

お客様各位

---

## カタログ等資料中の旧社名の扱いについて

---

2010年4月1日を以ってNECエレクトロニクス株式会社及び株式会社ルネサステクノロジが合併し、両社の全ての事業が当社に承継されております。従いまして、本資料中には旧社名での表記が残っておりますが、当社の資料として有効ですので、ご理解の程宜しくお願ひ申し上げます。

ルネサスエレクトロニクス ホームページ (<http://www.renesas.com>)

2010年4月1日

ルネサスエレクトロニクス株式会社

【発行】ルネサスエレクトロニクス株式会社 (<http://www.renesas.com>)

【問い合わせ先】 <http://japan.renesas.com/inquiry>

## ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りが無いことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。  
標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット  
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）  
特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害などを生じさせないようお客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、詳細につきましては製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

注 1. 本資料において使用されている「当社」とは、ルネサスエレクトロニクス株式会社およびルネサスエレクトロニクス株式会社とその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

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

## M32C/85 グループ

### インテリジェント I/O を使用したクロック同期形シリアル通信 2 チャンネルと PWM 出力 4 チャンネル(PWM 周期可変)の生成

#### 1. 要約

この資料はインテリジェント I/O の機能を用いた、クロック同期形シリアル通信 2 チャンネルと PWM 出力 4 チャンネル(PWM 周期可変)の生成を実現する方法を説明します。

#### 2. はじめに

この資料で説明する応用例は、次のマイコン、条件での利用に適用されます。

- ・マイコン :M32C/85 グループ 144 ピン版  
(100 ピン版の場合は、インテリジェント I/O 波形出力ポート 8 本とインテリジェント I/O シリアル通信ポート 6 本中の 5 本が共用となっています。)
- ・システムクロック :30MHz

M32C/85 グループと同様の SFR(周辺機能制御レジスタ)を持つ他の M16C ファミリでも本プログラムを使用することができます。ただし、一部の機能を機能追加等で変更している場合がありますのでマニュアルで確認してください。このアプリケーションノートをご使用に際しては十分な評価を行ってください。

#### 3. 応用例の説明

インテリジェント I/O は、

- ・フリーラン動作を行う 16 ビットベースタイマを 1 本
- ・時間計測または波形生成用 16 ビットレジスタを 8 本
- ・通信用 8 ビットシフトレジスタ 2 本を 2 組

備えています。

PWM 波形を OUTC1j(j=4~7)から出力します。

周期可変、デューティ可変の PWM 波形を出力します。

クロック同期形シリアル通信の送信データを ISTxDi 端子、転送クロックを ISCLKi 端子から出力し、受信データを ISRxDi 端子に入力します。(i=0~1)

表 1 に本応用例での使用端子を示します。

**表 1 使用端子**

用途	ピン番号	端子名	ポート番号
シリアル通信チャンネル 1	34	ISTxD1	P73
	33	ISCLK1	P74
	32	ISRxD1	P75
シリアル通信チャンネル 0	31	ISTxD0	P76
	30	ISCLK0	P77
	29	ISRxD0	P80
周期可変 PWM 出力	14	OUTC14	P140
	13	OUTC15	P141
	12	OUTC16	P142
	11	OUTC17	P143

表 2 ~ 表 4 に機能の選択を示します。本応用例は表に示す項目の中で” ”印の内容を選択した場合の動作説明です。

**表 2 チャンネル j (j=4~7) 反転波形出力モードにおける選択機能**

項目	内容	選択
出力初期値選択	初期値として"L"を出力	
	初期値として"H"を出力	
反転出力機能選択	出力反転しない	
	出力反転する	

**表 3 通信部 0 クロック同期形シリアル I/O モードにおける選択機能**

項目	内容	選択
カウントソース選択		選択出来ません
カウントソース分周比		選択出来ません
転送クロック	内部クロック	
	外部クロック	
転送クロック設定	f8	
	f2n (n=15)	
	ISCLK0 からの入力	
転送フォーマット	LSB ファースト	
	MSB ファースト	
ISRxD, ISTxD 極性切り替え	反転しない	
	反転する	
送信割り込み要因	G0TB レジスタ空	
	送信完了	

表 4 通信部 1 クロック同期形シリアル I/O モードにおける選択機能

項目	内容	選択
カウントソース選択	クロック停止	
	二相入力	
	f1	
カウントソース分周比	2 分周	
	:	
	64 分周	
	分周なし	
転送クロック	内部クロック	
	外部クロック	
転送クロック設定	チャンネル 3 の反転波形出力モードで生成	
	f8	
	f2n	
	ISCLK0 からの入力	
転送フォーマット	LSB ファースト	
	MSB ファースト	
ISRxD,ISTxD 極性切り替え	反転しない	
	反転する	
送信割り込み要因	G1TB レジスタ空	
	送信完了	

(1) チャンネル 0 による通信部 1 のデータ出力

チャンネル 0 を波形生成機能で使します。

動作モードは「通信機能の出力を使用」を選択します。

ISTxD1 端子への通信部 1 のデータ出力が選択されます。

G1PO0 レジスタとベースタイマの一致でベースタイマリセットは行いません。

ベースタイマはフリーランとします。

(2) ISCLK1 の出力設定

チャンネル 1 を波形生成機能で使します。

動作モードは「通信機能の出力を使用」を選択します。

ISCLK1 端子への通信部 1 のクロック出力が選択されます。

(3) チャンネル 3 による通信部 1 の転送クロックの生成

チャンネル 3 の波形生成機能の反転波形出力モードで通信部 1 の転送クロックを生成します。  
 チャンネル 3 波形生成割り込みを用い、その割り込み処理内で G1P03 レジスタを書き換えます。  
 図 1 に G1P03 レジスタの設定値フローを示します。

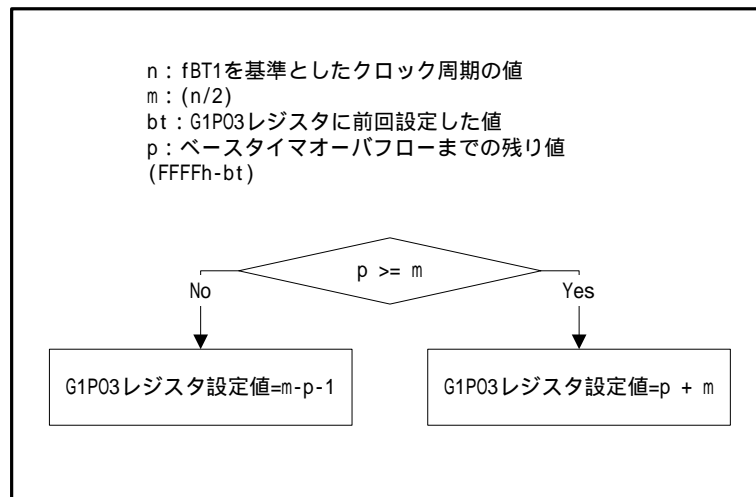


図 1 G1P03 レジスタ設定値フロー

ベースタイマのカウンタソースを fBT1、fBT1 を基準としたクロック周期の値を n とすると、通信部 1 の転送速度（転送クロックの周期）は次式で決定します。

$$\text{転送速度} : \frac{f_{BT1}}{n}$$

(4) チャンネル j (j=4~7)による "L" 幅と "H" 幅(PWM 周期)の変更

チャンネル j の波形生成機能の反転波形出力モードで PWM 出力の"L"幅と"H"幅を設定します。チャンネル j 波形生成割り込みを用い、その割り込み処理内で G1POj レジスタを書き換えます。図 2 に G1PO3 レジスタの設定値フローを示します。

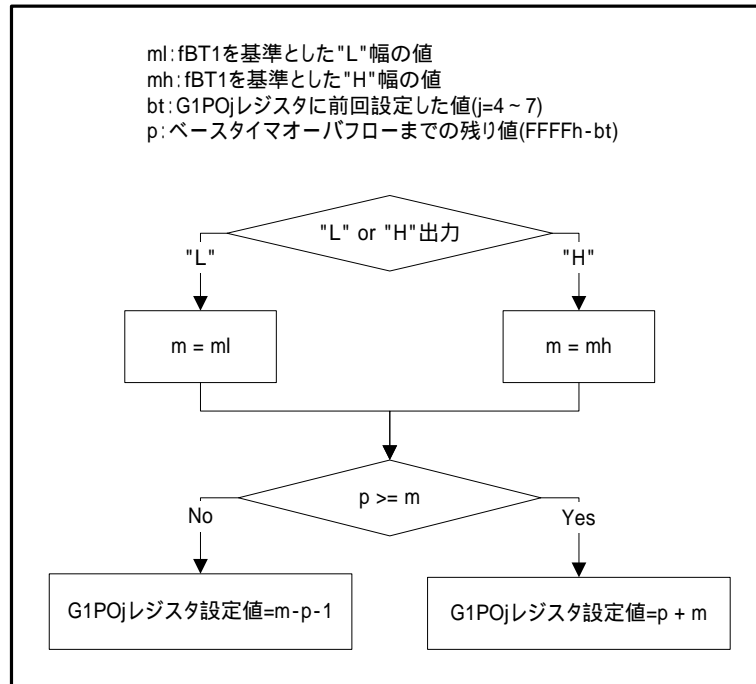


図 2 G1POj レジスタ設定値フロー

ベースタイマのカウンタソースを fBT1、fBT1 を基準とした"L"幅の値を ml、"H"幅の値を mh とすると、PWM 周期は次式で決定します。

$$\text{PWM 周期} = \frac{ml+mh}{fBT1}$$

#### 4. サンプルプログラムの仕様

- ・ 通信部 1 通信速度：約 9600bps
- ・ 通信部 0 通信速度：1Mbps

サンプルプログラムは次の表 5 の定義値を持ち、チャンネル j(j=4~7)PWM 出力の周期と”H”幅の比率を増加させます。可変回数終了後は、初期値に戻します。これを繰り返します。

