ButtonSw32 custom component Application Note
0.0

AN-ButtonSw32_00_A

ButtonSw32 basic applications

Associated Project: Yes
Associated Part Family: PSoC4, PSoC5LP
Software version: PSoC® Creator™ 4.0 SP1
Related application Notes: ButtonSw32 datasheet

This application note illustrates several projects utilizing custom button switch debouncer component, ButtonSw32, in various modes of operation. Several practical examples, including interactive LCD menu, are provided alongside this note.

Projects described in this AN:

1. Basic demo.
2. Long press demo.
3. Multiple debouncers (internal interrupt).
4. Multiple debouncers (external interrupt).
5. PSoC4 basic demo (TFF).
6. PSoC4 basic demo (pin clocking).
7. PSoC4 WDT timer.
8. PSoC4 WDT timer w/callback.
9. LCD menu (basic).
10. LCD menu (advanced).

Introduction

The ButtonSw32 component[*] is a debouncer which allows for easy interfacing switches and momentary switch buttons to PSoC projects. Presented below are some application examples utilizing the component. Accompanying projects are provided alongside the Application Note.

Standard feature of the Debouncer component include ability to process up to 32 buttons simultaneously without loss of performance. Component's ability to detect button pressed and released event allows for complex interactive HMI applications, such as menu browsing and parameters editing. Multiple instances of the component can be used in the project.

The Debouncer does not consume any UDB resources, performing all operations by CPU, and can be particularly useful for projects with limited UDB resources, such as PSoC4. The component can serve as a substitute for Cypress stock UDB Debouncer component when high-speed performance is not required, such as with manually operated switches and switch buttons.

The component was developed for educational and experimenting purposes as part of the PSoC Community Components library. Most of examples provided use an onboard LED to simulate user-defined action for buttons events.

The demo projects provided were intentionally developed using Creator v4.0 for compatibility with both older and newer versions of IDE (v4.2).

Examples were tested using CY8CKIT-059 PSoC5LP Prototyping Kit, and CY8CKIT-042 PSoC4200 Pioneer board. Basic projects are available in separate PSoC5 and PSoC4 versions.

[*] Hereafter referenced as a "Debouncer"

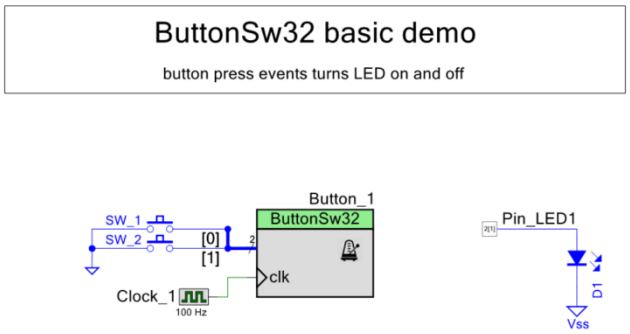

Figure 1. Debouncer basic example.

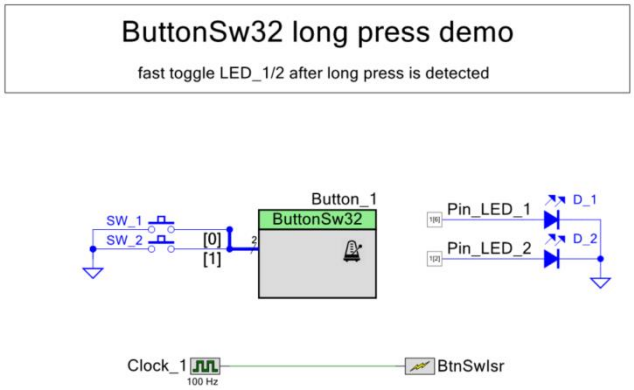

Figure 2. Long press example.

Basic demo

Associated project: ButtonSw_basic

PSoC5 basic example of switch debouncing using ButtonSw32 custom component operating in the internal interrupt mode (default). The component is configured for the buttons press event and internal interrupt. All switches are sampled at 100 Hz rate. Pressing any button toggles the LED on/off.

