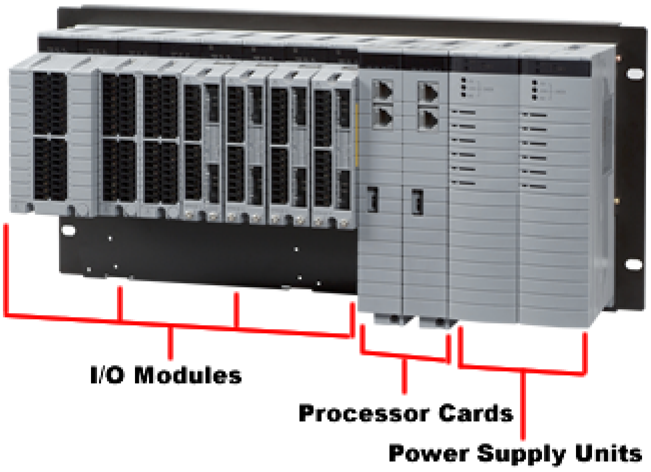
Controller System for Industrial Automation

The element linking the measurement and the final control element is the controller. Before the advent of computers, the controllers are usually single-loop PID controllers. These are manufactured to execute PID control functions. These days, the controllers can do a lot more, however, easily 80 to 90% of the controllers are still PID controllers.



Analogue vs Digital Controllers

It is indeed difficult to say that analogue controllers are definitely better than digital controllers. The point is, they both work. Analogue controllers are based on mechanical parts that cause changes to the process via the final control element. Again like final control elements, these moving parts are subjected to wear and tear over time and that causes the response of the process to be somewhat different with time. Analogue controllers control continuously.

Digital controllers do not have mechanical moving parts. Instead, they use processors to calculate the output based on the measured values. Since they do not have moving parts, they are not susceptible to deterioration with time. Digital controllers are not continuous. They execute at very high frequencies, usually 2-3 times a second.

Analogue controllers should not be confused with pneumatic controllers. Just because a controller is analogue does not mean it is pneumatic. Pneumatic controllers are those that use instrument air to pass measurement and controller signals instead of electronic signals. An analogue controller can use electronic signals. Compared to pneumatic controllers, electronic controllers (can be analogue or digital) have the advantage of not having the same amount of deadtime and lag due to the compressibility of the instrument ai

.

Basic Process Control Systems

Distributed Control System (DCS)

The DCS is a control system which collects the data from the field and decides what to do with them. Data from the field can either be stored for future reference, used for simple process control, use in conjunction with data from another part of the plant for advanced control strategies.

What must be in the DCS for it to be able to do so much?

Operator Console

These are like the monitors of our computers. They provide us with the feedback of what they are doing in the plant as well as the command we issue to the control system. These are also the places where operators issue commands to the field instruments.

Engineering Station

These are stations for engineers to configure the system and also to implement control algorithms.

History Module

This is like the harddisk of our PCs. They store the configurations of the DCS as well as the configurations of all the points in the plant. They also store the graphic files that are shown in the console and in most systems these days they are able to store some plant operating data.

Data Historian

These are usually extra pieces of software that are dedicated to store process variables, set points and output values. They are usually of higher scanning rates than that available in the history module.

Control Modules

These are like the brains of the DCS. Specially customized blocks are found here. These are customized to do control functions like PID control, ratio control, simple arithmetic and dynamic compensation. These days, advanced control features can also be found in them.

I/O

These manage the input and output of the DCS. Input and output can be digital or analogues. Digital I/Os are those like on/off, start/stop signals. Most of the process measurements and controller outputs are considered analogue. These are the points where the field instruments are hard-wired to.

All above mentioned elements are connected by using a network, nowadays very often used is Ethernet.

The practical and technological boundaries between a Distributed Control System DCS, Programmable Logic Controller PLC and Personal Computer PC control are blurring. Systems traditionally associated with process control are being used in discrete applications. Likewise, traditionally discrete solutions are used increasingly in both batch and continuous process control.

Today's control hardware are constructed from many of the same standard industry components such as Intel processors. Therefore the only real difference between control systems is at the software level.

ABB / Industrial IT - Advant Master DCS

Advant OCS (Open Control System) is an ABB solution for operators to improve their manufacturing productivity and achieve sustainable competitive advantages.

In 1992, based on the success of the Master systems in the 80's, the Master system began its evolution to Advant OCS. This evolution introduced high capacity controllers and I/O with an improved redundancy scheme. Also included were modern UNIX workstations, and in 1996 S800 I/O was added providing modular flexible remote I/O.

In 2000, Advant OCS with Master Software began its next step in the evolution process with the introduction of Industrial IT enabled products. ABB's commitment to protecting your investment continues with these enhancements by providing connectivity to our latest control offering.

