

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、誤りがな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。

## H8/300H Tiny シリーズ

### PWM による DTMF 発生器接続例

#### 要旨

PWM (Pules Width Modulation) を利用した D/A コンバータによる DTMF (Dual Tone Multi Frequency) 発生器で発生させたトーンを用いて、汎用電話器から電話を掛けます。

#### 動作確認デバイス

H8/36014CPU

#### 目次

1. 仕様 .....	2
2. 使用機能説明 .....	5
3. 動作原理 .....	6
4. ソフトウェア説明 .....	7
5. フローチャート .....	10
6. プログラムリスト .....	15

### 1. 仕様

- 図1にPWM (Pules Width Modulation) を利用したD/A コンバータによるDTMF (Dual Tone Multi Frequency) 発生器のハードウェア構成を示します。
- 本タスク例では、キーボードを押下することで各キーに対応した2種類の正弦波の合成波を接続されたスピーカに出力します。
- 本タスク例で使用しているキーボードはパソコン入力用のテンキーボードです。通常1ビットシリアル信号で出力されますが、今回の例ではパラレル入出力対応に改造してあります。

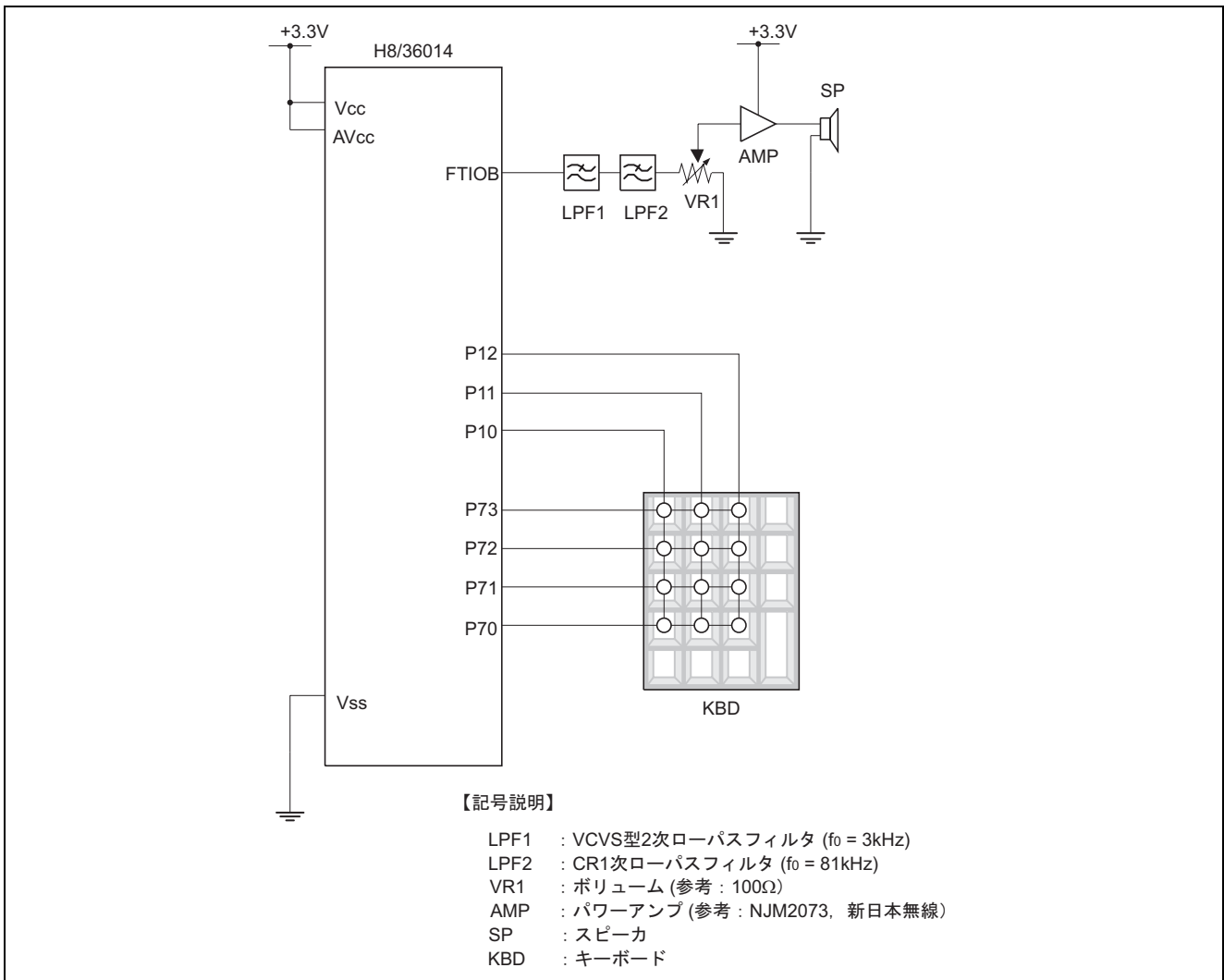


図1 ハードウェア構成

- PWM による D/A 変換
  - a. ローパスフィルタに PWM 波を通過させることで DC 電圧が得られます。
  - b. DC 電圧レベルは、PWM 波のデューティ比 (PWM 波の周期に対するハイレベルの期間の割合) で決まります。  

$$V_{out} = V \times H/T$$
 (V: 振幅, H: ハイレベルの期間, T: 周期)
  - c. 今回の実施例では、PWM 波の周波数を 31.25kHz に設定しているため、ローパスフィルタのカットオフ周波数を PWM クロックの周波数よりも十分に低い周波数の 3kHz に設定し、VCVS 型のアクティブ・ローパスフィルタを使用しました。
  - d. また、VCVS 型のアクティブ・ローパスフィルタは OP アンプを使用しているため高周波領域で動作が追従しなくなり、数 MHz 以上の高い周波数を通過させてしまうので、CR フィルタを併用して高周波領域の減衰を確保しています。
- 本タスク例の動作は以下のとおりです。
  - a. キーボードに接続した 7 本のポートのうち P70 ~ P73 を順次切替えて "H" レベルにします。
  - b. 次に残り 3 本のポート P10 ~ P12 の状態を監視し、"H" レベルを検出したポートにより押下されたキーを特定します。
  - c. 押下されたキーに対応する DTMF 音を出力するために、PWM を利用して D/A 変換します。
  - d. D/A 変換は、最初に、アンプの入力が所望の波高値になるような変調がかかった PWM 波形をハードウェアタイマ W の出力ポート (FTIOB) から出力します。PWM クロックの周波数よりも十分に低い周波数をローパスフィルタのカットオフ周波数に選んであるので、高い周波数の方形波は平滑されます。
  - e. 次に DTMF 音の周波数に応じてデューティ比を変化させます。この場合、低周波部分がローパスフィルタを通過して DTMF 音がスピーカに出力されます。
  - f. 図 2 に DTMF 周波数を示します。例えば、テンキーの "1" を押下した場合、697Hz と 1,209Hz の合成波がスピーカから出力されます。音量はボリュームで調整します。

