

2.10 ウォッチドッグタイマ

ウォッチドッグタイマ	使用機能	ウォッチドッグタイマ機能
------------	------	--------------

仕様

- (1) ウォッチドッグタイマ機能を使用して、ウォッチドッグ動作を行ないます。
- (2) タイマカウンタWDのオーバーフロー周期を122.88msに設定し、122.88ms以内にタイマカウンタWDをイニシャライズしなければ、内部リセット信号が発生するように動作させます。
- (3) 通常動作では、一定時間ごとにLEDが点灯/消灯を繰り返し、タイマカウンタWDがオーバーフローする前にタイマカウンタWDをイニシャライズするように設定します。
- (4) IRQ₀入力端子に接続したスイッチのオンにより、タイマカウンタWDがイニシャライズされずにオーバーフローし、内部リセット信号が発生するように動作させます。
- (5) LEDはポート7のP74出力端子に接続しているものとします。
- (6) 図1に、IRQ₀入力端子のスイッチ接続例を示します。

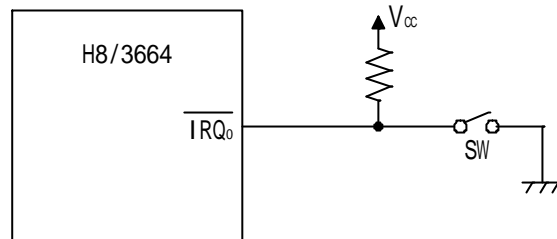


図1 IRQ₀入力端子のスイッチ接続例

使用機能説明

- (1) 本タスク例では、ウォッチドッグタイマ機能を使用して、ウォッチドッグ動作を行ないます。
- (a) 図2にウォッチドッグタイマ機能のブロック図を示します。以下にウォッチドッグタイマ機能のブロック図について説明します。
 - ・システムクロック () は、16MHzのクロックで、CPUおよび周辺機能を動作させるための基準クロックです。
 - ・プリスケアラS (PSS) は、 を入力とする13ビットのカウンタで、1サイクルごとにカウントアップします。
 - ・タイマカウンタWD (TCWD) は、8ビットのリード/ライト可能なアップカウンタで、入力する内部クロックによりカウントアップされます。入力するクロックは、 /8192です。
 - ・タイマコントロール/ステータスレジスタWD (TCSRWD) は、8ビットのリード/ライト可能なレジスタで、TCSRWD、TCWDの書き込み制御、ウォッチドッグタイマの動作制御、動作状態を示します。
 - ・以下に、本タスク例におけるTCWDのオーバーフロー周期の計算法を示します。

$$\begin{aligned}
 \text{TCWDオーバーフロー周期} &= \frac{1}{\text{システムクロック} / 8192} \times (256 - (\text{TCWDのリロード設定値})) \\
 &= 0.512\text{ms} \times (256 - 16) \\
 &= 122.88\text{ms}
 \end{aligned}$$

