

実践！自動車組込み技術者講座

FPGAとマイコンの連携システム (ソフトウェア基礎編)

~The first step is the only difficulty.



Content

| | | |
|-----|------------------------------------|-----|
| 0 | はじめに | 3 |
| 1 | マイコンボードの仕様 | 5 |
| 1.1 | 概要 | 5 |
| 1.2 | MCU 回路 | 7 |
| 1.3 | 基本 I/O 回路 | 17 |
| 2 | HEW による開発環境 | 29 |
| 2.1 | 概要 | 29 |
| 2.2 | E8a エミュレータパッケージ付属 CD-ROM のインストール手順 | 31 |
| 2.3 | ワークスペースとプロジェクト | 35 |
| 2.4 | HEW の起動 | 36 |
| 2.5 | 新規プロジェクトの作成 | 37 |
| 2.6 | コーディング、ビルド（ロードモジュールの生成） | 45 |
| 2.7 | E8a を使用したデバッグの準備 | 51 |
| 2.8 | プログラムの実行 | 59 |
| 3 | シリアルインターフェース | 64 |
| 3.1 | 概要 | 64 |
| 3.2 | 同期式通信と非同期式通信 | 66 |
| 4 | 距離データの応用 | 70 |
| 4.1 | 概要 | 70 |
| 4.2 | ソフトウェア詳細仕様 | 71 |
| 4.3 | マイコンの端子配置 | 73 |
| 4.4 | システム構成上の問題点 | 74 |
| 4.5 | 距離計システムを動作させる | 79 |
| 5 | おわりに | 101 |
| 6 | Appendix | 102 |
| 表 1 | 部品表 | 102 |
| | 参考文献・引用文献 | 103 |

0

はじめに



本編は、FPGAボードに搭載した超音波センサーで検出した反射物までの距離データをマイコンボード側で受信して、実際に反射物までの距離が人間の五感で分かるような近接センサーシステムを設計・製作します。その過程でソフトウェア技術者として必要な基礎技術の構成要素を知り、組込み技術者としてのロードマップを修得することを目標としています。

本編は、組込みソフトウェア技術者の入門者向けテキストですが、必ず通過する言語や設計、デバッグ方法論には触れずにハードルは低く設定してあります。

特徴

1. プログラム言語、設計論、コード書法、情報理論、工程管理、品質保証・管理、プロジェクト管理、業務知識（自動車の各種ECU、カメラ、モバイル機器の構造・構成要素など）には触れません。
2. マイコンボード基板の電子部品・電子回路が正常動作するか、測定器を使って確認します。
3. マイコンの電気的特性や電子部品のデータシートを読みます。ポートに流れる許容電流を把握する必要があることを理解します。
4. マイコンの開発環境(本編ではルネサスエレクトロニクス社のHew(High performance Embedded Workshop)を構築し・機能・操作方法を学びます。
5. ポート制御を行う簡単なプログラムを作成することで、開発環境を活用したソースコードの生成・コンパイル・アセンブル・リンク・実行といった一連の手順を理解します。
6. 入力／出力の意味を理解します。
7. 代表的なシリアルインタフェース通信規約を理解し、活用します。
8. ソフトウェア仕様書を理解します。
9. FPGAとマイコンボード間の電圧レベルが揃っていないときの電圧レベル変換対策を検討します。
10. 電圧レベル変換回路を通して、プルアップ抵抗、High レベル、Low レベル、ハイ

インピーダンスの意味を理解します。

11. FPG Aボードとマイコンボードを接続して実際に距離データをブザーやLED
に表示して、近接センサーとしての動きを確認します。

組込みソフトウェア開発における自分の立ち位置（担当レベル）で修得すべき技術が大きく異なります。

ひとつのハードルを越えたとき、組込みソフトウェア技術の新たな広がりを認識していた
できれば幸いです。

2014 年 2 月

1

マイコンボードの仕様



1. 1 概要

マイコンボード (Seamark-1 : Basic board) は、MCU回路と基本 I/O回路で構成されます。MCU回路と基本 I/O回路はGND以外は電氣的に接続されていません。マイコン H8/3694F のポートから入出力制御を行うには入出力ポートの端子割付を行い、MCU回路の I/O拡張端子 (Term8) と基本 I/O回路の各ターミナルをピンクリップ等で結線する必要があります。

特徴

- ・ 搭載MCU基板 : H8 / 3 6 9 4 F (ルネサス製)
- ・ 電子デバイスを実装
(LED、圧電ブザー (他励式)、半固定抵抗、7セグメントLED、タクトスイッチ (プルアップ抵抗の有無の設定可)、DIPスイッチ、Tr用ソケット、14ピンICソケット)
- ・ マイコン I/O端子と電子デバイスは未接続のため端子配置は自由に設計可能
- ・ ボードの電源はACアダプタで供給するか、E8aエミュレータのUSBケーブルから供給するかの2電源方式
- ・ マイコンの I/O端子を外部機器と接続するために拡張端子を実装
- ・ マイコンを動作させずに、電子デバイスの動作実験が可能

写真 1 にマイコンボード (Seamark-I : Basic board) の写真、図 1 にマイコンボードのブロック図を示します。

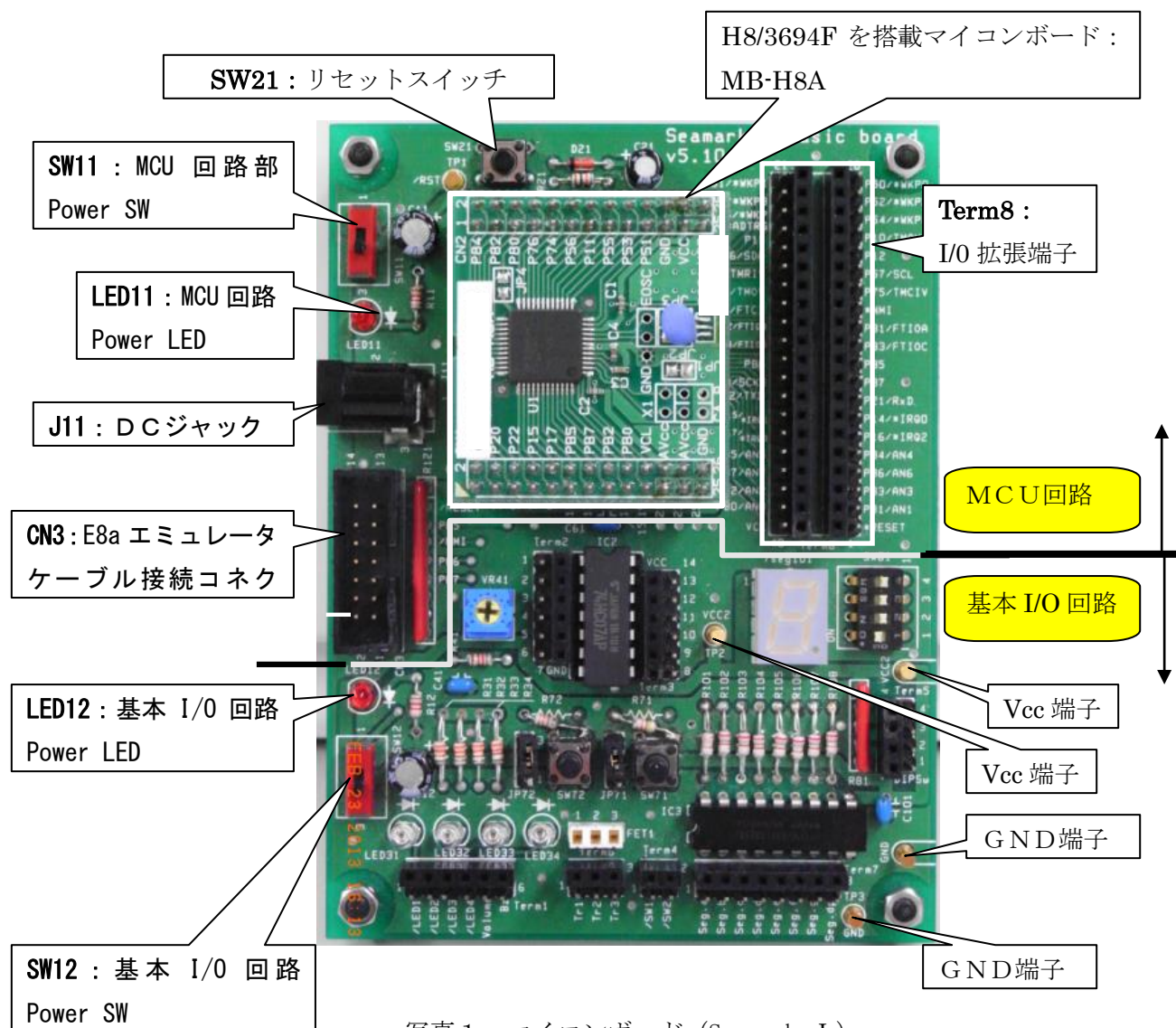


写真1 マイコンボード (Seamark- I)

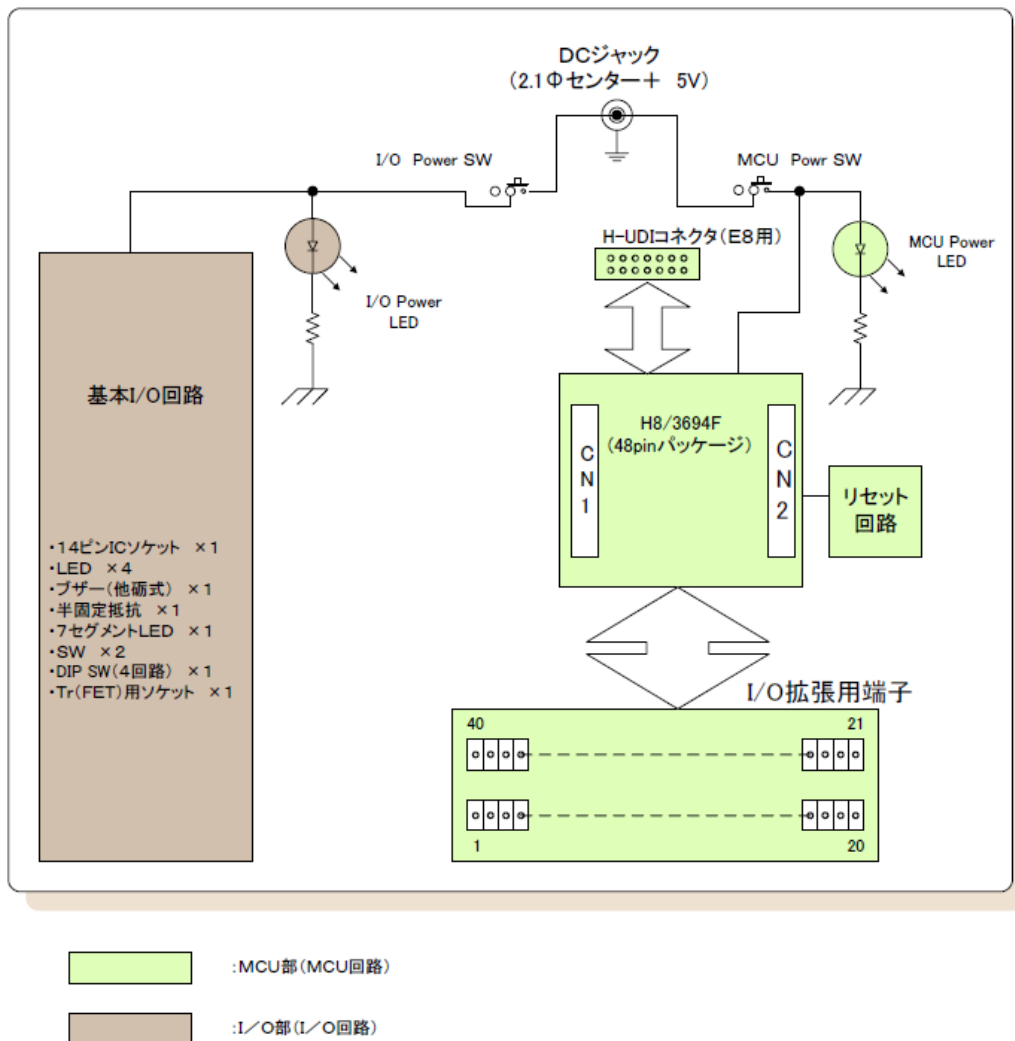


図 1 マイコンボード (Seamark-I) ブロック図

1. 2 MCU回路

MCU回路は下記より構成されています。

- ・H8/3694F 搭載マイコンボード MB-H8A (Sunhayato 製)
- ・I/O 拡張端子
- ・リセット回路
- ・電源回路
- ・H-UDI 接続回路

■ H8/3694F 搭載マイコンボード MB-H8A (Sunhayato 製)

H8/3694F 搭載マイコンボード MB-H8A の回路図を図 2 に、端子割付を表 1 に示します。

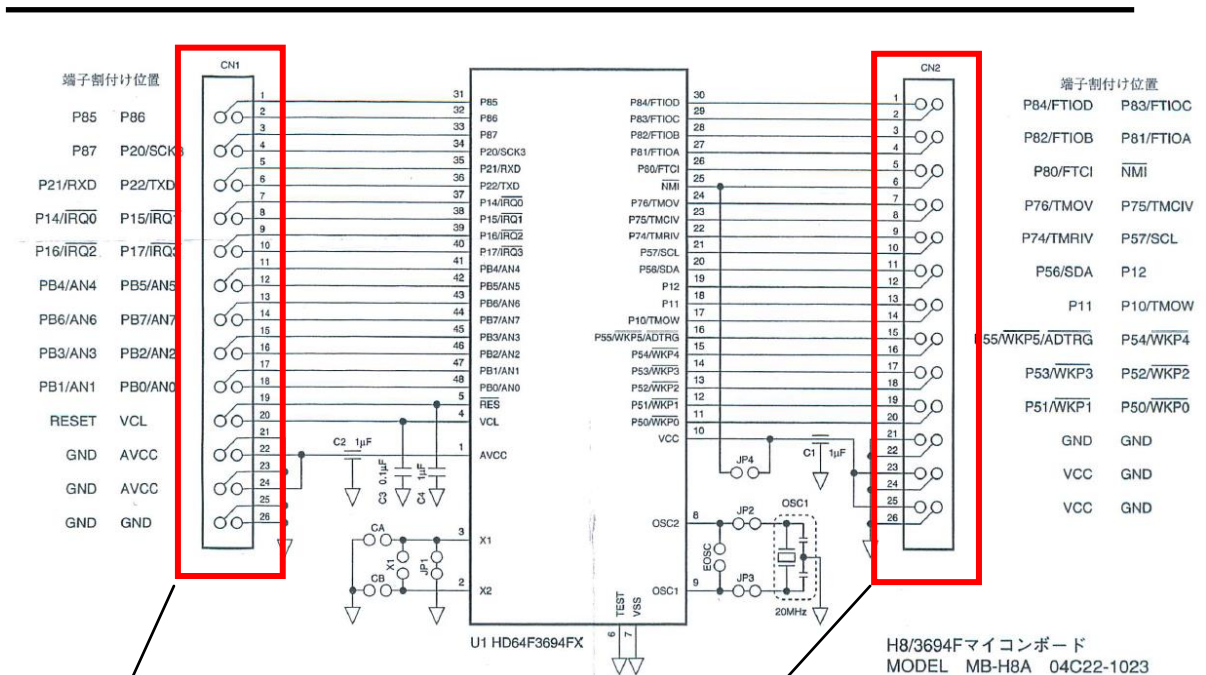


図 2 MB-H8A 回路図

| CN1 番号 | 3694F 端子番号 | 機 能 | 備 考 | CN2 | 3694F 端子番号 | 機 能 | 備 考 |
|-------------|---------------|---------------|------------------------------|-------------|---------------|----------------|----------------|
| 1 | 31 | P85 | ブートモード時H | 1 | 30 | P84/FTIOD | ブートモード時L |
| 2 | 32 | P86 | | 2 | 29 | P83/FTIOC | |
| 3 | 33 | P87 | | 3 | 28 | P82/FTIOB | |
| 4 | 34 | P20/SCK3 | | 4 | 27 | P81/FTIOA | |
| 5 | 35 | P21/RXD | | 5 | 26 | P80/FTCI | |
| 6 | 36 | P22/TXD | | 6 | 25 | NMI | |
| 7 | 37 | P14/IRQ0 | | 7 | 24 | P76/TMOV | |
| 8 | 38 | P15/IRQ1 | | 8 | 23 | P75/TMCIV | |
| 9 | 39 | P16/IRQ2 | | 9 | 22 | P74/TMRIV | |
| 10 | 40 | P17/IRQ3/TRGV | | 10 | 21 | P57/SCL | |
| 11 | 41 | PB4/AN4 | | 11 | 20 | P56/SDA | |
| 12 | 42 | PB5/AN5 | | 12 | 19 | P12 | |
| 13 | 43 | PB6/AN6 | | 13 | 18 | P11 | |
| 14 | 44 | PB7/AN7 | | 14 | 17 | P10/TMOW | |
| 15 | 45 | PB3/AN3 | | 15 | 16 | P55/WKP5/ADTRG | |
| 16 | 46 | PB2/AN2 | | 16 | 15 | P54/WKP4 | |
| 17 | 47 | PB1/AN1 | | 17 | 14 | P53/WKP3 | |
| 18 | 48 | PB0/ANO | | 18 | 13 | P52/WKP2 | |
| 19 | 5 | RES | | 19 | 12 | P51/WKP1 | |
| 20 | 4 | VCL | | 20 | 11 | P50/WKP0 | |
| 21,23,25,26 | 7 | VSS | Lでリセット 内部降圧電源端子 グラント端子 | 21,22,24,26 | 7 | VSS | グラント端子 電源端子 |
| 22,24 | 1 | AVCC | A/D変換用電源 | 23,25 | 10 | VCC | |

表 1 MB-H8A 端子配列

H8/3694F 搭載マイコンボード MB-H8A には、(株)ルネサスエレクトロニクス社の 16 ビット CISC マイコン H8/3694F が搭載されています。図 3 に H8/3694F の内部ブロック図を示します。

① H8/3694F

H8/3694Fの特長

- ・ 16bitCISCマイコン
- ・ 32kバイトのフラッシュメモリ
- ・ 2kバイトのRAM
- ・ 10ビット8チャンネルADコンバータ
- ・ ウォッチドッグタイマ
- ・ 多機能タイマの他にシリアル
- ・ I2Cバスインターフェイスを内蔵
- ・ エミュレータインターフェイスを内蔵
- ・ 高速20MHz動作
- ・ サブクロック

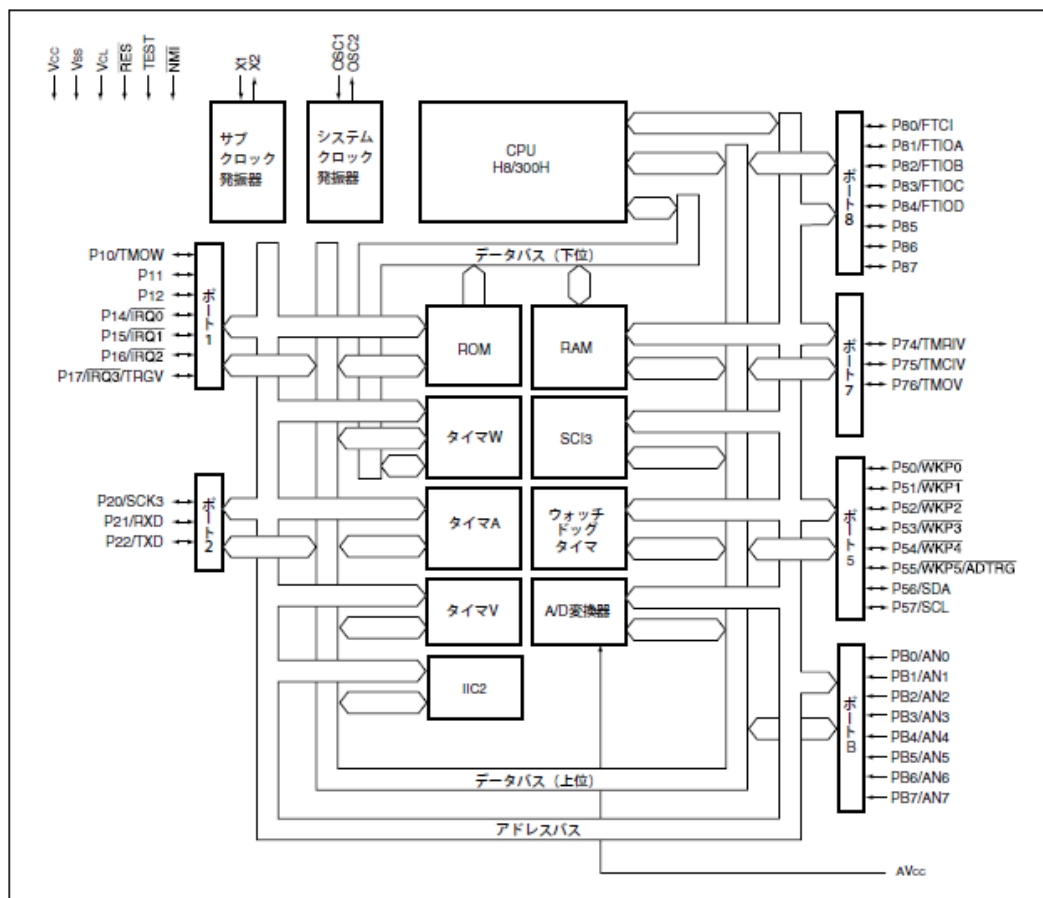


図3 F-ZTATTM 版、マスクROM 版 H8/3694 グループ内部ブロック図

マイコン内蔵フラッシュメモリは、最大書き込み回数を超えるとプログラムのダウンロード等ができなくなる可能性があります。消耗品として扱います。図 4 にピン配置、図 7.5 に H8/3694F のメモリマップを示します。

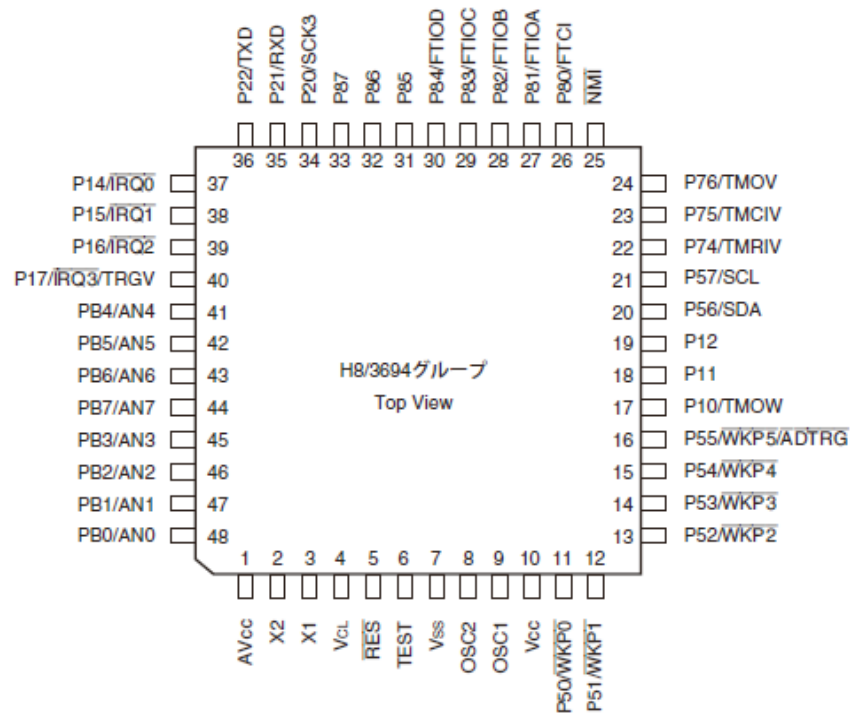


図 4 F-ZTATTM 版、マスク ROM 版 H8/3694 グループピン配置図
(FP-48F、FP-48B、TNP-48)

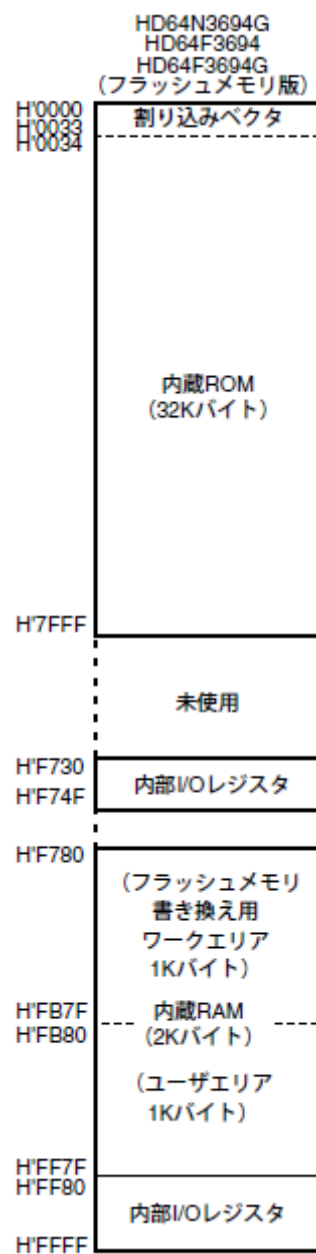


図5 メモリマップ

■ I/O拡張端子&回路

MCU H8/3694F は 48 ピンパッケージ（64 ピン版もあります）ですが、48 ピンのうち 40 ピンを基本 I/O 回路部の各種電子デバイスに接続できるように I/O 拡張用端子まで引き出しています。表 2 にマイコン H8/3694F のピン番号とマイコンボード I/O 拡張端子（Term8）ピン番号の関係を示します。

マイコンボードの I / O 拡張端子

(Term8) ピン番号

H8/3694F のピン番号

| I/O基板 | | H8/3694F端子 | | MCUボード | | 備考 |
|-------|------------------|------------|------|--------|------|------|
| pin番号 | 端子名称 | 端子番号 | CN番号 | pin番号 | | |
| 1 | *RESET | 5 | 1 | 19 | E8接続 | |
| 2 | PB1/AN1 | 47 | 1 | 17 | | |
| 3 | PB3/AN3 | 45 | 1 | 15 | | |
| 4 | PB6/AN6 | 43 | 1 | 13 | | |
| 5 | PB4/AN4 | 41 | 1 | 11 | | |
| 6 | P16/*IRQ2 | 39 | 1 | 9 | | |
| 7 | P14/*IRQ0 | 37 | 1 | 7 | | |
| 8 | P21/RxD | 35 | 1 | 5 | | |
| 9 | P87 | 33 | 1 | 3 | | E8接続 |
| 10 | P85 | 31 | 1 | 1 | | E8接続 |
| 11 | P83/FTIOC | 29 | 2 | 2 | E8接続 | |
| 12 | P81/FTIOA | 27 | 2 | 4 | | |
| 13 | *NMI | 25 | 2 | 6 | | |
| 14 | P75/TMCIV | 23 | 2 | 8 | | |
| 15 | P57/SCL | 21 | 2 | 10 | | |
| 16 | P12 | 19 | 2 | 12 | | |
| 17 | P10/TMOW | 17 | 2 | 14 | | |
| 18 | P54/*WKP4 | 15 | 2 | 16 | | |
| 19 | P52/*WKP2 | 13 | 2 | 18 | | |
| 20 | P50/*WKP0 | 11 | 2 | 20 | | |
| 21 | P51/*WKP1 | 12 | 2 | 19 | E8接続 | |
| 22 | P53/*WKP3 | 14 | 2 | 17 | | |
| 23 | P55/*WKP5/*ADTRG | 16 | 2 | 15 | | |
| 24 | P11 | 18 | 2 | 13 | | |
| 25 | P56/SDA | 20 | 2 | 11 | | |
| 26 | P74/TMRIV | 22 | 2 | 9 | | |
| 27 | P76/TMOV | 24 | 2 | 7 | | |
| 28 | P80/FTCI | 26 | 2 | 5 | | |
| 29 | P82/FTIOB | 28 | 2 | 3 | | |
| 30 | P84/FTIOD | 30 | 2 | 1 | | |
| 31 | P86 | 32 | 1 | 2 | | |
| 32 | P20/SCK3 | 34 | 1 | 4 | | |
| 33 | P22/TXD | 36 | 1 | 6 | | |
| 34 | P15/*IRQ1 | 38 | 1 | 8 | | |
| 35 | P17/*IRQ3 | 40 | 1 | 10 | | |
| 36 | PB5/AN5 | 42 | 1 | 12 | | |
| 37 | PB7/AN7 | 44 | 1 | 14 | | |
| 38 | PB2/AN2 | 46 | 1 | 16 | | |
| 39 | PB0/AN0 | 48 | 1 | 18 | | |
| 40 | VCL | 4 | 1 | 20 | | |

表 2 H8/3694F ピン番号とマイコンボード拡張端子（term 8）ピン番号

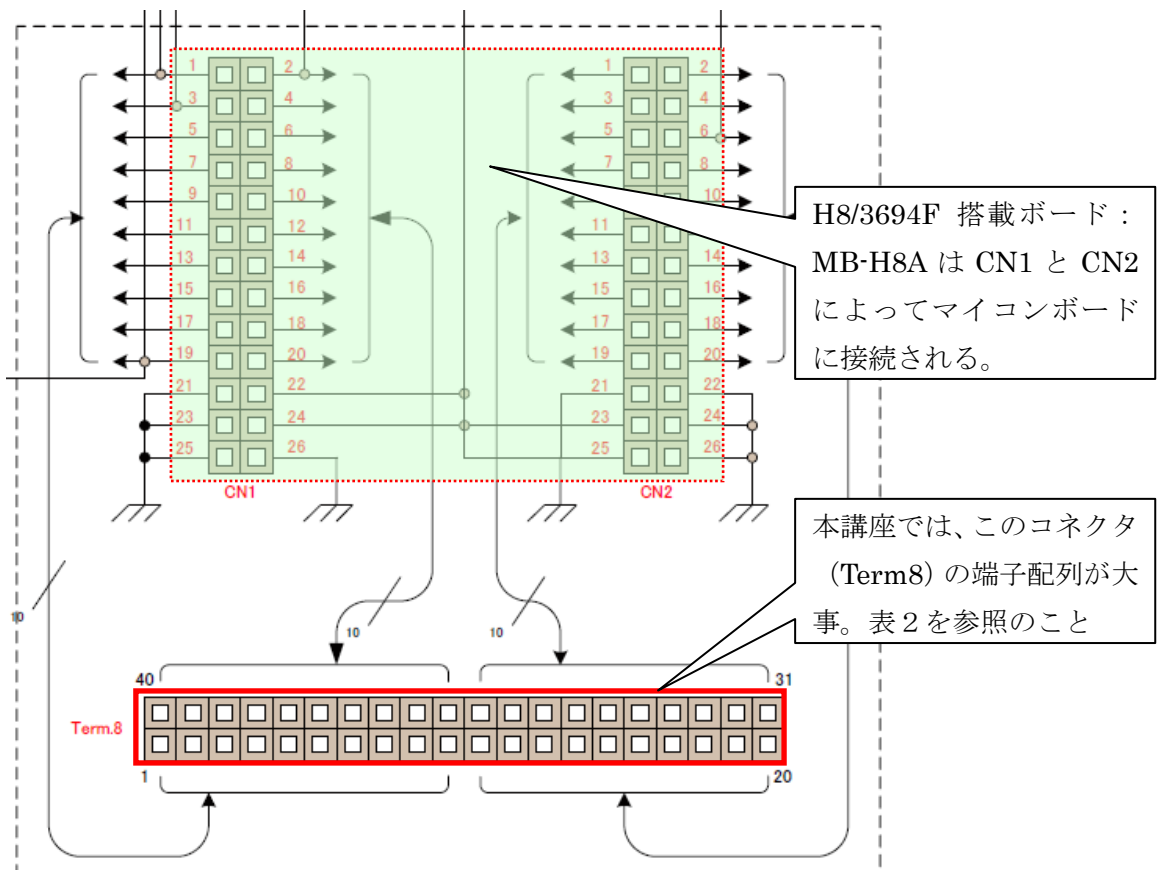


図 6 MB-H8A 接続コネクタと I/O 拡張端子

■ リセット回路

リセットスイッチ（SW21）を押下するとマイコンのリセット端子が Low になりMCUに外部リセットがかかります。

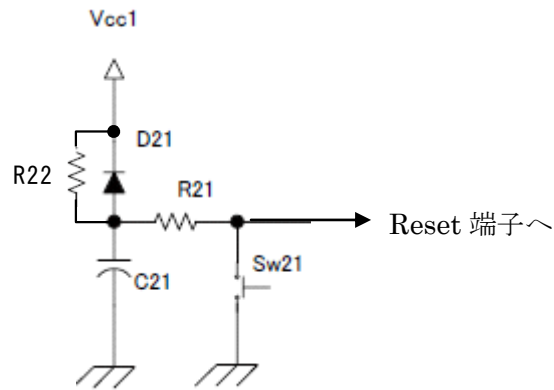


図7 リセット回路

◆参考：リセット動作の実際

図7のリセット回路を例に動作を説明します。

① パワーオンリセット

電源が投入されると、抵抗R22を通じてコンデンサC21を充電し、Reset端子電圧は徐々に上昇します。電源電圧が十分に上昇してからマイコンが安定動作する十分な時間が経過するとリセットは解除されます。

回路中のダイオードD21は、電源がオフになったとき速やかにコンデンサC21の電圧を下げて、次に電源が投入されたときに備える**放電促進ダイオード**です。

② マニュアルリセット

リセットスイッチ（Sw21）を押すとReset電圧端子は0Vになります。パワーオンリセット時の「マイコンが安定動作する十分な時間を待つ」は、リセットスイッチを押すことで実現します。リセットスイッチ（Sw21）を離すとReset端子は電源電圧まで上昇する過程で電源電圧のある一定レベルを超えるとリセットが解除されます。

