

**R8C/M12A**  
**マイコン**  
**実習マニュアル**  
**ブレッドボード版**

第 1.02 版

2015.04.20

株式会社日立ドキュメントソリューションズ

# 注意事項 (rev.6.0H)

## 著作権

- ・本マニュアルに関する著作権は株式会社日立ドキュメントソリューションズに帰属します。
- ・本マニュアルは著作権法および、国際著作権条約により保護されています。

## 禁止事項

ユーザーは以下の内容を行うことはできません。

- ・第三者に対して、本マニュアルを販売、販売を目的とした宣伝、使用、営業、複製などを行うこと
- ・第三者に対して、本マニュアルの使用権を譲渡または再承諾すること
- ・本マニュアルの一部または全部を改変、除去すること
- ・本マニュアルを無許可で翻訳すること
- ・本マニュアルの内容を使用しての、人命や人体に危害を及ぼす恐れのある用途での使用

## 転載、複製

本マニュアルの転載、複製については、文書による株式会社日立ドキュメントソリューションズの事前の承諾が必要です。

## 責任の制限

本マニュアルに記載した情報は、正確を期すため、慎重に制作したものです。が万一本マニュアルの記述誤りに起因する損害が生じた場合でも、株式会社日立ドキュメントソリューションズはその責任を負いません。

## その他

- ・本マニュアルに記載の情報は本マニュアル発行時点のものであり、株式会社日立ドキュメントソリューションズは、予告なしに、本マニュアルに記載した情報または仕様を変更することがあります。製作に当たりましては、最新の内容を確認いただきますようお願いいたします。
- ・すべての商標および登録商標は、それぞれの所有者に帰属します。

## 連絡先

株式会社 日立ドキュメントソリューションズ  
〒135-0016 東京都江東区東陽六丁目3番2号 イースト21タワー  
E-mail:himdx.m-carrally.dd@hitachi.com

# 目次

1. 概要.....	1
1.1 概要.....	1
1.2 マイコンの仕様.....	2
1.3 マイコンのピン配置.....	3
2. セット内容.....	4
2.1 部品表.....	4
2.2 セット以外に必要な部材.....	5
2.3 基本回路.....	6
2.3.1 回路図.....	6
2.3.2 ブレッドボード実装図.....	7
3. 開発環境の使い方.....	8
4. マイコンへプログラムを書き込む.....	9
4.1 RY-WRITER 基板との接続.....	9
4.2 書き込み手順.....	10
5. LED の点灯(I/O ポートの出力).....	12
5.1 概要.....	12
5.2 ブレッドボードの接続.....	12
5.3 プロジェクト.....	13
5.4 操作方法.....	13
5.5 プログラム.....	14
5.6 課題.....	15
6. スイッチの入力(I/O ポートの入力).....	16
6.1 概要.....	16
6.2 ブレッドボードの接続.....	16
6.3 プロジェクト.....	17
6.4 操作方法.....	17
6.5 プログラム.....	18
6.6 課題.....	19
7. ボリューム(0~5V の電圧)の入力(A/D 変換器).....	21
7.1 概要.....	21
7.2 ブレッドボードの接続.....	21
7.3 プロジェクト.....	22
7.4 操作方法.....	22
7.5 プログラム.....	23
7.6 課題.....	24
8. 圧電サウンダーから音を鳴らす(PWM 波形の出力).....	25
8.1 概要.....	25
8.2 ブレッドボードの接続.....	25

目次

8.3 プロジェクト.....	26
8.4 操作方法.....	26
8.5 プログラム.....	27
8.6 課題.....	28

1. 概要

# 1. 概要

## 1.1 概要

本マニュアルは、「R8CM12A ブレッドボードセット」を使ったマイコン実習について説明しています。本セットは、ブレッドボードを使い、半田不要でルネサス エレクトロニクス製の R8C/M12A マイコンを使った実習をすることができます。

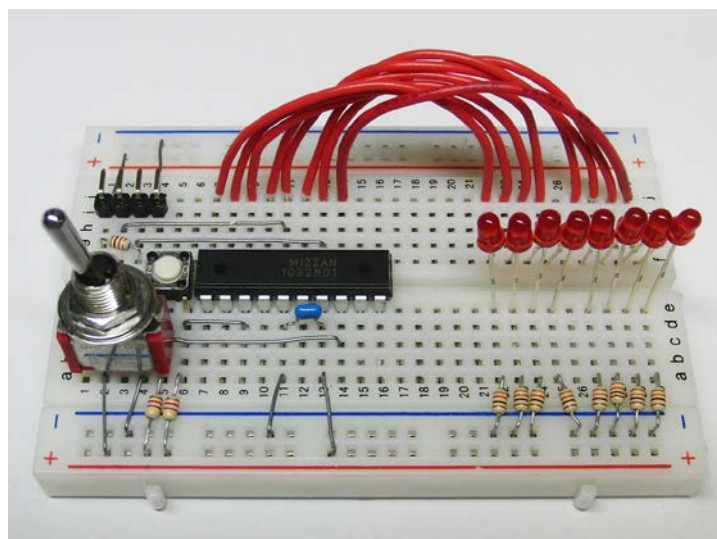
実習内容を下記に示します。

- LED 8 個の点滅(出力、タイマーの実習) ※タイマーは無限ループによるソフトタイマーです
- ディップスイッチの状態入力、LED へ出力(入出力、if 文による分岐についての実習)
- 電圧入力(ボリュームを使った A/D 変換実習)
- 音を鳴らす(圧電サウンダーを使った PWM 波形出力の実習)

その他、これらを組み合わせて、様々な制御実習をすることができます。



▲セット内容



▲実習中の例

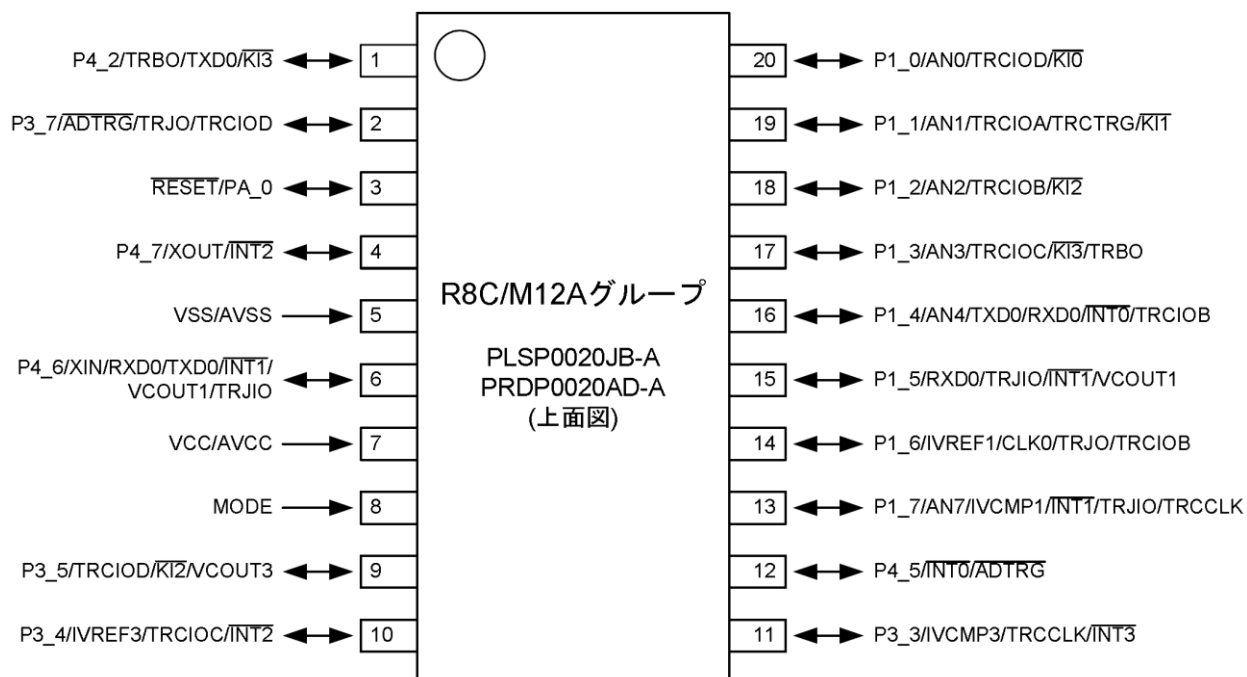
## 1. 概要

## 1.2 マイコンの仕様

パッケージ	20ピン DIP	
動作周波数/ 電源電圧	~5MHz (VCC=1.8V~5.5V) 5MHz~20MHz (VCC=2.7V~5.5V)	
メモリ	プログラム ROM	8192bytes (8KB) プログラム/イレーズ回数: 10,000 回
	データ フラッシュ	2048bytes (2KB) プログラム/イレーズ回数: 10,000 回
	RAM	512bytes
I/O ポート	<ul style="list-style-type: none"> <li>・CMOS 入出力: 17 端子、プルアップ抵抗を選択可能</li> <li>・大電流ポート: 8 端子</li> </ul>	
タイマ	タイマ RJ2	16ビット×1 タイマモード、パルス出力モード(周期ごとのレベル反転出力)、イベントカウンタモード、パルス幅測定モード、パルス周期測定モード
	タイマ RB2	8ビット×1 (8ビットプリスケアラ付)、または 16ビット×1 (選択可能) タイマモード、プログラマブル波形発生モード(PWM 出力)、プログラマブルワンショット発生モード、プログラマブルウェイトワンショット発生モード
	タイマ RC	16ビット×1 (キャプチャ/コンペアレジスタ 4 本付) タイマモード(アウトプットコンペア機能、インプットキャプチャ機能)、PWM モード(出力 3 本)、PWM2 モード(PWM 出力 1 本)
A/D コンバータ	分解能: 10ビット×6 チャンネル	
コンパレータ B	コンパレータ B1、コンパレータ B3	
通信	UART0 クロック同期形シリアル I/O / 非同期形シリアル I/O 兼用	
クロック発生回路	<ul style="list-style-type: none"> <li>・低速オンチップ(内蔵)オシレータ: 約 125kHz (リセット後のクロック) ※60~250kHz、標準は 125kHz</li> <li>・高速オンチップ(内蔵)オシレータ: 約 20.0MHz ※19.0~21.0MHz、標準は 20.0MHz</li> <li>・XIN クロック発振回路: 外付けで 2~20MHz のオシレータを搭載可能</li> </ul>	

1. 概要

1.3 マイコンのピン配置



ピン	I/O 名	詳細
1	P4_2	
2	P3_7	
3	RESET/ PA_0	リセット端子と PA_0 端子が兼用のピンです。リセット後はリセット端子です。“0”でリセット、“1”で実行です(実行時は、4.7kΩでプルアップしてください)。プログラムで PA_0 端子に切り替えることができます。
4	P4_7	外付けのクリスタル P4_7 が兼用のピンです。XIN クロック入力として使用するか、P4_7 として使用するかはプログラムで切り替えます。
5	VSS/AVSS	電源の 0V(GND)に接続します。
6	P4_6	外付けのクリスタル P4_6 が兼用のピンです。XIN クロック入力として使用するか、P4_6 として使用するかはプログラムで切り替えます。
7	VCC/AVCC	電源の 5V に接続します。
8	MODE	書き込み時は 0V、プログラム実行時は 4.7kΩでプルアップして電源を ON にします。
9	P3_5	
10	P3_4	
11	P3_3	
12	P4_5	
13	P1_7	
14	P1_6	パソコンからプログラムを書き込むとき、受信線として使用します。プログラム実行時は、P1_6 として使用します。この端子に負荷が接続されていると、書き込みできません。
15	P1_5	
16	P1_4	パソコンからプログラムを書き込むとき、送信線として使用します。プログラム実行時は、P1_4 として使用します。この端子に負荷が接続されていると、書き込みできません。
17	P1_3	
18	P1_2	
19	P1_1	
20	P1_0	

