

お客様各位

カタログ等資料中の旧社名の扱いについて

2010年4月1日を以ってNECエレクトロニクス株式会社及び株式会社ルネサステクノロジが合併し、両社の全ての事業が当社に承継されております。従いまして、本資料中には旧社名での表記が残っておりますが、当社の資料として有効ですので、ご理解の程宜しくお願ひ申し上げます。

ルネサスエレクトロニクス ホームページ (<http://www.renesas.com>)

2010年4月1日

ルネサスエレクトロニクス株式会社

【発行】ルネサスエレクトロニクス株式会社 (<http://www.renesas.com>)

【問い合わせ先】 <http://japan.renesas.com/inquiry>

ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りが無いことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。
標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）
特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害などを生じさせないようお客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、詳細につきましては製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

注 1. 本資料において使用されている「当社」とは、ルネサスエレクトロニクス株式会社およびルネサスエレクトロニクス株式会社とその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。

H8/300L Super Low Power シリーズ

PWM タイマを使用したサイン波

要旨

単一周波数での PWM タイマを用いて、256 サンプルングポイントのサイン波を生成する方法を示します。

動作確認デバイス

H8/38024

目次

1. 概要	2
2. PWM タイマアーキテクチャ.....	3
3. サイン波生成の理論.....	8
4. プログラムの概要.....	9
5. プログラムのフローチャート	10
6. プログラムリスト.....	11
参考文献	14

1. 概要

パルス幅変調 (PWM) とは、信号変調の一方式で、データを総時間に対するオン時間の割合 (デューティサイクルと呼びます) で表します。PWM では、瞬間的な DC 成分はデューティサイクルに直接比例します。主に、振幅の変化するアナログ信号をデジタルにエンコードした信号の制御に用います。

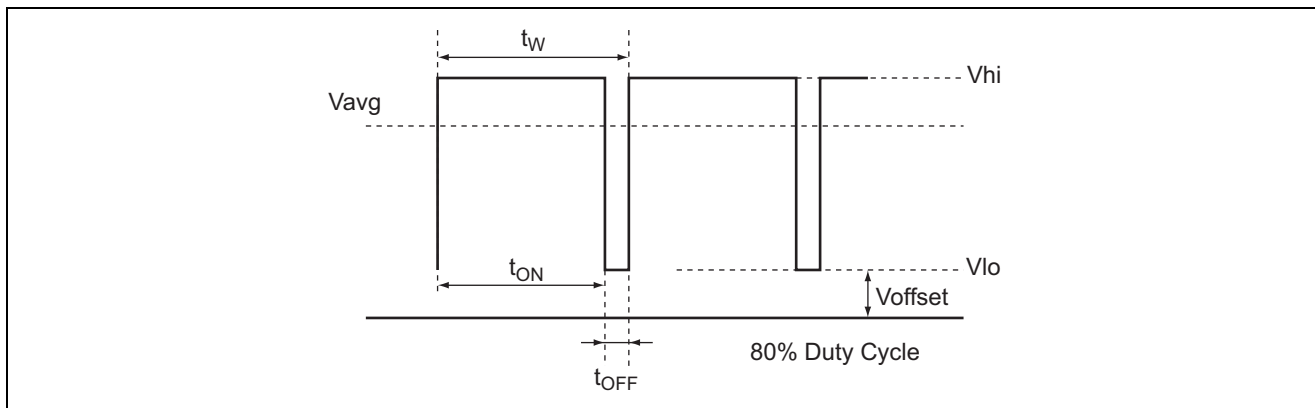


図 1 PWM のデューティサイクルと周波数

時間平均電圧 (V_{avg})、方形波のハイレベルとローレベルの電圧 (V_{hi} と V_{lo})、パーセントで表したデューティサイクル (D) の関係を以下に示します。

$$V_{avg} = (V_{hi} - V_{lo}) \times D + V_{offset} \dots\dots\dots [1]$$

$$\text{ここで, } D = t_{ON} / t_W \text{ すなわち } D = \text{PWDR 値} / 1023 \text{ (10 ビット PWM タイマの場合)} \dots\dots\dots [2]$$

$V_{hi} = 5V$, $V_{lo} = 0V$ (すなわち $V_{offset} = 0$)、 $D = 80\%$ の場合、 $V_{avg} = 4V$ となります。

