

PPG Function of F2MC-16LX MCU Series as PWM

Introduction	1
PPG Function	1
Important Register Configuration	2
Flow chart	3
Program Code	4

PPG Function of F2MC-16LX MCU Series as PWM

Introduction

PPG of few MCU families of F2MC16LX series like MB90420 PPG can be configured to function as PWM. This application note briefly describes functionality and configuration of PPG peripheral as PWM. Example code is described for MB90425 series.

MB90420 series has 16bit x3ch PPG. Special operational mode called PWM mode makes PPG peripheral of this particular series different than other series.

PPG Function

Example block diagram of PPG is shown Figure-1.

PPG peripheral block consists of a prescaler, 16 bit down counter and 16 bit data register for interval setting, 16 bit compare register for duty setting and also a pin control section.

Programmable pulse generator is used in applications where variable frequency pulse and duty ratios are required. Value stored in the reload register PRL (for low pulse width) and PRLH (for High pulse width) are down counted and reloaded. The pin output value is inverted upon a reload caused by counter borrow. This operation results in pulses of specified L pulse width and H pulse width.

PPG peripheral derives the clock using prescaler of 1/1, 1/4, 1/16, 1/32. PPG solution can be achieved by different ways using

different types of operation modes depending on customer's applications. They are

- PWM mode
- One shot mode

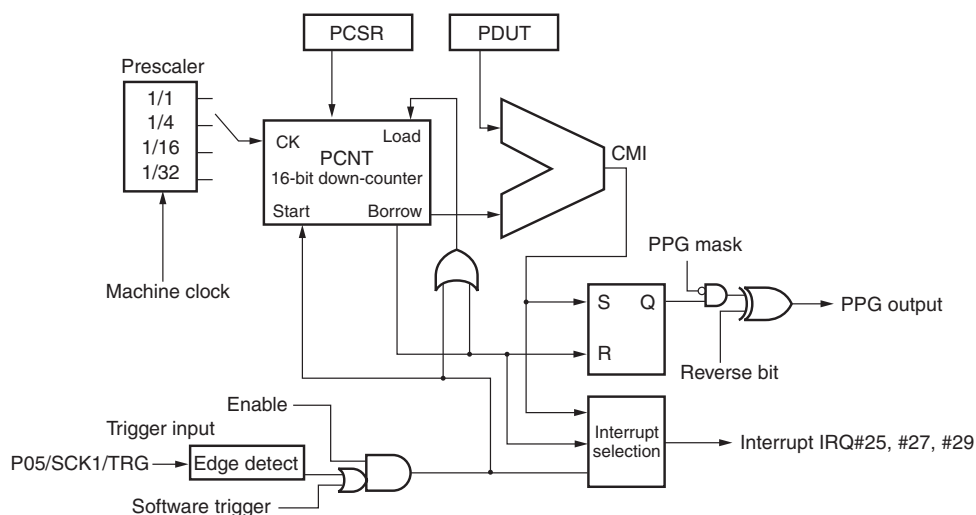
PWM operation enables continuous pulse outputs and one-shot mode outputs a single pulse based on edge detection of the trigger input.

PPG outputs a pulse in synchronization with an external or software trigger. The interval and duty cycle of pulse output can be freely selected by changing the value of two 16-bit registers. This feature helps PPG to operate in 'PWM mode'. This PWM function mode allows outputting pulses in synchronization with a trigger as specified by the program.

In this PWM operation mode pulses are continuously output after detection of start trigger. A register called PCSR controls the intervals between the output pulses and a register called PDUT controls duty ratio. Configuration of registers will be explained in the next sections.

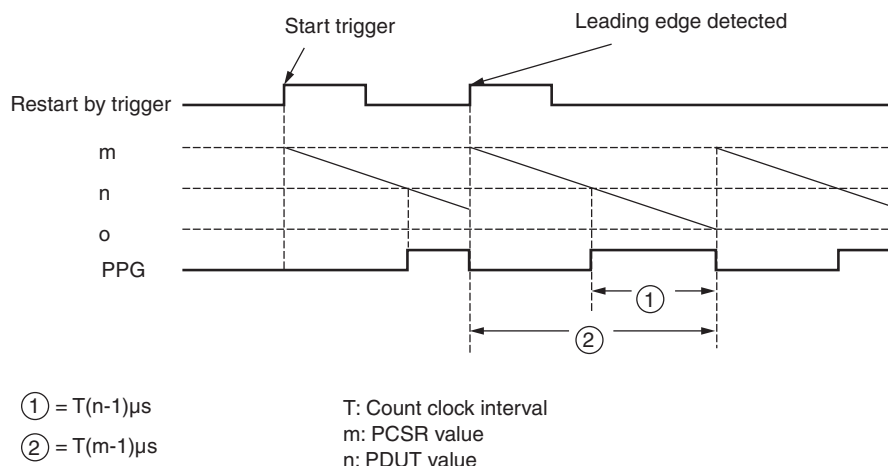
Following waveforms give a better understanding of this PWM mode and effect of PCSR and PDUT registers.

