

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。

H8/300H Tiny シリーズ

赤外光トランシーバによる文字送受信例

要旨

赤外光トランシーバを接続したマイコンと、パソコンの IrDA 通信ポートを使い、物理層による文字の送受信を行う。

動作確認デバイス

H8/300H Tiny シリーズ H8/36014CPU

目次

1. 仕様	2
2. 使用機能説明	5
3. 動作原理	6
4. ソフトウェア説明	7
5. フローチャート	11
6. プログラムリスト	19

1. 仕様

- (1) 図 1 に赤外光トランシーバを使用したデータ送受信のハードウェア構成を示します。
- (2) 本タスク例では、パソコンの IrDA ポートから送信された文字を赤外光トランシーバで受信し、受信した文字のアスキーコードをインクリメントし、その文字を赤外線トランシーバを経由してパソコンに返信します。パソコンのモニタにはインクリメントされたアスキーコードに対応した文字が表示されます。
- (3) 本タスク例における H8/36014 の動作電圧(Vcc)およびアナログ電源電圧(AVcc)は 3.3V、OSC クロック周波数は 10MHz です。

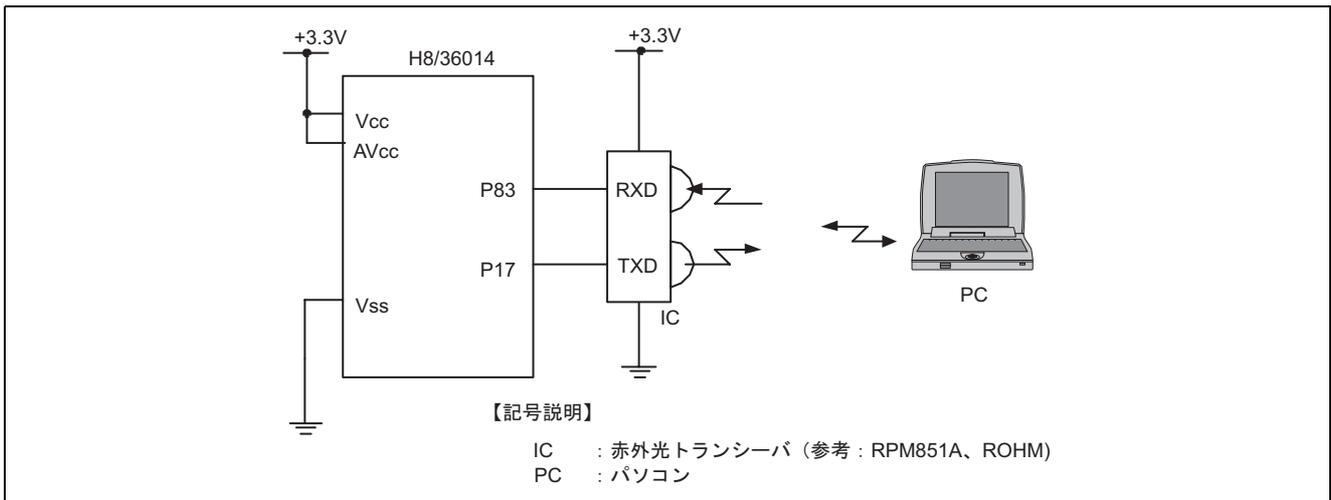


図 1 ハードウェア構成

(4) 本タスクで使用している赤外線トランシーバは、ROHM 製赤外光トランシーバ(型名：RPM851A)です。

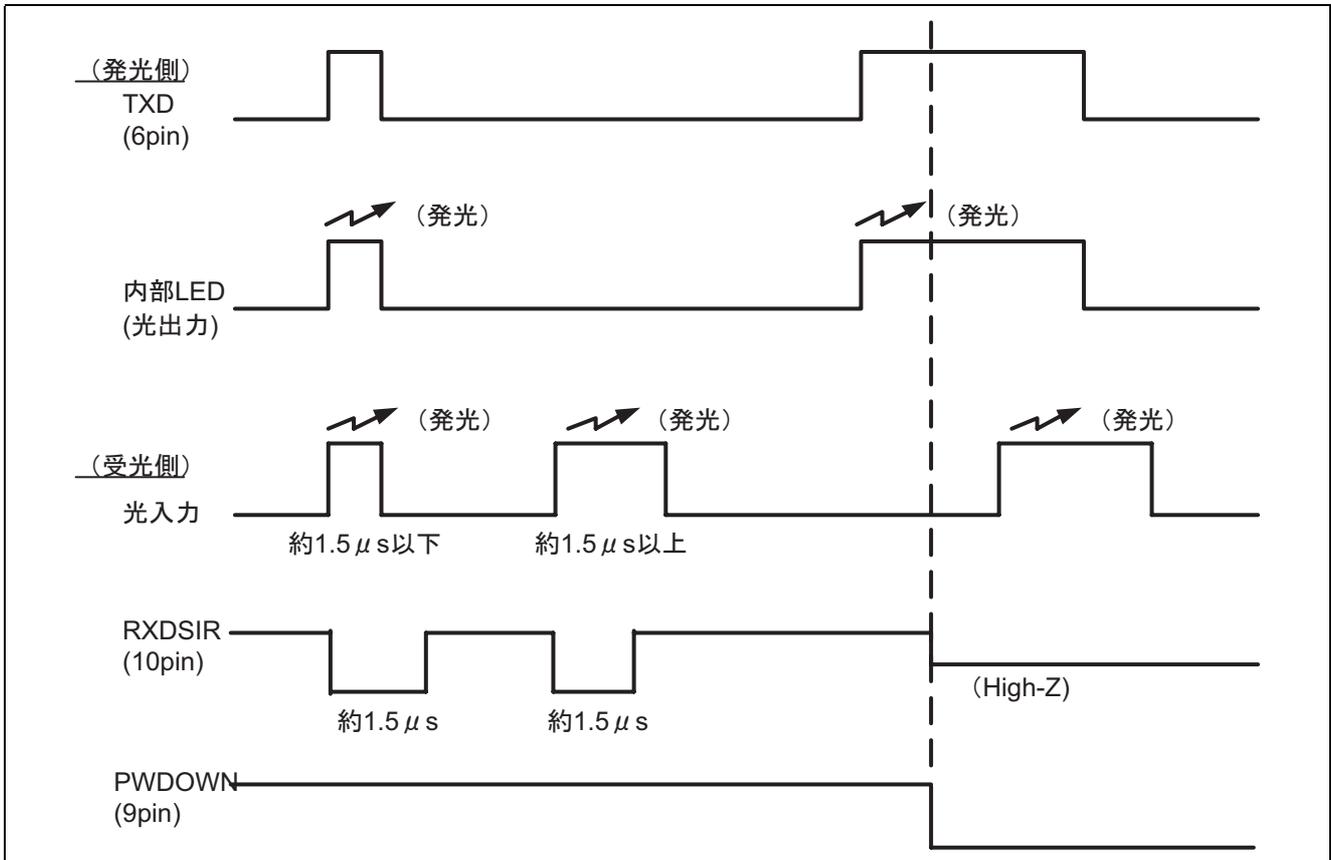


図2 赤外光トランシーバ動作タイミングチャート例

(a) RPM851A の特長を示します。

IrDA Ver.1.0 対応。

受信待機時低消費電流。(typ.220 μ A)

