RX62T
IR 補正によるブラシ付き DC モータの制御

R01AN0958JU0100
Rev.1.00
2013.02.08

はじめに

多くのアプリケーションでは、物や人などを移動するために、両方向に回転する単純なモータが必要とされています。多くの場合、最も単純なモータはブラシ付き DC モータになります。モータを MOSFET による H ブリッジ回路および PWM タイマを搭載したマイクロコントローラと組み合わせることで、両方向回転の単純なモータ制御が実現します。さらに、DC モータの数学的なモデルは非常に単純なものであるため、制限のある CPU 処理能力で簡単な IR 補正方式の速度制御を追加することが出来ます。この方法はモータを流れる電流を監視し、IR による電圧低下分をモータ駆動電圧（PWM）に加えるという調節を行うものです。

対象デバイス

RX62T

注意: このデモンストレーションは RX62T のためのものですが、この制御方式とコードの大部分は PWM タイマと ADC を搭載する他のルネサス MCU にも容易に移植することができます。

目次

1. 概要

このアプリケーションノートでは MCU のタイマによる PWM を使用しブラシ付き DC モータの速度と回転方向をどのように制御するかを示しており、同時に IR 補正と呼ばれる基本的な速度制御についても説明してあります。IR 補正は基本的には正帰還で、モータ電流の増加として表れる負荷の増加に応じてモータ電圧を高めるというものです。

2. ブラシ付き DC モータの制御

2.1 モータのモデル

このドキュメントは DC モータの理論を論じることを目的とはしていませんが、制御方法を知る上で必要なブラシ付き DC モータとそのモデルの基本について説明しています。ドキュメントの最後に参考資料を記載してありますので、より深く知ろうとお考えの方は、そちらをご覧ください。

図 1 は永久磁石固定子、巻線を巻いた回転子、およびブラシ（整流子）で構成される典型的な DC モータの図です。

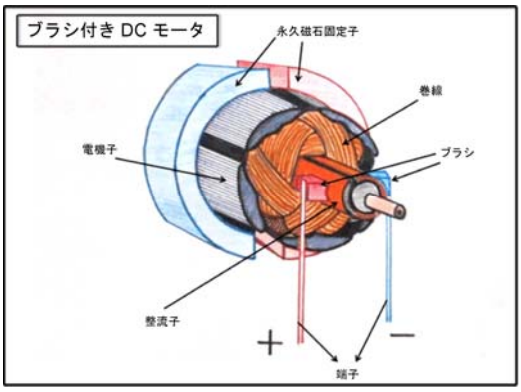

図 1 典型的なブラシ付き DC モータ

端子間には電圧が付加され、電流（I）が回転子の巻線を流れて磁束が発生します。ブラシは電流による磁束が永久磁石固定子の磁束と適切な角度を持つことで回転子を回転させるトルクを生じるように配置されています。固定子の磁束に対して適切な角度の磁束が維持され回転子を回転させるトルクが連続的に発生するよう、所定のタイミングでブラシが回転子の電流方向を切り替えます。

モータが回転するにつれて固定子の磁束が回転子の巻線を横切るため、巻線には起電力（emf）が発生します。この起電力はモータに付加された電圧によって生じるトルクと逆方向のトルクを生じさせます。モータをより速く回転させると、これに抗する起電力も一層大きくなります。これを次に詳細に検討します。

図 2 はブラシ付き DC モータの基本モデルを示しています。

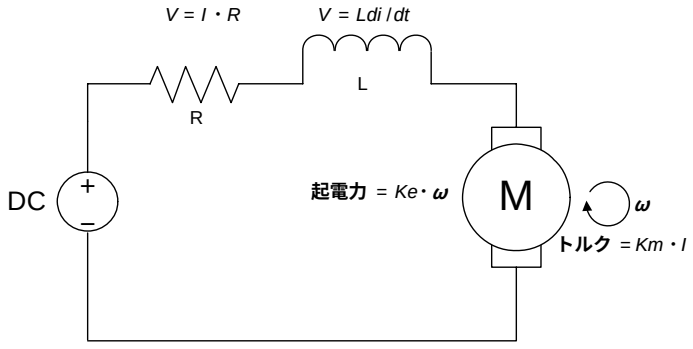

図 2 ブラシ付き DC モータの基本的なモデル

この図における基本的な用語の意味は以下の通りです。

Km はモータのトルク定数で、通常ニュートンメートル（Nm）で表されます。発生するトルクはこの定数の関数としてモータに供給される電流(I)に比例しています。

Ke はモータの誘導起電力定数です。起電力（電圧）はモータが回転することによって発生し、Ke の関数としての回転速度に比例します。

R はモータの抵抗値で、その大部分は巻線の抵抗ですがブラシにおける抵抗も含まれます。

トルク T=Km･I、起電力 emf=Ke･ωです。簡単な解析の範囲では、駆動されるトルクが emf によるトルクと一致するまでモータの回転数は上昇します。したがって、この解析では Km=Ke と設定することが出来ます。キルヒホッフの法則により、モータのモデルは次の式で表現できます。

$$L\frac{di}{dt} + RI = VDC - Ke \cdot \omega$$

ここでは一定の速度ωで回転している定常状態のモータを考えます。このとき、時間に依存する項 Ldi/dt を無視することが出来ますので、式は次のようになります。

$$RI = VDC - Ke \cdot \omega$$

これから次の式が得られます。

$$Ke \cdot \omega = VDC - RI$$

式 1　回転速度の関数としてのモータ端子電圧

特定の回転速度が必要な場合､希望の基準速度に Ke を乗じた値により H ブリッジ PWM が発生すべき電圧が決まります。ここで電流 I が一定ではないという点が問題となります。電流は負荷の増減に応じて変化しますので、ωを一定に保つには電流値を測り I･R による低下分を加減する必要が生じます。言い方を変えれば、I･R による電圧低下を補正するということになります。Ke=Km ですので、トルクを一定にするにはモータを流れる電流を制御しなければなりません。R による電圧降下が大きくなるほど駆動電圧を大きくしなければなりません。

