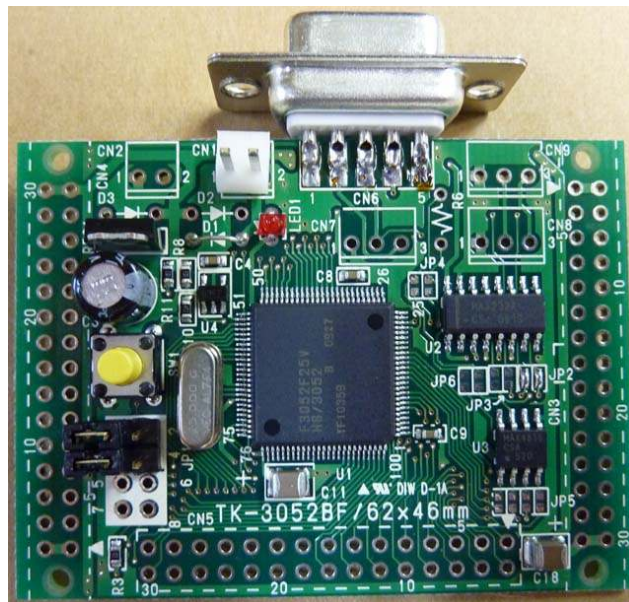


TK-3052

ユーザーズマニュアル (Ver.1.02)



第 1 章	TK-3052 の組み立て	P. 1
第 2 章	ハードウェアの設定	P. 3
第 3 章	プログラム開発について	P. 4
第 4 章	モニタプログラム“Hterm”について	P. 15
第 5 章	μ ITRON(HOS)の実装	P. 26
	付録(回路図, シルク図)	P. 29

第1章	TK-3052 の組み立て
	1. 部品の確認 2. 工具の用意 3. 部品のハンダ付け

1. 部品の確認

部品がそろっているか確認してください。不足部品があるときは巻末連絡先の東洋リンクスまでメールか FAX でご連絡ください。

TK-3052 部品表

部品番号	型名, 規格	メーカー	数量	備考
1 ●キット部品(完成品は実装済み)				
2 C1	47~100 μ F/16V		1	電解コンデンサ
3 CN1	B2P-SHF-1AA	JST	1	
4 CN3, 4	PS-30SD-D4TS1-1	航空電子	2	相当品可, ハンダ面に実装
5 CN6	HDEB-9S	HRS	1	相当品可, ケーブルハンダ付け仕様
6 JP1			1	ピンとソケットのセット
7 LED1	HLMPシリーズ/QLMPシリーズ	AVAGO	1	相当品可
8 REG1	TA4805F(S)	東芝	1	
9 SW1	SKHHAK/AM/DC	ALPS	1	相当品可
10 X1	25MHz		1	水晶(HC-49), または3端子セラロック
11 基板	TK-3052BF	東洋リンクス	1	表面実装部品 実装済み
12				
13 ●同封品				
14 電池BOX			1	単3×4本用, ケーブル付
15				
16 ●必要に応じ用意する部品(製品には含まれていません, パーツショップでご購入ください)				
17 CN2	B2P-SHF-1AA	JST	1	
18 CN5	PS-30SD-D4TS1-1	航空電子	1	相当品可, ハンダ面に実装
19 CN7, 8, 9	B3P-SHF-1AA	JST	3	
20 D1, 2, 3	D1N20		3	相当品可
21 R6	100 Ω		1	1/4W

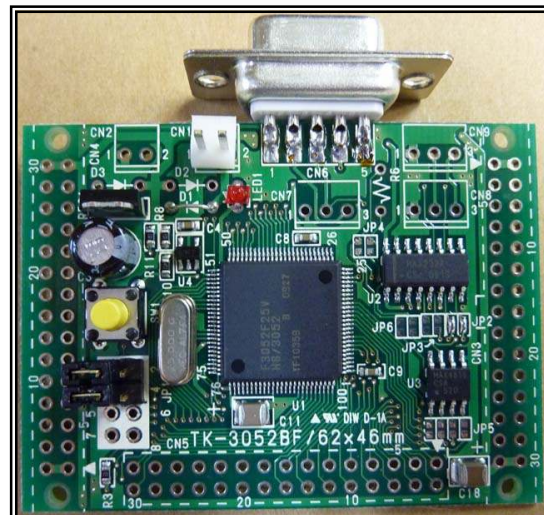
2. 工具の用意

部品の確認ができれば、次は工具を準備しましょう。最低限必要なものはハンダゴテ・ハンダ・コテ台・ニッパ・ラジオペンチです。ハンダゴテは20~30W 程度のものを用意して下さい。ハンダゴテは大変熱くなり、肌に触れると火傷の危険があるのでコテ台も用意しましょう。ニッパは余分なリード線を切るために、ラジオペンチはリード線を折り曲げるために使用します。



3. 部品のハンダ付け

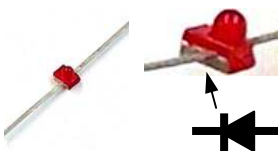
LED や電解コンデンサなどハンダ付けする部品のほとんどに取り付ける向き(極性)があり, 間違えると動かなかったり故障の原因になります。間違えて付けてしまわないよう十分に注意してハンダ付けを始めて下さい(写真参照)。もし, 間違えて取り付けた場合には, 真空式のハンダ吸取り器などを使用すれば, 無理しないで部品を引き抜くことができます。無理に引き抜くと部品の損傷や基板のスルーホールの導通が無くなり故障の原因となります。



注意してハンダ付けしたい部品を説明します。

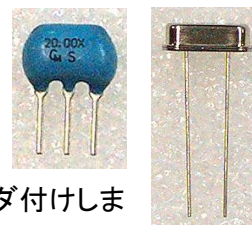
発光ダイオード(LED1)

赤, もしくは黄色の小さな部品が発光ダイオードです。リード線のついている根元を見て下さい。片側だけ銀色になっているのが見えます。この銀色側のリード線が基板上的記号, 三角の向いている側です。リード線を根元で曲げ, 基板上的記号に合わせて差し込みハンダ付けします。この部品もハンダ付けの前にもう一度差し込む方向を確認しましょう。



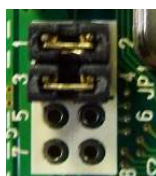
発振子(X1)

青色のセラロックか, 銀色の水晶のいずれかが入っています。セラロックの場合は 3 ヶ所, 水晶の場合は 2 ヶ所ハンダ付けします。



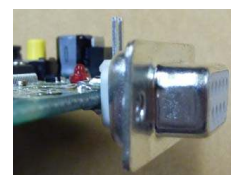
ジャンパ(JP1)

写真のように 1-2, 3-4 に実装します。



コネクタ(CN6)

写真のように基板をコネクタのピンではさむようにしてハンダ付けします。なお, RS-232C を 3 ピンコネクタ(CN7)で使うときは実装しません。なお, JP2 はショートしてください。



ダイオード(D1)

写真のようにリード線でショートします。



コネクタ CN3, CN4, CN5

基板の裏側に実装します。

全ての部品をハンダ付けし終わったら, 念のため, もう一度基板を見て確認してみましょう。

部品の付け忘れはありませんか?

取り付ける向きはあっていますか?

ハンダ付けはきれいにできていますか?

第2章

ハードウェアの設定

1. 動作モード(JP1)
2. RS-232C-Ch1(JP2)
3. RS-232C-Ch0(JP3, JP4)
4. RS-485(JP5, JP6)
5. バッテリバックアップ

1. 動作モード(JP1)

H8/3052 の動作モードを設定します。動作モードの詳細はルネサスエレクトロニクスが配布している「H8/3052B ハードウェアマニュアル」をご覧ください。なお、TK-3052 はモード 7(シングルチップアドバンスドモード)で使うことを前提にしています。他のモードで使う場合は、JP1 の未実装部分にジャンパを実装してください。

	JP1				
	1-2	3-4	5-6	7-8	
	FWE	MD2	MD1	MD0	
モード 1	OFF(0)	ON(0)	ON(0)	OFF(1)	
モード 2	OFF(0)	ON(0)	OFF(1)	ON(0)	
モード 3	OFF(0)	ON(0)	OFF(1)	OFF(1)	
モード 4	OFF(0)	OFF(1)	ON(0)	ON(0)	
モード 5	OFF(0)	OFF(1)	ON(0)	OFF(1)	
モード 6	OFF(0)	OFF(1)	OFF(1)	ON(0)	
モード 7	OFF(0)	OFF(1)	OFF(1)	OFF(1)	シングルチップアドバンスドモード(RUN)
ブートモード 5	ON(1)	ON(0)	ON(0)	OFF(1)	
ブートモード 6	ON(1)	ON(0)	OFF(1)	ON(0)	
ブートモード 7	ON(1)	ON(0)	OFF(1)	OFF(1)	シングルチップアドバンスドモード(書込み)
設定禁止	ON(1)	OFF(1)	ON(0)	ON(0)	
ユーザプログラムモード 5	ON(1)	OFF(1)	ON(0)	OFF(1)	
ユーザプログラムモード 6	ON(1)	OFF(1)	OFF(1)	ON(0)	
ユーザプログラムモード 7	ON(1)	OFF(1)	OFF(1)	OFF(1)	

2. RS-232C-Ch1(JP2)

TxD1, RxD1 を RS-232C で通信するときにショートします。コネクタは CN6 か CN7 のいずれかを使用します。

3. RS-232C-Ch0(JP3, JP4)

TxD0, RxD0 を RS-232C で通信するときに両方ともショートします(JP5,JP6 はオープン)。コネクタは CN8 か CN9 のいずれかを使用します。RS-485 は使用できません。

4. RS-485(JP5, JP6)

TxD0, RxD0 を RS-485 で通信するときに両方ともショートします(JP3,JP4 はオープン)。コネクタは CN8 と CN9 を使用します。端末抵抗として R6 に 100Ω を実装します。RS-232C-Ch0 は使用できません。

5. バッテリバックアップ

バッテリバックアップを行なうときは、CN2, D1, D2, D3 を実装します。3V 以上のバッテリーを CN2 に接続してください。

第3章

プログラム開発について

1. ツールのダウンロードとインストール
2. コンパイルとダウンロード
3. 開発環境について

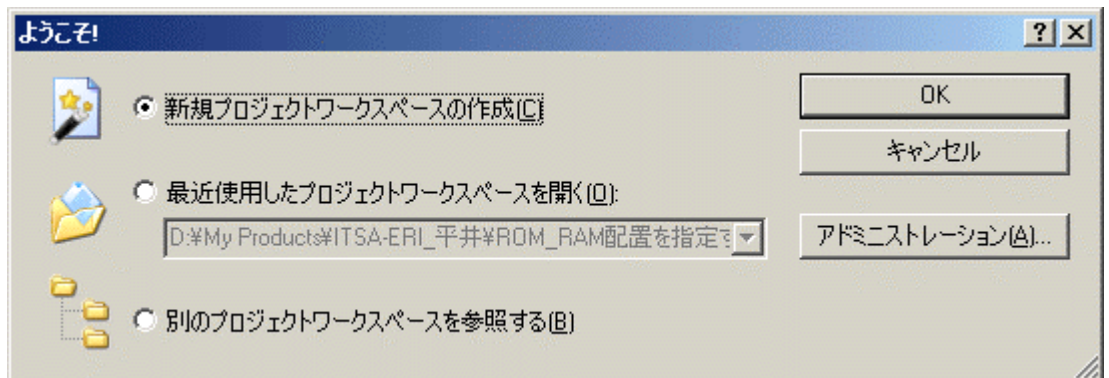
1. ツールのダウンロードとインストール

ルネサスエレクトロニクスの Web サイトから HEW と FDT をダウンロードします。詳細は弊社 CD の「_必ずお読みください」フォルダの中にある「ルネサスダウンロード.pdf」をご覧ください。

HEW, FDT ともに無償評価版です。ルネサスエレクトロニクスのサポートは一切ありません。また、HEW は、60 日間は製品版と同等の機能と性能のまま使用できますが、61 日目以降はコード+データサイズが 64K バイトまでに制限されます。H8/3052BF のフラッシュメモリサイズは 512K バイト、RAM サイズは 8K バイトありますので、コード+データサイズが 64K バイト以上のプログラムを作る場合は製品版 HEW の購入をおすすめします。

2. コンパイルとダウンロード

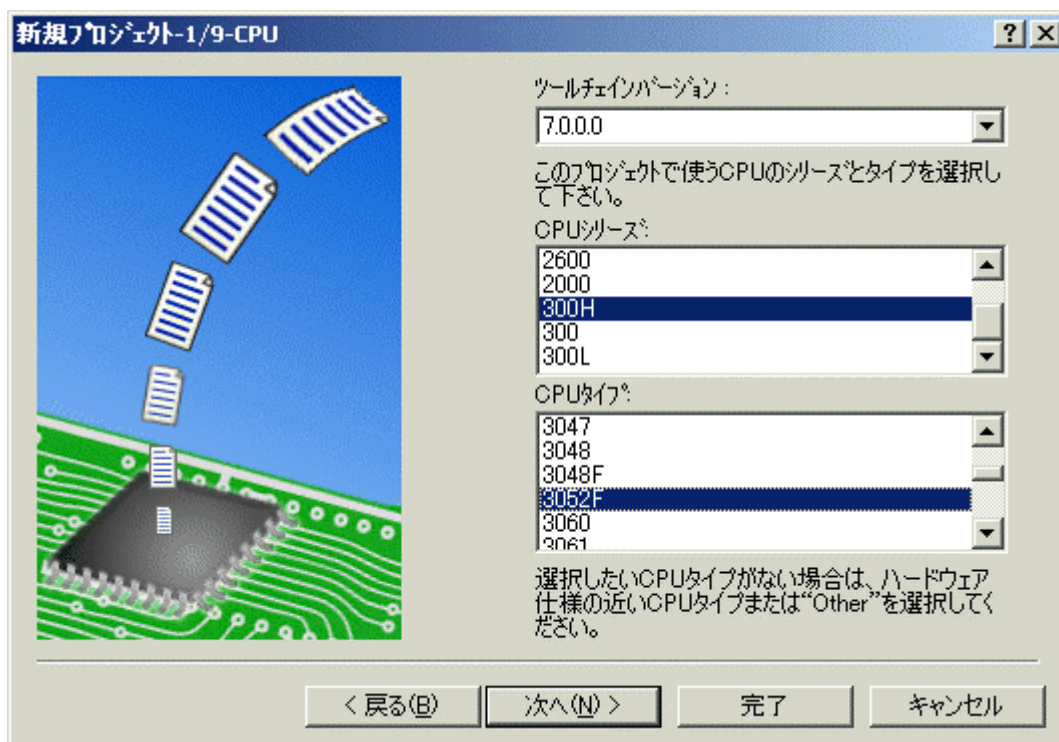
サンプルとして LED を点滅するプログラムを作成します。スタートメニューから、もしくはデスクトップの「HEW4」をクリックして HEW を起動します。すると「ようこそ！」ダイアログが表示されますので、「新規プロジェクトワークスペースの作成」を選択して「OK」をクリックします。



