

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情報の目的で使用しないでください。また、当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。
標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）
特定水準： 航空機器、航空宇宙機器、海中継機器、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。お客様に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がその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。

H8/300Hシリーズ

アプリケーションノート 内蔵I/O編

ルネサス16ビットシングルチップマイクロコンピュータ
H8ファミリ／H8/300Hシリーズ

ご注意

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とは、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。

はじめに

H8/3003、3042、3002 は、内部 32 ビット構成の H8/300H CPU を核に、豊富な各種周辺機能を内蔵した高性能マイクロコンピュータです。

1 チップ上に CPU、RAM、16 ビットインテグレートドタイマユニット (ITU)、プログラマブルタイミングパターンコントローラ (TPC)、シリアルコミュニケーションインタフェース (SCI) および DMA コントローラ (DMAC) などの周辺機能を内蔵しており、小規模システムから大規模システムまで幅広いアプリケーションに適用できます。

本アプリケーションノートは、各内蔵周辺機能を単独で使った場合の動作例を示した基礎編とその組み合わせでの使用例を示した応用編の 2 部構成となっています。

なお、本アプリケーションノートに掲載されている各タスクのプログラム、回路などの動作は確認しておりますが、実際にご使用になる場合には、改めて動作確認の上ご使用くださいますようお願い致します。

目次

第 1 章 H8/300H シリーズ アプリケーションノート使用手引き

1.1	基礎編構成	1
1.2	応用編構成	3

第 2 章 基礎編

2.1	パルス出力	5
2.1.1	仕様	5
2.1.2	使用機能説明	5
2.1.3	動作説明	7
2.1.4	ソフトウェア説明	8
2.1.5	フローチャート	9
2.1.6	プログラムリスト	10
2.2	2 相エンコーダカウント	13
2.2.1	仕様	13
2.2.2	使用機能説明	13
2.2.3	動作説明	15
2.2.4	ソフトウェア説明	17
2.2.5	フローチャート	18
2.2.6	プログラムリスト	20
2.3	パルスの High および Low 幅測定	24
2.3.1	仕様	24
2.3.2	使用機能説明	24
2.3.3	動作説明	25
2.3.4	ソフトウェア説明	26
2.3.5	フローチャート	27
2.3.6	プログラムリスト	29
2.4	相補 PWM3 相出力	33
2.4.1	仕様	33
2.4.2	使用機能説明	33
2.4.3	動作説明	35
2.4.4	ソフトウェア説明	37
2.4.5	フローチャート	38
2.4.6	プログラムリスト	39
2.5	長周期パルス出力	43
2.5.1	仕様	43
2.5.2	使用機能説明	43
2.5.3	動作説明	45
2.5.4	ソフトウェア説明	47
2.5.5	フローチャート	48

2.5.6	プログラムリスト	49
2.6	正相・逆相 PWM3 相出力	52
2.6.1	仕様	52
2.6.2	使用機能説明	52
2.6.3	動作説明	54
2.6.4	ソフトウェア説明	55
2.6.5	フローチャート	56
2.6.6	プログラムリスト	57
2.7	PWM5 相出力	61
2.7.1	仕様	61
2.7.2	使用機能説明	62
2.7.3	動作説明	63
2.7.4	ソフトウェア説明	65
2.7.5	フローチャート	66
2.7.6	プログラムリスト	67
2.8	外部信号を使用したタイマカウンタ同時リセット	71
2.8.1	仕様	71
2.8.2	使用機能説明	72
2.8.3	動作説明	74
2.8.4	ソフトウェア説明	76
2.8.5	フローチャート	77
2.8.6	プログラムリスト	78
2.9	バッファモードを使用したワンショットパルス出力	82
2.9.1	仕様	82
2.9.2	使用機能説明	82
2.9.3	動作説明	85
2.9.4	ソフトウェア説明	86
2.9.5	フローチャート	88
2.9.6	プログラムリスト	90
2.10	4 ビット×4 系統出力	95
2.10.1	仕様	95
2.10.2	使用機能説明	96
2.10.3	動作説明	98
2.10.4	ソフトウェア説明	99
2.10.5	フローチャート	101
2.10.6	プログラムリスト	106
2.11	調歩同期式 SCI	114
2.11.1	仕様	114
2.11.2	使用機能説明	115
2.11.3	動作説明	117
2.11.4	ソフトウェア説明	118
2.11.5	フローチャート	119
2.11.6	プログラムリスト	121
2.12	同時送受信動作	126
2.12.1	仕様	126

2.12.2	使用機能説明	126
2.12.3	動作説明	128
2.12.4	ソフトウェア説明	129
2.12.5	フローチャート	130
2.12.6	プログラムリスト	132
2.13	マルチプロセッサ通信	136
2.13.1	仕様	136
2.13.2	使用機能説明	137
2.13.3	動作説明	139
2.13.4	ソフトウェア説明	142
2.13.5	フローチャート	144
2.13.6	プログラムリスト	147
2.14	ブロック転送	156
2.14.1	仕様	156
2.14.2	使用機能説明	157
2.14.3	動作説明	158
2.14.4	ソフトウェア説明	159
2.14.5	フローチャート	160
2.14.6	プログラムリスト	161
第3章 応用編		
3.1	高速データ出力	165
3.1.1	仕様	165
3.1.2	考え方	166
3.1.3	使用機能説明	166
3.1.4	動作説明	169
3.1.5	ソフトウェア説明	170
3.1.6	フローチャート	172
3.1.7	プログラムリスト	175
3.2	SCI 連続送受信	180
3.2.1	仕様	180
3.2.2	考え方	180
3.2.3	使用機能説明	181
3.2.4	ソフトウェア説明	182
3.2.5	フローチャート	185
3.2.6	プログラムリスト	189
3.3	サイン波出力	196
3.3.1	仕様	196
3.3.2	考え方	196
3.3.3	使用機能説明	199
3.3.4	動作説明	201
3.3.5	ソフトウェア説明	203
3.3.6	フローチャート	205
3.3.7	プログラムリスト	209
3.4	4相ステッピングモータ応用例	217
3.4.1	仕様	217

3.4.2	考え方	217
3.4.3	使用機能説明	219
3.4.4	動作説明	222
3.4.5	ソフトウェア説明	223
3.4.6	フローチャート	226
3.4.7	プログラムリスト	231
3.5	5相ステップングモータ応用例	242
3.5.1	仕様	242
3.5.2	考え方	242
3.5.3	使用機能説明	244
3.5.4	動作説明	247
3.5.5	ソフトウェア説明	248
3.5.6	フローチャート	251
3.5.7	プログラムリスト	256

1. H8/300H シリーズ アプリケーションノート使用手引き

本アプリケーションノートは図 1.1 に示すように 2 部構成になっています。

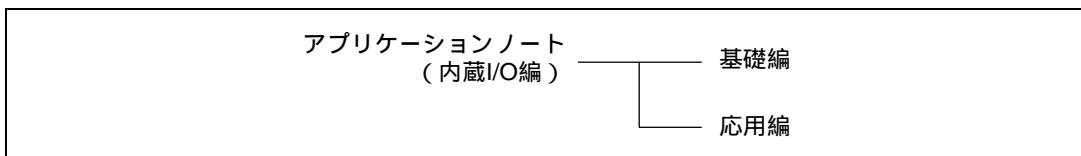


図 1.1 アプリケーションノート構成

- (1) 基礎編
単体タスク例をもとにH8/300Hの各周辺機能の動作例を説明したものです。
- (2) 応用編
組み合わせタスク例をもとにH8/300Hの各周辺機能を組み合わせたときの動作例を説明したものです。

1.1 基礎編構成

基礎編は図 1.2 に示す構成で周辺機能を単体で使用方法について説明しています。

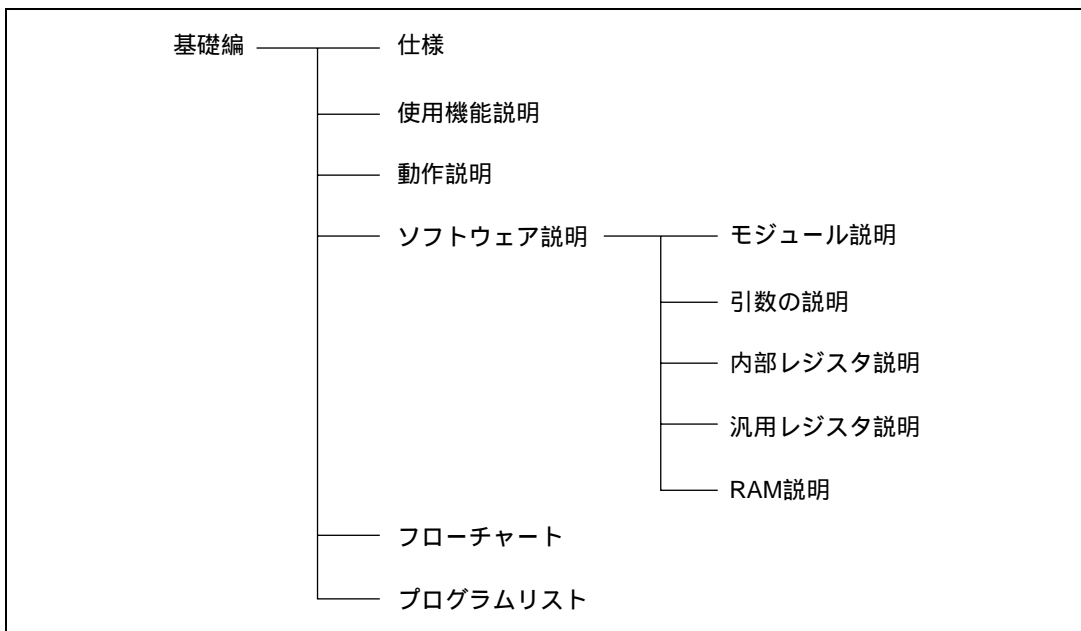


図 1.2 基礎編の構成

- (1) 仕様
タスク例のシステム仕様について説明しています。
- (2) 使用機能説明
タスク例で使用する周辺機能の特長および周辺機能の割り付けについて説明しています。
- (3) 動作説明
タスク例の動作をタイミングチャートを使用し説明しています。
- (4) ソフトウェア説明
 - (a) モジュール説明
タスク例を動作させるソフトウェアのモジュールについて説明しています。
 - (b) 引数の説明
モジュールを実行する際に必要な入力引数と、実行後の出力引数について説明しています。
 - (c) 内部レジスタ説明
モジュールで設定する周辺機能の内部レジスタ（タイマコントロールレジスタ、シリアルモードレジスタなど）について説明します。
 - (d) 汎用レジスタ説明
モジュールで使用する汎用レジスタ名（R0～R7）および機能について説明します。
 - (e) RAM説明
モジュールで使用するRAMのラベル名および機能について説明します。
- (5) フローチャート
タスク例を実行するソフトウェアについてゼネラルフローチャートを使用し説明します。
- (6) プログラムリスト
タスク例を実行するソフトウェアのプログラムリストを示します。

1.2 応用編構成

応用編は図 1.3 に示す構成で周辺機能を組み合わせて使用する方法について説明しています。

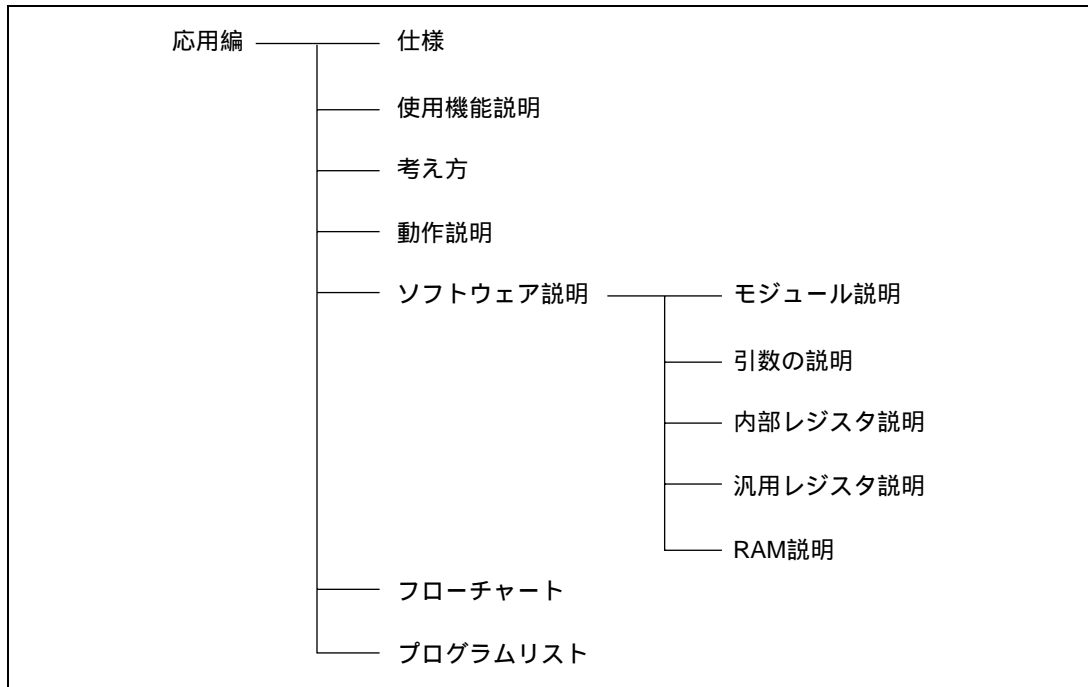


図 1.3 応用編の構成

- (1) 仕様
タスク例のシステム仕様について説明しています。
- (2) 考え方
タスク例のシステムを実現するための方法を説明しています。
- (3) 使用機能説明
タスク例で使用する周辺機能の特長および周辺機能の割り付けについて説明しています。
- (4) 動作説明
タスク例の動作をタイミングチャートを使用し説明しています。
- (5) ソフトウェア説明
 - (a) モジュール説明
タスク例を動作させるソフトウェアのモジュールについて説明しています。
 - (b) 引数の説明
モジュールを実行する際に必要な入力引数と、実行後の出力引数について説明しています。
 - (c) 内部レジスタ説明
モジュールで設定する周辺機能の内部レジスタ（タイマコントロールレジスタ、シリアルモードレジスタなど）について説明します。
 - (d) 汎用レジスタ説明
モジュールで使用する汎用レジスタ名（R0～R7）および機能について説明します。
 - (e) RAM説明
モジュールで使用するRAMのラベル名および機能について説明します。
- (6) フローチャート

1. H8/300H シリーズアプリケーションノート使用手引き

- タスク例を実行するソフトウェアについてゼネラルフローチャートを使用し説明します。
- (7) プログラムリスト
タスク例を実行するソフトウェアのプログラムリストを示します。

2. 基礎編

2.1 パルス出力

MCU : H8/3003

使用機能 : 16 ビットインテグレートドタイマユニット (ITU)

2.1.1 仕様

図 2.1 に示すように RAM に設定された周期のデューティ 50% のパルスを出力します。
16MHz で動作時、出力するパルスの周期は 125ns から 4.09ms の間で任意に設定できます。

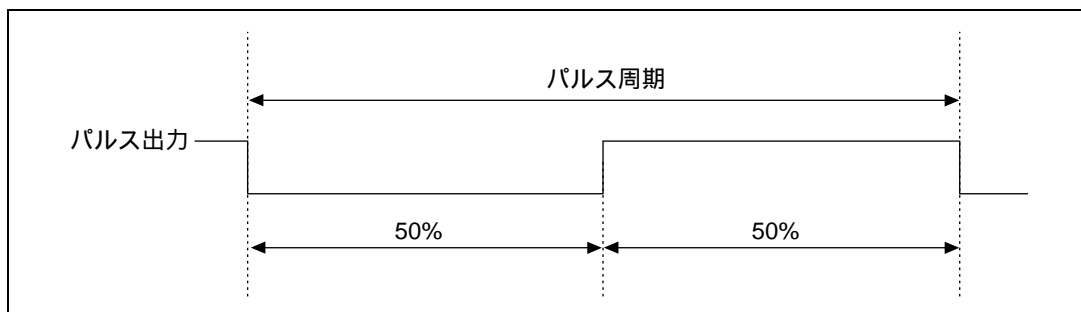


図 2.1 パルス出力例

2.1.2 使用機能説明

本タスク例では ITU/ch0 を使用してデューティ 50% のパルスを出力します。

図 2.2 に本タスク例で使用する ITU/ch0 のブロック図を示します。

ch0 では以下の機能を使用します。

- ソフトウェアを介さずハードウェアで自動的にパルスを出力する機能。(アウトプットコンペア)
- コンペアマッチ時、カウンタをクリアする機能。(カウンタクリア)
- コンペアマッチが起きるごとに出力が反転する機能。(トグル出力)

2. 基礎編

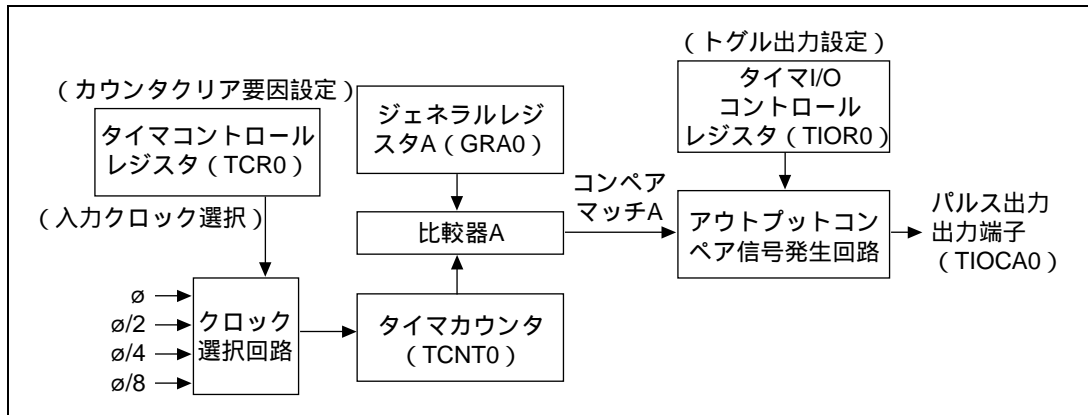


図 2.2 ITU/ch0 ブロック図

表 2.1 に本タスク例の機能割り付けを示します。表 2.1 に示すように ITU の機能を割り付け、パルスを出します。

表 2.1 ITU 機能割り付け

ITU 機能	機能
TCR0	TCNT に入力するクロック、およびカウンタクリア要因を選択する。
TIOCA0	パルスを出力する。
TIOA0	パルスの出力レベルを設定する。
GRA0	パルスの 1/2 周期を設定する。

2.1.3 動作説明

図 2.3 に動作原理を示します。図 2.3 に示すように H8/3003 のハードウェア処理およびソフトウェアの処理によりパルスを出力します。

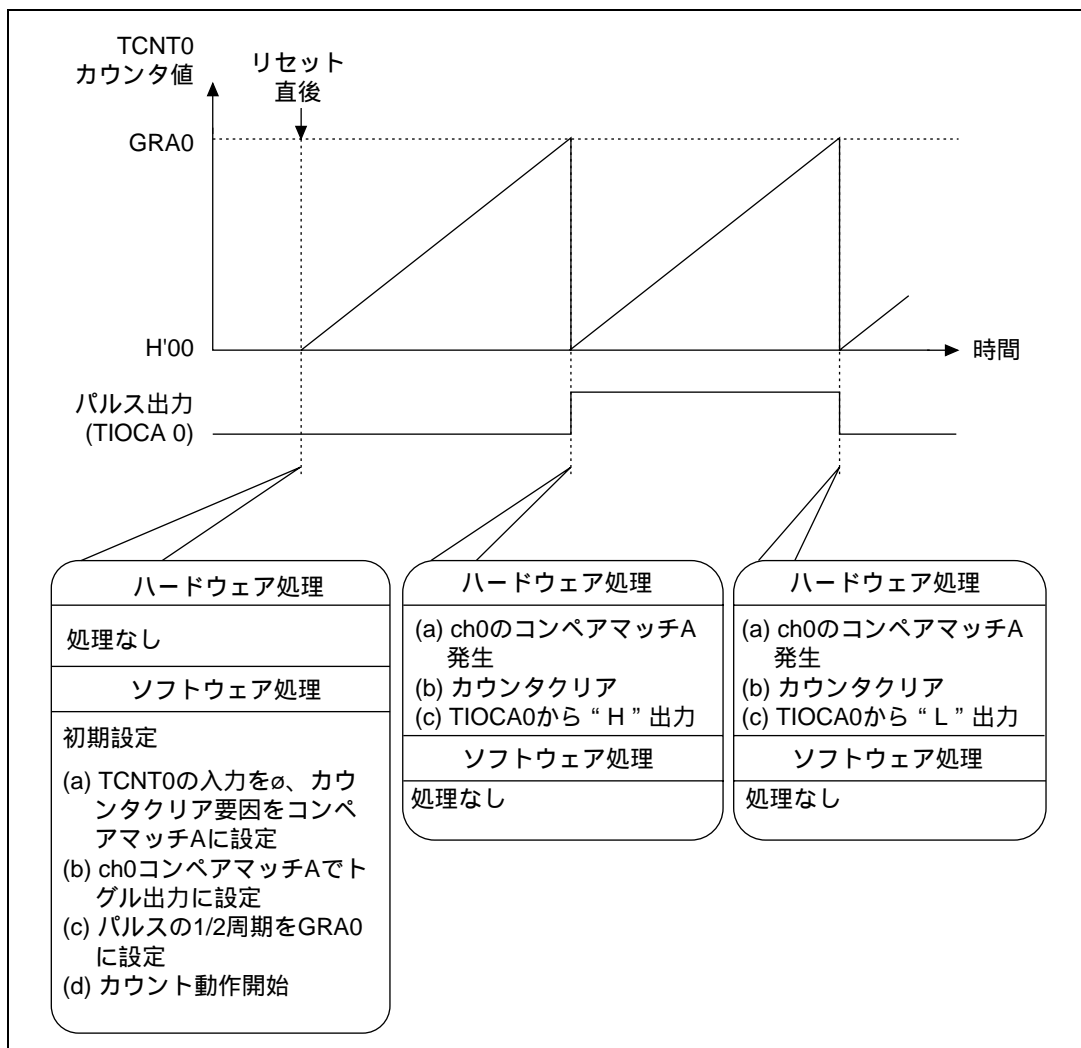


図 2.3 パルス出力動作原理

2. 基礎編

2.1.4 ソフトウェア説明

表 2.2 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	POUTMN	ITU、RAM の初期設定、およびパルス出力を行う。

表 2.3 引数の説明

ラベル名、レジスタ名	機能	データ長	使用モジュール名	入出力
PUL_CYC	パルスの周期に相当するタイマ値を設定する。 パルスの周期は以下の式にて求める。 パルス周期 (ns) = タイマ値 × 周期 (16MHz 動作時 62.5ns)	1ワード	メインルーチン	入力

表 2.4 使用内部レジスタ説明

レジスタ名	機能	使用モジュール名	
TSTR	タイマカウンタの動作 / 禁止を設定する。	メインルーチン	
ch0	TCR0	TCNT に入力するクロックおよびカウンタクリア要因を設定する。	メインルーチン
	TIOR0	コンペアマッチ A 発生時に、出力するパルスのレベルを設定する。	メインルーチン
	GRA0	出力パルスの 1/2 周期を設定する。	メインルーチン

表 2.5 使用汎用レジスタ説明

使用モジュール名	レジスタ名	機能
メインルーチン	R0	データ設定時のワークとして使用する。

- 使用 RAM 説明
本アプリケーション例では引数以外のRAMは使用していません。

2.1.5 フローチャート

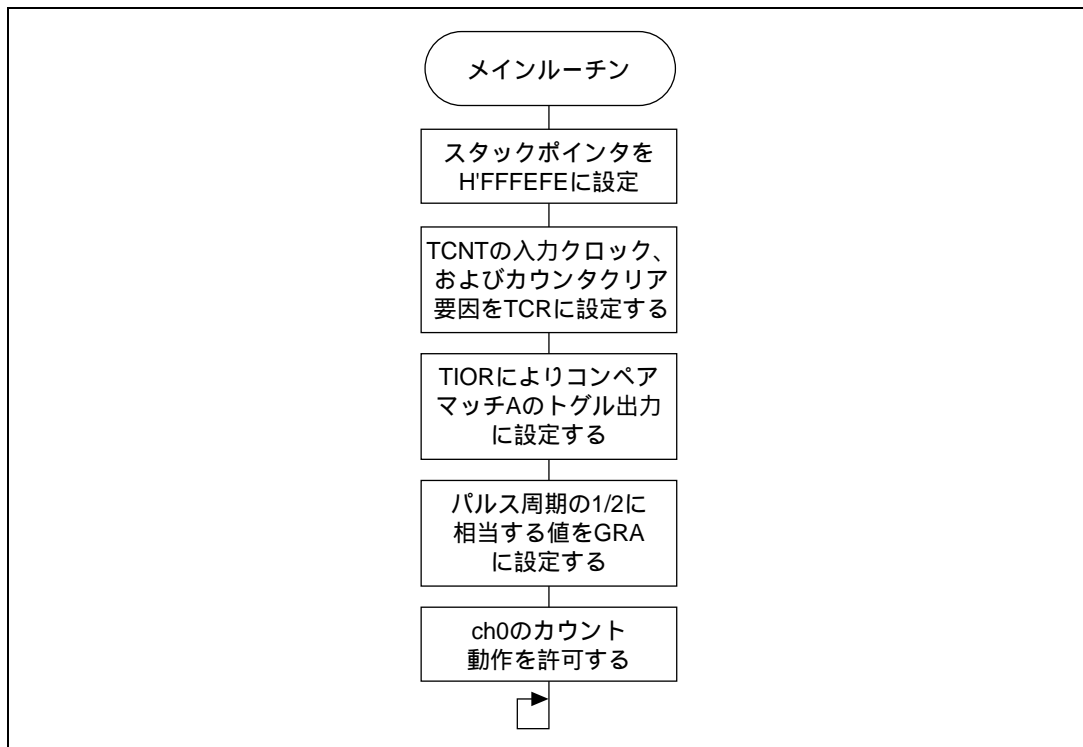


図 2.4 メインルーチン

2.1.6 プログラムリスト

```
1          1 ;*****
2          2 ;*
3          3 ;*      VECTOR ADDRESS
4          4 ;*
5          5 ;*****
6          6 ;
7 000000    7          .SECTION    VECT, CODE, LOCATE=H'000000
8          8 ;
9 000000 00010000  9 RES      .DATA.L    POUTMN
10         10 ;
11 00001C    11          .ORG      H'00001C
12         12 ;
13 00001C 00010000  13 NMI     .DATA.L    POUTMN
14 000020 00010000  14 TRAPA0  .DATA.L    POUTMN
15 000024 00010000  15 TRAPA1  .DATA.L    POUTMN
16 000028 00010000  16 TRAPA2  .DATA.L    POUTMN
17 00002C 00010000  17 TRAPA3  .DATA.L    POUTMN
18         18 ;
19 000030    19          .ORG      H'000030
20         20 ;
21 000030 00010000  21 IRQ0   .DATA.L    POUTMN
22 000034 00010000  22 IRQ1   .DATA.L    POUTMN
23 000038 00010000  23 IRQ2   .DATA.L    POUTMN
24 00003C 00010000  24 IRQ3   .DATA.L    POUTMN
25 000040 00010000  25 IRQ4   .DATA.L    POUTMN
26 000044 00010000  26 IRQ5   .DATA.L    POUTMN
27 000048 00010000  27 IRQ6   .DATA.L    POUTMN
28 00004C 00010000  28 IRQ7   .DATA.L    POUTMN
29 000050 00010000  29 WOVI0  .DATA.L    POUTMN
30 000054 00010000  30 CMI    .DATA.L    POUTMN
31         31 ;
32 000060    32          .ORG      H'000060
33 000060 00010000  33 IMIA0  .DATA.L    POUTMN
34 000064 00010000  34 IMIB0  .DATA.L    POUTMN
35 000068 00010000  35 OVI0   .DATA.L    POUTMN
36         36 ;
37 000070    37          .ORG      H'000070
38 000070 00010000  38 IMIA1  .DATA.L    POUTMN
39 000074 00010000  39 IMIB1  .DATA.L    POUTMN
40 000078 00010000  40 OVI1   .DATA.L    POUTMN
41         41 ;
42 000080    42          .ORG      H'000080
43 000080 00010000  43 IMIA2  .DATA.L    POUTMN
44 000084 00010000  44 IMIB2  .DATA.L    POUTMN
45 000088 00010000  45 OVI2   .DATA.L    POUTMN
```

```

46          46 ;
47 000090          47          .ORG          H'000090
48 000090 00010000 48 IMIA3          .DATA.L          POUTMN
49 000094 00010000 49 IMIB3          .DATA.L          POUTMN
50 000098 00010000 50 OVI3          .DATA.L          POUTMN
51          51 ;
52 0000A0          52          .ORG          H'0000A0
53 0000A0 00010000 53 IMIA4          .DATA.L          POUTMN
54 0000A4 00010000 54 IMIB4          .DATA.L          POUTMN
55 0000A8 00010000 55 OVI4          .DATA.L          POUTMN
56          56 ;
57 0000B0          57          .ORG          H'0000B0
58 0000B0 00010000 58 DEND0A         .DATA.L          POUTMN
59 0000B4 00010000 59 DEND0B         .DATA.L          POUTMN
60 0000B8 00010000 60 DEND1A         .DATA.L          POUTMN
61 0000BC 00010000 61 DEND1B         .DATA.L          POUTMN
62 0000C0 00010000 62 DEND2A         .DATA.L          POUTMN
63 0000C4 00010000 63 DEND2B         .DATA.L          POUTMN
64 0000C8 00010000 64 DEND3A         .DATA.L          POUTMN
65 0000CC 00010000 65 DEND3B         .DATA.L          POUTMN
66 0000D0 00010000 66 ERI0          .DATA.L          POUTMN
67 0000D4 00010000 67 RXI0          .DATA.L          POUTMN
68 0000D8 00010000 68 TXI0          .DATA.L          POUTMN
69 0000DC 00010000 69 TEI0          .DATA.L          POUTMN
70 0000E0 00010000 70 ERI1          .DATA.L          POUTMN
71 0000E4 00010000 71 RXI1          .DATA.L          POUTMN
72 0000E8 00010000 72 TXI1          .DATA.L          POUTMN
73 0000EC 00010000 73 TEI1          .DATA.L          POUTMN
74 0000F0 00010000 74 ADI          .DATA.L          POUTMN
75          75 ;
76          76 ;*****
77          77 ;*
78          78 ;*          RAM ALLOCATION
79          79 ;*
80          80 ;*****
81          81 ;
82 FFFF00          82          .SECTION          RAM,DATA,LOCATE=H'FFFF00
83          83 ;
84 FFFF00 00000002 84 PUL_CYC .RES.W          1          ;Pulse cycle time
85          85 ;
86          86 ;*****
87          87 ;*
88          88 ;*          SYMBOL DEFINITIONS
89          89 ;*
90          90 ;*****
91          91 ;
92          00FFFF60 92 TSTR          .EQU          H'FFFF60          ;Timer start register

```

2. 基礎編

```
93          93 ;
94          00FFFF64 94 TCR0 .EQU H'FFFF64 ;Timer control register
95          00FFFF65 95 TIOR0 .EQU H'FFFF65 ;Timer I/O control register
96          00FFFF6A 96 GRA0 .EQU H'FFFF6A ;General register A0
97          97 ;
98          98 ;*****
99          99 ;* *
100         100 ;* MAIN PROGRAM : POUTMN *
101         101 ;* *
102         102 ;*****
103         103 ;
104 010000 104 .SECTION PROG, CODE, LOCATE=H'010000
105         105 ;
106          00010000 106 POUTMN: .EQU $
107 010000 7A0700FFFEFE 107 MOV.L #H'FFFEFE, SP ;Initialize stack pointer
108 010006 F8A0 108 MOV.B #B'10100000, R0L
109 010008 3864 109 MOV.B R0L, @TCR0 ;Initialize TCR0
110 01000A F88B 110 MOV.B #B'10001011, R0L
111 01000C 3865 111 MOV.B R0L, @TIOR0 ;Initialize TIOR0
112         112 ;
113 01000E 6B00FF00 113 MOV.W @PUL_CYC, R0
114 010012 1110 114 SHLR.W R0
115 010014 6B80FF6A 115 MOV.W R0, @GRA0
116         116 ;
117 010018 F8E1 117 MOV.B #B'11100001, R0L
118 01001A 3860 118 MOV.B R0L, @TSTR ;Set TCNT0 start data
119         119
120 01001C 40FE 120 POUTMN99 BRA POUTMN99
121         121
122         122 .END
*****TOTAL ERRORS 0
*****TOTAL WARNINGS 0
```

2.2 2相エンコーダカウント

MCU : H8/3003

使用機能 : ITU (位相計数モード)

2.2.1 仕様

図 2.5 に示すように 2 本の外部クロックを入力し、そのパルスの位相差によりカウンタをカウントアップまたはカウントダウンします。またカウント数を測定し、結果を RAM に設定します。

タイマカウンタの初期値を H'8000 とし、+7FFF、-8000 までカウントできます。

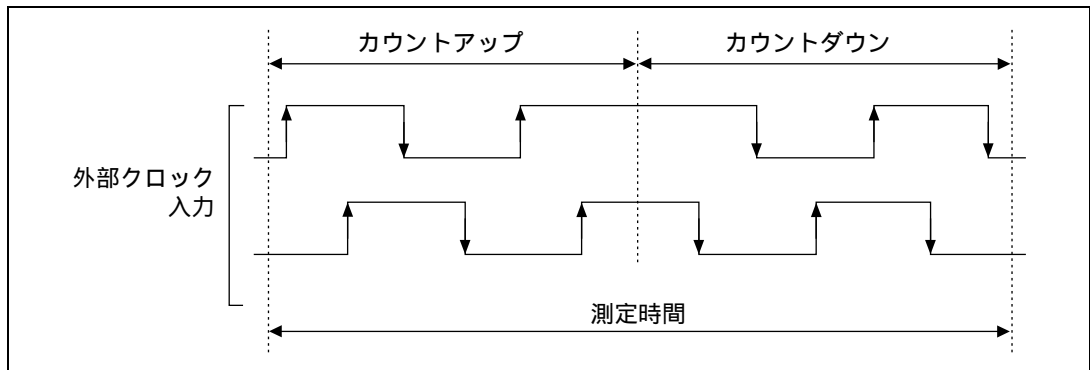


図 2.5 2相エンコーダカウント動作例

2.2.2 使用機能説明

本タスク例では、ITU の ch2 をアップ/ダウンカウンタとして使用し、ch0 で測定時間を生成します。ch0 のアウトプットコンペア出力端子と ch2 のインプットキャプチャ入力端子を接続し、ch0 のアウトプットコンペアにより、測定時間の終了を ch2 へ知らせます。

図 2.6 に ch0 のブロック図を示します。ch0 では、以下の機能を使用して測定時間を生成し、ch2 のインプットキャプチャ入力端子へパルス出力します。

- ソフトウェアを介さずハードウェアで自動的にパルスを出す機能。(アウトプットコンペア)

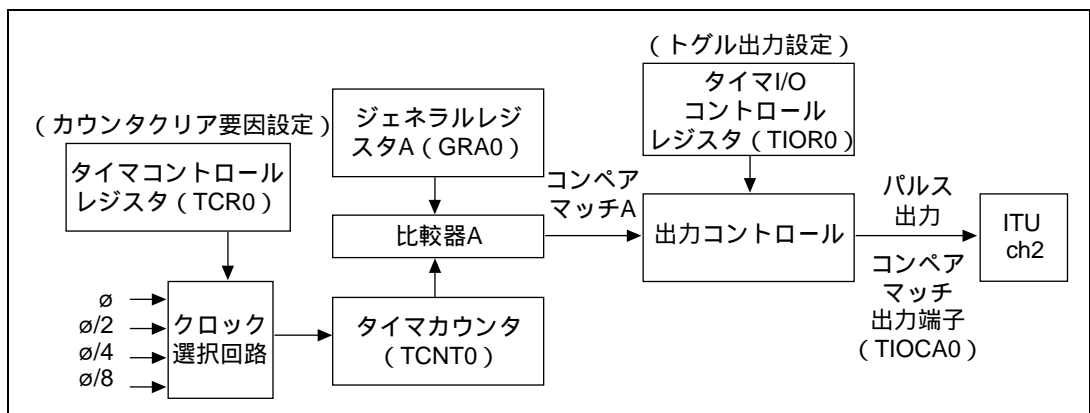


図 2.6 ITU/ch0 ブロック図

2. 基礎編

図 2.7 に ch2 のブロック図を示します。ch2 では以下の機能を使用してタイマカウンタをアップ / ダウンカウントします。入力キャプチャ両エッジ検出時のカウンタ値を測定結果とします。

- 2本の外部クロックの位相差を検出し、タイマカウンタをアップ / ダウンカウントする機能。(位相計数モード)
- パルスの入力エッジを検出を行い、そのときのタイマ値を内部レジスタに設定する機能。(入力キャプチャ)
- インพุットキャプチャ発生時、割り込み処理を起動する機能。
- パルスの入力エッジ検出時タイマカウンタをクリアする機能。
- タイマカウンタのオーバーフローまたはアンダフロー検出時、割り込み処理を起動する機能。

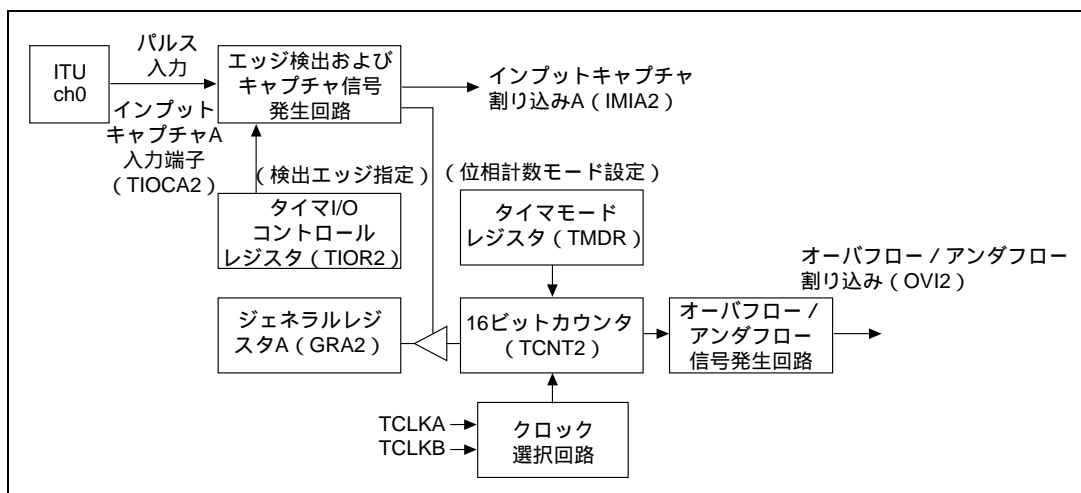


図 2.7 ITU/ch2 ブロック図

表 2.6 に本タスク例の機能割り付けを示します。表 2.6 に示すように ITU の機能を割り付け、2 相エンコーダパルスの 2 つの位相差を検出し、カウンタをアップ / ダウンカウントします。

表 2.6 ITU 機能割り付け

ITU 機能	機能	
TSTR	ch0、2のタイマカウンタ動作を許可 / 禁止する。	
TMDR	位相計数モード設定、および OVR フラグのセット条件を設定する。	
TCLKA	外部クロック入力端子。	
TCLKB	外部クロック入力端子。	
ch0	TCR0	カウンタクロック、カウンタクリア要因を選択する。
	TIOCA0	パルスを出力する。
	TIOR0	コンペアマッチ A でトグル出力に設定する。
	GRA0	カウンタの測定時間を設定する。
ch2	TIOCA2	インพุットキャプチャ信号入力端子。
	TIER2	IMFA、OVF による割り込みを許可する。
	TIOR2	エッジ検出時、インพุットキャプチャ A 発生に設定する。
	GRA2	インพุットキャプチャ A によりカウンタ結果が設定される。

2.2.3 動作説明

図 2.8 に動作原理を示します。図 2.8 に示すように H8/3003 のハードウェア処理およびソフトウェアの処理によりカウンタをカウントアップまたはカウントダウンします。

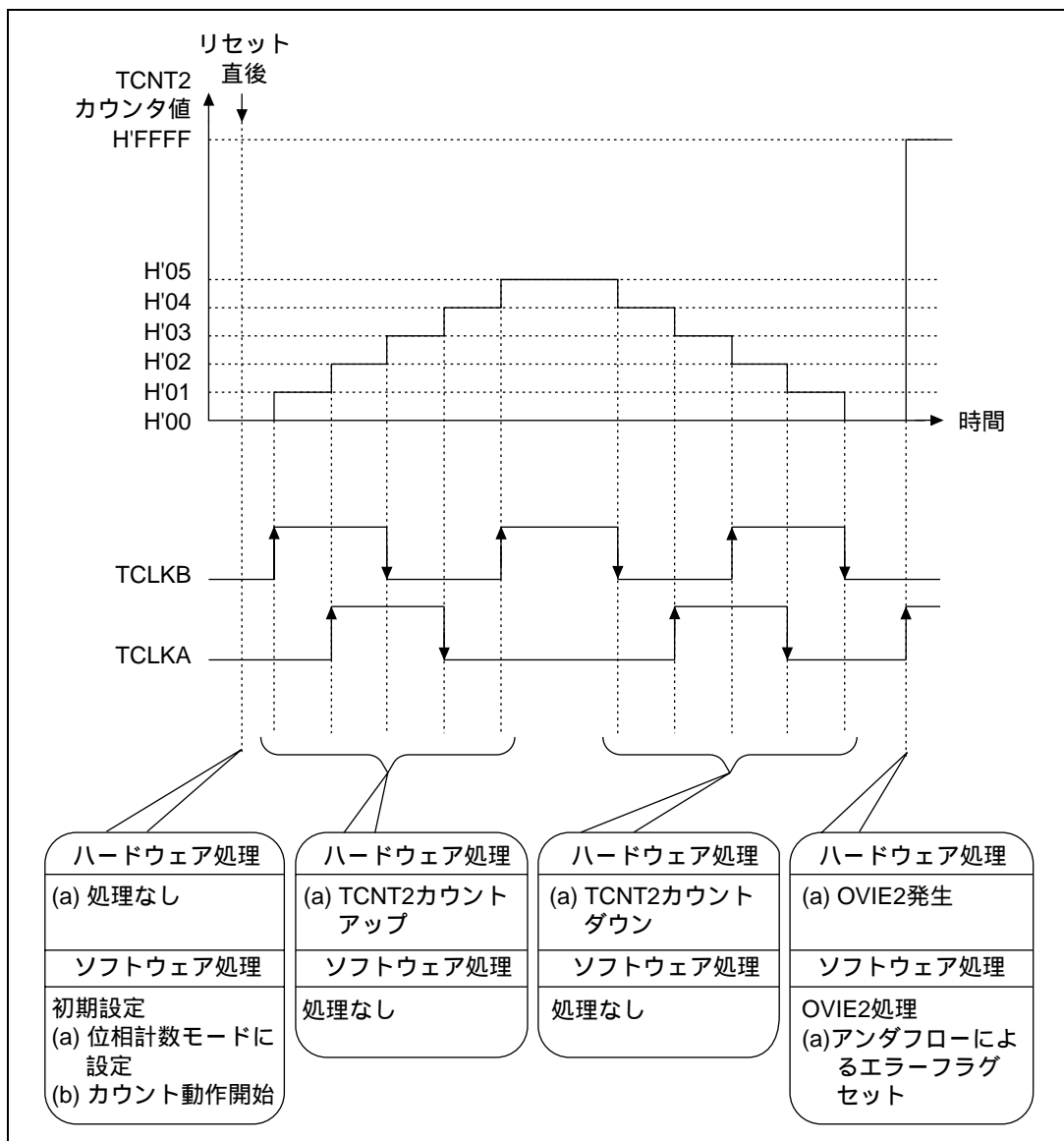


図 2.8 位相計数モードの動作原理 (1)

2. 基礎編

図2.8に示すようにH8/3003のハードウェア処理およびソフトウェアの処理によりカウンタのオーバーフロー/アンダフロー時、外部イベント発生時に割り込み処理を実施します。

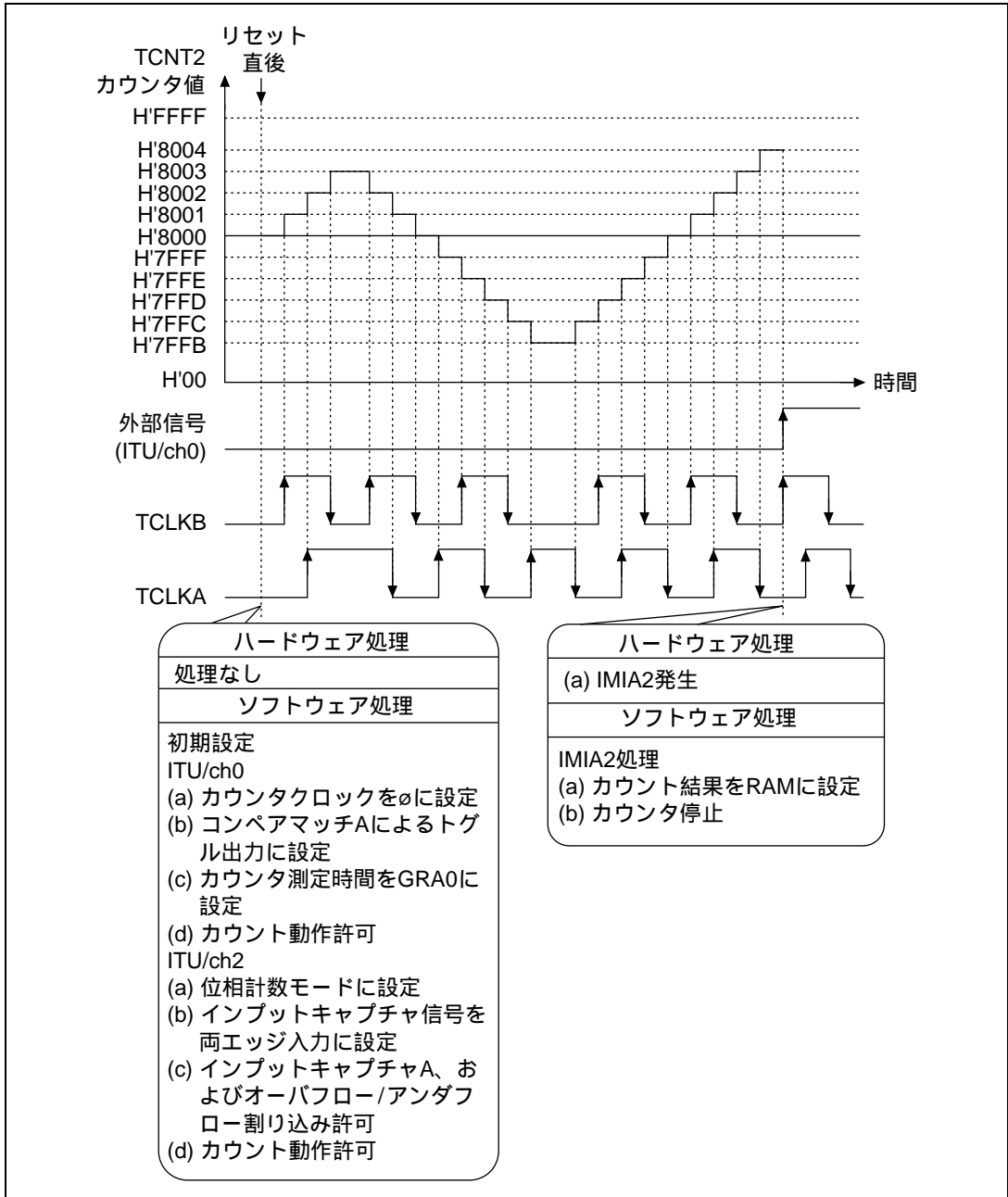


図 2.8 位相計数モードの動作原理 (2)

2.2.4 ソフトウェア説明

表 2.7 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	PHACNTMN	ITU および RAM の初期設定を行う。
カウンタ値測定	PHACNT1	IMIA2 により起動し、GRA の値からアップ / ダウンカウント結果を RAM に設定する。
エラーフラグ設定	PHACNT2	OVIE2 により起動し、オーバフロー / アンダフローによるエラーフラグを設定する。

表 2.8 引数の説明

ラベル名、レジスタ名	機能	データ長	使用モジュール名	入出力
MSR_TIM	カウンタ測定時間に相当するタイマ値を設定する。 測定時間は以下の式にて求まる。 測定時間 (ns) = タイマ値 × 周期 (16MHz 動作時 62.5ns)	1ワード	メインルーチン	入力
CNT_DATA	アップ / ダウンカウント結果を設定する。	1ワード	カウンタ値測定	出力
ERR_F	オーバフロー / アンダフローによるエラーフラグを設定する。	1ビット	エラーフラグ設定	出力

表 2.9 使用内部レジスタ説明

レジスタ名	機能	使用モジュール名	
TSTR	タイマカウンタの動作 / 禁止を設定する。	メインルーチン	
TMDR	位相計数モードの設定、および OVF フラグのセット条件を設定する。	メインルーチン	
ch0	TCR0	TCNT のカウンタクロック、およびカウンタクリア要因を選択する。	メインルーチン
	TIOR0	アウトプットコンペアをトグル出力に設定する。	メインルーチン
	GRA0	カウンタ測定時間を設定する。	メインルーチン
ch2	TIOR2	外部信号の立ち下がりエッジによりインプットキャプチャ A 発生に設定する。	メインルーチン
	GRA2	インプットキャプチャ A 時にカウンタ値が設定される。	カウンタ値測定
	TIER2	IMFA、OVF による割り込みを許可する。	メインルーチン
	TSR2	IMFA、OVF による割り込み発生を示す。	エラーフラグ設定

表 2.10 使用汎用レジスタ説明

使用モジュール名	レジスタ名	機能
メインルーチン	R0L	データ設定時のワークとして使用する。
カウンタ値測定	R0	データ設定時のワークとして使用する。

表 2.11 使用 RAM 説明

ラベル名、レジスタ名	機能	データ長	使用モジュール名
CNT_OVR	エラーフラグを格納する。	1バイト	エラーフラグ設定

2.2.5 フローチャート

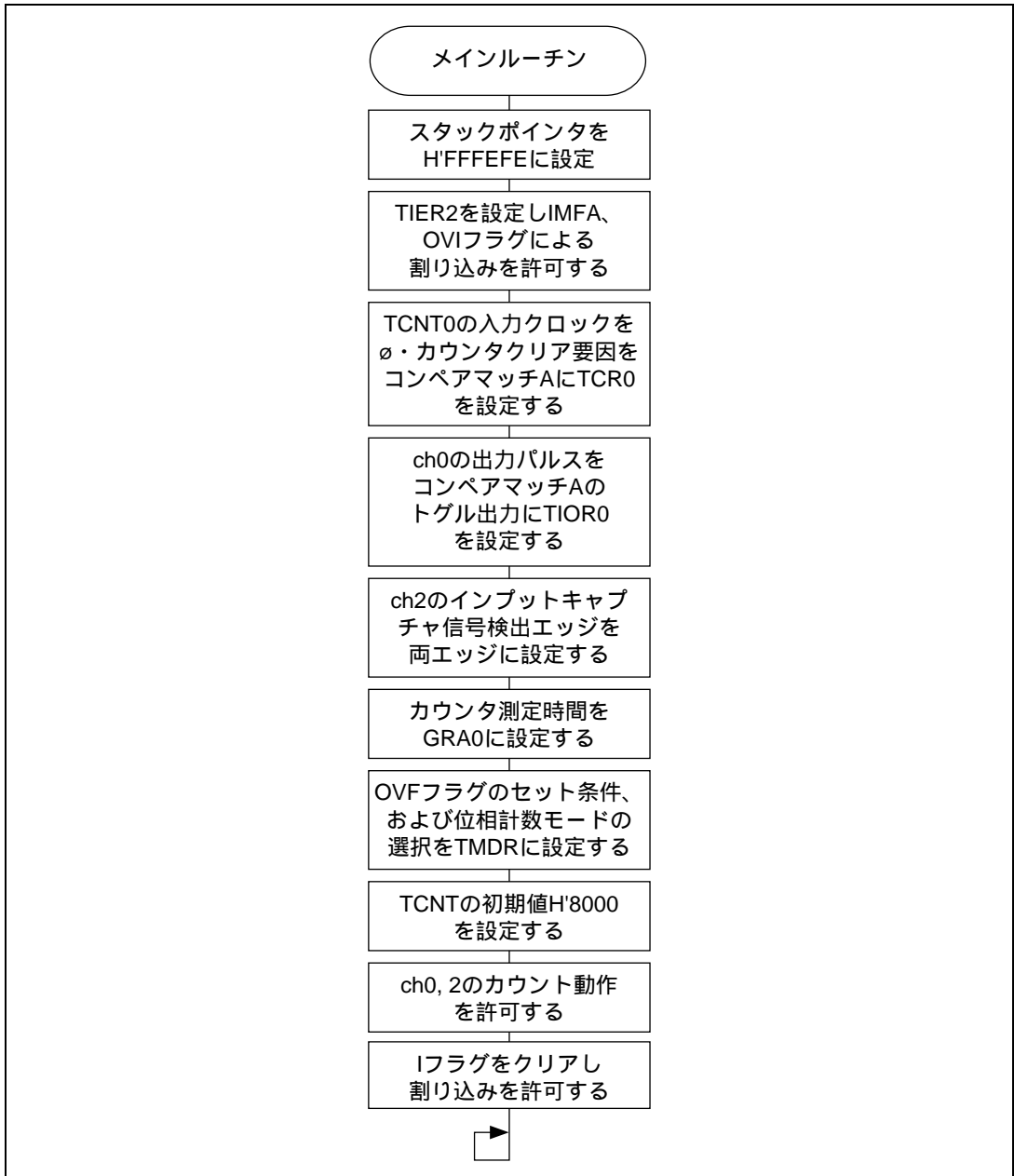


図 2.9 メインルーチン

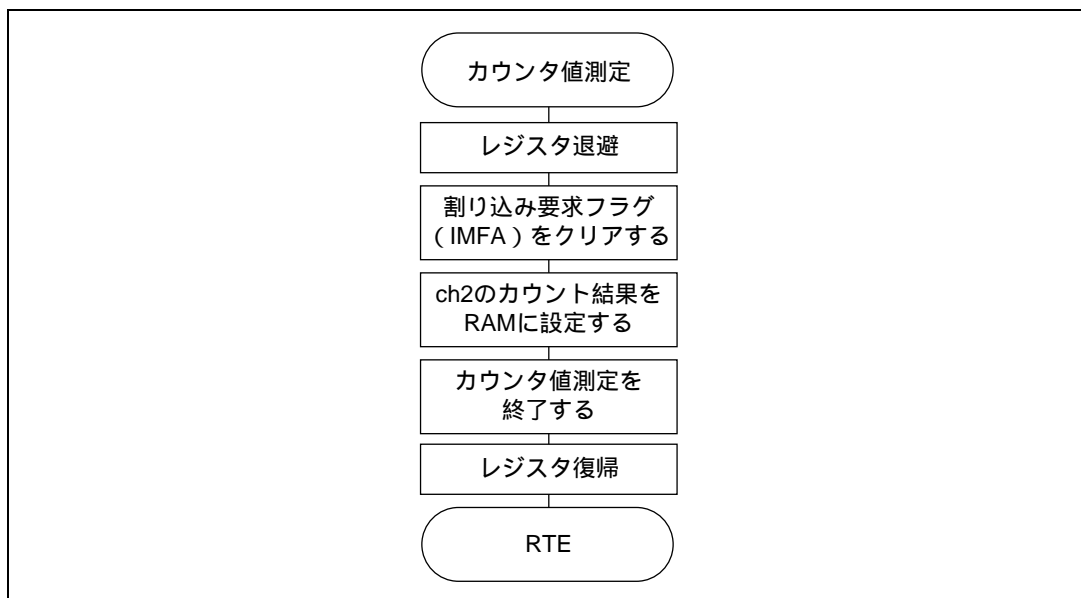


図 2.10 カウンタ値測定



図 2.11 エラーフラグ設定

2.2.6 プログラムリスト

```
1          1 ;*****
2          2 ;*
3          3 ;*          VECTOR ADDRESS
4          4 ;*
5          5 ;*****
6          6 ;
7 000000   7          .SECTION VECT, CODE, LOCATE=H'000000
8          8 ;
9 000000 00010000  9 RES          .DATA.L PHACNTMN
10         10 ;
11 00001C   11          .ORG          H'00001C
12 00001C 00010000 12 NMI          .DATA.L PHACNTMN
13         13 ;
14 000030   14          .ORG          H'000030
15 000030 00010000 15 IRQ0         .DATA.L PHACNTMN
16 000034 00010000 16 IRQ1         .DATA.L PHACNTMN
17 000038 00010000 17 IRQ2         .DATA.L PHACNTMN
18 00003C 00010000 18 IRQ3         .DATA.L PHACNTMN
19 000040 00010000 19 IRQ4         .DATA.L PHACNTMN
20 000044 00010000 20 IRQ5         .DATA.L PHACNTMN
21 000048 00010000 21 IRQ6         .DATA.L PHACNTMN
22 00004C 00010000 22 IRQ7         .DATA.L PHACNTMN
23 000050 00010000 23 WOVI0        .DATA.L PHACNTMN
24 000054 00010000 24 CMI          .DATA.L PHACNTMN
25         25 ;
26 000060   26          .ORG          H'000060
27 000060 00010000 27 IMIA0        .DATA.L PHACNTMN
28 000064 00010000 28 IMIB0        .DATA.L PHACNTMN
29 000068 00010000 29 OVI0         .DATA.L PHACNTMN
30         30 ;
31 000070   31          .ORG          H'000070
32 000070 00010000 32 IMIA1        .DATA.L PHACNTMN
33 000074 00010000 33 IMIB1        .DATA.L PHACNTMN
34 000078 00010000 34 OVI1         .DATA.L PHACNTMN
35         35 ;
36 000080   36          .ORG          H'000080
37 000080 00010032 37 IMIA2        .DATA.L PHACNT1
38 000084 00010000 38 IMIB2        .DATA.L PHACNTMN
39 000088 0001004C 39 OVI2         .DATA.L PHACNT2
40         40 ;
41 000090   41          .ORG          H'000090
42 000090 00010000 42 IMIA3        .DATA.L PHACNTMN
43 000094 00010000 43 IMIB3        .DATA.L PHACNTMN
44 000098 00010000 44 OVI3         .DATA.L PHACNTMN
45         45 ;
```

```

46 0000A0          46          .ORG      H'0000A0
47 0000A0 00010000 47 IMIA4     .DATA.L  PHACNTMN
48 0000A4 00010000 48 IMIB4     .DATA.L  PHACNTMN
49 0000A8 00010000 49 OVI4     .DATA.L  PHACNTMN
50
51 0000B0          51          .ORG      H'0000B0
52 0000B0 00010000 52 DEND0A   .DATA.L  PHACNTMN
53 0000B4 00010000 53 DEND0B   .DATA.L  PHACNTMN
54 0000B8 00010000 54 DEND1A   .DATA.L  PHACNTMN
55 0000BC 00010000 55 DEND1B   .DATA.L  PHACNTMN
56 0000C0 00010000 56 DEND2A   .DATA.L  PHACNTMN
57 0000C4 00010000 57 DEND2B   .DATA.L  PHACNTMN
58 0000C8 00010000 58 DEND3A   .DATA.L  PHACNTMN
59 0000CC 00010000 59 DEND3B   .DATA.L  PHACNTMN
60 0000D0 00010000 60 ERI0     .DATA.L  PHACNTMN
61 0000D4 00010000 61 RXI0     .DATA.L  PHACNTMN
62 0000D8 00010000 62 TXI0     .DATA.L  PHACNTMN
63 0000DC 00010000 63 TEI0     .DATA.L  PHACNTMN
64 0000E0 00010000 64 ERI1     .DATA.L  PHACNTMN
65 0000E4 00010000 65 RXI1     .DATA.L  PHACNTMN
66 0000E8 00010000 66 TXI1     .DATA.L  PHACNTMN
67 0000EC 00010000 67 TEI1     .DATA.L  PHACNTMN
68 0000F0 00010000 68 ADI      .DATA.L  PHACNTMN
69
70                70                ;*****
71                71                ;*
72                72                ;*      RAM ALLOCATION
73                73                ;*
74                74                ;*****
75                75                ;
76 FFFF00          76                .SECTION  RAM,DATA,LOCATE=H'FFFF00
77                77                ;
78 FFFF00 00000002 78 MSR_TIM   .RES.W   1          ;Measure time
79 FFFF02 00000002 79 CNT_DATA  .RES.W   1          ;count result
80 FFFF04 00000001 80 CNT_OVR   .RES.B   1
81          00000000 81 ERR_F     .EQU     0          ;Error flag
82                82                ;
83                83                ;*****
84                84                ;*
85                85                ;*      SYMBOL DEFINITIONS
86                86                ;*
87                87                ;*****
88                88                ;
89          00FFFF60 89 TSTR      .EQU     H'FFFF60   ;Timer start register
90          00FFFF62 90 TMDR      .EQU     H'FFFF62   ;Timer mode register
91                91                ;
92          00FFFF64 92 TCR0      .EQU     H'FFFF64   ;Timer control register0

```

2. 基礎編

```

93      00FFFF65      93  TIOR0      .EQU      H'FFFF65      ;Timer I/O control register0
94      00FFFF6A      94  GRA0      .EQU      H'FFFF6A      ;General register 0A
95
96      00FFFF79      96  TIOR2      .EQU      H'FFFF79      ;Timer I/O control register2
97      00FFFF7A      97  TIER2      .EQU      H'FFFF7A      ;Timer interrupt enable
                                ;register2
98      00FFFF7B      98  TSR2      .EQU      H'FFFF7B      ;Timer status register2
99      00000000      99  IMPA2      .EQU      0              ;Inputcapture flag2
100     00000002      100 OVF2      .EQU      2              ;Over/Under flow flag2
101     00FFFF7C      101 TCNT2      .EQU      H'FFFF7C      ;Timer counter 2
102     00FFFF7E      102 GRA2      .EQU      H'FFFF7E      ;General register A2
103
104     ;*****
105     ;*
106     ;*          MAIN PROGRAM : PHACNTMN          *
107     ;*
108     ;*****
109
110     010000          110          .SECTION  PROG,CODE,LOCATE=H'010000
111
112     00010000      112  PHACNTMN: .EQU      $
113     010000 7A0700FFFEFE 113  MOV.L      #H'FFFEFE,SP ;Initialize stack pointer
114
115     010006 F8FD      115          MOV.B      #B'11111101,R0L
116     010008 387A      116          MOV.B      R0L,@TIER2 ;Initialize TIER0(1->IMIEA)
117     01000A F8A0      117          MOV.B      #B'10100000,R0L
118     01000C 3864      118          MOV.B      R0L,@TCR0 ;Initialize TCR0
119     01000E F88B      119          MOV.B      #B'10001011,R0L
120     010010 3865      120          MOV.B      R0L,@TIOR0 ;Initialize TIOR0
121     010012 6B00FF00 121          MOV.W      @MSR_TIM,R0
122     010016 6B80FF6A 122          MOV.W      R0,@GRA0
123
124     01001A F88F      124          MOV.B      #B'10001111,R0L
125     01001C 3879      125          MOV.B      R0L,@TIOR2 ;Initialize TIOR2
126
127     01001E F8C0      127          MOV.B      #B'11000000,R0L
128     010020 3862      128          MOV.B      R0L,@TMDR ;Set phase counting mode
129
130     010022 79008000 130          MOV.W      #H'8000,R0
131     010026 6B80FF7C 131          MOV.W      R0,@TCNT2 ;Set Initialize data
132     01002A F8E5      132          MOV.B      #B'11100101,R0L
133     01002C 3860      133          MOV.B      R0L,@TSTR ;Start TCNT0,2
134
135     01002E 0700      135          LDC.B      #0,CCR ;Enable interrupt
136     010030 40FE      136  PHACNTMN99 BRA      PHACNTMN99
137

```



```

138          138 ;*****
139          139 ;*
140          140 ;*          NAME : PHACNT1(MEASUREMENT COUNTER) *
141          141 ;*
142          142 ;*****
143          143 ;*
144          144 ;*          ENTRY   : NOTHING
145          145 ;*          RETURNS  : CNT_DATA(COUNTER DATA)
146          146 ;*
147          147 ;*****
148          148 ;
149          00010032 149 PHACNT1 .EQU $
150          010032 6DF0 150          PUSH.W R0          ;Escape register
151          010034 7F7B7200 151          BCLR      #IMFA2,@TSR2 ;Clear IMFA0 request
152          152          ;(0->IMFA0)
153          010038 6B00FF7E 152          MOV.W      #GRA2,R0
154          01003C 6B80FF02 153          MOV.W      R0,@CNT_DATA ;Store count data
155          010040 F8F8 154          MOV.B      #B'11111000,R0L
156          010042 387A 155          MOV.B      R0L,@TIER2
157          010044 F8E0 156          MOV.B      #B'11100000,R0L
158          010046 3860 157          MOV.B      R0L,@TSTR  ;Stop counter
159          010048 6D70 158          POP.W      R0          ;Return register
160          01004A 5670 159          RTE
161          160 ;
162          161 ;*****
163          162 ;*
164          163 ;*          NAME : PHACNT1(MEASUREMENT COUNTER) *
165          164 ;*
166          165 ;*****
167          166 ;*
168          167 ;*          ENTRY   : NOTHING
169          168 ;*          RETURNS  : ERR_F(ERROR FLAG)
170          169 ;*
171          170 ;*****
172          171 ;
173          0001004C 172 PHACNT2 .EQU $
174          01004C 7F7B7220 173          BCLR      #OVF2,@TSR2 ;Clear OVF2 request
175          174          ;(0->OVF2)
176          010050 7F047000 174          BSET      #ERR_F,@CNT_OVR ;Set error flag
177          010054 5670 175          RTE
178          176 ;
179          177          .END
*****TOTAL ERRORS 0
*****TOTAL WARNINGS 0

```

2.3 パルスの High および Low 幅測定

MCU : H8/3003

使用機能 : ITU

2.3.1 仕様

図 2.12 に示すように、パルスの High 幅および Low 幅の時間を測定し、結果を RAM に設定します。16MHz で動作時、パルスの High 幅および Low 幅は 5.2 μ s から 4.08ms まで 62.5ns 単位で測定可能です。

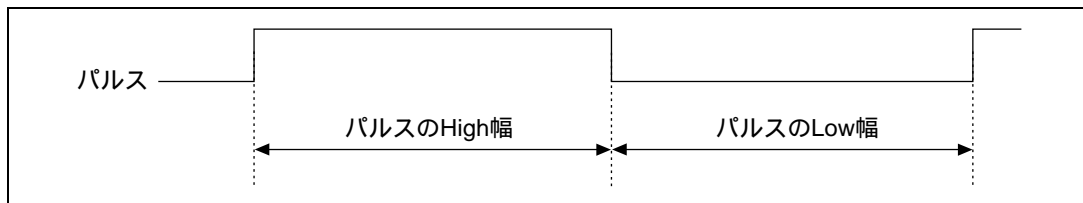


図 2.12 パルス幅測定タイミング

2.3.2 使用機能説明

本タスク例では、ch0 を使用してパルスの High および Low 幅を測定します。

図 2.13 に ch0 のブロック図を示します。本タスクでは、以下の機能を使用します。

- パルスの立ち上がりおよび立ち下がりエッジの検出を行い、そのときのタイマ値を内部レジスタに設定する機能。(インプットキャプチャ)
- インプットキャプチャ発生時タイマカウンタをクリアする機能。
- パルスの立ち上がりおよび立ち下がりエッジ検出時、割り込み処理を起動する機能。

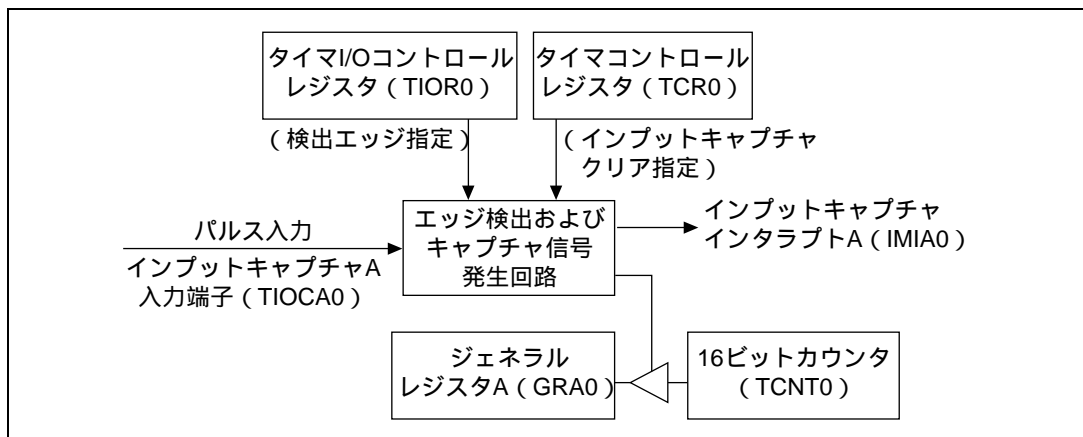


図 2.13 ITU/ch0 ブロック図

表 2.12 に本タスク例の機能割り付けを示します。表 2.12 に示すように ITU の機能を割り付け、パルスの High および Low 幅を測定しています。

表 2.12 ITU 機能割り付け

ITU 機能		機能
ch0	TCR0	カウンタクリア要因を選択する。
	TIOR0	インプットキャプチャ信号の入力エッジを選択する。
	TIOCA0	測定するパルスを入力する。
	GRA0	パルスの立ち上がりおよび立ち下がり時のカウンタ値を検出する。
	IMIA0	パルスの立ち上がりおよび立ち下がりにより、パルスの High および Low 幅測定を起動する。

2.3.3 動作説明

図 2.14 に動作原理を示します。図 2.14 に示すように H8/3003 のハードウェア処理およびソフトウェアの処理によりパルスの High および Low 幅を測定します。

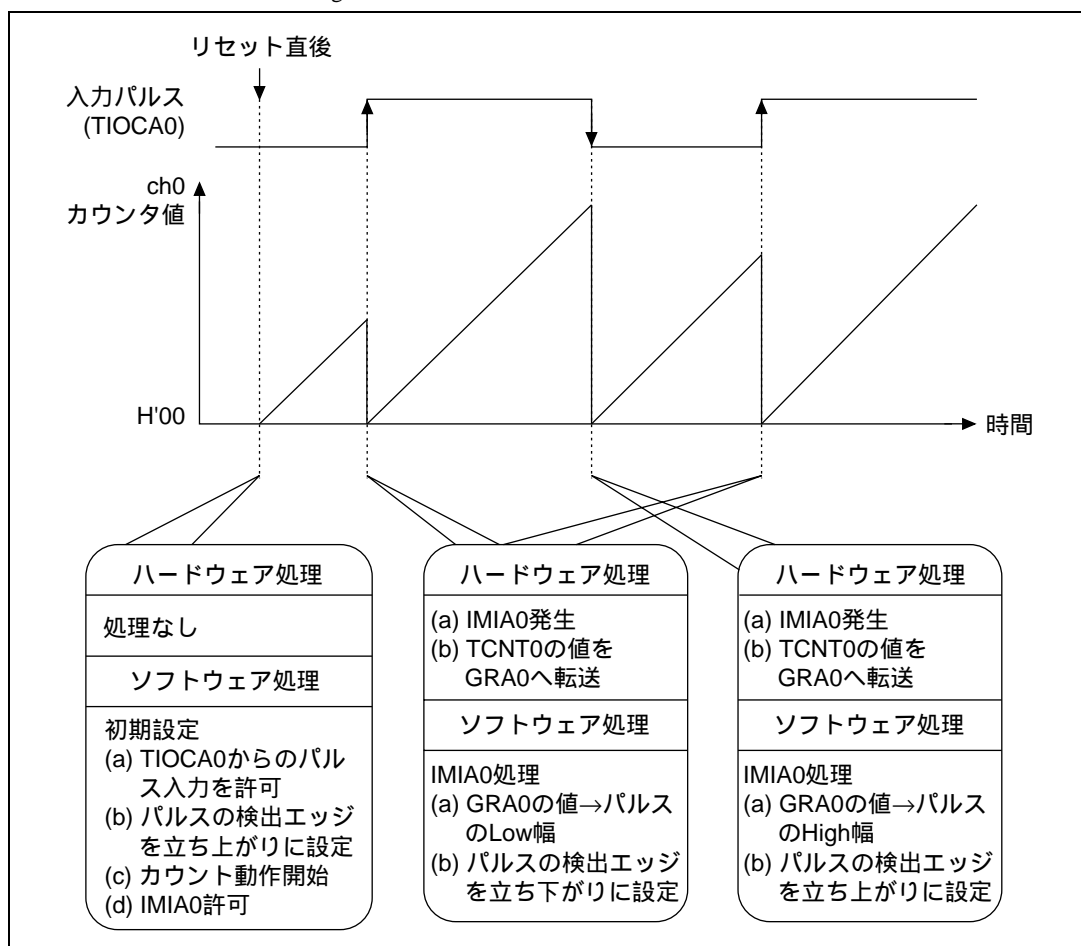


図 2.14 パルス幅測定動作原理

2. 基礎編

2.3.4 ソフトウェア説明

表 2.13 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	PWHLMN	ITU および RAM の初期設定を行う。
パルスの High および Low 幅測定	PWHL1	IMIA により起動し、GRA の値からパルスの High 幅および Low 幅を測定し、RAM に設定する。

表 2.14 引数の説明

ラベル名、レジスタ名	機能	データ長	使用モジュール名	入出力
PWH_HDATA	パルスの High 幅に相当するタイマ値を設定する。 パルスの High 幅は以下の式にて求まる。 パルスの High 幅 (ns) = タイマ値 × 周期 (16MHz 動作時 62.5ns)	1 ワード	パルスの High および Low 幅測定	出力
PWH_LDATA	パルスの Low 幅に相当するタイマ値を設定する。 パルスの Low 幅は以下の式にて求まる。 パルスの Low 幅 (ns) = タイマ値 × 周期 (16MHz 動作時 62.5ns)	1 ワード		

表 2.15 使用内部レジスタ説明

レジスタ名	機能	使用モジュール名	
TSTR	タイマカウンタの動作 / 禁止を設定する。	メインルーチン	
ch0	TCR0	TCNT のカウンタクロックの選択、およびカウンタクリア要因をインプットキャプチャ A に設定する。	メインルーチン、パルスの High および Low 幅測定
	TIOR0	パルスの立ち上がりおよび立ち下がり検出により TCNT から GRA への転送を行うように設定する。	メインルーチン
	TIER0	IMFA フラグによる割り込みを許可する。	メインルーチン
	GRA0	パルスの立ち上がりおよび立ち下がり時の TCNT の値が設定され、この値によりパルスの周期を求める。	パルスの High および Low 幅測定
	TSR0	インプットキャプチャ A の発生を示す。	パルスの High および Low 幅測定

表 2.16 使用汎用レジスタ説明

使用モジュール名	レジスタ名	機能
メインルーチン	R0	データ設定時のワークとして使用する。
パルスの High および Low 幅測定	R0	データ設定時のワークとして使用する。

- 使用 RAM 説明
本タスク例では引数以外の RAM は使用していません。

2.3.5 フローチャート

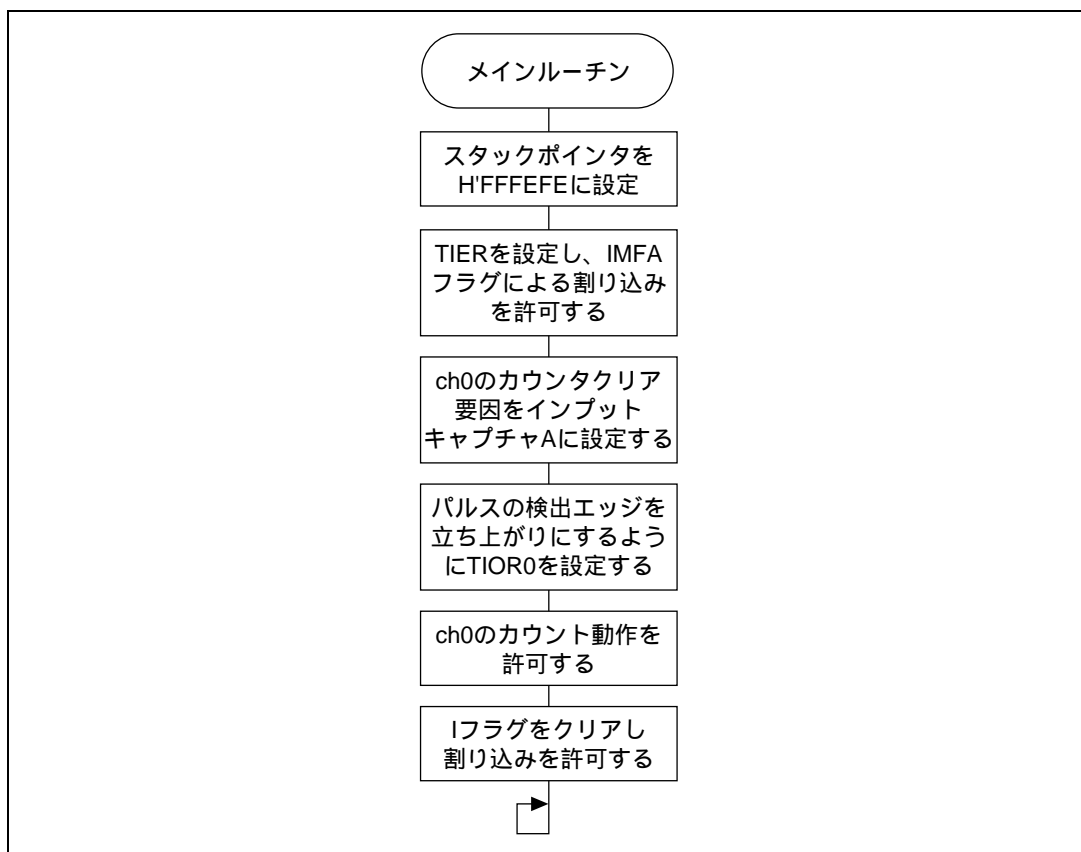


図 2.15 メインルーチン

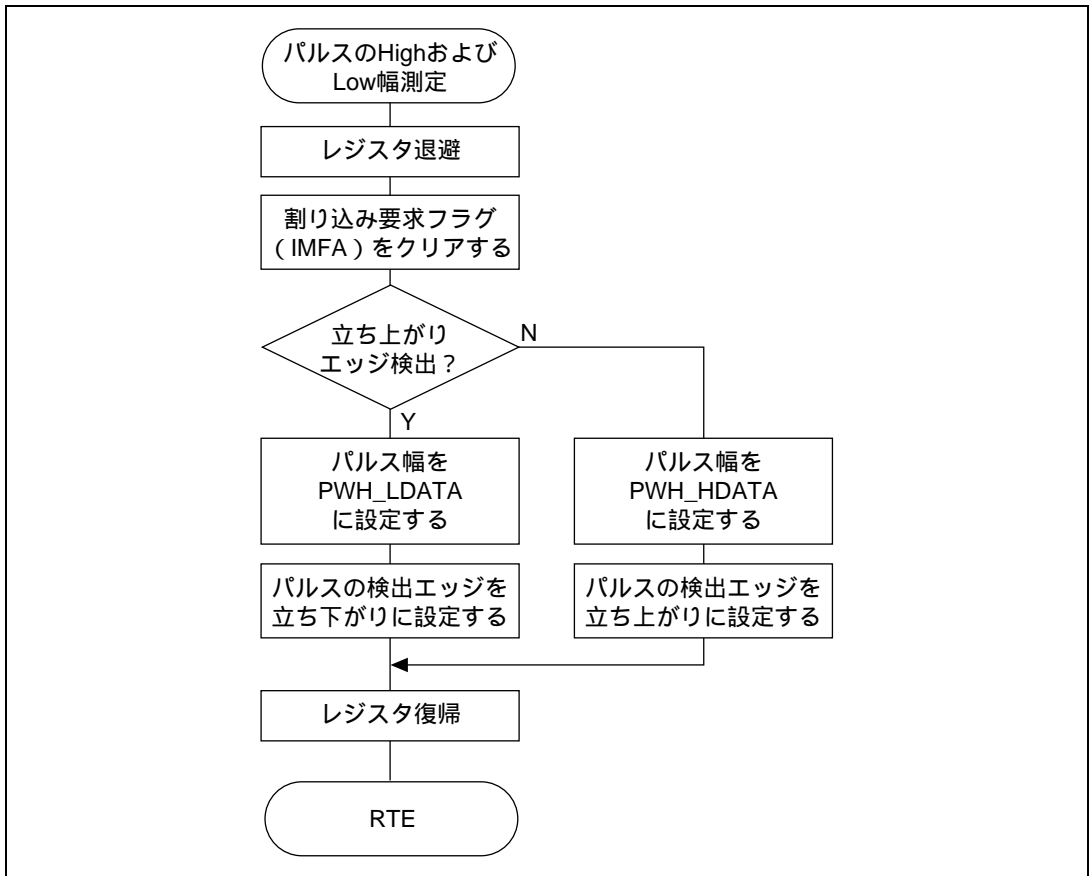


図 2.16 パルスの High および Low 幅測定

2.3.6 プログラムリスト

```

1          1 ;*****
2          2 ;*
3          3 ;*          VECTOR ADDRESS
4          4 ;*
5          5 ;*****
6          6 ;
7 000000   7          .SECTION VECT, CODE, LOCATE=H'000000
8          8 ;
9 000000 00010000  9 RES          .DATA.L PWHLMN
10         10 ;
11 00001C   11          .ORG          H'00001C
12 00001C 00010000  12 NMI          .DATA.L PWHLMN
13 000020 00010000  13 TRAPA0       .DATA.L PWHLMN
14 000024 00010000  14 TRAPA1       .DATA.L PWHLMN
15 000028 00010000  15 TRAPA2       .DATA.L PWHLMN
16 00002C 00010000  16 TRAPA3       .DATA.L PWHLMN
17         17 ;
18 000030   18          .ORG          H'000030
19 000030 00010000  19 IRQ0         .DATA.L PWHLMN
20 000034 00010000  20 IRQ1         .DATA.L PWHLMN
21 000038 00010000  21 IRQ2         .DATA.L PWHLMN
22 00003C 00010000  22 IRQ3         .DATA.L PWHLMN
23 000040 00010000  23 IRQ4         .DATA.L PWHLMN
24 000044 00010000  24 IRQ5         .DATA.L PWHLMN
25 000048 00010000  25 IRQ6         .DATA.L PWHLMN
26 00004C 00010000  26 IRQ7         .DATA.L PWHLMN
27 000050 00010000  27 WOVI0       .DATA.L PWHLMN
28 000054 00010000  28 CMI          .DATA.L PWHLMN
29         29 ;
30 000060   30          .ORG          H'000060
31 000060 0001001A  31 IMIA0       .DATA.L PWHL1
32 000064 00010000  32 IMIB0       .DATA.L PWHLMN
33 000068 00010000  33 OVI0        .DATA.L PWHLMN
34         34 ;
35 000070   35          .ORG          H'000070
36 000070 00010000  36 IMIA1       .DATA.L PWHLMN
37 000074 00010000  37 IMIB1       .DATA.L PWHLMN
38 000078 00010000  38 OVI1        .DATA.L PWHLMN
39         39 ;
40 000080   40          .ORG          H'000080
41 000080 00010000  41 IMIA2       .DATA.L PWHLMN
42 000084 00010000  42 IMIB2       .DATA.L PWHLMN
43 000088 00010000  43 OVI2        .DATA.L PWHLMN
44         44 ;
45 000090   45          .ORG          H'000090

```

2. 基礎編

```
46 000090 00010000      46 IMIA3      .DATA.L  PWHLMN
47 000094 00010000      47 IMIB3      .DATA.L  PWHLMN
48 000098 00010000      48 OVI3       .DATA.L  PWHLMN
49                               49 ;
50 0000A0                               50           .ORG      H'0000A0
51 0000A0 00010000      51 IMIA4      .DATA.L  PWHLMN
52 0000A4 00010000      52 IMIB4      .DATA.L  PWHLMN
53 0000A8 00010000      53 OVI4       .DATA.L  PWHLMN
54                               54 ;
55 0000B0                               55           .ORG      H'0000B0
56 0000B0 00010000      56 DEND0A     .DATA.L  PWHLMN
57 0000B4 00010000      57 DEND0B     .DATA.L  PWHLMN
58 0000B8 00010000      58 DEND1A     .DATA.L  PWHLMN
59 0000BC 00010000      59 DEND1B     .DATA.L  PWHLMN
60 0000C0 00010000      60 DEND2A     .DATA.L  PWHLMN
61 0000C4 00010000      61 DEND2B     .DATA.L  PWHLMN
62 0000C8 00010000      62 DEND3A     .DATA.L  PWHLMN
63 0000CC 00010000      63 DEND3B     .DATA.L  PWHLMN
64 0000D0 00010000      64 ERI0       .DATA.L  PWHLMN
65 0000D4 00010000      65 RXI0       .DATA.L  PWHLMN
66 0000D8 00010000      66 TXI0       .DATA.L  PWHLMN
67 0000DC 00010000      67 TEI0       .DATA.L  PWHLMN
68 0000E0 00010000      68 ERI1       .DATA.L  PWHLMN
69 0000E4 00010000      69 RXI1       .DATA.L  PWHLMN
70 0000E8 00010000      70 TXI1       .DATA.L  PWHLMN
71 0000EC 00010000      71 TEI1       .DATA.L  PWHLMN
72 0000F0 00010000      72 ADI        .DATA.L  PWHLMN
73                               73 ;
74                               74 ;*****
75                               75 ;*                               *
76                               76 ;*          RAM ALLOCATION          *
77                               77 ;*                               *
78                               78 ;*****
79                               79 ;
80 FFFF00                               80           .SECTION  RAM,DATA,LOCATE=H'FFFF00
81                               81 ;
82 FFFF00 00000002      82 PWH_HDATA  .RES.W   1           ;Pulse high width time
83 FFFF02 00000002      83 PWH_LDATA  .RES.W   1           ;Pulse low width time
84                               84 ;
85                               85 ;*****
86                               86 ;*                               *
87                               87 ;*          SYMBOL DEFINITIONS          *
88                               88 ;*                               *
89                               89 ;*****
90                               90 ;
91           00FFFF60      91 TSTR       .EQU      H'FFFF60      ;Timer start register
92                               92 ;
```



```

93      00FFFF64      93  TCR0      .EQU      H'FFFF64      ;Timer control register0
94      00FFFF65      94  TIOR0     .EQU      H'FFFF65      ;Timer I/O control register0
95      00000000      95  IOA0      .EQU      0              ;I/O control A0
96      00FFFF66      96  TIER0     .EQU      H'FFFF66      ;Timer interrupt enable
                                ;register0
97      00FFFF67      97  TSR0      .EQU      H'FFFF67      ;Timer status register0
98      00000000      98  IMFA0     .EQU      0              ;Inputcapture flag A0
99      00FFFF6A      99  GRA0      .EQU      H'FFFF6A      ;General register A0
100     ;
101     ;*****
102     ;*
103     ;*          MAIN PROGRAM : PWHLMN
104     ;*
105     ;*****
106     ;
107 010000      107          .SECTION  PROG,CODE,LOCATE=H'010000
108     ;
109      00010000      109 PWHLMN:   .EQU      $
110 010000 7A0700FFFEFE 110          MOV.L      #H'FFFEFE.SP ;Initialize stack pointer
111     ;
112 010006 F8F9      112          MOV.B      #B'11111001,R0L
113 010008 3866      113          MOV.B      R0L,@TIER0 ;Initialize TIER0(1->IMIEA)
114 01000A F8A0      114          MOV.B      #B'10100000,R0L
115 01000C 3864      115          MOV.B      R0L,@TCR0 ;Initialize TCR0
116     ;
117 01000E F88C      117          MOV.B      #B'10001100,R0L
118 010010 3865      118          MOV.B      R0L,@TIOR0 ;Initialize TIOR0
119     ;
120 010012 F8E1      120          MOV.B      #B'11100001,R0L
121 010014 3860      121          MOV.B      R0L,@TSTR ;Start TCNT0
122 010016 0700      122          LDC.B      #0,CCR ;Enable interrupt
123     ;
124 010018 40FE      124 PWHLMN99  BRA        PWHLMN99
125     ;
126     ;*****
127     ;*
128     ;*  NAME : PWHL1(MEASUREMENT PULSE H AND L)
129     ;*
130     ;*****
131     ;*
132     ;*  ENTRY : NOTHING
133     ;*  RETURNS : PWH_HDATA(PULSE HIGH PERIOD DATA)
134     ;*  RETURNS : PWH_LDATA(PULSE LOW PERIOD DATA)
135     ;*
136     ;*****
137     ;
138      0001001A      138 PWHL1     .EQU      $

```

2. 基礎編

```
139 01001A 6DF0          139          PUSH.W  R0          ;Escape register
140 01001C 7F677200      140          BCLR   #IMFA0,@TSR0 ;Clear IMFA0 request
                                ;(0->IMFA0)
141 010020 6B00FF6A      141          MOV.W  @GRA0,R0     ;Load now data
142 010024 7E657300      142          BTST  #IOA0,@TIOR0 ;High period measurement end?
143 010028 58600008      143          BNE   PWHL100      ;Yes
144                                144 ;
145 01002C 6B80FF02      145          MOV.W  R0,@PWH_LDATA ;No, store low data
146 010030 58000004      146          BRA   PWHL101
147                                147 ;
148 010034 6B80FF00      148 PWHL100 MOV.W  R0,@PWH_HDATA ;Yes, store high data
149                                149 ;
150 010038 7F657100      150 PWHL101 BNOT  #IOA0,@TIOR0 ;Invert IOA0
151                                151 ;
152 01003C 6D70          152          POP.W  R0          ;Return register
153 01003E 5670          153          RTE
154                                154
155                                155
156                                156          .END
****TOTAL ERRORS      0
****TOTAL WARNINGS    0
```

2.4 相補 PWM3 相出力

MCU : H8/3003

使用機能 : ITU (相補 PWM モード)