抵抗R21はC21から大きな電流が流れて回路が損傷しないように電制限の役目をする**電流制限抵抗**です。

■ 電源回路

ACアダプタを使用してAC 100 VをDC 5 Vに変換したものがDCジャック (J11) を通してマイコンボードに供給されます。

Power SW (SW11) をONにすると赤色LED (LED11) が点灯しDC 5 VがMCU回路に供給されたことを表示します。

Power SW (SW12) をONにすると赤色LED (LED12) が点灯しDC 5 Vが基本I/O回路に供給されたことを表示します。

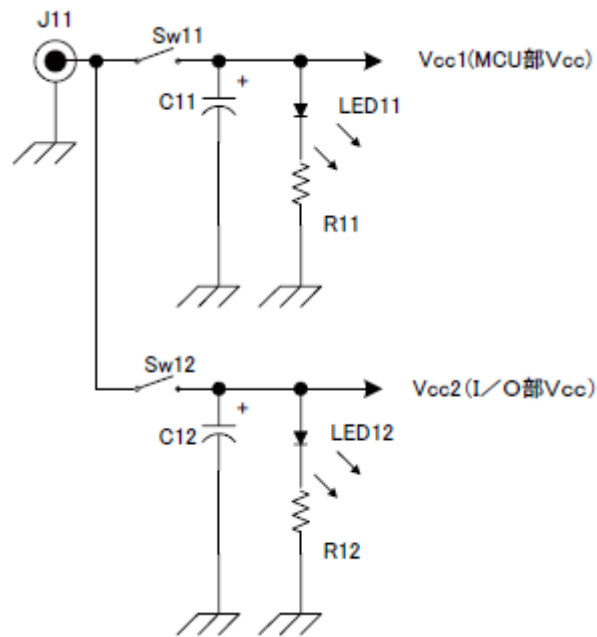


図 8 電源回路

■ H-UD I 接続回路

E 8 a (またはE 8) エミュレータ本体とユーザーシステム (マイコンボード: Seamark-I) を接続するコネクタ (CN 3) とその周辺回路、プルアップ抵抗から構成されます。

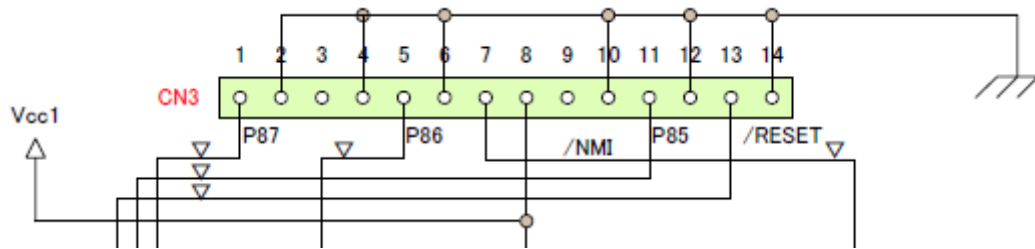


図 9 H-UD I 接続回路

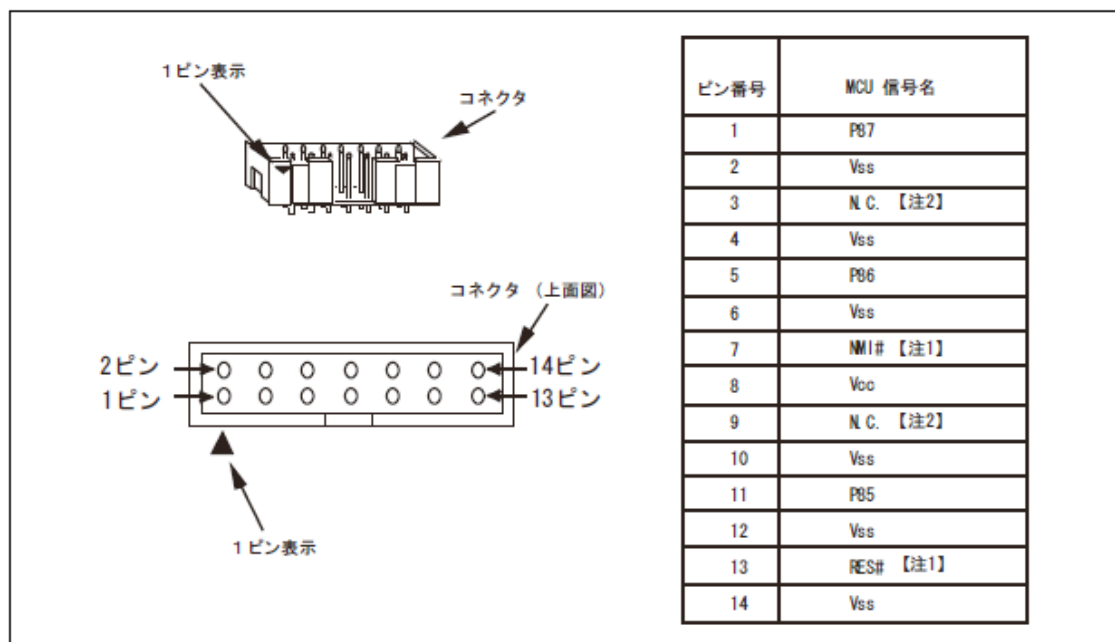


図 10 E 8 a 接続コネクタのピン配置

1. 3 基本 I/O 回路

基本 I/O 部は、下記電子部品より構成されています。

- ・ LED : 4
- ・ 圧電ブザー (他励式) : 1
- ・ 半固定抵抗 : 1
- ・ タクトスイッチ (プルアップ抵抗のオン/オフ可能なジャンパーSW 付き) : 2
- ・ DIP スイッチ (4 回路) : 1
- ・ 7 セグメント LED : 1
- ・ トランジスタ用ソケット : 1 (トランジスタは未実装)
- ・ 14 ピン IC ソケット : 1 (今回の講座では 74HC07AP を搭載します)

LED のを点灯/消灯、SW のオン/オフ操作時の入力端子の信号レベルの確認、TTL IC の使う実験する場合は、マイコンボードの I/O Power SW (SW12) をオンにします。

■ LED

基本 I/O ボードの LED1~4 は、基本 I/O 回路の Term1 の 1 番から 4 番 (/LED1~ /LED4) を Low にすると点灯します。このとき LED には約 10 mA の電流が流れるように設計されています。表 3 の H8/3694F DC 特性からポート 8 を出力 Low としたときのポート 8 に流れ込む電流は最大で 20.0 mA まで許容されていますので、ポート 8 (大電流出力ポート) を使用すると LED 1~4 を Low アクティブで直接ドライブできます。

それ以外のポートを使用して LED 1~4 をドライブするとマイコンの出力ポート許容電流の最大定格を超えてしまい、当該ポートが破壊される可能性があります。

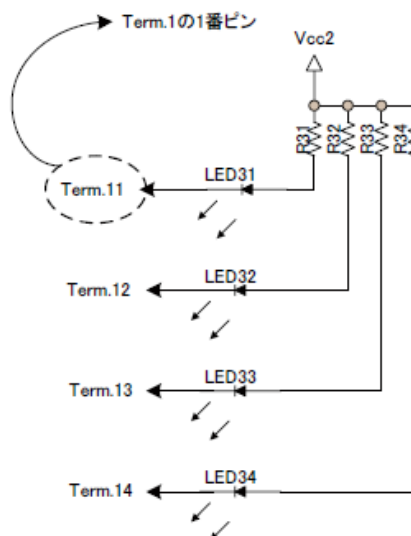


図 1 1 LED 回路

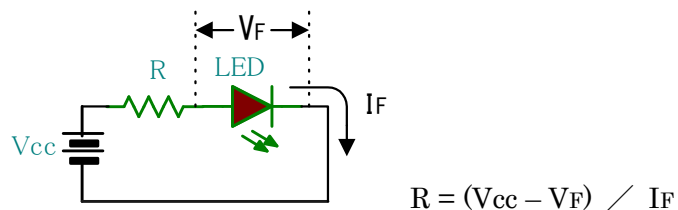
表 3 H8/3694F D C 特性

| 項目 | 記号 | 適用端子 | 測定条件 | 規格値 | | | 単位 |
|----------------------------------|---------------------|---------------|---------------------------|-----|-----|------|----|
| | | | | Min | Typ | Max | |
| 出力 Low レベル 許容電流 (1 端子あたり) | I _{OL} | ポート 8、SCL、SDA | V _{CC} =4.0~5.5V | — | — | 2.0 | mA |
| | | 以外の出力端子 | | | | | |
| | | ポート 8 | | — | — | 20.0 | mA |
| | | ポート 8、SCL、SDA | | | | 0.5 | mA |
| | | 以外の出力端子 | | | | | |
| | | ポート 8 | | — | — | 10.0 | mA |
| 出力 Low レベル 許容電流 (総和) | Σ I _{OL} | ポート 8、SCL、SDA | V _{CC} =4.0~5.5V | — | — | 40.0 | mA |
| | | 以外の出力端子 | | | | | |
| | | ポート 8、SCL、SDA | | — | — | 80.0 | mA |
| | | ポート 8、SCL、SDA | | — | — | 20.0 | mA |
| | | 以外の出力端子 | | | | | |
| 出力 High レベル 許容電流 (1 端子あたり) | - I _{OH} | 全出力端子 | V _{CC} =4.0~5.5V | — | — | 2.0 | mA |
| | | | | | | 0.2 | mA |
| 出力 High レベル 許容電流 (総和) | - Σ I _{OH} | 全出力端子 | V _{CC} =4.0~5.5V | — | — | 30.0 | mA |
| | | | | | | 8.0 | mA |

ポート 8 は、1 端子あたりの出力 Low レベルの許容電流（シンク許容電流）が 20mA あり、他のポート端子に比べて許容電流値は大きい。

◆ 参考：LED回路の設計

下記回路のLEDの順方向電圧を V_F 、順方向電流を I_F とするとLED回路の電流制限抵抗 R は次式で求められます。



順方向電圧や順方向電流を正確に知るにはデータシートを参照する必要があります。アバウトな計算をするときは、目安としては順方向電圧 V_F を 2.0 v 、順方向電流 I_F を 10 mA と考えて計算します。 V_{cc} を 5.0 v とすると電流制限抵抗は次式から算出できます。

$$R = (V_{cc} - V_F) / I_F = 3\text{ v} / 10\text{ mA} = 300\ \Omega$$

図 11 の回路では $330\ \Omega$ の抵抗が入っています。

■ 圧電ブザー

圧電ブザーは他励式ブザーです。外部から Term1 の 6 番（B z）に一定周波数のパルスを入力すると圧電ブザーが鳴動します。

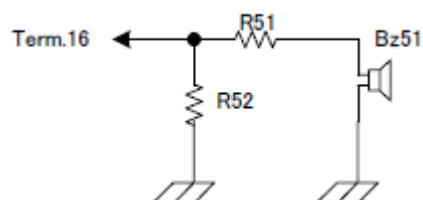


図 1 2 圧電ブザー回路

■ 半固定抵抗

半固定抵抗（VR41）を廻すことで5 vを分圧した電圧が Term1 の5番（Volume）に出力されます。

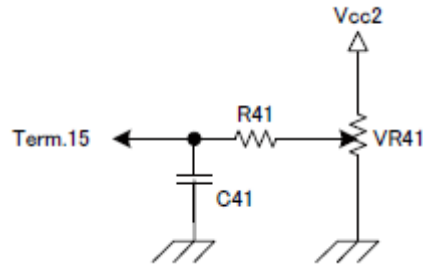


図 1 3 半固定抵抗回路

■ タクトスイッチ

タクトスイッチ1（SW71）とタクトスイッチ2（SW72）にはプルアップ抵抗を接続する／しないを設定できるジャンパSW（JP71、JP72）が取り付けられています。ジャンパSW（JP71、JP72）は、2と3を短絡するとプルアップ抵抗が Vcc に接続されます。プルアップ抵抗を接続した状態でタクトスイッチを押すと、Term4 の1番と2番には、Low レベルの信号が出力されます。

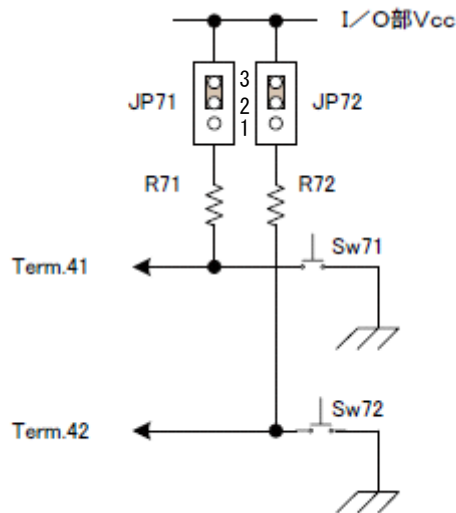


図 1 4 タクトスイッチ回路

■ DIPスイッチ

4回路を内蔵したDIPスイッチはプルアップ用の集合抵抗が接続されています。4回路のそれぞれのスライドスイッチを「ON」側へセットすると Term5 の1番から4番の各端子はLowになります。

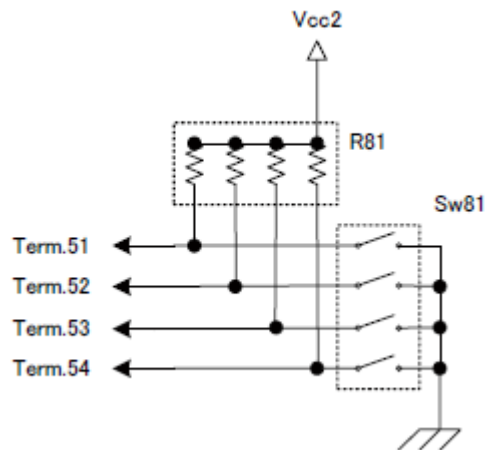


図 1 5 DIPスイッチ回路

■ 7セグメントLED

7セグメントLED (7seg101) はアノードコモン(共通)の7セグメントLEDです。シンクドライバ (IC3 : TD62083) を使ってドライブします。TD62083はインバータ方式のシンクドライバなので、各セグメントの制御端子 Term7 の1番から8番 (Seg. a ~ Seg. dp) を High にすると7セグメントLEDの各セグメントが点灯します。

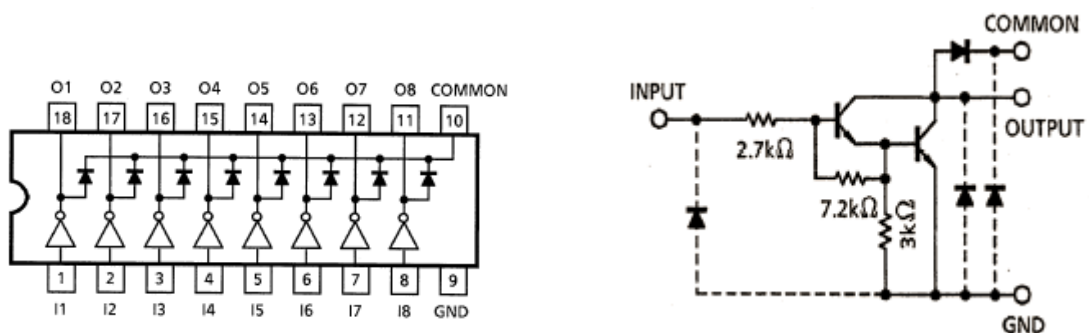


図 1 6 IC3 : TD62083 (IC3) ピン接続図と基本回路図

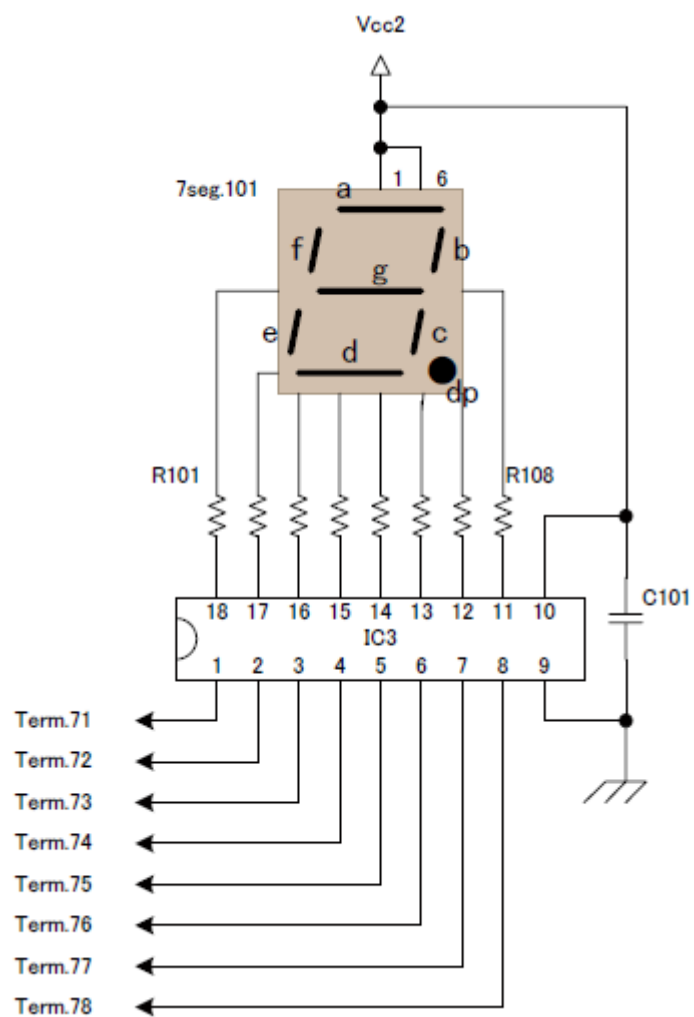


図 1 7 7セグメントLED回路

■ トランジスタ用ソケット

トランジスタやFETを接続するためのソケットで、電気的にはどこにも接続されていません。

■ 14ピンICソケット：IC2

14ピンICソケットの14ピンが5V電源（Vcc）へ、7ピンは0V（GND）に接続されています。

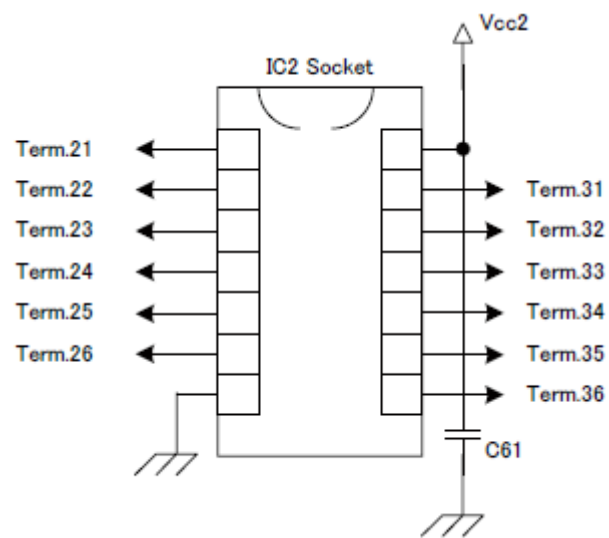


図 1 8 1 4 ピン I C ソケット回路図

【実習1】基本I/Oボードの動作確認

■ 下記電子デバイスのホットテストを実施します。E8aエミュレータは使いません。

- ① LED1～4
- ② 7セグメントLED
- ③ タクトスイッチ1, タクトスイッチ2

■ 手順

実習にあたり、まずACアダプタ (5.0v、2.1φ、センター+) をマイコンボードに接続します。ボードの基本I/O回路側の電源スイッチ (SW12) をONします。基本I/O回路側の電源LED (LED12) が点灯します。

① LED1～4のテスト：

LED1～4の端子 (Term1の1番 (/LED1) から4番 (/LED4)) を順次GNDへ接続します。

⇒ LED1からLED4は点灯しますか。

② 7セグメントLEDの端子 (Term7の1番 (Seg.a) から8番 (Seg.dp)) を順次Vccに接続します。

⇒ 7セグメントLEDのセグメントa～dpは順次点灯しますか。

③ SW1、SW2のプルアップ抵抗が「ある」場合と「ない」場合について、当該スイッチ端子 (/SW1、/SW2) の電圧レベルを観測します。

⇒ スイッチ1のプルアップ抵抗用ジャンパ (JP71) の2番と3番をショートしてプルアップ抵抗を接続とした状態で、スイッチ1 (SW71) を押したときは0v、離れたときは大よそ5vになりますか？

⇒ スイッチ2についてもスイッチ1同様に確認します。

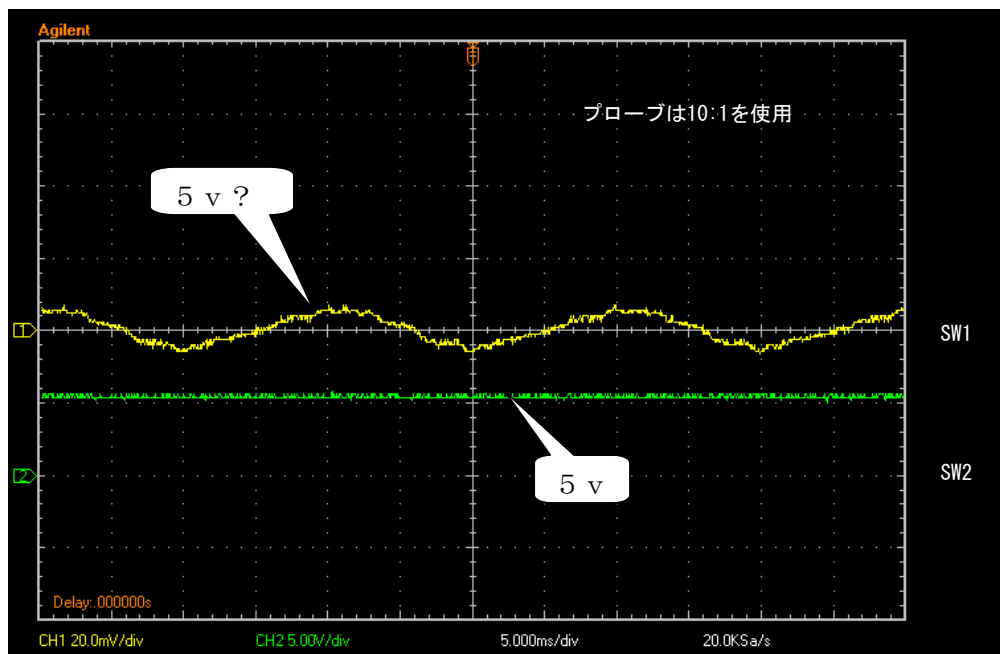
⇒ スイッチ1のプルアップ抵抗用ジャンパ (JP71) の1番と2番をショートしてプルアップ抵抗をVccから切り離れた状態で、スイッチ1 (SW71) を押したときは0v、離れたときは大よそ5vになりますか？

⇒ スイッチ2についてもスイッチ1同様に確認します。

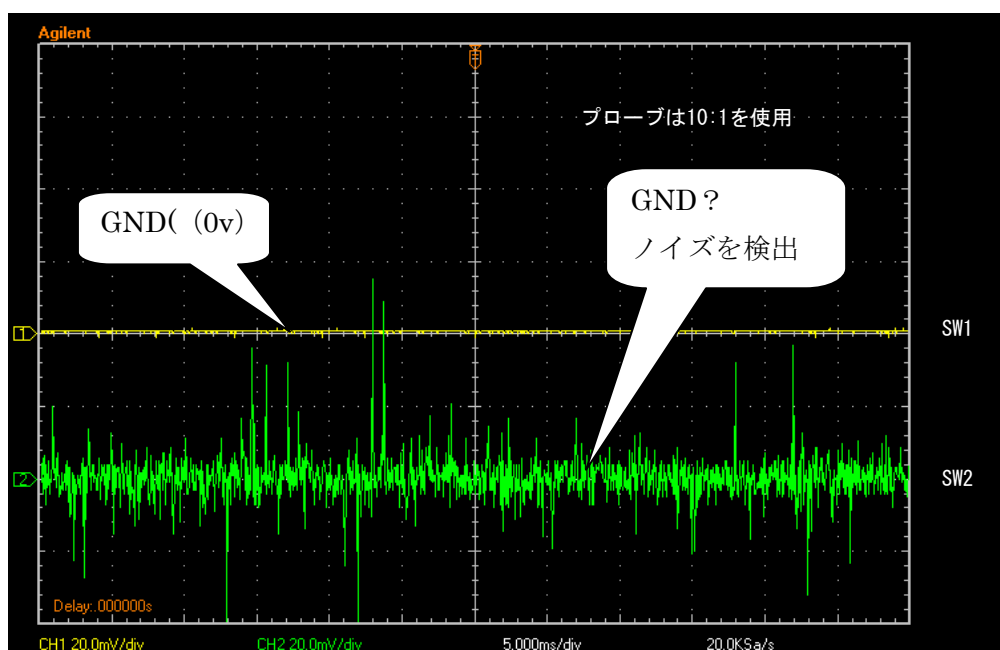
◆ひとやすみ◆

手作業でLEDの点消灯やスイッチ状態 (電圧レベル) を確認することを、プログラムコードを書いて自動的に処理しているのがソフトウェアということができます。即ち、ソフトウェアとは、「**手作業を自動化したもの**」という見かたもできます。LEDの点灯/消灯方法やスイッチの状態確認方法が手作業でできないということはソフトウェアの設計や動作確認ができないということです。

参考波形1：スイッチ1（SW1）がプルアップ抵抗なし、スイッチ2（SW2）がプルアップ抵抗ありのときの各スイッチが**オフ状態**の電圧波形
（電圧レンジ CH1(SW1)：20.0 mV/div CH2(SW2)：5.0 V/div）



参考波形2：スイッチ1（SW1）がプルアップ抵抗なし、スイッチ2（SW2）がプルアップ抵抗ありのときの各スイッチが**オン状態**の電圧波形
（電圧レンジ CH1(SW1)：20.0 mV/div CH2(SW2)：20.0 mV/div）

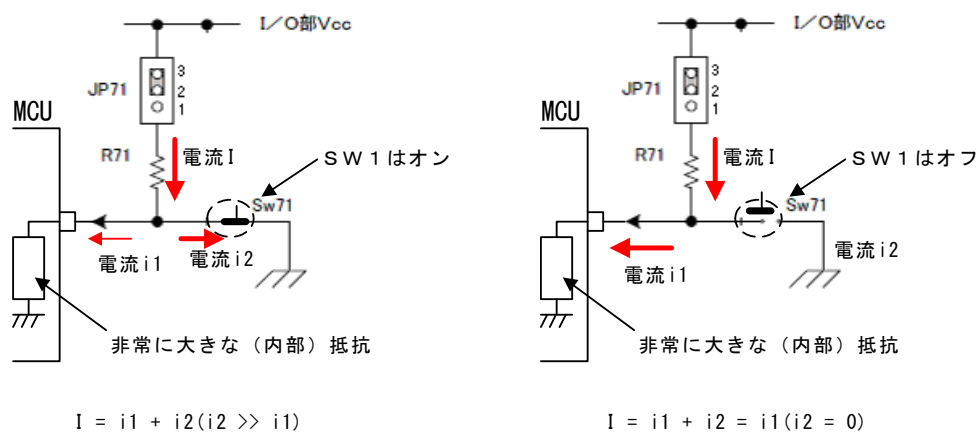


◆ 解説 1：プルアップ抵抗の機能

電流は電圧の高いところから低いところへ流れます。水の流れと一緒にです。

スイッチ 1 (SW71) をオンするとほとんどの電流が電源 V_{cc} からプルアップ抵抗 $R71$ 、そして短絡しているスイッチ 1 (SW71) を通して GND に向かって流れます。電流がマイコン入力ポート側へほとんど流れないのはマイコンの**入力ポートの内部抵抗が非常に大きい**ためです。このときマイコンポート側 (Term4 の/SW1) の端子電圧は 0 v です。

スイッチ 1 (SW71) をオフすると、電流は電源 V_{cc} からプルアップ抵抗 $R71$ を介して電流は全てマイコンポート側へ流れます。スイッチ 1 (SW71) がオフのためで、このとき、 V_{cc} (5 v) をプルアップ抵抗 ($4.7\text{ k}\Omega$) と入力ポートの内部抵抗で分圧するのでほとんど 5 v に近い電圧となります。



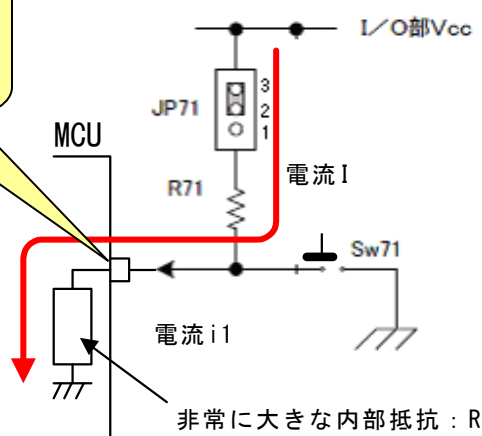
◆ 解説 2：プルアップ抵抗の値は？

プルアップ抵抗をどんどん小さくすると、限りなく V_{cc} に近い High レベル電圧を生成できます。抵抗値の小さいプルアップ抵抗が、High レベルとしては良いプルアップ抵抗なのでしょうか？たとえば 1Ω のプルアップ抵抗を付けた場合はスイッチを押したときは $5A$ ($5V/1\Omega = 5A$) も流れます。回路パターンが焼けて煙が出るでしょう。

それでは大きな抵抗を付ければいいのでしょうか？たとえば $1M\Omega$ の抵抗を付けると、スイッチを離したときはプルアップ抵抗の $1M\Omega$ とマイコン入力ポートの内部抵抗（MCU内部抵抗を仮に $1M\Omega$ とすると）で V_{cc} の $5V$ を分圧しますから、 $2.5V$ となる計算です。スイッチをオフしたときの電圧レベルが $2.5V$ では High とも Low とも判定しない「不定」レベルです。

結局、プルアップ抵抗値はマイコンのDC特性で許容している入力 High レベル電圧（H8/3694Fでは $V_{cc} \times 0.8$ ）より十分高く、かつスイッチをオンしたときに流れる電流がスイッチ回路の許容電流を越えない（一般的には数ミリアンペア程度）値に設定する必要があります。

MCU 端子電圧は、 R_{71} と MCU 内部抵抗 R で分圧される



$$MCU \text{ 端子電圧} = V_{cc} \times R / (R + R_{71})$$

2

HEWによる開発環境



2. 1 概要

マイコンのソフトウェア開発では、エディタ、コンパイラ、エミュレータといったツールが必要です。さらに開発ニーズや各種マイコン特性に合わせて多種多様な組み合わせが必要となります。統合開発環境 **High-performance Embedded Workshop** は、それらのツールを使いやすく統合するフレームワーク です。

HEWによる開発環境構築例を図1に示します。

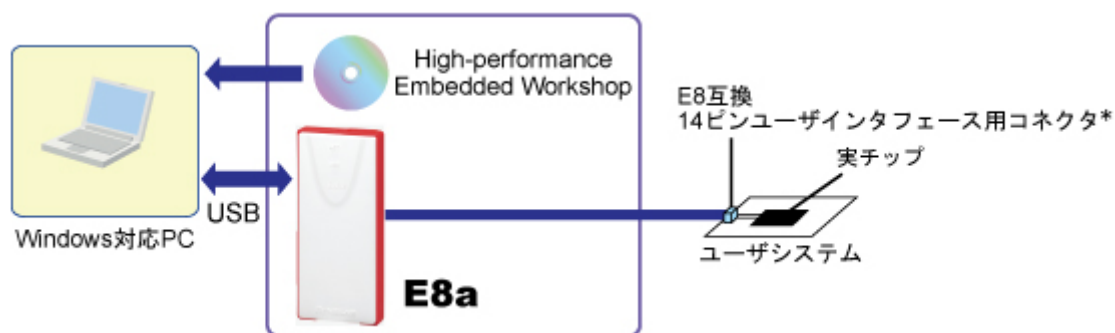


図1 HEWによる開発環境構築例

E8aエミュレータパッケージ付属のCD-ROMをインストールすると図2に示すHEW (High-performance Embedded system Workshop) の環境が提供されます。

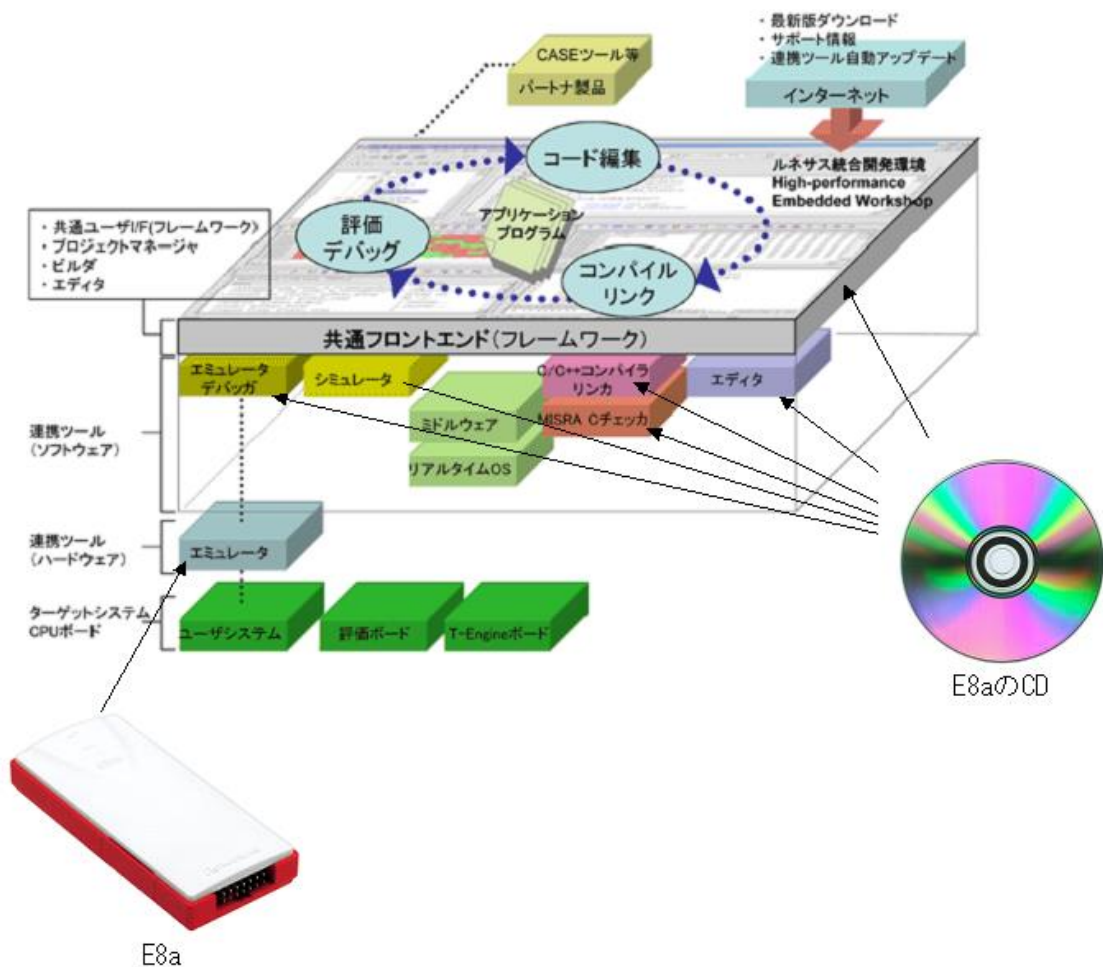
◆ 参考

ルネサスエレクトロニクス社のホームページから無償評価版の「H8SX, H8S, H8ファミリ用 C/C++コンパイラパッケージ」（シミュレータデバッガ付属）をダウンロードすることができます。