「新規プロジェクトワークスペースダイアログ」が表示されますので、次のように入力して「OK」をクリックします。



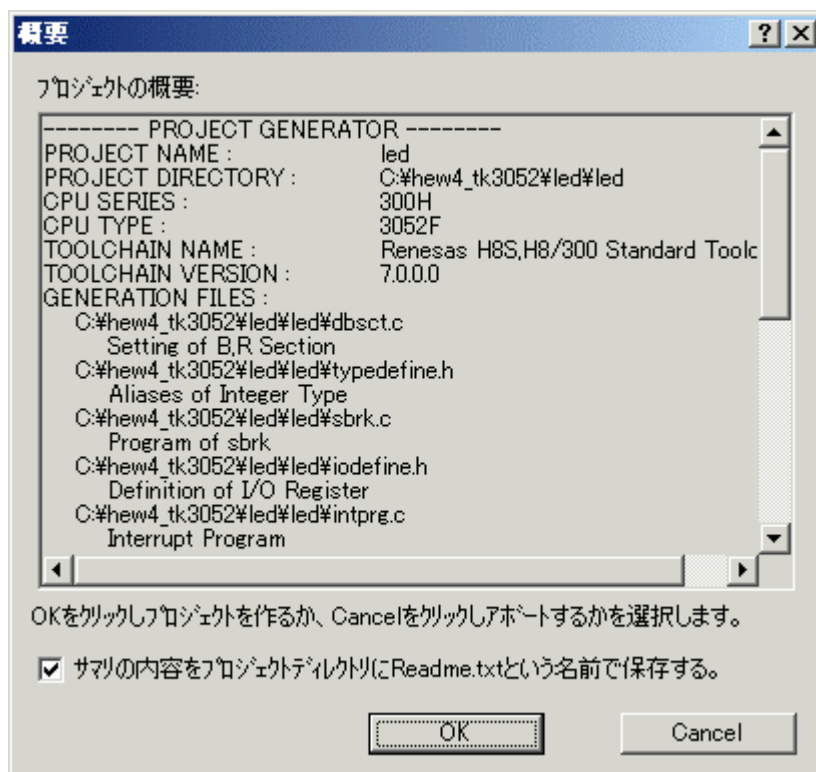
「新規プロジェクト-1/9-CPU」ダイアログが表示されますので、CPU シリーズは「300H」、CPU タイプは「3052F」を選択します。「次へ」をクリックします。



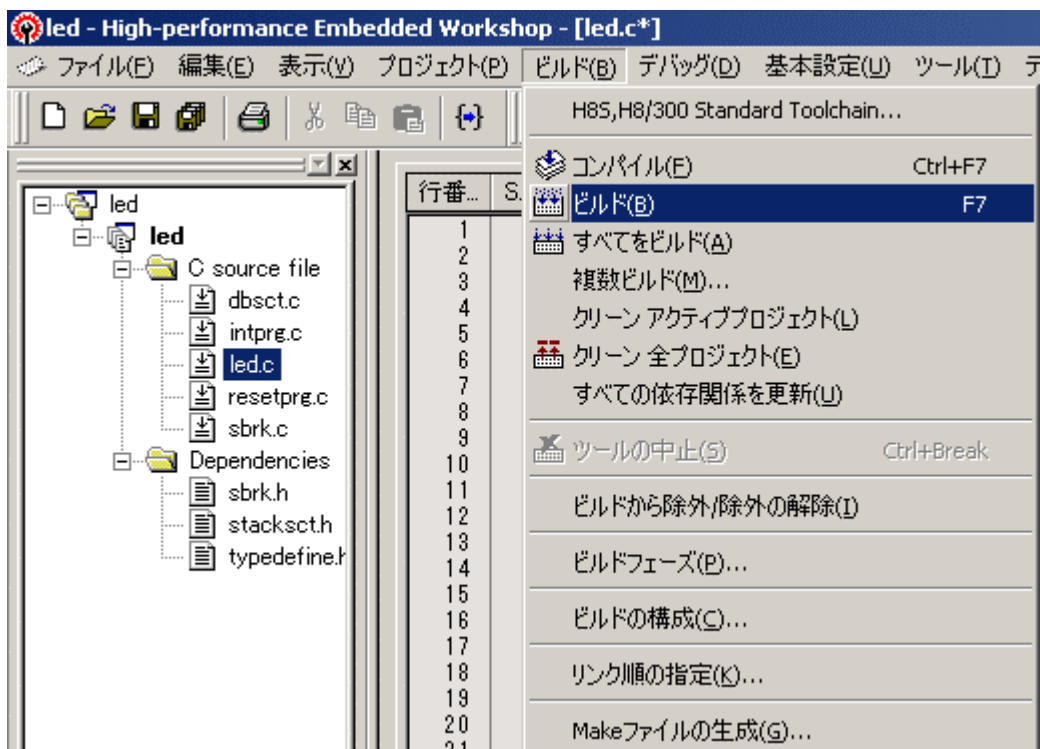
「新規プロジェクト-2/9-オプション」ダイアログが表示されます。シングルチップアドバンスドモードの場合、アドレス空間を「1M byte」にします。あとの項目は変更しませんので「完了」をクリックします。（変更が必要な場合もあります。そのときは「次へ」をクリックして設定します。）



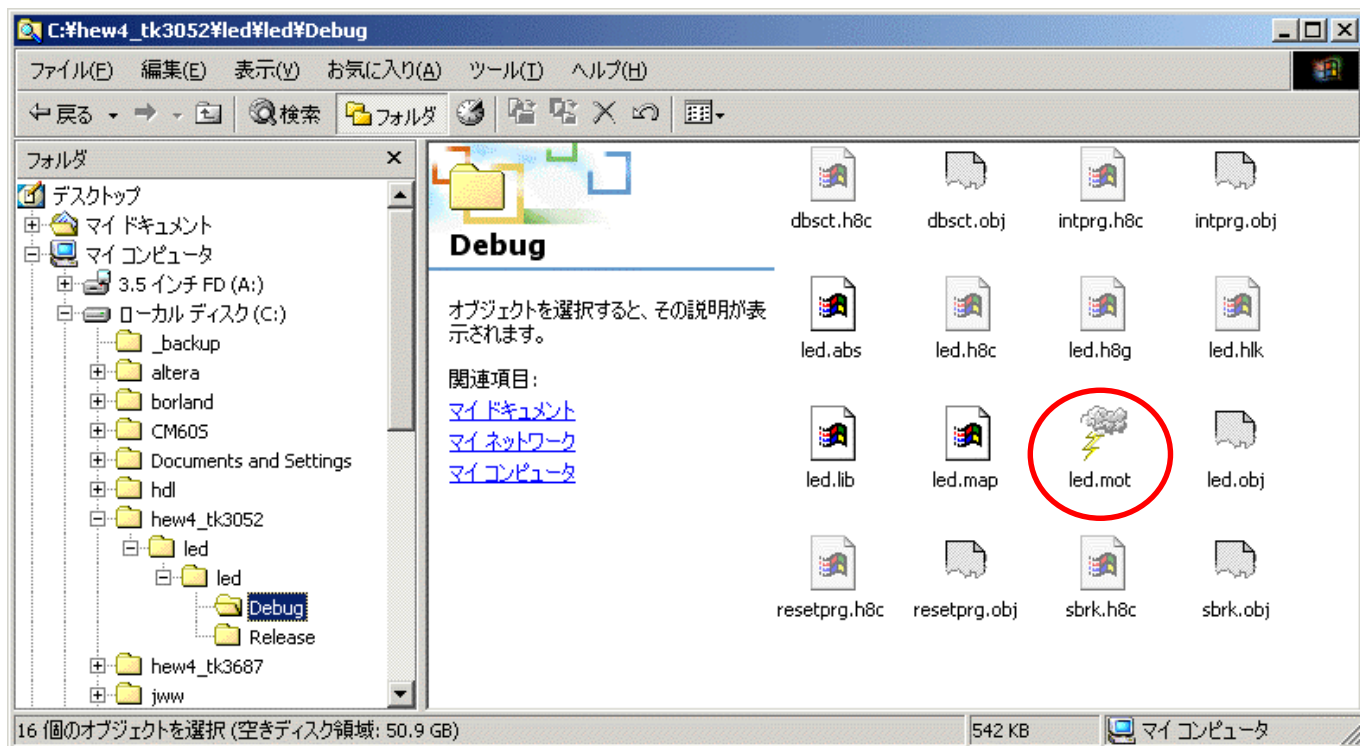
すると「概要」ダイアログが表示されます。「OK」をクリックすればプロジェクトの作成は終了です。



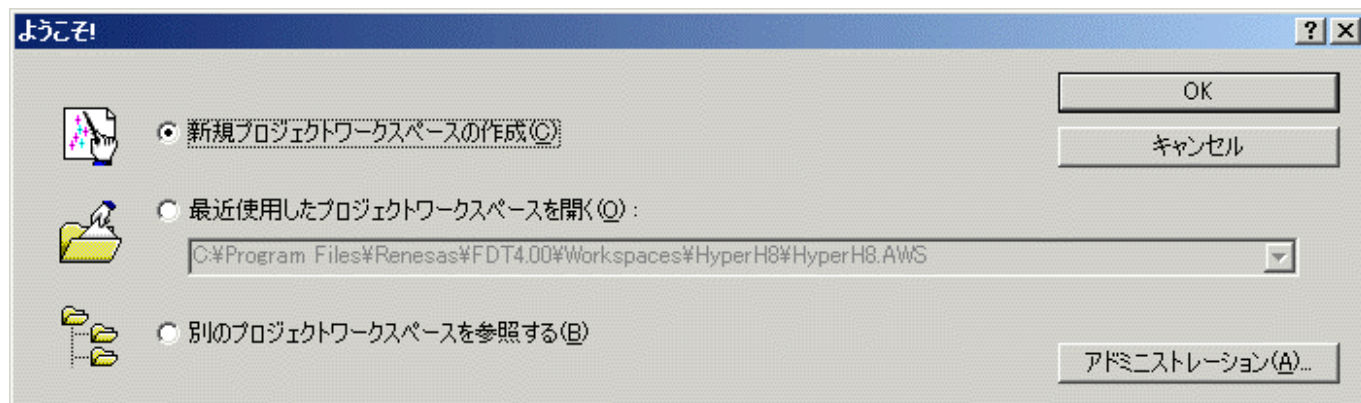
ビルドしましょう。メニューから「ビルド」→「ビルド」を選択します。



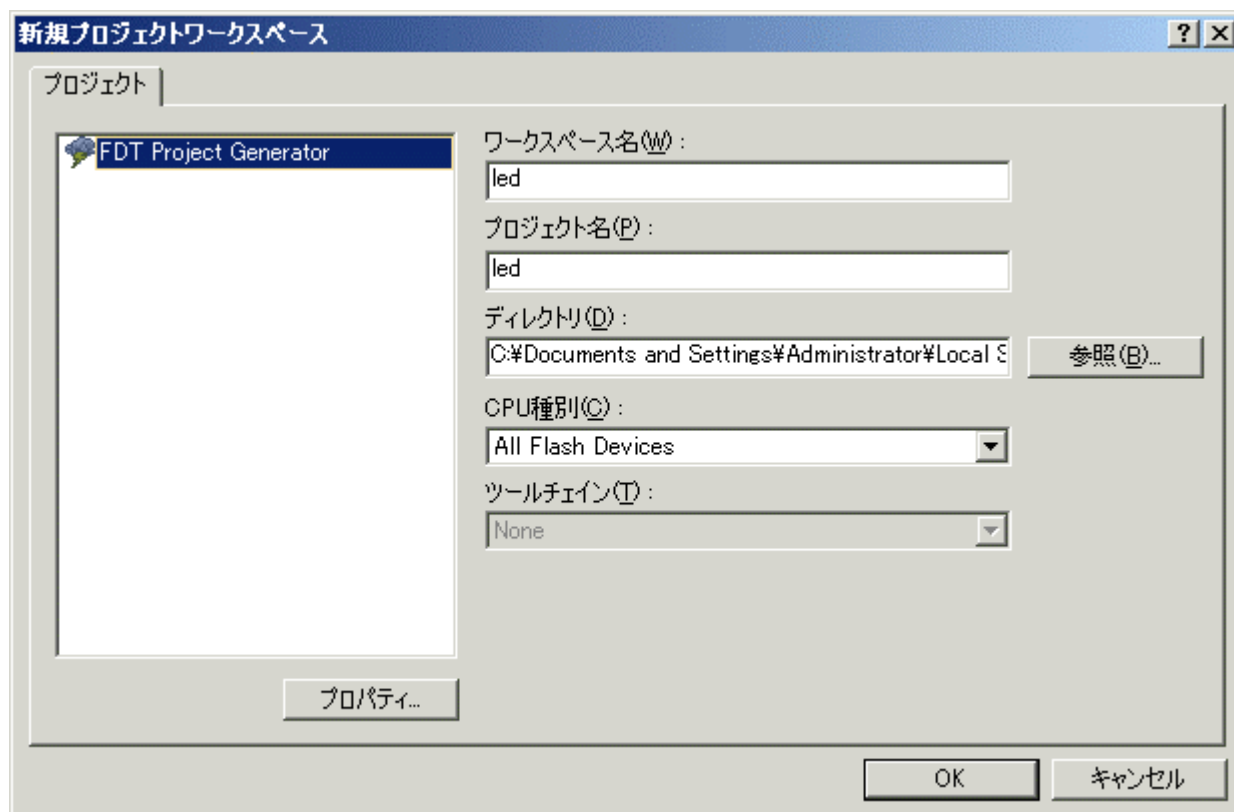
もしエラーが出たときは入力ミスの可能性が高いのでもう一度確認してください。なお、「L1110 (W) Cannot find "C" specified in option "start"」というワーニングが出ますが、これは気にしなくてかまいません。「led. mot」が作られていればビルド完了です。



