To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: http://www.renesas.com

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

H8/300L

PWM Sine Wave Generation (SineWave)

Introduction

Pulse Width Modulation (PWM) is a powerful technique for driving analog circuits with micro-controller's digital outputs. It is popular in areas such as DC motor drive control and digital-to-analog conversion in bit stream DACs. This application note demonstrates the generation of sine wave using PWM at a single frequency. For the detailed description of DAC, please refer to application note 'PWM as A DAC'.

Target Device

H8/300L Super Low Power (SLP) Series – H8/38024F

Contents

1. Overview

Pulse Width Modulation (PWM) refers to a form of signal modulation where data is represented by the ratio of the on time to the total time (known as the duty cycle). PWM has the property where the instantaneous DC component is directly proportional to the duty cycle. It is primarily used for controlling digital encoded analog signal of varying amplitude.

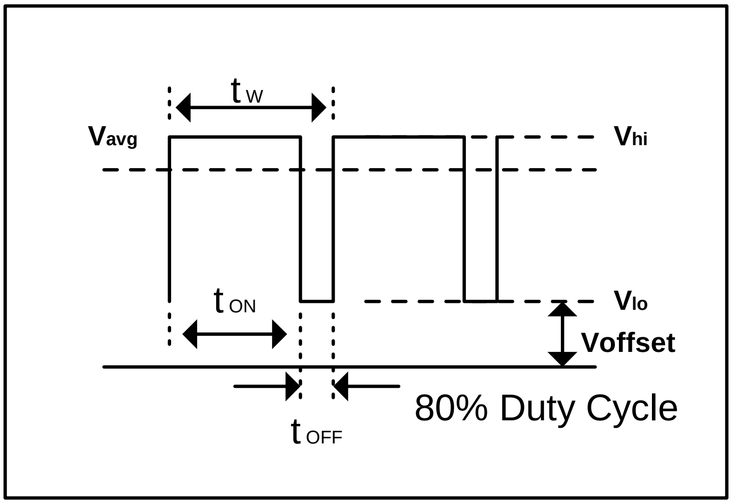

Fig 1.1 Duty Cycle and Frequency of PWM

The relationship between the time-average voltage (V_{avg}) the high and low voltages of the square wave (V_{hi} and V_{lo}) and the duty cycle (D) in percent is as follows:

$V_{avg} = (V_{hi} - V_{lo}) * D + V_{offset}$ ……………………………………………………………[1]

where $D = t_{ON} / t_W$ or D = PWDR value/1023 (for 10-bit PWM) ……………………………[2]

If V_{hi} is 5V, V_{lo} is 0V (i.e. $V_{offset} = 0$) and D is 80%, V_{avg} would be 4 V.

Another important parameter of PWM is the frequency. It is defined by number of pulse per second.

$f = 1 / t_W$……………………………………………………………………………………..[3]

where $t_W = t_{ON} + t_{OFF}$………………………………………………………………………...[4]

From equations [2] and [3], it can be proven that

$F = D / t_{ON}$……………………………………………………………………………….......[4]

2. PWM Architecture

The H8/38024F series micon has two on-chip 10-bit PWMs, designed as PWM1 and PWM2, with identical functions. It offers features including four conversion periods of 4096/Ø, 2048/Ø, 1024/Ø and 512/Ø, pulse division method for less ripple and module standby mode for power saving. There are 20 choices for input clock to the PWM.

