MICROCONTROLLER PROJECT LABORATORY

EDUCATIONAL STUDIES PROGRAM – HIGH SCHOOL STUDIES PROGRAM – SUMMER 2001
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

LABORATORY II
ADVANCED INPUT AND OUTPUT

GOALS

In this lab, students will:

1. Receive input from a keypad
2. Display text on an LCD
3. Experiment with a security system

SUPPLIES

Hardware:

1) Completed Lab I project
2) 2x16 LCD Display with backlight
3) 4x4 Matrix Keypad

Software:

4) rd2mon.asm
5) keytest.asm
6) lcdtest.asm
7) echo.asm
8) security.asm

Extra Credit:

9) Two additional transistors
10) One diode.

1.0 PROGRAMMING

From now on, we will use a program called RD2MON that I have adapted for this class. This program needs to be loaded once manually, and the boot vector must be set to 80. After that, all you need to program the chip is a serial connection.

A teacher will either give you a chip with RD2MON on it, or help you program yours.

1) Add a second wire to your right switch, so that when the switch is pressed, P2.7 (pin 28, row 13) is connected to +5 volts. This is necessary in order to run RD2MON when we are using port 2 in the future. **Check with your voltmeter.**

2) Use Windows Hyper Terminal to interact with RD2MON. To enter RD2MON, hold down the right switch while briefly pressing the left switch to generate a reset. You should see the message:
   Welcome to Microcontroller Project Laboratory 2001!
   To upload a new program, first do E0 for erase block 0. Then hit **D** for download, and from the transfer menu, choose "send text file" and pick the .obj file you are looking for (start with keytest.obj, generated by running **r keytest.asm -o**).
   That's it!
   From linux, you will need to open a connection to the serial port, then paste in the .obj file.

2.0 KEYPAD

1) Connect the keypad to Port 1 of the microcontroller, pins 1-8, with the bottom row closest to the microcontroller. You can either wedge it into the middle column of the breadboard if it fits, or connect 8 jumper wires to a more convenient location.

2) Make sure you have loaded **keytest.obj** as described in Part 1.
3) Do a normal reset (without the right button held down).
4) You should now see any keys you press on the keypad displayed on the computer, via the serial connection. Check all keys to make sure your keypad is not defective.

3.0 LCD

1) Connect the LCD module to the right of your breadboard, and connect its pins as follows.
(Be sure you know which is pin 1—look underneath and you should see a 1 and a 16. 1 should be closest to the corner).

Pin 1: Ground
Pin 2: Power
Pin 3: Contrast. Connect to the 3rd lead of a potentiometer, with the other two leads in 0 and +5.
Pin 4: Data select. Connect to P3.7 (pin 17 on the microcontroller)
Pin 5: Read/Write. Connect to P3.6 (pin 16 on the μC)
Pin 6: Enable. Connect to P3.5 (pin 15 on the μC)

Pins 7-14: Connect these 8 pins to port 3 on the microcontroller; the bottom-right most 8 pins. Be sure to connect them in increasing order.

Pin 15: LED power. Do not connect.
Pin 16: LED ground. Do not connect.

2) Program **lcdtest.obj** to your microcontroller, using RD2MON.
(You will need to go into programming mode. Be sure your right switch is configured to connect pin 28 to +5 volts when pressed, as well as connecting pin 30 to ground.)
3) Reset the chip into normal mode. It should display a timer on the LCD.

4.0 SECURITY

This program is modeled after home security systems. When the security system is on, it will detect a connection being broken, count down to 10, and set off an alarm. To deactivate the alarm or turn off the security system, you need a 4-digit code.

1) Pin 14 will be our alarm trigger. Connect a 10 KΩ resistor from pin 14 to ground. Then connect a long yellow "trigger" wire to +5 volts toward the lower left of the breadboard.
   The effect is that this pin will be held at 5 volts until the wire is disconnected. Then, it drops to 0 volts (ground). The 10K resistor prevents the waste of too much current.
2) Load **security.obj** into memory.
3) Run the program. It should say "Security on".
4) Test the alarm: Disconnect the trigger wire.
   It should count down from 10 to zero, then flash "alarm"
5) Enter the secret code to turn off the alarm. This is "1234" by default. Press "#" to try the code.
   You should see "security off"
6) Enter a new secret code of your choice. Hit "*" to program the new code.
7) Make sure the trigger wire is connected to +5V, then hit "#" again to turn the security back on.
8) Play around with the system until you are comfortable understanding how it works.

Congratulations! You can alarm your house!

5.0 EXTRA CREDIT

The rest of this lab is optional

Hardware required:

Speaker
3 NPN transistors (3904 variety will work)
Diode

1) Pin 13 generates an alarm tone. Try hooking it up through a transistor to a speaker as in Lab 1 and see if you can hear it. Find the location in the code that says "cpl SPEAKER".
2) Pin 12 generates a flashing light. First, hook this light up to a small LCD and see it flash. Then, connect this signal through two transistors in order to flash the LCD backlight LED:
   (These transistors are described with the flat side facing right.)
   a) The signal from pin 12 goes into the base (middle control lead) of the first transistor. This transistor gets power coming in the top lead to its collector.
   b) Connect the emitter output (bottom lead) of the first transistor into the base (middle lead) of a second transistor. This transistor has its emitter (bottom lead) connect to ground.
   c) Connect pin 16 of the LCD to the collector (top lead) of the second transistor, and connect a diode from +5 V to pin 15 of the LCD, with the dark edge toward the LCD.

   The diode drops the +5 volt supply down to about +4.2 volts (you can measure it) which is safe for the LCD backlight. The two transistors are needed because the LCD draws a huge current (100 mA), whereas the microcontroller supplies a tiny current. A single transistor could not multiple the microcontroller's current enough to light the light.

   Congratulations! You have a flashy alarm!