

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。

M16C/26A、M16C/29 グループ 三相モータ制御プログラムの考え方

1.0 要約

この資料では、三相モータ制御用タイマ機能の使用方法を紹介し、応用例を掲載しています。

2.0 はじめに

この資料で説明する応用例は M16C/26A、M16C/29 グループのマイコンでの利用に適用されます。

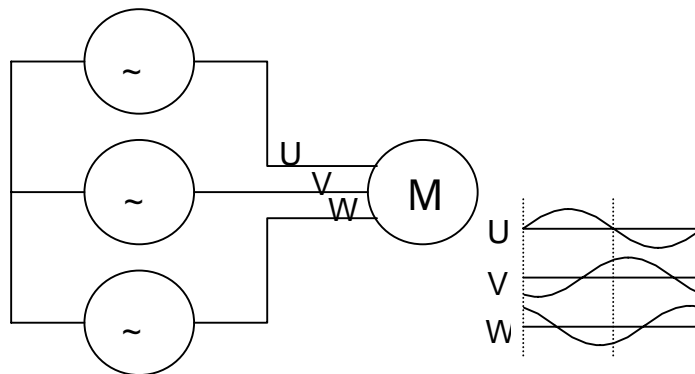
3.0 インバータ制御の概要

3.1 インバータ制御とは

インバータ制御とは周波数を任意に変化させてモータを駆動させる制御です。

例えば、三相モータは120°位相のずれた波形を与えることによって駆動するモータですが、商用電源を三相与えても周波数は商用電源の周波数に依存してしまいます。

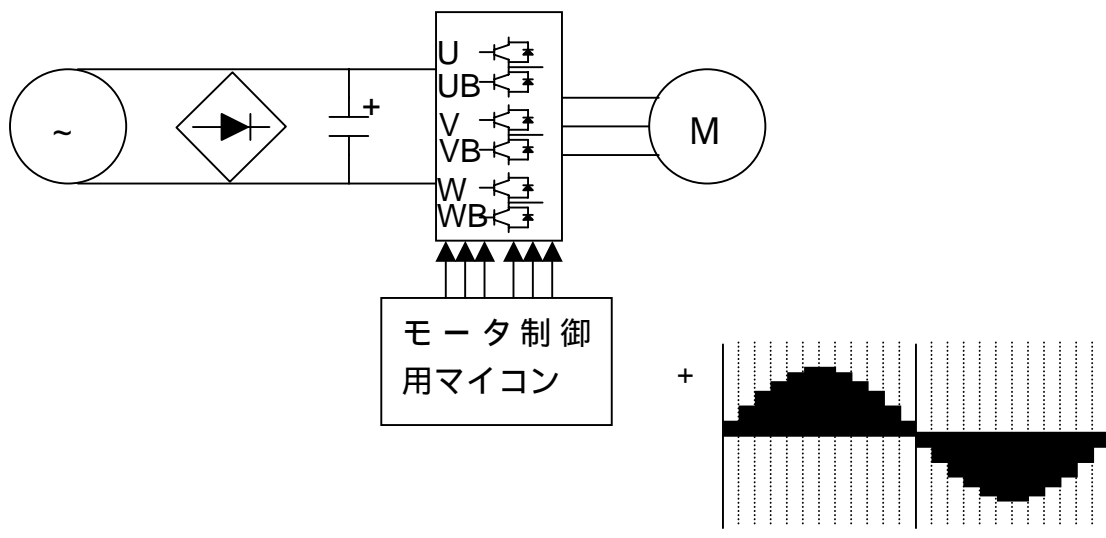
図3.1 三相モータの駆動



インバータ制御はその商用交流電源を一端直流電源に変換し、トランジスタのスイッチングによりモータ駆動用の周波数を作り出します。

このトランジスタのスイッチングはマイコンが制御するので、スイッチング間隔を変えることで任意のモータ駆動周波数を作り出すことができます。

図3.2 マイコンを使用したインバータ制御例



3.2 マイコンから出力される波形

マイコンのポートから交流波形を出力したり、モータ駆動用の高電圧を出力することはできないため、マイコンとモータ間に概念的に図3.3のようなパワートランジスタ回路が必要となります。図中のU、V、W、UB、VB、WBはマイコンの端子が出力する信号が入力されます。

図3.3 パワートランジスタ回路

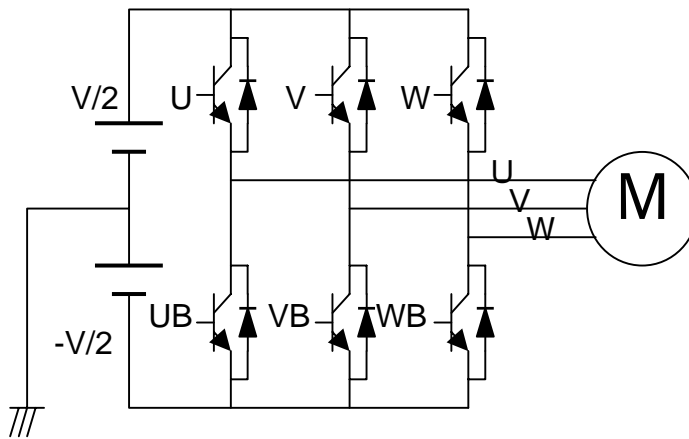
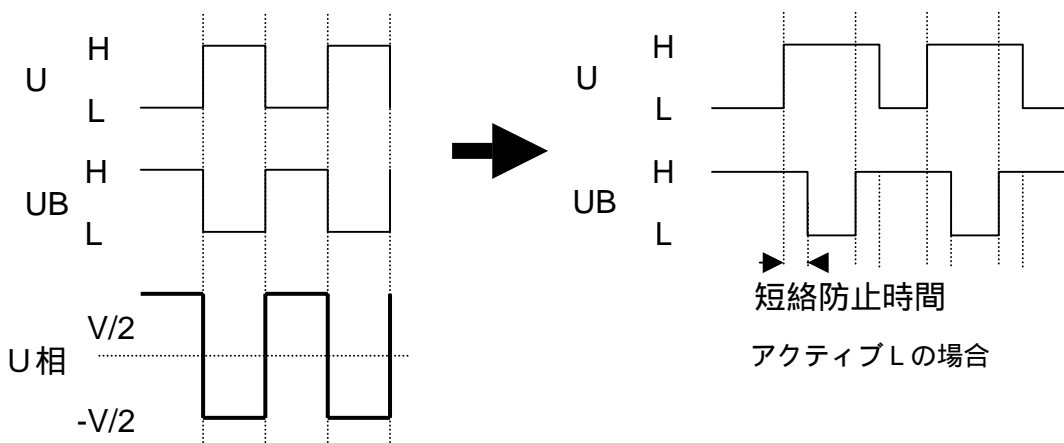


図3.3のU相だけを取り出した動作を解説しますと、U、UBに図3.4のようなON、OFF信号を交互に与えたとき、電圧レベルも同様に反転し、U相には交流（方形）波形が発生します。

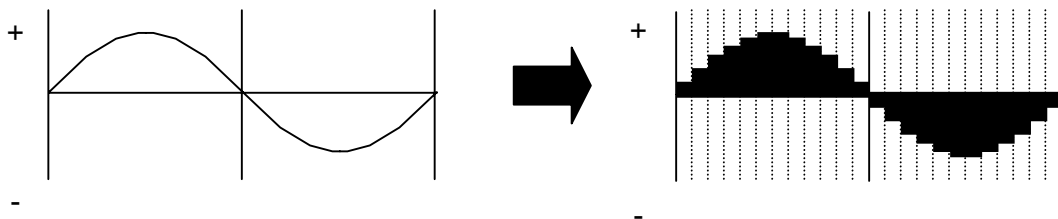
図3.4 マイコンの出力波形と生成される波形



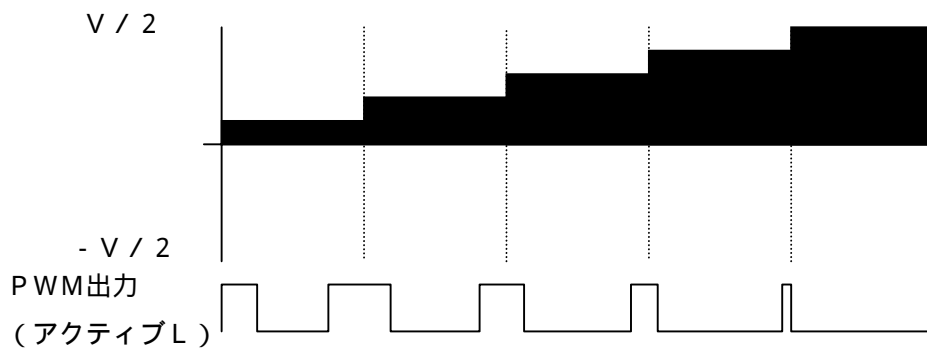
ここで注意することは正逆相が同時にONした場合、貫通電流が流れ、直流電源短絡を引き起こしてしまうことです。三相モータ制御用タイマ機能ではこの短絡を避けるため、切り替え時間差を作り、同時にONすることを防ぐ機能を持っています。この時間差を 短絡防止時間 と呼び、プログラム上の初期設定時に短絡防止タイマに値を設定するだけで、短絡防止時間を負荷した波形が出力されます。

同一時間内に同一面積分の電圧をモータに与えれば近似的に交流の正弦波形を与えたこととなりますので、マイコンからのHi/Low出力幅を変えていくことによって、交流波形出力を実現できます。

図3.5 交流の正弦波形を時分割して方形波に置き換える



分割が細かい程、正弦波形の電圧に近似します。



4.0 三相モータ制御用タイマ機能の使い方

4.1 三相波形を出力するには

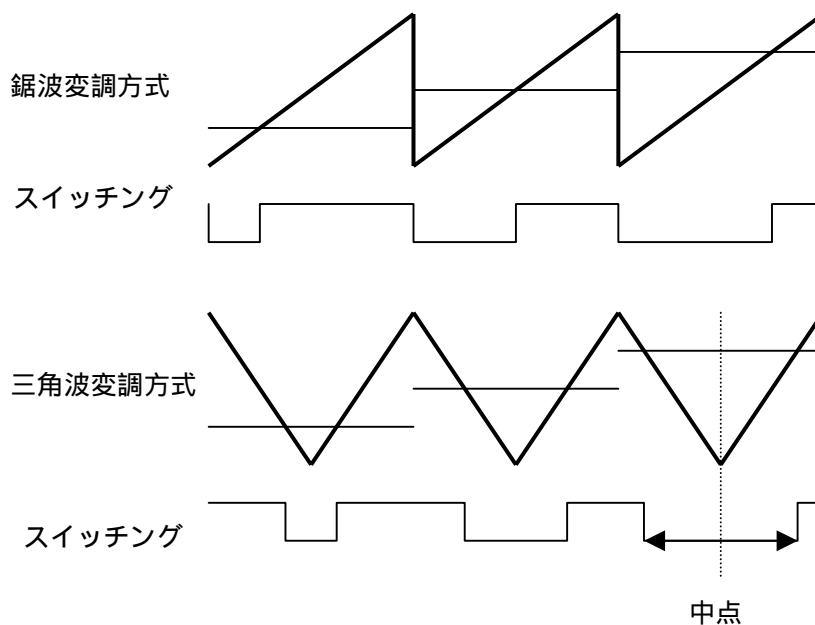
本章では、三相波形を出力する際の基本的な方法について解説します。

4.2 キャリア周波数

トランジスタをスイッチングする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に正弦波をのせたときその交点がスイッチング波形のレベル反転位置になります。

キャリア周波数には 鋸波変調方式 と 三角波変調方式 があります。

図4.1 キャリア周波数の変調方式



鋸波変調方式はキャリア周期の始まりを基準にデューティを可変させます。それに対して三角波変調は中点を基準に左右にデューティを可変させます。

4.3 PWMでの表現方法

三相モータ制御用タイマ機能は、鋸波変調なら1キャリア周期を、三角波変調なら1/2キャリア周期を基準に考えます。

その際、キャリア周期頂点のタイミングはTB2が生成します。このTB2のアンダフローのトリガでTAi (i = 4, 1, 2) のワンショットが起動します。このTAiがPWMデューティを決定します。

鋸波変調にする場合、短絡防止タイマのトリガをTAiの立ち上がり+立ち下がりに設定し、三角波変調にする場合は、立ち下がりのみに設定してください。

図4.2 鋸波変調時のタイマとPWMの関係

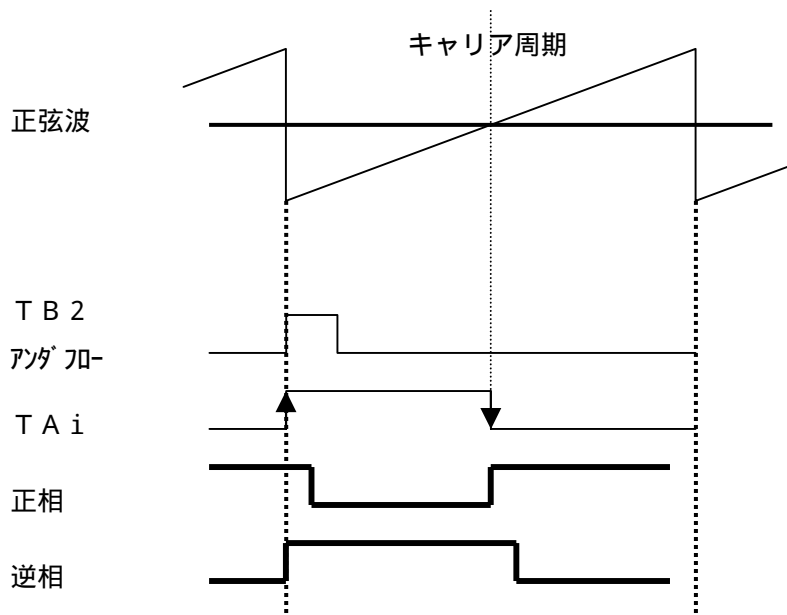
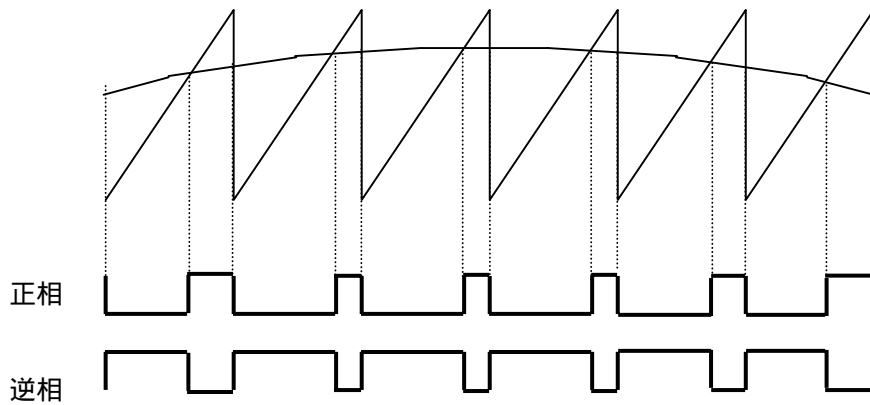
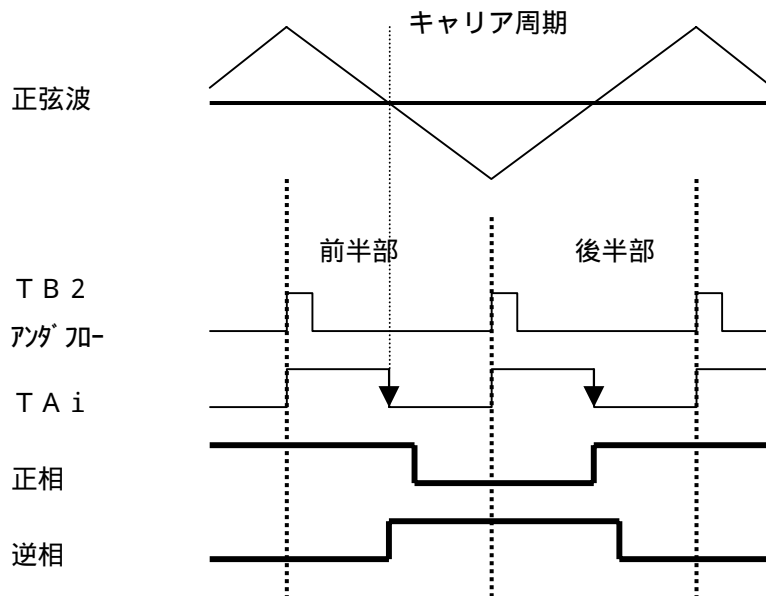
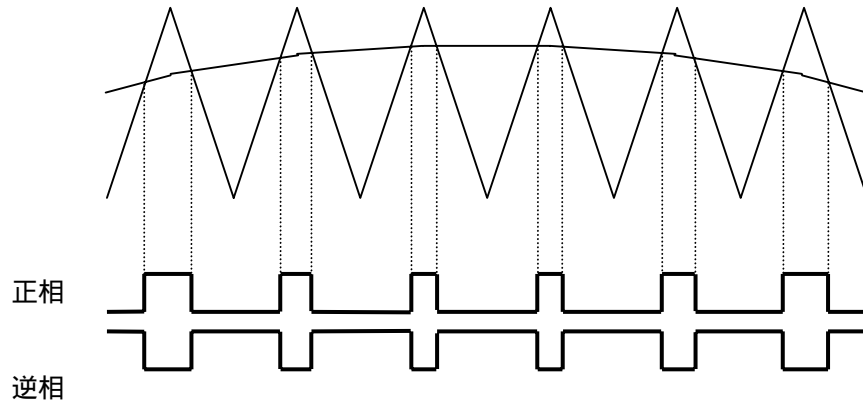


