

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

## PREPARATION

You will need a LaunchPad, a slide potentiometer, a ruler, and an optional  $1\text{k}\Omega$  resistor. If you have the Nokia 5110 you may use it. The Nokia 5110 LCD is optional. This is a data sheet for one of the possible slide pots that can be used with Labs 14-15: **BournsPTASlidePotentiometer.pdf** (/c4x/UTAustinX/UT.6.01x/asset/BournsPTASlidePotentiometer.pdf). You should find a data sheet for the slide pot that you have.

## STARTER PROJECT

Lab14\_MeasurementOfDistance

## PURPOSE

This lab has these major objectives: 1) an introduction to sampling analog signals using the ADC interface; 2) the development of an ADC device driver; 3) learning data conversion and calibration techniques; 4) the use of fixed-point numbers, which are integers used to represent non-integer values; 5) the development of an interrupt-driven real-time sampling device driver; 6) the development of a software system involving multiple files; and 7) learn how to debug one module at a time.

## SYSTEM REQUIREMENTS

In this lab you will design a distance meter. A linear slide potentiometer converts distance into resistance ( $0 \leq R \leq 10\text{k}\Omega$ ). Your software will use the 12-bit ADC built into the microcontroller. The ADC will be sampled at 40 Hz using SysTick interrupts. You will write a C function that converts the ADC sample into distance, with units of 0.001 cm. That data stream will be passed from the ISR into the main program using a mailbox, and the main program will output the data on a display. The display is optional.

## DEMONSTRATION OF LAB 14



	0:00 / 1:18	1.0x			
--	-------------	------	--	--	--

PROFESSOR JONATHAN VALVANO: Hi.

Let me illustrate the components of lab 14.

We have the slide pot, which is a transducer.

The microcontroller and the optional display.

Onto the slide pot, I've glued this ruler.

And then around the armature, I've twisted this solid wire.

Such that the tip of this wire right here represents the cursor, or the true distance I'm trying to measure.

The transducer, the slide pot, will convert the position into a resistance.

The electrical circuit will convert resistance into voltage.

Right here is PE1.

And that's connected to the A/D converter,

which will convert voltage into a number.

The software will convert that number into a distance

and then display it here on this optional display.

Such that if I were to slide the slide pot, I will be able to measure the distance or the position of where the cursor is.

So again, we have a distance measurement system

and the transducer is the slide pot.

It uses the A/D converter and some software

to convert numbers into distances.

All right, that's how it works.

Now you do it.

Some suggested slide pots are listed on the kit buying page, <http://edx-org-utaustinx.s3.amazonaws.com/UT601x/worldwide.html> (<http://edx-org-utaustinx.s3.amazonaws.com/UT601x/worldwide.html>). Luckily, any potentiometer that converts distance to resistance can be used. The pots on the buying page can be plugged into the breadboard, whereas others may require you to solder or wrap wires onto the pins. Depending on the size of your pot and how you attach the wires, the full scale range of distance measurement may be anywhere from 1.5 to 10 cm. The pot used in the photos and videos measures distance from 0 to 2 cm. You will use an electrical circuit to convert resistance into voltage ( $V_{in}$ ). Since the potentiometer has three leads, one possible electrical circuit is shown in Figure 14.1. The default ADC channel is AIN1, which is on PE2. The TM4C123 ADC will convert voltage into a 12-bit digital number (0 to 4095). This ADC is a successive approximation device with a conversion time on the order of several  $\mu$ sec. Your software will calculate distance with a resolution of 0.001 cm. The position measurements could be displayed to the computer screen via UART0 using the **TExaSdisplay** interface. If you have a Nokia display, you can instead output the position measurements to it. A periodic interrupt will be used to establish the real-time sampling.

