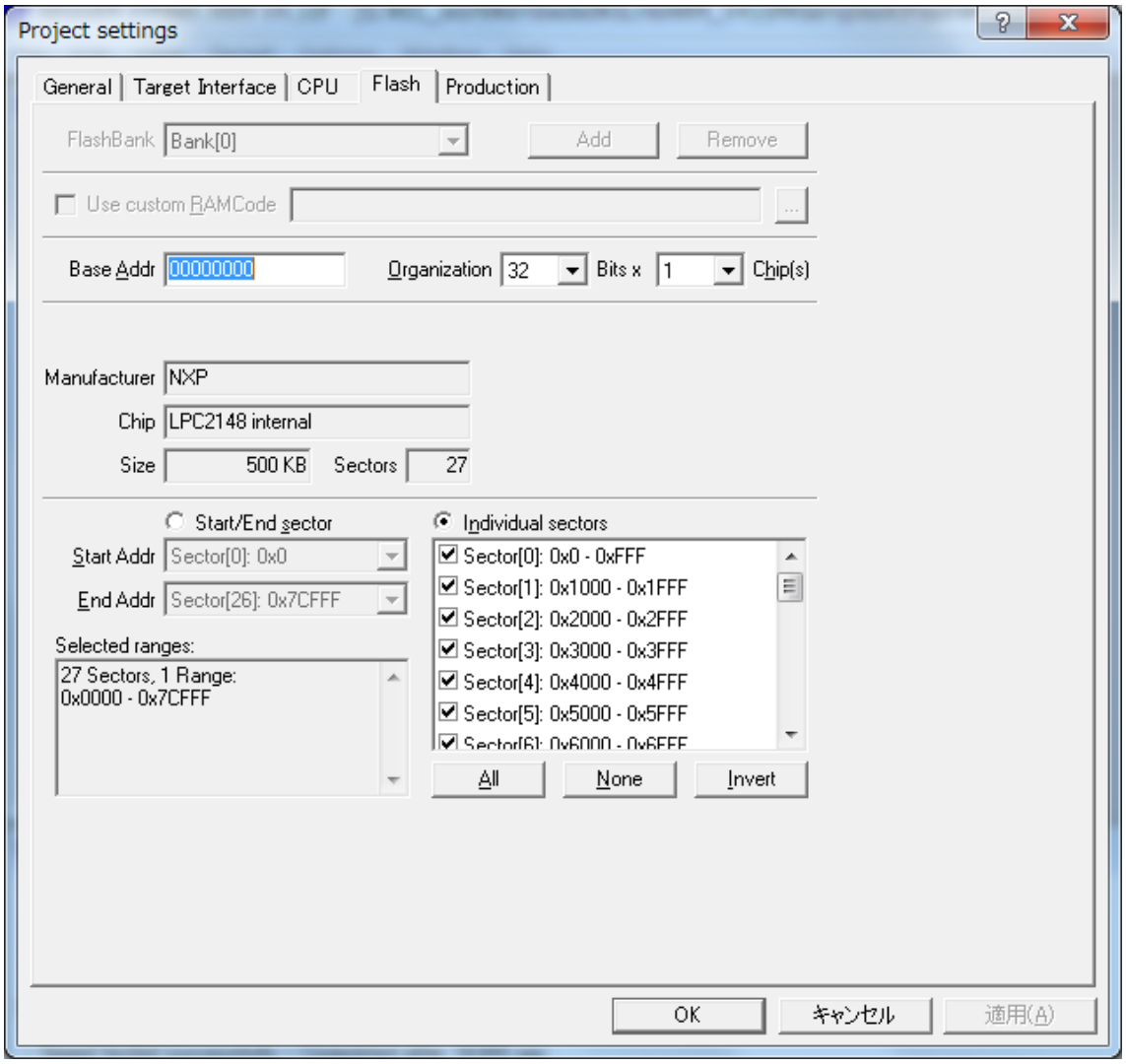
後は 「Options」→「Project settings」:



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

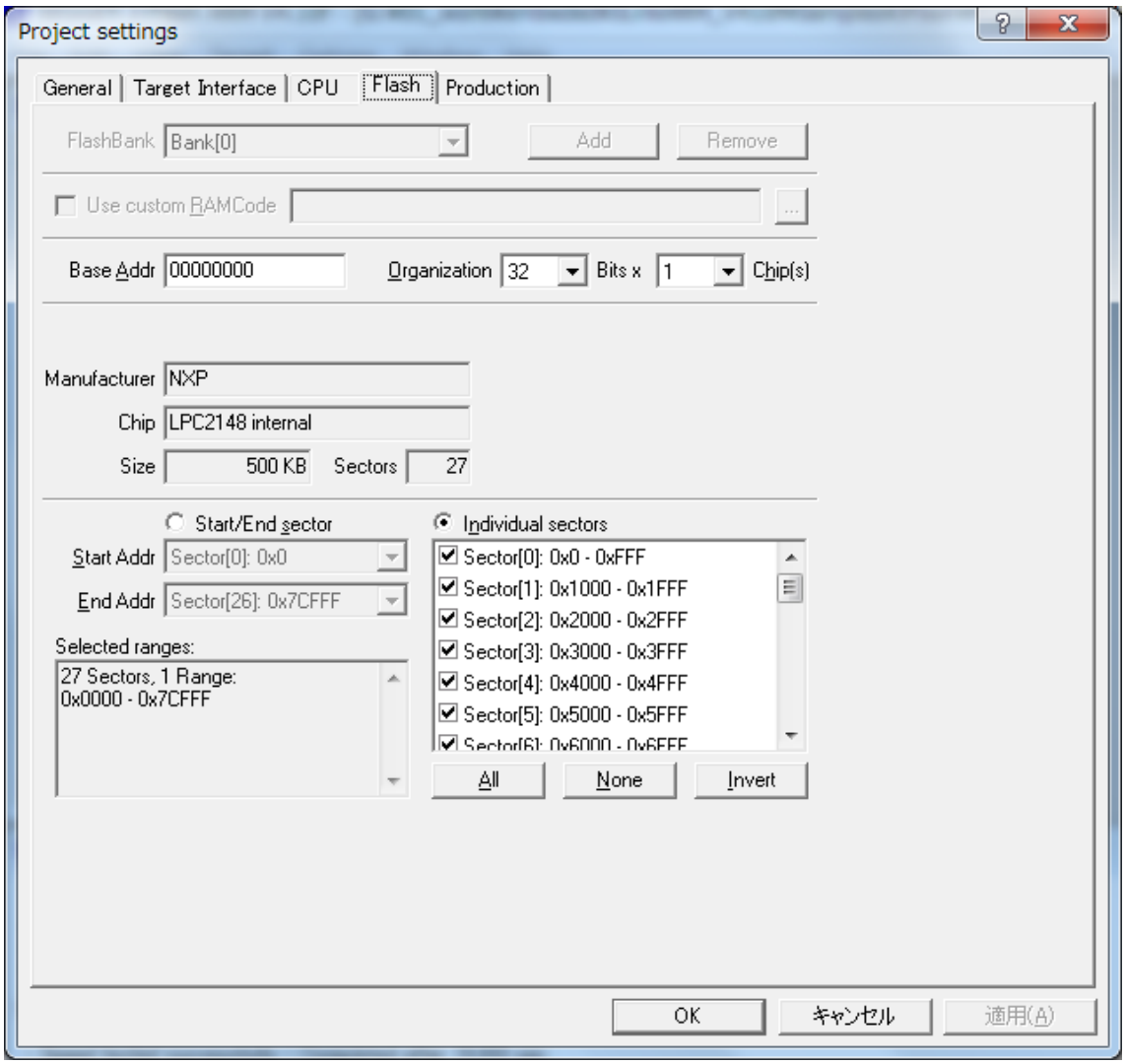


4.5 FLASH タブで，Opon project だったら、特に何も設定する必要はない。新しく作ったプロジェクトの場合は、次の設定が必要になる。この前の ARM タブで“Generic ARM7/ARM9”を選択した場合、FLASH タブで FLASH 型を選択できる。