定義値の設定範囲は、ベースタイマカウンソース周期約 33.3ns、最大割り込み処理時間約 20 μs の条件下です。

G1POj(j=4~7)レジスタに設定する、”H”幅値、”L”幅値は次の通りです。

$$\text{”H”幅値} = \frac{\text{チャンネル j PWM 周期値}}{100} \times \text{チャンネル j ”H”幅比率値}$$

$$\text{”L”幅値} = \text{チャンネル j PWM 周期値} - \text{”H”幅値}$$

表 5 サンプルプログラム定義値

項目	サンプルプログラムの定義値	設定範囲
通信部 1 転送クロック周期値/2	1562	約 601 ~ 約 64984
チャンネル 4 PWM 周期値	3000	G1PO4 レジスタに設定する値が、約 700 ~ 約 64900 になる値 (100 単位)
チャンネル 4 ”H”幅初期比率(%)	60	G1PO4 レジスタに設定する値が、約 601 ~ 約 64984 になる値
チャンネル 4 PWM 周期増加値	120	”
チャンネル 4 ”H”幅増加比率(%)	10	”
チャンネル 5 PWM 周期値	3000	G1PO5 レジスタに設定する値が、約 700 ~ 約 64900 になる値 (100 単位)
チャンネル 5 ”H”幅初期比率(%)	50	G1PO5 レジスタに設定する値が、約 601 ~ 約 64984 になる値
チャンネル 5 PWM 周期増加値	120	”
チャンネル 5 ”H”幅増加比率(%)	10	”
チャンネル 6 PWM 周期値	3000	G1PO6 レジスタに設定する値が、約 700 ~ 約 64900 になる値 (100 単位)
チャンネル 6 ”H”幅初期比率(%)	40	G1PO6 レジスタに設定する値が、約 601 ~ 約 64984 になる値
チャンネル 6 PWM 周期増加値	120	”
チャンネル 6 ”H”幅増加比率(%)	10	”
チャンネル 7 PWM 周期の値	3000	G1PO5 レジスタに設定する値が、約 700 ~ 約 64900 になる値 (100 単位)
チャンネル 7 ”H”幅初期比率(%)	40	G1PO7 レジスタに設定する値が、約 601 ~ 約 64984 になる値
チャンネル 7 PWM 周期増加値	120	”
チャンネル 7 ”H”幅増加比率(%)	10	”
PWM 周期可変回数	3	G1POj(j=4~7)レジスタに設定する値が、約 601 ~ 約 64984 になる値



## 4.1. サンプルプログラムの処理占有率

サンプルプログラムの処理占有率は、表 6 の条件で、次の表 7 の通りです。(測定値は参考値です。)

**表 6 処理占有率測定条件**

システムクロック	30MHz
周辺クロック f2n	n=15
ベースタイマカウントソース周期	1s÷30M=約 33.3ns
ベースタイマオーバーフロー周期	33.3ns×FFFFh=約 2,182μs
PWM 周期と通信部 1 転送クロック周期の 1 周期を生成するための割り込み回数(z とする。)	z=2
通信部 0 の転送クロック	30M÷f2n=1Mbps
通信部 1 の転送クロック	約 9600bps とする。
G1PO3 レジスタに書く値(x とする。)	x=30M÷9600÷z x=約 1562(小数点以下切り捨て)
インテリジェント I/O 波形生成機能 3 割り込み回数	FFFFh÷x=41
インテリジェント I/O 通信部 1 送信割り込み回数	FFFFh÷(x×z×8bit)=2
インテリジェント I/O 通信部 1 受信割り込み回数	"
PWM 周期	99900ns とする。
fBT1 を基準とした PWM 周期 j の値(y とする。) (j=4 ~ 7)	y=99900ns÷33.3ns y=3000
インテリジェント I/O 波形生成機能 j 割り込み回数 (j=4 ~ 7)	FFFFh÷y×z=43

**表 7 処理占有率**

割り込み処理	処理時間 (μs)	割り込み 回数	合計処理時間 (μs)	占有率
インテリジェント I/O 通信部 0 送信割り込み	1.6	168	268.8	12.31%
インテリジェント I/O 通信部 0 受信割り込み	1.6	168	268.8	12.31%
インテリジェント I/O 波形生成機能 3 割り込み (通信部 1 の転送クロック生成)	1.8	41	73.8	3.38%
インテリジェント I/O 通信部 1 送信割り込み	1	2	2	0.09%
インテリジェント I/O 通信部 1 受信割り込み	1	2	2	0.09%
インテリジェント I/O 波形生成機能 4 割り込み	3.2	43	137.6	6.30%
インテリジェント I/O 波形生成機能 5 割り込み	3.2	43	137.6	6.30%
インテリジェント I/O 波形生成機能 6 割り込み	3.2	43	137.6	6.30%
インテリジェント I/O 波形生成機能 7 割り込み	3.2	43	137.6	6.30%
計	19.8	553	1165.8	53%

占有率:  $\frac{\text{合計処理時間}}{\text{ベースタイマオーバーフロー周期}(2,182 \mu\text{s})}$   
 割り込み回数: ベースタイマオーバーフロー周期の割り込み回数

