**Pre-Lab Requirements**

* Install Energia 1.6.10E18 as directed in the installation document.
* Make sure to install the drivers for the MSP432 before connecting your board to your computer or laptop.
* Run the Blink program from the examples, under the Files drop down menu to ensure that the LauchPad is communicating with your computer.

**What You Need**

* MSP432 TI LaunchPad with the Blink program installed (This is an individual requirement, not a group requirement so bring your own LaunchPad.)
* Laptop
* myPartsKit & Breadboard

**Analog to Digital Conversion on the MSP432**

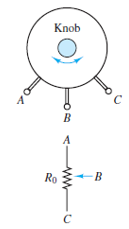
In this lab and the ones that follow, we will explore some of the capabilities of the TI MSP432 LaunchPad. In this lab, we will use the LaunchPad analog-to-digital converter (ADC) to begin learning how analog signals are converted to digital signals. We will explore this topic further in lecture. As electrical and computer engineers, it is important for us to understand the ADC process because all analog signals must be converted to digital if they are to be used in any type of digital application, including computers, communication via cell phones, or storage on digital media such as CDs and DVDs. The MSP432 line of microcontrollers from Texas Instruments features an 8-channel 10-bit ADC. This means that the MSP432 can perform analog to digital conversions on up to eight input signals, each with 10-bit resolution.

Figure 8.1a depicts a potentiometer, a three-terminal device, and its circuit symbol. Figure 8.1a - How a

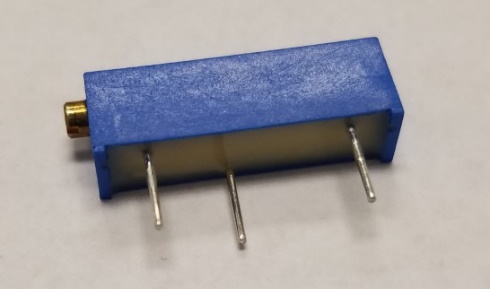
Figure 8.1b shows a picture of one of the potentiometers in your myParts Kit. A potentiometer works[4]. potentiometer has a fixed resistance R0, formed by a tightly wound coil of wire between Terminals A and C. Terminal B is connected to a wiper that slides along the coil as the knob is turned. The arrow in the circuit symbol represents the position of the slider along the length of the coil R0. The resistance from Terminal B to the other two terminals is determined by the wiper position. As RBA increases, RBC decreases - and vice versa - such that the sum RBA + RBC always equals R0.

Figure 8.1b – The myParts Kit potentiometer.

By connecting Terminal A to Vcc, and Terminal C to Gnd we create a voltage divider with two resistors - one between Termnals A and B, the other between Terminals B and C. By turning the knob, we change the values of the two resistors and thereby change the value of VB. When the shaft is turned all the way in one direction, the value of VB is 0 volts because there is no resistance between Terminals B and C. When the shaft is turned all the way in the opposite direction, the value of VB is equal to Vcc (approximately 3.3 volts) because all the resistance is between Terminals B and C. We will use the LaunchPad's ADC and the Energia command [analogRead](http://energia.nu/AnalogRead.html)() to read this changing voltage and convert it to a number between 0 and 1023 by connecting Terminal B to LaunchPad pin A0 (P1.0).

**Procedure**

1. Connect three wires from the potentiometer to the LaunchPad board as shown in Figure 8.2. The first wire goes to ground from one of the outer pins of the potentiometer. The second goes from Vcc to the other outer pin of the potentiometer. The third goes from analog input A3 (P5\_2) (Pin 12 on the board) to the middle pin of the potentiometer. You may connect it to any analog read pin by referencing to the datasheet.

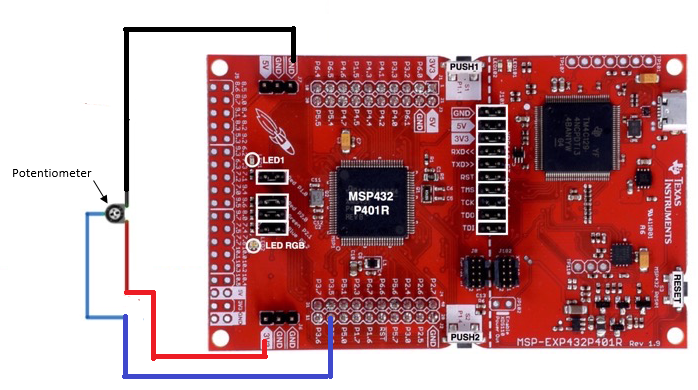
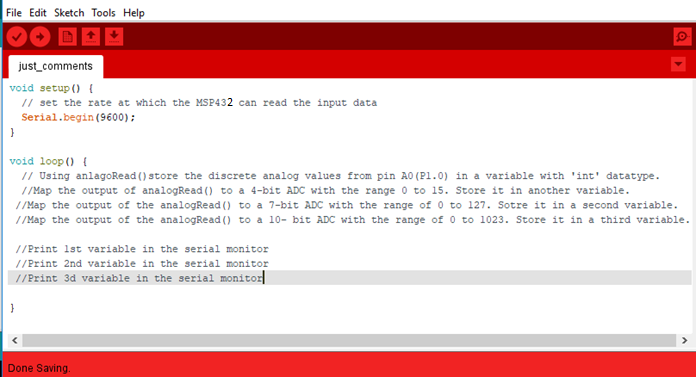
****

