

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 SLP シリーズ

Beep 音出力

要旨

8つのキースイッチに Beep 音(ドレミファソラシド)を割り当て、スイッチを押している間、Beep 音を出します。Beep 音は、PWM 出力波形を外部 D/A 変換回路で変換し、スピーカから出力します。

動作確認デバイス

H8/38024

目次

1. 仕様	2
2. 考え方	2
3. 使用機能説明	3
4. 動作説明	8
5. ソフトウェア説明	10
6. フローチャート	13
7. プログラムリスト	16

1. 仕様

- (1) 本タスク例では、8つのキースイッチに Beep 音を割り当て、スイッチを押している間、Beep 音を出力します。
- (2) Beep 音は、PWM 出力波形を外部 D/A 変換回路で変換し、スピーカから出力します。
- (3) PWM 波形は、High 幅を変化させます。High 幅の変化は、サイン波となるようなデータテーブルを、予め ROM に格納しておき、一定間隔ごとに PWM2 端子から出力します。
- (4) Beep 音出力の接続図を図 1 に示します。

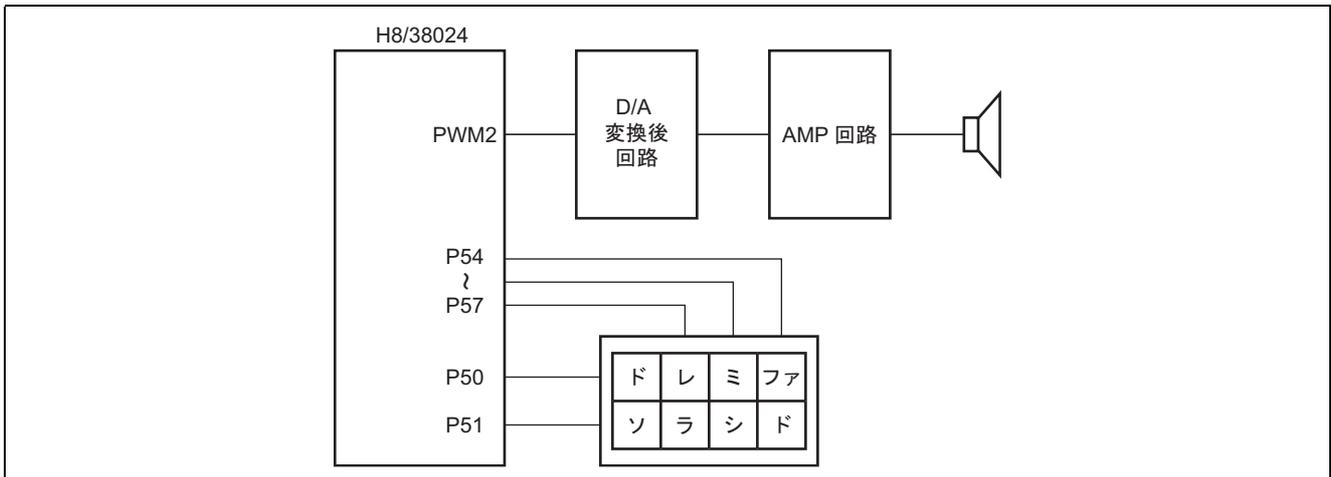


図1 本タスク例の構成

2. 考え方

- (1) PWM の 1 変換周期ごとに、PWM の High 幅を変化させます。
- (2) PWM-High 幅の変化により電圧値が変化するような回路を、外部に接続します。
- (3) 各音階は、PWM-High 幅のデータテーブルを予め ROM に格納しておき、PWM の 1 変換周期ごとに 1 データを出力します。
- (4) PWM の 1 変換周期は、タイマ FH によるコンペアマッチ割り込みにより求めます。