パワーダウン制御機能によりバッテリー使用機器に最適。

2.7V ~ 5.5V の電源電圧範囲。

上面、側面、双方の表面実装に対応するパッケージを採用。

(b) 図2 に赤外光トランシーバの動作タイミングチャートを示します。

(c) 図3 に赤外光トランシーバのブロック図及び応用回路例を示します。

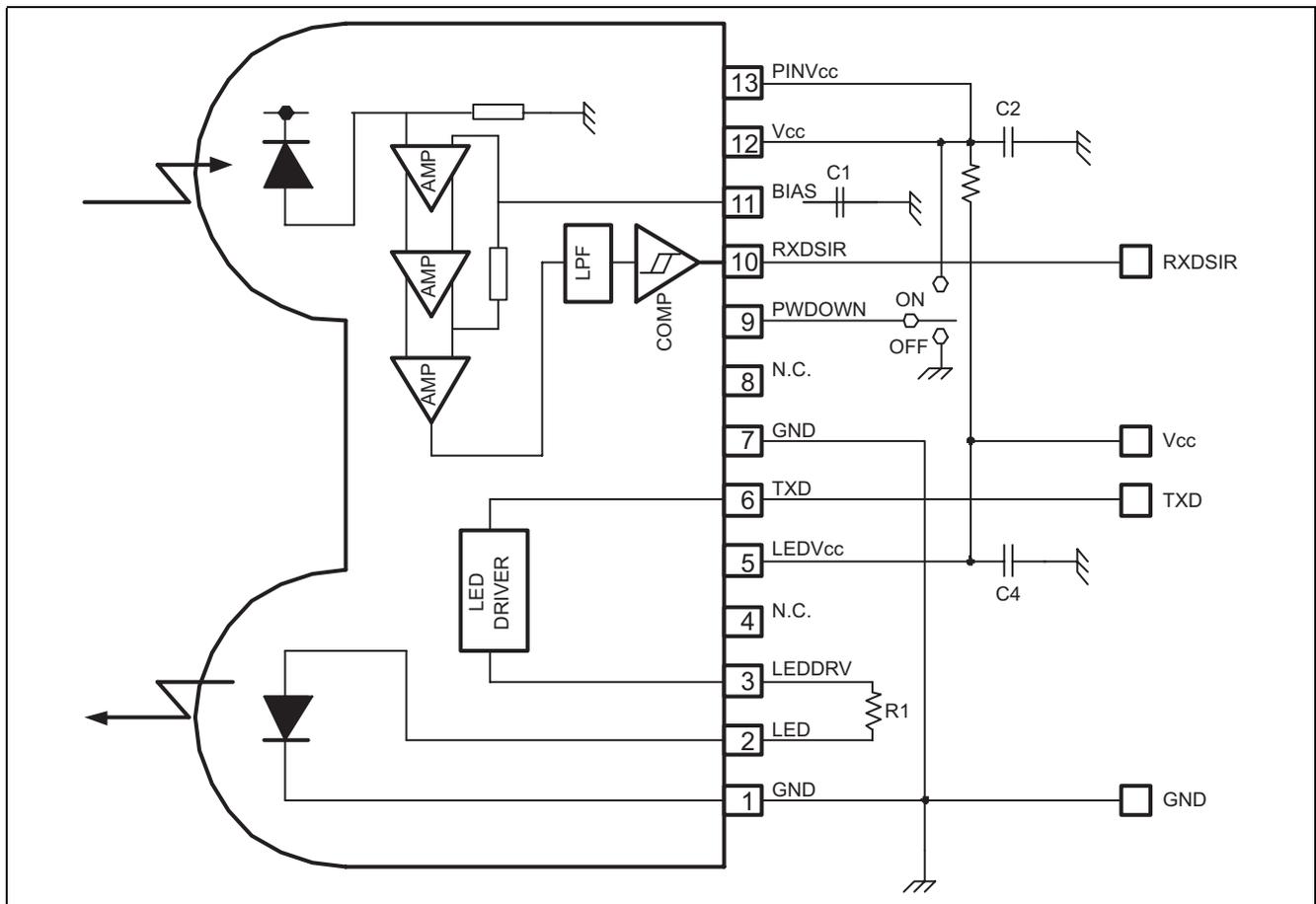


図3 赤外光トランシーバのブロック図及び応用回路例

(1) 本応用例の動作は以下の通りです。

パソコンのターミナルソフトによる IrDA1.0 に準拠した波形を用い、物理層による通信を行います。

(a) パソコンのキーボードから、例えばアスキーコード「'H31」の文字"1"を入力します。

(b) パソコンの IrDA 通信ポートから、変調された信号が LSB ファーストで送信されます。

(c) マイコンに接続した赤外光トランシーバが、信号を受信、マイコンが復調してデータ「'H31」を取り出します。

(d) マイコンで「'H32」にインクリメントした後、変調を行い即時に赤外光トランシーバ経由で返信します。

(e) パソコンの IrDA 通信ポートで受信した信号は復調され、データ「'H32」が取り出されます。アスキーコード「'H32」は文字"2"に相当するのでモニタに"2"が表示されます。

2. 使用機能説明

(1) 図 4 に本タスク例における H8/36014 の使用機能のブロック図を、表 1 に機能割付を示します。

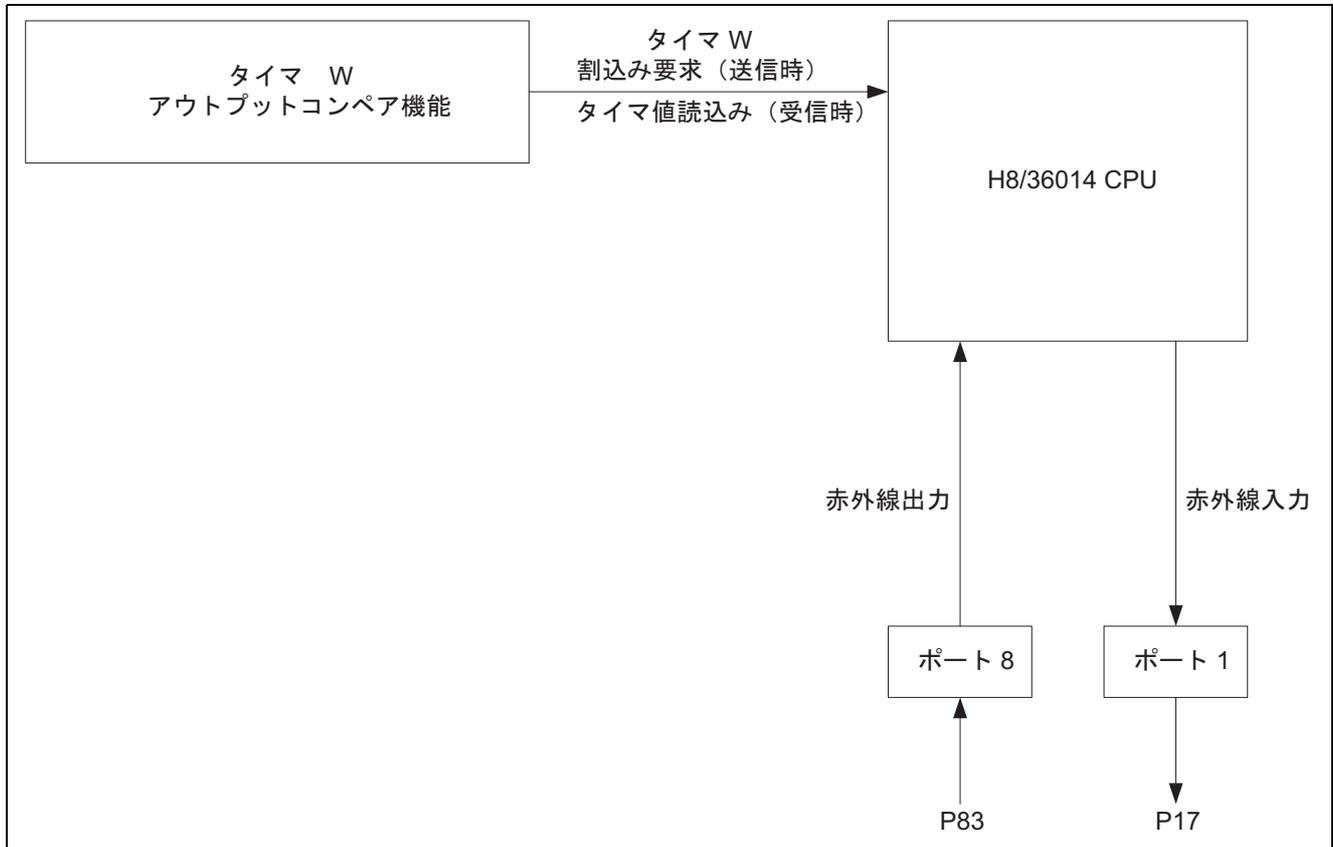


図 4 使用機能ブロック図

表1 機能割付け

使用機能	機能割付け
タイマ W	コンペアマッチ機能を使用してトグル出力を行います。ジェネラルレジスタ B (GRB) に値を設定することにより、出力周波数を変更できます。
ポート 1	ポート 1 の P17 出力端子より、赤外線データを送信します。
ポート 8	ポート 8 の P83 入力端子により、赤外線データを受信します。

