Sampling a Matrix Keypad

by David Dempster, July 2011

Abstract

How often have you needed to sample a keypad and then take a certain action depending on which key was pressed? Problems occur in loops where a key may be sampled multiple times, the solution is fairly straightforward, in this article David shows how to use edge detection to remove this problem from your program.

Requirements

Software:

- Any licence of Flowcode v3 or v4 for any variant.

Hardware:

- EB014 Keypad E-block
- EB006 Multiprogrammer
- EB005 LCD E-block

Overview

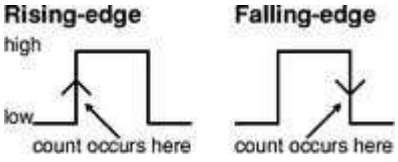
How often have you needed to sample a keypad and then take a certain action depending on which key was pressed? This situation occurs quite often within a control program where you need to sample inputs/sensors/keypad, perform some computations and make decisions, before controlling some output devices.

Most control programs make use of a 'Loop Forever' structure which means that the keypad will be sampled every time the program executes the 'Loop'. If your program just detects whether a key is pressed you may end up with the situation where actions are repeatedly carried out whilst the key is pressed. A sort of machine-gun effect – which clearly is not what you want to happen.

The solution is fairly straightforward and involves detecting the rising-edge and falling-edge of a key. Using this approach means the time-duration whilst the key is pressed becomes unimportant – it is just the edge-detection points that are important.

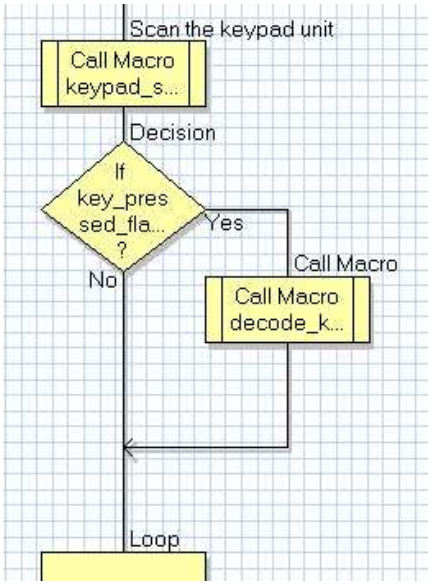
Edge Detection

The diagram below shows the rising- and falling-edges for a key on the keypad.



By taking 'snapshots' on a regular basis you can detect the rising-edge. Turning this idea into a computer program requires the program to remember the previous 'snapshot' and compare it with the current 'snapshot'. This article uses a 'flag' (which is really just another name for one of the PIC's internal registers or accumulators).

The 'flag' starts off in the 'down' state. By sampling the state of the keypad you can determine if a key has been pressed. If a key is pressed AND the 'flag' is 'DOWN' then you must have encountered a rising-edge. Transferring this event into a piece of Flowcode is very simple and can be achieved with a Decision Icon (as shown in the following piece of code).



Once the rising-edge has been detected the Flowcode sets the 'flag' to 'UP'. This ensures that the piece of code that detected the rising-edge cannot be executed again until the particular key has been released. A similar piece of code is used to detect the falling-edge as shown in the above piece of Flowcode. The only action that needs to be performed, after detecting the falling-edge, is to return the 'flag' to the 'DOWN' state so that the Flowcode is ready to detect when a key is pressed again (i.e. the next rising-edge).

The test program (see attached Flowcode file) that accompanies this article simply displays a message on an LCD panel to inform you which key had been pressed. The following three user macros handle everything associated with the keypad:

'keypad_init'
This user macro makes assignments to the variables listed below.
UP = 255
DOWN = 0
key_pressed_flag = DOWN
keypad_flag = DOWN
keypad_data = 0
keypad_value =0

'keypad_scan'
If a key has been pressed then this user macro returns the following values.
the binary value for the key that was pressed in the variable 'keypad_value'.
a flag to indicate that a key has been pressed. (i.e. 'key_pressed_flag' = UP)

'decode_keypad'
This user macro consists of a series of decision icons that determine which key was pressed and then carry out a particular function. In this article the 'function' simply displays on an LCD panel the numerical value of the key that was pressed. It should be noted that the 'key_pressed_flag' needs to be set to the OFF state when each 'function' is completed.

If you want to incorporate the attached Flowcode into your control program then you will need to alter the Flowcode contained inside the 'decode_keypad' macro according to your needs.

Please note: there is an article on the Forum that describes how to sample a set of push-buttons using the same sort of 'snapshot' technique.

I hope you will find this program useful.

Further reading

Below are some links to other resources and articles on related subjects, and technical documentation relating to the hardware used for this project...

Flowcode:	http://www.matrixmultimedia.com/flowcode.php
Learning Centre:	http://www.matrixmultimedia.com/lc_index.php
User Forums:	http://www.matrixmultimedia.com/mmforums
Product Support:	http://www.matrixmultimedia.com/sup_menu.php