

APPLICATIONS INFORMATION

Technical Paper
STP 99-11B

'SMART', CONSTANT-TORQUE, MICROSTEPPING DRIVE CIRCUITRY FOR UNDER \$10

by Paul Emerald and Ken Kadrmas

FOREWORD

Initially, (nearly) all step motor systems utilized simple, unipolar, constant-voltage drives; bipolar and/or PWM, constant-current circuitry was then deemed as too expensive to implement (power transistors were then large, expensive 'metal cans'). Later, the advent of integrated 'glue' logic (gates, flip-flops, etc.), and the continuing, relentless decline in semiconductor costs, precipitated an inexorable conversion from the venerable, low performance unipolar to two-phase, bipolar activation with its higher torque (at low-to-modest stepping rates).

Subsequently, new, higher performance designs began exploiting PWM (pulse-width-modulated), constant-current drives; and, also, stimulated the escalating utilization of fractional stepping modes. Even today, many microstepping drives are large, 'boxes' or 'boards' commanding prices that greatly exceed the cost of (even) the larger step motors; thus, many designers (especially 'nonelectronic' engineers) are often faced with very formidable decisions between drive system performance and the cost objectives. Step motors have become a low-cost, commodity item (particularly the small 'canstack' types); therefore, justifying 'hundreds' of dollars for microstepping electronics represents an extremely contrary circumstance for many of today's motion control systems designers.

INTRODUCTION

Recently, the emergence of microstepping power ICs has (greatly) alleviated the cost dilemma for many high-volume OEM applications. However, many other (usually smaller) organizations do not possess the expertise to develop the microcontroller, microprocessor, or computer-based control that is essential to utilizing these microstepping power ICs. Therefore, this set of users often face the tough decision of whether to purchase expensive 'boxes' or 'boards' vs the prolonged, costly effort involved in developing the electronics (this often

precipitates retaining consultants to define or create the system hardware and software).

This paper discusses the very practical application of bipolar and unipolar microstepping driver ICs and multi-chip modules and the related controller circuitry that makes designing with such devices easy, inexpensive, and productive. A 'reference' design technique is integral to the implementation of a system that includes the power circuitry, a low-cost, 8-bit, preprogrammed microcontroller and the other components needed to complete the control 'hardware'. The preprogrammed, 8-bit, microcontroller eliminates any need for software development; and, thereby, expedites the product creation and hastens the time to market.

The reference design can be utilized directly or integrated into a larger PCB. A further benefit is the compactness of the circuit layout. The power ratings available are: 50 V and ± 1.5 A (bipolar) and 46 V (unipolar) and either 1.25 A or 3 A. All of the reference designs support stepping formats that include full-step, half-step, quarter-step, and 1/8th (microstepping) increments.

One example that illustrates this reference design concept is a 50 V, ± 1.5 A bipolar drive. This detailed design includes: two microstepping ICs, a serial-to-parallel logic IC, an 8-bit microcontroller, crystal, eight diodes, and the various passive discrete components (about 25) to complete the entire drive electronics for a two-phase stepping motor.

To ease and simplify the design effort, the user only provides the following signals: (a) direction, (b) stepping clock (8x the full-step frequency), (c) mode logic (three inputs determine the operation from full-, half-, quarter-, or 1/8th stepping), (d) reset input (initiates a 'detent' position), and (e) recirculation control (this allows establishing the percent of fast- vs slow-decay in the phase winding). However, an optional microcontroller program offers 'automatic' recirculation control. This eliminates

'SMART', CONSTANT-TORQUE, MICROSTEPPING DRIVE CIRCUITRY FOR UNDER \$10

the need for evaluating the impact of stepping rate vs the sinusoidal current profile.

Although recirculation control may provide slight improvements (i.e., lower current ripple, reduced motor heating [a few degrees], and diminish audible noise levels [minimal differences]); this entails an evaluation of the motor (and step frequencies) to determine the proper ratio of fast- and slow-decay. The benefits of 'tuning' the recirculation ratios are small, and the time and effort required can be considerable. Hence, the uninitiated user should opt for the automatic recirculation control, and avoid the (essentially) unnecessary activity.

PRELUDE: 'INTEGRATED' MICROSTEPPING

The 'missing-link' for 'turn-key' stepper systems has been an 'off-the-shelf' controller/sequencer. However, this design issue has been resolved via programming an 8-bit microcontroller to deliver the logic signals needed to operate two low-cost, microstepping ICs. The PWM output currents are regulated (in this example) by sequencing the 3-bit DAC (digital-to-analog) inputs that control the current ratios for the two motor phases. The motor phases are driven by ratioed PWM currents that (digitally) emulate a cosine/sine relationship as illustrated in Figure 1. The winding currents are 90° out-of-phase (i.e., in quadrature), and the motors' rotational direction is determined by the phase relationship of the A vs B currents.

The cosine/sine relationships of the phase winding currents induce intermediate rotor positions which are between the usual full-step alignments. With the 1.8° PM hybrid stepper there are 200 full-step positions per shaft revolution, but this can be further subdivided via microstepping. The 360° electrical cycle of four full-steps is illustrated in Figure 2A; half-stepping in Figure 2B; quarter-stepping in Figure 2C; and microstepping in Figure 2D.

The microstepping power ICs used as an example are controlled by three digital-to-analog-conversion (DAC) inputs. Table 1 lists the ratioed currents that induce fractional subdivisions of one full step. Note: the correct mathematical calculations for microsteps #1 and #7 are: Phase A = 98.1% and Phase B = 19.5% (#1), and the

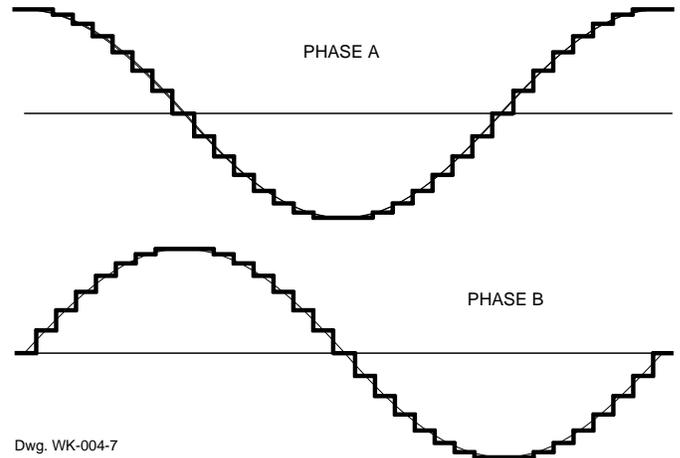


Figure 1: Current waveforms (1/8th step)

