

ユーザーズ・マニュアル

TK-78K0R/KE3L+USB

テセラ・テクノロジー株式会社
2010年4月
Rev 3.00

Windows および Windows XP は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

- ・本資料の内容は予告なく変更することがあります。
- ・文書による当社の承諾なしに本資料の転載複製を禁じます。
- ・本資料に記載された製品の使用もしくは本資料に記載の情報の使用に際して、当社は当社もしくは第三者の知的財産権その他の権利に対する保証または実施権の許諾を行うものではありません。上記使用に起因する第三者所有の権利にかかわる問題が発生した場合、当社はその責を負うものではありませんのでご了承ください。
- ・本資料に記載された回路、ソフトウェア、及びこれらに付随する情報は、半導体製品の動作例、応用例を説明するためのものです。従って、これら回路・ソフトウェア・情報をお客様の機器に使用される場合には、お客様の責任において機器設計をしてください。これらの使用に起因するお客様もしくは第三者の損害に対して、当社は一切その責を負いません。

取り扱いに関する注意事項

- ・落下など強い衝撃を与えないでください。
- ・過電圧の印加は行なわないでください。
- ・高温(40度以上)や低温(0度以下)外での使用は行なわないでください。
- ・USB ケーブルの接続が不十分な状態での使用は行なわないでください。
- ・USB ケーブルに過度の曲げ、引っ張りを与えないでください。
- ・濡らさないでください。
- ・感電には十分注意をしてください。
- ・本製品は CMOS 半導体と同様の取り扱いをしてください。使用者は本製品を使用する間、静電気を蓄積しないように注意してください。
- ・作業台を含む測定機器類はすべてのテストにおいて、アースを施してください。
- ・使用者はリストストラップなどを使用して静電気を逃がしてください。
- ・コネクタとデバイスピンは素手でさわらないでください。

目次

はじめに.....	5
第 1 章 準備編.....	7
1.1 使用する開発ツール/ソフトウェア.....	8
1.2 開発ツールのインストール.....	10
1.2.1 インストーラの構成.....	10
1.2.2 開発環境のインストール.....	10
1.3 USBドライバのインストール.....	15
1.3.1 Windows XP へのインストール.....	16
1.3.2 Windows2000 へのインストール.....	19
1.3.3 USBドライバのインストール完了確認.....	22
1.4 サンプル・プログラム.....	23
1.4.1 サンプル・プログラムの準備.....	23
1.4.2 サンプル・プログラムのフォルダ構成と概要.....	26
第 2 章 体験編.....	27
2.1 PM+の起動.....	29
2.2 PM+の紹介.....	30
2.3 ワークスペース(プロジェクト)の読み込み.....	32
2.4 リンカオプションの設定確認.....	34
2.4.1 「出力 1」タブ.....	34
2.4.2 「出力 2」タブ.....	36
2.5 コンパイラオプションの設定確認.....	37
2.5.1 「機能拡張」タブ.....	37
2.5.2 「スタートアップ・ルーチン」タブ.....	38
2.6 ロード・モジュール・ファイルの作成.....	39
2.7 デバッガの設定確認.....	41
2.8 キットの設定確認.....	43
2.9 デバッガ(ID78K0R-QB)の起動.....	44
2.10 プログラムの実行.....	47
2.11 プログラムの停止.....	49
2.12 デバッガ(ID78K0R-QB)の終了.....	50
2.13 PM+の終了.....	51
第 3 章 ハードウェア資料編.....	52
3.1 部品配置図.....	53
3.2 テスト用パッド配置図.....	53
3.3 各部品の説明.....	54
3.3.1 SW1、SW4.....	54
3.3.2 SW2.....	55
3.3.3 SW3 (RESET SW).....	55
3.3.4 SW5 (Filter).....	56
3.3.5 JP1.....	56

3.3.6 JP2、JP3.....	56
3.3.7 JP4.....	56
3.3.8 LED1 (POWER).....	57
3.3.9 U1,U2 (7Seg-LED).....	57
3.3.10 CN1, CN2.....	58
3.3.11 CN3.....	58
3.3.12 J1.....	58
3.3.13 VR1.....	58
3.3.14 USB1.....	58
3.3.15 USB2.....	58
3.3.16 FP1.....	58
3.4 半田ショートパッド.....	59
第4章 困った時は？.....	60
4.1 パソコンとキットを接続時、USBドライバが見つからない.....	60
4.2 デバッガを起動すると、エラーが出る.....	60
4.2.1 「インサーキット・エミュレータと通信できません。・・・」(F0100 or A0109).....	61
4.2.2 「ID コードが間違っています。」(Ff603).....	62
4.2.3 「このデバイスではオンチップ・デバッグが禁止されています。」(F0c79).....	62
4.2.4 「オンチップ・デバッグを使用禁止にしようとした。」(F0c33).....	62
第5章 その他.....	63
5.1 ワークスペースの新規作成方法.....	64
5.2 ソース・ファイルの追加登録方法.....	69
5.3 デバッガの便利な機能.....	71
5.3.1 わかりやすいボタン表示に変更したい.....	71
5.3.2 ソース一覧や関数一覧を表示したい.....	71
5.3.3 ブレーク・ポイントの設定/解除を行ないたい.....	72
5.3.4 グローバル変数の値を表示したい.....	73
5.3.5 グローバル変数の値をプログラム実行中でも表示したい.....	74
5.3.6 ローカル変数の値を表示したい.....	75
5.3.7 メモリの内容や SFR の値を表示したい.....	75
5.4 マイコン内蔵のフラッシュ・メモリを消去したい.....	76
5.5 回路図.....	81

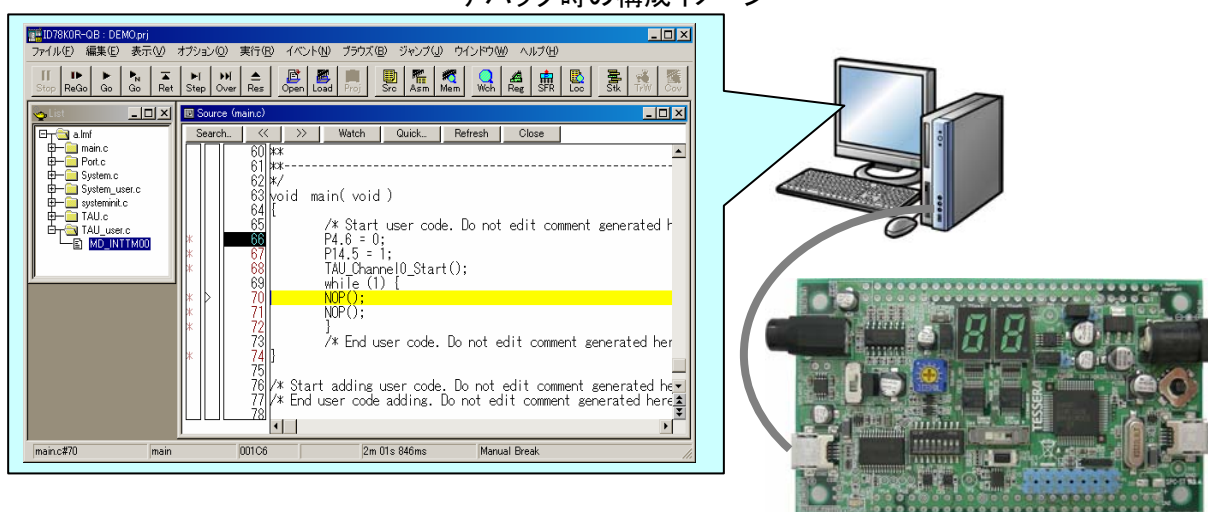
はじめに

TK-78K0R/KE3L+USB は NEC エレクトロニクス製 16bit All Flash マイコン「78K0R/KE3-L」を使用した応用システム開発を体験できるキットです。

開発ツールや USB ドライバなどをパソコンにインストールし、パソコンと本キットを USB 接続するだけでプログラム作成～ビルド～デバッグ～動作確認といった一連の開発フローを体験できます。

(本キットでは、マイコンの持つオンチップ・デバッグ機能を使用して、エミュレータを接続しない状態でのデバッグを実現しています。)

デバッグ時の構成イメージ



構成 この資料は、次の内容で構成しています。
はじめに第 1 章および第 2 章をお読み下さい。使用する開発ツールのインストール、およびサンプル・プログラムの動作確認が体験できます。
第 3 章以降は、サンプル・プログラムのカスタマイズ、ハードウェアのカスタマイズ用途などに利用してください。

第 1 章 準備編

おもにインストール方法に関する説明です。

第 2 章 体験編

統合開発環境(PM+)、統合デバッガ(ID78K0R-QB)の基本的な操作方法を、サンプル・プログラムを使用しながら体験できます。

第 3 章 ハードウェア資料編

TK-78K0R/KE3L+USB のハードウェアの構成などについて解説しています。

第 4 章 困った時は？

統合デバッガ(ID78K0R-QB)起動時のエラーなど、TK-78K0R/KE3L+USB を使用することで、問題があった場合の対処方法を記載しています。

第 5 章 その他

統合開発環境(PM+)のワークスペース(プロジェクト)の新規作成方法、ソース・ファイルの追加登録方法、統合デバッガの便利な機能などについて解説します。回路図もこの章に記載しています。

対象者 この資料は、78K0R 用開発ツールを初めて使用されるお客様を対象としています。
なお、使用するにあたってマイクロコンピュータ、C 言語、アセンブラの一般知識と Windows™ の操作方法に関する基礎知識を必要とします。

目的 この資料は、TK-78K0R/KE3L+USB の特徴やハードウェア構成、添付されている 78K0R 用開発ツールの基本的な操作方法をお客様に理解していただくことを目的としています。
資料を読みながら実際に開発ツールを使用することにより、お客様の理解をより深めていただけます。

第1章 準備編

この章では、以下について説明します。

- ・使用する開発ツールの概要
- ・使用する開発ツールのインストール方法
- ・サンプル・プログラムの準備方法と概要

TK-78K0R/KE3L+USB に含まれる開発ツールを使用するだけで、プログラム作成～ビルド～デバッグ～動作確認といった一連の開発フローを体験できます。

1.1 使用する開発ツール/ソフトウェア

- **デバイス・ファイル DF781026 V1.00**
デバイス固有の情報は、デバイス・ファイルに入っているため、開発ツールを使用するには、デバイス・ファイルが必要となります。
- **統合開発環境 PM+ V6.31**
Windows 上での統合開発環境です。
編集ウインドウとしてアイデアプロセッサ機能付きエディタを搭載し、コンパイラ、デバッガなどの開発ツールと連携して効率的な開発が行えます。
- **C コンパイラ CG78K0R W2.12 (サイズ限定版)**
78K0R 用の「オブジェクト・サイズ限定(64KByte)」C コンパイラです。
78K0R 用 C 言語、または ANSI-C 準拠の C 言語で書かれたソース・プログラムをアセンブリ言語に翻訳するツールです。
C 言語で書かれたソース・プログラムを入力し、78K0R 用アセンブラの入力となるソース・プログラムと、リンカの入力となるオブジェクト・プログラムを出力します。
- **アセンブラ RA78K0R W1.33 (サイズ限定版)**
78K0R 用の「オブジェクト・サイズ限定(64KByte)」アセンブラです。
78K0R 用アセンブリ言語で書かれたアセンブラ・ソース・プログラムを入力し、機械語のオブジェクト・プログラムを出力するツールです。出力したオブジェクト・プログラムは、デバッガの入力となります。
- **78K0R 統合デバッガ ID78K0R-QB V3.60**
C コンパイラ、アセンブラから出力されたオブジェクト・プログラムをデバッグするためのツールです。C ソース・レベルでのデバッグを実現し、変数の参照・変更やソース行単位でのステップ実行など、ソース・デバッグを簡単かつ効率的に行うことができます。
- **Starter Kit USB Driver**
TK-78K0R/KE3L+USB とパソコンを USB ケーブルで接続する際に必要な USB ドライバです。
- **HID クラスサンプルプログラム/USB・ヒューマン・インタフェース・デバイス・クラス用サンプル・ドライバ**
78K0R/KE3-L が HID デバイスとして認識され、キーボードとして動作するサンプルプログラムです。
第 2 章で使用します。
詳細は「USB HID(ヒューマン・インタフェース・デバイス)クラス ドライバ編 アプリケーション・ノート」をご覧ください。
- **CDC クラスサンプルプログラム / USB コミュニケーション・デバイス・クラス用サンプル・ドライバ**

78K0R/KE3-L が CDC デバイス(NEC Electronics K0R Virtual UART)として認識され、COM ポートとして動作するサンプルプログラムです。

詳細は「USB CDC(コミュニケーション・デバイス・クラス)ドライバ編 アプリケーション・ノート」をご覧ください。

●音声出力サンプルプログラム

音声出力サンプルプログラムです。

詳細は「ADPCM-SP 音声圧縮／伸長ソフトウェア・パッケージ アプリケーション・ノート」をご覧ください。

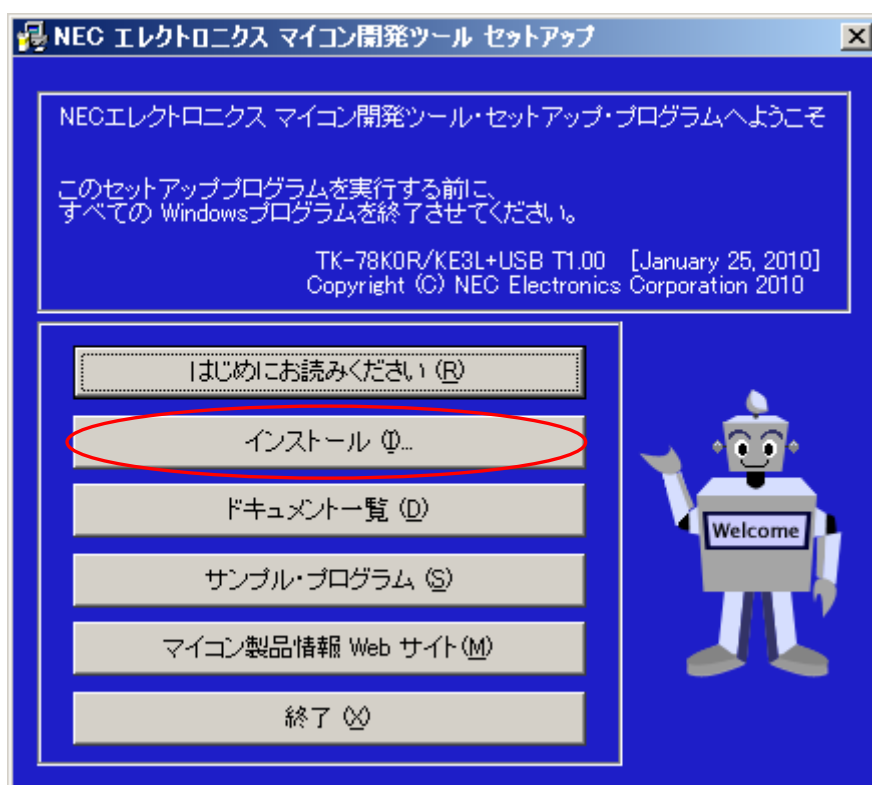
1.2 開発ツールのインストール

1.2.1 インストーラの構成

インストールパッケージには開発ツール、ドキュメントが含まれています。
これらは、インストーラを使用してインストールすることができます。

1.2.2 開発環境のインストール

- ① 本製品の CD を CD ドライブに挿入することにより自動的にインストーラのメニューが起動します。
この画面が自動的に起動しない場合は、エクスプローラ等から SETUP.EXE を起動してください。



- ② **インストール(I)...**をクリックしてください。

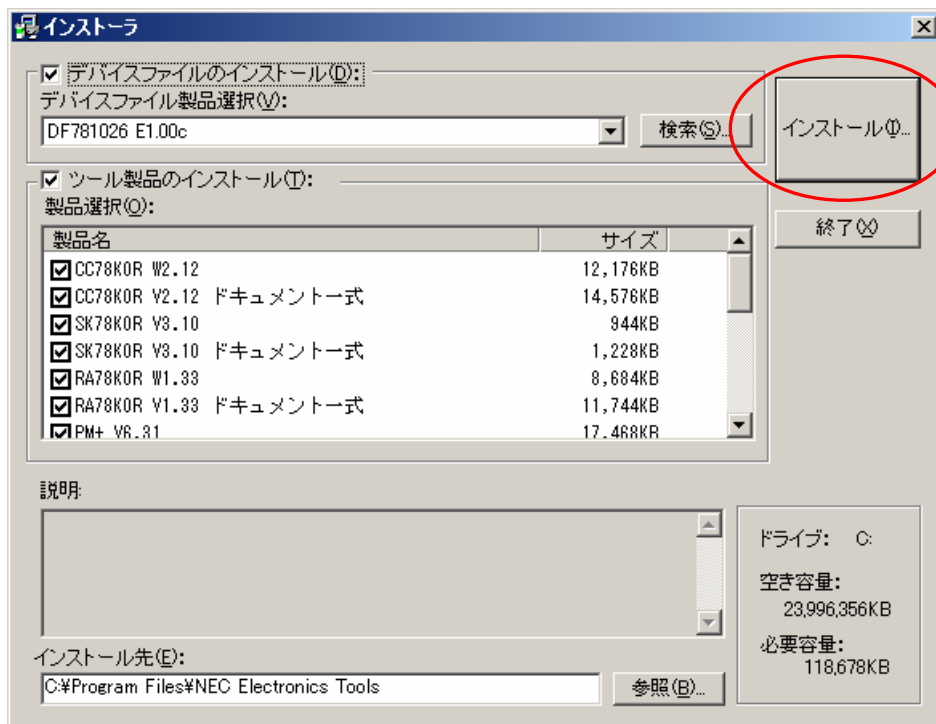
- ③ 以下のような、“インストーラ”ダイアログが開きます。
 インストールするツールをチェック・ボックスで選択します。
 (デフォルトでは、本キットを動作させるために必要なツールを全てインストールする設定になっています。)

“説明”には、選択されているツールの説明が表示されます。

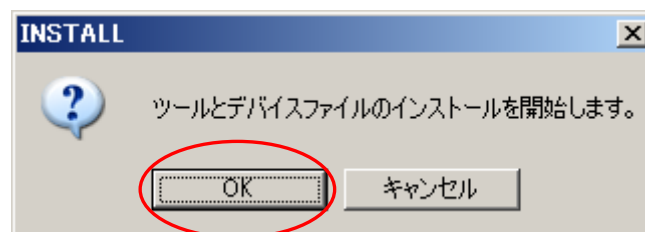
“インストール先(E):”を変更する場合には、**参照(B)...**を選択して変更してください。

すべての設定が完了したら**インストール(I)...**をクリックしてください。

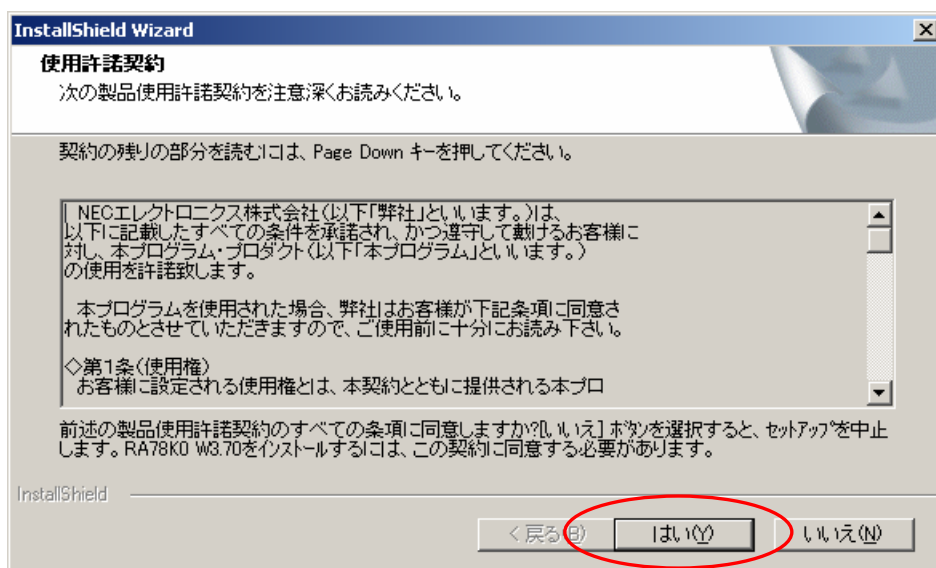
※この資料では、インストール・フォルダは、デフォルトの“NEC Electronics Tools”という名称で行ったと仮定して解説します。ツールを使用する場合には、スタートメニューの「プログラム」→「NEC Electronics Tools」からツールを起動してください。



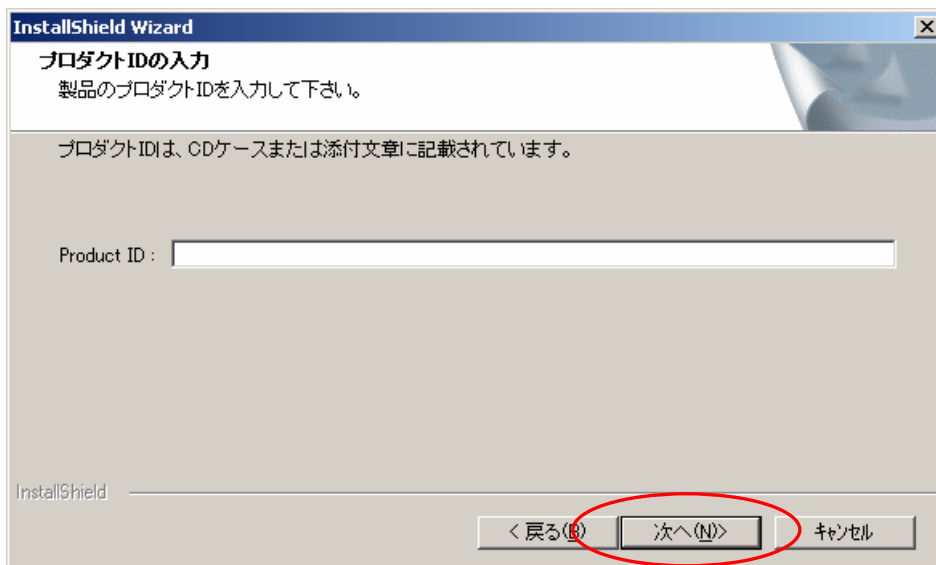
- ④ インストールの確認画面が表示されるので、**OK**をクリックします。



- ⑤ インストールを行う場合には、使用許諾契約に同意していただき、**はい(Y)**ボタンをクリックします。
いいえ(N)ボタンでインストールを中止します。

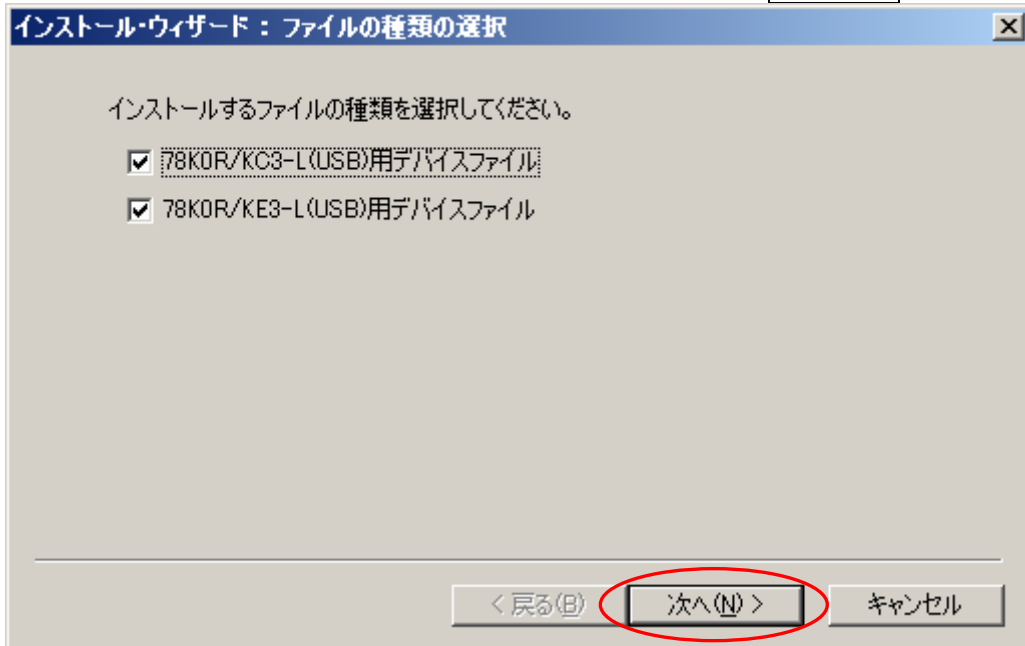


- ⑥ プロダクト ID を入力して、**次へ(N)**ボタンをクリックします。
※プロダクト ID は添付されている用紙と「はじめにお読みください(R)」に記載されています。



- ⑦ ファイルのコピーを開始します。

- ⑧ インストール途中、「ファイルの種類を選択」画面が表示されるので、**次へ(N)** をクリックします。



- ⑨ セットアップが完了すると次のダイアログがオープンするので、**OK** ボタンをクリックしてください。これで各開発ツールのインストールは終了です。



- ⑩ TK-78K0R/KE3L+USB をパソコンに接続して使用する際に、パソコンに“NEC Electronics Starter Kit Virtual UART”USBドライバをインストールする必要があります。

開発ツールをインストール後、「[1.3 USBドライバのインストール](#)」を参照してUSBドライバのインストールを行ってください。

インストールに関する注意事項

- ・ サポートしている Windows は WindowsXP および Windows 2000 です。
- ・ Windows へログインする際、Administrator 権限でログインしてインストールを行ってください。
- ・ ASCII 文字(半角英数字などの 1 バイト文字)のみ使用するフォルダへインストールしてください。漢字などの 2 バイト文字を含むフォルダや、/*:<>?|'¥, の 11 文字を含むフォルダへはインストールしないでください。正常に動作しない場合があります。
- ・ 日本語版 Windows にのみ対応しています。日本語版以外の Windows へインストールすることはできません。

制限事項について

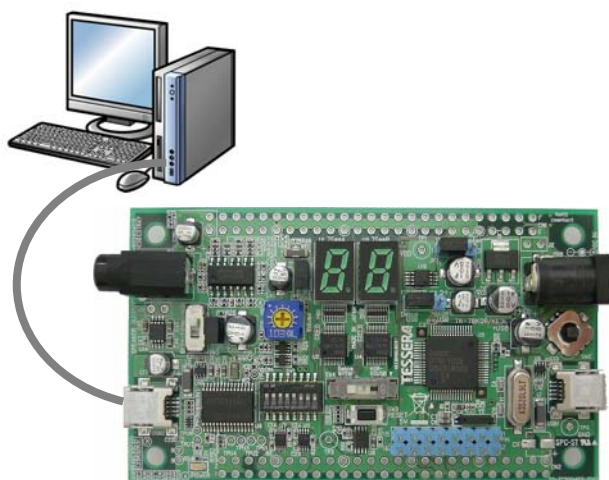
- ・ 本インストーラ内の RA78K0R、CC78K0R にはオブジェクト・サイズが 64KByte までの制限事項を設けています。

1.3 USB ドライバのインストール

TK-78K0R/KE3L+USB を使用する場合には、パソコンに“NEC Electronics Starter Kit Virtual UART”ドライバをインストールする必要があります。以下の手順でドライバのインストールを行ってください。本項の説明は、「[1.2 開発ツールのインストール](#)」で“Starter Kit USB Driver”がインストールされていることを前提に説明しています。

注意 USBハブ経由でTK-78K0R/KE3L+USBを接続しないでください。正常に動作しない場合があります。

まず、TK-78K0R/KE3L+USB の USB2 コネクタとパソコンを接続します。



以降、インストール方法は Windows の種類によって異なりますので、別々に説明します。

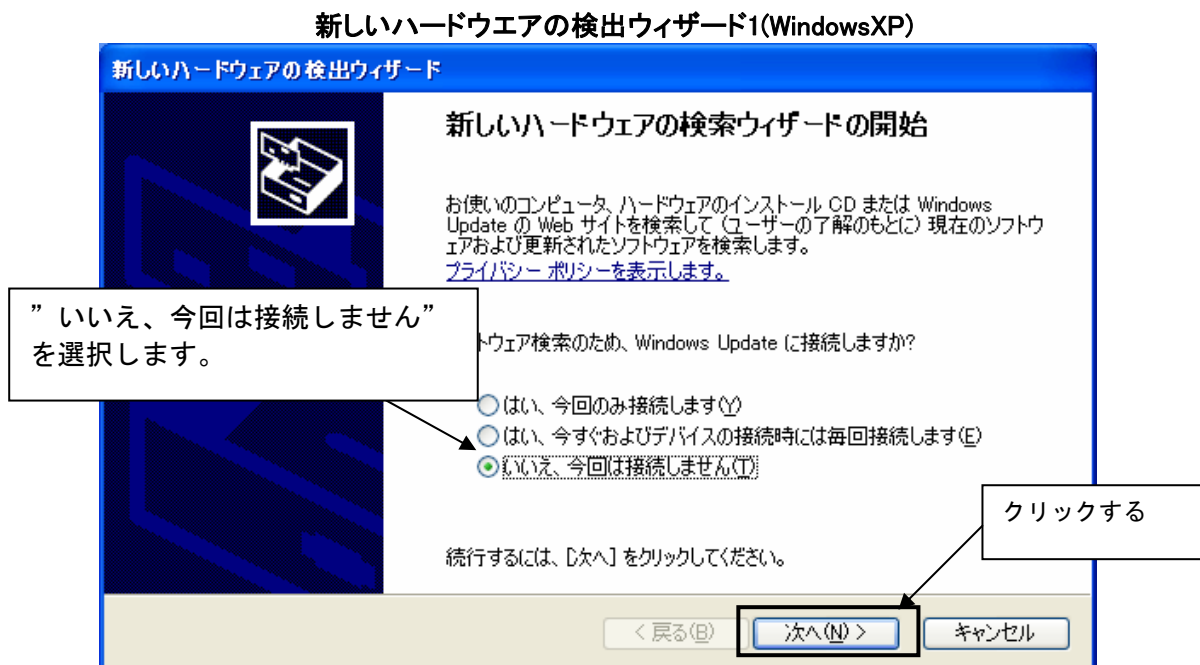
[Windows XP へのインストール](#)