式 1 によれば「モータを外から回転させると発電機になる」とも言うことが出来ます。電流 I が減少もしくは最終的に逆転するとき、I･R 項は次第に低下し最後にはマイナスとなります。相補モードでブリッジを駆動しているときには、モータに電圧を付加している時間が短くなり、電源からモータを遮断している時間が長くなります。この場合にも、回転速度の制御に関して I･R 補正が正しく働きます。駆動電流と還流電流に関しては図 6 と図 7 に示されています。

最後に、モータのストールについても検討しなければなりません。モータがストールするとモータが耐えられる値を超えた電流が流れ、モータに障害が発生します（ω=0 のとき emf は発生せず、一般に非常に小さい値の R によってのみ電流が制限されます）。この過電流もしくはストール状態は、障害を生じないようにするために適切に設計されたシステムで処理することが必要となります。補正もしくは制御のためにソフトウェアにより電流値を読み取ることや､過電流が発生した際に PWM を停止する目的である程度のハードウェアを追加することは一般的な手法です。具体的にはハードウェアおよびソフトウェアの概要に関する章に記されています。

2.2　ハードウェアの概要

モータを駆動するには、RX62T デモキットの 3 相駆動回路の内の 2 相分を利用します。ハードウェアには MCU からの PWM 信号をモータ駆動のための基本的な H ブリッジを構成する N チャネルパワーMOSFET を駆動するのに適したレベルに変換するゲートドライバが含まれています。ブリッジは相補モードで駆動されますので、4 区分動作を行うことができます。H ブリッジ全体を使用することで単にデューティサイクルを変え、ブリッジの片側を他方よりハイレベルとすることで両方向回転のモータ制御を実現できます。図 3 は DC モータの駆動に使用される標準的なハードウェアです。

電流値は下側シャント抵抗と上側シャント抵抗の 2 個所で監視できます。

注: 上側の監視回路は標準の RX62T デモキットの設計には組み込まれていませんが､このアルゴリズムの実現ために追加されました。

電流値のほか、バッテリ電圧の上限と下限を管理できるようにバッテリ電圧を監視する必要があります。モータによってはバッテリ電圧が低いときに実行しようとすると通常より大きな電流が流れ障害が発生することがあります。またバッテリ電圧が高いときに実行するとモータの回転速度が定格速度を超えることがあります。2 個の抵抗からなる簡単な分圧器が 24V バッテリの電圧を RX62T の ADC で扱うことができるレベルの電圧に分圧しています。

注: H ブリッジによる駆動ではモータにかかる最大の DC 電圧はバッテリ電圧です。例えば 48V のバッテリを使用しデューティ比が 50%に制限されていれば有効な電圧は 24V となりますが、モータ端子には 48V が断続的に付加されています。モータの絶縁特性でこのことを考慮した設計とすることが必要です。

最後に、基準速度（希望の回転速度）はデモボード上のポテンショメータを使ってセットされます。ポテンショメータの可動片の電圧は ADC によって読み込まれます。

注: 以下の回路図はモータ電流と下側バス電流の両方を監視できます。一般に上側電流の監視はコストが多少余分にかかりますが、モータを流れる電流を常に読み込むことが出来るという大きな利点があります。下側シャント抵抗では還流電流を測定することは出来ません。

