Servo To Go, Inc.

8117 Groton Lane

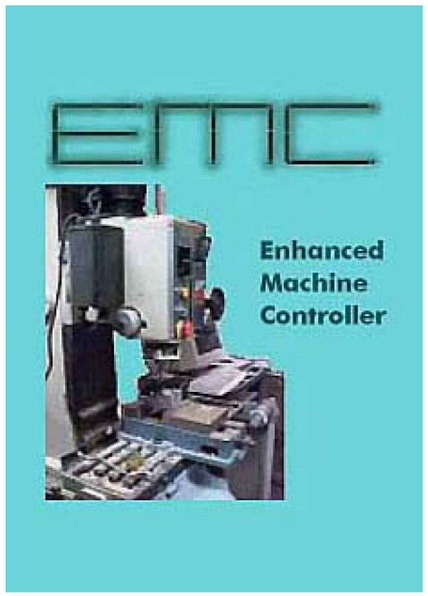
Indianapolis, IN 46260, USA

Phone: (317) 257-1655

Email: support@servotogo.com

http://www.servotogo.com

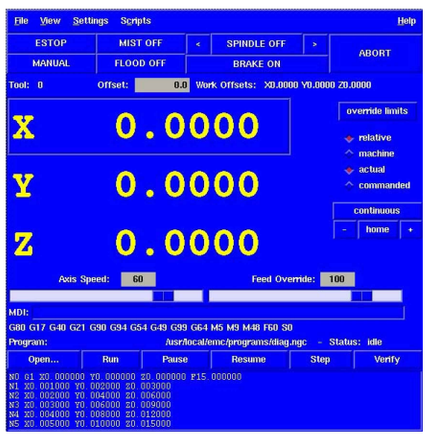
Need a machine control solution?



Servo To Go does EMC

see: www.linuxcnc.org

- Open source
- Active developer community
- Originally developed by NIST
- Comes standard with Servo To Go drivers



Control window from EMC running on RT-Linux

Potential Applications Include:

Robotics
Machine Tools
University Laboratories
Controls Research
Virtual Reality Actuators and Sensors
Motion Capture
Coordinate Measuring Machines
Motion Picture Camera Control
General Purpose Motion Related I/O

The full-length board is accessed by a set of registers located in the I/O space of the PC. The four main connectors are standard 50 pin IDC ribbon cables.

Connectors Details:

Connector Name	Pin Count	Description
P1	50	An Opto-22 compatible connector containing 24 bits of digital I/O.
P2	50	8 bits of digital I/O, 8 channels of analog input plus miscellaneous signals such as watchdog out.
P3	50	4 Channels of quadrature encoder input and analog output
P4	50	4 Channels of quadrature encoder input and analog output
P5	2	Optional auxiliary battery input. Maintains encoder counts when power is off.

Available Software

Windows Driver

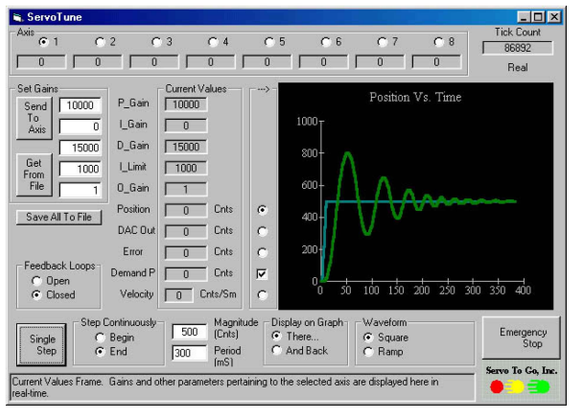
We have developed a VxD driver for Windows 95/98 and a kernel mode driver for Windows NT/2000. Both appear identical to an application. The driver includes a PID control algorithm and a linear trajectory generator. An interrupt service routine responds to an interrupt generated from the card. Relying on hardware interrupts gives reasonable response; whereas approaches that rely on the operating system for timing do not work near as well. Users can control up to eight axes of motion. The default sample period is one millisecond. Accessing the driver from Visual Basic or C is easy. Over 40 commands are available to the user. Examples include: SET_POSITION, SET_P_GAIN, SET_VELOCITY, and GET_ENCODER.

Visual Basic Application

The graphical servo tuning utility, “vbtune.exe”, is an example of using the driver with Visual Basic. But the program is a useful utility in its own right. The tuning utility lets users tune the gains of the servo algorithm by setting up either a step or ramp forcing function and observing the system response on a graph. A variety of variables can be graphed.

