

Example: Example_Push_Pull_PWM

Programming Language: C

Associated Part Families: CY24x23, CY27x43, CY8C29x66, CY8C24x94

Software Version: PD 5.2

Objective

CE52125 example creates a two phase symmetrical push pull PWM with the PWMDB8 User Module.

Overview

Using the PWMDB User Module, a two phase symmetrical PWM with dead time and variable duty cycle is created. The duty cycle of the dead band generator is controlled by two switches which increase or decrease the dead time of the PWM and subsequently the pulse width.

User Module List and Placement

The following table lists user modules in this project and the hardware resources occupied by each user module.

User Module	Placement
PWMDB	DBB10 and DBB11
LCD	Port_2

User Module Parameter Settings

The following tables show the user module parameter settings for each user module in the project.

PWMDB8_1		
Parameter	Value	Comments
Clock	VC1	The clock to the PWMDB8 is taken from VC1, which is 2 MHz.
Enable	High	The Enable signal is connected to VCC
Period	199	Period is set to 199. This results in a divider of 200; the output frequency is 2 MHz / 200 = 10 kHz.
Pulse Width	100	The pulse width of the reference PWM is set to 100. Varying the dead time from 0 to 100 varies the pulse width of both the phases.
Interrupt Type	Terminal Count	Not used in this project.
PWMOutput	None	Not used in this project.
DeadTime	50	This parameter sets the dead time count of the DB8 output. An 8-bit value in the range of zero to the minimum of the following: PWM Period parameter minus two, PWM Pulse Width parameter value minus two, or 255.
Phase1	Row_0_Output_0	Phase1 is routed to P1[4] through the GlobalOutOdd_4 net.
Phase2	Row_0_Output_1	Phase2 is routed to P1[5] through the GlobalOutOdd_5 net.
DeadBandKill	Low	When asserted high, Phase1 and Phase2 outputs are driven low.
DeadBandKill_Mode	DisableKill	Disabled in this project. Can be chosen as synchronous and asynchronous mode.
ClockSync	Sync to SysClk	As VC1 is derived from SysClk, clock synchronization is set to Sync To Sysclk
InvertDeadBandKill	Normal	If used, allows the user to invert the incoming DeadBand Kill signal.

PWMD8_1		
Parameter	Value	Comments
InvertEnable	Normal	If used, this parameter allows the user to invert the incoming Enable signal.

Note The deadband kill input, when made high makes the output from both the phases low. This feature may be used in feedback loop of a power supply where a comparator may be used to control the kill input in a feedback loop, thus regulating the output voltage.

LCD		
Parameter	Value	Comments
LCDPort	Port_2	The LCD is connected to Port 2.
BarGraph	Disable	Bar graph function is disabled.

Global Resources

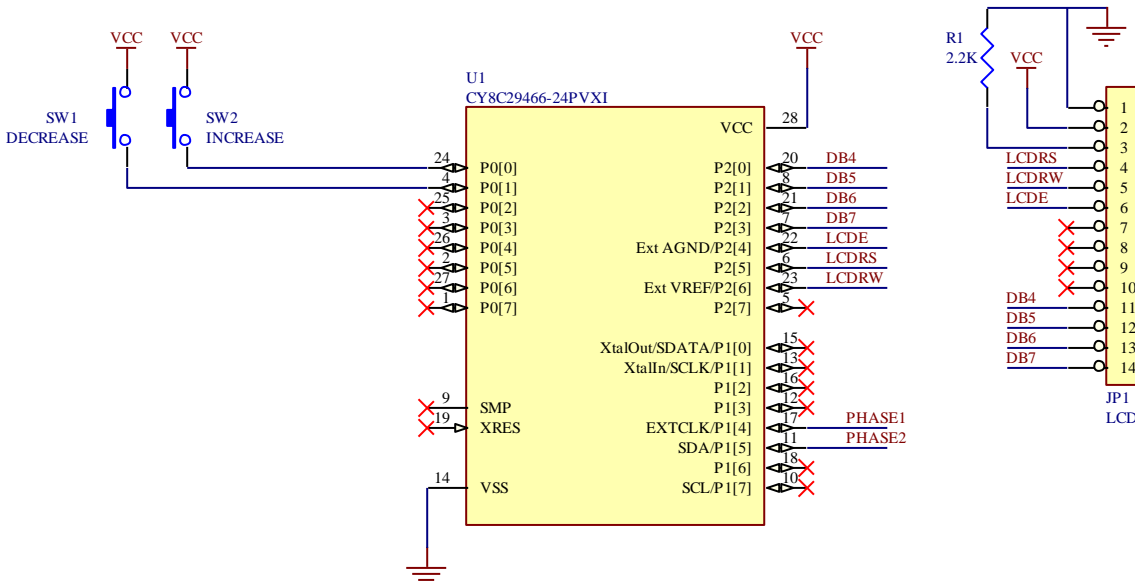
Important Global Resources		
Parameter	Value	Comments
Power Setting	5V/24MHz	Selects 5V operation and 24 MHz SysClk.
CPU Clock	SysClk/8	Sets CPU Clock to 3 MHz.
VC1	12	VC1 frequency set to 2 MHz.

