


- [Courseware \(/courses/UTAustinX/UT.6.01x/1T2014/courseware\)](/courses/UTAustinX/UT.6.01x/1T2014/courseware)
[Course Info \(/courses/UTAustinX/UT.6.01x/1T2014/info\)](/courses/UTAustinX/UT.6.01x/1T2014/info)
- [Discussion \(/courses/UTAustinX/UT.6.01x/1T2014/discussion/forum\)](/courses/UTAustinX/UT.6.01x/1T2014/discussion/forum)
[Progress \(/courses/UTAustinX/UT.6.01x/1T2014/progress\)](/courses/UTAustinX/UT.6.01x/1T2014/progress)
- [Questions \(/courses/UTAustinX/UT.6.01x/1T2014/a3da417940af4ec49a9c02b3eae3460b/\)](/courses/UTAustinX/UT.6.01x/1T2014/a3da417940af4ec49a9c02b3eae3460b/)
- [Syllabus \(/courses/UTAustinX/UT.6.01x/1T2014/a827a8b3cc204927b6efaa49580170d1/\)](/courses/UTAustinX/UT.6.01x/1T2014/a827a8b3cc204927b6efaa49580170d1/)
- [Embedded Systems Community \(/courses/UTAustinX/UT.6.01x/1T2014/e3df91316c544d3e8e21944fde3ed46c/\)](/courses/UTAustinX/UT.6.01x/1T2014/e3df91316c544d3e8e21944fde3ed46c/)

An embedded system uses its input/output devices to interact with the external world. Input devices allow the computer to gather information, and output devices can display information. Output devices also allow the computer to manipulate its environment. The tight-coupling between the computer and external world distinguishes an embedded system from a regular computer system. The challenge is under most situations the software executes much faster than the hardware. E.g., it might take the software only 1 us to ask the hardware to clear the LCD, but the hardware might take 1 ms to complete the command. During this time, the software could execute tens of thousands of instructions. Therefore, the synchronization between the executing software and its external environment is critical for the success of an embedded system. This chapter presents general concepts about interrupts, and specific details for the Cortex™-M microcontroller. We will then use periodic interrupts to cause a software task to be executed on a periodic basis. If a GPIO pin is configured as an input, it can also be armed to invoke an interrupt on falling edges, rising edges or both falling and rising edges. Using interrupts allows the software to respond quickly to changes in the external environment.

Learning Objectives:

- Appreciate the need to perform multiple tasks concurrently.
- Understand real-time performance measures such as bandwidth and latency
- Learn how interrupts can be used to minimize latency.
- Study the basics of interrupt programming: arm, enable, trigger, vector, priority, acknowledge.
- Understand how to use SysTick to create periodic interrupts
- Use SysTick to create sounds and spin motors.



 <https://courses.edx.org/courses/UTAustinX/UT...>
(<https://plus.google.com/108235383044095082735/posts>)



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

© 2014 edX, some rights reserved.

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