

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。

μ PD78F0714による単相インダクション・モータ制御

V/f制御による2相インバータ正弦波駆動編

μ PD78F0714

〔メモ〕

目次要約

第1章	制御方式	...	10
第2章	ハードウェア構成	...	15
第3章	ソフトウェア構成	...	17
第4章	プログラム・リスト	...	28

入力端子の印加波形

入力ノイズや反射波による波形歪みは誤動作の原因になりますので注意してください。

CMOSデバイスの入力が入力ノイズなどに起因して、 V_{IL} (MAX.) から V_{IH} (MIN.) までの領域にとどまるような場合は、誤動作を引き起こす恐れがあります。入力レベルが固定な場合はもちろん、 V_{IL} (MAX.) から V_{IH} (MIN.) までの領域を通過する遷移期間中にチャタリングノイズ等が入らないようご使用ください。

未使用入力の処理

CMOSデバイスの未使用端子の入力レベルは固定してください。

未使用端子入力については、CMOSデバイスの入力に何も接続しない状態で動作させるのではなく、プルアップかプルダウンによって入力レベルを固定してください。また、未使用の入出力端子が出力となる可能性（タイミングは規定しません）を考慮すると、個別に抵抗を介して V_{DD} または GND に接続することが有効です。

資料中に「未使用端子の処理」について記載のある製品については、その内容を守ってください。

静電気対策

MOSデバイス取り扱いの際は静電気防止を心がけてください。

MOSデバイスは強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、当社が出荷梱包に使用している導電性のトレーやマガジン・ケース、または導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを施してください。プラスチック板上に放置したり、端子を触ったりしないでください。

また、MOSデバイスを実装したボードについても同様の扱いをしてください。

初期化以前の状態

電源投入時、MOSデバイスの初期状態は不定です。

電源投入時の端子の出力状態や入出力設定、レジスタ内容などは保証しておりません。ただし、リセット動作やモード設定で定義している項目については、これらの動作ののちに保証の対象となります。

リセット機能を持つデバイスの電源投入後は、まずリセット動作を実行してください。

電源投入切断順序

内部動作および外部インタフェースで異なる電源を使用するデバイスの場合、原則として内部電源を投入した後に外部電源を投入してください。切断の際には、原則として外部電源を切断した後に内部電源を切断してください。逆の電源投入切断順により、内部素子に過電圧が印加され、誤動作を引き起こしたり、異常電流が流れ内部素子を劣化させたりする場合があります。

資料中に「電源投入切断シーケンス」についての記載のある製品については、その内容を守ってください。

電源OFF時における入力信号

当該デバイスの電源がOFF状態の時に、入力信号や入出力プルアップ電源を入れないでください。入力信号や入出力プルアップ電源からの電流注入により、誤動作を引き起こしたり、異常電流が流れ内部素子を劣化させたりする場合があります。

資料中に「電源OFF時における入力信号」についての記載のある製品については、その内容を守ってください。

- 本資料に記載されている内容は2005年5月現在のもので、今後、予告なく変更することがあります。量産設計の際には最新の個別データ・シート等をご参照ください。
- 文書による当社の事前の承諾なしに本資料の転載複製を禁じます。当社は、本資料の誤りに関し、一切その責を負いません。
- 当社は、本資料に記載された当社製品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、一切その責を負いません。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。
- 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、一切その責を負いません。
- 当社は、当社製品の品質、信頼性の向上に努めておりますが、当社製品の不具合が完全に発生しないことを保証するものではありません。当社製品の不具合により生じた生命、身体および財産に対する損害の危険を最小限度にするために、冗長設計、延焼対策設計、誤動作防止設計等安全設計を行ってください。
- 当社は、当社製品の品質水準を「標準水準」、「特別水準」およびお客様に品質保証プログラムを指定していただく「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。

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

特別水準：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

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、生命維持のための装置またはシステム等

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。意図されていない用途で当社製品の使用をお客様が希望する場合には、事前に当社販売窓口までお問い合わせください。

(注)

- (1) 本事項において使用されている「当社」とは、NECエレクトロニクス株式会社およびNEC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う。
- (2) 本事項において使用されている「当社製品」とは、(1)において定義された当社の開発、製造製品をいう。

はじめに

対象者 このアプリケーション・ノートは、 μ PD78F0714の機能を理解し、それらを使用した応用システムを設計するユーザを対象とします。対象製品を次に示します。

・ μ PD78F0714

目的 このアプリケーション・ノートでは、 μ PD78F0714のタイマ/カウンタ機能のシステム例としてPWM出力、A/Dコンバータ入力を使用したV/f制御による2相正弦波駆動方式のインバータ制御をユーザに理解していただくことを目的としています。

構成 このアプリケーション・ノートは大きく分けて次の内容で構成しています。

・制御方式
・ハードウェア構成
・ソフトウェア構成
・プログラム・リスト

読み方 このマニュアルの読者には、電気、論理回路、およびマイクロコンピュータに関する一般知識を必要とします。

ハードウェア機能の詳細（特にレジスタ機能とその設定方法など）、および電気的特性を知りたいとき

別冊の μ PD78F0714 **ユーザズ・マニュアル** (U16928J) を参照してください。

命令機能の詳細を理解しようとするとき

別冊の78K/0シリーズ **ユーザズ・マニュアル 命令編** (U12326J) を参照してください。

- 凡 例** データ表記の重み：左が上位桁，右が下位桁
 アクティブ・ロウの表記： \overline{xxx} （端子，信号名称に上線）
 メモリ・マップのアドレス：上部-上位，下部-下位
 注：本文中に付けた注の説明
 注意：気を付けて読んでいただきたい内容
 備考：本文の補足説明
 数の表記：2進数 ... xxxxまたはxxxxB
 10進数... xxxx
 16進数... xxxxH
 2のべき数を示す接頭語（アドレス空間，メモリ容量）：
 K（キロ） ... $2^{10} = 1024$
 M（メガ） ... $2^{20} = 1024^2$
 G（ギガ） ... $2^{30} = 1024^3$
 データ・タイプ：ワード ... 32ビット
 ハーフワード ... 16ビット
 バイト ... 8ビット

関連資料 関連資料は暫定版の場合がありますが，この資料では「暫定」の表示をしておりません。あらかじめご了承ください。

デバイスの関連資料

資料名	資料番号	
	和 文	英 文
μ PD78F0714 ユーザーズ・マニュアル	U16928J	U16928E
78K/0シリーズ ユーザーズ・マニュアル 命令編	U12326J	U12326E
μ PD78F0714によるインバータ制御 アプリケーション・ノート ゼロクロス検出による120度通電方式制御編	U17297J	U17297E
μ PD78F0714による単相インダクション・モータ制御 アプリケーション・ノート V/f制御による2相インバータ正弦波駆動編	このマニュアル	作成中

注意 上記関連資料は予告なしに内容を変更することがあります。設計などには，必ず最新の資料をご使用ください。

開発ツール（ソフトウェア）の資料（ユーザズ・マニュアル）

資料名	資料番号	
	和文	英文
RA78K0 Ver.3.80 アセンブラ・パッケージ	操作編	U17199J U17199E
	言語編	U17198J U17198E
	構造化アセンブリ言語編	U17197J U17197E
CC78K0 Ver.3.70 Cコンパイラ	操作編	U17201J U17201E
	言語編	U17200J U17200E
SM+ システム・シミュレータ	操作編	U17246J U17246E
	ユーザ・オープン・インタフェース編	U17247J U17247E
ID78K0-QB Ver.2.81 統合デバッグ	操作編	U16996J U16996E
PM plus Ver.5.20		U16934J U16934E

開発ツール（ハードウェア）の資料（ユーザズ・マニュアル）

資料名	資料番号	
	和文	英文
QB-78K0KX1H インサーキット・エミュレータ	U17081J	U17081E

フラッシュ・メモリ書き込み用の資料

資料名	資料番号	
	和文	英文
PG-FP3 フラッシュ・メモリ・プログラマ ユーザズ・マニュアル	U13502J	U13502E
PG-FP4 フラッシュ・メモリ・プログラマ ユーザズ・マニュアル	U15260J	U15260E

その他の資料

資料名	資料番号	
	和文	英文
SEMICONDUCTOR SELECTION GUIDE - Products and Packages -	X13769X	
半導体デバイス 実装マニュアル	注	
NEC半導体デバイスの品質水準	C11531J	C11531E
NEC半導体デバイスの信頼性品質管理	C10983J	C10983E
静電気放電（ESD）破壊対策ガイド	C11892J	C11892E
半導体 品質 / 信頼性ハンドブック	C12769J	-
マイクロコンピュータ関連製品ガイド 社外メーカ編	U11416J	-

注 「半導体デバイス実装マニュアル」のホーム・ページ参照

和文：<http://www.necel.com/pkg/ja/jissou/index.html>

英文：<http://www.necel.com/pkg/en/mount/index.html>

注意 上記関連資料は予告なしに内容を変更することがあります。設計などには、必ず最新の資料をご使用ください。

目 次

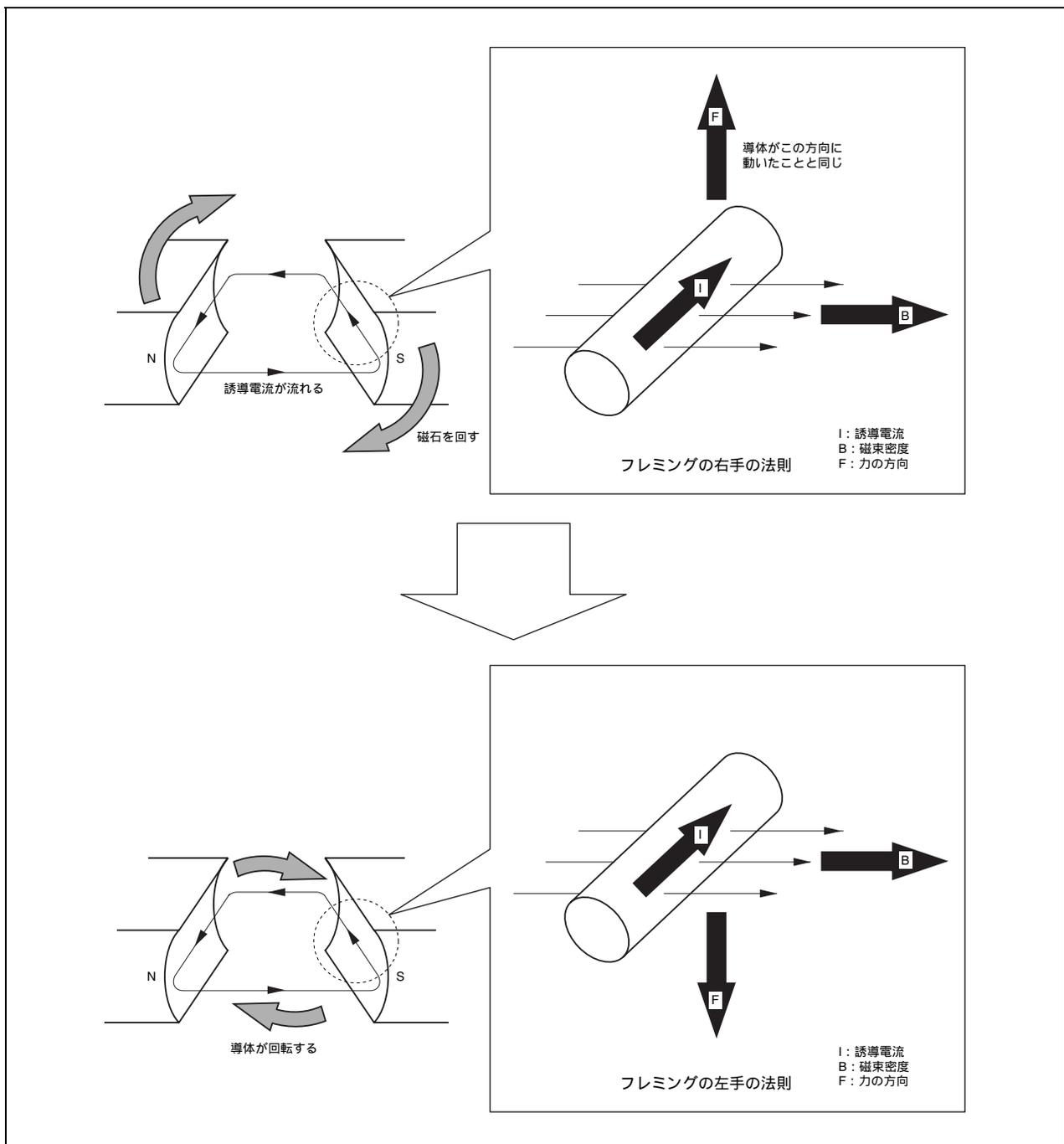
第1章 制御方式 ...	10
1.1 インダクション・モータ制御の概要 ...	10
1.2 単相インダクション・モータ制御の概要 ...	12
第2章 ハードウェア構成 ...	15
2.1 構 成 ...	15
第3章 ソフトウェア構成 ...	17
3.1 周辺I/O ...	17
3.2 ソフトウェア処理構造 ...	18
3.3 フロー・チャート ...	19
3.3.1 メイン処理 ...	19
3.3.2 モータ制御計算処理 ...	20
3.3.3 PWM割り込み処理 ...	21
3.3.4 A/Dコンバータ割り込み処理 ...	22
3.3.5 ハードウェア初期化 ...	24
3.3.6 ソフトウェア初期化 ...	25
3.4 テーブル類 ...	26
3.5 定数定義 ...	27
第4章 プログラム・リスト ...	28
4.1 プログラム・リスト (μ PD78F0714用) ...	28
4.1.1 プログラム処理定義 ...	28
4.1.2 ヘッダ・ファイル ...	28
4.1.3 定数定義 ...	28
4.1.4 シンボル定義 ...	30
4.1.5 メイン処理関数 ...	31
4.1.6 モータ制御計算処理関数 ...	32
4.1.7 PWM割り込み処理関数 ...	34
4.1.8 A/Dコンバータ割り込み処理関数 ...	35
4.1.9 LED表示関数 ...	36
4.1.10 ハードウェア初期化処理関数 ...	38
4.1.11 ソフトウェア初期化処理関数 ...	39