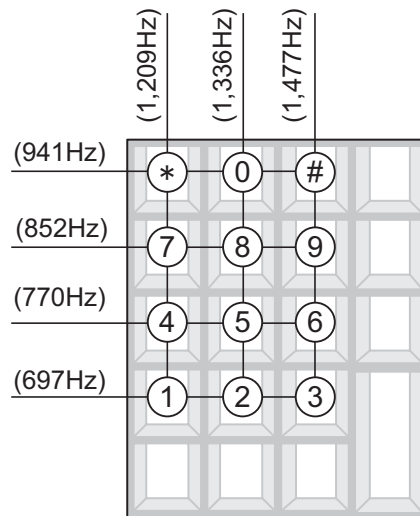


図 2 DTMF 周波数

- 動作確認
  - a. 本タスク例の動作確認は汎用の電話機を用いて実施しました。
  - b. 確認方法は、DTMF 発生器と受話器を 3cm 程度離し、発生させたトーン信号を用いてダイヤルリングし、電話の発信可否で判断しました。
  - c. 確認の結果、0~9 の数字を全て認識し正常に電話がつながりました。
- キー種類拡張方法
  - a. 今回のタスク例では 7 本の信号線を使用し、4 行 3 列の合計 12 種類のキーで、汎用の電話機のキー配列「1, 2, 3, 4, 5, 6, 7, 8, 9, 0, \*, #」を実現しました。
  - b. 図 3 に拡張キー配列を示します。キー「A, B, C, D」を拡張する場合は、1,633Hz 用の信号線を 1 本追加しキーパッドの列に追加します。