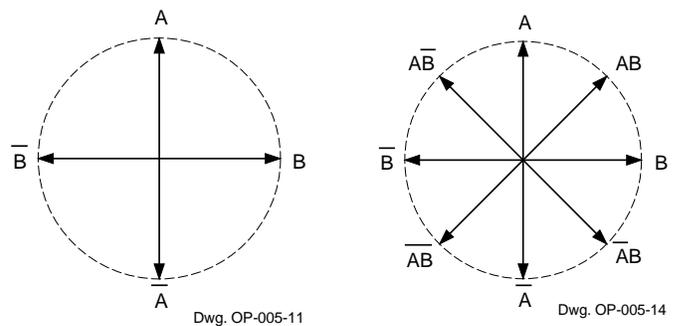


Figure 2A: Single-phase, full step Figure 2B: Half step

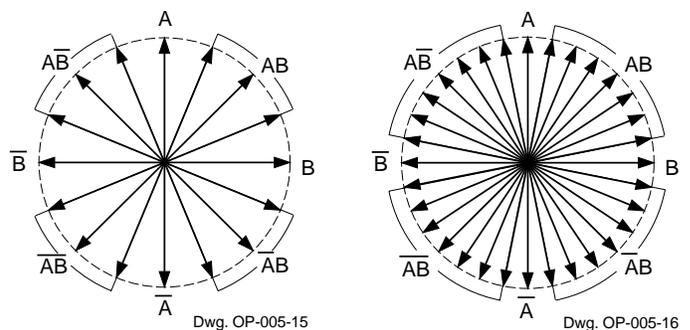


Figure 2C: 1/4 Step

Figure 2D: 1/8th Step

‘SMART’, CONSTANT-TORQUE, MICROSTEPPING DRIVE CIRCUITRY FOR UNDER \$10

reverse for #7 (19.5%; 98.1%). However, the ‘non-linear’ DAC tolerance is $\pm 3\%$; hence, this ‘9th’ current level (98.1%) was deemed as unnecessary since it is within the $\pm 3\%$ window. Further, the step motor positional accuracies are also $\leq \pm 3\%$ (for most PM hybrid motors, while most ‘canstack’ steppers are $> \pm 5\%$); therefore, the 98.1% current ratio is basically unnecessary for 1/8th-step operation, and this ‘9th’ current value is circumvented.

Table 1: 1/8th-Step subdivision current ratios

Step Series	Current Phase A	Current Phase B	Step Position
0	100.0%	0.0%	full step
1	100.0%*	19.5%	1/8 step
2	92.4%	38.2%	1/4 step
3	83.1%	55.5%	3/8 step
4	70.7%	70.7%	1/2 step
5	55.5%	83.1%	5/8 step
6	38.2%	92.4%	3/4 step
7	19.5%	100.0%	7/8 step
8	0.0%	100.0%	full step
x†	100.0%	100.0%	1/2 step

* Calculated (cosine) value $\approx 98.1\%$

† 2-Phase drive at 100% $\approx 141\%$ torque.

Similarly, the first subdivision with a 1/16th step format calculates to a 99.5% ratio for Phase A. Based upon the $\pm 3\%$ DAC tolerance and the step motor accuracy ($\approx \pm 3\%$), this 17th current ratio is also deemed as needless for low-cost, simple, open-loop, microstepping designs.

The non-linear DAC inputs determine the current ratios in each phase winding, and the direction of current is controlled by the logic level on another input (Phase). The magnitude of winding current is regulated by a current-sensing resistor, voltage reference, and a comparator in each power IC.

Another integrated function that can be utilized in microstepping applications is recirculation control. This provides ‘mixed-mode’ decay of the winding current and allows ‘tuning’ the percentage of fast decay in the coil with the stepping rates, etc. to enhance motor performance. However, this adds (yet) another level of com-

plexity (plus the added time and cost) to the development of the system software. The proper application of recirculation control allows the decaying current to effectively track the sinusoidal PWM waveform of Figure 1.

Without some measure of fast-decay during each descending half-cycle, the motor current is very susceptible to (serious) distortion as the winding current is diminishing toward zero. The effects of improper decay are: added heating (of the motor and circuitry), heightened audible noise levels, plus increased velocity ripple of the rotating shaft. A further complication of using mixed mode (fast- and slow-decay) is its directional dependency.

Having looked inward from the motor and driver circuitry, the various facets of a preprogrammed microcontroller can now be addressed. Clearly, an inexpensive, uncomplicated answer (small and a minimal parts count) is very beneficial to most systems designers. Additionally, many of today’s motion-control designs are being attempted by a growing, new constituency (i.e., ‘nonelectronic’ designers. This may include mechanical engineers, software programmers, etc.). Hence, a mounting interest in very basic, ‘turn-key’ solutions that escape the rising sophistication of today’s costly, high-performance motion systems.

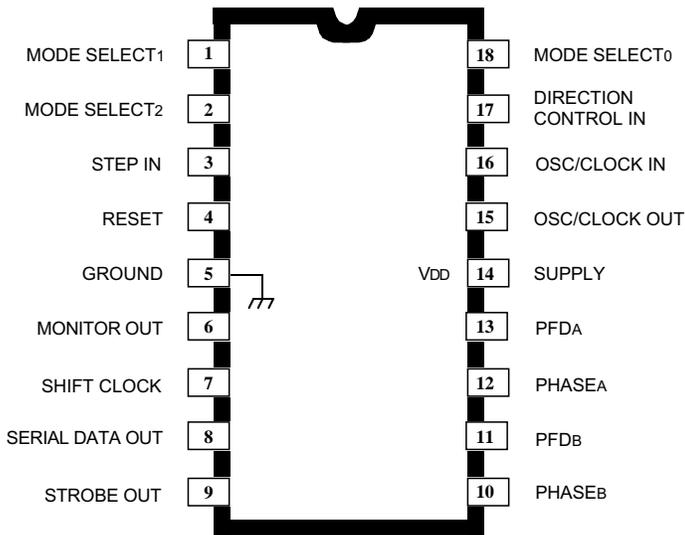
A very low-cost, serial, 8-bit, ‘preprogrammed’ microcontroller (Figure 3) is a pivotal component used in a new design alternative that exploits inexpensive, microstepping power ICs over the costly ‘boards’ and ‘boxes’ frequently used in step-motor systems applications. The projected hardware costs for this (reference) design are $< \$10$ for components (5 k to 10 k systems annually).

