

# Down Counter 7-bit (Count7) Example Project

## 1.0

## Features

- Demonstrates basic functionality of Count7 with routed enable signals
- Decrements count value on every press button event and displays it on LCD with user specified refresh rate

## General Description

The Count7 component is the most resource efficient implementation of a 7-bit counter that also exposes the count value directly as hardware signals. This example project demonstrates how to count press button events completely in hardware using the Count7 and Debouncer components.

## Development Kit Configuration

This example project is designed to run on CY8CKIT-001 from Cypress Semiconductor. A full description of the kit, along with more example programs and ordering information, can be found at <http://www.cypress.com/go/cy8ckit-001>.

For simplicity, the instructions describe the stepwise process to be followed when testing this design with CY8CKIT-001.

1. Connect P1[7] to SW1 on the CY8CKIT-001 development board
2. Build the project and program the hex file into the target device using a MiniProg3 programmer
3. Press SW1 to decrement the count value and observe the results on the LCD.

## Projects Configuration

The example project consists of the Debouncer, Count7, Status register, and Character LCD components. The design schematic is shown in [Figure 1](#). The blue annotation components in the schematic show the external connections required.

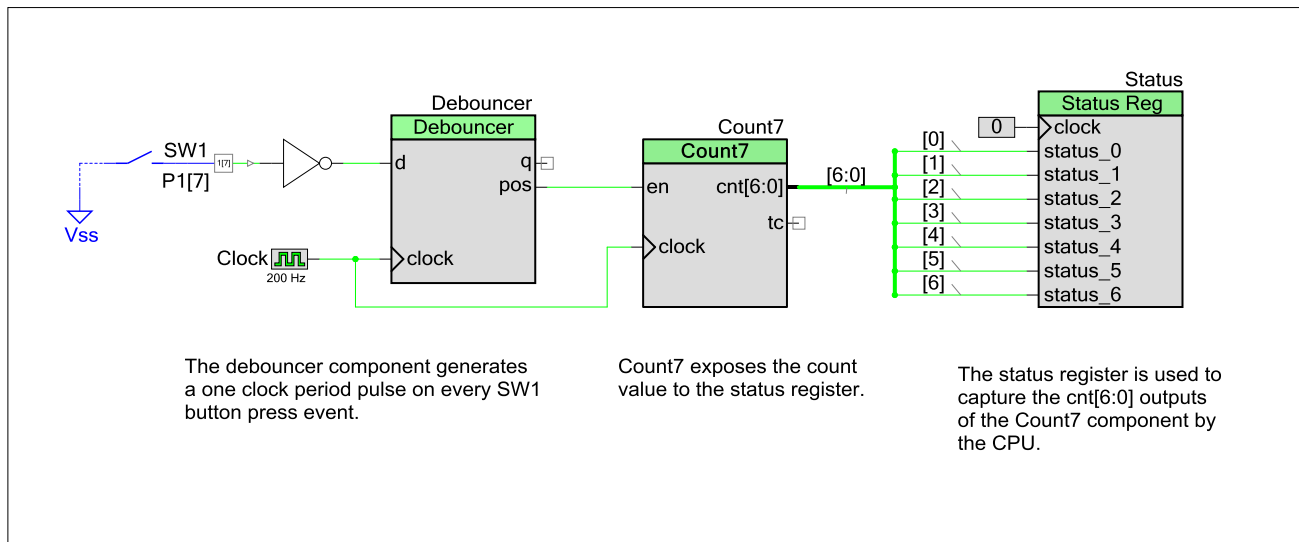
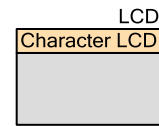
On every press button event the debouncer component generates a one clock period pulse on the counter enable input. Thus count7 is enabled for a one clock cycle every time the switch button on DVK is pressed. The status register is used to provide a count output value to the CPU.

## The Count7 datasheet example project

### Description:

- Decrements the count value on each SW1 press button event
- Prints the count value on the LCD with a user defined refresh rate
- Demonstrates the operation of a routed enable signal

## User Interface



**Figure 1.** Count7 Component Example Project Schematic

## Project Description

In the main firmware routine all the components are started and the initial count value is displayed on the LCD. In the forever loop the count output value is read and displayed on the LCD with a user defined refresh rate. Pressing SW1 on the development kit decrements the count value.

## Expected Results

After programming the initial count value of 0 will be displayed on the LCD. First SW1 press will reload the count with a period of 0x14. On every subsequent SW1 press button event the count value is decremented. If the count value is equal 0, it will be automatically reloaded with a period value on the next SW1 press.



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

Phone : 408-943-2600  
Fax : 408-943-4730  
Website : [www.cypress.com](http://www.cypress.com)

© Cypress Semiconductor Corporation, 2014. 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.

