

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

角速度センサジャイロスター接続例

要旨

角速度センサの出力を DC カットし、直流増幅回路を用いて必要な電圧に増幅します。この増幅後の信号をアナログ入力端子 AN0 に接続し、10 進数の角速度値 (単位: deg./sec.) を 7 セグメント LED に表示します。

動作確認デバイス

H8/300L Super Low Power シリーズ H8/38024CPU

目次

1. 仕様	2
2. 使用機能説明	7
3. 動作原理	9
4. ソフトウェア説明	11
5. フローチャート	15
6. プログラムリスト	20

1. 仕様

- 図 1 に圧電型振動ジャイロセンサ接続例のハードウェア構成を示します。図 1 に示すように、センサをアナログ入力端子 0 (AN0 端子) に接続します。
- AN0 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。
- 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 10 進数の角速度で表示します。A/D 変換は、0.5s 毎に行います。

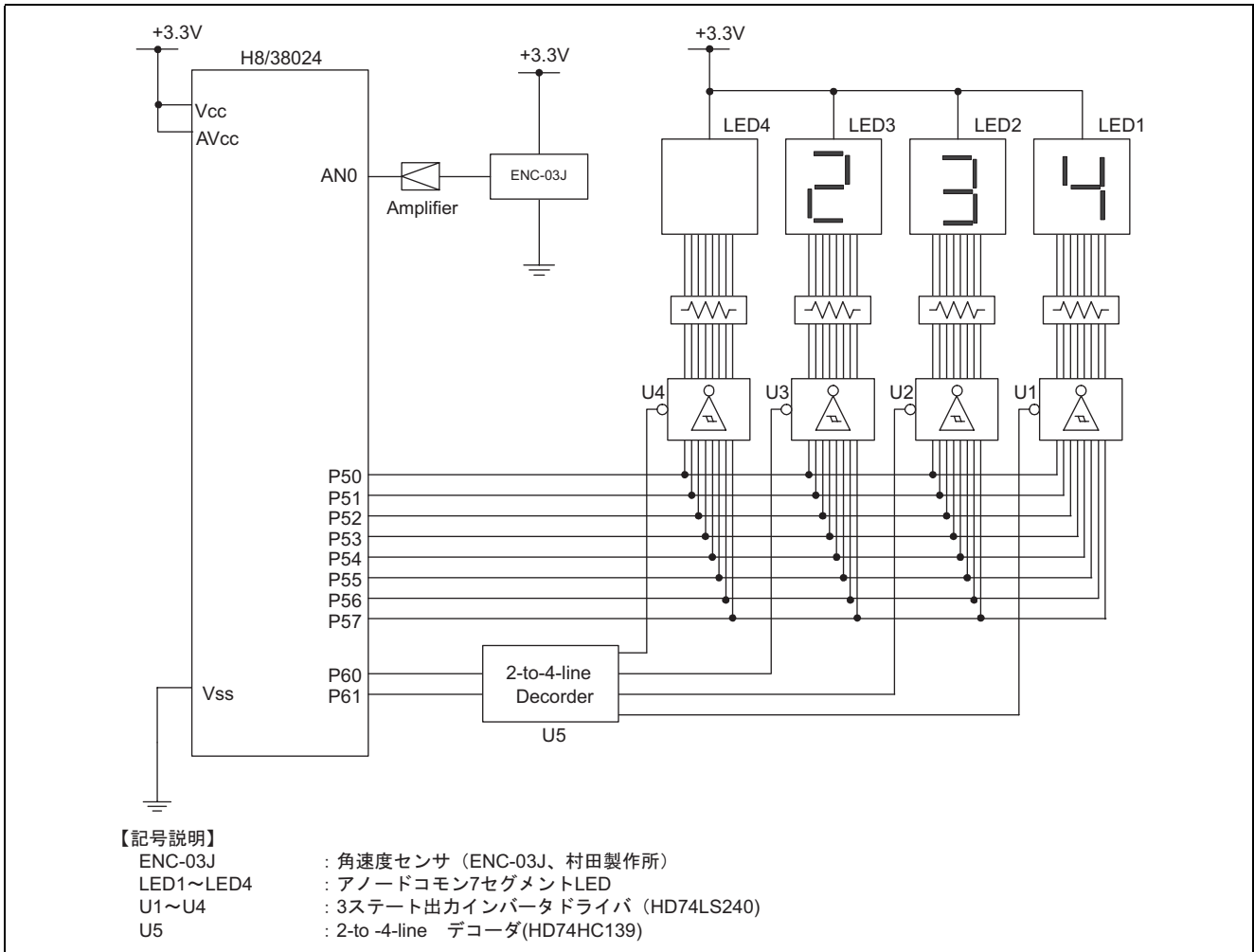


図 1 ハードウェア構成

4. 本タスク例における H8/38024 の動作電圧(V_{CC})およびアナログ電源電圧(AV_{CC})は 3.3V、OSC クロック周波数は 10MHz、ウオッチクロック周波数は 32.768kHz です。
5. 本タスク例で使用しているジャイロセンサは、株式会社村田製作所製の圧電型振動ジャイロセンサ(型名: ENC-03J)です。以下に仕様を示します。
 - A. 表 1 に圧電型振動ジャイロセンサの仕様を示します

表 1 ENC-03J 仕様 (参考)

項目	仕様
動作電圧	+3.0VDC
消費電流	4.5mA 以下 (電源電圧+3.0VDC にて)
検出範囲	$\pm 300^\circ / s$
感度	0.67 mV / ($^\circ / s$) $\pm 20\%$ (初期偏差)
静止時出力	基準出力 $\pm 0.6V$ DC (使用温度範囲にて)
基準出力	1.35 \pm 0.1V DC (使用温度範囲にて)
リニアリティ	$\pm 5.0\%$ FS 以内
感度温度変動	-20 ~ +10%以内 (25 を基準として使用温度範囲にて)
応答性	DC ~ 50Hz
使用温度範囲	-5 ~ +75
保存温度範囲	-30 ~ +85
外形寸法	15.5 \times 8.0 \times 4.3mm
重量	1.0g 以下

【注】 数値はすべて代表値です。

【注】 特に指定のない限り、室温($T_a=25^\circ C$)にて規定します。

【注】 基準出力は、4.7 μF のコンデンサで接地されるものとします。

- B. 図 2 にセンサ標準増幅回路を示します。

本回路では、センサの出力成分に含まれる DC 成分と高周波ノイズ成分を除去しています。本回路を参考にされる場合は、検出する動きによって定数を選択する必要があります。

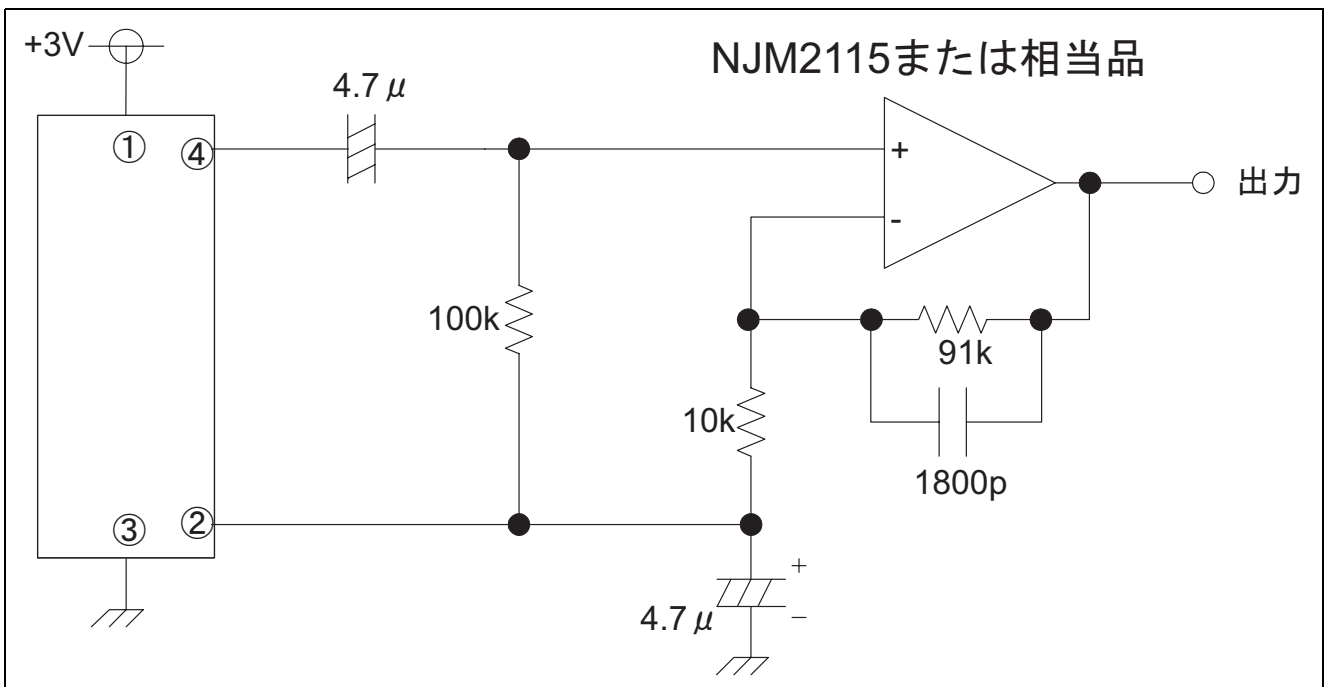


図 2 センサ標準増幅回路 (参照)