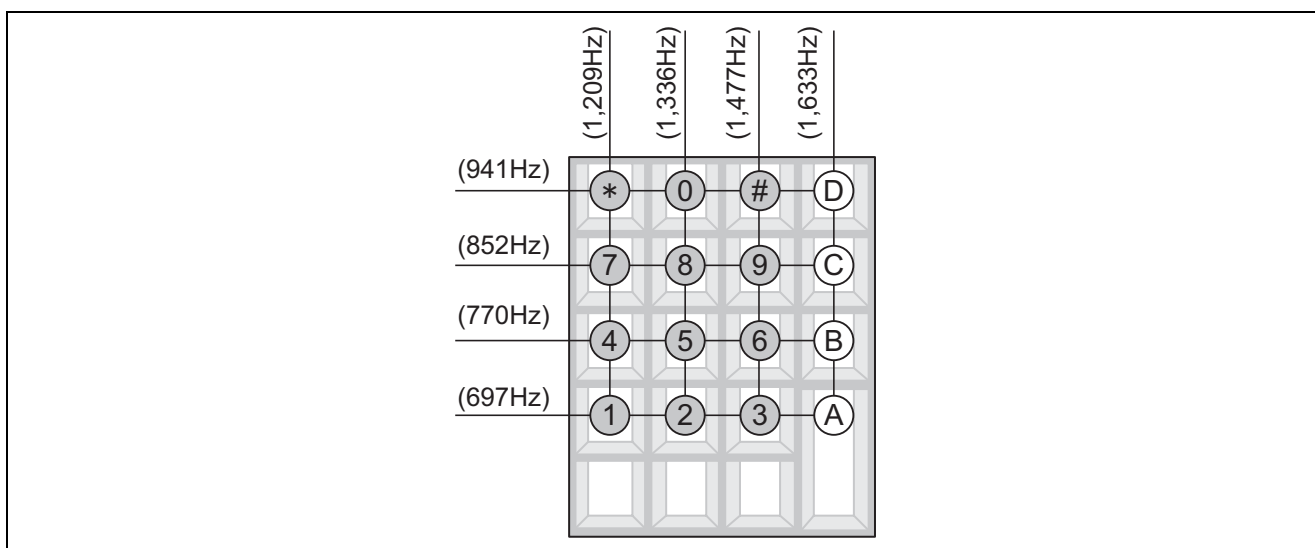


図 3 拡張キー配列

2. 使用機能説明

- 図 4 に本タスク例における H8/36014 の使用機能のブロック図を，表 1 に機能割付を示します。

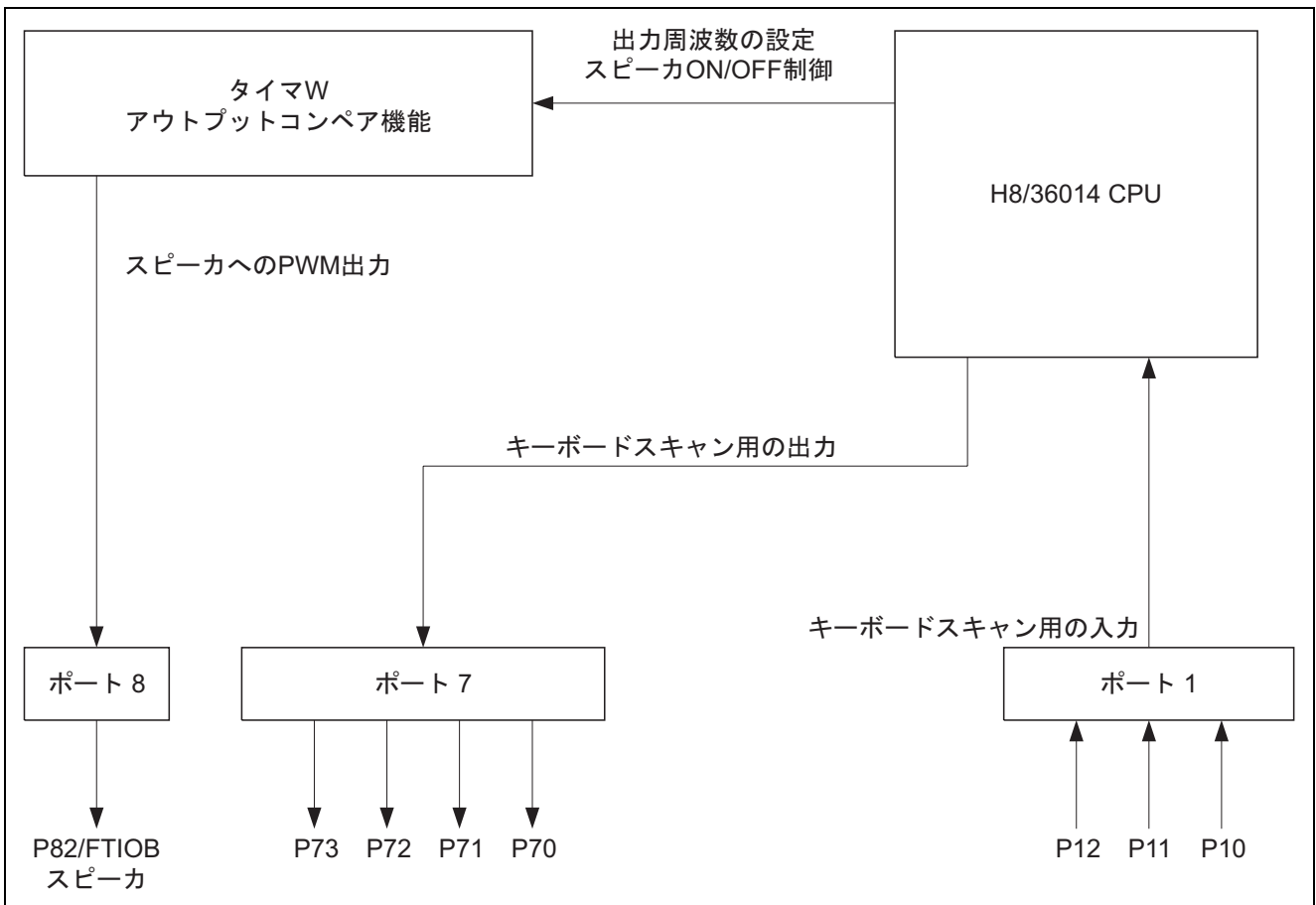


図 4 使用機能ブロック図

表1 機能割付け

使用機能	機能割付け
タイマ W	PWM 機能を使用して PWM 出力を行ないます。ジェネラルレジスタ B (GRB) へ値を設定することにより，出力周波数を変更します。
ポート 1	ポート 1 の P10～P12 入力端子より，キーボード入力を検出します。
ポート 7	ポート 7 の P70～P73 出力端子により，順次切り替えて H レベルにします。

3. 動作原理

1. 図 5 にタイマ W を使用した ,DTMF 出力を行う際の動作原理を示します。図 5 に示すようにタイマ W コンペアマッチ周期ごとに GRB の値を変化させることにより DTMF を出力しています。

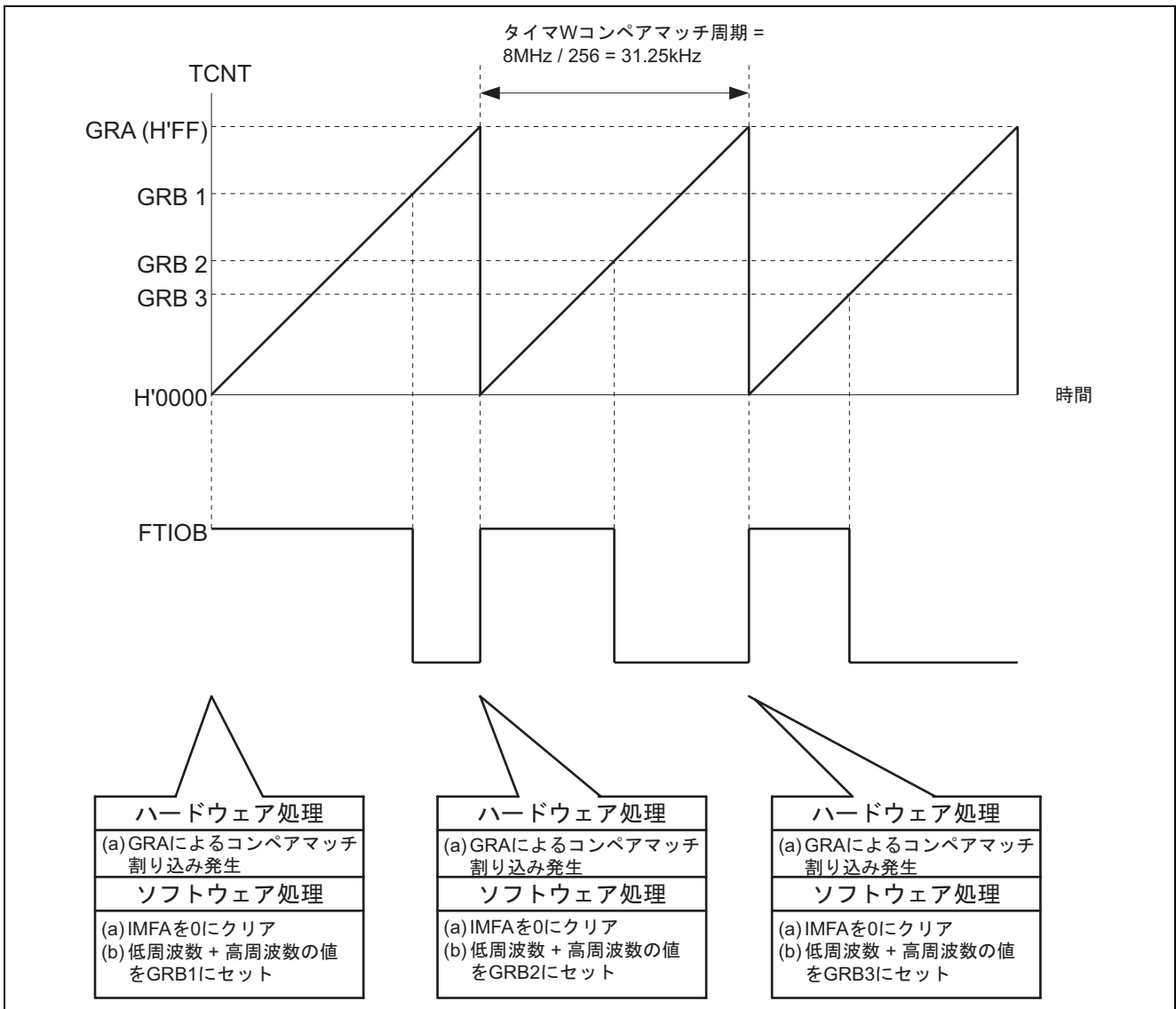


図 5 タイマ W を使用した DTMF 出力の動作原理

2. 本タスクのキーボードは、0~9, \*, #, の 12 種類のキーしかありません。A, B, C, D, が必要な場合は input\_key (void) と、ROM 定義の押しボタン周波数テーブル (PB) を変更します。なお、押しボタン周波数テーブルの変更箇所は「//」でコメントアウトしておりますので、「//」を外して使用してください。



## 4. ソフトウェア説明

### 4.1 モジュール説明

表 2 に本タスク例におけるモジュール説明を示します。

表 2 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	main	初期設定, 周波数出力
キーボード入力 処理ルーチン	input_key	キーボードをスキャンし, 対応するキーコードを戻す キーボードが押されていない場合: -1 キーボードが押された場合: 0 ~ 11
W タイマ割り込み 処理ルーチン	tmrw	サインテーブルのインデックス更新 PWM 出力更新

### 4.2 引数説明

本タスク例では, 引数を使用しておりません。

### 4.3 使用内部レジスタ説明

本タスク例の使用内部レジスタを表 3 に示します。

表 3 使用内部レジスタ説明

レジスタ名	機能説明	アドレス	設定値
TMRW	タイマモードレジスタ W : ジェネラルレジスタの機能やタイマの出力モード の選択	H'FF80	H'81
	CTS カウンタスタート : CTS = 1 のとき, TCNT がカウンタ開始を示す : CTS = 0 のとき, TCNT がカウンタ停止を示す	ビット 7	1
	PMWB : PMWB = 1 のとき, FTIOB 端子の出力モードを PWM 出力とする	ビット 0	1
TCRW	タイマコントロールレジスタ W : カウンタクロックの選択 : カウンタのクリア条件やタイマの出力レベルの設定	H'FF81	H'82
	CCLR カウンタクリア : CCLR = 1 のとき, コンペアマッチ A によって TCNT がクリア	ビット 7	1
	CKS2 クロックセレクト : CKS2 = 0, CKS1 = 0, CKS0 = 0 のとき, TCNT	ビット 6	0
	CKS1 入力クロックをシステムクロックの分周なしのク ロックに設定	ビット 5	0
	CKS0	ビット 4	0
TCB タイマ出力レベルセット B : TCB = 1 のとき, 最初のコンペアマッチ B が発生 するまでの FTIOB 端子の出力値を 1 とする	ビット 1	1	