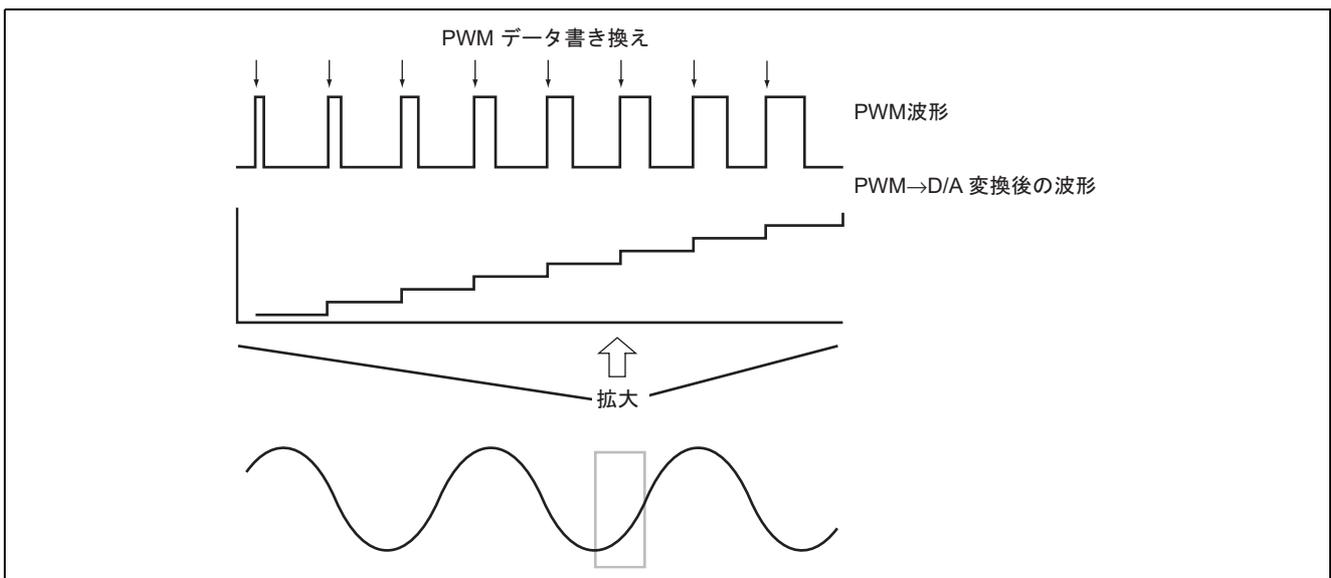


図2 Beep 音出力の波形例

3. 使用機能説明

(1) Beep 音出力における H8/38024 の使用機能について説明します。本タスク例における使用機能のブロック図を図 3 に示します。

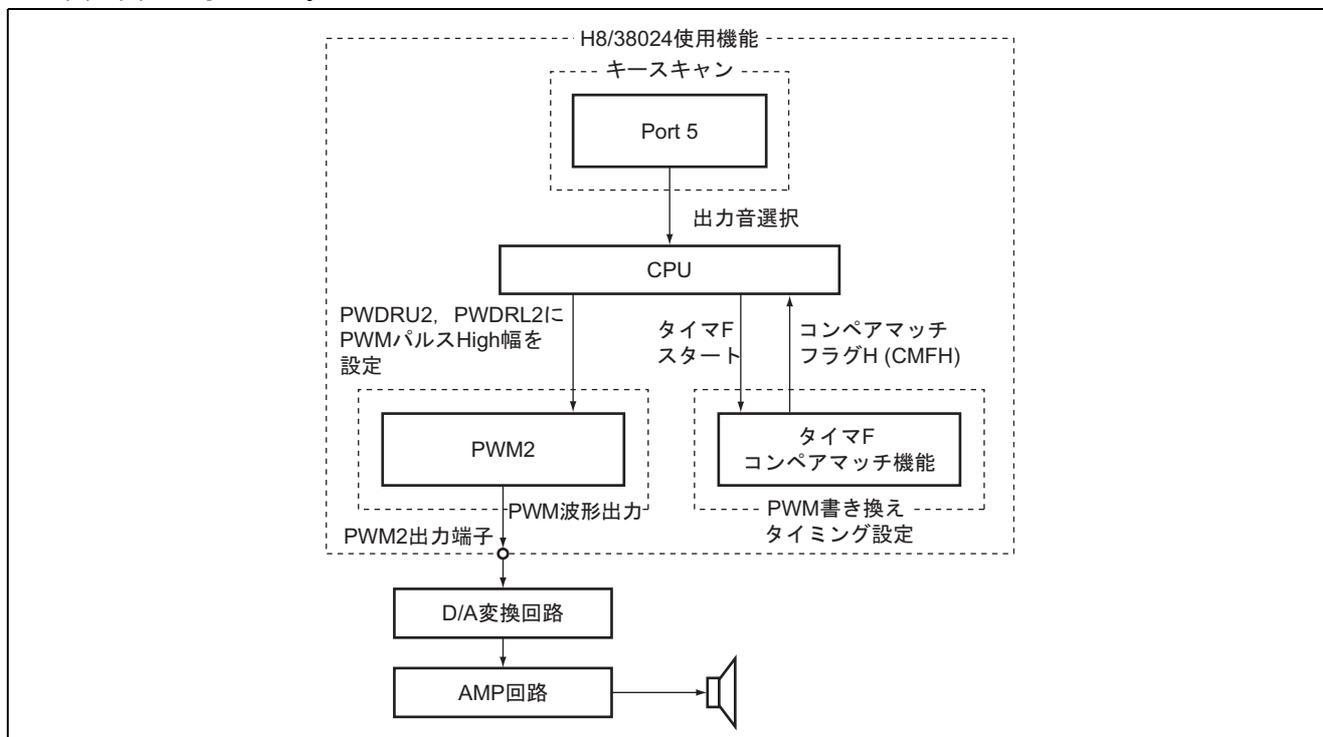


図 3 H8/38024 の使用機能

- ポート 5
キースイッチと接続しており、Beep 音を選択します。
- PWM2
スピーカへ出力するサイン波を生成するための PWM 波形を出力します。
- タイマ F コンペアマッチ機能
PWM の 1 変換周期毎に PWM パルス High 幅を変更するため、一定周期でフラグセットします。

(2) PWM2 の各機能を説明します。

(a) 10 ビット PWM 機能のブロック図を図 4 に示します。以下に 10 ビット PWM 機能のブロック図について説明します。

- システムクロック()は、5MHz のクロックで、CPU および周辺機能を動作させるための基準クロックです。
- PWM2 コントロールレジスタ(PWCR2)は、8 ビットのライト専用レジスタで、入力クロックの選択を行います。
- リップル低減をはかったパルス分割方式を使用しています。
- PWM2 データレジスタ U, L (PWDRU2, PWDRL2)は、ライト専用の 10 ビットのレジスタで、PWDRU2 が上位 2 ビット、PWDRL2 が下位 8 ビットの構成になっています。PWDRU2, PWDRL2 に書き込まれた内容は PWM 波形 1 周期の"High"レベル幅の合計に対応しています。PWDRU2, PWDRL2 に 10 ビットのデータをライトすると PWDRU2, PWDRL2 の内容が PWM 波形生成部に取り込まれ、PWM 波形を生成するデータが更新されます。また、10 ビットデータの設定は必ず、「PWDRL2 への下位 8 ビットのデータをライト」、「PWDRU2 への上位 2 ビットのデータをライト」の順で行わなければなりません。
- ポートモードレジスタ 9(PMR9)は、8 ビットのリード/ライト可能なレジスタで、ポート 9 の各端子機能の切り換えを制御します。PMR9 ビット 1 の P91/PWM2 端子機能切り換え(PWM2)により、P91/PWM2 端子を PWM2 出力端子機能に設定します。
- PWM2 出力端子(PWM2)から、パルス分割方式 PWM 波形を出力します。