図 3~図 6 にサンプルプログラムのタイミング図を示します。

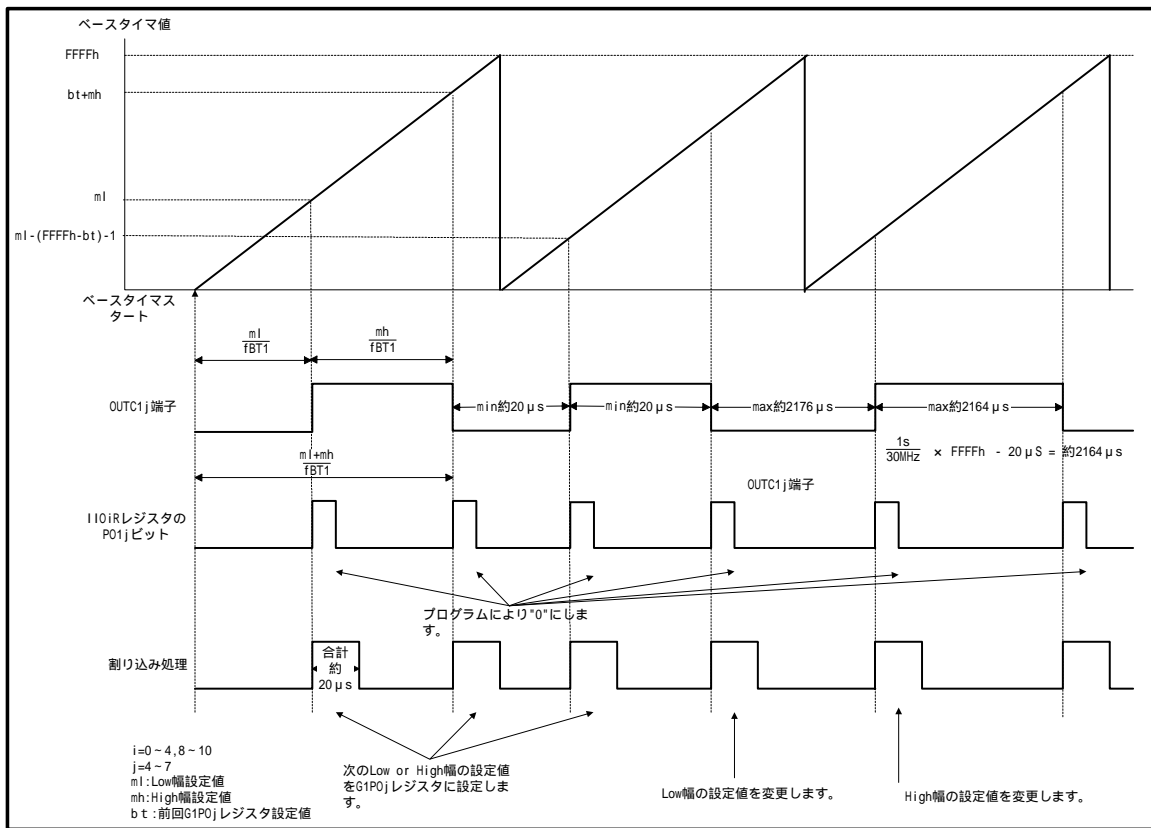


図 3 周期可変、デューティ可変 PWM 波形タイミング図

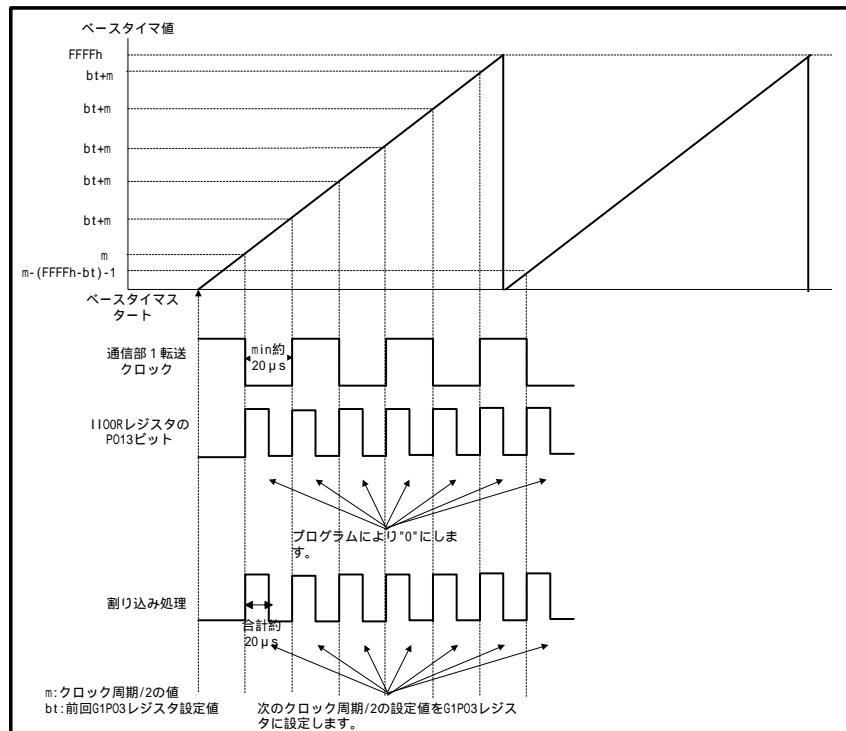


図 4 通信部 1 の転送クロック生成タイミング図

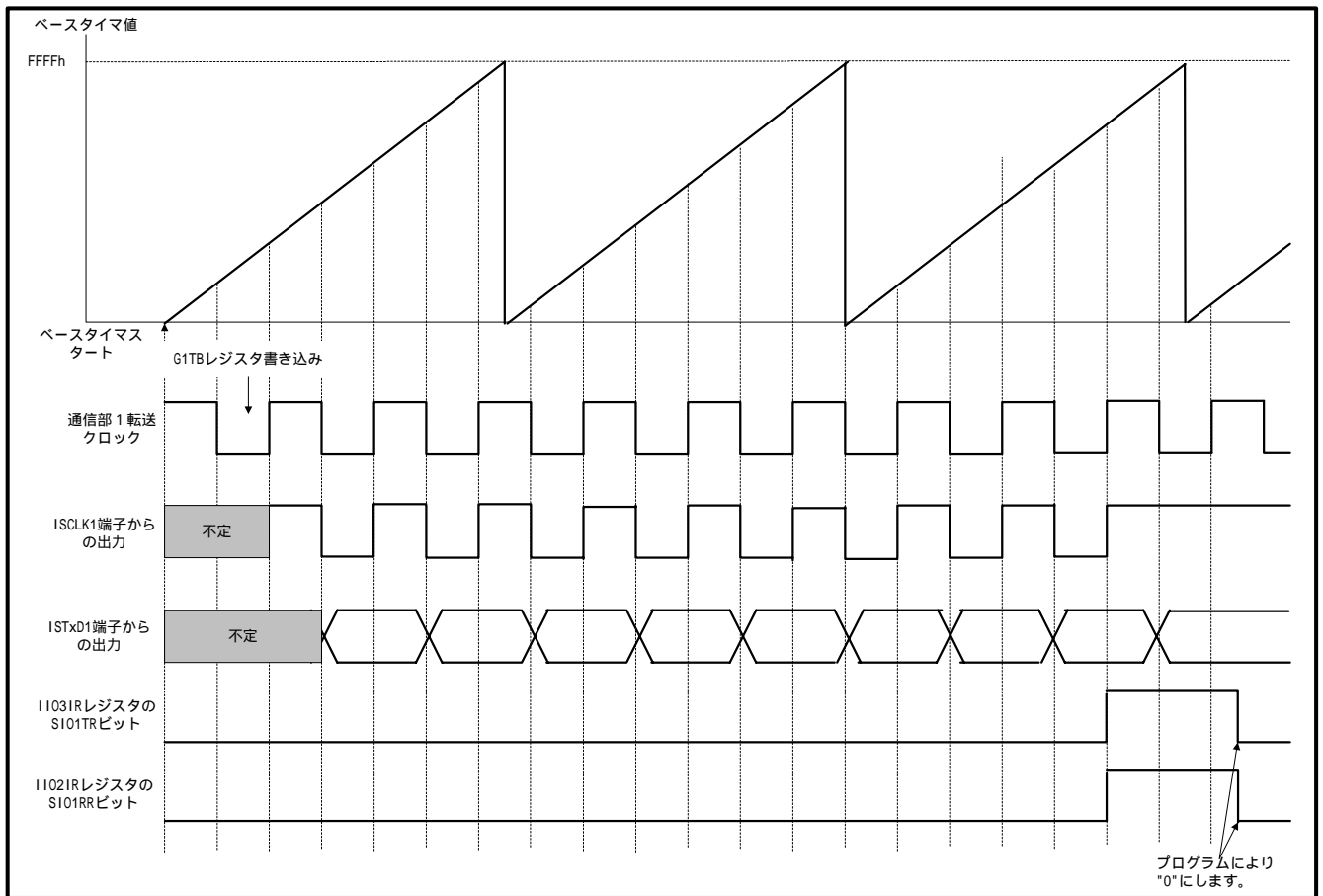


図 5 通信部 1 の送信、受信タイミング図

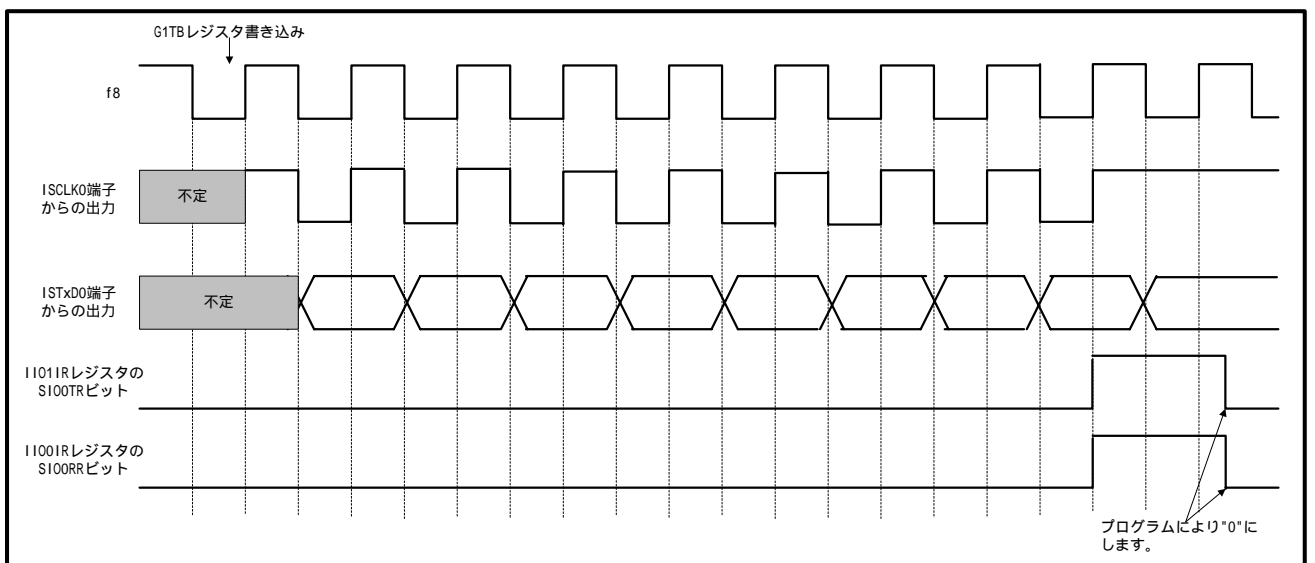


