

RH850/U2B6

レゾルバ付き永久磁石同期モータの制御

要旨

本アプリケーションノートでは、RH850/U2B6 でレゾルバ付き永久磁石同期モータを制御する方法について説明します。

動作確認デバイス

RH850/U2B6

本アプリケーションノートを他のマイコンへ適用する場合、そのマイコンの仕様にあわせて変更し、十分評価してください。

目次

1.	レゾルバ付永久磁石同期モータのベクトル制御	3
1.1	PMSM の動作	3
1.1.1	合成磁場とトルク	4
1.2	モータ制御の概念	4
1.3	U/V/W 相電圧から PWM 出力へ	6
1.3.1	三角波比較法	6
1.4	dq 座標系の PMSM の電圧方程式(参考資料)	7
1.4.1	電圧方程式	7
1.4.2	非干渉制御	7
2.	仕様	8
3.	動作条件	11
4.	使用端子と機能	12
5.	ハードウェア動作	13
5.1	動作概要	13
5.2	R/D コンバータ 3AL (RDC3AL)	13
5.3	sDMA Controller (sDMAC)	13
5.4	ペリフェラルインタコネクション (PIC)	13
5.5	A/D コンバータ (ADCK)	13
5.6	汎用タイマーモジュール (GTM)	14
5.7	CPU 割込み処理 PWM コンペア値生成	17
5.8	ファイル構成	17
5.9	関数一覧	18
5.10	関数仕様	19
5.11	フローチャート	23
6.	参考ドキュメント	29
	改訂記録	30

1. レゾルバ付永久磁石同期モータのベクトル制御

本章では、レゾルバ付永久磁石同期モータ (Permanent Magnetic Synchronous Motor、以下「PMSM」という。) のベクトル制御の概念について説明します。

1.1 PMSM の動作

PMSM は筐体などに固定されているステータと回転するロータで構成されています。ステータには 120° ごとに U 相コイル、V 相コイル、W 相コイルが配置されており、ロータには永久磁石が組み込まれています。

コイルの両端に電圧を印加すると電流が流れ、それにより磁場が生成されます。この磁場内に永久磁石を置くと永久磁石に引力や斥力が働きます。U 相、V 相、W 相のコイルに電流を流し、各コイルが作る磁場をベクトル合成した合成磁場に引き寄せられるようにロータが動きます。この合成磁場を回転させることにより、ロータが回転します。U 相、V 相、W 相の各コイルに、 120° の位相差をもつ正弦波の電流を流すと、磁場の大きさが同じで回転速度が一定な回転磁場を作ることができます。図 1.1 に PMSM の構造と合成磁場を示します。

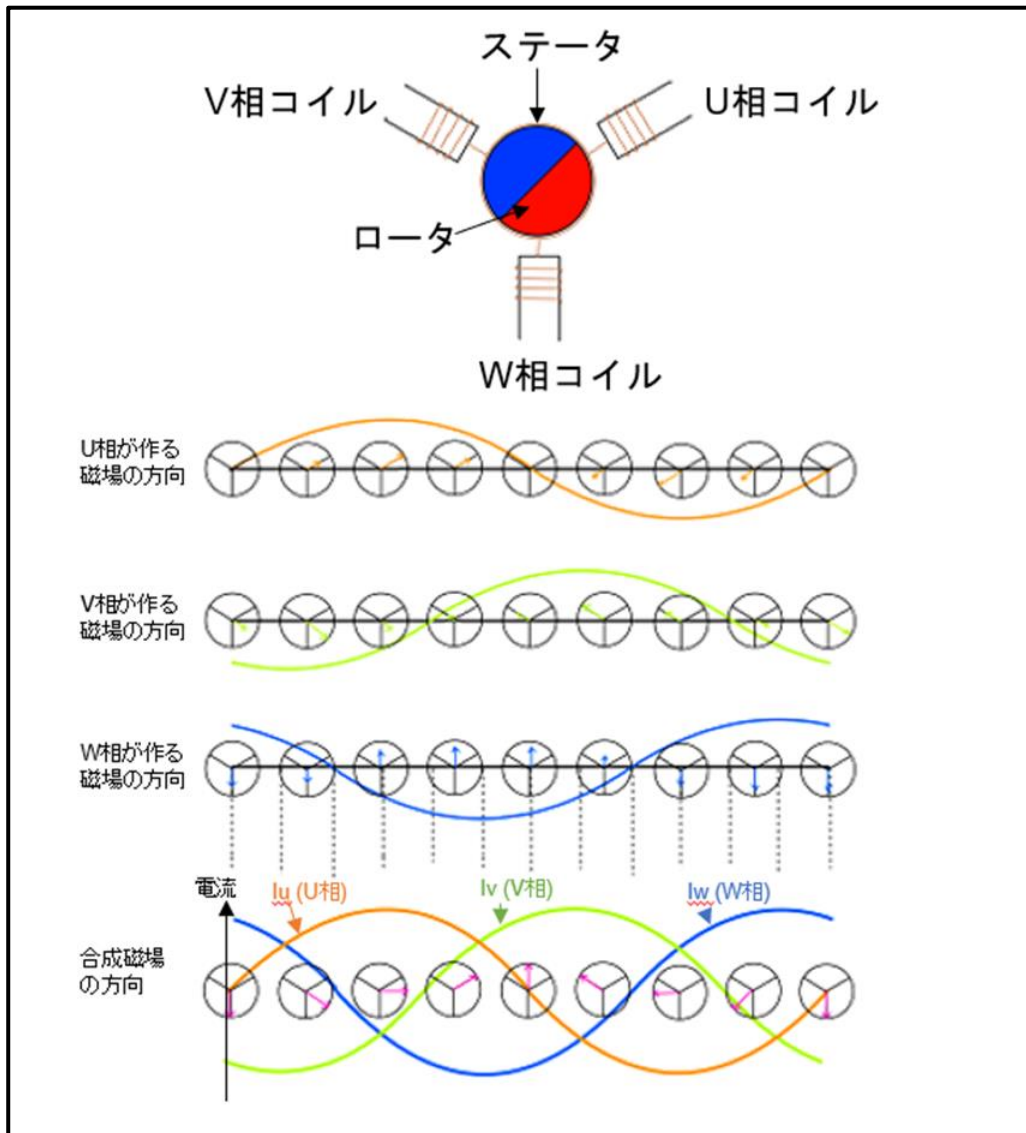


図 1.1 PMSM の構造と合成磁場

1.1.1 合成磁場とトルク

図 1.2 に合成磁場とトルクの関係を示します。

永久磁石の磁極と合成磁場のなす角を 90° にした場合、トルク(永久磁石を回す力)が最大になります。この磁極と直交した合成磁場を直交磁場と呼びます。ここで、永久磁石の磁極方向を d 軸、直交磁場の方向を q 軸とします。

モータを効率よく回転させるためには、常に直交磁場が生成されるよう制御する必要があります。直交磁場を生成するために、q 軸方向にコイルが配置されていると仮定し、そのコイルに流す電流を q 軸電流 (I_q) とします。このとき直交磁場の強さは q 軸電流の大きさに比例します。

モータの回転速度が想定よりも遅い場合は q 軸電流を大きくし、トルクを強め加速します。また、モータの回転速度が想定よりも速い場合は q 軸電流を小さくし、トルクを弱め減速します。

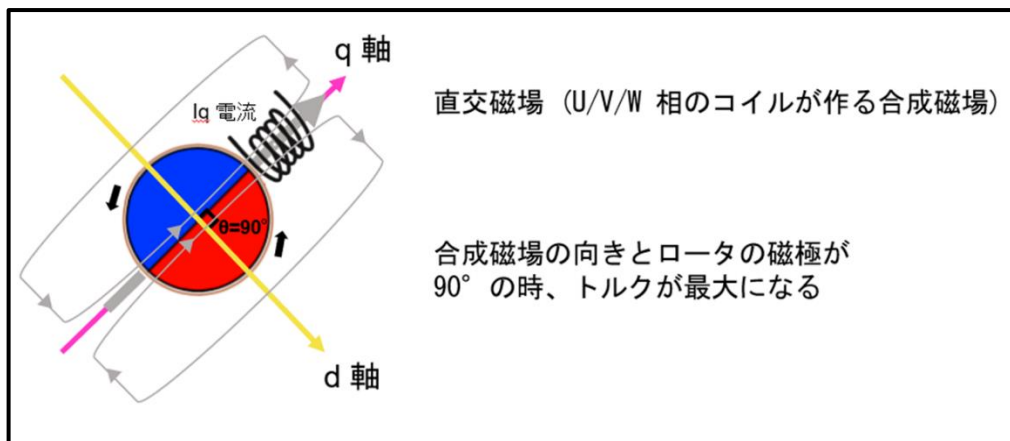


図 1.2 合成磁場とトルクの関係

1.2 モータ制御の概念

モータを効率良く制御するためには、回転速度に応じて d 軸、q 軸電流を調整する必要がありますので、この d 軸、q 軸電流を求めます。q 軸電流は流れる電流の大きさと流れる向きがあり、ベクトル量として取り扱われるため、この制御方法をベクトル制御と呼びます。

目標の回転速度と現在の回転速度の差分から d 軸、q 軸電流の目標値(指令値)を決めます。簡単なモデルではトルク成分に寄与しない d 軸電流を 0 とみなして処理します。d 軸、q 軸の各電流指令値と、測定したコイル電流から求めた d 軸、q 軸電流値の差分(偏差)から次の電圧値を求めます。このように、前回の結果(現状のモータ回転速度やコイル電流値)をフィードバックさせて次の出力に反映する制御をフィードバック制御と呼びます。

現在の電流値の偏差から次の電圧値を求めるには、現在の偏差に比例した成分と、過去の累積した偏差に比例した成分を加算します。この処理を繰り返しおこなうことにより出力電圧、その結果流れるコイル電流を調整し、目標の回転速度に合わせ込みます。この方法を PI 制御(Proportional-Integral control)と呼びます。

