LA-UR 95-1130

Conf-9506169--1

Los Alamos National Laboratory is operated by the University of California for the United States Department of Energy under contract W-7405-ENG-36.

TITLE: OPEN ARCHITECTURE CNC SYSTEM

AUTHOR(S): Aaron A. Lopez, John M. Edwards, & Jacob Tal

SUBMITTED TO: 24th Annual Symposium on Incremental Motion Control Systems and Devices, sponsored by Incremental Motion Control Systems Society

Los Alamos

Los Alamos National Laboratory
Los Alamos, New Mexico 87545

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

Open Architecture CNC System

Jacob Tal

Galil Motion Control Inc., Sunnyvale Ca.

Aaron Lopez and John M. Edwards

Los Alamos National Laboratory, Los Alamos, N.M.

Introduction

Computer numeric control (CNC) systems have traditionally utilized closed architecture controllers. In such systems the controller hardware consists of a special purpose computer with unique hardware structure. Such controllers have custom software which provide control as well as the user interface.

Due to the customized design of the hardware and software, such systems were available only as one complete package. There was no way for the user to separate their functions, and certainly no way to change them. An open architecture system allows the user to modify the hardware, software, or both.

The last decade has seen major changes in the areas of personal computers and motion controllers. Today, it is possible to perform the CNC function with a general purpose personal computer and a general purpose motion controller, thereby allowing the use of open architecture CNC systems.

These new systems have two major advantages: cost reduction and flexibility. Since general purpose computers and controllers are manufactured in large quantities, they are available at a significantly lower price than a closed architecture dedicated computer system.

Flexibility is achieved through the development of a user interface and control software which is independent of the hardware. Thus specific features needed for customs systems can be included and modified. Furthermore, the control software may take advantage of improved capabilities of new motion controllers to create CNC systems with increased performance.

In order for such systems to function properly, it is necessary to define the functions of all the elements and the interface between them. The following presentation describes such a system. In order to keep the scope of the paper manageable, the discussion is limited to CNC systems of a vertical milling machine, and the discussion is limited to a minimum set of functions. The presentation is aimed at forming the structure that can be expanded to cover additional types of CNC systems.

Open Architecture Controller

Definition

A controller whose architecture is said to be open, is one which implies the following to be true.

The controller is comprised of modules that allow the mixing and matching of various compatible hardware and software components. The hardware and software components may be from different manufactures. The modules will lend themselves to be readily modified to meet a required application. The modules will be adequately documented to expedite the modification and implementation process.

The controller software code is written in a high level language such that it may be readily revised to the latest version, without having adverse affects on the operation of existing software and hardware combinations.

The controller is designed to conform with existing up-to-date controller industry design standards.

The controller interactive communication language will be one which complies with, but not limited to, the industry de facto: EIA RS-274.

Reasonably achievable standards must be adhered to, in the design of such a controller, in order to protect the health, environment, and well being of the user.

A 3-Axis Milling Machine

In order to achieve an understanding throughout this paper, the machine tool which will be discussed, is the 3-axis and spindle milling machine as shown in figure 1. This machine tool is defined by ANSI / EIA-267-C.

The x-axis shall be horizontal and parallel to the work holding surface. It shall be the principal axis of motion in the positioning plane of the tool or workpiece.

The y-axis direction shall be in the direction to make a right-hand set of coordinates.

The z-axis shall be vertical on a single column machine with rotating tools, positive x direction shall be to the right when looking from the front of the machine to the column.

Clockwise spindle rotation shall be in the direction to advance a right-handed screw into the workpiece when looking into the negative z-axis direction.