Win32 Console Application

A console mode application, “stgconap.exe”, demonstrates how to access the Windows driver from C++. This program is useful in its own right, especially during development. A text menu presents options such as to display encoder counts, display analog input, display digital inputs, set an analog output voltage, or set a digital output bit.



DOS Application

example, “stg.exe”, includes routines that access all functions of the card. It is essentially the same as the console application: it presents a menu to select various functions, although a few of the functions differ. The program exercises all the functionality of the board and includes a "dummy" interrupt service routine which users can modify to perform a servo algorithm. These routines can form the basis of a custom driver. The source code has been released to the public domain, so anyone can use the routines for any purpose.

DESIGN MOTIVATION

To Lower the Cost

Conventional motion control cards have an on-board processor. An eight axis system may require multiple cards—each of which costs more than the computer it goes into.

Much of this expense is for the on-board processor or custom I.C. used to perform motion control. In addition, the cost of hardware design and software development is much greater. Developers of embedded systems are scarce to begin with, and they do not have the powerful development tools that are available for developers in a PC environment. For a relatively low volume product such as servo boards, development cost is spread across only a limited number of units, adding to the cost of each one.

The conventional thinking was that an on-board processor was necessary because the main CPU in the PC did not have the processing power to perform the calculations in a reasonable amount of time. This was true in the past--but no longer.

Another argument for having an on-board processor is that, without it, a real-time operating system would be required on the main CPU. This is also changing, after all—multimedia IS real time.

An extremely fast servo loop update frequency may be necessary for some applications, but for many, an update rate of 1 millisecond is more than sufficient. This makes an on-board processor unnecessary for most applications.

To Give Users More Control

When a control algorithm is executed on-board the interface card, it typically becomes a proprietary component of the system and cannot be changed. Users have no control over what algorithm is used. Researchers cannot experiment with new algorithms. An input/output only approach opens the system to other servo algorithms, controls experimentation, and applications that just require I/O with no control algorithm at all.

So remove the processor, reduce the cost, reduce the complexity, you're left with more control. You can make this card do what you want to do. Sometimes the dumb card is the smart choice.

Simply stated: We simplified the board, reduced the cost, and increased user control.

SERVO TO GO, INC.

Price List

Servo I/O card models:

STGII-8 - 8 Axis Model II Card.....$888
STGII-Manual - Model II Manual ..(by request) N/C

Special order boards:

STGII-6 - 6 Axis Model II Card $848
STGII-4 - 4 Axis Model II Card $808
STGII-2 - 2 Axis Model II Card $768
S8 - 8 Axis Model I Card.....$888

The above are single quantity prices. Delivery is one to two weeks, with faster delivery possible, depending on order size and available inventory. All prices are in US dollars. Visa, Mastercard and Discover cards are accepted. International orders must be pre paid via check or credit card.

Software:

All software can be downloaded from our website. It is open source under the GNU Public License, except for the DOS code which we released to the public domain.

- DOS example which demonstrates how to access each feature of the board.
- Windows drivers for 95/98 and NT/2000 which have a closed loop servo algorithm in an interrupt service routine.
- Visual Basic Servo Tune Utility which demonstrates how to communicate with the Windows driver from Visual Basic.
- Console mode application which demonstrates how to communicate with the Windows driver from C++.

Note: The Windows 95 and Windows NT driver was developed using Microsoft's Visual C++ 5.0, additionally, Vireo's VtoolsD driver development toolkit was also used to develop the Windows 95 VxD. It may be necessary, depending on your application requirements, to purchase these tools in order to compile the source code.

Brief Hardware Technical Specification

The Servo To Go ISA Bus Servo 1/O card is a low cost, general purpose, motion control input and output board which can control up to eight motors simultaneously from an ISA-bus based computer such as an IBM compatible PC. The following is a summary of the hardware functionality:

Encoder Input
Up to 8 channels of quadrature encoder input, with index pulse.
24 bit counters, expandable to 32 bits or more via software.
Single-ended or differential (RS422 compatible) input signals.

Analog Output
Up to 8 channels of analog output.
± 10 volt range.
13 bit resolution.

Digital Input and Output
32 bits, configurable in various input and output combinations.
Connector is compatible with Opto-22 boards.

Analog Input
8 channels of analog input.
13 bit resolution.
Range ± l 0V or ± 5V jumper selectable.

Interval Timers
Capable of interrupting the PC.
Timer interval is programmable to 10 minutes in 25 microseconds increments.
One general purpose time available.

Battery Backup
Can maintain encoder counts when power is off.

Watchdog Timer
Selectable time out period .001, .005, .1, .3, or 1 second.