3. 動作原理

図 5 にタイマ W を使用した、赤外線通信を行う際の動作原理を示します。図 5 に示すようにタイマ W コンペアマッチ周期ごとにトグル出力させることにより通信します。

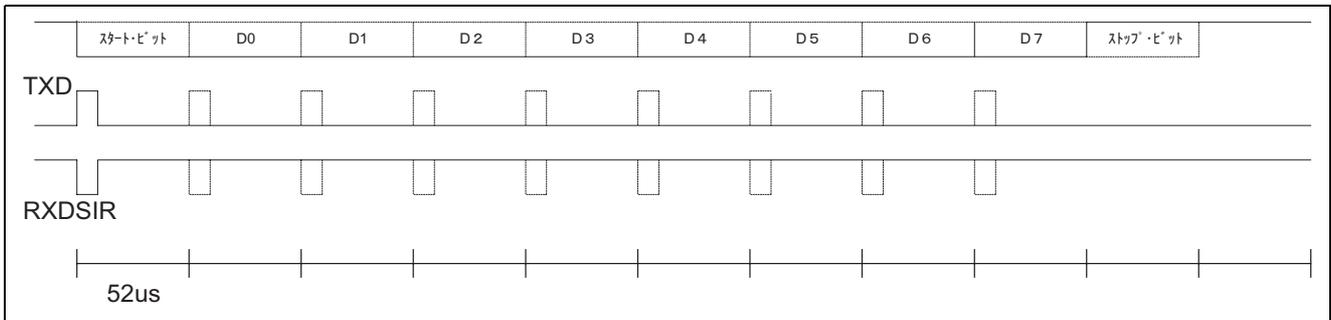


図 5 タイマ W を使用した赤外線通信の動作原理

本タスクでは 19.2kHz の通信を行っているため、1 ビット当たりの信号周期は約 52 μ s となります。

送信時：約 4.5 μ s の間、信号が変化します。負論理 (0 : High、1 : Low) となります。

受信時：RPM851A の制約上 1.5 μ s の間、信号が変化します。正論理 (1 : High、0 : Low) です。

受信ビットの参照は、毎 20 回行っています。

4. ソフトウェア説明

(1) モジュール説明

表 2 に本タスク例におけるモジュール説明を示します。

表 2 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	main	初期設定を行った後、赤外線データ受信処理ルーチンと赤外線データ送信処理ルーチンを交互に呼び出す
赤外線データ受信処理ルーチン	irda_rcv	赤外線を使用し、データを受信する
赤外線データ送信処理ルーチン	irda_snd	赤外線を使用し、データを送信する
W タイマ割り込み処理ルーチン	tmrw	52 μ s のタイマとして使用する

(2) 引数説明

本タスク例では、引数を使用しておりません。

(3) 使用マイコンのレジスタ説明

本タスク例の使用マイコンのレジスタを表 3 に示します。

表 3 使用内部レジスタ説明

レジスタ名	機能説明	アドレス	設定値
TMRW	タイマモードレジスタ W : ジェネラルレジスタの機能やタイマの出力モードの選択	H'FF80	H'80
CTS	カウンタスタート : CTS = 1 のとき、TCNT がカウンタ開始する : CTS = 0 のとき、TCNT がカウンタ停止する	ビット 7	1
TCRW	タイマコントロールレジスタ W : カウンタクロックの選択 : カウンタのクリア条件やタイマの出力レベルの設定	H'FF81	H'82
CCLR	カウンタクリア : CCLR = 1 のとき、コンペアマッチ A によって TCNT がクリアされる	ビット 7	1
CKS2	クロックセレクト : CKS2 = 0、CKS1 = 0、CKS0 = 0 のとき、TCNT 入力クロックをシステムクロックの分周なしのクロックに設定する	ビット 6	0
CKS1		ビット 5	0
CKS0		ビット 4	0
TIERW	タイマインタラプトイネーブルレジスタ W : タイマ W の割り込み要求を制御する	H'FF82	H'00 (初期設定時)
IMIEA	インプットキャプチャ/コンペアマッチ割り込みイネーブル A : IMIEA = 1 のとき、TSRW の IMFA による割り込み要求 IMIA がイネーブルとなる	ビット 0	1
TSRW	割り込み要求ステータスを表示する	H'FF83	H'00
IMFA	インプットキャプチャ/コンペアマッチフラグ A	ビット 0	0
TCNT	タイマカウンタ : システムクロックの 8 分周のクロックを入力とする 16 ビットのアップカウンタ	H'FF86	H'00
GRA	ジェネラルレジスタ A	H'FF88	H'FF

表 3 使用内部レジスタ説明

レジスタ名	機能説明	アドレス	設定値
PCR1	ポートコントロールレジスタ 1 ：ポート 1 の汎用入出力ポートとして使用する端子の入出力をビットごとに選択 PCR1 = H'A0 のとき、 ：P17、P15 端子は汎用出力端子として機能 上記以外は汎用入力端子として機能	H'FFE4	H'A0
PDR1	ポートデータレジスタ 1 ：ポート 1 の汎用入出力ポートデータレジスタ	H'FFD4	H'00
PCR8	ポートコントロールレジスタ 7 ：ポート 8 の汎用入出力ポートとして使用する端子の入出力をビットごとに選択 PCR8 = H'F7 のとき、 ：P83 端子は汎用入力端子として機能 上記以外は汎用出力端子として機能	H'FFEB	H'F7
PDR8	ポートデータレジスタ 8 ：ポート 8 の汎用入出力ポートデータレジスタ	H'FFDB	H'00 (初期設定時)

