



## Consumer Appliance, Widget, Gadget

# Debouncing 8 Buttons in 7 Clock Cycles

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

### 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):

OldGPIO	current GPIO	_GPIO	resulting _GPIO	resulting OldGPIO	Comment
0	0	*	0	0	Current value of GPIO equals previous value: Consider state to be valid.
1	1	*	1	1	
0	1	0	0	1	Input state has just changed and is not stable: Leave state of _GPIO unaffected.
1	0	0	0	0	
		1	1		

### SUMMARY:

(\*) don't care

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.

**EPROM usage:** 7 byte  
**RAM usage:** 2 byte  
**Clock Cycles:** 7 cycles

### 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.

Microchip Technology Incorporated, has been granted a nonexclusive, worldwide license to reproduce, publish and distribute all submitted materials, in either original or edited form. The author has affirmed that this work is an original, unpublished work and that he/she owns all rights to such work. All property rights, such as patents, copyrights and trademarks remain with author.

# Consumer Appliance, Widget, Gadget

---

## APPENDIX A: SOURCE CODE

```
;*****
;* 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  COMW      XORLW 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
```