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 as a DAC (DAC)

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 ways to generate different voltage outputs using PWM.

Target Device

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

Contents

1. Overview 3

2. PWM Architecture 4
2.1 Register Configuration 5
2.2 Basic Operation 9

3. Possible Settings 10
3.1 Conversion Period (Step 2) 10
3.2 Duty Cycle (Step 3) 12

4. Theory of DAC 13

5. Circuit Diagram 14

6. Further Measurement 15

7. Program Overview 17

8. Program Flowchart 18

9. Software Listing 19

Reference 22

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.

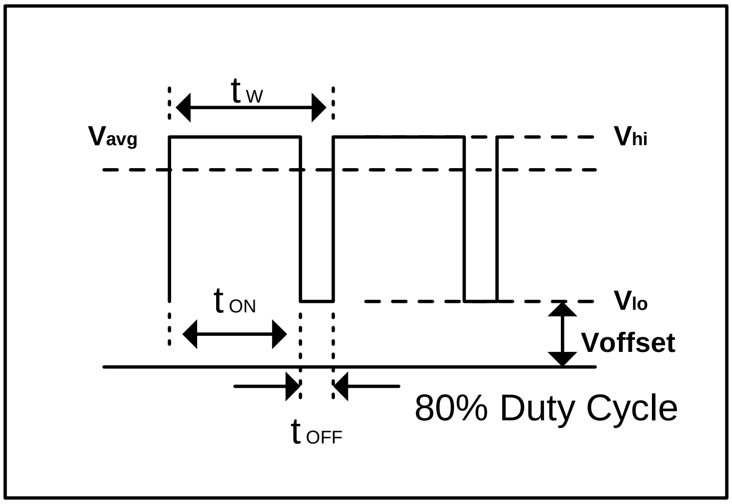

Figure 1 Duty Cycle and Frequency of PWM

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.