Button long press demo

Associated project: ButtonSw_long press_1c

PSoC5 example of Debouncer component operating in the external interrupt mode to achieve a 'long press' effect. The Debouncer is configured for buttons press and release events and external interrupt. All switches are sampled at 100 Hz rate. A short press of a button causes corresponding LED to blink once. When long press is detected (button stays pressed for longer than 1 sec) the LED starts blinking at fast pace of 10 Hz. Here an LED blinking simulates some user-defined action, such as a parameter increment/decrement on pressing Up/Down buttons.

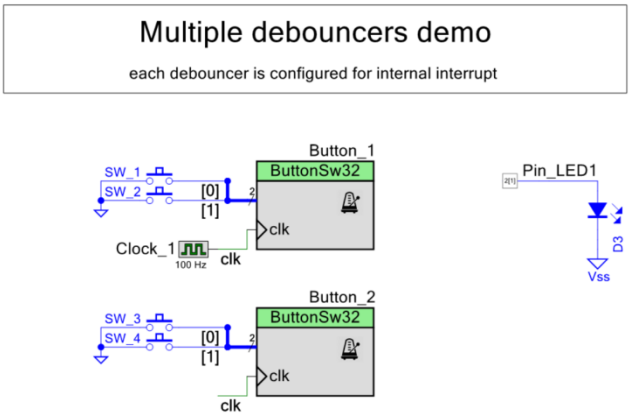

Figure 3. Multi-debouncer demo (internal interrupt).

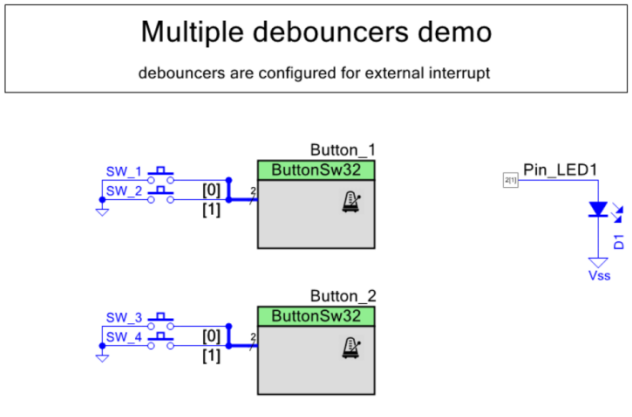

Figure 4. Multi-debouncer demo (external interrupt).

Multiple debouncers

Associated project: ButtonSw_2x

PSoC5 example of multiple instances of the Debouncer component in the project, operating in the internal interrupt mode. Multiple debouncers may be necessary when project needs more than 32 switches, or when debouncing intervals vary considerably for a set of switches. In such case a separate clocks must be used. In this demo all switches are sampled at same 100 Hz rate. The Debouncers are configured for buttons press event. Pressing buttons 1, 2 turns LED on, pressing buttons 3, 4 turns it off. In the internal mode each Debouncer consumes an interrupt, which are processed concurrently. To save resources it is possible to utilize a single clock and interrupt to service all debouncers synchronously, as shown in the next example.

Multiple debouncers (external interrupt)

Associated project: ButtonSw_2x_ext

PSoC5 example of multiple instances of the Debouncer component being handled using single external user-provided interrupt. This mode of operation saves interrupts and may be useful e.g. for PSoC4 where the number of available interrupts is limited. All switches are sampled at 100 Hz rate on clock interrupt, a flag is raised if button press event is detected. Pressing buttons 1, 2 turns LED on, pressing buttons 3, 4 turns it off. Using external mode saves interrupts, but requires extra custom code for handling of the user-provided interrupt.

