Application Note

Synchronous Operation Mode

For H8/3694 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

November 2008

Introduction

One of the functionality of Timer W is to generate waveforms based on pulse width techniques. This application is to describe Timer W based on synchronous mode to obtain PWM waveforms as output.

Target Device

H8/3694 group

Contents

1. Specifications

i. Deploy Timer W based on synchronous mode to obtain PWM waveforms as output.
ii. The output pulses generated by half of their cycle, otherwise known as high-level widths are to control their duty cycle.
iii. Settings in registers would enable complete duty cycle variation between 0% and 100%.

$$Duty\,Cycle = \frac{High\,level\ pulse\,width}{Pulse\,Period} \times 100\,(\%)$$

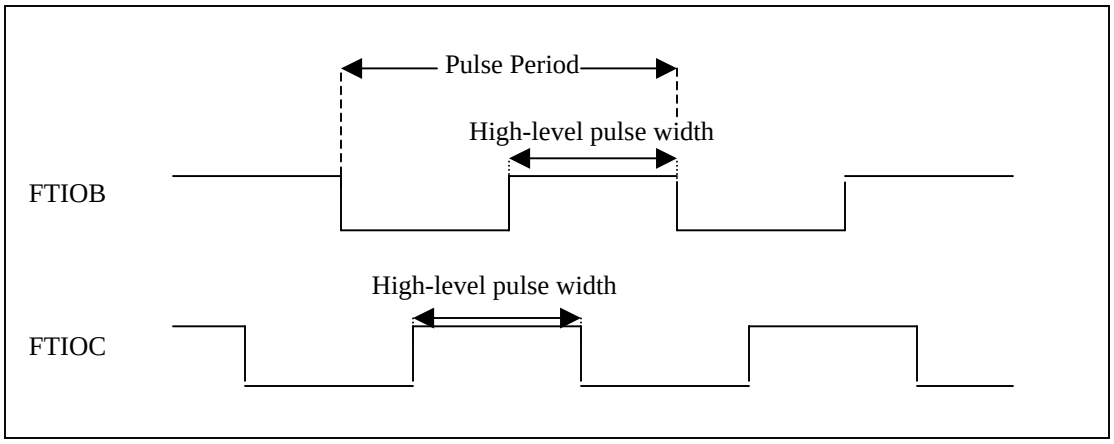


Figure 1 – PWM waveforms

2. Description of Functions used

This application note is to PWM waveforms generated by Timer W in synchronous mode. Figure 2 shows the block diagram of this operation. The following functions are being used:
- Counter clearing function upon an output compare.
- Two channels 0 and 1 of Timer W are used to generate PWM waveforms.

