# Microprocessor-Based/ DDC Fundamentals

## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>3</td>
</tr>
<tr>
<td>Definitions</td>
<td>3</td>
</tr>
<tr>
<td>Background</td>
<td>4</td>
</tr>
<tr>
<td>Computer Based Control</td>
<td>4</td>
</tr>
<tr>
<td>Direct Digital Control</td>
<td>4</td>
</tr>
<tr>
<td>Advantages</td>
<td>4</td>
</tr>
<tr>
<td>Lower Cost per Function</td>
<td>4</td>
</tr>
<tr>
<td>Application Flexibility</td>
<td>5</td>
</tr>
<tr>
<td>Coordinated Multifunction Capability</td>
<td>5</td>
</tr>
<tr>
<td>Precise and Accurate Control</td>
<td>5</td>
</tr>
<tr>
<td>Reliability</td>
<td>5</td>
</tr>
<tr>
<td>Controller Configuration</td>
<td>5</td>
</tr>
<tr>
<td>Types of Controllers</td>
<td>6</td>
</tr>
<tr>
<td>Zone-Level Controller</td>
<td>6</td>
</tr>
<tr>
<td>System-Level Controller</td>
<td>6</td>
</tr>
<tr>
<td>Controller Software</td>
<td>7</td>
</tr>
<tr>
<td>Operating Software</td>
<td>7</td>
</tr>
<tr>
<td>Application Software</td>
<td>7</td>
</tr>
<tr>
<td>Direct Digital Control Software</td>
<td>7</td>
</tr>
<tr>
<td>Energy Management Software</td>
<td>8</td>
</tr>
<tr>
<td>Optimum Start</td>
<td>8</td>
</tr>
<tr>
<td>Optimum Stop</td>
<td>8</td>
</tr>
<tr>
<td>Night Cycle</td>
<td>8</td>
</tr>
<tr>
<td>Night Purge</td>
<td>9</td>
</tr>
<tr>
<td>Enthalpy</td>
<td>9</td>
</tr>
<tr>
<td>Load Reset</td>
<td>9</td>
</tr>
<tr>
<td>Zero Energy Band</td>
<td>11</td>
</tr>
<tr>
<td>Distributed Power Demand</td>
<td>12</td>
</tr>
<tr>
<td>Building Management Software</td>
<td>12</td>
</tr>
</tbody>
</table>

® U.S. Registered Trademark
Copyright © 1998 Honeywell Inc. • All Rights Reserved

77-1123-1
## Controller Programming

- General ............................................................................................................. 12
- ProgrammingCategories .................................................................................. 13
- ConfigurationProgramming ............................................................................ 13
- SystemInitializationProgramming ................................................................. 13
- Data File Programming .................................................................................... 13
- Custom Control Programming ...................................................................... 13
  - Analyze Control Application ...................................................................... 14
  - Partition Into Control Loops ....................................................................... 14
  - Determine Inputs and Outputs ...................................................................... 14
  - Design, Write, and Compile Program ......................................................... 15
  - Debug, Install, Enter Data Files, and Test ............................................... 15

## Typical Applications

- Zone-Level Controller ..................................................................................... 15
  - Example 1. VAV Cooling Only ................................................................. 15
  - Example 2. VAV Cooling with Sequenced Electric Reheat ..................... 15
- System-Level Controller ............................................................................... 16
INTRODUCTION

This section discusses the types of microprocessor-based controllers used in commercial buildings. These controllers measure signals from sensors, perform control routines in software programs, and take corrective action in the form of output signals to actuators. Since the programs are in digital form, the controllers perform what is known as direct digital control (DDC). Microprocessor-based controllers can be used as stand-alone controllers or they can be used as controllers incorporated into a building management system utilizing a personal computer (PC) as a host to provide additional functions. A stand-alone controller can take several forms. The simplest generally controls only one control loop while larger versions can control from eight to 40 control loops. As the systems get larger, they generally incorporate more programming features and functions. This section covers the controller as a stand-alone unit. Refer to the Building Management System Fundamentals section for additional information on use of the controller in networked and building management systems.

DEFINITIONS

**Analog-to-digital (A/D) converter:** The part of a microprocessor-based controller that changes an analog input signal to a digital value for use by the microprocessor in executing software programs. Analog input values typically come from temperature, pressure, humidity, or other types of sensors or transducers.

**Application software:** Programs that provide functions such as direct digital control, energy management, lighting control, event initiated operations, and other alarm and monitoring routines.

**Configurable controller:** A controller with a set of selectable programs with adjustable parameters but without the ability to modify the programs.