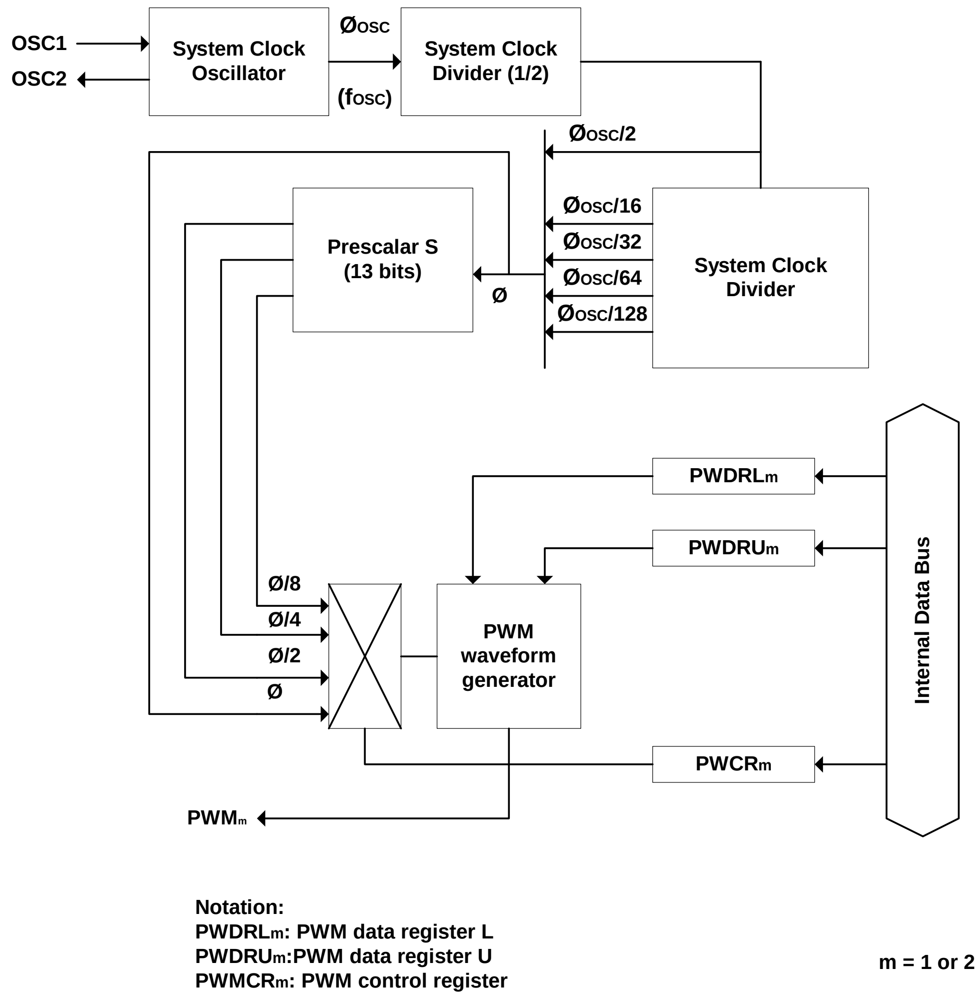

Figure 2 Block Diagram of the 10-bit PWM

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 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.

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.