図 4 . 3 三角波変調時のタイマと PWM の関係



基本的には前半部と後半部で対照と考えます

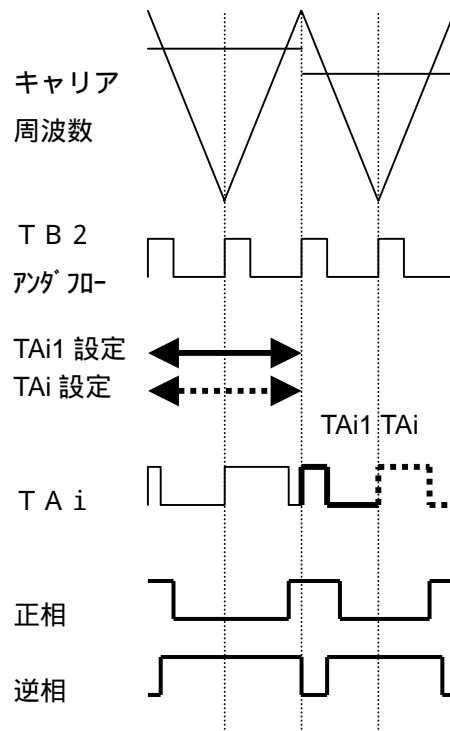
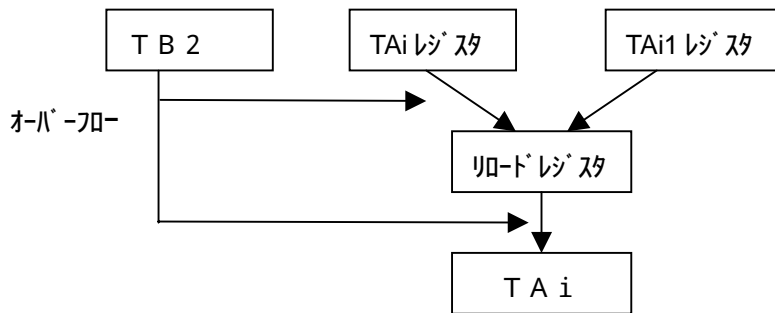
(実際は正弦波形の弧の一部なので、傾きがあり対照ではありません。)

前述の方法で三角波変調で波形を出力させようとする¹と1 / 2 キャリア周期毎に割り込み等でタイミングを生成し、T A i を再設定しなければなりません。

このプログラム（割り込み）負荷を軽減するために、1 キャリア周期に前半部と後半部のT A i 値を設定する機能があります。これが三相モード1です。

三相モード1はT A i に設定する値を2つのレジスタから交互に設定するモードです。

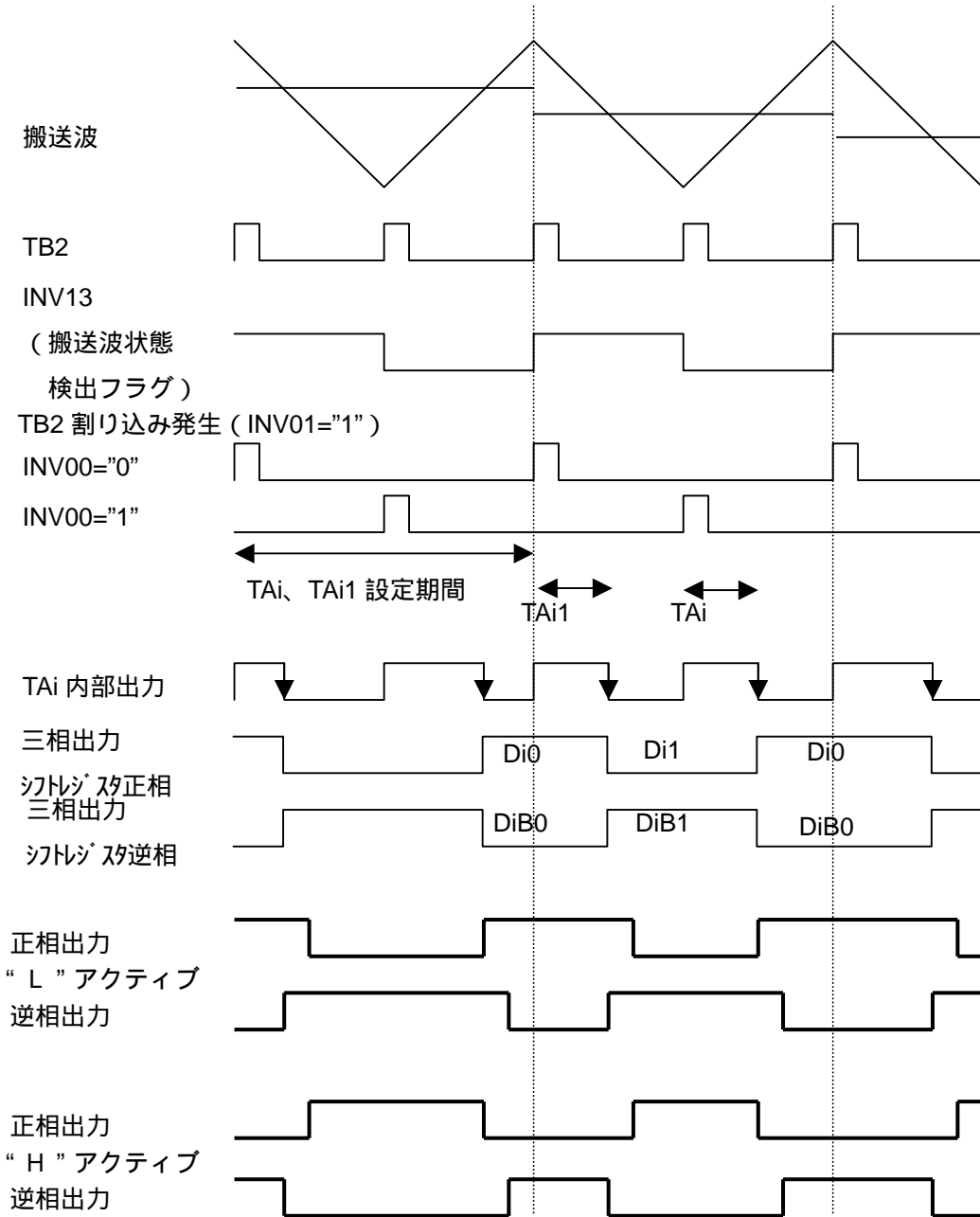
図4.4 三相モード1のTB2とTAi、TAi1レジスタの関係



TB2オーバーフロー付近でのTAi1レジスタ書き換えはできません。詳細はテクニカルニュースをご参照ください。

図 4 . 5 三相モード 1 の設定と出力の関係

(invc0=*0**11**B; invc1=01***1*B; idb0=00010101B; idb1=00101010B)



割り込み有効出力指定を初期値 (INV01="0") かつ割り込み頻度設定カウンタ (ICTB2) に " 2 " を設定した場合、初期状態では図中の INV00="1" と同様のタイミングで TB2 割り込みが発生します。INV00="0" と同様のタイミングが必要な場合は、割り込み頻度設定カウンタ (ICTB2) に初回のみ " 奇数 " を設定することで対応してください。

4.4 キャリア周波数の可変

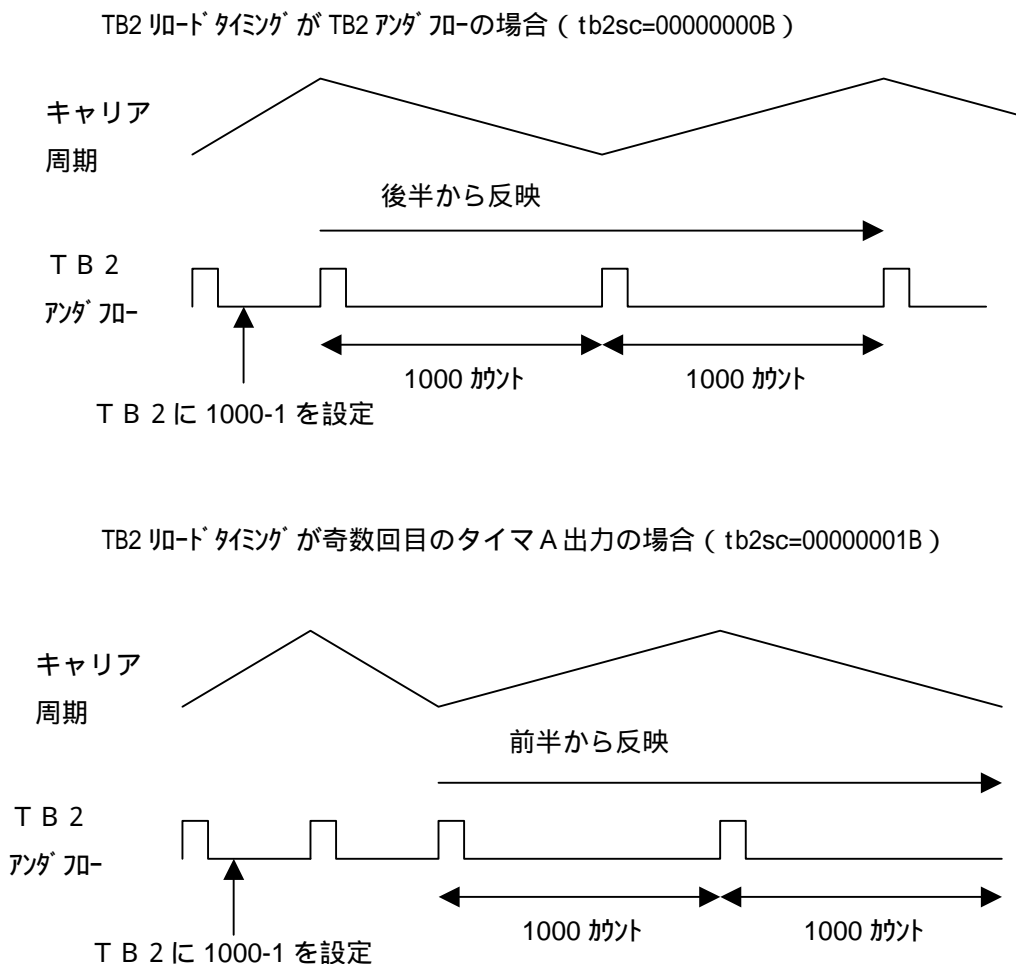
キャリア周波数を可変するには、TB2 割り込みの周期を変更します。ただし、三角波変調の場合、TB2 のアンダフロー 2 回でキャリア 1 周期となるので、注意が必要です。

三角波変調で 1 キャリア周期毎に制御する場合 (invc0=0*****B かつ invc0=0*****1*B) は TB2 リードタイミングを奇数回目のタイマ A 出力 (tb2sc=*****1B) に

三角波変調で 1 / 2 キャリア周期毎 (invc0=0*****B かつ invc0=1*****1*B もしくは invc0=*****B かつ invc0=*****0*B) は TB2 リードタイミングを TB2 アンダフロー (tb2sc=00000000B) に

設定すれば、キャリア周期の可変と T A i との設定のタイミングが一致します。

図 4 . 6 キャリア周期の設定と反映



4.5 三相出力バッファレジスタの使い方

三相出力バッファレジスタの値を書き換えることによって一定期間出力レベルを固定(連続 ON、連続 OFF)したり、正逆相とも同じレベルの波形を出力したりすることができます。

この機能を使用することによって二相変調出力(5章)や120°通電出力(6章)が可能となります。

三相出力バッファレジスタの内容は書き換え後、最初のTB2のアンダフローでシフトレジスタに転送されます。

図4.7 二相変調出力のイメージ波形(制御内容は5章)

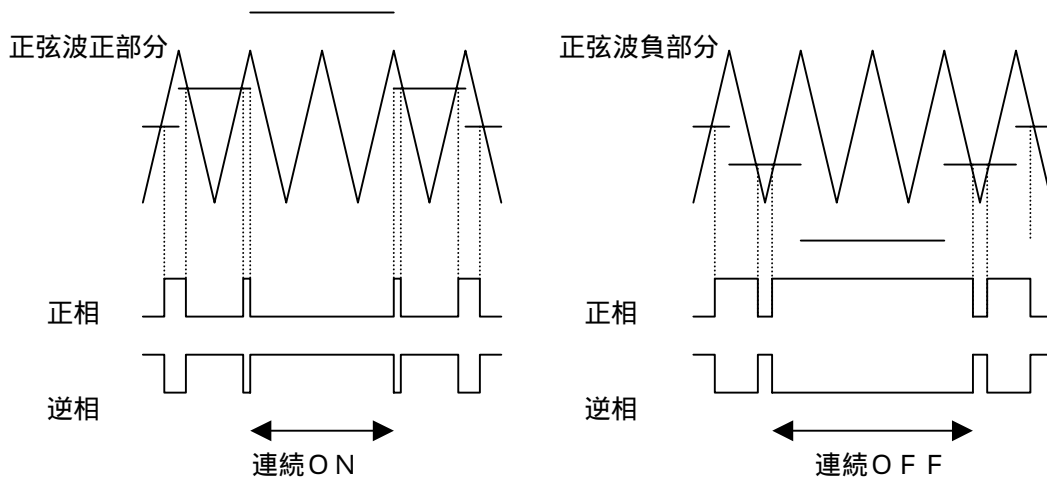
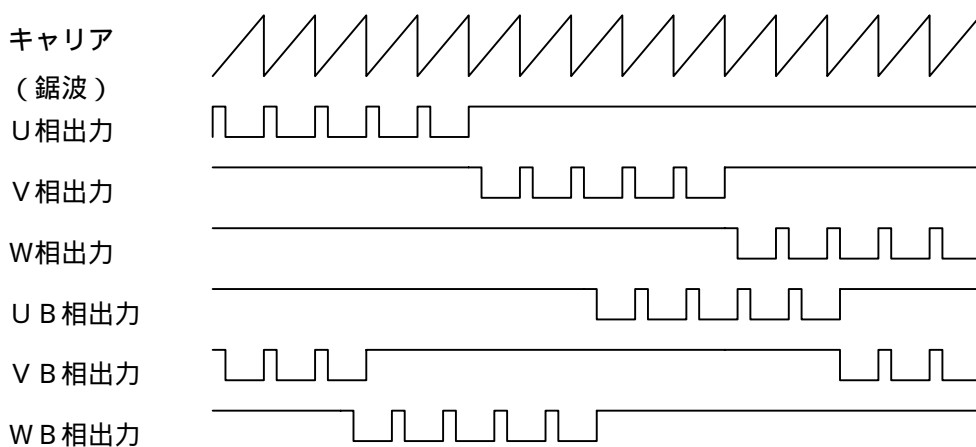


