A Simple Multi-Axis Stepper Motor Controller Using a PC Parallel Port

Paul J. Wilder,[1] James D. McBrayer,[2] and Kenneth A. Woechan[3]

Abstract

A custom web-enabled stepper motor control software program is presented for the multi-axis control of stepper motors via the parallel port of a personal computer (PC). A general interface circuit is also presented that allows control of step size and rotational direction of the associated stepper motor. The computer transmits a byte to the interface board via the parallel port and standard cable. Each word contains information on step size, direction of rotation, and a clock pulse for one step. An open loop control design was chosen for simplicity and low cost. The software runs on a PC or compatible computer, and is accessible through a web browser, either locally to the PC or via the Internet. The software and printed circuit board Gerber files, for photo-plots, will be available to educational institutions. The project can easily be incorporated into design, electronics, or mechanical curriculums for little expense. Relatively few components are required. The project also has applications in industry as an introductory tutorial on stepper motors and computer numerical control (CNC).

Index Terms - stepper motor, parallel port, personal computer control, multi-axis control

Introduction

This paper proposes a system to control stepper motors via the parallel port on a personal computer. The system is designed as an educational aid for use in engineering and technology programs. Implementation is easy using readily acquired components/off-the-shelf equipment found at your local electromechanical salvage house. Software to control system operation will be available to educational institutions for the cost of duplicating the diskette or by the Internet. The system can be added as a self-contained lab, or used as a supplement to existing courses to add diversity. The web-enabled software has three sections. The first section is a tutorial on stepper motor construction and theory of operation. Section two is an interactive control system, which allows the student to control one or more stepper motors by writing a short list of standard Gerber codes (*g* codes), or simply using the computer keyboard to input control commands. The third section is an optional quiz section that tests student comprehension of the tutorial through multiple choice questions. Student files are accessible only via password entry.

[1] School of Engineering Technology, University of Southern Mississippi, Gautier, MS 39553

[2] Department of Mechanical, Materials, and Aerospace Engineering, University of Central Florida, Orlando, FL 32816

[3] Senior Systems Engineer, Litton Integrated Systems, Ocean Springs, MS 39564

Overview

In the last decade the use of stepper motors for precision control in manufacturing, printing, robotics, and many other related areas has increased dramatically. Looking into a dot matrix or ink jet printer one finds at least two stepper motors. One for rotating the platen and one for print head positioning. Stepper motors are used anywhere precise positioning is required.

Many off-the-shelf options exist for controlling stepper motors. These range from stand-alone or PC based single axis controllers to complex multi-axial control versions. While offering many excellent features, most systems available are expensive. Prices start in the middle hundreds and rise into the thousands of dollars from there.

The system presented in this paper can be used to demonstrate stepper motor operation to engineering and technology students in a variety of programs. It can be constructed on a push-in board for a nominal component cost, or the students can fabricate a printed circuit board if their institution has the facilities to support board level manufacturing.

The system communicates with a PC via the standard parallel port. This port is typically used to communicate with a printer, plotter, or some other peripheral device. Eight data lines and nine control lines are used to transfer and receive information from the peripheral device.

Previous Work

As stated above, many ready built options exist for PC control of stepper motors. These typically require a bus card residing on the PC motherboard for I/O purposes. Some may include additional interface boards to connect between the PC and the stepper motor.

Zoghi and Lorenze[1] have developed a PC based system that inputs a digitized voltage from a slide resistor (POT), and converts this linear motion into rotational motion via a stepper motor. The system uses a PC's parallel port to drive power transistors, which activate the stepper motor coil circuits. This system has limited resolution as a result of the discrete nature of the windings in the POT. The smallest resolution the POT can have is one winding. Depending on the POT and A/D converter this may not translate into a step resolution equal to one step on the motor. Also, the computer's parallel port is not isolated from the interface board's electronics. The parallel port is left unprotected from inductive voltage spikes and noise that can be potentially damaging to the port structure.

Barbarello[2] suggests a partially automated PC based drilling system. Two stepper motors control the X and Y-axes of the drill, while the operator controls the Z, or plunging, axis. Barbarello's system also lacks isolation between the computer's parallel port and the discrete components in its control circuitry.

The two designs discussed above require the use of four data lines to control a single stepper motor. This implies that a single parallel port can control a maximum of two stepper motors. As will be shown later, the proposed system uses one data line to control the clocking of a stepper motor. This means that multiple stepper motors can be controlled simultaneously.

Perusich[3] proposes a system to use the PC to control a digital-to-analog (D/A) converter via the parallel port. The system has the advantage of isolating the parallel port from the interface board, but uses all eight parallel port data lines to control one component.

All three of the above designs use the Disk Operating System (DOS) to allow semi-automated control of the systems via the parallel port. None of the designs allow remote access. Through the use of its client/server architecture, the purposed system can be remotely accessed.