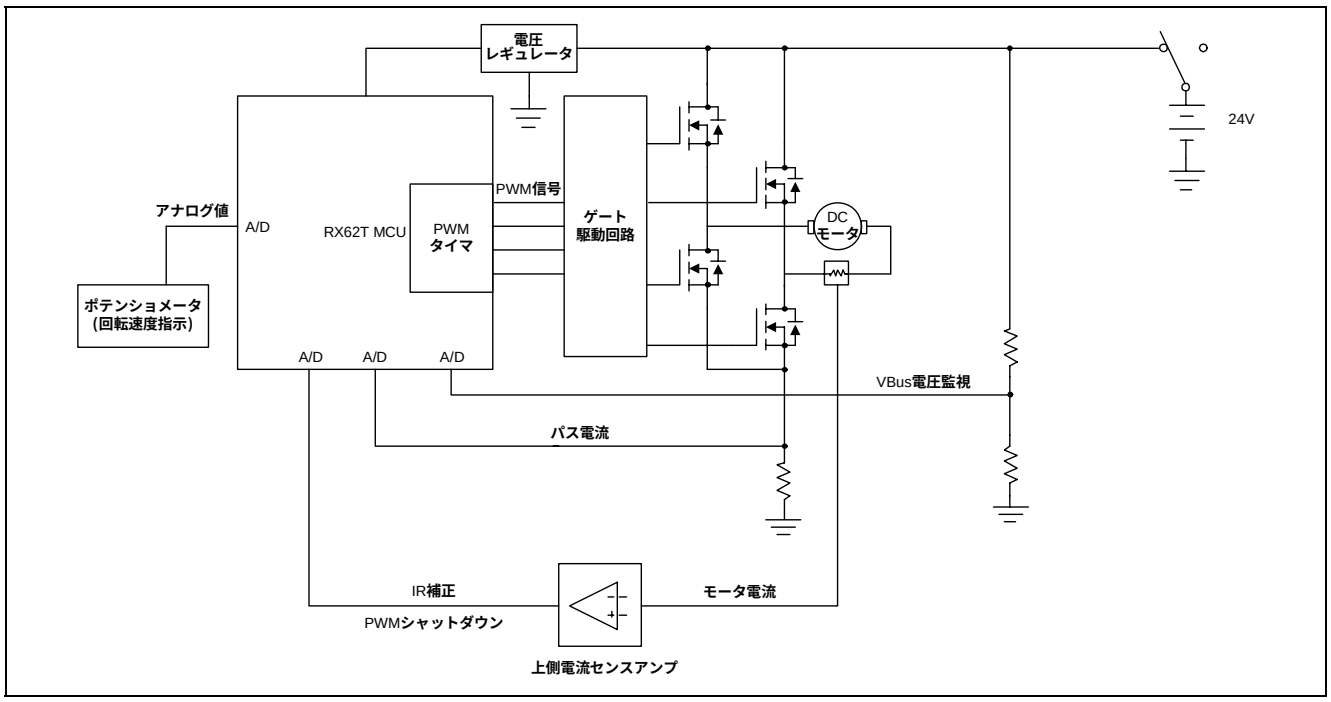

図 3　標準的な DC モータ駆動回路

3. ソフトウェアの説明

ここまでにモータのモデルとハードウェアについて見てきましたが、次にこれをモータ制御にどのように使用するかを説明します。デモプロジェクトのコードにおけるコメントは分かりやすく記述されているため、本アプリケーションノートでは主要な点についてのみ説明します。RX シリーズで有利な点は浮動小数点ユニットを搭載していることです。これにより固定小数点の場合の精度やスケーリングの問題を避け、アルゴリズムを直接に論じることが可能となります。

注: このソフトウェアの一部は分かりやすくするために複数行で記述されています（例として、モータ電圧と IR 補正のコードを区別しやすくするために、2 行に分けて記述しています）。これらの行を組み合わせることでソースファイルを容易に最適化することができます。

3.1 customize.h

このファイルについての詳細は省きますが、前述したたように、このソフトウェアでは浮動小数点演算を使用しており、システムを直接的に記述できます。このソフトウェアでは単精度演算を使用しており、パラメータは f サフィックスを使用して 1.23f や 1.0f などのように記述している点に注意してください。

customize.h は、PWM 定義、インバータハードウェア定義、およびモータ固有情報の 3 つのセクションに基本的に分けられます。

3.1.1 PWM の定義

キャリア周波数と（相補駆動を使用しているため）デッドタイムの 2 個の項目のみが対象となっています。キャリア周波数として 3,000 から 20,000 までの範囲の値が並べられています。このアルゴリズムでは単純な計算方法を使用していますのでこれ以上の周波数に拡張することもできます。

キャリアの値はこれをタイマとクロックカウントに関連付けている fix_par.h で使用されています。

3.1.2 インバータハードウェア

この章ではハードウェアとこれに関連したスケーリングを定義しています。コメントを参照すればコード内容が理解できるようになっています。これらの値は個々の ADC 読み込み時の変換定数（K）を計算するために使用されています。（詳細は 3.2「ADC スケーリング、オフセット、およびゲイン調整」を参照してください。）

```
/* MCU ADC definitions */
#define ADC_VREF        5.0f
#define ADC10_BIT_WGT  (ADC_VREF/1024.0f)  // in mV/bit
#define ADC12_BIT_WGT  (ADC_VREF/4096.0f)  // in mV/bit

/* Voltage divider for Monitoring VBus definitions */
#define VBUS_R1    47000.0f
#define VBUS_R2    10000.0f

/* Low-side shunt for Monitoring Motor current definitions */
/*  Low side shunts */
#define R_SHUNT_LO  0.100f          // 100 milli-ohms
#define I_AMP_GAIN  5.0f

/* High-side shunt for Monitoring Motor current definitions */
#define R_SHUNT_HI  0.050f          // 100 milli-ohms
#define I_AMP_GAIN_HIGHSIDE  50.0f   // LT1999 with 50 gain

/* dead-band in pot center to allow easier selection of 0 speed */
#define POT_DEADBAND   10.0f        // counts to ignore at middle of pot
```

3.1.3 モータの定義

モータの定義はモータに関して演算時に使用されるパラメータを作るために使われます。そのいくつかは他の演算定数を動的に計算するために使われています。これらの多くは、モータのデータシートに記載されています。巻線の抵抗値（R）が明示されていなければ、メータで測った値を出発点とし、モータのチューニングで上下に調整することができます。（4「ヒント、情報、および注意点」を参照してください。）

```
#define MOTOR_R          10.0f           // measured
#define MOTOR_VBEMF      24.0f
#define MOTOR_Ke        (24.0f/135.0f)  // volts-per-RPM (24/135)

/* System Parameters */
#define V_BUS_MAX        26.0f
#define V_BUS_MIN        22.0f
#define I_BUS_MAX        0.400f

// 24V Demo motor parameters
#define V_MOTOR_MAX      24.0f
#define V_MOTOR_MIN      22.0f
#define I_MOTOR_MAX      0.500f
#define MAX_RPM          135.0f      // Based on Motor
#define MIN_RPM          10.0f       // User selected based on requirements
#define K_RAMP           10.0f       // Ramp Constant in RPM/seconds
```

3.2 ADC スケーリング、オフセット、およびゲイン調整

ADC のスケーリングは customize.h ファイルにある値を使用して行われます。いくつかの定数は計算時に直接使用されており、他は実行時に定数値（K）を計算するために使用されます。関数 InitMotorEnvironment() では ADC チャネルの読み込み時にプログラムで使用される 3 個の定数値（K）を計算しています。ADC の値が読み込まれるとき、この定数を乗じて実際の値（ボルト、アンペア、などの実数値）に直接変換します。

k_vbus– VBus の ADC 読み取り値をボルトに換算するために使われます。

k_Ishunt– 下側シャント抵抗の電圧の ADC 読み取り値を電流に換算します（アンプゲインを含む）。

k_Ihigh_sense– 上側シャント抵抗の電圧の ADC 読み取り値を電流に換算します（アンプゲインを含む）。

これらの値は次の変換コードで直接使用されています。

```
v_bus = ADC_correction * (adc12_results[3] * k_vbus);

i_M = ADC_correction * ((k_Ihigh_sense * (float) ((int16_t) adc12_results[6]
      - i_M_offset)) );
```

計算を行うにあたって ADC_correction とは何かという疑問が生じるかも知れません。ADC の値には様々な誤差やエラーが含まれています。最も基本的な補正は ADC ゲインに関するものです。