第1章 制御方式

1.1 インダクション・モータ制御の概要

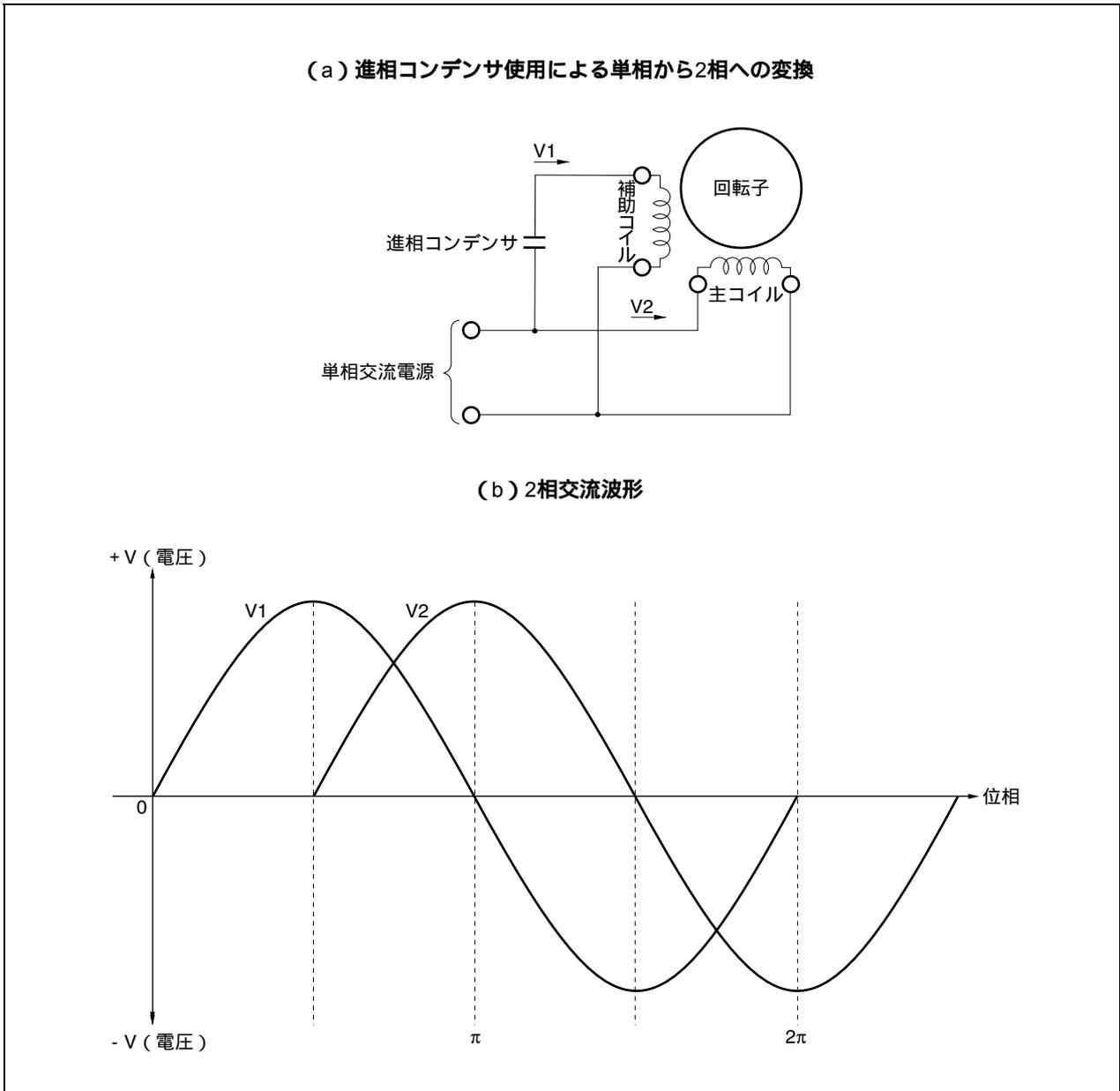
インダクション・モータの基本回転原理は、閉じた導体の周りの磁石を回転させると導体に誘導電流が流れ、誘導電流が流れたことにより、導体に力が働き回転します（図1-1参照）。

図1-1 インダクション・モータの回転原理



単相交流電源を使用するインダクション・モータは、単相インダクション・モータと呼ばれますが、単相交流電源では回転磁界ができないため、実際は進相コンデンサを使用して90°位相をずらし、2相交流になっています（図1-2参照）

図1-2 単相インダクション・モータ



1.2 単相インダクション・モータ制御の概要

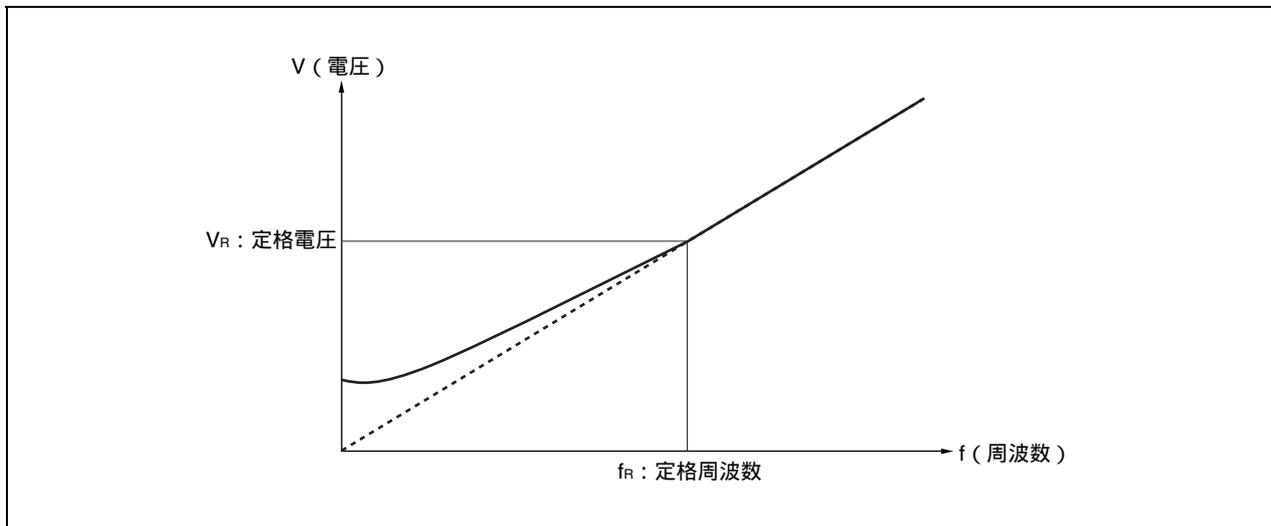
μ PD78F0714を使用した単相インダクション・モータ制御について説明します。

1.1 インダクション・モータ制御の概要で説明したように、単相インダクション・モータは、実際には 90° 位相をずらした2相交流で回転しているため、2相正弦波のインバータ制御をすることで回転数を制御します。

制御方法は、3相PWMインバータを使用し、V/f制御による2相の正弦波駆動を行います。

V/f制御の原理は電圧（V）と周波数（f）の比を保つことで周波数に依存しない回転磁界が生じ、このためトルクを粗一定に保つことができます。しかし、V/f比を一定に保っても低周波数領域では、回転磁界を一定に保つことができない（図1-3の直線（破線）参照）ため、電圧降下分を図1-3の曲線（実線）のように補填することで回転磁界を一定に保ちます。このアプリケーションでの制御では周波数をいくつかの領域に分けることでV/f比を線形として扱う方式を採用しています。

図1-3 V/f比の概念図



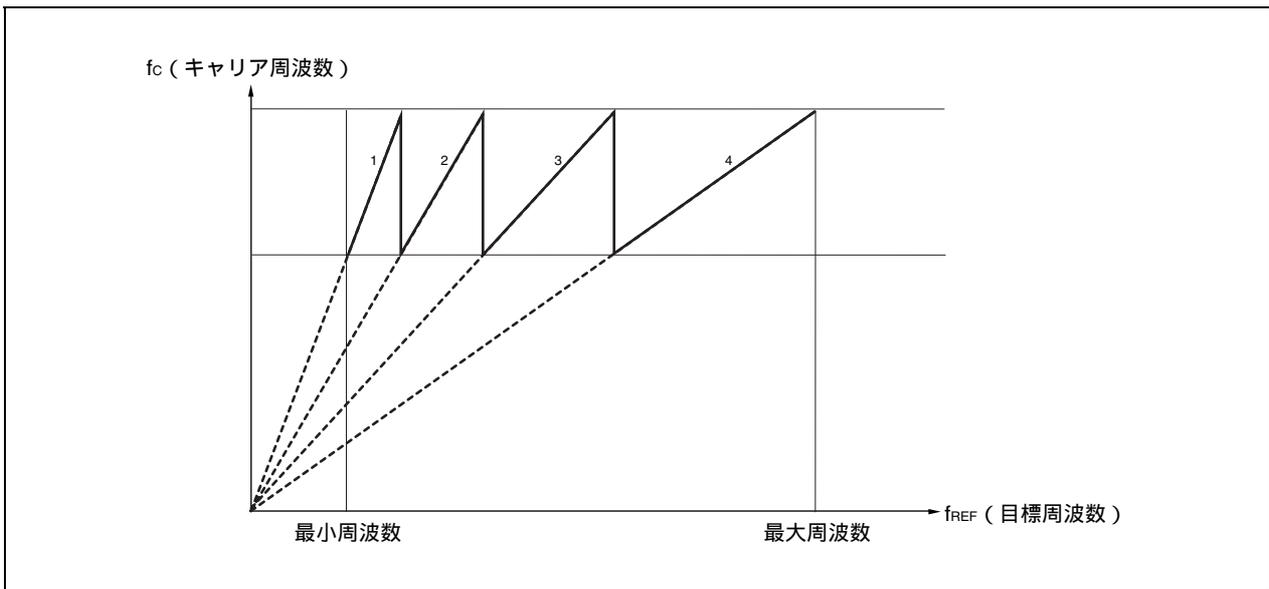
次にキャリア周波数（ f_c ）の選択方法について説明します。

目標周波数（ f_{REF} ）に対し、 $f_c = k \cdot f_{REF}$ とし、比例定数（ k ）はシステムにより値を決定します。

通常、キャリア波と正弦波の相対的位置が同じになるように整数（このアプリケーションでのシステムでは12の倍数）が選択されます。

比例定数（ k ）は、その値が小さくなるにつれて、基準正弦波がより多くのノイズ（正弦波からのずれ）を含むため、十分に大きな値とする必要がありますが、一方でその値が大きくなるにつれてインバータのスイッチング損失も増大します。したがって、キャリア周波数（ f_c ）の値に上限と下限を設けて目標周波数（ f_{REF} ）領域ごとに比例定数 k を決定します（図1-4参照）。