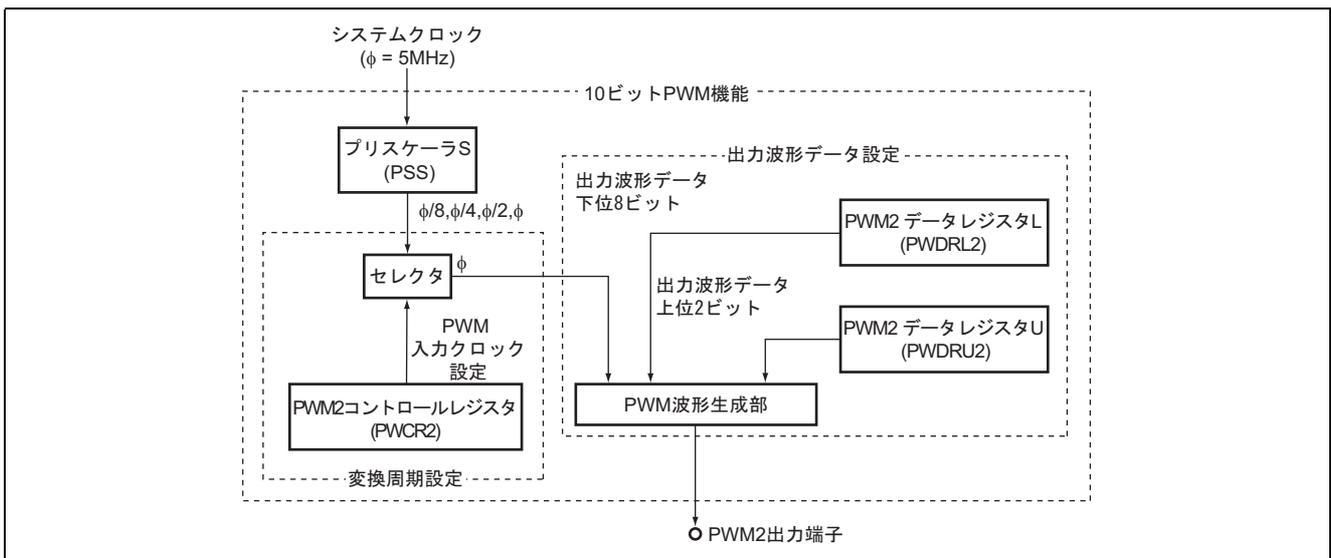


図 4 PWM2 のブロック図

(3) ポート 5 の機能を説明します。

(a) ポート 5 によるキー入力回路のブロック図を図 5 に示します

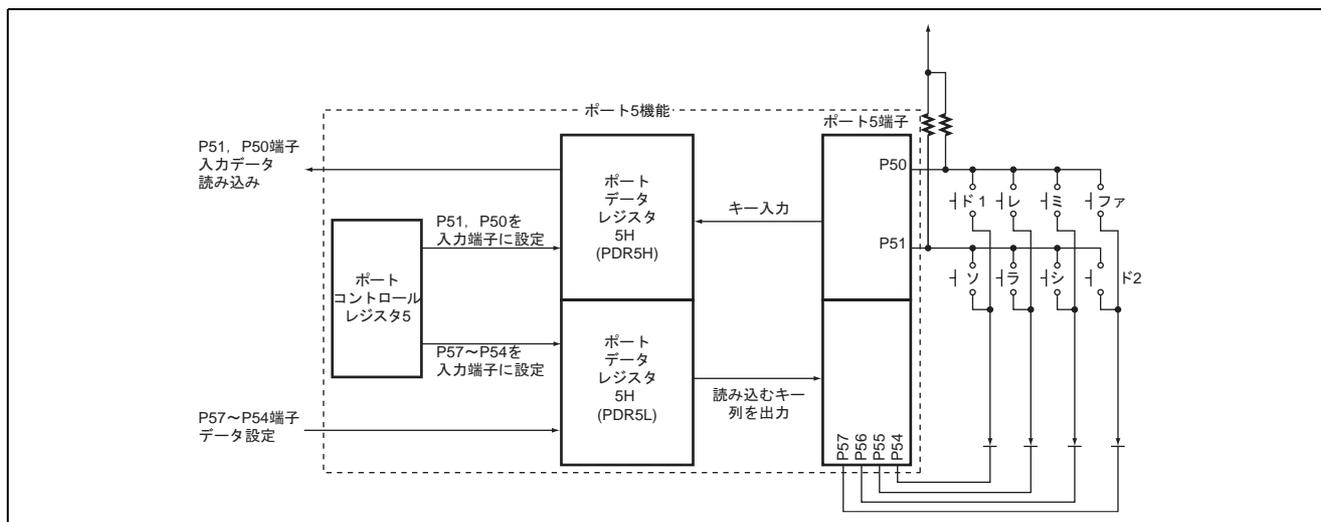


図 5 ポート 5 によるキー入力回路のブロック図

(b) 本タスク例では、ポート 5 を使用して、キー入力を行います。以下にキー入力回路のブロック図について説明します。

- ポートコントロールレジスタ 5 (PCR5)
ポート 5 の入出力設定を行います。PCR5=0xF0 とし、P57 ~ P54 を出力ポート、P53 ~ P50 を入力ポートに設定します。
- ポートデータレジスタ 5H (PDR5H)
PDR5 の上位 4 ビットを PDR5H とし、出力ポート P57 ~ P54 に格納するデータを設定します。P57 ~ P54 のうち、Low レベルに設定したキー列の状態が P51, P50 に反映されます。
- ポートデータレジスタ 5L (PDR5L)
PDR5 の下位 4 ビットを PDR5L とし、入力ポート P53 ~ P50 の値は、PDR5L に反映されます。PDR5L に格納された値は、PDR5H で選択されたキー列の状態です。本タスク例では、P51, P50 のみを使用します。

(4) タイマ F の機能を説明します

(a) タイマ F コンペアマッチ機能のブロック図を図 6 に示します。

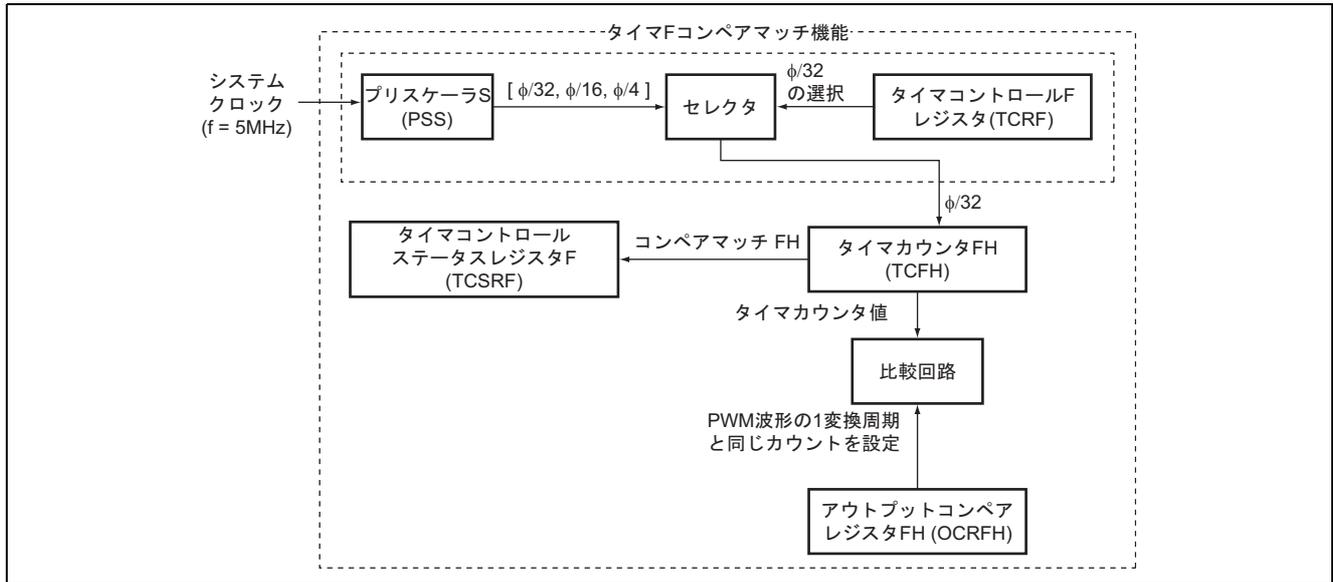


図 6 タイマ F コンペアマッチ機能のブロック図

(b) 本タスク例では、タイマ F コンペアマッチ機能を使用して、PWM の 1 変換周期である 102.4ms 毎に割り込みを発生させ、PWM パルス High 幅の変更を行います。

- システムクロック ()
5MHz で、CPU および周辺機能を動作させるための基準クロックです。
- プリスケール S (PSS)
c を入力とする 13 ビットのカウンタで、1 サイクルごとにカウントアップします。
- タイマカウンタ F (TCF)
16 ビットのリード/ライト可能なアップカウンタです。入力するクロックは、 /4、 /16、 /32 およびウォッチクロック ω_w の計 4 種類を選択可能です。本タスク例では、TFC の入力クロックに /4 を選択しています。
- タイマコントロールレジスタ F (TCRF)
8 ビットのリード/ライト可能なレジスタで、16 ビットモード、8 ビットモードの切り換え、4 種類の内部クロックおよび外部イベントの選択、TMOFH、TMOFL 端子の出力レベルの設定を行います。
- タイマコントロールステータスレジスタ F (TCSRf)
8 ビットのレジスタで、カウンタクリアの選択、オーバフローフラグのセット、コンペアマッチフラグのセット、オーバフローによる割り込み要求の許可の制御を行います。