2.4.1 仕様

図 2.17 に示すように正相・逆相がノンオーバーラップの関係にある PWM 波形を 3 相出力します。デューティは 0% ~ 100% まで任意に設定できます。

$$\text{デューティ} = \frac{\text{パルスの High 幅}}{\text{パルスの周期}} \times 100 (\%)$$

外部信号により、ITU 出力端子から通常のポートとして動作します。

16MHz 動作時、出力するパルスの周期は 375ns から 8.19ms の間で任意に設定できます。

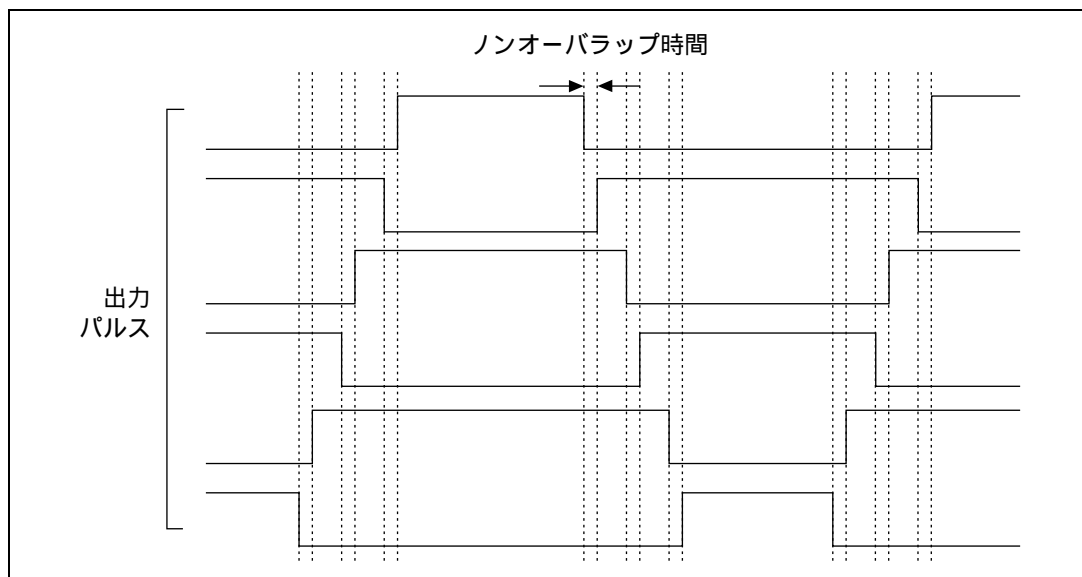


図 2.17 正相・逆相 PWM 3 相出力波形

2.4.2 使用機能説明

本タスク例では、ch3、4 を使用し正相と逆相がノンオーバーラップの関係にある PWM 波形を 3 相出力します。また、ch1 の入力キャプチャにより ch3、4 の出力設定を禁止します。

図 2.18 に本タスク例で使用する ITU/ch3、4 のブロック図を示します。本タスクでは、以下の機能を使用します。

- 正相と逆相がノンオーバーラップの関係にある PWM 波形を 3 相出力する機能。(相補 PWM モード)
- ch1 の入力キャプチャ信号により、ch3、4 の ITU 出力端子から通常ポートに切り換える機能。

2. 基礎編

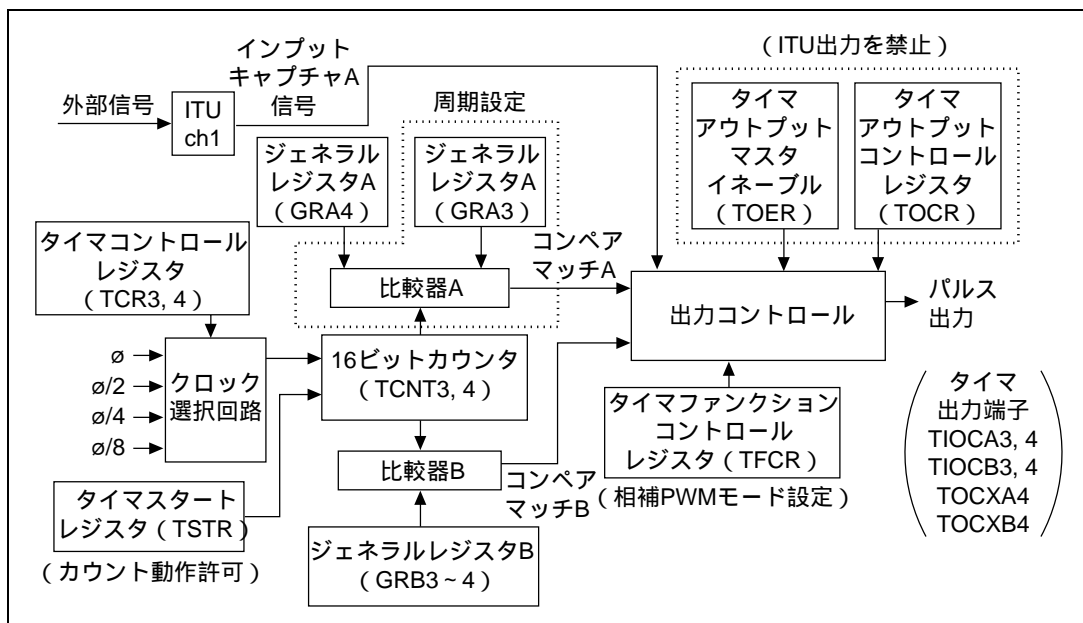


図 2.18 ITU/ch3、4 ブロック図

表 2.17 に本タスク例の機能割り付けを示します。表 2.17 に示すように ITU の機能を割り付け、チャンネル 3 とチャンネル 4 を使用して、正相・逆相がノンオーバーラップの関係にある PWM 波形を 3 相出力します。

表 2.17 ITU 機能割り付け

ITU 機能	機能	
TSTR	TCNT のカウント動作を許可 / 禁止する。	
TFCR	リセット同期 PWM モードに設定する。	
TOER	ch3、4 の出力設定を許可 / 禁止する。	
TOCR	相補 PWM モード出力を外部トリガにより禁止する。	
ch3	TCR3	TCNT に入力するクロックを選択する。
	TIOCA3	正相パルスを出力する。
	TIOCB3	逆相パルスを出力する。
	GRA3	出力パルスの 1/2 周期を設定する。
ch4	TCR4	TCNT に入力するクロックを選択する。
	TIOCA4	正相パルスを出力する。
	TOCXA4	
	TIOCB4	逆相パルスを出力する。
	TOCXB4	
	GRA4	TIOCA、および TOCXA 端子から出力されるパルスの変化点を設定する。
GRB4	TIOCB、および TOCXB 端子から出力されるパルスの変化点を設定する。	

2.4.3 動作説明

図 2.19 に動作原理を示します。図 2.19 に示すように H8/3003 のハードウェア処理およびソフトウェアの処理により相補 PWM 波形を出力します。

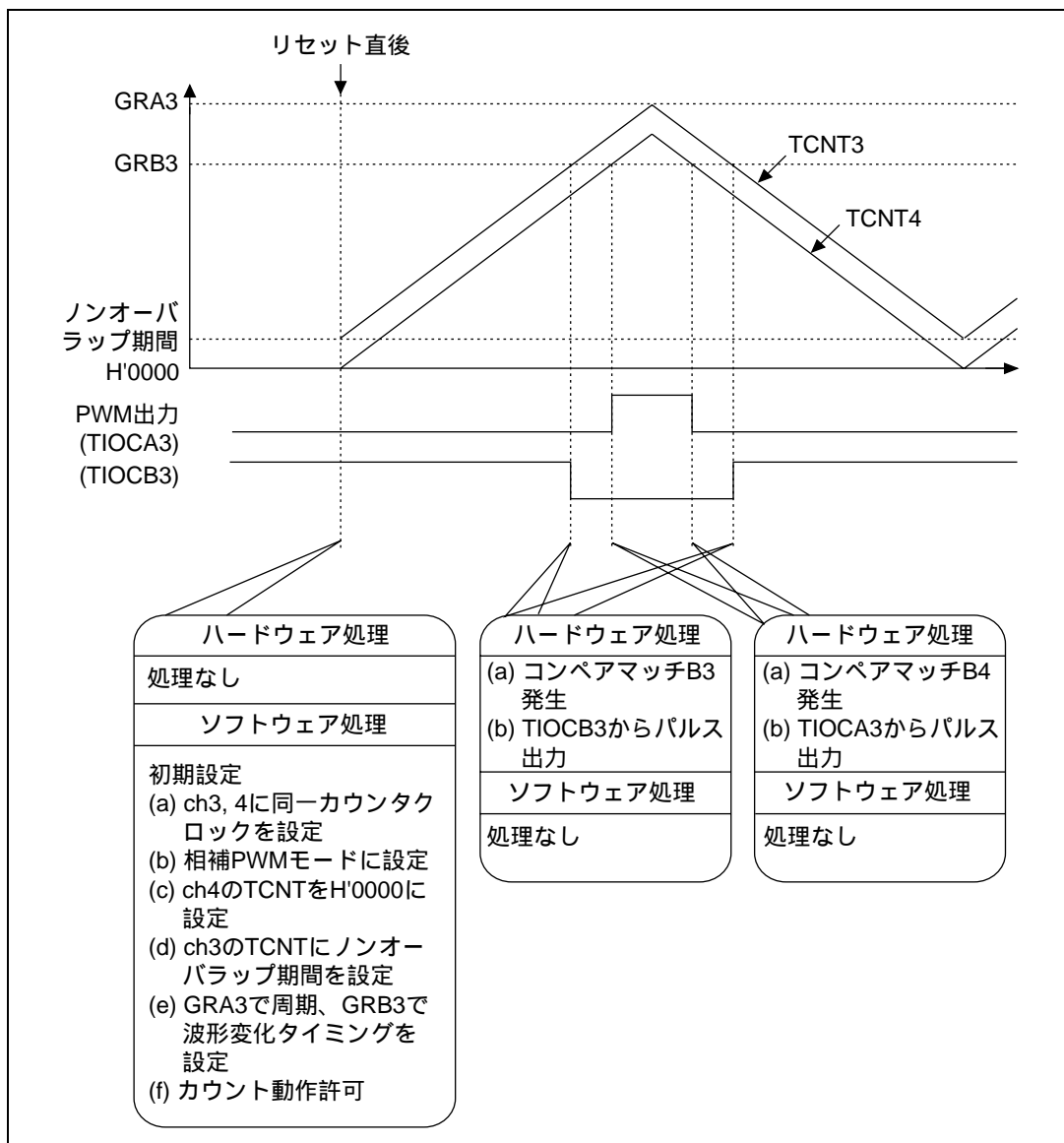


図 2.19 相補 PWM 1 相出力動作原理

図 2.20 に動作原理を示します。図 2.20 に示すように H8/3003 のハードウェア処理およびソフトウェアの処理により相補 PWM 波形を 3 相出力します。

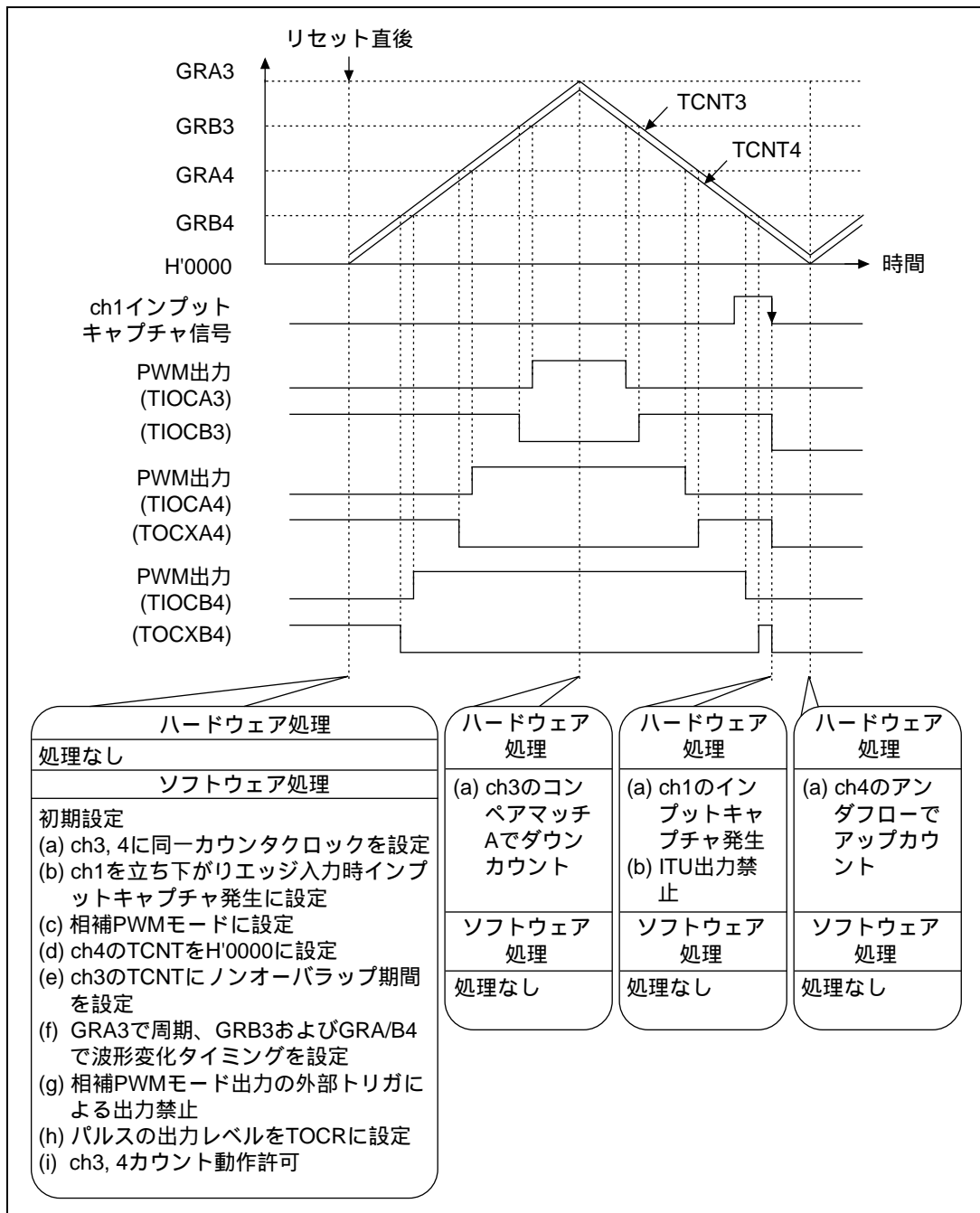


図 2.20 相補 PWM 3 相出力動作原理

2.4.4 ソフトウェア説明

表 2.18 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	CMPWM3MN	モード設定、出力パルスの周期および波形変化タイミングの設定、RAMの初期設定を行う。

表 2.19 引数の説明

ラベル名、レジスタ名	機能	データ長	使用モジュール名	入出力
PUL_CYC	PWMパルスの1/2周期に相当するタイム値を設定する。 周期は以下の式にて求まる。 周期 (ns) = タイマ値 × 周期 (16MHz 動作時 62.5ns) TCNT3の上限値-1	1ワード	メインルーチン	入力
PUL_TIMB3 PUL_TIMA4 PUL_TIMB4	PWMパルスの波形変化タイミングに相当するタイム値を設定する。 波形変化タイミングは以下の式にて求まる。 波形変化タイミング (ns) = タイマ値 × 周期 (16MHz 動作時 62.5ns) ただし、TCNT3とTCNT4がコンペアマッチする範囲で設定する。	1ワード	メインルーチン	入力
NON_OVR	ノンオーバーラップ期間を設定する。	1ワード	メインルーチン	入力

表 2.20 使用内部レジスタ説明

レジスタ名	機能	使用モジュール名	
TSTR	タイマカウンタの動作/禁止を設定する。	メインルーチン	
TFCR	ch3、4を相補PWMモードに設定する。	メインルーチン	
TOER	ch3、4の出力設定を許可/禁止する。	メインルーチン	
TOCR	相補PWMモードの外部トリガによる出力を禁止する。	メインルーチン	
ch1	TIOR1	パルスの立ち下がりエッジ入力によりインプットキャプチャA発生を選択する。	メインルーチン
ch3	TCNT3	ノンオーバーラップ期間を設定する。	メインルーチン
	TCR3	TCNTに入力するクロックを選択する。	メインルーチン
	GRA3	出力パルスの1/2周期を設定する。	メインルーチン
	GRB3	PWM出力波形変化タイミングを設定する。	メインルーチン
ch4	TCNT4	H'0000の初期値を設定する。	メインルーチン
	TCR4	TCNTに入力するクロックを選択する。	メインルーチン
	GRA4	PWM出力波形変化タイミングを設定する。	メインルーチン
	GRB4	PWM出力波形変化タイミングを設定する。	メインルーチン

表 2.21 使用汎用レジスタ説明

使用モジュール名	レジスタ名	機能
メインルーチン	R0	データ設定時のワークとして使用する。

- 使用 RAM 説明
本タスク例では引数以外のRAMは使用していません。

2.4.5 フローチャート



図 2.21 メインルーチン

2.4.6 プログラムリスト

```

1          1 ;*****
2          2 ;*
3          3 ;*          VECTOR ADDRESS
4          4 ;*
5          5 ;*****
6          6 ;
7 000000   7          .SECTION VECT, CODE, LOCATE=H'000000
8          8 ;
9 000000 00010000  9 RES          .DATA.L  CMPWM3MN
10         10 ;
11 00001C   11          .ORG          H'00001C
12 00001C 00010000  12 NMI          .DATA.L  CMPWM3MN
13 000020 00010000  13 TRAPA0       .DATA.L  CMPWM3MN
14 000024 00010000  14 TRAPA1       .DATA.L  CMPWM3MN
15 000028 00010000  15 TRAPA2       .DATA.L  CMPWM3MN
16 00002C 00010000  16 TRAPA3       .DATA.L  CMPWM3MN
17         17 ;
18 000030   18          .ORG          H'000030
19 000030 00010000  19 IRQ0         .DATA.L  CMPWM3MN
20 000034 00010000  20 IRQ1         .DATA.L  CMPWM3MN
21 000038 00010000  21 IRQ2         .DATA.L  CMPWM3MN
22 00003C 00010000  22 IRQ3         .DATA.L  CMPWM3MN
23 000040 00010000  23 IRQ4         .DATA.L  CMPWM3MN
24 000044 00010000  24 IRQ5         .DATA.L  CMPWM3MN
25 000048 00010000  25 IRQ6         .DATA.L  CMPWM3MN
26 00004C 00010000  26 IRQ7         .DATA.L  CMPWM3MN
27 000050 00010000  27 WOVI0       .DATA.L  CMPWM3MN
28 000054 00010000  28 CMI          .DATA.L  CMPWM3MN
29         29 ;
30 000060   30          .ORG          H'000060
31 000060 00010000  31 IMIA0        .DATA.L  CMPWM3MN
32 000064 00010000  32 IMIB0        .DATA.L  CMPWM3MN
33 000068 00010000  33 OVI0         .DATA.L  CMPWM3MN
34         34 ;
35 000070   35          .ORG          H'000070
36 000070 00010000  36 IMIA1        .DATA.L  CMPWM3MN
37 000074 00010000  37 IMIB1        .DATA.L  CMPWM3MN
38 000078 00010000  38 OVI1         .DATA.L  CMPWM3MN
39         39 ;
40 000080   40          .ORG          H'000080
41 000080 00010000  41 IMIA2        .DATA.L  CMPWM3MN
42 000084 00010000  42 IMIB2        .DATA.L  CMPWM3MN
43 000088 00010000  43 OVI2         .DATA.L  CMPWM3MN
44         44 ;
45 000090   45          .ORG          H'000090
46 000090 00010000  46 IMIA3        .DATA.L  CMPWM3MN

```

2. 基礎編

```
47 000094 00010000      47 IMIB3      .DATA.L  CMPWM3MN
48 000098 00010000      48 OVI3       .DATA.L  CMPWM3MN
49                               49 ;
50 0000A0                50           .ORG      H'0000A0
51 0000A0 00010000      51 IMIA4      .DATA.L  CMPWM3MN
52 0000A4 00010000      52 IMIB4      .DATA.L  CMPWM3MN
53 0000A6 00010000      53 OVI4       .DATA.L  CMPWM3MN
54                               54 ;
55 0000B0                55           .ORG      H'0000B0
56 0000B0 00010000      56 DEND0A     .DATA.L  CMPWM3MN
57 0000B4 00010000      57 DEND0B     .DATA.L  CMPWM3MN
58 0000B8 00010000      58 DEND1A     .DATA.L  CMPWM3MN
59 0000BC 00010000      59 DEND1B     .DATA.L  CMPWM3MN
60 0000C0 00010000      60 DEND2A     .DATA.L  CMPWM3MN
61 0000C4 00010000      61 DEND2B     .DATA.L  CMPWM3MN
62 0000C8 00010000      62 DEND3A     .DATA.L  CMPWM3MN
63 0000CC 00010000      63 DEND3B     .DATA.L  CMPWM3MN
64 0000D0 00010000      64 ERI0       .DATA.L  CMPWM3MN
65 0000D4 00010000      65 RXI0       .DATA.L  CMPWM3MN
66 0000D8 00010000      66 TXI0       .DATA.L  CMPWM3MN
67 0000DC 00010000      67 TEI0       .DATA.L  CMPWM3MN
68 0000E0 00010000      68 ERI1       .DATA.L  CMPWM3MN
69 0000E4 00010000      69 RXI1       .DATA.L  CMPWM3MN
70 0000E8 00010000      70 TXI1       .DATA.L  CMPWM3MN
71 0000EC 00010000      71 TEI1       .DATA.L  CMPWM3MN
72 0000F0 00010000      72 ADI        .DATA.L  CMPWM3MN
73                               73 ;
74                               74 ;*****
75                               75 ;*                      *
76                               76 ;*          RAM ALLOCATION          *
77                               77 ;*                      *
78                               78 ;*****
79                               79 ;
80 FFFF00                80           .SECTION  RAM,DATA,LOCATE=H'FFFF00
81                               81 ;
82 FFFF00 00000002      82 NON_OVR    .RES.W   1           ;Non-overlap time
83                               83 ;
84 FFFF02 00000002      84 PUL_CYC    .RES.W   1           ;Pulse cycle time
85                               85 ;
86 FFFF04 00000002      86 PUL_TIMB3  .RES.W   1           ;Pulse set/reset time
87 FFFF06 00000002      87 PUL_TIMA4  .REW.W   1           ;Pulse set/reset time
88 FFFF08 00000002      88 PUL_TIMB4  .RES.W   1           ;Pulse set/reset time
89                               89 ;
90                               90 ;*****
91                               91 ;*                      *
92                               92 ;*          SYMBOL DEFINITIONS          *
93                               93 ;*                      *
```

```

94          94 ;*****
95          95 ;
96          00FFFF60 96 TSTR      .EQU    H'FFFF60 ;Timer start register
97          00FFFF63 97 TFCR      .EQU    H'FFFF63 ;Timer function control register
98          00FFFF90 98 TOER      .EQU    H'FFFF90 ;Timer output master enable
                                     ;register
99          00FFFF91 99 TOCR      .EQU    H'FFFF91 ;Timer output control register
100         100 ;
101         00FFFF6E 101 TCR1      .EQU    H'FFFF6E ;Timer control register 0
102         00FFFF6F 102 TIOR1     .EQU    H'FFFF6F ;Timer I/O control register 0
103         103 ;
104         00FFFF82 104 TCR3      .EQU    H'FFFF82 ;Timer control register 3
105         00FFFF86 105 TCNT3     .EQU    H'FFFF86 ;Timer counter 3
106         00FFFF88 106 GRA3      .EQU    H'FFFF88 ;General register A3
107         00FFFF8A 107 GRB3      .EQU    H'FFFF8A ;General register B3
108         108 ;
109         00FFFF92 109 TCR4      .EQU    H'FFFF92 ;Timer control register 4
110         00FFFF96 110 TCNT4     .EQU    H'FFFF96 ;Timer counter 4
111         00FFFF98 111 GRA4      .EQU    H'FFFF98 ;General register A4
112         00FFFF9A 112 GRB4      .EQU    H'FFFF9A ;General register B4
113         113 ;
114         114 ;*****
115         115 ;*
116         116 ;*          MAIN PROGRAM : CMPWM3MN
117         117 ;*
118         118 ;*****
119         119 ;
120         010000 120          .SECTION  PROG, CODE, LOCATE=H'010000
121         121 ;
122         00010000 122 CMPWM3MN .EQU    $
123         010000 7A0700FFFEFE 123 MOV.L    #H'FFFEFE, SP ;Initialize stack pointer
124         010006 F8E0 124 MOV.B    #B'11100000, R0L
125         010008 3860 125 MOV.B    R0L, @TSTR ;Stop TCNT3,4
126         01000A F880 126 MOV.B    #B'10000000, R0L
127         01000C 3882 127 MOV.B    R0L, @TCR3 ;Initialize TCR3,4
128         01000E 3892 128 MOV.B    R0L, @TCR4
129         129 ;
130         010010 F88D 130 MOV.B    #B'10001101, R0L
131         010012 386F 131 MOV.B    R0L, @TIOR1 ;Initialize TIOR1
132         010014 F8E0 132 MOV.B    #B'11100000, R0L
133         010016 3863 133 MOV.B    R0L, @TFCR ;Set complementary PWM mode
134         010018 79000000 134 MOV.W    #H'0000, R0
135         01001C 6B80FF96 135 MOV.W    R0, @TCNT4 ;Initialize TCNT4
136         136 ;
137         010020 6B00FF00 137 MOV.W    @NON_OVR, R0
138         010024 6B80FF86 138 MOV.W    R0, @TCNT3 ;Set non-overlap time
139         139 ;

```

2. 基礎編

```
140 010028 6B00FF02      140          MOV.W    @PUL_CYC,R0
141 01002C 6B80FF88      141          MOV.W    R0,@GRA3    ;Set pulse cycle time
142 010030 6B00FF06      142          MOV.W    @PUL_TIMA4,R0
143 010034 6B80FF98      143          MOV.W    R0,@GRA4    ;Set pulse set/reset time
144 010038 6B00FF04      144          MOV.W    @PUL_TIMB3,R0
145 01003C 6B80FF8A      145          MOV.W    R0,@GRB3    ;Set pulse set/reset time
146 010040 6B00FF08      146          MOV.W    @PUS_TIMB4,R0
147 010044 6B80FF9A      147          MOV.W    R0,@GRB4    ;Set pulse set/reset time
148                                148 ;
149 01004B F8EF          149          MOV.B    #B'11101111,R0L
150 01004A 3891          150          MOV.B    R0L,@TOCR    ;Set outside trigger disable
151 01004C F8FF          151          MOV.B    #B'11111111,R0L
152 01004E 3890          152          MOV.B    R0L,@TOER    ;Set master enable all port
153                                153 ;
154 010050 F8F8          154          MOV.B    #B'11111000,R0L
155 010052 3860          155          MOV.B    R0L,@TSTR    ;Set TCNT1,3,4 start data
156                                156
157 010054 40FE          157 CMPWM3MN99 BRA    CMPWM3MN99
158                                158
159                                159          .END
****TOTAL ERRORS      0
****TOTAL WARNINGS    0
```

2.5 長周期パルス出力

MCU : H8/3003

使用機能 : ITU (PWM モード)

2.5.1 仕様

図 2.22 に示すように 32 ビットカウンタ動作を行い、パルスの High 幅を変化させデューティ変化する長周期パルスを出します。

デューティは 0% ~ 100% まで、1/65535 の分解能で設定できます。

16MHz 動作時、パルスの周期は 8.19ms から 268.4s まで 4.09ms 単位で設定可能です。

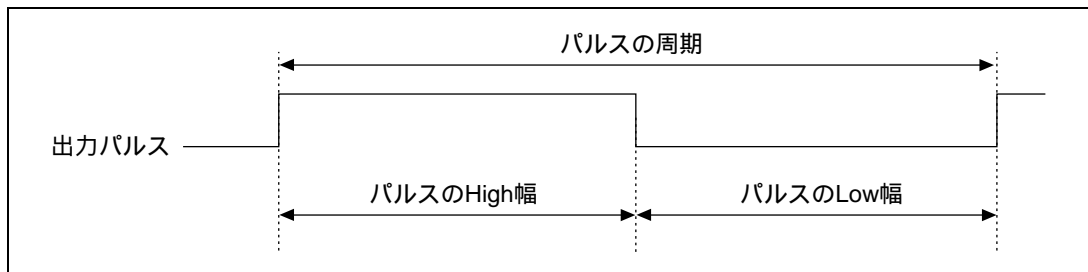


図 2.22 長周期パルス出力例

2.5.2 使用機能説明

本タスク例では ch0、3 を使用して 32 ビットカウンタ動作を行い、ch3 から長周期パルスを出します。

図 2.23 に本タスク例で使用する ITU/ch0 のブロック図を示します。ch0 では以下の機能を使用し、ch3 のタイマカウンタへ入力する外部クロックを生成します。

- ソフトウェアを介さずハードウェアで自動的にパルスを出力する機能。(アプトブットコンペア)
- コンペアマッチが起きるごとに出力が反転する機能。(トグル出力)

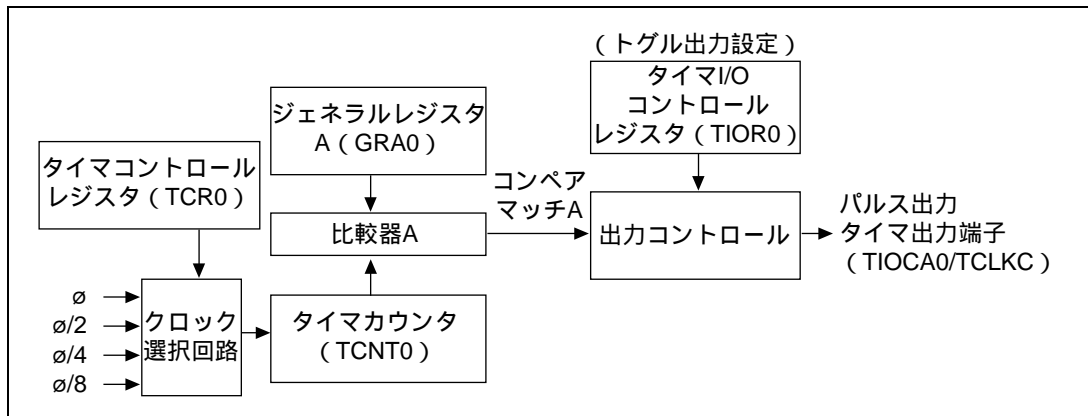


図 2.23 ITU/ch0 ブロック図

2. 基礎編

図 2.24 に本タスク例で使用する ITU/ch3 のブロック図を示します。ch3 では以下の機能を使用し、デューティパルスを出力します。

- パルス出力端子が自動的に PWM 端子となる機能。(PWM モード)

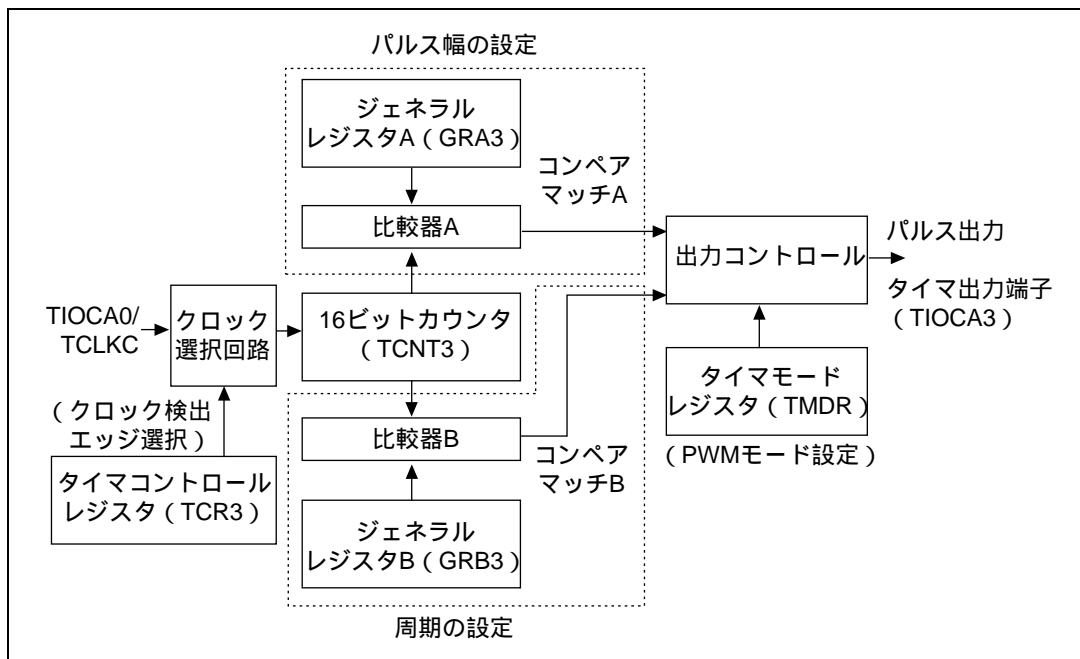


図 2.24 ITU/ch3 ブロック図

表 2.22 に本タスク例の機能割り付けを示します。表 2.22 に示すように ITU の機能を割り付け、32 ビットの動作をするタイマカウンタを生成します。

表 2.22 ITU 機能割り付け

ITU 機能		機能
TMDR		PWM モードに設定する。
ch0	TCR0	TCNT に入力するクロック、およびカウンタクリア要因を選択する。
	TIOR0	パルスの出力レベルを切り換える。
	TIOCA0	パルスを出力する。
	GRA0	パルスの 1/2 周期を設定する。
ch3	TCR3	TCNT に入力するクロック、およびカウンタクリア要因を選択する。
	TIOCA3	パルスを出力する。
	GRA3	パルス幅を設定する。
	GRB3	パルスの周期を設定する。

2.5.3 動作説明

図 2.25 に ITU/ch0 の動作原理を示します。図 2.25 に示すように H8/3003 のハードウェア処理およびソフトウェアの処理によりパルスを出力し、ch3 の外部クロックを生成します。

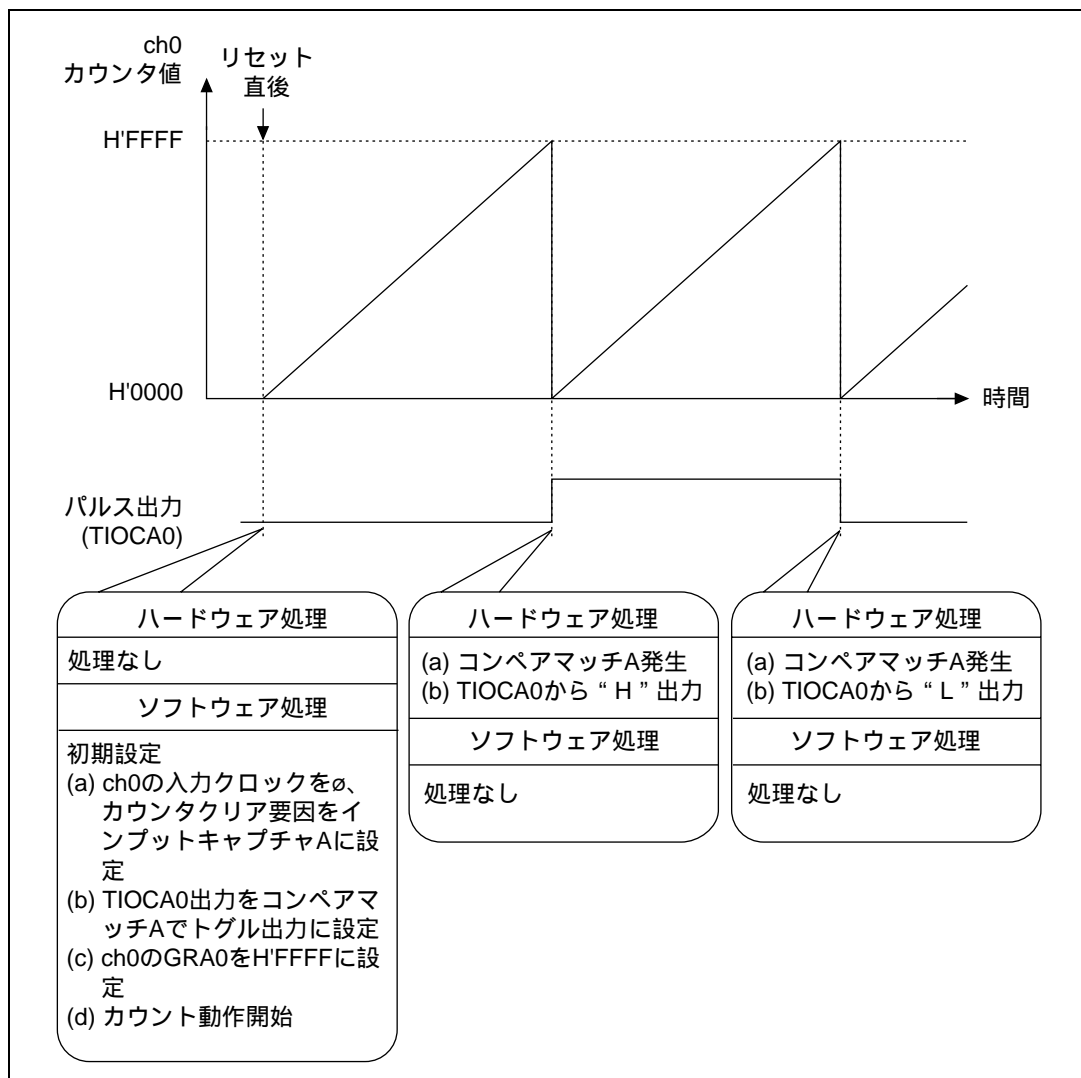


図 2.25 パルス出力動作原理 (ITU/ch0)

2. 基礎編

図 2.26 に ITU/ch 0、3 の動作原理を示します。図 2.26 に示すように H8/3003 のハードウェア処理およびソフトウェアの処理により長周期パルスを出力します。

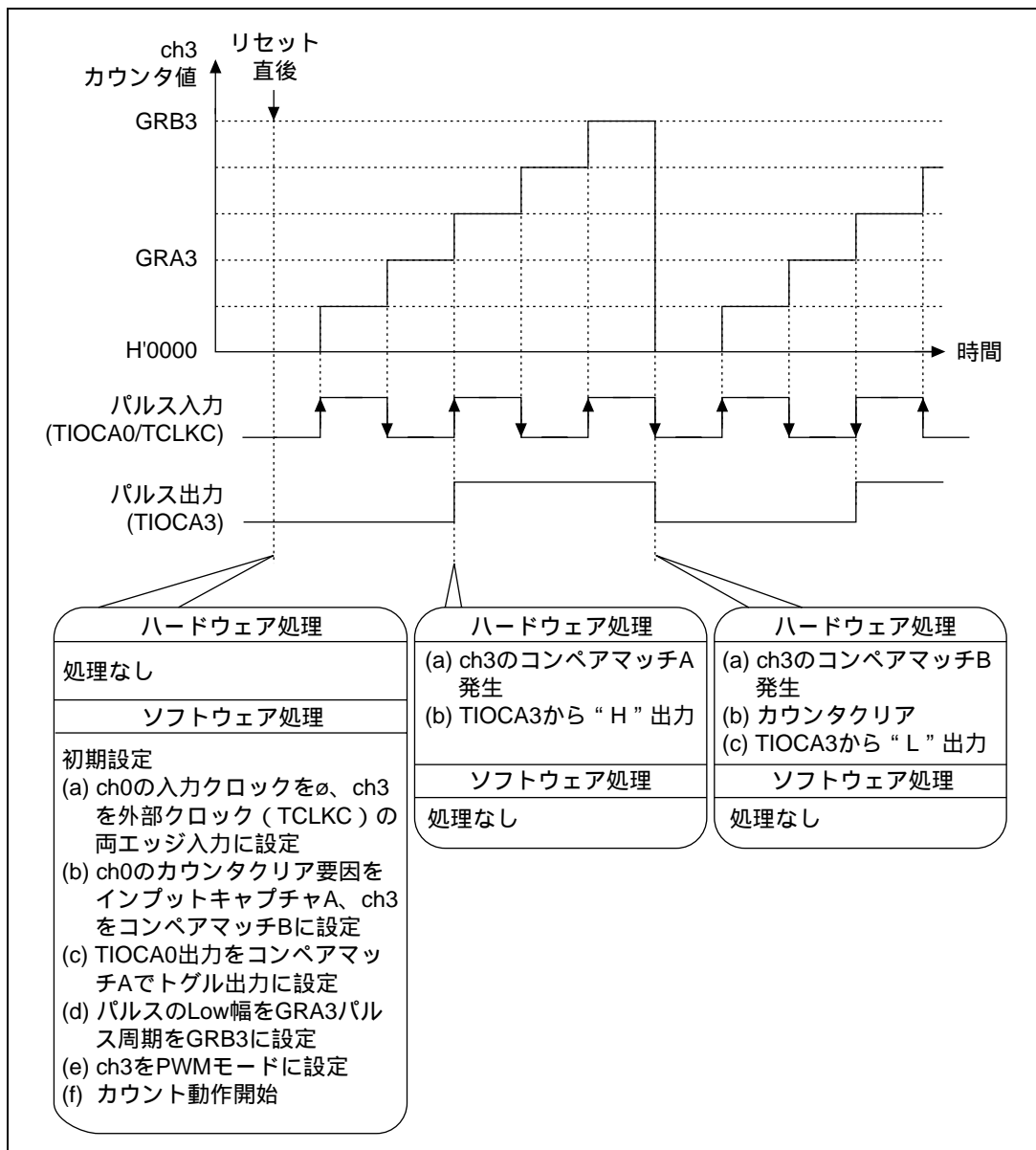


図 2.26 パルス出力動作原理 (ITU/ch0、ch3)

2.5.4 ソフトウェア説明

表 2.23 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	LPULMN	ch0、3 のカウンタを使用して 32 ビットカウンタ動作を行い、長周期パルスを出力します。

表 2.24 引数の説明

ラベル名、レジスタ名	機能	データ長	使用モジュール名	入出力
LPUL_WID	出力パルスの Low 幅に相当するタイマ値を設定する。 パルスの Low 幅は以下の式にて求まる。 $Low (ms) = \text{タイマ値} \times \text{外部クロック} (16\text{MHz 動作時 } 4.09\text{ms})$	1 ワード	メインルーチン	入力
LPUL_CYC	出力パルスの周期に相当するタイマ値を設定する。 周期は以下の式にて求まる。 $周期 (ms) = \text{タイマ値} \times \text{外部クロック} (16\text{MHz 動作時 } 4.09\text{ms})$	1 ワード	メインルーチン	入力

表 2.25 使用内部レジスタ説明

レジスタ名	機能	使用モジュール名	
TSTR	タイマカウンタの動作 / 禁止を設定する。	メインルーチン	
TMDR	ch3 を PWM モードに設定する。	メインルーチン	
ch0	TCR0	TCNT に入力するクロックを選択、およびカウンタクリア要因を設定する。	メインルーチン
	TIOR0	コンペアマッチ A 発生時に、TIOCA0 から出力するレベルを設定する。	メインルーチン
	GRA0	ch3 へ出力するパルスの 1/2 周期を設定する。	メインルーチン
ch3	TCR3	TCNT に入力するクロックを選択、およびカウンタクリア要因を設定する。	メインルーチン
	GRA3	パルスの Low 幅を設定する。	メインルーチン
	GRB3	パルスの周期を設定する。	メインルーチン

表 2.26 使用汎用レジスタ説明

使用モジュール名	レジスタ名	機能
メインルーチン	R0	データ設定時のワークとして使用する。

- 使用 RAM 説明
本タスク例では引数以外の RAM は使用していません。

2.5.5 フローチャート

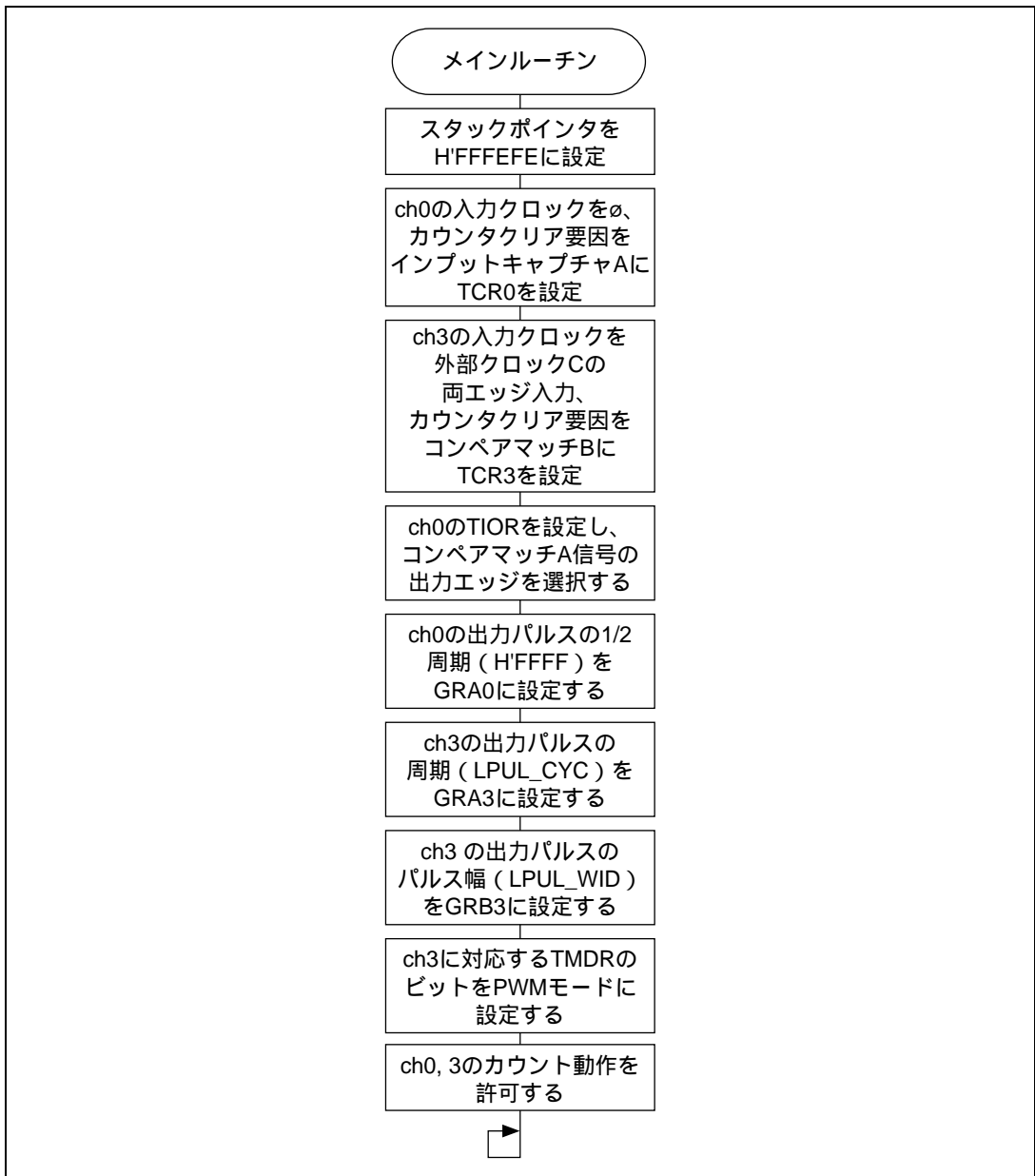


図 2.27 メインルーチン

2.5.6 プログラムリスト

```

1          1  ;*****
2          2  ;*
3          3  ;*          VECTOR ADDRESS
4          4  ;*
5          5  ;*****
6          6  ;
7 000000   7          .SECTION VECT, CODE, LOCATE=H'000000
8          8  ;
9 000000 00010000  9 RES          .DATA.L  LPULMN
10         10 ;
11 00001C   11         .ORG          H'00001C
12 00001C 00010000  12 NMI          .DATA.L  LPULMN
13 000020 00010000  13 TRAPA0       .DATA.L  LPULMN
14 000024 00010000  14 TRAPA1       .DATA.L  LPULMN
15 000028 00010000  15 TRAPA2       .DATA.L  LPULMN
16 00002C 00010000  16 TRAPA3       .DATA.L  LPULMN
17         17 ;
18 000030   18         .ORG          H'000030
19 000030 00010000  19 IRQ0         .DATA.L  LPULMN
20 000034 00010000  20 IRQ1         .DATA.L  LPULMN
21 000038 00010000  21 IRQ2         .DATA.L  LPULMN
22 00003C 00010000  22 IRQ3         .DATA.L  LPULMN
23 000040 00010000  23 IRQ4         .DATA.L  LPULMN
24 000044 00010000  24 IRQ5         .DATA.L  LPULMN
25 000048 00010000  25 IRQ6         .DATA.L  LPULMN
26 00004C 00010000  26 IRQ7         .DATA.L  LPULMN
27 000050 00010000  27 WOVI0        .DATA.L  LPULMN
28 000054 00010000  28 CMI          .DATA.L  LPULMN
29         29 ;
30 000060   30         .ORG          H'000060
31 000060 00010000  31 IMIA0        .DATA.L  LPULMN
32 000064 00010000  32 IMIB0        .DATA.L  LPULMN
33 000068 00010000  33 OVI0         .DATA.L  LPULMN
34         34 ;
35 000070   35         .ORG          H'000070
36 000070 00010000  36 IMIA1        .DATA.L  LPULMN
37 000074 00010000  37 IMIB1        .DATA.L  LPULMN
38 000078 00010000  38 OVI1         .DATA.L  LPULMN
39         39 ;
40 000080   40         .ORG          H'000080
41 000080 00010000  41 IMIA2        .DATA.L  LPULMN
42 000084 00010000  42 IMIB2        .DATA.L  LPULMN
43 000088 00010000  43 OVI2         .DATA.L  LPULMN
44         44 ;
45 000090   45         .ORG          H'000090

```

2. 基礎編

```
46 000090 00010000      46 IMIA3      .DATA.L  LPULMN
47 000094 00010000      47 IMIB3      .DATA.L  LPULMN
48 000098 00010000      48 OVI3       .DATA.L  LPULMN
49
49 ;
50 0000A0
50                    .ORG      H'0000A0
51 0000A0 00010000      51 IMIA4      .DATA.L  LPULMN
52 0000A4 00010000      52 IMIB4      .DATA.L  LPULMN
53 0000A8 00010000      53 OVI4       .DATA.L  LPULMN
54
54 ;
55 0000B0
55                    .ORG      H'0000B0
56 0000B0 00010000      56 DEND0A     .DATA.L  LPULMN
57 0000B4 00010000      57 DEND0B     .DATA.L  LPULMN
58 0000B8 00010000      58 DEND1A     .DATA.L  LPULMN
59 0000BC 00010000      59 DEND1B     .DATA.L  LPULMN
60 0000C0 00010000      60 DEND2A     .DATA.L  LPULMN
61 0000C4 00010000      61 DEND2B     .DATA.L  LPULMN
62 0000C8 00010000      62 DEND3A     .DATA.L  LPULMN
63 0000CC 00010000      63 DEND3B     .DATA.L  LPULMN
64 0000D0 00010000      64 ERI0       .DATA.L  LPULMN
65 0000D4 00010000      65 RXI0       .DATA.L  LPULMN
66 0000D8 00010000      66 TXI0       .DATA.L  LPULMN
67 0000DC 00010000      67 TEI0       .DATA.L  LPULMN
68 0000E0 00010000      68 ERI1       .DATA.L  LPULMN
69 0000E4 00010000      69 RXI1       .DATA.L  LPULMN
70 0000E8 00010000      70 TXI1       .DATA.L  LPULMN
71 0000EC 00010000      71 TEI1       .DATA.L  LPULMN
72 0000F0 00010000      72 ADI        .DATA.L  LPULMN
73
73 ;
74 ;*****
75 ;*
76 ;*          RAM ALLOCATION
77 ;*
78 ;*****
79 ;
80 FFFF00
80                    .SECTION  RAM,DATA,LOCATE=H'FFFF00
81
81 ;
82 FFFF00 00000002      82 LPUL_CYC   .RES.W   1          ;Output pulse cycle time
83 FFFF02 00000002      83 LPUL_WID   .RES.W   1          ;Output pulse low period time
84
84 ;
85 ;*****
86 ;*
87 ;*          SYMBOL DEFINITIONS
88 ;*
89 ;*****
90 ;
91          00FFFF60      91 TSTR       .EQU     H'FFFF60   ;Timer start register
92          00FFFF62      92 TMDR       .EQU     H'FFFF62   ;Timer mode register
```

```

93          93 ;
94      00FFFF64      94 TCR0      .EQU      H'FFFF64      ;Timer control register 0
95      00FFFF65      95 TIOR0     .EQU      H'FFFF65      ;Timer counter 0
96      00FFFF6A      96 GRA0      .EQU      H'FFFF6A      ;General register A0
97          97 ;
98      00FFFF82      98 TCR3      .EQU      H'FFFF82      ;Timer control register 1
99      00FFFF88      99 GRA3      .EQU      H'FFFF88      ;General register A1
100     00FFFF8A      100 GRB3     .EQU      H'FFFF8A      ;General register B1
101          101 ;
102          102 ;*****
103          103 ;*
104          104 ;*          MAIN PROGRAM : LPULMN
105          105 ;*
106          106 ;*****
107          107 ;
108 010000          108          .SECTION  PROG, CODE, LOCATE=H'010000
109          109 ;
110          00010000      110 LPULMN:   .EQU      $
111 010000 7A0700FFFE  111          MOV.L      #H'FFFEFE, SP ;Initialize stack pointer
112 010006 F880          112          MOV.B      #B'10000000, R0L
113 010008 3864          113          MOV.B      R0L, @TCR0      ;Initialize TCR0
114 01000A F8D6          114          MOV.B      #B'11010110, R0L
115 01000C 3882          115          MOV.B      R0L, @TCR3     ;Initialize TCR3
116          116 ;
117 01000E F883          117          MOV.B      #B'10000011, R0L
118 010010 3865          118          MOV.B      R0L, @TIOR0   ;Initialize TIOR0
119          119 ;
120 010012 7900FFFF  120          MOV.W      #H'FFFF, R0
121 010016 6B80FF6A  121          MOV.W      R0, @GRA0      ;Set outside signal cycle
                                     ;time
122 01001A 6B00FF00  122          MOV.W      @LPUL_CYC, R0
123 01001E 6B80FF8A  123          MOV.W      R0, @GRB3     ;Set pulse cycle time
124 010022 6B00FF02  124          MOV.W      @LPUL_WID, R0
125 010026 6B80FF88  125          MOV.W      R0, @GRA3     ;Set pulse low period time
126          126 ;
127 01002A F888          127          MOV.B      #B'10001000, R0L
128 01002C 3862          128          MOV.B      R0L, @TMDR     ;Set PWM mode (chl)
129 01002E F8E9          129          MOV.B      #B'11101001, R0L
130 010030 3860          130          MOV.B      R0L, @TSTR     ;Start TCNT0, 3
131          131
132 010032 40FE          132 LPULMN99  BRA      LPULMN99
133          133
134          134          .END
*****TOTAL ERRORS      0
*****TOTAL WARNINGS    0

```

2.6 正相・逆相 PWM3 相出力

MCU : H8/3003

使用機能 : ITU (リセット同期 PWM モード)

2.6.1 仕様

図 2.28 に示すようにパルス High 幅を変化させ、デューティ変化できるパルス(デューティパルス)を正相と逆相で 3 相出力します。

16MHz 動作時、出力するパルスの周期は 125ns から 4.09ms の間で任意に設定できます。

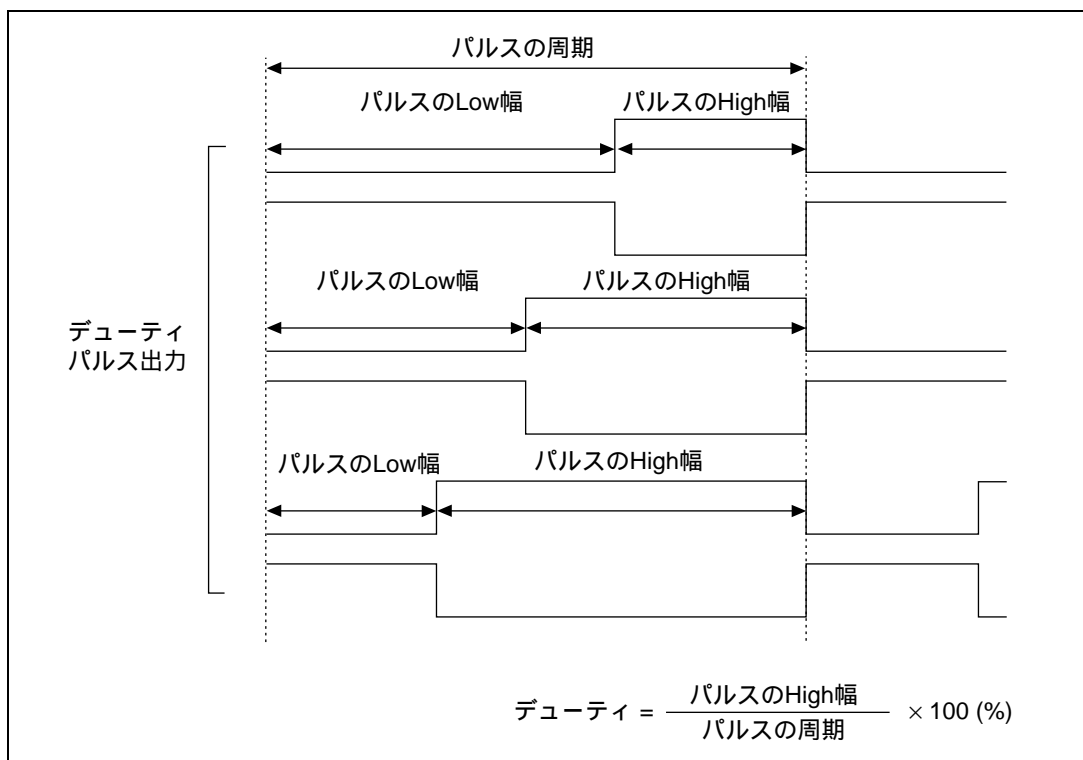


図 2.28 正相・逆相 PWM 3 相出力波形

2.6.2 使用機能説明

本タスク例では、ITU の ch3、4 を使用し、正相と逆相の PWM 波形を 3 相出力します。

図 2.29 に本タスク例で使用する ITU のブロック図を示します。

本タスク例では、ITU の以下の機能を使用しています。

- コンペアマッチ時、タイマカウンタをクリアする機能。
- ch3、4 を組み合わせることにより、一方の波形変化点が共通の関係となる正相と逆相の PWM 波形を出力する機能。(リセット同期 PWM モード)

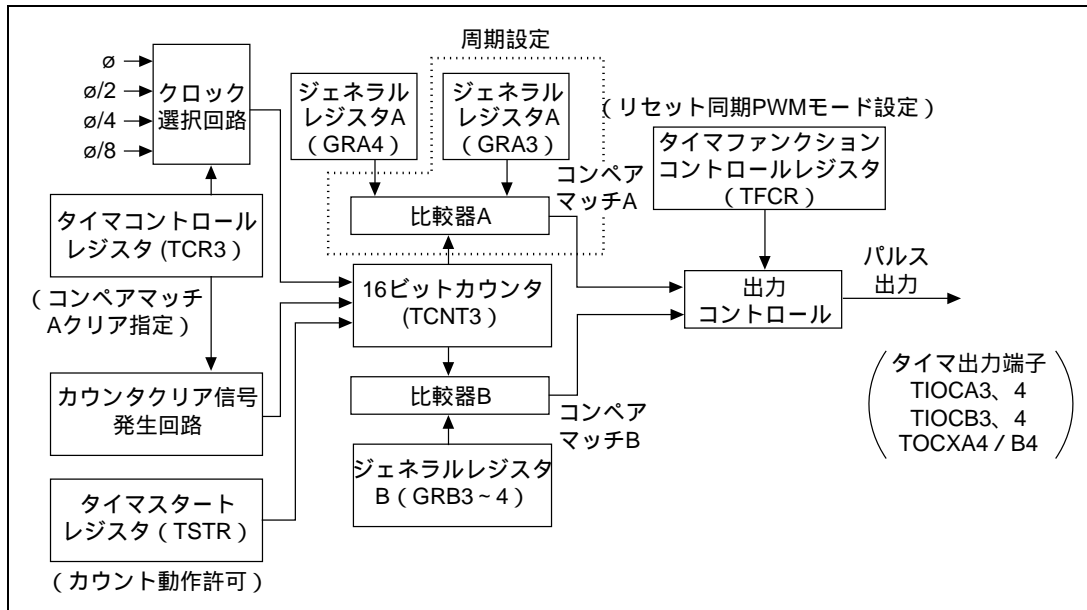


図 2.29 ITU ブロック図

表 2.27 に本タスク例の機能割り付けを示します。表 2.27 に示すように ITU の機能を割り付け、正相・逆相 PWM 波形を 3 相出力します。

表 2.27 ITU 機能割り付け

ITU 機能		機能
TSTR		TCNT のカウント動作を許可 / 禁止する。
TFCR		リセット同期 PWM モードに設定する。
ch3	TCR3	カウンタクロックおよびカウンタクリア要因を設定する。
	TIOCA3	正相パルスを出力する。
	TIOCB3	TIOCA の逆相パルスを出力する。
	GRA3	出力パルスの周期を設定する。
ch4	TIOCA4	正相パルスを出力する。
	TOCXA4	
	TIOCB4	TIOCB、および TOCXB の逆相パルスを出力する。
	TOCXB4	
	GRA4	TIOCA、および TOCXA 端子から出力されるパルスの変化タイミングを設定する。
	GRB4	TIOCB、および TOCXB 端子から出力されるパルスの変化タイミングを設定する。

2.6.3 動作説明

図 2.30 に動作原理を示します。図 2.30 に示すように H8/3003 のハードウェア処理およびソフトウェアの処理により正相・逆相 PWM 波形を 3 相出力します。

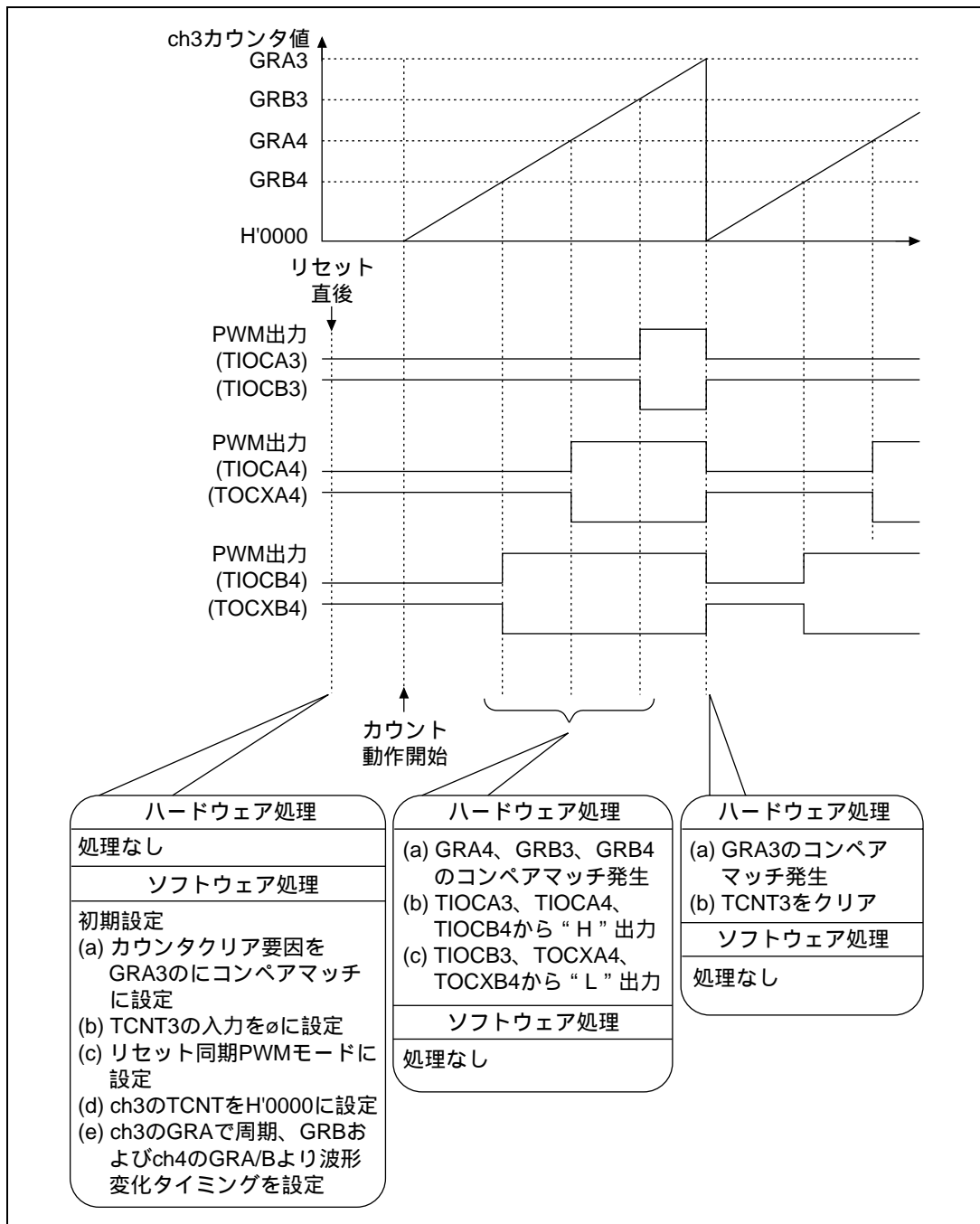


図 2.30 3相 PWM 出力動作原理

2.6.4 ソフトウェア説明

表 2.28 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	NRPWM3MN	出力パルスの波形変化タイミングを GRA4 および GRB3、4、周期を GRA3 に設定し、正相逆相 PWM 波形を 3 相出力します。

表 2.29 引数の説明

ラベル名、レジスタ名	機能	データ長	使用モジュール名	入出力
PUL_CYC	PWM パルスの周期に相当するタイマ値を設定する。 周期は以下の式にて求まる。 周期 (ns) = タイマ値 × 周期 (16MHz 動作時 62.5ns)	1 ワード	メインルーチン	入力
PUL_TIMB3 PUL_TIMA4 PUL_TIMB4	PWM パルスの波形出力タイミングに相当するタイマ値を設定する。 波形出力タイミングは以下の式にて求まる。 波形出力タイミング (ns) = タイマ値 × 周期 (16MHz 動作時 62.5ns)	1 ワード	メインルーチン	入力

表 2.30 使用内部レジスタ説明

レジスタ名	機能	使用モジュール名	
TSTR	タイマカウンタの動作 / 禁止を設定する。	メインルーチン	
TFCR	ch3、4 をリセット同期 PWM モードに設定する。	メインルーチン	
ch3	TCR3	カウンタクロックの選択、およびカウンタクリア要因を GRA のコンペアマッチに設定する。	メインルーチン
	TCNT3	H'0000 の初期値を設定する。	メインルーチン
	GRA3	パルスの周期を設定する。	メインルーチン
	GRB3	PWM 出力波形変化タイミングを設定する。	メインルーチン
ch4	GRA4	PWM 出力波形変化タイミングを設定する。	メインルーチン
	GRB4	PWM 出力波形変化タイミングを設定する。	メインルーチン

表 2.31 使用汎用レジスタ説明

使用モジュール名	レジスタ名	機能
メインルーチン	R0	データ設定時のワークとして使用する。

- 使用 RAM 説明
本タスク例では引数以外の RAM は使用していません。

2.6.5 フローチャート

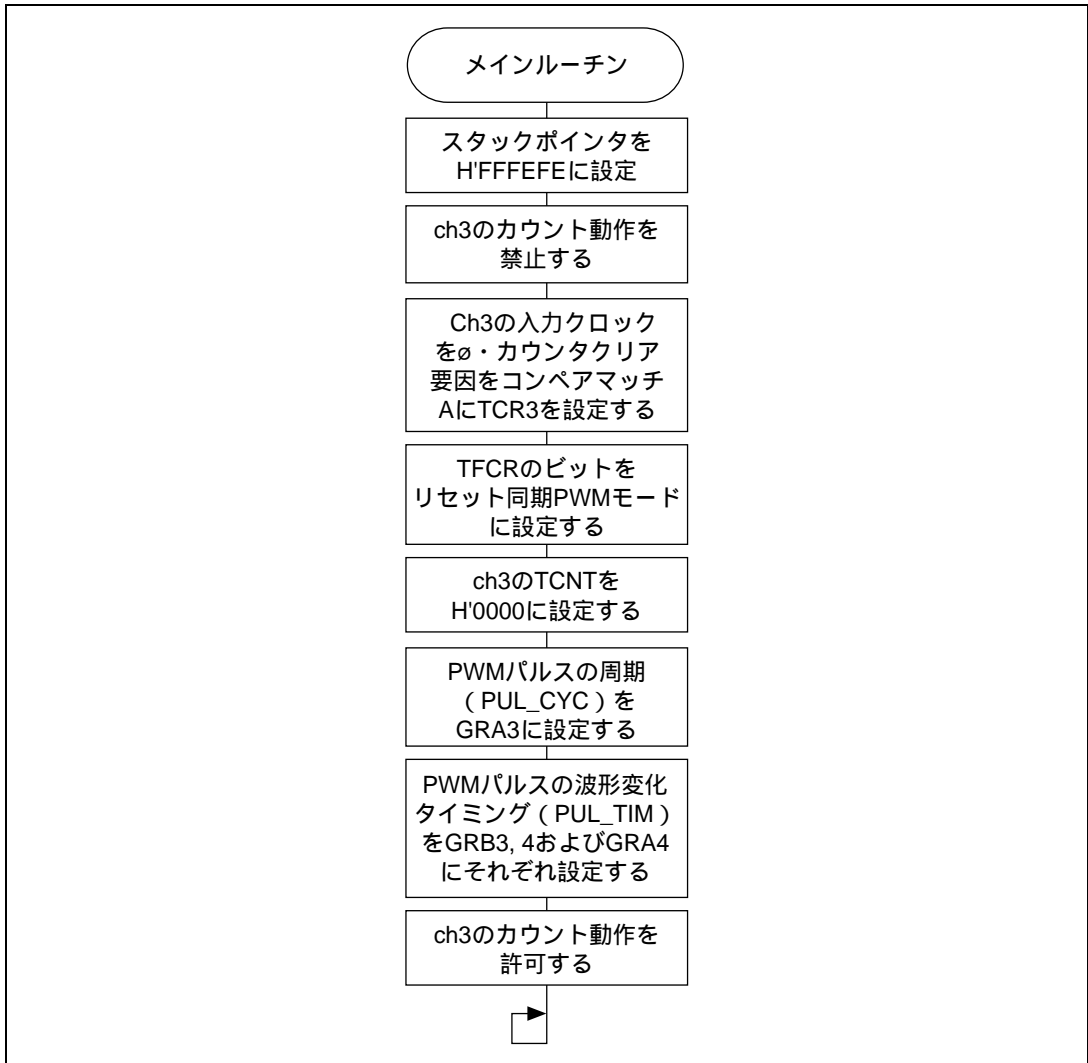


図 2.31 メインルーチン

2.6.6 プログラムリスト

```

1          1  ;*****
2          2  ;*
3          3  ;*          VECTOR ADDRESS
4          4  ;*
5          5  ;*****
6          6  ;
7 000000   7          .SECTION VECT, CODE, LOCATE=H'000000
8          8  ;
9 000000 00010000  9 RES          .DATA.L  NRPWM3MN
10         10 ;
11 00001C   11         .ORG          H'00001C
12 00001C 00010000  12 NMI          .DATA.L  NRPWM3MN
13 000020 00010000  13 TRAPA0       .DATA.L  NRPWM3MN
14 000024 00010000  14 TRAPA1       .DATA.L  NRPWM3MN
15 000028 00010000  15 TRAPA2       .DATA.L  NRPWM3MN
16 00002C 00010000  16 TRAPA3       .DATA.L  NRPWM3MN
17         17 ;
18 000030   18         .ORG          H'000030
19 000030 00010000  19 IRQ0         .DATA.L  NRPWM3MN
20 000034 00010000  20 IRQ1         .DATA.L  NRPWM3MN
21 000038 00010000  21 IRQ2         .DATA.L  NRPWM3MN
22 00003C 00010000  22 IRQ3         .DATA.L  NRPWM3MN
23 000040 00010000  23 IRQ4         .DATA.L  NRPWM3MN
24 000044 00010000  24 IRQ5         .DATA.L  NRPWM3MN
25 000048 00010000  25 IRQ6         .DATA.L  NRPWM3MN
26 00004C 00010000  26 IRQ7         .DATA.L  NRPWM3MN
27 000050 00010000  27 WOVI0        .DATA.L  NRPWM3MN
28 000054 00010000  28 CMI          .DATA.L  NRPWM3MN
29         29 ;
30 000060   30         .ORG          H'000060
31 000060 00010000  31 IMIA0        .DATA.L  NRPWM3MN
32 000064 00010000  32 IMIB0        .DATA.L  NRPWM3MN
33 000068 00010000  33 OVI0         .DATA.L  NRPWM3MN
34         34 ;
35 000070   35         .ORG          H'000070
36 000070 00010000  36 IMIA1        .DATA.L  NRPWM3MN
37 000074 00010000  37 IMIB1        .DATA.L  NRPWM3MN
38 000078 00010000  38 OVI1         .DATA.L  NRPWM3MN
39         39 ;
40 000080   40         .ORG          H'000080
41 000080 00010000  41 IMIA2        .DATA.L  NRPWM3MN
42 000084 00010000  42 IMIB2        .DATA.L  NRPWM3MN
43 000088 00010000  43 OVI2         .DATA.L  NRPWM3MN
44         44 ;
45 000090   45         .ORG          H'000090

```

2. 基礎編

```
46 000090 00010000      46 IMIA3      .DATA.L  NRPWM3MN
47 000094 00010000      47 IMIB3      .DATA.L  NRPWM3MN
48 000098 00010000      48 OVI3       .DATA.L  NRPWM3MN
49
49 ;
50 0000A0
50 .ORG        H'0000A0
51 0000A0 00010000      51 IMIA4      .DATA.L  NRPWM3MN
52 0000A4 00010000      52 IMIB4      .DATA.L  NRPWM3MN
53 0000A8 00010000      53 OVI4       .DATA.L  NRPWM3MN
54
54 ;
55 0000B0
55 .ORG        H'0000B0
56 0000B0 00010000      56 DEND0A     .DATA.L  NRPWM3MN
57 0000B4 00010000      57 DEND0B     .DATA.L  NRPWM3MN
58 0000B8 00010000      58 DEND1A     .DATA.L  NRPWM3MN
59 0000BC 00010000      59 DEND1B     .DATA.L  NRPWM3MN
60 0000C0 00010000      60 DEND2A     .DATA.L  NRPWM3MN
61 0000C4 00010000      61 DEND2B     .DATA.L  NRPWM3MN
62 0000C8 00010000      62 DEND3A     .DATA.L  NRPWM3MN
63 0000CC 00010000      63 DEND3B     .DATA.L  NRPWM3MN
64 0000D0 00010000      64 ERI0       .DATA.L  NRPWM3MN
65 0000D4 00010000      65 RXI0       .DATA.L  NRPWM3MN
66 0000D8 00010000      66 TXI0       .DATA.L  NRPWM3MN
67 0000DC 00010000      67 TEI0       .DATA.L  NRPWM3MN
68 0000E0 00010000      68 ERI1       .DATA.L  NRPWM3MN
69 0000E4 00010000      69 RXI1       .DATA.L  NRPWM3MN
70 0000E8 00010000      70 TXI1       .DATA.L  NRPWM3MN
71 0000EC 00010000      71 TEI1       .DATA.L  NRPWM3MN
72 0000F0 00010000      72 ADI        .DATA.L  NRPWM3MN
73
73 ;
74 ;*****
75 ;*
76 ;*          RAM ALLOCATION
77 ;*
78 ;*****
79 ;
80 FFFF00
80 .SECTION   RAM,DATA,LOCATE=H'FFFF00
81 ;
82 FFFF00 00000002      82 PUL_CYC    .RES.W   1          ;Pulse cycle Time
83 ;
84 FFFF02 00000002      84 PUL_TIMA4  .RES.W   1          ;Pulse change timing GRB3
85 FFFF04 00000002      85 PUL_TIMB3  .RES.W   1          ;Pulse change timing GRA4
86 FFFF06 00000002      86 PUL_TIMB4  .RES.W   1          ;Pulse change timing GRB4
87 ;
88 ;*****
89 ;*
90 ;*          SYMBOL DEFINTIONS
91 ;*
92 ;*****
```

```

93          93 ;
94          00FFFF60 94 TSTR      .EQU    H'FFFF60    ;Timer start register
95          00FFFF63 95 TF3CR    .EQU    H'FFFF63    ;Timer function control
                                           ;register

96          96 ;
97          00FFFF82 97 TCR3     .EQU    H'FFFF82    ;Timer control register3
98          00FFFF86 98 TCNT3    .EQU    H'FFFF86    ;Timer counter3
99          00FFFF88 99 GRA3     .EQU    H'FFFF88    ;General register A3
100         00FFFF8A 100 GRB3    .EQU    H'FFFF8A    ;General register B3
101         101 ;
102         00FFFF98 102 GRA4    .EQU    H'FFFF98    ;General register A4
103         00FFFF9A 103 GRB4    .EQU    H'FFFF9A    ;General register B4
104         104 ;
105         105 ;*****
106         106 ;*
107         107 ;*      MAIN PROGRAM : NRPWM3MN
108         108 ;*
109         109 ;*****
110         110 ;
111         010000    111          .SECTION  PROG, CODE, LOCATE=H'010000
112         112 ;
113         00010000 113 NRPWM3MN .EQU    $
114         010000 7A0700FFFEFE 114 MOV.L   #H'FFFEFE, SP ;Initialize stack pointer
115         010006 F8E0          115 MOV.B   #B'11100000, R0L
116         010008 3860          116 MOV.B   R0L, @TSTR   ;Set TCNT3 stop data
117         01000A F8A0          117 MOV.B   #B'10100000, R0L
118         01000C 3882          118 MOV.B   R0L, @TCR3  ;Initialize TCR3
119         119 ;
120         01000E F8F0          120 MOV.B   #B'11110000, R0L
121         010010 3863          121 MOV.B   R0L, @TF3CR ;Set reset-synchronized mode
122         010012 79000000      122 MOV.W   #H'0000, R0
123         010016 6B80FF86      123 MOV.W   R0, @TCNT3  ;Initialize TCNT3
124         124 ;
125         01001A 6B00FF00      125 MOV.W   @PUL_CYC, R0
126         01001E 6B80FF88      126 MOV.W   R0, @GRA3   ;Set pulse cycle time
127         010022 6B00FF02      127 MOV.W   @PUL_TIMA4, R0
128         010026 6B80FF98      128 MOV.W   R0, @GRA4   ;Set pulse change timing
129         01002A 6B00FF04      129 MOV.W   @PUL_TIMB3, R0
130         01002E 6B80FF8A      130 MOV.W   R0, @GRB3   ;Set pulse change timing
131         010032 6B00FF06      131 MOV.W   @PUL_TIMB4, R0
132         010036 6B80FF9A      132 MOV.W   R0, @GRB4   ;Set pulse change timing
133         133 ;
134         01003A F8E8          134 MOV.B   #B'11101000, R0L
135         01003C 3860          135 MOV.B   R0L, @TSTR   ;Start TCNT3
136         136 ;
137         01003E 40FE          137 NRPWM3MN99 BRA    NRPWM3MN99
138         138 ;

```

2. 基礎編

```
139                                139                .END
*****TOTAL ERRORS                0
*****TOTAL WARNINGS              0
```

2.7 PWM5 相出力

MCU : H8/3003

使用機能 : ITU (同期 / PWM モード)

2.7.1 仕様

図 2.32 に示すように、パルスの High 幅を変化させ、デューティ変化できるパルス (デューティパルス) を 5 相出力します。

デューティは ch0 から ch4 まで任意に設定できます。

$$\text{デューティ} = \frac{\text{パルスの High 幅}}{\text{パルスの周期}} \times 100 (\%)$$

16MHz で動作時、出力するパルスの周期は 125ns から 4.09ms の間で任意に設定できます。

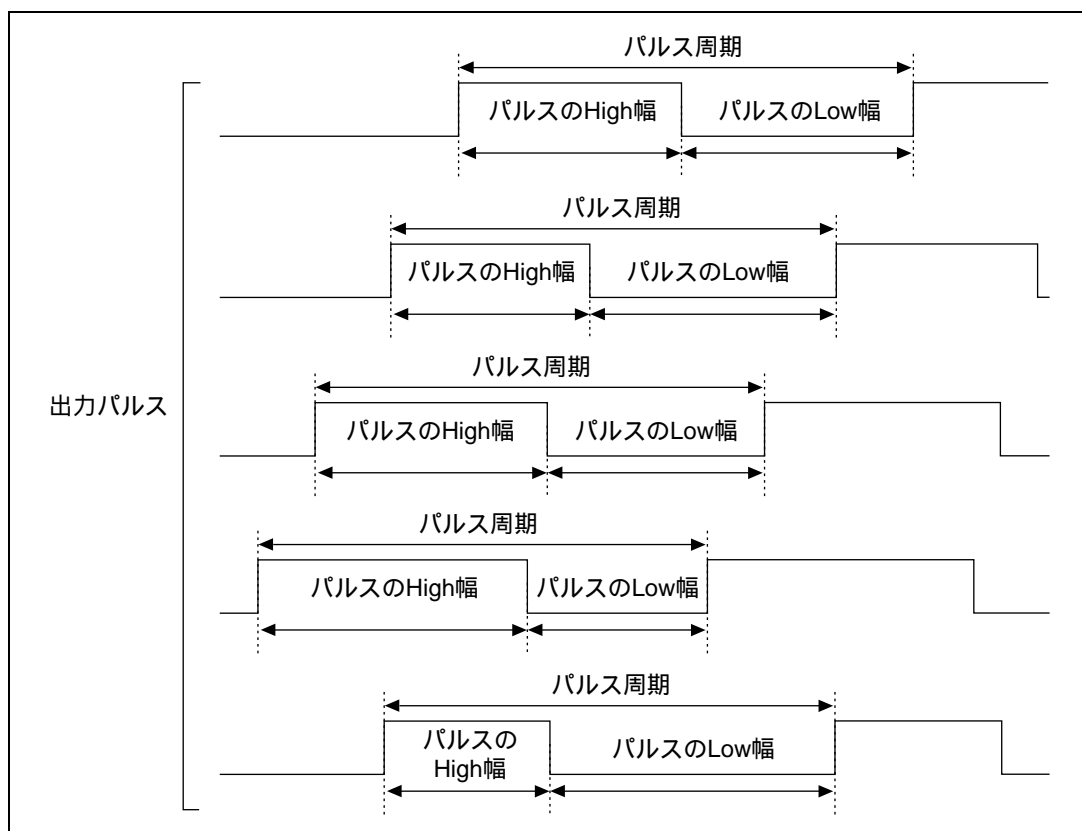


図 2.32 PWM 5 相出力例

2.7.2 使用機能説明

本タスク例では、ITU の ch0 ~ 4 を使用し、タイマカウンタを同期動作に設定し 5 相の PWM 波形を出力します。

図 2.33 に本タスク例で使用する ITU のブロック図を示します。

本タスク例では、ITU の以下の機能を使用しています。

- コンペアマッチ時、複数のタイマカウンタを同時にクリアする機能。（同期クリア）
- パルス出力端子が自動的に PWM 端子となる機能。（PWM モード）

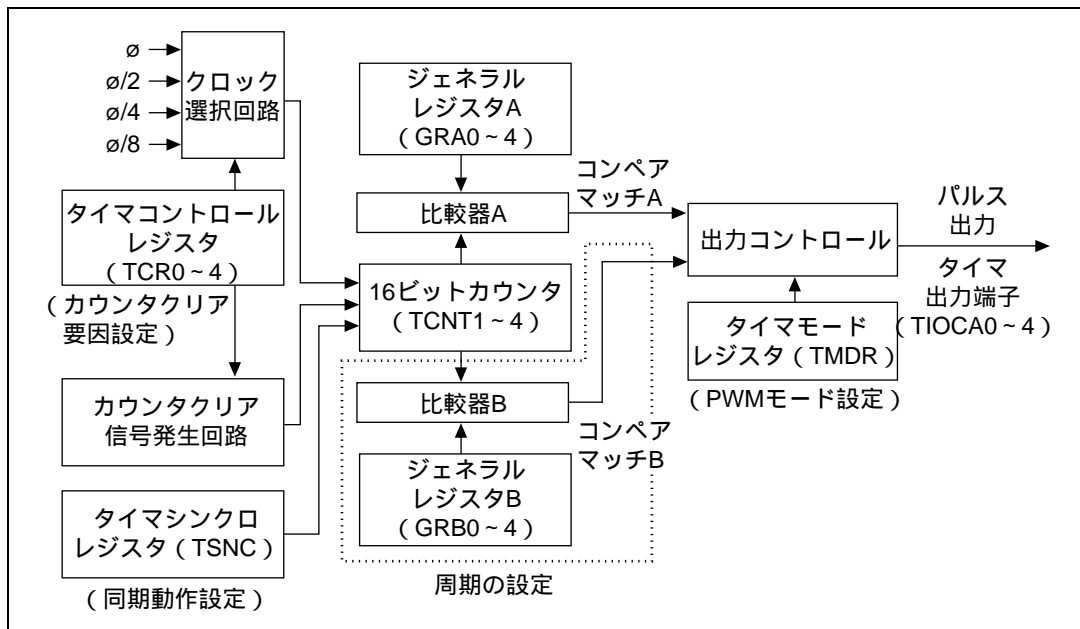


図 2.33 同期パルス出力ブロック図

表 2.32 に本タスク例の機能割り付けを示します。表 2.32 に示すように ITU の機能を割り付け、5 相の PWM 波形を出力します。

表 2.32 ITU 機能割り付け

ITU 機能	機能	
TSNC	ch0 から ch4 までを同期動作に設定する。	
TMDR	ch0 から ch4 までを PWM モードに設定する。	
ch0 ~ 4	TCR0 ~ 4	カウンタクロックおよびカウンタクリア要因を設定する。
	TIOCA0 ~ 4	パルスを出力する。
	GRA0 ~ 4	パルスのセットタイミングを設定する。
	GRB0 ~ 4	パルスの周期を設定する。

2.7.3 動作説明

図 2.34 に ch0 の動作原理を示します。図 2.34 に示すように H8/3003 のハードウェア処理およびソフトウェアの処理によりデューティパルスを出力します。

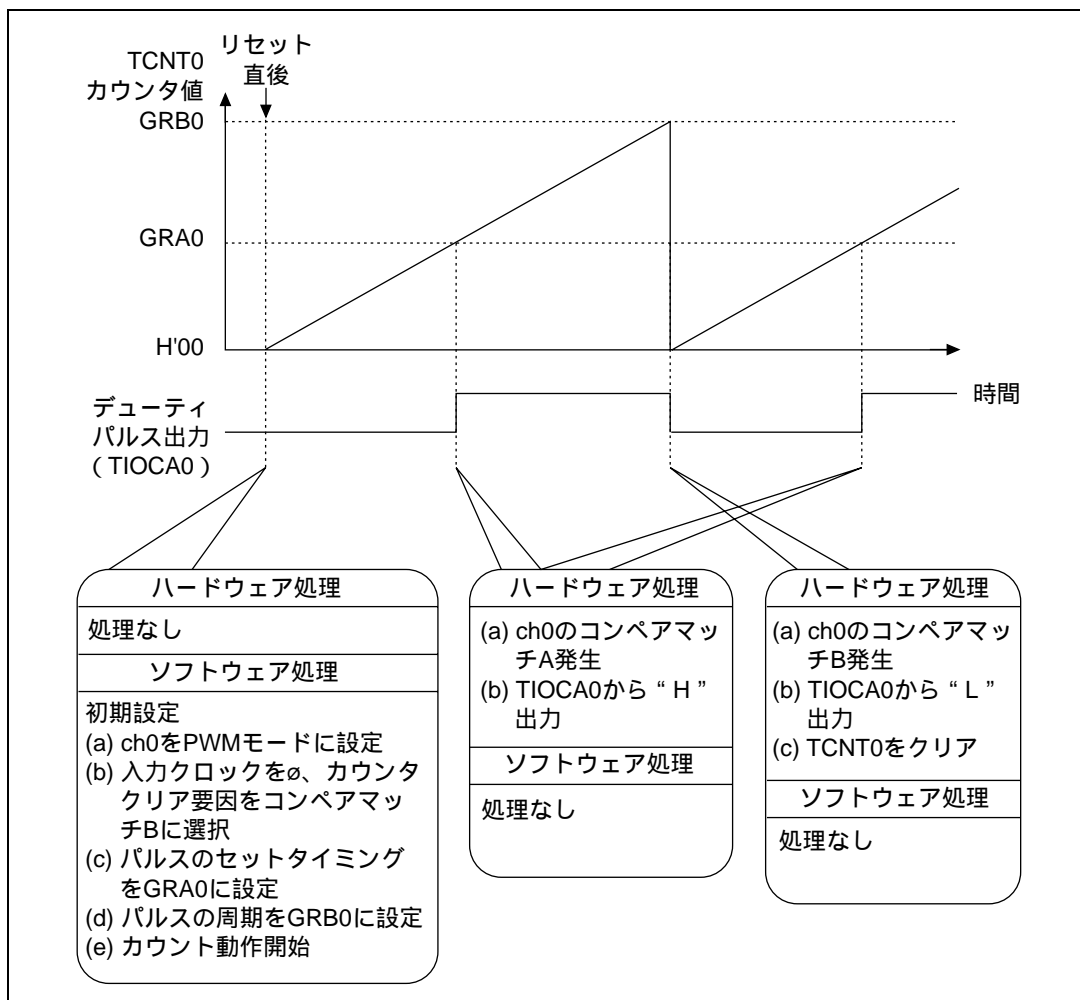


図 2.34 PWM 出力動作原理

図 2.35 に ch0 ~ ch4 までの動作原理を示します。図 2.35 に示すように H8/3003 のハードウェア処理およびソフトウェアの処理により ch0 に同期したデューティパルスを出力します。

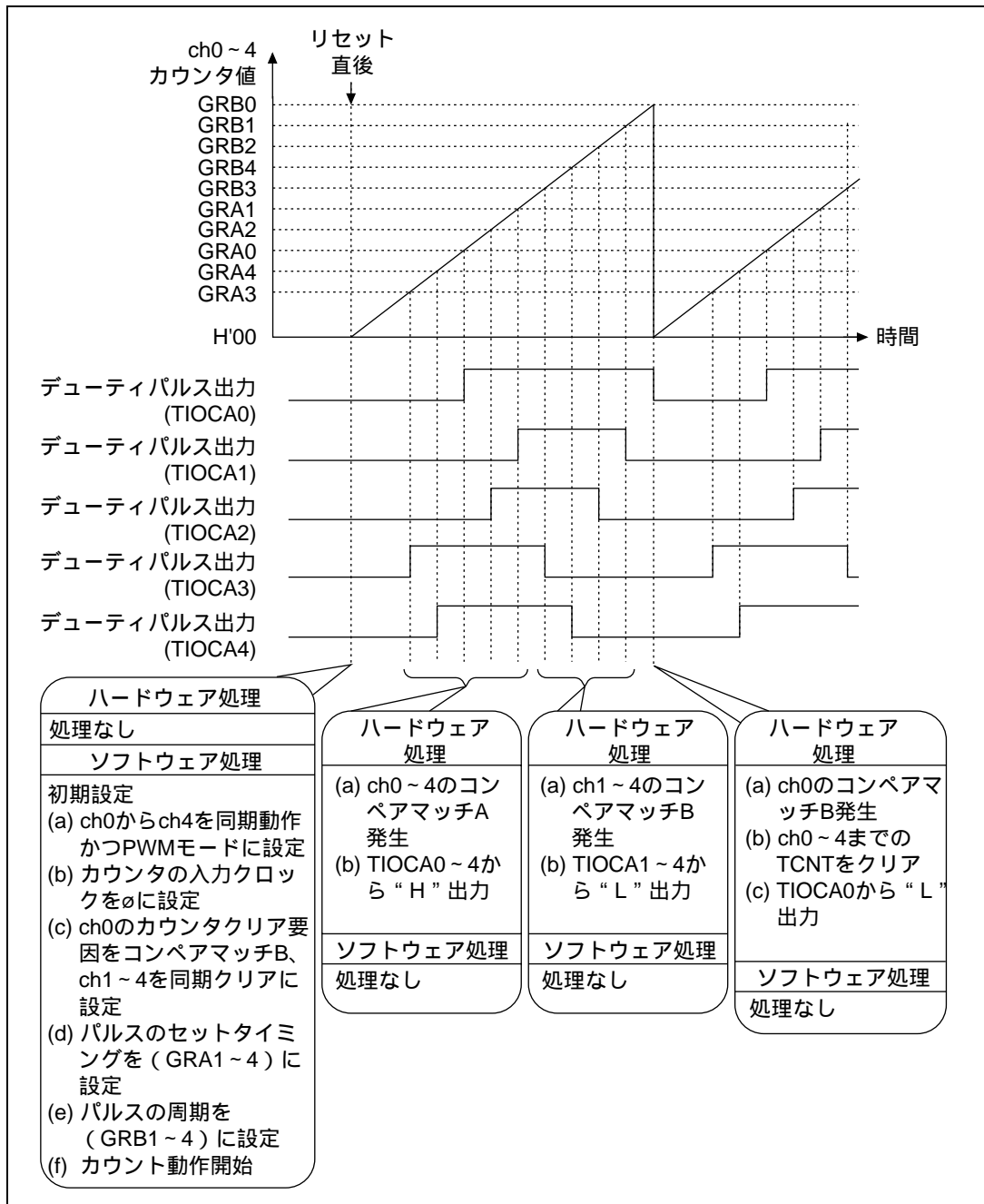


図 2.35 PWM 5 相出力動作原理

2.7.4 ソフトウェア説明

表 2.33 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	PWM5MN	モード設定、出力パルスの周期およびパルス幅の設定、RAM の初期設定を行い、5 相の PWM パルスを出力する。