図1 - 4 回転周波数とキャリア周波数の関係



比例定数 が決定すれば目標周波数（ f_{REF} ）に対しキャリア周波数（ f_c ）が決定し、キャリア波のパルス幅が決定できます。

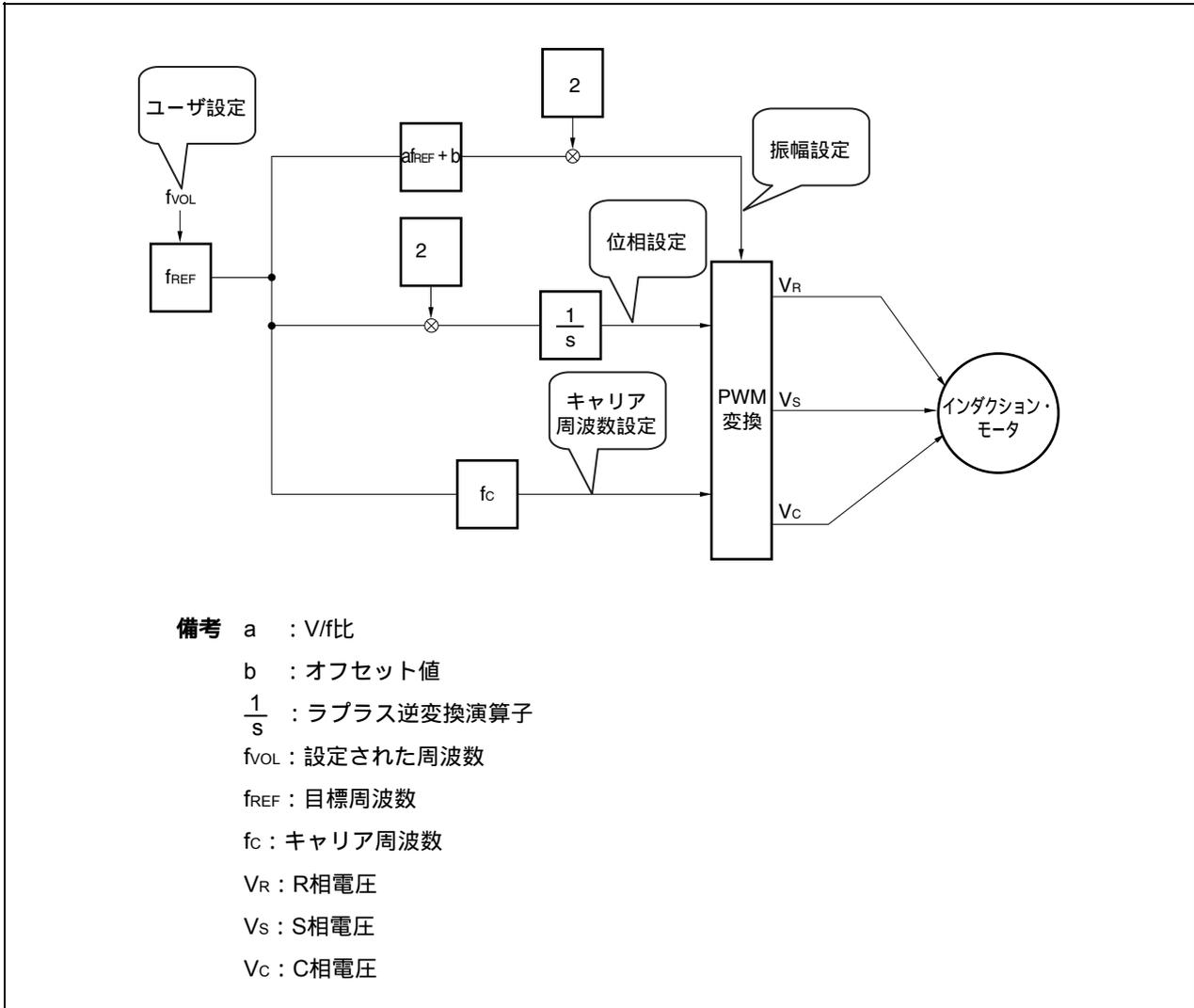
1/ f_{REF} 積分することで位相を算出することができ、PWM設定値は次の式で求められます。

$$V = \sqrt{2} (af_{REF} + b) \sin \left(2 \pi f_{REF} \left(\frac{1}{f_{REF}} \right) dt \right)$$

備考 a : V/f比
 b : オフセット値
 f_{REF} : 目標周波数

次に制御ブロック図を示します。

図1-5 制御ブロック図



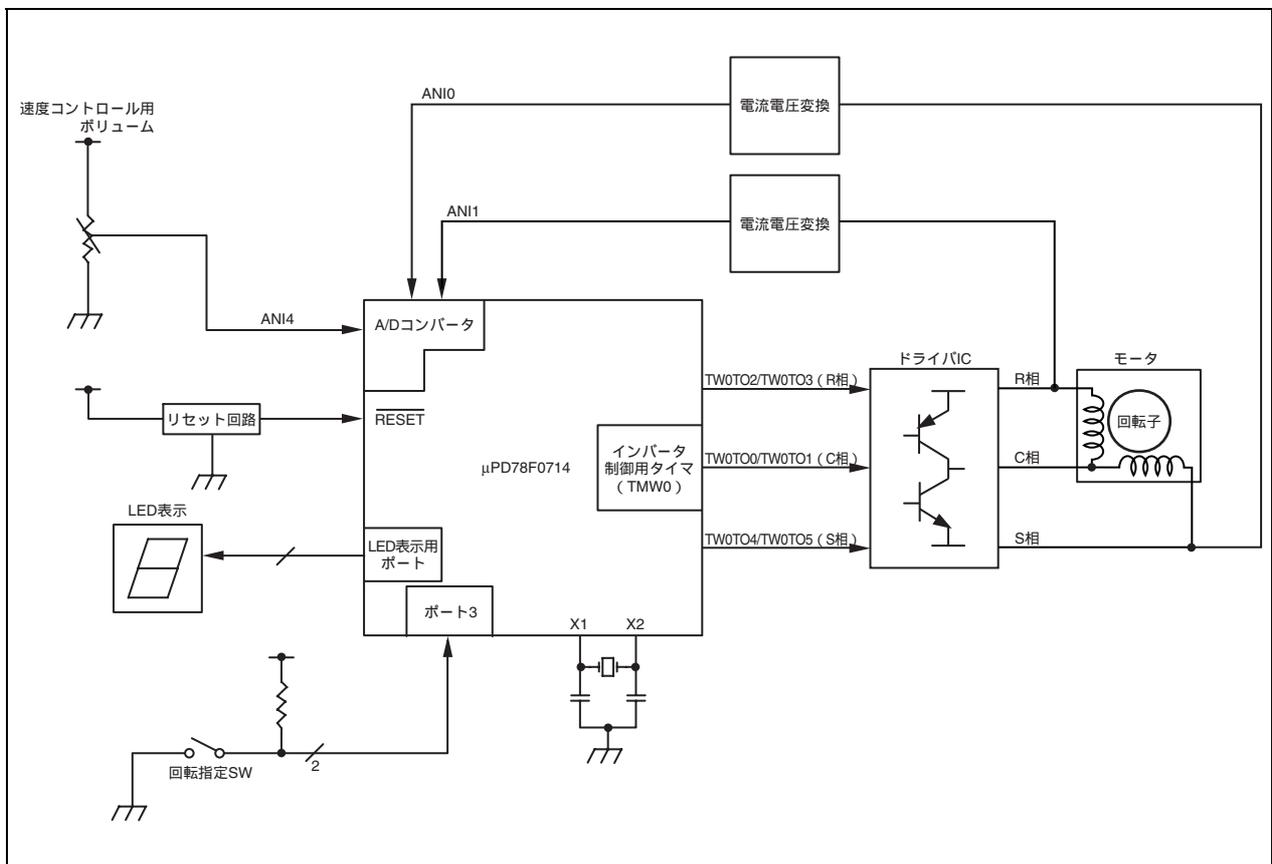
第2章 ハードウェア構成

ハードウェア構成について説明します。

2.1 構成

このアプリケーションでのシステムの主な機能を示します。このアプリケーションでのシステムは、電源投入後、回転指定SWを押すと、単相インダクション・モータが回転を開始します。

図2 - 1 全体のシステム構成



(1) 速度コントロール用ボリューム

モータの回転数の増減設定ボリューム

(2) 回転指定SW

CWおよびSTOP用スイッチ

(3) LED表示

回転数，エラーの表示LED

(4) ドライバIC

モータ駆動用ドライバ

(5) 電流電圧変換回路

モータの駆動電流を電圧に変換し，過電流を検出用として使用

第3章 ソフトウェア構成

ソフトウェア構成について説明します。

3.1 周辺I/O

このアプリケーションでのシステムでは、次のように周辺I/Oを使用しています。

表3 - 1 使用周辺I/O一覧

機能	周辺I/O機能名 (μ PD78F0714)
インバータ・タイマ	タイマW0 (TMW0)
過電流検出	ANI0, ANI1
設定スピード (ボリューム)	ANI4
CWキー入力	P30
STOPキー入力	P33
LED出力	P47-P40, P57-P50

(1) 周辺I/O機能の説明

(a) インバータ・タイマ

インバータ・タイマを使用してPWM波形を出力します。

このアプリケーションでのソフトウェアでは次のような設定になっています。

- ・インバータ・タイマ出力：ロウ・アクティブ
- ・デッド・タイム：3 μs
- ・対称三角波モード
- ・キャリア周波数は、設定スピードにより次のように設定しています。

設定スピード	キャリア周波数
~ 1200 rpm	6.5 kHz
1201 ~ 2400 rpm	13 kHz
2401 ~ 3600 rpm	19.5 kHz
3601 ~ 4800 rpm	26 kHz