表 3 使用内部レジスタ説明

レジスタ名	機能説明	アドレス	設定値
TIERW	タイマインタラプトイネーブルレジスタ W : タイマ W の割り込み要求を制御	H'FF82	H'00 (初期設定時)
IMIEA	インプットキャプチャ/コンペアマッチ割り込みイネーブル A : IMIEA = 1 のとき, TSRW の IMFA による割り込み要求 IMIA がイネーブルとなる	ビット 0	1
TSRW	割り込み要求ステータスを表示	H'FF83	H'00
IMFB	インプットキャプチャ/コンペアマッチフラグ B	ビット 1	0
IMFA	インプットキャプチャ/コンペアマッチフラグ A	ビット 0	0
TCNT	タイマカウンタ : システムクロックの 8 分周のクロックを入力とする 16 ビットのアップカウンタ	H'FF86	H'00
GRA	ジェネラルレジスタ A	H'FF88	H'FF
GRB	ジェネラルレジスタ B	H'FF8A	H'00
PCR1	ポートコントロールレジスタ 1 : 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 PCR1 = H'00 のとき, : P10 ~ P12 端子は汎用入力端子として機能	H'FFE4	H'00
PDR1	ポートデータレジスタ 2 : ポート 1 の汎用入出力ポートデータレジスタ	H'FFD4	H'00
PCR7	ポートコントロールレジスタ 7 : ポート 7 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 PCR7 = H'0F のとき, : P70 ~ P73 端子は汎用出力端子として機能	H'FFEA	H'0F
PDR7	ポートデータレジスタ 7 : ポート 7 の汎用入出力ポートデータレジスタ	H'FFDA	H'00 (初期設定時)

#### 4.4 使用 RAM 説明

表 4 に本タスク例における使用 RAM 説明を示します。

表 4 使用 RAM 説明

ラベル名	機能	アドレス	使用モジュールラベル名
inc1	サインテーブルの増分を格納, 低周波数用 (2 byte)	H'FB80	main,tmrw
inc2	サインテーブルの増分を格納, 高周波数用 (2 byte)	H'FB82	main,tmrw
pos1	サインテーブルの取り出し位置を格納, 低周波数用 (2 byte)	H'FB84	main,tmrw
pos2	サインテーブルの取り出し位置を格納, 高周波数用 (2 byte)	H'FB86	main,tmrw
key_data	入力キーデータを格納 (2 byte)	H'FB88	main
old_key	前回のキーデータを格納 (2 byte)	H'FB8A	main
i	キースキャン用ループカウンタを格納 (2 byte)	H'FB8C	input_key
j	キースキャン用ループカウンタを格納 (2 byte)	H'FB8E	input_key
key	キースキャン用キーデータ (2 byte)	H'FB90	input_key