このハードウェアプラットフォームでは、10 ビット ADC のチャネル 0 で正確な 4.25V の基準電圧を読むことができます。ADC ブロック全体が AVREF と AVCC で動いていますので、正確に 4.25V であることがわかっている電圧を読み込み、他の ADC の値をこの値（v_ref）との比率として読み込みます。ここから、すべての ADC チャネルに適用できるゲイン補正値（ADC_correction）が導かれます。

```
ADC_correction = 4.25f / v_ref;         // make a gain correction value
```

最後に、ADC は正の値のみを読み出すのに対して、モータ電流は通常正負両方の値であるため、ADC がこれらの値を扱えるようにこれらの信号を変換する方法が必要となります。このためモータ電流が 0 の時にアンプの出力が基準電圧（VRef）の 1/2 となるようなオフセットを通常加えています。オフセット値は正確なものではありませんので、モータ電流が 0 の状態(I=0)で何度か測定を行い、その平均値を取り、電流 0 のオ

フセット値として RAM に保存し電流値の算出に使用しています。ソフトウェアのスタート時にモータタイマの 8 ティック分の時間を確保し、読み込み値の平均値を取り保存します。

```
while (start_up_cnt1 !=0){};
i_v_offset /= 8;              // and average to get the offset
i_u_offset /= 8;              // and average to get the offset
i_M_offset /= 8;              // and average to get the offset
```

オフセット値を得た後、実際のモータ電流を算出するためにこのオフセット値とゲイン補正値を使用しています。

```
i_M = ADC_correction * ((k_Ihigh_sense * (float) ((int16_t) adc12_results[6]
      - i_M_offset)) );
```

要約すると、ADC から読み込まれた値はオフセット値を引いて正負両極性の値とします。これに上側電流センサの変換定数を乗じ､最後にゲイン補正値を掛けてアンペア単位のモータ電流の値(i_M)を得ています。

3.3 ハードウェアによる過電流検出

過電流の監視は RX62T の内蔵コンパレータによって行われます。モータ電流によってシャントの両端に生じる小さな電圧は図 3 にあるように LT1999 上側電流センサによって適切な電圧レベルに変換されます。このセンサは電流 0 が中間電圧（VREF/2）となるよう設定されています。モータの駆動は両方向ですのでコンパレータではウィンドウモードを使用し両方向の過電流を検出できるようにする必要があります。デモで使われているモータは小型で電流も小さいため、センサ電圧の許容範囲を中点に対して正負 1/8 に設定してあります。図 4 は時間に対するモータ電流の変化の一例です。モータは時間 0 にスタートし、負荷と回転速度の関数に応じてモータ電流が変化しています。ADC 端子の電圧として読み取られるモータ電流がウィンドウの限界を上または下に 16PCLK 以上の間超えると、ウィンドウコンパレータは内部で POE 回路を操作するように設定されていますので、PWM 信号はシャットダウンされます。PWM 信号はインバータのインアクティブな状態にプルアップもしくはプルダウンされているため、モータの駆動は停止されます。

注意: システムの慣性が大きくモータが回りつづけるときには､モータは発電機として機能し VBus に電流を供給します。設計者は能動的もしくは受動的なブレーキ機構でエネルギーを失わせる必要性を検討しなければなりません。

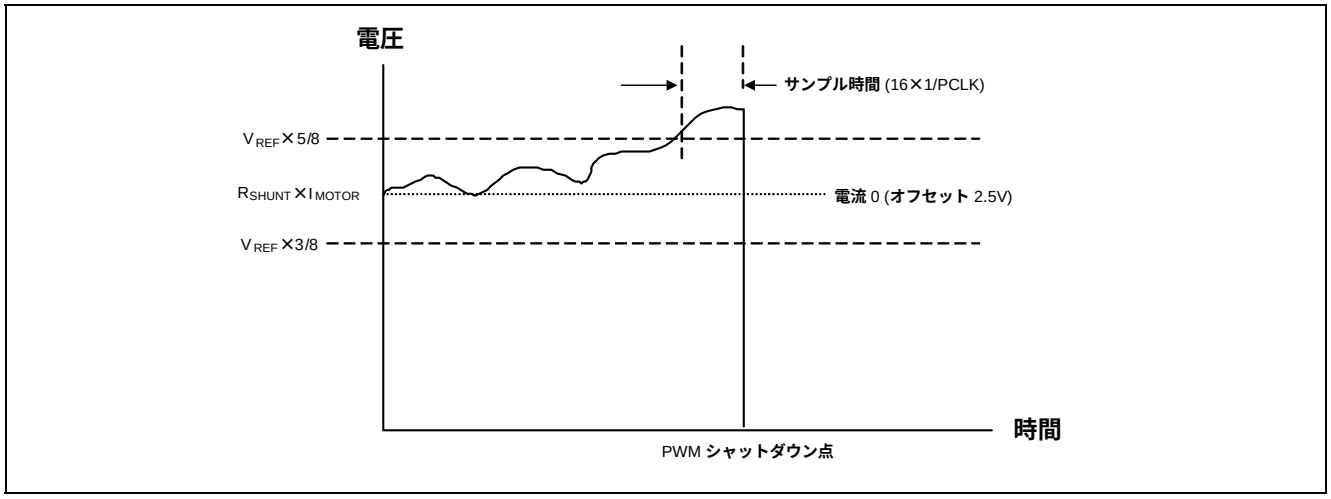

図 4 過電流の検出

3.4 モータ電圧の計算

今回の基本的な回路では VBus は通常バッテリ（もしくは低価格の電源装置）ですので、図 3 にあるように VBus の電圧は一定してしません。今回は VBus（この場合は電源電圧線）に抵抗による簡単な分圧回路を設け、電圧を監視しています。これにより次の機能が追加できます。

