Application Note

Watchdog Timer

For H8/36077 series

Contact Info:

Brazen Tek, Inc.
20121 Ventura Blvd.
Suite 310
Woodland Hills, CA 91364
USA
Tel/Fax: (818) 710-9262
E-mail: info@brazentek.com

August 2008

Introduction

The watchdog timer is operated.

Target Device

H8/36077 series

Contents

1. Specifications

i. H8/36077 StarterKit Plus SKP36077 is chosen.
ii. When the reset signal is disabled, WDT (watchdog timer) is started.
iii. The LEDs D2 and D1 on the board flash and WDT is enabled in the initial state.
iv. The WDT can operate in any operating mode when any of nine counter input clocks is selected.
v. Timer counter WD (TCWD) generates an internal reset signal within WDT so that it overflows.
vi. LEDs D2 and D1 are connected to SCL and SDA of 36077, P56 (pin 26) and P57 (pin 27).

2. Description of Functions used

Figure 2.1 shows the Watchdog timer block diagram. The block diagram is summarized as:

- The system clock (ϕ) is a 9.8304MHz oscillator clock that is used as the CPU operation.
- Prescaler S (PSS) is a 13-bit counter using the system clock (ϕ) as its input clock. When reset is enabled, PSS is initialized (H'0000). It cannot be read from or written to by the CPU.
- Timer counter WD (TCWD) is an 8-bit readable/writable up counter. When the internal reset signal is generated, it overflows from higher to lower bits (H'FF to H'00).
- Timer control/status register WD (TCSRWD) controls the WDT operation and indicates operating state.
- Depending on the TCWD set value, an overflow period in the range of 1 to 256 input clock cycles can be selected.
- The TCWD overflow cycle is calculated as follows:

$$\text{TCWD O.C} = \frac{1}{\frac{\phi}{8192}} \times (256 - (TCWD\ \text{Re}load\ Value))$$