Figure 8.2 - Circuit connections for interfacing the LaunchPad to the potentiometer.

1. Figure 8.3 provides pseudocode to guide you as you write the five lines of code needed to activate your LaunchPad's ADC. Energia is case sensitive, so carefully follow the syntax of all the functions and built-in commands provided in the Energia website reference: <http://energia.nu/Reference_Index.html>

Figure 8.3 - PseudoCode showing the steps to follow for the ADC conversion and printing on the serial monitor.

1. In the Set-up section of the programming space, write the following line of code:

Serial.begin(9600);

This line will begin serial communication at 9600 bits of data per second between your LaunchPad and computer.

1. The remaining code will be written In the Loop section of the programming space. Write the first line of code using the **analogRead()** function. This function uses the ADC embedded in the MSP432 Microcontroller to convert the analog voltage values from potentiometer Terminal B to digital values. Because the MSP432's ADC uses 10-bits, it outputs values between 0 and 1023(1023 =210 - 1). Since an ADC produces only integer values, choose an [**int**](http://energia.nu/Int.html) datatype variable in which to store the digital value from the ADC**.**

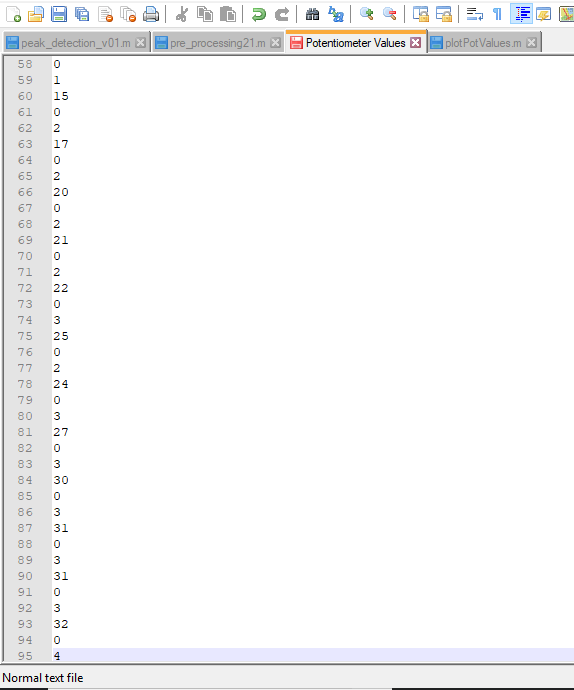
Store values from (P5\_2) using analogRead() into a variable of ‘int’ datatype;

1. Recall that the LaunchPad's ADC will convert the voltage at Terminal B using 10-bits. The **map function** is a built-in math function which re-maps a number from one range to another. We will use it to write the next three lines of code, which will produce equivalent 7-bit and 4-bit conversions from the Lau00000000000000000nchPad's 10-bit conversion. Store the new 7-bit and 4-bit values as ‘int’ datatypes also. Refer to <http://energia.nu/Map.html> for the syntax of the map function.

Map the analogRead() output to 3 different variables, one for 4-bit, one for 7-bit and one for 10-bit.

1. The last three lines of code are to print the digitized ADC output as 4-bit, 7-bit and 10-bit values on the serial monitor.

Serial.println(Your Variable Name);

1. Compile and run your code.
2. Debug it if necessary.
3. When it is running properly, the digitized potentiometer voltage will be displayed on the screen. Use the screwdriver in your myParts Kit to rotate the potentiometer knob to the position in which VB is 0v. You will know when you reach that point because your output will look like Figure 8.3. Note that every third number is a 0.

