

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

## 焦電型赤外線センサ接続例

### 要旨

焦電型赤外線センサをアナログ入力端子に接続し、7セグメント LED に表示します。

### 動作確認デバイス

H8/36014

### 目次

1. 仕様 .....	2
2. 使用機能説明 .....	6
3. 動作原理 .....	8
4. ソフトウェア説明 .....	10
5. フローチャート .....	15
6. プログラムリスト .....	17

### 1. 仕様

1. 図 1.1 に焦電型赤外線センサ接続例のハードウェア構成を示します。図 1.1 に示すように、センサをアナログ入力端子 1 (AN1 端子) に接続します。
2. AN1 端子の A/D 変換を行い、A/D 変換結果を I/O ポートに接続された 7 セグメント LED に表示します。
3. 7 セグメント LED 表示は、A/D 変換結果の 10 ビットデータを 16 進数で表示します。
4. A/D 変換は、0.49152s ごとに行います。

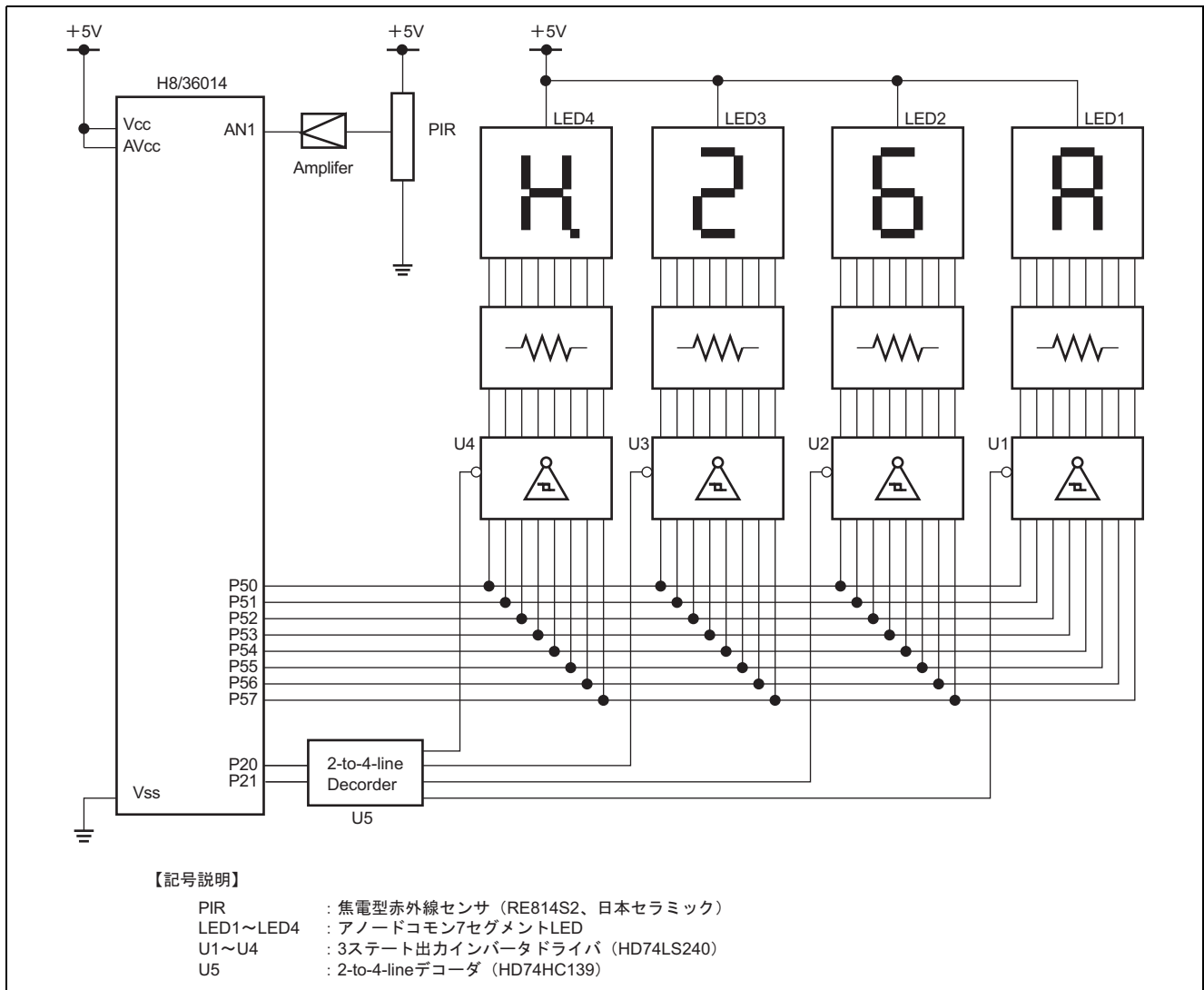


図 1.1 ハードウェア構成

5. 本タスク例における H8/36014 の動作電圧 (Vcc) およびアナログ電源電圧 (AVcc) は 5.0V、OSC クロック周波数は 16MHz です。

6. 本タスク例で使用している焦電型赤外線センサは、日本セラミック（株）製のデュアルタイプ焦電型赤外線センサ（型名：RE814S）です。以下に使用している焦電型赤外線センサの仕様を示します。
- A. 表 1.1 に焦電型赤外線センサ（RE814S）の定格を示します。

表 1.1 定格

型名	RE814S	
有効受光面サイズ	2×1mm（×2）	
感度波長	5～20 μm	
感度	≥1500V/W	
動作電圧	3.0～15.0V DC	
オフセット電圧	0.2V～1.0V	
保証温度範囲	動作	-20～+60℃
	保存	-30～+85℃