(4)使用 RAM 説明

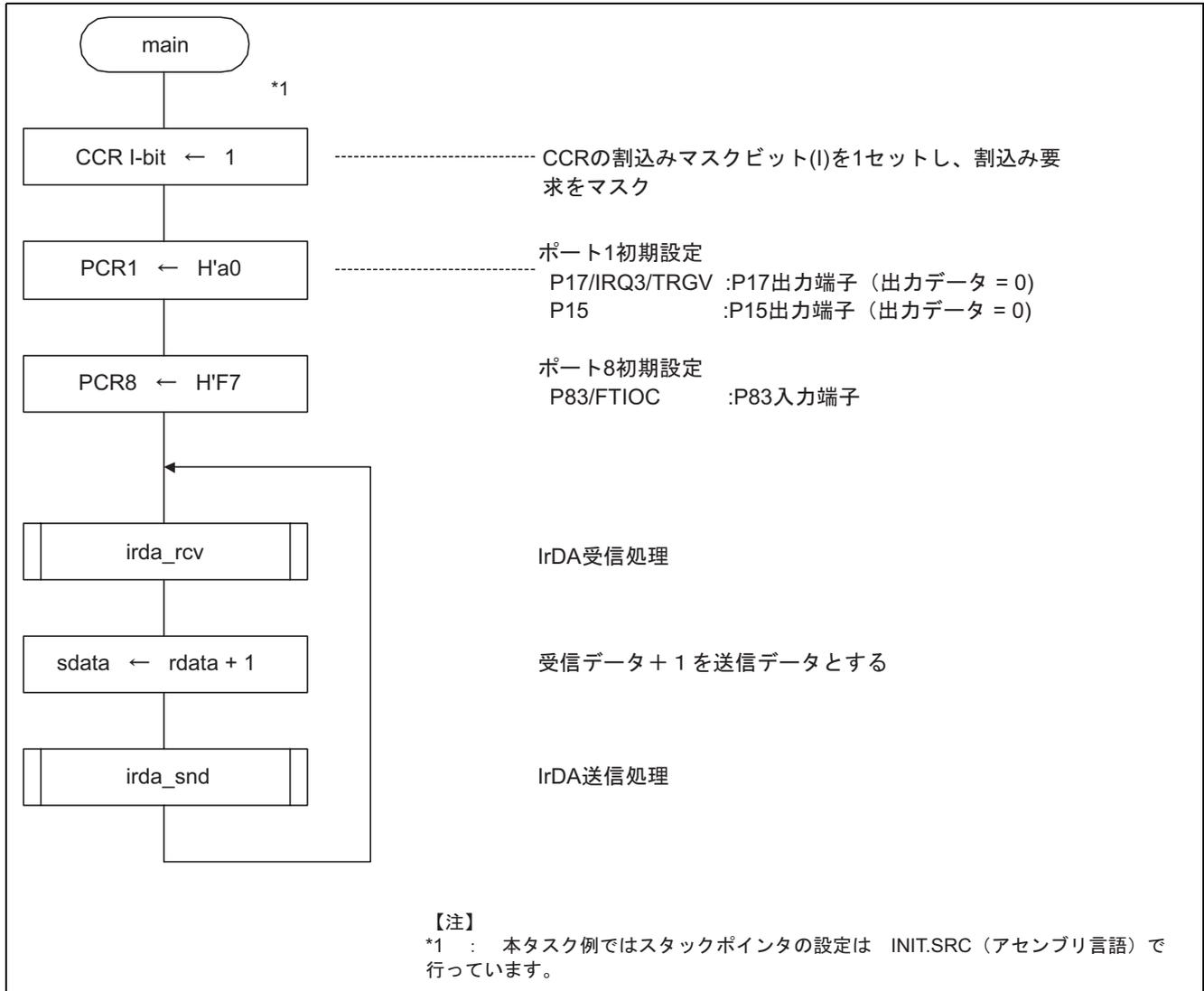
表 4 に本応用例における使用 RAM 説明を示します。

表 4 使用 RAM 説明

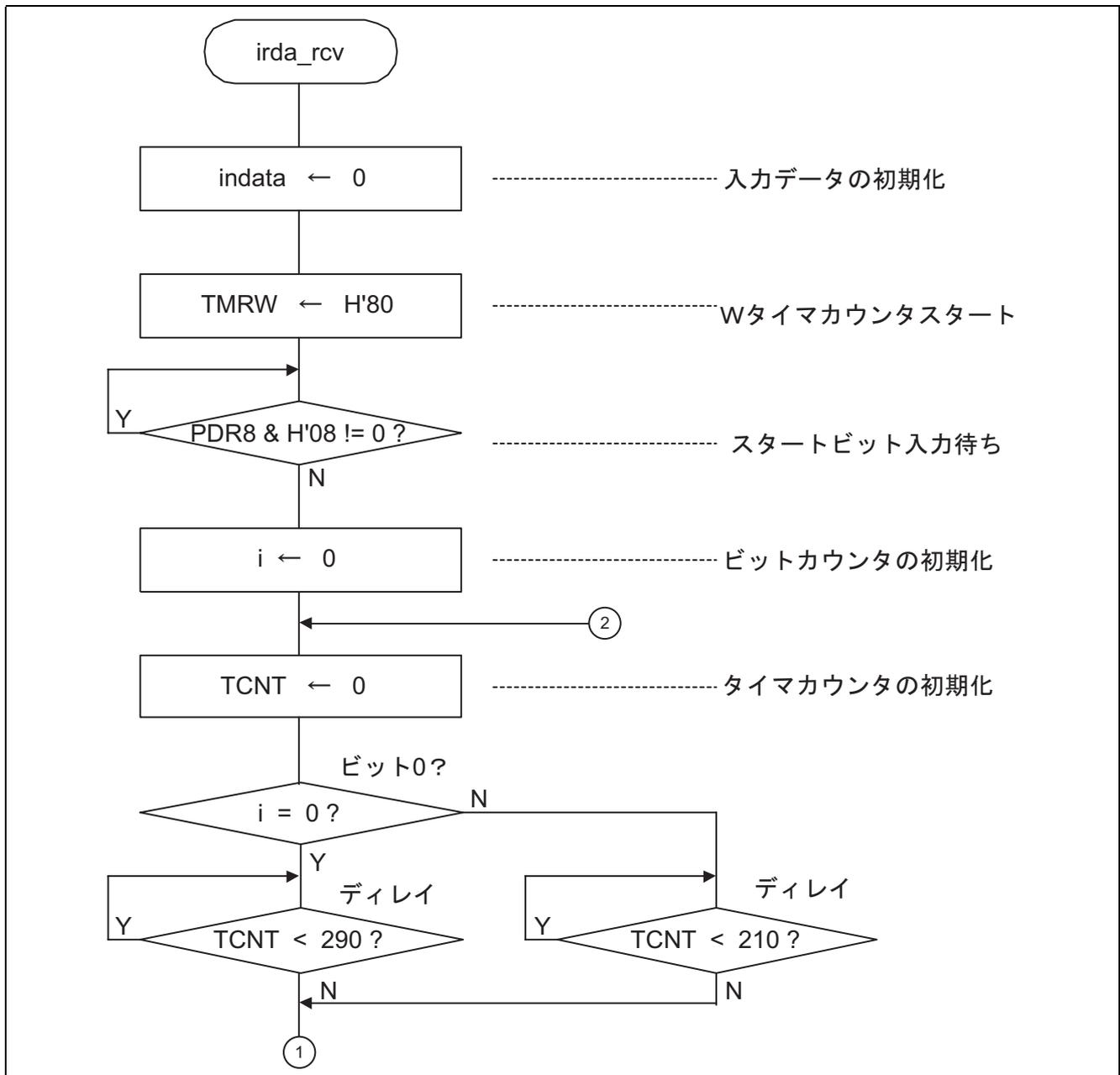
ラベル名	機能	アドレス	使用モジュールラベル名
sdata	送信データ (1 byte)	H'FB86	main,tmrw
rdata	受信データ (1 byte)	H'FB87	main,tmrw
bitdata	10 ビットの送信データを格納 (10 byte)	H'FB88	main,tmrw
i	ループカウンタを格納 (2 byte)	H'FB80	input_key
j	ループカウンタを格納 (2 byte)	H'FB82	input_key
bit	受信時にビットデータを格納 (1 byte)	H'FB92	main
indata	入力データを格納 (1 byte)	H'FB93	input_key
Wtimeup	W タイマタイムアップフラグ (2 byte)	H'FB84	input_key
bitpos	ビットの ON/OFF 判定用に使用 (1 byte)	H'FB94	input_key
dummy	ダミー (1 byte)	H'FB95	

