

```
'-----Title-----
' File.....pbp_vb_stepper_mot1.pbp
' Started....2/18/09
' Microcontroller used:  Microchip Technology PIC16F88
'                          microchip.com
' PicBasic Pro Code: micro-Engineering Labs, Inc.
'                          melabs.com

'-----Program Description-----
' Visual Basic.NET program controls PIC16F88 to change
' direction and speed of a stepper motor.

'-----Related Lesson-----
' See lesson on Stepper Motors at:
' http://cornerstonerobotics.
org/curriculum/lessons_year2/erii_stepper_motor.pdf
' See lesson on 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/VB/

'-----Visual Basic Code-----
' For the VB.NET code that interfaces with this PBP program,
' see: http://www.cornerstonerobotics.org/code/vb_stepper_motor1.pdf

'-----Comments-----
' WITH THE PIC16F88, BE CERTAIN TO HAVE SEPARATE POWER
' SOURCES FOR THE PIC AND THE STEPPER MOTOR.  MAKE SURE
' TO HAVE A COMMON GROUND BETWEEN THE PIC AND MOTOR.

'-----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
'      -----   -----           -----           -----
'      RB0        Step Mot Wire 1
'      RB1        Step Mot Wire 2
'      RB2        Step Mot Wire 3
'      RB3        Step Mot Wire 4
```

```

'      RB4      Receiver Pin      PICSI      MAX232 Pin 9
'      RB5      Transmit Pin     PICSO      MAX232 Pin 10
'      Vdd      +5 V
'      Vss      Ground
'      MCLR     4.7K Resistor to +5 V

' 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 RB4
' Pin 10     T2IN       Transmit Data from PIC RB5
'
' See schematic at:
http://www.cornerstonerobotics.org/schematics/pic\_vb\_servo1.pdf

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

i          VAR   BYTE      ' BYTE for i variable
Steps     VAR   WORD      ' WORD for Steps value
MODE      VAR   WORD      ' WORD for MODE value
Dx        VAR   BYTE      ' BYTE for direction variable, dx
Delay     VAR   WORD      ' WORD for variable Delay
PulseSeq  VAR   BYTE      ' Motor Pulse Sequence (1,2,3,4,1,...)
                        ' or (4,3,2,1,4,...)

PICSI     VAR   PORTB.4   ' Defines PORTB.4 name as PICSI
PICSO     VAR   PORTB.5   ' Defines PORTB.5 name as PICSO

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

TRISB = %00000000      ' Sets all PORTB pins to output

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.

start:

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

PulseSeq = 0

```

```
SELECT CASE Dx      ' Determine direction of rotation

CASE 0             ' If Dx = 0, CW rotation

    FOR i = 1 TO Steps ' Loop though i from 1 to whatever
                        ' the variable Steps is set to.

    IF PulseSeq >= 4 THEN

PulseSeq = 1        ' If PulseSeq >= 4, then restart
                    ' the 1,2,3,4 sequence at 1.

    ELSE

PulseSeq = PulseSeq + 1
                    ' If PulseSeq is not >= 4, increment
                    ' PulseSeq to next step in the sequence.

    ENDIF

' Send the correct signal to PORTB
' PulseSeq value 0, 1, 2, 3, or 4

    LOOKUP PulseSeq, [0, 8, 4, 2, 1], PORTB
                    ' If PulseSeq = 0, PORTB = 0 (%00000000)
                    ' If PulseSeq = 1, PORTB = 8 (%00001000)
                    ' If PulseSeq = 2, PORTB = 4 (%00000100)
                    ' If PulseSeq = 3, PORTB = 2 (%00000010)
                    ' If PulseSeq = 4, PORTB = 1 (%00000001)

    PAUSE Delay

    NEXT i

PORTB = 0           ' Equivalent to PORTB = %00000000
                    ' All coils are turned off; the motor stops.

    GOTO start

CASE 1             ' If Dx = 1, CCW rotation

    FOR i = 1 TO Steps ' Loop though i from 1 to whatever
                        ' the variable Steps is set to.

    IF PulseSeq <= 1 THEN

        PulseSeq = 4 ' If PulseSeq <= 1, then restart
                    ' the 4,3,2,1 sequence at 4.

    ELSE

        PulseSeq = PulseSeq - 1
                    ' If PulseSeq is not <= 1, decrementing
                    ' PulseSeq to next step in the sequence.

    ENDIF
```

```
LOOKUP PulseSeq, [0, 8, 4, 2, 1], PORTB
    ' If PulseSeq = 4, PORTB = 1 (%00000001)
    ' If PulseSeq = 3, PORTB = 2 (%00000010)
    ' If PulseSeq = 2, PORTB = 4 (%00000100)
    ' If PulseSeq = 1, PORTB = 8 (%00001000)
    ' If PulseSeq = 0, PORTB = 0 (%00000000)

PAUSE Delay

NEXT i

PORTB = 0
    ' Equivalent to PORTB = %00000000
    ' All coils are turned off; the motor stops.

END SELECT

GOTO start

END
```