EECS 388 Laboratory Exercise #06
Analog-to-Digital Converter
Multiple Measurements
October 31, 2018 – November 6, 2018

1 Introduction

In this lab you will write a task to read the Analog-to-Digital Converter (ADC), capture a time sequence of measurements, and transmit the measurements via the UART to the PC.

2 Rationale

There are many sensor situations in which you need to capture a sequence of measurements. In this lab you will measure the voltage across a capacitor as it charges from 0.0 V to 3.3 V.

2.1 Program Modifications

Make as copy of the ADC project from Lab #05. You will add a task as described below. You should remove the task that reads the potentiometer voltage.

2.2 ADC Background

The operation of the Analog-to-Digital Converter was described in class. The steps to program the ADC and sample the analog input were discussed in class and a sample task provided. Additional information on the ADC hardware and DriverLib software are at:

http://www.ittc.ku.edu/~gminden/Embedded_Systems/PDFs/TI_ConnectedLaunchPad_UM_spmu365c.pdf,

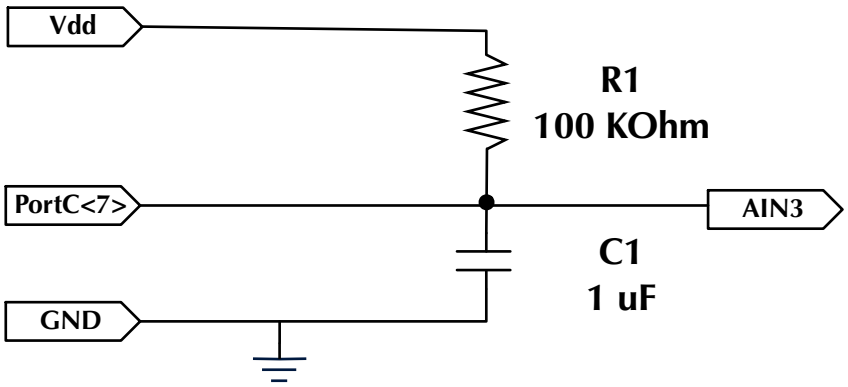
http://www.ittc.ku.edu/~gminden/Embedded_Systems/PDFs/TI_TM4C1294NCPDT.pdf, and

http://www.ittc.ku.edu/~gminden/Embedded_Systems/PDFs/TI_TIVA_DriverLib_UG-2.1.0.12573.pdf.

You should read those documents.

2.3 RC Circuit

The circuit you will be using is shown below.



You measure the voltage on the capacitor using AIN3. PortC<7> is used to discharge the capacitor.

3 New Tasks

Write a new FreeRTOS task. The task will do the following steps (states):

1. Initialize the required peripherals.
2. Wait for the Select button to be pushed and released.
3. Discharge the capacitor in the RC circuit by taking PortC<7> low and then high.
4. Collect 100 samples from the AIN3 using ADC0 as the capacitor charges.
5. Send the sample values to the PC workstation via the UART.
6. Go to step 2.

You will plot the collected voltage measurements and compared to expected voltages in your report.

3.1 Initialization Step

Initialized the required peripherals, e.g. the "User Switch 1" button, PortC, ADC0, and UARTstdio.

PortC<7> should be configured as an open drain output and the initial value of PortC<7> shall be '1'.

E.g.

```
GPIOPadConfigSet( GPIO_PORTC_BASE,
    GPIO_PIN_7, GPIO_STRENGTH_4MA, GPIO_PIN_TYPE_OD );
GPIOPinWrite( GPIO_PORTC_BASE, GPIO_PIN_7, 0x80 );
```

3.2 Wait for "User Switch 1" Step

Use techniques from EECS 388 Laboratory #03 to detect, de-bounce, and detect the release of the "User Switch 1".

3.3 Collection Step

Write a collection step that does the following:

1. Discharge the capacitor in the RC circuit.
2. Collect 100 samples from the ADC0<0> as the capacitor charges.

When the "User Switch 1" is released set PortC<7> to '0' for 0.5 mS and then set PortC<7> back to '1'.

Start collecting measurements.

Collect 100 ADC measurements using ADC0/AIN3 with a measurement interval of 1 mS. You should use **SysCtlDelay()** from DriverLib to delay between measurement samples. If you return to FreeRTOS with **vTaskDelay()** you will likely have jitter in your measurement intervals.

3.4 Transmit ADC Samples Step

When a measurement set is complete, the task should transmit the individual measurements via the UART to PuTTY.

3.5 Collect Samples in PuTTY

You should set up PuTTY to receive the measurement set and save the values to a file. You can use a text editor to select the samples, paste into Excel or other graphing program, and generate a plot of the sampled voltage.

3.6 Comparison of Measured Values to Expected Values

You should compare your measured voltage values to expected voltage values. The expected values should consider R = 100 KOhm, C = 1 uF, and Vdd = 3.3 V. The analytic formula for a charging capacitor is V(t) = Vdd * (1 – Exp(-t/(RC))).

Explain discrepancies between measured voltage and predicted voltage in your report.