(5) 本タスク例の機能割り付けを表 1 に示します。表 1 に示すように機能を割り付け、PWM 波形による Beep 音出力を行います。

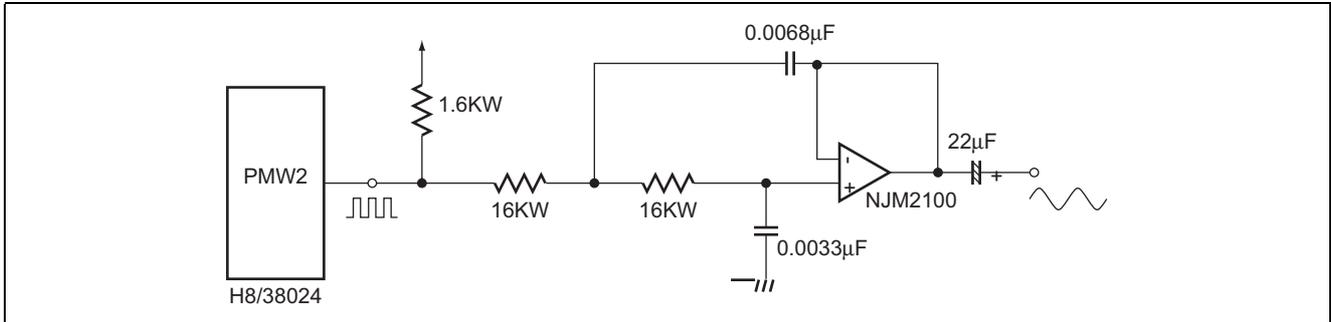
表1 機能割り付け

機能	機能割り付け
PSS	システムクロック(5MHz)を入力とする 13 ビットのアップカウンタ
TCRF	タイマ F に供給されるクロックを選択
TCSRFB	コンペアマッチ(102.4 μs 毎)による TCFH のクリアとフラグセット
TCFH	タイマ F カウンタ
OCRFB	PWM の 1 周期 102.4 μs となるように設定
PWCR2	10 ビット PWM に供給されるクロックを選択
PWDRU2	PWM 出力波形データの上位 2 ビットの設定
PWDRL2	PWM 出力波形データの下位 8 ビットの設定
PWM2 出力端子	PWM 波形出力端子
PCR5	ポート 5 の入出力機能を設定
PDR5H,PDR5L	2×4 のキースキャン
PMR5	ポート 5 入出力端子の設定
PUCR5	ポート 5 プルアップ MOS を OFF に設定

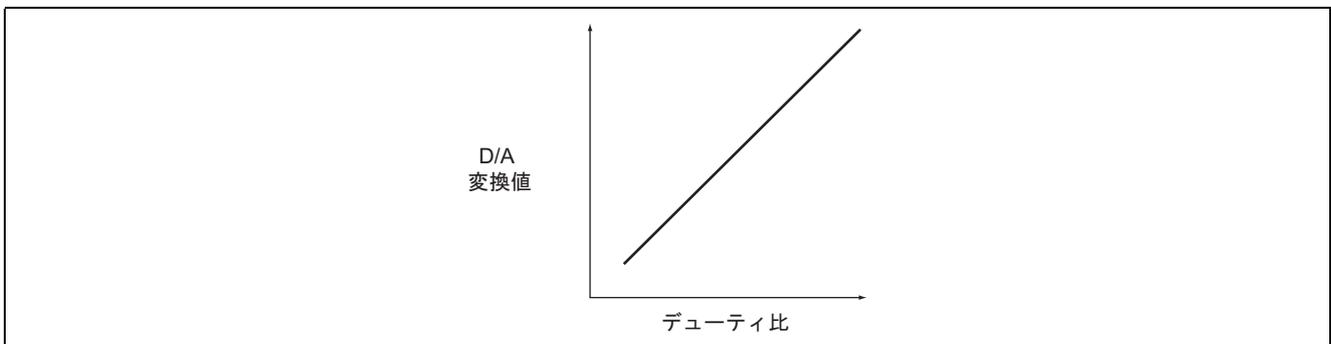
4. 動作説明

(1) Beep 音出力の動作説明を示します。

(a) PWM 波形をアナログ変換する PWM-D/A 変換回路を下図に示します。



- 本タスク例の PWM 最小変化幅は、25.6 μS です。PWM は、同じ波形を 4 回ずつ出力します。
- 本回路では、デューティ比が小さいとき、D/A 変換値が低く、デューティ比が大きいとき、D/A 変換値が高くなります。



(b) 各音階に対応する出力パルスの周波数を表 2 に示します。

表 2 各音階に対応する出力パルスの周波数

音階 項目	ド 1	レ	ミ	ファ	ソ	ラ	シ	ド 2
周期[ms]	1.95	1.74	1.54	1.43	1.23	1.13	1.02	0.92
周波数[Hz]	514	574	651	698	814	888	977	1085
1 周期サンプル数	19	17	15	14	12	11	10	9

- 周期，周波数計算方法

本タスク例での PWM パルス 1 変換周期

$$512 / 5 = 512 / 5\text{MHz} = 102.4 \mu\text{s}$$

各音階の周期

$$\text{音階の 1 周期サンプル数} \times 102.4 \mu\text{s}$$

各音階の周波数

$$1 / \text{音階の周期}$$

例：

音階"ド 1"の周期

$$\text{音階の 1 周期サンプル数} \times 102.4 \mu\text{s} = 19 \times 102.4 \mu\text{s} = 1.945\text{ms} = 1.95\text{ms}$$

音階"ド 1"の周波数

$$1 / \text{音階の周期} = 1 / 1.945\text{ms} = 514\text{Hz}$$

(c) データテーブル一覧

各音階に対応する PWDRU2,PWDRL2 の設定値を表 3 に示します。

表 3 各音階に対応する PWDRU2,PWDRL2 の設定値

音階 No	ド 1	レ	ミ	ファ	ソ	ラ	シ	ド 2
1	527	527	527	527	527	527	527	527
2	627	639	653	661	682	695	709	725
3	716	735	757	769	795	809	821	832
4	785	804	821	829	835	830	822	797
5	827	836	835	829	796	761	711	638
6	836	826	796	770	684	614	530	427
7	812	776	711	662	530	440	348	262
8	758	692	594	528	375	293	233	220
9	679	587	466	393	260	220	231	320
10	584	473	348	285	220	245	341	
11	482	367	260	225	256	359		
12	386	282	220	224	367			
13	304	230	231	284				
14	247	220	293	391				
15	220	247	396					
16	224	314						
17	262	410						
18	328							
19	415							

5. ソフトウェア説明

(1) モジュール説明

本タスク例のモジュールを表 4 に示します。

表 4 モジュール説明

モジュール名	関数名	機能
メインルーチン	main	グローバル変数, PWM2 設定を行う
Beep 音出力	beep	PWM2 デューティ設定ルーチン
ポート 5L 読み取り	keyread	P53 ~ P50 の内容を戻す

(2) 引数の説明

表 5 引数の説明

引数名	機能	関数名	データ長	入出力
beep_no	ビープ音の選択	beep	1 バイト	入力

(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 6 に示します。

表 6 使用内部レジスタ説明

レジスタ名		機能	アドレス	設定値
TCRF	CKSH2 CKSH1 CKSH0	タイマコントロールレジスタ F (クロックセレクト H) : CKSH2=1, CKSH1=1, CKSH0=0 のとき, TCFH は, 内部クロック /4 でカウントする	0xFFB6 ビット 6 ビット 5 ビット 4	CKSH2=1 CKSH1=1 CKSH0=0
	OVFH	タイマコントロールステータスレジスタ F (タイマオーバーフローフラグ H) : OVFH=0 のとき, TCF がオーバーフローしていない : OVFH=1 のとき, TCF がオーバーフローした	0xFFB7 ビット 7	0
TCSRFB	CMFH	タイマコントロールステータスレジスタ F (コンペアマッチフラグ H) : CMFH=0 のとき, コンペアマッチ F が発生していない : CMFH=1 のとき, コンペアマッチ F が発生した	0xFFB7 ビット 6	0
	OVIEH	タイマコントロールステータスレジスタ F (タイマオーバーフローインターラプトイネーブル H) : OVIEH=0 のとき, TCF のオーバーフローによる割り 込み要求を禁止 : OVIEH=1 のとき, TCF のオーバーフローによる割り 込み要求を許可	0xFFB7 ビット 5	0
	CCLRHB	タイマコントロールステータスレジスタ F (カウンタクリア H) : CCLRHB=0 のとき, コンペアマッチによる TCFH の クリアを禁止 : CCLRHB=1 のとき, コンペアマッチによる TCFH の クリアを許可	0xFFB7 ビット 4	1
TCFH		8 ビットタイマカウンタ FH : /4 を入力とする 8 ビットのアップカウンタ	0xFFB8	0x00
OCRFH		アウトプットコンペアレジスタ FH : TCFH と比較される。OCRFH と TCFH の値が一 致すると, TCSRFB の CMFH が 1 にセットされる。	0xFFBA	0x80
PMR5		ポートモードレジスタ 5 (P5n/~WKPn/SEGn+1 端子機能切り換え) ~SEGn+1 端子としてしない場合の端子機能 : PMR5=0x00 のとき, P5n/~WKPn/SEGn+1 端子 は, P5n 入出力端子として機能	0xFFCC	0x00
PWCR2	PWCR20 PWCR21	PWM2 コントロールレジスタ(クロックセレクト 1,0) : PWCR20=1, PWCR21=1 のとき, 10 ビット PWM に供給されるクロックを に設定	0xFFCD ビット 1 ビット 0	PWCR10=1 PWCR11=1

表 6 使用内部レジスタ説明(つづき)

レジスタ名	機能	アドレス	設定値
PWDRU2	PWM2 データレジスタ U : PWM 出力波形データの上位 2 ビットを設定	0xFFCE	0x00
PWDRL2	PWM2 データレジスタ L : PWM 出力波形データの下位 8 ビットを設定。	0xFFCF	0x00
PDR5	PDR5H ポートデータレジスタ 5(P57~P54) ~ PDR5 の上位 4 ビットを PDR5H とし, 出力ポート P57~P54 に格納するデータを設定する。P57~P54 のデータのうち, Low レベルに設定したキー列の状態が, P53~P50 に反映される。 : PDR5=0xE0 のとき, P54 のキー列(0,4)を選択する。 : PDR5=0xD0 のとき, P55 のキー列(1,5)を選択する。 : PDR5=0xB0 のとき, P56 のキー列(2,6)を選択する。 : PDR5=0x70 のとき, P57 のキー列(3,7)を選択する。	0xFFD8 ビット 7 ~ ビット 4	—
	PDR5L ポートデータレジスタ(P53~P50) ~ PDR5 の下位 4 ビットを PDR5L とし, 入力ポート P53~P50 の値が反映される。	0xFFD8 ビット 3 ~ ビット 0	—
PUCR5	ポートプルアップコントロールレジスタ 5 : PUCR5 = 0x00 のとき, プルアップ MOS を OFF にする。	0xFFE2	0x00
PCR5	ポートコントロールレジスタ 5 : PCR5=0xF0 のとき, P57~P54 を出力ポート, P53~P50 を入力ポートに設定。	0xFFE8	0xF0
PMR9	PWM2 ポートモードレジスタ 9(P91/PWM2 端子機能切り換え) : PWM2=0 のとき, P91/PWM2 端子を P91 出力端子機能に設定。 : PWM2=1 のとき, P91/PWM2 端子を PWM2 出力端子機能に設定。	0xFFEC ビット 1	1

