

## RL78/G13

R01AN3061JJ0100

Rev. 1.00

2015.12.15

## DMA を利用した多重 PWM 生成 CC-RL

### はじめに

アプリケーションの中には、小さなパッケージに対してハードウェアの対応範囲を上回る PWM 出力を求めるものもあります。このアプリケーション・ノートでは、追加で PWM 出力を生成するために、タイマや DMA コントローラをどのように使用するのか、ご紹介します。

### 対象デバイス

このアプリケーションは RL78/G13 ターゲット・ボード (QB-R5F100LE-TB) でテストしておりますが、DMA ユニットを持つ他の RL78 デバイスに簡単に移植できます。

本例では TAU (タイマ・アレイ・ユニット) 0・チャンネル 0 および DMA チャンネル 0 を使用しますが、他のタイマや DMA チャンネルでも可能です。

### 関連アプリケーション・ノート

本アプリケーション・ノートは以下のアプリケーション・ノートをもとに作成されました。併せてご参照ください。

- RL78/G13 Multiple PWM generation using DMA (R01AN0717EE0100) Application Note

### 目次

1. 多重 PWM 生成 .....	2
2. 本アプリケーションに使用されている周辺アーキテクチャ .....	6
3. 結論 Conclusion .....	7
4. フローチャート .....	8
5. Appendix: Code listing .....	9

## 1. 多重 PWM 生成

### 1.1 DMA について

ダイレクト・メモリ・アクセス・コントローラ、または DMA コントローラは、DMA をサポートしているペリフェラル、SFR レジスタ、および内部 RAM の間で、CPU を使わずにデータの転送をおこないます。その結果、CPU の通常の内部動作やデータ転送は並行して実行され、SFR と内部 RAM 間での同時転送が可能になります。さらには通信、タイマ、A/D を使用したリアルタイム制御も実行できます。

このアプリケーション・ノートでは、8つの separate PWM 出力が生成されますが、DMA チャンネル追加の微修正により、もっと増やすことも可能です。PWM 出力はベース時間が同一のため、完全に独立しているわけではない点にご注意下さい。

DMA は周辺機能割り込みによって起動し、データ転送をおこないます。DMA および CPU は同じバスを使用し、DMA は CPU よりもバス優先権が優位になります。使用の前に、DMA コントローラは特定の設定で初期化する必要があります。この設定には転送元アドレス、転送先アドレスおよび動作モードが含まれており、DMA コントローラ・レジスタに allocate されます。

### 1.2 本アプリケーションの概要

このアプリケーションでは、PWM 出力を生成するために、内部 DMA チャンネルを使用してポート 7・データ・レジスタの内容を変更します。

インターバル・タイマ・モードの TAU (タイマ・アレイ・ユニット) が、DMA チャンネル 0 設定のトリガとして使用されます。PWM 出力の更新周波数は TAU・ユニット 0・チャンネル 0 の期間で定義します。DMA はデータを RAM テーブルから P7 ポート・レジスタへと転送します。また、RAM テーブルは、MCU アプリケーション・ソフトウェアによって更新されます。