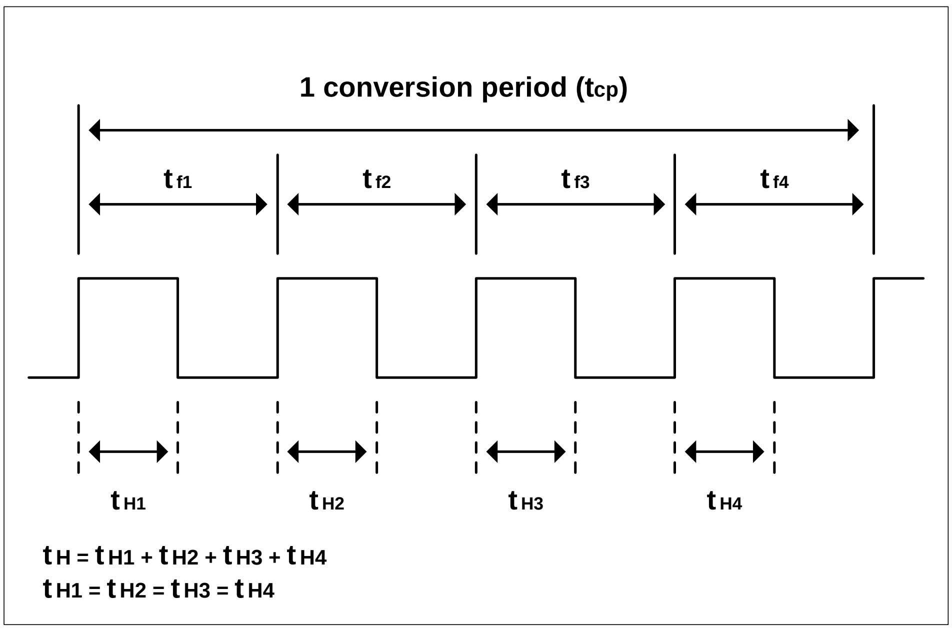

Figure 3 PWM Output Waveform

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. Possible Settings

3.1 Conversion Period (Step 2)

Assuming 50% duty cycle,

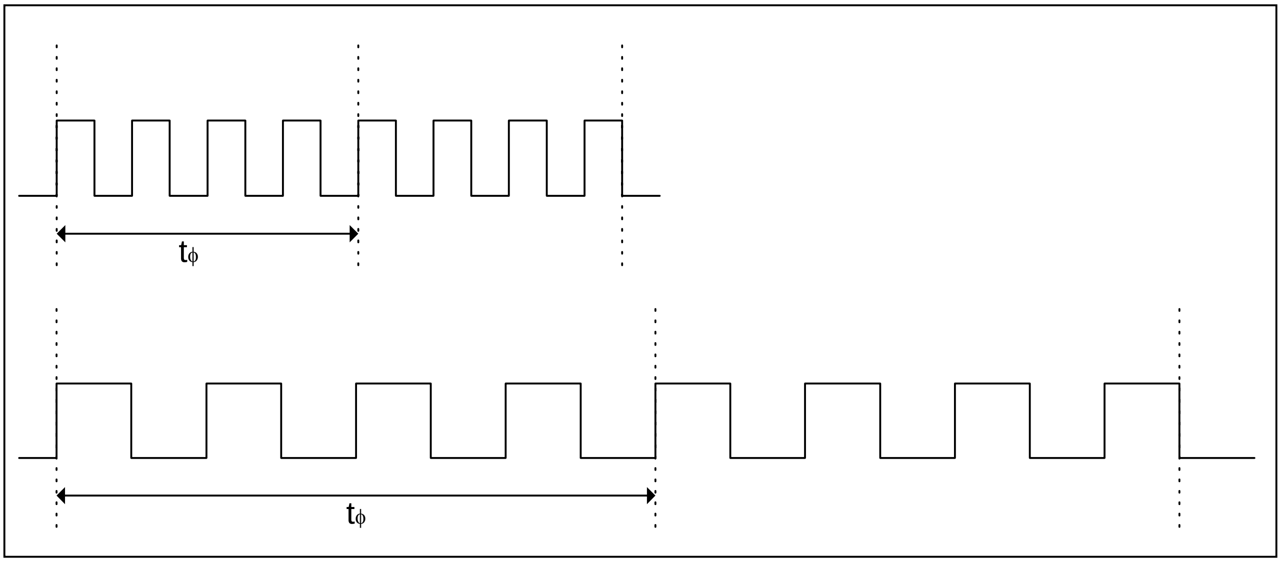

Figure 4 Conversion Period

Based on a fixed crystal input clock ($Ø_{osc}$), with the fixed system clock divider (1/2) and system clock divider, the 4 possible input clock settings for PWM to set to the different conversion periods are Ø, Ø/2, Ø/4 and Ø/8.

The user can change the system clock divider with the 5 options: $Ø_{osc}/2$, $Ø_{osc}/16$, $Ø_{osc}/32$, $Ø_{osc}/64$ and $Ø_{osc}/128$, and thus with a total of 5*4 = 20 possible settings.

For instance, based on a 10 MHz main clock, the effective 10 choices of input clock to PWM are 5000, 2500, 1250, 625, 312.5, 156.25, 78.125, 39.0625, 19.53125 and 9.765625 kHz. The detailed calculations are listed in the Table 3.1.

Main Clock, $Ø_{osc}$ (MHz)	System Clock Divider	Input Clock to Prescaler (S), Ø (kHz)	PWM Divider	Input Clock to PWM (kHz)	Conversion Period (us)
10	2	5000	1	5000	0.2
		5000	2	2500	0. 4
		5000	4	1250	0. 8
		5000	8	625	1.6
	16	625	1	625	1.6
		625	2	312.5	3.2
		625	4	156.25	6.4
		625	8	78.125	12.8
	32	312.5	1	312.5	3.2
		312.5	2	156.25	6.4
		312.5	4	78.125	12.8
		312.5	8	39.0625	25.6
	64	156.25	1	156.25	6.4
		156.25	2	78.125	12.8
		156.25	4	39.0625	25.6
		156.25	8	19.53125	51.2
	128	78.125	1	78.125	12.8
		78.125	2	39.0625	25.6
		78.125	4	19.53125	51.2
		78.125	8	9.765625	102.4

Table 2 put clock selection to PWM

3.2 Duty Cycle (Step 3)

Assuming the conversion period is fixed, when the PWDRUm and PWDRLm are changed, the pulse width will be changed