**Digital-to-analog (D/A) converter:** The part of a microprocessor-based controller that changes digital values from a software program to analog output signals for use in the control system. The analog signals are typically used to position actuators or actuate transducers and relays.

**Direct digital control:** A control loop in which a digital controller periodically updates a process as a function of a set of measured control variables and a given set of control algorithms.

**Microprocessor-based controller:** A device consisting of a microprocessor unit, digital input and output connections, A/D and D/A converters, a power supply, and software to perform direct digital control and energy management routines in a HVAC system.

**Operating software:** The main operating system and programs that schedule and control the execution of all other programs in a microprocessor-based controller. This includes routines for input/output (I/O) scanning, A/D and D/A conversion, scheduling of application programs, and access and display of control program variables.

**System-level controller:** A microprocessor-based controller that controls centrally located HVAC equipment such as variable air volume (VAV) supply units, built-up air handlers, and central chiller and boiler plants. These controllers typically have a library of control programs, may control more than one mechanical system from a single controller, and may contain an integral operating terminal.

**Zone-level controller:** A microprocessor-based controller that controls distributed or unitary HVAC equipment such as VAV terminal units, fan coil units, and heat pumps. These controllers typically have relatively few connected I/O devices, standard control sequences, and are dedicated to specific applications.
BACKGROUND

COMPUTER BASED CONTROL

Computer based control systems have been available as an alternative to conventional pneumatic and electronic systems since the mid 1960s. Early installations required a central mainframe or minicomputer as the digital processing unit. They were expensive, and application was limited to larger buildings. Reliability was also an issue since loss of the central computer meant loss of the entire control system. Advances in microtechnology, particularly in large scale integration (LSI), provided answers to both the cost and reliability issues. Introduction of microprocessors, i.e., a computer on a chip, and high-density memories reduced costs and package size dramatically and increased application flexibility (Fig. 1). Microprocessor programs include all the arithmetic, logic, and control elements of larger computers, thus providing computing power at a cost/performance ratio suitable for application to individual air handlers, heat pumps, VAV terminal units, or the entire equipment room. Microprocessor-based controllers allow digital control to be distributed at the zone level, equipment room level, or they can control an entire building.

A more detailed definition is provided in the ASHRAE 1995 HVAC Applications Handbook. “A digital controller can be either single- or multiloop. Interface hardware allows the digital computer to process signals from various input devices, such as the electronic temperature, humidity, and pressure sensors described in the section on Sensors. Based on the digitized equivalents of the voltage or current signals produced by the inputs, the control software calculates the required state of the output devices, such as valve and damper actuators and fan starters. The output devices are then moved to the calculated position via interface hardware, which converts the digital signal from the computer to the analog voltage or current required to position the actuator or energize a relay.”

In each of these definitions the key element for DDC is digital computation. The microprocessor unit (MPU) in the controller provides the computation. Therefore, the term digital in DDC refers to digital processing of data and not that HVAC sensor inputs or control outputs from the controller are necessarily in digital format. Nearly all sensor inputs are analog and most output devices are also analog. In order to accept signals from these I/O devices, A/D and D/A converters are included in the microprocessor-based controller. Figure 2 shows several inputs and outputs. The microprocessor usually performs several control functions.

DIRECT DIGITAL CONTROL

Inherent in microprocessor-based controllers is the ability to perform direct digital control. DDC is used in place of conventional pneumatic or electronic local control loops. There are several industry accepted definitions of DDC. DDC can be defined as “a control loop in which a digital controller periodically updates a process as a function of a set of measured control variables and a given set of control algorithms”.

ADVANTAGES

Digital control offers many advantages. Some of the more important advantages are discussed in the following.

LOWER COST PER FUNCTION

In general, microprocessor and memory costs keep coming down while inherent functionality keeps going up. Compared to earlier systems, physical size of the controller is also reduced while the number of discrete functions is increased. Digital control, using a microcomputer-based controller, allows more sophisticated and energy efficient control sequences to be applied at a lower cost than with non-digital controls; however, simple applications might be less costly with non-digital controls.
APPLICATION FLEXIBILITY

Since microprocessor-based controllers are software based, application flexibility is an inherent feature. A wide variety of HVAC functions can be programmed and, in addition, the controller can perform energy management, indoor air quality (IAQ), and/or building management functions. Changes in control sequences can easily be accommodated through software whether dictated by system performance or by changes in the owner’s use of the facility.