[Windows2000 へのインストール](#)

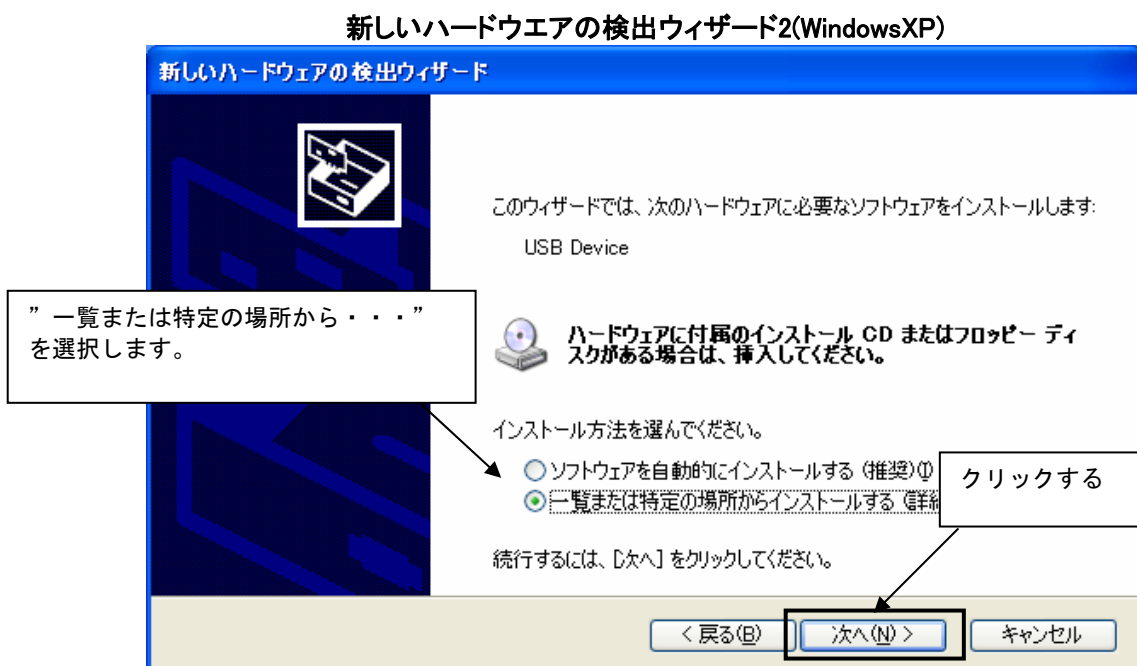
インストール終了後、「[1.3.3 USB ドライバのインストール完了確認](#)」へ進んでください。

1.3.1 Windows XP へのインストール

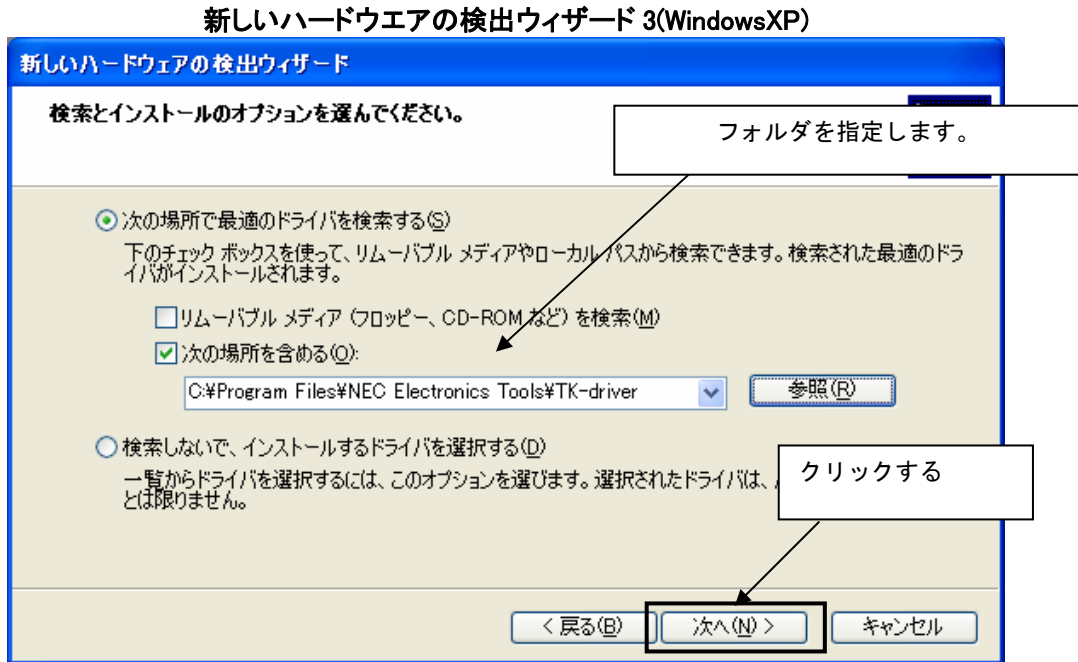
- ① TK-78K0R/KE3L+USB と接続すると、プラグ & プレイで接続が認識されて”新しいハードウェアの検出ウィザード”が起動します。”いいえ、今回は接続しません”を選択し、**次へ(N)**をクリックします。



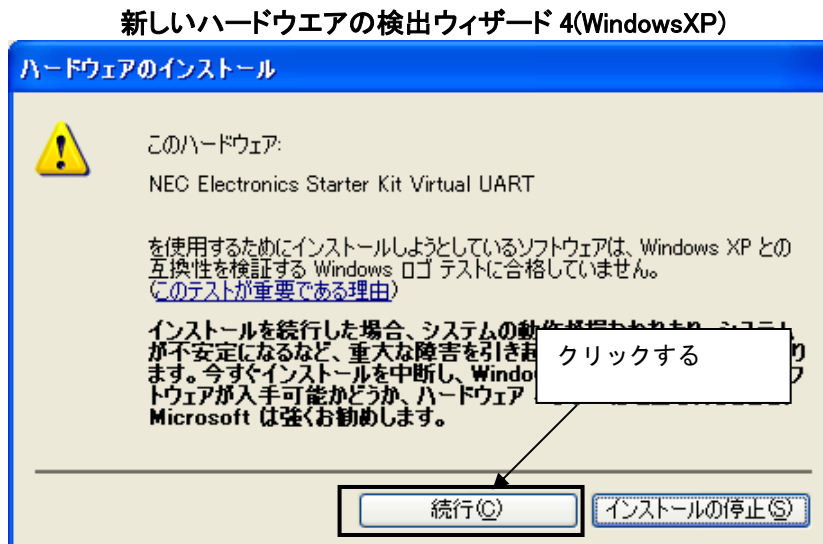
- ② 以下のウインドウが表示されたら“一覧または特定の場所からインストールする”を選択し、**次へ(N)**をクリックします。



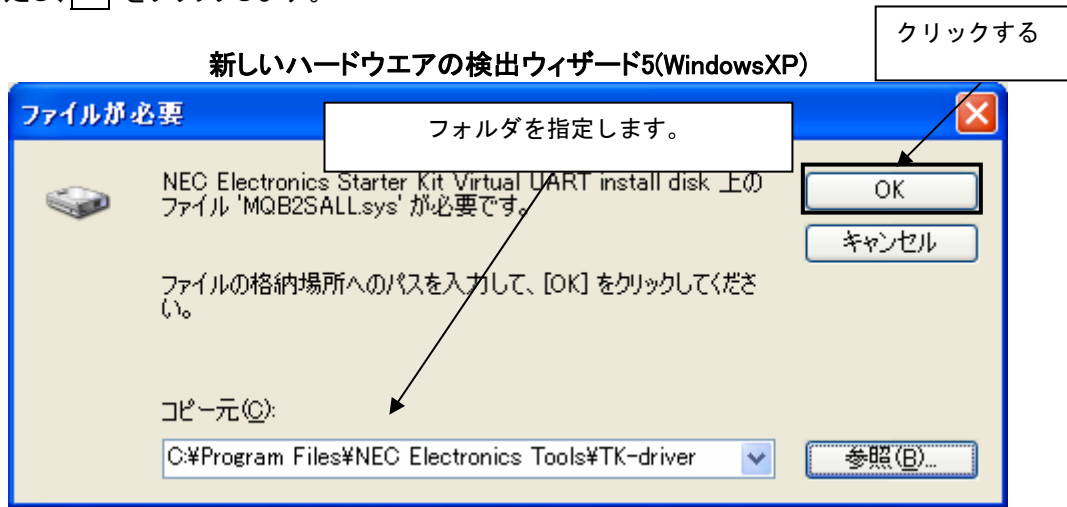
- ③ 以下のウインドウが表示されたら、“次の場所で最適のドライバを検索する”と“次の場所を含める”を選択し、「参照」をクリックします。インストール先のフォルダがデフォルト設定の場合、“C:\Program Files\NEC Electronics Tools\TK-driver”を指定し、「次へ(N)>」をクリックします。インストール先のフォルダがデフォルト設定で無い場合は、インストール先フォルダ内の“TK-driver”フォルダを指定してください。



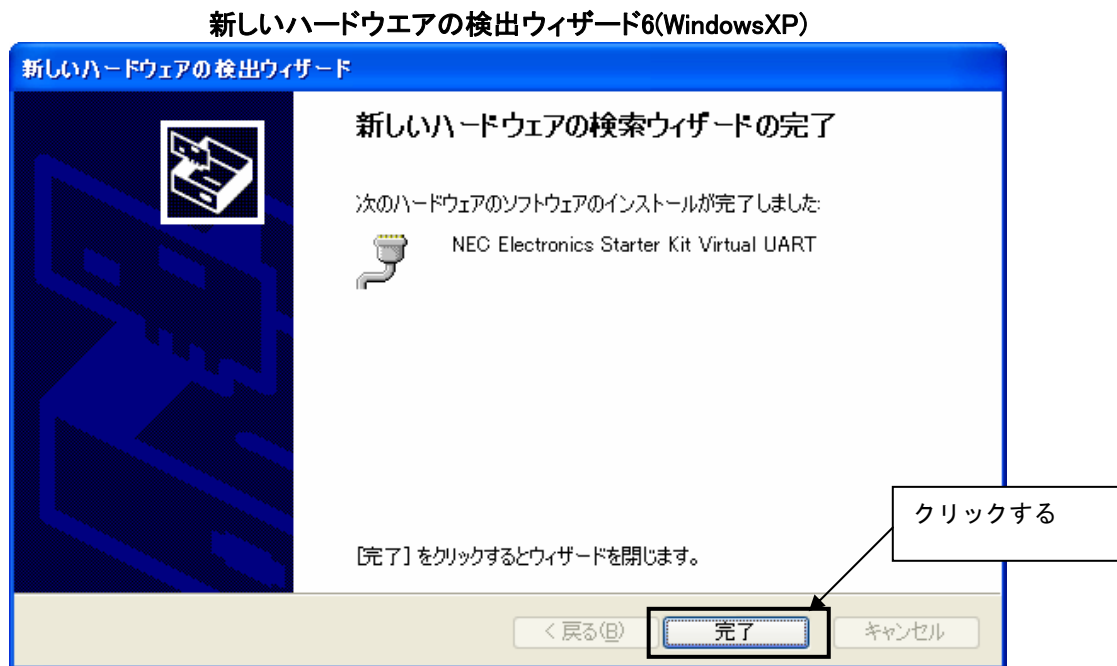
- ④ 以下のインストール確認画面が表示される場合には「続行(C)」をクリックします。



- ⑤ 以下のウィンドウが表示されたら、先ほどと同様に
”C:¥Program Files¥NEC Electronics Tools¥TK-driver”
を指定し、**OK** をクリックします。



- ⑥ 以下のウィンドウが表示されたら“NEC Electronics Starter Kit Virtual UART”ドライバのインストール
は完了です。**完了**をクリックします。

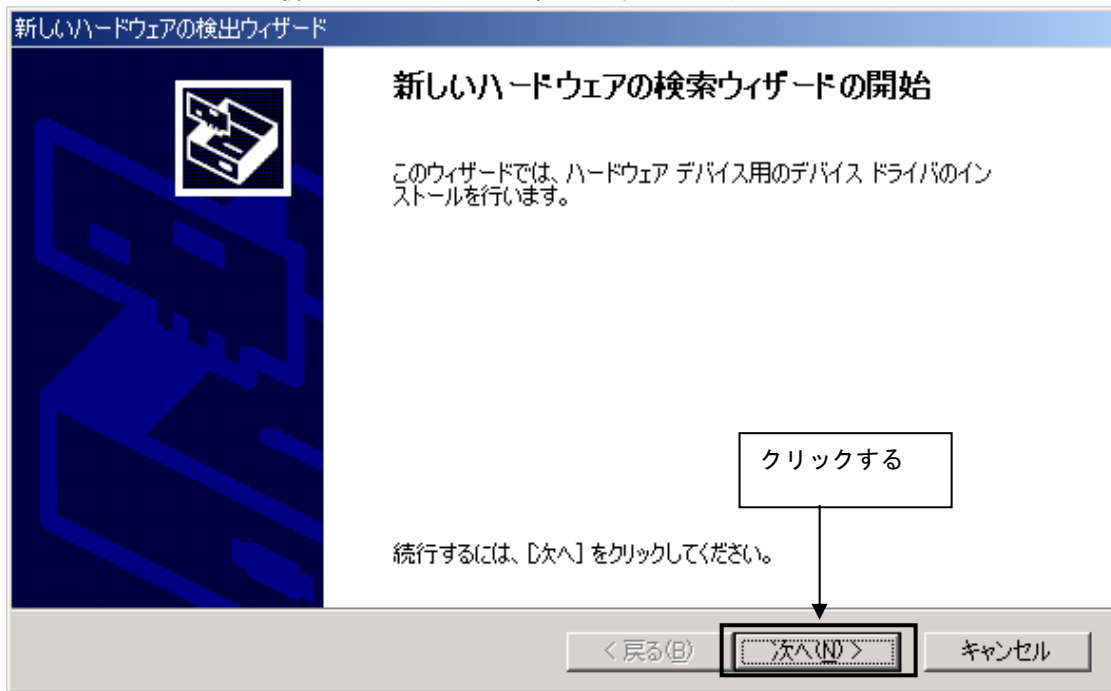


- ⑦ 「[1.3.3 USBドライバのインストール完了確認](#)」へ進んでください。

1.3.2 Windows2000 へのインストール

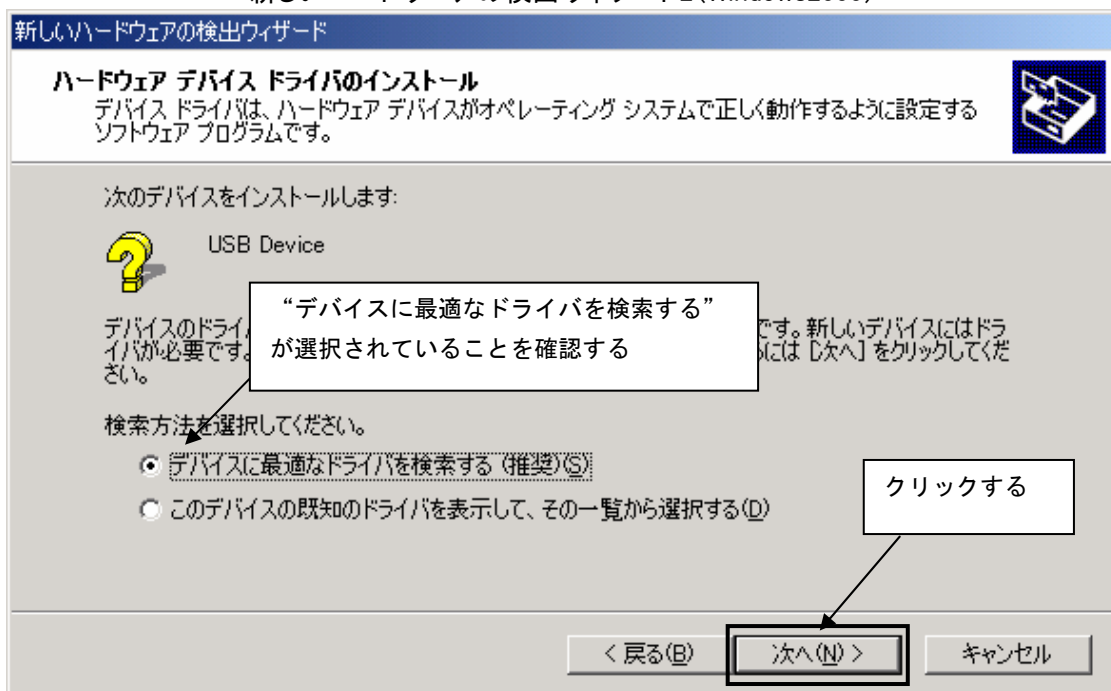
- ① TK-78K0R/KE3L+USB をパソコンと接続すると、プラグ & プレイで接続が認識され”新しいハードウェアの検出ウィザード”が起動します。 **次へ(N)>**をクリックします。

新しいハードウェアの検出ウィザード1 (Windows2000)

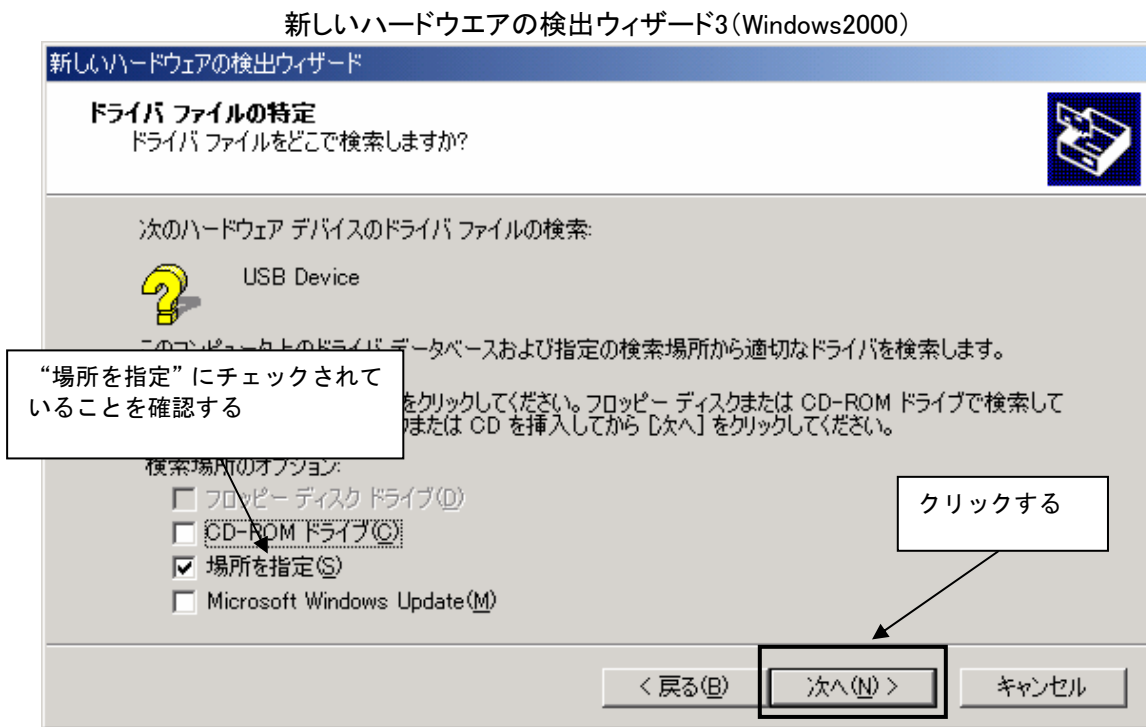


- ② 下記ウィンドウが表示されるので、“デバイスに最適なドライバを検索する。”が選択されていることを確認し、 **次へ(N)>**をクリックします。

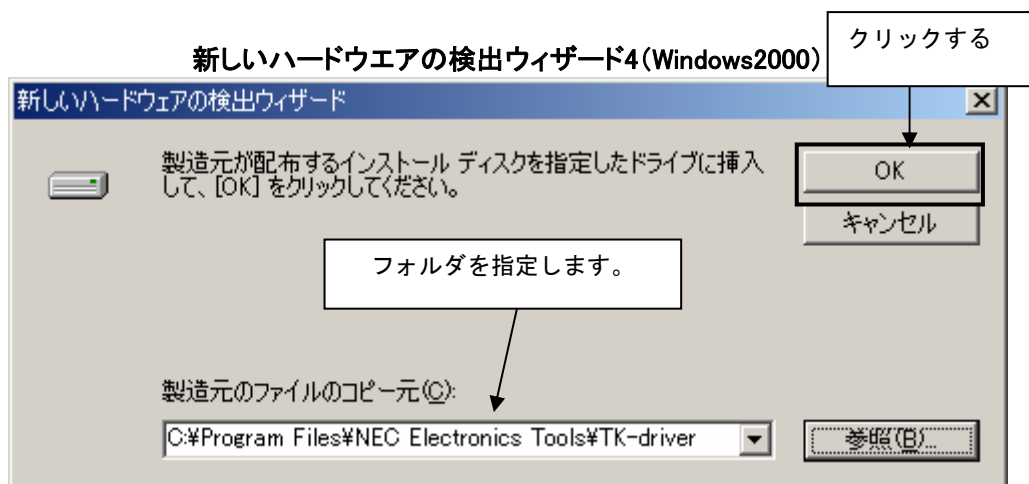
新しいハードウェアの検出ウィザード2 (Windows2000)



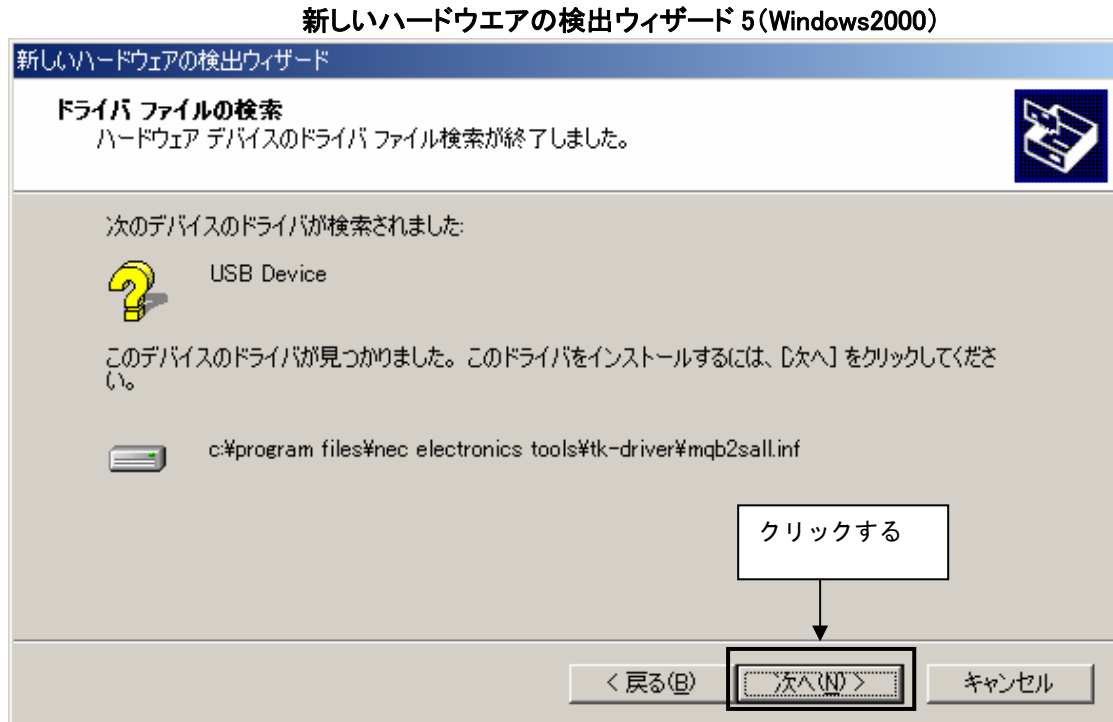
- ③ “場所を指定”のチェック・ボックスだけにチェックを入れ、**次へ(N)>**をクリックします。



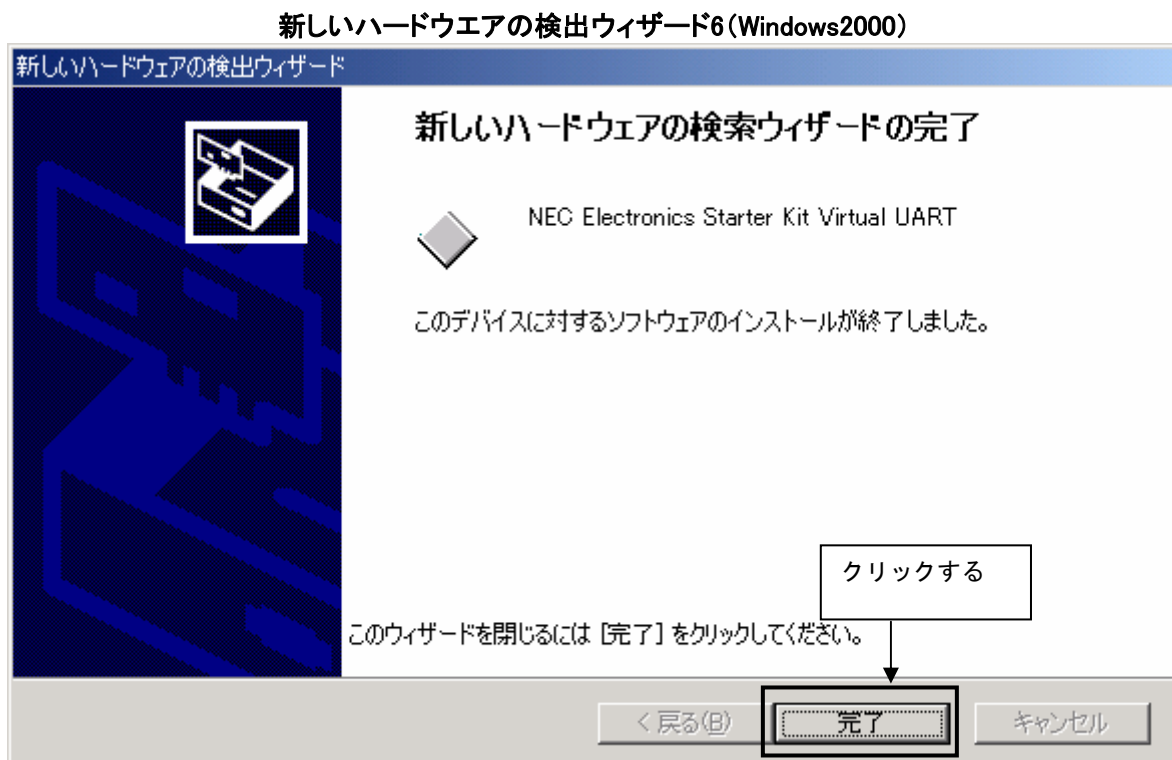
- ④ インストール先のフォルダがデフォルト設定の場合、“製造元のファイルのコピー元”に”C:\Program Files\NEC Electronics Tools\TK-driver”を入力し、**OK**をクリックします。インストール先のフォルダがデフォルト設定で無い場合は、インストール先フォルダ内の”TK-driver”フォルダを指定してください。



- ⑤ **次へ(N)** をクリックします。



- ⑥ **完了** をクリックして“NEC Electronics Starter Kit Virtual UART”ドライバのインストールを完了します。

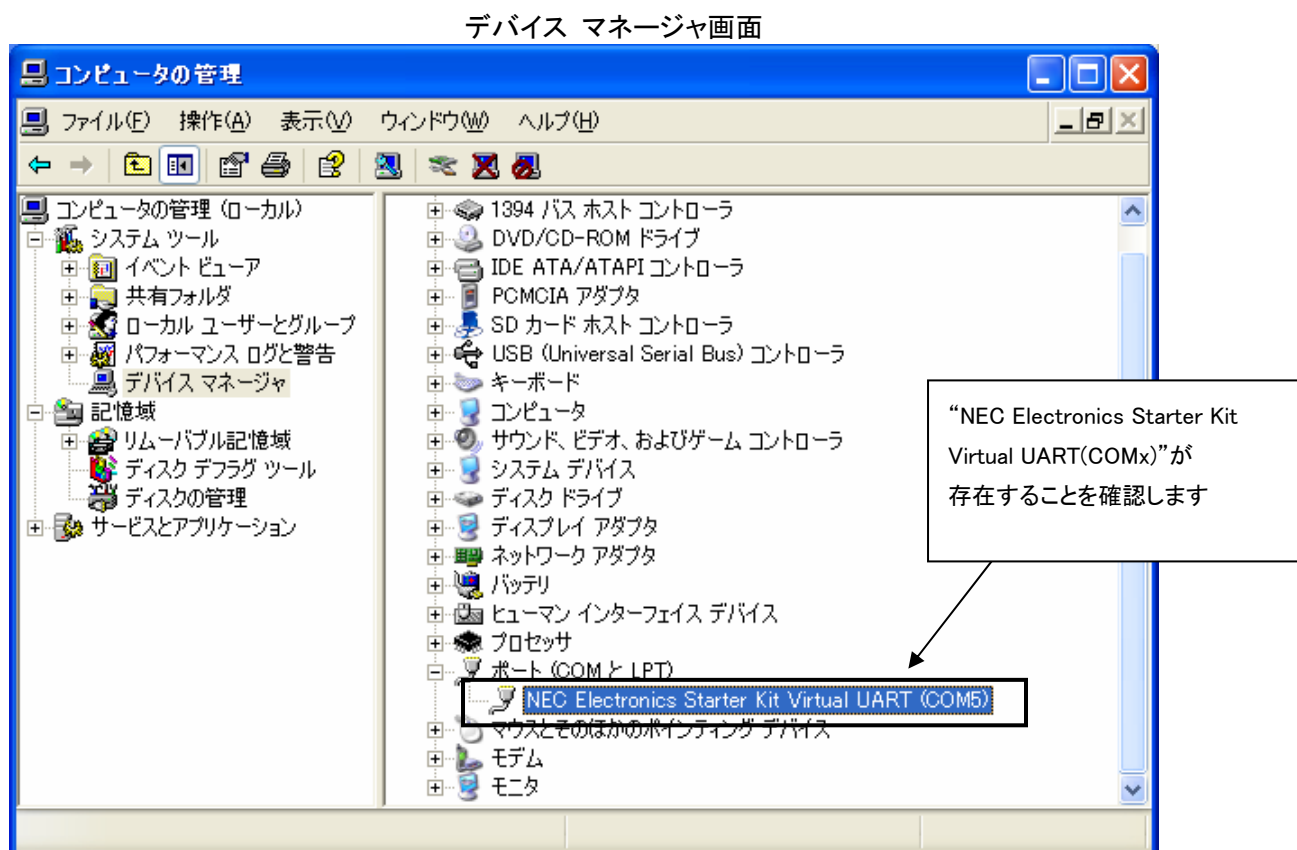


- ⑦ 「[1.3.3 USBドライバのインストール完了確認](#)」へ進んでください。

1.3.3 USB ドライバのインストール完了確認

ドライバのインストール完了後、以下の手順によりドライバが正常にインストールされていることを確認する事ができます。

“デバイス マネージャ”のタブをクリックし、“ポート (COM と LPT)”に“NEC Electronics Starter Kit Virtual UART”が表示され、“?”マークなどが表示されていないことでドライバが正常にインストールされていることを確認します。



