

Lab 2: Password Program with Keypad

Demo Due: October 16, 2015

Learning Objectives:

- Use debouncing techniques to debounce keypad presses
- Interface with a keypad
- Create a connector to interface with the external keypad
- Create a program that keeps several passwords

Datasheets and References (also on D2L)

[4x3 Keypad](#)

Hardware Provided for Lab:

1 - 8x2 LCD Screen (1.50" x 1")

1 - 4x3 keypad

Software Provided for Lab (Found on D2L):

- keypad.c, keypad.h, lab2p1.c

Lab Procedure and Demonstration:

Part 1: Keypad and LCD

Description:

In this part, you will interface the PIC32MX with the keypad and output the detected keys onto the LCD screen.

Note: This part will be using many of the pins on the PIC32MX. Therefore you may have to use analog pins for digital inputs. If you forget this, your keypad will appear to be broken, but it is not. Additionally, some internal pull-up resistors do not work properly. Therefore, you must make your own external pull-up circuits.

Requirements:

Using change-notification interrupts and open-drain collector configured pins, interface with the keypad. When a key is pressed, the number or letter on the keypad should be output to the LCD screen. The first line of the LCD screen should be written to first, then the second line of the LCD should be written to once the first line is full. After the second line is full, the keypad presses should overwrite characters on the first line of the LCD.

Hardware Design and Systems Integrator:

1. Using the wiring diagrams created in the pre-lab, solder a connector header of appropriate size onto the keypad. Create the male portion of the connector and insert wires into the appropriate slots. Use appropriate colors to identify rows and columns.
2. If you are using connectors from the previous lab, you should be able to attach terminals to the other side of the wires and insert them into the connector attached to the PIC.
 - a. If you are not using connectors, every connection must at least be a soldered connector or wire-wrapped.

Software Design:

1. Create a program that fulfills the requirements.
 - a. Use define statements to make it so that it is easier to change which pins you are using. There should be a define statement for references to LAT, PORT, TRIS, CNPU, CNEN and ODC registers.

Quality Assurance:

1. Test the connectivity on the connector and the header attached to the keypad.
2. Without using the keypad, test to see if the column pins will be pulled down when they are connected to the row pins. You can use jumper cables and connect each row pin to each column pin one at a time. You can view the voltage on the oscilloscope.
3. Test and verify that the datasheet is correct in regards to which pin on the keypad is connected to which row.

Part 2: 4-digit Password System

Description:

In this part, you will use the keypad and LCD to create a password system.

Requirements:

Create 4-digit password system that lets a user enter a four-digit number ('0' – '9') and then shows whether the entered four digits is a good password or not. There must also be a provision for adding passwords into the password database. The system must be able to hold at least four passwords.

Team:

1. Create a program that implements the following state diagram.
2. An invalid password is any password with "*" or "#". The program must display "Bad" or "Invalid" when these keys are pressed except for when going into "Set Mode."
3. "Set Mode" is entered when the user enters "***" in the "Enter" state. This is the "Program Mode Sequence"

