

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。

M32C/83、85 グループ

インテリジェント I/O を用いたステッピングモータ制御

1. 要約

この資料は M32C/83 グループのインテリジェント I/O グループ 2、および M32C/85 グループのインテリジェント I/O の波形生成機能を用いたステッピングモータ制御波形出力方法についてまとめたものです。

2. はじめに

この資料で説明する応用例は、次のマイコン、条件での利用に適用されます。

- ・マイコン :M32C/83 グループ
- :M32C/85 グループ

M32C/83、85 グループと同様の SFR(周辺機能制御レジスタ)を持つ他の M16C ファミリでも本プログラムを使用することができます。ただし、一部の機能を機能追加等で変更している場合がありますのでマニュアルで確認してください。このアプリケーションノートをご使用に際しては十分な評価を行ってください。

3. 応用技術の説明

3.1. ステッピングモータ制御

ステッピングモータは DC モータのように電源に接続するだけでは回転しません。

パルス信号を与えて決められたステップで回転するモータです。

2相ユニポーラ型のステッピングモータの制御方法について簡単に説明します。

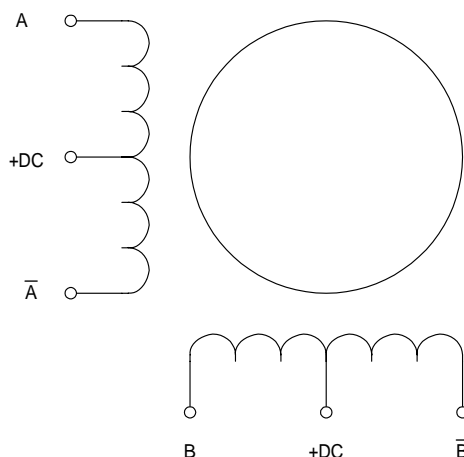


図 1 ユニポーラ型ステッピングモータ

図 1 の A、 \bar{A} 、B、 \bar{B} を決まった順番で励磁することによってステッピングモータは回転します。
この励磁方式にはいくつかの方法があります。

3.2. 励磁方式

3.2.1. 1 相励磁

常時 1 相だけ励磁する方式。消費電力は少ないが出力トルクが小さくダンピング効果が低いいため振動が発生しやすくなります。

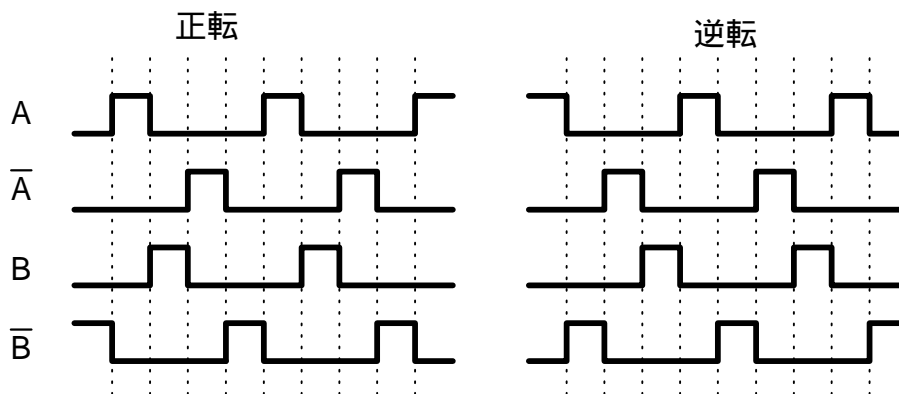


図 2 1 相励磁

3.2.2. 2 相励磁

常時 2 相を励磁する方式。1 相に比べて、2 倍の電力を消費するが、その分出力トルクが大きくダンピング効果が高いため、多く用いられています。

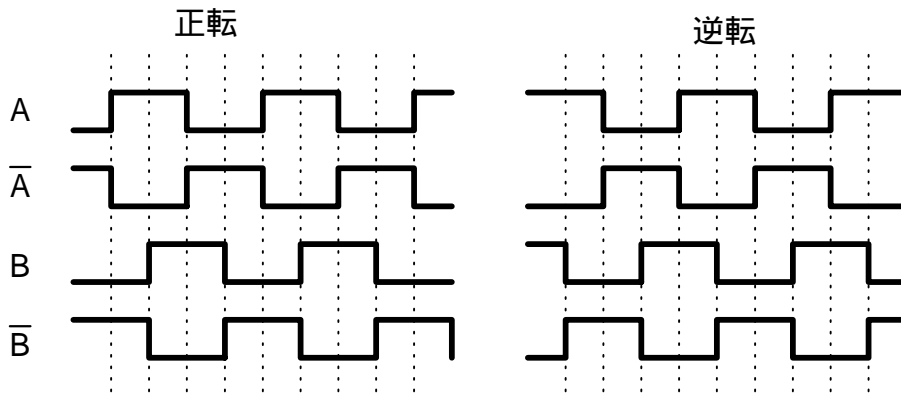


図 3 2相励磁

3.2.3. 1 - 2 相励磁

1 つの相と 2 つの相を交互に励磁する方式。ステップ角が 1 相励磁、2 相励磁の 1/2 になります (ハーフステップ動作)。このため回転がスムーズになり共振などが発生しにくくなります。

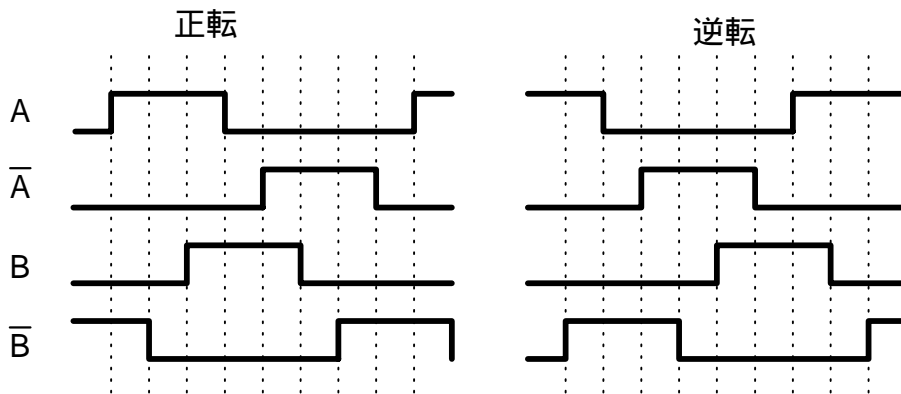


図 4 1 - 2相励磁

3.3. マイクロステップ制御

ステッピングモータは性質上低速回転時、大きな振動が発生します。マイクロステップ制御を行うことにより通常の矩形波制御に比べて回転がなめらかになります。

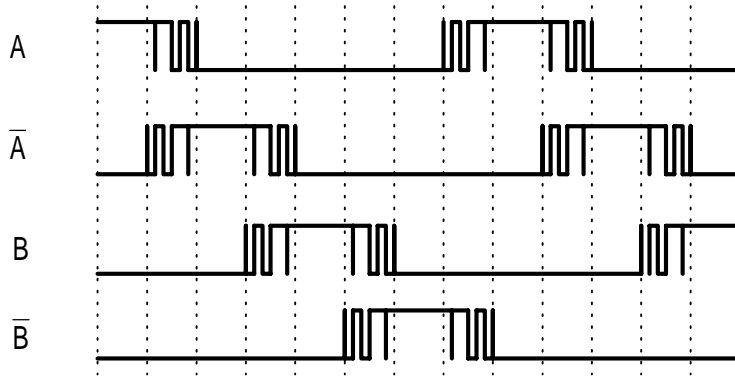


図 5 マイクロステップ制御（1 - 2 相励磁）

3.4. 波形生成機能の使い方

本章では、インテリジェント I/O の波形生成機能を使用したマイクロステップ制御について解説します。図 6 は M32C/83 とステッピングモータの接続例です。本接続例ではインテリジェント I/O グループ 2 を使用しています。解説では具体的にグループ 2 のレジスタの設定方法を説明します。