(4) 使用 RAM 説明

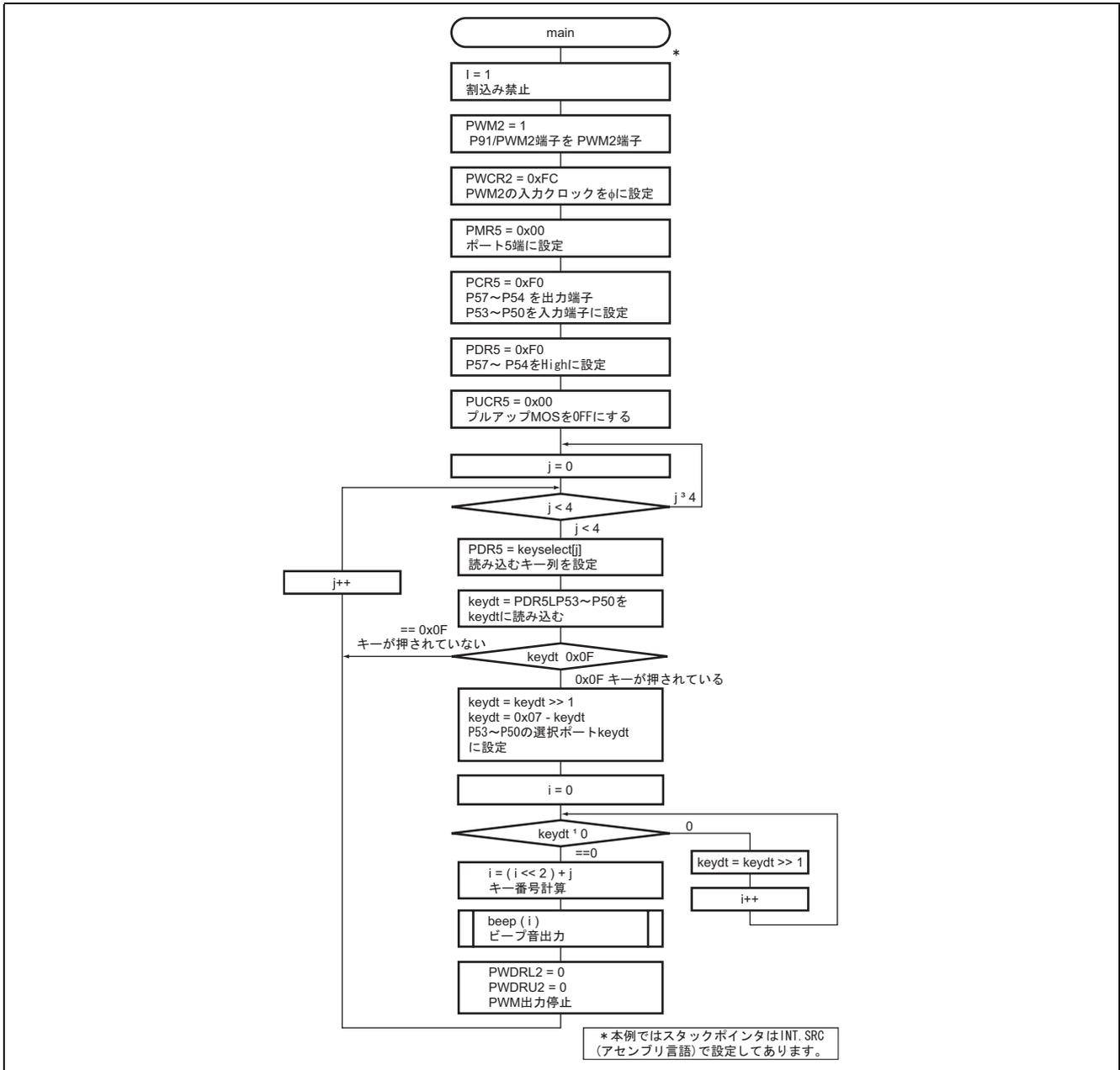
本タスク例の使用 RAM を表 7 に示します。

表 7 使用 RAM 説明

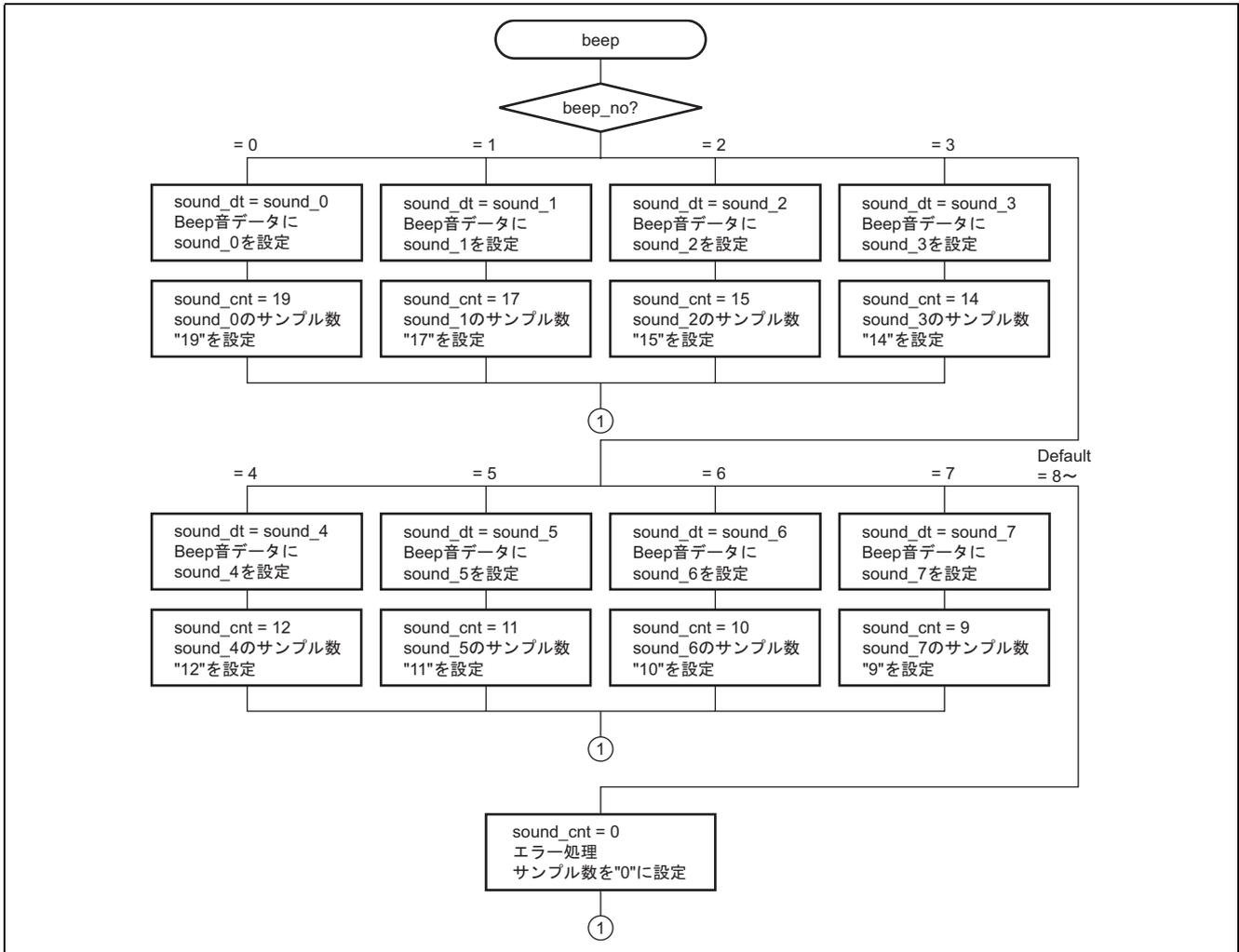
ラベル名	機能	メモリ消費量	使用モジュール名
keyselect	読み込みキー列を選択	4 バイト	main