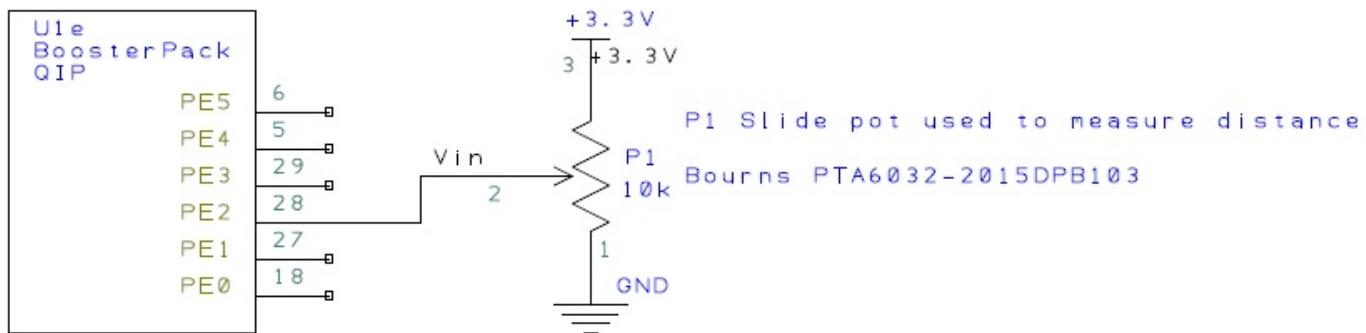


Figure 14.1. Possible electrical circuit to interface the distance transducer to the microcontroller. PE2 is ADC channel 1.

The left of Figure 14.2 shows a possible data flow graph of this system. Dividing the system into modules allows for concurrent development and eases the reuse of code. Each module will have a code file and a corresponding header file. The code file contains the actual implementation, and the header file has the prototypes for public functions. The SysTick initialization, SysTick ISR, and the main program will be in the main.c file. The ADC module will consist of the ADC.c and ADC.h files. The LCD module will consist of the Nokia5110.c and Nokia5110.h files. The right of Figure 14.2 shows a possible call graph.

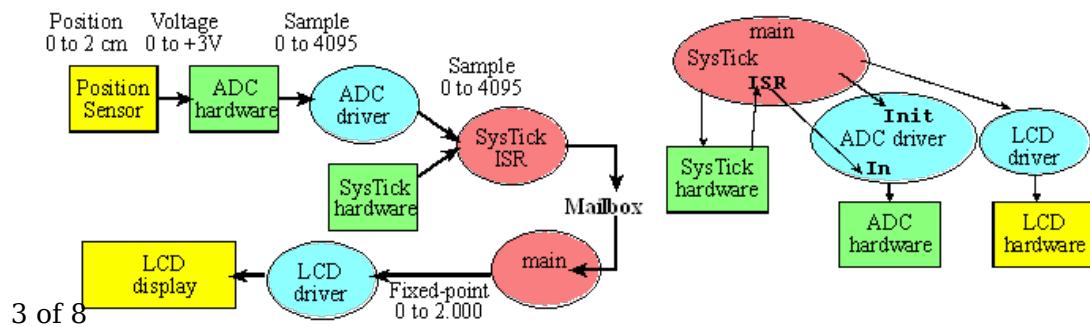


Figure 14.2. Data flow graph and call graph of the position meter system. Notice the hardware calls the ISR. If you do not have a Nokia 5110 LCD, you can use the UART and **TExaSdisplay**.

Help

You should make the distance resolution and accuracy as good as possible using the 12-bit ADC. The distance **resolution** is the smallest change in distance that your system can reliably detect. In other words, if the resolution were 0.01 cm, and the distance were to change from 1.00 to 1.01 cm, then your device would be able to recognize the change. Resolution will depend on the amount of electrical noise, the number of ADC bits, and the resolution of the output display software. Considering just the errors due to the 12-bit ADC, we expect the resolution to be about 2 cm/4096 or about 0.0005 cm.

**Accuracy** is defined as the absolute difference between the true position and the value measured by your device.

Accuracy is dependent on the same parameters as resolution, but in addition it is also dependent on the reproducibility of the transducer and the quality of the calibration procedure. Long-term drift, temperature dependence, and mechanical vibrations can also affect accuracy.

The **armature** is defined as the part that moves. In this lab, you will be measuring the position of the armature (the movable part) on the slide potentiometer, see Figure 14.3. Figure 14.4 shows a clear definition of “true” distance by using a ruler and a cursor.