(b) 過電流検出

ANI0, ANI1を使用して過電流の検出を行います。

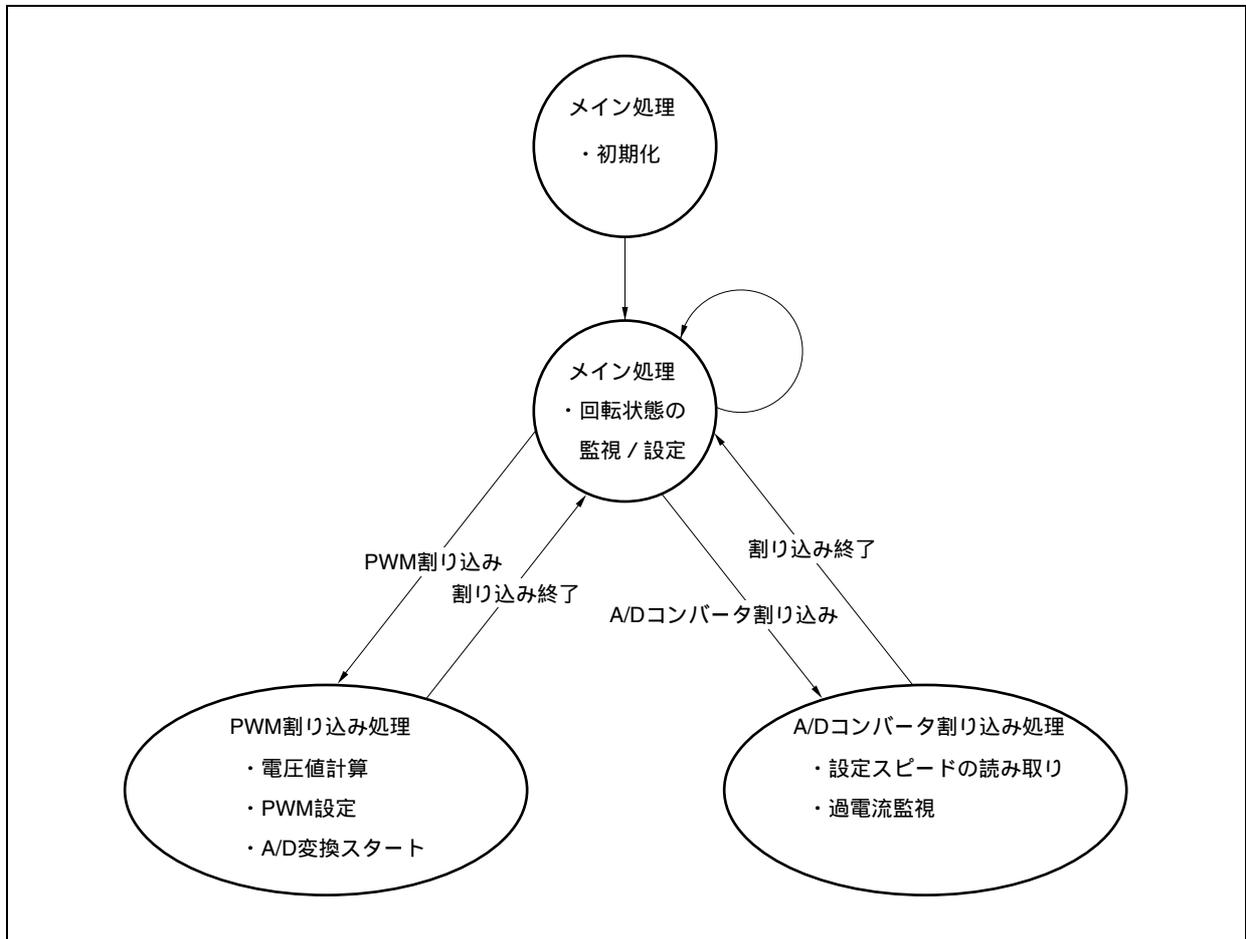
(c) 設定スピード (ボリューム) 入力

ANI4を使用してスピードの設定を行います。

3.2 ソフトウェア処理構造

次にソフトウェア処理構造を示します。

図3 - 1 ソフトウェア処理構造



このアプリケーションでのソフトウェアは、主に次の3つの処理で構成されています。

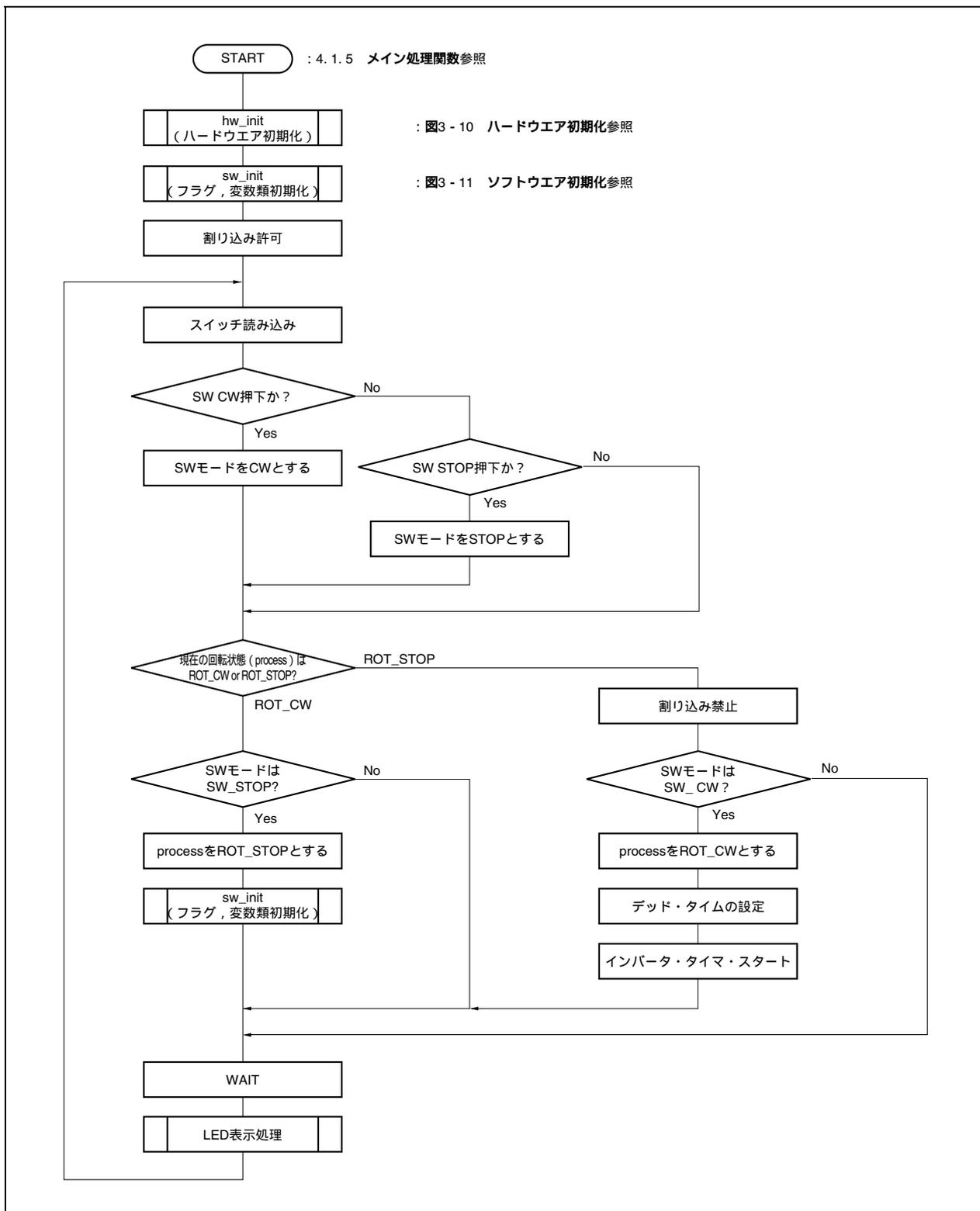
- ・メイン処理
 - 初期化，回転状態（CW / STOP）を設定します。
- ・PWM割り込み処理
 - 各相（R, S相）の電圧値をインバータ・タイマに設定します。
- ・A/Dコンバータ割り込み処理
 - 設定スピードの読み取り（ANI4），過電流監視（ANI0, ANI1）を行います。