図 1 に示されるような操作は、例えば高輝度 LED ドライバ制御のような、特殊な信号を生成するために使用することもできます。

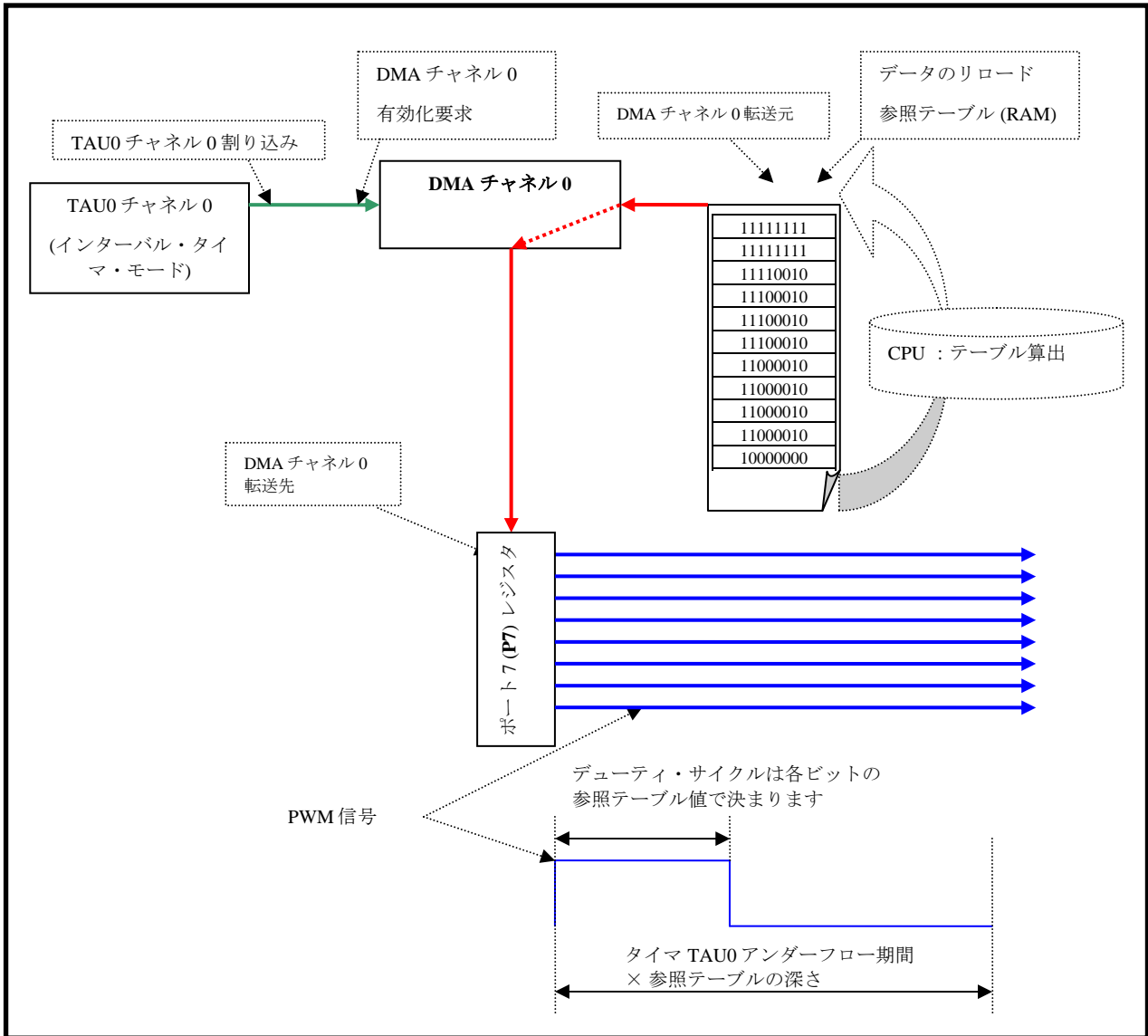


図.1 操作ブロック図

### 1.3 方式

データを RAM の PWM 参照テーブルから I/O ポートに自動転送する目的で、DMA チャンネル 0 を設定します。DMA カウンタは、DMA 転送終了割り込みが生成されたあと、参照テーブルの終端に到達すると同時に CPU によって再び初期化されます。

「PWM 参照テーブル」の深さは要求された PWM 分解能によって決まります。各 PWM 出力は、各 TAU0 チャンネル 0 割り込み（以後「タイマ・ユニット」と称します）で更新されます。このアプリケーション・ノートとコード例では、任意の選択として、8 ビット分解能、8 つの PWM をポート 7 に出力する設定になっています。ですから、出力 RAM データに 256 バイトが必要です。

「タイマ・ユニット」は、アンダーフローごとに DMA チャンネル 0 のトリガとなる TAU0 チャンネル 0 によって決まります。

それゆえに、PWM 出力の周波数は更新周波数と、P7 ポート・レジスタへ出力するのに使われる RAM 内部参照テーブルの長さによって決定されます。

$$\text{PWM 周波数} = 1 / ((\text{TAU0 チャンネル 0 期間}) \times (\text{PWM 参照テーブルのバイト数}))$$