$$= (0.833 \times 10^{-3}) \times (240)$$

$$= 200 \text{ ms}$$

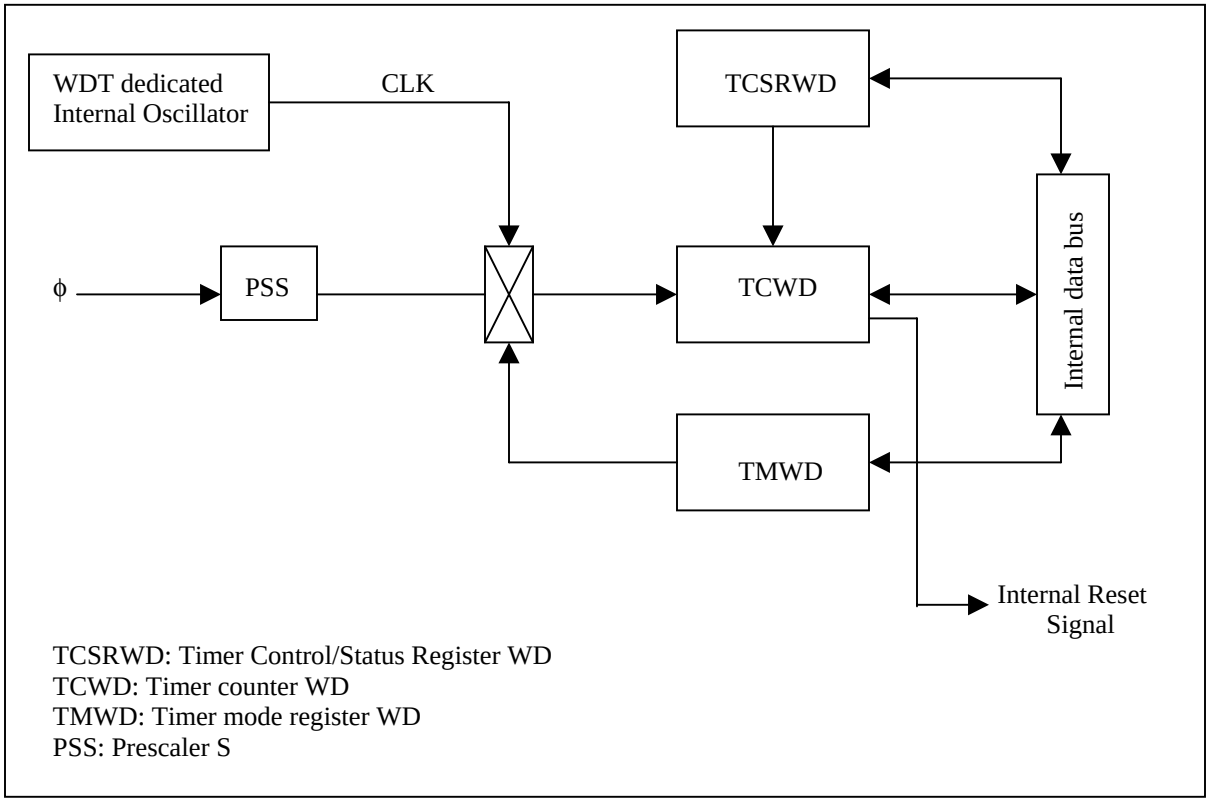


Figure 2.1 Watchdog Timer Block Diagram

3. Description of Operations

Watchdog timer operation is shown in figure 2.2.

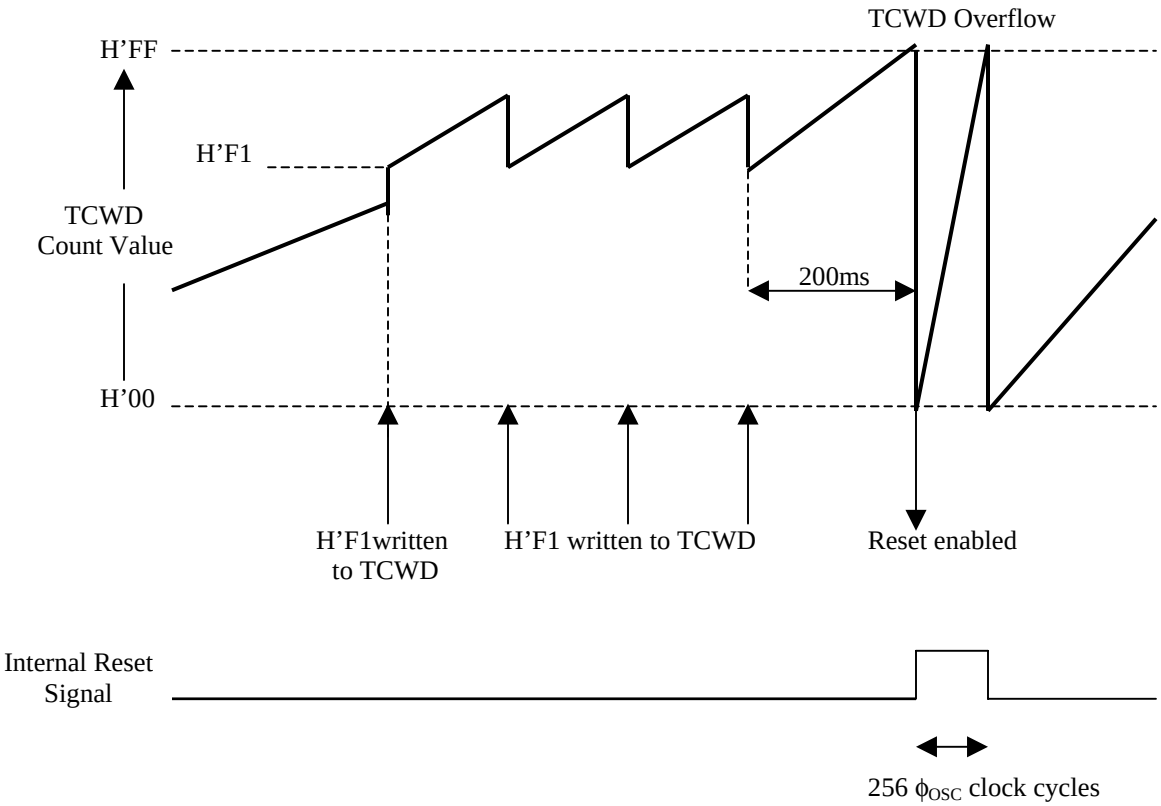


Figure 2.2 Watchdog Timer Operation

4. Description of Software

4.1 Description of Modules

Table 4.1 describes the software used in this App. Note.

