

Project Lab Sheet C: Number Input using LCD

Two projects must be submitted; see details on QMPlus.

1 Aims

The aim of this lab is to use the LCD and the buttons on the LCD shield to be able to input (or select) a 3 digital decimal number.

1.1 Earlier Labs and Example Projects

You may adapt code from earlier labs and other example projects to complete the requirements described here. The following example projects are relevant:

1. **LCD:** explains the LCD shield and presents a software API to display characters on the LCD. The program demonstrates most of the LCD features.
2. **LCD buttons:** explains how the buttons on the LCD shield are used; this requires the ADC, so the code from the ADC examples is also included. The program can be used to display the voltages measured by the ADC when the buttons are pressed.

2 Project Requirements

The requirements of the system are described below.

1. The user must be able to use the four buttons on the LCD shield (Up, Down, Left, Right) to compose a 3-digit number:
 - The left and right keys move from one number to the next.
 - The up and down keys change the value of the current digit.
2. Text should also be displayed to prompt the user to enter a number.
3. Pressing the 'Right' key when the cursor is positioned over the right-most digit, causes the digits currently displayed to be 'entered' as a number. This means the digits should be converted to a number.
4. The software to enter a number must be implemented in a way that allows it to be included in another system (for example, to do arithmetic on it). The functionality to enter a number should be presented through a simple API.
5. You should demonstrate the use of the API by using the number to set the frequency in Hz of an audio tone. The frequency changes when a new number is entered. However, if the frequency is below an audible range (*determine this*), an error is displayed and the frequency does not change.

3 Suggestions

This section contains some additional information that may be useful. You do not have to read this section if you do not wish to.

3.1 Software Requirements

Work out the detailed software requirements. Some areas to be clarified include:

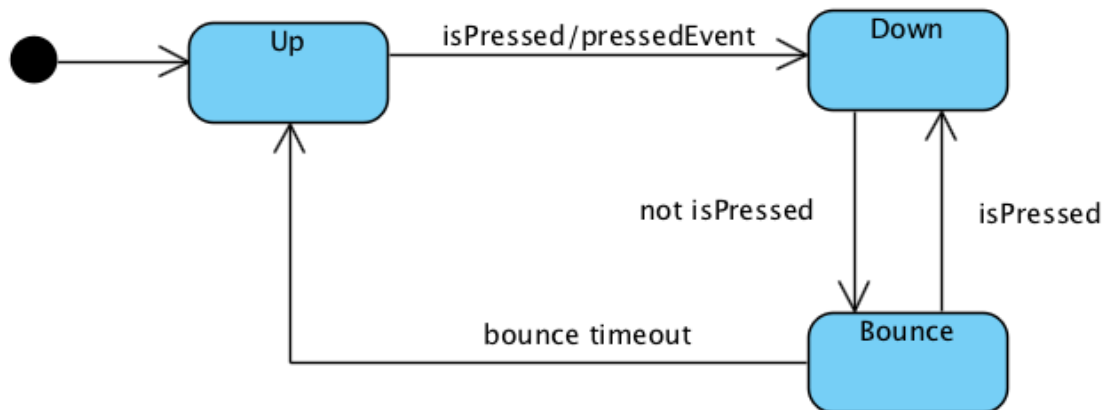
- What happens when the LEFT key is pressed at left hand digit on the LCD?
- What text will be displayed? Is it fixed, or a parameter in the API?
- What applications (very simple at first) will you use to show your number has been entered?

3.2 Software Design

The functionality of entering a number could be implemented with embedded delays; however, this would not meet requirement 4. Instead, it needs to be tasks that are executed (for example) in a cyclic system. At least two tasks are recommended:

1. A task to detect button press events.
2. A task to manage the state of the LCD, acting on the button press events.

The first task can be modelled on the task Task1PollInput in the demonstration project. The state diagram for this task is shown below.



The button is initially up. An event (represented by a variable) is generated when the button is pressed (closed). As long as it stays pressed, the system remains in state 'Down'. When the button is released (so that `isPressed()` returns false), it goes to state 'Bounce'. The system must remain in this state for a while before another pressed event can be generated.

This state diagram needs to be extended for 4 buttons. Consider whether you need 4 separate bounce states or just one.