#### 4.5 使用 ROM 説明

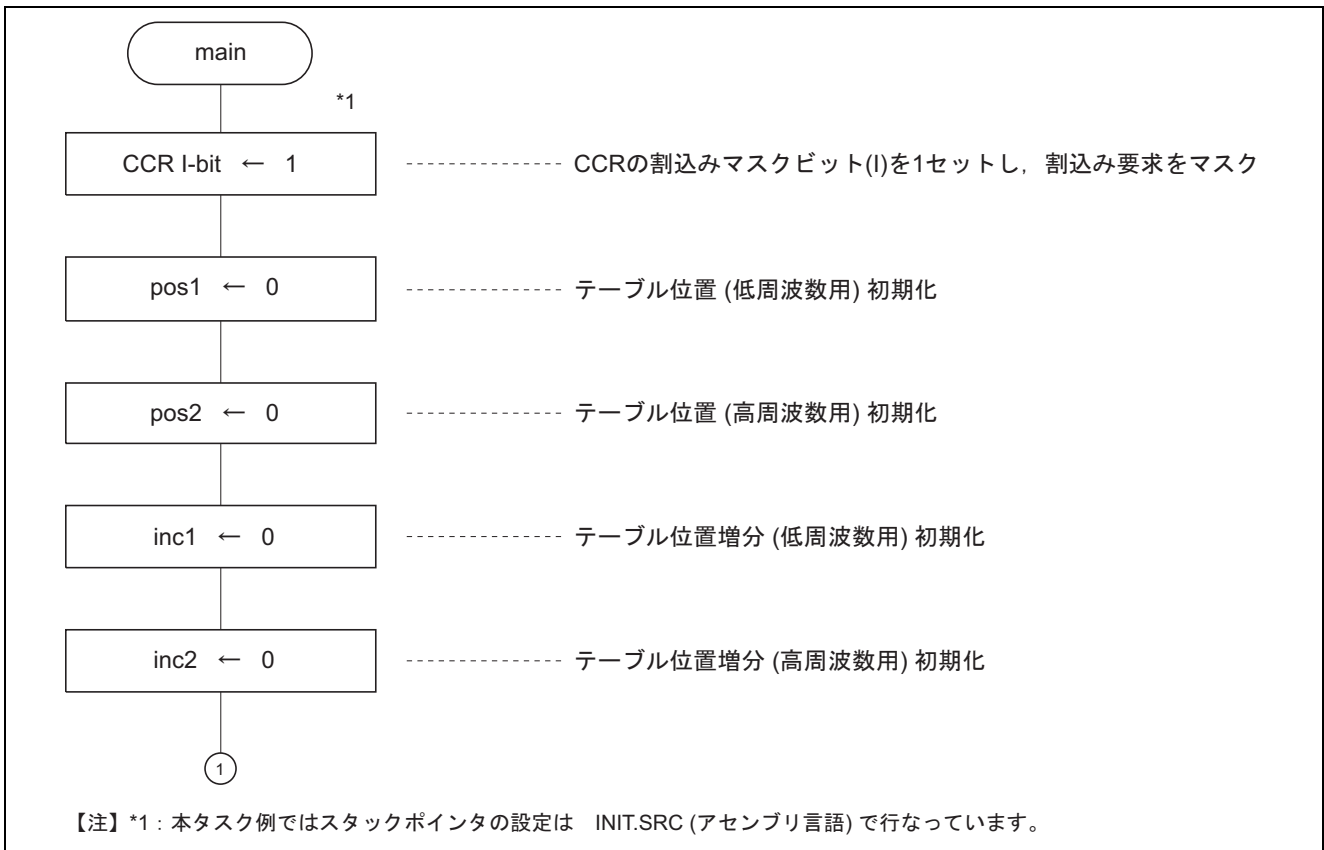
表 5 に本タスク例における使用 ROM 説明を示します。

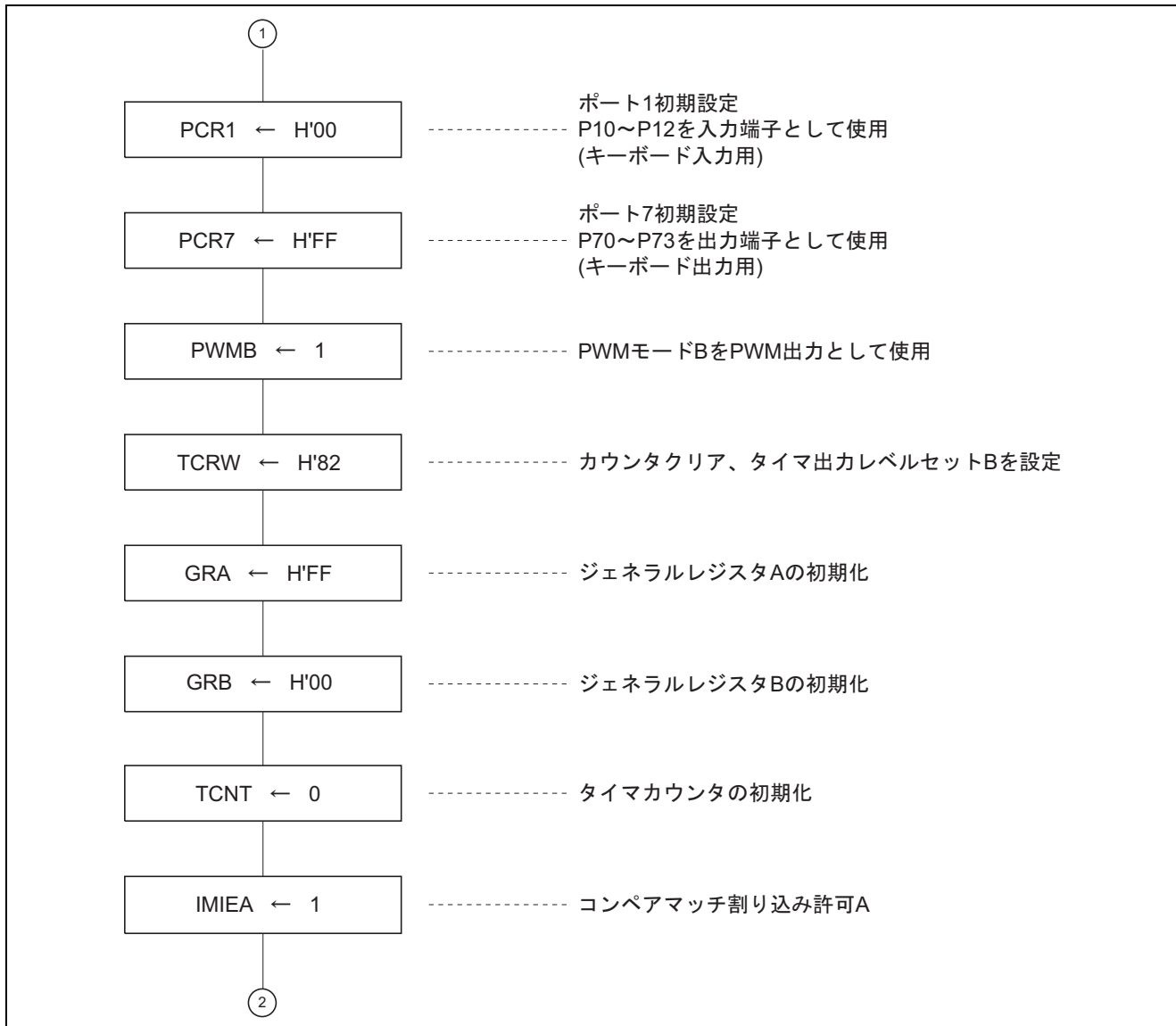
表 5 使用 ROM 説明

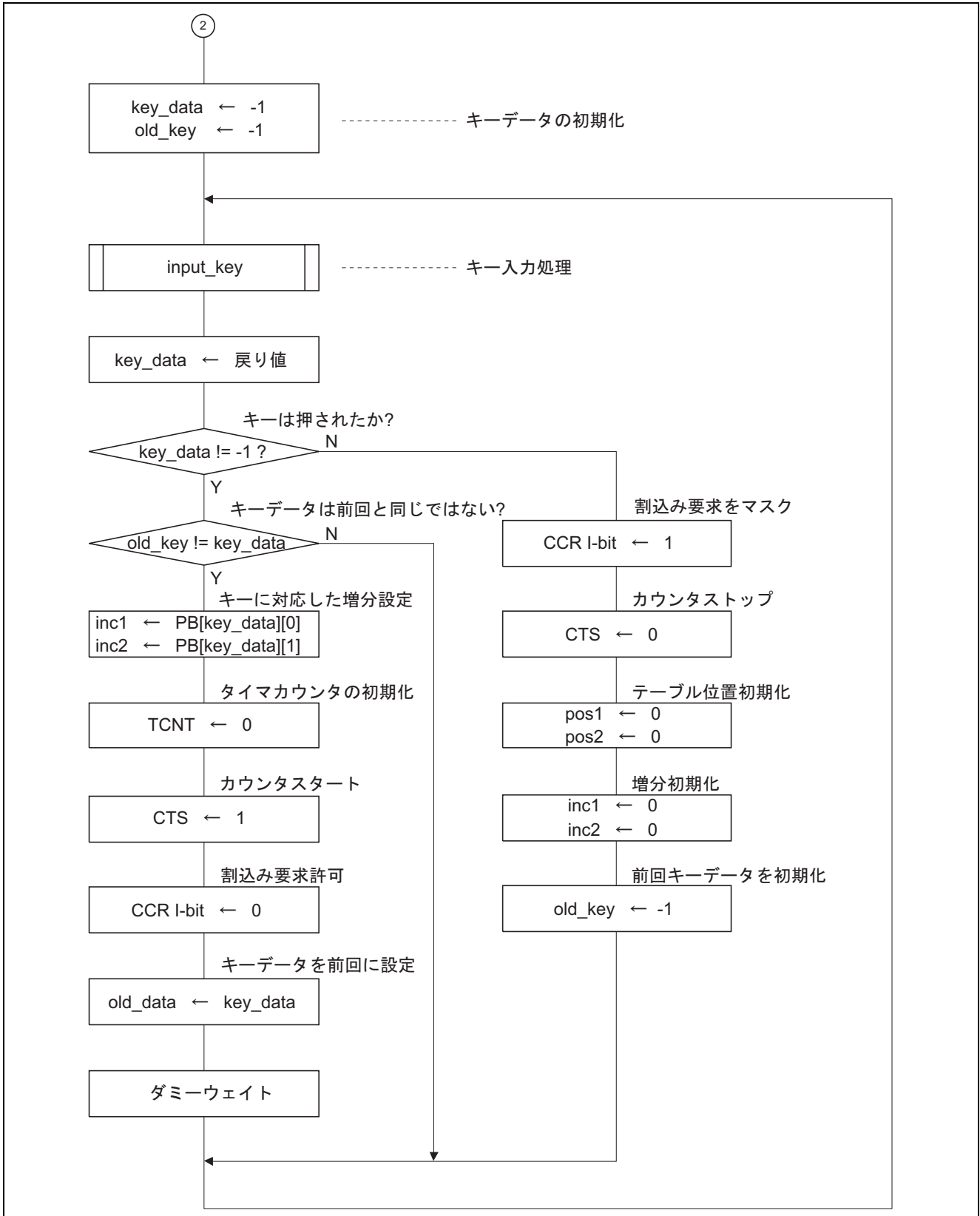
ラベル名	機能	アドレス	使用モジュールラベル名
PB	入力キーに対応したサインテーブルの増分を格納 (48 byte)	H'02D0	main,tmrw
sine	サインテーブル (256 byte)	H'0310	main,tmrw

