TI-RSLK MAX

Texas Instruments Robotics System Learning Kit

Texas Instruments

Module 14

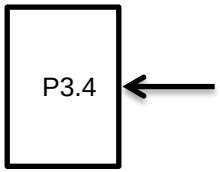
Quiz: I/O Triggered Interrupts

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 is 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

ti.com/rslk

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale (www.ti.com/legal/termsofsale.html) or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265