Birzeit University - Faculty of Engineering and Technology
Electrical & Computer Engineering Department - ENCS4330
Real-Time Applications & Embedded Systems - 1^{st} semester - 2023/2024

Project #3
16F877A PICMicro programming under MPLAB
Due: February 15, 2024

Instructor: Dr. Hanna Bullata

Simple calculator

We would like to build a simple calculator that is able to perform *only* multiplication mathematical operations on integer numbers. We'll assume each integer number will not exceed 2 digits (e.g. maximum number is 99).

The hardware should be composed of 2 16F877A microcontrollers in addition to a push button P to enter the numbers and a 16×2 character LCD. We'll call the first microcontroller the `master CPU` and the second microcontroller the `auxiliary CPU` or `co-processor`. The push button and the character LCD are connected to the master CPU.

The system should behave as follows:

1. When the system is powered up, the first line of the LCD should display the message ``Welcome to'' and the second line of the LCD should display the message ``multiplication''.

   The above message blinks 3 times with a 1 second delay between blinks. After 3 seconds, we move to the next step.

2. The first LCD line displays the message ``Number 1''. The cursor should then be positioned on the second LCD row. On the second LCD line, when we click on the push button P, the tenth part of the first number should increment by 1. Since originally the tenth part is 0, the first click should make it 1, the second click should make it 2, the third click should make it 3, etc until it reaches the value 9. On the tenth click, the tenth number goes back to 0, and so on.

3. If we leave P unclicked for over 2 seconds, the tenth part of the first number becomes fixed. Next, the unit part of the first number will be increased when we click the push button P from 0 to 9 with every click and then back to 0, and so on.

4. After entering the unit value, if we leave P unclicked for over 2 seconds, the first number becomes fixed. The LCD screen should display the multiplication symbol ``X''.

5. The LCD screen should display on the first line the message ``Number 2''.

6. Next, we need to proceed to entering the second integer number. That happens in the same order we entered the first number (e.g. the tenth part first, then the unit part).

7. Once the second integer number has been entered, the system should put on the LCD screen the sign ``='' and then execute the mathematical operation as follows:

   - The master CPU will send the first number (tenth and unit parts) to the co-processor in 1 shot.
   - Afterwards, the master CPU will send the unit digit of the second number to the co-processor.
   - The co-processor will mutiply number 1 by the unit digit of number 2 (make sure there is no overflow when doing that operation).

- The co-processor will return the result of the multiplication operation (3 digits) to the master CPU. That will happen in 2 steps: First the co-processor will return the 2 most significant digits in 1 shot and then send the least significant digit afterwards.
- During the above time, the master CPU will multiply the first number with the tenth digit of the second number.
- Once the master CPU gets the output from the co-processor, it should do the correct addition of the multiplication operation it did with the result it received from the co-processor and output the correct multiplication result of the 2 numbers on the second row of the LCD screen. At the same time, the first LCD line displays the message ``Result''.

8. The result of the multiplication operation stays on the LCD screen until push button P is clicked again. Once P is clicked, go to step **2** above.

What you should do

- Use proteus application to build the schematic for the whole system, including the master CPU, the co-processor, the 16×2 character LCD and the push button P.
- Use PortC of the master CPU and the co-processor for data transfer back and forth between both processors.
- Use pin PortB.0 of the master CPU for the push button.
- Use PortD of the master CPU to connect the character LCD in 4-bit mode. Remember to pull up the RS pin of the LCD using a 4.7KΩ resistor.
- Remember to add a 10KΩ pull-up resistor to push button P (or use the internal weak pull-up resistor of PortB), add a 4MHZ oscillator (with 2×15pF capacitors) and a 10KΩ pull-up resistor to the MCLR pin.
- Enable interrupts on push button P if you decide to use interrupts.
- Build the PIC assembly code for both the master CPU and the co-processor that implement the behavior described above under MPLAB IDE. You will have thus to build 2 projects, one for each processor.
- Assemble both projects and make sure you get a successful build on both. Use the simulator if you wish to make sure the behavior is correct.
- Send the zipped folder that contains the MPLAB code for both projects before the deadline. If the deadline is reached and you are still having problems with your code, just send it as is!