zilog

Application Note

A Unipolar Stepper Motor Drive Using the Z8 Encore!® MCU

AN012805-0708

Abstract

Stepper motors that feature unipolar drives are widely used in applications that require high torque loads and fast position attainment. The unipolar operation provides stable motor control using a relatively simple firmware, as compared to bipolar drives.

This Application Note discusses the use of a typical 8-wire or 6-wire stepper motor, in the unipolar mode of operation, with a Z8 Encore!® microcontroller.

Also discussed are the practical aspects of the full-step and half-step methods of driving a stepper motor, as well as a complete hardware and software implementation using the Z8 Encore!® MCU. A simple schematic using readily-available components, along with the full source code in C, is used to implement a stepper driver. This basic hardware and software combination can be incorporated into larger control circuits with suitable modifications.

Z8 Encore!® Flash MCU Overview

Zilog's Z8 Encore!® products are based on the new eZ8 CPU and introduce Flash memory to Zilog's extensive line of 8-bit microcontrollers. Flash memory in-circuit programming capability allows for faster development time and program changes in the field. The high-performance register-to-register based architecture of the eZ8 core maintains backward compatibility with Zilog's popular Z8 MCU.

The new Z8 Encore!® microcontrollers combine a 20 MHz core with Flash memory, linear-register SRAM, and an extensive array of on-chip peripherals. These peripherals make the Z8 Encore!® suitable for a variety of applications including motor control, security systems, home appliances, personal electronic devices, and sensors.

Discussion

A discussion about stepper motors is presented in this section.

General Overview of Stepper Motors

Stepper motors are characterized by high torque and are capable of handling large loads with precise movements. The advantages of stepper motors over *stepless* AC or DC motors include:

- No feedback requirement for position or speed control (open loop operation)
- Noncumulative positional errors
- Precise electronic speed control using digital technology
- Compact size for driving large loads at low speeds

With the advent of microcontrollers, stepper drive technology has advanced rapidly both in terms of flexibility and complexity. The new Z8 Encore!® series of Flash microcontrollers enable drive designers to implement full functionality while maintain a low component count.

Stepper Motor Construction

Figure 1 shows a typical stepper motor with four salient stator poles (the fixed part) and two rotor poles (the moving part). The stator poles are wound with a set of windings featuring two coils each, connected in series on opposite poles (forming two phases), while the rotor poles are permanent magnets comprised of soft alloy steel.

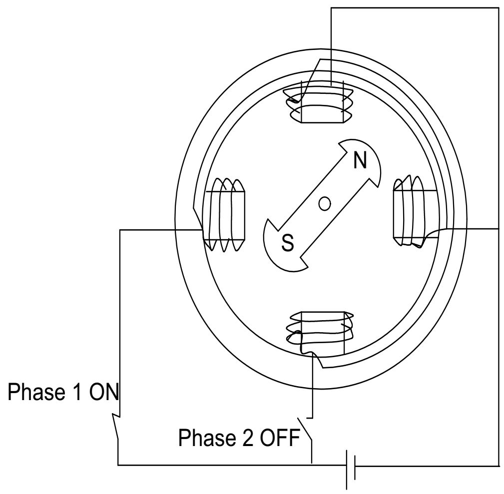

Figure 1. A Simple Stepper Motor

Sequentially energizing the two windings causes the rotor poles to align with the electric poles and create the rotor movement (see Figure 2). The equilibrium states of the rotor are called detent positions and are fixed according to the mechanical structure of the motor.

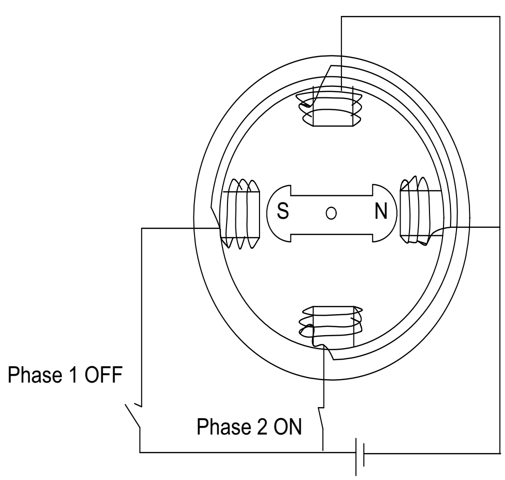

Figure 2. Sequential Winding Energization

Stepper motors feature stator windings that are wound in unipolar or bipolar fashion on any number of poles; the mechanical step size usually ranges from 0.9° to 18°. In the unipolar mode of operation, the current flow in the windings always remains in the same direction to achieve rotor movement. A bipolar mode, on the other hand, involves an alternate reversal of the current flow in the windings to achieve rotation.

An Electrical Drive for a Stepper Motor

An electrical drive provides the requisite amount of current to the windings of the stepper motor in a predefined sequence. A typical scheme for unipolar drives is shown in Figure 3.

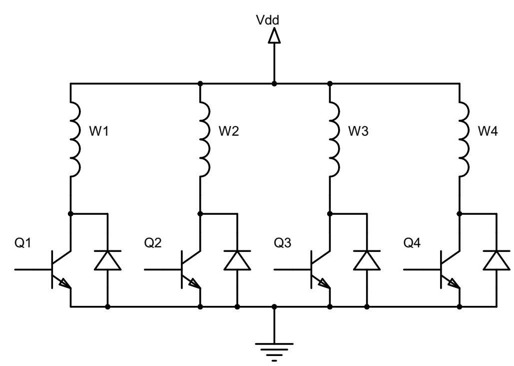

Figure 3. A Unipolar Drive Scheme for a 6- or 8-Lead Stepper Motor

The four windings are energized in a sequential manner to rotate the shaft. The number of steps required to make one revolution equals 360 ÷ (step size). For a 2-phase wound stepper motor with 1.8° mechanical step size, the total number of steps is 360 ÷ 1.8 = 200.

The number of steps performed by a stepper (resolution) can be increased by changing its sequence. To generate 200 steps, the full-step mode (see Table 1) is employed. To generate 400 steps (i.e., a resolution of 0.9) half-step mode (see Table 2) is required. The reversal of direction in both cases is achieved by reversing the sequence only.

The excitation method of the unipolar drive can be a single-phase On (1-ph On) or a two-phase On (2-ph On) operation. Table 3 shows the 1-ph On method and Table 4 shows the 2-ph On method for a four-phase wound motor.

The excitation scheme is S1–S2–S3–S4 for the clockwise rotation in the 1-ph On and 2-ph On methods (as per Table 2) for a four-phase motor, and S4–S3–S2–S1 for the counterclockwise rotation. The excitation scheme for the 2-ph On method requires 2 windings to be energized at the same time in an 8-sequence pattern.

The advantages of the 2-ph On method over the 1-ph On method are higher torque and stable, smooth rotations, due to the fact that two windings are energized at a time. The half-step mode effectively combines the 1-ph On and 2-ph On methods.

Table 1. Full-Step Mode

	Wdg 1	Wdg 2	Wdg 3	Wdg 4
Seq 1	1	0	1	0
Seq 2	1	0	0	1
Seq 3	0	1	0	1
Seq 4	0	1	1	0

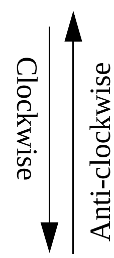

Table 2. Half-Step Mode

	Wdg 1	Wdg 2	Wdg 3	Wdg 4
Seq 1	1	0	1	0
Seq 2	1	0	0	0
Seq 3	1	0	0	1
Seq 4	0	0	0	1
Seq 5	0	1	0	1
Seq 6	0	1	0	0
Seq 7	0	1	1	0
Seq 8	0	0	1	0

Table 3. 1-Ph On Method

	1	2	3	4	5	6	7	8
Wdg 1	1	0	0	0	1	0	0	0
Wdg 2	0	1	0	0	0	1	0	0
Wdg 3	0	0	1	0	0	0	1	0
Wdg 4	0	0	0	1	0	0	0	1

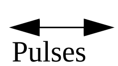

Table 4. 2-Ph On Method

	1	2	3	4	5	6	7	8
Wdg 1	1	1	0	0	1	1	0	0
Wdg 2	0	1	1	0	0	1	1	0
Wdg 3	0	0	1	1	0	0	1	1
Wdg 4	1	0	0	1	1	0	0	1

A better way to achieve higher resolution is by using the microstepping method, in which a mechanical step is subdivided into many steps. By energizing the windings in various combinations, a required angular rotation can be executed. The microstepping method is mainly used in applications that require a greater degree of control for precise movement, such as in X-Y table applications.

However, the hardware in this Application Note is limited to the construction of a unipolar drive, and the software uses the 2-ph On method in both full-step and half-step modes.

The Electrical and Mechanical Characteristics of a Stepper Motor

The amount of holding torque, a typical parameter of a stepper motor, determines the choice of motor for a particular application. Holding torque is defined as the maximum amount of torque that can be applied on the shaft of an energized but stopped motor without causing rotor slippage. The holding torque and winding current relation is nearly linear, as shown in Figure 4, and the minimum amount of torque on the Y-axis, appearing at zero current, is the detent torque.

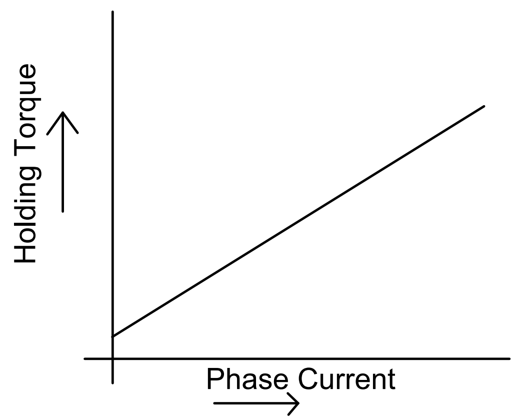

Figure 4. Holding Torque vs. Current

With more current in the windings, the magnetic force applied on the rotor by the stator increases. Electrical drives utilize this curve to minimize the winding current when the motor is stopped for a long time. Heat dissipation in the motor is thereby reduced, as well as energy use.

The pull-in torque characteristics of a stepper motor (see Figure 5) demonstrate the ability of the motor to start with a particular amount of torque placed on the shaft. The pull-out torque characteristics (in Figure 5) show the ability of the motor to sustain rotation at a particular torque without losing step synchronism.

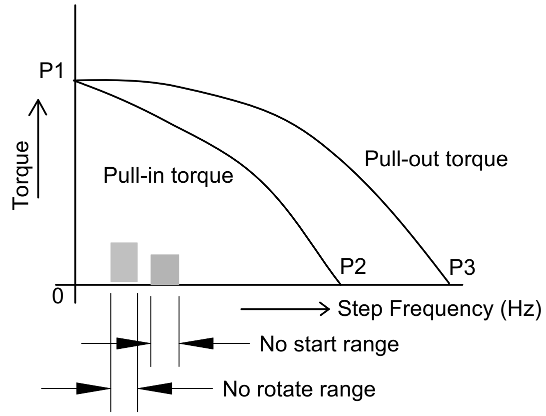

Figure 5. Torque vs. Speed Characteristics

The main parameter governing the torque-speed curve is the *L*/*r* ratio of the windings. The inductance (*L*) and the resistance (*r*) decide the maximum operating frequency of the motor, where the equation:

$$r = r_{winding} + r_{external}$$

determines the total winding resistance.

Increasing *r* by introducing external resistance $r_{external}$ in series with the windings has the effect of shifting the pull-out curve towards the right, thereby enabling the motor to rotate at higher speeds, although at lowered torque values.

The motor can lose speed or stall completely between the minimum and maximum pulse rates, or rotational speeds, applied to a motor. At these points, oscillations are created in the winding currents that result in an unstable rotation of the motor. Typically, these points are divided into lowmid- and high-frequency areas, as depicted in Figure 6.

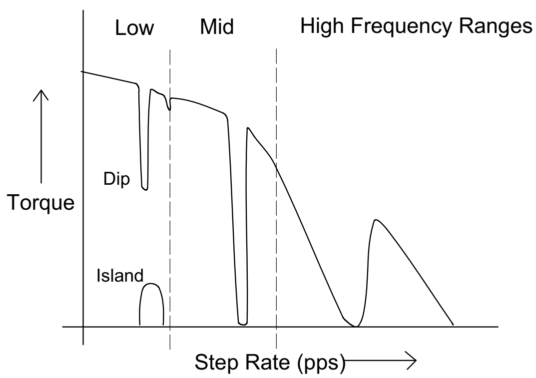

Figure 6. Typical Regions of Resonance

These resonance frequencies must be avoided when a drive is tuned for a particular motor. A mechanical damper, illustrated in Figure 7, is coupled to the motor shaft, and can be used to damp the resonance points.

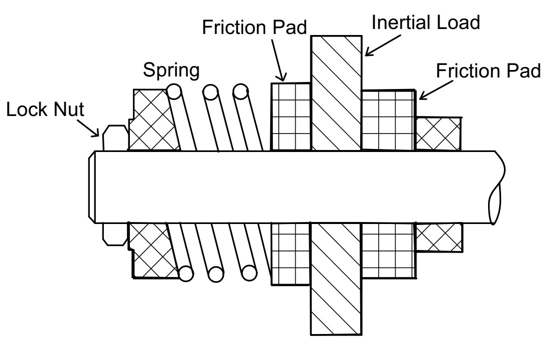

Figure 7. Spring-Friction Inertial Damper

Implementing Unipolar Stepper Drive

Unipolar Drive Hardware Description

The schematic (see Figure 14 on page 4) consists of generally-available components and is simple to build and test. The heart of the circuit is the Z8 Encore!® microcontroller, which operates at 18.432 MHz. With a V_{CC} of 3.3 V, the power consumption and heat dissipation of the Z8 Encore!® microcontroller is greatly reduced.

The circuit consists of buffer transistors Q1, Q2, Q3, and Q4 driving the power MOSFETs T1, T2, T3, and T4, respectively. There are four buffer-driver power MOSFET units on four GPIO pins that individually energize each of the four windings of the stepper motor.

Figure 8 presents a schematic diagram of a single stage of the power driver.

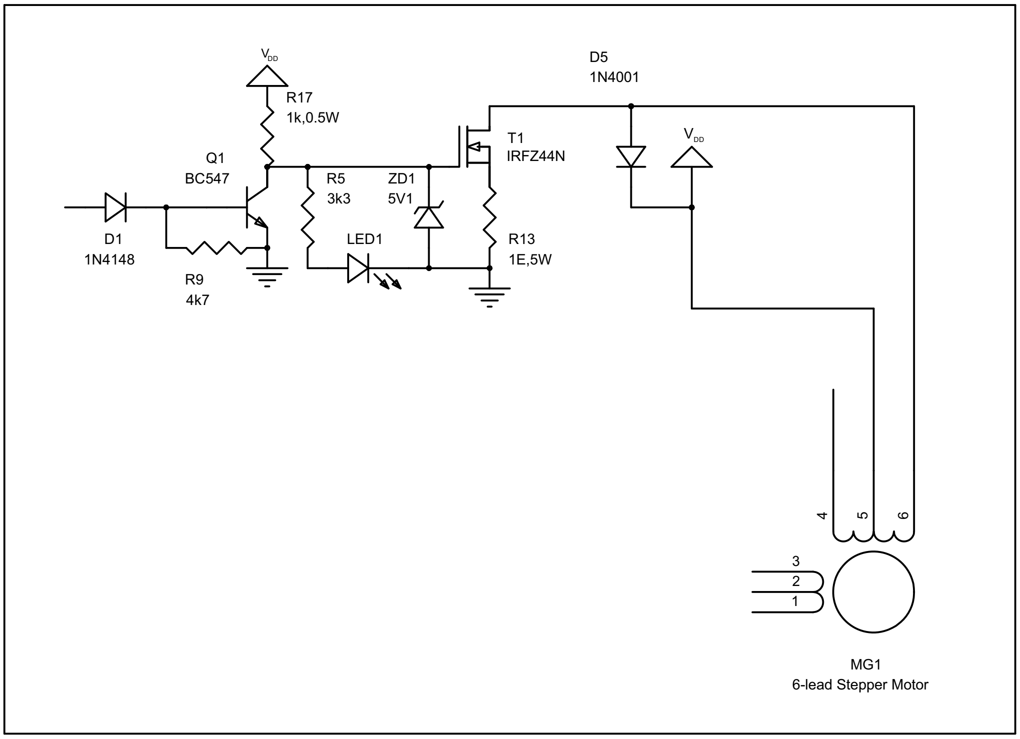

Figure 8. A Power Driver Stage Using N-Channel MOSFET

In Figure 8, note the following:

- Diode D1 protects the I/O pin from surges originating on the power circuit side.
- Resistor R5 drives the LED1 when the winding is energized.
- Transistor Q1 acts as a buffer driver for MOSFET T1 by switching off when the I/O pin is Low and vice-versa, allowing the MOSFET to conduct current and energize winding W1. The choice of N-channel MOSFETs depends on the voltage and current rating of the motor. In our example, IRFZ44N is chosen based on the motor specifications of 12V and 1.25Amps per winding. Because the MOSFET carries a high current, consider using a proper heat sink with a heat sink compound.
- Zener diode ZD1 ensures that the firing voltage on the gate of the MOSFET does not exceed 5V.
- Resistor R17 limits the gate current.
- Diode D5 removes any surges in the winding above a voltage level of V_{DD}.
- Resistor R9 pulls the Q1 base Low by default and prevents it from conducting at spurious signals.
- Power resistor R13 is used to balance the *L*/*r* ratio of the windings and to limit the current depending on the resistance of the windings. An additional potential divider across this resistor can be used to sense the winding current by utilizing other ADC inputs of the Z8 Encore!® microcontroller, as illustrated in Figure 9.

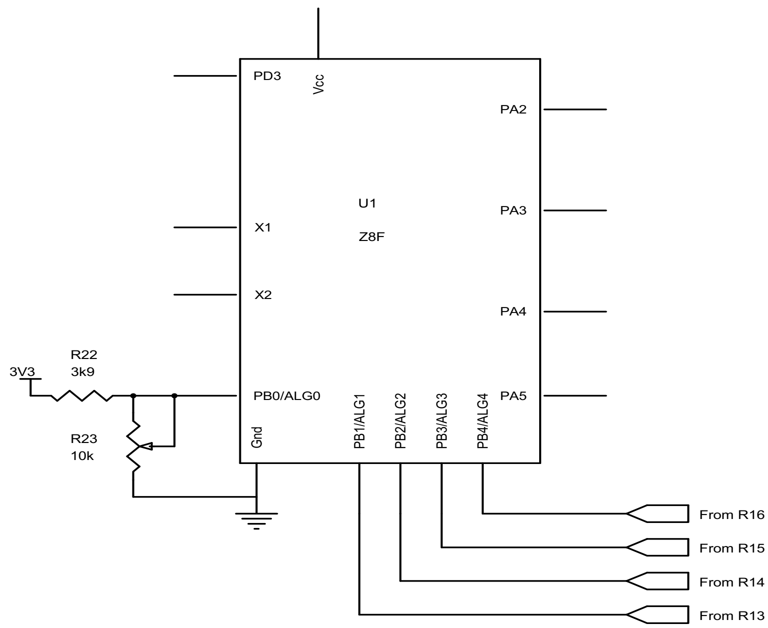

Figure 9. ADC Channel Inputs (Speed Potentiometer and Current Sensing)

- Switch SW1 (see Figure 14 on page 4), when connected to pin PD3, acts as the directional control input. Pressing SW1 reverses the direction of rotation, while speed remains unchanged.
- The half-step mode (selected at compile time) rotates the motor at exactly half the RPM of full-step mode at the same step frequency.
- The step frequency, and consequently the speed of the stepper motor, can be varied by using potentiometer R23. The voltage at the wiper pin of the potentiometer is fed to ADC input channel 0. The reference voltage is generated internally in the microcontroller and is used to compare the voltage on the ADC input.

This circuit is designed such that any type of unipolar wound motor, 8- or 6-lead, can be driven by it. The only component change that may be required is a MOSFET with appropriate voltage and current ratings.

Unipolar Drive Software Description

The source code of unipolar drive is written in ANSI compatible C and compiled with ZDS II, Zilog's Integrated Development Environment (IDE) for Z8 Encore!® series. The compilation options are provided in the source code (AN0128-SC02) available from www.zilog.com.

The main routine consists of three functions to initialize the peripherals:

- `init_timer0()` initializes Timer0
- `init_adc()` initializes the on-chip ADC
- `init_p3ad()` sets up the GPIO pin 3 of port D as an external triggered interrupt

These peripherals are interrupt-driven and their individual Interrupt Service Routines, are:

- `isr_timer0()`
- `isr_adc()`

- `isr_p3ad()`

After initialization, the *while* loop is executed infinitely, interrupted only by the completion of one of the following three events:

- Timer count roll-over
- ADC conversion complete
- Pressing of push-button SW1

The pulsed switching of voltage in the four windings of the motor is achieved by providing alternate On and Off periods in the `isr_timer0()` function. These on-off sequences are provided to the I/O pins, according to the sequence table for full (See Table 1 on page 3) or half-step modes (see Table 2 on page 3).

The Off period of the pulses is dependent on the current fall time in a motor winding, and essentially constant; the On period varies according to the desired speed of the motor. If a higher speed is required, the On time is reduced to produce narrower pulses, and vice-versa for lower speeds. Such variation in speed is achieved by mapping the position of the potentiometer (voltage available at wiper pin) with the exact time values to be placed into the `T0CPH` and `T0CPL` registers.

The `table.h` file in the source code (AN0128-SC02) available from www.zilog.com contains the 2-dimensional look-up table, in the format: `array[10][2]`, consisting of ten pairs of potentiometers:exact time values. Depending on the ADC calculation (position of the speed potentiometer) an appropriate value is chosen from the table and loaded into the timer registers. Varying the potentiometer from minimum to maximum position results in ten different motor speeds.

The look-up table can be expanded to include more number of pairs corresponding to different motor speeds mapped to potentiometer positions. These pairs can then be loaded into `Timer0`, as illustrated in the main routine.

The ADC control register is configured with One-shot conversion and with an enabled Internal Voltage reference. Furthermore, to set up the ADC, the alternate function of Port B must be enabled by setting the PDAF. The data direction is input for all, and the PDDD is set appropriately.

The priority for ADC is set to the highest by using IRQ0E0 and IRQ0E1. In this implementation, only the ADCDH register is used, but to get more accurate values by making full use of the 10 bits, the register ADCDL can also be used.

The `isr_adc()` routine demonstrates how other ADC inputs can be utilized for current sensing by using a resistor divider, as discussed in the The Electrical and Mechanical Characteristics of a Stepper Motor section on page 4.

A Low on the GPIO pin PA2/PA3/PA4/PA5 energizes the particular winding, such that the data bit is inverted before it is sent to the port pin.

`Timer0` is initialized by configuring `T0CTL` register with the appropriate Hexadecimal word.

The switch SW1 that is connected to pin PD3 (Pin3 of PortD), acts to reverse the direction of the motor. To set PD3 as a source of external interrupts, set the register PDDD to use pin PD3 as the output pin, select the highest priority (same priority as ADC) using IRQ1E0, IRQ1E1, and select the falling edge with IRQES. In addition, enable port D for interrupts by setting the PS register appropriately. To reverse the motor direction, reverse the sequence as per Table 1 or Table 2 on page 3, in the *'switch:case'* statement of the `isr_timer0()` routine.

To select full or half step mode at compile time, assign the value of variable step as `0x01` or `0x00`, in the `main()` routine.

Testing the Unipolar Stepper Drive

The test setup to demonstrate the working of unipolar stepper motor drive as described in this Application Note is illustrated in Figure 10.

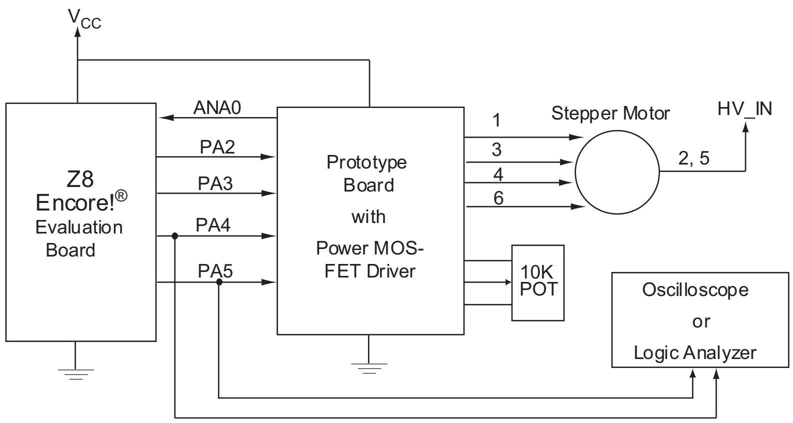

Figure 10. Test Setup for Unipolar Stepper Drive

Equipment Used

The software and hardware were tested using the following equipment:

- Z8ENCORE000ZCO—Z8 Encore!® Development Kit with the Z8F6403 MCU
- Z8F08200100KIT—Z8 Encore!® Evaluation Kit with the Z8F0822 MCU
- ZDSII–IDE, v4.5.0 for the Z8 Encore!® series of MCUs
- Tektronix TDA724D Digital Storage Oscilloscope with PC software
- HP 1661A Logic Analyzer
- Power circuitry on breadboard
- Stepper motor, with the following specifications:
  - Voltage and Current Rating = 12VDC, 1.25A per phase
  - Step angle = 1.8º
  - Torque = 3kgf-cm
  - Number of leads = 8

Test Procedure

The procedure to connect and tune-up a stepper motor to the drive circuit (see Figure 14) is discussed below:

1. Check that the voltage supplied to the motor windings (HV_IN) is the same as specified by the manufacturer, with a maximum tolerance of ±10%. Here 12V is supplied to the motor. The power supply should be able to source at least 2 times the winding current requirement of the motor.
2. As described in the The Electrical and Mechanical Characteristics of a Stepper Motor on page 4, avoid frequencies of nonoperation. These frequencies cause electrical resonance in the motor at high loads and can result in permanent damage to the motor. The nonoperation frequency range is usually specified by the motor manufacturer; however, these frequencies can be determined by testing at no-load.

3. To map the potentiometer values, code certain frequencies (that is, `Timer0` ON-time values that match the set speed) in the LUT provided in the *if…else* statement of the `main()` routine in the source code (AN0128-SC02) available from www.zilog.com.
4. Proper cooling arrangements must be made because the motor heats up with continuous operation. Likewise mount the power components (transistors, MOSFETS, and other components) with appropriate spacing between them and provide adequate heat-sinking.
5. Identify the stepper motor leads for 4 sets of windings, with a multimeter in continuity mode. Tie together one end of each winding and connect the other end to V_{DD}. For a 6-lead motor, tie together one end of the middle two wires and connect the other end to V_{DD}.
6. Connect the prototyping board to the appropriate pins on the connectors of the Z8 Encore!® Development/Evaluation board and power-up the motor.
7. Download the executable binary onto the microcontroller using ZDS II utility. Execute the code and test the motor direction from the shaft end. If it is reversed, interchange any two winding leads and power-up the motor again.
8. Observe the variation in speed by changing the speed potentiometer position from minimum to maximum. To reverse the direction, press the switch SW1.
9. The Digital Storage Oscilloscope (DSO) can also be used to check the speed variation. Observe any irregularities in speed and check the frequency (pps) using a DSO, keeping in mind not to use the nonoperational frequencies. Perform in-circuit debugging using the On-chip Debug system supplied with the Z8 Encore!® Development/Evaluation Kit.
10. To obtain the results as shown in the Oscilloscope Charts on page 10, match the pulses to that shown in Chart 1 and Chart 2, and the voltage and current waveforms to that shown in Chart 3.

Test Results

The Z8 Encore!®-based stepper drive works with the 2-ph On principle and provides stable motor control.

Using the potentiometer a smooth variation of speed was obtained, with typical pulse widths ranging from 3.1 ms (frequency of 277 pps) to 29.3 ms (33 pps). The Off time between successive pulses remained constant at 500 µs. These values can be modified in the source code (AN0128-SC02) available from www.zilog.com.

The direction control was achieved by using SW1 and the mode was selected using variable step in the `main()` routine. The rotational speed in the half-step mode was reduced to half of that of the full-step mode at the same pps frequency, leading to a better step resolution.

Oscilloscope Charts

Figure 11 is a screen shot of the full-step waveforms at the base pins of buffer transistors Q1, Q2, Q3, and Q4, in clockwise motor rotation. This schema matches the timing steps in Table 1 on page 3 for the full-step method discussed earlier.

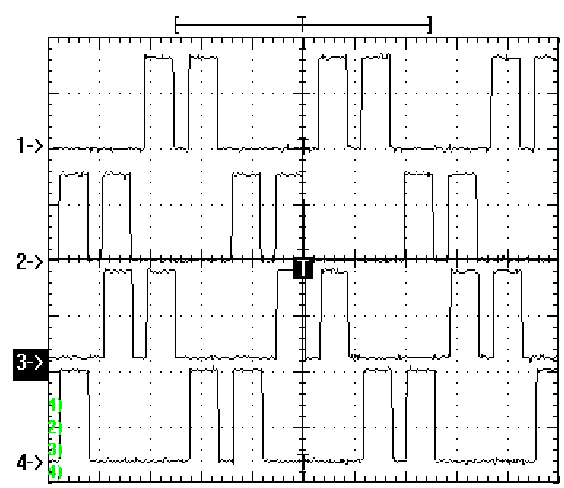

Figure 11. Full-Step Waveforms - Buffer Transistors

Figure 12 is a screen shot of the waveforms in full-step mode, available at the gate pins of MOSFETs T1,T2, T3, and T4. Note that the logic is the inverse of Figure 11 (although not on same time scale).

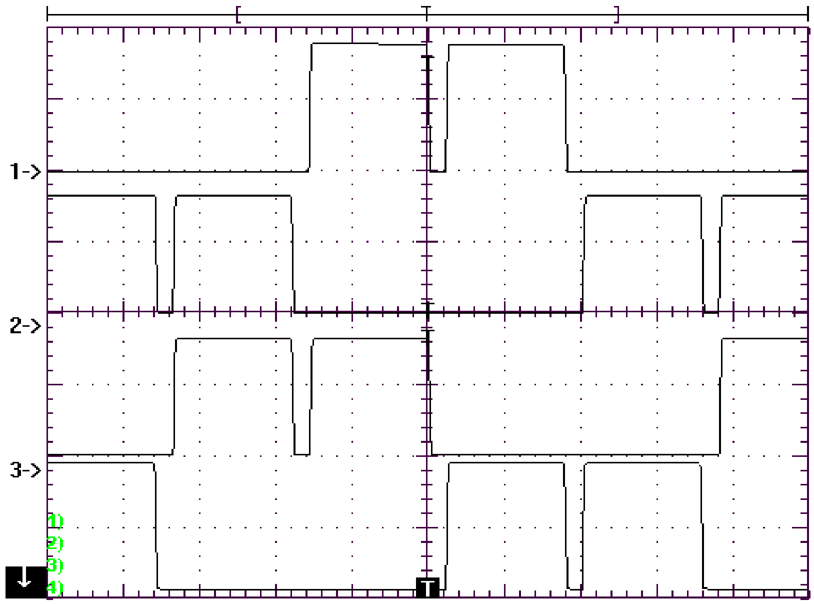

Figure 12. Full-Step Waveforms - MOSFET Gate Pins

Figure 13 is a screen shot showing the typical voltage (top) and current waveform (below) in full-step mode, available at the Gate and Source of MOSFET T1.

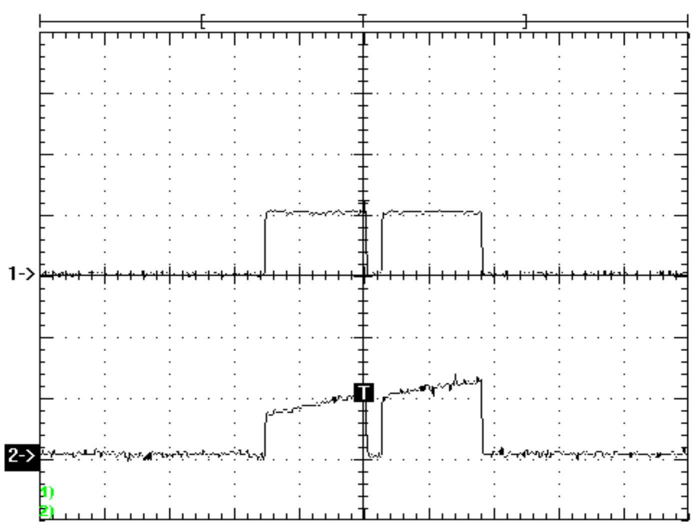

Figure 13. Full-Step Waveforms—Gate and Source of MOSFET T1

Summary

This Application Note successfully demonstrates the operations of a unipolar stepper motor drive based on the Z8 Encore!® series of Flash microcontrollers. Only one 16-bit timer, one 10-bit ADC channel, and an external interrupt I/O pin are used, leaving other resources free for the designer. The ANSI-C code, including an array table, occupies a mere 1689 bytes, with a RAM requirement of only 38 bytes.

The powerful peripheral features of the Z8 Encore!® series of microcontrollers make the Z8 Encore!® MCU convenient to use in complex motor control applications.

Appendix A— Reference

Further details about stepper motors, the Z8 Encore!® MCU and the ZDS II–IDE can be found in the references listed in Table 5.

Table 5. List of References

Topic	Document
Stepper motors	Stepping motors and their Microprocessor Controls – Takashi Kenjo; Oxford Press, 1984.
Zilog Developer Studio (ZDS II–IDE) v4.1.0	Zilog Developer Studio II—Z8 Encore!® User Manual (UM0131).
Z8 Encore!® product specifications	Z8 Encore!® Microcontrollers with Flash Memory and 10-Bit A/D Converter Product Specification (PS0176).

Appendix B—Glossary

Definitions for terms and abbreviations used in this Application Note are listed in Table 6. .

Table 6. Abbreviations/Acronyms

Term/Abbreviation	Definition
MOSFET	Metal-Oxide Semiconductor Field Effect Transistor, N-channel or P-channel.
BJT	Bipolar Junction Transistor, npn or pnp.
ADC	Analog-to-Digital Converter.
Flash	Read-Only Memory for Code and Constant Data Storage.
Emulator	Equipment used to mimic the functioning of a microprocessor.
PPS	Pulse Per Second, a measure of frequency.

Appendix C—Schematic Diagrams

Figure 14 illustrates a schematic for a unipolar stepper motor drive using the Z8 Encore! MCU.

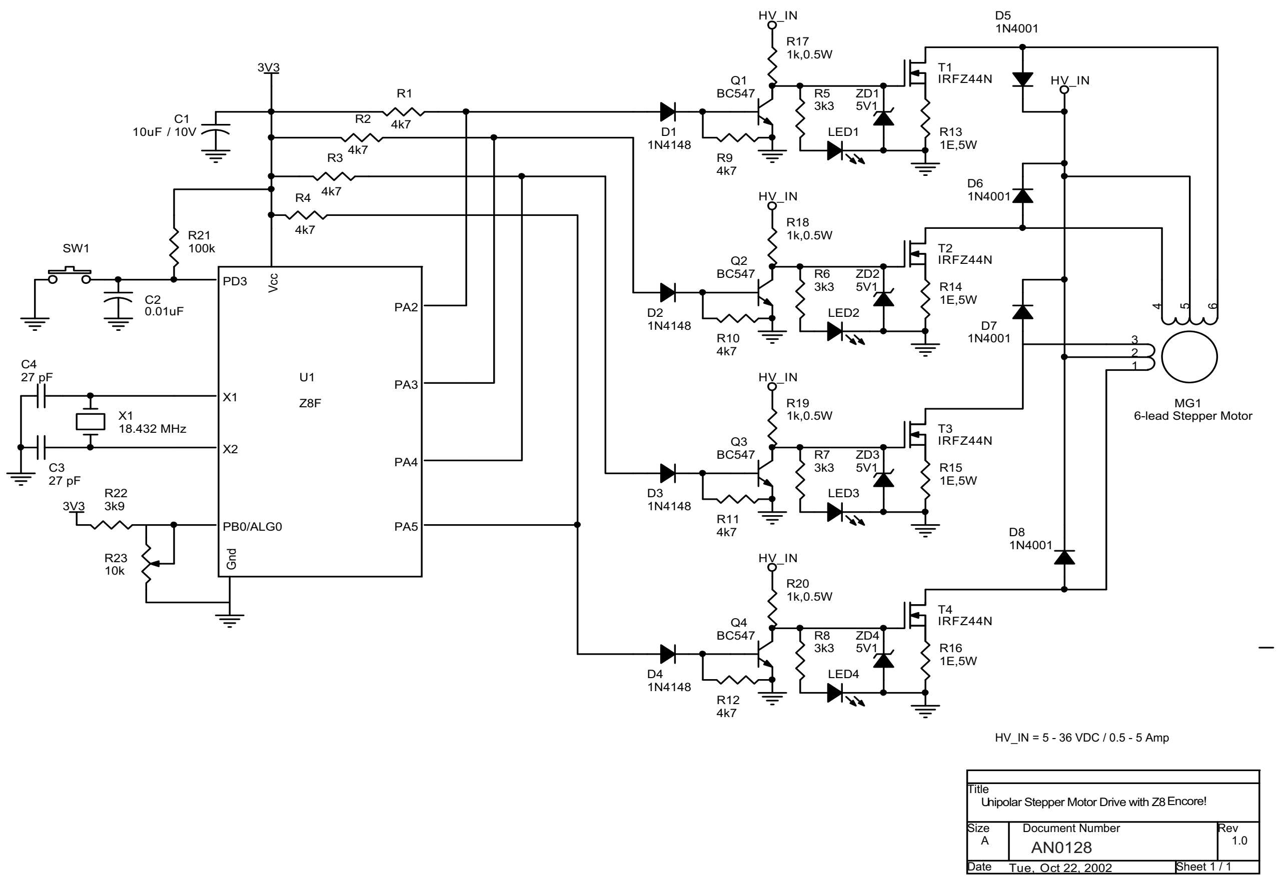

Figure 14. Schematic for A Unipolar Stepper Motor Drive Using the Z8 Encore!® MCU

Appendix D—Flowcharts

This appendix illustrates flowcharts of the various routines developed for a unipolar stepper motor application using the Z8 Encore!® MCU.

Figure 15 illustrates the flow of the main routine.

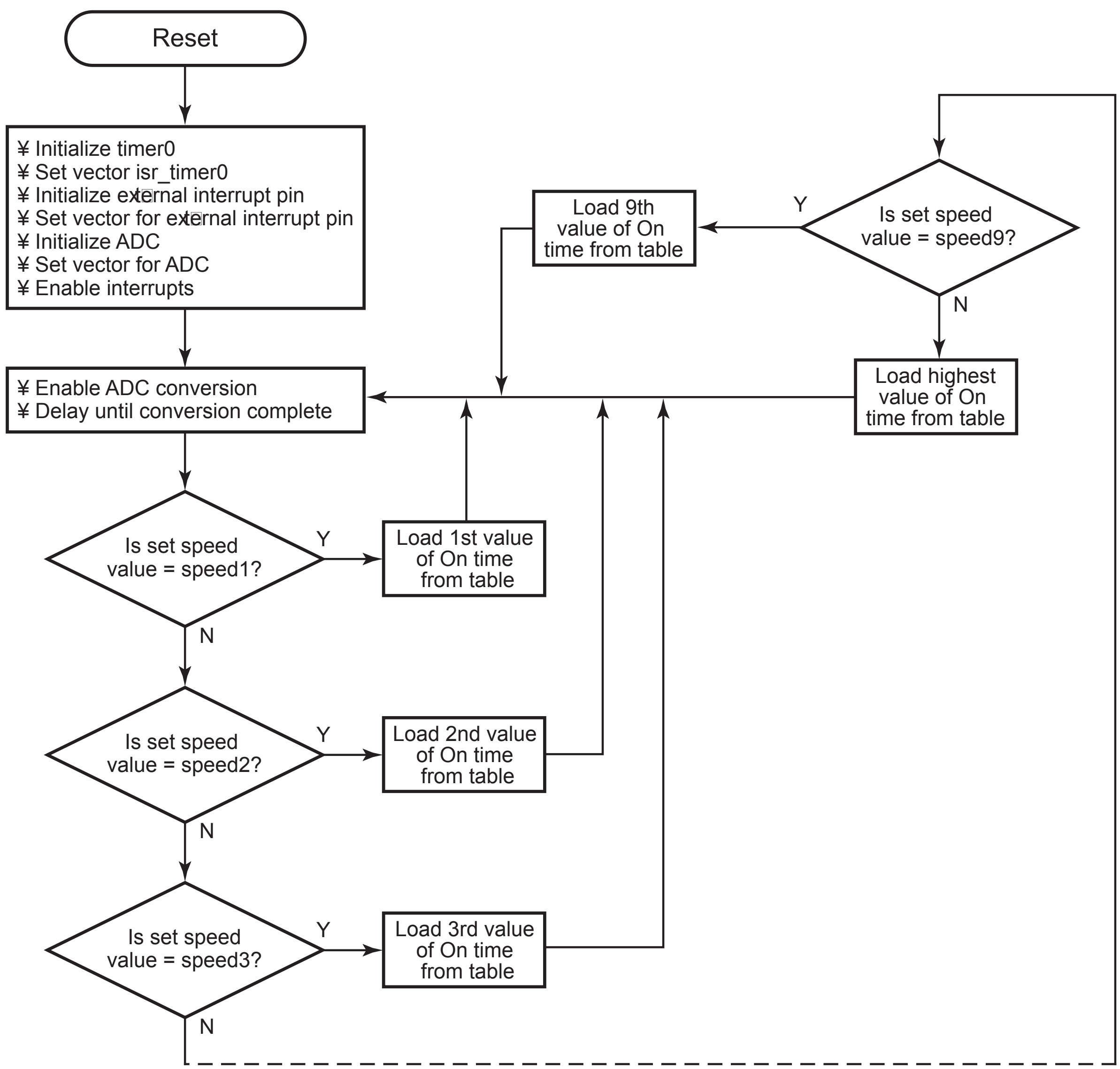

Figure 15. The Main Routine

Figure 16 illustrates the flow of the external interrupt pin routine.

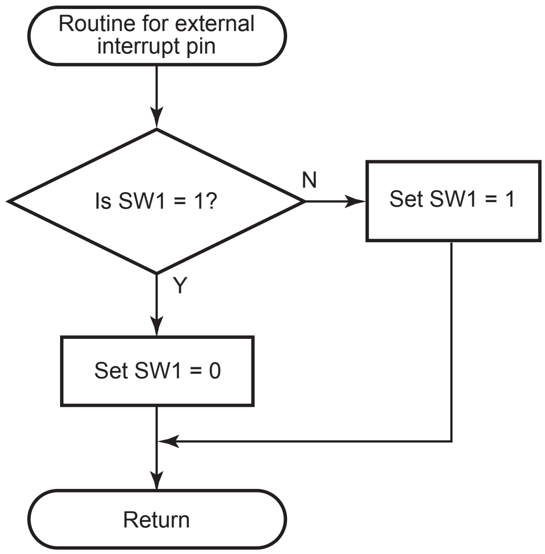

Figure 16. The External Interrupt Pin Routine

Figure 17 illustrates the flow of the ADC interrupt routine.

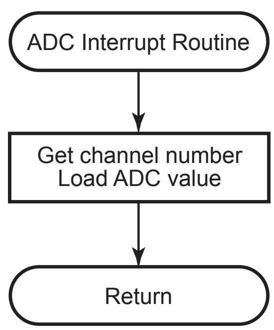

Figure 17. The ADC Interrupt Routine

Figure 18 illustrates the flow of the `Timer0` interrupt routine.

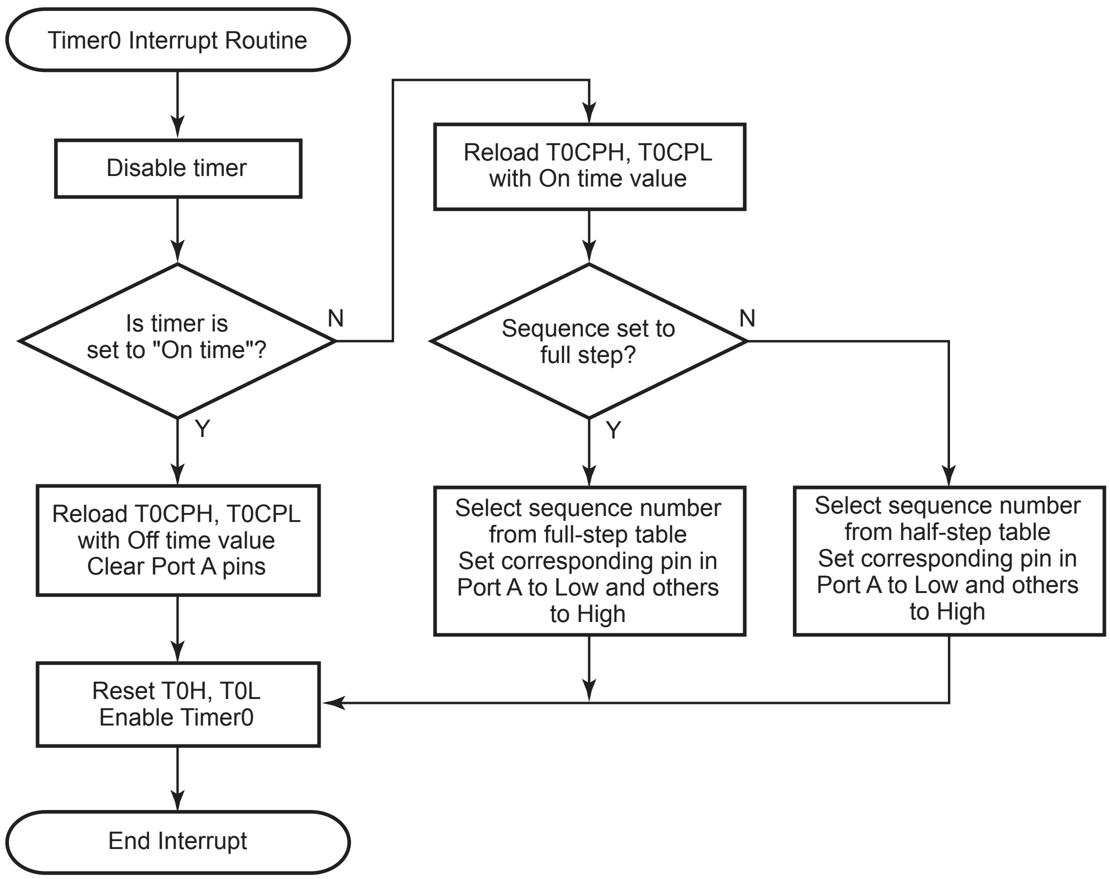

Figure 18. The Timer0 Interrupt Routine

Warning: DO NOT USE IN LIFE SUPPORT

LIFE SUPPORT POLICY

ZILOG'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS PRIOR WRITTEN APPROVAL OF THE PRESIDENT AND GENERAL COUNSEL OF ZILOG CORPORATION.

As used herein

Life support devices or systems are devices which (a) are intended for surgical implant into the body, or (b) support or sustain life and whose failure to perform when properly used in accordance with instructions for use provided in the labeling can be reasonably expected to result in a significant injury to the user. A critical component is any component in a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system or to affect its safety or effectiveness.

Document Disclaimer