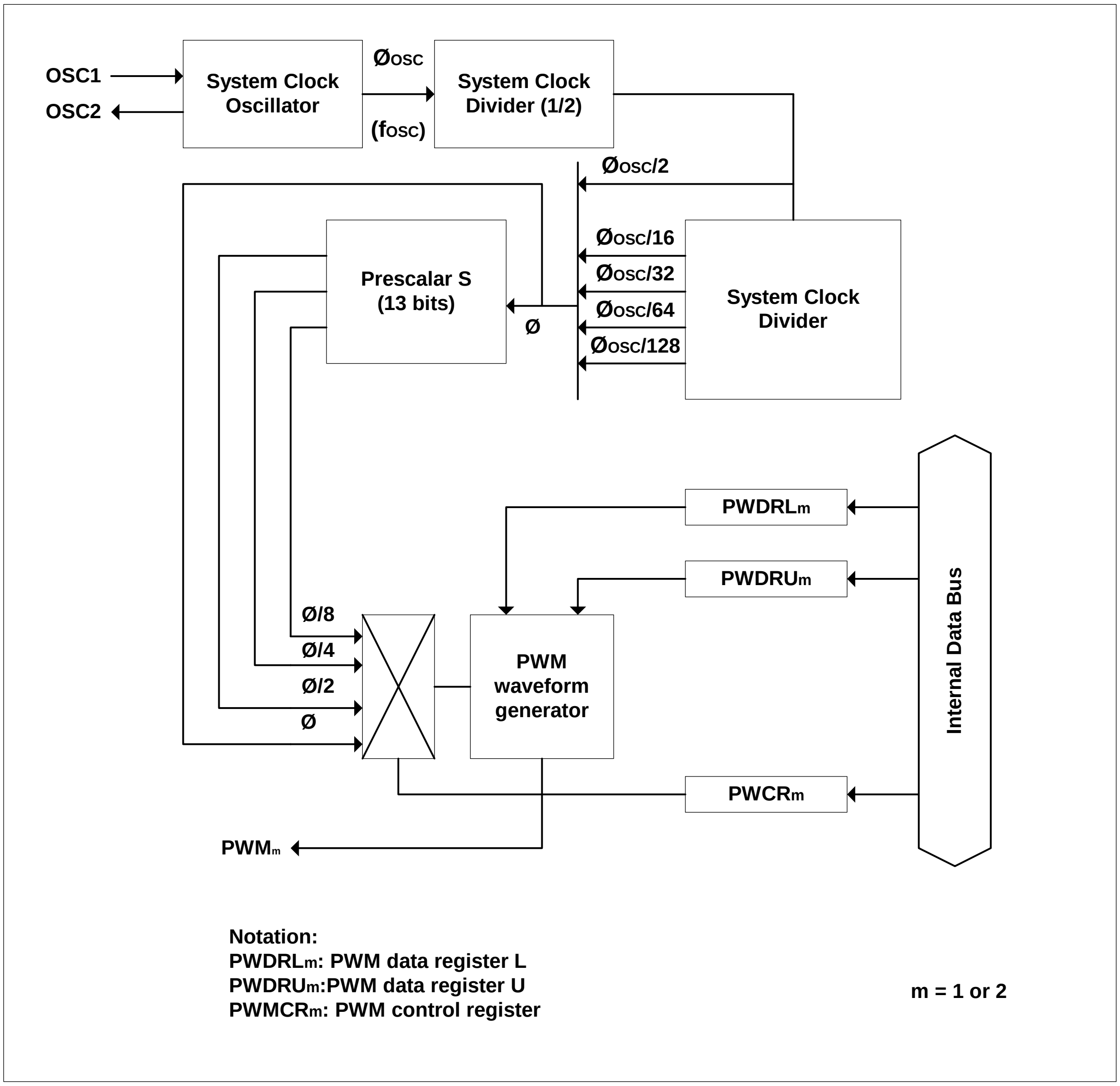

Figure 2.1 Block Diagram of the 10-bit PWM

2.1 Register Configuration

Name	Abbreviation	R/W	Initial Value	Address
PWM1 Control Register	PWCR1	W	H'FC	H'FFD0
PWM1 Data Register U	PWDRU1	W	H'FC	H'FFD1
PWM1 Data Register L	PWDRL1	W	H'00	H'FFD2
PWM2 Control Register	PWCR2	W	H'FC	H'FFCD
PWM2 Data Register U	PWDRU2	W	H'FC	H'FFCE
PWM2 Data Register L	PWDRL2	W	H'00	H'FFCF
Clock Stop Register 2	CKSTPR2	R/W	H'FF	H'FFFB

Table 2.1 Register Configuration

Port Mode Register (PMR9)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	-	**-**	**-**	**-**	**PIOFF**	**-**	**PWM_2**	**PWM_1**
Initial Value	**1**	**1**	**1**	**1**	**0**	**-**	**0**	**0**
Read/Write	**-**	**-**	**-**	**-**	**R/W**	**W**	**R/W**	**R/W**

PWR9 is an 8-bit read/write register controlling the selection of the $P9_0$ and $P9_1$ pin functions.