MICROCONTROLLER OPERATION

Although ‘hardware’ control of the microstepping ICs is feasible, without a specific (ASIC), monolithic IC controller the prime solution becomes a ‘software’ option. From the user’s perspective, a ‘preprogrammed’ microcontroller appears little, or no, different than a ‘dedicated’ controller and sequencer IC expressly created for microstepping applications of the two power ICs. Further, the flexibility of a software-based drive is cer-

'SMART', CONSTANT-TORQUE, MICROSTEPPING DRIVE CIRCUITRY FOR UNDER \$10

tainly a basic benefit (high-volume production of 8-bit microcontrollers transposes to low-cost circuitry).



Dwg. PP-071A

Figure 3: Microcontroller IC connections

As an indicator of the logic signals needed to control the power ICs, Table 2 lists the A3955 inputs to the 3-bit DAC for 1/8th-step operation. These I/O signals are serial data from the microcontroller, then converted to a parallel mode by a 74HC595 as the 'interface' between the microcontroller and the two microstepping power ICs.

The complexities of control and sequencing should be very evident from Table 3. This only involves four full-steps (the 360° electrical cycle shown in Figure 1), but should corroborate the value of the microcontroller operation of: six DAC input lines, two direction signals (θ), and two lookup tables, which control the PFD (percent-fast-decay) for both phase windings. Note: Table 3 corresponds with one direction (only); two 'lookup' tables are required to control the phase relationship of the PFD inputs (i.e., reversing direction involves transposing the A0 and A1 signals in Table 3; this inverts the slow- and fast-decay with direction).

Table 2: Digital-to-analog truth table

D ₂	D ₁	D ₀	DAC%
1	1	1	100.0
1	1	0	92.4
1	0	1	83.1
1	0	0	70.7
0	1	1	55.5
0	1	0	38.2
0	0	1	19.5
0	0	0	0

Table 3 incorporates an 'automatic' recirculation control concept into the 'lookup' tables. The user can disregard the necessity to: evaluate the motor current waveforms, eliminate adjusting the PFD ratio, and delete six passive components from the bill-of-material and PCB layout. This reference design is also suitable for utilizing the A3957 (and 1/16th stepping with very little modifications).

The versatility offered by software control allows the operating modes listed in Table 4. This table itemizes the various logic inputs that determine direction, stepping format, reset, 1/8th vs 1/16th sub-steps, etc. Note: during power-up, Pin #7 is sampled for a pull-up or pull-down resistor to establish the fractional step limit. A pull up sets up a 1/8th-step format (A3955) and pull down sets up 1/16th-operation for the A3957.¹

Table 3 incorporates an 'automatic' recirculation control concept into the 'lookup' tables. The user can disregard the necessity to: evaluate the motor's current waveforms, eliminate adjusting the PFD ratio, and delete six passive components from the bill-of-material and PCB layout. This reference design is also suitable for utilizing the A3957 (and 1/16th stepping with minimal modification). The longer truth table of the A3957 is not included.

Table 5 lists the microcontroller lead connections and provides the essence of the circuit operation (a schematic illustrating a typical stepper design follows in Figure 4). A brief description of the microcontroller I/O should clarify the connections of the various elements of the drive electronics.

***'SMART', CONSTANT-TORQUE,
MICROSTEPPING DRIVE
CIRCUITRY FOR UNDER \$10***

Table 3: Input logic sequence and operating modes for two-phase step motor

Driver IC #1 (Phase A)							Driver IC #2 (Phase B)							Step Motor		
θ	D ₂	D ₁	D ₀	PFD	Current	MO	θ	D ₂	D ₁	D ₀	PFD	Current	MO	Torque		$\Delta \angle$
														Value	Angle	
1	1	1	1	X	100.0%	X	0	0	0	0	X	0.0%	D	1.000	0.00	11.03
1	1	1	1	A0°	100.0%	F	1	0	0	1	A1	19.5%	S	1.019	11.03	11.03
1	1	1	0	A0°	92.4%	F	1	0	1	0	A1	38.2%	S	1.000	22.46	11.43
1	1	0	1	A0°	83.1%	F	1	0	1	1	A1	55.5%	S	0.999	33.74	11.28
1	1	0	0	A0°	70.7%	F	1	1	0	0	A1	70.7%	S	1.000	45.00	11.26
1	0	1	1	A0°	55.5%	F	1	1	0	1	A1	83.1%	S	0.999	56.26	11.26
1	0	1	0	A0°	38.2%	F	1	1	1	0	A1	92.4%	S	1.000	67.54	11.28
1	0	0	1	A0°	19.5%	F	1	1	1	1	A1	100.0%	S	1.019	78.97	11.43
0	0	0	0	X	0.0%	D	1	1	1	1	X	100.0%	X	1.000	90.00	11.03
0	0	0	1	A1	-19.5%	S	1	1	1	1	A0°	100.0%	F	1.019	101.03	11.03
0	0	1	0	A1	-38.2%	S	1	1	1	0	A0°	92.4%	F	1.000	112.46	11.43
0	0	1	1	A1	-55.5%	S	1	1	0	1	A0°	83.1%	F	0.999	123.74	11.28
0	1	0	0	A1	-70.7%	S	1	1	0	0	A0°	70.7%	F	1.000	135.00	11.26
0	1	0	1	A1	-83.1%	S	1	0	1	1	A0°	55.5%	F	0.999	146.26	11.26
0	1	1	0	A1	-92.4%	S	1	0	1	0	A0°	38.2%	F	1.000	157.54	11.28
0	1	1	1	A1	-100.0%	S	1	0	0	1	A0°	19.5%	F	1.019	168.97	11.43
0	1	1	1	X	-100.0%	X	0	0	0	0	X	0.0%	D	1.000	180.00	11.03
0	1	1	1	A0°	-100.0%	F	0	0	0	1	A1	-19.5%	S	1.019	191.03	11.03
0	1	1	0	A0°	-92.4%	F	0	0	1	0	A1	-38.2%	S	1.000	202.46	11.43
0	1	0	1	A0°	-83.1%	F	0	0	1	1	A1	-55.5%	S	0.999	213.74	11.28
0	1	0	0	A0°	-70.7%	F	0	1	0	0	A1	-70.7%	S	1.000	225.00	11.26
0	0	1	1	A0°	-55.5%	F	0	1	0	1	A1	-83.1%	S	0.999	236.26	11.26
0	0	1	0	A0°	-38.2%	F	0	1	1	0	A1	-92.4%	S	1.000	247.54	11.28
0	0	0	1	A0°	-19.5%	F	0	1	1	1	A1	-100.0%	S	1.019	258.97	11.43
1	0	0	0	X	0.0%	D	0	1	1	1	X	-100.0%	X	1.000	270.00	11.03
1	0	0	1	A1	19.5%	S	0	1	1	1	A0°	-100.0%	F	1.019	281.03	11.03
1	0	1	0	A1	38.2%	S	0	1	1	0	A0°	-92.4%	F	1.000	292.46	11.43
1	0	1	1	A1	55.5%	S	0	1	0	1	A0°	-83.1%	F	0.999	303.74	11.28
1	1	0	0	A1	70.7%	S	0	1	0	0	A0°	-70.7%	F	1.000	315.00	11.26
1	1	0	1	A1	83.1%	S	0	0	1	1	A0°	-55.5%	F	0.999	326.26	11.26
1	1	1	0	A1	92.4%	S	0	0	1	0	A0°	-38.3%	F	1.000	337.54	11.28
1	1	1	1	A1	100.0%	S	0	0	0	1	A0°	-19.5%	F	1.019	348.97	11.43
1	1	1	1	X	100.0%	X	1	0	0	0	X	0.0%	D	1.000	360.00	11.03

