AN11580

Implementing PWM dithering on the LPC18xx SCTimer/PWM

Rev. 1 — 6 August 2014 Application note

Document information

Info	Content
Keywords	Pulse Width Modulation (PWM), LPC1800, LPC18xx, LPC1857, ARM Cortex-M3, MCB1800, State Configurable Timer (SCT)/PWM, Dithering, Dither, SCTimer/PWM.
Abstract	This application note describes the PWM dithering process and how PWM dithering increases the average PWM resolution using the dither engine available in the State Configurable Timer/PWM peripheral of the LPC18xx microcontrollers.

Revision history

Rev	Date	Description
1	20140806	Initial version.

Contact information

For additional information, please visit: http://www.nxp.com

For sales office addresses, please send an email to: salesaddresses@nxp.com

1. Introduction

This application note describes NXP's unique State-Configurable Timer (SCTimer)/PWM peripheral and the SCTimer/PWM's dither engine. The application example is implemented on LPC1857, a Cortex-M3 based microcontroller with internal flash. However, this application note can also be used to implement PWM dithering on other NXP microcontrollers containing a similar State Configurable Timer (SCTimer)/PWM peripheral and dither engine. The dither engine is available on all flash-based LPC18xx and LPC43xx parts as well as on the LPC15xx parts. The example code developed on the LPC1857 in this application note can be ported to the LPC43xx without any changes.

1.1 Pulse-Width Modulation (PWM)

Pulse-width modulation (PWM) or pulse-duration modulation (PDM) can be used to encode information. In particular, PWM is often used to control the power supplied to an electrical device, especially to a device with an inertial load such as motor. One such example is the DC motor. The speed of a DC motor can be controlled by varying the average voltage across the motor armature winding. The average voltage is generated by applying a high-speed PWM signal across the armature of the motor. Other PWM examples are lamp dimmers or battery chargers. In all these applications, the power transferred to the load is controlled by changing the duty cycle of the PWM.

Duty cycle is the proportion of on-time to regular interval (or period) of the PWM signal. The duty cycle can only be varied in discrete steps. The smallest possible step duration is the PWM's resolution. To accomplish high resolution, either the PWM block and hence the counter must run at very high speed, or some other, special mechanism must be employed.

In some applications high-resolution in each PWM cycle is not a mandatory requirement. Instead, high **average** PWM resolution across a particular number of cycles is sufficient. If high resolution is not possible by simply decreasing the PWM step duration, PWM dithering can be used to achieve high average PWM resolution. Digital dithering varies the duty cycle of one or more PWM cycles by one counter clock over a few switching periods, so that the average duty cycle has a value between these two adjacent duty cycle levels.

Dithering is available on flash-based LPC18xx parts. Therefore, to increase average PWM resolution using dithering in an application, only LPC18xx microcontrollers with internal flash should be used.

1.2 System components and system setup

Fig 1 depicts the main system components required by this application example. The application was developed and tested on the Keil MCB1800 evaluation board (OM13039, 598). The Keil evaluation board contains an LPC1857 microcontroller.

Software included in this application note was developed and built with the Keil uVision4 Integrated Development Environment (IDE), LPCXpresso IDE version7, and the IAR Embedded Workbench IDE Version 7. This application example uses NXP's LPCOpen software platform version 2.12.

Connect the SCTimer/PWM peripheral outputs to the device pins using the Pin Multiplexer (PinMux) as shown in Fig 1.

This SCTimer/PWM peripheral supports

- 8 inputs
- 16 outputs
- 16 match/capture registers
- 16 events
- 32 states

Up to six SCT PWM outputs can be dithered, although the LPC18xx can generate more than six PWM outputs. To demonstrate the dithering feature, the code in this application example uses two PWM outputs “PWM1” (SCTOUT10) and “PWM2” (SCTOUT14) on port pins PD_12 and PD_11.

