# INTELLIGENT THREE-AXES MOTION CONTROLLER FOR ISA BUS COMPUTER

**Wahidah M., Zaifulnahar N.J., Mohd Uzir K.,**
**Ahmad Hairi A.B. & P.N. Rao**
**Faculty of Electrical Engineering**
**Institut Teknologi Mara**
**40450 Shah Alam, Selangor**
**Malaysia**
**email: wahidah@engr.itm.edu.my**

**Abstract**

This paper describes the design and implementation of an intelligent motion controller for Industry Standard Architecture (ISA) bus computer. The motion controller card has been built and plugged in the ISA bus of personal computer (PC) to enable the PC to perform the switching and controlling processes. The card provides four control modes of operations such as positional control, proportional velocity control, trapezoidal velocity profile and integral velocity profile. The controller has been tested successfully on a simple stepper motor arrangement and an analog-to-digital converter circuitry (DAC). The control methodology is implemented using Visual Basic 6.0 language environment and adaptable to various industrial application settings.

## 1.0 Introduction

The traditional CNC controllers are based on proprietary designs of the various developers. A number of industrial automation devices require a general purpose device that can be easily programmed for any automation requirement. Considering the importance of a general purpose device in industrial automation, a motion controller card which is sufficiently intelligent and can be adapted to various applications has been designed. The card is capable of controlling 3-axes movements with the controlling software executed on the Pentium host computer platform.

Software implementation enables the complex control schemes and supervisory functions to be incorporated to different applications without any alteration in the hardware [1]. Furthermore, the software can be installed in a PC which can serve the need of industrial automation. This paper describes the design of a motion controller card, decoding scheme and its application to a stepper motor and DAC. This paper also focuses on the software design methodology.

## 2.0 Motion Controller

The motion controller card is developed using two major items; 8255 Programmable Peripheral Interface (PPI) and HCTL 1100 digital motion control integrated circuits and designed for ISA bus system. The motion controller offers four modes of operations such as positional control, proportional velocity control, trapezoidal velocity profile and integral velocity profile and provides the following features:

- Selectable card address
- Pulse Width Modulation (PWM) output
- 24 bit general purpose Input/Output lines
- 24 bit Output lines for 3 axes motion
- ± 10V maximum analog voltage range
- 3 analog channels

Figure 1 shows the block diagram of a motion controller card. The card consists of a 74LS245 Octal Bus Transceiver, two 74LS244 Octal Buffers, a 74LS688 8-bit comparator, a 74LS138 three-to-eight decoder, two 8255 PPIs and three HCTL-1100s.

### 2.1 Programmable Peripheral Interface

The Intel 8255 PPIs are used to enable three HCTL 1100, transmit data to HCTL 1100 and provide 24 bit general purpose input or output ports. The 8255 PPIs are preferred, as it is compatible with the IBM PC. Furthermore, 8255 PPI provides data handshaking if desired.

### 2.2 Digital Motion Control Integrated Circuit

The HCTL-1100 is a high performance, general purpose motion control IC, fabricated in HP CMOS technology . It frees the host processor for other tasks by performing all the time-intensive functions of digital motion [2]. The HCTL-1100 ICs allows three motors to be connected to the motion controller card to

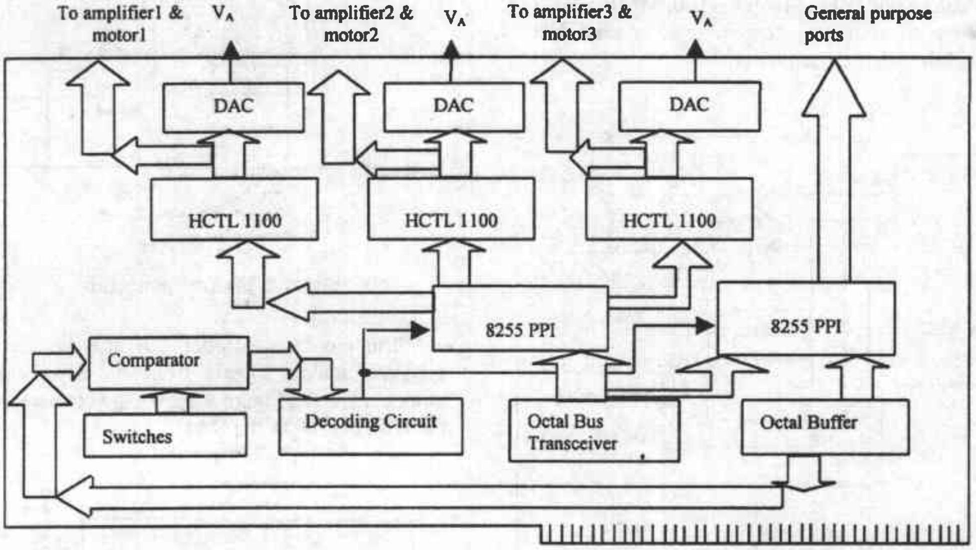


Figure 1. Block diagram of an intelligent motion controller card

perform three-axes motion that is X, Y and Z movement. The output of these ICs can also be connected to DAC to generate analog outputs.