A versatile and complete range of process I/O systems within the Advant family enables optimal user configurations:

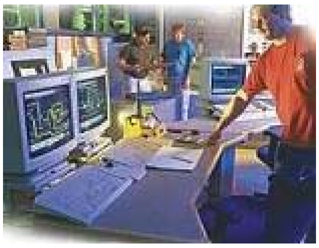
S100I/O - A rack-based I/O system for AC400 controllers
S600I/O - A rack-based I/O system for AC100 controllers
S800I/O - A highly modularized and flexible I/O-system

Numerous characteristics and functions facilitate and improve operation, monitoring, and reengineering of each process in a company. 800xA Operations (Process Portal) and the proven AdvaCommand for Unix solution (based on HP-UX) are available as an operator station for Advant

OCS with Master Software.

The intuitive operator software provides consistent access and interaction with data from multiple control and I/O to plant and enterprise information.

ABB Advant Master Control Systems



Honeywell Experion™ Process Knowledge System (PKS)

Experion is Honeywell's unified system for process, business, and asset management that helps industrial manufacturers increase their profitability and productivity.

Experion takes customers well beyond Distributed Control System (DCS) functionality with an advanced automation platform solution and innovative application integration to improve business performance and peace of mind. And there's no need to worry about upgrading from TDC 2000®/TDC 3000® or TotalPlant® Solution (TPS).

The unique, patent pending design of Series C combines sleek styling and function to provide process I/O with reduced footprint, easier installation and maintenance, and longer life. The Series C form factor benefits extend to multiple modules, such as the Series C C300 Controller, the Fieldbus Interface Module, the Control Firewall, and HART analog modules.

The Control Execution Environment (CEE) is the common core software used in the various controllers supported by ExperionTM. This includes the C200 Process Controller, the C300 Process Controller, the Application Control Environment (ACE) and the C200 Simulation Environment (SIM-C200). The CEE provides an execution and scheduling environment where control strategies are configured from a rich set of standard and optional function blocks using a single builder tool, Control Builder.

Function blocks are grouped and wired together in a container to perform a specific control function such as a valve control strategy. The Control Execution Environment (CEE) supports two types of containers: the Control Module in which continuous and discrete controls are combined; and an SCM, which is used for sequence control. Function blocks support the complete control application range, such as continuous, discrete and batch control.

Emerson Process Management / DeltaV

DeltaV is the creation of Emerson Process Management's technological innovators, who worked in an off-site "out-of-the-box" think tank to build an automation system that could integrate and leverage today's digital world and cutting-edge technological innovations to make a value step-change in the process industries.

The name DeltaV is derived from the engineering equation for acceleration: dv/dt, the change in velocity over the change in time. The DeltaV system makes planning, engineering, installing, commissioning, training, operating, and maintaining your process EASY, which accelerates your success in improving your plant performance.

The DeltaV system scales the complete range of applications from an isolated process area to a complete plant-wide automation system. Whether you need tens of I/O or tens of thousands of I/O-any size you want! The DeltaV system provides all the tools to manage your process easier than ever before.

The complete family of controllers is available to power your most advanced control strategies. Full controller and power supply redundancy is available for your mission-critical applications. The controller and I/O sub-system is rated for Class I, Division 2 and Zone 2 environments to reduce your installation costs.

DeltaV workstations are based on the latest Intel-based microprocessors running the Microsoft Windows XP /Windows 2003 operating system. A complete range of applications is provided to cover system configuration, operator interface, engineering, maintenance, and integration functions.

The DeltaV control network—a high-speed Ethernet LAN—provides system communications and connects the various system nodes. The control network can be fully redundant. DeltaV remote services extend the operations, engineering, and diagnostic applications across your enterprise network.

Unlike PLC/HMI solutions, the completely integrated DeltaV system features a single database that coordinates all configuration activities. System configuration is globally distributed in the run-time environment.

Emerson - DeltaV

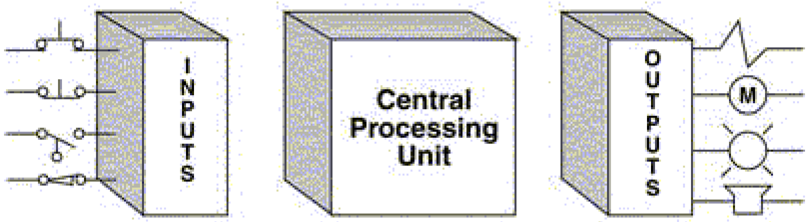
Hybrid Systems

Programmable Logic Controller PLC

A **PROGRAMMABLE LOGIC CONTROLLER** (PLC) is an industrial computer control system that continuously monitors the state of input devices and makes decisions based upon a custom program, to control the state of devices connected as outputs.

