

ECE3623 Embedded System Design Laboratory

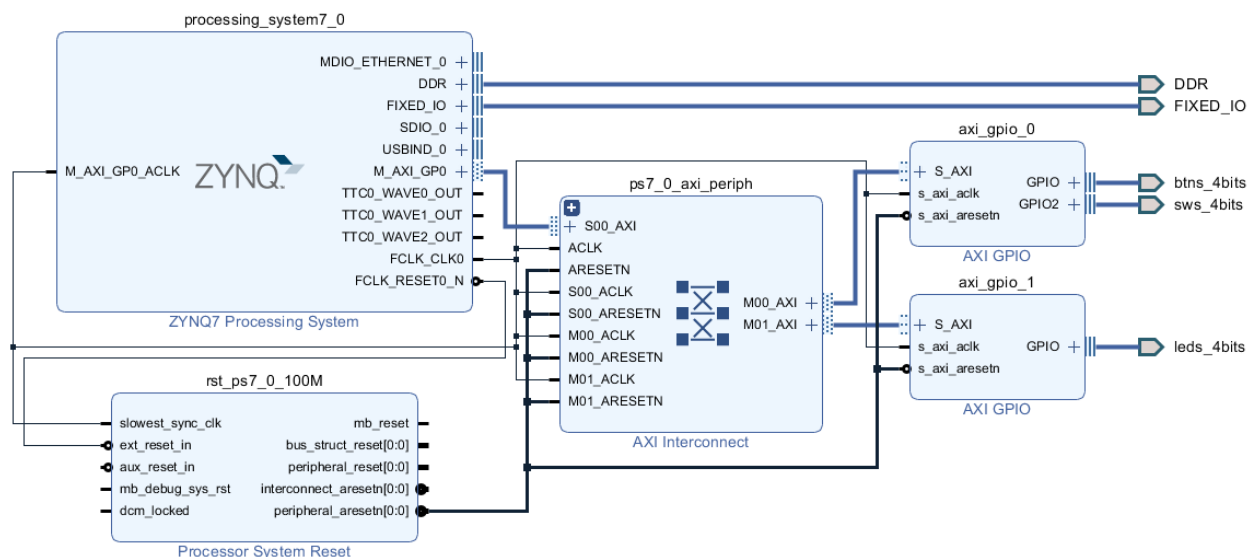
Dennis Silage, PhD
silage@temple.edu



Task Management in FreeRTOS on Zybo

In this Lab 2 you will investigate task management in the FreeRTOS environment on the Zybo board. In Lab 1 you created a Xilinx Vivado block design with axi_gpio_0 as a single channel for the Zybo push buttons and axi_gpio_1 as single channel for the Zybo LEDs as shown below.

In this Lab 2 you are modify that Vivado hardware design with axi_gpio_0 as dual channel for the Zybo push buttons on channel 0 and slide switches on channel 1, as shown below. This Vivado hardware design will be used in this and several Labs to follow.



The BTNs and SWs GPIO are at `XPAR_AXI_GPIO_0_DEVICE_ID` but the GPIO has two channels. The GPIO function calls for the BTNs and SWs must specify the correct channel number (0 or 1). The LEDs GPIO are at `XPAR_AXI_GPIO_1_DEVICE_ID`. The GPIO function calls for the LEDs must specify the correct channel number (0).

Launch SDK and search and add the `freertos_hello_world.c` application as described in the *Introduction to FreeRTOS on Zybo* posted on Canvas. You are to modify `freertos_hello_world.c` to be the new `main()` application (using another name) that performs the following FreeRTOS task management operations with the Vivado hardware design. Engineering projects are often stated as procedures which must be rendered as process control software on the appropriate embedded hardware.

1. Create a FreeRTOS task *TaskLED* at priority 1 that initially clears a 4-bit counter and displays the results in the LEDs. Within this FreeRTOS task if SW0 (slide switch 0) is OFF the counter and LED display increment with an appropriate delay. Within this FreeRTOS task if SW0 is ON the counter and LED display decrement with an appropriate delay. Within this FreeRTOS task SW1, SW2 and SW3 are ignored.

2. Create a FreeRTOS task *TaskBTN* that reads the BTNs at priority 1. Within this FreeRTOS task if BTN0 is depressed and no other BTN has been depressed first then *TaskLED* is suspended (*vTaskSuspend()*). Within this FreeRTOS task if BTN1 and no other BTN has been depressed first then *TaskLED* is resumed (*vTaskResume()*).

Within this FreeRTOS task if BTN2 is depressed and no other BTN has been depressed first then *TaskSW* (see below) is suspended (*vTaskSuspend()*). Within this FreeRTOS task if BTN3 and no other BTN has been depressed first then *TaskSW* is resumed (*vTaskResume()*).

Because of BTN *bounce* you should make sure the BTN is depressed and stable before executing the FreeRTOS task management function. You should research software approaches to push button *debounce*.