図4.8 120°通電出力のイメージ波形(制御内容は6章)



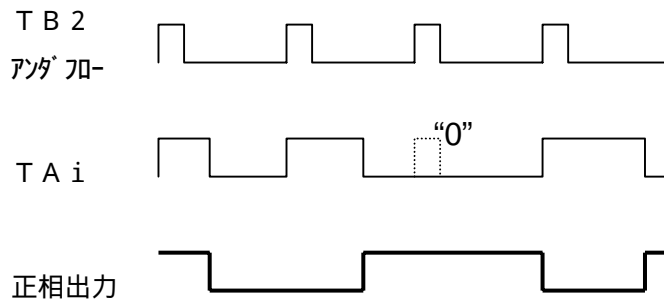
4.4 使用上の注意事項

三相モータ制御用タイマ機能を使用するにあたって以下の点にご注意ください。

(1) T A i に設定するデータの下限值

T A i に “ 0 ” (短絡防止タイマカウンツsf 2 の場合、“ 0 ”、“ 1 ”) を設定すると、T A i タイマはカウントを行わず、立ち下がりエッジが発生しないため、出力波形は現状のレベルのまま保持されます。意図的な場合を除いて設定しないでください。

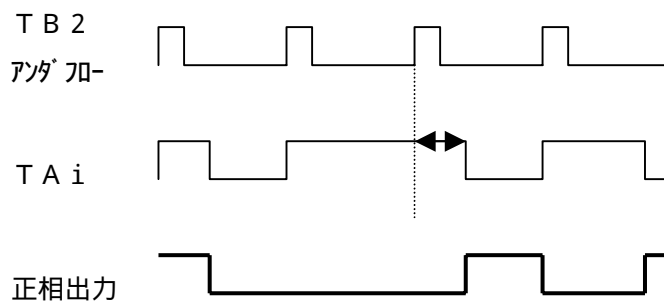
図 4 . 9 T A i = 0 設定時の正相波形



(2) T A i に設定するデータの上限值

T A i に “ T B 2 設定値より大きな値 ” (短絡防止タイマカウンツsf 2 の場合、“ T B 2 設定値- 1 より大きな値 ”) を設定すると、T A i タイマはT B 2 周期内でカウントをしたままになり、立ち下がりエッジが発生しないため、出力波形は現状のレベルのまま保持されます。その結果、下図の様に波形が反転するため、意図的な場合を除いて設定しないでください。

図 4 . 1 0 T A i > T B 2 設定時の出力波形



(3) 短絡防止タイマカウント中の短絡防止タイマの再起動

設定される $T A i$ データによって、短絡防止タイマカウント中に短絡防止タイマの起動要因が発生した場合、短絡防止タイマが再起動されません。

三角波変調モードの場合：短絡防止タイマカウントソース f_1

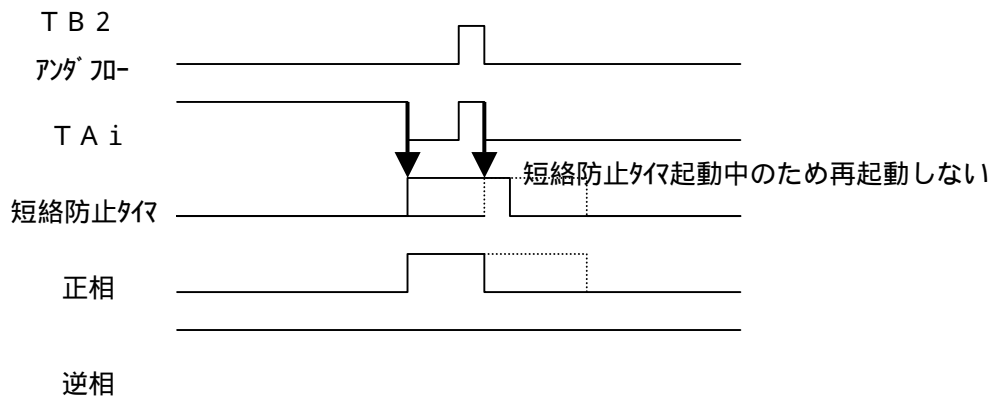
$$\begin{aligned} ((T B 2 \text{ 設定値} + 1) - T A i 1 \text{ 設定値}) + T A i \text{ 設定値} < \text{短絡防止タイマ設定値} \\ T A i 1 \text{ 設定値} + ((T B 2 \text{ 設定値} + 1) - T A i \text{ 設定値}) < \text{短絡防止タイマ設定値} \end{aligned}$$

鋸波変調モードの場合：短絡防止タイマカウントソース f_1

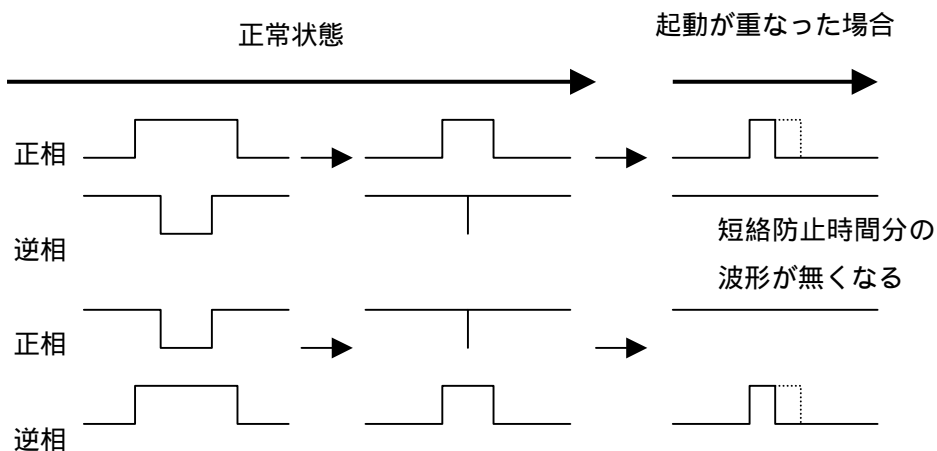
$$\begin{aligned} ((T B 2 \text{ 設定値} + 1) - T A i \text{ 設定値}) - 1 < \text{短絡防止タイマ設定値} \\ T A i \text{ 設定値} - 1 < \text{短絡防止タイマ設定値} \end{aligned}$$

の条件となる場合、短絡防止タイマの再起動が行われないので注意が必要です。

図 4.11 短絡防止タイマの再起動



PWMデューティの変化と短絡防止時間

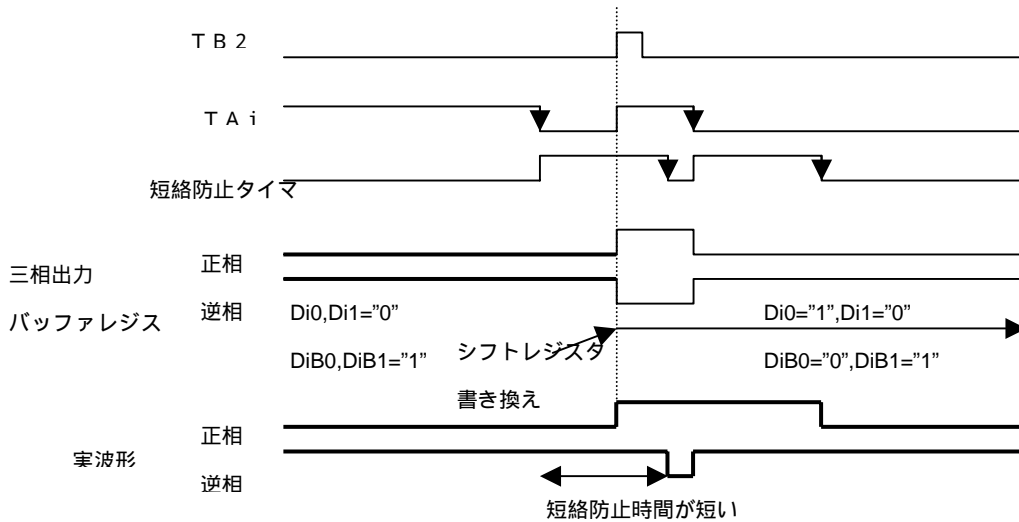


- (4) 三相出力バッファレジスタ書き換え時の出力波形
 短絡防止時間タイマ設定トリガ選択ビットを“0”(invc1=000****B)の場合において、
 三相出力バッファレジスタで固定出力からPWM出力に変更する際、

正相をL固定からPWM出力変更し、 $T A i$ 、 $T A i 1 > T B 2$ - 短絡防止時間

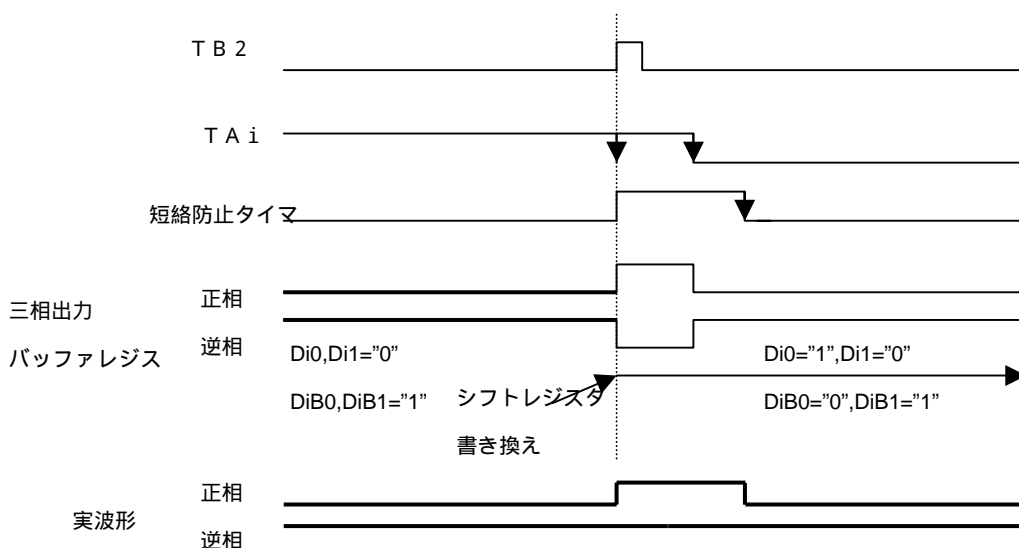
となると出力される短絡防止時間が短くなる場合があります。

図4.12 例 短絡防止時間が短くなる波形



上記において、三相出力バッファレジスタで固定出力からPWM出力に変更する際、 $T A i$ 、 $T a i 1$ に設定するデータは、設定できる最大値(短絡防止タイマカウンツf1の場合、 $T B 2$ 設定値f2の場合、 $T B 2$ 設定値-1)を設定してください。

図4.13 $T A i$ 、 $T A i 1$ 設定値を最大値にした場合



5.0 三相正弦波出力の実現方法

5.1 三相波形を出力するには

本章では、三相モータ制御用タイマ機能を使用した三相正弦波出力の方法を紹介します

5.2 三相モータ制御用タイマ機能での出力方法

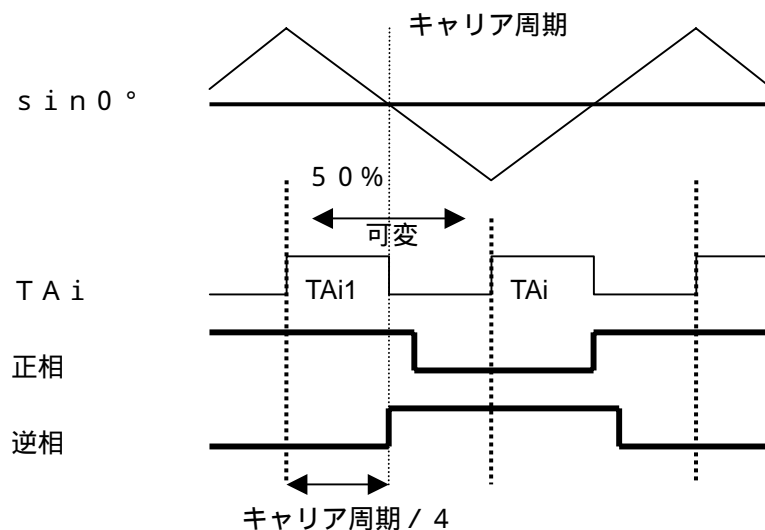
三相モータ制御用タイマ機能の三角波変調かつ三相モード1を選択します。

5.3 PWMデータの算出

以下にTAiに設定する値の算出方法例を解説します。

出力波形は $\sin 0^\circ$ をデューティ50%を基準とし、TAiに設定する値はこれを基準に+・すればよいこととなります。

図5.1 TAiとPWMデータの関係

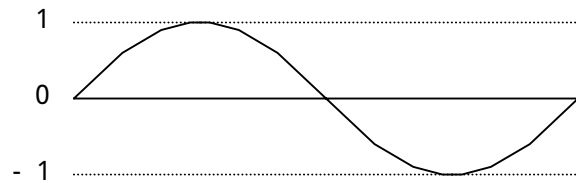


デューティ50% = キャリア周期 / 4 ですから

TAi1データ = キャリア周期 / 4 - 可変値 となります。

ところで正弦波は -1 ~ 0 ~ 1の値となるので、

図5.2 正弦波



デューティが - 50 ~ 0 ~ 50 % となる可変値を求めるには、

$$\begin{aligned} \text{可変値} &= 50\% \times \sin N^\circ \\ &= \text{キャリア周期} / 4 \times \sin N^\circ \end{aligned}$$

となります。
よって、幅設定値は

$$\begin{aligned} \text{TAi1データ} &= \text{キャリア周期} / 4 - \text{キャリア周期} / 4 \times \sin N^\circ \\ \text{TAiデータ} &= \text{キャリア周期} / 2 - \text{TAi1データ} \end{aligned}$$

で求められます。

ベクトル演算等で求めた電流指令値を使用する場合は、

$$\begin{aligned} \text{TAi1データ} &= \text{キャリア周期} / 4 - \text{算出した電流指令値} \times \text{定数} \\ \text{TAiデータ} &= \text{キャリア周期} / 2 - \text{TAi1データ} \end{aligned}$$

となります。

5.4 負荷に対する出力波形の可変

PWMへの負荷情報は、各相に対して出力するPWM波形に変調率を乗算する形で反映します。

$$T_{Ai1} \text{データ} = \text{キャリア周期} / 4 - \text{キャリア周期} / 4 \times \sin N^\circ \times \text{変調率}$$