- B. 図 1.2 にアプリケーション参考回路を示します。

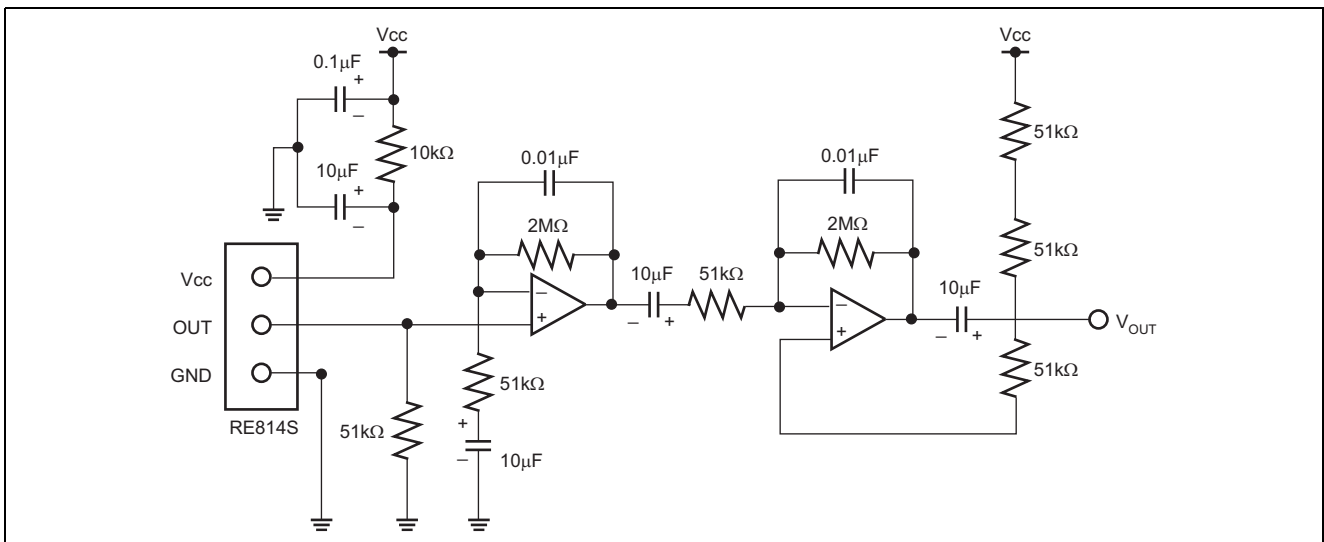


図 1.2 センサ出力アンプ回路例（参考）

7. 焦電センサの特徴を以下に示します。
- A. 誘電体の自発分極が変化することで、表面に帯電している電荷が増減する現象を焦電効果といいます。
- B. 本タスク例で使用している焦電型赤外線センサ（型名：RE814S）はデュアルタイプで、同一の形状のセンサ素子を電極が逆になるように付けられています。この接続方法は背景のゆらぎの影響を防止する効果や、外乱光などが急に入射したときの誤信号を防止する利点があります。
- C. デュアル素子は同時に二つの受光電極へ赤外線が当たった場合は相互に打ち消しあい、信号は出力されません。このため、人が動いたときだけ出力が得られるような使い方をします（人が静止している場合は信号が出力されません）。
8. 本タスク例の動作例は以下の通りです。
- A. センサで人体から発生する赤外線を検出すると、図 1.2 の回路で約 1600 倍に反転増幅されてアナログポートに入力されます。
- B. センサが人体を検出しない場合や人体が静止している場合は、センサからの出力がないため、電源電圧の 1/3 の電圧（オフセット値）がアナログポートに入力されます。H8/36014 の場合のオフセット値は  $5V/3=1.67V$  で LED 表示は「H'155」となります。
- $$\frac{1.67 (V) \times 1023}{5 (V)} = 341 = H'155$$
- C. センサが人体を検出した場合は、出力電圧が反転増幅されて LED 表示は「H'155」より小さい値になります。
- D. 人体が通過した場合の LED 表示は、一度オフセット値「H'155」より小さい値になり、次にオフセット値「H'155」より大きい値になり、最終的に出力電圧はオフセット値に収束し LED 表示は「H'155」となります。

(9) 本タスク例では、7セグメント LED を表示させるためにポート出力を 3 ステート出力インバータドライバ (HD74LS240) に入力して、ドライバの出力を 7セグメント LED のカソードに接続しています。また 4 個の 7セグメント LED を表示させるためのポートはすべての 7セグメント LED に接続されており、7セグメント LED の表示切り替えは、3 ステートインバータドライバのイネーブル端子により制御しています。

また、7セグメント LED の表示切り替えを行うための信号生成は、2-to-4-line デコーダ (HD74HC139) を使用して、2本のポート出力により制御します。図 1.3 に 7セグメント LED 制御方法について示します。

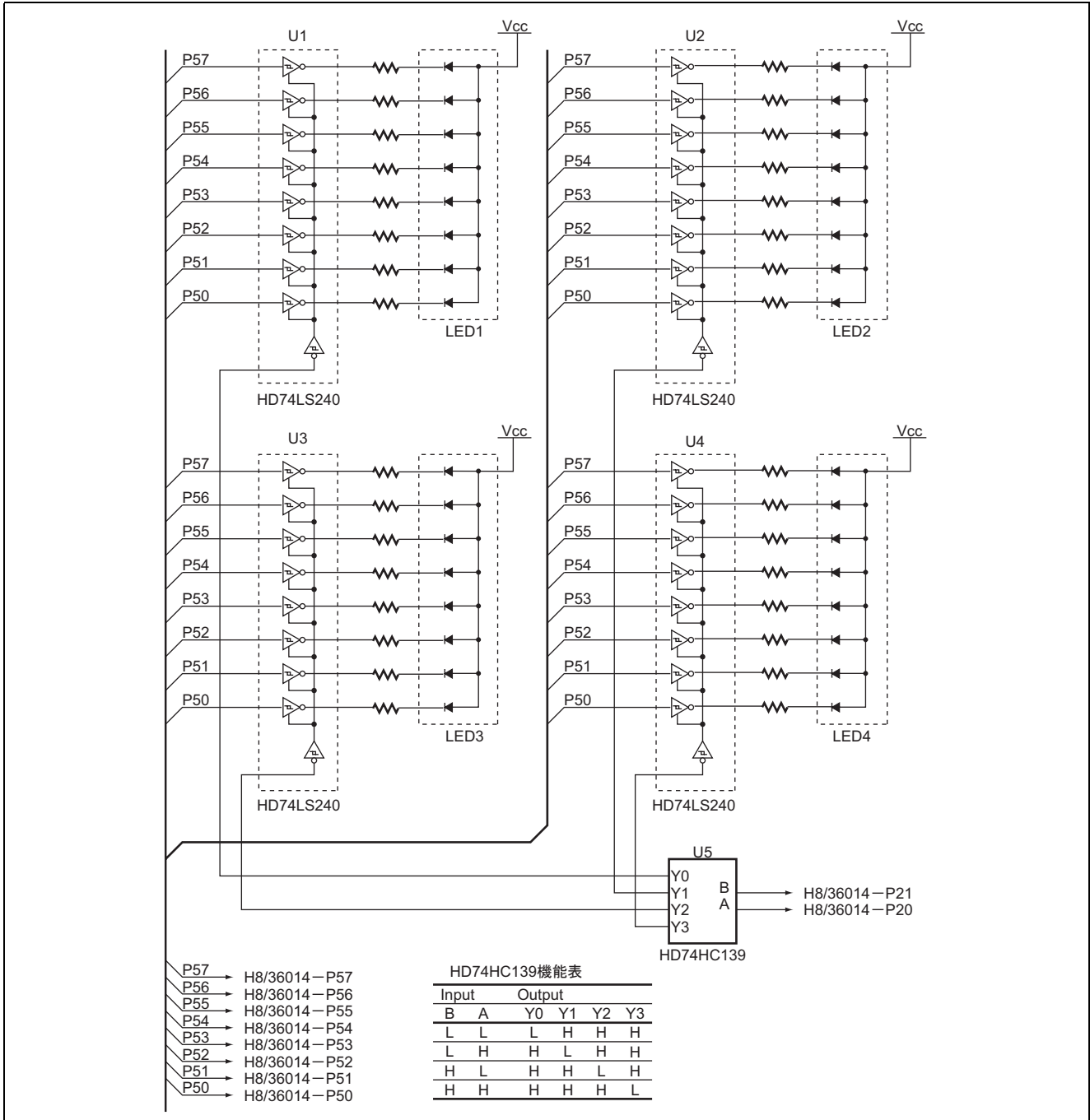


図 1.3 7セグメント LED 制御方法

10. 本タスク例では、A/D 変換結果を 7 セグメント LED に 16 進数で表示 (H'3FF~H'000) させます。図 1.4 に A/D 変換結果の LED 表示方法を示します。

