

# Generating a Large Number of Independent PWMs



## Introduction

Many applications, such as adjusting LED brightness and controlling the speed of electric motors, require an analog signal as an input; for example, an analog waveform can be used to progressively dim and brighten an LED. Pulse width modulation (PWM) using an MSP430™ microcontroller (MCU) provides a low-cost method to generate an analog signal using a digital source.

The analog signal produced by a PWM is in the form of unipolar square waves toggling between on and off states. The behavior of the PWM output is determined by its duty cycle and frequency. The average voltage of a PWM output can be used in a variety of applications. To control the toggling of the PWM output, timer modules on an MSP430 MCU can be used.

This document describes how to use the multiple time base method with the Timer A3 and Timer B0 modules and their corresponding capture/compare registers (CCRs) on the MSP430FR2476 MCU to generate independent PWMs at varying frequencies and duty cycles. Each CCR on each timer module is used to generate a unique PWM output. See the [Multiple Time Bases on a Single MSP430 Timer Module](#) application note for a more detailed explanation of the multiple time base method. To get started, [download the project files and example code](#).

## Implementation

This solution uses the MSP430FR2476 MCU and the UART backchannel of the MCU to allow the user to interact with the code and modify the features of the independent PWMs. A 16 MHz clock and a 9600 baud rate are used for the stand-alone demo and a 115200 baud rate is used for the GUI demo. The PWMs are output on general-purpose input/output (GPIO) pins found on the MCU. When the code is not executing PWM-specific tasks, the microcontroller enters low-power mode 3 (LPM3) to reduce power consumption. The main program exits low-power mode when an interrupt is triggered from either a UART command or from a timer module.

Continuous mode and output mode 4 (toggle mode) are used to initialize the timer modules. Timer

interrupts occur when the value in a timer's counter register is equal to the value in a timer's CCR. When this timer interrupt occurs, the output is toggled between high and low states, generating a PWM at varying intervals. The duty cycle and frequency of the PWM are determined by the value in its corresponding CCR; therefore, the values in the CCRs are changed through UART communication to change these properties of the PWMs. PWMs are initialized to a default duty cycle of 20% and a frequency of 1000 Hz.

Figure 1 shows how the MSP430FR2476 can interact with the UART backchannel or another host MCU or processor to change the CCR values. Only Timer A3 is shown here for demonstration.

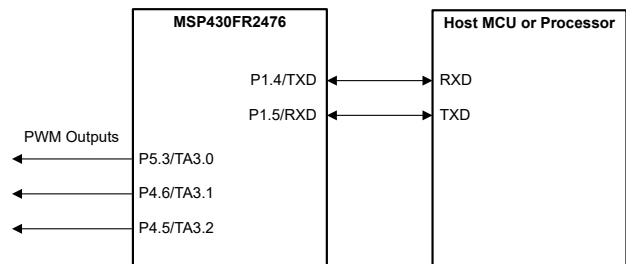


Figure 1. Generating PWMs Block Diagram

Table 1 lists the UART commands for the stand-alone code. Invalid commands are ignored.

Table 1. Valid UART Commands

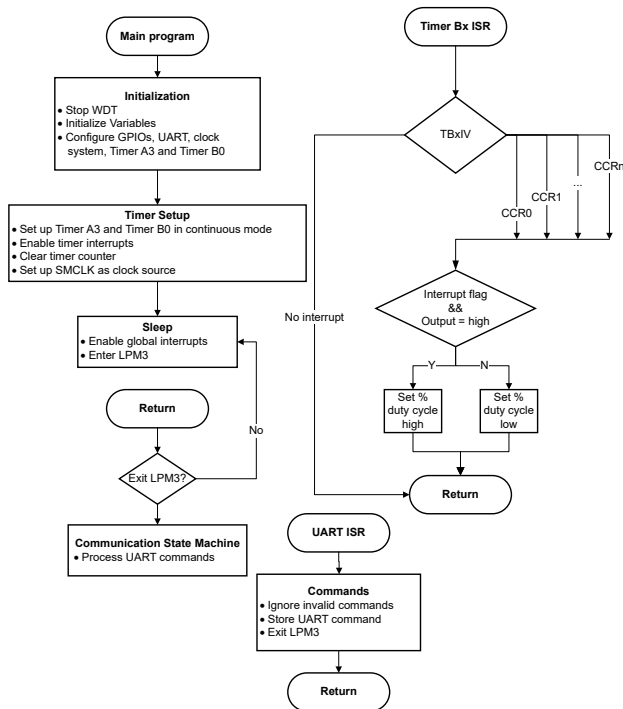
UART Command (hex)	UART Command (keyboard character)	Description
0x61	'a'	Choose TA3CCR0 to modify
0x62	'b'	Choose TA3CCR1 to modify
0x63	'c'	Choose TA3CCR2 to modify
0x64	'd'	Choose TB0CCR0 to modify
0x65	'e'	Choose TB0CCR1 to modify
0x66	'f'	Increase duty cycle by 10%
0x67	'g'	Decrease duty cycle by 10%
0x68	'h'	Increase frequency by 500 Hz
0x69	'i'	Decrease frequency by 500 Hz
0x6A	'j'	Flag to signal start and end of user inputted digits for frequency