Note The above table lists the global resources specific to the PWMD8. Other parameters are left at their default value.

Hardware Connections

The schematic diagram shows the connections for the project. SW1 and SW2 are connected to P0[0] and P0[1] respectively and are used to control the pulse width of the outputs. An alphanumeric LCD display is connected to P2 and displays the dead time value of the PWMD8. The outputs of the PWM are available on P1[0] and P1[1]. The project may also be tested on the CY3210 PSoC Eval1 board.

Figure 1. Project Schematic Diagram

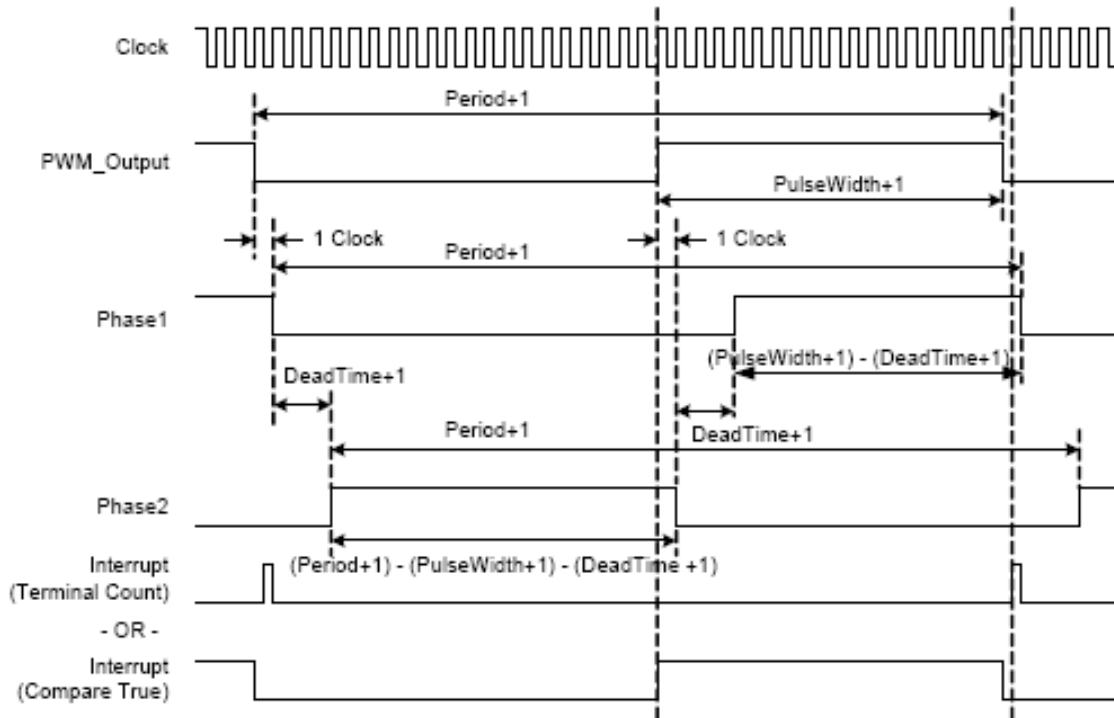


- SW1 and SW2 may be assembled on the bread board area of the CY3210 board. One end of the switches are connected to P0[0] and P0[1] on J6. The other end of the switches are tied to VDD.
- The LCD is connected to J9.
- The outputs are observed on P1[4] and P1[5] on J7.

Operation

The following timing diagram of a PWMDB8 User Module is taken from the user module data sheet.