例としてV/F制御を行うならば、出力周波数の可変に合わせて変調率の値を可変させ、PWM波形長(V)と出力波形(F)の関係を制御します。

図5.3 V/F制御

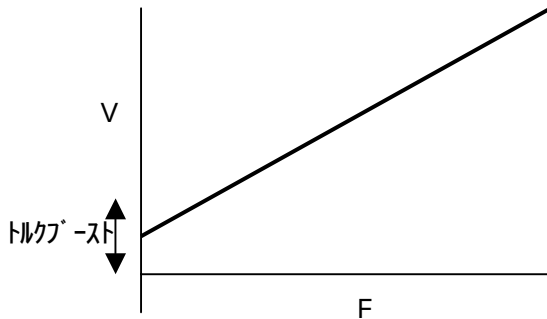
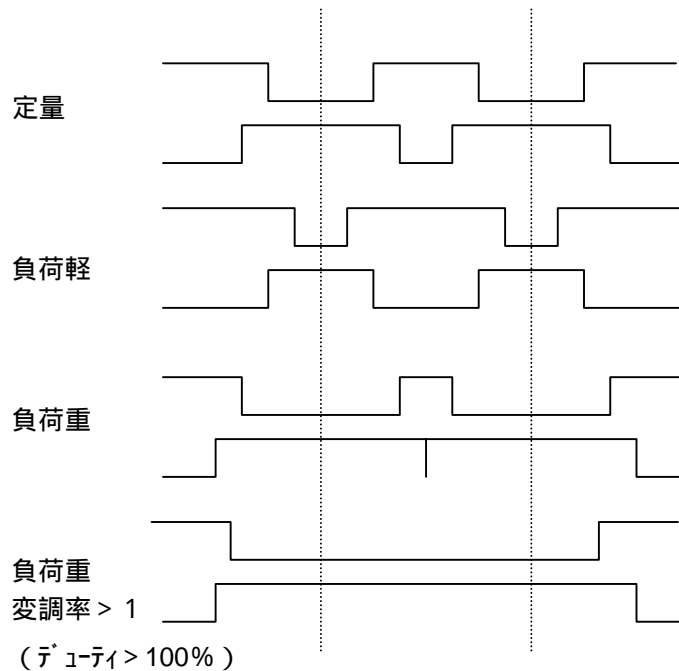


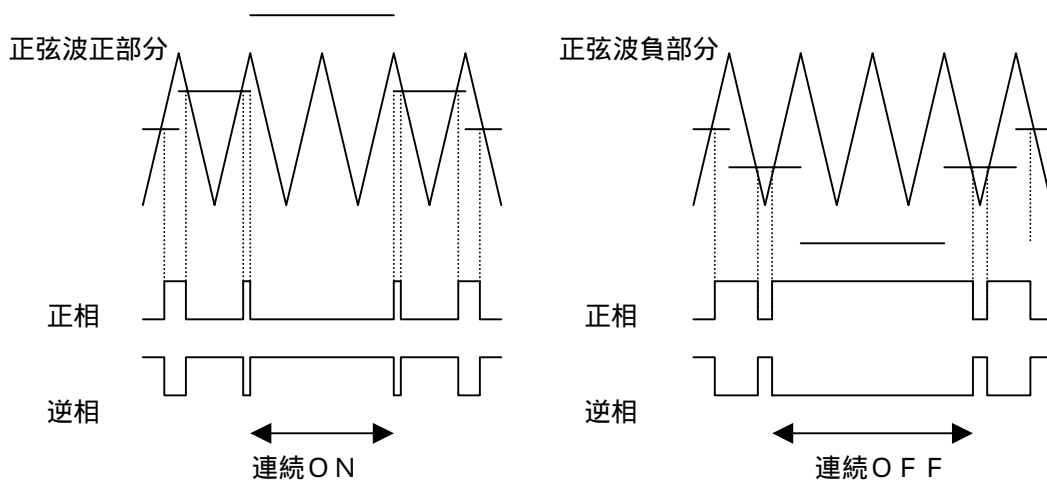
図5.4 変調率と出力波形の関係



5.5 二相変調出力（連続ON、連続OFF）

PWMデューティが一定の幅より広がったり、狭まったりした際、短絡防止時間分の損失をなくすためキャリア周期より長いON/OFFをする制御があります。

図5.5 二相変調出力のイメージ波形

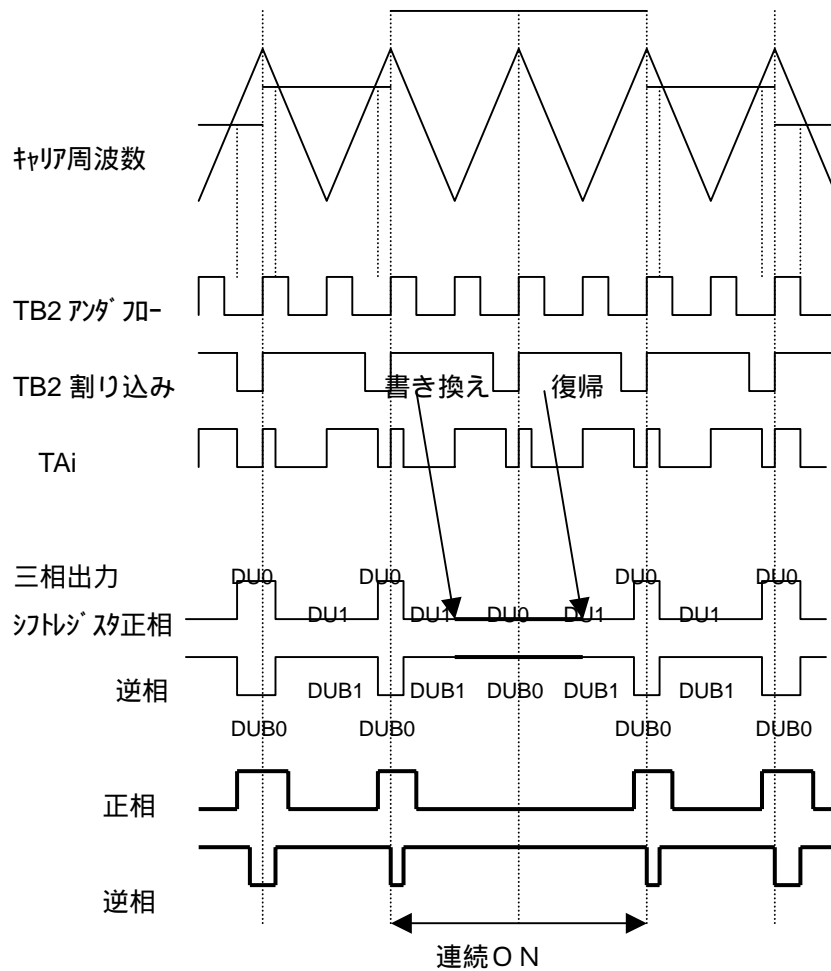


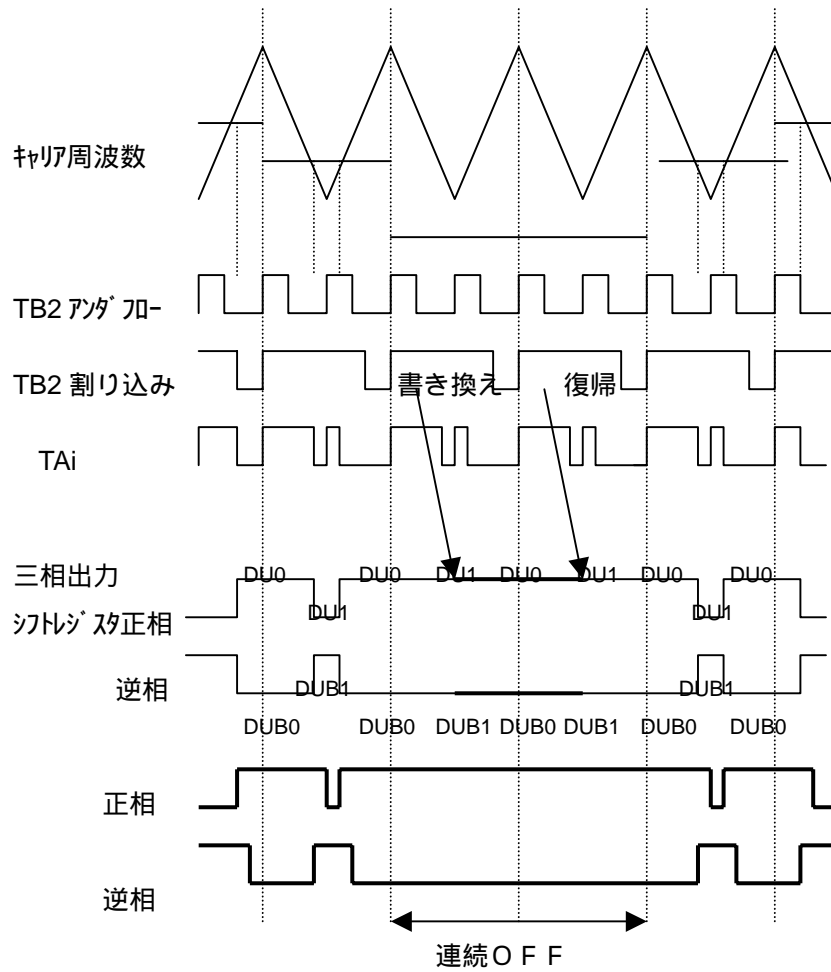
5.5.1 三相出力バッファレジスタの書き換えで連続ON / OFF 波形出力を実現する

下図のように三相出力バッファレジスタの内容をタイミングよく書き換えることによって連続ON / OFF 波形出力を実現します。

この方法は三相出力バッファレジスタを直接書き換えるため、書き換えるビットを間違えると短絡波形を出力する可能性がありますので、正逆相同時アクティブ出力禁止ビット (invc0 の4ビット) を必ず“1”にして、短絡防止を行ってください。

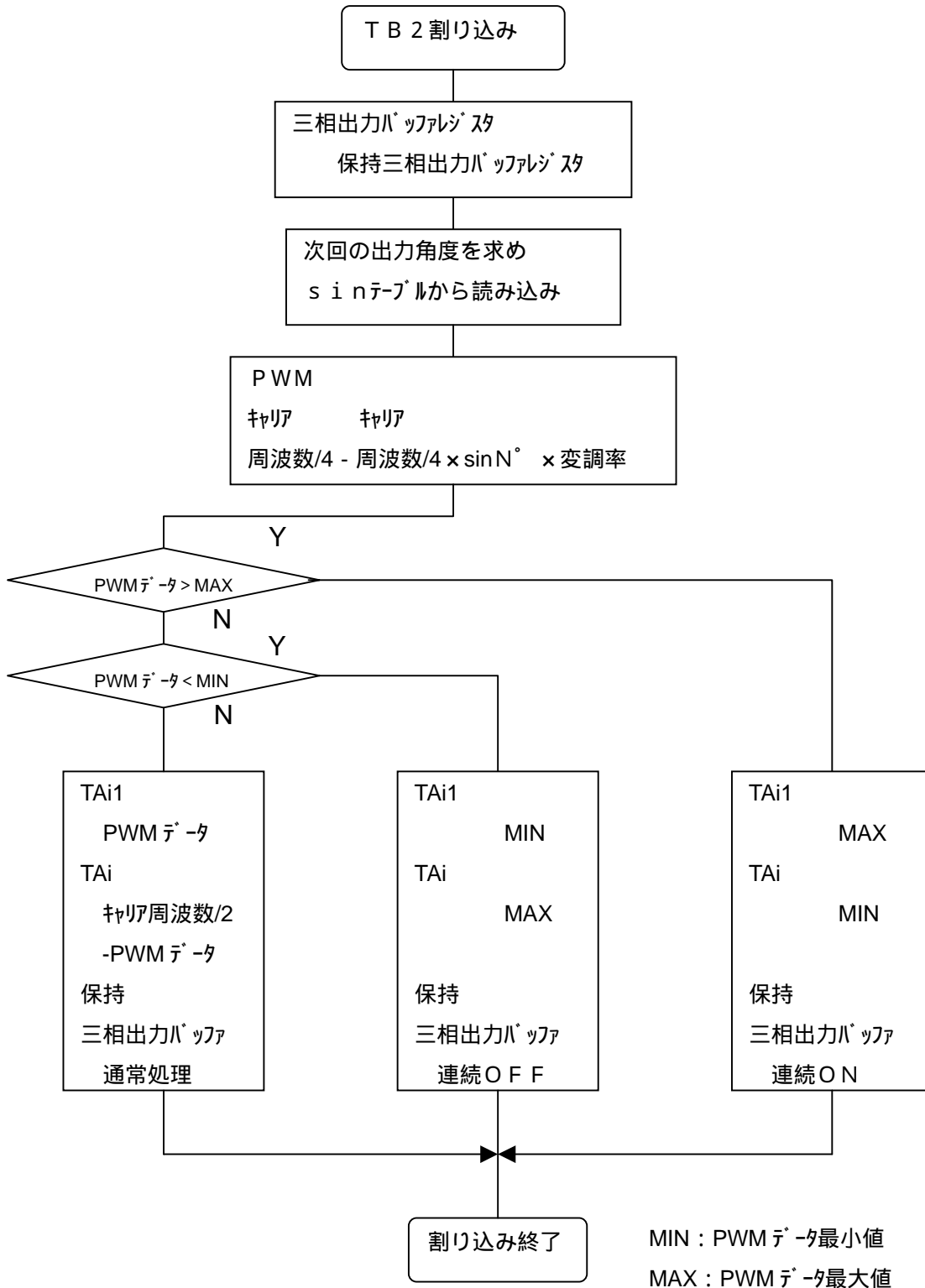
図5.6 三相出力バッファレジスタ書き換えのタイミング





三相出力バッファレジスタの内容は書き換え後、最初のTB2のアンダフローでシフトレジスタに転送されます。

図 5 . 7 T B 2 割り込み処理フローチャート例 (方法 1)