## 2. セット内容

## 2.1 部品表

部品番号	名称	型式	写真	メーカー	数量
	ブレッドボード	EIC-801		(株)秋月電子通商	1
U1	マイコン R8C/M12A	R5F2M122ANDD ※ROM:8KB RAM:512Bytes		ルネサス エレクトロニクス(株)	1
CN1	ピンヘッダ (オスL型)	C-01627 ※4ピン分	 ※長い側をブレッドボードに挿します	(株)秋月電子通商	1
SW1	タクトスイッチ	B3F-1050		オムロン(株)	1
SW2	トグルスイッチ (1回路2接点 (ON-ONタイプ))	P-03913		(株)秋月電子通商	1
C1	積層セラミック コンデンサ	RPEF11H104Z2K1A01B 0.1 $\mu$ F(104) 5.08mm ピッチ		(株)村田製作所	1
LED1~ LED8	LED	EBR3338S 直径 $\phi$ 3mm 赤		スタンレー電気(株)	8





## 2. セット内容

SW3	ディップ スイッチ	A6D-4103		オムロン(株)	1
BZ1	圧電 サウンダー	PKM13EPYH4002-B0		(株)村田製作所	1
VR1	ボリューム	CT-6P 10k $\Omega$ (103)		日本電産コパル 電子(株)	1
R1~R2	抵抗	CFS1/4C 4.7k $\Omega$ (黄・紫・赤・金)		コア(株)	2
R3~R11	抵抗	CFS1/4C 1k $\Omega$ (茶・黒・赤・金)		コア(株)	9
	単線	$\phi$ 0.5mm 被覆あり		各社	約 1.5m

## 2.2 セット以外に必要な部材

パソコンから R8C/M12A にプログラムを書き込むとき、下記部材が必要です。

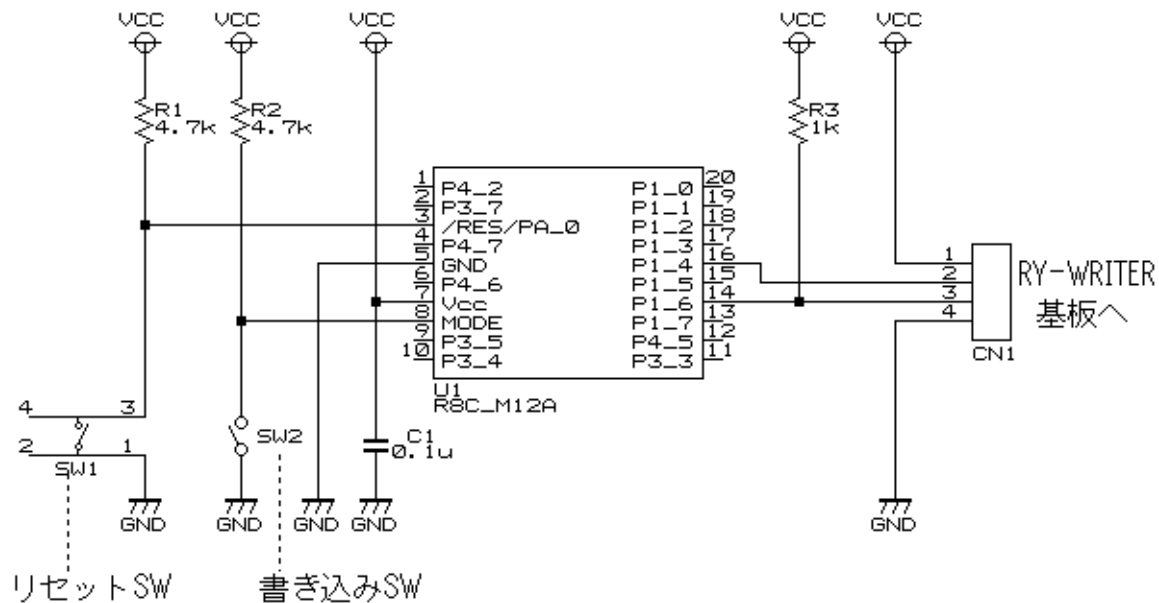
部品番号	名称	型式	写真	メーカー	数量
	RY-WRITER 基板	M-S183		日立インターメ ディックス(株)	1
	USB ケーブル	A コネクタ-ミニ B コネクタ タイプ		各社	1

## 2. セット内容

## 2.3 基本回路

## 2.3.1 回路図

基本的な回路を下図に示します。実習に応じて、部品を追加していきます。



CN1	このコネクタと RY-WRITER 基板を接続して、電源(5V)の供給、プログラムの書き込みを行います。 <b>パソコンの USB コネクタの 5V 電源を使用しますので、モータなど、大電流が流れる回路は接続しないでください。</b>
SW1	マイコンをリセットさせます。 ただし、リセット端子をプログラムで PA_0 端子に切り替えた場合は、リセットしません(電源を OFF にするまでマイコンはリセットしません)。
SW2	リセットスイッチ(SW1)を押しながらこのスイッチを上にして、リセットスイッチを離すと、プログラム書き込みモードになります。 リセットスイッチ(SW1)を押しながらこのスイッチを下にして、リセットスイッチを離すと、書き込まれているプログラムを実行します。

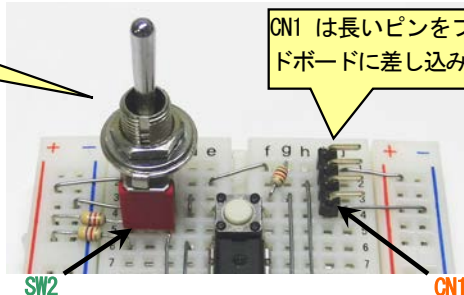
2. セット内容

2.3.2 ブレッドボード実装図

基本部品のブレッドボードへの実装を下図に示します。実習に応じて、実装部品を追加していきます。

SW2 は背が高いので、いちばん最後に差し込みます

CN1 は長いピンをブレッドボードに差し込みます



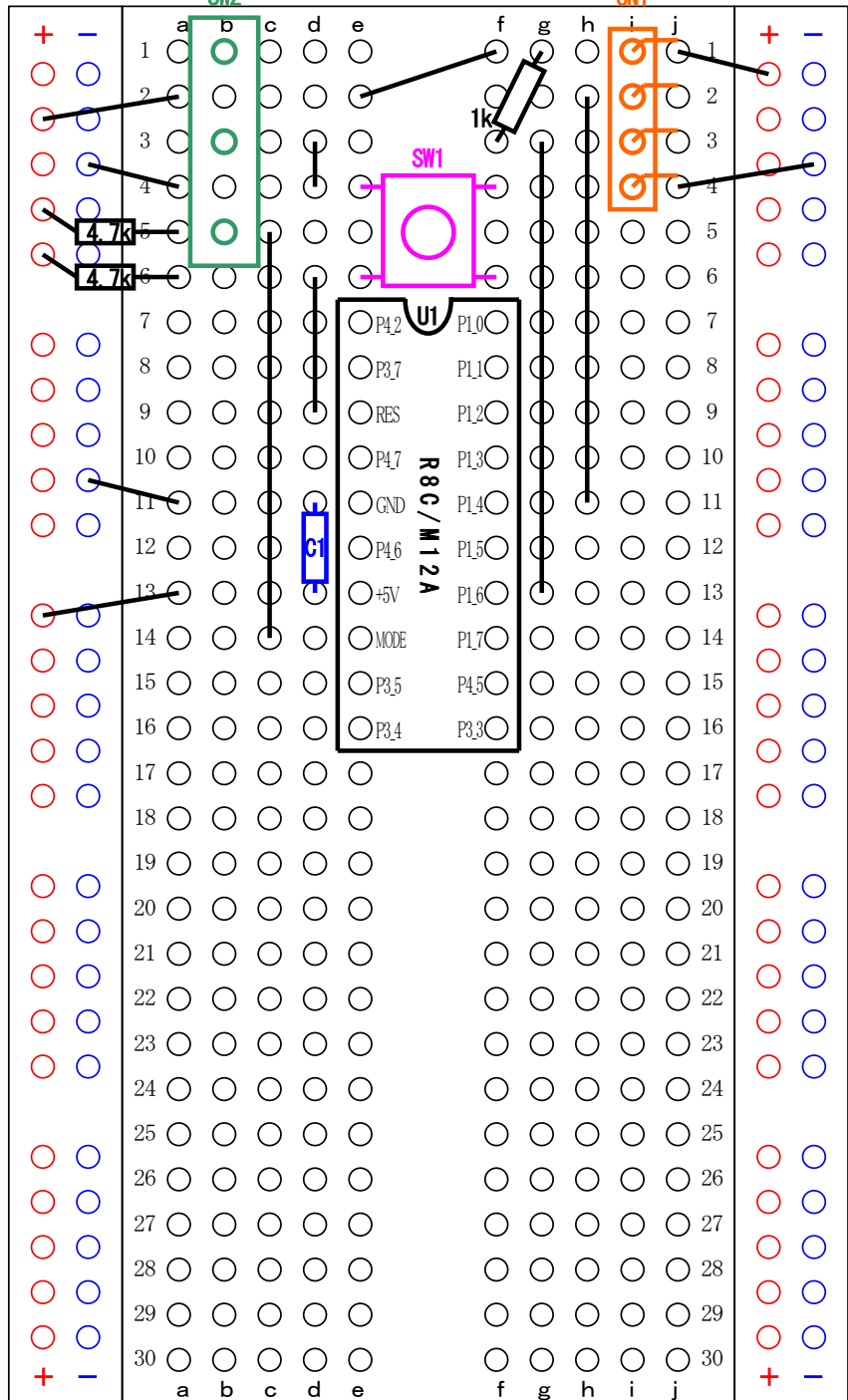
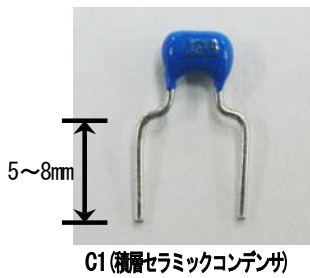
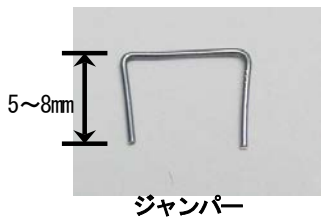
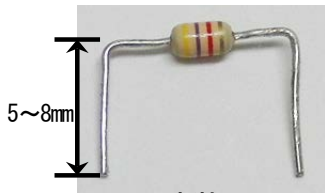
【注意事項】

※U1 は、向きがあります。

抵抗, SW1, SW2, C1 は向きがありません。

CN1 は、長いピンをブレッドボードに差し込みます。

ブレッドボードに差し込むリードの足は、5~8mmに切ります。



### 3. 開発環境の使い方

R8C/M12A マイコンのプログラム開発として、「ルネサス統合開発環境(無償評価版)」を使用します(以下、「ルネサス統合開発環境」とします)。

<p><b>1</b></p> <p><b>免責事項</b></p> <p>「マニュアル」、「ソフトウェア」は万全な体制で制作されており、通常の使用環境においては正常に動作するように作成されていますが、万が一「マニュアル」、「ソフトウェア」による損失・損害が発生した時には、『ジャパンマイコンカーラー実行委員会』はいかなる場合も責任を負いません。個人の免責が取れる範囲内であらかじめ了承した上でご使用くださるようお願いをいたします。</p> <p><a href="#">サンプルプログラム、書き込みソフトのダウンロード(H8編)</a> 2010.10.07更新</p> <p><a href="#">サンプルプログラム、書き込みソフトのダウンロード(R8C編)</a> 2011.07.07更新</p> <p><a href="#">マイコンに関する資料(H8編)</a> 2009.05.25更新</p> <p><a href="#">マイコンに関する資料(R8C編)</a> 2011.05.11更新</p>	<p>マイコンカーラーのダウンロードページ http:// www.mcr.gr.jp/ tech/download/ main01.html にアクセスします。</p> <p>「マイコンに関する資料(R8C 編)」をクリックします。</p>
--	---

