MICROCOMPUTER CONTROL OF A MACHINE TOOL

by

Professor D. French, R. Rier, and B. Hughes
Dept. of Mechanical Engineering
The University of Waterloo
Waterloo, Ontario, Canada N2L 3G1

The requirement of a low cost Controller for Machine Tools, which could perform similar functions to more expensive systems, is necessary if small companies are to be able to use sophisticated techniques for manufacture.

At the University of Waterloo, in the Department of Mechanical Engineering, a number of systems have been designed. The systems have varied both in the hardware, and software used.

When designing these systems the criteria used, was that the system had to be made from commercial products, and easy for the operator to use. In addition the input data had to conform to the accepted Industrial standards. Consequently the systems were designed to use a Microcomputer, using a high level language for input data and manipulation, and machine code for interpolation and pulse train generation. The velocity (feedrate) of the Machine axes was dependant on the efficiency of the program, and how data was passed between the high level language and the machine code. Systems have been designed using Basic and machine code, Pascal and machine code. In order to increase the efficiency of the machine operations, it was necessary to use a language which allowed machine code to be embedded as required, in addition graphics was also required to allow the cutter path to be displayed. For these reasons it was decided to use Forth as the language on the I.B.M. personal computer. The parallel output port of the I.B.M. P.C. was used to output the Axis pulse trains to the amplifier circuits of the Stepping Motors used to drive the Machine tool.

The system uses the format as specified in the E.I.A. RS 274-D Standards for data input. The format classification being:-"n3,g2x2.4, y2.4,a2.4,f3.3,s2,t4,m2;" thus the system is capable of providing Contouring and Point to Point modes, the contouring being Linear and Circular arc.

The operation of the system is as follows:-
When the system is started the screen is divided into four quadrents, the Top right-hand quadrent being used for the selection of the different operating modes. The remainder of the screen being used for graphic display of the cutter centre path. There are seven operating modes which are displayed in a menu, enabling the desired mode of operation to be selected by the operator.

These modes are:-

1) Tool offset data. This allows the operator to key in data with respect to tool length and diameter, once the data has been entered it can be displayed in order to ensure the data is correct.

2) Jogging. This mode allows the operator to move the machine

axes in order to align the part, each axis can be moved separately, the amount of movement being input from the keyboard specified by the number of 1/1000 of inches to be moved. The system has a flating zero, so that once the machine has been positioned the zero datum can be set to that point.

3) Manual operating mode. This enables the operator to operate the machine by inputting data in the normal format to perform a machining operation.

4) Manual operating mode with graphics. This mode operates in the same manner as the manual mode, but in addition displays the cutter centre path on the screen.

5) Disk operating mode. Programs to operate the machine tool may be stored on disk. These programs can be called by this mode of operation, thus enabling the machine tool to execute operations previously stored, or input to a disk on another microcomputer.

6) Graphic mode from disk. Prior to carrying out the machining operation it is possible in this mode to show the cutter centre path generated from the data stored on the disk. This allows the operator to determine if the cutter path is correct before starting the machining operation.

7) End. This mode clears all data, closes files, and displays the program name of the machining operation which has just been completed.

The graphic display requests the maximum and minimum limits of the part in order that the three views can be displayed in the three screen quadrents correctly. Once this data has been input to the system it can be used continuously or until the operator wishes to change these values.

In the disk operating mode it is possible to carry out the machining operation "one machining operation at a time", in which case the machine stops after each operation and will wait for the command to continue, or in the Auto-mode, will operate continuously until the machining operation has been completed. During the machining operation it is possible to change the feedrate by using a numeric input. The number keys are programmed to input a condition which allows the programmed feedrate to be reduced by the percentage specified. For example by keying in a "4" will change the feedrate to 40% of the programmed feedrate. This method of changing the feedrate enables the machine to be stopped by depressing the "0" key.

A further innovation allows a Hard copy to be made of the screen display, thus in the graphics mode, after the display has been checked a hard copy can be made for the operator.

During machining operations the machine table movements are displayed on the screen. When a table movement occurs during a machining operation the axes which are carrying out the movement are enclosed in a rectangle which emphasises the movement for the operator.

Provision has been made for tool changing. A machine with a single spindle and no tool changer, will, when encountering a toolchange code, cause the machine tool to stop to allow the operator to change tools.

It was found that with the flexibility of the Forth language additional functions are very easy to implement. To illustrate this a quadratic surface was implemented. The quadratic surface was of the form ,z=x/1000 * (x-z)^2/1000+ I(x-z)/1000 +k + Y/1000*(y-a)^2/1000 + j(y-a)/1000 + 1. The surface operation is called by a G-code, the input values x,y,z,a,i,j,k,l define the surface. The R plane values (0-9) is used to specify the number of different surfaces to be cut. The i and j values are the distance between calculations of the z values of the surface. Linear interpolation is carried out along this incremental distance. Thus it is possible to implement special functions within the system by writing a number of Forth screens and incorporating them into the main program.

Problems - Advantages in Using Forth

FORTH has proven itself to be a suitable language for control of this type of process. The language is fast, has an excellent interface to machine language routines and has good graphics support. System development time is relatively short. With regard to PC-Forth the editor is comprehensive; functions compile and load quickly and a good selection of graphics primitives are included.

When the controller updates the screen and gets and analyzes the next control command the machine stops moving momentarily and a dwell mark is left on the workpiece. This problem occurs regardless of the language in use. It could be improved by: eliminating the screen updates, performing some preprocessing on the input program before execution begins, moving the program from disk to a queue type of data structure in RAM before processing begins or by using a coprocessor type of system.

FORTH's use of screens and virtual memory is beneficial for an application where a sequence of instructions must be stored and the executed. Any segment of the instructions can be accessed directly. There is no need to sequentially read through a file as in some other languages. The use of a screen file for program storage allows a small database of programs to be used. Any portion of these programs may be copied to a new program by using the editor. A further decrease in NC programming time could be realized by using the modules stored in the screen file as functions within the NC program being developed. The primary disadvantage to using the screen files for storage is that the operator would have to have a knowledge of the language FORTH, and the FORTH editor to create and store programs. This problem has been reduced by creating a set of utilities, written in FORTH, that will copy a conventionally formatted DOS file and store it in the specified screens in the current screen file.

FORTH makes intelligent use of pointers, representing all variables as a pointer. This makes the implementation of stacks, queues and other data structures relatively straightforward, which also allows variables to be shared between the FORTH and assembly language functions. In addition it also eliminates the double storage of variables that is required if machine language routines are to be used from a language such as BASIC, where the value of the variable must be poked into a known memory location which will be accessed by the machine language program.

The structure of FORTH makes it simple to alter the program after it

has been completed by simply adding or subtracting new modules. A module must always be defined before it is called. This is not as convenient as with a language such as C, which uses a two pass compiler that will allow functions in other programs or functions that have not been declared to be called. C also allows recursion.

FORTH's use of pointers and its flexibility can lead to problems with respect to the user interface. Most error and type checking functions must be written by the programmer. These functions are necessary to edit the operator's input and recover gracefully from errors. The development of these functions increases program development time and increases the amount of code and execution time required.