

Timer Example Project

1.00

Features

- 16-bit UDB Timer
- Sync block to ensure error-free timing

General Description

This example project demonstrates the working of the 16-bit UDB Timer component and shows how to read the Period, Capture and Counter values.

Development Kit Configuration

The following configuration instructions provide a guideline to test this design. For simplicity, the instructions describe the stepwise process to be followed when testing this design with the PSoC Development Kit (CY8CKIT-001) board, but can be generalized for the PSoC 3 Development Kit (CY8CKIT-030) and PSoC 5 Development Kit (CY8CKIT-050) as well.

1. Set LCD power jumper J12 to ON position and leave the rest of the board at default configuration.
2. Observe Capture_out - P0_0 and TC - P0_1 on an oscilloscope.
3. Ensure that the Character LCD is connected to header P18 on the development board.

Project Configuration

This project has a schematic as shown in Figure 1. The Clock component - clock_1 is connected to the Clock input of the timer and is set to 10kHz. The Clock component - Clock_2 of 200 Hz frequency is connected to the Sync block before going to the Capture input (Global clock signals cannot be used as data inputs because the timing for these signals, when used in this mode, is not guaranteed to meet setup with respect to another clock). The timer is configured (see Figure 2) to have a period of 35536 and a capture on each rising edge of the capture input. In order to be able to observe the capture_out and TC outputs of the Timer, these are brought out using two digital pins P0_0 and P0_1. The interrupt output is connected to an ISR component. The Character LCD is left in default mode.

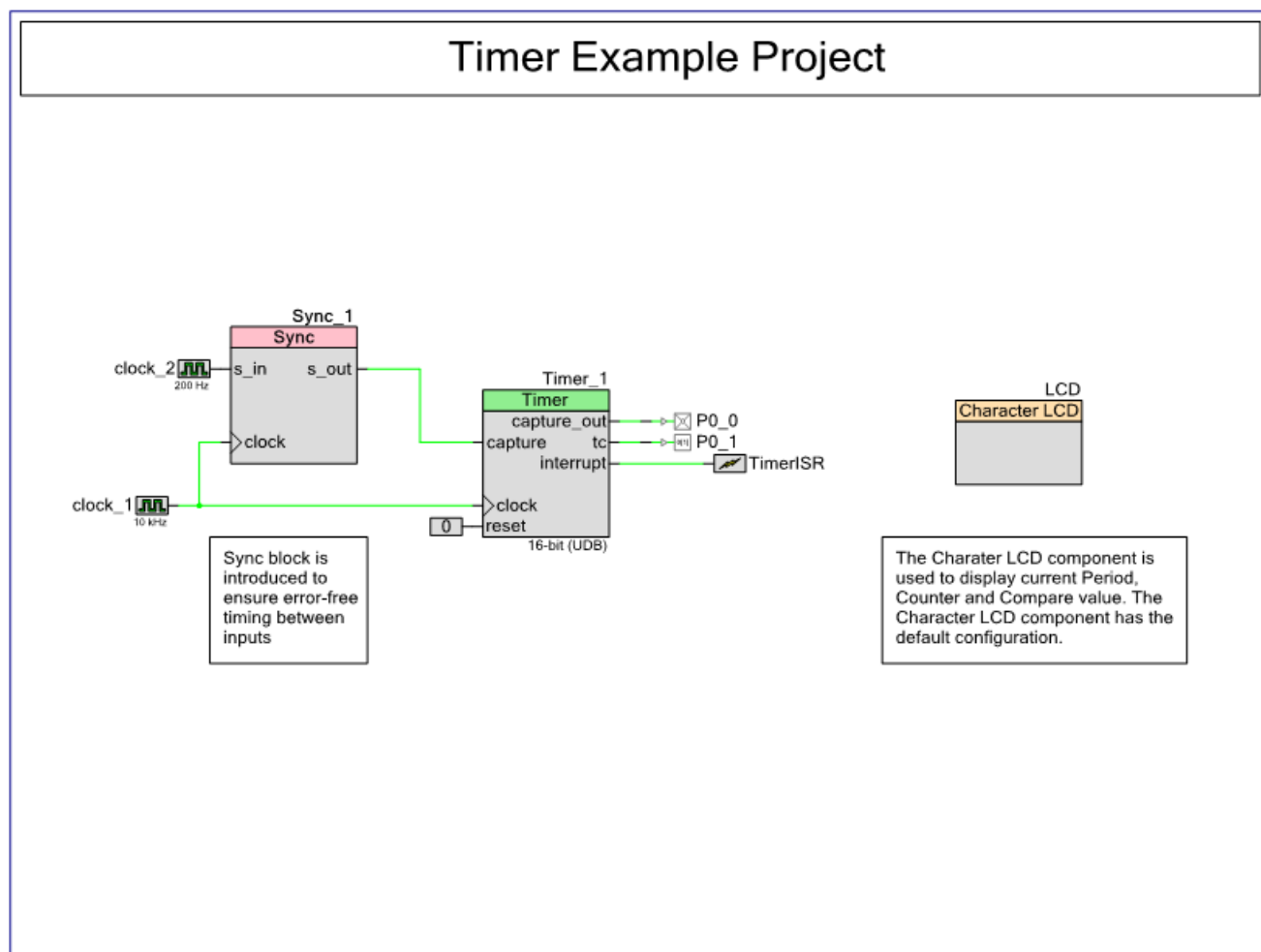


Figure 1. TopDesign schematic

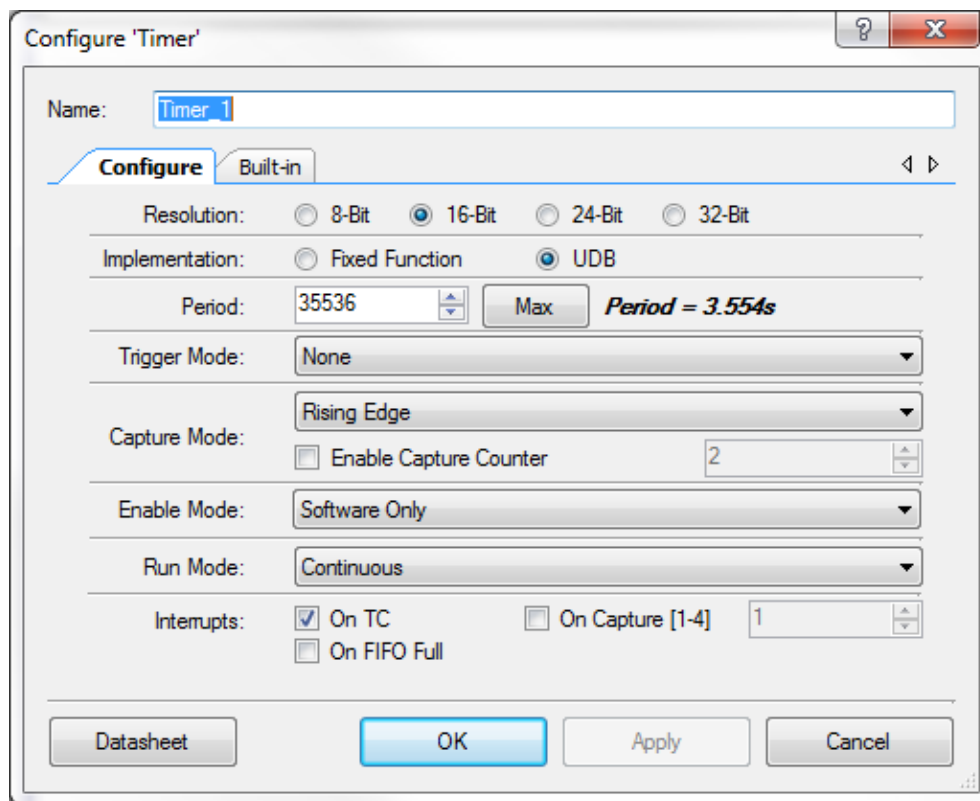


Figure 2. Timer configuration

Project Description

The LCD, Timer and interrupt are started in the main function. Global interrupts are enabled and the timer interrupt vector is assigned. The Timer in the example project counts from the Period value 35536 down to the terminal count 0. The Timer is configured to trigger the interrupt on the terminal count event. On reaching the terminal count, the triggered ISR routine reads the status register to clear the interrupt and also increments the interrupt count. The Character LCD is used to display the test name and Period, Capture, and Count values. The interrupt count that is returned by the ISR is also displayed on the Char LCD. These are updated continually in the 'forever' loop.

Expected Results

Capture_out P0_0: This pin gives a high pulse for every rising edge of the capture input.

TC P0_1: This pin gives a high pulse for every 35536 clock cycles (3.554s).

Char LCD displays the following:

1st row:

TMR-16 (To indicate that a 16 bit timer is used)

Period value = 8ACF

Captured value = Varying (decrements to 0 and reloads)

2nd row:

Counter value = Varying (decrements to 0 and reloads)

IntCnt: Displays the interrupt count



Figure 3. Expected output on LCD

Related Material

Example Projects

- Counter

Application Notes

- [AN54181 - PSoC® 3 - Getting started with a PSoC 3 design project](#)

Training

- [PSoC 3 and PSoC 5 103: Introduction to Digital Peripherals](#)



Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

Phone : 408-943-2600
Fax : 408-943-4730
Website : www.cypress.com

© Cypress Semiconductor Corporation, 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.

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.