## 5. フローチャート

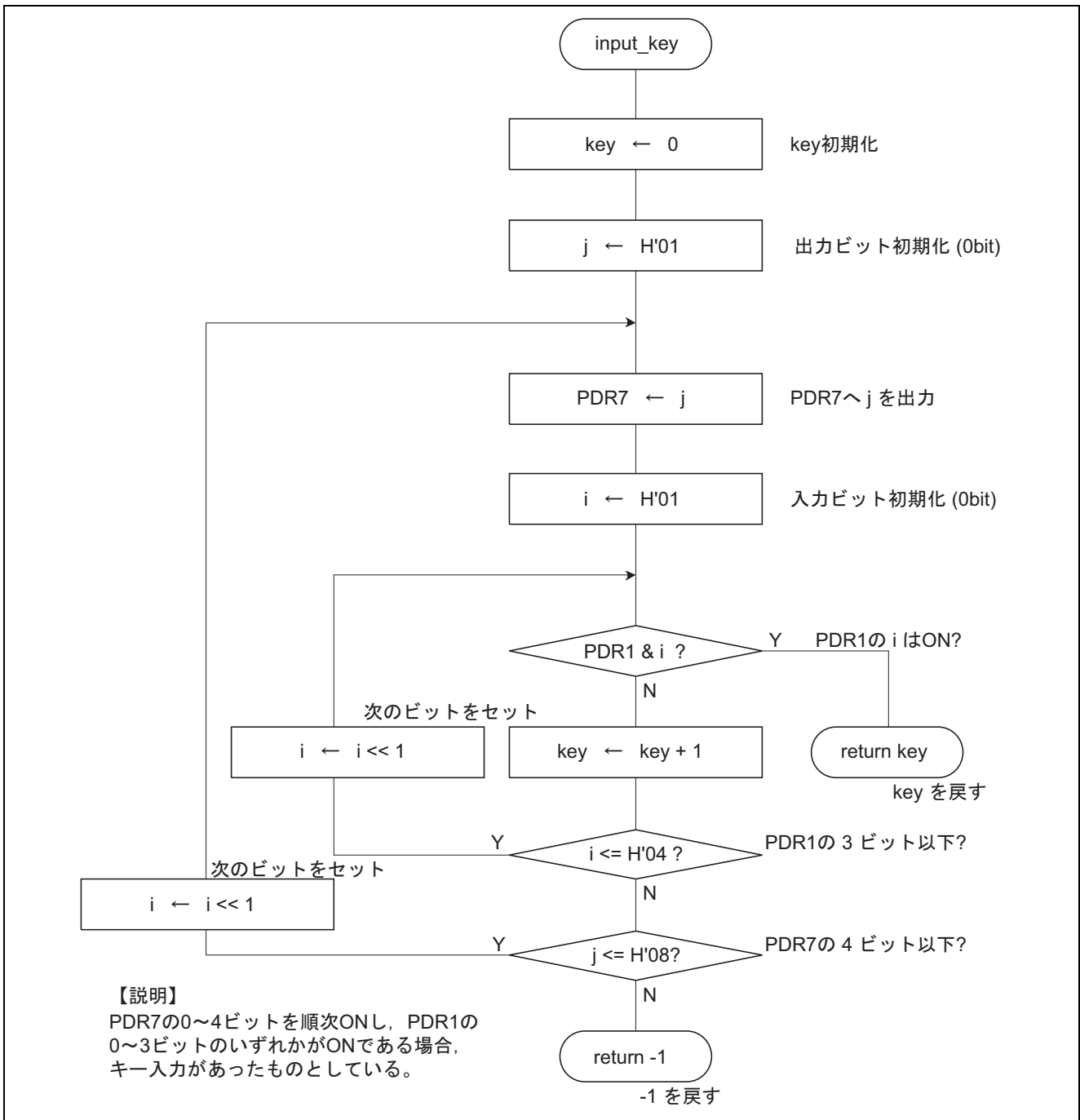
### 5.1 メインルーチン (main)