<http://japan.renesas.com/products/tools/ide/hew/downloads.jsp>

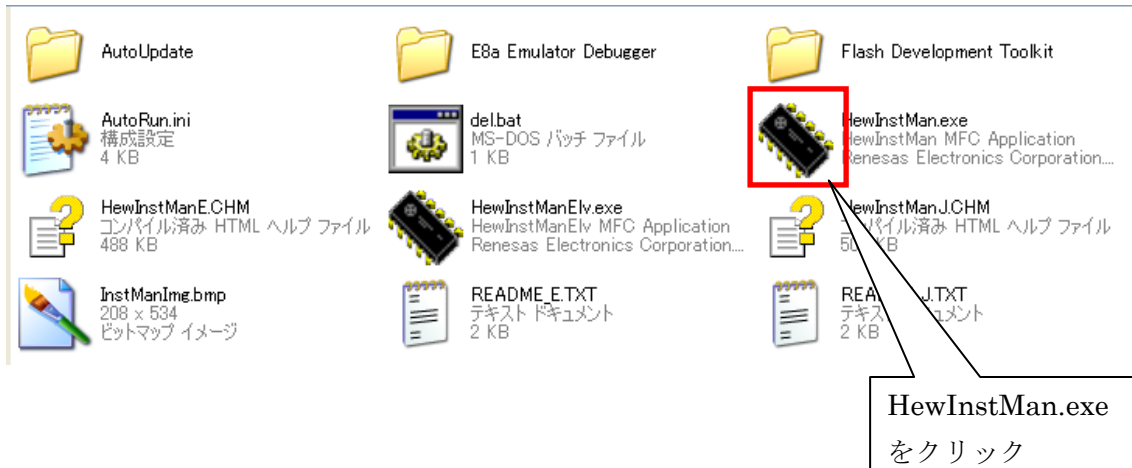
このパッケージには
ルネサス統合開発環境 HEW（High-performance-Embedded Workshop）が同包されていますが、エミュレータ・デバッガは付属しておりません。

したがってターゲットボードでのエミュレーションデバッグはできませんが、エディタ、ビルド（コンパイル／リンク）、シミュレータ機能を使うことはできます。



2. 2 E8a エミュレータパッケージ付属CD-ROMのインストール手順

① CD-ROM を開くと下記のような各種ファイルが見えます。



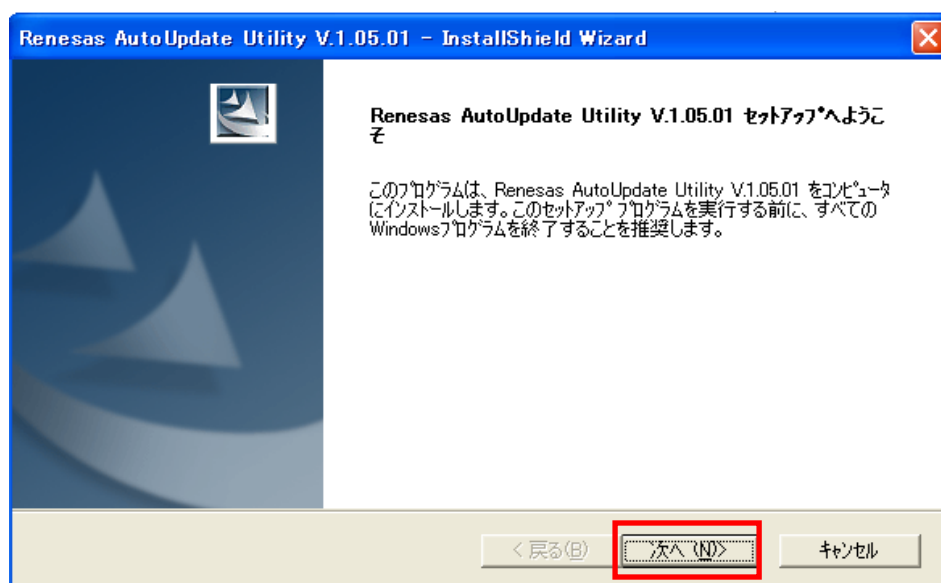
② **HewInstMan.exe** をクリックして起動します。

③ **1 of 3 Installer** : 「E8a Emulator Software」のインストール
表示された画面の指示に従って「次へ (N)」をクリックします。
最後に下記画面が表示されます。「完了」をクリックします。



「E8a Emulator Software」のインストールが完了すると、自動的に「**2 of 3 Installer**」が実行されます。

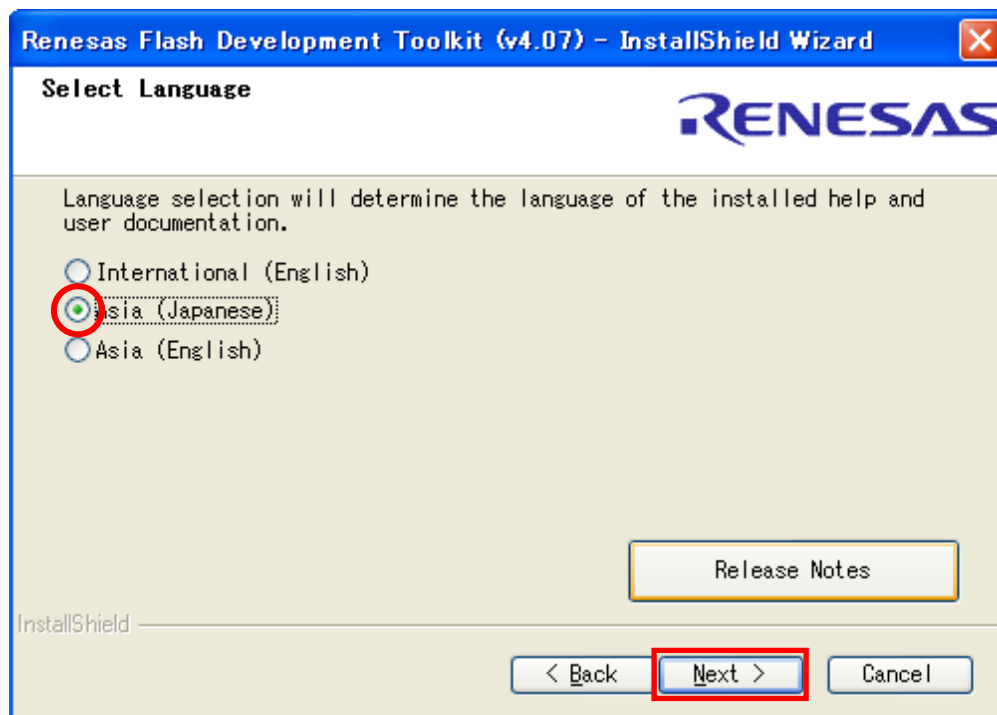
④ **2 of 3 Installer** : 「Renesas AutoUpdate Utility」のインストール
表示された画面の指示に従って「次へ (N)」をクリックします。



「Renesas AutoUpdate Utility」のインストールが完了すると、自動的に「3 of 3 Installer」が実行されます。

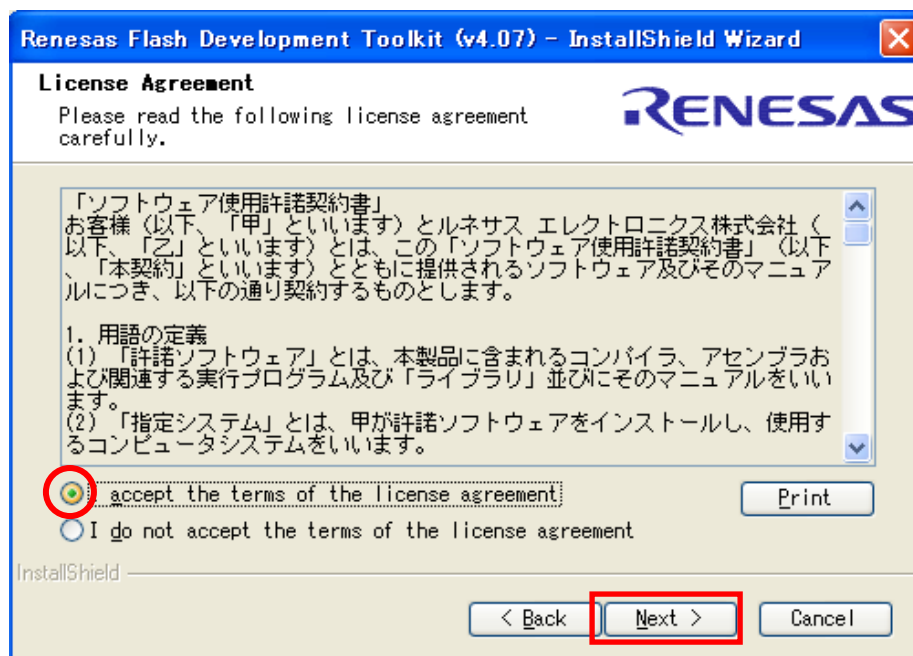
⑤ **3 of 3 Installer** : 「Renesas Flash Development Tool」のインストール

表示された画面の指示に従って「次へ (N)」をクリックします。言語選択画面では「Asia(Japanese)」にチェックをいれて、「Next >」をクリックします。



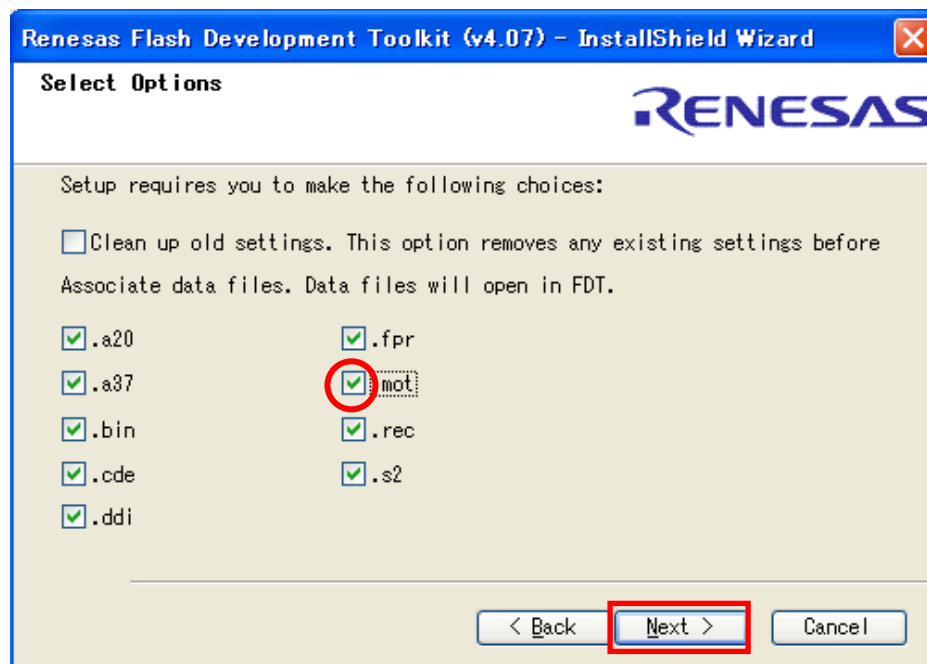
⑥ ライセンスの認証

License 承認画面では「accept the terms of the license agreement」にチェックを入れて「Next >」をクリックします。



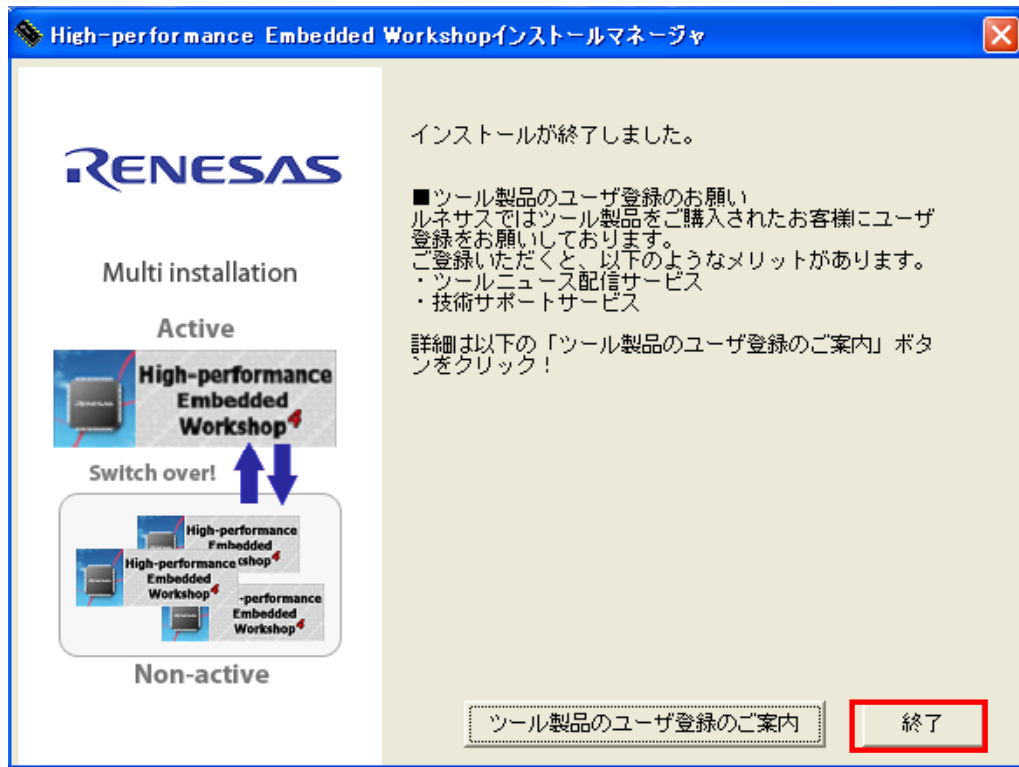
⑦ オプション選択

オプション選択画面では「.mot」にチェックを入れて「Next >」をクリックします。



⑧ インストール終了

下記画面が表示されたら、「終了」をクリックします。



2. 3 ワークスペースとプロジェクト

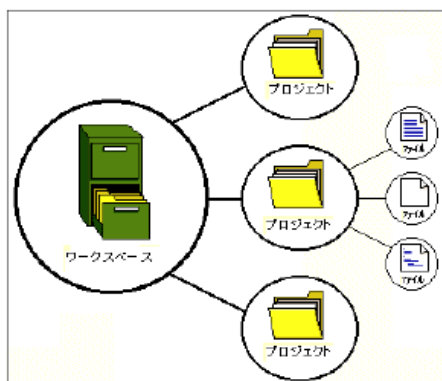
High-performance Embedded Workshopでは、ワークスペースとプロジェクトという概念があります。

・ ワークスペース

High-performance Embedded Workshopを用いてプログラムを作成する場合の、一番大きな管理単位です。ワークスペースには、最低1つのプロジェクトが必要で、この1つのプロジェクトは、ワークスペース作成時に自動的に作成されます。また、ワークスペースは複数のプロジェクトを持つことができます。

・ プロジェクト

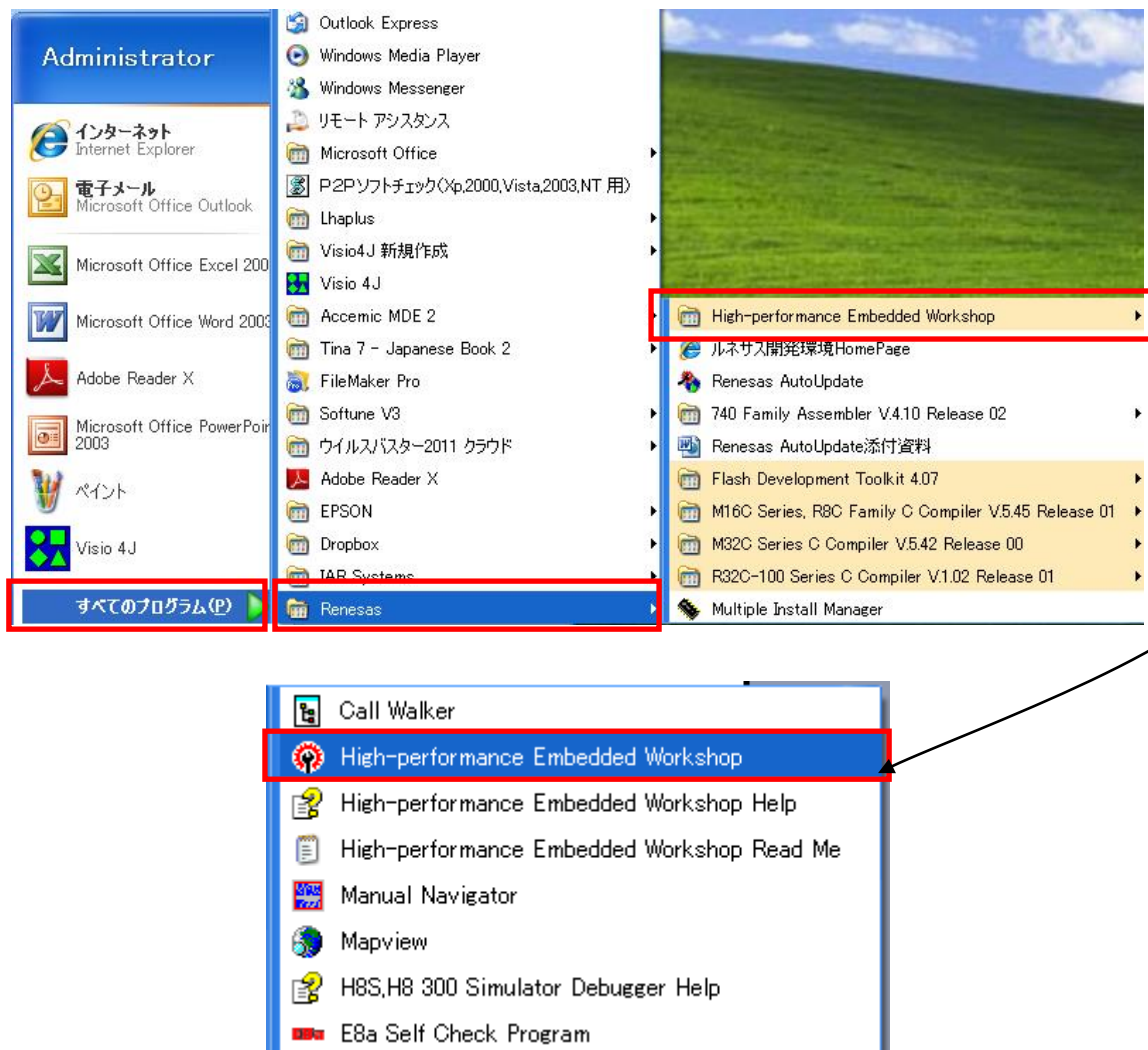
プログラムを作成する場合、特定機能をライブラリ化してモジュールを階層化する場合があります。このような場合、ライブラリ用のプロジェクトを作成、追加することができます。



2. 4 HEWの起動

インストールが全て完了したら、HEW を起動します。

「すべてのプログラム (P)」 > 「Renesas」 > 「High-performance Embedded Workspop」 > 「High-performance Embedded Workspop」を選択します。

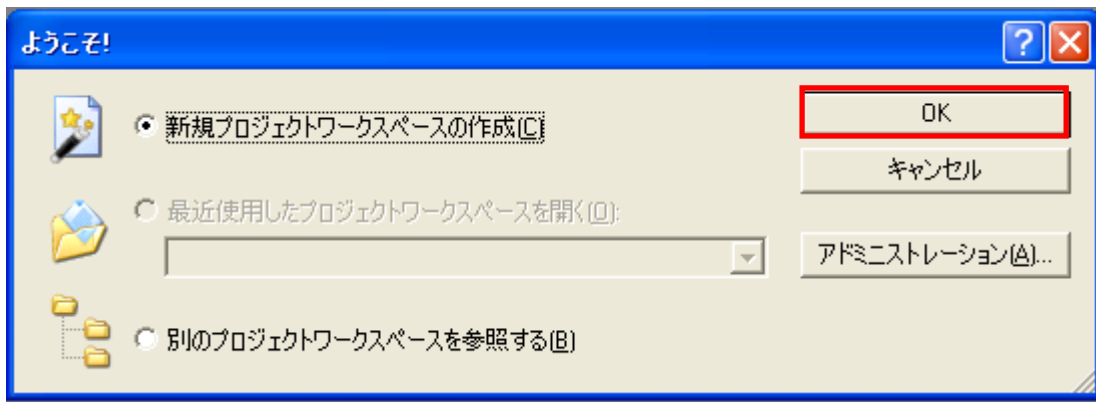


次に示す「ようこそ！」画面が表示されます。下記の手順にしたがって新規プロジェクトの生成を行います。

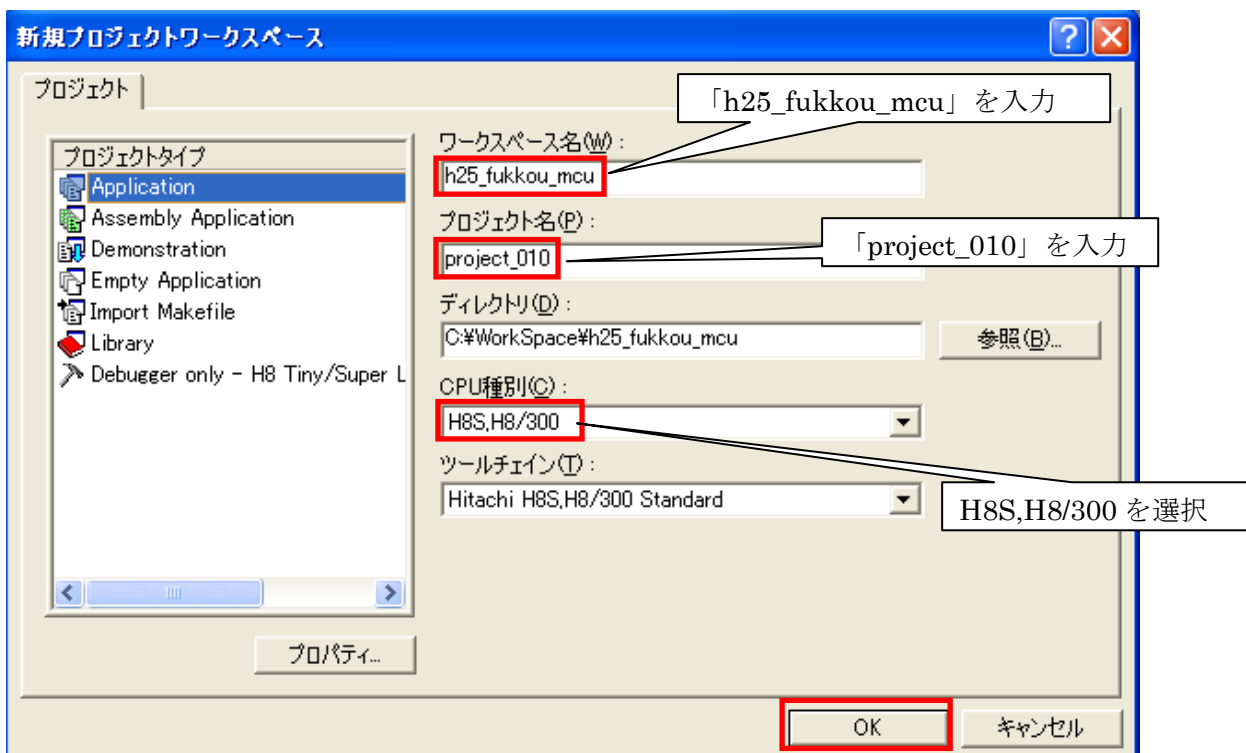
2. 5 新規プロジェクトの作成

新規プロジェクトを作成する手順を示します。

「ようこそ」画面で「新規プロジェクトワークスペースの作成」にチェックをいれて「OK」をクリックします。



ワークスペース名は「h25_fukkou_mcu」、プロジェクト名は「project_010」とします。







◆ 参考

I/O ライブラリ使用：

チェックを入れると標準入出力ライブラリを活用できます。

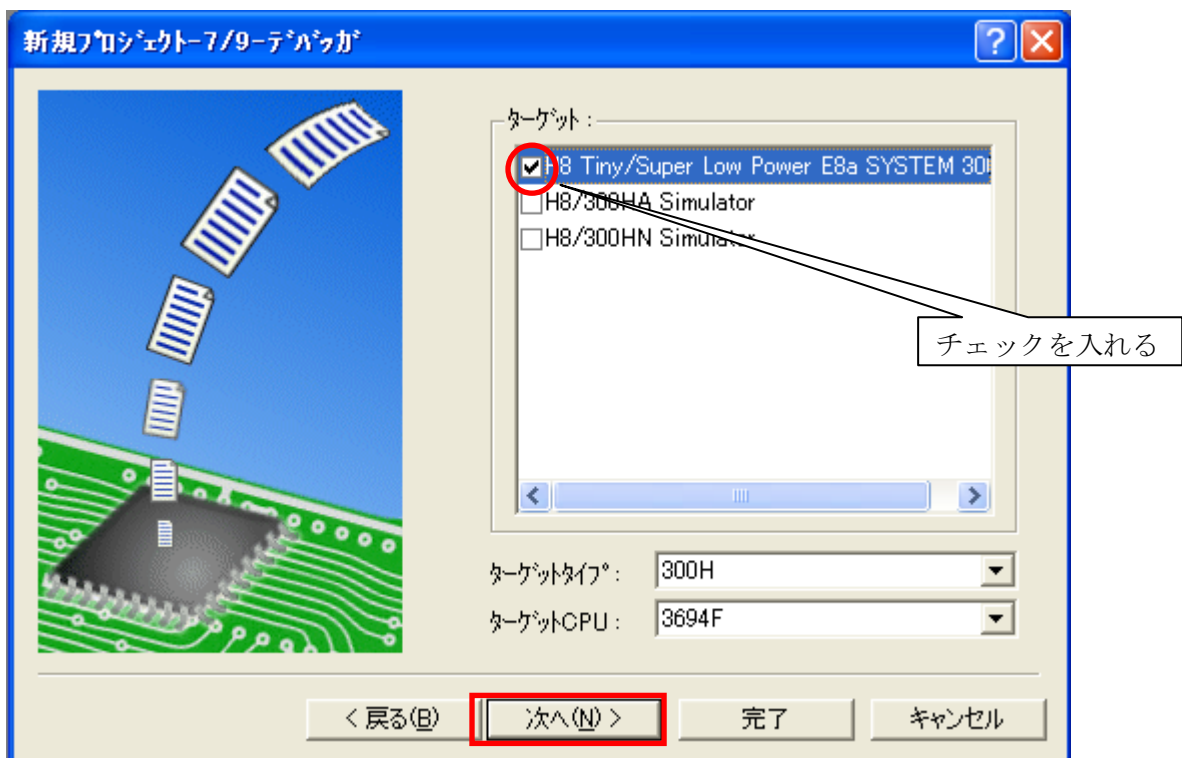
ヒープメモリ使用：

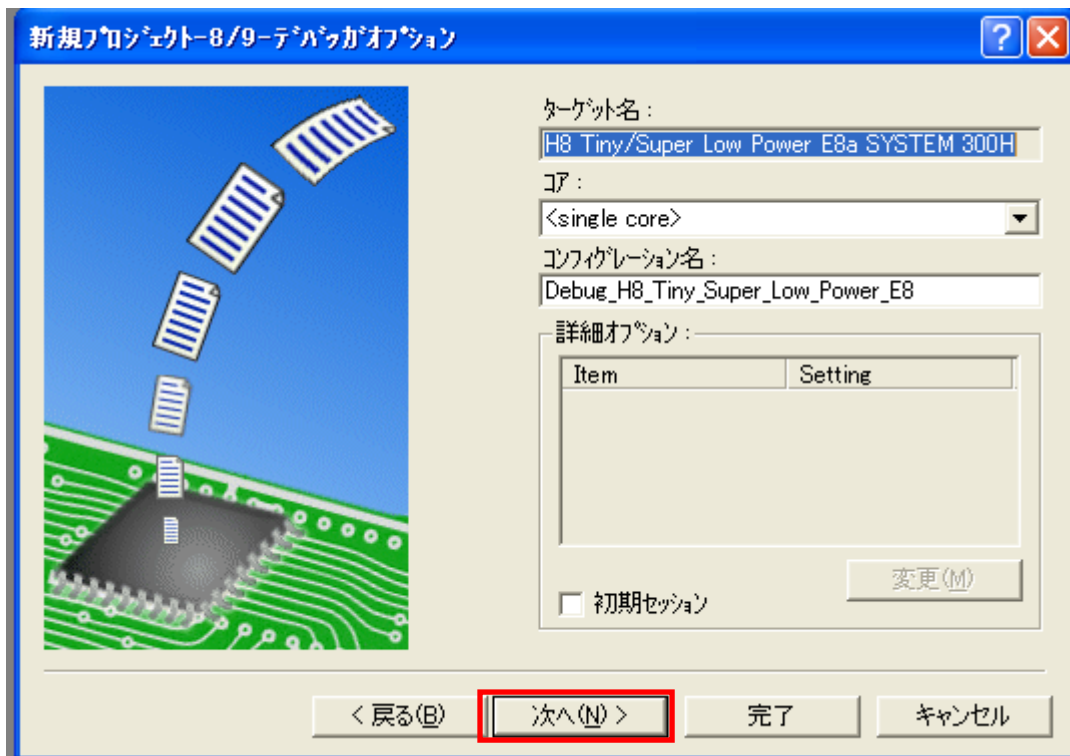
ヒープメモリとはライブラリ関数 `malloc`, `realloc`, `calloc`, `new` で使用する領域のこと、チェックを入れるとヒープ領域管理用の関数 `sbrk()` を使用できます。

I/Oレジスタ定義ファイル：

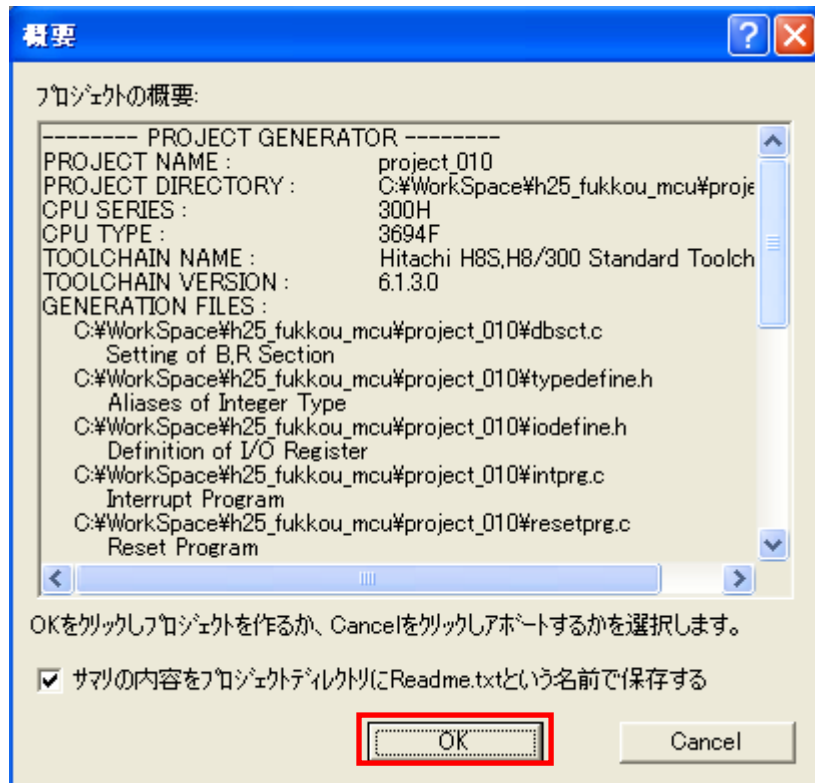
チェックするとC言語で記述したI/O定義ファイル ("`iodef.h`") を生成します。