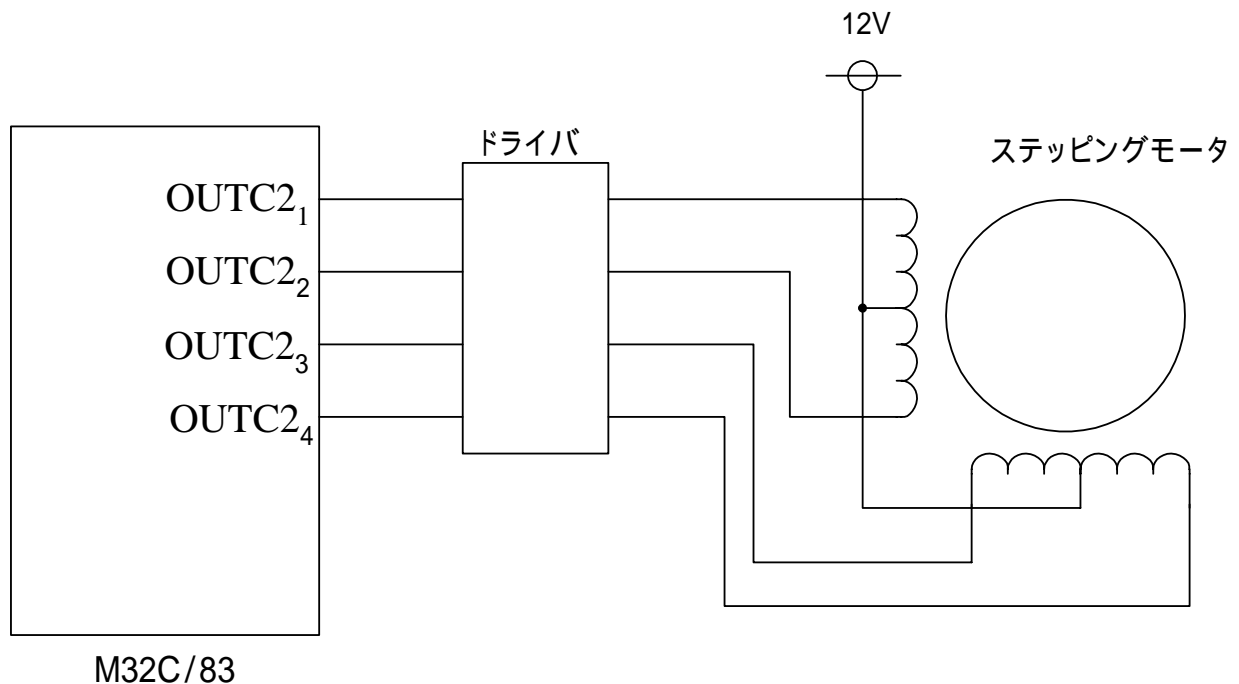


図 6 ステッピングモータ接続例

3.4.1. キャリア周波数

PWMパルス幅の基準となる周波数をキャリア周波数と呼びます。このキャリア周波数を基準に波形のデューティを変化させていきます。

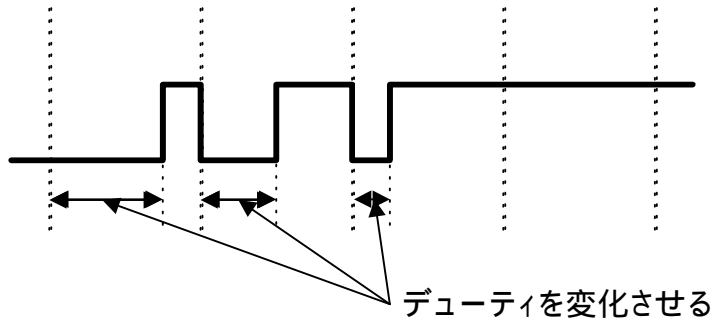


図 7 キャリア周波数

グループ2 ベースタイマのリセットで割り込み要求が発生します。その割り込み処理で波形生成レジスタの値を変更することによってPWM出力パルスのL幅を変更しています。グループ2 ベースタイマのリセット条件は「グループ2 波形生成レジスタ 0 (G2PO0 レジスタ)の値との一致」としています。

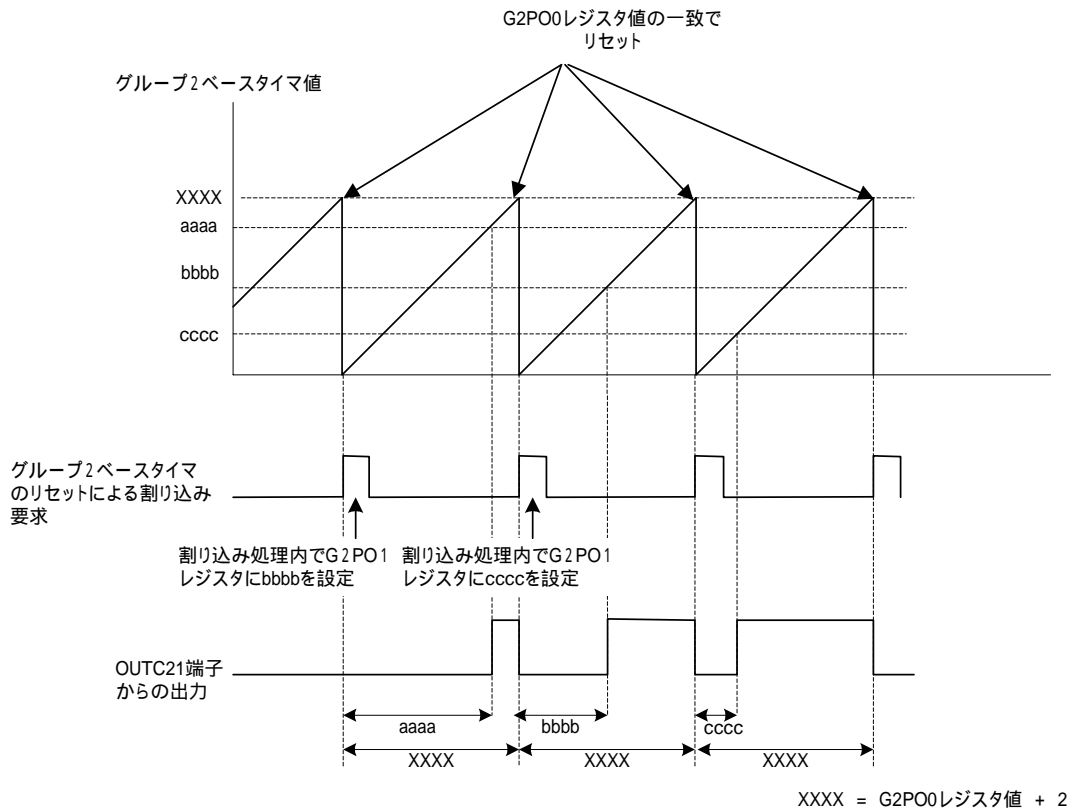


図 8 PWM出力タイミング

3.4.2. デューティ比の決め方

PWM パルスのデューティは図 9 の様に sin 波形と重ねて算出しています。
sin 値が負の部分は L 出力しています。正の部分で一定値以上の部分は H 出力しています。

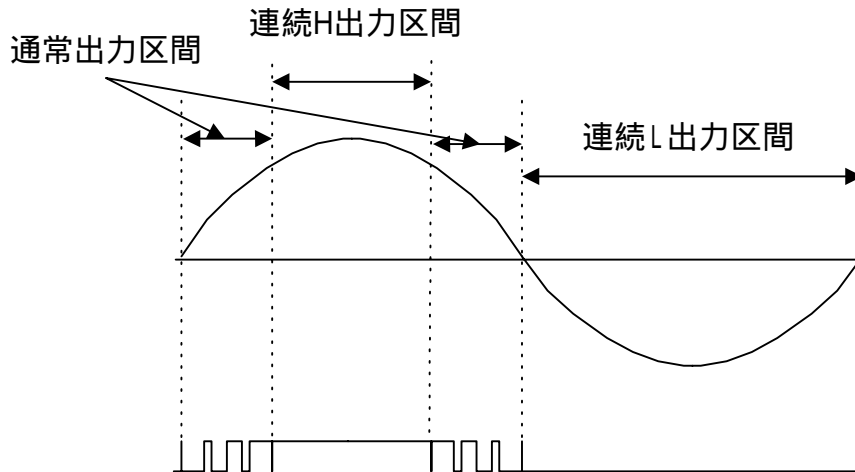


図 9 Sin 波形と PWM 出力の関係

波形生成レジスタへの設定値は以下のように算出します。

通常出力区間

$$\text{波形生成レジスタ値 (PWM出力のL幅)} = \text{PWM周期} - \text{PWM周期} \times \sin N^\circ$$