## 1.4 PWM 参照テーブルの構成

DMA チャンネル 0 のアクセスする RAM に格納された PWM 参照テーブルをより簡単に理解するため、8 つの PWM 出力がポート 7 に出力され、16 個の値が使われていると仮定しましょう。つまり、4 ビット分解能の PWM となります。表 1 は、TAU0 チャンネル 0 がタイマ周期をアンダーフローする度に、DMA によって RAM データが P7 データ・レジスタへ自動的に転送される代表例を示しています。テーブル値と生成される PWM 出力値の関係は図 2 に示します。色つきでハイライトされている項目は、PWM ステータスの変更が起きているポイントを示しています。

この例では、ビット 7 (b7) はポート 7.7 の出力となり、ビット 6 (b6) はポート 7.6 の、ビット 1 (b1) はポート 7.1 の出力となります。

参照テーブル方式を使用することの利点として、同じタイム・ベースを使いながら、各出力チャンネルの色々な PWM 周波数を許可できる点があります。非常に柔軟性のある PWM 波形が PWM 参照テーブルのデータに基づいて生成でき、機能が基本的な PWM から強化された波形生成へと拡張できます。

CPU の主な働きは、テーブル値の初期化または修正です。一度そのタスクが実行されると、DMA は PWM 参照テーブルからポートへ、データの転送を独立しておこないます。

表.1 RAM における PWM 参照テーブルの構成

インデックス (ベース時間ごとに address per each time base)	参照テーブル・データ (RAM)							
	b7	b6	b5	b4	b3	b2	b1	b0
0	1	0	X	X	X	X	1	X
1	1	0	X	X	X	X	1	X
2	1	0	X	X	X	X	1	X
3	1	0	X	X	X	X	1	X
4	1	1	X	X	X	X	1	X
5	1	1	X	X	X	X	1	X
6	1	1	X	X	X	X	1	X
7	0	1	X	X	X	X	1	X
8	0	0	X	X	X	X	1	X
9	0	0	X	X	X	X	1	X
10	0	0	X	X	X	X	1	X
11	0	0	X	X	X	X	1	X
12	0	1	X	X	X	X	0	X
13	0	1	X	X	X	X	0	X
14	0	1	X	X	X	X	0	X
15	0	1	X	X	X	X	0	X

図2は DMA 転送後においてポート出力が PWM 参照テーブルの値にどのように対応するかを示しています。分かりやすさのため、出力が PWM 参照テーブル表 1 にて同色で示された値に対応して変化する箇所につき色の縦の点線が示されています。