Help

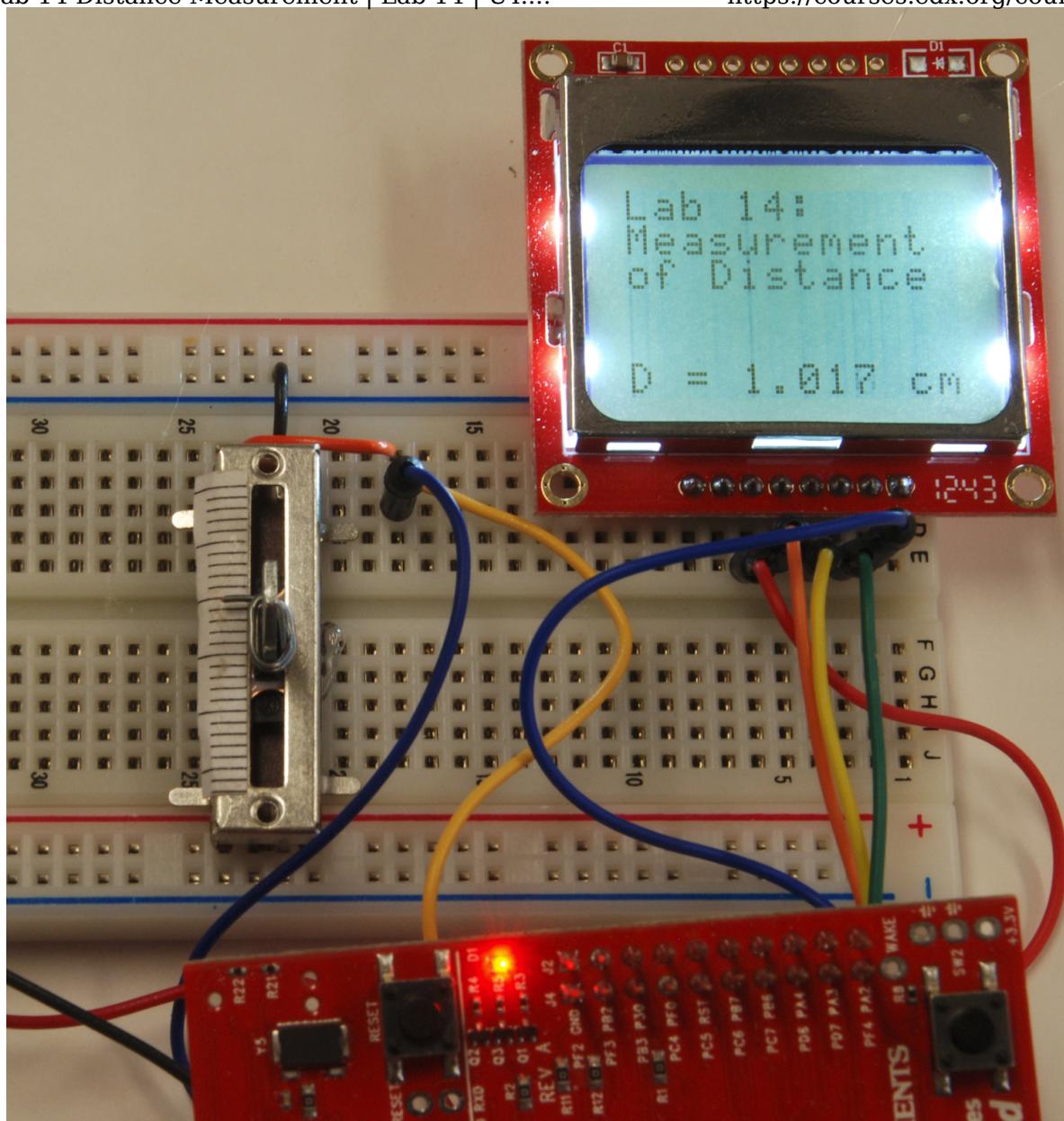


Figure 14.3. Hardware setup for Lab 14, showing the slide pot and optional Nokia. The slide pot is used to measure distance.