図 6 通信部 0 の送信、受信タイミング図

図 7~図 11 にサンプルプログラムのレジスタ設定フロー図を示します。

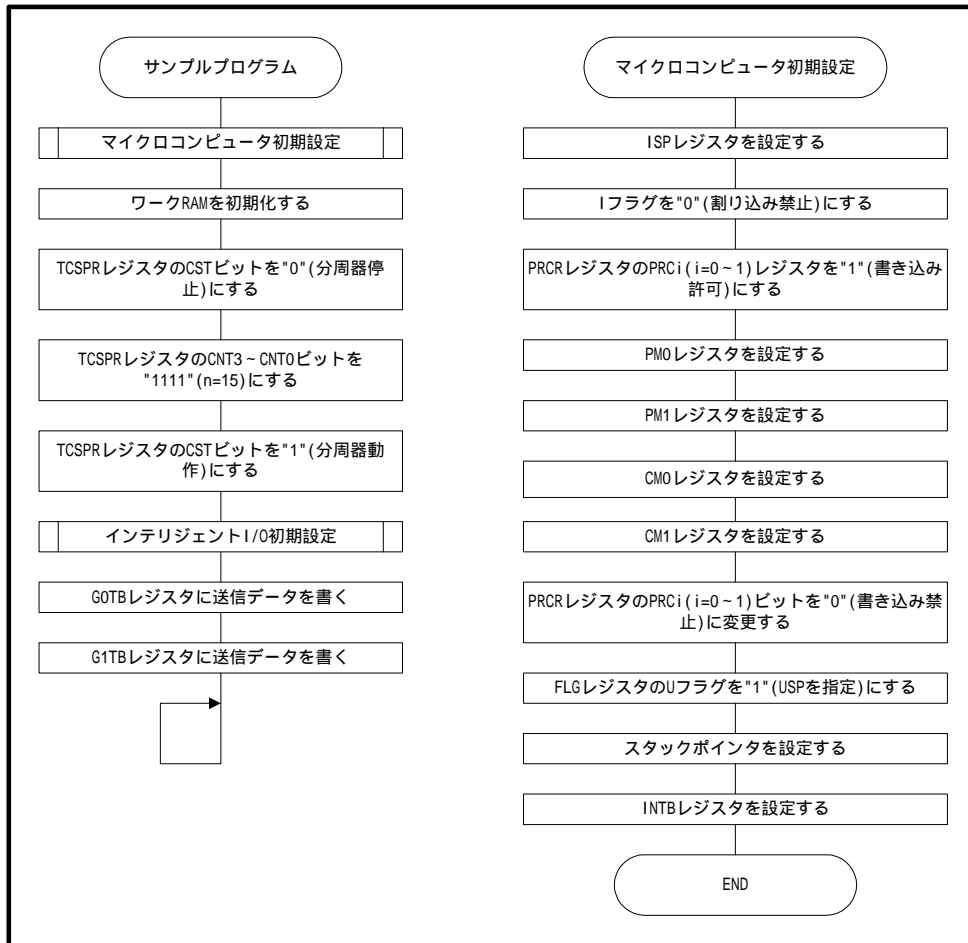


図 7 レジスタ設定フロー図(1)



図 8 レジスタ設定フロー図(2)

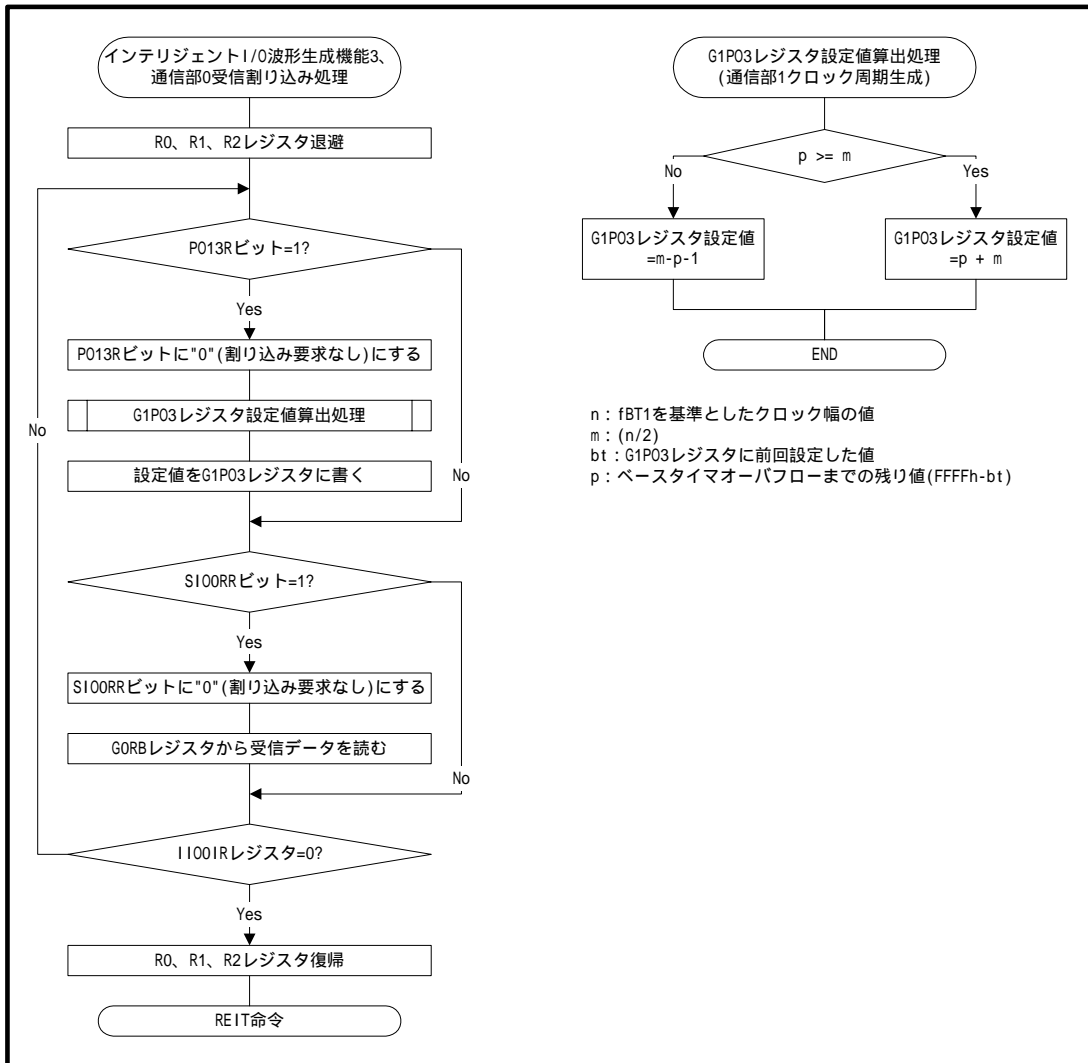


図 9 レジスタ設定フロー図(3)

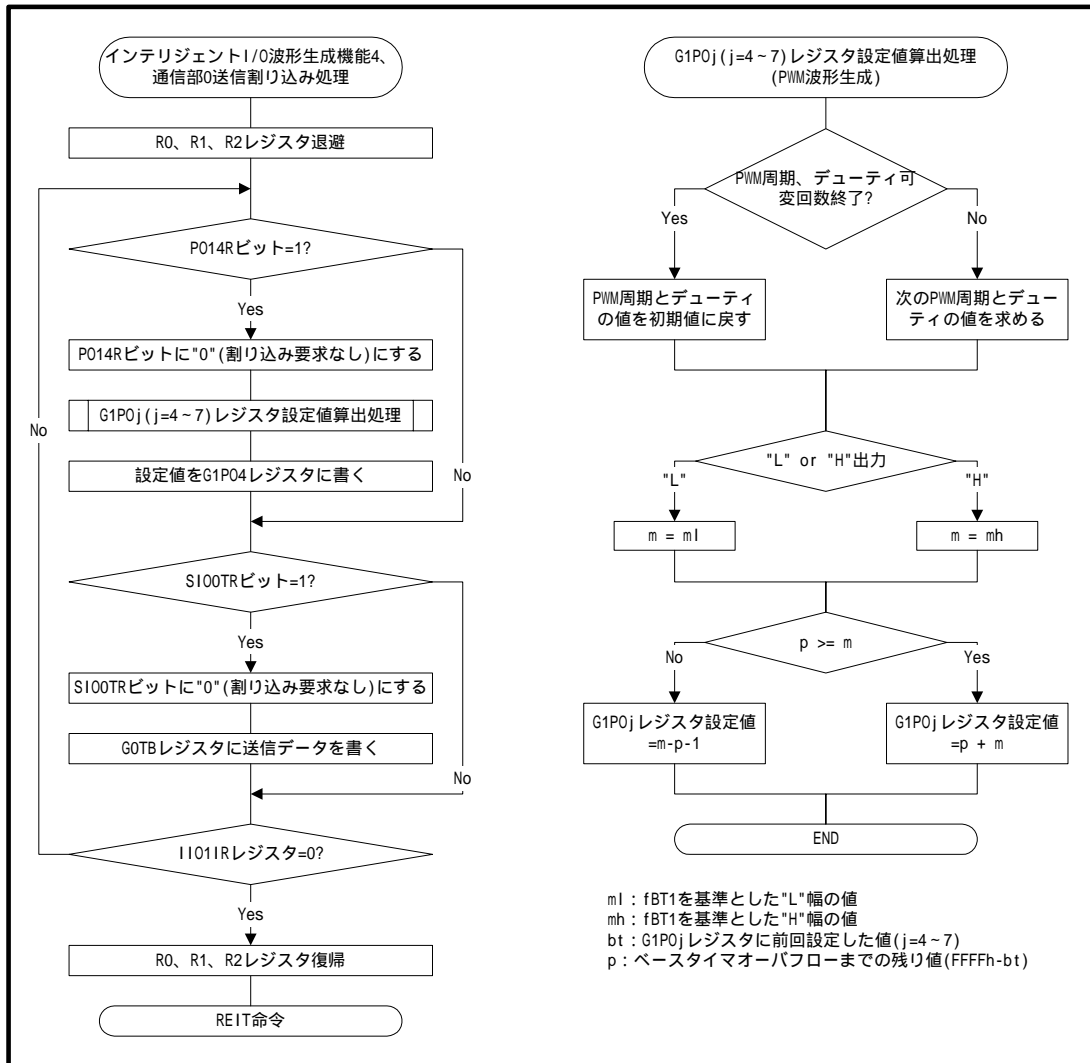


図 10 レジスタ設定フロー図(4)