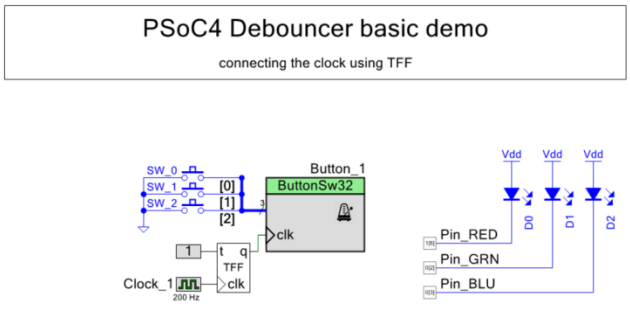

Figure 5. PSoC4 demo using TFF.

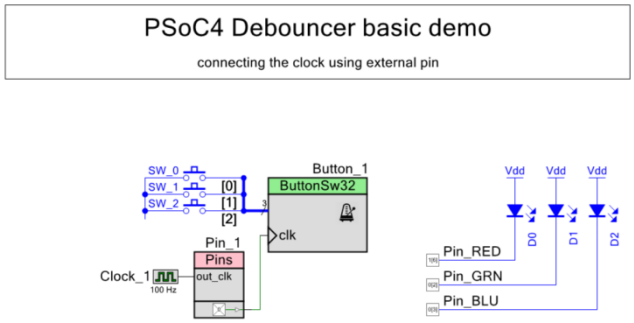

Figure 6. PSoC4 demo using pin clocking

PSoC4 basic demo (TFF)

Associated project: ButtonSw_P4_basic

PSoC4 example of the Debouncer component operating in the internal interrupt mode (default). All switches are sampled at 100 Hz rate. On PSoC4 a clock can't connect to an interrupt directly, so a TFF element is used to interface both. Since TFF divides frequency by half, the clock is set to 200 Hz. The Debouncer is configured for the buttons press event and internal interrupt. Pressing buttons 0, 1 or 2 toggles the red, green or blue LEDs. Using TFF consumes a macrocell; to save resources, use either pin clocking or WDT timer as shown in the next examples.

PSoC4 basic demo (pin clocking)

Associated project: ButtonSw_P4_pin_clock

PSoC4 example of the Debouncer component using pin clocking. On PSoC4 a clock can't connect to an interrupt directly, a pin clocking capability of PSoC4 is used to interface both. All switches are sampled at clock rate of 100 Hz. The Debouncer is configured for the buttons press event and internal interrupt. Pressing buttons toggles the LED on/off. When using pin clocking, the Debouncer does not consume UDB resources, but needs a spare pin. Using WDT timer as low frequency clock can further simplify design, as shown in the next two examples.

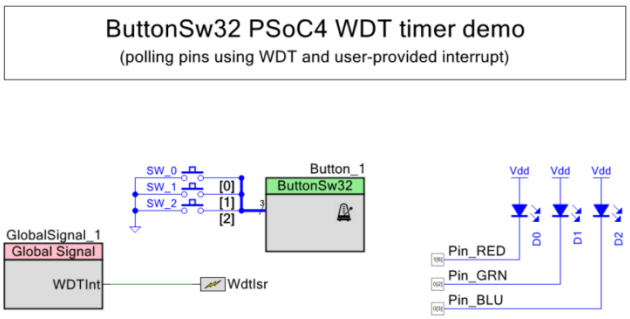

Figure 7. PSoC4 demo using WDT timer.

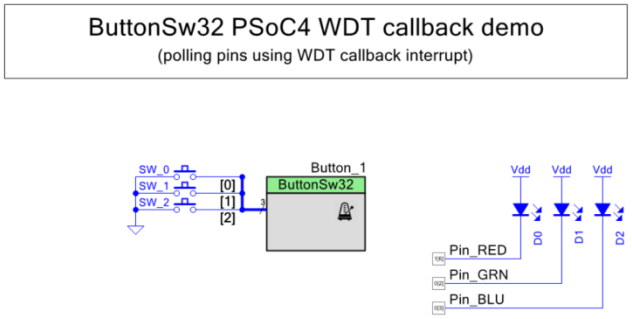

Figure 9. PSoC4 demo using WDT callback.

PSoC4 WDT timer

Associated project: ButtonSw_P4_WDT

PSoC4 example of the Debouncer utilizing WDT timer for polling pins state. The Debouncer is configured for external interrupt; pins state is checked in user-provided interrupt. Using PSoC4 build-in WDT timer instead of a clock is convenient due to the system limited hardware resources. In this example, the WDT0 timer interrupt configured as 'User provided' (Fig. 8), and triggered by the Global Signal. Pressing the buttons 0, 1 or 2 toggles the red, green or blue LED on the CY8KIT-042 Pioneer board.

PSoC4 WDT timer with callback

Associated project: ButtonSw_P4_WDT_callback

PSoC4 example of the Debouncer utilizing WDT timer with callback for polling pins state. In this example, the WDT0 timer interrupt configured as 'Auto generated' (Fig. 10). Debouncer is set for external interrupt, and pins state is processed using interrupt callback procedure, which automatically clears the interrupt source. Using PSoC4 build-in WDT timer instead of a clock is very convenient due to the system limited hardware resources. Pressing the buttons 0, 1 or 2 toggles the red, green or blue LED on the CY8KIT-042 Pioneer board.

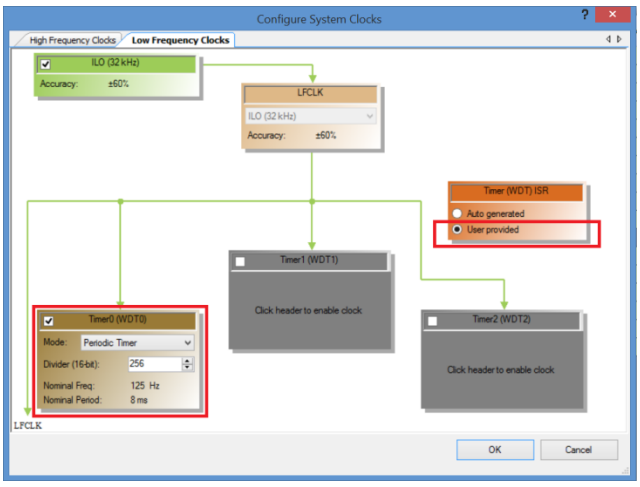

Figure 8. WDT timer configuration (user provided).

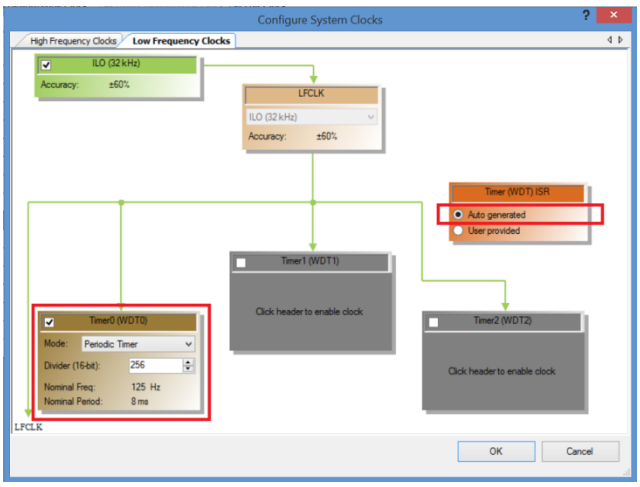

Figure 10. WDT timer configuration (auto generated).

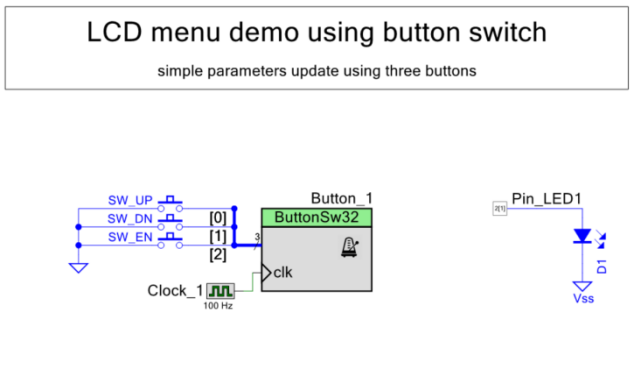

Figure 11. Basic menu demo using character LCD.