このほかに、PWM タイマの重要なパラメータとして周波数 (f) があります。周波数は、1 秒あたりのパルス数で表します。

$$f = 1 / t_W \dots\dots\dots [3]$$

$$\text{ここで, } t_W = t_{ON} + t_{OFF} \dots\dots\dots [4]$$

式[2]と[3]から、以下が証明されます。

$$f = D / t_{ON} \dots\dots\dots [5]$$

2. PWM タイマアーキテクチャ

H8/38024 グループマイクロコンピュータは、10 ビットの PWM タイマを 2 つ内蔵しています (PWM1 と PWM2)。4 種類の変換周期 ($4096/\phi$, $2048/\phi$, $1024/\phi$, $512/\phi$) があり、リップルの少ないパルス分割方法を採用し、モジュールスタンバイモードで電力消費を抑えることができます。

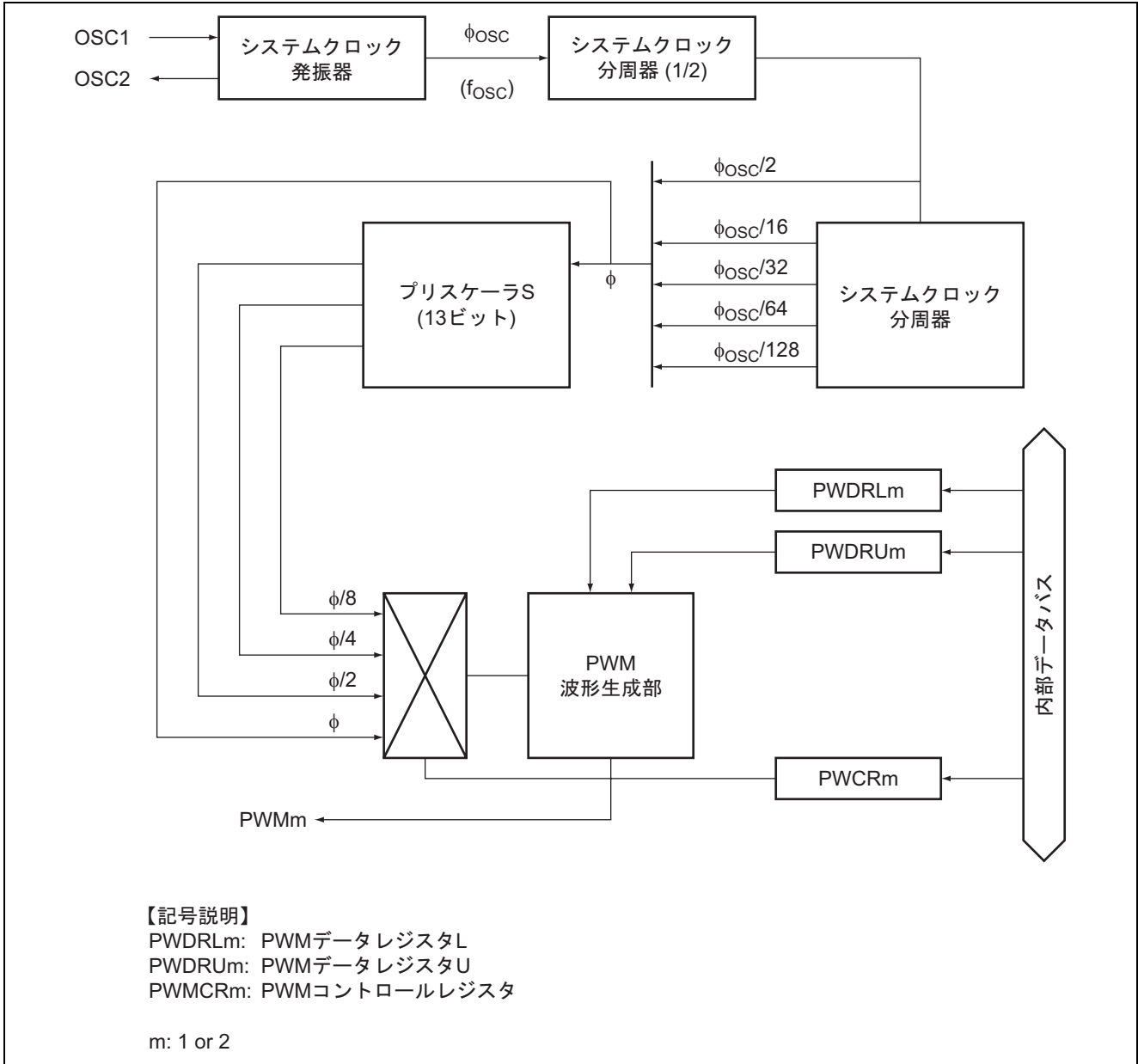


図2 10ビットPWMタイマのブロック図

2.1 レジスタ構成

表 1 レジスタ構成

レジスタ名	略称	R/W	初期値	アドレス
PWM1 コントロールレジスタ	PWCR1	W	H'FC	H'FFD0
PWM1 データレジスタ U	PWDRU1	W	H'FC	H'FFD1
PWM1 データレジスタ L	PWDRL1	W	H'00	H'FFD2
PWM2 コントロールレジスタ	PWCR2	W	H'FC	H'FFCD
PWM2 データレジスタ U	PWDRU2	W	H'FC	H'FFCE
PWM2 データレジスタ L	PWDRL2	W	H'00	H'FFCF
クロック停止レジスタ 2	CKSTPR2	R/W	H'FF	H'FFFB

2.1.1 ポートモードレジスタ (PMR9)

ビット	7	6	5	4	3	2	1	0
	—	—	—	—	PIOFF	—	PWM ₂	PWM ₁
初期値	1	1	1	1	0	—	0	0
R/W	—	—	—	—	R/W	W	R/W	R/W

PMR9 は 8 ビットのリード / ライト可能なレジスタで、P90 と P91 の端子機能を選択します。

- ビット 3: P9₂ ~ P9₀ 昇圧回路制御 (PIOFF)
P9₂ ~ P9₀ の昇圧回路のオン / オフを制御します。

PIOFF	説明
0	大電流ポートの昇圧回路をオンにします
1	大電流ポートの昇圧回路をオフにします

- ビット 2: 予約ビット
予約ビットです。書き込む値は常に 0 にしてください。
- ビット 1, 0: P9_n/PWM 端子機能選択
P9_n/PWM_{n+1} 端子を P9_n として使用するか PWM_{n+1} として使用するかを選択します。

WKP _{n+1}	説明
0	P9 _n 出力端子とします
1	PWM _{n+1} 出力端子とします