6. 圧電型振動ジャイロセンサ ENC-03J は、セラミックバイモルフ素子を振動させることによってコリオリの力を検出する振動型の角速度センサです。センサ応答範囲が DC ~ 50Hz で物体の運動検出に適しており、ビデオカメラの手振れ検出やポインティングデバイス、HMD の動き検出に応用されています。
7. 本タスク例の動作は以下の通りです。
 - A. ENC-03J からのセンサ出力電圧を直流増幅回路で増幅します。増幅後の信号をアナログ入力端子 0 (AN0 端子) に入力して A/D 変換します。
 - B. 静止状態では、表 1 の基準電圧 1.35VDC がアナログ入力端子 0 (AN0 端子) に入力されますが、LED 表示は「・・・0」(単位 : deg/sec) にしてあります。
 - C. 図 1 の表示例「・234」では、角速度が 234 (deg/sec) であることを示します。
 - D. センサを検出回転方向と逆に回転させると LED 表示は「-178」となり、検出回転方向と反対方向に角速度の大きさが 178 (deg/sec) であることを示します。
8. 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 の表示切替は、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 の表示切替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 3 に 7 セグメント LED 制御方法について示します。

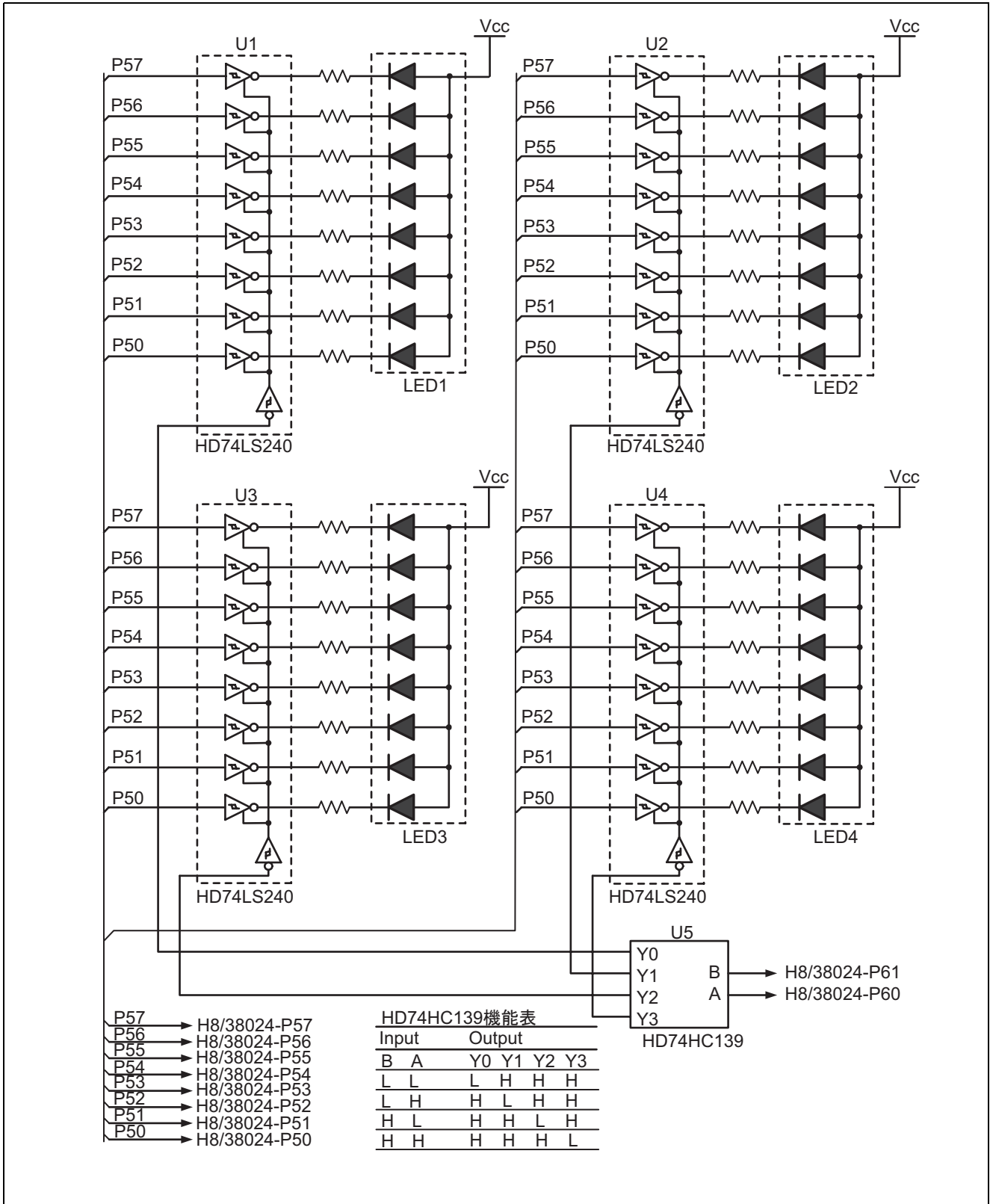


図3 7セグメントLED制御方法

9. 本タスク例では、10進数の角速度値（単位:deg./sec.）を7セグメントLEDに表示させます。図4に角速度値のLED表示方法を示します

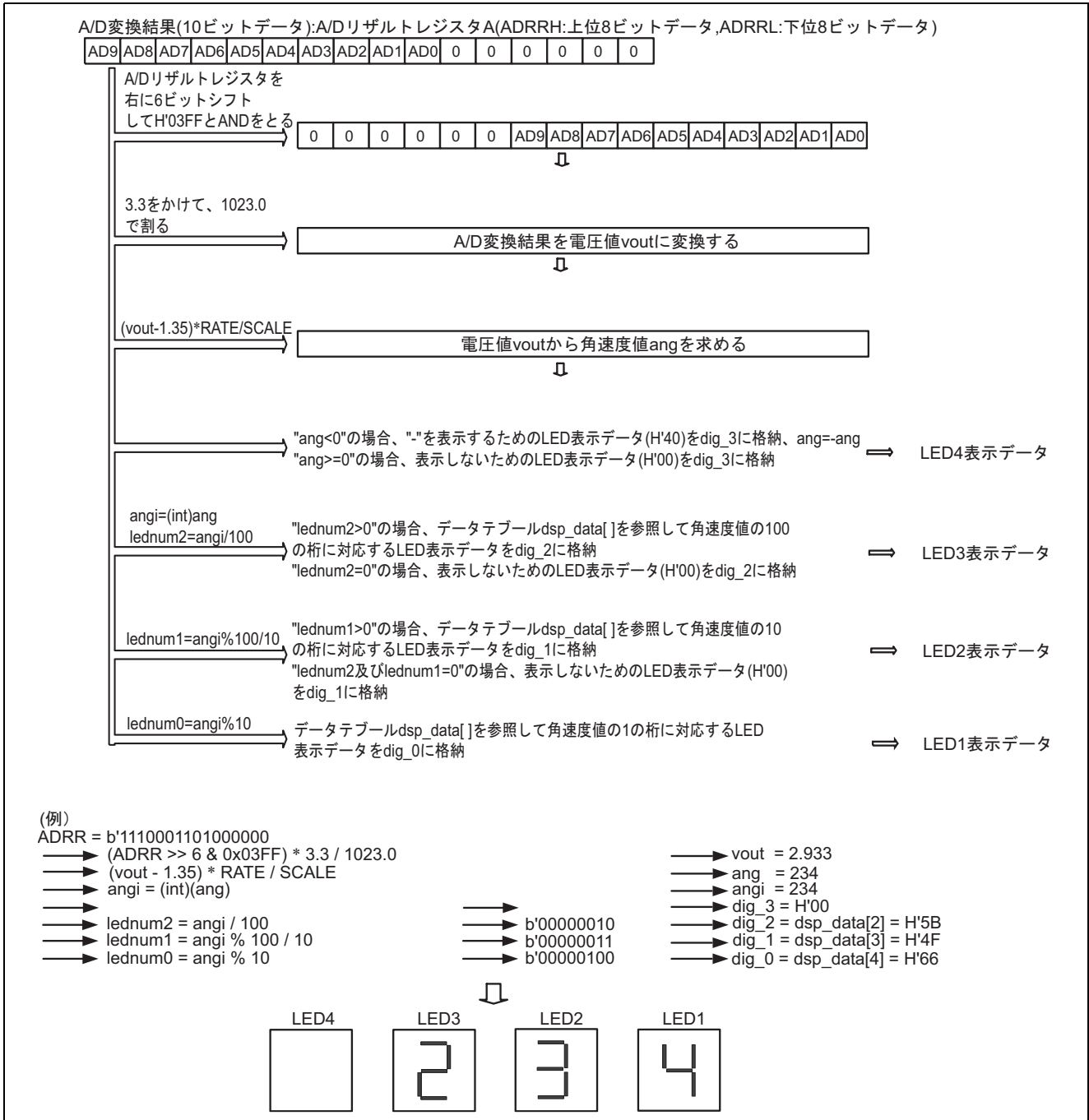


図4 角速度値のLED表示方法

2. 使用機能説明

1. 図 5 に本タスク例における H8/38024 の使用機能のブロック図を、表 2 に機能割付を示します。

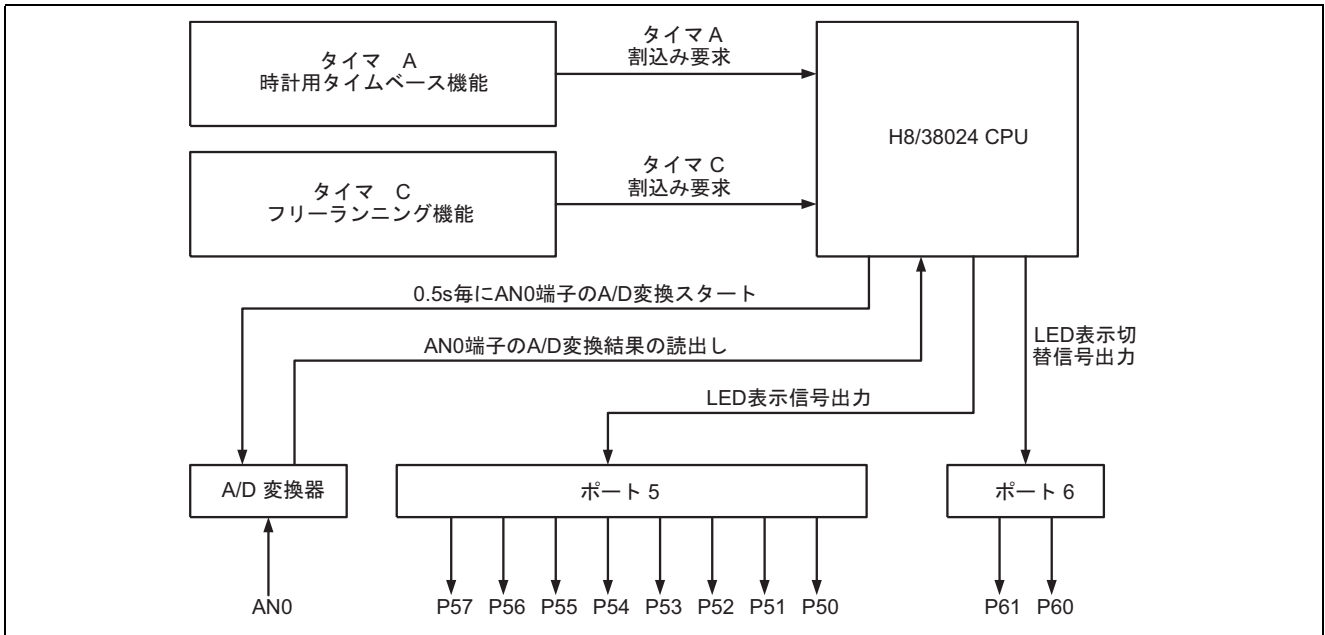


図 5 使用機能ブロック図

表 2 機能割付け

使用機能	機能割付け