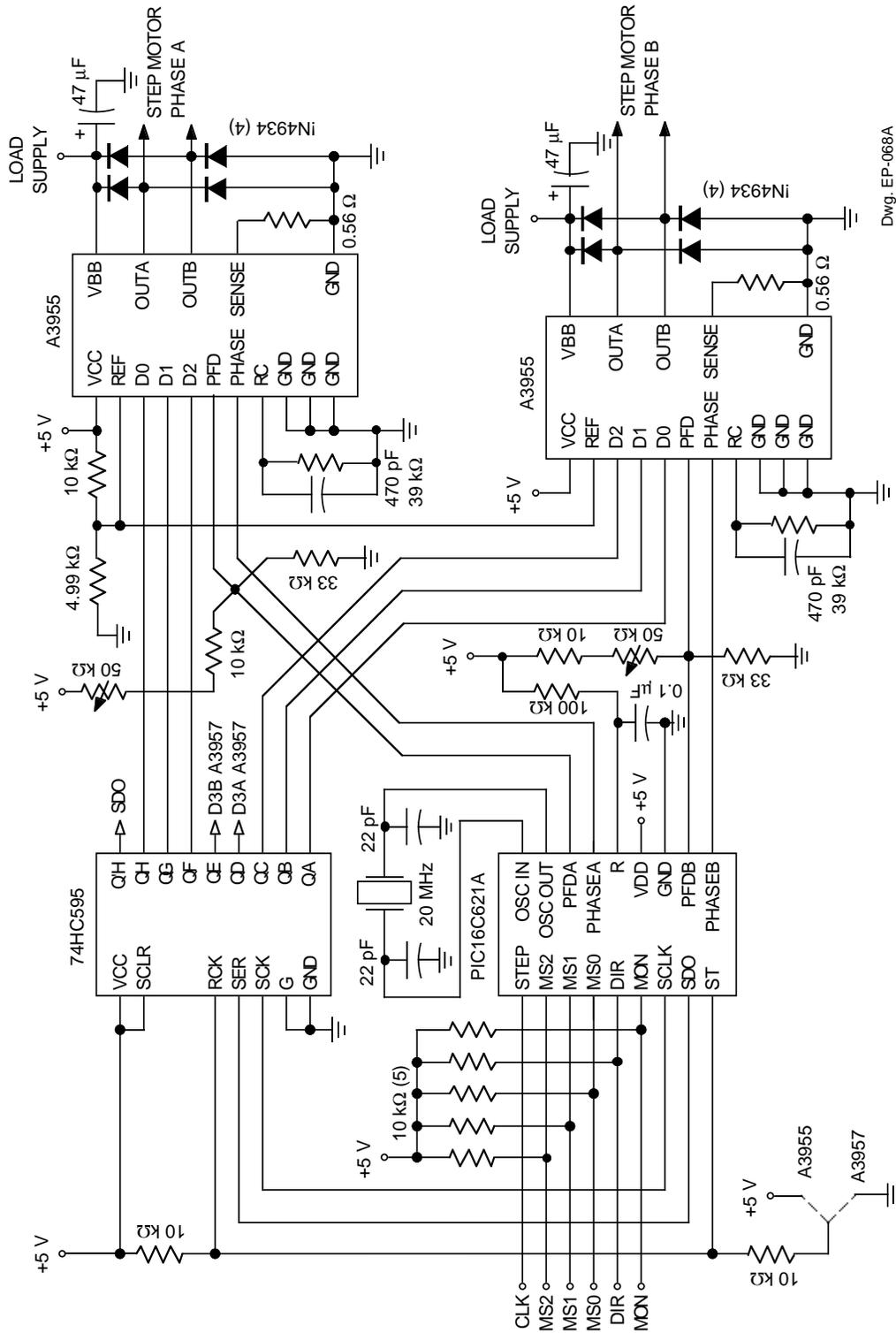
Input codes: θ = Phase input (direction), PFD = Percent fast decay, MO = Operating mode.

Operating modes: D = Disabled (output off), F = Fast decay, S = Slow decay,
X = Irrelevant, A0° = Automatic fast decay, A1 = Automatic slow decay.

Motor codes: Value = Torque vector magnitude, Angle = Rotor position, $\Delta \angle$ = Delta angle

NOTES: Logic sequencing and operating modes correspond to FOUR full steps.
Preprogramming of microcontroller provides automatic recirculation control.

**'SMART', CONSTANT-TORQUE,
MICROSTEPPING DRIVE
CIRCUITRY FOR UNDER \$10**



Dwg. EP-068A

Figure 4: Low-cost microstepping drive circuitry

***‘SMART’, CONSTANT-TORQUE,
MICROSTEPPING DRIVE
CIRCUITRY FOR UNDER \$10***

Table 4: Controller/sequencer IC operational logic

Binary inputs				Operating mode (Command executed on L → H of CLK)	Comments (Applicable power ICs)
DIR	MS2	MS1	MS0		
0	0	0	0	CW, Full step (single-phase)	A3955/57
0	0	0	1	CW, Half step (constant torque)	A3955/57
0	0	1	0	CW, 1/4 step (constant torque)	A3955/57
0	0	1	1	CW, 1/8th step (constant torque)	A3955/57
0	1	0	0	CW, 1/16th step (constant torque)	A3957 only
0	1	0	1	Disable A3955/57 holding torque ²	At present position
0	1	1	0	Enable A3955/57 holding torque	From present position
0	1	1	1	Reset PIC16C621A sequencer IC	A3955/57
1	0	0	0	CCW, full step (single-phase)	A3955/57
1	0	0	1	CCW, half step (constant torque)	A3955/57
1	0	1	0	CCW, 1/4 step (constant torque)	A3955/57
1	0	1	1	CCW, 1/8th step (constant torque)	A3955/57
1	1	0	0	CCW, 1/16th step (constant torque)	A3957 only
1	1	0	1	Disable A3955/57 holding torque ²	At present position
1	1	1	0	Enable A3955/57 holding torque	From present position
1	1	1	1	Reset PIC16C621A sequencer IC	A3955/57