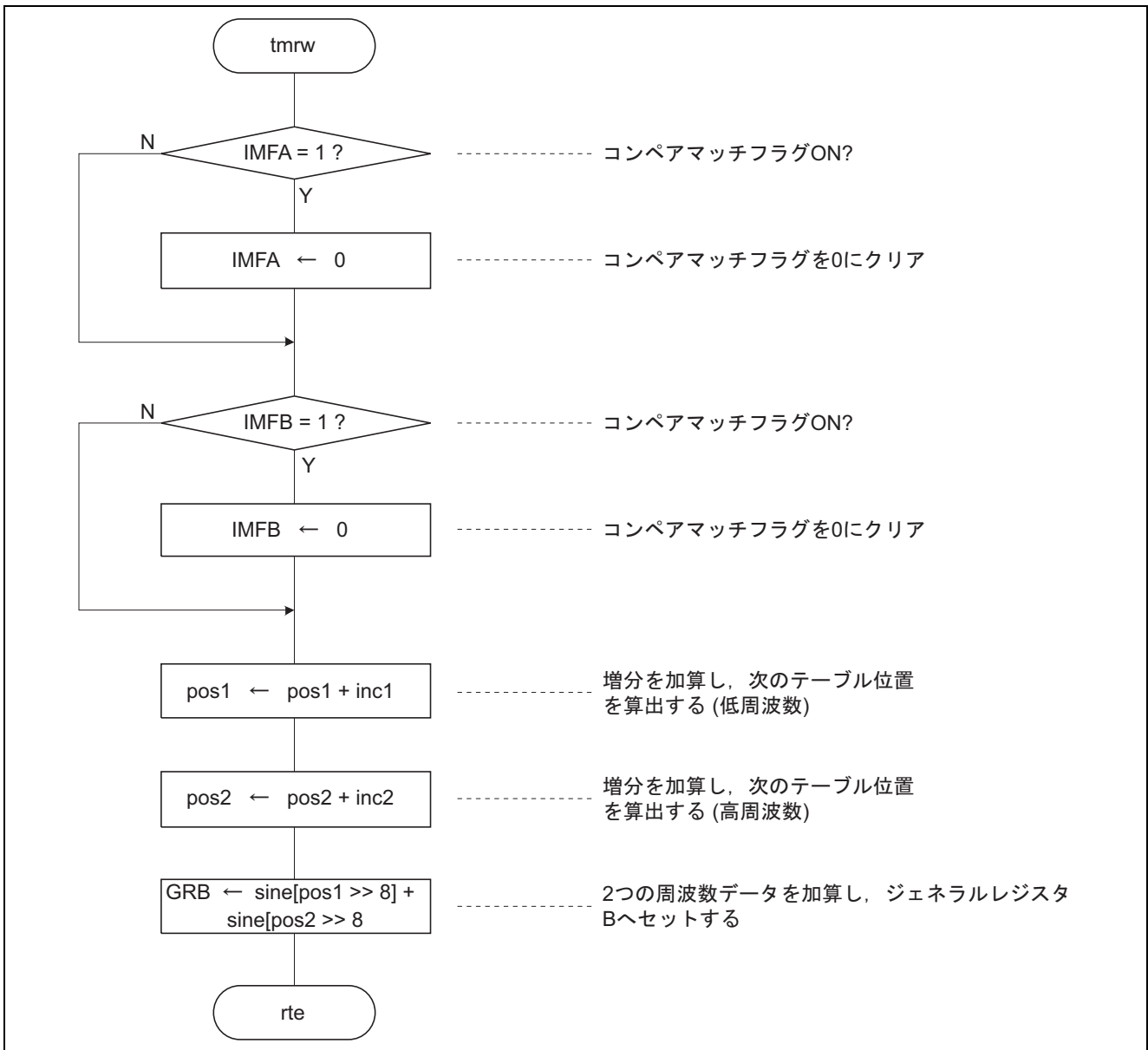




5.2 キーボード入力処理ルーチン (input\_key)



5.3 W タイマ割り込み処理ルーチン (tmrw)





## 6. プログラムリスト

INIT.SRC (プログラムリスト)

```

.export _INIT
.import _main
;
.section P, CODE
_INIT:
mov.w    #h'ff80,r7
ldc.b    #b'10000000, ccr
jmp    @_main
;
.end

```

```
/* H8/300H tiny Series -H8/36014- Application note */
```

```
/* 応用編 */
```

```
/* DTMF 出力例 */
```

```
#include <machine.h>
```

```
/* Symbol definition */
```

```
struct BIT {
    unsigned char b7:1;      /* bit 7 */
    unsigned char b6:1;      /* bit 6 */
    unsigned char b5:1;      /* bit 5 */
    unsigned char b4:1;      /* bit 4 */
    unsigned char b3:1;      /* bit 3 */
    unsigned char b2:1;      /* bit 2 */
    unsigned char b1:1;      /* bit 1 */
    unsigned char b0:1;      /* bit 0 */
};
```

```
#define PDR1 *(volatile unsigned char *)0xFFD4 /* Port data register 1 */
```

```
#define PCR1 *(volatile unsigned char *)0xFFE4 /* Port control register 1 */
```

```
#define PDR7 *(volatile unsigned char *)0xFFDA /* Port data register 7 */
```

```
#define PCR7 *(volatile unsigned char *)0xFFEA /* Port control register 7 */
```

```
#define PDR7_BIT (*(struct BIT *)0xFFDA)
```

```
#define TEST_OUT PDR7_BIT.b4
```

```
#define TMRW *(volatile unsigned char *)0xFF80 /* Timer mode register W */
```

```
#define TMRW_BIT (*(struct BIT *)0xFF80) /* Timer mode register W */
```

```
#define CTS TMRW_BIT.b7 /* Counter Start*/
```

```
#define PWMB TMRW_BIT.b0 /* PWM mode B */
```

```
#define TCRW *(volatile unsigned char *)0xFF81 /* Timer Control Register W */
```

```
#define TCRW_BIT (*(struct BIT *)0xFF81) /* Timer Control Register W */
```

```
#define TIERW *(volatile unsigned char *)0xFF82 /* Timer Interrupt Enable Register W */
```

```
#define TIERW_BIT (*(struct BIT *)0xFF82) /* Timer Interrupt Enable Register */
```

```
#define IMIEA TIERW_BIT.b0 /*input-capture/compare-match interrupt enable A */
```

```
#define TSRW *(volatile unsigned char *)0xFF83 /* Timer Status Register W */
```

```
#define TSRW_BIT (*(struct BIT *)0xFF83) /* Timer Status Register W */
```

```
#define IMFA TSRW_BIT.b0 /* input-capture/compare-match Flag A */
```

```
#define IMFB TSRW_BIT.b1 /* input-capture/compare-match Flag B */
```

```
#define TIOR0 *(volatile unsigned char *)0xFF84 /* Timer I/O control Register 0 */
```

```

#define GRA *(volatile unsigned short *)0xFF88      /* General Register A */
#define GRB *(volatile unsigned short *)0xFF8A      /* General Register B */
#define TCNT*(volatile unsigned short *)0xFF86     /* Timer Counter */

// 3.3V 8MHz      8000000 / 256 = 31250
#define FREQ697    697L * 256 * 256 / 31250        /* 697Hz */
#define FREQ770    770L * 256 * 256 / 31250        /* 770Hz */
#define FREQ852    852L * 256 * 256 / 31250        /* 852Hz */
#define FREQ941    941L * 256 * 256 / 31250        /* 941Hz */
#define FREQ1209   1209L * 256 * 256 / 31250       /* 1209Hz */
#define FREQ1336   1336L * 256 * 256 / 31250       /* 1336Hz */
#define FREQ1477   1477L * 256 * 256 / 31250       /* 1477Hz */
#define FREQ1633   1633L * 256 * 256 / 31250       /* 1633Hz */

#pragma interrupt (tmrw)

/* 関数定義 */
extern void INIT(void);                          /* Stack pointer set */
void main(void);                                 /* main routine */
int input_key(void);                             /* Input key */
void tmrw(void);                                 /* Timer W interrupt routine */

/* ROM define */
const unsigned short PB[16][2] = {               /* Push Button - freq. table */
    FREQ697, FREQ1209,                            /* 1 */
    FREQ697, FREQ1336,                            /* 2 */
    FREQ697, FREQ1477,                            /* 3 */
    FREQ770, FREQ1209,                            /* 4 */
    FREQ770, FREQ1336,                            /* 5 */
    FREQ770, FREQ1477,                            /* 6 */
    FREQ852, FREQ1209,                            /* 7 */
    FREQ852, FREQ1336,                            /* 8 */
    FREQ852, FREQ1477,                            /* 9 */
    FREQ941, FREQ1209,                            /* * */
    FREQ941, FREQ1336,                            /* 0 */
    FREQ941, FREQ1477,                            /* # */
    // FREQ697, FREQ1633                            /* A */
    // FREQ770, FREQ1633,                          /* B */
    // FREQ852, FREQ1633,                          /* C */
    // FREQ941, FREQ1633,                          /* D */
};

const unsigned char sine[256] = {                /* Sine Table */
    0x40, 0x42, 0x43, 0x45, 0x46, 0x48, 0x49, 0x4B,
    0x4C, 0x4E, 0x4F, 0x51, 0x52, 0x54, 0x55, 0x57,
    0x58, 0x59, 0x5B, 0x5C, 0x5E, 0x5F, 0x60, 0x62,
    0x63, 0x64, 0x65, 0x66, 0x68, 0x69, 0x6A, 0x6B,
    0x6C, 0x6D, 0x6E, 0x6F, 0x70, 0x71, 0x72, 0x73,
    0x74, 0x75, 0x75, 0x76, 0x77, 0x77, 0x78, 0x79,
    0x79, 0x7A, 0x7A, 0x7B, 0x7B, 0x7C, 0x7C, 0x7C,
    0x7D, 0x7D, 0x7D, 0x7D, 0x7D, 0x7D, 0x7D, 0x7E,
    0x7D, 0x7D, 0x7D, 0x7D, 0x7D, 0x7D, 0x7D, 0x7C,
    0x7C, 0x7C, 0x7B, 0x7B, 0x7A, 0x7A, 0x79, 0x79,
    0x78, 0x77, 0x77, 0x76, 0x75, 0x75, 0x74, 0x73,
    0x72, 0x71, 0x70, 0x6F, 0x6E, 0x6D, 0x6C, 0x6B,
    0x6A, 0x69, 0x68, 0x66, 0x65, 0x64, 0x63, 0x62,
    0x60, 0x5F, 0x5E, 0x5C, 0x5B, 0x59, 0x58, 0x57,

