

Eclipse & Blackfin Toolchain(gcc)

インストール&チュートリアル

第7版

ご注意：このドキュメントは旧版となりました。

下記 URL から最新版を参照ください。

<http://kaneko-sys.co.jp/support/>

金子システム株式会社

ご注意

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、当社ホームページを通じて公開される情報を参照ください。
2. 当社から提供する情報の正確性と信頼性には万全を尽くしていますが、誤りがないことを保証するものではありません。当社はその使用に対する責任を一切負いません。その使用によって第三者の特許権、著作権その他知的財産が侵害された場合でも、同様に責任を負いません。
3. 本資料は、当社の書面による事前の明示同意がない限り、いかなる形式でも複製できません。

目次

1	はじめに.....	4
2	必要なハードウェア.....	4
3	ソフトウェアインストール.....	5
3.1	必要なソフトウェアパッケージ.....	5
3.2	Blackfin Toolchain のインストール.....	6
3.3	gnICE+ドライバのインストール.....	7
3.4	Blackfin Toolchain のリンカスクリプトの修正.....	7
3.5	Eclipse のインストール.....	8
3.6	Eclipse の起動.....	9
3.7	Blackfin Plugins のインストール.....	11
4	Blackfin アプリケーションの作成.....	15
4.1	プロジェクトの作成.....	16
4.2	gdb によるデバッグ.....	22
5	Silicon Revision について.....	28
6	更新履歴.....	29

1 はじめに

本書は、Eclipse と GNU Toolchain for the Blackfin Processor(以下 Blackfin Toolchain)を使ったアプリケーションの開発チュートリアルです。Windows を対象に、Blackfin Toolchain と Eclipse のインストール及び、簡単なサンプルプログラムの構築、デバッグまでの手順を解説します。

2 必要なハードウェア

ここでは、以下のハードウェアを使用します。

表 1 ハードウェア一覧

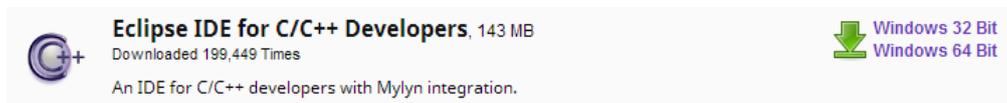
製品名	写真
<p>金子システム ACB-BF592</p> <p>以下よりお求めいただけます。 http://kaneko-sys.co.jp/shop/</p>	 <p>The image shows a blue printed circuit board (PCB) development kit. It features a central integrated circuit (IC) labeled 'U1'. Other components include resistors (R1, R16), capacitors (C1, C2), and a USB connector (U3). The board is populated with various electronic components and has a header strip along the bottom edge. The text 'KANEKO SYSTEM' and 'ACB-BF592-A' are visible on the board.</p>
<p>gnICE+</p> <p>Digikey にて購入可能です Digikey 品番 : 190-2255-ND</p>	 <p>The image shows a small, rectangular black component with a green header strip on the left side. The text on the component reads 'BLACKfin gnICE+', 'Hi-Speed USB JTAG ICE V1.1', and 'Enhanced IO-Voltage Compliance'. There is also a small logo and a website URL 'http://blackfin.uconn.org'.</p>

3 ソフトウェアインストール

3.1 必要なソフトウェアパッケージ

必要なパッケージは、以下の通りです。あらかじめ下記のファイルをダウンロードしてください。

- Blackfin Toolchain [blackfin-toolchain-win32-2014R1.exe](http://sourceforge.net/projects/adi-toolchain/files/2014R1/2014R1-RC1/blackfin-toolchain-win32-2014R1.exe/download)
<http://sourceforge.net/projects/adi-toolchain/files/2014R1/2014R1-RC1/blackfin-toolchain-win32-2014R1.exe/download>
- Eclipse Kepler RC2 Packages [Eclipse IDE for C/C++ Developers](https://www.eclipse.org/downloads/packages/release/kepler/sr2)
<https://www.eclipse.org/downloads/packages/release/kepler/sr2>



※Windows 32bit 版と 64bit 版があります。ご使用の環境によってダウンロードするファイルを選択してください。

※Eclipse Helios は、後述する Blackfin Plugins がサポート対象外になったようで、動作しません。

- Java Runtime
<http://java.com/ja/>

Eclipse は Java で作成されているため、動作には Java Runtime が必要です。ライセンスに同意のうえ、手順に沿ってインストールしてください。なお、java のインストール手順はここでは解説しませんので、詳しくは java のサイトを参照してください。

3.2 Blackfin Toolchain のインストール

blackfin-toolchain-win32-2014R1.exe をダブルクリックし、インストールしてください。



図 1 Blackfin Toolchain インストーラアイコン

インストールの条件を変更する必要はありません。

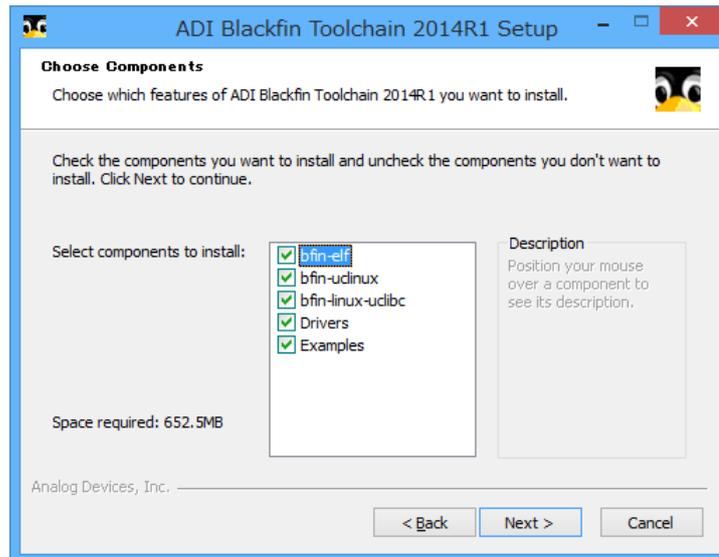


図 2 Blackfin Toolchain インストール画面（コンポーネント選択）

インストールが無事終われば、完了です。

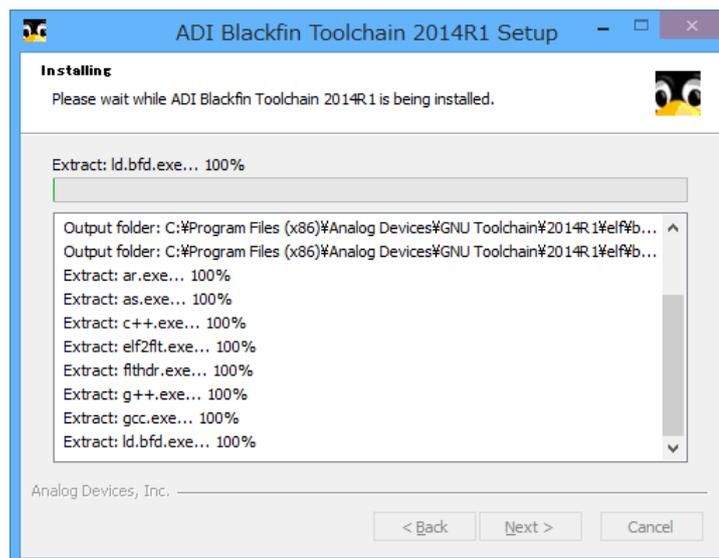


図 3 Blackfin Toolchain インストール画面（インストール中）

3.3 gnICE+ドライバのインストール

gnICE+を USB ケーブルで接続し、必要なドライバをインストールしてください。ドライバファイルは、標準構成で以下のフォルダに格納されています。

【32bit 版 Windows】

C:\Program Files\Analog Devices\GNU Toolchain\2014R1\gnICE-drivers

【64bit 版 Windows】

C:\Program Files (x86)\Analog Devices\GNU Toolchain\2014R1\gnICE-drivers

3.4 Blackfin Toolchain のリンクスクリプトの修正

ADSP-BF592 用のソースファイルをビルドする場合、Blackfin Toolchain のリンクスクリプトの設定ミスにより、オブジェクトのコードサイズが 16K バイト以下 (※1)でないと、正しくリンクできないことが分かっています。以下の作業でリンクスクリプトを修正し、正しいメモリマップに変更します。