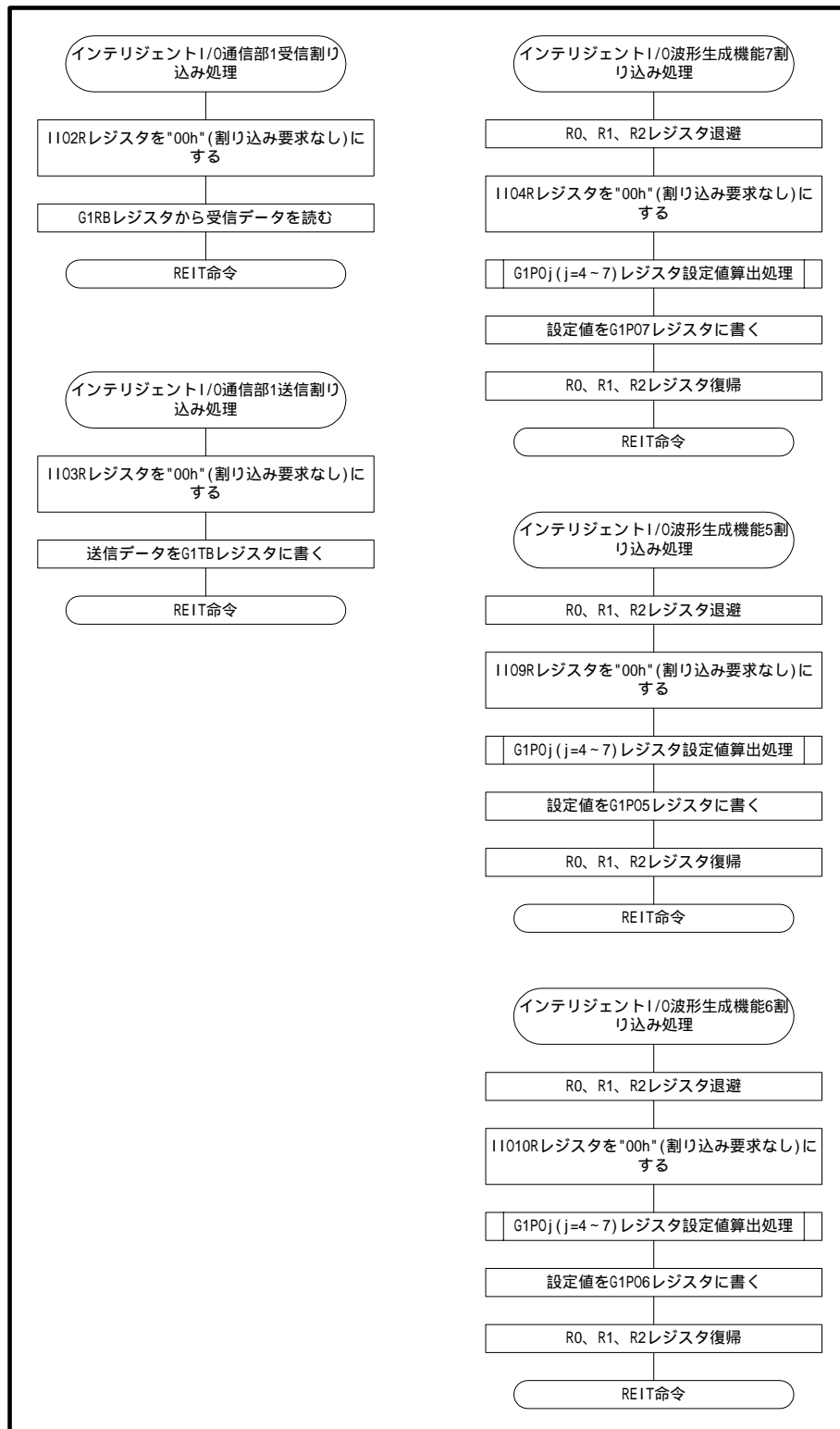


図 11 レジスタ設定フロー図(5)



## 5. サンプルプログラム

```

; "FILE COMMENT"*****
; System Name   : M32C/85 Program Collection
; File Name    : rjj05b0718_src.a30
; Version     : 1.00
; Contents    : Generation of two channels of
;              :   clock synchronous serial communication and
;              :   four channels of PWM output uses Intelligent I/O.
;              :   (PWM cycle changeability)
; Customer   :
; Model      :
; Order      :
; CPU        : M32C/85 Group
; Compiler   : NC308WA(V.5.20 Release 1)
; OS         : Nothing
; Programmer :
; Note       :
;*****
; Copyright,2005 RENESAS TECHNOLOGY CORPORATION
; AND RENESAS SOLUTIONS CORPORATION
;*****
; History    : 2005.2.28 Ver 1.00
; "FILE COMMENT END"*****

;*****
;
;      Include
;*****
;      .LIST      OFF
;      .INCLUDE   sfr32c8586.inc      ;Reads the file that defined SFR
;      .LIST      ON
;
;*****
;      Symbol definition
;*****
RAM_TOP      .EQU   00400H      ;Start address of RAM
ROM_TOP      .EQU   0FFA000H    ;Start address of ROM
VECT_TOP     .EQU   0FFBD00H    ;Start address of variable vector
FIXED_VECT_TOP .EQU  0FFFFDCH   ;Start address of fixed vector

SIO_1_CLK    .EQU   1562        ; Communication Unit 1 SI/O clock pulse cycle / 2

PULSE_CYCLE_4 .EQU   3000        ; ch4 initial pulse cycle
H_DUTY_4     .EQU   60          ; initial high duty rate (%)

```

```

PULSE_UP_4      .EQU    1200      ; up value of pulse cycle
DUTY_UP_4       .EQU     10       ; up rate of high duty

PULSE_CYCLE_5   .EQU    3000      ; ch5 initial pulse cycle
H_DUTY_5        .EQU     50       ; initial high duty rate (%)
PULSE_UP_5      .EQU    1200      ; up value of pulse cycle
DUTY_UP_5       .EQU     10       ; up rate of high duty

PULSE_CYCLE_6   .EQU    3000      ; ch6 initial pulse cycle
H_DUTY_6        .EQU     40       ; initial high duty rate (%)
PULSE_UP_6      .EQU    1200      ; up value of pulse cycle
DUTY_UP_6       .EQU     10       ; up rate of high duty

PULSE_CYCLE_7   .EQU    3000      ; ch7 initial pulse cycle
H_DUTY_7        .EQU     30       ; initial high duty rate (%)
PULSE_UP_7      .EQU    1200      ; up value of pulse cycle
DUTY_UP_7       .EQU     10       ; up rate of high duty

VARIABLE_NUM    .EQU     3

;
;*****
;
; Allocation of work RAM area
;*****
;
SECTION          WORKRAM,DATA
ORG              RAM_TOP

WORKRAM_TOP:

read_buff0:      .BLKB  1          ; Area of receive data buffer for sample
read_buff1:      .BLKB  1          ; Area of receive data buffer for sample
write_buff0:     .BLKB  1          ; Area of send data buffer for sample
write_buff1:     .BLKB  1          ; Area of send data buffer for sample

back_g1po3:     .BLKW  1
back_g1po4:     .BLKW  1
back_g1po5:     .BLKW  1
back_g1po6:     .BLKW  1
back_g1po7:     .BLKW  1

pulse_cyc_4:    .BLKB  1
pulse_cyc_5:    .BLKB  1
pulse_cyc_6:    .BLKB  1
pulse_cyc_7:    .BLKB  1

pulse_w_4:      .BLKW  1
pulse_w_5:      .BLKW  1

```

```

pulse_w_6:      .BLKW  1
pulse_w_7:      .BLKW  1

high_duty_4:    .BLKW  1
high_duty_5:    .BLKW  1
high_duty_6:    .BLKW  1
high_duty_7:    .BLKW  1

                .ALIGN

WORKRAM_END:

;*****
;
;      Stack area
;*****
                .SECTION      STACK,DATA,ALIGN
                .BLKB         30
                .ALIGN

STACK_TOP:

                .BLKB         30
                .ALIGN

ISTACK_TOP:

;
;*****
;
;      Program area
;*****
                .SECTION      PROGRAM, CODE      ; Declares section name and section type
                .ORG          ROM_TOP           ; Declares start address

;=====
;
;      Start up
;=====

RESET:

    LDC      #ISTACK_TOP,ISP      ;set istack pointer

    FCLR    I                    ; Interrupt disabled

; Sets Processor mode, System clock and Main clock division
    MOV.B   #03H, prcr           ;Removes protect
    MOV.B   #10000000B,pm0       ; Single-chip mode
    MOV.B   #00000000B,pm1       ; Flash memory version
    MOV.B   #00001000B,cm0       ; Xcin-Xcout High
    MOV.B   #00100000B,cm1       ; Xin-Xout High
    MOV.B   #00010010B,mcd       ; No division mode