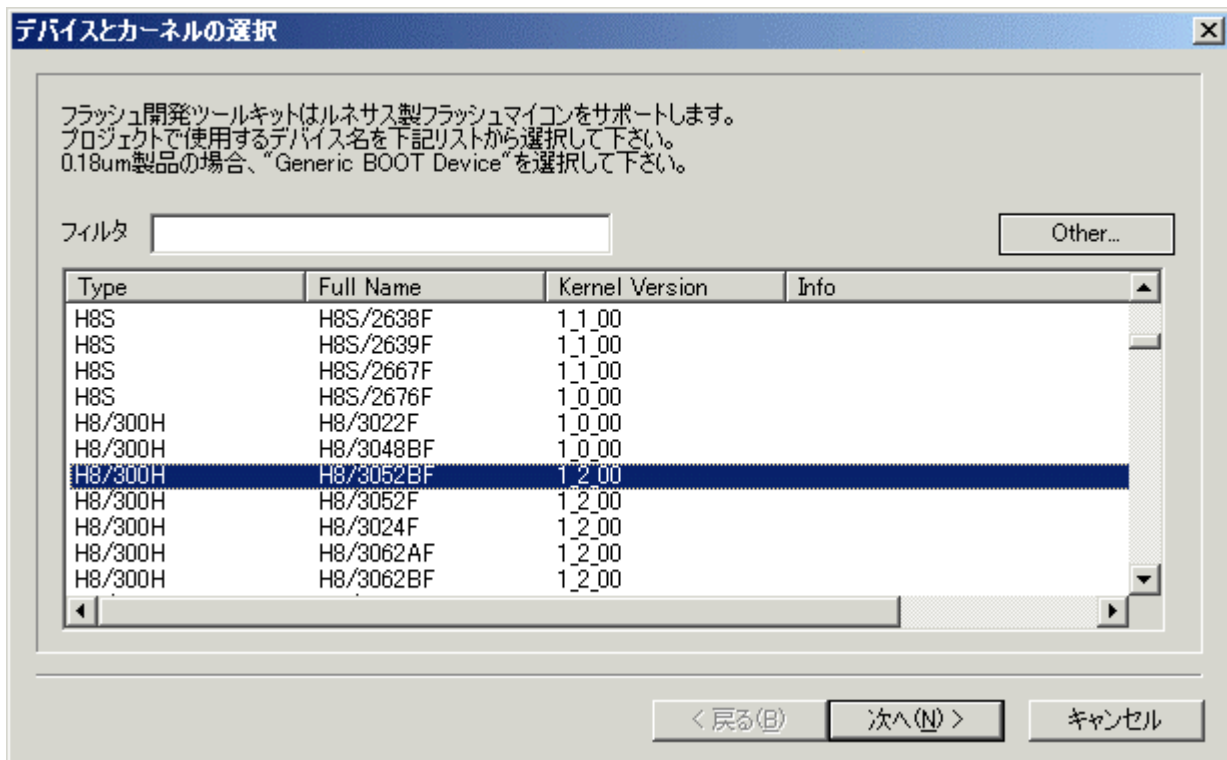
次に、生成された「led. mot」を TK-3052 の H8/3052BF にダウンロードしましょう。スタートメニューから FDT を起動します。すると、「ようこそ！」ダイアログが開きますので、「新規プロジェクトワークスペースの作成」を選択して「OK」をクリックします。



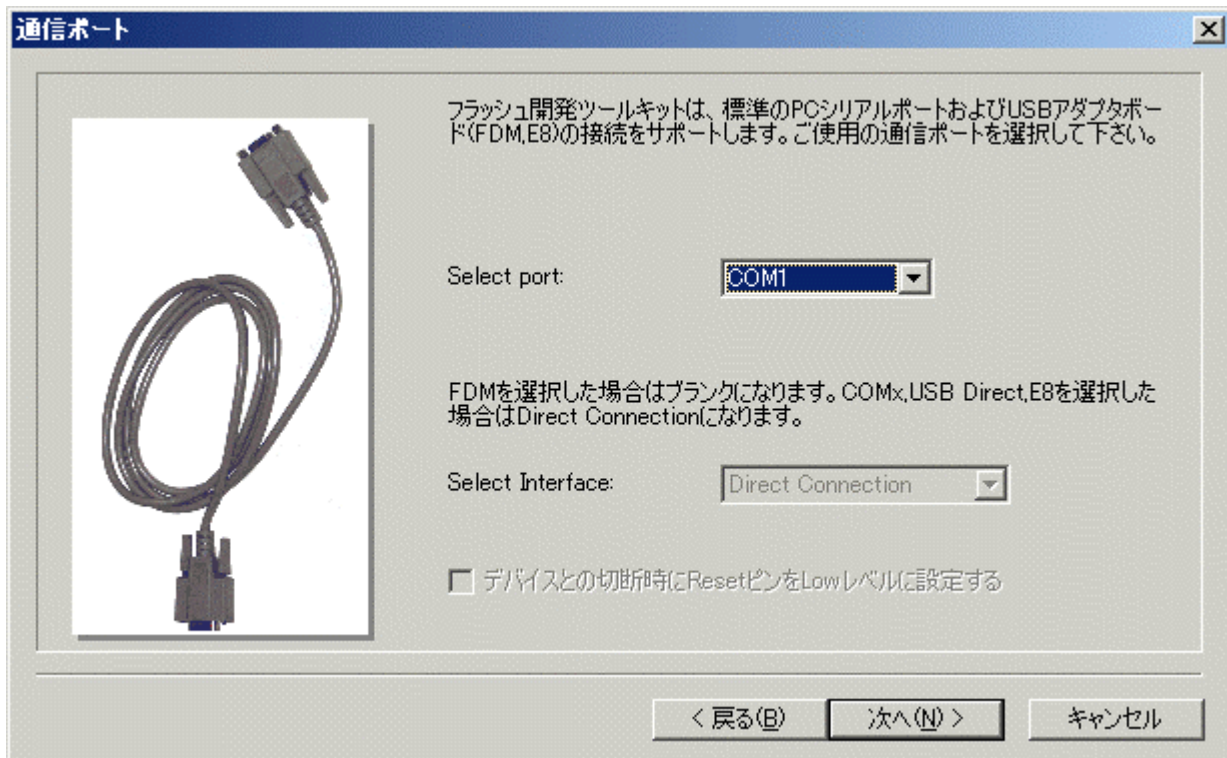
「新規プロジェクトワークスペース」ダイアログが開きます。次のように入力して「OK」をクリックします。



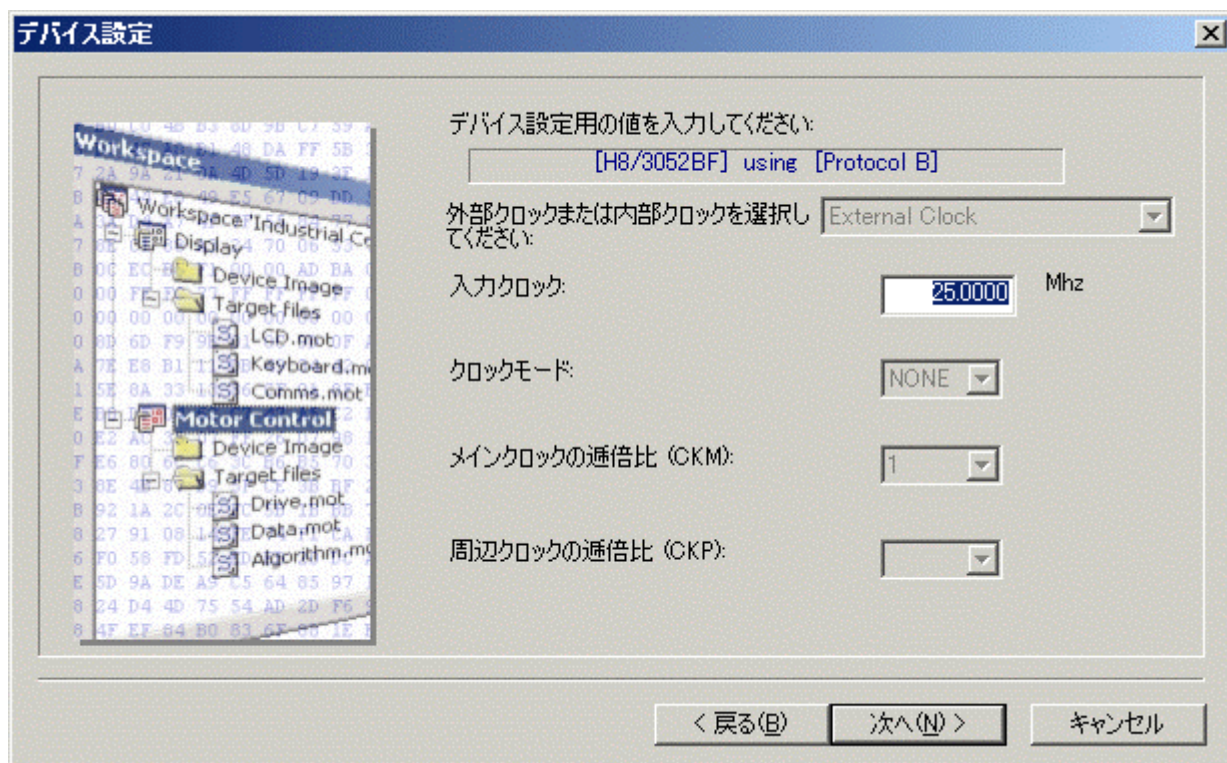
「デバイスとカーネルの選択」ダイアログが開きます。H8/3052BF を選択して「次へ」をクリックします。



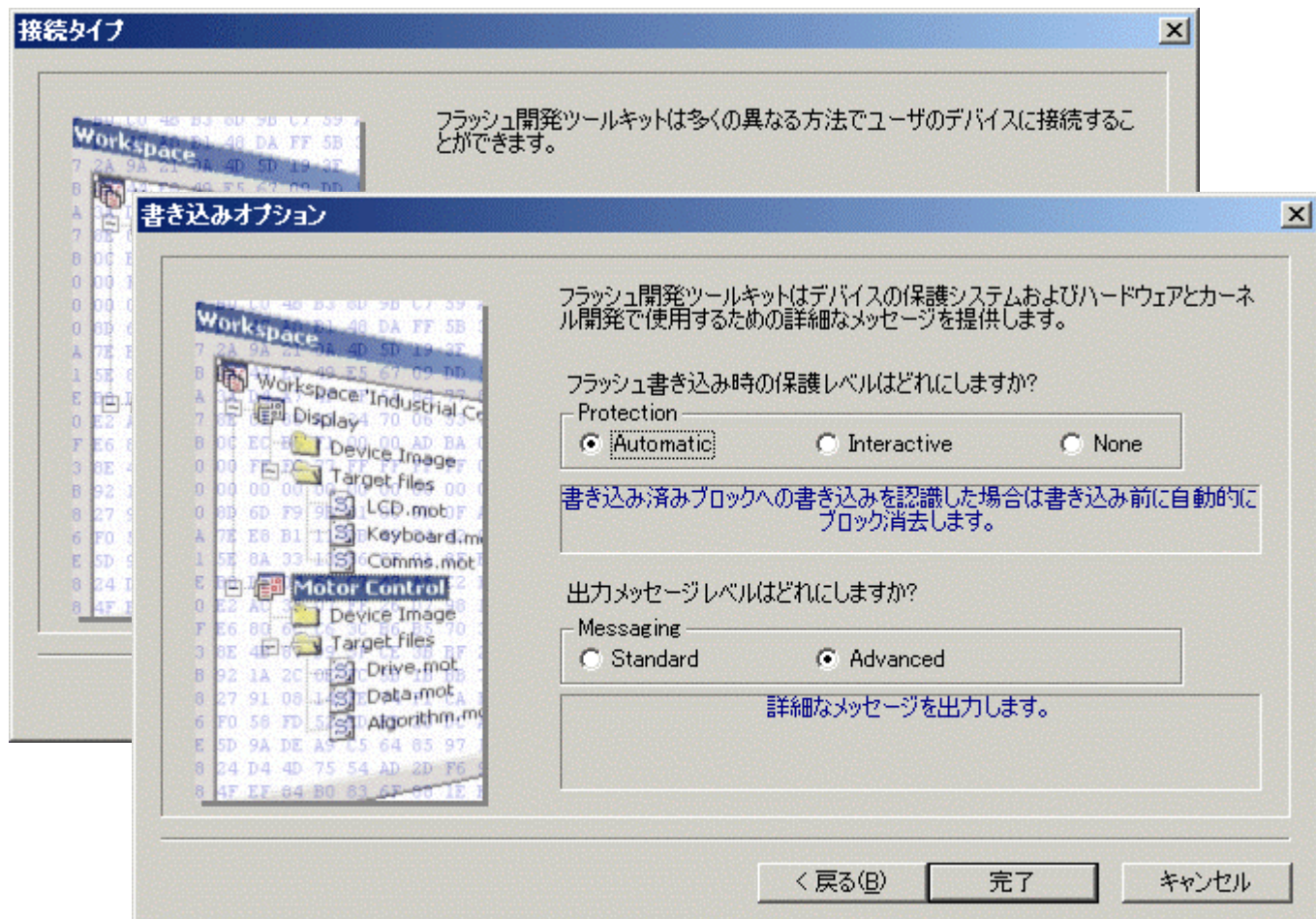
「通信ポート」ダイアログが開きます。使用するポートを選択して「次へ」をクリックします。



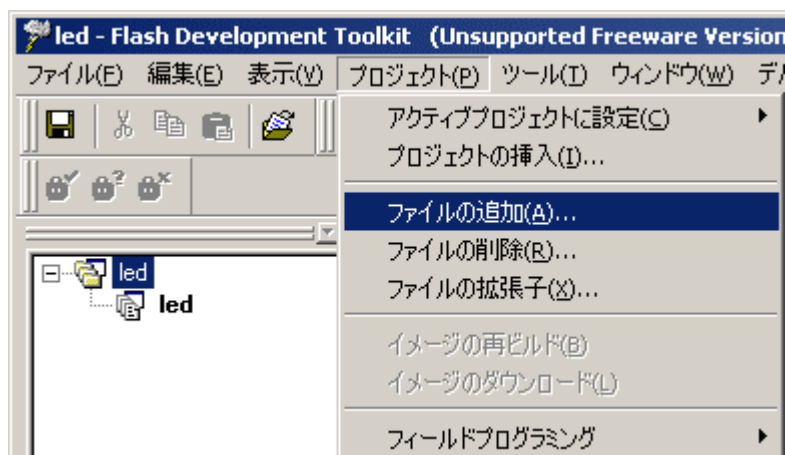
「デバイス選定」ダイアログが開きます。「入力クロック」が 25MHzになっているか確認し、「次へ」をクリックします。



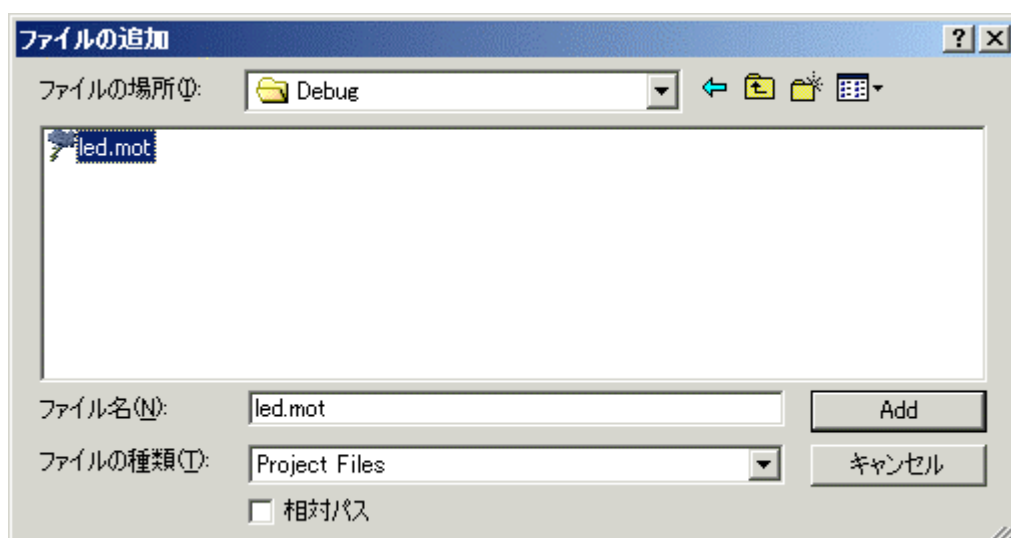
「接続タイプ」、「書き込みオプション」ダイアログが順番に開きます。これらは変更しません。「次へ」をクリックして先に進み、最後に「完了」をクリックします。