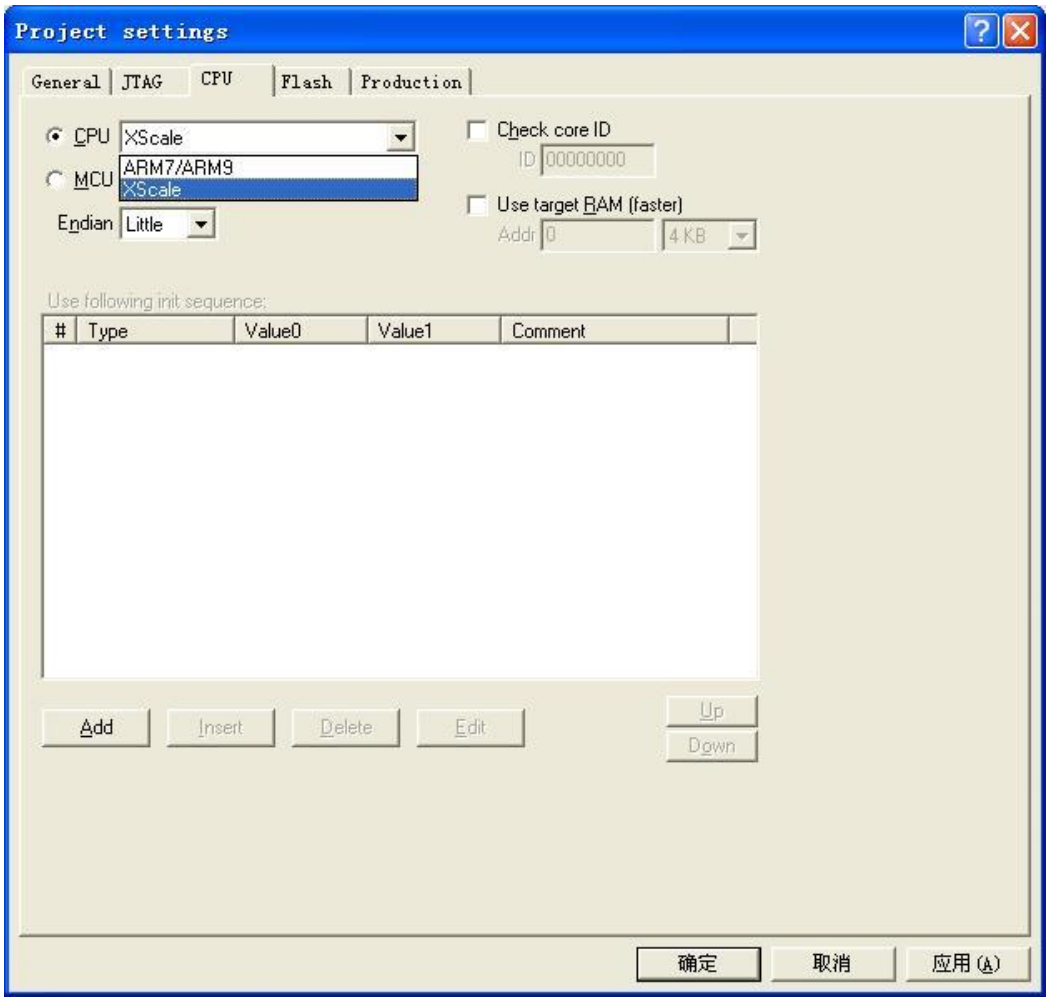


4.6 主なメーカーの FLASH をサポートする。また最新のデバイスをサポートするためにアップグレ ードを継続する。

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



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



五、ARM9 ボード(Mini2440 シリーズ)書込み手順

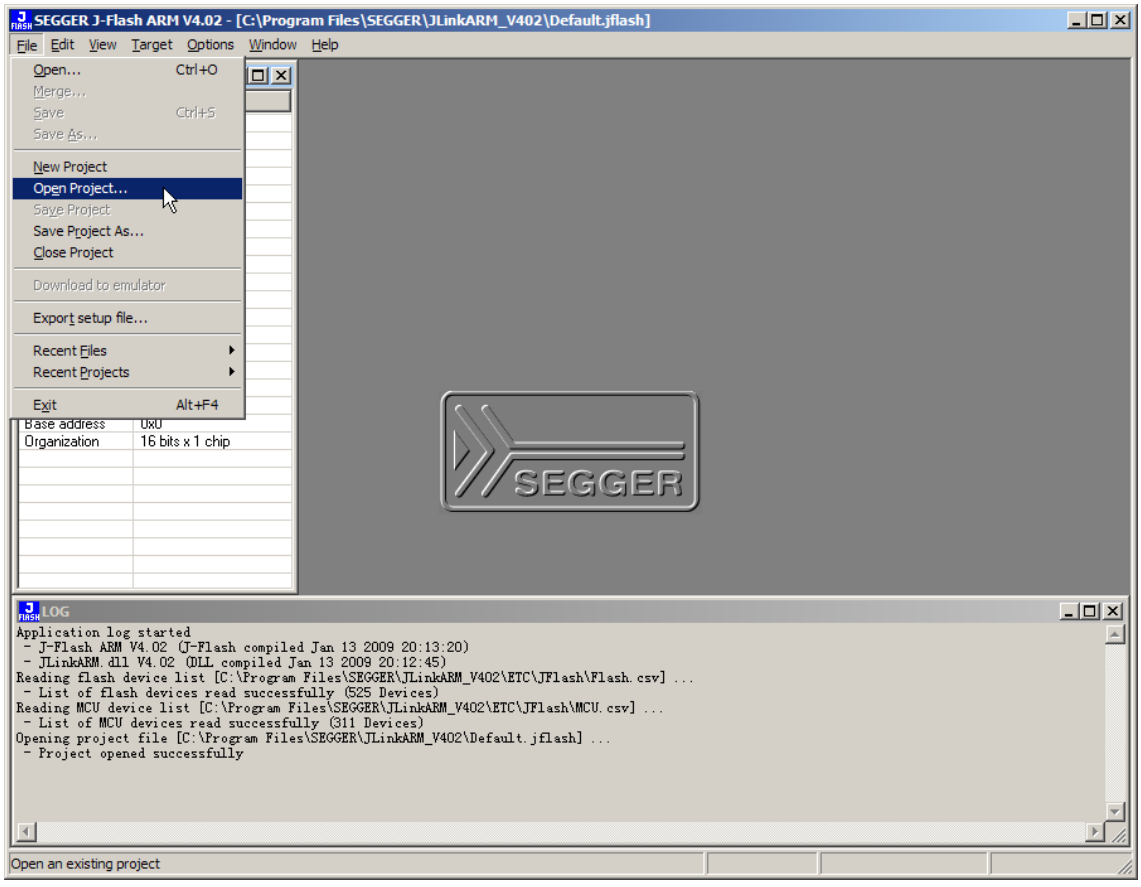