連続H出力区間

$$\text{波形生成レジスタ値} = 0$$

連続L出力区間

$$\text{波形生成レジスタ値} = \text{PWM周期} + 1$$

設定することにより図 9 の出力波形を出力する。

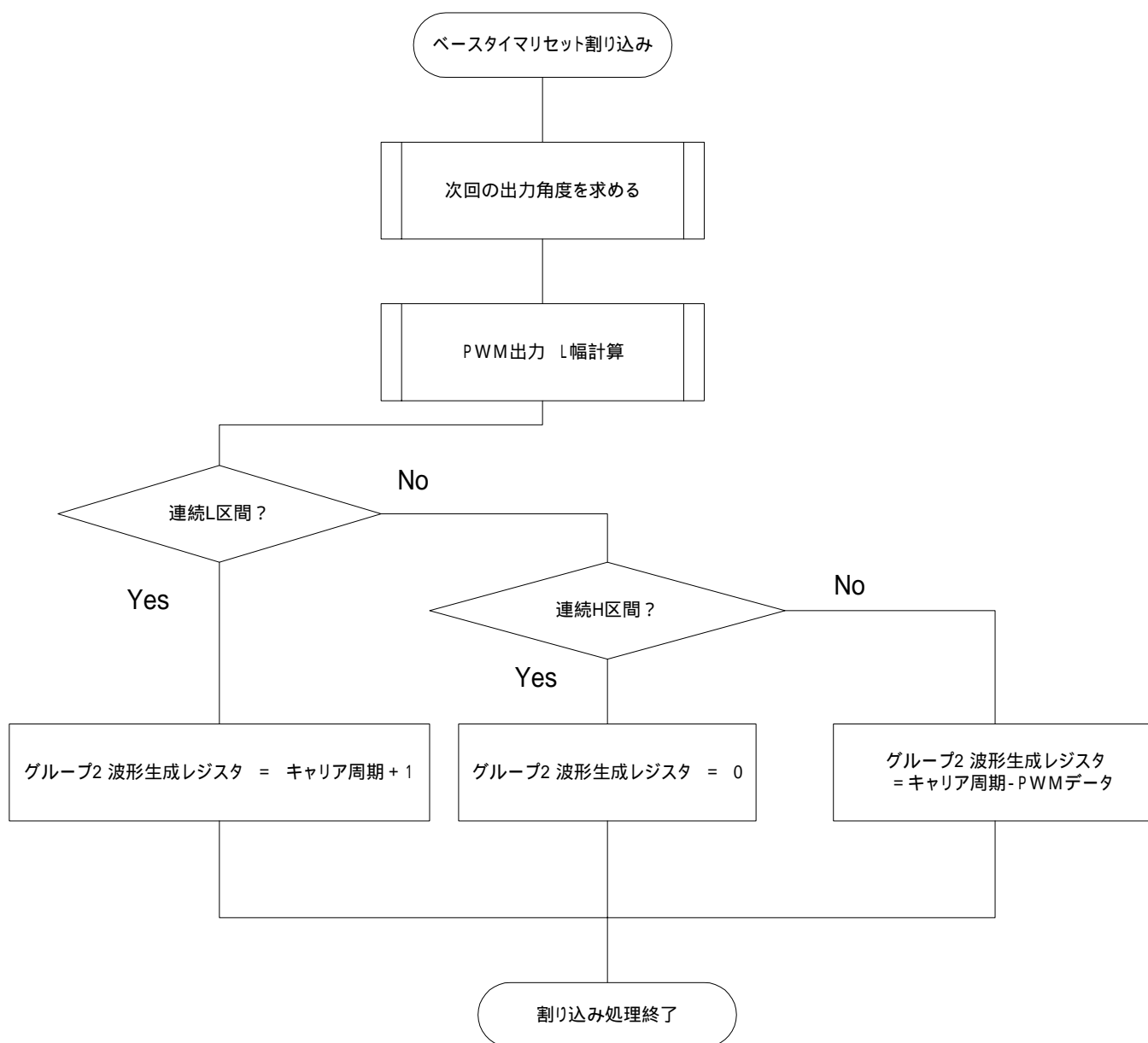


図 10 ベースタイマリロード割り込み処理例

3.4.3. 回転速度の変更

ステッピングモータの回転速度は出力する PWM 出力のキャリア周波数を上げると加速し、下げると減速します。

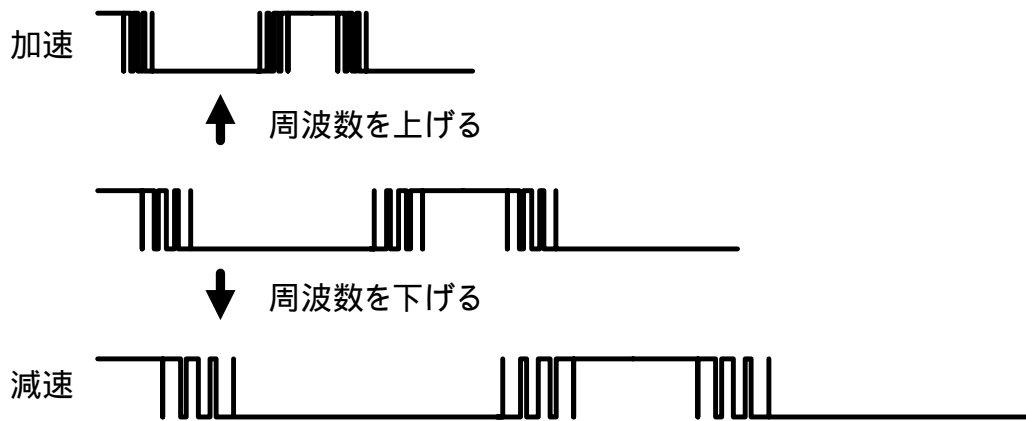


図 11 回転速度変更

図 12はキャリア周波数の変更方法の1例です。キャリア周波数は固定しておき、パルスL幅の変更するキャリア周期の回数を変更することによりキャリア周波数を変更しています。

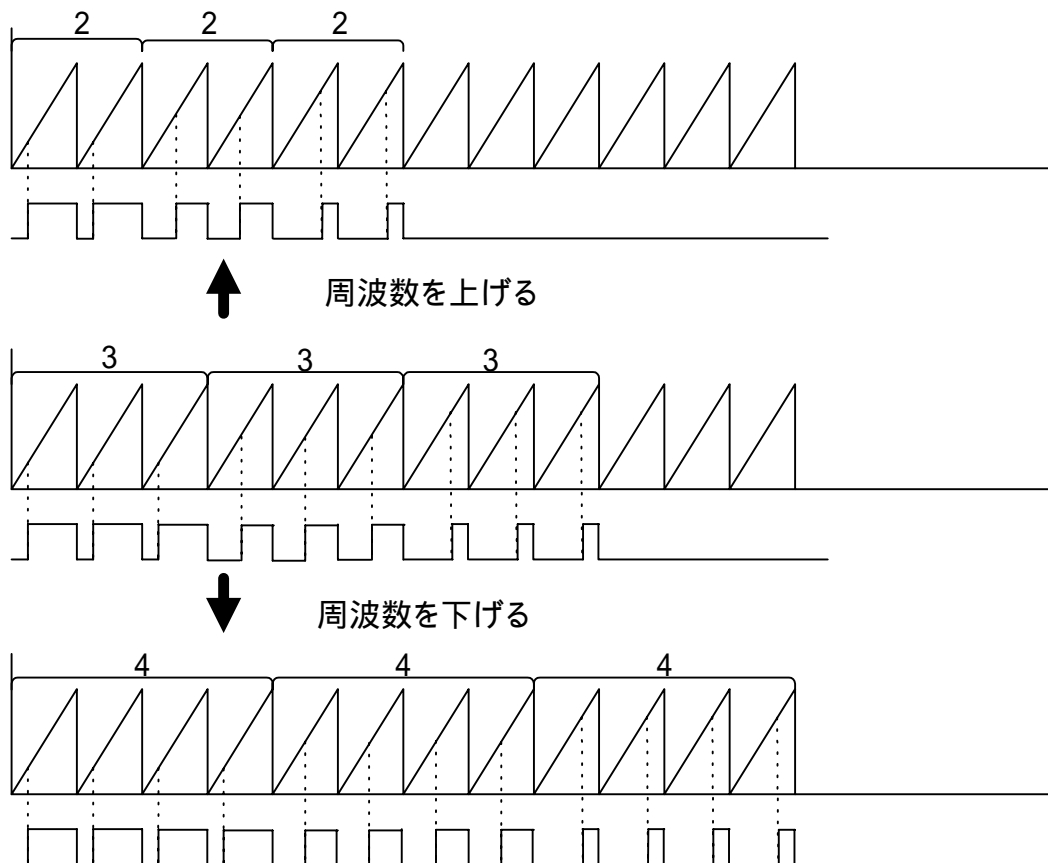


図 12 キャリア周波数変更方法例

3.4.4. 回転方向の変更

ステッピングモータの回転方向の変更は用意しているsin値テーブルの参照方向を変更することによって行います。図 13では、48°点で回転方向を切り替えています。

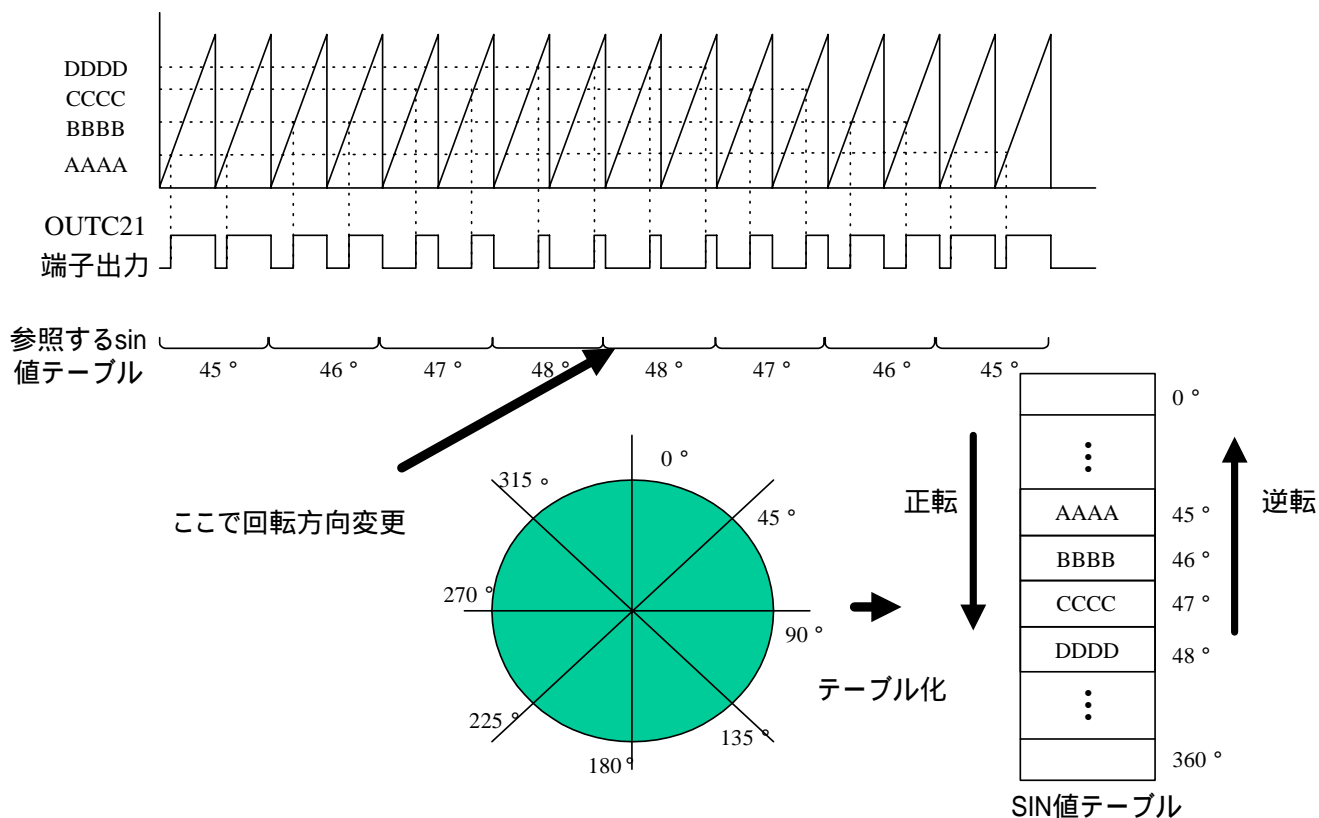


図 13 回転方向の変更

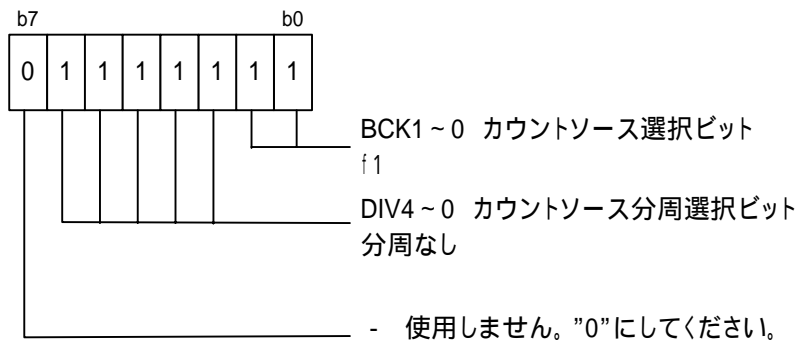
3.5. 設定方法

「3.4波形生成機能の使い方」を実現するための初期設定手順と設定値を示します。各レジスタの詳細は、M32C/83グループハードウェアマニュアルを参照願います。

(1) 割り込み禁止

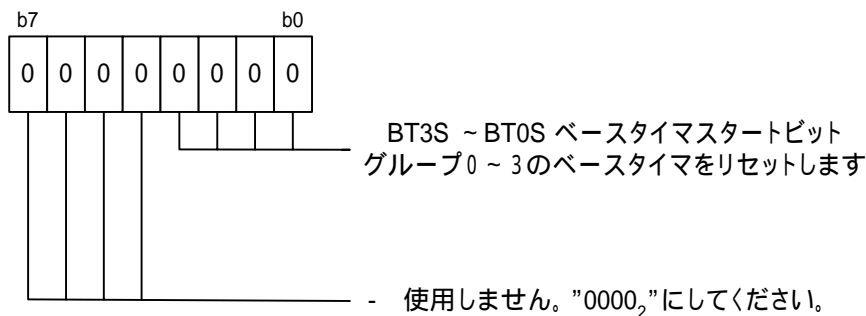
IFLAG = "0" または、使用するインテリジェント I/O からの割り込み要求が割り付けられる IIOkIC レジスタ (k=0~11) の ILV2~0 を "000₂" にしてください。