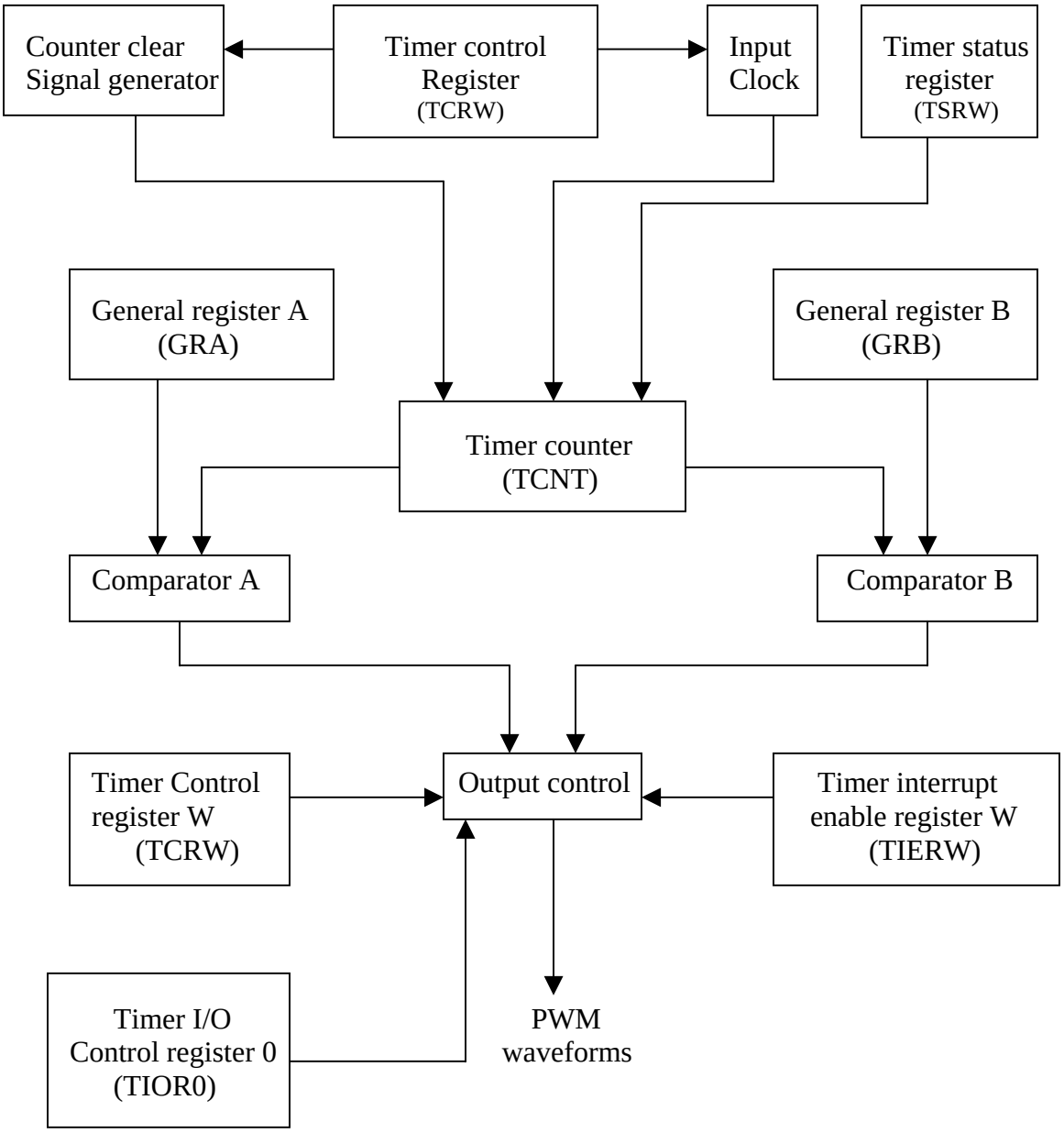


Figure 2 – Timer W synchronous operation mode

3. Description of Operations

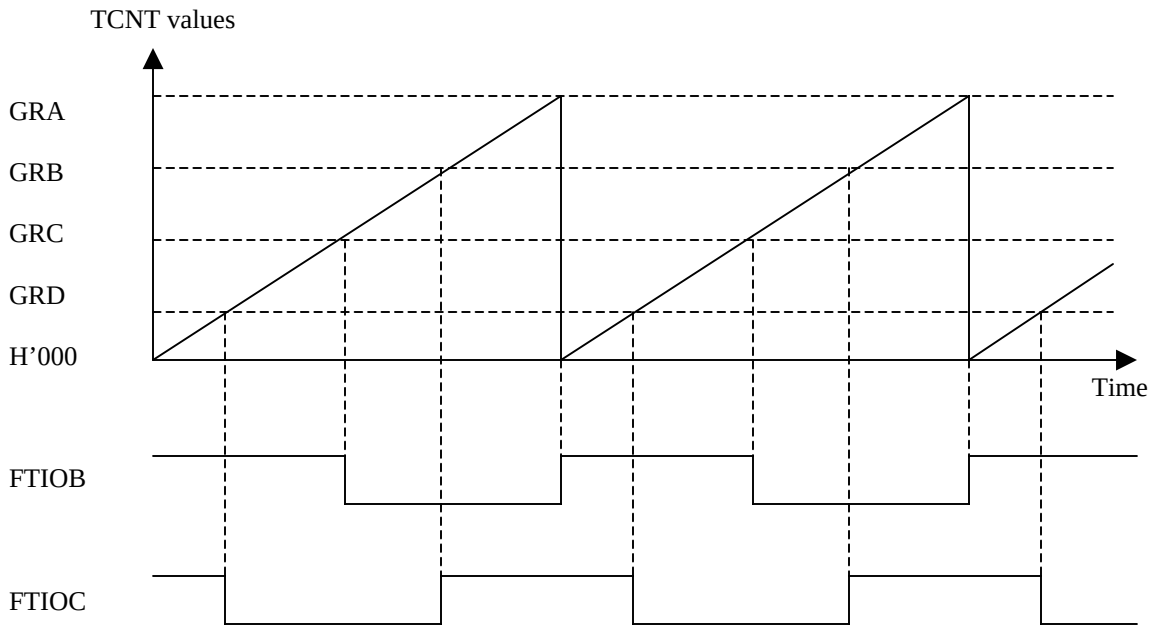


Figure 3 – Synchronous operation of Timer W

4. Description of Software

4.1 Description of Modules

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

Module Name	Label Name	Function
Main	main	Sets timing of general registers. Using Synchronous operation, it generates PWM waveforms

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.