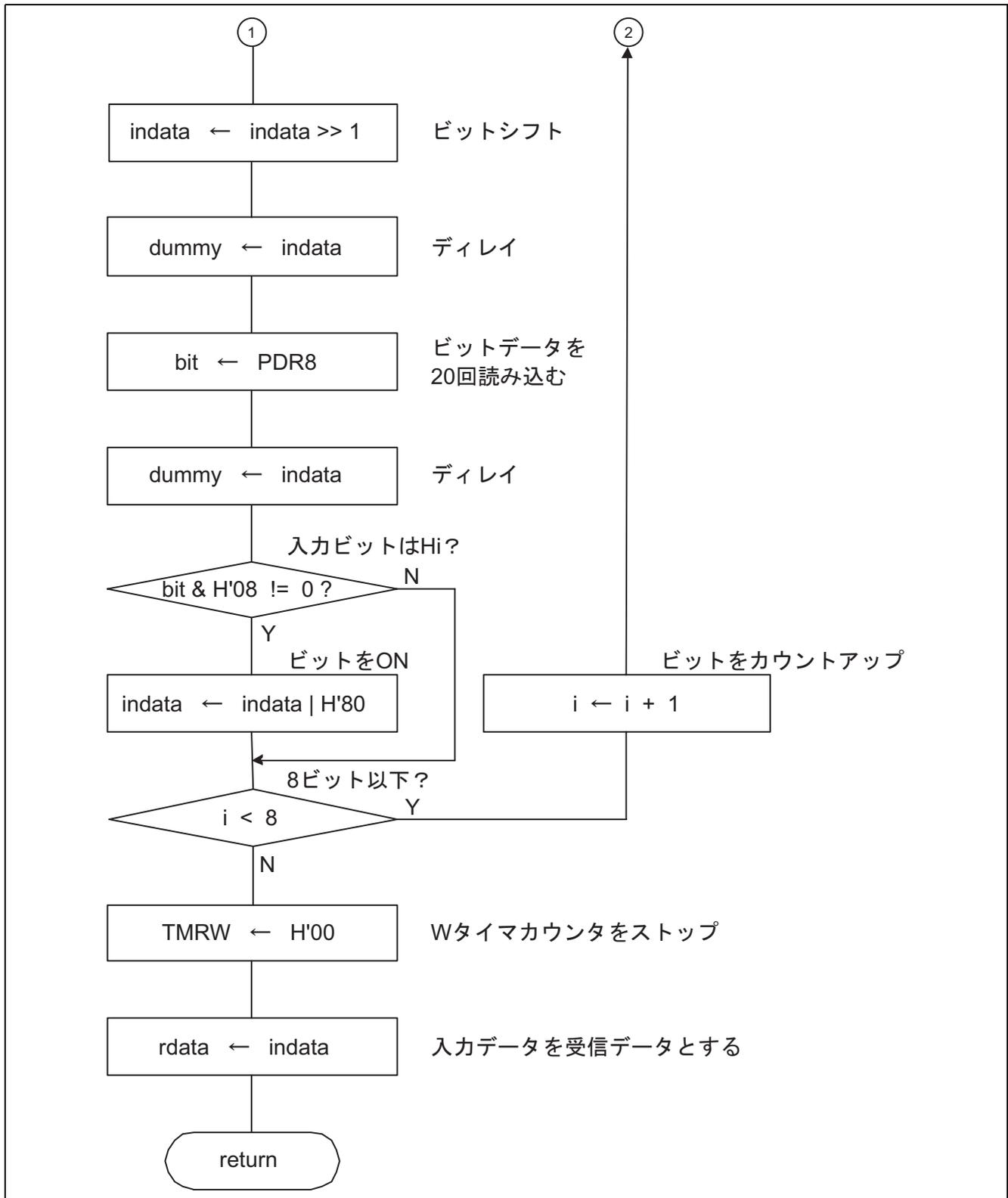
5. フローチャート

5.1 メインルーチン(main)

