

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

' File.....16F877A_servo1.pbp
' Started....1/8/08
' Microcontroller used: Microchip Technology 16F877A
' microchip.com
' PicBasic Pro Code: micro-Engineering Labs, Inc.
' melabs.com

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

' Basic servo program with the format for controlling
' servo pulses using PicBasic Pro PULSOUT command.
' Servo cycles between counterclockwise and
' clockwise positions.

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

' servo1.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

' servo1.pbp is also used in the lesson HACKING SERVOS at:
' http://www.cornerstonerobotics.org/curriculum/lessons_year2/erii17_hacking_servos.pdf

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

' WITH THE PIC16F877A, MAKE SURE TO HAVE SEPARATE POWER
' SUPPLIES 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, (PORTB = 0), as LOW
' since that will set the correct polarity of the
' PULSOUT statement.

' Discussion about basic servo pulse control may be found
' at www.seattlerobotics.org/guide/servos.html or
' www.geocities.com/hobby_robotics/was.htm

' Servos may be modified or hacked to allow
' for continuous rotation so they can be used
' as motors on small robots. The book
' Amphibionics by Karl Williams gives an
' in depth treatment on how to modify servos.
' Also see Lesson 17 in the Year Two curriculum
' on the cornerstonerobotics.org web site.

'-----New PicBasic Pro Commands-----

```
' The PicBasic Pro Compiler Manual is on line at:
' http://www.microengineeringlabs.com/resources/index.htm#Manuals

' PULSOUT pin, period
' This command sends a pulse to pin for the period defined.
'
' For example:
'
'     PULSOUT 0,200      Sends a pulse out on pin RB0 for 2.0 ms.
'                       The period,(200), is multiplied by the
'                       increment for a 4 MHz oscillator (10 us)
'                       to get a pulse out time of 2.0 ms.
'
' Look around page 121 in the PicBasic Pro Compiler Manual
'
' Another PBP command that may be substituted for PULSOUT
' is PAUSEUS. See:
' http://cornerstonerobotics.org/code/servo2.pdf

'-----Revision History-----
' 1/1/09      Change from 16F88 to 16F877A

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

' See schematic at:
' http://www.cornerstonerobotics.org/schematics/pic16f877a\_servo\_1\_2\_3.pdf

'     PIC16F877A Pin      Wiring
'     -----
'     RB0                 Servo Control Wire
'     Vdd                 +5 V
'     Vss                 Ground
'     MCLR                4.7K Resistor to +5 V

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

'     i      VAR      BYTE      ' BYTE for counter variable, i

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

'     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.
'                           ' To set just one pin such as RB0, to
'                           ' LOW, enter PORTB.0 = 0.

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

start:
```

```
FOR i = 1 TO 40      ' Counterclockwise position:
                    ' Send signal 40 times. To change the time
                    ' the servo remains in one position, change
                    ' from 40 to another value.

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.

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 ms and LOW for 19 ms = 20 ms.

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

FOR i = 1 TO 40     ' Send clockwise signal 40 times

PULSOUT 0,200       ' Pulse Width:
                    ' Sends a pulse out on pin RB0 for 2.0 ms.
                    ' The period,(200), is multiplied by the
                    ' increment for a 4 MHz oscillator (10 us)
                    ' to get a pulse out time of 2.0 ms.

PAUSE 20 - 2        ' Pulse Interval:
                    ' Pause 20 ms less pulse width (2.0 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

GOTO start          ' Makes the program run forever.

END
```