MIN : PWM デューティ最小値
MAX : PWM デューティ最大値

短絡防止タイマカウンタ f1 の場合

MIN:1 MAX:TB2 設定値

5.5.2 タイマ設定値で連続ON / OFF 波形出力を実現する

下図のようにTAi、TAi1レジスタに0を設定し、シフトレジスタの内容がシフトしない状態を作り出すことによって実現します。

図5.8 TAiデータセットのタイミングと反映の関係

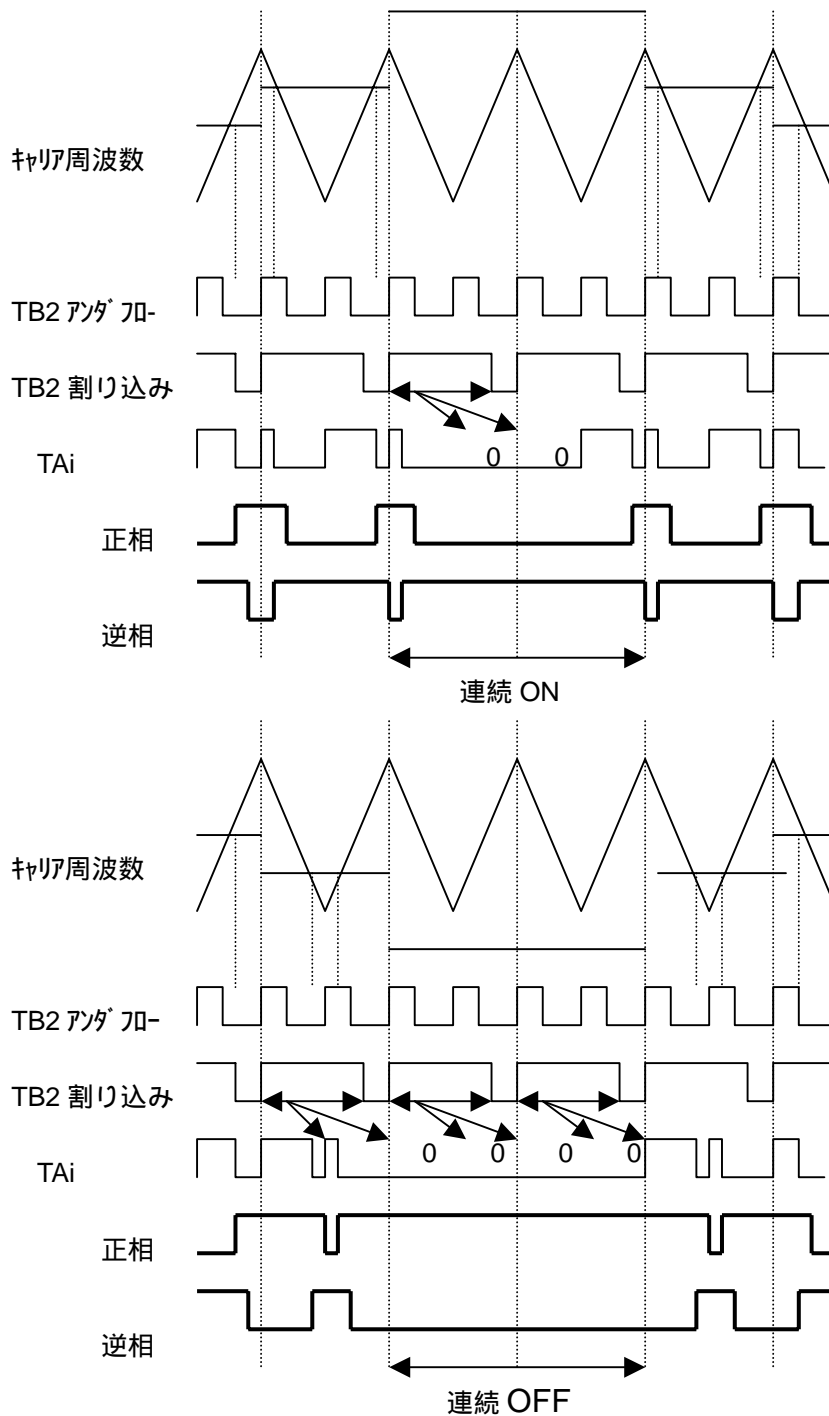
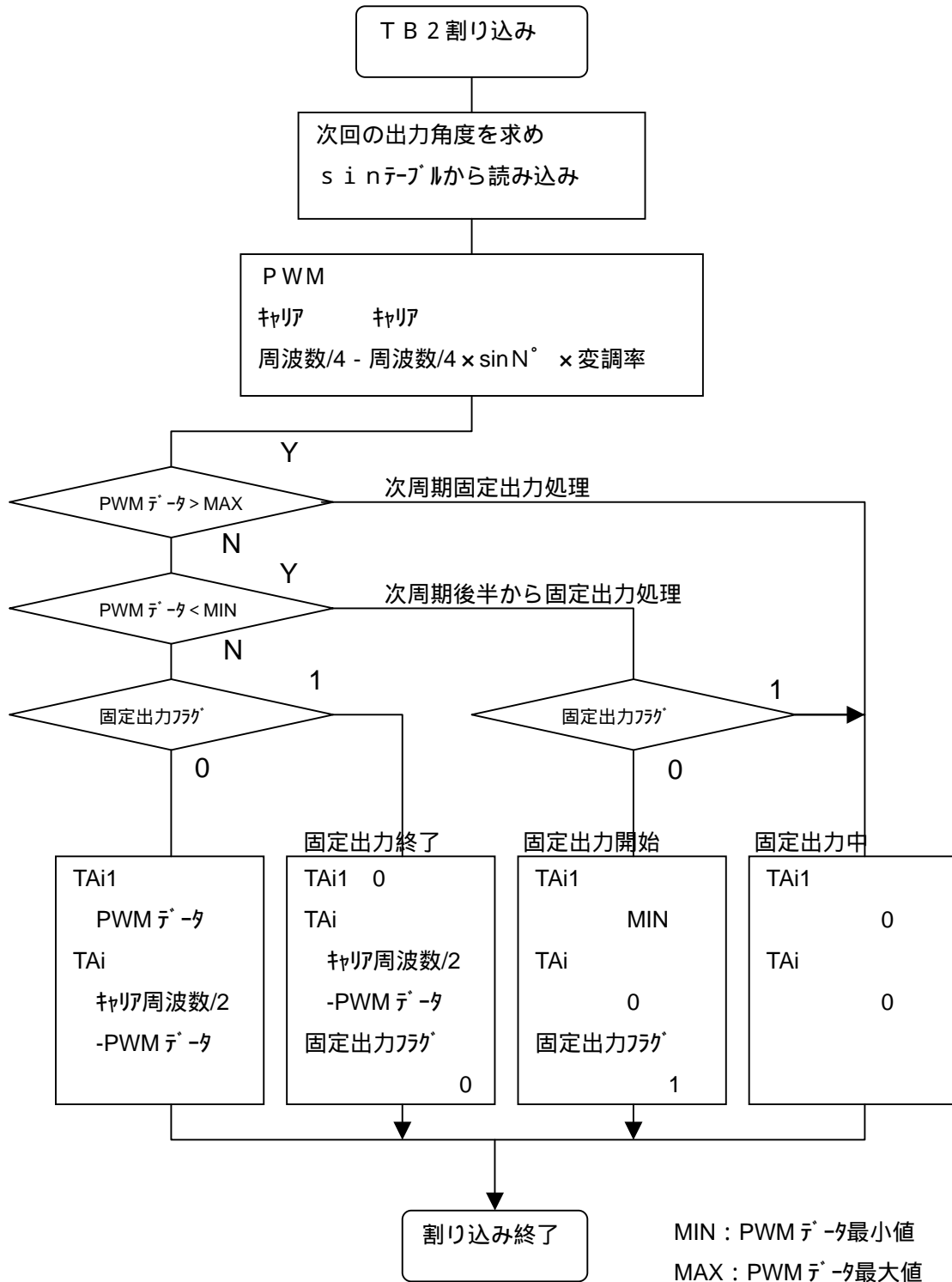


図5.9 TB2 割り込み処理フローチャート例 (方法2)



短絡防止タイマカウンタ f1 の場合

MIN:1 MAX:TB2 設定値

5.6 出力周波数の可変

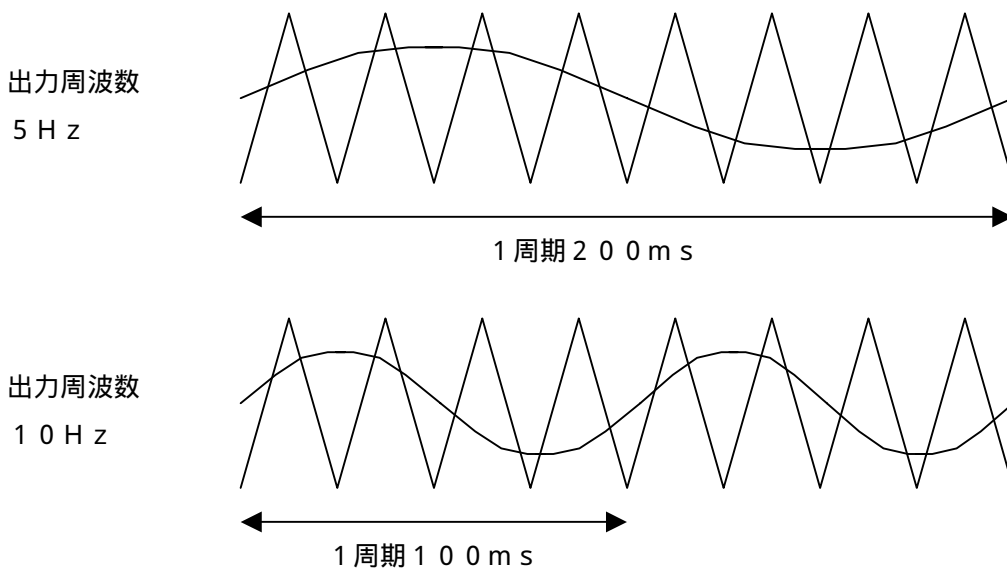
出力周波数の可変は、正弦波テーブルを飛ばし読みする等、正弦波の出力角度を操作することにより実現できます。以下に例として誘導モータの出力角度の操作方法を記します。

例えば、キャリア周波数 3.6 KHz、データテーブルを 1° 単位で 360 個用意した場合、このとき出力周波数が 10 Hz ならば、データテーブルをキャリア毎に 1 つずつ読み出します。これを 5 Hz に変更するには、正弦波の周期が 2 倍となるので、データテーブルを 2 度読みすれば良いことになります。

(基 準)

出力角度	10 Hz 時の sinN°	5 Hz 時の sinN°
0°	0	0
1°	0.0175	0
2°	0.0349	0.0175
3°	0.0523	0.0175
4°	0.0698	0.0349
5°	0.0872	0.0349
6°	0.1045	0.0523
7°	0.1219	0.0523
8°	0.1392	0.0698
9°	0.1564	0.0698
10°	0.1736	0.0872

図 5.10 出力周波数の可変



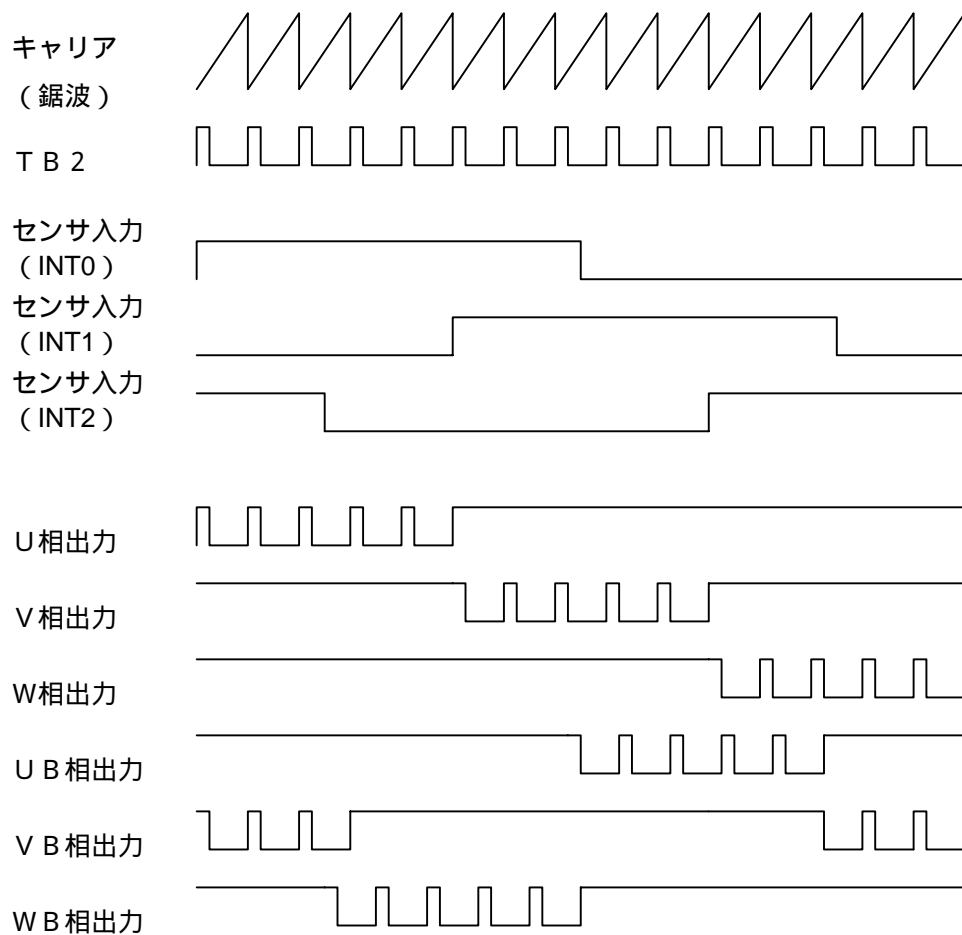
6.0 120°通電制御の実現方法

6.1 120°通電波形を出力するには

本章では、三相モータ制御用タイマ機能を使用した120°通電波形の出力例を紹介します。

6.2 三相モータ制御用タイマ機能での出力方法

図6.1 120°通電制御のセンサ入力と波形出力の関係

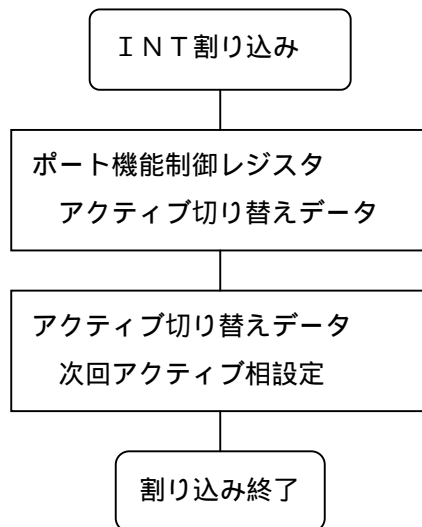
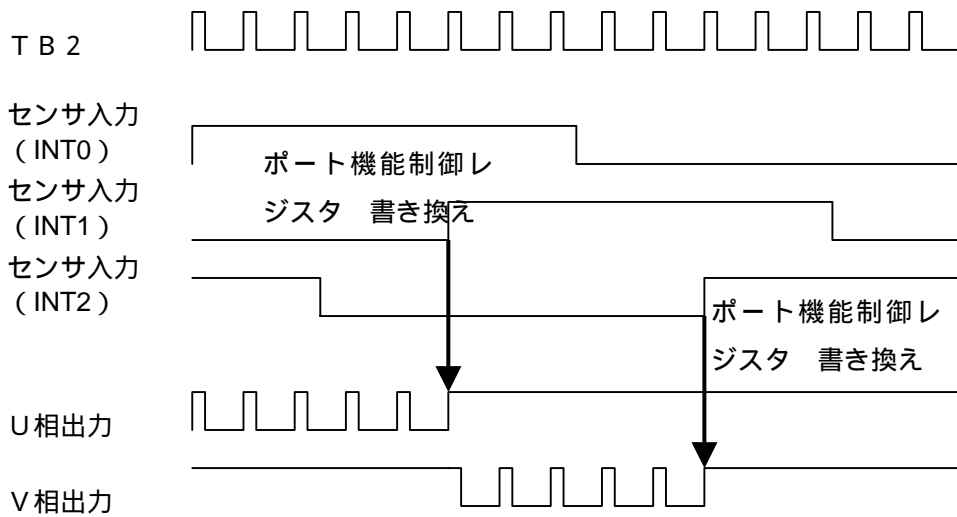


変調モード選択を鋸波変調モードとし、三相モード0を使用します。INT割り込み内で各相のアクティブをポート機能制御レジスタを書き換えることによって切り替えます。

(invc0=*1**1100; invc1=00000*0*; idb0=00000000; idb1=00111111; pfcr=00*****)