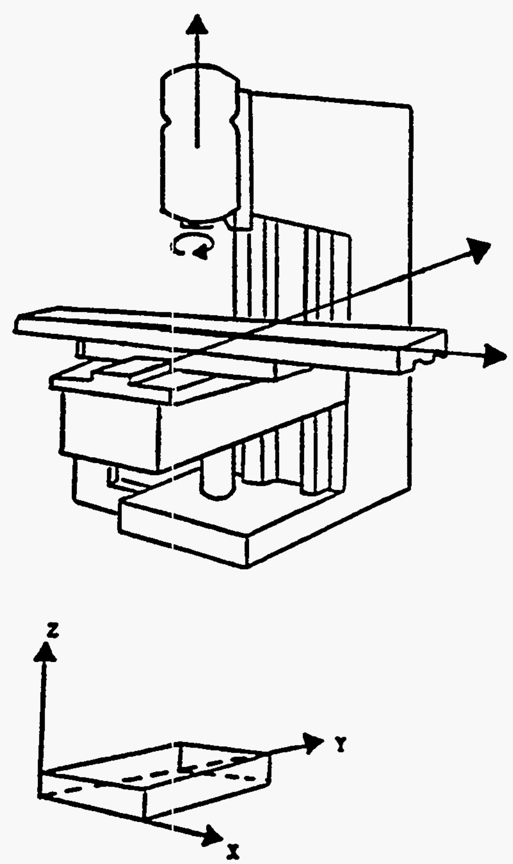


Figure 1
Vertical Mill Machine

The System Elements

The functional elements of a CNC system are illustrated in Figure 2. The first two elements, the user interface and the G-code translator are software programs performed at the personal computer level.

The third element is a general-purpose motion controller with firmware capable of receiving commands from the PC and controlling motors and processing inputs and outputs.

The user interface is the program which receives the inputs from the user and displays the information as required. This program varies significantly between different CNC systems and depends on the purpose of the specific machine.

To allow interface between various user interface programs and a variety of motion controllers, it is best to use a standard programming language, such as that defined by EIA RS-274 or commonly referred to as G-code. Such a language allows user interfaces of different forms with the same output. The desired motion is defined by a G-code file of standard instructions. Since motion controllers operate with different instructions, it is necessary to translate the G-code instruction to the controller language. In most cases the motion program is available as a file that must be translated. In other cases, the instructions may be manually entered one at a time into the controller and must be interpreted. In either case such a translation program is required between the user interface and the motion controller. The translation program may reside as part of a user application or as a separate program incorporated into the overall control package.

The motion controller, amplifiers and motors are standard components that are routinely used in motion control. The controller generates an output signal, typically an analog signal in the range between -10V and 10V. Such a signal is sent to the servo drive which, in turn generates a signal which is fed back to the controller, typically in the form of encoder counts in quadature, to allow closed-loop position control.

The motion controller may also incorporate non-motion features, enabling the user to manipulate external equipment such as pumps, valves, switches, solenoids ...

The following section illustrates a typical set of requirements for a vertical milling machine, and shows how the interface is accomplished.

Motion Control Requirements for a Milling Machine

A vertical milling machine is a system where a spinning tool moves along a certain trajectory and cuts the material, typically metal, to the desired form. To accomplish this function, the requirements are stated in terms of the elements.

To accomplish the required motion, the controller must be capable of generating three-dimensional motion along a straight line and a two-dimensional motion along circles. Furthermore, the generated motion must be accomplished at a programmable feed rate. In order to accomplish some auxiliary functions, such as tool selection, the controller must have the capability of reading some digital input signals and generating some output signals. Although the numbers of these lines vary greatly, a total number of 500 lines is typically sufficient.

