

ELEC 2220 Computer Systems
Homework #18
Due: Wednesday, July 22 at 1:15 pm

PART 1 – STUDY QUESTIONS ABOUT INTERRUPTS (not for submission)

Referring to the in-class exercise posted for Wednesday, July 15, it is desired to change the Scanner operation from busy-wait to interrupt driven, with the DATA_VALID signal connected to pin PB2 to trigger interrupts as “External Interrupt Signal 2” (EXTI2) on our microcontroller. In preparation for doing this, look up the following information from the documents on the course web site.

1. In the Microcontroller Technical Reference Manual (Chapter 12 – “Interrupts and Events”)
 - a. What must be done to select pin PB2 as the source of the EXTI2 interrupt signal? (what register must be programmed?)
 - b. What has to be done to enable EXTI2 to trigger an interrupt? (What register must be programmed?) Note that there are 23 different sources of “external interrupts” that can be individually enabled or masked.
 - c. What has to be done to select the rising edge of EXTI2 to trigger the interrupt? (What register must be programmed?) This would allow the interrupt to be triggered when the scanner’s DATA_VALID signal goes from low to high.
 - d. Referring to Table 61 in this chapter, which external interrupt number (position in the interrupt vector table) corresponds to EXTI2?
 - e. In the startup code file, what is the name of the interrupt handler for the EXTI2 interrupt (look in the interrupt vector table.)
2. Chapter 11 of the textbook, the *STM32F4xx Cortex-M4 Programming Manual* (Chapter 4.3), and the *ARM Cortex-M4 User Guide* (Chapter 4.2) discuss the Nested Vectored Interrupt Controller (NVIC). (These two documents contain basically the same information.)
 - a. What NVIC register must be programmed to enable the EXTI2 signal to interrupt the CPU, what is the address of that register, and what bit of that register must be set?
 - b. What NVIC register is used to change the priority of the EXTI2 interrupt, what is the address of that register, and what bits of that register must be programmed? How many priority levels does our microcontroller support?

The above information will be the basis for the interrupt-driven programming project described on the next page – although the project will use the push-button on the board to trigger interrupts. Recall that this button is connected to pin PA0 (whereas the above refers to pin PB2).

PART 2 - NEXT PAGE

PART 2 – INTERRUPT-DRIVEN PROGRAMMING PROJECT (to be submitted)

Since the Final Project requires that the User Button generate interrupt requests, the purpose of this assignment is to practice setting up that button to trigger interrupts and to execute an interrupt handler. This will be done by modifying the design of Homework #17 as follows.

1. When the board first resets, perform any required initialization of variables, GPIO ports, etc.
2. Turn OFF all four LEDs and “do nothing” *until an interrupt is triggered by the user button*, and then go to Step 3.
3. Turn on one LED every half-second in a clockwise pattern as follows:
Green ON
Green-Orange ON
Green-Orange-Red ON
Green-Orange-Red-Blue ON (PD12-13-14-15)
All OFF (for a half-second)
Repeat this pattern *until the next user-button interrupt*, and then go to Step 4.
4. Turn on one LED every second in a counter-clockwise pattern as follows:
Blue ON
Blue-Red ON
Blue-Red-Orange ON
Blue-Red-Orange-Green ON (PD15-14-13-12)
All OFF (for one second)
Repeat this pattern *until the next user-button interrupt*, and then go back to Step 2.

Steps 2-4 are to be repeated continuously.

As with previous programs, partition the program into subroutines, with each subroutine performing a specific function (LED_ON, LED_OFF, DELAY, etc.).

Note that the previous “Check Button” function will need to be revised to make it an interrupt handler.

Also – the button may have to be “debounced” to ensure that there is only one action taken for each button press.

Finally, note that the questions in PART 1 refer to pin PB2, whereas the User Button is connected to PA0, so you will need to select the appropriate register options for PA0.

Deliverables:

1. Submit a printout of the source program. Ensure that the program includes descriptive “comments”.
2. Demonstrate the operation of the program via a Zoom meeting. Alternatively, you may upload to Canvas or email me a **video** that demonstrates the above behavior. I must see and hear you in the video, describing the operation of the board.