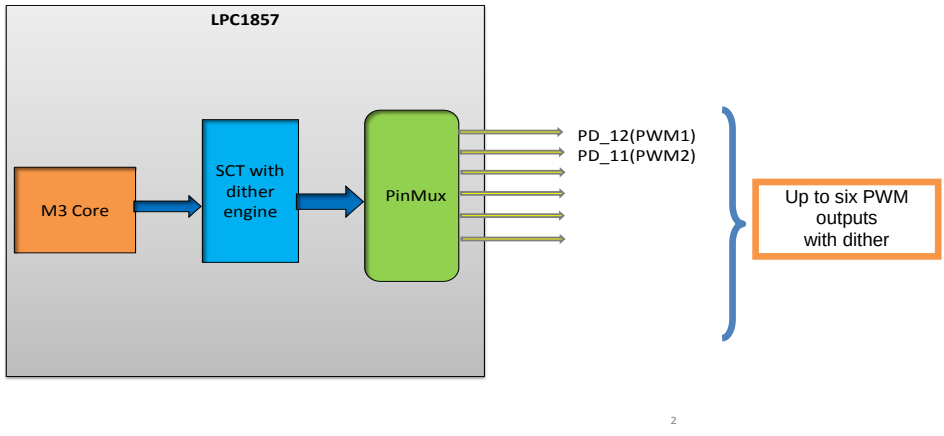

Fig 1. Main system components used in this application example

2. Theory of operation

In its simplest, single-state configuration, the SCTimer/PWM peripheral operates as an event controlled uni-directional or bi-directional counter. Events can be configured to be counter-match events, certain input or output levels, transitions on an input or output pin, or a combination of match and input/output behavior. In response to an event, the SCTimer/PWM output or outputs can transition, or the SCTimer/PWM can perform other actions such as generating an interrupt, starting, stopping or resetting the counter. Multiple simultaneous actions are allowed for each event. Furthermore, any number of events can trigger one specific action of the SCTimer/PWM. An action or multiple actions of the SCTimer/PWM uniquely define an event. A state is defined by which events are enabled to trigger the SCTimer/PWM’s action or actions in any stage of the counter. Events not selected for this state are ignored.

PWM output can be generated in the SCTimer/PWM single state configuration. To generate edge as well as center-aligned PWM output, the SCTimer/PWM block uses

match and match reload registers. Match reload registers are shadow registers which are used to safely update the PWM period or duty cycle while the application is running.

Fig 2 shows two plain PWM signals of 40 kHz frequency without dithering generated by the SCTimer/PWM block.