コイル電流値を座標変換(3 相 2 相変換→回転座標変換)することにより、q 軸電流値と d 軸電流値を求めます。図 1.3 に座標変換を示します。

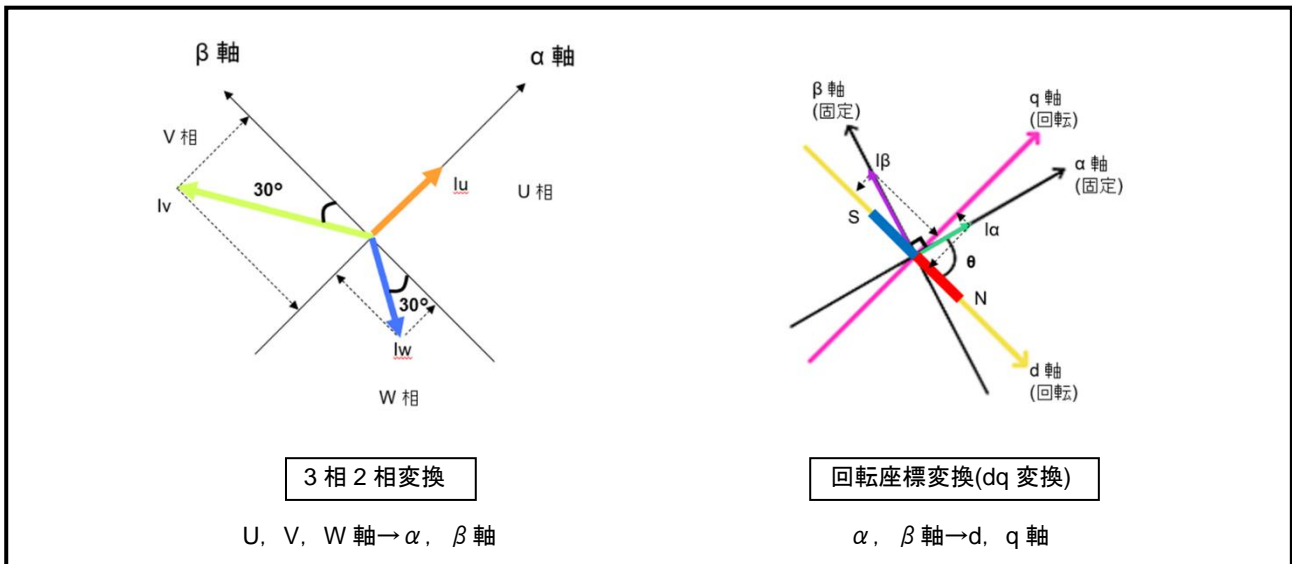


図 1.3 座標変換

n 回目に測定した U/V/W 各コイルに流れる電流値 I_u 、 I_v 、 I_w を 3 相 2 相変換することにより α 、 β 軸の電流値 I_α 、 I_β を求めます。

$$\begin{pmatrix} I_\alpha(n) \\ I_\beta(n) \end{pmatrix} = \sqrt{\frac{2}{3}} \begin{pmatrix} \cos 0 & \cos \frac{2}{3}\pi & \cos \frac{4}{3}\pi \\ \sin 0 & \sin \frac{2}{3}\pi & \sin \frac{4}{3}\pi \end{pmatrix} \begin{pmatrix} I_u(n) \\ I_v(n) \\ I_w(n) \end{pmatrix}$$

3 相 2 相変換で得た I_α 、 I_β を回転座標変換することにより d、q 軸の電流値 I_d と I_q を求めます。

$$\begin{pmatrix} I_d(n) \\ I_q(n) \end{pmatrix} = \begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} I_\alpha(n) \\ I_\beta(n) \end{pmatrix}$$

d 軸電流と q 軸電流の偏差を求め、PI 制御により d 軸電圧、q 軸電圧を求めます。

$$\varepsilon_d(n) = I_{dt} - I_d(n)$$

$$\varepsilon_q(n) = I_{qt} - I_q(n)$$

$$V_d(n) = V_d(n-1) + K_P \cdot \{\varepsilon_d(n) - \varepsilon_d(n-1)\} + K_I \cdot \varepsilon_d(n) \cdot \Delta t$$

$$V_q(n) = V_q(n-1) + K_P \cdot \{\varepsilon_q(n) - \varepsilon_q(n-1)\} + K_I \cdot \varepsilon_q(n) \cdot \Delta t$$

備考	I_{dt}	: d 軸電流の指令値
	I_{qt}	: q 軸電流の指令値
	$I_d(n)$: n 回目にサンプリングした d 軸電流
	$I_q(n)$: n 回目にサンプリングした q 軸電流
	K_P	: 比例ゲイン
	K_I	: 積分ゲイン
	Δt	: サンプリング時間
	ε_d	: d 軸電流の偏差
	ε_q	: q 軸電流の偏差

ここで算出した q 軸電圧、d 軸電圧を座標変換(固定座標変換、2 相 3 相変換)することにより、U/V/W 各相の出力電圧を求めます。固定座標変換は、次式になります。

$$\begin{pmatrix} V_{\alpha}(n) \\ V_{\beta}(n) \end{pmatrix} = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} V_d(n) \\ V_q(n) \end{pmatrix}$$

2 相 3 相変換は、次式になります。

$$\begin{pmatrix} V_u(n) \\ V_v(n) \\ V_w(n) \end{pmatrix} = \sqrt{\frac{2}{3}} \begin{pmatrix} \cos 0 & \sin 0 \\ \cos \frac{2}{3}\pi & \sin \frac{2}{3}\pi \\ \cos \frac{4}{3}\pi & \sin \frac{4}{3}\pi \end{pmatrix} \begin{pmatrix} V_{\alpha}(n) \\ V_{\beta}(n) \end{pmatrix}$$

1.3 U/V/W 相電圧から PWM 出力へ

マイクロコントローラから実際に PWM 波形を出力するためには、操作量である d 軸電圧、q 軸電圧を三相電圧変換した U/V/W 相電圧(V_u , V_v , V_w)を PWM のデューティに反映させる必要があります。その方法について説明します。

1.3.1 三角波比較法

三角波比較法は PWM 制御をおこなう制御方法の一つで、図 1.4 に示すように、U/V/W 相電圧波形と三角波の大小比較で PWM 生成をおこないます。この方法はアナログ回路で PWM 生成した時代のなごりですが、今なお最も一般的に広く使われています。この三角波をキャリア波(搬送波)と呼び、その周波数は PWM の動作周期となります。U/V/W 相電圧と三角波を比較し、U/V/W 相電圧が三角波より大きいとき PWM をハイレベル(アクティブレベル)にし、三角波より小さいときはロウレベル(インアクティブレベル)にします。これにより、平均電圧を変化させていけば、図 1.4 の U/V/W 相電圧(正弦波)を疑似的に再現できます。詳細は、専門書を参照してください。

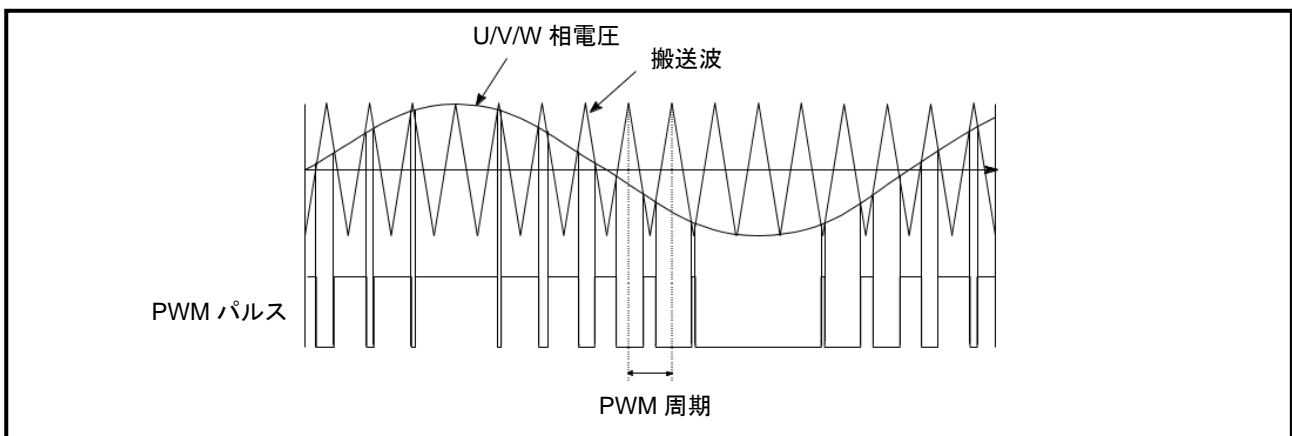


図 1.4 三角波比較法

この三角波比較法は、三角波変調とも呼びます。この三角波比較法を使い、永久磁石同期モータの 3 相電圧を PWM 制御すれば良いことになります。この PWM 出力は、相補 PWM 出力(デッドタイム付き PWM 出力)でおこないます。

1.4 dq 座標系の PMSM の電圧方程式(参考資料)

1.4.1 電圧方程式

dq 座標系における PMSM の電圧方程式は以下の式になります。電圧方程式を用いて、電流 i_d 、 i_q を電圧 v_d 、 v_q に変換できます。ただし、本サンプルプログラムでは、PI 制御を用いて電圧に変換していません。

$$\begin{bmatrix} v_d \\ v_q \end{bmatrix} = \begin{bmatrix} Ra + pL_d & -\omega L_q \\ \omega L_d & Ra + pL_q \end{bmatrix} \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \begin{bmatrix} 0 \\ \omega \phi_a \end{bmatrix} \quad p = \frac{d}{dt}$$

