

```
'-----Title-----
' File.....step_mot_one_revolution.pbp
' Started....2/14/09
' Microcontroller used:  Microchip Technology 16F88
'                          microchip.com
' PicBasic Pro Code: micro-Engineering Labs, Inc.
'                          melabs.com
' Stepper Motor Used:  Jameco #237623
' (#237623 - 4.8V, 1500mA, 1.8 Degree Step Angle or 200 Steps/Revolution)
'-----Program Description-----
' Program drives stepper motor to rotate one revolution clockwise in 1 sec
' then one revolution counter-clockwise in 1 sec then to stop.
'-----Schematic-----
' See schematic at:
'
' http://www.cornerstonerobotics.org/schematics/pic\_programming\_step\_mot1.pdf
'-----Related Lesson-----
' See the lesson Stepper Motor Control with a PIC at:
' http://www.cornerstonerobotics.org/curriculum/lessons\_year2/erii\_stepper\_motor.pdf
' Lesson also includes a section on how to figure out how to hook
' up a stepper motor with six leads when a data sheet for the
' motor is unavailable.
'-----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.
'-----New PicBasic Pro Command-----
' The PicBasic Pro Compiler Manual is on line at:
' http://www.microengineeringlabs.com/resources/index.htm#Manuals
' LOOKUP Index,[Constant{,Constant...}],Var
'
' The LOOKUP statement can be used to retrieve values from a table of 8-
' bit constants.  If Index is zero, Var is set to the value of the first
' Constant.  If Index is one, Var is set to the value of the second
' Constant.  And so on.  If Index is greater than or equal to the number
' of entries in the constant list, Var remains unchanged.
' Look around page 102 in the PicBasic Pro Compiler Manual
'-----PIC Connections-----
'
'          PIC16F88 Pin          Wiring
'          -----          -----
```

---

```

'          RB0          Stepper Motor Control Wire 1
'          RB1          Stepper Motor Control Wire 2
'          RB2          Stepper Motor Control Wire 3
'          RB3          Stepper Motor Control Wire 4
'          Vdd          +5 V
'          Vss          Ground
'          MCLR         4.7K Resistor to +5 V

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

i          VAR          BYTE      ' BYTE for counter variable, i
Delay      VAR          WORD      ' WORD for variable Delay
PulseSeq   VAR          BYTE      ' Motor Pulse Sequence (1,2,3,4,1,..)
                                     ' or (4,3,2,1,4,..)

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

TRISB = %00000000      ' Sets all PortB pins to output

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

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

PulseSeq = 0           ' Sets PulseSeq to 0

' Rotates stepper motor 1 revolution in 1 second CW then CCW

Delay = 5              ' 1sec / 200 steps = 5ms delay

' Clockwise rotation:

FOR i = 1 TO 200      ' For Jameco #237623 stepper motor,
                       ' 200 steps = 1 rev
    GOSUB CW_StepMtr
NEXT i

PAUSE 500             ' Pause 500 ms before CCW rotation begins

' Counter-clockwise rotation:

FOR i = 1 TO 200      ' For Jameco #237623 stepper motor,
                       ' 200 steps = 1 rev
    GOSUB CCW_StepMtr
NEXT i

PORTB = 0             ' Equivalent to PORTB = %00000000
                       ' All coils are turned off.

END

CW_StepMtr:           ' CW_StepMtr subroutine

```

---

---

```
IF PulseSeq >= 4 THEN

    PulseSeq = 1          ' Restart the 1,2,3,4 sequence at 1

ELSE

    PulseSeq = PulseSeq + 1
                        ' 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          ' Pause 5ms

RETURN

CCW_StepMtr:          ' CCW_StepMtr subroutine

IF PulseSeq <= 1 THEN

    PulseSeq = 4      ' Restart the 4,3,2,1 sequence at 4

ELSE

    PulseSeq = PulseSeq - 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          ' Pause 5ms

RETURN
```