Bits 3: P92 to P90 step-up circuit control (PIOFF)

Bit 3 turns the $P9_2$ to $P9_0$ step-up circuit on and off.

PIOFF	**Description**
0	Large-current port step-up circuit is turned on
1	Large-current port step-up circuit is turned off

Bit 2: Reserved bit

This bit is reserved; it can only be written with 0.

Bits 1 and 0: $P9_n$/PWM pin function switches

These pins select whether pin $P9_n/PWM_{n+1}$ is used as $P9_n$ or as PWM_{n+1}.

WKP_{n+1}	Description
0	Functions as $P9_n$ output pin
1	Functions as PWM_{n+1} output pin

PWM Control Register (PWCRm)

Bit	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	PWCRm1	PWCRm0
Initial Value	1	1	1	1	1	1	0	0
Read/Write	-	-	-	-	-	-	W	W

PWCRm is an 8-bit write-only register for input clock selection. Upon reset, PWCRm is initialized to H'FC.

Bits 7 to 2: Reserved bits.

Bits 1 and 0: Clock select 1 (PWCRm1, PWCRm0)

Bits 1 and 0 select the clock supplied to the 10-bit PWM. These bits are write-only bits; they are always read as 1.

Bit 1 PWCRm1	Bit 0 PWCRm0	Description
0	0	The input clock is Ø (tØ * = 1/Ø) The conversion period is 512/ Ø, with a minimum modulation width 1/2Ø.
0	1	The input clock is Ø/2 (tØ * = 2/Ø) The conversion period is 1024/ Ø, with a minimum modulation width 1/Ø.
1	0	The input clock is Ø/4 (tØ * = 4/Ø) The conversion period is 2048/ Ø, with a minimum modulation width 2/Ø.
1	1	The input clock is Ø/8 (tØ * = 8/Ø) The conversion period is 4096/ Ø, with a minimum modulation width 4/Ø.

*: Period of PWM input clock.

PWM Data Registers U and L (PWDRUm, PWDRLm)

PWDRUm

Bit	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	PWDRUm1	PWDRUm0
Initial Value	1	1	1	1	1	1	0	0
Read/Write	-	-	-	-	-	-	W	W

PWDRLm

Bit	7	6	5	4	3	2	1	0
	PWDRLm7	PWDRLm6	PWDRLm5	PWDRLm4	PWDRLm3	PWDRLm2	PWDRLm1	PWDRLm0
Initial Value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

PWDRUm and PWDRLm form a 10-bit write-only register, with the upper 2 bits assigned to PWDRUm and the lower 8 bits to PWDRLm. The value written to PWDRUm and PWDRLm gives the total high-level width of one PWM waveform cycle.

When 10-bit data is written to PWDRUm and PWDRLm, the register contents are latched in the PWM waveform generator, updating the PWM waveform generation data. The 10-bit data should always be written in the following sequences:

1. Write the lower 8 bits to PWDRLm
2. Write the upper 2 bits to PWDRUm for the same channel

PWDRUm and PWDRLm are write-only registers. If they are read, all bits are read as 1.

Upon reset, PWDRUm is initialized to H'FC and PWDRLm to H'00.

Clock Stop Register 2 (CKSTPR2)

Bit	7	6	5	4	3	2	1	0
	-	-	-	PW2CKSTP	AECKSTP	WDCKSTP	PW1CKSTP	LDCKSTP
Initial Value	1	1	1	1	1	1	1	1
Read/Write	-	-	-	R/W	R/W	R/W	R/W	R/W

CKSTPR2 is an 8-bit read/write register that performs module standby mode control for peripheral modules. Only the bit relating to the PWM is described here.

Bits 4 and 1: PWM module standby mode control.

PWmCKSTP	Description
0	PWMm is set to module standby mode
1	PWMm module standby mode is cleared

2.2 Basic Operation

To use the 10-bit PWM, set the registers in the following 3 steps:

1. **Select PWM channel:** Set PWM1 or PWM2 in PMR9 to 1 for PWM channel to be used, so that pin P90/PWM1 or P91/PWM2 is designated as the PWM output pin.