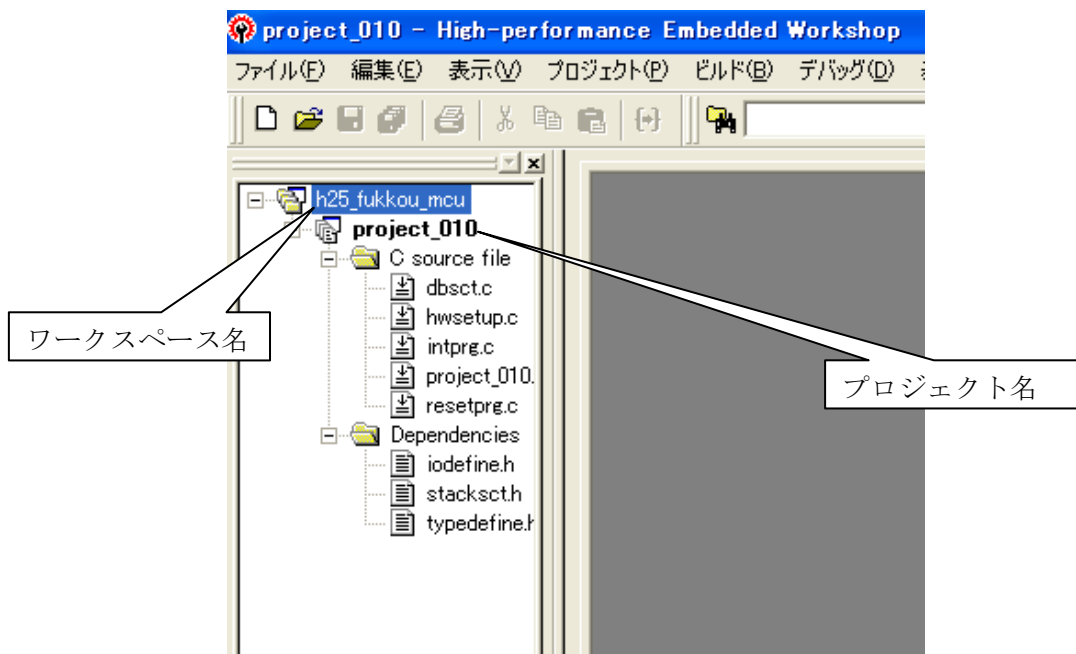




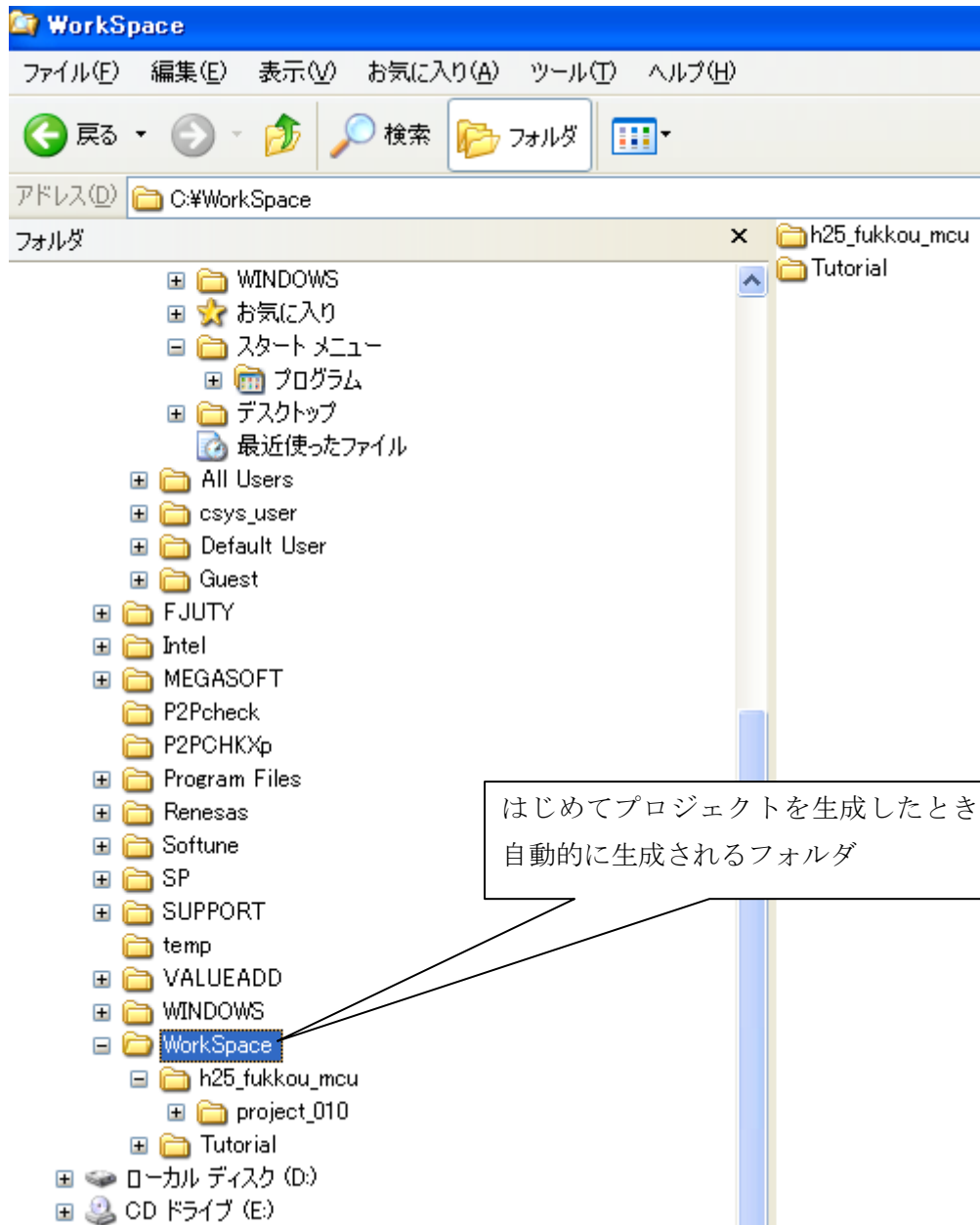
「概要」画面で「OK」をクリックします。



新規プロジェクトの生成が完了です。次のプロジェクト project_010 のH E W初期画面が表示されます



PC上のフォルダ構成



新規プロジェクト「project_010」の生成が完了するとパソコンの「Workspace」フォルダの中に「h25_fukkou_mcu」というワークスペースが、その下部に「project_010」というプロジェクトが生成されます。

2. 6 コーディング、ビルド（ロードモジュールの生成）

図 4 にHEWのメインウィンドウを示します。メインウィンドウには3つのウィンドウがあります。それぞれワークスペースウィンドウ、エディタウィンドウ、アウトプットウィンドウと称します。このメインウィンドウでコード生成、ビルド（コンパイル／アセンブル／リンク）を行います。

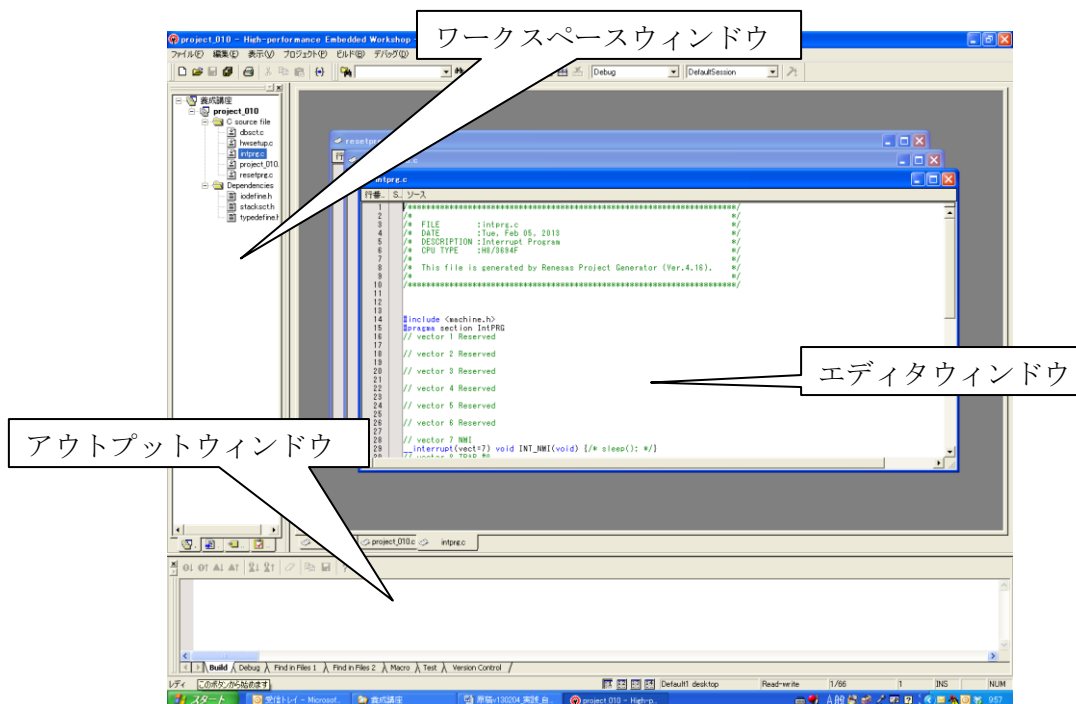


図 4 HEWのメインウィンドウ

【実習2】スイッチ1（SW1）の操作でLED1 が点灯／消灯するプログラムを作成します。

ワークスペースウィンドウでソースコードの生成・ビルドを行い、ダウンロードモジュールを作成します。

- ①LED1：P80（ポート8の第0ビット）に出力ポートとして割り付けます。
- ②SW1：P10（ポート1の第0ビット）に入力ポートとして割り付けます。

ポートを入力として使用するか、出力として使用するかはポートコントロールレジスタ（PCR）で指定します。ポートコントロールレジスタは、読み書き専用ポート以外は各ポートごとに用意されており、1で出力ポート、0で入力ポートに切り替わります。使用に先立って入力／出力の設定が必要です。一般には「hwsetup.c」ファイルの中で設定します。

ポートコントロールレジスタ8（PCR8）

PCR8 はポート 8 の汎用入出力ポートとして使用する端子の入出力をビットごとに選択します。

| ビット | ビット名 | 初期値 | R/W | 説 明 |
|-----|-------|-----|-----|---|
| 7 | PCR87 | 0 | W | 汎用入出力ポートの機能が選択されているとき、このビットを1にセットすると対応する端子は出力ポートとなり、0にクリアすると入力ポートとなります。 |
| 6 | PCR86 | 0 | W | |
| 5 | PCR85 | 0 | W | |
| 4 | PCR84 | 0 | W | |
| 3 | PCR83 | 0 | W | |
| 2 | PCR82 | 0 | W | |
| 1 | PCR81 | 0 | W | |
| 0 | PCR80 | 0 | W | |

P80(LED1)は出力ポートなので0を1に書き換える。

コーディング例

```
IO.PCR8 = 0x01;
```

ポートコントロールレジスタ1（PCR1）

PCR1 はポート 1 の汎用入出力ポートとして使用する端子の入出力をビットごとに選択します。

| ビット | ビット名 | 初期値 | R/W | 説 明 |
|-----|-------|-----|-----|---|
| 7 | PCR17 | 0 | W | PMR1により汎用入出力ポートの機能が選択されているとき、このビットを1にセットすると対応する端子は出力ポートとなり、0にクリアすると入力ポートとなります。 ビット3はリザーブビットです。 |
| 6 | PCR16 | 0 | W | |
| 5 | PCR15 | 0 | W | |
| 4 | PCR14 | 0 | W | |
| 3 | — | — | — | |
| 2 | PCR12 | 0 | W | |
| 1 | PCR11 | 0 | W | |
| 0 | PCR10 | 0 | W | |

P10(SW1)は入力ポートなので0を0に書き換える。

コーディング例

IO.PCR1 = 0x00;

ポートデータレジスタ8 (PDR8)

PDR8 はポート 8 の汎用入出力ポートデータレジスタです。

| ビット | ビット名 | 初期値 | R/W | 説 明 |
|-----|------|-----|-----|---|
| 7 | P87 | 0 | R/W | 汎用出力ポートの出力値を格納します。 このレジスタをリードすると、PCR8 がセットされているビットはこのレジスタの値が読み出されます。PCR8 がクリアされているビットはこのレジスタの値にかかわらず端子の状態が読み出されます。 |
| 6 | P86 | 0 | R/W | |
| 5 | P85 | 0 | R/W | |
| 4 | P84 | 0 | R/W | |
| 3 | P83 | 0 | R/W | |
| 2 | P82 | 0 | R/W | |
| 1 | P81 | 0 | R/W | |
| 0 | P80 | 0 | R/W | |

LED1(P80)が点灯のとき 0 (Low)、消灯のとき 1 (High)を書きます。

コーディング例

IO.PDR8.BIT.B0 = 0; /* LED1 点灯 */

IO.PDR8.BIT.B0 = 1; /* LED1 消灯 */

ポートデータレジスタ 1 (PDR1)

PDR1 はポート 1 の汎用入出力ポートデータレジスタです。

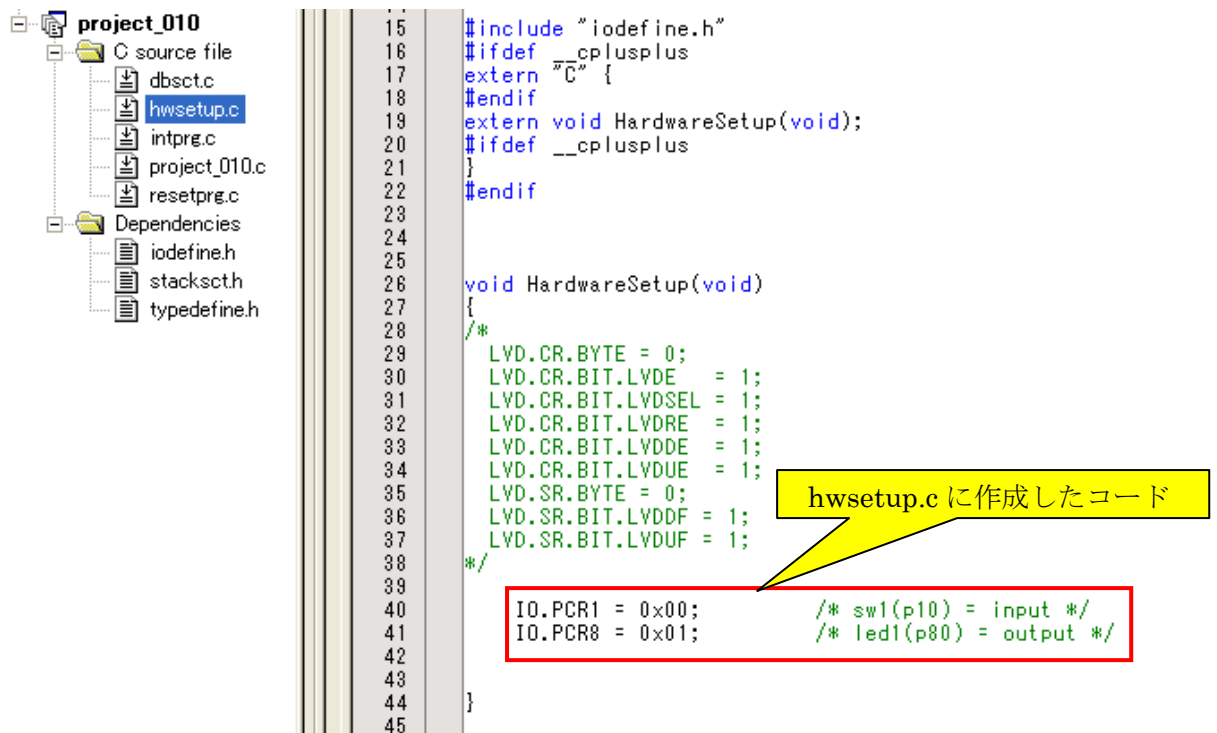
| ビット | ビット名 | 初期値 | R/W | 説 明 |
|-----|------|-----|-----|--|
| 7 | P17 | 0 | R/W | PDR1 はポート 1 の出力値を格納するレジスタです。 このレジスタをリードすると、PCR1 がセットされているビットはこのレジスタの値が読み出されます。PCR1 がクリアされているビットはこのレジスタの値にかかわらず端子の状態が読み出されます。 ビット3はリザーブビットです。リードすると常に1が読み出されます。 |
| 6 | P16 | 0 | R/W | |
| 5 | P15 | 0 | R/W | |
| 4 | P14 | 0 | R/W | |
| 3 | — | 1 | — | |
| 2 | P12 | 0 | R/W | |
| 1 | P11 | 0 | R/W | |
| 0 | P10 | 0 | R/W | |

SW1 を押すと P10 には 0 (Low)、離すと 1 (High)が入力されます。

P10 の c 言語での記述 : IO.PDR1.BIT.B0

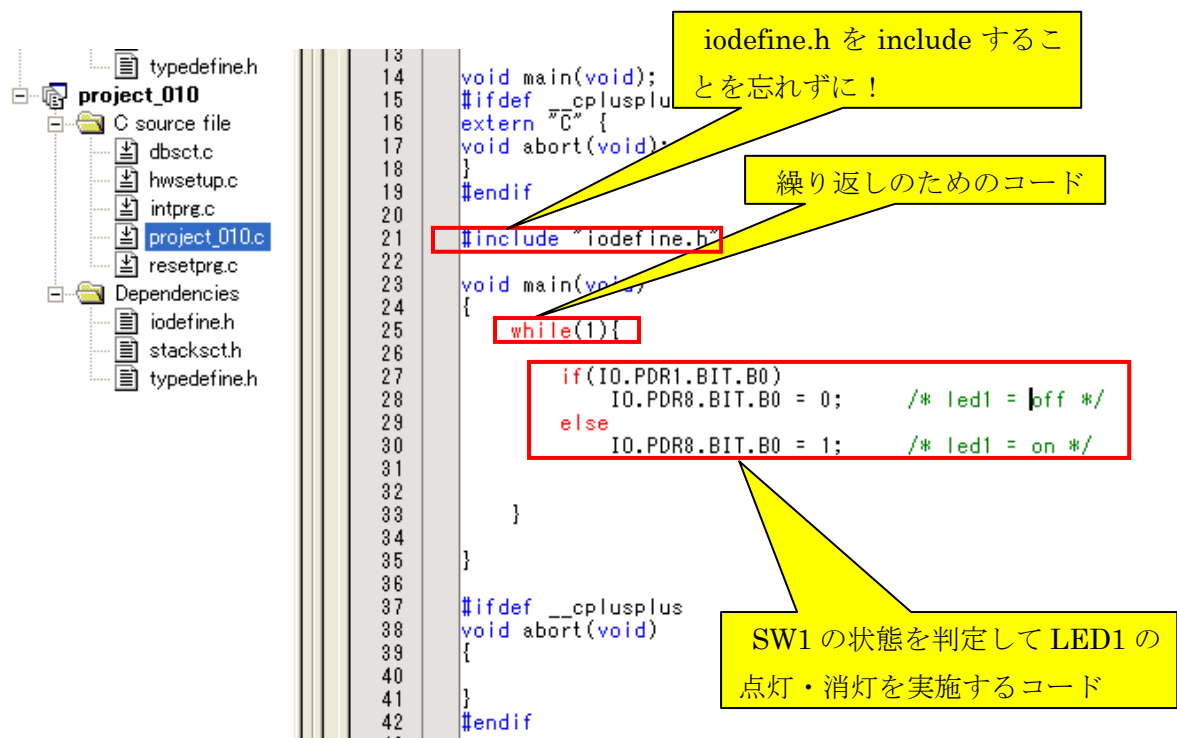
■ 手順

- ① hwsetup.c 内に L E D 1 の P80 を出力ポートに設定するコードを書きます。
- ② hwsetup.c 内に S W 1 の P10 を入力ポートに設定するコードを書きます。

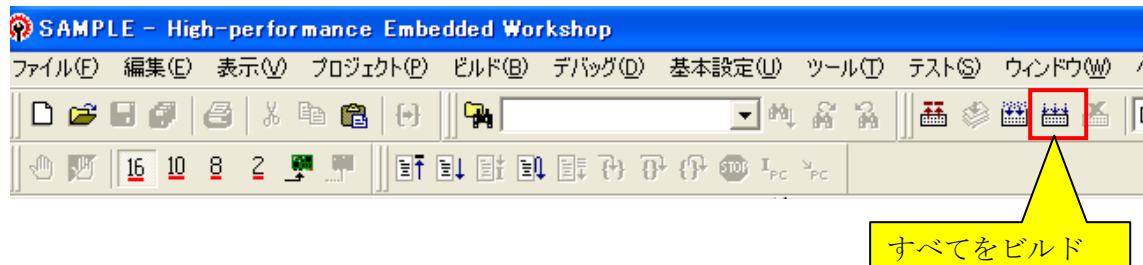


```
15 #include "iodefine.h"
16 #ifdef __cplusplus
17 extern "C" {
18 #endif
19 extern void HardwareSetup(void);
20 #ifdef __cplusplus
21 }
22 #endif
23
24
25
26 void HardwareSetup(void)
27 {
28     /*
29     LVD.CR.BYTE = 0;
30     LVD.CR.BIT.LVDE = 1;
31     LVD.CR.BIT.LVDSEL = 1;
32     LVD.CR.BIT.LVDRE = 1;
33     LVD.CR.BIT.LVDDE = 1;
34     LVD.CR.BIT.LVDUE = 1;
35     LVD.SR.BYTE = 0;
36     LVD.SR.BIT.LVDDF = 1;
37     LVD.SR.BIT.LVDUF = 1;
38     */
39
40     IO.PCR1 = 0x00; /* sw1(p10) = input */
41     IO.PCR8 = 0x01; /* led1(p80) = output */
42
43
44 }
45
```

- ③ project_010.c 内の関数 main()内に S W 1 が O N (押す) なら L E D 1 点灯、S W 1 が O F F (離す) なら L E D 1 消灯するように動作するプログラムコードを書きます。これらのコードが繰り返し実行されるように繰り返し文で構成します。

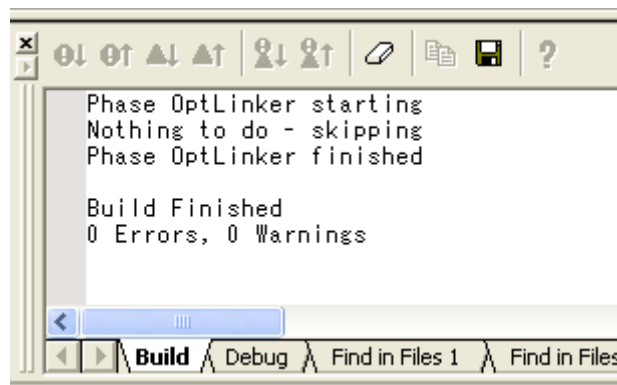


④「全てビルド」を実行します。



⑤正常に終了するとアウトプットウィンドウに下記メッセージが表示されます。

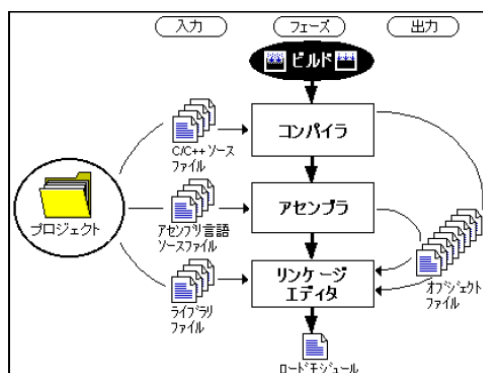
⑥エラーが発生したらエラー発生要因（コーディングミス等）を取り除いてから、再度④に戻って実行します。



◆ 参考：ビルドについて

ビルド処理の一般的な流れを以下の図に示します。ビルドの各フェーズにおいて、1セットのプロジェクトファイルについてビルド処理を行います。それが完了すると、次のフェーズに移ります。

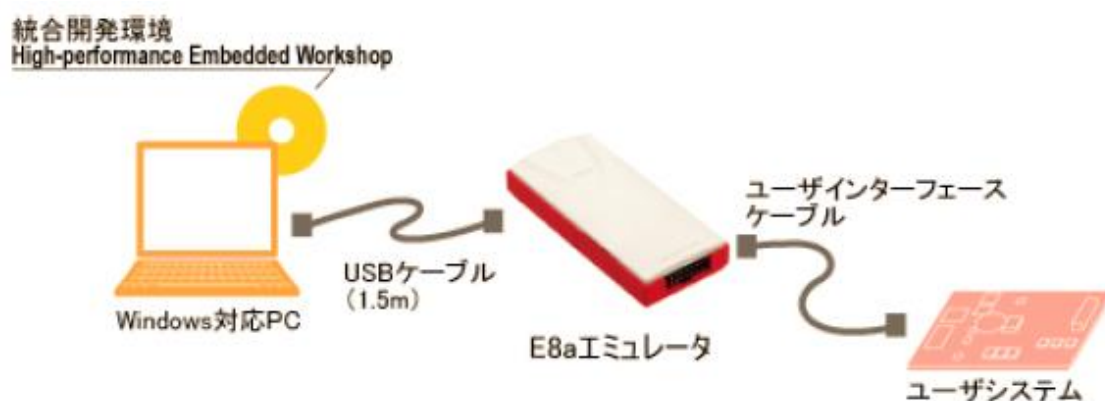
下記の例では、第一のフェーズがコンパイラ、第二のフェーズがアセンブラ、そして最後のフェーズがリンケージエディタです。コンパイラのフェーズでは、プロジェクトのC/C++ソースファイルを順次コンパイルします。アセンブラのフェーズでは、アセンブリ言語のソースファイルを順次アセンブルします。リンケージエディタのフェーズでは、すべてのライブラリファイルと、コンパイラフェーズとアセンブラフェーズからの出力ファイルをリンクして、ロードモジュールを作成します。



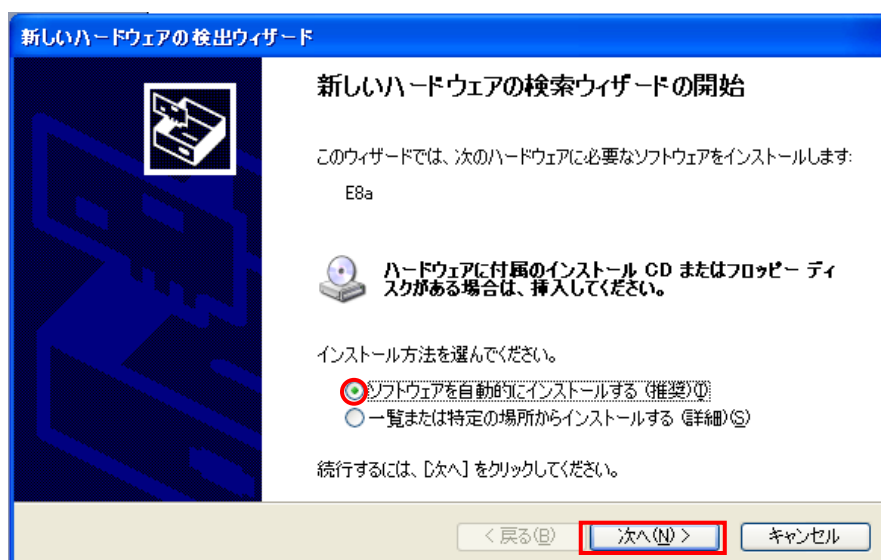
2. 7 E8a を使用したデバッグの準備

① ケーブル接続

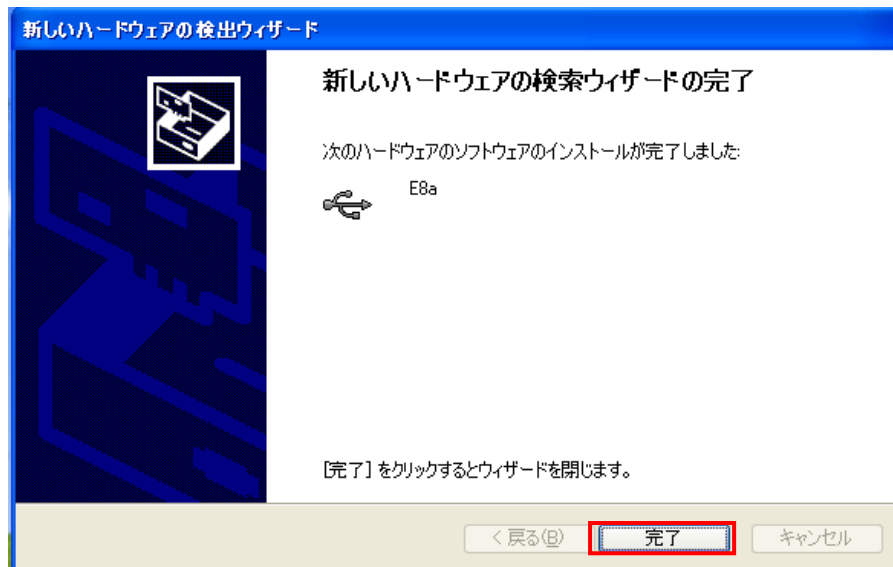
E8a 本体をパソコンとマイコンボードに接続します。パソコンと E 8 a 本体間は付属の U S B ケーブルで、E 8 a 本体とマイコンボード間は付属の 1 4 ピンユーザーインターフェースケーブルで接続します。



パソコンと E 8 a 本体を U S B で接続したとき、E 8 a ドライバをインストールするために下記の画面が表示されることがあります。画面の手順とおり E 8 a 付属の C D - R O M を挿入して「次へ (N)」をクリックします。

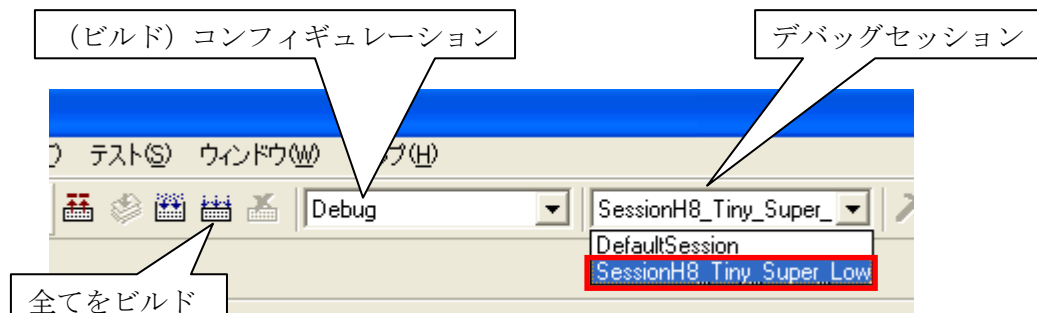


ドライバのインストールが終了すると下記画面が表示されます。



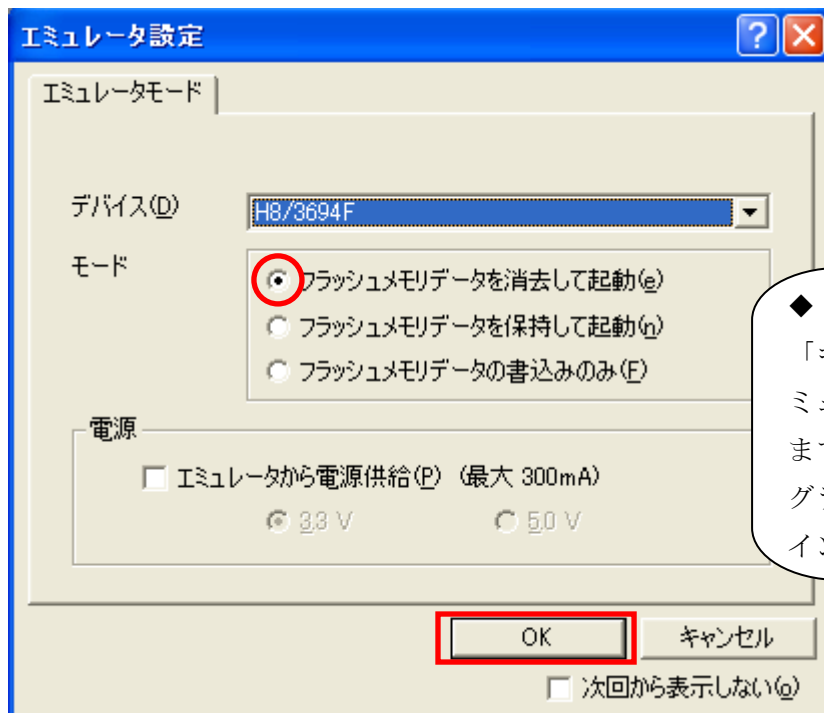
② エミュレータとの接続

デバッグセッションを「Debug」から「SessionH8_Tiny_Super_Low_Power_E8」に切り替えると「エミュレータ設定」画面が表示されます



デバッグセッションの切り替え

「エミュレータ設定」画面ではデバイスが「H8/3694F」であることを確認のうえ下記のモードを選択し「OK」をクリックします。



◆ 参考

「キャンセル」を選択した場合、エミュレータは「disconnect」状態のままで、デバッグツールバー、デバッグランツールバーが表示されないメインウィンドウが表示されます。

◆ 参考

エミュレータ起動モードの説明

【フラッシュメモリデータを消去して起動】

本モードは、ターゲットデバイスのフラッシュメモリ内にE 8エミュレータ用プログラムが存在しない場合に使用します。E 8 aをはじめて起動するときや、E 8 aエミュレータソフトウェアバージョンアップの際、またIDコードを変更したい場合、こちらを選んでください。

【フラッシュメモリデータを保持して起動】

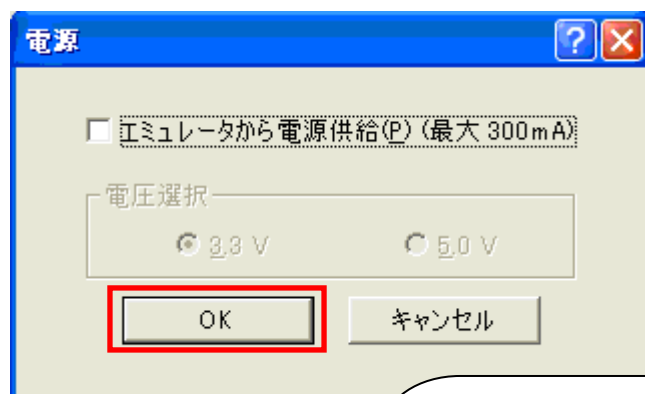
本モードは、ターゲットデバイスのフラッシュメモリ内にE 8エミュレータ用プログラムが存在している場合に使用します。入力するIDコードは上記で設定したコードを入力してください。間違ったコードを入力すると、フラッシュメモリ上のプログラムを全て消去します。

【フラッシュメモリデータの手書きのみ】

本モードは、フラッシュメモリのライターとしてE 8 aエミュレータを使用します。プログラムのデバッグはできません。

ダウンロードするロードモジュールをワークスペースに登録し、ダウンロードしてください。

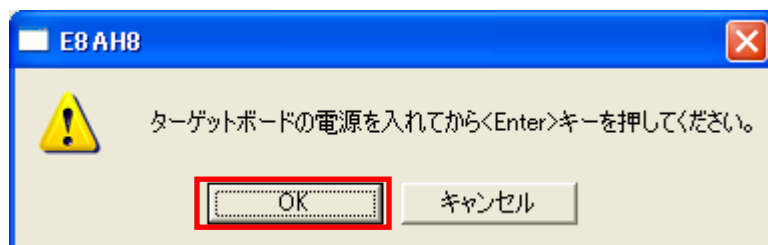
「電源」画面が表示されます。ACアダプタから電源を供給しますので、チェックをいれないまま「OK」を選択します。



◆ 参考

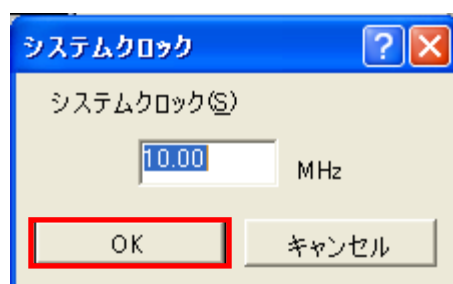
マイコンボードの電源供給をACアダプタではなくエミュレータ（パソコンのUSB）から供給する場合は、チェックを入れます。

「E8AH8」画面が表示されます。

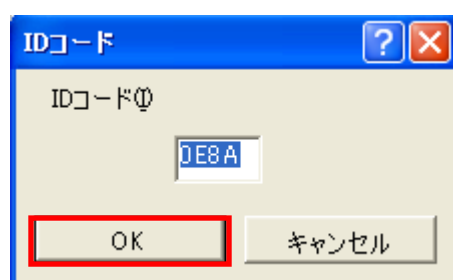


マイコンボードのDCジャック（J11）にACアダプタが挿入されていることを確認のうえ、MCU回路のパワースイッチ（SW11）、基本I/O回路のパワースイッチ（SW12）をONにしてPower LEDが点灯したことを確認してから「OK」をクリックします。

