Timer Project
1.10

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.

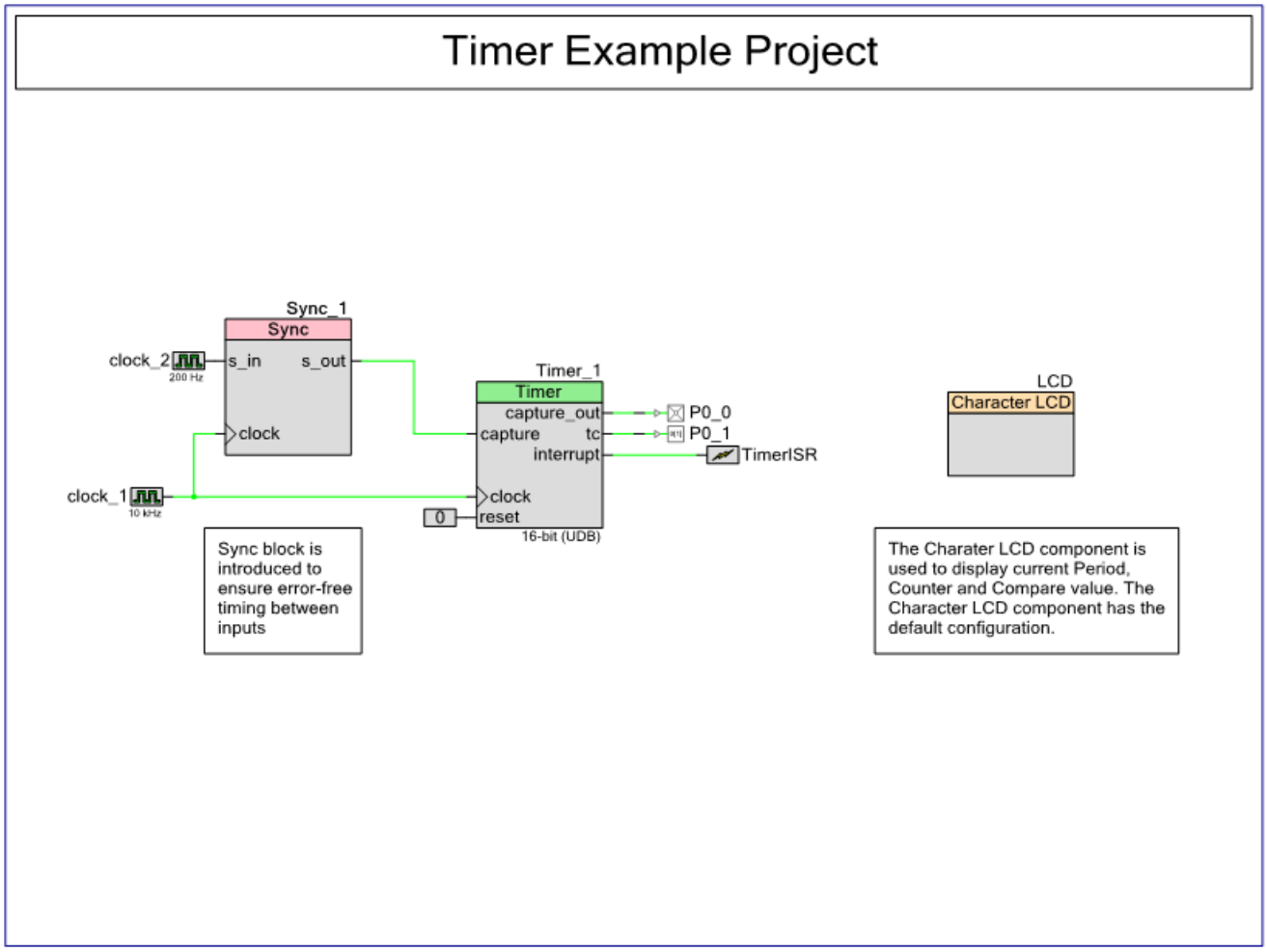

Figure 1. TopDesign schematic

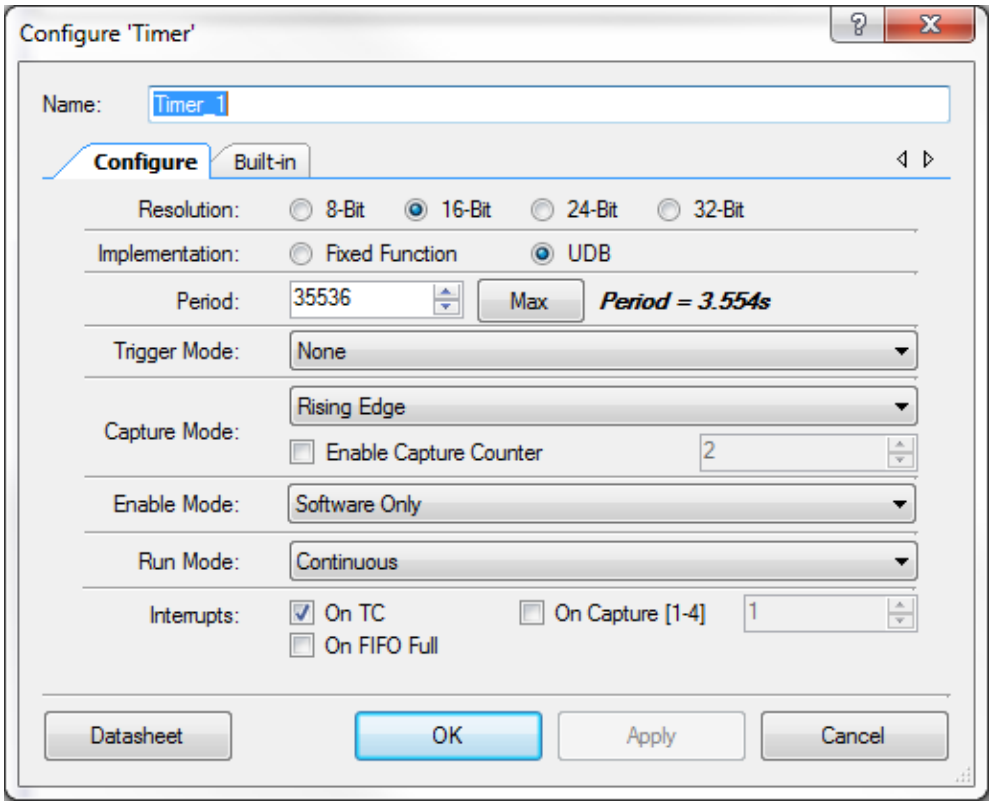

Figure 2. Timer configuration

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, 2012-2017. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress's patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage ("Unintended Uses"). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, WICED, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.