2. **Select Conversion Period:** Set bits PWCRm1 and PWCRm0 in the PWM control register (PWCRm) to select a conversion period.

3. **Set Pulse Width:** Set output waveform data in PWDRUm and PWDRLm. Data should be first written to PWDRLm and then to PWDRUm for the same channel.

One conversion period consists of 4 pulses, as shown in Fig. 2.2. The total of the high-level pulse widths during this period (T_H) corresponds to the data in PWDRUm and PWDRLm. The waveform will be changed at the next conversion period.

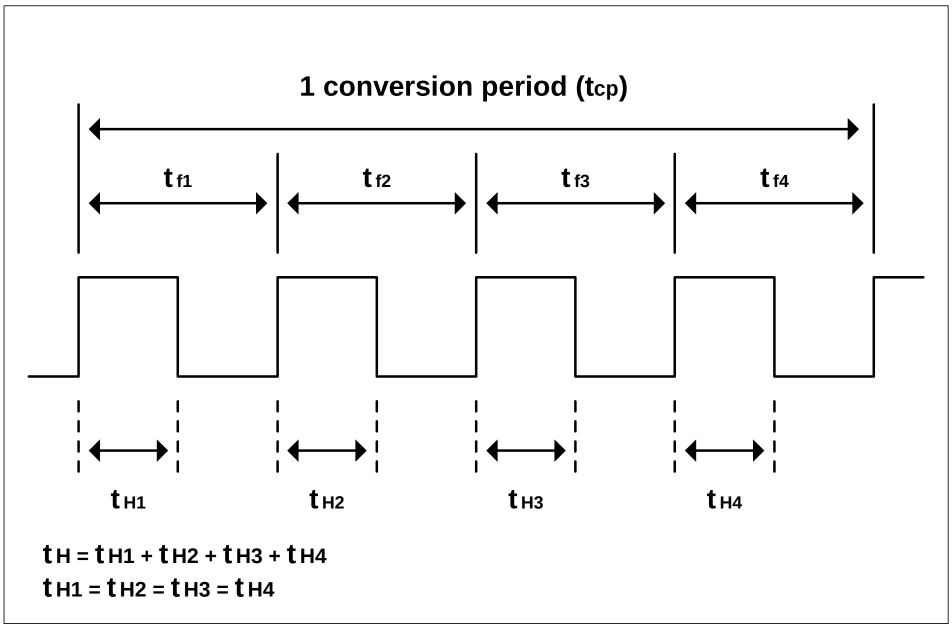

Fig 2.2 PWM Output Waveform

T_H = (data value in PWDRUm and PWDRLm + 4) * $t_Ø/2$

Where $t_Ø$ is the PWM input clock period: 1/Ø, 2/Ø, 4/Ø or 8/Ø

3. Theory Of Sine Wave Generation

The basic working principle is based on the generation of DC voltage level. (Details can be found in AN entitle PWM as a DAC). If the generated DC voltage level is in a sinusoidal manner, a sine wave is generated.

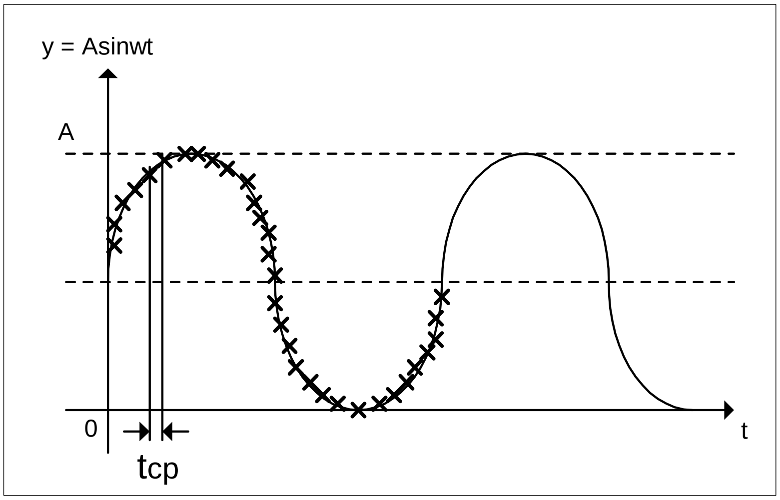

Figure 2.3 A typical Sine Wave

Based on a fixed Conversion Period (step 2), and when the Duty Cycle (step 3) is changed, the DC level voltage will be changed in the next conversion period. For a sine wave of 128 points, between AVcc (at amplitude A) and AVss (at zero ground),

The period of the sine wave, T = 256 * $t_{cp,}$ where t_{cp} is the conversion period

From AVss to AVcc, it will take 256/2 = 128 conversion periods. If the step change is uniform and equals to AVcc/128, then a triangular wave is generated.