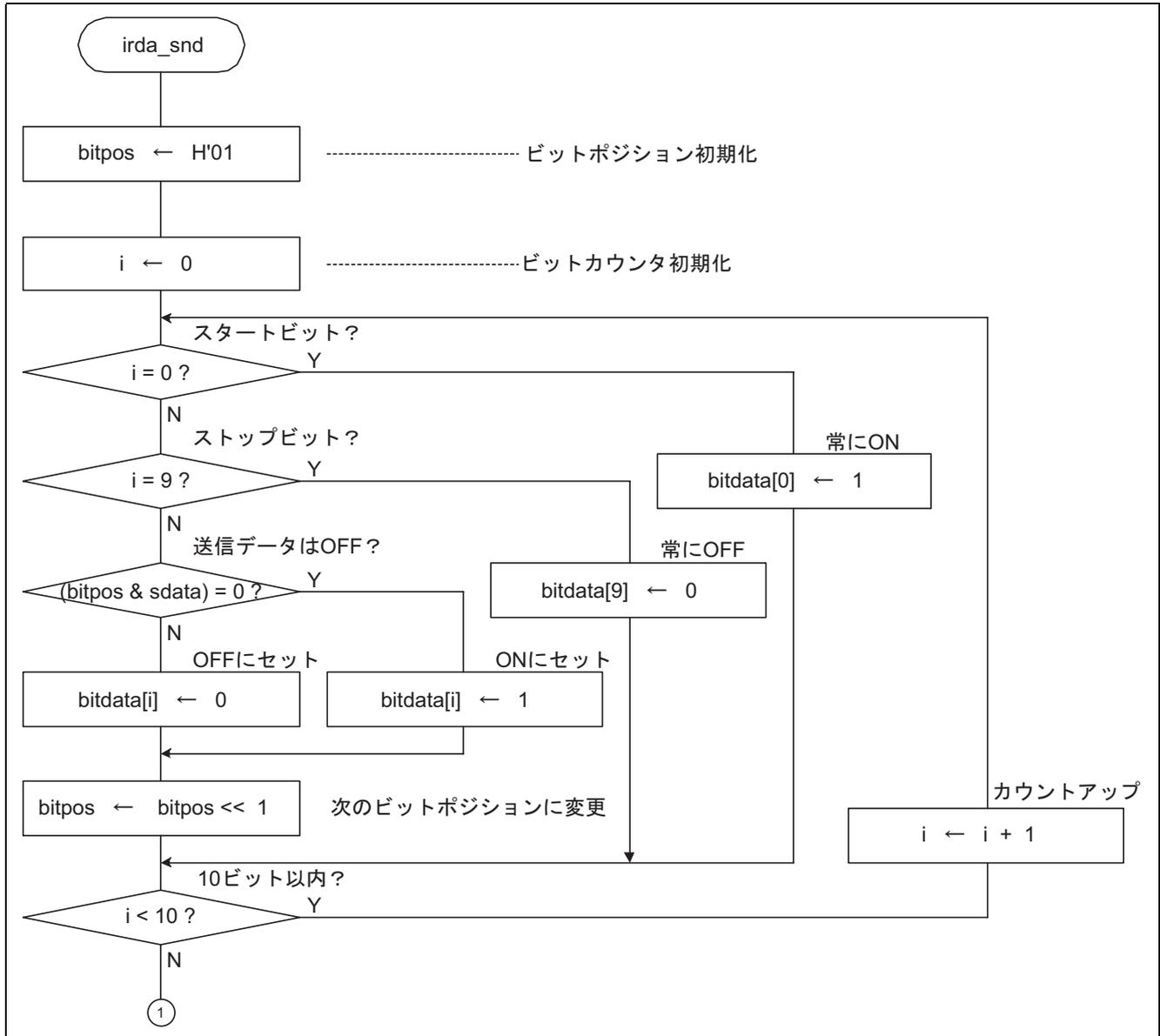


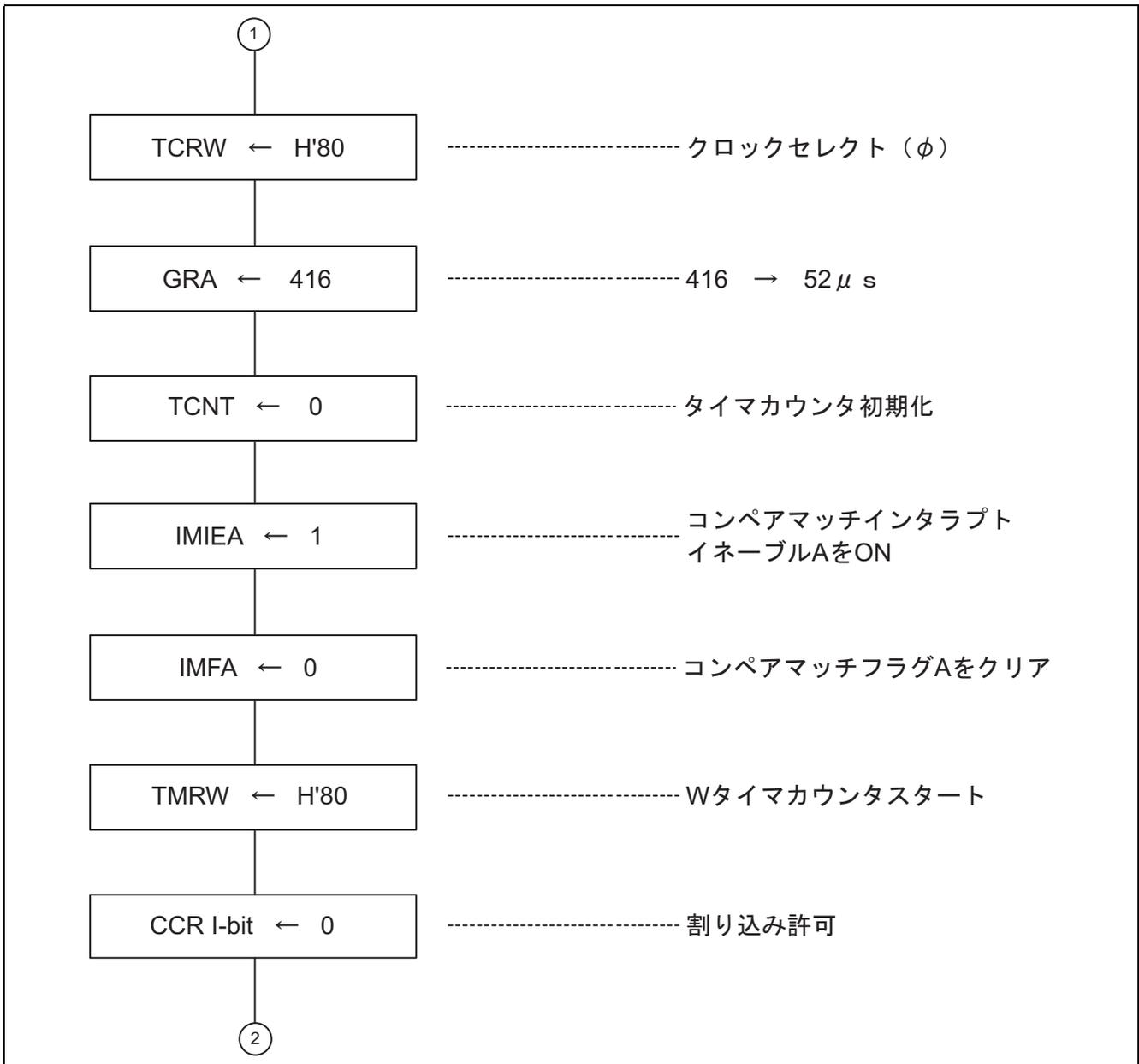
5.2 赤外線データ受信処理ルーチン(irda_rcv)