表 2.34 引数の説明

ラベル名、レジスタ名	機能	データ長	使用モジュール名	入出力
PUL_CYC0~4	PWM パルスの周期に相当するタイマ値を設定する。 周期は以下の式にて求まる。 周期 (ns) = タイマ値 × 周期 (16MHz 動作時 62.5ns)	1 ワード	メインルーチン	入力
PUL_SET0~4	PWM パルスのセットタイミングに相当するタイマ値を設定する。 セットタイミングは以下の式にて求まる。 セットタイミング (ns) = タイマ値 × 周期 (16MHz 動作時 62.5ns)	1 ワード	メインルーチン	入力

表 2.35 使用内部レジスタ説明

レジスタ名	機能	使用モジュール名	
TSTR	タイマカウンタの動作 / 禁止を設定する。	メインルーチン	
TSNC	ch0~4 を同期動作に設定する。	メインルーチン	
TMDR	ch0~4 を PWM モードに設定する。	メインルーチン	
ch0	TCR0	カウンタクリア要因をコンペアマッチ B に設定する。	メインルーチン
	GRA0	パルスのセットタイミングを設定する。	メインルーチン
	GRB0	パルスの周期を設定する。	メインルーチン
ch1~4	TCR1~4	カウンタクリア要因を同期クリアに設定する。	メインルーチン
	GRA1~4	パルスのセットタイミングを設定する。	メインルーチン
	GRB1~4	パルスの周期を設定する。	メインルーチン

表 2.36 使用汎用レジスタ説明

使用モジュール名	レジスタ名	機能
メインルーチン	R0	データ設定時のワークとして使用する。

- 使用 RAM 説明
本タスク例では引数以外の RAM は使用していません。

2.7.5 フローチャート

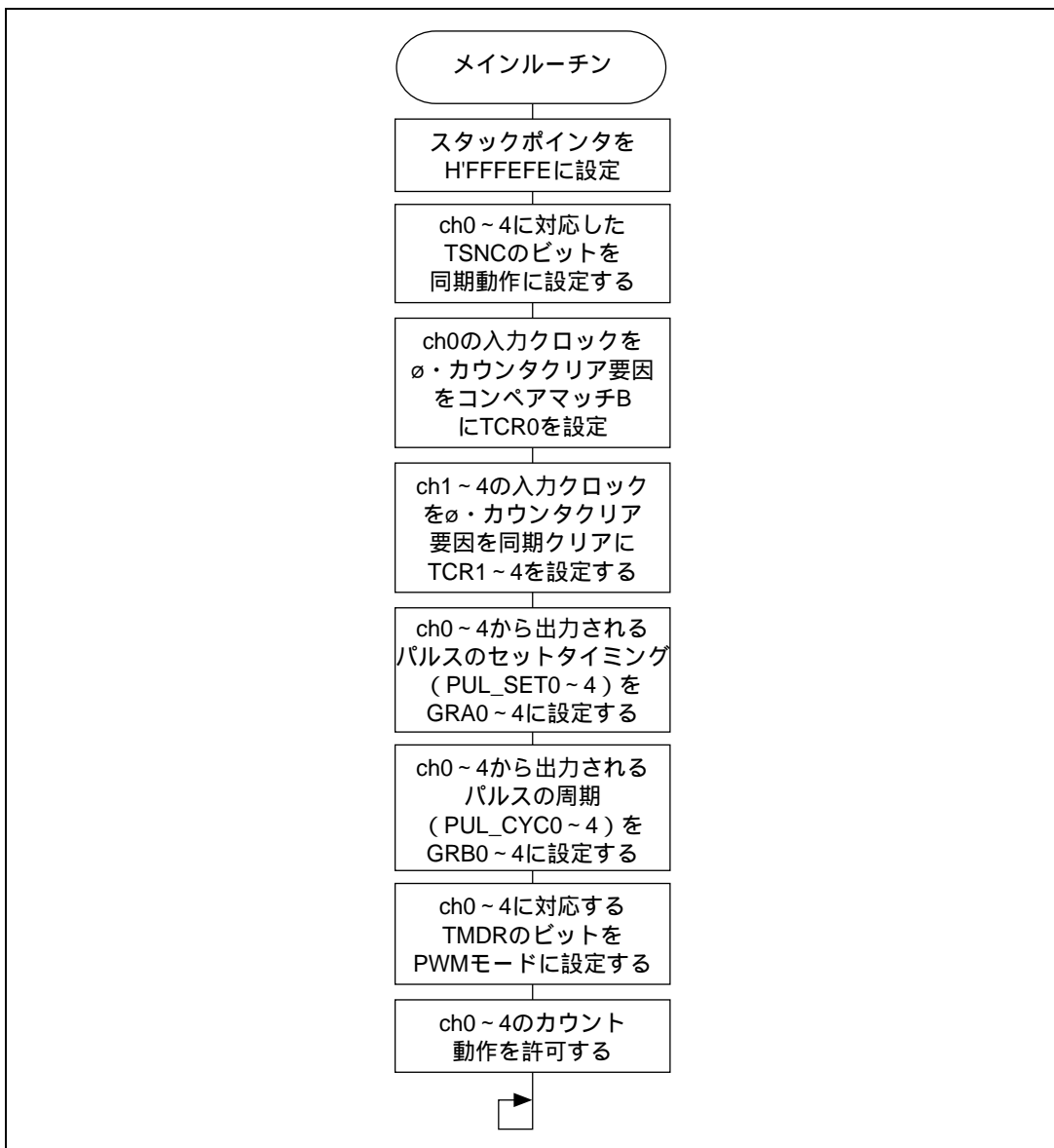


図 2.36 メインルーチン

2.7.6 プログラムリスト

```

1          1 ;*****
2          2 ;*
3          3 ;*          VECTOR ADDRESS
4          4 ;*
5          5 ;*****
6          6 ;
7          7          .CPU      300HA
8 000000   8 ;          .SECTION VECT, CODE, LOCATE=H'000000
9          9 ;
10 000000 00010000 10 RES          .DATA.L  PWM5MN
11          11 ;
12 00001C   12          .ORG      H'00001C
13 00001C 00010000 13 NMI          .DATA.L  PWM5MN
14 000020 00010000 14 TRAPA0       .DATA.L  PWM5MN
15 000024 00010000 15 TRAPA1       .DATA.L  PWM5MN
16 000028 00010000 16 TRAPA2       .DATA.L  PWM5MN
17 00002C 00010000 17 TRAPA3       .DATA.L  PWM5MN
18          18 ;
19 00030    19          .ORG      H'000030
20 000030 00010000 20 IRQ0         .DATA.L  PWM5MN
21 000034 00010000 21 IRQ1         .DATA.L  PWM5MN
22 000038 00010000 22 IRQ2         .DATA.L  PWM5MN
23 00003C 00010000 23 IRQ3         .DATA.L  PWM5MN
24 000040 00010000 24 IRQ4         .DATA.L  PWM5MN
25 000044 00010000 25 IRQ5         .DATA.L  PWM5MN
26 000048 00010000 26 IRQ6         .DATA.L  PWM5MN
27 00004C 00010000 27 IRQ7         .DATA.L  PWM5MN
28 000050 00010000 28 WOVI0        .DATA.L  PWM5MN
29 000054 00010000 29 CMI          .DATA.L  PWM5MN
30          30 ;
31 000060    31          .ORG      H'000060
32 000060 00010000 32 IMIA0        .DATA.L  PWM5MN
33 000064 00010000 33 IMIB0        .DATA.L  PWM5MN
34 000068 00010000 34 OVI0         .DATA.L  PWM5MN
35          35 ;
36 000070    36          .ORG      H'000070
37 000070 00010000 37 IMIA1        .DATA.L  PWM5MN
38 000074 00010000 38 IMIB1        .DATA.L  PWM5MN
39 000078 00010000 39 OVI1         .DATA.L  PWM5MN
40          40 ;
41 000080    41          .ORG      H'000080
42 000080 00010000 42 IMIA2        .DATA.L  PWM5MN
43 000084 00010000 43 IMIB2        .DATA.L  PWM5MN
44 000088 00010000 44 OVI2         .DATA.L  PWM5MN
45          45 ;

```

2. 基礎編

```

46 000090          46          .ORG      H'000090
47 000090 00010000 47 IMIA3    .DATA.L  PWM5MN
48 000094 00010000 48 IMIB3    .DATA.L  PWM5MN
49 000098 00010000 49 OVI3     .DATA.L  PWM5MN
50                50 ;
51 0000A0          51          .ORG      H'0000A0
52 0000A0 00010000 52 IMIA4    .DATA.L  PWM5MN
53 0000A4 00010000 53 IMIB4    .DATA.L  PWM5MN
54 0000A8 00010000 54 OVI4     .DATA.L  PWM5MN
55                55 ;
56 0000B0          56          .ORG      H'0000B0
57 0000B0 00010000 57 DEND0A   .DATA.L  PWM5MN
58 0000B4 00010000 58 DEND0B   .DATA.L  PWM5MN
59 0000B8 00010000 59 DEND1A   .DATA.L  PWM5MN
60 0000BC 00010000 60 DEND1B   .DATA.L  PWM5MN
61 0000C0 00010000 61 DEND2A   .DATA.L  PWM5MN
62 0000C4 00010000 62 DEND2B   .DATA.L  PWM5MN
63 0000C8 00010000 63 DEND3A   .DATA.L  PWM5MN
64 0000CC 00010000 64 DEND3B   .DATA.L  PWM5MN
65 0000D0 00010000 65 ERI0     .DATA.L  PWM5MN
66 0000D4 00010000 66 RXI0     .DATA.L  PWM5MN
67 0000D5 00010000 67 TXI0     .DATA.L  PWM5MN
68 0000DC 00010000 68 TEI0     .DATA.L  PWM5MN
69 0000E0 00010000 69 ERI1     .DATA.L  PWM5MN
70 0000E4 00010000 70 RXI1     .DATA.L  PWM5MN
71 0000E8 00010000 71 TXI1     .DATA.L  PWM5MN
72 0000EC 00010000 72 TEI1     .DATA.L  PWM5MN
73 0000F0 00010000 73 ADI      .DATA.L  PWM5MN
74                74 ;
75                75 ;*****
76                76 ;*                               *
77                77 ;*          RAM ALLOCATION          *
78                78 ;*                               *
79                79 ;*****
80                80 ;
81 FFFFE0          81          .SECTION  RAM,DATA,LOCATE=H'FFFEF0
82                82 ;
83 FFFFE0 00000002 83 PUL_CYC0 .RES.W  1          ;Pulse cycle time (ch0)
84 FFFFE2 00000002 84 PUL_SET0 .RES.W  1          ;Pulse cycle timing (ch0)
85                85 ;
86 FFFFE4 00000002 86 PUL_CYC1 .RES.W  1          ;Pulse cycle time (ch1)
87 FFFFE6 00000002 87 PUL_SET1 .RES.W  1          ;Pulse set timing (ch1)
88                88 ;
89 FFFFE8 00000002 89 PUL_CYC2 .RES.W  1          ;Pulse cycle time (ch2)
90 FFFFEA 00000002 90 PUL_SET2 .RES.W  1          ;Pulse set timing (ch2)
91                91 ;
92 FFFFE4 00000002 92 PUL_CYC3 .RES.W  1          ;Pulse cycle time (ch3)

```

```

93 FFFEFE 00000002 93 PUL_SET3 .RES.W 1 ;Pulse set timing (ch3)
94 ;
95 FFFF00 00000002 95 PUL_CYC4 .RES.W 1 ;Pulse cycle time (ch4)
96 FFFF02 00000002 96 PUL_SET4 .RES.W 1 ;Pulse set timing (ch4)
97 ;
98 ;*****
99 ;* *
100 ;* SYMBOL DEFINITIONS *
101 ;* *
102 ;*****
103 ;
104 00FFFF60 104 TSTR .EQU H'FFFF60 ;Timer start register
105 00FFFF61 105 TSNC .EQU H'FFFF61 ;Timer syncro register
106 00FFFF62 106 TMDR .EQU H'FFFF62 ;Timer mode register
107 ;
108 00FFFF64 108 TCR0 .EQU H'FFFF64 ;Timer control register0
109 00FFFF6A 109 GRA0 .EQU H'FFFF6A ;General register A0
110 00FFFF6C 110 GRB0 .EQU H'FFFF6C ;General register B0
111 ;
112 00FFFF6E 112 TCR1 .EQU H'FFFF6E ;Timer control register1
113 00FFFF74 113 GRA1 .EQU H'FFFF74 ;General register A1
114 00FFFF76 114 GRB1 .EQU H'FFFF76 ;General register B1
115 ;
116 00FFFF78 116 TCR2 .EQU H'FFFF78 ;Timer control register2
117 00FFFF7E 117 GRA2 .EQU H'FFFF7E ;General register A2
118 00FFFF80 118 GRB2 .EQU H'FFFF80 ;General register B2
119 ;
120 00FFFF82 120 TCR3 .EQU H'FFFF82 ;Timer control register3
121 00FFFF88 121 GRA3 .EQU H'FFFF88 ;General register A3
122 00FFFF8A 122 GRB3 .EQU H'FFFF8A ;General register B3
123 ;
124 00FFFF92 124 TCR4 .EQU H'FFFF92 ;Timer control register4
125 00FFFF98 125 GRA4 .EQU H'FFFF98 ;General register A4
126 00FFFF9A 126 GRB4 .EQU H'FFFF9A ;General register B4
127 ;
128 ;*****
129 ;* *
130 ;* MAIN PROGRAM : PWM5MN *
131 ;* *
132 ;*****
133 ;
134 010000 134 .SECTION PROG,CODE,LOCATE=H'010000
135 ;
136 00010000 136 PWM5MN: .EQU $
137 010000 7A0700FFFE 137 MOV.L #H'FFFE,SP ;Initialize stack pointer
138 010006 F8FF 138 MOV.B #B'11111111,R0L

```

2. 基礎編

```
139 010008 3861          139          MOV.B    R0L,@TSNC    ;Set synchronization mode
                                   ;ch0~4
140
141 01000A F8C0          141          MOV.B    #B'11000000,R0L
142 01000C 3864          142          MOV.B    R0L,@TCR0    ;Initialize TCR0
143 01000E F8E0          143          MOV.B    #B'11100000,R0L
144 010010 386E          144          MOV.B    R0L,@TCR1    ;Initialize TCR1~4
145 010012 3878          145          MOV.B    R0L,@TCR2
146 010014 3882          146          MOV.B    R0L,@TCR3
147 010016 3892          147          MOV.B    R0L,@TCR4
148
148          ;
149 010018 6B00FEF2      149          MOV.W    @PUL_SET0,R0
150 01001C 6B80FF6A      150          MOV.W    R0,@GRA0    ;Set pulse set timing (ch0~4)
151 010020 6B00FEF6      151          MOV.W    @PUL_SET1,R0
152 010024 6B80FF74      152          MOV.W    R0,@GRA1
153 010028 6B00FEFA      153          MOV.W    @PUL_SET2,R0
154 01002C 6B80FF7E      154          MOV.W    R0,@GRA2
155 010030 6B00FEFE      155          MOV.W    @PUL_SET3,R0
156 010034 6B80FF88      156          MOV.W    R0,@GRA3
157 010038 6B00FF02      157          MOV.W    @PUL_SET4,R0
158 01003C 6B80FF98      158          MOV.W    R0,@GRA4
159
160 010040 6B00FEF0      160          MOV.W    @PUL_CYC0,R0
161 010044 6B80FF6C      161          MOV.W    R0,@GRB0    ;Set pulse cycle time (ch0~4)
162 010048 6B00FEF4      162          MOV.W    @PUL_CYC1,R0
163 01004C 6B80FF76      163          MOV.W    R0,@GRB1
164 010050 6B00FEF8      164          MOV.W    @PUL_CYC2,R0
165 010054 6B80FF80      165          MOV.W    R0@GRB2
166 010058 6B00FEFC      166          MOV.W    @PUL_CYC3,R0
167 01005C 6B80FF8A      167          MOV.W    R0,@GRB3
168 010060 6B00FF00      168          MOV.W    @PUL_CYC4,R0
169 010064 6B80FF9A      169          MOV.W    R0,@GRB4
170
171 010068 F89F          171          MOV.B    #B'10011111,R0L
172 01006A 3862          172          MOV.B    R0L,@TMDR    ;Set PWM mode (ch0~4)
173
174 01006C F8FF          174          MOV.B    #B'11111111,R0L
175 01006E 3860          175          MOV.B    R0L,@TSTR    ;Set TCNT start data (ch0~4)
176
177 010070 40FE          177  PWM5MN99  BRA     PWM5MN99
178
179
. END
*****TOTAL ERRORS      0
*****TOTAL WARNINGS    0
```


2.8 外部信号を使用したタイマカウンタ同時リセット

MCU : H8/3003

使用機能 : ITU (同期 / PWM モード)

2.8.1 仕様

図 2.37 に示すように外部信号の立ち下がりに同期して 4 チャンネルのパルスを出力します。外部信号の立ち下がりからの遅延時間およびパルス幅は以下に示す範囲で可変できます。

250ns 遅延時間 < 外部信号の周期 パルス幅

62.5ns パルス幅 < 外部信号の周期 遅延時間

16MHz 動作時、外部信号のパルス周期は 320ns ~ 4.09ms の間で任意に設定できます。

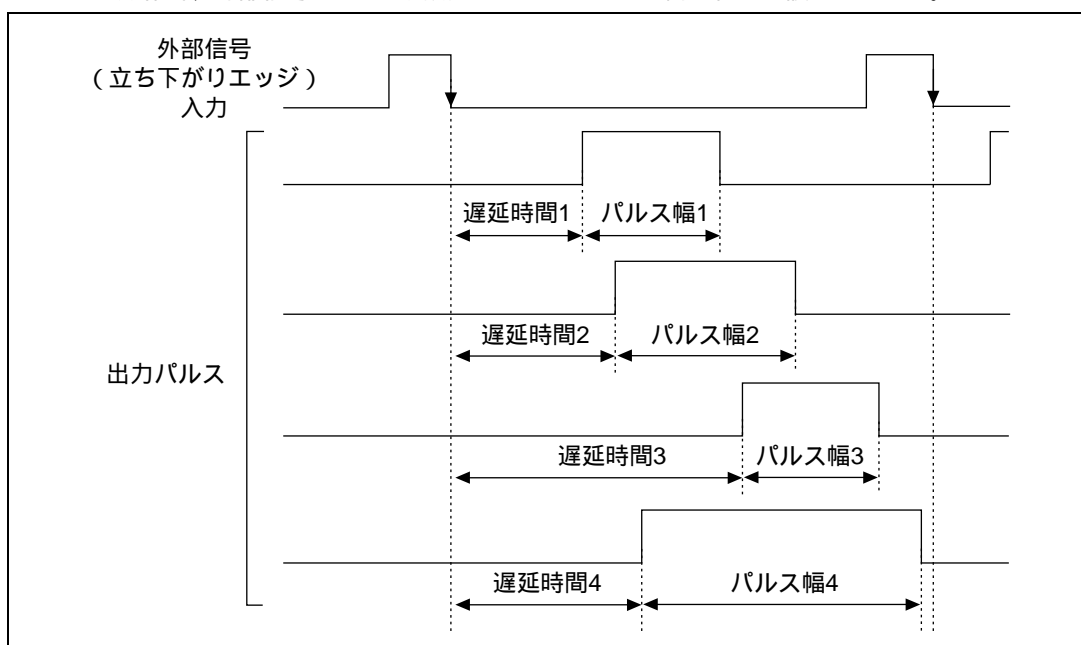


図 2.37 同期パルス出力例

2.8.2 使用機能説明

本タスク例では、外部信号を使用した複数のタイマカウンタ同時リセットを行い、4相のパルスを出力します。

図 2.38 に本タスク例で使用する ITU/ch0 のブロック図を示します。ch0 では、以下の機能を使用し外部信号のエッジを検出します。

- パルスの立ち下がりエッジを検出する機能。(インプットキャプチャ)
- パルスの立ち下がりエッジ検出時タイマカウンタをクリアする機能。

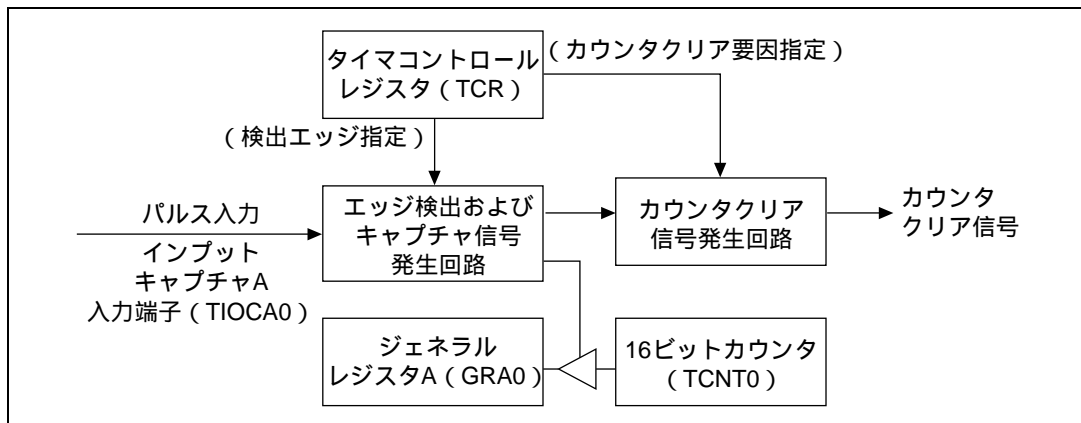


図 2.38 ITU/ch0 ブロック図

図 2.39 に本タスク例で使用する ITU/ch1 ~ 4 のブロック図を示します。ch1 ~ 4 では、以下の機能を使用し ch0 に同期したカウンタクリアを行い、4 相 PWM 波形を出力します。

- 複数のタイマカウンタを同時クリアする機能。(同期動作)
- パルス出力端子が自動的に PWM 端子となる機能。(PWM モード)

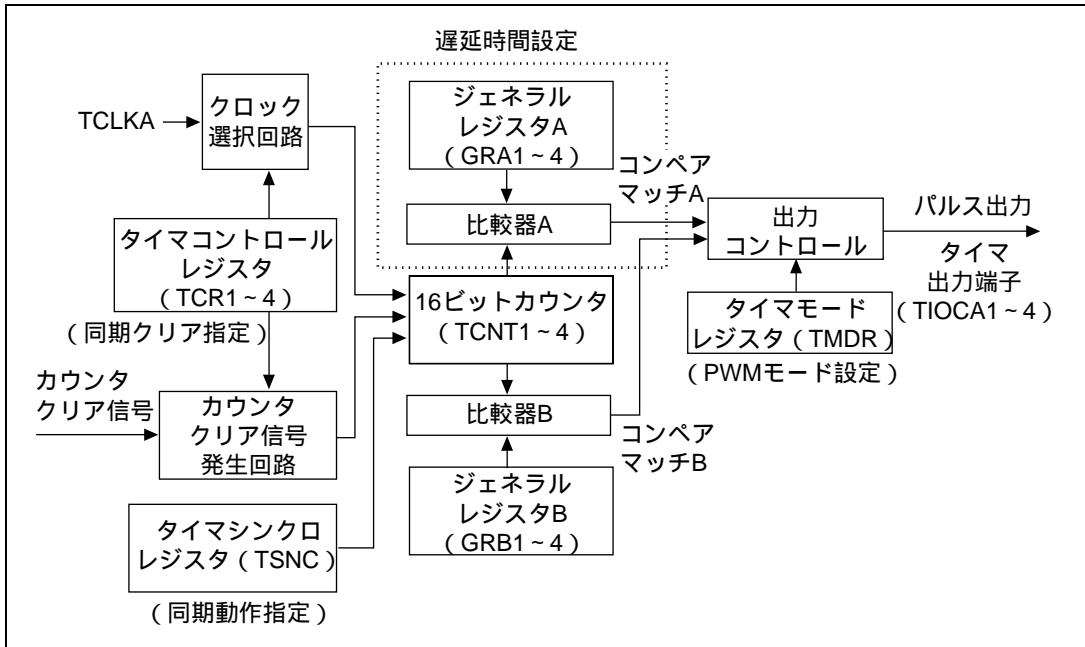


図 2.39 同期パルス出力ブロック図

表 2.37 に本タスク例の機能割り付けを示します。表 2.37 に示すように ITU の機能を割り付け、ch0 に立ち下がりエッジが入力される度に各チャンネルごとに遅延時間を持ったパルスを出力します。

表 2.37 ITU 機能割り付け

ITU 機能	機能	
TSNC	ch0 から ch4 までを同期動作に設定する。	
TMDR	ch1 から ch4 までを PWM モードに設定する。	
TCLKA	ch1 から ch4 のカウンタ外部クロック A 端子。	
ch0	TCR0	カウンタクリア要因を設定する。
	TIOCA0	出力パルスの基準となる外部信号を入力する。
	GRA0	外部信号の立ち下がりにより、カウンタ値を検出する。
ch1 ~ 4	TCR1 ~ 4	カウンタクロックおよびカウンタクリア要因を設定する。
	TIOCA1 ~ 4	パルスを出力する。
	GRA1 ~ 4	遅延時間を設定する。
	GRB1 ~ 4	遅延時間 + パルス幅に相当するタイマ値を設定する。

2.8.3 動作説明

- 1 チャンネルパルス出力の場合
 図2.40に動作原理を示します。図2.40に示すようにH8/3003のハードウェア処理およびソフトウェアの処理によりch1からPWMパルスを出力します。

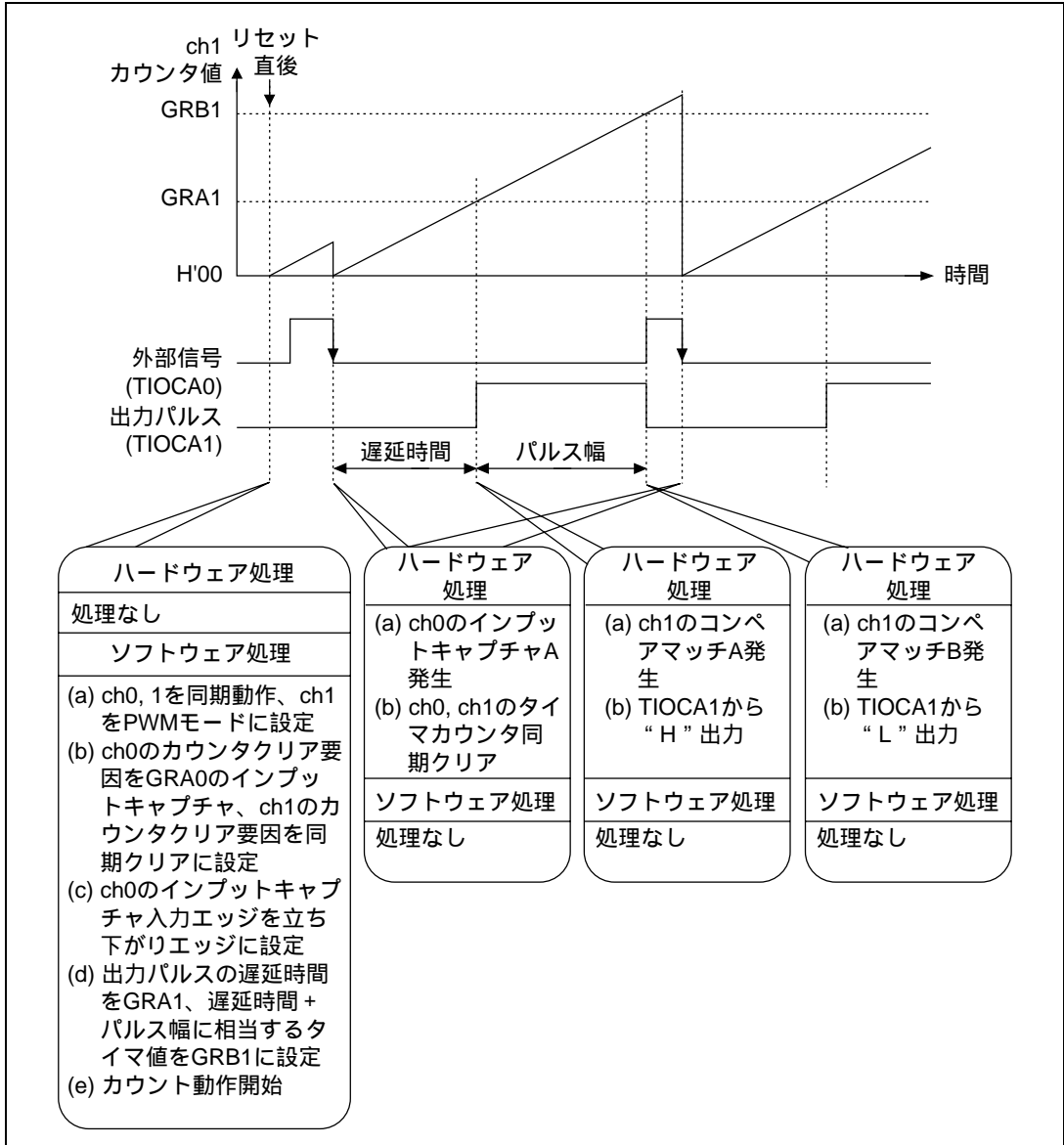


図 2.40 パルス出力動作原理

- 4 チャンネルパルス出力の場合
 図2.41に4チャンネルの出力パルス動作原理を示します。図2.41に示すようにH8/3003のハードウェア処理およびソフトウェアの処理によりch0に同期したデューティパルスを出力します。

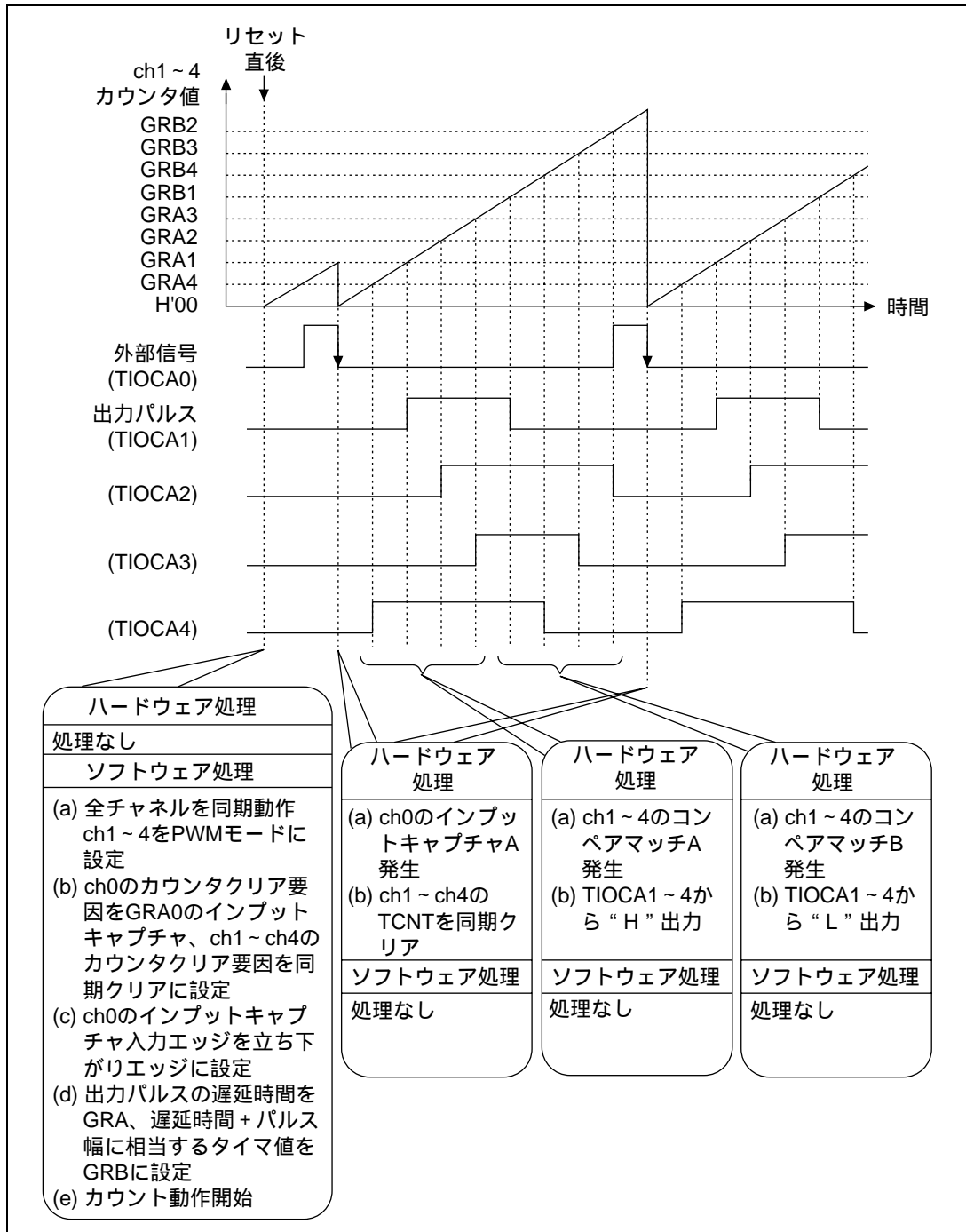


図 2.41 4チャンネルパルス出力動作原理

2. 基礎編

2.8.4 ソフトウェア説明

表 2.38 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	CNTRSMN	遅延時間およびパルス幅を GRA1～4、GRB1～4 に設定し、ch0 に同期した 4 相の PWM パルスを出力する。

表 2.39 引数の説明

ラベル名、レジスタ名	機能	データ長	使用モジュール名	入出力
SET_DLY1～4	出力パルスの遅延時間に相当するタイマ値を設定する。 遅延時間は以下の式にて求まる。 遅延時間 (ns) = タイマ値 × 周期 (16MHz 動作時 62.5ns)	1 ワード	メインルーチン	出力
SET_WID1～4	出力パルスの遅延時間 + パルス幅に相当するタイマ値を設定する。 パルス幅は以下の式にて求まる。 パルス幅 (ns) = タイマ値 × 周期 (16MHz 動作時 62.5ns)	1 ワード	メインルーチン	出力

表 2.40 使用内部レジスタ説明

レジスタ名	機能	使用モジュール名
TSTR	タイマカウンタの動作 / 禁止を設定する。	メインルーチン
TSNC	タイマカウンタ同期動作の許可 / 禁止を設定する。	
TMDR	PWM モードの許可 / 禁止を設定する。	
ch0	TCR	カウンタクリア要因をインプットキャプチャ A に設定する。
	TIOR	外部信号の立ち下がり検出時、TCNT から GRA への転送を行い、同時にカウンタをクリアするように設定する。
ch1～4	TCR	TCNT に入力するクロックの選択およびカウンタクリア要因を同期クリアに設定する。
	GRA	出力パルスの遅延時間を設定する。
	GRB	出力パルスの遅延時間 + パルス幅に相当するタイマ値を設定する。

表 2.41 使用汎用レジスタ説明

使用モジュール名	レジスタ名	機能
メインルーチン	R0	データ設定時のワークとして使用する。

- 使用 RAM 説明
本タスク例では引数以外の RAM は使用していません。

2.8.5 フローチャート



図 2.42 メインルーチン

2.8.6 プログラムリスト

```

1          1  ;*****
2          2  ;*
3          3  ;*          VECTOR ADDRESS
4          4  ;*
5          5  ;*****
6          6  ;
7 000000   7          .SECTION VECT, CODE, LOCATE=H'000000
8          8  ;
9 000000 00010000  9 RES          .DATA.L  CNTRSMN
10         10 ;
11 00001C   11         .ORG          H'00001C
12 00001C 00010000  12 NMI          .DATA.L  CNTRSMN
13 000020 00010000  13 TRAPA0       .DATA.L  CNTRSMN
14 000024 00010000  14 TRAPA1       .DATA.L  CNTRSMN
15 000028 00010000  15 TRAPA2       .DATA.L  CNTRSMN
16 00002C 00010000  16 TRAPA3       .DATA.L  CNTRSMN
17         17 ;
18 000030   18         .ORG          H'000030
19 000030 00010000  19 IRQ0         .DATA.L  CNTRSMN
20 000034 00010000  20 IRQ1         .DATA.L  CNTRSMN
21 000038 00010000  21 IRQ2         .DATA.L  CNTRSMN
22 00003C 00010000  22 IRQ3         .DATA.L  CNTRSMN
23 000040 00010000  23 IRQ4         .DATA.L  CNTRSMN
24 000044 00010000  24 IRQ5         .DATA.L  CNTRSMN
25 000048 00010000  25 IRQ6         .DATA.L  CNTRSMN
26 00004C 00010000  26 IRQ7         .DATA.L  CNTRSMN
27 000050 00010000  27 WOVI0        .DATA.L  CNTRSMN
28 000054 00010000  28 CMI          .DATA.L  CNTRSMN
29         29 ;
30 000060   30         .ORG          H'000060
31 000060 00010000  31 IMIA0        .DATA.L  CNTRSMN
32 000064 00010000  32 IMIB0        .DATA.L  CNTRSMN
33 000068 00010000  33 OVI0         .DATA.L  CNTRSMN
34         34 ;
35 000070   35         .ORG          H'000070
36 000070 00010000  36 IMIA1        .DATA.L  CNTRSMN
37 000074 00010000  37 IMIB1        .DATA.L  CNTRSMN
38 000078 00010000  38 OVI1         .DATA.L  CNTRSMN
39         39 ;
40 000080   40         .ORG          H'000080
41 000080 00010000  41 IMIA2        .DATA.L  CNTRSMN
42 000084 00010000  42 IMIB2        .DATA.L  CNTRSMN
43 000088 00010000  43 OVI2         .DATA.L  CNTRSMN
44         44 ;
45 000090   45         .ORG          H'000090

```



```

46 000090 00010000      46 IMIA3      .DATA.L  CNTRSMN
47 000094 00010000      47 IMIB3      .DATA.L  CNTRSMN
48 000098 00010000      48 OVI3       .DATA.L  CNTRSMN
49                               49 ;
50 0000A0                50                .ORG      H'0000A0
51 0000A0 00010000      51 IMIA4      .DATA.L  CNTRSMN
52 0000A4 00010000      52 IMIB4      .DATA.L  CNTRSMN
53 0000A8 00010000      53 OVI4       .DATA.L  CNTRSMN
54                               54 ;
55 0000B0                55                .ORG      H'0000B0
56 0000B0 00010000      56 DEND0A     .DATA.L  CNTRSMN
57 0000B4 00010000      57 DEND0B     .DATA.L  CNTRSMN
58 0000B8 00010000      58 DEND1A     .DATA.L  CNTRSMN
59 0000BC 00010000      59 DEND1B     .DATA.L  CNTRSMN
60 0000C0 00010000      60 DEND2A     .DATA.L  CNTRSMN
61 0000C4 00010000      61 DEND2B     .DATA.L  CNTRSMN
62 0000C8 00010000      62 DEND3A     .DATA.L  CNTRSMN
63 0000CC 00010000      63 DEND3B     .DATA.L  CNTRSMN
64 0000D0 00010000      64 ERI0       .DATA.L  CNTRSMN
65 0000D4 00010000      65 RXI0       .DATA.L  CNTRSMN
66 0000D8 00010000      66 TXI0       .DATA.L  CNTRSMN
67 0000DC 00010000      67 TEI0       .DATA.L  CNTRSMN
68 0000E0 00010000      68 ERI1       .DATA.L  CNTRSMN
69 0000E4 00010000      69 RXI1       .DATA.L  CNTRSMN
70 0000E8 00010000      70 TXI1       .DATA.L  CNTRSMN
71 0000EC 00010000      71 TEI1       .DATA.L  CNTRSMN
72 0000F0 00010000      72 ADI        .DATA.L  CNTRSMN
73                               73 ;
74                               74 ;*****
75                               75 ;*                                     *
76                               76 ;*          RAM ALLOCATION              *
77                               77 ;*                                     *
78                               78 ;*****
79                               79 ;
80 FFFF00                80                .SECTION  RAM,DATA,LOCATE=H'FFFF00
81                               81 ;
82 FFFF00 00000002      82 SET_DLY1   .RES.W   1          ;Delay time
83 FFFF02 00000002      83 SET_WID1   .RES.W   1          ;Pulse width time
84                               84 ;
85 FFFF04 00000002      85 SET_DLY2   .RES.W   1          ;Delay time
86 FFFF06 00000002      86 SET_WID2   .RES.W   1          ;Pulse width time
87                               87 ;
88 FFFF0B 00000002      88 SET_DLY3   .RES.W   1          ;Delay time
89 FFFF0A 00000002      89 SET_WID3   .RES.W   1          ;Pulse width time
90                               90 ;
91 FFFF0C 00000002      91 SET_DLY4   .RES.W   1          ;Delay time
92 FFFF0E 00000002      92 SET_WID4   .RES.W   1          ;Pulse width time

```

2. 基礎編

```

93          93 ;
94          94 ;*****
95          95 ;*                                     *
96          96 ;*          SYMBOL DEFINITIONS          *
97          97 ;*                                     *
98          98 ;*****
99          99 ;
100         00FFFF60 100 TSTR      .EQU    H'FFFF60      ;Timer start register
101         00FFFF61 101 TSNC      .EQU    H'FFFF61      ;Timer synchro register
102         00FFFF62 102 TMDR      .EQU    H'FFFF62      ;Timer mode register
103         103 ;
104         00FFFF64 104 TCR0      .EQU    H'FFFF64      ;Timer control register0
105         00FFFF65 105 TIOR0     .EQU    H'FFFF65      ;Timer I/O control register0
106         106 ;
107         00FFFF6E 107 TCR1      .EQU    H'FFFF6E      ;Timer control register1
108         00FFFF74 108 GRA1      .EQU    H'FFFF74      ;General register A1
109         00FFFF76 109 GRB1      .EQU    H'FFFF76      ;General register B1
110         110 ;
111         00FFFF78 111 TCR2      .EQU    H'FFFF78      ;Timer control register2
112         00FFFF7E 112 GRA2      .EQU    H'FFFF7E      ;General register A2
113         00FFFF80 113 GRB2      .EQU    H'FFFF80      ;General register B2
114         114 ;
115         00FFFF82 115 TCR3      .EQU    H'FFFF82      ;Timer control Register3
116         00FFFF88 116 GRA3      .EQU    H'FFFF88      ;General register A3
117         00FFFF8A 117 GRB3      .EQU    H'FFFF8A      ;General register B3
118         118 ;
119         00FFFF92 119 TCR4      .EQU    H'FFFF92      ;Timer control Register4
120         00FFFF98 120 GRA4      .EQU    H'FFFF98      ;General register A4
121         00FFFF9A 121 GRB4      .EQU    H'FFFF9A      ;General register B4
122         122 ;
123         123 ;*****
124         124 ;*                                     *
125         125 ;*          MAIN PROGRAM : CNTRSMN          *
126         126 ;*                                     *
127         127 ;*****
128         128 ;
129 010000    129          .SECTION  PROG,CODE,LOCATE=H'010000
130         130 ;
131         00010000 131 CNTRSMN:  .EQU    $
132 010000    7A0700FFFEFE 132      MOV.L    #H'FFFEFE,SP ;Initialize stack pointer
133 010006    F8FF          133      MOV.B    #B'11111111,R0L
134 010008    3861          134      MOV.B    R0L,@TSNC   ;Initialize TSNC
135         135 ;
136 01000A    F8A0          136      MOV.B    #B'10100000,R0L
137 01000C    3864          137      MOV.B    R0L,@TCR0   ;Initialize TCR0
138 01000E    F8E0          138      MOV.B    #B'11100000,R0L
139 010010    386E          139      MOV.B    R0L,@TCR1   ;Initialize TCR1~4

```

```
140 010012 3878      140      MOV.B    R0L,@TCR2
141 010014 3882      141      MOV.B    R0L,@TCR3
142 010016 3892      142      MOV.B    R0L,@TCR4
143                143      ;
144 010018 F88D      144      MOV.B    #B'10001101,R0L
145 01001A 3865      145      MOV.B    R0L,@TIOR0 ;Initialize TIOR0
146                146
147 01001C 6B00FF00  147      MOV.W    @SET_DLY1,R0
148 010020 6B80FF74  148      MOV.W    R0,@GRA1 ;Set delay time1~4
149 010024 6B00FF04  149      MOV.W    @SET_DLY2,R0
150 010028 6B80FF7E  150      MOV.W    R0,@GRA2
151 01002C 6B00FF06  151      MOV.W    @SET_DLY3,R0
152 010030 6B80FF88  152      MOV.W    R0,@GRA3
153 010034 6B00FF0C  153      MOV.W    @SET_DLY4,R0
154 010038 6B80FF98  154      MOV.W    R0,@GRA4
155                155
156 01003C 6B00FF02  156      MOV.W    @SET_WID1,R0
157 010040 6B80FF76  157      MOV.W    R0,@GRB1 ;Set width1~4
158 010044 6B00FF06  158      MOV.W    @SET_WID2,R0
159 010048 6B80FF80  159      MOV.W    R0,@GRB2
160 01004C 6B00FF0A  160      MOV.W    @SET_WID3,R0
161 010050 6B80FF8A  161      MOV.W    R0,@GRB3
162 010054 6B00FF0E  162      MOV.W    @SET_WID4,R0
163 010058 6B80FF9A  163      MOV.W    R0,@GRB4
164                164
165 01005C F89E      165      MOV.B    #B'10011110,R0L
166 01005E 3862      166      MOV.B    R0L,@TMDR ;Set PWM mode ch1~4
167 010060 F8FE      167      MOV.B    #B'11111110,R0L
168 010062 3860      168      MOV.B    R0L,@TSTR ;Start TCNT1~4
169                169
170 010064 40FE      170      CNTRS99  BRA    CNTRS99
171                171
172                172      .END
*****TOTAL ERRORS      0
*****TOTAL WARNINGS    0
```

2.9 バッファモードを使用したワンショットパルス出力

MCU : H8/3003

使用機能 : ITU (バッファ動作)

2.9.1 仕様

図 2.43 に示すように外部信号の立ち下がりに同期してワンショットパルスを出力します。外部信号の立ち下がりからの遅延時間およびパルス幅は以下に示す範囲で可変できます。

$2\mu\text{s}$ 遅延時間 < 基準パルスの周期 パルス幅

62.5 ns パルス幅 < 基準パルスの周期 遅延時間

基準パルスの周波数は、400kHz から入力可能です。

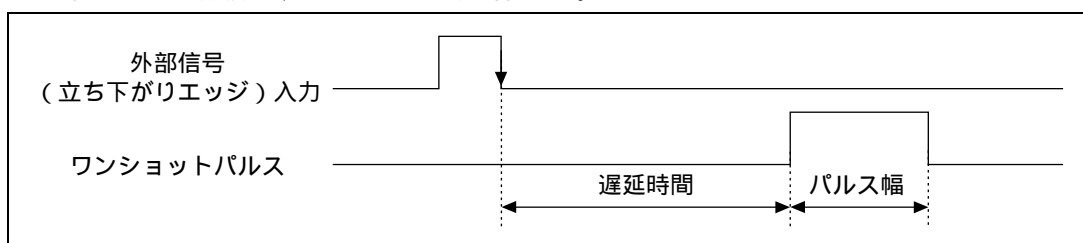


図 2.43 ワンショットパルス出力

2.9.2 使用機能説明

本タスク例ではグループ 0 の DMAC/ch0A、0B、ITU/ch0、3 を使用してワンショットパルスを出力します。

図 2.44 に本タスク例で使用する ITU/ch3 のブロック図を示します。

本タスク例は ITU/ch3 の以下の機能を使用し、ワンショットパルスを出力します。

- コンペアマッチ発生時、バッファレジスタの内容をジェネラルレジスタに転送する機能。(バッファ動作)
- ソフトウェアを介さずハードウェアで自動的にパルスを出力する機能。(アウトプットコンペア)
- コンペアマッチが起きるごとに出力が反転する機能。(トグル出力)
- インพุットキャプチャ時、複数のタイマカウンタを同時クリアする機能。(同期クリア)

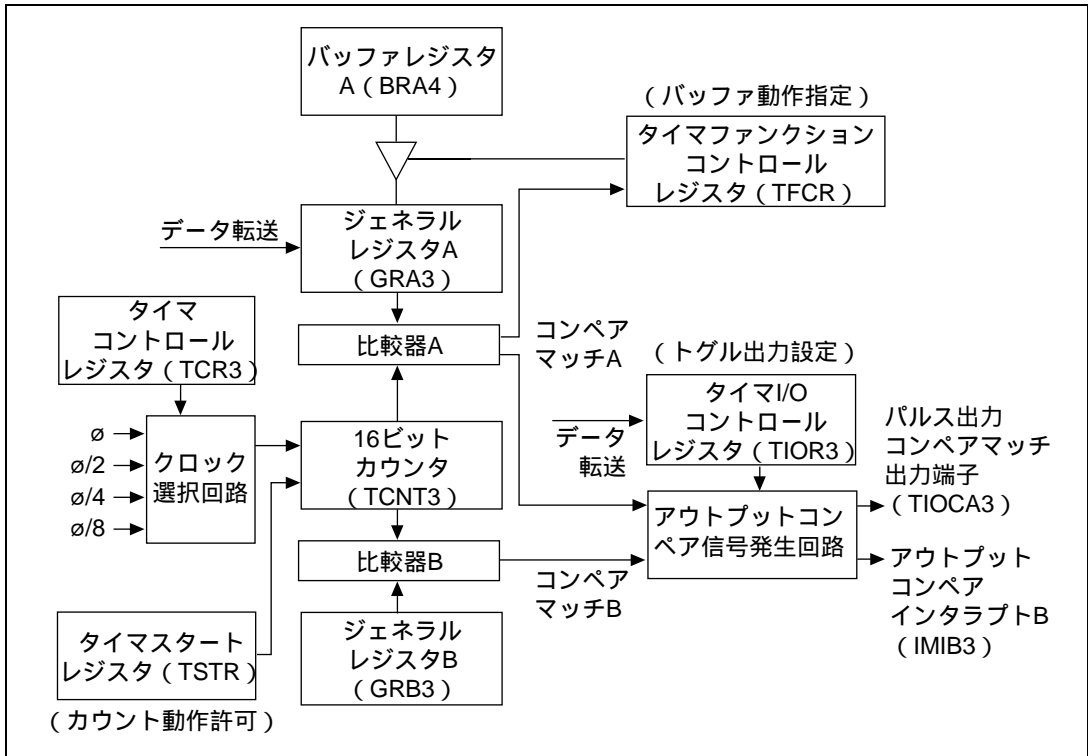


図 2.44 ITU/ch3 ブロック図

図 2.45 に本タスク例で使用する DMAC および ITU のブロック図を示します。本タスク例は DMAC および ITU の以下の機能を使用し、ワンショットパルスを出力します。

- ITU のインプットキャプチャ発生時に DMAC を起動する機能。

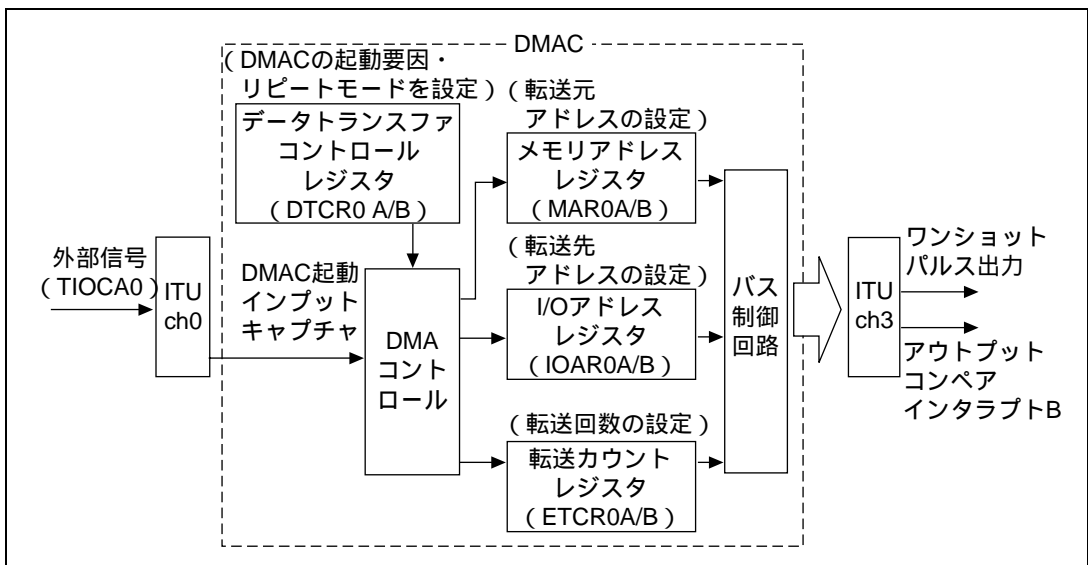


図 2.45 ワンショットパルス出力ブロック図

2. 基礎編

表 2.42 に本タスク例の機能割り付けを示します。本タスク例は表 2.42 に示すように H8/300H の各機能を割り付け、ワンショットパルスを出力します。

表 2.42 H8/300H 機能割り付け

H8/300H の内蔵機能		機能	
ITU	TFCR		バッファ動作の設定をする。
	TSTR		タイマカウンタの動作 / 停止を選択する。
	ch0	TCR0	カウンタクリア要因を設定する。
		TIOR0	インプットキャプチャ機能を選択する。
		TIOCA0	ワンショットパルス出力の基準となるパルスを入力する。
	ch3	TCR3	TCNT3 入力クロック、およびカウンタクリア要因を選択する。
		TIOR3	コンペアマッチ発生時に、TIOCA3 から出力するレベルを設定する。
		TIER3	IMFB フラグによる割り込み要求を許可 / 禁止する。
		TSR3	コンペアマッチの発生を示す。
		GRA3	ワンショットパルスの遅延時間の設定をする。
		BRA3	ワンショットパルスのリセットタイミングを設定する。
		GRB3	ワンショットパルス出力禁止タイミングを設定する。
		TIOCA3	ワンショットパルスを出力する。
DMAC	ch0 A/B	DTCR0 A/B	DMAC の各 ch の動作を設定する。
		MAR0 A/B	各レジスタに転送するデータのアドレスを設定する。
		IORA0 A/B	各 ch の転送先レジスタのアドレスを設定する。
		ETCR0 A/B	各 ch の転送回数を設定する。

2.9.3 動作説明

図 2.46 に動作原理を示します。図 2.46 に示すように H8/3003 のハードウェア処理およびソフトウェアの処理により、ワンショットパルスを出力します。

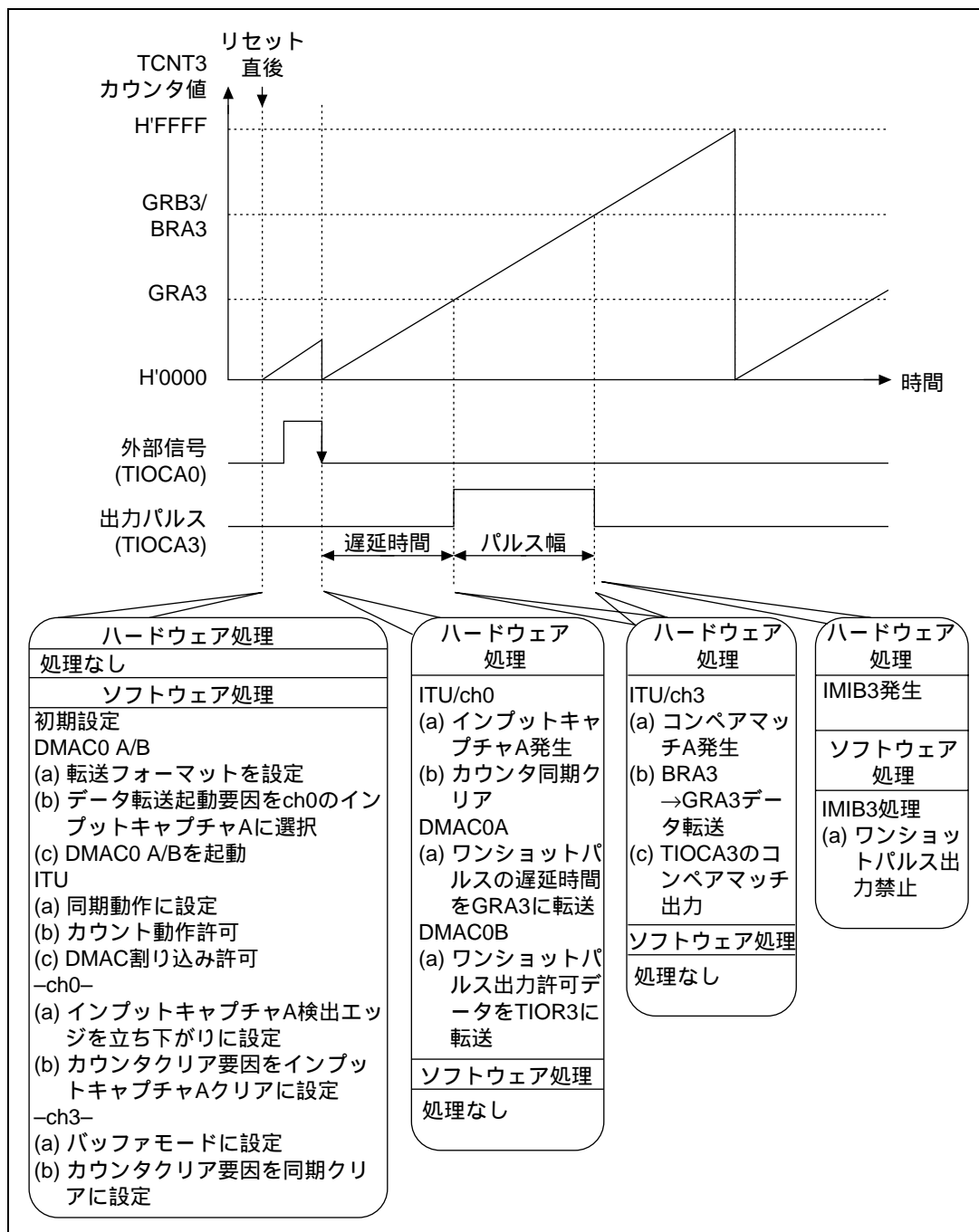


図 2.46 ワンショットパルス出力動作原理

2.9.4 ソフトウェア説明

表 2.43 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	ONEMN	遅延時間およびパルス幅を GRA3、BRA3 に設定し、ワンショットパルスを出力する。
パルス出力禁止	POUTDLE	パルス出力を禁止する。

表 2.44 引数の説明

ラベル名、レジスタ名	機能	データ長	使用モジュール名	入出力
SET_DLY	ワンショットパルスの遅延時間に相当するタイマ値を設定する。 遅延時間は以下の式にて求まる。 遅延時間 (ns) = タイマ値 × 周期 (16MHz 動作時 62.5ns)	1 ワード	メインルーチン	入力
ONE_RST	ワンショットパルスのリセットタイミングに相当するタイマ値を設定する。 リセットタイミングは以下の式にて求まる。 パルスリセットタイミング (ns) = タイマ値 × 周期 (16MHz 動作時 62.5ns)	1 ワード	メインルーチン	入力
IO_CNTR	ワンショットパルス出力許可データを設定する。	1 ワード	メインルーチン	出力

表 2.45 使用内部レジスタ説明

レジスタ名		機能	使用モジュール名	
ITU	TSTR	タイマカウンタの動作 / 停止を選択する。	メインルーチン	
	TFCR	GRA3、BRA3 をバッファ動作に設定する。		
	TSNC	同期動作を設定する。		
	ch0	TCR0	カウンタクリア要因を設定する。	パルス出力禁止 メインルーチン / パルス出力禁止 メインルーチン
		TIOR0	入力パルスの立ち下がりエッジを検出する。	
	ch3	TCR3	TCNT に入力するクロック、およびカウンタクリア要因を設定する。	
		TIOR3	コンペアマッチ A 発生時に、TIOCA3 から出力するレベルを設定する。	
		TIER3	IMFB フラグによる割り込みを許可する。	
		TSR3	GRB によるコンペアマッチの発生を示す。	
		GRA3	ワンショットパルスの遅延時間を設定する。	
		BRA3	ワンショットパルスのリセットタイミングを設定する。	
	GRB3	ワンショットパルス出力禁止タイミングを設定する。		
DMAC	ch0 A/B	DTCR0	DMAC の各チャンネルの動作を設定する。	
		MAR0	各レジスタに転送するデータのアドレスを設定する。	
		IOAR0	各チャンネルの転送先レジスタのアドレスを設定する。	
		ETCR0	各チャンネルの転送回数を設定する。	

表 2.46 使用汎用レジスタ説明

使用モジュール名	レジスタ名	機能
メインルーチン	ER0	データ設定時のワークとして使用する。
パルス出力禁止	ROL	データ設定時のワークとして使用する。

- 使用 RAM 説明
本タスク例では引数以外のRAMは使用していません。

2.9.5 フローチャート

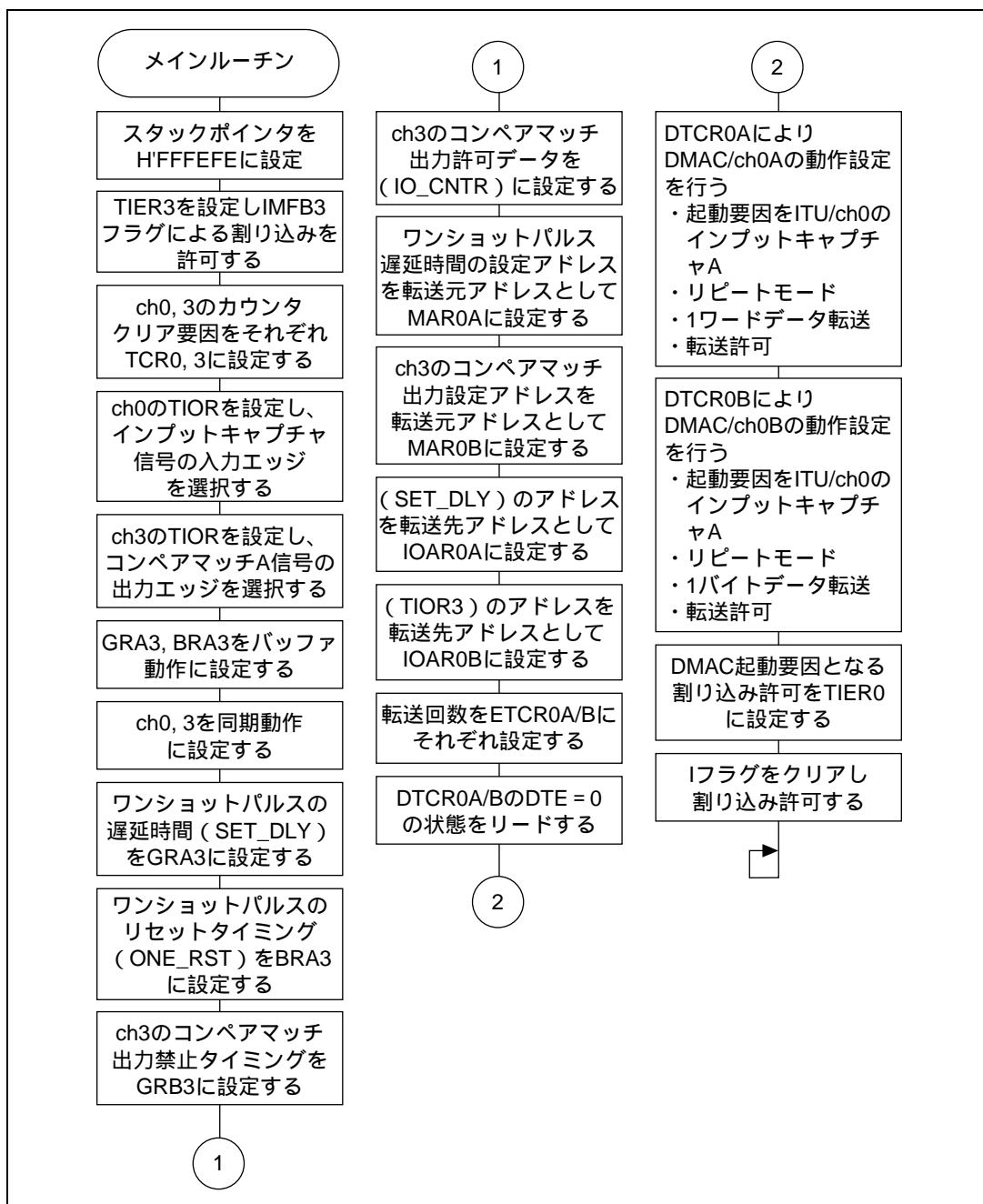


図 2.47 メインルーチン



図 2.48 パルス出力禁止

2.9.6 プログラムリスト

```
1          1 ;*****
2          2 ;*
3          3 ;*          VECTOR ADDRESS
4          4 ;*
5          5 ;*****
6          6 ;
7          7          .CPU      300HA
8 000000    8 ;          .SECTION VECT, CODE, LOCATE=H'000000
9          9 ;
10 000000 00010000 10 RES          .DATA.L  ONEMN
11          11 ;
12 00001C    12          .ORG      H'00001C
13 00001C 00010000 13 NMI          .DATA.L  ONEMN
14 000020 00010000 14 TRAPA0     .DATA.L  ONEMN
15 000024 00010000 15 TRAPA1     .DATA.L  ONEMN
16 000028 00010000 16 TRAPA2     .DATA.L  ONEMN
17 00002C 00010000 17 TRAPA3     .DATA.L  ONEMN
18          18 ;
19 000030    19          .ORG      H'000030
20 000030 00010000 20 IRQ0       .DATA.L  ONEMN
21 000034 00010000 21 IRQ1       .DATA.L  ONEMN
22 000038 00010000 22 IRQ2       .DATA.L  ONEMN
23 00003C 00010000 23 IRQ3       .DATA.L  ONEMN
24 000040 00010000 24 IRQ4       .DATA.L  ONEMN
25 000044 00010000 25 IRQ5       .DATA.L  ONEMN
26 000048 00010000 26 IRQ6       .DATA.L  ONEMN
27 00004C 00010000 27 IRQ7       .DATA.L  ONEMN
28 000050 00010000 28 WOVI0     .DATA.L  ONEMN
29 000054 00010000 29 CMI        .DATA.L  ONEMN
30          30 ;
31 000060    31          .ORG      H'000060
32 000060 00010000 32 IMIA0     .DATA.L  ONEMN
33 000064 00010000 33 IMIB0     .DATA.L  ONEMN
34 000068 00010000 34 OVI0      .DATA.L  ONEMN
35          35 ;
36 000070    36          .ORG      H'000070
37 000070 00010000 37 IMIA1     .DATA.L  ONEMN
38 000074 00010000 38 IMIB1     .DATA.L  ONEMN
39 000078 00010000 39 OVI1      .DATA.L  ONEMN
40          40 ;
41 000080    41          .ORG      H'000080
42 000080 00010000 42 IMIA2     .DATA.L  ONEMN
43 000084 00010000 43 IMIB2     .DATA.L  ONEMN
44 000088 00010000 44 OVI2      .DATA.L  ONEMN
45          45 ;
```

```

46 000090          46          .ORG      H'000090
47 000090 00010000 47 IMIA3    .DATA.L  ONEMN
48 000094 0001007A 48 IMIB3    .DATA.L  POUTDLE
49 000098 00010000 49 OVI3     .DATA.L  ONEMN
50
51 0000A0          51          .ORG      H'0000A0
52 0000A0 00010000 52 IMIA4    .DATA.L  ONEMN
53 0000A4 00010000 53 IMIB4    .DATA.L  ONEMN
54 0000A8 00010000 54 OVI4     .DATA.L  ONEMN
55
56 0000B0          56          .ORG      H'0000B0
57 0000B0 00010000 57 DEND0A   .DATA.L  ONEMN
58 0000B4 00010000 58 DEND0B   .DATA.L  ONEMN
59 0000B8 00010000 59 DEND1A   .DATA.L  ONEMN
60 0000BC 00010000 60 DEND1B   .DATA.L  ONEMN
61 0000C0 00010000 61 DEND2A   .DATA.L  ONEMN
62 0000C4 00010000 62 DEND2B   .DATA.L  ONEMN
63 0000C8 00010000 63 DEND3A   .DATA.L  ONEMN
64 0000CC 00010000 64 DEND3B   .DATA.L  ONEMN
65 0000D0 00010000 65 ERI0     .DATA.L  ONEMN
66 0000D4 00010000 66 RXI0     .DATA.L  ONEMN
67 0000D8 00010000 67 TXI0     .DATA.L  ONEMN
68 0000DC 00010000 68 TEI0     .DATA.L  ONEMN
69 0000E0 00010000 69 ERI1     .DATA.L  ONEMN
70 0000E4 00010000 70 RXI1     .DATA.L  ONEMN
71 0000E8 00010000 71 TXI1     .DATA.L  ONEMN
72 0000EC 00010000 72 TEI1     .DATA.L  ONEMN
73 0000F0 00010000 73 ADI      .DATA.L  ONEMN
74
75                75          ;*****
76                76          ;*
77                77          ;*          RAM ALLOCATION
78                78          ;*
79                79          ;*****
80
81 FFFF00          81          .SECTION  RAM,DATA,LOCATE=H'FFFF00
82
83 FFFF00 00000002 83 SET_DLY  .RES.W  1          ;Pulse delay time
84 FFFF02 00000002 84 ONE_RST  .RES.W  1          ;Pulse reset timing
85 FFFF04 00000001 85 IO_CNTR  .RES.B  1          ;I/O control
86
87                87          ;*****
88                88          ;*
89                89          ;*          SYMBOL DEFINITIONS
90                90          ;*
91                91          ;*****
92                92          ;

```

2. 基礎編

```

93      00FFFF20      93  MAR0A      .EQU      H'FFFF20      ;Memory address register 0A
94      00FFFF24      94  ETCR0A     .EQU      H'FFFF24      ;Execute transfer control
                                     ;register 0A
95      00FFFF26      95  IOAR0A     .EQU      H'FFFF26      ;I/O address register 0A
96      00FFFF27      96  DTCR0A     .EQU      H'FFFF27      ;Data transfer control
                                     ;register 0A
97
98      00FFFF28      98  MAR0B     .EQU      H'FFFF28      ;Memory address register 0B
99      00FFFF2C      99  ETCR0B     .EQU      H'FFFF2C      ;Execute transfer control
                                     ;register 0B
100     00FFFF2E      100 IOAR0B     .EQU      H'FFFF2E      ;I/O address register 0B
101     00FFFF2F      101 DTCR0B     .EQU      H'FFFF2F      ;Data transfer control
                                     ;register 0B
102
103
104     00FFFF60      104 TSTR        .EQU      H'FFFF60      ;Timer start register
105     00FFFF61      105 TSNC        .EQU      H'FFFF61      ;Timer sync register
106     00FFFF63      106 TFCR        .EQU      H'FFFF63      ;Timer function control
                                     ;register
107
108     00FFFF64      108 TCR0        .EQU      H'FFFF64      ;Timer control register0
109     00FFFF65      109 TIOR0       .EQU      H'FFFF65      ;Timer I/O control register0
110     00FFFF66      110 TIER0       .EQU      H'FFFF66      ;Timer interrupt enable
                                     ;register
111     00000000      111 IMIEA      .EQU      0
112
113     00FFFF82      113 TCR3        .EQU      H'FFFF82      ;Timer control register3
114     00FFFF83      114 TIOR3       .EQU      H'FFFF83      ;Timer I/O control register3
115     00FFFF84      115 TIER3       .EQU      H'FFFF84      ;Timer interrupt enable
                                     ;register3
116     00FFFF85      116 TSR3        .EQU      H'FFFF85      ;Timer status register3
117     00000001      117 IMPB3       .EQU      1          ;Inputcapture flag3
118     00FFFF86      118 TCNT3       .EQU      H'FFFF86      ;Timer counter3
119     00FFFF88      119 GRA3        .EQU      H'FFFF88      ;General register A3
120     00FFFF8A      120 GRB3        .EQU      H'FFFF8A      ;General register B3
121     00FFFF8C      121 BRA3        .EQU      H'FFFF8C      ;Buffer register A3
122
123     ;*****
124     ;*
125     ;*          MAIN PROGRAM : ONEMN
126     ;*
127     ;*****
128
129 010000      129          .SECTION  PROG, CODE, LOCATE=H'010000
130
131     00010000      131 ONEMN:      .EQU      $
132 010000 7A0700FFFEFE 132          MOV.L    #H'FFFEFE, SP ;Initialize stack pointer

```

```

133          133 ;
134          134 ;-----ITU INITIALIZE-----
135          135 ;
136 010006 F8FA 136          MOV.B    #B'11111010,R0L
137 010008 3884 137          MOV.B    R0L,@TIER3 ;Initialize TIER3(1->IMIEB)
138 01000A F8A0 138          MOV.B    #B'10100000,R0L
139 01000C 3864 139          MOV.B    R0L,@TCR0 ;Initialize TCR0
140 01000E F8E0 140          MOV.B    #B'11100000,R0L
141 010010 3882 141          MOV.B    R0L,@TCR3 ;Initialize TCR3
142          142
143 010012 F88D 143          MOV.B    #B'10001101,R0L
144 010014 3865 144          MOV.B    R0L,@TIOR0 ;Initialize TIOR0
145 010016 F8C1 145          MOV.B    #B'11000001,R0L
146 010018 3863 146          MOV.B    R0L,@TFCR ;Initialize TFCR
147          147
148 01001A F8E9 148          MOV.B    #B'11101001,R0L
149 01001C 3861 149          MOV.B    R0L,@TSNC ;Initialize TSNC
150          150
151 01001E 6B00FF00 151         MOV.W    @SET_DLY,R0
152 010022 6B80FF88 152         MOV.W    R0,@GRA3 ;Set delay time
153 010026 6B00FF02 153         MOV.W    @ONE_RST,R0
154 01002A 6B80FF8C 154         MOV.W    R0,@BRA3 ;Set pulse reset timing
155 01002E 6B80FF8A 155         MOV.W    R0,@GRB3 ;Set pulse output disable
                                ;time
156 010032 F88B 156          MOV.B    #B'10001011,R0L
157 010034 3804 157          MOV.B    R0L,@IO_CNTR ;Set pulse output enable data
158          158
159          159 ;
160          160 ;-----DMAC INITIALIZE-----
161          161 ;
162 010036 7A0000FFFF00 162        MOV.L    #SET_DLY,ER0
163 01003C 01006B80FF20 163        MOV.L    ER0,@MAR0A ;Set base address
164 010042 7A0000FFFF04 164        MOV.L    #IO_CNTR,ER0
165 010048 01006B80FF28 165        MOV.L    ER0,@MAR0B ;Set base address
166          166
167 01004E F888 167          MOV.B    #H'88,R0L
168 010050 3826 168          MOV.B    R0L,@IOAR0A ;Set execute address
169 010052 F883 169          MOV.B    #H'83,R0L
170 010054 382E 170          MOV.B    R0L,@IOAR0B ;Set execute address
171          171
172 010056 79000101 172          MOV.W    #H'0101,R0
173 01005A 6B80FF24 173          MOV.W    R0,@ETCR0A ;Set execute count
174 01005E 6B80FF2C 174          MOV.W    R0,@ETCR0B ;Set execute count
175          175
176 010062 2827 176          MOV.B    @DTCR0A,R0L
177 010064 282F 177          MOV.B    @DTCR0B,R0L
178          178

```

2. 基礎編

```

179 010066 F8D0      179          MOV.B    #B'11010000,R0L
180 010068 3827      180          MOV.B    R0L,@DTCR0A ;Set 0Achannel control data
181 01006A F890      181          MOV.B    #B'10010000,R0L
182 01006C 382F      182          MOV.B    R0L,@DTCR0B ;Set 0Bchannel control data
183                183
184 01006E F8E9      184          MOV.B    #B'11101001,R0L
185 010070 3860      185          MOV.B    R0L,@TSTR   ;Start TCNT1,3
186                186
187 010072 7F667000  187          BSET     #IMIEA,@TIER0 ;Enable DMAC
188                188
189 010076 0700      189          LDC.B    #0,CCR      ;Enable interrupt
190                190
191 010078 40FE      191  ONEMN99  BRA     ONEMN99
192                192 ;
193                193 ;*****
194                194 ;*
195                195 ;*      NAME : POUTDLE(PULSE OUTUPT DISABLE) *
196                196 ;*
197                197 ;*****
198                198 ;*
199                199 ;*      ENTRY   : NOTHING
200                200 ;*      RETURNS  : TIOR3(TIMER I/O REG 3)
201                201 ;*
202                202 ;*****
203                203 ;
204                204 POUTDLE  .EQU    $
205 01007A 01006DF0  205          PUSH.L   ER0        ;Escape register
206 01007E 7F857210  206          BCLR     #IMFB3,@TSR3 ;Clear IMFB3 request
                                   ;(0->IMFB3)
207 010082 F889      207          MOV.B    #B'10001001,R0L
208 010084 3883      208          MOV.B    R0L,@TIOR3  ;Set pulse output disable
                                   ;data
209 010086 01006D70  209          POP.L    ER0        ;Return register
210 01008A 5670      210          RTE
211                211
212                212
213                213          .END
*****TOTAL ERRORS      0
*****TOTAL WARNINGS    0

```


2.10 4ビット×4系統出力

MCU : H8/3003

使用機器 : プログラマブルタイミングパターンコントローラ (TPC) (4ビット×4系統出力)

2.10.1 仕様

図 2.49 に示すように、TPC 出力を使用して 4 ビット×4 系統の非同期パルスを出力します。TPC の起動要因には、ITU のコンペアマッチを使用します。

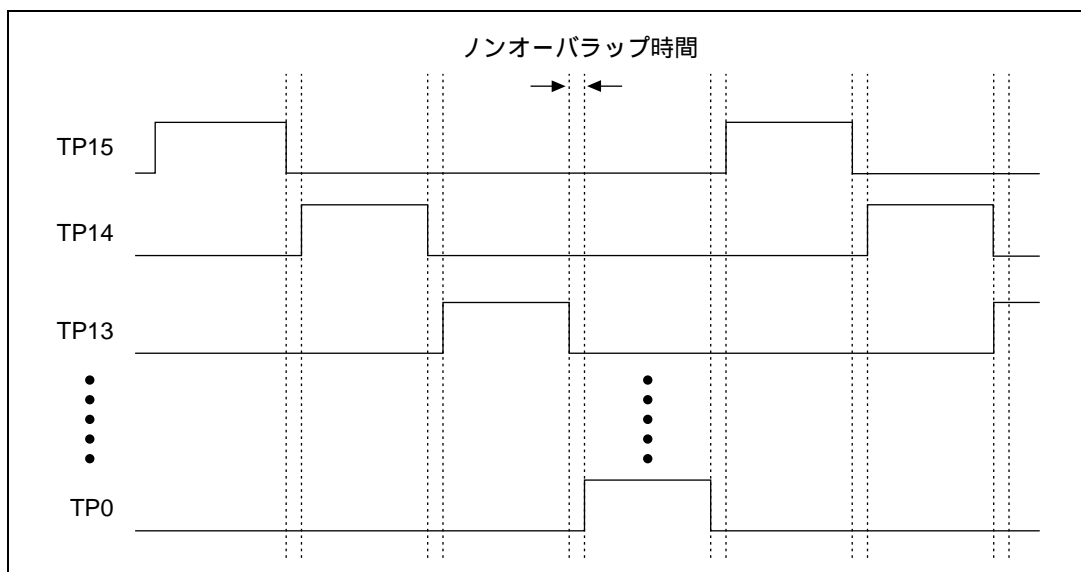


図 2.49 4ビット×4系統出力例

2.10.2 使用機能説明

本タスク例では、ITU4 チャンネル、4 ビット単位の TPC 出力グループ 3~0 を使用し、4 ビット×4 系統の非同期パルスを出力します。

図 2.50 に本タスク例で使用する TPC のブロック図を示します。本タスク例では、以下の機能を使用しています。

- 4 ビット単位のグループで出力トリガ信号が選択可能で最大 4 ビット×4 系統出力が可能。
- ITU の 4 チャンネルのコンペアマッチ信号の中からグループごとに出力トリガ信号の選択が可能。
- 複数のパルス出力の間のノンオーバーラップ期間を設定することが可能。

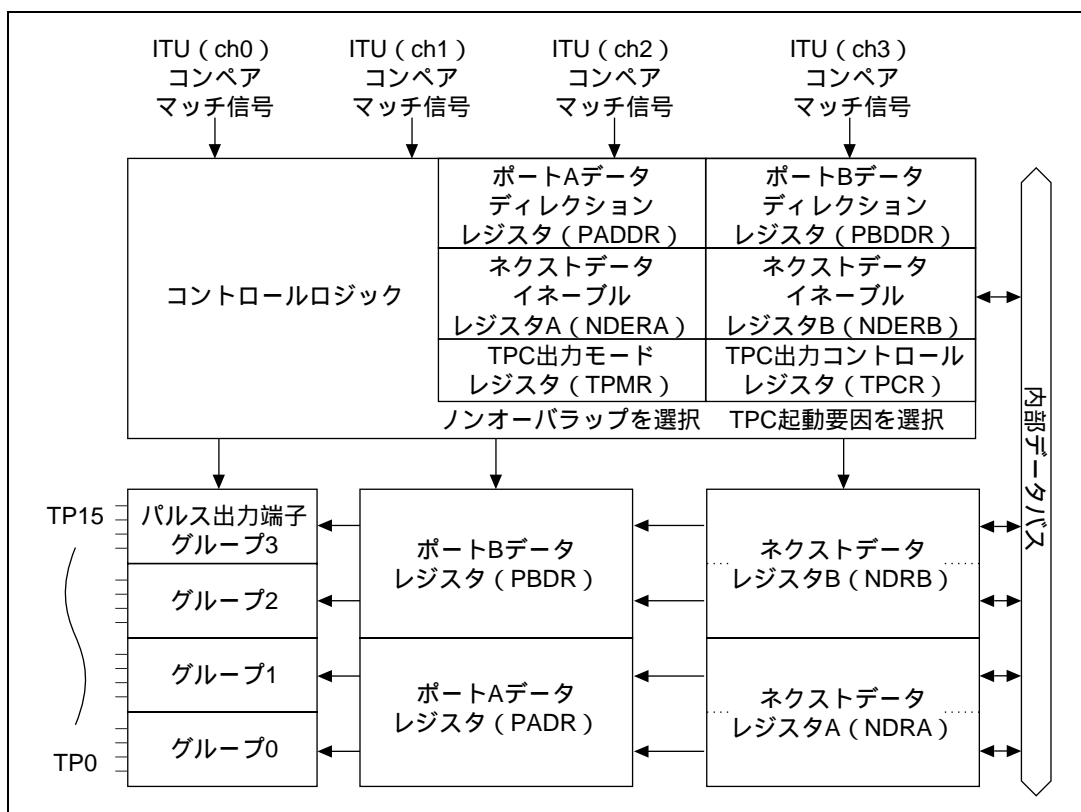


図 2.50 タイミングパターンコントローラブロック図

表 2.47 に本タスク例の機能割り付けを示します。表 2.47 に示すように ITU および TPC の機能を割り付け、4 ビット×4 系統出力を行います。

表 2.47 TPC 機能割り付け

内蔵周辺機能	内蔵機能	機能
TPC	TP15~TP0	4 ビット×4 系統の出力を行う
	TPMR	ノンオーバーラップモードを設定する
	TPCR	TPC 出力の出力トリガ信号を設定する
	NDERH	TP15~TP8 の TPC 出力を許可する
	NDERL	TP7~TP0 の TPC 出力を許可する
	NDRA	次に出力する TPC 出力データを格納する
	NDRB	次に出力する TPC 出力データを格納する
	NDRC	次に出力する TPC 出力データを格納する
	NDRD	次に出力する TPC 出力データを格納する
	PADDR	TPC 出力端子の設定をする
	PBDDR	TPC 出力端子の設定をする
	PADR	TP7~TP0 の出力データを格納する
	PBDR	TP15~TP8 の出力データを格納する
ITU	GRA0~3	ノンオーバーラップ時間を設定する
	GRB0~3	TPC 出力トリガの周期を設定する
	TCR0~3	カウンタクロックおよびカウンタクリア要因を設定する
	TSR0~3	コンペアマッチの発生を示す
	TIER0~3	IMFA 割り込みを許可する
	TSTR	タイマのカウント動作を許可する

2.10.3 動作説明

図 2.51 に TPC 出力グループ 3 を使用したデータ出力動作原理を示します。H8/3003 のハードウェア処理およびソフトウェアの処理により 4 相のノンオーバーラップ出力を行います。

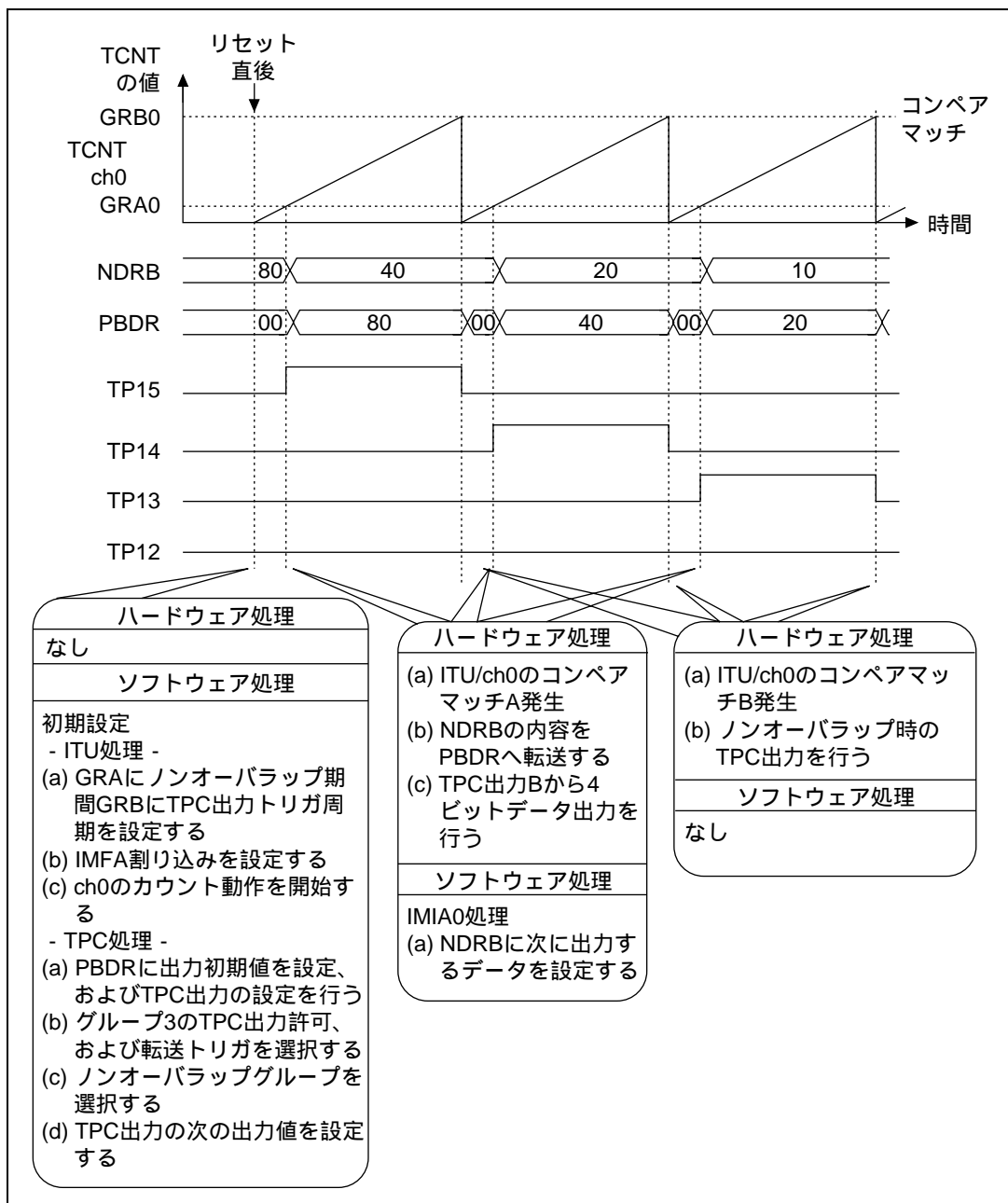


図 2.51 4 相ノンオーバーラップ出力動作例

2.10.4 ソフトウェア説明

表 2.48 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	TPC16MN	TPC、ITU の初期設定を行う
データ設定 0	SETDAT0	TPC の NDRA (グループ 0) に次に出力するデータを設定する
データ設定 1	SETDAT1	TPC の NDRA (グループ 1) に次に出力するデータを設定する
データ設定 2	SETDAT2	TPC の NDRB (グループ 2) に次に出力するデータを設定する
データ設定 3	SETDAT3	TPC の NDRB (グループ 3) に次に出力するデータを設定する

表 2.49 引数の説明

ラベル名、レジスタ名	機能	データ長	使用モジュール名	入出力
INC_ADR0	NDAT_TAB の番地に加算するデータを格納する	1 バイト	メインルーチン	出力
			データ設定 0	入力
INC_ADR1	NDAT_TAB の番地に加算するデータを格納する	1 バイト	メインルーチン	出力
			データ設定 1	入力
INC_ADR2	NDAT_TAB の番地に加算するデータを格納する	1 バイト	メインルーチン	出力
			データ設定 2	入力
INC_ADR3	NDAT_TAB の番地に加算するデータを格納する	1 バイト	メインルーチン	出力
			データ設定 3	入力