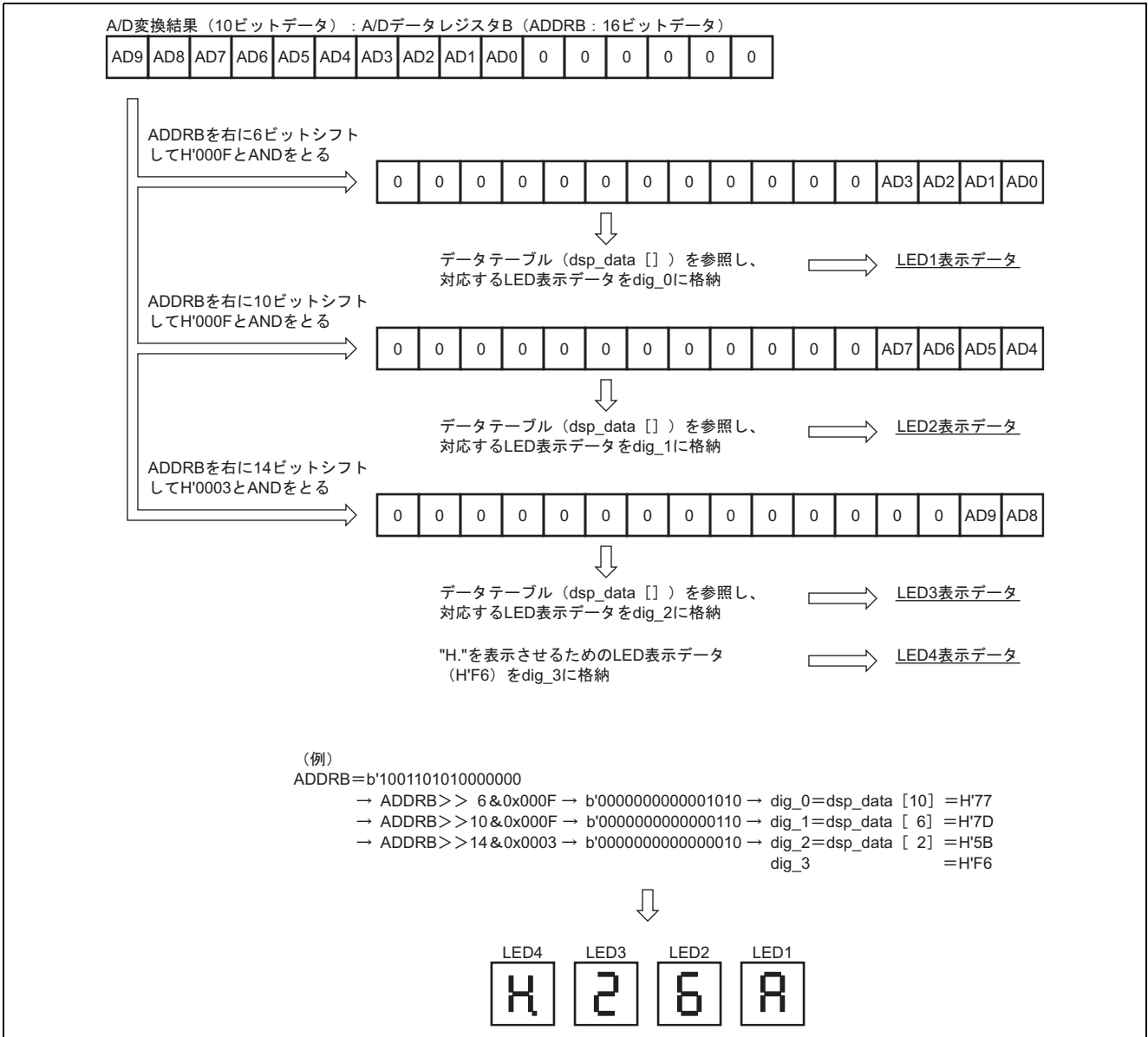


図 1.4 A/D 変換結果の LED 表示方法

2. 使用機能説明

1. 図 2.1 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 2.1 に機能割り付けを示します。

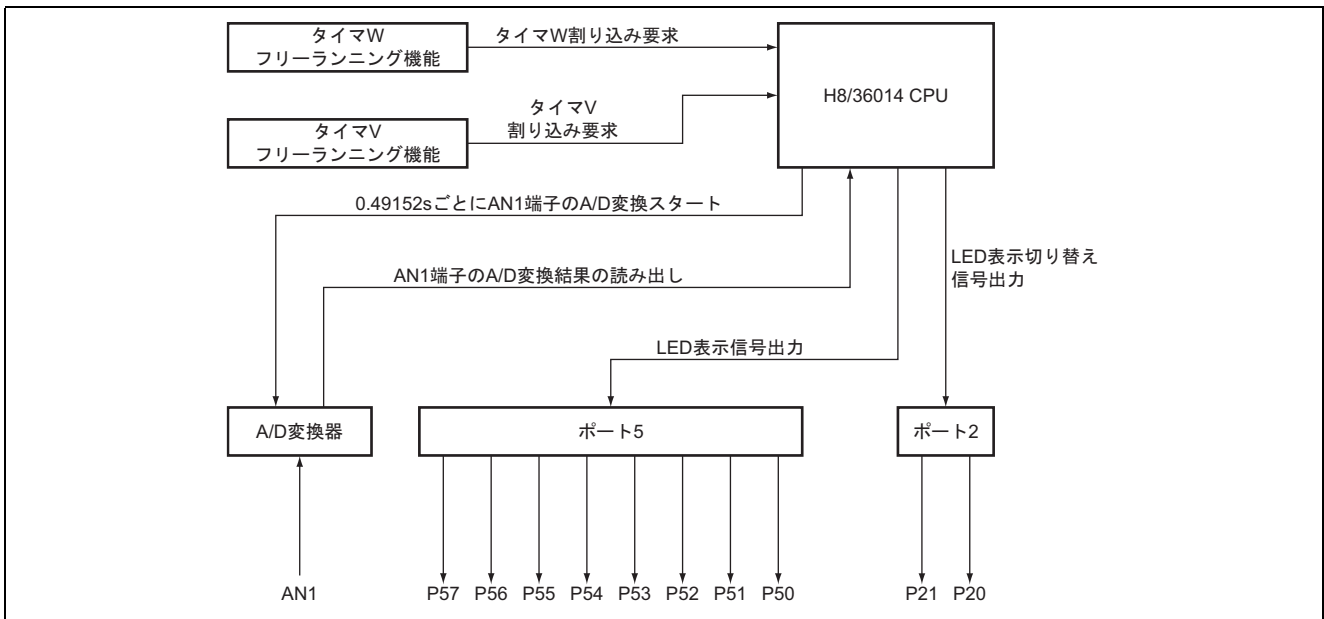


図 2.1 使用機能ブロック図

表 2.1 機能割り付け

使用機能	機能割り付け
タイマ W	タイマ W フリーランニング機能を使用してアナログ入力端子 1 (AN1) の A/D 変換を行います。タイマ W オーバフロー周期 32.768ms により A/D 変換周期を計測します。
タイマ V	タイマ V フリーランニング機能を使用して 7 セグメント LED の表示切り替え制御を行います。タイマ V オーバフロー周期 2.048ms ごとに 4 個の 7 セグメント LED を順番に点灯させることによるダイナミック点灯を行います。
A/D 変換器	A/D 変換器のアナログ入力端子 1 (AN1) に接続された焦電型赤外線センサの赤外線検出量に応じた出力電圧値を A/D 変換します。
ポート 2	ポート 2 の P20、P21 出力端子により、4 個の 7 セグメント LED の表示切り替えを行います。P20、P21 出力端子は 2-to-4-line デコーダの入出端子に接続されています。
ポート 5	ポート 5 の P50~P57 出力端子により、7 セグメント LED の表示を行います。AN1 端子の A/D 変換結果である 10 ビットデータを 3 桁の 16 進数表示データに変換して LED に出力します。

2. 使用する 7セグメント LED の接続図を図 2.2 に示します。図 2.2 に示すようにポート 5 から "High" を出力することにより対応する LED のセグメントが点灯します。また、ポート 5 出力と LED 表示データの関係を表 2.2 に示します。