備考 v_d, v_q : 各相電機子電圧

R_a : 各相電機子抵抗

L_d, L_q : 各相自己インダクタンス

i_d, i_q : 各相電機子電流

ω : モータ角速度

ϕ : 永久磁石磁束 $\phi_a = \sqrt{\frac{2}{3}} \phi$

p : 微分演算子

この式の導出過程の詳細については、専門書を参照してください。

実際の制御では、この式は使用しておりません(非干渉制御導出にのみ使用)。

1.4.2 非干渉制御

1.4.1 に示されたモータの電圧方程式にラプラス変換を施し、変形すると以下の式になります。

$$I_d = \frac{1}{Ra + sL_d} (V_d + \omega L_q I_q)$$

$$I_q = \frac{1}{Ra + sL_q} \left(V_q - \omega \left(L_d I_d + \sqrt{\frac{2}{3}} \phi \right) \right)$$

備考 s : ラプラス演算子

この式から分かるように q 軸側の情報が d 軸側に、d 軸側の情報が q 軸側にあらわれています。この情報をあらかじめ引いておく制御方法を「非干渉制御」と言います。

この非干渉制御をおこなう場合、モータのパラメータを前もって知っておく必要があります。また、この制御はフィードフォワード制御となります。

実際の電流制御は PI 制御を用いますので、各電流制御(I_d , I_q)が安定していない時に他の項の影響があらわれると PI 制御の調整が難しくなります。そのため、非干渉制御を用います。

2. 仕様

本アプリケーションノートでは、12ビット A/D コンバータ(ADCK)、R/D コンバータ(RDC3AL)、Generic Timer Module(GTM)、ペリフェラルインタコネクション(PIC)、sDMA Controller(sDMAC)を用いてモータ制御をおこないます。

A/D コンバータでモータ電流値を測定し、R/D コンバータで角度情報を取得します。変換完了時に CPU にてモータ電流値と角度情報をもとにベクトル制御をおこない、PWM コンペア値を生成します。そのコンペア値をもとに GTM が三角波比較方式で正弦波 PWM 波形を生成し出力します。

本アプリケーションノートでは、CPU は電流指令値の設定と PWM コンペア値の生成のみを行い、その他の制御は CPU を介さずにハードウェアのみでおこなっています。

RH850/U2B6 は 2 モータに対応しています。

表 2.1 にモータ制御の仕様を、表 2.2 に各モジュールの設定内容を、**図 2.1** に 2 モータ対応のシステム構成図(RH850/U2B6)を、

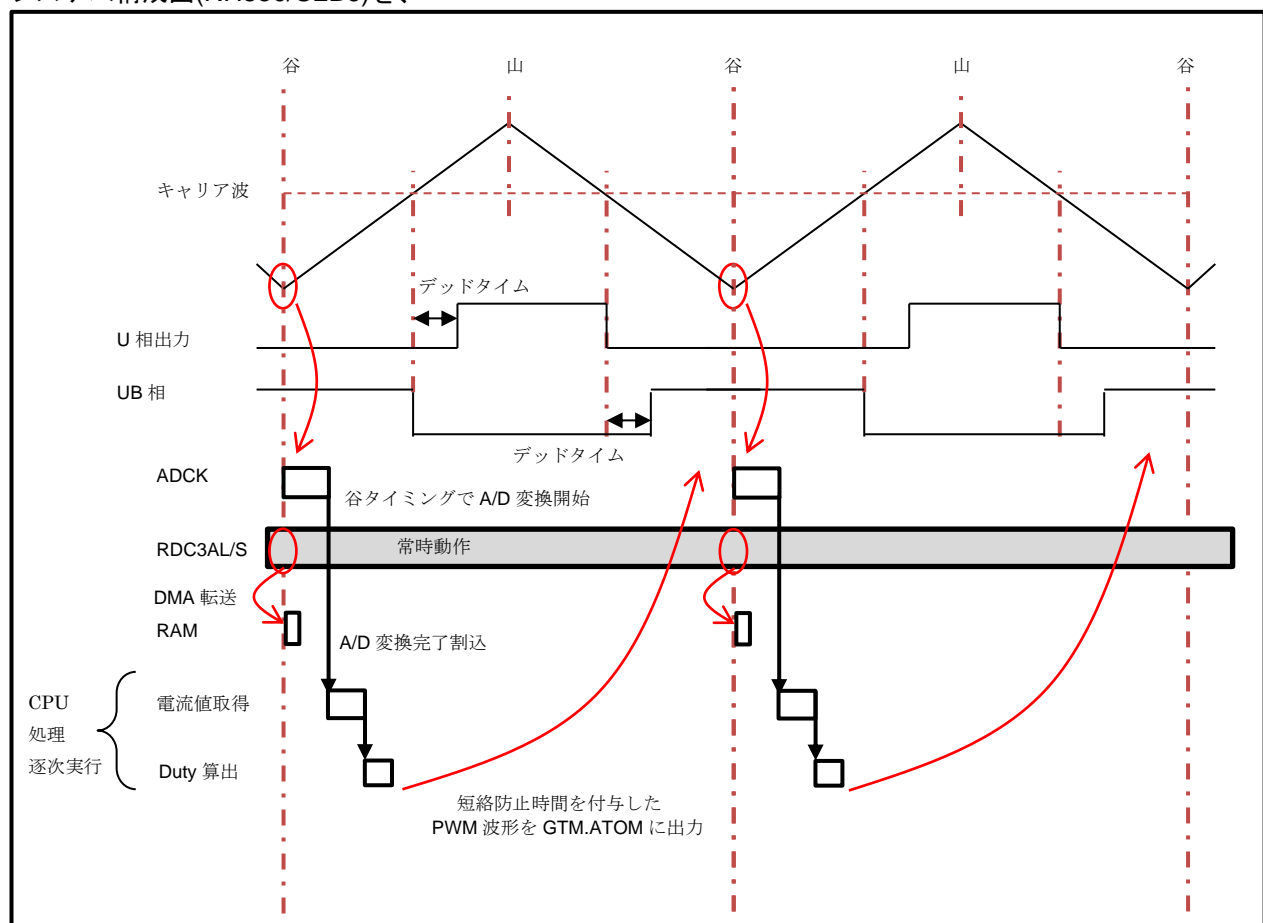


図 2.2 に全自動処理の制御フロー図を示します。

表 2.1 モータ制御の仕様

項目	仕様
出力波形	相補三相 PWM 波形
キャリア周期	8 kHz (125 usec/サイクル)
制御方式	180 度通電駆動方式
アクティブレベル	High アクティブ
短絡防止時間	2.5 usec
コンペアレジスタ、キャリア周期の更新タイミング	キャリア波の谷
A/D 変換開始タイミング	キャリア波の谷
RAM 転送タイミング	キャリア波の谷
割り込み	A/D 変換完了時 転送完了時 キャリア波谷時
モータ電流	ADCK0 で U 相、V 相、W 相の電流値を取得
モータの角度情報	RDC3AL でレゾルバからの角度情報を取得

表 2.2 各モジュールの設定内容

周辺機能	設定内容
RDC3AL	<ul style="list-style-type: none"> ・ RDC3AL 内部で生成した 10kHz の励磁信号を使用 ・ PI 補償器帯域は自動設定 ・ BIST, 自己診断は実施しない ・ 角度変換モードはモード 1 を使用
sDMAC	<ul style="list-style-type: none"> ・ ハードウェア転送要因 ・ GTM からの信号をトリガに RDC3AL からの角度値を RAM に転送
PIC	<ul style="list-style-type: none"> ・ ADCK0 スキャングループ 4 のトリガに GTM からの信号を使用
ADCK	<ul style="list-style-type: none"> ・ スキャングループ 4(SG4)の仮想チャンネル 0(V 相),1(W 相),2(U 相)を A/D 変換 ・ T&H A/D 変換モード ・ マルチサイクルスキャンモードでトリガ毎に 1 回スキャン ・ スキャングループ 4 ハードウェアトリガ入力有効 ・ スキャングループ 4 終了割り込み信号を許可
GTM	<ul style="list-style-type: none"> ・ SOMP モード (アップダウンカウントモード) ・ キャリア周期: 125us ・ デッドタイム: 2.5us ・ キャリア波の谷タイミングで A/D 変換起動
CPU 処理	<ul style="list-style-type: none"> ・ AD 完了割り込み関数にて PWM コンペア値を生成 ・ ADCK からの電流値と RDC3AL からの角度を取得

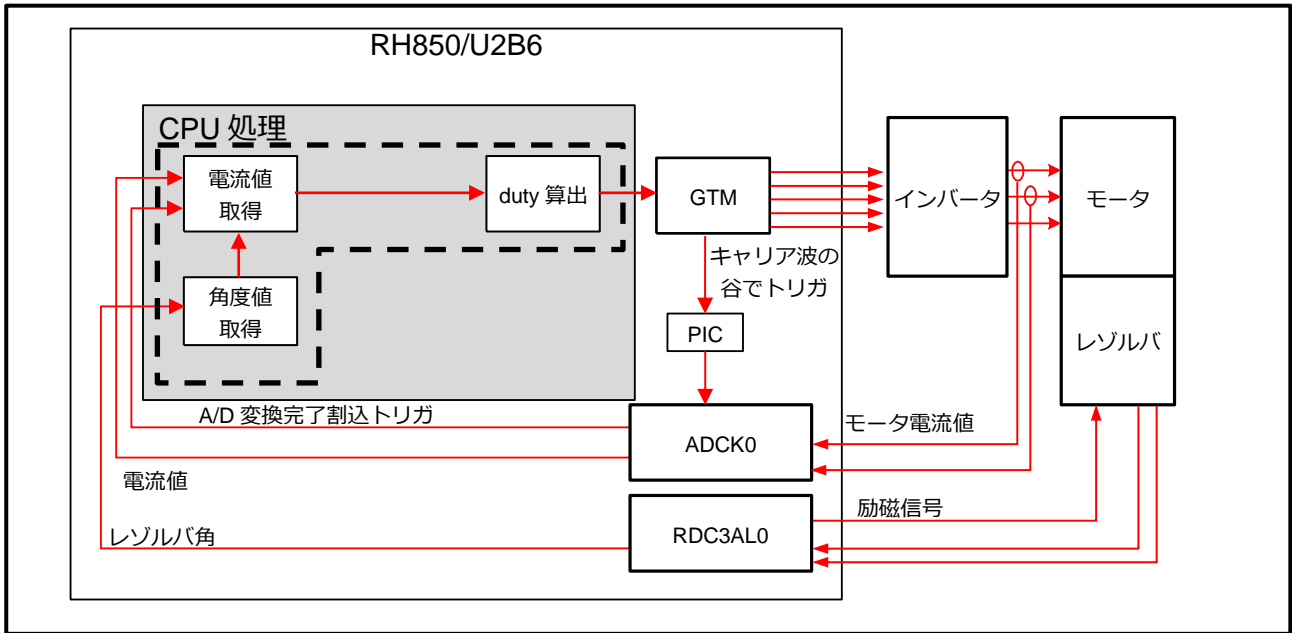


図 2.1 システム構成図 (RH850/U2B6)