COORDINATED MULTIFUNCTION CAPABILITY

Although basic environmental control and energy management operate as independent programs, it is best to have them incorporated as an integrated program in order to provide more efficient control sequences. For example, sensing the temperatures of several zones to determine the average demand, or the zone with the greatest demand for cooling, will provide improved efficiency and control over merely sampling a representative zone for a chiller reset program. An added feature is that the sensors providing zone comfort control can serve a dual function at no added cost. These benefits require controller-to-controller communications which is discussed in the Building Management System Fundamentals section.

PRECISE AND ACCURATE CONTROL

Proportional control has the inherent problem of offset. The wider the throttling range is set for control stability, the greater the offset. With the microprocessor-based controller, the offset can easily be corrected by the simple addition of integral action. For even more accurate control over a wide range of external conditions, adaptive control algorithms, available in some microprocessor-based controllers, can be employed. With adaptive control, system performance automatically adjusts as conditions vary. The need for manual fine tuning for seasonal changes is eliminated. These items are discussed in the Control Fundamentals section.

RELIABILITY

Digital controllers should be conservatively designed and should incorporate self-checking features so they notify the operator immediately if anything goes wrong. Input and output circuits should be filtered and protected from extraneous signals to assure reliable information to the processor.

CONTROLLER CONFIGURATION

The basic elements of a microprocessor-based (or microprocessor) controller (Fig. 3) include:

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

In addition, a communications port is not only a desirable feature but 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.

Program memory holds the basic instruction set for controller operation as well as for the application programs. Memory size and type vary depending on the application and whether the controller is considered a dedicated purpose or general purpose device. Dedicated purpose configurable controllers normally have standard programs and are furnished with read only memory (ROM) or programmable read only memory (PROM.) General purpose controllers often accommodate a variety of individual custom programs and are supplied with field-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.

![Fig. 3. Microprocessor Controller Configuration for Automatic Control Applications.](image-url)
All input signals, whether analog or digital, undergo conditioning (Fig. 3) to eliminate the adverse effects of contact bounce, induced voltage, or electrical transients. Time delay circuits, electronic filters, and optical coupling are commonly used for this purpose. Analog inputs must also be linearized, scaled, and converted to digital values prior to entering the microprocessor unit. Resistance sensor inputs can also be compensated for leadwire resistance. For additional information about electronic sensors see the Electronic Control Fundamentals section.

Performance and reliability of temperature control applications can be enhanced by using a single 12-bit A/D converter for all controller multiplexed inputs, and simple two-wire high resistance RTDs as inputs.

A/D converters for DDC applications normally range from 8 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, 0 to 99.9, or 0 to 9.99 depending on the decimal placement. This range of outputs adequately covers normal control and display ranges for most HVAC control applications. D/A converters generally range from 6 to 10 bits.

The output multiplexer (Fig. 3) provides the reverse operation from the input multiplexer. It takes a serial string of output values from the D/A converter and routes them to the terminals connected to a transducer or a valve or damper actuator.

The communication port (Fig. 3) allows interconnection of controllers to each other, to a master controller, to a central computer, or to local or portable terminals.

**TYPES OF CONTROLLERS**

Microprocessor-based controllers operate at two levels in commercial buildings: the zone level and the system level. See Figure 4.

**ZONE-LEVEL CONTROLLER**

Zone-level controllers typically control HVAC terminal units that supply heating and cooling energy to occupied spaces and other areas in the building. They can control VAV terminal units, fan coil units, unit ventilators, heat pumps, space pressurization equipment, laboratory fume hoods, and any other zone control or terminal unit device. Design of a zone-level controller is usually dictated by the specific requirements of the application. For example, the controller for a VAV box is frequently packaged with an integral damper actuator and has only the I/O capacity necessary to meet this specific application. On the other hand, a zone-level controller for a packaged heating/cooling unit might have the controller packaged in the thermostat housing (referred to as a smart thermostat or smart controller). Zone level control functions may also be accomplished with bus-connected intelligent sensors and actuators.

**SYSTEM-LEVEL CONTROLLER**

System-level controllers are more flexible than zone-level controllers in application and have more capacity. Typically, system-level controllers are applied to systems in equipment rooms including VAV central supply systems, built-up air handlers, and central chiller and boiler plants. Control sequences vary and usually contain customized programs written to handle the specific application requirements.

The number of inputs and outputs required for a system-level controller is usually not predictable. The application of the controller must allow both the number and mix of inputs and outputs to be variable. Several different packaging approaches have been used:
- Fixed I/O configuration.
- Universal I/O configuration.
- Card cage with plug-in function boards.
- Master/Slave I/O modules.