上記の画面では TK-78K0R/KE3L+USB は“COM5”に割り当てられました。
ID78K0R-QBを使用しない場合には、この COM ポートを使用して TK-78K0R/KE3L+USB とパソコン間でシリアル通信を行うことが可能です。
接続する USB ポートを変更すると COM ポート番号も変更になります。ご注意ください。

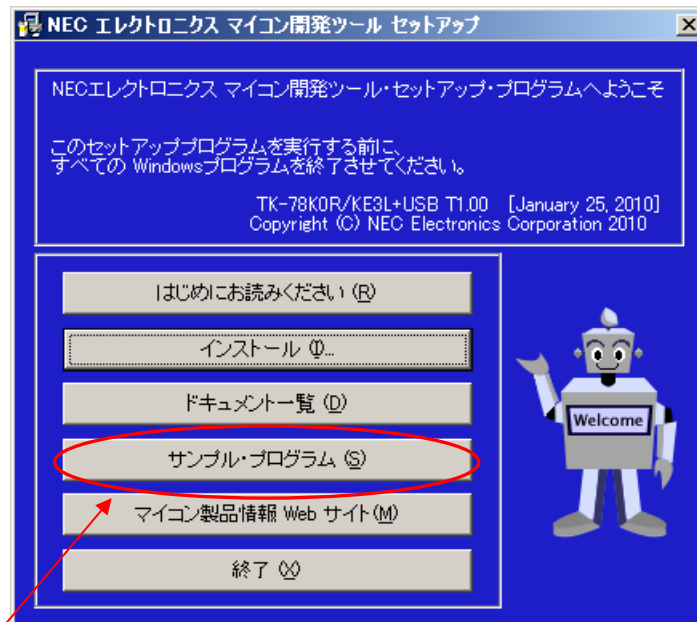
※注意 マイコンとの通信時は“ハードウェア変更のスキャン”を行わないでください。

1.4 サンプル・プログラム

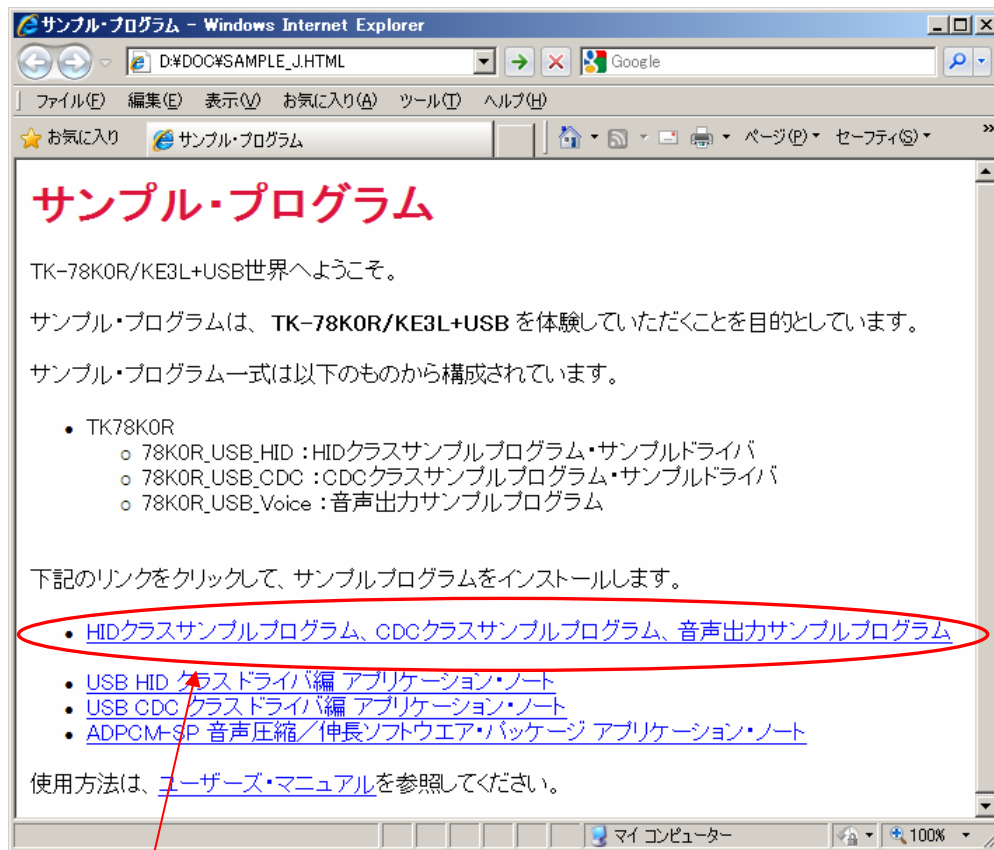
使用するサンプル・プログラムの準備方法と概要について説明します。
サンプル・プログラムの詳細は「[第4章 サンプル・プログラムの解説](#)」を参照してください。

1.4.1 サンプル・プログラムの準備

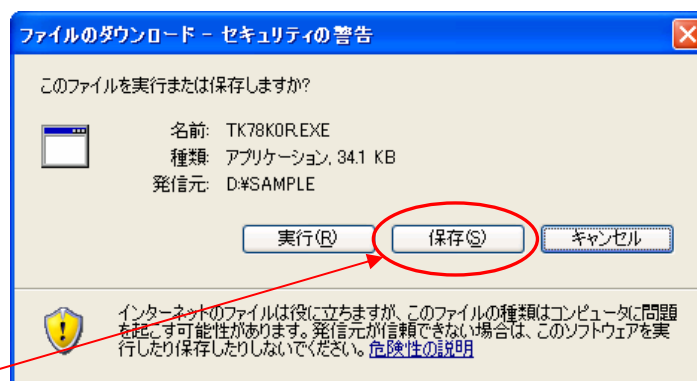
- ① 本製品の CD を CD ドライブに挿入することにより自動的にインストーラのメニューが起動します。この画面が自動的に起動しない場合は、エクスプローラ等から SETUP.EXE を起動してください。



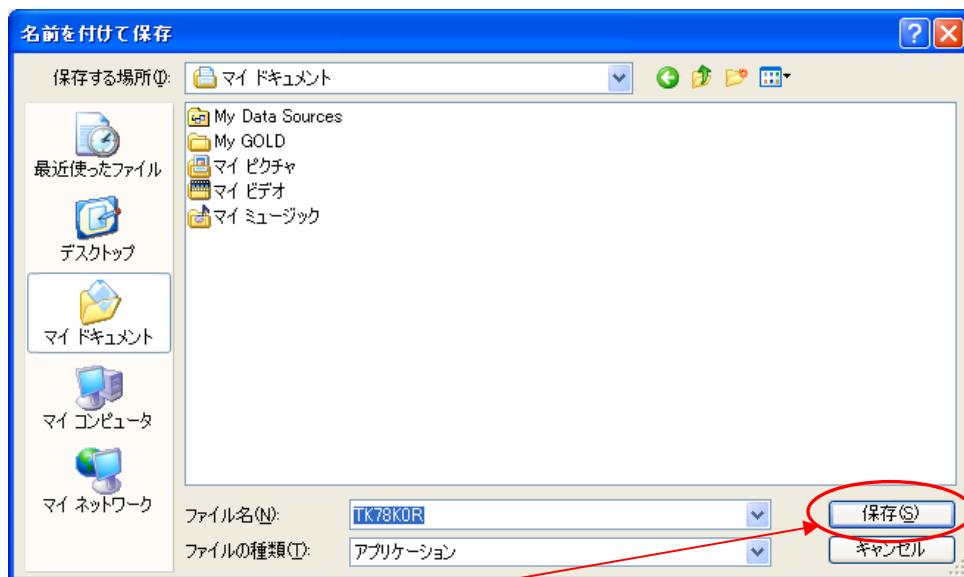
- ② **サンプル・プログラム (S)**をクリックしてください。



- ③ 「HID クラスサンプルプログラム、CDC クラスサンプルプログラム、音声出力サンプルプログラム」をクリックします。



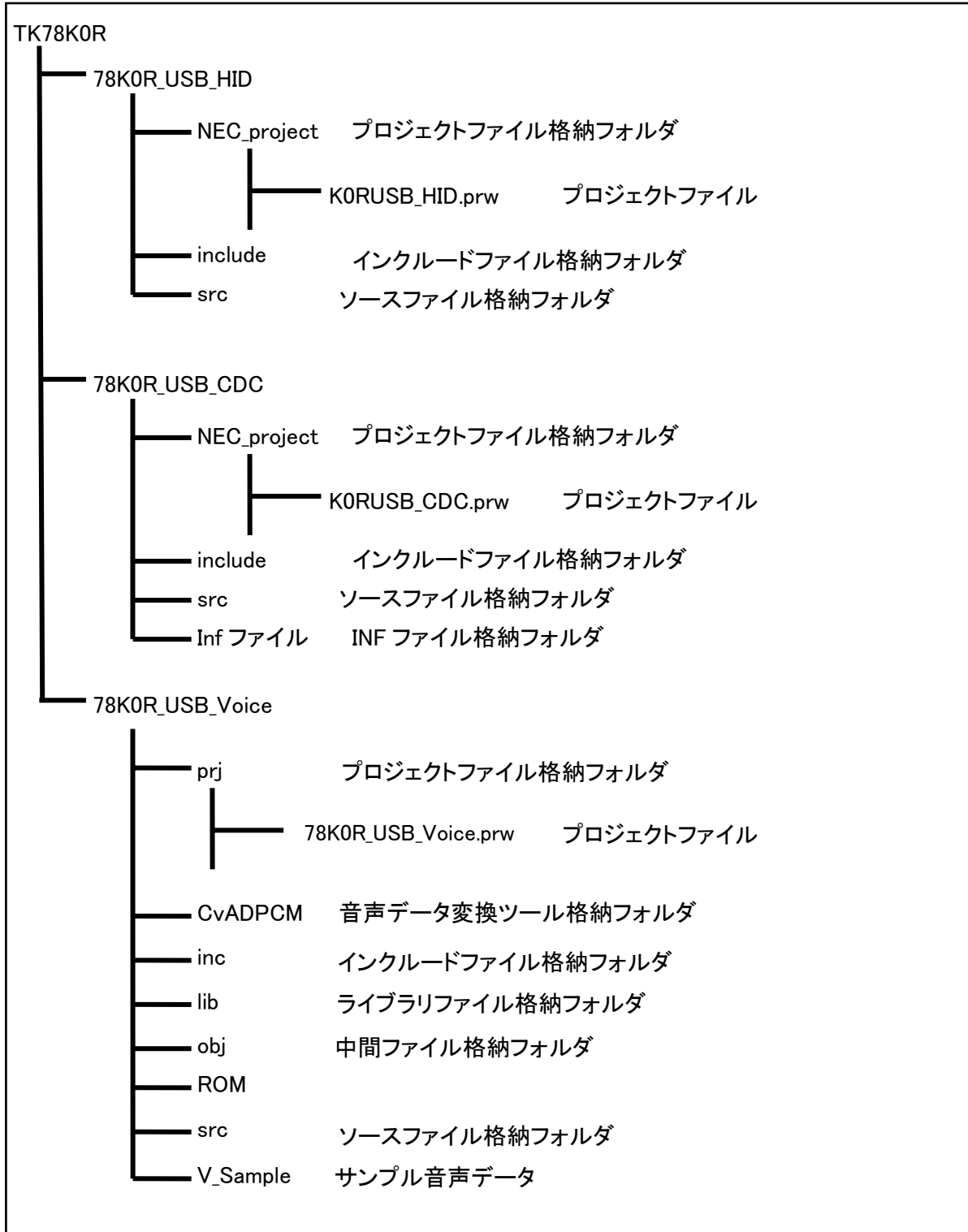
- ④ **保存(S)**をクリックします。



- ⑤ **保存(S)**をクリックします。
- ⑥ 指定したフォルダに自己解凍形式のサンプル・プログラム一式(TK78K0R.exe)がコピーされます。このファイルを実行すると、“TK78K0R”フォルダが作成され、更にそのフォルダの下にサンプル・プログラムが格納されるフォルダが作成されます。

1.4.2 サンプル・プログラムのフォルダ構成と概要

サンプル・プログラムは次のようなフォルダ構成です。



第2章 体験編

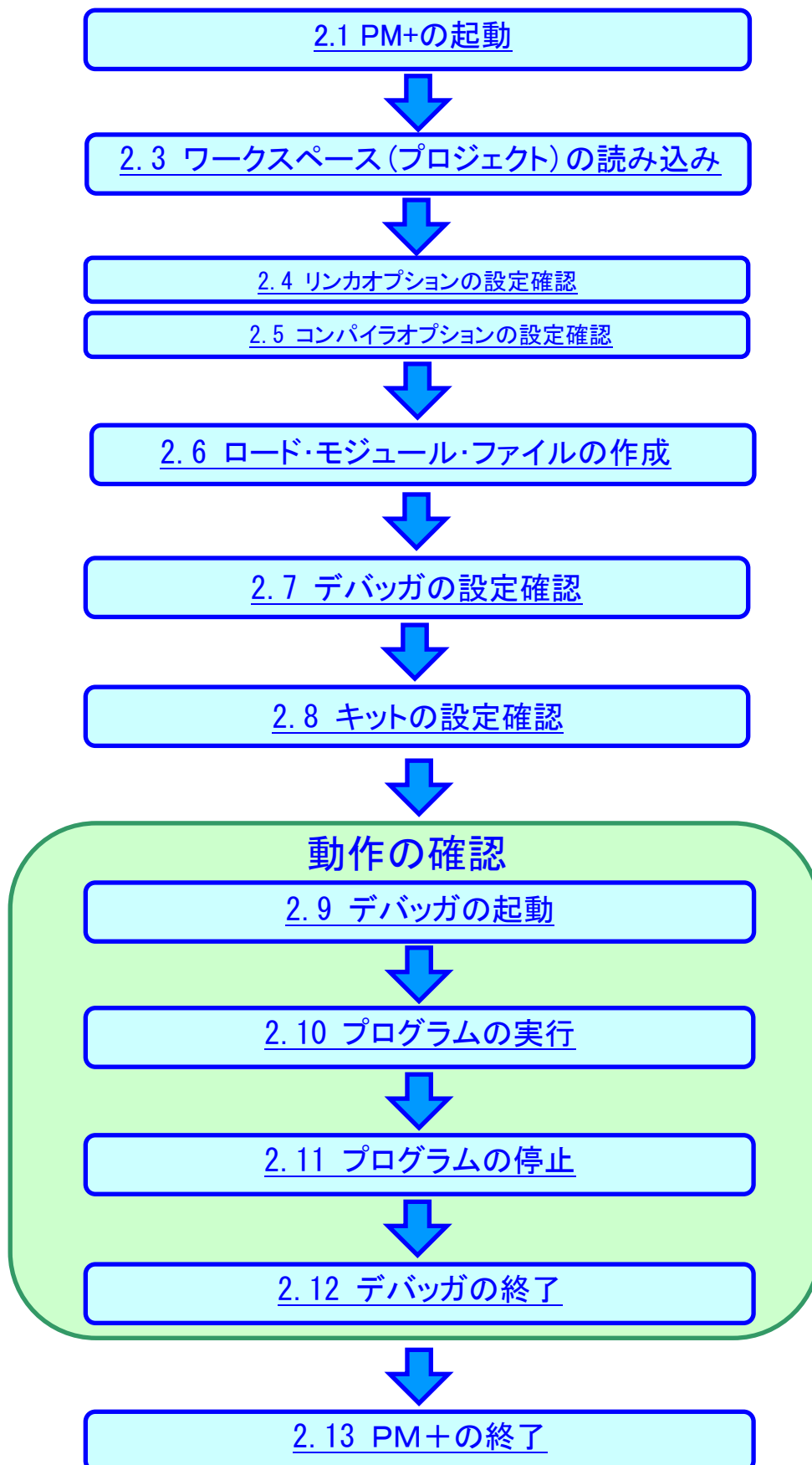
この章では、サンプル・プログラムを使用して、以下の開発ツールを体験します。

- ・統合開発環境(PM+)でのビルド
- ・統合デバッガ(ID78K0R-QB)でのプログラム実行

TK-78K0R/KE3L+USB 用サンプル・プログラムとして、「[1.4 サンプル・プログラム](#)」で準備したプログラムを使用します。

サンプル・プログラムを[ビルド](#)し、ID78K0R-QB でプログラム実行することを通して、開発ツール(PM+, ID78K0R-QB)の基本的な操作方法と、アプリケーション・プログラムの作成時に必要な[プロジェクト・ファイル](#)の概念を理解できます。

全体の流れを次に示します。



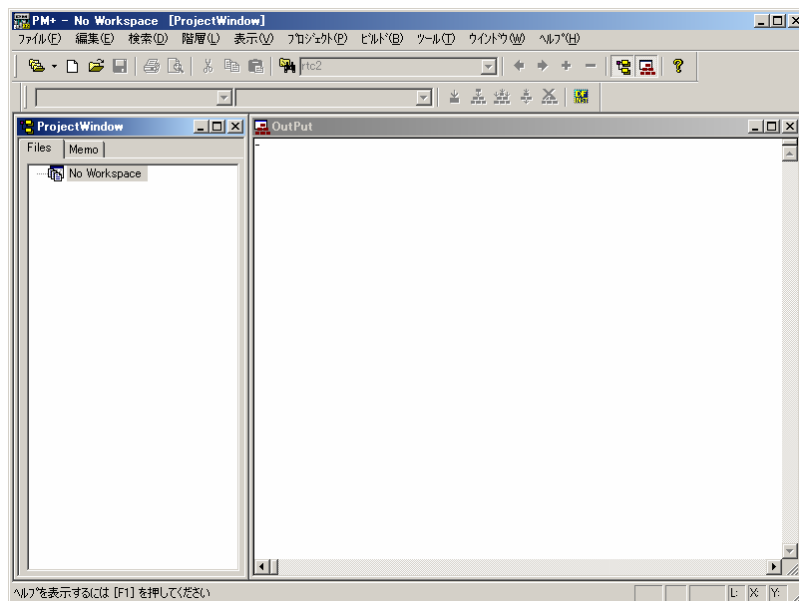
2.1 PM+の起動

それでは、実際に各開発ツールを使用してみましょう。

まず、PM+を起動します。

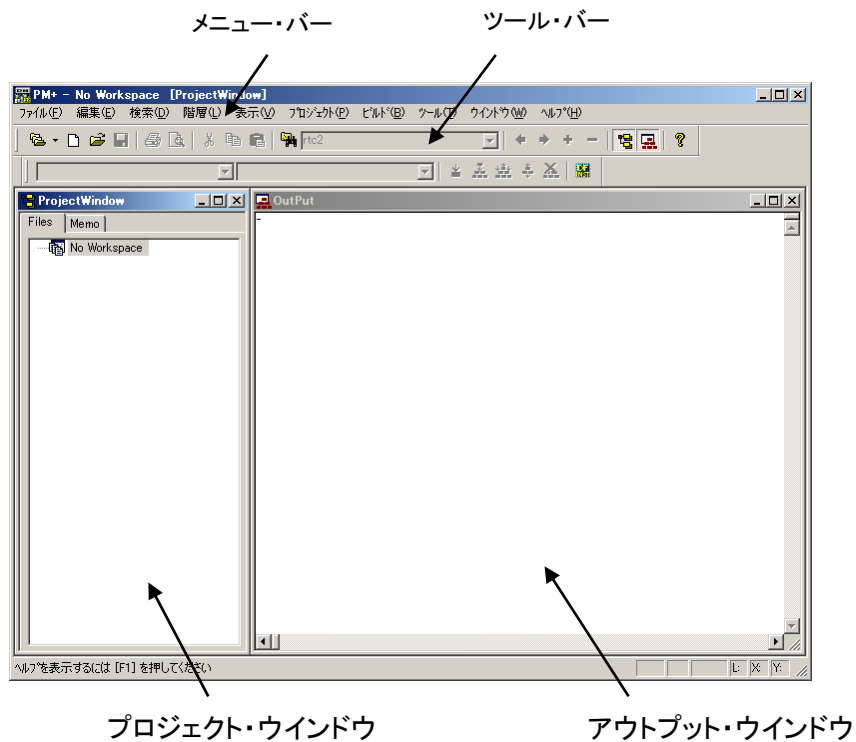
Windows スタート・メニューの[プログラム(P)]→[NEC Electronics Tools]→[PM+ V6.31]を選択してください

PM+が
起動します。



2.2 PM+の紹介

PM+では、アプリケーション・プログラムや環境設定を一つのプロジェクトとして扱い、エディタでのプログラム作成、ソース管理、ビルド、デバッガの起動といった一連の作業を管理します。
また、1つ以上のプロジェクト・ファイルをまとめてワークスペースとして管理します。



プロジェクト・ウインドウ : プロジェクト名やそのソース・ファイル、インクルード・ファイルがツリー構造で表示されるウインドウです。

アウトプット・ウインドウ : [ビルド](#)の実行過程が表示されるウインドウです。

- ➡ メニュー・バー、およびツール・バーの詳細については、PM+のヘルプを参照してください。メニュー・バーの[ヘルプ]→[PM+のヘルプ]で参照できます。

プロジェクトとは？

PM+が管理する単位で、PM+のもとで開発されるアプリケーション・システムと環境を指します。
PM+は、プロジェクトの情報をまとめて“プロジェクト・ファイル”に保存、参照します。

プロジェクト・ファイルとは？

プロジェクトで使用するソース・ファイル、デバイス名、コンパイルのためのツール・オプション、使用するエディタやデバッガなどの情報を“プロジェクト情報”として、各プロジェクト・ファイルに保存したファイルです。

ファイル名は“△△△△. prj”です。

プロジェクト・ファイルは、ワークスペースの新規作成時に設定するフォルダに作成されます。

プロジェクト・グループとは？

アプリケーション・システム内の複数のプロジェクトを、関連するプロジェクトごとにまとめたものです。
1つのプロジェクト・グループで登録できる各プロジェクトの対象デバイスは同じでなければいけません。

ワークスペースとは？

1つのアプリケーション・システムに必要な全てのプロジェクト、またはプロジェクト・グループを管理する単位です。

複数のプロジェクト・ファイルのファイル名をワークスペース・ファイルに保存、参照します。

ファイル名は“△△△△. prw”です。

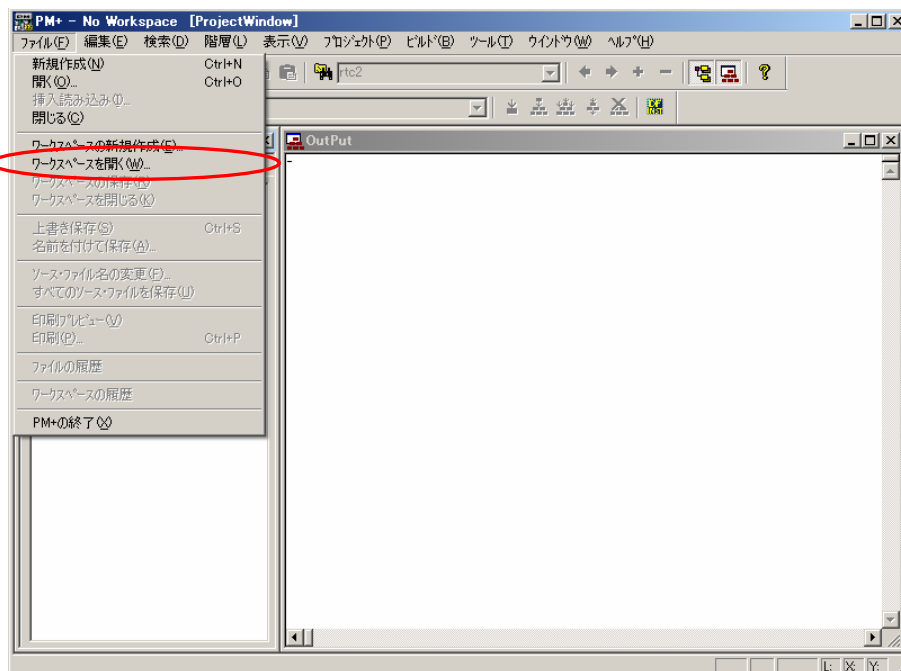
2.3 ワークスペース(プロジェクト)の読み込み

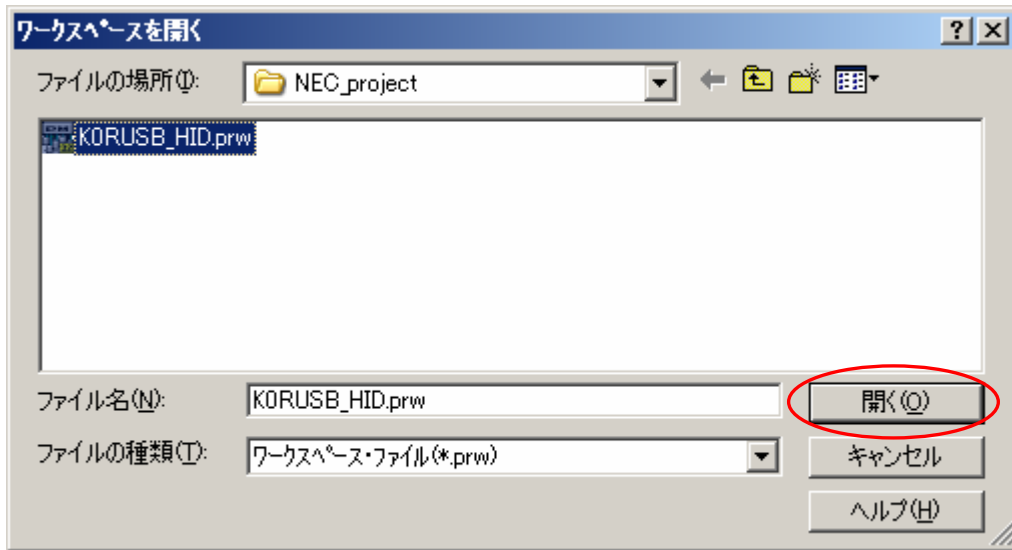
この章では、「[1.4 サンプル・プログラム](#)」で準備した、あらかじめ作成されているワークスペースを使用します。

➡ ワークスペースを新規に作成する方法は、「[第5章 その他](#)」で説明します。

この章で使用するワークスペースには、サンプル・プログラムのビルド環境が保存されています。

PM+のメニューの [ファイル(F)]→[ワークスペースを開く(W)...] を選択し、「TK78K0R¥78K0R_USB_HID ¥NEC_project」から“K0RUSB_HID.prw”を指定してください。

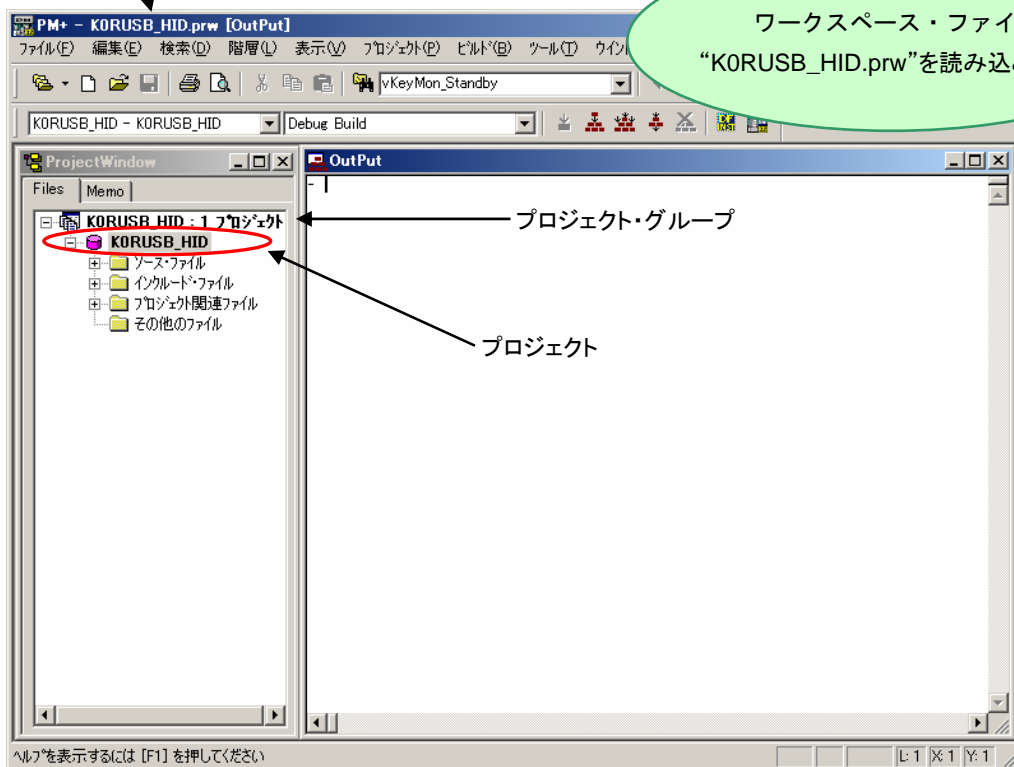




“KORUSB_HID.prw”を指定して **開く(O)** ボタンを押してください。



ワークスペース名: KORUSB_HID.prw



ワークスペース・ファイル“KORUSB_HID.prw”には、プロジェクト“KORUSB_HID”を1つだけ含んでいます。この後は プロジェクト“KORUSB_HID”を対象に操作を行います。