There are two considerations in sine wave generation:

- Frequency of waveform: This depends on the choices of main clock, system clock divider, PWM divider and number of sampling points for the sine wave.

- Shape of sine wave: This depends only on the number of sampling points.

A 256 data points of sine wave is generated and stored into an array LUT[i], as according to the equation below:

LUT[i] = (int)(sin(2*i* π/256)*512+511)

n	Y = Sin(n*2*π/256)	LUT[n] = Y * 512 + 511
0	0	511
2	0.0245	524
3	0.0491	562
4	0.0736	587
:	:	:
252	-0.0980	461
253	-0.0736	473
254	-0.0491	488
255	-0.0245	498

Table 2.1 Calculation of a 256 data points sine wave

4. Program Overview

In this program, the main clock ($Ø_{osc}$) is chosen to be 10 MHz, divided by 2 (for active mode)

SYSCR1 = H'07

During the initialization, P91 is configured as PWM2 output pin and the input clock to PWM2 is set at Ø. At Ø = 5 MHz, the conversion period, 512/Ø = 102.4 us.

PMR9 = H'F2

PWCR2 = H'FC

Interrupt bit in CCR is masked to enable the initialization of Asynchronous Event Counter (AEC). AEC is an event counter, which generates interrupt once overflows, and charges the analog output at every conversion period. As the AEC is configured as 2 independent 8-bit counters, only OVL flag will be set when overflow occurs.

ECCSR = H'10 [To set ECH, ECL as 2 independent 8-bit event counter channels]

ECCR = H'90 [To select clock Ø/4 used by ECH and Ø/2 used by ECL]

AEGSR = H'40 [To sense rising edge at AEVH and AEVL]

ECCSR = H'1F [To enable ECH, ECL event clk i/p. 0-ECL & 0-ECH are set, 1-stop reset and count-up]

IRR2 = H'00 [To clear IRREC flag]

IENR2 = H'01 [To enable AEC interrupt request]

As the program runs, each AEC interrupt will cause OVL flag to change, and enable Function storeCount(LUT[i]) to load the next analog signal to the PWM. When the counter i points to end of the LUT table, a sine wave is generated.

For PWM, the overflow period of ECL is

$2/Ø * 2^8 = 5M/2 * 256 = 102.4$ us (same as the conversion period of PWM2)