Universal I/O allows software to define the function of each set of terminals.
Zone- and system-level controllers should be equipped with a communications port. This allows dynamic data, setpoints, and parameters to be passed between a local operator terminal, a central building management system, and/or other controllers. Data passed to other controllers allows sensor values to be shared and interaction between zone-level programs and system-level programs to be coordinated. For example, night setback and morning warmup can be implemented at the zone-level controller based on operational mode information received from the system-level controller.

CONTROL SOFTWARE

Although microprocessor-based controller hardware governs, to some extent, how a controller is applied, 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.

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. For example, the same hardware and operating software can be used for a new or existing building control by using different programs to match the application. An existing building, for example, might require energy management software to be added to the existing control system. A new building, however, might require a combination of direct digital control and energy management software.

DIRECT DIGITAL CONTROL SOFTWARE

DDC software can be defined as a set of standard DDC operators and/or high-level language statements assembled to accomplish a specific control action. Key elements in most direct digital control programs are the PID and the enhanced EPID and ANPID algorithms. For further information, refer to the Control Fundamentals section.

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. Some other typical operators are shown in Table 1. These operators are computer statements that denote specific DDC operations to be performed in the controller. Math, time/calendar, and other calculation routines (such as calculating an enthalpy value from inputs of temperature and humidity) are also required.

The use of preprogrammed operators saves time when writing control sequences and makes understanding of the control sequence the equivalent of reading a pneumatic control diagram. Programming schemes often allow program operators to be selected, positioned, and connected graphically. The alternative to using preprogrammed operators is to write an equivalent control program using the programming language furnished for the controller.
Table 1. Typical DDC Operators.

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence</td>
<td>Allows several controller outputs to be sequenced, each one operating over a full output range.</td>
</tr>
<tr>
<td>Reversing</td>
<td>Allows the control output to be reversed to accommodate the action of a control valve or damper actuator.</td>
</tr>
<tr>
<td>Ratio</td>
<td>Translates an analog output on one scale to a proportional analog output on a different scale.</td>
</tr>
<tr>
<td>Analog controlled digital output</td>
<td>Allows a digital output to change when an analog input reaches an assigned value. Also has an assignable dead band feature.</td>
</tr>
<tr>
<td>Digital controlled analog output</td>
<td>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.</td>
</tr>
<tr>
<td>Analog controlled analog output</td>
<td>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.</td>
</tr>
<tr>
<td>Maximum input</td>
<td>Selects the highest of several analog input values as the analog output.</td>
</tr>
<tr>
<td>Minimum input</td>
<td>Selects the lowest of several analog input values as the analog output.</td>
</tr>
<tr>
<td>Delay</td>
<td>Provides a programmable time delay between execution of sections of program code.</td>
</tr>
<tr>
<td>Ramp</td>
<td>Converts fast analog step value changes into a gradual change.</td>
</tr>
</tbody>
</table>

ENERGY MANAGEMENT SOFTWARE

Microprocessor-based controllers can combine control and energy management functions in the controller to allow sensor and data file sharing and program coordination. Energy management functions can be developed via the above DDC operators, math functions, and time clock values, or they can be separate program subroutines.

A summary of energy management programs possible for integration into microprocessor-based controllers follows:

**Optimum Start**

Based on measurements of indoor and outdoor temperatures and a historical multiplier adjusted by startup data from the previous day, the optimum start program (Fig. 5) calculates a lead time to turn on heating or cooling equipment at the optimum time to bring temperatures to proper level at the time of occupancy. To achieve these results the constant volume AHU optimum start program delays AHU start as long as possible, while the VAV optimum start program often runs the VAV AHU at reduced capacity. Unless required by IAQ, outdoor air dampers and ventilation fans should be inactive during preoccupancy warmup periods. For weekend shutdown periods, the program automatically adjusts to provide longer lead times. This program adapts itself to seasonal and building changes.

**Optimum Stop**

The optimum stop program (Fig. 6) uses stored energy to handle the building load to the end of the occupancy period. Based on the zone temperatures that have the greatest heating and greatest cooling loads, and the measured heating and cooling drift rates, the program adjusts equipment stop time to allow stored energy to maintain the comfort level to the end of the occupancy period. This program adapts itself to changing conditions.
Night Cycle

The night cycle program (Fig. 7) maintains a low temperature limit (heating season) or high temperature limit (cooling season) during unoccupied periods by cycling the air handling unit while the outdoor air damper is closed. Digital control systems often reduce fan capacity of VAV AHU systems to accomplish this and reduce energy usage.

