**《嵌入式系统及应用》实验报告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **年级、专业、班级** | | **2019级计算机科学与技术卓越** | | | **姓名** | **李燕琴 李放** |
| **实验题目** | **Measurement of Distance** | | | | | |
| **实验时间** | **2021.5.15** | | **实验地点** | **DS3 305** | | |
| **实验成绩** |  | | **实验性质** | **□验证性 √设计性 □综合性** | | |
| 教师评价：  □算法/实验过程正确； □源程序/实验内容提交 □程序结构/实验步骤合理；  □实验结果正确； □语法、语义正确； □报告规范；  其他：  评价教师签名： | | | | | | |
| 一、实验目的  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. | | | | | | |
| 二、实验项目内容  In this lab you will design a distance meter. A linear slide potentiometer converts distance into resistance (0 ≤ R ≤ 10 k). 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.  Some suggested slide pots are listed on the kit buying page, <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 (**Vin**). 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 usec. 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.*  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.  Figure 14.3 shows a possible call graph. A call grap    *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.*  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.    *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”.*  Due to the mass of the armature and the friction between the armature and the frame, the distance signal has a very low frequency response. One way to estimate the bandwidth 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)=1cm\*sin(2πft), we calculate the maximum velocity of this sine wave to be 1cm\*2πf. Therefore we estimate the maximum frequency using 20cm/sec = 1cm\*2πf, 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 Δ 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. However in this case, the expected measurement resolution is 0.0005 cm, but the display resolution is 0.001 cm. This means the display may be the limiting factor. However we expect electrical noise and uncertainty about exactly where the measurement point is to determine accuracy and not the display or the ADC resolution. Did you notice the least significant digit flickering in the video? 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 the noise and transducers 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/.  A third approach, shown in Figure 14.5, is use a linear equation to convert the ADC sample to position (Δ of 0.001 cm.) Since the transducer is 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.  **Distance** = (A\***ADCdata**)>>10 + B where A and B are calibration constants  *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>. The linearity then is defined how well the points in figure match the fitted straight line and given as the r2 value. An r2 value of 1 means the data is perfectly linear. The data in Figure 14.5 is pretty close to linear (r2=0.9991). The real board grader will measure linearity and require an r2 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. | | | | | | |
| 三、实验过程或算法（源程序）  在本实验中，我们需要实现一个测距器，通过滑动变阻器控制输入的电压，通过AD转换器将电压值转为数字信号，最后根据电压计算距离，并将距离输出在显示屏上。  AD转换器的初始化：    初始化ADC0，初始通道为1，初始序列为3.  具体情况如图所示    从AD转换器读取数据：    根据AD转换器的采样值计算距离，注意这里的距离计算公式可以根据校准结果调整    为了将结果输出到屏幕上，需要将计算结果转换成字符串，存储在名为String的全局字符数组中。    通过时钟中断，我们可以控制采样率，每隔一段时间进行一次采样    当采样完成后，会将全局变量flag置1，这样main函数需要等待flag为1的时刻，将距离数据（存储在全局变量中）输出到显示屏上，最后将flag置零并继续进行轮询。 | | | | | | |
| 四、实验结果及分析和（或）源程序调试过程  运行测试：      上板测试效果： | | | | | | |
| 五、实验心得  通过这次实验，我们小组成员学会了AD转换器的原理和使用方法，并使用AD转换器设计了一个简单的距离转换器。同时我们还学会了在LCD屏上进行显示，取得了不错的效果。 | | | | | | |