Note：「四、Open Link で J-Flash ARM 使用方法」を参照して行えますが、元々J-FLASH ARM プロジェクトの中、ARM9 を使えるものがありません。それ以外、Open Link では Nand Flash を直接書き込めないため、本章で説明します。

ARM9 ボードを書き込む用のリソースダウンロード URL：

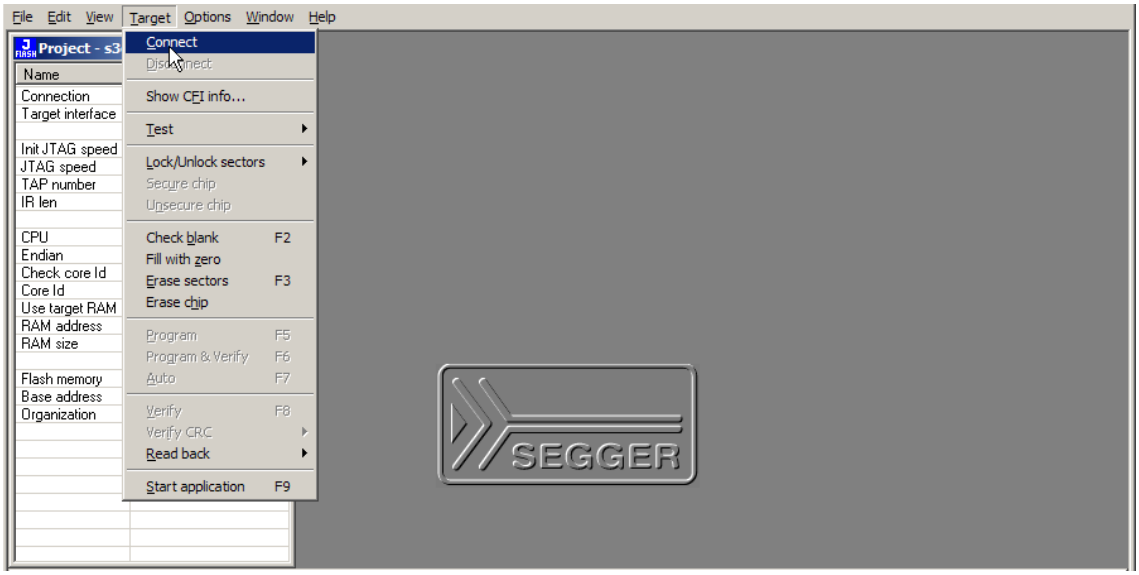
http://www.dragonwake.com/download/open-link/arm9-resource.zip