![Fig. 7. Night Cycle.](image)

Night Purge

The night purge program uses cool, night outdoor air to precool the building before the mechanical cooling is turned on. Digital control systems often reduce fan capacity of VAV AHU systems during Night Purge to reduce energy usage. Outdoor temperature, outdoor RH or dewpoint, and space temperature are analyzed. One hundred percent outdoor air is admitted under the following typical conditions:
1. Outdoor air above a summer-winter changeover point, such as 50°F.
2. Outdoor temperature below space temperature by a specified RH or determined differential.
3. Outdoor air dewpoint less than 60°F.
4. Space temperature above some minimum for night purge such as 75°F.

Enthalpy

The enthalpy program (Fig. 8) selects the air source that requires the least total heat (enthalpy) removal to reach the design cooling temperature. The selected air source is either the return air with a selectable minimum amount of outdoor air or a mixture of outdoor and return air as determined by local control from discharge-air or space temperature measurement. Measurements of return-air enthalpy and return-air dry bulb are compared to outdoor air conditions and used as criteria for the air source selection. A variation of this is comparing the outside air enthalpy to a constant (such as 27.5 Btu per pound of dry air) since the controlled return air enthalpy is rather stable.

![Fig. 8. Enthalpy Decision Ladder.](image)

Load Reset

The load reset program (Fig. 9) assures that only the minimum amount of heating or cooling energy is used to satisfy load requirements. Samples of zone demands are taken and the zone with the greatest load is used to reset the temperature of the heating or cooling source.

![Fig. 9. Typical Load Reset Schedules.](image)
Without knowledge of the actual instantaneous demands of the loads that load reset controls, systems must run at the theoretical worst-case setpoints for temperature and pressure.

As with most powerful application programs, load reset requires a great depth of knowledge of the HVAC and control process. The problem is that if load demands indicate that a higher chilled water temperature would be appropriate and the temperature increase is made, all loads are upset because of the chilled water temperature change. The loads must adjust and stabilize before another load reset adjustment is made. Monitoring and tuning is required to assure high performance control and loop stability. Three parameters determine the performance and stability of the system: the magnitude of the incremental corrections, the time interval between correction, and the magnitude of the hysteresis between raising and lowering the temperature.

Sun, weather, and occupancy (building utilization) dictate load reset demands. The sun and weather effects are relatively slow and occur as the sun and seasons change. Occupancy changes are abrupt and occur over brief periods of time.

If a chiller plant with 44F design chilled water temperature is controlled to increase chilled water temperature any time all control valves are less than 80 percent open. Two air handling units with different control sequences are compared.

1. Control valves are on a VAV AHU chilled water coil and are part of a 55F discharge air temperature control loop.
   a. The most demanding AHU valve closes to below 80 percent open.
   b. The load reset program raises the chilled water temperature setpoint.
   c. The chiller unloads to maintain the raised setpoint.
   d. As the chilled water temperature increases, the discharge air temperature increases.
   e. The discharge air temperature controls open the AHU valves to maintain the discharge air temperature setpoint.
   f. The most demanding AHU valve opens to greater than 80 percent but less than 95 percent.
   g. The other AHU valves open increasing the chiller load.
   h. The two temperature loops stabilize in time. The chiller loop is usually set for a fast response and the discharge air loop is set for a slow response.

2. Control valves are on single zone AHU chilled water coil, controlled from space temperature at 76F.
   a. The most demanding AHU valve closes to below 80 percent open.
   b. The load reset program raises the chilled water temperature setpoint.
   c. The chiller unloads to maintain the raised setpoint.
   d. As the chilled water temperature increases, the discharge air temperature increases.
   e. The space temperature control opens the valves to maintain the space temperature setpoint. This response takes several minutes in space temperature control.
   f. The most demanding AHU valve opens to greater than 80 percent but less than 95 percent.
   g. The other AHU valves open increasing the chiller load.
   h. The two temperature loops stabilize in time. The chiller loop is usually set for a fast response and the discharge air loop is set for a slow response.

The events of the first scenario occur within seconds because both loops (leaving water temperature controlling the chiller load and discharge air temperature controlling chilled water flow) are close coupled and fast. Because the two loops oppose each other (a chilled water temperature rise causes discharge air temperature to rise which demands more chilled water flow), a few minutes must be allowed for system stabilization. The chilled water temperature control loop should be fast and keep the chilled water near the new setpoint while the AHU temperature loops slowly adjust to the new temperature.

Hysteresis is a critical load reset parameter. Water temperature is raised if all valves are less than 80 percent open but, is not lowered until one valve is greater than 95 percent open. This 15 percent dead band allows lengthy periods of stability between load reset increases and decreases. Properly tuned load reset programs do not reverse the commands more than once or twice a day.