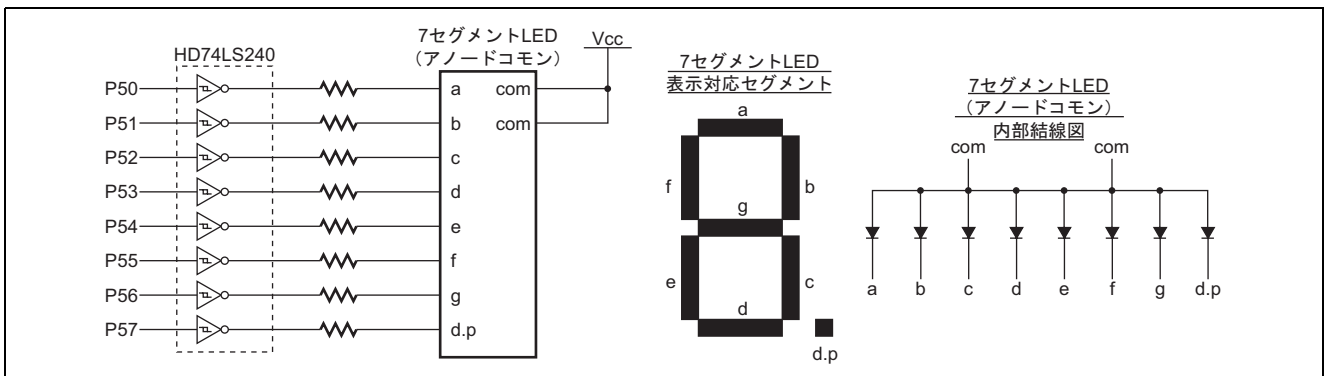


図 2.2 7セグメント LED 接続図および内部結線図



表 2.2 ポート5出力と7セグメントLED表示データの関係

LED表示	ポート5出力データ								LED表示	ポート5出力データ							
	P57	P56	P55	P54	P53	P52	P51	P50		P57	P56	P55	P54	P53	P52	P51	P50
	0	0	1	1	1	1	1	1		0	1	1	1	0	1	1	1
	0	0	0	0	0	1	1	0		0	1	1	1	1	1	0	0
	0	1	0	1	1	0	1	1		0	0	1	1	1	0	0	1
	0	1	0	0	1	1	1	1		0	1	0	1	1	1	1	0
	0	1	1	0	0	1	1	0		0	1	1	1	1	0	0	1
	0	1	1	0	1	1	0	1		0	1	1	1	0	0	0	1
	0	1	1	1	1	1	0	1		0	1	0	0	0	0	0	0
	0	0	1	0	0	1	1	1		1	1	1	1	0	1	1	0
	0	1	1	1	1	1	1	1									
	0	1	1	0	1	1	1	1									

3. 動作原理

1. 図 3.1 にタイマ W 使用した、AN1 端子の A/D 変換を行う際の動作原理を示します。図 3.1 に示すように、本タスク例では、A/D 変換割り込みを使用せずに tmrw ルーチンの中で、タイマ W オーバフローフラグより A/D 変換周期 (0.49152s) の計測、A/D 変換終了判定を行っています。

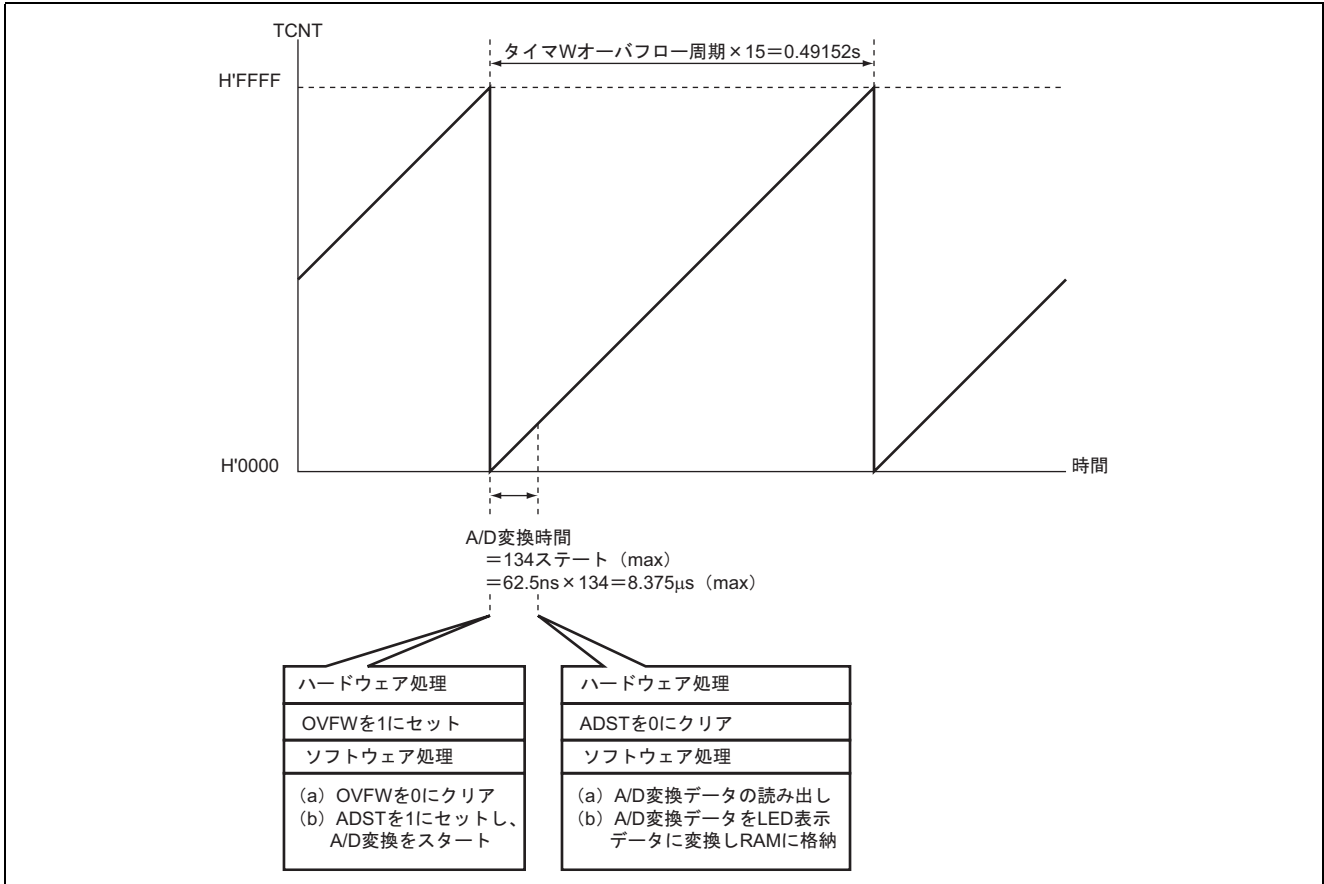


図 3.1 タイマ W を使用した AN1 端子の A/D 変換の動作原理

2. 7セグメントLEDの表示制御の動作原理について説明します。図3.2はLED4~LED1に"3210"を表示する場合の動作原理について説明しています。図3.2に示すようにタイマVオーバーフロー周期ごとにLED1~LED4を順番に表示させることにより7セグメントLEDのダイナミック表示を行っています。