### 2.3 Three-to-eight Decoder

The 74LS138 is used to decode the PC address lines and allows the motion controller card to function at a unique address.

### 2.4 Eight Bit Comparator

The 74LS688 is used to compare the conditions of 8-bit switches with the software switches. If there are identical, the motion controller will start operating. With this arrangement, a selectable base address can be implemented.

### 2.4 Decoding Network

The motion controller card can be inserted into any available 8 or 16-bit slot on PC's motherboard. Figure 2 shows the address decoding network for the motion controller card. The microcomputer address lines A2, A3 and A4 are connected to the select inputs of the 74LS138. The enable line of the 74LS138 is driven from the output of the 74LS688, which will be activated when PC address lines A5, A7, A8 and A9 match with the condition of the 8-bit switches. With this arrangement, the motion controller card will operate in the address range of 280H-2EFH or 300H-31FH.

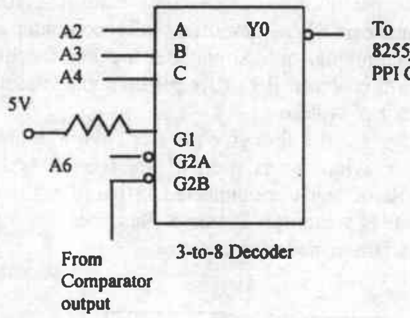


Figure 2 Motion Controller decoding network

### 2.5 Interfacing the HCTL-1100 to the 8255 PPI

The 8255 PPI has 3 eight-bit ports, A, B and C. Each port can be programmed to operate as input, output ,bidirectional or control ports. To control the HCTL-1100, Port B and C are used to generate control signals and enable the HCTL-1100. Port A is used to send data to the HCTL-

1100. The physical connection between the 8255 PPI and three HCTL-1100s is shown in Figure 3. The output lines of the HCTL-1100, MC0 to MC7 can be used to drive a stepper motor or send data to digital-to-analog converter.

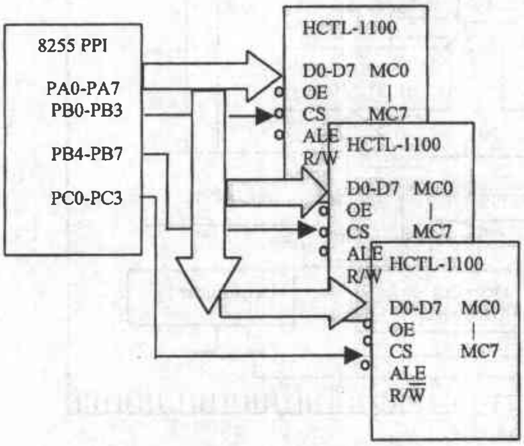


Figure 3 Connections between HCTL-1100 and 8255 PPI

## 3.0 Motion Controller Applications

The Motion Controller can be used for closed loop position and velocity control of various types of motors such as dc brush motors, dc brushless motors, stepper motor and others. Figure 4 shows the possible arrangement for implementing the stepper motor closed loop control system. A linear or PWM amplifier, microcomputer and an incremental optical encoder are interfaced to the motion controller to implement the closed loop control system.

A 1.8 degree per step, hybrid motor with four windings is used in the control system. A pair of leads is connected in parallel giving two sets of windings. Figure 5 illustrates the windings and the connections.

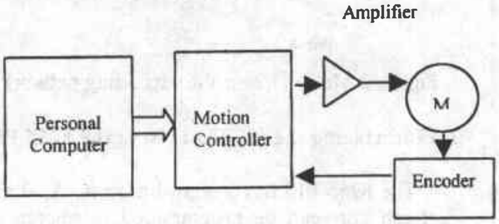


Figure 4 Stepper motor interface

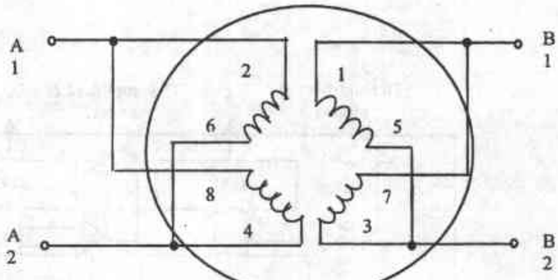


Figure 5 Motor Connections

The motion controller can also be used to generate analog signals from its three analog output channels. Figure 6 shows the connection of the HCTL-1100 to the DAC.