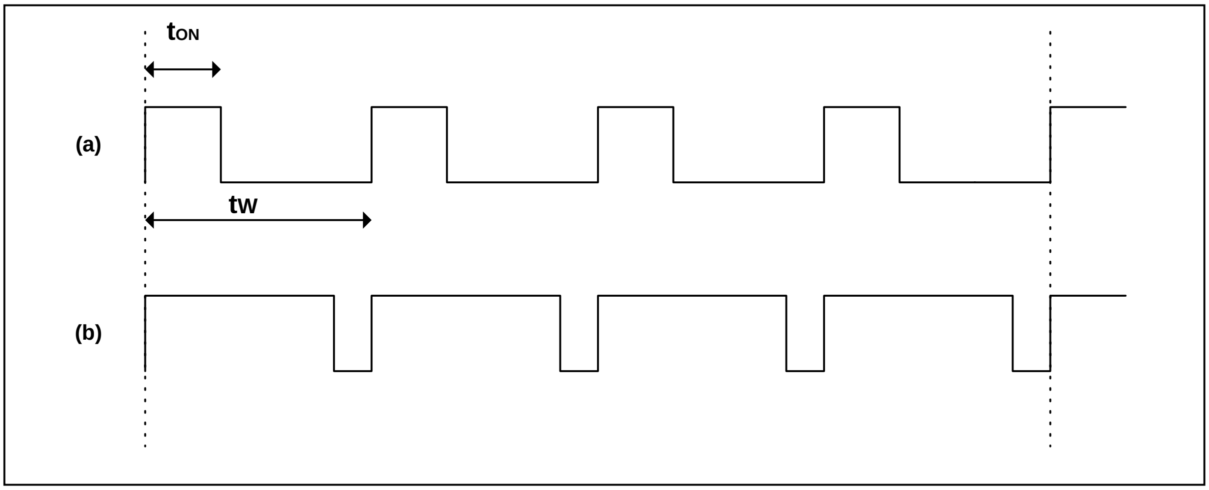

Figure 5 Duty Cycle

Duty Cycle = t_{ON}/t_W

Duty cycle = 33.33% [case (a)] and 83.33% [case (b)]

4. Theory of DAC

Depending on the application, the designer may want to fix the conversion period and vary the duty cycle or vice versa. For a more complex requirement, even both parameters may need to be changed to achieve the required output waveform.

The focus of this application is to generate a DC voltage with reasonable ripple after passing through a low-pass filter (a simple RC circuit). The PWM waveform will be charging through a capacitor when output at HI and discharging when output at LO.

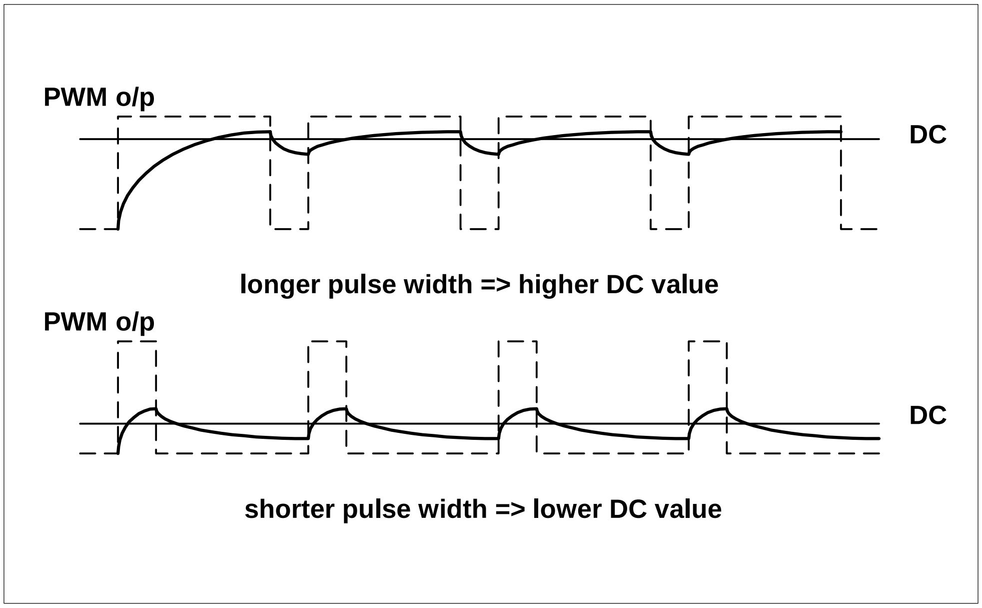

Figure 6 Conversion of PWM waveform to DC voltage

For simplicity, the conversion period (step 2) is fixed at 512/Ø and vary the duty cycle (step 3) to achieve the DAC function.