3.3 フロー・チャート

3.3.1 メイン処理

図3 - 2にメイン処理についてのフローを示します。

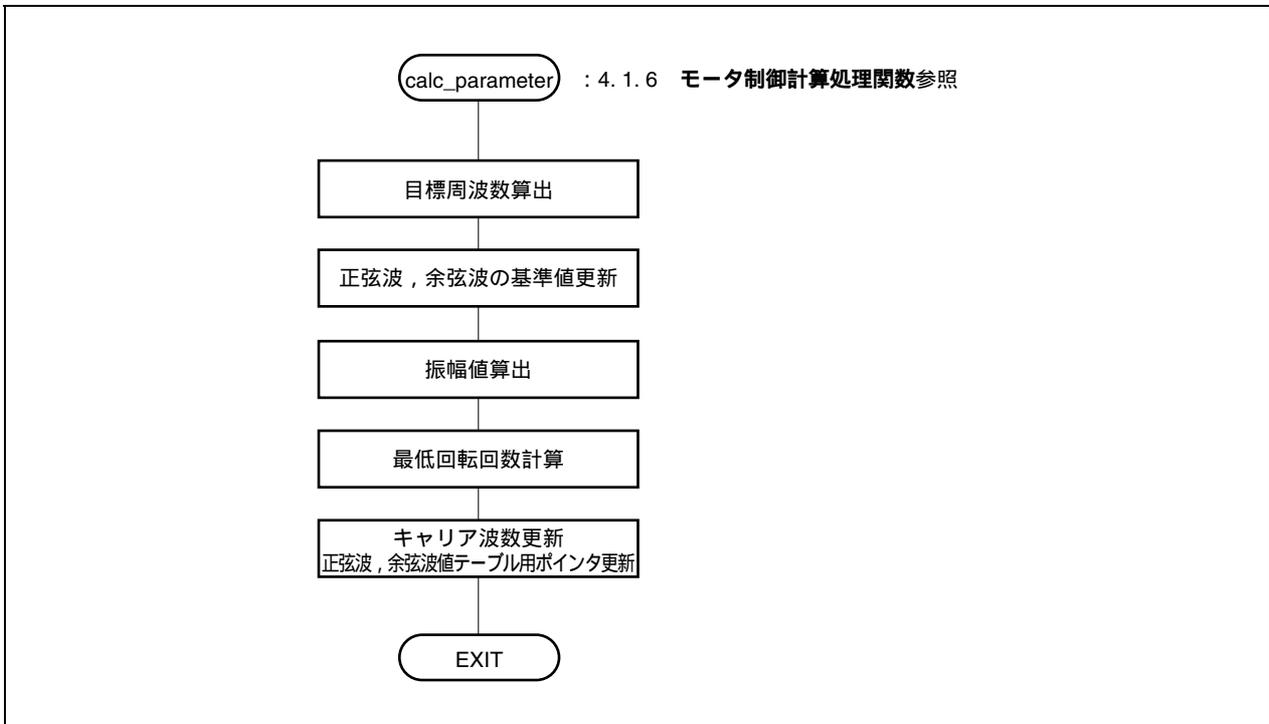
図3 - 2 メイン処理



3.3.2 モータ制御計算処理

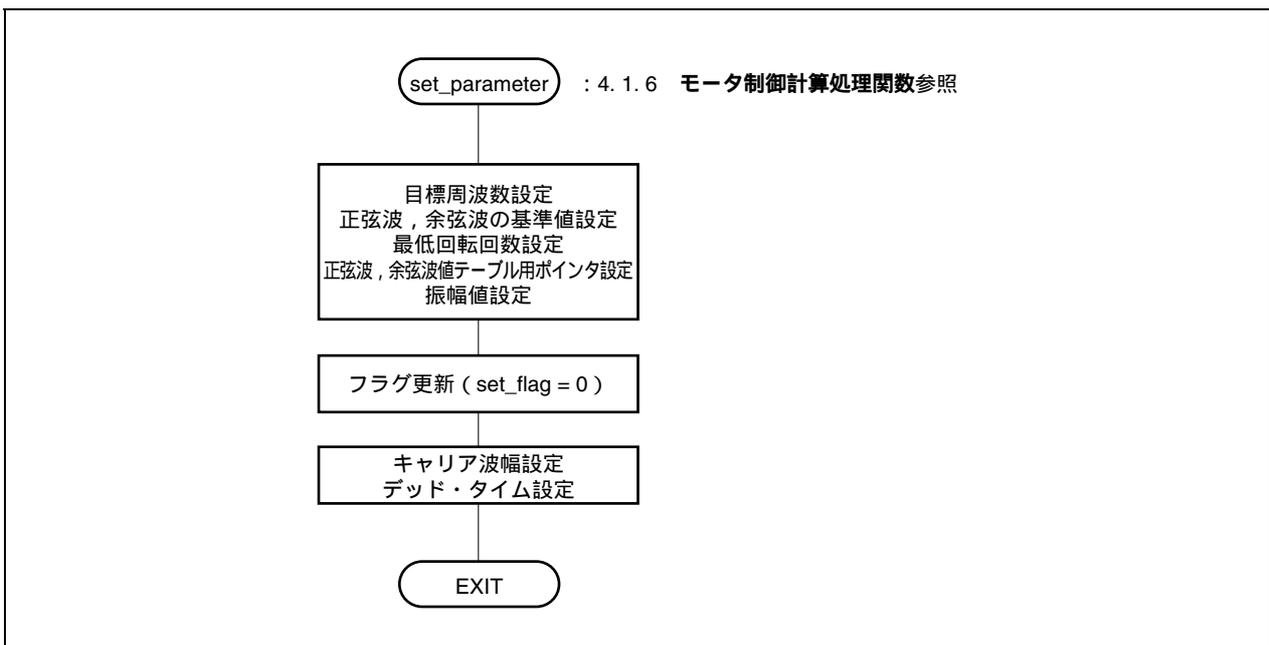
(1) calc_parameter関数

図3 - 3 calc_parameter関数



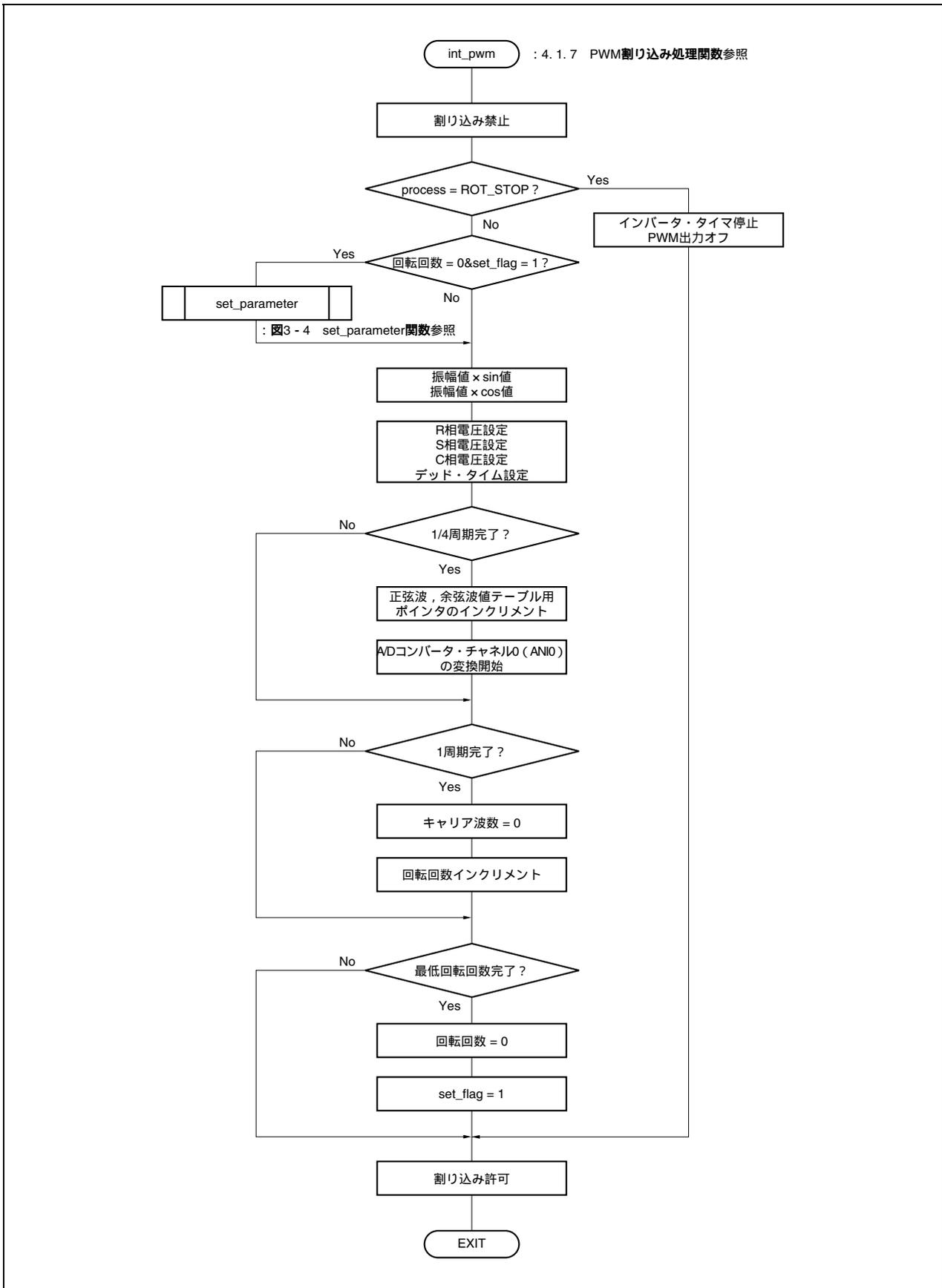
(2) set_parameter関数

図3 - 4 set_parameter関数



3.3.3 PWM割り込み処理

図3 - 5 PWM割り込み処理



3.3.4 A/Dコンバータ割り込み処理

図3-6 A/Dコンバータ割り込み処理

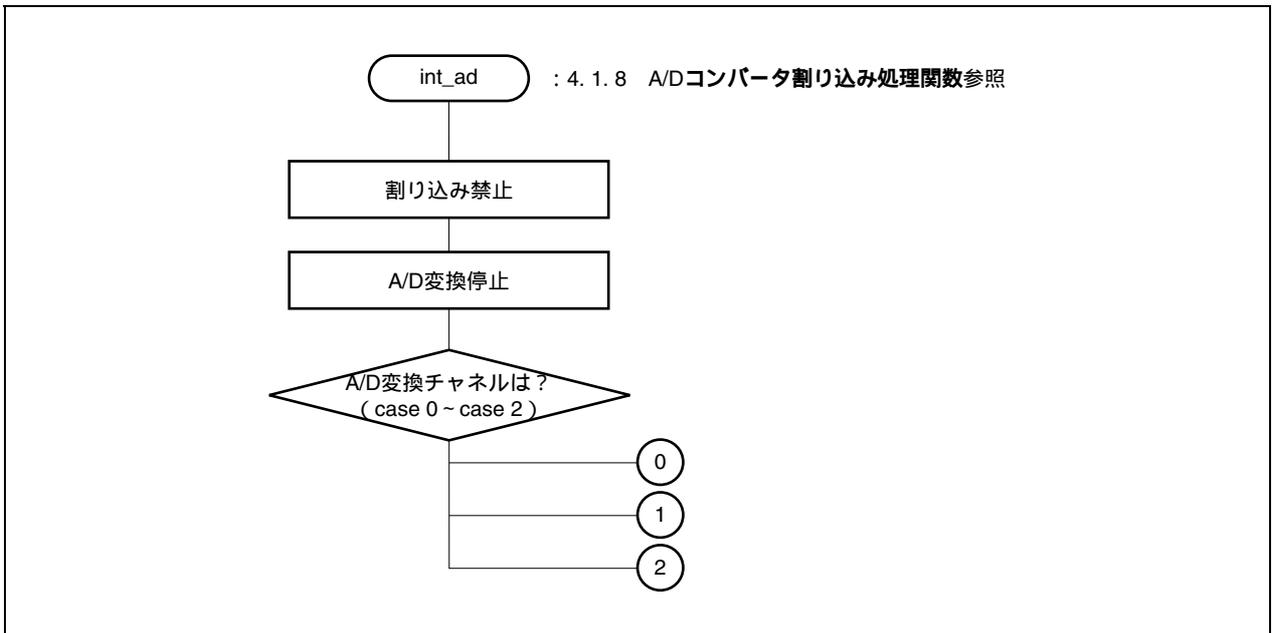


図3-7 case 0 (A/Dコンバータ・チャンネル1割り込み処理)

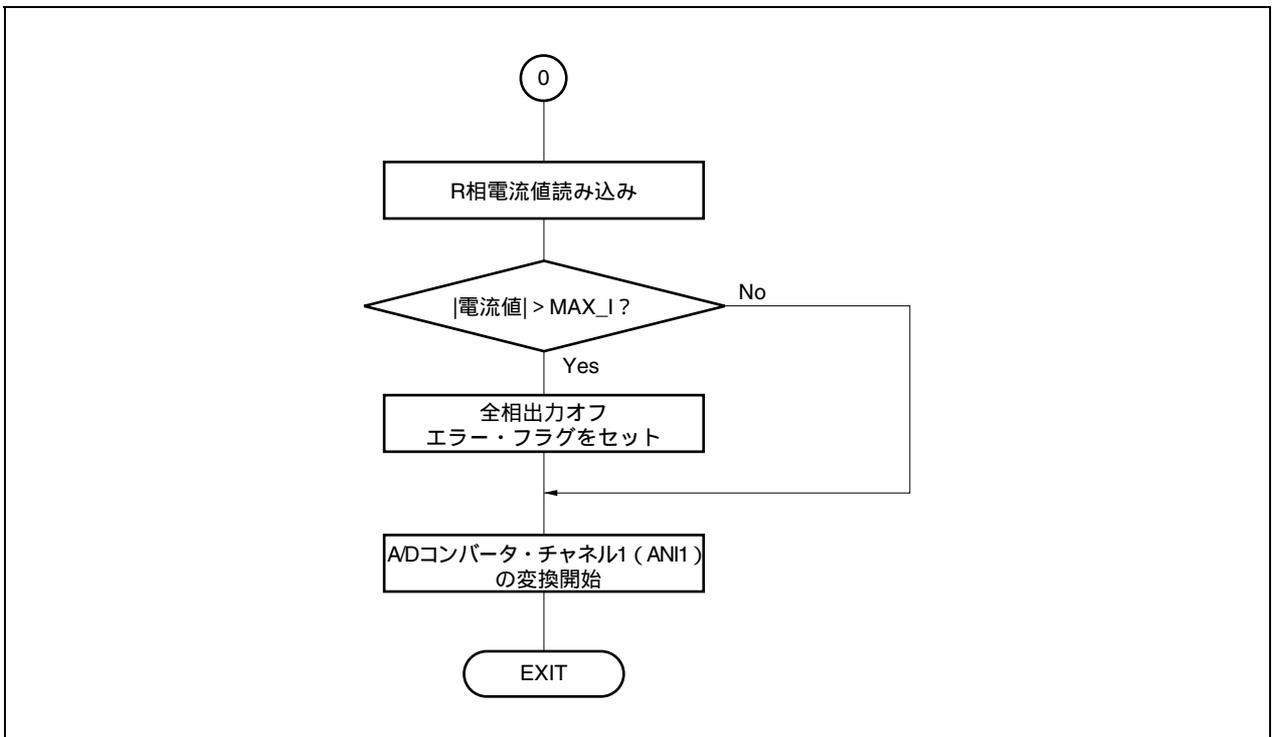


図3 - 8 case 1 (A/Dコンバータ・チャンネル4割り込み処理)