//4-bit conversion, sample 1

//7-bit conversion, sample 1

//10-bit conversion, sample 1

//4-bit conversion, sample 2

//7-bit conversion, sample 2

//10-bit conversion, sample 2

Figure 8.4 - Text file showing sample serial monitor values

1. Now rotate the potentiometer knob quickly toward the maximum VB. Remember that the maximum value of the 10-bit ADC is 1023. Remember that your code tells the monitor to print the 4-bit ADC, 7-bit ADC and 10-bit ADC in order, then repeat (loop).
2. Copy the serial monitor values to a text file. The first value must be a 0 (for the 4-bit ADC), the next value must be a 1 (for the 7-bit ADC) and the third value must be between 10 and 15 (for the 10-bit ADC).

Part 4 – Analyze the Data

1. Calculate the step size for each of your conversions:

10-bit step = Vcc/1024 = ­­­­\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

7-bit step = Vcc/128 = ­­­­\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4-bit step = Vcc/16 = ­­­­ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Use the values you captured from your ADC to plot all three of your conversions in a single MATLAB graph by following these steps:
2. Save your .txt in the same folder as your MATLAB code. Delete the 0s at the beginning of the file.
3. Write a .m script to open and read the text file line by line into a variable using the fopen() and fscan() functions.
4. Now plot the variable using the plot function. Remember that every third value in your file belongs to the same conversion. In other words, each ADC sample has been converted to a 4-bit sample, a 7-bit sample and a 10-bit sample and written in that order. Then the next sample has been written the same way. The syntax for reading and plotting every third value is as follows:

plot(A(1:3:end)\*(1023/X))

1. ‘A’ is the variable to which the .txt values were stored.
2. ‘1’ corresponds to the first value of the text file. To plot starting with the second value, replace, 1 with 2.
3. ‘3’ mean plot every third value.
4. ‘end’ means follow this pattern until the end of the text (data) file.
5. (1023/X) is the scaling factor where X = 2n - 1 for the n-bit conversion. This scaling factor will allow all three conversions to be plotted on the same scale so that you can compare the quality of the three analog to digital conversions.

Maximum of 10-bit values

Scaling Factor =

Maximum of n-bit values

1. Make each conversion plot a different color.
2. Label your X and Y axes and label your plot using the Matlab Legend function.
3. Your final graph should look similar to the one in Figure 8.5.

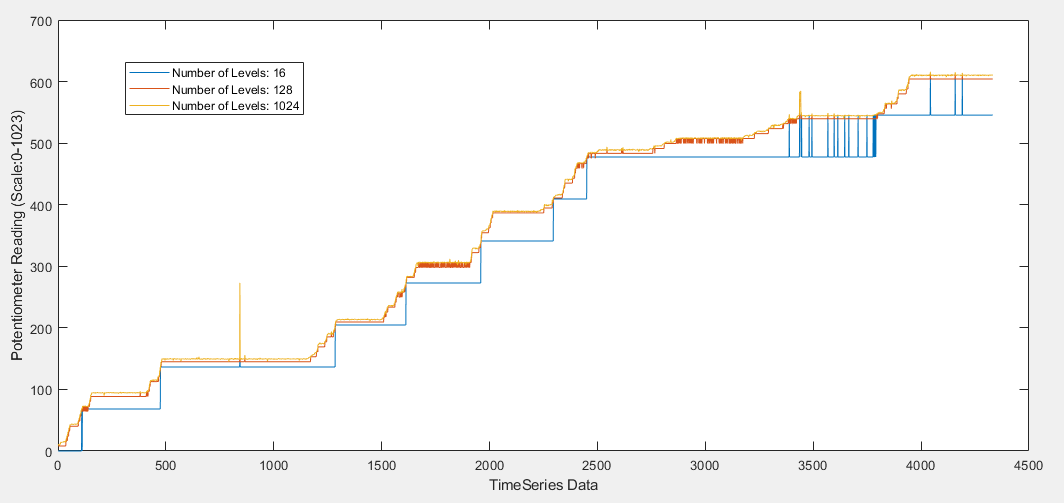


Fig 8.5 – MATLAB plot comparing 4-bit, 7-bit and 10-bit ADC quantized output values.

1. Answer the following questions. Discuss your code, your graph and your answers with your instructor.
2. Which conversion yields the smoothest curve? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
3. Which conversion yields the most blocky curve? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
4. The more bits we use in an ADC, the more expensive it is. Which of your curves looks like the best choice in a trade-off between a smooth curve and cost? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**References**

1. <http://energia.nu/Tutorial_ReadAnalogVoltage.html>
2. <http://energia.nu/Tutorial_AnalogInput.html>
3. <http://www.egr.msu.edu/classes/ece480/capstone/spring13/group04/application/Application%20note-karl.pdf>
4. Rizzoni, Principles & Applications of Electrical Engineering, Chapter-2 for more information on potentiometers.