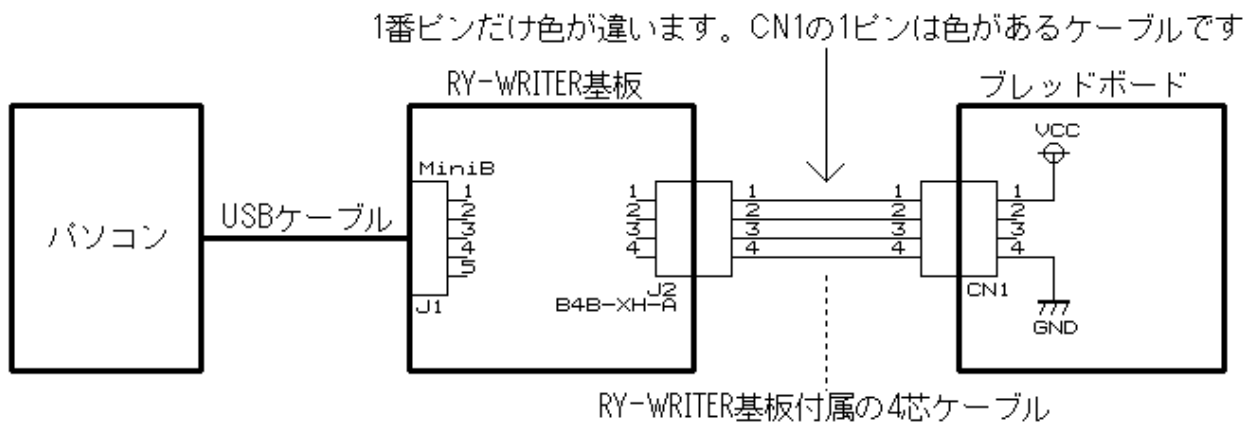
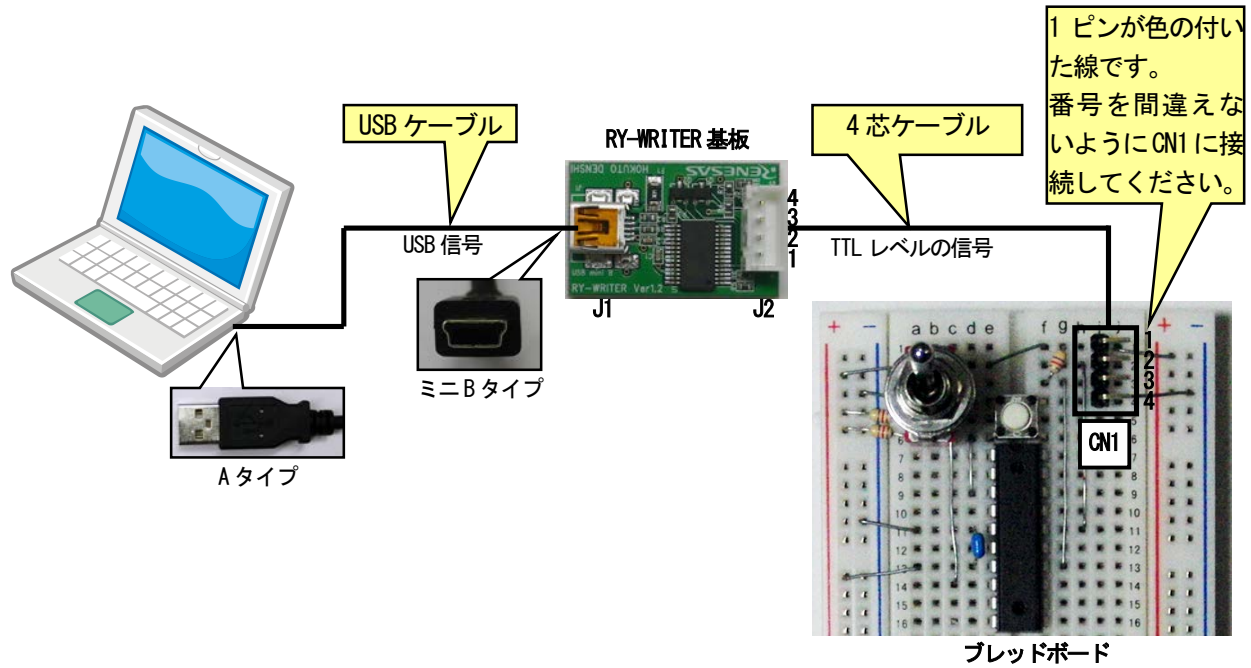
<p><b>2</b></p> <p><b>ダウンロード</b> <b>～マイコンに関する資料(R8C編)～</b></p> <p>R8Cマイコンに関する資料を掲載しています。 キットや基板は、<a href="#">リンク集</a>の、日立インターメディックス(株)で取り扱っています。</p> <p>●ルネサス統合開発環境 操作マニュアル(R8C/38A版) 第1.23版 2011.05.26 ルネサス統合開発環境のダウンロード方法、インストール方法、操作方法、及び書き込み方法を掲載しています。ルネサス統合開発環境は、ルネサス エレクトロニクス(株)のサイトよりダウンロードして下さい(マニュアル内に方法を記載しています)。 →<a href="#">DOWNLOAD</a> (PDF 約8.0MB)</p> <p>●マイコン実習マニュアル(R8C/38A版) 1.14版 2011.05.30</p>	<p>「ルネサス統合開発環境 操作マニュアル(R8C/38A 版)」のDOWNLOAD をクリックして、マニュアルをダウンロードしてください。</p> <p>マニュアルに従って、</p> <ul style="list-style-type: none"> <li>・ルネサス統合開発環境のインストール</li> <li>・R8C Writer のインストール</li> <li>・ルネサス統合開発環境の操作についての習得を行ってください。</li> </ul>
--	---

## 4. マイコンへプログラムを書き込む

ここでは、RY-WRITER 基板を使って、R8C/M12A マイコンにプログラムを書き込む方法を説明します。

### 4.1 RY-WRITER 基板との接続

パソコン、RY-WRITER 基板、ブレッドボードのコネクタ(CN1)の接続図を、下図に示します。



4. マイコンへプログラムを書き込む

4.2 書き込み手順

1	<p>②書き込みスイッチを上にする</p> <p>①リセットスイッチを押しながら</p> <p>4 芯ケーブル</p> <p>RY-WRITER 基板</p> <p>USB ケーブル</p>	<p>①リセットスイッチを押しながら</p> <p>②書き込みスイッチを上にします。</p> <p>③リセットスイッチを離します。</p>
---	---	---

2	<p>R8C Writer</p>	<p>ルネサス統合開発環境の「ツール→R8C Writer」をクリックします。</p> <p>※「R8C Writer」が無い場合は、「ルネサス統合開発環境操作マニュアル (R8C/38A 版)」を参照して、R8C Writer を登録してください。</p>
---	-------------------	---

3	<p>①</p> <p>②</p>	<p>書き込みソフトが起動します。</p> <p>①「通信ポート」に「COM ○ : Prolific USB-to-Serial...」(○は番号です)と記載があるのを確認してください。無い場合はパソコンと RY-WRITER 基板との接続が正しくありません。接続を確認してください。</p> <p>②「書き込み開始」をクリックすると、書き込みを開始します。</p> <p>※書き込み後、ベリファイチェックする 「書き込み後、ベリファイチェックする」のチェックを付けると、書き込み後、書き込みデータが正しいか確認します。チェック ON を推奨します。</p> <p>※書き込み完了時、自動終了する 「書き込み完了時、自動終了する」のチェックを付けると、書き込み完了時に R8C Writer が自動終了します。</p>
---	-------------------	--

4. マイコンへプログラムを書き込む

4		<p>R8C Writer で書き込み中です。 書き込みが完了したら、終了をクリックして、R8C Writerを終了させてください。「書き込み完了時、自動終了する」のチェックが付いている場合は、自動で終了します。</p>
---	--	--

5		<p>もし、書き込みができなければ、左画面のようなエラー画面が出てきます。USB ケーブルやマイコンの状態を確認して、<b>OK</b>をクリック、再度書き込みを実行してください。</p>
---	--	--

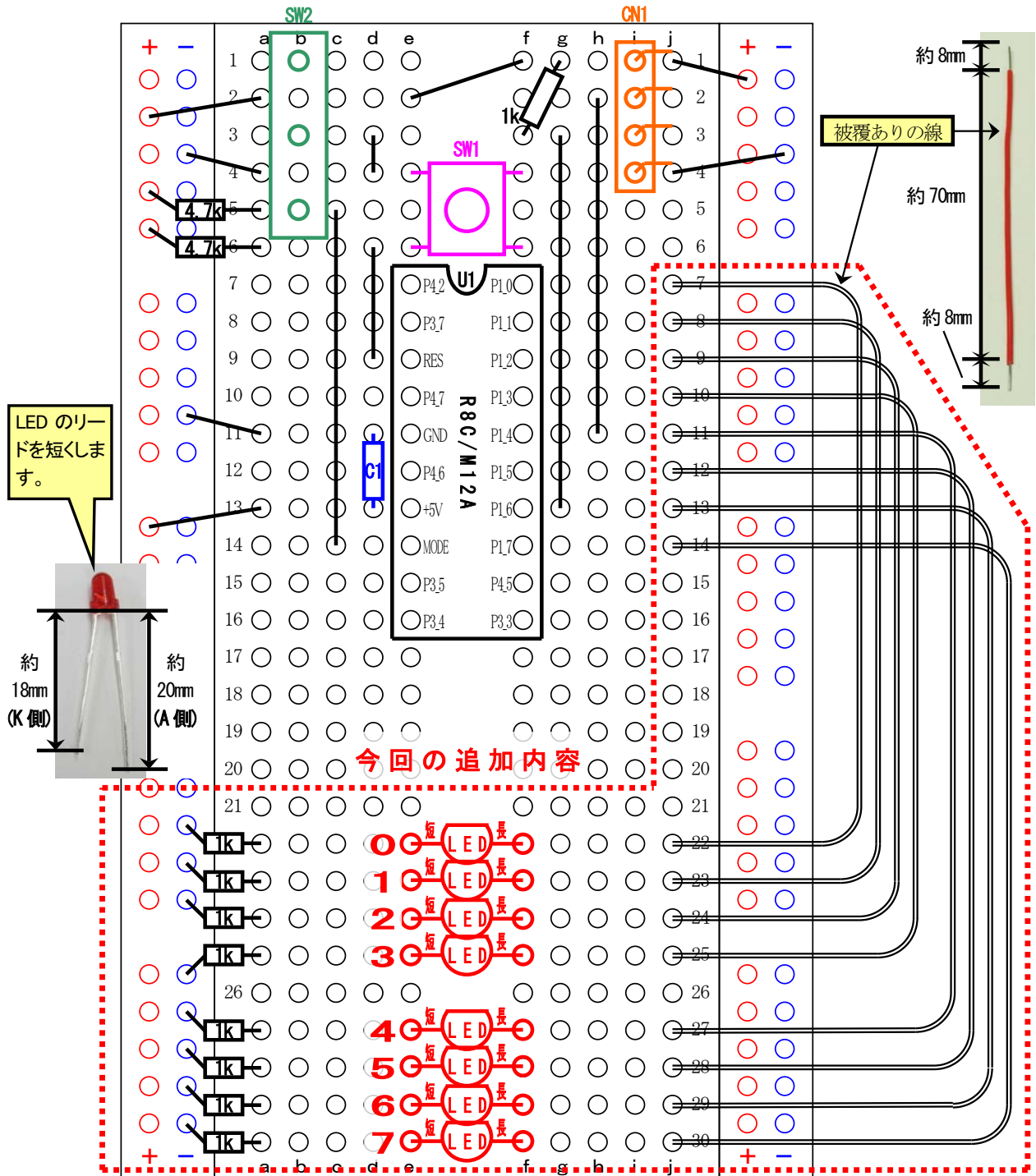
6		<p>書き込みが終わったら、次の手順で書き込んだプログラムを実行させます。 ①リセットスイッチを押しながら ②書き込みスイッチを下にします。 ③リセットスイッチを離します。</p>
---	--	--