注意: 開発ツールをインストールしたフォルダがデフォルト設定で無い場合、いくつかファイルが見つからない旨のメッセージが出ることがありますが、無視してください。

2.4 リンカオプションの設定確認

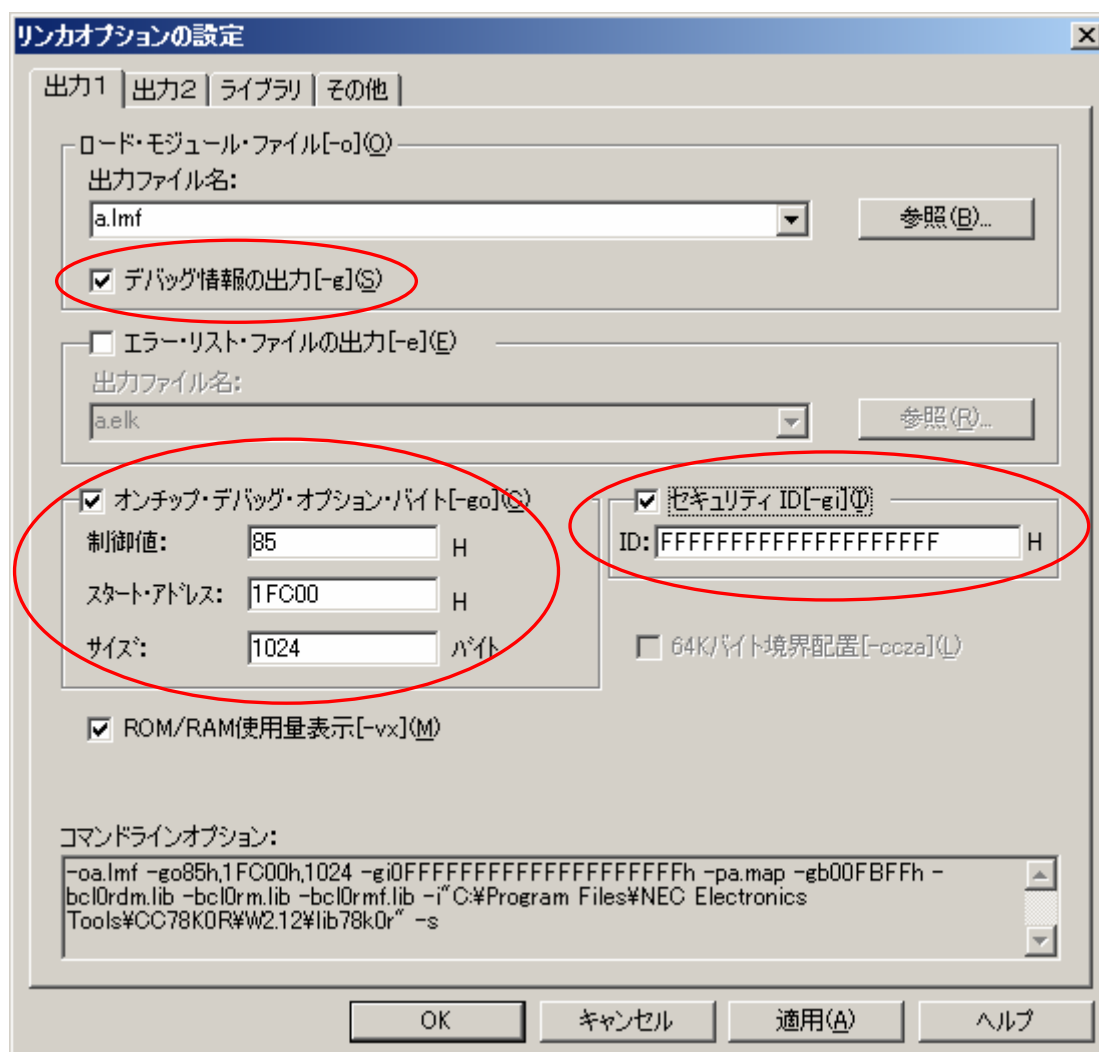
リンカオプションはプロジェクト・ファイルで設定済みの内容ですが、デバッグを行なう上で大切な設定内容が含まれているため、ここでは以下 3 点を中心に説明を行います。

- ・デバッグ情報の出力設定
- ・オンチップ・デバッグの設定(動作許可/禁止設定、およびセキュリティ ID の設定)
- ・ウォッチドッグ・タイマの設定

PM+のメニュー・バーから[ツール(T)] → [リンカオプションの設定] を選択します。

2.4.1 「出力 1」タブ

「リンカオプションの設定」画面で「出力 1」タブを選択し、以下の設定を確認します。



・ロード・モジュール・ファイル設定エリア

「デバッグ情報の出力」にチェックを入れます。この設定を行うことにより、ソース・デバッグ(C言語のソース・ファイル上でブレーク・ポイント設定を行う、変数名をウオッチ・ウインドウに登録する等)が可能になります。

また、このエリアでロード・モジュール・ファイル名を指定することも可能です。

・オンチップ・デバッグ・オプション・バイト設定エリア

チェックを入れます。また、「制御値」に「85」を入力します。この設定を行うことにより、マイコンの持つオンチップ・デバッグ機能が有効になります。

※制御値に関する詳細は 78K0R/KE3-L のユーザーズ・マニュアル(U19878J)を参照してください。

また、「スタート・アドレス」に「1FC00」、「サイズ」に「1024」が入力されていることを確認します。この設定を行うことにより、モニタープログラム領域(オンチップ・デバッグ時にデバッガが使用する内蔵フラッシュ・メモリ領域)が確保されます。

具体的には、「制御値」が内蔵フラッシュ・メモリ内の C3H 番地に配置され、FFH が内蔵フラッシュ・メモリの次の番地に配置されます。このため、次の領域にはセグメントを配置することができません。

[オンチップ・デバッグで使用される領域]

- ・2H 番地、3H 番地
- ・CEH 番地～D7H 番地
- ・「スタート・アドレス」に設定した番地から「サイズ」で指定した分の領域

・セキュリティ ID 設定エリア

チェックを入れます。また、セキュリティ ID の値を入力します。セキュリティ ID はデバッガ起動時に認証を行なうための任意の ID コードです(10 バイト長)。セキュリティ ID はマイコンの内蔵フラッシュ・メモリ内 C4H～CDH に配置され、デバッガを起動した際にコンフィギュレーション・ダイアログに入力した値との比較が行なわれます。値が一致しない場合、デバッガが起動できなくなるため、マイコン内のプログラムが漏洩することを防止する機能として有効です。

デバッグ時、特にセキュリティを気にしない場合は“FFFFFFFFFFFFFFFF”を入力しておくことをお勧めします。(フラッシュ・メモリを消去した場合の値が、この値になるため。)

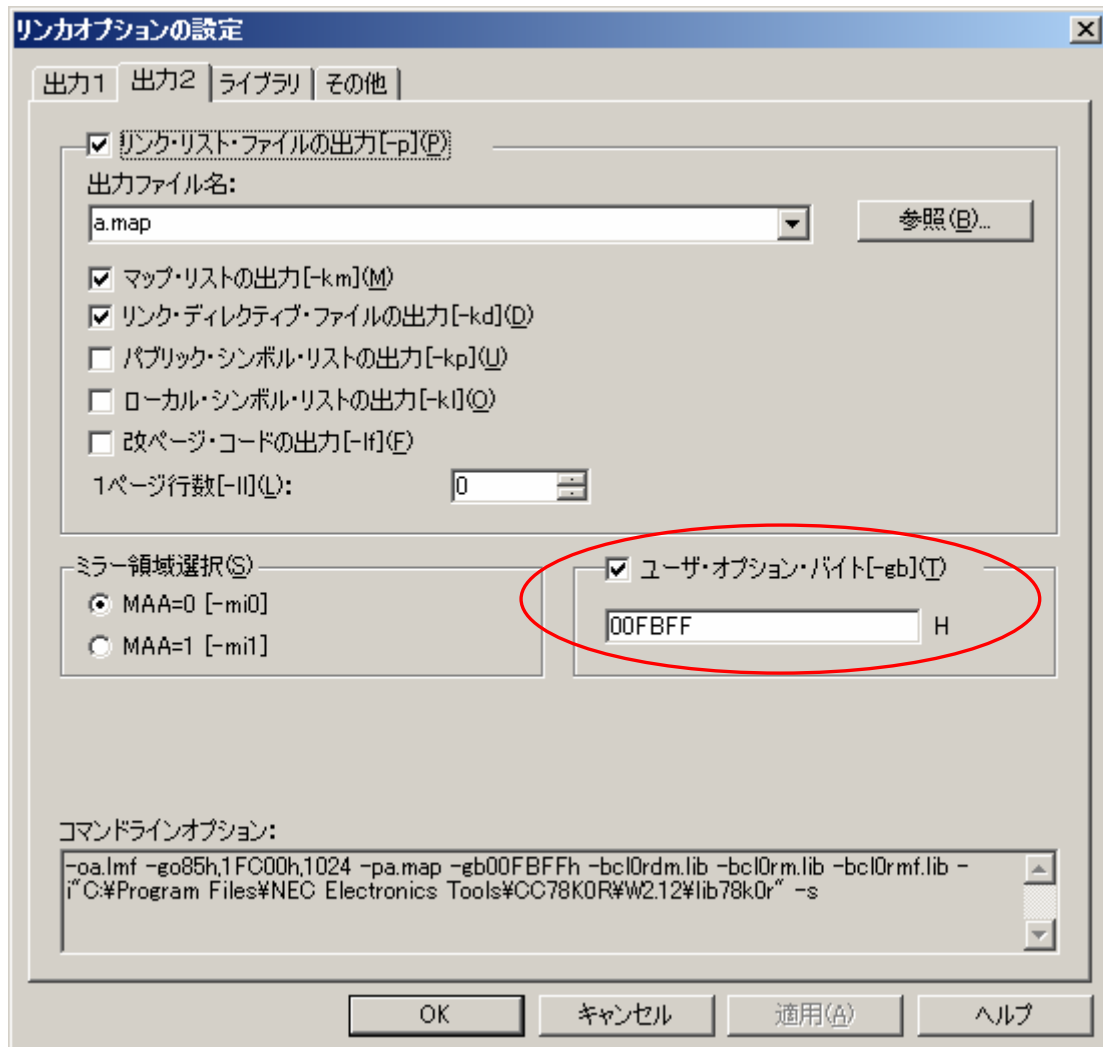
C4H～CDH 番地のセキュリティ ID がわからなくなってしまう場合や、オンチップ・デバッグ・オプション・バイト設定エリアの設定値を間違えてしまった場合などはデバッガ(ID78K0R-QB)が起動できなくなってしまう。

その場合、「WriteEZ5」を起動し、内蔵フラッシュ・メモリを消去することで ID78K0R-QB の接続が可能になります。

詳細は「5.4 マイコン内蔵のフラッシュ・メモリを消去したい」をご覧ください。

2.4.2 「出力 2」タブ

「リンクオプションの設定」画面で「出力 2」タブを選択し、以下の設定を確認します。



・ユーザ・オプション・バイト設定エリア

チェックを入れます。また、“00FBFF”を入力します。ここでは、ウォッチドッグ・タイマの設定、低電圧検出回路の設定、システム予約領域の設定が行われます。入力した 3 バイトはマイコン内蔵フラッシュ・メモリの C0H～C2H 番地に配置されます。それぞれ、以下のような意味を持ちます。

- ・C0H 番地: ウォッチドッグ・タイマの設定
- ・C1H 番地: 低電圧検出回路の設定
- ・C2H 番地: システム予約領域の設定(必ず FFH に設定)

今回は、ウォッチドッグ・タイマを停止、低電圧検出回路のデフォルト・スタート機能も停止に設定しています。

詳細は 78K0R/KE3-L のユーザーズ・マニュアル(U19878J)を参照してください。

2.5 コンパイラオプションの設定確認

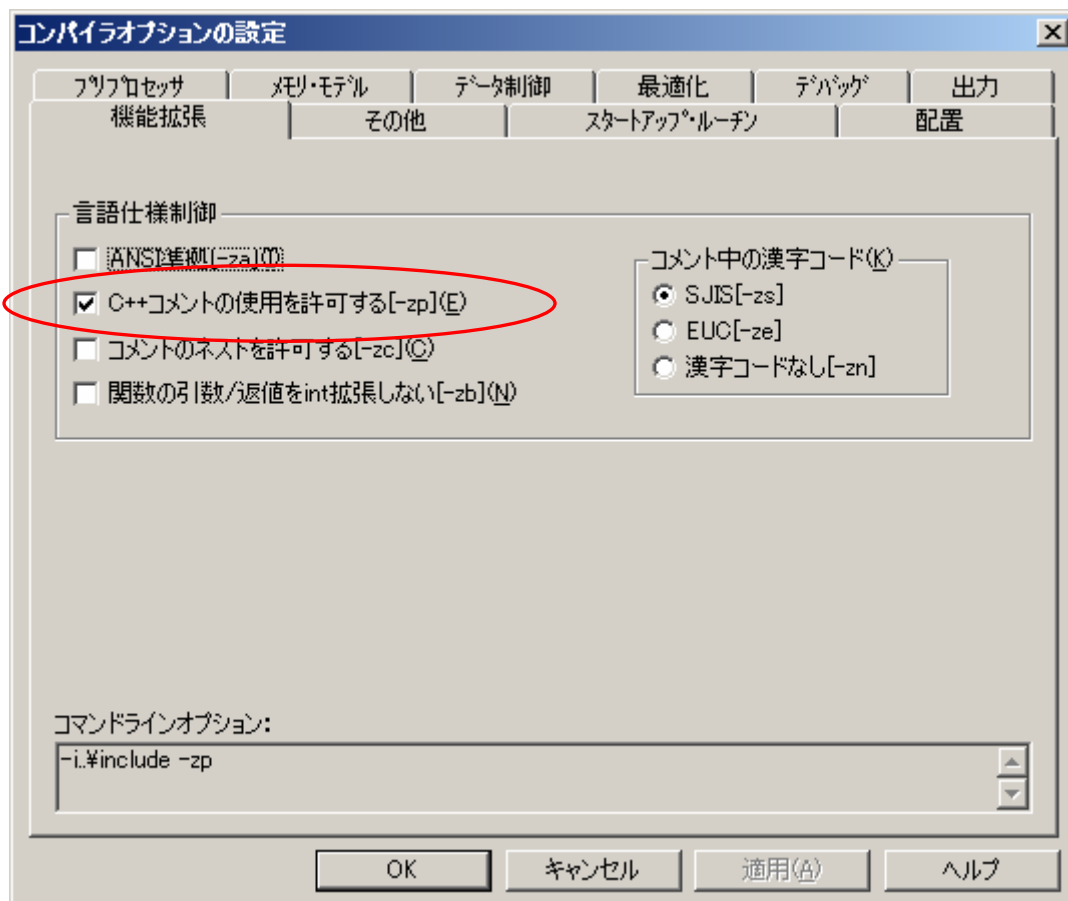
リンカオプションと同様にプロジェクト・ファイルで設定済みの内容ですが、便利な機能であるため、ここでは以下の2点を説明します。

- ・C++コメントの使用を許可する方法
- ・乗算器を使用する方法

PM+のメニュー・バーから[ツール(T)] → [コンパイラオプションの設定] を選択します。

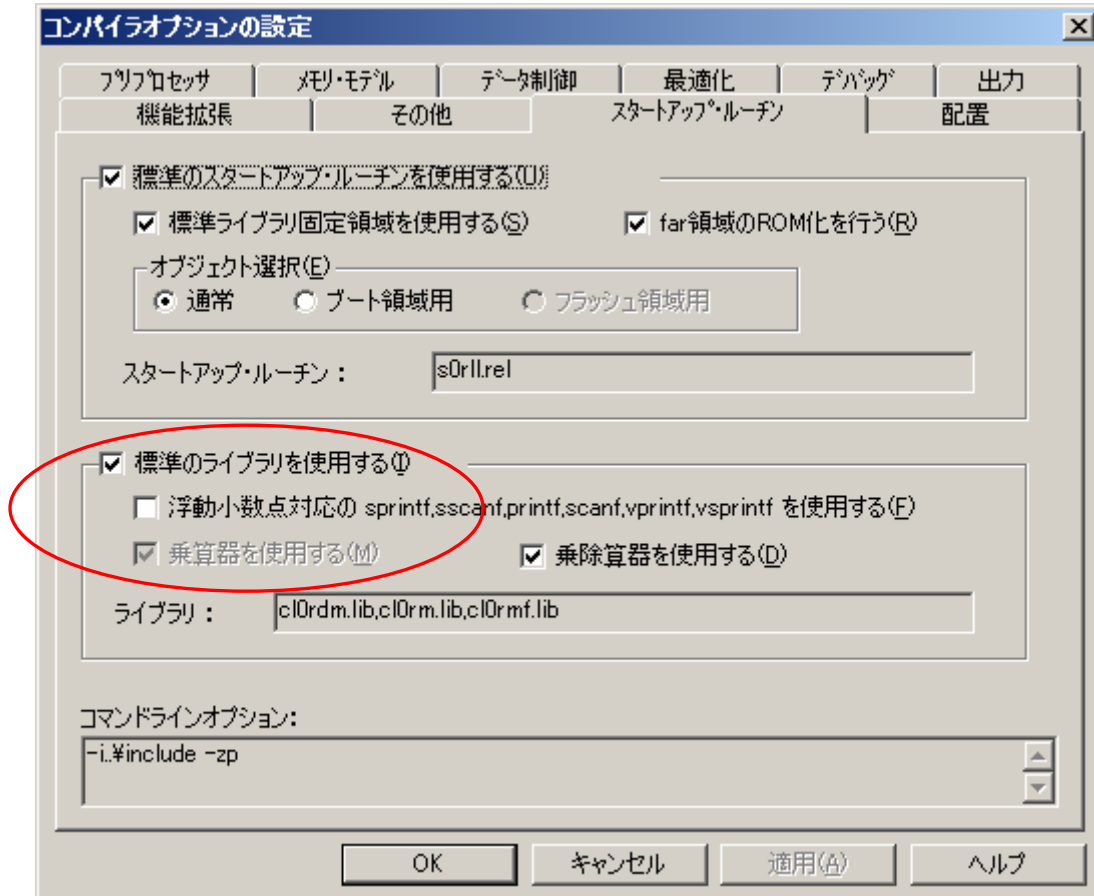
2.5.1 「機能拡張」タブ

「機能拡張」タブを選択し、「C++コメントの使用を許可する」のチェック・ボックスにチェックを入れます。このオプション設定により、Cソース・プログラム記述の中の”//”から改行までを、コメント文と認識できるようになります。プログラムをエディットする際、便利な機能です。




2.5.2 「スタートアップ・ルーチン」タブ

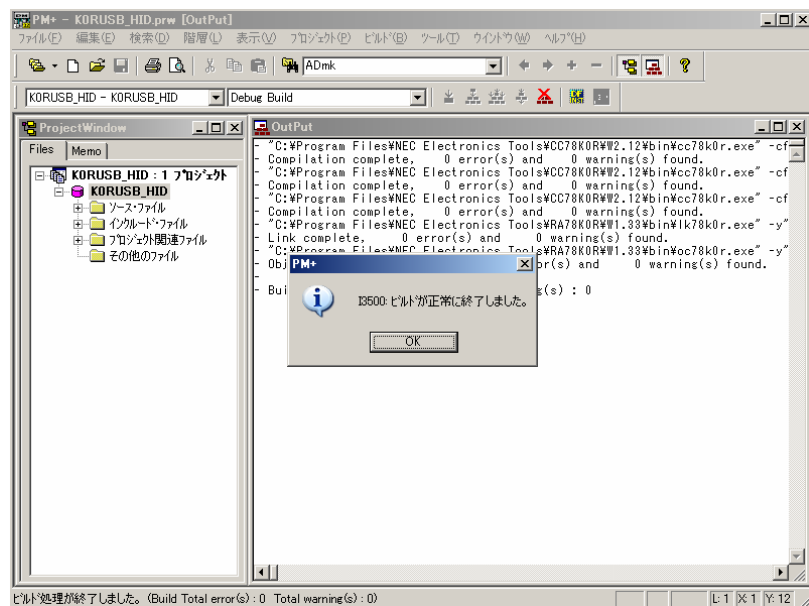
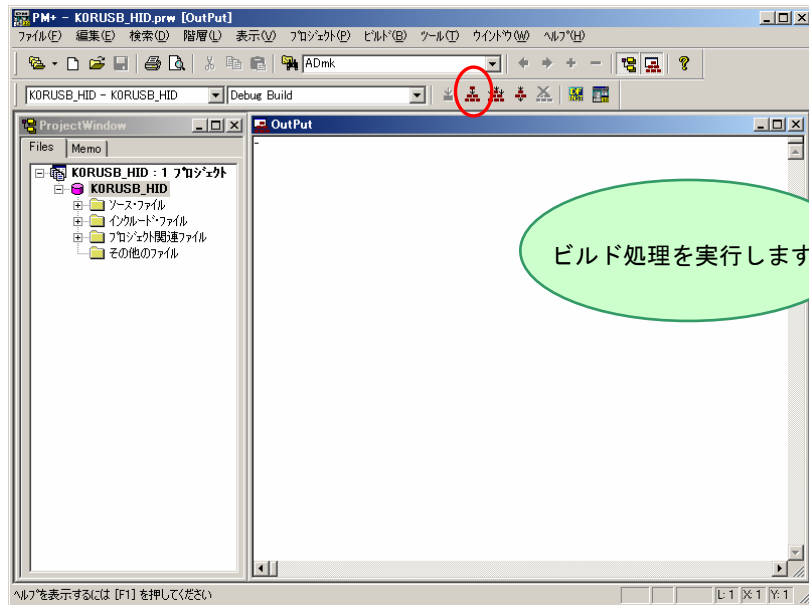
「スタートアップ・ルーチン」タブを選択し、「標準のライブラリを使用する」にチェックを入れ、さらに「乗算器を使用する」にチェックを入れます。78K0R/KE3-L は乗算器を内蔵しているため、このチェックを行うことにより乗算時の動作速度が高速になります。



2.6 ロード・モジュール・ファイルの作成

作成したプログラムをコンパイル、アセンブル、リンクしてロード・モジュール・ファイルを作成します。この1連の作業を**ビルド**と呼びます。

PM+のビルド・ボタン 、またはメニューの [ビルド(B)]→[ビルド(B)] を選択してください。



ビルド処理を正常に終了しました。

ビルドとは？

プロジェクトに登録されているソース・ファイルから実行形式ファイルなどを作成する機能です。
PM+がコンパイル、アセンブル、リンクなどを自動的に実行します。
また、PM+は、2度目以降のビルドでは、前回のビルドから更新されたファイルを自動的に検出し、該当するファイルのみをコンパイル、アセンブルすることにより、ビルドにかかる時間を短縮しています。

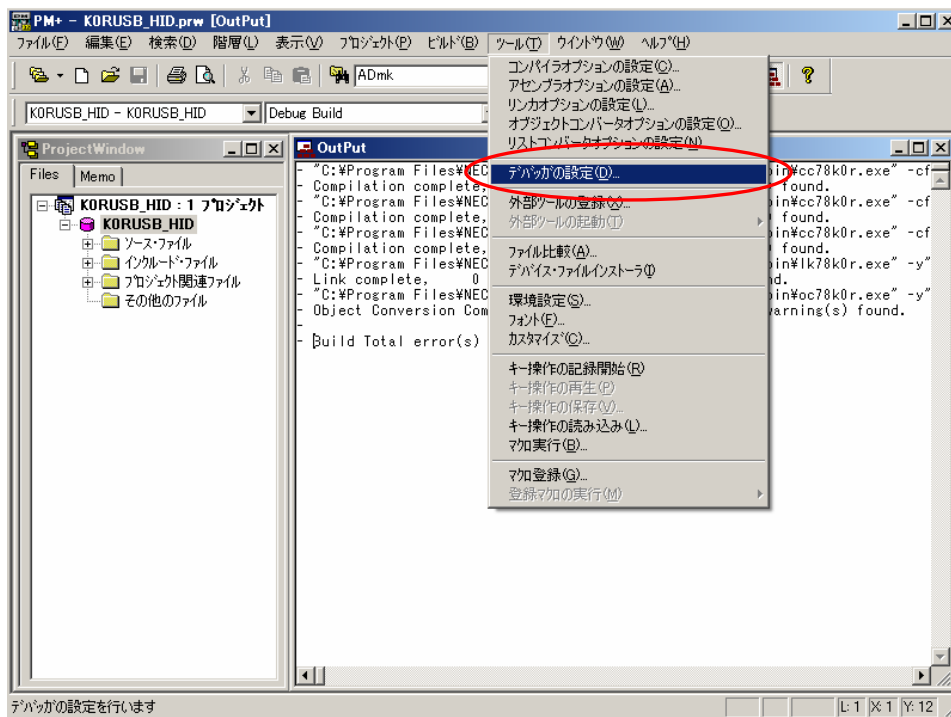
リビルドとは？

ビルドは、前回から更新されたソース・ファイルのみをコンパイル、アセンブルしますが、リビルドではすべてのソース・ファイルをコンパイル、アセンブルします。
コンパイラオプション等、各種設定を変更したときは、ビルドではなくリビルドを選択する必要があります。

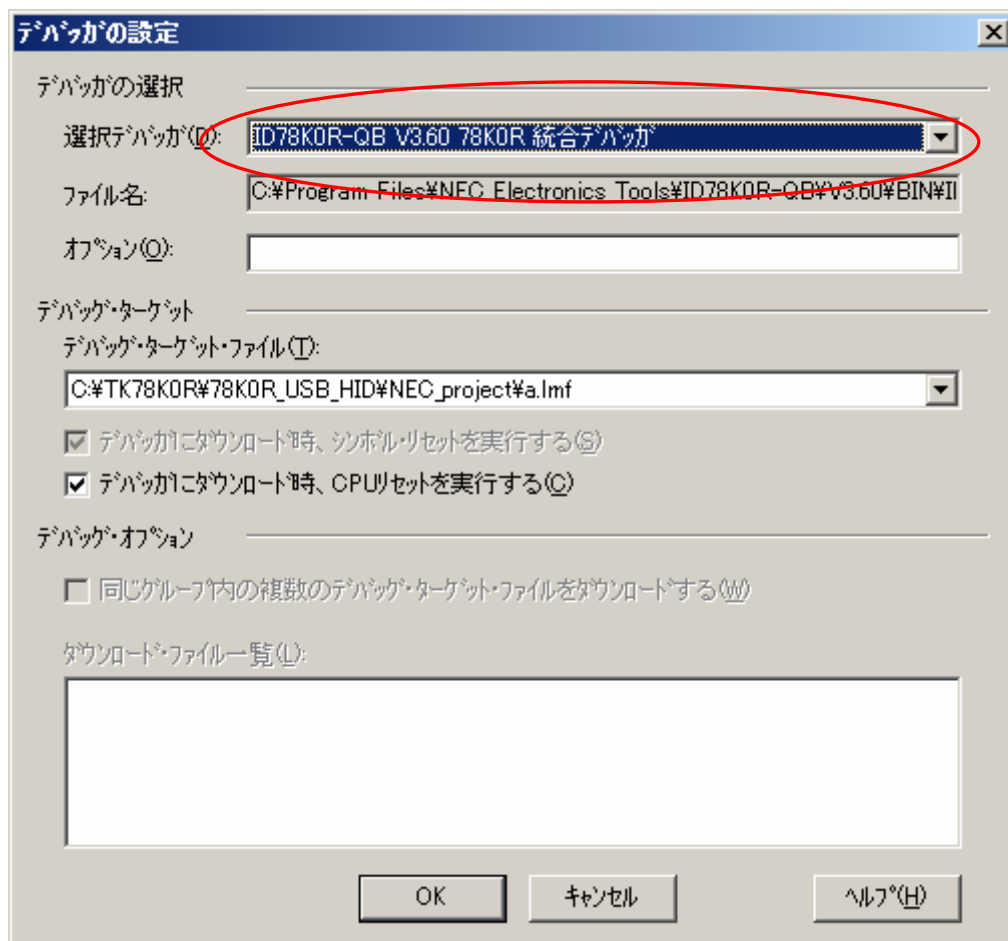
2.7 デバッガの設定確認

ビルドが終了したので、デバッグを行うためにデバッガの設定を行います。
こちらプロジェクト・ファイルで設定済みの内容ですが、デバッグを行なう上で必要な設定であるため、説明しておきます。