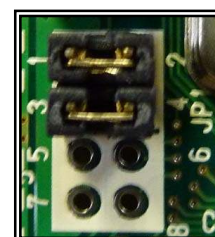
「led. mot」をプロジェクトに追加します。メニューから「プロジェクト」→「ファイルの追加」を選択します。



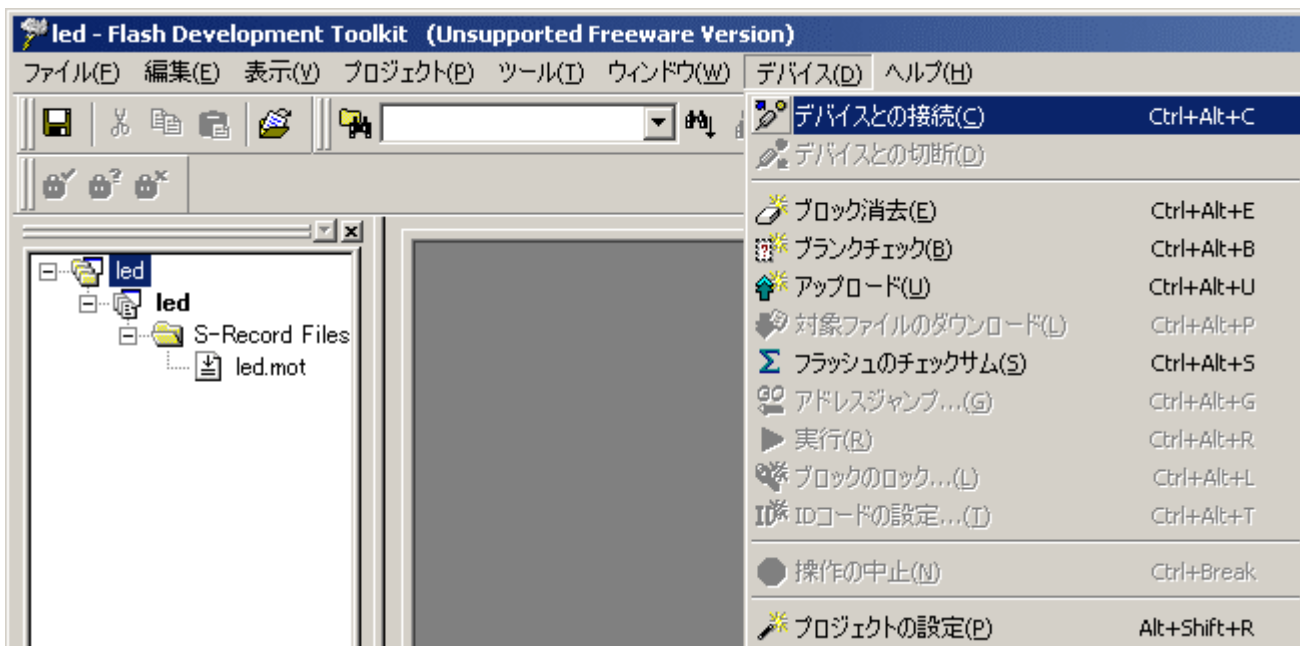
「ファイルの追加」ダイアログが開きます。作成された「led. mot」を選択して「Add」をクリックします。



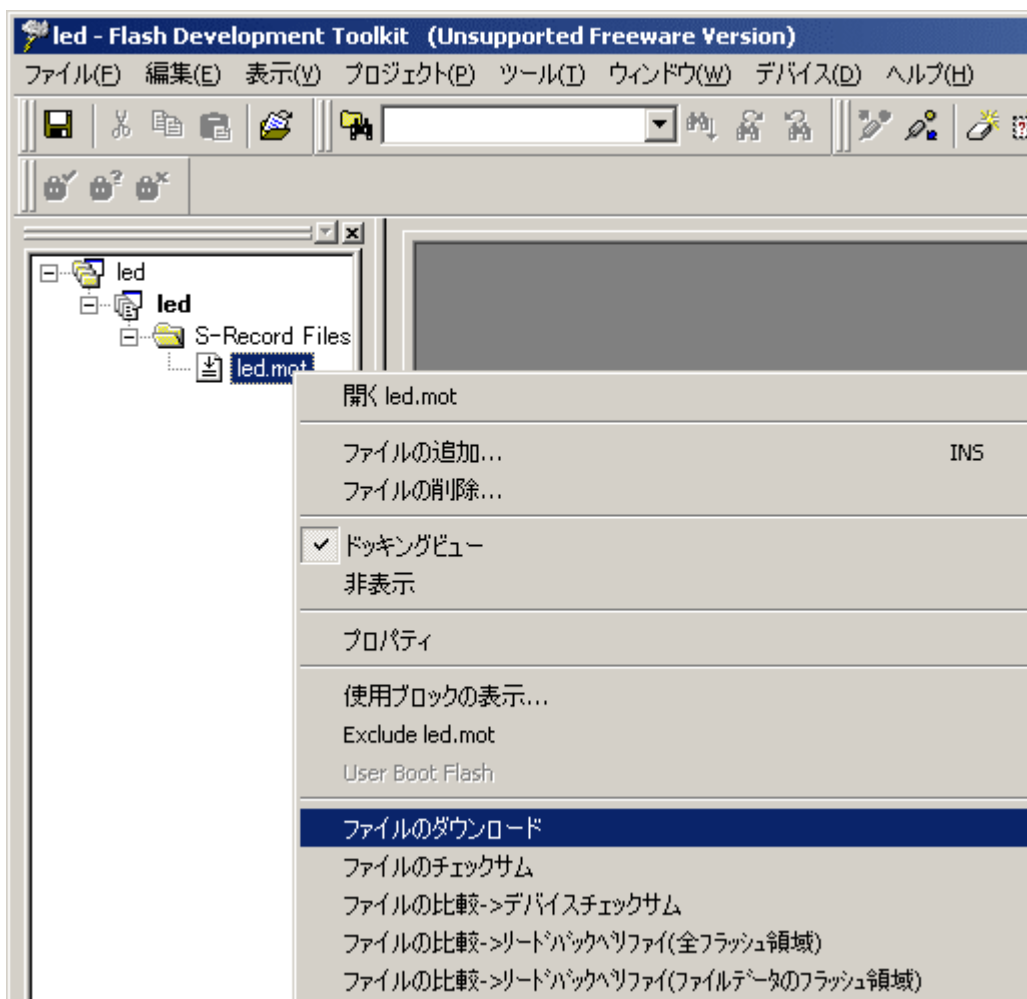
では、ダウンロードしましょう。まず TK-3052 の JP1 の 1-2 と 3-4 を両方ともショートします。そして、TK-3052 のシリアルポート (CN6, または CN7) とパソコンのシリアルポートをストレートケーブルで接続します。それから電源をオンします。



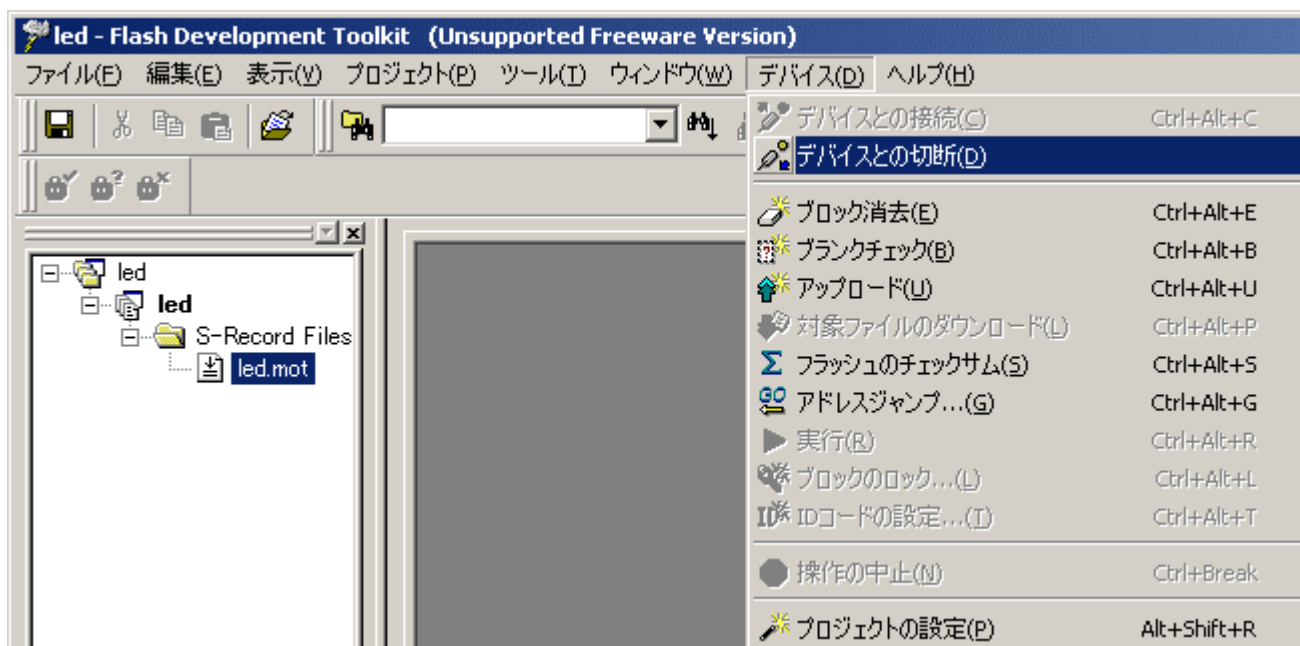
メニューから「デバイス」→「デバイスとの接続」を選択します。「接続が成功しました」と表示されたら接続 OK です。(表示されないときは、ケーブル、ポート番号を確認してください。)



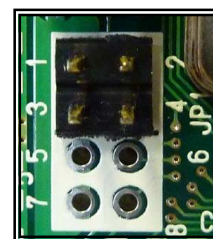
次に、左側のプロジェクトウィンドウの「led. mot」を右クリックします。表示されたメニューから「ファイルのダウンロード」を選択します。「書き込みが完了しました」と表示されたらダウンロード終了です。



最後に、メニューから「デバイス」→「デバイスとの切断」を選択します。「Disconnecting」、「Disconnected」と表示されたら終了です。



では、いったん電源をオフしましょう。JP1 の 1-2 と 3-4 の両方をオープンにしてください。電源をオンするとフラッシュメモリにダウンロードされたプログラムが実行されます。



3. 開発環境について

無償評価版 HEW は前述したように 60 日を過ぎるとサイズ制限が生じます。それで、製品版の HEW を購入することをおすすめします。

ルネサスエレクトロニクスが開催しているセミナーで使用することを目的に作られたモニタプログラムが Web サイトで公開されています。RAM にプログラムをダウンロードするタイプのリモートデバッガです。RAM にダウンロードするため、あまり大きなサイズのプログラムをデバッグすることはできませんが、C 言語によるソースデバッグも可能です。これについては次の第 4 章で説明します。

H8/300H シリーズのデバッガとして有限会社イエローソフトの「YellowScope」があります。RAM にプログラムをダウンロードするタイプのリモートデバッガです。RAM にダウンロードするため、(RAM を外付けしない限り)あまり大きなサイズのプログラムをデバッグすることはできませんが、C ソースデバッグも可能な多機能なソフトです。また、コンパイラやフラッシュメモリへのダウンロードツールも付属しているので、HEW のサイズ制限を気にすることなく、これだけで開発環境が整います。

本格的には ICE を用意することになりますが、数十万円コースとなります。

その他に、フリーソフトも幾つかあるようです。興味のある方は探してみてください。

第4章

モニタプログラム“Hterm”について

1. モニタプログラムのダウンロード
2. 組み込み型モニタのカスタマイズ
3. Hterm の使い方

1. モニタプログラムのダウンロード

ルネサスエレクトロニクスの Web サイトからダウンロードします。詳細は弊社 CD の「__必ずお読みください」フォルダの中にある「ルネサスダウンロード.pdf」をご覧ください。

2. 組み込み型モニタのカスタマイズ

ルネサスエレクトロニクスが配布しているモニタプログラムは、マイコンに実装する「組み込み型モニタ」と、パソコン側通信ソフト「Hterm」で構成されています。「組み込み型モニタ」は使用するマイコンにあわせてカスタマイズします。カスタマイズの詳細については、ダウンロードしたファイルに含まれている「readme.htm」をご覧ください。ここでは TK-3052 用にカスタマイズする方法を説明します。

「組み込み型モニタ」のカスタマイズは、ダウンロードしたソースリストを修正してビルドします。これによって生成された「MONITOR.MOT」を、FDT を使って H8/3052 にダウンロードします。

ここで問題になるのが HEW のサイズ制限です。制限前の HEW や製品版 HEW では問題なくビルドできますが、サイズ制限がかかるとビルドできません。

それで、TK-3052 用にカスタマイズした「MONITOR.MOT」を CD に収録しました。ビルドできない場合はご利用ください。なお、ルネサスエレクトロニクスはモニタプログラムの収録を原則認めていないため、生成された「MONITOR.MOT」のみ収録します。その他のファイルはルネサスエレクトロニクスのサイトからダウンロードしてください。

ダウンロードしたファイルの「monitor.hws」をダブルクリックし HEW を起動します。まず、「monitor.c」を次のように修正します。

```
/*
*****
/*      H8/300H Monitor Program (Advanced Mode)      Ver. 3.0A      */
/*      Copyright (C) 2003 Renesas Technology Corp.      */
*****
#include "3052s.h" /* */
#pragma global_register (monitor=ER5) /* Return Address */
void (*monitor) (void); /* */
*****
/*      User Initialize Module      */
/*      Input  ER5 <-- Return Address      */
/*      Output Nothing      */
/*      Used Stack Area --> 0(0) Byte      */
*****
#pragma noregsave (INITIALIZE) /* Non Register Save */
void INITIALIZE (void) /* */
{ /* */
    monitor (); /* Goto Monitor Program */
} /* */
```