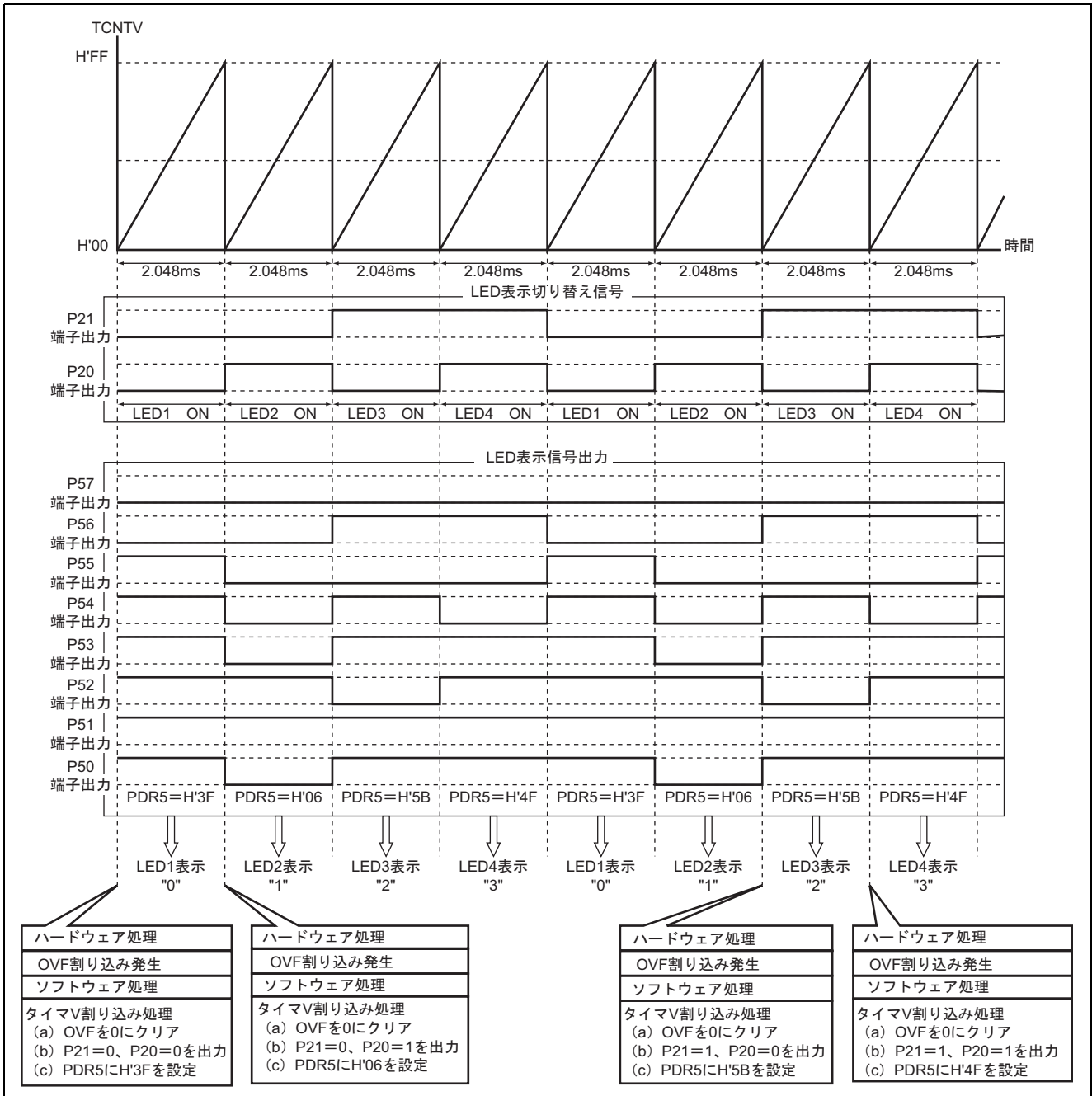


図 3.2 7セグメントLED表示制御の動作原理

## 4. ソフトウェア説明

### 4.1 モジュール説明

表 4.1 に本タスク例におけるモジュール説明を示します。

表 4.1 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、割り込みの許可
タイマ W 割り込み 処理ルーチン	tmrw	割り込みフラグのクリア、A/D 変換データを LED 表示データに変換し RAM に格納する
タイマ V 割り込み 処理ルーチン	tmrv	割り込みフラグのクリア、LED 表示データの出力と LED 表示切り替えの 制御

### 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 4.2 に示します。

表 4.2 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
TCRV0	タイマコントロールレジスタ V0 : TCNTV の入力クロックの選択、TCNTV のクリア条件指定、 各割り込み要求を制御	H'FFA0	H'03 (初期設定時)
CMIEB	コンペアマッチインタラプトイネーブル B : 0 のとき TCSR の CMFB による割り込み要求を禁止	ビット 7	0
CMIEA	コンペアマッチインタラプトイネーブル A : 0 のとき TCSR の CMFA による割り込み要求を禁止	ビット 6	0
OVIE	タイマオーパフローインタラプトイネーブル : 0 のとき TCSR の OVF による割り込み要求を禁止 : 1 のとき TCSR の OVF による割り込み要求を許可	ビット 5	0/1
CCLR1	カウンタクリア 1~0	ビット 4	0
CCLR0	: TCNTV のクリア条件を指定 CCLR1=0、CCLR0=0 設定時、 : TCNTV のクリア禁止	ビット 3	0
CKS2	クロックセレクト 2~0	ビット 2	0
CKS1	: TCRV1 の ICKS0 との組合せにより、TCNTV に入力する クロックとカウント条件を選択	ビット 1	1
CKS0	CKS2=0、CKS1=1、CKS0=1、ICKS0=1 設定時、 : TCNTV は内部クロック $\phi/128$ の立ち下がりエッジで カウント	ビット 0	1

レジスタ名	機能説明	アドレス	設定値
TCSR V	タイマコントロール/ステータスレジスタ V : ステータスフラグの表示、およびコンペアマッチによる出力を制御	H'FFA1	H'10
	CMFB コンペアマッチフラグ B : TCNTV と TCORB の値が一致したとき 1 にセット	ビット 7	0
	CMFA コンペアマッチフラグ A : TCNTV と TCORA の値が一致したとき 1 にセット	ビット 6	0
	OVF タイマオーバフローフラグ : TCNTV の値がオーバフローしたときに 1 にセット : OVF=1 の状態で OVF をリードした後、OVF に 0 をライトしたとき 0 にクリア	ビット 5	0
	OS3 アウトプットセレクト 3~2 : コンペアマッチ B による TMOV 端子の出力レベルを設定	ビット 3	0
	OS2 OS3=0、OS2=0 設定時、 : 変化なし	ビット 2	0
	OS1 アウトプットセレクト 1~0 : コンペアマッチ A による TMOV 端子の出力レベルを設定	ビット 1	0
	OS0 OS1=0、OS0=0 設定時、 : 変化なし	ビット 0	0
TCRV1	タイマコントロールレジスタ V1 : TRGV 端子のエッジセレクト、TRGV 入カインエーブル、TCNTV の入カクロックの選択	H'FFA5	H'E3
	TVEG1 TRGV 入力エッジセレクト 1~0 : TRGV 端子の入カエッジの選択	ビット 4	0
	TVEG0 TREG1=0、TREG0=0 設定時、 : TRGV 端子からのトリガ入力を禁止	ビット 3	0
	TRGE TRGV 入カインエーブル : TVEG1、TVEG0 で選択されたエッジ入力による TCNTV カウントアップの許可/禁止 TREG=0 設定時、 : TRGV 端子入力による TCNTV カウントアップの開始とコンペアマッチによる TCNTV クリア時の TCNTV カウントアップの停止を禁止	ビット 2	0
	ICKS0 インターナルクロックセレクト 0 : TCRV0 の CKS2~CKS0 との組み合わせにより、TCNTV に入力するクロックとカウント条件を選択 CKS2=0、CKS1=1、CKS0=1、ICKS0=1 設定時、 : TCNTV は内部クロック $\phi/128$ の立ち下がリエッジでカウント	ビット 0	1
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'80
	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
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'30
	CKS2 クロックセレクト	ビット 6	0
	CKS1 : CKS2=0、CKS1=1、CKS0=1 のとき、TCNT 入力	ビット 5	1
	CKS0 クロックをシステムクロックの 8 分周のクロックに設定	ビット 4	1

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)
OVIE	タイマオーバーフロー割り込みイネーブル : OVIE=0 のとき、OVF による割り込み要求を禁止 : OVIE=1 のとき、OVF による割り込み要求を許可	ビット 7	0/1
TSRW	割り込み要求ステータスを表示	H'FF83	H'00
OVF	タイマオーバーフロー : OVF=0 のとき、TCNT がオーバーフローしていないことを示す : OVF=1 のとき、TCNT がオーバーフローしたことを示す	ビット 7	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
ADCSR	A/D コントロール/ステータスレジスタ : A/D 変換器の制御ビットと変換ステータスビットにより構成	H'FFB8	H'01 (初期設定時)
ADF	A/D エンドフラグ : 単一モードで A/D 変換が終了したとき 1 にセット : ADF=1 の状態をリード後、0 をライトしたとき 0 にクリア	ビット 7	0
ADIE	A/D インタラプトイネーブル : 0 のとき ADF による A/D 変換終了割り込み要求を禁止	ビット 6	0
ADST	A/D スタート : 1 にセットすることにより A/D 変換を開始 : 単一モードで A/D 変換が終了すると自動的に 0 にクリア	ビット 5	0/1
SCAN	スキャンモード : 0 のとき A/D 変換モードは単一モード	ビット 4	0
CKS	クロックセレクト : 0 のとき A/D 変換時間=134 ステート (max)	ビット 3	0
CH2	チャンネルセレクト	ビット 2	0
CH1	: アナログ入力チャンネルを選択	ビット 1	0
CH0	CH2=0、CH1=0、CH0=1 設定時 (SCAN=0 のとき)、 : AN1	ビット 0	1
ADCR	A/D コントロールレジスタ : 外部トリガによる A/D 変換開始をイネーブルに設定	H'FFB9	H'7E
TRGE	トリガイネーブル : 0 のとき、外部トリガ端子 ( $\overline{\text{ADTRG}}$ ) のエッジ入力による A/D 変換の開始を禁止	ビット 7	0
ADDRB	A/D データレジスタ B : AN1 の A/D 変換結果を格納	H'FFB2	—
PCR2	ポートコントロールレジスタ 2 : ポート 2 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 PCR2=H'03 のとき、 : P22 端子は汎用入力端子として機能、P21、P20 端子は汎用出力端子として機能	H'FFE5	H'03
PDR2	ポートデータレジスタ 2 : ポート 2 の汎用入出力ポートデータレジスタ	H'FFD5	H'F8