Figure 1.



PPG Function of F2MC-16LX MCU Series as PWM

Figure 2. Timing of PWM when edge is detected and restart is enabled



Important Register Configuration

PCNT is a PPG control and status register for upper and lower bits from bit 0 to bit 15. PCNTH indicates 8 upper register bits

PCNTH

CNTE	STGR	MDSE	RTRG	CKS1	CKS0	PGMS	
------	------	------	------	------	------	------	--

CNTE bit starts or stops 16 bit down counter operation. It enables the operation of clock among the four types of available internal clocks as shown in the above figure-2. The counter value by reset or counter borrow may be initialized to FFFF.

Setting **STGR** bit enables software trigger required for this operation. One can understand the significance of this bit by observing Figure-2 PWM timing waveforms.

MDSE bit is for selecting mode one-shot mode or PWM mode. If '0' functions in PWM mode.

Based on the software trigger, **RTRG** bit is used to restart the operation.

2 bits **CKS0** and **CKS1** are used to select clock of the 16-bit counter based on the input machine clock frequency. Hardware manual gives complete details.

PGMS is PPG output mask selection bit that can be used depending on the H or L polarity pulse output.

PCNTL

EGS1	EGS0	IREN	IRQF	IRS1	IRS0	POEN	OSEL
------	------	------	------	------	------	------	------

PCNTL indicates lower 8-bit register.

EGS0 and **EGS1** are 2 bits for selecting leading edge, trailing edge or both. This is very clearly indicated in the above **Figure 2**.

Setting **IREN** bit enables PPG interrupt and **IREF** is status flag that indicates the interrupt status.

IRS0 and **IRS1** are for setting type of interrupt source cause such as software trigger, counter borrow or duty cycle matching.

POEN bit is for selecting/enabling dual port pin function either as port pin or PPG output pin.

PGMS and **OSEL** bits are for the selection of normal/L/H polarity PPG output.

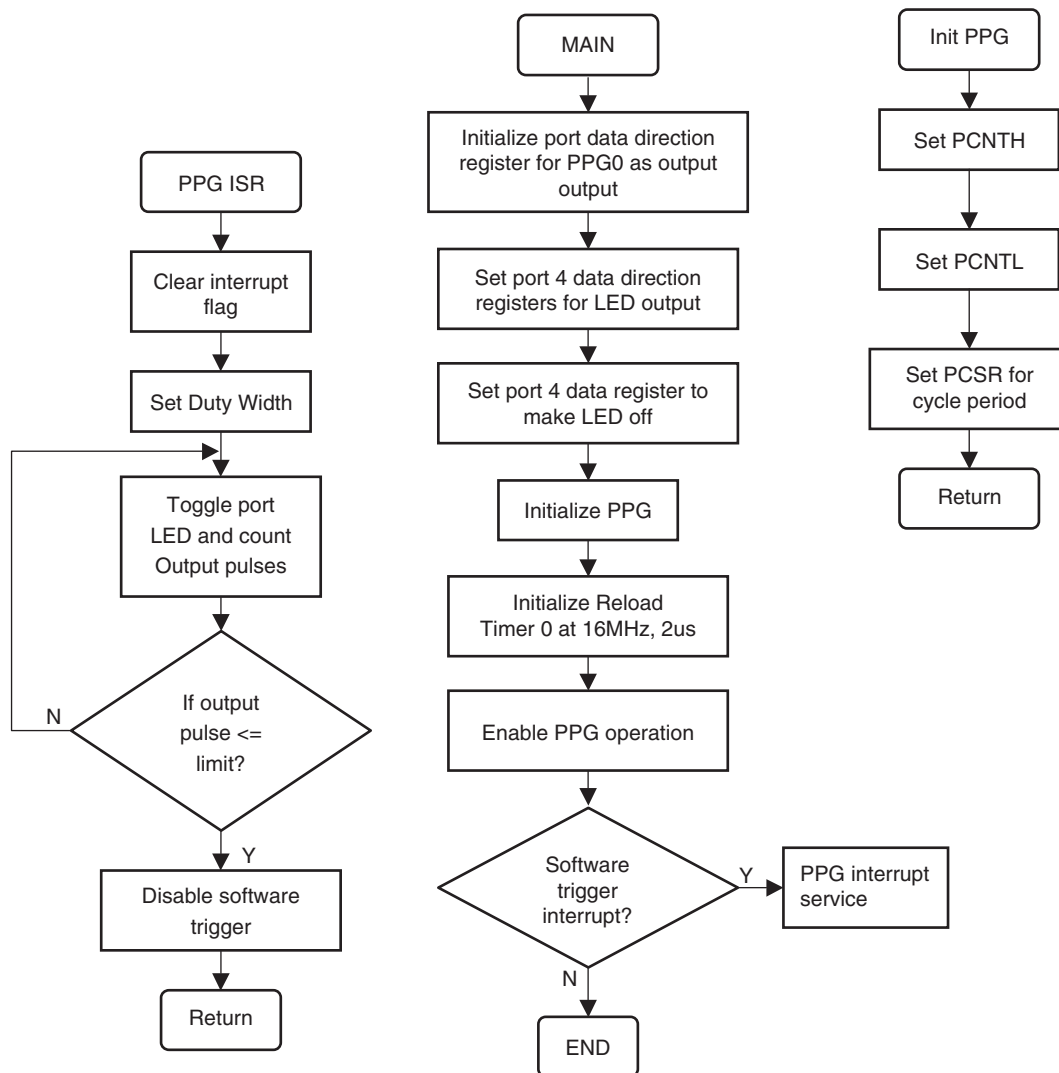
PDCR register is PPG down counter register. This is a 16 bit register that includes 8 upper bit register PDCRH and 8 lower bit register PDCRL. Counter value can be specified using this word accessible register.