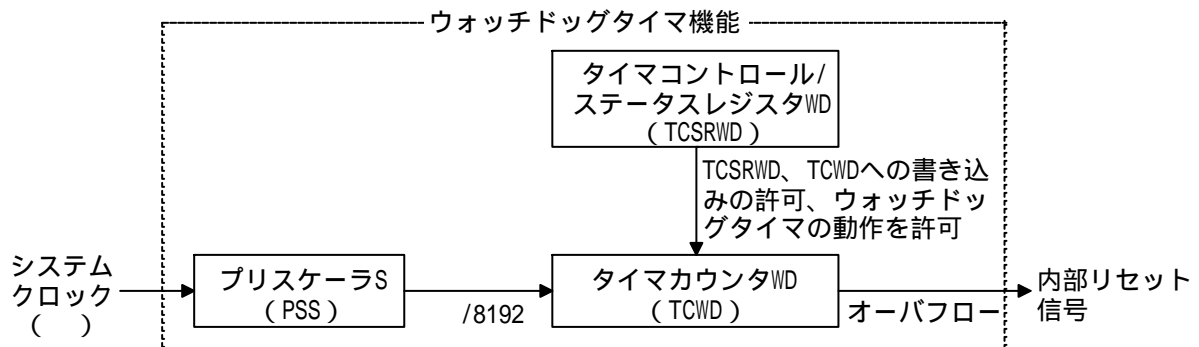


図2 ウォッチドッグタイマ機能のブロック図

ウォッチドッグタイマ	使用機能	ウォッチドッグタイマ機能
------------	------	--------------

使用機能説明

(2) 表1に本タスク例の機能割付けを示します。表1に示すように機能を割付け、ウォッチドッグタイマ機能によるウォッチドッグ動作を行ないます。

表1 機能割付け

機能	機能割付け
PSS	システムクロックを入力とする13ビットのカウンタ
TCWD	システムクロックの8192分周のクロックを入力とする8ビットのカウンタ
TCSRWD	TCSRWD、TCWDの書き込み制御、ウォッチドッグタイマの動作制御、動作状態を示す
$\overline{\text{IRQ}}_0$	SW入力の入力端子
P74	LED出力