次に、「monitor. sub」を次のように修正します。

```
INPUT      ADVANCED, MONITOR
LIST       MONITOR. MAP
SHOW       SYMBOL
OUTPUT     MONITOR. MOT
FORM       STYPE
CHANGE     INFORMATION=1320
NOOPTIMIZE
LIBRARY    ADVANCED, 3052
LIBRARY    MONITOR
DEFINE     $BRR=13
DEFINE     $STACK=FFFF00
START      VECTOR/0, ROM/100, RAM/FFDF10, USER/FFE000, SCI/FFFFB8
```

修正箇所は以上です。ビルドすると「Release」フォルダに「MONITOR. MOT」が生成されます。このファイルを、FDT を使って H8/3052 にダウンロードしてください。

ちなみに、このカスタマイズによって「MONITOR. MOT」は次のような設定になります。なお、モード 7 のアドレスは 1M バイトモード(00000~FFFFFF)ですが、上のリストのように「monitor. sub」は 16M バイトモード(000000~FFFFFF)で指定しなければなりません。

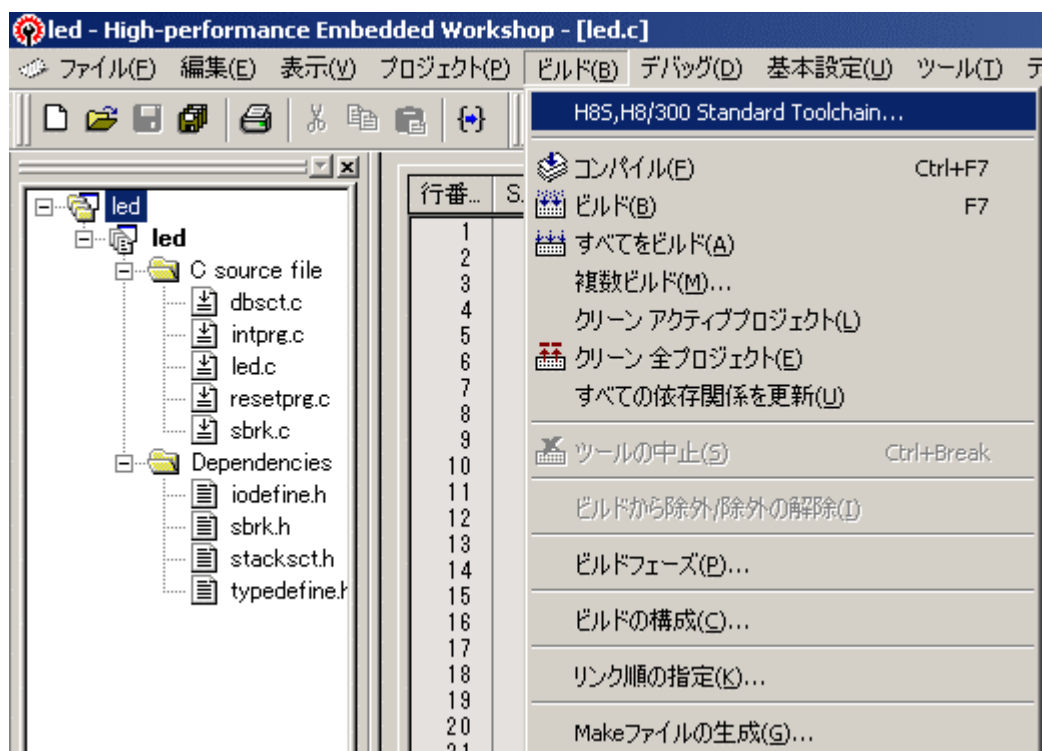
シリアルポート	:	チャンネル 1
ボーレート	:	38400 ボー
動作モード	:	モード7(シングルチップアドバンスモード)
MONITOR VECTOR	:	00000~000FF
MONITOR ROM	:	00100~059EB
MONITOR RAM	:	FDF10~FDFF3
USER VECTOR	:	FE000~FE0FF
USER 使用可能エリア	:	FE100~FFCFF(ここがユーザのプログラムとワークエリア)
USER スタック初期値	:	FFF00(スタックエリアは FFD00~FFEFF の 200h バイト)

3. Hterm の使い方

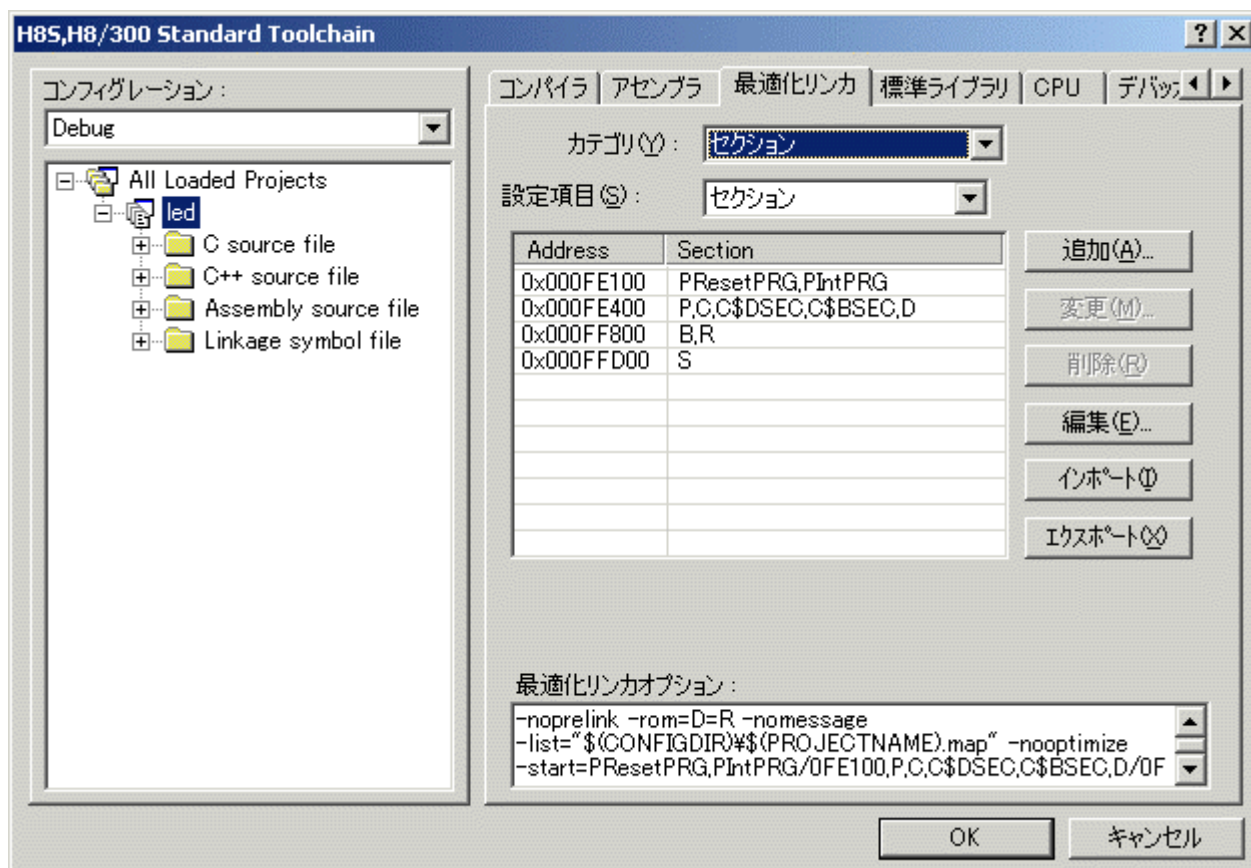
「組込み型モニタ」の機能については、ダウンロードしたファイルに含まれている「monitor. htm」で詳しく説明されています。実のところ「組込み型モニタ」と一般のターミナルソフト(例:ハイパーターミナル)を使えばモニタを使うことはできます。「Hterm」は一般のターミナルソフトと異なり、「組込み型モニタ」をより便利に使うことができるような機能を加えた「組込み型モニタ」専用のターミナルソフトです。

「Hterm」の詳しい使い方は「Hterm」のヘルプファイル「hterm. hlp」をご覧ください。ここでは第 2 章で作成したプログラムを例に、「Hterm」でダウンロード・実行するまでを説明します。

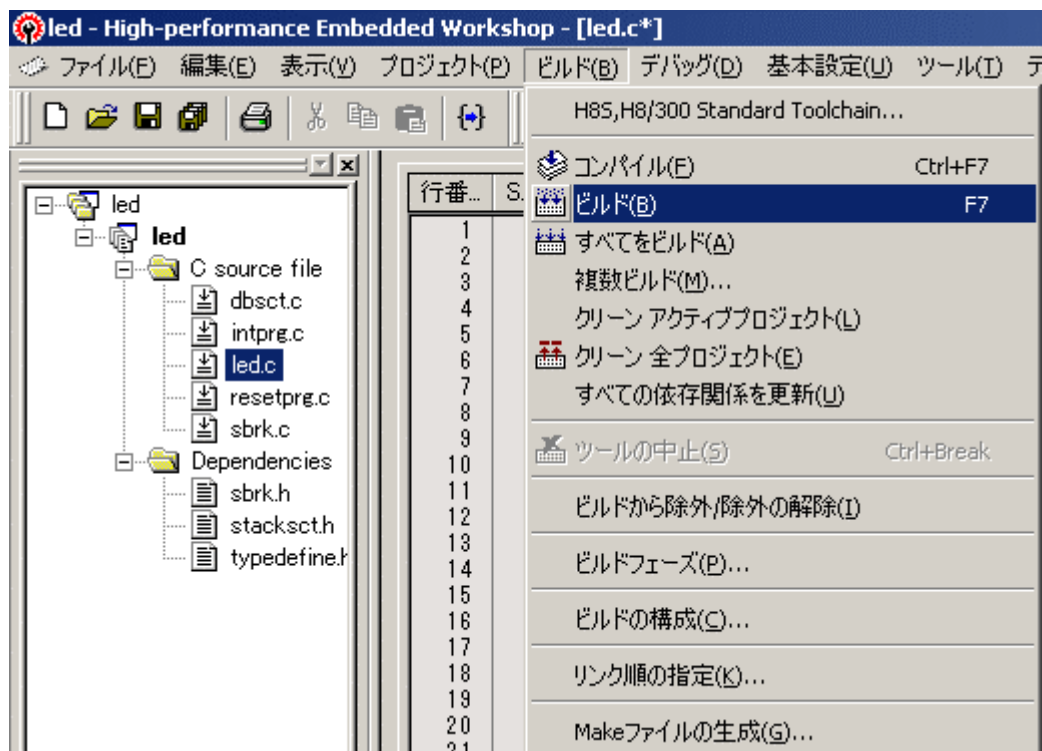
では、第2章で作った「led」プロジェクトを開いてください。第2章ではROM(フラッシュメモリ)のエリアにプログラムがロケーションされました。これをRAMのエリアに変更します。メニューから「ビルド」→「H8S,H8/300 Standard Toolchain...」を選択します。



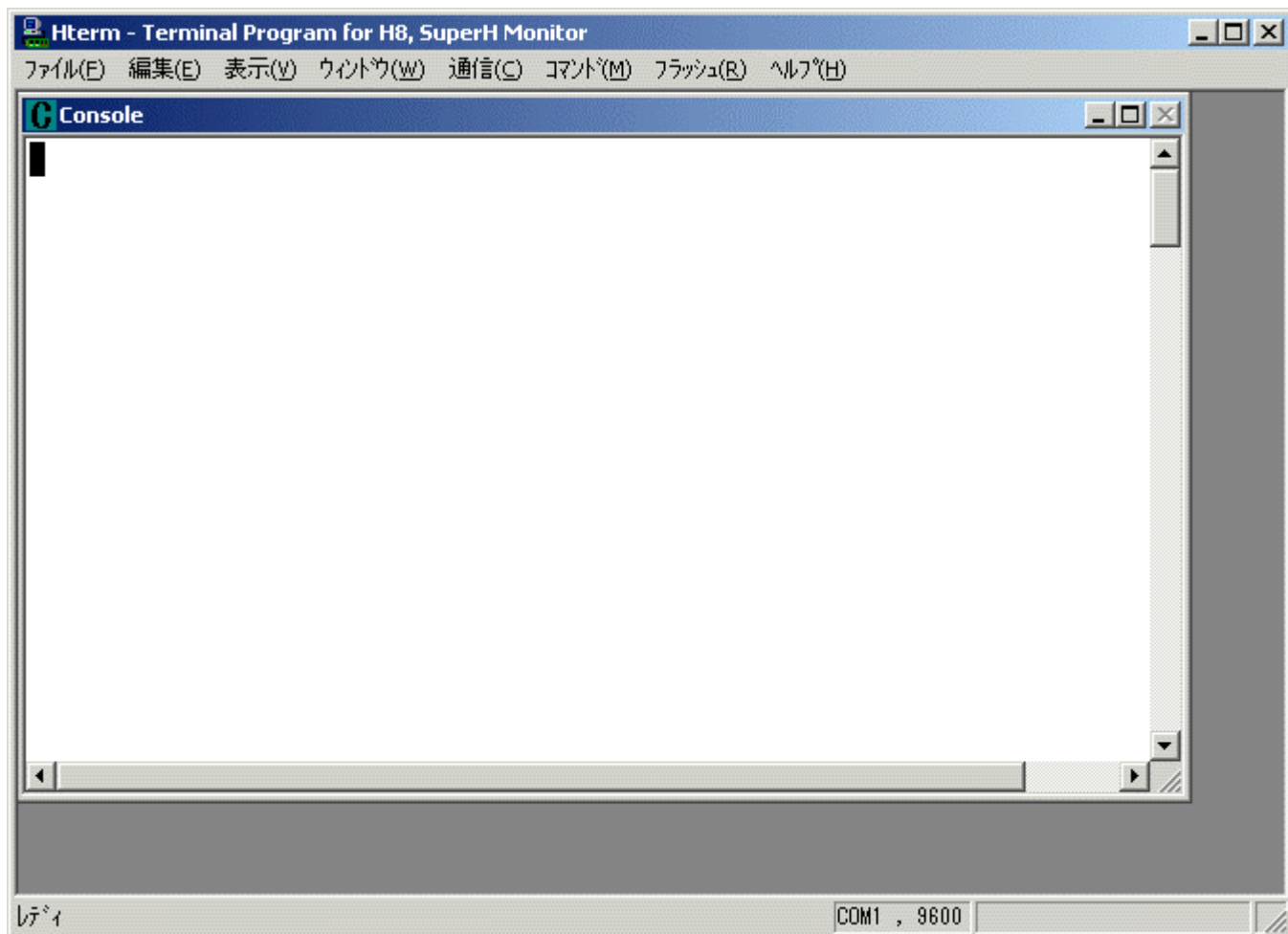
「H8S,H8/300 Standard Toolchain」ダイアログが開きます。「最適化リンカ」タブを選択し、カテゴリは「セクション」を選択します。アドレスを次のように変更してください。変更したら「OK」をクリックします。