The Functional Elements

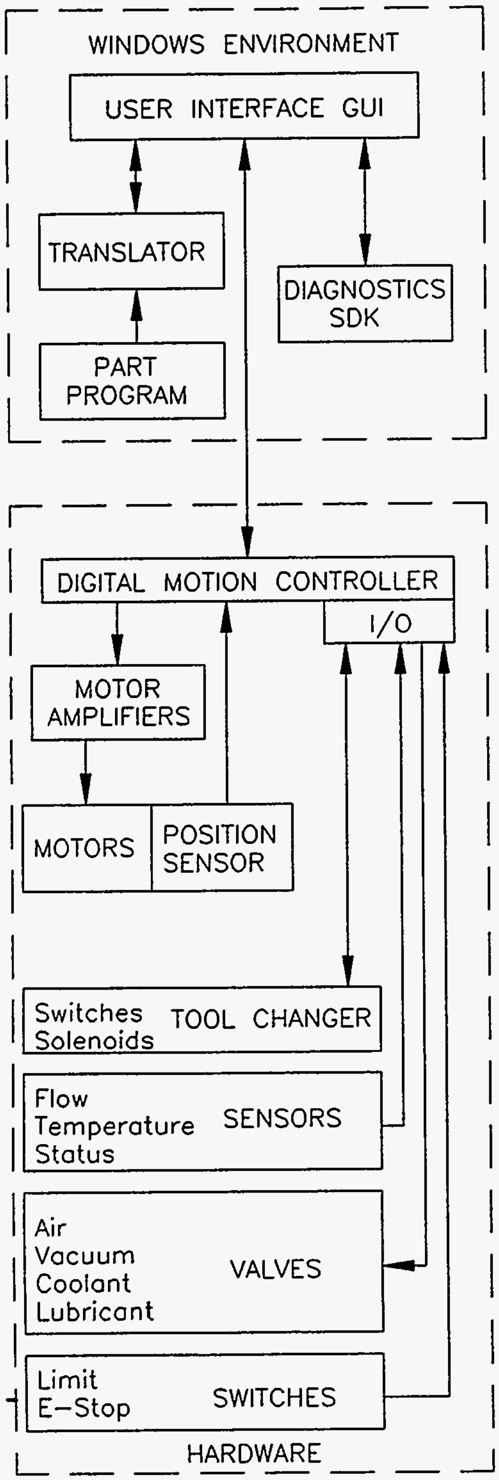


Figure 2
Functional Elements

The motion controller must also have the ability to read some auxiliary signals. These include the feed rate, which is usually set by a potentiometer or a multi position switch. The feed rate must be programmable and overridden by a potentiometer or multi position switch. The motion controller must also read the position of a hand wheel used for the position adjustment of the various axes.

The Controller Motion Requirements

Programmable motion control is the primary purpose of any CNC controller. The following functions are specific to the motion controller hardware and are the entity of the firmware.

Table I Motion Functions:

Abort motion
Acceleration rate
After distance trip-point
After motion trip-point
After absolute position trip-point
After relative distance trip-point
After elapsed time
After at speed trip-point
After vector distance trip-point
Begin motion
Circular segment
Deceleration
Define error limit
Acceleration feed forward
Find edge
Home
Jog Mode
Position absolute
Position relative

Servo Here (motors on)

Slew Speed (axis velocity)

Stop Motion (stop program)

Vector acceleration

Vector position

Vector speed

Programmable timer (dwell)

Independent time constant for smoothing

Linear interpolation end

Linear interpolation distance

Linear interpolation mode

Vector deceleration

Required Commands

G-code and M-codes are often modified and augmented by CNC manufacturers. Most G and M codes include instructions that are specific to each machine. In order to limit this discussion, and limit the scope of this example, we restrict the requirements to the following instructions : G, M, S, F, H, and T codes which are uniformly agree on by EIA / RS-274.

Interactive Language

G-code as defined by EIA / RS-274 is the industry standard for CNC machine tool controllers and will thus be the standard applied to a vertical milling machine using an open architecture motion controller which is PC based. The interactive language will be comprised of G words, and M words. Other words in the format will be as follows, End-of-block word, O word, S word, T word, F word and H word.

Tables II, III, and IV provide a listing of those words which are designed to provided the basic code to accomplish interactive communication with an open architecture controller and operator.