動作原理

(1) 図3に動作原理を示します。図3に示すようなハードウェア処理、およびソフトウェア処理によりウォッチドッグタイマ機能によるウォッチドッグ動作を行ないます。

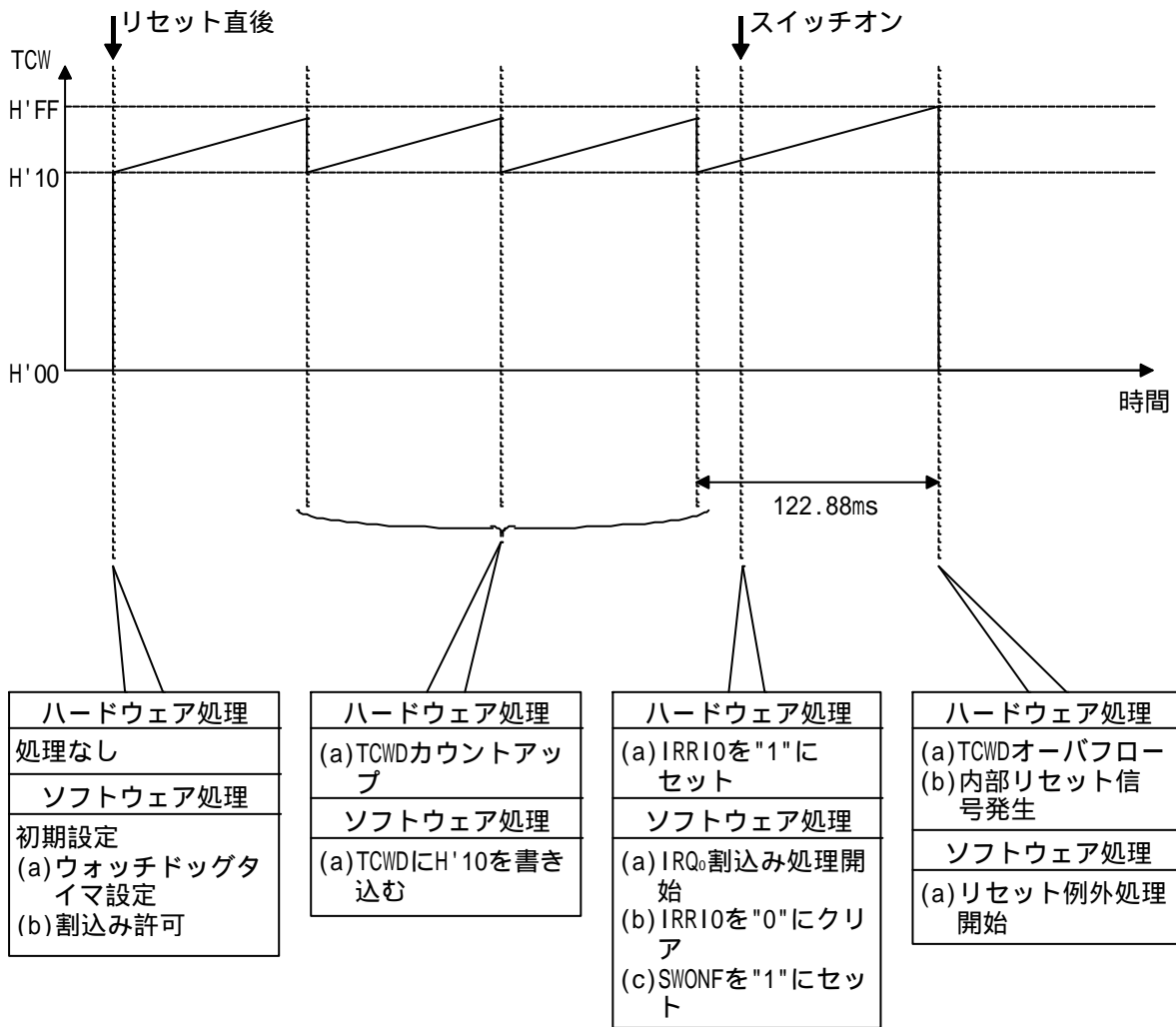


図3 ウォッチドッグタイマ機能によるウォッチドッグ動作原理

ウォッチドッグタイマ	使用機能	ウォッチドッグタイマ機能
------------	------	--------------

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	ウォッチドッグタイマ機能の設定、割込みの許可、LEDの制御、およびIRQ ₀ に接続したSWのONの判定を行なう
スイッチオン	IRQ0	IRQ ₀ 割込み処理ルーチンで、SWONFを"1"にセット

(2) 引数の説明

本タスク例では、引数は使用していません。

(3) 使用内部レジスタ説明

表3に本タスク例におけるH8/3664の使用内部レジスタ説明を示します。

表3 H8/3664の使用内部レジスタ説明

レジスタ名	機能	アドレス	設定値
TCSRWD	B6WI タイマコントロール/ステータスレジスタWD (ビット6書き込み禁止) : B6WI="0"のとき、TCSRWDビット6への書き込みを許可 : B6WI="1"のとき、TCSRWDビット6への書き込みを禁止	H'FFC0 ビット7	1
	TCWE タイマコントロール/ステータスレジスタWD (タイマカウンタW書き込み許可) : TCWE="1"のとき、TCWDへの8ビットデータの書き込みを許可	H'FFC0 ビット6	1
	B4WI タイマコントロール/ステータスレジスタWD (ビット4書き込み禁止) : B4WI="0"のとき、TCSRWDビット4への書き込みを許可 : B4WI="1"のとき、TCSRWDビット4への書き込みを禁止	H'FFC0 ビット5	1
	TCSRWE タイマコントロール/ステータスレジスタWD (タイマコントロール/ステータスレジスタW書き込み許可) : TCSRWE="1"のとき、TCSRWDビット2およびビット0への書き込みを許可	H'FFC0 ビット4	1
	B2WI タイマコントロール/ステータスレジスタWD (ビット2書き込み禁止) : B2WI="0"のとき、TCSRWDビット2への書き込みを許可 : B2WI="1"のとき、TCSRWDビット2への書き込みを禁止	H'FFC0 ビット3	1
	WDON タイマコントロール/ステータスレジスタWD (ウォッチドッグタイマオン) : WDON="1"のとき、ウォッチドッグタイマの動作を許可	H'FFC0 ビット2	1
	BOWI タイマコントロール/ステータスレジスタWD (ビット0書き込み禁止) : BOWI="0"のとき、TCSRWDビット0への書き込みを許可 : BOWI="1"のとき、TCSRWDビット0への書き込みを禁止	H'FFC0 ビット1	1
	WRST タイマコントロール/ステータスレジスタWD (ウォッチドッグタイマリセット) : WRST="0"のとき、TCWDがオーバフローし内部リセット信号が発生していないことを示す : WRST="1"のとき、TCWDがオーバフローし内部リセット信号が発生したことを示す	H'FFC0 ビット0	0
TCWD	タイマカウンタWD : システムクロックの8192分周のクロックを入力とする8ビットのカウンタ	H'FFC1	H'10