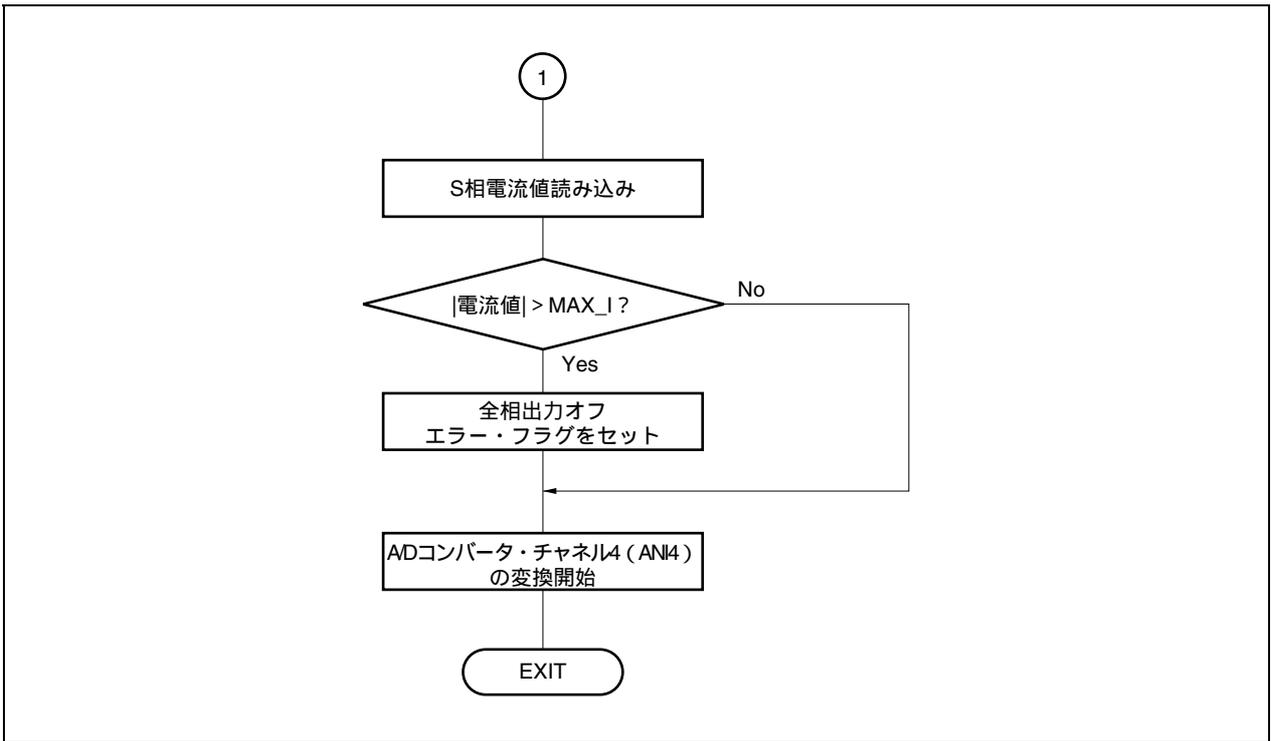
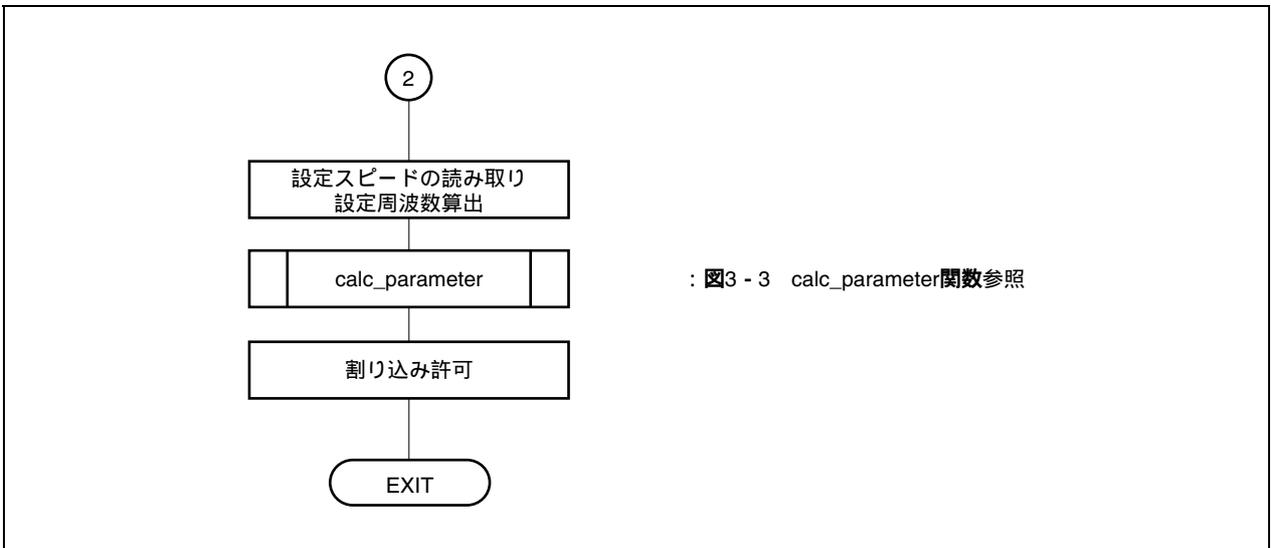
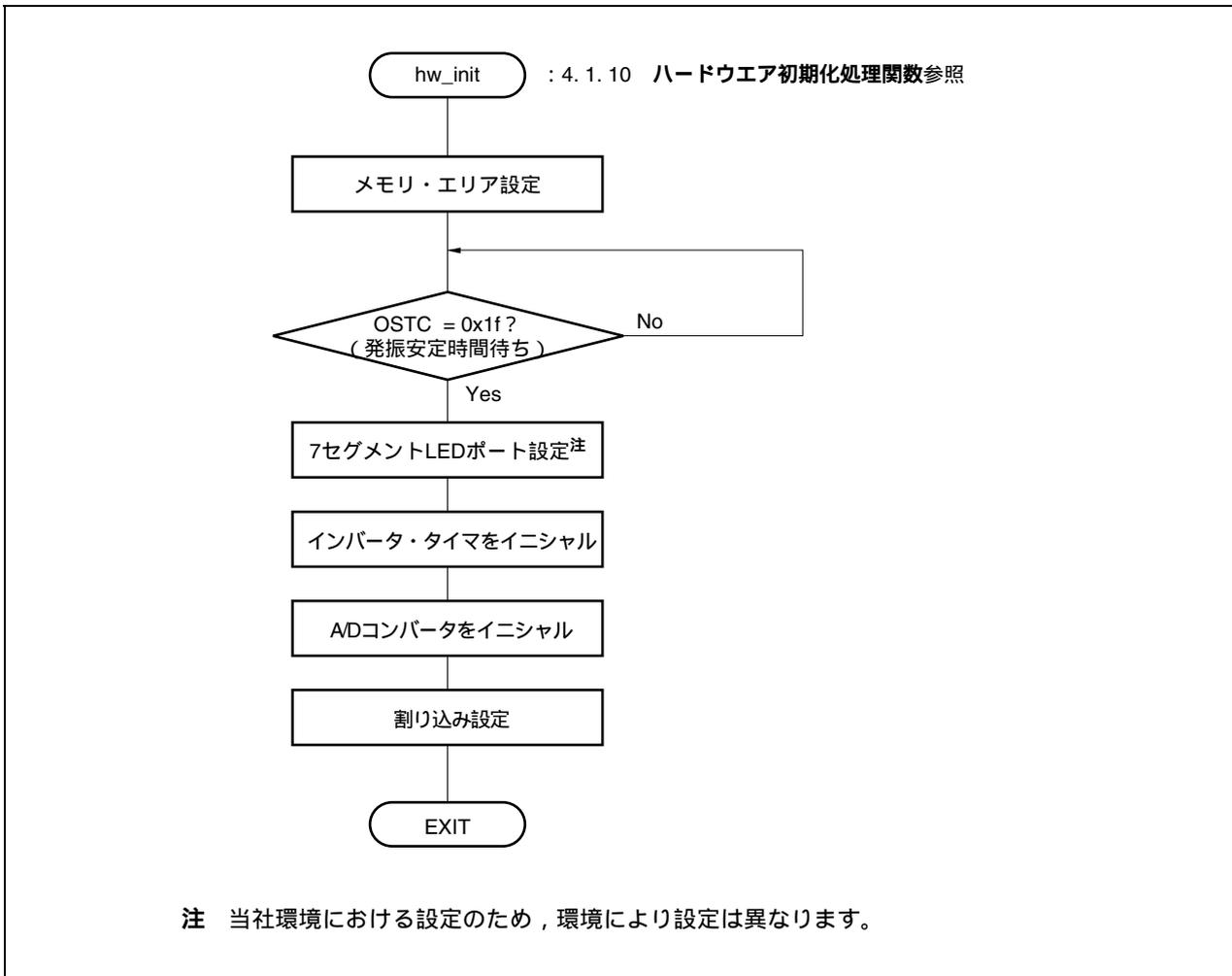


図3 - 9 case 2 (モータ制御計算処理)



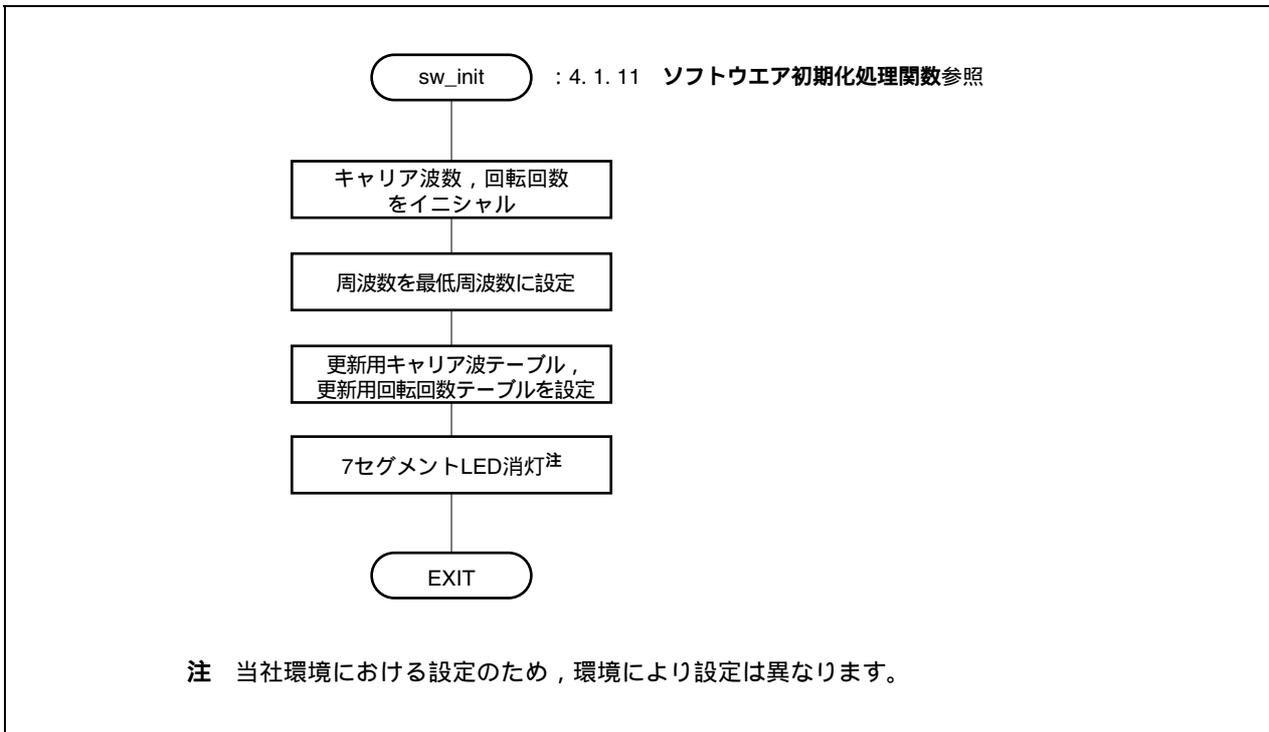
3.3.5 ハードウェア初期化

図3 - 10 ハードウェア初期化



3.3.6 ソフトウェア初期化

図3 - 11 ソフトウェア初期化



3.4 テーブル類

(1) キャリア波数テーブル・データ

キャリア波数を設定するテーブルです。

```
unsigned short set_carrier[ 64 ] = {191,180,171,162,310,295,283,271,260,250,
                                     241,232,224,217,210,203,197,191,186,180,
                                     175,171,166,162,238,232,227,221,217,212,
                                     207,203,199,195,191,187,184,180,177,174,
                                     171,168,165,162,213,210,206,203,200,197,
                                     194,191,188,186,183,180,178,175,173,171,
                                     169,166,164,162}
```

(2) 正弦波値テーブル・データ

$\sin\theta$ 値を設定するテーブルです。 θ は $0^\circ \sim 90^\circ$ (sin値を512倍)です。

```
unsigned short sinwt[ 96 ] = {0, 8, 16, 25, 33, 41,50, 58, 66, 75, 83, 91, 99,108,116,
                               124,132,140,148,156,164,172,180,188,195,203,211,
                               218,226,233,241,248,256,263,270,277,284,291,298,
                               304,311,318,324,331,337,343,349,356,362,367,373,
                               379,384,390,395,401,406,411,416,421,425,430,434,
                               439,443,447,451,455,459,462,466,469,473,476,479,
                               482,484,487,489,492,494,496,498,500,502,503,505,
                               506,507,508,509,510,510,511,511,511}
```