タイマ A	タイマ A 時計用タイムベース機能を使用してアナログ入力端子 0 (AN0) の A/D 変換を行う周期である 0.5s を計測します。A/D 変換周期毎にタイマ A 割り込みを使用します。
タイマ C	タイマ C フリーランニング機能を使用して 7 セグメント LED の表示切替制御を行います。タイマ C オーバフロー周期 3.2768ms 毎に 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 0 (AN0) に接続された角速度センサからの出力電圧値を A/D 変換します。
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。AN0 端子の A/D 変換結果である 10 ビットデータを小数の角速度表示データ(単位:deg./sec.)に変換して LED に出力します。
ポート 6	ポート 6 の P60、P61 出力端子により、4 個の 7 セグメント LED の表示切替を行います。P60、P61 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。

2. 使用する 7 セグメント LED の接続図を図 6 に示します。図 6 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 3 に示します。

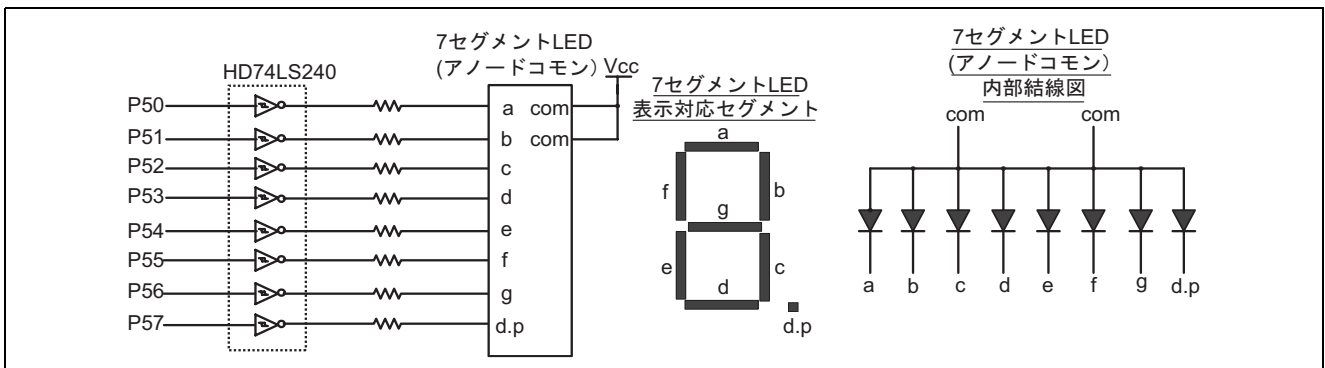


図 6 7 セグメント LED 接続図および内部結線図

表3 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	0	0	0	0	0	0
	0	0	0	0	0	1	1	0									
	0	1	0	1	1	0	1	1									
	0	1	0	0	1	1	1	1									
	0	1	1	0	0	1	1	0									
	0	1	1	0	1	1	0	1									
	0	1	1	1	1	1	0	1									
	0	0	1	0	0	1	1	1									
	0	1	1	1	1	1	1	1									
	0	1	1	0	1	1	1	1									

3. 動作原理

1. 図7にタイマAを使用した、AN0端子のA/D変換を行う際の動作原理を示します。図7に示すように、本タスク例では、A/D変換割込みを使用せずに tmra 割込み処理ルーチンの中で、A/D変換終了判定を行っています。