ソフトウェア説明

表3 H8/3664の使用内部レジスタ説明

レジスタ名	機能	アドレス	設定値
TMWD	タイマモードレジスタWD (クロックセレクト3~0) : TCWDに入力する内部クロック /8192分周を選択	H'FFC2	H'FF
PDR7 P74	ポートデータレジスタ7 (ポートデータレジスタ7 ₄) : P7 ₄ =0のとき、P7 ₄ 端子の出力レベルは"Low" : P7 ₄ =1のとき、P7 ₄ 端子の出力レベルは"High"	H'FFDA ビット4	0
PCR7 PCR74	ポートコントロールレジスタ7 (ポートコントロールレジスタ7 ₄) : PCR7 ₄ ="0"のとき、P7 ₄ 端子をP7 ₄ 入力端子機能に設定 : PCR7 ₄ ="1"のとき、P7 ₄ 端子をP7 ₄ 出力端子機能に設定	H'FFEA ビット4	1
PMR1 IRQ0_SET	ポートモードレジスタ1 (P14/IRQ ₀ 端子の機能を選択) : IRQ0_SET="1"のとき、IRQ ₀ 端子をIRQ ₀ 入力端子機能に設定	H'FFE0 ビット4	1
IEGR1 IEG0	割込みエッジセレクトレジスタ1 (IRQ ₀ エッジセレクト) : IEG0="0"のとき、IRQ ₀ 端子入力の検出エッジに立ち下がりエッジを選択 : IEG0="1"のとき、IRQ ₀ 端子入力の検出エッジに立ち上がりエッジを選択	H'FFF2 ビット0	0
IENR1 IEN0	割込みイネーブルレジスタ1 (IRQ ₀ 割込み要求イネーブル) : IEN0="1"のとき、IRQ ₀ 端子の割込み要求を許可	H'FFF4 ビット0	1
IRR1 IRR10	割込みフラグレジスタ1 (IRQ ₀ 割込み要求フラグ) : IRR10="0"のとき、IRQ ₀ 端子の割込みが要求されていない : IRR10="1"のとき、IRQ ₀ 端子の割込みが要求されている	H'FFF6 ビット0	0

