Benjamin Babtsov
02/25/2015
98322328

Basic Micro-Controller Applications (I/O, LCD Interface, A/D) –
Part 2 Micro-Controller ADC and LCD Implementation of Ohmmeter

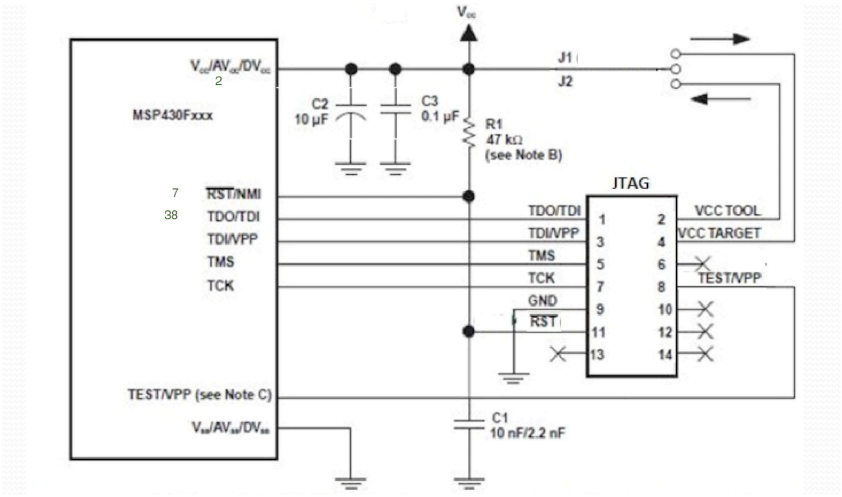
Introduction

This module required us to build a system that would measure resistance between two nodes and display the result on an LCD screen. The system was supposed to handle any resistance between the range of 1k ohm and 1 mega ohm (for any other resistance, the system would just display "out of range"), and be accurate within 5% of the actual resistance value. In order to implement such a system, the hardware we were expected to use included: a microcontroller, an LCD screen, wires and resistors.

Using hardware alone isn't enough, in order to complete the design successfully, one has to configure the microcontroller's analog to digital converter (ADC) to measure a particular voltage, then process that voltage and convert it to the appropriate resistance value. An entire port should be configured to interface with the LCD, and finally, software needs to be written to handle all the middleware between getting the signal from the ADC and outputting the right values to the LCD.

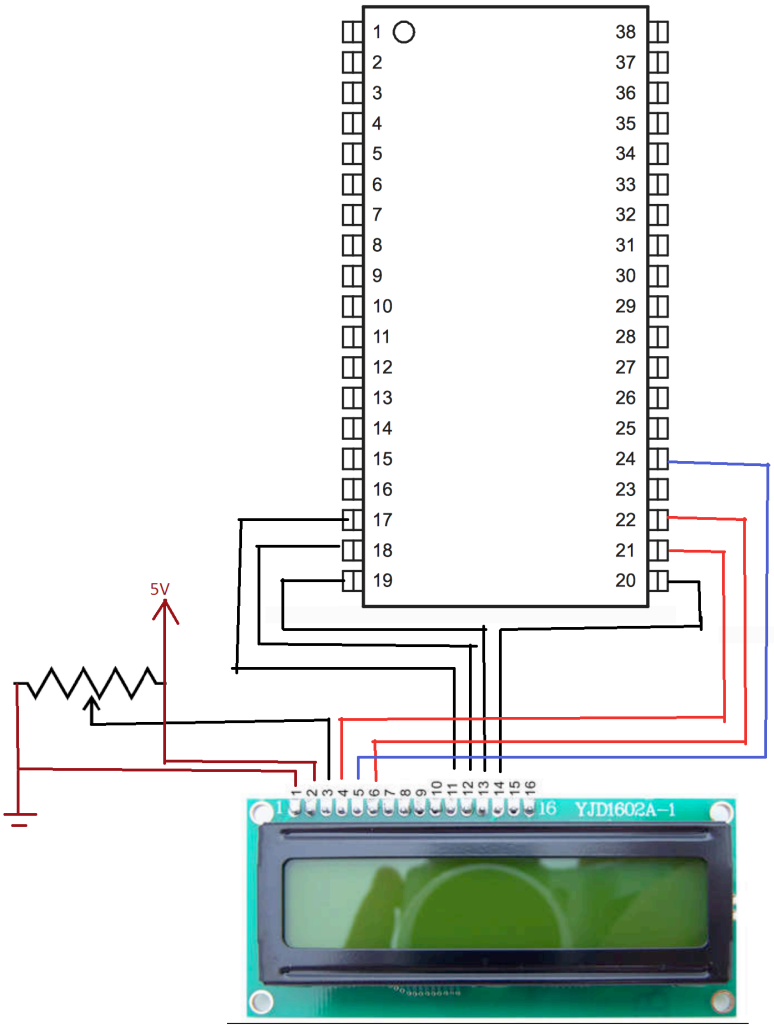
Design

The following wiring was used to connect the programmer to the microcontroller:



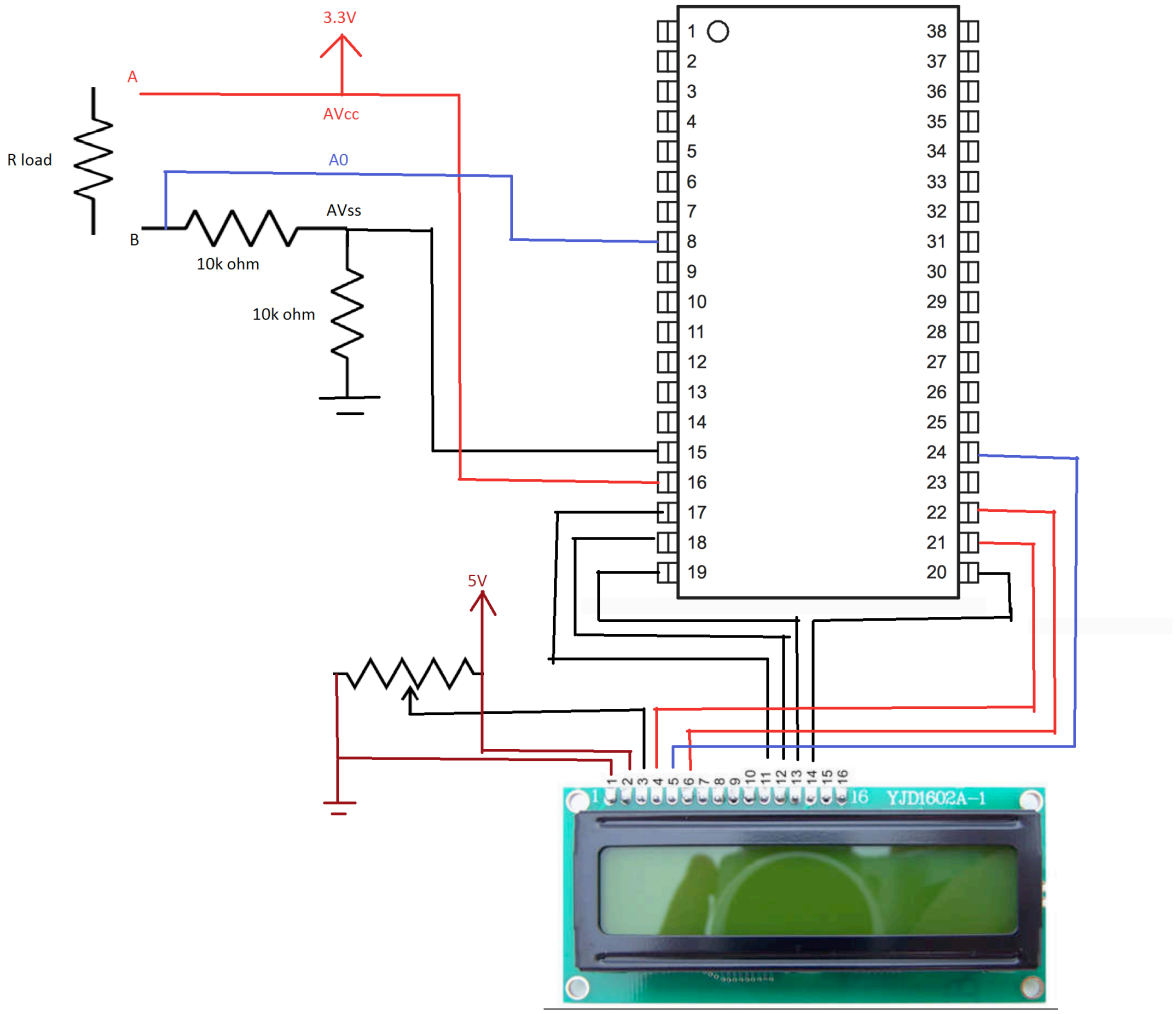
(Connection J1 was established in this case and 2.2nF was used for C1)

The next step in the design would be to establish an interface between the microcontroller and the LCD. Here is the wiring that was done to achieve that:



(As can be seen from the diagram, the 4-bit mode was used in this case, so only the upper nibble was connected while the lower nibble was left disconnected).