あらためてビルドします。メニューから「ビルド」→「ビルド」を選択してください。RAM エリアにロケーションされたファイルが生成されます。



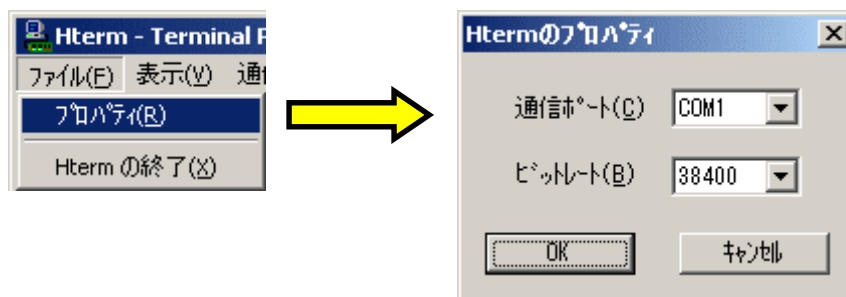
続いて、「Hterm」を起動します。ダウンロードしたファイルから「Hterm. exe」をダブルクリックしてください。



「Hterm」はダウンロードして初めての起動の際は使用するポートは COM1、ボーレートは 9600 ボーに設定されます。このままだと「組込み型モニタ」とつながらないので、メニューから「通信」→「切断」を選択して、いったん通信を切断します。



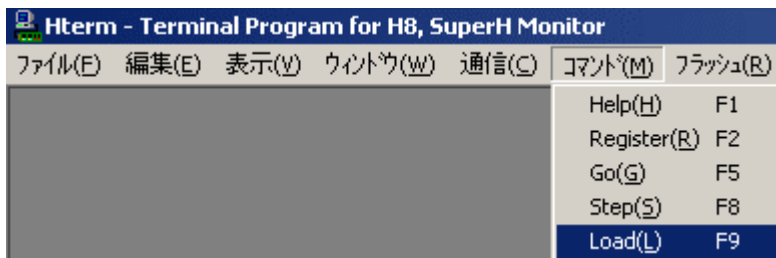
ここであらためて、38400 ボーで通信を開始します。メニューから「ファイル」→「プロパティ」を選択すると、「Htermのプロパティ」ダイアログが開くので、使用するポートとボーレート(ビットレート)を設定して「OK」をクリックします。



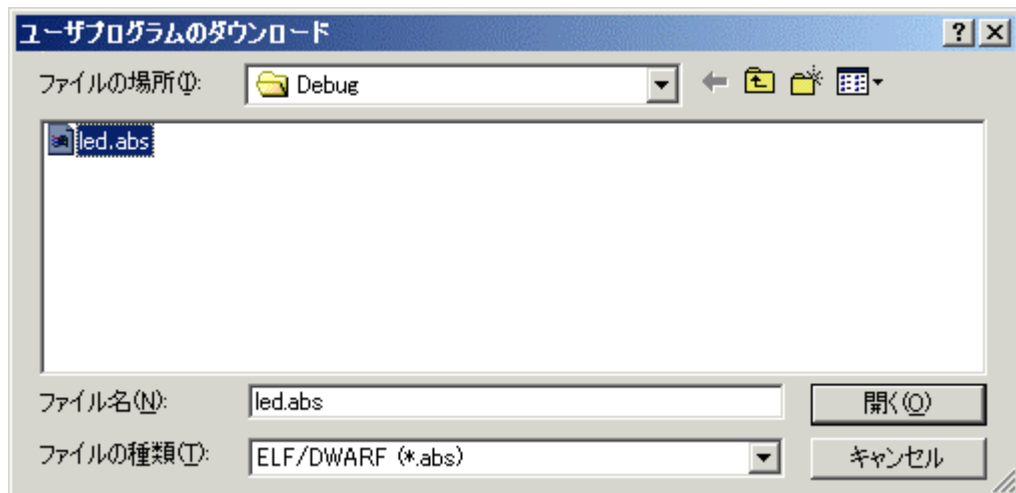
次に、TK-3052 とパソコンをシリアルケーブルでつなぎ電源をオンします。「Hterm」の「Console」ウィンドウに、次のように表示されれば接続完了です。なお、設定値は保存され次の起動からはこの設定でスタートします。



続いてプログラムをダウンロードします。メニューから「コマンド」→「Load」を選択します(ファンクションキー, F9 でも OK)。



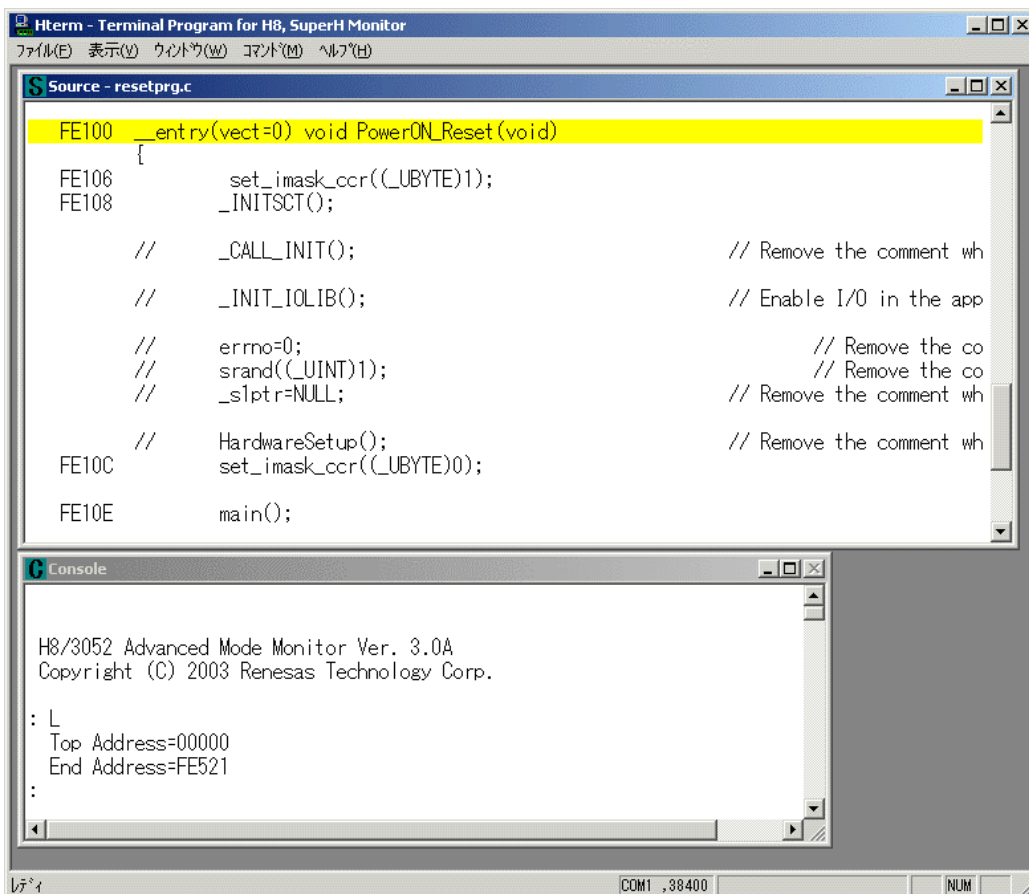
すると、「ユーザプログラムのダウンロード」ダイアログが開きます。「led.abs」を選択し「開く」をクリックします。(注意:「led.mot」ではないことに気をつけてください。MOT ファイルではなく ELF/DWARF ファイルを使います。)



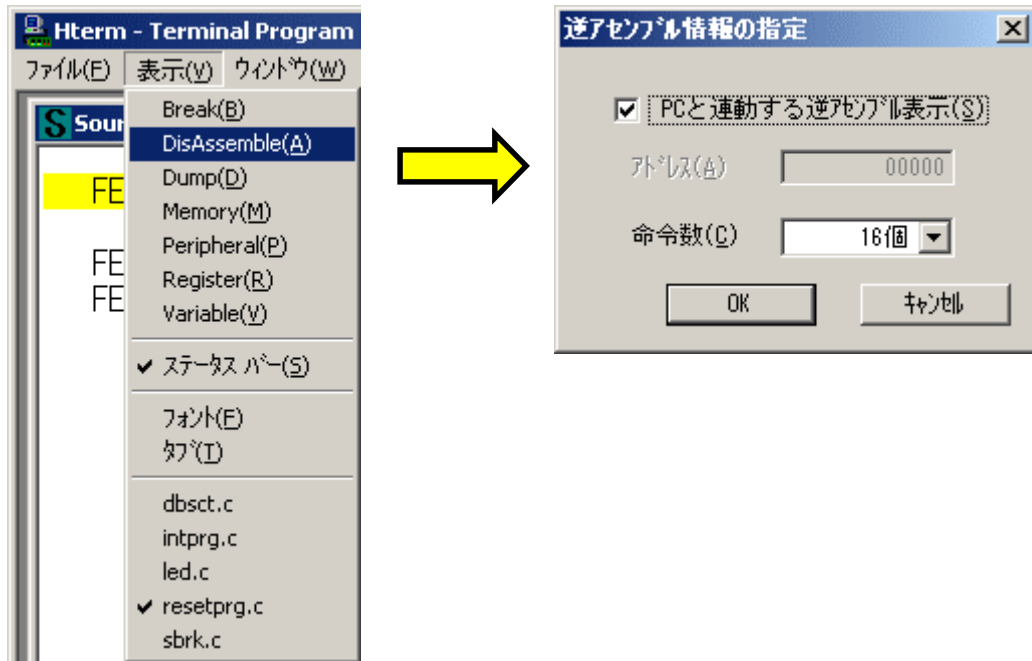
ダウンロードが終了すると、「ソースプログラムを表示しますか?」と尋ねられるので「はい」をクリックします。



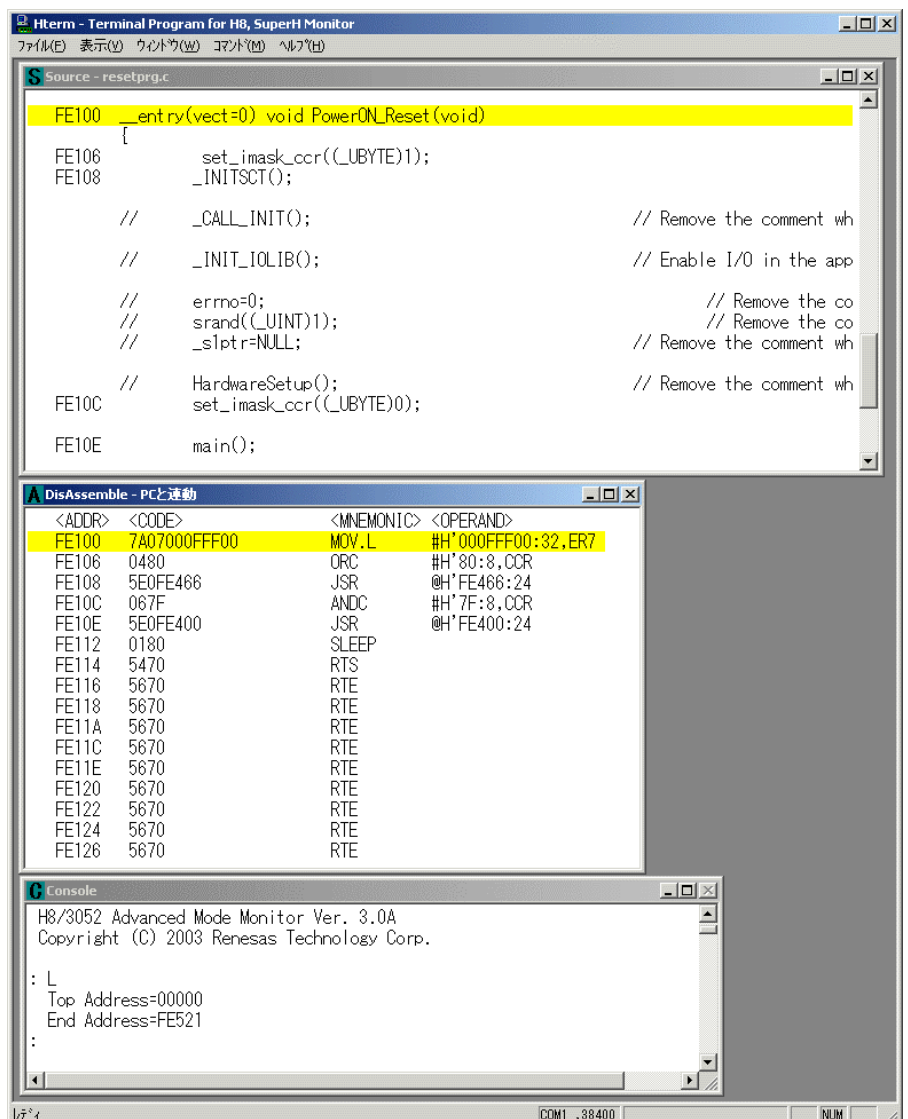
右のようにソースファイルが表示されダウンロード完了です。なお、黄色でマークされている行は、プログラムカウンタが指しているアドレスです。



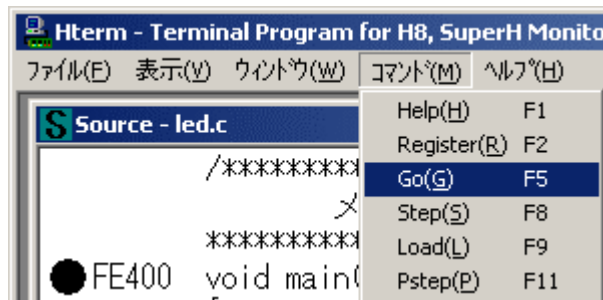
「Hterm」はソースプログラム上でデバッグできますが、マシン語レベルでもデバッグできるモニタです(例:通常のステップ実行は、C言語で1行ずつステップ実行するのではなく、マシン語レベルで1行ずつステップ実行する、C言語で1行ずつステップ実行するPstepが別に用意されている)。それで、ソースファイルの表示と併せて、逆アセンブルしたリストも表示しておくくと便利です。メニューから「表示」→「DisAssemble」を選択します。すると、「逆アセンブル情報の指定」ダイアログが開きます。「PCと連動する逆アセンブル表示」にチェックを入れて「OK」をクリックします。