PM+メニュー・バーの[ツール(T)] → [デバッガの設定] を選択します。



「選択デバッガ」に「ID78K0R-QB V3.60 78K0R 総合デバッガ」が選択されていることを確認します。



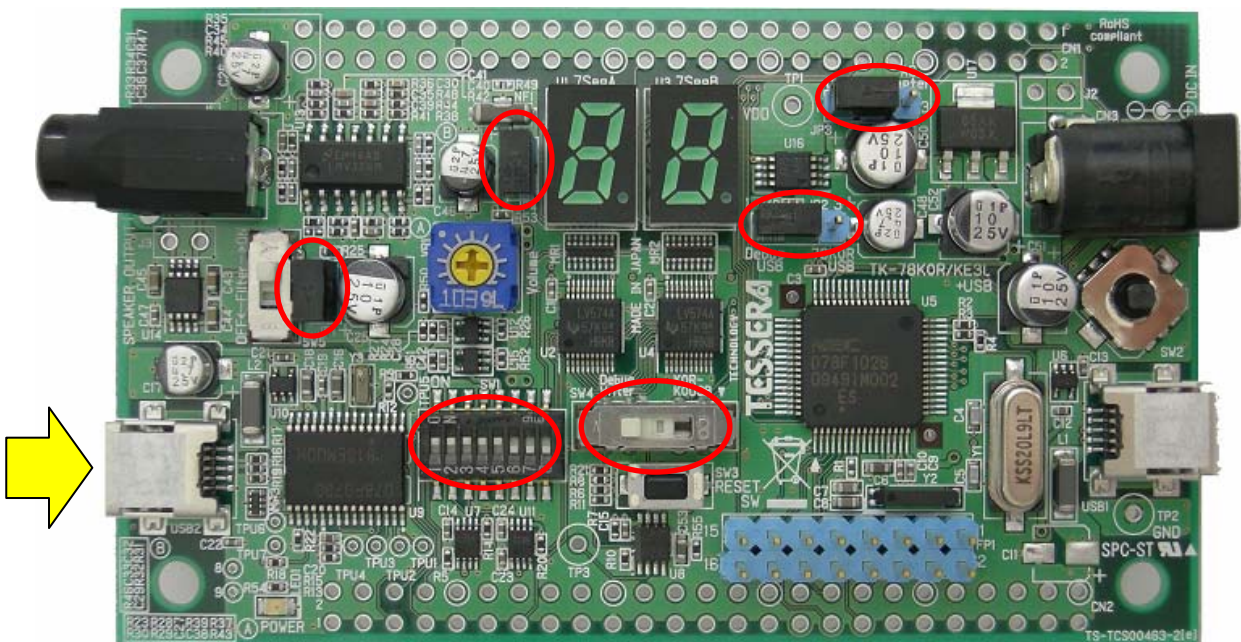
ここで、「ID78K0R-QB V3.60 78K0R 総合デバッガ」が選択できない場合は、メニュー・バーの [プロジェクト]→[プロジェクトの設定]→[ツールバージョンの設定]→[詳細設定]で“ID78K0R-QB”を選択してください。

2.8 キットの設定確認

以下の様にジャンパピンとスイッチの設定をします。

JP1	ショート
JP2	1-2 ショート
JP3	1-2 ショート
JP4	ショート
SW4	Debug Writer


SW1							
1	2	3	4	5	6	7	8
ON	ON	ON	ON	ON	OFF	OFF	OFF



設定が完了したら TK-78K0R/KE3L+USB の「USB2」コネクタと、パソコンの USB コネクタを、USB ケーブルで接続します。

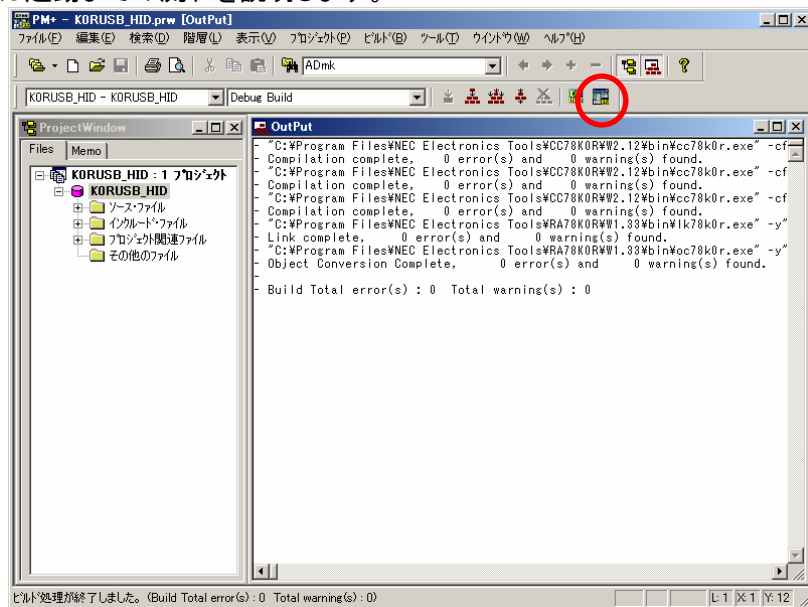
「新しいハードウェアの検出ウィザード」画面が表示される場合には、「[1.3 USB ドライバのインストール](#)」を参照し、USB ドライバのインストールを行ってください。

2.9 デバッガ(ID78K0R-QB)の起動

PM+のデバッグ・ボタン 、またはメニューの [ビルド(B)]→[デバッグ(D)] を選択してください。

ここで、デバッグ・ボタンが表示されていない場合は、「2.7 デバッガの設定確認」を参照し、設定を行ってください。

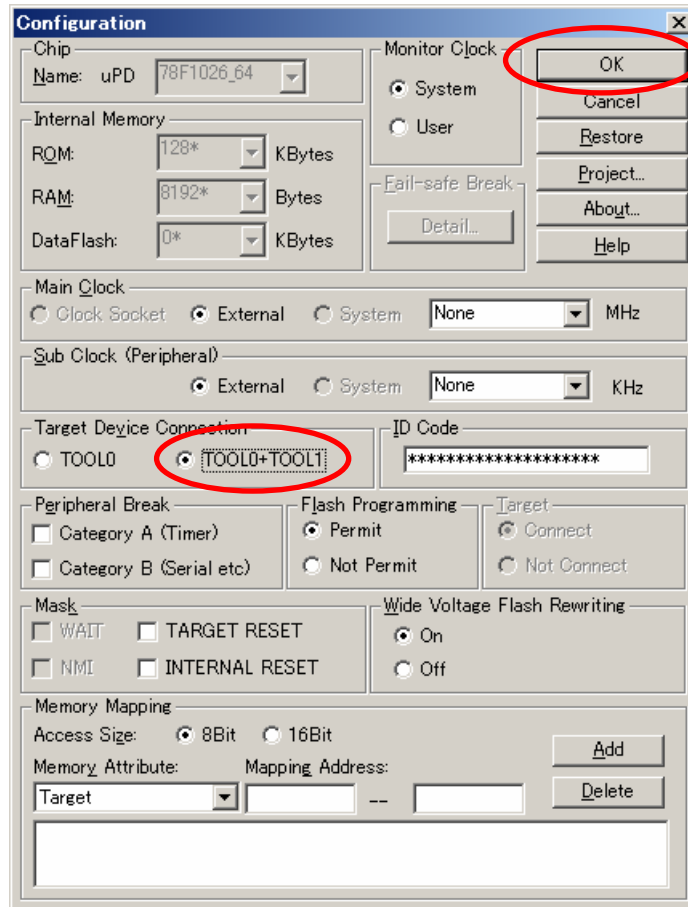
ここでは、デバッガ起動までの流れを説明します。



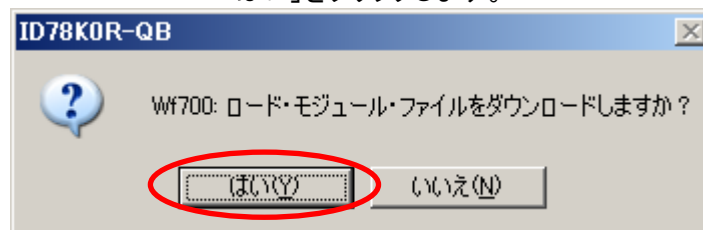
ID78K0R-QB が起動します。

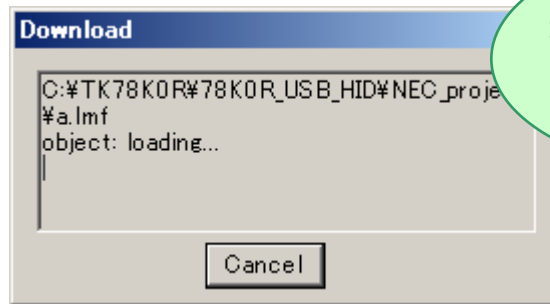


コンフィギュレーション・ダイアログが表示されるので、「ID Code」に「FFFFFFFFFFFFFFFFFFFFF」(F が 20 個)を入力し、「Target Device Connection」に「TOOL0+TOOL1」を選択して「OK」をクリックします。

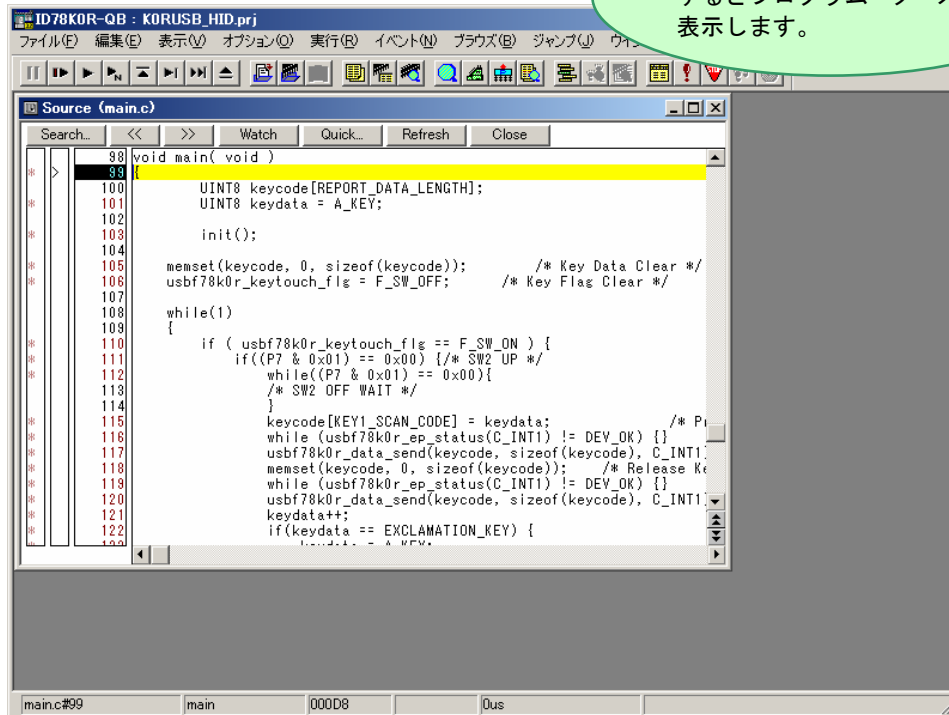


「ロード・モジュール・ファイル」のダウンロード確認画面が表示されるので、「はい」をクリックします。





ID78K0R-QB が起動し、プログラムをマイコン内蔵フラッシュ・メモリへダウンロードします。




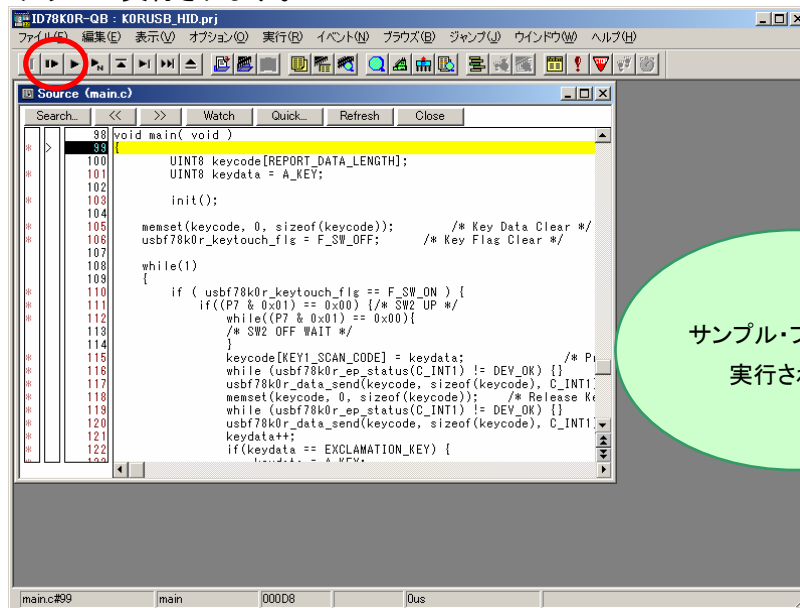
ダウンロードが正常に終了するとプログラム・ソースを表示します。

注意:この状態では、プログラムのダウンロード(内蔵フラッシュ・メモリへの書き込み)が完了しただけで、まだプログラムが実行されていないため、デモンストレーションは動作しません。デモンストレーションを動作させるには、次章「[2.10 プログラムの実行](#)」に進んでください。

2.10 プログラムの実行

プログラムを実行します。

ID78K0R-QB のリスタート・ボタン 、またはメニューの [実行(R)]→[リスタート(R)] を選択してください。サンプル・プログラムが実行されます。

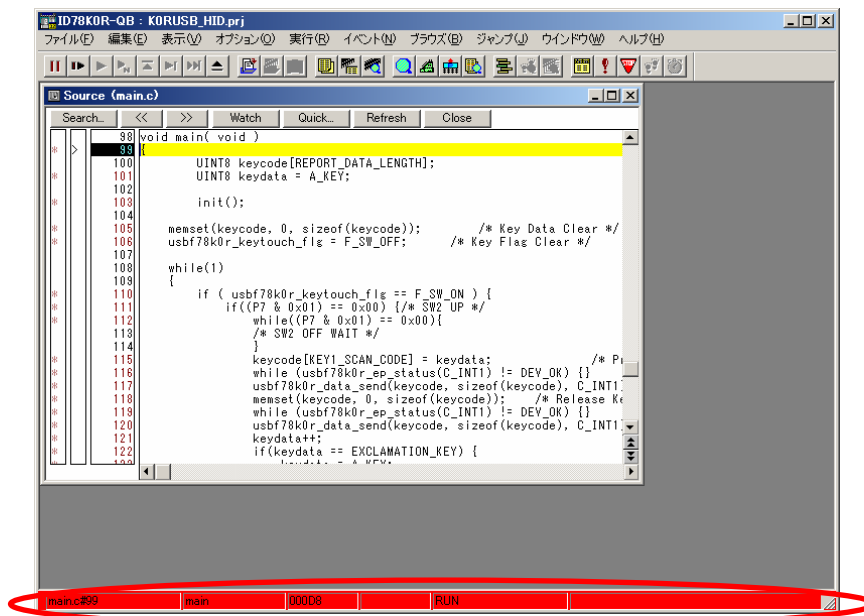


```

ID78K0R-QB : KORUSB_HID.prj
ファイル(F) 編集(E) 表示(V) オプション(O) 実行(R) イベント(I) プラウス(B) ジャンプ(J) ウインドウ(W) ヘルプ(H)

Source (main.c)
Search: << >> Watch Quick Refresh Close
39 void main( void )
100     UINT8 keycode[REPORT_DATA_LENGTH];
101     UINT8 keydata = A_KEY;
102
103     init();
104
105     memset(keycode, 0, sizeof(keycode)); /* Key Data Clear */
106     usbf78k0r_keytouch_flg = F_SW_OFF; /* Key Flag Clear */
107
108     while(1)
109     {
110         if ( usbf78k0r_keytouch_flg == F_SW_ON ) {
111             if((P7 & 0x01) == 0x00) { /* SW2 UP */
112                 while((P7 & 0x01) == 0x00){
113                     /* SW2 OFF WAIT */
114                 }
115                 keycode[KEY1_SCAN_CODE] = keydata; /* Pr
116                 while (usbf78k0r_ep_status(C_INT1) != DEV_OK) {}
117                 usbf78k0r_data_send(keycode, sizeof(keycode), C_INT1);
118                 memset(keycode, 0, sizeof(keycode)); /* Release Ke
119                 while (usbf78k0r_ep_status(C_INT1) != DEV_OK) {}
120                 usbf78k0r_data_send(keycode, sizeof(keycode), C_INT1);
121                 keydata++;
122                 if(keydata == EXCLAMATION_KEY) {
123                     /* ... */
124                 }
125             }
126         }
127     }
main.c#99 | main | 000D8 | 0us
  
```

サンプル・プログラムが
実行されます。

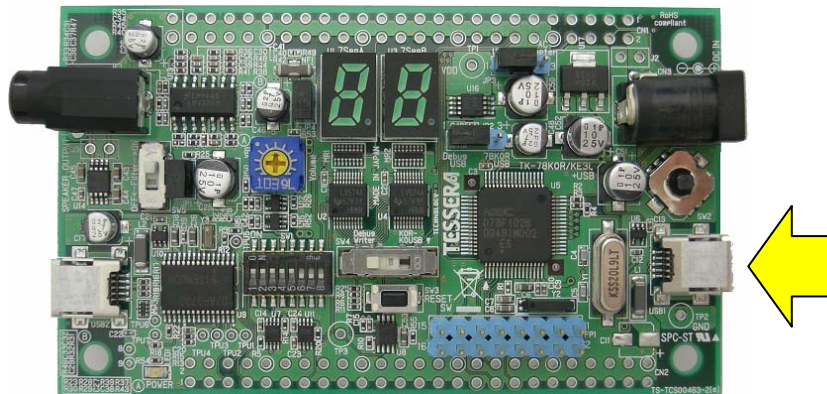
```

ID78K0R-QB : KORUSB_HID.prj
ファイル(F) 編集(E) 表示(V) オプション(O) 実行(R) イベント(I) プラウス(B) ジャンプ(J) ウインドウ(W) ヘルプ(H)

Source (main.c)
Search: << >> Watch Quick Refresh Close
39 void main( void )
100     UINT8 keycode[REPORT_DATA_LENGTH];
101     UINT8 keydata = A_KEY;
102
103     init();
104
105     memset(keycode, 0, sizeof(keycode)); /* Key Data Clear */
106     usbf78k0r_keytouch_flg = F_SW_OFF; /* Key Flag Clear */
107
108     while(1)
109     {
110         if ( usbf78k0r_keytouch_flg == F_SW_ON ) {
111             if((P7 & 0x01) == 0x00) { /* SW2 UP */
112                 while((P7 & 0x01) == 0x00){
113                     /* SW2 OFF WAIT */
114                 }
115                 keycode[KEY1_SCAN_CODE] = keydata; /* Pr
116                 while (usbf78k0r_ep_status(C_INT1) != DEV_OK) {}
117                 usbf78k0r_data_send(keycode, sizeof(keycode), C_INT1);
118                 memset(keycode, 0, sizeof(keycode)); /* Release Ke
119                 while (usbf78k0r_ep_status(C_INT1) != DEV_OK) {}
120                 usbf78k0r_data_send(keycode, sizeof(keycode), C_INT1);
121                 keydata++;
122                 if(keydata == EXCLAMATION_KEY) {
123                     /* ... */
124                 }
125             }
126         }
127     }
main.c#99 | main | 000D8 | RUN
  
```

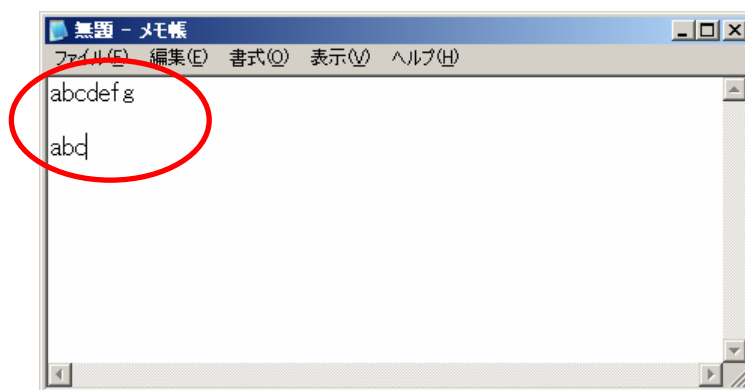
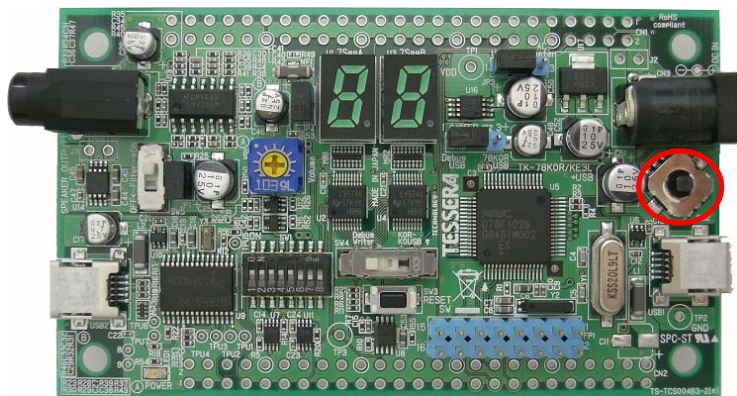
プログラムの実行中はステータス・バーが赤く変化します。

USB1 コネクタと PC を USB ケーブルで接続します。



「スタート」→「プログラム」→「アクセサリ」→「メモ帳」を開きます。

下図のように、「SW2」の「UP」を押すとアルファベットが入力され、「PUSH」で改行されることを確認します。




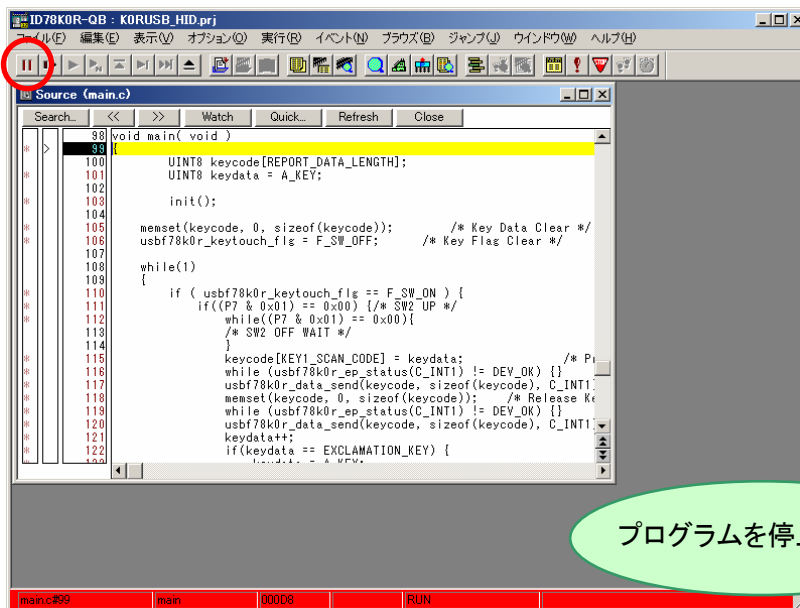
これでサンプル・プログラムが正常に動作していることが確認できました。

- サンプル・プログラムの詳細は、「USB HID(ヒューマン・インタフェース・デバイス)クラス ドライバ 編 アプリケーション・ノート」をご覧ください。

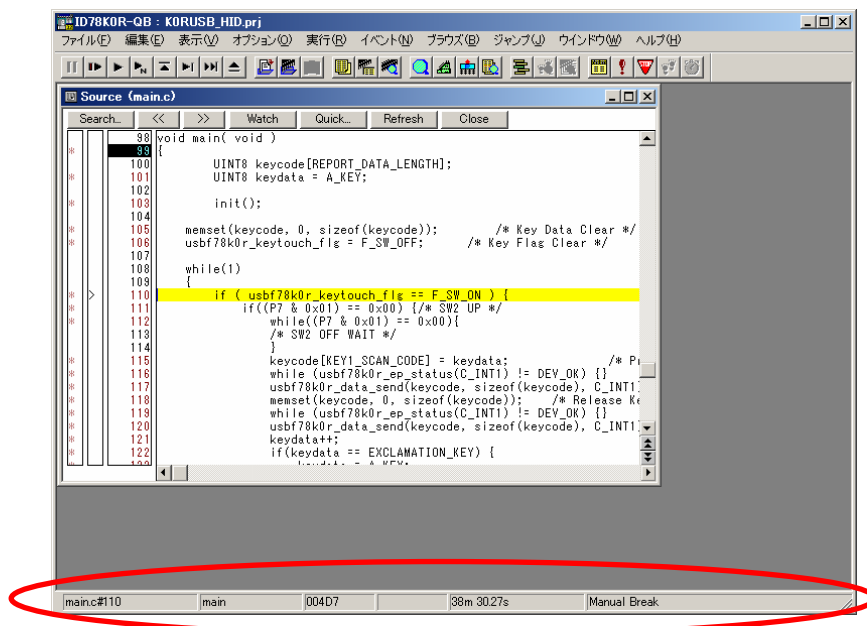
2.11 プログラムの停止

プログラムを停止します。

ID78K0R-QB の停止ボタン 、またはメニューの [実行(R)]→[ストップ(S)] を選択してください。



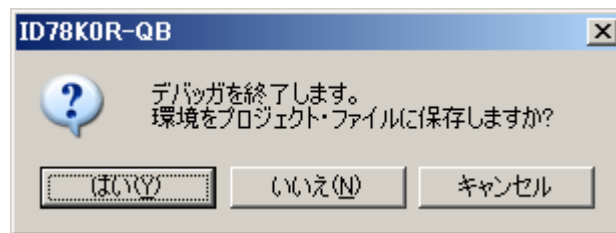
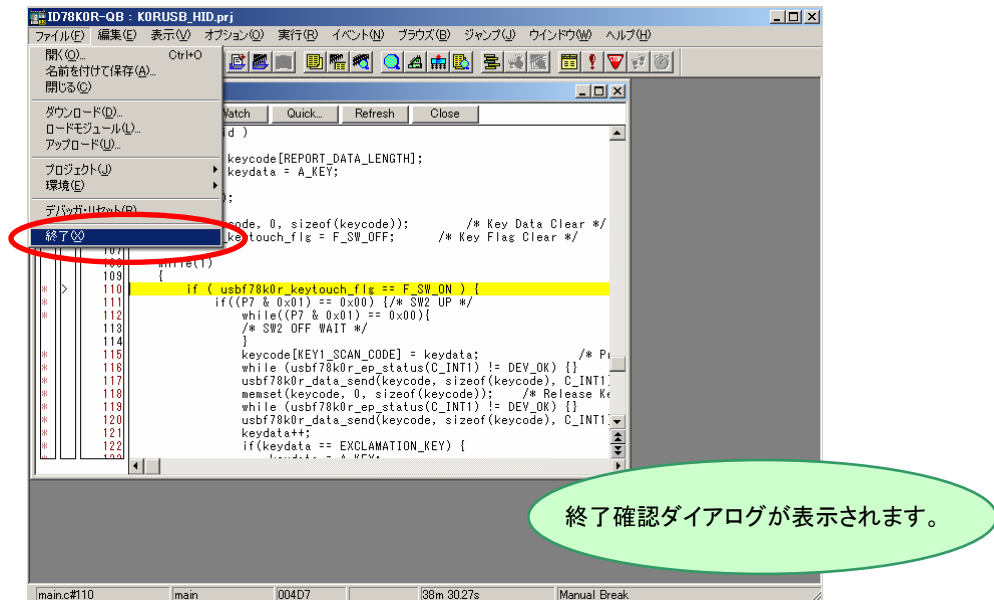
プログラムを停止します。