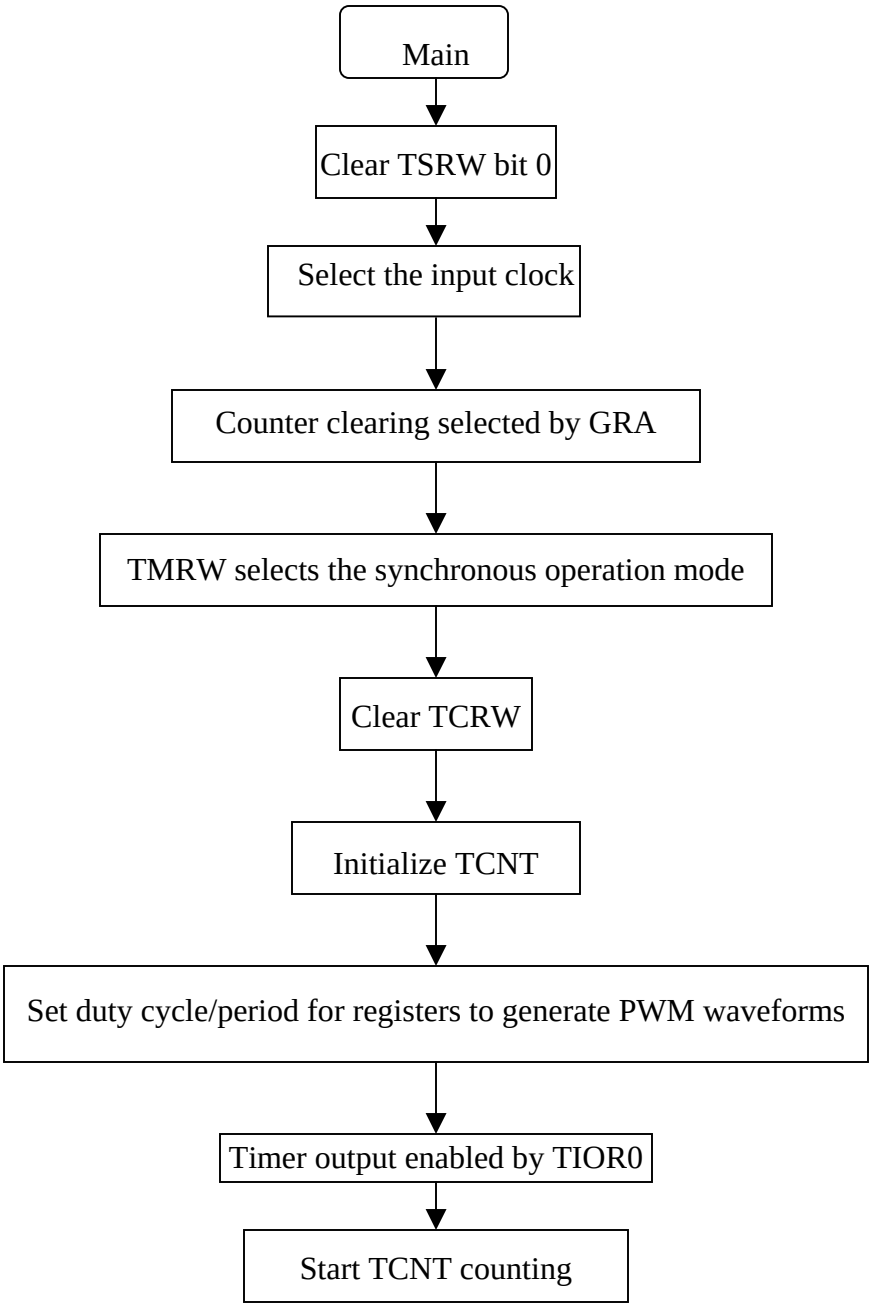
Register/Name	Function
TSRW	Enables/Disables Channels 0 or 1 counter timer operation
TCNT	Timer counter
TMRW	Set for synchronous operation mode
TCRW	Triggers input clock and resets channel 0 of timer counter
TIOR0	Enables/Disables timer output
TIERW	Sets initial output value when output compare has occurred
GRA	Sets a duty cycle for one of the PWM waveforms
GRB	Sets a duty cycle for one of the PWM waveforms
GRC	Sets a period for one of the PWM waveforms
GRD	Sets a period for one of the PWM waveforms

