Using the ADC Exploration Board

The purpose of this instructional board is see how a computer does Analog to Digital Conversion (ADC) to convert an analog voltage (one of infinite possibilities) into a describe digital numerical value that in turn can be manipulated by a computer in computations. Details of ADC are contained in the supporting Power Point titled DSP Fundamentals (digital signal processing). ADC is the cornerstone of DSP (and frankly any handling of real world signals by computers today). The ADC Exploration Board simulates an 8-bit ADC circuit that is common in most programmable interface controllers (PICs) that are manufactured today.

The board allows the user to set an unknown voltage by means of a variable resistor adjustment that is attached to one input of a comparator integrated circuit. The other input to the comparator is the powering battery voltage which serves as the reference voltage upon which the conversion is based. The user then goes through the individual steps (just as is accomplished inside the computer ADC circuit) of switching in and out portions of a resistor ladder circuit starting with bit 7 (MSB) and ending in turn with bit 0 (LSB) using the output indicator of the comparator. The user records if the individual bits are high (1) or low (0) to build the byte that represents the numerical value of the unknown voltage relative to the reference voltage.

The following steps can be used as an example exercise of how an ADC works:

1. Measure and record the battery voltage, this is the reference voltage and will be close to 5 volts. (The 9 volts from the battery is stepped down and regulated to 5 volts.) For the purposes of this example we will assume the voltage is 5.02 volts.
2. Adjust and measure the unknown voltage by using the variable resistor. Record this voltage as the unknown voltage that will be measured (digitized). For the purposes of this example we will assume the voltage is 2.0 volts.
3. The resolution of an 8-bit ADC is 256. Calculate what the digital equivalent of 2.0 volts is and record the value for reference to the exercise outcome. Use the following formula to calculate the predicted ADC value:

$$\frac{Unknown\ Voltage}{Reference\ Voltage} * 256 = Predicted\ ADC\ Value$$

$$\frac{2.0}{5.02} * 256 = 101.9$$

   With rounding the expected ADC value would be 102.

4. Using the calculator that comes with your computer operating system, convert 101 decimal to binary. 101 = 0b01100110
5. Set the bit switches with bit 7 = 1 and all the other bits = 0. This represents the first guess of half way (128/256) minimum possible (0 volts) and maximum (5.02 volts).
6. Then turn on the power to the board. If the comparator output is high (indicated by the LED being illuminated) then the interim guess (128) is too high. You therefore recognize the guess is

too high and set the bit 7 switch to 0. If the guess is too low, the output of the comparator is low (indicated by the LED being off). In this case leave the switch for bit 7 set on (1). During the remaining switch manipulations remember if the switch is on (1) and the LED is also on, turn the switch off (0); if the switch is on (1) and the LED is off, leave the switch on (1).

7. Move on to bit 6 and turn the switch on. As noted above (which is the thought process that is being accomplished in the computer program), if the LED is on, turn the switch off; if the LED is off, leave the switch set. In this exercise example the LED should be off so leave the switch on (1).
8. Move on to bit 5 and turn the switch on. The LED should be off so leave the switch on (1).
9. Move on to bit 4 and turn the switch on. The LED should be on so turn the switch off (0).
10. Move on to bit 3 and turn the switch on. The LED should be on so turn the switch off (0).
11. Move on to bit 2 and turn the switch on. The LED should be off so leave the switch on (1).
12. Move on to bit 1 and turn the switch on. The LED should be on so turn the switch off (0).
13. Move on to bit 0 and turn the switch on. The LED should be off so leave the switch on (1).
14. Finally, record the individual bits to make the ADC byte 01100110.

Now it is very possible that your lower bits may not be exactly as illustrated here, this is because of rounding error and noise injected into the ADC process. There is a limit to the accuracy of ADC conversions, particularly in the lower bits which needs to be considered in the particular application. How most designer handle this situation is to use an ADC of higher bit resolution (10, 12, 14, 16 bits or more) and after the conversion is complete, throw out the lower bits to eliminate the noise and retain the high bits to reflect a more stable representation of the unknown voltage.

You can use the supporting Visual Basic program to simulate what is being accomplished by using the board during the exercise.