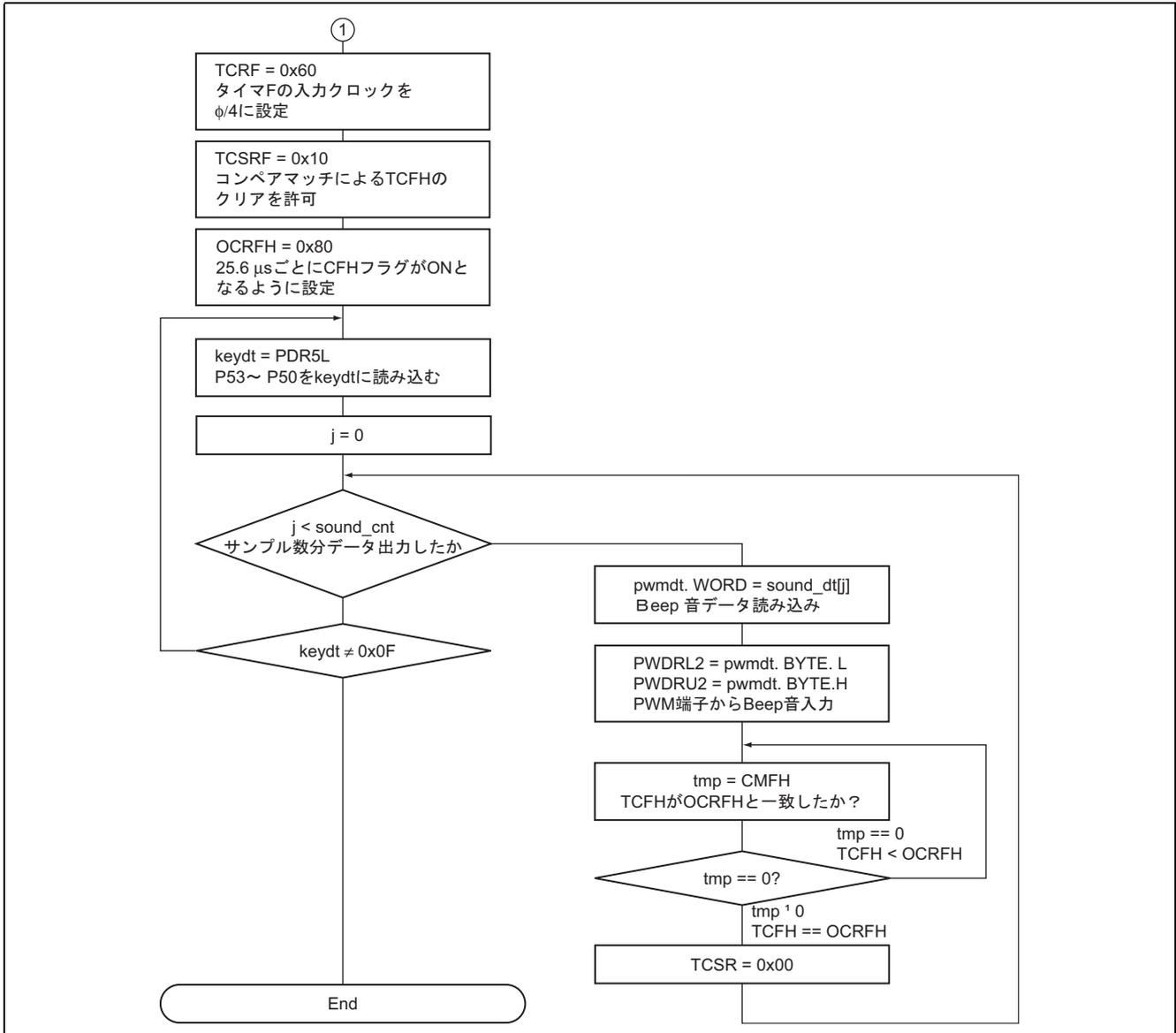
6. フローチャート

(1) メインルーチン

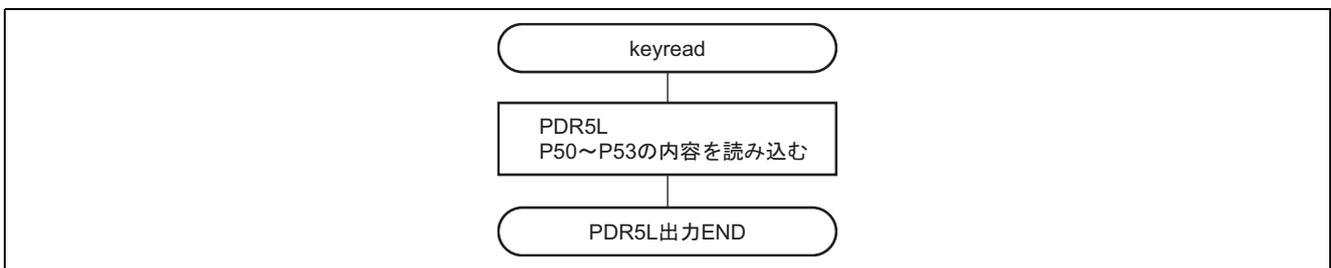


(2) Beep 音出力





(3) ポート 5L 読み取り



7. プログラムリスト

7.1 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

/*****/
/*                                     */
/* H8/300L Super Low Power Series      */
/*   -H8/38024 Series-                 */
/* Application Note                     */
/*                                     */
/* 'Beep Sound Output by 10-bit PWM Function' */
/*                                     */
/* Function                             */
/* : 10bit PWM                          */
/*                                     */
/* External Clock : 10MHz               */
/* Internal Clock : 5MHz                */
/* Sub Clock      : 32.768kHz           */
/*                                     */
/*****/

#include <machine.h>

/*****/
/* Symbol Definition                    */
/*****/
struct BIT {
    unsigned char  b7:1;    /* bit7 */
    unsigned char  b6:1;    /* bit6 */
    unsigned char  b5:1;    /* bit5 */
    unsigned char  b4:1;    /* bit4 */
    unsigned char  b3:1;    /* bit3 */
    unsigned char  b2:1;    /* bit2 */
    unsigned char  b1:1;    /* bit1 */
    unsigned char  b0:1;    /* bit0 */
};

struct P4BIT {
    unsigned char  H:4;     /* bit7-bit4 */
    unsigned char  L:4;     /* bit3-bit0 */
};

#define TCRF      *(volatile unsigned char *)0xFFB6 /* Timer Control Register F */
#define TCRF_BIT  (*(struct BIT *)0xFFB6)          /* Timer Control Register F */
#define TOLH      TCRF_BIT.b7                     /* Toggle Output Level F */
#define CKSH2     TCRF_BIT.b6                     /* Clock Select H2 */
#define CKSH1     TCRF_BIT.b5                     /* Clock Select H1 */

```

```

#define CKSH0      TCRF_BIT.b4                /* Clock Select H0                */
#define TCSRFB    *(volatile unsigned char *)0xFFB7 /* Timer Control Status Register F */
#define TCSRFB_BIT (*(struct BIT *)0xFFB7)      /* Timer Control Status Register F */
#define OVFB      TCSRFB_BIT.b7              /* Timer Overflow Flag H          */
#define CMFB      TCSRFB_BIT.b6              /* Compare Match Flag H          */
#define OVIEH     TCSRFB_BIT.b5              /* Timer Overflow Interrupt Enable */
#define CCLRHB    TCSRFB_BIT.b4              /* Output Select 3                */
#define TCFB      *(volatile unsigned char *)0xFFB8 /* Timer Counter FH                */
#define OCRFB     *(volatile unsigned char *)0xFFBA /* Output Compare Register FH     */
#define PMR5      *(volatile unsigned char *)0xFFCC /* Port mode register 5           */
#define PWCR2     *(volatile unsigned char *)0xFFCD /* PWM2 Control Register          */
#define PWCR2_BIT (*(struct BIT *)0xFFCD)      /* PWM2 Control Register          */
#define PWCR21    PWCR2_BIT.b1              /* Clock Select                    */
#define PWCR20    PWCR2_BIT.b0              /* Clock Select                    */
#define PWDRU2    *(volatile unsigned char *)0xFFCE /* PWM Data Registers U           */
#define PWDRL2    *(volatile unsigned char *)0xFFCF /* PWM Data Registers L           */
#define PDR5      *(volatile unsigned char *)0xFFD8 /* Port data register 5           */
#define PDR5_BIT  (*(struct P4BIT *)0xFFD8)     /* Port mode register 5           */
#define PDR5H     PDR5_BIT.H                /* P57-P54                         */
#define PDR5L     PDR5_BIT.L                /* P53-P50                         */
#define PUCR5     *(volatile unsigned char *)0xFFE2 /* Port pull-up control register 5 */
#define PCR5      *(volatile unsigned char *)0xFFE8 /* Port control register 5        */
#define PMR9_BIT  (*(struct BIT *)0xFFEC)      /* Port Mode Register 9           */
#define PWM2      PMR9_BIT.b1                /* P91/PWM2 pin function switches */