## 5. LED の点灯(I/O ポートの出力)

### 5.1 概要

本章では、LED を 8 個、点灯させます。

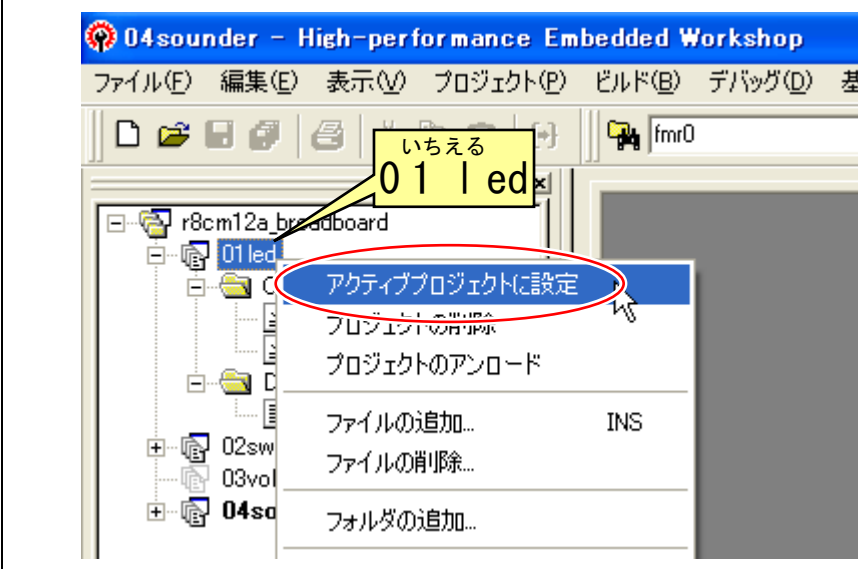
### 5.2 ブレッドボードの接続





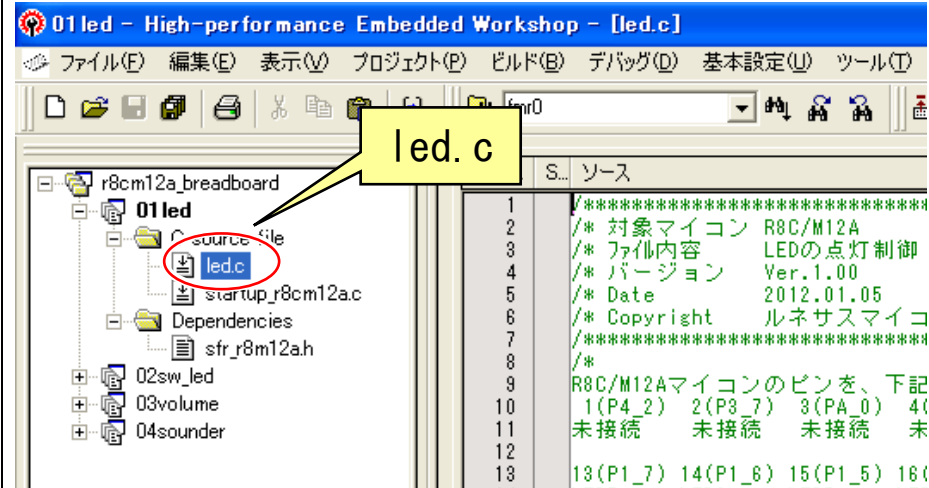
## 5.3 プロジェクト

1



「01led」で右クリック、「アクティブプロジェクトに設定」をクリックします。  
 ※「01led」が既に太字ならアクティブプロジェクトになっていますので、次へ進んでください。

2



「led.c」をダブルクリックして、プログラムを開きます。  
 「ビルド→ビルド」して、プログラムをMOTファイルに変換します。  
 「ビルド→R8C Writer」でプログラムを書き込みます。

```

1  /******
2  /* 対象マイコン R8C/M12A
3  /* ファイル内容 LEDの点灯制御
4  /* バージョン Ver.1.00
5  /* Date 2012.01.05
6  /* Copyright ルネサスマイコ
7  /******
8
9  R8C/M12Aマイコンのピンを、下記
10 1(P4_2) 2(P3_7) 3(PA_0) 4(
11 未接続 未接続 未接続 未
12
13 13(P1_7) 14(P1_6) 15(P1_5) 16(

```

## 5.4 操作方法

操作は特にありません。LED がどのように点灯するか確認してください。

## 5.5 プログラム

<pre> 1 : /***** 2 : /* 対象マイコン R8C/M12A */ 3 : /* ファイル内容 LED の点灯制御 */ 4 : /* バージョン Ver.1.00 */ 5 : /* Date 2012.03.01 */ 6 : /* Copyright ルネサスマイコンカーラーリー事務局 */ 7 : /***** 8 : /* 9 : R8C/M12A マイコンのピンを、下記のように結線します。 10 : 1(P4_2) 2(P3_7) 3(PA_0) 4(P4_7) 6(P4_6) 9(P3_5) 10(P3_4) 11(P3_3) 12(P4_5) 11 : 未接続 未接続 未接続 未接続 未接続 未接続 未接続 未接続 未接続 12 : 13 : 13(P1_7) 14(P1_6) 15(P1_5) 16(P1_4) 17(P1_3) 18(P1_2) 19(P1_1) 20(P1_0) 14 : LED7 LED6 LED5 LED4 LED3 LED2 LED1 LED0 15 : */ 16 : 17 : /*=====*/ 18 : /* インクルード */ 19 : /*=====*/ 20 : #include "sfr_r8m12a.h" /* R8C/M12A SFR の定義ファイル */ 21 : 22 : /*=====*/ 23 : /* プロトタイプ宣言 */ 24 : /*=====*/ 25 : void init( void ); 26 : void timer( unsigned long timer_set ); 27 : 28 : /***** 29 : /* メインプログラム */ 30 : /***** 31 : void main( void ) 32 : { 33 :     init(); /* 初期化 */ 34 : 35 :     while( 1 ) { 36 :         pd1 = 0b00001111; 37 :         timer( 1000 ); 38 :         pd1 = 0b11110000; 39 :         timer( 1000 ); 40 :         pd1 = 0b00000000; 41 :         timer( 1000 ); 42 :     } 43 : } 44 : 45 : /***** 46 : /* R8C/M12A スペシャルファンクションレジスタ(SFR)の初期化 */ 47 : /***** 48 : void init( void ) 49 : { 50 :     /* ポートの入出力設定 */ 51 :     pd1 = 0xff; /* ポート 1 の入出力設定 */ 52 :     pd3 = 0xff; /* ポート 3 の入出力設定 */ 53 :     pd4 = 0xff; /* ポート 4 の入出力設定 */ 54 :     pda = 0xff; /* ポート A の入出力設定 */ 55 : } 56 : 57 : /***** 58 : /* タイマ本体 */ 59 : /* 引数 タイマ値 1=1ms */ 60 : /***** 61 : void timer( unsigned long timer_set ) 62 : { 63 :     int i; 64 : 65 :     do { 66 :         for( i=0; i&lt;871; i++ ); 67 :     } while( timer_set-- ); 68 : } 69 : 70 : /***** 71 : /* end of file */ 72 : /***** </pre>	<p>20 行 R8C/M12A マイコンの内蔵周辺機能のレジスタを定義しているファイルです。R8C/M12A マイコンを使うときは必ず include します。</p> <p>33 行 init 関数へ移動して、R8C/M12A マイコンの内蔵周辺機能を初期化します。</p> <p>36、38、40 行 pd1(ポート1)にデータを出力します。</p> <p>37、39、41 行 timer 関数の引数の値の時間[ms]、この行で止まって時間稼ぎをします。</p> <p>51~54 行 pd1でポート1の端子を入力用にするか、出力用にするか決めます。pd3(ポート3)、pd4(ポート4)、pda(ポートA)も同様です。ビットを"1"にすると、設定したポートのビットが出力端子に、"0"にすると入力端子になります。未接続端子は出力にしておきます。 例) pd1 = 0x55 ポート1のbit6,4,2,0が出力端子、その他は入力端子</p> <p>66 行 この1行で1msの時間稼ぎをします。それを、timer_set 回、繰り返します。よって、timer_set[ms]の時間稼ぎとなります。</p>
---	---

## 5.6 課題

1. 次のパターンで点灯するプログラムを書きなさい。(1…点灯 0…消灯)

- ① 「1000 0001」を 0.5 秒点灯
- ② 「1100 0011」を 0.5 秒点灯
- ③ 「1110 0111」を 0.5 秒点灯
- ④ 「1111 1111」を 0.5 秒点灯
- ⑤ 「0000 0000」を 1 秒点灯
- ⑥ ①に戻って繰り返す

2. 次のパターンで点灯するプログラムを書きなさい。(1…点灯 0…消灯)

- ① 「0000 0001」を 0.2 秒点灯
- ② 「0000 0010」を 0.2 秒点灯
- ③ 「0000 0100」を 0.2 秒点灯
- ⋮
- ⑧ 「1000 0000」を 0.2 秒点灯
- ⑨ ①に戻って繰り返す

3. for 文の書式を調べなさい。

また、次のプログラムがどう動作するか、調べなさい。

```
void main( void )
{
    int i;                // 変数の宣言

    init();              // マイコン機能の初期化

    while( 1 ) {         // 無限ループ
        for( i=0; i<10; i++ ) { // 0~9 まで、10 回繰り返す
            pl = i;      // ポート 1 に変数 i の値を出力
            timer( 500 ); // 時間稼ぎ
        }               // for 文のカッコ閉じ
    }                   // while 文のカッコ閉じ
}
```

4. for 文を使って、0～255 をカウントするループを作り、カウント値を LED に出力するプログラムを作りなさい。出力する間隔は、0.2 秒とする。また、255 の次は 0 に戻って、出力し続けるものとする。