１．Nor　Flash に書き込む

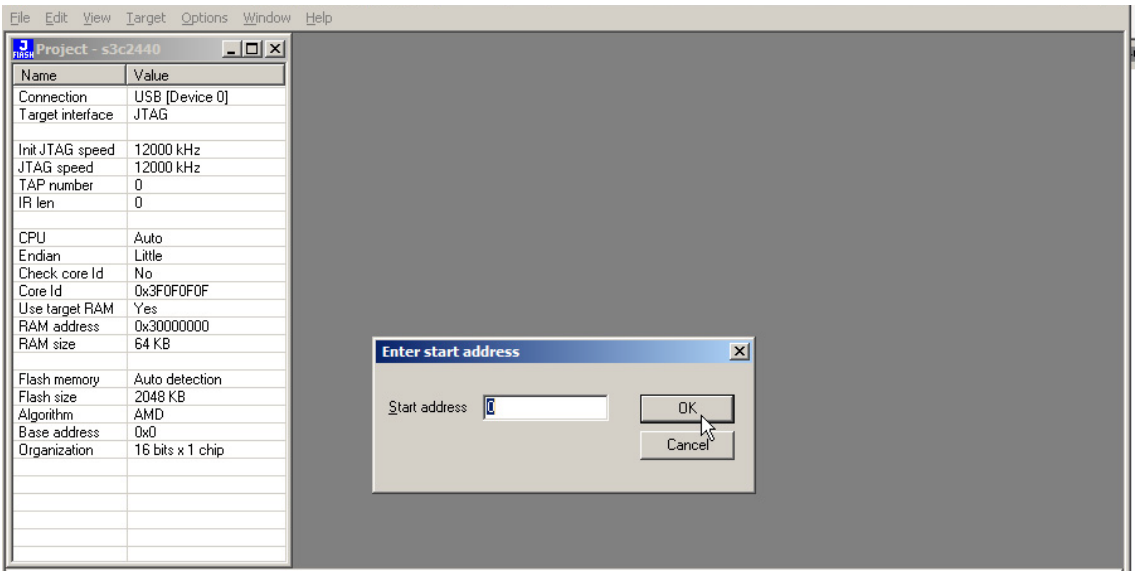
①　“File -> Open -> Open Project…”を選べ、「¥open-link¥s3c2440.jflash」を開いてください。