プログラムを停止すると、ステータス・バーの色が元に戻ります。

2.12 デバッガ(ID78K0R-QB)の終了

ID78K0R-QB メニューの [ファイル(F)]→[終了(X)] を選択してください。



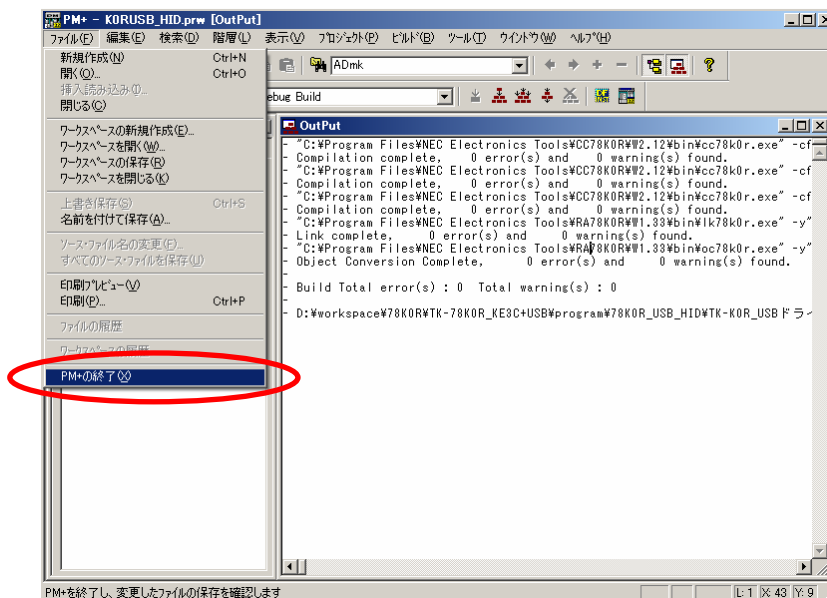
はい(Y) を押すと現在の環境を保存後、ID78K0R-QB が終了します。

開いたウィンドウの種類や、大きさ、位置などが保存されますので、保存することをお勧めします。

いいえ(N) を押すと現在の環境を保存しないで、ID78K0R-QB が終了します。

2.13 PM+の終了

PM+のメニューの [ファイル(F)]→[PM+の終了(X)] を選択してください。



PM+が終了します。

以上で体験編は終了です。

今回体験した以外にも、開発ツールの基本的な操作方法や便利な機能を「[5章 その他](#)」に記載しています。そちらも是非ご覧下さい。

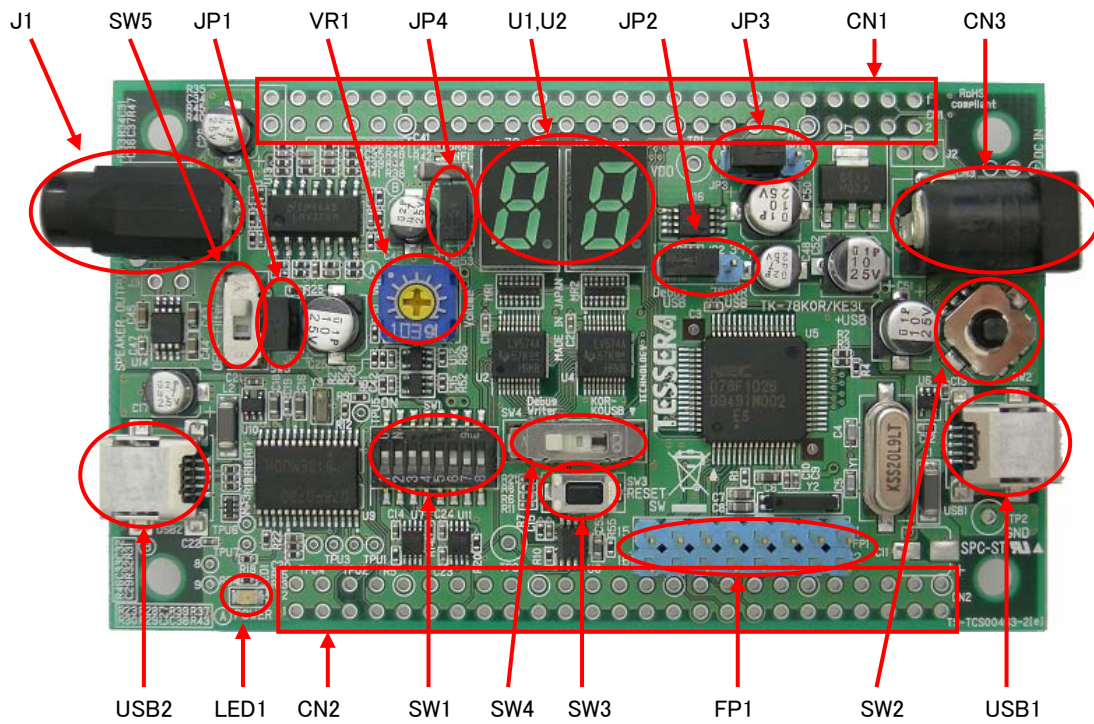
第3章 ハードウェア資料編

本章では、TK-78K0R/KE3L+USB が持つハードウェアに関して説明します。

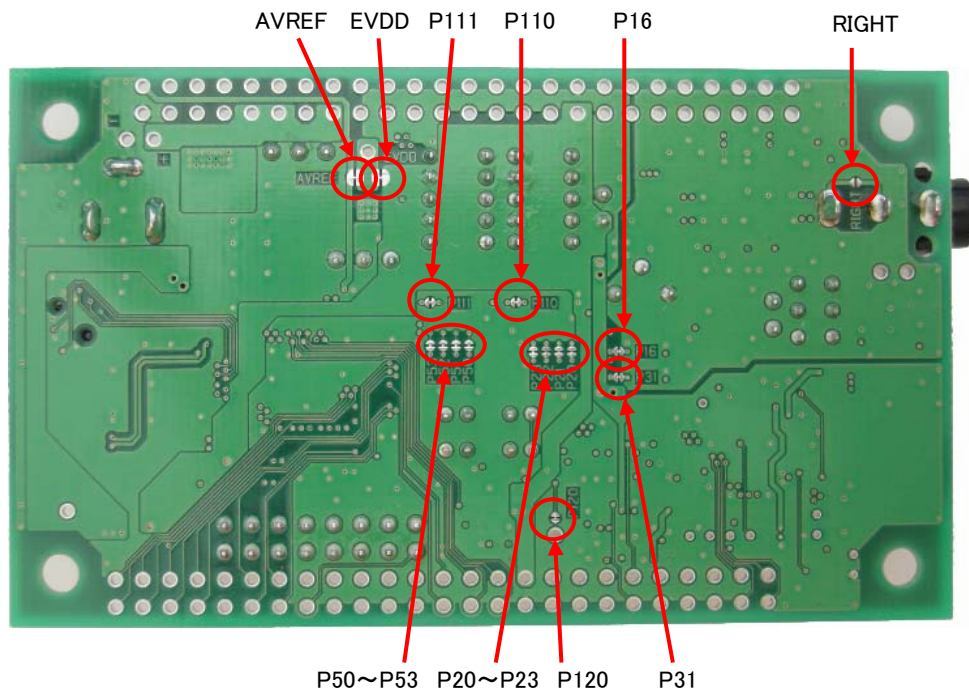
マイコン	μ PD78F1026 ※78K0R/KE3-L(USB)
動作周波数	外部メイン・システム・クロック: 20MHz サブシステム・クロック: 32.768KHz 高速内蔵発振クロック: 20、8、1 MHz
Audio	PWM 出力 Filter: LMV324M:NS(U13) Power AMP: LM4865M:NS(U14) Speaker 出力: 0.3W(簡易モニタ用) Jack: φ 3.5mm モノラルフォノジャック 音声フィルタ ON/OFF スイッチ(SW5) 音量調整ボリューム(VR1)
インターフェイス	USB MINI B (USB1) USB Function USB MINI B (USB2) デバッグ・シリアル通信用 拡張コネクタ (CN1,2) コネクタ未実装 MINICUBE2 接続用コネクタ(FP1)
供給電圧	5V (USB または AC アダプタ)
動作確認用 ハードウェア	・4 方向+センター・プッシュ付きスイッチ(SW2) ・ディップ・スイッチ(SW1) ・7Seg-LED(U1, U2) ・リセット・スイッチ(SW3)

※()内はキット上のシルク(表示名)を示しています。

3.1 部品配置図



3.2 テスト用パッド配置図



3.3 各 부품の説明

3.3.1 SW1、SW4

SW1 の bit1~3 はデバッグモード設定、bit4,5 は動作モード設定、bit6~8 はマイコンの P41,P42,P17 端子に接続されている Dip スイッチです。

- 本製品付属の ID78K0R-QB を使用する場合は以下の設定にしてください。

SW1

Bit 1	ON/OFF ※1
Bit 2	ON
Bit 3	ON

SW4 Debug/Writer 側

- ※1 ON : 電源導入と同時にマイコンはフラッシュ・メモリに書き込まれているプログラムを実行します。
OFF : ID78K0R-QB が起動するまでマイコンはリセット状態になります。

- ID78K0R-QB を使用せずにマイコン内蔵フラッシュ・メモリに書き込んだプログラムを実行する場合、及び MINICUBE2 を接続してデバックする場合には、以下の設定にして電源を入れ直してください。

SW1

Bit 1	OFF
Bit 2	OFF
Bit 3	OFF

SW4 K0R <--> K0USB 側 または センター

- ※ P11,P12 を拡張コネクタで使用する場合には SW4 をセンターに設定してください。

- Bit4 は POWER LED 点灯切り替えスイッチです。

ON	マイコンに電源が供給されると LED1 が点灯します。
OFF	マイコンに電源が供給されても LED1 は点灯しません。

- Bit5 は音声ミュート切り替えスイッチです。

ON	P31 から Low 出力すると、音声出力をミュートすることができます。
OFF	音声出力をミュートしません。

- bit6～8 はマイコンのポートに接続されています。ON で”Low”、OFF で”Open”になりますので、使用する場合はマイコン内蔵のプルアップ抵抗オプション・レジスタ (PU1, PU4)を ON に設定する必要があります。(マイコン内蔵のプルアップ抵抗オプション・レジスタ設定の詳細は、78K0R/KE3-L のユーザーズ・マニュアル(U19878J)を参照してください。)

SW1

Bit 6	P42
Bit 7	P43
Bit 8	P17

3.3.2 SW2

SW2 は 4 方向 + センタープッシュ付きスイッチです。押すまたは方向が指定されると”Low”、離すと”Open”になりますので、使用する場合にはマイコン内蔵のプルアップ抵抗オプション・レジスタ(PU7)を ON に設定する必要があります。(マイコン内蔵のプルアップ抵抗オプション・レジスタ設定の詳細は、78K0R/KE3-L のユーザーズ・マニュアル(U19878J)を参照してください。)

SW1 端子表 (ALPS SKRHADE010)

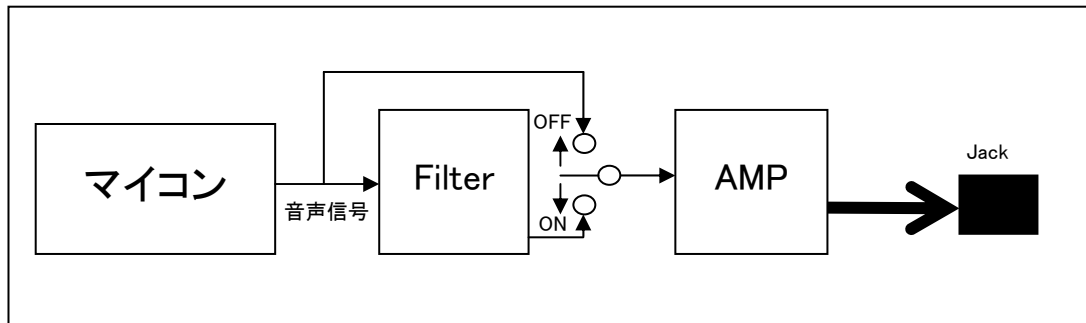
SW1	信号名	接続先マイコン端子名称	備考
1pin	P70	P70 /KR0	UP
2pin	P71	P71 /KR1	CENTER PUSH
3pin	P72	P72 /KR2	LEFT
4pin	P74	P74/KR4/INTP8	RIGHT
5pin	GND	GND	
6pin	P73	P75/KR5/INTP9	DOWN

3.3.3 SW3 (RESET SW)

SW3 はリセットスイッチです。
押すことによってマイコンをリセットできます。

3.3.4 SW5 (Filter)

SW5 は Filter(LMV324M)の使用／未使用を選択するスライド・スイッチです。”OFF”側に設定すると Filter を使用せずにマイコンからの音声信号は AMP に入力されます。”ON”側に設定するとマイコンからの音声信号は Filter を経由してから AMP に入力されます。



3.3.5 JP1

マイコンからの音声信号を Filter へ接続しているジャンパーショートピンです。通常はショートに設定して使用してください。詳細は回路図をご覧ください。

3.3.6 JP2、JP3

JP2、JP3 は電源供給元の切り替えジャンパーショートピンです。

JP2	JP3	
1-2 ショート	1-2 ショート	USB2 コネクタに接続した USB 電源を使用します。
2-3 ショート	1-2 ショート	USB1 コネクタに接続した USB 電源を使用します。
-	2-3 ショート	CN3 に接続した AC アダプタから電源を供給します。

3.3.7 JP4

音声回路への電源供給ジャンパーショートピンです。通常はショートに設定して使用してください。詳細は回路図をご覧ください。

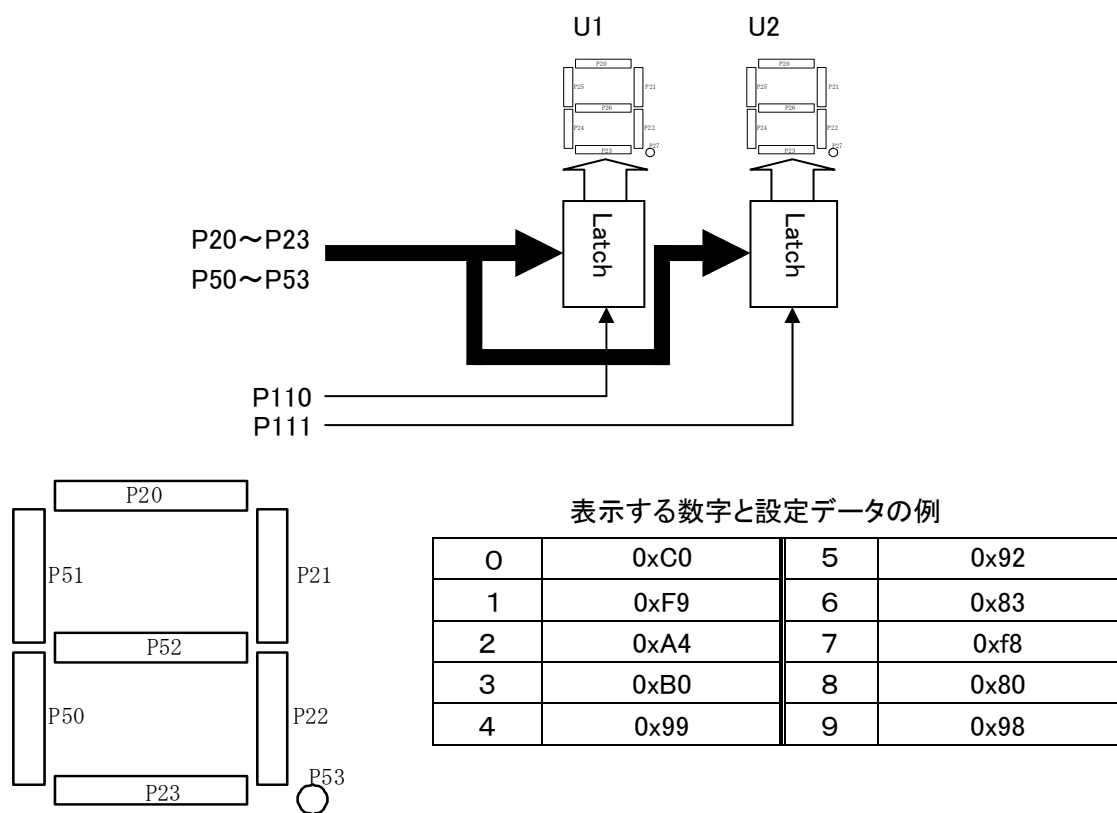
3.3.8 LED1 (POWER)

POWER LED です。マイコンに電源が投入されると点灯します。
SW1 の bit4 を OFF にすることで消灯することができます。

3.3.9 U1,U2 (7Seg-LED)

U1 と U2 は 7segLED です。

P20~P23、P50~P53 に 7segLED に出力するデータをセットして、P110 または P111 を Low から High にすることで、データがラッチされ、7segLED を点灯することができます。



U1 に「1」を U2 に「2」を表示する場合

```

PM2 = 0xf0;           // P2 の下位 4bit を出力モードに設定
PM5 = 0xf0;           // P5 の下位 4bit を出力モードに設定
PM11.0 = 0; PM11.1 = 0; // P110, P111 を出力モードに設定
P2 = 0xf9; P5 = 0xf9>>4; // P2, P5 に「1」のデータを設定
P11.0 = 0; P11.1 = 1; // P110 を Low にして High へ
P2 = 0xa4; P5 = 0xa4>>4; // P2, P5 に「2」のデータを設定
P11.1 = 0; P11.1 = 1; // P111 を Low にして High へ

```

3.3.10 CN1, CN2

CN1, CN2 は拡張用コネクタです。
コネクタは未実装です。(ヒロセ製 HIF-3H-50DA-2.54DSA)

3.3.11 CN3

AC アダプタ接続用のコネクタです。
AC アダプタ(+5V)を接続してください。

・適合プラグ : ϕ 2.1mm DC プラグ(センタープラス)

3.3.12 J1

外部スピーカー接続用の Jack です。音声の再生時はここにスピーカーを接続してください。

・適合プラグ : ϕ 3.5mm フォノプラグ (モノラル)

3.3.13 VR1

音声出力の音量コントロールです。
P31 出力によってミュートがされている場合にはミュートが有効になります。

3.3.14 USB1

USB ファンクション・コントローラ用 USB コネクタです。

3.3.15 USB2

デバッグ、シリアル通信用 UBS コネクタです。
「NEC Electronics Starter Kit Virtual UART」として認識されます。

3.3.16 FP1

MINICUBE2 を接続するためのインターフェイスです。

3.4 半田ショートパッド

半田ショートパッドを利用して回路を切り離すもしくは接続することにより、ボード上の回路をカスタマイズすることができます。

半田ショート用パッドは、下図のような形をしています。

オープンにする場合には、パッドのくびれている部分をカッター等で切断してください。ショートする場合には、半田ごて等を使用して半田ショートパッドに半田を盛ってショートしてください。

半田ショートパッド
(オープン時の形状)



半田ショートパッド
(ショート時の形状)



半田ショートパッド名	出荷時状態	接続	
		ショート時	オープン時
P120	ショート	ショート時	P120 へ USB1 のバスパワー検出信号を接続
		オープン時	P120 を汎用入出力として使用可能
P16	ショート	ショート時	P16 を音声回路へ接続
		オープン時	P16 を汎用入出力として使用可能
P31	ショート	ショート時	P31 を音声ミュート回路へ接続
		オープン時	P31 を汎用入出力として使用可能
AVREF	ショート	ショート時	VDD = AVREF
		オープン時	VDD 端子と異なる電圧を AVREF 端子へ供給可能
EVDD	ショート	ショート時	VDD = EVDD
		オープン時	VDD 端子と異なる電圧を EVDD 端子へ供給可能
P110,P111	ショート	ショート時	P110,P111 を 7SegLED データラッチ回路へ接続。
		オープン時	P110,P111 を汎用入出力として使用可能
P20~P23、 P50~P53	ショート	ショート時	7segLED データラッチ回路に接続
		オープン時	汎用入出力として使用可能
RIGHT	オープン	ショート時	J1 の外部スピーカー出力をステレオコネクタ用接続にします。
		オープン時	モノラルコネクタ用接続です。

備考: グランド端子はキット内で全てショートされています。

第4章 困った時は？

本章では、トラブルに関して対処方法を説明します。

4.1 パソコンとキットを接続時、USB ドライバが見つからない

チェック・ポイント 1

パソコンとキットの間に USB ハブを使用している場合、ハブを排除してください。(USB ハブ使用時の動作は保証していません。)

チェック・ポイント 2

「[1.2 開発ツールのインストール](#)」時、「Starter Kit USB Driver」をインストールしたか確認してください。インストールしていなければインストールしてください。

チェック・ポイント 3

「[1.3 USB ドライバのインストール](#)」を参考にして、キット上のスイッチ設定が正しいかどうか確認してください。

チェック・ポイント 4

上記 3 つをチェックして問題が無ければ、一度パソコンから USB ケーブルを抜き、再度パソコンとキットを接続してください。「新しいハードウェアの検出ウィザード」が出ますが、以降の操作は「[1.3 USB ドライバのインストール](#)」を参考にして、確実に行ってください。また、インストール完了後、必ず「[1.3.3 USB ドライバのインストール完了確認](#)」を行ってください。

4.2 デバッグを起動すると、エラーが出る

これにはいくつか要因があります。

エラーの種類に応じて対処が異なりますので、まずエラーの内容を確認してください。

以下、エラーの内容ごとに対処方法を説明します。

4.2.1 「インサーキット・エミュレータと通信できません。・・・」(F0100 or A0109)

チェック・ポイント 1

パソコンとキットの間に USB ハブを使用している場合、ハブを排除してください。(USB ハブ使用時の動作は保証していません。)

チェック・ポイント 2

[「1.3 USB ドライバのインストール」](#)を参考にして、キット上のスイッチ設定が正しいかどうか確認してください。

チェック・ポイント 3

[「1.3.3 USB ドライバのインストール完了確認」](#)を参考にして、USB ドライバのインストールが正しく完了しているかどうか確認してください。

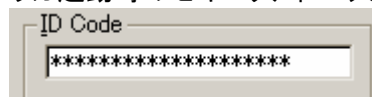
チェック・ポイント 4

上記 3 つをチェックして問題が無ければ、デバッグを一度終了し、パソコンから USB ケーブルを抜いてください。その後、再度 USB ケーブルがパソコン側、キット側ともに十分に挿入されていることを確認した上でデバッグを再度起動してください。

4.2.2 「ID コードが間違っています。」(Ff603)

マイコン内蔵のフラッシュ・メモリが記憶しているセキュリティIDと、デバッグ起動時に入力したIDコードが一致しない場合に出るエラーです。

デバッグ起動時のセキュリティID 入力欄



チェック・ポイント 1

セキュリティIDとして正しい値を入力し、コンフィギュレーション・ウィンドウの[OK]ボタンを押してください。

チェック・ポイント 2

セキュリティID が分からない場合、マイコン内蔵のフラッシュ・メモリを一度消去する必要があります。ただし、消去する前に「[2.4 リンカオプションの設定確認](#)」を参考にして、セキュリティID の設定を行っているか確認してください。また、ここで設定したセキュリティID の値を覚えておいてください。

その後、「[5.4 マイコン内蔵のフラッシュ・メモリを消去したい](#)」を参考にしてフラッシュ・メモリを消去してください。

4.2.3 「このデバイスではオンチップ・デバッグが禁止されています。」(F0c79)

マイコン内蔵フラッシュ・メモリの C3H 番地(オンチップ・デバッグ・オプション・バイト)に設定されている値が正しくないため、デバッグが起動できないというエラーです。この場合、フラッシュ・メモリを一度消去する必要があります。

チェック・ポイント 1

「[2.4 リンカオプションの設定確認](#)」を参考にして、オンチップ・デバッグ・オプション・バイト設定エリアの内容が正しいかどうかを確認してください。正しくない場合は正しい設定をしてください。

チェック・ポイント 2

「[5.4 マイコン内蔵のフラッシュ・メモリを消去したい](#)」を参考にして、フラッシュ・メモリを消去してください。

4.2.4 「オンチップ・デバッグを使用禁止にしようとした。」(F0c33)

おもに「[2.4 リンカオプションの設定確認](#)」が正しく行なわれていない状態でデバッグを起動(ダウンロード)した場合に発生するエラーです。「[4.2.3 「このデバイスではオンチップ・デバッグが禁止されています。」\(F0c79\)](#)」と同じ対処を行なってください。

第5章 その他

この章では、お客様自身でプログラムを開発するに当たり、開発ツールを使用する上で必要な操作や便利な操作についての解説、および本キットの回路図を示します。

[6.1 ワークスペース\(プロジェクト\)の新規作成](#)

[6.2 ソース・ファイルの追加登録方法](#)

[6.3 デバッガの便利な機能](#)

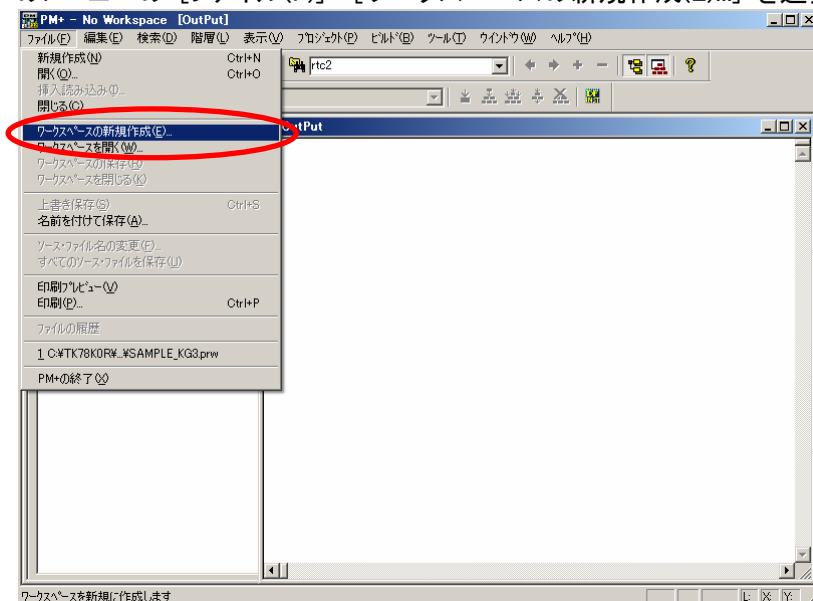
[6.4 回路図](#)

5.1 ワークスペースの新規作成方法

ワークスペースおよびプロジェクトを新規作成します。

PM+では、ビルドに必要な基本項目を含んだプロジェクトを、ステップに沿って項目を入力していただくだけで、簡単に新規作成することができます。

PM+のメニューの [ファイル(F)]→[ワークスペースの新規作成(E)...] を選択してください。



ワークスペースの新規作成
ダイアログが開きます。

<各項目の説明>

ワークスペース・ファイル名(W)

→ プロジェクト・ファイルを管理するワークスペース・ファイル名を指定します。
ファイルタイプは、.prw が自動的に付きます。
また、同名のプロジェクト・ファイル(.prj)を同時に作成します。

フォルダ位置(F)

→ ワークスペース・ファイルを保存するフォルダを絶対パスで指定します。
参照(R)... ボタンを押すと、参照ダイアログから選択できます。

プロジェクト・グループ名(G)

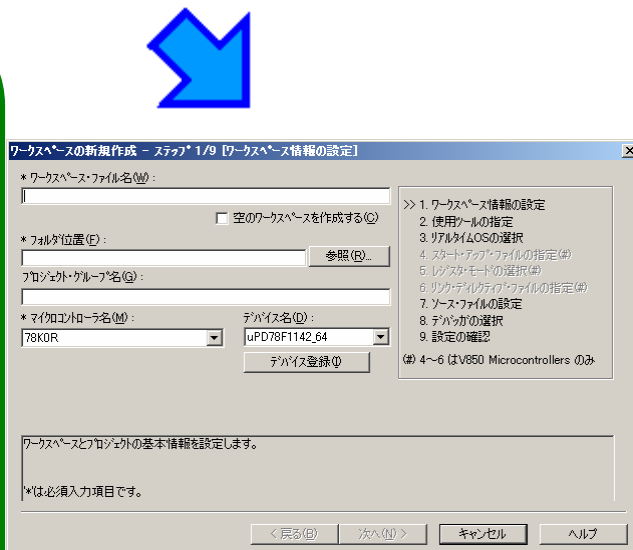
→ 機能単位で複数プロジェクトをまとめて管理したい場合に指定します。
指定しない場合は、ワークスペース・ファイル名と同じになります。

マイクロコントローラ名(S)

→ 使用するデバイスのマイクロコントローラ名を指定します。

デバイス名(D)

→ 使用するデバイス名を指定します。



次のページで、ここで設定する
具体的な内容を記述しています。

ワークスペース情報の設定を、次のように入力してください。

ワークスペース・ファイル名(W)

→ test

フォルダ位置(F)

→ C:¥TK78K0R

プロジェクト・グループ名(G)

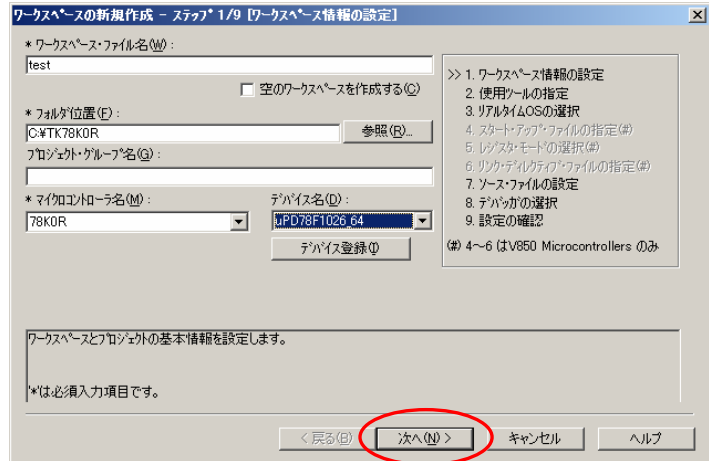
→ 入力しません

マイクロコントローラ名(S)

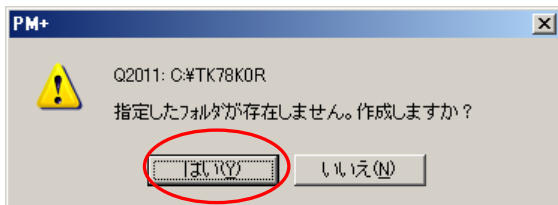
→ 78K0R

デバイス名(D)