Proposed Solution

A web-enabled system is proposed, which provides tutorials, testing, and hands-on experience with stepper motor control. This system integrates both hardware and software into a single interactive educational tool for teaching students about stepper motors and their control. An easy to use system can be created, using system level software to handle stepper motor control. The software uses browser technology, allowing interactive student participation, from both local and remote locations. The software includes an interactive tutorial that students can use to learn and understand the theory of operation, and the construction, of stepper motors. The software handles parallel port management, as well as the control decoding for the interface board.

Software Design

The proposed system uses software to control the stepper motor, versus previous designs that use DOS based software. This design allows the students to access the program via the Internet. The program is written in Visual Basic Professional, Version 6.0. The program runs under version 4.0 WinNT of the Windows operating system. It uses Internet Explorer 5.0 as a web browser, for it's human-machine interface (HMI). The NTPort Library 2.1[4] parallel port device driver is used to enhance the lack of discrete parallel port control in Visual Basic running under a Windows NT environment.

As previously stated, the control software is made up of three sections. Section one is a tutorial on the stepper motor construction and theory of operation. Students read about stepper motor design and construction techniques. The information is supported by figures and graphic animation where possible. Topics covered include operation, motor speed, step resolution, rotation direction, magnetics, armatures, and control via the interface circuitry. On-line help is also included.

Section two is an interactive control system that allows the student to control one or more stepper motors by writing a short list of standard *g*-codes, or simply by using the computer keyboard to input control commands. The open loop design requires no feedback control, as the motor position is settable manually from the keyboard. *G*-codes are used in modern CNC machining and computer aided manufacturing. The *g*-code program is executed simply by selecting the *go* option from the menu. The *g*-code program is checked and an error message displayed if the system does not return to the starting position at the end of program execution. Possible program assignments include programming for: a rack and pinion steering controller; a track drive system for construction equipment; a print head positioning controller; or a conveyor belt materials handling application.

Section three is an optional test section that tests student comprehension of sections one and two. Twelve questions are randomly asked from a pool of sixty. The pool is divided into four sections of fifteen questions each. Three questions from each section are presented to the student. Questions are of the multiple-choice format. Student testing can be time limited. The system records all testing information in a secure file. Accessibility to student files is by password entry only.

Interface/Driver Circuit Design

Two PS2501 quad opto-coupler integrated circuits (ICs) are used for parallel port isolation from the interface/driver board electronics. Three motor driver kits, from E-LAB Digital Engineering, Inc.,[5] are used for project design simplicity and low cost. The kits include both the EDE1200 motor driver

IC, and the ULN2003A power IC. This package is capable of driving stepper motors with up to a 500 mA current requirement. The ULN2003A Power ICs are 5vdc Transistor-Transistor Logic (TTL) that use complementary metal oxide semiconductor (CMOS) technology. ULN2003A Power ICs are capable of operating over a 5 to 50V_{DC} range for V_{out}. The design incorporates light emitting diodes (LED) on each motor coil circuit for indicating activation. An optional blocking diode can be added for protection against reverse wiring of the power supply. The schematic diagram is shown in Figure 1 Interface Schematic.