After it, the ADC was connected, as illustrated in the following diagram:



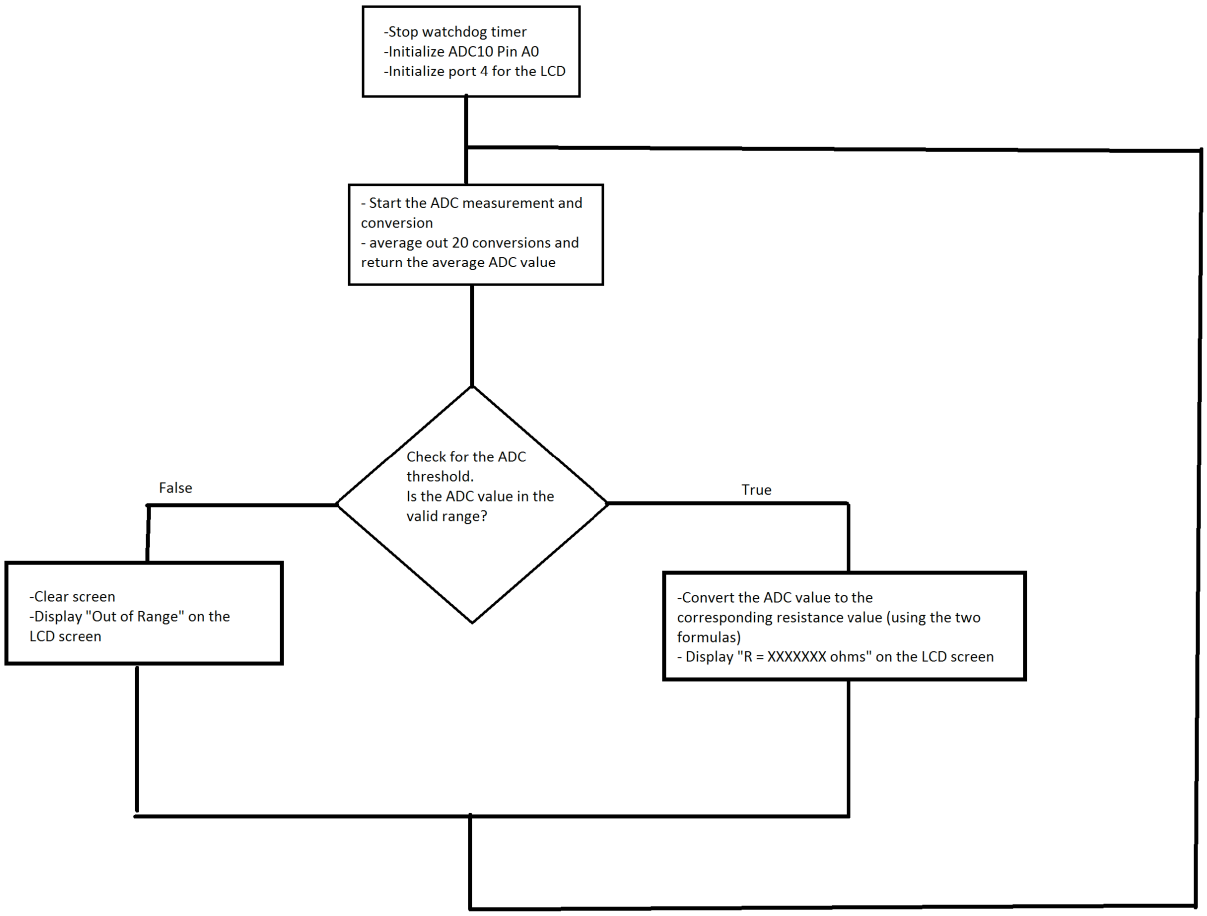
(I've tried many configurations for the ADC circuit; this particular one seemed to work well.)

As can be seen from the above diagram, the resistance between the A0 (ADC pin) and AVss (ADC ground) is 10k ohm. This value was chosen because a higher resistance would cause the ADC to deviate when no load is connected (open circuit). Also notice that the AVss node itself is pulled down to ground (via the second 10k resistor). This also was necessary to make the whole circuit work.

The following formulas were used to convert from the value read by the ADC to resistance:
Voltage = (ADC_reading / 1023)*3.3
Resistance = 1000*(33 / Voltage - 10)

The first formula describes the conversion from digital value between 0 to 1023 to the corresponding analog value between 0V and 3.3V. The second formula uses the concept of voltage divider and yields the resistance in terms of the analog voltage value that was measured by the ADC.

The following is the software outline:



Bill of materials:

Part	Cost
TI MSP430F2274 microcontroller	$5.22
LCD screen	$1.85
2 10K resistors	$0.06
Potentiometer	$0.30
TOTAL	$7.43

(The cost of the programming interface isn't included since it's not a necessary part of the final product)

While measuring the resistances after the system was built, I noticed that the possible resistance values appeared only as discrete values, yet they were pretty close to the actual resistance.

Conclusion

As a conclusion, I'd say that the design was a success. Of course the reason why the system can show only discrete values for the resistance is because the ADC itself is inherently discrete and can't make precise voltage measurements (information must be lost). Nevertheless, the system successfully managed to display the correct resistance over 3 decades (from 1k to 1 mega ohms). I major difficulty in this assignment was to figure out the appropriate circuit topology to connect to the ADC (pins A0, AVcc and AVss) in such a manner that the readings would be consistent throughout all the resistance range. At first I attempted to hook AVss straight to ground, and to have a big resistance between the A0 and Vss. Unfortunately, that configuration didn't seem to work for high resistances. Eventually, I decided to use the topology described above, and it proved to be a much better implementation.