Almost any production line, machine function or process can be automated using a PLC. The speed and accuracy of the operation can be greatly enhanced using this type of control system. But the biggest benefit in using a PLC is the ability to change and replicate the operation or process while collecting and communicating vital information.

A PLC consists of following main parts:



What is a PLC input/output?

INPUT	**OUTPUT**
Sensing Devices	Valves
Switches and Pushbuttons	Solenoids
Proximity Sensors	Motor
Limit Switches	Actuators
Pressure Switches	Pumps

PLC Operations consist of four steps

1. Input Scan: Scans the state of the Inputs
2. Program Scan: Executes the program logic
3. Output Scan: Energize/de-energize the outputs
4. Housekeeping

Rockwell Automation AB -

SLC 500 Programmable Logic Controller

The SLC 500 family is a growing family of small programmable controllers built around two hardware options: a fixed controller or a modular controller. The modular controller offers you maximum flexibility in system configuration. Wth its multiple processor choices, numerous power supply options and extensive I/O capacity, the modular SLC 500 controller allows you to create a system specifically designed for your application.

The fixed controller provides the power supply, inputs and outputs, and processor in one unit. It also offers a 2-slot expansion chassis for increased flexibility. The programming tools and most I/O modules are compatible between the two hardware options, so you can cost effectively solve a broad range of applications.

PLC-5 / 1771 Controller System

PLC-5 processors are available in a large range of forcible I/O (512 maximum through 3072 maximum) and maximum user memory (6K through 100K words). All are capable of controlling remotely-located I/O. The maximum number of I/O locations ranges from 5 through 125.

A PLC-5 processor communicates across the 1771 backplane to 1771 I/O modules in the chassis in which the processor resides. A PLC-5 processor can communicate with I/O across a DeviceNet or Universal Remote I/O link. Selected models of PLC-5 processors can communicate with I/O across a ControlNet or Extended Local I/O link. I/O adapters for 1771 I/O are available for ControlNet, Universal Remote I/O, and Extended Local I/O links. General communication messages can be sent from or received by PLC-5 processors across DeviceNet, ControlNet, or Ethernet networks, as well as Data Highway Plus, RS-232-C, RS-422-A, or RS-423-A networks. You can add a DeviceNet port to any PLC-5 processor with a 1771-SDN scanner module. Each Ethernet PLC-5 processor has an on-board Ethernet port, and you can add an Ethernet port to any PLC-5 processor with a 1785-ENET Ethernet Interface Module.

Allen-Bradley

SLC 500 Processors

Allen-Bradley

PLC-5 Family

Siemens

Simatic S7-300

Siemens Simatic PLC's -

Simatic S7

SIMATIC S7 lets you implement an array of different technologies with integrated solutions.

It is easy to parameterize functions in the TIA range, using the interactive screen forms embedded in STEP 7.

The STEP 7 basic package already includes blocks for control tasks. These blocks can be loaded onto any CPU. In the field of microautomation, STEP 7-Micro/WIN also offers a user-friendly assistant for programming technological functions.

For the count, measure, control and position processes, there are CPUs available featuring the technological functions as an integral part of the operating system.

The technological tasks are executed by way of the inputs and outputs directly integrated on the CPU.

The Technology CPUs 315T and 317T integrate performant PLCopen-certified technology and motion control functions right into the standard SIMATIC CPU.

Simatic ET200S

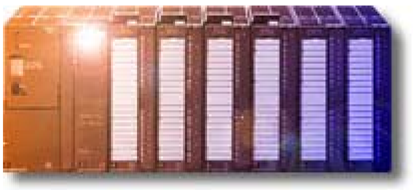
SIMATIC ET 200S is the distributed I/O station that guarantees lasting savings in life-cycle costs with the highest flexibility. Installation couldn't be easier and the bit-modular design of the ET 200S enables multifunctional use of the station.

The multifunctional ET200S can now communicate over PROFINET as well as over PROFIBUS, giving the proven and field-tested I/O station all the advantages of Ethernet communication. PROFIBUS-DP is the fastest, most standardised network at field level. It has been standardised in accordance with the European Norm EN 50170. PROFIBUS is completely integrated in the new SIMATIC world, both in terms of hardware and software.

Modules for the ET 200S include power modules, digital or analog input and output modules, technology modules and motor starters. The ET200S is also equipped for fail-safe signal modules and motor starters. Completely new is the ET 200S frequency converter, which makes it possible to perform variable-speed drive tasks for up to 4 kW power.

The SIMATIC ET 200 distributed I/O system makes it possible to connect digital and analog inputs/outputs with the central controller. ET 200 also allows to use intelligent I/O modules in distributed configurations.

Terminal modules with FastConnect technology that needs no stripping of cables help minimize rewiring time.



Siemens

Simatic ET200

.