Scenario 1 initial parameters could be; command chilled water temperature increments of 0.3F, a load reset program execution interval of 4.0 minutes, a decrement threshold of 80 percent, (most demanding valve percent open), an increment threshold of 95 percent (most demanding valve percent open), a start-up chilled water temperature setpoint of 45F, a maximum chilled water temperature setpoint of 51F, and a minimum chilled water temperature setpoint of 44F. The load reset chilled water temperature program may include an AUTO-MANUAL software selector and a manual chilled water temperature setpoint for use in the manual mode.

Unlike scenario 1, the events within scenario 2 occur over several minutes (not seconds) because when the chilled water temperature setpoint is raised, it takes several minutes for the water temperature rise and the resulting air temperature increase to be fully sensed by the space temperature sensor.

Scenario 2 parameters could be the same as scenario 1 with the exception of the execution interval which should be about 15 minutes.

All parameters should be clearly presented and easily commandable. Figure 10 is an example of dynamic data display for scenario 1.
Load reset works best when the number of monitored loads are between 2 and 30. If any monitored load is undersized or stays in full cooling for any reason, reset will not occur. See the Air Handling Systems Control Applications section and the Chiller, Boiler, and Distribution System Control Applications section for other load reset examples.

**Zero Energy Band**

The zero energy band (Fig. 11) program provides a dead band where neither heating nor cooling energy is used. This limits energy use by allowing the space temperature to float between minimum and maximum values. It also controls the mixed-air dampers to use available outdoor air if suitable for cooling. On multizone fan systems with simultaneous heating and cooling load capability, zone load reset controls the hot and cold deck temperature setpoints.
Distributed Power Demand

The distributed power demand program (Fig. 12) is only applicable to microprocessor controllers with intercommunications capability. The demand program is resident in a single controller which monitors the electrical demand and transmits the required load shed or restore messages to other controllers on the communications bus or within the network. Each individual controller has prioritized load shed tables so that when a message to shed a specific number of kilowatts is received it can respond by shedding its share of the load. The basic demand program normally utilizes a sliding window demand algorithm and has provision for sequencing so that the same loads are not always shed first when a peak occurs.

It should be noted that there is interaction between the power demand program, duty cycle program, time schedule programs, and optimum start and stop programs. Therefore, a priority structure program is necessary to prevent control contentions.

BUILDING MANAGEMENT SOFTWARE

Microprocessor-based controllers are used extensively as data gathering panels (DGP) for building management systems. Since a microprocessor-based controller is already in place to provide DDC, IAQ, and EMS functions, many sensors and data files can be shared with building management system (BMS) functions. The distribution of many BMS functions into controllers throughout the premises increases the overall system reliability. The following BMS software is normally included in the controller.

Alarm lockout: Permits designated alarm points to be locked out from reporting process depending on the status of another point, e.g., discharge temperature alarm can be locked out when fan is off and during initial startup periods.

Alarm monitoring: Scans all analog and digital points and tests for alarm status. Sets of high and low limits for analog inputs are stored in the controller.

Communications module: Controls transmissions between controllers and between controllers and a central computer based on an established bus protocol.

Global points: Allows designated points to share their data with other bus connected devices.

Run time: Accumulates equipment on or off time and transmits totals periodically to the central system. On-off cycle counting can also be accumulated as a maintenance indicator. Alarm annunciation occurs if run time or cycle count limits are exceeded.

Time and event programs: Initiates a predetermined series of control actions based on an alarm condition, a point status change, time of day, or elapsed time. Points acted upon can be resident in any controller.

CONTROLER PROGRAMMING

GENERAL

The term programming as it pertains to microprocessor-based controllers relates primarily to setting up the controller for the given application. Zone-level controllers require initialization, selection of control algorithms and parameters, definition of control sequences, and establishing reference data bases. For zone-level controllers, the programming effort can be as simple as selecting the applicable control sequence from a library of programs resident in a configurable controller. For highly customized applications, usually encountered at the system controller level, a problem oriented language or a subset of a high-level language can be used to define control loops and sequences.

The means of entering a program can vary from a keypad and readouts on the controller to an operator terminal in a large centrally based computer configuration. Sophistication of the entry device is directly related to how well defined and fixed the control application is compared to the degree of customization.
or end-user modifications required. If considerable customization or modification is required, data entry could require a centrally based computer or a portable PC.

**PROGRAMMING CATEGORIES**

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.

Some controllers require all four levels of program entry while other controllers, used for standardized applications, require fewer levels.