2.1.2 PWM コントロールレジスタ (PWCRm)

ビット	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PWCRm1	PWCRm0
初期値	1	1	1	1	1	1	0	0
R/W	—	—	—	—	—	—	W	W

PWCRm は 8 ビットのライト専用レジスタで、入力クロックを選択します。リセットと同時に、PWCRm は HFC に初期化されます。

- ビット 7~2: 予約ビット
- ビット 1, 0: クロックセレクト 1 (PWCRm1, PWCRm0)
10 ビット PWM タイマへの入力クロックを選択します。ライト専用ビットで、読み出すと常に 1 が読み出されます。

ビット 1 PWCRm1	ビット 0 PWCRm0	説明
0	0	入力クロックを ϕ ($t\phi^* = 1/\phi$) にします 変換周期は $512/\phi$, 最小変調幅は $1/2\phi$ となります
0	1	入力クロックを $\phi/2$ ($t\phi^* = 2/\phi$) にします 変換周期は $1024/\phi$, 最小変調幅は $1/\phi$ となります
1	0	入力クロックを $\phi/4$ ($t\phi^* = 4/\phi$) にします 変換周期は $2048/\phi$, 最小変調幅は $2/\phi$ となります
1	1	入力クロックを $\phi/8$ ($t\phi^* = 8/\phi$) にします 変換周期は $4096/\phi$, 最小変調幅は $4/\phi$ となります

【注】 * PWM タイマ入力クロック周期

2.1.3 PWM データレジスタ U, L (PWDRUm, PWDRLm)

PWDRUm

ビット	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PWDRUm1	PWDRUm0
初期値	1	1	1	1	1	1	0	0
R/W	—	—	—	—	—	—	W	W

PWDRLm

ビット	7	6	5	4	3	2	1	0
	PWDRLm7	PWDRLm6	PWDRLm5	PWDRLm4	PWDRLm3	PWDRLm2	PWDRUm1	PWDRUm0
初期値	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W

PWDRUm と PWDRLm は 10 ビットのライト専用レジスタを構成しています。上位 2 ビットは PWDRUm に、下位 8 ビットは PWDRLm に割り当てられています。PWDRUm および PWDRLm に書き込んだ値が、PWM 波形の 1 周期の総ハイレベル幅となります。

PWDRUm と PWDRLm に 10 ビットデータを書き込むと、レジスタの内容が PWM 波形ジェネレータにラッチされ、PWM 波形生成データが更新されます。10 ビットデータは、必ず次の手順で書き込んでください。

