R8C/32C

REG05B0074-0100
Rev.1.00
Apr 21, 2010

Multiple PWM Generation using the R8C/32C DTC in repeat mode

Introduction

In some applications it is required to have a small package with more PWM outputs than available in hardware in the device. The purpose of this application note is to show how to use a Timer and the DTC module to generate additional PWM outputs.

This application has been tested using MB_R8C32 board but it is easily portable to any other R8C device having a DTC unit. This software is also configured to be adapted easily in case of using RSKR8C35 to obtain visual effect with PWM outputs on LED0 to LED3.

Similarly this example uses Timer RA and DTCD0 but any other Timer or DTC channel can be used.

Contents

1. Multiple PWM generation

1.1 The DTC

The Data Transfer Controller, or DTC, is a peripheral that transfers data from memory to memory without using the CPU. In this application note 8 separate (The PWM outputs have the same time base, so are not fully independent) PWM outputs are generated. However more could be added with minor modifications by using additional DTC channels or changing the size of data transfer.

The DTC is activated by a peripheral function interrupt to perform data transfers. The DTC and CPU use the same bus, with the DTC taking bus priority over the CPU. Before use, the DTC module needs a data transfer structure to be initialized. The structure will be read and updated each time the DTC is activated. This structure includes a transfer source address, a transfer destination address and operating modes which are allocated in the DTC control data area. Each time the DTC is activated, the DTC control data area is read and updated.

1.2 Application overview

In this application an internal DTC channel in repeat mode is used to modify the contents of the Port1 data register in order to generate the PWM outputs.

Timer RA is used as the trigger source for DTCD0 control data configuration. The frequency of update of the PWM outputs is defined by the Timer RA period. DTCD0 transfers data from a RAM table, which is updated by the MCU application software, to the P1 port register.

Such operation described in **Figure 1** could also be used to generate specific signals for applications such as High-brightness LED drive control, for example.

