

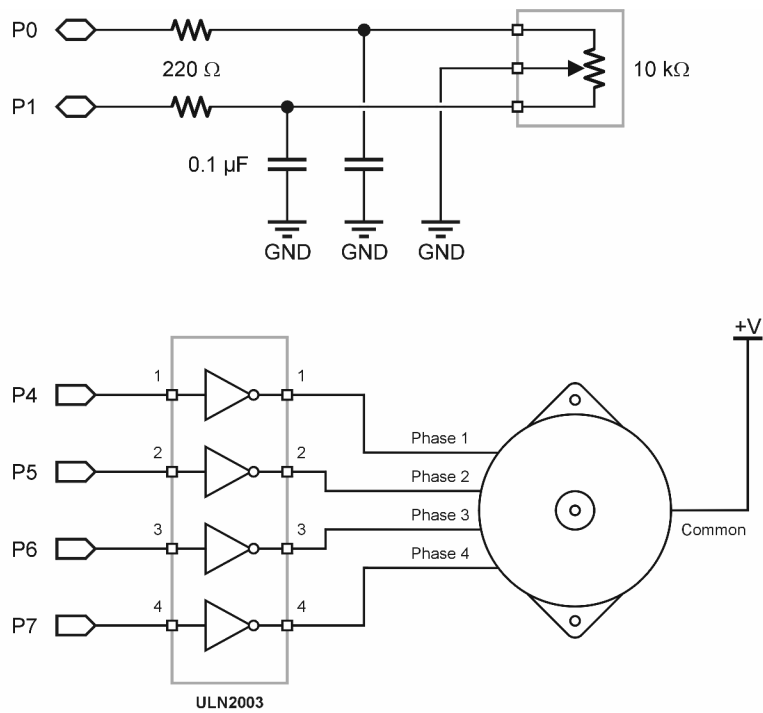
Experiment #26: Stepper Motor Control

This experiment demonstrates the control of a small 12-volt unipolar stepper motor. Stepper motors are used as precision positioning devices in robotics and industrial control applications.

New PBASIC Elements and Commands to Know:

- ABS

Building the Circuit



Motor Connections

Use the table below when connecting your stepper motor to the driver circuit.

Manufacturer	Mitsumi ¹	Howard Industries ²
Degrees per Step	7.5	3.6
Steps per Revolution	48	100
Phase 1	Black	Brown
Phase 2	Orange	Green
Phase 3	Brown	Red
Phase 4	Yellow	White
Common	Red	Black

¹ Current StampWorks motor supplied as PN #27964

² Motor originally supplied with StampWorks kit

Source Code

```
' =====  
'  
' File..... Ex26 - Stepper.BS2  
' Purpose... Stepper Motor Control  
' Author.... (C) Parallax, Inc., All Rights Reserved  
' E-mail.... support@parallax.com  
' Started...  
' Updated... 01 SEP 2004  
'  
' {$STAMP BS2}  
' {$PBASIC 2.5}  
'  
' =====  
'  
' -----  
' Program Description  
' -----  
'  
' This program demonstrates unipolar stepper motor control. The pot allows the  
' program to control speed and direction of the motor.  
'  
' -----  
' Revision History  
' -----  
'  
' -----  
' I/O Definitions  
' -----  
  
PotCW          PIN    0          ' clockwise pot input  
PotCCW         PIN    1          ' counter-clockwise pot input  
Coils          VAR    OUTB       ' output to stepper coils
```

```

' -----
' Constants
' -----

#SELECT $STAMP
#CASE BS2, BS2E, BS2PE
  Scale      CON      $100          ' x 1.00
#CASE BS2SX
  Scale      CON      $066          ' x 0.40
#CASE BS2P
  Scale      CON      $060          ' x 0.375
#ENDSELECT

Mitsumi      CON      48            ' 7.5 degrees per step
Howard       CON      100           ' 3.6 degrees per step

StpsPerRev   CON      Mitsumi       ' set motor type

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

speed        VAR      Word          ' delay between steps
idx          VAR      Byte          ' loop counter
stpIdx       VAR      Byte          ' step index (for EE table)
rcRt         VAR      Word          ' rc reading - right
rcLf         VAR      Word          ' rc reading - left
diff         VAR      Word          ' difference between readings

' -----
' EEPROM Data
' -----
'
'           4321          ' phase connections
' -----
Step1        DATA    %0011
Step2        DATA    %0110
Step3        DATA    %1100
Step4        DATA    %1001

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

Setup:
  DIRB = %1111          ' make stepper pins outputs
  speed = 15           ' set starting speed

' -----
' Program Code
' -----

Main:
  FOR idx = 1 TO StpsPerRev      ' 1 revolution forward
    GOSUB Step_Fwd