(2) G2BCR0 レジスタ



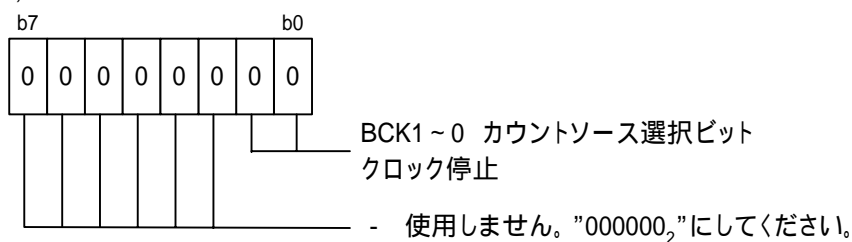
この設定により、BTSR レジスタにクロックが供給され、BTSR レジスタの設定が可能になります

(3) BTSR レジスタ



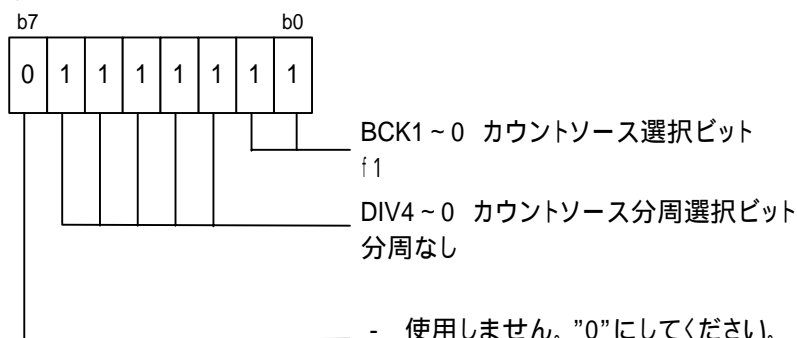
この設定により、グループ0 ~ 3のベースタイマがリセットされます。GiBCR0 レジスタでベースタイマの動作クロックを設定した後、GiBCR 1 レジスタの BTS ビットを "1" にすると、グループ i のベースタイマは "0000₁₆" からカウントします。

(4) G2BCR0 レジスタ



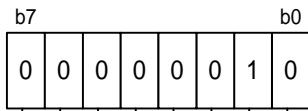
グループ2 および BTSR レジスタを使用しない場合は、グループ2 に供給されるクロックを停止させてください。

(5) G2BCR0 レジスタ



この設定により、以下(6)~(13)、(20)で示す各レジスタにクロックが供給されます。各レジスタの設定値を設定直後に有効にするため、必ず "0 1111111₂" にしてください。

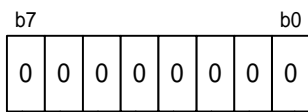
(6) G2BCR1レジスタ



- 使用しません。"0"にしてください。
- RST1 ベースタイマリセット要因選択ビット1
ベースタイマとG2PO0レジスタの一致
でベースタイマリセットする
- 使用しません。"0"にしてください。
- 予約ビット 必ず"0"に設定してください
- BTS ベースタイマスタートビット
ベースタイマリセット
- 予約ビット 必ず"00₂"に設定してください
- 使用しません。"0"にしてください。

グループ2インテリジェントI/O 関連レジスタ設定後、"1" (ベースタイマカウント開始) にしてください。

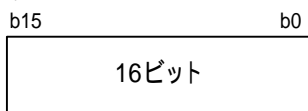
(7) G2POCR0レジスタ



- MOD2 ~ 0 動作モード選択ビット
単相波形出力モード
- 使用しません。"00₂"にしてください。
- RLD GiPO0 ~ 7 レジスタ値リロードタイミング選択ビット
書き込み時にリロード
- 使用しません。"00₂"にしてください。

GiPO0レジスタの設定値が書き込み直後有効になります。初期設定でGiPO0レジスタを設定する場合、必ずRLDビットを"0"にしてください。

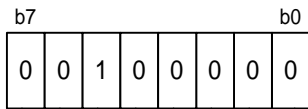
(8) G2PO0レジスタ



スタート時のPWM周期を設定します
設定値をnとすると周期は次の通りになります

$$\frac{\text{分周比}}{f_1} \times (n+2)$$

(9) G2POCR0レジスタ

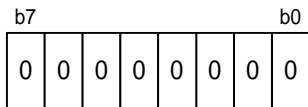


MOD2 ~ 0 動作モード選択ビット
単相波形出力モード
- 使用しません。"00₂"にしてください。

RLD GiPO0 ~ 7 レジスタ値リロードタイミング選択ビット
ベースタイマリセット時にリロード
- 使用しません。"00₂"にしてください。

RLD ビットを"1"にし、GiPO0、GiPOj レジスタのリロードタイミングをベースタイマリセット時に変更します。

(10) G2POCR1 ~ 4レジスタ



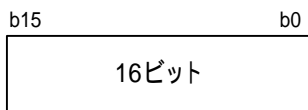
MOD2 ~ 0 動作モード選択ビット
単相波形出力モード
- 使用しません。"0"にしてください。

IVL 出力初期値選択ビット
初期値として"0"を出力

RLD GiPO0 ~ 7 レジスタ値リロードタイミング選択ビット
書き込み時にリロード
- 使用しません。"00₂"にしてください。

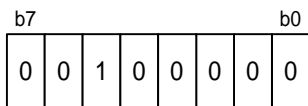
GiPO0レジスタの設定値が書き込み直後有効になります。初期設定でGiPO0 レジスタを設定する場合、必ずRLD ビットを"0"にしてください。

(11) G2PO1 ~ 4レジスタ



スタート時のPWM波形のL幅を設定します
ここでは初期値として出力がL固定になるように設定値をキャリア周期より長く設定していません

(12) G2POCR1 ~ 4レジスタ

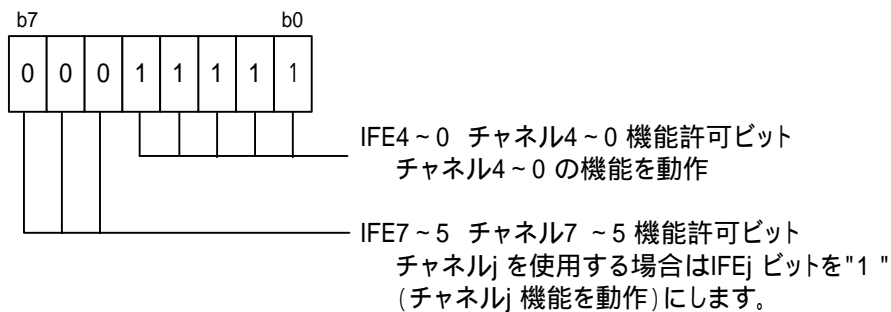


MOD2 ~ 0 動作モード選択ビット
単相波形出力モード
- 使用しません。"00₂"にしてください。

RLD GiPO0 ~ 7 レジスタ値リロードタイミング選択ビット
ベースタイマリセット時にリロード
- 使用しません。"00₂"にしてください。

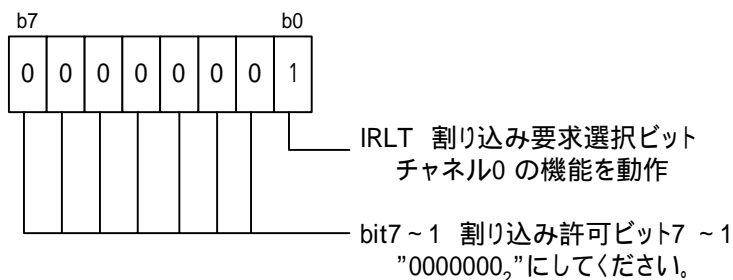
RLD ビットを"1"にし、G2PO1 ~ G2PO4 レジスタのリロードタイミングをベースタイマリセット時に変更します。