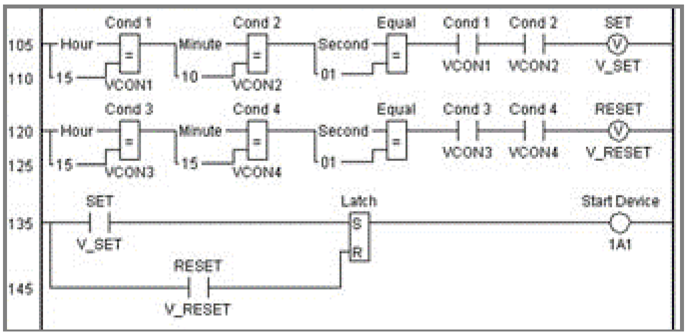
PLC Ladder Logic

PLC Ladder Logic

Ladder logic is a method of drawing electrical logic schematics. It is now a graphical language very popular for programming Programmable Logic Controllers (PLCs). It was originally invented to describe logic made from relays. The name is based on the observation that programs in this language resemble ladders, with two vertical "rails" and a series of "rungs" between them. In Germany and elsewhere in Europe, the style is to draw the rails horizontal along the top and bottom of the page while the rungs are drawn sequentially from left to right.

A program in ladder logic, also called a ladder diagram, is similar to a schematic for a set of relay circuits. Ladder logic is useful because a wide variety of engineers and technicians can understand and use it without much additional training because of the resemblance.

Ladder logic is widely used to program PLCs, where sequential control of a process or manufacturing operation is required. Ladder logic is useful for simple but critical control systems, or for reworking old hardwired relay circuits. As programmable logic controllers became more sophisticated it has also been used in very complex automation systems.



Most manufacturers of programmable logic controllers also provide associated ladder logic programming systems. Typically, the ladder logic languages from two manufacturers will not be completely compatible; ladder logic is better thought of as a set of closely related programming languages rather than one language. Even different models of programmable controller within the same family may have different ladder notation such that programs cannot be seamlessly interchanged between models.

Ladder logic can be thought of as a rule-based language, rather than a procedural language. A "rung" in the ladder represents a rule. When implemented with relays and other electromechanical devices, the various rules "execute" simultaneously and immediately. When implemented in a programmable logic controller, the rules are typically executed sequentially by software, in a loop. By executing the loop fast enough, typically many times per second, the effect of simultaneous and immediate execution is obtained. In this way it is similar to other rule-based languages, like spreadsheets or SQL. However, proper use of programmable controllers requires understanding the limitations of the execution order of rungs.

****

For more details please see the following files at “Fieldbus Class room “

Process control systems (1.75 Mb pdf file)

PLC Primer (252K PDF file)

Programming and Documentation Pads (ALL) (720K PDF file)

PLC Program Listing (184K PDF file)

Input/Output Listing (200K PDF file)

Register-Word/Internal Output Usage Listing (196K PDF file)

PLC Automation Project Notes (168K PDF file)

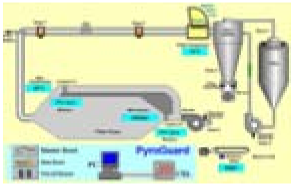
PLC Glossary of Terms **(92K PDF file)**

Logic Symbols, Truth Tables and Equivalent Ladder/PLC Logic Diagrams **(20K PDF file)**

Electrical Relay Diagram and P&ID Symbols **(88K PDF file)**

****

Supervisory Control And Data Aquisition SCADA



Following we describe the SCADA systems in terms of their architecture, their interface to the process hardware, the functionality and the application development facilities they provide.

SCADA systems have made substantial progress over the recent years in terms of functionality, scalability, performance and openness such that they are an alternative to in house development even for very demanding and complex control systems.

What does SCADA mean?

SCADA stands for Supervisory Control And Data Acquisition. As the name indicates, it is not a full control system, but rather focuses on the supervisory level. As such, it is a purely software package that is positioned on top of hardware to which it is interfaced, in general via Programmable Logic Controllers (PLC's), or other commercial hardware modules.

SCADA systems are used not only in industrial processes: e.g. steel making, power generation (conventional and nuclear) and distribution, chemistry, but also in some experimental facilities such as nuclear fusion. The size of such plants range from a few 1000 to several 10 thousands input/output (I/O) channels. However, SCADA systems evolve rapidly and are now penetrating the market of plants with a number of I/O channels of several 100 thousands I/O's

SCADA systems used to run on DOS, VMS and UNIX; in recent years all SCADA vendors have moved to NT, Windows XP, Windows Server 2003 and some also to Linux.

1. Architecture

This section describes the common features of the SCADA products.