5. Program Flowchart

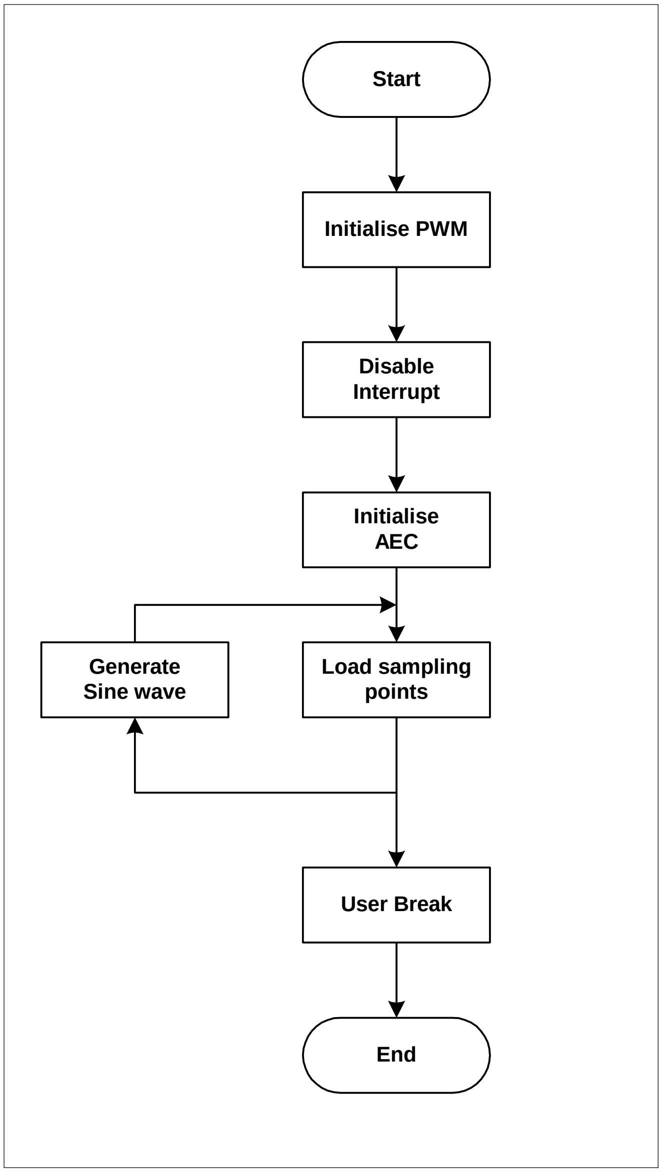

Figure 6.1 Flowchart of Sine Wave Generation

6. Software Listing

```
/****************************************************************************/
/*                                                                          */
/*  FILE:           PWM_Sinewave.c                                          */
/*  DATE:           Thu, Mar 20, 2003                                       */
/*  DESCRIPTION:    Using PWM to generate sine wave                         */
/*  CPU TYPE:       H8/38024F                                               */
/*                                                                          */
/*  This file is generated by Hitachi Project Generator (Ver.2.1).     */
/*                                                                          */
/****************************************************************************/


#include <machine.h>
#include "iodefine.h"
#include <math.h>
/****************************************************************************/
/*  Function define                                                         */
/****************************************************************************/


void init_PWM(unsigned char);
void storeCount(unsigned short);
void aecint( void );
void init_AEC(unsigned char,unsigned char,unsigned char,unsigned char,unsigned char);


/****************************************************************************/
/*  RAM define                                                              */
/****************************************************************************/
unsigned char PWDR_L2, PWDR_U2, r=0;
unsigned int i=0, frequency = 21;
unsigned int final=0, lowcnt=0;
float pi=3.141596;


unsigned int LUT[256];
```

```
/******************************************************************************/
/*  Main Program                                                              */
/******************************************************************************/
void main ( void )
{
   init_PWM(0);                              // select PWM2 to have (512/5Mhz) conversion period
   set_imask_ccr(1);                         // Interrupt Disable
   init_AEC(2,1,1,1,1);

   for (i=0; i<256; i++)
   {
         LUT[i] = (int)(sin(2*i*pi/256)*512+511);
   }
   while (1)
  {
              ;
   }
}

void init_PWM(unsigned char selClk2)
{
   if (selClk2 <= 3)                         // Check if valid, otherwise PWM2 is off
   {         P_IO.PMR9.BIT.PWM2 = 1;         // Configure P91 as PWM2 output pin
             P_PWM2.PWCR2.BYTE = selClk2;    // Clock select for PWM2,write only
   }
}

/**************************************************************/
/*  Write each digital code into PWDR registers              */
/**************************************************************/
void storeCount(unsigned short PWDRval_2)
{
   P_PWM2.PWDRL2.BYTE = (unsigned char)(PWDRval_2 & 0x00FF);      // Write lower 8bits of 10bits data
   P_PWM2.PWDRU2.BYTE = (unsigned char) ((PWDRval_2 & 0x0300) >> 8);       // Write upper 8bits of 10bits data
}
```

```
/************************************************************/
/*  AEC Interrupt Service Routine                             */
/************************************************************/
void aecint (void)
{
   P_SYSCR.IRR2.BIT.IRREC = 0;                               // Clear IRREC flag

   if(P_AEC.ECCSR.BIT.OVL == 1)                              // Check for ECL overflow flag
   {          P_AEC.ECCSR.BIT.OVL = 0;                  // Clears flag

              final = LUT[lowcnt];
              storeCount(final);                             // Write Sinewave digital code into PWM registers
              lowcnt += frequency;
              if(lowcnt>255) lowcnt = lowcnt-256;            // If reached end of 1period, then reset
              }
}
// Only OVL flag will set when overflow occurs
void init_AEC(unsigned char csH,unsigned char csL,unsigned char esH,unsigned char esL,unsigned char intr)
{
   P_AEC.ECCSR.BIT.CH2 = 1;                             // ECH & ECL as independent 8-bit counter
   P_AEC.ECCR.BIT.ACKH = csH;                                // 0x00:AEVH as i/p
                                                             // 0x01:divide by 2
                                                             // 0x02:divide by 4
                                                             // 0x03:divide by 8

   P_AEC.AEGSR.BIT.AHEGS = esH;                              // 0x00:Falling edge sense on AEVH pin
                                                             // 0x01:Rising edge sense on AEVH pin
                                                             // 0x02:Both edges sense on AEVH pin
                                                             // 0x03:AEVH pin usage prohibited

   P_AEC.ECCR.BIT.ACKL = csL;                                // 0x00:AEVL as i/p
                                                             // 0x01:divide by 2
                                                             // 0x02:divide by 4
                                                             // 0x03:divide by 8
```

```
P_AEC.AEGSR.BIT.ALEGS = esL;            // 0x00:Falling edge sense on AEVL pin
                                        // 0x01:Rising edge sense on AEVL pin
                                        // 0x02:Both edges sense on AEVL pin
                                        // 0x03:AEVL pin usage prohibited

P_AEC.ECCSR.BIT.CUEL = 0;               // ECH event clk i/p is Disable, ECL value held
P_AEC.ECCSR.BIT.CUEH = 0;               // ECL event clk i/p is Disable, ECH value held
P_AEC.ECCSR.BIT.CRCL = 0;               // 0-ECL is reset, 1-stop reset & countup
P_AEC.ECCSR.BIT.CRCH = 0;               // 0-ECH is reset, 1-stop reset & countup

P_AEC.ECCSR.BIT.OVL = 0;                // Clears flag
P_AEC.ECCSR.BIT.OVH = 0;                // Clears flag

P_AEC.ECCSR.BIT.CUEL = 1;               // ECH event clk i/p is Enabled
P_AEC.ECCSR.BIT.CUEH = 1;               // ECL event clk i/p is Enabled
P_AEC.ECCSR.BIT.CRCL = 1;               // 0-ECL is reset, 1-stop reset & countup
P_AEC.ECCSR.BIT.CRCH = 1;               // 0-ECH is reset, 1-stop reset & countup

P_SYSCR.IRR2.BIT.IRREC = 0;             // Clear IRREC flag
P_SYSCR.IENR2.BIT.IENEC = intr;         // AEC Interrupt Request, 1-Enable, 0-Disable

set_imask_ccr(0);                       // Interrupts, 0-Enable, 1-Disable
// set_imask_ccr() comes as a pair
```