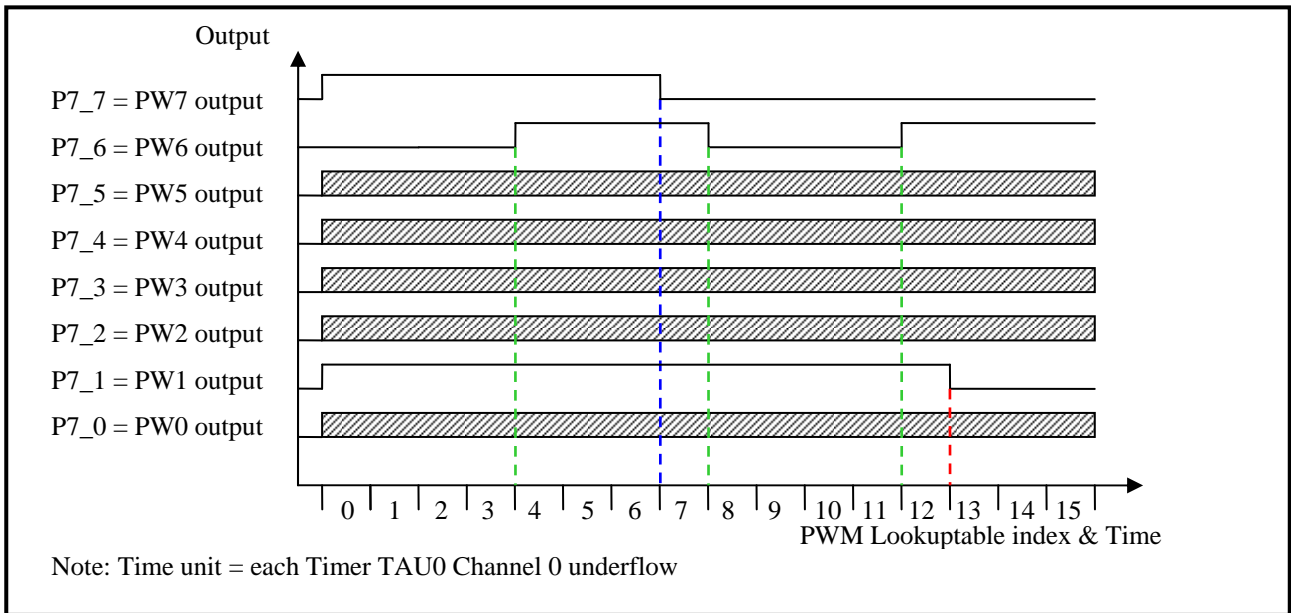


図.2 DMA 転送中のポート出力状態と PWM 参照テーブルとの関係

### 1.5 PWM 参照テーブル算出

PWM 参照テーブルの初期化は、極めて直接的におこなわれます。本アプリケーション・ソフトウェアでは、PWM チャンネル番号ごとに、インデックス 0 から N までの PWM 参照テーブルを 1 にセットして（各 PWM のハイレベルを意味します）、そして、インデックス N+1 からインデックス最大値までを 0 にクリアしなければなりません。そこで、PWM 出力番号 7 が、新しい値「156」で 1 にセットされるという例を取り上げてみましょう。この例では PWM 分解能は 8 ビットで、PWM 参照テーブルは 256 個の入力またはインデックスを必要としていると仮定します。この例では、本アプリケーション・ソフトウェア PWM 参照テーブルを以下のように初期化します。

