

Experiment #10: Electronic Multi-Meter – A Voltage, Temperature, and Light Meter

ECE 367 – Microprocessor Design (Spring 2013)

PROFESSOR: Robert Becker

T.A.: Chenjie Tang

MWF – 10:00AM – 11:50PM

T Lab: 8:00AM – 10:50AM

Prepared by: Mitchell Hedditch

UIN: 677318273

Date Prepared: Monday, April 8th, 2013

Date Submitted: Tuesday, April 9th, 2013

I. Logic Diagram

II. Schematic Diagram

A. See attached sheet following this page.

- III. 9S12 Assembler Program
 - A. See next page

IV. USER MANUAL

A. Start Up

1. To start this system, use the USB to miniUSB cable and connect to computer. If you have a USB-outlet adapter, power may be supplied this way as well.
2. Verify the system is in "Run" mode (the switch on the microcontroller board).
3. Press the reset button to begin (left-most black button).

B. Operation

1. After pressing the reset button, the system directions will be displayed on the screen.
2. The system will start in the default modes of:
 - i. Binary A/D Display
 - ii. Temperature Mode
 - iii. Centigrade Scale
3. To change Mode press:
 - i. (A) for Temperature
 - The temperature meter will measure the surrounding air in the range of 0C-100C (32F-212F).
 - While in temperature mode, to toggle between centigrade and Fahrenheit, press the "E" button.
 - ii. (B) for voltage
 - To use the voltage meter, touch the orange wire (probe) to the voltage you would like to measure. It can measure any voltage within the 0V-5V range, with a precision of 0.2V.
 - iii. (C) for light -The light meter will read and display corresponding light values indicated by 5 different brightness categories:
 - Dark
 - Medium Low
 - Medium
 - Medium High
 - Bright
4. To change A/D display, press the "D" button to toggle continuously between binary, decimal, and hexadecimal output display.
5. You can change modes at any time, and the system will run until you turn it off or power is no longer available.

C. Shut Down

1. To shut the system down, disconnect the power source (USB cable) from the breadboard.

V. Conclusion.

A. How well does your project meet the specifications?

1. My project meets the full specifications according to ECE 367 Spring 2013 experiment 10.

B. What were the most difficult issues in realizing the system?

1. The temperature calculations were somewhat difficult to understand how to implement them and determine an accurate number with respect to normal temperatures.
2. The ADC0804 did give me some issues, but it turns out that I was using the wrong resistor values (1k instead of 10k). After I corrected those, it worked great.

C. Were you able to add extra features? If so, explain them.

1. No, this system only contains the standard features as set forth by the system requirements.

D. What would you have done differently if you were to do this project again?

1. I would have improved on the accuracy of the temperature gauge by increasing the number of bits read from the thermistor, therefore giving me a larger range over which to measure.
2. I would have reduced/reused code where available.

E. What did you learn from working on this project?

1. How to work with transducers/sensors.
2. How to use the onboard A/D converter.
3. How to use an external A/D converter
4. How to use a Parallel In/Serial Out microchip.
5. How to calculate the ranges for:
 - i. Thermistor
 - ii. Photocell
 - iii. Voltage measurement
6. How to properly implement transducers/sensors.
7. How successive approximation works.
8. It seems like the ADC0804 is more stable than the NanoCore12 A/D converter. This may have something to do with the fact that we can obtain improved accuracy from the NanoCore12 over the ADC0804.
9. I'm sure there is more, but it's way too early for me to remember. ☺