(4) 使用RAM説明

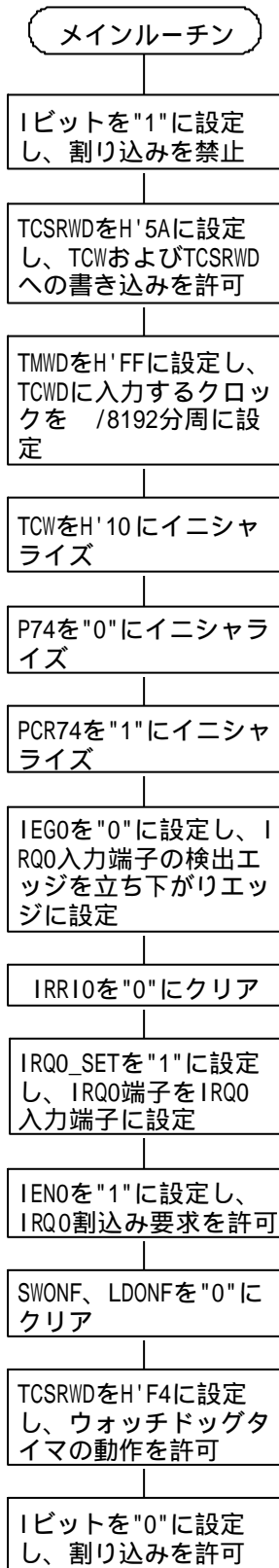
表4に本タスク例で使用するRAMの説明を示します。

表4 使用RAM説明

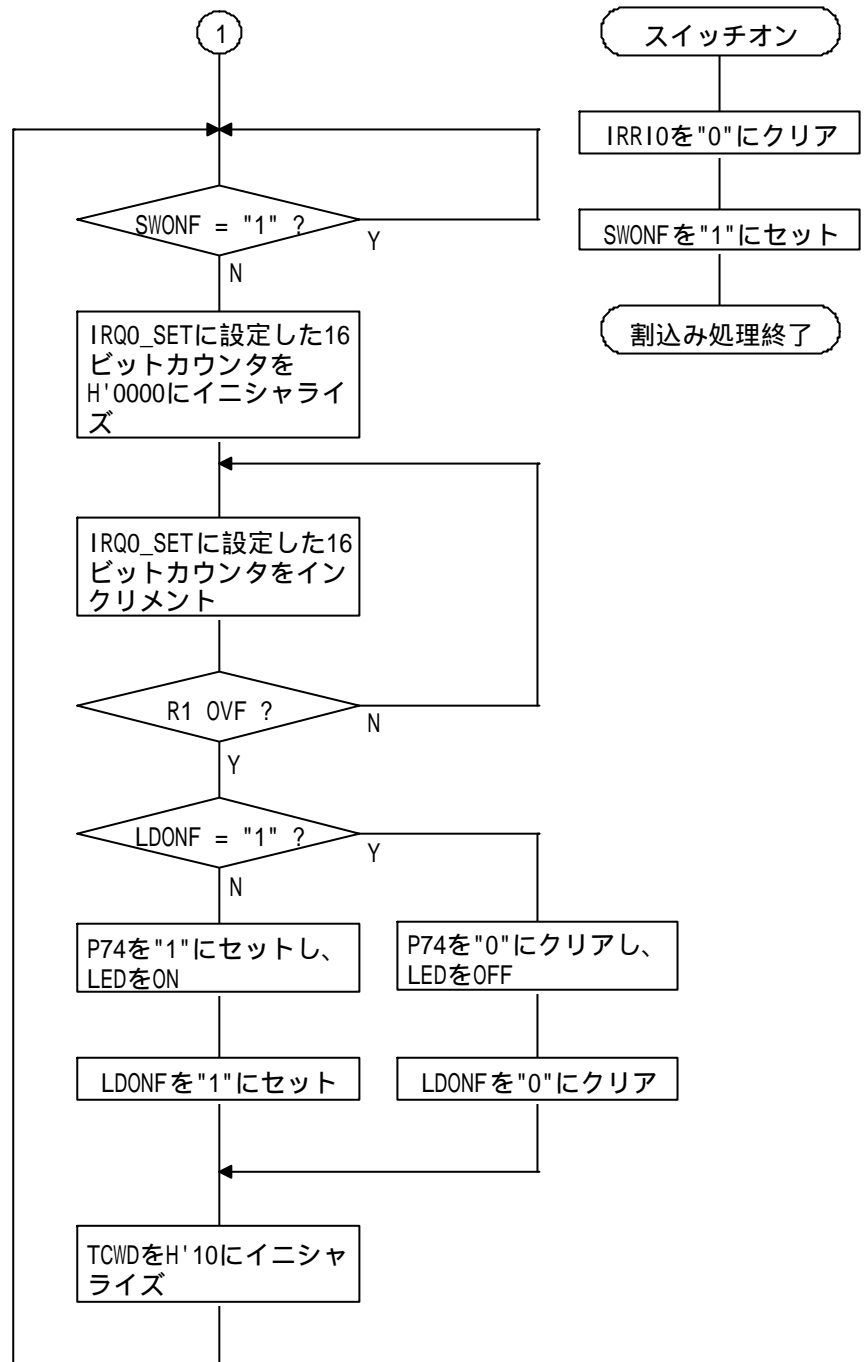
ラベル名	機能	アドレス	使用モジュール名
counter_int	LEDのON、OFFを行うアップカウンタ	H'FB80	メインルーチン
USRF	SWONF	スイッチ入力のON/OFFを判定するフラグ	メインルーチン スイッチオン
	LDONF	LEDのON/OFFを判定するフラグ	H'FB82 ビット1

フローチャート

(a) メインルーチン



(b) IRQ0割り込み処理ルーチン



本例ではスタックポインタはINIT.SRC (アセンブリ言語) で設定してあります。

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   #'FF80,R7
        LDC.B   #'10000000,CCR
        JMP     @_main
;
        .END

/*****
/*
/*      H8/300H Tiny Series -H8/3664-
/*      Application Note
/*
/*      'Watchdog Timer'
/*
/*      Function
/*      : Watchdog Timer
/*
/*      External Clock : 16MHz
/*      Internal Clock : 16MHz
/*      Sub Clock      : 32.768kHz
/*
/*
*****/

#include <machine.h>

/*****
/*      Symbol Definition
*****/

struct BIT {
    unsigned char    b7:1;    /* bit7 */
    unsigned char    b6:1;    /* bit6 */
    unsigned char    b5:1;    /* bit5 */
    unsigned char    b4:1;    /* bit4 */
    unsigned char    b3:1;    /* bit3 */
    unsigned char    b2:1;    /* bit2 */
    unsigned char    b1:1;    /* bit1 */
    unsigned char    b0:1;    /* bit0 */
};