Table 4.2 - Internal Registers

4.4 Description of RAM

None used for this AN

5. Flowcharts



6. Program Listing

Initialize source (Program list)

```
.EXPORT_INIT
.IMPORT_main
;
.SECTION P,CODE
_INIT:
MOV.W #H'FF80,R7
LDC.B #B'10000000,CCR
JMP @_main
;
.END
/************************************************************/
/* */
/* H8/3694 group MCU */
/* Application Note */
/* */
/* 'Synchronous Operation Mode'                */
/*                                             */
/* :Timer W Synchronous Mode                   */
/*                                             */
/* External Clock: 16 MHz                      */
/* Internal Clock: 16 MHz                      */
/* 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 TCRW *(volatile unsigned char *)H'FF81 /* Timer Control Register
W */
#define TIOR0 *(volatile unsigned char *)H'FF84 /* Timer I/O Control
Register 0 */
#define TIOR1 *(volatile unsigned char *)H'FF85 /* Timer I/O Control
Register 1 */
#define TSRW *(volatile unsigned char *)H'FF83 /* Timer Status Register
W */
#define IMIFA_0 TSR_0_BIT.b0 /* Input Capture/Compare match Flag A
Channel 0 */
#define TIERW *(volatile unsigned char *)H'FF82 /* Timer Interrupt
Enable Register W */
#define TCNT *(volatile unsigned short *)H'FF86 /* Timer Counter */
#define GRA *(volatile unsigned short *)H'FF88 /* General Register A */
#define GRB *(volatile unsigned short *)H'FF8A /* General Register B */
#define GRC *(volatile unsigned short *)H'FF8C /* General Register C */
#define GRD *(volatile unsigned short *)H'FF8E /* General Register D */
#define TMRW *(volatile unsigned char *)H'FF80 /* Timer Mode Register W
*/
/************************************************************/
/* Function Definition */
/************************************************************/
extern void INIT ( void ); /* SP Set */
void main ( void );
extern void _INITSCT();
/************************************************************/
/* Vector Address */
/************************************************************/
#pragma section V1 /* VECTOR SECTION SET */
void (*const VEC_TBL1[])(void) = { /* H'00 - H'0F */
INIT                                    /* 00 Reset
};
#pragma section                         /* P */
/************************************************************/
/* Main Program                         */
/************************************************************/
void main ( void )
```

```
{
_INITSCT();


set_imask_ccr(1);                    /* Disable interrupts */
TSRW = H'FC;                         /* Timer Stop */
TIOR0 = H'00;                        /* PWM initial output "0" */
TCRW = H'20;                         /* GRA Compare match Clear
Mode */
GRA = H'320;                         /* Period: Cycle 50us */
GRB = H'230;                         /* Duty cycle 30% */
GRC = H'230;                         /* Duty cycle 30% */
GRD = H'230;                         /* Duty cycle 30% */
TIERW = H'F1;                        /* FTIOB, FTIOC, FTIOD Output
Enable */
TSRW = H'FD;                         /* TCNT Start */
set_imask_ccr(0);                    /* Interrupt Enable */
while(1) {
;
}
}
```

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.