②“Target -> Connect”を選択



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



④“Target -> Auto”をクリックして書込みを自動に始める

２．Nand Flash に書き込む

Open-Link で Nand Flash に書き込めないため、まず、上記の手順で Nor Flash に「¥u-boot.bin」を書き込む必要です。

①シリアルケーブルで ARM9 ボードが PC と接続にし、ハイパーターミナルを設定

ハイパーターミナルの設定は「Mini2440 ボードマニュアル」の P41 をご参照ください。

ARM9 ボードに Nor Flash モードで電源を入れ、u-boot を起動します。

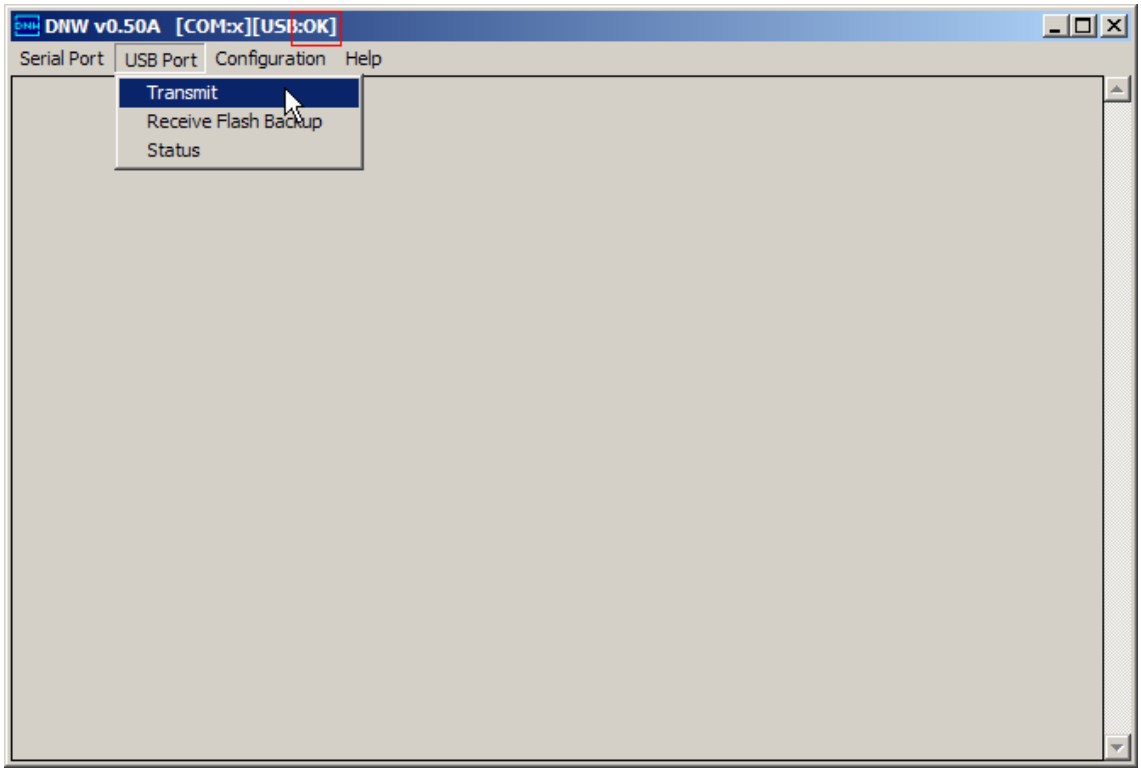
ハイパーターミナルに下記のようなメッセージが表示されます、0 を減るまでにスペースキーを押します。

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

②USB ケーブルで ARM ボードの USB デバイスポートが PC と接続

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

インストール方法は「Mini2440 ボードマニュアル」の P128 をご参照ください。実際のドライバファイルは上記解凍フォルダ「¥drivers¥dnw_win7_64bit」(Win7) 或いは「¥drivers¥dnw_xp」を使います。
インストール後、下記 USB 右側に「OK」が表示されれば、ドライバは正常にインストールされたことを明らかにします。



④u-boot のメニューに従い、ファイルを Nand Flash に書き込む事が出来ます。
例えば、「n」：“Download u-boot to Nand Flash”と言うメニューを選択する場合、USBケーブルでDNWツールを使ってファイルをNand Flashにダウンロードします。
※DNW ツールが「¥tools」にある
実は、弊社の Mini2440/Micro2440 ボードの場合、ボード自身は Nor Flash に既にブートローダを書き込んでいますので、そのブートロードを利用すれば、上記のようなダウンロード操作は必要ありません。弊社以外の ARM9 ボードで Open Link を使う際、上記の手順を参照してください。
※上記の手順は Mini2440 ボードで検証済み

六、OpenLink フォームウエア更新手順

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

１．更新用リソースダウンロード URL：

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

２．圧縮ファイル中の sam-ba_2.10.exe をインストールしてから PC を再起動

３．OpenLink のカーバーを取り出す

４．AT91SAM7S64 の既存フォームウエアを消す

① AT91SAM7S64（U1）の電源を切断（USB ケーブルを抜く）

② ERASE の二つ PIN をショート

③ USB ケーブルで OpenLink を PC と接続

④ 60 秒を待って USB ケーブルを抜く

⑤ ERASE のショートを戻す

５．SAM-BA Boot フォームウエア更新の準備

① USB ケーブルを抜く（AT91SAM7S64 電源を切断）

② TST の２つ PIN をショート

③ USB ケーブルで PC と接続

④ 60 秒を待って USB ケーブルを抜く

⑤ TST のショートを戻す

６．USB ドライバーインストール

① USB ケーブルで PC と接続

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

＊Windows XP の場合、デフォールトのままインストールできます。

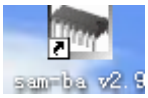
＊Windows 7 の場合、OS が自動に別のデバイスドライバ（正しくないドライバー）をインストールします。正しくインストールするため、手動インストール必要です。（32bit の Win7 のみをサポート）