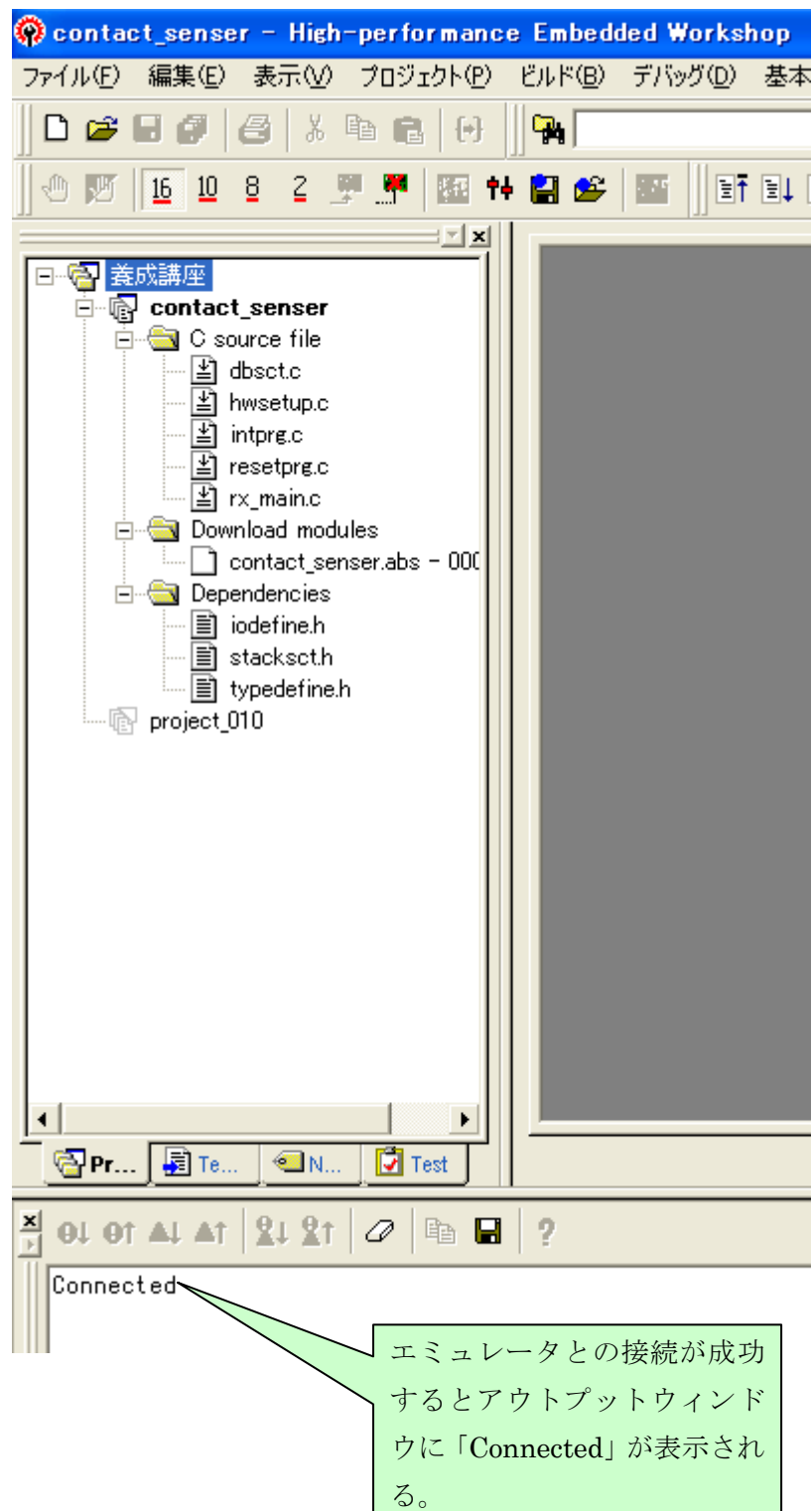
「システムクロック」画面で「OK」をクリックします。



「IDコード」画面で「OK」をクリックします。

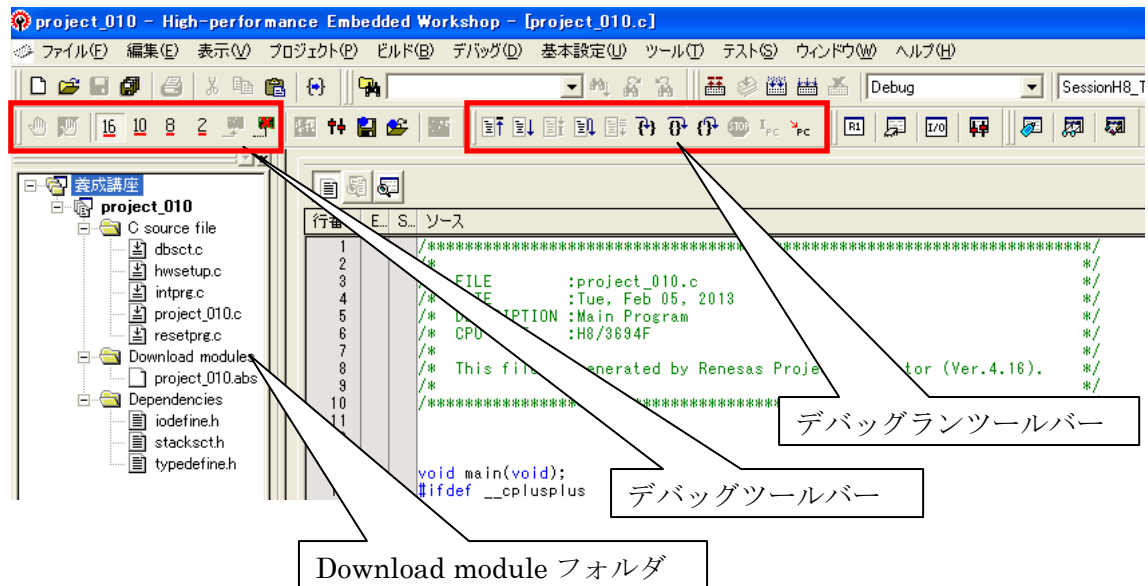


マイコンボードと E 8 a の接続が成功するとアウトプットウィンドウに「Connected」が表示されます。



③ 接続完了時のメインウィンドウ

マイコンボードと E 8 a の接続が完了するとアウトプットウィンドウに「connect」の表示とともに下記のようにターゲットと接続完了したときにのみ表示されるデバッグツールバーやデバッグランツールバーと「Download module」フォルダが見えるようになります。

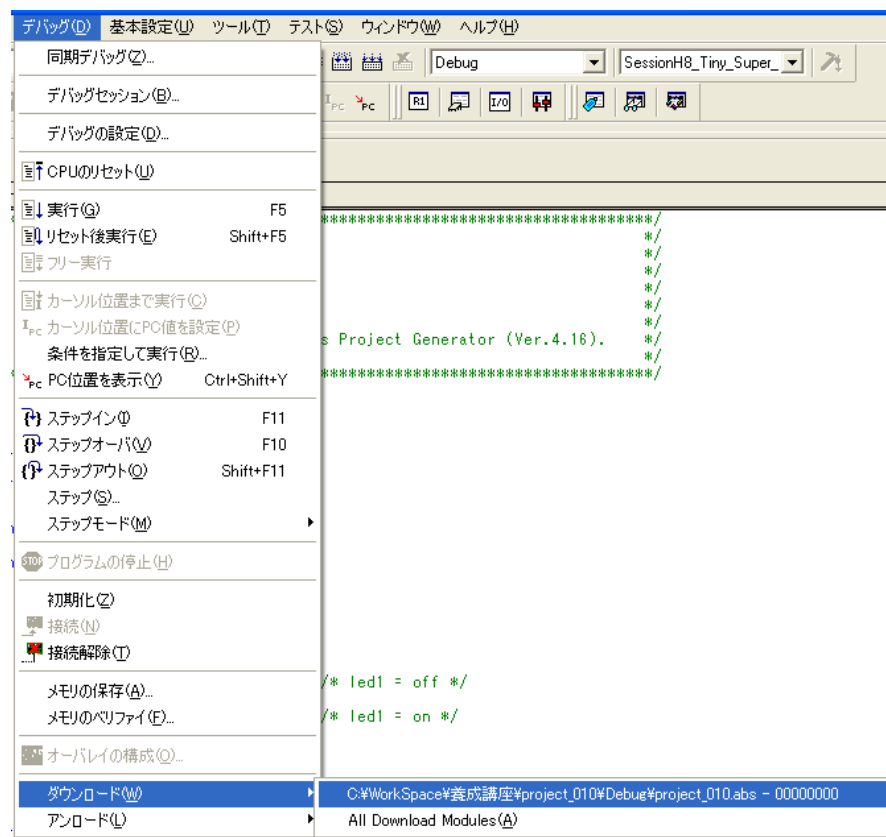


主なツールバーの説明



④ ダウンロード

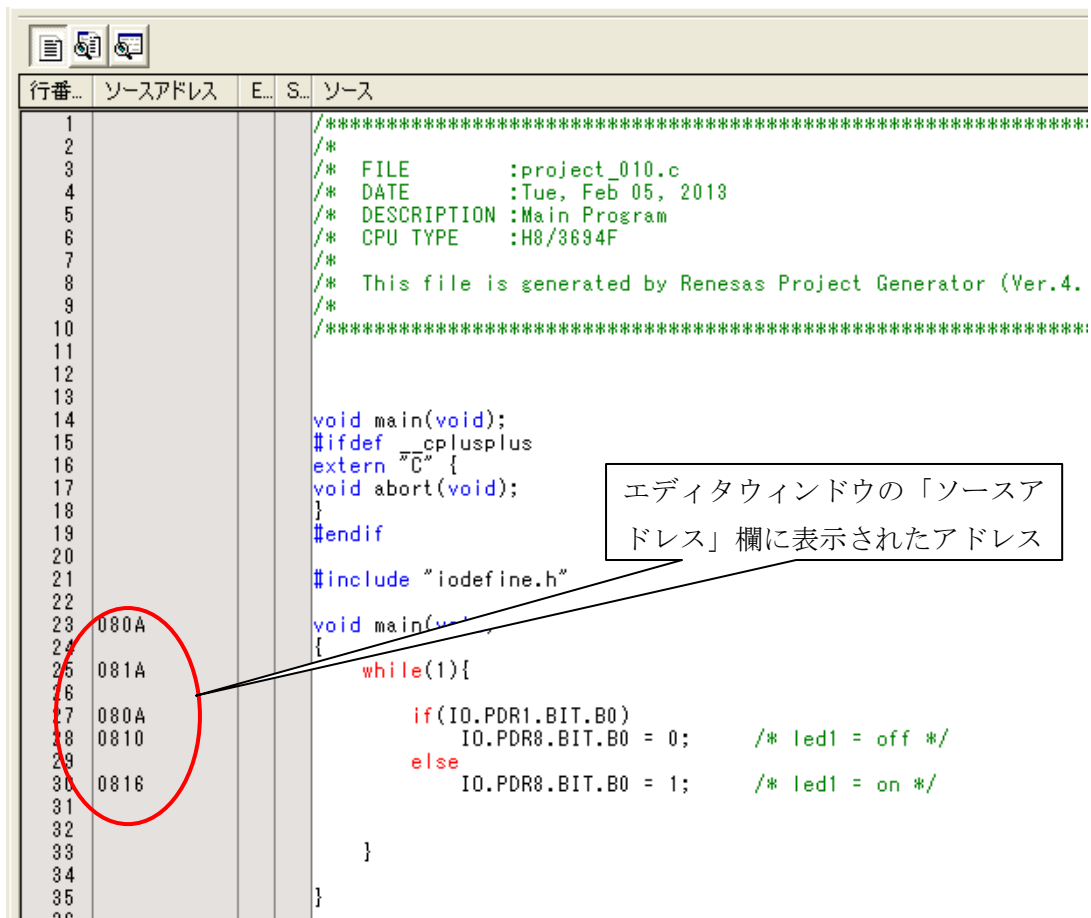
ツールバーの「デバッグ」から下記のようにダウンロード対象モジュール名 (project_010.abs) をクリックします。ロードモジュールがMCUへダウンロードされます。



ダウンロードが完了するとエディタウィンドウの「ソースアドレス」欄にアドレスが見えます。

◆ 参考

実際のマイコン内蔵フラッシュメモリにプログラムが書き込まれるのはユーザープログラム実行直前です。



2. 8 プログラムの実行

① CPUの初期化

デバッグランツールバーの「CPUリセット」をクリックすると、プログラムカウンタ（PC）は初期化されます。すなわちプログラムカウンタは実行するプログラムの先頭アドレスを示します。エディタウィンドウに黄色のバーが表示されますが、この黄色のバーが示すアドレスの命令コードが次回実行される命令コードになります。

② プログラムの実行

デバッグランツールバーの「実行」をクリックすると、プログラムは実行されます。

③ プログラム停止

プログラム実行中にだけ表示されるデバグランツールバーの赤い「STOP」をクリックするとプログラムは停止します。プログラムが停止すると、停止した命令コードの次回実行アドレスにある命令コードを黄色のバーが表示します。

④ ブレーク設定

ブレーク設定にはソフトウェアブレークとハードウェアブレークがあります。

■ ソフトウェアブレーク

ソフトウェアブレークは、ブレークしたい番地の命令を停止命令に書き換えることで実現します。当該番地がRAM上にある場合は問題ありませんが、フラッシュメモリ上にある場合はフラッシュメモリ書き換え機能を使ってフラッシュメモリ上の命令を停止命令に書き換えます。

フラッシュメモリには最大書き込み回数という制限があります。したがって、頻繁にソフトウェアブレークを使用するのはマイコンの信頼性という観点からあまりお勧めはできません。

■ ハードウェアブレイク

ハードウェアブレークはマイコン内蔵の H-UDI（ヒューマン・ユーザーデバッグインタフェース：内蔵デバッグ機能）を使用します。H-UDI はブレーク設定した命令の番地を常に監視しており、設定番地とメモリアクセスアドレスが一致するとプログラムを停止させます。

| 行番 | ソースアドレス | E... | S/Wブレ... | ソース |
|----|---------|------|----------|--|
| 15 | | | | <code>#ifndef __cplusplus</code> |
| 16 | | | | <code>extern "C" {</code> |
| 17 | | | | <code>void abort(void);</code> |
| 18 | | | | <code>}</code> |
| 19 | | | | <code>#endif</code> |
| 20 | | | | |
| 21 | | | | <code>#include "iodefine.h"</code> |
| 22 | | | | |
| 23 | | | | <code>void main(void)</code> |
| 24 | | | | <code>{</code> |
| 25 | | | | <code>while(1){</code> |
| 26 | | | | <code>if(IO.PDR1.BIT.B0 == 0)</code> |
| 27 | 081A | | | <code>IO.PDR8.BIT.B0 = 0; /* led1 = 点灯 */</code> |
| 28 | 080A | | | <code>else</code> |
| 29 | 0810 | | | <code>IO.PDR8.BIT.B0 = 1; /* led1 = 消灯 */</code> |
| 30 | | | | |
| 31 | 0818 | | | |
| 32 | | | | |

ハードウェアブレイク
本欄をダブルクリックするとブレイクポイントの設定／解除ができます。

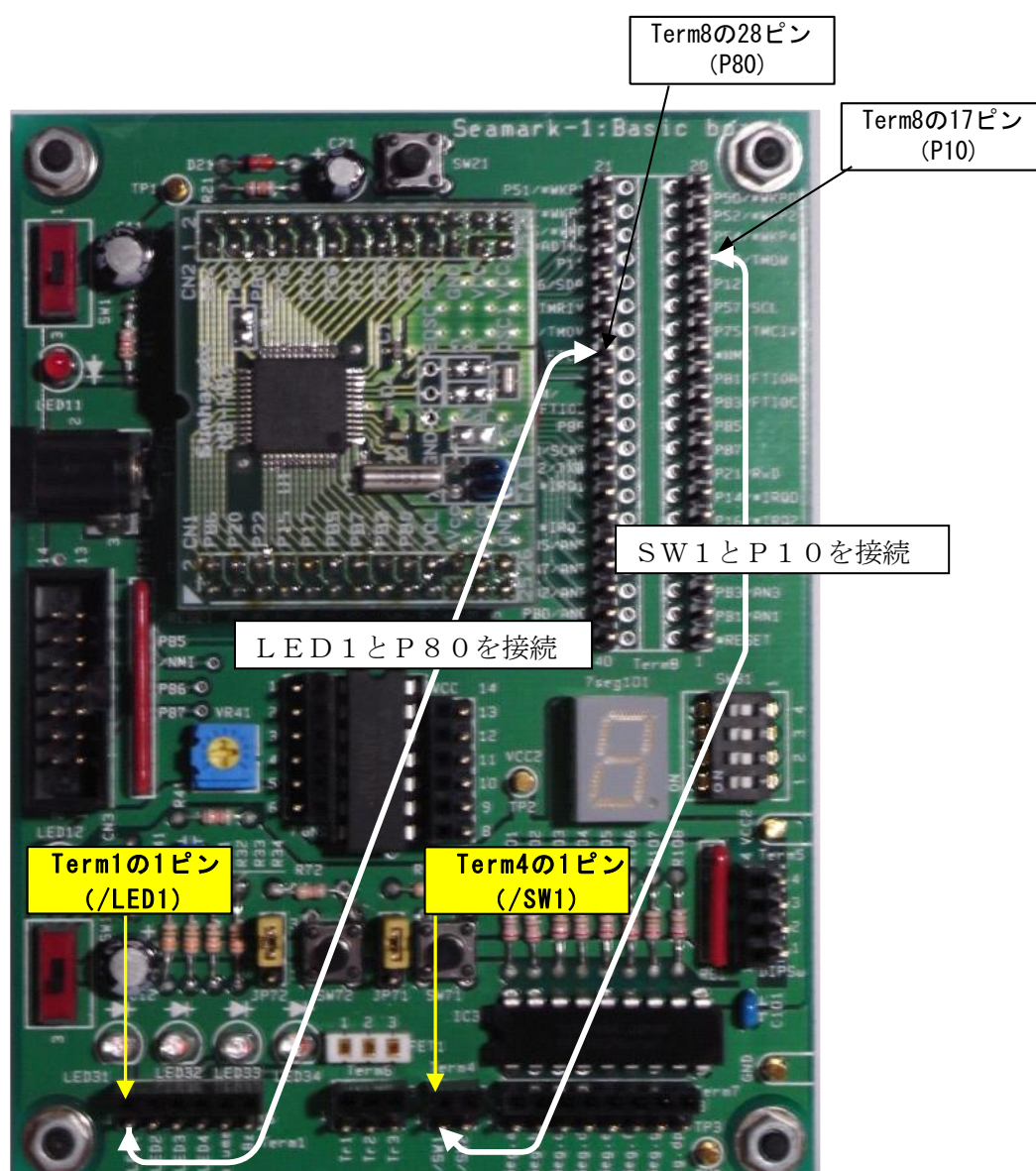
ソフトウェアブレイク
本欄をダブルクリックするとブレイクポイントの設定／解除ができます。

【実習3】実際にロードモジュールをダウンロードしてプログラムを実行し、プログラム動作を確認しましょう。

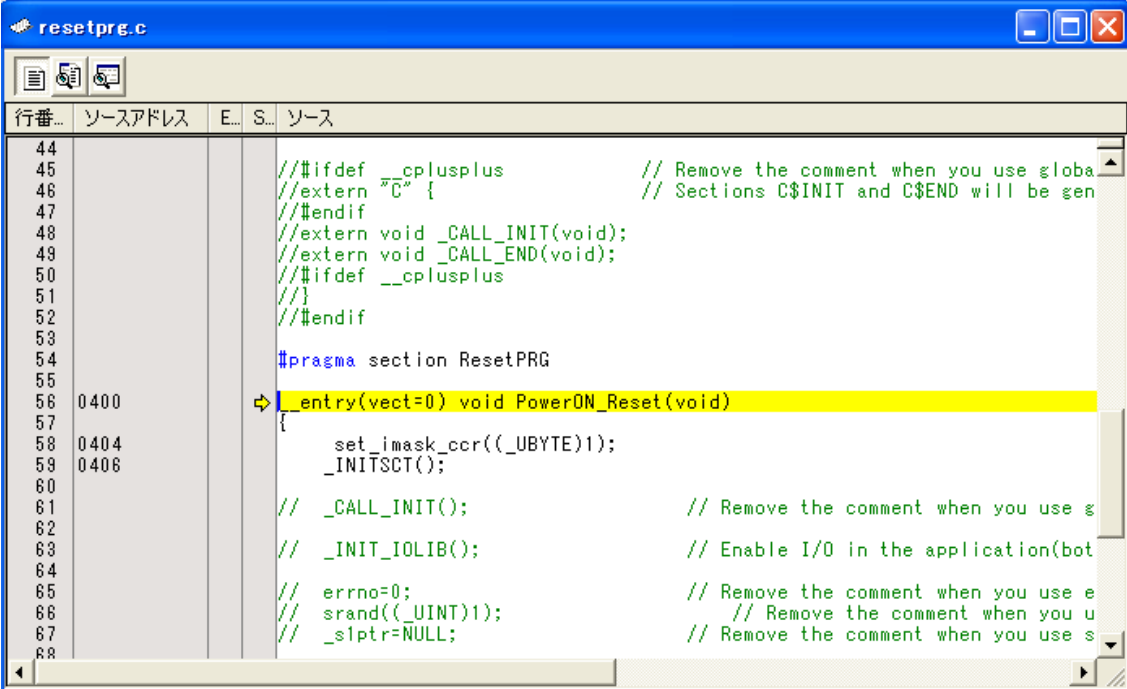
実習2で作成したロードモジュール「project_010.abs」をダウンロードしてプログラムを実行します。

■ 手順

- ① マイコンボード上で回路を作る。



- ② 「project_010.abs」をダウンロードします。⇒ 「8.4 ④ダウンロード」参照
- ③ エディタウィンドウの「ソースアドレス」欄にアドレスが表示されていることを確認します。
- ④ [標準]ツールバーの「CPUリセット」をクリックします。0x0400番地のソースコードに黄色のバーが表示されます。



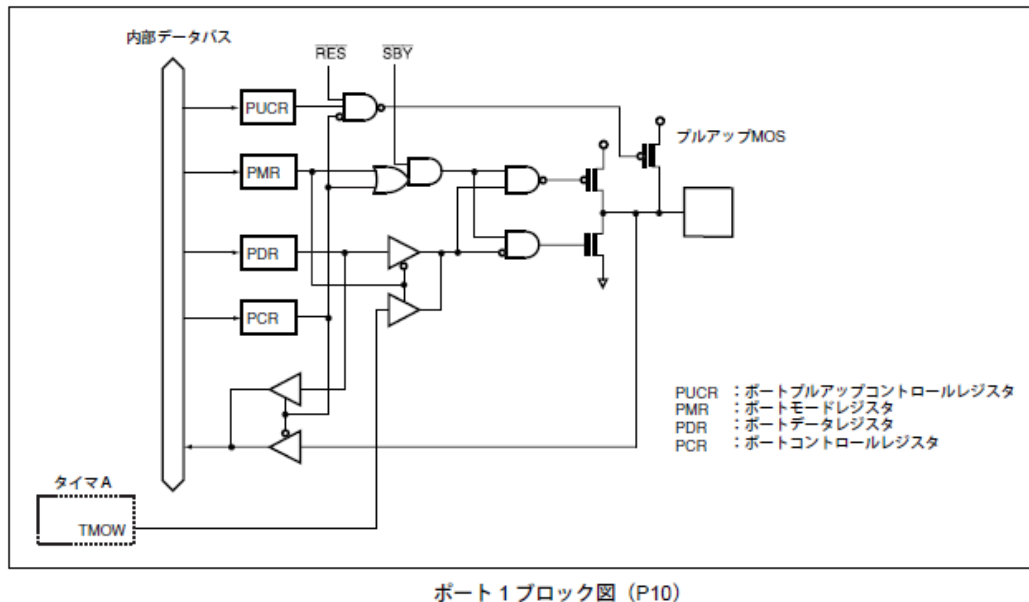
The screenshot shows a window titled 'resetprg.c' with a standard Windows-style title bar. Below the title bar is a toolbar with icons for file operations. The main area is a table with four columns: '行番...' (Line Number), 'ソースアドレス' (Source Address), 'E...' (Error), and 'S...' (Status). The 'ソース' (Source) column contains C code. Line 56 is highlighted in yellow, and a mouse cursor points to the start of the line. The code includes preprocessor directives, extern declarations, and a function definition for 'PowerON_Reset'.

| 行番... | ソースアドレス | E... | S... | ソース |
|-------|---------|------|------|---|
| 44 | | | | |
| 45 | | | | //#ifdef __cplusplus // Remove the comment when you use globa |
| 46 | | | | //extern "C" { // Sections C\$INIT and C\$END will be gen |
| 47 | | | | //#endif |
| 48 | | | | //extern void _CALL_INIT(void); |
| 49 | | | | //extern void _CALL_END(void); |
| 50 | | | | //#ifdef __cplusplus |
| 51 | | | | // |
| 52 | | | | //#endif |
| 53 | | | | |
| 54 | | | | #pragma section ResetPRG |
| 55 | | | | |
| 56 | 0400 | | | _entry(vect=0) void PowerON_Reset(void) |
| 57 | | | | { |
| 58 | 0404 | | | set_imask_ccr((_UBYTE)1); |
| 59 | 0406 | | | _INIT_SCT(); |
| 60 | | | | |
| 61 | | | | // _CALL_INIT(); // Remove the comment when you use g |
| 62 | | | | |
| 63 | | | | // _INIT_IOLIB(); // Enable I/O in the application(bot |
| 64 | | | | |
| 65 | | | | // errno=0; // Remove the comment when you use e |
| 66 | | | | // srand((_UINT)1); // Remove the comment when you u |
| 67 | | | | // _slptr=NULL; // Remove the comment when you use s |
| 68 | | | | |

- ⑤ [標準ツールバー]の「実行」をクリックします。プログラムが実行されます。RUN状態になります。[標準ツールバー]の「STOP」が赤で表示されます。
- ⑥ RUN状態でSW1の操作に対するLED1の点灯／消灯状態を確認します。
 - ・SW1を押したとき、LED1は点灯しますか？
 - ・SW1を離したとき、LED1は消灯しますか？
 - ・リセットスタート後のLED1は消灯していますか？

◆ 参考

マイコンポートの入出力ポート P10 のブロック図です。



ブロック図より下記のことわかります。

- ① プルアップ抵抗を内蔵している
- ② 出力ポートはCMOS構造になっている
- ③ 入力ポートのときは、MCU端子レベルを直接読み込んでいる

【課題2】CPUリセット後の「実行」でLED2が点灯、SW2を押すとLED2は消灯、SW2を離すとLED2は点灯プログラムを作成してデバッグして動作を確認しよう。

3

シリアルインターフェース



3. 1 概要

シリアル通信とは、一般的にパソコンとモデムなどの機器を接続して通信を行うために考えられた手段です。データを1ビットごとに送受信し、8ビット分のデータがそろった時点で1バイトのデータとして扱います。シリアル通信は、古くから汎用インターフェースとして使用されてきました。たとえば、モデム・インターフェースとしての規格として考えられたRS-232Cはパソコンなどに標準で装備されており、外部への通信手段として使用することができます。また、組込みシステムの開発では、モデムなど外部との通信インターフェースとしての用途だけでなく、図1のようにパソコンと基板を接続しシリアル通信を利用しデバッグを行うことがあります。

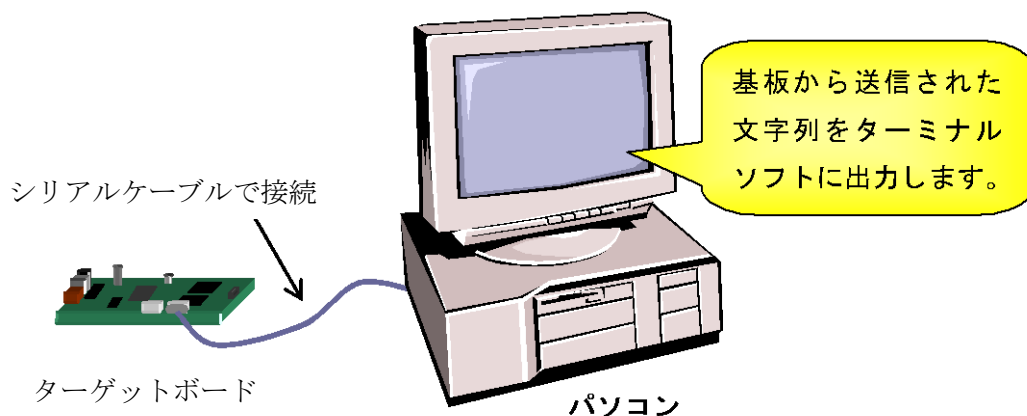


図 1 基板と PC 間のシリアル通信

近年ではシリアル通信の高速化が進み、イーサネット、IEEE1394 や USB などの高速通信が主流となってきています。表 1 にいくつかのシリアル通信規格の用途をまとめます。

表 1 シリアル通信規格の種類と用途

| 規格 | 用途 |
|-----------------------|---|
| RS-232C | モデム・インタフェースの標準規格であり、シリアルポートにモデムを接続して電話回線経由でのデータ通信を行う。 |
| イーサネット (IEEE802.3) | 100Mbps、1Gbps の通信速度をもち主に TCP/IP ネットワークの LAN 用プロトコルとして使用される。 |
| IEEE1394 | 100M、200M、400Mbps の転送速度が規定化されており、映像や音声などの大量のデータ転送に使用される。 |
| USB | 1.5Mbps、12Mbps の通信速度をもち、主にマウスやキーボードなどのパソコンの周辺機器に使用される。 |

また、シリアル通信のほか、パラレル通信といわれるものもあります。これは、一度に8本の配線を使用して1バイトのデータを送る手段です（図 2）。パラレル通信のほうがシリアル通信よりも速くデータを送信することができます。しかし、データ通信は一般的に長距離のものが多く、配線コストがかからないものを選択することがしばしばあります。そこで、配線本数が多く、ケーブルそのものも太くなってしまうパラレルより、受信・送信・グラウンドの3本の配線で通信のできるシリアル通信が選択されるのです。

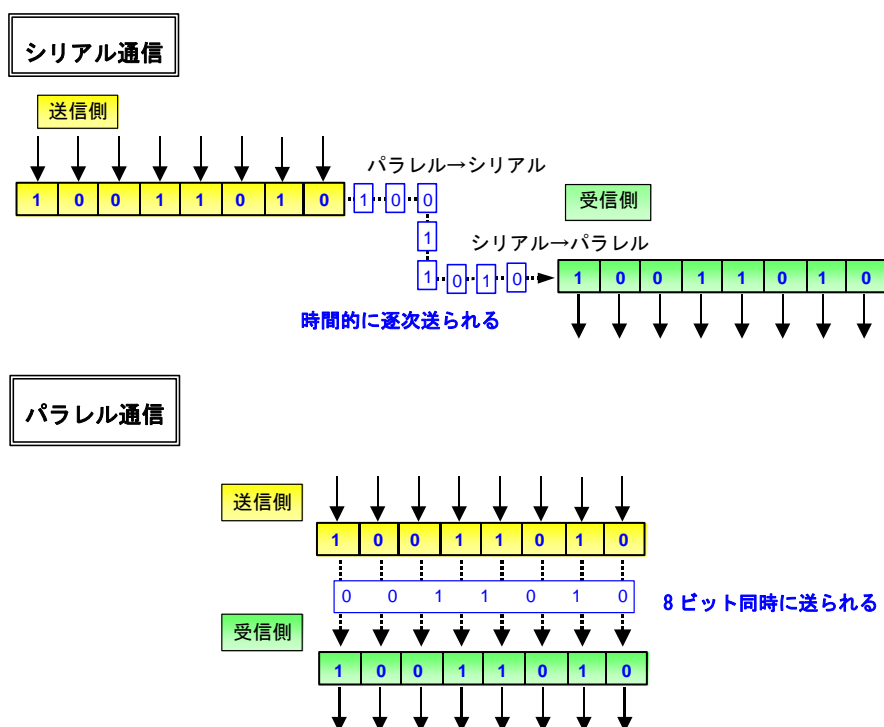


図 2 シリアル通信とパラレル通信

3.2 同期通信と非同期通信

シリアル通信は、送信側と受信側で同期を取りながらデータの送受信を行います。

では、なぜ通信をするために同期が必要なのでしょう。身近な例で考えてみましょう。

私たちがキャッチボールをするとき、突然ボールを投げると正しくキャッチすることができないので、相手に何か合図をします。合図を出すことによりキャッチボールが成立します。シリアル通信においても、キャッチボールのように合図を出すことにより、通信を成立させることができます（図 1）。この合図を出し合うことを「同期をとる」といいます。

データを送信する側は、データを一定間隔で送信することができたとしても、受信側にはどのタイミングでデータの送信が開始され終了するのかわかりません。そのためにシリアル通信には送受信の制御をするための基本的な手段が用意されています。

代表的な手段として、同期通信と非同期通信の 2 つの通信手段があります。この 2 つの通信手段については次項で詳述します。

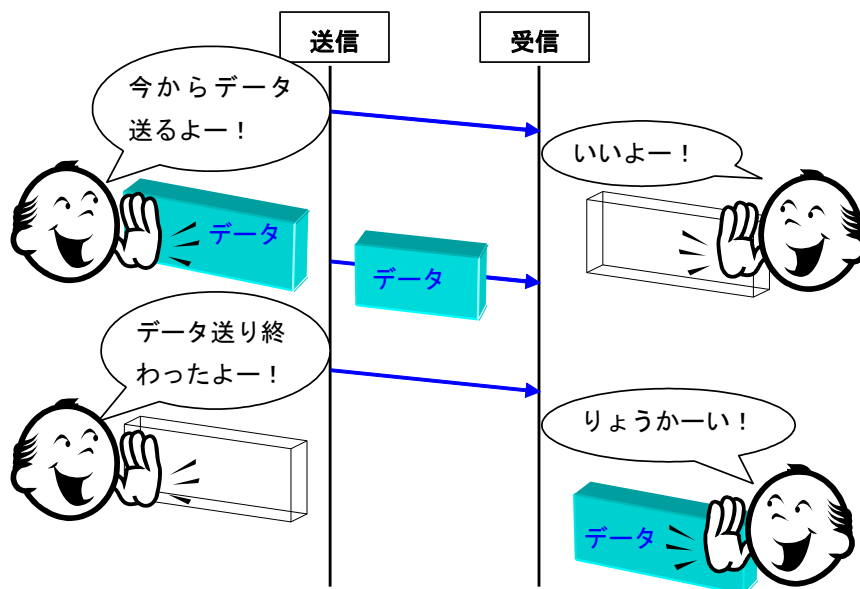


図 1 シリアル通信の同期

3.2.1 同期通信

同期通信は、データ線と制御信号線の配線をもっており、データを送っていないときも一定の速度で通信を行うために制御信号を送り続ける通信手段です。この制御信号のことをクロックといいます。同期通信では、データをこのクロックに合わせて送信します（図 2）。