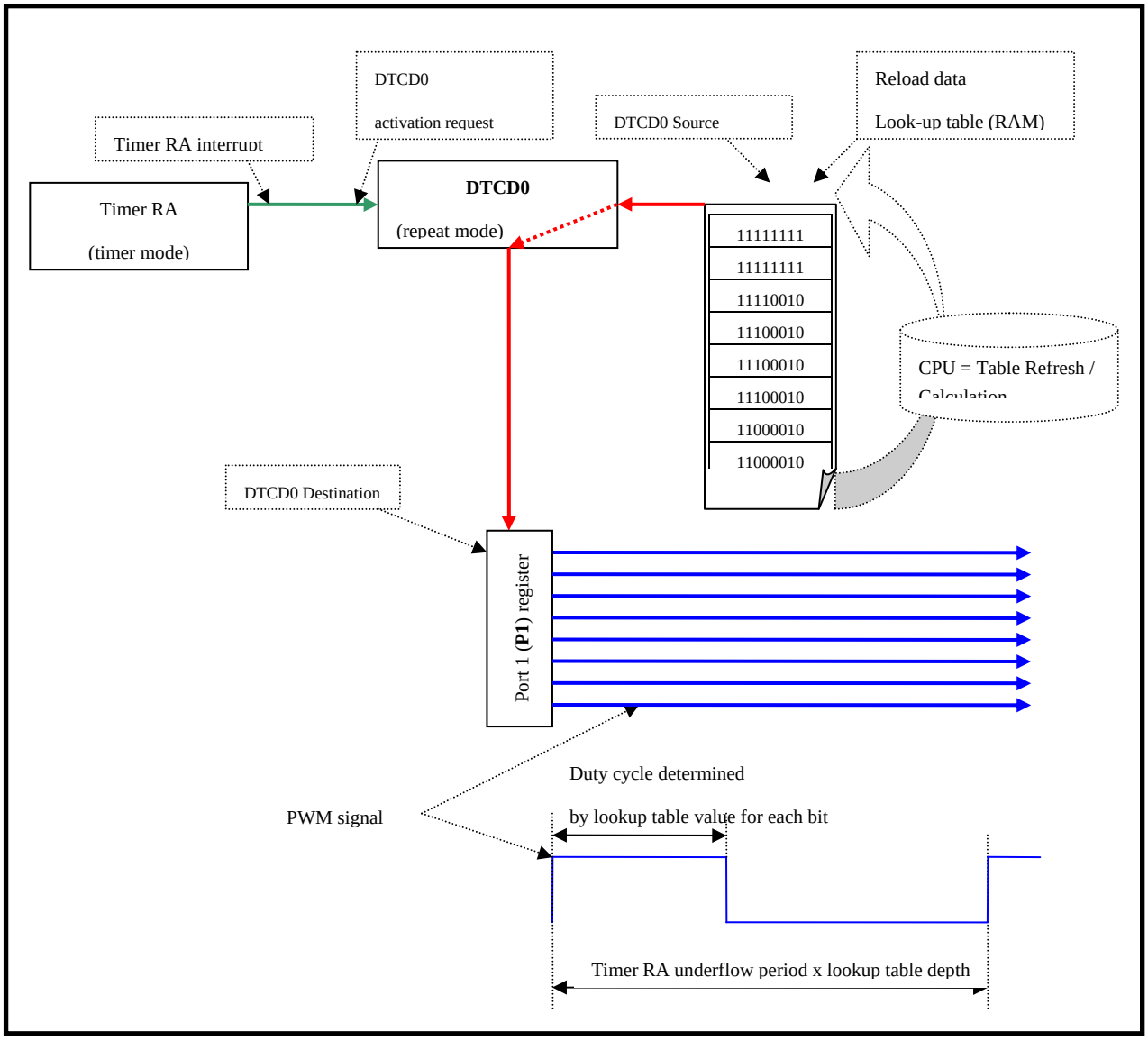

Figure 1 Operation Block Diagram

1.3 Principle

The DTC is configured in order to automatically transfer data from a “PWM look-up table” in memory to the I/O port. The DTC counter is automatically re-initialized at the end of table.

The required PWM resolution defines the depth of the “PWM look-up table”. Each PWM output is updated at each Timer RA interrupt (here after called “time unit”). In this application note, the arbitrary choice is made for 8-bit resolution and 8 PWM outputs to port P1, so 256 bytes are needed for the output RAM data.

The time unit is defined by Timer RA which triggers DTC at each underflow.

The frequency of the PWM outputs is therefore determined by the frequency of updates and the length of the table in RAM used to update the P1 port register.

PWM frequency = 1/ ((TimerRA period) x (PWM look-up table size in bytes))

1.4 PWM look up table organization

For easier understanding of the PWM Look-up table stored in RAM that the DTC accesses, lets assume 8 PWM outputs are implemented to port P1 as example and 16 values are used (4-bit resolution).**Table 1** shows a representation of this. The table contents are sent automatically every Timer RA underflow period to P1 data register by the DTC. The relationship between the table values and PWM output generated is as shown in **Figure 2**. The coloured highlighted values indicate where a change in the PWM state occurs.

An advantage for using the look-up table method is to permit different PWM frequencies for each output channel, using the same time base. Very flexible PWM waveforms can be created based on the PWM Look-up table data, extending the functionality from basic PWM to enhanced waveform generation.

The main function of the CPU is the initialization or modification of values in the table. Once this task is done, the DTC performs the transfer of data from the PWM Look-up table to the port independently.

Table 1 PWM Look-up table organization in RAM memory

Table index (address per each time base)	Look-up table data (in RAM)							
	b7	b6	b5	b4	b3	b2	b1	b0
0	1	0	X	X	X	X	1	X
1	1	0	X	X	X	X	1	X
2	1	0	X	X	X	X	1	X
3	1	**0**	X	X	X	X	1	X
4	1	**1**	X	X	X	X	1	X
5	1	1	X	X	X	X	1	X
6	**1**	1	X	X	X	X	1	X
7	**0**	**1**	X	X	X	X	1	X
8	0	**0**	X	X	X	X	1	X
9	0	0	X	X	X	X	1	X
10	0	0	X	X	X	X	1	X
11	0	**0**	X	X	X	X	1	X
12	0	**1**	X	X	X	X	**1**	X
13	0	1	X	X	X	X	**0**	X
14	0	1	X	X	X	X	0	X
15	0	1	X	X	X	X	0	X

Figure 2 shows how the port outputs respond to the data values in the PWM Look-up table after transfer by the DTC. For clarity colored vertical dotted lines are shown where the outputs change state corresponding to the same colored values in PWM loop-up **Table 1**.

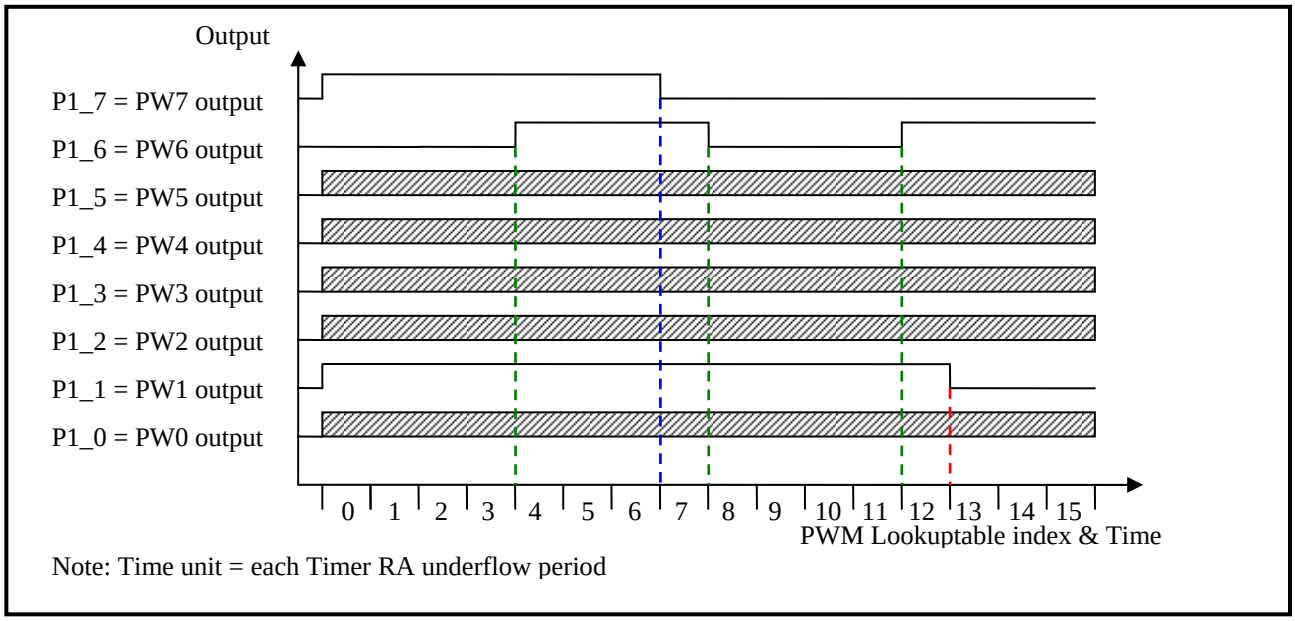

Figure 2 Port output states & PWM Look-up table relationship during DTC transfers

1.5 PWM look-up table calculation

Initialization of the PWM loop-up table is fairly straight forward. In the application software, for each PWM channel number, the look-up table has to be set to 1 from index 0 to N (which indicates high setting of each PWM), and then cleared from index N+1 to the maximum index of table. Therefore, let's take an example that the PWM output number 7 is to set with a new value: 156. Let's assume in this example that the PWM resolution is 8-bit, here the PWM look-up table requires 256 entries or indexes. In this case, application software initializes the look-up table as described below:

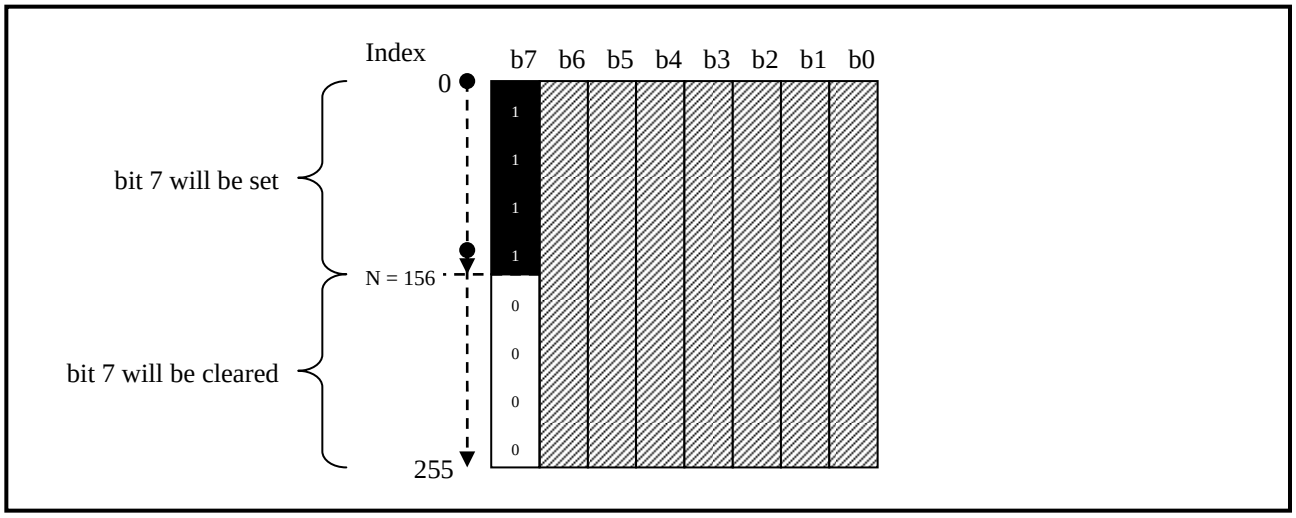

Figure 3 PWM Look-up table initialization for PWM generation

2. Peripheral architecture used for this application

2.1 Timer RA, operation timing in timer mode

The R8C32C has a number of timers integrated. Timer RA in Timer mode has been selected to trigger the DTC. In this mode Timer RA counts an internally generated count source. When the timer underflows, the contents of the reload register is copied into the count source register and the count is then continued, as shown in the **Figure 4**.

The divide ratio of the count source is determined by values set in TRAPRE and TRA registers:

Timer RA Underflow Frequency = 1/(TRAPRE+1)(TRA+1)

The underflow period of Timer RA is the base of DTC activation for PWM generation in this application note. In addition, Timer RA configuration has to be carried out with consideration to its underflow frequency as this has an impact on the internal bus load. In other terms more frequently transfers are done, less bus bandwidth is left for CPU activity.