これで、ソースファイルと併せて、逆アセンブルのリストも表示されました。C言語がどのようにマシン語に変換されているかも理解できます。

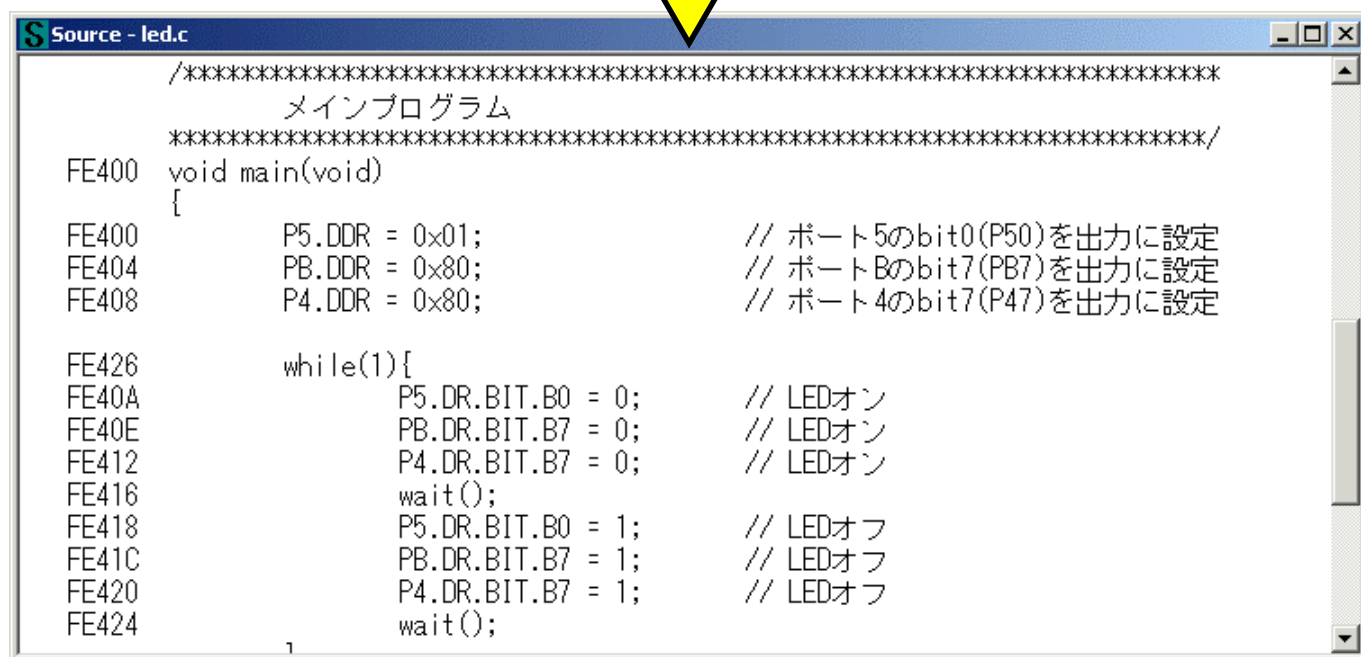
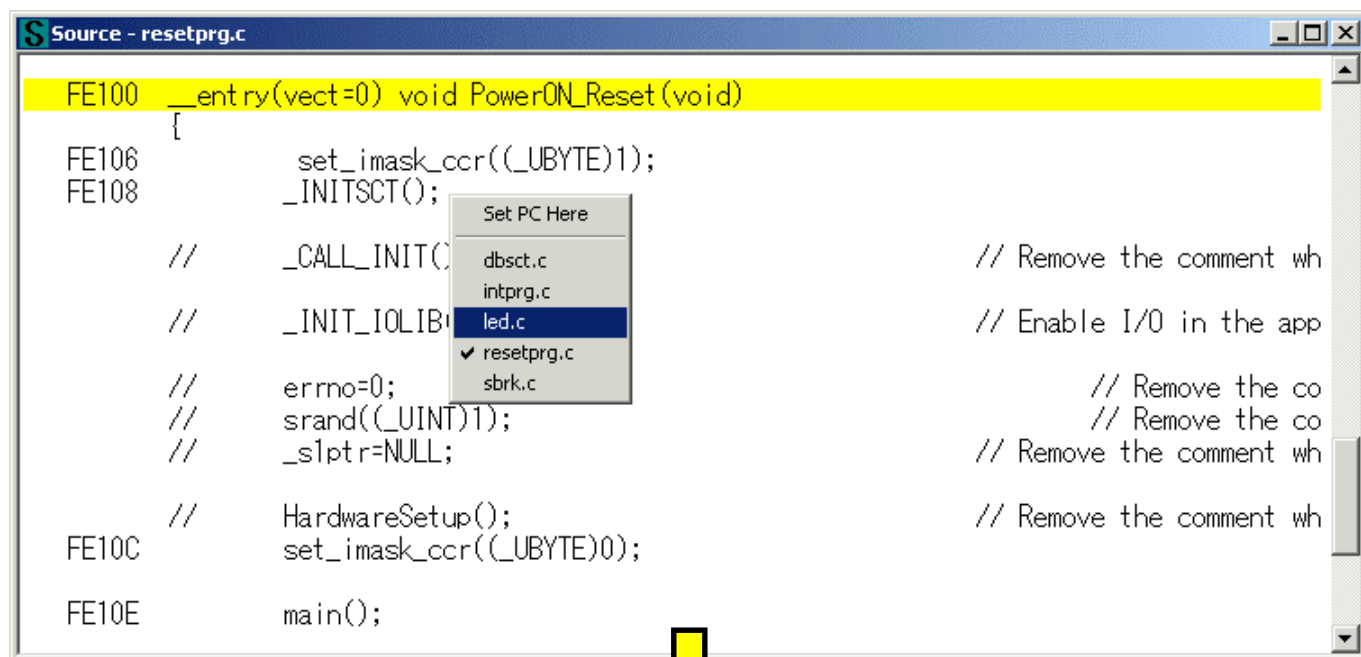


ダウンロードしたプログラムを実行してみましょう。ダウンロードした時点でプログラムカウンタはスタートアドレスを指しています。メニューから「コマンド」→「Go」を選択します(ファンクションキー, F5 でも OK)。第 2 章と同じように動作します。なお, 終了するときは TK-3052 のリセットスイッチをオンします。リセットするとプログラムカウンタは 0 にクリアされます。



「Hterm」の「Go」はプログラムカウンタや任意のアドレスから実行する機能だけで, 任意のアドレスで停止するにはブレークを張る必要があります。では, ダウンロードした後スタートアドレスから実行し, main 関数の先頭でブレークしてみましょう。

まず, メニューから「コマンド」→「Load」を選択して(ファンクションキー, F9 でも OK)プログラムをダウンロードします。ソースプログラムと逆アセンブルのリストも表示してください。ソースプログラムは「resetprg. c」が表示されています。main 関数は「led. c」の中にあるので, ソースプログラムの表示を切り替えます。ソースプログラムのウィンドウの上で右クリックしてください。下のようメニューが表示されますので「led. c」を選択してください。



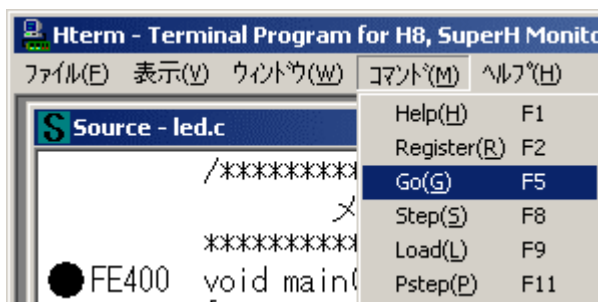
次にブレイクを張ります。ブレイクを張りたい行にマウスカーソルをあわせてダブルクリックしてください。

```
Source - led.c
/*****
   メインプログラム
 *****/
void main(void)
{
    P5.DDR = 0x01;           // ポート5のbit0(P50)を出力に設定
    PB.DDR = 0x80;         // ポートBのbit7(PB7)を出力に設定
    P4.DDR = 0x80;         // ポート4のbit7(P47)を出力に設定

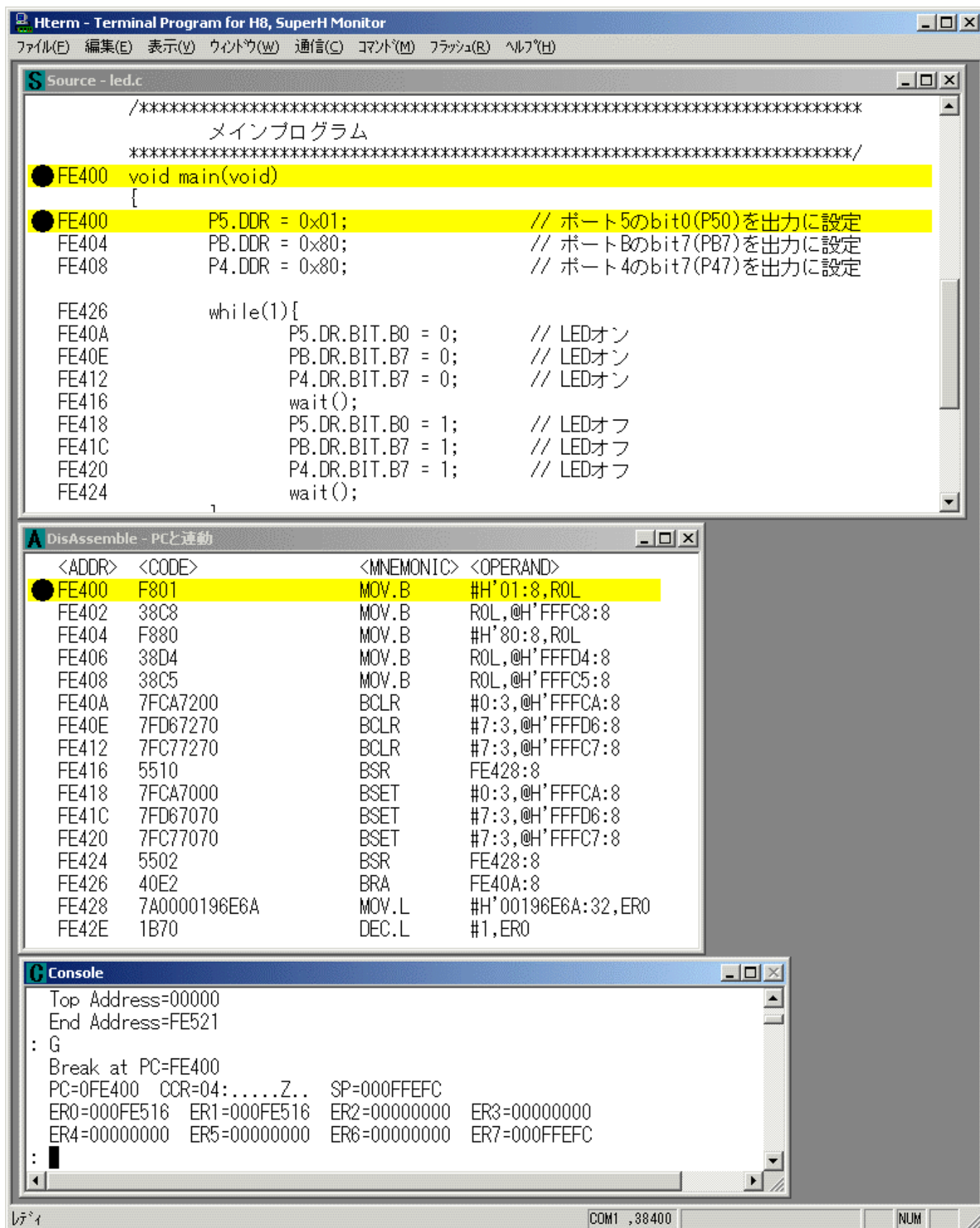
    while(1){
        P5.DR.BIT.B0 = 0;   // LEDオン
        PB.DR.BIT.B7 = 0;   // LEDオン
        P4.DR.BIT.B7 = 0;   // LEDオン
        wait();
        P5.DR.BIT.B0 = 1;   // LEDオフ
        PB.DR.BIT.B7 = 1;   // LEDオフ
        P4.DR.BIT.B7 = 1;   // LEDオフ
        wait();
    }
}
```

上図では「●」が2箇所に表示されていますが、main関数の入り口のアドレスとmain関数の1行目のアドレスが同じ値であるためです。

では、プログラムを実行してみましょう。ダウンロードした時点でプログラムカウンタはスタートアドレスを指しています。メニューから「コマンド」→「Go」を選択します(ファンクションキー、F5でもOK)。

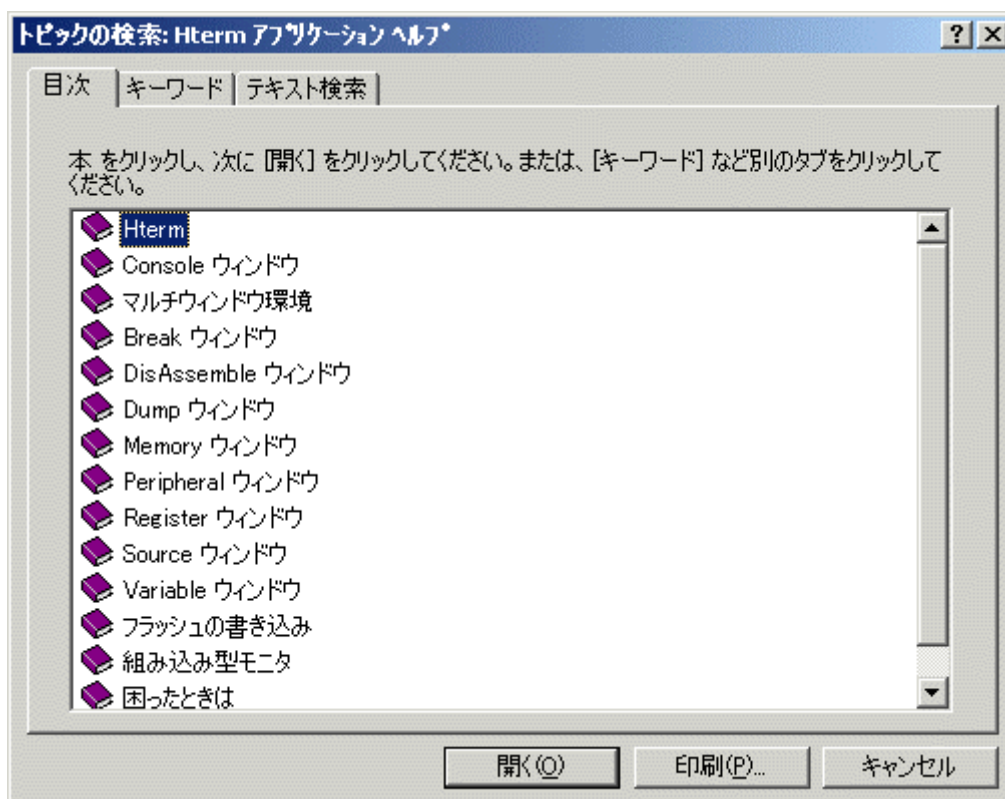


すると、下のよう設定したブレイクアドレスで停止します。



あとは、メニューから「コマンド」→「Step」でステップ実行したり(ファンクションキー, F8 でも OK), 別の行にブレイクを張ってメニューから「コマンド」→「Go」で実行したり(ファンクションキー, F5 でも OK)して、デバッグを行います。