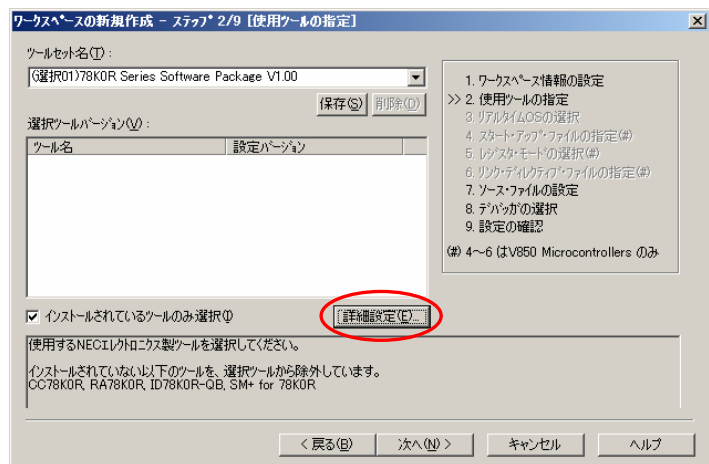
→ uPD78F1026_64



次へ(N) > ボタンを押してください。



はい(Y) ボタンを押してください。



詳細設定 ボタンを押してください。

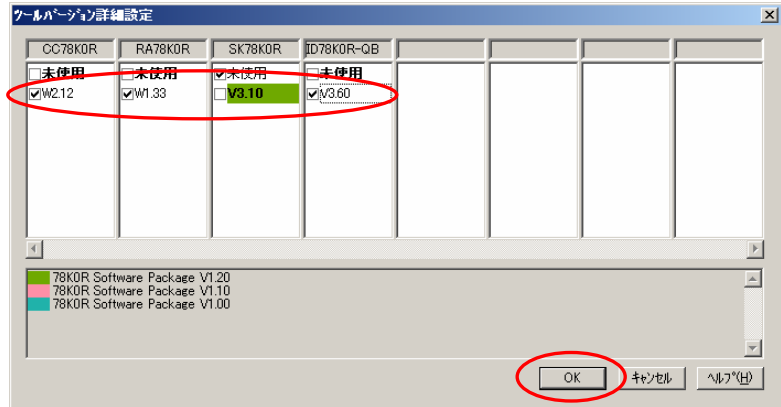


使用する各ツールのバージョンを次のように設定します。

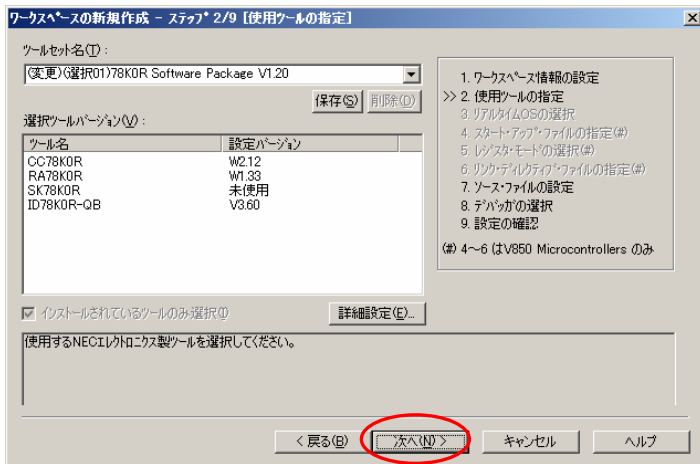
CC78K0R: W2.12

RA78K0R: W1.33

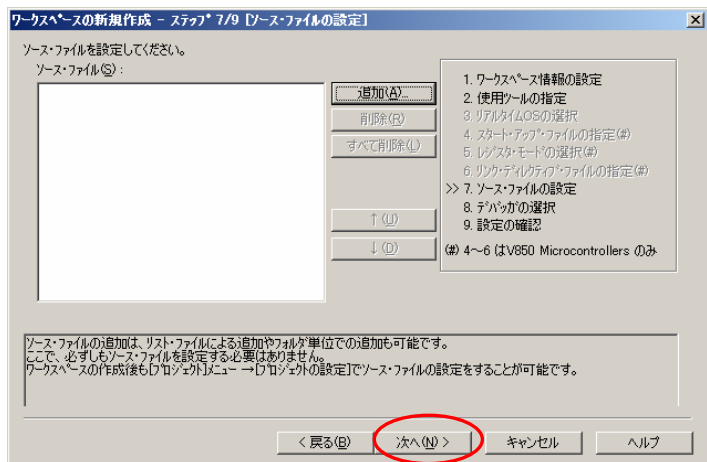
ID78K0R-QB: V3.60



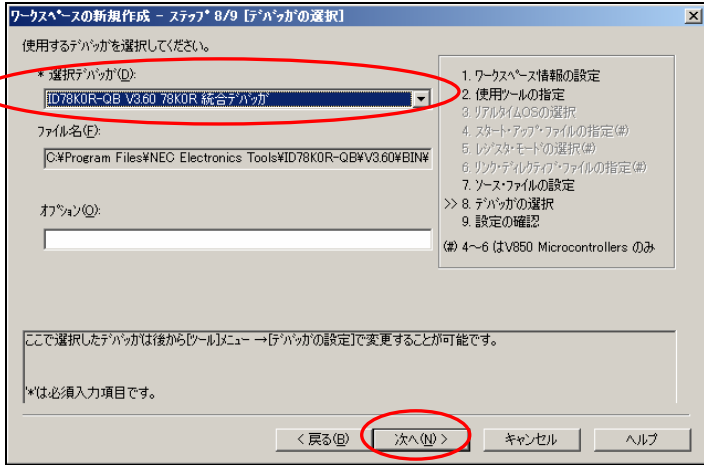
上記の様にツールを選択し、**OK**ボタンを押してください。



次へ(N) > ボタンを押してください。

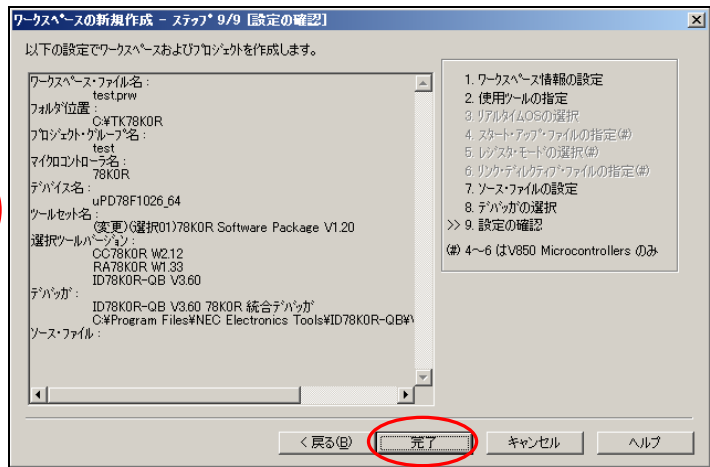


次へ(N) > ボタンを押してください。



ID78K0R-QB V3.60 を選択してください。

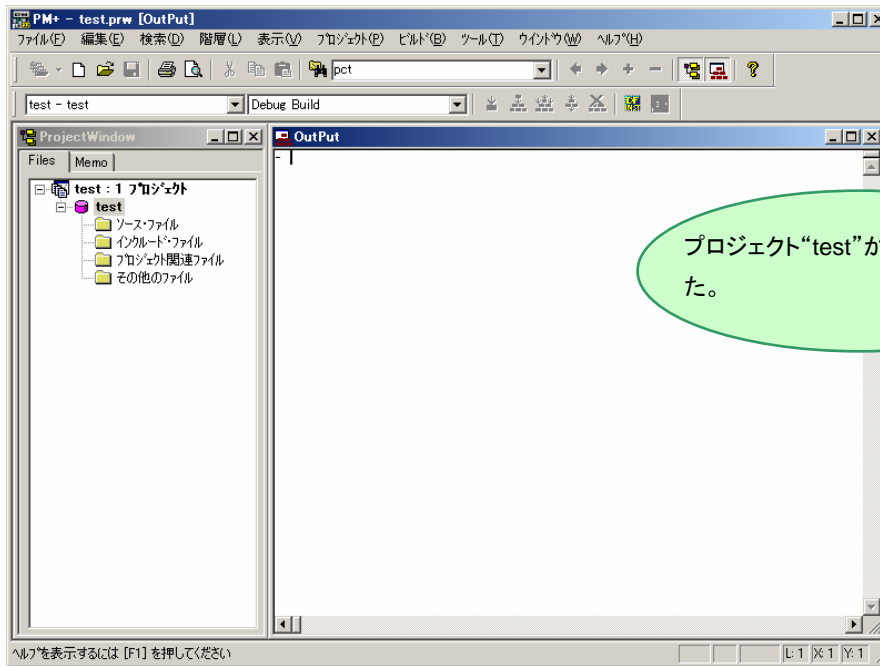
次へ(N) > ボタンを押してください。



プロジェクト情報の設定内容を
確認します。

完了 ボタンを押してください。





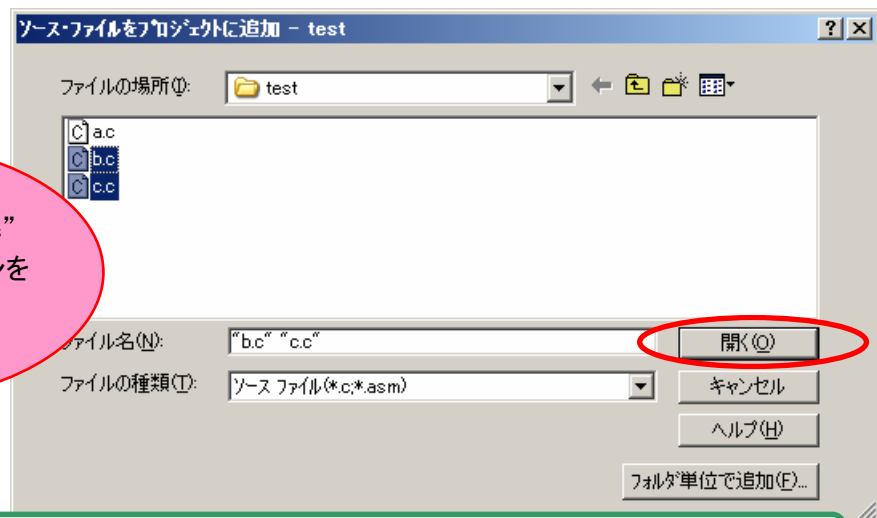
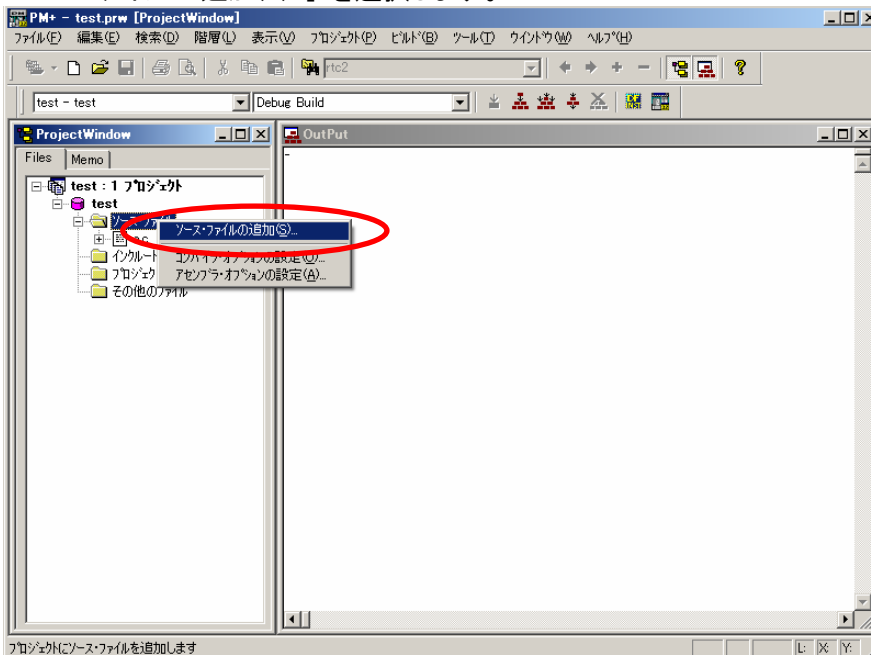
これで、ワークスペースおよびプロジェクトの作成は完了です。
ソース・ファイルは、後から随時追加登録することができます。
詳細については、「[ソース・ファイルの追加登録方法](#)」をご覧ください。
また、オンチップ・デバッグ機能を使用するための設定等も必要になりますので、「[2.4 リンカオプションの設定確認](#)」、「[2.5 コンパイラオプションの設定確認](#)」、「[2.7 デバッガの設定確認](#)」も合わせてご覧ください。

5.2 ソース・ファイルの追加登録方法

プロジェクトにソース・ファイルを追加登録する方法を紹介します。

ここでは、ソース・ファイル“a.c”が登録されている状態で、ソース・ファイル“b.c”と“c.c”を追加登録する例を示します。

PM+のプロジェクト・ウインドウのソース・ファイルにカーソルをあわせて、右クリックメニューで表示される「ソース・ファイルの追加(S)...」を選択します。

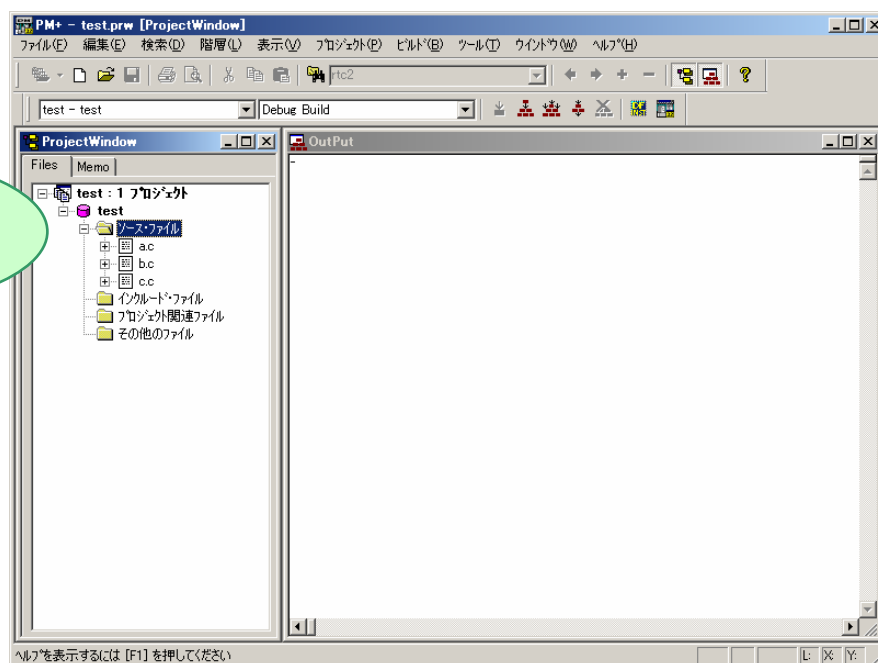


ソース・ファイル“b.c”と“c.c”を選択し、開く(O) ボタンを押します。

Ctrl キーを押しながらソース・ファイルを選択すると、複数個を同時に選択できます。



プロジェクトにソース・
ファイル“b.c”と“c.c”が
追加登録されます。



5.3 デバッガの便利な機能

本章では、デバッガ(ID78K0R-QB)の便利な機能について、説明します。

5.3.1 わかりやすいボタン表示に変更したい

実行制御(プログラムの実行/停止/ステップ実行/リセットなど)や各種ウインドウのオープンは、メイン・ウインドウ上の下記ボタンで行なうことができますが、慣れていない方には分かりにくいかもしれません。



こういった場合、メニュー・バーの[オプション]→[デバッガ・オプション]を選択し、Tool Bar Picture 設定エリアで[Pictures and Text]にチェックをつけてください。

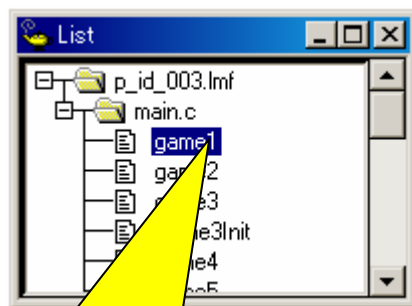


この設定を行うと、以下のようにボタンが文字と合わせて表示されるようになり、分かりやすくなります。

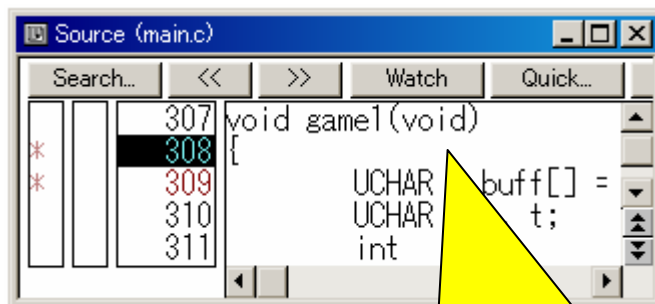
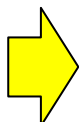


5.3.2 ソース一覧や関数一覧を表示したい

ソース・ファイルや関数の一覧が見たい場合はメニュー・バーの[ブラウザ]→[その他]→[List]を選択して List ウィンドウを開きます。このウィンドウは、ソース・ウィンドウと連動しているので、一覧を見るだけでなく、ソース・ウィンドウで表示するファイルや関数を変更したい際に便利な機能です。



例えば関数「game1」をクリックすると



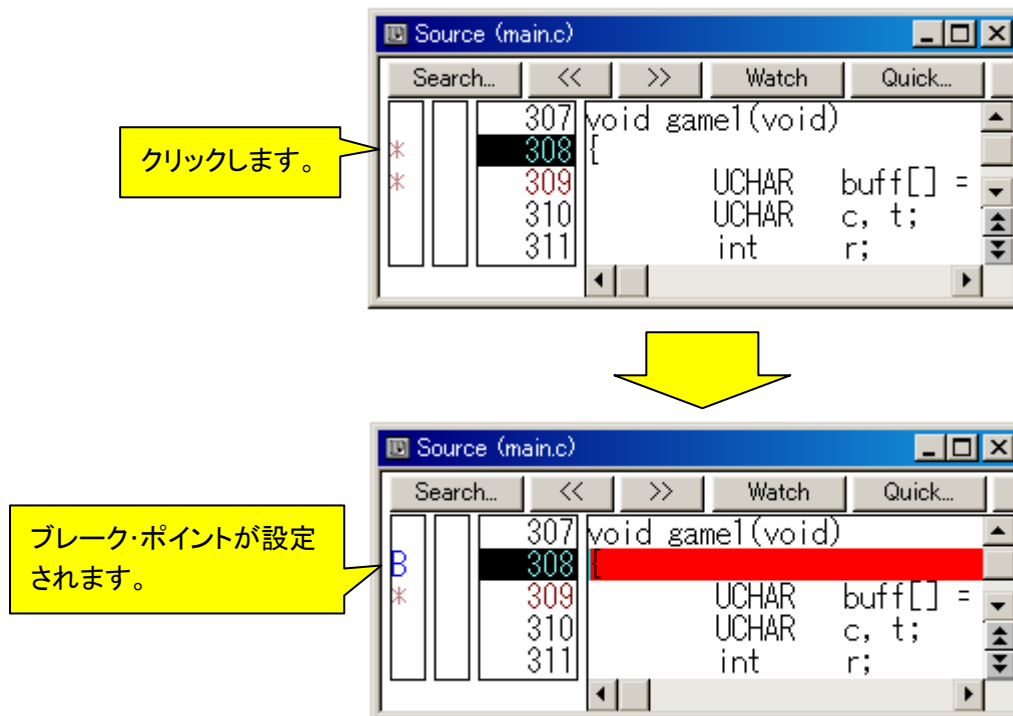
ソース・ウィンドウで「game1」が表示されます。

5.3.3 ブレーク・ポイントの設定/解除を行ないたい

ブレーク・ポイントの設定/解除はソース・ウインドウの「*」マークをクリックすることで行なうことができます。

一度クリックすると「B」マークに変化します(ブレーク・ポイントが設定された状態になります)。

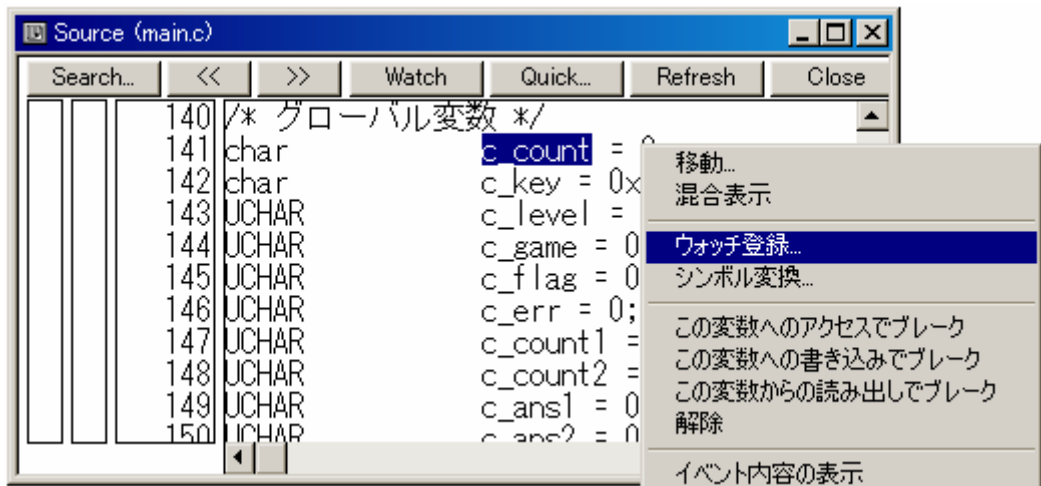
「B」マークをクリックすると、「*」マークに戻ります(ブレーク・ポイントが解除された状態になります)。



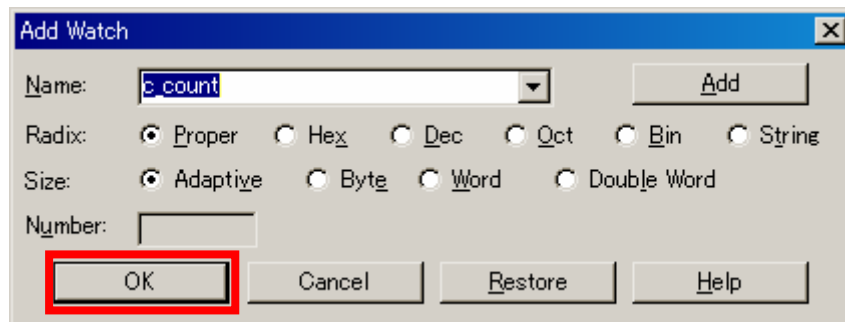
5.3.4 グローバル変数の値を表示したい

グローバル変数の表示は、ウォッチ・ウィンドウで行います。ウォッチ・ウィンドウにグローバル変数を登録する方法はいくつかありますが、ここではソース・ウィンドウから登録を行なう方法を紹介します。

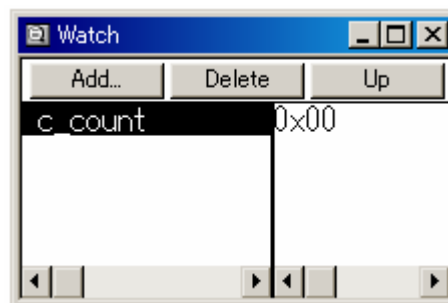
①ソース・ウィンドウ上で変数名をドラッグし、右クリック・メニューの[ウォッチ登録]を選択します。



②ウォッチ登録ダイアログが開くので、[OK]ボタンを押します。



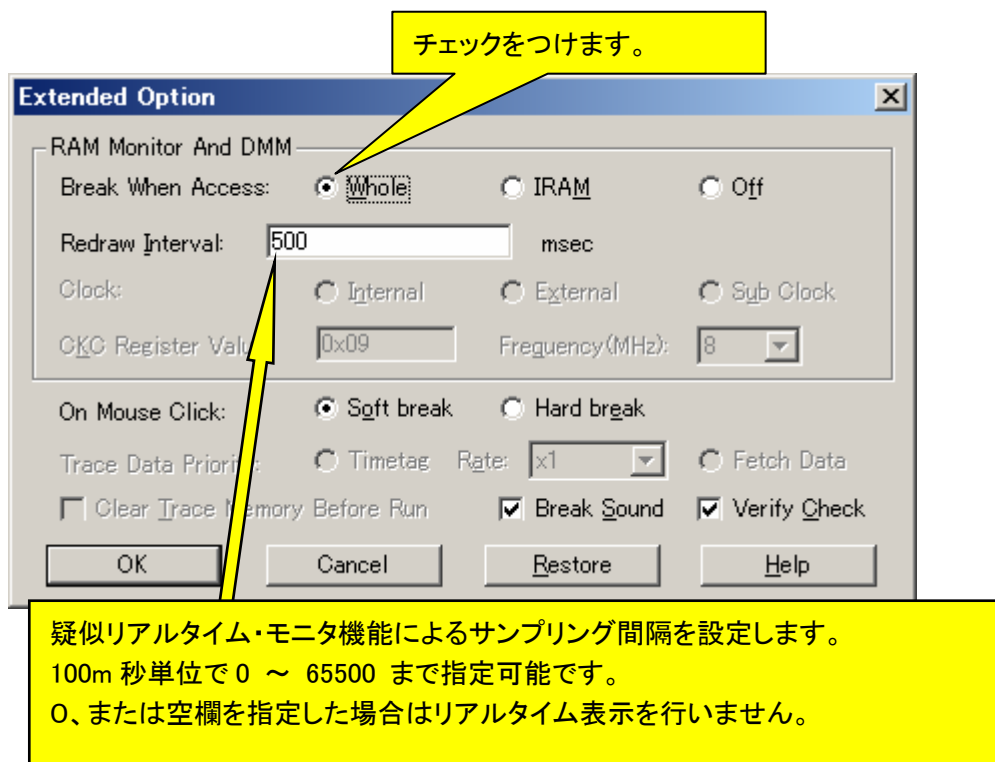
③ウォッチ登録が完了します。



5.3.5 グローバル変数の値をプログラム実行中でも表示したい

疑似リアルタイム・モニタ機能により内部 RAM、汎用レジスタ、SFR の値をプログラム実行中も参照が可能です。

メニュー・バーの[オプション]→[拡張オプション]を選び、[RAM Monitor And DMM]エリアに関して以下の設定を行います。



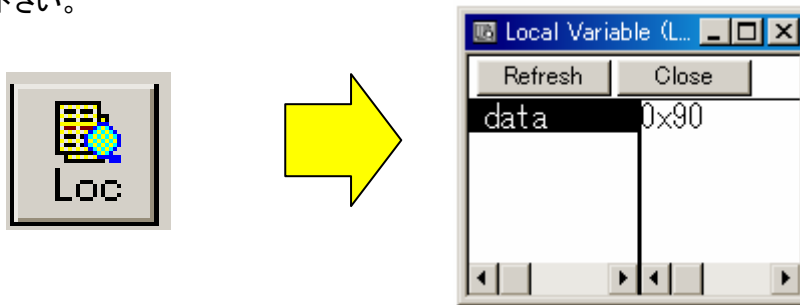
以上で設定は完了です。便利な機能ですが、注意事項があります。

- ・ 変数読み出しの瞬間、内部的には 1 瞬ブレークしています。
- ・ メモリ・ウィンドウを開いた状態で疑似リアルタイム・モニタ機能を使用すると操作性が著しく悪くなります。(表示しているメモリの内容もモニタしているため)
- ・ 疑似リアルタイム・モニタ機能使用時はメモリ・ウィンドウを閉じることを推奨します。

5.3.6 ローカル変数の値を表示したい

ローカル変数の表示は、ローカル変数ウインドウで行ないます。下記ボタンでローカル変数ウインドウを開くことができます。

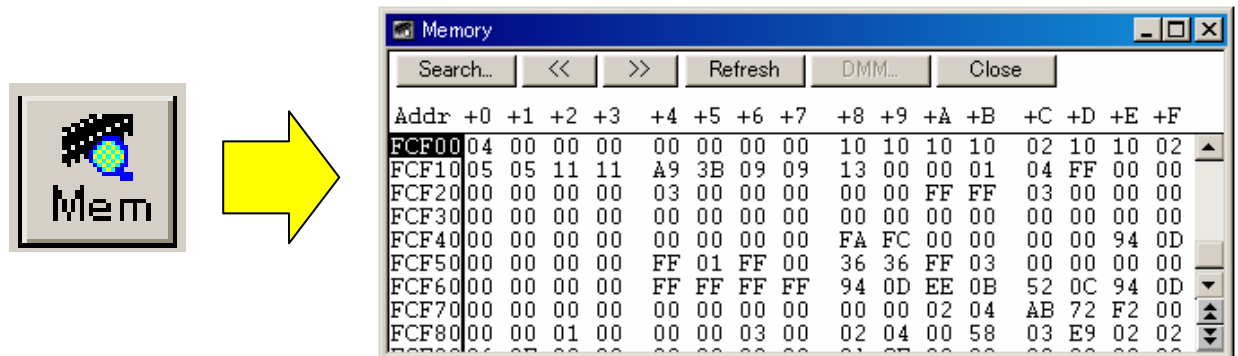
ローカル変数は、グローバル変数と異なり、プログラム実行中に値を参照することが出来ませんので、ご注意ください。



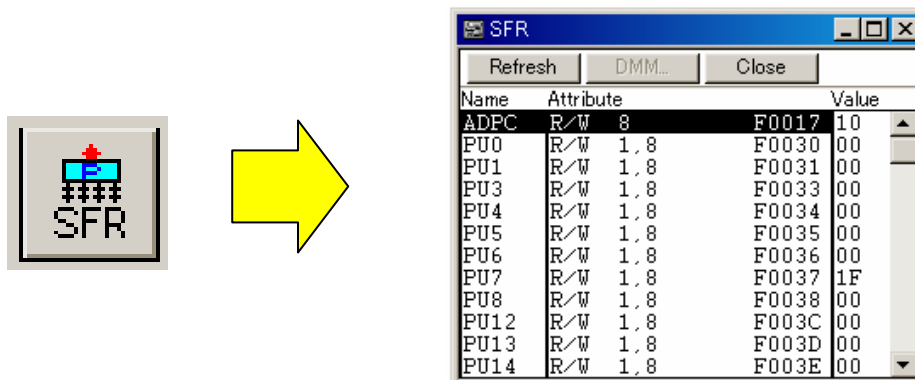
5.3.7 メモリの内容や SFR の値を表示したい

ローカル変数の表示と同様に以下のボタンで表示可能です。

・メモリの内容を表示する場合、以下のボタンを押します。



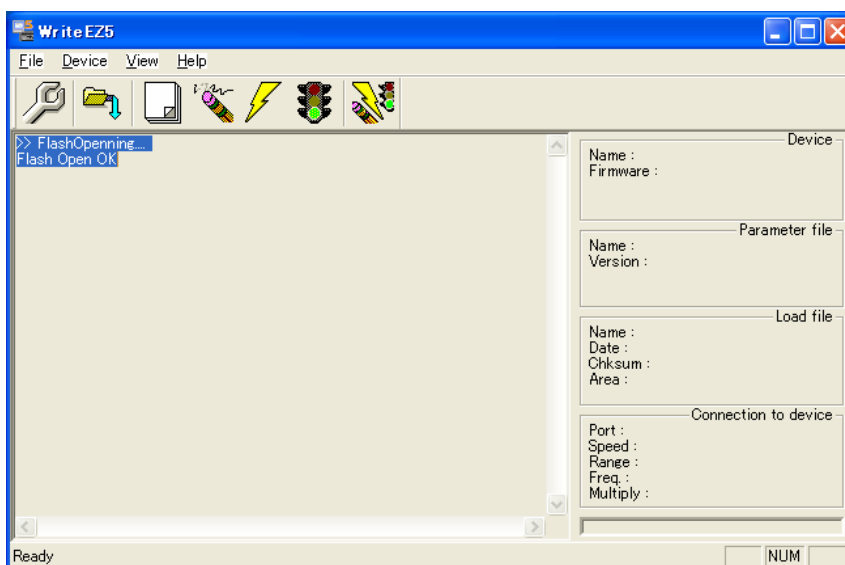
・SFR の値を表示する場合、以下のボタンを押します。



5.4 マイコン内蔵のフラッシュ・メモリを消去したい

WriteEZ5 はセキュリティ ID を忘れた場合や、意図しない値を書いてしまった場合、フラッシュ・メモリを消去する場合に使用します。フラッシュ・メモリを消去することによって、セキュリティ ID を「FFFFFFFFFFFFFFFFFFFFFFFF」に設定することができます。