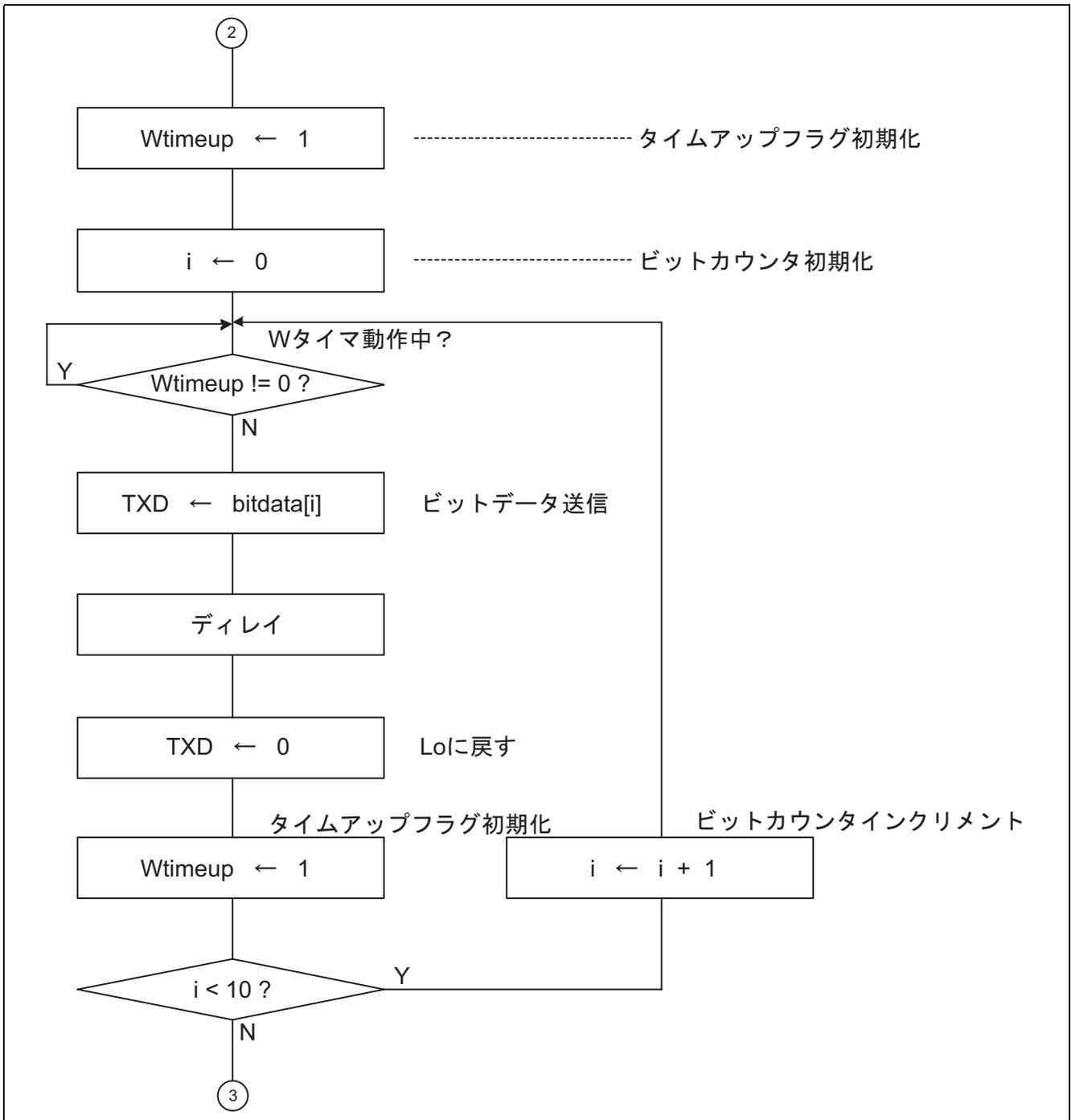


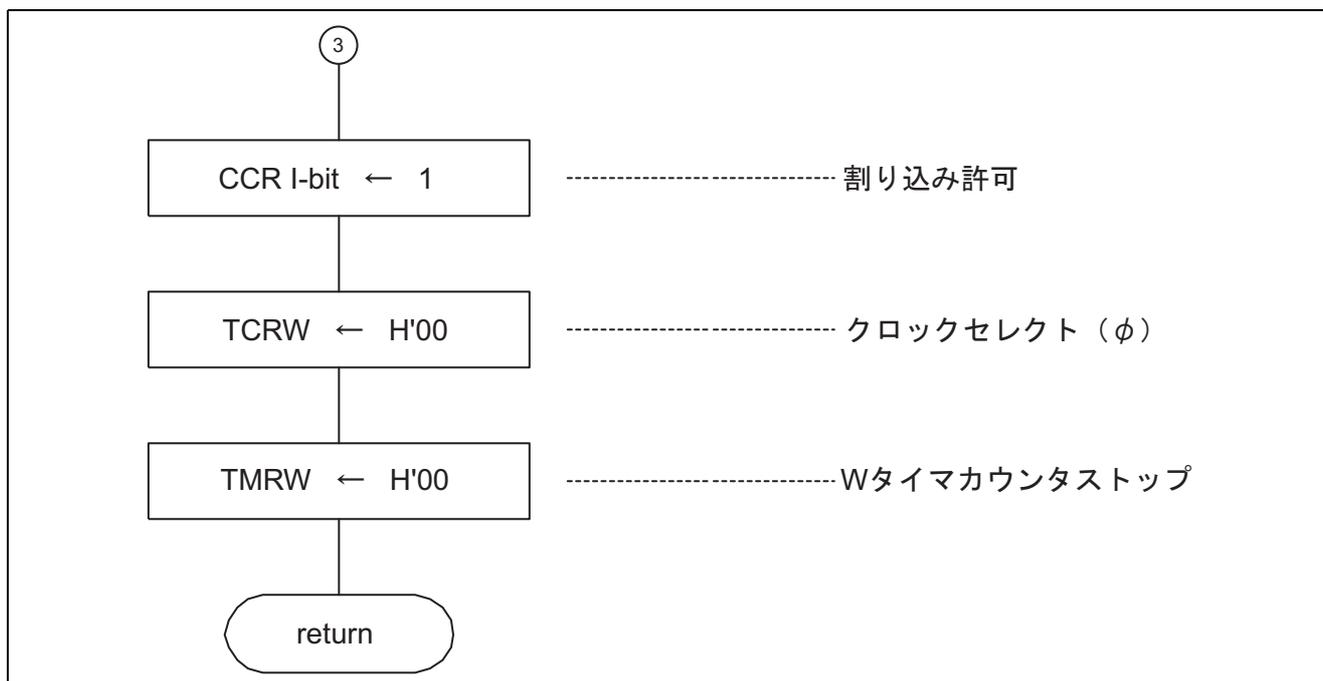


5.3 赤外線データ送信処理ルーチン(irda_snd)