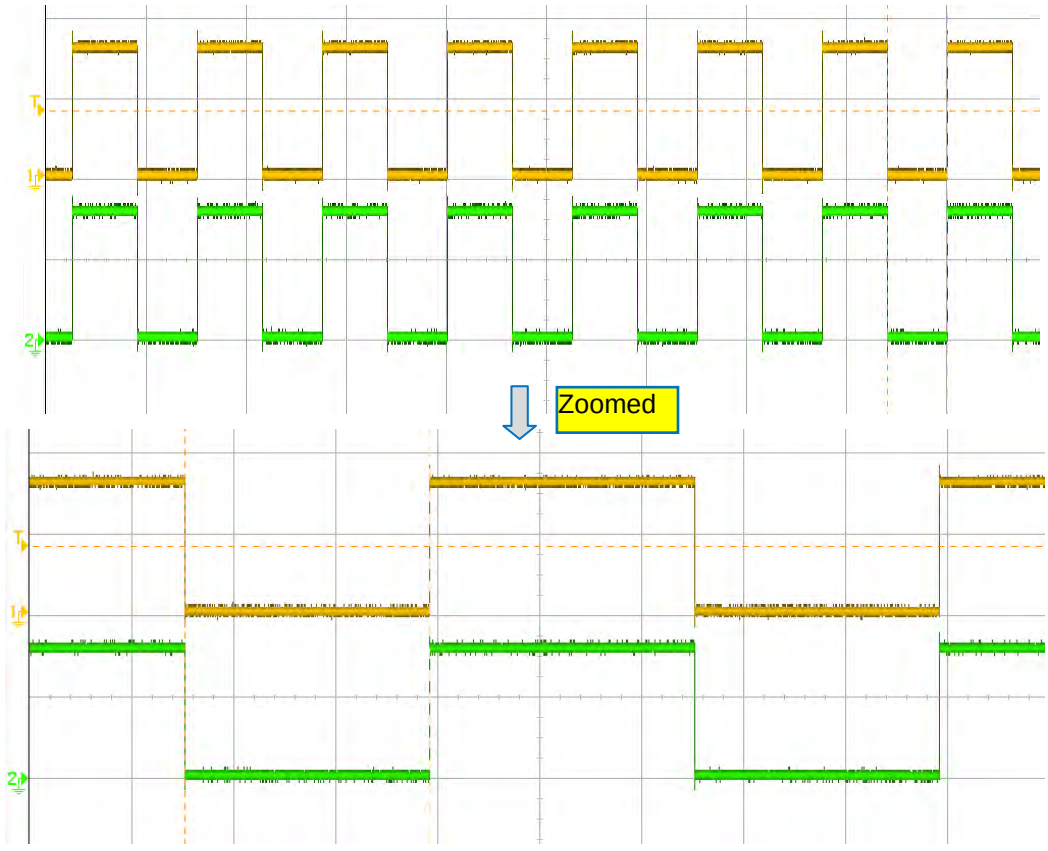

Fig 2. Scope screen capture of SCTimer/PWM module generated PWM (no dithering)

The duty cycle resolution of the PWM depends on the counter frequency that is used to generate PWM output. Therefore, the maximum PWM resolution is given by the counter and the integer match value stored in the match register. Instead of higher PWM resolution, which cannot be achieved with the given counter and match register combination, higher **average** PWM resolution might be sufficient in some applications and can be achieved with the SCTimer/PWM.

To achieve higher **average** PWM resolution without increasing the counter frequency, the LPC18xx SCTimer/PWM contains a dither engine. The dither engine contains six fractional match registers and two 16-bit or one 32-bit dither condition register. The first six match registers can be configured to have a fractional portion to their match values.

Higher average PWM resolution is achievable on the match registers with associated fractional match registers by using the dithering mechanism. The dither engine delays the assertion of a match by one counter clock on every n (0 to 15) out of 16 counter

cycles. The value of n is specified as a 4-bit value in the FRACMAT register associated with each of the first six match registers.

Fig 3 shows the dithering mechanism used in the LPC18xx SCTimer/PWM. In this figure, MATCH0 determines the PWM period and MATCH1 determines the PWM duty cycle (PWM width). One of the PWM duty cycles is delayed by one counter clock cycle.

The second part (lower figure) of the Fig 3 shows the mechanism in detail. This part of the figure shows that the pulse width (on-time) of the first PWM cycle is more than another two cycles. This one cycle delay is determined by the fractional match register.

Dithering can be disabled on any of the match registers by loading all zeroes (the default value) into corresponding FRACMAT registers.

The dither condition register controls the advancement of the dither pattern. When the dither condition register contains all zeroes (the default value), the dither engine advances to the next count in the dither pattern every time the SCTimer/PWM counter reaches zero (i.e. at the start of every new SCTimer/PWM counter cycle).

It is possible, using this register, to alter that behavior by qualifying the advancement through the dither pattern with designated events. There is one global dither condition register per 16-bit SCTimer/PWM block. This register controls advancement through the dither patterns for all the match registers associated with that half of the SCTimer/PWM.

Setting one or more bits in this register to one will cause the dither engine to advance to the next element in the dither pattern (i.e. increment the 16-state cycle counter) only following SCTimer/PWM counter cycles during which one or more of the designated dither events have occurred. This application example does not use this feature.

For more information about dither advancement feature, see the LPC18xx User Manual UM10430 (section 6).