(13) G2FEレジスタ



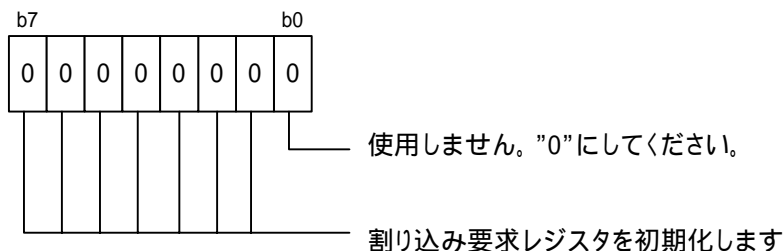
使用しないチャンネルj のIFEj ビットを"0"にしてください。

(14) IIO6IEレジスタ



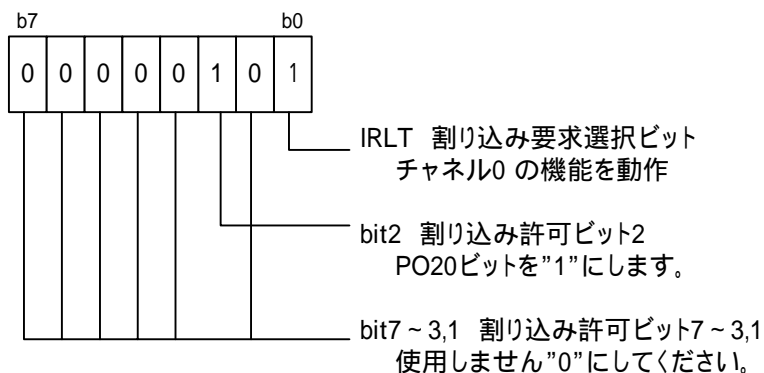
IRLTビットとbit7~1を同時に"1"にしないでください。

(15) IIO6IRレジスタ



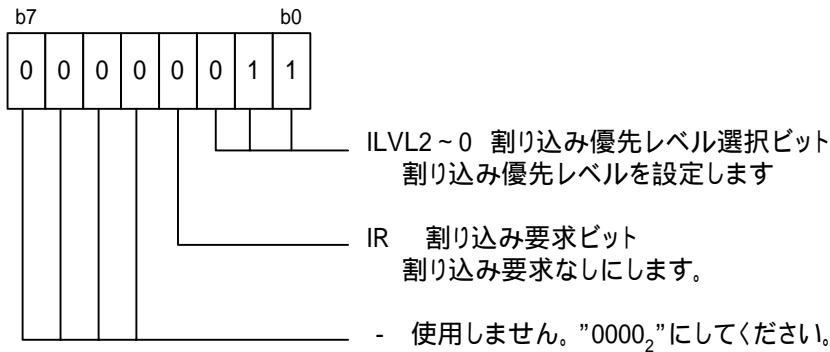
必ず、IIO6IRレジスタに"00₁₆"を設定してください。

(16) IIO6IEレジスタ

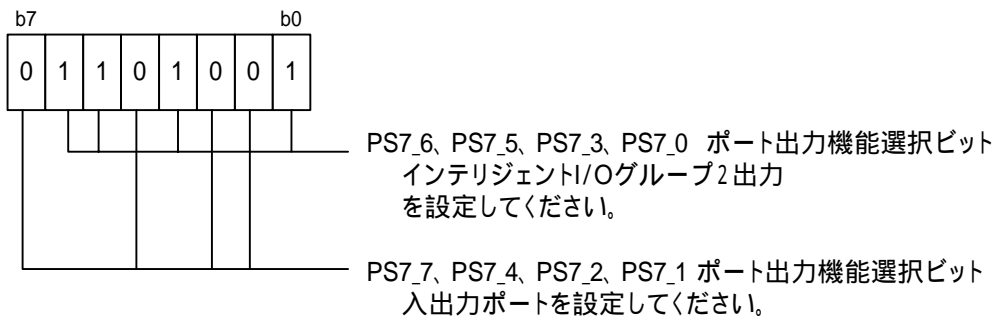


使用しない割り込みの割り込み要求ビットを"0"にします。

(17) IIO6ICレジスタ

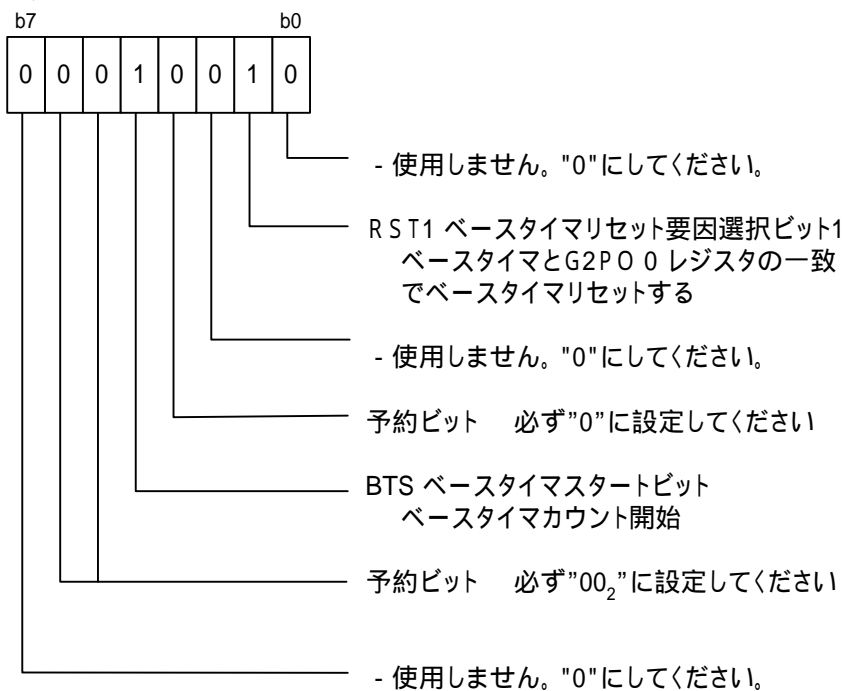


(18) 機能選択レジスタA7



(19) 割り込み許可(Iフラグ = "1")

(20) G2BCR1レジスタ



4. 参考プログラム例

ステッピングモータのマイクロステップ制御を実現する応用プログラム例を示します。応用プログラム例は各ユーザアプリケーションに応じて変更及び調整が必要です。

図 14 は、サンプルプログラム例の出力結果です。

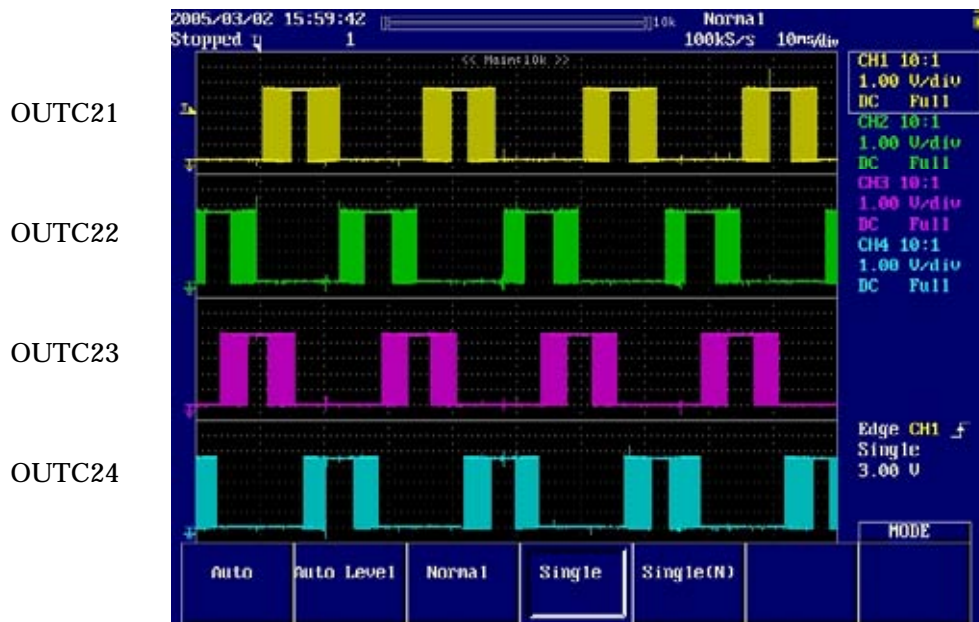


図 14 オシロスコープによる測定結果

メインクロック：30MHz
キャリア周波数：20KHz
回転速度：40PPS

4.1. M32C/83 用

```

/*****
/*  include file                */
/*****
#include "sfr32c83.h"

/*****
/*  Function prototype declaration  */
/*****
void init_motor( void );
void main_pro( void );
#pragma INTERRUPT/B iio_int
void iio_int( void );

```

```

void angle( void );
void i_con( void );
void pwm_set( void );

/*****/
/* Define for sine wave output */
/*****/
#define MCLK      30000000          /* Operation frequency:30MHz */
#define CARR_CLK  20000           /* Carrier frequency:20KHz */
#define C_DIV     1              /* Division rate of base timer */

#define CARR_SET   MCLK / CARR_CLK / C_DIV /* Carrier cycle */
#define DTT_SET   CARR_SET / 20 /* Judgment value of "H" fixed section */

/*****/
/* sin value table */
/*****/
const short sin_tbl[720]
={
    572, 1144, 1715, 2286, 2856, 3425, 3993, 4560, 5126, 5690,
    6252, 6813, 7371, 7927, 8481, 9032, 9580, 10126, 10668, 11207,
    11743, 12275, 12803, 13328, 13848, 14364, 14876, 15383, 15886, 16384,
    16876, 17364, 17846, 18323, 18795, 19260, 19720, 20174, 20621, 21063,
    21497, 21926, 22347, 22762, 23170, 23571, 23965, 24351, 24730, 25101,
    25465, 25821, 26169, 26509, 26842, 27165, 27481, 27788, 28087, 28377,
    28659, 28932, 29196, 29451, 29697, 29935, 30163, 30381, 30591, 30791,
    30982, 31164, 31336, 31498, 31651, 31794, 31928, 32051, 32165, 32270,
    32364, 32449, 32523, 32588, 32643, 32688, 32723, 32748, 32763, 32767,
    32763, 32748, 32723, 32688, 32643, 32588, 32523, 32449, 32364, 32270,
    32165, 32051, 31928, 31794, 31651, 31498, 31336, 31164, 30982, 30791,
    30591, 30382, 30163, 29935, 29697, 29451, 29196, 28932, 28659, 28378,
    28087, 27788, 27481, 27166, 26842, 26510, 26169, 25821, 25465, 25101,
    24730, 24351, 23965, 23571, 23170, 22762, 22347, 21926, 21497, 21063,
    20621, 20174, 19720, 19260, 18795, 18323, 17847, 17364, 16877, 16384,
    15886, 15383, 14876, 14364, 13848, 13328, 12803, 12275, 11743, 11207,
    10668, 10126, 9580, 9032, 8481, 7927, 7371, 6813, 6252, 5690,
    5126, 4560, 3993, 3425, 2856, 2286, 1715, 1144, 572, 0,
    -572, -1143, -1715, -2286, -2856, -3425, -3993, -4560, -5126, -5690,
    -6252, -6813, -7371, -7927, -8481, -9032, -9580, -10126, -10668, -11207,
    -11743, -12275, -12803, -13328, -13848, -14364, -14876, -15383, -15886, -16384,
    -16876, -17364, -17846, -18323, -18795, -19260, -19720, -20174, -20621, -21062,
    -21497, -21926, -22347, -22762, -23170, -23571, -23965, -24351, -24730, -25101,
    -25465, -25821, -26169, -26509, -26841, -27165, -27481, -27788, -28087, -28377,
    -28659, -28932, -29196, -29451, -29697, -29935, -30163, -30381, -30591, -30791,
    -30982, -31164, -31336, -31498, -31651, -31794, -31928, -32051, -32165, -32270,
    -32364, -32449, -32523, -32588, -32643, -32688, -32723, -32748, -32763, -32767,
    -32763, -32748, -32723, -32688, -32643, -32588, -32523, -32449, -32364, -32270,