Module Name	Label Name	Function
Main	main	Enables WDT, interrupts and monitors IRQ0 input pin (pin 51)'s state
Switch On	IRQ0	Sets SWONF high while IRQ0's interrupt handling routine is active

Table 4.1 Description of Modules

4.2 Description of Arguments

None are used for this AN.

4.3 Description of Internal Registers

Table 4.2 describes the internal registers used in this AN.

Register	Name	Function	Address	Bit	Setting
TCSRWD	B6WI	Timer control/status register WD (bit 6 write inhibit) If B6WI is set to 0, writing to TCWE is enabled. If B6WI is set to 1, writing to TCWE is disabled.	H'FFC0	7	1
	TCWE	Timer counter WD write enable When TWCE is set to 1, writing to TCWD is enabled	H'FFC0	6	1
	B4WI	Timer control/status register WD (bit 4 write inhibit) If B4WI is set to 0, writing to TCSRWE is enabled. If B4WI is set to 1, writing to TCSRWE is disabled.	H'FFC0	5	1
	TCSRWE	Timer control/status register write enable If TCSWRE is set to 1, writing to WDON and WRST is enabled.	H'FFC0	4	1
	B2WI	Timer control/status register WD (bit 2 write inhibit) If B2WI is set to 0, writing to WDON is enabled. If B2WI is set to 1, writing to WDON is disabled.	H'FFC0	3	1
	WDON	Watchdog timer on When WDON is set to 1, WDT is enabled	H'FFC0	2	1
	B0WI	Timer control/status register WD (bit 0 write inhibit) If B0WI is set to 0, writing to WRST is enabled. If B0WI is set to 1, writing to WRST is disabled.	H'FFC0	1	1
	WRST	Watchdog timer reset When WRST is set to 0, TCWD has not overflown When WRST is set to 1, TCWD has overflown	H'FFC0	0	1

Register	Name	Function	Address	Bit	Setting
TCWD		Timer counter WD An 8-bit Timer counter incremented by clock input of system clock/8192	H'FFC1		H'10
TMWD		Timer mode register WD (clock option 3 to 0) System clock/8192 is selected as TCWD input clock	H'FFC2		H'FF
PDR5	P55	Port data register 5 If PCR5 bits are set to 1, PDR5 is enabled to read	H'FFD8	5	0
PCR5	PCR55	Port control register 5 If PCR55 is set to 0, P55 is an input pin If PCR55 is set to 1, P55 is an output pin	H'FFE8	5	1
PMR1	IRQ0	Port mode register 1 If IRQ0 is set to 1, IRQ0 becomes an input pin	H'FFE0	4	1
IEGR1	IEG0	Interrupt edge select register 1 If IEG0 is set to 0, the falling edge of IRQ0 is chosen If IEG0 is set to 1, the rising edge of IRQ0 is chosen	H'FFF2	0	0
IENR1	IEN0	Interrupt enable register 1 If IEN0 is set to 1, IRQ0 is enabled	H'FFF4	0	1
IRR1	IRRI0	Interrupt flag register 1 If IRRI0 is set to 0, IRQ0 is not requested If IRRI0 is set to 1, IRQ0 is requested	H'FFF6	0	0