表 2.50 使用内部レジスタ

レジスタ名	機能	使用モジュール名	
TPC	PADDR	TP7 ~ TP0 までの TPC 出力を許可	メインルーチン
	PBDDR	TP15 ~ TP8 までの TPC 出力を許可	メインルーチン
	PADR	TP7 ~ TP0 出力パターンデータを格納	メインルーチン
	PBDR	TP15 ~ TP8 出力パターンデータを格納	メインルーチン
	TPMR	TP15 ~ TP0 をノンオーバーラップ出力に設定	メインルーチン
	TPCR	TP15 ~ TP0 の出力トリガを ITUch0 のコンペアマッチに設定	メインルーチン
	NDERA	TPC 出力 TP7 ~ TP0 を許可する	メインルーチン
	NDERB	TPC 出力 TP15 ~ TP8 を許可する	メインルーチン
	NDRA	TP7 ~ TP0 の次に出力するパターンを設定	メインルーチン データ設定 1、2
	NDRB	TP15 ~ TP8 の次に出力するパターンを設定	メインルーチン データ設定 3、4
ITU	GRA0 ~ 3	ノンオーバーラップ時間を設定	メインルーチン
	BRB0 ~ 3	TPC 出力トリガの周期を設定	メインルーチン
	TCR0 ~ 3	TCR を以下のように設定する <ul style="list-style-type: none"> GRB のコンペアマッチでカウンタクリア 内部クロック でカウント 	メインルーチン
	TSR0 ~ 3	コンペアマッチの発生を示す	メインルーチン
	TIER0 ~ 3	IMFA 割り込みを許可	メインルーチン
	TSTR	TCNT0 カウント動作を許可	メインルーチン

2. 基礎編

- 使用 RAM 説明
本タスクでは引数以外のRAMは使用していません。

表 2.51 データテーブル説明

テーブル名	機能	データ長	データ容量
NDAT_TAB0	TPC グループ 0 から出力するデータを格納	1 バイト	5 バイト
NDAT_TAB1	TPC グループ 1 から出力するデータを格納	1 バイト	5 バイト
NDAT_TAB2	TPC グループ 2 から出力するデータを格納	1 バイト	5 バイト
NDAT_TAB3	TPC グループ 3 から出力するデータを格納	1 バイト	5 バイト

- 使用汎用レジスタ

31	ERn	RnH	RnL	0
0	全ルーチンにてデータ設定時のワークとして使用			
1				
2				
3				
4				
5				
6				
7				

図 2.52 使用汎用レジスタ

2.10.5 フローチャート



図 2.53 メインルーチン

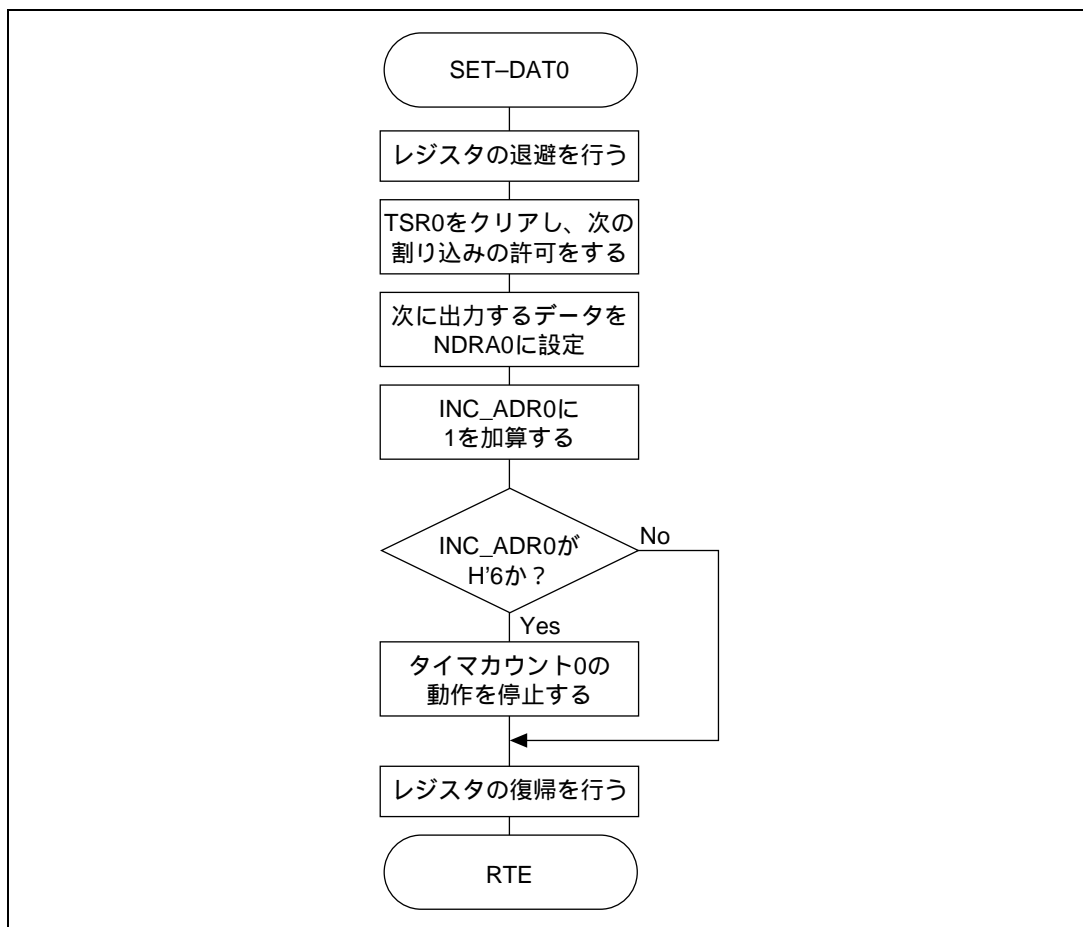


図 2.54 データ設定 0

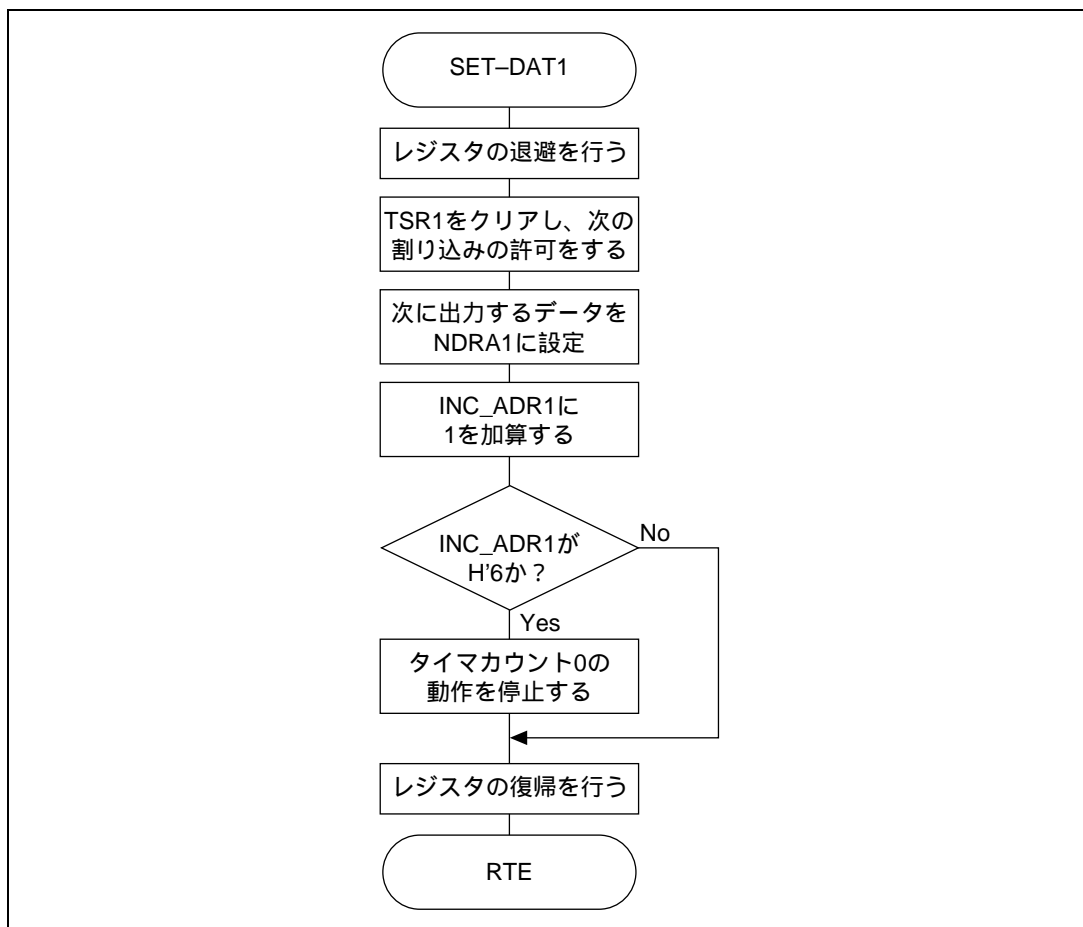


図 2.55 データ設定 1

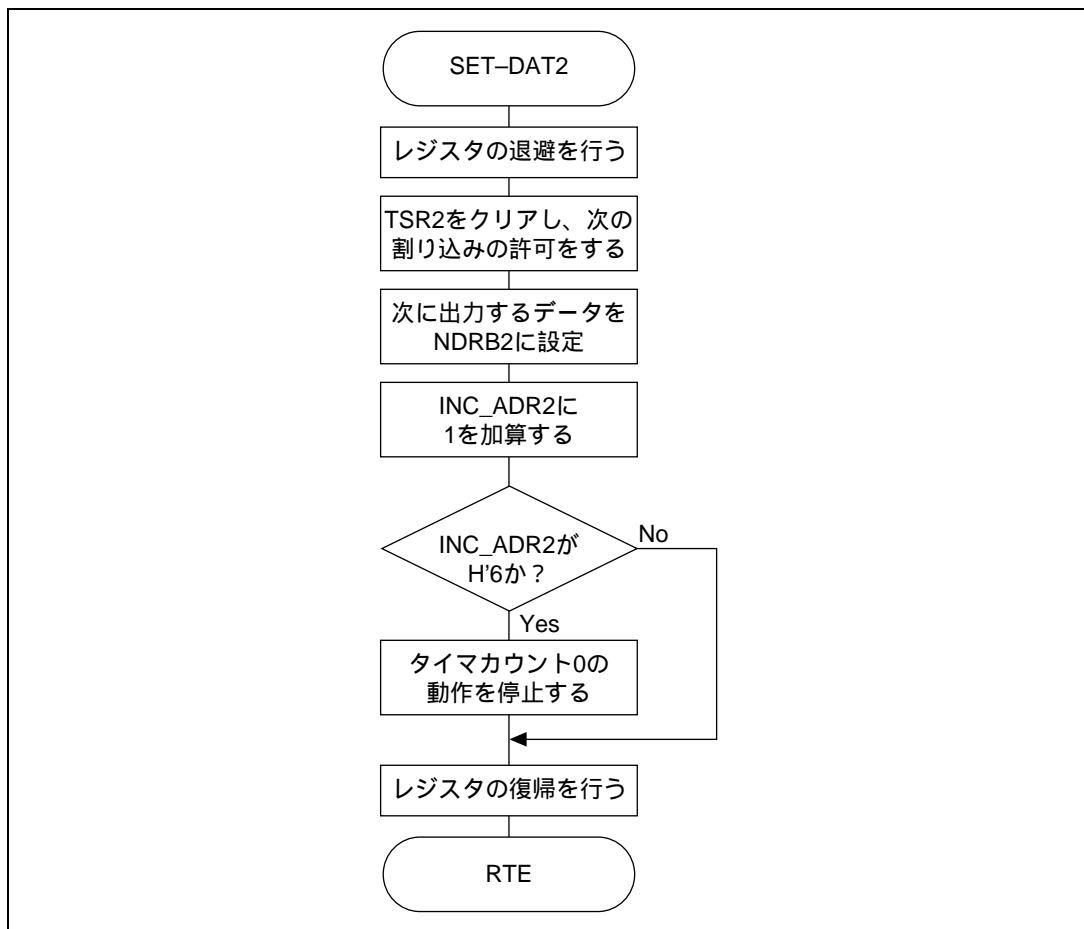


図 2.56 データ設定 2

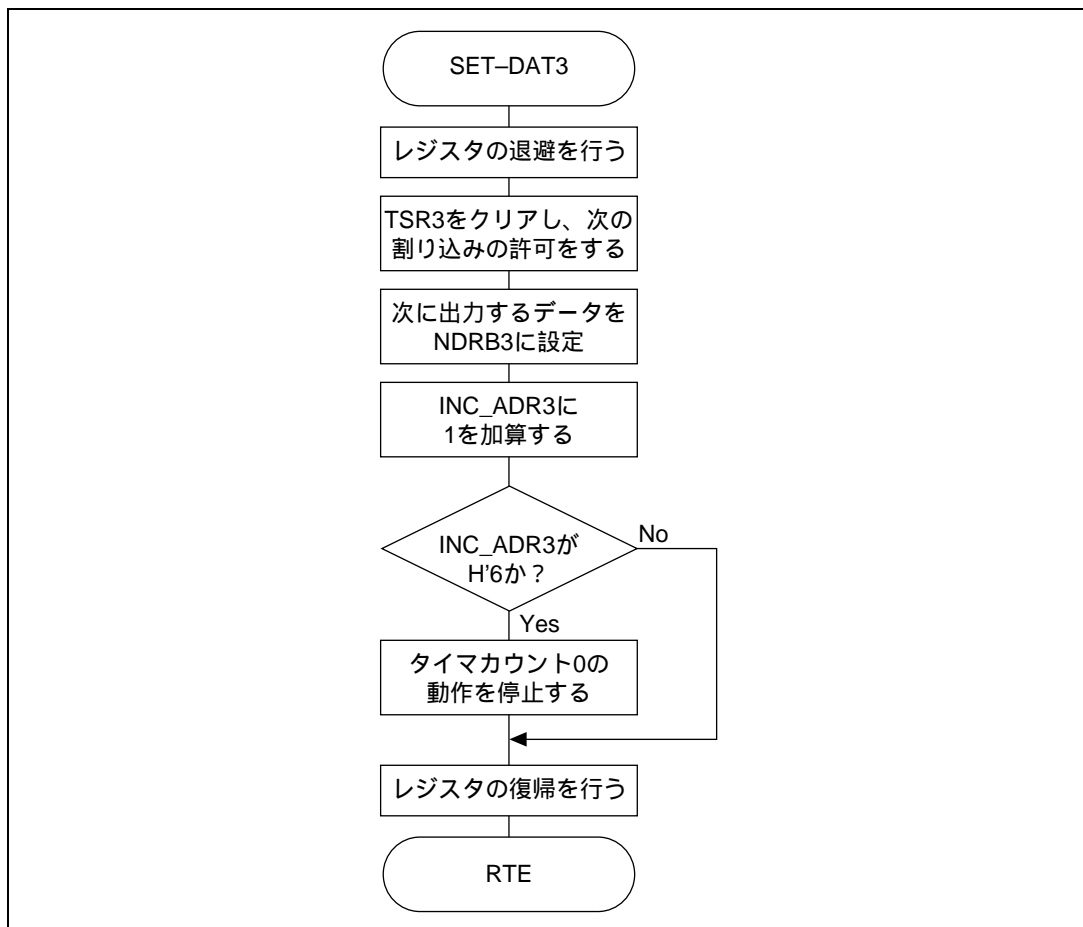


図 2.57 データ設定 3

2.10.6 プログラムリスト

```

1          1  ;*****
2          2  ;*
3          3  ;*          VECTOR ADDRESS
4          4  ;*
5          5  ;*****
6          6
7          7          .CPU      300HA
8 000000   8          .SECTION  VECT, CODE, LOCATE=H'000000
9          9  ;
10 000000 00010000 10 RES          .DATA.L  TPC16MN
11          11 ;
12 00001C   12          .ORG      H'00001C
13 00001C 00010000 13 NMI          .DATA.L  TPC16MN
14          14 ;
15 000030   15          .ORG      H'000030
16 000030 00010000 16 IRQ0         .DATA.L  TPC16MN
17 000034 00010000 17 IRQ1         .DATA.L  TPC16MN
18 000038 00010000 18 IRQ2         .DATA.L  TPC16MN
19 00003C 00010000 19 IRQ3         .DATA.L  TPC16MN
20 000040 00010000 20 IRQ4         .DATA.L  TPC16MN
21 000044 00010000 21 IRQ5         .DATA.L  TPC16MN
22 000048 00010000 22 IRQ6         .DATA.L  TPC16MN
23 00004C 00010000 23 IRQ7         .DATA.L  TPC16MN
24 000050 00010000 24 WOVI0        .DATA.L  TPC16MN
25 000054 00010000 25 CMI          .DATA.L  TPC16MN
26          26 ;
27 000060   27          .ORG      H'000060
28 000060 000100CB 28 IMIA0        .DATA.L  SETDAT0
29 000064 00010000 29 IMIB0        .DATA.L  TPC16MN
30 000068 00010000 30 OVI0         .DATA.L  TPC16MN
31          31 ;
32 000070   32          .ORG      H'000070
33 000070 00010102 33 IMIA1        .DATA.L  SETDAT1
34 000074 00010000 34 IMIB1        .DATA.L  TPC16MN
35 000078 00010000 35 OVI1         .DATA.L  TPC16MN
36          36 ;
37 000080   37          .ORG      H'000080
38 000080 0001013C 38 IMIA2        .DATA.L  SETDAT2
39 000084 00010000 39 IMIB2        .DATA.L  TPC16MN
40 000088 00010000 48 OVI2         .DATA.L  TPC16MN
41          41 ;
42 000090   42          .ORG      H'000090
43 000090 00010176 43 IMIA3        .DATA.L  SETDAT3
44 000094 00010000 44 IMIB3        .DATA.L  TPC16MN
45 000098 00010000 45 OVI3         .DATA.L  TPC16MN

```

```

46          46 ;
47 0000A0          47          .ORG      H'0000A0
48 0000A0 00010000 48 IMMIA4          .DATA.L  TPC16MN
49 0000A4 00010000 49 IMIB4          .DATA.L  TPC16MN
50 0000A8 00010000 50 OVI4          .DATA.L  TPC16MN
51          51 ;
52 0000B0          52          .ORG      H'0000B0
53 0000B0 00010000 53 DEND0A         .DATA.L  TPC16MN
54 0000B4 00010000 54 DEND0B         .DATA.L  TPC16MN
55 0000B8 00010000 55 DEND1A         .DATA.L  TPC16MN
56 0000BC 00010000 56 DEND1B         .DATA.L  TPC16MN
57 0000C0 00010000 57 DEND2A         .DATA.L  TPC16MN
58 0000C4 00010000 58 DEND2B         .DATA.L  TPC16MN
59 0000CB 00010000 59 DEND3A         .DATA.L  TPC16MN
60 0000CC 00010000 60 DEND3B         .DATA.L  TPC16MN
61 0000D0 00010000 61 ERI0          .DATA.L  TPC16MN
62 0000D4 00010000 62 RXI0          .DATA.L  TPC16MN
63 0000D8 00010000 63 TXI0          .DATA.L  TPC16MN
64 0000DC 00010000 64 TEI0          .DATA.L  TPC16MN
65 0000E0 00010000 65 ERI1          .DATA.L  TPC16MN
66 0000E4 00010000 66 RXI1          .DATA.L  TPC16MN
67 0000E8 00010000 67 TXI1          .DATA.L  TPC16MN
68 0000EC 00010000 68 TEI1          .DATA.L  TPC16MN
69 0000F0 00010000 69 ADI           .DATA.L  TPC16MN
70          70 ;
71          71 ;*****
72          72 ;* *
73          73 ;*      RAM ALLOCATION *
74          74 ;* *
75          75 ;*****
76          76 ;
77 FFFF00          77          .SECTION  RAM,DATA,LOCATE=H'FFFF00
78          78 ;
79 FFFF00 00000001 79 INC_ADR0       .RES.B   1          ;Next data address 0
80 FFFF01 00000001 80 INC_ADR1       .RES.B   1          ;Next data address 1
81 FFFF02 00000001 81 INC_ADR2       .RES.B   1          ;Next data address 2
82 FFFF03 00000001 82 INC_ADR3       .RES.B   1          ;Next data address 3
83          83 ;
84          84 ;*****
85          85 ;* *
86          86 ;*      SYMBOL DEFINITIONS *
87          87 ;* *
88          88 ;*****
89          89 ;
90          90 ;***** ITU *****
91          00FFFF60 91 TSTR:          .EQU     H'FFFF60   ;Timer start register
92          00000003 92 STR3:          .EQU     3          ;Timer 3 count start bit

```

2. 基礎編

```

93      00000002      93  STR2:      .EQU      2          ;Timer 2 count start bit
94      00000001      94  STR1:      .EQU      1          ;Timer 1 count start bit
95      00000000      95  STR0:      .EQU      0          ;Timer 0 count start bit
96      ;
97      ;***** ITU ch0 *****
98      00FFFF64      98  TCR0:      .EQU      H'FFFF64    ;timer control register
99      00FFFF66      99  TIER0:     .EQU      H'FFFF66    ;timer interrupt enable
                                     ;register
100     00FFFF67      100 TSR0      .EQU      H'FFFF67    ;timer status register
101     00FFFF6A      101 GRA0H:     .EQU      H'FFFF6A    ;General register A
102     00FFFF6C      102 GRB0H:     .EQU      H'FFFF6C    ;General register B
103     ;
104     ;***** ITU ch1 *****
105     00FFFF6E      105 TCR1:      .EQU      H'FFFF6E    ;timer control register
106     00FFFF70      106 TIER1:     .EQU      H'FFFF70    ;timer interrupt enable
                                     ;register
107     00FFFF71      107 TSR1:      .EQU      H'FFFF71    ;timer status register
108     00FFFF74      108 GRA1H:     .EQU      H'FFFF74    ;General register A
109     00FFFF76      109 GRB1H:     .EQU      H'FFFF76    ;General register B
110     ;
111     ;***** ITU ch2 *****
112     00FFFF78      112 TCR2:      .EQU      H'FFFF78    ;timer control register
113     00FFFF7A      113 TIER2:     .EQU      H'FFFF7A    ;timer interrupt enable
                                     ;register
114     00FFFF7B      114 TSR2:      .EQU      H'FFFF7B    ;timer status register
115     00FFFF7E      115 GRA2H:     .EQU      H'FFFF7E    ;General register A
116     00FFFF80      116 GRB2H:     .EQU      H'FFFF80    ;General register B
117     ;
118     ;***** ITU ch3 *****
119     00FFFF82      119 TCR3:      .EQU      H'FFFF82    ;timer control register
120     00FFFF84      120 TIER3:     .EQU      H'FFFF84    ;timer interrupt enable
                                     ;register
121     00FFFF85      121 TSR3:      .EQU      H'FFFF85    ;timer status register
122     00FFFF88      122 GRA3H:     .EQU      H'FFFF88    ;General register A
123     00FFFF8A      123 GRB3H:     .EQU      H'FFFF8A    ;General register B
124     ;
125     ;***** TPC *****
126     00FFFFA0      126 TPMP:      .EQU      H'FFFFA0    ;TPC output mod register
127     00FFFFA1      127 TPCR:      .EQU      H'FFFFA1    ;TPC output control register
128     00FFFFA2      128 NDERB:     .EQU      H'FFFFA2    ;Next data enable register B
129     00FFFFA3      129 NDERA:     .EQU      H'FFFFA3    ;Next data enable register A
130     00FFFFA4      130 NDRB3:     .EQU      H'FFFFA4    ;Next data register 3
131     00FFFFA5      131 NDRA1:     .EQU      H'FFFFA5    ;Next data register 1
132     00FFFFA6      132 NDRB2:     .EQU      H'FFFFA6    ;Next data register 2
133     00FFFFA7      133 NDRA0:     .EQU      H'FFFFA7    ;Next data register 0
134     00FFFFD1      134 PADDR:     .EQU      H'FFFFD1    ;Port A data direction
                                     ;register

```

```

135      00FFFFD3      135  PADR:      .EQU      H'FFFFD3      ;Port A data register
136      00FFFFD4      136  PBDDR:     .EQU      H'FFFFD4      ;Port B data direction
                                           ;register
137      00FFFFD6      137  PBDR:      .EQU      H'FFFFD6      ;Port B data register
138      ;
139      ;*****
140      ;*
141      ;*          MAIN PROGRAM : TPC16MN
142      ;*
143      ;*****
144      ;
145      010000      145          .SECTION  PROG, CODE, LOCATE=H'010000
146      ;
147      00010000      147  TPC16MN:  .EQU      $
148      010000  7A0700FFFEFE  148          MOV.L      #H'FFFEFE, SP ;Init. stack pointer
149      ;
150      ;***** INITIALZE ITU ch0 *****
151      010006  7900000A      151          MOV.W      #H'000A, R0
152      01000A  6B80FF6A      152          MOV.W      R0, @GRA0H      ;Set non overlap time
153      01000E  790001F4      153          MOV.W      #D'500, R0
154      010012  6B80FF6C      154          MOV.W      R0, @GRB0H      ;Set timer counter
155      010016  F840          155          MOV.B      #H'40, R0L
156      01001B  3864          156          MOV.B      R0L, @TCR0      ;Init. TCR0(1->CCLR1)
157      01001A  F801          157          MOV.B      #H'01, R0L
158      01001C  3866          158          MOV.B      R0L, @TIER0      ;Init. TIER0(1->IMIEA)
159      ;
160      ;***** INITIALZE ITU chl *****
161      01001E  7900000A      161          MOV.W      #H'000A, R0
162      010022  6B80FF74      162          MOV.W      R0, @GRA1H      ;Set non overlap time
163      010026  790003E8      163          MOV.W      #D'1000, R0
164      01002A  6B80FF76      164          MOV.W      R0, @GRB1H      ;Set timer counter
165      01002E  F840          165          MOV.B      #H'40, R0L
166      010030  386E          166          MOV.B      R0L, @TCR1      ;Init. TCR1(1->CCLR1)
167      010032  F801          167          MOV.B      #H'01, R0L
168      010034  3870          168          MOV.B      R0L, @TIER1      ;Init. TIER1(1->IMIEA)
169      ;
170      ;***** INITIALZE ITU ch2 *****
171      010036  7900000A      171          MOV.W      #H'000A, R0
172      01003A  6B80FF7E      172          MOV.W      R0, @GRA2H      ;Set non overlap time
173      01003E  790005DC      173          MOV.W      #D'1500, R0
174      010042  6B80FF80      174          MOV.W      R0, @GRB2H      ;Set timer counter
175      010046  F840          175          MOV.B      #H, 40, R0L
176      010048  3878          176          MOV.B      R0L, @TCR2      ;Init. TCR2(1->CCLR1)
177      01004A  F801          177          MOV.B      #H'01, R0L
178      01004C  387A          178          MOV.B      R0L, @TIER2      ;Init. TIER2(1->IMIEA)
179      ;
180      ;***** INITIALZE ITU ch3 *****

```

2. 基礎編

```

181 01004E 7900000A 181          MOV.W    #H'000A,R0
182 010052 6B80FF88 182          MOV.W    R0,@GRA3H    ;Set non overlap time
183 010056 790007D0 183          MOV.W    #D'2000,R0
184 01005A 6B80FFBA 184          MOV.W    R0,@GRB3H    ;Set timer counter
185 01005E F840 185          MOV.B    #H'40,R0L
186 010060 3882 186          MOV.B    R0L,@TCR3    ;Init. TCR3(1->CCLR1)
187 010062 F801 187          MOV.B    #H'01,R0L
188 010064 3884 188          MOV.B    R0L,@TIER3   ;Init. TIER3(1->IMIEA)
189          189          ;
190          ;***** INITIALZE TPC *****
191 010066 6A28000101AC 191         MOV.B    @NDAT_TAB0,R0L
192 01006C 6A20000101B1 192         MOV.B    @NDAT_TAB1,R0H
193 010072 1480 193          OR.B     R0L,R0H
194 010074 30D3 194          MOV.B    R0H,@PADR    ;Set first output data
                               ;(Gr1,Gr0)
195 010076 6A28000101B6 195         MOV.B    @NDAT_TAB2,R0L
196 01007C 6A20000101BB 196         MOV.B    @NDAT_TAB3,R0H
197 010082 1480 197          OR.B     R0L,R0H
198 010084 30D6 198          MOV.B    R0H,@PBDR    ;Set first outupt data
                               ;(Gr3,Gr2)
199 010086 F8FF 199          MOV.B    #H'FF,R0L
200 010088 38D1 200          MOV.B    R0L,@PADDR
201 0100BA 38D4 201          MOV.B    R0L,@PBDDR
202 0100BC 38A3 202          MOV.B    R0L,@NDERA
203 01008E 38A2 203          MOV.B    R0L,@NDERB   ;Set output TPC
204 010090 38A0 204          MOV.B    R0L,@TPMR    ;Set output non overlap
205 010092 F8E4 205          MOV.B    #H'E4,R0L
206 010094 38A1 206          MOV.B    R0L,@TPCR    ;Init. TPCR
207 010096 6A28000101AD 207         MOV.B    @NDAT_TAB0+1,R0L
208 01009C 38A7 208          MOV.B    R0L,@NDRA0   ;Set second output data(Gr0)
209 01009E 6A28000101B2 209         MOV.B    @NDAT_TAB1+1,R0L
210 0100A4 38A5 210          MOV.B    R0L,@NDRA1   ;Set second output data (Gr1)
211 0100A6 6A28000101B7 211         MOV.B    @NDAT_TAB2+1,R0L
212 0100AC 38A6 212          MOV.B    R0L,@NDRB2   ;Set second output data(Gr2)
213 0100AE 6A28000101BC 213         MOV.B    @NDAT_TAB3+1,R0L
214 0100B4 38A4 214          MOV.B    R0L,@NDRB3   ;Set second output data(Gr3)
215          215          ;
216 0100B6 F802 216          MOV.B    #H'02,R0L
217 0100B8 3800 217          MOV.B    R0L,@INC_ADR0 ;Init. INC_ADR0
218 0100BA 3801 218          MOV.B    R0L,@INC_ADR1 ;Init. INC_ADR1
219 0100BC 3802 219          MOV.B    R0L,@INC_ADR2 ;Init. INC_ADR2
220 0100BE 3803 220          MOV.B    R0L,@INC_ADR3 ;Init. INC_ADR3
221 0100C0 F8EF 221          MOV.B    #H'EF,R0L
222 0100C2 3860 222          MOV.B    R0L,@TSTR     ;Start ITU ch0-3
223 0100C4 0700 223          LDC.B    #0,CCR       ;Enable interrupt
224          224          ;
225 0100C6 40FE 225          TPC16MN99  BRA    TPC16MN99

```



```

226          226 ;
227          227 ;*****
228          228 ;*
229          229 ;*   NAME : SINDAT0 (Set next data to TPC Gr0) *
230          230 ;*
231          231 ;*****
232          232 ;*
233          233 ;*   ENTRY  : INC_ADR0 (Address of output data) *
234          234 ;*   RETURNS : NOTHING
235          235 ;*
236          236 ;*****
237          237 ;
238 0100C8 01006DF0 238 SETDAT0:  PUSH.L  ER0          ;Escape register
239 0100CC 01006DF1 239          PUSH.L  ER1
240 0100D0 F800     240          MOV.B   #0,R0L      ;TSR clr
241 0100D2 2867     241          MOV.B   #TSR0,R0H
242 0100D4 3867     242          MOV.B   R0L,@TSR0
243 0100D6 7A0000000000 243          MOV.L   #D'0,ER0    ;Register clr
244 0100DC 2800     244          MOV.B   @INC_ADR0,R0L
245 0100DE 7A01000101AC 245          MOV.L   #NDAT_TAB0,ER1
246 0100E4 0A81     246          ADD.L   ER0,ER1
247 0100E6 6810     247          MOV.B   @ER1,R0H
248 0100E8 30A7     248          MOV.B   R0H,@NDRA0  ;Set NDRA0(Gr0)
249 0100EA 8801     249          ADD.B   #D'1,R0L    ;NDAT_TAB0 Address
                                   ;incliment
250 0100EC 3800     250          MOV.B   R0L,@INC_ADR0 ;INC_ADR0+1 ->INCADR0
251 0100EE A805     251          CMP.B   #D'5,R0L    ;All output end?
252 0100F0 58600004 252          BNE    SETDAT099   ;No
253 0100F4 7F607200 253          BCLR.B  #STR0,@TSTR  ;ITU ch0 Stop
254 0100F8 01006D71 254 SETDAT099: POP.L   ER1
255 0100FC 01006D70 255          POP.L   ER0
256 010100 5670     256          RTE
257          257 ;
258          258 ;*****
259          259 ;*
260          260 ;*   NAME : SINDAT1 (Set next data to TPC Gr1) *
261          261 ;*
262          262 ;*****
263          263 ;*
264          264 ;*   ENTRY  : INC_ADR1 (Address of outupt data) *
265          265 ;*   RETURNS : NOTHING
266          266 ;*
267          267 ;*****
268          268 ;
269 010102 01006DF0 269 SETDAT1:  PUSH.L  ER0          ;Escape register
270 010106 01006DF1 270          PUSH.L  ER1
271 01010A F800     271          MOV.B   #0,R0L      ;TSR clr

```

2. 基礎編

```
272 01010C 2071      272          MOV.B    @TSR1,R0H
273 01010E 3871      273          MOV.B    R0L,@TSR1
274 010110 7A0000000000 274          MOV.L    #D'0,ER0
275 010116 2801      275          MOV.B    @INC_ADR1,R0L
276 010118 7A01000101B1 276          MOV.L    #NDAT_TAB1,ER1
277 01011E 0A81      277          ADD.L    ER0,ER1
278 010120 6810      278          MOV.B    @ER1,R0H
279 010122 30A5      279          MOV.B    R0H,@NDRA1    ;Set NDRA1(Gr1)
280 010124 8801      280          ADD.B    #D'1,R0L    ;NDAT_TAB1 Address
                                ;incliment

281 010126 3801      281          MOV.B    R0L,@INC-ADR1 ;INC_ADR1+1 -> INC_ADR1
282 010128 A805      282          CMP.B    #D'5,R0L    ;All output end?
283 01012A 58600004  283          BNE     SETDAT199    ;No
284 01012E 7F607210  284          BCLR.B  #STR1,@TSTR  ;ITU ch1 Stop
285 010132 01006D71  285 SETDAT199: POP.L    ER1
286 010136 01006D70  286          POP.L    ER0
287 01013A 5670      287          RET
288
288 ;
289 ;*****
290 ;*
291 ;*   NAME : SINDAT2 (Set next data to TPC Gr2) *
292 ;*
293 ;*****
294 ;*
295 ;*   ENTRY : INC_ADR2 (Address Of output data) *
296 ;*   RETURNS : NOTHING *
297 ;*
298 ;*****
299 ;
300 01013C 01006DF0  300 SETDAT2:  PUSH.L   ER0          ;Escape register
301 010140 01006DF1  301          PUSH.L   ER1
302 010144 F800      302          MOV.B    #0,R0L      ;TSR clr
303 010146 207B      303          MOV.B    @TSR2,R0H
304 010148 387B      304          MOV.B    R0L,@TSR2
305 01014A 7A0000000000 305          MOV.L    #D'0,ER0
306 010150 2802      306          MOV.B    @INC_ADR2,R0L
307 010152 7A01000101B6 307          MOV.L    #NDAT_TAB2,ER1
308 010158 0A81      308          ADD.L    ER0,ER1
309 01015A 6810      309          MOV.B    @ER1,R0H
310 01015C 30A6      310          MOV.B    R0H,@NDRB2   ;Set NDRB2(Gr2)
311 01015E 8801      311          ADD.B    #D'1,R0L    ;NDAT_TAB2 Address incliment
312 010160 3802      312          MOV.B    R0L,@INC_ADR2;INC_ADR2+1 -> INC_ADR2
313 010162 A805      313          CMP.B    #D'5,R0L    ;All output end?
314 010164 58600004  314          BNE     SETDAT299    ;No
315 010168 7F607220  315          BCLR.B  #STR2,@TSTR  ;ITU ch2 Stop
316 01016C 01006D71  316 SETDAT299: POP.L    ER1
317 010170 01006D70  317          POP.L    ER0
```

```

318 010174 5670          318          RTE
319                      319          ;
320                      320          ;*****
321                      321          ;*
322                      322          ;*      NAME : SINDAT3 (Set next data to TPC Gr3) *
323                      323          ;*
324                      324          ;*****
325                      325          ;*
326                      326          ;*      ENTRY : INC_ADR3 (Address of output data) *
327                      327          ;*      RETURNS : NOTHING
328                      328          ;*
329                      329          ;*****
330                      330          ;
331 010176 01006DF0      331 SETDAT3:  PUSH.L  ER0          ;Escape register
332 01017A 01006DF1      332          PUSH.L  ER1
333 01017E F800          333          MOV.B   #0,R0L      ;TSR clr
334 010180 2085          334          MOV.B   @TSR3,R0H
335 010182 3885          335          MOV.B   R0L,@TSR3
336 010184 7A0000000000 336          MOV.L   #D'0,ER0
337 01018A 2803          337          MOV.B   @INC_ADR3,R0L
338 01018C 7A01000101BB 338          MOV.L   #NDAT_TAB3,ER1
339 010192 0A81          339          ADD.L   ER0,ER1
340 010194 6810          340          MOV.B   @ER1,R0H
341 010196 30A4          341          MOV.B   R0H,@NDRB3   ;Set NDRB3(Gr3)
342 010198 8801          342          ADD.B   #D'1,R0L   ;NDAT_TAB3 Address
                                     ;incliment
343 01019A 3803          343          MOV.B   R0L,@INC_ADR3 ;INCADR3+1 -> INC_ADR3
344 01019C A805          344          CMP.B   #D'5,R0L   ;All output end?
345 01019E 58600004      345          BNE    SETDAT399   ;No
346 0101A2 7F607230      346          BCLR.B #STR3,@TSTR  ;ITU ch3 Stop
347 0101A6              347 SETDAT399: POP.L   ER1
348 0101A6 01006D70      348          POP.L   ER0
349 0101AA 5670          349          RTE
350                      350          ;
351                      351          ;*****
352                      352          ;
353                      353          ;*      DATA TABLE (OUTPUT PATTERN)
354                      354          ;*
355                      355          ;*****
356 0101AC 0908070605      356 NDAT_TAB0  .DATA.B  H'09,H'08,H'07,H'06,H'05
357 0101B1 8070605040      357 NDAT_TAB1  .DATA.B  H'80,H'70,H'60,H'50,H'40
358 0101B6 0706050403      358 NDAT_TAB2  .DATA.B  H'07,H'06,H'05,H'04,H'03
359 0101BB 6050403020      359 NDAT_TAB3  .DATA.B  H'60,H'50,H'40,H'30,H'20
360                      360          ;
361                      361          .END
****TOTAL ERRORS      0
****TOTAL WARNINGS    0

```

2.11 調歩同期式 SCI

MCU : H8/3003

使用機能 : シリアルコミュニケーションインタフェース (SCI) (調歩同期式モード)

2.11.1 仕様

- 図 2.58 に示すようにコンソールと、3 バイトのデータを調歩同期式で送受信します。
- 転送フォーマットは 9600BPS、8 ビットデータ、1 ストップビットおよびノンパリティです。
- RTS、CTS による通信制御を行います。

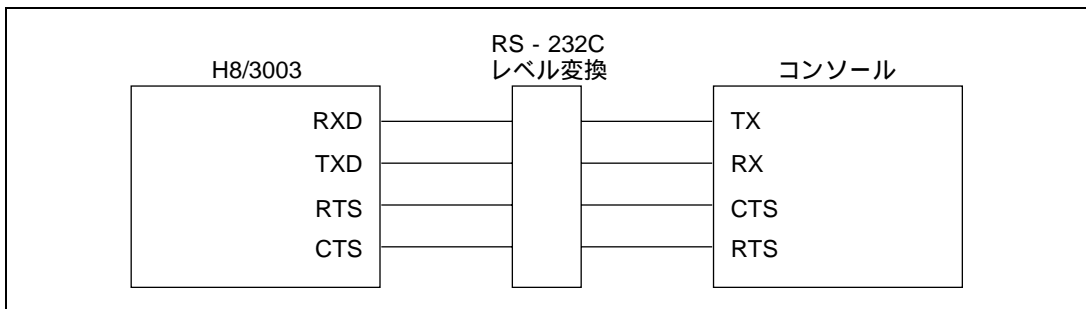


図 2.58 H8/3003 による調歩同期式 SCI インタフェースブロック図

2.11.2 使用機能説明

本タスク例ではデータの送受信を SCI/ch0 を使用して行います。また、通信制御端子 (RTS、CTS) は PORT1 を使用します。

図 2.59 に、39 本タスク例で使用する SCI の送信ブロック図を示します。本タスクでは図 2.59 に示す SCI の以下の機能を使用してコンソールヘータの送信を行います。

- キャラクタ単位で同期をとる調歩同期方式でデータの通信を行う機能。(調歩同期式モード)
- 送信完了時に割り込みを発生する機能 (TEI 割り込み)

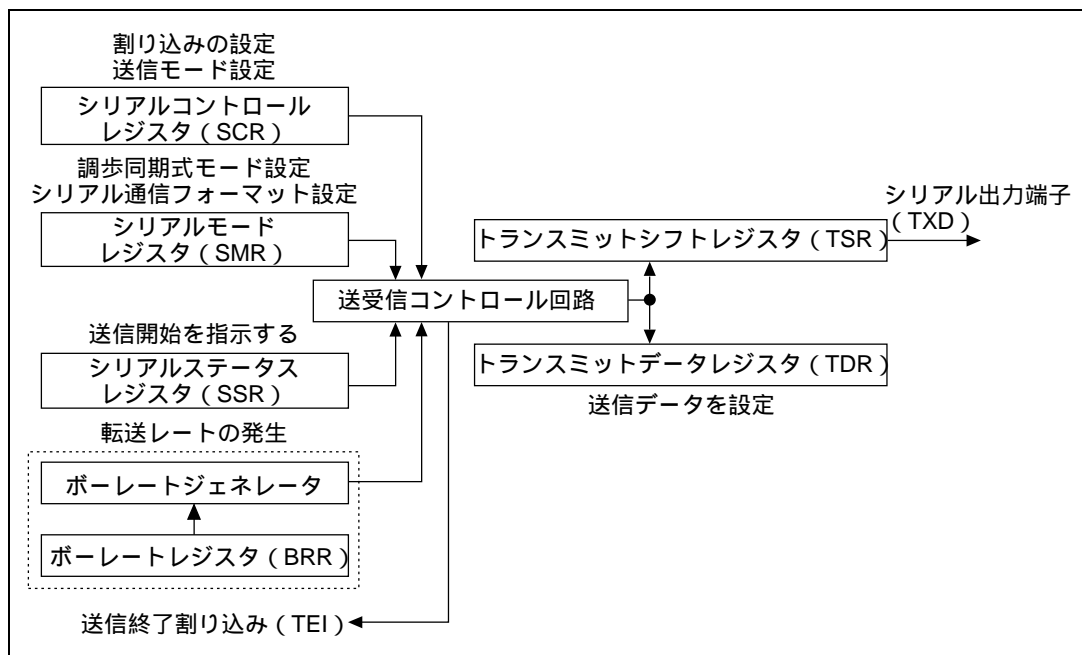


図 2.59 SCI 送信ブロック図

2. 基礎編

図 2.60 に本タスク例で使用する SCI の受信ブロック図を示します。本タスクでは図 2.60 に示す SCI の以下の機能を使用してコンソールからのデータを受信します。

- キャラクタ単位で同期をとる調歩同期方式でデータの通信を行う機能。(調歩同期式モード)
- 受信完了時に割り込みを発生する機能 (RXI 割り込み)

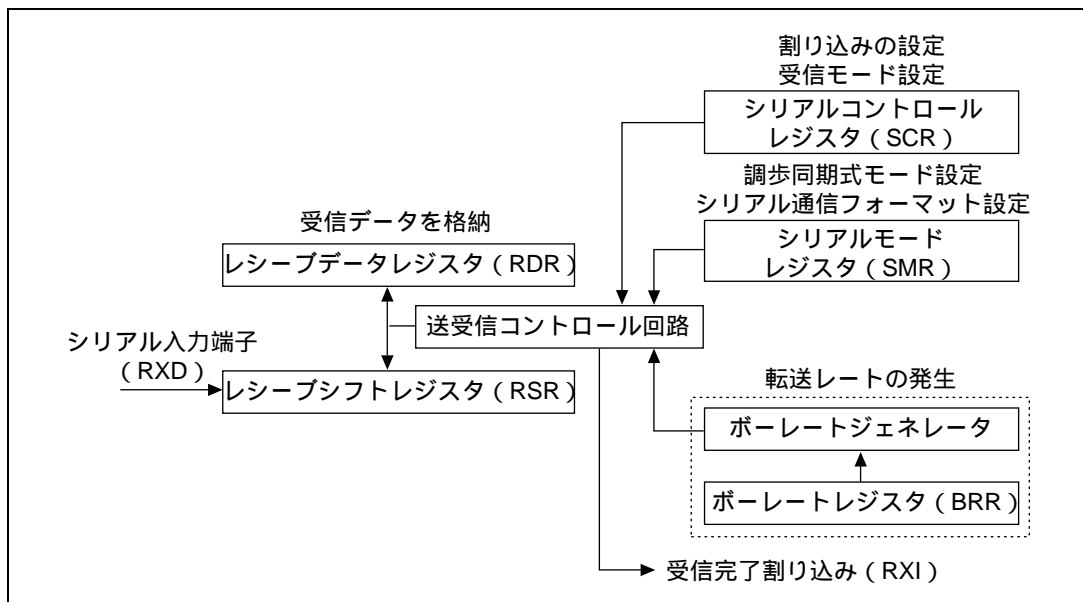


図 2.60 SCI 受信ブロック図

表 2.52 に本タスク例の機能割り付けを示します。表 2.52 に示すように SCI の機能を割り付け、コンソールとインタフェースを行います。

表 2.52 SCI 機能割り付け

SCI 機能	機能
RXD	コンソールからデータを受信する
TXD	コンソールへデータを送信する
SMR	SCI を調歩同期式モードに設定し、転送フォーマットを設定する
SCR	送受信割り込みを許可し、SCI を送受信モードに設定する
SSR	TDRE にて送信開始を指示する
RDR	コンソールから受信したデータを格納する
TDR	コンソールへ送信するデータを設定する
BRR	転送レートを設定する

2.11.3 動作説明

図 2.61 に本タスクの動作原理を示します。図 2.61 に示すタイミングでハードウェア処理およびソフトウェア処理を行い、コンソールとのインタフェースを実行します。

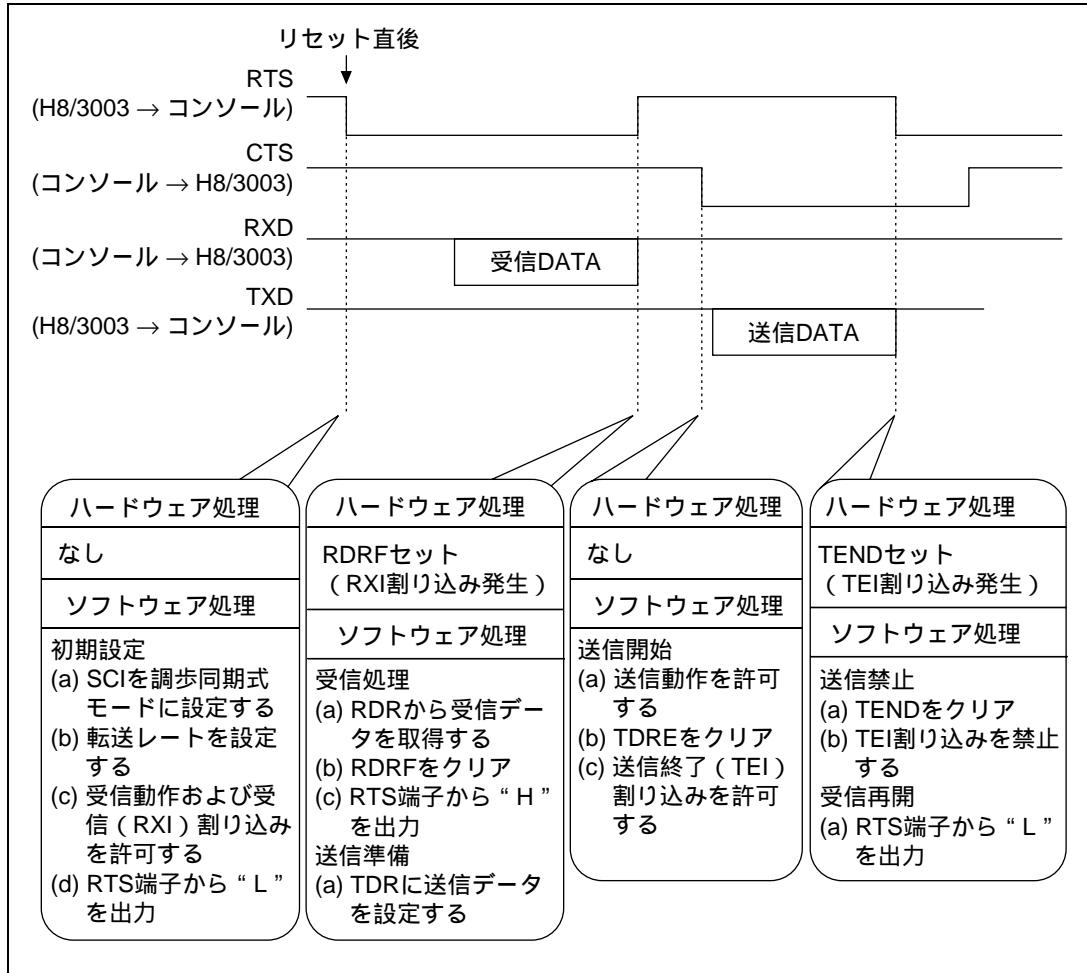


図 2.61 調歩同期式 SCI 動作原理

2.11.4 ソフトウェア説明

表 2.53 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	ASCMN	SCI の初期設定、および送受信の管理を行う。
データ受信完了	ASCRX	RXI 割り込みで起動し、データの受信を行う。
データ送信完了	ASCTE	TEI 割り込みで起動し、送信が完了したことを通知する。

表 2.54 引き数の説明

ラベル名、レジスタ名	機能	データ長	使用モジュール名	入出力
R6L	コンソールから受信したデータを設定する。	1 バイト	データ受信完了	出力
			メインルーチン	入力
RXENDF	受信完了を示すフラグ。 1: 受信完了 0: 受信動作中	1 ビット	データ受信完了	出力
			メインルーチン	入力
TXENDF	送信完了を示すフラグ。 1: 送信完了 0: 送信動作中	1 ビット	データ送信完了	出力
			メインルーチン	入力

表 2.55 使用内部レジスタ説明

レジスタ名	機能	使用モジュール名
SMR	SCI のモード (調歩同期式)、転送フォーマットおよびボーレートジェネレータへのクロック選択 (クロック入力) を設定する。	メインルーチン
SCR	割り込み (RXI、TEI) を許可し、SCI の送受信の許可を設定する。	メインルーチン
SSR	TDRE (b7) をクリアすることにより、送信開始を指示する。	メインルーチン
RDR	コンソールから受信したデータを設定する。	データ受信完了
TDR	コンソールへ送信するデータを設定する。	メインルーチン
BRR	転送レートを設定する。	メインルーチン
P9DDR	ポート 9 の入出力を設定する。	メインルーチン
P9DR	PRQ 端子および CTS 端子の操作を行う。	メインルーチン

表 2.56 汎用レジスタ説明

モジュール名	レジスタ名	機能
メインルーチン	R0L	SCI、引き数 RAM を設定するときのワークとして使用する。
	R1L	送受信データ数のカウンタとして使用する。
	R6L	受信データを受け取る引き数として使用する。
データ受信完了	R6L	受信データを渡す引き数として使用する。

- 使用 RAM 説明
本タスク例では引き数以外の RAM は使用していません。

2.11.5 フローチャート

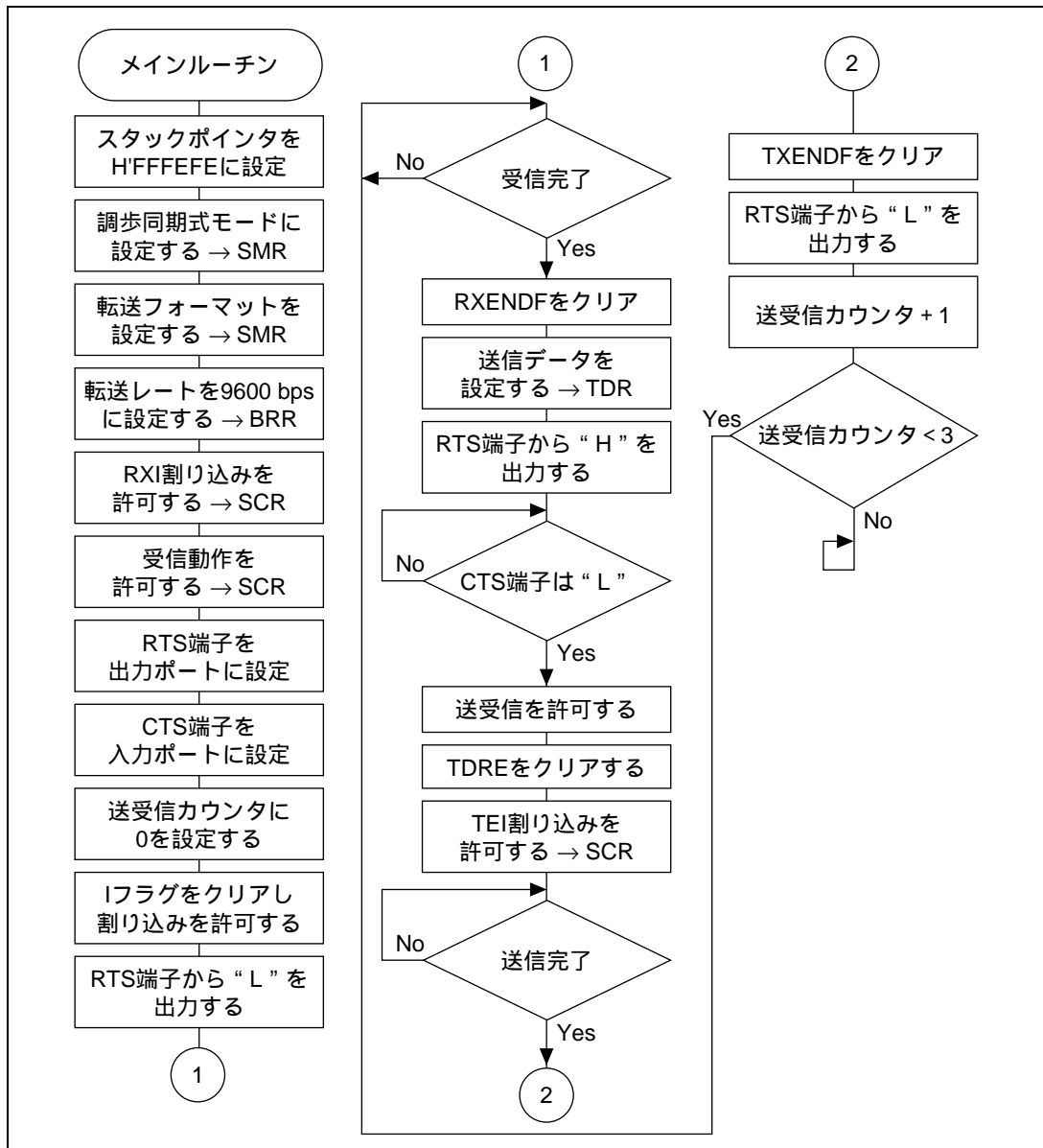


図 2.62 メインルーチン

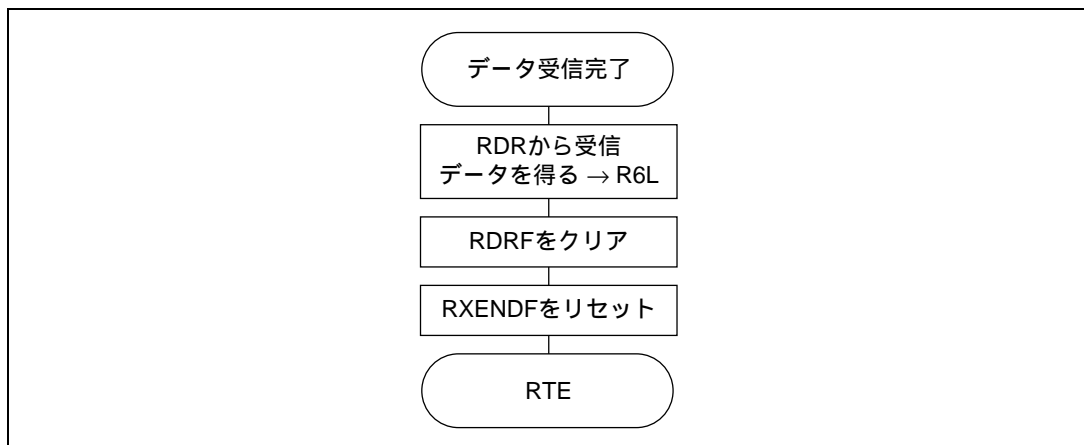


図 2.63 データ受信完了



図 2.64 データ送信完了

2.11.6 プログラムリスト

```

1          1 ;*****
2          2 ;*
3          3 ;*          VECTOR ADDRESS
4          4 ;*
5          5 ;*****
6          6
7          7          .CPU      300HA
8 000000   8          .SECTION  VECT, CODE, LOCATE=H'000000
9          9 ;
10 000000 00010000 10 RES          .DATA.L  ASCMN
11          11 ;
12 00001C   12          .ORG      H'00001C
13 00001C 00010000 13 NMI          .DATA.L  ASCMN
14 000020 00010000 14 TRAP0       .DATA.L  ASCMN
15 000024 00010000 15 TRAP1       .DATA.L  ASCMN
16 000028 00010000 16 TRAP2       .DATA.L  ASCMN
17 00002C 00010000 17 TRAP3       .DATA.L  ASCMN
18          18 ;
19 000030   19          .ORG      H'000030
20 000030 00010000 20 IRQ0        .DATA.L  ASCMN
21 000034 00010000 21 IRQ1        .DATA.L  ASCMN
22 000038 00010000 22 IRQ2        .DATA.L  ASCMN
23 00003C 00010000 23 IRQ3        .DATA.L  ASCMN
24 000040 00010000 24 IRQ4        .DATA.L  ASCMN
25 000044 00010000 25 IRQ5        .DATA.L  ASCMN
26 000048 00010000 26 IRQ6        .DATA.L  ASCMN
27 00004C 00010000 27 IRQ7        .DATA.L  ASCMN
28 000050 00010000 28 WOVI        .DATA.L  ASCMN
29 000054 00010000 29 CMI          .DATA.L  ASCMN
30          30 ;
31 000060   31          .ORG      H'000060
32 000060 00010000 32 IMIA0       .DATA.L  ASCMN
33 000064 00010000 33 IMIB0       .DATA.L  ASCMN
34 000068 00010000 34 OVI0        .DATA.L  ASCMN
35          35 ;
36 000070   36          .ORG      H'000070
37 000070 00010000 37 IMIA1       .DATA.L  ASCMN
38 000074 00010000 38 IMIB1       .DATA.L  ASCMN
39 000078 00010000 39 OVI1        .DATA.L  ASCMN
40          40 ;
41 000080   41          .ORG      H'000080
42 000080 00010000 42 IMIA2       .DATA.L  ASCMN
43 000084 00010000 43 IMIB2       .DATA.L  ASCMN
44 000088 00010000 44 OVI2        .DATA.L  ASCMN
45          45 ;

```

2. 基礎編

```
46 000090          46          .ORG      H'000090
47 000090 00010000 47 IMIA3    .DATA.L  ASCMN
48 000094 00010000 48 IMIB3    .DATA.L  ASCMN
49 000098 00010000 49 OVI3     .DATA.L  ASCMN
50
51 0000A0          51          .ORG      H'0000A0
52 0000A0 00010000 52 IMIA4    .DATA.L  ASCMN
53 0000A4 00010000 53 IMIB4    .DATA.L  ASCMN
54 0000A8 00010000 54 OVI4     .DATA.L  ASCMN
55
56 0000B0          56          .ORG      H'0000B0
57 0000B0 00010000 57 DEND0A   .DATA.L  ASCMN
58 0000B4 00010000 58 DEND0B   .DATA.L  ASCMN
59 0000B8 00010000 59 DEND1A   .DATA.L  ASCMN
60 0000BC 00010000 60 DEND1B   .DATA.L  ASCMN
61 0000C0 00010000 61 DEND2A   .DATA.L  ASCMN
62 0000C4 00010000 62 DEND2B   .DATA.L  ASCMN
63 0000C8 00010000 63 DEND3A   .DATA.L  ASCMN
64 0000CC 00010000 64 DEND3B   .DATA.L  ASCMN
65 0000D0 00010000 65 ERI0     .DATA.L  ASCMN
66 0000D4 0001005A 66 RXI0     .DATA.L  ASCIRX
67 0000D8 00010000 67 TXI0     .DATA.L  ASCMN
68 0000DC 00010066 68 TEI0     .DATA.L  ASCITE
69 0000E0 00010000 69 ERI1     .DATA.L  ASCMN
70 0000E4 00010000 70 RXI1     .DATA.L  ASCMN
71 0000E8 00010000 71 TXI1     .DATA.L  ASCMN
72 0000EC 00010000 72 TEI1     .DATA.L  ASCMN
73 0000F0 00010000 73 ADI      .DATA.L  ASCMN
74
75                75 ;*****
76                76 ;*
77                77 ;*      RAM ALLOCATION
78                78 ;*
79                79 ;*****
80
81 FFFF00          81          .SECTION  RAM,DATA,LOCATE=H'FFFF00
82
83 FFFF00 00000001 83 FLAG     .RES.B   1          ;Flag area
84      00000000 84 RXENDF   .EQU     0          ;Receive end flag
85      00000001 85 TXENDF   .EQU     1          ;Transmit end flag
86
87                87 ;*****
88                88 ;*
89                89 ;*      SYMBOL DEFINITIONS
90                90 ;*
91                91 ;*****
92                92 ;
```

```

93          93 ;***** SCI ch0 *****
94          00FFFFB0 94 SMR0      .EQU    H'FFFFB0      ;Serial mode register
95          95 ;
96          00FFFFB1 96 BRR0      .EQU    H'FFFFB1      ;Bit rate register
97          97 ;
98          00FFFFB2 98 SCRO      .EQU    H'FFFFB2      ;Sirial control register
99          00000006 99 RIE      .EQU    6              ;Receive interrupt enable
100         00000005 100 TE      .EQU    5              ;Transmit enable
101         00000004 101 RE      .EQU    4              ;Receive enable
102         00000003 102 MPIE     .EQU    3              ;Multi processor interrupt
                                     ;enable
103         00000002 103 TEIE     .EQU    2              ;Transmit end interrupt
                                     ;enable

104         104 ;
105         00FFFFB3 105 TDR0      .EQU    H'FFFFB3      ;Transmit data register
106         106 ;
107         00FFFFB4 107 SSR0      .EQU    H'FFFFB4      ;Serial status register
108         00000007 108 TDRE     .EQU    7              ;Transmit data register empty
109         00000006 109 RDRF     .EQU    6              ;Receive data register full
110         110 ;
111         00FFFFB5 111 RDR0      .EQU    H'FFFFB5      ;Receive data register
112         112 ;***** PORT 9 *****
113         00FFFFD0 113 P9DDR     .EQU    H'FFFFD0      ;Port 9 data direction
                                     ;register

114         114 ;
115         00FFFFD2 115 P9DR      .EQU    H'FFFFD2      ;Port 9 data register
116         00000004 116 RTS      .EQU    4              ;RTS bit
117         00000005 117 CTS      .EQU    5              ;CTS bit
118         118 ;
119         119 ;*****
120         120 ;*
121         121 ;*          MAIN PROGRAM : ASCMN          *
122         122 ;*
123         123 ;*****
124         124 ;
125         010000          125          .SECTION  PROG,CODE,LOCATE=H'010000
126         126 ;
127         00010000 127 ASCMN:   .EQU    $
128         010000 7A0700FFFEFE 128          MOV.L    #H'FFFEFE,SP      ;Set stack pointer
129         010006 F800          129          MOV.B    #B'00000000,R0L ;Asyncro mode,8bit,np,
                                     ;stop1

130         010008 38B0          130          MOV.B    R0L,@SMR0      ;
131         01000A F833          131          MOV.B    #51,R0L      ;CKS0,1=0,N=51
132         01000C 38B1          132          MOV.B    R0L,@BRR0     ;Then 9600bps
133         01000E 7FB27060 133          BSET.B   #RIE,@SCRO     ;Enable RXI interrupt
134         010012 7FB27040 134          BSET.B   #RE,@SCRO     ;Enable receive
135         010016 P8D0          135          MOV.B    #H'D0,R0L      ;CTS is input port

```

2. 基礎編

```

136 010018 38D0      136          MOV.B    R0L,@P9DDR      ;RTS is output port
137                137 ;
138 01001A F900      138          MOV.B    #H'0,R1L       ;Set TXRXCNT=0
139                139 ;
140 01001C 067F      140          ANDC    #B'01111111,CCR ;Clear I-FLAG
141 01001E 7FD27240  141          BCLR.B  #RTS,@P9DR      ;RTS is 'Low'
142 010022          142  ASCMN01:
143 010022 7E007300  143          BTST.B  #RXENDF,@FLAG   ;Receive complete ?
144 010026 47FA      144          BEQ     ASCMN01        ;No
145                145 ;
146 010028 7F007200  146          BCLR.B  #RXENDF,@FLAG   ;Clear RXENDF
147 01002C 3EB3      147          MOV.B    RGL,@TDR0      ;Set transmit data
                                ;(receive data)
148 01002E 7FD27040  148          BSET.B  #RTS.@P9DR      ;RTS is 'High'
149                149 ;
150 010032          150  ASCMN02:
151 010032 7ED27350  151          BTST.B  #CTS,@P9DR      ;CTS is 'High' ?
152 010036 47FA      152          BEQ     ASCMN02        ;No
153                153 ;
154 010038 7FB27050  154          BSET.B  #TE,@SCR0       ;Enable transmit
155 01003C 7FB47270  155          BCLR.B  #TDRE,@SSR0     ;Start transmit
156 010040 7FB27020  156          BSET.B  @TEIE,@SCR0     ;Enable TEI interrupt
157                157 ;
158 010044          158  ASCMN03:
159 010044 7E007310  159          BTST.B  #TXENDF,@FLAG   ;Transmit complete ?
160 010048 47FA      160          BEQ     ASCMN03        ;No
161                161 ;
162 01004A 7F007210  162          BCLR.B  #TXENDF,@FLAG   ;Clear TXENDF
163 01004E 7FD27240  163          BCLR.B  #RTS,@P9DR      ;RTS is 'Low'
164                164 ;
165 010052 0A09      165          INC.B    R1L             ;TXRXCNT + 1
166 010054 A903      166          CMP.B    #3,R1L         ;TXRXCNT < 3
167 010056 45CA      167          BCS     ASCMN01        ;Yes
168                168 ;
169 010058          169  ASCMN99:
170 010058 40FE      170          BRA     ASCMN99
171                171 ;
172                172 ;*****
173                173 ;*
174                174 ;*          NAME : ASCIRX(SCI receive interrupt) *
175                175 ;*
176                176 ;*****
177                177 ;*
178                178 ;*          ENTRY : NOTHING
179                179 ;*          RETURNS : RXENDF (Receive end flag) *
180                180 ;*
181                181 ;*****

```

```
182                                     182 ;
183         0001005A                   183 ASCIRX:      .EQU      $
184 01005A 2EB5                        184             MOV.B     @PDR0,R6L      ;Get receive data to R6L
185 01005C 7FB47260                    185             BCLR.B     #RDRF,@SSR0      ;Clear RDRF
186 010060 7F007000                    186             BSET.B     #RXENDF,@FLAG ;Set RXENDF
187 010064 5670                        187             RTE
188                                     188 ;
189                                     189 ;*****
190                                     190 ;*
191                                     191 ;*          NAME : ASCITE(SCI transmit interrupt) *
192                                     192 ;*
193                                     193 ;*****
194                                     194 ;*
195                                     195 ;*          ENTRY : NOTHING
196                                     196 ;*          RETURNS : TXENDF (Transmit end flag) *
197                                     197 ;*
198                                     198 ;*****
199                                     199 ;
200         00010066                   200 ASCITE:      .EQU      $
201 010066 7F007010                    201             BSET.B     #TXENDF,@FLAG ;Set TXENDF
202 01006A 7F827250                    202             BCLR.B     #TE,@SCR0      ;Disable transmit
203 01006E 7F827220                    203             BCLR.B     #TEIE,@SCR0    ;Disable TEI interrupt
204 010072 5670                        204             RTE
205                                     205 ;
206                                     206             .END
*****TOTAL ERRORS 0
*****TOTAL WARNINGS 0
```

2.12 同時送受信動作

MCU : H8/3003

使用機器 : SCI (同時送受信)

2.12.1 仕様

- 図 2.65 に示すように、H8/3003 と H4019 との間で 1 バイトのデータの送受信を行います。
- データの送受信はクロック同期式フォーマットで行います。クロックはマスタ側の H8/3003 が、スレーブ側の H4019 に供給します。
- H8/3003 はデータの送信と同時に H4019 からのデータを受信します。

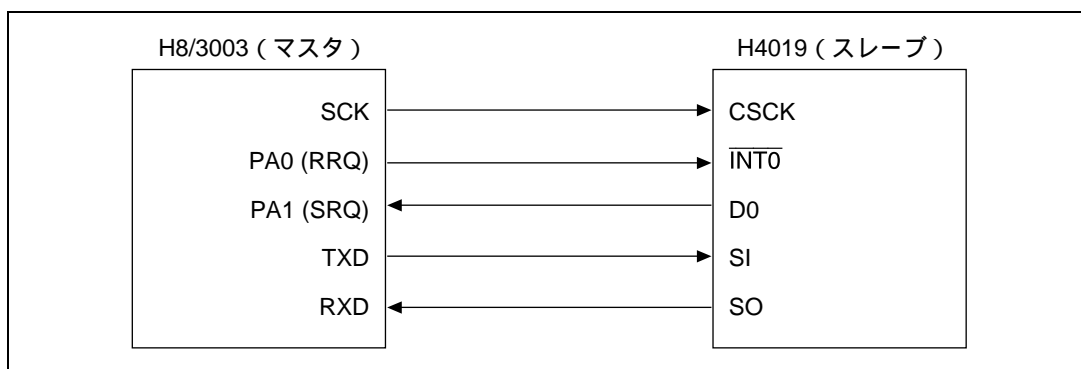


図 2.65 H8/3003 によるクロック同期式 SCI インタフェースブロック図

2.12.2 使用機能説明

本タスク例ではデータの送受信を SCI を使用して行います。また、通信制御端子 (RRQ、SRQ) は PORT1 を使用します。

図 2.66 に本タスク例で使用する SCI のブロック図を示します。図 2.66 に示す SCI の以下の機能を使用して送受信動作を同時に行います。

- クロックに同期してシリアルデータの通信を行う機能。(クロック同期式モード)
- 送信と受信を同時に行う機能。(同時送受信動作)
- 受信完了時に割り込みを発生する機能 (RXI 割り込み)

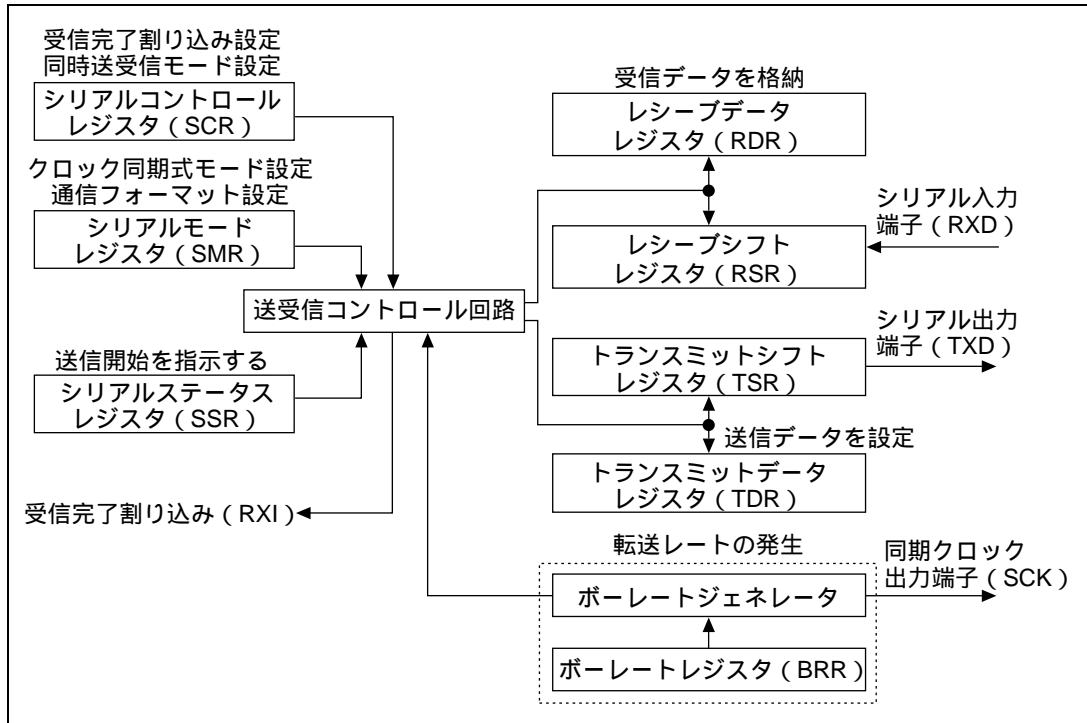


図 2.66 SCI ブロック図

表 2.57 に本タスク例の機能割り付けを示します。表 2.57 に示すように SCI の機能を割り付け、H4019 とデータの同時送受信を行います。

表 2.57 SCI 機能割り付け

SCI 機能	機能
SCK	同期クロックを送信する
RXD	H4019 からデータを受信する
TXD	H4019 へデータを送信する
SMR	SCI をクロック同期式モードに設定する
SCR	受信割り込みを許可し、SCI を送受信モードに設定する
SSR	TDRE ビットにより送信開始を指示する
RDR	H4019 から受信したデータを設定する
TDR	H4019 へ送信するデータを設定する
BRR	転送レートを設定する
PADDR	ポート A の入出力を設定する
PADR	PRQ の送信および SRQ の受信を行う

2.12.3 動作説明

図 2.67 に本タスクの動作原理を示します。図 2.67 に示すタイミングでハードウェア処理およびソフトウェア処理を行い、H4019 とのインタフェースを行います。

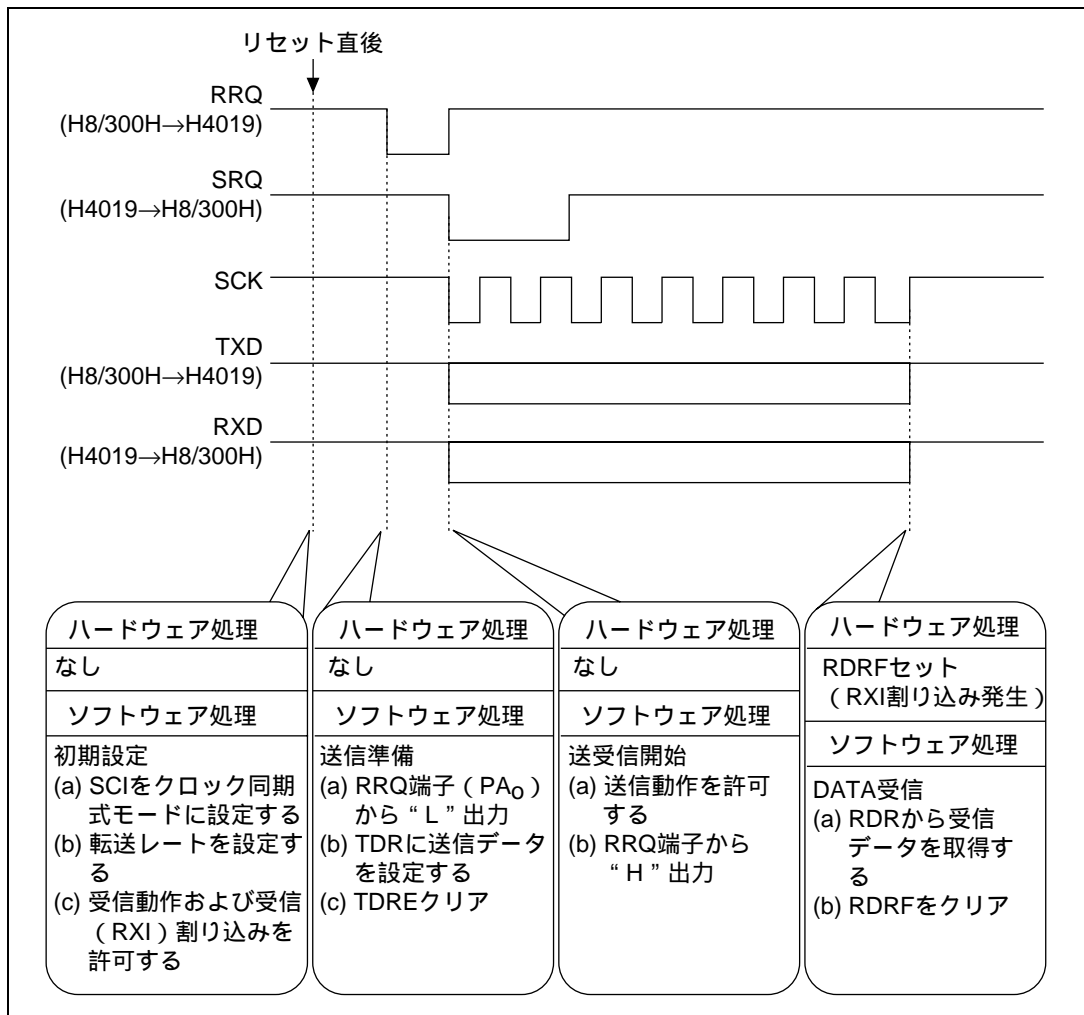


図 2.67 同時送受信動作動作原理

2.12.4 ソフトウェア説明

表 2.58 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	CSCMN	SCI の初期設定、および送受信の管理を行う。
データ受信完了	CSCIRX	RXI 割り込みで起動し、データの受信を行う。

表 2.59 引数の説明

ラベル名、レジスタ名	機能	データ長	使用モジュール名	入出力
R6L	コンソールから受信したデータを設定する。	1 バイト	データ受信完了	出力
			メインルーチン	入力
RXENDF	受信完了を示すフラグ。 1: 受信完了 0: 受信動作中	1 ビット	データ受信完了	出力
			メインルーチン	入力

表 2.60 使用内部レジスタ説明

レジスタ名	機能	使用モジュール名
SMR	SCI のモード(クロック同期式)、転送フォーマットおよびポートジェネレータへのクロック選択(クロック入力)を設定する。	メインルーチン
SCR	割り込み(RXI)を許可し、SCI の送受信の許可を設定する。	メインルーチン
SSR	TDRE (b7) をクリアすることにより、送信開始を指示する。	メインルーチン
RDR	H4019 から受信したデータを設定する。	データ受信完了
TDR	H4019 へ送信するデータを設定する。	メインルーチン
BRR	転送レートを設定する。	メインルーチン
PADDR	ポート A の入出力を設定する。	メインルーチン
PADR	PRQ 端子および CTS 端子の操作を行う。	メインルーチン

表 2.61 汎用レジスタ説明

モジュール名	レジスタ名	機能
メインルーチン	R0L	SCI、引数 RAM を設定するときのワークとして使用する。
	R6L	受信データを受け取る引数として使用する。
データ受信完了	R6L	受信データを渡す引数として使用する。

表 2.62 使用 RAM 説明

ラベル名、レジスタ名	機能	データ長	使用モジュール名
RVDATA	受信したデータを設定する。	1 バイト	メインルーチン
TRDATA	送信するデータを設定する。	1 ビット	メインルーチン

2.12.5 フローチャート

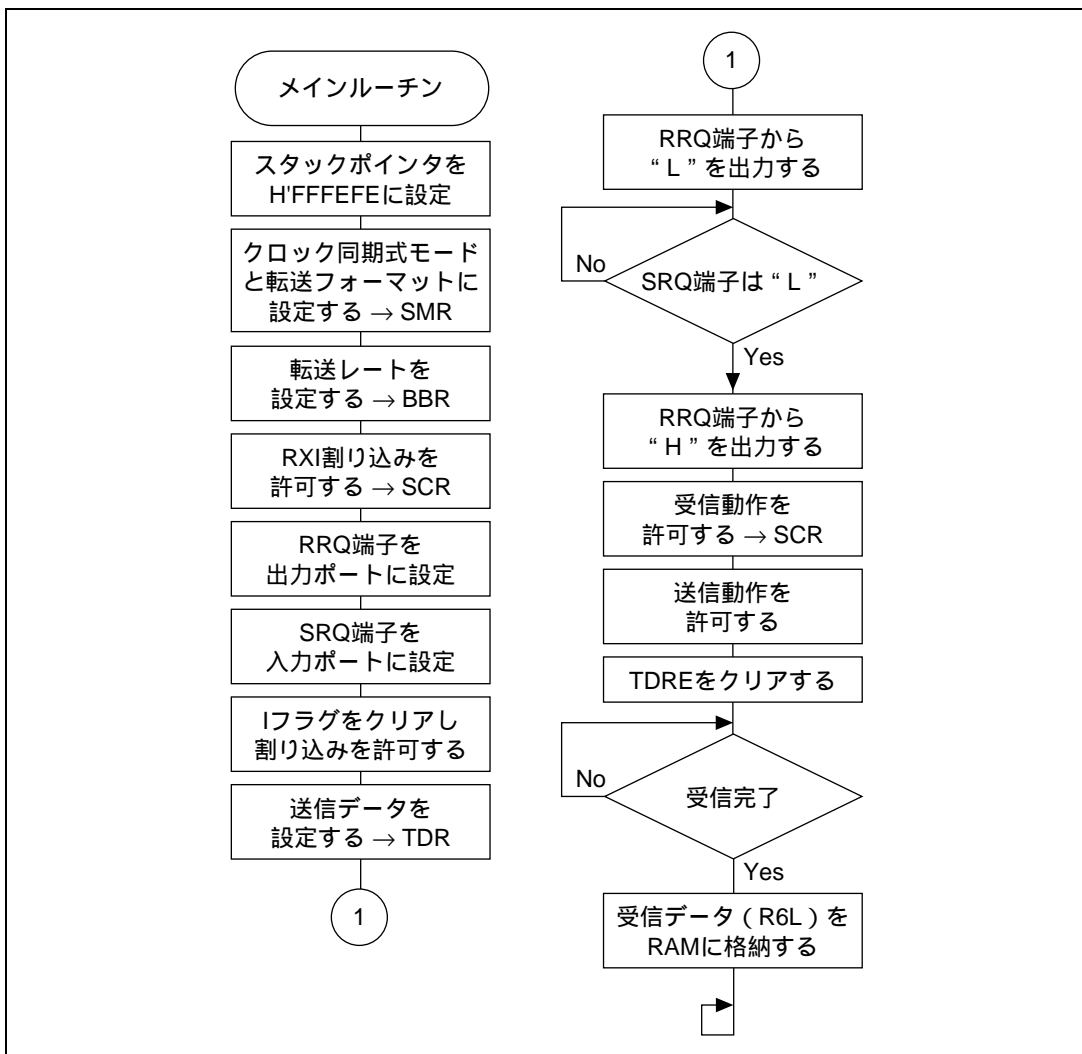


図 2.68 メインルーチン



図 2.69 データ受信完了

2.12.6 プログラムリスト

```

1          1 ;*****
2          2 ;*
3          3 ;*          VECTOR ADDRESS
4          4 ;*
5          5 ;*****
6          6 ;
7 000000   7          .SECTION VECT, CODE, LOCATE=H'000000
8          8 ;
9 000000 00010000  9 RES          .DATA.L  CSCMN
10         10 ;
11 00001C   11          .ORG          H'00001C
12 00001C 00010000 12 NMI          .DATA.L  CSCMN
13         13 ;
14 000030   14          .ORG          H'000030
15 000030 00010000 15 IRQ0         .DATA.L  CSCMN
16 000034 00010000 16 IRQ1.DATA.L CSCMN
17 000038 00010000 17 IRQ2         .DATA.L  CSCMN
18 00003C 00010000 18 IRQ3         .DATA.L  CSCMN
19 000040 00010000 19 IRQ4         .DATA.L  CSCMN
20 000044 00010000 20 IRQ5         .DATA.L  CSCMN
21 000048 00010000 21 IRQ6         .DATA.L  CSCMN
22 00004C 00010000 22 IRQ7         .DATA.L  CSCMN
23 000050 00010000 23 WOVI        .DATA.L  CSCMN
24 000054 00010000 24 CMI          .DATA.L  CSCMN
25         25 ;
26 000060   26          .ORG          H'000060
27 000060 00010000 27 IMIA0        .DATA.L  CSCMN
28 000064 00010000 28 IMIB0        .DATA.L  CSCMN
29 000068 00010000 29 OVI0        .DATA.L  CSCMN
30         30 ;
31 000070   31          .ORG          H'000070
32 000070 00010000 32 IMIA1        .DATA.L  CSCMN
33 000074 00010000 33 IMIB1        .DATA.L  CSCMN
34 000078 00010000 34 OVI1        .DATA.L  CSCMN
35         35 ;
36 000080   36          .ORG          H'000080
37 000080 00010000 37 IMIA2        .DATA.L  CSCMN
38 000084 00010000 38 IMIB2        .DATA.L  CSCMN
39 000088 00010000 39 OVI2        .DATA.L  CSCMN
40         40 ;
41 000090   41          .ORG          H'000090
42 000090 00010000 42 IMIA3        .DATA.L  CSCMN
43 000094 00010000 43 IMIB3        .DATA.L  CSCMN
44 000098 00010000 44 OVI3        .DATA.L  CSCMN
45         45 ;

```

```

46 0000A0          46          .ORG      H'0000A0
47 0000A0 00010000 47 IMIA4      .DATA.L  CSCMN
48 0000A4 00010000 48 IMIB4      .DATA.L  CSCMN
49 0000A8 00010000 49 OVI4       .DATA.L  CSCMN
50
51 0000B0          51          .ORG      H'0000B0
52 0000B0 00010000 52 DEND0A     .DATA.L  CSCMN
53 0000B4 00010000 53 DEND0B     .DATA.L  CSCMN
54 0000B8 00010000 54 DEND1A     .DATA.L  CSCMN
55 0000BC 00010000 55 DEND1B     .DATA.L  CSCMN
56 0000C0 00010000 56 DEND2A     .DATA.L  CSCMN
57 0000C4 00010000 57 DEND2B     .DATA.L  CSCMN
58 0000C8 00010000 58 DEND3A     .DATA.L  CSCMN
59 0000CC 00010000 59 DEND3B     .DATA.L  CSCMN
60 0000D0 00010000 60 ERI0       .DATA.L  CSCMN
61 0000D4 00010044 61 RXI0       .DATA.L  CSCIRX
62 0000D8 00010000 62 TXI0       .DATA.L  CSCMN
63 0000DC 00010000 63 TEI0       .DATA.L  CSCMN
64 0000E0 00010000 64 ERI1       .DATA.L  CSCMN
65 0000E4 00010000 65 RXI1       .DATA.L  CSCMN
66 0000E8 00010000 66 TXI1       .DATA.L  CSCMN
67 0000EC 00010000 67 TEI1       .DATA.L  CSCMN
68 0000F0 00010000 68 ADI        .DATA.L  CSCMN
69
70                70                ;*****
71                71                ;*
72                72                ;*      RAM ALLOCATION
73                73                ;*
74                74                ;*****
75                75                ;
76 FFFF00          76                .SECTION  RAM,DATA,LOCATE=H'FFFF00
77                77                ;
78 FFFF00 00000001 78 TRDATA     .RES.B   1          ;Transmit data
79 FFFF01 00000001 79 RVDATA     .RES.B   1          ;Receive data
80 FFFF02 00000001 80 FLAG       .RES.B   1          ;Flags
81          00000000 81 RXENDF     .EQU     0          ;Receive end flag
82                82                ;
83                83                ;*****
84                84                ;*
85                85                ;*      SYMBOL DEFINITIONS
86                86                ;*
87                87                ;*****
88                88                ;
89                89                ;***** SCI ch0 *****
90          00FFFFB0 90 SMR0       .EQU     H'FFFFB0   ;Serial mode register
91                91                ;
92          00FFFFB1 92 BRR0       .EQU     H'FFFFB1   ;Bit rate register

```

2. 基礎編

```

93          93 ;
94          00FFFFB2 94 SCRO .EQU H'FFFFB2 ;sirical control register
95          00000006 95 RIE .EQU 6 ;Receive interrupt enable
96          00000005 96 TE .EQU 5 ;Transmit enable
97          00000004 97 RE .EQU 4 ;Receive enable
98          98 ;
99          00FFFFB3 99 TDR0 .EQU H'FFFFB3 ;Transmit data register
100         100 ;
101         00FFFFB4 101 SSR0 .EQU H'FFFFB4 ;Serial status register
102         00000007 102 TDRE .EQU 7 ;Transmit data register empty
103         00000006 103 RDRF .EQU 6 ;Receive data register full
104         104 ;
105         00FFFFB5 105 RDR0 .EQU H'FFFFB5 ;Receive data register
106         106 ;
107         107 ;***** PORT A *****
108         108 ;
109         00FFFFD1 109 PADDR .EQU H'FFFFD1 ;Port A data direction
110         110 ; ;register
111         00FFFFD3 111 PADR .EQU H'FFFFD3 ;Port A data register
112         00000000 112 RRQ .EQU 0 ;RRQ bit
113         00000001 113 SRQ .EQU 1 ;SRQ bit
114         114 ;
115         115 ;*****
116         116 ;* *
117         117 ;* MAIN PROGRAM : CSCMN *
118         118 ;* *
119         119 ;*****
120         120 ;
121         010000 121 .SECTION PROG,CODE,LOCATE=H'010000
122         122 ;
123         00010000 123 CSCMN: .EQU $
124         010000 7A0700FFFEFE 124 MOV.L #H'FFFEFE,SP ;Set stack pointer
125         010006 F880 125 MOV.B #B'10000000,R0L;Clock mode,8bit
126         010008 38B0 126 MOV.B R0L,@SMR0
127         01000A F803 127 MOV.B #3,R0L ;CKS0,1=0,N=3
128         01000C 38B1 128 MOV.B R0L,@BRR0 ;Then 1Mbps
129         01000E 7FB27060 129 BSET.B #RIE,@SCRO ;Enable RXI interrupt
130         010012 7FB27040 130 BSET.B #RE,@SCRO ;Enable receive
131         010016 F801 131 MOV.B #H'01,R0L ;SRQ is input port
132         010018 38D1 132 MOV.B R0L,@PADDR ;RRQ is output port
133         133 ;
134         01001A 067F 134 ANDC #B'01111111,CCR;Clear C-FLAG
135         01001C 7FD37200 135 BCLR.B #RRQ,@PADR ;RRQ is 'Low'
136         010020 2800 136 MOV.B @TRDATA,R0L ;Set transmit data to TDR
137         010022 38B3 137 MOV.B R0L,@TDR0
138         010024 7FB47270 138 BCLR.B #TDRE,@SSR0 ;Start transmit