5. for 文を使って、2 番目の課題と同じ動作をするようにプログラムを作りなさい。

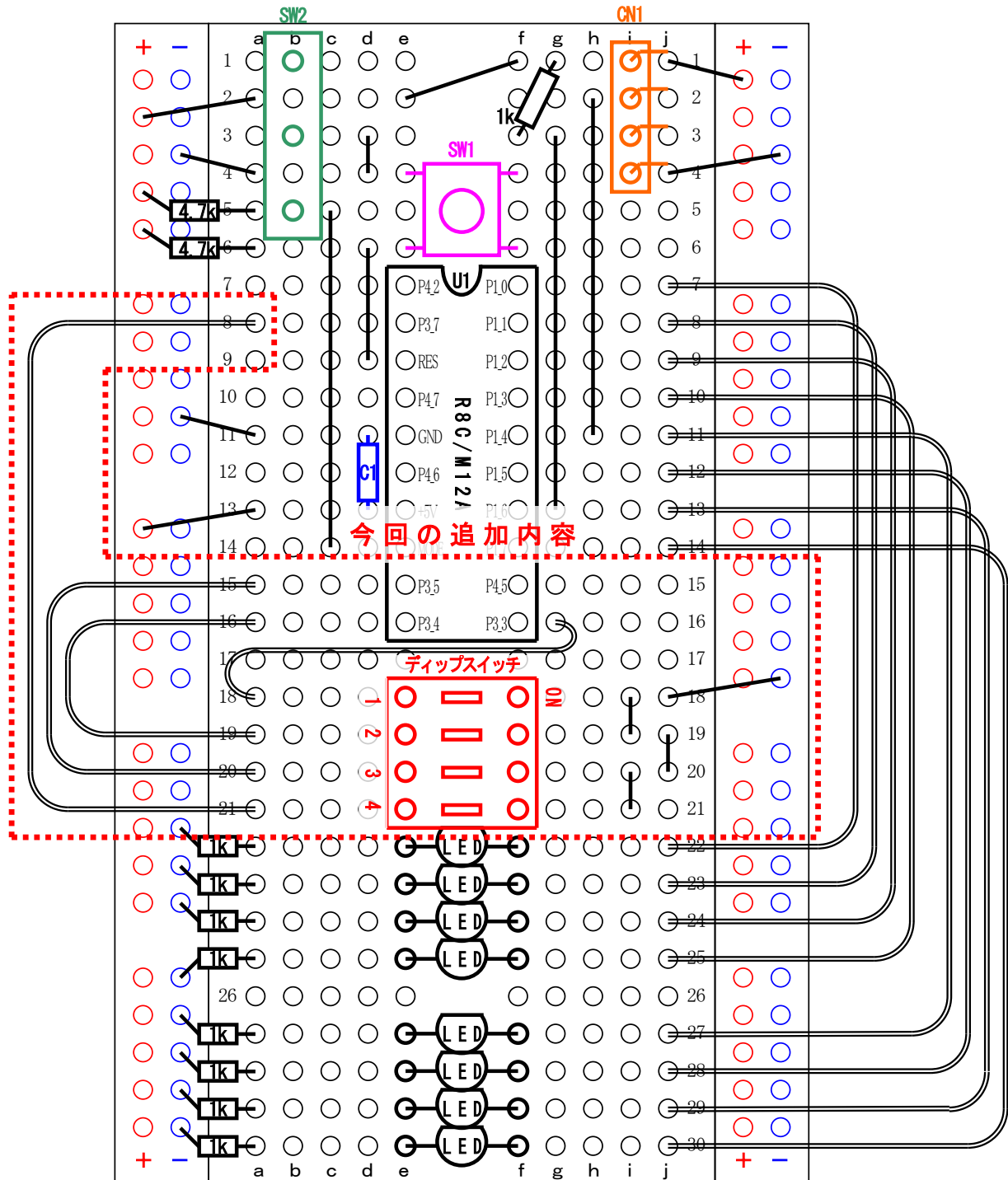
ヒント:10 進数にすると点灯する値は、1、2、4、8・・・となる

## 6. スイッチの入力(I/O ポートの入力)

### 6.1 概要

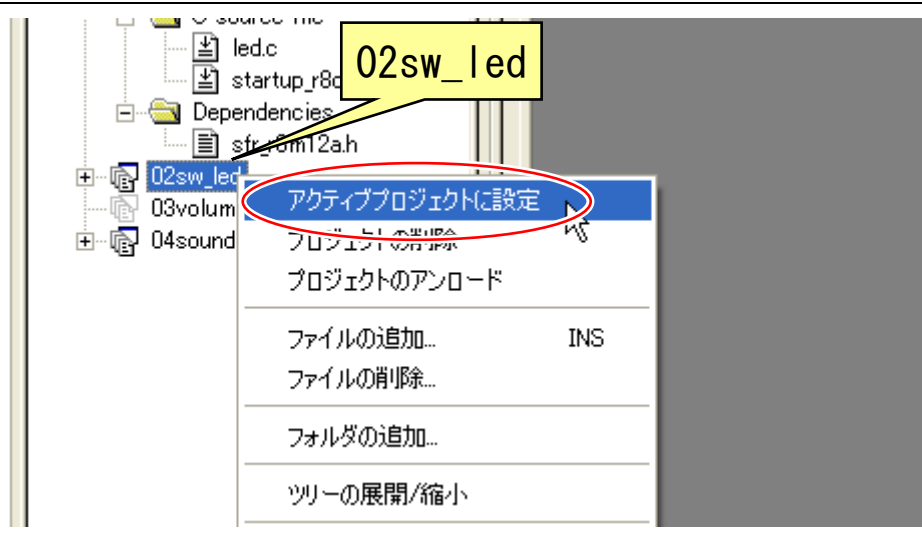
本章では、ディップスイッチ(4bit)の状態を入力、入力状態に応じてLEDを点灯させます。

### 6.2 ブレッドボードの接続



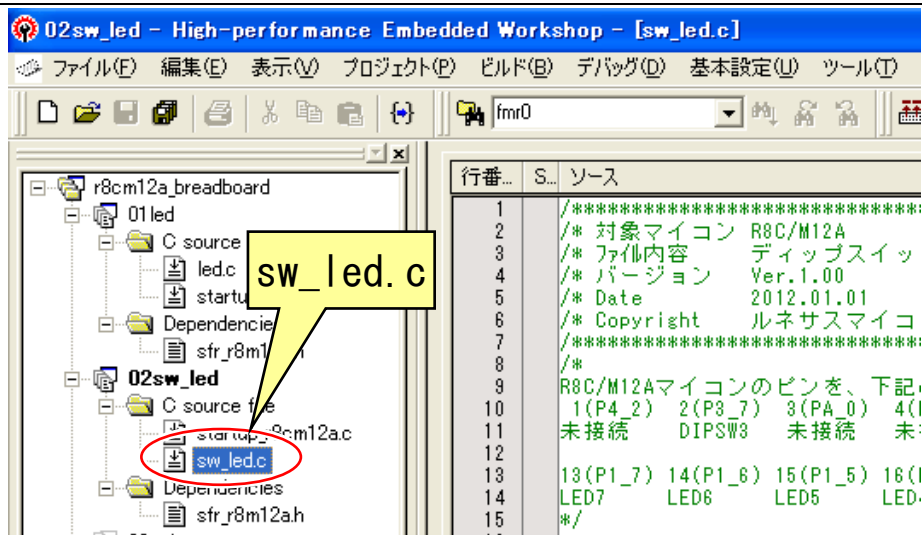
## 6.3 プロジェクト

1



「02sw\_led」で右クリック、「アクティブプロジェクトに設定」をクリックします。

2



「sw\_led.c」をダブルクリックして、プログラムを開きます。「ビルド→ビルド」して、プログラムをMOTファイルに変換します。「ビルド→R8C Writer」でプログラムを書き込みます。

```

1  /******
2  /* 対象マイコン R8C/M12A
3  /* ファイル内容 ディップスイッチ
4  /* バージョン Ver.1.00
5  /* Date 2012.01.01
6  /* Copyright ルネサスマイコ
7  /******
8  /*
9  R8C/M12Aマイコンのピンを、下記:
10 1(P4_2) 2(P3_7) 3(PA_0) 4(I
11 未接続 DIPSW3 未接続 未:
12
13 13(P1_7) 14(P1_6) 15(P1_5) 16(I
14 LED7 LED6 LED5 LED:
15 */

```

## 6.4 操作方法

ディップスイッチをON/OFFすると、LEDの付き方が変わります。どう変わるか、確認してください。

## 6. スイッチの入力(I/O ポートの入力)

## 6.5 プログラム

<pre> 1 : /***** 2 : /* 対象マイコン R8C/M12A */ 3 : /* ファイル内容 デイップスイッチの入力、LED の点灯制御 */ 4 : /* バージョン Ver.1.00 */ 5 : /* Date 2012.03.01 */ 6 : /* Copyright ルネサスマイコンカーラーリー事務局 */ 7 : /***** 8 : /* 9 : R8C/M12A マイコンのピンを、下記のように結線します。 10 : 1(P4_2) 2(P3_7) 3(PA_0) 4(P4_7) 6(P4_6) 9(P3_5) 10(P3_4) 11(P3_3) 12(P4_5) 11 : 未接続 DIPSW3 未接続 未接続 未接続 DIPSW2 DIPSW1 DIPSW0 未接続 12 : 13 : 13(P1_7) 14(P1_6) 15(P1_5) 16(P1_4) 17(P1_3) 18(P1_2) 19(P1_1) 20(P1_0) 14 : LED7 LED6 LED5 LED4 LED3 LED2 LED1 LED0 15 : */ 16 : 17 : /*=====*/ 18 : /* インクルード */ 19 : /*=====*/ 20 : #include "sfr_r8m12a.h" /* R8C/M12A SFR の定義ファイル */ 21 : 22 : /*=====*/ 23 : /* プロトタイプ宣言 */ 24 : /*=====*/ 25 : void init( void ); 26 : void timer( unsigned long timer_set ); 27 : 28 : /***** 29 : /* メインプログラム */ 30 : /***** 31 : void main( void ) 32 : { 33 :     init(); /* 初期化 */ 34 : 35 :     while( 1 ) { 36 :         p1 = p3; 37 :     } 38 : } 39 : 40 : /***** 41 : /* R8C/M12A スペシャルファンクションレジスタ(SFR)の初期化 */ 42 : /***** 43 : void init( void ) 44 : { 45 :     /* ポートの入出力設定 */ 46 :     pd1 = 0xff; /* ポート 1 の入出力設定 */ 47 :     pd3 = 0x47; /* ポート 3 の入出力設定 */ 48 :     pd4 = 0xff; /* ポート 4 の入出力設定 */ 49 :     pda = 0xff; /* ポート A の入出力設定 */ 50 : 51 :     /* 端子のプルアップ設定 */ 52 :     pur3 = 0xb8; /* ポート 3 のプルアップ設定 */ 53 : } 54 : 55 : /***** 56 : /* タイマ本体 */ 57 : /* 引数 タイマ値 1=1ms */ 58 : /***** 59 : void timer( unsigned long timer_set ) 60 : { 61 :     int i; 62 : 63 :     do { 64 :         for( i=0; i&lt;871; i++ ); 65 :     } while( timer_set-- ); 66 : } 67 : 68 : /***** 69 : /* end of file */ 70 : /***** </pre>	<p>20 行 R8C/M12A マイコンの内蔵周辺機能のレジスタを定義しているファイルです。R8C/M12A マイコンを使うときは必ず include します。</p> <p>33 行 init 関数へ移動して、R8C/M12A マイコンの内蔵周辺機能を初期化します。</p> <p>36 行 p1(ポート1)に p3(ポート3)の状態を出力します。p1 には LED が、p3 の bit7、bit5、bit4、bit3 にはデイップスイッチが接続されているので、4bit のデイップスイッチの値が LED に出力されます。接続されていないビットの LED がどう点灯するかも確認してください。</p> <p>47 行 p3 の bit7、bit5、bit4、bit3 はデイップスイッチの値を入力するので"0"にします。その他の端子は出力にしておきます。</p> <p>52 行 マイコンにはプルアップ抵抗が内蔵されています。pur3 で、p3 の入力端子を、内蔵されているプルアップ抵抗と接続します。ブレッドボード側でプルアップ抵抗を用意する必要はありません。</p>
--	--

## 6.6 課題

1. if 文の書式を調べなさい。

2. 次のプログラムを実行して、どのような動作になるか調べなさい。

```
void main( void )
{
    int i;                // 変数の宣言

    init();              // マイコン機能の初期化

    while( 1 ) {         // 無限ループ
        if( p3 == 0x08 ) { // p3(ディップスイッチ)の値は 0x08 か
            pl = 0x55;    // LED に 0x55 を出力
        } else {
            pl = 0xaa;    // LED に 0xaa を出力
        }                // if 文のカッコ閉じ
    }                    // while 文のカッコ閉じ
}
```

3. 次のプログラムを実行して、どのような動作になるか調べなさい。

```
void main( void )
{
    int i;                // 変数の宣言

    init();              // マイコン機能の初期化

    while( 1 ) {         // 無限ループ
        if( (p3 & 0x08) == 0x08 ) { // p3(ディップスイッチ)の値は 0x08 か
            pl = 0x55;    // LED に 0x55 を出力
        } else {
            pl = 0xaa;    // LED に 0xaa を出力
        }                // if 文のカッコ閉じ
    }                    // while 文のカッコ閉じ
}
```

## 6. スイッチの入力(I/O ポートの入力)

4. 下表のように、ポート3のbit5とbit4の状態によりLEDが点灯するよう、プログラムを作りなさい。なお、ポート3のbit5とbit4の値以外は、“1”か“0”か分からないものとする。

ポート3 bit5 の値	ポート3 bit4 の値	このときの LED の付き方
0	0	00000011
0	1	00001111
1	0	00111111
1	1	11111111