Hardware Architecture

One distinguishes two basic layers in a SCADA system: the "client layer" which caters for the man machine interaction and the "data server layer" which handles most of the process data control activities. The data servers communicate with devices in the field through process controllers. Process controllers, e.g. PLC's, are connected to the data servers either directly or via networks or fieldbuses that are proprietary (e.g. Siemens H1), or non-proprietary (e.g. Profibus). Data servers are connected to each other and to client stations via an Ethernet LAN.

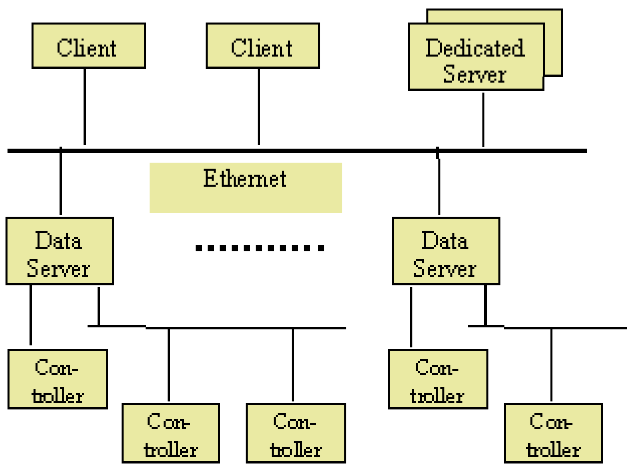


Figure 1: Typical Hardware Architecture

Software Architecture

The products are multi-tasking and are based upon a real-time database (RTDB) located in one or more servers. Servers are responsible for data acquisition and handling (e.g. polling controllers, alarm checking, calculations, logging and archiving) on a set of parameters, typically those they are connected to.