同期通信は、同期をクロックで行います。クロックの立ち上がり、立下りタイミングでデ

ータの送信・受信をおこないます。全二重通信を想定した場合、信号線はデータ線 2 本とクロック線 1 本の合計 3 本が必要です。1 バイトデータの送信・受信には連続した 8 個のクロックで行います。クロックに同期するという意味で**クロック同期式通信**と呼ばれます。

特徴

- ・ 3 線（送信線、受信線、クロック線）
- ・ 調歩同期式通信方式に比べて、スタートビット、ストップビットがない分だけ伝送効率がよい。

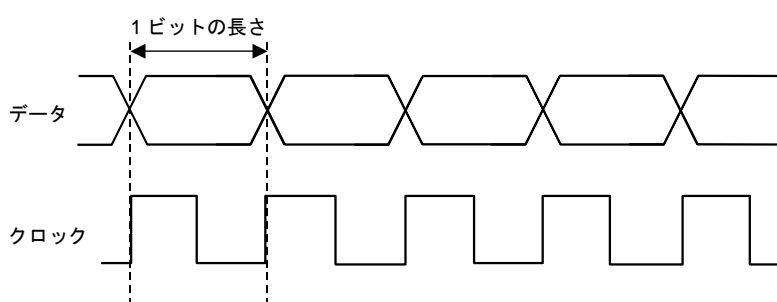


図 2 同期通信のタイムチャート

3.2.2 非同期通信

非同期通信は、制御信号線がなくデータ線だけの通信となります。しかし、データを送受信するためには、同期をとるための信号が必須です。そのため、非同期通信では 1 バイトのデータを送信するために最初と最後に合図となるデータを付加し、制御信号としています（図 3）。

非同期通信では、データを送信していないとき、常にデータ線を「1」の状態を保ちます。これをアイドル状態といいます。送信側はデータの送信をはじめるとき、まずデータ線を「0」にします。これをスタートビットといいます。受信側はこのデータを受け、その後のデータの通信スピードを決定します。送信側は 1 バイト分のデータを送信が終わるとデータ線を「1」の状態にします。これをストップビットといいます。ストップビットは 1 ビット以上の長さのデータを送ります。

調歩式（非同期式）は「歩みを調える」方式ですから、同期対象をクロックに求めずそれぞれのデータの直前に 1 ビット分のスペース（スタートビット）を付加して、このスペースで通信開始タイミング計り、事前に決めた通信速度でスペースに続くデータをサンプリングします。通信速度で決定される 1 ビット長を厳守する必要があります。クロックに同期しないという意味で非同期式通信また**調歩同期式通信**（Universal Asynchronous Receiver/Transmitter (UART)）と呼ばれます。

特徴

- ・ 2 線式（送信線、受信線）
- ・ クロック線が不要

同期式に比べると、スタート、ストップのタイミングビット分だけ伝送効率は悪い。

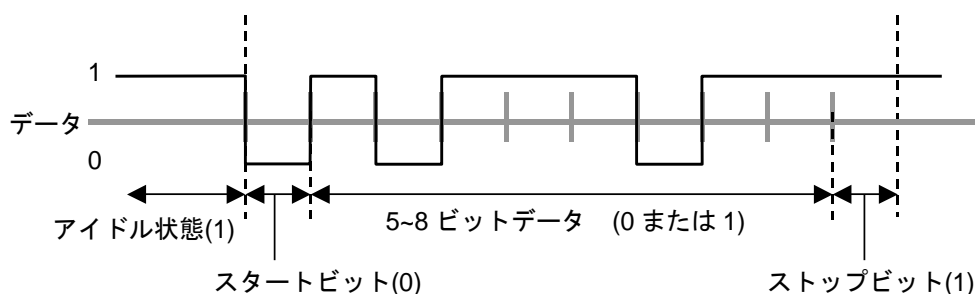


図 3 非同期通信のタイムチャート

3.2.3 シリアル通信の初期化

パソコン上でシリアル通信をするためにはさまざまな設定を行います。組込みシステムでは、電源投入時またはシリアル通信を使用する前にシリアル通信の初期化をしなければなりません。表 2 は、非同期通信の設定仕様例です。

表 2 非同期通信の設定値例

| 設定項目 | 設定内容 |
|-----------|----------|
| ① ビットレート | 38400bps |
| ② データ長 | 8 ビット |
| ③ パリティビット | なし |
| ④ ストップビット | 1 ビット |
| ⑤ フロー制御 | なし |

- ① ビットレートは、一秒間あたりに送るデータの量をビット数で示したものです。単位は bps (bit per second) を使用します。
- ② データ長とは、スタートビットとストップビットを省いた実際のデータ長のことを言い、「5 ビット～8 ビット」のデータ長を選択することができます。大抵のシステムでは 7 ビットもしくは 8 ビットを選択します。
- ③ パリティとは、送信データ中にデータの喪失がないかをチェックするためのものです。偶数パリティ、奇数パリティ、なしの選択ができます。パリティチェックは送

信したデータ内に 1 のデータがいくつあるかを数え、送信側と受信側でデータの比較をする方法です。

- ④ 1 バイトのデータ送信の完了を示すために付加するビットの数を指定します。1 ビットまたは 2 ビットを選択します。
- ⑤ フロー制御とは、受信側の処理が間に合わずデータを取りこぼしてしまいそうとき、送信側のデータを送信する速度を遅くしたり、止めたりする機能です。フロー制御には「XON/XOFF」、「ハードウェア制御」があり、XON/XOFF を選択した場合は、データの送信停止要求及び再送要求ができます。ハードウェア制御を選択した場合は、データ線とは別に制御線を用意してフロー制御を行います。

このような設定を行うことで非同期通信を使用することができます。

注意：最近では L S B から送信する方式（L S B ファースト）が主流のようですが、M S B から送信する方式（M S B ファースト）もあるので、データ送信方向（L S B ファースト／M S B ファースト）の確認も必要です。

◆参考

調歩同期式通信と RS-232-C は同義と捕らえがちですが、RS-232-C は O S I 参照モデルの物理層にあたる部分の仕様です。コネクタの形状、Low (0) / High (1) の電気的表現方法、信号の論理的特性などが決まっています。

現在では多くのパソコンで標準の入出力インターフェースとして RS-232-C を採用しており、さらに通信方式として調歩同期式通信を使っているため、「調歩同期式通信 = RS-232-C」と誤解されているようです。

4

距離データの応用



4. 1 概要

FPGAボードで計測した距離データをマイコン側で受信して、当該距離に応じた警報を発するシステムを構築します。

■ システムブロック図

図1にシステムブロック図を示します。FPGAボードとマイコンボードの通信距離が長い場合はFPGA基板のRS-232-Cコネクタ（J3）からRS-232-Cケーブルを使用してマイコンボードと通信を行うという方法もありますが、今回は近接したボード同士での通信なので、RS-232-Cケーブルを介さずに、FPGAボードの送信ポート（P70）とマイコンボードのRxD端子（P21）を直接接続して通信を行います。

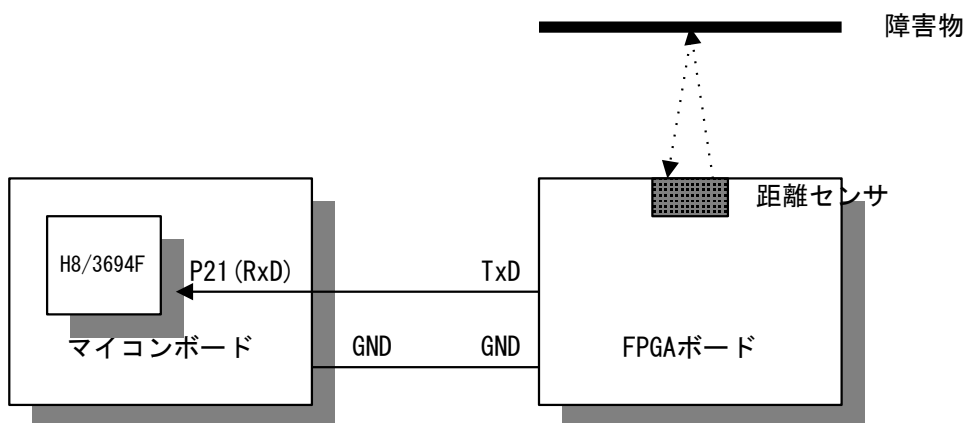


図1 システムブロック図

■ システム仕様

①通信：調歩同期式 UART（FPGA からマイコンへ方向通信）

②マンマシンインタフェース：

距離データに応じて

- ・ 警報（圧電ブザー）を鳴動
- ・ LED を点滅
- ・ 7セグメント LED 1 桁に距離を表示

4. 2 ソフトウェア詳細仕様

■ FPGAボードとマイコンボード間の通信

FPGAとマイコンボード間は図2に示す6バイトのデータを受け渡しを行います。

| BYTE | 1 | 2 | 3 | 4 | 5 | 6 |
|------|-------------|------|---------------|-----|---------|-----------|
| | SOD | 1000 | 100 | 10 | 1 | EOD |
| | StartOfData | mm | mm | mm | mm | EndOfData |
| | 0x42(B) | ... | 0x00~0x09 × 4 | ... | 0x45(E) | |
| | BinaryData | | | | | |
| | 0x44(D) | ... | 0x30~0x39 × 4 | ... | 0x45(E) | |
| | DecimalData | | | | | |

図2 FPGA・マイコンボード間データフレーム

通信の例：42.8cmのとき

Binary Data : 0x42 0x00 0x04 0x02 0x08 0x45

Decimal Data : 0x30 0x30 0x34 0x32 0x38 0x45

■ 通信プロトコル

- ・ Baud Rate : 38400bps
- ・ Data Bits : 8 (LSB ファースト)
- ・ Stop Bit : 1
- ・ Parity Bit : none
- ・ Xon/Xoff : none

■ 制御周期タイミングチャート

設定する t_1 のタイミングチャートを図 3 に示します。

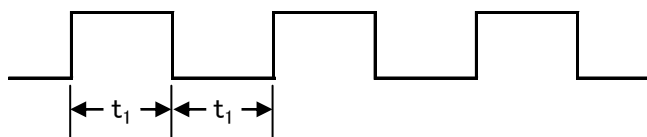


図 3 t_1 のタイミングチャート

t_1 を表 1 に示します。

| 入力値範囲(cm) | 入力値<10 | 10≤入力値<20 | 20≤入力値<30 | 30≤入力値<40 | 40≤入力値<50 | 50≤入力値<60 | 60≤入力値<70 | 70≤入力値<80 | 80≤入力値<90 | 90≤入力値<100 | 100≤入力値 |
|---------------|--------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|------------|---------|
| t_1 設定値(ms) | 0 | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900 | 1000 |

表 1 t_1 設定値

■ 7segment LED

・距離単位（cmとm）を区別するためにセンチメートル表記のときは dp を点灯、メートル表記のときは dp を消灯します。

・入力値と 7seg. LED に表示する数字の関係を表 2 に示します。

| 入力値範囲(cm) | 入力値<10 | 10≤入力値<20 | 20≤入力値<30 | 30≤入力値<40 | 40≤入力値<50 | 50≤入力値<60 | 60≤入力値<70 | 70≤入力値<80 | 80≤入力値<90 | 90≤入力値<100 | 100≤入力値 |
|-------------|--------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|------------|---------|
| 7seg. LED表示 | 0. | 1. | 2. | 3. | 4. | 5. | 6. | 7. | 8. | 9. | 9. |

表 2 入力値と 7seg. LED に表示する数字の関係

■ 圧電ブザー

t_1 の周期で圧電ブザーが鳴動します。

■ LED

t_1 の周期でLEDが点滅します。

どのLEDが点滅するかは、P80 端子を基本 I/O 基板の Term1 の/LED1 から /LED4 のどの接続するかで決まります。

4. 3 マイコンの端子配置

表3にマイコンの端子定義表を示します。

端子定義表

ターゲット：H8/3694F

| Pin No | 記号 | I/O | Reset時の状態 | 端子記号 | 端子名称 | I/O | 機能 | INZ時の状態 | 備考 |
|--------|------------------|-----|-----------|------|----------|-----|-----------------|---------|--------|
| 1 | AVDD | I | H | | | | | | |
| 2 | X2 | O | | | | | | | |
| 3 | X1 | I | | | | | | | |
| 4 | VOL | I | H | | | | | | |
| 5 | /RES | I | L | | /RES | I | リセット | H | |
| 6 | TEST | I | L | | | | | | |
| 7 | Vss | I | L | | | | | | |
| 8 | OSC2 | O | | | | | | | |
| 9 | OSC1 | I | | | | | | | |
| 10 | Vcc | I | H | | | | | | |
| 11 | P50 /WKP0 | I/O | X | P50 | 汎用入出力ポート | O | 7seg. LED「a」対応 | L | |
| 12 | P51 /WKP1 | I/O | X | P51 | 汎用入出力ポート | O | 7seg. LED「b」対応 | L | |
| 13 | P52 /WKP2 | I/O | X | P52 | 汎用入出力ポート | O | 7seg. LED「c」対応 | L | |
| 14 | P53 /WKP3 | I/O | X | P53 | 汎用入出力ポート | O | 7seg. LED「d」対応 | L | |
| 15 | P54 /WKP4 | I/O | X | P54 | 汎用入出力ポート | O | 7seg. LED「e」対応 | L | |
| 16 | P55 /WKP5 /ADTRG | I/O | X | P55 | 汎用入出力ポート | O | 7seg. LED「f」対応 | L | |
| 17 | P10 TMOV | I/O | X | | | | | | |
| 18 | P11 | I/O | X | | | | | | |
| 19 | P12 | I/O | X | | | | | | |
| 20 | P56 SDA | I/O | X | P56 | 汎用入出力ポート | O | 7seg. LED「g」対応 | L | |
| 21 | P57 SCL | I/O | X | P57 | 汎用入出力ポート | O | 7seg. LED「dp」対応 | L | |
| 22 | P74 TMRIV | I/O | X | | | | | | |
| 23 | P75 TMCIV | I/O | X | | | | | | |
| 24 | P76 TMOV | I/O | X | TMOV | タイマV出力端子 | O | ブザー出力制御 | H | |
| 25 | /NMI | I | | | | | | | E8接続使用 |
| 26 | P80 FTCL | I/O | X | P80 | 汎用入出力ポート | O | LED出力制御 | L | |
| 27 | P81 FTIOA | I/O | X | | | | | | |
| 28 | P82 FTIOB | I/O | X | | | | | | |
| 29 | P83 FTIOC | I/O | X | | | | | | |
| 30 | P84 FTIOD | I/O | X | | | | | | |
| 31 | P85 | I/O | X | | | | | | E8接続使用 |
| 32 | P86 | I/O | X | | | | | | E8接続使用 |
| 33 | P87 | I/O | X | | | | | | E8接続使用 |
| 34 | P20 SOK3 | I/O | X | | | | | | |
| 35 | P21 RXD | I/O | X | RxD | | I | uart受信端子 | H | |
| 36 | P22 TXD | I/O | X | | | | | | |
| 37 | P14 /IRQ0 | I/O | X | | | | | | |
| 38 | P15 /IRQ1 | I/O | X | | | | | | |
| 39 | P16 /IRQ2 | I/O | X | | | | | | |
| 40 | P17 /IRQ3 /TRGV | I/O | X | | | | | | |
| 41 | PB4 AN4 | I | X | | | | | | |
| 42 | PB5 AN5 | I | X | | | | | | |
| 43 | PB6 AN6 | I | X | | | | | | |
| 44 | PB7 AN7 | I | X | | | | | | |
| 45 | PB3 AN3 | I | X | | | | | | |
| 46 | PB2 AN2 | I | X | | | | | | |
| 47 | PB1 AN1 | I | X | | | | | | |
| 48 | PB0 AN0 | I | X | | | | | | |

※ H:High L:Low X:ハイインピーダンス

表3 マイコンポートのピンアサイン表

4. 4 システム構成上の問題点

FPGA ボードの TxD 端子の信号レベルは High のとき 3.3 v です。FPGA 側で High を出力してもマイコン側では必ずしも High と認識しません。マイコン側では最低でも 3.5 v ($V_{cc} \times 0.7 = 3.5v$) 以上ないと High レベルと認識しません (表 4 参照)。

| 項目 | 記号 | 適用端子 | 測定条件 | 規格値 | | | 単位 | 備考 |
|------------------|-----------------|---|----------------------------|-----------------------|-----|-----------------------|----|----|
| | | | | Min | Typ | Max | | |
| 入力 High レベル電圧 | V _{IH} | RES、NMI WKPF0~WKPF5 IRQ0~IRQ3 ADTRG TMRIV、TMCIV | V _{cc} =4.0~5.5V | V _{cc} ×0.8 | — | V _{cc} +0.3 | V | |
| | | FTCI、FTIOA FTIOB、FTIOC FTIOD SCK3 TRGV | | V _{cc} ×0.9 | | V _{cc} +0.3 | V | |
| | | RXD、SCL、SDA | V _{cc} =4.0~5.5V | V _{cc} ×0.7 | — | V _{cc} +0.3 | V | |
| | | P10~P12 P14~P17 P20~P22 | | | | | | |
| | | P50~P57 P74~P76 P80~P87 | | V _{cc} ×0.8 | — | V _{cc} +0.3 | V | |
| | | PB0~PB7 | AV _{cc} =4.0~5.5V | AV _{cc} ×0.7 | — | AV _{cc} +0.3 | V | |

表 4 H8/3694F の D C 特性

図 4 からマイコンボード側では、1.0v から 3.5v 間は「不定」であることがわかります。FPGA の送信データの 3.3v (High レベル) を何とかして 5v 程度までレベル変換する必要があります。

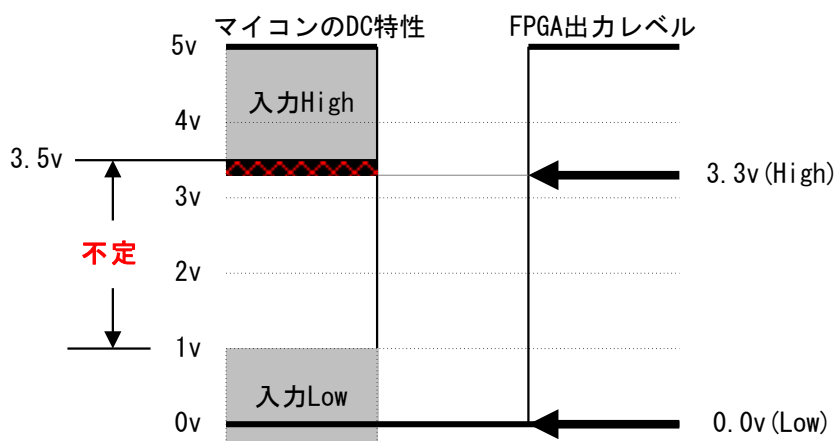


図 4 F P G A の出力信号レベルとマイコンボードの入力信号レベル

■ 対策

FPGA ボードの TxD 端子の 3.3 v を 5.0 v 信号に変換するために、電圧レベル変換回路を介してデータの送受信を行います。



図 5 電圧レベル変換

今回の電圧レベル変換回路には、汎用ロジック IC 「74HC07」を使用します。FPGA ボードの TxD 端子には 0/1 のデータとして 0.0v/3.3v の電圧レベルの信号が出力されます。

74HC07 の出力は、図 10.6 の Truth Table (真理値表) および System Diagram から Nch オープンドレイン回路になっていることがわかります。したがって出力端子にはプルアップ抵抗が必要です。3.3v を 5.0v にレベル変換しますのでプルアップ抵抗は 5 v に吊ります。

74HC07 の Vcc をいくらにするかを検討します。マイコンボードの 14 ピンソケット (IC2) は、14 番ピンは 5v、7 番ピンが GND につながっています。表 10.6 から、Vcc が 4.5 v のとき入力信号レベルが 3.15 v 以上ないと High レベルとは認識しないことがわかります。こ

のことから V_{CC} を 5v にした場合は入力信号レベルは 3.5v 以上程度は必要と推測できます。74HC07 の V_{CC} を 5v にした場合は、FPGA の High のレベル 3.3v を High と認識できない可能性があります。このような理由からここでは 74HC07 の V_{CC} を F P G A ボードから供給するとして 3.3v にします。汎用ロジック IC 74HC07 の電源（74HC07 の 14 番ピン）には 3.3v を印加します。

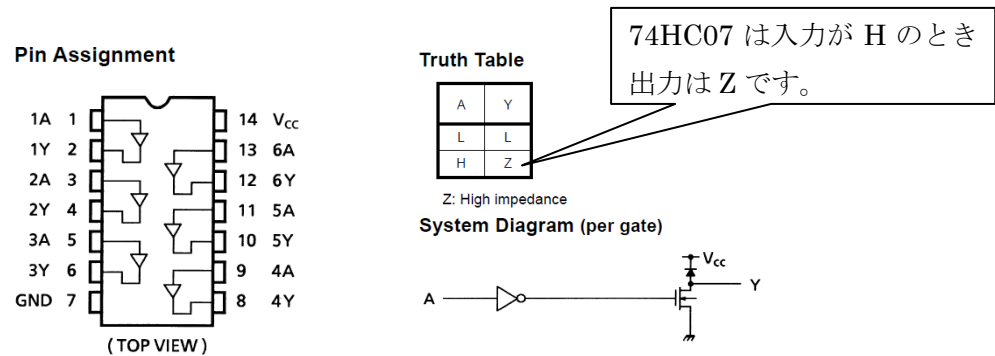


図 6 74HC07AP の端子配置と内部構造

V_{CC} が 4.5v のとき入力信号レベルが 3.15v 以上ないと High レベルとは認識しないことがわかる。

| Characteristics | Symbol | Test Conditions | Ta = 25°C | | | | Ta = -40 to 85°C | | Unit | |
|--------------------------|--------|--------------------------------|----------------------------|------|------|------|------------------|------|------|---|
| | | | VCC (V) | Min | Typ. | Max | Min | Max | | |
| High-level input voltage | VIH | — | 2.0 | 1.50 | — | — | 1.50 | — | V | |
| | | | 4.5 | 3.15 | — | — | 3.15 | — | | |
| | | | 6.0 | 4.20 | — | — | 4.20 | — | | |
| Low-level input voltage | VIL | — | 2.0 | — | — | 0.50 | — | 0.50 | V | |
| | | | 4.5 | — | — | 1.35 | — | 1.35 | | |
| | | | 6.0 | — | — | 1.80 | — | 1.80 | | |
| Low-level output voltage | VOL | VIN = VIL | IOL = 20 μA | 2.0 | — | 0.0 | 0.1 | — | 0.1 | V |
| | | | | 4.5 | — | 0.0 | 0.1 | — | 0.1 | |
| | | | IOL = 4 mA IOL = 5.2 mA | 2.0 | — | 0.0 | 0.1 | — | 0.1 | |
| | | | | 4.5 | — | 0.17 | 0.26 | — | 0.33 | |
| | | | | 6.0 | — | 0.18 | 0.26 | — | 0.33 | |
| Output off-state current | IOZ | VIN = VIH or VIL VOUT = VCC | 6.0 | — | — | ±0.5 | — | ±5.0 | μA | |
| Input leakage current | IIN | VIN = VCC or GND | 6.0 | — | — | ±0.1 | — | ±1.0 | μA | |
| Quiescent supply current | ICC | VIN = VCC or GND | 6.0 | — | — | 1.0 | — | 10.0 | μA | |

表 6 74HC07 の D C 特性の一例

【実習4】電圧レベル変換回路の設計

マイコンボードの基本 I/O 部の 14pin IC ソケットには 74HC07 が実装されています。0v/3.3v を入力すると出力側で 0v/5.0v にレベル変換される回路を 74HC07 を使って設計しましょう。

■ 手順

① 回路設計

- 74HC07 の 14 番ピン (Vcc) を 3.3V に接続します。

⇒ 74HC07 を IC ソケットから取り出し、74HC07 の 14 番ピンだけを横に伸ばすようにして、74HC07 を再度 IC ソケットに挿入したときに 74HC07 の 14 番ピンがマイコンボードの IC ソケットに接触しないような配慮・工夫が必要です。

⇒ FPGA ボードに 3.3v があるので、74HC07 の 14 番ピン (Vcc) にはそこから Vcc を供給するようにします。

- 74HC07 の出力端子 (2 番ピン) は Nch オープンドレインですから、4.7k Ω 程度のプルアップ抵抗で 5v 電源に吊ります。

⇒ マイコンボードに 5.0v があります。

- マイコンボード、FPGA ボード、汎用ロジック IC 74HC07 のそれぞれの GND を接続します。

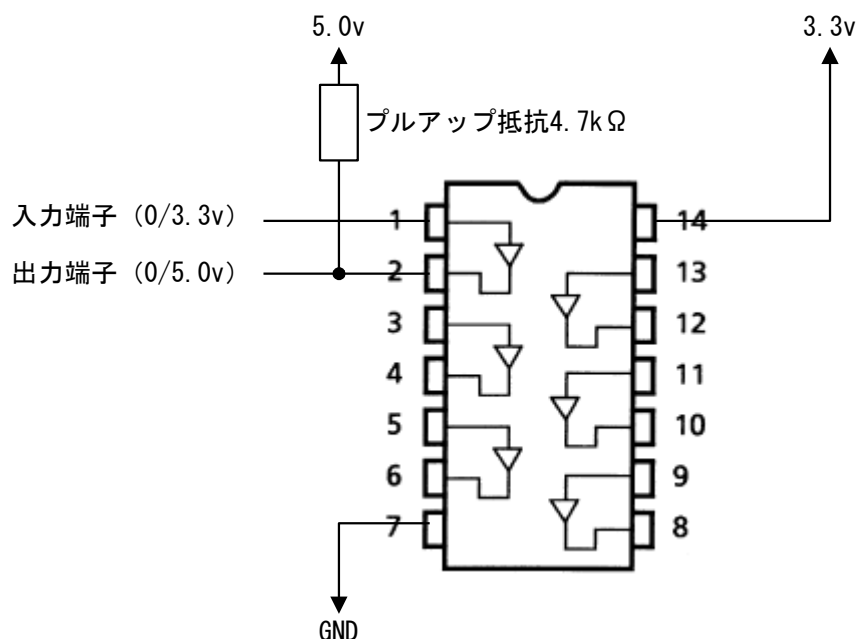


図7 74HC07 による電圧レベル変換回路

② 動作確認

・ 74HC07 の 1 番ピン（入力）に 3.3v を入力したときの 2 番ピン（出力）には何ボルトが出力されますか。 ⇒ 5.0v になってますか。

・ 74HC07 の 1 番ピン（入力）に 0v を入力したときの 2 番ピン（出力）には何ボルトが出力されますか。 ⇒ 0.0v になってますか。

4. 5 距離計システムを動作させる

■ システム全体の回路

電圧レベル変換回路を挿入したシステム全体の結線図を図8に示します。

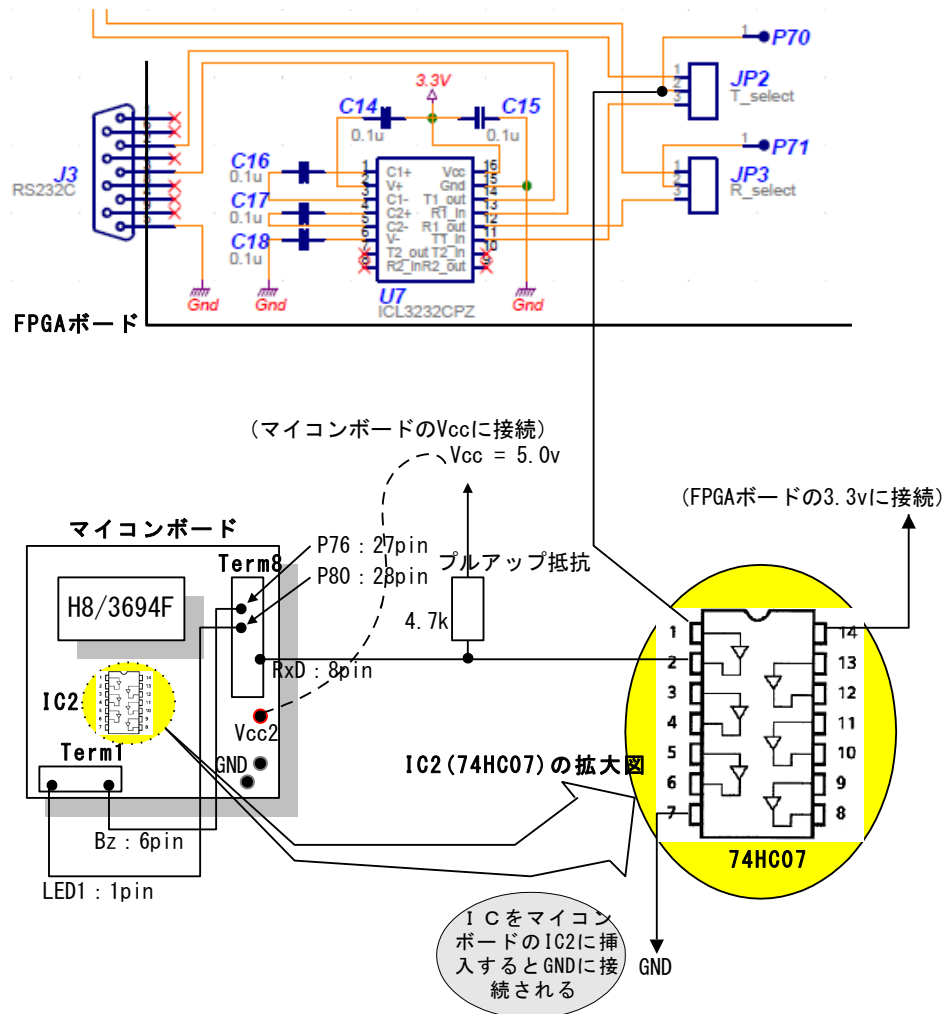


図8 距離計測システム結線図

■ 回路結線

- ① マイコンボード、FPGAボードの電源供給、停止を確認 (E8aエミュレータとの接続を未接続とする)
- ② マイコンボードとFPGAボード間のGNDの共通化:
マイコンボードのGND端子とFPGAボードのGND端子をつなぐ。
- ③ 74HC07の挿入: 74HC07の14番ピン (Vcc) の足を跳ね上げた状態でICソケットに挿入する。

ケット（IC2）に挿入。

- ④ LED1の接続：マイコンボードのLED1端子（Term1の1ピン）とP80（Term8の28ピン）をつなぐ。
- ⑤ 圧電ブザーの接続：マイコンボードのBz端子（Term1の6ピン）とP76（Term8の27ピン）をつなぐ。
- ⑥ FPGAとマイコンボード間の通信線の接続：
 - ・74HC07の2番ピンとマイコンボードのRx D端子（Term8の8ピン）をつなぐ。
 - ・74HC07の1番ピンとFPGAボードのJP2の2番ピンをつなぐ
- ⑦ プルアップ抵抗接続：4.7kΩの抵抗を介してマイコンボードのVcc（5v）端子につなぐ。
- ⑧ 74HC07に電源供給：跳ね上がっている74HC07の14番ピンとFPGAボードの3.3v端子をつなぐ。

■ マイコンプログラムの実装

① マイコン側プログラムの基本動作およびプログラム構成

FPGAから1バイト受信するごとにシリアル通信割り込み処理が動作し、1バイト単位に受信データを解析します。SODからEODまでのデータが揃ったところで割り込み処理で作成した距離データをメイン関数（main()）で取り込み、距離データに応じたマンマシンインターフェース出力（ブザー、LED、7segment LED）します。

これを実現するために、マイコン側プログラムではFPGAからのデータをベクタ番号23のSCI3割り込み処理で1バイト単位に受信します。EODまで正常に受信したとき、「距離データ更新要求フラグ」をセットします。メインプログラム中ではこの「距離データ更新要求フラグ」をポーリングしており、フラグがセットされているときに最新の距離データに更新します。メインプログラムでは最新距離データに応じたブザー、LED等の点滅等を実施します。

図9にメイン関数（main（））のフローチャートを示します。

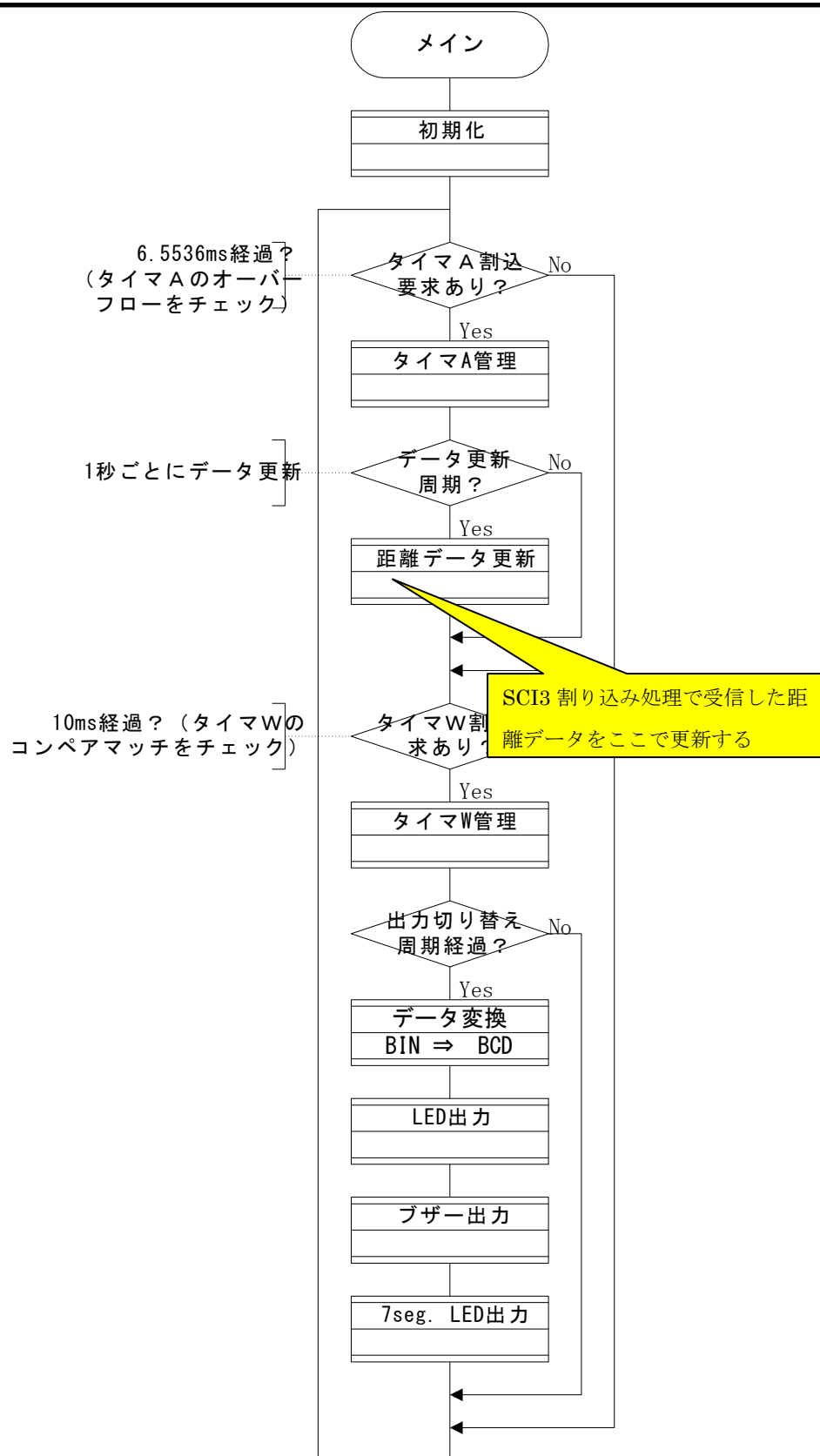


図9 メイン関数のフローチャート

② 新規プロジェクトの挿入

いままで使用してきた「h25_fukkou_mcu」というワークスペースのなかの「contact_sensor」という名称のプロジェクトを新たに挿入して、そのプロジェクトの中に完成済みのソースファイル（ans_contact_sensor.c と ans_intprg.c）を追加するという手順でプログラムを作成します。