この不具合は Blackfin Toolchain バージョン 2014R1 以前で確認しており、今後のバージョンでは修正されている可能性があります。

エクスプローラから、以下のフォルダを開き、bf592.ld ファイルをテキストエディタで編集します。

【32bit Windows 版】

C:\Program Files\Analog Devices\GNU Toolchain\2014R1\elf\bfm-elf\lib

【64bit Windows 版】

C:\Program Files (x86)\Analog Devices\GNU Toolchain\2014R1\elf\bfm-elf\lib

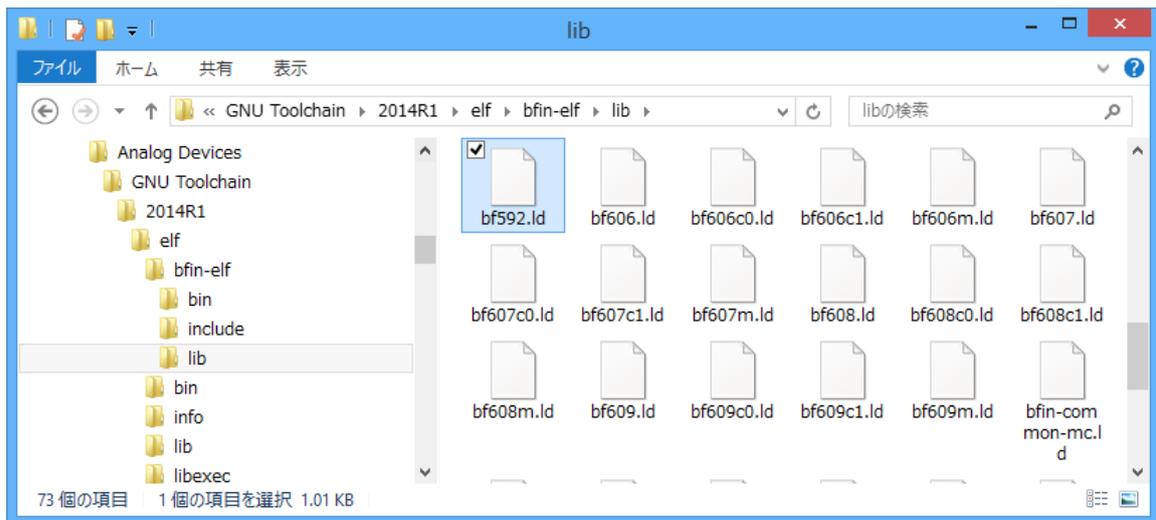


図 4 bf592.ld ファイルの格納場所

※1 ADSP-BF592 の L1 インストラクション RAM は 32K バイトでキャッシュなし

```

bf592.ld
1  /*
2  * The default linker script for standalone executables running on
3  * ADSP-BF592 processor.
4  *
5  * Copyright (C) 2007-2010 Analog Devices, Inc.
6  *
7  * The authors hereby grant permission to use, copy, modify, distribut
8  * and license this software and its documentation for any purpose, pr
9  * that existing copyright notices are retained in all copies and that
10 * notice is included verbatim in any distributions. No written agreee
11 * license, or royalty fee is required for any of the authorized uses.
12 * Modifications to this software may be copyrighted by their authors.
13 * and need not follow the licensing terms described here, provided th
14 * the new terms are clearly indicated on the first page of each file
15 * they apply.
16 */
17
18 MEMORY
19 {
20     MEM_L1_CODE : ORIGIN = 0xFFA00000, LENGTH = 0x8000
21     /* MEM_L1_CODE_CACHE : ORIGIN = 0xFFA04000, LENGTH = 0x4000 */
22     MEM_L1_SCRATCH : ORIGIN = 0xFFB00000, LENGTH = 0x1000
23     MEM_L1_DATA_A : ORIGIN = 0xFF800000, LENGTH = 0x8000
24
25     MEM_L2 : ORIGIN = 0xFEB00000, LENGTH = 0x0
26 }

```

図 5 bf592.ld ファイル修正箇所

bf592.ld ファイルの 20 行目、LENGTH の値を 0x8000 に修正します。

```
MEM_L1_CODE : ORIGIN = 0xFFA00000, LENGTH = 0x8000
```

bf592.ld ファイルの 21 行目、キャッシュの設定をコメントにします。

```
/* MEM_L1_CODE_CACHE : ORIGIN = 0xFFA04000, LENGTH = 0x4000 */
```

編集が完了したら、ファイルを保存して、エディタを閉じてください。

3.5 Eclipse のインストール

Eclipse はインストーラーがなく、zip ファイルを解凍すればインストールが完了します。作業するワークフォルダに展開してください。

ここでは、例として” C:\Analog Devices”に解凍します。

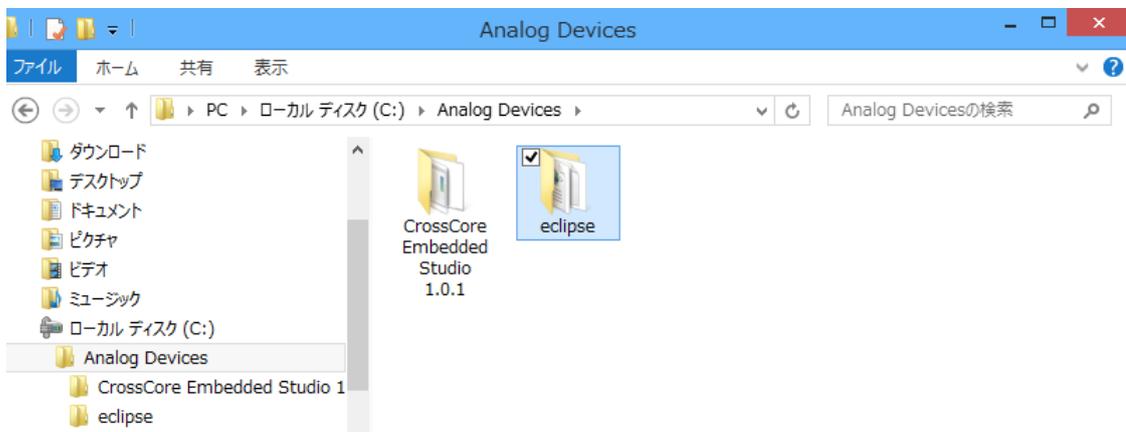


図 6 Eclipse の解凍先フォルダ例

3.6 Eclipse の起動

初回のみ、Eclipse の設定を行う必要があります。
最初に、解凍した eclipse フォルダにある、eclipse.exe を起動します。



図 7 Eclipse.exe アイコン

起動すると、以下のダイアログが表示されます。このフォルダは、ソースやプロジェクトを保存する作業フォルダ（ワークスペース）になりますので、ご使用の環境によって決めてください。以下では、X:\¥Blackfin フォルダを例に進めます。

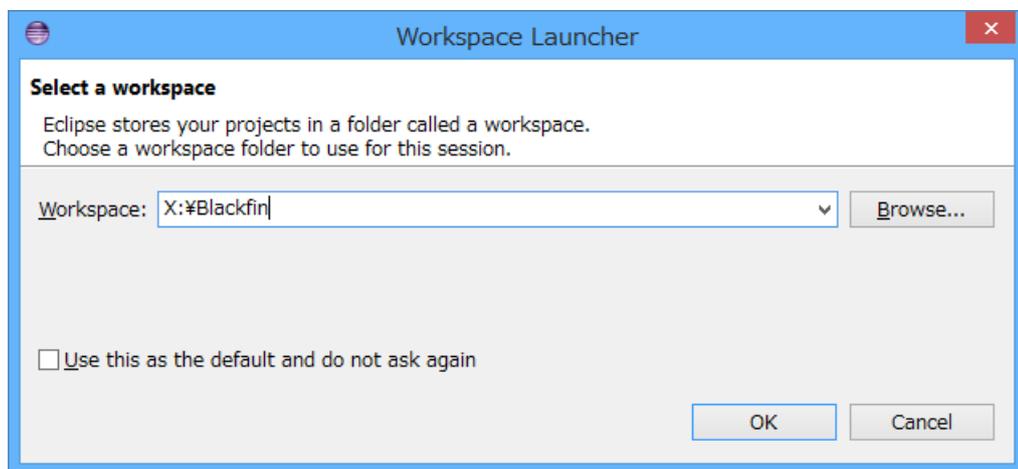


図 8 Eclipse 作業フォルダ指定

作業フォルダの指定が終わると、以下の画面が表示されます。

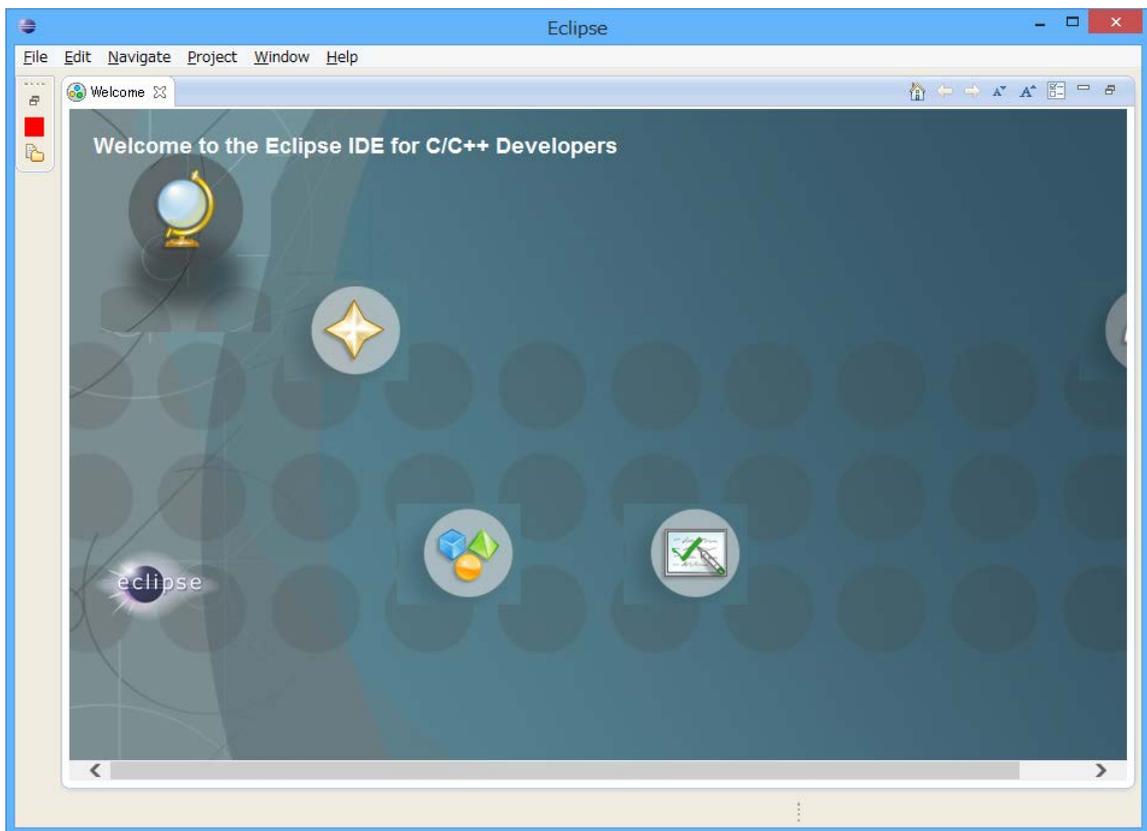


図 9 Eclipse 起動画面

3.7 Blackfin Plugins のインストール

Blackfin Plugins とは、Eclipse で Blackfin Toolchain を便利に使うための拡張機能の事です。このプラグインをインストールしておけば、コンパイラやデバッガ等の設定を手動で設定する必要がなくなり、プログラムと関係のない作業から解放され、簡単にプログラムの作成ができます。