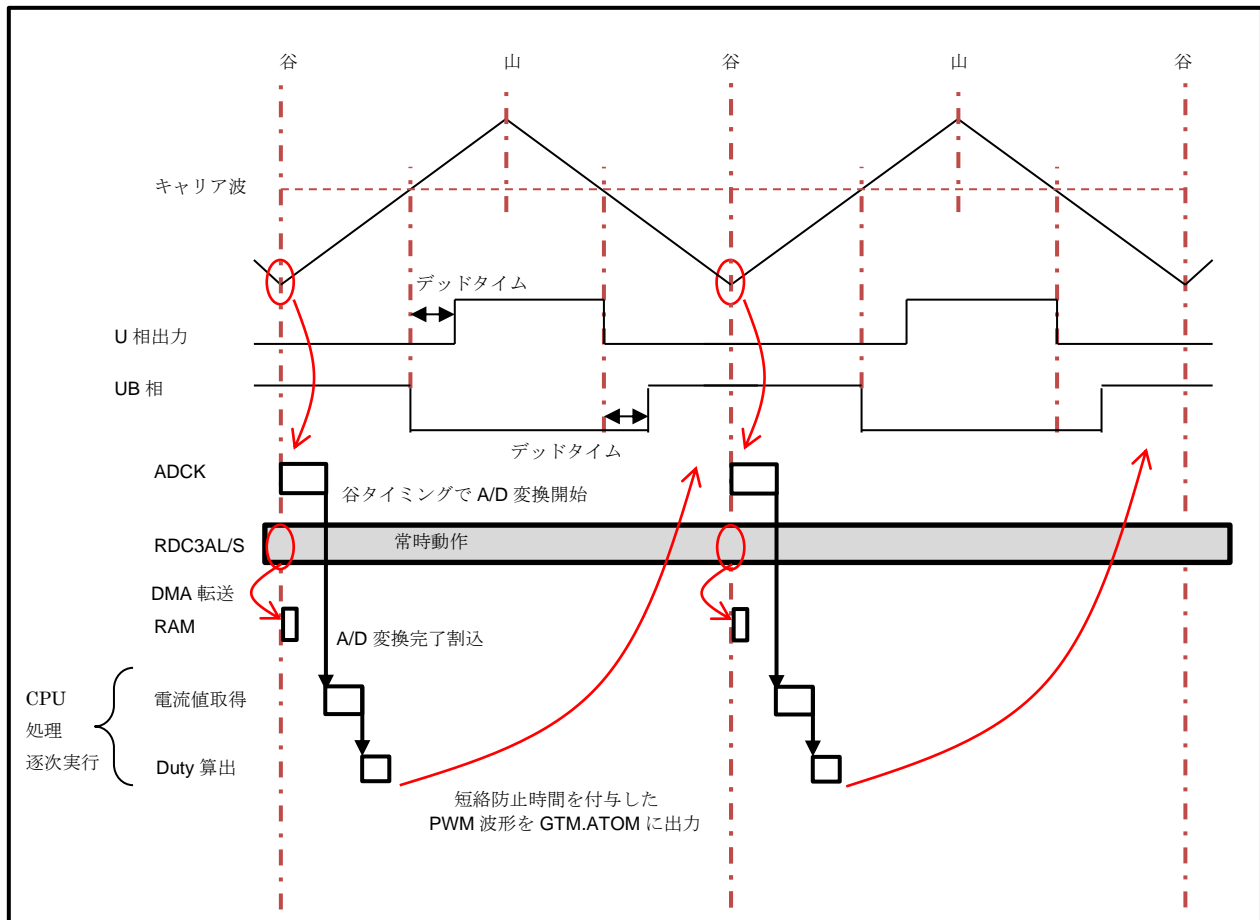


図 2.2 制御フロー図

3. 動作条件

本アプリケーションノートのサンプルコードは、下記の条件で動作を確認しています。

表 3.1 動作確認条件

項目	内容
使用マイコン	RH850/U2B6
動作周波数	<ul style="list-style-type: none">・メイン OSC: 40MHz・PLL0: 400MHz・PLL1: 80MHz・CPU クロック: 400MHz・GTM-ATOM 動作クロック : 80MHz
動作電圧	<ul style="list-style-type: none">・VDD=1.25V・PVCC=5.0V
統合開発環境	ルネサスエレクトロニクス製 CS+ for CC E8.07.00g6
C コンパイラ	ルネサスエレクトロニクス製 RH850 ファミリ用 C コンパイラパッケージ V2.02.00 コンパイルオプション 統合開発環境のデフォルト設定
サンプルコードのバージョン	Version 1.00

4. 使用端子と機能

表 4.1 に使用端子と機能を示します。

表 4.1 使用端子と機能

モジュール	端子名	入出力	機能
ADCK	ADCK0I00	入力	V 相電流の測定
	ADCK0I01	入力	W 相電流の測定
	ADCK0I02	入力	U 相電流の測定
RDC3AL	RDC3AL0S1	入力	レゾルバ信号入力($\cos \theta$)
	RDC3AL0S3	入力	レゾルバ信号入力($\cos \theta$)
	RDC3AL0S2	入力	レゾルバ信号入力($\sin \theta$)
	RDC3AL0S4	入力	レゾルバ信号入力($\sin \theta$)
	RDC3AL0RSO	出力	励磁信号出力
	RDC3AL0COM	出力	励磁信号用コモン電圧出力
PORT	P12_0: ATOM0_CH0_OUT	出力	U 相正相 PWM 出力
	P12_1: ATOM0_CH1_OUT	出力	V 相正相 PWM 出力
	P12_2: ATOM0_CH2_OUT	出力	W 相正相 PWM 出力
	P12_4: ATOM0_CH0_OUT_N	出力	U 相逆相 PWM 出力
	P12_5: ATOM0_CH1_OUT_N	出力	V 相逆相 PWM 出力
	P12_6: ATOM0_CH2_OUT_N	出力	W 相逆相 PWM 出力

表 4.2 PORT 設定

レジスタ名	設定値	機能
PCR12_0	0x0000004D	ATOM0ch0 出力
PCR12_1	0x0000004D	ATOM0ch1 出力
PCR12_2	0x0000004D	ATOM0ch2 出力
PCR12_4	0x0000004D	ATOM0ch0_N 出力
PCR12_5	0x0000004D	ATOM0ch1_N 出力
PCR12_6	0x0000004D	ATOM0ch2_N 出力

表 4.3 INTC2 設定

レジスタ名	設定値	機能
EIC70	0x004F	INTSDMAC0CH0 sDMAC 転送終了割り込み
EIC86	0x004F	INTGTM0IOS000 GTM 共有割り込み
EIC445	0x004F	INTADCK0I4 SG4 A/D 変換完了割り込み

5. ハードウェア動作

5.1 動作概要

モータ制御の動作を説明します。

- (1) キャリア波の谷タイミングで GTM から PIC 経由で A/D コンバータにトリガ信号が入力されると、ADCK0 グループ 4 の仮想チャンネル 0~3 を A/D 変換し、モータの U 相と V 相、W 相の電流値を取得します。同時に、sDMAC が RDC3AL で変換された角度を RAM へ転送します。
- (2) A/D 変換が完了すると、A/D 変換完了割り込み関数が呼び出され、PWM コンペア値を算出します。
- (3) PWM コンペア値は、GTM ATOM のシャドウレジスタに格納され、次のリロードタイミングで GTM ATOM のコンペアレジスタに設定されます。そして対応する GTM ATOM 端子から三相 PWM 波形が出力されます。

5.2 R/D コンバータ 3AL (RDC3AL)

レゾルバから出力されるアナログ信号をデジタル変換し、角度値(レゾルバ角)、角速度を算出します。

5.3 sDMA Controller (sDMAC)

GTM からキャリア波に同期して出力される信号をもとに、RDC3AL からの角度値を RAM に転送します。

5.4 ペリフェラルインタコネクション (PIC)

GTM からキャリア波に同期して出力される信号をもとに、ADCK ハードウェアトリガ信号を生成します。

5.5 A/D コンバータ(ADCK)

モータ電流値を A/D 変換します。本アプリケーションノートでは、GTM からの出力信号に応じて PIC が生成するトリガにより A/D 変換を開始させます。トリガが入力されると、ADCK0I00 で V 相の電流値を、ADCK0I01 で W 相の電流値を、ADCK0I02 で U 相の電流値をそれぞれ変換します。W 相の電流値は処理では使用しません。

5.6 汎用タイマーモジュール (GTM)

GTM は割込み関数にて転送されたコンペア値をもとに三相 PWM 波形を出力します。

GTM は 24 ビットベースカウンタ(CHx_CN)が 0 から CHx_CM0 の間で遷移し、24 ビットコンペアレジスタ(CHx_CM1)とのコンペア一致からデューティを生成し、短絡防止時間を付加した三相 PWM 波形を出力します。図 5.1 に三相 PWM 波形タイミング図を示します。