5. Circuit Diagram

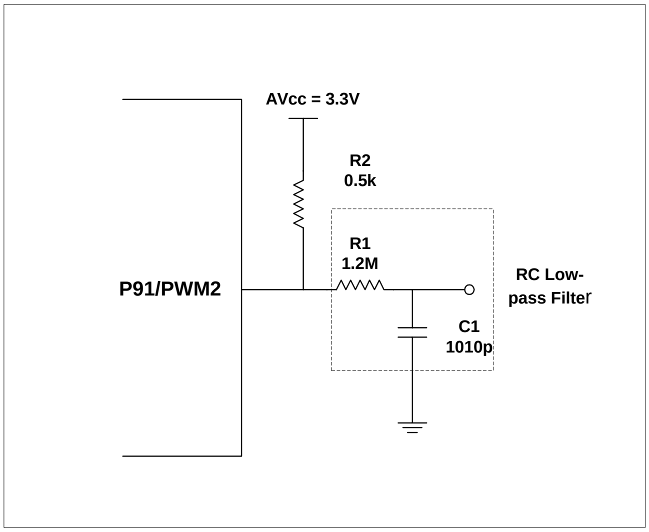

Figure 7 Low-pass RC circuit for DAC

To achieve DAC, output of PWM2 is connected to pull-up register and RC low-pass filter. The pull register R2 is used to drive more current to the load. Adjust register R1 to eliminate loading effect. Output waveform is measured and 3-dB cut-off frequency is recorded. The capacitance C1 can then be determined by the equation below:

$$f = \frac{1}{2\pi RC} \Rightarrow C = \frac{1}{2\pi fC}$$

In this example, R1 = 1.2 MΩ and C1 = 1010 pF,

$$f = \frac{1}{2\pi * 1.2M * 1010p} = 131.32Hz$$

The selection of R, C and conversion period is inter-related. If the conversion period is too short, or the capacitor value is too large, hence the analog value may not be achieved.

Since PWDR is a 10-bit counter (2^{10} = 1023), it allows 0 (min) equals to 0V (GND) and 1023 (max) equals to 5.0V (VCC). To achieve a specific voltage, V_x

$$V_x = V_{CC} * \frac{PWDRvx}{PWDR\max}$$

Note that the offset voltage is determined by the value of R1 and R2. Small resistance causes larger leakage current flow through R1 and C1. Large R1 may cause larger offset voltage. Without changing the cut-off frequency, it is better to change R2 value to control the amount of current and therefore the offset voltage.

Value of R1, R2 and C1 may change accordingly depends on circuit to be connected. For example, to drive a low impedance speaker, small R1 and R2 will be desirable for impedance matching purpose. Value of C1 will depend on the frequency bandwidth required and the noise consideration.

6. Further Measurement

Basic testing environment/conditions are:

1. ALE300L Emulator
2. $V_{CC} = AV_{CC} = 3.3V$
3. R1 = 1.2 MΩ, R2 = 0.5 kΩ and C1 = 1010 pF

   [5% tolerance for R1 & R2, 50V for C1]
4. No load
5. Voffset = 375 mV

Clock frequency, Ø = 5MHz

Clock Select: PWCRm1 = 1 and PWCRm0 = 1

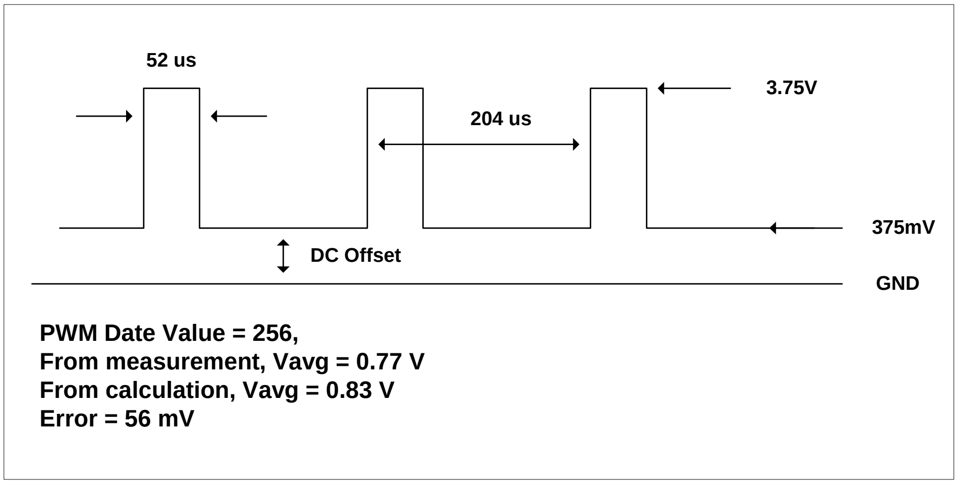

Figure 8 Measurement at PWM data value of H'100

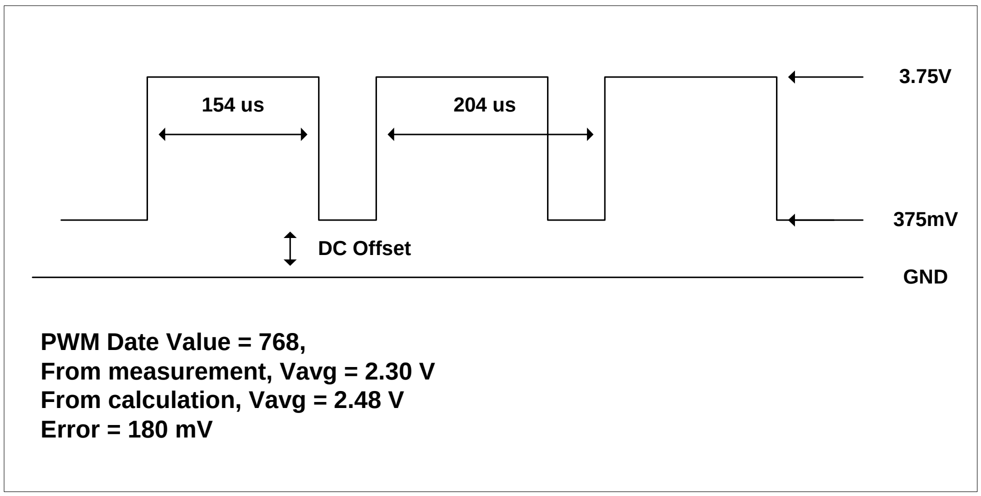

Figure 9 Measurement at PWM data value of H'300

From table 6.1, it shows that the maximum conversion error is about 5 to 10%.

PWM Value	Calculated Voltage(V)	Measured Voltage(V)	Error(mV)(%)
0	0 V	31 mV	-31
50	161 mV	169 mV	-8
100	323 mV	323 mV	0 (0%)
150	484 mV	460 mV	24 (5%)
200	645 mV	602 mV	43 (6.6%)
250	807	745 mV	62 (7.7%)
300	968	895 mV	73 (7.5%)
350	1.13	1.03	100 (8.8%)
400	1.29	1.18	110 (8.5%)
450	1.45	1.34	112 (7.7%)
500	1.61	1.47	143 (8.9%)
550	1.77	1.62	150 (8.5%)
600	1.94	1.75	185 (9.5%)
650	2.10	1.90	196 (9.3%)
700	2.26	2.04	220 (9.7%)
750	2.42	2.18	239 (9.9%)
800	2.58	2.33	250 (9.7%)
850	2.74	2.47	272 (9.9%)
900	2.90	2.62	283 (9.8%)
950	3.07	2.77	295 (9.6%)
1000	3.23	2.91	316 (9.8%)
1010	3.26	2.94	318 (9.8%)
1023	3.30	2.97	330 (10%)

Table 3 Comparison between the Measured Voltages and the Calculated Voltages

7. Program Overview

In this program, the main clock ($Ø_{osc}$) is set to 10 MHz by default, 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/8. At Ø = 5 MHz, the conversion period, 4096/Ø = 0.8192 ms.

PMR9 = H'F2

PWCR2 = H'FF

At the end of each conversion period, data point of equal spacing will be loaded into the PWDRL2 and PWDRU2 registers for 1024 cycles to form a up slope RAM, then subsequently another 1024 cycles for down slope RAM. This generates a triangular waveform continuously. To change the frequency of the triangular waveform, a simple way is to modify the delay loop preceding the writing to the PWM data register.