アクティブ相の切り替えは三相出力バッファの書き換えによっても可能です。

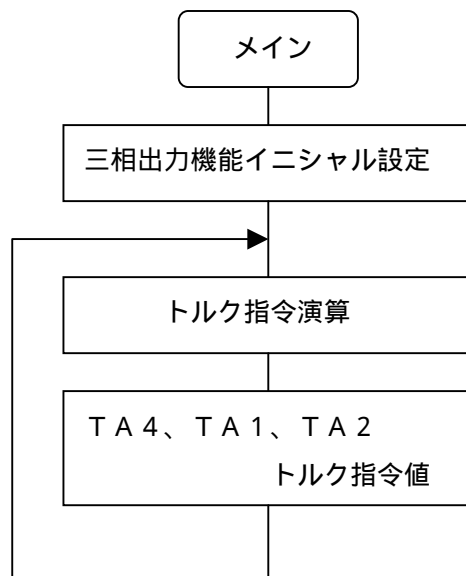
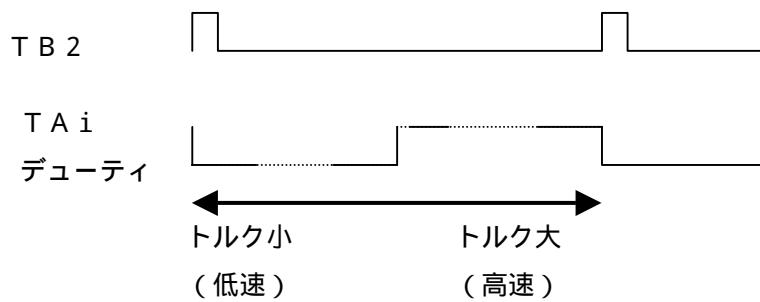
図6.2 センサ割り込みと出力相



6.3 速度制御

120°通電制御では基本的にトルクと速度は比例関係にあります。具体的にはトルク指令値が変わる毎にTA4、TA1、TA2値を書き換え制御します。

図6.3 トルク指令値とTAiの関係



7.0 参考プログラム例

以降に三相モータ駆動波形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。

7.1 正弦波出力参考プログラム

三相正弦波出力を行う際の例を示します。

```

/*****
*   正弦波出力参考プログラム
*
*****/

/*****
/*                                     */
/*                                     */
/*   S F R 設定                         */
/*                                     */
*****/

volatile char invc0;
#pragma ADDRESS invc00348h           /* 三相 P W M 制御レジスタ 0 */
volatile char invc1;
#pragma ADDRESS invc10349h           /* 三相 P W M 制御レジスタ 1 */
volatile char ictb2;
#pragma ADDRESS ictb2 034dh           /* 割り込み頻度設定カウンタ */
volatile char idb0;
#pragma ADDRESS idb0 034ah           /* 三相出力バッファレジスタ 0 */
volatile char idb1;
#pragma ADDRESS idb1 034bh           /* 三相出力バッファレジスタ 1 */
volatile char ta1mr;
#pragma ADDRESS ta1mr 0397h           /* タイマ A 1 モードレジスタ */
volatile char ta2mr;
#pragma ADDRESS ta2mr 0398h           /* タイマ A 2 モードレジスタ */
volatile char ta4mr;
#pragma ADDRESS ta4mr 039ah           /* タイマ A 4 モードレジスタ */
volatile char tb2mr;
#pragma ADDRESS tb2mr 039dh           /* タイマ B 2 モードレジスタ */
volatile char tb2sc;
#pragma ADDRESS tb2sc039eh           /* タイマ B 2 特殊モードレジスタ */
volatile char trgsr;
#pragma ADDRESS trgsr 0383h           /* トリガー選択レジスタ */
volatile short tb2;
#pragma ADDRESS tb2 0394h            /* タイマ B 2 */

volatile char dtt;
#pragma ADDRESS dtt 034ch            /* 短絡防止タイマ */
volatile short ta4;

```

```

#pragma ADDRESS ta4 038eh          /* タイマ A 4 */
volatile short ta1;
#pragma ADDRESS ta1 0388h          /* タイマ A 1 */
volatile short ta2;
#pragma ADDRESS ta2 038ah          /* タイマ A 2 */
volatile short ta41;
#pragma ADDRESS ta41 0346h         /* タイマ A 4 - 1 */
volatile short ta11;
#pragma ADDRESS ta11 0342h         /* タイマ A 1 - 1 */
volatile short ta21;
#pragma ADDRESS ta21 0344h         /* タイマ A 2 - 1 */
volatile char tb2ic;
#pragma ADDRESS tb2ic 005ch         /* タイマ B 2 割り込み制御レジスタ */
volatile char tabsr;
#pragma ADDRESS tabsr 0380h         /* カウント開始フラグ */
volatile char prcr;
#pragma ADDRESS prcr 000ah         /* プロテクトレジスタ */

/*****
/*
/*   イニシャライズ
/*
/*
*****/
void main_ini(void);

#define CLK 20000000                /*マイコン周波数  H z*/
#define CARR 20000                  /*キャリア周波数  H z*/
#define DTT_TM 40                   /*短絡防止時間   x 0.1 μ s */

#define carr_set_2 ((CLK/CARR)/2)   /*キャリア周期 1 / 2      */
#define carr_set_4 ((CLK/CARR)/4)   /*キャリア周期 1 / 4      */
#define dtt_set ((CLK*DTT_TM)/1000000)/*短絡防止タイマ値      */

```

```

void main_ini()
{
    ictb2=1;          /* T B 2 アンダフロー 2 回に 1 回割り込み*/

    prcr=0x02;      /* プロテクト解除*/
    invc0=0x16;     /* 三相モータ制御用タイマ機能選択 出力停止状態 */
    invc1=0x42;     /* 三相モード 1 */
    tb2sc=0x01;    /* リロードタイミングをキャリア 1 周期に同期 */
    prcr=0x00;     /* プロテクト*/

    idb0=0x15;     /*三相出力バッファレジスタ 0 設定*/
    idb1=0x2a;     /*三相出力バッファレジスタ 1 設定*/

    ta1mr=0x12;    /*ワンショットパルスモード */
    ta2mr=0x12;    /*ワンショットパルスモード */
    ta4mr=0x12;    /*ワンショットパルスモード */
    tb2mr=0x00;    /*タイマモード */
    trgsr=0x45;    /*トリガ選択レジスタ T B 2 トリガ*/

    tb2=carr_set_2-1; /*キャリア周期 1 / 2 */
    dtt=dtc_set;     /*短絡防止タイマ値 */
    ta4=ta1=ta2=carr_set_4; /*デューティ 5 0 %*/
    ta41=ta11=ta21=carr_set_4; /*デューティ 5 0 %*/

    tb2ic=0x07;     /* 割り込みレベル 7 */

    asm(" FSET I"); /* 割り込み許可 */

    tabsr=0x96;     /*カウント開始フラグ タイマスタート*/

    prcr=0x02;     /* プロテクト解除*/
    invc0=0x1e;     /* 出力許可*/
    prcr=0x00;     /* プロテクト*/
}

```



```

/*****/
/*                                     */
/*  sinテーブル                       */
/*                                     */
/*  演算精度を上げるため、sinテーブルにFFFF/2を乗算して用意します。*/
/*                                     */
/*****/

```

```

const short sin_tbl[610]=
{
    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

```

};

```

/*****/
/*
/*   メイン処理での演算例（誘導モータの場合）
/*   s i n テーブルの刻み値算出
/*   出力周波数に対応したトルク指令値算出
/*
/*****/
void main_pro(void);

signed short out_bin=100;    /*出力周波数値 仮値*/
signed short torq=1500;     /*トルクデータ 仮値*/
signed short tq_dat;        /*トルク指令値×キャリア / 4*/
signed short sin_cut;      /*S I N ポインタ生成用刻み値*/

void main_pro()
{

/* s i n 刻み値 = 23040 × 出力周波数 ÷ キャリア周波数   23040=360° × 64   */
  sin_cut=(signed short)((((signed long)out_bin*23040)/CARR));

  tq_dat=(signed short)((((signed long)torq*carr_set_4)/1000)); /*トルク指令値×キャリア/4*/

}

```

```

/*****/
/*
/*   T B 2 割り込み処理
/*
/*****/

void tb2_int(void);

void pwm_uvw_set(void);
void pwm_uvwa_set(void);
void pwm_uvwb_set(void);
void pwm_buf(void);
void i_con(void);
void angle(void);

unsigned char idb0_b=0x15; /*三相出力バッファレジスタ 0 作業用*/
unsigned char idb1_b=0x2a; /*三相出力バッファレジスタ 1 作業用*/
signed short tq_dat; /*トルク指令値×キャリア / 4*/
signed short sinpt_sum; /*S I Nポイント総計用カウンタ*/
signed short sin_pt; /*S I Nテーブルポイント*/
signed short pwm_u_w; /*U相P W M指令値*/
signed short pwm_v_w; /*V相P W M指令値*/
signed short pwm_w_w; /*W相P W M指令値*/
#define pwm_max (carr_set_2-1) /*P W M設定最大値*/
#define pwm_min 1 /*P W M設定最小値*/

/*   T B 2 割り込み (二相変調なし)
*/

#pragma INTERRUPT/B tb2_int
void tb2_int(void)
{
    angle(); /*S I Nテーブルポイントの生成 */
    i_con(); /*P W M指令値の算出 */
    pwm_uvw_set(); /*P W M指令値上限補正 タイマ設定*/
}

```

```
/*      T B 2 割り込み ( タイマ値書き換えによる二相変調有り )      */
```

```
#pragma INTERRUPT/B tb2_int
```

```
void tb2_int(void)
```

```
{
```

```
    angle();                /* S I N テーブルポインタの生成 */
```

```
    i_con();                /* P W M 指令値の算出          */
```

```
    pwm_uvwa_set();        /* P W M 指令値上限補正   タイマ設定*/
```

```
}
```

```
/*      T B 2 割り込み ( 三相出力バッファ書き換えによる二相変調有り )      */
```

```
#pragma INTERRUPT/B tb2_int
```

```
void tb2_int(void)
```

```
{
```

```
    pwm_buf();              /* 三相出力バッファ設定   */
```

```
    angle();                /* S I N テーブルポインタの生成 */
```

```
    i_con();                /* P W M 指令値の算出          */
```

```
    pwm_uvwb_set();        /* P W M 指令値上限補正   タイマ設定*/
```

```
}
```

```

/*****/
/*
/*   T B 2 割り込み用演算モジュール例 (誘導モータの場合)   */
/*       s i n 角度の算出                                     */
/*       P W M デューティの算出                             */
/*
/*****/

void angle()
{
    sinpt_sum=sin_cut+sinpt_sum;    /*sin 角度の総計 sin 刻み値+sin 角度の総計*/
    if(sinpt_sum>23040)              /*sin 角度の総計最大値か? 23040=360° × 64*/
    {
        sinpt_sum=sinpt_sum-23040; /*sin 角度の総計最大値補正*/
    }
    else
    {
    }
    sin_pt=sinpt_sum>>6;             /*sin 角度 sin 角度の総計 ÷ 64*/
}

void i_con()
{
/*   U 相 pwm 指令値=キャリア/4-(sin N° × (トルク指令値 × キャリア/4))   */
    pwm_u_w=carr_set_4-(signed short)((((signed long)sin_tbl[sin_pt]*(signed long)(tq_dat*2))>>16);

/*   V 相 pwm 指令値=キャリア/4-(sin(N+120)° × (トルク指令値 × キャリア/4))   */
    pwm_v_w=carr_set_4-
        (signed short)((((signed long)sin_tbl[sin_pt+120]*(signed long)(tq_dat*2))>>16);

/*   W 相 pwm 指令値=キャリア/4-(sin(N+240)° × (トルク指令値 × キャリア/4))   */
    pwm_w_w=carr_set_4-
        (signed short)((((signed long)sin_tbl[sin_pt+240]*(signed long)(tq_dat*2))>>16);

}

```

```

/*****/
/*
/*   T B 2 割り込み用モジュール
/*   P W Mデータの設定
/*
/*****/

/*           二相変調なし           */

void pwm_uvw_set()
{

/* U相 P W M補正*/
  if(pwm_max<pwm_u_w)                /*デューティ M A Xか?*/
  {
    ta41=pwm_max;                    /*前半 M A X値 */
    ta41=pwm_max;                    /*前半 M A X値 */
    ta4=pwm_min;                     /*後半 M I N値 */
  }
  else
  {
    if(pwm_min>pwm_u_w)              /*デューティ M I Nか?*/
    {
      ta41=pwm_min;                 /*前半 M I N値 */
      ta41=pwm_min;                 /*前半 M I N値 */
      ta4=pwm_max;                  /*後半 M A X値 */
    }
    else                             /*M I N < デューティ < M A X*/
    {
      ta41=pwm_u_w;                 /*前半 P W M指令値 */
      ta41=pwm_u_w;                 /*前半 P W M指令値 */
      ta4=carr_set_2-pwm_u_w;       /*後半 キャリア周期/2-U相 P W M 指令値*/
    }
  }
}

/* V相 P W M補正*/
  if(pwm_max<pwm_v_w)                /*デューティ M A Xか?*/
  {
    ta11=pwm_max;                   /*前半 M A X値 */
    ta11=pwm_max;                   /*前半 M A X値 */
    ta1=pwm_min;                    /*後半 M I N値 */
  }
  else
  {
    if(pwm_min>pwm_v_w)              /*デューティ M I Nか?*/
    {
      ta11=pwm_min;                 /*前半 M I N値 */

```

```

    ta11=pwm_min;          /*前半 MIN値 */
    ta1=pwm_max;          /*後半 MAX値 */
}
else                      /*MIN < デューティ < MAX*/

{
    ta11=pwm_v_w;        /*前半 PWM指令値 */
    ta11=pwm_v_w;        /*前半 PWM指令値 */
    ta1=carr_set_2-pwm_v_w; /*後半 キャリア周期/2-V相 PWM 指令値*/
}
}

/*W相PWM補正*/
if(pwm_max<pwm_w_w)      /*デューティMAXか?*/
{
    ta21=pwm_max;        /*前半 MAX値 */
    ta21=pwm_max;        /*前半 MAX値 */
    ta2=pwm_min;         /*後半 MIN値 */
}
else
{
    if(pwm_min>pwm_w_w)  /*デューティMINか?*/
    {
        ta21=pwm_min;    /*前半 MIN値 */
        ta21=pwm_min;    /*前半 MIN値 */
        ta2=pwm_max;     /*後半 MAX値 */
    }
    else                  /*MIN < デューティ < MAX*/
    {
        ta21=pwm_w_w;    /*前半 PWM指令値 */
        ta21=pwm_w_w;    /*前半 PWM指令値 */
        ta2=carr_set_2-pwm_w_w; /*後半 キャリア周期/2-W相 PWM 指令値*/
    }
}
}
}

```



```

/* タイマ値 " 0 " 設定による二相変調有り */

struct tag{
    char bit0:1;
    char bit1:1;
    char bit2:1;
}flag_buf;
#define arm_u_flg flag_buf.bit0 /*U相連続アーム出力判定フラグ*/
#define arm_v_flg flag_buf.bit1 /*V相連続アーム出力判定フラグ*/
#define arm_w_flg flag_buf.bit2 /*W相連続アーム出力判定フラグ*/

void pwm_uvwa_set()
{
/* U相PWM補正*/
    if(pwm_max<pwm_u_w) /*デューティMAXか?*/
    {
        ta41=0; /*固定出力開始 / 中*/
        ta41=0; /*固定出力開始 / 中*/
        ta4=0; /* 0 */
    }
    else
    {
        if(pwm_min>pwm_u_w) /*デューティMINか?*/
        {
            if(arm_u_flg==1) /*固定出力中 */
            {
                ta41=0; /*前半 0 */
                ta41=0; /*前半 0 */
                ta4=0; /*後半 0 */
            }
            else /*固定出力開始 */
            {
                ta41=pwm_min; /*前半 MIN値 */
                ta41=pwm_min; /*前半 MIN値 */
                ta4=0; /*後半 0 */
                arm_u_flg=1; /*固定出力フラグセット */
            }
        }
        else /*MIN < デューティ < MAX*/
        {
            if(arm_u_flg==1) /*固定出力終了*/
            {
                ta41=0; /*前半 0 */
                ta41=0; /*前半 0 */
                ta4=carr_set_2-pwm_u_w; /*後半 キャリア周期/2-U相PWM指令値*/
                arm_u_flg=0; /*固定出力フラグクリア */
            }
        }
    }
}

