A Review Paper on Microprocessor Based Controller Programming

Anurag Negi, Akshay Rohilla, Harish Joshi
Dronacharya college of engineering
Maharishi Dayanand University, Rohtak

Abstract- Specific hardware as well as software programming is required for designing of microprocessor. Type of software may be operating system or application software. Programming requires knowledge of system configuration and controller specific programming. Microprocessor can control directly at digital level called Direct Digital Control (DDC).

Index Terms- Controller Software, DDC, Controller Configuration, Controller Programming, Custom Level Programming, Digital Form

I. INTRODUCTION

Computer based controllers were used in 1960s. They were having one mainframe computer and because of this they were very expensive. With the development in the field of microprocessors cost of operating a system by micro controller became very small. In actual a microprocessor is a computer on a chip, and high-density memories reduced costs and package size dramatically and increased application flexibility. controllers take input from sensors perform the programmed task on input data and provide output to actuators. Since the programs are written in digital form, so controllers perform direct digital control (DDC). A direct digital control is the controller which updates the process as function of measured output variable and input provided for interaction with the outside world analog is to be converted into digital format. For this purpose analog to digital and digital to analog controllers are used as shown in fig 1.
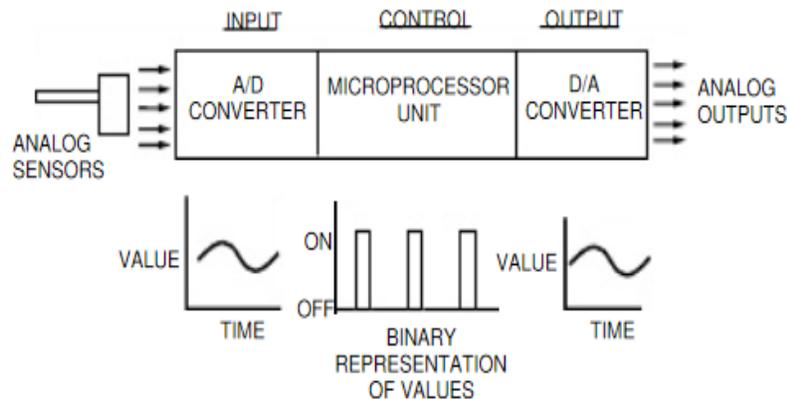

Figure1: A microprocessor based control system use A/D- D/A converter

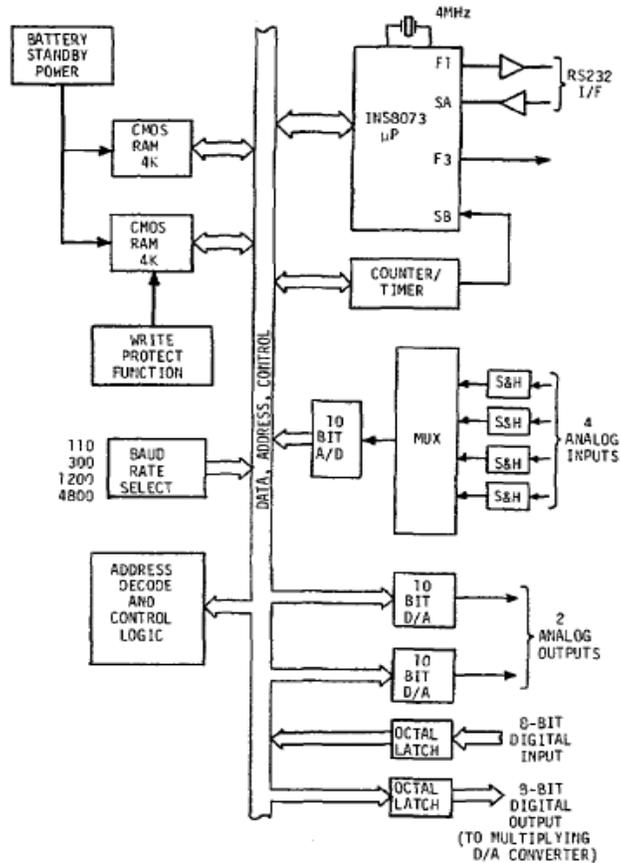Block diagram of convertor is shown in figure2 [1].

Figure2: Microprocessor based digital control system
Figure 2 shows the analog input and output through A/D and D/A converter.