メニューの Help → Install New Software を選択します。

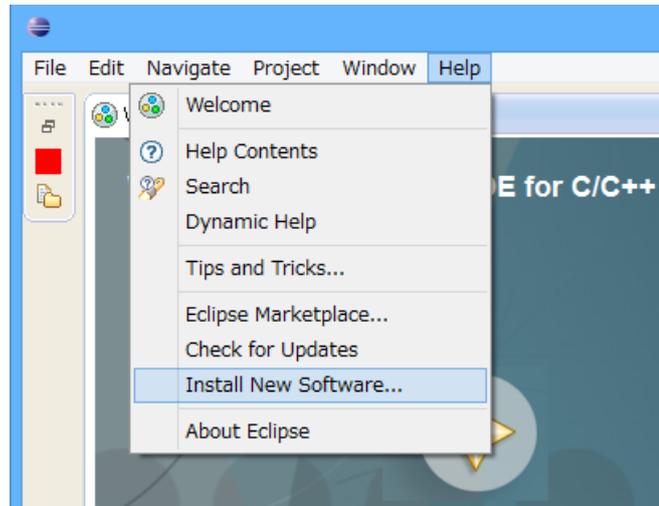


図 10 Eclipse Help メニュー

Work with のリストボックスに、

<http://download.eclipse.org/tools/cdt/releases/kepler/>

と入力し、CDT Optional Features - C/C++ GDB Hardware Debugging にチェックを入れ、Next ボタンをクリックし、ライセンス条項を確認して Finish ボタンをクリックします。インストール後、Eclipse を再起動するか問われますので、OK ボタンをクリックして、再起動してください。

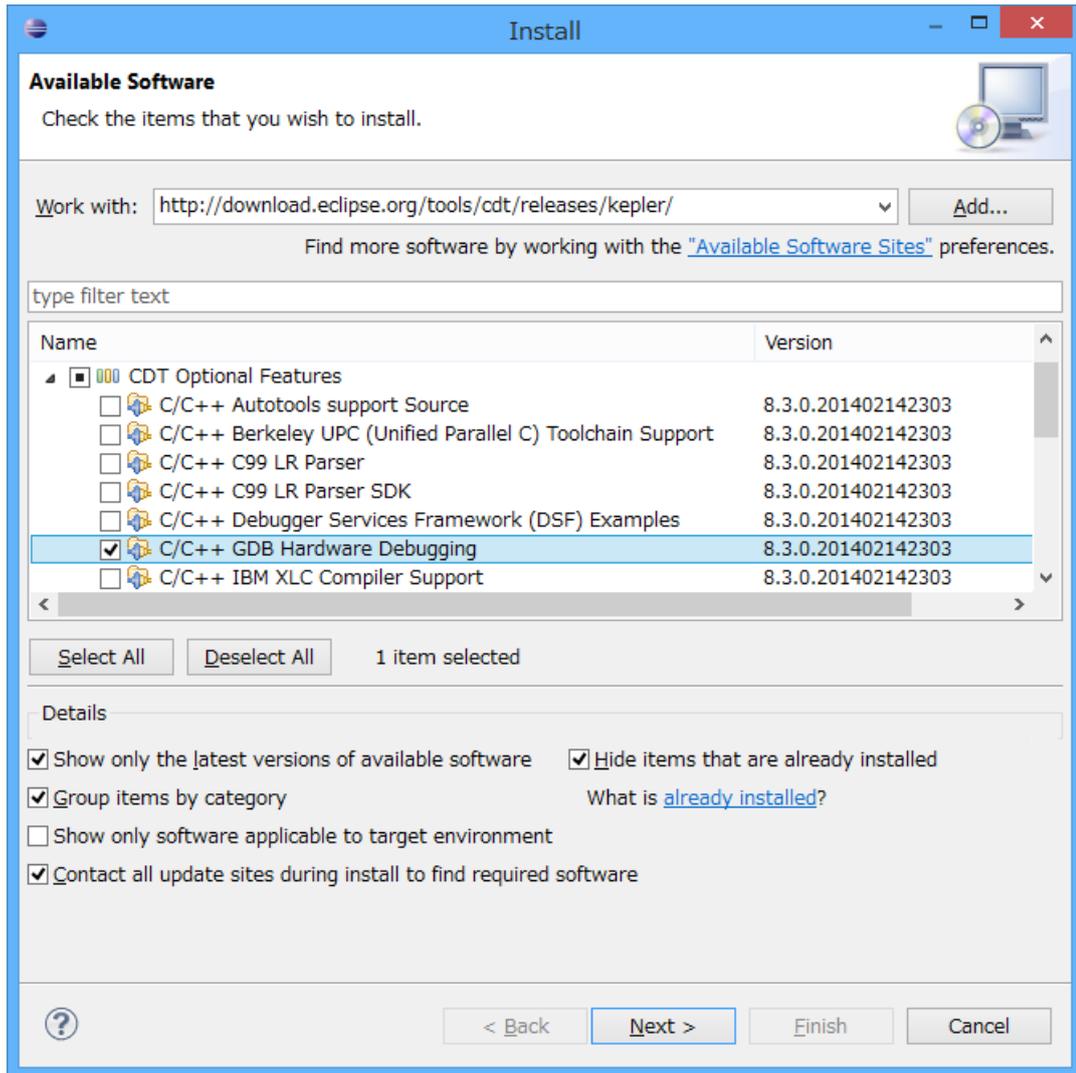


図 11 Eclipse Install ダイアログ

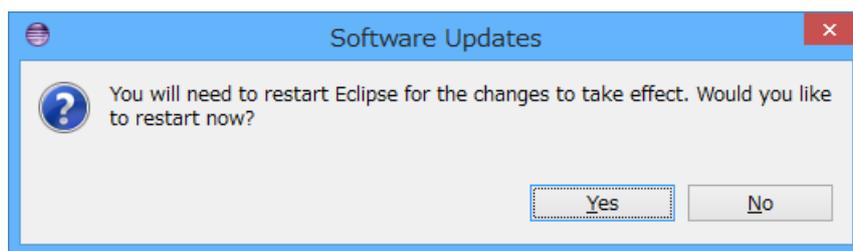


図 12 再起動確認ダイアログ

再度、メニューの Help → Install New Software を選択します。

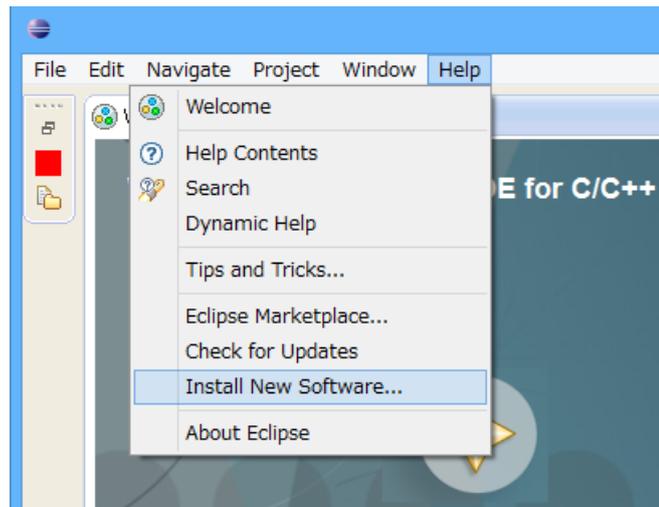


図 13 Eclipse Help メニュー

Install ダイアログの Add ボタンを押して、以下のように入力します。

Name: Blackfin Linux

Location: https://sourceforge.net/projects/adi-toolchain/files/eclipse/update_site/

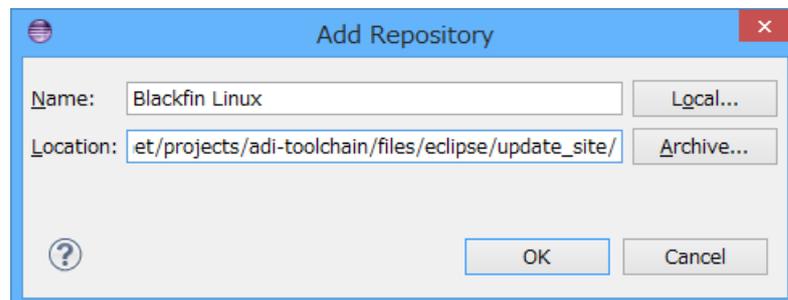


図 14 Add Repository

正しくデータが取得できると、以下のようにリストが更新されます。Blackfin Debug と Blackfin GNU Toolchain にチェックを入れて、Next ボタンをクリックし、ライセンス条項を確認した後、Finish ボタンをクリックします。インストール後、Eclipse を終了し、再起動してください。