Generating a simple PWM waveform requires the use of two SCTimer/PWM match registers as shown in Fig 3. The first, programmed with the value “MATCH0”, sets the PWM output and clears the counter to end one counter cycle and start the next one. Therefore the MATCH0 value decides the period of the PWM signal. The second match register, programmed with the value “MATCH1”, clears the PWM output.

The period of the PWM waveform is:

[MATCH0 + 1] x Tc

where Tc is the period of the SCT counter clock (=1/Fc where Fc is the frequency of the SCT counter clock).

The Pulse-Width of the PWM waveform WITHOUT DITHERING is:

[MATCH1 + 1] x Tc

Note that the pulse-width resolution without dithering is Tc.

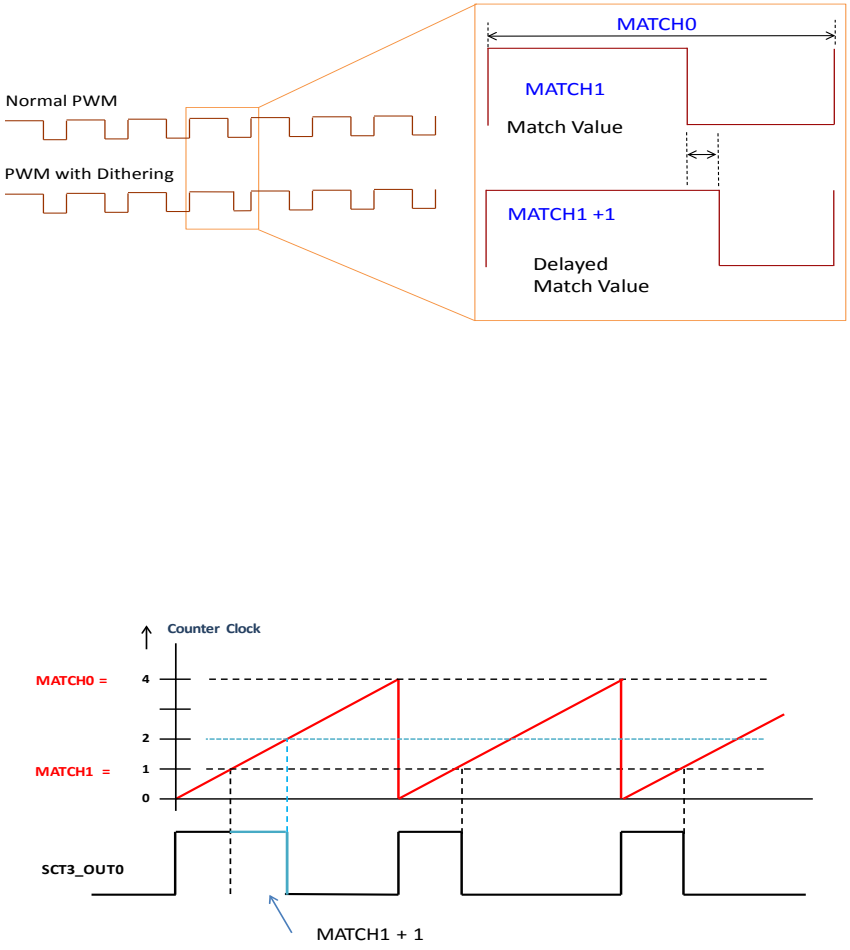

Fig 3. Generating PWM dithering

To achieve average higher resolution, the Fractional Match register (FRACMAT) associated with the Match1 register can be programmed with a value N where N is between 0-15 including these two numbers. This fractional match value N specifies the number of times during each set of sixteen counter cycles that the MATCH1 value (and the clearing of the output pulse) will be delayed by one counter clock.

Fig 4 shows the dither pattern corresponding to the value programmed in fraction match register (FRACMAT). When the programmed value is 1, the PWM width of 8th PWM cycle will be delayed by one counter clock cycle. When the programmed value is 2, PWM width of 4th and 12th PWM cycle will be delayed by one counter clock cycle and so on. See Fig 4.