```

```

MOV.B #00H, prcr          ;Protects all registers

LDC #0080h,FLG
LDC #STACK_TOP,SP        ;set stack pointer
LDC #VECT_TOP,INTB       ;set initial value in interrupt table register

; Clears WORKRAM area
MOV.W #0, R0
MOV.W #(WORKRAM_END-WORKRAM_TOP)/2, R3
MOV.W #WORKRAM_TOP, A1
SSTR.W

;
;=====
; Intelligent I/O
;=====
;

MOV.B #00001111B,tcspr    ; n = 15 (f2n)
BSET cst_tcspr

MOV.B #00000010B,cca      ; Communication Clock Select Register
; ||+-----; (CCS1,0):Communication Unit 0 Clock Select Bit
; ||          ; 00:Do not set this value
; ||          ; 01:f1
; ||          ; *10:f2n
; ||          ; 11:f8
; ++-----; (CCS3,2):Communication Unit 1 Clock Select Bit
;           ; *00:Clock output from the channel i (i=1,2,3)
;           ; 01:f1
;           ; 10:f2n
;           ; 11:f8

MOV.B #01111111B,g1bcr0   ; Base Timer Control Register 10
;|+++++-----; (BCK1,0):Count Source Select bit
;|+++++      ; 00:Clock stops
;|+++++      ; 01:Do not set to this value
;|+++++      ; 10:Two-phase pulse signal is applied
;|+++++      ; *11:f1
;|+++++-----; (DIV4,3,2,1,0):Count Source Division Ratio Select bit
;|           ; 0000:Divide-by-2
;|           ; 0001:Divide-by-4
;|           ; 0010:Divide-by-6
;|           ;      :
;|           ;      :
;|           ; 1110:Divide-by-64
;|           ; *1111:No Division
;+-----; (IT):Base Timer Interrupt Select bit

```

```

;          ; *0:Bit 15 overflows
;          ; 1:Bit 14 overflows

MOV.B #0000000B,g1bcr1      ; Base Timer Control Register 11
;|+++++|+-----; (b0):Nothing assigned. When write, set to "0".
;|+++++|+-----; (RST1):Base Timer Reset Cause Select Bit 1
;|+++++|      ; *0:The base timer is not reset by matching with the G1P00 register
;|+++++|      ; 1:The base timer is reset by matching with the G1P00 register
;|+++++|+-----; (RST2):Base Timer Reset Cause Select Bit 2
;|+++++|      ; *0:The base timer is not reset by applying "L"to the INT1 pin
;|+++++|      ; 1:The base timer is reset by applying "L"to the INT1 pin
;|+++++|+-----; (b3):Reserved Bit. Set to "0".
;|+++|+-----; (BTS):Base Timer Start Bit
;|+++|      ; *0:Base timer is reset
;|+++|      ; 1:Base timer starts counting
;|++|+-----; (UD1,0):Counter Increment/Decrement Control Bit
;|++|      ; *00:Counter increment mode
;|++|      ; 01:Counter increment/decrement mode
;|++|      ; 10:Two-phase pulse signal processing mode
;|++|      ; 11:Do not set to this value mode
;|+-----; (b7):Nothing assigned. When write, set to "0".

MOV.B #00000111B,g1pocr0    ; Waveform Generation Register 10
;|+++++|+++-----; (MOD2,1,0):Operation Mode Select Bit
;|+++++|      ; 000:Single waveform output mode
;|+++++|      ; 001:SR waveform output mode
;|+++++|      ; 010:Phase-delayed waveform output mode
;|+++++|      ; *111:Use communication function output
;|+++++|+-----; (b3):Nothing assigned. When write, set to "0".
;|+++|+-----; (IVL):Output Initial Value
;|+++|      ; *0:Outputs "L"as initial value
;|+++|      ; 1:Outputs "H"as initial value
;|++|+-----; (RLD):G1P0j Register Value Reload Timing Select Bit
;|++|      ; *0:Reloads the G1P0j register when write
;|++|      ; 1:Reloads the G1P0j register when the base timer is reset
;|+-----; (BTRE):Base Timer Reset Enable Bit
;|+-----; 0:Enables base timer reset when bit 15 in the base timer overflows
;+-----; (INV):Inverse Output Function Select Bit
;          ; *0:Output is not inversed
;          ; 1:Output is inversed

MOV.B #00000111B,g1pocr1    ; Waveform Generation Register 11
;|+++++|+++-----; (MOD2,1,0):Operation Mode Select Bit
;|+++++|      ; 000:Single waveform output mode
;|+++++|      ; 001:SR waveform output mode
;|+++++|      ; 010:Phase-delayed waveform output mode

```

```

;||||      ; *111:Use communication function output
;||||+-----; (b3):Nothing assigned. When write, set to "0".
;|||+-----; (IVL):Output Initial Value
;|||      ; *0:Outputs "L"as initial value
;|||      ; 1:Outputs "H"as initial value
;||+-----; (RLD):G1POj Register Value Reload Timing Select Bit
;||      ; *0:Reloads the G1POj register when write
;||      ; 1:Reloads the G1POj register when the base timer is reset
;|+-----; (BTRE):Base Timer Reset Enable Bit
;|      ; 0:The BTRE bit is provided in the G1POCR0 register only.
;|      ; Set each bit 6 in the G1POCR1 to G1POCR7 registers to "0".
;+-----; (INV):Inverse Output Function Select Bit
;      ; *0:Output is not inversed
;      ; 1:Output is inversed

MOV.B #00000010B,g1pocr3      ; Waveform Generation Register 13
; | +++-----; (MOD2,1,0):Phase-delayed waveform output mode
; +-----; (RLD):G1POj Register Value Reload Timing Select Bit
;      ; 0:Reloads the G1POj register when write

MOV.B #00000010B,g1pocr4      ; Waveform Generation Register 14
MOV.B #00000010B,g1pocr5      ; Waveform Generation Register 15
MOV.B #00000010B,g1pocr6      ; Waveform Generation Register 16
MOV.B #00000010B,g1pocr7      ; Waveform Generation Register 17
;|||||+++-----; (MOD2,1,0):Operation Mode Select Bit
;|||||      ; 000:Single waveform output mode
;|||||      ; 001:SR waveform output mode
;|||||      ; *010:Phase-delayed waveform output mode
;|||||      ; 111:Use communication function output
;||||+-----; (b3):Nothing assigned. When write, set to "0".
;|||+-----; (IVL):Output Initial Value
;|||      ; *0:Outputs "L"as initial value
;|||      ; 1:Outputs "H"as initial value
;||+-----; (RLD):G1POj Register Value Reload Timing Select Bit
;||      ; *0:Reloads the G1POj register when write
;||      ; 1:Reloads the G1POj register when the base timer is reset
;|+-----; (BTRE):Base Timer Reset Enable Bit
;|      ; 0:The BTRE bit is provided in the G1POCR0 register only.
;|      ; Set each bit 6 in the G1POCR1 to G1POCR7 registers to "0".
;+-----; (INV):Inverse Output Function Select Bit
;      ; *0:Output is not inversed
;      ; 1:Output is inversed

MOV.W #1,g1po3
MOV.W #1,back_g1po3

```

```

MOV.B #0,pulse_cyc_4
MOV.W #H_DUTY_4,high_duty_4
MOV.W #PULSE_CYCLE_4,pulse_w_4
MOV.W #PULSE_CYCLE_4,R0
MOV.W R0,R1
MOV.W #0,R2
DIVU.W #100
MULU.W #H_DUTY_4,R0
SUB.W R0,R1
MOV.W R1,g1po4
MOV.W R1,back_g1po4

```

```

MOV.B #0,pulse_cyc_5
MOV.W #H_DUTY_5,high_duty_5
MOV.W #PULSE_CYCLE_5,pulse_w_5
MOV.W #PULSE_CYCLE_5,R0
MOV.W R0,R1
MOV.W #0,R2
DIVU.W #100
MULU.W #H_DUTY_5,R0
SUB.W R0,R1
MOV.W R1,g1po5
MOV.W R1,back_g1po5

```

```

MOV.B #0,pulse_cyc_6
MOV.W #H_DUTY_6,high_duty_6
MOV.W #PULSE_CYCLE_6,pulse_w_6
MOV.W #PULSE_CYCLE_6,R0
MOV.W R0,R1
MOV.W #0,R2
DIVU.W #100
MULU.W #H_DUTY_6,R0
SUB.W R0,R1
MOV.W R1,g1po6
MOV.W R1,back_g1po6