Required G and M Codes

Table II The G Words:

Instruction	**Definition**
G00	point to point positioning
G01	linear interpolation
G02	arc clockwise
G03	arc counter-clockwise
G04	dwell (wait)
G17	XY plane selection
G18	ZX plane selection
G19	YZ plane selection
G20-32	unassigned
G40	tool offset cancel
G41	tool offset left
G42	tool offset right
G43	cutter offset, inside corner
G50-59	reserved for adaptive control
G44	cutter offset, outside corner
G64	continuous mode
G70	inch programming
G71	metric programming
G72	circular interpolation-cw
G73	circular interpolation-ccw
G74	cancel circular interpolation
G75	select circular interpolation
G81-89	fixed cycles
G90	absolute dimension input
G91	incremental dimension input
G92	define relative origin
G94	in - mm per minute feed rate
G95	in-mm per spindle revolution
G96	constant surface speed ft.-meters

Table III The M Words:

M00	end of program
M01	pause
M02	end of program (subroutine)
M03	spindle cw
M04	spindle ccw
M05	spindle off
M06	tool change
M07	coolant #2 on
M08	coolant #1 on
M09	coolant off
M10	clamp tool
M 11	unclamp tool
M 19	Oriented spindle stop
M20-29, 50-57, 60-89	user defined I / O
M30	end of data and rewind
M98	execute subroutine
M99	end of file

Table IV Other Words:

End-of-block	represented by the semicolon (;)
O word	first word in a program and assigns the programs number
S word	sets the desired spindle speed in RPM as specified in the set-up parameters
T word	sets the tool magazine in position for the next tool change operation: M06
H word	sets the desired tool length and radius compensation with the tool selected using a T word
F word	sets the desired feed rate as specified in the set-up parameters

User Interface Requirements

The user interface program allows the user to specify motion requirements, and displays information for monitoring motion. In most cases, motion requirements are specified prior to execution of motion and are available as a G-code file that can be stored on a disk. An alternative method is one where the operator specifies motion requirements and edits the G-code program through the user interface.

Displayed information varies according to the purpose of the machine. As a minimum, the display illustrates the position of the three axis. Often the display includes the spindle speed and the G-code instruction which is being executed. As well as displaying position, absolute, relative, Distance to go, and machine coordinates, which are set-up during homing.

Controller Non Motion Requirements

During the operation of a vertical milling machine it is necessary to have capability to operate and utilize features that are not directly controlling motion. These features are required to monitor as well as control devices which are external to the machine tool, but are essential to the fabrication process.

Controller Setup Parameters

Prior to operating a machine tool such as a vertical milling machine, the integrator must setup each specific operation with relation to the machine tool mechanics. Features available at the graphical user interface must also be setup with respect to the functionality of the interface program. These GUI features should have the flexibility to be implemented by the user if so desired. This setup procedure is easily accomplished through a window with editing capability. The following setup parameters are a listing to be included in such a window.

Table V Required Setup Parameters

Counts to the inch x-axis	Counts / inch
Counts to the inch y-axis	Counts / inch
Counts to the inch z-axis	Counts / inch
Counts to the inch spindle	Counts / inch
Maximum vector feed rate	inches / minute
Homing Speed	inches / minute
Maximum x-axis traverse	inches / minute
Maximum y-axis traverse	inches / minute
Maximum z-axis traverse	inches / minute
Maximum spindle speed	counts / minute
Maximum spindle speed	RPM
Maximum jog speed	inches / minute
Maximum x-axis distance traverse (stroke)	inches
Maximum y-axis distance traverse (stroke)	inches
Maximum z-axis distance traverse (stroke)	inches
Minimum x-axis distance traverse (stroke)	inches
Minimum y-axis distance traverse (stroke)	inches
Minimum z-axis distance traverse (stroke)	inches
Limit switch's present	yes / no
Limit switch forward x-axis	yes / no
Limit switch reverse x-axis	yes / no
Limit switch forward y-axis	yes / no
Limit switch reverse y-axis	yes / no
Limit switch forward z-axis	yes / no
Limit switch reverse z-axis	yes / no
Home switch's present	yes / no
Home switch present x-axis	yes / no
Home switch present y-axis	yes / no
Home switch present z-axis	yes / no
Motor polarity x-axis (reverse polarity)	normal / reverse
Motor polarity y-axis (reverse polarity)	normal / reverse
Motor polarity z-axis (reverse polarity)	normal / reverse
Motor polarity spindle (reverse polarity)	normal / reverse
Error limit x-axis	counts
Error limit y-axis	counts
Error limit z-axis	counts
Error limit spindle	counts
Off on error	yes / no
Off on error x-axis	yes / no
Off on error y-axis	yes / no
Off on error z-axis	yes / no
Off on error spindle	yes / no
Backlash compensation per direction x-axis	counts / rev
Backlash compensation per direction y-axis	counts / rev
Backlash compensation per direction z-axis	counts / rev
Proportional gain x-axis	real number
Integral gain x-axis	real number
Derivative gain x-axis	real number
Proportional gain y-axis	real number
Integral gain y-axis	real number
Derivative gain y-axis	real number