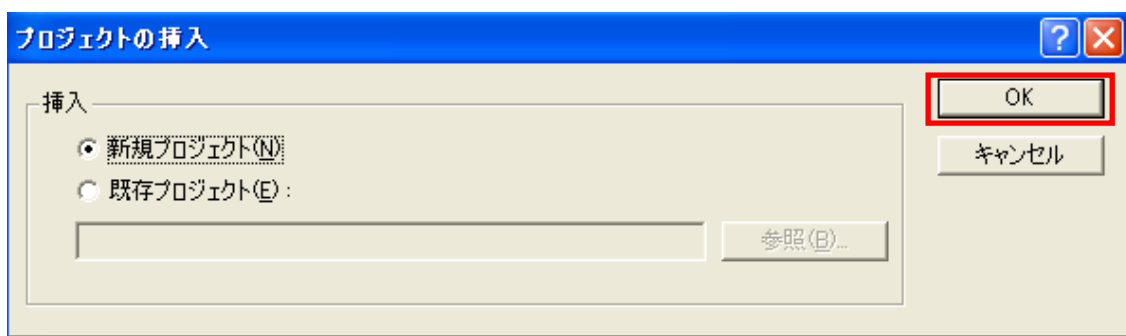
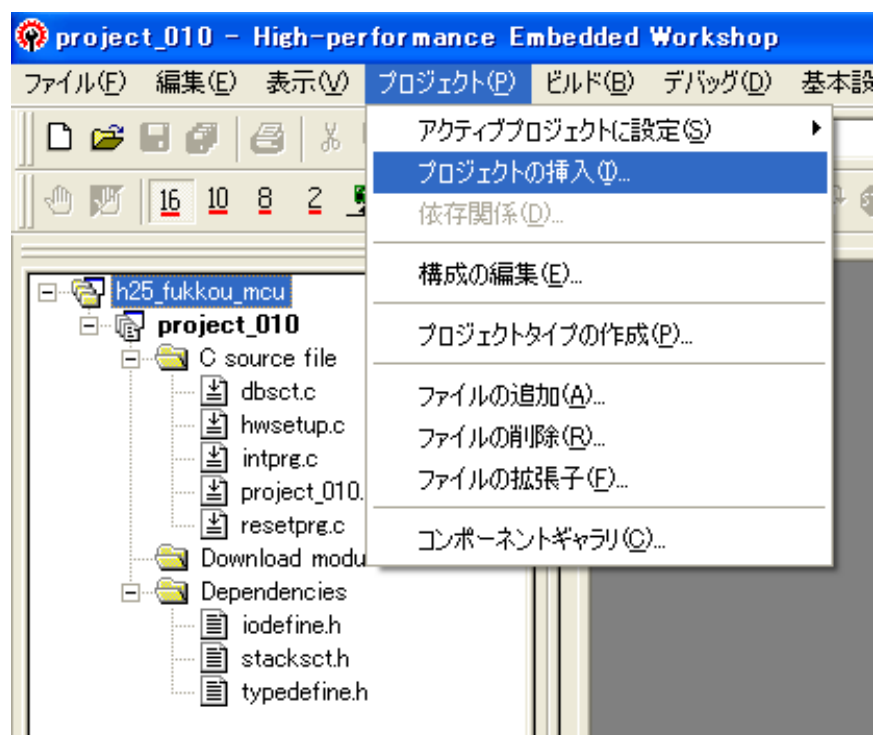
ふたつの完成済みソースコードファイルは、次のとおりです。

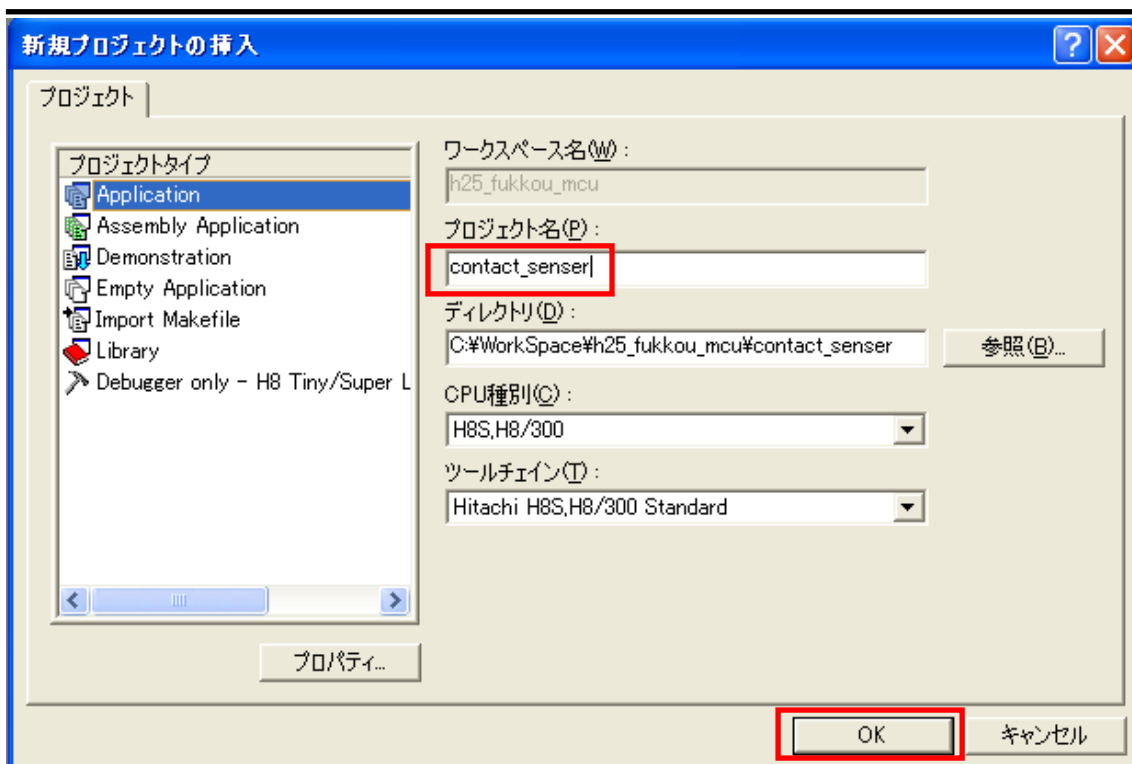
- ans_contact_sensor.c : 新規プロジェクト挿入時に自動生成される「contact_sensor.c」の代わりとなるソースコードファイル。ポートの初期設定等は一般には hwsetup.c のなかで実施しますが、今回は ans_contact_sensor.c のなかで実施しています。したがって hwsetup.c ファイルは追加しません。プロジェクト「contact_sensor」生成時に生成された hwsetup.c をそのまま使います。
- ans_int_prg.c : FPGA からのデータを受信する割り込みルーチンを組み込んだ割り込み処理プログラムのソースコードファイル

③ 「contact_sensor」プロジェクトの作成

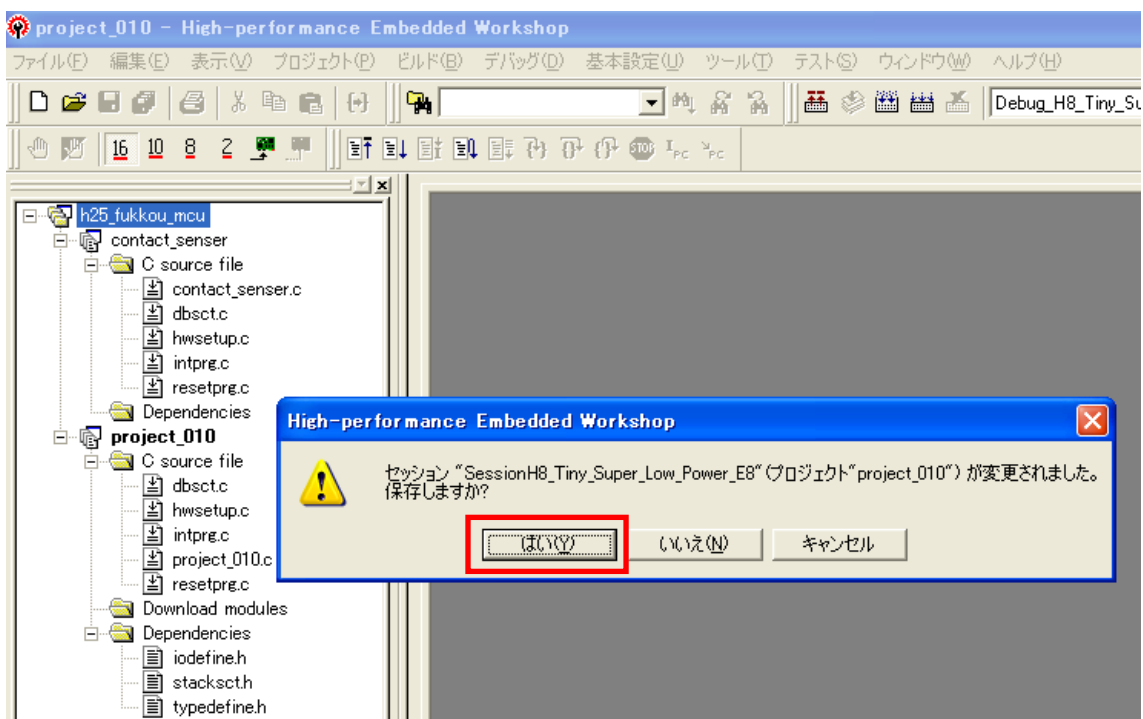
「h25_fukkou_mcu」ワークスペースの中には、すでに実習 2 で作成したプロジェクト「project_010」があります。この「h25_fukkou_mcu」ワークスペースの中に新規に「contact_sensor」という名称のプロジェクトを作成します。手順を示します。

「プロジェクト (P)」>「プロジェクトの挿入 (I)」を選択すると「プロジェクト挿入」画面が表示されます。「新規プロジェクト (N)」にチェックを入れて「OK」をクリックします。





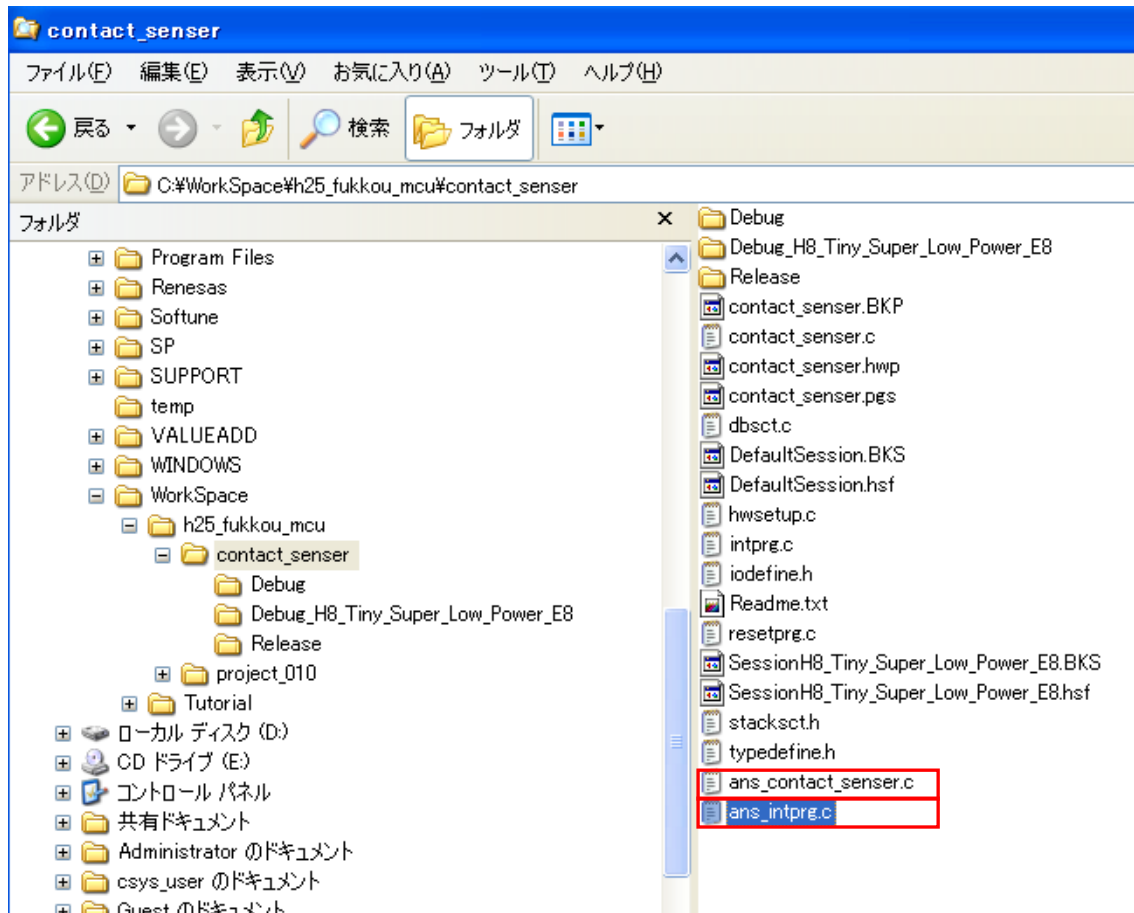
プロジェクト名に「contact_sensor」と入力して「OK」をクリックすると「2. 4 新規プロジェクトの作成」で説明した一連の画面が表示されます。「新規プロジェクト-1/9-CPU」から「新規プロジェクト-9/9-生成ファイル名」まで同様の手順を踏むことで新規プロジェクト「contact_sensor」生成時の初期画面が下記のように表示されます。



ワークスペースウィンドウに「contact_senser」プロジェクトが生成されたことを確認して「はい」をクリックします。

③ ファイル「ans_contact_senser.c」と「ans_intprg.c」の「contact_senser」フォルダへのコピー

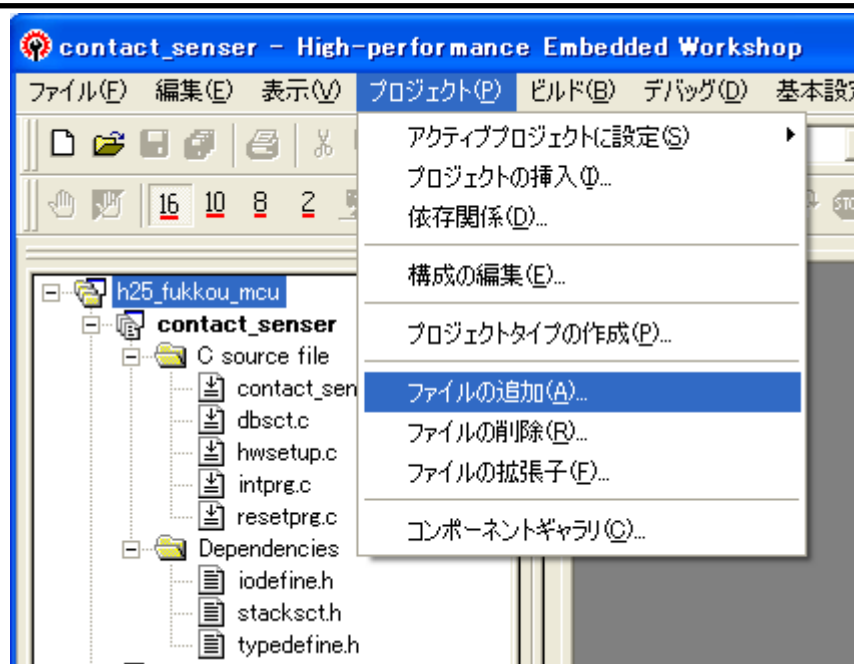
添付 CD-ROM 中の「距離計システムソースコード」からファイル ans_contact_senser.c と ans_intprg.c を WorkSpace>h25_fukkou_mcu>contact_senser フォルダにコピーします。



ワークスペースウィンドウに表示されたプロジェクト「contact_senser」のファイル群はHEWが自動生成したものです。③でファイルコピーした「ans_contact_senser.c」と「ans_intprg.c」はワークスペースウィンドウには見えません。

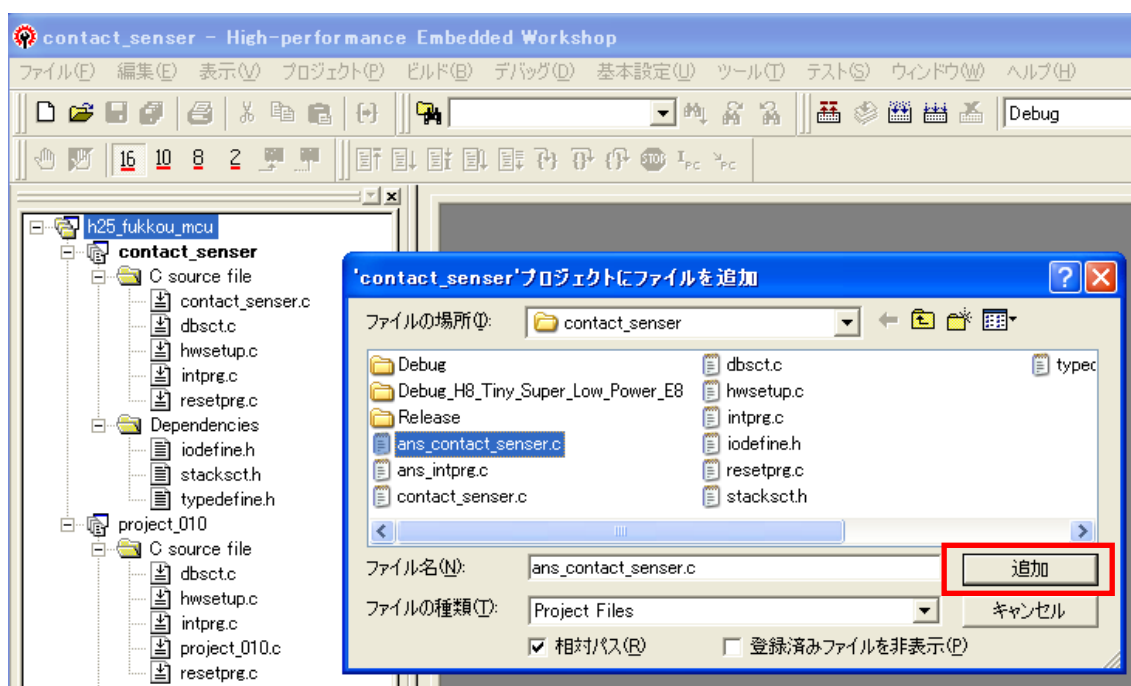
⑤ 「ans_contact_senser.c」と「ans_intprg.c」のプロジェクトへの追加

「ans_contact_senser.c」ファイルを当該プロジェクトに追加します。HEWの画面から「プロジェクト (P)」>「ファイルの追加 (A)」を選択して、左クリックします。

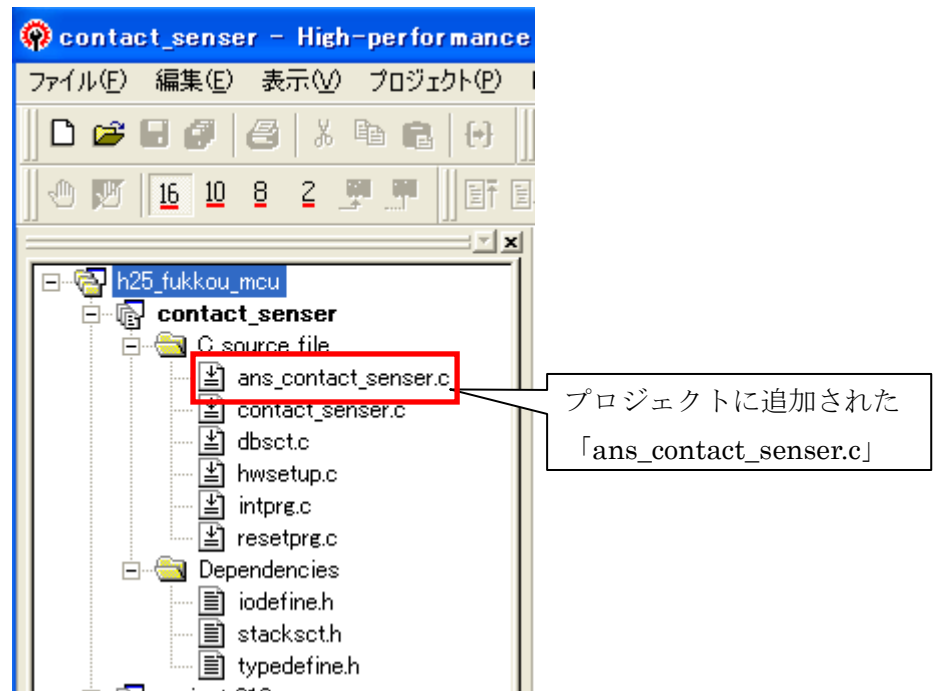


「プロジェクトにファイルを追加」の画面が表示されます。

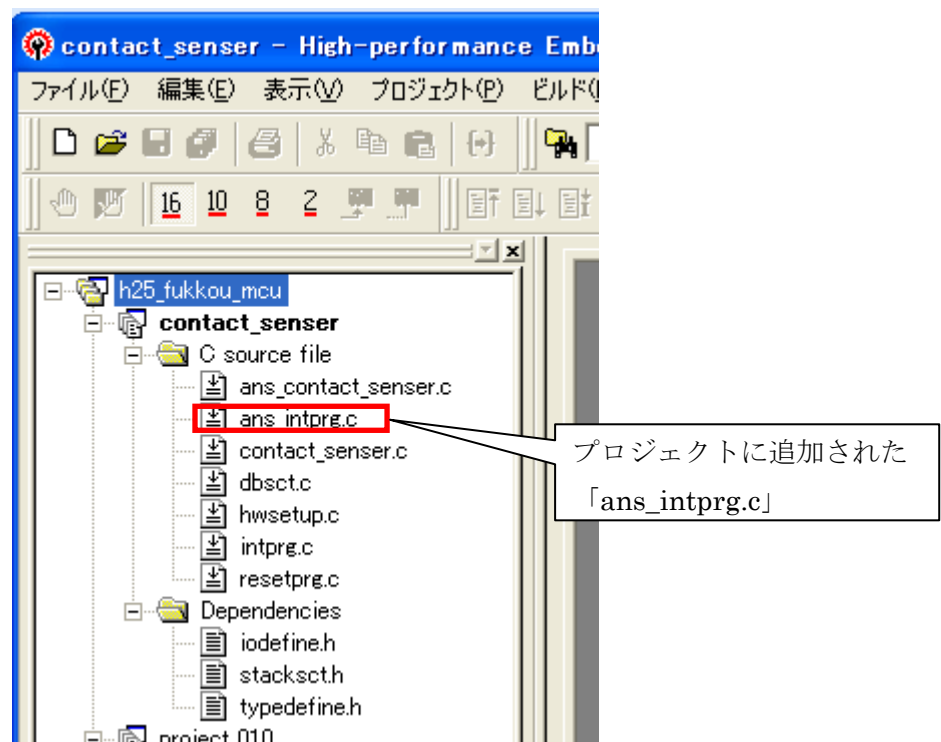
「ans_contact_sensor.c」を選択して「追加」をクリックします。



ワークスペースウィンドウに「ans_contact_sensor.c」が表示されます。



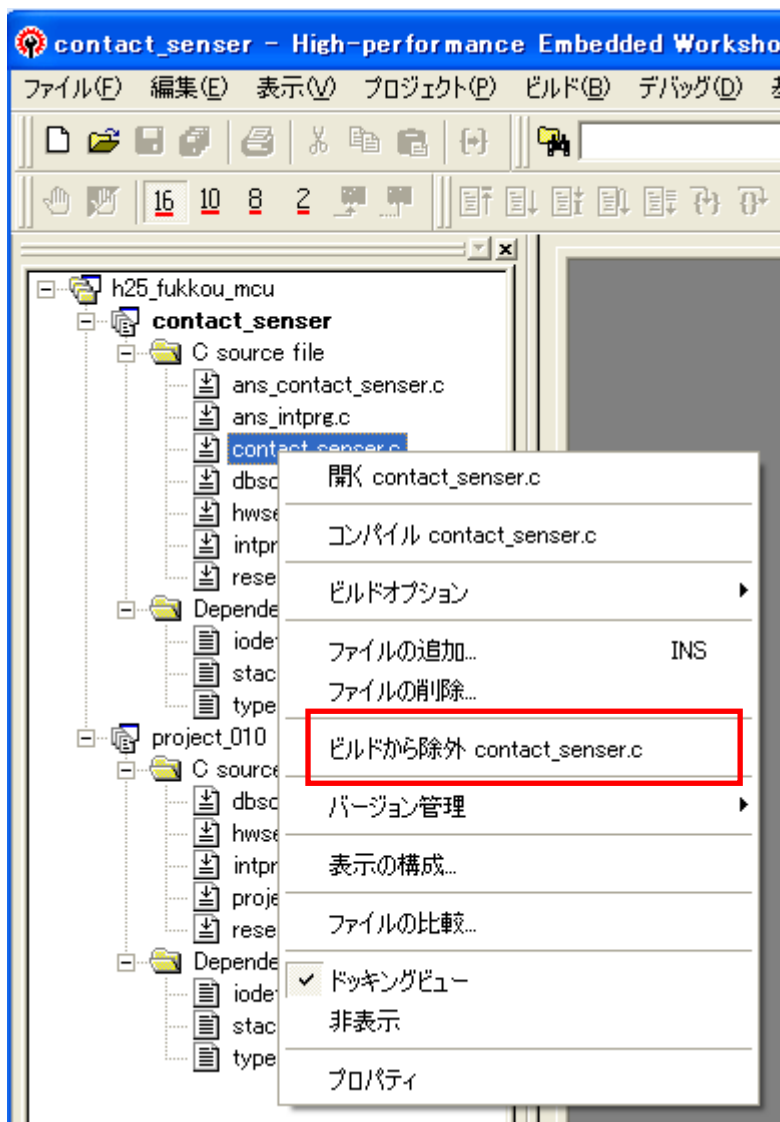
同様に「ans_intprg.c」を「contact_sensor」プロジェクトに追加します。



この状態でビルドを行うとメイン関数（main（ ））が「contact_senser.c」と「ans_contact_senser.c」の両ファイルに存在することになりコンパイルエラーになりますので「contact_senser.c」はビルドから除外します。同様に理由で「intprg.c」もビルドから除外します。

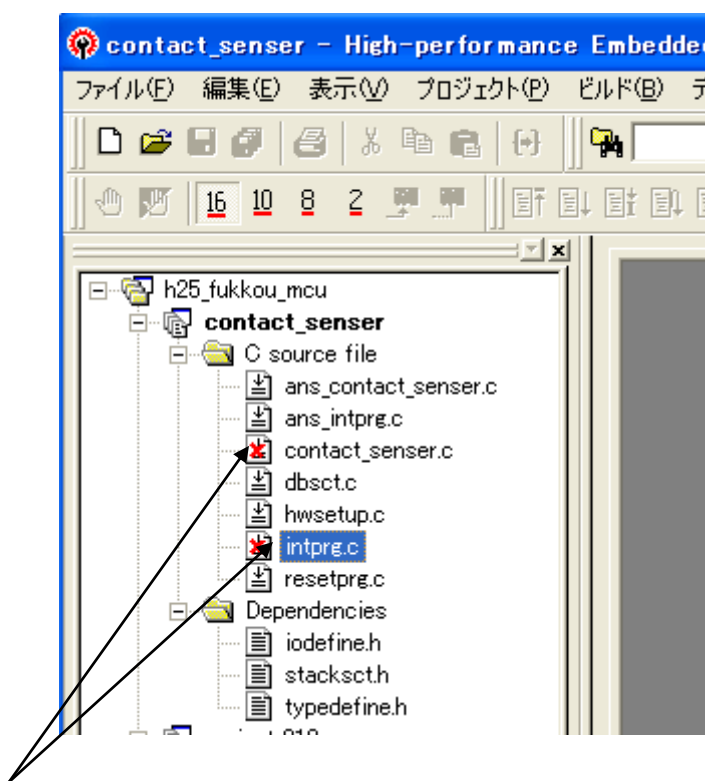
⑥ 「contact_senser.c」と「intprg.c」のビルドからの除外

「HEW」のプロジェクトウィンドウから「contact_senser.c」を選択して右クリックします。「ビルドから除外 contact_senser.c」にカーソルを合わせます。



そのまま左クリックすると、プロジェクトウィンドウのファイル「contact_senser.c」

に赤い「×」印が出ます。同様に「intprg.c」についてもビルドから除外します。



ビルドから除外されたファイル

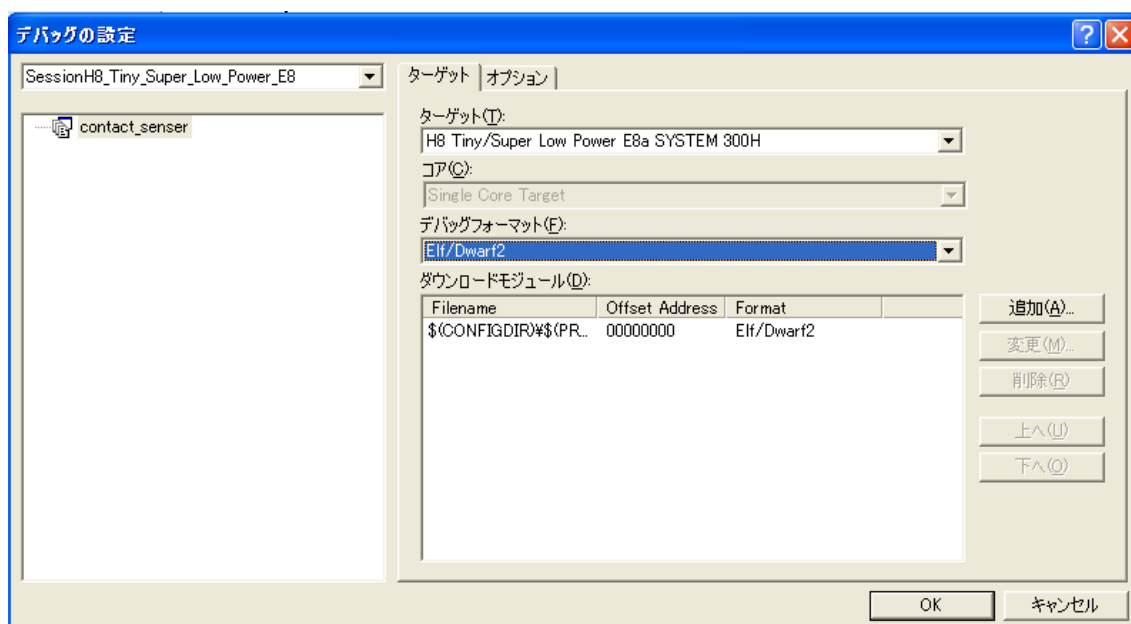
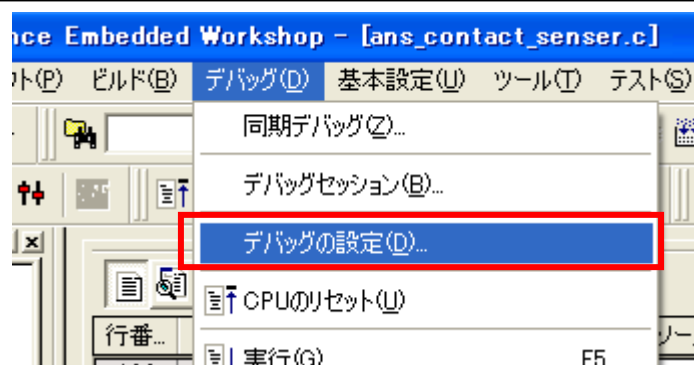
以上で、「contact_sensor」プロジェクトに最終ソースコードのファイルが揃いました。「ビルド」を行い「no error」を確認後、E 8 a エミュレータを接続してダウンロードモジュール「contact_sensor.abs」をマイコンのフラッシュメモリへダウンロードします。