```

```

MOV.B #0,pulse_cyc_7
MOV.W #H_DUTY_7,high_duty_7
MOV.W #PULSE_CYCLE_7,pulse_w_7
MOV.W #PULSE_CYCLE_7,R0
MOV.W R0,R1
MOV.W #0,R2
DIVU.W #100
MULU.W #H_DUTY_7,R0
SUB.W R0,R1
MOV.W R1,g1po7

```

```

MOV.W  R1,back_g1po7

MOV.B  #00000000B,g1fs      ; Function Select Register 1
MOV.B  #11111011B,g1fe      ; Function Enable Register 1

MOV.B  #00000000B,g0cr      ; SI/O Communication Control Register 0
MOV.B  #00000000B,g1cr      ; SI/O Communication Control Register 1
      ; |+-----; (TE):Transmit Enable Bit
      ; |          ; *0:Transmit disable
      ; |          ; 1:Transmit enable
      ; +-----; (RE):Receive Enable Bit
      ;          ; *0:Receive disable
      ;          ; 1:Receive enable

MOV.B  #10000001B,g0mr      ; SI/O Communication Mode Register 0
MOV.B  #10000001B,g1mr      ; SI/O Communication Mode Register 1
      ;||  |+-----; (GMD1,0):Communication Mode Select Bit
      ;||  |          ; 01:Clock synchronous serial I/O mode
      ;||  +-----; (CKDIR):Internal/External Clock Select Bit
      ;||          ; *0:Internal clock
      ;||          ; 1:External clock
      ;|+-----; (UFORM):Transfer Direction Select Bit
      ;|          ; *0:LSB first
      ;|          ; 1:MSB first
      ;+-----; (IRS):Transmit Interrupt Cause Select Bit
      ;          ; 0:No data in the transmit buffer
      ;          ; *1:Transmission is completed

MOV.B  #00100000B,g0erc      ; SI/O Expansion Receive Control Register 0
MOV.B  #00100000B,g1erc      ; SI/O Expansion Receive Control Register 1
      ; +-----; (RSHTE):Receive Shift Operation Enable Bit
      ;          ; 0:Receive shift operation disabled
      ;          ; *1:Receive shift operation enabled

; Sets Interrupt Enable Register
; (note)If an interrupt request is used for interrupt,
;       set bit 1,2,4 to 7 to "1"after the IRLT bit is set to "1".
MOV.B  #00000001B,iio0ie
MOV.B  #00000001B,iio1ie
MOV.B  #00000001B,iio2ie
MOV.B  #00000001B,iio3ie
MOV.B  #00000001B,iio4ie
MOV.B  #00000001B,iio9ie
MOV.B  #00000001B,iio10ie
      ; +-----; (IRLT):Interrupt Request Select Bit
      ;          ; 1:Interrupt request is used for interrupt
  
```



```

; Clears Interrupt Request Register
; Requests no interrupt
AND.B #0,iio0ir
AND.B #0,iio1ir
AND.B #0,iio2ir
AND.B #0,iio3ir
AND.B #0,iio4ir
AND.B #0,iio9ir
AND.B #0,iio10ir

MOV.B #00100101B,iio0ie
; | +-----; (P013E):Intelligent I/O Waveform Generation Function 3 Interrupt Request
; +-----; (SI00RE):Intelligent I/O Communication Unit 0 Receive Interrupt Request
MOV.B #00100101B,iio1ie
; | +-----; (P014E):Intelligent I/O Waveform Generation Function 4 Interrupt Request
; +-----; (SI00TE):Intelligent I/O Communication Unit 0 Transmit Interrupt Request
MOV.B #00100001B,iio2ie
; +-----; (SI01RE):Intelligent I/O Communication Unit 1 Receive Interrupt Request
MOV.B #00100001B,iio3ie
; +-----; (SI01TE):Intelligent I/O Communication Unit 1 Transmit Interrupt Request
MOV.B #00000101B,iio4ie
; +-----; (P017E):Intelligent I/O Waveform Generation Function 7 Interrupt Request
MOV.B #00000011B,iio9ie
; +-----; (P015E):Intelligent I/O Waveform Generation Function 5 Interrupt Request
MOV.B #00000011B,iio10ie
; +-----; (P016E):Intelligent I/O Waveform Generation Function 6 Interrupt Request

; Sets interrupt priority level
MOV.B #3,iio0ic
MOV.B #3,iio1ic
MOV.B #3,iio2ic
MOV.B #3,iio3ic
MOV.B #3,iio4ic
MOV.B #3,iio9ic
MOV.B #3,iio10ic

; Port setting
MOV.B #00001111B,ps8 ; Function Select Register A8
; |||+-----; (PS8_0):1:P140=OUTC14
; ||+-----; (PS8_1):1:P141=OUTC15
; |+-----; (PS8_2):1:P142=OUTC16
; +-----; (PS8_3):1:P143=OUTC17

MOV.B #00000000B,ps2 ; Function Select Register A2
; +-----; (PS2_0):0:P80=I/O port

```

```

BCLR    pd8_0                ; p80: ISRxDO input

MOV.B   #00011000B,psc      ; Function Select Register C
; | | +-----; (PSC_3):1:P73=ISTxD1
; | +-----; (PSC_4):1:P74=ISCLK1
; +-----; (PSC_6):0:Selected by the PSD1_6 bit

MOV.B   #00000000B,psl1     ; Function Select Register B1
; | | +-----; (PSL1_3):0:Selected by the PSC_3 bit
; | | +-----; (PSL1_4):0:Selected by the PSC_4 bit
; | +-----; (PSL1_6):0:Selected by the PSC_6 bit
; +-----; (PSL1_7):0:P77=ISCLK0 output

MOV.B   #11011000B,ps1     ; Function Select Register A1
; | | +-----; (PS1_3):1:Selected by the PSL1_3 bit
; | | +-----; (PS1_4):1:Selected by the PSL1_4 bit
; | +-----; (PS1_6):1:Selected by the PSL1_6 bit
; +-----; (PS1_7):1:Selected by the PSL1_7 bit

BCLR    pd7_5                ; P75: ISRxD1 input

MOV.B   #00000011B,psd1    ; Function Select Register D1
; +-----; (PSD1_6):0:P76=ISTxD0

MOV.B   #00000000B,ips     ; Input Function Select Register
; | +-----; (IPS0):0:P77=ISCLK0, P80=ISRxDO
; +-----; (IPS1):0:P74=ISCLK1, P75=ISRxD1

BSET    bts_g1bcr1         ; Base timer starts counting

FSET    I                    ; Interrupt enabled

MOV.B   #00110000B,g0cr    ; SI/O Communication Control Register 0
MOV.B   #00110000B,g1cr    ; SI/O Communication Control Register 1
; | +-----; (TE):Transmit Enable Bit
; |           ; 0:Transmit disable
; |           ; *1:Transmit enable
; +-----; (RE):Receive Enable Bit
;           ; 0:Receive disable
;           ; *1:Receive enable

MOV.B   #55H,write_buff0
MOV.B   write_buff0,g0tb    ; Communication Unit 0 data write

MOV.B   #0AAH,write_buff1
MOV.B   write_buff1,g1tb    ; Communication Unit 1 data write
;

```

MAIN:

JMP MAIN

;

=====

; I100 interrupt processing program

=====

I100\_INT:

PUSHM R0,R1,R2

I100\_00:

BTST po13r

JZ I100\_02

BCLR po13r ; Interrupt request clear

; Communication Unit 1 SI/O clock generation

MOV.W back\_g1po3,R0

MOV.W #SIO\_1\_CLK,R1

MOV.W #0FFFFH,R2

SUB.W R0,R2

CMP.W R1,R2

JGEU ?+

SUB.W #1,R1

SUB.W R2,R1

JMP I100\_01

?:

ADD.W R0,R1

I100\_01:

MOV.W R1,g1po3

MOV.W R1,back\_g1po3

I100\_02:

BTST sio0rr

JZ ?+

BCLR sio0rr ; Interrupt request clear

MOV.B g0rb,read\_buff0 ; Read reception data

?:

AND.B #24H,iio0ir

JNZ I100\_00

```

    POPM    R0,R1,R2
    REIT
;
;=====
;    I101 interrupt processing program
;=====
I101_INT:
    PUSHM  R0,R1,R2

I101_00:
    BTST   po14r
    JZ     I101_10

    BCLR   po14r           ; Interrupt request clear

; ch4 pulse generation
    INC.B  pulse_cyc_4
    CMP.B  #VARIABLE_NUM*2+1,pulse_cyc_4
    JLEU   ?+

    MOV.B  #0,pulse_cyc_4
    MOV.W  #PULSE_CYCLE_4,pulse_w_4
    MOV.W  #H_DUTY_4,high_duty_4
    JMP    I101_01

?:
    BTST   0,pulse_cyc_4   ; High output ?
    JNZ    ?+              ; Yes jump

    ADD.W  #DUTY_UP_4,high_duty_4
    ADD.W  #PULSE_UP_4,pulse_w_4
I101_01:
    MOV.W  pulse_w_4,R0
    MOV.W  R0,R1
    MOV.W  #0,R2
    DIVU.W #100
    MULU.W high_duty_4,R0
    SUB.W  R0,R1
    MOV.W  R1,R0           ; Low width->R0
    JMP    I101_02

?:
    MOV.W  pulse_w_4,R0
    MOV.W  #0,R2
    DIVU.W #100
    MULU.W high_duty_4,R0   ; High width->R0

I101_02:

```

```

MOV.W  back_g1po4,R1
MOV.W  #0FFFFH,R2

SUB.W  R1,R2
CMP.W  R0,R2
JGEU   ?+

SUB.W  #1,R0
SUB.W  R2,R0
JMP    I101_03

?:
ADD.W  R1,R0

I101_03:
MOV.W  R0,g1po4
MOV.W  R0,back_g1po4

I101_10:
BTST   sio0tr
JZ     ?+

BCLR   sio0tr           ; Interrupt request clear
INC.B  write_buff0
MOV.B  write_buff0,g0tb ; Communication Unit 0 data write

?:
AND.B  #24H,iio1ir
JNZ   I101_00

POPM   R0,R1,R2
REIT

;
;=====
;      I102 interrupt processing program
;=====
I102_INT:
AND.B  #0,iio2ir           ; Interrupt request clear

MOV.B  g1rb,read_buff1     ; Read reception data
REIT

;
;=====
;      I103 interrupt processing program
;=====
I103_INT:
AND.B  #0,iio3ir           ; Interrupt request clear

```

```

    INC.B  write_buff1
    MOV.B  write_buff1,g1tb      ; Communication Unit 0 data write
    REIT

;
;=====
;    I104 interrupt processing program
;=====
I104_INT:
    PUSHM  R0,R1,R2

    AND.B  #0,iio4ir           ; Interrupt request clear

; ch7 pulse generation
    INC.B  pulse_cyc_7
    CMP.B  #VARIABLE_NUM*2+1,pulse_cyc_7
    JLEU   ?+

    MOV.B  #0,pulse_cyc_7
    MOV.W  #PULSE_CYCLE_7,pulse_w_7
    MOV.W  #H_DUTY_7,high_duty_7
    JMP    I104_01

?:
    BTST   0,pulse_cyc_7       ; High output ?
    JNZ    ?+                  ; Yes jump

    ADD.W  #DUTY_UP_7,high_duty_7
    ADD.W  #PULSE_UP_7,pulse_w_7
I104_01:
    MOV.W  pulse_w_7,R0
    MOV.W  R0,R1
    MOV.W  #0,R2
    DIVU.W #100
    MULU.W high_duty_7,R0
    SUB.W  R0,R1
    MOV.W  R1,R0              ; Low width->R0
    JMP    I104_02

?:
    MOV.W  pulse_w_7,R0
    MOV.W  #0,R2
    DIVU.W #100
    MULU.W high_duty_7,R0    ; High width->R0

I104_02:
    MOV.W  back_g1po7,R1
    MOV.W  #0FFFFH,R2

```

```

SUB.W  R1,R2
CMP.W  R0,R2
JGEU   ?+

SUB.W  #1,R0
SUB.W  R2,R0
JMP    I104_03

?:
ADD.W  R1,R0

I104_03:
MOV.W  R0,g1po7
MOV.W  R0,back_g1po7

POPM   R0,R1,R2
REIT

;
;=====
;      I109 interrupt processing program
;=====
I109_INT:
PUSHM  R0,R1,R2

AND.B  #0,iio9ir          ; Interrupt request clear

; ch5 pulse generation
INC.B  pulse_cyc_5
CMP.B  #VARIABLE_NUM*2+1,pulse_cyc_5
JLEU   ?+

MOV.B  #0,pulse_cyc_5
MOV.W  #PULSE_CYCLE_5,pulse_w_5
MOV.W  #H_DUTY_5,high_duty_5
JMP    I109_01

?:
BTST   0,pulse_cyc_5      ; High output ?
JNZ    ?+                 ; Yes jump

ADD.W  #DUTY_UP_5,high_duty_5
ADD.W  #PULSE_UP_5,pulse_w_5

I109_01:
MOV.W  pulse_w_5,R0
MOV.W  R0,R1
MOV.W  #0,R2
DIVU.W #100
MULU.W high_duty_5,R0

```

```

SUB.W  R0,R1
MOV.W  R1,R0          ; Low width->R0
JMP    I109_02

?:
MOV.W  pulse_w_5,R0
MOV.W  #0,R2
DIVU.W #100
MULU.W high_duty_5,R0 ; High width->R0

I109_02:
MOV.W  back_g1po5,R1
MOV.W  #0FFFFH,R2

SUB.W  R1,R2
CMP.W  R0,R2
JGEU   ?+

SUB.W  #1,R0
SUB.W  R2,R0
JMP    I109_03

?:
ADD.W  R1,R0

I109_03:
MOV.W  R0,g1po5
MOV.W  R0,back_g1po5

POPM   R0,R1,R2
REIT

;
;=====
;      I1010 interrupt processing program
;=====
I1010_INT:
PUSHM  R0,R1,R2

AND.B  #0,iio10ir      ; Interrupt request clear

; ch6 pulse generation
INC.B  pulse_cyc_6
CMP.B  #VARIABLE_NUM*2+1,pulse_cyc_6
JLEU   ?+

MOV.B  #0,pulse_cyc_6
MOV.W  #PULSE_CYCLE_6,pulse_w_6
MOV.W  #H_DUTY_6,high_duty_6

```



```

        JMP      I1010_01
?:
        BTST    0,pulse_cyc_6          ; High output ?
        JNZ     ?+                     ; Yes jump

        ADD.W   #DUTY_UP_6,high_duty_6
        ADD.W   #PULSE_UP_6,pulse_w_6
    
```

```

I1010_01:
        MOV.W   pulse_w_6,R0
        MOV.W   R0,R1
        MOV.W   #0,R2
        DIVU.W  #100
        MULU.W  high_duty_6,R0
        SUB.W   R0,R1
        MOV.W   R1,R0                 ; Low width->R0
        JMP     I1010_02
    
```

```

?:
        MOV.W   pulse_w_6,R0
        MOV.W   #0,R2
        DIVU.W  #100
        MULU.W  high_duty_6,R0       ; High width->R0
    
```

```

I1010_02:
        MOV.W   back_g1po6,R1
        MOV.W   #0FFFFH,R2

        SUB.W   R1,R2
        CMP.W   R0,R2
        JGEU   ?+

        SUB.W   #1,R0
        SUB.W   R2,R0
        JMP     I1010_03
    
```

```

?:
        ADD.W   R1,R0
    
```

```

I1010_03:
        MOV.W   R0,g1po6
        MOV.W   R0,back_g1po6

        POPM   R0,R1,R2
        REIT
    
```

```

;
;=====
;      Dummy interrupt processing program
    
```

```

;=====
dummy:
    REIT
;
;*****
;
;    Setting of variable vector table
;*****
;
;SECTION    VECT, ROMDATA
;ORG        VECT_TOP+(44*4)
;
;LWORD     I100_INT    ; Intelligent I/O Interrupt 0, CAN3
;LWORD     I101_INT    ; Intelligent I/O Interrupt 1, CAN4
;LWORD     I102_INT    ; Intelligent I/O Interrupt 2
;LWORD     I103_INT    ; Intelligent I/O Interrupt 3
;LWORD     I104_INT    ; Intelligent I/O Interrupt 4
;LWORD     dummy       ; CAN 5
;LWORD     dummy       ; dummy
;LWORD     dummy       ; dummy
;LWORD     dummy       ; Intelligent I/O Interrupt 8
;LWORD     I109_INT    ; Intelligent I/O Interrupt 9, CAN0
;LWORD     I1010_INT   ; Intelligent I/O Interrupt 10, CAN1
;
;*****
;
;    Setting of fixed vector table
;*****
;
;SECTION    F_VECT, ROMDATA
;ORG        FIXED_VECT_TOP
;
;LWORD     dummy       ;Undefined instruction
;LWORD     dummy       ;Overflow
;LWORD     dummy       ;BRK instruction execution
;LWORD     dummy       ;Address match
;LWORD     dummy       ;
;LWORD     dummy       ;Watchdog timer
;LWORD     dummy       ;
;LWORD     dummy       ;NMI
;LWORD     RESET       ;Reset
;
;
;END
    
```

## 6. 参考ドキュメント

ハードウェアマニュアル

M32C/85 グループハードウェアマニュアル Rev.1.0

(最新版をルネサス テクノロジホームページから入手してください。)

テクニカルアップデート/テクニカルニュース

(最新の情報をルネサス テクノロジホームページから入手してください。)

## 7. ホームページとサポート窓口

ルネサス テクノロジホームページ

<http://www.renesas.com/jpn/>

M16C ファミリー MCU 技術サポート窓口

E-mail: [csc@renesas.com](mailto:csc@renesas.com)

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2005.02.28	-	初版発行

### 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留意ください。

### 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 テクノロジー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 テクノロジー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 テクノロジー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 テクノロジーは、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 テクノロジー半導体製品のご購入に当たりますとは、事前にルネサス テクノロジー、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 テクノロジーホームページ(<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、慎重に制作したのですが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 テクノロジー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 テクノロジーは、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 テクノロジー、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 テクノロジー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 テクノロジー、ルネサス販売または特約店までご照会ください。