1. PWDRLm に下位 8 ビットを書き込む
2. 同一チャンネルの PWDRUm に上位 2 ビットを書き込む

PWDRUm および PWDRLm はライト専用レジスタです。読み出すと常に 1 が読み出されます。

リセットと同時に、PWDRUm は H'FC に初期化され、PWDRLm は H'00 に初期化されます。

2.1.4 クロック停止レジスタ 2 (CKSTPR2)

ビット	7	6	5	4	3	2	1	0
	—	—	—	PW2CKSTP	AECKSTP	WDCKSTP	PW1CKSTP	LDCKSTP
初期値	1	1	1	1	1	1	1	1
R/W	—	—	—	R/W	R/W	R/W	R/W	R/W

CKSTPR2 は 8 ビットのリード/ライト可能なレジスタで、周辺モジュールのモジュールスタンバイモードを制御します。ここでは、PWM タイマに関連するビットのみを説明します。

- ビット 4, 1: PWM モジュールスタンバイモードコントロール

PWmCKSTP	説明
0	PWm をモジュールスタンバイモードにします
1	PWm のモジュールスタンバイモードを解除します

2.2 基本動作

10 ビット PWM タイマを用いるには、以下の 3 ステップでレジスタを設定してください。

1. 端子の設定: PMR9 の PWM1 または PWM2 ビットを 1 にセットして、P90/PWM1 または P91/PWM2 端子を PWM 出力端子に設定します。
2. 変換周期の選択: PWM コントロールレジスタ (PWCRm) の PWCRm1 および PWCRm0 ビットを設定し、変換周期を選択します。
3. パルス幅の設定: PWDRUm および PWDRLm に出力波形データを設定します。最初に PWDRLm、次に同一チャンネルの PWDRUm の順に書き込んでください。

図 3 に示すように、1 変換周期は 4 パルスからなります。この周期 (T_H) のハイレベル幅の合計は、PWDRUm および PWDRLm に設定したデータに対応する値です。波形は、次の変換周期から更新されます。

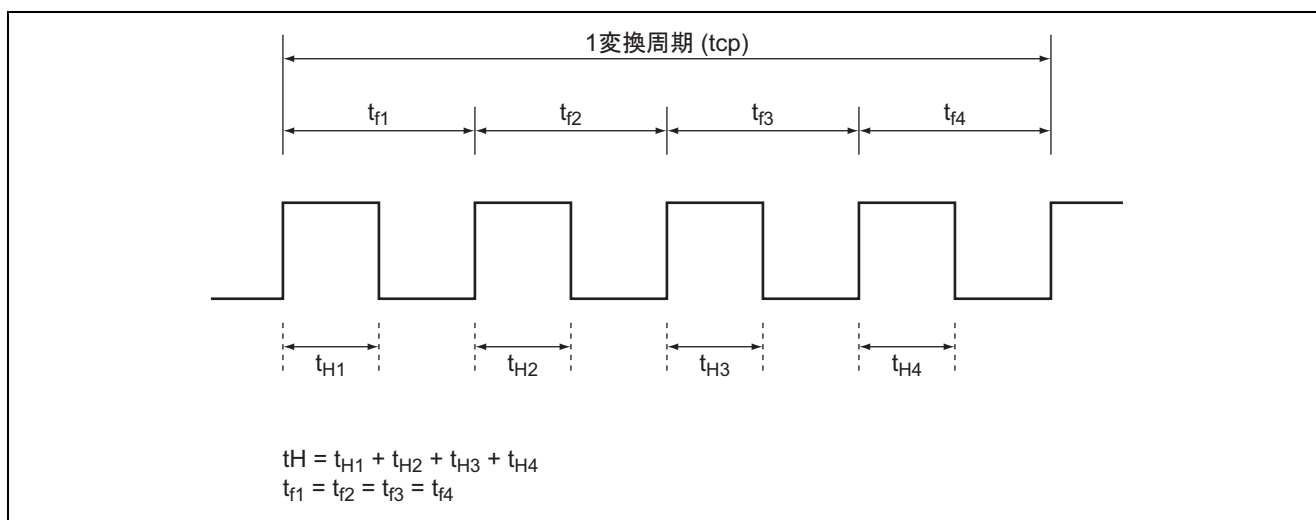


図 3 PWM 出力波形

$$T_H = (\text{PWDRUm および PWDRLm の設定値} + 4) \times t_{\phi}/2$$

ここで、 t_{ϕ} は PWM 入力クロック周期 (1/φ, 2/φ, 4/φ, 8/φ)

3. サイン波生成の理論

基本的な動作原理は、DC 電圧レベルの生成に基づきます（詳細は、アプリケーションノート「PWM タイマを使用した DAC」（RJS06B0001-0200）を参照）。生成した DC 電圧レベルが正弦波を描いていけばよいわけです。

