

- [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/)

To determine the time, one simply reads the **NVIC_ST_CURRENT_R** register. Program 9.2 shows how to measure the elapsed time between calls to a function. Assume the system calls the function **Action()** over and over. The variable **Now** is the time (in 12.5ns units) when the function has been called. The variable **Last** is the time (also in 12.5ns units) when the function was called previously. To measure elapsed time between successive called to **Action()**, we perform a time subtraction. Since the SysTick counts down we subtract **Last - Now**. Since the time is only 24 bits and the software variables are 32 bits we "and" with 0x00FFFFFF to create a 24-bit difference.

```

unsigned long Now;        // 24-bit time at this call (12.5ns)
unsigned long Last;       // 24-bit time at previous call (12.5ns)
unsigned long Elapsed;    // 24-bit time between calls (12.5ns)
void Action(void){        // function under test
    Now = NVIC_ST_CURRENT_R;    // what time is it now?
    Elapsed = (Last-Now)&0x00FFFFFF; // 24-bit difference
    Last = Now;                // set up for next...
}

```

Program 9.2. Use of SysTick to measure elapsed time.

The first measurement will be wrong because there is no previous execution from which to measure. The system will be accurate as long as the elapsed time is less than 0.209 second. More precisely, as long as the elapsed time is less than $2^{24} \times 12.5\text{ns}$. This is similar to the problem of using an analog clock to measure elapsed time. For example you notice the clock says 10:00 when you go to sleep, and you notice it says 7:00 when you wake up. As long as you are sure you slept less than 12 hours, you are confident you slept for 9 hours.

Our TM4C123 microcontroller has some 32-bit and some 64-bit timers, but we will use SysTick because it is much simpler to configure. We just have to be aware that we are limited to 24 bits.

CHECKPOINT 9.4

If we activate the PLL and change the bus clock to 50 MHz (20ns), what is the longest elapsed time we could measure with Program 9.2?

Hide Answer

The longest time is $2^{24} \times 20\text{ns}$ is 335.5ms.



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>)

© 2014 edX, some rights reserved.

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