Pulse Width Modulator(PWM) example project
2.0

Features

- PWM with 8 bit Resolution and Fixed Function Implementation mode example.

General Description

This example project demonstrates the working of the PWM Fixed Function Block.

Development kit configuration

1. This project is written for a 2X16 LCD display as the one available in Cypress kit CY8CKIT-001.
2. Build the project and program the hex file on to the target device using MiniProg3.
3. Connect pins as described below and power cycle the device.
4. Observe the results on the LCD.

Project configuration

This project consists of the PWM component with digital output pins, clock, Logic Low, Control Register and Status Register components. The top design schematic is shown in Figure 1. The Clock input is used to increment or decrement the counter on each rising edge of the clock and a Control Register component is used to write a zero to kill the input. The Status register component is used to read the current status of the PWM output.

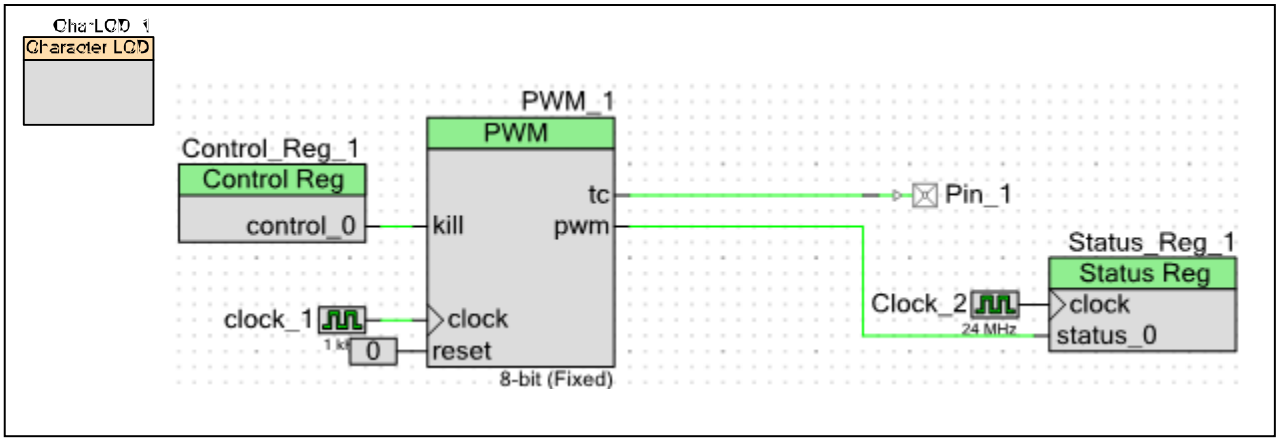

Figure 1: Top design schematic

The Character LCD Component has the default configuration. It is used to display the current Period, Compare values and the current status of the PWM output. The PWM Component configuration is shown below in Figure 2.

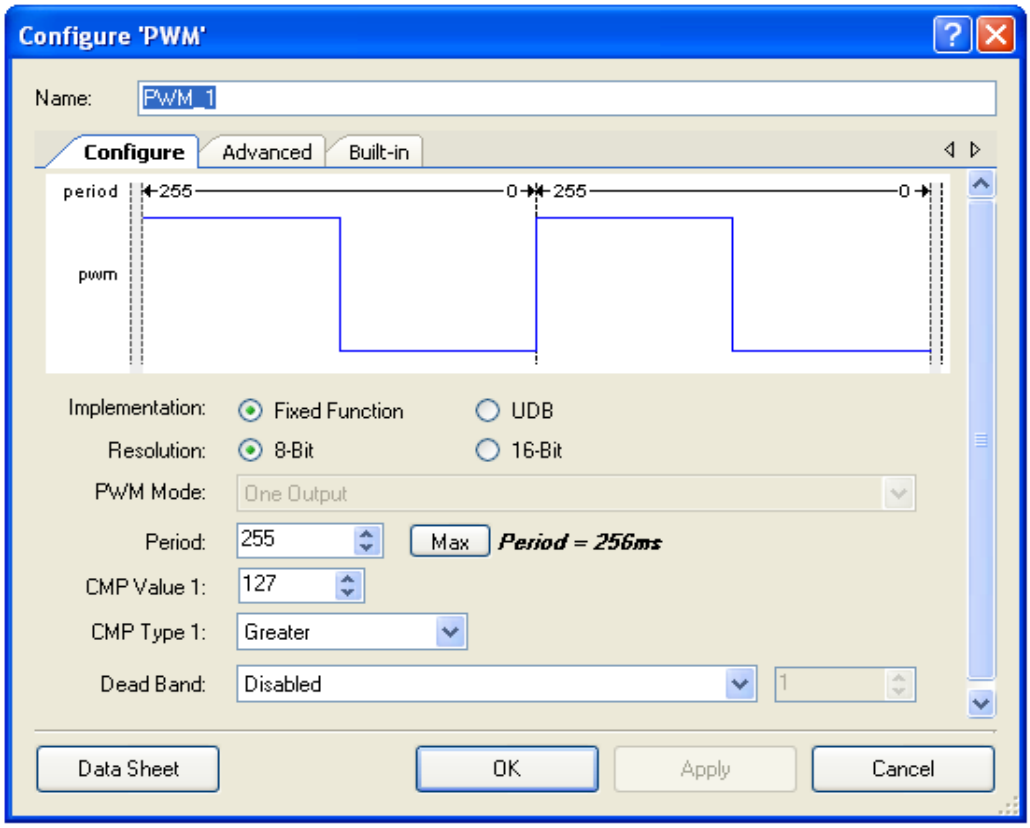

Figure 2: PWM Component configuration

Project description

This is an example to use the PWM Component with designs in PSoC Creator. The PWM component is configured to set Period value to 255, PWM_1_ReadPeriod() function is used to find the current period value. Compare value is set to 127, PWM_1_ReadCompare() function is used to read the current compare value. The Control Register Component is used to specify the desired behavior of Kill Input. In the “Fixed Function” PWM implementation low to kill input will not kill the deadband outputs. Deadband outputs are visible only if the deadband parameter is enabled. Set the value of Control Register output by writing to Control Register using Control_Reg_1_Write() function. The Status Register Component is used to read the current status of PWM output using Status_Reg_1_Read() function. On rising edge of clock input, the PWM starts down counting from the Period value 255 to zero.

Expected results

The LED connected to P0.0 shows the TC output. It turns on when the period counter is equal to zero.
The first row of character LCD displays the test name.
The second row of character LCD displays the current Period, Compare value and the current status of PWM output.

© Cypress Semiconductor Corporation, 2009-2012. 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.

PSoC® is a registered trademark, and PSoC Creator™ and Programmable System-on-Chip™ are trademarks of Cypress Semiconductor Corp. All other trademarks or registered trademarks referenced herein are property of the respective corporations.

Any 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.