「Hterm」には、ほかにもレジスタの表示と変更、メモリの表示と変更、I/O の表示と変更など、デバッグに便利ないろいろな機能が含まれています。ヘルプファイルを参考にいろいろ試してみてください。(下図は「Hterm」のヘルプのメニューです。)



第5章

μITRON(HOS)の実装

1. 必要なものを用意する
2. HOS を実装する

パソコンのプログラムは Windows や Linux など OS の上で実行されます。一方、組み込みシステムのプログラムにおいては OS を実装せずにプログラムすることが多かったように感じます。しかし、近年の組み込みシステムの高機能化や大規模化、ネットワーク対応や GUI の搭載などにより、システムを効率よく制御するため OS の必要性が高まってきました。ITRON は社団法人トロン協会が推進し、日本の半導体メーカー、ソフト開発メーカーが参画したトロンプロジェクトの中で作成された組み込みシステム向け汎用リアルタイム OS の標準仕様です。

ところで、μITRON を手に入れようとしても、μITRON という名前の OS はありません。トロンプロジェクトでは、ITRON の仕様書は配布していますが、実際に動くコードは配布していないからです。企業や団体、個人が、配布されている仕様書を元にコードを書き、「μITRON 仕様準拠 OS」として配布、販売しています。このマニュアルで利用する「HOS」もプロジェクト HOS が開発した μITRON 仕様のフリーの OS です。（「μITRON 仕様準拠している」と言うために μITRON の全ての仕様を持たせる必要はありません。必要な最低限の機能も仕様書に規定されています。）

「HOS」は「Hyper Operating System」の略です。（ちなみに、90 年代半ばに連載され、OVA やテレビアニメ、映画にもなったコミック「起動警察パトレイバー」に登場するレイバーと呼ばれる作業用ロボットに搭載されている OS の名称が「HOS」で、これにちなんで名付けられたそうです。）「HOS」は高い移植性を持っていて、H8 だけではなく、SH や ARM, IA32 にも移植されています。ライセンス条項に従う限り、商用、非商用に関係なく、自由に利用し再配布することができます。（ライセンス条項はプロジェクト HOS の配布ファイル「licence.txt」を参照、元の著作権情報を削除しない、著作者は利用した損害の責任はとらない、著作者はサポートの義務は負わない、など）

実際のところ、H8 程度の規模のマイコンであれば OS を実装しなくてもほとんどのプログラムはできてしまいます。プログラムの開発効率もそれほど変わりません。それでも、リアルタイム OS を使うと、これまでとは違った感覚でプログラムすることができます。最初の目標は、プロジェクト HOS が配布している H8/3048(16MHz)用のサンプルプログラムを TK-3052(つまり H8/3052, 25MHz)で動かすことです。

μITRON4.0 仕様準拠ソフトウェアの製品マニュアルに入れるよう強く推奨されている文があります。それに従い、ここで入れておきます。

- TRON は“The Real-time Operating system Nucleus”の略称です。
- ITRON は“Industrial TRON”の略称です。
- μITRON は“Micro Industrial TRON”の略称です。
- TRON, ITRON, および μITRON は、特定の商品ないしは商品群を指す名称ではありません。

μITRON4.0 仕様準拠ソフトウェアの製品マニュアルに入れるよう推奨されている文があります。それに従い、ここで入れておきます。

- μITRON4.0 仕様は、トロン協会が定めたオープンなリアルタイムカーネル仕様です。μITRON4.0 仕様の仕様書は、トロン協会のホームページ(<http://www.assoc.tron.org/>)から入手することができます。

1. 必要なものを用意する

HOS のソースコード

プロジェクト HOS のサイト (<https://sourceforge.jp/projects/hos/>) から「hos-v4」をダウンロードしてください。HOS-V4 の中にも圧縮形式の異なるファイルがあるのですが、Windows の場合は「HOS-V4<バージョンナンバー>LZH」をダウンロードします。このマニュアル執筆時点のファイル名は「h4_102.lzh」でした。

μITRON の仕様書

トロンプロジェクトのサイト (<http://www.assoc.tron.org/itron/home-j.html>) から「μITRON4.0 仕様」というドキュメントをダウンロードしてください。

参考書

「μITRON プログラミング入門—H8 マイコンと HOS で始める組み込み開発—」(著者 濱原和明, オーム社, 2005 年発行)。これから、この本の内容に基づいて HOS を実装します。以下、「参考書」とはこの本を指します。

2. HOS を実装する

TK-3052 とプロジェクト HOS のサンプルプログラムの違いをまとめておきます。(実装に影響する部分のみ)

	TK-3052	プロジェクト HOS のサンプルプログラム
CPU	H8/3052BF	H8/3048
クロック	25MHz	16MHz
フラッシュメモリ	512K バイト	128K バイト
RAM	8K バイト	4K バイト

この違いを念頭におきつつ実装します。最初に、参考書の 287 ページから 300 ページまでの「A. 2 HEW による HOS-V4 カーネルライブラリの構築」、「A. 3 HEW にプロジェクトを登録する」、「A. 4 HEW でコンフィギュレータを使用する(カスタムビルドフェーズ)」、「A. 5 HOS-V4 のアプリケーションをビルドするまで」の作業を順に行なってください。参考書では H8/3048(H8/300H)をメインに、H8/3664(H8/Tiny)を補足的に説明していますが、当然 H8/3048(H8/300H)の部分当てはめします。「A. 5. 3 セクションの宣言」でセクションの割付を指定しますが、この部分は H8/3052BF にあわせて次のように設定します。



続いて、参考書の 19 ページから 33 ページの「2. 1 サンプルプログラムを動かそう」の作業を行ないます。TK-3052 にあわせて変更する部分は「2. 1. 2 ターゲット別カスタマイズ」の部分です。

22 ページに「ostimer. c」の修正について説明されていますが TK-3052 の場合、次のように変更します。

```
/* OSタイマの初期化 */
void OsTimer_Initialize(VP_INT exinf)
{
    /* ITU0 (システムタイマ) 初期化 */
    TCRO = 0x23; /* φ/8でカウント */
    TSRO &= 0xfe; /* ステータスレジスタクリア */
    TIERO = 0x01; /* 割り込み許可 */
    GRAO = 3125 - 1; /* 1 msec のインターバル (25MHz時) */
    TSTR |= 0x01; /* TCNT0動作開始 */
}
```

24 ページに「h8_sci. h」の修正について説明されていますが、TK-3052 の場合、次のように変更します。

```
/* SCI通信速度定義 (25MHzのとき) */
#define SCI_4800 162
#define SCI_9600 80
#define SCI_19200 40
#define SCI_31250 24
#define SCI_38400 19
#define SCI_57600 13 /* 誤差 -3.1% ぎりぎりいけるかも (^); */
```

さて、30 ページまで進むと、ビルドしたプログラムをフラッシュメモリにダウンロードします。参考書どおり「Hterm」を使用することもできますし、FDTを使うこともできます。FDTの使い方については、本マニュアルの第3章をご覧ください。

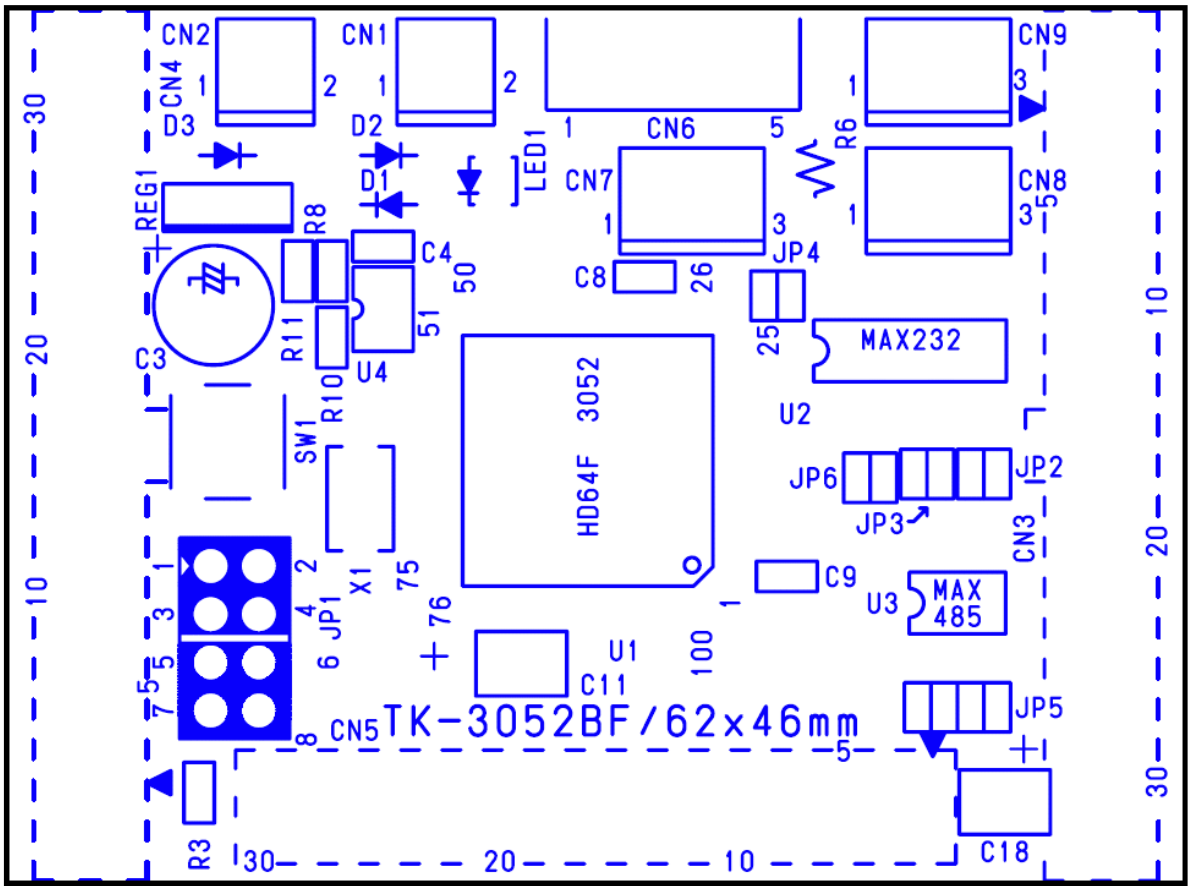
次に参考書の 33 ページでプログラムを実行しています。表示するターミナルソフトに「Hterm」を使っています。ほかにも、ターミナルソフトとして WindowsXP までは標準搭載されていた「ハイパーターミナル」を、それ以降は「TeraTerm」などのターミナルソフトを使うこともできます。



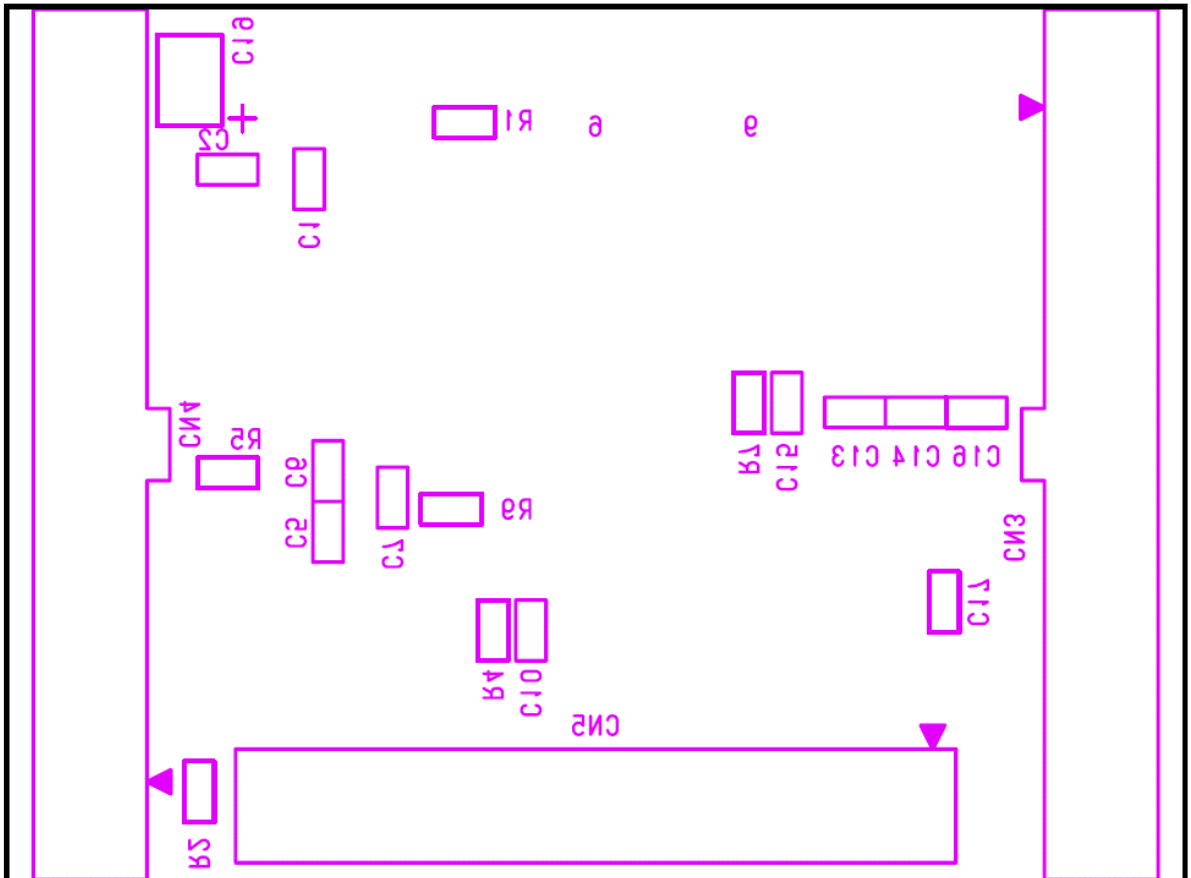
いかがでしょうか。無事に実装できましたか。ぜひ参考書の残りの部分にも挑戦してみてください。

付録

部品面



半田面



株式会社 東洋リンクス

ユーザサポート係(10:00～17:00, 土日祭は除く)

〒102-0093 東京都千代田区平河町 1-2-2, 朝日ビル

TEL:03-3234-0559 / FAX:03-3234-0549

E-mail : toyolinx@va. u-netsurf. jp

<http://www2. u-netsurf. ne. jp/~toyolinx>

- ★ 本書の内容は将来予告なしに変更することがあります。(2010年4月作成)
- ★ 本書の著作権は(株)東洋リンクスが所持しています。
- ★ 万一、不足部品や破損部品があった場合は(株)東洋リンクスまでお問い合わせください。
- ★ 掲載された回路、プログラム等を利用した結果生じたトラブルについて弊社は責任を負いかねますので、あらかじめご了承ください。
- ★ 本書で用いている固有名詞は一般に各メーカーの商標、もしくは登録商標です。