※インストール途中で Security Warning が表示される場合、OK を押して続行してください。

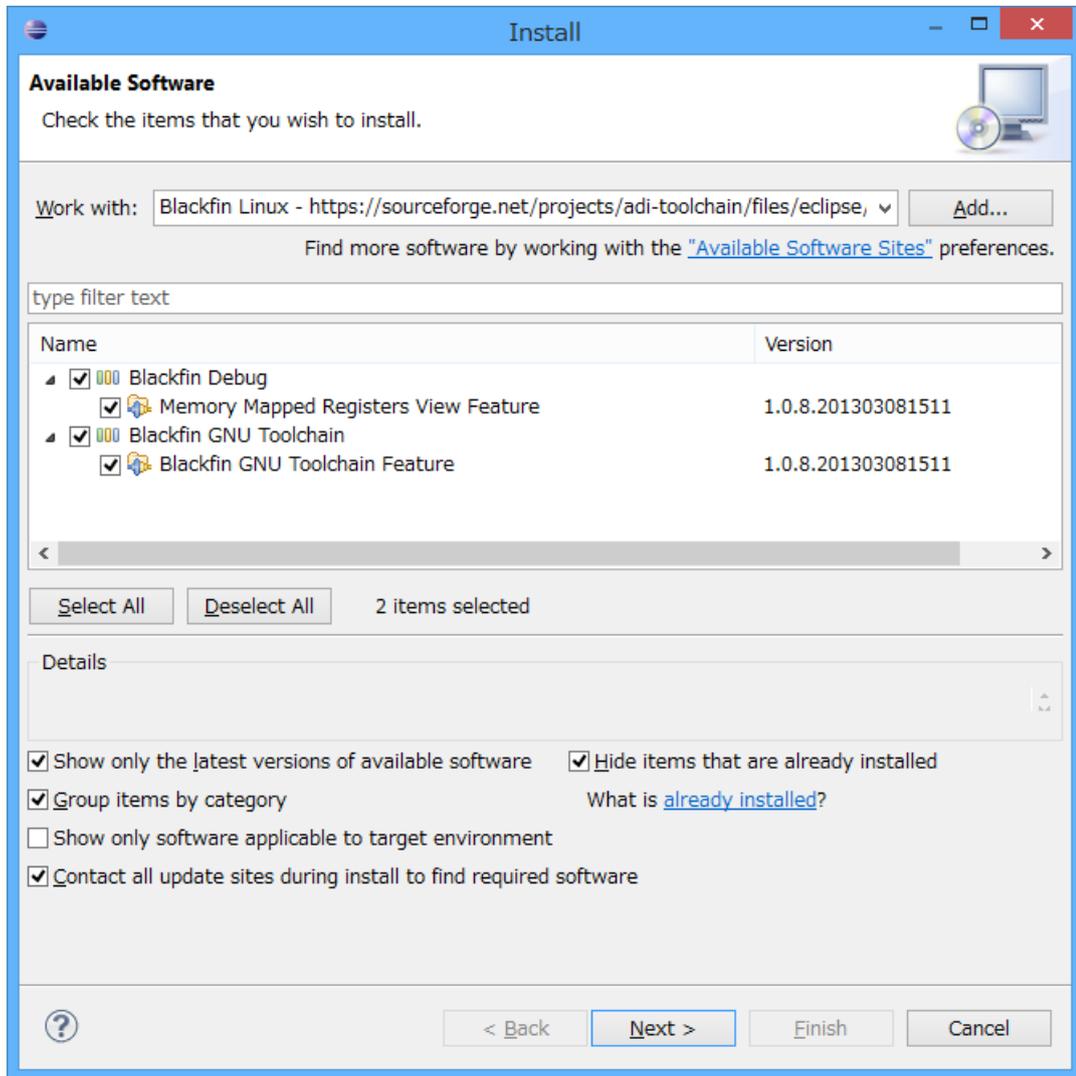


図 15 Eclipse Install ダイアログ

以上でインストールは完了です。

4 Blackfin アプリケーションの作成

ここでは、Blackfin アプリケーションの作成からデバッグまでの手順を解説します。

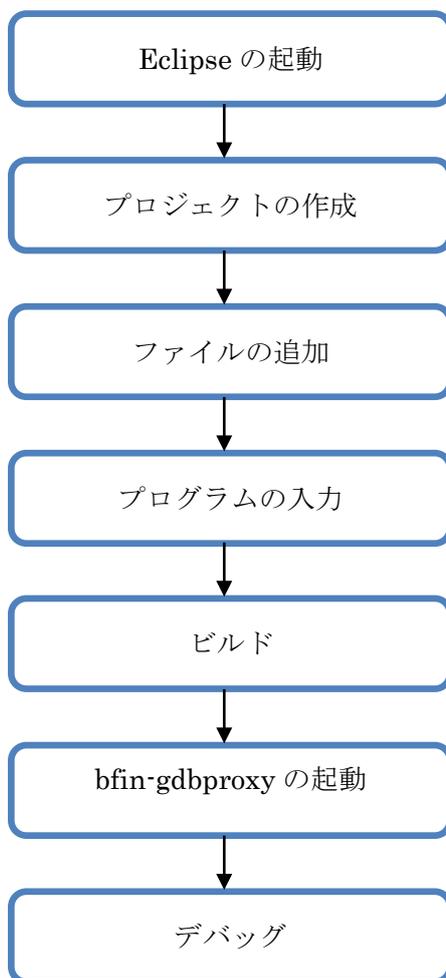


図 16 Eclipse での開発の流れ

4.1 プロジェクトの作成

プロジェクトを作成します。

Eclipse を起動し、New → C Project を選択します。

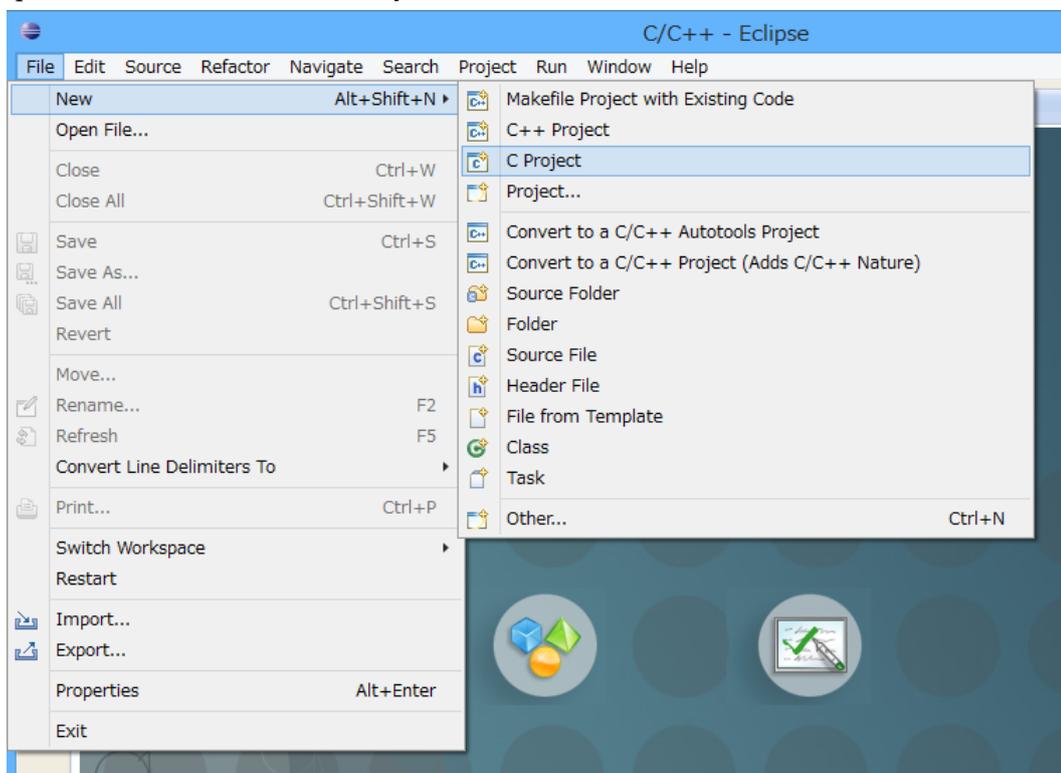


図 17 Eclipse New C Project

プロジェクト名に、led_blink と入力します（この名前はアプリケーションによって変更してください）。Project Type には Empty Project、Toolchains には Blackfin Bare Metal (bfin-elf) を選択し、Next ボタンをクリックします。