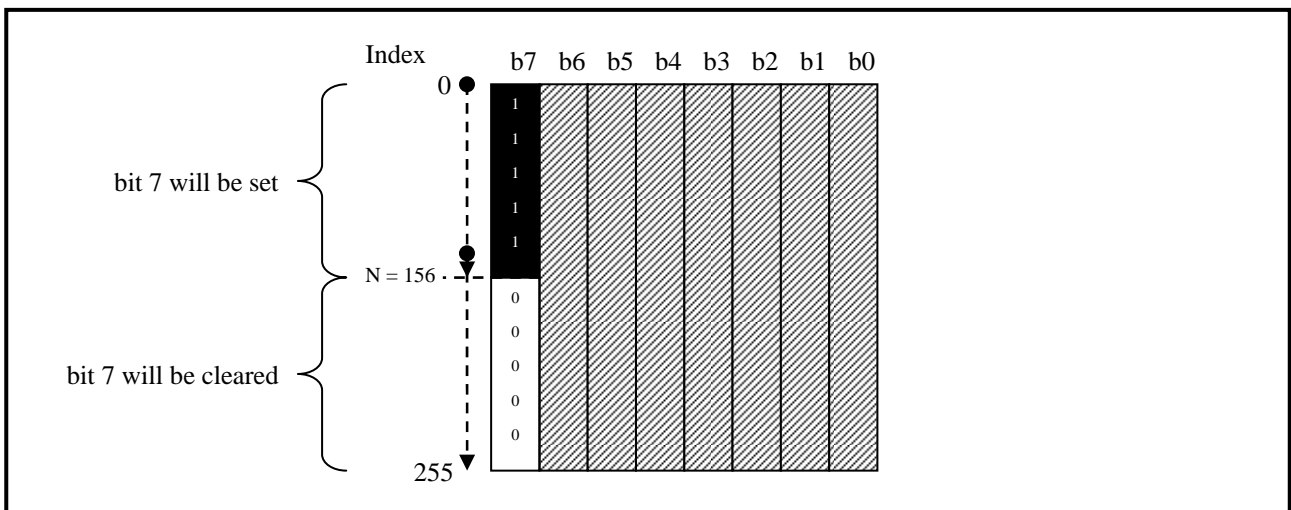


図.3 PWM 生成のための PWM 参照テーブル初期化

PWM 参照テーブルの値を変更することにより、他の信号波形を生成することも可能です。たとえば本例では、ワンショット信号生成のウェイトをプログラムで変更できます。

## 2. 本アプリケーションに使用されている周辺アーキテクチャ

### 2.1 TAU0 チャンネル 0、インターバル・タイマ・モードにおける実行タイミング

RL78/G13 は多くのタイマを持っています。タイマ・アレイ・ユニットには 8 つの 16 ビット・タイマがあります。各 16 ビット・タイマはチャンネルと呼ばれており、独立したタイマとして使用することができます。タイマ・アレイ・ユニット 0 チャンネル 0 は、インターバル・タイマ・モードにおいて、DMA のトリガに選択されています。このモードでは TAU は内部で生成されたカウント・ソースをカウントして、INTTM00 (タイマ割り込み) を一定間隔で生成する、参照タイマとして使用することができます。タイマがアンダーフローしたときは、TDR00 レジスタがカウント・ソース・レジスタにコピーされ、図 4 に示されるように、カウントが再開します。

割り込み生成間隔は以下の式によって算出できます：

$$\text{INTTM0n (タイマ割り込み) の生成間隔} = \text{カウント} \cdot \text{クロック期間} \times (\text{TDR0n の設定値} + 1)$$