Reference

1. H8/38024 Series, H8/38024F-ZTAT™ Hardware Manual
2. H8/300L Super Low Power Series – Low-cost CPU Board CPUBD-38024F User's Manual

Revision Record

Rev.	Date	Description	
		Page	Summary
1.00	Sep.03	-	First edition issued

Keep safety first in your circuit designs!

1. Renesas Technology Corporation puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage.
   Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of nonflammable material or (iii) prevention against any malfunction or mishap.

Notes regarding these materials

1. These materials are intended as a reference to assist our customers in the selection of the Renesas Technology Corporation product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Renesas Technology Corporation or a third party.
2. Renesas Technology Corporation assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
3. All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corporation without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corporation or an authorized Renesas Technology Corporation product distributor for the latest product information before purchasing a product listed herein.
   The information described here may contain technical inaccuracies or typographical errors. Renesas Technology Corporation assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors.
   Please also pay attention to information published by Renesas Technology Corporation by various means, including the Renesas Technology Corporation Semiconductor home page (http://www.renesas.com).
4. When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corporation assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
5. Renesas Technology Corporation semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Renesas Technology Corporation or an authorized Renesas Technology Corporation product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
6. The prior written approval of Renesas Technology Corporation is necessary to reprint or reproduce in whole or in part these materials.
7. If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.
   Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
8. Please contact Renesas Technology Corporation for further details on these materials or the products contained therein.