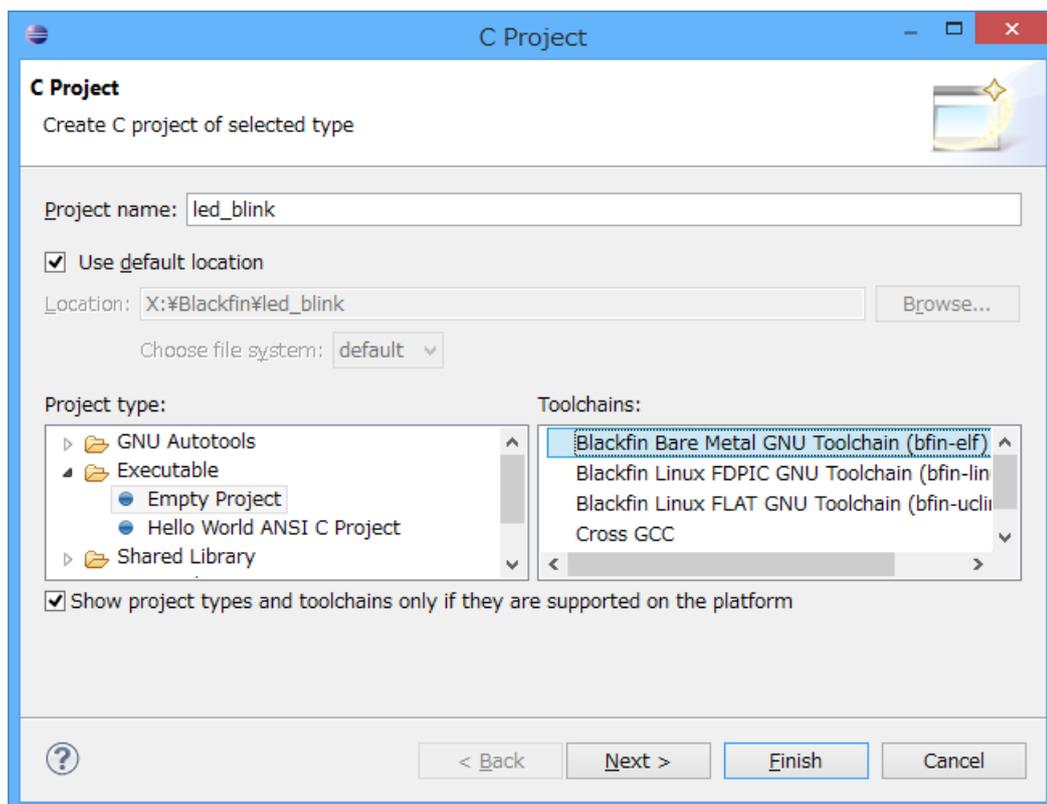


図 18 C Project ダイアログ

Select Configurations 画面では、そのまま Next ボタンをクリックします。

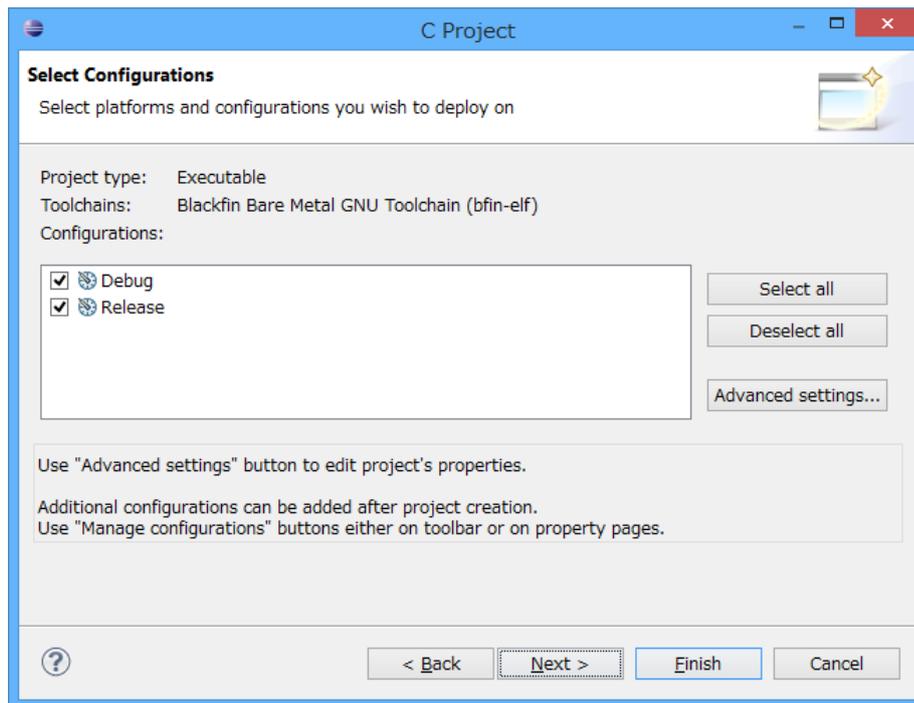


図 19 C Project ダイアログ

Select Processor 画面では、使用する Blackfin プロセッサを選択します。ここでは BF592 を選択します（プロジェクト作成後でも変更可能です）。Silicon Revision は any にします。Silicon Revision については、「5 Silicon Revision について」を参照してください。

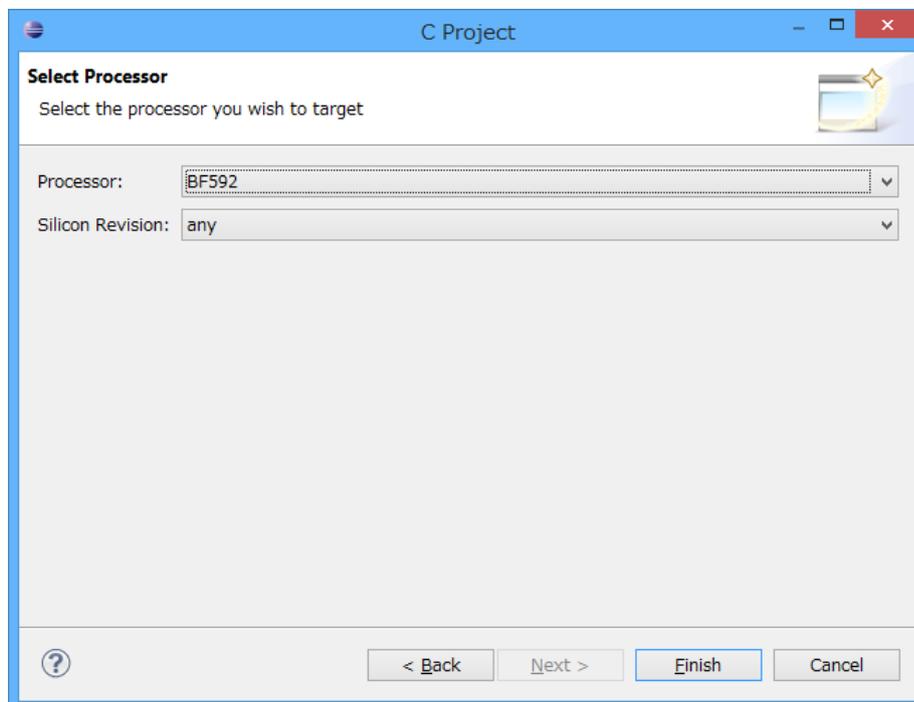


図 20 C Project ダイアログ

プロジェクトを作成すると、Project Explorer にプロジェクトが追加されます。

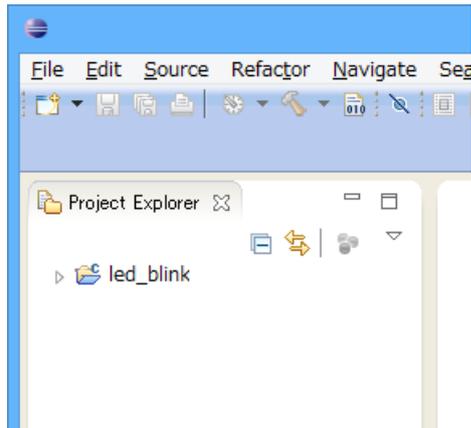


図 21 Project Explorer

もし、Welcome タブが表示されて上図のように表示されない場合は、タブの×を押して Project Explorer を表示させてください。

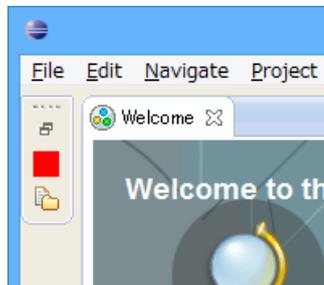


図 22 Welcome タブ

プロジェクトには何もありませんので、ソースファイルを追加します。プロジェクトを右クリックしてメニューを開き、New → Source File をクリックします。