```



```

139 010028          139  CSCMN01:
140 010028 7ED37310 140          BTST.B   #SRQ,@PADR   ;SRQ is 'Low' ?
141 01002C 46FA     141          BNE     CSCMN01   ;No
142                142  ;
143 01002E 7FD37000 143          BSET.B   #PRQ,@PADR   ;RRQ is 'High'
144 010032 7FB27050 144          BSET.B   #TE,@SCR0    ;Enable transmit
145                145  ;
146 010036          146  CSCMN02:
147 010036 7E027300 147          BTST.B   #RXENDF,@FLAG ;Receive complete ?
148 01003A 47FA     148          BEQ     CSCMN02   ;No
149                149  ;
150 01003C 7F027200 150          BCLR.B   #RXENDF,@FLAG ;Clear RXENDF
151 010040 3E01     151          MOV.B   R6L,@RVDATA   ;Set receive data
152 010042          152  CSCMN99:
153 010042 40FE     153          BRA     CSCMN99
154                154  ;
155                155  ;*****
156                156  ;*
157                157  ;*      NAME : CSCIRX(STOP COUNTER)
158                158  ;*
159                159  ;*****
160                160  ;
161                161  ;*      ENTRY : NOTHING
162                162  ;*      RETURNS : TSTR(TIMER START REG)
163                163  ;*
164                164  ;*****
165                165  ;
166          00010044 166  CSCIRX:  .EQU    $
167 010044 2EB5     167          MOV.B   @RDR0,R6L   ;Get receive data to R6L
168 010046 7FB47260 168          BCLR.B   #RDRF,@SSR0   ;Clear RDRF
169 01004A 7F027000 169          BSET.B   #RXENDF,@FLAG ;Set RXENDF
170 01004E 5670     170          RTE
171                171  ;
172                172  .END
****TOTAL ERRORS      0
****TOTAL WARNINGS    0

```

2.13 マルチプロセッサ通信

MCU : H8/3003

使用機器 : SCI (マルチプロセッサ通信)

2.13.1 仕様

- 図 2.70 に示すように 3 つの H8/3003 が共通のシリアル通信回線を用いてデータの送受信を行います。
- マスタ側 H8/3003 は 2 つのスレーブ側 H8/3003 へ各々データを送信し、スレーブ側 H8/3003 は自局へのデータのみ受信します。
- データは 9600BPS、8 ビットデータ、1 ストップビットおよびノンパリティで送受信します。

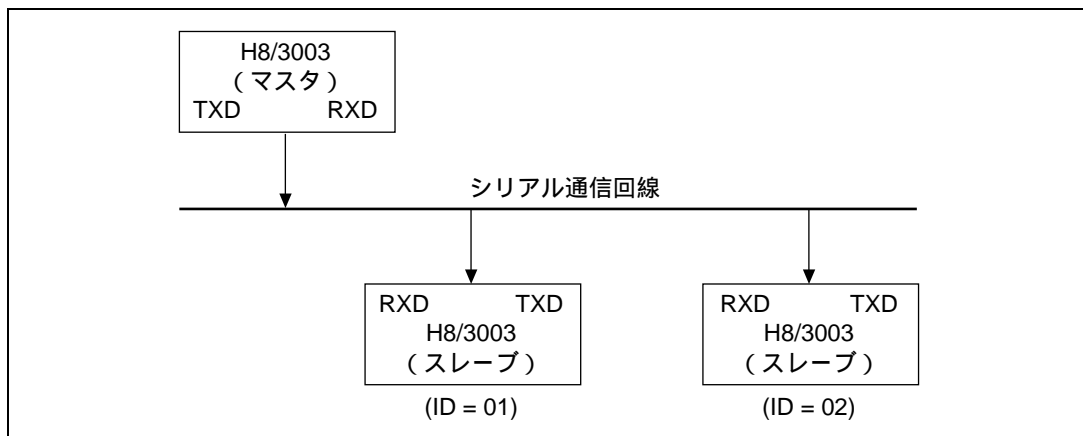


図 2.70 マルチプロセッサ機能を使用した調歩同期式 SCI インタフェースブロック図

2.13.2 使用機能説明

本タスク例では、SCIのマルチプロセッサ通信機能を使用して、マルチプロセッサ通信を行います。

図 2.71 に本タスク例で使用する、マスタ側 SCI のブロック図を示します。本タスク例では、SCI の以下の機能を使用してマスタ側送信を行います。

- キャラクタ単位で同期をとる調歩同期方式でデータの通信を行う機能。(調歩同期式モード)
- マルチプロセッサビットを付加したフォーマットで通信する機能。(マルチプロセッサ通信機能)
- 送信開始時に割り込みを発生する機能。(TXI 割り込み)
- 送信完了時に割り込みを発生する機能。(TEI 割り込み)

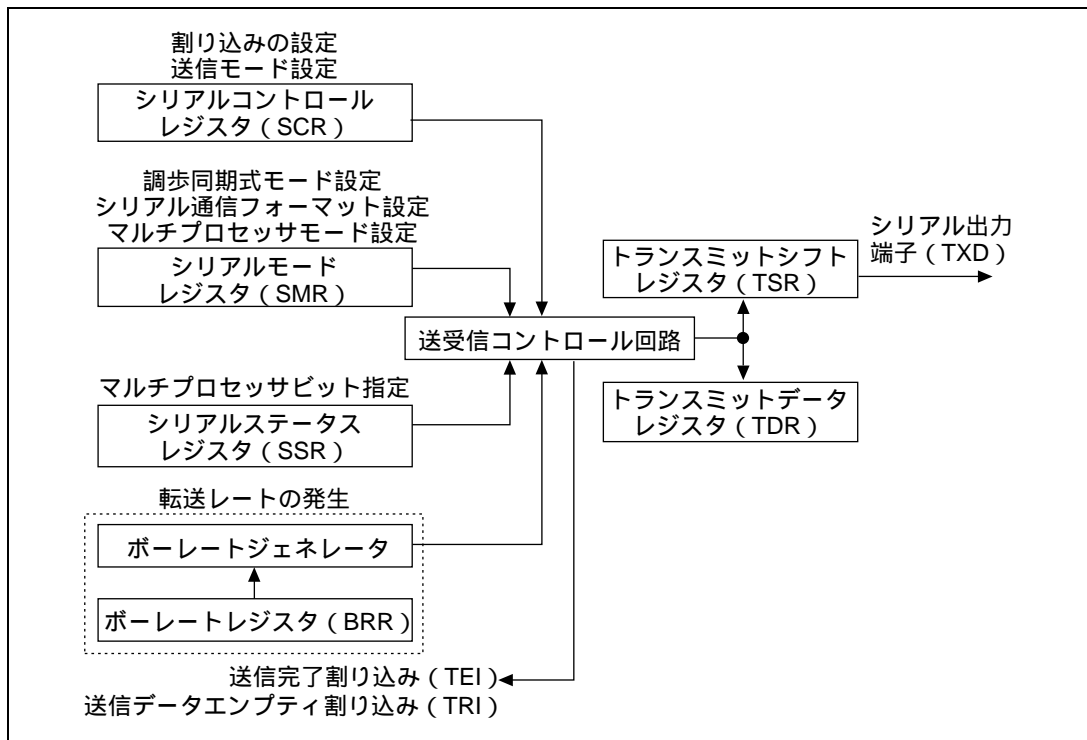


図 2.71 マスタ側 SCI ブロック図

2. 基礎編

図 2.72 に本タスク例で使用する、スレーブ側 SCI のブロック図を示します。本タスク例では、SCI の以下の機能を使用してスレーブ側受信を行います。

- キャラクタ単位で同期をとる調歩同期方式でデータの通信を行う機能。(調歩同期式モード)
- マルチプロセッサビットを付加したフォーマットで通信する機能。(マルチプロセッサ通信機能)
- マルチプロセッサビット受信時に割り込みを発生する機能。(マルチプロセッサ割り込み)
- 受信完了時に割り込みを発生する機能。(RXI 割り込み)

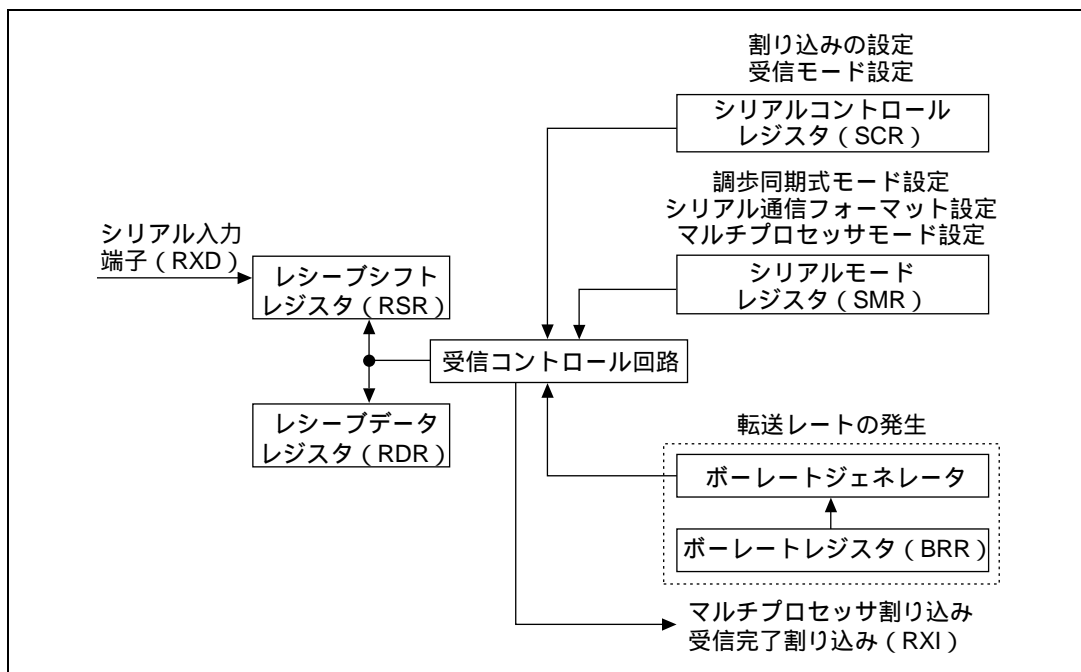


図 2.72 スレーブ側 SCI ブロック図

表 2.63 に本タスク例の機能割り付けを示します。表 2.63 に示すように SCI の機能を割り付け、マルチプロセッサ通信を行います。

表 2.63 SCI 機能割り付け

SCI 機能	機能
RXD	H8/3003 からデータを受信する
TXD	H8/3003 へデータを送信する
SMR	SCI を調歩同期式モード、マルチプロセッサモードに設定する
SCR	送 / 受信割り込みを許可し、SCI を送 / 受信モードに設定する
SSR	送信を開始する / マルチプロセッサビットを設定する
RDR	H8/3003 から受信したデータを設定する
TDR	H8/3003 へ送信するデータを設定する
BRR	転送レートを設定する

2.13.3 動作説明

- マスタ側動作

図2.73に本タスクのマスタ側動作原理を示します。図2.73に示すタイミングでハードウェア処理およびソフトウェア処理を行い、受信局のH8/3003ヘータを送信します。

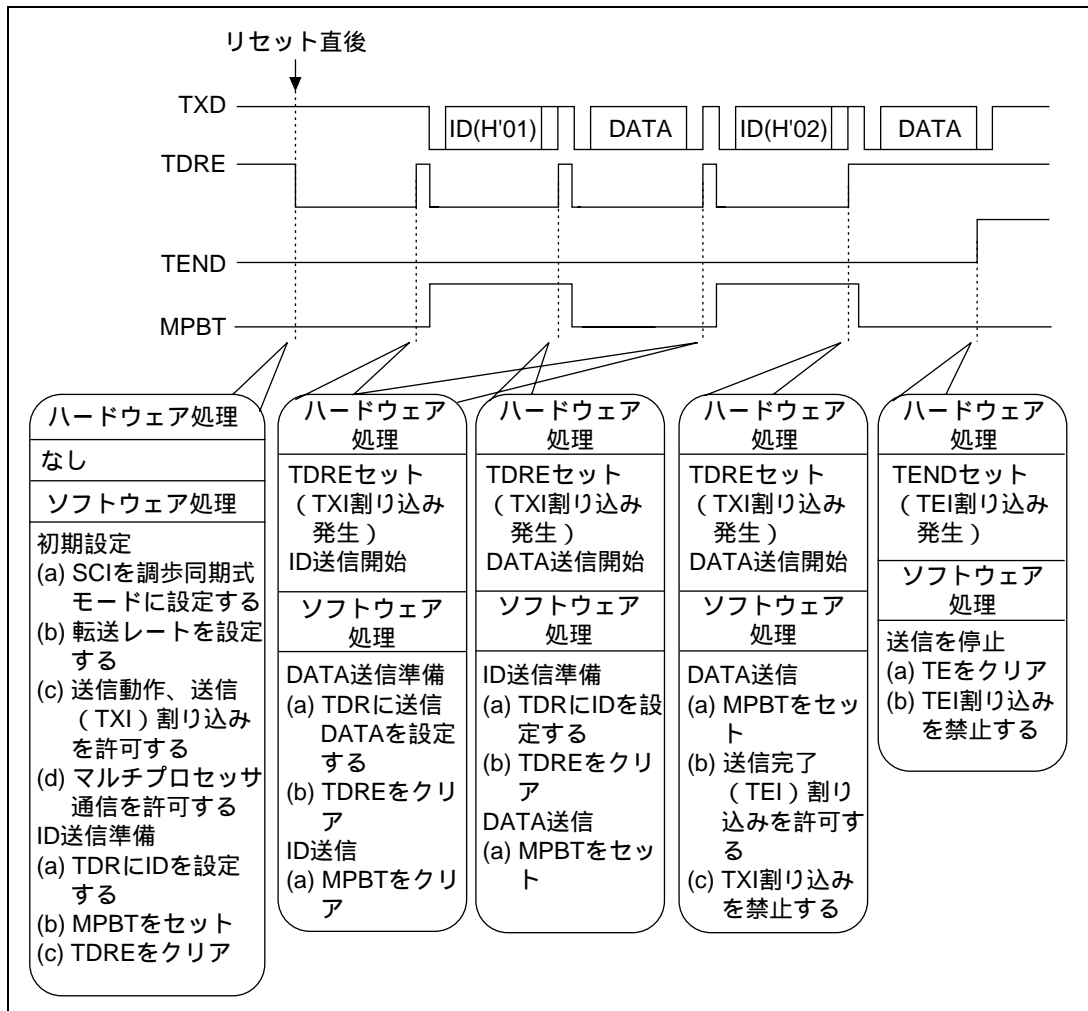


図 2.73 マルチプロセッサ通信 (マスタ側) 動作原理

• スレーブ側動作

図2.74に本タスクのスレーブ側動作原理を示します。図2.74に示すタイミングでハードウェア処理およびソフトウェア処理を行い、送信局からのデータを受信します。

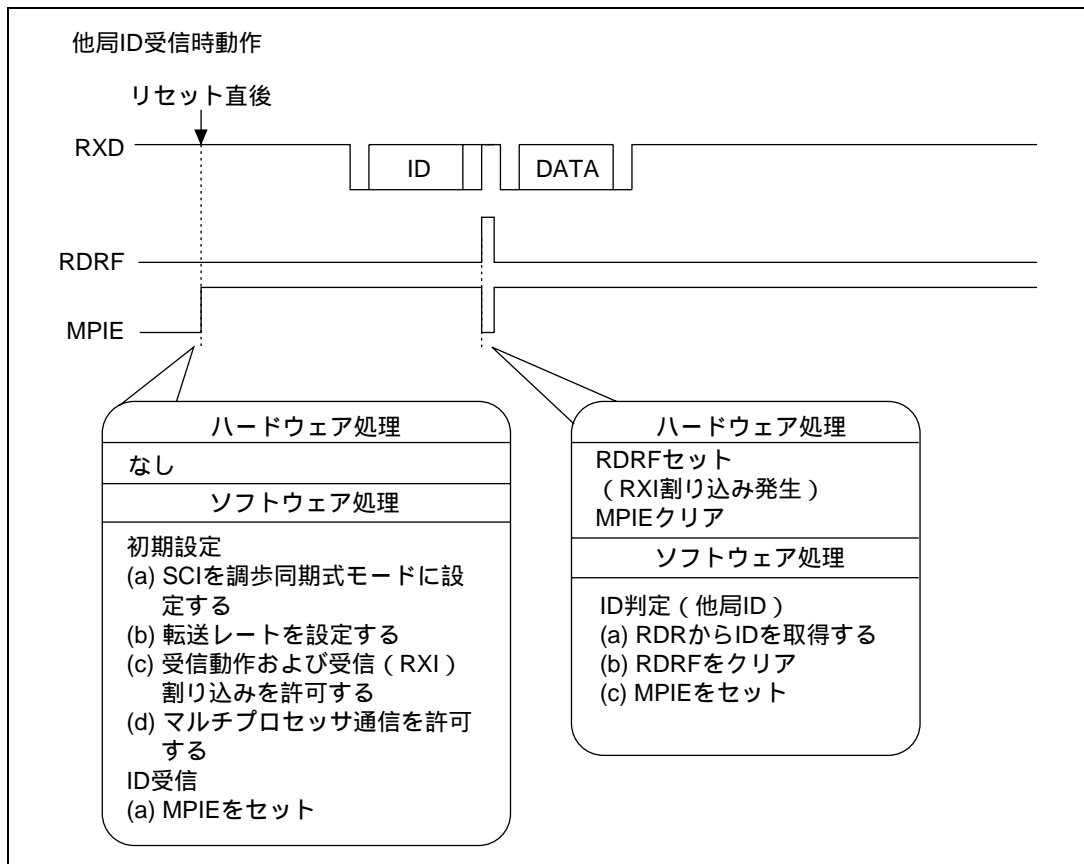


図 2.74 マルチプロセッサ通信 (スレーブ側) 動作原理 (1)

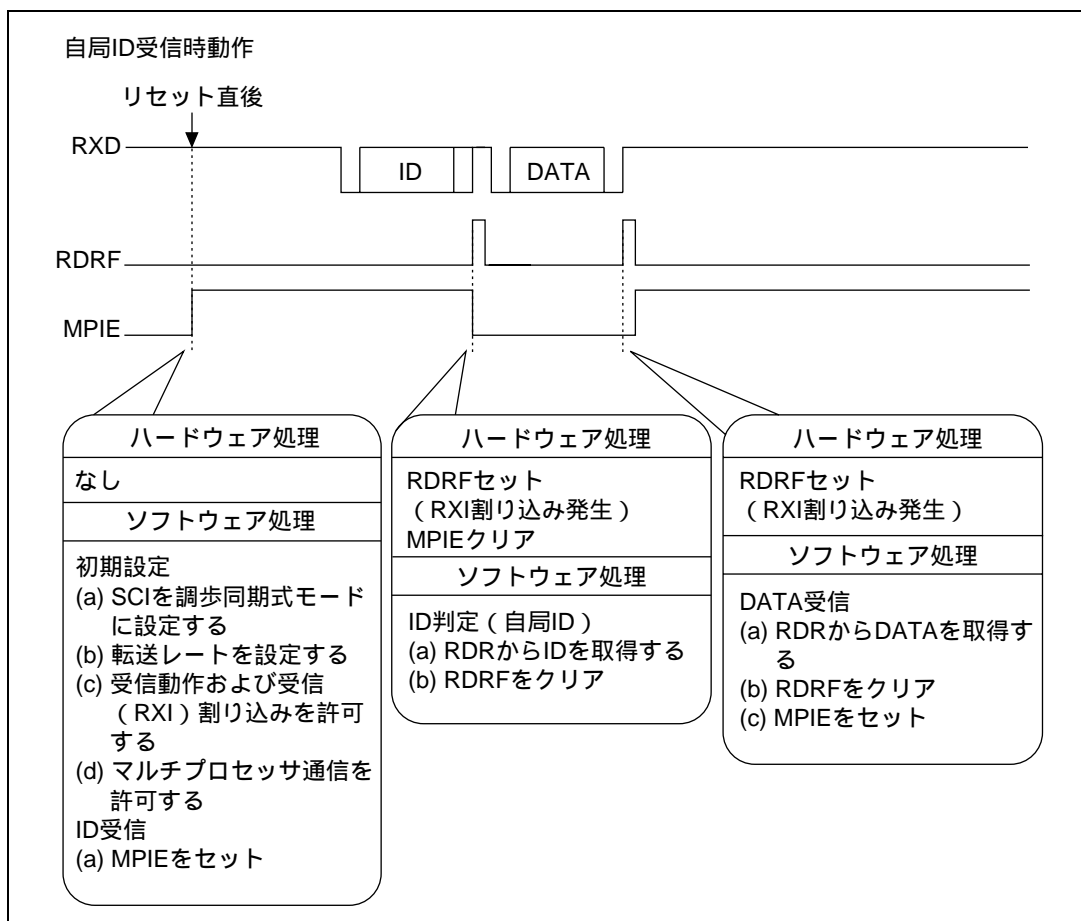


図 2.74 マルチプロセッサ通信(スレーブ側)動作原理(2)

2. 基礎編

2.13.4 ソフトウェア説明

- マスタ側ソフトウェア説明

表 2.64 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	MPMASMN	SCI の初期設定を行う。
データ送信	MPSCITX	TXI 割り込みで起動し、ID およびデータの送信を行う。
送信終了	MPSCITE	TEI 割り込みで起動し、SCI 送信を停止する。

表 2.65 引数の説明

ラベル名、レジスタ名	機能	データ長	使用モジュール名	入出力
TXDATA	スレーブ側 H8/3003 へ送信する ID、データを格納するバッファ。	4 バイト	メインルーチン	出力
			データ送信	入力

表 2.66 使用内部レジスタ説明

レジスタ名	機能	使用モジュール名
SMR	SCI のモード (調歩同期式)、転送フォーマットおよびボーレートジェネレータへのクロック選択 (クロック入力) を設定する。	メインルーチン
SCR	割り込み (TXI、TEI) を許可し、SCI の送信許可 / 禁止を設定する。	メインルーチン 送信終了
SSR	TDRE (b7) をクリアすることにより、送信開始を指示する。	メインルーチン データ送信
TDR	スレーブ側 H8/3003 へ送信する ID、データを設定する。	メインルーチン データ送信
BRR	転送レートを設定する。	メインルーチン

表 2.67 汎用レジスタ説明

モジュール名	レジスタ名	機能
メインルーチン	ROL	SCI、引数 RAM を設定するときのワークとして使用する。
データ送信	ROL	送信データを TDR へ設定するときのワークとして使用する。

表 2.68 使用 RAM 説明

ラベル名	使用モジュール名	データ長	機能
TXCNT	データ送信	1 バイト	送信したデータ数をカウントする。

- スレーブ側ソフトウェア説明

表 2.69 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	MPSRVMN	SCI の初期設定を行う。
データ受信	MPSCIRX	RXI 割り込みで起動し、ID およびデータの受信を行う。

表 2.70 引数の説明

ラベル名、レジスタ名	機能	データ長	使用モジュール名	入出力
R6L	受信した ID、データを設定する。	1 バイト	データ受信	出力
			メインルーチン	入力
IDRCVF	自局 ID 受信を示すフラグ。 1 : ID 受信あり 0 : ID 受信なし	1 バイト	データ受信	出力
			メインルーチン	入力
DTRCVF	データ受信を示すフラグ。 1 : データ受信あり 0 : データ受信なし	1 バイト	データ受信	出力
			メインルーチン	入力

表 2.71 使用内部レジスタ説明

レジスタ名	機能	使用モジュール名
SMR	SCI のモード (調歩同期式)、転送フォーマットおよびボーレートジェネレータへのクロック選択 (クロック入力) を設定する。	メインルーチン
SCR	割り込み (RXI) を許可し、SCI の受信許可を設定する。	メインルーチン
RDR	マスタ側 H8/3003 から受信した ID、データを設定する。	データ受信
BRR	転送レートを設定する。	メインルーチン

表 2.72 汎用レジスタ説明

モジュール名	レジスタ名	機能
メインルーチン	R0L	SCI、引数 RAM を設定するときのワークとして使用する。
	R6L	受信したデータを受け取る引数として使用する。
データ受信	R6L	受信データをメインルーチンへ渡す引数として使用する。

表 2.73 使用 RAM 説明

ラベル名	使用モジュール名	データ長	機能
RXID	メインルーチン	1 バイト	受信した ID を設定する。
RXDATA	メインルーチン	1 バイト	受信したデータを設定する。
MYID	データ受信	1 バイト	自局 ID を設定する。

2.13.5 フローチャート

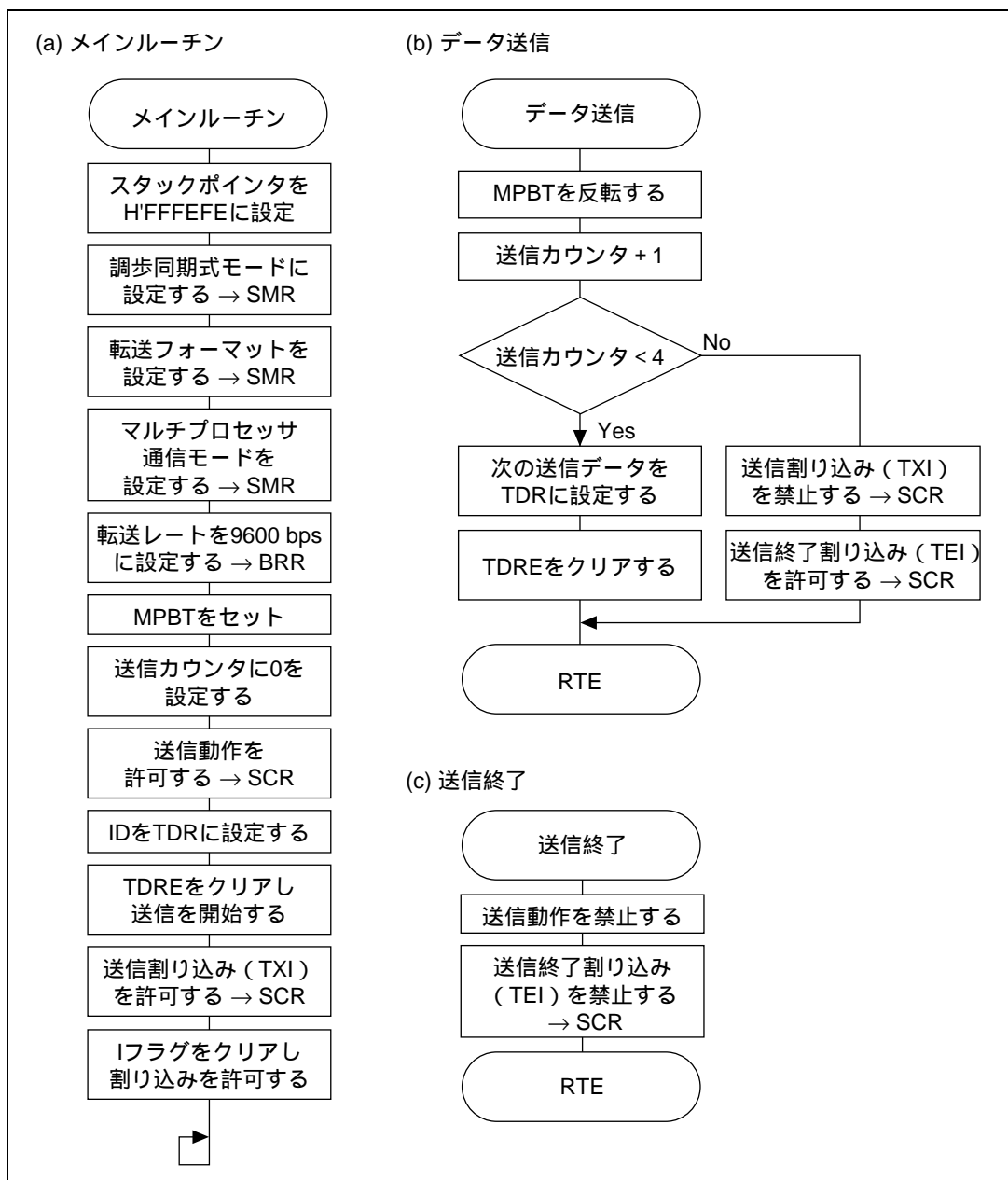


図 2.75 マスタ側ソフトウェアフローチャート

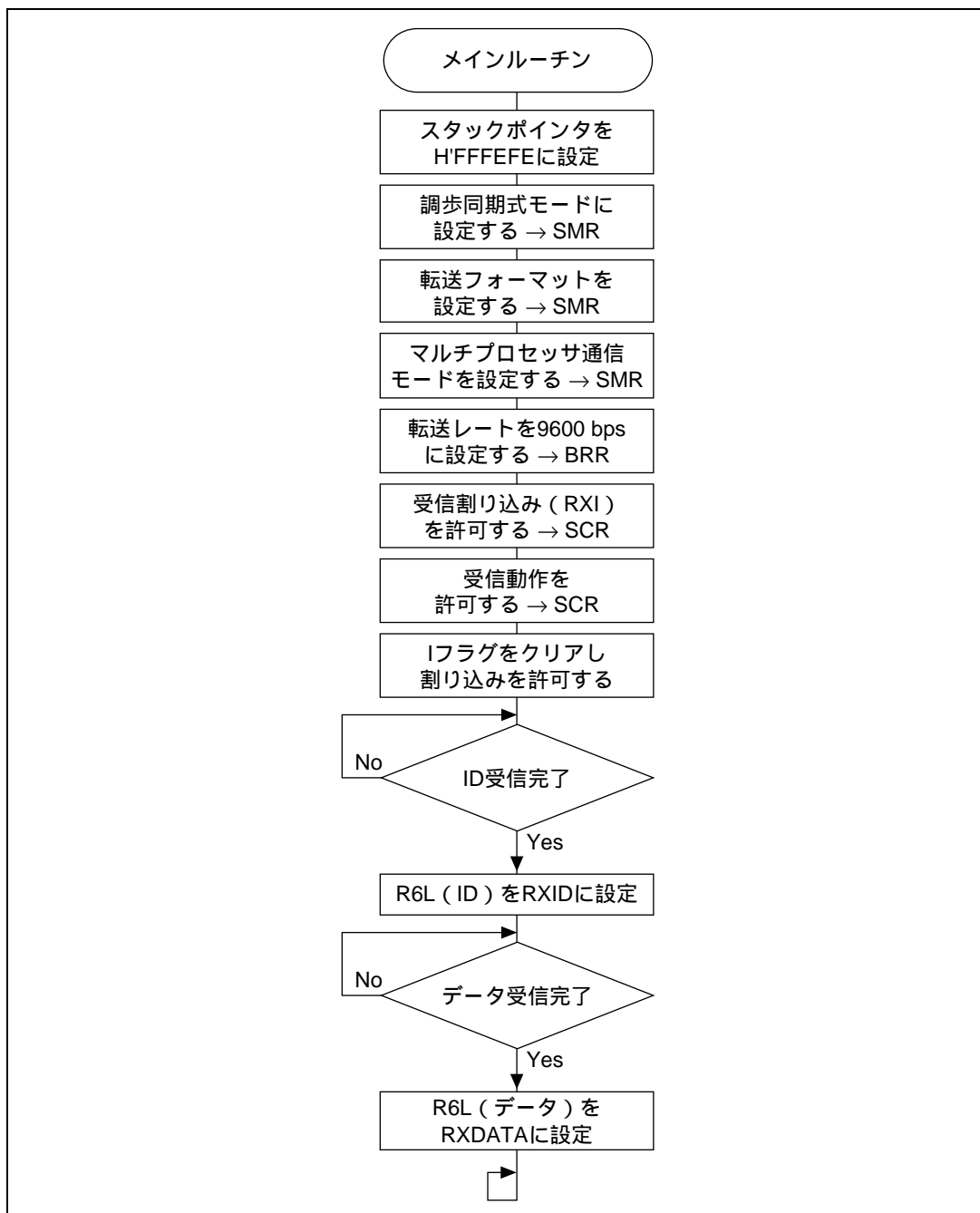


図 2.76 スレーブ側ソフトウェアフローチャート (メインルーチン)

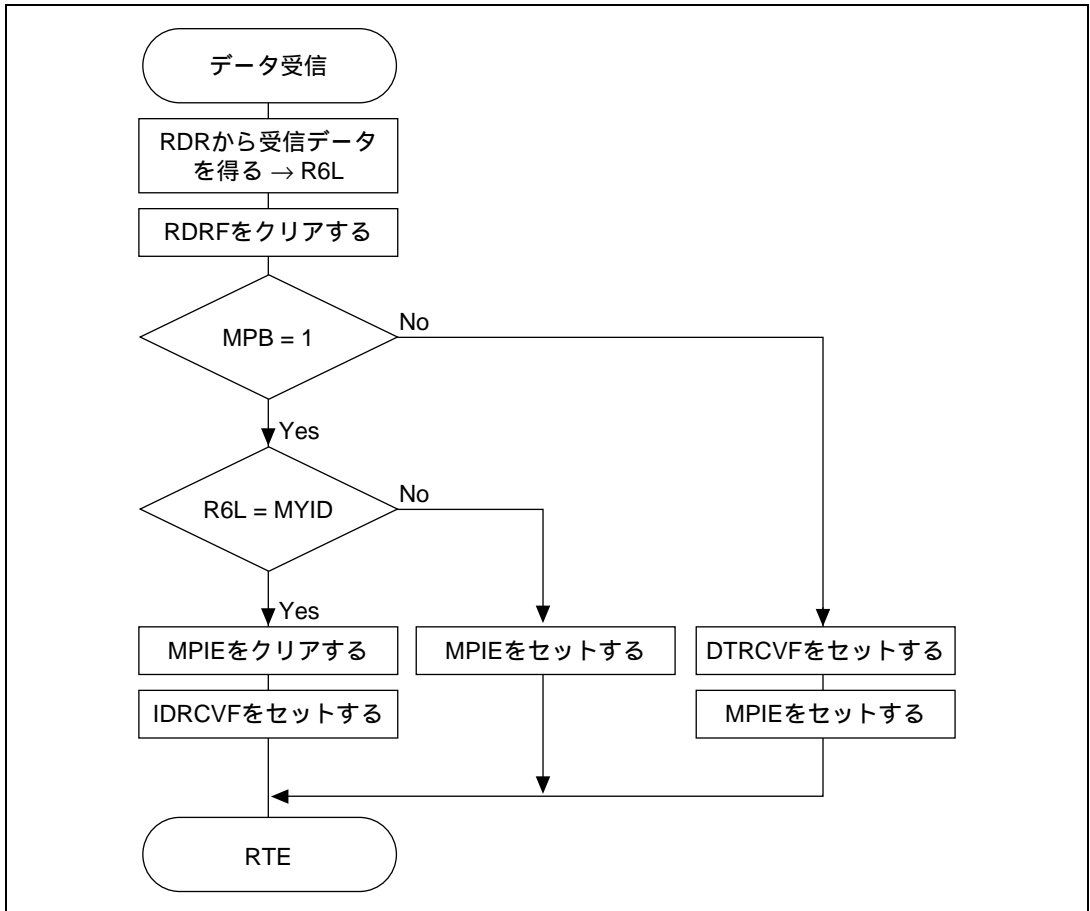


図 2.77 スレーブ側ソフトウェアフローチャート (データ受信)

2.13.6 プログラムリスト

```

1          1 ;*****
2          2 ;*
3          3 ;*          VECTOR ADDRESS
4          4 ;*
5          5 ;*****
6          6 ;
7          7          .CPU      300HA
8 000000   8          .SECTION  VECT, CODE, LOCATE=H'000000
9          9 ;
10 000000 00010000 10 RES          .DATA.L  MPMASMN
11          11 ;
12 00001C   12          .ORG      H'00001C
13 00001C 00010000 13 NMI          .DATA.L  MPMASMN
14 000020 00010000 14 TRAP0        .DATA.L  MPMASMN
15 000024 00010000 15 TRAP1        .DATA.L  MPMASMN
16 000028 00010000 16 TRAP2        .DATA.L  MPMASMN
17 00002C 00010000 17 TRAP3        .DATA.L  MPMASMN
18          18 ;
19 000030   19          .ORG      H'000030
20 000030 00010000 20 IRQ0         .DATA.L  MPMASMN
21 000034 00010000 21 IRQ1         .DATA.L  MPMASMN
22 000038 00010000 22 IRQ2         .DATA.L  MPMASMN
23 00003C 00010000 23 IRQ3         .DATA.L  MPMASMN
24 000040 00010000 24 IRQ4         .DATA.L  MPMASMN
25 000044 00010000 25 IRQ5         .DATA.L  MPMASMN
26 000048 00010000 26 IRQ6         .DATA.L  MPMASMN
27 00004C 00010000 27 IRQ7         .DATA.L  MPMASMN
28 000050 00010000 28 WOVI        .DATA.L  MPMASMN
29 000054 00010000 29 CMI          .DATA.L  MPMASMN
30          30 ;
31 000060   31          .ORG      H'000060
32 000060 00010000 32 IMIA0        .DATA.L  MPMASMN
33 000064 00010000 33 IMIB0        .DATA.L  MPMASMN
34 000068 00010000 34 OVI0         .DATA.L  MPMASMN
35          35 ;
36 000070   36          .ORG      H'000070
37 000070 00010000 37 IMIA1        .DATA.L  MPMASMN
38 000074 00010000 38 IMIB1        .DATA.L  MPMASMN
39 000078 00010000 39 OVI1         .DATA.L  MPMASMN
40          40 ;
41 000080   41          .ORG      H'000080
42 000080 00010000 42 IMIA2        .DATA.L  MPMASMN
43 000084 00010000 43 IMIB2        .DATA.L  MPMASMN
44 000088 00010000 44 OVI2         .DATA.L  MPMASMN
45          45 ;

```

2. 基礎編

```

46 000090          46          .ORG      H'000090
47 000090 00010000 47 IMIA3    .DATA.L  MPMASMN
48 000094 00010000 48 IMIB3    .DATA.L  MPMASMN
49 000098 00010000 49 OVI3     .DATA.L  MPMASMN
50
51 0000A0          51          .ORG      H'0000A0
52 0000A0 00010000 52 IMIA4    .DATA.L  MPMASMN
53 0000A4 00010000 53 IMIB4    .DATA.L  MPMASMN
54 0000A8 00010000 54 OVI4     .DATA.L  MPMASMN
55
56 0000B0          56          .ORG      H'0000B0
57 0000B0 00010000 57 DEND0A   .DATA.L  MPMASMN
58 0000B4 00010000 58 DEND0B   .DATA.L  MPMASMN
59 0000B8 00010000 59 DEND1A   .DATA.L  MPMASMN
60 0000BC 00010000 60 DEND1B   .DATA.L  MPMASMN
61 0000C0 00010000 61 DEND2A   .DATA.L  MPMASMN
62 0000C4 00010000 62 DEND2B   .DATA.L  MPMASMN
63 0000C8 00010000 63 DEND3A   .DATA.L  MPMASMN
64 0000CC 00010000 64 DEND3B   .DATA.L  MPMASMN
65 0000D0 00010000 65 ERI0     .DATA.L  MPMASMN
66 0000D4 00010000 66 RXI0     .DATA.L  MPMASMN
67 0000D8 0001003A 67 TXI0     .DATA.L  MPSCITX
68 0000DC 00010070 68 TEI0     .DATA.L  MPSCITE
69 0000E0 00010000 69 ERI1     .DATA.L  MPMASMN
70 0000E4 00010000 70 RXI1     .DATA.L  MPMASMN
71 0000E8 00010000 71 TXI1     .DATA.L  MPMASMN
72 0000EC 00010000 72 TEI1     .DATA.L  MPMASMN
73 0000F0 00010000 73 ADI      .DATA.L  MPMASMN
74
75
76
77
78
79
80
81 FFFF00          81          .SECTION  RAM,DATA,LOCATE=H'FFFF00
82
83 FFFF00 00000004 83 TXDATA   .RES.B   4          ;Transmit data
84 FFFF04 00000001 84 TXCNT    .RES.B   1          ;Transmit counter
85
86
87
88
89
90
91
92      00FFFFB0   92 SMR0     .EQU     H'FFFFB0   ;Serial mode register

```

```

93          93 ;
94          00FFFFB1 94 BRR0 .EQU H'FFFFB1 ;Bit rate register
95          95 ;
96          00FFFFB2 96 SCRO .EQU H'FFFFB2 ;Srial control register
97          00000007 97 TIE .EQU 7 ;Transmit interrupt enable
98          00000005 98 TE .EQU 5 ;Transmit enable
99          00000002 99 TEIE .EQU 2 ;Transmit end interrupt
;enable

100         100 ;
101         00FFFFB3 101 TDR0 .EQU H'FFFFB3 ;Transmit data register
102         102 ;
103         00FFFFB4 103 SSR0 .EQU H'FFFFB4 ;Serial status register
104         00000007 104 TDRE .EQU 7 ;Transmit data register empty
105         00000000 105 MPBT .EQU 0 ;Multi processor bit transfer
106         106 ;
107         107 ;*****
108         108 ;* *
109         109 ;* MAIN PROGRAM : MPMASMN *
110         110 ;* *
111         111 ;*****
112         112 ;
113 010000 113 .SECTION PROG, CODE, LOCATE=H'010000
114         114 ;
115         00010000 115 MPMASMN: .EQU $
116 010000 7A0700FFFEFE 116 MOV.L #H'FFFEFE, SP ;Set stack pointer
117 010006 F804 117 MOV.B #B'00000100, R0L ;Asynco mode, 8bit, np,
;stop1, multi processor m
118 010008 38B0 118 MOV.B R0L, @SMR0
119 01000A F833 119 MOV.B #51, R0L ;CKS0, 1=0, N=51
120 01000C 38B1 120 MOV.B R0L, @BRR0 ;Then 9600bps
121 01000E 7FB47000 121 BSET.B #MPBT, @SSR0 ;Set MPBT
122         122 ;
123 010012 790001AA 123 MOV.W #H'01AA, R0 ;Set ID=01 & data
124 010016 6B80FF00 124 MOV.W R0, @TXDATA+0
125 01001A 79000255 125 MOV.W #H'0255, R0 ;Set ID=02 & data
126 01001E 6B80FF02 126 MOV.W R0, @TXDATA+2
127         127 ;
128 010022 F800 128 MOV.B #H'0, R0L ;Set TXCNT=0
129 010024 3804 129 MOV.B R0L, @TXCNT
130         130 ;
131 010026 2800 131 MOV.B @TXDATA, R0L ;Set ID(H'01) to TDR0
132 010028 3883 132 MOV.B R0L, @TDR0
133 01002A 7FB27050 133 BSET.B #TE, @SCRO ;Enable transmit
134 01002E 7FB47270 134 BCLR.B #TDRE, @SSR0 ;Start transmit
135 010032 7FB27070 135 BSET.B #TIE, @SCRO ;Enable TXI interrupt
136         136 ;
137 010036 067F 137 ANDC #B'01111111, CCR ;Clear I-FLAG

```

2. 基礎編

```
138 010038          138 MPMAS99:
139 010038 40FE      139          BRA          MPMAS99
140                140 ;
141                141 ;*****
142                142 ;*
143                143 ;* NAME : MPSCITX(SCI transmit interrupt) *
144                144 ;*
145                145 ;*****
146                146 ;*
147                147 ;* ENTRY : TXDATA (Transmit data) *
148                148 ;* RETURNS : NOTHING *
149                149 ;*
150                150 ;*****
151                151 ;
152          0001003A 152 MPSCITX: .EQU      $
153 01003A 6DF0      153          PUSH.W      R0          ;Save registers
154 01003C 6DF1      154          PUSH.W      R1
155                155 ;
156 01003E 7FB47100 156          BNOT.B      #MPBT,@SSR0 ;Invert MPBT
157 010042 F100      157          MOV.B        #0,R1H      ;TXCNT = TXCNT + 1
158 010044 2904      158          MOV.B        @TXCNT,R1L
159 010046 0A09      159          INC          R1L
160 010048 3904      160          MOV.B        R1L,@TXCNT
161 01004A A904      161          CMP.B        #4,R1L      ;TXCNT >= 4 ?
162 01004C 58400012 162          BCC          MPTX10    ;Yes
163                163 ;
164 010050 78106A2800FFFF00 164          MOV.B        @(TXDATA,R1),R0L;load next trans data
165 010058 38B3      165          MOV.B        R0L,@TDR0    ;Set next data to TDR0
166 01005A 7FB47270 166          BCLR.B      #TDRE,@SSR0    ;Start transmit
167 01005E 58000008 167          BRA          MPTX99
168                168 ;
169 010062          169 MPTX10:
170 010062 7FB27270 170          BCLR.B      #TIE,@SCR0    ;Disable TXI
171 010066 7FB27020 171          BSET.B      #TEIE,@SCR0    ;Enable TEI interrupt
172                172 ;
173 01006A          173 MPTX99:
174 01006A 6D71      174          POP.W      R1
175 01006C 6D70      175          POP.W      R0
176 01006E 5670      176          RTE
177                177 ;
178                178 ;*****
179                179 ;*
180                180 ;* NAME : MPSCITI(SCI transmit end interrupt) *
181                181 ;*
182                182 ;*****
183                183 ;*
184                184 ;* ENTRY : NOTHING *
```



```
185          185  ;*      RETURNS : NOTHING          *
186          186  ;*
187          187  ;*****
188          188  ;
189          00010070 189  MPSCITE:  .EQU      $
190  010070 7FB27250 190          BCLR.B  #TE,@SCR0  ;Disable transmit
191  010074 7FB27220 191          BCLR.B  #TEIE,@SCR0 ;disable TEI interrupt
192  010078 5670    192          RTE
193          193  ;
194          194          .END
****TOTAL ERRORS      0
****TOTAL WARNINGS    0
```

2. 基礎編

```
1          1 ;*****
2          2 ;*
3          3 ;*          VECTOR ADDRESS
4          4 ;*
5          5 ;*****
6          6 ;
7          7          .CPU          300HA
8 000000    8          .SECTION  VECT, CODE, LOCATE=H'000000
9          9 ;
10 000000 00010000 10 RES          .DATA.L  MPSRVMN
11          11 ;
12 00001C    12          .ORG          H'00001C
13 00001C 00010000 13 NMI          .DATA.L  MPSRVMN
14          14 ;
15 000030    15          .ORG          H'000030
16 000030 00010000 16 IRQ0         .DATA.L  MPSRVMN
17 000034 00010000 17 IRQ1         .DATA.L  MPSRVMN
18 000038 00010000 18 IRQ2         .DATA.L  MPSRVMN
19 00003C 00010000 19 IRQ3         .DATA.L  MPSRVMN
20 000040 00010000 20 IRQ4         .DATA.L  MPSRVMN
21 000044 00010000 21 IRQ5         .DATA.L  MPSRVMN
22 000048 00010000 22 IRQ6         .DATA.L  MPSRVMN
23 00004C 00010000 23 IRQ7         .DATA.L  MPSRVMN
24 000050 00010000 24 WOVI        .DATA.L  MPSRVMN
25 000054 00010000 25 CMI          .DATA.L  MPSRVMN
26          26 ;
27 000060    27          .ORG          H'000060
28 000060 00010000 28 IMIA0        .DATA.L  MPSRVMN
29 000064 00010000 29 IMIB0        .DATA.L  MPSRVMN
30 000068 00010000 30 OVI0         .DATA.L  MPSRVMN
31          31 ;
32 000070    32          .ORG          H'000070
33 000070 00010000 33 IMIA1        .DATA.L  MPSRVMN
34 000074 00010000 34 IMIB1        .DATA.L  MPSRVMN
35 000078 00010000 35 OVI1         .DATA.L  MPSRVMN
36          36 ;
37 000080    37          .ORG          H'000080
38 000080 00010000 38 IMIA2        .DATA.L  MPSRVMN
39 000084 00010000 39 IMIB2        .DATA.L  MPSRVMN
40 000088 00010000 40 OVI2         .DATA.L  MPSRVMN
41          41 ;
42 000090    42          .ORG          H'000090
43 000090 00010000 43 IMIA3        .DATA.L  MPSRVMN
44 000094 00010000 44 IMIB3        .DATA.L  MPSRVMN
45 000098 00010000 45 OVI3         .DATA.L  MPSRVMN
46          46 ;
47 0000A0    47          .ORG          H'0000A0
```

```

48 0000A0 00010000      48 IMIA4      .DATA.L  MPSRVMN
49 0000A4 00010000      49 IMIB4      .DATA.L  MPSRVMN
50 0000A8 00010000      50 OVI4       .DATA.L  MPSRVMN
51                               51 ;
52 0000B0                               52          .ORG      H'0000B0
53 0000B0 00010000      53 DEND0A     .DATA.L  MPSRVMN
54 0000B4 00010000      54 DEND0B     .DATA.L  MPSRVMN
55 0000B8 00010000      55 DEND1A     .DATA.L  MPSRVMN
56 0000BC 00010000      56 DEND1B     .DATA.L  MPSRVMN
57 0000C0 00010000      57 DEND2A     .DATA.L  MPSRVMN
58 0000C4 00010000      58 DEND2B     .DATA.L  MPSRVMN
59 0000C8 00010000      59 DEND3A     .DATA.L  MPSRVMN
60 0000CC 00010000      60 DEND3B     .DATA.L  MPSRVMN
61 0000D0 00010000      61 ERI0       .DATA.L  MPSRVMN
62 0000D4 00010032      62 RXI0       .DATA.L  MPSCIRX
63 0000D8 00010000      63 TXI0       .DATA.L  MPSRVMN
64 0000DC 00010000      64 TEI0       .DATA.L  MPSRVMN
65 0000E0 00010000      65 ERI1       .DATA.L  MPSRVMN
66 0000E4 00010000      66 RXI1       .DATA.L  MPSRVMN
67 0000E8 00010000      67 TXI1       .DATA.L  MPSRVMN
68 0000EC 00010000      68 TEI1       .DATA.L  MPSRVMN
69 0000F0 00010000      69 ADI        .DATA.L  MPSRVMN
70                               70 ;
71                               71 ;*****
72                               72 ;*
73                               73 ;*      RAM ALLOCATION
74                               74 ;*
75                               75 ;*****
76                               76 ;
77 FFFF00                               77          .SECTION  RAM,DATA,LOCATE=H'FFFF00
78                               78 ;
79 FFFF00 00000001      79 RXID       .RES.B   1          ;Receive ID cord
80          00000001      80 MYID       .EQU     H'01      ;My ID cord
81                               81 ;
82 FFFF01 00000001      82 RXDATA     .RES.B   1          ;Receive data
83                               83 ;
84 FFFF02 00000001      84 RCVFLAG    .RES.B   1          ;Flag area
85          00000000      85 IDRCVF     .EQU     0          ;ID cord compare flag
86          00000001      86 DTRCVF     .EQU     1          ;Data receive flag
87                               87 ;
88                               88 ;*****
89                               89 ;*
90                               90 ;*      SYMBOL DEFINITIONS
91                               91 ;*
92                               92 ;*****
93                               93 ;
94          00FFFFB0      94 SMR0       .EQU     H'FFFFB0   ;Serial mode register

```

2. 基礎編

```

95          95 ;
96          00FFFFB1 96 BRR0 .EQU H'FFFFB1 ;Bit rate register
97          97 ;
98          00FFFFB2 98 SCRO .EQU H'FFFFB2 ;Sirial control register
99          00000006 99 RIE .EQU 6 ;Receive interrupt enable
100         00000004 100 RE .EQU 4 ;Receive enable
101         00000003 101 MPIE .EQU 3 ;Multi processor interrupt
           ;enable
102         102 ;
103         00FFFFB3 103 TDR0 .EQU H'FFFFB3 ;Transmit data register
104         104 ;
105         00FFFFB4 105 SSR0 .EQU H'FFFFB4 ;Serial status register
106         00000006 106 RDRF .EQU 6 ;Receive data register full
107         00000002 107 TEND .EQU 2 ;Transmit end
108         00000001 108 MPB .EQU 1 ;Multi processor bit
109         109 ;
110         00FFFFB5 110 RDR0 .EQU H'FFFFB5 ;Receive data register
111         111 ;
112         112 ;*****
113         113 ;* *
114         114 ;* MAIN PROGRAM : MPSRVMN *
115         115 ;* *
116         116 ;*****
117         117 ;
118 010000 118 .SECTION PROG,CODE,LOCATE=H'010000
119         119 ;
120         00010000 120 MPSRVMN: .EQU $
121 010000 7A0700FFFEFE 121 MOV.L #H'FFFEFE,SP ;Set stack pointer
122 010006 F804 122 MOV.B #B'00000100,R0L ;Asyncro mode,8bit,np,
           ;stop1,multi processor m
123 010008 38B0 123 MOV.B R0L,@SMR0
124 01000A F833 124 MOV.B #51,R0L ;CKS0,1=0,N=51
125 01000C 38B1 125 MOV.B R0L,@BRR0 ;Then 9600bps
126 01000E 7FB27060 126 BSET #RIE,@SCRO ;Enable RXI interrupt
127 010012 7FB27040 127 BSET #RE,@SCRO ;Enable receive
128         128 ;
129 010016 067F 129 ANDC #B'01111111,CCR ;Clear C-FLAG
130 010018 130 MPSRV01:
131 010018 7E027300 131 BTST #IDRCVF,@RCVFLAG;ID receive ?
132 01001C 47FA 132 BEQ MPSRV01 ;No
133         133 ;
134 01001E 7F027200 134 BCLR #IDRCVF,@RCVFLAG;Clear flag
135 010022 3E00 135 MOV.B R6L,@RXID ;Store receive ID
136         136 ;
137 010024 137 MPSRV02:
138 010024 7E027310 138 BTST #DTRCVF,@RCVFLAG;DATA receive ?
139 010028 47FA 139 BEQ MPSRV02 ;No

```

```

140                                140 ;
141 01002A 7F027210                141          BCLR      #DTRCVF,@RCVFLAG;Clear flag
142 01002E 3E01                    142          MOV.B    R6L,@RXDATA ;Store receive DATA
143 010030                          143 MP SRV99:
144 010030 40FE                    144          BRA      MP SRV99
145                                145 ;
146                                146 ;*****
147                                147 ;          SCI RX INTERRUPT
148                                148 ;*****
149          00010032                149 MP SCIRX: .EQU    $
150 010032 6DF0                    150          PUSH.W  R0          ;Save registers
151 010034 6DF1                    151          PUSH.W  R1
152                                152 ;
153 010036 2EB5                    153          MOV.B    @RDR0,R6L ;Load receive data
154 010038 7FB47260                154          BCLR      #RDRF,@SSR0 ;Clear RDRF
155                                155 ;
156 01003C 7EB47310                156          BTST    #MPB,@SSR0 ;MPB = 1 ?
157 010040 5870001A                157          BEQ     MPRX02 ;No
158                                158 ;
159 010044 AE01                    159          CMP.B    #MYID,R6L ;Receive ID = My ID ?
160 010046 5860000C                160          BNE     MPRX01 ;No
161                                161 ;
162 01004A 7FB27230                162          BCLR      #MPIE,@SCR0 ;Clear MPIE
163 01004E 7F027000                163          BSET     #IDRCVF,@RCVFLAG;Set IDRCVF
164 010052 58000010                164          BRA      MPRX99 ;Exit
165                                165 ;
166 010056                          166 MP RX01:
167 010056 7FB27030                167          BSET     #MPIE,@SCR0 ;Set MPIE
168 01005A 58000008                168          BRA      MPRX99 ;Exit
169                                169 ;
170 01005E                          170 MP RX02:
171 01005E 7F027010                171          BSET     #DTRCVF,@RCVFLAG;Set DTRCVF
172 010062 7FB27030                172          BSET     #MPIE,@SCR0 ;Set MPIE
173                                173 ;
174 010066                          174 MP RX99:
175 010066 6D71                    175          POP.W   R1
176 010068 6D70                    176          POP.W   R0
177 01006A 5670                    177          RTE
178                                178 ;
179                                179          .END
*****TOTAL ERRORS          0
*****TOTAL WARNINGS        0

```

2.14 ブロック転送

MCU : H8/3003

使用機器 : DMAC コントローラ (DMAC) (ブロック転送)

2.14.1 仕様

- 図 2.78 に示すように、外部信号の立ち下がりエッジ検出ごとに RAM に設定した 6 バイトデータを I/O ポートに転送し、16 ビットのデータ出力を行います。
- DMAC の起動は、外部信号の立ち下がりエッジ検出により起動します。

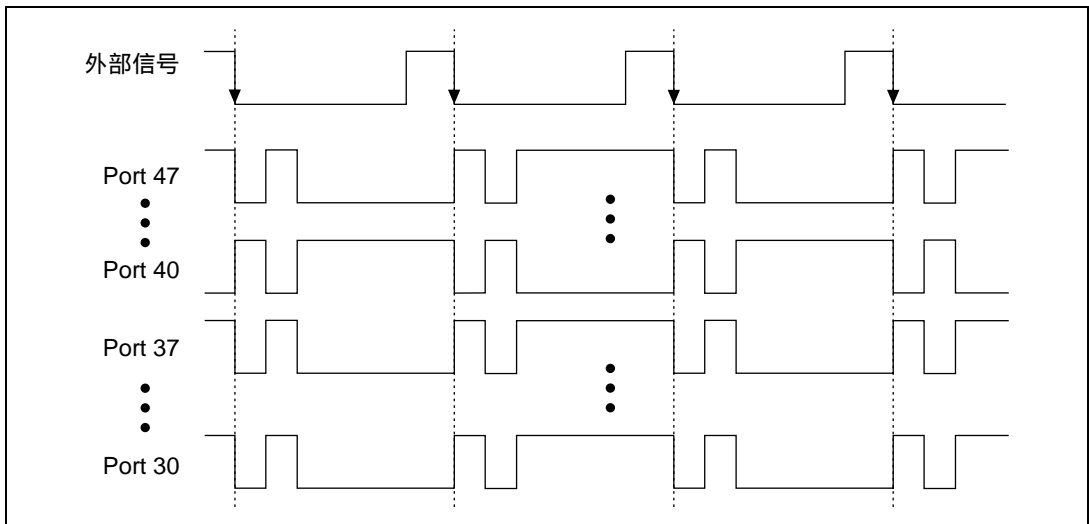


図 2.78 波形出力例

2.14.2 使用機能説明

本タスク例ではグループ0のDMAC/ch0A、ch0Bを使用してブロック転送を行います。

図2.79に本タスク例で使用するDMACブロック図を示します。本タスク例はDMACの以下の機能を使用してブロック転送を行います。

- 外部リクエストによりDMACを起動する機能（DREQによるDMAC起動）
- DMAC起動時にブロック単位でデータを転送する機能（ブロック転送モード）
- データ転送終了時に割り込みを発生する機能（DEND割り込み）

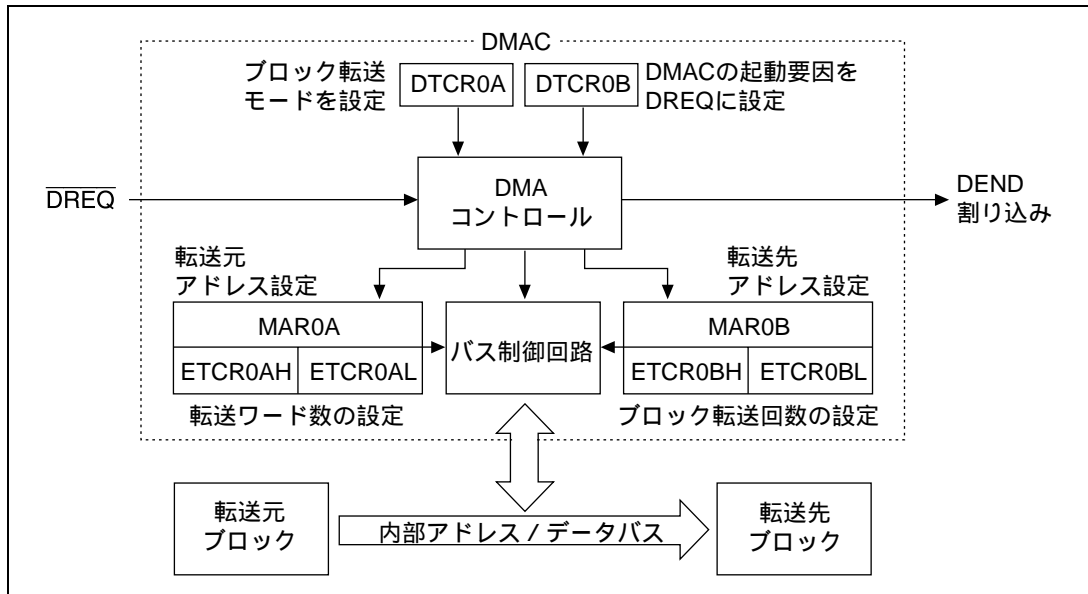


図 2.79 DMA コントローラブロック図

表 2.74 に本タスク例の機能割り付けを示します。表 2.74 に示すようにDMACの機能を割り付け、ブロック転送を行います。

表 2.74 DMAC 機能割り付け

DMAC 機能	機能
DREQ	DMAC 起動トリガとなる外部パルスを入力する
DTCR0A	DMAC をブロック転送モードに設定する
DTCR0B	DMAC 起動トリガを外部パルスの立ち下がりに設定する
MAR0A	転送元アドレスを設定する
MAR0B	転送先アドレスを設定する
ETCR0AH ETCR0AL	転送ワード数を設定する
ETCR0BH ETCR0BL	転送回数を設定する

2.14.3 動作説明

図 2.80 に動作原理を示します。図 2.80 に示すタイミングでハードウェア処理およびソフトウェア処理を行い、ブロック転送を実行します。

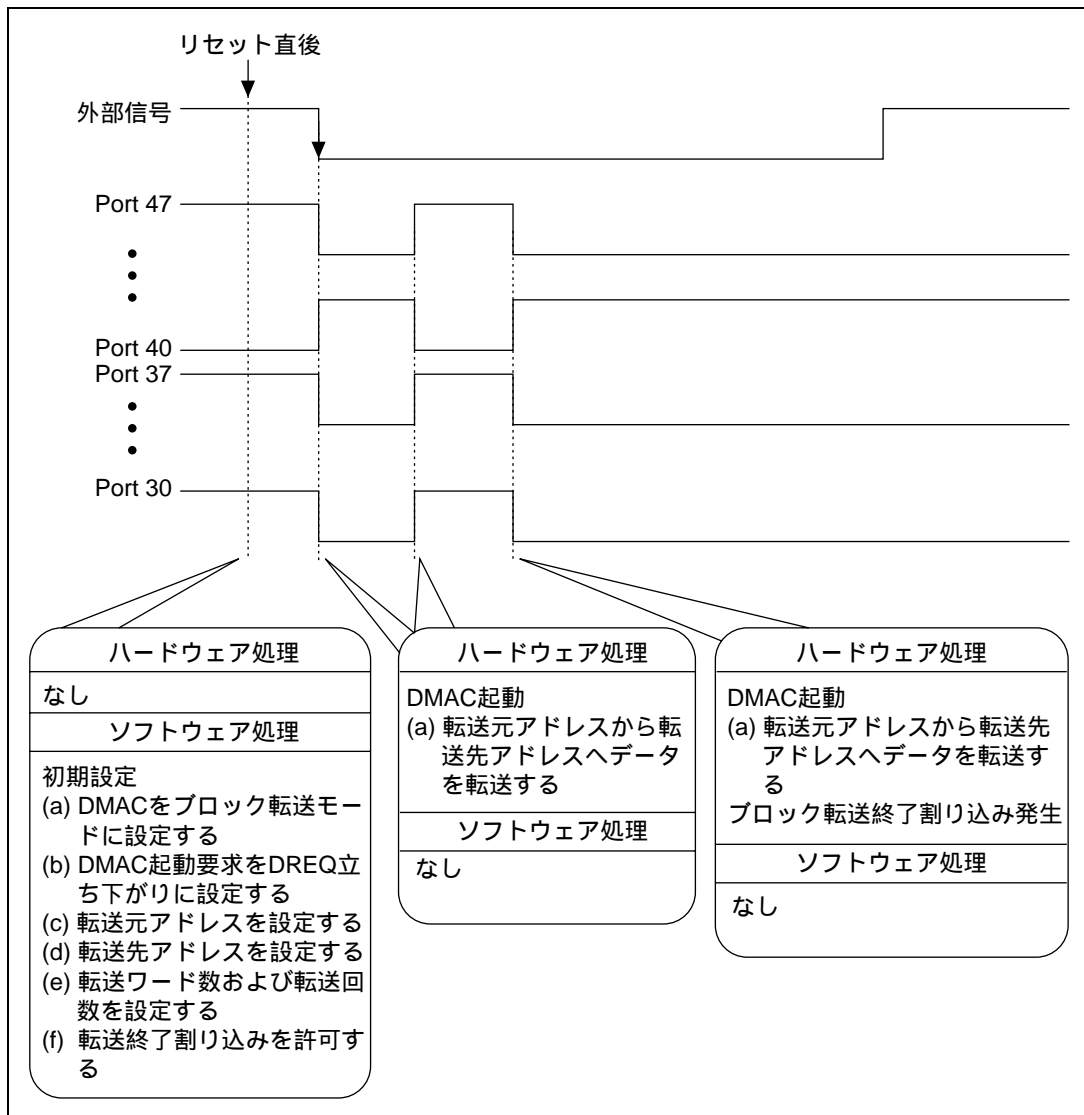


図 2.80 ブロック転送動作原理

2.14.4 ソフトウェア説明

表 2.75 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	BLKMN	DMAC の初期設定を行う。
ブロック転送完了	BLKEND	TEND 割り込みで起動し、DMAC の再設定を行う。

- 引き数の説明
本タスク例ではモジュール間の引き数はありません。

表 2.76 使用内部レジスタ説明

レジスタ名	機能	使用モジュール名
DTCR0A	DMAC をブロック転送モードに設定する。	メインルーチン
DTCR0B	DMAC 起動トリガを外部パルスの立ち下がりに設定する。	メインルーチン
MAR0A	転送元アドレスを設定する。	メインルーチン
MAR0B	転送先アドレスを設定する。	メインルーチン
ETCR0AH ETCR0AL	転送ワード数を設定する。	メインルーチン
ETCR0BH	転送回数を設定する。	メインルーチン

表 2.77 汎用レジスタ説明

モジュール名	レジスタ名	機能
メインルーチン	R0L	DMAC を設定するときのワークとして使用する。
	R0	

- 使用 RAM 説明
本タスク例ではRAMを使用しません。

2.14.5 フローチャート

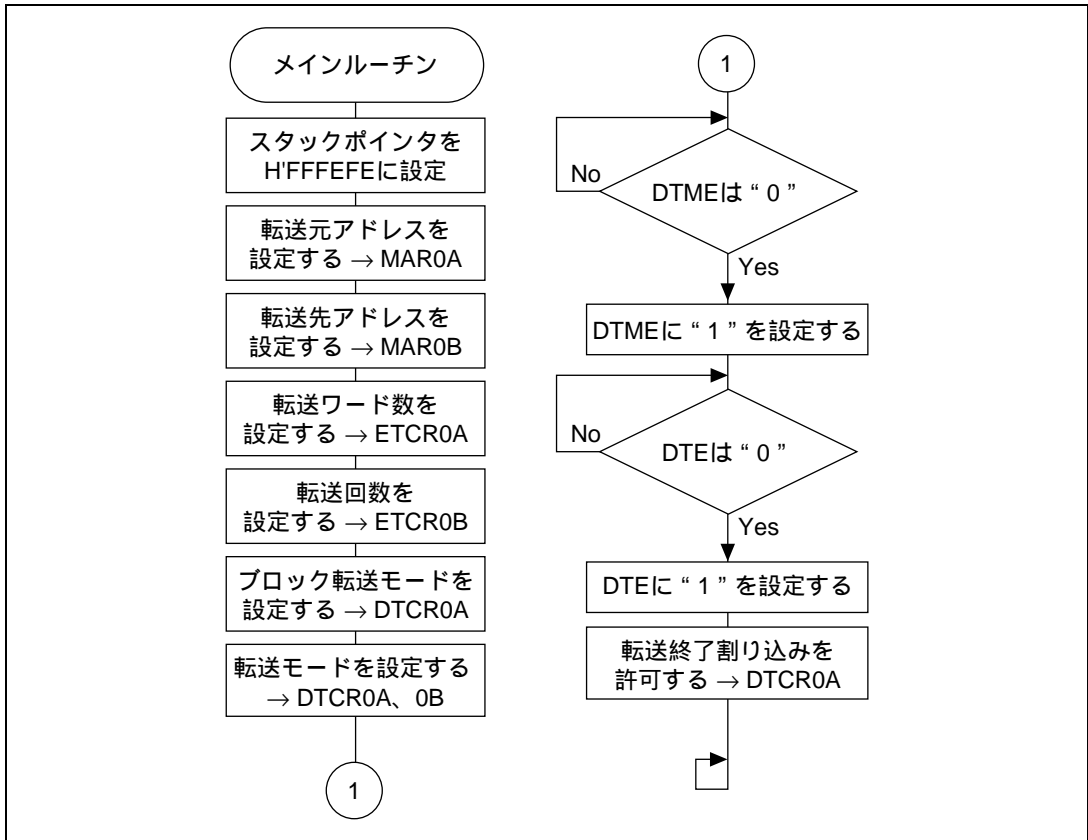


図 2.81 メインルーチン

2.14.6 プログラムリスト

```

1          1 ;*****
2          2 ;*
3          3 ;*          VECTOR ADDRESS
4          4 ;*
5          5 ;*****
6          6
7          7          .CPU      300HA
8 000000   8          .SECTION  VECT, CODE, LOCATE=H'000000
9          9 ;
10 000000 00010000 10 RES          .DATA.L  BLKMN
11          11 ;
12 00001C   12          .ORG      H'00001C
13 00001C 00010000 13 NMI          .DATA.L  BLKMN
14          14 ;
15 000030   15          .ORG      H'000030
16 000030 00010000 16 IRQ0         .DATA.L  BLKMN
17 000034 00010000 17 IRQ1         .DATA.L  BLKMN
18 000038 00010000 18 IRQ2         .DATA.L  BLKMN
19 00003C 00010000 19 IRQ3         .DATA.L  BLKMN
20 000040 00010000 20 IRQ4         .DATA.L  BLKMN
21 000044 00010000 21 IRQ5         .DATA.L  BLKMN
22          22 ;
23 000050   23          .ORG      H'000050
24 000050 00010000 24 WOVI        .DATA.L  BLKMN
25 000054 00010000 25 CMI          .DATA.L  BLKMN
26          26 ;
27 000060   27          .ORG      H'000060
28 000060 00010000 28 TGAI0       .DATA.L  BLKMN
29 000064 00010000 29 TGBB0       .DATA.L  BLKMN
30 000068 00010000 30 TOVI0       .DATA.L  BLKMN
31          31 ;
32 000070   32          .ORG      H'000070
33 000070 00010000 33 TGAI1       .DATA.L  BLKMN
34 000074 00010000 34 TGBI1       .DATA.L  BLKMN
35 000078 00010000 35 TOVI1       .DATA.L  BLKMN
36          36 ;
37 000080   37          .ORG      H'000080
38 000080 00010000 38 TGAI2       .DATA.L  BLKMN
39 000084 00010000 39 TGBI2       .DATA.L  BLKMN
40 000088 00010000 40 TOVI2       .DATA.L  BLKMN
41          41 ;
42 000090   42          .ORG      H'000090
43 000090 00010000 43 TGAI3       .DATA.L  BLKMN
44 000094 00010000 44 TGBI3       .DATA.L  BLKMN
45 000098 00010000 45 TOVI3       .DATA.L  BLKMN

```

2. 基礎編

```
46                                     46 ;
47 0000A0                             47 .ORG      H'0000A0
48 0000A0 00010000                    48 TGAI4    .DATA.L  BLKMN
49 0000A4 00010000                    49 TGBI4    .DATA.L  BLKMN
50 0000A8 00010000                    50 TOVI4    .DATA.L  BLKMN
51                                     51 ;
52 0000B0                             52 .ORG      H'0000B0
53 0000B0 00010000                    53 DEND0    .DATA.L  BLKMN
54 0000B4 00010000                    54 DEND1    .DATA.L  BLKMN
55 0000B8 00010000                    55 DEND2    .DATA.L  BLKMN
56 0000BC 00010000                    56 DEND3    .DATA.L  BLKMN
57                                     57 ;
58 0000D0                             58 .ORG      H'0000D0
59 0000D0 00010000                    59 ERI0     .DATA.L  BLKMN
60 0000D4 00010000                    60 RXI0     .DATA.L  BLKMN
61 0000D8 00010000                    61 TXI0     .DATA.L  BLKMN
62 0000DC 00010000                    62 TEND0    .DATA.L  BLKMN
63 0000E0 00010000                    63 ERI1     .DATA.L  BLKMN
64 0000E4 00010000                    64 RXI1     .DATA.L  BLKMN
65 0000E8 00010000                    65 TXI1     .DATA.L  BLKMN
66 0000EC 00010000                    66 TEND1    .DATA.L  BLKMN
67 0000F0 00010000                    67 ADI      .DATA.L  BLKMN
68                                     68 ;
69                                     69 ;*****
70                                     70 ;*
71                                     71 ;*      SYMBOL DEFINITIONS
72                                     72 ;*
73                                     73 ;*****
74                                     74 ;
75                                     75 ;***** DMAC0A *****
76 00FFFF20                            76 MAR0AR:  .EQU    H'FFFF20  ;Memory address register
77 00FFFF24                            77 ETCR0AH: .EQU    H'FFFF24  ;Transmit cunt register 'H'
78 00FFFF25                            78 ETCR0AL: .EQU    H'FFFF25  ;Transmit cunt register 'L'
79 00FFFF27                            79 DTCR0A:  .EQU    H'FFFF27  ;Data ransfer control
                                           ;register
80 00000007                            80 DTE:     .EQU    7          ;Data transfer enable bit
81 00000003                            81 DTIE    .EQU    3          ;Data transfer interrupt
                                           ;enable bit
82                                     82 ;***** DMAC0B *****
83 00FFFF28                            83 MAR0BR:  .EQU    H'FFFF28  ;Memory address register
84 00FFFF2C                            84 ETCR0BH: .EQU    H'FFFF2C  ;Transmit cunt register 'H'
85                                     85 ;
86 00FFFF2F                            86 DTCR0B   .EQU    H'FFFF2F  ;Data transfer control
                                           ;register
87 00000007                            87 DTME    .EQU    7          ;Data transfer master enable
                                           ;bit
88                                     88 ;***** PORT 9 *****
```

```

89      00FFFFFFD0      89 P9DDR:      .EQU      H'FFFFFFD0      ;Port 9 data direction
                                           ;register
90      00FFFFFFD2      90 P9DR:       .EQU      H'FFFFFFD2      ;Port 9 data register
91      ;***** PORT A *****
92      00FFFFFFD1      92 PADDR:      .EQU      H'FFFFFFD1      ;Port A data direction
                                           ;register
93      00FFFFFFD3      93 PADR:       .EQU      H'FFFFFFD3      ;Port A data register
94      ;
95      ;*****
96      ;*
97      ;*          MAIN PROGRAM : BLKMN
98      ;*
99      ;*****
100     ;
101     010000          101          .SECTION  PROG, CODE, LOCATE=H'010000
102     ;
103     00010000      103 BLKMN:     .EQU      $
104     010000 7A0700FFFE 104          MOV.L     #H'FFFEFE, SP      ;Set stack pointer
105     010006 F8FF      105          MOV.B     #H'FF, R0L
106     010008 38D0      106          MOV.B     R0L, @P9DDR      ;Port 9 is output port
107     01000A 38D1      107          MOV.B     R0L, @PADDR      ;Port A is output port
108     ;
109     01000C 7A0000010054 109          MOV.L     #SRCADD, ER0      ;Set source address
110     010012 01006B80FF20 110          MOV.L     ER0, @MAR0AR      ;
111     010018 7A0000FFFFFFD2 111          MOV.L     #P9DR, ER0      ;Set distnation address
112     01001E 01006B80FF28 112          MOV.L     ER0, @MAR0BR      ;
113     ;
114     010024 79000002      114          MOV.W     #2, R0          ;Set transfer count=2
115     010028 6B80FF2C      115          MOV.W     R0, @ETCR0BH
116     ;
117     01002C F803      117          MOV.B     #H'03, R0L      ;Set block size = 3word
118     01002E 3824      118          MOV.B     R0L, @ETCR0AH
119     010030 3825      119          MOV.B     R0L, @ETCR0AL
120     ;
121     010032 F80E      121          MOV.B     #B'00001110, R0L ;Distnatino address fix
122     010034 382F      122          MOV.B     R0L, @DTCR0B      ;DREQ falling edge
                                           ;request
123     ;
124     010036 F857      124          MOV.B     #B'01010111, R0L ;Block trans.mode, word
                                           ;size
125     010038 3827      125          MOV.B     R0L, @DTCR0A      ;Source address increment
126     ;
127     01003A          127 BLKMN01:
128     01003A 7E2F7370      128          BTST     #DTME, @DTCR0B      ;Is DTME 'Low' ?
129     01003E 46FA      129          BNE     BLKMN01          ;No
130     ;
131     010040 7F2F7070      131          BSET     #DTME, @DTCR0B      ;Set DTME

```

2. 基礎編

```
132                                132 ;
133 010044                        133 BLKMN02:
134 010044 7E277370              134          BTST    #DTE,@DTCR0A    ;DTE is 'Low' ?
135 010048 46FA                   135          BNE     BLKMN02        ;No
136                                136 ;
137 01004A 7F277070              137          BSET    #DTE,@DTCR0A    ;Set DTE
138 01004E 7F277030              138          BSET    #DTIE,@DTCR0A  ;Enable DEND0 interrupt
139                                139 ;
140 010052                        140 BLKMN99:
141 010052 40FE                   141          BRA     BLKMN99
142                                142 ;
143                                143 ;*****
144                                144 ;*
145                                145 ;*      DATA TABLE (OUTPUT PATTERN)
146                                146 ;*
147                                147 ;*****
148          00010054              148 SRCADD:    .EQU    $
149 010054 1111                   149          .DATA.W  H'1111
150 010056 2222                   150          .DATA.W  H'2222
151 010058 3333                   151          .DATA.W  H'3333
152 01005A 4444                   152          .DATA.W  H'4444
153                                153 ;
154                                154          .END
****TOTAL ERRORS                0
****TOTAL WARNINGS              0
```

3. 応用編

3.1 高速データ出力

MCU : H8/3003

使用機器 : ITU、TPC、DMAC

3.1.1 仕様

図 3.1 に示すように外部信号の立ち上がりエッジを検出することにより、12 ビットのデータを出力します。

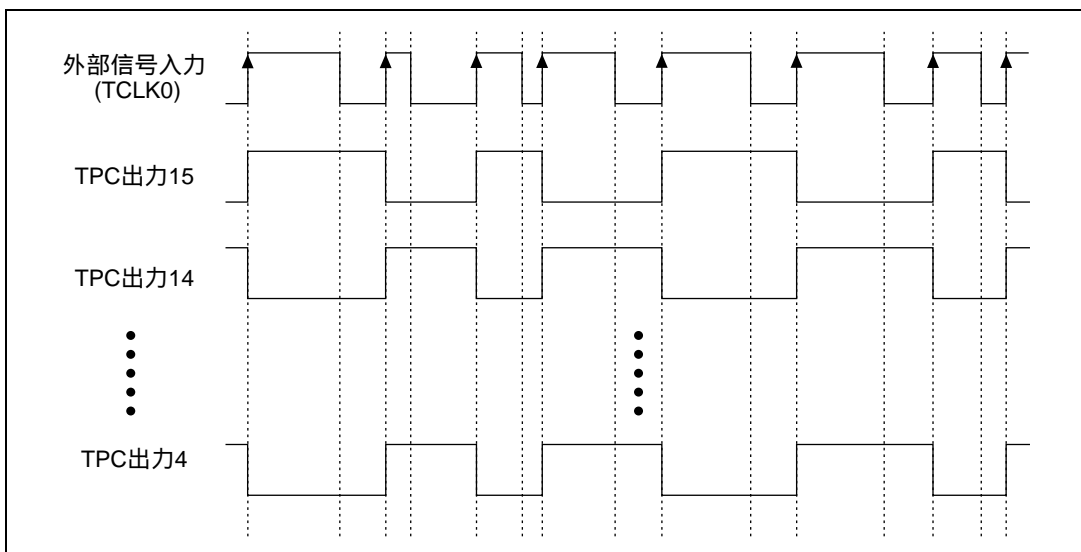


図 3.1 12 ビットデータ出力例

3.1.2 考え方

図 3.2 に本タスクで使用する内蔵機能のブロック図を示します。

外部信号が入力されるごとに ITU がコンペアマッチ信号を発生させ DMAC を起動します。

DMAC は出力パターンデータテーブルの値を TPC に転送します。

上記の動作を繰り返すことにより、外部から立ち上がりエッジに応じたりアルタイムポート制御が可能です。

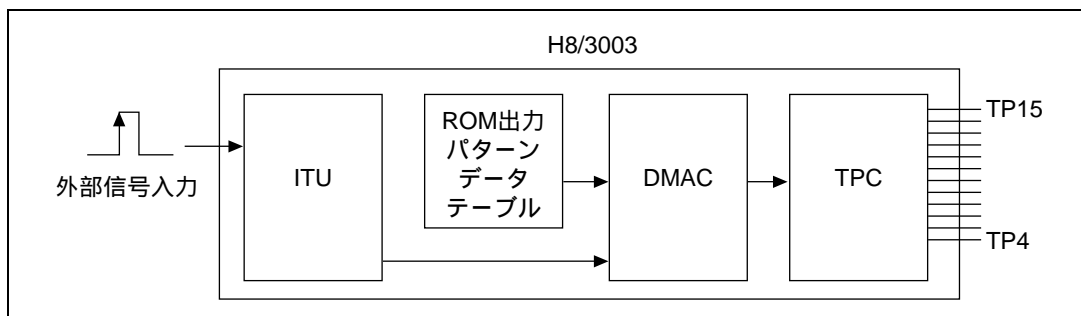


図 3.2 H8/3003 ブロック図

3.1.3 使用機能説明

図 3.3 に本タスクで使用する内蔵機能のブロック図を示します。

本タスク例は、H8/3003 の機能を以下に示すように使用して高速にデータを出力します。

- 出力パターンデータテーブル :
TPCから出力するデータのパターンをROM上に設定します。
- ITU :
コンペアマッチAが発生するごとにタイマカウンタのクリアを行います。また、DMAC0およびTPCを起動します。(GRAにはH'0001を設定し、外部クロックの立ち上がりエッジでカウントを行います)
- DMAC0A :
ITUのコンペアマッチAで起動し、出力パターンデータテーブルから出力データをNDRに転送します。
- TPC :
ITUのコンペアマッチAで起動し、DMACを使用することによりCPUの介在なくデータを出力します。最大16ビットデータを出力することが可能ですが、本タスクでは12ビットデータを出力します。

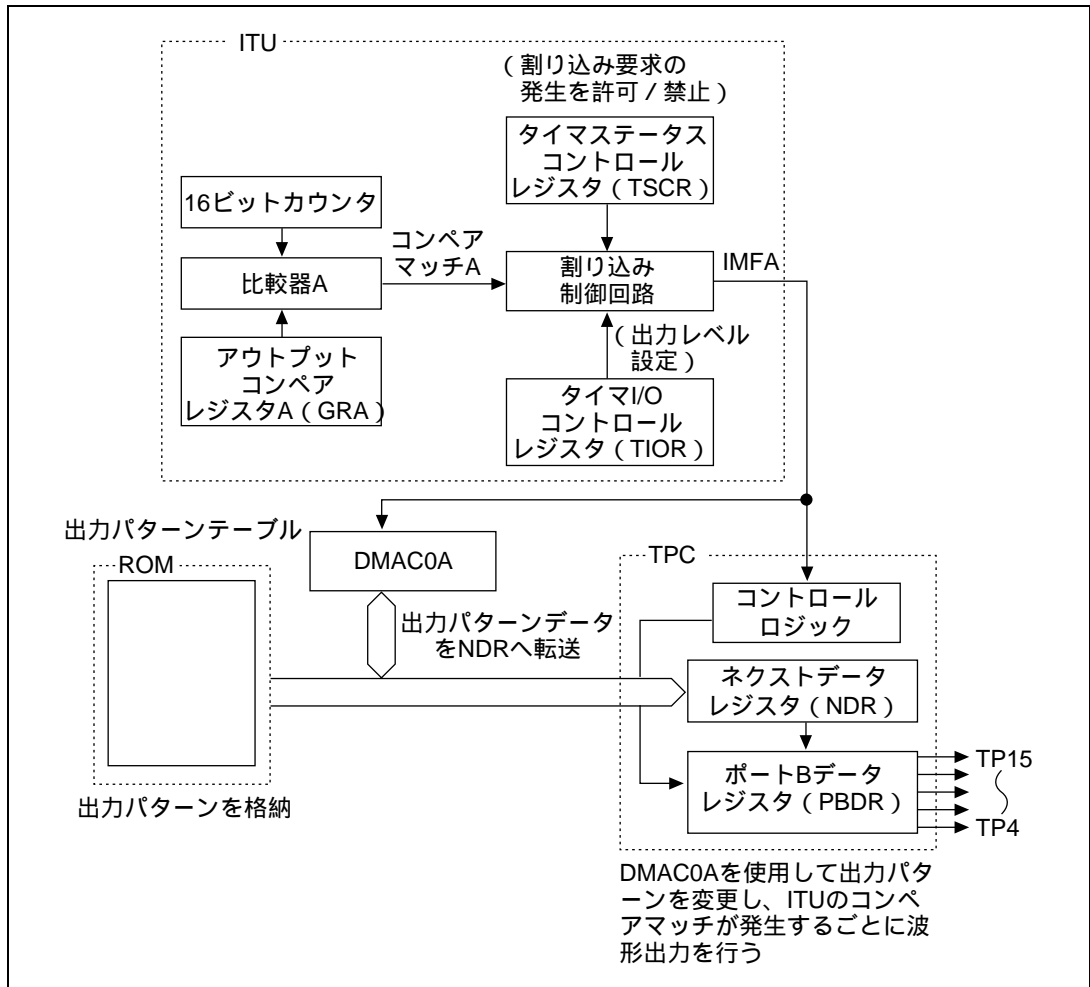


図 3.3 高速データ出力ブロック図

3. 応用編

表 3.1 に本タスク例の機能割り付けを示します。表 3.1 に示すように内蔵 I/O の機能を割り付け、高速データ出力を行います。

表 3.1 H8/3003 機能割り付け

H8/3003 機能		機能
ITU (ch0)	TCLK0	外部信号の立ち上がりエッジを検出
	TCNT0	16 ビットカウンタ
	GRA0	アウトプットコンペアレジスタ
	TCR0	カウントクロックの選択、カウンタクリア要因の選択
	TIOR0	GRA をアウトプットコンペアレジスタに設定
	TSR0	コンペアマッチやオーパフローのステータスを示す
	TIER0	割り込み許可 / 禁止を選択
	TSTR	TCNT のカウント動作の許可 / 禁止を設定
DMAC0A	DTCR0A	DMAC の動作を制御する
	MAR0A	出力パターンデータテーブルの先頭アドレスを設定
	IORA0A	NDR のアドレスを設定
	ETCR0A	転送回数を設定
TPC	PADDR	TPC 出力端子を設定
	PADR	TPC 出力 TP7 ~ TP14 の出力データを格納
	PBDDR	TPC 出力端子を設定
	PBDR	TPC 出力 TP15 ~ TP8 の出力データを格納
	TPMR	TPC のノンオーバーラップモードを選定
	TPCR	TPC 出力の出力トリガ信号を選定
	NDERB	TPC 出力 TP15 ~ TP8 の許可 / 禁止を設定
	NDERA	TPC 出力 TP7 ~ TP4 の許可 / 禁止を設定
	NDRB	TPC 出力 TP15 ~ TP8 の次に出力するデータを格納
	NDRA	TPC 出力 TP7 ~ TP4 の次に出力するデータを格納
	TP15 ~ TP4	TPC データ出力端子

3.1.4 動作説明

図 3.4 に動作原理を示します。図 3.4 に示すように H8/3003 のハードウェア処理およびソフトウェア処理により高速にデータの出力を行います。

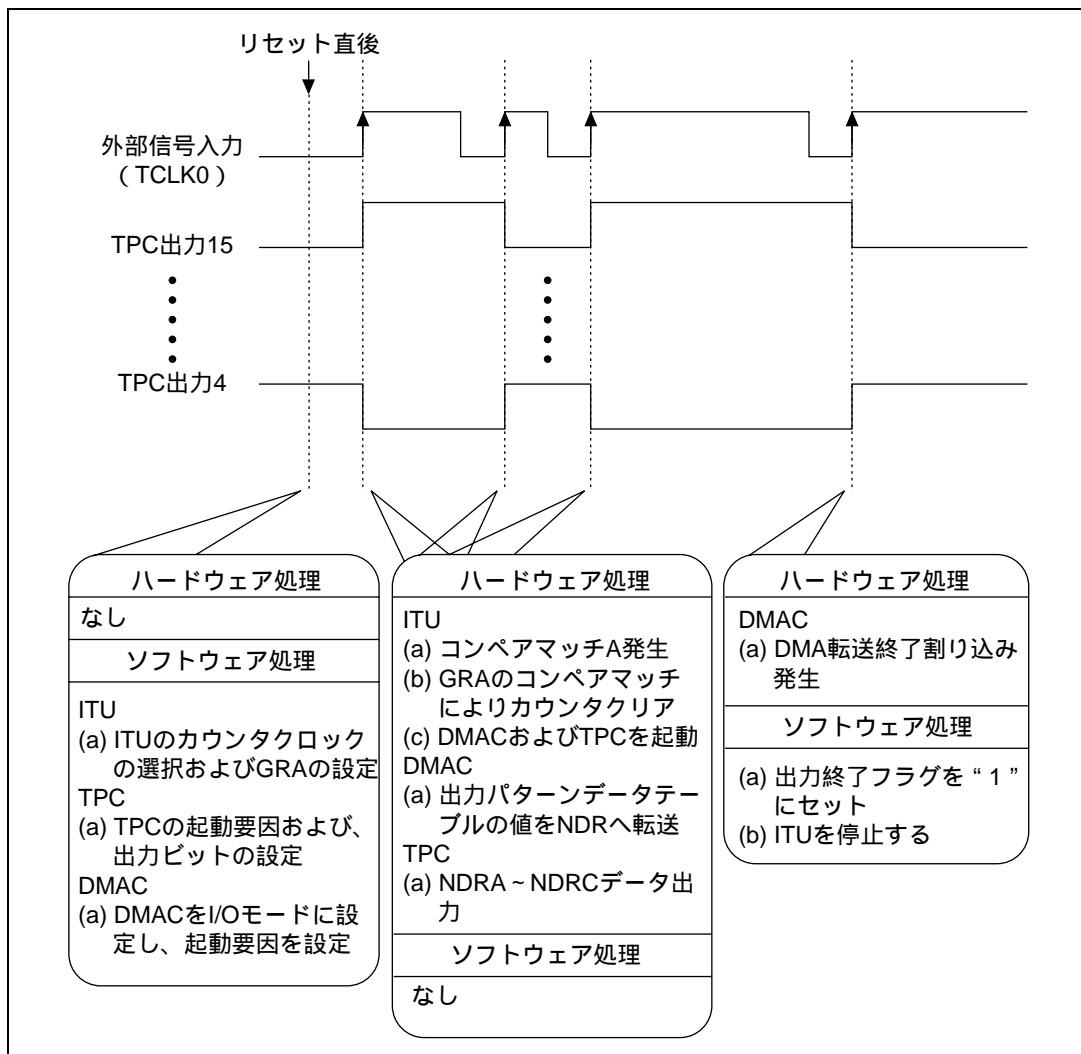


図 3.4 高速データ出力動作原理

3.1.5 ソフトウェア説明

表 3.2 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	STPCMN	ITU、TPC および DMAC0A の初期設定を行う。
DMA 割り込み	DMAEND	出力終了フラグをセットする。