```

```

-32165,-32051,-31928,-31794,-31651, -31498,-31336,-31164,-30982,-30791,
-30591,-30382,-30163,-29935,-29698, -29451,-29196,-28932,-28659,-28378,
-28087,-27788,-27481,-27166,-26842, -26510,-26169,-25821,-25465,-25101,
-24730,-24351,-23965,-23571,-23170, -22762,-22347,-21926,-21498,-21063,
-20621,-20174,-19720,-19260,-18795, -18323,-17847,-17364,-16877,-16384,
-15886,-15384,-14876,-14364,-13848, -13328,-12803,-12275,-11743,-11207,
-10668,-10126, -9580, -9032, -8481, -7927, -7371, -6813, -6252, -5690,
-5126, -4561, -3994, -3425, -2856, -2286, -1715, -1144, -572, 0,
572, 1144, 1715, 2286, 2856, 3425, 3993, 4560, 5126, 5690,
6252, 6813, 7371, 7927, 8481, 9032, 9580, 10126, 10668, 11207,
11743, 12275, 12803, 13328, 13848, 14364, 14876, 15383, 15886, 16384,
16876, 17364, 17846, 18323, 18795, 19260, 19720, 20174, 20621, 21063,
21497, 21926, 22347, 22762, 23170, 23571, 23965, 24351, 24730, 25101,
25465, 25821, 26169, 26509, 26842, 27165, 27481, 27788, 28087, 28377,
28659, 28932, 29196, 29451, 29697, 29935, 30163, 30381, 30591, 30791,
30982, 31164, 31336, 31498, 31651, 31794, 31928, 32051, 32165, 32270,
32364, 32449, 32523, 32588, 32643, 32688, 32723, 32748, 32763, 32767,
32763, 32748, 32723, 32688, 32643, 32588, 32523, 32449, 32364, 32270,
32165, 32051, 31928, 31794, 31651, 31498, 31336, 31164, 30982, 30791,
30591, 30382, 30163, 29935, 29697, 29451, 29196, 28932, 28659, 28378,
28087, 27788, 27481, 27166, 26842, 26510, 26169, 25821, 25465, 25101,
24730, 24351, 23965, 23571, 23170, 22762, 22347, 21926, 21497, 21063,
20621, 20174, 19720, 19260, 18795, 18323, 17847, 17364, 16877, 16384,
15886, 15383, 14876, 14364, 13848, 13328, 12803, 12275, 11743, 11207,
10668, 10126, 9580, 9032, 8481, 7927, 7371, 6813, 6252, 5690,
5126, 4560, 3993, 3425, 2856, 2286, 1715, 1144, 572, 0,
-572, -1143, -1715, -2286, -2856, -3425, -3993, -4560, -5126, -5690,
-6252, -6813, -7371, -7927, -8481, -9032, -9580,-10126,-10668,-11207,
-11743,-12275,-12803,-13328,-13848, -14364,-14876,-15383,-15886,-16384,
-16876,-17364,-17846,-18323,-18795, -19260,-19720,-20174,-20621,-21062,
-21497,-21926,-22347,-22762,-23170, -23571,-23965,-24351,-24730,-25101,
-25465,-25821,-26169,-26509,-26841, -27165,-27481,-27788,-28087,-28377,
-28659,-28932,-29196,-29451,-29697, -29935,-30163,-30381,-30591,-30791,
-30982,-31164,-31336,-31498,-31651, -31794,-31928,-32051,-32165,-32270,
-32364,-32449,-32523,-32588,-32643, -32688,-32723,-32748,-32763,-32767,
-32763,-32748,-32723,-32688,-32643, -32588,-32523,-32449,-32364,-32270,
-32165,-32051,-31928,-31794,-31651, -31498,-31336,-31164,-30982,-30791,
-30591,-30382,-30163,-29935,-29698, -29451,-29196,-28932,-28659,-28378,
-28087,-27788,-27481,-27166,-26842, -26510,-26169,-25821,-25465,-25101,
-24730,-24351,-23965,-23571,-23170, -22762,-22347,-21926,-21498,-21063,
-20621,-20174,-19720,-19260,-18795, -18323,-17847,-17364,-16877,-16384,
-15886,-15384,-14876,-14364,-13848, -13328,-12803,-12275,-11743,-11207,
-10668,-10126, -9580, -9032, -8481, -7927, -7371, -6813, -6252, -5690,
-5126, -4561, -3994, -3425, -2856, -2286, -1715, -1144, -572, 0,

```

};

/******

```

/* Variable definition */
/*****/
signed short pwm_a_w; /* Instruction value of A one-phase PWM */
signed short pwm_a_r; /* Instruction value of A two-phase PWM */
signed short pwm_b_w; /* Instruction value of B one-phase PWM */
signed short pwm_b_r; /* Instruction value of B two-phase PWM */

signed short out_bin = 40; /* Temporary value of output frequency */
signed short sinpt_sum; /* Counter for sin pointer total */
signed short sin_pt; /* sin table pointer */
signed short sin_cut; /* Cut value for sin pointer generation */

/*****/
/* Initialization function of motor control part */
/*****/
void init_motor( void )
{
    /* iio initialization */
    g2bcr0 = 0x7f;
    btsr = 0x00; /* all base timer stop */
    g2bcr0 = 0x00; /* group2 clock stop */

    /* iio group2 initialization */
    g2bcr0 = 0x7f;
    /* 01111111B */
    /* ||| | | | +---; (BCK1,0):Count Source Select bit */
    /* ||| | | | ; 11:f1 */
    /* | +---+---; (DIV4-0):Count Source Division Ratio Select bit */
    /* | ; 1111:No Division */
    /* +-----; (IT):Base Timer Interrupt Select bit */
    /* ; 0:Bit 15 overflows */

    g2bcr1 = 0x02;
    /* 0000010B */
    /* ||| | | | +---; (RST0):Base Timer Reset Cause Select Bit 0 */
    /* ||| | | | ; 0:The base timer is not reset by synchronizing */
    /* ||| | | | ; with the group 1 base timer reset */
    /* ||| | | | +---; (RST1):Base Timer Reset Cause Select Bit 1 */
    /* ||| | | | ; 1:The base timer is reset by matching */
    /* ||| | | | ; with the G2PO0 register */
    /* ||| | | +---; (RST2):Base Timer Reset Cause Select Bit 2 */
    /* ||| | | ; 0:The base timer is not reset */
    /* ||| | | ; by applying "L" to the INT1 pin */
    /* ||| | +---; (b3):Reserved Bit. Set to "0". */
    /* || | +---; (BTS):Base Timer Start Bit */
    /* || | ; 0:Base timer is reset */
    /* | +---; (b6,5):Reserved Bit. Set to "0". */

```

```

/* +-----; (PRP):Parallel Real-Time Port Funtion Select bit */
/*          ; 0:RTP output mode */

g2pocr0 = 0x00;
g2po0 = CARR_SET-2; /* Carrier frequency setting */
g2pocr0 = 0x20;
/* 00100000B */
/* |||||+++-; (MOD2,1,0):Operation Mode Select Bit */
/* ||||| ; 000:Single waveform output mode */
/* |||||+-----; (PRT):Parallel Real-time Port Output Trigger */
/* ||||| ; Select bit */
/* ||||| ; 0:Not triggered... */
/* |||+-----; (IVL):Output Initial Value */
/* ||| ; 0:Outputs "L"as initial value */
/* ||+-----; (RLD):G1POj Register Value Reload Timing Select Bit */
/* || ; 1:Reloads the GiPOj register */
/* || ; when the base timer is reset */
/* |+-----; (RTP):Real-time Port Function Select bit */
/* | ; 0:Not used */
/* +-----; (INV):Inverse Output Function Select Bit */
/*          ; 0:Output is not inversed */

g2pocr1 = 0x00;
g2pocr2 = 0x00;
g2pocr3 = 0x00;
g2pocr4 = 0x00;
g2po1 = CARR_SET; /* Cycle of channel 1 */
g2po2 = CARR_SET; /* Cycle of channel 2 */
g2po3 = CARR_SET; /* Cycle of channel 3 */
g2po4 = CARR_SET; /* Cycle of channel 4 */

g2pocr1 = 0x20;
g2pocr2 = 0x20;
g2pocr3 = 0x20;
g2pocr4 = 0x20;

g2fe = 0x1f; /* Enables functions for channel 0-4 */

/* iio group2 interrupt initialization */
iio6ie = 0x01; /* Interrupt request is used for interrupt */
iio6ir = 0x00; /* Interrupt request bit clear */
iio6ie = 0x05; /* Enables an interrupt[PO20] */
iio6ic = 0x03; /* Interrupt priority level setting */

/* Port setting */
ps7 = 0x69; /* iio group2 output */

```

```

/* interrupt enable */
_asm("fset i");

/* PWM output start */
g2bcr1 = 0x12;
/* 00010010B */
/* +-----; (BTS):Base Timer Start Bit */
/* ; 1:Base timer starts counting */

main_pro();
}

/*****
/* Operation example in the main processing */
/* Cut value calculation of sin table */
*****/
void main_pro( void )
{
    /* sin cut value = 23040 * Output frequency / Carrier frequency */
    /* 23040 = 360°* 64 */
    sin_cut=(signed short)((((signed long)out_bin * 23040) / CARR_CLK);
}

/*****
/* Base timer interrupt processing */
*****/
void iio_int( void )
{
    iio6ir=0x00; /* Interrupt request clear */
    angle(); /* Generation of sin table pointer */
    i_con(); /* Calculation of PWM instruction value */
    pwm_set(); /* Upper bound correction of PWM instruction value */
                /* -> Timer setting */
}

/*****
/* Calculation of sin angle */
*****/
void angle( void )
{
    sinpt_sum += sin_cut; /* sin pointer total += sin cut value */
    if(sinpt_sum > 23040) { /* Is it sin pointer total maximum value? */
        /* 23040 = 360°* 64 */
        sinpt_sum -= 23040; /* sin pointer total maximum value correction */
    }
    sin_pt = sinpt_sum >> 6; /* sin pointer <- sin pointer total / 64 */
}

