

The following is the full program:

```
\ This software assumes that the Power I/O Wildcard is mounted at on Wildcard
\ Port 1 and its address jumpers are set to 0, providing a Wildcard address of
\ 04. A unipolar stepper motor and power supply are connected to the lower
\ four of the Wildcard's output lines, connecting to the field connector as follows:
\ Field Pin   Field Signal Name   Motor Connection
\    16           Out 0           Brown (Phase A )
\    14           Out 1           Red   (Phase B )
\    12           Out 2           Yellow(Phase A')
\    10           Out 3           Blue  (Phase B')
\    17           V+Field         White (Winding A center tap)
\    17           V+Field         Black (Winding B center tap)
\    17           V+Field         +12 V Field Supply
\    9            Out GND         Field Supply GND Return
```

ANEW StepperController

HEX

```
C000 04 XCONSTANT PwrOutput \ The address of the Power I/O Wildcard output byte
\ with Wildcard mounted on Wildcard Port 1 with its address jumpers set to 0.
```

```
8 CONSTANT NumSteps           \ Number of steps in the stepper cycle
NumSteps 1- CONSTANT NumSteps-1 \ Number of steps - 1
```

```
\ Create a constant array to hold step waveforms in the cycle as bytes,
\ only lower 4 bits are valid, one bit for each motor winding (aka 'phase')
NumSteps 1 1 DIM.CONSTANT.ARRAY: Cycle \ 1-dim array of bytes
\ and store steps for a 4-phase 8-step cycle for 400 half-steps per revolution
01 0 Cycle C!  03 1 Cycle C!  02 2 Cycle C!  06 3 Cycle C!
04 4 Cycle C!  0C 5 Cycle C!  08 6 Cycle C!  09 7 Cycle C!
```

VARIABLE CycleIndex \ Holds index into the Cycle array

DECIMAL

```
: StepperOff ( -- ) 0 PwrOutput C! ;

: TakeStep ( flag -- ) \ move clockwise one step on true, ccw on false
\ Takes about 460 usec to execute
IF \ We increment or decrement CycleIndex forcing it to be within
\ range, so it never needs initialization
  CycleIndex @ 1+ DUP NumSteps-1 U> IF DROP 0 ENDIF
ELSE
  CycleIndex @ DUP 0 = IF DROP NumSteps ENDIF 1-
ENDIF
\ and fetch the byte from the array and output it to the Power I/O Wildcard
DUP CycleIndex ! Cycle C@ PwrOutput C!
;

: Steps ( u1 \ u2 \ flag -- )
\ moves u1 steps at a rate of u2 steps/sec, clockwise if flag is true,
\ ccw otherwise. u2 is clamped to between 16 and 2000 steps per second
Locals{ &flag &Steps/sec &Steps }
\ Calculate the per loop delay time needed to implement &Steps/sec
\ The 487. is the loop execution time in usecs found by benchmarking
1E6 &Steps/sec 16 MAX 2000 MIN FLOT F/ 487. F- FIXX TO &Steps/sec
&Steps 0
DO
  &Steps/sec MICROSEC.DELAY
  &flag TakeStep
LOOP
;
```

The information provided herein is believed to be reliable; however, Mosaic Industries assumes no responsibility for inaccuracies or omissions. Mosaic Industries assumes no responsibility for the use of this information and all use of such information shall be entirely at the user's own risk.

Mosaic Industries