```

```

NEXT
PAUSE 500

FOR idx = 1 TO StpsPerRev          ' 1 revolution back
  GOSUB Step_Rev
NEXT
PAUSE 500

Step_Demo:
HIGH PotCW                        ' discharge caps
HIGH PotCCW
PAUSE 1
RCTIME PotCW, 1, rcRt              ' read clockwise
RCTIME PotCCW, 1, rcLf            ' read counter-clockwise

rcRt = (rcRt */ Scale) MAX 600    ' set speed limits
rcLf = (rcLf */ Scale) MAX 600
diff = ABS(rcRt - rcLf)           ' get difference

IF (diff >= 25) THEN              ' allow for dead band
  IF (rcLf < rcRt) THEN
    speed = 60 - (rcRt / 10) MIN 2 ' calculate speed
    GOSUB Step_Fwd                ' step forward
  ELSE
    speed = 60 - (rcLf / 10) MIN 2
    GOSUB Step_Rev                ' step reverse
  ENDIF
ENDIF

GOTO Step_Demo

' -----
' Subroutines
' -----

Step_Fwd:
stpIdx = stpIdx + 1 // 4          ' point to next step
READ (Step1 + stpIdx), Coils      ' output step data
PAUSE speed                       ' pause between steps
RETURN

Step_Rev:
stpIdx = stpIdx + 3 // 4          ' point to previous step
READ (Step1 + stpIdx), Coils
PAUSE speed
RETURN

```

Behind the Scenes

Stepper motors differ from standard DC motors in that they do not spin freely when power is applied. For a stepper motor to rotate, the power source must be continuously pulsed in specific patterns. The step sequence (pattern) determines the direction of the stepper's rotation. The time between sequence steps determines the rotational speed. Each step causes the stepper motor to rotate a fixed angular increment. The stepper motor supplied with the StampWorks kit rotates 7.5 degrees per step. This means that one full rotation (360 degrees) of the stepper requires 48 steps (100 with the Howard motor).

The step sequences for the motor are stored in `DATA` statements. The `step_Fwd` subroutine will read the next sequence from the table to be applied to the coils. The `step_Rev` subroutine is identical except that it will read the previous step. Note the trick with the modulus (`//`) operator used in `step_Rev`. By adding the maximum value of the sequence to the current value and then applying the modulus operator, the sequence goes in reverse. Here's the math:

```
0 + 3 // 4 = 3
3 + 3 // 4 = 2
2 + 3 // 4 = 1
1 + 3 // 4 = 0
```

This experiment reads both sides of the 10 k Ω potentiometer to determine its relative position. The differential value between the two readings is kept positive by using the `ABS` function. The position is used to determine the rotational direction and the strength of the position is used to determine the rotational speed. Remember, the shorter the delay between steps, the faster the stepper will rotate. A dead-band check is used to cause the motor to stop rotating when the `RC TIME` readings are nearly equal.

Challenge

Rewrite the program to run the motor in half steps (96 for Mitsumi motor, 200 for Howard motor). Here's the step sequence:

```
Step1 = %0001
Step2 = %0011
Step3 = %0010
Step4 = %0110
Step5 = %0100
Step6 = %1100
Step7 = %1000
Step8 = %1001
```