```

```

/*****/
/* Calculation of PWM duty */
/*****/

void i_con( void )
{
    /* PWM instruction value = (sin N° * Carrier) */
    pwm_a_w = (signed short)((((signed long)sin_tbl[sin_pt] * (signed long)(CARR_SET)) >> 15);
    pwm_a_r = (signed short)((((signed long)sin_tbl[sin_pt + 180] * (signed long)(CARR_SET)) >> 15);
    pwm_b_w = (signed short)((((signed long)sin_tbl[sin_pt + 90] * (signed long)(CARR_SET)) >> 15);
    pwm_b_r = (signed short)((((signed long)sin_tbl[sin_pt + 270] * (signed long)(CARR_SET)) >> 15);
}

/*****/
/* Module for base timer interrupt */
/* Setting of PWM data */
/*****/

void pwm_set(void)
{
    /* A-phase PWM */
    /* Is the duty the maximum? */
    if(0 > pwm_a_w) {
        g2po1 = CARR_SET+1; /* Group2 Waveform Generation Register:"L" Output */
    } else {
        /* Is the duty minimum? */
        if(CARR_SET - DTT_SET < pwm_a_w) {
            g2po1 = 0; /* Group2 Waveform Generation Register:"H" Output */
        } else {
            /* minimum < duty < maximum */
            g2po1 = CARR_SET - pwm_a_w;
            /* Group2 Waveform Generation Register */
        }
    }

    /* Is the duty the maximum? */
    if(0 > pwm_a_r) {
        g2po2 = CARR_SET+1; /* Group2 Waveform Generation Register:"L" Output */
    } else {
        /* Is the duty minimum? */
        if(CARR_SET - DTT_SET < pwm_a_r) {
            g2po2 = 0; /* Group2 Waveform Generation Register:"H" Output */
        } else {
            /* minimum < duty < maximum */
            g2po2 = CARR_SET - pwm_a_r;
            /* Group2 Waveform Generation Register */
        }
    }
}

```

```

/* B-phase PWM */
/* Is the duty the maximum? */
if(0 > pwm_b_w) {
    g2po3 = CARR_SET+1; /* Group2 Waveform Generation Register:"L" Output */
} else {
    /* Is the duty minimum? */
    if(CARR_SET - DTT_SET < pwm_b_w) {
        g2po3 = 0; /* Group2 Waveform Generation Register:"H" Output */
    } else {
        /* minimum < duty < maximum */
        g2po3 = CARR_SET - pwm_b_w;
        /* Group2 Waveform Generation Register */
    }
}
/* Is the duty the maximum? */
if(0 > pwm_b_r) {
    g2po4 = CARR_SET+1; /* Group2 Waveform Generation Register:"L" Output */
} else {
    /* Is the duty minimum? */
    if(CARR_SET - DTT_SET < pwm_b_r) {
        g2po4 = 0; /* Group2 Waveform Generation Register:"H" Output */
    } else {
        /* minimum < duty < maximum */
        g2po4 = CARR_SET - pwm_b_r;
        /* Group2 Waveform Generation Register */
    }
}
}
}

```

4.2. M32C/85 用

```

/*****
/* include file */
/*****
#include "sfr32c85.h"

/*****
/* Function prototype declaration */
/*****
void init_motor( void );
void main_pro( void );
#pragma INTERRUPT/B iio_int
void iio_int( void );
void angle( void );
void i_con( void );

```



```

void pwm_set( void );

/*****
/*  Define for sine wave output          */
*****/
#define MCLK      30000000          /* Operation frequency:30MHz */
#define CARR_CLK  20000           /* Carrier frequency:20KHz */
#define C_DIV     1              /* Division rate of base timer */

#define CARR_SET  MCLK / CARR_CLK / C_DIV /* Carrier cycle */
#define DTT_SET   CARR_SET / 20 /* Judgment value of "H" fixed section */

/*****
/*  sin value table                      */
*****/
const short sin_tbl[720]
={
    572, 1144, 1715, 2286, 2856, 3425, 3993, 4560, 5126, 5690,
    6252, 6813, 7371, 7927, 8481, 9032, 9580, 10126, 10668, 11207,
    11743, 12275, 12803, 13328, 13848, 14364, 14876, 15383, 15886, 16384,
    16876, 17364, 17846, 18323, 18795, 19260, 19720, 20174, 20621, 21063,
    21497, 21926, 22347, 22762, 23170, 23571, 23965, 24351, 24730, 25101,
    25465, 25821, 26169, 26509, 26842, 27165, 27481, 27788, 28087, 28377,
    28659, 28932, 29196, 29451, 29697, 29935, 30163, 30381, 30591, 30791,
    30982, 31164, 31336, 31498, 31651, 31794, 31928, 32051, 32165, 32270,
    32364, 32449, 32523, 32588, 32643, 32688, 32723, 32748, 32763, 32767,
    32763, 32748, 32723, 32688, 32643, 32588, 32523, 32449, 32364, 32270,
    32165, 32051, 31928, 31794, 31651, 31498, 31336, 31164, 30982, 30791,
    30591, 30382, 30163, 29935, 29697, 29451, 29196, 28932, 28659, 28378,
    28087, 27788, 27481, 27166, 26842, 26510, 26169, 25821, 25465, 25101,
    24730, 24351, 23965, 23571, 23170, 22762, 22347, 21926, 21497, 21063,
    20621, 20174, 19720, 19260, 18795, 18323, 17847, 17364, 16877, 16384,
    15886, 15383, 14876, 14364, 13848, 13328, 12803, 12275, 11743, 11207,
    10668, 10126, 9580, 9032, 8481, 7927, 7371, 6813, 6252, 5690,
    5126, 4560, 3993, 3425, 2856, 2286, 1715, 1144, 572, 0,
    -572, -1143, -1715, -2286, -2856, -3425, -3993, -4560, -5126, -5690,
    -6252, -6813, -7371, -7927, -8481, -9032, -9580, -10126, -10668, -11207,
    -11743, -12275, -12803, -13328, -13848, -14364, -14876, -15383, -15886, -16384,
    -16876, -17364, -17846, -18323, -18795, -19260, -19720, -20174, -20621, -21062,
    -21497, -21926, -22347, -22762, -23170, -23571, -23965, -24351, -24730, -25101,
    -25465, -25821, -26169, -26509, -26841, -27165, -27481, -27788, -28087, -28377,
    -28659, -28932, -29196, -29451, -29697, -29935, -30163, -30381, -30591, -30791,
    -30982, -31164, -31336, -31498, -31651, -31794, -31928, -32051, -32165, -32270,
    -32364, -32449, -32523, -32588, -32643, -32688, -32723, -32748, -32763, -32767,
    -32763, -32748, -32723, -32688, -32643, -32588, -32523, -32449, -32364, -32270,
    -32165, -32051, -31928, -31794, -31651, -31498, -31336, -31164, -30982, -30791,
    -30591, -30382, -30163, -29935, -29698, -29451, -29196, -28932, -28659, -28378,

```

```
-28087,-27788,-27481,-27166,-26842, -26510,-26169,-25821,-25465,-25101,
-24730,-24351,-23965,-23571,-23170, -22762,-22347,-21926,-21498,-21063,
-20621,-20174,-19720,-19260,-18795, -18323,-17847,-17364,-16877,-16384,
-15886,-15384,-14876,-14364,-13848, -13328,-12803,-12275,-11743,-11207,
-10668,-10126, -9580, -9032, -8481, -7927, -7371, -6813, -6252, -5690,
-5126, -4561, -3994, -3425, -2856, -2286, -1715, -1144, -572, 0,
572, 1144, 1715, 2286, 2856, 3425, 3993, 4560, 5126, 5690,
6252, 6813, 7371, 7927, 8481, 9032, 9580, 10126, 10668, 11207,
11743, 12275, 12803, 13328, 13848, 14364, 14876, 15383, 15886, 16384,
16876, 17364, 17846, 18323, 18795, 19260, 19720, 20174, 20621, 21063,
21497, 21926, 22347, 22762, 23170, 23571, 23965, 24351, 24730, 25101,
25465, 25821, 26169, 26509, 26842, 27165, 27481, 27788, 28087, 28377,
28659, 28932, 29196, 29451, 29697, 29935, 30163, 30381, 30591, 30791,
30982, 31164, 31336, 31498, 31651, 31794, 31928, 32051, 32165, 32270,
32364, 32449, 32523, 32588, 32643, 32688, 32723, 32748, 32763, 32767,
32763, 32748, 32723, 32688, 32643, 32588, 32523, 32449, 32364, 32270,
32165, 32051, 31928, 31794, 31651, 31498, 31336, 31164, 30982, 30791,
30591, 30382, 30163, 29935, 29697, 29451, 29196, 28932, 28659, 28378,
28087, 27788, 27481, 27166, 26842, 26510, 26169, 25821, 25465, 25101,
24730, 24351, 23965, 23571, 23170, 22762, 22347, 21926, 21497, 21063,
20621, 20174, 19720, 19260, 18795, 18323, 17847, 17364, 16877, 16384,
15886, 15383, 14876, 14364, 13848, 13328, 12803, 12275, 11743, 11207,
10668, 10126, 9580, 9032, 8481, 7927, 7371, 6813, 6252, 5690,
5126, 4560, 3993, 3425, 2856, 2286, 1715, 1144, 572, 0,
-572, -1143, -1715, -2286, -2856, -3425, -3993, -4560, -5126, -5690,
-6252, -6813, -7371, -7927, -8481, -9032, -9580,-10126,-10668,-11207,
-11743,-12275,-12803,-13328,-13848, -14364,-14876,-15383,-15886,-16384,
-16876,-17364,-17846,-18323,-18795, -19260,-19720,-20174,-20621,-21062,
-21497,-21926,-22347,-22762,-23170, -23571,-23965,-24351,-24730,-25101,
-25465,-25821,-26169,-26509,-26841, -27165,-27481,-27788,-28087,-28377,
-28659,-28932,-29196,-29451,-29697, -29935,-30163,-30381,-30591,-30791,
-30982,-31164,-31336,-31498,-31651, -31794,-31928,-32051,-32165,-32270,
-32364,-32449,-32523,-32588,-32643, -32688,-32723,-32748,-32763,-32767,
-32763,-32748,-32723,-32688,-32643, -32588,-32523,-32449,-32364,-32270,
-32165,-32051,-31928,-31794,-31651, -31498,-31336,-31164,-30982,-30791,
-30591,-30382,-30163,-29935,-29698, -29451,-29196,-28932,-28659,-28378,
-28087,-27788,-27481,-27166,-26842, -26510,-26169,-25821,-25465,-25101,
-24730,-24351,-23965,-23571,-23170, -22762,-22347,-21926,-21498,-21063,
-20621,-20174,-19720,-19260,-18795, -18323,-17847,-17364,-16877,-16384,
-15886,-15384,-14876,-14364,-13848, -13328,-12803,-12275,-11743,-11207,
-10668,-10126, -9580, -9032, -8481, -7927, -7371, -6813, -6252, -5690,
-5126, -4561, -3994, -3425, -2856, -2286, -1715, -1144, -572, 0,
```