レジスタ名	機能説明	アドレス	設定値
PMR5	ポートモードレジスタ 5 : ポート 5 の端子機能を設定	H'FFE1	H'00
POF7	P57 端子機能切り替え : 0 のとき P57 汎用入出力ポート機能	ビット 7	0
POF6	P56 端子機能切り替え : 0 のとき P56 汎用入出力ポート機能	ビット 6	0
WKP5	P55/WKP5/ADTRG 端子機能切り替え : 0 のとき P55 汎用入出力ポート機能	ビット 5	0
WKP4	P54/WKP4 端子機能切り替え : 0 のとき P54 汎用入出力ポート機能	ビット 4	0
WKP3	P53/WKP3 端子機能切り替え : 0 のとき P53 汎用入出力ポート機能	ビット 3	0
WKP2	P52/WKP2 端子機能切り替え : 0 のとき P52 汎用入出力ポート機能	ビット 2	0
WKP1	P51/WKP1 端子機能切り替え : 0 のとき P51 汎用入出力ポート機能	ビット 1	0
WKP0	P50/WKP0 端子機能切り替え : 0 のとき P50 汎用入出力ポート機能	ビット 0	0
PUCR5	ポートプルアップコントロールレジスタ 5 : 入力ポートに設定されたポート 5 の各端子のプルアップ MOS をビットごとに制御 PUCR5=H'00 のとき、 : P57~P50 端子のプルアップ MOS はオフ	H'FFD1	H'00
PDR5	ポートデータレジスタ 5 : ポート 5 の汎用入出力ポートデータレジスタ	H'FFD8	H'00
PCR5	ポートコントロールレジスタ 5 : ポート 5 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 PCR5=H'FF のとき、 : P57~P50 端子は汎用出力端子として機能	H'FFE8	H'FF

#### 4.4 使用 RAM 説明

表 4.3 に本タスク例における使用 RAM 説明を示します。

表 4.3 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
dig_0	LED1 の表示データを格納 (1 バイト)	H'FB80	main,tmrw
dig_1	LED2 の表示データを格納 (1 バイト)	H'FB81	main,tmrw
dig_3	LED3 の表示データを格納 (1 バイト)	H'FB82	main,tmrw
dig_4	LED4 の表示データを格納 (1 バイト)	H'FB83	main,tmrw
cnt	LED1~LED4 の表示切り替えのための 8 ビットカウンタ (1 バイト)	H'FB84	main,tmrw
counter_sub	A/D 取得間隔調整のための 8 ビットカウンタ (1 バイト)	H'FB85	main,tmrw

#### 4.5 データテーブル説明

本タスク例では7セグメントLEDの表示データを1次列のデータテーブルとしてROMに格納しています。表4.4に7セグメントLED表示データテーブル (dsp\_data[ ]) 説明を示します。

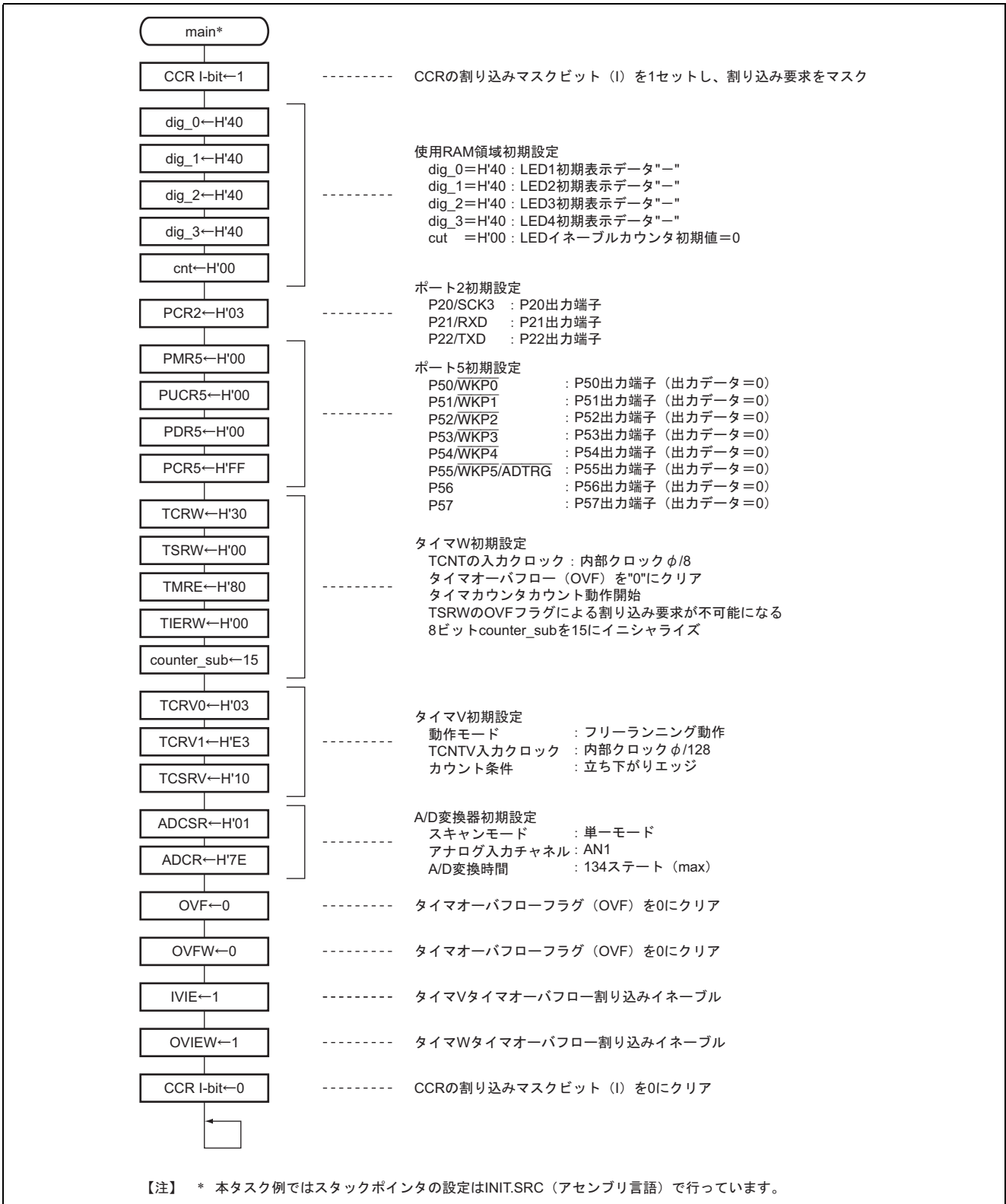
表 4.4 7セグメントLED表示データテーブル (dsp\_data[]) 説明