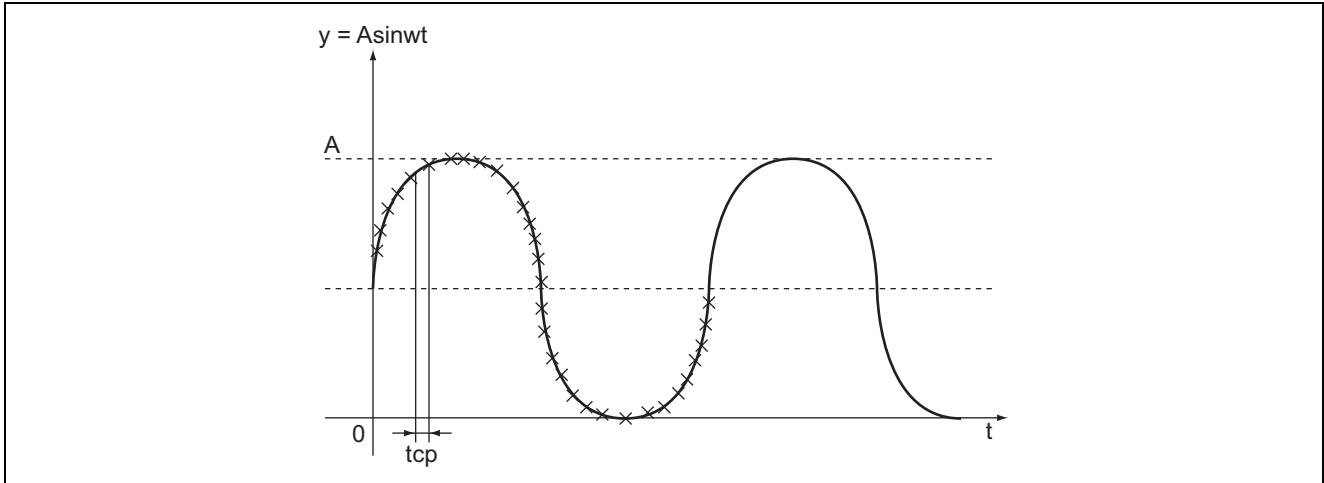


図 4 サイン波の代表例

変換周期（ステップ 2）を固定して、デューティサイクル（ステップ 3）を変化させると、DC 電圧レベルは次の変換周期から更新されます。V_{CC}（振幅 A）と V_{SS}（ゼログラウンド）の間で、128 ポイントのサイン波を生成するには、以下のようにします。

サイン波の周期 $T = 256 \times tcp$ ，ここで tcp は変換周期

V_{SS} から V_{CC} 間は、 $256/2 = 128$ 変換周期が必要です。変化値が一定で V_{CC}/128 に等しければ、三角波が生成されます。

サイン波の生成には、次の 2 点を考慮する必要があります。

- 波形の周波数: メインクロック，システムクロック分周器，PWM 分周器，サイン波のサンプリングポイント数によって決まります。
- サイン波の形: サンプリングポイントの数のみで決まります。

以下の式にしたがい、256 のサイン波データポイントを生成し、配列 LUT[i] に格納します。

$$LUT[i] = (\text{int}) (\sin (2 \times i \times \pi/256) \times 512 + 511)$$

表 2 256 データポイントのサイン波の計算

n	Y = Sin (n × 2 × π/256)	LUT[n] = Y × 512 + 511
0	0	511
2	0.0245	524
3	0.0491	562
4	0.0736	587
•	•	•
•	•	•
252	-0.0980	461
253	-0.0736	473
254	-0.0491	488
255	-0.0245	498

4. プログラムの概要

ここで示すプログラムでは、メインクロック (ϕ_{osc}) は 10MHz に設定し、2分周します (アクティブモード)。

```
SYSCR1 = H'07
```

この初期化中に、P91 を PWM2 出力端子に設定し、PWM2 の入力クロックを ϕ に設定します。 $\phi = 5\text{MHz}$ では、変換周期は $512/\phi = 102.4\mu\text{s}$ です。

```
PMR9 = H'F2
PWCR2 = H'FC
```