/*****/
/* Function define */
/*****/
extern void INIT ( void ); /* SP Set */
void main( void );
void beep( unsigned char beep_no );
unsigned char keyread( void );

unsigned char keyselect[4] = {
    0xE0,
    0xD0,
    0xB0,
    0x70,
};

/*****/
/* Vector Address */
/*****/
#pragma section V1 /* VECTOR SECTOIN SET */
void (*const VEC_TBL1[])(void) = {
    INIT /* 00 Reset */
};

extern unsigned short sound_0[19]; /* Saund Sample Data */
extern unsigned short sound_1[17]; /* Saund Sample Data */
extern unsigned short sound_2[15]; /* Saund Sample Data */
extern unsigned short sound_3[14]; /* Saund Sample Data */
extern unsigned short sound_4[12]; /* Saund Sample Data */
extern unsigned short sound_5[11]; /* Saund Sample Data */
extern unsigned short sound_6[10]; /* Saund Sample Data */
extern unsigned short sound_7[9]; /* Saund Sample Data */

```

```

#pragma section /* P */
/*****/
/* Main Program */
/*****/
void main( void )
{
    unsigned char i,j,keydt;

    set_imask_ccr(1); /* Interrupt Disable */

    PWM2 = 1; /* Pin function Select PWM2 */
    PWCR2 = 0xFC; /* Initialize PWM Input Clock */

    PMR5 = 0x00; /* Pin function Select Port5 */
    PCR5 = 0xF0; /* P57-54 Output,P53-50 Input Port */
    PDR5 = 0xF0; /* P57-54 Port "1"set */
    PUCR5 = 0x00; /* Port5 pull-up OFF */

    while(1){
        for(j = 0; j < 4; j++){
            PDR5 = keyselect[j]; /* Set Key Select */

            keydt = keyread();
            if(keydt != 0x0F){ /* Touch Key? */
                keydt = keydt>>1; /* What Key? */
                keydt = 0x07 - keydt;
                for(i = 0; keydt != 0; i++){
                    keydt = keydt>>1;
                }

                i = i<<2;
                i = i+j;
                beep(i); /* Beep Sound Output */

                PWDRL2 = 0; /* Stop PWM */
                PWDRU2 = 0;
            }
        }
    }

/*****/
/* Beep Sound Output */
/*****/
void beep( unsigned char beep_no )
{
    unsigned char i,j,keydt,sound_cnt,tmp;
    unsigned short *sound_dt;
    union REG{
        unsigned short WORD;
        struct {
            unsigned char H;
            unsigned char L;
        }BYTE;
    };
    union REG pwmtd;

```

```

switch ( beep_no ){
  case 0:                                /* Select Beep Sound 0      */
    sound_dt = sound_0;
    sound_cnt = 19;
    break;

  case 1:                                /* Select Beep Sound 1      */
    sound_dt = sound_1;
    sound_cnt = 17;
    break;

  case 2:                                /* Select Beep Sound 2      */
    sound_dt = sound_2;
    sound_cnt = 15;
    break;

  case 3:                                /* Select Beep Sound 3      */
    sound_dt = sound_3;
    sound_cnt = 14;
    break;

  case 4:                                /* Select Beep Sound 4      */
    sound_dt = sound_4;
    sound_cnt = 12;
    break;

  case 5:                                /* Select Beep Sound 5      */
    sound_dt = sound_5;
    sound_cnt = 11;
    break;

  case 6:                                /* Select Beep Sound 6      */
    sound_dt = sound_6;
    sound_cnt = 10;
    break;

  case 7:                                /* Select Beep Sound 7      */
    sound_dt = sound_7;
    sound_cnt = 9;
    break;

  default:                               /* Select Beep Sound Error? */
    sound_cnt = 0;
    break;
}

TCRF = 0x60;                             /* Initialize Clock Select   */
TCSRFB = 0x10;                           /* Initialize Overflow Interrupt */
OCRFBH = 0x80;                           /* Initialize Compare MatchFH Value */

do{
  keydt = PDR5L;
  for(j = 0; j < sound_cnt; j++){
    pwmdt.WORD = sound_dt[j];
    PWDRL2 = pwmdt.BYTE.L;                /* Set PWM Output Data      */
    PWDRU2 = pwmdt.BYTE.H;
    do{

```

```

        tmp = CMFH;
    }while( tmp == 0);
    TCSRFR = 0x10;                /* Initialize Overflow Interrupt */
}
}while(keydt != 0x0F);          /* Touch Key? */
}

/*****/
/* KeyRead */
/*****/
unsigned char keyread( void )
{
    return(PDR5L);
}
/*****/
/* Saund Sample Data */
/*****/
#pragma section PCMDT
unsigned short  sound_0[19] =
{
527,
627,
716,
785,
827,
836,
812,
758,
679,
584,
482,
386,
304,
247,
220,
224,
262,
328,
415,
};

unsigned short  sound_1[17] =
{
527,
639,
735,
804,
836,
826,
776,
692,
587,
473,
367,
282,
230,
220,

```

```
247,  
314,  
410,  
};
```

```
unsigned short sound_2[15] =  
{  
527,  
653,  
757,  
821,  
835,  
796,  
711,  
594,  
466,  
348,  
260,  
220,  
231,  
293,  
396,  
};
```

```
unsigned short sound_3[14] =  
{  
527,  
661,  
769,  
829,  
829,  
770,  
662,  
528,  
393,  
285,  
225,  
224,  
284,  
391,  
};
```

```
unsigned short sound_4[12] =  
{  
527,  
682,  
795,  
835,  
796,  
684,  
530,  
375,  
260,  
220,  
256,  
367,  
};
```

```

unsigned short  sound_5[11] =
{
527,
695,
809,
830,
761,
614,
440,
293,
220,
245,
359,
};

```

```

unsigned short  sound_6[10] =
{
527,
709,
821,
822,
711,
530,
348,
233,
231,
341,
};

```

```

unsigned short  sound_7[9] =
{
527,
725,
832,
797,
638,
427,
262,
220,
320,
};

```

リンクアドレス指定

セクション名	アドレス
CV1	0x0000
P,C,D,DPCMDT	0x0100

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2003.12.19	—	初版発行

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、慎重に制作したものです。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。