

ELEC 2220
Homework #16
Due: Wednesday, July 14

The purpose of this exercise is to practice setting up and using an interrupt service routine to process an external “event”. In this case, the “event” will be the pressing of a key on a keyboard.

One of the CodeWarrior “components” is the “IT_Keyboard”, described in class. You should also study the example in Chapter 18.2 of the text. This component simulates the operation of a standard matrix keyboard, with 5 rows of 4 keys per row (20 keys total). The exact configuration is shown in the CodeWarrior “help” documentation. The keypad must be connected to two I/O ports to implement a standard keyboard scanning algorithm. The IT_keyboard triggers an interrupt if any key is pressed while the corresponding row line or column is “active” (low).

Set up the IT_keyboard component with the columns driven by Port T, the row lines scanned (read) through Port AD, and with the keyboard triggering an IRQ interrupt (“vector” number 6 in the keyboard setup window). **For this exercise, only the keys labeled with decimal digits 0 through 9 are to be used.**

You will also need the seven-segment display from the previous project, connected to PORTA and PORTB as before, although this time we will use only the right-most digit, so “multiplexing” will not be necessary.

Finally, you will need an “IO_LED” component (a set of 8 LEDs), which is to be connected to PORTP.

The main program is to display the following repeating patterns on the 8 LEDs:

00000001-00000010-00000100-00001000-00010000-00100000-01000000-10000000-00000001....

There is to be approximately a half-second delay between changes, with the delay created by a software loop (or loops).

Whenever a key is pressed, the main program is to be interrupted, with the interrupt service routine (ISR) determining the number of the pressed key, displaying it on the right-most digit of the display, and then returning to the main program. That number should remain on the display until another key is pressed.

Note that the interrupt should not disturb what the main program is doing to the 8 LEDs, i.e. the cycling of the LED patterns should continue as if no interrupt had occurred.

Submit a printout of your source program and also email me the assembly language source file (main.asm), so that I can run your program on my PC. I will have my debugger configured to use the I/O ports specified above.