CCR の割り込みビットをマスクして非同期イベントカウンタ (AEC) の初期化を可能にします。AEC はオーバフローしたときに割り込みを発生するイベントカウンタで、変換周期ごとにアナログ出力をチャージします。AEC は 2 つの独立した 8 ビットカウンタで構成されているため、オーバフロー時には OVL フラグのみがセットされます。

```
ECCSR = H'10    [ECH と ECL を 2 つの独立した 8 ビットイベントカウンタチャンネルに設定する]
ECCR = H'90     [ECH 用にクロック  $\phi/4$  を、ECL 用に  $\phi/2$  を選択する]
AEGSR = H'40    [AEVH および AEVL の立ち上がりエッジを検出する]
ECCSR = H'1F    [ECH および ECL のイベントクロック入力を許可する。
                0: ECL および ECH をセット
                1: ストップリセットおよびカウントアップ]
IRR2 = H'00     [IRREC フラグをクリアする]
IENR2 = H'01    [AEC 割り込み要求を許可する]
```

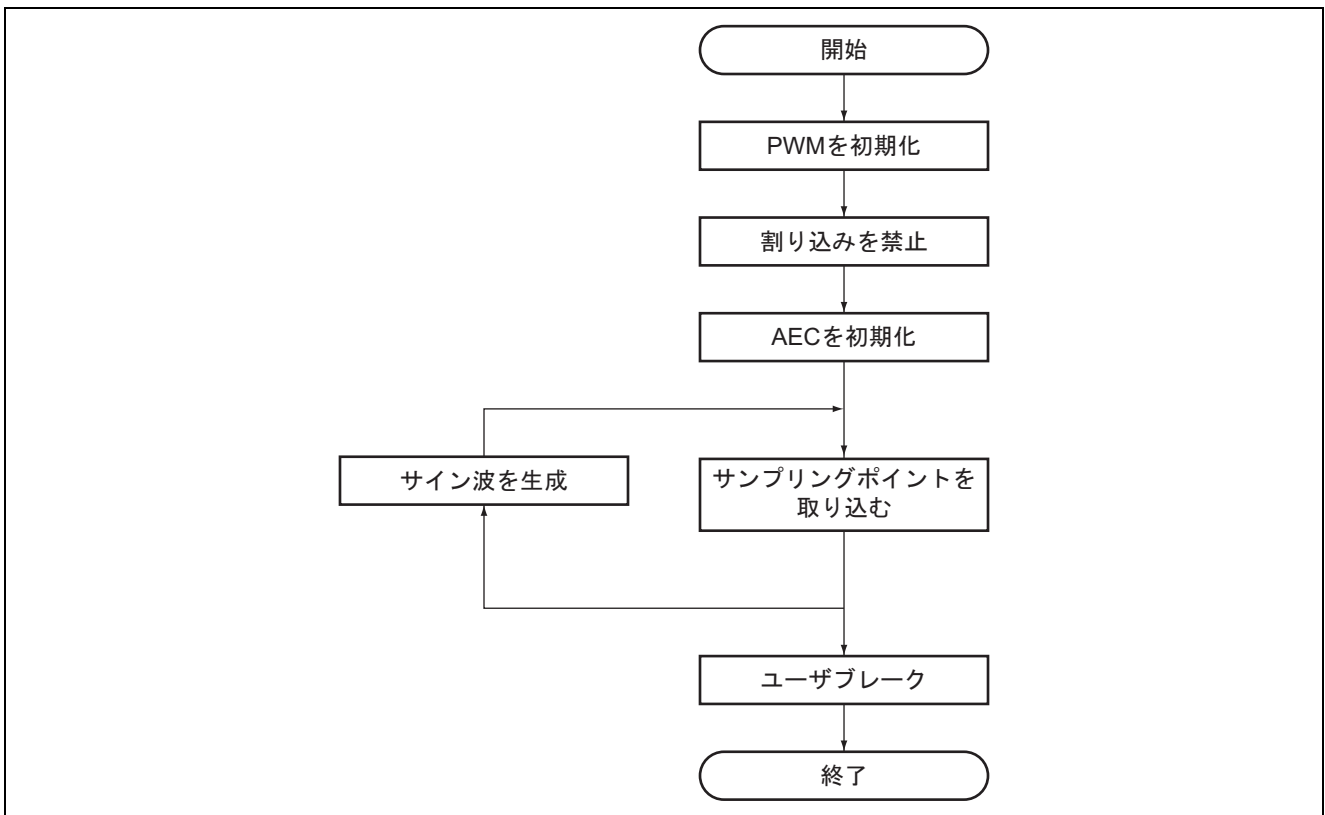
プログラムを実行すると、それぞれの AEC 割り込みが発生するたびに OVL フラグが更新され、関数 storeCount (LUT[i]) が PWM に次のアナログ信号をロードします。カウンタ i が LUT テーブルの終端を示すと、サイン波が生成されます。