(3) 余弦波値テーブル・データ

$\cos\theta$ 値を設定するテーブルです。 θ は $0^\circ \sim 90^\circ$ (cos値を512倍)です。

```
unsigned short coswt[ 96 ] = {512,511,511,511,510,510,509,508,507,506,505,503,
                               502,500,498,496,494,492,489,487,484,482,479,476,
                               473,469,466,462,459,455,451,447,443,439,434,430,
                               425,421,416,411,406,401,395,390,384,379,373,367,
                               362,356,349,343,337,331,324,318,311,304,298,291,
                               284,277,270,263,255,248,241,233,226,218,211,203,
                               195,188,180,172,164,156,148,140,132,124,116,108,
                               99,91,83,75,66,58,50,41,33,25,16,8}
```

3.5 定数定義

このアプリケーションでのシステムで使用する主な定数を示します。

シンボル	用途	値
F_INV1	キャリア波の範囲定義定数	21
F_INV2	キャリア波の範囲定義定数	41
F_INV3	キャリア波の範囲定義定数	61
FCBYF1	1周期当たりのキャリア波数： ~ 20 Hz	384
FCBYF2	1周期当たりのキャリア波数： 21 Hz ~ 40 Hz	192
FCBYF3	1周期当たりのキャリア波数： 41 Hz ~ 60 Hz	128
FCBYF4	1周期当たりのキャリア波数： 61 Hz ~ 80 Hz	96
TERM	キャリア波数/4	FCBYF1/4
VBYF_RATE	V/f比 × 2 × 1.4142	3
VBYF_OFFSET	V/f比 × 2 × 1.4142	55
NUM_POLE	極対数	2
MIN_FREQ	設定周波数最小値	17
MAX_FREQ	設定周波数最大値	80
WAIT	ウェイト	10
MAX_I	電流MAX.値	800
ADM_DEF	A/D変換停止の設定, A/D変換スピードの設定, A/D変換待機モードの設定	09H
ADM_START	A/D変換スタート	89H
ADM_STOP	A/D変換停止	09H

第4章 プログラム・リスト

4.1 プログラム・リスト (μ PD78F0714用)

4.1.1 プログラム処理定義

```
#pragma sfr
#pragma EI
#pragma DI
#pragma INTERRUPT INTTWOUD      int_pwm rb1
#pragma INTERRUPT INTAD        int_ad  rb2
```

4.1.2 ヘッダ・ファイル

```
/*
ヘッダ・ファイル
*/
#include <stdlib.h>
#include "table.h" /* sin値,cos値,キャリア波幅テーブル */
```

4.1.3 定数定義

```
/*
外部I/O定義
*/
#define BASE_IO      0xc20000

#define WRESET      0x0d /* Watch Dog Timer */
#define SW          0x0e /* SW */
#define DIPSW       0x0f
#define DA1         0x0a
#define DA2         0x0b
#define DA3         0x0c
#define MODE        0x10

#define LED11       0x03 /* LED */
#define LED12       0x02 /* LED */
#define LED13       0x01 /* LED */
#define LED14       0x00 /* LED */
#define LED21       0x05 /* LED */
#define LED22       0x04 /* LED */
#define LED31       0x07 /* LED */
#define LED32       0x06 /* LED */
#define LED41       0x09 /* LED */
#define LED42       0x08 /* LED */
```

```
/*
*****
/*      定数定義
*****
#define F_INV1      21
#define F_INV2      41
#define F_INV3      61

#define FCBYF1      384          /* 1周期当たりのキャリア波数    ~ 20 Hz */
#define FCBYF2      192          /* 1周期当たりのキャリア波数  21 Hz ~ 40 Hz */
#define FCBYF3      128          /* 1周期当たりのキャリア波数  41 Hz ~ 60 Hz */
#define FCBYF4      96           /* 1周期当たりのキャリア波数  61 Hz ~ 80 Hz */

#define TERM        FCBYF1 >> 2      /* キャリア波数/4 */

#define VBYF_RATE   3               /* V/f比 × 2 × 1.4142 */
#define VBYF_OFFSET 55             /* V/f比 × 2 × 1.4142 */

#define DEAD_TIME   0x0a           /* デッド・タイム */

#define NUM_POLE    2               /* 極対数 */
#define MIN_FREQ    17             /* 設定周波数最小値 */
#define MAX_FREQ    80             /* 設定周波数最大値 */

#define WAIT        10             /* ウェイト */
#define MAX_I       800            /* 電流 MAX.値 */

#define LED_1       1               /* LED番号 */
#define LED_2       2               /* LED番号 */
#define LED_3       3               /* LED番号 */
#define LED_4       4               /* LED番号 */

#define LOW         0
#define HIGH        1

#define ADM_DEF     9               /* A/D変換停止, A/D変換スピードの設定 */
                                   /* A/D変換待機モードの設定 */
#define ADM_START   (ADM_DEF | 0x80) /* A/D変換スタート */
#define ADM_STOP    ADM_DEF        /* A/D変換停止 */

#define CE0_ACTIVE  1

/*
*****
/*      フラグ定義
*****
unsigned short  error_flag;          /* エラー・フラグ */
unsigned short  set_flag;           /* パラメータ・セット・フラグ */

#define ERROR1    1                  /* 過電流エラー */
#define ERROR2    2                  /* 速度差エラー */
```

```
#define SW_STOP      0          /* 回転停止 */
#define SW_CW       1          /* 順回転 */
#define SW_CCW      2          /* 逆回転 */

#define ROT_STOP    0          /* 回転停止 */
#define ROT_CW     1          /* 順回転 */
#define ROT_CCW    2          /* 逆回転 */
```

4.1.4 シンボル定義

```
/* *****
/*      変数定義
/* *****
unsigned char   sw_mode;          /* 指定回転 */
unsigned char   process;        /* 回転状態 */

unsigned short  f_ref_temp;      /* 目標周波数 */
unsigned short  offset;         /* sin, cos値基準値, キャリア波周波数設定値/2 */
unsigned char   ptr_rate;
unsigned short  num_pulse;      /* キャリア波数 */
unsigned short  v_amp;         /* 設定電圧振幅値 */
unsigned short  default_rot;    /* 最低回転回数 */
unsigned short  n_f_ref_temp;   /* 目標周波数:更新用 */
unsigned short  n_offset;      /* sin, cos値基準値, キャリア波周波数設定値/2:更新用 */
unsigned char   n_ptr_rate;
unsigned short  n_num_pulse;    /* キャリア波数:更新用 */
unsigned short  n_v_amp;       /* 設定電圧振幅値:更新用 */
unsigned short  n_default_rot; /* 最低回転回数:更新用 */

unsigned short  num_carrier;    /* キャリア波数 */
unsigned short  num_nc;        /* キャリア波カウント数 */
unsigned short  term;          /* 1/4回転回数 */
unsigned short  rot_num;       /* 回転回数 */

signed short    sgn_ph;        /* 位相比用変数 */
signed short    sgn_sin_amp;   /* sin振幅値 */
signed short    sgn_cos_amp;   /* cos振幅値 */

unsigned short  ex_set_value_c;
unsigned short  ex_set_value_r;
unsigned short  ex_set_value_s;

unsigned short  f_ref;         /* 設定周波数 */

unsigned short  v_r;
unsigned short  v_s;
unsigned long   sv;
unsigned long   cv;

const unsigned short led_pat[10] = { 0xfc, 0x60, 0xda, 0xf2, 0x66, 0xb6, 0xbe, 0xe0, 0xfe,
```

```

                                0xe6 }];

extern unsigned short  set_carrier[ 64 ];
extern unsigned short  sinwt[ 96 ];
extern unsigned short  coswt[ 96 ];

```

4.1.5 メイン処理関数

```

/*****
/*      関数定義
*****/
void main( void );                /* メイン */
void calc_parameter( void );      /* 制御パラメータの計算 */
void set_parameter( void );      /* 制御パラメータの更新 */

void hw_init( void );            /* ハードウェア初期化 */
void sw_init( void );           /* ソフトウェア初期化 */
void output_data( unsigned short reg, unsigned short data ); /* I/O出力 */
unsigned short input_data( unsigned short reg ); /* I/O入力 */
void led( unsigned short led_number, unsigned short led_data ); /* LED */

/*****
/*      インダクション・モータV/f制御プログラム・メイン処理
*****/
void main( void )
{
  unsigned short  ic;
  unsigned short  sw_set;

  /* 初期化 */
  hw_init();                /* ハードウェア初期化 */
  sw_init();                /* ソフトウェア初期化 */

  EI();                      /* 割り込み許可 */

  while( 1 ) {

    sw_set = ~input_data( SW ) & 0x07; /* 運転ボタン読み込み */
    if ( sw_set & 0x01 ) {
      sw_mode = SW_CW;
    } else if ( sw_set & 0x02 ) {
      sw_mode = SW_CCW;
    } else if ( sw_set & 0x04 ) {
      sw_mode = SW_STOP;
    }

    switch ( process ) {
      case ROT_STOP:
        DI();
        if ( sw_mode == SW_CW ) {

```

```

        process = ROT_CW;
        TWODTIME = DEAD_TIME;          /* デッド・タイム 3 usec */
        CEO = 1;
    } else if ( sw_mode == SW_CCW ) {
        process = ROT_CW;              /* 単方向のみ */
        TWODTIME = DEAD_TIME;          /* デッド・タイム 3 usec */
        CEO = 1;
    }
    EI();
    break;

case ROT_CW:
    if ( (sw_mode == SW_CCW) || (sw_mode == SW_STOP) ) {
        process = ROT_STOP;
        sw_init();
    }
    break;

case ROT_CCW:
    if ( (sw_mode == SW_CW) || (sw_mode == SW_STOP) ) {
        process = ROT_STOP;
        sw_init();
    }
    break;
}

ic = WAIT;
while(ic--);

/* LED表示 */
led( LED_2, f_ref_temp );             /* LED表示 */

if ( error_flag ) {
    CEO = 0;                           /* PWM全相出力禁止 */
    process = ROT_STOP;
    led( LED_1, 1000 | error_flag );    /* LED表示 */
}
}
}

```

4.1.6 モータ制御計算処理関数

```

/*****
/*      インバータ周波数, キャリア波, 振幅値の更新      */
*****/
void set_parameter( void )
{

/* インバータ周波数更新 17 Hz ~ 80 Hz */
    f_ref_temp = n_f_ref_temp;

```

```

/* sin,cos値オフセット設定 */
    offset = n_offset;                /* キャリア波1/4相当 */
    num_pulse = n_num_pulse;         /* キャリア波数設定 */
    ptr_rate = n_ptr_rate;

    TW0CM3 = offset << 1;
    TW0BFCM3 = offset << 1;

/* 振幅値設定 */
    v_amp = n_v_amp;                 /* V/f比×2 */

/* 最低回転回数設定 */
    default_rot = n_default_rot;

    set_flag = 0;                   /* パラメータ・セット・フラグ・クリア */

    num_carrier= 0;                 /* キャリア波数クリア */
    num_nc = 0;                    /* キャリア波カウント数クリア */
    term = 1;                       /* 1/4周回数クリア */
}

/*****
/*      インバータ周波数, キャリア波, 振幅値の計算      */
*****/
void calc_parameter( void )
{

/* インバータ周波数更新 7 Hz~70 Hz */
    if ( f_ref > n_f_ref_temp ) {
        n_f_ref_temp = n_f_ref_temp + 1;
    } else if ( f_ref < n_f_ref_temp ) {
        n_f_ref_temp = n_f_ref_temp - 1;
    }

    if ( n_f_ref_temp < MIN_FREQ ) n_f_ref_temp = MIN_FREQ;
    if ( n_f_ref_temp > MAX_FREQ ) n_f_ref_temp = MAX_FREQ;

/* sin,cos値オフセット設定 */
    n_offset = set_carrier[n_f_ref_temp - MIN_FREQ]; /* キャリア波1/4相当 */

/* 振幅値設定 */
    n_v_amp = VBYF_RATE * n_f_ref_temp + VBYF_OFFSET; /* V/f比×2 */

/* 最低回転回数設定 */
    n_default_rot = (n_f_ref_temp >> 2) & 0x3fff; /* 周波数の1/4回~約0.25sec */

    if ( n_f_ref_temp < F_INV1 ) { /* ~20 Hz */
        n_num_pulse = FCBYF1; /* キャリア波数設定 */
        n_ptr_rate = 1;
    }
}