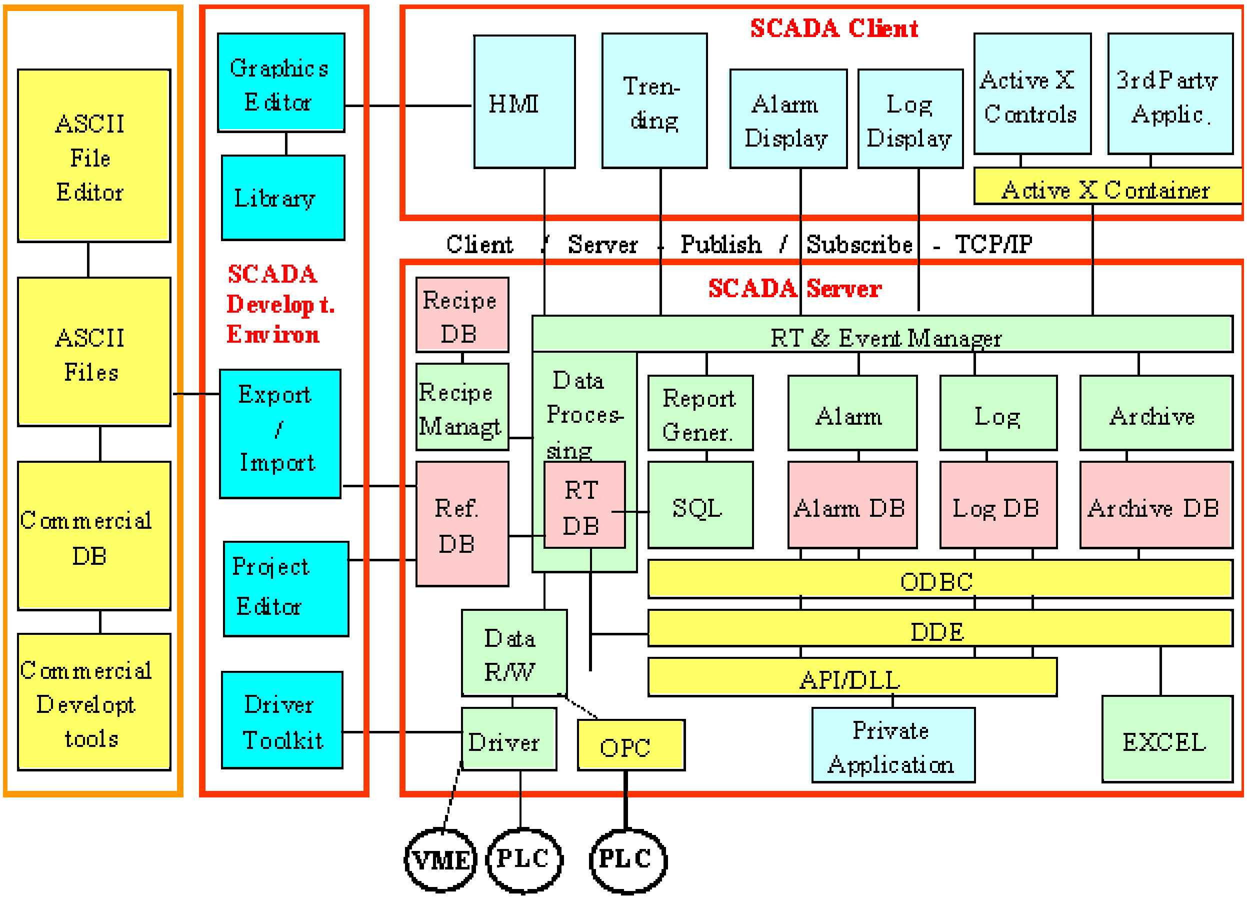


Figure 2: Generic Software Architecture

However, it is possible to have dedicated servers for particular tasks, e.g. historian, datalogger, alarm handler. The figure above shows a generic SCADA software architecture.

Communications

Internal Communication:

Server-client and server-server communication is in general on a publish-subscribe and event-driven basis and uses a TCP/IP protocol, i.e., a client application subscribes to a parameter which is owned by a particular server application and only changes to that parameter are then communicated to the client application.

Access to Devices

The data servers poll the controllers at a user defined polling rate. The polling rate may be different for different parameters. The controllers pass the requested parameters to the data servers. Time stamping of the process parameters is typically performed in the controllers and this time-stamp is taken over by the data server. If the controller and communication protocol used support unsolicited data transfer then the products will support this too.

The products provide communication drivers for most of the common PLC's and widely used field-buses, e.g., Modbus. Some of the drivers are based on third party products (e.g., Applicom cards) and therefore have additional cost associated with them. VME on the other hand is generally not supported.

A single data server can support multiple communications protocols: it can generally support as many such protocols as it has slots for interface cards.

The effort required to develop new drivers is typically in the range of 2-6 weeks depending on the complexity and similarity with existing drivers, and a driver development toolkit is provided for this.

Interfacing

Application Interfaces / Openness:
The provision of OPC client functionality for SCADA to access devices in an open and standard manner is developing. There still seems to be a lack of devices/controllers, which provide OPC server software, but this improves rapidly as most of the producers of controllers are actively involved in the development of this standard.

The products also provide an Open Data Base Connectivity (ODBC) interface to the data in the archive/logs, but not to the configuration database, an ASCII import/export facility for configuration data, a library of APIs supporting C, C++, and Visual Basic (VB) to access data in the RTDB, logs and archive. The API often does not provide access to the product's internal features such as alarm handling, reporting, trending, etc.

The PC products provide support for the Microsoft standards such as Dynamic Data Exchange (DDE) which allows e.g. to visualize data dynamically in an EXCEL spreadsheet, Dynamic Link Library (DLL) and Object Linking and Embedding (OLE).

Database
The configuration data are stored in a database that is logically centralised but physically distributed and that is generally of a proprietary format.

For performance reasons, the RTDB resides in the memory of the servers and is also of proprietary format.

The archive and logging format is usually also proprietary for performance reasons, but some products do support logging to a Relational Data Base Management System (RDBMS) at a slower rate either directly or via an ODBC interface.

Scalability

Scalability is understood as the possibility to extend the SCADA based control system by adding more process variables, more specialized servers (e.g. for alarm handling) or more clients. The products achieve scalability by having multiple data servers connected to multiple controllers. Each data server has its own configuration database and RTDB and is responsible for the handling of a sub-set of the process variables (acquisition, alarm handling, archiving).

Redundancy

The products often have built in software redundancy at a server level, which is normally transparent to the user. Many of the products also provide more complete redundancy solutions if required.

2. Functionality

Access Control

Users are allocated to groups, which have defined read/write access privileges to the process parameters in the system and often also to specific product functionality.

MMI

The products support multiple screens, which can contain combinations of synoptic diagrams and text.

They also support the concept of a "generic" graphical object with links to process variables. These objects can be "dragged and dropped" from a library and included into a synoptic diagram.

Most of the SCADA products that were evaluated decompose the process in "atomic" parameters (e.g. a power supply current, its maximum value, its on/off status, etc.) to which a Tag-name is associated. The Tag-names used to link graphical objects to devices can be edited as required. The products include a library of standard graphical symbols, many of which would however not be applicable to the type of applications encountered in the experimental physics community.

Standard windows editing facilities are provided: zooming, re-sizing, scrolling... On-line configuration and customization of the MMI is possible for users with the appropriate privileges. Links can be created between display pages to navigate from one view to another.

Trending

The products all provide trending facilities and one can summarize the common capabilities as follows:

- the parameters to be trended in a specific chart can be predefined or defined on-line
- a chart may contain more than 8 trended parameters or pens and an unlimited number of charts can be displayed (restricted only by the readability)
- real-time and historical trending are possible, although generally not in the same chart
- historical trending is possible for any archived parameter
- zooming and scrolling functions are provided
- parameter values at the cursor position can be displayed

The trending feature is either provided as a separate module or as a graphical object (ActiveX), which can then be embedded into a synoptic display. XY and other statistical analysis plots are generally not provided.