5. 下表のようにポート3の特定のビットの状態を、ポート1の特定のビットに出力するプログラムを作りなさい。なお、書かれている端子以外の状態は分からないものとする。また、書かれている端子以外の状態は変えないものとする。

読み込む端子	出力する端子
ポート3のbit7	ポート1のbit7
ポート3のbit5	ポート1のbit5
ポート3のbit4	ポート1のbit3
ポート3のbit3	ポート1のbit1

### ■ポイント—ビット単位で入出力する

p1、p3などのレジスタは、ポート単位(8bit単位)で入力や出力を行います。次のように、1端子ごとに端子の状態を読み込んだり、値を出力をすることができます。

```
p1_5 = 1;           // ポート1のbit5を"1"にする

if( p3_3 == 1 ) {  // ポート3のbit3の状態をチェック
    プログラム
}

```

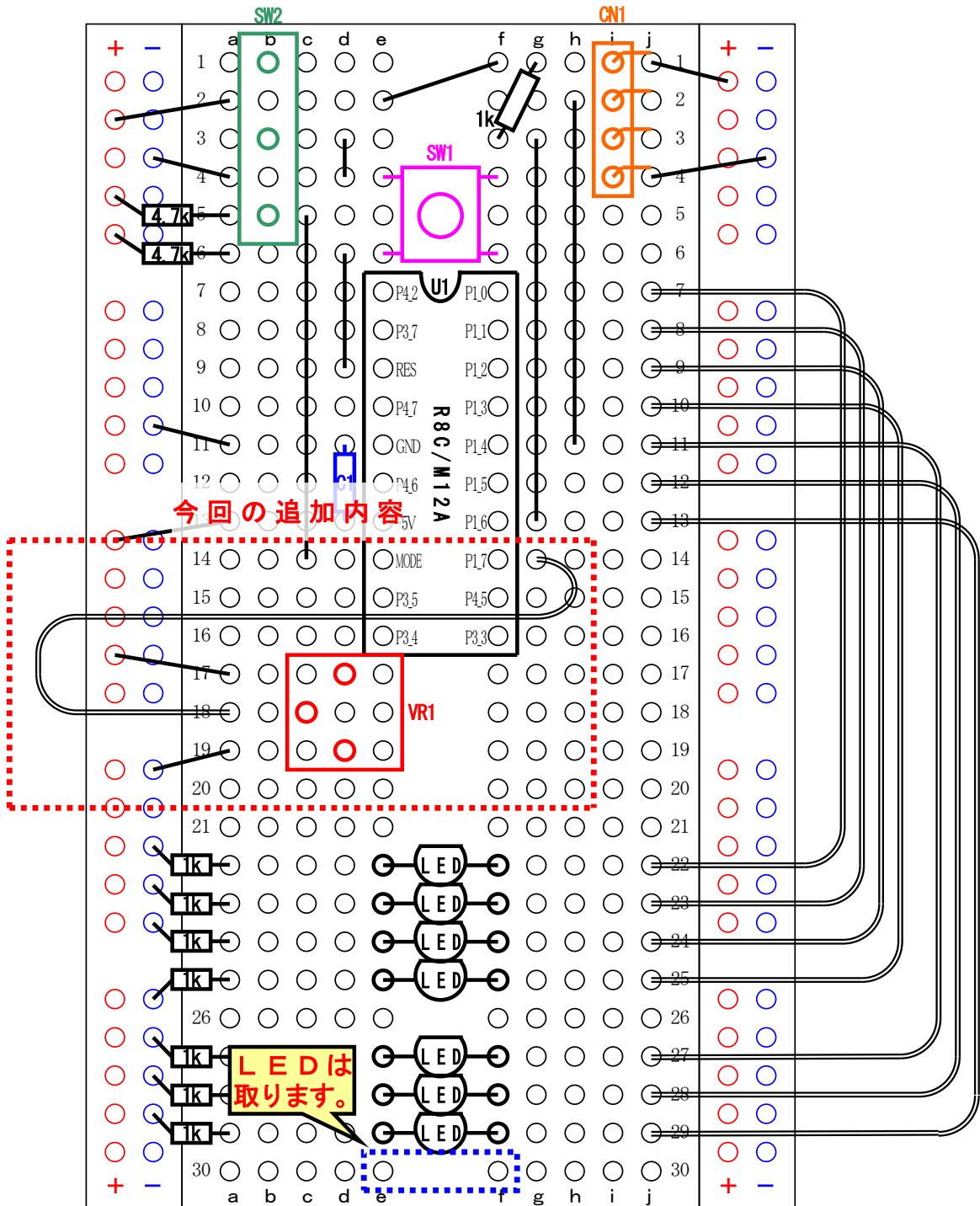


## 7. ボリューム(0~5V の電圧)の入力(A/D 変換器)

### 7.1 概要

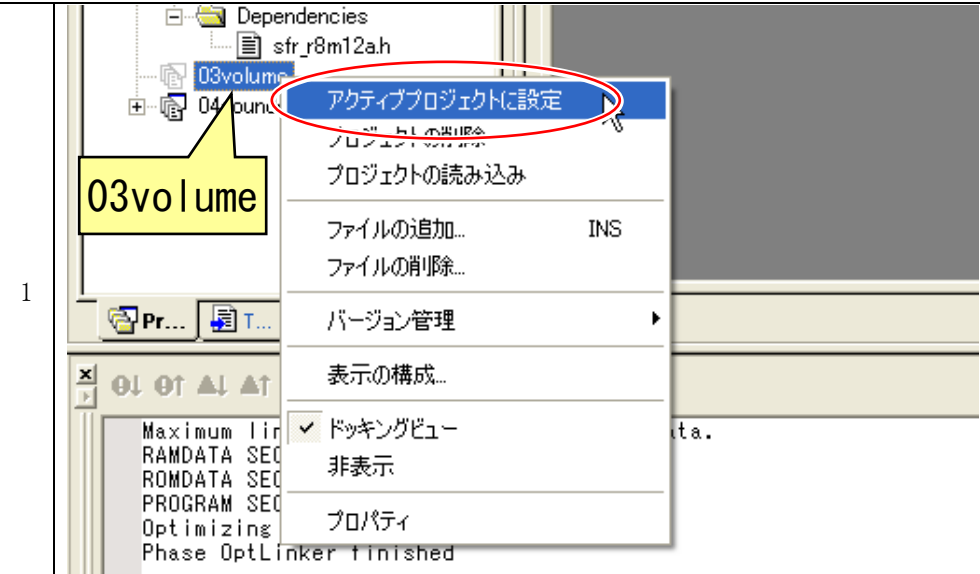
本章では、ボリュームから出力された 0~5V の電圧をマイコンで A/D 変換してデジタル値に変換します。

### 7.2 ブレッドボードの接続



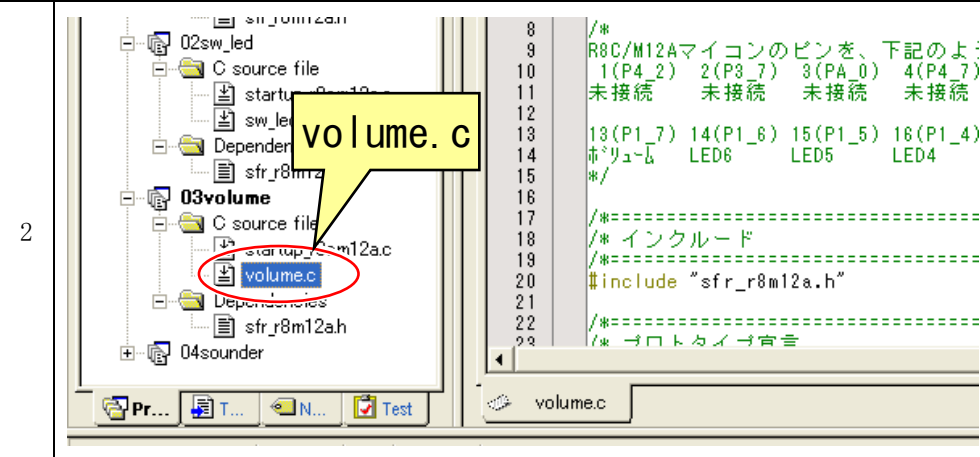
## 7.3 プロジェクト

1



「03volume」で右クリック、「アクティブプロジェクトに設定」をクリックします。

2



「volume.c」をダブルクリックして、プログラムを開きます。

「ビルド→ビルド」して、プログラムをMOTファイルに変換します。

「ビルド→R8C Writer」でプログラムを書き込みます。

```

8  /*
9  R8C/M12Aマイコンのピンを、下記のよ
10  1(P4_2) 2(P3_7) 3(PA_0) 4(P4_7)
11  未接続 未接続 未接続 未接続
12
13  13(P1_7) 14(P1_6) 15(P1_5) 16(P1_4)
14  ボリューム LED6 LED5 LED4
15  */
16
17  /*=====
18  /* インクルード
19  /*=====
20  #include "sfr_r8m12a.h"
21
22  /*=====
23  /* プロトタイプ宣言

```

## 7.4 操作方法

ボリュームを回すと、LED の付き方が変わります。どう変わるか、確認してください。

7. ボリューム(0~5V の電圧)の入力(A/D 変換器)