**CONFIGURATION PROGRAMMING**

Configuration programming consists of selecting which preprogrammed control sequence to use. It requires the selection of hardware and/or software packages to match the application requirements. Configuration programming can be as simple as selecting a specific controller model that matches the specific application requirements, or it can require keyboard selection of the proper software options in a more complex controller. Universal type controllers, typically applied as zone-level controllers for VAV or other terminal units, are usually preprogrammed with several control sequences resident in memory. In these cases, configuration programming requires selecting the proper control sequence to match the application through device strapping or keyboard code entry.

**SYSTEM INITIALIZATION PROGRAMMING**

System initialization programming consists of entering appropriate startup values using a keypad or a keyboard. Startup data parameters include setpoint, throttling range, gain, reset time, time of day, occupancy time, and night setback temperature. These data are equivalent to the settings on a mechanical control system, but there are usually more items because of the added functionality of the digital control system.

**DATA FILE PROGRAMMING**

Data file programming may or may not be required depending on whether the controller is a fixed-function or variable-function device. Zone-level controllers are typically fixed function since the applications and control sequences are generally standardized. In these controllers, the input terminals are dedicated to a control relay or specific type of actuator. The need for data files is minimized. The processor always knows what to look for as it scans those points, and it knows how to process the data.

System-level controllers are variable-function and are more universal in application. These controllers must be able to perform a wide variety of control sequences with a broad range of sensor input types and control output signals. System-level controllers require more extensive data file programming. For the controller to properly process input data, for example, it must know if the point type is analog or digital. If the point is analog, the controller must know the sensor type, the range, whether or not the input value is linear, whether or not alarm limits are assigned, what the high and low alarm limit values are if limits are assigned, and if there is a lockout point. See Table 2. If the point is digital, the controller must know its normal state (open or closed), whether the given state is an alarm state or merely a status condition, and whether or not the condition triggers an event-initiated program.

<p>| <strong>Table 2. Typical Data File for Analog Input.</strong> |</p>
<table>
<thead>
<tr>
<th><strong>Point Address</strong></th>
<th><strong>User Address</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Point type</td>
<td>Regular or calculation</td>
</tr>
<tr>
<td>Sensor</td>
<td>Platinum (0 to 100F)</td>
</tr>
<tr>
<td>Physical terminal assigned</td>
<td>16</td>
</tr>
<tr>
<td>Use code</td>
<td>Cold deck dry bulb</td>
</tr>
<tr>
<td>Engineering unit</td>
<td>F</td>
</tr>
<tr>
<td>Decimal places for display</td>
<td>XXX.X</td>
</tr>
<tr>
<td>High limit</td>
<td>70.0</td>
</tr>
<tr>
<td>Low limit</td>
<td>40.0</td>
</tr>
<tr>
<td>Alarm lockout point</td>
<td>Point address</td>
</tr>
<tr>
<td>Point descriptor</td>
<td>Cold deck temperature</td>
</tr>
<tr>
<td>Alarm priority</td>
<td>Critical</td>
</tr>
</tbody>
</table>

**CUSTOM CONTROL PROGRAMMING**

Custom control programming is the most involved programming category. Custom control programming requires a step-by-step procedure that closely resembles standard computer programming. A macro view of the basic tasks is shown in Figure 13.
Analyze Control Application

The systems analysis step in writing a custom control programs requires that the control engineer thoroughly understand the process controlled. The output of the systems analysis is normally a system drawing and a concise and clearly stated sequence of operation.

Partition Into Control Loops

The next step is to partition the entire process into individual control loops. The Control Fundamentals section defines a control loop as a process in which a controller compares the measured value of a controlled variable to a desired value or setpoint. The resulting output of the controller goes to an actuator that causes a control agent to lessen the deviation between actual and desired values (Fig. 14). Control loops can be complex when limit control is needed or when several actuators are controlled in sequence to maintain the controlled variable. At this step a flow chart should be drawn showing all relationships influencing the controlled variable between the controller and actuators.

Determine Inputs and Outputs

The next step in custom control programming is to determine the inputs and outputs associated with each control loop. This establishes the data file associated with the program.

A typical central fan system may require several control loops including various combinations of:
- Discharge-air temperature control
- Mixed-air temperature control
- Hot-deck temperature control
- Cold-deck temperature control
- Humidity or dewpoint control
- IAQ control
- Ventilation control
- Supply fan static pressure control
- Return fan airflow control
Design, Write, and Compile Program

The actual process of designing and writing the control loop programs can be a very complex or a relatively straightforward procedure, depending on the language processing software provided for the controller. The microprocessor-based controller understands instructions only at the most elementary language level, i.e., strings of 1s and 0s or machine code. Because of this, language processing software is often required. This software translates the instructions of a control program written in an easier-to-use high-level language into actual machine code. The assembler is normally associated with a lower level assembly language while the compiler, object oriented, or interpreter is normally associated with a higher level language. Most system level controllers today are programmed using an object oriented (graphical) language.