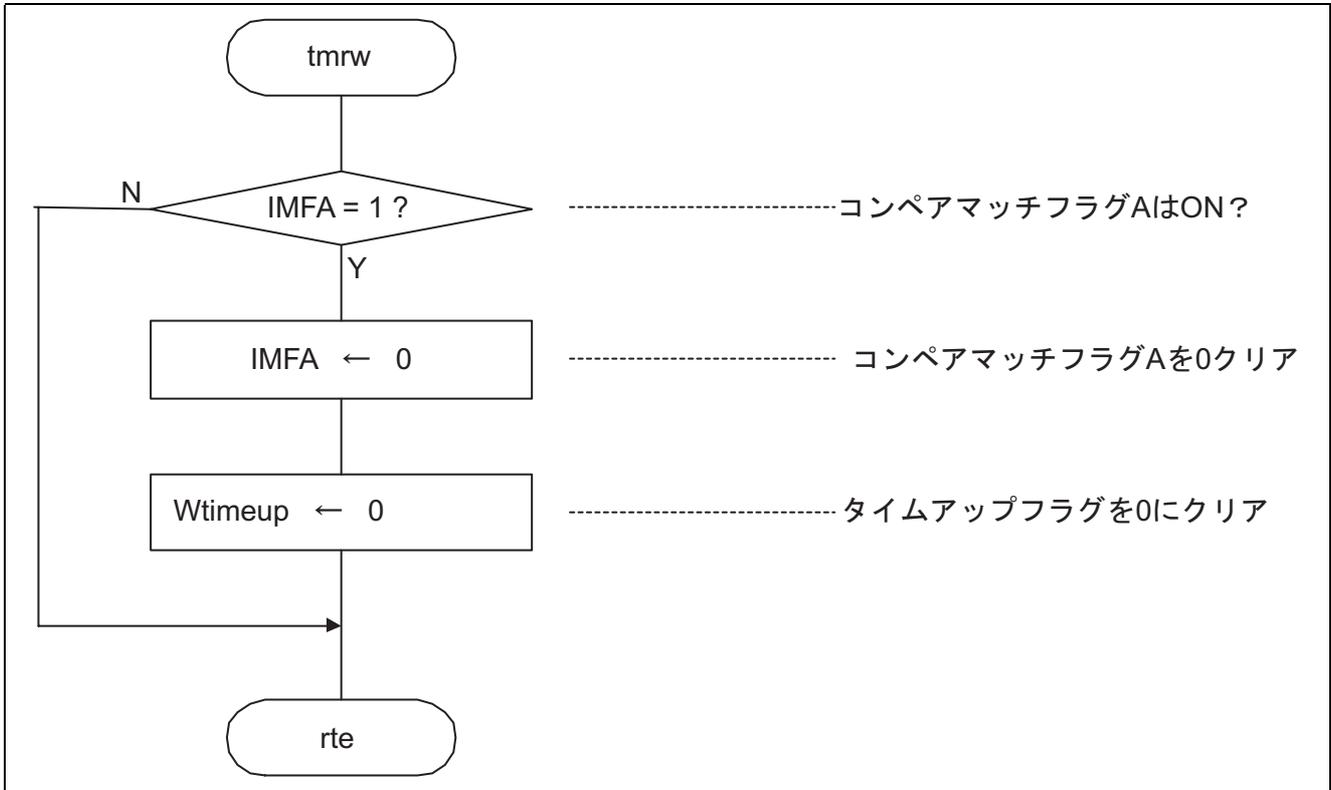








5.4 W タイマ割り込み処理ルーチン(tmrw)



6. プログラムリスト

INIT.SRC (プログラムリスト)

```
.export  _INIT
.import  _main
;
.section  P,CODE
_INIT:
mov.w   #h'ff80,r7
ldc.b   #b'10000000,ccr
jmp     @_main
;
.end

/* H8/300H tiny Series -H8/36014- Application note */
/* 応用編 */
/* IrDA 例 */
```

```
#include <machine.h>
```

```
/* Symbol definition */
```

```
struct BIT {
```

```
    unsigned char b7:1;      /* bit 7 */
    unsigned char b6:1;      /* bit 6 */
    unsigned char b5:1;      /* bit 5 */
    unsigned char b4:1;      /* bit 4 */
    unsigned char b3:1;      /* bit 3 */
    unsigned char b2:1;      /* bit 2 */
    unsigned char b1:1;      /* bit 1 */
    unsigned char b0:1;      /* bit 0 */
```

```
};
```

```
#define PDR1 *(volatile unsigned char *)0xFFD4
```

```
#define PDR1_BIT (*(struct BIT *)0xFFD4)
```

```
#define TXD      PDR1_BIT.b7
```

```
#define TST      PDR1_BIT.b5
```

```
#define PCR1 *(volatile unsigned char *)0xFFE4
```

```
#define PDR2 *(volatile unsigned char *)0xFFD5
```

```
#define PCR2 *(volatile unsigned char *)0xFFE5
```

```
#define PDR8 *(volatile unsigned char *)0xFFDB
```

```
#define PCR8 *(volatile unsigned char *)0xFFEB
```

```
#define SMR *(volatile unsigned char *)0xFFA8
```

```
#define BRR *(volatile unsigned char *)0xFFA9
```

```
#define SCR3 *(volatile unsigned char *)0xFFAA
```

```
#define TDR *(volatile unsigned char *)0xFFAB
```

```
#define SSR *(volatile unsigned char *)0xFFAC
```

```
#define RDR *(volatile unsigned char *)0xFFAD
```

```
#define PMR1 *(volatile unsigned char *)0xFFE0
```

```
/* Port data register 1 */
```

```
/* Port data register 1 */
```

```
/* Transfer Data */
```

```
/* Receive Data */
```

```
/* Port control register 1 */
```

```
/* Port data register 2 */
```

```
/* Port control register 2 */
```

```
/* Port data register 8 */
```

```
/* Port control register 8 */
```

```
/* Serial mode register */
```

```
/* Bit rate register */
```

```
/* Serial control register 3 */
```

```
/* Transmit data register */
```

```
/* Serial status register */
```

```
/* Receive data register */
```

```
/* Port Mode register 1 */
```

```

#define TMRW *(volatile unsigned char *)0xFF80      /* Timer mode register W */
#define TCRW *(volatile unsigned char *)0xFF81      /* Timer control register W */
#define TCRW_BIT (*(struct BIT *)0xFF81)           /* Timer Control Register W */
#define TIERW *(volatile unsigned char *)0xFF82     /* Timer interrupt enable register W */
#define TIERW_BIT (*(struct BIT *)0xFF82)          /* Timer Interrupt Enable Register */
#define IMIEA TIERW_BIT.b0                          /* Compare match Interrupt Enable A */
#define TSRW *(volatile unsigned char *)0xFF83     /* Timer status register W */
#define TSRW_BIT (*(struct BIT *)0xFF83)          /* Timer Status Register W */
#define IMFA TSRW_BIT.b0                            /* Compare match flag A */
#define TCNT *(volatile unsigned short *)0xFF86    /* Timer Counter */
#define GRA *(volatile unsigned short *)0xFF88     /* General register A */

#pragma interrupt (tmrw)
/* Function define */
extern void INIT(void);                          /* Stack pointer set */
void irda_rcv( void );                          /* receive routine */
void irda_snd( void );                          /* send routine */
void tmrw(void);                                /* Timer W interrupt routine */
void main(void);                                /* main routine */

/* RAM define */
volatile unsigned char sdata;                   /* send data */
volatile unsigned char rdata;                   /* receive data */
unsigned char bitdata[10];                      /* bit data (send) */
int i,j;                                        /* loop counter */
unsigned char bit, indata;                      /* input data */
volatile int Wtimeup;                           /* W timer time up */
unsigned char bitpos;                           /* bit position */
char dummy;

/* Vector address */
#pragma section V1                               /* Vector section set */
void (*const VEC_TBL1[])(void) = {
    INIT                                         /* H'0000 Reset vector */
};

#pragma section V2                               /* Vector section set */
void (*const VEC_TBL2[])(void) = {
    tmrw                                        /* H'002a Timer W interrupt vector */
};
#pragma section                                  /* P */

```

```

/*****
/* Main program
*****/
void main(void)
{
    set_imask_ccr(1);                /* CCR I-bit = 1 */

    PCR1 = 0xA0;                    /* Port2 bit7,bit5 output */
    PCR8 = 0xF7;                    /* Port8 bit3 input */

    while(1){
        irda_rcv();                 /* receive routine */
        sdata = rdata + 1;
        irda_snd();                 /* send routine */
    }
}

```

```

/*****/
/*  receive routine                                     */
/*****/
void irda_rcv( void )
{
    indata = 0;                                         /* input data initialize */

    TMRW = 0x80;                                       /* W Timer Counter Count Start */
    while(PDR8 & 0x08);                                /* wait for start bit */

    for(i=0;i<8;i++){
        TCNT = 0;                                       /* Timer Counter zero clear */
        if(i == 0)
            while(TCNT < 290);                         /* 36.25us */
        else
            while(TCNT < 210);                         /* 26.25us */
        indata >>= 1;                                   /* shift input data */
        dummy = indata;                                /* dummy wait(8 cycle) */
        bit = PDR8 & PDR8
            & PDR8 & PDR8 & PDR8 & PDR8 & PDR8 & PDR8 & PDR8 & PDR8;
        dummy = indata;                                /* dummy wait(8 cycle) */
        if(bit & 0x08) {
            indata |= 0x80;                             /* if input is H-level */
            /* then data set */
        }
    }
    TMRW = 0x00;                                       /* W Timer Counter Count Stop */

    rdata = indata;
}

```

```

/*****
/*      send routine      */
/*****
void irda_snd( void )
{
    /* set send data      */
    bitpos = 0x01;          /* set bit position (0 bit) */
    for(i = 0; i < 10; i++) {
        if(i == 0) {
            bitdata[0] = 1;    /* set start bit      */
        } else if(i == 9) {
            bitdata[9] = 0;    /* set stop bit      */
        } else {
            if((bitpos & sdata) == 0) {
                bitdata[i] = 1;    /* set data bit      */
            } else {
                bitdata[i] = 0;
            }
            bitpos <<= 1;    /* shift bit position */
        }
    }

    TCRW = 0x80;          /* Clock Select( ) */
    GRA = 416;          /* 52us */
    TCNT = 0;          /* Timer Counter zero clear */
    IMIEA = 1;    /* Compare match Interrupt Enable A */
    IMFA = 0;    /* Compare match flag A */
    TMRW = 0x80;    /* W Timer Counter Count Start */
    set_imask_ccr(0);    /* CCR I-bit = 0 */

    Wtimeup = 1;
    for(i = 0; i < 10; i++) {
        while(Wtimeup);    /* wait 52us */
        TXD = bitdata[i];    /* TXD ON */
        for(j = 0; j < 1; j++);
        TXD = 0;    /* TXD OFF */
        Wtimeup = 1;
    }

    set_imask_ccr(1);    /* CCR I-bit = 1 */
    TCRW = 0x00;    /* Clock Select( ) */
    TMRW = 0x00;    /* W Timer Counter Count Stop */
}

/*****
/* Timer W Interrupt(every 52us) */
/*****
void tmrw(void)
{
    if ( IMFA == 1 ) {
        IMFA = 0;    /* Clear Compare match flag A */
        Wtimeup = 0;    /* set time up */
    }
}

```


改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2003.12.22	—	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 テクノロジ製品をご購入いただくための参考資料であり、本資料中に記載の技術情報についてルネサス テクノロジが所有する知的財産権その他の権利の実施、使用を許諾するものではありません。
2. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他応用回路例の使用に起因する損害、第三者所有の権利に対する侵害に関し、ルネサス テクノロジは責任を負いません。
3. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他全ての情報は本資料発行時点のものであり、ルネサス テクノロジは、予告なしに、本資料に記載した製品または仕様を変更することがあります。ルネサス テクノロジ半導体製品のご購入に当たりますは、事前にルネサス テクノロジ、ルネサス販売または特約店へ最新の情報をご確認頂きますとともに、ルネサス テクノロジホームページ(<http://www.renesas.com>)などを通じて公開される情報に常にご注意ください。
4. 本資料に記載した情報は、正確を期すため、慎重に制作した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 テクノロジはその責任を負いません。
5. 本資料に記載の製品データ、図、表に示す技術的な内容、プログラム及びアルゴリズムを流用する場合は、技術内容、プログラム、アルゴリズム単位で評価するだけでなく、システム全体で十分に評価し、お客様の責任において適用可否を判断してください。ルネサス テクノロジは、適用可否に対する責任を負いません。
6. 本資料に記載された製品は、人命にかかわるような状況の下で使用される機器あるいはシステムに用いられることを目的として設計、製造されたものではありません。本資料に記載の製品を運輸、移動体用、医療用、航空宇宙用、原子力制御用、海底中継用機器あるいはシステムなど、特殊用途へのご利用をご検討の際には、ルネサス テクノロジ、ルネサス販売または特約店へご照会ください。
7. 本資料の転載、複製については、文書によるルネサス テクノロジの事前の承諾が必要です。
8. 本資料に関し詳細についてのお問い合わせ、その他お気づきの点がございましたらルネサス テクノロジ、ルネサス販売または特約店までご照会ください。