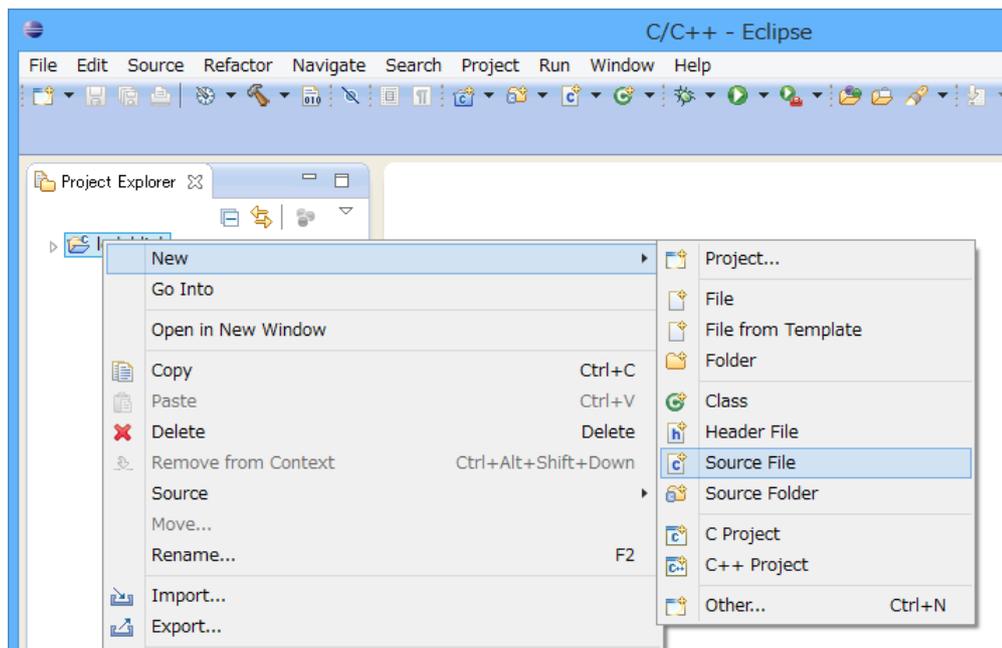


図 23 Source File の追加

追加するファイル名を Source file テキストボックスに入力します。ここでは、main.c とし、Finish ボタンを押します。

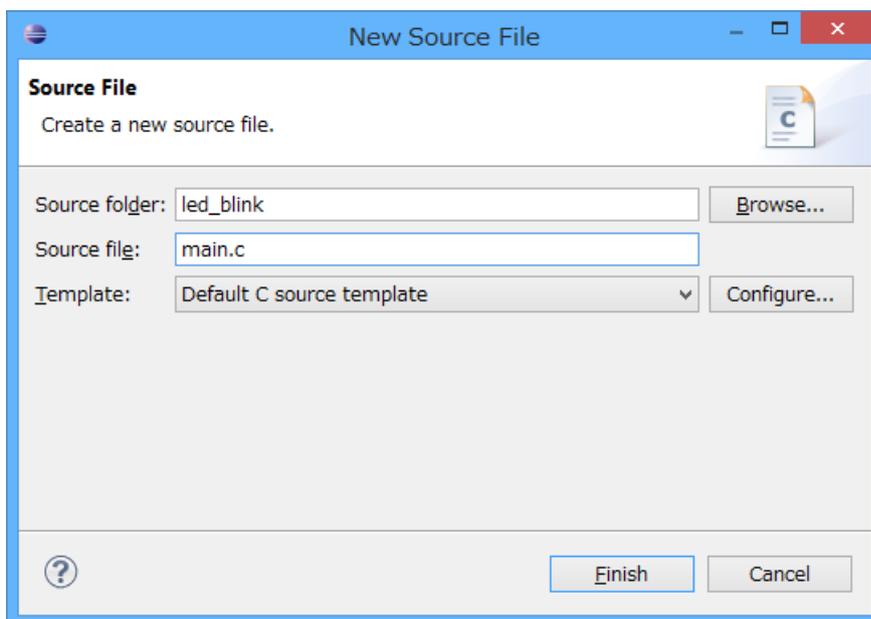


図 24 Source File の追加

プロジェクトに main.c が追加されます。

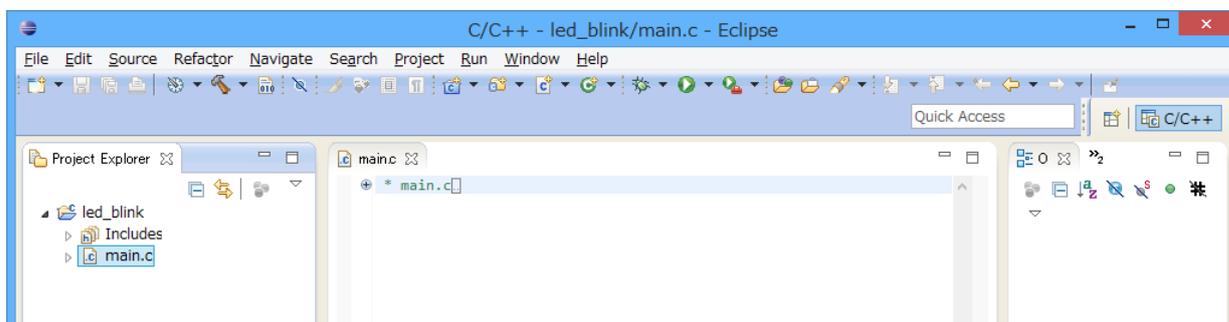


図 25 main.c の追加

これで、プログラムの入力準備が整いました。実際にプログラムを入力してみます。ここでは、ACB-BF592 の 37 番ピンに接続された LED を点滅させるプログラム例です。

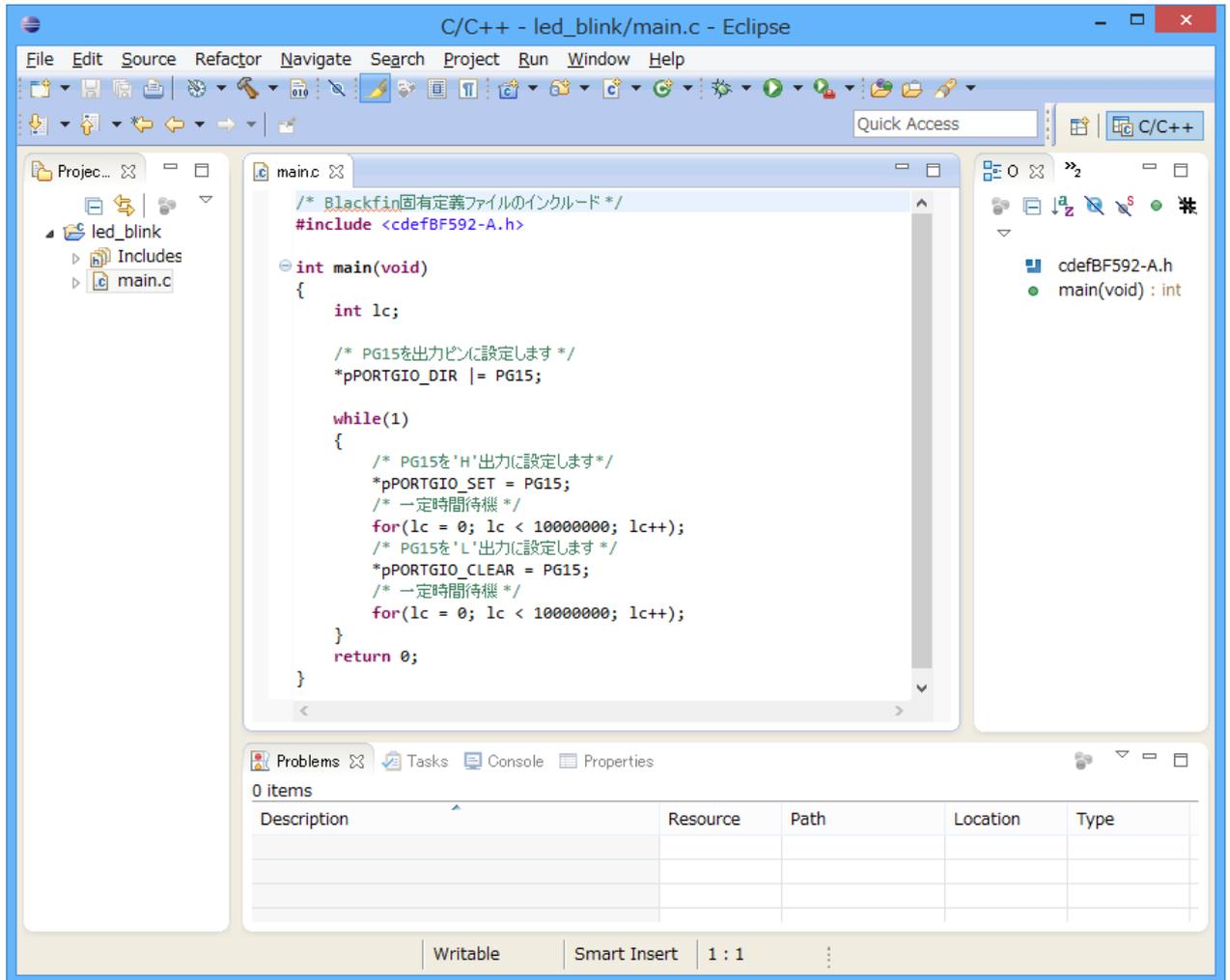


図 26 プログラム入力例

```

/* Blackfin固有定義ファイルのインクルード */
#include <cdefBF592-A.h>

int main(void)
{
    int lc;

    /* PG15を出力ピンに設定します */
    *pPORTGIO_DIR |= PG15;

    while(1)
    {
        /* PG15を'H'出力に設定します*/
        *pPORTGIO_SET = PG15;
        /* 一定時間待機 */

```

```

    for(lc = 0; lc < 10000000; lc++);

    /* PG15を'L'出力に設定します */
    *pPORTGIO_CLEAR = PG15;

    /* 一定時間待機 */
    for(lc = 0; lc < 10000000; lc++);

}

return 0;

}

```

※プログラムをそのまま入力すると、**Error** が表示されますが、ファイル保存後に消えるため、ここでは無視します。

プロジェクトをビルドするには、**Ctrl+B** を押すか、**Project Explorer** の **led_blink** を右クリックし、**Build Project** を選択します。

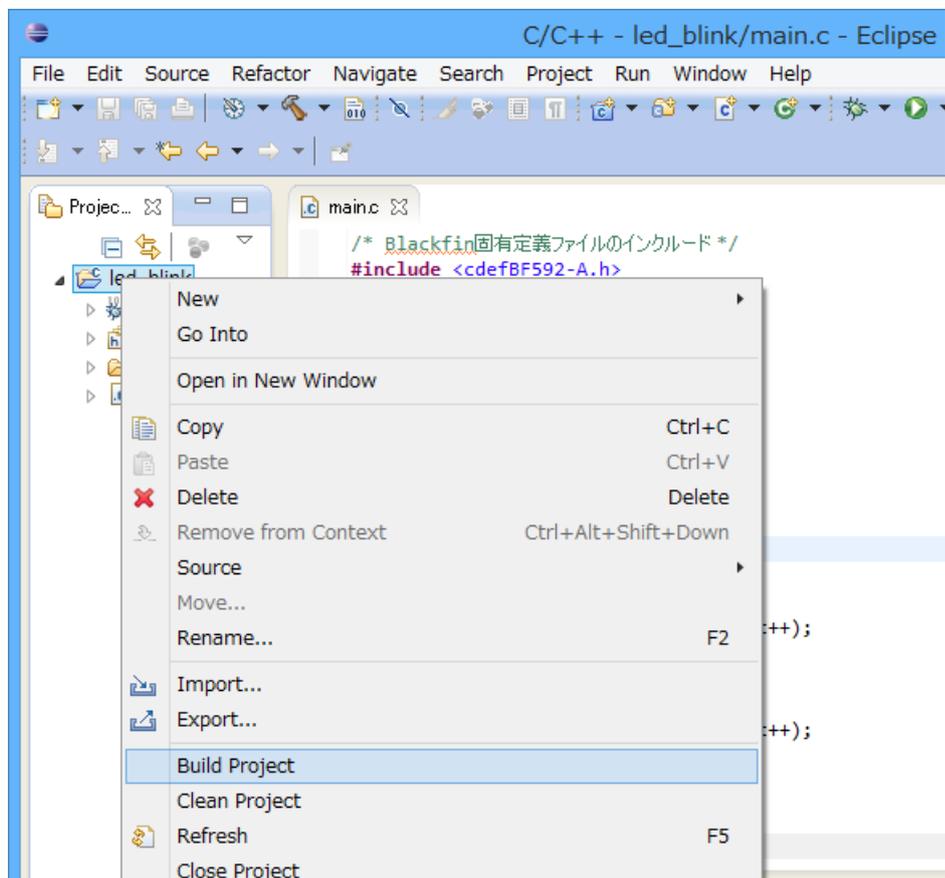


図 27 ビルド

下部の **Problem** タブのリストに、エラーが表示されなければビルド完了です。

4.2 gdb によるデバッグ

ここでは **gnICE+** を使って、**Eclipse** からデバッグを行う方法を説明します。**ACB-BF592** に **gnICE+** を接続し、電源を入れてください。