Figure 2. Timing Diagram of PWMDB8



$$\text{Width of Phase 1} = (PulseWidth + 1) - (DeadTime + 1)$$

$$\text{Width of Phase 2} = (Period + 1) - (PulseWidth + 1) - (DeadTime + 1)$$

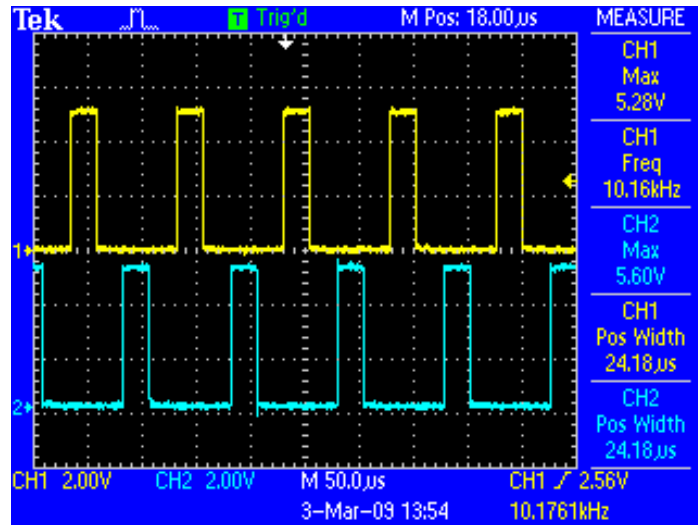
From these relations, by setting $(Pulsewidth + 1)$ to half of $(Period + 1)$ and varying the dead time from 0 to $(Pulsewidth + 1)$, symmetrically varies the On time of Phase1 and Phase2. When DeadTime is 0, both Phase1 and Phase2 have equal width and hence a duty cycle of 50%. When DeadTime equals $(PulseWidth + 1)$, the width of both Phase1 and Phase2 is 0 and results in 0% duty cycle. In the PWMDB8 user module parameter, the Period is set to 199 and Pulsewidth is set to 99. With this setting, varying the dead time from 0 to 99 produces a symmetrical duty cycle of 0 to 50% on both the phases.

Firmware

On reset, all hardware settings from the device configuration are loaded into the device and *main.c* is executed. The following operations are performed in *main.c*:

1. PWMDB8 is started.
2. LCD is initialized.
3. An infinite Loop is entered.
4. SW1 is checked. If found active, the *dead_time* variable is decreased by 10. If the value of *dead_time* is less than or equal to 10, the dead time value is set to 0 and "Max Duty" is displayed on LCD. If not maximum duty cycle, the LCD is updated with the dead time value. The dead time of the PWMDB8 is updated. Wait till SW1 is released.
5. SW2 is checked. If found active, the *dead_time* variable is increased by 10. If the value of *dead_time* greater than or equal to 90, the dead time value is limited to 99 and "Min Duty" is displayed on LCD. Otherwise, the *dead_time* value is displayed on LCD. The dead time of the PWMDB8 is updated. Wait till SW2 is released.

Figure 3. Output Signals on Phase1 and Phase2



PSoC is a registered trademark of Cypress Semiconductor Corp. "Programmable System-on-Chip," and PSoC Designer are trademarks of Cypress Semiconductor Corp. All other trademarks or registered trademarks referenced herein are the property of their respective owners.

Cypress Semiconductor
 198 Champion Court
 San Jose, CA 95134-1709
 Phone: 408-943-2600
 Fax: 408-943-4730
<http://www.cypress.com/>

© Cypress Semiconductor Corporation, 2009-2011. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress product. Nor does it convey or imply any license under patent or other rights. Cypress products are not warranted nor intended to be used for medical, life support, life saving, critical control or safety applications, unless pursuant to an express written agreement with Cypress. Furthermore, Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress products in life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

This Source Code (software and/or firmware) is owned by Cypress Semiconductor Corporation (Cypress) and is protected by and subject to worldwide patent protection (United States and foreign), United States copyright laws and international treaty provisions. Cypress hereby grants to licensee a personal, non-exclusive, non-transferable license to copy, use, modify, create derivative works of, and compile the Cypress Source Code and derivative works for the sole purpose of creating custom software and or firmware in support of licensee product to be used only in conjunction with a Cypress integrated circuit as specified in the applicable agreement. Any reproduction, modification, translation, compilation, or representation of this Source Code except as specified above is prohibited without the express written permission of Cypress.

Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Use may be limited by and subject to the applicable Cypress software license agreement.