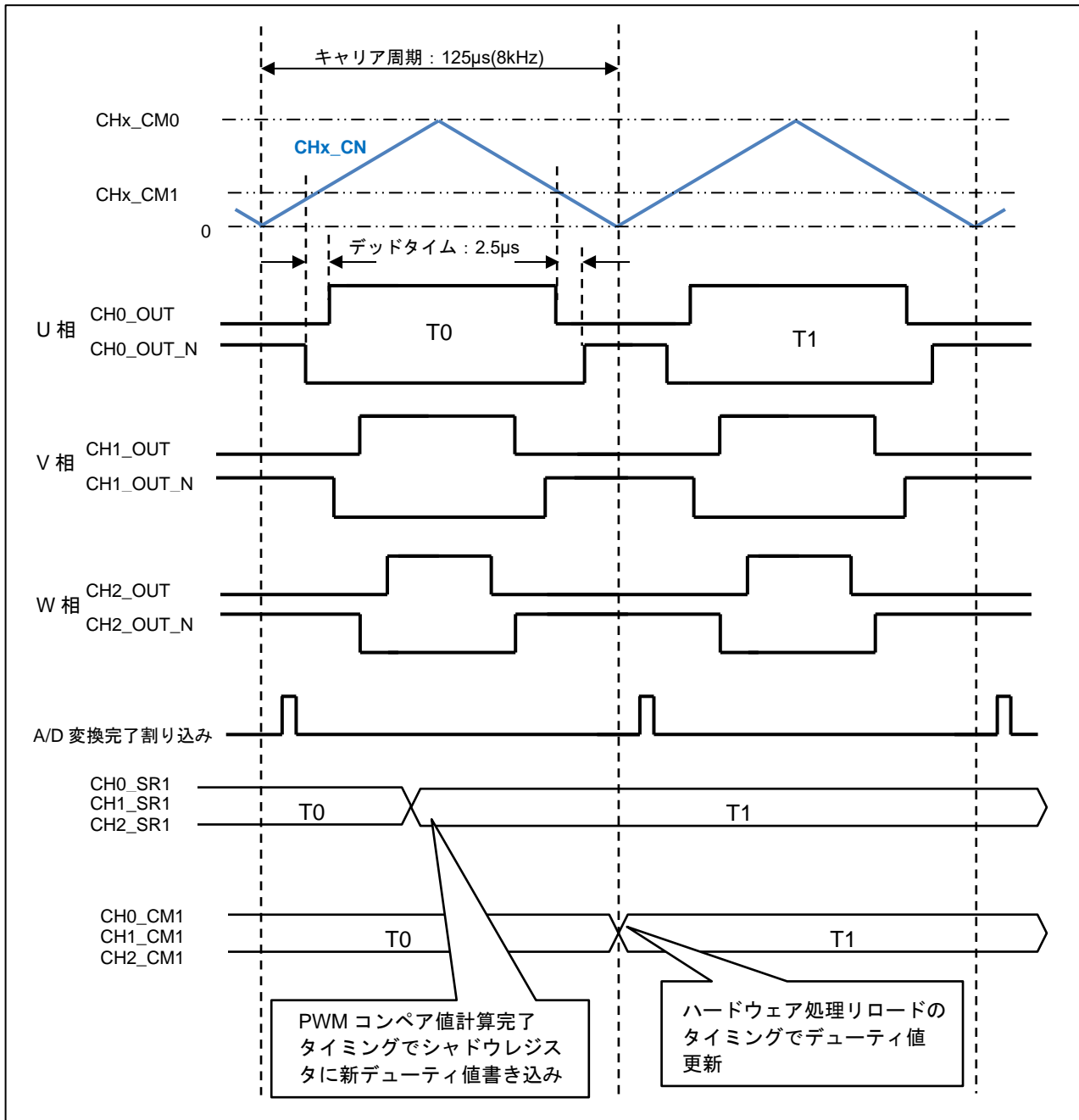


図 5.1 三相 PWM 波形タイミング図

表 5.1 GTM モジュール設定

項目	内容
メインクロック	160MHz
使用クラスタ	クラスタ 0
メインクロック分周	2 分周
割り込み制御	ATOM 周期割り込みを許可

表 5.2 GTM 共通設定

レジスタ名	設定値	機能
GTM_CTRL.RF_PROT	0	RST および FORCINT の保護
GTM_CLS_CLK_CFG	0x0000002A	各クラスタに 2 分周のクロックを供給
GTM_IRQ_SEL000	0x00010000	ATOM0ch0 の割り込みを INTC に転送する
GTM_DMA_SEL00	0x00000024	GTM 出力から DMA トリガーを選択
TBU_CHEN	0x00000056	TBU チャンネル ch0 の有効化

表 5.3 CMU 設定

レジスタ名	設定値	機能
CMU_CLK_EN	0x0000002A	ch0-ch2 まで有効化
CMU_GCLK_NUM	0x00000001	デフォルト値。分周を変更しない。
CMU_GCLK_DEN	0x00000001	デフォルト値。分周を変更しない。
CMU_CLK_0_CTRL	0x00000000	デフォルト値。分周を変更しない。
CMU_CLK_1_CTRL	0x00000000	デフォルト値。分周を変更しない。
CMU_CLK_2_CTRL	0x00000000	デフォルト値。分周を変更しない。
CMU_CLK_CTRL	0x00000000	デフォルト値。

表 5.4 ATOM0 ユニット共通設定

レジスタ名	設定値	機能
ATOM0_AGC_GLB_CTRL	0x00150000	ch0-ch2 まで CM0, CM1, SL, CLK_SRC のリロードを有効化
ATOM0_AGC_ENDIS_CTRL	0x0000002A	ch0-ch2 までチャンネルを有効化
ATOM0_AGC_ENDIS_STAT	0x0000002A	ch0-ch2 までチャンネルを有効化
ATOM0_AGC_OUTEN_CTRL	0x0000002A	ch0-ch2 までチャンネル出力を有効化
ATOM0_AGC_OUTEN_STAT	0x0000002A	ch0-ch2 まで出力を有効化
ATOM0_AGC_FUPD_CTRL	0x0015002A	ch0-ch2 まで CN0 の強制アップデートを有効化
ATOM0_AGC_INT_TRIG	0x0000002A	ch0-ch2 まで割り込みトリガを有効化

表 5.5 ATOM0 CH0-CH2 設定

レジスタ名	設定値	機能
ATOM0_CH0_CTRL	0x01040002	SOMP モードを選択 初期信号レベルは High を選択 リセットトリガに CM0 コンペア一致を選択 トリガ出力に TRIG_CCU0 を選択
ATOM0_CH0_CN0	0x00000000	0 からカウントを開始する
ATOM0_CH0_CM0	0x00001388	CN0 は 1 ずつアップダウンするため、目標周期の半分の値を設定する
ATOM0_CH0_SR0	0x00001388	
ATOM0_CH0_CM1	0x00000001	出力用のコンペア値。初回コンペア一致のため、0 ではない値を設定しておく
ATOM0_CH0_SR1	0x00000001	
ATOM0_CH0_IRQ_NOTIFY	0x00000003	割り込み要因をクリア
ATOM0_CH0_IRQ_EN	0x00000003	割り込みを許可

表 5.6 DTM 設定

レジスタ名	設定値	機能
CDTM0_DTM4_CTRL	0x00000001	CTRL2 の更新はおこなわない Clock は CMU_CLK0 を選択
CDTM0_DTM4_CH_CTRL1	0x00000000	DTM_IN_T を DTM_OUT_N にルーティングしない
CDTM0_DTM4_CH_CTRL2	0x00888888	ch0-ch2 までデッドタイム機能を使用する
CDTM0_DTM4_CH_CTRL3	0x00000000	デフォルト値
CDTM0_DTM4_CH0_DTV	0x00C800C8	両エッジに 2.5us のデッドタイムを設定
CDTM0_DTM4_CH1_DTV	0x00C800C8	両エッジに 2.5us のデッドタイムを設定
CDTM0_DTM4_CH2_DTV	0x00C800C8	両エッジに 2.5us のデッドタイムを設定
CDTM0_DTM4_CH0_DTV_SR	0xC0C8C0C8	両エッジに 2.5us のリロードを設定
CDTM0_DTM4_CH1_DTV_SR	0xC0C8C0C8	両エッジに 2.5us のリロードを設定
CDTM0_DTM4_CH2_DTV_SR	0xC0C8C0C8	両エッジに 2.5us のリロードを設定

5.7 CPU 割込み処理 PWM コンペア値生成

本アプリケーションノートでは、CPU 割込み処理内にて PWM コンペア値を生成します。

5.8 ファイル構成

表 5.7 にサンプルコードで使用するファイルを示します。なお、統合開発環境で自動生成されるファイルは除きます。本サンプルプログラムはルネサス評価基板での動作を確認しております。

動作確認のための処理が含まれています。

表 5.7 ファイル構成

ファイル名	概要	備考
main0.c	CPU0 メインモジュール	CPU1: main1.c, CPU2: main2.c CPU1,CPU2 メインは無限ループ
userint.c	割込み処理モジュール	-
rdc3al.c	R/D コンバータ 3A 設定モジュール	-
pic.c	PIC 設定モジュール	-
adck.c	A/D コンバータ設定モジュール	-
gtm.c	GTM 設定モジュール	-
emu3.c	PWM コンペア値生成処理モジュール	-
port.c	端子設定モジュール	-
sdmac.c	sDMAC 設定モジュール	-
emu3.h	PWM コンペア値生成処理用ヘッダファイル	-
sdmac.h	sDMAC 設定モジュール用ヘッダファイル	-
iodef.h	レジスタのヘッダファイル	-
typedefs.h	型定義ヘッダファイル	-

5.9 関数一覧

表 5.8 に関数を示します。

表 5.8 関数

関数名	概要
rdc3al_init	R/D コンバータ 3A の初期化
pic_init	PIC の初期化
adck_init	A/D コンバータの初期化
gtm_init	GTM の初期化
gtm_atom_enable	GTM ATOM 動作許可
int_init	割り込み設定
int_ADCK0_VCH0_irq	ADCK SG4 A/D 変換完了割り込み処理
int_GTM0_ATOM_CH0_irq	GTM 共有割り込み処理
int_SDMAC0_CH0_irq	sDMAC 転送終了割り込み処理
port_init	端子の初期化
sdmac0_ch0_init	sDMAC の初期化
sdmac0_ch0_start	sDMAC 転送許可