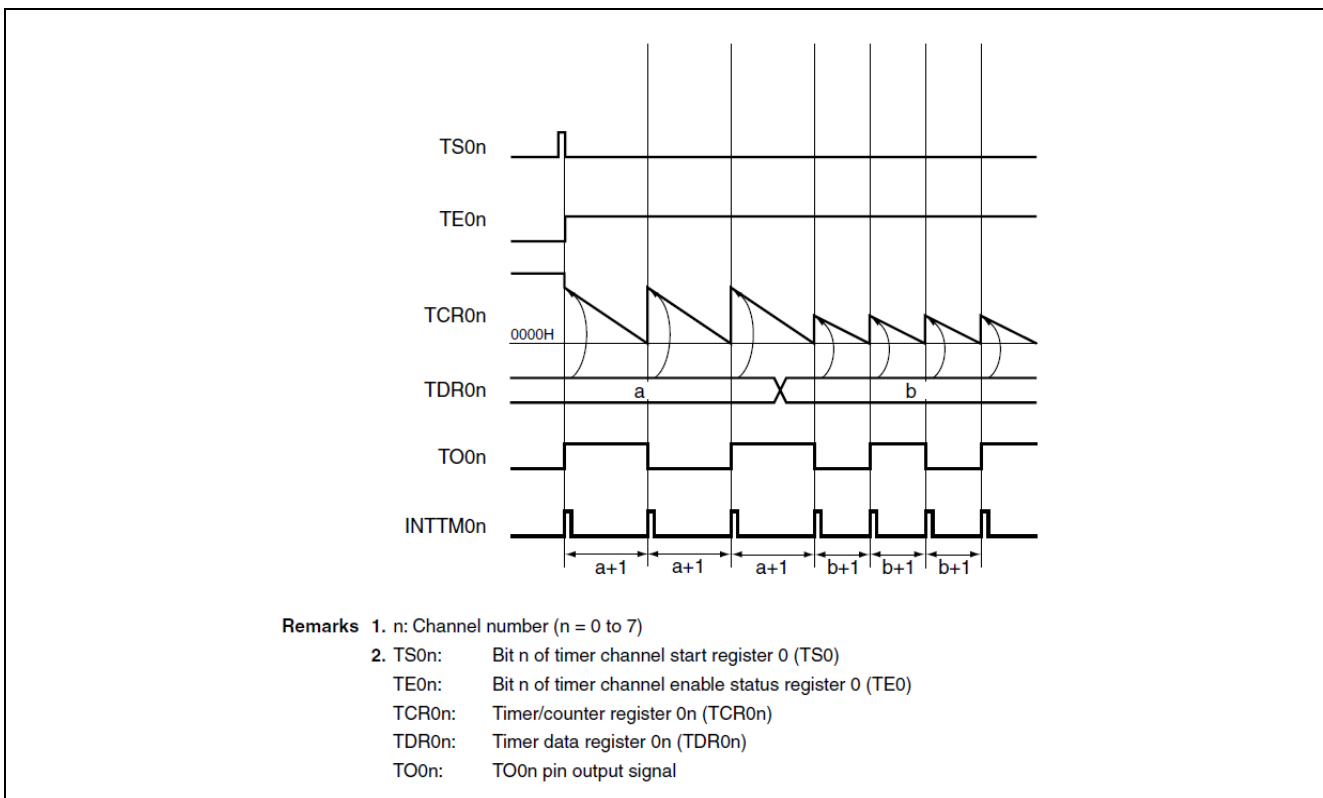


図.4 カウンタ値(TDR0n)がカウント期間中に再書き込みされた場合の TAU 動作例

本アプリケーション・ノートにおいて、TAU のアンダーフロー期間は、PWM 生成のための DMA 起動の基礎となります。さらには、内部バス負荷にインパクトがあるので、TAU 設定はそのアンダーフロー周波数を考慮に入れてなされなければなりません。言い方を変えれば、転送頻度が上がれば上がるほど、CPU の活動に残されるバス帯域が少なくなるということです。

## 2.2 DMA 操作と設定

ソフトウェア・トリガ (DMA チャンネル 0 のための STG0) または IFC03 ビットから IFC00 ビットまでによって特定されたスタート・ソース・トリガが入力されるときに、DMA 転送が開始されます。各トリガ、すなわち TAU0 チャンネル 0 のアンダーフローにおいて、DMA は DMC0 レジスタ (DMA チャンネル 0 モード制御)、DRC0 (DMA チャンネル 0 動作制御)、および DBC0 レジスタ (DMA チャンネル 0 カウント) の設定に従って、データをソース・アドレスからデスティネーション・アドレスへ転送します。

RL78/G13 の DMA の主な特徴と、本アプリケーション・ノートにおける影響は以下の通りです。

- 転送単位は 8 ビットまたは 16 ビットです。このアプリケーション・ノートでは、PWM 生成の出力ポートはポート 7 で、結果として転送サイズは 8 ビットとなります。
- 転送の最大数は 1024 です。DMA 転送が実行される度に、この数字を表すレジスタがディクリメントされます。それは DBC0 レジスタです。この DBC0 レジスタを読み出すことにより、転送の残り回数を確認できます。しかしながら、このレジスタは DMA 転送中はアプリケーション・ソフトで修正することはできません。このアプリケーション例では、転送数は 256 が選択されています。
- 1 回の転送が 2 クロックで処理されるので転送タイプは 2 サイクルとなります。その間 CPU は停止します。
- RL78/G13 の DMA で利用可能な転送モードはシングル転送です。言い換えれば、トリガごとにデータ (8 ビットまたは 16 ビット) の転送がなされます。
- 転送の要求は、A/D コンバータ、シリアル・インターフェース、そして本アプリケーションの TAU0 チャネル 0 などの多様なペリフェラル・ハードウェアから選択可能です。
- 転送は内部 RAM と SFR の間でおこなわれます。

## 2.3 PWM 分解能の増大

本アプリケーション例での転送数は 256 となるよう選択されています。これは、それぞれの PWM 出力の 8 ビットの分解能と同じです。

PWM 分解能の最大は 10 ビットで、これは DMA 転送数の最大値 1024 に対応しています。

内部バス帯域と PWM 生成能力に影響があるので、DMA 動作とタイミングを考慮に入れなければなりません。

## 2.4 バス負荷

TAU0 チャネル 0 がアンダーフローするたびに、各転送で 2 サイクルを必要とします。さらには、総転送がそれぞれ完了するたびに、DMA チャネル 0 を再設定しなければなりません。DMA チャネル 0 割り込みサービスルーチンで DMA が再び初期化されます。