表 3.3 引き数の説明

ラベル名、レジスタ名	機能	データ長	使用モジュール名	入出力
DMA_FL A	出力がすべて終了したことを示すフラグ 0：全ビット出力未終了 1：全ビット出力終了	1 ビット	メインルーチン	入出力
			DMA 割り込み	入出力

表 3.4 使用内部レジスタ

内蔵機能	レジスタ名	機能
ITU	GRA	アウトプットコンペアの値 (H'01) を設定
	TCR	ITU を以下のように設定する <ul style="list-style-type: none"> GRA のコンペアマッチでカウンタクリア 外部信号の立ち上がりエッジでカウント TCLKA 端子でカウント
	TIOR	GRA をアウトプットコンペアレジスタに設定し、端子出力を禁止
	TIER	IMFA 割り込みを許可
	TSTR	TCNT0 のカウント動作を開始する
TPC	PADDR	TP7 ~ TP4 までの TPC 出力を許可
	PBDDR	TP15 ~ TP8 までの TPC 出力を許可
	PADR	TP7 ~ TP4 までの出力データを格納
	PBDR	TP15 ~ TP8 までの出力パターンを格納
	TPMR	TP15 ~ TP4 までを通常動作に設定
	TPCR	TP15 ~ TP4 の出力トリガを ITU0 のコンペアマッチに設定
	NDERA	TPC 出力 TP7 ~ TP4 を許可
	NDERB	TPC 出力 TP15 ~ TP8 を許可
	NDRA	次の出力パターンデータを格納する
NDRB	次の出力パターンデータを格納する	
DMAC0A	DTCR0A	DMAC0A を以下のように設定する <ul style="list-style-type: none"> データサイズをワードサイズに設定 MAR をインクリメントに設定 データ転送を I/O モードに設定 起動要因を ITUch0 のコンペアマッチ A に設定
	MAR0A	出力パターンデータテーブルの転送元アドレスを設定
	IOAR0A	NDRB のアドレス (転送先) を設定する
	ETCR0A	転送回数を設定

- 使用 RAM 説明
本タスクではRAMを使用しません。

表 3.5 データテーブルの説明

テーブル名	機能	データ長	データ容量
OPAT_TAB	TPC より出力するデータを格納	1ワード	15ワード

- 汎用レジスタ説明

	31	ERn	RnH	RnL	0
0	データ設定時のワークとして使用				
1					
2					
3					
4					
5					
6					
7					

図 3.5 汎用レジスタ説明

3.1.6 フローチャート

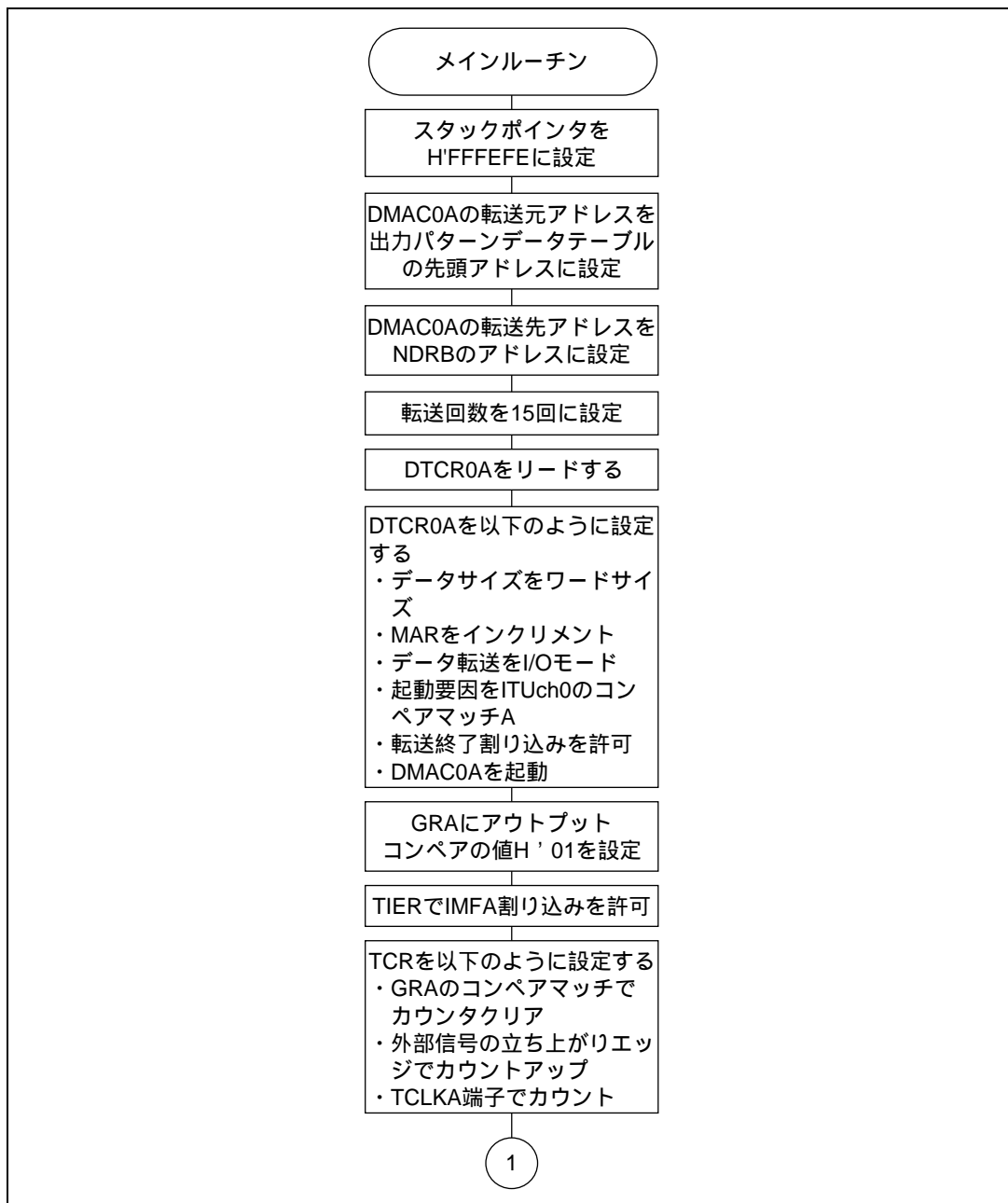


図 3.6 メインルーチン (1)

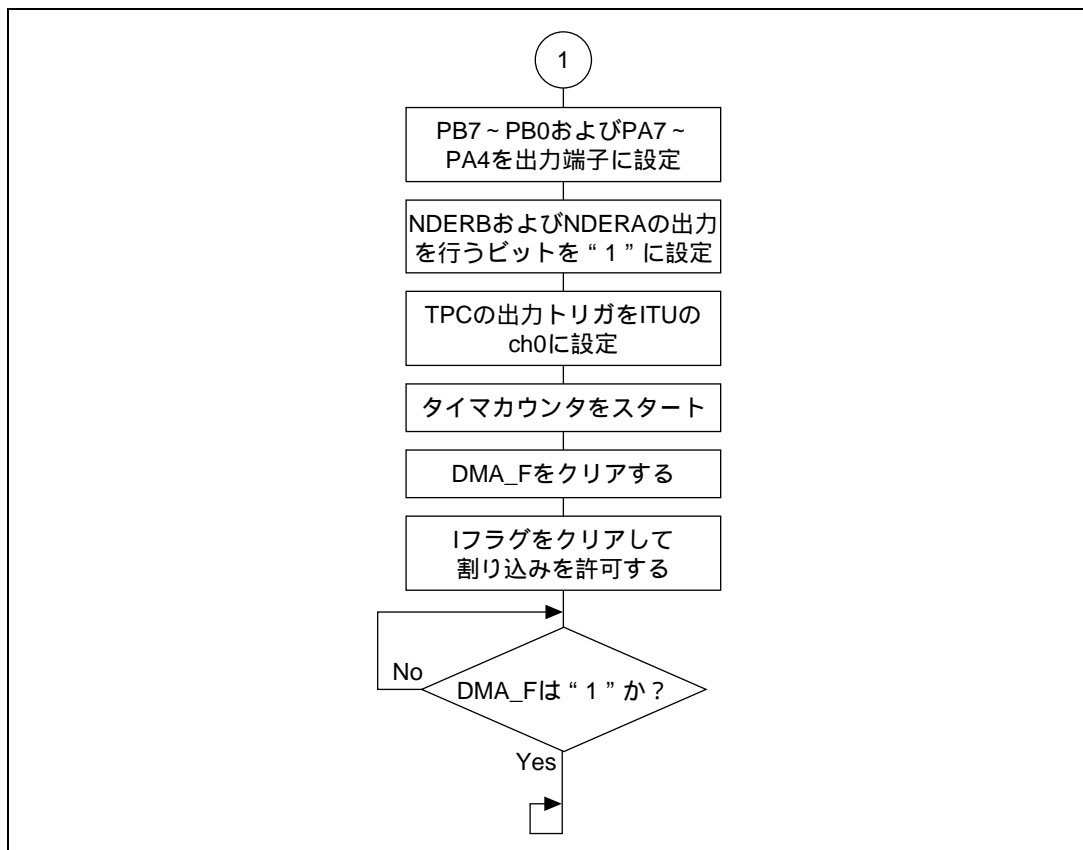


図 3.6 メインルーチン (2)



図 3.7 DMA 割り込み

3.1.7 プログラムリスト

```

1          1 ;*****
2          2 ;*
3          3 ;*          VECTOR ADDRESS
4          4 ;*
5          5 ;*****
6          6
7          7          .CPU      300HA
8 000000    8          .SECTION VECT, CODE, LOCATE=H'000000
9          9 ;
10 000000 00010000 10 RES          .DATA.L  STPCMN
11          11 ;
12 00001C    12          .ORG      H'00001C
13 00001C 00010000 13 NMI          .DATA.L  STPCMN
14          14 ;
15 000030    15          .ORG      H'000030
16 000030 00010000 16 IRQ0         .DATA.L  STPCMN
17 000034 00010000 17 IRQ1         .DATA.L  STPCMN
18 000038 00010000 18 IRQ2         .DATA.L  STPCMN
19 00003C 00010000 19 IRQ3         .DATA.L  STPCMN
20 000040 00010000 20 IRQ4         .DATA.L  STPCMN
21 000044 00010000 21 IRQ5         .DATA.L  STPCMN
22 000048 00010000 22 IRQ6         .DATA.L  STPCMN
23 00004C 00010000 23 IRQ7         .DATA.L  STPCMN
24 000050 00010000 24 WOVI0        .DATA.L  STPCMN
25 000054 00010000 25 CMI          .DATA.L  STPCMN
26          26 ;
27 000060    27          .ORG      H'000060
28 000060 00010000 28 IMIA0        .DATA.L  STPCMN
29 000064 00010000 29 IMIB0        .DATA.L  STPCMN
30 000068 00010000 30 OVI0         .DATA.L  STPCMN
31          31 ;
32 000070    32          .ORG      H'000070
33 000070 00010000 33 IMIA1        .DATA.L  STPCMN
34 000074 00010000 34 IMIB1        .DATA.L  STPCMN
35 000078 00010000 35 OVI1         .DATA.L  STPCMN
36          36 ;
37 000080    37          .ORG      H'000080
38 000080 00010000 38 IMIA2        .DATA.L  STPCMN
39 000084 00010000 39 IMIB2        .DATA.L  STPCMN
40 000088 00010000 40 OVI2         .DATA.L  STPCMN
41          41 ;
42 000090    42          .ORG      H'000090
43 000090 00010000 43 IMIA3        .DATA.L  STPCMN
44 000094 00010000 44 IMIB3        .DATA.L  STPCMN
45 000098 00010000 45 OVI3         .DATA.L  STPCMN
46          46 ;

```

3. 応用編

```

47 0000A0          47          .ORG      H'0000A0
48 0000A0 00010000 48 IMIA4     .DATA.L  STPCMN
49 0000A4 00010000 49 IMIB4     .DATA.L  STPCMN
50 0000A8 00010000 50 OVI4     .DATA.L  STPCMN
51                51 ;
52 0000B0          52          .ORG      H'0000B0
53 0000B0 0001005C 53 DEND0A    .DATA.L  DMAEND
54 0000B4 00010000 54 DEND0B    .DATA.L  STPCMN
55 0000B8 00010000 55 DEND1A    .DATA.L  STPCMN
56 0000BC 00010000 56 DEND1B    .DATA.L  STPCMN
57 0000C0 00010000 57 DEND2A    .DATA.L  STPCMN
58 0000C4 00010000 58 DEND2B    .DATA.L  STPCMN
59 0000C8 00010000 59 DEND3A    .DATA.L  STPCMN
60 0000CC 00010000 60 DEND3B    .DATA.L  STPCMN
61 0000D0 00010000 61 ERI0     .DATA.L  STPCMN
62 0000D4 00010000 62 RXI0     .DATA.L  STPCMN
63 0000D8 00010000 63 TXI0     .DATA.L  STPCMN
64 0000DC 00010000 64 TEI0     .DATA.L  STPCMN
65 0000E0 00010000 65 ERI1     .DATA.L  STPCMN
66 0000E4 00010000 66 RXI1     .DATA.L  STPCMN
67 0000E8 00010000 67 TXI1     .DATA.L  STPCMN
68 0000EC 00010000 68 TEI1     .DATA.L  STPCMN
69 0000F0 00010000 69 ADI      .DATA.L  STPCMN
70                70 ;
71                71 ;*****
72                72 ;*
73                73 ;*          RAM ALLOCATION
74                74 ;*
75                75 ;*****
76                76 ;
77 FFFF00          77          .SECTION  RAM,DATA,LOCATE=H'FFFF00
78                78 ;
79 FFFF00 00000001 79 DMA_FL A: .RES.B  1          ;Flag area
80          00000000 80 DMA_ENDF: .EQU    0          ;DMA end flag
81                81 ;
82                82 ;*****
83                83 ;*
84                84 ;*          SYMBOL DEFINITIONS
85                85 ;*
86                86 ;*****
87                87 ;
88                88 ;***** DMAC0A *****
89          00FFFF20 89 MAR0AR:   .EQU    H'FFFF20    ;Memory address register
90          00FFFF24 90 ETCR0AH:  .EQU    H'FFFF24    ;Transmit cunt register
91          00FFFF26 91 IOAR0A:   .EQU    H'FFFF26    ;I/O address register
92          00FFFF27 92 DTCR0A:   .EQU    H'FFFF27    ;Data transfer control
                                     ;register

```

```

93      00000007      93 DTE:      .EQU      7      ;Data transfer enable bit
94
95      ;***** ITU *****
96      00FFFF60      96 TSTR:      .EQU      H'FFFF60      ;Timer start register
97
98      ;***** ITU ch0 *****
99      00FFFF64      99 TCR0:      .EQU      H'FFFF64      ;timer control register
100     00FFFF65      100 TIOR0:     .EQU      H'FFFF65      ;timer I/O control register
101     00FFFF66      101 TIER0:     .EQU      H'FFFF66      ;timer interrupt enable
;register
102     00FFFF67      102 TSR0      .EQU      H'FFFF67      ;timer status register
103     00FFFF6A      103 GRA0H:     .EQU      H'FFFF6A      ;General register
104
105     ;***** TPC *****
106     00FFFA0      106 TPMP:      .EQU      H'FFFA0      ;TPC output mod register
107     00FFFA1      107 TPCR:      .EQU      H'FFFA1      ;TPC output control register
108     00FFFA2      108 NDERB:     .EQU      H'FFFA2      ;Next data enable register B
109     00FFFA3      109 NDERA:     .EQU      H'FFFA3      ;Next data enable register A
110     00FFFD1      110 PADDR:     .EQU      H'FFFD1      ;Port A data direction
;register
111     00FFFD4      111 PBDDR:     .EQU      H'FFFD4      ;Port B data direction
;register

112
113     ;*****
114     ;*
115     ;*      MAIN PROGRAM : STPCMN      *
116     ;*
117     ;*****
118
119     010000      119      .SECTION  PROG, CODE, LOCATE=H'010000
120
121     00010000      121 STPCMN:     .EQU      $
122     010000 7A0700FFFE 122      MOV.L     #H'FFFEFE, SP ;Init. stack pointer
123
124     ;***** INITIALZE ITU *****
125     010006 79000000      125      MOV.W     #H'0, R0
126     01000A 6B80FF6A      126      MOV.W     R0, @GRA0H      ;Set non overlap time
127     01000E F801      127      MOV.B     #H'01, R0L
128     010010 3866      128      MOV.B     R0L, @TIER0      ;Init. TIER0(1->IMIEA)
129     010012 F824      129      MOV.B     #H'24, R0L
130     010014 3864      130      MOV.B     R0L, @TCR0      ;Init. TCR0(1->CCLR0, TPSC2)
131
132     ;***** INITIALZE DMAC0A *****
133     010016 7A0000010076 133      MOV.L     #OPAT_TAB, ERO
134     01001C 01006B80FF20 134      MOV.L     ERO, @MAR0AR      ;Set MAR0A
135     010022 F8A4      135      MOV.B     #H'A4, R0L
136     010024 3826      136      MOV.B     R0L, @IOAR0A      ;Set IOAR0A

```

3. 応用編

```

137 010026 7900000F      137          MOV.W      #H'0F,R0
138 01002A 6B80FF24      138          MOV.W      R0,@ETCR0AH      ;Store transfer count
139 01002E 2827          139          MOV.B      @DTCR0A,R0L      ;Read DTCR0A
140 010030 F848          140          MOV.B      #H'48,R0L
141 010032 3827          141          MOV.B      R0L,@DTCR0A      ;Init. DTCR0A
142 010034 7F277070      142          BSET      #DTE,@DTCR0A      ;Start DMAC0A
143                          143          ;
144                          144          ;***** INITIALZE TPC *****
145 010038 F8F0          145          MOV.B      #H'F0,R0L
146 01003A 38D1          146          MOV.B      R0L,@PADDR      ;Init. PADDR
147 01003C F8FF          147          MOV.B      #H'FF,R0L
148 01003E 38D4          148          MOV.B      R0L,@PBDDR      ;Init. PBDDR
149 010040 38A2          149          MOV.B      R0L,@NDERB      ;Set output TPC
150 010042 F8F0          150          MOV.B      #H'F0,R0L
151 010044 38A3          151          MOV.B      R0L,@NDERA      ;Set output TPC
152 010046 F800          152          MOV.B      #H'00,R0L
153 010048 38A1          153          MOV.B      R0L,@TPCR      ;Init. TPCR
154                          154          ;
155 01004A F801          155          MOV.B      #H'01,R0L
156 01004C 3860          156          MOV.B      R0L,@TSTR      ;Start ITU ch0
157 01004E 7F007200      157          BCLR      #DMA_ENDF,@DMA_FL ;Clear DMA_ENDF
158 010052 0700          158          LDC.B     #0,CCR          ;Enable interrupt
159                          159          ;
160 010054 7E007300      160 STPCMN01: BTST      #DMA_ENDF,@DMA_FL ;DMA_ENDF = "1" ?
161 010058 47FA          161          BEQ      STPCMN01      ;No
162 01005A 40FE          162 STPCMN99: BRA      STPCMN99      ;Loop
163                          163          ;
164                          164          ;*****
165                          165          ;*
166                          166          ;*   NAME : DMAEND (End of DMAC0A transmit)
167                          167          ;*
168                          168          ;*****
169                          169          ;*
170                          170          ;*   ENTRY  : NOTHING
171                          171          ;*   RETURN : DMA_ENDF (DMAC end flag)
172                          172          ;*
173                          173          ;*****
174                          174          ;
175          0001005C      175 DMAEND:   .EQU      $
176 01005C 6DF0          176          PUSH.W    R0
177 01005E 7F607200      177          BCLR.B   #0,@TSTR      ;Stop ITU ch0
178 010062 7F667200      178          BCLR.B   #0,@TIER0     ;Disable timer
179          010066      179          BCLR.B   #0,@TSR0     ;clear timer
180          01006A      180          BCLR.B   #3,@DTCR0A   ;comparemech interrupt
181          01006E      181          BCLR.B   #3,@DTCR0A   ;comparemech flag
181 01006E 7F007000      181          BSET      #DMA_ENDF,@DMA_FL ;Set DMA_ENDF

```

```
182 010072 6D70          182          POP.W      R0
183 010074 5670          183          RTE
184                      184 ;
185                      185 ;*****
186                      186 ;*                               *
187                      187 ;*          DATA TABLE (OUTPUT PATTERN)      *
188                      188 ;*                               *
189                      189 ;*****
190 010076 1110222033304440 190 OPAT_TAB   .DATA.W   H'1110,H'2220,H'3330,H'4440,H'5550,
191 01007E 5550666077708880 191 *          H'6660,H'7770,H'8880,H'9990,H'AAA0,
192 010086 9990AAA0BBB0CCCO 192 *          H'BBB0,H'CCC0,H'DDD0,H'EEE0,H'FFF0
      01008E DDD0EEE0FFF0
193                      193 ;
194                      194          .END
*****TOTAL ERRORS      0
*****TOTAL WARNINGS    0
```

3.2 SCI 連続送受信

MCU : H8/3003

使用機器 : ITU、TPC、DMAC

3.2.1 仕様

- H8/3003 の SCI をクロック同期式モードに設定し、H8/3003 と 100 バイトのデータを連続送信および連続受信します。
- DMAC を使用し、CPU を介さずにデータをメモリ TDR、RDR メモリへ設定します。
- 送信側がクロックマスタになります。

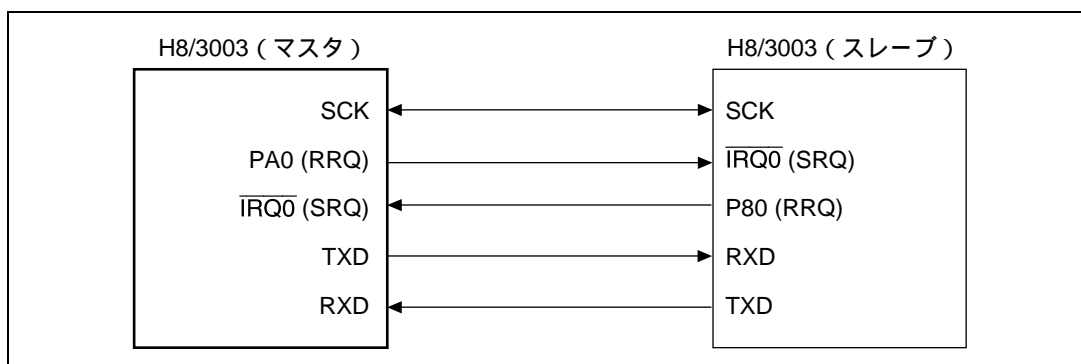


図 3.8 H8/3003 によるクロック同期式 SCI インタフェースブロック図

3.2.2 考え方

I/O ポートで制御する信号線の考え方を以下に示します。

- RRQ 信号 (レシーブ・リクエスト)
受信要求をスレーブ側に送信します。受信要求を送信した後、100 μ s 後に SCK からシリアルクロック、TXD からシリアルデータをスレーブ側に送信します。
- SRQ 信号 (センド・リクエスト)
送信要求を受信します。送信要求を受信した後、100 μ s 後に SCK からシリアルクロック、RXD からシリアルデータをスレーブ側に受信します。

3.2.3 使用機能説明

図 3.9 に本タスクで使用する H8/3003 の内蔵機能を示します。

図 3.9 に示すように DMAC および SCI を使用して高速にシリアル通信を行います。

- 送信用データテーブル：送信データを ROM 上に設定します。
- 受信データテーブル：受信データを RAM 設定します。
- DMAC0A
 - 送信時：SCI 送信完了割り込みで起動し、SCI データテーブルの内容を SCI に転送します。
 - 受信時：SCI 受信完了割り込みで起動し、受信データを SCI データテーブルに転送します。
- SCI：H8/3003 とシリアルデータの送信および受信を行います。

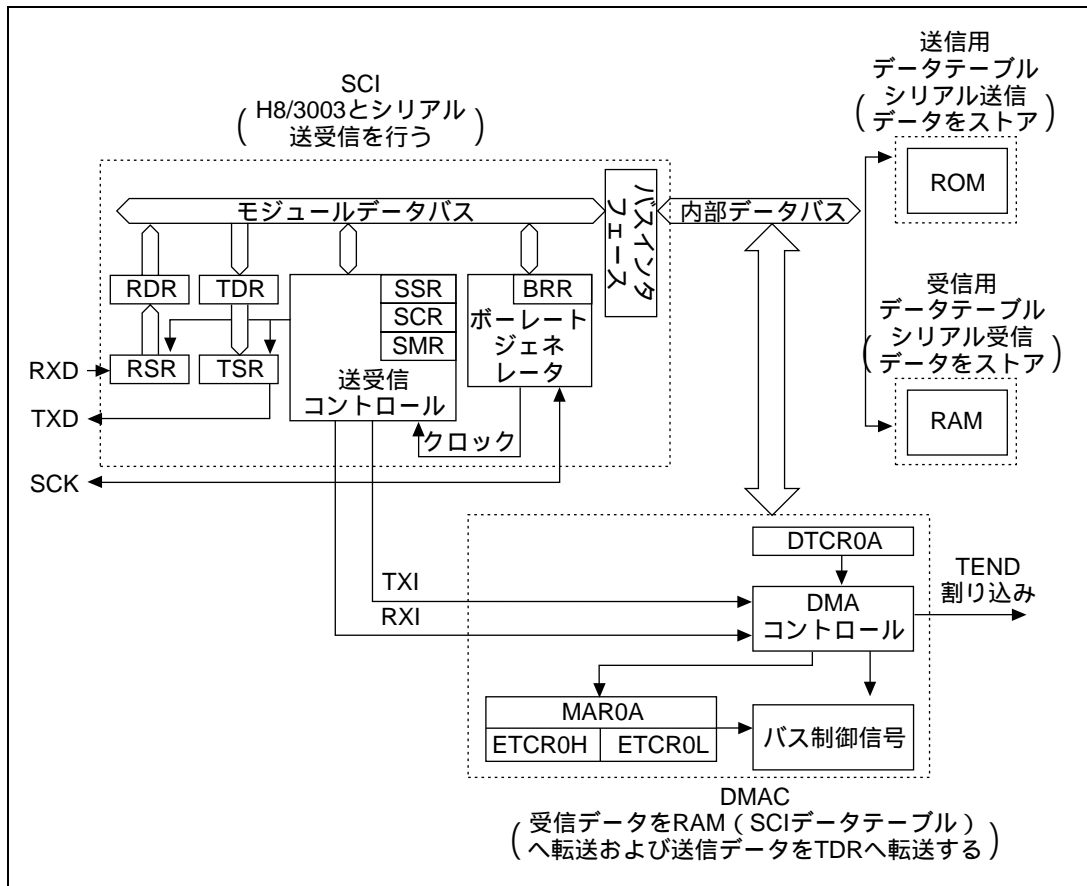


図 3.9 SCI および DMAC ブロック図

3. 応用編

表 3.6 に本タスク例の機能割り付けを示します。表 3.6 に示すように SCI および DMAC の機能を割り付け、H8/3003 とインタフェースします。

表 3.6 H8/3003 機能割り付け

H8/3003 機能		機能
SCI	SCK	転送クロックを送信する。また、受信時には転送クロックを受信する。
	RXD	受信データ入力端子。
	TXD	送信データ出力端子。
	SMR	SCI をクロック同期式モードに設定する。
	SCR	受信割り込みを許可し、SCI を送信モードに設定する。
	SSR	送信を開始する。
	RDR	H8/3003 から受信したデータを設定する。
	TDR	H8/3003 へ送信するデータを設定する。
	BRR	転送レートを設定する
PortA	PADDR	ポート A の入出力を設定する。
	PADR	RRQ の送信を行う。
DMAC0A	DTCR0A	DMAC の各チャンネルの動作を制御する。
	MAR0A	SCI 送信完了割り込みで起動するとき転送元アドレスを設定する。 SCI 受信完了割り込みで起動するとき転送先アドレスを設定する。
	IOAR0A	SCI 送信完了割り込みで起動するとき転送先アドレスを設定する。 SCI 受信完了割り込みで起動するとき転送元アドレスを設定する。
	ETCR0A	転送回数を設定する。

3.2.4 ソフトウェア説明

表 3.7 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	HSCIMN	I/O ポート、SCI および DMAC の初期設定を行う
データ送信	HSCITX	TDR に送信データを設定し、送信を開始する
データ受信	HSCIRX	IRQ0 割り込みで起動し、SCI を受信動作に設定。また、DMAC の再設定を行う
DMA 割り込み	DMAEND	WRIT_F または READ_F のクリアを行い、送信終了フラグまたは受信終了フラグをセットする。

表 3.8 使用内部レジスタ

内蔵機能	レジスタ名	機能
SCI	SMR	SCI を以下のように設定する <ul style="list-style-type: none"> SCI の動作モードをクロック同期式モードに設定 ポーレートジェネレータのクロックソースを に設定
	SCR	送信時と受信時に SCI を以下のように設定する 送信動作：送信データエンプティ割り込みを許可 送信動作を許可 SCK 端子を同期クロック出力 受信動作：受信データフル割り込みを許可 受信動作を許可 SCK 端子を同期クロック入力
	SSR	送信動作：TDRE をクリアして送信動作を開始する 受信動作：RDRF をクリアして受信動作を許可する
	RDR	H8/300H から受信したデータを格納する
	TDR	H8/300H に送信するデータを設定する
	BRR	転送レートを設定する
DMAC	DTCR0A	DMAC0A を以下のように設定する <ul style="list-style-type: none"> データサイズをバイトサイズに設定 MAR をインクリメントに設定 データ転送を I/O モードに設定 転送を許可する 起動要因を SCI の受信完了割り込みまたは送信完了割り込みに設定
	MAR0A	送信時：SCI データテーブルの転送元アドレスを設定 受信時：SCI データテーブルの転送先アドレスを設定
	IOAR0A	送信時：TDR（転送先）のアドレスを設定 受信時：RDR（転送元）のアドレスを設定
	ETCR0A	転送回数を設定
I/O	PADDR	ポート A の入出力を設定
	PADR	RRQ の送信を行う
割り込み コントローラ	IER	IRQ0 割り込みを許可
	ISR	IRQ0 入力ステータスを示す

表 3.9 引数の説明

ラベル名、レジスタ名	機能	データ長	使用モジュール名	入出力
ROL	送信するデータを格納する	1 バイト	メインルーチン データ送信	出力 入力
WRIT_F	ライト動作中であることを示すフラグ	1 ビット	データ送信 データ受信	出力 入力
READ_F	リード動作中であることを示すフラグ	1 ビット	データ送信 データ受信	入力 出力
WRIT_ERRF	送信が正常に終了したことを示すフラグ 0：送信正常終了 1：送信異常終了	1 ビット	データ送信 データ受信	出力 入力
READ_ERRF	受信が正常に終了したことを示すフラグ 0：受信正常終了 1：受信異常終了	1 ビット	データ送信 データ受信	入力 出力

3. 応用編

表 3.10 使用 RAM 説明

レジスタ名	機能	データ長	使用モジュール名
READ_END	リード動作が終了したことを示すフラグです。	1 ビット	DMA 割り込み
WRIT_END	ライト動作が終了したことを示すフラグです。	1 ビット	DMA 割り込み

表 3.11 データテーブルの説明

レジスタ名	機能	データ長	データ容量
TXSCI_DAT	送信データを設定します。	1 バイト	100 バイト
RXSCI_DAT	受信データを設定します。	1 バイト	100 バイト

- 使用汎用レジスタ

31	ERn	RnH	RnL	0
0	全ルーチンにてデータ設定時のワークとして使用			
1		RAMクリアの カウンタとして使用	RAMクリア データを設定	
2				
3				
4				
5				
6				
7				

図 3.10 使用汎用レジスタ

3.2.5 フローチャート

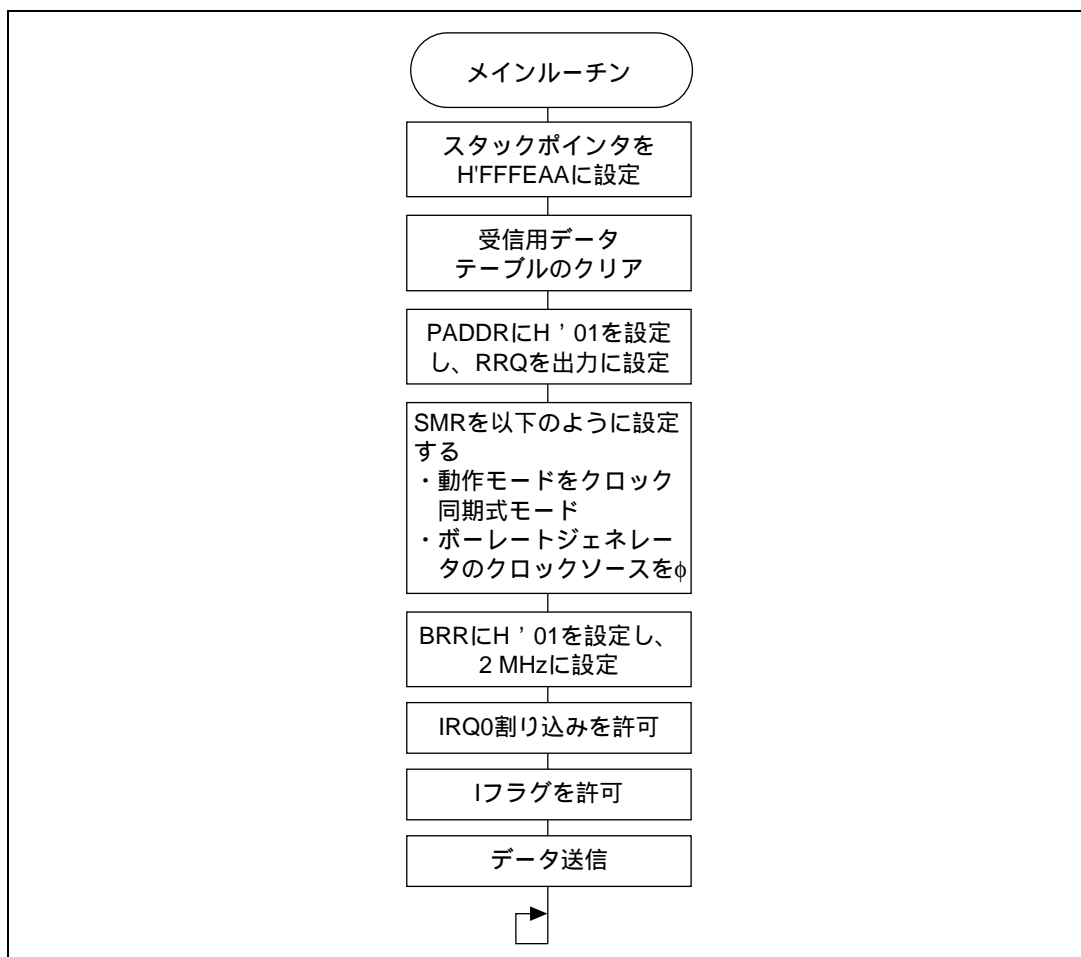


図 3.11 メインルーチン

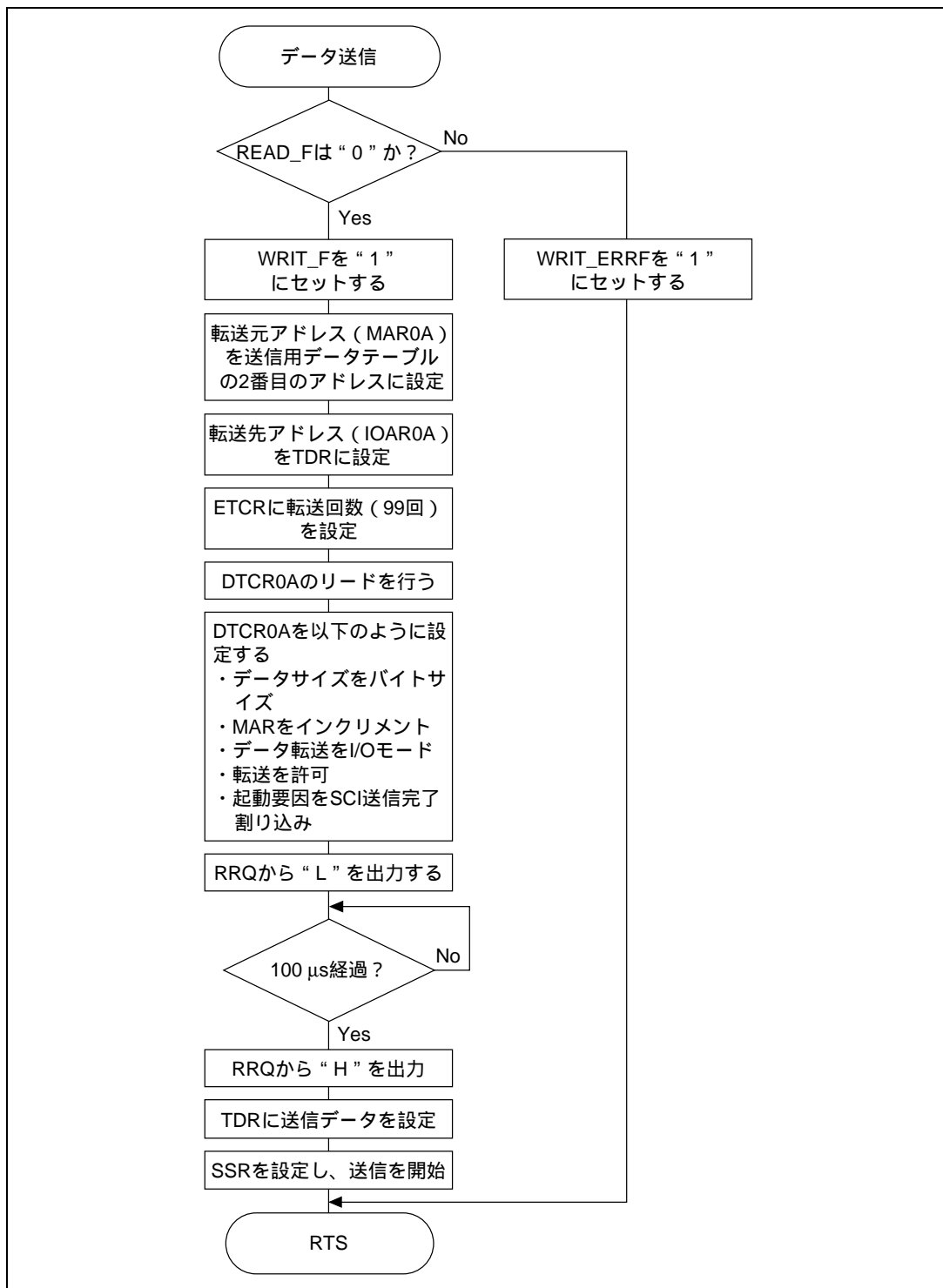


図 3.12 データ送信

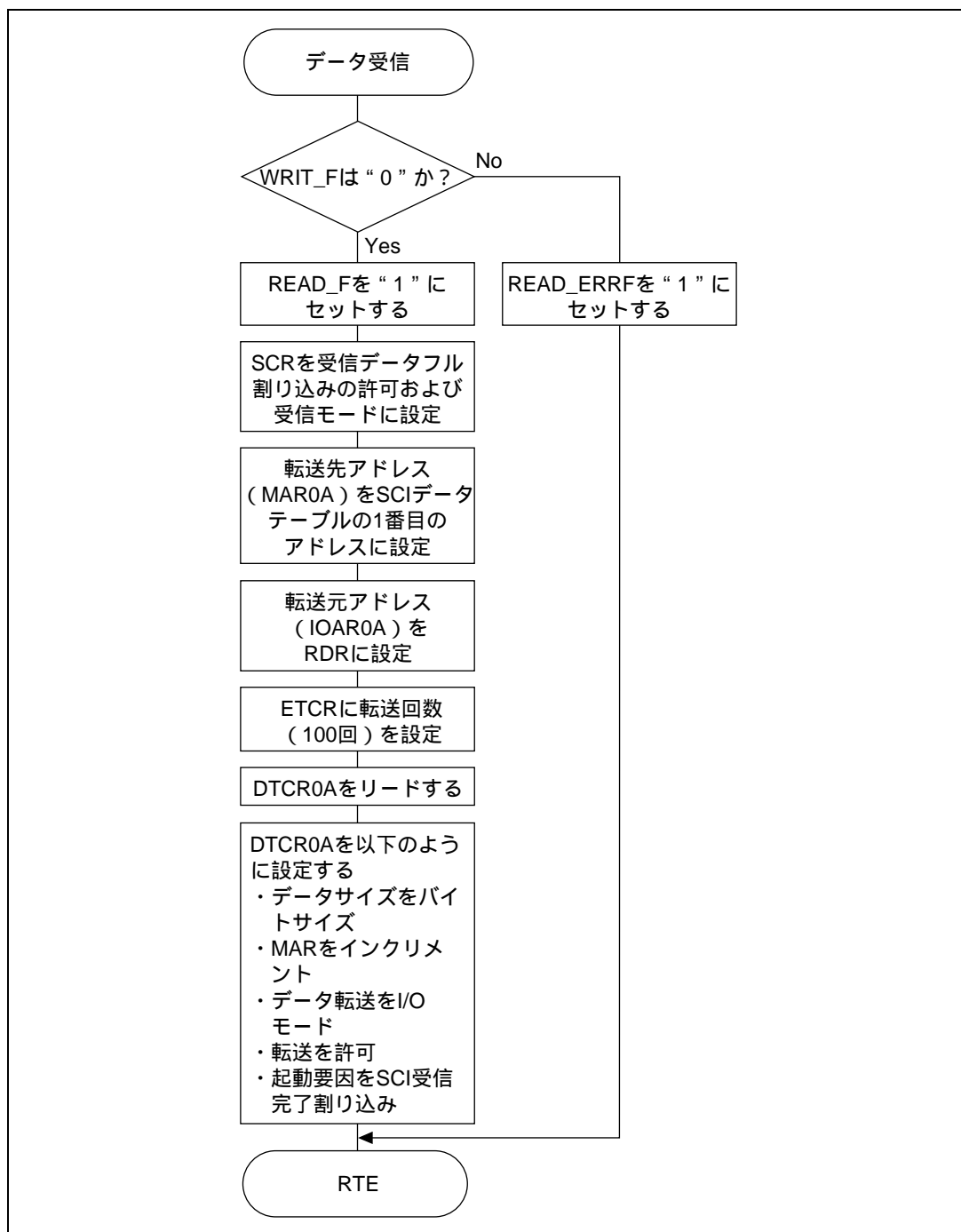


図 3.13 データ受信

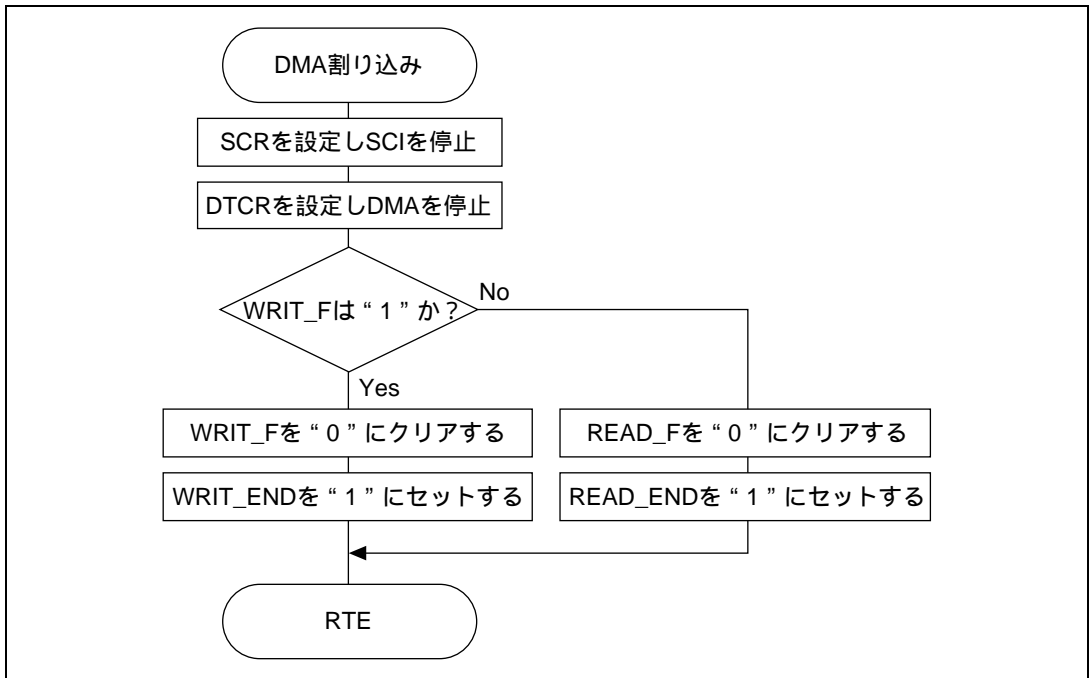


図 3.14 DMA 割り込み

3.2.6 プログラムリスト

```

1          1 ;*****
2          2 ;*
3          3 ;*          VECTOR ADDRESS
4          4 ;*
5          5 ;*****
6          6 ;
7          7          .CPU      300HA
8 000000   8          .SECTION  VECT, CODE ,LOCATE=H'000000
9          9 ;
10 000000 00010000 10 RES          .DATA.L   HSCIMN
11         11 ;
12 00001C   12          .ORG      H'00001C
13 00001C 00010000 13 NMI          .DATA.L   HSCIMN
14         14 ;
15 000030   15          .ORG      H'000030
16 000030 0001008E 16 IRQ0         .DATA.L   HSCIRX
17 000034 00010000 17 IRQ1         .DATA.L   HSCIMN
18 000038 00010000 18 IRQ2         .DATA.L   HSCIMN
19 00003C 00010000 19 IRQ3         .DATA.L   HSCIMN
20 000040 00010000 20 IRQ4         .DATA.L   HSCIMN
21 000044 00010000 21 IRQ5         .DATA.L   HSCIMN
22 000048 00010000 22 IRQ6         .DATA.L   HSCIMN
23 00004C 00010000 23 IRQ7         .DATA.L   HSCIMN
24 000050 00010000 24 WOVI0        .DATA.L   HSCIMN
25 000054 00010000 25 CMI          .DATA.L   HSCIMN
26         26 ;
27 000060   27          .ORG      H'000060
28 000060 00010000 28 IMIA0        .DATA.L   HSCIMN
29 000064 00010000 29 IMIB0        .DATA.L   HSCIMN
30 000068 00010000 30 OVI0         .DATA.L   HSCIMN
31         31 ;
32 000070   32          .ORG      H'000070
33 000070 00010000 33 IMIA1        .DATA.L   HSCIMN
34 000074 00010000 34 IMIB1        .DATA.L   HSCIMN
35 000078 00010000 35 OVI1         .DATA.L   HSCIMN
36         36 ;
37 000080   37          .ORG      H'000080
38 000080 00010000 38 IMIA2        .DATA.L   HSCIMN
39 000084 00010000 39 IMIB2        .DATA.L   HSCIMN
40 000088 00010000 40 OVI2         .DATA.L   HSCIMN
41         41 ;
42 000090   42          .ORG      H'000090
43 000090 00010000 43 IMIA3        .DATA.L   HSCIMN
44 000094 00010000 44 IMIB3        .DATA.L   HSCIMN
45 000098 00010000 45 OVI3         .DATA.L   HSCIMN
46         46 ;

```

3. 応用編

```

47 0000A0          47          .ORG      H'0000A0
48 0000A0 00010000 48 IMIA4     .DATA.L   HSCIMN
49 0000A4 00010000 49 IMIB4     .DATA.L   HSCIMN
50 0000A8 00010000 50 OVI4      .DATA.L   HSCIMN
51                51 ;
52 0000B0          52          .ORG      H'0000B0
53 0000B0 000100CC 53 DEND0A    .DATA.L   DMAEND
54 0000B4 00010000 54 DEND0B    .DATA.L   HSCIMN
55 0000B8 00010000 55 DEND1A    .DATA.L   HSCIMN
56 0000BC 00010000 56 DEND1B    .DATA.L   HSCIMN
57 0000C0 00010000 57 DEND2A    .DATA.L   HSCIMN
58 0000C4 00010000 58 DEND2B    .DATA.L   HSCIMN
59 0000C8 00010000 59 DEND3A    .DATA.L   HSCIMN
60 0000CC 00010000 60 DEND3B    .DATA.L   HSCIMN
61 0000D0 00010000 61 ERI0      .DATA.L   HSCIMN
62 0000D4 00010000 62 RXI0      .DATA.L   HSCIMN
63 0000D8 00010000 63 TXI0      .DATA.L   HSCIMN
64 0000DC 00010000 64 TEI0      .DATA.L   HSCIMN
65 0000E0 00010000 65 ERI1      .DATA.L   HSCIMN
66 0000E4 00010000 66 RXI1      .DATA.L   HSCIMN
67 0000E8 00010000 67 TXI1      .DATA.L   HSCIMN
68 0000EC 00010000 68 TEI1      .DATA.L   HSCIMN
69 0000F0 00010000 69 ADI       .DATA.L   HSCIMN
70                70 ;
71                71 ;*****
72                72 ;*
73                73 ;*          RAM ALLOCATION
74                74 ;*
75                75 ;*****
76                76 ;
77 FFFFEA          77          .SECTION  RAM,DATA,LOCATE=H'FFFEA
78 FFFFEA 00000064 78 RXSCI_DAT .RES.B   D'100
79                79 ;
80 FFFF0E 00000001 80 CSC_FL A  .RES.B   1          ;Flag area
81          00000000 81 WRIT_F   .EQU    0          ;Write action flag
82          00000001 82 READ_F   .EQU    1          ;Read action flag
83          00000002 83 WRIT_ERRF .EQU    2          ;Write error flag
84          00000003 84 READ_ERRF .EQU    3          ;Read error flag
85          00000004 85 WRIT_ENDF .EQU    4          ;Write end flag
86          00000005 86 READ_ENDF .EQU    5          ;Read end flag
87                87 ;
88                88 ;*****
89                89 ;*
90                90 ;*          SYMBOL DEFINITIONS
91                91 ;*
92                92 ;*****
93                93 ;

```



```

94          94 ;***** DMAC0A *****
95      00FFFF20 95 MAR0AR: .EQU H'FFFF20 ;Memory address register
96      00FFFF24 96 ETCR0AH: .EQU H'FFFF24 ;Transmit cunt register (H)
97      00FFFF25 97 ETCR0AL: .EQU H'FFFF25 ;Transmit cunt register (L)
98      00FFFF26 98 IOAR0A: .EQU H'FFFF26 ;I/O address register
99      00FFFF27 99 DTCR0A: .EQU H'FFFF27 ;Data transfer control register
100     00000007 100 DTE: .EQU 7 ;Data transfer enable bit
101          101 ;
102          102 ;
103          103 ;***** SCI *****
104     00FFFFB0 104 SMR0: .EQU H'FFFFB0 ;Serial mode register
105     00FFFFB1 105 BRR0: .EQU H'FFFFB1 ;bit rate register
106     00FFFFB2 106 SCR0: .EQU H'FFFFB2 ;Serial control register
107     00FFFFB3 107 TDR0: .EQU H'FFFFB3 ;Transmit data register
108     00FFFFB4 108 SSR0: .EQU H'FFFFB4 ;Serial status register
109     00000007 109 TDRE .EQU 7 ;Transmit data register enable
;bit
110          110 ;
111          111 ;***** I/O *****
112     00FFFFD1 112 PADDR: .EQU H'FFFFD1 ;Port A data direction register
113     00FFFFD3 113 PADR: .EQU H'FFFFD3 ;Port A data register
114     00000000 114 RRQ .EQU 0 ;RRQ signal bit
115          115 ;
116          116 ;***** INT CONTRAL *****
117     00FFFFF5 117 IER: .EQU H'FFFFF5 ;IRQ enable register
118     00000000 118 IRQOE .EQU 0 ;IRQ0 enable bit
119     00FFFFF6 119 ISR: .EQU H'FFFFF6 ;IRQ status register
120     00000000 120 IRQOF .EQU 0 ;IRQ0 status bit
121          121 ;
122          122 ;*****
123          123 ;* *
124          124 ;* MAIN PROGRAM : HSCIMN *
125          125 ;* *
126          126 ;*****
127          127 ;
128     010000 128 .SECTION PROG, CODE, LOCATE=H'010000
129          129 ;
130     00010000 130 HSCIMN: .EQU $
131     010000 7A0700FFFEAA 131 MOV.L #H'FFFEAA, SP ;Initialize stack pointer
132          132 ;
133     010006 F100 133 MOV.B #0, R1H ;Counter clr
134     010008 F900 134 MOV.B #0, R1L
135     01000A 7A0000FFFEAA 135 MOV.L #RXSCI_DAT, ER0
136     010010 6889 136 HSCIMN01: MOV.B R1L, @ER0
137     010012 0B70 137 INC.L #1, ER0
138     010014 8101 138 ADD.B #1, R1H
139     010016 A164 139 CMP.B #D'100, R1H ;Counter = 100 ?

```

3. 応用編

```

140 010018 46F6          140          BNE      HSCIMN01      ;No
141                      141      ;
142                      142      ;***** Initialize I/O *****
143 01001A F801          143          MOV.B   #H'01,R0L
144 01001C 38D3          144          MOV.B   R0L,@PADR      ;Initialize PADR
145 01001E 38D1          145          MOV.B   R0L,@PADDR     ;Initialize PADDR(RRQ)
146                      146      ;
147                      147      ;***** Initialize SCIO *****
148 010020 F880          148          MOV.B   #H'80,R0L
149 010022 38B0          149          MOV.B   R0L,@SMR0     ;Initialize SMAR0(1 -> C/A)
150 010024 F801          150          MOV.B   #H'01,R0L
151 010026 38B1          151          MOV.B   R0L,@BRR0     ;SET BRR0(2M bit/s)
152                      152      ;
153                      153      ;***** Initialize INT CONTROLER *****
154 010028 7FF57000      154          BSET    #IRQ0E,@IER
155                      155      ;
156 01002C 0700          156          LDC.B   #0.CCR      ;Enable interrupt
157                      157      ;
158 01002E 5E010034      158          JSR     @HSCITX
159 010032 40FE          159 HSCIMN99:  BRA     HSCIMN99
160                      160      ;
161                      161      ;*****
162                      162      ;*
163                      163      ;*      NAME : HSCITX(Output data)
164                      164      ;*
165                      165      ;*****
166                      166      ;*
167                      167      ;*      ENTRY : WRIT_F(Write action flag)
168                      168      ;*      READ_ERRF(Read error flag)
169                      169      ;*      RETURNS : READ_F(Read action flag)
170                      170      ;*      WRIT_ERRF(Write error flag)
171                      171      ;*
172                      172      ;*****
173                      173      ;
174 010034 7E0E7310      174 HSCITX:   BTST   #READ_F,@CSC_FL A ;READ_F = 0 ?
175 010038 5860004C      175          BNE     HSCITX01      ;No
176 01003C 7F0E7000      176          BSET   #WRIT_F,@CSC_FL A ;1 -> WRIT_F
177                      177      ;
178 010040 7A00000100FA  178          MOV.L   #TXSCI_DAT,ER0
179 010046 0B70          179          INC.L   #1,ER0
180 010048 01006B80FF20  180          MOV.L   ER0,@MAR0AR     ;Set MAR0A
181 01004E F8B3          181          MOV.B   #H'B3,R0L
182 010050 3826          182          MOV.B   R0L,@IOAR0A    ;Set IOAR0A(TDR0)
183 010052 79000063      183          MOV.W   #D'99,R0
184 010056 6B80FF24      184          MOV.W   R0,@ETCR0AH    ;Store tranfer count
185 01005A 2827          185          MOV.B   @DTCR0A,R0L
186 01005C F88C          186          MOV.B   #H'8C,R0L

```

```

187 01005E 3827          187          MOV.B    R0L,@DTCR0A ;Initialize DTCR0A
                                   (1 -> DTIE, DTS2A)
188                      188 ;
189 010060 7FD37200     189          BCLR    #RRQ,@PADR ;RRQ -> "L" output
190 010064 F100         190          MOV.B    #0,R1H ;Counter clr
191 010066 8101         191 HSCITX02: ADD.B    #H'01,R1H
192 010068 A1C8         192          CMP.B    #H'C8,R1H ;100us pass?
193 01006A 46FA         193          BNE     HSCITX02
194                      194 ;
195 01006C 7FD37000     195          BSET    #RRQ,@PADR ;RRQ -> "H" output
196                      196 ;
197 010070 7EB47370     197 HSCITX03: BTST    #TDRE,@SSR0 ;TDRE = 1?
198 010074 47FA         198          BEQ     HSCITX03 ;No
199 010076 6A28000100F4 199          MOV.B    @TXSCI_DAT,R0L
200 01007C 38B3         200          MOV.B    R0L,@TDR0 ;Set TDR
201 01007E 7FB47270     201          BCLR.B  #7,@SSR0
202 010082 F8A0         202          MOV.B    #H'A0,R0L
203 010084 38B2         203          MOV.B    R0L,@SCR0 ;Initialize SCR0(1 -> TIE,TE)
204 010086 5470         204 HSCITX99  RTS
205                      205 ;
206 010088 7F0E7020     206 HSCITX01: BSET    #WRIT_ERRF,@CSC_FL A ;1 -> WRIT_ERRF
207 01008C 40F8         207          BRA     HSCITX99
208                      208 ;
209                      209 ;*****
210                      210 ;*
211                      211 ;*      NAME : HSCIRX(Input data)
212                      212 ;*
213                      213 ;*****
214                      214 ;*
215                      215 ;*      ENTRY : READ_F(Read action flag)
216                      216 ;*      WRIT_ERRF(Write error flag)
217                      217 ;*      RETURNS : WRIT_F(Write action flag)
218                      218 ;*      READ_ERRF(Read error flag)
219                      219 ;*
220                      220 ;*****
221                      221 ;
222 01008E 01006DF0     222 HSCIRX:  PUSH.L  ERO ;Escape register
223 010092 7E0E7300     223          BTST    #WRIT_F,@CSC_FL A;WRIT_F = 0?
224 010096 5860002C     224          BNE     HSCIRX01 ;No
225 01009A 7F0E7010     225          BSET    #READ_F,@CSC_FL A;1 -> READ_F
226 01009E F852         226          MOV.B    #H'52,R0L
227 0100A0 38B2         227          MOV.B    R0L,@SCR0 ;Initialize SCR0
                                   ;(1 -> RIE,RE,CKE1)
228                      228 ;
229 0100A2 7A0000FFFEAA 229          MOV.L    #RXSCI_DAT,ERO
230 0100A8 01006B80FF20 230          MOV.L    ERO,@MAR0AR ;Set MAR0A
231 0100AE F8B5         231          MOV.B    #H'B5,R0L

```

3. 応用編

```

232 0100B0 3826          232          MOV.B    R0L,@IOAR0A      ;Set IOAR0A(RDR0)
233 0100B2 F864          233          MOV.B    #D'100,R0L
234 0100B4 3825          234          MOV.B    R0L,@ETCR0AL    ;Store tranfer count
235 0100B6 2827          235          MOV.B    @DTCR0A,R0L     ;Read DTCR0A
236 0100B8 F80D          236          MOV.B    #H'0D,R0L
237 0100BA 3827          237          MOV.B    R0L,@DTCR0A     ;Initialize DTCR0A
                                   ;(1->DTIE,DTS2A,
                                   ; DTS0A)
238 0100BC 7F277070      238          BSET    #DTE,@DTCR0A    ;Start DMAC0A
239                                239          ;
240 0100C0 01006D70      240 HSCIRX99:  POP.L    ERO
241 0100C4 5670          241          RTE
242                                242          ;
243 0100C6 7F0E7030      243 HSCIRX01:  BSET    #READ_ERRF,@CSC_FLA ;1 -> READ_ERRF
244 0100CA 40F4          244          BRA     HSCIRX99
245                                245          ;
246                                246          ;*****
247                                247          ;*
248                                248          ;*      NAME : DMAEND(Set next data)
249                                249          ;*
250                                250          ;*****
251                                251          ;*
252                                252          ;*      ENTRY : WRIT_F(Write action flag)
253                                253          ;*      RETURNS : WRIT_F(Write action flag)
254                                254          ;*      READ_F(Read action flag)
255                                255          ;*      WRIT_ENDF(Write end flag)
256                                256          ;*      READ_ENDF(Read end flag)
257                                257          ;*
258                                258          ;*****
259                                259          ;
260 0100CC F800          260 DMAEND:   MOV.B    #'00,R0L
261 0100CE 38B2          261          MOV.B    R0L,@SCR0
262 0100D0 3827          262          MOV.B    R0L,@DTCR0A     ;Initialize DTCR0A
                                   ;(0 -> DTIE,DTS2A)
263 0100D2 7E0E7300      263          BTST    #WRIT_F,@CSC_FLA ;WRIT_F = 1?
264 0100D6 5870000C      264          BEQ    DMAEND01         ;No
265                                265          ;
266 0100DA 7F0E7200      266          BCLR    #WRIT_F,@CSC_FLA ;WRIT_F Clear
267 0100DE 7F0E7040      267          BSET    #WRIT_ENDF,@CSC_FLA ;1 -> WRIT_ENDF
268 0100E2 5800000C      268          BRA     DMAEND99
269                                269          ;
270 0100E6 7F0E7210      270 DMAEND01:  BCLR    #READ_F,@CSC_FLA  ;READ_F Clear
271 0100EA 7F0E7050      271          BSET    #READ_ENDF,@CSC_FLA ;1 -> READ_ENDF
272 0100EE 7FF67200      272          BCLR    #IRQ0F,@ISR      ;IRQ0F Clr
273                                273          ;
274 0100F2 5670          274 DMAEND99  RTE
275                                275          ;

```

```

276                                276 ;
277                                277 ;*****
278                                278 ;*                               *
279                                279 ;*           DATA TABLE (WRITE DATA TABLE)   *
280                                280 ;*                               *
281                                281 ;*****
282 0100F4 010203040506070B 282 TXSCI_DAT  .DATA.B   D'01,D'02,D'03,D'04,D'05,D'06,
                                                                D'07,D'08,D'09,D'10,
283 0100FC 090A0B0C0D0E0F10 283 *           D'11,D'12,D'13,D'14,D'15,D'16,
                                                                D'17,D'18,D'19,D'20,
284 010104 1112131415161718 284 *           D'21,D'22,D'23,D'24,D'25,D'26,
                                                                D'27,D'28,D'29,D'30,
285 01010C 191A1B1C1D1E1F20 285 *           D'31,D'32,D'33,D'34,D'35,D'36,
                                                                D'37,D'38,D'39,D'40
286 010114 2122232425262728 286 *           D'41,D'42,D'43,D'44,D'45,D'46,
                                                                D'47,D'48,D'49,D'50,
287 01011C 292A2B2C2D2E2F30 287 *           D'51,D'52,D'53,D'54,D'55,D'56,
                                                                D'57,D'58,D'59,D'60,
288 010124 3132333435363738 288 *           D'61,D'62,D'63,D'64,D'65,D'66,
                                                                D'67,D'68,D'69,D'70,
289 01012C 393A3B3C3D3E3F40 289 *           D'71,D'72,D'73,D'74,D'75,D'76,
                                                                D'77,D'78,D'79,D'80,
290 010134 4142434445464748 290 *           D'81,D'82,D'83,D'84,D'85,D'86,
                                                                D'87,D'88,D'89,D'90
291 01013C 494A4B4C4D4E4F50 291 *           D'91,D'92,D'93,D'94,D'95,D'96,
                                                                D'97,D'98,D'99,D'100

010144 5152535455565758
01014C 595A5B5C5D5E5F60
010154 61626364

292                                292
293                                293                                .END

*****TOTAL ERRORS          0
*****TOTAL WARNINGS        0

```

3.3 サイン波出力

MCU : H8/3042

使用機器 : ITU、DMAC、D/A

3.3.1 仕様

- 図 3.15 に示すように RAM に設定された 100 ビットのビット列 (1010111.....) に対応した波形を出力します。
- RAM に設定された 1 ビットのデータを以下に示す周波数のサイン波形で出力します。
0 : 2400Hz のサイン波を全波出力する。
1 : 1200Hz のサイン波形を半波出力する。
- RAM に設定されたデータは MSB ファーストで出力します。
- 本タスク例を応用することにより MSK モデムの機能が実現できます。

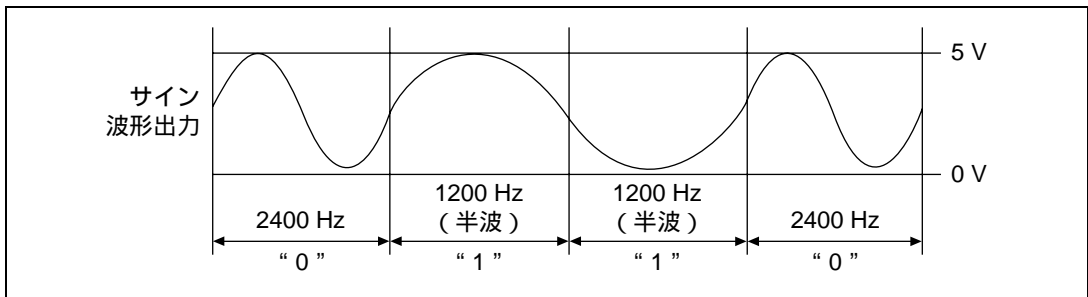


図 3.15 サイン波形出力例

3.3.2 考え方

- サイン波出力動作
図 3.16 に D/A 出力を使用してサイン波形を出力する例を示します。動作概要は以下のとおりです。
 - 図 3.16 に示すように D/A 変換器を使用し、アナログ電圧を出力します。
 - 出力するアナログ電圧に対応するサイン波 1 周期分の D/A データを ROM のデータテーブル上に確保しておきます。
 - データテーブル上の D/A データを出力周期 (T) ごとに D/A へ転送し、電圧を階段状に変化させます。
 - 階段状の波形を各種のフィルター回路を通すことにより、図 3.16 の点線のようなサイン波形を出力します。

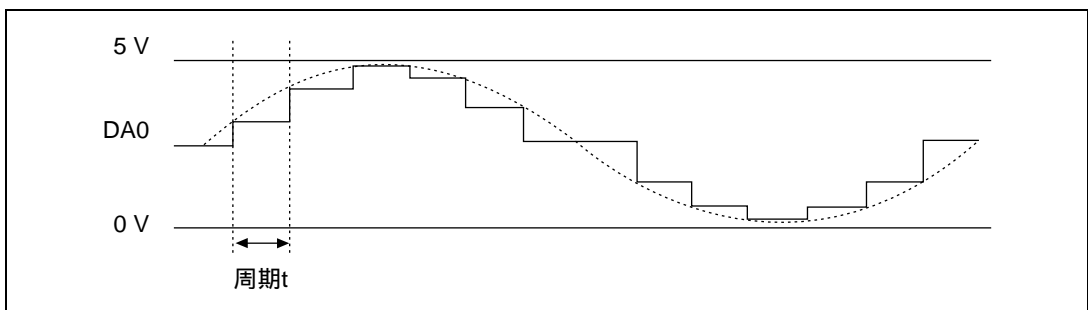


図 3.16 サイン波形出力例

- データフォーマット

図3.17に“0101”の4ビットのビットパターンに合わせて波形を出力した例を示します。

図3.17に示すように、“1”の出力は1200Hz半波出力となります。このため、次に出力する波形は、2400Hzまたは、1200Hzの逆相の波形を出力しなければなりません。

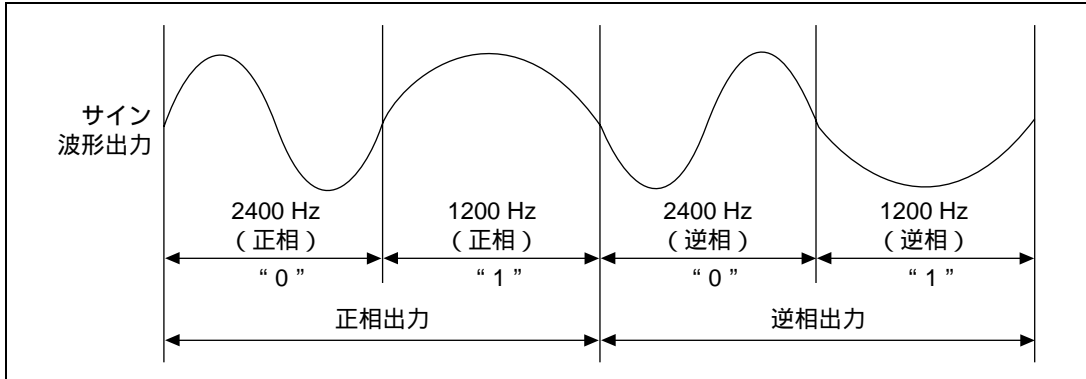


図 3.17 “0101” 4 ビットデータ出力例

- 1200Hz、2400Hz 正相、逆相波形出力方法

図3.18に各波形出力のために使用する波形データテーブルを示します。

図3.18に示すように、1200Hz正相半波、1200Hz逆相半波および2400Hz全波のデータを波形データテーブルとして確保しておきます。

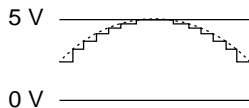
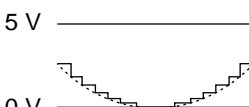
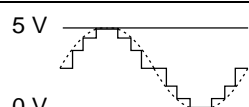
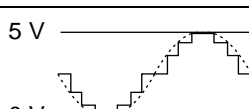
表3.12に出力波形と、波形データテーブルの使用方を示します。表3.12に示すように、波形データテーブルの先頭番地とアドレスのインクリメントおよびデクリメントを判定し、波形を出力します。

サイン波形 データテーブル ROM	アドレス
1200Hz 半波データ (正相)	S12USTADR (1200Hz正相半波データテーブル先頭アドレス)
1200Hz 半波データ (逆相)	S12LSTADR (1200Hz逆相半波データテーブル先頭アドレス)
2400Hz 全波データ	S24STADR (2400Hzデータテーブル先頭アドレス)
	S24EDADR (2400Hzデータテーブル最終アドレス)

図 3.18 サイン波形データテーブルと出力波形の関係

3. 応用編

表 3.12 出力波形および波形データテーブルの使用方法

波形名称	出力波形	先頭アドレス	インクリメント/ デクリメント	内容
“1”出力		S12USTADR	インクリメント	データテーブルの S12USTADR から S12LSTADR-1 番地までをインクリメントしてデータを出力します。
“1”出力		S12LSTADR	インクリメント	データテーブルの S12LSTADR から S24STADR-1 番地までをインクリメントしてデータを出力します。
“0”出力		S24STADR	インクリメント	データテーブルの S24STADR から S24EDADR までをインクリメントしてデータを出力します。
“0”出力		S24EDADR	デクリメント	データテーブルの S24EDADR から S24STADR までをデクリメントしてデータを出力します。

3.3.3 使用機能説明

図 3.19 に本タスクで使用する H8/3042 の内蔵機能を示します。本タスク例では、H8/3042 の機能を以下のように使用しサイン波を出力します。

- サイン波形データテーブル：D/A 変換を行うためのサイン波形データテーブルを ROM 上に設定します。
- ITU：コンペアマッチ A が起きるごとにタイマカウンタのクリアを行います。また、DMAC0A を起動します。
- DMAC0A：ITU のコンペアマッチ A で起動し、サイン波形データテーブルから出力データを D/A 変換器の DADR に転送します。
- D/A：サイン波形データを D/A 変換し、出力を行います。また、 AV_{CC} を基準電圧としてアナログ変換電圧範囲を設定することができます。

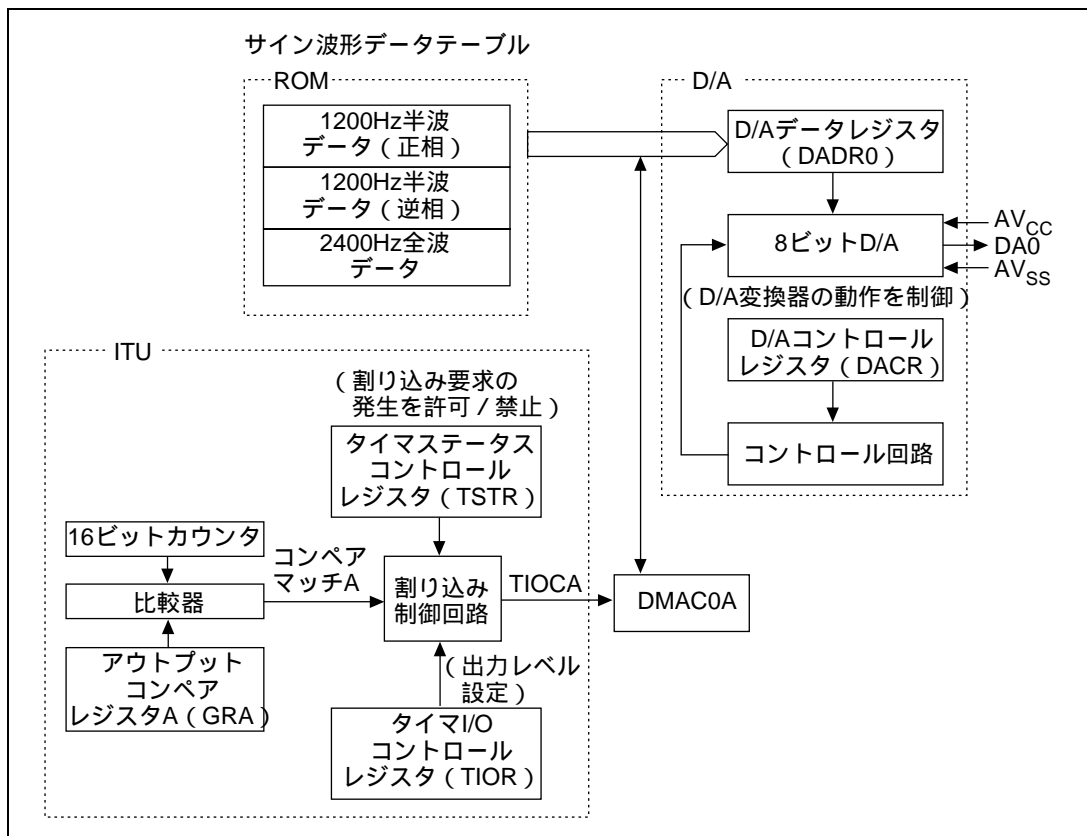


図 3.19 サイン波出力回路ブロック図

3. 応用編

表 3.13 に本タスク例の機能割り付けを示します。表 3.13 に示すように H8/3042 の機能を割り付け、サイン波形を出力します。

表 3.13 H8/3042 機能割り付け

H8/3042 の機能		機能
ITU (ch0)	TCNT0	一定周期でコンペアマッチを発生させる 16 ビットカウンタ
	GRA0	アウトプットコンペアレジスタ
	TCR0	カウンタクロックの選択およびカウンタクリア要因を選択
	TIOR0	GRA をアウトプットコンペアレジスタに設定
	TSR0	コンペアマッチやオーパフローのステータスを示す
	TIER0	割り込みの許可 / 禁止を選択
	TSTR	タイマカウンタのカウンタ動作の許可 / 禁止を設定
DMAC0A	DTCR0A	DMAC0A の動作を制御する
	MAR0A	サイン波形に応じたデータテーブルの先頭アドレスを設定
	IORA0A	DADR のアドレスを設定
	ETCR0A	転送回数を設定
D/A	DADR	変換を行うデータを格納する
	DACR	D/A 変換器の動作を制御する
	AVcc	アナログ部の電源および基準電圧
	AVss	アナログ部のグランドおよび基準電圧
	DA0	アナログ出力

3.3.4 動作説明

図 3.20 に動作原理を示します。図 3.20 に示すように H8/3042 のハードウェア処理およびソフトウェアの処理によりサイン波形を出力します。

(1)基本サイン波出力

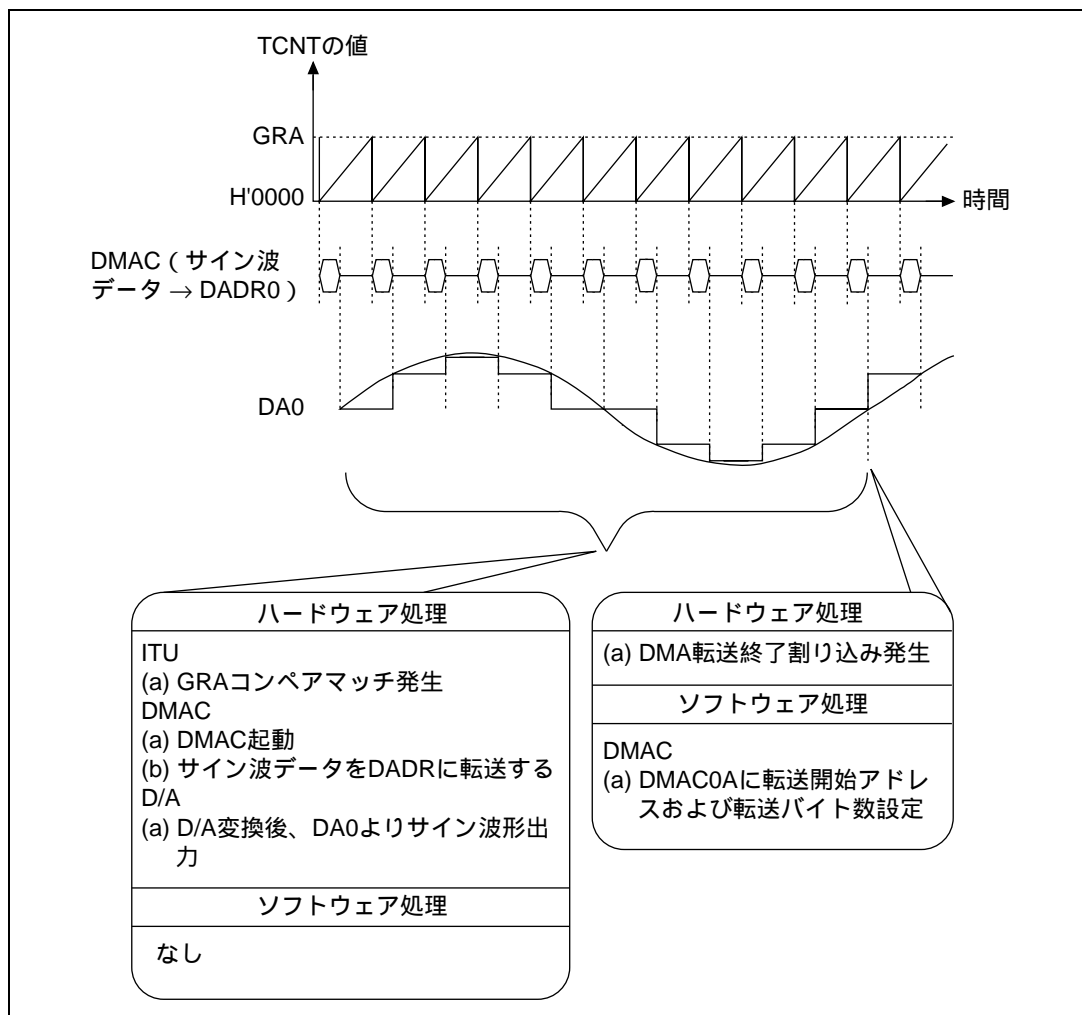


図 3.20 サイン波出力動作原理 (1)

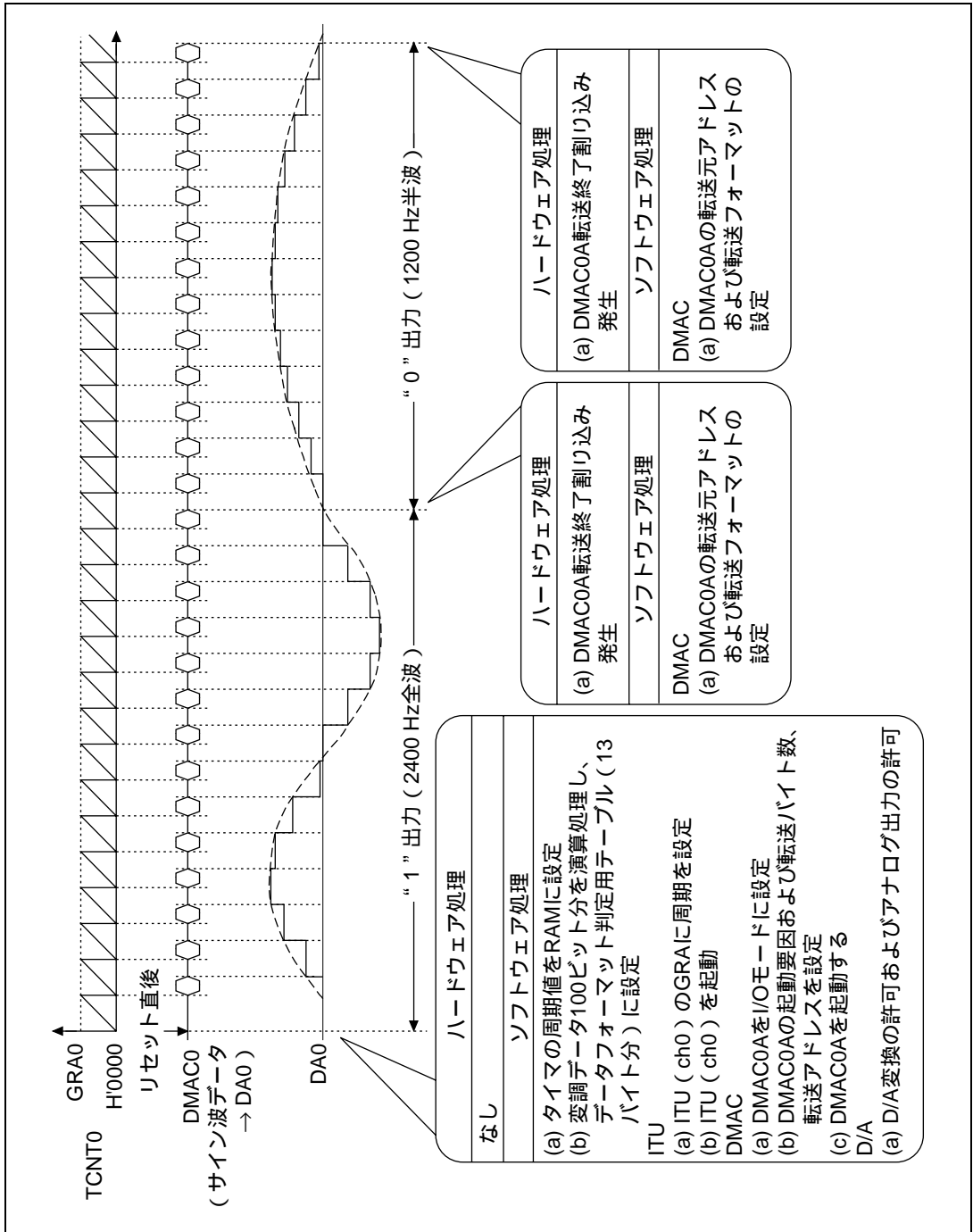


図 3.20 サイン波出力動作原理 (2)

3.3.5 ソフトウェア説明

表 3.14 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	SINMN	スタックポインタ、16ビットタイマ (ITU)、DMAC および D/A の初期設定を行う
DMAC スタート	SINWAVEA	DMAC0A 転送終了割り込みで起動し、DMAC0A の再設定および、再起動を行う

表 3.15 使用内部レジスタ

内蔵機能	レジスタ名	機能
ITU	GRA0	データ出力周期を設定
	TIER0	IMFA 割り込みを許可
	TCR0	ITU を以下のように設定する <ul style="list-style-type: none"> GRA のコンペアマッチでカウンタクリア 内部クロック でカウントアップ
	TIOR0	GRA をアウトプットコンペアレジスタに設定し、端子出力を禁止
	TSTR0	TCNT のカウント動作を許可する
DMAC0A	DTCR0A	DMAC0A を以下のように設定する <ul style="list-style-type: none"> データサイズをバイトサイズに設定 MAR をインクリメントまたはデクリメントに設定 (モジュールによって異なる) データ転送を I/O モードに設定 転送を許可または禁止を設定 (モジュールによって異なる) 起動要因を ITUch0 のコンペアマッチ A に設定
	MAR0A	サイン波形データテーブルの転送元アドレスを設定
	IOAR0A	DADR のアドレス (転送先) を設定
	ETCR0A	転送回数を設定
D/A	DADR	D/A 変換を行うデータ (サイン波形データテーブル) を格納
	DACR	D/A 変換器を以下のように設定する <ul style="list-style-type: none"> チャンネル 0 の D/A 変換を許可 アナログ出力 DA0 を許可

表 3.16 引数の説明

ラベル名、レジスタ名	機能	データ長	使用モジュール名	入出力
SIN_MOD	現在出力している波形のタイプを示すフラグ。 H'01 : "1" 出力 (正相) のとき H'02 : "1" 出力 (逆相) のとき H'04 : "0" 出力 (正相) のとき H'08 : "0" 出力 (逆相) のとき	1 バイト	メインルーチン	出力
			SINWAVE	入出力
SIN_LWCOUN	DMA 割り込みが実行されるごとに加算していくアップカウンタ。32 回カウント後、サイン波形データの処理を行う。	1 バイト	メインルーチン	入出力
			SINWAVE	入出力

3. 応用編

ラベル名、レジスタ名	機能	データ長	使用モジュール名	入出力
SIN_FCOUNT	DMA 割り込みが実行されるごとに加算していくアップカウンタ。100 回カウント後、周辺機能を停止する。	1 バイト	メインルーチン	入力
			SINWAVE	入出力
SIN_ADR	次に使用する変調データの番地を示す。SINLWDAT が変調データを読み込むと SINADR はインクリメントし次に使用する変調データの番地を示す。	1 ロングワード	メインルーチン	入力
			SINWAVE	入出力
SIN_DAT	現在使用している変調データを格納する。 1 ビットシフトして変調データを取り出す。32 ビットすべての変調データを取り出し後、SINADR の示すアドレスから変調データを読み込む。	1 ロングワード	メインルーチン	入出力
			SINWAVE	入出力

表 3.17 使用 RAM の説明

ラベル名	機能	データ長	使用モジュール名
OUT_DAT	変調データ (1010111...) を RAM に設定します。	100 ビット	SINWAVEA

表 3.18 データテーブルの説明

テーブル名	機能	データ長	データ容量
S12USTADR S12LSTADR S24STADR S24EDADR	D/A 変換を行うサイン波形データを設定する	バイト	60 バイト

• 使用汎用レジスタ

31	ERn	RnH	RnL	0
0	全ルーチンにてデータ設定時のワークとして使用			
1	割り込み処理内でワークとして使用			
2				
3				
4				
5				
6				
7				

図 3.21 使用汎用レジスタ

3.3.6 フローチャート

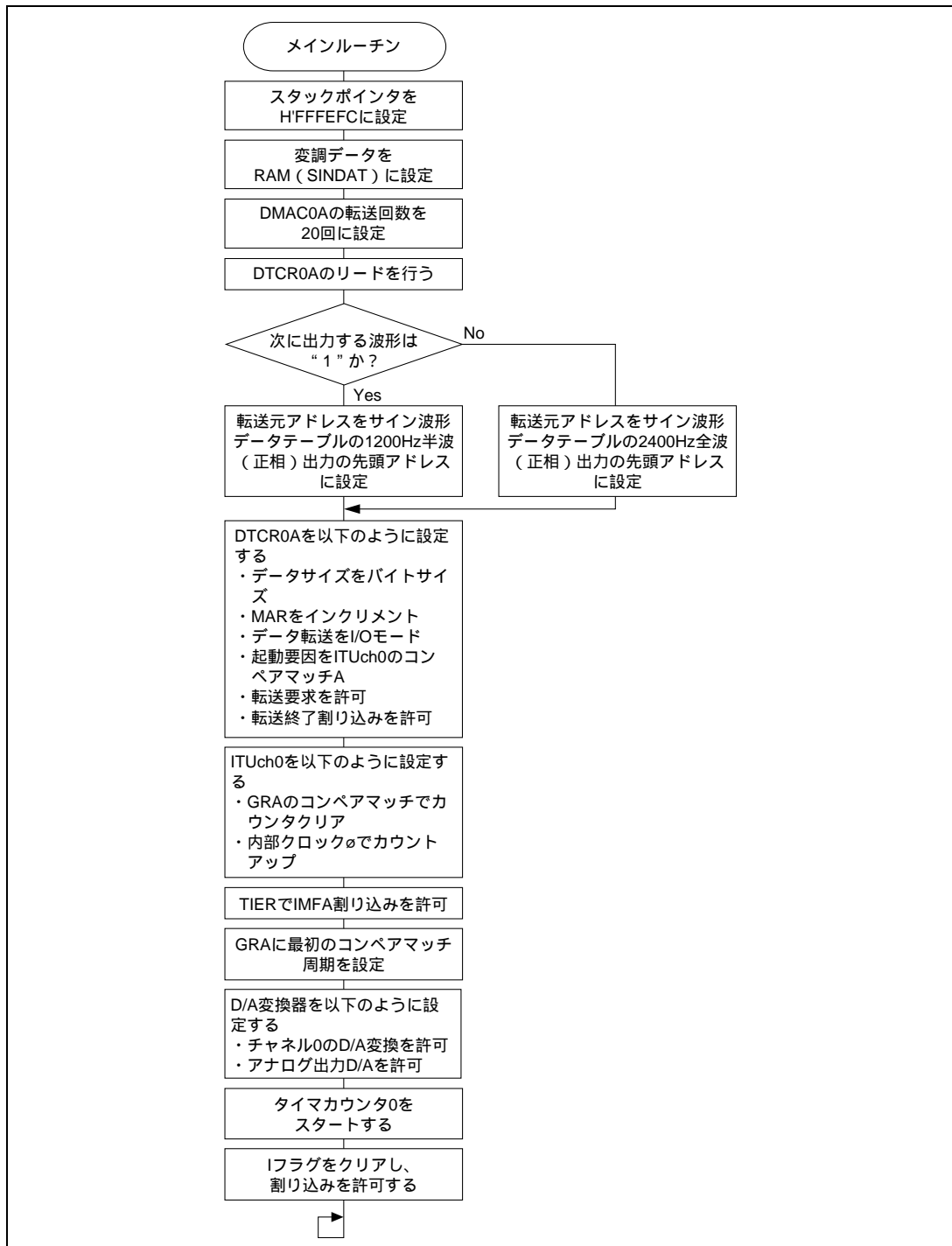


図 3.22 メインルーチン

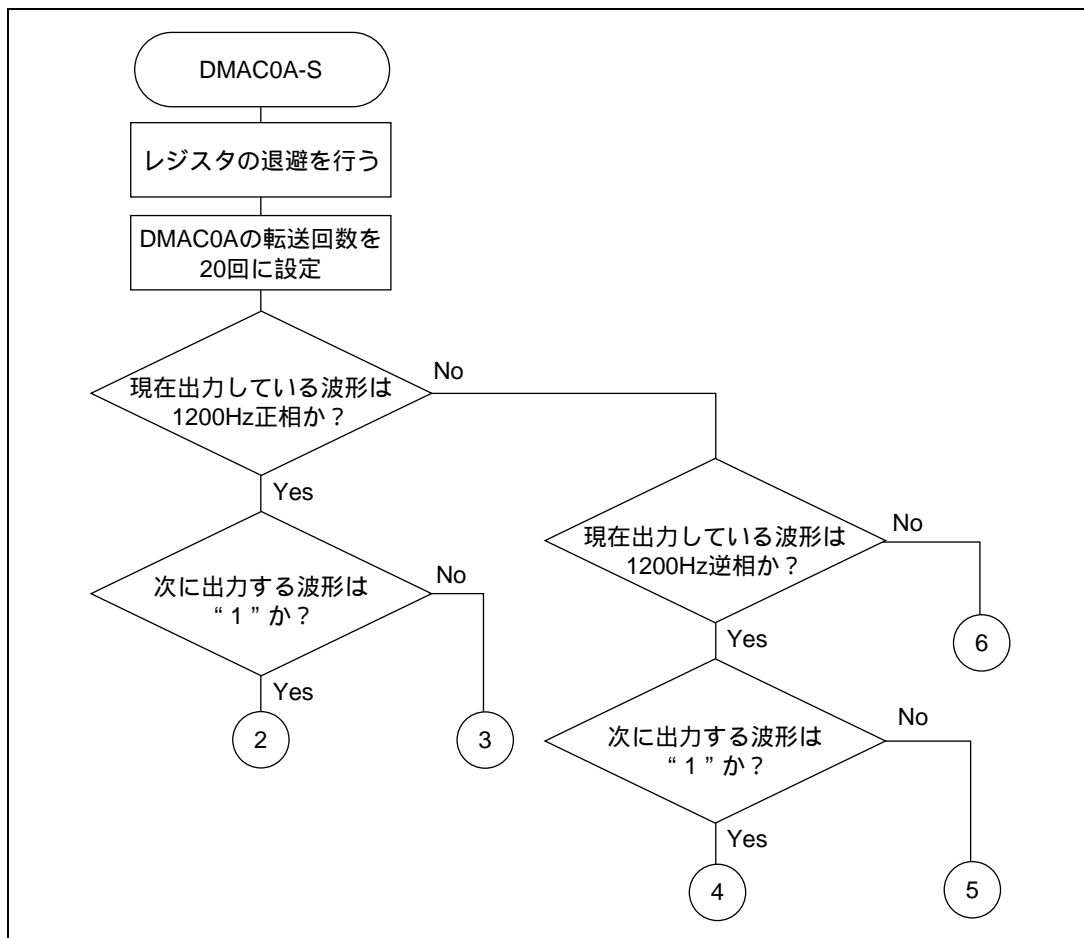


図 3.23 DMAC スタート (1)