PWM で、ECL のオーバフロー周期は次のようになります。

$$2/\phi \times 2^8 = 5\text{M}/2 \times 256 = 102.4\mu\text{s} \text{ (PWM2 の変換周期と同じ)}$$

5. プログラムのフローチャート

1. サイン波生成のフローチャート



6. プログラムリスト

```

/*****/
/* FILE:          PWM_Sinewave.c                               */
/* DATE:          Thu, Mar 20, 2003                           */
/* DESCRIPTION:    Using PWM to generate sine wave            */
/* CPU TYPE:      H8/38024F                                    */
/*                                                        */
/* This file is generated by Hitachi Project Generator (Ver.2.1). */
/*****/

#include <machine.h>
#include "iodefine.h"
#include <math.h>
/*****/
/* Function define                                           */
/*****/

void init_PWM(unsigned char);
void storeCount(unsigned short);
void aecint( void );
void init_AEC(unsigned char,unsigned char,unsigned char,unsigned char,unsigned char);

/*****/
/* RAM define                                               */
/*****/
unsigned char PWDR_L2, PWDR_U2, r=0;
unsigned int i=0, frequency = 21;
unsigned int final=0, lowcnt=0;
float pi=3.141596;

unsigned int LUT[256];

/*****/
/* Main Program                                           */
/*****/
void main ( void )
{
    init_PWM(0);                // select PWM2 to have (512/5Mhz) conversion period
    set_imask_ccr(1);           // Interrupt Disable
    init_AEC(2,1,1,1,1);

    for (i=0; i<256; i++)
    {
        LUT[i] = (int)(sin(2*i*pi/256)*512+511);
    }
    while (1)
    {
        ;
    }
}void init_PWM(unsigned char selClk2)
{
    if (selClk2 <= 3)           // Check if valid, otherwise PWM2 is off
    {
        P_IO.PMR9.BIT.PWM2 = 1; // Configure P91 as PWM2 output pin
        P_PWM2.PWCR2.BYTE = selClk2; // Clock select for PWM2,write only
    }
}

```

```

/*****
/* Write each digital code into PWDR registers
*/
/*****
void storeCount(unsigned short PWDRval_2)
{
    P_PWM2.PWDR2.BYTE = (unsigned char)(PWDRval_2 & 0x00FF);
                                // Write lower 8bits of 10bits data
    P_PWM2.PWDRU2.BYTE = (unsigned char) ((PWDRval_2 & 0x0300) >> 8);
                                // Write upper 8bits of 10bits data
}

/*****
/* AEC Interrupt Service Routine
*/
/*****
void aecint (void)
{
    P_SYSCR.IRR2.BIT.IRREC = 0;
                                // Clear IRREC flag

    if(P_AEC.ECCSR.BIT.OVL == 1)
                                // Check for ECL overflow flag
    {
        P_AEC.ECCSR.BIT.OVL = 0;
                                // Clears flag

        final = LUT[lowcnt];
        storeCount(final);
                                // Write Sinewave digital code into PWM registers
        lowcnt += frequency;
        if(lowcnt>255) lowcnt = lowcnt-256;
                                // If reached end of lperiod, then reset
    }
}