- バッテリ高電圧検出
- バッテリ低電圧検出
- 一貫性のある駆動のためのモータ電圧の修正

これらの機能でより安全な制御を実現し製品の寿命を長くすることができます。例えば、製品でバッテリを充電する発電ブレーキを使用するときには過充電と過電流を検出することができます。バッテリ電圧が大幅に低下したとき、これをモータに付加するとモータの駆動電流が通常より大きくなることがあります。これを避ければモータの寿命は長くなります。

さらに、VBus を監視して駆動電圧を補正することを行わなければ IR 補正は不正確なものとなります（アルゴリズムは VBus で使用できるより高い電圧による駆動を試みます）。

この状況を簡単に解説します。図 5 はモータ制御を行っている時間に対する VBus の変化です。青色の線は時間によるバッテリ電圧の変化を示しています。満充電のバッテリを使用し、時間 0 からモータがスタートします。この例では VBus 最大電圧（VBUS_MAX）を 26VDC に設定しています。VBus がこの電圧を超えることがあれば、このデモソフトでは単にフラグ（v_bus_err）をセットし VBus エラーを表示しています。

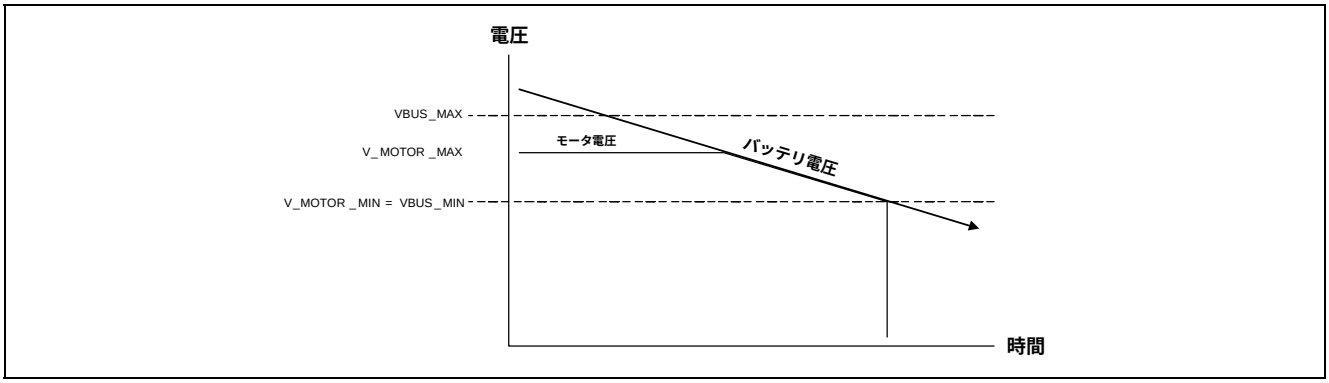

図 5　VBus 対モータ電圧

モータの最大回転数からスタートし、モータの仕様として定められたモータ最大電圧を付加します。例えば、このアプリケーションでは 24VDC モータを使用していますので customize.h 内で VMOTOR_MAX は 24.0f に設定します。モータが最大回転数で回転するためには、駆動電圧を 24VDC に維持します。26VDC が使用可能ですので、このモータ電圧を使用可能な電圧の関数として設定しなければなりません（モータ電圧は使用可能な電圧と希望の電圧の比率の関数となります。）

最初に、基準速度の関数から指示されるモータ電圧を計算します。

```
V_motor_commanded =  MOTOR_Ke * ref_speed;
```

次に、IR 補正を行う場合（IR_Comp_on==TRUE）には、モータの抵抗値（R）による電圧低下分を加えます。

```
V_Motor_Drive = V_motor_commanded + (I_Motor * MOTOR_R);
```

図 5 によれば、ある時点でモータ駆動電圧（計算値）が実際の VBus 電圧より高いことがあります。このときには VBus の電圧を追って、モータ駆動電圧を実際に供給されている VBus 電圧にあわせます（IR 補正が不安定になることを防ぐため、モータ駆動電圧を実際にバッテリから供給されている範囲に修正しています）。

```
if ( V_Motor_Drive > max_v_avail)
{
  V_Motor_Drive = max_v_avail;
}

if ( V_Motor_Drive < -max_v_avail)
{
  V_Motor_Drive = -max_v_avail;
}
```

最後に、この値を VBus に依存する値とし、さらに PWM カウント値に変換します。

```
V_motor_cnt = (int16_t) (V_Motor_Drive/v_bus * ((float) CARRIER_CONSTANT_2));
```

PWM カウントはキャリア定数の 1/2 を示す CARRIER_CONSTANT_2 に電圧比を掛けたものです。デッドタイムを無視すれば、この値が PWM の最大値です。カウント値が決まると、V_motor_count の符号によって、U 相（+モータドライブ）を V 相（-モータドライブ）より大きくするか否かを決めます。これはモータの回転方向を定めますので、モータが正しい方向に回転するよう U 相と V 相のカウント値を計算します。コードを見ればお分かりのように、デューティ比 50%に相当するキャリア定数の 1/4 を元にして、右回り（CW）ならばカウント値の 1/2 を U 相に加え 1/2 を V 相から減じて、U>V としています。以上でモータは計算で求められた V_Motor_Drive により右回りに駆動されます。

```
if (0 > V_motor_cnt)
{
    V_motor_cnt = (int16_t)-V_motor_cnt;
    pwm_u = (uint16_t)(CARRIER_CONSTANT_4 - (V_motor_cnt/2));
    pwm_v = (uint16_t)(CARRIER_CONSTANT_4 + (V_motor_cnt/2));
}
else
{
    pwm_u = (uint16_t)(CARRIER_CONSTANT_4 + (V_motor_cnt/2));
    pwm_v = (uint16_t)(CARRIER_CONSTANT_4 - (V_motor_cnt/2));
}
```

注: このプログラムでは VBus エラー発生時に、フラグをセットする以外の対応はしていません。設計者は個々のシステムの要件に応じてこのときの対処を決めなければなりません。

