

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

' File.....16F877A_servo4.pbp
' Started....6/1/05
' Microcontroller used: Microchip Technology 16F877A
' microchip.com
' PicBasic Pro Code: micro-Engineering Labs, Inc.
' melabs.com

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

' Three momentary switches rotate the servo into three
' different positions: left, center, and right.
' Discussion about basic servo pulse control may be found
' at www.seattlerobotics.org/guide/servos.html or
' www.geocities.com/hobby_robotics/was.htm

'-----Related Lesson-----'

' servo4.pbp (the 16F88 program) is used in
' the lesson PIC PROGRAMMING 3 SERVOS at:
' http://cornerstonerobotics.org/curriculum/lessons_year2/erii13_pic_programming3_servos.pdf

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

' WITH THE PIC16F877A, MAKE SURE TO HAVE SEPARATE +5V
' POWER SOURCES FOR THE PIC AND THE SERVO. MAKE SURE TO
' HAVE A COMMON GROUND BETWEEN THE PIC AND SERVO. We use one 9V
' battery and two 78L05 voltage regulators. See
' discussion about voltage regulators at:
' http://cornerstonerobotics.org/curriculum/lessons_year2/erii3_diodes_power_supplies_voltage_reg.pdf

' Also, initialize the state of PORTB as LOW
' since that will set the correct polarity of the
' PULSOUT statement. See PULSOUT in PicBasic Pro
' Compiler manual by microEngineering Labs, Inc.
' Look around page 121 in the PicBasic Pro Compiler Manual

'-----Connections-----'

' See schematic at:
' http://www.cornerstonerobotics.org/schematics/pic16f877a_servo4.pdf

PIC16F877A Pin	Wiring
-----	-----
RB0	Servo Control Wire
RB1	Momentary Switch 1
RB2	Momentary Switch 2
RB3	Momentary Switch 3
Vdd	+5 V
Vss	Ground
MCLR	4.7K Resistor to +5 V

```
'-----Revision History-----
' 11/14/07 Change MCU from 16F84A to 16F88
' 11/14/07 Add 16F88 oscillator initialization
' 11/27/07 Add power supply warning and
'           changed title from servo1.pbp to servo4.pbp
' 1/2/09   Change MCU from 16F88 to 16F877A

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

    i      VAR      BYTE      ' BYTE for counter, i

'-----Switch Connection Pins-----

    switch1  VAR PORTB.1      ' Labels PORTB.1 as switch1
    switch2  VAR PORTB.2      ' Labels PORTB.2 as switch2
    switch3  VAR PORTB.3      ' Labels PORTB.3 as switch3

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

    TRISB = %00001110      ' Sets RB1,RB2,RB3 as inputs,
                          ' all other pins as outputs.

    PORTB = %00000000      ' Equivalent to: PORTB = 0
                          ' Sets all PORTB pins to LOW(0 volts)
                          ' Make certain to include this
                          ' initialization as it sets the
                          ' proper polarity of pulses in
                          ' the PULSOUT command.

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

start:

    IF switch1 = 1 THEN GOSUB left
                          ' If the switch on PORTB.1 is pushed,
                          ' PORTB.1 becomes high (+5 volts) and
                          ' the comparison is true; the program
                          ' jumps to the subroutine labeled left.

    IF switch2 = 1 THEN GOSUB center
                          ' If the switch on PORTB.2 is pushed,
                          ' PORTB.2 becomes high (+5 volts) and
                          ' the comparison is true; the program
                          ' jumps to the subroutine labeled center.

    IF switch3 = 1 THEN GOSUB right
                          ' If the switch on PORTB.3 is pushed,
                          ' PORTB.3 becomes high (+5 volts) and
                          ' the comparison is true; the program
                          ' jumps to the subroutine labeled right.

    GOTO start          ' Jumps to label loop

    END                ' Place subroutines after END
```

'Subroutines:

left:

```
FOR i = 1 TO 10      ' Send left signal 10 times

PULSOUT 0,100      ' Pulse Width:
                   ' Sends a pulse out on pin RB0 for 1.0 mS.
                   ' The period,(100) is multiplied by the
                   ' increment for a 4 MHz oscillator (10 uS)
                   ' to get a pulse out time of 1.0 mS.
                   ' To get the full range of your
                   ' servo, you may have to decrease the period
                   ' to less than 100, being careful not to go too
                   ' low thereby forcing the servo to stop
                   ' mechanically against the internal mechanical
                   ' stops.

PAUSE 20 - 1       ' Pulse Interval:
                   ' Pause 20 ms less pulse width (1.0 ms)
                   ' This equation keeps the period of
                   ' the servo pulse a constant 20 ms, HIGH
                   ' for 1.0 ms and LOW for 19 ms = 20 ms.

NEXT i              ' Go back to the FOR statement and do
                   ' next count, i

LOW 1              ' Bring switch1,(RB1),state to LOW to prevent
                   ' the pin's voltage from floating

RETURN             ' Returns execution to statement following
                   ' GOSUB left command
```

center:

```
FOR i = 1 TO 10      ' Send center signal 10 times

PULSOUT 0,150      ' Pulse Width:
                   ' Sends a pulse out on pin RB0 for 1.5 mS.
                   ' The period,(150) is multiplied by the
                   ' increment for a 4 MHz oscillator (10 uS)
                   ' to get a pulse out time of 1.5 mS.

PAUSE 20 - 15/10   ' Pulse Interval:
                   ' Pause 20 ms less pulse width (15/10 = 1.5 ms)
                   ' This equation keeps the period of
                   ' the servo pulse a constant 20 ms, HIGH
                   ' for 1.5 ms and LOW for 18.5 ms = 20 ms.
                   ' Note: In PicBasic Pro, numbers and variables
                   ' must always be non-negative integers.
                   ' 20 - 1.5 is not permissible since 1.5 is not
                   ' an integer. To get around the problem, use
                   ' integers 15/10 to create the same value.

NEXT i              ' Go back to the FOR statement and do
```

```

                                ' next count, i

LOW 2                            ' Bring switch2,(RB2),state to LOW to prevent
                                ' the pin's voltage from floating

RETURN                           ' Returns execution to statement following
                                ' GOSUB center command
right:

FOR i = 1 TO 10                  ' Send right signal 10 times

PULSOUT 0,200                    ' Pulse Width:
                                ' Sends a pulse out on pin RB0 for 2.0 mS.
                                ' The period,(100) is multiplied by the
                                ' increment for a 4 MHz oscillator (10 uS)
                                ' to get a pulse out time of 2.0 mS.
                                ' To get the full range of your
                                ' servo, you may have to increase the period
                                ' to more than 200, being careful not to go too
                                ' high thereby forcing the servo to stop
                                ' mechanically against the internal mechanical
                                ' stops.

PAUSE 20 - 2                     ' Pulse Interval:
                                ' Pause 20 ms less pulse width (2 ms)
                                ' This equation keeps the period of
                                ' the servo pulse a constant 20 ms, HIGH
                                ' for 2 ms and LOW for 18 ms = 20 ms.

NEXT i                            ' Go back to the FOR statement and do
                                ' next count, i

LOW 3                            ' Bring switch3,(RB3),state to LOW to prevent
                                ' the pin's voltage from floating

RETURN                           ' Returns execution to statement following
                                ' GOSUB right command
```