Overview of Direct Digital Controls

For Building Automation Systems

This document is meant to be a brief introduction to the ideas behind Direct Digital Control (DDC) of Building Automation Systems (BAS).

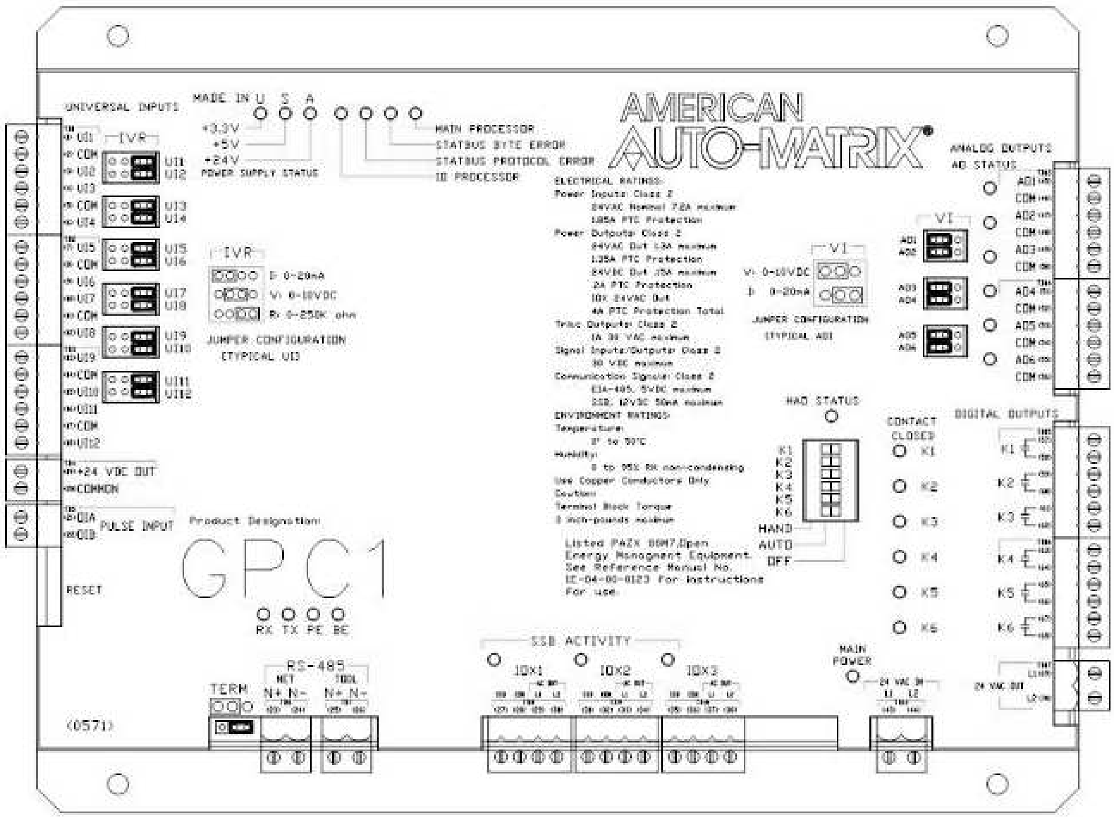
Essentially, direct digital controls in BAS systems these days are dedicated computers, complete with an embedded OS, usually a programming language of some sort, ROM, RAM, input and output ports, communications port(s), and so forth. Mounted in a "black box". Although the device may, in fact, not be in a black box. Sometimes they're simply an electronic board that is mounted on a back plane board, which in turn is attached inside a box or cabinet of the installer's choosing. Or the "computer on a board" may simply be screwed down somewhere inside the housing of the equipment which it is controlling. These days there are rather complete computer systems buried inside all sorts of equipment. Which are replacing hard wired, discrete control components such as switches, relays, timers, sequencers, and so forth. They're also replacing custom designed, special purpose electronic control boards (or IC chips) which have been commonly used for control purposes. The reason for this? It's often both easier and cheaper to simply program a general purpose microprocessor system to accomplish the same functions as custom built circuitry or a specialized IC chip. Add, that when changes to the operation of the device is desirable, it's simply a matter of re-programming the microprocessor system vice having to design and test a new discrete electronic system and then re-tool a manufacturing production line in order to make the new circuit board. In fact, many modern DDC controllers made these days have their software (firmware) for their operating system (OS) and other functions residing in "flash" memory. So that upgrades or changes can be made in the field simply by downloading the new version of the OS and control programming into their memories.