	COUNTER CYCLE														
FRACMAT	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0000																
0001									D							
0010					D								D			
0011					D				D				D			
0100			D				D				D				D	
0101			D				D		D		D				D	
0110			D		D		D				D		D		D	
0111			D		D		D		D		D		D		D	
1000		D		D		D		D		D		D		D		D
1001		D		D		D		D	D	D		D		D		D
1010		D		D	D	D		D		D		D	D	D		D
1011		D		D	D	D		D	D	D		D	D	D		D
1100		D	D	D		D	D	D		D	D	D		D	D	D
1101		D	D	D		D	D	D	D	D	D	D		D	D	D
1110		D	D	D	D	D	D	D		D	D	D	D	D	D	D
1111		D	D	D	D	D	D	D	D	D	D	D	D	D	D	D

Fig 4. Dither pattern in SCT

Therefore, with dithering, the AVERAGE Pulse-Width of the PWM waveform becomes:

[MATCH1 +1 + N/16] x Tc

where N is the value in the fractional match register and Tc is the counter resolution.

If we change the value of N from 1 to 15, we will get an average PWM width over 16 counter cycles in the steps of Tc/16. Note that now the effective pulse-width resolution becomes Tc/16. This is how dithering improves the average PWM resolution over 16 counter cycles.

Fig 5 shows the dither pattern (0xF) that is the value programmed in the factional match register (FRACMAT) and the corresponding fractional match reload register. In this case, N=15 and the average PWM width over 16 cycles is

(MATCH1+1+15/16) x Tc.

As shown in Fig 5 dithering is applied in all 1 to 15 counter clock cycles.

The PD_11 (scope channel1, yellow color waveform) shows the normal 40 kHz waveform and PD_12 (scope channel2, green color waveform) shows the 40 kHz waveform with dithering.