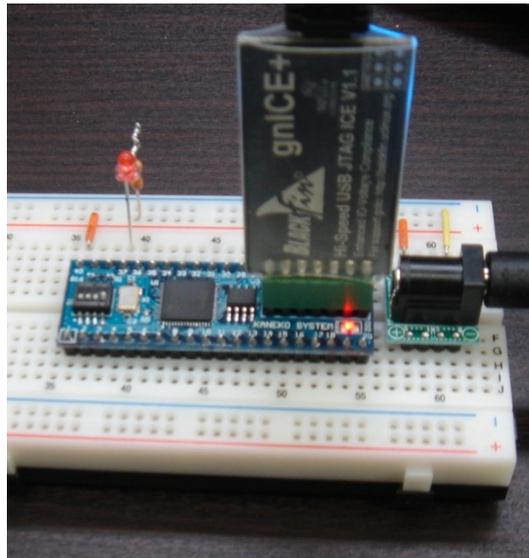


図 28 gnICE+接続例

最初に `bfm-gdbproxy` を起動する必要があります。

コマンドプロンプトを起動し、ワークフォルダ (`x:\¥Blackfin`) に移動後、`bfm-gdbproxy` と入力します。

```
C:\> C:\¥WINDOWS¥system32¥cmd.exe
c:\¥>x:
x:\¥>cd Blackfin
x:\¥Blackfin>bfm-gdbproxy
```

図 29 コマンドプロンプト

接続が正しく行われると、次のように表示されます。

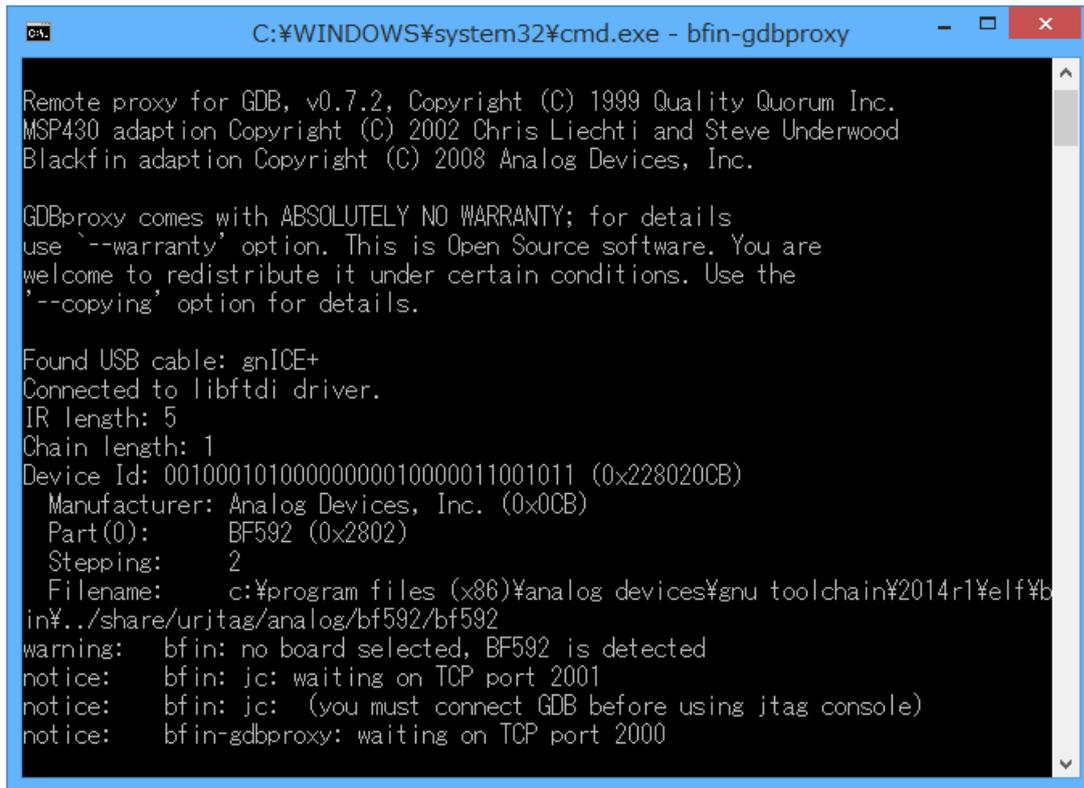


図 30 bfin-gdbproxy 接続画面

Eclipse のデバッグの設定を行います。虫アイコンの横にある▼をクリックしてメニュー表示させ、Debug As → を選択します。

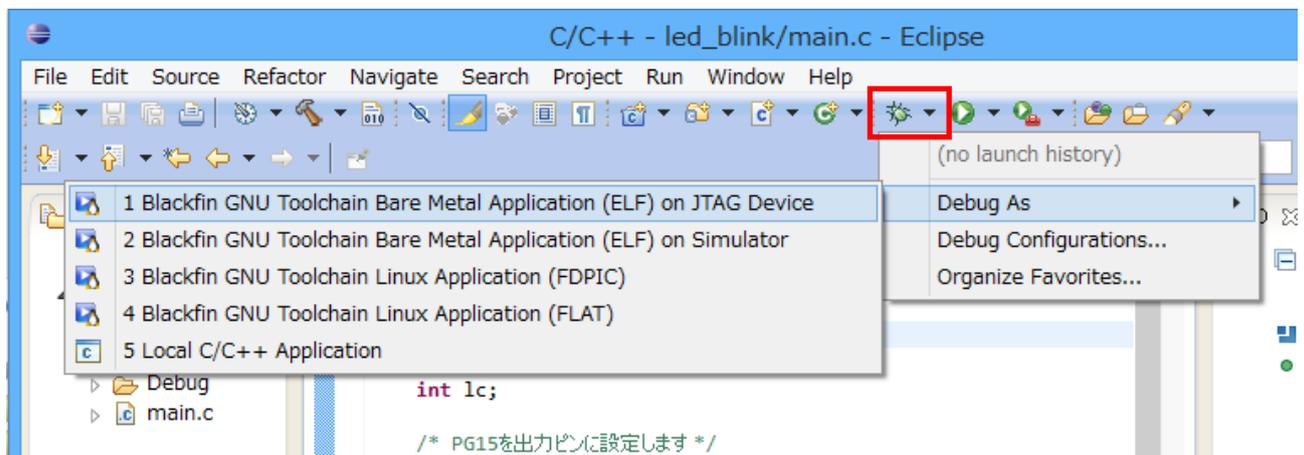
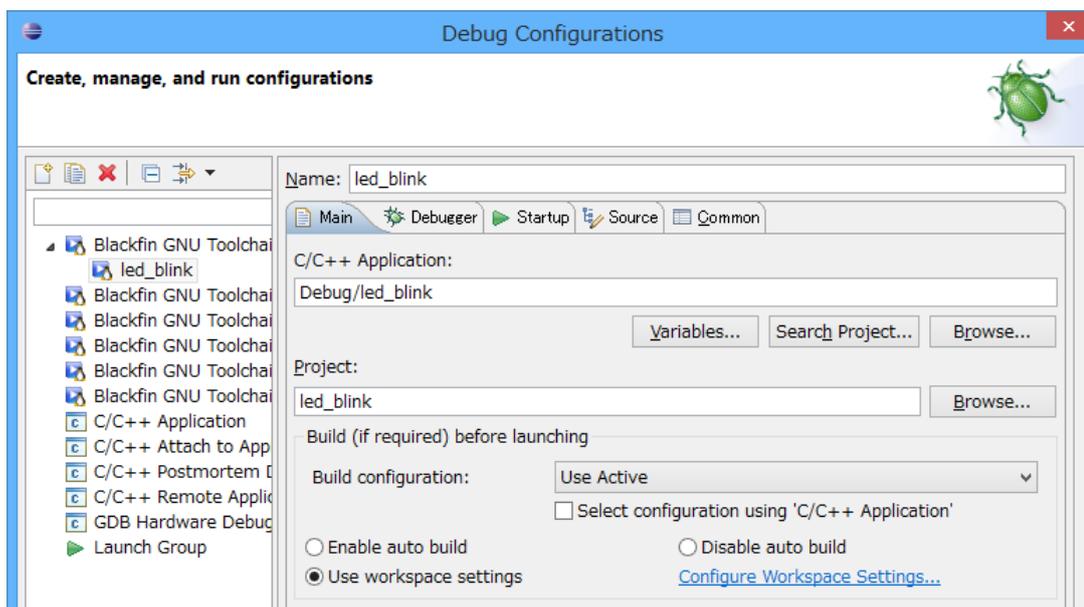


図 31 Debug Configuration メニュー

Debug Configuration メニューが表示され、リストに led_blink が自動的に作成されます。



☒ 32 Debug Configuration メニュー

Debugger タブを選択し、Remote Target の Port Number を 2000 に変更します。

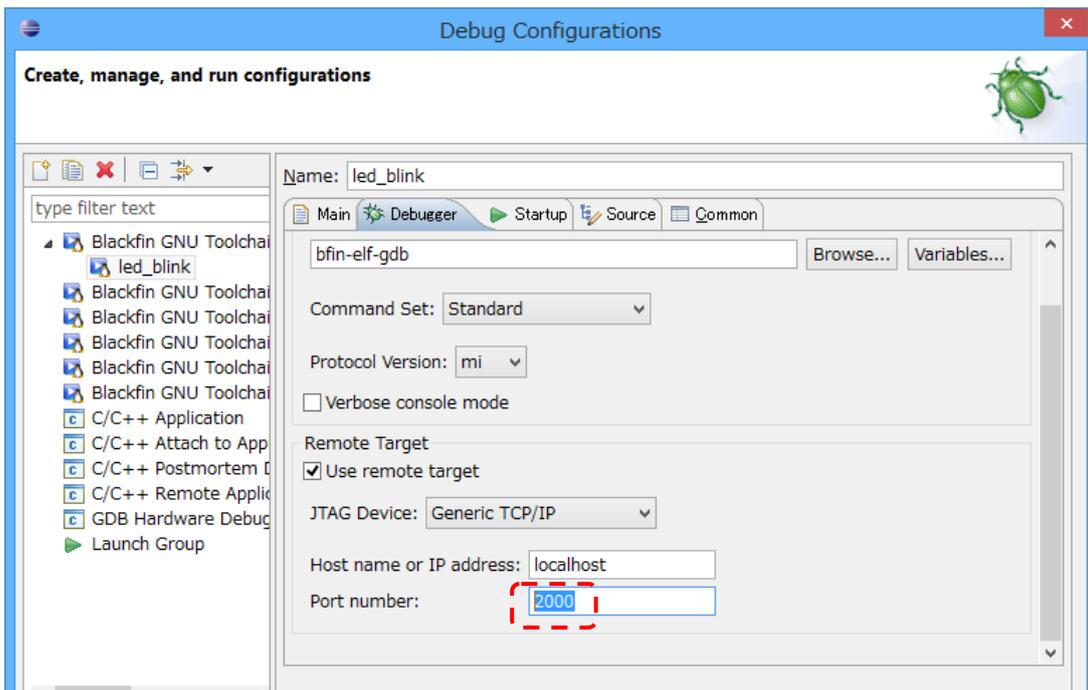


図 33 Remote Target の Port Number

Startup タブを選択し、Initialization Commands の下図の位置に monitor reset を追加します。最後に、Apply を押して設定を保存し、Debug ボタンを押します。