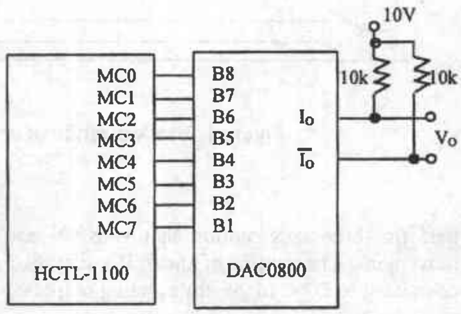


Figure 6 The connection between HCTL-1100 and DAC0800

## 4.0 Software Design Methodology

The software starts with port initialization, which assigns Port A, B, and C of the 8255 as output ports. Then $\overline{ALE}$ signal of the HCTL-1100 is pulsed low for 2 seconds followed by the activation of the $\overline{CS}$ and write signals of the HCTL-110 and the deactivation of the $\overline{ALE}$ signal. Enough time should be given to the HCTL-1100 to send data to the output devices. Here the delay is set to 2 seconds. After this, data is send to the motor or digital-to-analog converter. As soon as the output operation is completed, the $\overline{CS}$ and write signals are deactivated. The same sequence is applied to read operation except that the R/$\overline{W}$ signal is pulsed high instead of low for write operation. This software methodology is applied to all HCTL-1100 operating modes.

The software sequence for DAC is really straightforward. To output an analog voltage, the digital data is sent directly to the digital-to-analog converter using the above software methodology. However, controlling the stepper motor requires an additional sequence. Using the above software methodology and based on the motor configuration described, the commutator is programmed as follows:

Status Register (R07H) = 11H
Commutator Ring (R18H) = 32
X Register (R1AH) = 8
Y Phase OverLap (R1BH) = 0
Offset (RICH) = 0
Max. Phase Advance (R1FH) = 0
Velocity Timer (R19H) = 0

The Status Register is used to control whether the commutator works on the basis of quadrature counts or full cycles.

## 5.0 Results and Discussion

The functionality of the motion controller has been tested on a stepper motor arrangement and DAC circuitry. Figure 7 shows the commutator results for phase A, phase B, phase C and phase D; and Table 1 shows the digital-to-analog converter output for various digital inputs.

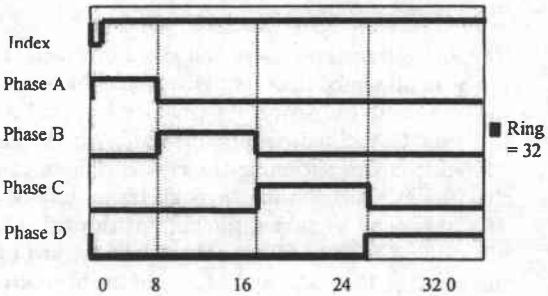


Figure 7 Commutator Output

Table 1 HCTL-1100 and DAC outputs.

| User input data (decimal) | HCTL-1100 Output (Hex) | DAC output (Volt) |
|---|---|---|
| 255 | 1111 1111 | -9.920 |
| 254 | 1111 1110 | -9.840 |
| 129 | 1000 0001 | -0.080 |
| 128 | 1000 0000 | 0.000 |
| 127 | 0111 1111 | +0.080 |
| 1 | 0000 0001 | +9.920 |
| 0 | 0000 0000 | +10.000 |

Even though the motion controller card is only tested on a simple stepper motor circuitry and DAC, the motor controller has produced excellent results and shown good performance. The testing will be enhanced in the future to include four modes of motion such as jogging, independent axis positioning, point-to-point positioning, linear interpolation and linear and 2-D coordinated motion.

## 6.0 Conclusions

The design and implementation of an intelligent three-axes motion controller for ISA bus computer has been described. The motion controller has been tested successfully on a simple stepper motor arrangement and analog-to-digital converter circuitry. The software methodology described in this paper is adaptable for various industrial applications setting.

## References

[1] J.P Agrawal, O. Farook and C.R Sekhar, "Software Switch For Motion Controllers" *Proceedings of IEEE 1994*, pp. 1751-1757 1993.

[2] *General PurposeMotion Control IC,* Hewlett Packard Technical Data , pp. 1-39,1990

[3] Ronald L. Krutz, *Interfacing Techniques InDigital Design With Emphasis On Microprocessors,* Wiley, 1988

[4] R. E. Vears, *Microprocessor Interfacing,* Butterworth-Heinemann, 1992

[5] David Pritty, Duncan Smeed and Peter Barrie, *Practical Interfacing To Popular Microprocessors*, Addison-Wesley, 1985

[6] Harold S. Stone, *Microcomputer Interfacing,* Wiley, 1982

[7] Murray Sargent III and Richard L. Shoemaker, *Interfacing Microcomputer to the Real World,* Wiley, 1983