8. Program Flowchart

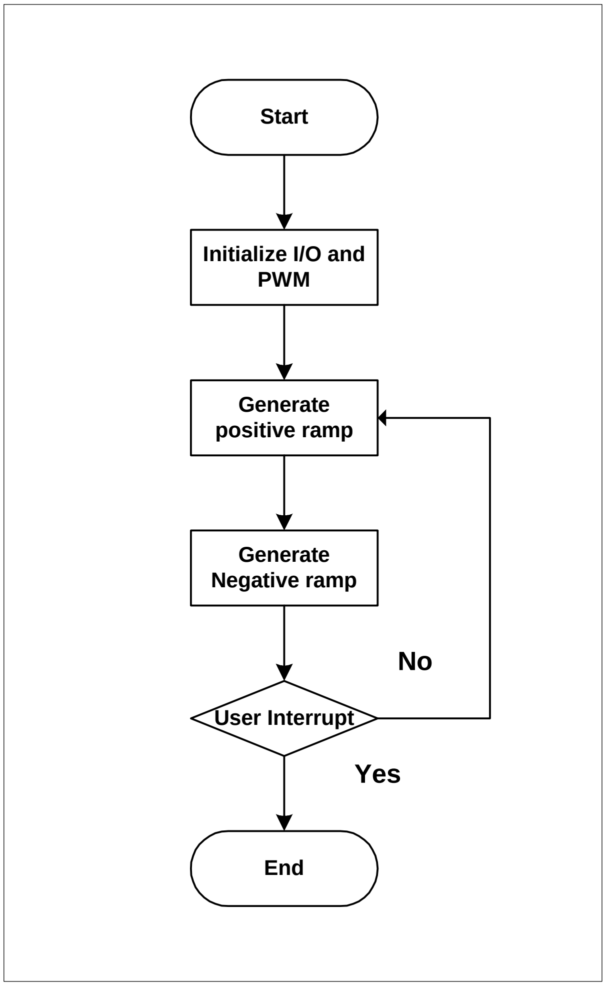

Figure 10 Flowchart of DAC

9. Software Listing

```
/************************************************************************/
/*  FILE              :DAC_1.c                                          */
/*  DATE              :Thu, Feb 06, 2003                                */
/*  DESCRIPTION :Usng PWM to generate triangular wave                   */
/*  CPU TYPE       :H8/38024F                                           */
/*                                                                      */
/*  This file is generated by Hitachi Project Generator (Ver.2.1).      */
/************************************************************************/
/*  External Clock : 10MHz                                              */
/*  Internal Clock : 5MHz                                               */
/*  Sub Clock      : 32.768kHz                                          */
/************************************************************************/

#include <machine.h>
#include "iodefine.h"

/************************************************************/
/*  Function define                                         */
/************************************************************/

void init_PWM(unsigned char);
void storeCount(unsigned short);

/************************************************************/
/*  RAM define                                              */
/************************************************************/

unsigned char PWDR_L2, PWDR_U2;
int i=0, k=0, j=0;
unsigned int f=100;
```

```
/*************************************************************************/
/*  FILE              :DAC_1.c                                           */
/*  DATE              :Thu, Feb 06, 2003                                 */
/*  DESCRIPTION :Usng PWM to generate triangular wave                    */
/*  CPU TYPE       :H8/38024F                                            */
/*                                                                       */
/*  This file is generated by Hitachi Project Generator (Ver.2.1).       */
/*************************************************************************/
/*  External Clock : 10MHz                                               */
/*  Internal Clock : 5MHz                                                */
/*  Sub Clock      : 32.768kHz                                           */
/*************************************************************************/

#include <machine.h>
#include "iodefine.h"

/************************************************************/
/*  Function define                                         */
/************************************************************/

void init_PWM(unsigned char);
void storeCount(unsigned short);

/************************************************************/
/*  RAM define                                              */
/************************************************************/

unsigned char PWDR_L2, PWDR_U2;
int i=0, k=0, j=0;
unsigned int f=100;
```

```
/*************************************************************/
/*  Main Program                                             */
/*************************************************************/

void main ( void )
{
        init_PWM(3);                            // select PWM2 to have (4096/5Mhz) conversion
period

   while (1)
   {       for(i=0;i<1024;i++)                  // increments
           {       for(k=0;k<f;k++);            // For delay, f control frequency of wave
                   storeCount(i);          // Write digital code into PWM registers
           }
           for(j=1023;j>=0;j--)                 // decrements
           {       for(k=0;k<f;k++);
                   storeCount(j);               // Write digital code into PWM registers
           }
   }
}

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
        }
}

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

}
```

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.