Assignment 3

Assigned: Tuesday 04/30/2019

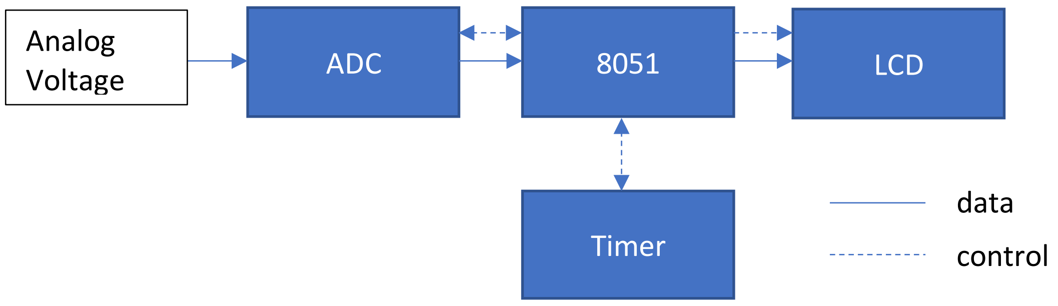
Due: Monday 05/13/2019, by 11:55PM

Turn in: Canvas

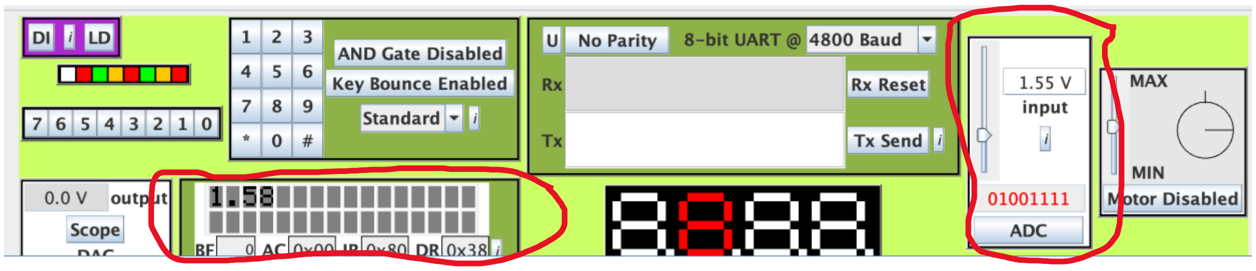
File to turn in: assignment3.zip

Note: make sure to add comments on the status of the code.

In this assignment, you are going to simulate a voltmeter using a 8051 microcontroller, Analog to digital converter (ADC), and a LCD in Edsim.



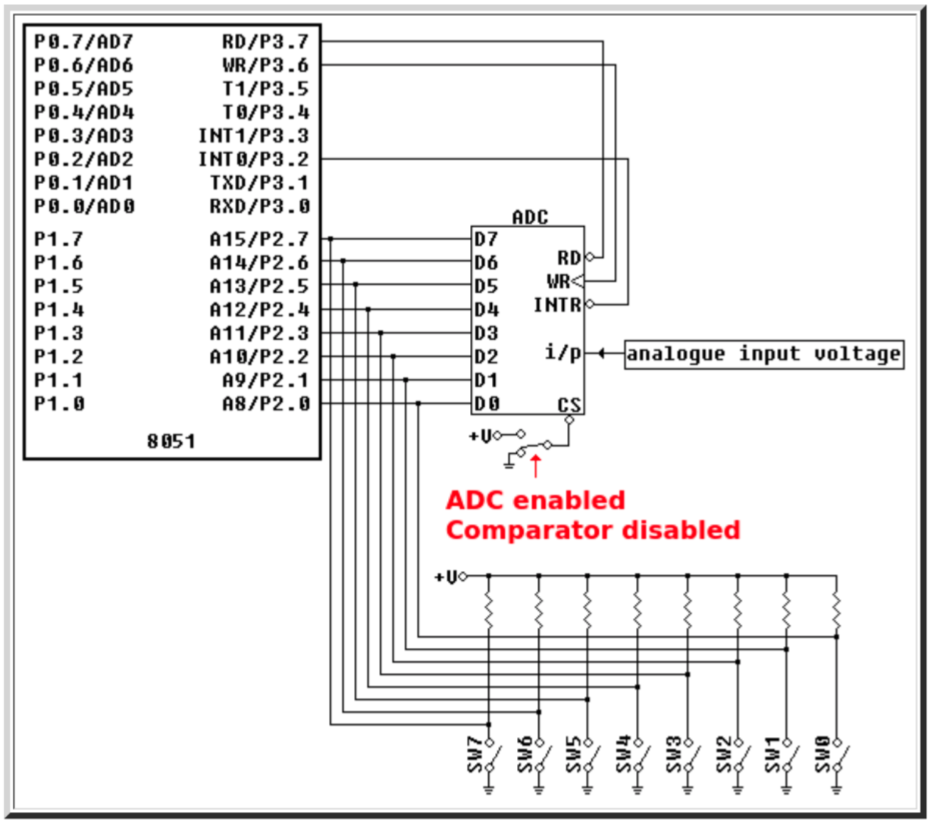
Voltage is going to be measured and converted to digital values by ADC, and the results are going to be read by 8051 ports, using interrupts, and displayed on LCD. See below for an example. Note that there is a slight error between the value on the ADC slider and the one displayed on the LCD, which is typical in voltmeters. This error however must be less than 5%.



ADC:

ADCs are used for data acquisition. Digital computers use binary values, physical world is analog. ADCs are required to convert real world physical quantities into digital numbers so that microcontrollers can read, process and store them.

The schematic below shows how the ADC may be interfaced with the 8051 in the Edsim51 simulator.



The function of the ADC pins are as follows:

- CS-bar is the chip select
    - To access the ADC, this pin must be low
- INTR-bar is the interrupt line - goes low when a conversion is complete
    - This is an output pin and when the conversion is finished, it goes low to signal
- RD-bar enables the data lines
    - This is an input pin and active low. This pin helps to get data out of the ADC
- WR-bar is cleared and then set to start a conversion
    - To inform the ADC to start the conversion process

In the assignment, the CS-bar line is connected to ground when the ADC is selected (and the comparator disabled in the peripheral panel (see the diagram above). The INTR-bar line goes LOW once a conversion is complete, therefore it is connected to one of the external interrupt pins on the 8051. In this way, the 8051 will be interrupted when a conversion is complete, and data is ready for reading.

The data lines are tri-state (hence the inverted triangle symbol) which means this chip can be memory mapped and the data lines can be connected directly to the data bus. In the assignment connect the data lines to port 2, since they are tri-state the port can also be used for something else. Only when the conversion is complete P3.7 is cleared which enables the data lines and the analogue conversion appears on port 2. You must also ensure that all the switches (SW0-SW7) are open so as not to force any P2 bits to 0.

The WR-bar line is used for starting a conversion. Clearing this line resets the internal successive-approximation register and the 8-bit shift register. When the line is set conversion begins.

Therefore, taking a reading from the ADC is a two-step process:

1. Clear and then set WR-bar to initiate a conversion.
2. Sometime later, the INTR-bar line will go LOW to indicate the conversion is complete. This will cause an external 0 interrupt and it is up to the external 0 ISR to read the data by clearing P3.7 and reading the data from port 2.

For this assignment, once the program completes initialization, it enters an endless loop. A timer interrupts every 10ms and the timer ISR initiates an ADC conversion. When the conversion is complete, the ADC causes an external interrupt. The external ISR for the ADC interrupt then updates the LCD accordingly.

LCD:

LCD should be connected to port P1 of 8051, and Set Interface data length to 8 bits, 2 line, 5x7 character font, also set the entry mode to increment with no shift. The display should be updated every time a new sample from the ADC is converted.

CONVERSION:

For conversion from the ADC's 8-bit conversion range of 0-255 to display 0.00 to 5.00 on the LCD, you can multiply the ADC read value by 2 to get the range from 0-255 to 0-510 (a.k.a. *range scaling*) which is approximately the same as the 0.00 to 5.00 voltage range (you just need to have the LCD panel display the decimal point after the most significant digit). This will result in a slight error (less than 5%) which is why the values shown in Edsim screen capture figure on page 1 differ slightly. The result of the conversion will be three decimal digits, which should be placed in registers R5, R6, and R7, so that the result displayed on the LCD should look like (R5).(R6)(R7) so R5 is the most significant digit then R6, then R7.

Note that the conversion of binary to decimal can occur either before or after the range scaling. If you convert before range scaling, you need to implement a decimal multiplication by 2 for three digits. If you convert after range scaling, you will end up with nine bits representation (to accommodate the 0-510 range). In this case, you have to figure out how to do division by 10 for a nine-bit value to generate the three decimal digits.

TESTING YOUR PROGRAM:

Run your program, then slide the ADC slider up and down. The LCD panel should display a voltage value that is close to the one shown next to the slider. Submit five screen shots of your program showing voltage values between 0-1, 1-2, 2-3, 3-4, and 4-5 volts. The code shown in the panel should have your name and ID in the comments at the beginning.