■ プログラムの書き込みフォーマット（ロードモジュール）について

コンパイラは、以下の 5 種類のロードモジュールを選択出力できます。

- ・リロケータブル E L F 形式
- ・アブソリュート E L F 形式
- ・S タイプ形式
- ・H E X 形式
- ・バイナリ形式

メニューバーから「デバッグ」>「デバッグ設定」の順に選択するとデバッグ設定画面が表示されます。



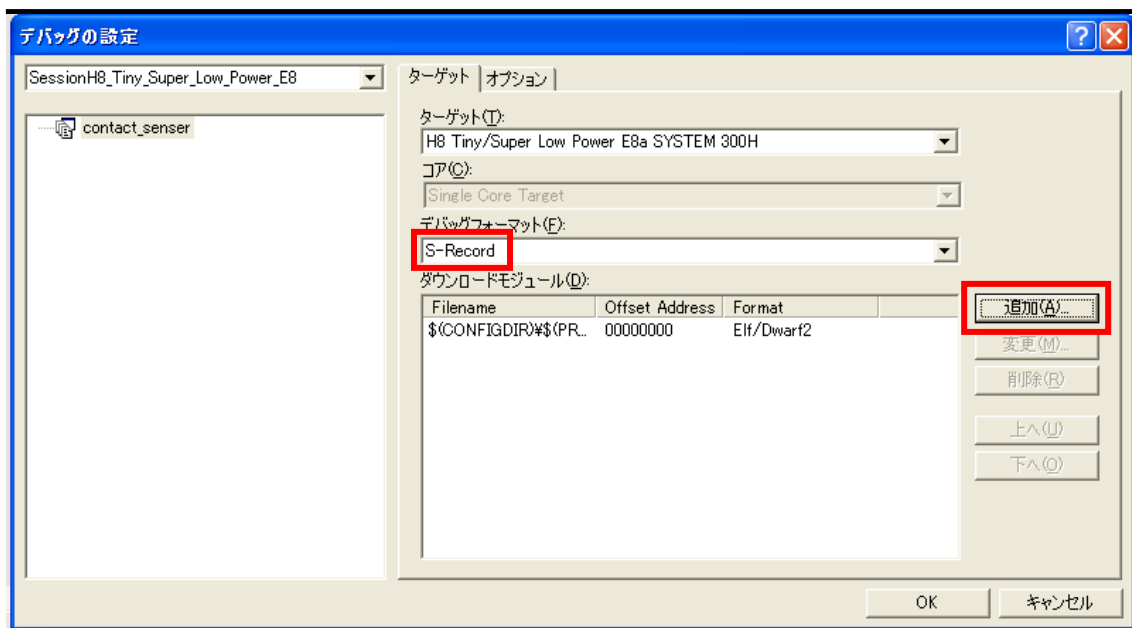
「デバッグフォーマット (F)」には「Elf/Dwarf2」が設定されています。このフォーマットはデバッグ用です。

上記状態では、ビルドしたときの生成ロードモジュールが「Elf/Dwarf2」フォーマット形式のファイル、ここでは「contact_senser.abs」が生成されます。

書き込みにはSタイプ形式（ここでは「S-Record」）を設定します。Sタイプ形式のコードをフラッシュメモリに書き込むことでマイコンボード単独（デバッグを接続しない状態）で動作します。

■ Sタイプ形式のロードモジュールの書き込み手順

「デバッグの設定」画面のデバッグフォーマット (F)」から「S-Record」を選択します。次に「追加 (A)」をクリックします。



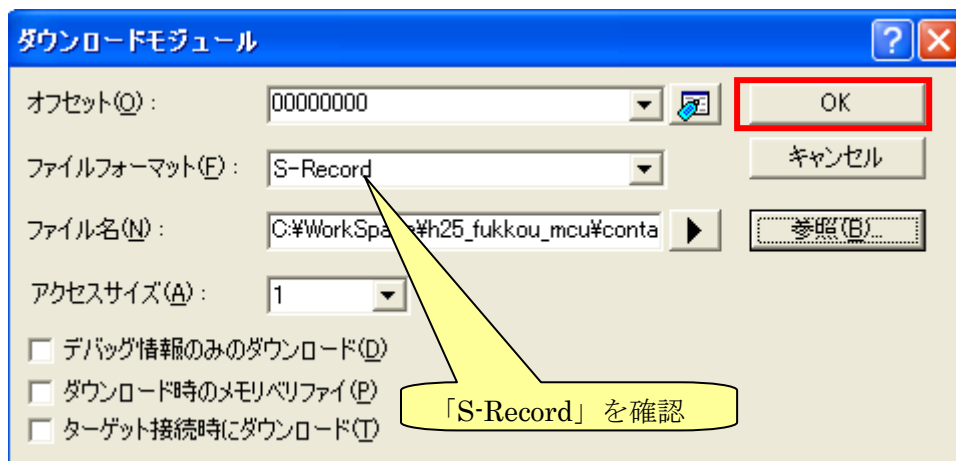
下記「ダウンロードモジュール」画面が表示されます。「参照 (B)」をクリックします。



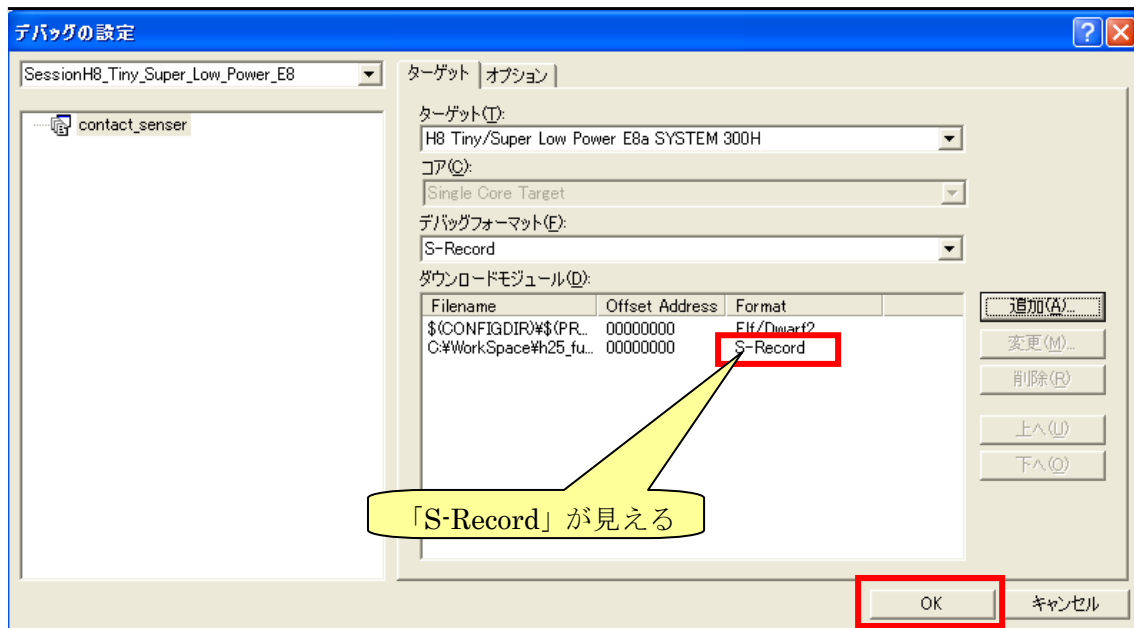
下記「ダウンロードモジュールの選択」画面が表示され、ウィンドウの拡張子「.mot」のファイルが見えます。拡張子「.mot」の当該ファイルを選択して「選択」をクリックします。



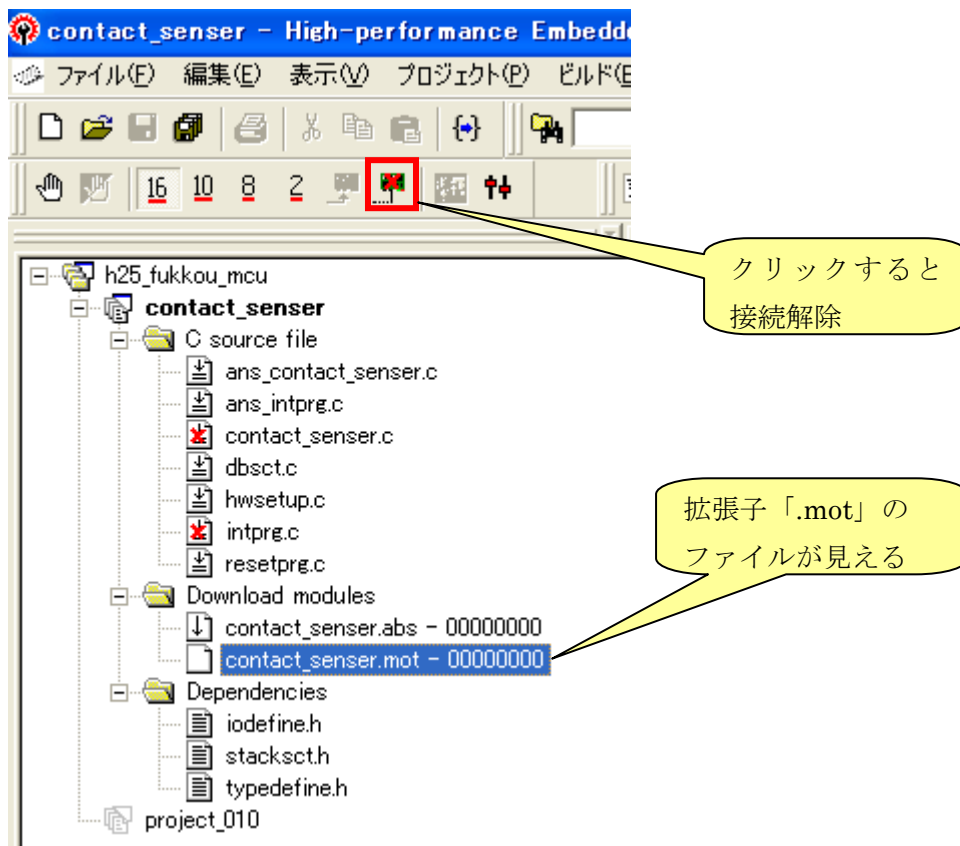
再度、下記「ダウンロードモジュール」画面が表示されます。ファイルフォーマットは「S-Record」、ファイルの拡張子は「.mot」であることを確認して「o k」をクリックします。



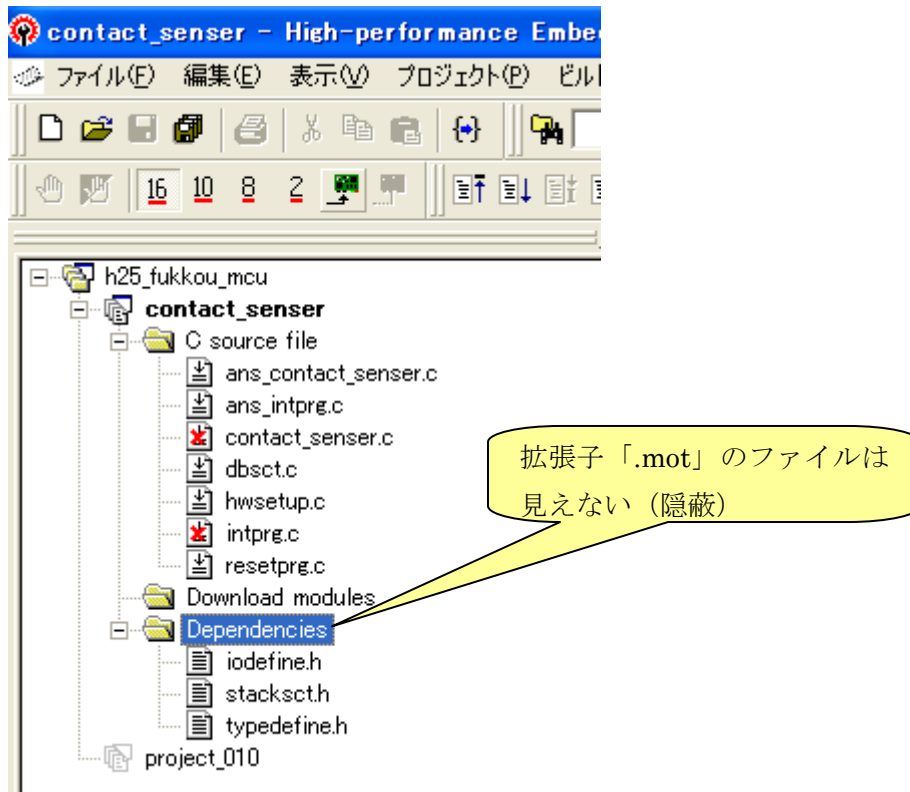
再度、「デバッグの設定」画面は表示されて、「ダウンロードモジュール」ウィンドウの「Format」欄に「S-Record」が見えるのを確認して「o k」をクリックします。



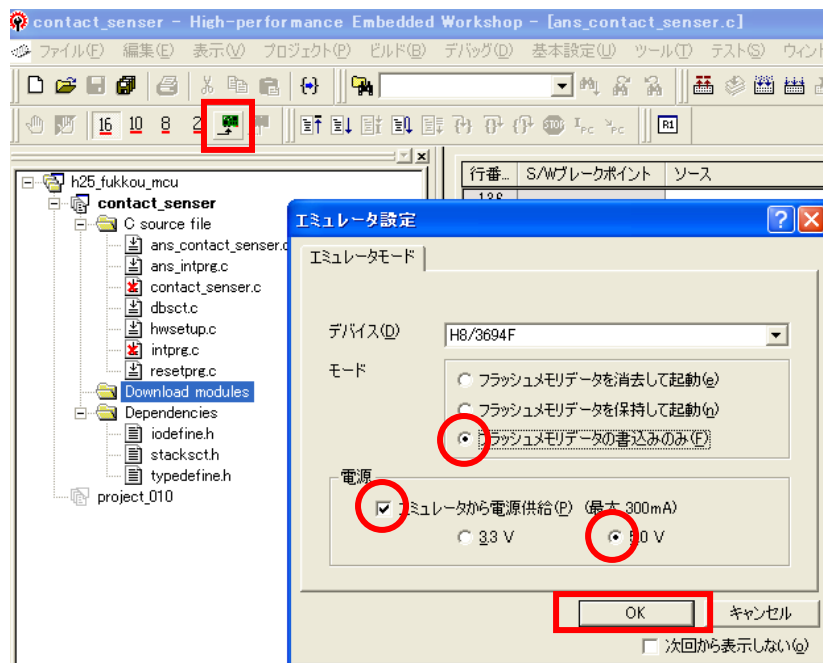
HEWのプロジェクトウィンドウの「Download modules」のSタイプ形式のロードモジュール「contact_senser.mot」が見えます。確認したらE8aとの接続を解除します（E8aと未接続時は、当該操作は不要）。



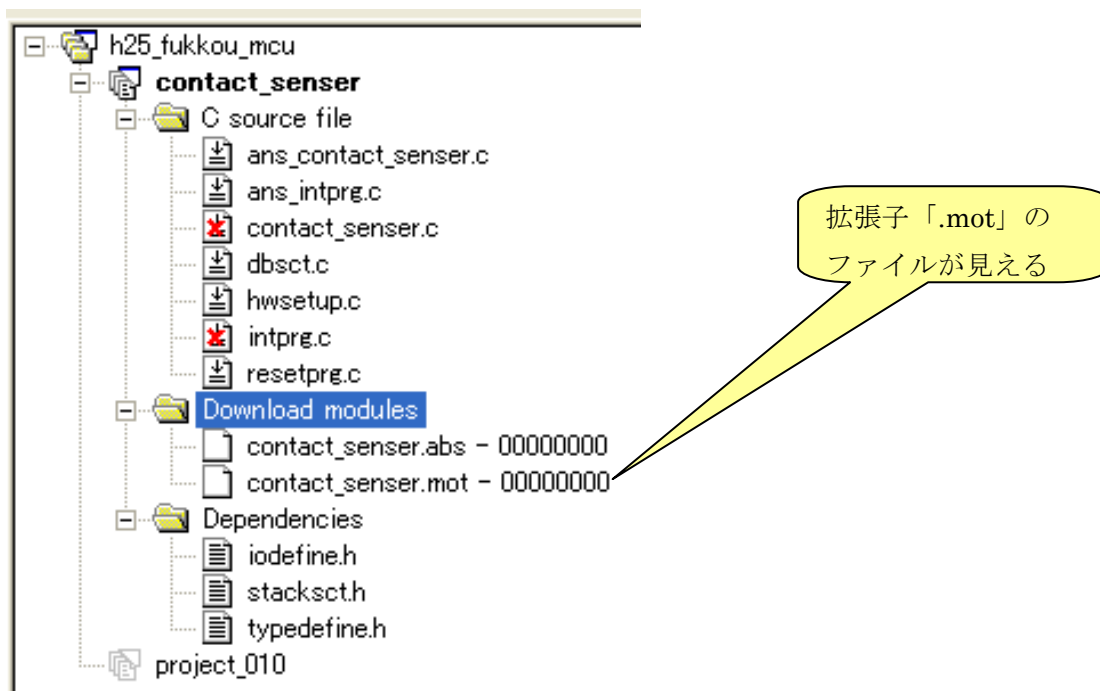
接続解除すると「Download modules」の中はプロジェクトウィンドウから隠蔽されます。



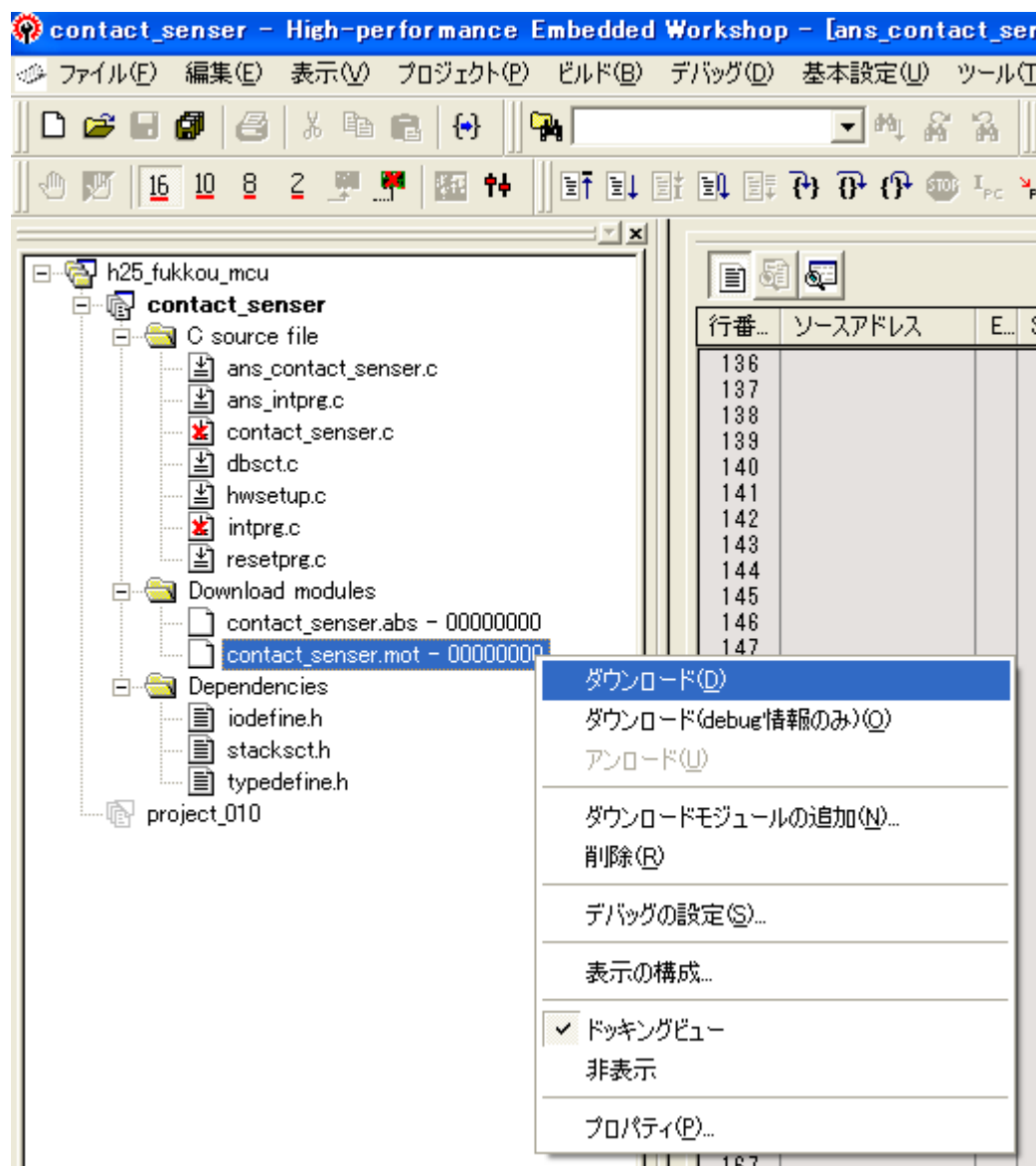
今度はE 8 a を単にフラッシュメモリデータの書き込み器として使用するために、E 8 a と接続します。「エミュレータから電源を供給 (P)」をチェック>「5.0 v」にチェック>「フラッシュメモリデータの書き込み (F)」にチェック>「OK」をクリックします。



接続完了すると、プロジェクトウィンドウの「Download modules」に隠蔽されていたロードモジュールが見えます。



拡張子「.mot」ファイル、「contact_senser.mot」を選択＞「ダウンロード (D)」にポインタを合わせる＞左クリック。S 形式のロードモジュールがフラッシュメモリに書き込まれます。



書き込み動作が終了したら、E 8 a との接続を解除して、HEWを閉じます。マイコンボードのE 8 a のコネクタもはずします。

いままではE 8 a から電源を供給していましたが、マイコンボード単独で動作されるためには、E 8 a に代わる電源が必要ですのでACアダプタを使います。

マイコンボードにACアダプタを接続してボードのPower SW をONにするとマイコンボードは動作します。

■ 距離計システムの動作

ロードモジュールの書き込みが終了したら、マイコン側プログラムを実行します。FPGAボードの距離センサーの前に手のひらをかざして前後させてみましょう。「10.2 ソフトウェア詳細仕様」に記述されているように、手のひらと距離センサーとの距離に応じてマイコンボード上のブザー鳴動間隔が変化、LEDの点滅周期が変化、7セグメントLEDには距離に応じた数字が表示されます。

注意：7セグメントLED出力は、結線が未処理のため表示しません。

注意：受信側の処理について

FPGAボードには送信データ形式を切り替えるスイッチがあります。スイッチを押すごとに、Binary Data 形式 (SOD=0x42) と Decimal Data 形式 (SOD=0x44) を切り替えます。

【実習5】マイコンボード側の RxD 端子と FPGA の送信端子の波形を観測してみましょう。

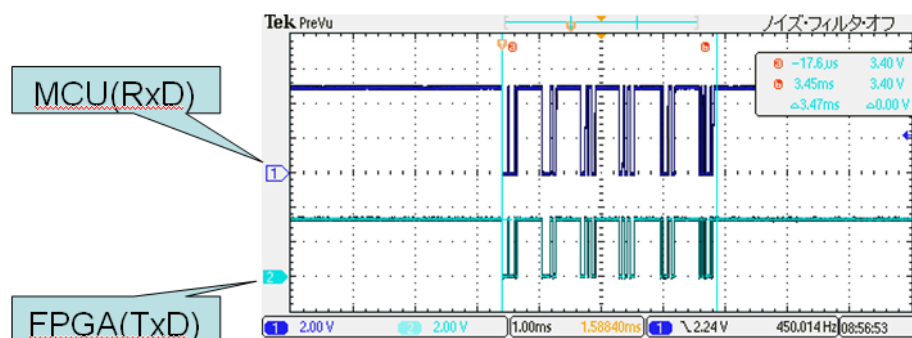
■ 手順

オシロスコープでマイコン側受信データと FPGA 送信データを2チャンネルで観測します。

⇒ FPGA 側では約 3.3v で出力しているがマイコンボード側の RxD 端子では約 5 v にレベル変換されていますか。

⇒ SOD 1 バイト、データ 4 バイト、EOD 1 バイトの全体で 6 バイトのデータを送受信していることがわかりますか？

⇒ 先頭の 1 バイトデータをスタートビットからストップビットまで 1 ビット単位にデータを確認しましょう。

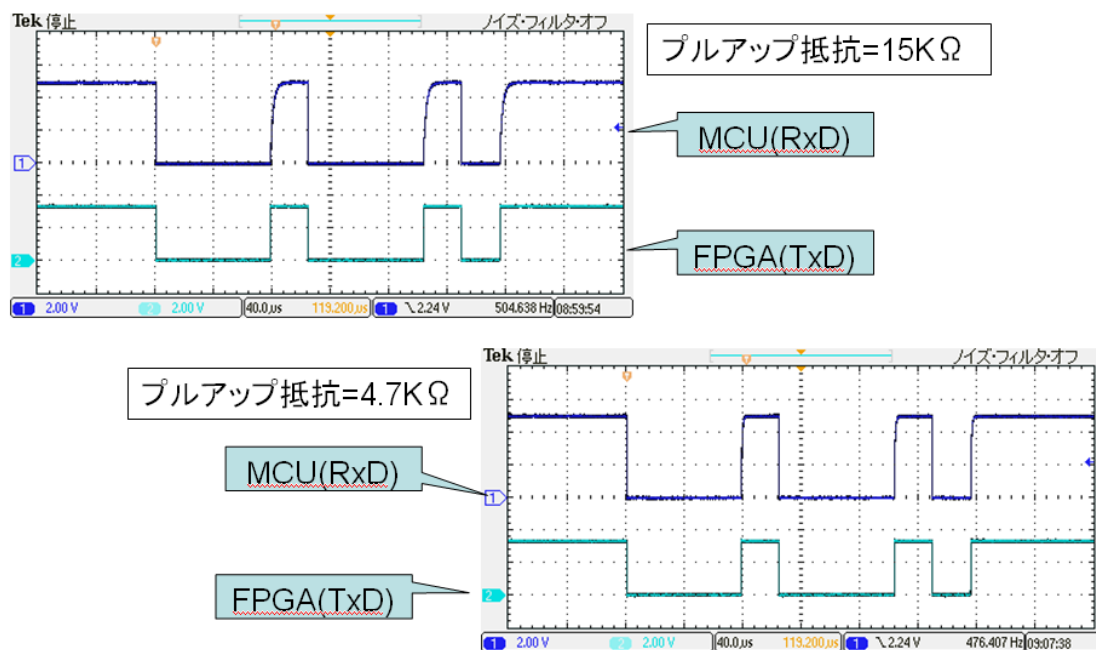


参考：マイコンボードの受信波形とFPGAボードの送信波形

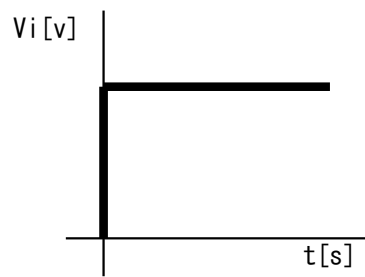
【実習6】74HC07 の出力端子（マイコンボードの RxD 端子）のプルアップ抵抗値を替えたときの波形を観測して、比較してみましょう。

■ 手順

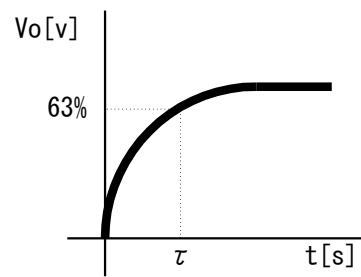
- ① プルアップ抵抗 4.7 k Ω のとき、FPGA 送信端子とマイコンボード受信端子の 2 チャンネルで観測します。
- ④ プルアップ抵抗 15 k Ω のとき、FPGA 送信端子とマイコンボード受信端子の 2 チャンネルで観測します。
- ⑤ ①と④で観測した波形を比較しましょう。
⇒ プルアップ抵抗が 4.7 k Ω 時の通信波形と 15 k Ω 時の通信波形を観測すると、4.7 k Ω よりも 15 k Ω の方が、信号の立ち上がりが鈍っていることが分かります。



送信側の方形波（a. 入力波形）は受信側では回路常数によって立ち上がりが多少鈍った波形になります（b. 出力波形）。63%まで立ち上がる時間を時定数といい τ で表します。この時定数はコンデンサ容量と抵抗値で決まります（式（1））。



a. 入力波形



b. 出力波形

$$\tau = CR \quad \dots \dots \dots \text{(式 1)}$$

τ : 時定数 (単位 : SEC) / C : コンデンサ容量 (単位 : F) / R : 抵抗値 (単位 : Ω)

プルアップ抵抗を小さくすれば τ が小さくなることが分かります。プルアップ抵抗により波形整形が可能です。

5

おわりに



本編の発展的学習としては、次のようなことが考えられます。

1. F P G Aボードとマイコンボードを RS-232-C ケーブルで接続した通信形態とする。
2. 近接状況を通知するマンマシンインタフェースを改良する。
3. ソフトウェア仕様書を新規作成して、仕様にしたがった新たにプログラム開発を試みる。

さらに専門的に要素技術ごとにより深く学ぶこと、次に記載するような事項を修得することをお勧めします。

1. マイコンのハードウェアマニュアルの精読
2. 測定技術の調査
3. 開発環境（HEW）のより詳細機能の活用
4. 設計論、コード書法を学ぶ
7. S P I、I 2 C、C A N、L I Nなどの通信規格を理解する

6

Appendix



表1 マイコンボード部品表

| No | 回路部名称 | 部品番号 | 品名 | 型番・仕様 | 数量 | メーカー | 備考 |
|----|-----------|-----------|------------------|-------------------|----|-----------|--|
| 1 | 電源回路部 | J11 | DCジャック | 21φセンター+ | 1 | | MCU部 Power(Vcc1)供給確認用 I/O部 Power(Vcc2)供給確認用 |
| | | sw11 | MCU Power スイッチ | スライド スイッチ | 1 | | |
| | | sw12 | I/O Power スイッチ | スライド スイッチ | 1 | | |
| | | LED11 | MCU部 Power LED | 8550形 | 1 | | |
| | | LED12 | I/O部 Power LED | 8550形 | 1 | | |
| | | R11 | 炭素皮膜抵抗 | 1.2K | 1 | | |
| | | R12 | 炭素皮膜抵抗 | 1.2K | 1 | | |
| | | C11 | 電解コンデンサ | 25v 47μF | 1 | | |
| | | C12 | 電解コンデンサ | 25v 47μF | 1 | | |
| 2 | リセット回路 | D21 | 整流用ダイオード | | 1 | | |
| | | C21 | 電解コンデンサ | 35v 10μF | 1 | | |
| | | R21 | 炭素皮膜抵抗 | 10K 1/8W | 1 | | |
| | | sw21 | タクト スイッチ | 小 | 1 | | |
| | | | | | 1 | | |
| 3 | LED回路 | LED31 | 小型LED(赤色) | SLP-881 A | 1 | SANYO | |
| | | LED32 | 小型LED(赤色) | SLP-881 A | 1 | SANYO | |
| | | LED33 | 小型LED(赤色) | SLP-881 A | 1 | SANYO | |
| | | LED34 | 小型LED(赤色) | SLP-881 A | 1 | SANYO | |
| | | R31~R34 | 炭素皮膜抵抗 | 330 1/8W | 4 | | |
| 4 | Volume回路 | R41 | 炭素皮膜抵抗 | 1.2K 1/8W | 1 | | 10k~20kΩ程度 |
| | | VR41 | 半固定抵抗 | 20K | 1 | | |
| | | C41 | セラミックコンデンサ | 01μF | 1 | | |
| 5 | 圧電プザー回路 | R51 | 炭素皮膜抵抗 | 1.2K 1/8W | 1 | | |
| | | R52 | 炭素皮膜抵抗 | 47K 1/8W | 1 | | |
| | | Bz51 | 圧電プザー | PKM11-4A11(他断式) | 1 | MURATA | |
| 6 | 基本ゲート回路 | IC2 | ICソケット | 14PIN | 1 | | |
| | | C61 | セラミックコンデンサ | 01μF | 1 | | |
| 7 | Sw入力回路 | sw71 | タクト スイッチ | 小 | 1 | | |
| | | sw72 | タクト スイッチ | 小 | 1 | | |
| | | R71 | 炭素皮膜抵抗 | 47K 1/8W | 1 | | |
| | | R72 | 炭素皮膜抵抗 | 47K 1/8W | 1 | | |
| | | JP71 | ジャンパピン(ショートピン付き) | 3pin ピンヘッダを流用 | 1 | | |
| | | JP72 | ジャンパピン(ショートピン付き) | 3pin ピンヘッダを流用 | 1 | | |
| 8 | DIPスイッチ回路 | R81 | 集合抵抗 | コモン 47k 5端子 | 1 | | |
| | | sw81 | dipスイッチ | 4回路 | 1 | | |
| 9 | Tr(FET)回路 | FET1 | Tr(FET)用ソケット | | | | 未実装(ソケットのみ用意) |
| 10 | 7seg 回路 | 7seg101 | 小型赤色7seg LED | GL9A040G(アノードコモン) | 1 | SHARP | IF=20mA |
| | | IC3 | シンゴドライバ(8回路) | TD62083APG | 1 | TOSHIBA | |
| | | R101~R106 | 炭素皮膜抵抗 | 270 | 8 | | |
| | | C101 | セラミックコンデンサ | 01μF | 1 | | |
| 11 | MCU回路 | CN1 | ピンヘッダソケット | 26pin(2列) | 1 | | MB+H8A挿入コネクタ |
| | | CN2 | ピンヘッダソケット | 26pin(2列) | 1 | | MB+H8A挿入コネクタ |
| | | | マイコンボード | MB+H8A | 1 | Sunhayato | H8/3694F搭載 |
| 12 | E8接続回路 | CN3 | E8用コネクタ | H-LDI(14pin) | 1 | | |
| | | R121 | 集合抵抗 | コモン 47K 9端子 | 1 | | ブルアップ用 |

◇ 参考・引用 文献 ◇

1. E8a エミュレータ ユーザーズマニュアル 別冊 (H8/300H Tiny シリーズ接続時の注意事項)、(株)ルネサスエレクトロニクス
2. E8a エミュレータ ユーザーズマニュアル、(株)ルネサスエレクトロニクス
3. H8/3694 グループ ハードウェアマニュアル、(株)ルネサスエレクトロニクス
4. High-performance Embedded Workshop V4.09 ユーザーズマニュアル、(株)ルネサスエレクトロニクス
5. TD62083 データシート、(株)東芝
6. TC74HC07 データシート、(株)東芝
7. 新プロトコルハンドブック、朝日新聞社
8. MB-H8A 取り扱い説明書、サンハヤト(株)

平成 25 年度文部科学省委託
「東日本大震災からの復興を担う専門人材育成支援事業」
東北の復興を担う自動車組込みエンジニア育成支援プロジェクト
実践！自動車組込み技術者講座 F P G Aとマイコンの連携システム
(ソフトウェア基礎編)

平成 26 年 3 月
学校法人日本コンピュータ学園（東北電子専門学校）
〒980-0013 宮城県仙台市青葉区花京院一丁目 3 番 1 号
TEL : 022-224-6501

●本書の内容を無断で転記、掲載することは禁じます。