II. CONTROLLER CONFIGURATION

The basic elements of a microprocessor-based (or micro-processor) controller (F i g . 3) include:

—The microprocessor
—A program memory
—A working memory
—A clock or timing de vices
—A means of getting data in and out of the system

In addition ,communication port is also a requirement for program tuning or interfacing with a central computer or building management system. Timing for microprocessor operation is provided by a battery-backed clock. The clock operates in the microsecond range controlling execution of program instructions. Basic instruction set for controller operation as well as for the application programs are held by programme memory. Memory size and type vary depending on the application and whether the controller is considered a dedicated purpose or general purpose device. Dedicated purpose system has generally have standard programmes and read only memory (ROM) or programmable read only memory (PROM.). general purpose systems have variety of individual programmes- alterable memories such as electrically erasable, programmable, read only memory (EEPROM) or flash memory. Memories used to hold the program for a controller must be nonvolatile, that is, they retain the program data during power outages. A/D converters for DDC applications normally range from8 to 12 bits depending on the application. An 8 - bit A/D converter provides a resolution of one count in 256. A 12-bit A/D converter provides a resolution of one count in 4096. If the A/D converter is set up to provide a binary coded decimal (BCD)output, a 12-bit converter can provide values from 0 to 999, 0to 99.9, or 0 to 9.99 depending on the decimal placement [3].
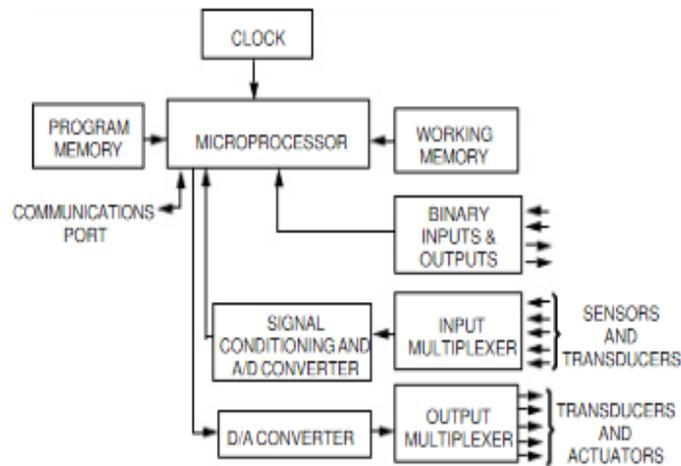.

Figure3: Microprocessor Controller Configuration for automatic Control Applications

III. CONTROLLER SOFTWARE

Although use of microprocessor controller for any application depends upon the hardware but software determines the functionality. Controller software falls basically into two categories:
1. Operating software which controls the basic operation of the controller
2. Application software which addresses the unique control requirements of specific applications

3.1 Operating software

volatile memory is generally used to store operating system such as ROM, PROM. Operating software includes the operating system (OS) and routines for task scheduling, I/O scanning; priority interrupt processing, A/D and D/A conversion, and access and display of control program variables such as set points, temperature values, parameters, and data file information. Tasks are scheduled sequentially and interlaced with I/O scanning and other routine tasks in such a way as to make

operation appear almost simultaneous. If any higher priority task appears to operating software then current going task is halted and data held in registers and accumulators are transferred to temporary registers. Interrupt register processes these interrupt requests. When interrupt task is over then normal routine is started and data is transferred back from temporary registers to mainstream.

3.2 Application Software

Application software includes direct digital control, energy management, lighting control, and event initiated programs plus other alarm

and monitoring software typically classified as building management functions. The system allows application programs to be used individually or in combination.

3.2.1 DIRECT DIGITAL CONTROL SOFTWARE

DDC software is used for specific control actions. These are set of standard DDC operators. Key elements in most direct

digital control programs are the PID and the enhanced EPID and ANPID algorithms. While the P, PI, PID, EPID, and ANPID operators provide the basic control action, there are many other operators that enhance and extend the control program .Table 1 shows some other typical operators. These operators are computer statements that denote specific DDC operations to be performed in the controller .

Table 1. Typical DDC Operators.

Operator	Description
Sequence	Allows several controller outputs to be sequenced, each one operating over a full output range.
Reversing	Allows the control output to be reversed to accommodate the action of a control valve or damper actuator.
Ratio	Translates an analog output on one scale to a proportional analog output on a different scale.
Analog controlled digital output	Allows a digital output to change when an analog input reaches an assigned value. Also has an assignable dead band feature.
Digital controlled analog output	Functionally similar to a signal switching relay. One state of the digital input selects one analog input as its analog output; the other state selects a second analog input as the analog output.
Analog controlled analog output	Similar to the digital controlled analog output except that the value and direction of the analog input selects one of the two analog signals for output.
Maximum input	Selects the highest of several analog input values as the analog output.
Minimum input	Selects the lowest of several analog input values as the analog output.
Delay	Provides a programmable time delay between execution of sections of program code.
Ramp	Converts fast analog step value changes into a gradual change.