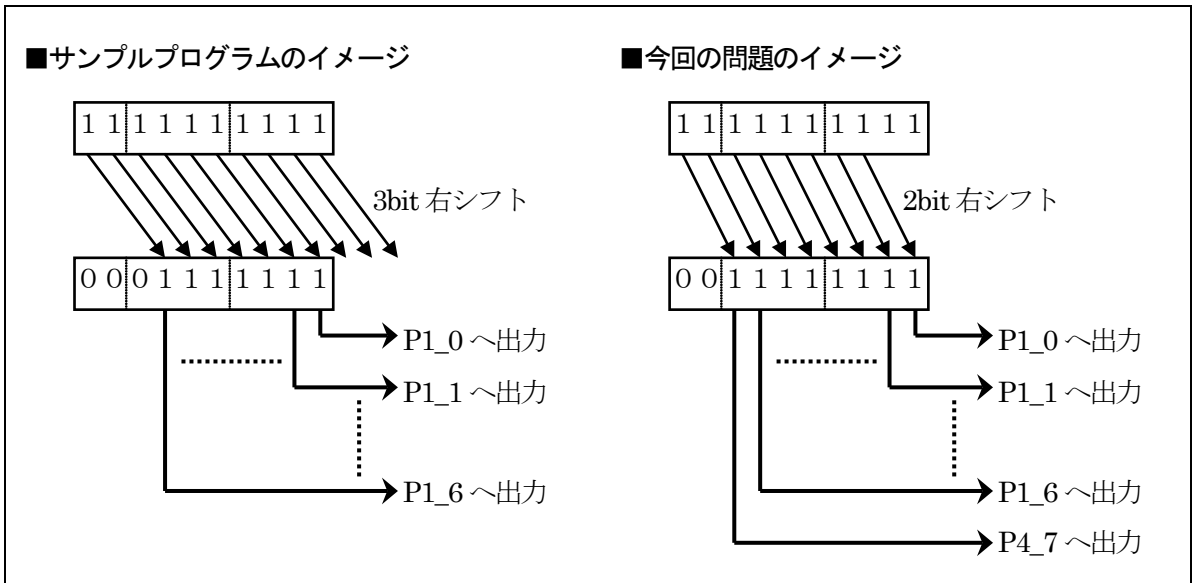
7.5 プログラム

<pre> 1 : /****** 2 : /* 対象マイコン R8C/M12A */ 3 : /* ファイル内容 ボリュームを使った A/D 変換 */ 4 : /* バージョン Ver.1.00 */ 5 : /* Date 2012.03.01 */ 6 : /* Copyright ルネサスマイコンカーラーリ事務局 */ 7 : /****** 8 : /* 9 : R8C/M12A マイコンのピンを、下記のように接続します。 10 : 1(P4_2) 2(P3_7) 3(PA_0) 4(P4_7) 6(P4_6) 9(P3_5) 10(P3_4) 11(P3_3) 12(P4_5) 11 : 未接続 未接続 未接続 未接続 未接続 未接続 未接続 未接続 未接続 12 : 13 : 13(P1_7) 14(P1_6) 15(P1_5) 16(P1_4) 17(P1_3) 18(P1_2) 19(P1_1) 20(P1_0) 14 : ボリューム LED6 LED5 LED4 LED3 LED2 LED1 LED0 15 : */ 16 : 17 : /*=====*/ 18 : /* インクルード */ 19 : /*=====*/ 20 : #include "sfr_r8m12a.h" /* R8C/M12A SFR の定義ファイル */ 21 : 22 : /*=====*/ 23 : /* プロトタイプ宣言 */ 24 : /*=====*/ 25 : void init( void ); 26 : void timer( unsigned long timer_set ); 27 : int get_ad7( void ); 28 : 29 : /****** 30 : /* メインプログラム */ 31 : /****** 32 : void main( void ) 33 : { 34 :     int ad; 35 : 36 :     init(); /* 初期化 */ 37 : 38 :     while( 1 ) { 39 :         ad = get_ad7(); /* ad = 0~1023 */ 40 :         ad = ad &gt;&gt; 3; /* ad = 0~127 にする */ 41 :         pl = ad; 42 :     } 43 : } 44 : 45 : /****** 46 : /* R8C/M12A スペシャルファンクションレジスタ(SFR)の初期化 */ 47 : /****** 48 : void init( void ) 49 : { 50 :     /* ポートの入出力設定 */ 51 :     pd1 = 0x7f; /* ポート 1 の入出力設定 */ 52 :     pd3 = 0xff; /* ポート 3 の入出力設定 */ 53 :     pd4 = 0xff; /* ポート 4 の入出力設定 */ 54 :     pda = 0xff; /* ポート A の入出力設定 */ 55 : } 56 : 57 : /****** 58 : /* タイマ本体 */ 59 : /* 引数 タイマ値 1=1ms */ 60 : /****** 61 : void timer( unsigned long timer_set ) 62 : { 63 :     int i; 64 : 65 :     do { 66 :         for( i=0; i&lt;871; i++ ); 67 :     } while( timer_set-- ); 68 : } 69 : 70 : /****** 71 : /* P1_7 (AN7) 端子を A/D 変換 */ 72 : /* 戻り値 A/D 変換値(0~1023) */ 73 : /****** 74 : int get_ad7( void ) 75 : { 76 :     int i; 77 : 78 :     /* A/D コンバータの設定 */ 79 :     mstad = 0; /* A/D コンバータを有効にする */ 80 :     admod = 0x03; /* 単発モードに設定 */ 81 :     adinsel = 0x81; /* 入力端子 AN7 (P1_7) を選択 */ 82 :     adcon0 = 0x01; /* A/D 変換スタート */ 83 : 84 :     while( adcon0 &amp; 0x01 ); /* A/D 変換終了待ち */ 85 :     i = adl; /* A/D 値取得(0~1023) */ 86 : 87 :     return i; 88 : } 89 : 90 : /****** 91 : /* end of file */ 92 : /****** </pre>	<p>20 行 R8C/M12A マイコンの内蔵周辺機能のレジスタを定義しているファイルです。R8C/M12A マイコンを使うときは必ず include します。</p> <p>36 行 init 関数へ移動して、R8C/M12A マイコンの内蔵周辺機能を初期化します。</p> <p>39~41 行 AN7 端子(P1_7)から電圧を入力し、0~5V を 0~1023 に変換します。その値を ad 変数に代入します。次に ad 変数を 3 ビット右シフトして、0~1023(0b11 1111 1111)→0~127(0b00 0111 1111)の値に変換します。最後に ad 変数の値を pl に代入して、7 個の LED に値を出力します。</p> <p>51 行 pl(ポート 1)の bit7 はボリュームの電圧入力なので、入力端子にします。</p> <p>74~88 行 マイコン内蔵周辺機能である、A/D 変換器を使って、P1_7 端子の電圧 0~5V を 0~1023 の値に変換します。A/D 変換値が 511 なら 電圧=5×A/D 変換値÷1023 =5×511×1023 ÷ 1023 と、2.50V が入力されていることが分かります。</p> <p>81 行 プログラムを変えることによって、P1_0、P1_1、P1_2、P1_3、P1_4 端子の電圧を A/D 変換することもできます。 P1_7 端子にする…0x81 (今回) P1_4 端子にする…0x80 P1_3 端子にする…0x11 P1_2 端子にする…0x10 P1_1 端子にする…0x01 P1_0 端子にする…0x00</p> <p>A/D 変換する端子を変えた場合、85 行の adl も変えます。 P1_7 端子…ad1 (今回) P1_4 端子…ad0 P1_3 端子…ad1 P1_2 端子…ad0 P1_1 端子…ad1 P1_0 端子…ad0</p>
--	--

### 7.6 課題

1. 右ビットシフト(>>)、左ビットシフト(<<))について説明しなさい。

2. P4\_7 端子に 8 個目の LED を追加して、A/D 変換値を 2 ビット右シフトさせ 0~255 に変換したときに、0~255 の値を、P4\_7、P1\_6~P1\_0 端子に接続している 8 個の LED に出力するプログラムを作りなさい。



3. A/D 変換値の値を 7 ビット右シフトさせて 0~7 に変換したとき、次の表のように 8 個の LED が点灯するようプログラムを作りなさい。LED は課題の 2 番で接続した、8 個の状態とする。  
 ※ビットシフト命令はここでは使わないこと。

A/D 変換値	LED の付き方
0	0000 0001
1	0000 0011
2	0000 0111
3	0000 1111
4	0001 1111
5	0011 1111
6	0111 1111
7	1111 1111

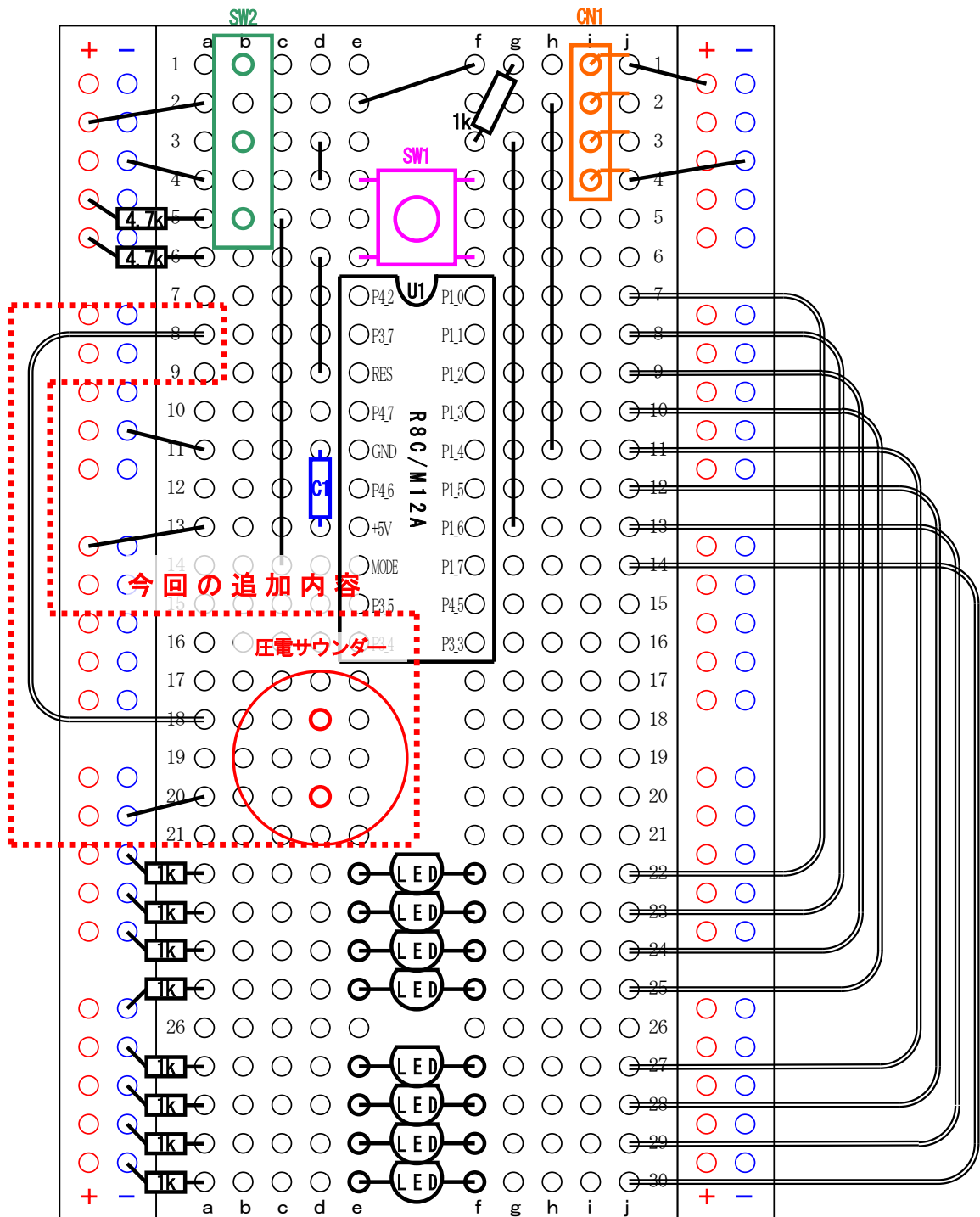
4. 3 の問題をビットシフト命令(>>、または<<))を使って、プログラムしなさい。  
 ※ヒント:0xff の値を、右シフトさせていきます