```

```

0x55, 0x54, 0x52, 0x51, 0x4F, 0x4E, 0x4C, 0x4B,
0x49, 0x48, 0x46, 0x45, 0x43, 0x42, 0x40, 0x3F,
0x3D, 0x3B, 0x3A, 0x38, 0x37, 0x35, 0x34, 0x32,
0x31, 0x2F, 0x2E, 0x2C, 0x2B, 0x29, 0x28, 0x26,
0x25, 0x24, 0x22, 0x21, 0x1F, 0x1E, 0x1D, 0x1B,
0x1A, 0x19, 0x18, 0x17, 0x15, 0x14, 0x13, 0x12,
0x11, 0x10, 0x0F, 0x0E, 0x0D, 0x0C, 0x0B, 0x0A,
0x09, 0x08, 0x08, 0x07, 0x06, 0x06, 0x05, 0x04,
0x04, 0x03, 0x03, 0x02, 0x02, 0x01, 0x01, 0x01,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01,
0x01, 0x01, 0x02, 0x02, 0x03, 0x03, 0x04, 0x04,
0x05, 0x06, 0x06, 0x07, 0x08, 0x08, 0x09, 0x0A,
0x0B, 0x0C, 0x0D, 0x0E, 0x0F, 0x10, 0x11, 0x12,
0x13, 0x14, 0x15, 0x17, 0x18, 0x19, 0x1A, 0x1B,
0x1D, 0x1E, 0x1F, 0x21, 0x22, 0x24, 0x25, 0x26,
0x28, 0x29, 0x2B, 0x2C, 0x2E, 0x2F, 0x31, 0x32,
0x34, 0x35, 0x37, 0x38, 0x3A, 0x3B, 0x3D, 0x3E,
};

/* RAM define */
unsigned short incl; /* incremental width(LOW) */
unsigned short inc2; /* incremental width(HIGH) */
unsigned short pos1; /* table position(LOW) */
unsigned short pos2; /* table position(HIGH) */

int key_data, old_key; /* Input key code New & Old */
int i,j; /* Loop counter */
int key; /* Input key code */

/* Vector address */
#pragma section V1 /* Vector section set */
void (*const VEC_TBL1[])(void) = {
    INIT /* H'0000 Reset vector */
};

#pragma section V2 /* Vector section set */
void (*const VEC_TBL2[])(void) = {
    tmrw /* H'002a Timer W interrupt vector */
};
#pragma section /* P */

```

```

/*****/
/* Main program */
/*****/
void main(void)
{
    set_imask_ccr(1); /* CCR I-bit = 1 */

    pos1 = 0; /* table position(LOW) zero clear */
    pos2 = 0; /* table position(HIGH) zero clear */
    incl = 0; /* incremental width(LOW) zero clear */
    inc2 = 0; /* incremental width(HIGH) zero clear */

    PCR1 = 0x00; /* Port1 0-2bit input */

    PCR7 = 0xff; /* Port7 0-3bit output */

    PWMB = 1; /* PWM output */
    TCRW = 0x82; /* initialize TCRW */
    GRA = 0xff; /* General Register A */
    GRB = 0x00; /* General Register B */
    TCNT = 0; /* initialize Timer Counter */
    IMIEA = 1; /* input-capture/compare-match interrupt enable A */

    key_data = old_key = -1; /* Initialize key data */
    while(1){
        key_data = input_key(); /* Input key */
        if ( key_data != -1) { /* KEY ON */
            if ( old_key != key_data ) {
                incl = PB[key_data][0]; /* Set increase data(LOW) */
                inc2 = PB[key_data][1]; /* Set increase data(HIGH) */
                TCNT = 0; /* initialize Timer Counter */
                CTS = 1; /* Counter Start */
                set_imask_ccr(0); /* CCR I-bit = 0 */
                old_key = key_data; /* Save current key data */
                for(i = 0;i < 100;i++); /* dummy loop */
            }
        } else { /* KEY OFF */
            set_imask_ccr(1); /* CCR I-bit = 1 */
            CTS = 0; /* Counter Stop */
            pos1 = 0; /* table position(LOW) zero clear */
            pos2 = 0; /* table position(HIGH) zero clear */
            incl = 0; /* incremental width(LOW) zero clear */
            inc2 = 0; /* incremental width(HIGH) zero clear */
            old_key = -1; /* Clear old input data */
        }
    }
}

```

```

/*****/
/* input key data */
/*****/
int input_key(void)
{
    key = 0; /* Set start key code */
    for ( j=0x01; j<=0x08; j<<=1 ) {
        PDR7 = j; /* Output level to line */
        for ( i=0x01; i<=0x04; i<<=1 ) {
            if ( PDR1 & i ) { /* Read level from line */
                return key;
            }
            key++;
        }
    }
    return -1;
}

/*****/
/* Timer W Interrupt */
/*****/
void tmrw(void)
{
    if(IMFA == 1){ /* IMFA = 1 ? */
        IMFA = 0; /* Clear IMFA to 0 */
    }
    if(IMFB == 1){ /* IMFB = 1 ? */
        IMFB = 0; /* Clear IMFB to 0 */
    }

    pos1 += incl; /* next pos (LOW) */
    pos2 += inc2; /* next pos (HOGH) */
    /* sine table look up */

    GRB = sine[pos1 >> 8] + sine[pos2 >> 8];
}

```

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](mailto:csc@renesas.com)

## 改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2003.12.22	—	初版発行
2.00	2006.09.01	1, 9	誤記修正

### 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。