IV. CONTROLLER PROGRAMMING

Controller programming makes the controller usable for a specific control action. Programming of microcomputer -based controllers can be subdivided into four discrete categories:

1. Configuration programming
2. System initialization programming
3. Data file programming
4. Custom control programming

Configuration programming matches the hardware and software matches the control action required. Both hardware and software should be matched for application requirement

System initialization programming consists of entering of appropriate startup values from keyboard . Star tup data parameters consist of se t point, throttling range, gain, reset time, ti me of day, occupancy time, and night setbac k temperature . These data are equivalent to the settings on a mechanical control system, but number of item is higher due to added functionality in digital control system.

Requirement of data file programming depends upon whether the system variables are fixed or variable. For example at zonal level program ming where input sensors are fixed and progr ammer knows which relay will get output the n the use of data file programming is irrele vant. But at the system level programming where controller controls wide variety of senso rs and gives output to various relays, use of dat a file programming is must. For the controller to properly process input data, for example, it must know if the point type is analog or digita l. If the point is analog, the controller must k now the sensor
type, the range, whether or not the input value is linear, whether or not alarm limits are assig ned, what the high and low alarm limit values are if limits are assigned, and if there is a loc kout point. See Table
2.If the point is digital, the controller must kno w its normal state (open o r closed)[8], whether the given
state is an alarm state or merely a status condi tion, and whether or not the condition triggers

an event-initiated program.

Table 2. Typical Data File for Analog Input

Point Address	User Address
Point Type	Regular or Calculation
Sensor	Platinum(0 to 100F)
Physical Terminal Assigned	16
Use Code	Cold Deck Dry Bulb
Engineering Unit	F
Decimal Places For display	XXX.X
High Limit	70.0
Low Limit	40.0
Alarm Lockout Point	Point Address
Point Descriptor	Cold Deck Temperature
Alarm Priority	Critical

V.   CONCLUSION

Microprocessor based controllers although depends upon the hardware of controller but the main behavior is defined in software programming. Application software is used if a specific controlling action is needed. Before programming the controller values initial parameters is considered. Complexity of programming also depends upon the number of controllers to be controlled, input is analog or digital. If many inputs are coming to controller then a data file has to be maintained so that just by looking into that file constraints of programming can be identified.
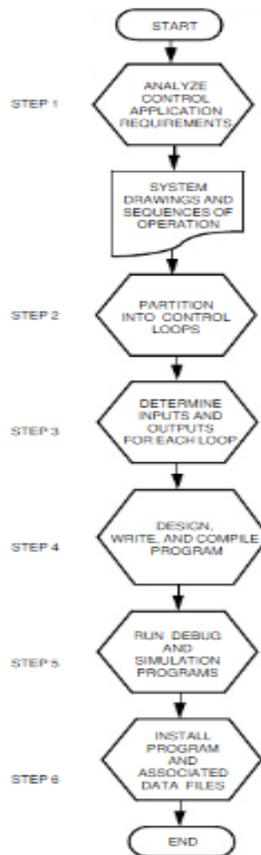

Figure4: Steps for custom level programming

REFERENCES

[1]   S. D. Kraft and Edward T. Wall,‖ Experimental Microprocessor-Based Adaptive Control System‖ IEEE Control
        Systems Magazine

[2]   *w*ww.itswtech.org/Lec/dr.basim/3/lec1.pdf

[3]   http://www.ddc-online.org/Digital-Control-Systems/Introduction-to-Direct-Digital-Control-Systems.html [4] Robert Yung, Stefan Rusu, Ken Shoemaker,‖ Future Trend of Microprocessor Design‖ ESSCIRC 2002 [5]   Katz, P.: 1981, ‒Digital Control System‖, Springer-Verlag, Berlin

[6]   Alfred C. Weaver, ‒A real-time, multi-task programming language for microprocessor-based industrial process
        control‖, ACM '78 Proceedings of the 1978
        annual conference - Volume 2 Pages 522 – 525

[7]   Chang-Jiu Chen, Wei-Min Cheng, Hung-Yue Tsai and Jen- Cheieh Wu,‖ A Quasi-Delay-Insensitive Microprocessor
        Core Implementation for Microcontrollers‖, Journal Of Information Science And Engineering 25, 543-557 (2009) [8]   J. H. Lee, W. C. Lee, and K. R. Cho, ‒A novel asynchronous pipeline architecture for CISC type embedded
        controller ‒ A8051,‖ in Proceedings of the 45th
        Midwest Symposium on Circuits and Systems,
        Vol. 2, 2002, pp.
        675-678.