## 8. 圧電サウンダーから音を鳴らす(PWM 波形の出力)

### 8.1 概要

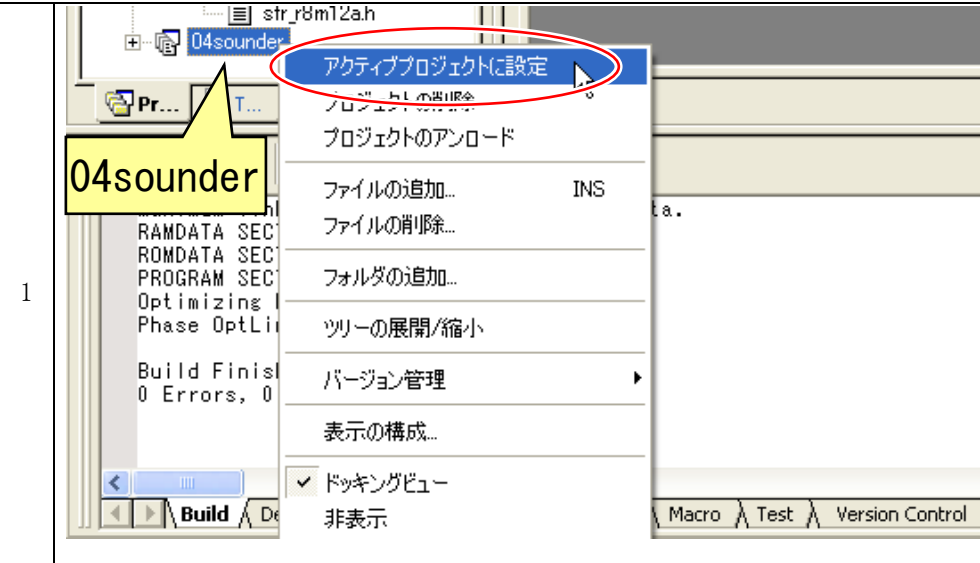
本章では、圧電サウンダーから音を鳴らします。

### 8.2 ブレッドボードの接続



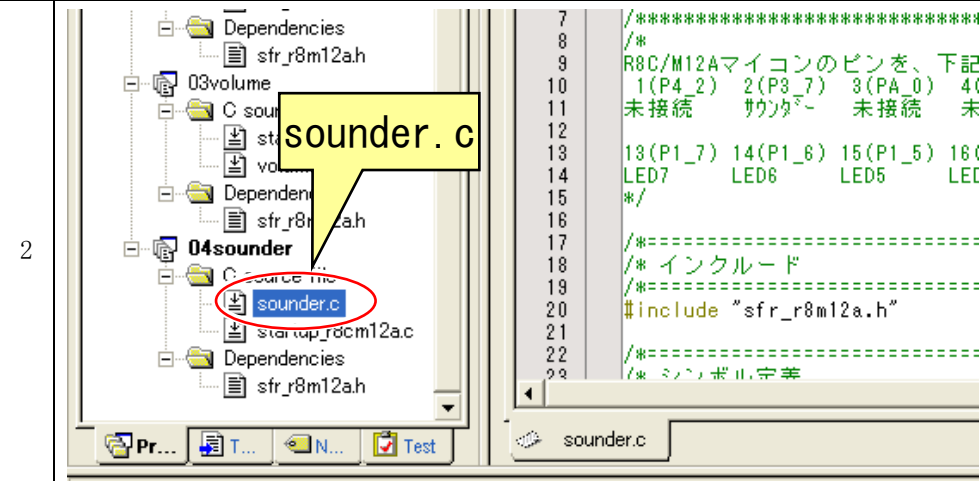
### 8.3 プロジェクト

1



「04sounder」で右クリック、「アクティブプロジェクトに設定」をクリックします。

2



```
7 //*****  
8 /*  
9 R8C/M12Aマイコンのピンを、下記  
10 1(P4_2) 2(P3_7) 3(PA_0) 4(  
11 未接続 サウンダー 未接続 未  
12  
13 13(P1_7) 14(P1_6) 15(P1_5) 16(  
14 LED7 LED6 LED5 LEC  
15 */  
16  
17 /*-----  
18 /* インクルード  
19 /*-----  
20 #include "sfr_r8m12a.h"  
21  
22 /*-----  
23 /* シンボル定義
```

「sounder.c」をダブルクリックして、プログラムを開きます。  
「ビルド→ビルド」して、プログラムをMOTファイルに変換します。  
「ビルド→R8C Writer」でプログラムを書き込みます。

### 8.4 操作方法

操作は特にありません。圧電サウンダーから鳴る音を確認してください。

### 8.5 プログラム

<pre> 1 : /****** */ 2 : /* 対象マイコン R8C/M12A */ 3 : /* ファイル内容 LEDの点灯制御 */ 4 : /* バージョン Ver.1.00 */ 5 : /* Date 2012.03.01 */ 6 : /* Copyright ルネサスマイコンカーラーリ事務局 */ 7 : /****** */ 8 : /* 9 : R8C/M12Aマイコンのピンを、下記のように結線します。 10 : 1(P4_2) 2(P3_7) 3(PA_0) 4(P4_7) 6(P4_6) 9(P3_5) 10(P3_4) 11(P3_3) 12(P4_5) 11 : 未接続 サウンダー 未接続 未接続 未接続 未接続 未接続 未接続 12 : 13 : 13(P1_7) 14(P1_6) 15(P1_5) 16(P1_4) 17(P1_3) 18(P1_2) 19(P1_1) 20(P1_0) 14 : LED7 LED6 LED5 LED4 LED3 LED2 LED1 LED0 15 : */ 16 : 17 : /*===== */ 18 : /* インクルード */ 19 : /*===== */ 20 : #include "sfr_r8m12a.h" /* R8C/M12A SFRの定義ファイル */ 21 : 22 : /*===== */ 23 : /* シンボル定義 */ 24 : /*===== */ 25 : 26 : /* 圧電サウンダー用 */ 27 : #define BEEP_500Hz 10000 /* 500Hz : (1/ 500)/(1/(20M/4)) */ 28 : #define BEEP_1000Hz 5000 /* 1000Hz: (1/1000)/(1/(20M/4)) */ 29 : 30 : /* 3オクターブ目の音階 */ 31 : #define DO_3 38226 /* ド */ 32 : #define DO#_3 36081 /* ド# */ 33 : #define RE_3 34056 /* レ */ 34 : #define RE#_3 32144 /* レ# */ 35 : #define MI_3 30340 /* ミ */ 36 : #define FA_3 28637 /* ファ */ 37 : #define FA#_3 27030 /* ファ# */ 38 : #define SO_3 25513 /* ソ */ 39 : #define SO#_3 24081 /* ソ# */ 40 : #define RA_3 22729 /* ラ */ 41 : #define RA#_3 21454 /* ラ# */ 42 : #define SI_3 20250 /* シ </pre> <p>中略</p> <pre> 72 : /*===== */ 73 : /* プロトタイプ宣言 */ 74 : /*===== */ 75 : void init( void ); 76 : void timer( unsigned long timer_set ); 77 : void sounder( unsigned int tone, int time ); 78 : 79 : /*===== */ 80 : /* グローバル変数の宣言 */ 81 : /*===== */ 82 : int tempo = 60; /* テンポ */ 83 : 84 : /****** */ 85 : /* メインプログラム */ 86 : /****** */ 87 : void main( void ) 88 : { 89 :     init(); /* 初期化 */ 90 : 91 :     /* 92 :     長さは、四分音符を4として、二分音符は8、八分音符は2となります。 93 :     休符も同様に、四分休符を4として、二部休符は8、八分休符は2となります。 94 :     */ 95 :     /* 音階, 長さ */ 96 :     sounder( DO_4, 4 ); 97 :     sounder( RE_4, 4 ); 98 :     sounder( MI_4, 4 ); 99 :     sounder( FA_4, 4 ); 100 :    sounder( SO_4, 4 ); 101 :    sounder( RA_4, 4 ); 102 :    sounder( SI_4, 4 ); 103 :    sounder( DO_5, 4 ); 104 :    sounder( 0, 0 ); 105 :    while( 1 ); 106 : 107 :    // 大きな古時計 108 :    /* 音階, 長さ */ 109 :    sounder( RE_4, 4 ); // お 110 :    sounder( SO_4, 4 ); // お 111 :    sounder( FA#_4, 2 ); // き 112 :    sounder( SO_4, 2 ); // な 113 :    sounder( RA_4, 4 ); // のっ 114 :    sounder( SO_4, 2 ); // ぼ 115 :    sounder( RA_4, 2 ); // の 116 :    sounder( SI_4, 2 ); // ふ 117 :    sounder( SI_4, 2 ); // る 118 :    sounder( DO_5, 2 ); // ど 119 :    sounder( SI_4, 2 ); // け 120 :    sounder( MI_4, 4 ); // い 121 :    sounder( RA_4, 2 ); // お 122 :    sounder( RA_4, 2 ); // じ 123 :    sounder( SO_4, 4 ); // い </pre>	<p>20 行 R8C/M12A マイコンの内蔵周辺機能のレジスタを定義しているファイルです。R8C/M12A マイコンを使うときは必ず include します。</p> <p>26～70 行 圧電サウンダーの音を鳴らすときの、設定値を定義しています。 音階は(ド、レ、ミ、ファ・・・など)、計算すると周波数が分かりますので、その周波数に応じて設定値を計算し、定義しています。 3オクターブ目のド～シを定義しています。</p> <p>82 行 temp 変数にテンポを設定します。 テンポとは、1 分間に四分音符(四分休符)を何回演奏するかという早さです。大きいと早く、小さいと遅くなります。</p> <p>89 行 init 関数へ移動して、R8C/M12A マイコンの内蔵周辺機能を初期化します。</p> <p>96～245 行 sounder 関数で、圧電サウンダーに音を鳴らします。 長さは、四分音符を 4として、二分音符は 8、八分音符は 2とします。休符も同様に、四分休符を 4として、二部休符は 8、八分休符は 2とします。関数の音階部分は、定義した音をセットします。長さは音符(休符)に応じて、設定します。</p> <p>109～245 行 大きな古時計の音階データです。 大きな古時計を鳴らす場合は、96～105 行を削除してください。</p>
--	---

## 8. 圧電サウンダーから音を鳴らす(PWM 波形の出力)

<pre> 124 :    sounder( SO_4, 2 ); // さ 125 :    sounder( SO_4, 2 ); // ん 126 :    sounder( FAU_4, 4 ); // の 127 :    sounder( MI_4, 2 ); // と 128 :    sounder( FAU_4, 2 ); // け 129 :    sounder( SO_4, 10 ); // い 130 :    sounder( 0, 2 );  中略  246 : } 247 : 248 : /***** 249 : /* R8C/M12A スペシャルファンクションレジスタ(SFR)の初期化 */ 250 : /***** 251 : void init( void ) 252 : { 253 :     /* ポートの入出力設定 */ 254 :     pd1 = 0xff;          /* ポート1の入出力設定 */ 255 :     pd3 = 0xff;          /* ポート3の入出力設定 */ 256 :     pd4 = 0xff;          /* ポート4の入出力設定 */ 257 :     pda = 0xff;          /* ポートAの入出力設定 */ 258 : 259 :     /* タイマRJ2の設定 */ 260 :     p37sel1 = 1;        /* P3_7をTRJ0端子にする */ 261 :     p37sel0 = 0;        /*  */ 262 :     msttrj = 0;         /* タイマRJ2を有効にする */ 263 :     trjmr = 0x31;       /* パルス出力モードに設定 */ 264 :     trj = 0;            /* TRJ×1/20M*2の時間でON/OFFする*/ 265 :     //tstart_trjcr = 1; /* タイムスタート(今回はまだしない) */ 266 : } 267 : 268 : /***** 269 : /* タイマ本体 */ 270 : /* 引数 タイマ値 l=1ms */ 271 : /***** 272 : void timer( unsigned long timer_set ) 273 : { 274 :     int i; 275 : 276 :     do { 277 :         for( i=0; i&lt;871; i++ ); 278 :     } while( timer_set-- ); 279 : } 280 : 281 : /***** 282 : /* 圧電サウンダーを鳴らす */ 283 : /* 引数 音階、長さ */ 284 : /***** 285 : void sounder( unsigned int tone, int time ) 286 : { 287 :     if( tone == 0 ) { 288 :         tstart_trjcr = 0;          /* タイマ停止 */ 289 :     } else { 290 :         trj = tone; 291 :         tstart_trjcr = 1;          /* タイマ開始 */ 292 :     } 293 :     if( time != 0 ) { 294 :         timer( 15000L * time / tempo ); 295 :     } 296 : } 297 : 298 : /***** 299 : /* end of file */ 300 : /***** </pre>	<p>260～264 行 R8C/M12A マイコン内蔵周辺機能のタイマ RJ2 を使って、圧電サウンダーを鳴らす音階(パルス)を作ります。</p> <p>260～261 行 パルス出力端子は、p1_6 端子か、p3_7 端子を選べます。 p3_7 端子から出力する場合は 260～261 行を下記にします。 260 : p37sel1 = 1; 261 : p37sel0 = 0; p1_6 端子から出力する場合は 260～261 行を下記にします。 260 : p16sel1 = 1; 261 : p16sel0 = 0;</p> <p>264 行 trj には、波形の ON 幅、OFF 幅を設定します。 例えば、4 オクターブ目のドの音は、周期 3.82ms の波形です。ON 幅は、周期の 1/2 なので 1.91ms となります。 trj=タイマ RJ のカウントアップする時間÷ON 幅 = 1.91ms ÷ 100ns = 19113 となります。</p> <p>288 行 音階が 0 なら出力停止します。</p> <p>290～291 行 音階が 0 以外なら、trj に値をセットして、タイマ RJ の出力を開始します。</p> <p>294 行 長さが 0 以外なら、長さ分、タイマを実行して時間稼ぎをします。</p>
--	---

## 8.6 課題

1. 「大きな古時計」を演奏するよう、プログラムを改造しなさい。
2. チャルメラの音楽(ドレミ～レド、ドレミレドレ～)を演奏しなさい。
3. R8C/M12A マイコンと圧電サウンダーの接続を P1\_6 端子にして、音楽が鳴るようプログラムを改造しなさい。  
※P1\_6 端子につながっている LED は外してよい。