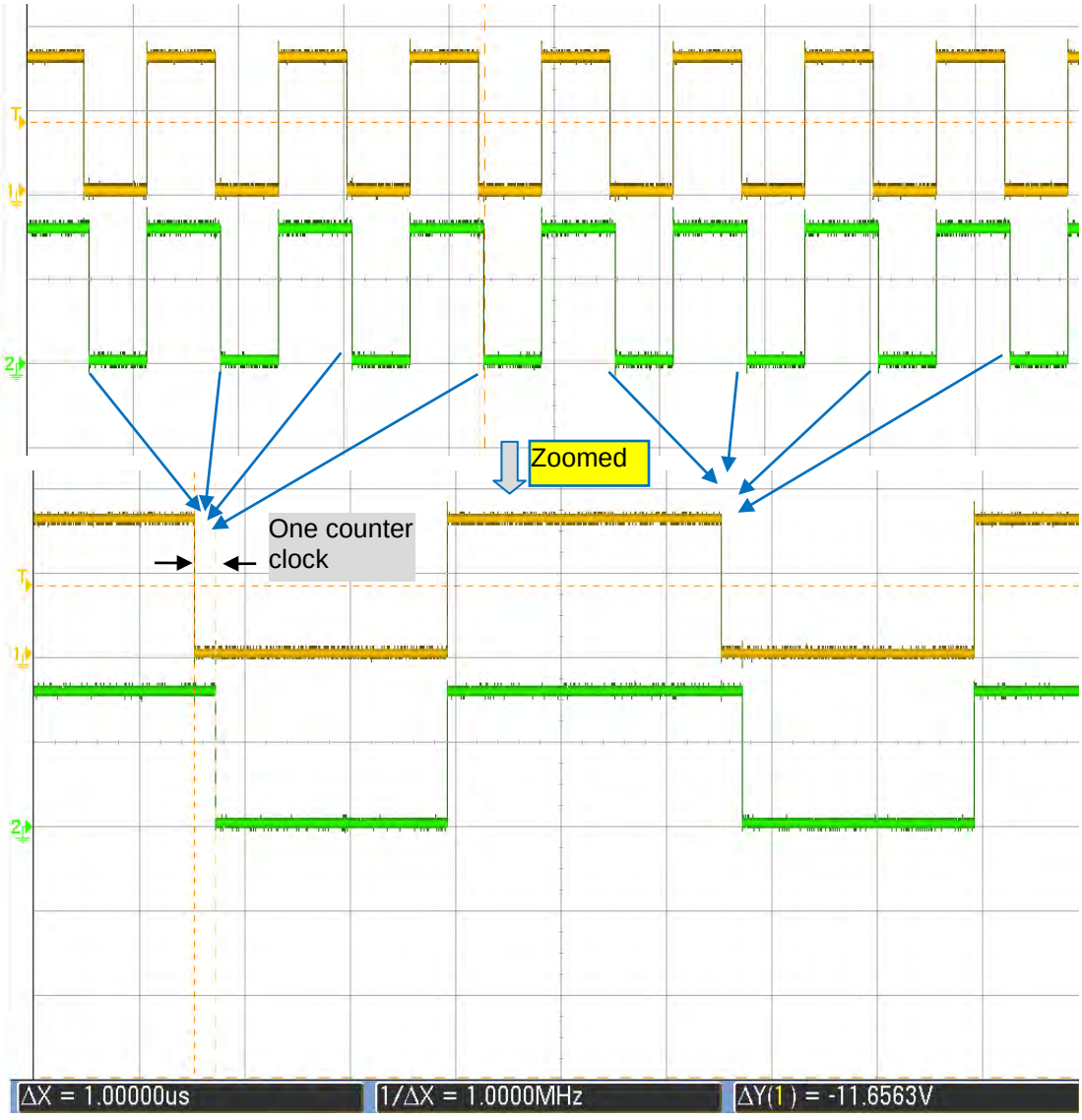

Fig 5. Waveform with the dither pattern value of 0xF (PD_11 normal PD_12 dithered)

Fig 6 shows the PWM dithering with dither pattern 0x8. In this case N=8 and the average PWM width over 16 cycles is

(MATCH1+1+8/16) x Tc.

As shown in Fig 6, dithering is applied in alternating PWM cycles.