#define TCSRWD      *(volatile unsigned char *)0xFFC0 /* Timer Control/Status Register WD */
#define TCSRWD_BIT (*(struct BIT *)0xFFC0) /* Timer Control/Status Register WD */
#define B6WI       TCSRWD_BIT.b7 /* Bit-6 Write Disable */
#define TCWE       TCSRWD_BIT.b6 /* Timer Counter W Write Enable */
#define B4WI       TCSRWD_BIT.b5 /* Bit-4 Write Disable */
#define TCSRWE     TCSRWD_BIT.b4 /* Timer Control/Status Register W Write Enable */
#define B2WI       TCSRWD_BIT.b3 /* Bit-2 Write Disable */
#define WDON       TCSRWD_BIT.b2 /* Watchdog Timer ON */
#define B0WI       TCSRWD_BIT.b1 /* Bit-0 Write Disable */
#define WRST       TCSRWD_BIT.b0 /* Watchdog Timer Reset */
#define TCWD       *(volatile unsigned char *)0xFFC1 /* Timer Counter WD */
#define TMWD       *(volatile unsigned char *)0xFFC2 /* Timer Mode WD */
#define PDR7_BIT  (*(struct BIT *)0xFFDA) /* Port Data Register 7 */
#define P74       PDR7_BIT.b4 /* Port Data Register 7 bit4 */
#define PCR7_BIT  (*(struct BIT *)0xFFEA) /* Port Control Register 7 */
#define PCR74     PCR7_BIT.b4 /* Port Control Register 7 bit4 */
#define IEGR1_BIT (*(struct BIT *)0xFFF2) /* Interrupt Edge Select Register 2 */
#define IEGO       IEGR1_BIT.b0 /* IEGO Edge Select */
#define IENR1_BIT (*(struct BIT *)0xFFF4) /* Interrupt Enable Register 1 */
#define IENO       IENR1_BIT.b0 /* IENO Inetrrupt Enable */
#define IRR1_BIT  (*(struct BIT *)0xFFF6) /* Interrupt Request Register 1 */
#define IRR10     IRR1_BIT.b0 /* IRR10 Interrupt Request Register */
#define PMR1_BIT  (*(struct BIT *)0xFFE0) /* Prot Mode Register 1 */
#define IRQO_SET  PMR1_BIT.b4 /* Prot Mode Register 1 bit4 */

#pragma interrupt (IRQ0)

```

ウォッチドッグタイマ	使用機能	ウォッチドッグタイマ機能
プログラムリスト		
<pre> /*****/ /* 関数定義 */ /*****/ extern void INIT (void); /* SP Set */ void main (void); void IRQ0 (void); /*****/ /* RAM define */ /*****/ unsigned int counter_int; unsigned char USRF; /* User Flag Erea */ #define USRF_BIT (*(struct BIT *)&USRF) #define SWONF USRF_BIT.b0 #define LDONF USRF_BIT.b1 /*****/ /* Vector Address */ /*****/ #pragma section V1 /* VECTOR SECTOIN SET */ void (*const VEC_TBL1[])(void) = { INIT /* 00 Reset */ }; #pragma section V2 /* VECTOR SECTOIN SET */ void (*const VEC_TBL2[])(void) = { IRQ0 /* IRQ0 Interrupt */ }; #pragma section /* P */ /*****/ /* Main Program */ /*****/ void main (void) { set_imask_ccr(1); /* Interrupt Disable */ TCSRWD = 0x5A; /* TCWD And TCSRWD Write Enable */ TMWD = 0xFF; /* Initialize TMWD */ TCWD = 0x10; /* Initialize TCWD */ P74 = 0; /* Initialize P74 Terminal Output Level */ PCR74 = 1; /* Initialize PCR74 Output Terminal Function */ IEGO = 1; /* Initialize IRQ0 Terminal Input Edge */ IRRI0 = 0; /* Initialize IRQ0 Interrupt Request Flag */ IRQ0_SET = 1; /* Initialize IRQ0 Input Terminal */ IENO = 1; /* IRQ0 Interrupt Enable */ SWONF = 0; /* Initialize SWONF */ LDONF = 0; /* Initialize LDONF */ TCSRWD = 0xF4; /* Watchdog Timer On */ set_imask_ccr(0); /* Interrupt Enable */ while(1){ while(SWONF == 1){ ; } counter_int = 0; /* counter Clear */ do{ counter_int ++; /* counter Countup */ }while(counter_int != 0); /* counter_int = FFFF ? */ if(LDONF == 1){ </pre>		

プログラムリスト

```

    P74 = 0;          /* Turn Off LED */
    LDONF = 0;       /* LDONF Clear */
}else{
    P74 = 1;          /* Turn On LED */
    LDONF = 1;       /* LDONF set */
}

TCWD = 0x10;        /* Initialize TCWD */

}

}

/*****
/* IRQ0 Interrupt */
/*****
void IRQ0 ( void )
{
    IRR10 = 0;       /* Clear IRR10 */
    SWONF = 1;      /* Set SWONF */
}

```

リンクアドレス指定

セクション名	アドレス
CV1	H'0000
CV2	H'001C
P	H'0100
B	H'FB80