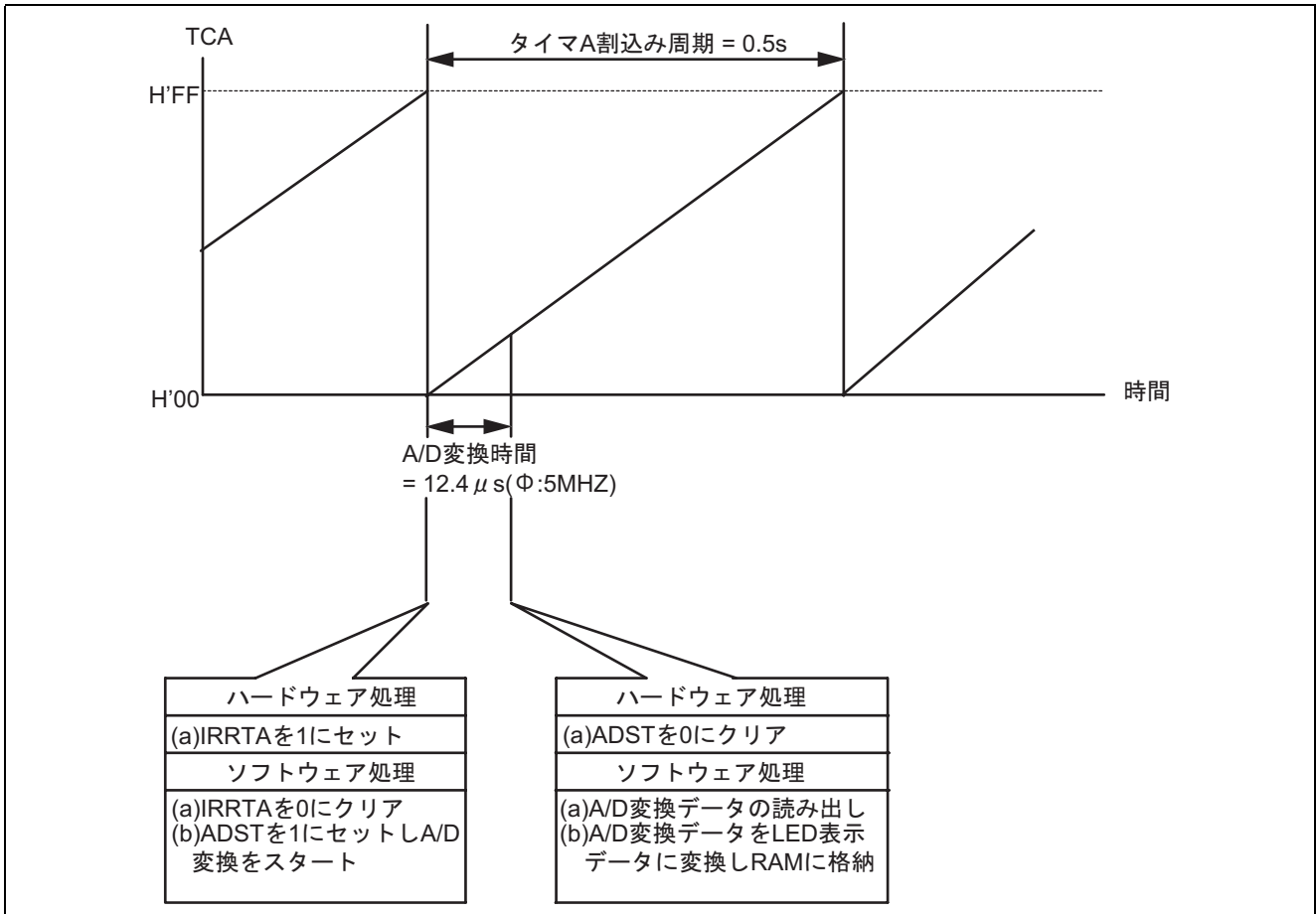


図7 タイマAを使用したAN0端子のA/D変換の動作原理

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8はLED4~LED1に”234”を表示する場合の動作原理について説明しています。図8に示すようにタイマCオーバフロー周期ごとに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。

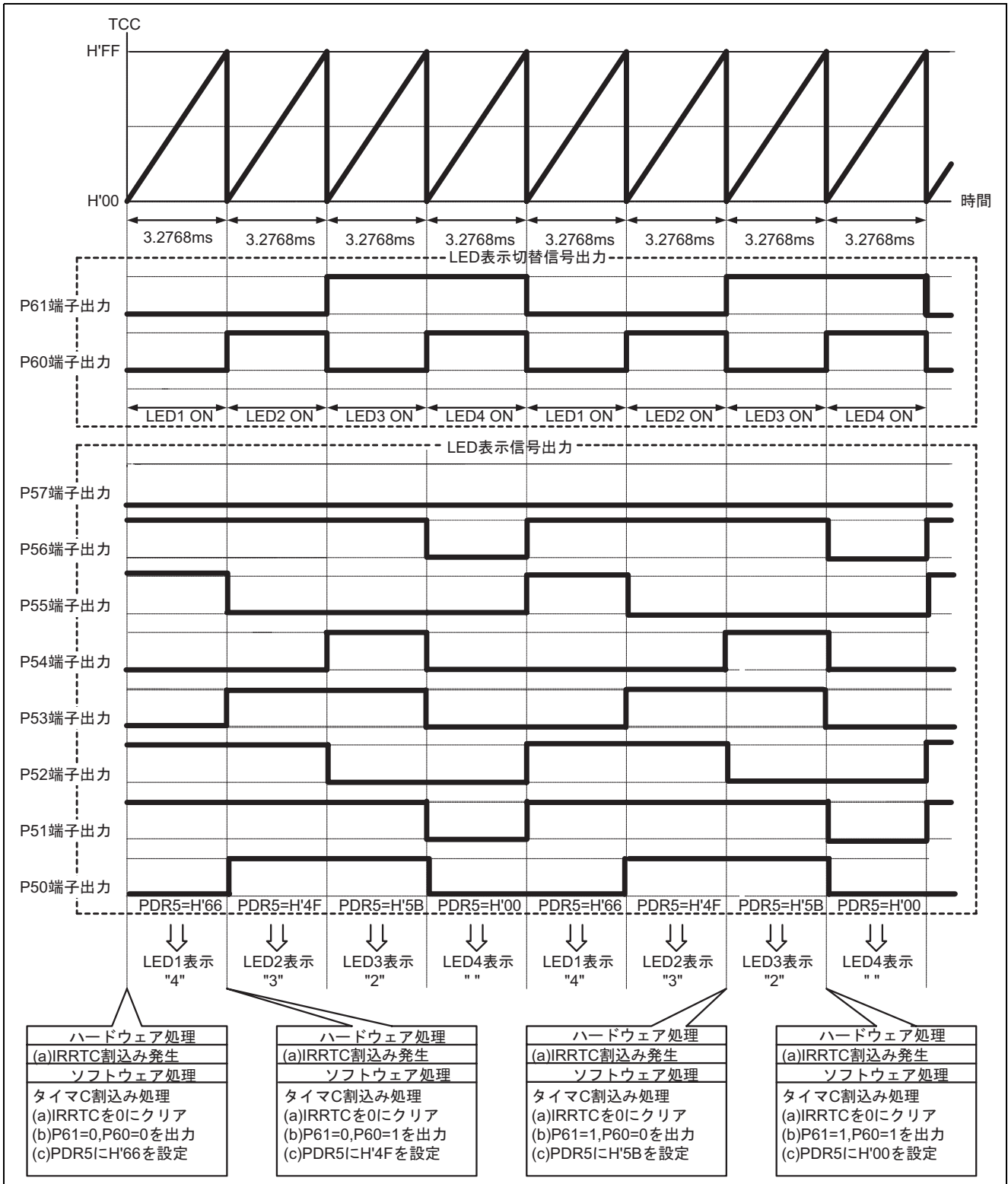


図8 7セグメントLED表示制御の動作原理

4. ソフトウェア説明

1. モジュール説明

表 4 に本タスク例におけるモジュール説明を示します。

表 4 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 A 割り込み処理ルーチン	tmra	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 C 割り込み処理ルーチン	tmrc	割り込みフラグのクリア、LED 表示データの出力と LED 表示切替の制御