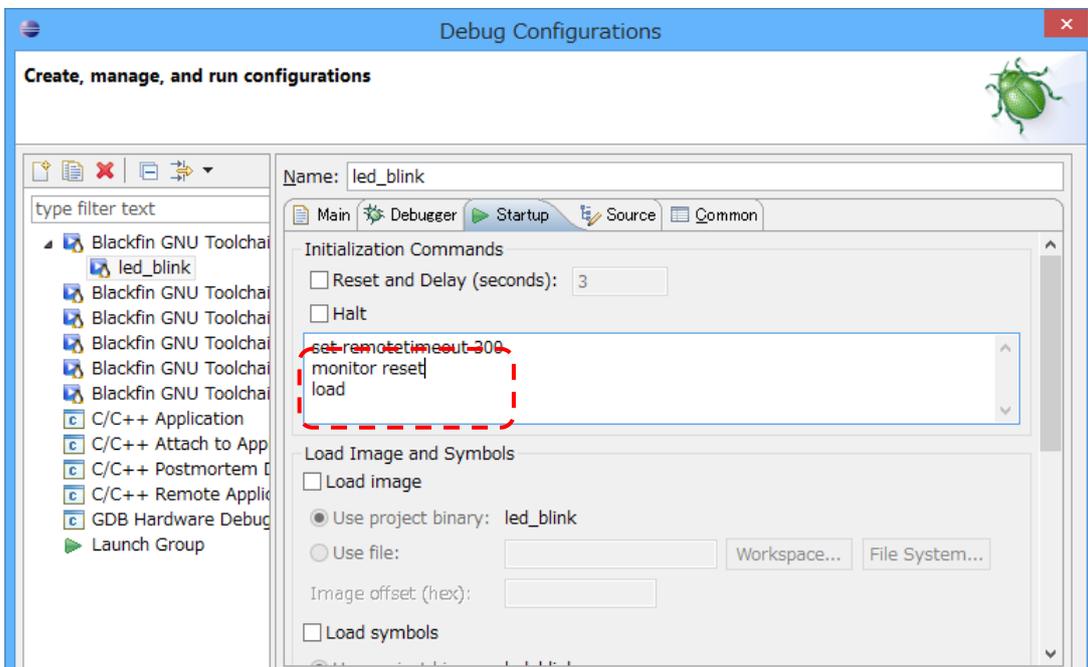


図 34 Startup の Initialization Commands

Eclipse のデバッグ画面に変更するかどうか聞かれますので、Yes を押します。

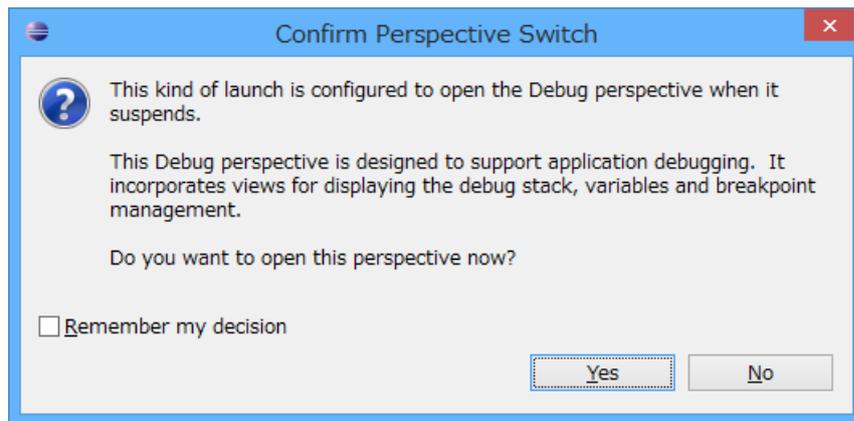


図 35 Confirm Perspective Switch ダイアログ

問題がなければ、main の最初のコードで停止します。

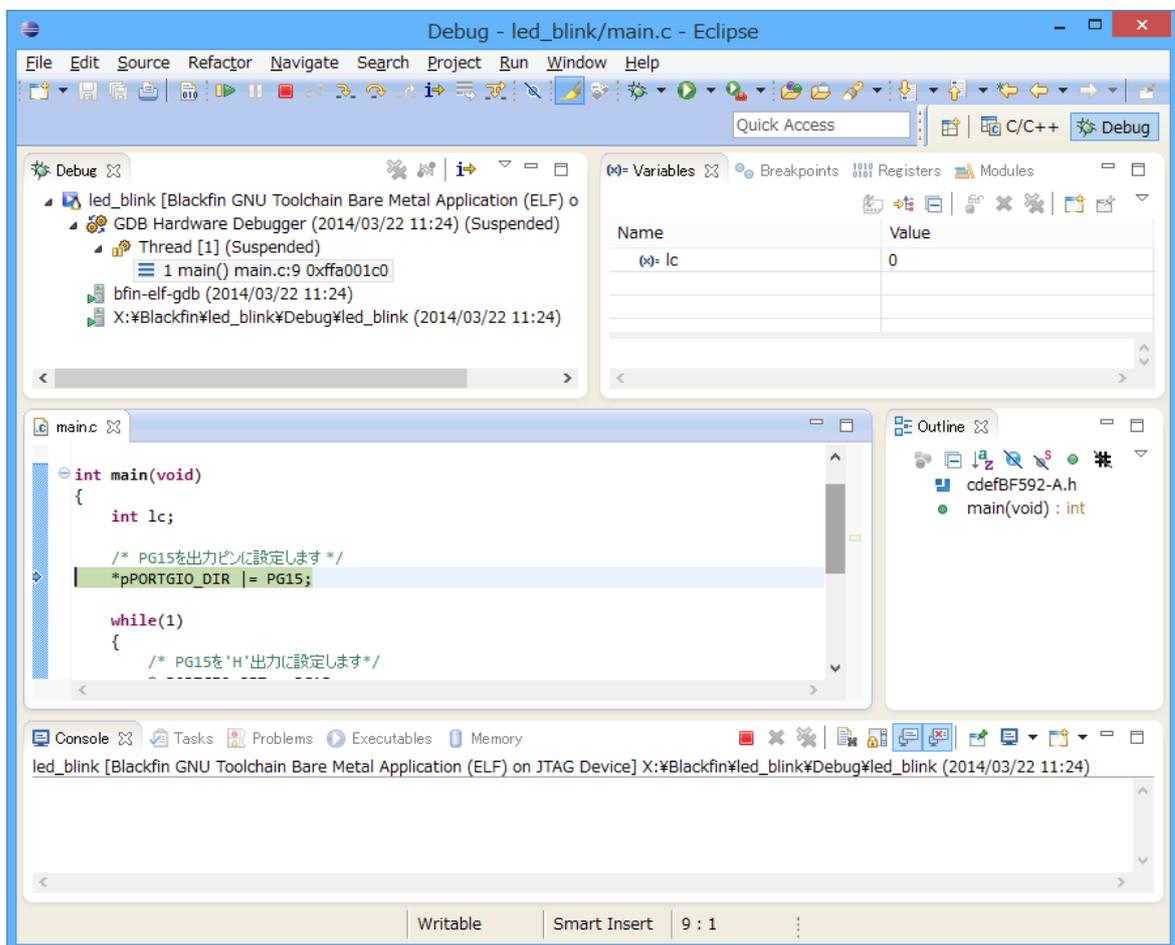


図 36 ステップ実行画面

この状態でステップ実行が可能です。そのまま実行させると、ACB-BF592 の 37 番ピンに接続された LED が点滅します。

5 Silicon Revision について

シリコンリビジョンとは、Blackfin チップのバージョンの事です。0.0 から始まり、新しいほど数字が大きくなります。新しい ADSP-BF592 では特に目立った問題はありませんが、ADSP-BF53x や ADSP-BF51x などの古いリビジョンでは、DMA の優先順位がコア優先から変更できない、内蔵 SRAM のメモリ配置にミスがあるなどの問題があります。具体的な内容は、Analog Devices 社のサイトに IC アノマリとして公開されており、一度確認されることをお勧めします。

<http://www.analog.com/jp/processors-dsp/Blackfin/processors/ic-anomalies/resources/index.html>

最近では Blackfin プロセッサのバグもかなり修正されており、どの Blackfin プロセッサでも問題ないのですが、特に理由がなければ、新しいリビジョンのチップを使用してください。

なお、コンパイラに Silicon Revision を指定するのは、これらのバグを回避する対策を指示するためで、any を指定すると全リビジョンに対応しますが、コード又は、データサイズが若干増加するケースがあります。

6 更新履歴

版	更新日	更新内容
第 1 版	2012/02/13	初版発行
第 2 版	2012/07/05	P.21 サンプルソースのコメントミスを修正
第 3 版	2012/07/28	P.24 図 37 monitor reset コマンドの入力を追加
第 4 版	2012/08/01	全体 Eclipse Juno に説明入れ替え
第 5 版	2012/09/18	全体 Blackfin Toolchain 2012R1 に説明入れ替え
第 6 版	2013/07/04	全体 Blackfin Toolchain 2012R2 に説明入れ替え Eclipse Juno SR1 に説明入れ替え
第 7 版	2014/03/22	全体 Blackfin Toolchain 2014R1 に説明入れ替え Eclipse Kepler RC2 に説明入れ替え