Mode-select inputs (pins 1, 2, and 18)

These three inputs (MS2, MS1, and MS0) determine the stepping format, disable/enable motor power, and reset the controller/sequencer. In conjunction with the direction input, the mode inputs control the sixteen operating states listed in Table 4. Note: deactivating stepper power in any position except ‘detent’ (i.e., a single phase activated) results in the motor rotor advancing, or retracting, from its intermediate position and alignment with a natural (i.e., minimum-reluctance flux field) orientation. The absolute position may be affected by inertia, load, fractional position, ringing, etc. and cannot be determined without feedback. Phase currents must be maintained to immobilize the rotor/load in any intermediate position. ²

Step (clock) input (pin 3)

The controller stepping-clock frequency must be a multiple of the actual stepping rate. The A3955 requires a stepping-clock input frequency equal to eight times the intended stepping rate for the motor; the A3957 requires a stepping clock rate that is sixteen times the actual stepping rate of the motor. However, neither design necessitates that the step frequency be varied should the operating mode(s) be switched during operation. Shifting from one stepping format does not mandate a simultaneous (and equivalent) change in the clock frequency.

Using a 20 MHz crystal (maximum limit for the PIC16C621A) allows a 50 kHz stepping clock for the A3955, and this equates to 6 250 full steps per second (50 kHz/8). For the A3957 this 50 kHz stepping clock translates to 3 125 full steps per second (50 kHz/16). These frequencies represent the attainable limits with the PIC16C621A.

***'SMART', CONSTANT-TORQUE,
MICROSTEPPING DRIVE
CIRCUITRY FOR UNDER \$10***

Table 5: Controller/sequencer pin descriptions

Pin	Function	Description	Comments, connections, etc.
1	Input	Mode select 1	Static and/or dynamic control of stepping mode
2	Input	Mode select 2	Static and/or dynamic control of stepping mode
3	Input	Step in	Governs full-step rate (± 8 for A3955; ± 16 for A3957)
4	Input	Reset	Resets DIR., MS2, MS1, and MS0 to 0000 (detent)
5	Power	Ground	Power return
6	Output	Monitor out	Signals full-step rotor alignment (active low)
7	I/O	± 8 , ± 16 /Shift CLK	Pull up for A3955; pull down for A3957/shift CLK out
8	Output	Serial data out	Shifts 8-bit serial data to 74HC595 serial input
9	Output	Strobe out	Latches 8-bit data into 74HC595 (latch clock in)
10	Output	Phase B	Controls current direction in phase B (A3955/57)
11	Output	PFD B	Phase B recirculation control (A3955/57)
12	Output	Phase A	Controls current direction in phase A (A3955/57)
13	Output	PFD A	Phase A recirculation control (A3955/57)
14	Power	Supply	(V_{DD}) Recommended range: 4.5 V to 5.5 V
15	Output	Osc/clock out	Crystal oscillator connection
16	Input	Osc/clock in	Crystal oscillator connection/ext clock input
17	Input	Direction control in	Determines direction of step motor rotation
18	Input	Mode select 0	Static and/or dynamic control of stepping mode

Although not a necessity, a stepping clock with a 50% duty cycle represents the simplest technique for providing an appropriate (≤ 50 kHz) stepping clock rate. The step clock varies depending upon the start, acceleration, slewing, deceleration, and stop trajectories mandated by the motion system control and 'point-to-point' timing objectives.

Reset input (pin 4)

Per Table 4, the preprogrammed microcontroller incorporates two 'software' reset states that are serially loaded with MS2, MS1, and MS0 all 'high'. However, the direct 'hardware' reset is actuated with a logic '0' (i.e., 'active low') on this input. An input low level 'defaults' to the 0000 binary state in Table 4, and sets the rotor to its natural (or 'detent') position with one-phase energized.

Monitor output (pin 6)

An output 'low' signal indicates a rotor alignment corresponding to a single phase on position. Any changes in the operating mode (microstepping to full-step, etc.) should coincide with the interval that the monitor output is in the 'low' state'. This alleviates noise problems, excessive ringing, etc. that may result from changing the stepping modes 'on-the-fly'. 'Nonlinear' (such as 'S-curve') acceleration profiles should exploit this signal to achieve very smooth, quiet stepper operation.

‘SMART’, CONSTANT-TORQUE, MICROSTEPPING DRIVE CIRCUITRY FOR UNDER \$10

Shift clock output (pin 7)

This I/O pin serves a dual purpose. On power up, the microcontroller samples this terminal as an input. Per Table 5, connecting a pull-up resistor results in 1/8th-step format; while a pull-down resistor configures the controller for its 1/16th-step mode (A3957 only). Clearly, this provides versatility, simplicity, and cost-effectiveness for most users.

Operating in its output mode, this I/O constitutes the shift clock signal for the 74HC595. Data is transferred from the microcontroller serial-data output (pin 8) to the serial-data input of the 74HC595. This 8-bit serial format is converted into parallel signals controlling the 3-bit (or 4-bit) DAC input lines to the two microstepping ICs. Serial data from the serial data output is ‘valid’ on the low-to-high clock transitions and eight clock pulses shift serial control signals into the 74HC595. A basic timing diagram (showing serial data, shift clock, and strobe) is depicted in Figure 5. Signal timing is controlled by the preprogrammed microcontroller; data entered into the 74HC595 shift register is then latched by the low-to-high transition of the strobe input.

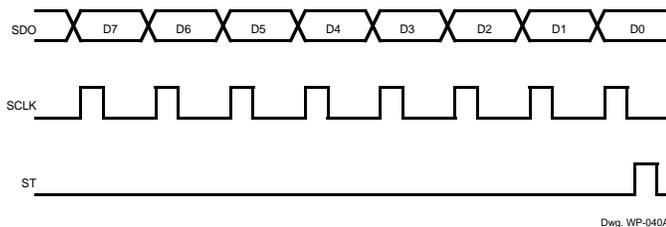


Figure 5: Serial data, shift clock, strobe

Serial data output (pin 8)

Table 4 enumerates the binary signal instructions that control each of the microstepping power ICs. The first 3-bits (or 4-bits) control the digital-to-analog conversion in one power IC, while the next 3 (or 4-bits) ratio the second driver current. The microcontroller monitors all the various static inputs (i.e., Direction, Mode Select(s), Reset), and by exploiting the Stepping Clock for its input frequency, transfers the 8-bit data commands to the driver ICs via the serial-to-parallel interface IC. Per Table 3, the

μcontroller utilizes ‘look-up’ tables to provide overall control of direction, stepping format, and recirculation mode (PFD). The μcontroller ‘reads’ inputs and (then) outputs time-based signals to control both microstepping ICs.