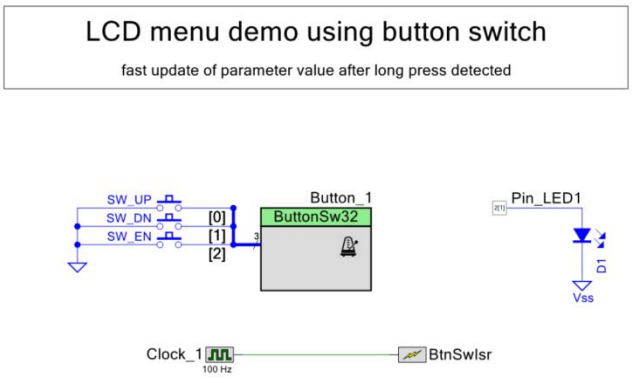

Figure 13. Advanced menu demo using character LCD.

LCD menu demo (basic)

Associated project: ButtonSw_CharLCD_1c

Basic example of the scrollable menu using 2x16 character LCD and three buttons. Cypress stock CharLCD component is used to interface LCD and PSoC. Three buttons (Up, Down and Enter) allow for vertical scrolling of the menu and modifying parameters values, which can be of integer, float or enumerated range types. Pressing the Enter button enters/exits parameter edit mode. A press on the Up/Down button increments/decrements parameter value by a predefined step.

This example doesn't have advanced features, e.g. button long press detection or acceleration. Such features can be implemented using Debouncer in external interrupt mode as shown in the next example.

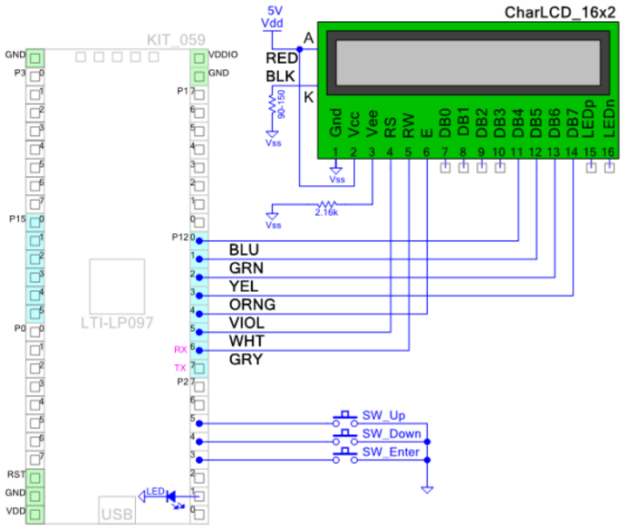

Figure 12. CY8KIT-059 wiring schematic.

LCD menu demo (advanced)

Associated project: ButtonSw_CharLCD_ex_1d

Advanced example of the scrollable menu using 2x16 character LCD and three buttons. Cypress stock CharLCD component is used to interface LCD and PSoC. Three buttons (Up, Down and Enter) allow for vertical scrolling of the menu and modifying parameters values, which can be of integer, float or enumerated range types. Pressing the Enter button enters/exits parameter edit mode. A short press on the Up/Down button increments/decrements a parameter value by a predefined step.

This example brings additional features: a long press and accelerated parameter update by using external interrupt mode. The interrupt code implements custom software Timer with delay function, configured for periodic code execution. If button Up (or Down) is pressed for longer than 1 sec, the Timer enters a "long press" mode, starting rapid (~10 Hz) updating of parameter value until the button is released. This feature is particularly useful when parameter range is too wide to be updated manually in step-by-step fashion.

Figure 14. Character LCD scrolling menu.