Table 4.2 Internal Registers

4.4 Description of RAM

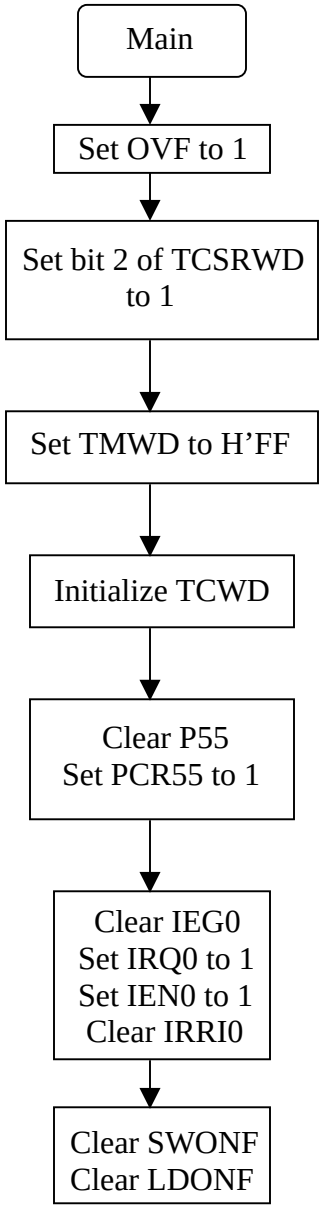
Table 4.3 describes the functionality of RAM in this AN.

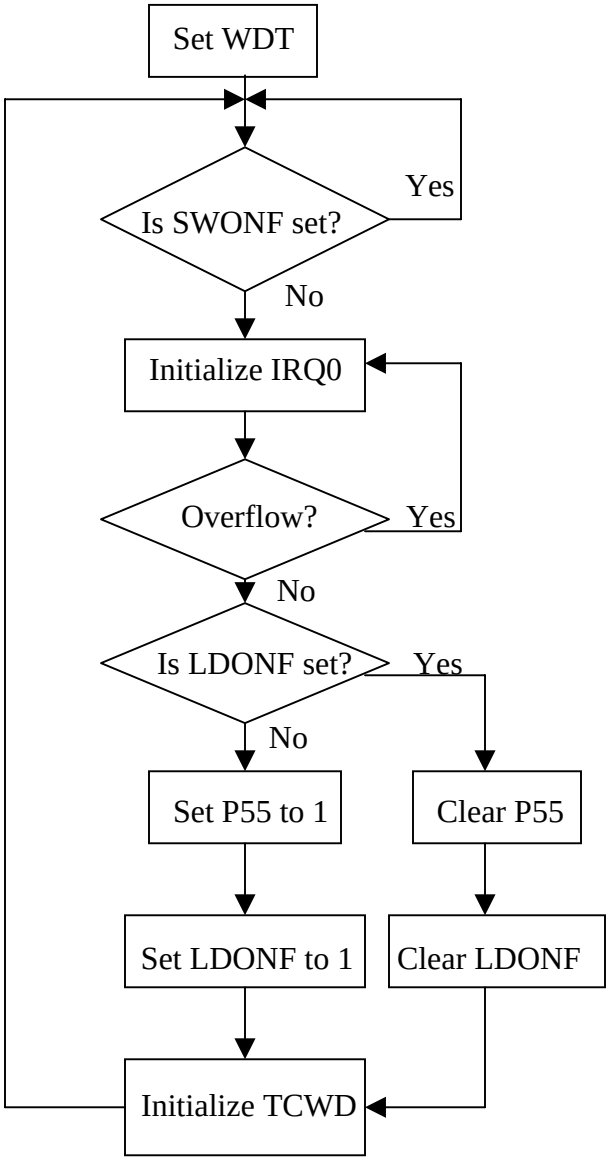
Name	Function	Address	Used in
Count-up	The counter that toggles LEDs D1 and D2	H'FB80	Main
USRF/SWONF	On/Off state of the switch	H'FB82	Main

Name	Function	Address	Used in
USRF/LDONF	On/Off state of the LEDs D1 and D2	H’FB82	Main