Proportional gain z-axis	real number
Integral gain z-axis	real number
Derivative gain z-axis	real number
Proportional gain spindle	real number
Integral gain spindle	real number
Derivative gain spindle	real number
Integrator gain x-axis	real number
Integrator gain y-axis	real number
Integrator gain z-axis	real number
Integrator gain spindle	real number
Controller address	real number
Activate input bits	yes / no
Activate output bits	yes / no

Controller Graphical User Interface

The Graphical User Interface (GUI) is the element in the open architecture control system which allows interaction between the operator and the vertical milling machine. This software element consists of programs that reside in the Windows environment, and make use of Windows features. This type of structure permits incorporation of third party interface programs, such as diagnostics and tuning. The GUI is programmed in a high level language for example visual basic, and visual C++.

The GUI is comprised of graphical images that simulate specific tasks that are normally carried out by hardware in a CNC type controller. These windows may be designed to emulate commercially available controllers. A graphical user interface should include the following minimum set of windows.

Main Window:

The main window makes use of standard Windows features such as, icons, toolbar, point and click, drag and drop, pull down screens ect.

Parameter Window:

To expedite the integration process, a parameter window will incorporate machine tool setup parameters from a table menu.

Jog Window:

A jog window will permit manual axis jog, homing, and incremental positioning.

MDI Window:

The machine data input window enables part creation, editing, program search, program load as well as program execution.

Multi Position Window:

A numerical position display window should include: actual, absolute, machine distance to go and relative positioning information.

Execution Window:

The execution window will provide, part programs, cycle start, cycle stop, feed hold, rewind, program display, current line display, program search, program load and execution of these features.

Status Window:

A highly desirable feature, is a window with live controller querying capability.

Tool Selection Window:

A table menu will display user definable characterized tools which are numerically assigned. This window simplifies, tool to part program compatibility process.

Toolpath Display Window:

Real time graphical toolpath display easily allows the user a method to test part programs prior to actually cutting material (Dry Run).

On-line Help Window:

An on-line user manual window makes available to the user information normally contained in hard copy.

Controller Environment

Hardware:

- An IBM compatible PC with 486 DX microprocessor at 66 MHz (the Intel Pentium microprocessor is recommended)
- A hard disk with a space of 200Mb
- A minimum of 8 Mb of RAM memory (16 Mb of RAM is highly recommended)
- A 3.5 inch high density floppy disk
- Industrial quality SVGA monitor
- A Galil DMC 1000 series motion controller supporting 3 axes of motion, spindle and 72 input bits and 40 output bits
- A touch screen is recommended

Software:

- Microsoft Windows 3.1, Windows 95 or higher operating environment
- Microsoft DOS 5.0 or higher
- Galil WSDK (Windows servo design kit)
- CNC for Windows Zero M
- Visual Basic ver. 3.0
- Galil VBX (Visual Basic developers kit)

Benefits

Benefits attainable through the employment of an open architecture CNC system include : cost, flexibility, maintainability, and ease of use.