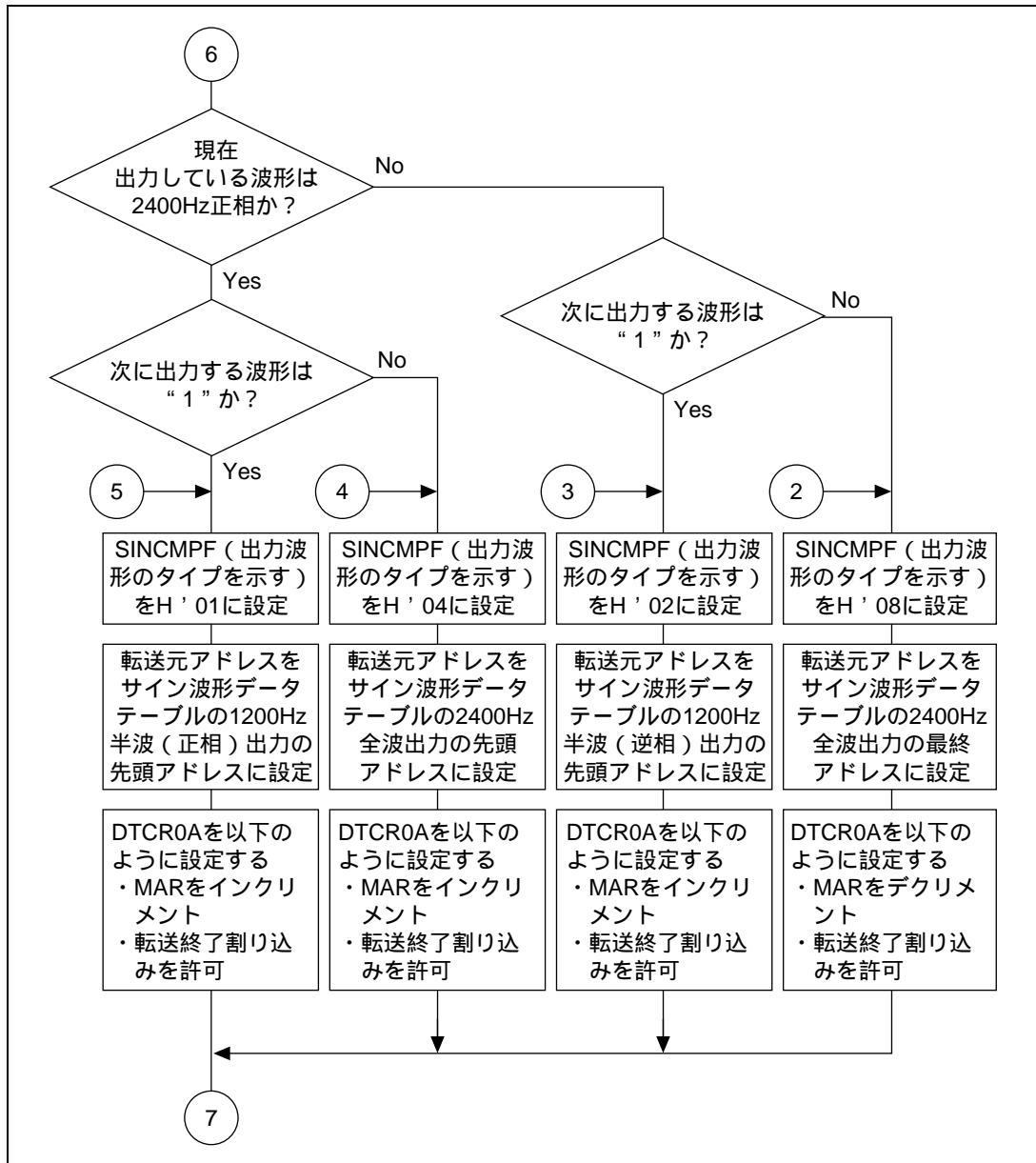


図 3.23 DMAC スタート (2)

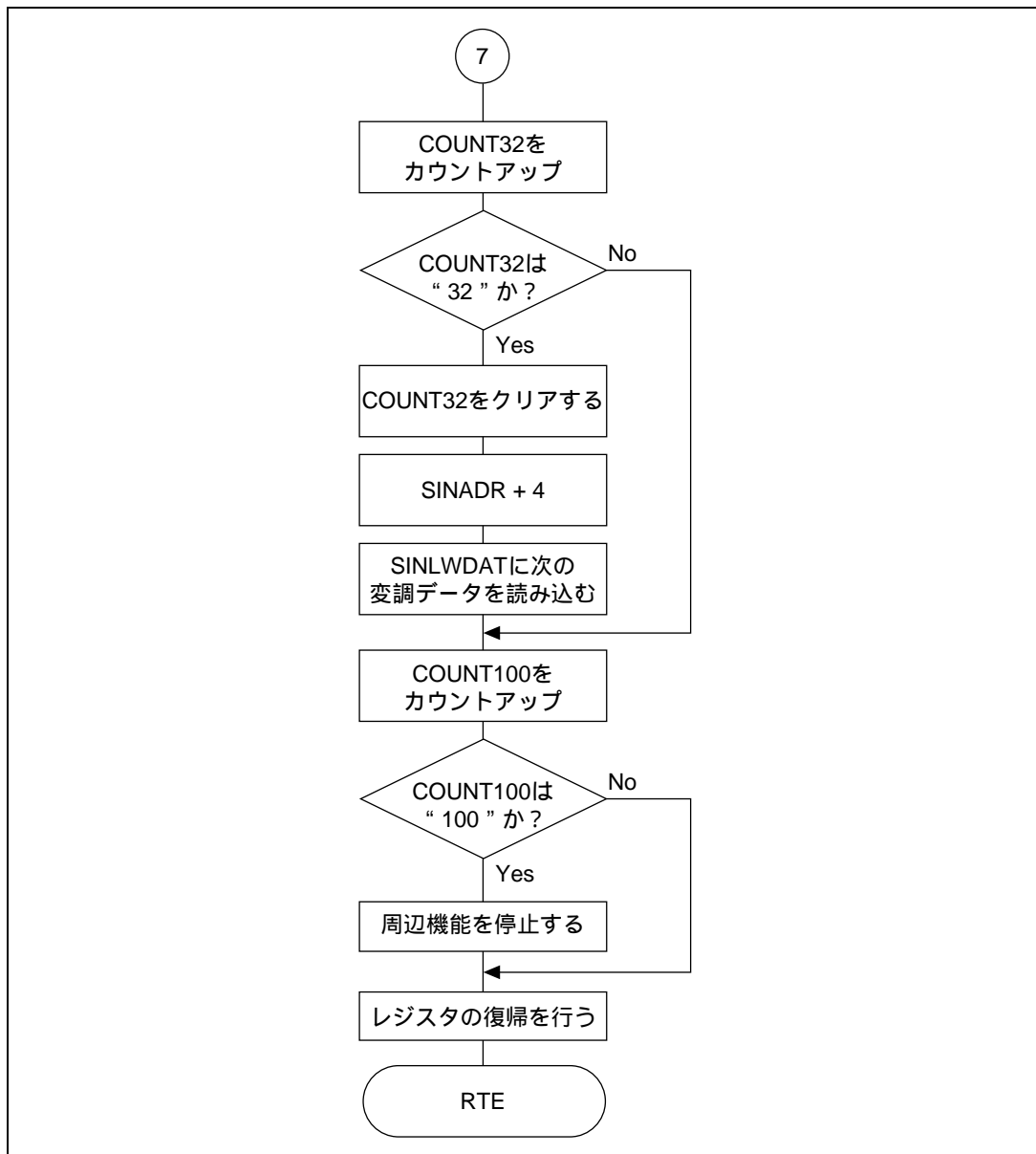


図 3.23 DMAC スタート (3)

3.3.7 プログラムリスト

```

1          1  ;*****
2          2  ;*
3          3  ;*          VECTOR ADDRESS
4          4  ;*
5          5  ;*****
6          6  ;
7          7          .CPU          300HA
8 000000   8          .SECTION  VECT, CODE ,LOCATE=H'000000
9          9  ;
10 000000 00010000 10 RES          .DATA.L  SINMN
11          11 ;
12 00001C   12          .ORG          H'00001C
13 00001C 00010000 13 NMI          .DATA.L  SINMN
14          14 ;
15 000030   15          .ORG          H'000030
16 000030 00010000 16 IRQ0         .DATA.L  SINMN
17 000034 00010000 17 IRQ1         .DATA.L  SINMN
18 000038 00010000 18 IRQ2         .DATA.L  SINMN
19 00003C 00010000 19 IRQ3         .DATA.L  SINMN
20 000040 00010000 20 IRQ4         .DATA.L  SINMN
21 000044 00010000 21 IRQ5         .DATA.L  SINMN
22          22 ;
23 000050   23          .ORG          H'000050
24 000050 00010000 24 WOVI        .DATA.L  SINMN
25 000054 00010000 25 CMI          .DATA.L  SINMN
26          26 ;
27 000060   27          .ORG          H'000060
28 000060 00010000 28 TGAI0        .DATA.L  SINMN
29 000064 00010000 29 TGBB0        .DATA.L  SINMN
30 000068 00010000 30 TOVI0        .DATA.L  SINMN
31          31 ;
32 000070   32          .ORG          H'000070
33 000070 00010000 33 TGAI1        .DATA.L  SINMN
34 000074 00010000 34 TGBI1        .DATA.L  SINMN
35 000078 00010000 35 TOVI1        .DATA.L  SINMN
36          36 ;
37 000080   37          .ORG          H'000080
38 000080 00010000 38 TGAI2        .DATA.L  SINMN
39 000084 00010000 39 TGBI2        .DATA.L  SINMN
40 000088 00010000 40 TOVI2        .DATA.L  SINMN
41          41 ;
42 000090   42          .ORG          H'000090
43 000090 00010000 43 TGAI3        .DATA.L  SINMN
44 000094 00010000 44 TGBI3        .DATA.L  SINMN
45 000098 00010000 45 TOVI3        .DATA.L  SINMN
46          46 ;

```

3. 応用編

```

47 0000A0          47          .ORG      H'0000A0
48 0000A0 00010000 48  TGAI4     .DATA.L  SINMN
49 0000A4 00010000 49  TGBI4     .DATA.L  SINMN
50 0000A8 00010000 50  TOVI4     .DATA.L  SINMN
51                51  ;
52 0000B0          52          .ORG      H'0000B0
53 0000B0 000100AC 53  DEND0     .DATA.L  SINWAVE
54 0000B4 00010000 54  DEND1     .DATA.L  SINMN
55 0000B8 00010000 55  DEND2     .DATA.L  SINMN
56 0000BC 00010000 56  DEND3     .DATA.L  SINMN
57                57  ;
58 0000D0          58          .ORG      H'0000D0
59 0000D0 00010000 59  ERI0      .DATA.L  SINMN
60 0000D4 00010000 60  RXI0      .DATA.L  SINMN
61 0000D8 00010000 61  TXI0      .DATA.L  SINMN
62 0000DC 00010000 62  TEND0     .DATA.L  SINMN
63 0000E0 00010000 63  ERI1      .DATA.L  SINMN
64 0000E4 00010000 64  RXI1      .DATA.L  SINMN
65 0000E8 00010000 65  TXI1      .DATA.L  SINMN
66 0000EC 00010000 66  TEND1     .DATA.L  SINMN
67 0000F0 00010000 67  ADI       .DATA.L  SINMN
68                68  ;
69                69  ;*****
70                70  ;*                               *
71                71  ;*          RAM ALLOCATION                          *
72                72  ;*                               *
73                73  ;*****
74                74  ;
75 FFFFEFC         75                .SECTION  RAM,DATA,LOCATE=H'FFFEFC
76                76  ;
77 FFFFEFC 00000004 77  SIN_ADR   .RES.L   1           ;Address of outupt data
78 FFFF00 00000004 78  SIN_DAT   .RES.L   1           ;Next data buff
79 FFFF04 00000001 79  SIN_MOD   .RES.B   1           ;Sin wave type
80 FFFF05 00000001 80  SIN_LWCOUN .RES.B   1           ;Count of 1 long ward data
81 FFFF06 00000001 81  SIN_FCOUNT .RES.B   1           ;Count of 100 bits data
82                82  ;
83 FFFF07 0000000D 83  OUT_DAT   .RES.B   D'13        ;Output data buff
84                84  ;
85                85  ;*****
86                86  ;*                               *
87                87  ;*          SYMBOL DEFINITIONS                          *
88                88  ;*                               *
89                89  ;*****
90                90  ;
91                91  ;***** DMAC0A *****
92                92  MAR0AR:  .EQU    H'FFFF20      ;Memory address register
93                93  ETCR0AH: .EQU    H'FFFF24      ;Transmit cunt register

```

```

94      00FFFF26      94 IOAR0A:      .EQU      H'FFFF26      ;I/O address register
95      00FFFF27      95 DTOR0A:      .EQU      H'FFFF27      ;Data transfer control
                                           ;register
96      00000007      96 DTE:          .EQU      7          ;Data transfer enable bit
97      ;
98      ;***** ITU *****
99      00FFFF60      99 TSTR:          .EQU      H'FFFF60      ;Timer start register
100     ;
101     ;***** ITU ch0 *****
102     00FFFF64      102 TCR0:          .EQU      H'FFFF64      ;timer control register
103     00FFFF65      103 TIOR0:          .EQU      H'FFFF65      ;timer I/O control register
104     00FFFF66      104 TIER0:          .EQU      H'FFFF66      ;timer interrupt enable
                                           ;register
105     00FFFF67      105 TSR0:          .EQU      H'FFFF67      ;timer status register
106     00FFFF68      106 TCNT0:          .EQU      H'FFFF68      ;timer counter register
107     00FFFF6A      107 GRA0H:          .EQU      H'FFFF6A      ;General register
108     ;
109     ;***** D/A *****
110     00FFFFDE      110 DACR:          .EQU      H'FFFFDE      ;D/A control register
111     ;
112     ;*****
113     ;*
114     ;*          MAIN PROGRAM : SINMN
115     ;*
116     ;*****
117     ;*
118     010000      118          .SECTION  PROG,CODE,LOCATE=H'010000
119     ;
120     00010000      120 SINMN:          .EQU      $
121     010000 7A0700FFFEFC 121          MOV.L      #H'FFFEFC,SP ;Initialize stack pointer
122     ;
123     010006 F8F0      123          MOV.B      #H'F0,R0L      ;Initialize RAM data
124     010008 3807      124          MOV.B      R0L,@OUT_DAT
125     01000A F8F0      125          MOV.B      #H'F0,R0L
126     01000C 3808      126          MOV.B      R0L,@OUT_DAT+1
127     01000E F8F0      127          MOV.B      #H'F0,R0L
128     010010 3809      128          MOV.B      R0L,@OUT_DAT+2
129     010012 F8F0      129          MOV.B      #H'F0,R0L
130     010014 380A      130          MOV.B      R0L,@OUT_DAT+3
131     010016 F8F0      131          MOV.B      #H'F0,R0L
132     010018 380B      132          MOV.B      R0L,@OUT_DAT+4
133     01001A F8F0      133          MOV.B      #H'F0,R0L
134     01001C 380C      134          MOV.B      R0L,@OUT_DAT+5
135     01001E F8F0      135          MOV.B      #H'F0,R0L
136     010020 380D      136          MOV.B      R0L,@OUT_DAT+6
137     010022 F8F0      137          MOV.B      #H'F0,R0L
138     010024 380E      138          MOV.B      R0L,@OUT_DAT+7

```

3. 応用編

```

139 010026 F8F0      139          MOV.B      #H'F0,R0L
140 010028 380F      140          MOV.B      R0L,@OUT_DAT+8
141 01002A F8F0      141          MOV.B      #H'F0,R0L
142 01002C 3810      142          MOV.B      R0L,@OUT_DAT+9
143 01002E F8F0      143          MOV.B      #H'F0,R0L
144 010030 3811      144          MOV.B      R0L,@OUT_DAT+10
145 010032 F8F0      145          MOV.B      #H'F0,R0L
146 010034 3812      146          MOV.B      R0L,@OUT_DAT+11
147 010036 F8F0      147          MOV.B      #H'F0,R0L
148 010038 3813      148          MOV.B      R0L,@OUT_DAT+12
149                    149          ;
150                    150          ;***** INITIALIZE ITU *****
151 01003A F820      151          MOV.B      #H'20,R0L
152 01003C 3864      152          MOV.B      R0L,@TCR0      ;Set non overlap time
153 01003E F801      153          MOV.B      #H'01,R0L
154 010040 3866      154          MOV.B      R0L,@TIER0      ;Initialize TIER0(1->IMIEA)
155 010042 7900FF00  155          MOV.W      #H'FF00,R0
156 010046 6B80FF6A  156          MOV.W      R0,@GRA0H      ;Initialize TCR0(1->CCLR1)
157 01004A 7F677200  157          BCLR      #0,@TSR0
158                    158          ;
159                    159          ;***** INITIALIZE D/A *****
160 01004E F840      160          MOV.B      #H'40,R0L
161 010050 38DE      161          MOV.B      R0L,@DACR      ;Initialize D/AC
162                    162          ;
163                    163          ;***** INITIALIZE DMAC *****
164 010052 7A0000FFFF07 164          MOV.L      #OUT_DAT,ER0 ;Set address of transmit data
165 010058 01006B80FFFC 165          MOV.L      ER0,@SIN_ADR
166 01005E 01006901    166          MOV.L      @ER0,ER1      ;load next data bit
167 010062 1031        167          SHLL.L    ER1
168 010064 01006B81FF00 168          MOV.L      ER1,@SIN_DAT
169 01006A 5840000C    169          BCC      SINMN01      ;Branch next data = '0'
170                    170          ;
171 01006E 7A000001019A 171          MOV.L      #S12USTADR,ER0 ;Output wave is 1200Hz
172 010074 FA01          172          MOV.B      #H'01,R2L
173 010076 58000008    173          BRA      SINMN02
174                    174          ;
175 01007A 7A00000101C2 175 SINMN01:  MOV.L      #S24STADR,ER0 ;Output wave is 2400Hz
176 010080 FA04          176          MOV.B      #H'04,R2L
177                    177          ;
178 010082 01006B80FF20 178 SINMN02:  MOV.L      ER0,@MAR0AR
179 010088 3A04          179          MOV.B      R2L,@SIN_MOD ;Store next wave type to
                                ;SIN_MOD
180 01008A F8DC        180          MOV.B      #H'DC,R0L
181 01008C 3826        181          MOV.B      R0L,@IOAR0A ;Set IOAR0A
182 01008E 79000014    182          MOV.W      #H'0014,R0
183 010092 6B80FF24    183          MOV.W      R0,@ETCR0AH ;Store transfer counter
184 010096 2827        184          MOV.B      @DTCR0A,R0L ;Load DTCR0A

```

```

185 010098 F808          185          MOV.B    #H'08,R0L
186 01009A 3827          186          MOV.B    R0L,@DTCR0A ;init. DTCR0A
187 01009C 7F277070      187          BSET     #DTE,@DTCR0A ;Start DMAC ch0 A
188                      188 ;
189 0100A0 F801          189          MOV.B    #H'01,R0L ;Initialize counter
190 0100A2 3805          190          MOV.B    R0L,@SIN_LWCOUN
191 0100A4 3806          191          MOV.B    R0L,@SIN_FCOUN
192                      192 ;
193 0100A6 3860          193          MOV.B    R0L,@TSTR ;Timer start
194 0100A8 0700          194          LDC.B    #0,CCR ;I flag clr
195                      195 ;
196 0100AA 40FE          196 SINMN99:  BRA     SINMN99
197                      197 ;
198                      198 ;*****
199                      199 ;* *
200                      200 ;* NAME : SINWAVE(Set next data of sin wave DMAC) *
201                      201 ;* *
202                      202 ;*****
203                      203 ;* *
204                      204 ;* ENTRY : SIN_ADR (Address of output data) *
205                      205 ;* ENTRY : SIN_DAT (Next data buff) *
206                      206 ;* ENTRY : SIN_MOD (Sin wave type) *
207                      207 ;* ENTRY : SIN_LWCOUN (Count of 1 long ward data) *
208                      208 ;* ENTRY : SIN_FCOUN (Count of 100 bits data) *
209                      209 ;* RETURNS : NOTHING *
210                      210 ;* *
211                      211 ;*****
212                      212 ;
213          000100AC      213 SINWAVE:  .EQU    $
214 0100AC 01006DF0      214          PUSH.L  ER0 ;Escape register
215 0100B0 01006DF1      215          PUSH.L  ER1
216                      216 ;
217 0100B4 7F677200      217          BCLR.B  #0,@TSR0 ;Clear time comparemech flag
218                      218 ;
219 0100B8 79000014      219          MOV.W    #H'0014,R0 ;Set transfer data inning
220 0100BC 6B80FF24      220          MOV.W    R0,@ETCR0AH
221 0100C0 2827          221          MOV.B    @DTCR0A,R0L
222                      222 ;
223 0100C2 01006B00FF00  223          MOV.L    @SIN_DAT,ER0 ;Load next data bit
224 0100C8 1030          224          SHLL.L  ER0
225 0100CA 01006B80FF00  225          MOV.L    ER0,@SIN_DAT
226 0100D0 6701          226          BST     #0,R1H
227                      227 ;
228 0100D2 2904          228          MOV.B    @SIN_MOD,R1L
229 0100D4 A901          229          CMP.B    #H'01,R1L ;SIN_MOD = '1' ?
230 0100D6 5860000A      230          BNE     SINWAVE01 ;No
231 0100DA 7301          231          BTST   #0,R1H ;Next data = '1' ?

```

3. 応用編

```

232 0100DC 5870003C      232          BEQ      SINWAVE05      ;Yes
233 0100E0 5800002A      233          BRA      SINWAVE04      ;No
234                      234      ;
235 0100E4 A902          235 SINWAVE01:  CMP.B   #H'02,R1L      ;SIN_MOD = '2' ?
236 0100E6 5860000A      236          BNE      SINWAVE02      ;No
237 0100EA 7301          237          BTST    #0,R1H      ;Next data = '1' ?
238 0100EC 58700048      238          BEQ      SINWAVE07      ;Yes
239 0100F0 58000036      239          BRA      SINWAVE06      ;No
240                      240      ;
241 0100F4 A904          241 SINWAVE02:  CMP.B   #H'04,R1L      ;SIN_MOD = '4' ?
242 0100F6 5860000A      242          BNE      SINWAVE03      ;No
243 0100FA 7301          243          BTST    #0,R1H      ;Next data = '1' ?
244 0100FC 58700038      244          BEQ      SINWAVE07      ;No
245 010100 58000026      245          BRA      SINWAVE06      ;Yes
246                      246      ;
247 010104 7301          247 SINWAVE03:  BTST    #0,R1H      ;Next data = '1' ?
248 010106 58700012      248          BEQ      SINWAVE05      ;No
249 01010A 58000000      249          BRA      SINWAVE04      ;Yes
250                      250      ;
251 01010E 7A00000101AE   251 SINWAVE04:  MOV.L   #S12LSTADR,ER0 ;Set MAR0AR
252 010114 F908          252          MOV.B   #H'08,R1L      ;Set DTCR0A
253 010116 F102          253          MOV.B   #H'02,R1H      ;Set SIN_MOD
254 010118 58000026      254          BRA      SINWAVE08
255                      255      ;
256 01011C 7A00000101D5   256 SINWAVE05:  MOV.L   #S24EDADR,ER0 ;Set MAR0AR
257 010122 F928          257          MOV.B   #H'28,R1L      ;Set DTCR0
258 010124 F108          258          MOV.B   #H'08,R1H      ;Set SIN_MOD
259 010126 58000018      259          BRA      SINWAVE08
260                      260      ;
261 01012A 7A000001019A   261 SINWAVE06:  MOV.L   #S12USTADR,ER0 ;Set MAR0AR
262 010130 F908          262          MOV.B   #H'08,R1L      ;Set DTCR0A
263 010132 F101          263          MOV.B   #H'01,R1H      ;Set SIN_MOD
264 010134 5800000A      264          BRA      SINWAVE08
265                      265      ;
266 010138 7A00000101C2   266 SINWAVE07:  MOV.L   #S24STADR,ER0 ;Set MAR0AR
267 01013E F908          267          MOV.B   #H'08,R1L      ;Set DTCR0A
268 010140 F104          268          MOV.B   #H'04,R1H      ;Set SIN_MOD
269                      269      ;
270 010142 3104          270 SINWAVE08:  MOV.B   R1H,@SIN_MOD
271 010144 01006B80FF20   271          MOV.L   ER0,@MAR0AR
272 01014A 3927          272          MOV.B   R1L,@DTCR0A
273 01014C 7F667000      273          BSET.B  #0,@TIER0
274 010150 7F277070      274          BSET.B  #DTE,@DTCR0A
275                      275      ;
276 010154 2805          276          MOV.B   @SIN_LWCOUN,R0L ;Increment counter
277 010156 0A08          277          INC.B   R0L
278 010158 3805          278          MOV.B   R0L,@SIN_LWCOUN

```



```

279                                279 ;
280 01015A A820                    280          CMP.B    #32,R0L      ;32bit out end ?
281 01015C 58600020                281          BNE     SINWAVE09   ;No
282 010160 F800                    282          MOV.B    #0,R0L      ;Clear counter
283 010162 3805                    283          MOV.B    R0L,@SIN_LWCOUN
284                                284 ;
285 010164 01006B00FFFC            285          MOV.L    @SIN_ADR,ER0 ;Set next data address
286 01016A 7A1000000004            286          ADD.L    #4,ER0
287 010170 01006B80FEFC            287          MOV.L    ER0,@SIN_ADR
288                                288 ;
289 010176 01006900                289          MOV.L    @ER0,ER0      ;Set next data
290 01017A 01006B80FF00            290          MOV.L    ER0,@SIN_DAT
291                                291 ;
292 010180 2806                    292 SINWAVE09: MOV.B    @SIN_FCOUNT,R0L ;Increment counter
293 010182 0A08                    293          INC.B    R0L
294 010184 3806                    294          MOV.B    R0L,@SIN_FCOUNT
295                                295 ;
296 010186 A864                    296          CMP.B    #100,R0L     ;100bit out end ?
297 010188 58600004                297          BNE     SINWAVE99   ;No
298                                298 ;
299 01018C F800                    299          MOV.B    #0,R0L      ;Yes,Stop timer
300 01018E 3860                    300          MOV.B    R0L,@TSTR
301                                301 ;
302 010190 01006D71                302 SINWAVE99: POP.L    ER1
303 010194 01006D70                303          POP.L    ER0
304 010198 5670                    304          RTE
305                                305 ;
306                                306 ;*****
307                                307 ;*
308                                308 ;*          DATA TABLE (OUTPUT DATA TAB)
309                                309 ;*
310                                310 ;*****
311                                311 ;
312 01019A 8093A7B9CA                312 S12USTADR  .DATA.B H'80,H'93,H'A7,H'B9,H'CA
313 01019F DAE7F1F9FD                313          .DATA.B H'DA,H'E7,H'F1,H'F9,H'FD
314 0101A4 FFFDF9F1E7                314          .DATA.B H'FF,H'FD,H'F9,H'F1,H'E7
315 0101A9 DACAB9A793                315          .DATA.B H'DA,H'CA,H'B9,H'A7,H'93
316 0101AE 806C584635                316 S12LSTADR  .DATA.B H'80,H'6C,H'58,H'46,H'35
317 0101B3 25180E0602                317          .DATA.B H'25,H'18,H'0E,H'06,H'02
318 0101B8 0002060E18                318          .DATA.B H'00,H'02,H'06,H'0E,H'18
319 0101BD 253546586C                319          .DATA.B H'25,H'35,H'46,H'58,H'6C
320 0101C2 80A7CAE7F9                320 S24STADR  .DATA.B H'80,H'A7,H'CA,H'E7,H'F9
321 0101C7 FFF9E7CAA7                321          .DATA.B H'FF,H'F9,H'E7,H'CA,H'A7
322 0101CC 8058351806                322          .DATA.B H'80,H'58,H'35,H'18,H'06
323 0101D1 00061835                323          .DATA.B H'00,H'06,H'18,H'35
324 0101D5 58                      324 S24EDADR  .DATA.B H'58
325                                325 ;

```

3. 応用編

```
326                                326                .END
*****TOTAL  ERRORS              0
*****TOTAL  WARNINGS            0
```

3.4 4相ステッピングモータ応用例

MCU : H8/3003

使用機器 : ITU、TPC、DMAC

3.4.1 仕様

- 図 3.24 に示すように H8/3003 の内蔵機能のうち、ITU、TPC、DMAC を使用し、4 相のステッピングモータを制御します。
- ステッピングモータは 2 相励磁方式で制御します。
- 本タスクでは、ステッピングモータを停止 正転 停止 逆転 停止の動作を繰り返します。
- 本タスクではソフトウェアの介在なしに、スルーアップおよびスルーダウンの処理を行います。
- ドライバの保護のため貫通電流防止期間を設けます。

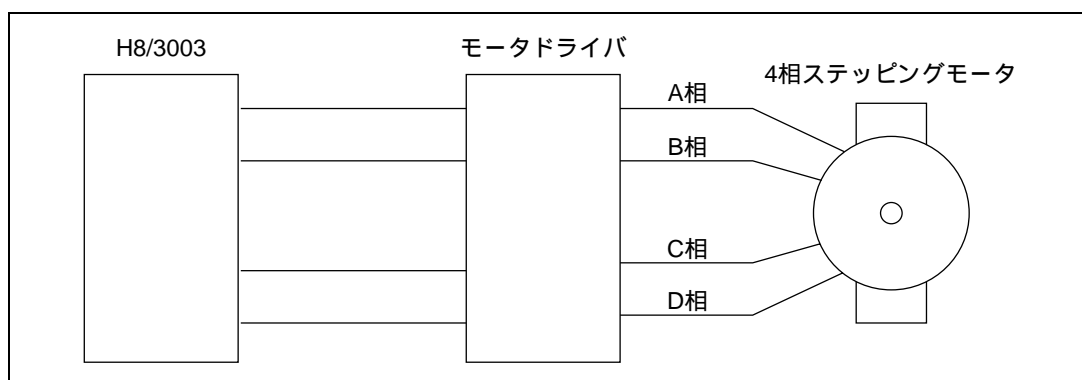


図 3.24 4相ステッピングモータ制御回路図

3.4.2 考え方

- ステッピングモータ動作例

図3.25に2相励磁で4相ステッピングモータを動作させる例を示します。動作概要は以下のとおりです。

- 図3.25に示すようにパルスが“H”になっているときに、対応する相を励磁します。
- では、D相とA相を同時に励磁します。このときロータは、D相とA相の間に位置します。
- では、A相とB相を同時に励磁します。このときロータは、A相とB相の間に位置します。以下、2相励磁方式は、隣り合う2つの相(A、D相、A、B相、B、C相、C、D相)を励磁し、ロータを回転させます。
- 逆転動作の場合は、D、C相 C、B相 B、A相 A、D相を順番に励磁することでステッピングモータを逆転させます。
- 停止動作は、正転動作の最後の相または、逆転動作の最後の相を一定時間励磁し続けることでステッピングモータを停止させます。

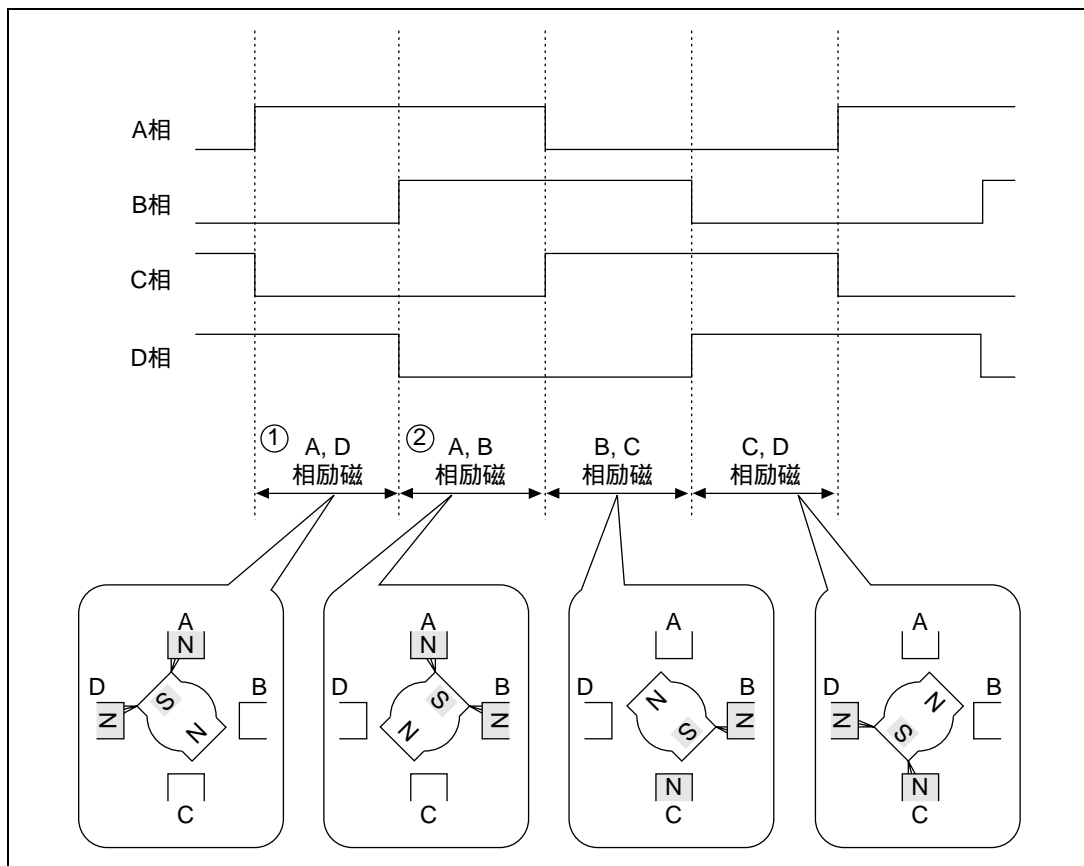


図 3.25 ステッピングモータ動作例

● ノンオーバーラップ時間

図3.26に示すように出力パターン切り替え時に貫通電流防止期間 n （ノンオーバーラップ時間）を挿入します。励磁相の切り替え時に生じるターンオフの遅れにより、ドライバが破壊する危険性があり、これを防ぐためにノンオーバーラップ時間を挿入し、時間遅れをもたせて対策します。

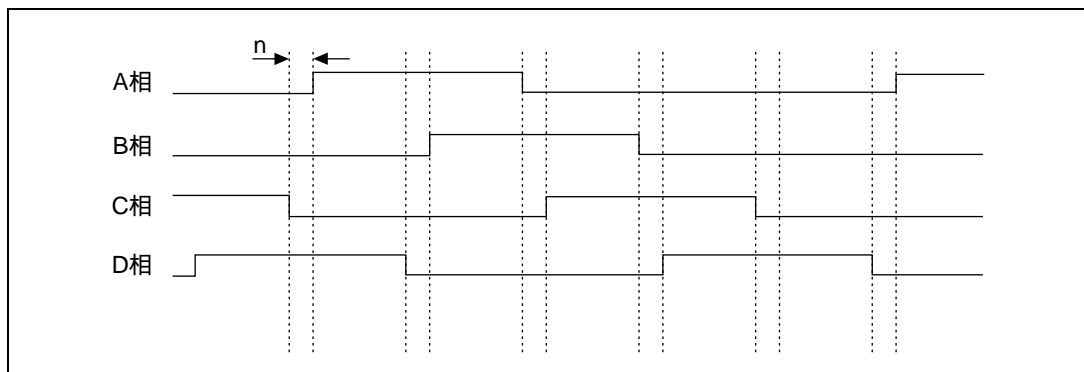


図 3.26 ノンオーバーラップ時間出力例

- スルーアップ、スルーダウン動作

図3.27に示すような加減速制御されたパルスを出力します。スルーアップ、スルーダウン動作を行うことにより、モータの脱調を防止します。

モータを動作させる際に、急に周期の短いパルスを出力すると、モータは負荷に追いつけず、回転しないことがあります。これを防止する対策としてスルーアップおよびスルーダウンを行います。

動作原理を以下に示します。

- 徐々にパルス周期を短くし、設定した量のパルスを出力する。(スルーアップ)
- 一定のパルス周期で設定した量のパルスを出力する。(定速)
- 徐々にパルス周期を長くし、設定した量のパルスを出力する。(スルーダウン)

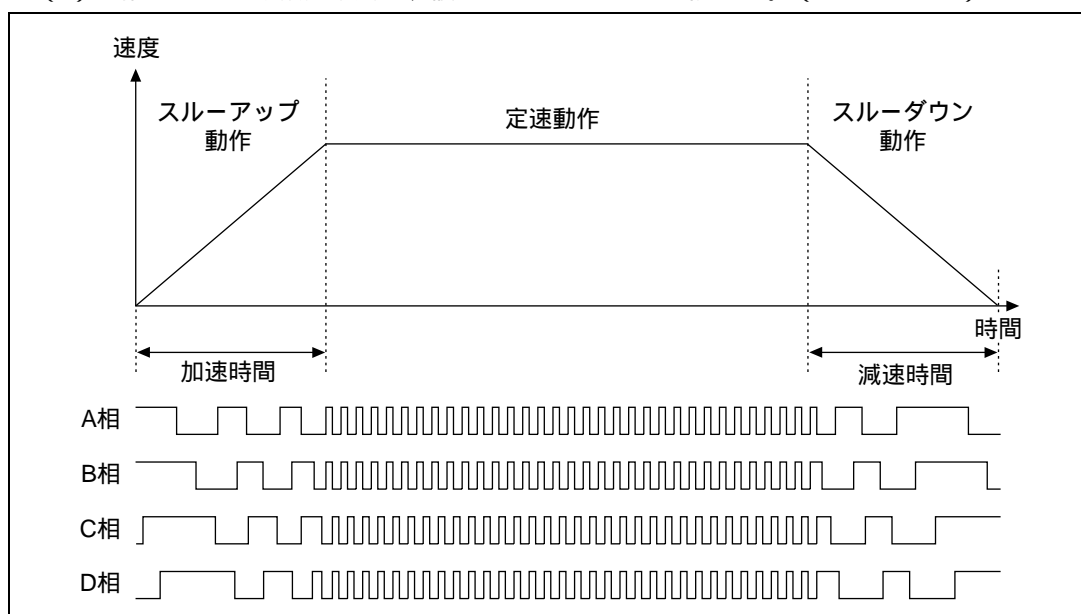


図 3.27 スルーアップ、スルーダウン動作

3.4.3 使用機能説明

図 3.28 に本タスクで使用する H8/3003 の内蔵機能を示します。

図 3.28 に示すように DMAC を 2 チャンネル使用し、4 相パルス出力パターンを TPC し、4 相のパルスを出力します。また、パルス周期を ITU の GRB に転送し、4 相パルスの出力タイミングを制御します。

- 周期データテーブル：ステップ周期を変更するためのテーブル (ITU のタイマ値) を ROM 上に設定します。
- 出力パターンテーブル：4 相パルスを出力するための出力パターンを ROM 上に設定します。
- DMAC0A：ITU のコンペアマッチ A で起動します。
出力パターンデータテーブルから出力パターンを TPC の NDR に転送します。
- DMAC0B：ITU のコンペアマッチ A で起動します。
周期データテーブルからパルスの周期データを ITU の GRB に転送します。
- ITU
 - コンペアマッチ A：DMAC0A、B および TPC を起動します。
 - コンペアマッチ B：タイマカウンタのクリアを行います。また、TPC を起動します。

3. 応用編

- TPC :
 - DMAC を使用することにより CPU の介在なくデータを出力します。
 - 最大 16 ビットデータを出力することが可能ですが、本タスクでは 4 ビットのデータを出力します。
 - ノンオーバーラップ期間をもったパルスを出力することが可能です。
 - コンペアマッチ B : 波形が “H” から “L” に変化するパルスの出力を行います。“L” から “H” に変化する出力は保留します。
 - コンペアマッチ A : コンペアマッチ B で保留された “L” から “H” の出力を行います。(GRA の設定値分遅延します)

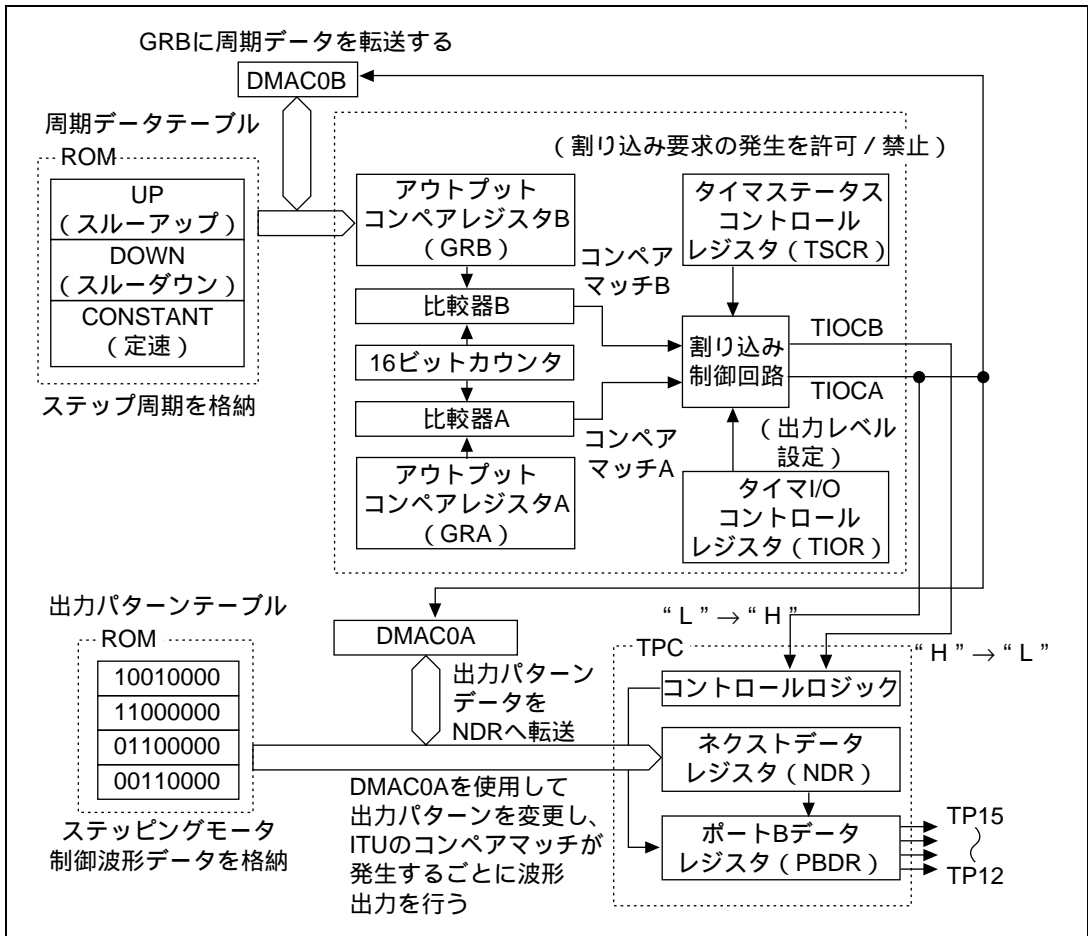


図 3.28 4相ステップングモータ制御ブロック図

表 3.19 に本タスク例の機能割り付けを示します。表 3.19 に示すように H8/3003 の機能を割り付け、2 相励磁のステッピングモータ出力波形を生成します。

表 3.19 H8/3003 機能割り付け

H8/3003 の機能		機能
ITU (ch0)	TCNT0	16 ビットカウンタ
	GRA0	アウトプットコンペアレジスタ A
	GRB0	アウトプットコンペアレジスタ B
	TCR0	カウントクロックの選択およびカウンタクリア要因の選定
	TIOR0	GRA および GRB をアウトプットコンペアレジスタに設定
	TIER0	割り込み要求の許可 / 禁止を制御
	TSR0	コンペアマッチやオーバフローのステータスレジスタ
	TSTR	TCNT のカウント動作の許可 / 禁止を設定
DMAC0A/B	DTCR0A、B	DMAC の各チャンネルの動作を制御
	MAR0A、B	転送元アドレスを設定
	IORA0A、B	転送先アドレスを設定
	ETCR0A、B	転送回数を設定
TPC	PBDDR	TPC 出力端子を設定
	PBDR	TPC 出力 B の出力データを格納
	TPMR	TPC 出力 B のノンオーバーラップモードを設定
	TPCR	TPC 出力の出力トリガ信号を設定
	NDERB	TPC 出力 B の許可 / 禁止を設定
	NDRB	TPC 出力 B の次に出力するデータを格納
	TP15 ~ TP12	4 相のステッピングモータ出力波形を発生

3.4.4 動作説明

4相パルス出力例（ノンオーバーラップあり）

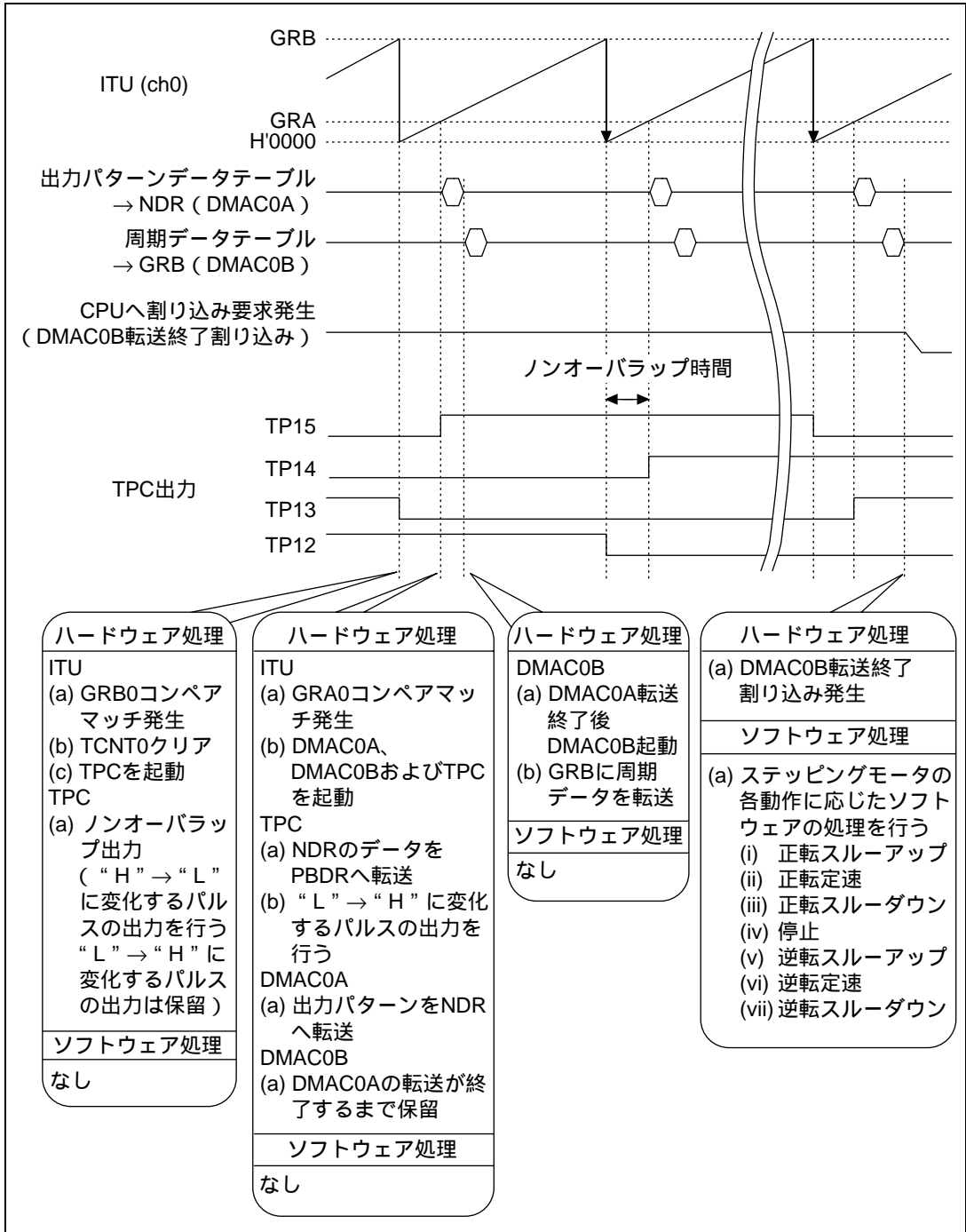


図 3.29 ステッピングモータ・タイミング説明

3.4.5 ソフトウェア説明

表 3.20 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	STP4MN	ITU、TPC および DMAC0A、B の初期設定を行い、正転のスルーアップ動作を行う。
モータ制御	STP41	<p>モータの各動作（正転定速、停止など）を判定し、制御を行う。</p> <ul style="list-style-type: none"> • DMAC0A、B の設定を行い、正転のスルーアップ動作を行う。 • DMAC0B の設定を行い、正転の定速動作を行う。 • DMAC0A、B の設定を行い、正転のスルーダウン動作を行う。 • DMAC0A、B の設定を行い、停止動作を行う。 • DMAC0A、B の設定を行い、逆転のスルーアップ動作を行う。 • DMAC0B の設定を行い、逆転の定速動作を行う。 • DMAC0A、B の設定を行い、逆転のスルーダウン動作を行う。 • DMAC0A、B の設定を行い、停止動作を行う。

3. 応用編

表 3.21 使用内部レジスタの説明

内蔵機能	レジスタ名	機能
ITU	GRA0	ノンオーバーラップ時間を設定する
	GRB0	タイマ周期を設定する
	TIER0	IMFA 割り込みを許可する
	TCR0	ITU を以下のように設定する <ul style="list-style-type: none"> GRB のコンペアマッチでカウンタクリア 内部クロック でカウント
	TIOR0	GRA および GRB をアウトプットコンペアレジスタに設定し、端子出力を禁止する
	TSTR	TCNT0 のカウント動作を許可する
DMAC0A	DTCR0A	DMAC0A を以下のように設定する <ul style="list-style-type: none"> データサイズをバイトサイズに設定 MAR をインクリメントに設定 データ転送をリピートモードに設定 転送の許可および禁止を選択（モジュールによって異なる） 起動要因を ITUch0 のコンペアマッチ A に設定
	MAR0A	出力パターンデータテーブルの転送元アドレスを設定する
	IOAR0A	NDRB のアドレス（転送先）を設定する
	ETCR0A	転送回数を設定する
DMAC0B	DTCR0B	DMAC0B を以下のように設定する <ul style="list-style-type: none"> データサイズをワードサイズに設定 MAR をインクリメントまたはデクリメントに設定（処理内容によって異なる） データ転送を I/O モードに設定（処理内容によってアイドルモードに設定）転送許可 割り込み要求（DEND）を許可 起動要因を ITUch0 のコンペアマッチ A に設定
	MAR0B	周期データテーブルの転送元アドレスを設定する
	IOAR0B	GRB のアドレス（転送先）を設定する
	ETCR0B	転送回数を設定する
TPC	PBDDR	TP15～TP12 までの TPC 出力を許可する
	PBDR	出力パターンデータを格納する
	TPMR	TP15～TP12 をノンオーバーラップ出力に設定する
	TPCR	TP15～TP12 の出力トリガを ITUch0 のコンペアマッチに設定する
	NDRB	TPC 出力 TP15～TP12 を許可する
	NDRB	次の出力パターンデータを格納する

- 引数の説明
本タスクでは引数を使用していません。

表 3.22 使用 RAM 説明

ラベル名、レジスタ名	機能	データ長	使用モジュール名
CTRLMD	4相ステップモータの動作状態を示します。 H'01：正転スルーアップ動作終了 H'02：正転定速動作終了 H'03：正転スルーダウン動作終了 H'00：正転停止動作終了 H'81：逆転スルーアップ動作終了 H'82：逆転定速動作終了 H'83：逆転スルーダウン動作終了 H'84：逆転停止動作終了	1バイト	モータ制御

表 3.23 データテーブル説明

テーブル名	機能	データ長	データ容量
PATTBL	4相パルスを出力するためのパターンを設定します。	バイト	4バイト
UPTBL DOWNTBL CNSTBL	ステップ周期を変更するためのデータを設定します。	ワード	121ワード

31	ERn	RnH	RnL	0
0	全ルーチンにてデータ設定時のワークとして使用			
1				
2				
3				
4				
5				
6				
7				

図 3.30 使用汎用レジスタ

3.4.6 フローチャート

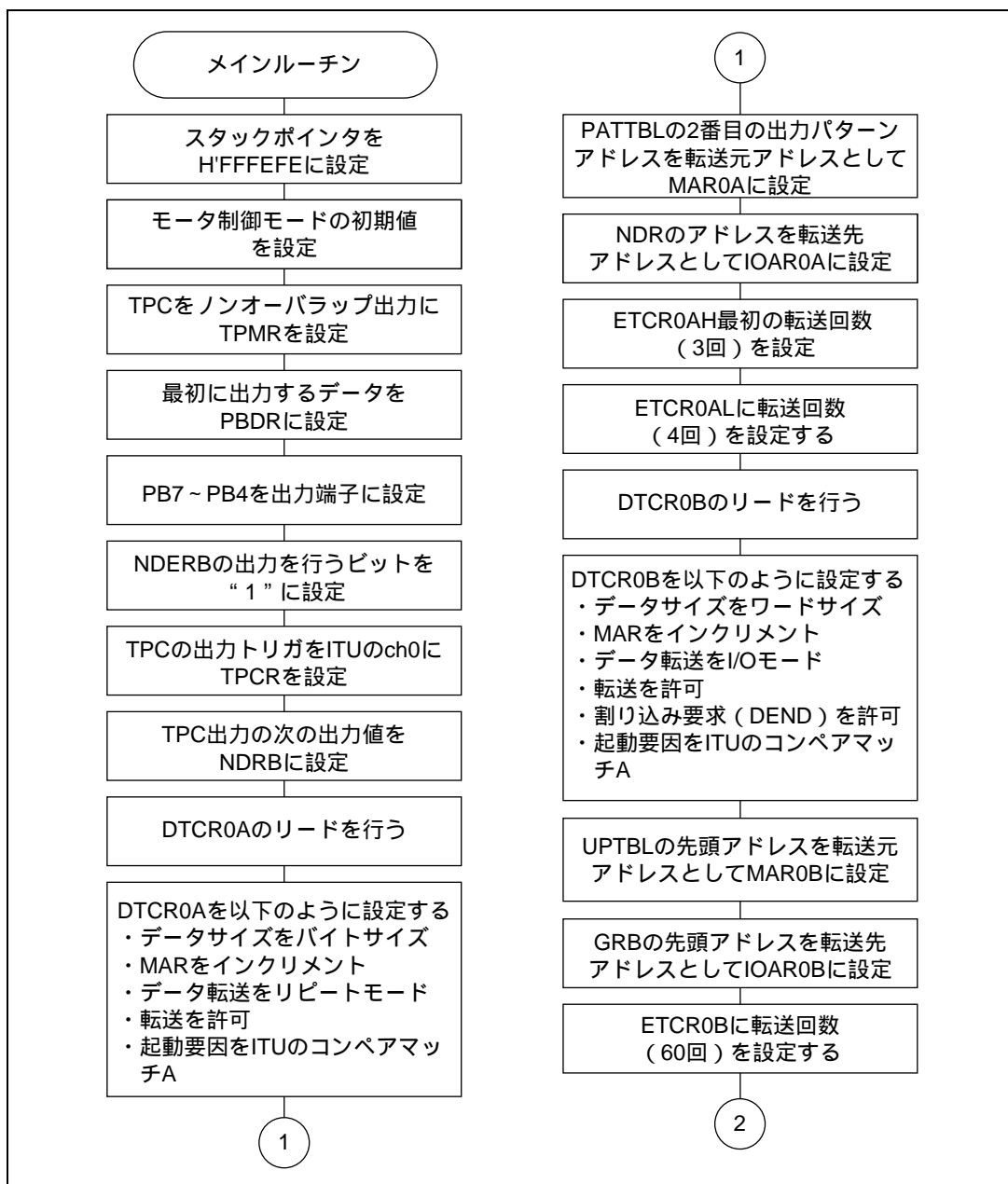


図 3.31 メインルーチン (1)

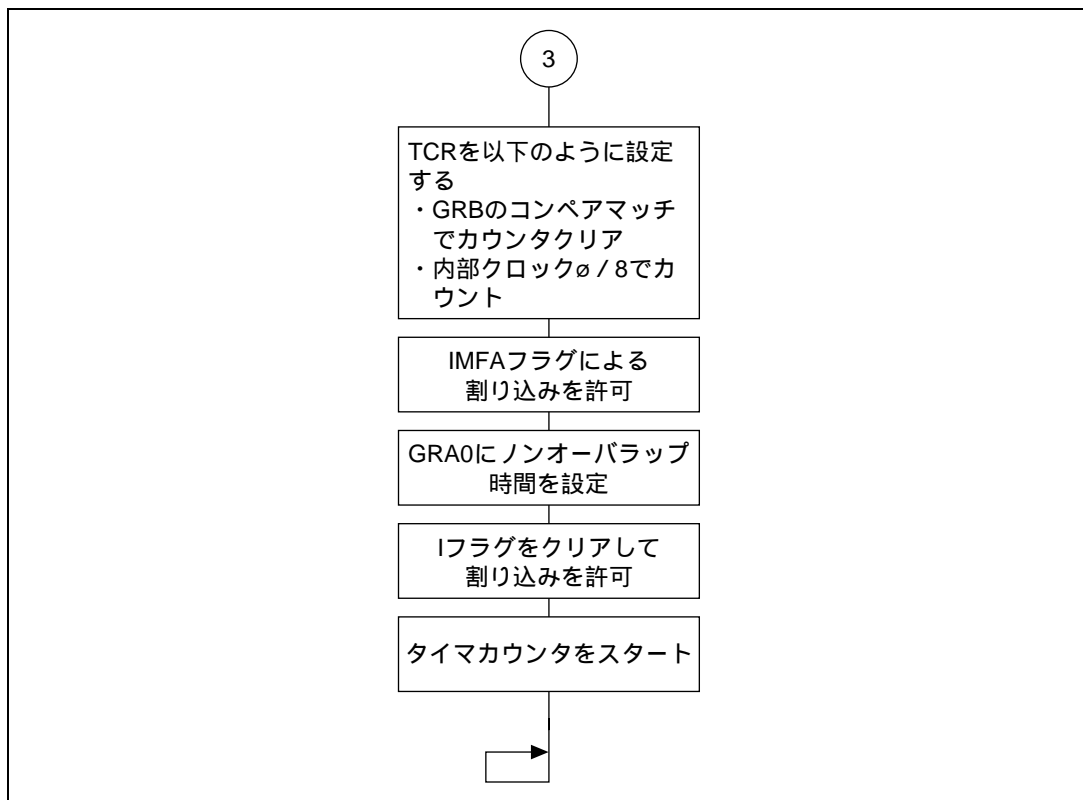


図 3.31 メインルーチン (2)

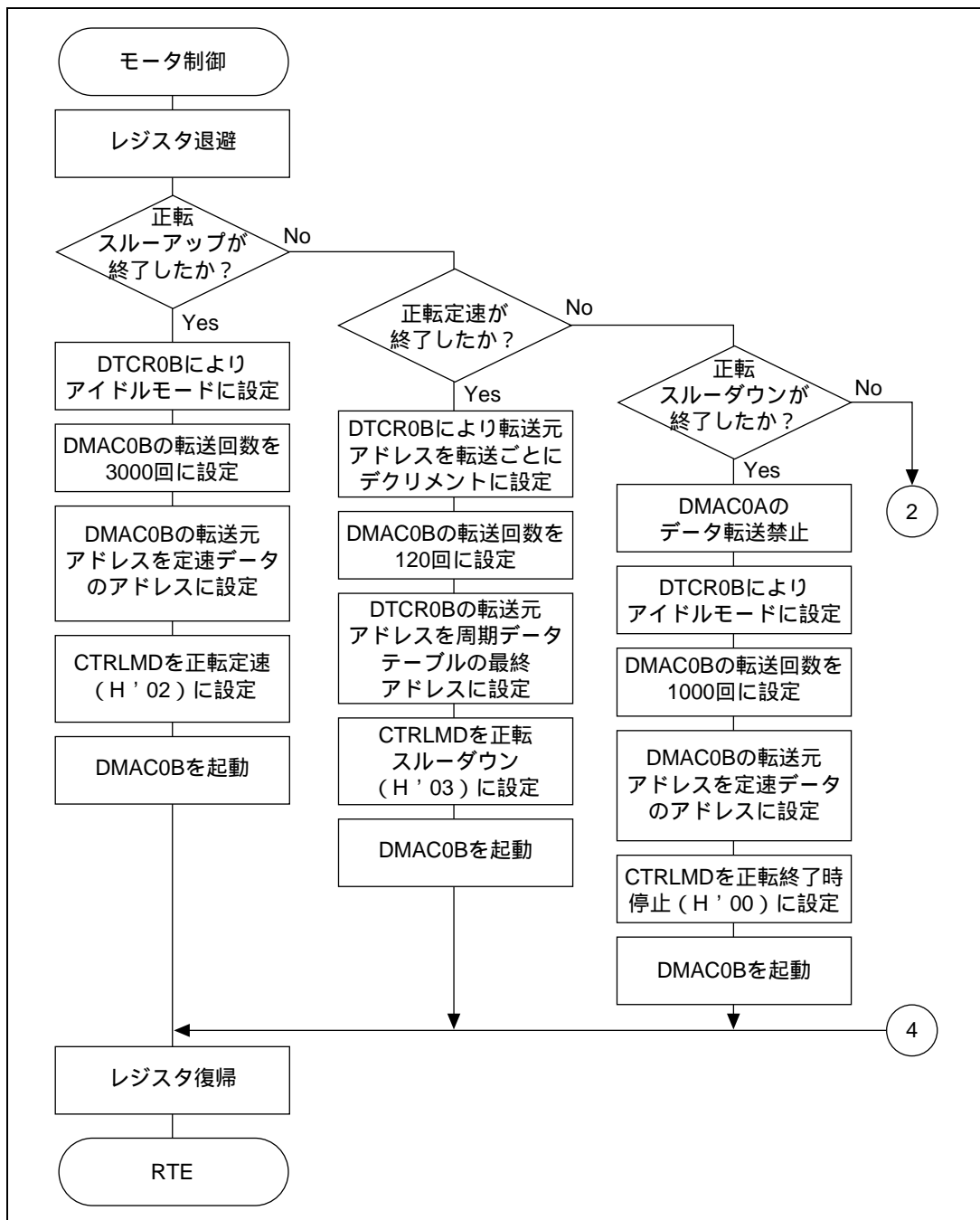


図 3.32 モータ制御 (1)

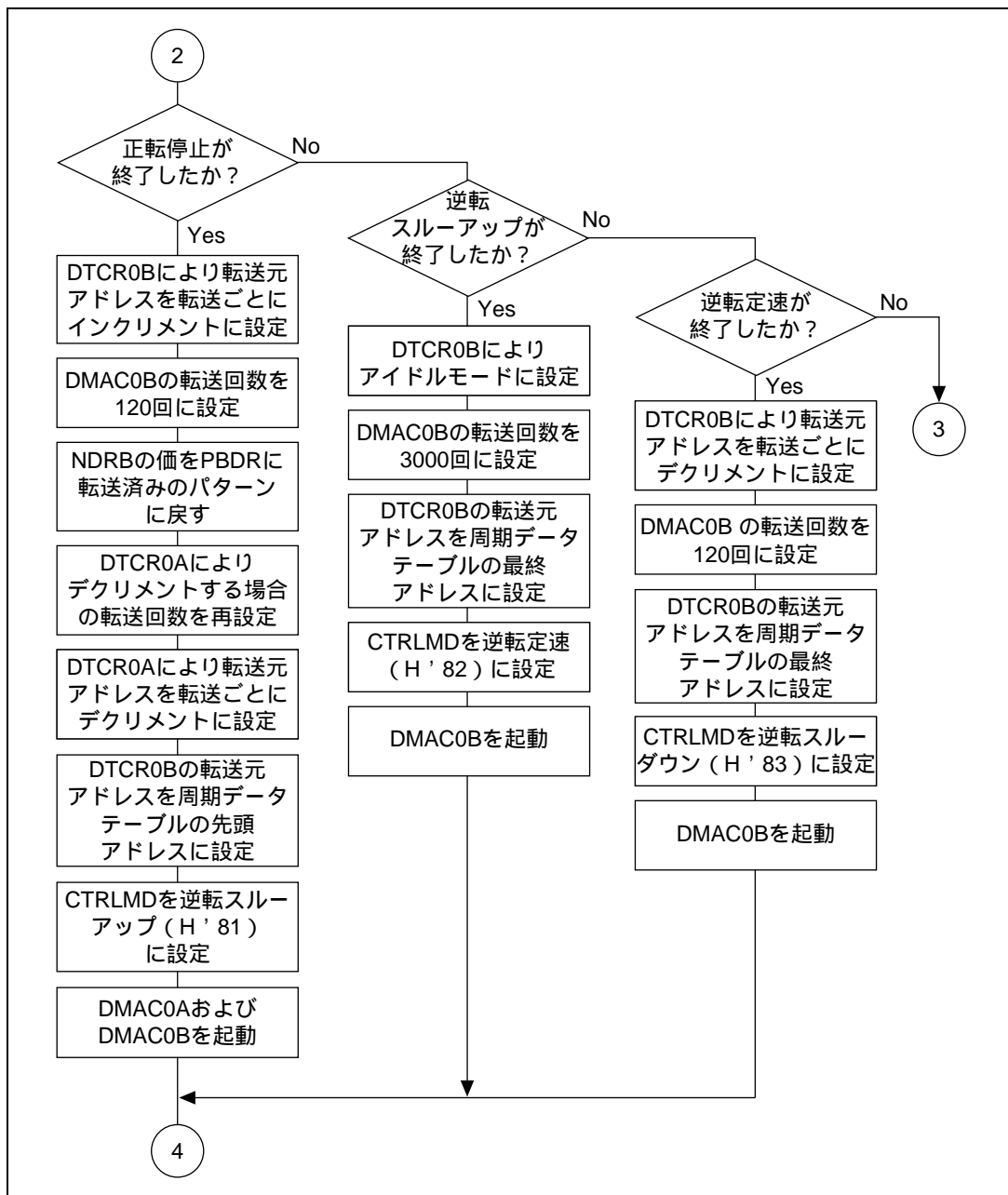


図 3.32 モータ制御 (2)

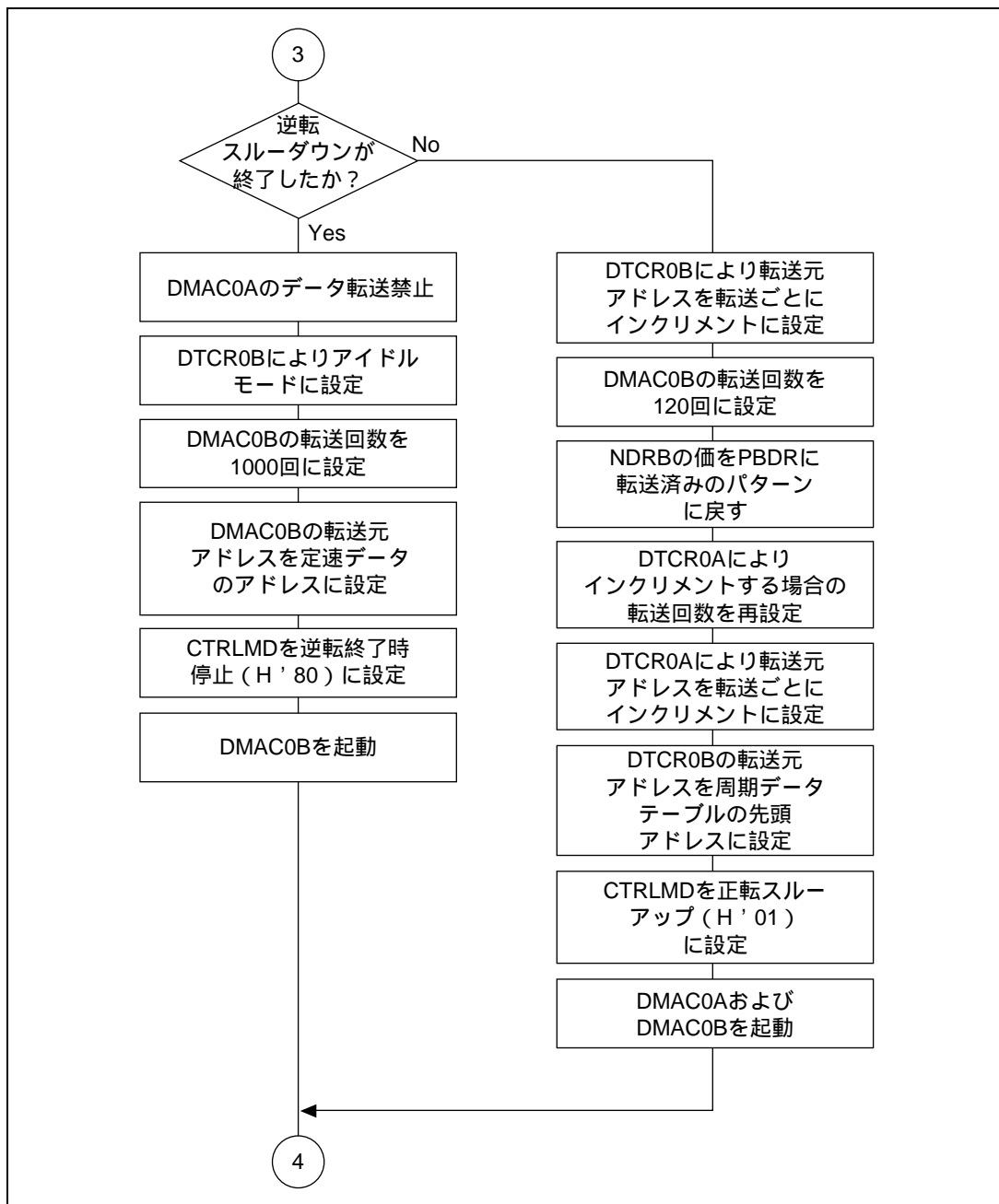


図 3.32 モータ制御 (3)

3.4.7 プログラムリスト

```

1          1  ;*****
2          2  ;*
3          3  ;*          VECTOR ADDRESS
4          4  ;*
5          5  ;*****
6          6  ;
7  000000   7          .SECTION   VECT, CODE, LOCATE=H'000000
8          8  ;
9  000000   00010000   9  RES          .DATA.L   STP4MN
10         10 ;
11  00001C   11          .ORG          H'00001C
12  00001C   00010000   12 NMI          .DATA.L   STP4MN
13  000020   00010000   13 TRAPA0       .DATA.L   STP4MN
14  000024   00010000   14 TRAPA1       .DATA.L   STP4MN
15  000028   00010000   15 TRAPA2       .DATA.L   STP4MN
16  00002C   00010000   16 TRAPA3       .DATA.L   STP4MN
17         17 ;
18  000030   18          .ORG          H'000030
19  000030   00010000   19 IRQ0         .DATA.L   STP4MN
20  000034   00010000   20 IRQ1         .DATA.L   STP4MN
21  000038   00010000   21 IRQ2         .DATA.L   STP4MN
22  00003C   00010000   22 IRQ3         .DATA.L   STP4MN
23  000040   00010000   23 IRQ4         .DATA.L   STP4MN
24  000044   00010000   24 IRQ5         .DATA.L   STP4MN
25  000048   00010000   25 IRQ6         .DATA.L   STP4MN
26  00004C   00010000   26 IRQ7         .DATA.L   STP4MN
27  000050   00010000   27 WOVI0        .DATA.L   STP4MN
28  000054   00010000   28 CMI          .DATA.L   STP4MN
29         29 ;
30  000060   30          .ORG          H'000060
31  000060   00010000   31 IMIA0        .DATA.L   STP4MN
32  000064   00010000   32 IMIB0        .DATA.L   STP4MN
33  000068   00010000   33 OVI0         .DATA.L   STP4MN
34         34 ;
35  000070   35          .ORG          H'000070
36  000070   00010000   36 IMIA1        .DATA.L   STP4MN
37  000074   00010000   37 IMIB1        .DATA.L   STP4MN
38  000078   00010000   38 OVI1         .DATA.L   STP4MN
39         39 ;
40  000080   40          .ORG          H'000080
41  000080   00010000   41 IMIA2        .DATA.L   STP4MN
42  000084   00010000   42 IMIB2        .DATA.L   STP4MN
43  000088   00010000   43 OVI2         .DATA.L   STP4MN
44         44 ;
45  000090   45          .ORG          H'000090
46  000090   00010000   46 IMIA3        .DATA.L   STP4MN

```

3. 応用編

```

47 000094 00010000      47 IMIB3      .DATA.L      STP4MN
48 000098 00010000      48 OVI3      .DATA.L      STP4MN
49                                49 ;
50 0000A0                                50          .ORG          H'0000A0
51 0000A0 00010000      51 IMIA4      .DATA.L      STP4MN
52 0000A4 00010000      52 IMIB4      .DATA.L      STP4MN
53 0000A8 00010000      53 OVI4      .DATA.L      STP4MN
54                                54 ;
55 0000B0                                55          .ORG          H'0000B0
56 0000B0 00010000      56 DEND0A     .DATA.L      STP4MN
57 0000B4 0001007C      57 DEND0B     .DATA.L      STP41
58 0000B8 00010000      58 DEND1A     .DATA.L      STP4MN
59 0000BC 00010000      59 DEND1B     .DATA.L      STP4MN
60 0000C0 00010000      60 DEND2A     .DATA.L      STP4MN
61 0000C4 00010000      61 DEND2B     .DATA.L      STP4MN
62 0000C8 00010000      62 DEND3A     .DATA.L      STP4MN
63 0000CC 00010000      63 DEND3B     .DATA.L      STP4MN
64 0000D0 00010000      64 ERI0      .DATA.L      STP4MN
65 0000D4 00010000      65 RXI0      .DATA.L      STP4MN
66 0000D8 00010000      66 TXI0      .DATA.L      STP4MN
67 0000DC 00010000      67 TEI0      .DATA.L      STP4MN
68 0000E0 00010000      68 ERI1      .DATA.L      STP4MN
69 0000E4 00010000      69 RXI1      .DATA.L      STP4MN
70 0000E8 00010000      70 TXI1      .DATA.L      STP4MN
71 0000EC 00010000      71 TEI1      .DATA.L      STP4MN
72 0000F0 00010000      72 ADI       .DATA.L      STP4MN
73                                73 ;
74                                74 ;*****
75                                75 ;*
76                                76 ;*          RAM ALLOCATION          *
77                                77 ;*
78                                78 ;*****
79                                79 ;
80 FFFF00                                80          .SECTION     RAM,DATA,LOCATE=H'FFFF00
81                                81 ;
82 FFFF00 00000001      82 CTRLMD     .RES.B      1          ;Motor control mode
83                                83 ;
84                                84 ;*****
85                                85 ;*
86                                86 ;*          SYMBOL DEFINITIONS          *
87                                87 ;*
88                                88 ;*****
89                                89 ;
90                                90 ;***** DMAC0A *****
91                                91 ;
92          00FFFF21      92 MAR0A     .EQU          H'FFFF21          ;Memory address register 0A

```

93	00FFFF24	93	ETCR0AH	.EQU	H'FFFF24	;Execute transfer control ;register 0AH
94	00FFFF25	94	ETCR0AL	.EQU	H'FFFF25	;Execute transfer control ;register 0AL
95	00FFFF26	95	IOAR0A	.EQU	H'FFFF26	;I/O address register 0A
96	00FFFF27	96	DTCR0A	.EQU	H'FFFF27	;Data transfer control ;register 0A
97		97	;			
98		98	;		***** DMAC0B *****	
99		99	;			
100	00FFFF29	100	MAR0B	.EQU	H'FFFF29	;Memory address register 0B
101	00FFFF2C	101	ETCR0B	.EQU	H'FFFF2C	;Execute transfer control ;register 0B
102	00FFFF2E	102	IOAR0B	.EQU	H'FFFF2E	;I/O address register 0B
103	00FFFF2F	103	DTCR0B	.EQU	H'FFFF2F	;Data transfer control ;register
104		104	;			
105		105	;		***** ITU ch0 *****	
106		106	;			
107	00FFFF60	107	TSTR	.EQU	H'FFFF60	;Timer start register
108	00FFFF64	108	TCR0	.EQU	H'FFFF64	;Timer control register0
109	00FFFF66	109	TIER0	.EQU	H'FFFF66	;Timer interrupt enable ;register0
110	00FFFF6A	110	GRA0	.EQU	H'FFFF6A	;General register A0
111	000006A	111	GRA0L	.EQU	H'6A	
112	00FFFF6C	112	GRB0	.EQU	H'FFFF6C	;General register B0
113	000006C	113	GRB0L	.EQU	H'6C	
114		114	;			
115		115	;		***** TPC *****	
116		116	;			
117	00FFFA0	117	TPMR	.EQU	H'FFFA0	;TPC output mode register
118	00FFFA1	118	TPCR	.EQU	H'FFFA1	;TPC output control register
119	00FFFA2	119	NDERB	.EQU	H'FFFA2	;Next data enable register B
120	00000A4	120	NDRBL	.EQU	H'A4	
121	00FFFA4	121	NDRB	.EQU	H'FFFA4	;Next data register B
122		122	;			
123		123	;		***** PORT B *****	
124		124	;			
125	00FFFD6	125	PBDR	.EQU	H'FFFD6	;PortB data register
126	00FFFD4	126	PBDDR	.EQU	H'FFFD4	;PortB data direction ;register
127		127	;			
128		128	;		*****	
129		129	;	*	*	
130		130	;	MAIN PROGRAM : STP4MN	*	
131		131	;	*	*	
132		132	;	*****		

3. 応用編

```

133          133 ;
134 010000          134          .SECTION   PROG, CODE, LOCATE=H'010000
135          135 ;*
136          00010000 136 STP4MN      .EQU      $
137 010000 7A0700FFFE 137          MOV.L      #'FFFEFE, SP      ;Initialize stack pointer
138 010006 F801      138          MOV.B      #'01, R0L      ;Initialize mdoe counter
139 010008 3800      139          MOV.B      R0L, @CTRLMD
140          140 ;
141          141 ;***** INITIALIZE TPC *****
142          142 ;
143 01000A F8F8      143          MOV.B      #B'1111000, R0L
144 01000C 38A0      144          MOV.B      R0L, @TPMR      ;Initialize TPMR(TP15-12
;nonoverlap)
145 01000E F890      145          MOV.B      #B'10010000, R0L
146 010010 38D6      146          MOV.B      R0L, @PBDR      ;Initialize PBDR
147 010012 F8F0      147          MOV.B      #B'11110000, R0L
148 010014 38D4      148          MOV.B      R0L, @PBDDR      ;Initialize PBDDR(B4-B7
;output)
149 010016 38A2      149          MOV.B      R0L, @NDRB      ;Initialize NDRB(TP15-12
;enable)
150 010018 F83F      150          MOV.B      #B'00111111, R0L
151 01001A 38A1      151          MOV.B      R0L, @TPCR      ;Initialize TPCR(Select ITU0
;trigger)
152 01001C F090      152          MOV.B      #B'10010000, R0H
153 01001E 30A4      153          MOV.B      ROH, @NDRB      ;Initialize NDRB
154          154 ;
155          155 ;***** INITIALIZE DMAC0A *****
156          156 ;
157 010020 2827      157          MOV.B      @DTCR0A, R0L      ;Dummy read
158 010022 3027      158          MOV.B      R0H, @DTCR0A      ;Initialize DTCR0A(repeat
;mode)
159 010024 7A000001035B 159          MOV.L      #PATTBL+1, ER0      ;Initialize MAR0A
160 01002A 01006B80FF21 160          MOV.L      ER0, @MAR0A
161 010030 F0A4      161          MOV.B      #NDRBL, R0H      ;Initialize IOAR0A
162 010032 3026      162          MOV.B      R0H, @IOAR0A
163 010034 F803      163          MOV.B      #'03, R0L      ;Store first transfer count
164 010036 3824      164          MOV.B      R0L, @ETCR0AH
165 010038 F804      165          MOV.B      #04, R0L      ;Store transfer count
166 01003A 3825      166          MOV.B      R0L, @ETCR0AL
167          167 ;
168          168 ;***** INITIALIZE DMAC0B *****
169          169 ;
170 01003C 202F      170          MOV.B      @DTCR0B, R0H      ;Dummy read
171 01003E F0C8      171          MOV.B      #B'11001000, R0H      ;Initialize DTCR0B(I/O
;mode)
172 010040 302F      172          MOV.B      R0H, @DTCR0B
173 010042 7A0000010268 173          MOV.L      #UPTBL, ER0      ;Initialize MAR0B

```

```

174 010048 01006B80FF29 174          MOV.L    ER0,@MAR0B
175 01004E F86C          175          MOV.B    #GRB0L,R0L      ;Initialize IOAR0B
176 010050 382E          176          MOV.B    R0L,@IOAR0B
177 010052 7900003C      177          MOV.W    #D'60,R0      ;Store transfer count
178 010056 6B80FF2C      178          MOV.W    R0,@ETCR0B
179                                179          ;
180                                180          ;***** INITIALIZE ITU *****
181                                181          ;
182 01005A F8C3          182          MOV.B    #B'11000011,R0L
183 01005C 3864          183          MOV.B    R0L,@TCR0      ;Initialize TCR0(GRB timer
                                ;clear0)
184 01005E F0F9          184          MOV.B    @B'11111001,R0H
185 010060 3066          185          MOV.B    R0H,@TIER0      ;Initialize TIER0(Interport
                                ;enamie)
186 010062 7900000A      186          MOV.W    #D'10,R0      ;Initialize GRA0
187 010066 6B80FF6A      187          MOV.W    R0,@GRA0
188                                188          ;
189                                189          ;***** ENABLE INTERRUPT *****
190                                190          ;
191 01006A 0700          191          LDC.B    #B'00000000,CCR
192                                192          ;
193                                193          ;***** START ITU *****
194                                194          ;
195 01006C 7A0000145855  195          MOV.L    #H'145855,ER0    ;Initialize motor
196 010072 1B70          196 STP4MN1  DEC.L    #1,ER0
197 010074 46FC          197          BNE     STP4MN1
198 010076 F8E1          198          MOV.B    #B'11100001,R0L
199 010078 3860          199          MOV.B    R0L,@TSTR      ;Set TCNT0 start flag
200 01007A 40FE          200 STP4MN99  BRA     STP4MN99
201                                201          ;
202                                202          ;*****
203                                203          ;*
204                                204          ;*          NAME : STP41(MOTOR CONTROL)
205                                205          ;*
206                                206          ;*****
207                                207          ;*
208                                208          ;*          ENTRY :   NOTHING
209                                209          ;*          RETURNS :  NOTHING
210                                210          ;*
211                                211          ;*****
212                                212          ;
213                                213 STP41      .EQU     $
214 01007C 01006DF0      214          PUSH.L   ER0
215 010080 2800          215          MOV.B    @CTRLMD,R0L
216                                216          ;
217 010082 A801          217          CMP.B    #H'01,R0L      ;Branch if slue up(CW)
                                ;completed

```

3. 応用編

```

218 010084 5870002A      218          BEQ      NEXT_C_CW
219                      219      ;
220 010088 A802          220          CMP.B   #'02,R0L      ;Branch if constant(CW)
                                           ;completed
221 01008A 58700048      221          BEQ      NEXT_D_CW
222                      222      ;
223 01008E A803          223          CMP.B   #'03,R0L      ;Branch if slue down(CW)
                                           ;completed
224 010090 58700066      224          BEQ      NEXT_S_CW
225                      225      ;
226 010094 A800          226          CMP.B   #'0,R0L      ;Branch if stop completed
227 010096 58700088      227          BEQ      NEXT_U_CCW
228                      228      ;
229 01009A A881          229          CMP.B   #'81,R0L      ;Branch if slue up(CCW)
                                           ;completed
230 01009C 587000EA      230          BEQ      NEXT_C_CCW
231                      231      ;
232 0100A0 A882          232          CMP.B   #'82,R0L      ;Branch if constant(CCW)
                                           ;completed
233 0100A2 58700108      233          BEQ      NEXT_D_CCW
234                      234      ;
235 0100A6 A883          235          CMP.B   #'83,R0L      ;Branch if slue down(CCW)
                                           ;completed
236 0100A8 58700126      236          BEQ      NEXT_S_CCW
237                      237      ;
238 0100AC A880          238          CMP.B   #'00,R0L      ;Branch if stop completed
239 0100AE 58700148      239          BEQ      NEXT_U_CW
240                      240      ;
241                      241      ;***** NEXT CONSTANT CW *****
242                      242      ;
243 0100B2 F858          243 NEXT_C_CW   MOV.B   #'01011000,R0L
244 0100B4 382F          244          MOV.B   R0L,@DTCR0B      ;Store report mode data to
                                           ;DTCR0B
245 0100B6 79000BB8      245          MOV.W   #'3000,R0      ;Store transfer count to
                                           ;ETCR0B
246 0100BA 6B80FF2C      246          MOV.W   R0,0ETCR0B
247 0100BE 7A0000010358    247          MOV.L   #CNSTBL,ER0      ;Store data address to MAR0B
248 0100C4 01006B60FF29    248          MOV.L   ER0,@MAR0B
249 0100CA F802          249          MOV.B   #'02,R0L      ;Set mode flag
250 0100CC 3800          250          MOV.B   R0L,@CTRLMD
251 0100CE 7F2F7070      251          BSET   #7,@DTCR0B      ;DMAC channel 1 start
252 0100D2 5800018C      252          BRA     EXIT
253                      253      ;
254                      254      ;***** NEXT SLUE DOWN CW *****
255                      255      ;
256 0100D6 F868          256 NEXT_D_CW   MOV.B   #'01101000,R0L
                                           ;Set MAR0B decrement

```

```

257 0100D8 382F          257          MOV.B    R0L,@DTCR0B
258 0100DA 79000078      258          MOV.W    #D'120,R0          ;Store transfer count to
                                   ;ETCR0B
259 0100DE 6B80FF2C      259          MOV.W    R0,@ETCR0B
260 0100E2 7A0000010356     260          MOV.L    #DOWNTBL,ER0     ;Store data address to
                                   ;MAR0B
261 0100E8 01006B80FF29     261          MOV.L    ER0,@MAR0B
262 0100EE F803          262          MOV.B    #H'03,R0L        ;Set mode flag
263 0100F0 3800          263          MOV.B    R0L,@CTRLMD
264 0100F2 7F2F7070      264          BSET    #7,@DTCR0B        ;DMAC1 start
265 0100F6 58000168      265          BRA     EXIT
266                                266          ;
267                                267          ;***** NEXT STOP *****
268                                268          ;
269 0100FA 7F277270      269  NEXT_S_CW  BCLR    #7,@DTCR0A        ;Stop DMAC channel0
270 0100FE F858          270          MOV.B    #B'01011000,R0L
                                   ;Store repeat mode data to
                                   ;DTCR0B
271 010100 382F          271          MOV.B    R0L,@DTCR0B
272 010102 790003E8      272          MOV.W    #D'1000,R0        ;Store transfer count to
                                   ;ETCR0B
273 010106 6B80FF2C      273          MOV.W    R0,@ETCR0B
274 01010A 7A0000010358     274          MOV.L    #CNSTBL,ER0     ;Store data address to MAR0B
275 010110 01006B80FF29     275          MOV.L    ER0,@MAR0B
276 010116 F808          276          MOV.B    #H'00,R0L        ;Set mode flag
277 010118 3800          277          MOV.B    R0L,@CTRLMD
278 01011A 7F2F7070      278          BSET    #7,@DTCR0B        ;DMAC channel1 start
279 01011E 58000140      279          BRA     EXIT
280                                280          ;
281                                281          ;***** NEXT SLUE UP CCW *****
282                                282          ;
283 010122 F848          283  NEXT_U_CCW  MOV.B    #B'01001000,R0L
                                   ;Set MAR0B increment
284 010124 382F          284          MOV.B    R0L,@DTCR0B
285 010126 7900007B      285          MOV.W    #D'120,R0        ;Store Transfer count to
                                   ;ETCR0B
286 01012A 6B80FF2C      286          MOV.W    R0,@ETCR0B
287 01012E 7EA47370      287          BTST    #7,@NDRB          ;Set NDRB
288 010132 58600010      288          BNE     PTRN1_2CCW
289 010136 7EA47360      289          BTST    #6,@NDRB
290 01013A 58600020      290          BNE     PTRN3CCW
291 01013E F860          291          MOV.B    #H'60,R0L
292 010140 38A4          292          MOV.B    R0L,@NDRB
293 010142 5800001C      293          BRA     NEXTCCW
294                                294          ;
295 010146 7EA47360      295  PTRN1_2CCW  BTST    #6,@NDRB
296 01014A 58600008      296          BNE     PTRN2CCW