5.10 関数仕様

サンプルコードの関数仕様を示します。

rdc3al_init

概要	R/D コンバータ 3AL の初期化
ヘッダ	iodefine.h
宣言	void rdc3al_init (void)
説明	R/D コンバータ 3AL の初期設定をおこないます。
引数	なし
リターン値	なし
備考	なし

pic_init

概要	PIC の初期化
ヘッダ	iodefine.h
宣言	void pic_init (void)
説明	PIC の初期設定をおこないます。
引数	なし
リターン値	なし
備考	なし

adck_init

概要	A/D コンバータの初期化
ヘッダ	iodefine.h
宣言	void adck_init (void)
説明	A/D コンバータの初期設定をおこないます。
引数	なし
リターン値	なし
備考	なし

gtm_init

概 要	GTM の初期化
ヘッダ	iodefine.h
宣 言	void gtm_init (void)
説 明	GTM の初期設定をおこないます
引 数	なし
リターン値	なし
備 考	なし

gtm_atom_enable

概 要	GTM ATOM 動作許可
ヘッダ	iodefine.h
宣 言	void gtm_atom_enable
説 明	GTM ATOM の動作を許可します
引 数	なし
リターン値	なし
備 考	なし

int_init

概 要	割込み設定の初期化
ヘッダ	iodefine.h
宣 言	void int_init (void)
説 明	割込み設定の初期化をおこないます
引 数	なし
リターン値	なし
備 考	なし

`int_ADCK0_VCH0_irq`

概要	ADCK SG4 の A/D 変換完了割り込み処理
ヘッダ	<code>iodefine.h</code> , <code>emu3.h</code>
宣言	<code>void int_ADCK0_VCH0_irq (void)</code>
説明	ADCK SG4 の A/D 変換完了割り込み処理をおこないません
引数	なし
リターン値	なし
備考	なし

`int_GTM0_ATOM_CH0_irq`

概要	GTM 共有割り込み処理
ヘッダ	<code>iodefine.h</code>
宣言	<code>void int_GTM0_ATOM_CH0_irq (void)</code>
説明	GTM 共有割り込み割り込み処理をおこないません
引数	なし
リターン値	なし
備考	なし

`int_SDMAC0_CH0_irq`

概要	sDMAC 転送終了割り込み処理
ヘッダ	<code>iodefine.h</code>
宣言	<code>void int_SDMAC0_CH0_irq (void)</code>
説明	sDMAC 転送終了割り込み処理をおこないません
引数	なし
リターン値	なし
備考	なし

port_init

概 要	端子の初期化
ヘッダ	iodefine.h
宣 言	void port_init (void)
説 明	端子の初期設定をおこないます
引 数	なし
リターン値	なし
備 考	なし

sdmac0_ch0_init

概 要	sDMAC の初期化
ヘッダ	iodefine.h, sdmac.h, emu3.h
宣 言	void sdmac0_ch0_init (void)
説 明	sDMAC の初期設定をおこないます
引 数	なし
リターン値	なし
備 考	なし

sdmac0_ch0_start

概 要	sDMAC 転送許可
ヘッダ	iodefine.h, sdmac.h
宣 言	void sdmac0_ch0_start (void)
説 明	sDMAC の転送を許可します
引 数	なし
リターン値	なし
備 考	なし

5.11 フローチャート

図 5.2 にメイン処理、図 5.3 に RDC3AL の初期設定、図 5.4 に PIC の初期設定、図 5.5 に ADCK の初期設定、図 5.6 に GTM の初期設定、図 5.7 に端子の初期設定、図 5.15 に sDMAC の初期設定のフローチャートを示します。