```

```

        n_v_amp += 25;
    } else if ( n_f_ref_temp < F_INV2 ) { /* 21 Hz~40 Hz */
        n_num_pulse = FCBYF2; /* キャリア波数設定 */
        n_ptr_rate = 2;
        if ( n_f_ref_temp < 44 ) n_v_amp += 55;
    } else if ( n_f_ref_temp < F_INV3 ) { /* 41 Hz~60 Hz */
        n_num_pulse = FCBYF3; /* キャリア波数設定 */
        n_ptr_rate = 3;
    } else { /* 61 Hz~80 Hz */
        n_num_pulse = FCBYF4; /* キャリア波数設定 */
        n_ptr_rate = 4;
        if ( n_v_amp > 325 ) n_v_amp = 325;
    }
    n_ptr_rate *= NUM_POLE; /* 極対数 */
}

```

4.1.7 PWM割り込み処理関数

```

/*****
/*      PWM割り込み処理
*****/
__interrupt void int_pwm(void)
{
    DI();

    if ( process == ROT_STOP ) { /* 停止 */
        CEO = 0; /* PWM全相出力禁止 */
    } else {

        if ( (rot_num == 0) && (set_flag == 1) ) set_parameter(); /* 制御パラメータの更新 */

        sv = sinwt[ num_carrier ];
        cv = coswt[ num_carrier ];
        v_r = (unsigned short)((unsigned long)v_amp * sv) >> 10;
        v_s = (unsigned short)((unsigned long)v_amp * cv) >> 10;

        if ( ( v_r < offset ) || ( v_s < offset ) ) {
            TW0BFCM0 = offset; /* C相：一定値 */
            TW0BFCM1 = offset + sgn_sin_amp * v_r; /* R相：基本正弦波 */
            TW0BFCM2 = offset + sgn_cos_amp * v_s; /* S相： /2遅延 */

            ex_set_value_c = offset; /* C相：一定値 */
            ex_set_value_r = offset + sgn_sin_amp * v_r; /* R相：基本正弦波 */
            ex_set_value_s = offset + sgn_cos_amp * v_s; /* S相： /2遅延 */
        } else {

            TW0BFCM0 = ex_set_value_c; /* C相：一定値 */
            TW0BFCM1 = ex_set_value_r; /* R相：基本正弦波 */
            TW0BFCM2 = ex_set_value_s; /* S相： /2遅延 */
        }
    }
}

```

```

    }

    if ( (num_nc + ptr_rate) >= TERM ) { /* 1/4周期 */
        sgn_sin_amp = sgn_ph * sgn_sin_amp;
        sgn_cos_amp = - sgn_ph * sgn_cos_amp;
        sgn_ph = - sgn_ph;
        term++; /* 1/4周 回数インクリメント */
        ADS = 0x00; /* ANI0選択 */
        ADM = ADM_START; /* A/D変換スタート */
        num_nc = 0;
    } else {
        num_carrier += sgn_ph * ptr_rate; /* キャリア波番号インクリメント */
        num_nc += ptr_rate; /* キャリア波カウント数インクリメント */
    }

    if ( term > 4 ) { /* 1周期以下 */
        num_carrier = 0; /* キャリア波数クリア */
        num_nc = 0; /* キャリア波カウント数クリア */
        term = 1; /* 1/4周 回数クリア */
        rot_num++; /* 回転回数インクリメント */
    }

    if ( rot_num == default_rot ) {
        rot_num = 0; /* 最低回転回数完了 カウンタ・クリア */
        set_flag = 1; /* パラメータ・セット・フラグ */
    }
}

IF1L &= ~0x04;
EI();
}

```

4.1.8 A/Dコンバータ割り込み処理関数

```

/*****
/*      設定周波数A/Dコンバータ割り込み処理      */
*****/
__interrupt void int_ad(void)
{
    signed short   iua; /* 電流値 */
    signed short   iva; /* 電流値 */
    unsigned short vol; /* 設定値 */

    DI();

    ADM = ADM_STOP;
    switch( ADS & 0x07 ) {
        case 0x00:
            iua = ( ( ADCR >> 6 ) & 0x3ff ) - 0x200;

```

```

        if ( abs(iua) > MAX_I ) {
            CEO = 0;                /* PWM全相出力禁止 */
            RTPM01 = 0x3f;
            error_flag = ERROR1;    /* エラー番号セット */
        }
        ADS = 0x01;                /* ANI1選択 */
        ADM = ADM_START;          /* A/D変換スタート */
        break;
    case 0x01:
        iva = (( ADCR & 0x3ff ) - 0x200);
        if ( abs(iva) > MAX_I ) {
            CEO = 0;                /* PWM全相出力禁止 */
            RTPM01 = 0x3f;
            error_flag = ERROR1;    /* エラー番号セット */
        }

        ADS = 0x04;                /* ANI4選択 */
        ADM = ADM_START;          /* A/D変換スタート */
        break;
    default:
        vol = ~( ADCR >> 6 ) & 0x3ff;        /* 設定値読み出し */
        f_ref = ((vol + 1) >> 4 ) + MIN_FREQ; /* 設定周波数算出 */
        EI();
        calc_parameter();                /* 制御パラメータの計算 */
        P01 = 0;                        /* 割り込み開始出力 */
        break;
    }

    IF1H &= ~0x10;
    EI();
}

```

4.1.9 LED表示関数

```

/*****
/*      LED数値表示サブルーチン                                     */
/*      no   : 表示エリア番号(1-4)                               */
/*      data : 表示データ(0-99)                                   */
*****/
void led( unsigned short led_number, unsigned short led_data )
{
    unsigned short led_i;
    unsigned short led_ro;
    unsigned short led_ha;

    switch ( led_number ) {
        case LED_1:
            led_i = led_data / 10;
            led_ro = led_i / 10;
            led_ha = led_ro / 10;

```

```

        led_data = led_data - led_i * 10;
        led_i = led_i - led_ro * 10;
        led_ro = led_ro - led_ha * 10;

        output_data( LED11, ~led_pat[led_ha] );
        output_data( LED12, ~led_pat[led_ro] );
        output_data( LED13, ~led_pat[led_i] );
        output_data( LED14, ~led_pat[led_data] );
        break;

    case LED_2:
        led_i = led_data/10;
        led_ro = led_data - 10 * led_i;

        output_data( LED21, ~led_pat[led_i] );
        output_data( LED22, ~led_pat[led_ro] );
        break;

    case LED_3:
        led_i = led_data/10;
        led_ro = led_data - 10 * led_i;

        output_data( LED31, ~led_pat[led_i] );
        output_data( LED32, ~led_pat[led_ro] );
        break;

    case LED_4:
        led_i = led_data/10;
        led_ro = led_data - 10 * led_i;

        output_data( LED41, ~led_pat[led_i] );
        output_data( LED42, ~led_pat[led_ro] );
        break;
    }
}

/*****
/*      外部I/O出力
/*      reg   : 出力レジスタ番号
/*      data  : 出力データ
*****/
void output_data( unsigned short reg, unsigned short data )
{
    P5 = (unsigned char)reg;
    P4 = (unsigned char)(data & 0xff);
    P7 = (unsigned char)(((data >> 8) & 0x07) | (P7 & 0xf8));
    PM4 = 0x00;
    PM7 = PM7 & 0xf8;
    P66 = 0;
    P66 = 1;
}

```

```

}

/*****
/*      外部I/O入力                                          */
/*      reg   : 入力レジスタ番号                            */
/*      data  : 入力データ                                  */
/*****
unsigned short input_data( unsigned short reg )
{
    unsigned short data ;

    P5 = (unsigned char)reg;
    PM4 = 0xff;
    PM7 |= 0x07;
    P65 = 0;
    data = ((P7 << 8) & 0x07) | P4;
    P65 = 1;
    PM4 = 0x00;
    PM7 = PM7 & 0xf8;
    return data;
}

```

4.1.10 ハードウェア初期化処理関数

```

/*****
/*      ハードウェア(周辺I/O)初期化                          */
/*****
void hw_init (){

    IMS = 0xc8;                                          /* メモリ・エリア設定 */

    /* メイン・クロック発振,高速モード設定 */
    while(OSTC != 0x1f);                                /* 発振安定時間待ち */
    WDTM = 0x77;
    MCM = 0x03;                                          /* X1入力クロック選択 */

    /* FPGAアクセス ポート・モード・レジスタ初期化 */
    PM7 = 0xf8;                                          /* P70-P72 output */
    PM6 = 0x9f;                                          /* P65,P66 output */
    PM5 = 0xe0;                                          /* P50-P54 output */
    PM4 = 0x00;                                          /* P40-P47 output */

    /* TMW0 */
    TW0C = 0x10;                                         /* 停止,fx/4 ->5 MHz,TW0UDCのアンダフローごとに発生 */
    TW0M = 0x04;
    TW0OC = 0x00;                                        /* TW0TO0-TW0TO5は出力許可 */

    /* A/D */
    ADM = 0x1d;                                          /* A/D変換停止,セレクト・モード */
                                                    /* A/D変換時間 = 3.6 us,A/D変換待機モード */
}

```

```

ADS = 0x00;
PFM = 0x00;
PFT = 0x00;

/* 外部割り込み */
EGP = 0x00;          /* Rising edge disable */
EGN = 0x00;          /* Falling edge disable */

/* 割り込みマスク */
MK0L = 0xff;
MK0H = 0xfd;
MK1L = 0xff;
MK1H = 0xcf;

/* 優先順位 */
PR1H = 0xef;
}

```

4.1.11 ソフトウェア初期化処理関数

```

/*****
/*      ソフトウェア (変数) 初期化      */
*****/
void sw_init () {

    num_carrier = 0;          /* キャリア波数 */
    num_nc = 0;              /* キャリア波数 */
    term = 1;                /* 1/4回転回数 */
    rot_num = 0;             /* 回転回数 */
    offset = 96;             /* sin, cos値オフセット */
    n_ptr_rate = 1;

    sgn_sin_amp = 1;
    sgn_cos_amp = 1;
    sgn_ph = 1;

    f_ref_temp = MIN_FREQ;   /* 目標周波数 */
    f_ref = MIN_FREQ;        /* 設定周波数 */

    process = ROT_STOP;      /* 回転状態フラグ */
    sw_mode = SW_STOP;       /* 指定回転フラグ */

    error_flag = 0;          /* エラー・フラグ */
    set_flag = 1;

    n_f_ref_temp = MIN_FREQ;
    n_offset = 96;           /* キャリア波1/4相当 */
    n_num_pulse = 384;       /* キャリア波数設定 */
    n_ptr_rate = 1;
    n_v_amp = 71;           /* 振幅値設定 */
}

```

```
n_default_rot = 4;                                /* 最低回転回数設定 */

output_data( LED11, 0 );
output_data( LED12, 0 );
output_data( LED13, 0 );
output_data( LED14, 0 );
output_data( LED21, 0 );
output_data( LED22, 0 );
output_data( LED31, 0 );
output_data( LED32, 0 );
output_data( LED41, 0 );
output_data( LED42, 0 );

}
```

〔メモ〕

〔メモ〕

〔メモ〕

【発 行】

NECエレクトロニクス株式会社

〒211-8668 神奈川県川崎市中原区下沼部1753

電話（代表）：044(435)5111

—— お問い合わせ先 ——

【ホームページ】

NECエレクトロニクスの情報がインターネットでご覧になれます。

URL(アドレス) <http://www.necel.co.jp/>

【営業関係，技術関係お問い合わせ先】

半導体ホットライン

(電話：午前 9:00～12:00，午後 1:00～5:00)

電 話 : 044-435-9494

E-mail : info@necel.com

【資料請求先】

NECエレクトロニクスのホームページよりダウンロードいただくか，NECエレクトロニクスの販売特約店へお申し付けください。
