ECE 218
EXERCISE 2. ANALOG TO DIGITAL CONVERSION

This exercise will introduce you to the PIC24 ADC and C header files. You will interface a voltage divider circuit, implemented with a potentiometer, to an analog input and run a program to read the value and convert it to a Binary Coded Decimal (BCD) value. You will gain more experience with the debugger as well.

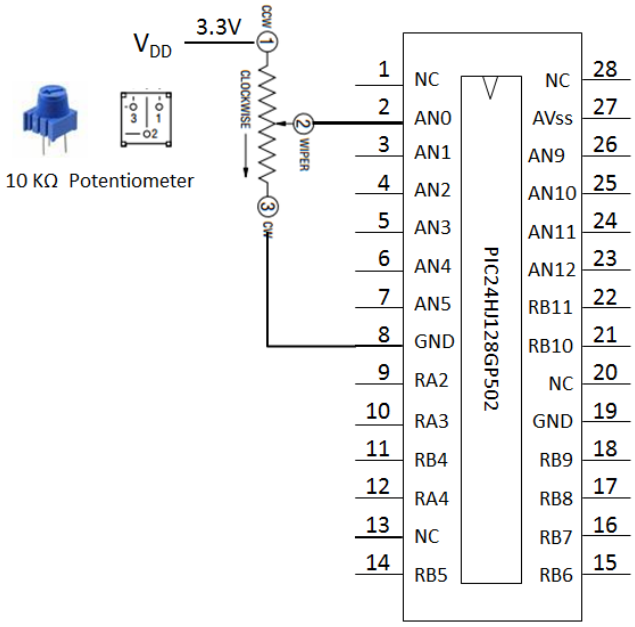


Figure 1. Voltage Divider Circuit Connected to PIC Analog Input 0

ANALYZE EXERCISE 2 PROJECT

Download the Exercise 2 project from Nexus. Next launch the MPLab X IDE and open the project from within the application. Since the project has been written in advance, you will not need to specify any of the configurations.

Notice in the Projects window that there are two source files and one header file in this project. **Review these files and explain how they are related.**

Which functions are declared in the adc.h header file? Where are these functions defined?

In the **adc.c** file find the function that initializes the ADC. **What value is written to the AD1PCFGL? Why?**

The **main_adc.c** main program calls the initialization function, *initADC*, and then has an infinite loop that converts the analog voltage on pin AN0 to a digital value (the *readADC* function) and then converts each 12-bit digital value to a 5 character array, where the characters are the corresponding decimal digits of the *adcvalue*.

Assume we run the program with a voltage of 1.0V at pin A0. What would we expect the *adcvalue* to be? What would the *count* array values be? You may need to look up the ASCII table online.

SET UP HARDWARE

We will apply a fixed voltage to pin AN1 (pin 3 of the PIC24 chip) using a 10K potentiometer connected as a voltage divider, as shown in Figure 1. We will use the Vdd (3.3V) and GND (0V) signals that are available on the Microstick II board. The GND signal is available from pin 8, and the 3.3V signal is available from the header on the Microstick II board that we soldered on.

Once the voltage divider circuit is constructed on the breadboard, measure the voltage of the wiper node to verify that it varies between the values 3.3V and 0. Remember that the maximum range of the ADC on the Microstick II board is GND(0V) – Vdd (3.3V).

Now connect the voltage divider circuit output (the wiper node) to the analog pin AN0 on the Microstick II.

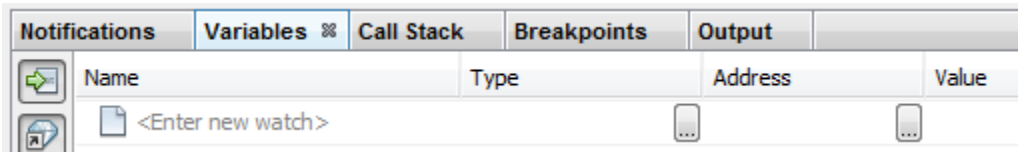
Note the value of the voltage on AN1. _____

RUN PROGRAM

Build the project and download to the board in **debug** mode.

Set a breakpoint at the *delay* function. (Click on the line number to toggle break point off and on).

Add the *count* and *adcvalue* variables to the watch list by going to the Variables tab in the status window, double-clicking on <Enter new watch> and selecting them.



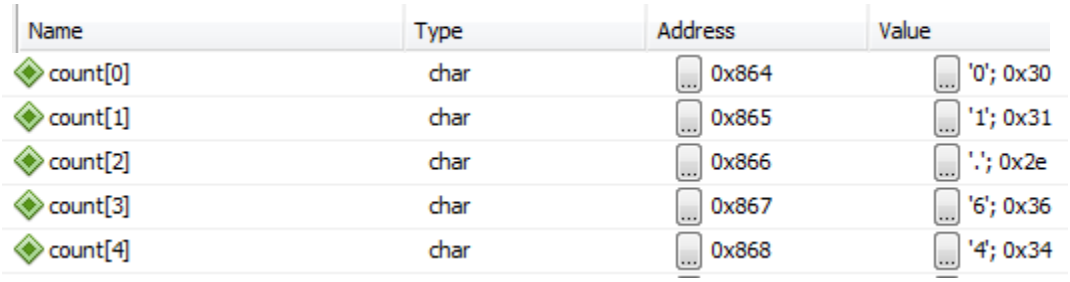
Run the code to the breakpoint and check the value of the *adcvalue* and *count* variables. **Does it match the expected value for your input?_____ If not, how much is it off by?_____**

Rotate the potentiometer and check that your count variable is changing as expected. **Write down the measured voltage and *count* values for 8 different potentiometer settings and graph these using Excel or Matlab to show the linear relationship.**

voltage	*count*	*voltage*	*count*

CHALLENGE

Consider if you wanted the count array to contain the characters for the actual voltage to two significant digits. For example, if the voltage was 1.64, the char array would contain the following ASCII codes:



What steps would be needed and where would these steps go in the main program? If there is time, try it!

Hint: To calculate the voltage from the *adcvalue*, you will need to declare a floating point variable. To compute integer values for each digit, you may need to force some floating point operations by specifying constants in floating point notation (like 10.0).