各関数には必要に応じてクロック供給設定、レジスタ同期処理が挿入されていますが、フローチャートでは省略されています。



図 5.2 CPU0 メイン処理

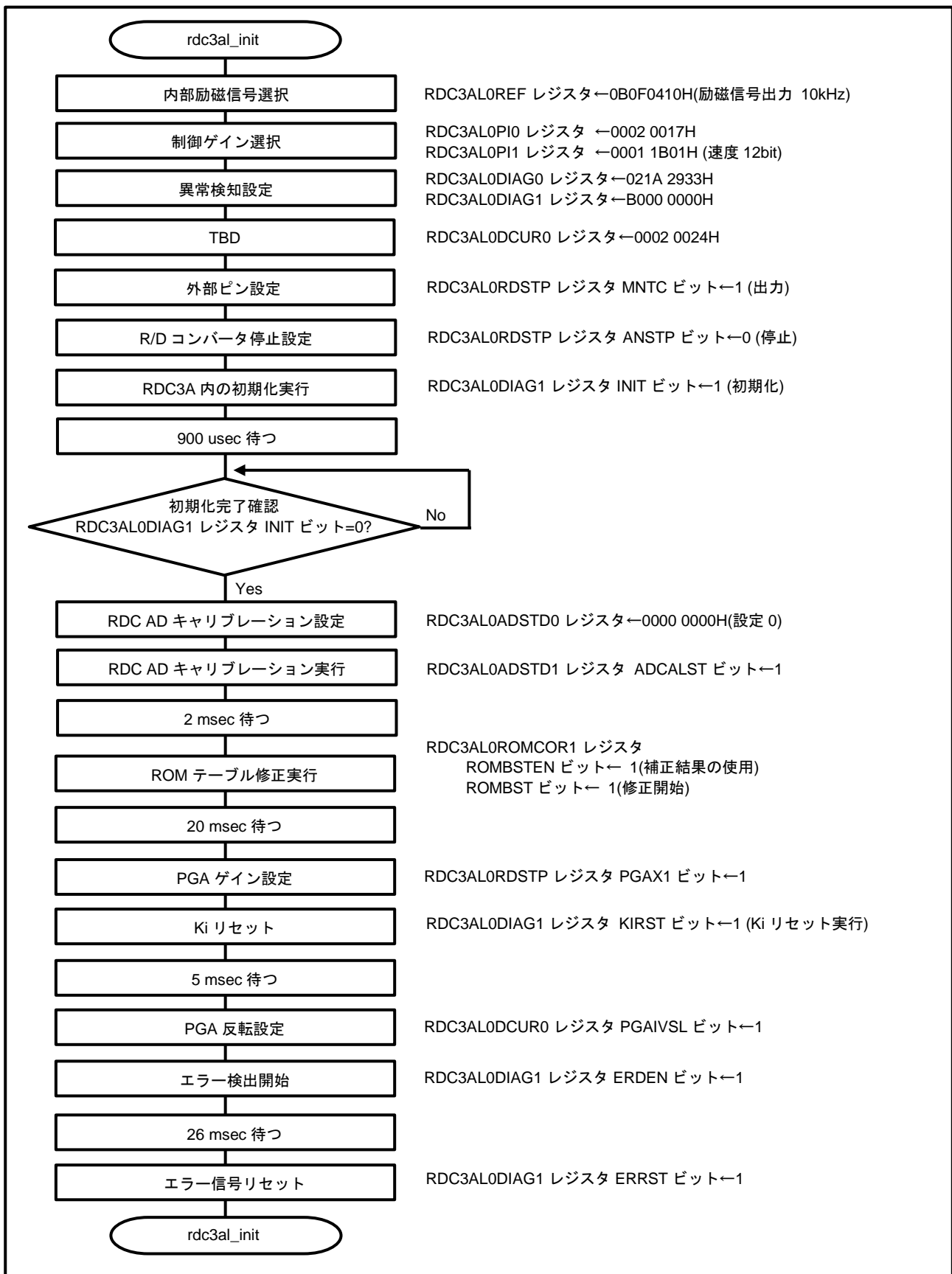


図 5.3 RDC3AL の初期設定

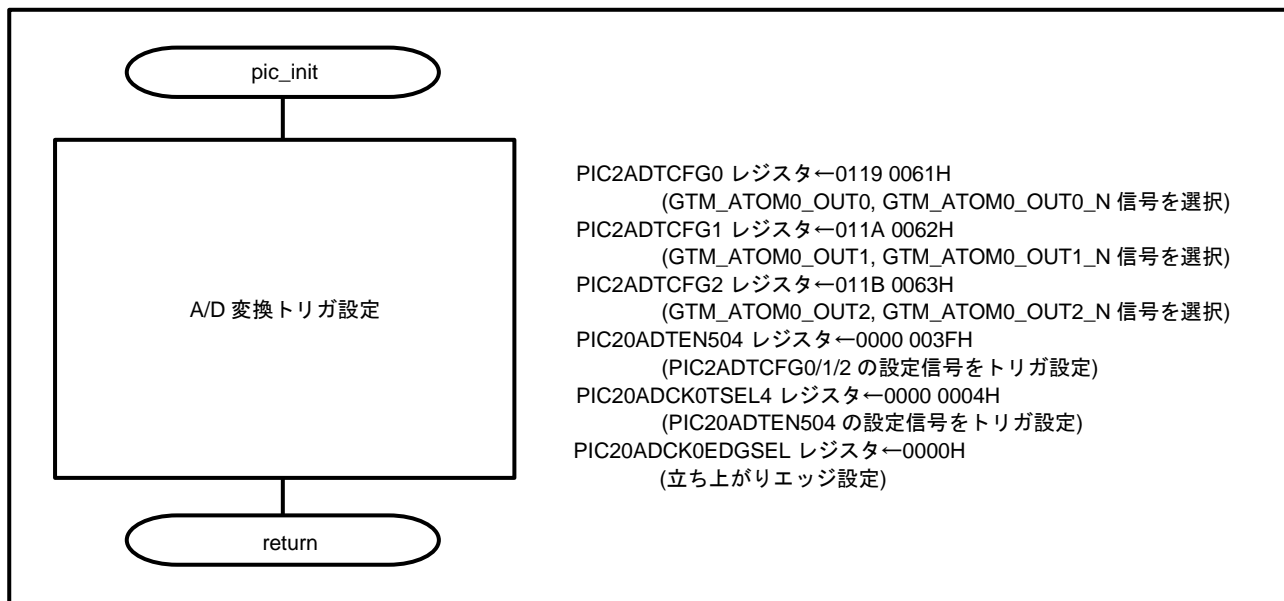


図 5.4 PIC の初期設定

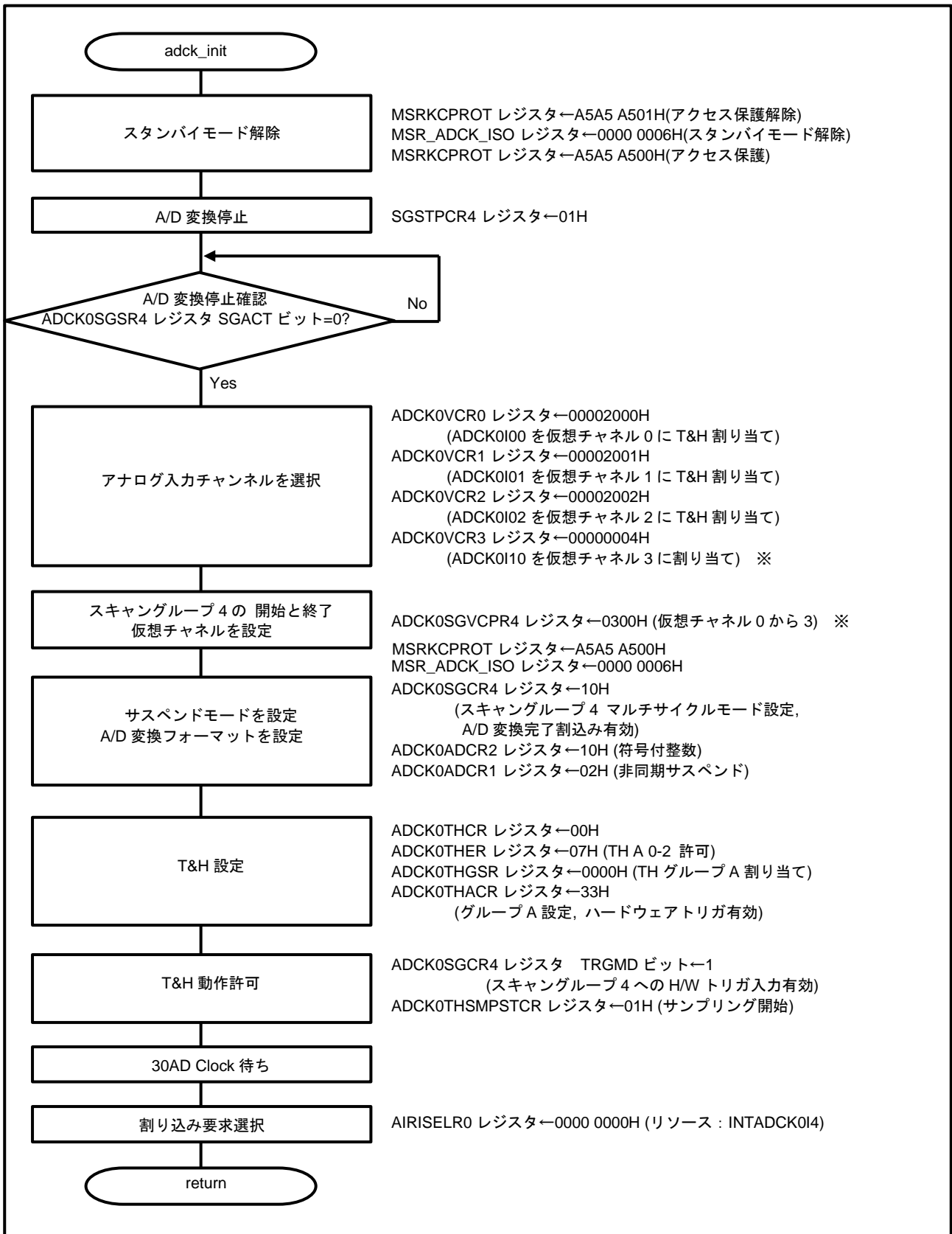


図 5.5 ADCKの初期設定

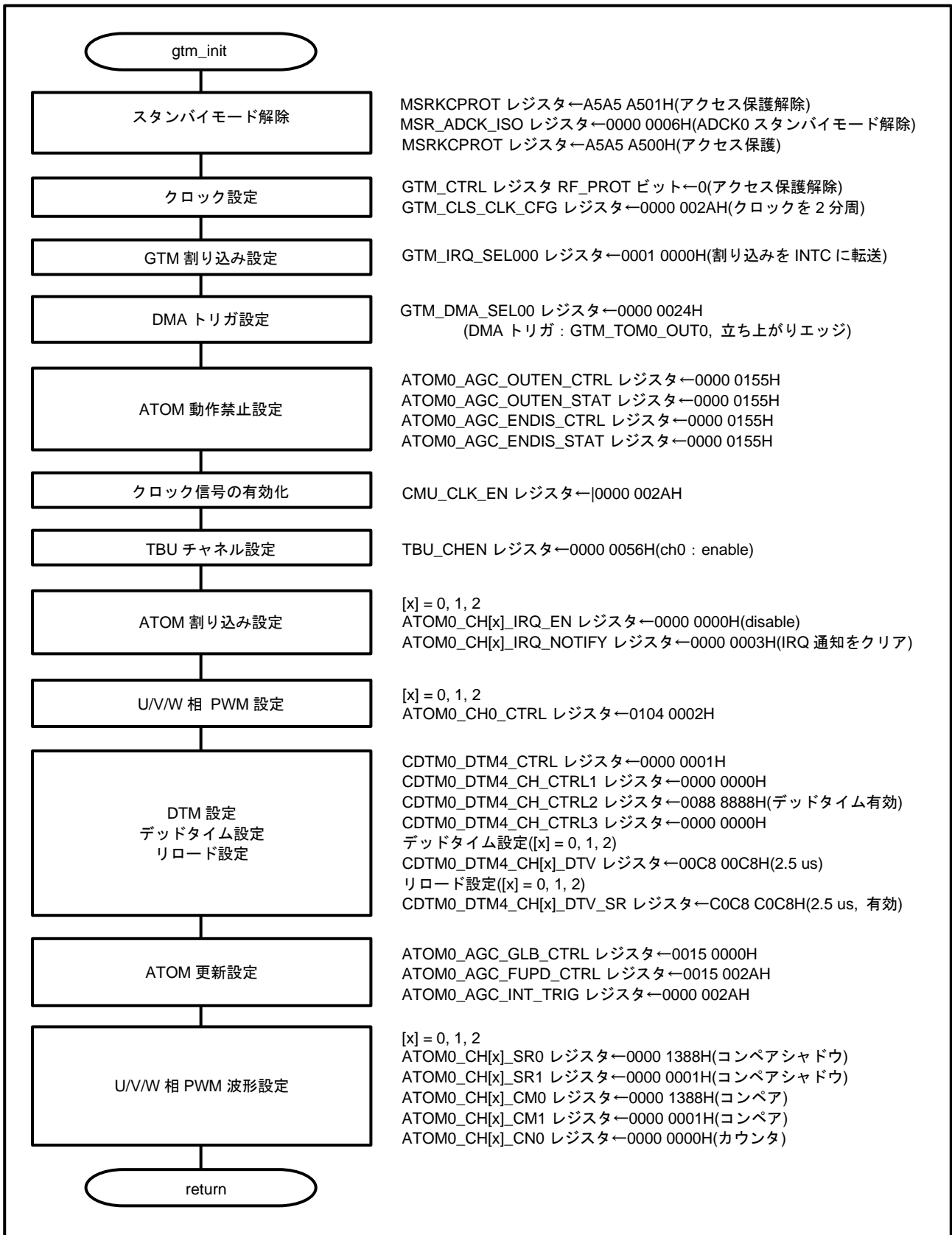


図 5.6 GTM の初期設定

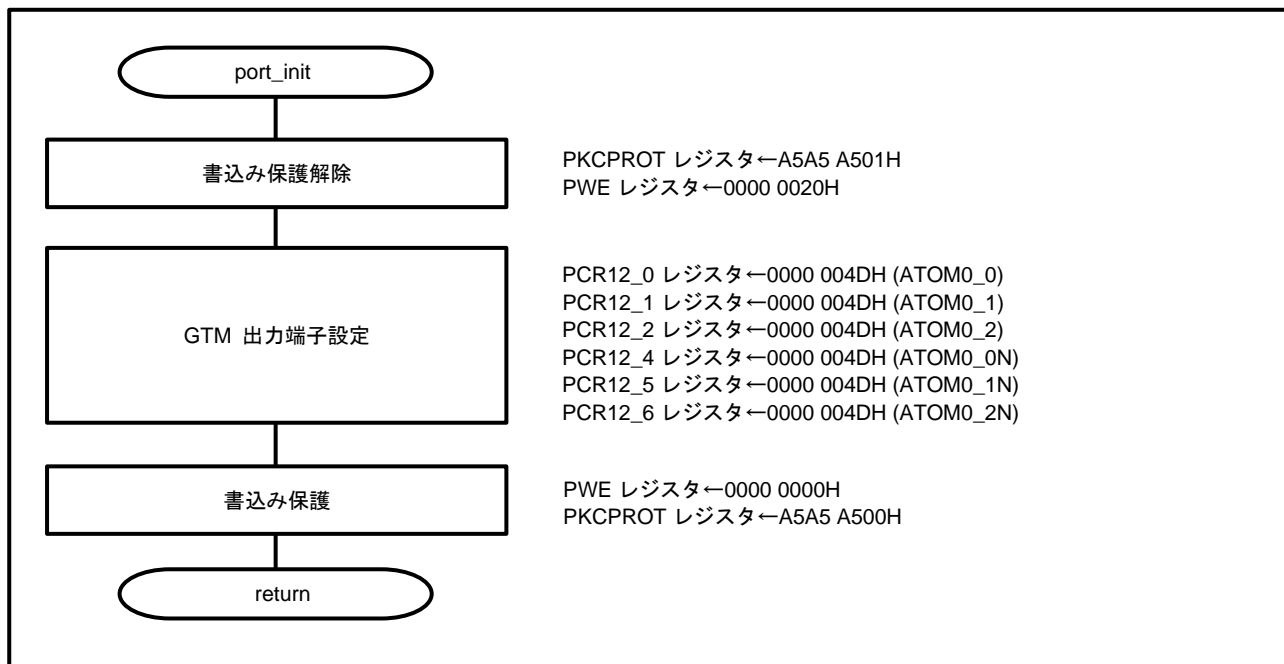


図 5.7 端子の設定

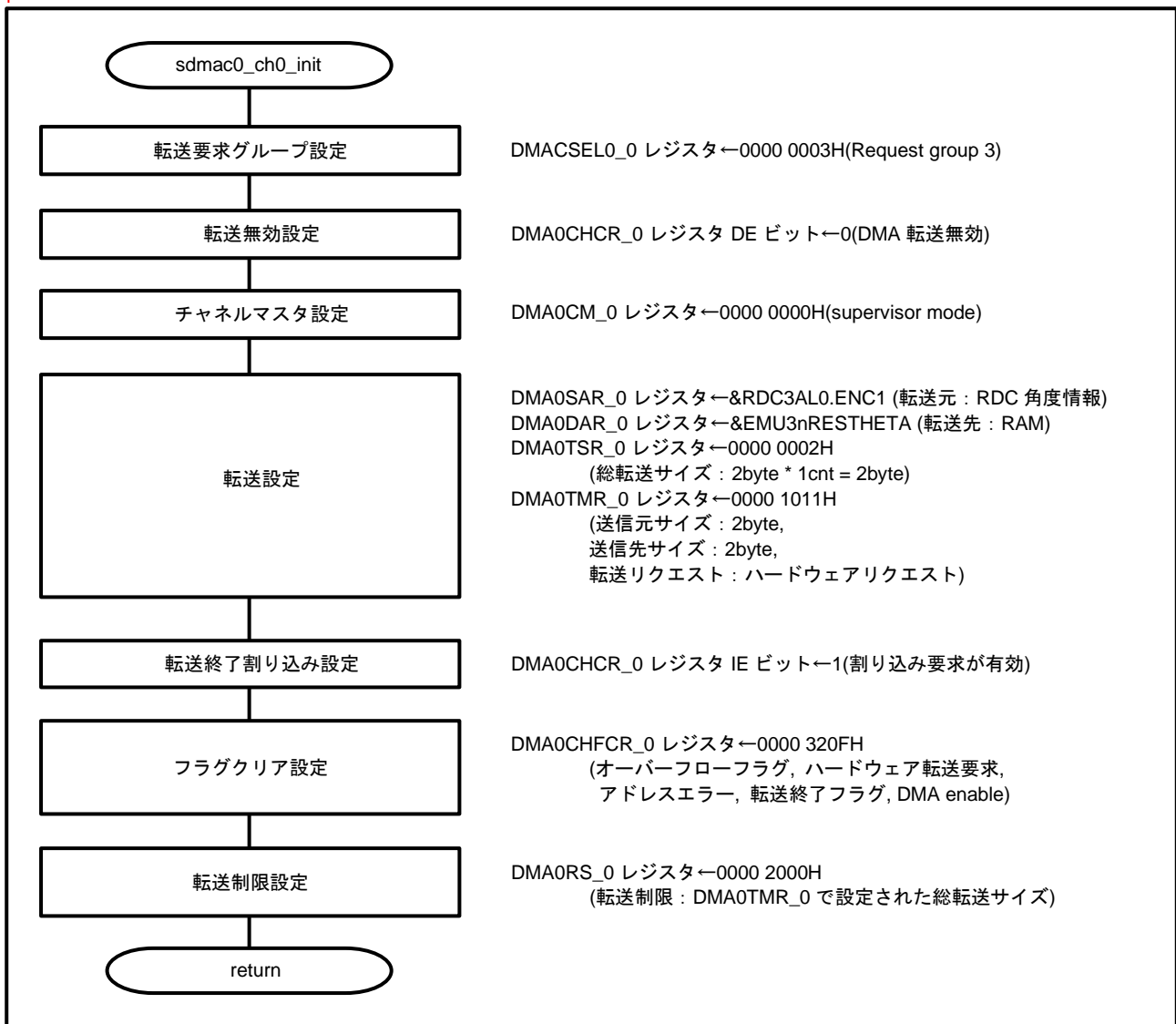


図 5.8 sDMAC の設定

6. 参考ドキュメント

C コンパイラマニュアル

CS+ V4.00.00 統合開発環境ユーザズマニュアル RH850 コーディング編

CS+ V4.00.00 統合開発環境ユーザズマニュアル RH850 ビルド編

RH850/ U2B Group ユーザズマニュアル ハードウェア編 Rev.0.50

RH850/U2B Group Resolver to Digital Converter ユーザズマニュアル ハードウェア編 Rev.0.50

GTM-IP Specification v4.1-draft

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2022.06.24	-	初版発行

製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本ドキュメントおよびテクニカルアップデートを参照してください。

1. 静電気対策

CMOS製品の取り扱いの際は静電気防止を心がけてください。CMOS製品は強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、当社が出荷梱包に使用している導電性のトレーやマガジンケース、導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを施してください。プラスチック板上に放置したり、端子を触ったりしないでください。また、CMOS製品を実装したボードについても同様の扱いをしてください。

2. 電源投入時の処置

電源投入時は、製品の状態は不定です。電源投入時には、LSIの内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。

3. 電源オフ時における入力信号

当該製品の電源がオフ状態のときに、入力信号や入出力プルアップ電源を入れしないでください。入力信号や入出力プルアップ電源からの電流注入により、誤動作を引き起こしたり、異常電流が流れ内部素子を劣化させたりする場合があります。資料中に「電源オフ時における入力信号」についての記載のある製品は、その内容を守ってください。

4. 未使用端子の処理

未使用端子は、「未使用端子の処理」に従って処理してください。CMOS製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI周辺のノイズが印加され、LSI内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。

5. クロックについて

リセット時は、クロックが安定した後、リセットを解除してください。プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後に切り替えてください。リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。

6. 入力端子の印加波形

入力ノイズや反射波による波形歪みは誤動作の原因になりますので注意してください。CMOS製品の入力がノイズなどに起因して、 $V_{IL}(\text{Max.})$ から $V_{IH}(\text{Min.})$ までの領域にとどまるような場合は、誤動作を引き起こす恐れがあります。入力レベルが固定の場合はもちろん、 $V_{IL}(\text{Max.})$ から $V_{IH}(\text{Min.})$ までの領域を通過する遷移期間中にチャタリングノイズなどが入らないように使用してください。

7. リザーブアドレス（予約領域）のアクセス禁止

リザーブアドレス（予約領域）のアクセスを禁止します。アドレス領域には、将来の拡張機能用に割り付けられている リザーブアドレス（予約領域）があります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

8. 製品間の相違について

型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。同じグループのマイコンでも型名が違えば、フラッシュメモリ、レイアウトパターンの相違などにより、電気的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ輻射量などが異なる場合があります。型名が違う製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

ご注意書き

1. 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。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、お客様の責任において、お客様の機器・システムを設計ください。これらの使用に起因して生じた損害（お客様または第三者いずれに生じた損害も含まれます。以下同じです。）に関し、当社は、一切その責任を負いません。