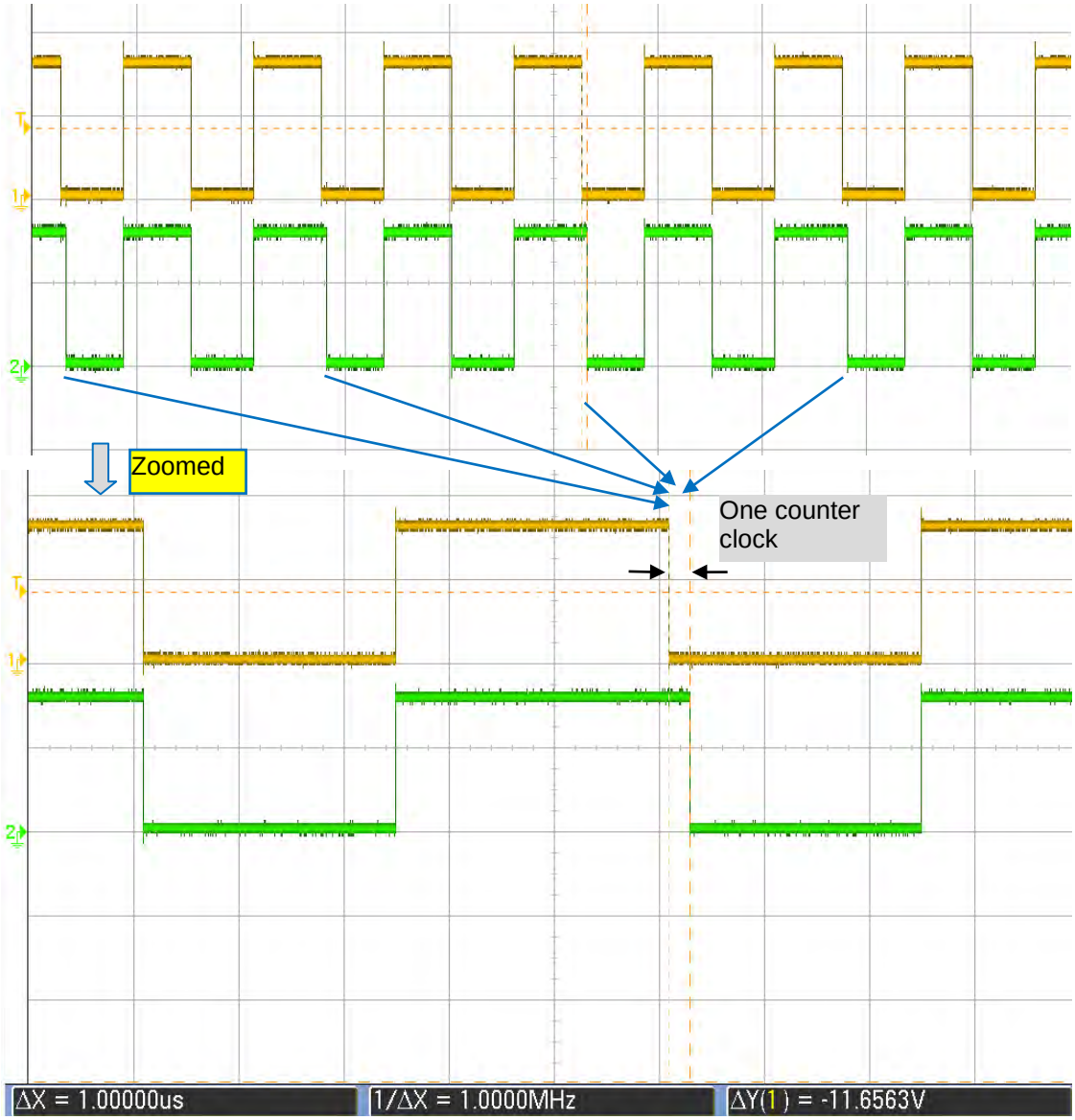

Fig 6. Waveform with the dither pattern value of 0x8 (PD_11 normal PD_12 dithered)

3. Application setup and firmware overview

NXP's LPCopen platform is used to develop the firmware on Keil uVision4 IDE, on LPCxpresso IDE, as well as on the IAR Embedded Workbench.

In this example, the peripheral clock of LPC1857 is set to 180 MHz, which is the input clock to SCTimer/PWM. This application example only uses two SCT outputs COUT_10 and COUT_14. CTOUT_10 and CTOUT_14 functions are connected to pins PD_12 and PD_11 respectively. The SCTimer/PWM block's input clock divider is set to 179 (180-1) in the SCTimer/PWM control register, which provides a divided clock to the

SCTimer/PWM counter. The SCTimer/PWM counter clock is 180 MHz/(179+1), which is equal to 1 MHz. Therefore in this example, the SCTimer/PWM counter is clocked at 1 MHz.

The PWM resolution is 1/1 MHz = 1.00 us.

When dithering is used the average resolution is Tc/16 which is (1.00/16) uS.

The demo setup with MCB1800 is shown in the Fig 7. The MCB1800 board may be powered by either connecting any of the Micro USB or by connecting a power adaptor. In Fig 7, the board is powered using the micro USB connector P4.

Scope probes are connected on PD_11 and PD_12. The PWM signal on port PD_12 uses dithering. The output is shown in Fig 6.