2. 引数説明

本タスク例では、引数を使用しておりません。

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 5 に示します。

表 5 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
TMA	タイマモードレジスタ A : プリスケーラ、入力クロックの選択	H'FFB0	H'0C (初期設定時)
TMA3	インターナルクロックセレクト 3 : タイマ A の動作モードの選択 TMA3 = 1 設定時、 : プリスケーラ W の出力をカウントする時計用 タイムベースとして動作	ビット 3	1
TMA2	インターナルクロックセレクト 2~0 : TMA3 = 1 のとき、時計用タイムベース (32.768kHz)を選択	ビット 2	0/1
TMA1	TMA2 = 1、TMA1 = 0、TMA0 = 0 設定時、 : TCA リセット	ビット 1	0
TMA0	TMA2 = 0、TMA1 = 0、TMA0 = 1 設定時、 : TCA オーバフロー周期は 0.5s	ビット 0	0/1
TMC	タイマモードレジスタ C : オートリロードの選択、カウンタのアップ/ダウン 制御、入力クロックの制御	H'FFB4	H'1B
TMC7	オートリロード機能の選択 : TMC7 = 0 のとき、インターバル機能を選択	ビット 7	0
TMC6	カウンタアップ/ダウン制御	ビット 6	0
TMC5	: TMC6 = 0、TMC5 = 0 のとき、TCC はアップカ ウンタ	ビット 5	0
TMC2	クロックセレクト	ビット 2	0
TMC1	: TMC2 = 0、TMC1 = 1、TMC0 = 1 のとき、内部 クロック /64 でカウント	ビット 1	1
TMC0		ビット 0	1
TLC	タイマロードレジスタ C : TCC のリロード値を設定	H'FFB5	H'00
AMR	A/D モードレジスタ : A/D 変換スピードの設定、外部トリガの選択、ア ナログ入力端子の指定	H'FFC6	H'34
CKS	A/D 変換スピードの設定 = 5MHz 設定時、 : CKS = 0 のとき、12.4 μ s	ビット 7	0

レジスタ名		機能説明	アドレス	設定値
	TRGE	外部トリガ入力による A/D 変換の開始を許可または禁止 : TRGE = 0 のとき、外部トリガによる A/D 変換の開始を禁止	ビット 6	0
	CH3	チャンネルセレクト 3~0 : CH3 = 0、CH2 = 1、CH1 = 0、CH0 = 0 のとき、ANO を選択	ビット 3	0
	CH2		ビット 2	1
	CH1		ビット 1	0
	CH0		ビット 0	0
ADSR	A/D スタートレジスタ : A/D 変換の開始または停止を指定	H'FFC7	-	
	ADSF	A/D 変換の開始および終了の確認 リード時 : ADSF = 0 のとき、A/D 変換の終了 : ADSF = 1 のとき、A/D 変換中 ライト時 : ADSF = 0 のとき、A/D 変換を強制終了 : ADSF = 1 のとき、A/D 変換を開始	ビット 7	0/1
ADRRH	A/D リザルトレジスタ : 上位 8 ビットが格納	H'FFC4	-	
ADRRL	A/D リザルトレジスタ : 下位 2 ビットがビット 7、ビット 6 に格納	H'FFC5	-	
PUCR6	ポートプルアップコントロールレジスタ 6 : 入力ポートに設定されたポート 6 の各端子のプルアップ MOS をビットごとに制御 PUCR6 = H'00 のとき、 : P67 ~ P60 端子のプルアップ MOS はオフ	H'FFE3	H'00	
PDR6	ポートデータレジスタ 6 : ポート 6 の汎用入出力ポートデータレジスタ	H'FFD9	H'00	
PCR6	ポートコントロールレジスタ 6 : ポート 6 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 PCR6 = H'FF のとき、 : P67 ~ P60 端子は汎用出力端子として機能	H'FFE9	H'FF	
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'FFCC	H'00	
	WKP7	P57/_WKP7/SEG7 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
	WKP6	P56/_WKP6/SEG6 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/SEG5 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/SEG4 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/SEG3 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/SEG2 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/SEG1 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/SEG0 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

レジスタ名	機能説明	アドレス	設定値
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'FFE2	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
IENR1	割り込み許可レジスタ 1 : 割り込み要求の許可 / 禁止を制御	H'FFF3	-
IENR1 IENTA	タイマ A 割り込み要求イネーブル : 1 のとき、タイマ A のオーバーフロー割り込み要求を許可	ビット 5	1
IRR1	割り込み要求レジスタ 1 : タイマ A、IRQ4、IRQ3、IRQAEC、IRQ1、IRQ0 割り込み要求が発生すると対応するフラグが 1 にセットされる	H'FFF6	-
IRR1 IRRTA	タイマ A 割り込み要求フラグ : タイマ A のカウンタ値がオーバーフロー(H'FF H'00)したときに 1 にセット : IRRTA に 0 をライトしたときに 0 クリア	ビット 7	0/1
IENR2	割り込み許可レジスタ 2 : 割り込み要求の許可 / 禁止を制御	H'FFF4	-
IENR2 IENTC	タイマ C 割り込み要求イネーブル : 1 のとき、タイマ C のオーバーフローまたはアンダーフロー割り込み要求を許可	ビット 1	1
IRR2	割り込み要求レジスタ 2 : 直接遷移、A/D 変換器、タイマ G、タイマ FH、タイマ FL、タイマ C、非同期イベントカウンタ割り込み要求が発生すると対応するフラグが 1 にセットされる	H'FFF7	-
IRR2 IRRTC	タイマ C 割り込み要求フラグ : タイマ C のカウンタ値がオーバーフロー(H'FF H'00)、またはアンダーフロー(H'00 H'FF)したときに 1 にセット : IRRTC に 0 をライトしたときに 0 クリア	ビット 7	0/1

4. 使用 RAM 説明

表 6 に本タスク例における使用 RAM 説明を示します。

表 6 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
vout	電圧値	H'FB80	tmra
ang	角速度値	H'FB84	tmra
angi	10 倍角速度値	H'FB88	tmra
SCALE	スケール・ファクタ	H'FB8A	main, tmra
RATE	増幅率	H'FB8E	main, tmra
*ptr	dig_0 のアドレスを格納するポイント	H'FB92	tmrc
dig_0	LED1 の表示データを格納 (1 byte)	H'FB94	main, tmra, tmrc
dig_1	LED2 の表示データを格納 (1 byte)	H'FB95	main, tmra
dig_2	LED3 の表示データを格納 (1 byte)	H'FB96	main, tmra
dig_3	LED4 の表示データを格納 (1 byte)	H'FB97	main, tmra
cnt	LED1 ~ LED4 の表示切替のための 8 ビットカウンタ (1 byte)	H'FB98	main, tmrc
lednum0	LED1 の表示データ	H'FB99	tmra
lednum1	LED2 の表示データ	H'FB9A	tmra
lednum2	LED3 の表示データ	H'FB9B	tmra