```

```

// Only OVL flag will set when overflow occurs
void init_AEC(unsigned char csH,unsigned char csL,unsigned char esH,unsigned char esL,unsigned char
intr)
{
    P_AEC.ECCSR.BIT.CH2 = 1;                // ECH & ECL as independent 8-bit counter
    P_AEC.ECCR.BIT.ACKH = csH;              // 0x00:AEVH as i/p
                                           // 0x01:divide by 2
                                           // 0x02:divide by 4
                                           // 0x03:divide by 8

    P_AEC.AEGSR.BIT.AHEGS = esH;           // 0x00:Falling edge sense on AEVH pin
                                           // 0x01:Rising edge sense on AEVH pin
                                           // 0x02:Both edges sense on AEVH pin
                                           // 0x03:AEVH pin usage prohibited

    P_AEC.ECCR.BIT.ACKL = csL;              // 0x00:AEVL as i/p
                                           // 0x01:divide by 2
                                           // 0x02:divide by 4
                                           // 0x03:divide by 8

    P_AEC.AEGSR.BIT.ALEGS = esL;           // 0x00:Falling edge sense on AEVL pin
                                           // 0x01:Rising edge sense on AEVL pin
                                           // 0x02:Both edges sense on AEVL pin
                                           // 0x03:AEVL pin usage prohibited

    P_AEC.ECCSR.BIT.CUEL = 0;               // ECH event clk i/p is Disable, ECL value held
    P_AEC.ECCSR.BIT.CUEH = 0;               // ECL event clk i/p is Disable, ECH value held
    P_AEC.ECCSR.BIT.CRCL = 0;               // 0-ECL is reset, 1-stop reset & countup
    P_AEC.ECCSR.BIT.CRCH = 0;               // 0-ECH is reset, 1-stop reset & countup

    P_AEC.ECCSR.BIT.OVL = 0;                // Clears flag
    P_AEC.ECCSR.BIT.OVH = 0;                // Clears flag

    P_AEC.ECCSR.BIT.CUEL = 1;               // ECH event clk i/p is Enabled
    P_AEC.ECCSR.BIT.CUEH = 1;               // ECL event clk i/p is Enabled
    P_AEC.ECCSR.BIT.CRCL = 1;               // 0-ECL is reset, 1-stop reset & countup
    P_AEC.ECCSR.BIT.CRCH = 1;               // 0-ECH is reset, 1-stop reset & countup

    P_SYSCR.IRR2.BIT.IRREC = 0;              // Clear IRREC flag
    P_SYSCR.IENR2.BIT.IENEC = intr;         // AEC Interrupt Request, 1-Enable, 0-Disable

    set_imask_ccr(0);                       // Interrupts, 0-Enable, 1-Disable
                                           // set_imask_ccr() comes as a pair
}

```

参考文献

- H8/38024, H8/38024S, H8/38024R, H8/38124 Group Hardware Manual
- H8/300L Super Low Power Series – Low-cost CPU Board CPUBD-38024F User's Manual
- アプリケーションノート , PWM タイマを使用した DAC (RJS06B0001-0200)

ホームページとサポート窓口

ルネサステクノロジホームページ

<http://japan.renesas.com/>

お問合せ先

<http://japan.renesas.com/inquiry>

csc@renesas.com

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2003.07.28	—	初版発行
2.00	2006.09.15	1 ~ 8, 14	内容変更

安全設計に関するお願い

1. 弊社は品質、信頼性の向上に努めておりますが、半導体製品は故障が発生したり、誤動作する場合があります。弊社の半導体製品の故障又は誤動作によって結果として、人身事故、火災事故、社会的損害などを生じさせないような安全性を考慮した冗長設計、延焼対策設計、誤動作防止設計などの安全設計に十分ご留意ください。

本資料ご利用に際しての留意事項

1. 本資料は、お客様が用途に応じた適切なルネサス テクノロジ製品をご購入いただくための参考資料であり、本資料中に記載の技術情報についてルネサス テクノロジが所有する知的財産権その他の権利の実施、使用を許諾するものではありません。
2. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他応用回路例の使用に起因する損害、第三者所有の権利に対する侵害に関し、ルネサス テクノロジは責任を負いません。
3. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他全ての情報は本資料発行時点のものであり、ルネサス テクノロジは、予告なしに、本資料に記載した製品または仕様を変更することがあります。ルネサス テクノロジ半導体製品のご購入に当たりましては、事前にルネサス テクノロジ、ルネサス販売または特約店へ最新の情報をご確認頂きますとともに、ルネサス テクノロジホームページ(<http://www.renesas.com>)などを通じて公開される情報に常にご注意ください。
4. 本資料に記載した情報は、正確を期すため、慎重に制作したのですが万一本資料の記述誤りに起因する損害がお客様に生じた場合には、ルネサス テクノロジはその責任を負いません。
5. 本資料に記載の製品データ、図、表に示す技術的な内容、プログラム及びアルゴリズムを流用する場合は、技術内容、プログラム、アルゴリズム単位で評価するだけでなく、システム全体で十分に評価し、お客様の責任において適用可否を判断してください。ルネサス テクノロジは、適用可否に対する責任を負いません。
6. 本資料に記載された製品は、人命にかかわるような状況の下で使用される機器あるいはシステムに用いられることを目的として設計、製造されたものではありません。本資料に記載の製品を運輸、移動体用、医療用、航空宇宙用、原子力制御用、海底中継用機器あるいはシステムなど、特殊用途へのご利用をご検討の際には、ルネサス テクノロジ、ルネサス販売または特約店へご照会ください。
7. 本資料の転載、複製については、文書によるルネサス テクノロジの事前の承諾が必要です。
8. 本資料に関し詳細についてのお問い合わせ、その他お気づきの点がございましたらルネサス テクノロジ、ルネサス販売または特約店までご照会ください。