3.5 PWM 駆動

以上で計算手順が示されましたので、これによるモータ駆動の波形とモータ電圧を示します。図 6 はモータが右回り（CW）のときの実際の PWM 駆動の状況を示しています。実際にモータに電圧を付与している部分は約 20μS もしくはサイクルの約 35%で、電圧では 24V×0.3 もしくは電源電圧の 30%です。+側が長くドライブされており、正方向（このアプリケーションノートでは右回り（CW）を意味しています）駆動になっています。相補ドライブを行っていますので、還流期間のモータ電流は止められず、上側もしくは下側の MOSFET でモータが短絡されている点に注意してください。これはモータで発電ブレーキが働いていることになります。モータがより低い電圧で駆動されている（例えばモータが負荷によって回転させられている）ときにはより大きな発電ブレーキ力が働き、モータの回転数が維持されます。実際にモータが外から回される速度が充分に速いときには、電流は逆に流れ、逆のモータ電圧（ブレーキ効果の追加）が発生します。

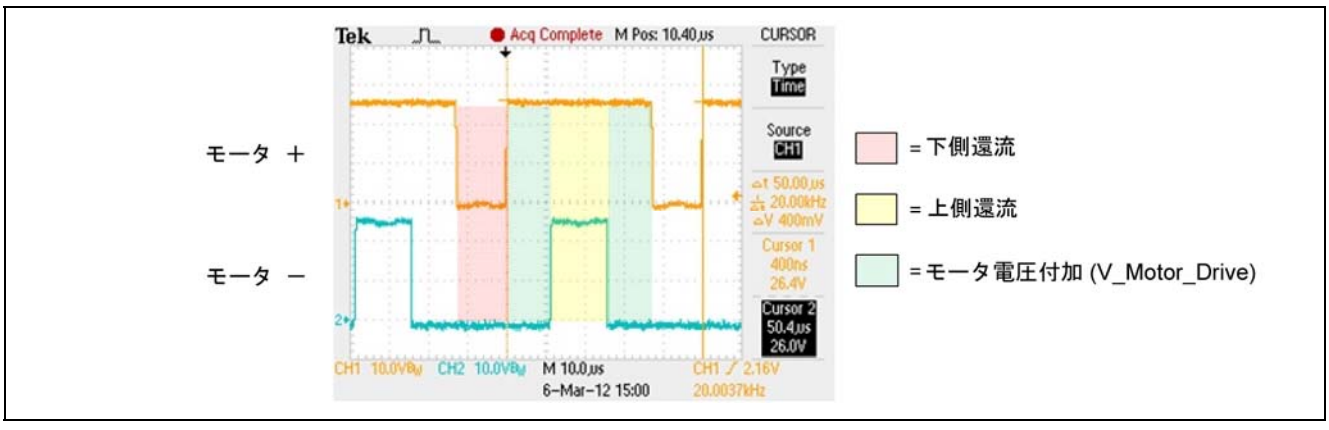

図 6 右回転（CW）方向のモータ駆動

比較のために、次の図はモータが左回り（CCW）の状況を示しています。このときには発生するモータ電圧によりモータの－側のモータドライブ時間が長くなっています。

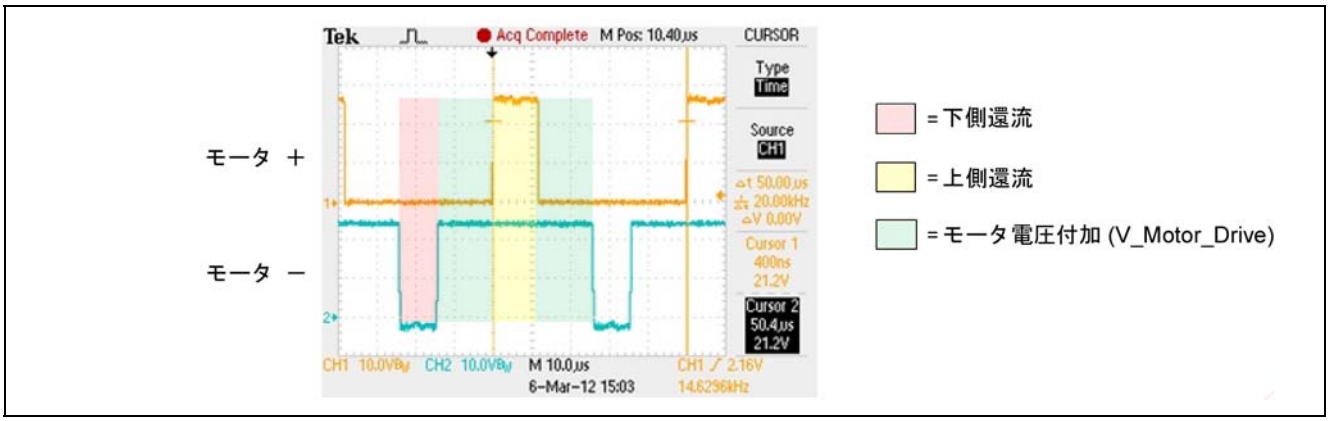

図 7　左回転（CCW）方向のモータ駆動

3.6　モータ制御の状態遷移

ここまでにモータの駆動方法を紹介してきましたので、今度はもう一段上のレベルから、デモソフトウェアでモータがどのように制御されるかを決める簡単な状態遷移（ステートマシン）をより抽象的なレベルで紹介します。

図 8 はモータの状態と操作を決定する UI を示す簡単な状態遷移図です。

ユーザの指示前（読みこまれる指示は多くの場合不適切）にモータが作動することを防ぐため、つまり電源投入時にモータが突然動かないよう、状態遷移は ERROR（エラー）状態から開始されます。デモソフトでは ERROR 状態から抜け出すには、回転速度として 0 が指示されなければなりません。これをユーザからの最初の指示とみなします。

STOP（停止）状態に入ると、UI（またはその他の制御メカニズム）により 0 以外の指示速度があることを確認し START 状態に移ります。

START（スタート）状態はモータが実際に動作する前に必要な管理を行うための一時的な状態です。管理内容としては、ブレーキ信号の解除や、ブートストラップコンデンサをチャージする時間の確保などがあります（これは例で示されています）。デモソフトでは main()関数の while(1)ループから制御が戻ると直ちに次の状態に移行しています。状態の遷移をクロックや他のメカニズムによって制御することも可能ですが、本アプリケーションノートは状態遷移について詳述することを目的としていません。

次に RAMP（速度変更）状態に移ります。RAMP 状態は、モータ制御の基準速度がユーザからの指示速度と異なっている状態です。RAMP は customize.h ファイルにある定義の値に基づいています。基準速度は指示速度と一致するまで増速もしくは減速されます。この処理は一定の間隔で発生するモータ制御割り込み時に実行されます(これは基本的に変化幅（rampvalue）とモータタイマのキャリア周波数によって決まる小さなステップです)。これによりユーザにより定義されたスムーズな速度変更が実行されます。このデモソフトでは直線的な速度変化（台形制御）のみを行っています。ユーザからの指示速度に変更があると RAMP 状態に入り、「全速前進」から「全速後退」への移行であってもモータ速度をスムーズに変化させます。指示速度と基準速度が一致すると、RUN 状態に移行します。

RUN（定常運転）状態では、指示速度を監視します。指示速度が基準速度と異なる時には、モータ速度の急激な変化を避けるために RAMP 状態に戻ります。速度 0 が指示されたときの制御も RAMP 状態で行われ、指示速度と基準速度の両方が 0 の速度で安定した時点で RUN 状態に戻り、次に STOP 状態に戻ります。

MCU が過電流を検出すると、ERROR 状態に移ります。UI で速度 0 を指示すると、ERROR 状態を抜け出すことができます。実際のシステムでは、エラー発生時にはさらに複雑なタスクが必要となる場合がありますが、このような非常に小さなモータを使用したデモンストレーションのアプリケーションでは、そこまでの処理は必要ありません。

BRAKE（制動）状態が用意されていますが、制動に関する操作は行っていません。慣性が大きなシステムでは、バス上の電流が大きくなり過ぎないようにモータの回転エネルギーを解消させる制動機構が必要となります。実際のシステムでは、バッテリの充電や他の「グリーンエネルギー」機能によってエネルギーを再生することも考えられます。