5. データテーブル説明

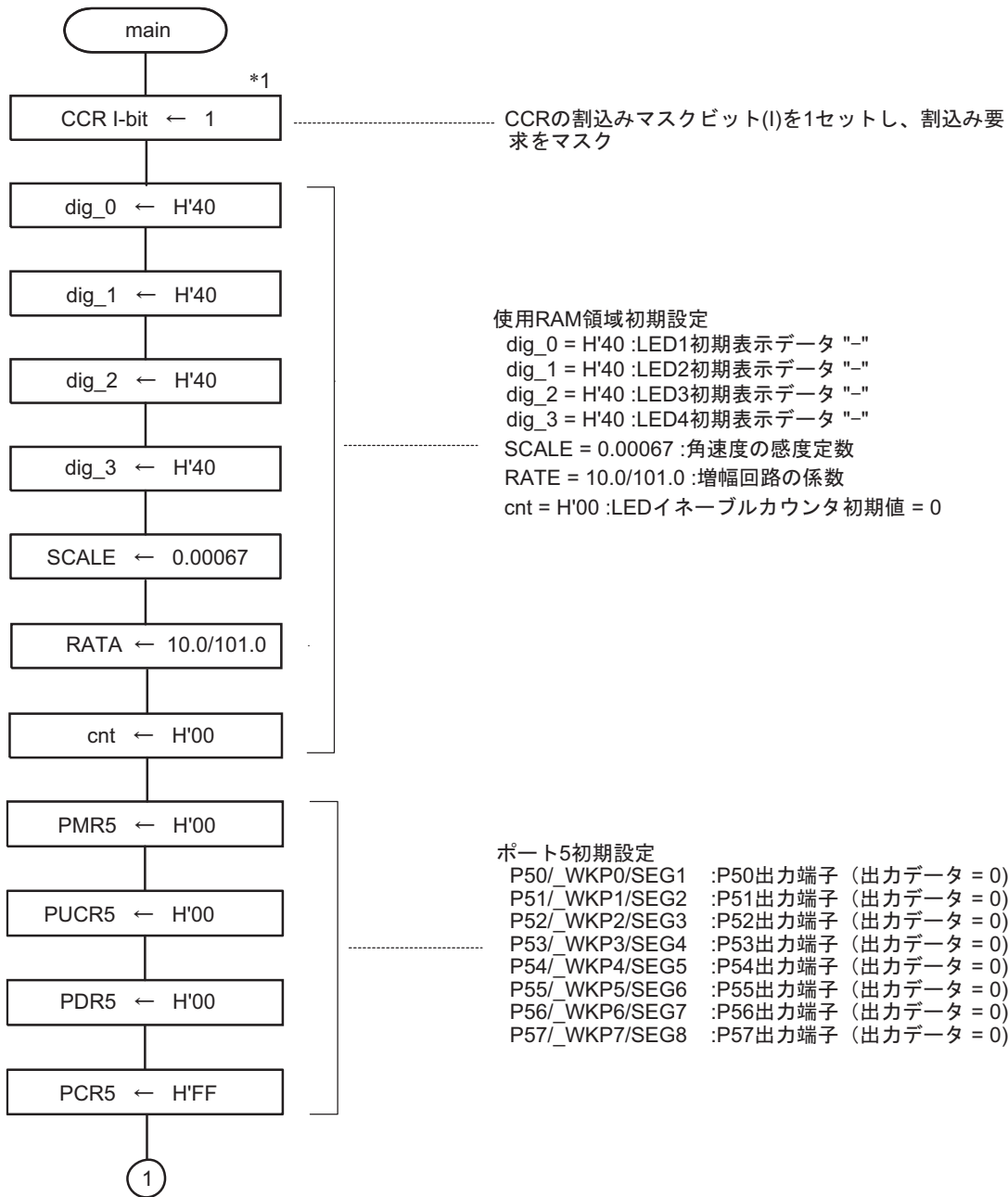
本タスク例では 7 セグメント LED の表示データを 1 次元配列のデータテーブルとして ROM に格納しています。表 7 に 7 セグメント LED 表示データテーブル(dsp_data [])説明を示します。

表 7 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 byte	H'10FC
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 byte	H'10FD
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 byte	H'10FE
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 byte	H'10FF
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 byte	H'1100
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 byte	H'1101
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 byte	H'1102
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 byte	H'1103
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 byte	H'1104
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 byte	H'1105

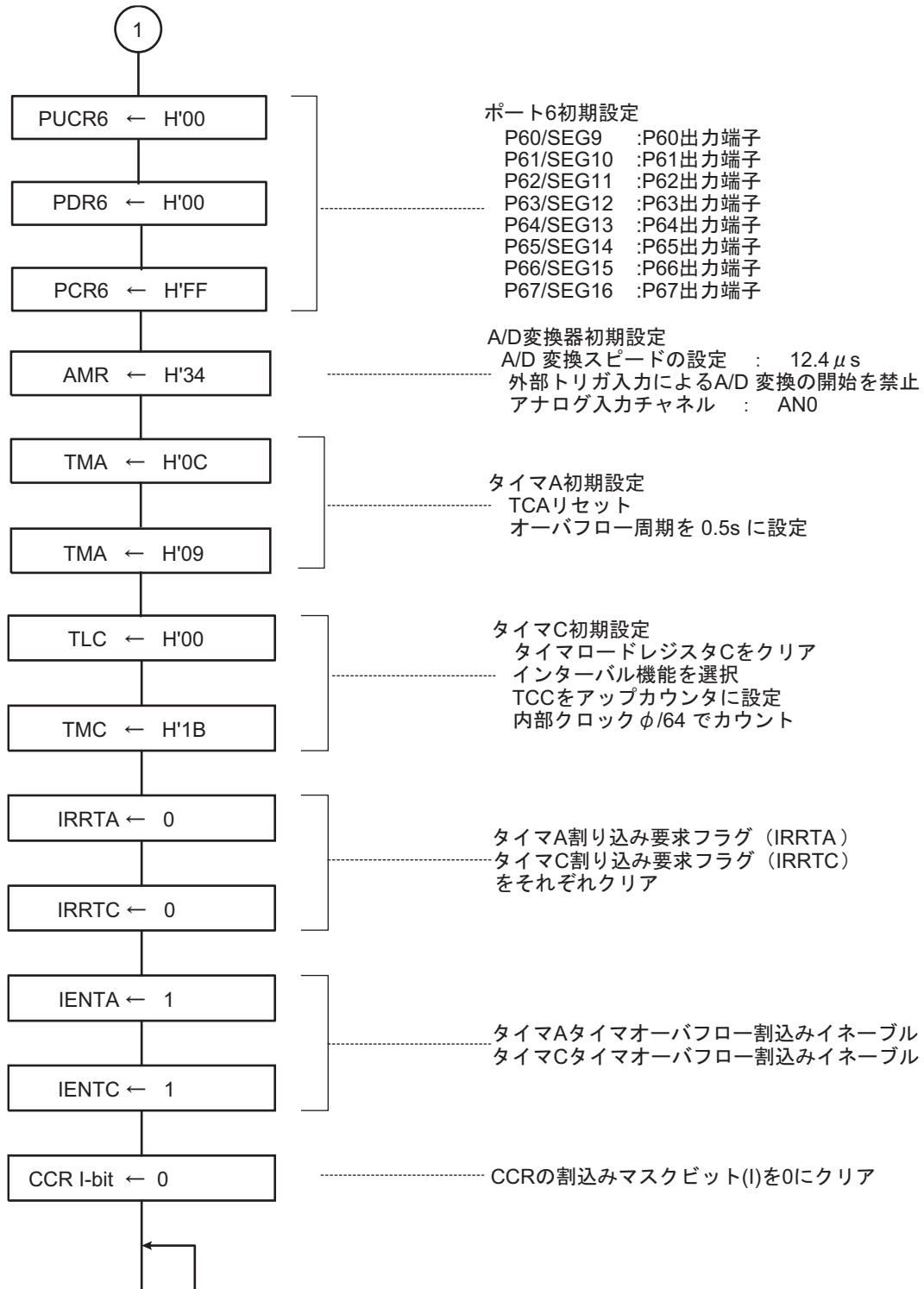
5. フローチャート

1. メインルーチン(main)