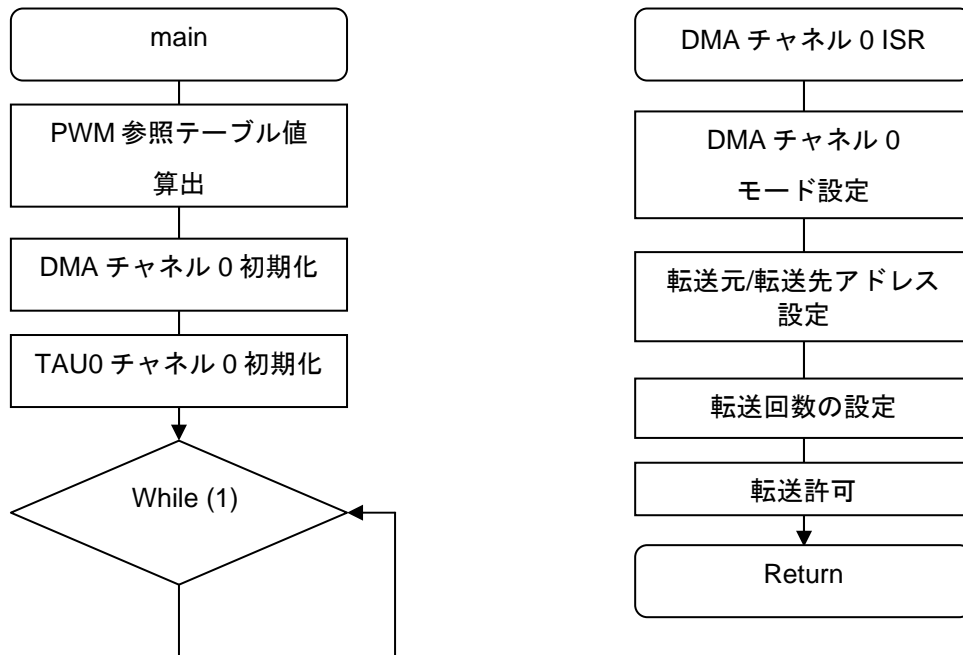
## 3. 結論

DMA、I/O ポート、そして PWM 参照テーブルを使用することで、追加コスト無しで PWM チャネルが追加できます。

シングル DMA チャネル、8 ビット I/O ポート、1 つのタイマ、それと 10% 未満の CPU バス帯域を使えば、8 ビット分解能を持つ 8 つの PWM を 2kHz で生成するのは容易です。(すべてのタイマ TAU0 チャネル 0 のアンダーフローと 256 転送ごとの DMA チャネル 0 再設定の転送に 2 サイクルが要ります。)

しかしながら、影響なくアプリケーション・ソフトウェアを実行できる十分な帯域を CPU が持てるよう、PWM 周波数には注意しなければなりません。

## 4. フローチャート





## 5. Appendix: コード・リスト

```

void main(void)
{
    uint8_t i = 0U;
    int bit_number = 0;
    int k = 0;

    __low_level_init();
    EI();// enable interrupt

    /* Set initial values for the 8 PWM channels */
    pwm_dutycycle[0U] = 1U;
    pwm_dutycycle[1U] = 64U;
    pwm_dutycycle[2U] = 96U;
    pwm_dutycycle[3U] = 128U;
    pwm_dutycycle[4U] = 160U;
    pwm_dutycycle[5U] = 192U;
    pwm_dutycycle[6U] = 224U;
    pwm_dutycycle[7U] = 250U;

    /* Scan 8 PWM channels */
    for (i = 0U, bit_number = 1; i < 8U; i++)
    {
        /* Start at high value */
        k = MAX_TABLE - 1;
        /* if PWM register content is different from highest value */
        if (pwm_dutycycle[i] < (MAX_TABLE - 1U))
        {
            for (;k >= pwm_dutycycle[i]; k--)
            {
                /* Clear bit to zero (0) from Max-table to PWM value */
                PWM0_7[k] &= ~bit_number;
            }
        }
        if (pwm_dutycycle[i] > 0U)
        {
            for (;k >= 0; k--)
            {
                /* Set bit to one (1) from PWM value to 0 */
                PWM0_7[k] |= bit_number;
            }
        }
        /* Next bit position */
        bit_number <<= 1;
    }

    DMA0_Init();
    TAU00_Init();

    while (1U)
    {
        ;
    }
}