Strobe output (pin 9)

After the 8-bit serial data has been loaded into the shift register, a low-to-high transition on the strobe output transfers the serial data from the shift register into the eight ‘D’ flip-flops that compose the parallel-data outputs. This ‘latched’ data controls microstepping current ratios for both power ICs, and is ‘updated’ after eight step clocks.

Phase control outputs (pins 10 and 12)

Two of the microcontroller outputs are devoted to direction control; per Table 5, pin #10 furnishes signals for phase B, while pin #12 provides logic control of phase A. The phase signals follow the sequencing itemized in Table 3 to provide the 1/8th-step ‘sinusoidal’ waveforms as illustrated in Figure 1. Phase outputs (θ in Table 3) are parallel, and connected directly to the power IC inputs (A3955 in the schematic). Obviously, both phase signals must coincide with the 8-bit serial-to-parallel conversion signals that control the phase current ratios (D2, D1, and D0).

PFD control outputs (pins 11 and 13)

Another pair of microcontroller outputs constitute parallel, direct control of the recirculation paths during each PWM current cycle. The percentage of fast-decay (or four-quadrant) to slow-decay (or two-quadrant) recirculation determines how effectively the output current tracks a sinusoidal waveform. In Table 3, ‘mixed-mode’ operation was utilized for those portions of each microstepping cycle in which the current is decreasing (denoted by $A0^\circ$ in Table 3). The application of ‘mixed-mode’ decay allows the PWM current to properly decay during each fixed-off-time interval. An illustration of ‘improper’ (i.e., insufficient) decay is presented in Figure 6A, and a more effective ‘sinusoidal’ waveform is shown in Figure 6B.

**'SMART', CONSTANT-TORQUE,
MICROSTEPPING DRIVE
CIRCUITRY FOR UNDER \$10**

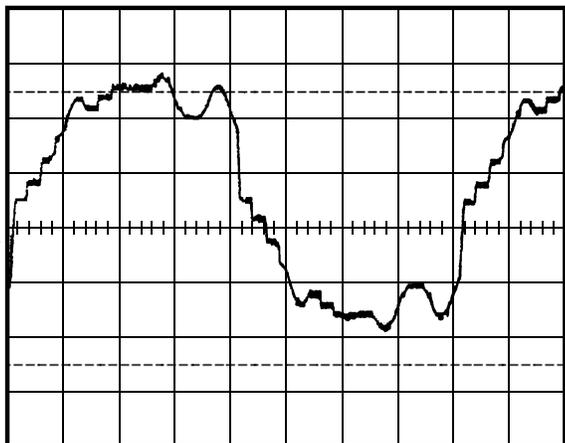


Figure 6A: 100 Hz, 1/8th step, 'slow decay'

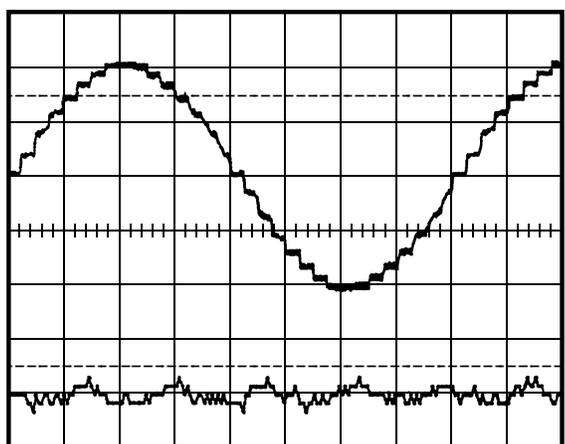


Figure 6B: 100 Hz, 1/8th step, 'mixed decay'

The circuit schematic (Figure 4) includes a pair of 49.9 k Ω variables and allows setting the ratios of fast- and slow-decay to match the individual characteristics of the step motor, step frequency, supply voltage, etc. As the stepping frequency is increased, the waveform anomalies become more pronounced, and the benefit of 'mixed mode' (or fast decay) also becomes very noticeable.

A comprehensive discussion of the 'mixed-mode' operation is included in Allegro Technical Paper STP 97-5A.

Another, more recent evaluation revealed that it is very feasible for designers to exploit automatic recirculation control. Tuning two potentiometers results in minimal benefits; thus, six components can be deleted from the circuit schematic which applies 'mixed-mode' recirculation. Although the cost impact is insignificant, omitting six passives (R₁, R₄, R₅, R₁₅, R₁₆, R₁₈) proves favorable.

The 'mystique' of recirculation control prompted an evaluation that revealed only a small (less than 5°C) motor temperature differential after realizing the thermal equilibrium (\approx 60 minutes of operation). This (and the waveform contrasts of Figures 7A, 7B, 7C, and 7D) establishes the validity of applying fully digital (automatic) recirculation control.

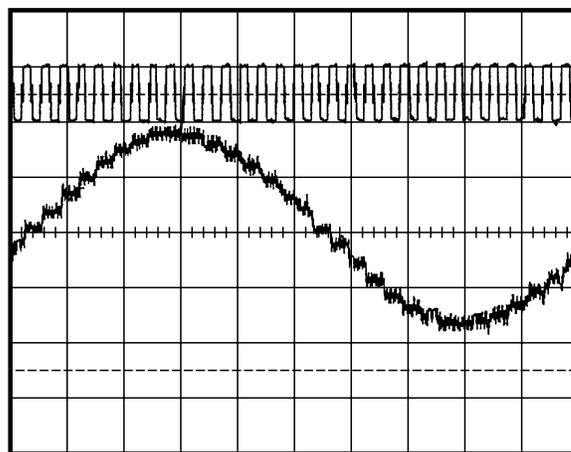


Figure 7A: 200 Hz Waveform (slow/mixed)