Table 4.3 Functionality of RAM

5. Flowcharts





6. Program Listing

Initialization (Program listing)

```
Define H
/************************************************************
Initialization Program (in Assembly)
************************************************************/
.EXPORT _INIT
.IMPORT _main
;
.SECTION I,REGISTER
_INIT:
MOV.W #H'FF80,R7
LDC.B #B'10000000,CCR
JMP @_main
;
.END
/************************************************************/
/* */
/* H8/36077 Series */
/* Application Note */
/* */
/* 'Watchdog Timer' */
/* */
/* Function */
/* */
/* External Clock : 9.83MHz */
/* Internal Clock : 9.83MHz */
/* Sub Clock : 32.768kHz */
/* */
/************************************************************/
    #include <machine.h>
/************************************************************/
/* Symbol Defnition */
/************************************************************/
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 PDR5_BIT (*(struct BIT *)0xFFD8)     /* Port Data Register 5 */
#define P55 PDR5_BIT.b5                      /* Port Data Register 5 bit5 */
#define PCR5_BIT (*(struct BIT *)0xFFE8)     /* Port Control Register 5 */
#define PCR55 PCR5_BIT.b5                    /* Port Control Register 5 bit5 */
#define IEGR1_BIT (*(struct BIT *)0xFFF2)    /* Interupt Edge Select Register 2 */
#define IEG0 IEGR1_BIT.b0                    /* IEG0 Edge Select */
#define IENR1_BIT (*(struct BIT *)0xFFF4)    /* Interrupt Enable Register 1 */
#define IEN0 IENR1_BIT.b0                    /* IEN0 Inetrrupt Enable */
#define IRR1_BIT (*(struct BIT *)0xFFF6)     /* Interrupt Request Register 1 */
#define IRRI0 IRR1_BIT.b0                    /* IRRI0 Interrupt Request Register */
#define PMR1_BIT (*(struct BIT *)0xFFE0)     /* Port Mode Register 1 */
#define IRQ0_SET PMR1_BIT.b4                 /* Port Mode Register 1 bit4 */
#pragma interrupt (IRQ0)

/************************************************************/
/* Function Definition */
/************************************************************/
extern void INIT ( void );           /* Stack Pointer set */
void main ( void );
void IRQ0 ( void );
/************************************************************/
/* RAM define */
/************************************************************/
unsigned int count_up;
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 */
P55 = 0;                            /* Initialize P55 Terminal Output Level */
PCR55 = 1;                          /* Initialize PCR55 Output Terminal Function */
IEG0 = 1;                           /* Initialize IRQ0 Terminal Input Edge */
IRRI0 = 0;                          /* Initialize IRQ0 Interrupt Request Flag */
IRQ0_SET = 1;                       /* Initialize IRQ0 Input Terminal */
IEN0 = 1;                           /* IRQ0 Interrupt Enabie */
SWONF = 0;                          /* Initialize SWONF */
LDONF = 0;                          /* Initialize LDONF */
TCSRWD = 0xF4;                      /* Watchdog Timer On */
set_imask_ccr(0);                   /* Interrupt Enable */
while(1){
while(SWONF == 1){                  /* Is SWONF set? */
;
}
counter_int = 0;                    /* counter Clear */
do{
counter_int ++;                     /* counter Countup */
}while(count_up != 0);              /* Is count_up = FFFF ? */
if(LDONF == 1){
P55 = 0; /* Turn Off LED & LED2 */
LDONF = 0; /* LDONF Clear */
}else{
P55 = 1; /* Turn On LED1 & LED2 */
LDONF = 1; /* LDONF set */
}
TCWD = 0x10; /* Initialize TCWD */
}
}
/************************************************************/
/* IRQ0 Interrupt */
/************************************************************/
void IRQ0 ( void )
{
IRRI0 = 0; /* Clear IRRI0 */
SWONF = 1; /* Set SWONF */                                  }
```

BRAZEN TEK
WHERE TECHNOLOGY AND CREATIVITY MEET

Important Notes

1. This document is provided for reference purposes only. Brazen Tek neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights.
2. Brazen Tek shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. This application note or the technology described in this document is not for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use.
4. Brazen Tek assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
5. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Brazen Tek makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Brazen Tek.