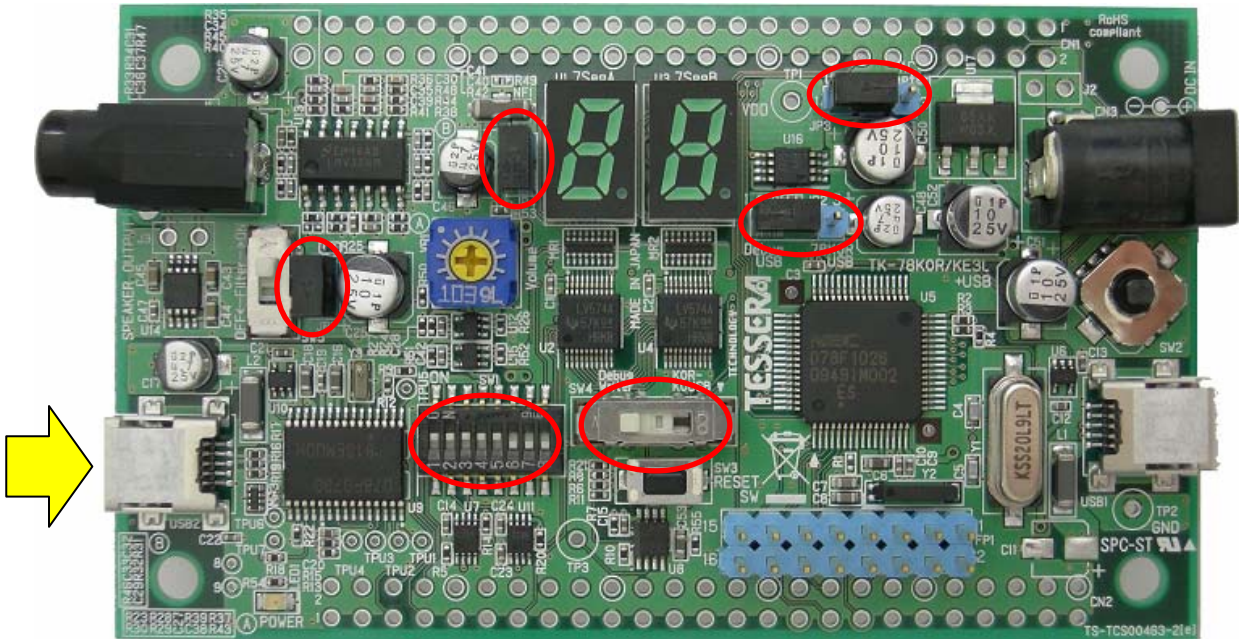
- ①NEC Electronics Tools から WriteEZ5 を起動します。



②以下の様にジャンパピンとスイッチの設定します。

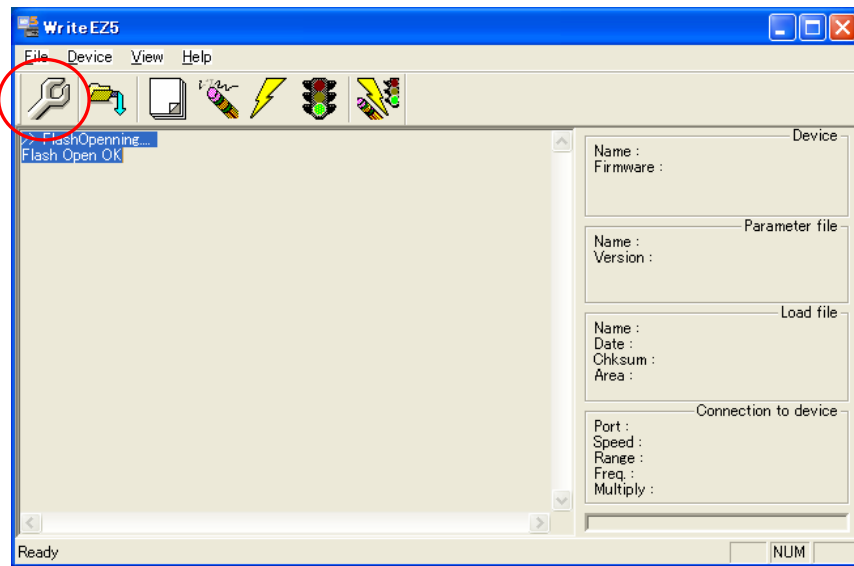
JP1	ショート
JP2	1-2 ショート
JP3	1-2 ショート
JP4	ショート
SW4	Debug Writer

SW1							
1	2	3	4	5	6	7	8
ON	ON	ON	ON	ON	OFF	OFF	OFF

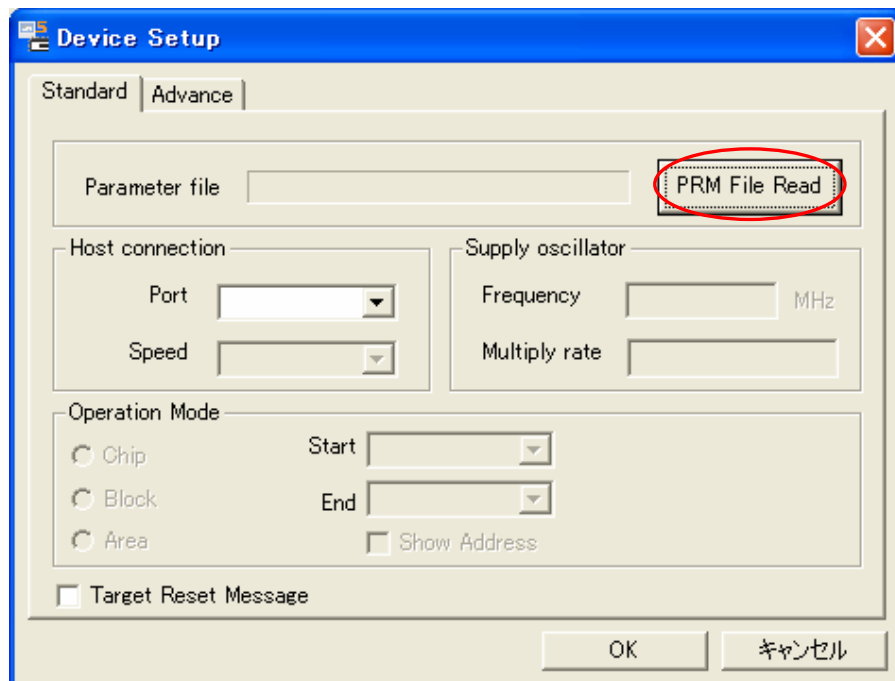


③設定が完了したら TK-78K0R/KE3L+USB の「USB2」コネクタと、パソコンの USB コネクタを、USB ケーブルで接続します。

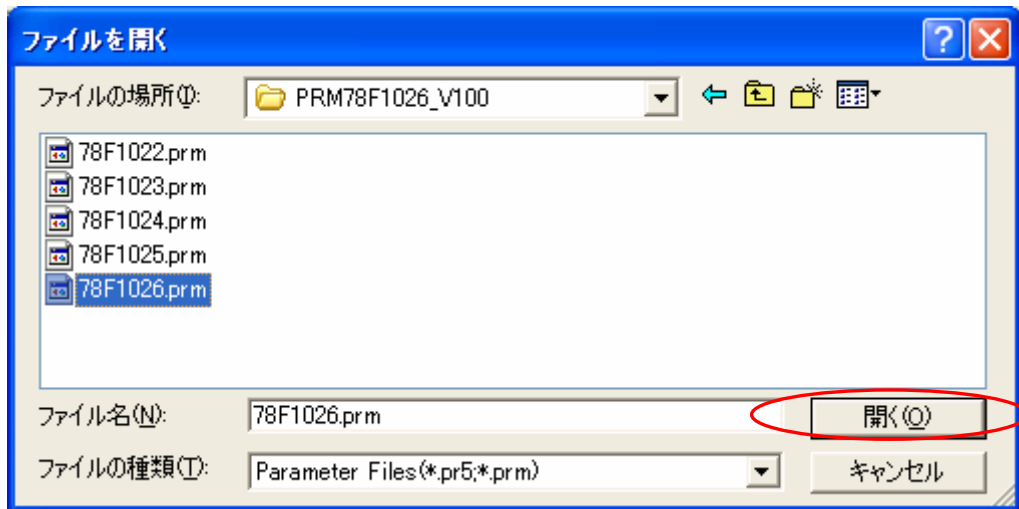
④Setup ボタンを押します。



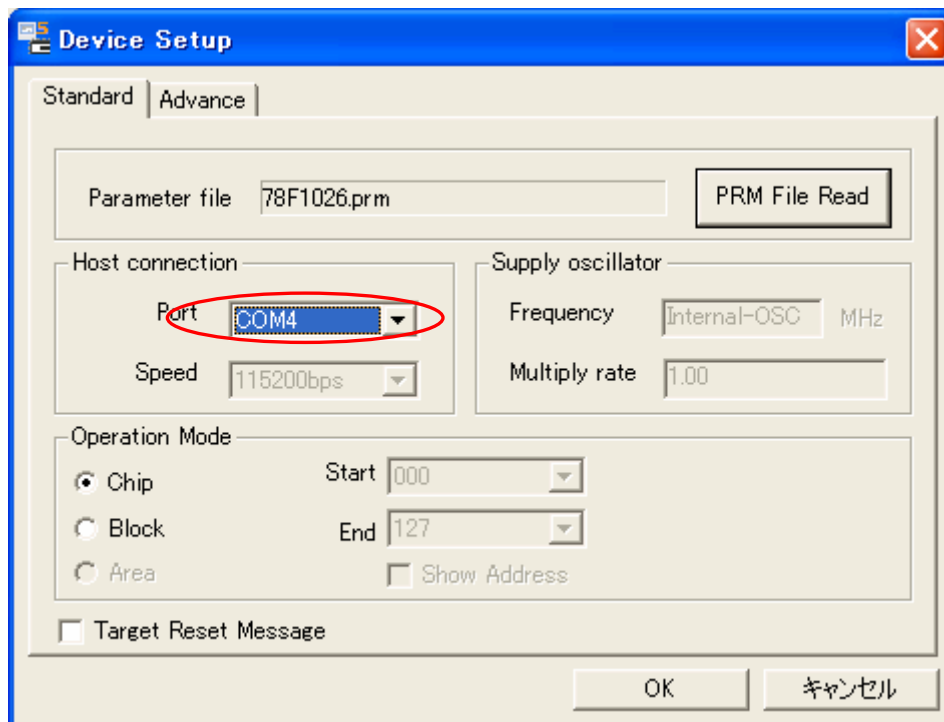
⑤「PRM File Read」ボタンをクリックします。



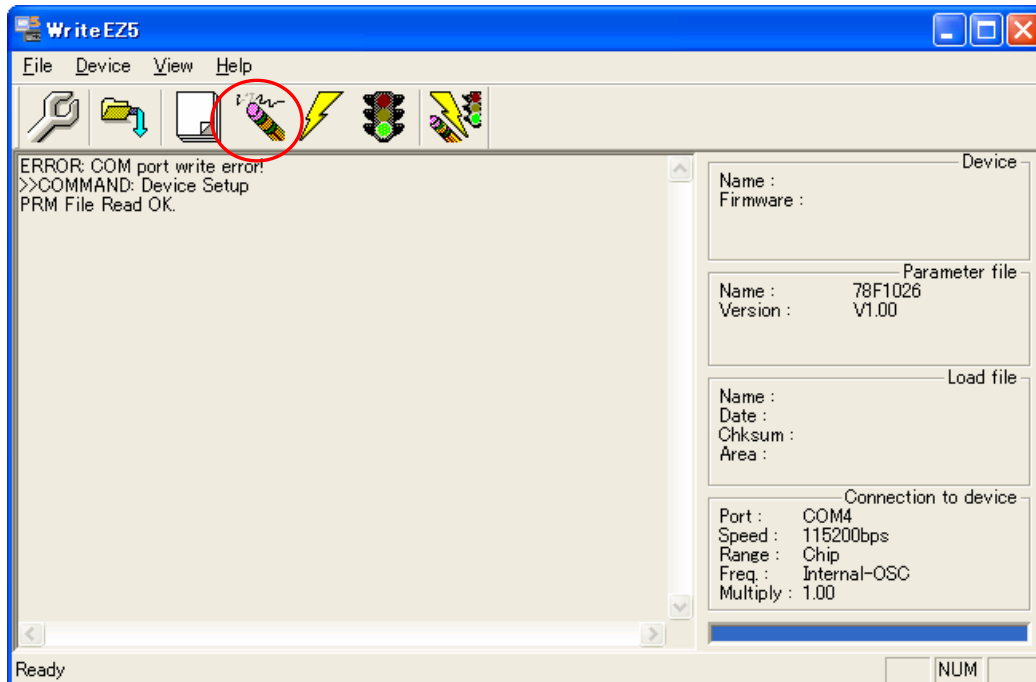
⑥付属 CD から“¥PRM78F1026_V100¥78F1026.prm”を選択します。



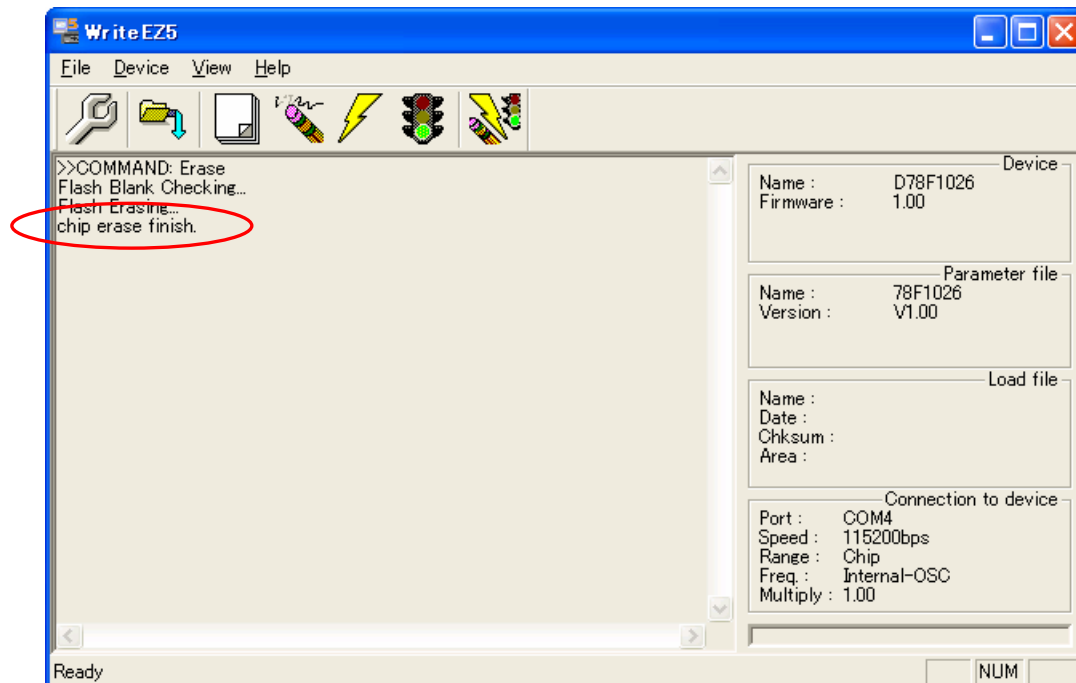
⑦「Port」に TK-78K0R/KE3L+USB が接続されている COM ポートを選択します。



⑧「Erase」ボタンをクリックします。

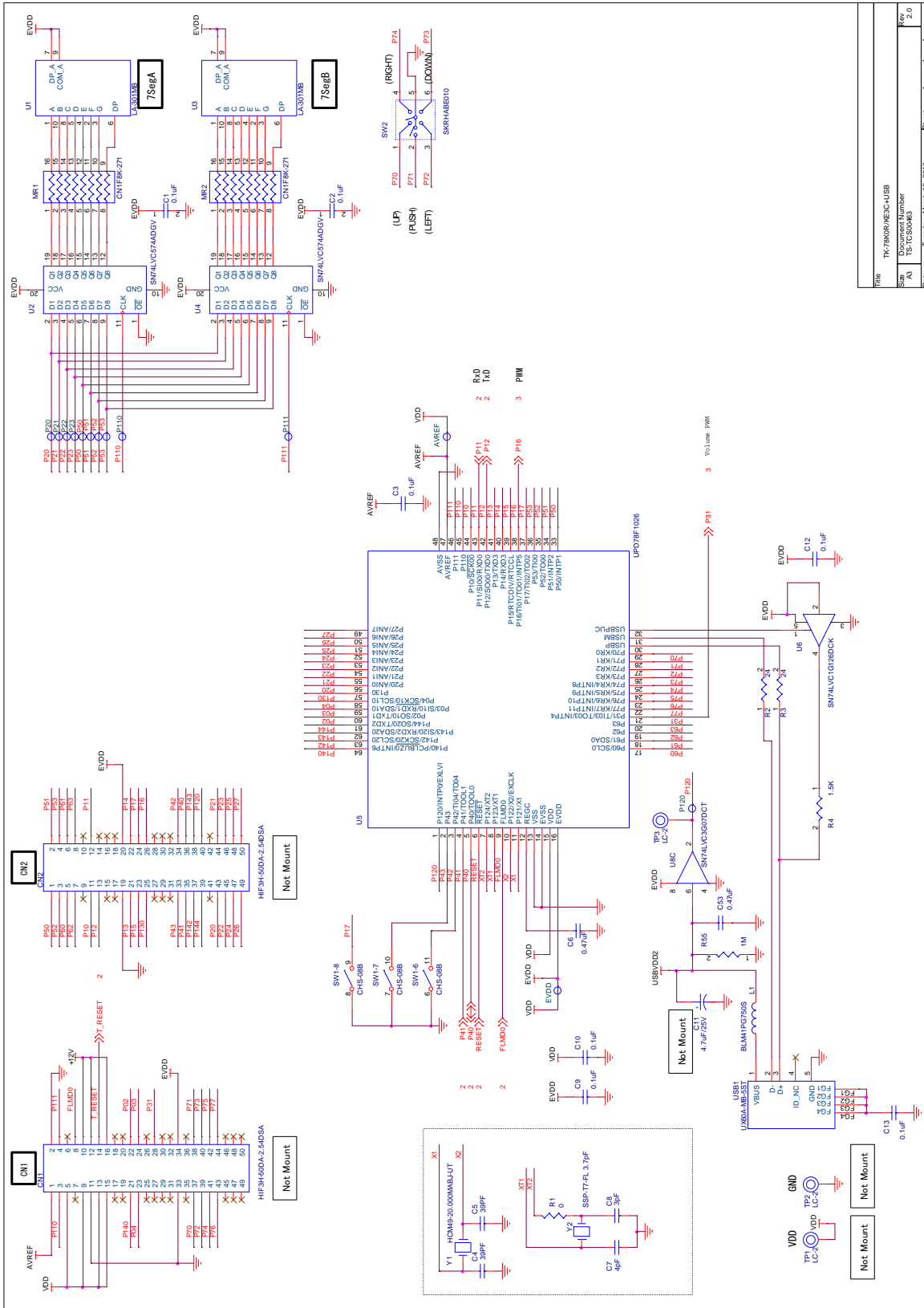


⑨下記のように「chip erase finish.」が表示されれば完了です。

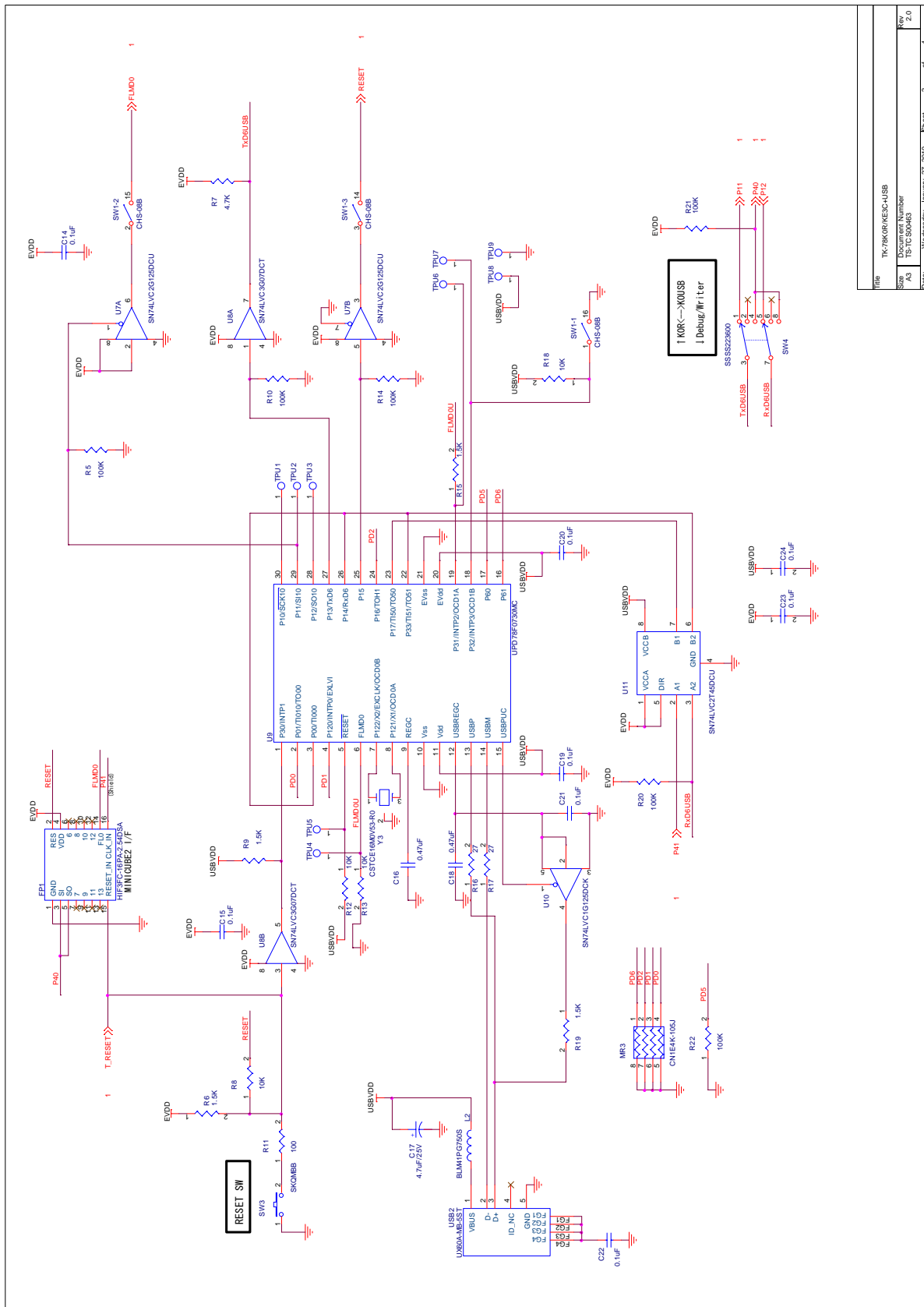


5.5 回路図

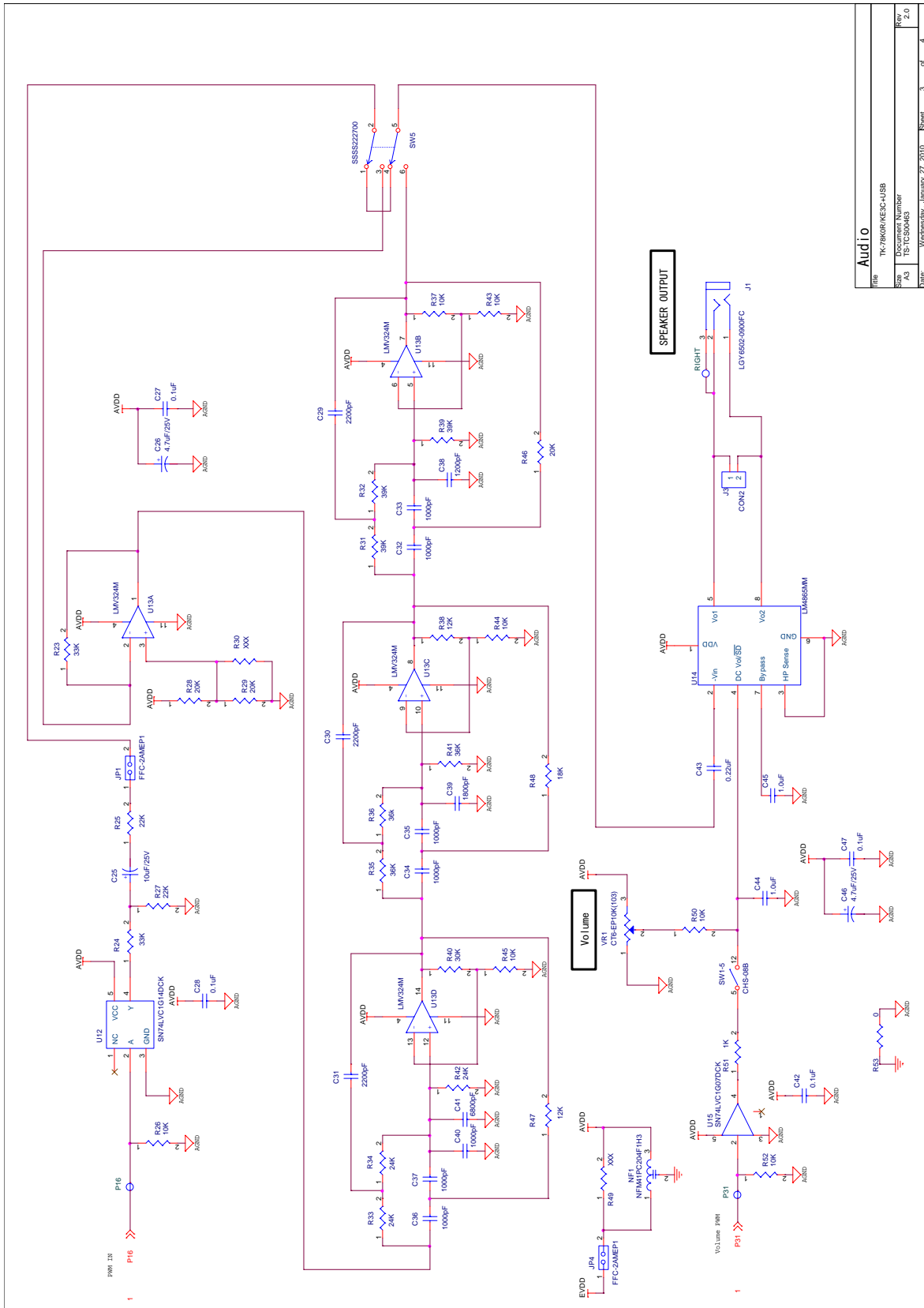
次ページより本キットの回路図を示します。



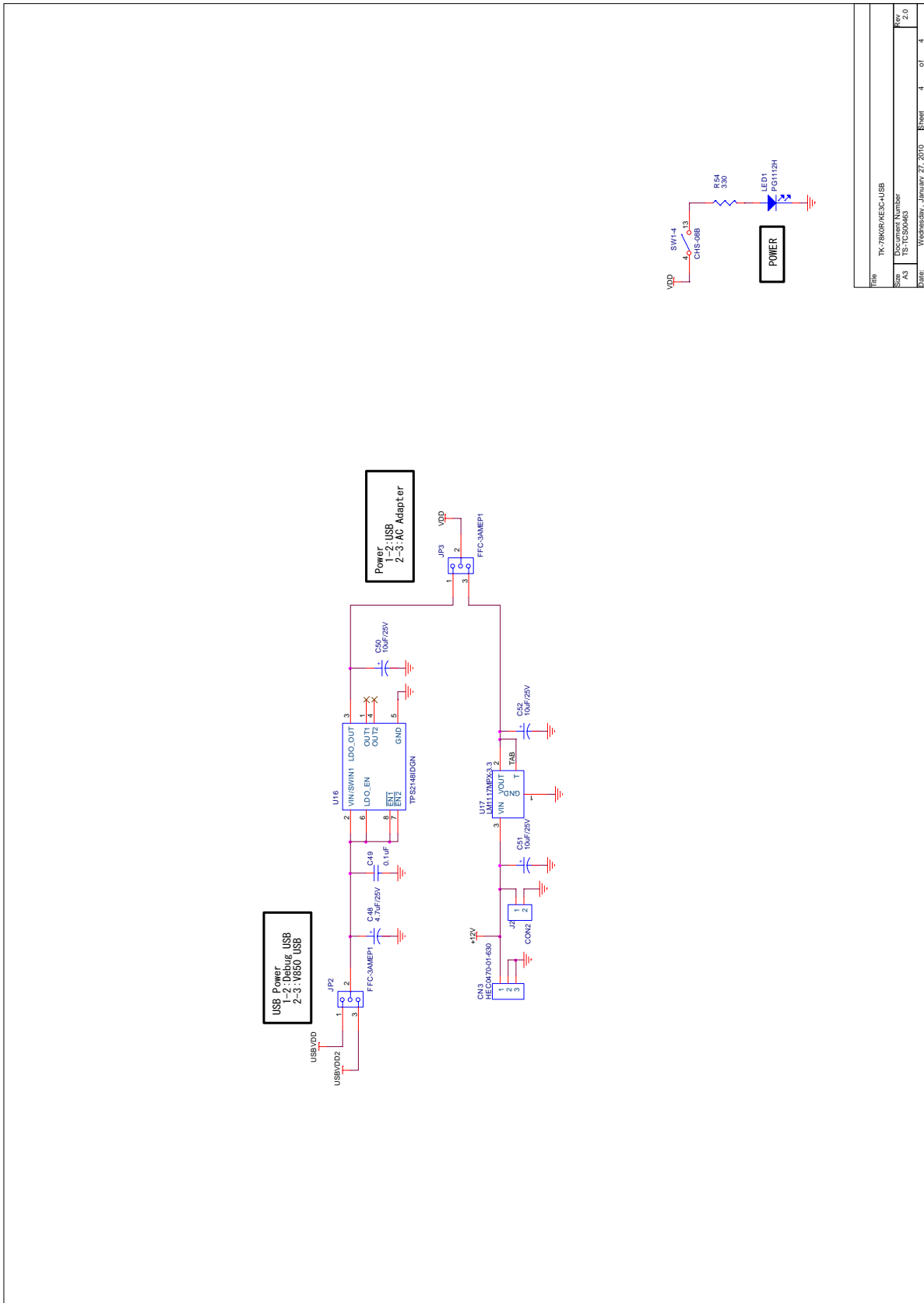
File	TK-78K0R/KE3L+USB
Sub	Document Number
Part	TS-TC500463
Print	Jan 2010, March 15, 2010
Sheet	1 of 4
Rev	2.0



Title		TK-78K0R/KE3L+USB
Size	Document Number	TS-TC-S0043
A3	Date	Wednesday, January 27, 2010
Sheet		2 of 4
Rev		2.0



Audio	
File	TK-78K0R/KE3L+USB
Size	Document Number
Rev	TS-TC500483
Date	Wednesday, January 27, 2010
Sheet	3 of 4



Title	TK-78K0R/KE3L+USB
Size	Document Number
AS	PS-1C3004S
Date	Wednesday, January 27, 2010
Sheet	4 of 4
Rev.	2.0