- Cost savings are achieved through the utilization of components which are mass manufactured and sold at competitive prices.
- Flexibility in this type of controller allow it to adapt to changing demands and requirements in the manufacture of products.
- Maintainability can easily be accomplished by having integrated software tools such as tuning and diagnostics. Up-grading this type of CNC controller system through software revisions will keep the machine tool in a competitive state with minimal effort.
- Ease of use is perhaps this CNC system most valuable attribute. This CNC system minimizes training time by making use of Windows environment.

Example

To illustrate the feasibility of an open architecture CNC system, consider the following case. A typical 3-axis milling machine with spindle is refitted with an Austin 486-33 PC incorporated with a Galil DMC-1040.

This system includes standard hardware components: motors with position sensors, motor amplifiers, limit switches, E-stop and other such components. This controller consists of a Microsoft Windows 3.1 environment, Galil servo design kit, G-code translator and part programs.

The graphical user interface is CNC-PC for Windows, which emulates the Fanuc 0 CNC machine tool controller.

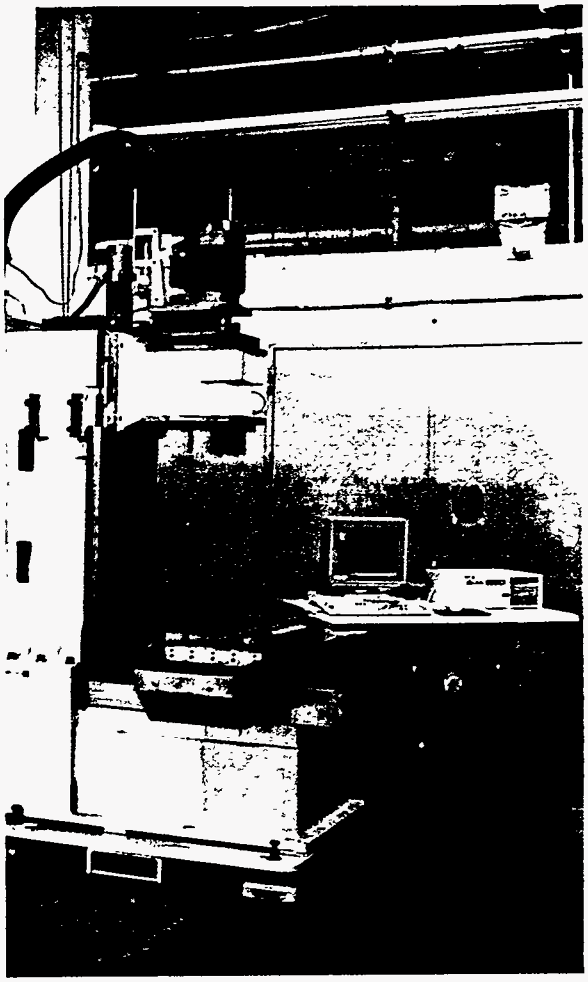
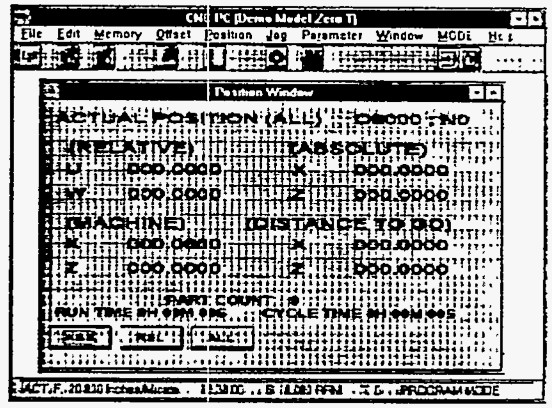
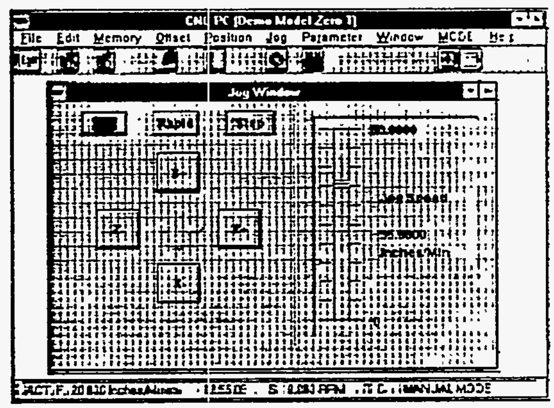