'SMART', CONSTANT-TORQUE, MICROSTEPPING DRIVE CIRCUITRY FOR UNDER \$10

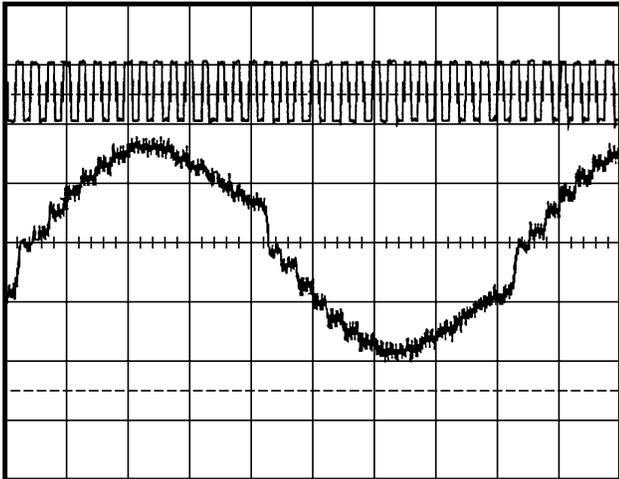


Figure 7B: 800 Hz Waveform (slow/mixed)

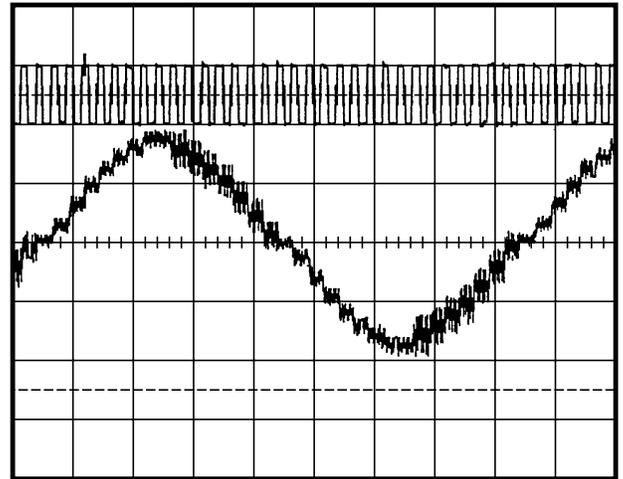


Figure 7D: 800 Hz Waveform (slow/fast)

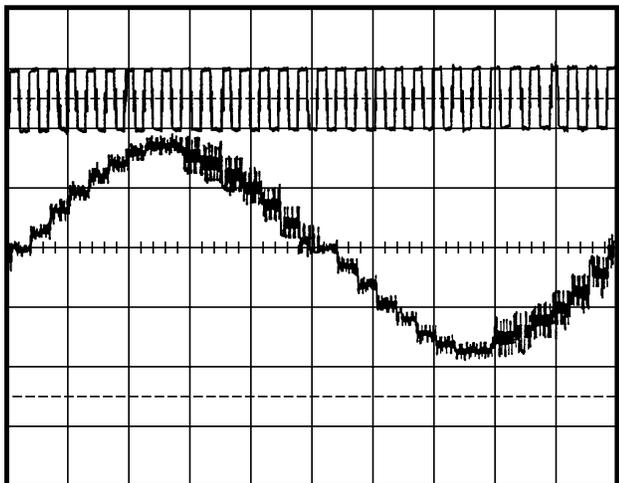


Figure 7C: 200 Hz Waveform (slow/fast)

Oscillatoroutput/clock output (pin 15)

This is one of the device connections used for the crystal oscillator (or ceramic resonator) that supplies the microcontroller with an oscillator/time base (≤ 20 MHz). The controller/sequencer may (also) use an external clock input (from a system computer, etc.). If so, this connection is left open and only the oscillator input is utilized.

Oscillator input/clock input (pin 16)

This device connection serves as one connection for a parallel-cut crystal (pin 15 is the other). In the schematic in Figure 4, two 20 pF capacitors are advised; and a series

resistor from pin 15) may be required if AT-cut crystals are utilized.

Also, this terminal is used if an external clock is utilized in lieu of a crystal or ceramic resonator. The 20 MHz frequency limit applies to using any external clock source (microprocessor, etc.).

Direction control input (pin 17)

This input controls the sequencing of the winding currents. Per Table 4, a 'low' induces clockwise motor rotation, while a 'high' results in counter-clockwise rotation of the shaft. Reversing direction may necessitate deceleration (perhaps full stop) to avoid problems associated with the load inertia causing the motor to overshoot the proposed stop/reverse position.

This closes the description of the microcontroller; but, because the PIC16C621A does not control the actual PWM phase currents, leaves an unresolved issue. The 3- or 4-bit data inputs determine the output current ratios, but do not set the phase currents. This is a separate aspect of the PWM control.

CONTROL of PHASE CURRENTS

Prior to establishing the technique for setting the PWM output current, a very consequential factor cannot be neglected. Customarily, the (nominal) motor currents are specified for two phases on. This refers to the 70.7%

'SMART', CONSTANT-TORQUE, MICROSTEPPING DRIVE CIRCUITRY FOR UNDER \$10

ratios listed in Table 1 and the (1/8th-step) constant torque curve shown in Figure 8. The single-phase on current, to induce the same torque, requires the peak value to be set to 1.4 times that of two-phases on (both on constitutes the rms value). Therefore, unless a motor is specified for unipolar drive, the PWM output current must correspond to 140% of rated current for two-phase operation. This results in the constant-torque curve illustrated in Figure 8, and induces very smooth, quiet motion, and rated torque. Disregarding this peak (vs rms) current setting results in significant torque loss.

Selecting standard resistor values to fit the PWM formula entails adjusting the voltage reference to the sensing resistor, vice versa, or iterations of both. Per the schematic, using the R2/R3 divider, with two 0.56 Ω current-sensing resistors; the single-phase on current value is approximately ±1.0 A per phase and corresponds to a 710 mA two-phase drive.

As mentioned, this value of phase current equates to two-phase operation at ≈710 mA/phase, and corresponds to the constant torque half-step loci in Figure 8. The integrated non-linear 3-bit (or 4-bit) DACs control the sine/cosine currents that induce the fractional step positions. An extension of the (peak) current formula provides the ratios.

$$I_{TRIP} = SRCR \times V_{REF} / (3 \times R_S)$$

The internal digital-to-analog converter reduces the reference input voltage to the current sensing comparator in increments as itemized in Tables 1 and 2. The eight step reference current ratios (SRCRs) induce the intermediate rotor loci which constitute fractional (or microstepping) increments.

UNIPOLAR MICROSTEPPING

Another low-cost microstepping alternative is the combination of a preprogrammed microcontroller and two power multi-chip modules created for microstepping. The unipolar drivers are specified for operation to 46 V (maximum); the MOSFET output current ratings are: 1.25 A (SLA7042M) and 3.0 A (SLA7044M). These power modules provide step formats that include: full, half, quarter, and 1/8th stepping. The 3.0 A rating satisfies many of the larger, higher-power motor applications that are (often) beyond other driver ratings. Both power devices offer programmable current control for unipolar designs; with NMOS outputs internal power (heating) is minimized and system cooling or heatsinking needs are alleviated.

