APPLICATION NOTE

AN1281

NE5570: A theory of operation and applications

Authors: Larry Engh, Carl Fenger, Les Hadley, Dan Linebarger

1988 Aug

Philips
Semiconductors

PHILIPS

NE5570: A theory of operation and applications AN1281

Authors: Larry Engh, Carl Fenger, Les Hadley, Dan Linebarger

The Philips Semiconductors NE5570 monolithic CMOS controller is described with basic applications showing its adaptation to microprocessor-controlled servo systems. The controller (NE5570) is adaptable to general three-phase brushless motor control by means of a serial data input command format which provides multiple function addressing within the chip. These functions include rotational directions, 60 or 120 degrees commutation select, and current mode control to each phase on a dynamic basis by continuously programmed pulse width modulator.

INTRODUCTION TO CIRCUIT FEATURES

An 8-Bit Digital-to-Analog Converter

The internal register drives an 8-bit digital-to-analog converter (DAC). The voltage output of this converter is applied to the positive input of the error amp. This op amp is normally used for integrating the current measure at the output of op amp one. The error amplifier output drives the pulse width modulator (PWM).

For ease of use, the DAC voltage is continuously present to the error amplifier input. Even during changes of value, the transition is smooth to a new set point of the pulse width modulator. This is achieved through use of precision ratio resistors and CMOS transmission gates. The DAC can be quickly updated through the serial interface. The integrator time constant should be chosen to complement the motor dynamics, i.e., transient response, gain, and phase margin.

The accuracy of the DAC features excellent differential linearity. This is helpful for precision control. Integral linearity is very good; this allows consistent production tolerance for the overall system. The DAC reference input directly drives the resistor string of the DAC. The nominal reference voltage is 5V, and this is the voltage at which accuracy is specified. For consistent system performance, the reference voltage should be well filtered. Because the DAC is a CMOS design, voltages other than precisely 5V can be tolerated, however.

The Operational Amplifier

The op amp used for current sense must have a common-mode range which includes ground and negative down to 0.5V. A special design was required to meet this performance objective. The input is level-shifted with a source-follower pair and applied to a p-channel differential pair. This technique preserves the very high input impedance of a CMOS input down to -0.5V. At this voltage, the input protection network begins to draw current, and inputs should be current limited if negative spikes are present.

Output Stage

The output stage is an inverter design. It features moderate idle current with good capacitive drive capability and stability. For high resistance loads, the output can swing very nearly rail-to-rail. Outputs are tested at 100mA source-sink. (See Figure 1.)

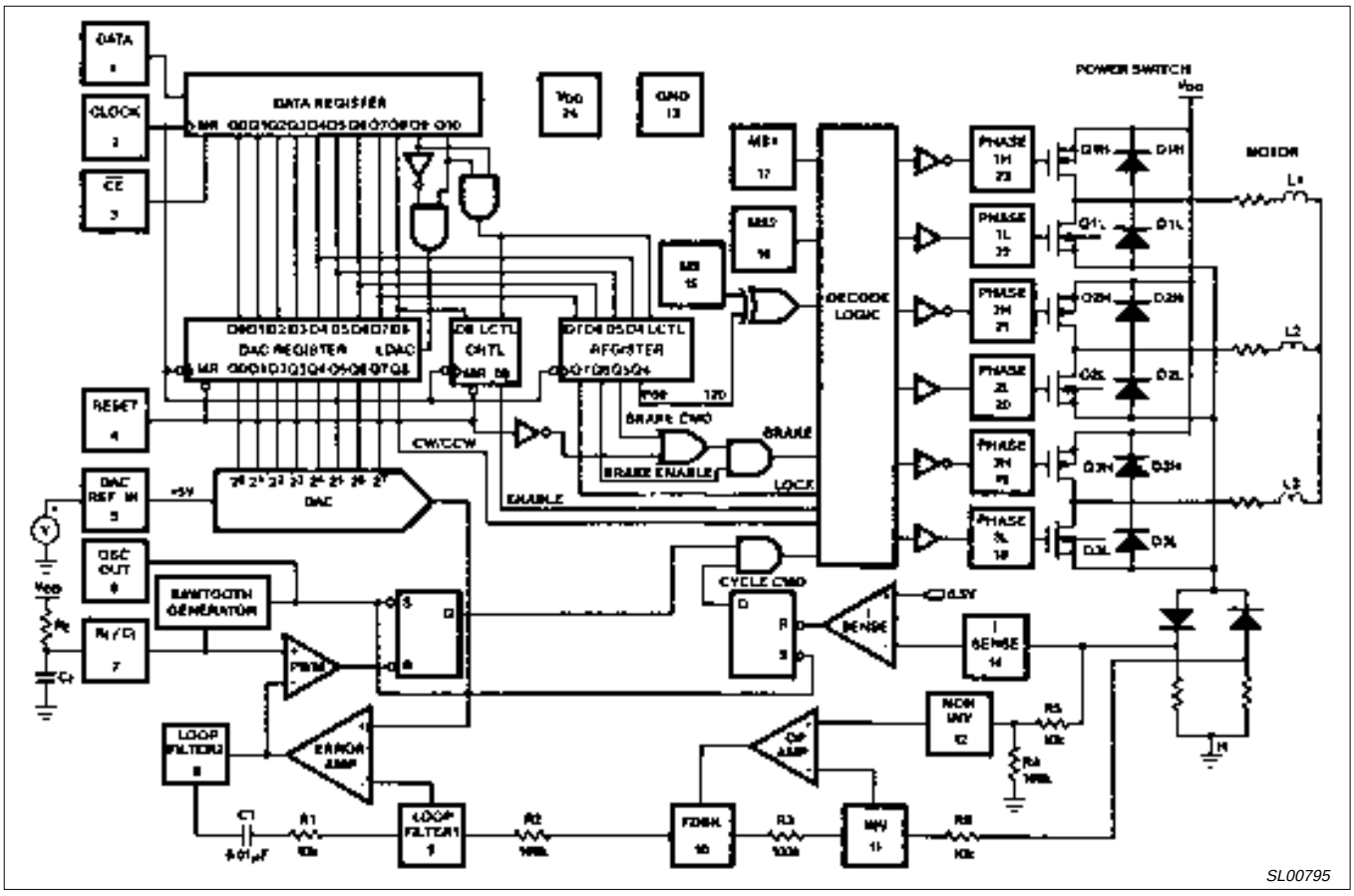


Figure 1. NE5570 Block Diagram

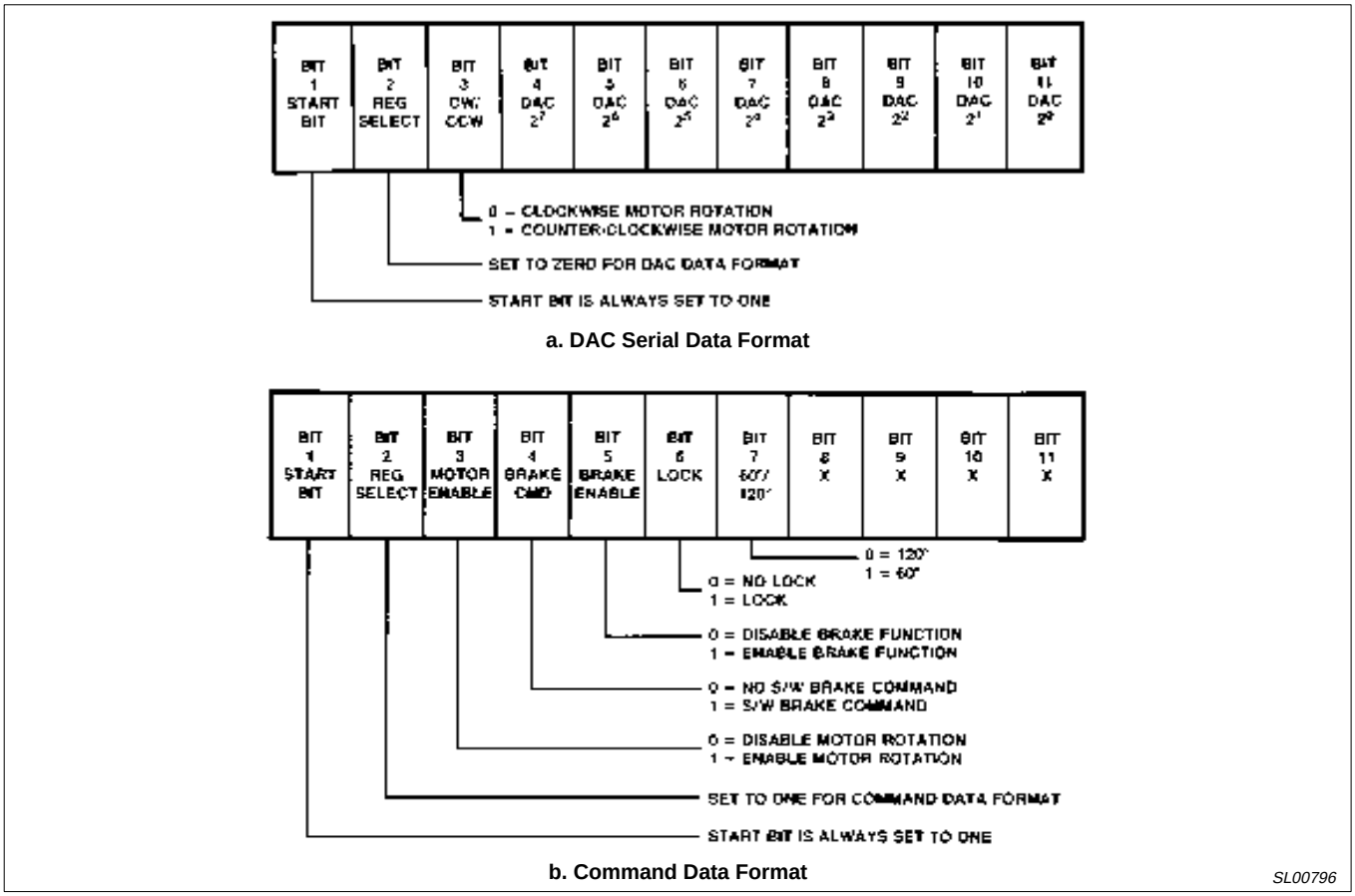


Figure 2.

Serial Data Entry-DAC Input

Three pins are used to allow the input of a serial word into an 11-bit serial-to-parallel data register. When CE is active, data may be clocked into the DATA pin using a rising edge clock into pin CLOCK. These three pins (DATA, CLOCK, CHIP ENABLE) are compatible with standard TTL input levels. The first two bits of the serial word select either the DAC data register or the Command control register.

In order to select the DAC register, the first two data bits must be 1 followed by a 0. The remaining nine bits are used to set the DAC to the desired level and control the direction of the motor rotation (CW or CCW). All eleven data bits must be shifted in before the DAC can be selected. The DAC register has an internal serial address of 10. (See Figure 2a.)

NOTE:
Output change will occur on the falling edge of the 11th clock bit.

The Control and Command Registers

The control register has an internal binary address of 11 (see Figure 2b). When the
control register has been selected by clocking in an 11-bit serial word with the first two bits equal to 11 in binary, the command register can select any of five different motor control commands. This register can direct the outputs to switch at the oscillator frequency in order to drive the motor, lock the motor in a fixed position, smoothly brake the motor, disable the driving outputs, or switch between 60 degree and 120 degree commutating mode. When the run command is selected, the output phase pre-drivers will generate pulse width-modulated signals that can drive the gates of a six transistor FET bridge which in turn can drive the phases of the motor. The purpose of the decode logic is to take the rotor position information (from Hall effect devices) at HS1, HS2, HS3 (Figure 3a), Pins 17, 16, and 15, together with the active state of ENABLE, CW/CCW, LOCK, BRAKE, and CYCLE to determine which of six outputs to turn on or off. The PWM duty cycle is then varied to regulate the current through the motor to a preset level controlled by the previous state of the DAC register.

Command Data Format

1. The RUN command is the normal mode of operation for driving the motor. The input format for selecting run mode is shown below (Bit 3 High):

clk cycles	1	2	3	4	5	6	7	8	9	10	11
cmd data	1	1	1	0	0	0	0	0	0	0	0

Active on falling edge of 11th clock.

2. BRAKE ENABLE: The brake enable command function allows a versatile way to brake the motor during a fault or power failure condition. For example, assume that previous commands have given the DAC a value above 0. The run command can be issued with the brake enable (Bit 5) register set to 1. The outputs will switch until RESET (Pin 4) is set to a Low state. This will reset the DAC register to 0 and cause the low-phase (sink) outputs P1L, P2L, and P3L to the High state, braking the motor to a halt.

clk cycles	1	2	3	4	5	6	7	8	9	10	11
cmd data	1	1	1	0	1	0	0	0	0	0	0

Active on falling edge of 11th clock.

3. BRAKE COMMAND: When set High, the brake command bit (Bit 4) forces the phase outputs to go into the brake condition im-

mediately. This may be used to gently brake the motor when a fault is detected by a software routine or a loss of motor control. When activated, the brake signal causes the 6 FET pre-driver outputs to go to the High state. This turns the p-channel drivers off and the n-channel drivers on. A moving motor will generate back EMF and current will flow through the n-channel drivers. The motor torque generated by this current opposes the motion of the motor, thus slowing it.

clk cycles	1	2	3	4	5	6	7	8	9	10	11
cmd data	1	1	1	1	1	0	0	0	0	0	0

Active on falling edge of 11th clock.

4. LOCK: When the lock bit (Bit 6) of the command register is set High, the outputs Phase 1H and Phase 3L will be pulse width-modulated regardless of the state of the Hall effect sensor inputs. The detent torque created will force the motor into a fixed position.

clk cycles	1	2	3	4	5	6	7	8	9	10	11
cmd data	1	1	1	1	1	1	0	0	0	0	0

Active on falling edge of 11th clock.

5. ENABLE: Enable is a digital on/off switch (Bit 3) and can be used to insure that the outputs are completely off when no drive is required.

clk cycles	1	2	3	4	5	6	7	8	9	10	11
cmd data enable)	1	1	1	0	0	0	0	0	0	0	0
cmd data disabled)	1	1	0	0	0	0	0	0	0	0	0

6. 60° or 120° Commutation Select (60-120): Setting this bit Low allows the Hall effect sensor inputs to commutate at 120 electrical degrees. When set High,

clk cycles	1	2	3	4	5	6	7	8	9	10	11
cmd data	1	1	1	0	0	0	1	0	0	0	0

the sensor inputs can be reconfigured for driving motors with 60 degree commutation cycles. This is done by connecting Hall sensor 2 to the Pin HS3. Hall sensor 3 is connected to Pin HS2. HS1 is unchanged.

The Sawtooth Oscillator

The oscillator develops a sawtooth waveform on pin R_T/C_T (Pin 7) by charging a capacitor C_T through the resistor R_T. The internal DC trip points of the oscillator are at V_{DD} and V_{DD}. At the end of each timing cycle, a low duty cycle output pulse is generated on “OSC OUT” (Pin 6). This output pulse sets the output of two internal R/S latches High, which makes the “CYCLE COMMAND” (internal) Active-High. This will allow the outputs to go to the last programmed state with respect to the condition of the Hall sensor inputs.

The Pulse Width Modulator (PWM) Comparator

Under current mode control, instantaneous current in the motor is summed at the input of the

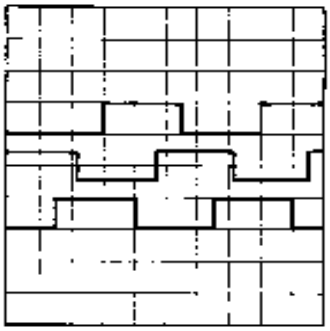
op amp (Pin 12) to obtain the current mode feedback signal. Thus the error loop controlling the PWM duty cycle is normally responsive to motor current (see Figure 4).

The output of the error amplifier (Pin 8) is compared by the PWM to the analog ramp of the oscillator. When the ramp voltage exceeds the voltage at the output of the error amplifier, one R/S latch is reset, the “Cycle Command”

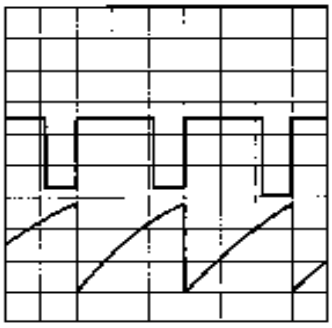
is then Low and the output drivers are turned off. This condition is maintained until the next “OSC OUT” pulse is again generated by the oscillator. This prevents double pulsing at the outputs due to switching noise at the inputs of the PWM within a cycle (see Figures 3b and c).

Overcurrent Protection

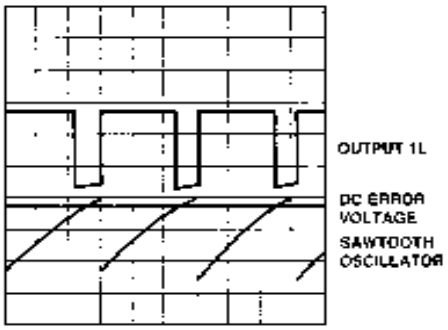
The current sense comparator input (Pin 14) allows the outputs to be shut off by resetting the other R/S (overcurrent) latch. Overcurrent protection is provided by the I-sense comparator with the input connected to a voltage node that is sensitive to the forward motor current. When the voltage at Pin 14 exceeds 0.5V, the R/S latch is reset and the outputs will be turned off until the next oscillator cycle. This provides cycle-by-cycle current limiting. The 0.5V reference voltage is derived from the external 5V reference at $DAC_{REF\ IN}$ (Pin 5).



a. Hall Sensor Commutation



b. PWM Duty Cycle Generation



c. Error Amplifier Output vs Sawtooth and Duty Cycle

SL00797

Figure 3.

NE5570 APPLICATIONS

Driving the Brushless Motor Commutation

Today's DC brushless motor is the result of an effort to improve overall DC motor reliability by eliminating contact brushes to the armature. A secondary result is that RFI is greatly reduced. Motor manufacturers are now working to improve the rotor magnetics with rare earth materials such as neodymium iron and samarium cobalt. Inherent in the process of removing brush-type mechanical commutation to the rotating magnetics is the introduction of some method of electrically switching field polarity synchronously with rotor position. Hall effect magnetic field sensors and the necessary logic for commutation are now built as an integral part of the motor. TTL-compatible outputs make interfacing to external controllers a simple matter. With just a few logic gates driven by the Hall signals (as shown in Figure 5), an

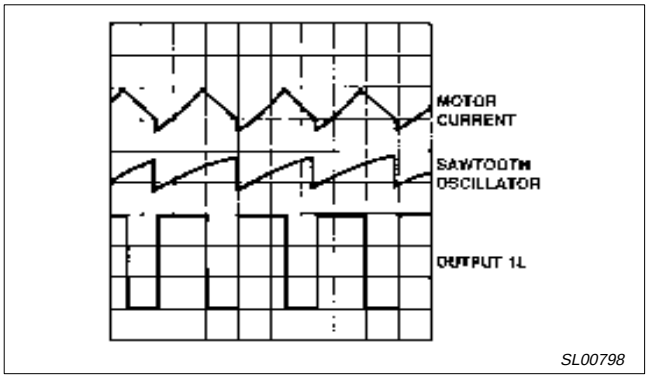


Figure 4. Motor Current vs Ramp and Duty Cycle

electrically-commutated brushless motor may be made to run on a single DC supply.

A Hall element gives a positive output when in close proximity to the north pole of a rotor magnet. The three-phase windings are connected in a star configuration with a 6-switch drive as shown in Figure 6. Programming the switches in the proper sequence will create a rotating vector field, driving the rotor in a clockwise or counter-clockwise direction. Phase signals are programmed as shown in the example.

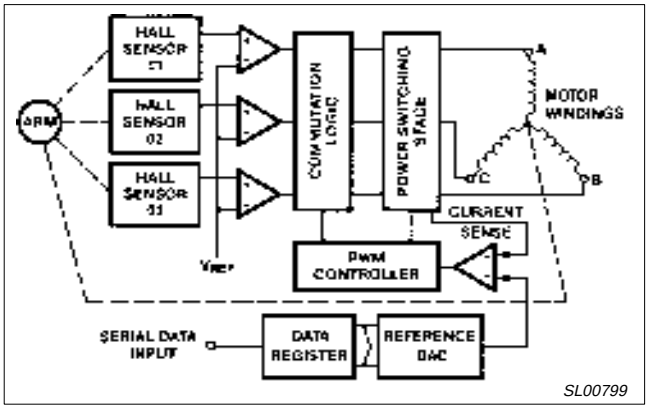


Figure 5. Basic Control Block

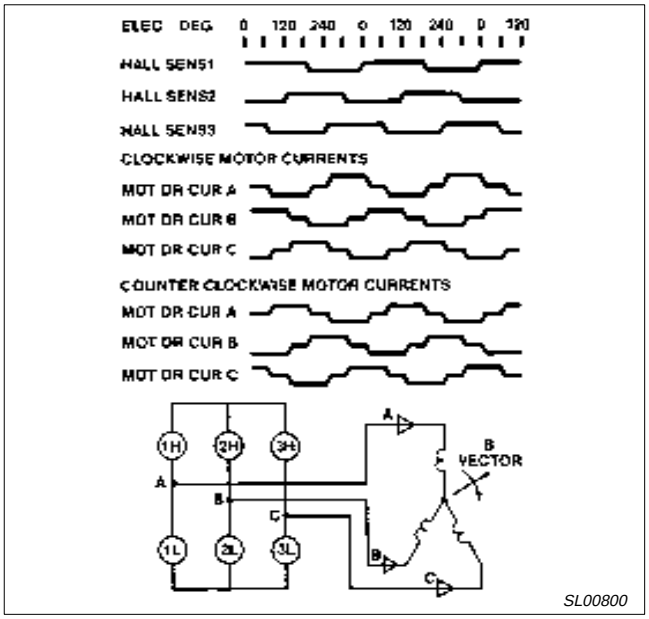


Figure 6. Motor Commutation

PWM Controls Motor Current

With the motor commutation provided from shaft position, field coils are switched in the proper manner to develop self-synchronous rotation. This differs from the AC induction motor which is dependent on line frequency and so operates at fixed speeds that are submultiples of 50, 60, or 400Hz. For example: the brushless motor, which is synchronous with the commutation signals, develops rotational velocity based on its speed-torque characteristics much as does a DC brush motor. Speed then must be approached as a function of torque, average current, acceleration and load inertia in addition to supply voltage. High efficiency results from controlling speed by means of pulse width-modulated drive to the field windings of the motor. Normally PWM frequency is set at a rate orders of magnitude above the commutation rate. Generally, this is determined by the motor inductance and field resistance. Ideal switching rates are above the audio range, that is, 15 to 20kHz and greater.

With a fixed supply voltage, a change in the PWM duty cycle controls the average current in the motor. This develops a control variable capable of determining motor torque under varying load conditions.

Instantaneous control of duty cycle allows acceleration modeling, making an algorithm under microprocessor control a repeatably accurate technique for various ramp-up/ramp-down subroutines.

Constant Torque with the NE5570

Using current sense feedback to force the brushless motor switching currents to null at some average fixed level results in a constant torque output. (See Figure 7.)

With the NE5570, input commands to the current control DAC force the PWM duty cycle to generate a fixed output current to the motor. Once a current control command is received by the NE5570, the DAC latches the data into an 8-bit register and the motor is under local closed-loop control. With this configuration, a remotely located microprocessor or serial command module can generate new torque control commands and the controller will instantaneously respond with the required output current level. The NE5570 also contains internal automatic cycle-by-cycle overcurrent protection to limit the duty cycle in case of an overload such as a stalled rotor condition. This feature is independent of the DAC control signal to the PWM. Current monitoring may be carried out by use of a simple resistive shunt in the driver current return leads, or a current transformer may be implemented for developing step-up voltage gain and lower impedance.

Constant Velocity (Voltage Mode) and Acceleration Control Programming

By providing velocity feedback in addition to a motor drive current loop, an additional degree of control may be added. Velocity feedback may be derived from the commutation signals in cases where low cost is of primary concern or a tach generator may be added if increased inertia is not prohibited. A shaft encoder may also be utilized if a direct microprocessor control loop is required.

Velocity feedback may be addressed in either of two ways: Digital, where the shaft speed and position are under constant monitoring by the microprocessor; Analog loop, in which case direct voltage feedback is summed within the PWM control loop as shown in Figures 8, 9, and 10.

Multiple Motors Under Serial Bus Control

The requirement to control a number of brushless motors of various sizes and at different physical locations represents an interesting challenge. An ideal solution is demonstrated in an example which uses the Inter-IC (I^2C) bus in modified form. The NMOS SCN8400 provides serial bus control using standard two-wire bus architecture, data plus clock; however, the NE5570 requires an additional chip enable signal input line (CE) in order to operate. As shown in Figure 11, this is easily implemented by using a separate I/O port signal line for each motor

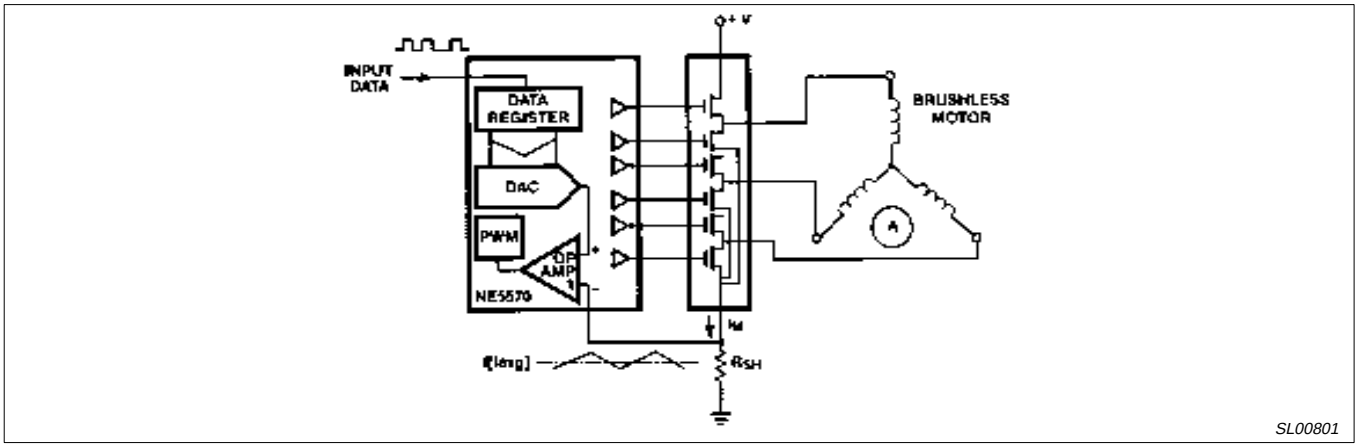


Figure 7. Current Mode Feedback Provides Constant Torque

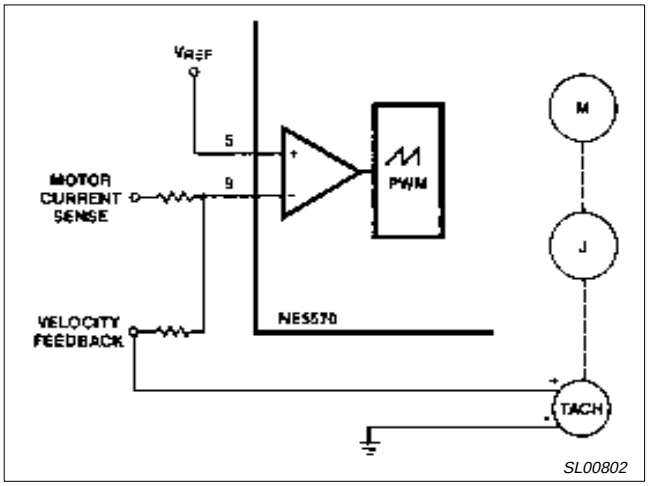


Figure 8. Summing Current and Velocity Signals in Local Analog Loop

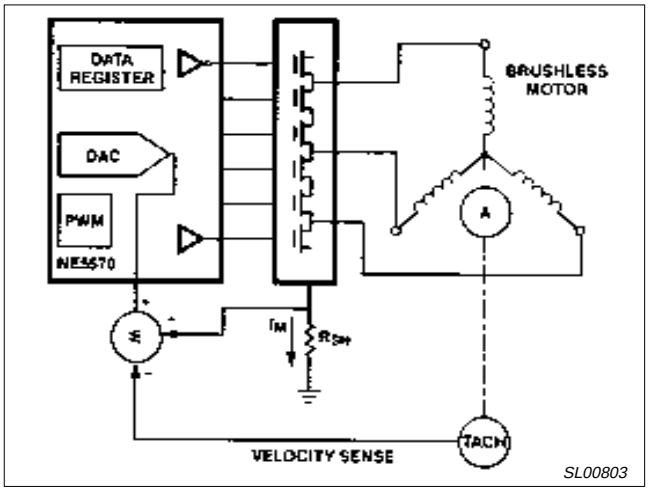


Figure 9. Two Loop Control With Local Feedback-Torque/Velocity

control module. To select a particular motor, the I/O port sends the CE signal which enables the respective NE5570 to receive the proper input data commands. This allows several motors within a radius of approximately 10 to 12 feet to be controlled by a single I^2C processor simultaneously. (See Figure 11.)

ESD Protection

The NE5570 must be protected from external power supply transients which exceed 16V — even for a few nanoseconds (see Figure 12). This is due to the presence of a snap-back clamp circuit connected from the supply pin (Pin 24) to ground. This device forms a negative resistance switch (NPN) which, if not current-limited after turn-on, can short the supply to ground. The resulting current obviously will destroy the device.

Power-Up Sequence

In order to insure that the outputs of the NE5570 are in defined states at power-up, some timing delays must be added externally. It is essential that the V_{REF} supply (Pin 5) come up before the 12V supply. Minimum delay of the 12V supply with respect to V_{REF} is 1µs. Second, reset must be programmed Low (active) for a period lasting until both the V_{DD} supply (Pin 24) and V_{REF} (Pin 5) are active.

By adding a PNP transistor, as shown in Figure 13, with an RC delay circuit connected from V_{REF} (+5V) to the base of the transistor, initial power-up starts with the condition that the base is at 0V and the transistor is in conduction shorting Pin 4 to Ground. As the base-emitter voltage approaches 0V, Q1 turns off, setting the internal logic in the proper states. Pin 4 is now accessible to external reset signals for normal programming.

A SERIAL BUS MICROPROCESSOR

Interface for the NE5570

Figure 14 shows a completed design example of a brushless motor controller which utilizes an SCN8400 microprocessor with a piggyback 2732 PROM as an automatic function generator. Command selection is via a small keypad attached to the PC board. The procedure calls for entry of the DAC commands in three-digit format with a range from 0 to 255. The complete controller is mounted on a single PC board as shown in Figure 15. Lock, Brake, and Disable commands, in addition to Reverse, are entered on separate keys from the DAC commands.

Clock and Data from the microprocessor (SCN8400) are all that comprise a normal I^2C bus using the standard Philips I^2C peripherals; however, the NE5570 requires a

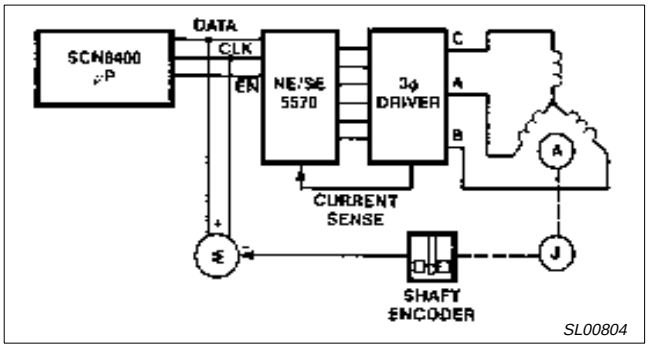


Figure 10. Velocity Feedback With Microprocessor ControllerResponse – Serial Bus

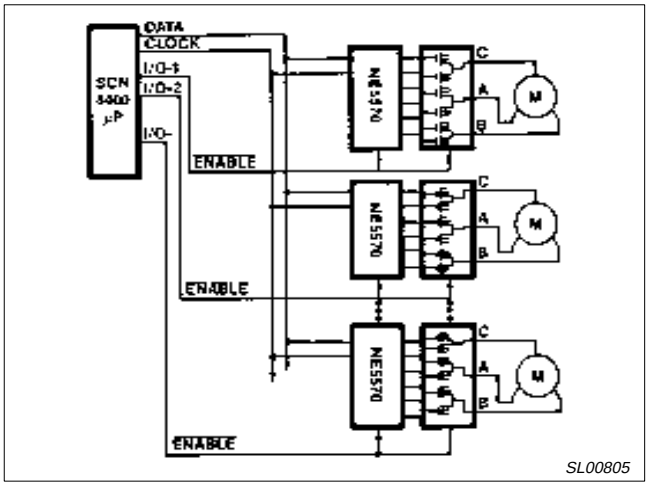


Figure 11. Multiple Controllers on Serial Bus

Chip Enable and this must be provided from a separate I/O port (Pin 27).

A Microprocessor-Controlled Servo with Torque or Velocity Feedback (Figure 14)

Input data loading of the command word is illustrated in Figure 14. After lowering the chip enable input (Pin 3), an 11-bit word is loaded in sequence. Data bits are latched during rising edges of the clock. After 11 clock pulses, the chip enable is deactivated by raising Pin 3 High.

The 11-bit data word is organized into two fields: a rotation field and a speed field. From Figure 15, Bits 1, 2, and 3 determine the direction of rotation; a 101 indicates clockwise, a 100 indicates counter-clockwise. Data Bits 4-11 are decoded as one of 256 possible speeds, from 11 to 255.

Commands are as follow:

Lock — On this command the motor magnets oppose each other and result in the motor coming to an immediate halt in a fixed position with detent torque. *H4Brake|During

motor operation, issuing the brake command causes the motor to come to an immediate halt, thus ending in a non-torqued or disabled state.

Disable — Issuing the disable command causes the driver FETs to immediately disengage and causes an immediate condition of zero torque. If issued during motor rotation, the motor will immediately cease to be driven and will slow to a stop from its own frictional losses.

Run — The run command tells the NE5570 to activate immediately using the command that is currently resident in the 11-bit shift register. When coming out of a Lock, Brake, or Disabled state, the NE5570 must be given both a speed/rotation command plus an additional run command to start.

The special function commands are defined by the 11-bit words shown in Figure 17.

Interfacing to the NE5570 may be accomplished by almost any microcontroller or computer through the use of 3 I/O port lines. Emulation of the motor commands may easily be implemented via

port commands. Multiple NE5570s may be implemented and controlled via a central processor by address decoding logic and the chip enable lines. As an example, Figure 14 illustrates the interfacing to an SCN8400 microcontroller via the I²C bus. The Data input line is connected to the SDA (serial data) pin on the SCN8400 microcontroller (Pin 2). The Clock input line from the NE5570 is connected to the serial clock pin (Pin 3), and the Chip Enable input is connected to Port 2, Bit 2 (Pin 27 on the 8400). To emulate the commands via the I²C port, a series of instructions are incorporated in a software routine called “motor” listed in Figure 18.

In this routine, the rotation control word is pre-loaded in the 3 MSBs of Register 1, and the remaining 8 bits defining speed are pre-loaded in Register r0. This routine accomplishes the following: disables any I²C device that may also be on the bus by issuing the “disable address” (it is a good idea to disable any CLIPs peripherals on the bus when using the bus for any non-I²C operations); second, toggles the chip enable High-Low-High once (an optional procedure to ensure that the shift register is initialized); third, programs the interface for a no-acknowledge mode (this tells the microcontroller to not generate an extra clock pulse for a peripheral acknowledgement; the NE5570 does not generate an acknowledgement). The routine then activates the NE5570 via port instructions, and transmits a series of 11 bits, the 3 MSBs from register f1, and all 8 bits from register r0 in high-bit to low-bit order. Assuming that these registers are properly loaded with a correct motor command, all motor control possibilities can be transmitted via this routine. To finish, the routine deactivates the chip enable line, “wakes up” the CLIPs peripherals that may be on the bus by issuing a STOP command, restores the normal I²C mode of operation, and returns.

This configuration may be extended to control multiple NE5570s by simply utilizing different port lines connected directly or through a decoder to each chip enable line on the motor controller ICs. The same routine may be used; just activate the appropriate chip enables. For additional serially-loading peripherals, refer to application notes AN163 and AN1681 “TEA1017: Using the C-Bus Devices on the I²C Bus”.

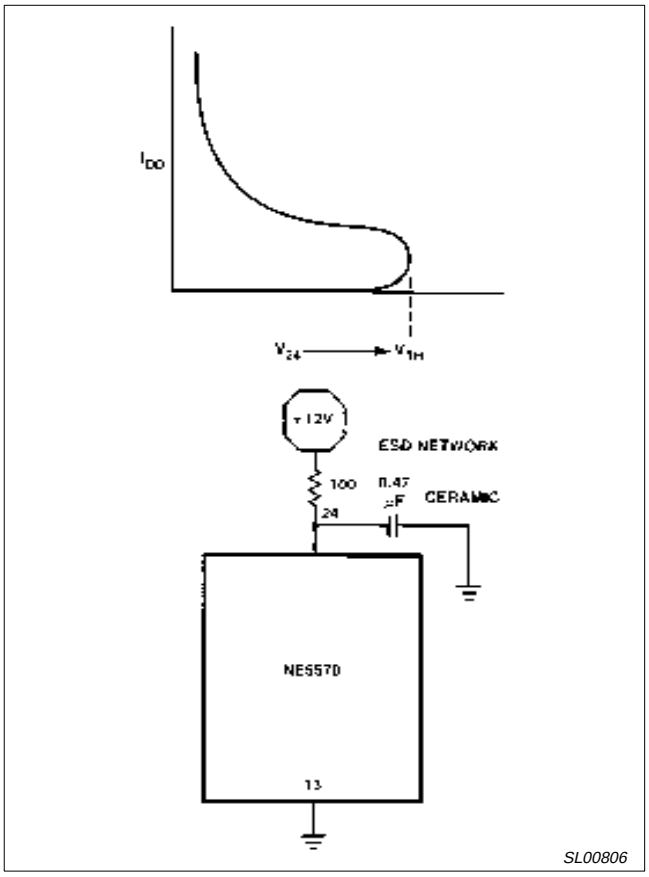


Figure 12. Recommended Protection Circuit for NE5570

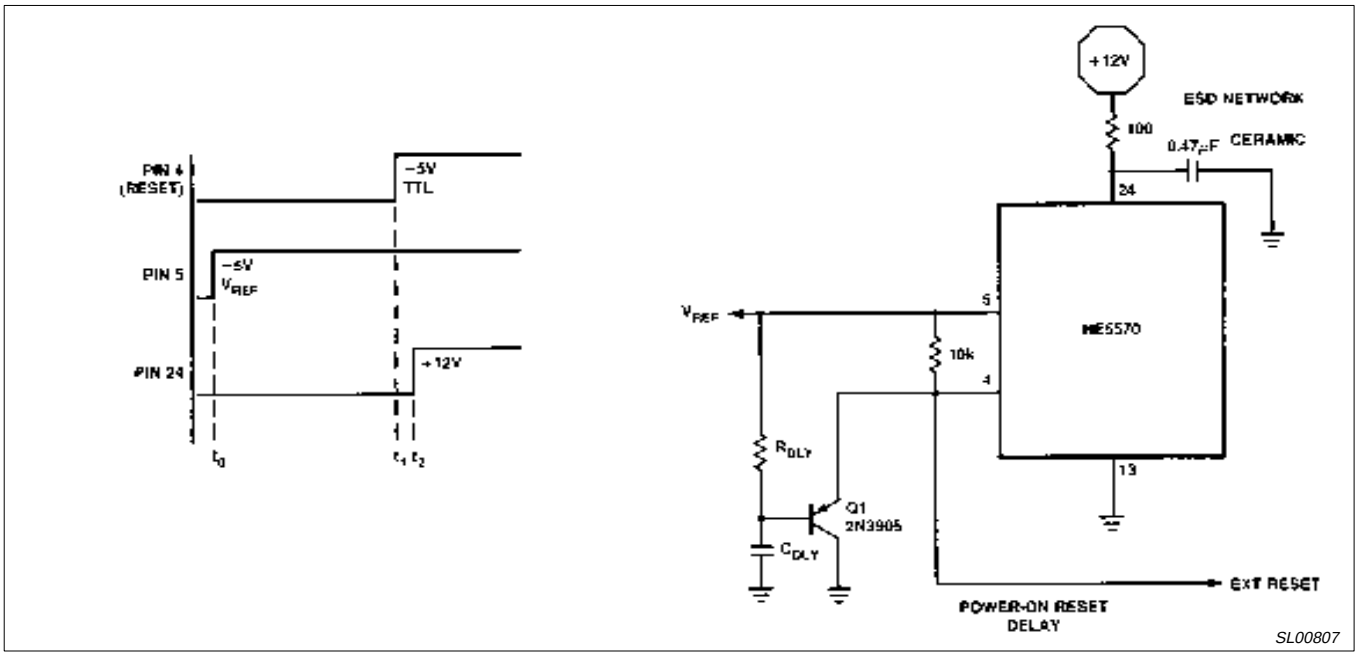


Figure 13. Supply and Reference Start Sequence

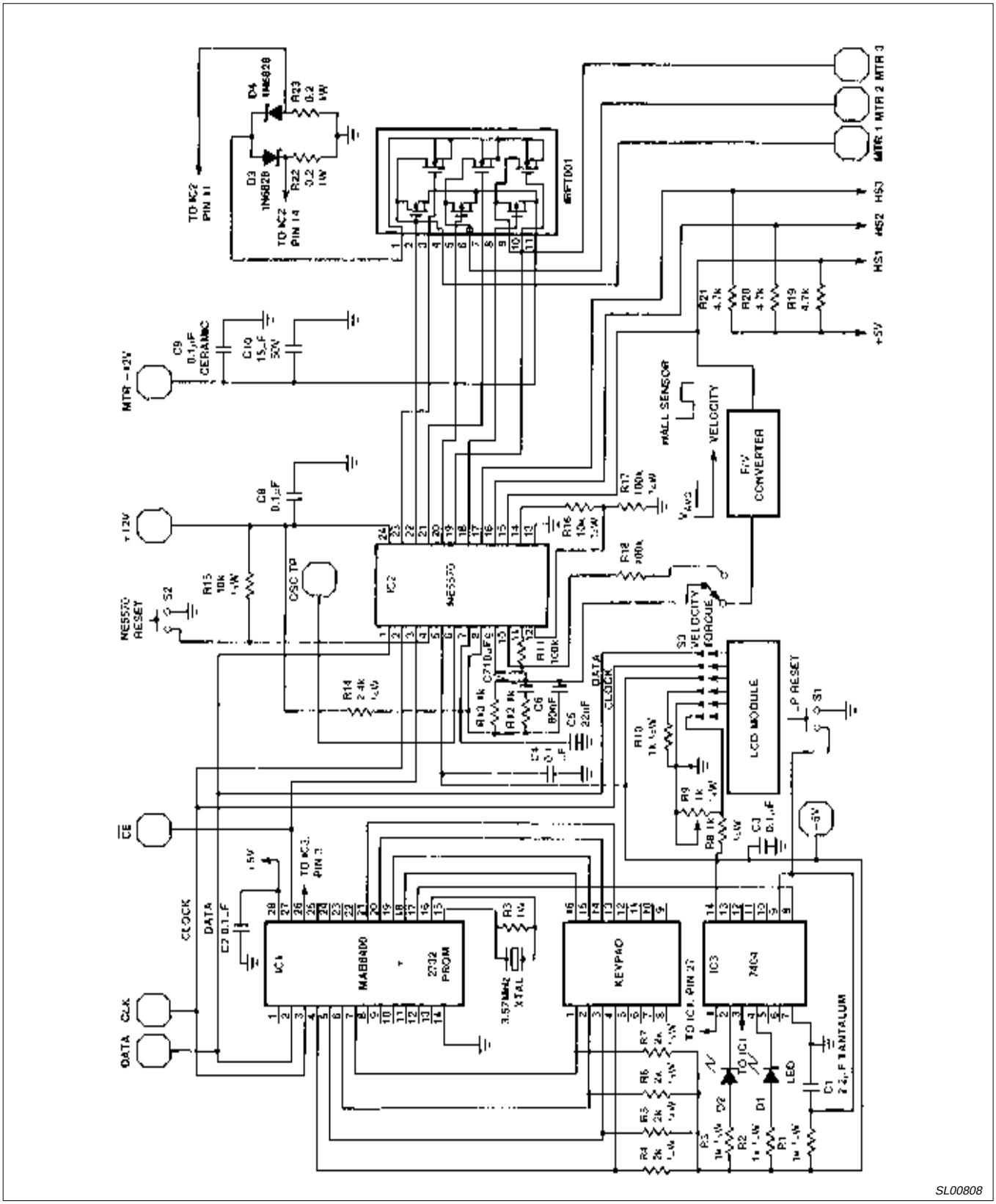


Figure 14. SCN8400 Serial Bus Microprocessor Controller

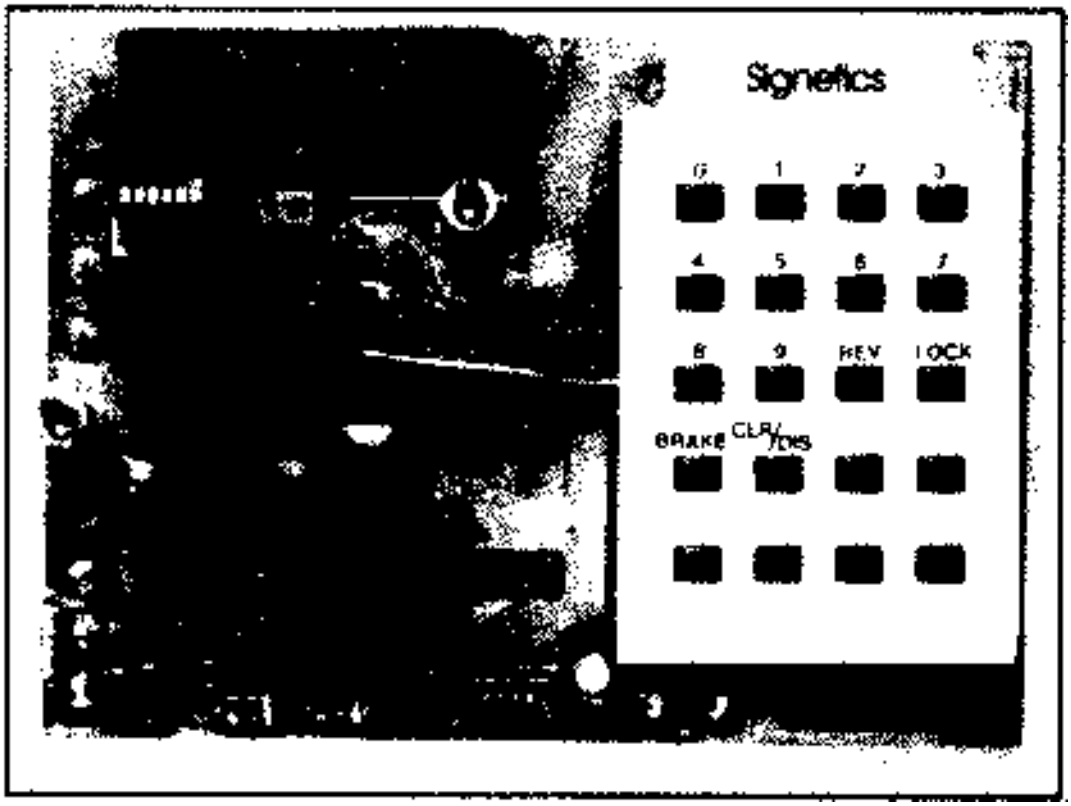


Figure 15. Constant Current Motor Drive With the NE5570

NOTES:

1. Fenger, Carl, “AN163: Small Area Networks Using Serial Data Transfer,” Philips Semiconductors Linear Data and Applications Manual, Volume II, Sunnyvale, CA, 1985.
2. Fenger, Carl, “AN1681: TEA1017: Using the C-Bus Devices on the I2C Bus, “Linear Division, Philips Semiconductors Corporation, 1986.
3. Holt, Charles A., Introduction to Electromagnetic Fields and Waves; John Wiley and Sons, 1963.
4. Rutkowski, George B., Handbook of Integrated Circuit Operational Amplifiers; Prentice Hall, Inc., 1975.
5. Airpax Brushless Motors-Model K85901; Pub. Catalog CMO-834, 1983, Airpax Corporation, Cheshire, CN.
6. MotorCon ’85, Official Proceedings of the 7th International Conference, Chicago, IL, Intertec Communications, Inc., Oxnard, CA.
7. PCI ’82, Volume 4, Conference Proceedings, San Francisco, CA, Intertec Communications, Inc., Oxnard, CA.

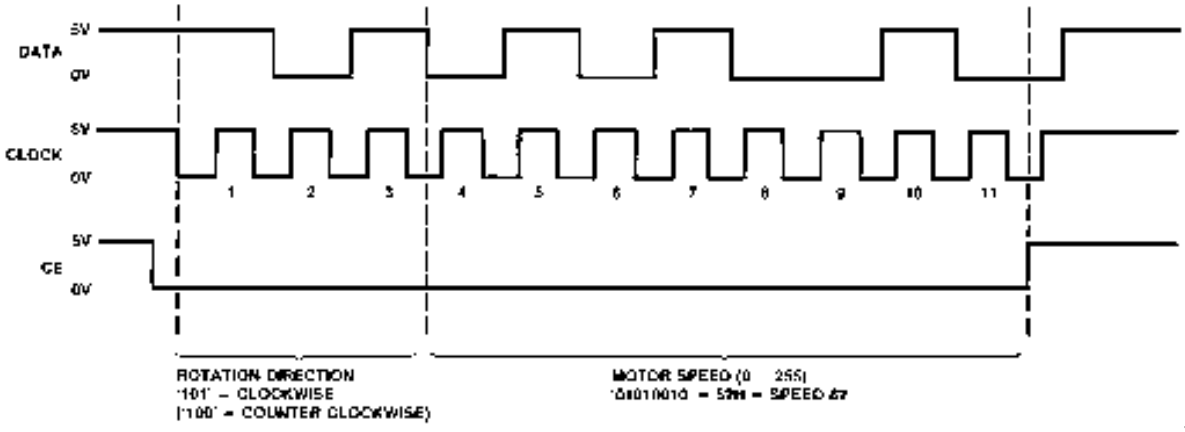


Figure 16. A Typical Command for the NE5570 Motor Controller

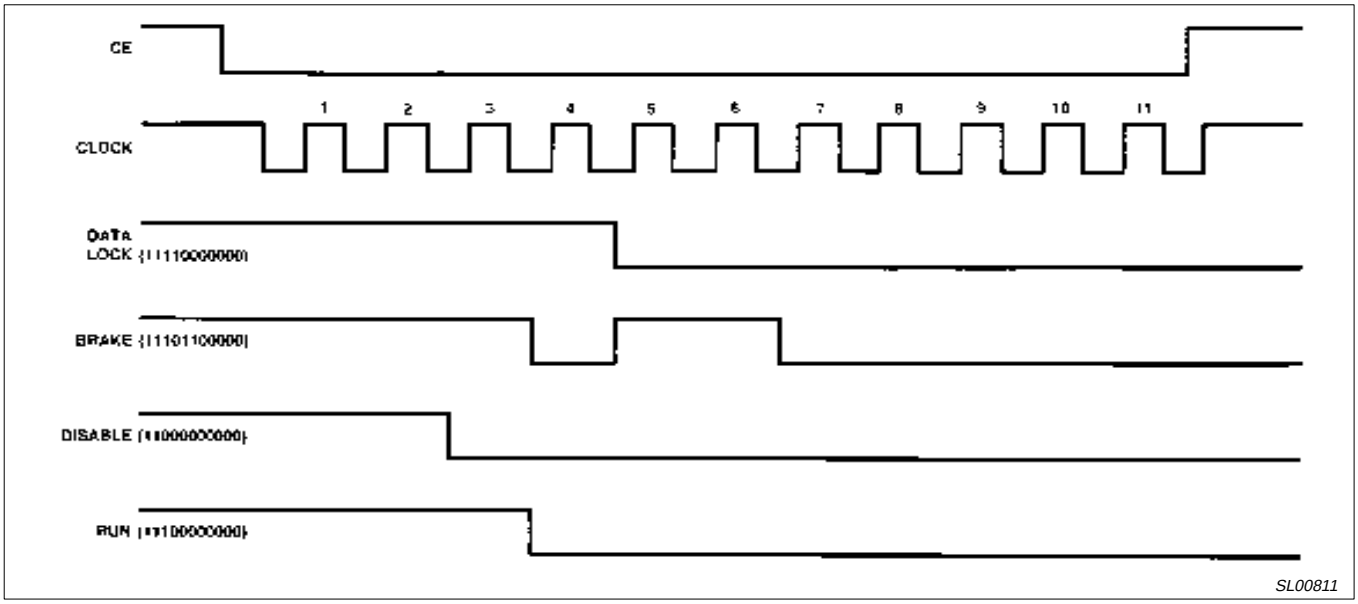


Figure 17. NE5560 Special Function Commands

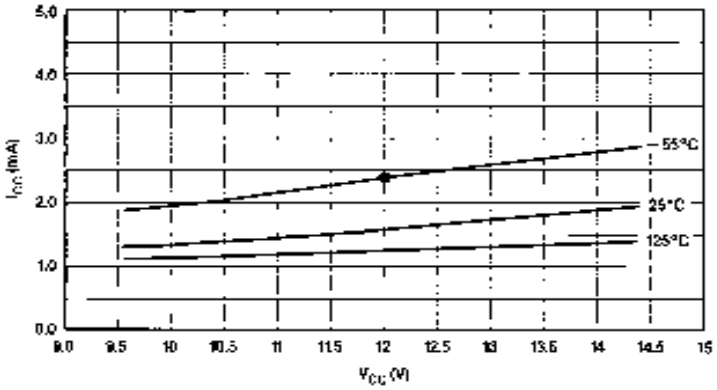
```
Motor           mov s1, #H'18'      ;Reset S10 status.
                mov s0, #H'02'      ;Load iic disable.
                mov s1, #H'0F8'     ; Start condition
                call pinwt          ;
                orL p2, #H'02'      ;
                nop                 ;
                anL p2, #H'0FD'     ;Clear motor controller
                nop                 ;
                orL p2, #H'02'      ;
            *
            *   mov s1, #H'18'      ;Initial bus status register to a know
                                    ;state.
                mov s2, #H'07'      ;Leave acknowledge mode
            *                       ;and program bus clock frequency
                                    (45kHz).

                mov s1, #H'0EB'     ;Prepare for a 3-bit transmission.
                mov a, rl           ;send rotation command.
                mov s0, a           ;Transmit first 3-bits.
                call pinwt          ;Wait for transmission complete.
                mov a, r0           ;Get speed command.
                mov s0, a           ;Transmit.
                call pinwt          ;Wait for transmission complete.
                orL p2, #H'02'      ;Deactivate.
                mov s1, #H'0D8'     ;Stop condition
                mov s1, #H'18'      ;reset sio
                mov s2, #H'41'      ;restore ack mode.
                ret
            *
            *
pinwt           mov a, s1           ;Wait for end of serial transmission
                jb4 pinwt           ;via polling of PIN bi in reg s1.
                ret                 ;
```

SL00812

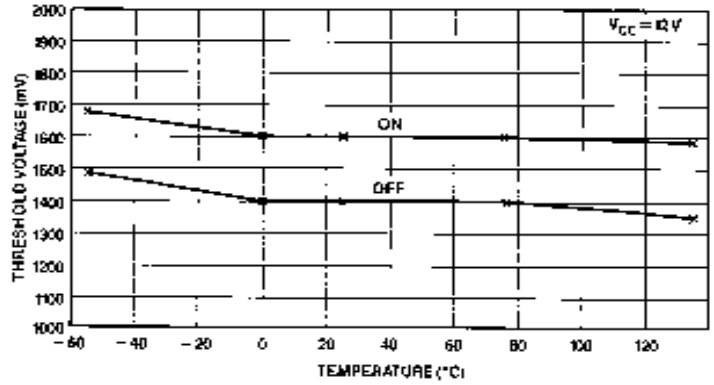
Figure 18. Motor-Controller Routine for Use on the I^2C Bus

TYPICAL PERFORMANCE CHARACTERISTICS

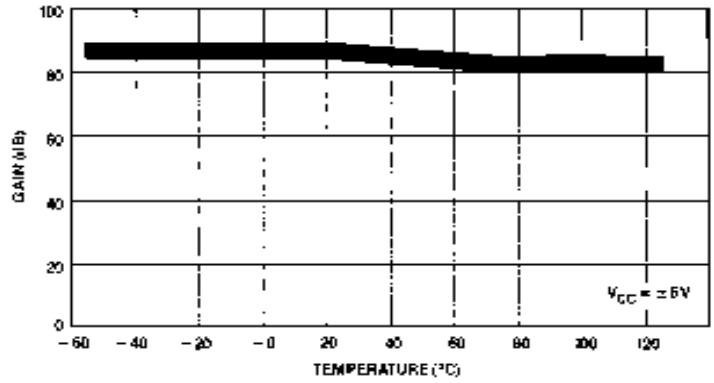
I_{CC} vs V_{CC}



NE5570 Logic Threshold Voltage vs Temperature



NE5570 Op Amp Gain vs Temperature

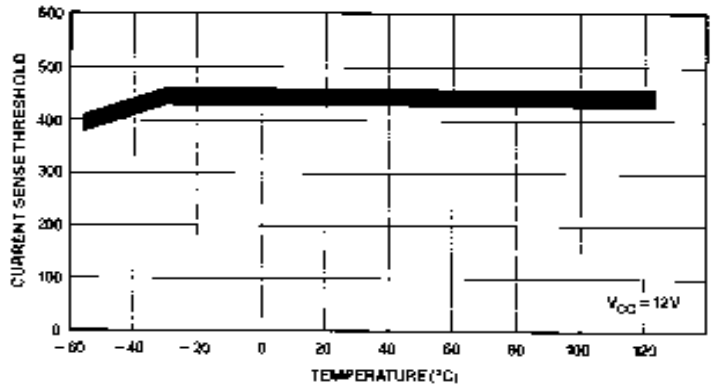


SL00813

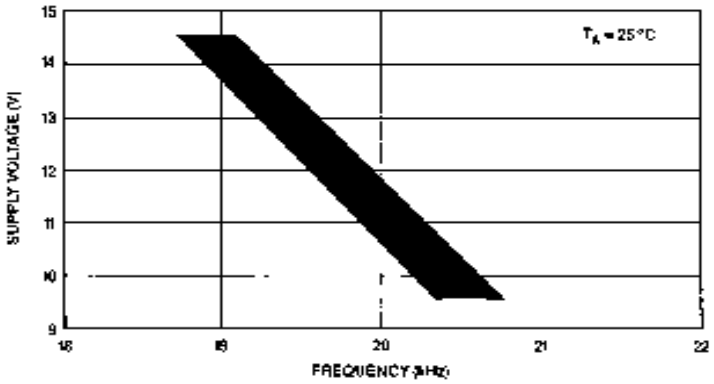
Figure 19. Typical Performance Characteristics

TYPICAL PERFORMANCE CHARACTERISTICS (Continued)

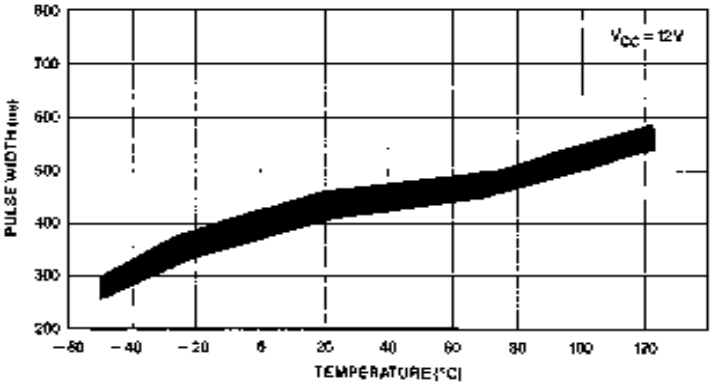
NE5570 Current Sense Comparator Threshold vs Temperature



NE5570 Oscillator Frequency vs Supply Voltage



NE5570 Oscillator Output Pulse Width

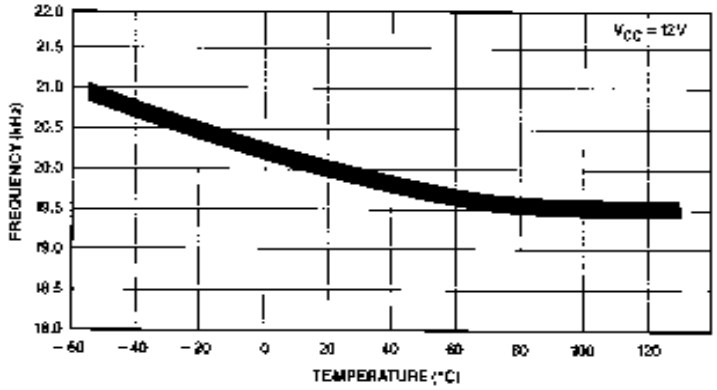


SL00814

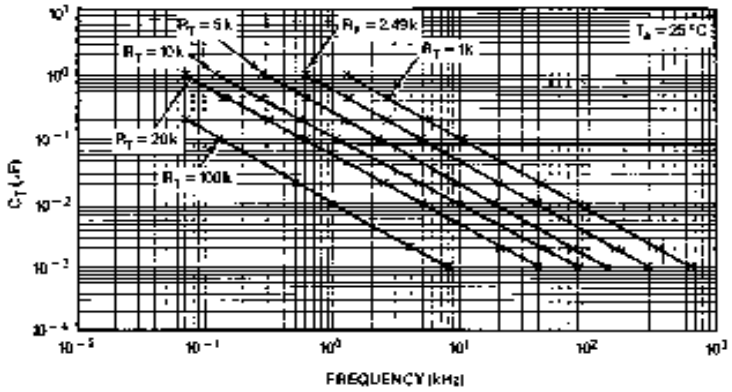
Figure 20. Typical Performance Characteristics (cont.)

TYPICAL PERFORMANCE CHARACTERISTICS (Continued)

NE5570 Oscillator Frequency vs Temperature



Oscillator Frequency vs R_T and C_T



SL00815

Figure 21. Typical Performance Characteristics (cont.)