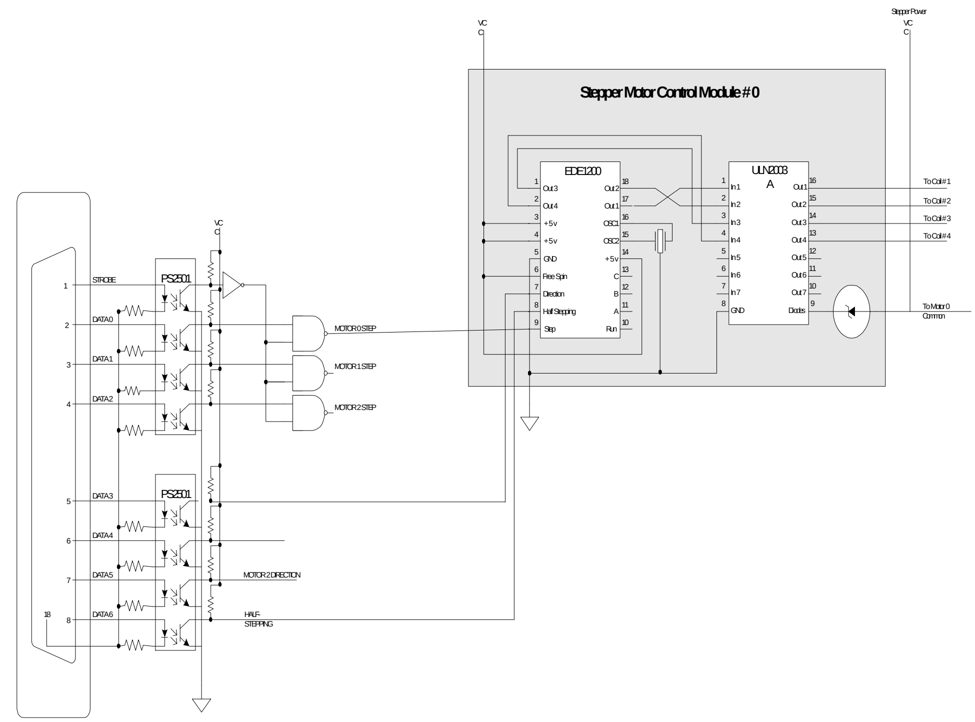

Figure 1 Interface Schematic

The EDE1200 controller IC requires three logic signals to control each stepper motor. The direction control determines rotational direction in either clockwise or counter-clockwise mode, and is in the counter-clockwise mode when the direction control is low. Each stepper motor's direction is individually controllable. The step signal is a negative edge triggered signal that controls the EDE1200's increment pulse. Each positive to negative transition of the clock increments the stepper motor one step in the direction determined by the direction control signal, with a step size determined by the full/half-step control signal. Stepper motors may be stepped individually. The full/half-step control determines the EDE1200 operation in the full/half-step mode, and is in half-step mode when the signal is low. The full/half-step control is the only control that is shared by all motors in the system.

The stepper motor interface circuit incorporates the logic that decodes the various stepper motor commands. Pin 1, the strobe signal, of the parallel port (LPT) is used as the step signal to each module. This signal is gated for individual control of the stepper motors. Three data signals, LPT pins 2, 3, and 4, enable the stepper motor on each of the three axes. The interface circuit logically ANDs the stepper motor designation bit with the strobe signal, to create the step signal for each module. LPT data pin 5, 6, and 7 are used as the EDE1200 direction control signals. LPT data pin 8 is used as the EDE1200 full/half-step increment control signal.

Arbitrary assignment of the stepper motor controls is as follows. Let LPT data bit 0 (Pin 2) be designated stepper motor 0 control; stepper motor number 0 shall be the X direction stepper motor. Also let LPT data bit 1 (Pin 3) be designated stepper motor 1 control; stepper motor number 1 shall be the Y direction stepper motor. Finally, let LPT data bit 2 (Pin 4) be designated stepper motor number 2 control; stepper motor number 2 shall be the Z direction stepper motor. The interface circuit allows individual control of rotational direction and step for each motor, while all share the same increment (full/half step). Three PF35T48041, 7.5-degree step angle, unipolar stepper motors are used. These motors use 5 VDC at 200 mA.

Conclusions

Implementing the Simple Multi-Axis Stepper Motor Controller system is accomplished using off-the-shelf equipment found at local electronics retailers and electromechanical salvage houses. The Simple Multi-Axis Stepper Motor Controller system is designed as an interactive educational aid for use in engineering and technology programs, where it can be added as a self contained lab or used as a supplement to existing courses to add diversity. The system also has applications in industry as an introductory tutorial on stepper motors and computer numerical control (CNC).

References

[1] Zoghi, Behbood, and Roger Lorenzo "PC-BASED OPEN LOOP STEPPER MOTOR," Unpublished manuscript, Texas A&M University.

[2] Perusich, Karl, and Andrew Stump (1992) "Using a PC Parallel Port as a Successive Approximation A/D Converter," Journal of Electrical Technology

[3] Barbarello, James (1997) *PC Hardware Projects Volume 2*, Prompt Publications, Indianapolis, IN.

[4] Hai Li (2000) "NTPort Library 2.1", URL http://www.zealsoftstudio.com/.

[5] E-LAB Digital Engineering, Inc. (2000), URL http://www.elabinc.com

Paul J. Wilder

Paul J. Wilder received the Bachelor of Science degree in Engineering from the University of Vermont in 1986, and the Master of Science degree in Computer Engineering from the University of Central Florida in 1996. He is currently an assistant professor of Engineering Technology, and the Computer, Electronics, and Software Engineering Technology program coordinator at the University of Southern Mississippi's Gulf Coast campus in Gautier, MS. From 1989-1991 he held a faculty position at Vermont Technical College in Randolph Center, VT. His research interests include multiprocessor design, memory design, and embedded microcontroller systems.

James D. McBrayer

James D. McBrayer is Professor and Program Director, Aerospace Engineering, in the Department of Mechanical, Materials, and Aerospace Engineering at the University of Central Florida in Orlando, Florida. Dr. McBrayer received the BS in Aeronautical Engineering and the MS in Engineering Mechanics from Saint Louis University, the diploma from the vonKarman Institute in Brussels, and the D.Sc. in Mechanical and Aerospace Engineering from Washington University in Saint Louis. His research interests include laboratory instrumentation for aerospace experiments via the WWW, aerodynamics, gyroplanes, and Wing-In-Ground effect vehicles. He is campus representative for ASEE and is a Fellow of ABET.

Kenneth A. Woechan

Ken Woechan has over 26 years of experience in system, hardware, and software design, development, deployment, documentation, and maintenance. Expertise includes mission/operational analysis, requirement definition, system allocation, software/ hardware allocation, software/hardware design/development, software/hardware test, system integration/test, final operational testing, and system maintenance. Application knowledge focuses on tactical and non-tactical communications, combat systems, intelligence collection systems and electronic warfare systems. Ken is completing a BS degree in engineering technology at the University of Southern Mississippi.