Alarm Handling

Alarm handling is based on limit and status checking and performed in the data servers. More complicated expressions (using arithmetic or logical expressions) can be developed by creating derived parameters on which status or limit checking is then performed. The alarms are logically handled centrally, i.e., the information only exists in one place and all users see the same status (e.g., the acknowledgement), and multiple alarm priority levels (in general many more than 3 such levels) are supported.

It is generally possible to group alarms and to handle these as an entity (typically filtering on group or acknowledgement of all alarms in a group). Furthermore, it is possible to suppress alarms either individually or as a complete group. The filtering of alarms seen on the alarm page or when viewing the alarm log is also possible at least on priority, time and group. However, relationships between alarms cannot generally be defined in a straightforward manner. E-mails can be generated or predefined actions automatically executed in response to alarm conditions.

Logging / Archiving

The terms logging and archiving are often used to describe the same facility. However, logging can be thought of as medium-term storage of data on disk, whereas archiving is long-term storage of data either on disk or on another permanent storage medium. Logging is typically performed on a cyclic basis, i.e., once a certain file size, time period or number of points is reached the data is overwritten. Logging of data can be performed at a set frequency, or only initiated if the value changes or when a specific predefined event occurs. Logged data can be transferred to an archive once the log is full. The logged data is time-stamped and can be filtered when viewed by a user. The logging of user actions is in general performed together with either a user ID or station ID. There is often also a VCR facility to play back archived data.

Report Generation

One can produce reports using SQL type queries to the archive, RTDB or logs. Although it is sometimes possible to embed EXCEL charts in the report, a "cut and paste" capability is in general not provided. Facilities exist to be able to automatically generate, print and archive reports.

Automation

The majority of the products allow actions to be automatically triggered by events. A scripting language provided by the SCADA products allows these actions to be defined. In general, one can load a particular display, send an Email, run a user defined application or script and write to the RTDB.

The concept of recipes is supported, whereby a particular system configuration can be saved to a file and then re-loaded at a later date.

Sequencing is also supported whereby, as the name indicates, it is possible to execute a more complex sequence of actions on one or more devices. Sequences may also react to external events.

3. Application Development

Configuration

The development of the applications is typically done in two stages. First the process parameters and associated information (e.g. relating to alarm conditions) are defined through some sort of parameter definition template and then the graphics, including trending and alarm displays are developed, and linked where appropriate to the process parameters. The products also provide an ASCII Export/Import facility for the configuration data (parameter definitions), which enables large numbers of parameters to be configured in a more efficient manner using an external editor such as Excel and then importing the data into the configuration database.

However, many of the PC tools now have a Windows Explorer type development studio. The developer then works with a number of folders, which each contains a different aspect of the configuration, including the graphics.

The facilities provided by the products for configuring very large numbers of parameters are not very strong. However, this has not really been an issue so far for most of the products to-date, as large applications are typically about 50k I/O points and database population from within an ASCII editor such as Excel is still a workable option.

Online modifications to the configuration database and the graphics are generally possible with the appropriate level of privileges.

Development Tools

The following development tools are provided as standard:

- Graphics Editor, with standard drawing facilities including freehand, lines, squares circles, etc. It is possible to import pictures in many formats as well as using predefined symbols including e.g. trending charts, etc. A library of generic symbols is provided that can be linked dynamically to variables and animated as they change. It is also possible to create links between views so as to ease navigation at run-time.
- Database Configuration Tool (usually through parameter templates). It is in general possible to export data in ASCII files so as to be edited through an ASCII editor or Excel.
- Scripting Language
- Application Program Interface (API) supporting C, C++, VB
- Driver Development Toolkit to develop drivers for hardware that is not supported by the SCADA product.

Object Handling

The products in general have the concept of graphical object classes, which support inheritance. In addition, some of the products have the concept of an object within the configuration database. In general the products do not handle objects, but rather handle individual parameters, e.g., alarms are defined for parameters, logging is performed on parameters, and control actions are performed on parameters. The support of objects is therefore fairly superficial.

4. Evolution

SCADA vendors release one major version and one to two additional minor versions once per year. These products evolve thus very rapidly so as to take advantage of new market opportunities, to meet new requirements of their customers and to take advantage of new technologies.

As was already mentioned, most of the SCADA products that were evaluated decompose the process in "atomic" parameters to which a Tag-name is associated. This is impractical in the case of very large processes when very large sets of Tags need to be configured. As the industrial applications are increasing in size, new SCADA versions are now being designed to handle devices and even entire systems as full entities (classes) that encapsulate all their specific attributes and functionality. In addition, they will also support multi-team development.