配列名	データ	データ説明	データサイズ	アドレス
dsp_data[0]	H'3F	LEDに"0"を表示させるためのポート5出力データ	1バイト	H'021C
dsp_data[1]	H'06	LEDに"1"を表示させるためのポート5出力データ	1バイト	H'021D
dsp_data[2]	H'5B	LEDに"2"を表示させるためのポート5出力データ	1バイト	H'021E
dsp_data[3]	H'4F	LEDに"3"を表示させるためのポート5出力データ	1バイト	H'021F
dsp_data[4]	H'66	LEDに"4"を表示させるためのポート5出力データ	1バイト	H'0220
dsp_data[5]	H'6D	LEDに"5"を表示させるためのポート5出力データ	1バイト	H'0221
dsp_data[6]	H'7D	LEDに"6"を表示させるためのポート5出力データ	1バイト	H'0222
dsp_data[7]	H'27	LEDに"7"を表示させるためのポート5出力データ	1バイト	H'0223
dsp_data[8]	H'7F	LEDに"8"を表示させるためのポート5出力データ	1バイト	H'0224
dsp_data[9]	H'6F	LEDに"9"を表示させるためのポート5出力データ	1バイト	H'0225
dsp_data[10]	H'77	LEDに"A"を表示させるためのポート5出力データ	1バイト	H'0226
dsp_data[11]	H'7C	LEDに"b"を表示させるためのポート5出力データ	1バイト	H'0227
dsp_data[12]	H'39	LEDに"C"を表示させるためのポート5出力データ	1バイト	H'0228
dsp_data[13]	H'5E	LEDに"d"を表示させるためのポート5出力データ	1バイト	H'0229
dsp_data[14]	H'79	LEDに"E"を表示させるためのポート5出力データ	1バイト	H'022A
dsp_data[15]	H'71	LEDに"F"を表示させるためのポート5出力データ	1バイト	H'022B

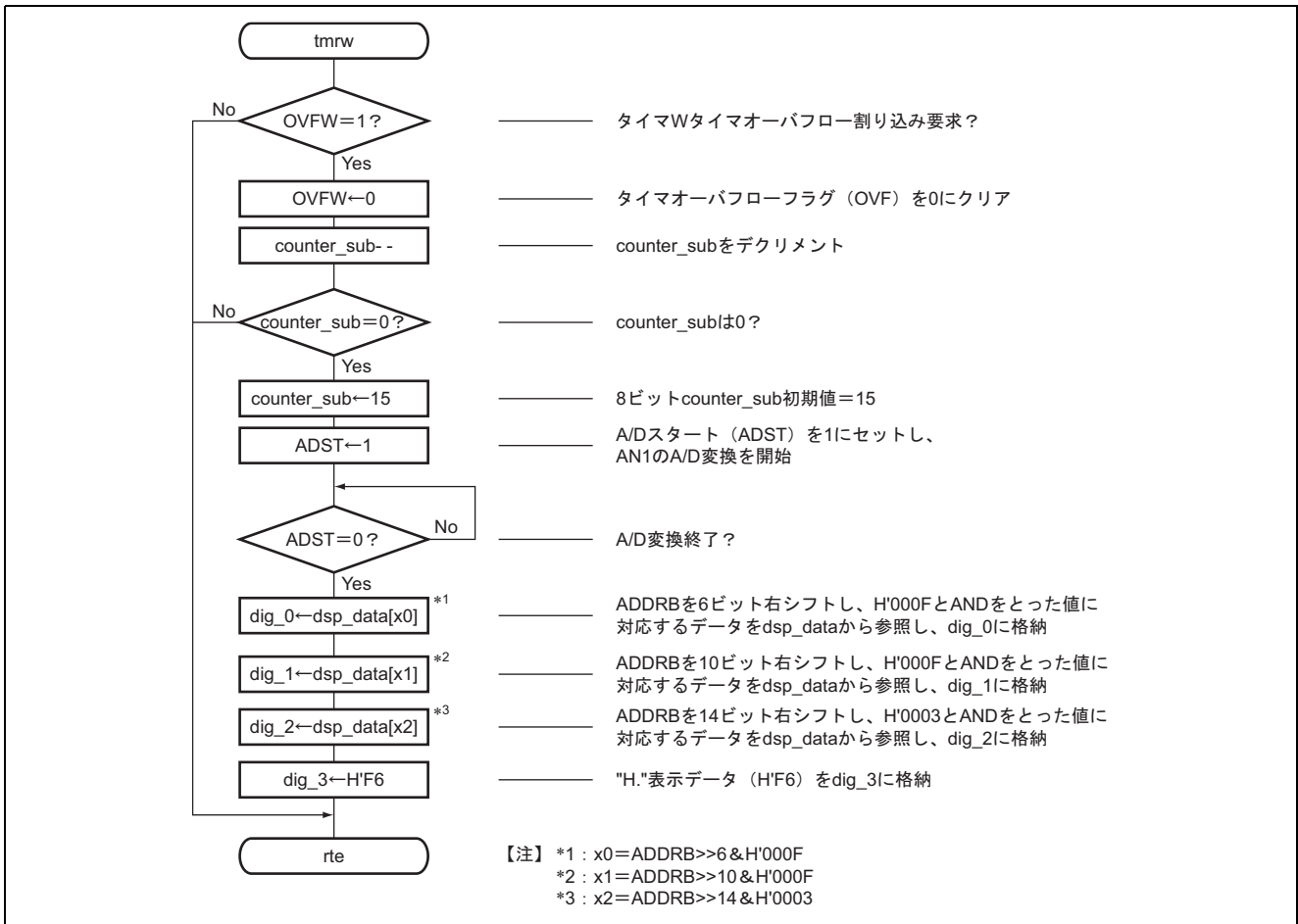


### 5. フローチャート

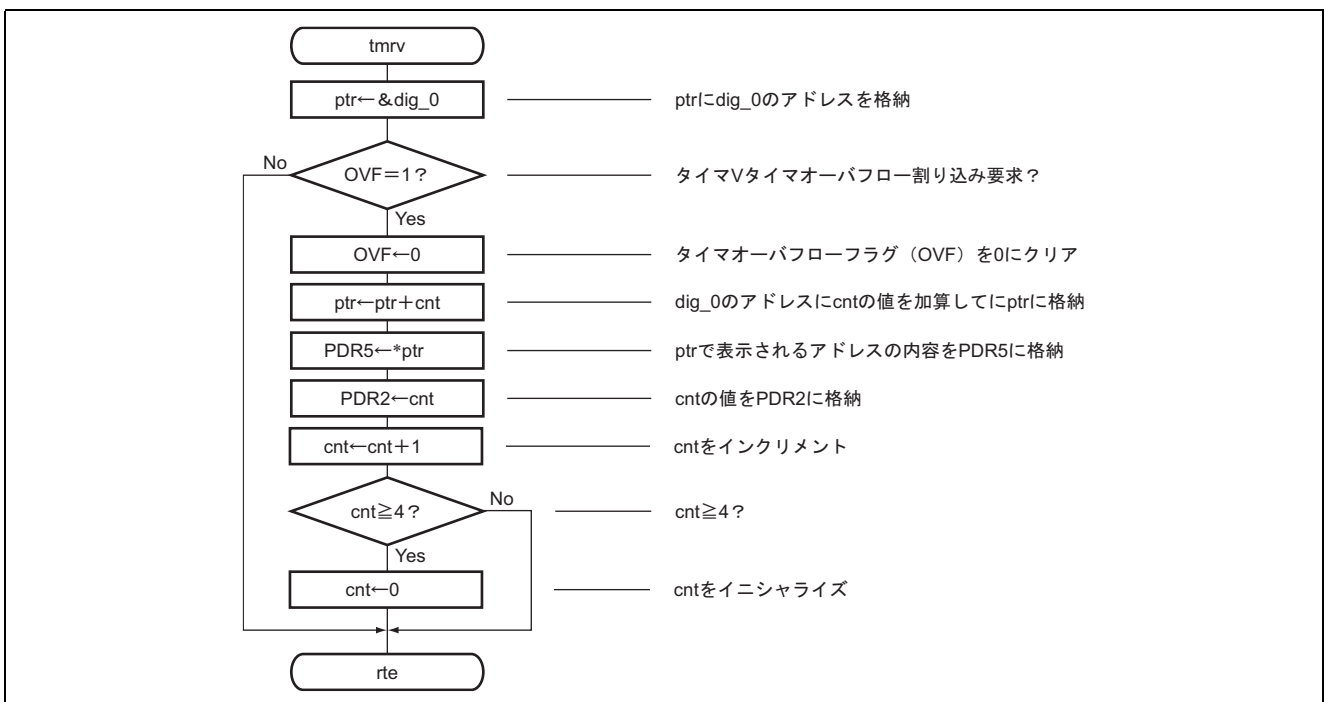
#### 1. メインルーチン (main)