A single DDC controller designed for control of commercial/industrial grade equipment, that's about the size of an average hardcover book, may be used to replace what before took a dozen (or many dozens) of discrete relays, timers, selectors, sequencers, analog proportional controllers, analog to digital converters, digital to analog converters, and so forth. All of which took a LOT of room to install and mount. And all of which entail laborious (and manpower consuming) design, layout, installation, and wiring. To implement what's called "relay logic" to control whatever.

Then, if a change to this relay logic was required, for whatever reason, there must needs be a lot more labor intensive redesign, re-wiring, component replacement and/or relocation, and so forth.

In the case of the average DDC installation, if one needs to change the way the system is operated, typically the input and output wiring and component location remains as it was and all one needs to do is to change the control programming code itself.

This is an especially important consideration in the control of commercial/industrial equipment and installations since almost no two buildings or installations, and the equipment they contain, are the same.



The above is a drawing of a typical DDC controller. The actual controller measures approximately 8.5 inches in width and is 6.5 inches in height. Connections on the left hand side allow one to attach wires from input devices (sensors and such). They're "universal" inputs. Which means they can accommodate sensors that are dry contacts; sensors which output varying, proportional voltage or milliamp signals; electrical resistance type sensors, etc. On the upper left are connections for wiring analog signal outputs (varying voltage or milliamp) to actuators (positioning motors and such). Below that are connections for wiring relays controlled by the controller to other, larger relays; motor starters, or whatever devices one wants to start, stop, or enable using a relay output. Below left are connections to wire the controller into a network of controllers so that it may exchange data with other controllers, or receive data or commands from other control devices. And the network allows a central monitoring and control station, usually a desktop or laptop PC, to collect information and display it to maintenance personnel, and allows them to make changes to setpoints, schedules, or whatever. Middle bottom has connections for adding even more input and output devices, which can communicate with this controller via a separate, dedicated network connection. In case the "onboard" number of inputs and outputs is not adequate for the application. The controller has RAM, ROM, flash memory, an OS, a programming language interpreter so that custom control programs may be written for it and downloaded into it which it will run, etc. In fact, one can actually write several, separate programs for it that'll be downloaded into partitioned memory areas so that the controller could be used to control several, different types of equipment simultaneously. It also contains numerous pre-made control loops, mathematical routines, routines that make scheduling simplicity itself, and so forth in EEPROM that can be utilized, which makes the programming of this controller much simpler to accomplish. Memory is battery backed so that upon power failure, then restoration it will commence its control functions properly. Having not lost previous saved data, setpoint information, and so on. The program within it will automatically restart. The above is just an example. There are many designs and models from many manufacturers.

One might wonder why DDC controllers are not even smaller. After all, even modern wrist watches often have full fledged microprocessor systems buried within them. The answer is simple. DDC controllers interface with, receive inputs from, and send output to commercial/industrial grade sensors and actuators of all sorts. Thus they contain electronic circuitry capable of handling voltages and current flows FAR beyond anything that a wrist watch microprocessor has to deal with. So components of the inputs and outputs must be beefier in design to accommodate this. In addition, the power supply and conditioner within a DDC device is more rugged and larger. Add that the selection of the particular microprocessors used, and the associated circuitry and chips such as RAM, ROM, IO chips, D-A and A-D chips, and so forth is different. More durably constructed components, able to tolerate greater extremes of hostile operating conditions and environments are normally used. Mountings and connections are sturdier; heat sinks are larger, and so forth. And DDC device designers aim at having their equipment function reliably under adverse conditions, for 15 to 20 years. Or even longer.

Such devices these days are controlling heating, air conditioning, ventilation, and refrigeration equipment for buildings. They're also controlling lighting, security systems, fire detection and protection systems, water supply equipment, waste water handling equipment, and much more.

In addition to simply controlling things, DDC systems are commonly used for data collection and analysis. Commonly these days' data is collected and analyzed to determine such things as energy usage and efficiency within a building and its various parts. For purposes of billing customer occupants who've rented/leased space within the building. And/or for the purpose of determining where and what tactics might be employed, programming changes made, and so forth to reduce the utility bills and energy consumption. Data collected over time (trends), or live observation of current operating condition data can also help maintenance and operating personnel detect faults and failures (or impending failures) when they happen, or even before total failure of some component. It's not uncommon in buildings with DDC systems and suitably trained and observant in-house maintenance personnel to detect and start corrective actions on a problem before space occupants are even aware that there is a problem.