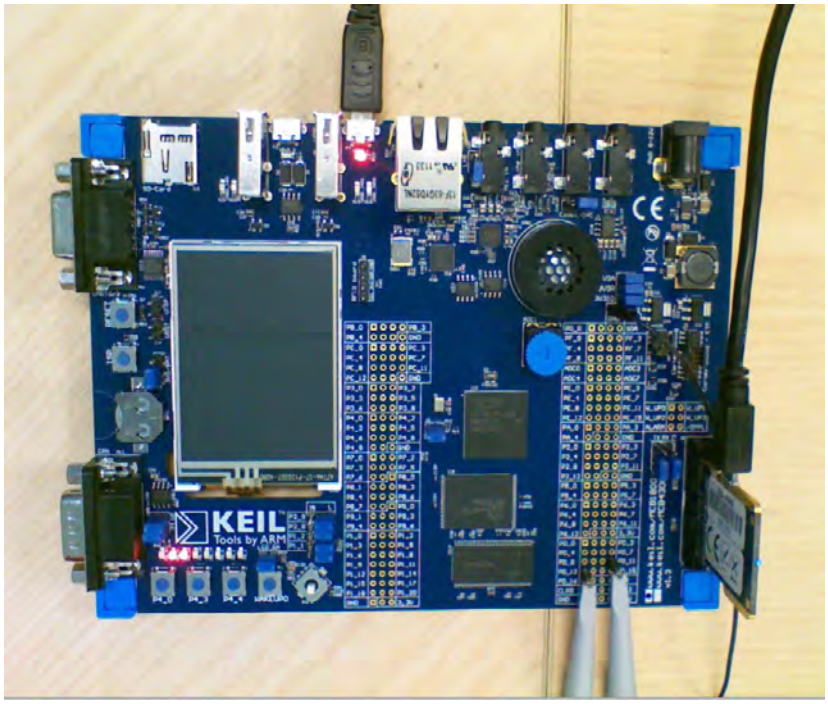

Fig 7. Demo setup

The frequency of these two PWM outputs, PWM1and PWM2 can be calculated as shown below. The peripheral clock is set to 180 MHz. The prescaler value (PRE_L) is set to 179, which clocks SCTimer/PWM block at 1 MHz as calculated before. The PWM duty cycle is set to 50%.

The PWM frequency generated in this application example is

Fpwm = 1 MHz/(24+1) = 40 kHz

In this application, the user must specify the required frequency value. An LPCOpen platform function is available to calculate the value by which match register will be loaded.

The user can change duty cycle and frequency by just changing the values of the constants.

Table 1 shows PWM signals PWM1, PWM2, SCTimer/PWM outputs CTOUT and the corresponding port pins PD_12 and PD_11.

Table 1. PWM output

PWM	SCT_OUT	PORT
PWM1	CTOUT_10	PD_12
PWM2	CTOUT_14	PD_11

4. Expected output

Fig 6 shows the expected output from two PWM outputs. The software is set to generate two PWM outputs at frequency of 40 kHz. Port PD_2 is set to generate dithering in alternative PWM period because a dither pattern 0x8 is programmed to fractional match and fractional match reload registers. Port PD_11 will just generate a 40 kHz PWM output.

5. Conclusion

NXPs unique peripheral SCTimer/PWM with the integrated dither engine can be used to generate higher average PWM resolution without consuming more core processing power. To achieve this high average PWM resolution, the SCTimer/PWM block is not required to run at very high frequency thereby saving power. The integrated dither engine can be used in applications where higher average PWM resolution is needed.

6. References

LPC18xx user manual UM10430

7. Legal information

7.1 Definitions

Draft — The document is a draft version only. The content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included herein and shall have no liability for the consequences of use of such information.

7.2 Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors accepts no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from national authorities.

7.3 Trademarks

Notice: All referenced brands, product names, service names and trademarks are property of their respective owners.

8. Contents

1. Introduction 3
1.1 Pulse-Width Modulation (PWM) 3
1.2 System components and system setup 3
2. Theory of operation 4
3. Application setup and firmware overview 10
4. Expected output 12
5. Conclusion 12
6. References 12
7. Legal information 13
7.1 Definitions 13
7.2 Disclaimers 13
7.3 Trademarks 13
8. Contents 14

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 6 August 2014
Document identifier: AN11580