PSCR register a PPG interval setting register of 16 bits that includes RCSRH and PCSRL, each having 8 bits. Duty cycle register should be set before setting this bit.

PDUT register is for setting duty cycle. Always note that $PSCR < PDUT$. This register of 16 bit that includes PDUTH and PDUTL is bit configurable.

For more details hardware manual can be referred. But the flow chart and below source code gives better idea in configuring these register to use PPG.

Flow chart



PPG Function of F2MC-16LX MCU Series as PWM

Program Code

In the below example code, PWM Waveform is output on port pin PPG0 and the same can be input to port pin P40. LED can be made to toggle if it is connected at port pin P40. The code demonstrates PWM mode by setting up the PPG Period (Cycle Setting Register PCSR) and the Duty Cycle Register (PDUT). $PDUT < PCSR$ should be ensured for correct operation.

```
/*-----*/
/*    MAIN.C    */
/*-----*/

#include "mb90425.h"

IO_WORD Intensity = 0;    /*variable to control the intensity of the LED */

int temp;

void main(void)
{
    DDR0 = 0xDF;          /* Set Port Data Direction Register: Output */
    DDR4 = 0xF0;          /* Set port to Output */
    PDR4 = 0xFF;          /* LED's off */

    InitPPG();            /* clock select to lus */
    InitReloadTimer();
    InitIrqLevels();

    __set_il(7);          /* set ILM to 7, allow all interrupts */
    __EI();               /* enable interrupts at all */
    PCNT0_CNTE = 1;       /* Enable PPG Operation */
    PCNTH0_STGR = 1;      /* Software Trigger */

    while(1)
        __asm("\tnop");
}

/* -----*/
/* Utility Routines    */
/*-----*/

void InitPPG (void)
{
    PCNTH0_STGR = 0;      /* Software Trigger */
    PCNTH0_MDSE = 0;      /* CONTINUOUS PWM OPERATION */
    PCNTH0_RTRG = 0;      /* ENABLE RESTART AS CONTINUOUS OPERATION */
    PCNTH0_CKS1 = 0;      /* USE SYSTEM CLOCK */
    PCNTH0_CKS0 = 0;
    PCNTH0_PGMS = 0;      /* PPG O/P used when 1 written to PGMS */
    PCNTL0_EGS1 = 0;      /* RISING EDGE SELECT */
    PCNTL0_EGS0 = 1;
    PCNTL0_IREN = 1;      /* Interrupt Enabled */
    PCNTL0_IRS1 = 0;
    PCNTL0_IRS0 = 1;      /* 1 allows external trigger */
    PCNTL0_POEN = 1;      /* ENABLE PPG0 OUTPUT */
    PCNTL0_OSEL = 0;      /* Normal polarity Output select */

    PCSR0 = 0xFFFF;      /* SET THE PERIOD - write to Cycle register before Duty Register */
    Intensity = 0xFFFE;
}
```

Application Note

```
void InitReloadTimer(void)
{
    TMRLR0 = 0x0010;    /* Reload value for timer 0, was 0x0010 */
    TMCSR0 = 0x81B;     /* prescaler 2us at 16 MHz */
}

/* -----*/
/*          Interrupt Service routines          */
/*-----*/

/* The interrupt has been set up in Vectors.c */

__interrupt void PPG0 (void)
{
    PDR4_P47 = ~PDR4_P47;    /* Toggles LED D8 */
    PCNTL0_IRQF = 0;         /* Clear Interrupt Flag */
    PDUT0 = Intensity;       /* Set Duty Width */
    Intensity = Intensity - 0x30;

    if (Intensity < 0x30)
    {
        PCNTH0_STGR = 0;     /* Software Trigger */
    }
}

__interrupt void ReloadTimer(void)
{
    TMCSR0_UF = 0;
}
```

FUJITSU MICROELECTRONICS AMERICA, INC.

Corporate Headquarters

1250 East Arques Avenue, M/S 333, Sunnyvale, CA 94088-3470

Tel: (800) 866-8608 Fax: (408) 737-5999

E-mail: fmicrc@fmi.fujitsu.com Internet: <http://www.fma.fujitsu.com>

© 2003 Fujitsu Microelectronics America, Inc.
All rights reserved. All company and product
names are trademarks or registered
trademarks of their respective owners.

MCU-AN-20977-5/2003