```
'---------------Title--------------

' File......pbp_vb_servo1.pbp
' Started....1/25/09
' Microcontroller used:  Microchip Technology PIC16F88
'                        microchip.com
' PicBasic Pro Code: micro-Engineering Labs, Inc.
'                    melabs.com

'--------Program Desciption--------

' Visual Basic.NET program controls PIC16F88 to change
' positions of a hobby servo.

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

' pbp_vb_servo1.pbp is used in the lesson Visual Basic 1 at:
' http://cornerstonerobotics.
org/curriculum/lessons_year2/erii_visual_basic1.pdf

'--Visual Basic 2008 Express Edition--

' To download VB 2008 Express Edition, see:
' http://www.microsoft.com/express/download/

'--------Visual Basic Code---------

' For the VB.NET code that interfaces with this PBP program,
' see: http://www.cornerstonerobotics.org/code/vb_servo1.pdf

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

' WITH THE PIC16F88, MAKE SURE TO HAVE SEPARATE 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, (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, Hacking Servos at:
' http://www.cornerstonerobotics.
org/curriculum/lessons_year2/erii17_hacking_servos.pdf
```

```
'-------PicBasic Pro Commands------

' The PicBasic Pro Compiler Manual is on line at:
' http://www.melabs.com/support/index.htm then under the
' Compiler Documentation: click on PICBASIC PRO Compiler
' Manual.

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

'   16F88 Pin   Function      Name Given      Wiring
'                             In Program
'   ---------  ------------   ----------    ----------
'
'      RB4                      servo       Servo Control Wire
'      RB2     Receiver Pin     PICSI       MAX232 Pin 9
'      RB5     Transmit Pin     PICSO       MAX232 Pin 10
'
' See the schematic for the PIC power and MCLR connections


' MAX232 Pin  Datasheet          Function and Wiring
'            Designation
' ---------   ---------   ------------------------------------
'
'  Pin 7      T2OUT      Receive Data to Male RS232 DB9 Pin 2
'  Pin 8      R2IN       Transmit Data from Male RS232 DB9 Pin 3
'  Pin 9      R2OUT      Receive Data to PIC RB2
'  Pin 10     T2IN       Transmit Data from PIC RB5
'
' See schematic at:
http://www.cornerstonerobotics.org/schematics/pic_vb_servo1.pdf

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

     MODE        VAR   WORD      ' WORD for MODE value
     P0          VAR   BYTE      ' BYTE for position variable P0
     c0          VAR   BYTE      ' BYTE for counter variable c0
     servo       VAR   PORTB.4   ' Defines PORTB.4 name as servo
     PICSI       VAR   PORTB.2   ' Defines PORTB.2 name as PICSI
     PICSO       VAR   PORTB.5   ' Defines PORTB.5 name as PICSO

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

    PORTB = %00000000       ' Eqivalent 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.

    ANSEL = 0               ' Changes analog bits to digital.
```

```
    OSCCON = $60            ' Sets the internal oscillator in the
                            ' 16F88 OSCCON register to 4 MHz

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

    MODE = 188              ' Sets RX/TX speed to 188 (4800 baud)
                            ' MODE = 84 (9600 baud)
                            ' MODE = 396 (2400 baud)
                            ' See appendix in PicBasic Pro manual
                            ' for other MODE examples.
Main:

    SERIN2 PICSI, MODE, [P0]
                            ' PIC receives Command input
                            ' Format:  SERIN2 Pin, Mode, [Item1]
                            ' Pin = PICSI, Declared in variables
                            ' Mode = 188 (4800 baud rate)
                            ' [Item1} = [P0]

    FOR c0 = 0 TO 30        ' Send signal 30 times.  Our servo needed
                            ' 30 repetitions for the servo to rotate
                            ' through its full range.

    PULSOUT servo,P0        ' Sends a pulse, P0, out on servo pin(RB4).
                            ' The period, P0, is multiplied by the
                            ' increment for a 8 MHz oscillator
                            ' (10 us) to get a pulse out time.
                            ' For example, if P0 = 100,
                            ' 100 * 10us = 1000 us = 1 ms

    PAUSE 20 - P0/100       ' Pause 20 ms less pulse width (P0/100)
                            ' If P0 = 100, p0/100 = 100/100 = 1 ms.
                            ' This equation keeps the period of
                            ' the servo pulse a constant 20 ms.

    NEXT c0                 ' Go back to the FOR statement and do
                            ' next value of c0

    GOTO Main

    END
```