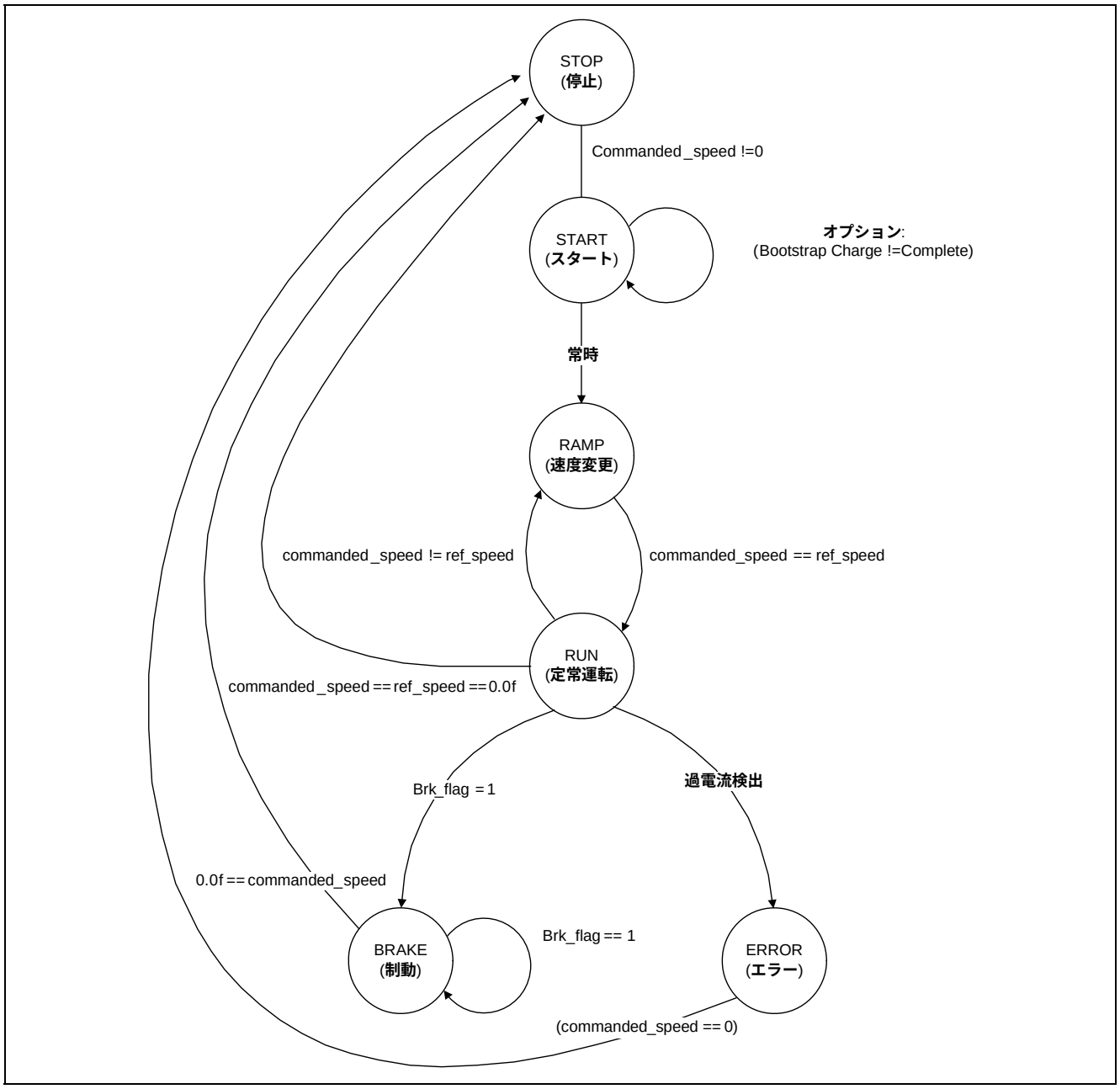

図 8　モータ制御のステートマシン

4. ヒント、情報、および注意点

4.1 チューニング

モータ駆動のチューニングに関しては、理論的な内容は省き、簡単に注意点を記載します。

IR 補正は基本的に正帰還系です。負荷が大きくなるほど駆動電圧も大きくなります。多くのケースでこの点が問題となることはお分かりと思います。IR 補正では R はゲインに相当すると考えることができます。定義ファイルで R により大きな値を与えることは、より大きな駆動電圧をモータに付加するアルゴリズムとなります。ゲインが大きすぎると補正が正確に設定されていないブラシ付き DC モータの制御では振動が観察されることがあります。IR 補正の調整に関してはウェブ上で多くのホワイトペーパーを確認することができ、そのなかでも最も有効なものはモータ制御システムの製造会社のものです。

4.2 不安定性

IR 補正のチューニングとは別に、他にも不安定性の要因があります。一例として、定義されたモータの最大電圧が VBus で供給される電圧を超えている場合があります。3.4「モータ電圧の計算」にあるように、VBus がモータの最大電圧以下であると、ソフトウェアは電圧をモータの最小電圧に引き下げ、その後にフラグをセットして PWM を停止しています。これを行わないとアルゴリズムではモータの電圧を実際には不可能な値まで上げようと試み続けます(VBus が 22V で、指示速度に達するために 24V の指示が必要とされる場合)。

4.3 下側シャント抵抗による動作

このデモコードは下側シャント抵抗のみを利用して動作させることが可能ですが、このモードにおける十分なテストはなされていません。このモードを使用するには HEW のビルドダイアログ C/C++にある定義を USE_ADC1 から USE_ADC0 に変更します。

5. 限定された試験内容について

この製品に含まれるソフトウェアは IR 補正のデモに必要な限られた範囲の試験のみが行われています。ソフトウェアは最小限の機能のドライバを使用し、デモコードを実現するための最小限のステートマシンを採用しています。実際の製品にそのまま使用されることを想定したものではありません。

モータ制御のアルゴリズムは限られた摩擦負荷を使用してテストされました。さまざまな負荷条件でモータ速度の精度と安定性を判断するための動力計を使用した測定は実施していません。

6. 参考資料

6.1 外部資料

Brushed DC electric Motor, Wiki - http://en.wikipedia.org/wiki/Brushed_DC_electric_motor

Joliet Technolgies, DC Drive Fundamentals - http://www.joliettech.com/dc_drive_fundamentals.htm#ir-comp

eCircuit DC Motor Model - http://www.ecircuitcenter.com/circuits/dc_motor_model/dcmotor_model.htm

Carnegie Melon Control Tutorials for Matlab, Example DC Motor Speed Modeling

http://www.engin.umich.edu/group/ctm/examples/motor/motor.html

Linear Technologies High Side Current Sense Amplifier LT1999 Data Sheet

http://www.np.edu.sg/alpha/nbk/alphastu/DCMotor.html

Autotrol Motor 220-0106A, Mouser # 708-2200106A, http://www.autotrol.com/

6.2 ルネサス

RX62T Motor Demonstration Kit Schematics

RX62T Hardware Manual, R01UH0034EJ0110 Rev.1.10

7. 用語

ブートストラップコンデンサ－ 上側ゲートドライバのゲート電圧を作り出すために使われる小容量のコンデンサで、これによりゲート駆動のために別の電源を追加する必要がなくなります。

還流－ 両方の上側 MOSFET もしくは両方の下側 MOSFET のいずれかがオンのときに、実効的にモータをショートしているためにモータ電流が循環して流れる現象。

ホームページとサポート窓口

ルネサスエレクトロニクスホームページ
http://japan.renesas.com/

お問合せ先
http://japan.renesas.com/inquiry

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2013.02.08	—	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. 未使用端子の処理
   【注意】未使用端子は、本文の「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。未使用端子は、本文「未使用端子の処理」で説明する指示に従い処理してください。
2. 電源投入時の処置
   【注意】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. リザーブアドレス（予約領域）のアクセス禁止
   【注意】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、将来の機能拡張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。
4. クロックについて
   【注意】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。
5. 製品間の相違について
   【注意】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うと、内部 ROM、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。お客様の機器・システムの設計において、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て、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。
2. 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ものですが、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。
3. 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に関し、当社は、何らの責任を負うものではありません。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。
4. 当社製品を改造、改変、複製等しないでください。かかる改造、改変、複製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。
5. 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）、交通用信号機器、
   防災・防犯装置、各種安全装置等
   当社製品は、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（原子力制御システム、軍事機器等）に使用されることを意図しておらず、使用することはできません。たとえ、意図し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損害が生じても、当社は一切その責任を負いません。なお、ご不明点がある場合は、当社営業にお問い合わせください。
6. 当社製品をご使用の際は、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他の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。
7. 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。
8. 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。お客様がかかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
9. 本資料に記載されている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。また、当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途に使用しないでください。当社製品または技術を輸出する場合は、「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。
10. お客様の転売等により、本ご注意書き記載の諸条件に抵触して当社製品が使用され、その使用から損害が生じた場合、当社は何らの責任も負わず、お客様にてご負担して頂きますのでご了承ください。
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禁じます。

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。

ルネサス エレクトロニクス株式会社

■営業お問合せ窓口 http://www.renesas.com

※営業お問合せ窓口の住所・電話番号は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス販売株式会社　〒100-0004　千代田区大手町2-6-2（日本ビル） (03)5201-5307

■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：http://japan.renesas.com/contact/