

```

1  {$STAMP BS2}
2  ' {$PBASIC 2.5}
3
4  ' -----[Program Description]-----
5  ' basic stepper motor control using a 10k POT, and built in 4-phase motor controller on PDB
6  '
7  ' -----
8
9  ' -----[I/O Definitions]-----
10
11 PotCW          PIN      0          ' clockwise POT input
12 PotCCW        PIN      1          ' counter-clockwise POT input
13 Coils          VAR      OUTB       ' output to stepper coils
14
15
16 ' -----[Constants]-----
17
18 Scale          CON      $100       ' to scale RCTIME
19
20 Mitsumi        CON      48         ' steps / revs by type
21 Howard         CON      100        ' " " " "
22
23 RevSteps       CON      Mitsumi    ' steps per revolution
24 NumSteps       CON      4          ' use 4-step sequence
25 LastStep       CON      NumSteps - 1 ' last step in sequence
26
27 #DEFINE Testing = 0
28
29
30 ' -----[Variables]-----
31
32 idx            VAR      Byte       ' loop counter
33 stpIdx         VAR      Nib        ' step pointer
34 stpDelay       VAR      Byte       ' delay for speed control
35
36 rcRt          VAR      Word       ' rc reading - right
37 rcLf          VAR      Word       ' rc reading - left
38 diff          VAR      Word       ' difference in readings
39
40
41 ' -----[EEPROM Data]-----
42
43 '
44 '
45
46 Step1         DATA    %1100
47 Step2         DATA    %0110
48 Step3         DATA    %0011
49 Step4         DATA    %1001
50
51
52 ' -----[Initialization]-----
53
54 Setup:
55   DIRB = %1111          ' make P4..P7 outputs
56   stpDelay = 5         ' set step delay
57
58
59 ' -----[Program Code]-----
60
61 Demo:
62   FOR idx = 1 TO RevSteps      ' 1 rev forward
63     GOSUB Step_Fwd
64   NEXT
65   PAUSE 200
66

```

```

67   FOR idx = 1 TO RevSteps           ' 1 rev back
68   GOSUB Step_Rev
69   NEXT
70   PAUSE 200
71
72 Main:
73   HIGH PotCW                       ' read clockwise position
74   PAUSE 1
75   RCTIME PotCW, 1, rcRt
76
77   HIGH PotCCW                       ' read ccw position
78   PAUSE 1
79   RCTIME PotCCW, 1, rcLf
80
81   rcRt = (rcRt */ Scale) MAX 600    ' set speed limit
82   rcLf = (rcLf */ Scale) MAX 600
83   diff = ABS(rcRt - rcLf)          ' get differences
84   stpDelay = 100 - (diff / 6) MIN 2 ' calculate step delay
85
86   IF (diff < 25) THEN              ' allow for dead band
87     GOTO Main
88   ELSE                               ' do a step
89     IF (rcLf < rcRt) THEN
90       GOSUB Step_Fwd
91     ELSE
92       GOSUB Step_Rev
93     ENDIF
94   ENDIF
95
96   GOTO Main                         ' repeat
97
98 ' -----[Sub-Routines]-----
99
100 ' turn stepper clockwise one full step
101
102 Step_Fwd:
103   stpIdx = stpIdx + 1 // NumSteps   ' point to next step
104   GOTO Do_Step
105
106
107 ' turn stepper counter-clockwise one full step
108
109 Step_Rev:
110   stpIdx = stpIdx + LastStep // NumSteps ' point to previous step
111   GOTO Do_Step
112
113 ' read new step data and output to pins
114
115 Do_Step:
116   READ (Step1 + stpIdx), Coils      ' output new coil data
117   PAUSE stpDelay                   ' pause between steps
118   RETURN
119

```