Birzeit University - Faculty of Engineering and Technology
Electrical & Computer Engineering Department - ENCS4330
Real-Time Applications & Embedded Systems - 2^{nd} semester - 2022/23

Project #4
16F877A PICMicro programming under MPLAB
Due: July 24, 2023

Instructor: Dr. Hanna Bullata

Simple calculator

We would like to build a simple calculator that is able to perform mathematical operations on integer numbers. We'll assume the following:

- Each integer number will not exceed 65535 (2 bytes).
- The supported mathematical operations are only the + (addition), / (division) and % (modulo). The division operation should output both the quotient and remainder parts.

The hardware should be composed of a 16F877A microcontroller, a push button P to enter the numbers and a 16×2 character LCD. The system should behave as follows:

- When the system is powered up, the first line of the LCD should display the message `Enter Operation`.
- On the second line, when we click on the push button, the ten thousandth part of the first number should increment by 1. Since originally the ten thousandth 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 6. On the seventh click, the ten thousandth number goes back to 0, and so on.
- If we leave P unclicked for over 2 seconds, the ten thousandth part of the first number becomes fixed. Next, the thousandth part of the first number will be increased when we click the push button P. If the ten thousandth part is 6, then the thousandth part should not exceed 5 before rolling back to 0. Otherwise, if the ten thousandth part is lower than 6, then the thousandth part is allowed to increase with every click from 0 to 9 and then back to 0, and so on.
- If we leave P unclicked for over 2 seconds, the thousandth part of the first number becomes fixed. Next, the hundredth part of the first number will be increased when we click the push button P. If the ten thousandth part is 6 and the thousandth part is 5, then the hundredth part should not exceed 5 before rolling back to 0. Otherwise, the hundredth part is allowed to increase with every click from 0 to 9 and then back to 0, and so on.
- The same behavior described above applies for both the tenth and unit values of the first number.
- After entering the unit value, if we leave P unclicked for over 2 seconds, the first number becomes fixed. Next, we need to choose the mathematical operation. The default selected operation is the + operation (addition). Clicking on P will select and display the operation / (division). Clicking for the second time will select and display the operation % (modulo). Further clickings will toggle between the +, the / and the % operations.
- If we leave P unclicked for over 2 seconds, the mathematical operation has been selected. 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 ten thousandth part first, then the thousandth part, then the hundredth part, then the tenth part, then the unit part).

- Once the second integer number has been entered, the system should put the sign `=` and then execute the mathematical operation and display the result after the `=`. In addition, the first LCD line should display the message `result`.
- After 3 seconds, the system should display on the first row the message `Keep? [1:Y, 2:N]`. The idea here is that the user might want to keep the numbers he/she selected but chage the operation. A single click means yes, a double-click means no. If the user makes a double-click (No), go back to the top when the system was powered up and repeat the same steps (e.g. select the first number, the mathematical operation and the second number). If the user makes a single click (Yes), display the 2 numbers selected previously and position the cursor on the operation type. As mentioned above, the default operation is the addition operation. Clicking on the push button P will toggle between +, / and % operations. If we leave P unclicked for over 2 seconds, the mathematical operation has been selected. The system should put the sign `=` and then execute the mathematical operation and display the result after the `=`.
- After 3 seconds, go back to the previous step and display on the first row the message `Keep?  [1:Y, 2:N]` and repeat the same steps described above.

What you should do

- Build the controller described above on a bread board. Remember to add a 10KΩ pull-up resistor to push button P (or use the internal weak pull-up resistor if using PortB), add a 4MHZ oscillator (with 2 × 15pF capacitors) and a 10KΩ pull-up resistor to the MCLR pin.
- Use the LCD in the 4-bit mode. Connect pin D_4 of the LCD to RA_0, pin D_5 to RA_1, pin D_6 to RA_2, pin D_7 to RA_3, pin RS to RA_4 and pin E to RA_5. Remember to pull up the RS pin using a 4.7KΩ resistor.
- Enable interrupts on push button P.
- Build the PIC assembly code that implements the behavior described above under MPLAB IDE.
- Assemble your code and make sure you get a successful build. Use the simulator if you wish to make sure the behavior is correct.
- Send the zipped folder that contains the MPLAB project (including your source code) before the deadline. If the deadline is reached and you are still having problems with your code, just send it as is!