Object-oriented languages often are custom software packages tailored to the requirements of a specific vendor's controller. Control sequences are built by selecting preprogrammed control blocks, for example the PID algorithm, and linking them with other control blocks. Although this process requires little or no knowledge of programming, it does require in-depth knowledge of the control blocks and the specific HVAC process.

Debug, Install, Enter Data Files, and Test

Regardless of the custom control program used, each program must be debugged to assure proper operation. When programs are written on a host machine, special debug and simulation programs are frequently employed prior to installing the program in the controller. Debug programs test for syntax (language) and procedural errors. Simulation programs allow inputs and outputs to be simulated and a static test of the program to be run. After debug and error correction, the program and associated data files are loaded into the controller and a full system check is made under normal operating conditions to assure proper operation.

Some systems allow graphically constructed programs to be monitored live in their actual executing environment with inputs, outputs, and intermediate signal values updating continuously.

TYPICAL APPLICATIONS

ZONE-LEVEL CONTROLLER

Zone-level controllers can be applied to a variety of types of HVAC unitary equipment. Several control sequences can be resident in a single zone-level controller to meet various application requirements. The appropriate control sequence is selected and set up through either a PC for the system or through a portable operator’s terminal. The following two examples discuss typical control sequences for one type of zone-level controller used specifically for VAV air terminal units. For further information on control of terminal units, refer to the Individual Room Control Applications section. As stated in the introduction, the following applications are for stand-alone controllers. See the Building Management System Fundamentals section for network applications.

EXAMPLE 1. VAV COOLING ONLY

In a pressure independent VAV cooling only air terminal unit application the zone-level controller controls the primary airflow independent of varying supply air pressures. The airflow setpoint of the controller is reset by the thermostat to vary airflow between field programmable minimum and maximum settings to satisfy space temperatures. On a call for less cooling, the damper modulates toward minimum. On a call for more cooling, the damper modulates toward maximum. The airflow control maintains the airflow at whatever level the thermostat demands and holds the volume constant at that level until a new level is called for. The minimum airflow setting assures continuous ventilation during light loads. The maximum setting limits fan loading, excessive use of cool air, and/or noise during heavy loads.

EXAMPLE 2. VAV COOLING WITH SEQUENCED ELECTRIC REHEAT

In a VAV cooling air terminal unit application with sequenced electric reheat, an adjustable deadband is provided between the cooling and the reheat cycle. During cooling the control mode is constant discharge temperature, variable volume. On a call for less cooling, the damper modulates toward minimum flow. The damper remains at minimum cooling through a deadband. On a call for reheat, the damper goes from minimum flow to reheat flow to ensure proper air distribution and prevent excessively high discharge temperatures and to protect the reheat elements. In this sequence, duct heaters are cycled and...
staged by a PI algorithm with software heat anticipation. See Figure 15. During reheat, the control mode changes to constant volume, variable discharge temperature.

An example of this approach follows for control of a hot water converter:

Step 1—Develop flow schematic of the process to be controlled (Fig. 16).

Fig. 15. Control Sequence for VAV Cooling with Sequenced Electric Reheat.

Fig. 16. Schematic of Steam to Hot Water Converter.

Step 2—Identify required sensors, actuators, and operational data (Fig. 17). Refer to the Chiller, Boiler, and Distribution System Control Applications section for a symbol legend.

Fig. 17. Schematic Illustrating Sensors, Actuators, and Operational Data for Steam to Hot Water Converter.

If the DDC system is provided with a BMS having a color monitor, a graphic may be required to be displayed with live, displayable and commandable points (12 total). If a BMS is not provided, the points may be required to be displayed on a text terminal (fixed or portable) at the system level controller.
**Step 3**—Write a detailed sequence of operation for the process.

The hot water pump starts anytime the outside air temperature drops to 52°F, subject to a software on-off-auto function.

When hot water pumping is proven by a current sensing relay, converter controls are energized. Hot water temperature setpoint varies linearly from 120°F to 170°F as the outside air temperature varies from 60°F to 0°F. The converter steam valve is modulated to maintain a converter leaving water temperature according to a varying setpoint schedule.

The steam valve closes anytime hot water pumping is not proven and anytime the valve actuator loses motive power.

**Step 4**—Develop the detailed flowchart.

**Step 5**—Write the program.

See the Air Handling System Control Applications section and the Chiller, Boiler, and Distribution System Control Applications section for other examples of Microprocessor-Based/DDC systems.

![Flowchart Example](image_url)

*Fig. 18. Flowchart Example.*