```

```

    }
    else          /*通常処理*/
    {
        ta41=pwm_u_w;          /*前半 PWM指令値 */
        ta41=pwm_u_w;          /*前半 PWM指令値 */
        ta4=carr_set_2-pwm_u_w; /*後半 キャリア周期/2-U相 PWM 指令値*/
    }
}

/* V相PWM補正*/
if(pwm_max<pwm_v_w)          /*デューティMAXか?*/
{
    ta11=0;                  /*固定出力開始 / 中*/
    ta11=0;                  /*固定出力開始 / 中*/
    ta1=0;                   /* 0 */
}
else
{
    if(pwm_min>pwm_v_w)      /*デューティMINか?*/
    {
        if(arm_v_flg==1)    /*固定出力中 */
        {
            ta11=0;         /*前半 0 */
            ta11=0;         /*前半 0 */
            ta1=0;          /*後半 0 */
        }
        else                /*連続アーム開始 */
        {
            ta11=pwm_min;   /*前半 MIN値 */
            ta11=pwm_min;   /*前半 MIN値 */
            ta1=0;          /*後半 0 */
            arm_v_flg=1;    /*固定出力フラグセット */
        }
    }
}
else          /*MIN < デューティ < MAX*/
{
    if(arm_v_flg==1)      /*固定出力終了*/
    {
        ta11=0;           /*前半 0 */
        ta11=0;           /*前半 0 */
        ta1=carr_set_2-pwm_v_w; /*後半 キャリア周期/2-V相 PWM 指令値*/
        arm_v_flg=0;      /*固定出力フラグクリア */
    }
    else                /*通常処理*/
    {
        ta11=pwm_v_w;     /*前半 PWM指令値 */
        ta11=pwm_v_w;     /*前半 PWM指令値 */
        ta1=carr_set_2-pwm_v_w; /*後半 キャリア周期/2-V相 PWM 指令値*/
    }
}

```

```

    }
  }
}

/*W相PWM補正*/
if(pwm_max<pwm_w_w)          /*デューティMAXか?*/
{
  ta21=0;                    /*固定出力開始 / 中*/
  ta21=0;                    /*固定出力開始 / 中*/
  ta2=0;                      /* 0 */
}
else
{
  if(pwm_min>pwm_w_w)        /*デューティMINか?*/
  {
    if(arm_w_flg==1)        /*固定出力中 */
    {
      ta21=0;                /*前半 0 */
      ta21=0;                /*前半 0 */
      ta2=0;                  /*後半 0 */
    }
    else                    /*固定出力開始 */
    {
      ta21=pwm_min;          /*前半 MIN値 */
      ta21=pwm_min;          /*前半 MIN値 */
      ta2=0;                  /*後半 0 */
      arm_w_flg=1;           /*固定出力フラグセット */
    }
  }
  else                      /*MIN < デューティ < MAX*/
  {
    if(arm_w_flg==1)        /*固定出力終了*/
    {
      ta21=0;                /*前半 0 */
      ta21=0;                /*前半 0 */
      ta2=carr_set_2-pwm_w_w; /*後半 キャリア周期/2-W相PWM指令値*/
      arm_w_flg=0;           /*連続アームフラグクリア */
    }
    else                    /*通常処理*/
    {
      ta21=pwm_w_w;          /*前半 PWM指令値 */
      ta21=pwm_w_w;          /*前半 PWM指令値 */
      ta2=carr_set_2-pwm_w_w; /*後半 キャリア周期/2-W相PWM指令値*/
    }
  }
}
}
}

```

```

/* 三相出力バッファレジスタ設定による二相変調有り */

void pwm_buf()
{
    idb0=idb0_b;      /*三相出力バッファレジスタ 0 設定*/
    idb1=idb1_b;      /*三相出力バッファレジスタ 1 設定*/
}

void pwm_uvwb_set()
{
    /* U 相 P W M 補正*/
    if(pwm_max<pwm_u_w)          /*デューティ M A X か?*/
    {
        idb0_b=idb0_b&0xfc;
        idb0_b=idb0_b|0x02;

        ta41=pwm_max;           /*前半 M A X 値 */
        ta41=pwm_max;           /*前半 M A X 値 */
        ta4=pwm_min;            /*後半 M I N 値 */
    }
    else
    {
        if(pwm_min>pwm_u_w)      /*デューティ M I N か?*/
        {
            idb1_b=idb1_b&0xfc;
            idb1_b=idb1_b|0x01;
            ta41=pwm_min;        /*前半 M I N 値 */
            ta41=pwm_min;        /*前半 M I N 値 */
            ta4=pwm_max;         /*後半 M A X 値 */
        }
        else                      /*M I N < デューティ < M A X*/
        {
            idb0_b=idb0_b&0xfc;
            idb0_b=idb0_b|0x01;
            idb1_b=idb1_b&0xfc;
            idb1_b=idb1_b|0x02;

            ta41=pwm_u_w;        /*前半 P W M 指令値 */
            ta41=pwm_u_w;        /*前半 P W M 指令値 */
            ta4=carr_set_2-pwm_u_w; /*後半 キャリア周期/2-U相 PWM 指令値*/
        }
    }
}

```

```

/* V相PWM補正*/
if(pwm_max<pwm_v_w)                /*デューティMAXか?*/
{
    idb0_b=idb0_b&0xf3;
    idb0_b=idb0_b|0x08;

    ta11=pwm_max;                    /*前半 MAX値 */
    ta11=pwm_max;                    /*前半 MAX値 */
    ta1=pwm_min;                      /*後半 MIN値 */
}
else
{
    if(pwm_min>pwm_v_w)              /*デューティMINか?*/
    {
        idb1_b=idb1_b&0xf3;
        idb1_b=idb1_b|0x04;

        ta11=pwm_min;                /*前半 MIN値 */
        ta11=pwm_min;                /*前半 MIN値 */
        ta1=pwm_max;                  /*後半 MAX値 */
    }
    else                              /*MIN<デューティ<MAX*/
    {
        idb0_b=idb0_b&0xf3;
        idb0_b=idb0_b|0x04;
        idb1_b=idb1_b&0xf3;
        idb1_b=idb1_b|0x08;

        ta11=pwm_v_w;                /*前半 PWM指令値 */
        ta11=pwm_v_w;                /*前半 PWM指令値 */
        ta1=carr_set_2-pwm_v_w;      /*後半 キャリア周期/2-V相PWM指令値*/
    }
}

/*W相PWM補正*/
if(pwm_max<pwm_w_w)                /*デューティMAXか?*/
{
    idb0_b=idb0_b&0xcf;
    idb0_b=idb0_b|0x20;

    ta21=pwm_max;                    /*前半 MAX値 */
    ta21=pwm_max;                    /*前半 MAX値 */
    ta2=pwm_min;                      /*後半 MIN値 */
}
else
{
    if(pwm_min>pwm_w_w)              /*デューティMINか?*/
    {

```

```

    idb1_b=idb1_b&0xcf;
    idb1_b=idb1_b|0x10;

    ta21=pwm_min;          /*前半 MIN値 */
    ta21=pwm_min;          /*前半 MIN値 */
    ta2=pwm_max;           /*後半 MAX値 */
}
else                        /*MIN < デューティ < MAX*/
{
    idb0_b=idb0_b&0xcf;
    idb0_b=idb0_b|0x10;
    idb1_b=idb1_b&0xcf;
    idb1_b=idb1_b|0x20;

    ta21=pwm_w_w;          /*前半 PWM指令値 */
    ta21=pwm_w_w;          /*前半 PWM指令値 */
    ta2=carr_set_2-pwm_w_w; /*後半 キャリア周期/2-W相 PWM 指令値*/
}
}
}

```

7.2 120°通電参考プログラム

120°通電出力を行う際の例を示します。

```

/*****
*   120°通電参考プログラム
*
*****/
/*****/
/*           */
/*   S F R 設定           */
/*           */
/*****/
volatile char invc0;
#pragma ADDRESS   invc0   0348h   /* 三相PWM制御レジスタ0 */
volatile char invc1;
#pragma ADDRESS   invc1   0349h   /* 三相PWM制御レジスタ1 */
volatile char ictb2;
#pragma ADDRESS   ictb2   034dh   /* 割り込み頻度設定カウンタ */
volatile char idb0;
#pragma ADDRESS   idb0   034ah   /* 三相出力バッファレジスタ0 */
volatile char idb1;
#pragma ADDRESS   idb1   034bh   /* 三相出力バッファレジスタ1 */
volatile char ta1mr;
#pragma ADDRESS   ta1mr   0397h   /* タイマA1モードレジスタ */
volatile char ta2mr;
#pragma ADDRESS   ta2mr   0398h   /* タイマA2モードレジスタ */
volatile char ta4mr;
#pragma ADDRESS   ta4mr   039ah   /* タイマA4モードレジスタ */
volatile char tb2mr;
#pragma ADDRESS   tb2mr   039dh   /* タイマB2モードレジスタ */
volatile char tb2sc;
#pragma ADDRESS   tb2sc   039eh   /* タイマB2特殊モードレジスタ */
volatile char trgsr;
#pragma ADDRESS   trgsr   0383h   /* トリガー選択レジスタ */
volatile short tb2;
#pragma ADDRESS   tb2     0394h   /* タイマB2 */
volatile char dtt;
#pragma ADDRESS   dtt     034ch   /* 短絡防止タイマ */
volatile short ta4;
#pragma ADDRESS   ta4     038eh   /* タイマA4 */
volatile short ta1;
#pragma ADDRESS   ta1     0388h   /* タイマA1 */
volatile short ta2;
#pragma ADDRESS   ta2     038ah   /* タイマA2 */
volatile short ta41;
#pragma ADDRESS   ta41    0346h   /* タイマA4 - 1 */

```

```
volatile short ta11;
#pragma ADDRESS ta11 0342h /* タイマ A 1 - 1 */
volatile short ta21;
#pragma ADDRESS ta21 0344h /* タイマ A 2 - 1 */
volatile char tb2ic;
#pragma ADDRESS tb2ic 005ch /* タイマ B 2 割り込み制御レジスタ */
volatile char tabsr;
#pragma ADDRESS tabsr 0380h /* カウント開始フラグ */
volatile char prcr;
#pragma ADDRESS prcr 000ah /* プロテクトレジスタ */
volatile char ifsr;
#pragma ADDRESS ifsr 035fh /* 外部割り込み要因選択レジスタ */
volatile char int0ic;
#pragma ADDRESS int0ic 005dh /* I N T 0 割り込み制御レジスタ */
volatile char int1ic;
#pragma ADDRESS int1ic 005eh /* I N T 1 割り込み制御レジスタ */
volatile char int2ic;
#pragma ADDRESS int2ic 005fh /* I N T 2 割り込み制御レジスタ */
volatile char p7;
#pragma ADDRESS p7 03edh /* ポート P 7 レジスタ */
volatile char p8;
#pragma ADDRESS p8 03f0h /* ポート P 8 レジスタ */
volatile char pd7;
#pragma ADDRESS pd7 03efh /* ポート P 7 方向レジスタ */
volatile char pd8;
#pragma ADDRESS pd8 03f2h /* ポート P 8 方向レジスタ */
volatile char tprc;
#pragma ADDRESS tprc 025ah /* 三相プロテクト制御レジスタ */
volatile char pfcr;
#pragma ADDRESS pfcr 0358h /* ポート機能制御レジスタ */
```

```
/*
 *
 *   イニシャライズ
 *
 */
```

```
void main_ini(void);
```

```
#define CLK 2000000 /*マイコン周波数 H z*/
```

```
#define CARR 20000 /*キャリア周波数 H z*/
```

```
#define carr_set (CLK/CARR) /*キャリア周期 */
```



```

void main_ini()
{
    ictb2=1;      /* T B 2 アンダーフロー 2 回に 1 回割り込み*/

    prcr=0x02;   /* プロテクト解除 */
    invc0=0x44;  /*三相モータ制御用タイマ機能選択 出力停止状態*/
    invc1=0x60;  /*三相モード 0 短絡防止タイマ無効*/
    tb2sc=0x02;  /* リードタイミングを TB2 に同期 */
    prcr=0x00;   /* プロテクト */

    p7=0x3c;     /* V、V B、W、W B ポート非アクティブ設定*/
    pd7=0x3c;    /* V、V B、W、方向出力設定*/
    p8=0x03;     /* U、U B ポート非アクティブ設定*/
    pd8=0x03;    /* U、U B ポート方向出力設定*/
    tprc=0x01;   /*ポート機能制御レジスタへの書き込み許可*/
    pfcrc=0x00;  /*ポート機能制御レジスタポート出力*/

    idb0=0x00;   /*三相出力バッファレジスタ 0 設定*/
    idb1=0x3f;   /*三相出力バッファレジスタ 1 設定*/

    ta1mr=0x12;  /*ワンショットパルスモード */
    ta2mr=0x12;  /*ワンショットパルスモード */
    ta4mr=0x12;  /*ワンショットパルスモード */
    tb2mr=0x00;  /*タイマモード */
    trgsr=0x45;  /*トリガ選択レジスタ T B 2 トリガ*/

    tb2=carr_set; /*キャリア周期 */
    dtt=1;        /*短絡防止タイマ最小設定*/

    ta4=1;        /*トルク 0 %*/
    ta1=1;        /*トルク 0 %*/
    ta2=1;        /*トルク 0 %*/

    /*位置入力に I N T を使用*/
    ifsr=0x07;    /* I N T 0 , 1 , 2 両エッジ*/
    int0ic=7;     /* I N T 0 , 1 , 2 割り込みレベル設定*/
    int1ic=7;
    int2ic=7;

    asm(" FSET I"); /* 割り込み許可 */

    tabsr=0x96;   /*カウント開始フラグ タイマスタート*/

    prcr=0x02;   /* プロテクト解除 */
    invc0=0x4c;  /* 出力許可*/

```

```

prcr=0x00; /* プロテクト */
}

/*****/
/* */
/* メイン処理での演算 */
/* 速度に対応したトルク指令値 ( P W Mデューティ ) の設定 */
/* */
/*****/
void dc_pwm_set(void);

unsigned short tk_dat_w; /*トルク指令値*/

void dc_pwm_set(void)
{
    ta4=ta1=ta2=tk_dat_w;
}

/*****/
/* */
/* 角度検出 ( I N T使用時 ) 割り込み */
/* 各相のアクティブ相の三相出力バッファレジスタよる切り替え */
/* */
/*****/

#pragma INTERRUPT/B int0_int
void int0_int(void);
#pragma INTERRUPT/B int1_int
void int1_int(void);
#pragma INTERRUPT/B int2_int
void int2_int(void);
void rad_to_pwm(void);

unsigned char pfc_r_b=0x00; /*次回出力相*/
signed short rad_datw=0; /*入力角度*/
struct tag{
    char bit0:1;
    char bit1:1;
    char bit2:1;
    char bit3:1;

}dc_buf;
#define edge0_flg dc_buf.bit1
#define edge1_flg dc_buf.bit2

```

```
#define edge2_flg dc_buf.bit3
```

```
void int0_int(void)
{
    pfcr=pfcr_b;          /*出力相変更*/

    if(edge0_flg==0)
    {
        edge0_flg=1;
        rad_datw=60;      /*次回角設定*/
    }
    else
    {
        edge0_flg=0;
        rad_datw=240;     /*次回角設定*/
    }

    rad_to_pwm();        /*次回出力相設定*/
}