### 2. タイマ W 割り込み処理ルーチン (tmrw)



### 3. タイマ V 割り込み処理ルーチン (tmrv)



## 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 */
/* モーションセンサ接続例 */

#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 PDR2      *(volatile unsigned char *)0xFFD5 /* Port data register 2 */
#define PCR2      *(volatile unsigned char *)0xFFE5 /* Port control register 2 */

#define PMR5      *(volatile unsigned char *)0xFFE1 /* Port mode register 5 */
#define PUCR5     *(volatile unsigned char *)0xFFD1 /* Port pull-up control register 5 */
#define PDR5      *(volatile unsigned char *)0xFFD8 /* Port data register 5 */
#define PCR5      *(volatile unsigned char *)0xFFE8 /* Port control register 5 */
#define TMRW      *(volatile unsigned char *)0xFFE0 /* Timer mode register W */
#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 OVIEW     TIERW_BIT.b7 /* Timer Overflow Interrupt Enable W */
#define TSRW      *(volatile unsigned char *)0xFF83 /* Timer status register W */
#define TSRW_BIT  (*(struct BIT *)0xFF83) /* Timer Status Register W */
#define OVFW      TSRW_BIT.b7 /* Timer Over flow W */

#define TCRV0     *(volatile unsigned char *)0xFFA0 /* Timer control register V0 */
#define TCRV0_BIT (*(struct BIT *)0xFFA0)
#define OVIE      TCRV0_BIT.b5 /* Timer overflow interrupt enable */
#define TCSRW     *(volatile unsigned char *)0xFFA1 /* Timer control/status register V */
#define TCSRW_BIT (*(struct BIT *)0xFFA1)

```

```

#define OVF          TCSR_V_BIT.b5          /* Timer overflow flag          */
#define TCRV1        *(volatile unsigned char *)0xFFA5 /* Timer control register V1    */

#define ADCSR        *(volatile unsigned char *)0xFFB8 /* A/D control/status register  */
#define ADCSR_BIT    (*(struct BIT *)0xFFB8)
#define ADST         ADCSR_BIT.b5          /* A/D start                    */
#define ADCR         *(volatile unsigned char *)0xFFB9 /* A/D control register         */
#define ADDRA        *(volatile unsigned int *)0xFFB0 /* A/D data register A          */
#define ADDRb        *(volatile unsigned int *)0xFFB2 /* A/D data register A          */
#define ADDRc        *(volatile unsigned int *)0xFFB4 /* A/D data register A          */
#define ADDRd        *(volatile unsigned int *)0xFFB6 /* A/D data register A          */

#pragma interrupt (tmrw)
#pragma interrupt (tmrv)

/* 関数定義 */
extern void INIT(void); /* Stack pointer set */
void main(void); /* main routine */
void tmrw(void); /* Timer W interrupt routine */
void tmrv(void); /* Timer V interrupt routine */

/* Data table */
const unsigned char dsp_data[16] =
{
    0x3f, /* LED display data = "0" */
    0x06, /* LED display data = "1" */
    0x5b, /* LED display data = "2" */
    0x4f, /* LED display data = "3" */
    0x66, /* LED display data = "4" */
    0x6d, /* LED display data = "5" */
    0x7d, /* LED display data = "6" */
    0x27, /* LED display data = "7" */
    0x7f, /* LED display data = "8" */
    0x6f, /* LED display data = "9" */
    0x77, /* LED display data = "A" */
    0x7c, /* LED display data = "B" */
    0x39, /* LED display data = "C" */
    0x5e, /* LED display data = "D" */
    0x79, /* LED display data = "E" */
    0x71, /* LED display data = "F" */
};

/* RAM define */
unsigned char dig_0; /* Dig-0 LED display data store */
unsigned char dig_1; /* Dig-1 LED display data store */
unsigned char dig_2; /* Dig-2 LED display data store */
unsigned char dig_3; /* Dig-3 LED display data store */
unsigned char cnt; /* LED enable counter */
unsigned char counter_sub;

/* 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 V3                          /* Vector section set */
void (*const VEC_TBL3[]) (void) = {
    tmrv                                     /* H'002c Timer V interrupt vector */
};
#pragma section                             /* P */

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

    dig_0 = 0x40;                            /* Used RAM area initialize */
    dig_1 = 0x40;                            /* Used RAM area initialize */
    dig_2 = 0x40;                            /* Used RAM area initialize */
    dig_3 = 0x40;                            /* Used RAM area initialize */
    cnt = 0x00;                              /* Used RAM area initialize */

    PCR2 = 0x03;                             /* Port 2 initialize */

    PMR5 = 0x00;                             /* Port 5 initialize */
    PUCR5 = 0x00;
    PDR5 = 0x00;
    PCR5 = 0xff;

    TCRW = 0x30;                             /* Timer W initialize */
    TSRW = 0x00;                             /* Clock Select */
    TMRW = 0x80;                             /* Clear OVF */
    TIERW = 0x00;                            /* Timer Counter Count Start */
    counter_sub = 15;                        /* OVF Interrupt Disable */
    /* Initialize 8bit Counter_sub */

    TCRV0 = 0x03;                            /* Timer V initialize */
    TCRV1 = 0xe3;                            /* Internal clock select */
    TCSRv = 0x10;                            /* Clear OVF to 0 */

    ADCSR = 0x01;                            /* A/D converter initialize */
    ADCR = 0x7e;

    OVF = 0;                                /* Clear OVF to 0 */
    OVFW = 0;                                /* Clear OVF to 0 */
    OVIE = 1;                                /* Timer V OVF interrupt enable */
    OVIEW = 1;                               /* Timer W OVF interrupt enable */

    set_imask_ccr(0);                       /* CCR I-bit = 0 */

    while(1);
}

```

```

/*****
/* Timer W Interrupt */
/*****
void tmrw(void)
{
    if ( OVFW == 1 ) {
        OVFW = 0; /* Clear OVF */
        counter_sub--; /* Decrement 8bit Counter */
        if ( counter_sub == 0x00 ){ /* 8bit Counter != H'00 */
            counter_sub = 15; /* Initialize 8bit Counter_sub */
            ADST = 1; /* A/D converter start */
            while(ADST == 1); /* A/D converter end ? */
            dig_0 = dsp_data[ADDRB >> 6 & 0x000f]; /* Dig-0 LED display data set */
            dig_1 = dsp_data[ADDRB >> 10 & 0x000f]; /* Dig-1 LED display data set */
            dig_2 = dsp_data[ADDRB >> 14 & 0x0003]; /* Dig-2 LED display data set */
            dig_3 = 0xf6; /* Dig-3 LED display data set */
        }
    }
}

/*****
/* Timer V Interrupt */
/*****
void tmrv(void)
{
    unsigned char *ptr; /* Pointer set */

    ptr = &dig_0; /* LED display data store address set */

    while(OVF == 1){ /* OVF = 1 ? */
        OVF = 0; /* Clear OVF to 0 */
        ptr += cnt; /* LED display data read */
        PDR5 = *ptr; /* LED display data output */
        PDR2 = cnt; /* LED enable data output */
        cnt++; /* "cnt" increment */
        if (cnt >= 4){ /* 4 times end ? */
            cnt = 0; /* "cnt" initialize */
        }
    }
}

```

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2003.09.24	—	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。