Lab 9

Submission: January 21st, 2020

-------------------------------------------------------------------------------------------------------------------

In this lab you need to work on periodic ADC (analog to digital converter). Hence, you need to measure the room temperature with a temperature sensor and TM4C123 MC. You need to show the result in a 7-segment display. The PIN configuration and connection with TM4C123 and 7-SED was also discussed on last quarter. Please check Lab 7 to configure ADC of TM4C123 MC. Also, check lecture 8 to refresh ADC.

In Lab 9, you need to satisfy following requirements:

(a) ADC need to generate interrupt periodically in every 10-sec. Therefore, you need to check the temperature after every 10-sec. That temperature needs to be displayed in 7-SED.

(b) You need to use 2-Timers. Among them one for ADC (10-sec) and other for normal uses (1-ms).

(c) Timers should not initiate interrupt. However, ADC will initiate interrupt.

You can check this link:

<https://microcontrollerslab.com/adc-tm4c123g-tiva-c-launchpad-measure-analog-voltage-signal/>

The following are the steps to configure the ADC to sample a single channel at a periodic rate:

1. Enable the ADC clock in the **SYSCTL\_RCGCADC\_R** register.
2. Bits 3 – 0 of the **ADC0\_PC\_R** register specify the maximum sampling rate of the ADC. Set the maximum sampling rate at 125 kHz (**ADC0\_PC\_R** = 0x00). However, for 250 KHz (0x1), 500 KHz (0x2) and 1 MHz (0x3). 125 KHz will require less power and produce a longer sampling time, creating a more accurate conversion.
3. Set the priority of each of the four sequencers. Use just one sequencer, so the priorities are irrelevant, except for the fact that no two sequencers should have the same priority. The default configuration has Sample Sequencer 0 with the highest priority, and Sample Sequencer 3 as the lowest priority.
4. Configure the timer to run at the desired sampling frequency. Enable the Timer0 clock by setting bit 0 of the **SYSCTL\_RCGCTIMER\_R** register. First set bit 5 of the **TIMER0\_CTL\_R** register to activate TAOTE, which is the Timer A output trigger enable. Secondly, do not arm any Timer0 interrupts. The rate at which the timer rolls over determines the sampling frequency.
5. Before configuring the sequencer, we need to disable it. To disable sequencer 3, we write a 0 to bit 3 (ASEN3) in the **ADC0\_ACTSS\_R** register. Disabling the sequencer during programming prevents erroneous execution if a trigger event were to occur during the configuration process.
6. Configure the trigger event for the sample sequencer in the **ADC0\_EMUX\_R** register. For this example, we write a 0101(**ADC0\_EMUX\_R**  = 0x5) to bits 15–12 (EM3) specifying timer trigger mode for sequencer 3.
7. For each sample in the sample sequence, configure the corresponding input source in the ADC0\_SSMUXn register. In this example, we write the channel number (0, 1, 2, or 3) to bits 3–0 in the **ADC0\_SSMUX3\_R** register.
8. For each sample in the sample sequence, Configure the sample control bits in the corresponding nibble in the ADC0\_SSCTLn register. When programming the last nibble, ensure that the END bit is set. Failure to set the END bit causes unpredictable behavior. Sequencer 3 has only one sample, so we write a 0110 (=0x06) to the **ADC0\_SSCTL3\_R** register. Bit 3 is the TS0 bit, which we clear because we are not measuring temperature. Bit 2 is the IE0 bit, which we set because we want to request an interrupt when the sample is complete. Bit 1 is the END0 bit, which is set because this is the last (and only) sample in the sequence. Bit 0 is the D0 bit, which we clear because we do not wish to use differential mode.
9. If interrupts are to be used, write a 1 to the corresponding mask bit in the **ADC0\_IM\_R** register. We want an interrupt to occur when the conversion is complete (set bit 3, MASK3).
10. We enable the sample sequencer logic by writing a 1 to the corresponding ASENn. To enable sequencer 3, we write a 1 to bit 3 (ASEN3) in the **ADC0\_ACTSS\_R** register.
11. The priority of the ADC0 sequencer 3 interrupts are in bits 13–15 of the **NVIC\_PRI4\_R** register.
12. Since we are requesting interrupts, we need to enable interrupts in the NVIC. ADC sequencer 3 interrupts are enabled by setting bit 17 in the **NVIC\_EN0\_R** register.
13. Lastly, we must enable interrupts in the PRIMASK register.

The timer starts the conversion at a regular rate. Bit 3 (INR3) in the **ADC0\_RIS\_R** register will be set when the conversion is done. This bit is armed and enabled for interrupting, so conversion complete will trigger an interrupt. The IN3 bit in the **ADC0\_ISC\_R** register triggers the interrupt. The ISR acknowledges the interrupt by writing a 1 to bit 3 (IN3). The 12-bit result is read from the **ADC0\_SSFIFO3\_R** register.

See also: <https://microcontrollerslab.com/adc-tm4c123g-tiva-c-launchpad-measure-analog-voltage-signal/>