```

```
void int1_int(void)
{
    pfcr=pfcr_b;          /*出力相変更*/

    if(edge1_flg==0)
    {
        edge1_flg=1;
        rad_datw=180;     /*次回角設定*/
    }
    else
    {
        edge1_flg=0;
        rad_datw=0;       /*次回角設定*/
    }

    rad_to_pwm();        /*次回出力相設定*/
}

```

```
void int2_int(void)
{
    pfcr=pfcr_b;          /*出力相変更*/

```

```

if(edge2_flg==0)
{
    edge2_flg=1;
    rad_datw=300;    /*次回角設定*/
}
else
{
    edge2_flg=0;
    rad_datw=120;   /*次回角設定*/
}

rad_to_pwm();      /*次回出力相設定*/
}

```

```

void rad_to_pwm(void)
{
    switch(rad_datw)
    {
        case 0:
            pfc_r_b=0x21;    /*次回出力相設定  U          W B  */
            break;
        case 60:
            pfc_r_b=0x24;    /*次回出力相設定  V          W B  */
            break;
        case 120:
            pfc_r_b=0x06;    /*次回出力相設定  V  U B          */
            break;
        case 180:
            pfc_r_b=0x12;    /*次回出力相設定          W  U B          */
            break;
        case 240:
            pfc_r_b=0x18;    /*次回出力相設定          W          V B  */
            break;
        case 300:
            pfc_r_b=0x09;    /*次回出力相設定  U          V B          */
            break;
        default:
            pfc_r_b=0x00;    /*次回出力相設定  o f f          */
            break;
    }
}

```

```

volatile char  ta0mr;
#pragma ADDRESS ta0mr    0396h
volatile char  ta0ic;

```

```
#pragma ADDRESS      ta0ic      0055h
volatile short ta0;
#pragma ADDRESS      ta0        0386h

#pragma INTERRUPT/B  ta0_int
void ta0_int(void);
void ta0_int(void)
{
    pfcr=pfcr_b;      /*出力相変更*/

    rad_datw=rad_datw+60;
    if(rad_datw==360) rad_datw=0;
    rad_to_pwm();     /*次回出力相設定*/
}

void main(void)
{
    main_ini();

    tk_dat_w=carr_set>>1;

    dc_pwm_set();

    ta0mr=0x00;
    ta0=0xffff;
    ta0ic=0x07;
    tabsr=0x97;      /*カウント開始フラグ タイマスタート*/

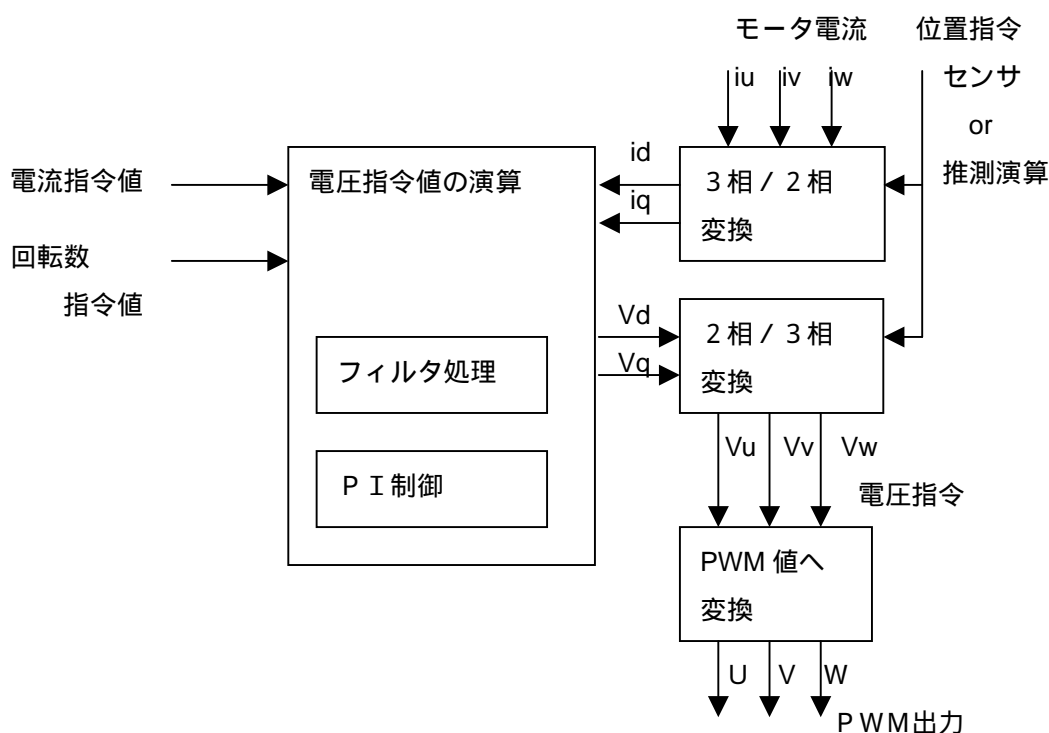
    while(1)
    {
    }
}
}
```

7.3 フィードバック制御参考プログラム

以降ではベクトル制御のプログラム開発例として、以下の制御モジュールを掲載します。

- ・ 3相 / 2相変換
- ・ 2相 / 3相変換
- ・ P I 制御
- ・ フィルタ処理

PWM出力については前項までの参考プログラムをご参照ください。



7.3.1 3相 / 2相変換モジュール

以下に3相 / 2相変換の一例を記します。

$$\begin{pmatrix} i_d \\ i_q \end{pmatrix} = \sqrt{\frac{2}{3}} \begin{pmatrix} \cos & \cos(+240) & \cos(+120) \\ -\sin & -\sin(+240) & -\sin(+120) \end{pmatrix} \begin{pmatrix} i_u \\ i_v \\ i_w \end{pmatrix}$$

以降のサンプルプログラムでは演算精度を出すために

i_u : U相電流値	$\times 2^6$
i_v : V相電流値	$\times 2^6$
i_w : W相電流値	$\times 2^6$
i_d : d軸電流値	$\times 2^6$
i_q : q軸電流値	$\times 2^6$
\sin テーブル	$\times 2^{15}$

として演算しています。

```

/*****
*      3相 / 2相変換参考プログラム
*
*****/
void dq_samp(void);

signed short iu_6w;      /* U相電流値[2^6]      */
signed short iv_6w;      /* V相電流値[2^6]      */
signed short iw_6w;      /* W相電流値[2^6]      */
signed short work_6w;    /* work[2^6]          */
signed short id_6w;      /* d軸電流[2^6]        */
signed short iq_6w;      /* q軸電流[2^6]        */
signed short sin_pt;     /* sinテーブルのインデックス */

void dq_samp()
{
    iw_6w=-(iu_6w+iv_6w);      /* W相電流値[2^6]=-(U相電流値[2^6]+V相電流値[2^6]) */
    work_6w=(signed short)((
        (sin_tbl[sin_pt+90]*(long)iu_6w)
        +(sin_tbl[sin_pt+90+240]*(long)iv_6w)
        +(sin_tbl[sin_pt+90+120]*(long)iw_6w)
    )>>15);
    /* work[2^6]=(
    /*      cos  × U相電流値[2^6]          */
    /*      +cos( +240) × V相電流値[2^6]      */
    /*      +cos( +120) × W電流値[2^6] )      */
    /*      sinテーブルは[2^15]              */

    id_6w=(short)((((long)work_6w*26755)>>15); /* d軸電流[2^6]=work[2^6] × 2/3 */

    work_6w=(signed short)((
        -(sin_tbl[sin_pt]*(long)iu_6w)
        -(sin_tbl[sin_pt+240]*(long)iv_6w)
        -(sin_tbl[sin_pt+120]*(long)iw_6w)
    )>>15);
    /* work[2^6]=(
    /*      -sin  × U相電流値[2^6]          */
    /*      -sin( +240) × V相電流値[2^6]      */
    /*      -sin( +120) × W電流値[2^6] )      */
    /*      sinテーブルは[2^15]              */

    iq_6w=(short)((((long)work_6w*26755)>>15); /* q軸電流[2^6]=work[2^6] × 2/3 */
}

```


7.3.2 2相 / 3相変換モジュール

以下に2相 / 3相変換の一例を記します。

$$\begin{pmatrix} V_u \\ V_v \\ V_w \end{pmatrix} = \sqrt{\frac{2}{3}} \begin{pmatrix} \cos & -\sin(\) \\ \cos(+240) & -\sin(+240) \\ \cos(+120) & -\sin(+120) \end{pmatrix} \begin{pmatrix} V_d \\ V_q \end{pmatrix}$$

以降のサンプルプログラムでは演算精度を出すために

V_d : d軸電圧値 $\times 2^6$
 V_q : q軸電圧値 $\times 2^6$
 V_u : U相電圧指令値 $\times 2^6$
 V_v : V相電圧指令値 $\times 2^6$
 V_w : W相電圧指令値 $\times 2^6$
 \sin テーブル $\times 2^{15}$

として演算しています。

```

/*****
*      2相 / 3相変換参考プログラム
*
*****/
void uvw_samp(void);

signed short vd_6w;      /* d 軸電圧[2^6]      */
signed short vq_6w;      /* q 軸電圧[2^6]      */
signed short vu_6w;      /* U 相電圧指令[2^6] */
signed short vv_6w;      /* V 相電圧指令[2^6] */
signed short vw_6w;      /* W 相電圧指令[2^6] */
signed short work_6w;    /* work[2^6]          */
signed short sin_pt;     /* sin テーブルのインデックス */

void uvw_samp()
{
    work_6w=(signed short)((
        (sin_tbl[sin_pt+90]*(long)vd_6w)
        -(sin_tbl[sin_pt]*(long)vq_6w)
        )>>15);
        /* work[2^6]=(
            /*      cos   x d 軸電圧[2^6]      */
            /*      -sin  x q 軸電圧[2^6]      */
            /*      sin テーブルは[2^15]        */

    vu_6w=(short)(((long)work_6w*26752)>>15); /* U 相電圧指令[2^6]=work[2^6] x 2/3 */

    work_6w=(signed short)((
        (sin_tbl[sin_pt+90+240]*(long)vd_6w)
        -(sin_tbl[sin_pt+240]*(long)vq_6w)
        )>>15);
        /* work[2^6]=(
            /*      cos( +240) x d 軸電圧[2^6]      */
            /*      -sin( +240) x q 軸電圧[2^6]      */
            /*      sin テーブルは[2^15]        */

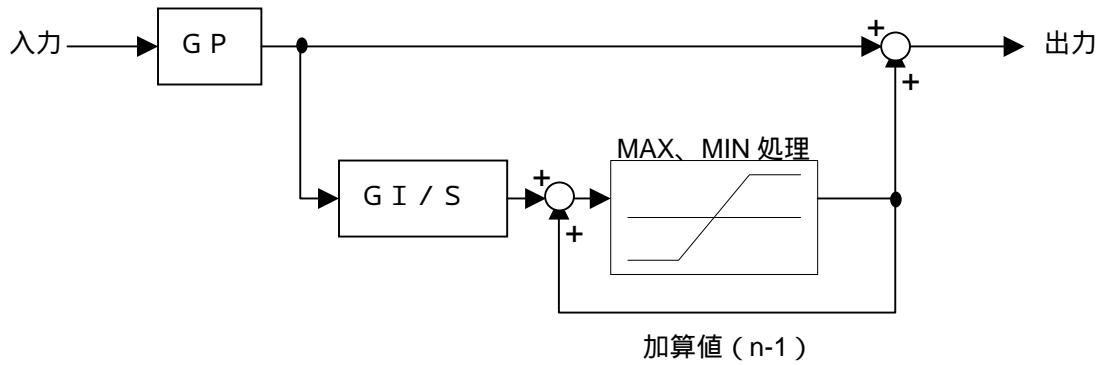
    vv_6w=(short)(((long)work_6w*26752)>>15); /* V 相電圧指令[2^6]=work[2^6] x 2/3 */

    vw_6w=-((vu_6w+vv_6w));
        /* W 相電圧指令[2^6]=
            /*      -(U 相電圧指令[2^6]+V 相電圧指令[2^6])
}

```

7.3.3 P I 制御モジュール

以下に P I 制御の一例を記します。



以降のサンプルプログラムでは演算精度を出すために

入力データ	$\times 2^6$
出力データ	$\times 2^6$
ゲイン P	$\times 2^{10}$
ゲイン I ÷ 制御周期	$\times 2^{18}$

として演算しています。

```

/*****
*   P I 制御参考プログラム
*
*****/
void PI_samp(void);

signed short data_in_6w;          /* 入力データ[2^6] */
signed short data_out_6w;         /* 出力データ[2^6] */
signed short GP_10w;              /* ゲイン P [2^10] */
signed short GI_t_18w;            /* (ゲイン I ÷ 制御周期 t)[2^18] */
signed long sum_16l;              /* 加算値[2^16] */
signed long work1_16l;            /* work1[2^16] */
signed long work2_16l;            /* work2[2^16] */
#define work2_limit 500           /* work2 リミット値 500 は仮データ */

void PI_samp()
{
    work1_16l = (long)data_in_6w * GP_10w; /*work1[2^16]=入力データ[2^6] × ゲイン P [2^10]*/

    sum_16l = sum_16l + ((work1_16l >> 11) * GI_t_18w) >> 7;
        /* 加算値[2^16]=加算値[2^16]+(work1[2^16] ÷ 2^11) × (ゲイン I ÷ t) [2^18] ÷ 2^7 */

    work2_16l = work1_16l + sum_16l;        /* work2[2^16]=work1[2^16]+加算値[2^16] */

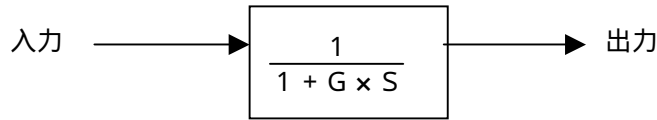
    if(work2_16l > work2_limit)
    {
        work2_16l = work2_limit;
    }else
    {
        if(work2_16l < -work2_limit)
        {
            work2_16l = -work2_limit;
        }
    }

    data_out_6w = (short)(work2_16l >> 10); /* 出力データ[2^6]=work2[2^16] ÷ 2^10 */
}

```

7.3.4 フィルタ制御モジュール

以下にフィルタ制御の一例を記します



以降のサンプルプログラムでは演算精度を出すために

入力データ × 2⁶
出力データ × 2⁶
ゲイン P × 2¹⁰
ゲイン I ÷ 制御周期 × 2¹⁸

として演算しています。

```

/*****
*      フィルタ制御参考プログラム
*
*****/
void fill_samp(void);

signed short data_in_6w;      /* 入力データ[2^6] */
signed short fill_out_6w;     /* 出力データ[2^6] */
signed short G_t21;          /* 1 ÷ (ゲイン × 制御周期) [2^21] */
signed long fill_22l;         /* 加算値[2^22] */

void fill_samp()
{
    fill_22l=fill_22l
        +(long)((G_t21*(long)(data_in_6w-fill_out_6w))>>5);
        /* 加算値[2^16]=加算値[2^22]
           +(1 ÷ (ゲイン × 制御周期) [2^21]
           × (入力データ[2^6]-出力データ[2^6])
           ) ÷ 2^5
        */
    fill_out_6w=(short)(fill_22l>>16); /* 出力データ[2^6]=加算値[2^22]>>16 */
}

```

8.0 参考ドキュメント

データシート

M16C/26A グループデータシート

M16C/29 グループデータシート

9.0 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/>

ルネサス製品全般に関するお問合せ先

カスタマ・サポート・センター：csc@renesas.com

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2004.09.01	-	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注意ください。

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すは、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、その他お気付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。