



# Module 14

Quiz: Real-time Systems

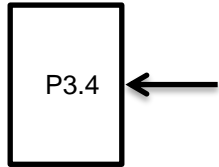


# Quiz: Real-time Systems

## Q1 Edge-triggered Interrupts

Write C code that counts the number of times an input on P3.4 goes from 1 to 0. You may assume the input does not bounce. Set the priority to level 4. You may assume this is the only interrupt on Port 3. However the other pins on Port 3 may be used for input or output. So, write friendly code. Show the initialization and the ISR

```
uint32_t Count; // number of falling edges
void Input_Init(void);
void PORT3_IRQHandler (void);
```



This results in **Count** becoming 2.

## Q2 Interrupts

List all the conditions that must be true for a Port 1 GPIO interrupt to be generated? Does it matter the order in which these conditions occur?

## Q3 Priority

Assume there are two interrupts running on the system. One interrupt on P1.6 and a second interrupt on P1.5. If the two requests occur at the same time, what happens?

- A) Both are serviced, but P1.5 goes before P1.6
- B) Both are serviced, but P1.6 goes before P1.5
- C) P1.5 is serviced, but P1.6 is lost
- D) P1.6 is serviced, but P1.5 is lost
- E) Both are lost
- F) None of the above

## Q4 Priority

Assume there are two interrupts running on the system. One interrupt on Port 1 and a second interrupt on Port 2. The Port 1 interrupt has priority 2 and the Port 2 interrupt has priority 7.

- a) What happens if the two requests occur at the same time?
- b) What happens if Port 1 occurs first, and while running the Port 1 ISR, the Port 2 is triggered?
- c) What happens if Port 2 occurs first, and while running the Port 2 ISR, the Port 1 is triggered?

## Q5 Acknowledgement

The following is one correct way to service two interrupts on the same port. If P6.2 is triggered, semaphore **SW1** is set. If P6.3 is triggered, semaphore **SW2** is set. If pin x has been triggered, **P6->IV** returns a number  $2^{(x+1)}$ , and automatically clears that one bit.

```
void PORT6_IRQHandler(void) {
    uint8_t status;
    status = P6->IV;
    if(status==0x06){ // check for P6.2
        SW1 = 1; // signal semaphore
        status = P6->IV;
    }
    if(status==0x08){
        SW2 = 1; // signal semaphore
    }
}

// Consider alternate solution, which does have a bug.
void PORT6_IRQHandler(void) {
    if(P6->IFG&0x04){ // check for P6.2
        P6->IFG &= ~0x04; // acknowledge, clear flag bit 2
        SW1 = 1; // signal semaphore
    }
    if(P6->IFG&0x08){ // check for P6.8
        P6->IFG &= ~0x08; // acknowledge, clear flag bit 3
        SW2 = 1; // signal semaphore
    }
}
```

This alternate solution works most of the time. However, very rarely an interrupt is lost (edge occurs but the semaphore is never set). Why? Explain the bug in this alternate solution

## IMPORTANT NOTICE FOR TI DESIGN INFORMATION AND RESOURCES

Texas Instruments Incorporated ("TI") technical, application or other design advice, services or information, including, but not limited to, reference designs and materials relating to evaluation modules, (collectively, "TI Resources") are intended to assist designers who are developing applications that incorporate TI products; by downloading, accessing or using any particular TI Resource in any way, you (individually or, if you are acting on behalf of a company, your company) agree to use it solely for this purpose and subject to the terms of this Notice.

TI's provision of TI Resources does not expand or otherwise alter TI's applicable published warranties or warranty disclaimers for TI products, and no additional obligations or liabilities arise from TI providing such TI Resources. TI reserves the right to make corrections, enhancements, improvements and other changes to its TI Resources.

You understand and agree that you remain responsible for using your independent analysis, evaluation and judgment in designing your applications and that you have full and exclusive responsibility to assure the safety of your applications and compliance of your applications (and of all TI products used in or for your applications) with all applicable regulations, laws and other applicable requirements. You represent that, with respect to your applications, you have all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. You agree that prior to using or distributing any applications that include TI products, you will thoroughly test such applications and the functionality of such TI products as used in such applications. TI has not conducted any testing other than that specifically described in the published documentation for a particular TI Resource.

You are authorized to use, copy and modify any individual TI Resource only in connection with the development of applications that include the TI product(s) identified in such TI Resource. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT OF TI OR ANY THIRD PARTY IS GRANTED HEREIN, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information regarding or referencing third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of TI Resources may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

TI RESOURCES ARE PROVIDED "AS IS" AND WITH ALL FAULTS. TI DISCLAIMS ALL OTHER WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, REGARDING TI RESOURCES OR USE THEREOF, INCLUDING BUT NOT LIMITED TO ACCURACY OR COMPLETENESS, TITLE, ANY EPIDEMIC FAILURE WARRANTY AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY YOU AGAINST ANY CLAIM, INCLUDING BUT NOT LIMITED TO ANY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON ANY COMBINATION OF PRODUCTS EVEN IF DESCRIBED IN TI RESOURCES OR OTHERWISE. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, DIRECT, SPECIAL, COLLATERAL, INDIRECT, PUNITIVE, INCIDENTAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES IN CONNECTION WITH OR ARISING OUT OF TI RESOURCES OR USE THEREOF, AND REGARDLESS OF WHETHER TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You agree to fully indemnify TI and its representatives against any damages, costs, losses, and/or liabilities arising out of your non-compliance with the terms and provisions of this Notice.

This Notice applies to TI Resources. Additional terms apply to the use and purchase of certain types of materials, TI products and services. These include; without limitation, TI's standard terms for semiconductor products (<http://www.ti.com/sc/docs/stdterms.htm>), [evaluation modules](#), and [samples](http://www.ti.com/sc/docs/sampterm.htm) (<http://www.ti.com/sc/docs/sampterm.htm>).

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265  
Copyright © 2018, Texas Instruments Incorporated