# Lab 1: Basic Analog and Digital I/O

*Instructor's Guide*

## Lab Introduction

This lab introduces basic concepts of digital and analog I/O. Future labs will make use of these concepts.

## Background Note

### Power System

None of the labs require a 5.0 volt power supply, unless the temperature sensor used in Lab 5 ($I^2C$) is substituted with a peripheral that requires 5.0 volt logic. Note that the temperature sensor breakout is capable of working with 3.3 and 5.0 volt logic, but not all devices work this way. A student may also want to use something that requires 5.0 volt logic for the final project, so you may want to consider having that option available.

## Instructor Review

The majority of the lab includes the exact procedure to complete it. The only portion that is left to the student to figure out is part of the firmware. So it would be best to evaluate the lab based on the following of directions, quality of the code, and proper behavior. Use the tests listed in Part 5 of the procedure to evaluate the behavior.

Here is a sample solution for the firmware.

```
void main()
{
    //The sum of up to eight ADC readings, used for averaging
    int32 accumulatedVoltage = 0;

    //The number of samples currently summed together in accumulatedVoltage
    uint8 numSamples = 0;

    //SNIPPED: Initialization code and displayStr found at the end of part 4

    //Loop forever
    for(;;)
    {
        //Wait for a sample to be ready
        ADC_IsEndConversion(ADC_WAIT_FOR_RESULT);

        //Read the sample, convert it to mV, and add it to the acc. voltage
        accumulatedVoltage += ADC_CountsTo_mVolts( ADC_GetResult16() );
```

```
        //Increment the number of samples read
        ++numSamples;

        //When 8 samples have been read, average and print to the display
        if(numSamples == 8)
        {
            //Right shifting 3 is the equivalent of dividing by 8, but faster
            accumulatedVoltage = accumulatedVoltage >> 3;

            //Convert the number to a string with leading spaces
            sprintf(displayStr, "%4ld", accumulatedVoltage);

            //Print the number to the display
            Display_Position(1, 4);
            Display_PrintString(displayStr);

            //Reset sample reading variables
            accumulatedVoltage = 0;
            numSamples = 0;
        }
    }
}
```