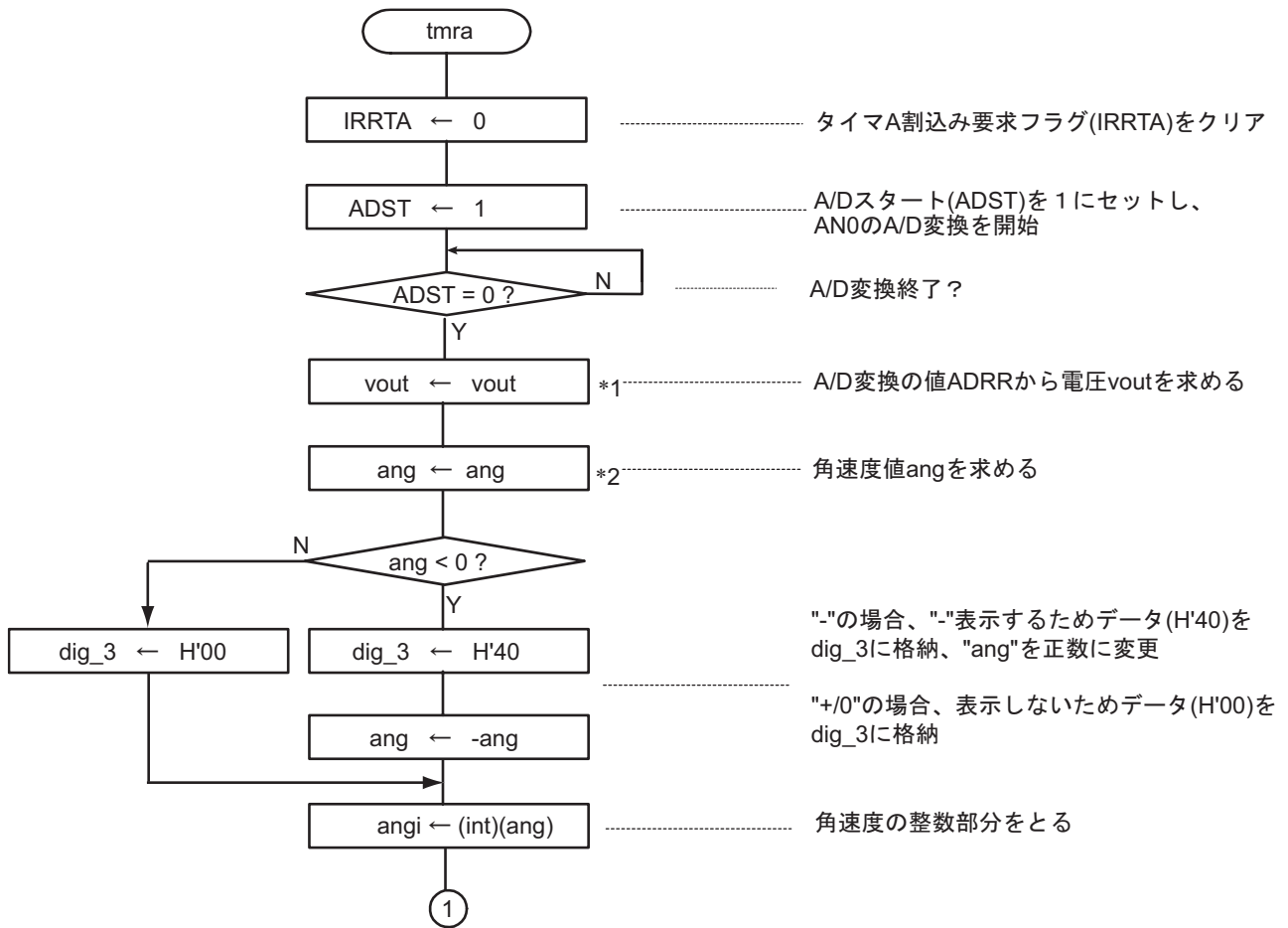


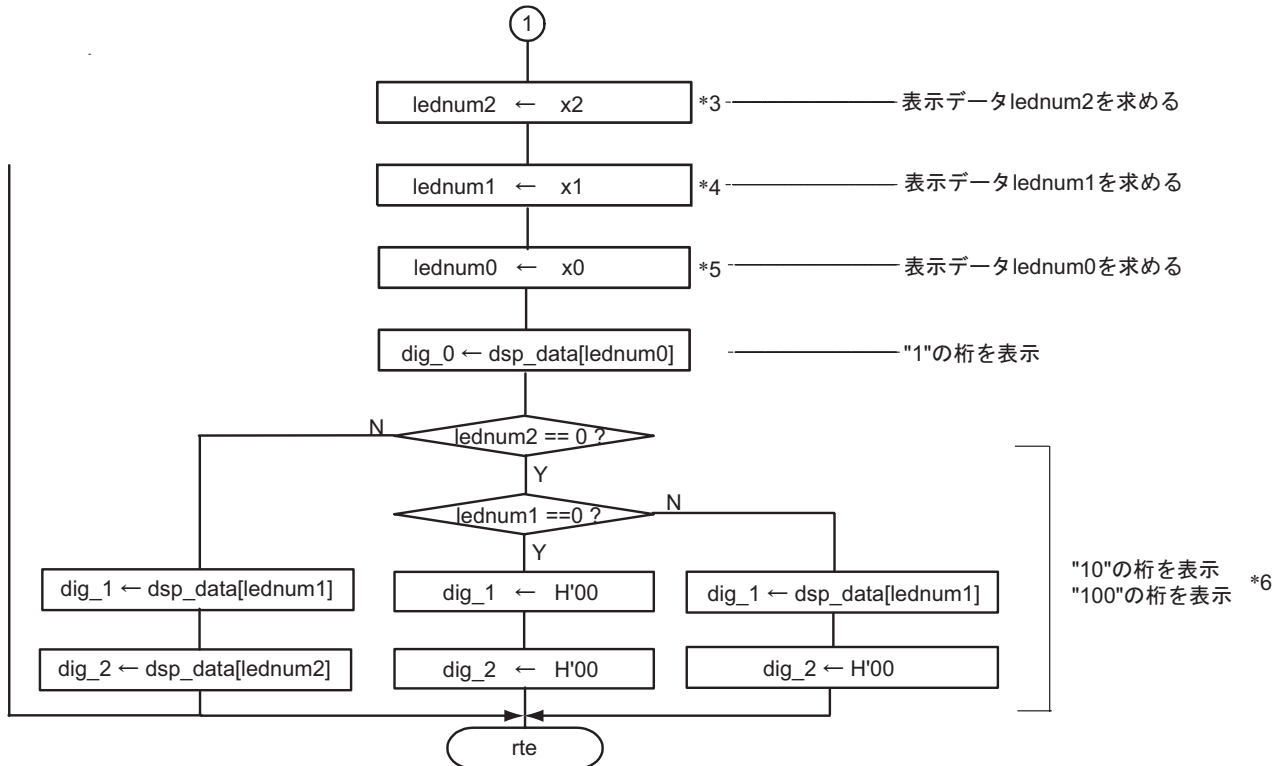
【注】

*1 : 本タスク例ではスタックポインタの設定は INIT.SRC (アセンブリ言語) で行っています。



2. タイマ A 割込み処理ルーチン(tmra)



