(https://www.edx.org)

UTAustinX: UT.6.01x Embedded Systems - Shape the World

KarenWest (/dashboard) ▼

Courseware (/courses/UTAustinX/UT.6.01x/1T2014/courseware) **Course Info (/courses/UTAustinX/UT.6.01x/1T2014/info)**

Discussion (/courses/UTAustinX/UT.6.01x/1T2014/discussion/forum) **Progress (/courses/UTAustinX/UT.6.01x/1T2014/progress)**

Questions (/courses/UTAustinX/UT.6.01x/1T2014/a3da417940af4ec49a9c02b3eae3460b/)

Syllabus (/courses/UTAustinX/UT.6.01x/1T2014/a827a8b3cc204927b6efaa49580170d1/)

Embedded Systems Community (/courses/UTAustinX/UT.6.01x/1T2014/e3df91316c544d3e8e21944fde3ed46c/)

Help

DEFINITIONS (4/4 points)

Please match the following terms with the letter of their appropriate definitions.

thread

C **Answer:** C

trigger

D **Answer:** D

context switch

A **Answer:** A

interface latency

B **Answer:** B

A. The steps that occur as control is passed from foreground to background or from background to foreground.

B. The time between when new input is available, and the time when the software reads the input data.

C. The path of action of software as it executes.

D. A hardware event that causes an interrupt.

EXPLANATION

As software executes we define a **thread** as the path of execution, or the action caused by executing software. In this class, the execution of the main program is one thread, and the execution of ISRs are other threads. Each time an interrupt is triggered we consider a new thread is created, and when the ISR finishes we define that thread as dead or deleted. One reason for considering each execution of an ISR as a new thread is because local variables cannot be used to pass data from one interrupt occurence to another.

Examples of hardware **triggers** in this chapter are the Count bit in SysTick and the RIS bits in the edge-trigger interrupts for each port pin.

A real-time system has a bounded **latency**. This means we can guarantee that new input will be recognized and processed within a finite time.

When an interrupt is triggered, the state of the main program is saved on the stack, IPSR is set, the PC is loaded with the vector address of the ISR, and the LR is set to 0xFFFFFFF9. This is called a **context switch**. Another context switch occurs when the ISR finishes and control is passed back to the main program.

Check Hide Answer

I/O SYNCHRONIZATION (1/1 point)

Which I/O synchronization method is best for real-time systems?

○ Blind

● Interrupt ✓

○ Busy-wait

○ None of the above

EXPLANATION

Interrupt synchronization is best for real-time systems because it can guarantee bounded latency.

Check Hide Answer

INTERRUPT I/O SYNCHRONIZATION (1/1 point)

What I/O condition should cause an interrupt when interfacing an input device?

- ● Busy to done, new input ready. ✓
- ○ Done to busy, I/O device is searching for new input.
- ○ As frequently as possible.
- ○ Periodically, at a fixed rate.
- ○ None of the above

EXPLANATION

Basically, we trigger an interrupt when there is some task the software must perform. For an input device, the software needs to input and process data once that data is available. We interrupt on the busy-to-done transition of the input device.

Help

Check Hide Answer

CONTEXT SWITCHING (1/1 point)

Which of the following steps does not occur when an interrupt is triggered, suspending the foreground and launching a thread in the background?

- ● The software executes BX LR. ✓
- ○ Eight registers are pushed on the stack.
- ○ The LR gets a special value of 0xFFFFFFF9.
- ○ The PC is loaded with the vector address of the corresponding ISR.
- ○ The IPSR is loaded with the interrupt number

EXPLANATION

The software executes BX LR at the end of the interrupt service. This instruction causes 8 registered to be popped off the stack returning back to the place the software was executing before the interrupt.

Check Hide Answer

OUTPUT DEVICE INTERFACING (1/1 point)

What I/O condition should cause an interrupt when interfacing an output device?

- ○ Periodically, at a fixed rate.
- ○ As frequently as possible.
- ○ Done to busy, I/O device is just begun outputting the next data.
- ○ The PC is loaded with the vector address of the corresponding ISR.
- ● Busy to done, the output device is now idle. ✓

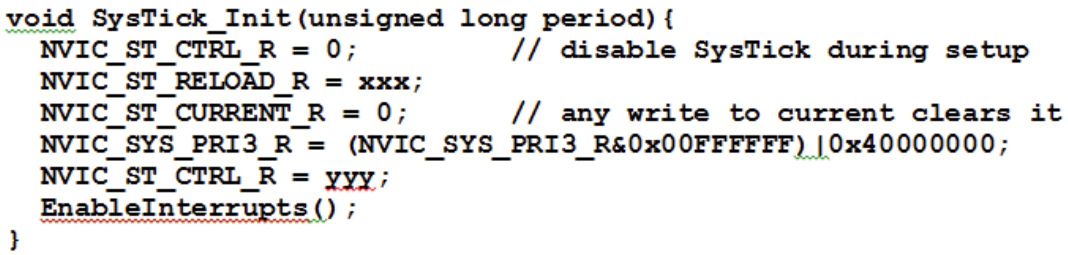
EXPLANATION

Basically, we trigger an interrupt when there is some task the software must perform. For an output device, the software needs to output when the output device is idle. We interrupt on the busy-to-done transition of the output device.

Check Hide Answer

SYSTICK INITIALIZATION (2/2 points)

Assume the bus clock is operating at 80 MHz. The SysTick initialization executes these instructions.



```
void SysTick_Init(unsigned long period){
  NVIC_ST_CTRL_R = 0;          // disable SysTick during setup
  NVIC_ST_RELOAD_R = xxx;
  NVIC_ST_CURRENT_R = 0;       // any write to current clears it
  NVIC_SYS_PRI3_R = (NVIC_SYS_PRI3_R&0x00FFFFFF)|0x40000000;
  NVIC_ST_CTRL_R = yyy;
  EnableInterrupts();
}
```

a) What value goes in the **xxx** place to make the interrupt frequency 10 kHz? Give your answer in decimal format.

7999

7999

Answer: 7999

b) What value goes in the **yyy** place to activate interrupts? Give your answer in decimal format.

7

7

Answer: 7

EXPLANATION

Interrupting 10,000 times per second means the interrupt period should be 100us. The interrupt period will be 12.5ns*(RELOAD+1). If RELOAD is 7999, then the interrupt period is 12.5ns*(7999+1) = 12.5ns*8000 =100,000ns = 100us

The SysTick clock source is 1, the SysTick interrupt arm is 1, and the SysTick counter enable is 1.

Check Hide Answer

Help

About (https://www.edx.org/about-us) Jobs (https://www.edx.org/jobs)
Press (https://www.edx.org/press) FAQ (https://www.edx.org/student-faq)
Contact (https://www.edx.org/contact)

EdX is a non-profit created by founding partners Harvard and MIT whose mission is to bring the best of higher education to students of all ages anywhere in the world, wherever there is Internet access. EdX's free online MOOCs are interactive and subjects include computer science, public health, and artificial intelligence.

(http://www.meetup.com/edX-Global-Community/)

(http://www.facebook.com/EdxOnline)

(https://twitter.com/edXOnline)

(https://plus.google.com/108235383044095082735/posts)

(http://youtube.com/user/edxonline)

Terms of Service and Honor Code -
Privacy Policy (https://www.edx.org/edx-privacy-policy)