In any event, for anyone interested in learning about DDC controls in the BAS applications, it's not a field where simply being interested in or knowing something about, “computers” is enough. Computers are, in all actuality, as dumb as a box of rocks. Well, maybe a little smarter, but not by much. They do not think, per se, in the way we humans can think. Nor are they much good at innovating, creating new ideas or things which never existed before, foreseeing possible future problems or events before they actually occur, adapting to ever changing situations, coming up with new uses for old ideas or methods, and so forth. They simply do what they’re told to do. Right or wrong. Although they can do what they do very fast. And repetitively, without tiring, without getting bored, and without their attention wandering.

DDC devices run programs, written by humans. If the program is faulty, or if the programmer did not include some needed routine or piece of data that’d prevent improper operation of some sort of equipment, or if the programmer did not make adequate provisions for “what if” situations, and so forth … then the computer does not know the difference. It’ll run the program and do what it was told, and ONLY what it was told, until it fails or is turned off. Even if what it is doing is wrong and possibly destructive to the controlled equipment, wasteful, dangerous, and so forth. It’s the old GIGO principle. Garbage-In, Garbage-Out.

So a DDC technician or engineer also needs to know, or to learn about, those things which the DDC controllers will be controlling. So that he or she may be able to adequately ensure that the right sensors and actuators are selected in order to be able to adequately and properly get the job done (the inputs and outputs). And that the DDC controller itself (the computer) is properly programmed and configured to get its job done in proper and acceptable fashion. Then, be able to test a live and operational system in order to be able to determine that everything is running correctly.

This does not mean a DDC technician or engineer must be expert about every system that he or she works with and seeks to control. But one needs to know at least the basics of the equipment to be controlled itself, plus related rules and regulations and guidelines published by the governing authorities for that type of equipment. i.e. Applicable fire codes, electrical codes, mechanical codes, and so forth. One does not need to be able to break out the gages and torch set to install or repair a DX (direct expansion) condensing unit in order to be a DDC technician. But if one is going to control it, one had BETTER know what it is, what it does, the basics of how it’s built and is supposed to operate (as per manufacturer’s instructions), and what TO DO plus what NOT TO DO when operating it. Or when writing a program for a controller that’ll operate it. And it’s kind of hard to set up a system to monitor and take action upon failure of a 3 phase power supply if one doesn’t even know what a 3 phase power supply is, or what’s harmful about or what constitutes a “phase failure”, or “phase imbalance” condition.

Get the idea? Being a computer geek and knowing how to program a computer game is not enough. You need to learn about the real world equipment you’re trying to get that “computer in a box” to control and monitor.

This is the reason that on this site I have placed some introductory and instructional material about a variety of subjects. Note that the material IS introductory in nature.

In the HVAC (heating, ventilation, and air conditioning) area alone, a person wanting to be a decent HVAC technician, who can do actual and adequate installation, servicing, and repairs to commercial and industrial grade HVAC equipment, is facing an average of 5 years academic study and apprenticed field experience. Even then, he or she is going to be a relative “newbie”. Adequate for most work, but not an expert. A top technician, a real expert, usually has 10 to 20 years experience, the equivalent of a two-year (or better) college degree in formal academic training. Numerous other factory or advanced classes he or she has attended over time. Plus he or she will normally haven taken and passed a number of certification/licensing tests of various sorts. (Talking about what is usual and normal and expected here in the U.S.) Residential HVAC work requires much less as in the case of residential equipment, most of it is pretty simple and basic stuff.

Same as above applies to electricians as concerns the amount of training and experience a commercial or industrial equipment electrician must achieve. Commercial/industrial fire alarm or security system technicians also have a lot to learn and spend many years becoming good at what they do. It is much the same with other technicians of various sorts.

So don’t look for those kinds of instructional materials here. The fields are far too complex, and entail too much knowledge for me to put such manuals here. Not to mention, 5 gigs of storage space is not nearly enough for a decent, advanced reference library.

However, you’re welcome to use this document or any others I have placed on this site if you’re interested and find them useful to you. They should all be free to use, copy, and distribute.

A person wanting to become a DDC technician or engineer needs to become familiar with the types of equipment to be controlled. He or she should also be familiar with computer systems, basic networking principles and practices, have at least a passing knowledge of some sort of programming language, understand the basics of electricity and electronics (advanced knowledge is NOT required unless you’re getting into the business of designing DDC controllers themselves), and have a knowledge of control theory itself.

After that, it’s simply a matter of learning at least one manufacturer’s line of DDC controllers and support software. Exact details vary from manufacturer to manufacturer, but the principles are the same. Learn one system, then learning another, different one, is a piece of cake.