**Table 1. Valid UART Commands (continued)**

UART Command (hex)	UART Command (keyboard character)	Description
0x6B	'k'	Flag to signal start and end of user inputted digits for duty cycle

When one of the keyboard characters in [Table 1](#) is pressed while running the stand-alone code, the PWM for a specific timer and CCR is able to be modified. The UART interrupt service routine (ISR) saves a valid command to a global variable that is used by the main program to determine how to modify the PWM. The main program calculates how many counts to add to the CCR or how to change the CCR's value. The actual value is changed in a timer module's ISR.

[Figure 2](#) shows how the interrupt service routines of the timer and UART work with the main program to produce and modify various PWMs.


**Figure 2. Generating PWMs Software Flow Chart**

The stand-alone code and GUI code can be implemented on other MSP430 devices; however, the timers and pins that can output a PWM vary by device. Therefore, the HAL file must be modified for the GUI code and the GPIO and timer initialization must be modified for all code used.

## Performance

This solution uses the Timer A3 and Timer B0 modules to output PWMs at a desired duty cycle

and frequency. [Table 2](#) lists the CCRs on the MSP430FR2476 and their corresponding GPIO pins. These can be added by initializing the pins in the code and by adding the appropriate timer initialization.

**Table 2. Timer-Pin Connections**

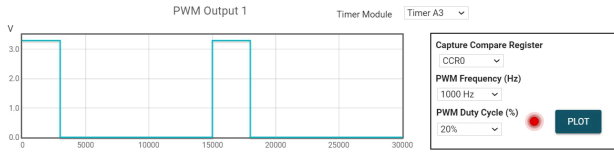
Timer A0	
TA0.1 (CCR1)	P1.1
TA0.2 (CCR2)	P1.2
Timer A2 <sup>(1)</sup>	
TA2.1 (CCR1)	P5.7
TA2.2 (CCR2)	P6.0
Timer A3 <sup>(1)</sup>	
TA3.0 (CCR0)	P5.3
TA3.1 (CCR1)	P4.6
TA3.2 (CCR2)	P4.5
Timer B0	
TB0.0 (CCR0)	P6.2
TB0.1 (CCR1)	P4.7
TB0.2 (CCR2)	P5.0
TB0.3 (CCR3)	P5.1
TB0.4 (CCR4)	P5.2
TB0.5 (CCR5)	P4.3
TB0.6 (CCR6)	P4.4

- (1) Timer A2 and Timer A3 require the pin remapping functionality for the MSP430FR2476. The lines `SYSCFG3 |= TA2RMP;` and `SYSCFG3 |= TA3RMP;` must be included in the port initialization code.

To run the stand-alone demo, connect the [MSP430FR2476 LaunchPad™ Development Kit](#) to a PC. Open a terminal in either Code Composer Studio (CCS) or another PC terminal program and ensure Serial Terminal is selected, the baud rate is 9600, and the serial port is connected to the device's backchannel UART. Any of the keyboard commands outlined in [Table 1](#) can be typed into the terminal to change a PWM's properties. Using a logic analyzer connected to the corresponding GPIO pins is recommended for this implementation to visualize the PWM outputs.

To run the GUI demo, the MSP430FR2476 launchpad should be connected to the PC. The GUI is used as the host and sends commands to the CCS code. A maximum of three PWMs can be viewed on the GUI at a time. Drop-down lists can be used to change which timer module is being accessed, which CCR is being modified or viewed, and the duty cycle and frequency of the PWM. The GUI plot and the PWM output from the MCU are not modified until the PLOT button is pressed. When PLOT is pressed, the LED turns on to indicate that the MCU has received the data from the GUI; otherwise, the LED remains off when using the drop-down lists to change parameters.

An example of the GUI plot and drop-down lists is shown in [Figure 3](#).



**Figure 3. Generating PWMs GUI Demo**

**To Get Started**

1. Watch the training video [Generating a Large Number of Independent PWMs using a Housekeeping MCU](#) to learn how to use the GUI to visualize PWM outputs and change their properties.
2. Order an [MSP430FR2476 LaunchPad Development Kit](#) to evaluate the *Generating a Large Number of Independent PWMs* example GUI and code.
3. Download and test the [Generating a Large Number of Independent PWMs example GUI](#).
4. Evaluate the [Generating a Large Number of Independent PWMs example code](#) using the [MSP430FR2476 LaunchPad Development Kit](#).

Part Number	Key Features
MSP430FR2476	16 MHz MCU with 64KB FRAM, 8KB SRAM comparator, 12-bit ADC, UART, SPI, I2C, timer

## IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to [TI's Terms of Sale](#) or other applicable terms available either on [ti.com](http://ti.com) or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265  
Copyright © 2022, Texas Instruments Incorporated