```

3. 応用編

```

297 01014E F830          297          MOV.B    #'30,R0L
298 010150 38A4          298          MOV.B    R0L,@NDRB
299 010152 5800000C      299          BRA      NEXTCCW
300
301 010156 F890          301 PTRN2CCW  MOV.B    #'90,R0L
302 010158 38A4          302          MOV.B    R0L,@NDRB
303 01015A 58000004      303          BRA      NEXTCCW
304
305 01015E F8C0          305 PTRN3CCW  MOV.B    #'C0,R0L
306 010160 38A4          306          MOV.B    R0L,@NDRB
307 010162 F8B0          307 NEXTCCW  MOV.B    #'B'10110000,R0L
                                           ;MAR0A decrement
308 010164 3827          308          MOV.B    R0L,@DTCR0A
309 010166 2024          309          MOV.B    @ETCR0AH,R0H
310 010168 F805          310          MOV.B    #'D'5,R0L
311 01016A 1808          311          SUB.B    R0H,R0L
312 01016C 3824          312          MOV.B    R0L,@ETCR0AH
313 01016E 7A00001026B   313          MOV.L    #UPTBL,ER0      ;Set data address
314 010174 01006B80FF29  314          MOV.L    ER0,@MAR0B
315 01017A F881          315          MOV.B    #'H'81,R0L     ;Set mode flag
316 01017C 3800          316          MOV.B    R0L,@CTRLMD
317 01017E 7F2F7070     317          BSET    #7,@DTCR0B     ;DMAC channel1 start
318 010182 7F277070     318          BSET    #7,@DTCR0A     ;DTCR0A start
319 010186 580000D8      319          BRA      EXIT
320
321
322
323 01018A F858          323 NEXT_C_CCW  MOV.B    #'01011000,R0L
                                           ;Store repeat mode data to
                                           ;DTCR0B
324 01018C 382F          324          MOV.B    R0L,@DTCR0B
325 01018E 79000BB8      325          MOV.W    #'D'3000,R0    ;Store transfer count to
                                           ;ETCR0B
326 010192 6B80FF2C      326          MOV.W    R0,@ETCR0B
327 010196 7A000010358   327          MOV.L    #CNSTBL,ER0   ;Store data address to MAR0B
328 01019C 01006B80FF29  328          MOV.L    ER0,@MAR0B
329 0101A2 F882          329          MOV.B    #'H'82,R0L     ;Set mode flag
330 0101A4 3800          330          MOV.B    R0L,@CTRLMD
331 0101A6 7F2F7070     331          BSET    #7,@DTCR0B     ;DMAC channel1 start
332 0101AA 580000B4      332          BRA      EXIT
333
334
335
336 0101AE F868          336 NEXT_D_CCW  MOV.B    #'B'01101000,R0L
                                           ;Set MAR0B decrement
337 0101B0 382F          337          MOV.B    R0L,@DTCR0B

```



```

338 0101B2 79000078      338          MOV.W      #D'120,R0      ;Store transfer count to
                                           ;ETCR0B
339 0101B6 6B80FF2C      339          MOV.W      R0,@ETCR0B
340 0101BA 7A000010356    340          MOV.L      #DOWNTBL,ER0    ;Store data address to MAR0B
341 0101C0 01006B80FF29    341          MOV.L      ER0,@MAR0B
342 0101C6 F883                342          MOV.B      ##83,R0L      ;Set mode flag
343 0101C8 3800                343          MOV.B      R0L,@CTRLMD
344 0101CA 7F2F7070      344          BSET      #7,@DTCR0B    ;DMAC channel1 start
345 0101CE 58000090      345  BRA      EXIT
346                                346  ;
347                                347  ;***** NEXT STOP *****
348                                348  ;
349 0101D2 7F277270      349  NEXT_S_CCW  BCLR      #7,@DTCR0A    ;Stop DMAC channel0
350 0101D6 F858                350          MOV.B      #B'01011000,R0L
                                           ;Store repeat mode data to
                                           ;DTCR0B
351 0101D8 382F                351          MOV.B      R0L,@DTCR0B
352 0101DA 790003E8      352          MOV.W      #D'1000,R0      ;Store transfer count to
                                           ;ETCR0B
353 0101DE 6B80FF2C      353          MOV.W      R0,@ETCR0B
354 0101E2 7A000010358    354          MOV.L      #CNSTBL,ER0    ;Store data address to MAR0B
355 0101E8 01006B80FF29    355          MOV.L      ER0,@MAR0B
356 0101EE F880                356          MOV.B      ##'80,R0L      ;Set mode flag
357 0101F0 3800                357          MOV.B      R0L,@CTRLMD
358 0101F2 7F2F7070      358          BSET      #7,@DTCR0B    ;DMAC channel1 start
359 0101F6 58000058      359          BRA      EXIT
360                                360  ;
361                                361  ;***** NEXT SLUE UP CW *****
362                                362  ;
363 0101FA F848                363  NEXT_U_CW  MOV.B      #B01001000,R0L
                                           ;Set MAR0B increment
364 0101FC 382F                364          MOV.B      R0L,@DTCR0B
365 0101FE 79000078      365          MOV.W      #D'120,R0      ;Store transfer count to
                                           ;ETCR0B
366 010202 6B80FF2C      366          MOV.W      R0,@ETCR0B
367 010206 7EA47370      367          BTST      #7,@NDRB      ;Set NDRB
368 01020A 58600010      368          BNE      PTRN1_2CW
369 01020E 7EA47360      369          BTST      #6,@NDRB
370 010212 58600020      370          BNE      PTRN3CW
371 010216 F890                371          MOV.B      ##'90,R0L
372 010218 38A4                372          MOV.B      R0L,@NDRB
373 01021A 5800001C      373          BRA      NEXTCW
374                                374  ;
375 01021E 7EA47360      375  PTRN1_2CW  BTST      #6,@NDRB
376 010222 58600008      376          BNE      PTRN2CW
377 010226 F8C0                377          MOV.B      ##'C0,R0L
378 010228 38A4                378          MOV.B      R0L,@NDRB

```

3. 応用編

```

379 01022A 5800000C          379          BRA          NEXTCW
380                                380 ;
381 01022E F860              381 PTRN2CW  MOV.B      #'60,R0L
382 010230 38A4              382          MOV.B      R0L,@NDRB
383 010232 58000004          383          BRA          NEXTCW
384                                384 ;
385 010236 F830              385 PTRN3CW  MOV.B      #'30,R0L
386 010238 38A4              386          MOV.B      R0L,@NDRB
387 01023A F890              387 NEXTCW  MOV.B      #'10010000,R0L
                                                ;MAR0 increment
388 01023C 3827              388          MOV.B      R0L,@DTCR0A
389 01023E 2024              389          MOV.B      @ETCR0AH,R0H
390 010240 F805              390          MOV.B      #'D'5,R0L
391 010242 1808              391          SUB.B      R0H,R0L
392 010244 3824              392          MOV.B      R0L,@ETCR0AH
393 010246 7A0000010268      393          MOV.L      #UPTBL,ERO ;Store data address to MAR0B
394 01024C 01006B80FF29      394          MOV.L      ERO,@MAR0B
395 010252 F801              395          MOV.B      #'H'01,R0L ;Set mode flag
396 010254 3800              396          MOV.B      R0L,@CTRLMD
397 010256 7F2F7070          397          BSET      #7,@DTCR0B ;DMAC channel1 start
398 01025A 7F277070          398          BSET      #7,@DTCR0A ;DMAC channel0 start
399 01025E 58000000          399          BRA          EXIT
400                                400 ;
401 010262 01006D70          401 EXIT     POP.L      ERO
402 010266 5670              402          RTE
403                                403 ;
404                                404 ;*****
405                                405 ;* *
406                                406 ;* DATA TABLE (TIMER PERIOD) *
407                                407 ;* *
408                                408 ;*****
409                                409 ;
410                                410 ;***** SLUE UP & DOWN DATA *****
411                                411 ;
412                                412 UPTBL     .EQU      $
413 010268 F424B1646EA361D2    413          .DATA.W   D'62500,D'45412,D'28323,D'25042,D'21761,D'20058,
414 010270 55014E5A47B34371    414 +          D'18355,D'17265,D'16175,D'15401,D'14626,D'14039
         010278 3F2F3C29392236D7
415 010280 348C32BB30EA2F6E    415          .DATA.W   D'13452,D'12987,D'12522,D'12142,D'11762,D'11444,
416 010288 2DF22CB42B752A66    416 +          D'11125,D'10854,D'10582,D'10347,D'10111,D'9905
         010290 2956286B277F26B1
417 010298 25E226BB247423D1    417          .DATA.W   D'9698,D'9915,D'9332,D'9169,D'9005,D'8857,
418 0102A0 232D22992205217F    418 +          D'8709,D'8575,D'8441,D'8319,D'8196,D'8084
         0102A8 20F9207F20041F94
419 0102B0 1F241EBD1E551DF5    419          .DATA.W   D'7972,D'7869,D'7765,D'7669,D'7573,D'7484,
420 0120B8 1D951D3C1CE31C90    420 +          D'7395,D'7312,D'7229,D'7151,D'7073,D'7000
         0102C0 1C3D1BEF1BA11B58

```

```

421 0102C8 1B0F1ACB1A861A46 421 .DATA.W D'6927,D'6859,D'6790,D'6726,D'6661,D'6600,
422 0102D0 1A0519C8198B1951 422 + D'6539,D'6481,D'6423,D'6368,D'6313,D'6261
    0102D8 191718E018A91875
423 0102E0 1841181017DE17AF 423 .DATA.W D'6209,D'6160,D'6110,D'6063,D'6015,D'5970,
424 0102E8 177F1752172416F9 424 + D'5924,D'5881,D'5838,D'5797,D'5755,D'5716
    0102F0 16CE16A5167B1654
425 0102F8 162C160615E015BB 425 .DATA.W D'5676,D'5638,D'5600,D'5563,D'5526,D'5491,
426 010300 159615731550152E 426 + D'5456,D'5422,D'5388,D'5356,D'5323,D'5292
    010308 150C14EC14CB14AC
427 010310 148C146E144F1432 427 .DATA.W D'5260,D'5230,D'5199,D'5170,D'5140,D'5112,
428 010318 141413F813DC13C1 428 + D'5084,D'5057,D'5029,D'5002,D'4975,D'4950
    010320 13A513BA136F1356
429 010328 133C1323130A12F2 429 .DATA.W D'4924,D'4899,D'4874,D'4850,D'4825,D'4802,
430 010330 12D912C212AA1294 430 + D'4778,D'4756,D'4733,D'4711,D'4688,D'4667
    010338 127D12671250123B
431 010340 1225121011FB11E7 431 .DATA.W D'4645,D'4624,D'4603,D'4583,D'4562,D'4542,
432 010348 110211BE11AA1197 432 + D'4522,D'4503,D'4483,D'4465,D'4446
    010350 118311711115E
433 ;
434 00010356 434 DOWNTBL .EQU $
435 010356 115E 435 .DATA.W D'4446
436 ;
437 ;***** CONSTANT *****
438 ;
439 00010358 439 CNSTBL .EQU $
440 010358 115E 440 .DATA.W D'4446
441 ;
442 ;*****
443 ;* *
444 ;* DATA TABLE (OUTPUT PATTERN) *
445 ;* *
446 ;*****
447 ;
448 0001035A 448 PATTBL .EQU $
449 01035A 603090C0 449 .DATA.B H'60,H'30,H'90,H'C0
450 ;
451 .END

****TOTAL ERRORS 0
****TOTAL WARNINGS 0

```

3.5 5相ステッピングモータ応用例

3.5.1 仕様

- 図 3.33 に示すように H8/3003 の内蔵機能のうち、ITU、TPC、DMAC を使用し、5 相のステッピングモータを制御します。
- ステッピングモータは 2 相励磁方式で制御します。
- 本タスクでは、ステッピングモータを停止 正転 停止 逆転 停止の動作を繰り返します。
- 本タスクではソフトウェアの介在なしに、スルーアップおよびスルーダウンの処理を行います。
- ドライバの保護のため貫通電流防止期間を設けます。

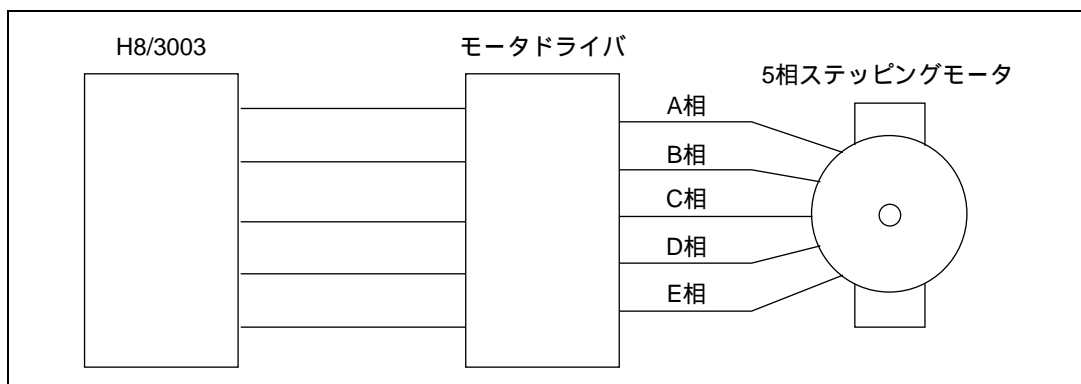


図 3.33 5相ステッピングモータ制御回路図

3.5.2 考え方

- ステッピングモータ動作例

図3.34に2相励磁で5相ステッピングモータを動作させる例を示します。動作概要は以下のとおりです。

 - 図 3.34 に示すようにパルスが “ H ” になっているときに、対応する相を励磁します。
 - では、E 相と A 相を同時に励磁します。このときロータは、E 相と A 相の間に位置します。
 - では、A 相と B 相を同時に励磁します。このときロータは、A 相と B 相の間に位置します。以下、2 相励磁方式は、隣り合う 2 つの相 (E、A 相、A、B 相、B、C 相、C、D 相、D、E 相) を励磁し、ロータを回転させます。
 - 逆転動作の場合は、E、D 相 D、C 相 C、B 相 B、A 相 A、E 相を順番に励磁することでステッピングモータを逆転させます。
 - 停止動作は、正転動作の最後の相または、逆転動作の最後の相を一定時間励磁し続けることでステッピングモータを停止させます。

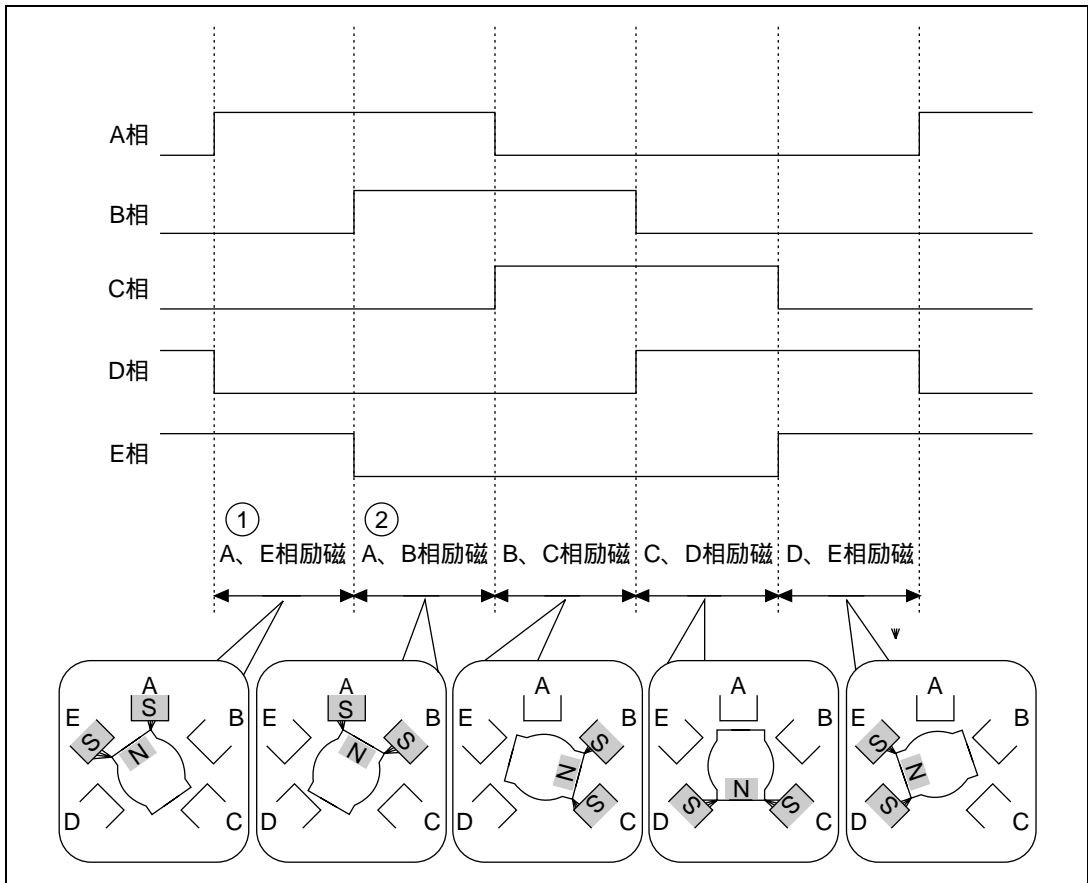


図 3.34 ステッピングモータ動作例

- ノンオーバーラップ時間

図3.35に示すように出力パターン切り換え時に貫通電流防止期間 n （ノンオーバーラップ時間）を挿入します。励磁相の切り換え時に生じるターンオフの遅れにより、ドライバが破壊する危険性があり、これを防ぐためにノンオーバーラップ時間を挿入し、時間遅れをもたせて対策します。

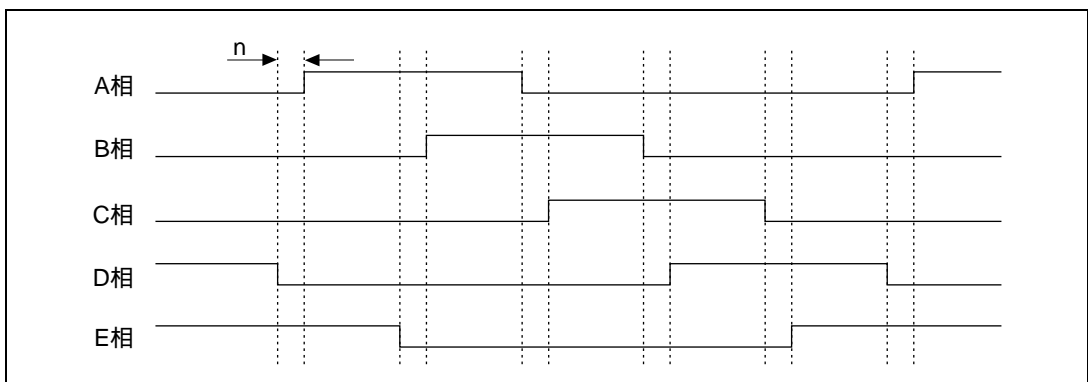


図 3.35 ノンオーバーラップ時間出力例

• スルーアップ、スルーダウン動作

図3.36に示すような加減速制御されたパルスを出力します。スルーアップ、スルーダウン動作を行うことにより、モータの脱調を防止します。

モータを動作させる際に、急に周期の短いパルスを出力すると、モータは負荷に追いつけず、回転しないことがあります。これを防止する対策としてスルーアップおよびスルーダウンを行います。

動作原理を以下に示します。

- (1) 徐々にパルス周期を短くし、設定した量のパルスを出力する。(スルーアップ)
- (2) 一定のパルス周期で設定した量のパルスを出力する。(定速)
- (3) 徐々にパルス周期を長くし、設定した量のパルスを出力する。(スルーダウン)

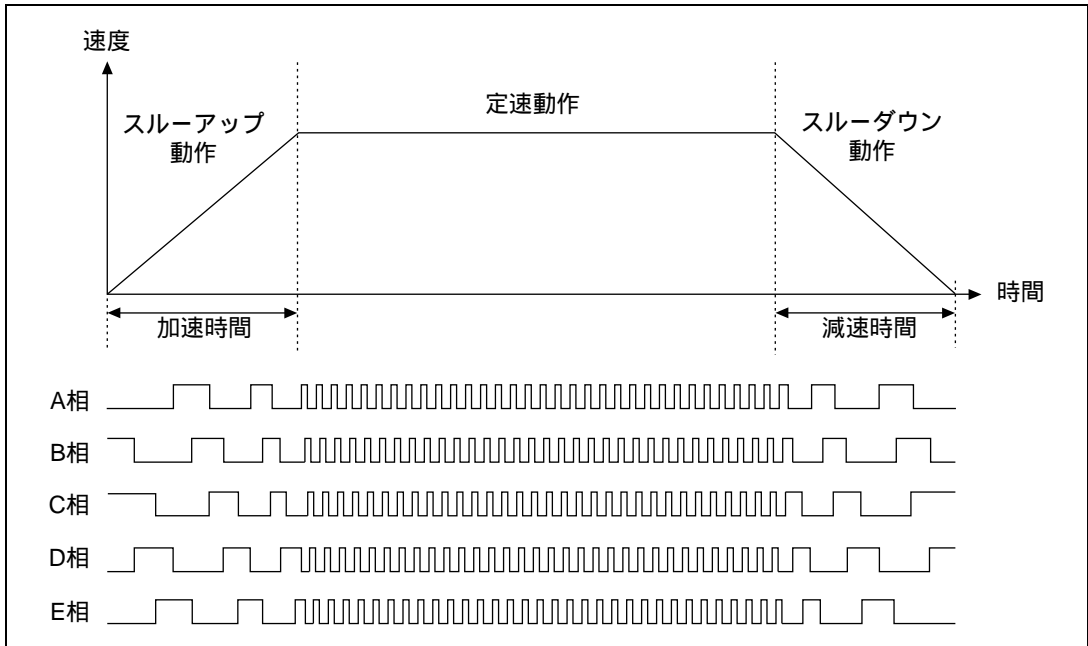


図 3.36 スルーアップ、スルーダウン動作

3.5.3 使用機能説明

図 3.37 に本タスクで使用する H8/3003 の内蔵機能を示します。

図 3.37 に示すように DMAC を 2 チャンネル使用し、5 相パルス出力パターンを TPC し、5 相のパルスを出力します。

また、パルス周期を ITU の GRB に転送し、5 相パルスの出力タイミングを制御します。

- 周期データテーブル：ステップ周期を変更するためのテーブル (ITU のタイマ値) を ROM 上に設定します。
- 出力パターンテーブル：5 相パルスを出力するための出力パターンを ROM 上に設定します。
- DMAC0A：ITU のコンペアマッチ A で起動します。
出力パターンデータテーブルから出力パターンを TPC の NDR に転送します。
- DMAC0B：ITU のコンペアマッチ A で起動します。
周期データテーブルからパルスの周期データを ITU の GRB に転送します。
- ITU
 - コンペアマッチ A：DMAC0A、B および TPC を起動します。
 - コンペアマッチ B：タイマカウンタのクリアを行います。また、TPC を起動します。

- TPC
 - DMAC を使用することにより CPU の介在なくデータを出力します。
 - 最大 16 ビットデータを出力することが可能ですが、本タスクでは 5 ビットのデータを出力します。
 - ノンオーバーラップ期間をもったパルスを出力することが可能です。
- コンペアマッチ B：波形が“H”から“L”に変化するパルスの出力を行います。
“L”から“H”に変化する出力は保留します。
- コンペアマッチ A：コンペアマッチ B で保留された“L”から“H”の出力を行います。
(GRAの設定値分遅延します)

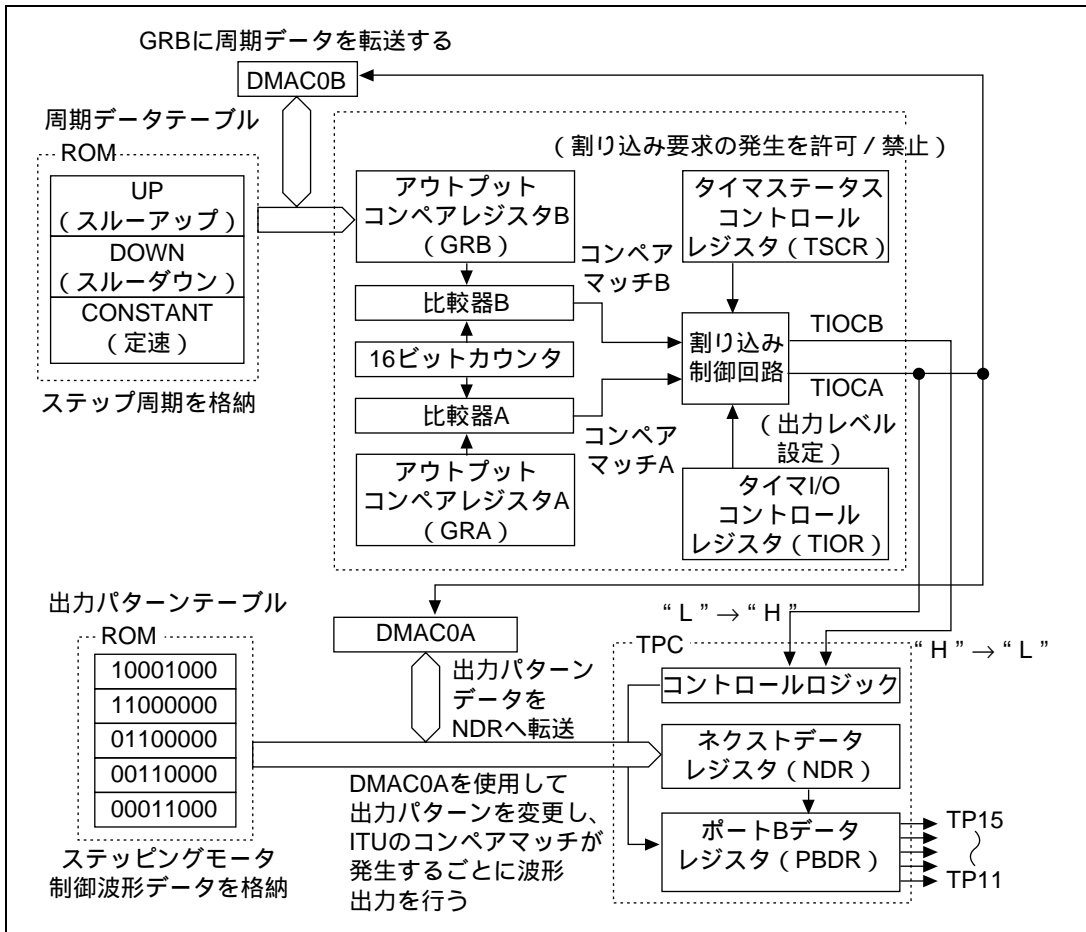


図 3.37 5相ステップングモータ制御ブロック図

3. 応用編

表 3.24 に本タスク例の機能割り付けを示します。表 3.24 に示すように H8/3003 の機能を割り付け、2 相励磁のステッピングモータ出力波形を生成します。

表 3.24 H8/3003 機能割り付け

H8/3003 の機能		機能
ITU (ch0)	TCNT0	16 ビットカウンタ
	GRA0	アウトプットコンペアレジスタ A
	GRB0	アウトプットコンペアレジスタ B
	TCR0	カウントクロックの選択およびカウンタクリア要因の選定
	TIOR0	GRA および GRB をアウトプットコンペアレジスタに設定
	TIER0	割り込み要求の許可 / 禁止を制御
	TSR0	コンペアマッチやオーバフローのステータスレジスタ
	TSTR	TCNT のカウント動作の許可 / 禁止を設定
DMAC0A、B	DTCR0A、B	DMAC の各チャンネルの動作を制御
	MAR0A、B	転送元アドレスを設定
	IORA0A、B	転送先アドレスを設定
	ETCR0A、B	転送回数を設定
TPC	PBDDR	TPC 出力端子を設定
	PBDR	TPC 出力 B の出力データを格納
	TPMR	TPC 出力のノンオーバーラップモードを設定
	TPCR	TPC 出力の出力トリガ信号を設定
	NDERB	TPC 出力 B の許可 / 禁止を設定
	NDRB	TPC 出力 B の次に出力するデータを格納
	TP15 ~ TP11	5 相のステッピングモータ出力波形を発生

3.5.4 動作説明

5相パルス出力例（ノンオーバーラップあり）

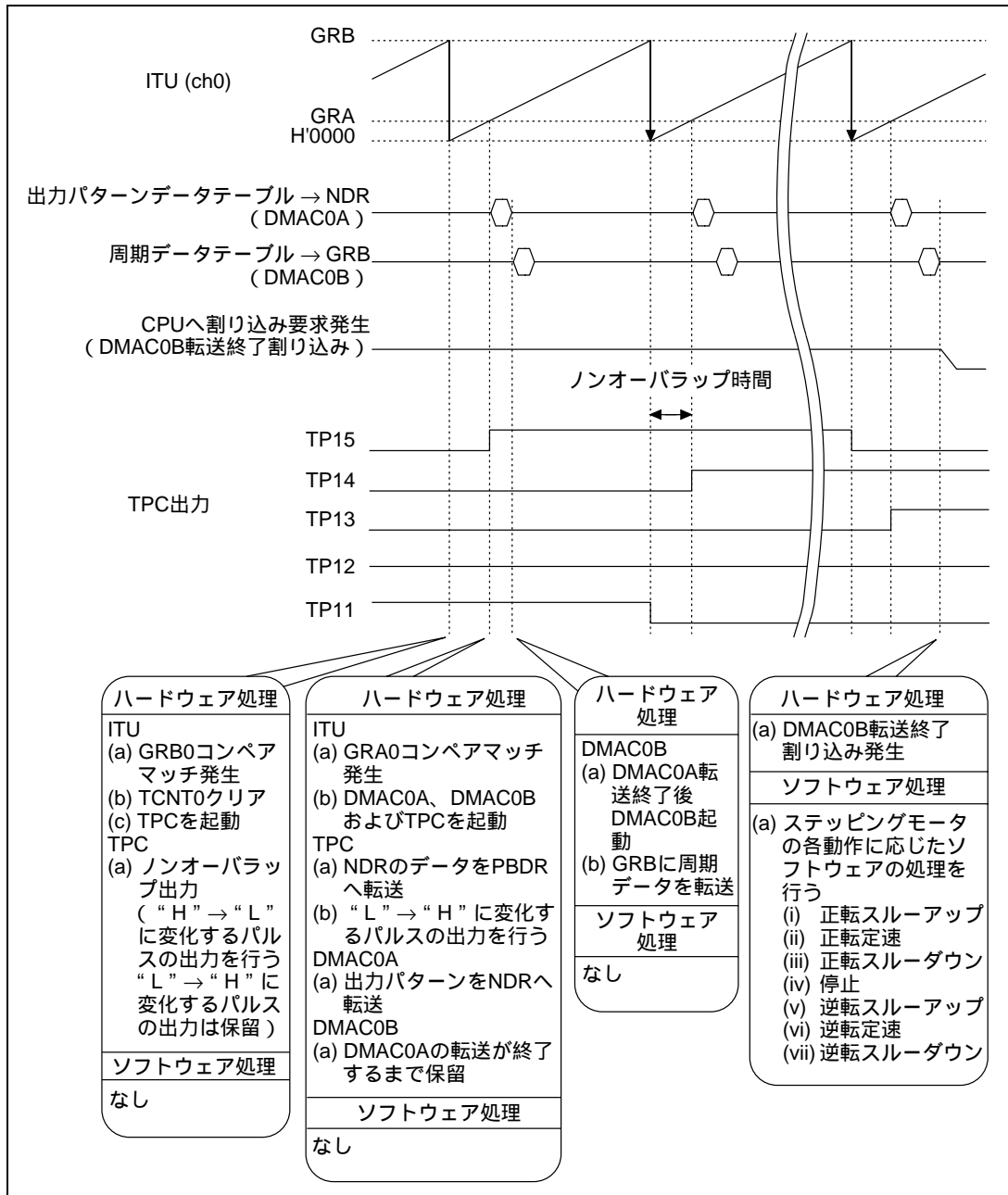


図 3.38 ステッピングモータ・タイミング説明

3.5.5 ソフトウェア説明

表 3.25 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	ST5MIN	ITU、TPC および DMAC0A、B の初期設定を行い、正転のスルーアップ動作を行う。
モータ制御	DENDINT	モータの各動作（正転定速、停止など）を判定し、制御を行う。 <ul style="list-style-type: none"> DMAC0A、B の設定を行い、正転のスルーアップ動作を行う。 DMAC0B の設定を行い、正転の定速動作を行う。 DMAC0A、B の設定を行い、正転のスルーダウン動作を行う。 DMAC0A、B の設定を行い、停止動作を行う。 DMAC0A、B の設定を行い、逆転のスルーアップ動作を行う。 DMAC0B の設定を行い、逆転の定速動作を行う。 DMAC0A、B の設定を行い、逆転のスルーダウン動作を行う。 DMAC0A、B の設定を行い、停止動作を行う。

表 3.26 使用内部レジスタの説明

内蔵機能	レジスタ名	機能
ITU	GRA0	ノンオーバーラップ時間を設定
	GRB0	タイマ周期を設定
	TIER0	IMFA 割り込みを許可
	TCR0	ITU を以下のように設定する <ul style="list-style-type: none"> GRB のコンペアマッチでカウンタクリア 内部クロック でカウント
	TIOR0	GRA および GRB をアウトプットコンペアレジスタに設定し、端子出力を禁止
	TSTR	TCNT0 のカウント動作を許可する
DMAC0A	DTCR0A	DMAC0A を以下のように設定する <ul style="list-style-type: none"> データサイズをバイトサイズに設定 MAR をインクリメントに設定 データ転送をリピートモードに設定 転送の許可および禁止を選択（モジュールによって異なる） 起動要因を ITUch0 のコンペアマッチ A に設定
	MAR0A	出力パターンデータテーブルの転送元アドレスを設定
	IOAR0A	NDRB のアドレス（転送先）を設定
	ETCR0A	転送回数を設定
DMAC0B	DTCR0B	DMAC0B を以下のように設定する <ul style="list-style-type: none"> データサイズをワードサイズに設定 MAR をインクリメントまたはデクリメントに設定（処理内容によって異なる） データ転送を I/O モードに設定（処理内容によってアイドルモードに設定） 転送を許可する 割り込み要求（DEND）を許可する 起動要因を ITUch0 のコンペアマッチ A に設定
	MAR0B	周期データテーブルの転送元アドレスを設定
	IOAR0B	GRB のアドレス（転送先）を設定
	ETCR0B	転送回数を設定

内蔵機能	レジスタ名	機能
TPC	PBDDR	TP15～TP11までのTPC出力を許可
	PBDR	出力パターンデータを格納
	TPMR	TP15～TP11をノンオーバーラップ出力に設定
	TPCR	TP15～TP11の出力トリガをITUch0のコンペアマッチに設定
	NDERB	TPC出力TP15～TP11を許可する
	NDRB	次の出力パターンデータを格納する

- 引数の説明

本タスクでは引数のRAMは使用していません。

表 3.27 使用 RAM 説明

ラベル名	機能	データ長	使用モジュール名
PDAT_F	5相ステッピングモータの動作状態を示すフラグです。 H'01：正転スルーアップ動作が終了したことを示します。 H'02：正転定速動作が終了したことを示します。 H'04：正転スルーダウン動作が終了したことを示します。 H'08：正転停止動作が終了したことを示します。 H'10：逆転スルーアップ動作が終了したことを示します。 H'20：逆転定速動作が終了したことを示します。 H'40：逆転スルーダウン動作が終了したことを示します。 H'80：逆転停止動作が終了したことを示します。	1バイト	モータ制御

表 3.28 データテーブル説明

テーブル名	機能	データ長	データ容量
TBL	5相パルスを出力するためのパターンを設定します。	バイト	5バイト
UP、DOWN、CONSTANT	ステップ周期を変更するためのデータを設定します。		2540バイト

3. 応用編

- 使用汎用レジスタ

	31	ERn	RnH	RnL	0
0	全ルーチンにてデータ設定時のワークとして使用				
1					
2					
3					
4					
5					
6					
7					

図 3.39 使用汎用レジスタ

3.5.6 フローチャート

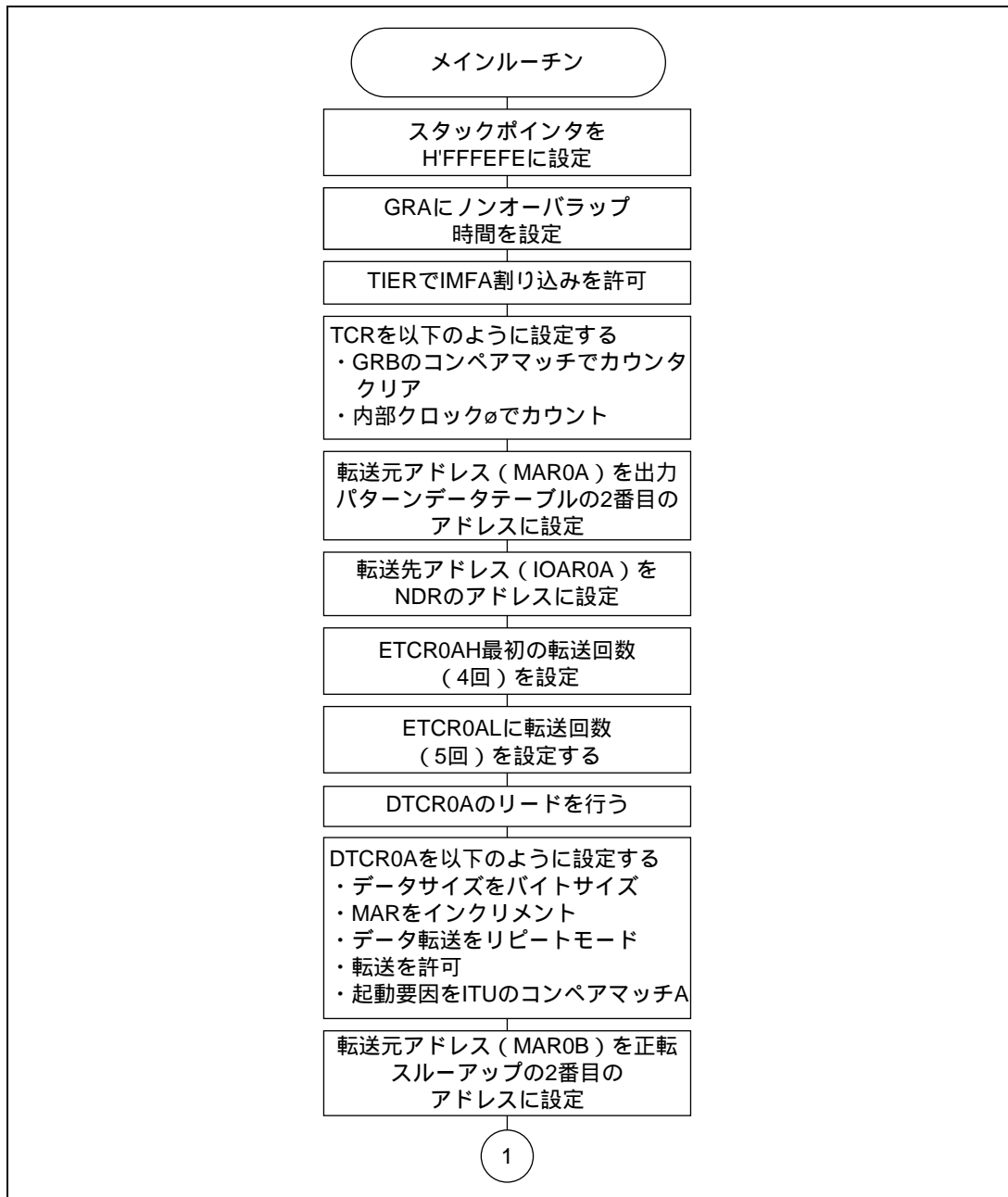


図 3.40 メインルーチン (1)

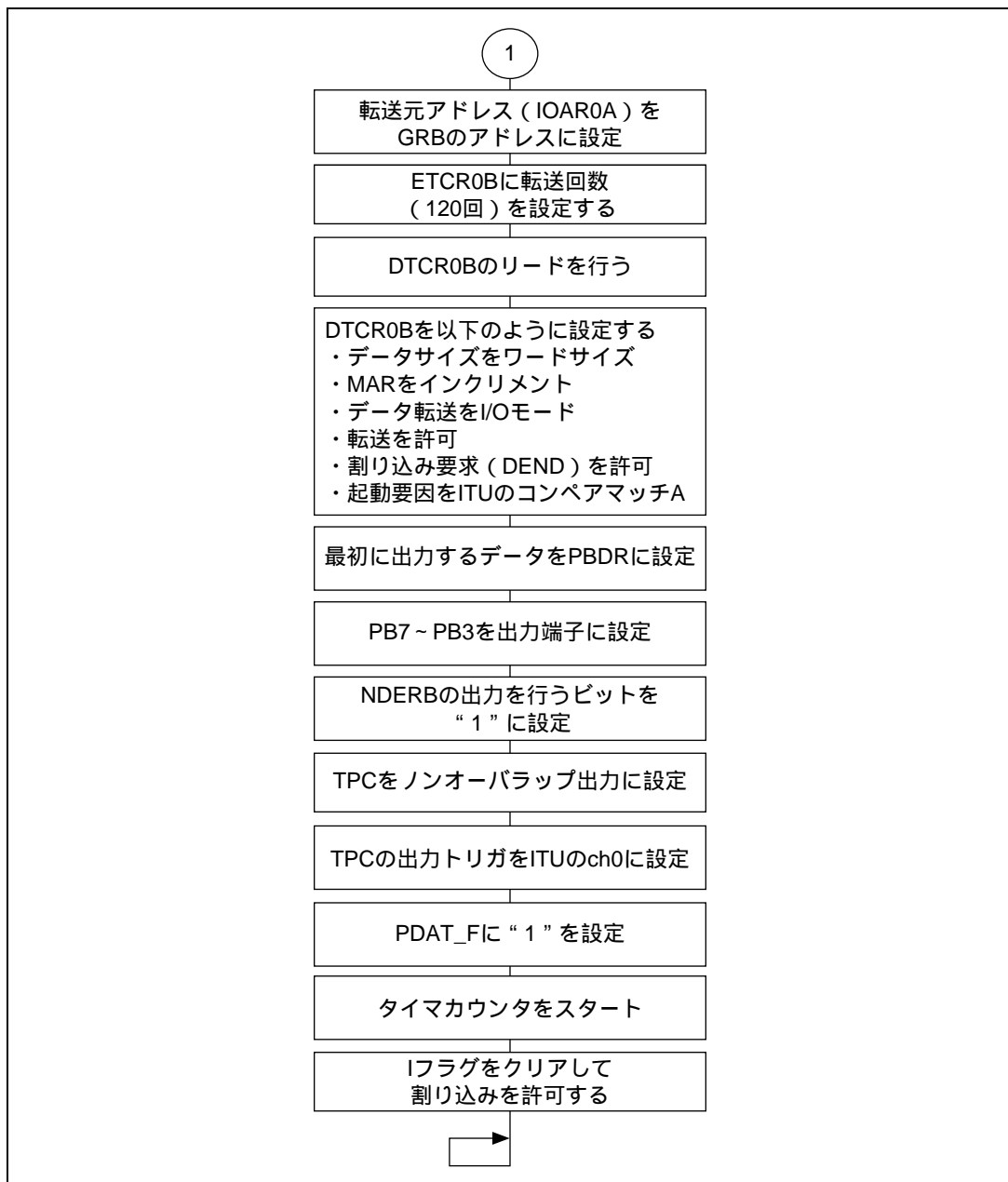


図 3.40 メインルーチン (2)

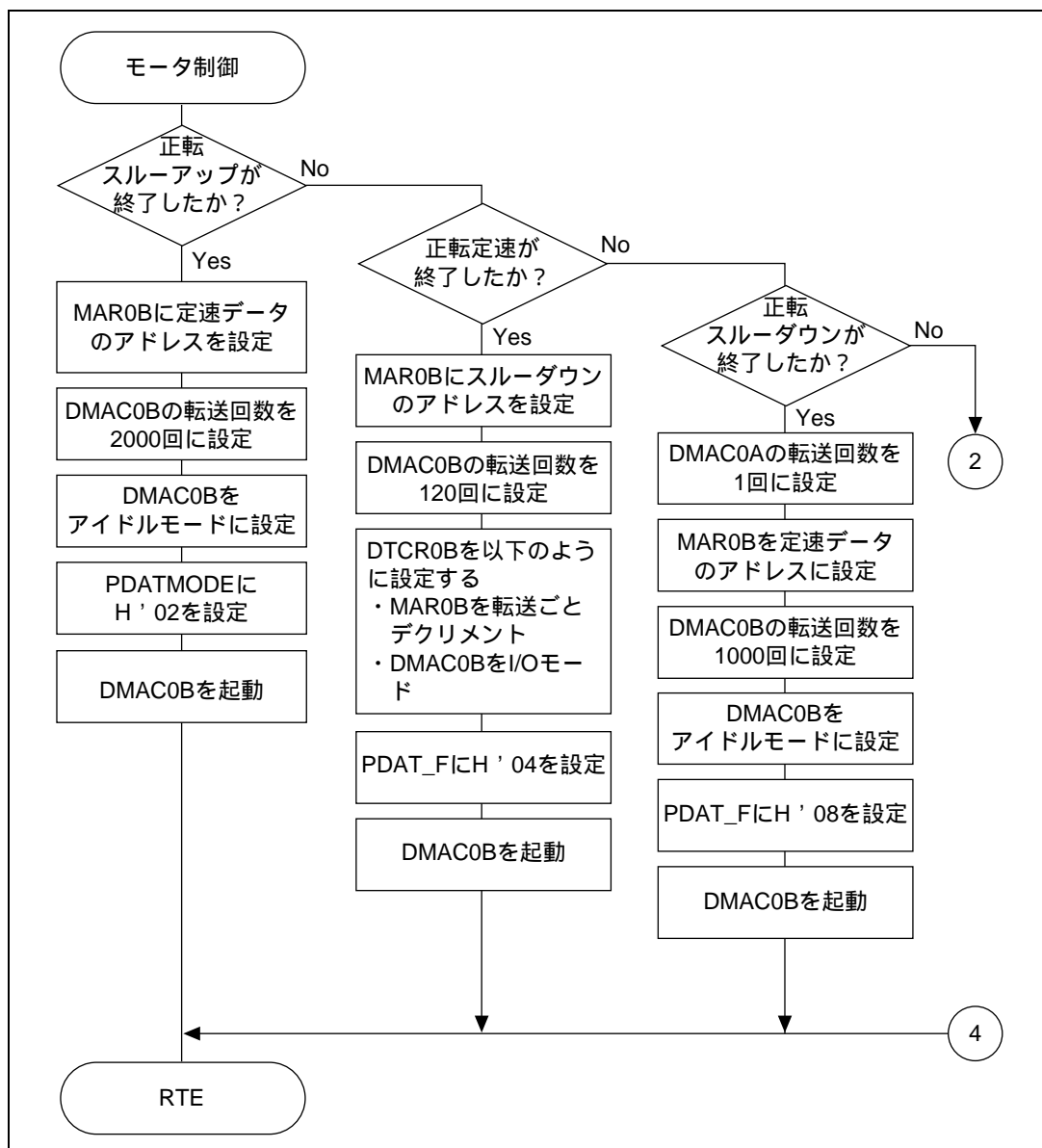


図 3.41 モータ制御 (1)

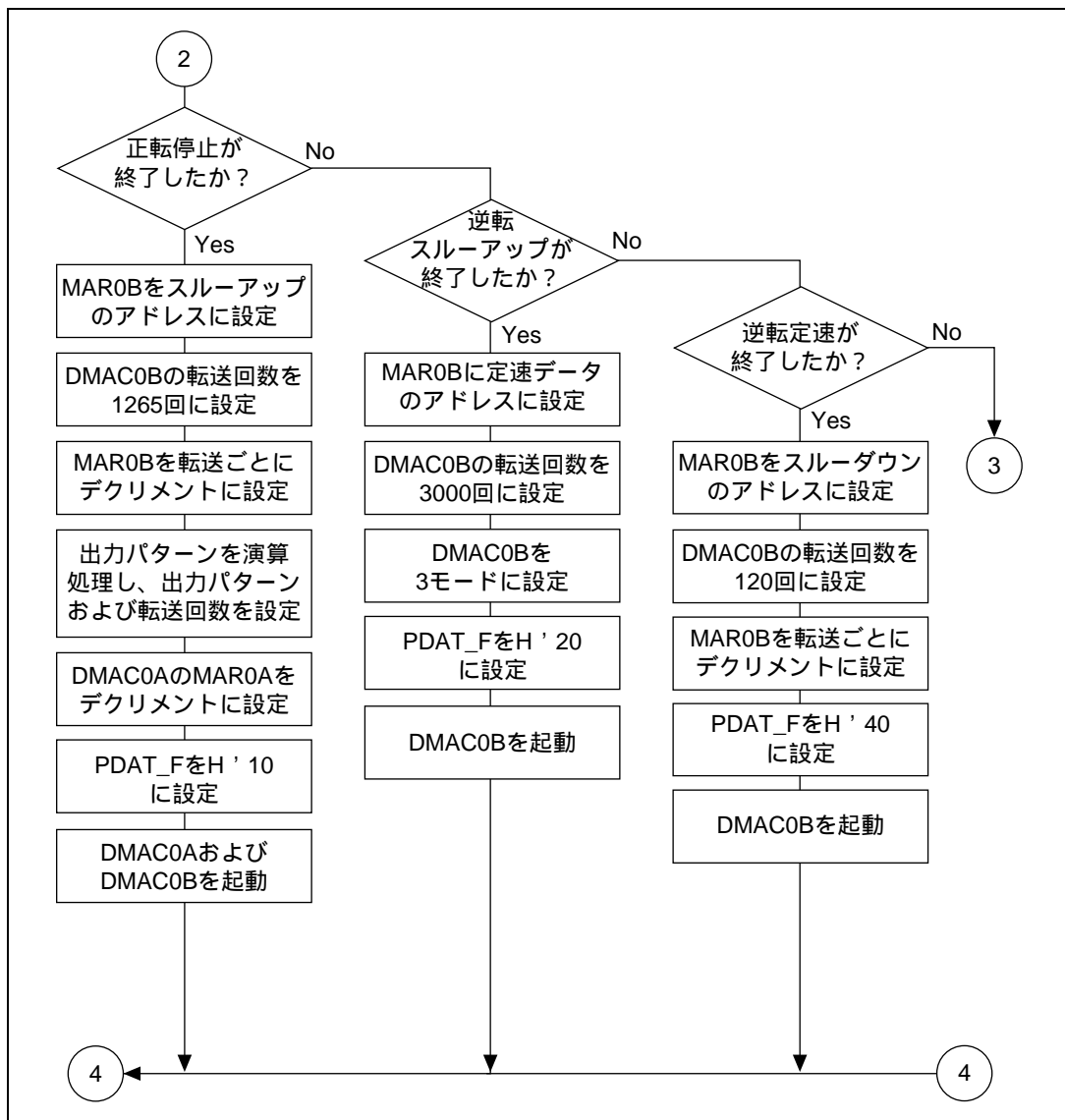


図 3.41 モータ制御 (2)

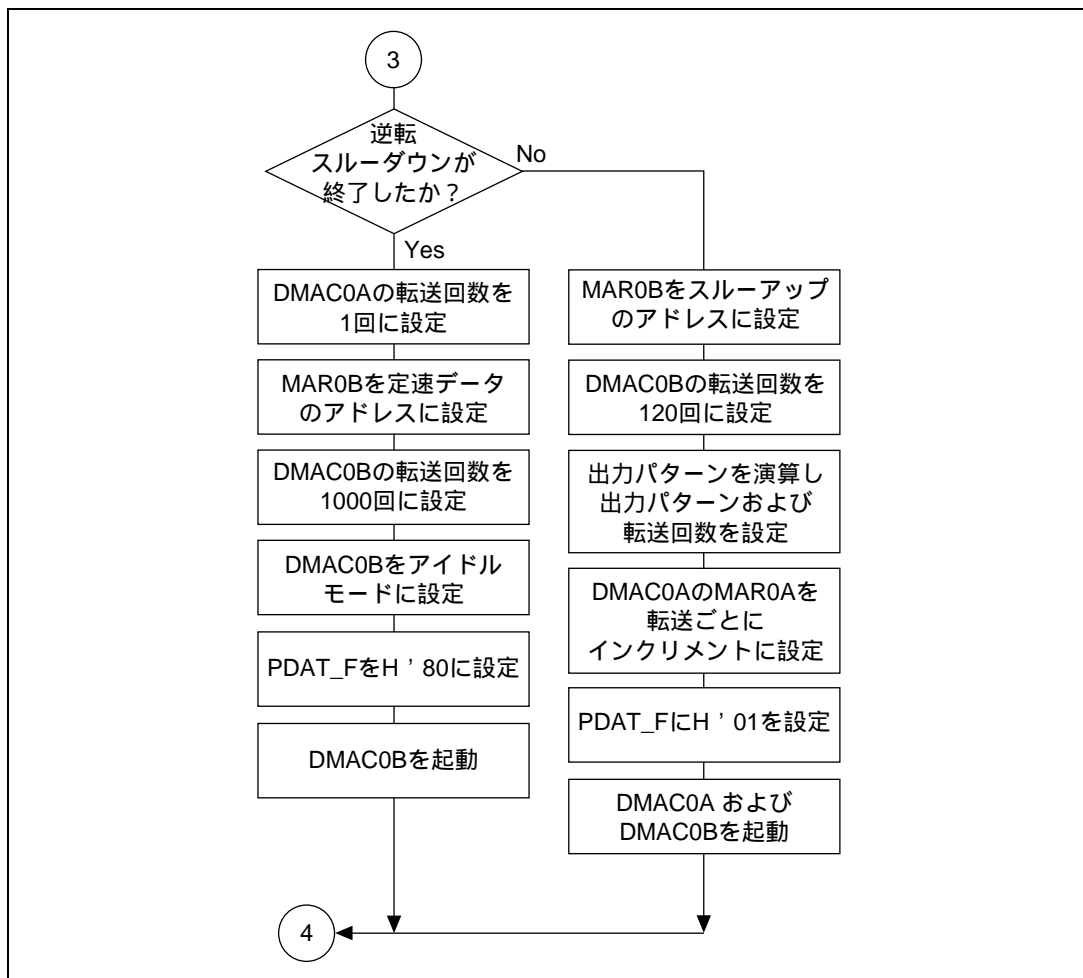


図 3.41 モータ制御 (3)

3.5.7 プログラムリスト

```

1          1  ;*****
2          2  ;*
3          3  ;*          VECTOR ADDRESS
4          4  ;*
5          5  ;*****
6          6
7          7          .CPU          300HA
8  000000    8          .SECTION    VECT, CODE, LOCATE=H' 000000
9          9  ;
10  000000    00010000    10 RES          .DATA.L    STP5MN
11         11  ;
12  00001C    12          .ORG          H'00001C
13  00001C    00010000    13 NMI          .DATA.L    STP5MN
14         14  ;
15  000030    15          .ORG          H'000030
16  000030    00010000    16 IRQ0         .DATA.L    STP5MN
17  000034    00010000    17 IRQ1         .DATA.L    STP5MN
18  000038    00010000    18 IRQ2         .DATA.L    STP5MN
19  00003C    00010000    19 IRQ3         .DATA.L    STP5MN
20  000040    00010000    20 IRQ4         .DATA.L    STP5MN
21  000044    00010000    21 IRQ5         .DATA.L    STP5MN
22  000048    00010000    22 IRQ6         .DATA.L    STP5MN
23  00004C    00010000    23 IRQ7         .DATA.L    STP5MN
24  000050    00010000    24 WOVI0        .DATA.L    STP5MN
25  000054    00010000    25 CMI          .DATA.L    STP5MN
26         26  ;
27  000060    27          .ORG          H'000060
28  000060    00010000    28 IMIA0        .DATA.L    STP5MN
29  000064    00010000    29 IMIB0        .DATA.L    STP5MN
30  000068    00010000    30 OVI0         .DATA.L    STP5MN
31         31  ;
32  000070    32          .ORG          H'000070
33  000070    00010000    33 IMIA1        .DATA.L    STP5MN
34  000074    00010000    34 IMIB1        .DATA.L    STP5MN
35  000078    00010000    35 OVI1         .DATA.L    STP5MN
36         36  ;
37  000080    37          .ORG          H'000080
38  000080    00010000    38 IMIA2        .DATA.L    STP5MN
39  000084    00010000    39 IMIB2        .DATA.L    STP5MN
40  000088    00010000    40 OVI2         .DATA.L    STP5MN
41         41  ;
42  000090    42          .ORG          H'000090
43  000090    00010000    43 IMIA3        .DATA.L    STP5MN
44  000094    00010000    44 IMIB3        .DATA.L    STP5MN
45  000098    00010000    45 OVI3         .DATA.L    STP5MN
46         46  ;

```

```

47 0000A0          47          .ORG      H'0000A0
48 0000A0 00010000 48 IMIA4      .DATA.L   STP5MN
49 0000A4 00010000 49 IMIB4      .DATA.L   STP5MN
50 0000A8 00010000 50 OVI4       .DATA.L   STP5MN
51                51 ;
52 0000B0          52          .ORG      H'0000B0
53 0000B0 00010000 53 DEND0A     .DATA.L   STP5MN
54 0000B4 00010080 54 DEND0B     .DATA.L   DENDINT
55 0000B8 00010000 55 DEND1A     .DATA.L   STP5MN
56 0000BC 00010000 56 DEND1B     .DATA.L   STP5MN
57 0000C0 00010000 57 DEND2A     .DATA.L   STP5MN
58 0000C4 00010000 58 DEND2B     .DATA.L   STP5MN
59 0000C8 00010000 59 DEND3A     .DATA.L   STP5MN
60 0000CC 00010000 60 DEND3B     .DATA.L   STP5MN
61 0000D0 00010000 61 ERI0       .DATA.L   STP5MN
62 0000D4 00010000 62 RXI0       .DATA.L   STP5MN
63 0000D8 00010000 63 TXI0       .DATA.L   STP5MN
64 0000DC 00010000 64 TEI0       .DATA.L   STP5MN
65 0000E0 00010000 65 ERI1       .DATA.L   STP5MN
66 0000E4 00010000 66 RXI1       .DATA.L   STP5MN
67 0000E8 00010000 67 TXI1       .DATA.L   STP5MN
68 0000EC 00010000 68 TEI1       .DATA.L   STP5MN
69 0000F0 00010000 69 ADI        .DATA.L   STP5MN
70                70 ;
71                71 ;*****
72                72 ;*                               *
73                73 ;*          RAM ALLOCATION          *
74                74 ;*                               *
75                75 ;*****
76                76 ;
77 FFFF00          77          .SECTION  RAM,DATA,LOCATE=H'FFFF00
78                78 ;
79 FFFF00 00000001 79 PDAT_F     .RES.B    1          ;Mod data
80                80 ;
81                81 ;*****
82                82 ; *                               *
83                83 ;*          SYMBOL DEFINITIONS          *
84                84 ;*                               *
85                85 ;*****
86                86 ;
87                87 ; ***** DMAC0A *****
88                88 ;
89                89 MAR0AE:     .EQU      H'FFFF21      ;Memory address register
90                90 ETCR00AH:   .EQU      H'FFFF24      ;Transmit cunt register 'H'
91                91 ETCR0AL:   .EQU      H'FFFF25      ;Transmit cunt register 'L'
92                92 IOAR0A:     .EQU      H'FFFF26      ;I/O address register

```

3. 応用編

```

93      00FFFF27      93  DTCR0A:      .EQU      H'FFFF27      ;Data transfer control
                                           ;register
94      00000007      94  DTE0A:      .EQU      7      ;Data transfer enable bit
95      ;
96      ;***** DMAC0B *****
97      00FFFF28      97  MAR0BE:      .EQU      H'FFFF28      ;Memory address register
98      00FFFF2C      98  ETCR0BH:      .EQU      H'FFFF2C      ;Transmit cunt register
99      00FFFF2E      99  IOAR0B:      .EQU      H'FFFF2E      ;I/O address register
100     00FFFF2F      100 DTCR0B:      .EQU      H'FFFF2F      ;Data transfer control
                                           ;register
101     00000007      101 DTE0B:      .EQU      7      ;Ddata transfer enable bit
102     ;
103     ;***** ITU *****
104     00FFFF60      104 TSTR:      .EQU      H'FFFF60      ;Timer start registre
105     ;
106     ;***** ITU ch0 *****
107     00FFFF64      107 TCR0:      .EQU      H'FFFF64      ;timer control register
108     00FFFF65      108 TIOR0:      .EQU      H'FFFF65      ;timer I/O control register
109     00FFFF66      109 TIER0:      .EQU      H'FFFF66      ;timer interrupt enable
                                           ;register
110     00FFFF67      110 TSR0      .EQU      H'FFFF67      ;timer status register
111     00FFFF6A      111 GRA0H:      .EQU      H'FFFF6A      ;General register
112     ;
113     ;***** TPC *****
114     00FFFA0      114 TPMP:      .EQU      H'FFFA0      ;TPC output mod register
115     00FFFA1      115 TPCR:      .EQU      H'FFFA1      ;TPC output control register
116     00FFFA2      116 NDERB:      .EQU      H'FFFA2      ;Next data enable register B
117     00FFFA4      117 NDRB:      .EQU      H'FFFA4      ;Next data register B
118     00FFFD4      118 PBDDR:      .EQU      H'FFFD4      ;Port B data direction
                                           ;register
119     00FFFD6      119 PBDR:      .EQU      H'FFFD6      ;Port B data register
120     ;
121     ;*****
122     ;*
123     ;*      MAIN PROGRAM : STPSMN      *
124     ;*
125     ;*****
126     ;
127     010000      127      .SECTION      PROG.CODE,LOCATE=H'010000
128     ;
129     00010000      129 STP5MN      .EQU      $
130     010000      7A0700FFFEFE      130      MOV.L      #H'FFFEFE,SP      ;Initialize stack pointer
131     ;
132     ;***** INITIALIZE ITU *****      *
133     010006      7900000A      133      MOV.W      #H'000A,R0
134     01000A      6B00FF6A      134      MOV.W      R0,@GRA0H      ;Set non overiap time
135     01000E      F801      135      MOV.B      #H'01,ROL

```

```

136 010010 3866          136          MOV.B    R0L,@TIER0      ;initialize TIER0(1->IMIEA)
137 010012 F843          137          MOV.B    ##H'43,R0L
138 010014 3864          138          MOV.B    R0L,@TCR0      ;initialize TCR0(1->CCLR1)
139                                139          ;
140                                140          ; ***** INITIALIZE DMACOA *****
141 010016 7A0000103D2    141          MOV.L    #TBL,ER0
142 01001C 0B70          142          INC.L    #1,ER0
143 01001E 01006B80FF21  143          MOV.L    ER0,@MAR0AE    ;SET MAR0A
144 010024 F8A4          144          MOV.B    ##H'A4,R0L
145 010026 3826          145          MOV.B    R0L,@IOAR0A    ;SET IOAR0A
146 010028 F804          146          MOV.B    ##H'04,R0L
147 01002A 3824          147          MOV.B    R0L,@ETCR0AH   ;STORE FIRST TRANSFER
                                ;COUNT
148 01002C F805          148          MOV.B    ##H'05,R0L
149 01002E 3825          149          MOV.B    R0L,@ETCR0AL   ;STORE TRANSFER COUNT
150 010030 2827          150          MOV.B    @DTCR0A,R0L    ;READ DTCR0A
151 010032 F810          151          MOV.B    ##H'10,R0L
152 010034 3827          152          MOV.B    R0L,@DTCR0A    ;INITIALZE DTCR0A
153 010036 7F277070      153          BSET    #DTE0A,@DTCR0A ;START DMACOA
154                                154          ;
155                                155          ; ***** INITIALIZE DMACOB *****
156 01003A 7A0000102E0    156          MOV.L    #UP,ER0
157 010040 0B70          157          INC.L    #1,ER0
158 010042 01006B80FF28  158          MOV.L    ER0,@MAR0BE    ;SET MAR0B
159 010048 F86C          159          MOV.B    ##H'6C,R0L
160 01004A 382E          160          MOV.B    R0L,@IOAR0B    ;SET IOAR0B
161 01004C 79000078      161          MOV.W    #D'120,R0
162 010050 6B80FF2C      162          MOV.W    R0,@ETCR0BH    ;STORE TRANSFER COUNT
163 010054 282F          163          MOV.B    @DTCR0B,R0L    ;READ DTCR0B
164 010056 F848          164          MOV.B    ##H'48,R0L
165 010058 382F          165          MOV.B    R0L,@DTCR0B    ;INITIALZE DTCR0B
166 01005A 7F2F7070      166          BSET    #DTE0B,@DTCR0B ;START DMACOB
167                                167          ;
168                                168          ; ***** INITIALIZE TPC *****
169 01005E F888          169          MOV.B    ##H'88,R0L
170 010060 38D6          170          MOV.B    R0L,@PBDR      ;SET FIRST OUTPUT DATA
171 010062 F8F8          171          MOV.B    ##H'F8,R0L
172 010064 38D4          172          MOV.B    R0L,@PBDDR
173 010066 38A2          173          MOV.B    R0L,@NDERB     ;SET OUTPUT TPC
174 010068 F8FC          174          MOV.B    ##H'FC,R0L
175 01006A 38A0          175          MOV.B    R0L,@TPMR      ;SET OUTPUT NON OVERLAP
176 01006C F80F          176          MOV.B    ##H'0F,R0L
177 01006E 38A1          177          MOV.B    R0L,@TPCR      ;INITIALZE TPCR
178                                178          ;
179 010070 F801          179          MOV.B    ##H'01,R0L
180 010072 3800          180          MOV.B    R0L,@PDAT_F    ;SET PDAT_F(01)
181 010074 F8E1          181          MOV.B    ##H'E1,R0L

```

3. 応用編

```

182 010076 3860          182          MOV.B    R0L,@TSTR      ;START ITU ch0
183 010078 0700          183          LOC.B    #0,CCR      ;Enable interrupt
184                      184          ;
185 01007A 0000          185 LOOP:      NOP
186 01007C 0000          186          NOP
187 01007E 40FA          187          BRA     LOOP
188                      188          ;
189                      189          ;*****
190                      190          ;*
191                      191          ;*          DENDINT : (DMAC0B END INTERRUPT PROGRAM) *
192                      192          ;*
193                      193          ;*****
194                      194          ;*
195                      195          ;*          ENTRY : NOTHING
196                      196          ;*          RETURN : NOTHING
197                      197          ;*
198                      198          ;*****
199                      199          ;
200 010080 01006DF0      200 DENDINT:    PUSH,L    ERO
201 010084 2800          201          MOV.B    @PDAT_F,R0L
202 010086 A880          202          CMP.B    #H'80,R0L
203 010088 587001B6      203          BEQ     DENDINT13
204                      204          ;
205 01008C A801          205          CMP.B    #H'01,R0L
206 01008E 58700024      206          BEQ     DENDINT01
207                      207          ;
208 010092 A802          208          CMP.B    #H'02,R0L
209 010094 58700044      209          BEQ     DENDINT02
210                      210          ;
211 010098 A804          211          CMP.B    #H'04,R0L
212 01009A 58700064      212          BEQ     DENDINT03
213                      213          ;
214 01009E A808          214          CMP.B    #H'08,R0L
215 0100A0 5870008A      215          BEQ     DENDINT04
216                      216          ;
217 0100A4 A810          217          CMP.B    #H'10,R0L
218 0100A6 58700120      218          BEQ     DENDINT10
219                      219          ;
220 0100AA A820          220          CMP.B    #H'20,R0L
221 0100AC 58700140      221          BEQ     DENDINT11
222                      222          ;
223 0100B0 A840          223          CMP.B    #H'40,R0L
224 0100B2 58700160      224          BEQ     DENDINT12
225                      225          ;
226                      226          ; ***** NEXT CONSTANT CW *****
227                      227          ;
228 0100B6 7A0000103D0    288 DENDINT01:  MOV.L    #CONSTANT,ERO

```

```

229 0100BC 01006B80FF28 229          MOV.L    ER0,@MAR0BE      ;SET DATA ADDRESS
230 0100C2 790007D0      230          MOV.W    #D'2000,R0
231 0100C6 6B80FF2C      231          MOV.W    R0,@ETCR0BH     ;STORE TRANSFER COUNT
232 0100CA 282F          232          MOV.B    @DTCR0B,R0L     ;READ DTCR0B
233 0100CC F818          233          MOV.B    #H'18,R0L
234 0100CE 382F          234          MOV.B    R0L,@DTCR0B    ;DMAC08 IDOL MODE
235 0100D0 F802          235          MOV.B    #H'02,R0L
236 0100D2 3800          236          MOV.B    R0L,@PDAT_F    ;SET PDAT_F(02)
237 0100D4 7F2F7070    237          BSET    #DTE0B,@DTCR0B ;START DMAC0B
238 0100D8 580001FE    238          BRA     DENDINT99
239                                239          ;
240                                240          ; ***** NEXT SLUE DOWN CW *****
241                                241          ;
242 0100DC 7A00000103CE 242 DENDINT02: MOV.L    #DOWN,ER0
243 0100E2 01006B80FF28 243          MOV.L    ER0,@MAR0BE     ;Set data address
244 0100E8 79000078      244          MOV.W    #D'120,R0
245 0100EC 6B80FF2C      245          MOV.W    R0,@ETCR0BH     ;Store transfer count
246 0100F0 282F          246          MOV.B    @DTCR0B,R0L     ;READ DTCR0B
247 0100F2 F868          247          MOV.B    #H'68,R0L
248 0100F4 382F          248          MOV.B    R0L,@DTCR0B    ;Set DTCR0B
                                ;(DTID->1,DTIE->1)
249 0100F6 F804          249          MOV.B    #H'04,R0L
250 0100F8 3800          250          MOV.B    R0L,@PDAT_F    ;Set PDAT_F(04)
251 0100FA 7F2F7070    251          BSET    #DTE0B,@DTCR0B
252                                252          ;
253 0100FE 580001D8    253          BRA     DENDINT99
254                                254          ;
255                                255          ; ***** NEXT STOP *****
256                                256          ;
257 010102 F801          257 DENDINT03: MOV.B    #H'01,R0L
258 010104 3825          258          MOV.B    R0L,@ETCR0AL    ;STORE FIRST TRANSFER
                                ;COUNT
259 010105 3824          259          MOV.B    R0L,@ETCR0AH    ;STORE TRANSFER COUNT
260                                260          ;
                                ;Stop DMAC0A
261 010108 7A00000103D0 261          MOV.L    #CONSTANT,ER0
262 01010E 01006B80FF28 262          MOV.L    ER0,@MAR0BE     ;Set data address
263 010114 790003E8      263          MOV.W    #D'1000,R0
264 010118 6B80FF2C      264          MOV.W    R0,@ETCR0BH     ;Store transfer count
265 01011C 282F          265          MOV.B    @DTCR0B,R0L     ;Read DTCR0B
266 01011E F818          266          MOV.B    #H'18,R0L
267 010120 382F          267          MOV.B    R0L,@DTCR0B    ;Set DTCR0B(RPE->1,
                                ;DTIE->1)
268 010122 F808          268          MOV.B    #H'08,R0L
269 010124 3800          269          MOV.B    R0L,@PDAT_F    ;Set PDAT_F(08)
270 010126 7F2F7070    270          BSET    #DTE0B,@DTCR0B ;Start DMAC0B
271 01012A 580001AC    271          BRA     DENDINT99
272                                272          ;

```

3. 応用編

```

273                273 ; ***** NEXT SLUE UP CCW *****
274                274 ;
275 01012E 7A0000102E0 275 DENDINT04:  MOV.L    #UP,ER0
276 010134 01006B80FF28 276                MOV.L    ER0,@MAR0BE ;Set data address
277 01013A 79000078      277                MOV.W    #D'120,R0
278 01013E 6B80FF2C      278                MOV.W    R0,@ETCR0BH ;STORE TRANSFER COUNT
279 010142 282F          279                MOV.B    @DTCR0B,R0L ;Read DTCR0B
280 010144 F848          280                MOV.B    ##'48,R0L
281 010146 382F          281                MOV.B    R0L,@DTCR0B ;Set DTCR0B(DTIE->1)
282                282 ;
283 010148 7EA47370      283                BTST    #7,@NDRB ;Set NDRB
284 01014C 58600018      284                BNE     DENDINT05
285 010150 7EA47360      285                BTST    #6,@NDRB
286 010154 58700028      286                BEQ     DENDINT06
287 010158 F860          287                MOV.B    ##'60,R0L
288 01015A 38A4          288                MOV.B    R0L,@NDRB
289 01015C F803          289                MOV.B    ##'03,R0L
290 01015E 3824          290                MOV.B    R0L,@ETCR0AH
291 010160 F805          291                MOV.B    ##05,R0L
292 010162 3825          292                MOV.B    R0L,@ETCR0AL
293 010164 5800004C      293                BRA     DENDINT09
294                294 ;
295 010168 7EA47350      295 DENDINT05:  BTST    #6,@NDRB
296 01016C 58700028      296                BEQ     DENDINT07
297 010170 F8C0          297                MOV.B    ##'C0,R0L
298 010172 38A4          298                MOV.B    R0L,@NDRB
299 010174 F802          299                MOV.B    ##'02R0L
300 010176 3824          300                MOV.B    R0L,@ETCR0AH
301 010178 F805          301                MOV.B    ##'05,R0L
302 01017A 3825          302                MOV.B    R0L,@ETCR0AL
303 01017C 58000034      303                BRA     DENDINT09
304                304 ;
305 010180 7EA47350      305 DENDINT06:  BTST    #5,@NDRB
306 010184 58700020      306                BEQ     DENDINT08
307 010188 F830          307                MOV.B    ##'30,R0L
308 01018A 38A4          308                MOV.B    R0L,@NDRB
309 01018C F804          309                MOV.B    ##'04,R0L
310 01018E 3824          310                MOV.B    R0L,@ETCR0AH
311 010190 F805          311                MOV.B    ##'05,R0L
312 010192 3825          312                MOV.B    R0L,@ETCR0AL
313 010194 5800001C      313                BRA     DENDINT09
314                314 ;
315 010198 F888          315 DENDINT07:  MOV.B    ##'88,R0L
316 01019A 38A4          316                MOV.B    R0L,@NDRB
317 01019C F801          317                MOV.B    #01,R0L
318 01019E 3824          318                MOV.B    R0L,@ETCR0AH
319 0101A0 F805          319                MOV.B    ##'05,R0L

```



```

320 0101A2 3825          320          MOV.B    R0L,@ETCR0AL
321 0101A4 5800000C     321          BRA      DENDINT09
322                    322 ;
323 0101A8 F818         323 DENDINT08: MOV.B    ##H'18,R0L
324 0101AA 38A4         324          MOV.B    R0L,@NDRB
325 0101AC F805         325          MOV.B    #05,R0L
326 0101AE 3824         326          MOV.B    R0L,@ETCR0AH
327 0101B0 F805         327          MOV.B    ##H'05,R0L
328 0101B2 3825         328          MOV.B    R0L,@ETCR0AL
329                    329 ;
330 0101B4 2827         330 DENDINT09: MOV.B    @DTCR0A,R0L    ;Read DTCR0A
331 0101B6 F830         331          MOV.B    ##H'30,R0L
332 0101B8 3827         332          MOV.B    R0L,@DTCR0A    ;Set DTCR0A(DTID->1,
                                   ;RPE->1)
333                    333 ;                                   ;Start DMAC0A
334                    334 ;
335 0101BA F810         335          MOV.B    ##H'10,R0L
336 0101BC 3800         336          MOV.B    R0L,@PDAT_F    ;Set PDAT_F(10)
337                    337 ;
338 0101BE 7F2F7070     338          BSET    #DTE0B,@DTCR0B ;Start DMAC0B
339 0101C2 7F277070     339          BSET    #DTE0A,@DTCR0A ;Start DMAC0A
340                    340 ;
341 0101C6 58000110     341          BRA      DENDINT99
342                    342 ;
343                    343 ; ***** NEXT CONSTANT CCW *****
344                    344 ;
345 0101CA 7A00000103D0  345 DENDINT10: MOV.L    #CONSTANT,ER0
346 0101D0 01006B80FF28 346          MOV.L    ER0,@MAR0BE    ;SET DATA ADDRESS
347 0101D6 79000BB8     347          MOV.W    #D'3000,R0
348 0101DA 6B80FF2C     348          MOV.W    R0,@ETCR0BH    ;STORE TRANSFER COUNT
349 0101DE 282F         349          MOV.B    @DTCR0B,R0L    ;READ DTCR0B
350 0101E0 F818         350          MOV.B    ##H'18,R0L
351 0101E2 382F         351          MOV.B    R0L,@DTCR0B    ;DMAC0B IDOL MODE
352 0101E4 F820         352          MOV.B    ##H'20,R0L
353 0101E6 3800         353          MOV.B    R0L,@PDAT_F    ;SET PTAT_F(20)
354 0101E8 7F2F7070     354          BSET    #DTE0B,@DTCR0B ;START DMAC0B
355 0101EC 580000EA     355          BRA      DENDINT99
356                    356 ;
357                    357 ; ***** NEXT SLUE DOWN CCW *****
358                    358 ;
359 0101F0 7A00000103CE  359 DENDINT11: MOV.L    #DOWN,ER0
360 0101F6 01006B80FF28 360          MOV.L    ER0,@MAR0BE    ;Set data address
361 0101FC 79000078     361          MOV.W    #D'120,R0
362 010200 6B80FF2C     362          MOV.W    R0,@ETCR0BH    ;Store transfer count
363 010204 282F         363          MOV.B    @DTCR0B,R0L    ;READ DTCR0B
364 010206 F868         364          MOV.B    ##H'68,R0L

```

3. 応用編

```

365 010208 382F          365          MOV.B      R0L,@DTCR0B      ;Set DTCR0B
                                           ;(DTID->1,DTIE->1)
366 01029A F840          366          MOV.B      #H'40,R0L
367 01020C 3800          367          MOV.B      R0L,@PDAT_F      ;Set PTAT_F(40)
368 01020E 7F2F7070      368          BSET       #DTE0B,@DTCR0B
                                           ;Start DMAC0B
369 010212 580000C4      369          BRA        DENDINT99
370                                370          ;
371                                371          ; ***** NEXT STOP *****
372                                372          ;
373 010216 F801          373 DENDINT12: MOV.B      #H'01,R0L
374 010218 3825          374          MOV.B      R0L,@ETCR0AL      ;STORE FIRST TRANSFER COUNT
375 01021A 3824          375          MOV.B      R0L,@ETCR0AH      ;STORE TRANSFER COUNT
376                                376          ;
                                           ;DMAC0A Stop
377 01021C 7A0000103D0    377          MOV.L      #CONSTANT,ER0
378 010222 01006B80FF28    378          MOV.L      ER0,@MAR0BE      ;Set data address
379 010228 790003E8      379          MOV.W      #D'1000,R0
380 01022C 6B80FF2C      380          MOV.W      R0,@ETCR0BH      ;Store transfer count
381 010230 282F          381          MOV.B      @DTCR0B,R0L      ;Read DTCR0B
382 010232 F818          382          MOV.B      #H'18,R0L
383 010234 382F          383          MOV.B      R0L,@DTCR0B      ;Set DTCR0B(RPE->1,DTIE->1)
384 010236 F880          384          MOV.B      #H'80,R0L
385 010238 3800          385          MOV.B      R0L,@PDAT_F      ;Set PDAT_F(80)
386 01023A 7F2F7070      386          BSET       #DTE0B,@DTCR0B
                                           ;Start DMAC0B
387 01023E 58000098      387          BRA        DENDINT99
388                                388          ;
389                                389          ; ***** NEXT SLUE UP CW *****
390                                390          ;
391 010242 7A0000102E0    391 DENDINT13: MOV.L      #UP,ER0
392 010248 01006B80FF28    392          MOV.L      ER0,@MAR0BE      ;Set data address
393                                393          ;
394 01024E 79000078      394          MOV.W      #D'120,R0
395 010252 6B80FF2C      395          MOV.W      R0,@ETCR0BH      ;Store transfer count
396                                396          ;
397 010256 282F          397          MOV.B      @DTCR0B,R0L      ;Read DTCR0B
398 010258 F848          398          MOV.B      #H'48,R0L
399 01025A 382F          399          MOV.B      R0L,@DTCR0B      ;Set DTCR0B(DTIE->1)
400                                400          ;
401 01025C 7EA47370      401          BTST       #7,@NDRB      ;Set NDRB
402 010260 58600018      402          BNE       DENDINT14
403 010264 7EA47360      403          BTST       #6,@NDRB
404 010268 58700028      404          BEQ       DENDINT15
405 01026C F860          405          MOV.B      #H'60,R0L
406 01026E 38A4          406          MOV.B      R0L,@NDRB
407 010270 F803          407          MOV.B      #H'03,R0L
408 010272 3824          408          MOV.B      R0L,@ETCR0AH

```

```

409 010274 F805          409          MOV.B    #H'05,R0L
410 010276 3825          410          MOV.B    R0L,@ETCR0AL
411 010278 5800004C      411          BRA      DENDINT18
412                    412 ;
413 01027C 7EA47360      413 DENDINT14: BTST    #6,@NDRB
414 010280 58700028      414          BEQ      DENDINT16
415 010284 F8C0          415          MOV.B    #H'C0,R0L
416 010286 38A4          416          MOV.B    R0L,@NDRB
417 010288 F804          417          MOV.B    #H'04,R0L
418 01028A 3824          418          MOV.B    R0L,@ETCR0AH
419 01028C F805          419          MOV.B    #H'05,R0L
420 01028E 3825          420          MOV.B    R0L,@ETCR0AL
421 010290 58000034      421          BRA      DENDINT18
422                    422 ;
423 010294 7EA47350      423 DENDINT15: BTST    #5,@NDRB
424 010298 58700020      424          BEQ      DENDINT17
425 01029C F830          425          MOV.B    #H'30,R0L
426 01029E 38A4          426          MOV.B    R0L,@NDRB
427 0102A0 F802          427          MOV.B    #H'02,R0L
428 0102A2 3824          428          MOV.B    R0L,@ETCR0AH
429 0102A4 F805          429          MOV.B    #H'05,R0L
430 0102A6 3825          430          MOV.B    R0L,@ETCR0AL
431 0102A8 5800001C      431          BRA      DENDINT18
432                    432 ;
433 0102AC F888          433 DENDINT16: MOV.B    #H'88,R0L
434 0102AE 38A4          434          MOV.B    R0L,@NDRB
435 0102B0 F805          435          MOV.B    #H'05,R0L
436 0102B2 3824          436          MOV.B    R0L,@ETCR0AH
437 0102B4 F805          437          MOV.B    #H'05,R0L
438 0102B6 3825          438          MOV.B    R0L,@ETCR0AL
439 0102B8 5800000C      439          BRA      DENDINT18
440                    440 ;
441 0102BC F818          441 DENDINT17: MOV.B    #H'18,R0L
442 0102BE 38A4          442          MOV.B    R0L,@NDRB
443 0102C0 F801          443          MOV.B    #01,R0L
444 0102C2 3824          444          MOV.B    R0L,@ETCR0AH
445 0102C4 F805          445          MOV.B    #H'05,R0L
446 0102C6 3825          446          MOV.B    R0L,@ETCR0AL
447                    447
448                    448 ;
449 0102C8 2827          449 DENDINT18: MOV.B    @DTCR0A,R0L ;READ DTCR0A
450 0102CA F810          450          MOV.B    #H'10,R0L
451 0102CC 3827          451          MOV.B    R0L,@DTCR0A ;Set DTCR0A(RPE->1)
452                    452 ; ;Start DMAC0A
453                    453 ;
454 0102CE F801          454          MOV.B    #H'01,R0L
455 0102D0 3800          455          MOV.B    R0L,@PDAT_F ;Set PDAT_F(01)

```

3. 応用編

```

456                                     456 ;
457 0102D2 7F2F7070                   457          BSET          #DTE0B,@DTCR0B
                                         ;Start DMAC0B
458 0102D6 7F277070                   458          BSET          #DTE0A,@DTCR0A
                                         ;Start DMAC0A
459                                     459 ;
460 0102DA 01006D70                   460 DENDINT99: POP,L      ER0
461 0102DE 5670                         461          RTE
462                                     462 ;
463                                     463 ;*****
464                                     464 ;*                               *
465                                     465 ;*          DATA TABLE (TIMER PERIOD) *
466                                     466 ;*                               *
467                                     467 ;*****
468                                     468 ;
469 0102E0 F424B1646EA361D2           469 UP          .DATA.W      D'62500,D'45412,D'28323,D'25042,D'21761,D'20058,
470 0102E8 55014E5A47B34371           470 +          D'18355,D'17265,D'16175,D'15401,D'14626,D'14039
      0102F0 3F2F3C29392236D7
471 0102F8 348C32BB30EA2F6E           471          .DATA.W      D'13452,D'12987,D'12522,D'12142,D'11762,D'11444,
472 010300 2DF22CB42B752A66           472 +          D'11125,D'10854,D'10582,D'10347,D'10111,D'9905
      010308 2956286B277F26B1
473 010310 25E226BB247423D1           473          .DATA.W      D'9698,D'9915,D'9332,D'9169,D'9005,D'8857,
474 010318 232D22992205217F           474 +          D'8709,D'8575,D'8441,D'8319,D'8196,D'8084
      010320 20F9207F20041F94
475 010328 1F241EBD1E551DF5           475          .DATA.W      D'7972,D'7869,D'7765,D'7669,D'7573,D'7484,
476 012330 1D951D3C1CE31C90           476 +          D'7395,D'7312,D'7229,D'7151,D'7073,D'7000
      010338 1C3D1BEF1BA11B58
477 010340 1B0F1ACB1A861A46           477          .DATA.W      D'6927,D'6859,D'6790,D'6726,D'6661,D'6600,
478 010348 1A0519C8198B1951           478 +          D'6539,D'6481,D'6423,D'6368,D'6313,D'6261
      010350 191718E018A91875
479 010358 1841181017DE17AF           479          .DATA.W      D'6209,D'6160,D'6110,D'6063,D'6015,D'5970,
480 010360 177F1752172416F9           480 +          D'5924,D'5881,D'5838,D'5797,D'5755,D'5716
      010368 16CE16A5167B1654
481 010370 162C160615E015BB           481          .DATA.W      D'5676,D'5638,D'5600,D'5563,D'5526,D'5491,
482 010378 159615731550152E           482 +          D'5456,D'5422,D'5388,D'5356,D'5323,D'5292
      010380 150C14EC14CB14AC
483 010388 148C146E144F1432           483          .DATA.W      D'5260,D'5230,D'5199,D'5170,D'5140,D'5112,
484 010390 141413F813DC13C1           484 +          D'5084,D'5057,D'5029,D'5002,D'4975,D'4950
      010398 13A5138A136F1356
485 0103A0 133C1323130A12F2           485          .DATA.W      D'4924,D'4899,D'4874,D'4850,D'4825,D'4802,
486 0103A8 12D912C212AA1294           486 +          D'4778,D'4756,D'4733,D'4711,D'4688,D'4667
      0103B0 127D12671250123B
487 0103B8 1225121011FB11E7           487          .DATA.W      D'4645,D'4624,D'4603,D'4583,D'4562,D'4542,
488 0103C0 11D211BE11AA1197           488 +          D'4522,D'4503,D'4483,D'4465,D'4446
      0103CB 11831171115E
489 0103CE 115E                         489 DOWN     .DATA.W      D'4446
490                                     490 ;*

```

```
491          491 ;***** CONSTANT *****
492          492 ;*
493 0103D0 115E 493 CONSTANT .DATA.W D'4446
494          494 ;*
495          495 ;
496          496 ;*****
497          497 ;* *
498          498 ;* DATA TABLE (OUTPUT PATTERN) *
499          499 ;* *
500          500 ;*****
501          501 ;
502 0103D2 603090C0 502 TBL .DATA.B H'60,H'30,H'90,H'C0
503          503 ;
504          504 .END
*****TOTAL ERRORS 0
*****TOTAL WARNINGS 0
```

H8/300Hシリーズ アプリケーションノート 内蔵I/O編

発行年月 1994年2月 第1版

2000年6月 Rev.2.00

発行 株式会社ルネサス テクノロジ 営業企画統括部

〒100-0004 東京都千代田区大手町 2-6-2

編集 株式会社ルネサス小平セミコン 技術ドキュメント部

© 2000 Renesas Technology Corp., All rights reserved. Printed in Japan.



営業お問合せ窓口
株式会社ルネサス販売

<http://www.renesas.com>

本			社	〒100-0004	千代田区大手町2-6-2 (日本ビル)	(03) 5201-5350
京	浜	支	社	〒212-0058	川崎市幸区鹿島田890-12 (新川崎三井ビル)	(044) 549-1662
西	東	京	支	〒190-0023	立川市柴崎町2-2-23 (第二高島ビル2F)	(042) 524-8701
札	幌	支	店	〒060-0002	札幌市中央区北二条西4-1 (札幌三井ビル5F)	(011) 210-8717
東	北	支	社	〒980-0013	仙台市青葉区花京院1-1-20 (花京院スクエア13F)	(022) 221-1351
い	わ	き	支	〒970-8026	いわき市平小太郎町4-9 (損保ジャパンいわき第二ビル3F)	(0246) 22-3222
茨	城	支	社	〒312-0034	ひたちなか市堀口832-2 (日立システムプラザ勝田1F)	(029) 271-9411
新	潟	支	店	〒950-0087	新潟市東大通1-4-2 (新潟三井物産ビル3F)	(025) 241-4361
松	本	支	社	〒390-0815	松本市深志1-2-11 (昭和ビル7F)	(0263) 33-6622
中	部	営	本	〒460-0008	名古屋市中区栄3-13-20 (栄センタービル4F)	(052) 261-3000
浜	松	支	店	〒430-7710	浜松市板屋町111-2 (浜松アクタワー10F)	(053) 451-2131
西	部	営	本	〒541-0044	大阪市中央区伏見町4-1-1 (明治安田生命大阪御堂筋ビル)	(06) 6233-9500
北	陸	支	社	〒920-0031	金沢市広岡3-1-1 (金沢パークビル8F)	(076) 233-5980
中	国	支	社	〒730-0036	広島市中区袋町5-25 (広島袋町ビルディング8F)	(082) 244-2570
松	山	支	店	〒790-0003	松山市三番町4-4-6 (GEエジソンビル松山2号館3F)	(089) 933-9595
鳥	取	支	店	〒680-0822	鳥取市今町2-251 (日本生命鳥取駅前ビル)	(0857) 21-1915
九	州	支	社	〒812-0011	福岡市博多区博多駅前2-17-1 (ヒロカネビル本館5F)	(092) 481-7695
鹿	児	島	支	〒890-0053	鹿児島市中央町12-2 (明治安田生命鹿児島中央町ビル)	(099) 284-1748

技術的なお問合せおよび資料のご請求は下記へどうぞ。

総合お問合せ窓口：カスタマサポートセンタ E-Mail: csc@renesas.com



H8/300H シリーズ
アプリケーションノート 内蔵 I/O 編



ルネサス エレクトロニクス株式会社
神奈川県川崎市中原区下沼部1753 〒211-8668

RJJ05B0473-0200H