3. Create a FreeRTOS task *TaskSW* that reads the SWs at priority 1. Within this FreeRTOS task if SW1 is ON and SW2 and SW3 are OFF then *TaskBTN* is suspended (*vTaskSuspend()*). If SW2 is ON and SW1 and SW3 are OFF then *TaskBTN* is resumed (*vTaskResume()*).

If SW3 is ON and SW1 and SW2 are OFF then *TaskLED* is suspended (*vTaskSuspend()*). If SW3 is OFF and SW1 and SW2 are ON then *TaskLED* is resumed (*vTaskResume()*).

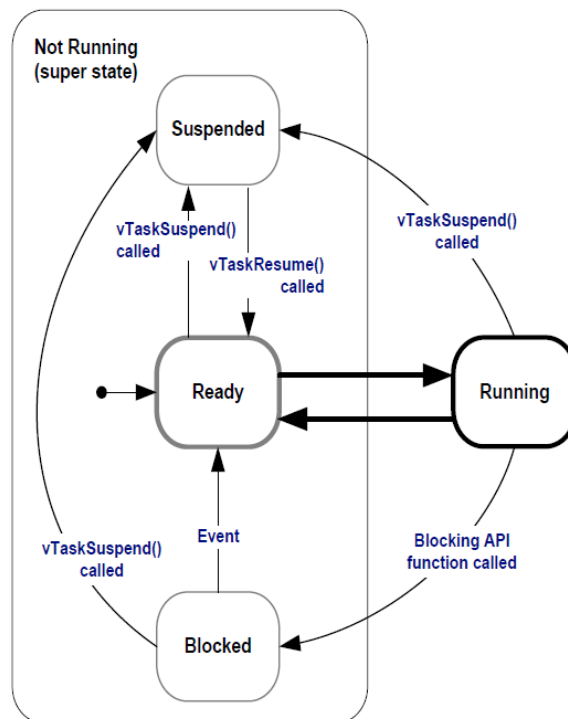
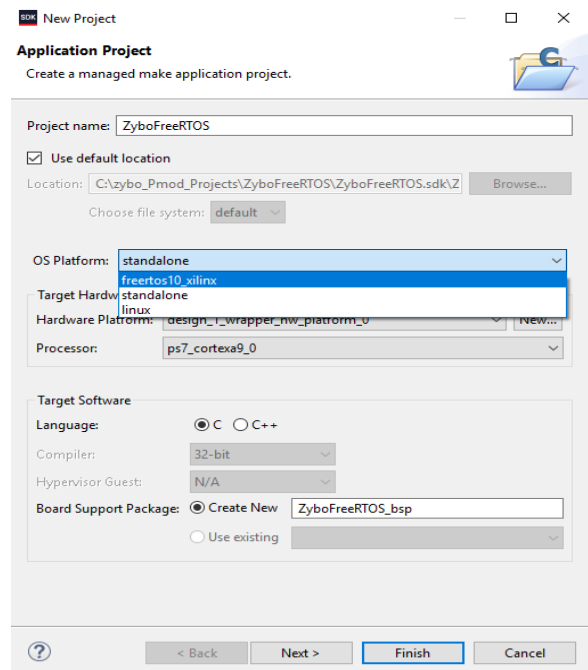
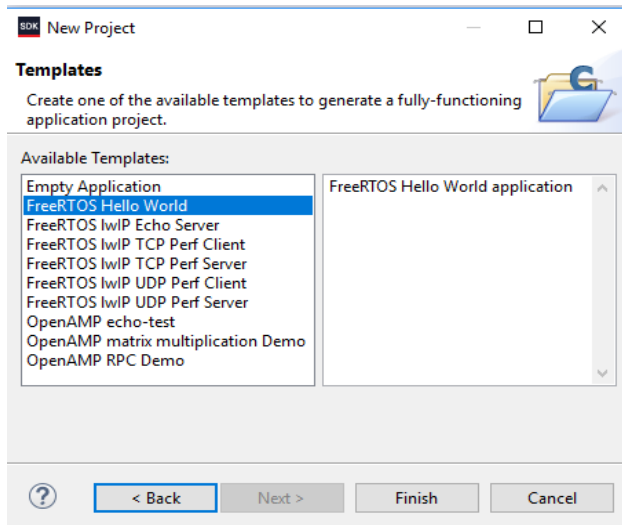
Within this FreeRTOS task SW0 is ignored.

You are to discuss and demonstrate *all possible conditions* of the task management interactions between the BTNs and SWs and the resulting LEDs. What actions determine the operation of the task management since all the tasks have the same priority?

Are there any unintentional conflicts in the specifications? This can often occur in engineering design. Can the conflicts be ameliorated by changing the specifications and modifying the control software?

The FreeRTOS the task management functions *vTaskSuspend()* and *vTaskResume()* utilizes the *handles* of the created tasks.

The completed Laboratories should be archived on your laptop and will form the basis of the Exams.



You are to use the *Project Report Format* posted on *Canvas*. You are to upload your *Report* to *Canvas* for time and date stamping to avoid a late penalty. This Laboratory is for the week starting Tuesday February 2nd and due no later than 11:59 PM Tuesday February 9th.

Spring 2021