Figure 3
3-Axis Milling Machine



Position display shows the information on each axis

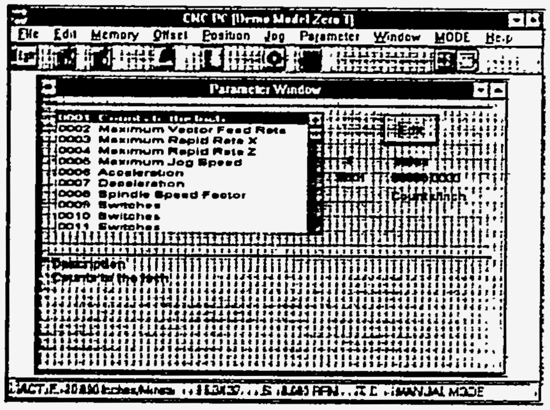


The jog panel provides joystick capability

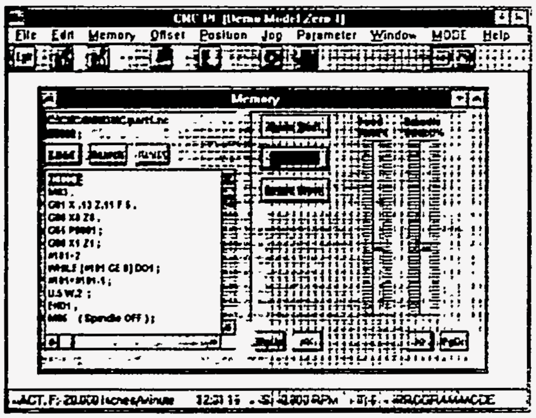
Figure 4
CNC-PC for Windows Illustration

The PC based platform has enhanced this machine tool to a system that is state of the art. This system employs a well-supported array of hardware and software products that give it flexibility, and computing power at a low cost.

The PC based open architecture CNC system can easily be customized and expanded. More memory may be added, latest versions of software can be installed and if desired, the machine tool may be put on a network with other machine tools or communication systems. CNC-PC for Windows, Fanuc 0 emulator is comprised of screens some of which are displayed below. The screens are shown in the 2-axis format, in actual practice the screens display 3-axis of information.



The parameters window, permits parameter set up



The main panel provides program execution

Figure 5
CNC-PC for Windows Illustrations

Summary

In this paper, an alternative solution to the traditional CNC machine tool controller has been introduced. Software and hardware modules have been described and their incorporation in a CNC control system has been outlined.

This type of CNC machine tool controller demonstrates that technology is accessible and can be readily implemented into an open architecture machine tool controller. Benefit to the user is greater controller flexibility, while being economically achievable.

PC based, motion as well as non-motion features will provide flexibility through a Windows environment. Up-grading this type of controller system through software revisions will keep the machine tool in a competitive state with minimal effort. Software and hardware modules are mass produced permitting competitive procurement and incorporation.

Open architecture CNC systems provide diagnostics thus enhancing maintainability, and machine tool up-time.

A major concern of traditional CNC systems has been operator training time. Training time can be greatly minimized by making use of Windows environment features.

References

1. Lynch, M. "Computer Numerical Control for Machining", Mc Graw Hill, 1992
2. Galil Motion Control Inc. "DMC-1000 Technical Reference Guide, version 1.4",1994
3. Electronic Industries Association, "EIA Standard RS-274-D", 1988
4. Electronic Industries Association, "EIA Standard EIA-267-C", 1990
5. CNC Software and Systems, Inc. "CNC for Windows System Zero", 1995

MS-DOS, Windows, and Visual Basic are trademarks of Microsoft Corporation

Fanuc 0 is the trade mark of Fanuc LTD.

CNC-PC for Windows is the trademark of Software and Systems Inc.