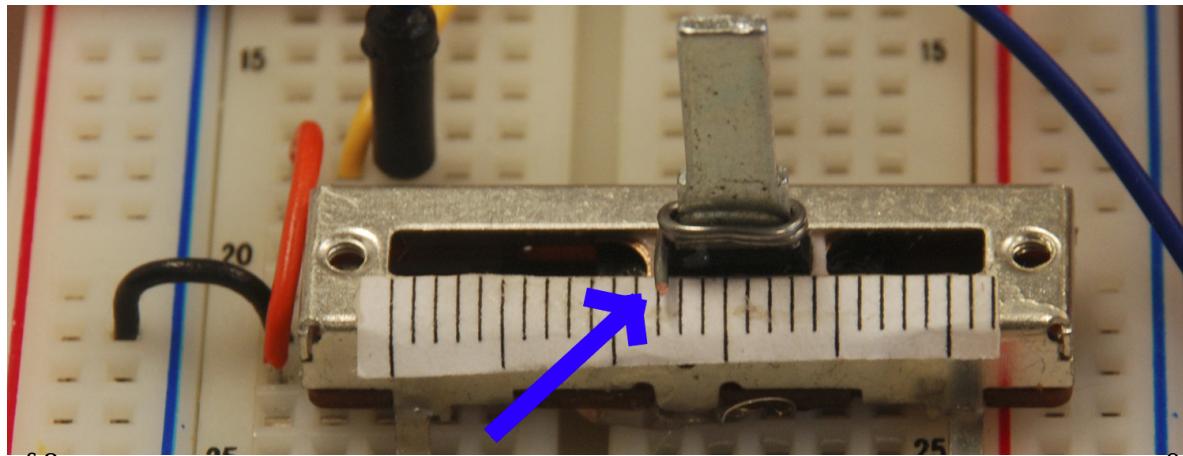


Figure 14.4. Close-up photo of the transducer. The tip of the cursor defines "truth".

Help

Due to the mass of the armature and the friction between the armature and the frame, the distance signal will exist only at a very low frequencies. One way to estimate the frequency components of the distance signal is to measure the maximum velocity at which you can move the armature. For example if you can move the armature 2 cm in 0.1sec, its velocity will be 20cm/sec. If we model the distance as a signal sine wave  $x(t)=1\text{cm}\sin(2\pi ft)$ , we calculate the maximum velocity of this sine wave to be  $1\text{cm}\cdot2\pi f$ . Therefore we estimate the maximum frequency using  $20\text{cm/sec} = 1\text{cm}\cdot2\pi f$ , to be 3 Hz. A simpler way to estimate maximum frequency is to attempt to oscillate it as fast as possible. For example, if we were able to oscillate it 10 times a second, then we would estimate the maximum frequency to be 10 Hz. According to the **Nyquist Theorem**, we need a sampling rate greater than 20 Hz. Consequently, you will create a system with a sampling rate of 40 Hz. A SysTick interrupt will be used to establish the real-time periodic sampling.

You will sample the ADC exactly every 0.025 sec and calculate distance using **decimal fixed-point** with  $\Delta$  of 0.001 cm. This means if the distance is 1.234 cm, we will store the integer 1234 in the computer. Conversely, if the integer in the computer is 567, it will mean the distance is 0.567 cm. If you do display the results, we suggest you include units. In general, when we design a system we choose a display resolution to match the expected measurement resolution. If we consider just the 12-bit ADC, the expected measurement resolution is  $2\text{cm}/4096=0.0005\text{ cm}$ . However we expect electrical noise to be the limiting factor in measurement resolution, and not the display or the ADC resolution. Furthermore, we expect uncertainty about exactly where the measurement point is to determine accuracy. Watch the Lab 14 demonstrant video again and notice the least significant digit flicker; this flickering is caused by electrical noise. You should expect your least significant digit to flicker as well. We made you display the thousandth digit just you can see that the ADC is not the limiting factor for resolution. In most data acquisition systems, noises in the electrical circuit and in the transducer are significant sources of error.

When a transducer is not linear, you could use a **piece-wise linear interpolation** to convert the ADC sample to distance. In this approach, there are two small tables Xtable and Ytable. The Xtable contains the ADC results and the Ytable contains the corresponding positions. The ADC sample is passed into the lookup function. This function first searches the Xtable for two adjacent of points that surround the current ADC sample. Next, the function uses linear interpolation to find the position that corresponds to the ADC sample. This is a very general approach and can be used for most applications.

A second approach to the conversion is to implement **Cubic Interpolation**. One description of Cubic Interpolation can be found in the following document online: <http://paulbourke.net/miscellaneous/interpolation/> (<http://paulbourke.net/miscellaneous/interpolation/>).

