

LAB 6

OCx INTERRUPT

The purpose of this lab is to utilize a timer interrupt that is more easily adjustable and precise than the real time interrupt (RTI). In this exercise, this will be demonstrated by software.

Tasks:

1. Initialize the OCx interrupt of your choice
2. Prompt the user to enter a value in decimal that is in the range of 0 to $65,535 \approx 2^{16}$; only accepting valid entries.
3. Reserve five memory spaces and store each decimal digit into its respective memory.
Ex. User inputs 6. Acca = ascii \$36 store it into variable 1.
User inputs 5. Acca = ascii \$35 store it into variable 2.
User inputs 5. Acca = ascii \$35 store it into variable 3.
User inputs 3. Acca = ascii \$33 store it into variable 4.
User inputs 5. Acca = ascii \$35 store it into variable 5.
4. We now need to change this number to hex (ex: FFFF) and add it to TOCx. Use the least significant digits in var 1 to 5 for the conversion.
5. Convert the decimal number to hexadecimal and store it in memory location \$3010-\$3011.
6. Update the TOCx register with the newly converted number.
7. Connect an LED to PORTA to toggle at the frequency of TOCx.
8. Prompt the user for another input and repeat this process in a continuous loop.

Hints:

Use the MUL opcode to do the conversion. MUL multiplies A and B and stores the result in D. Thus, the numbers you multiply have to be 8-bits maximum (255). Here is one way to get around that:

$$65,535 = 6 * 10000 + 5 * 1000 + 5 * 100 + \dots$$

10000 and 1000 are more than 8-bits. Instead, do this:

$$65,535 = (6 * 40) * 250 + (5 * 4) * 250 + 5 * 100 + \dots$$

Experiment with this process and discover what happens when TOCx is updated with very small values. Remember to include flowcharts of all functions, and properly document your code. This should be an easy and interesting lab.