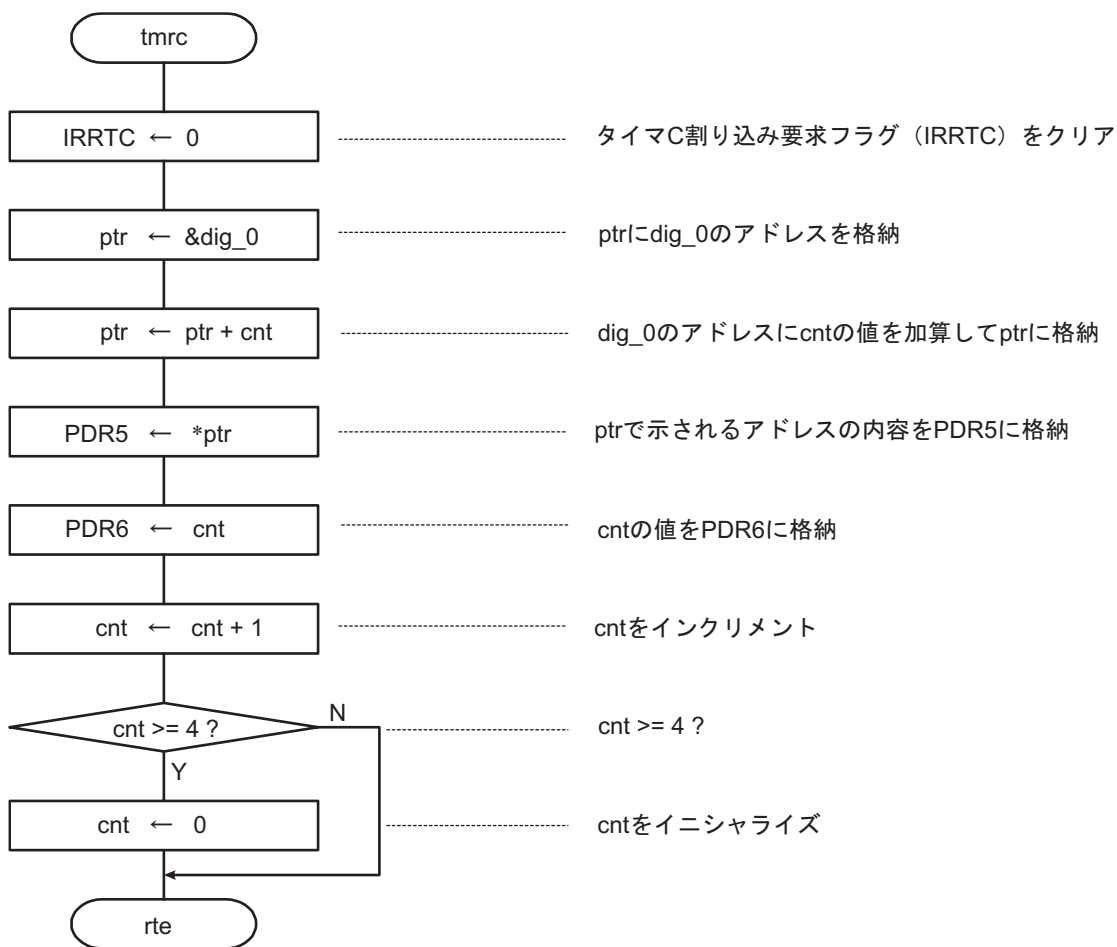


【注】

- *1 : $vout = (ADRR \gg 6 \& 0x03ff) * 3.3 / 1023.0$
- *2 : $ang = (vout - 1.35) * RATE / SCALE$
- *3 : $x2 = (\text{unsigned char})(angi / 100)$
- *4 : $x1 = (\text{unsigned char})(angi \% 100 / 10)$
- *5 : $x0 = (\text{unsigned char})(angi \% 10)$
- *6 : "lednum2 = 0"の場合 :
 - (1)"lednum1 = 0"の場合、"100"及び"10"の桁は表示しないため、データ(H'00)をdig_2及びdig_1に格納
 - (2)"lednum1 > 0"の場合、"100"の桁は表示しないため、データ(H'00)をdig_2に格納。"10"の桁は表示データをdsp_dataから参照し、dig_1に格納

"lednum2 > 0"の場合、"100"及び"10"の桁は表示データをdsp_dataから参照し、dig_2及びdig_1に格納

3. タイマ C 割込み処理ルーチン(tmrc)



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
    
```

```

/* Super Low Power Series -H8/38024- Application note */
/* 応用編 */
/* 角速度センサジャイロスター接続例 */
    
```

```
#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 PMR5 *(volatile unsigned char *)0xFFCC /* Port mode register 5 */
#define PUCR5 *(volatile unsigned char *)0xFFE2 /* 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 PUCR6 *(volatile unsigned char *)0xFFE3 /* Port pull-up control register 6 */
#define PDR6 *(volatile unsigned char *)0xFFD9 /* Port data register 6 */
#define PCR6 *(volatile unsigned char *)0xFFE9 /* Port control register 6 */

#define TMA *(volatile unsigned char *)0xFFB0 /* Timer mode register A */
#define CKSTPR1 *(volatile unsigned char *)0xFFFA /* Clock stop register 1 */

#define TMC *(volatile unsigned char *)0xFFB4 /* Timer mode register C */
#define TLC *(volatile unsigned char *)0xFFB5 /* Timer Load register C */
#define ADDR *(volatile unsigned int *)0xFFC4 /* A/D result register(word access) */
#define ADDRH *(volatile unsigned int *)0xFFC4 /* A/D result register(byte access) */
#define ADDRLL *(volatile unsigned int *)0xFFC5 /* A/D result register(byte access) */
#define AMR *(volatile unsigned char *)0xFFC6 /* A/D mode register */
#define ADSR *(volatile unsigned char *)0xFFC7 /* A/D start register */
#define ADSR_BIT (*(struct BIT *)0xFFC7) /* A/D start */
#define ADST ADSR_BIT.b7 /* A/D start */
#define IRR1 *(volatile unsigned char *)0xFFFF6 /* Interrupt request register 1 */
#define IRR1_BIT (*(struct BIT *)0xFFFF6) /* Interrupt request register 1 */
#define IRRTA IRR1_BIT.b7 /* Timer A interrupt request flag */
    
```

```

#define IENR1 *(volatile unsigned char *)0xFFFF3 /* Interrupt enable register 1 */
#define IENR1_BIT (*(struct BIT *)0xFFFF3) /* Interrupt enable register 1 */
#define IENTA IENR1_BIT.b7 /* Timer A interrupt enable */

#define IRR2 *(volatile unsigned char *)0xFFFF7 /* Interrupt request register 2 */
#define IRR2_BIT (*(struct BIT *)0xFFFF7) /* Interrupt request register 2 */

#define IRRTC IRR2_BIT.b1 /* Timer C interrupt request flag */
#define IENR2 *(volatile unsigned char *)0xFFFF4 /* Interrupt enable register 2 */
#define IENR2_BIT (*(struct BIT *)0xFFFF4) /* Interrupt enable register 2 */
#define IENTC IENR2_BIT.b1 /* Timer C interrupt enable */

#pragma interrupt (tmra)
#pragma interrupt (tmrc)

/* 関数定義 */
extern void INIT(void); /* Stack pointer set */
void main(void); /* main routine */
void tmra(void); /* Timer A interrupt routine */
void tmrc(void); /* Timer C interrupt routine */

/* Data table */
const unsigned char dsp_data[10] =
{
    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" */
};

/* 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 */
float vout; /* OPamp analog voltage output */
float ang; /* angular velocity result */
int angi; /* intger angular velocity result */
float SCALE; /* Scale factor const */
float RATE; /* Amplify rate const */
unsigned char lednum0; /* Dig-0 LED display data */
unsigned char lednum1; /* Dig-1 LED display data */
unsigned char lednum2; /* Dig-2 LED display data */
unsigned char *ptr; /* Pointer set */
/* 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) = {
    tmra                                     /* H'0016 Timer A interrupt vector */
};
#pragma section V3                          /* Vector section set */
void (*const VEC_TBL3[])(void) = {
    tmrc                                     /* H'001a Timer C 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 */
    SCALE = 0.00067;                        /* Scale factor */
    RATE = 10.0/101.0;                      /* Amplify rate */
    cnt = 0x00;                             /* Used RAM area initialize */

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

    PUCR6 = 0x00;                           /* Port 6 initialize */
    PDR6 = 0x00;
    PCR6 = 0xFF;

    AMR = 0x34;                             /* A/D converter initialize (AN0) */

    TMA = 0x0c;                             /* Clear Timer Counter A to 0 */
    TMA = 0x09;                             /* Timer A initialize */
    TLC = 0x00;                             /* Clear Timer Load register C */
    TMC = 0x1b;                             /* Timer C initialize */

    IRRTA = 0;                              /* Clear IRRTA to 0 */
    IRRTC = 0;                              /* Clear IRRTC to 0 */
    IENTA = 1;                              /* Timer A interrupt enable */
    IENTC = 1;                              /* Timer C interrupt enable */

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

    while (1);
}
    
```



```

/*****/
/* Timer A Interrupt */
/*****/
void tmra(void)
{
    IRRTA = 0; /* Clear IRRTA to 0 */
    ADST = 1; /* A/D converter start */
    While (ADST == 1); /* A/D converter end ? */
    vout = (ADRR >> 6 & 0x03ff) * 3.3 / 1023.0; /* Output voltage */
    ang = (vout - 1.35) * RATE / SCALE; /* Angular velocity sensor result */
    if (ang < 0) {
        dig_3 = 0x40; /* Dig-3 LED display data set */
        ang = -ang; /* ang = -ang */
    }else{
        dig_3 = 0x00; /* Dig-3 LED display data set */
    }
    angi = (int)(ang); /* int angular velocity result */
    lednum2 = (unsigned char)(angi / 100); /* Compute Dig-2 LED display data */
    lednum1 = (unsigned char)(angi % 100 / 10); /* Compute Dig-1 LED display data */
    lednum0 = (unsigned char)(angi % 10); /* Compute Dig-0 LED display data */
    dig_0 = dsp_data[lednum0]; /* Dig-0 LED display data set */
    if (lednum2 == 0){
        if (lednum1 == 0){
            dig_1 = 0x00; /* Dig-1 LED display data set */
            dig_2 = 0x00; /* Dig-2 LED display data set */
        }else{
            dig_1 = dsp_data[lednum1]; /* Dig-1 LED display data set */
            dig_2 = 0x00; /* Dig-2 LED display data set */
        }
    }else{
        dig_1 = dsp_data[lednum1]; /* Dig-1 LED display data set */
        dig_2 = dsp_data[lednum2]; /* Dig-2 LED display data set */
    }
}
/*****/
/* Timer C Interrupt */
/*****/
void tmrc(void)
{
    IRRTC = 0; /* Clear IRRTC to 0 */

    ptr = &dig_0; /* LED data store address set */
    ptr += cnt; /* LED display data read */
    PDR5 = *ptr; /* LED display data output */
    PDR6 = 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	2003.09.18	—	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。