The bipolar and unipolar driver reference designs are eminently suitable for the very swift creation of step-motor-based motion-control designs within the stepping cost solution to the control and sequencing aspects of applying integrated microstepping ICs. With driver

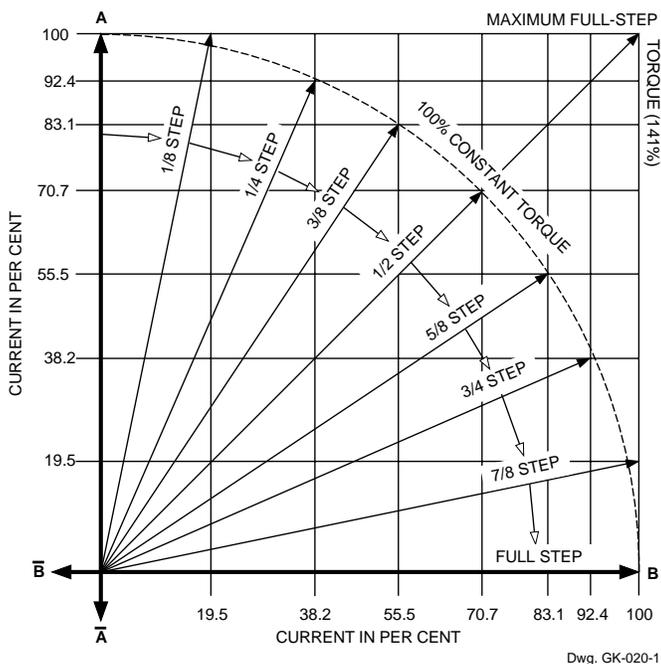


Figure 8: Constant torque vectors (1/8th step)

The driver ICs regulate the winding currents, and the user establishes the design value via a stable voltage reference and a current-sensing resistor for each winding (R₈ and R₁₉, Figure 4). The single-phase on value (100% in Figure 8) is calculated based upon the following formula:

$$I_{TRIP} = V_{REF} / (3 \times R_S)$$

'SMART', CONSTANT-TORQUE, MICROSTEPPING DRIVE CIRCUITRY FOR UNDER \$10

capabilities to ≤ 50 V and ± 1.5 A, a design can be implemented with only four ICs and various discrete components. Currents exceeding ≈ 1.0 A should incorporate external free-wheeling diodes (1N4934s in the schematic) to reduce the power IC dissipation and heating via shunting the recirculation currents through the outboard diodes.

The preprogrammed microcontroller relieves designers of the necessity to develop both the hardware and software to implement expensive, complex motion control. Of special interest for small-to-modest volume systems applications, a 'turn-key' design can save considerable development time and cost. Some volume users may (also) opt for exploiting this hardware design to alleviate the necessity of burdening staff and schedules to develop a system that might (virtually) become a duplicate.

Motion-systems designers are now offered (yet another alternative between developing the system (themselves) from the component level and high-cost, packaged, 'board-level' hardware that may embody performance and technology 'overkill' for today's step-motor applications. The relative cost of utilizing this technique could easily translate to an 'order-of-magnitude' difference (i.e., lower) in the 'cost-per-axis' for motion control systems.

Hence, the dominant advantages of utilizing these reference designs for stepper control are cost and the simplicity of and speed of implementation.

REFERENCES

¹Emerald, P.; The Merits, the Mystique, and the Myths of Microstepping (Part 2); PCIM '99 Europe; Nürnberg, Germany; June 22-24, 1999

Emerald, P. and Kadrmas, K.; The Merits, the Mystique, and the of Myths of Microstepping; AMD&C; PowerSystems World '98; Santa Clara, CA; pp 45-57

Emerald, P., Pickett, R., and Rowan, T.; Integrated 'Mixed-Mode' Microstepping Drivers Afford Smoother, Quieter Operation and Reduce Step Motor Heating; PCIM Europe '98; Nürnberg, Germany; May 1998

Emerald, P., Peppiette, R., and Seliverstov, A.; Monolithic, Programmable, Full-Bridge Motor Driver Integrates PWM Current Control & 'Mixed-Mode' Microstepping; PCIM '97 Hong Kong; October 1997; Allegro Paper STP 97-5A

Emerald, P.; Design Regimen for Resolving the Selection of Integrated Power Electronics for Step Motors; Rocky Mountain Technology Expo; Denver, CO; April 1997

²Emerald, P., Sasaki, M., and Takahashi, H.; CMOS Step Motor IC and Power Multi-Chip Module Combine to Integrate Versatile, Multi-Mode PWM Operation and Microstepping; PowerSystems World '96; Las Vegas, NV; September 7-13, 1996

Freeman, J.; Stepping Motor SpecsmanShip; Incremental Motion Control Systems and Devices Proceedings; June 1987; pp. 297-304

Leenhouts, A.; Step Motor System Design Handbook, 2nd Edition; Published by Litchfield Engineering; Kingman, Arizona; Copyright 1997

McCarthy, K.; Getting More Out of Microstepping; Motion Control; March 1991; pp. 12-16

Mazzola, T., and Floyd, R.; Intelligent Microsteppers for Stepper Motor Controls; Motion Control; September 1991; pp. 16-21

Raskin, C.; Designing with Motion Handbook, Chapter 7, Stepper Motion; pp 83-98

Viegnat, N.; Microstep Operation of Stepper Motors; Motion Control; March 1991; pp. 48-50

Step Motor and Servo Motor Systems and Controls; Compumotor Division of Parker Hannifin; 1996-1997 Edition, p. A29

Thinkscap® 5; Stepper Motor Drives, Technical Publication from Portescap; publication date unknown.

This paper was presented at PowerSystems World, Chicago, IL on November 10, 1999. Reprinted by permission.

**'SMART', CONSTANT-TORQUE,
MICROSTEPPING DRIVE
CIRCUITRY FOR UNDER \$10**

GENERAL INFORMATION

World-Wide Web

Complete, up-to-date, on-line information, when you need it, where you need it is available on the World-Wide Web at

www.allegromicro.com

CD-ROM

Detailed technical information is also available in CD-ROM form (available from any Allegro representative or sales office, franchised distributor, or via the web) as the

Allegro Electronic Data Book, AMS-702

Product Information Center

Printed data sheets, application notes, technical papers, samples, and the *Allegro Electronic Data Book* (CD-ROM) are also available by calling (8 am to 5 pm, ET)

1-508-ALLEGRO
(1-508-255-3476)