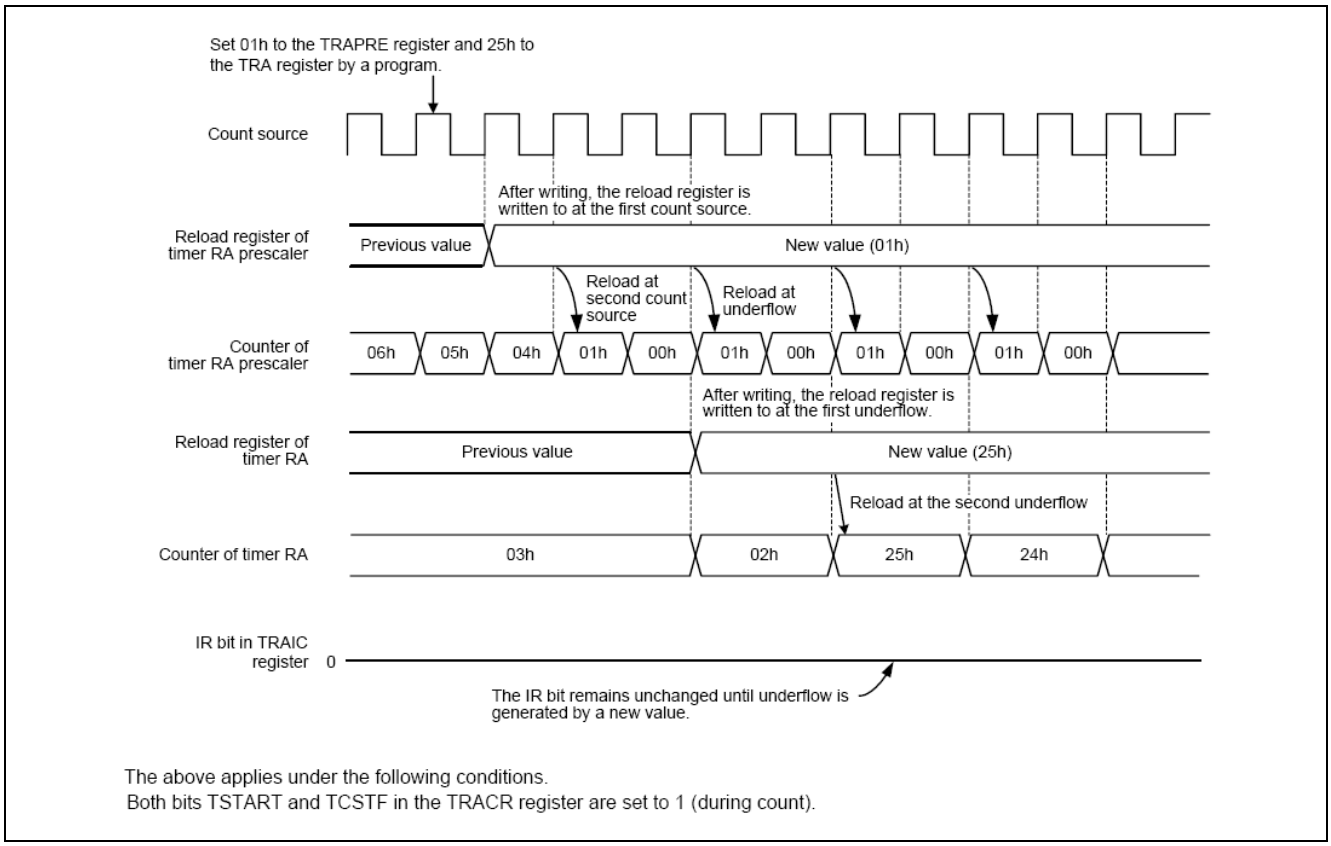

Figure 4 Operating Example of Timer RA when Counter Value is Rewritten during Count Operation

2.2 DTC, operation and configuration

2.2.1 Operation timing

At each activation source request i.e. Timer RA underflow, the DTC reads the appropriate vector address, which takes at least 2 cycles. The DTC reads the control data which defines the next transfer configuration, such as source address, destination address, etc… This requires at least four clock cycles. Next the transfer itself takes place, the required number of cycles differs depending on the required operation e.g. data read or write, unit of transfers (1-byte, 2-byte) and the memory area (internal RAM, ROM, SFR, etc). Lastly a write-back operation takes place, updating control data registers. The number of clock cycles required to write back control data also differs depending on the control data settings. The summary of DTC timing is described below:

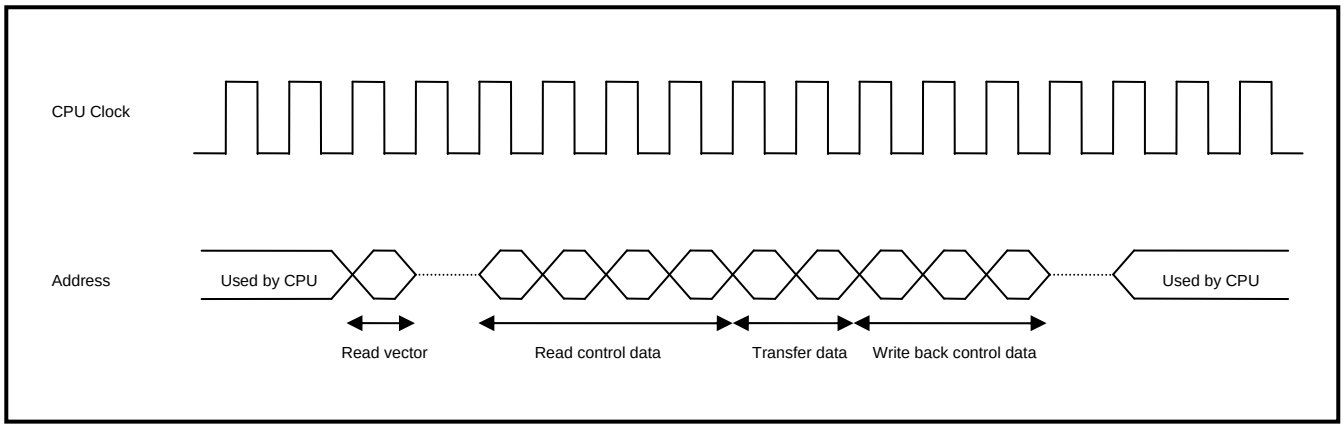

Figure 5 Example of DTC Operation Timing

2.2.2 Execution cycle count

The total number of cycles following DTC activation to completion is calculated by the required cycles for data read, data write and Control Write-Back as described in *Table 2*. This number impacts the DTC transfer capabilities, main software execution and determines the maximum time base frequency for PWM generation.

Table 2 Operation & required Number of Cycles following DTC Activation

Vector Read	Control Data Read Write (J)	Data Read	Data Write	Internal Operation
1	5 to 7	(Note 1)	(Note 1)	2
1	5 to 7	(Note 1)	(Note 1)	2

Note 1:
The number of clock cycles for data read/write is shown in Table 3 Number of Clock Cycles Required for Data Transfers.

In this application, transfers are made from a RAM look-up table and output port register, P1 as described earlier.

As a result, the DTC is configured as *Repeat Mode,* with the source address as *"repeat area*" and the *destination address fixed*.

Therefore, the *Data Read* clock cycles parameter for **Table 2** is determined by Internal RAM– 1-byte transfer), whilst the Data Write clock cycle parameter is SFR 1 byte access as shown in **Table 3** below:

Table 3 Number of Clock Cycles Required for Data Transfers

Operation	Unit of Transfers	Internal RAM (During DTC Transfers)		Internal ROM (Program ROM)	Internal ROM (Data Flash)	SFR (Word Access)		SFR (Byte Access)	SFR (DTC control data area)	
		Even Address	Odd Address			Even Address	Odd Address		Even Address	Odd Address
Data read	**1-byte SK1**	**1**		1	2	2		2	1	
	2-byte SK2	1	2	2	4	2	4	4	1	2
Data write	**1-byte SL1**	1		-	-	**2**		**2**	1	
	2-byte SL2	1	2	-	-	2	4	4	1	2

Finally the Control Write-Back operation, clock cycle count is determined by **Table 4 Specifications of Control Data Write-Back Operation** *(Source repeat Area, destination fixed).*

Table 4 Specifications of Control Data Write-Back Operation

Bits b3 to b0 in DTCCR Register	Operating Mode	Address Control		Control Data to be Written Back				Number of Clock Cycles
		Source	Destination	DTCCTj Register	DTRLDj Register	DTSARj Register	DTDARj Register	
00X0b	Normal mode	Fixed	Fixed	Written back	Written back	Not written back	Not written back	1
01X0b		Incremented	Fixed	Written back	Written back	Written back	Not written back	2
10X0b		Fixed	Incremented	Written back	Written back	Not written back	Written back	2
11X0b		Incremented	Incremented	Written back	Written back	Written back	Written back	3
0X11b	Repeat mode	**Repeat area**	**Fixed**	**Written back**	**Written back**	**Written back**	**Not written back**	**2**
1X11b			Incremented	Written back	Written back	Written back	Written back	3
X001b		Fixed	Repeat area	Written back	Written back	Not written back	Written back	2
X101b		Incremented		Written back	Written back	Written back	Written back	3

j = 0 to 23
X: 0 or 1

In summary for this application, with the size of the block to be transferred by one activation as one byte, the number of DTC execution cycles is as follows:

- Vector read: 1 cycle
- Internal Operation: 1 cycle
- Control Data Read: 4 cycles
- Data Read from RAM: 1 cycle for one byte transfer
- Data Write to SFR area: 2 cycles
- Control Write-Back: 2 cycles
- Internal Operation: 1 cycle

Therefore, a DTC operation requires 1+1+4+1+2+2+1 = **12 clock cycles for each byte transfer**.

In this application example the CPU clock is configured to 20MHz, meaning the DTC requires 12x50ns = **600ns as a minimum time between each DTC activation.**

Figure 6 Shows diagram of the Operation timing related to the previous calculated number of DTC execution cycles.