A third approach, shown in Figure 14.5, is use a **linear equation** to convert the ADC sample to distance, with  $\Delta$  of 0.001 cm. Since most slide potentiometers are linear, we will use this simple approach. Let **ADCdata** be the 12-bit ADC sample and let **Distance** be the distance in 0.001-cm units. You will find a linear equation, with slope and offset, to convert the ADC sample into distance.

$$\text{Distance} = (A \cdot \text{ADCdata}) \gg 10 + B \quad \text{where } A \text{ and } B \text{ are calibration constants}$$

## Calibration

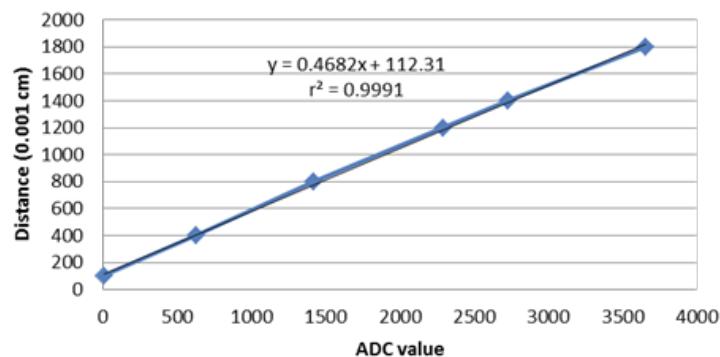


Figure 14.5. A linear equation used to convert ADC value to distance.

One measure of linearity is first fit a straight line to the Distance versus ADC data using linear regression. For more information on linear regression see [http://en.wikipedia.org/wiki/Simple\\_linear\\_regression](http://en.wikipedia.org/wiki/Simple_linear_regression) ([http://en.wikipedia.org/wiki/Simple\\_linear\\_regression](http://en.wikipedia.org/wiki/Simple_linear_regression)). The linearity then is defined how well the points in figure match the fitted straight line and given as the  $r^2$  value. An  $r^2$  value of +1 or -1 means the data is perfectly linear. The data in Figure 14.5 is pretty close to linear ( $r^2=0.9991$ ). The real board grader will measure linearity and require an  $r^2$  value above 0.96 to pass.

Furthermore we expect you to notice that calibration drift will be the limiting factor for accuracy. Calibration drift occurs when the constants A and B change over time.

## Specific Requirements

1. You must connect a slide pot to ADC channel 0, 1, or 8. You will need to define some place on the slide pot as "true" distance.
2. You must use ADC0, sequencer 3, and software start (because these are the only modes simulated by TExaS).
3. You must use SysTick interrupts to sample the ADC at 40 Hz. In particular, the ADC should be started exactly every 25 msec.
4. You must convert the ADC to distance, storing your measurement into the global variable **Distance**. For example, if the "true" distance is 1.2 cm, then your software should set **Distance** to 1200. The SysTick ISR will store the 32-bit calibrated distance in the variable **Distance** (called a mailbox) and set a flag. Reread Section 12.2 in the ebook to see how a mailbox can be used to pass data from the background into the foreground.
5. It is optional to output the results to the UART or to the Nokia LCD. However, if you do output you must perform the output in the main program and use a mailbox to pass the data from the ISR to the main. The main program will collect data from the mailbox and convert the distance with a  $\Delta$  of 0.001 cm into a string, using your **UART\_ConvertDistance()** function that you developed as part of Lab 11. The string can be display on the UART with **UART\_OutString** or on the Nokia with **Nokia\_OutString**. Include units on your display.
6. We expect the **accuracy** to be better than 0.02 cm in simulation, and better than 0.1 cm on the real board. Accuracy is defined as the difference between truth and measured. Only the simulation grader will check your accuracy.
7. We expect the **linearity** to be better  $r^2=0.96$ . Linearity is defined as the  $r^2$  value calculated when the distance versus ADC sample is fit to a straight line. Only the real board grader will check your linearity.



Help

[about](https://www.edx.org/about-us) (<https://www.edx.org/about-us>) [Jobs](https://www.edx.org/jobs) (<https://www.edx.org/jobs>)  
[Press](https://www.edx.org/press) (<https://www.edx.org/press>) [FAQ](https://www.edx.org/student-faq) (<https://www.edx.org/student-faq>)  
[Contact](https://www.edx.org/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>)