 2. 当社製品または本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
 3. 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。
 4. 当社製品を組み込んだ製品の輸出入、製造、販売、利用、配布その他の行為を行うにあたり、第三者保有の技術の利用に関するライセンスが必要となる場合、当該ライセンス取得の判断および取得はお客様の責任において行ってください。
 5. 当社製品を、全部または一部を問わず、改造、変更、複製、リバースエンジニアリング、その他、不適切に使用しないでください。かかる改造、変更、複製、リバースエンジニアリング等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。
 6. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。
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、産業用ロボット等
高品質水準： 輸送機器（自動車、電車、船舶等）、交通管制（信号）、大規模通信機器、金融端末基幹システム、各種安全制御装置等
当社製品は、データシート等により高信頼性、Harsh environment 向け製品と定義しているものを除き、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じても、当社は一切その責任を負いません。
 7. あらゆる半導体製品は、外部攻撃からの安全性を 100%保証されているわけではありません。当社ハードウェア/ソフトウェア製品にはセキュリティ対策が組み込まれているものもありますが、これによって、当社は、セキュリティ脆弱性または侵害（当社製品または当社製品が使用されているシステムに対する不正アクセス・不正使用を含みますが、これに限られません。）から生じる責任を負うものではありません。当社は、当社製品または当社製品が使用されたあらゆるシステムが、不正な改変、攻撃、ウイルス、干渉、ハッキング、データの破壊または窃盗その他の不正な侵入行為（「脆弱性問題」といいます。）によって影響を受けないことを保証しません。当社は、脆弱性問題に起因したまたはこれに関連して生じた損害について、一切責任を負いません。また、法令において認められる限りにおいて、本資料および当社ハードウェア/ソフトウェア製品について、商品性および特定目的との合致に関する保証ならびに第三者の権利を侵害しないことの保証を含め、明示または黙示のいかなる保証も行いません。
 8. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合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いますが、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、データシート等において高信頼性、Harsh environment 向け製品と定義しているものを除き、耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他社会的損害等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、お客様の機器・システムとしての出荷保証を行ってください。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、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
 10. 当社製品の環境適合性等の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。かかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
 11. 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・システムに使用することはできません。当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
 12. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものとなります。
 13. 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禁じます。
 14. 本資料に記載されている内容または当社製品についてご不明な点がございましたら、当社の営業担当者までお問合せください。
- 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直接的、間接的に支配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。

(Rev.5.0-1 2020.10)

本社所在地

〒135-0061 東京都江東区豊洲 3-2-24（豊洲フォレシア）

www.renesas.com

お問合せ窓口

弊社の製品や技術、ドキュメントの最新情報、最寄の営業お問合せ窓口に関する情報などは、弊社ウェブサイトをご覧ください。

www.renesas.com/contact/

商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。すべての商標および登録商標は、それぞれの所有者に帰属します。