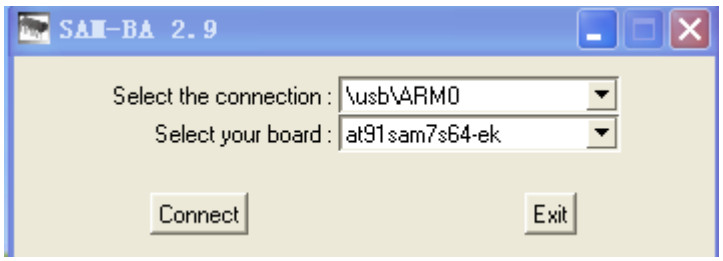
コントロール パネル—>ハードウェアとサウンド—>デバイスとプリンター—>デバイスを右クリック—>ハードウエア—>プロパティ—>ドライバー—>ドライバーの更新—>ドライバーソフトウエアを手動で検索してインストール—>コンピュータ上のディスクドライバーの一覧から選択します—>ディスク使用

参照パスは sam-ba_2.10.exe のインストール先となります。

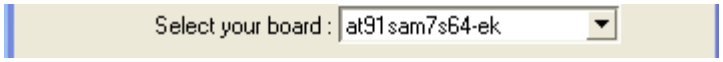
例：「G:¥02_tools¥embedded¥SAM-BA v2.10¥drv」

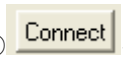
７．SAM－BA ツールを起動してからフォームウエアを更新



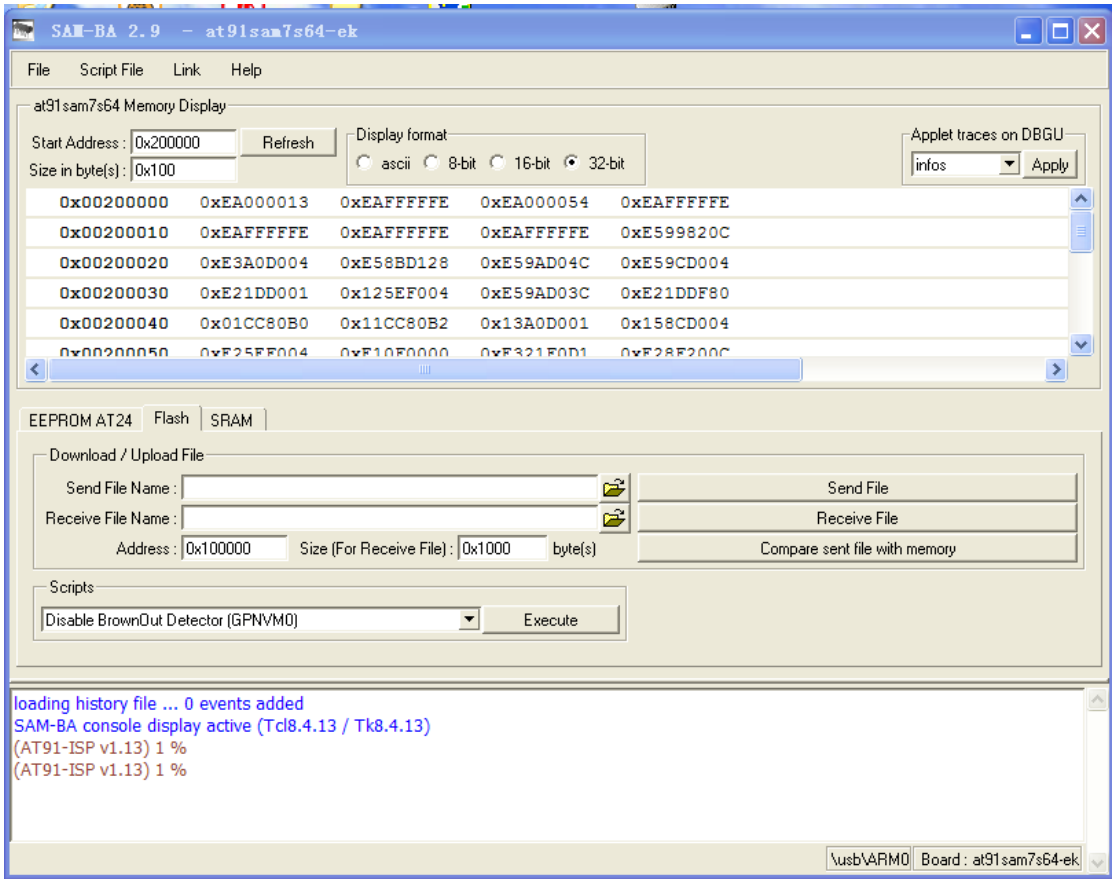


ボードの種類は at91sam7s64-ek です。



③ 

をクリック



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

更新完了後、OpenLink は普通通り使用できます。