Debouncing 8 Buttons in 7 Clock Cycles

APPLICATION OPERATION:

The code fragment described here is capable of protecting against noise and debouncing with eight button inputs (five only on PIC12C5xx due to limited inputs) in only seven clock cycles. A variable called ‘_GPIO’ is provided which contains the debounced states of all inputs in ‘GPIO’. Therefore, no changes are necessary concerning the number and position of the inputs.

Call this routine periodically (for example by using the quick code snippet 'Implementing software timer interrupts'), but for proper operation the interval may not be less than T/3, where T is the duration of button bouncing. An interval of T/2 should be a good value. Each button will be checked and will be considered valid if the button state did not toggle since the last execution of the code.

The code is based on the following truth table (performed parallel for all bits in GPIO):

<table>
<thead>
<tr>
<th>OldGPIO</th>
<th>currentGPIO</th>
<th>_GPIO</th>
<th>resulting_GPIO</th>
<th>resultingOldGPIO</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>*</td>
<td>0</td>
<td>0</td>
<td>Current value of GPIO equals previous value: Consider state to be valid.</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>*</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Input state has just changed and is not stable: Leave state of _GPIO unaffected.</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

SUMMARY:

To protect your software against the consequences of noise and button bouncing call this code fragment periodically and use _GPIO instead of GPIO to read out input lines.

AUTHOR’S NOTE:

Code was written for a PIC12C5XX originally, and therefore uses the symbol GPIO for I/O access. Change it appropriately to work on other PICmicro™ microcontrollers and duplicate the code to add more buttons if necessary.

Author: Marc Hoffknecht
Aachen, Germany
email: hoffknecht@online.de

(*) don’t care

EPROM usage: 7 byte
RAM usage: 2 byte
Clock Cycles: 7 cycles
;***********************************************************************
;* Quick-Code-Snippet: Debouncing five/eight(*) buttons in seven cycles   *
;*     *
;* (*) eight for other PICmicro's                        by Marc Hoffknecht *
;***********************************************************************

processor 12c508
radix dec
include "p12c508.inc"

#define __12C508

CBLOCK 0x0C ; start of RAM
ENDC

#define COMWXORLW 255 ; (com)plement (w)

CBLOCK
_GPIO ; reserve file-registers for
OldGPIO ; the variables
ENDC

MOVF GPIO, W ; initialize variables
MOVWF OldGPIO
MOVWF_GPIO

Loop

MOVF GPIO, W ; for a button bouncing of T ms,
XORWF OldGPIO, W ; call this about every T/2 ms.
ANDWF _GPIO
XORWF OldGPIO
COMW
ANDWF OldGPIO, W
IORWF _GPIO

NOP ; for evaluation, use debugger here
; to check _GPIO and modify GPIO
GOTO Loop

;***********************************************************************

END