```

```

/*****
* Function Name : DMA0_Init
* Description : DMA Channel 0 initialization
* Argument : none
* Return Value : none
* Calling Functions : none
*****/
void DMA0_Init(void)
{
    /* DMA operation enable
       b7 DMA operation enable bit
       b6:b1 Reserved set to 0
       b0 DMA transfer mode bit
    */
    DRC0 = 0x80U;

    NOP();// no operation
    NOP();// no operation

    /* Disable INTDMA0 interrupt */
    DMAMK0 = 1U;
    /* Set INTDMA0 low priority */
    DMAPR10 = 1U;
    DMAPR00 = 1U;
    /* Configure DMA
       b7 DMA transfer start software trigger
       b6 Selection of DMA transfer direction : RAM to SFR
       b5 Specification of transfer data size for DMA transfer : 8 bits
       b4 Pending of DMA transfer
       b3:b0 Selection of DMA start source (IFC03-0) : INTTM00
    */
    DMC0 = 0x42U;
    /* Configure DMA Channel 0 SFR address register : Port 7 */
    DSA0 = 0x07U;
    /* Configure DMA Channel 0 RAM address register : PWM0_7[MAX_TABLE] look-up table */
    DRA0 = (uint16_t)&PWM0_7;
    /* Configure DMA Channel 0 byte count register : 256 */
    DBC0 = 0x0100U;
    /* Clear INTDMA0 interrupt flag */
    DMAIF0 = 0U;
    /* Enable INTDMA0 interrupt */
    DMAMK0 = 0U;
    /* Start DMA Channel 0 operations */
    DST0 = 1U;
}

```

```

/*****
* Function Name : TAU00_Init
* Description : TAU Unit 0 Channel 0 initialization
* Argument : none
* Return Value : none
* Calling Functions : none
*****/
void TAU00_Init(void)
{
    /* Supplies input clock to TAU Unit 0 */
    TAU0EN = 1U;
    /* Configure format of Timer Clock Select register : fClk */
    TPS0 = 0x0000U;
    /* Stop all channels */
    TT0 = 0x0AFFU;
    /* Configure TAU0 Channel 0 in Interval timer mode */
    TMR00 = 0x0000U;
    /* Configure TAU0 Channel 0 period for PWM frequency around 2kHz */
    TDR00 = 0x003EU;
    /* Enable operation (start) trigger of channel 0 */
    TS0 |= 0x0001U;
}

/*****
* Function Name : __near r_dmac0_interrupt
* Description : DMA Channel 0 interrupt service routine, relaunch each PWM period
* Argument : none
* Return Value : none
* Calling Functions : none
*****/
static void __near r_dmac0_interrupt(void)
{
    /* Configure DMA
    b7 DMA transfer start software trigger
    b6 Selection of DMA transfer direction : RAM to SFR
    b5 Specification of transfer data size for DMA transfer : 8 bits
    b4 Pending of DMA transfer
    b3:b0 Selection of DMA start source (IFC03-0) : INTTM00
    */
    DMC0 = 0x42U;
    /* Configure DMA Channel 0 SFR address register : Port 7 */
    DSA0 = 0x07U;
    /* Configure DMA Channel 0 RAM address register : PWM0_7[MAX_TABLE] look-up table */
    DRA0 = (uint16_t)&PWM0_7;
    /* Configure DMA Channel 0 byte count register : 256 */
    DBC0 = 0x0100U;
    /* Start DMA Channel 0 operations */
    DST0 = 1U;
}

```

## 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/>

お問い合わせ先

<http://japan.renesas.com/inquiry>

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。

# Revision Record

Rev.	Date 日付	Description	
		Page	Summary
1.0	2015.12.15	—	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）がありません。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っていると、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。型名が異なる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い合わせください。ご使用に際しては、特定の物質の含有・使用を規制するRoHS指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
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用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

営業お問い合わせ窓口

<http://www.renesas.com>

営業お問い合わせ窓口の住所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 エレクトロニクス株式会社 〒135-0061 東京都江東区豊洲3-2-24 (豊洲フォレシア)

技術的なお問合せおよび資料のご請求は下記へどうぞ。  
総合お問い合わせ窓口：<http://japan.renesas.com/contact/>