APPLICATION OPERATION

The idea is to replace the electromechanical switches with capacitive sensor buttons (PCB round or square pads). PIC12CXXX are very suitable for this purpose and with few components we may design a low cost fully electronic switch or regulator. The method uses a simple RC delay circuit (Figure 1), when the time constant changes when the sensor is touched.

FIGURE 1: RC DELAY CIRCUIT

To read the state of the sensor, the microcontroller must perform only two steps (Figures 2 and 3):

Step 1: Changing the state of output from “0” to “1” (write operation - Twr).
Step 2: Reading the state of input (read operation - Trd).

FIGURE 2: UNTOUCHED SENSOR

If the result of read operation is “0”, it means that the sensor is touched. Human capacitance has been connected serially to capacitor C and the time constant of the circuit has become bigger.

Because of the small capacitance of human’s fingers, the interval time between Twr and Trd should be less then 1-2 μs, so we recommend the use of 4 MHz internal clock of PIC12CXXX.

The figure under Graphical Hardware Representation contains an example with two sensor buttons. The trick here is that for reading the sensor SB0: GP0 is set as input, GP1 is set as output and for reading the SB1: GP0 is set as output, GP1 is set as input. The outputs GP4,5 are connected to LEDs and show the state of sensor buttons. The other pins are not discussed here and they can be used for example as SYNCRO input and TRIAC DRIVER output.

The value of resistor R1 must be adjusted and it determines the sensibility of the buttons. The values of capacitors C1 and C2 are not so important. We recommend the use of the same type inputs of microcontroller, with equal input impedances.
Electromechanical Switch Replacement

GRAPHICAL HARDWARE REPRESENTATION

FLOWCHART

MICROCHIP TOOLS USED

Assembler/Compiler Version:
MPLAB 3.22, MPASM 1.5
APPENDIX A: SOURCE CODE

;Software listing: sensor.asm
;***********************************************************************
; Connecting sensor buttons to PIC12C508
; Osc.: F=4MHz (internal)
; Written by Vladimir Velchev 06.1997. (C) AVEX
; Version 1.00
;***********************************************************************

; GP0 - sensor button SB0 (input/output)
; GP1 - sensor button SB1 (input/output)
; GP2 - not used (output)
; GP3 - not used (input)
; GP4 - LED1 for sensor button SB0 (output: 0=LED ON, 1=LED OFF)
; GP5 - LED2 for sensor button SB1 (output: 0=LED ON, 1=LED OFF)

LIST  P=12C508

#include <p12C508.inc>

;*** Equates
SB0 equ 0  ;sensor button SB0
SB1 equ 1  ;sensor button SB1
SB0_MASK equ B'00000001'  ;bit mask for SB0
SB1_MASK equ B'00000010'  ;bit mask for SB1
OUT0 equ 0  ;driving output for SB1
OUT1 equ 1  ;driving output for SB0
LED1 equ 4  ;LED for button SB0
LED2 equ 5  ;LED for button SB1
IOSET equ B'00001000'  ;initial I/O port settings
POOL INPUT, others- outputs

org 0  ;RESET vector

;*** Code Starting Point
MAIN:
; Initial setup
movlw IOSET  ;init GPIO
tris GPIO
clrf GPIO  ;reset all outputs (=0)

Main_Loop:
clrwdt  ;clear watchdog timer

; Space for user code
;
;
;
;
call READ_SENSORS  ;call subroutine for sensors
btfss STATUS,C  ;skip if sensor SB0 touched (C=1)
goto SB0_untouched  ;else - go to turn off LED1
bcf GPIO,LED1  ;LED1=OFF
goto Check_SB1  ;go to checking next button

SB0_untouched:bsf GPIO,LED1  ;LED1=ON

Check_SB1: btfss STATUS,DC  ;skip if sensor SB1 touched (DC=1)
goto SB1_untouched  ;else - go to turn off LED2
bcf GPIO,LED2  ;LED2=OFF
goto Main_Loop  ;go to beginning

SB1_untouched:bsf GPIO,LED2  ;LED2=ON

goto Main_Loop  ;go to beginning

;*** Subroutine - READ_SENSORS
; Input:
; Output: Flags: C=1 if sensor SB0 touched, DC=1 if sensor SB1 touched
READ_SENSORS:

bcf STATUS,C  ;clear carry flag
bcf STATUS,DC  ;clear digit carry flag
;Read sensor button SB0
  movlw  IOSET|SB0_MASK ; set SB0 as input (GP0)
  tris   GPIO
  bsf   GPIO,OUT1 ; send strobe to output ___---
  btfss  GPIO,SB0 ; skip if sensor SB0 untouched (GP0=1)
  bsf   STATUS,C ; else - set C flag
;Discharging the capacitance of the input
  movlw  IOSET&(~SB0_MASK) ; set SB0 as output (GP0)
  tris   GPIO
  bcf   GPIO,OUT0 ; reset output 0 (discharging)
  bcf   GPIO,OUT1 ; reset output 1
;Read sensor button SB1
  movlw  IOSET|SB1_MASK ; set SB1 as input (GP1)
  tris   GPIO
  bsf   GPIO,OUT0 ; send strobe to output ___---
  btfss  GPIO,SB1 ; skip if sensor SB1 untouched (GP1=1)
  bsf   STATUS,DC ; else - set DC flag
;Discharging the capacitance of the input
  movlw  IOSET&(~SB1_MASK) ; set SB1 as output (GP1)
  tris   GPIO
  bcf   GPIO,OUT1 ; reset output 1 (discharging)
  bcf   GPIO,OUT0 ; reset output 0
  return
end ; end of program
Electromechanical Switch Replacement

NOTES: