

'-----Title-----

' File.....bend_sensor1.pbp
' Started....3/4/08
' Microcontroller used: Microchip Technology PIC16F88
' microchip.com
' PicBasic Pro Code: micro-Engineering Labs, Inc.
' melabs.com

'-----Program Description-----

' Program converts analog input from a bend sensor
' into a change in the position of a hobby servo.

'-----Comments-----

' Servo used is a Futaba S3003 standard servo.
' The bend or flex sensor is Jameco #150551,
' 0 degrees - 10K ohms, 90 degrees - 30-40K ohms.

'-----PIC Connections-----

16F88 Pin	Wiring
RA0	Center bend sensor voltage divider
RA4	LCD Register Select(RS)
RB0	Servo Control Wire (Futaba white wire)
RB3	LCD Enable(E)
RB4	LCD (DB4)
RB5	LCD (DB5)
RB6	LCD (DB6)
RB7	LCD (DB7)

' See schematic for the other usual PIC connections

'-----LCD Connections-----

LCD Pin	Wiring
1	Ground(Vss)
2	+ 5v(Vdd)
3	Center of 20K Pot(Contrast)
4	RA4(Register Select,RS)
5	Ground(Read/Write,R/W)
6	RB3(Enable)
7	No Connection(DB0)
8	No Connection(DB1)
9	No Connection(DB2)
10	No Connection(DB3)
11	RB4(DB4)
12	RB5(DB5)
13	RB6(DB6)
14	RB7(DB7)

'-----Revision History-----

```

'-----Constants/Defines-----

' To free up AN0 (Pins RA0) for an analog input,
' the default LCD data lines, DB4-DB7,
' function was removed from RA0 - RA3.
' They are relocated to PORTB.4 - PORTB.7 (RB4-RB7)
' using the LCD DEFINE statements below. All other
' default LCD pins and functions are left unchanged.

    DEFINE LCD_DREG  PORTB      'Sets PORTB as LCD data port
    DEFINE LCD_DBIT  4          'Start data connections to bit 4

    DEFINE ADC_BITS  10        'Sets the number of bits in
                                'the result to 10

'-----Variables-----

    bend  VAR      WORD      'Word for bend sensor voltage divider
    servo VAR      WORD      'Word to store calculated servo value

'-----Initialization-----

    ANSEL = %00000001  'Leaves AN0 in analog mode, but
                        'changes other analog bits to digital.
                        'See table below.

'
'   Analog Bit      Analog or Digital      PIC16F88 Pin
'   -----
'   AN0             Analog                 RA0
'   AN1             Digital                 RA1
'   AN2             Digital                 RA2
'   AN3             Digital                 RA3
'   AN4             Digital                 RA4
'   AN5             Digital                 RB6
'   AN6             Digital                 RB7

    ADCON1 = %10000000  'Right justifies 10-bit value of bend
                        'in 16-bit WORD. Adds "0" in the
                        '6 Most Significant bits of the Word,
                        'shifting the 10-bit value of bend to
                        'the right.

    OSCCON = $60        'Sets the internal oscillator in the
                        '16F88 to 4 MHz
    PORTB = 0          'A must for the PULSOUT command,
                        'establishes proper polarity for servo

'-----Main Code-----

    PAUSE 1000        'Pause to allow LCD to setup

start:

    ADCIN 0, bend    'Read analog voltage on AN0 and
                        'convert to 10-bit digital value

```

```
                                'and store as bend.

LCDOUT $FE,1,"bend = ", DEC bend
                                'Clears LCD screen, displays
                                '"bend = " and the 10-bit
                                'value of bend

' The range of bend values is from about 350 to 620. Use the
' following formula to convert bend values to the range of
' the servo pulses, 60 - 220.

servo = (bend/8)*5-160           'Convert bend value to servo
                                'PULSOUT value

IF servo >220 THEN servo = 220   'Create upper threshold to protect servo

IF servo <60 THEN servo = 60    'Create lower threshold to protect servo

LCDOUT $FE,$C0,"servo =", DEC servo
                                'LCD jumps to beginning of second
                                'line and displays "servo = "
                                'and the 10-bit value of servo.

PULSOUT 0,servo                 'Send position signal to RB0. Pulse out
                                'time varies from 2.2 msec to 0.6 msec.

PAUSE 20                        'Pause 20 msec, gives proper period for
                                'pulses to servo

GOTO start                       'Start over again

END
```