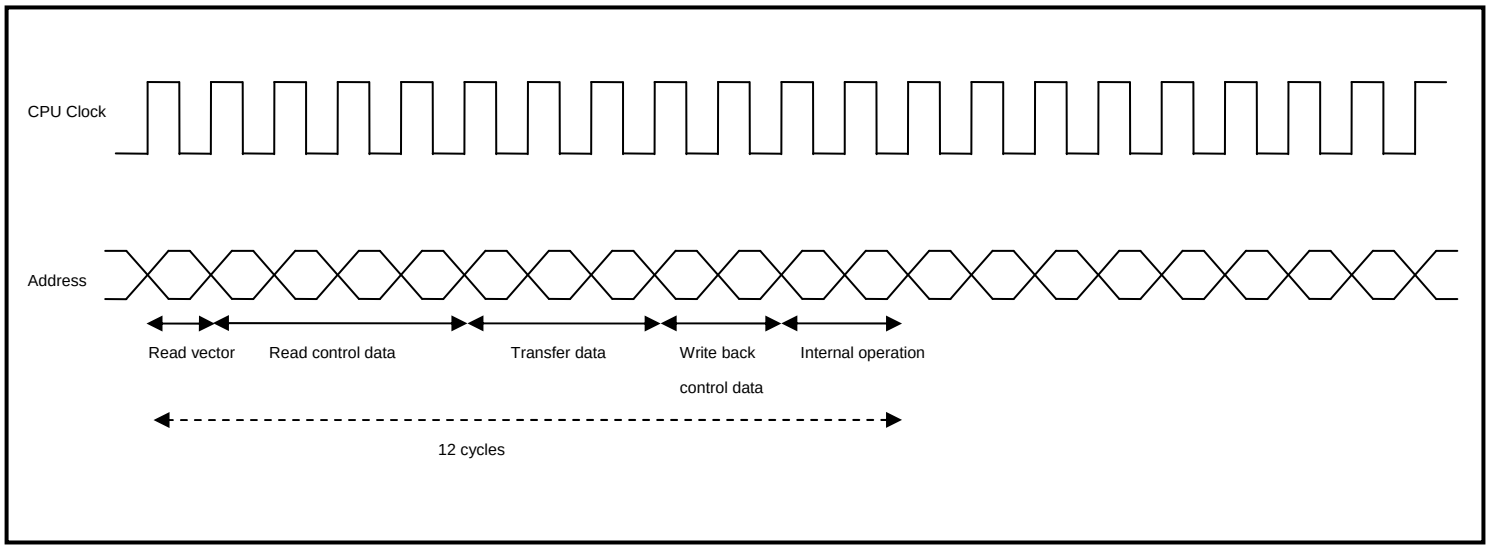

Figure 6 DTC Operation Timing for this application

2.3 Increasing PWM resolution

As explained previously, the number of transfer in this application software is 255.

To increase the PWM resolution beyond 8-bit, which is the maximum 255 repeat block transfer size of the DTC, the chain transfer capability of DTC could be used in order to activate another DTC channel.

Don’t forget that DTC operation and timings have to be taken in consideration as they impact the internal bus bandwidth and PWM generation capability

2.4 Using RSKR8C35

The software allow user to run software on RSKR8C35 board and obtain easily visual effect with LEDs.

The condition is to uncomment “#define RSKR8C35” in the main.h file and to rebuild the project.

It permits to use the timer RB for periodical PWM values updates to Port P3, as a result a changing lighting intensity on LEDs.

3. Internal bus resource impact

3.1 DTC operation trace

As described previously, the DTC requires 600ns minimum for each transfer operation in this application at 20MHz frequency. As the DTC and CPU use the same bus with the DTC having priority over the CPU there is an impact on the CPU instruction execution time.

This chart shows a trace taken by an emulator of instruction executed during DTC transfer at maximum occurrence frequency in blue and CPU usage in red:

Cycle	Label	Address	Data	Size	Type	R/W	DataAccess				Relative Time
-000038	002C44	0428		16b	0 DMA	W	0	-	0	1	1 -00:00:00.100.000.200
-000037	002C44	--00		8b	1 -	-	1	QC	0	1	1 -00:00:00.100.000.150
-000036	00C06F	--FE		8b	0 IB	R	0	-	1	1	1 -00:00:00.100.000.100
-000035	00C06F	FE--		16b	0 -	-	1	CB	0	1	1 -00:00:00.100.000.050
-000034	002C31	00--		16b	0 DMA	R	0	-	0	1	1 00:00:00.100.000.000
-000033	002C31	00--		16b	0 -	-	1	-	0	1	1 00:00:00.100.000.050
-000032	002C40	0103		16b	0 DMA	R	0	-	0	1	1 00:00:00.100.000.100
-000031	002C42	FFD7		16b	0 DMA	R	0	-	0	1	1 00:00:00.100.000.150
-000030	002C44	0428		16b	0 DMA	R	0	-	0	1	1 00:00:00.100.000.200
-000029	002C46	00E1		16b	0 DMA	R	0	-	0	1	1 00:00:00.100.000.250
-000028	000428	--FE		16b	1 DMA	R	0	-	0	1	1 00:00:00.100.000.300
-000027	0000E1	--FE		8b	0 DMA	W	1	-	0	1	1 00:00:00.100.000.350
-000026	0000E1	--FE		8b	0 DMA	W	0	-	0	1	1 00:00:00.100.000.400
-000025	002C42	FFD6		16b	0 DMA	W	0	-	0	1	1 00:00:00.100.000.450
-000024	002C44	0429		16b	0 DMA	W	0	-	0	1	1 00:00:00.100.000.500
-000023	002C44	--00		8b	1 -	-	1	-	0	1	1 00:00:00.100.000.550
-000022	00C070	--FF		8b	1 IW	R	0	-	1	1	1 00:00:00.100.000.600
-000021	00C071	--04		8b	0 IB	R	0	-	2	1	1 00:00:00.100.000.650
-000020	00C071	04--		16b	0 -	-	1	RB	1	1	1 00:00:00.100.000.700
-000019	002C31	00--		16b	0 DMA	R	0	QC	0	1	1 00:00:00.100.000.750
-000018	002C31	00--		16b	0 -	-	1	-	0	1	1 00:00:00.100.000.800
-000017	002C40	0103		16b	0 DMA	R	0	-	0	1	1 00:00:00.100.000.850

Figure 7 Executed instructions during DTC transfer at maximum occurrence frequency

Therefore it can be seen that the DTC activation period can't be fewer than 750 ns as it is the minimum time for DTC transfer plus CPU application software overhead.

3.2 Bus load analysis

Table 5 shows the relationship between Timer RA underflow frequency and bus bandwidth available for CPU code execution for an 8-bit resolution PWM resolution, i.e. 255 transfers of the DTC.

Table 5 Bus load impact with PWM frequency

Timer RA Underflow Period	Timer RA Underflow Frequency	PWM frequency	TRAPRE Register Value	TRA Register Value	Required Number of cycles for DTC	Time required for each DTC transfer (in µs)	BUS load for DTC transfers
50ns	20MHz	-	0	0	12	0.750µs	100%
0.750µs	1.333MHz	5.208kHz	not possible	Not possible	12	0.750µs	Around 100%
0.8µs	1.25MHz	4.88kHz	4	4	12	0.750µs	93.5%
1µs	1MHz	3.906kHz	5	2	12	0.750µs	75%
2µs	500kHz	1.95kHz	4	10	12	0.750µs	37.5%
5µs	200kHz	781.25Hz	10	10	12	0.750µs	15%
10µs	100kHz	390.63Hz	10	20	12	0.750µs	7.5%
15µs	66.67kHz	260.42Hz	10	30	12	0.750µs	5%
20µs	50kHz	195.31Hz	10	40	12	0.750µs	3.75%
25µs	40kHz	156.25Hz	5	100	12	0.750µs	3%
50µs	20Hz	78.13Hz	10	100	12	0.750µs	1.5%

4. Conclusion

Use of the DTC, an I/O port and a PWM look-up table allows for additional PWM channels to be implemented at no additional cost.

Eight PWM's with 8-bit resolution at 2kHz can easily be produced, using a single DTC channel, an 8-bit I/O port, 1 timer and less than 50% CPU bus bandwidth.

Care has to be taken, though, that the frequency of the PWMs allows sufficient bandwidth for the CPU to execute the application software without impact.

5. Appendix: Code listing

Main function:

```
/***********************************************************************
* Function Name : main
* Description   : Main function and set some SFR registers
* Argument      : none
* Return Value  : none
***********************************************************************/
void main(void)
{
    unsigned char i;

    /* Set initial values for the 8 PWM channels */
    pwm[0] = 32;
    pwm[1] = 64;
    pwm[2] = 96;
    pwm[3] = 128;
    pwm[4] = 160;
    pwm[5] = 192;
    pwm[6] = 224;
    pwm[7] = 250;

    /* Initialize MCU */
    mcu_init();

    /* Initialise Look-up Table for each of 8 channels */
    for (i = 0; i<8; i++)
    {
        UpdatePWM_DutyCycle(i,pwm_duty_cycle[i]);
    }

    /* Data Transfer Controller configuration */
    dtc_config();

    /* Timer RA configuration */
    timer_RA_config();

    /* Main loop */
    while(1);
}
```

MCU initialize function:

```
/***********************************************************************
Name        : mcu_init
Parameters  : None
Returns     : None
Description : MCU initialize
***********************************************************************/
void mcu_init(void)
{
    /* Initialize p1 to zero before definition */
    p1 = 0x00;
    /* P1 drive capacity  = HIGH (20 mA) */
    p1drr = 0xFF;
    /* Set port P1 to output */
    pd1 = 0xFF;
    /* Set High-speed on-chip oscillator clock to system clock */
    /* Protect off */
    prcr |= 0x01;
    /* High-speed on-chip oscillator on */
    fra0 |= 0x01;
    /* This setting is an example of waiting time for the */
    /* oscillation stabilization. Please evaluate the time */
    /* for oscillation stabilization by a user. */
    while( i <= 255 ) i++;
    /* Selects High-speed on-chip oscillator */
    fra0 |= 0x02;
    /* Protect on */
    prcr &= 0xFE;
}
```