};

```
/*
Variable definition
*/
```

```

signed short pwm_a_w;      /* Instruction value of A one-phase PWM */
signed short pwm_a_r;      /* Instruction value of A two-phase PWM */
signed short pwm_b_w;      /* Instruction value of B one-phase PWM */
signed short pwm_b_r;      /* Instruction value of B two-phase PWM */

signed short out_bin = 40; /* Temporary value of output frequency */
signed short sinpt_sum;    /* Counter for sin pointer total */
signed short sin_pt;       /* sin table pointer */
signed short sin_cut;      /* Cut value for sin pointer generation */

/*****/
/* Initialization function of motor control part */
/*****/
void init_motor( void )
{
    /* iio initialization */
    g1bcr0 = 0x7f;
    /* 01111111B */
    /* ||| ||| +---; (BCK1,0):Count Source Select bit */
    /* ||| ||| ; 11:f1 */
    /* |+++++----; (DIV4-0):Count Source Division Ratio Select bit */
    /* | ; 11111:No Division */
    /* +-----; (IT):Base Timer Interrupt Select bit */
    /* ; 0:Bit 15 overflows */

    g1bcr1 = 0x02;
    /* 00000010B */
    /* ||| ||| +---; (RST0):Base Timer Reset Cause Select Bit 0 */
    /* ||| ||| ; 0:The base timer is not reset by synchronizing */
    /* ||| ||| ; with the group 1 base timer reset */
    /* ||| ||| +---; (RST1):Base Timer Reset Cause Select Bit 1 */
    /* ||| ||| ; 1:The base timer is reset by matching */
    /* ||| ||| ; with the G2PO0 register */
    /* ||| ||| +-----; (RST2):Base Timer Reset Cause Select Bit 2 */
    /* ||| ||| ; 0:The base timer is not reset */
    /* ||| ||| ; by applying "L" to the INT1 pin */
    /* ||| +-----; (b3):Reserved Bit. Set to "0". */
    /* ||| +-----; (BTS):Base Timer Start Bit */
    /* ||| ; 0:Base timer is reset */
    /* |+++++----; (b6,5):Reserved Bit. Set to "0". */
    /* +-----; (PRP):Parallel Real-Time Port Funtion Select bit */
    /* ; 0:RTP output mode */

    g1pocr0 = 0x00;
    g1po0 = CARR_SET-2; /* Carrier frequency setting */
    g1pocr0 = 0x20;
    /* 00100000B */
}

```

```

/* | | | | | ++---; (MOD2,1,0):Operation Mode Select Bit */
/* | | | | | ; 000:Single waveform output mode */
/* | | | | | +-----; (PRT):Parallel Real-time Port Output Trigger */
/* | | | | | ; Select bit */
/* | | | | | ; 0:Not triggered... */
/* | | | | | +-----; (IVL):Output Initial Value */
/* | | | | | ; 0:Outputs "L"as initial value */
/* | | | | | +-----; (RLD):GiPOj Register Value Reload Timing Select Bit */
/* | | | | | ; 1:Reloads the GiPOj register */
/* | | | | | ; when the base timer is reset */
/* | | | | | +-----; (RTP):Real-time Port Function Select bit */
/* | | | | | ; 0:Not used */
/* | | | | | +-----; (INV):Inverse Output Function Select Bit */
/* | | | | | ; 0:Output is not inverted */

```

```

g1pocr4 = 0x00;
g1pocr5 = 0x00;
g1pocr6 = 0x00;
g1pocr7 = 0x00;
g1po4 = CARR_SET; /* Cycle of channel 1 */
g1po5 = CARR_SET; /* Cycle of channel 2 */
g1po6 = CARR_SET; /* Cycle of channel 3 */
g1po7 = CARR_SET; /* Cycle of channel 4 */

```

```

g1pocr4 = 0x20;
g1pocr5 = 0x20;
g1pocr6 = 0x20;
g1pocr7 = 0x20;

```

```

g1fe = 0xf1; /* Enables functions for channel 0-4 */

```

```

/* iio group2 interrupt initial set */
iio3ie = 0x01; /* Interrupt request is used for interrupt */
iio3ir = 0x00; /* Interrupt request bit clear */
iio3ie = 0x05; /* Enables an interrupt[PO20] */
iio3ic = 0x03; /* Interrupt priority level setting */

```

```

/* Port setting */
ps8 = 0x0f; /* iio output */

```

```

/* interrupt enable */
_asm("fset i");

```

```

/* PWM output start */
g1bcr1 = 0x12;
/* 00010010B */
/* +-----; (BTS):Base Timer Start Bit */

```

```

/*          ; 1:Base timer starts counting */

main_pro();
}

/*****/
/* Operation example in the main processing */
/* Cut value calculation of sin table */
/*****/
void main_pro( void )
{
    /* sin cut value = 23040 * Output frequency / Carrier frequency */
    /* 23040 = 360°* 64 */
    sin_cut=(signed short)((((signed long)out_bin * 23040) / CARR_CLK);
}

/*****/
/* Base timer interrupt processing */
/*****/
void iio_int( void )
{
    iio3ir=0x00;    /* Interrupt request clear */
    angle();       /* Generation of sin table pointer */
    i_con();       /* Calculation of PWM instruction value */
    pwm_set();     /* Upper bound correction of PWM instruction value */
}                /* -> Timer setting */

/*****/
/* Calculation of sin angle */
/*****/
void angle( void )
{
    sinpt_sum += sin_cut; /* sin pointer total += sin cut value */
    if(sinpt_sum > 23040) { /* Is it sin pointer total maximum value? */
        /* 23040 = 360°* 64 */
        sinpt_sum -= 23040; /* sin pointer total maximum value correction */
    }
    sin_pt = sinpt_sum >> 6; /* sin pointer <- sin pointer total / 64 */
}

/*****/
/* Calculation of PWM duty */
/*****/
void i_con( void )
{
    /* PWM instruction value = (sin N° * Carrier) */
    pwm_a_w = (signed short)((((signed long)sin_tbl[sin_pt] * (signed long)(CARR_SET)) >> 15);
}

```

```
pwm_a_r = (signed short)((((signed long)sin_tbl[sin_pt + 180] * (signed long)(CARR_SET)) >> 15);
pwm_b_w = (signed short)((((signed long)sin_tbl[sin_pt + 90] * (signed long)(CARR_SET)) >> 15);
pwm_b_r = (signed short)((((signed long)sin_tbl[sin_pt + 270] * (signed long)(CARR_SET)) >> 15);
}
```

```

/*****/
/* Module for base timer interrupt */
/* Setting of PWM data */
/*****/
void pwm_set(void)
{
    /* A-phase PWM */
    /* Is the duty the maximum? */
    if(0 > pwm_a_w) {
        g1po4 = CARR_SET+1; /* Waveform Generation Register:"L" Output */
    } else {
        /* Is the duty minimum? */
        if(CARR_SET - DTT_SET < pwm_a_w) {
            g1po4 = 0; /* Waveform Generation Register:"H" Output */
        } else {
            /* minimum < duty < maximum */
            g1po4 = CARR_SET - pwm_a_w;
            /* Waveform Generation Register */
        }
    }
}

/* Is the duty the maximum? */
if(0 > pwm_a_r) {
    g1po5 = CARR_SET+1; /* Waveform Generation Register:"L" Output */
} else {
    /* Is the duty minimum? */
    if(CARR_SET - DTT_SET < pwm_a_r) {
        g1po5 = 0; /* Waveform Generation Register:"H" Output */
    } else {
        /* minimum < duty < maximum */
        g1po5 = CARR_SET - pwm_a_r;
        /* Waveform Generation Register */
    }
}

/* B-phase PWM */
/* Is the duty the maximum? */
if(0 > pwm_b_w) {
    g1po6 = CARR_SET+1; /* Waveform Generation Register:"L" Output */
} else {
    /* Is the duty minimum? */
    if(CARR_SET - DTT_SET < pwm_b_w) {

```

```

        g1po6 = 0;      /* Waveform Generation Register:"H" Output */
    } else {
        /* minimum < duty < maximum */
        g1po6 = CARR_SET - pwm_b_w;
            /* Waveform Generation Register */
    }
}
/* Is the duty the maximum? */
if(0 > pwm_b_r) {
    g1po7 = CARR_SET+1; /* Waveform Generation Register:"L" Output */
} else {
    /* Is the duty minimum? */
    if(CARR_SET - DTT_SET < pwm_b_r) {
        g1po7 = 0;      /* Waveform Generation Register:"H" Output */
    } else {
        /* minimum < duty < maximum */
        g1po7 = CARR_SET - pwm_b_r;
            /* Waveform Generation Register */
    }
}
}

```

5. 参考ドキュメント

ハードウェアマニュアル

M32C/83 グループハードウェアマニュアル Rev.1.0

M32C/85 グループハードウェアマニュアル Rev.1.0

(最新版をルネサス テクノロジホームページから入手してください。)

テクニカルアップデート/テクニカルニュース

(最新の情報をルネサス テクノロジホームページから入手してください。)

6. 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://www.renesas.com/jpn/>

M16C ファミリ MCU 技術サポート窓口

E-mail: csc@renesas.com

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2005.03.31	-	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。