DTC configuration function:

```
/***********************************************************************
* Function Name : dtc_config
* Description   : configure DTC module
* Argument      : none
* Return Value  : none
***********************************************************************/
void dtc_config( void )
{
    /* Setting Control Data No. of Timer RA interrupt to "No. 0" */
    dtc_vect_tra = 0x00;
    /* Setting DTC Control Register in DTCD0
    b7:b6        Set to 0
    b5    RPTINT Interrupt generation disabled
    b4    CHNE   Chain transfers disabled
    b3    DAMOD  Destination address is fixed
    b2    SAMOD  Invalid for the repeat area
    b1    RPTSEL Transfer source is the repeat area
    b0    MODE   Repeat mode */
    dtccr_dtcd0 = 0x03;
    /* Setting DTC Block Size Register in DTCD0: 1-byte is set */
    dtbls_dtcd0 = 1;
    /* Setting DTC Transfer Count Register in DTCD0: one-time is set to
transfer */
    dtcct_dtcd0 = 1;
    /* Setting DTC Transfer Count Reload Register in DTCD0 */
    dtrld_dtcd0 = 255;
    /* Setting DTC source address register in DTCD0 to PWM0_7 table */
    dtsar_dtcd0 = (unsigned long)&PWM0_7;
    /* Setting DTC destination address register in DTCD0 to "0x00E1" (P1
register address)*/
    dtdar_dtcd0 = (unsigned long)&p1;
    /* Non-maskable interrupts not generated is selected in DTC Activation
Control Register*/
    dtctl = 0x00;
    /* Setting DTC activation enable registers */
    dtcen0 = 0x00;
    dtcen1 = 0x00;
    dtcen2 = 0x00;
    dtcen3 = 0x00;
    dtcen4 = 0x00;
    dtcen5 = 0x00;
    /* Activation of Timer RA interruption enable */
    dtcen6 = 0x40;
}
```

Look-up table update calculation function:

```
/***********************************************************************
Name        : UpdatePWM_DutyCycle
Parameters  : unsigned char pwm_channel in range 0 to 7 corresponding to P10
to P17
              unsigned char duty_cycle in range 0 to 255 (0% is 0 -> 100% is
255)
Returns     : None
Description : Initialise or update PWM look-up table in RAM
***********************************************************************/
void UpdatePWM_DutyCycle (unsigned char pwm_channel, unsigned char duty_cycle)
{
    unsigned int pwm_table_index;
    unsigned char channel_mask;

    /* Create bit mask to allow setting or clearing correct bit in look-up
table for PWM channel being updated */
    channel_mask = (0x01<<pwm_channel);

    for (pwm_table_index = 0; pwm_table_index <= duty_cycle;
pwm_table_index++)
    {
        /* Set bits in look-up table up to desired PWM duty cycle */
        pwm_table[pwm_table_index] |= channel_mask;
    }

    for ( ; pwm_table_index <= MAX_TABLE-1; pwm_table_index++)
    {
        /* Clear bits in look-up table higher than desired PWM duty cycle */
        pwm_table[pwm_table_index] &= ~channel_mask;
    }
}
```

Timer RA configuration function:

```
/***********************************************************************
 Name        : timer_RA_config
 Parameters  : None
 Returns     : None
 Description : timer RA configuration, will be activation request for DTC
 module
 ***********************************************************************/
 void timer_RA_config(void)
 {
     /* Configure Timer RA Mode Register
     b7     TCKCUT            Provides count source
     b6:b4  TCK0,TCK1,TCK2    f1 selected
     b3                       Set to 0
     b2:b0  TMOD0,TMOD1,TMOD2 Timer mode */
     tramr = 0x00;

     /* Configure Timer RA I/O Control Register
     b7:b6:b5:b4:b3         Set to 0 in timer mode
     b2             TOENA   Set to 0 in timer mode
     b1             TOPCR   Set to 0 in timer mode
     b0             TEDGSEL Set to 0 in timer mode */
     traioc = 0x00;

     /* set up timer value for a 5 msec count */
     /* 50ns*5*100) -> generate 25µs */
     trapre = 5 - 1;
     tra = 100 - 1;
     /* start timer */
     tstart_tracr = 1;
 }
```

Website and Support

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

Inquiries
http://www.renesas.com/inquiry

Revision Record

		Description	
Rev.	**Date**	**Page**	**Summary**
1.00	Apr.21.2010	—	First edition issued

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this manual, refer to the relevant sections of the manual. If the descriptions under General Precautions in the Handling of MPU/MCU Products and in the body of the manual differ from each other, the description in the body of the manual takes precedence.

1. Handling of Unused Pins
   Handle unused pins in accord with the directions given under Handling of Unused Pins in the manual.
   — The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.
2. Processing at Power-on
   The state of the product is undefined at the moment when power is supplied.
   — The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.
   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.
   In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.
3. Prohibition of Access to Reserved Addresses
   Access to reserved addresses is prohibited.
   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.
4. Clock Signals
   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.
   — When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.
5. Differences between Products
   Before changing from one product to another, i.e. to one with a different type number, confirm that the change will not lead to problems.
   — The characteristics of MPU/MCU in the same group but having different type numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different type numbers, implement a system-evaluation test for each of the products.

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.
   - "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.
   - "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.
   - "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

RENESAS

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
7F, No. 363 Fu Shing North Road Taipei, Taiwan, R.O.C.
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
1 harbourFront Avenue, #06-10, keppel Bay Tower, Singapore 098632
Tel: +65-6213-0200, Fax: +65-6278-8001

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 1.0