As far as new technologies are concerned, the SCADA products are now adopting:

- Web technology, ActiveX, Java, etc.
- OPC as a means for communicating internally between the client and server modules. It should thus be possible to connect OPC compliant third party modules to that SCADA product.

5. Engineering

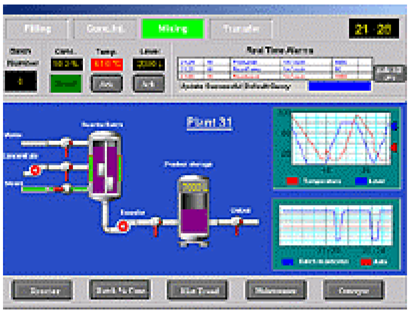
Whilst one should rightly anticipate significant development and maintenance savings by adopting a SCADA product for the implementation of a control system, it does not mean a "no effort" operation. The need for proper engineering can not be sufficiently emphasized to reduce development effort and to reach a system that complies with the requirements, that is economical in development and maintenance and that is reliable and robust. Examples of engineering activities specific to the use of a SCADA system are the definition of:

- A library of objects (PLC, device, subsystem) complete with standard object behavior (script, sequences, ...), graphical interface and associated scripts for animation
- Templates for different types of "panels", e.g. alarms
- Instructions on how to control e.g. a device ...

- A mechanism to prevent conflicting controls (if not provided with the SCADA)
- Alarm levels, behavior to be adopted in case of specific alarms, ...

Wonderware- Intouch

Ivensys - Wonderware / Intouch



InTouch® 9.0 software with SmartSymbols and the IOSetRemoteReferences script function enables users to quickly and easily create and deploy graphical representations of real-time industrial process applications that connect to InTouch tag servers, ArchestrA® Object Templates in Wonderware's Industrial Application Server and I/O Servers.

With SmartSymbols, users can very easily create graphic templates that can be used throughout the entire application. Users can create a graphical object once, attach animations and then save that object as a SmartSymbol. Users can also create standard libraries of SmartSymbols that adhere to their company's standards for color and animation, resulting in graphics that conform to existing practices without requiring a great deal of administration and management. These libraries of SmartSymbols can be exported and imported into other InTouch applications resulting in standards for graphics that can be easily implemented throughout an entire organization. Developing entire InTouch HMI applications becomes as simple as choosing the SmartSymbol graphic from the library manager, selecting the instance reference and dropping it into a window.

When testing applications or modifying graphical objects, users only need to edit the SmartSymbol graphic template and all instances throughout the application will be automatically updated with the new information, resulting in tremendous time savings and a significant reduction in potential errors.

In addition, the IOSetRemoteReferences script function enables users to create graphical faceplates, which can be quickly modified at runtime. Faceplates can be created to model devices and their controls used throughout the application such as valves, pumps and motors. To leverage the IOSetRemoteReferences script function, a user would first create a SmartSymbol graphic template and then associate it with tags using a remote style reference. At runtime, whenever a particular condition occurs or a device such as a push of a button is activated, the IOSetRemoteReferences function updates all of the data references. This update is very fast because all of the data sources in the window are updated using one line of script.

Intellution- iFIX

GE Fanuc - Intellution / iFIX

Proficy HMI/SCADA - iFIX is a powerful Client/Server based HMI/SCADA solution that provides process visualization, data acquisition and supervisory control over manufacturing and production processes. Proficy HMI/SCADA - iFIX gives Operators and Process Engineers the power and security to precisely monitor and control every aspect of their process, equipment and resources. The result is a faster response to production issues, with improved quality, reduced waste, faster time-to-market and increased profitability.

Powerful Distributed Client/Server Architecture

Collects, processes and distributes real-time data with unparalleled flexibility and scalability. The Proficy HMI/SCADA - iFIX architecture enables users to leverage multiple clients, including iClient TS - a solution that leverages Microsoft Terminal Server technology to seamlessly extend the reach of your HMI/SCADA applications.

Faster system development and deployment

The Intellution WorkSpace delivers point-and-click simplicity to application development. Through the use of powerful yet easy-to-use Wizards, Proficy HMI/SCADA - iFIX dramatically accelerates the development process. In addition, Intellution's Animation Experts drive internal third-party ActiveX control without VBA Programming.

Simplified application integration

Through Proficy HMI/SCADA - iFIX's patented Secure Containment technology, you can fully leverage third-party applications within the Proficy HMI/SCADA - iFIX environment... and do so without compromising your system's reliability.

Enhanced security and accountability

Proficy HMI/SCADA - iFIX boasts powerful new security and eSignature capabilities, designed to enable access restriction at a very granular level, as well as deliver a vehicle for capturing complete audit trail information - outstanding functionality for businesses in the regulated industries or for any company who simply wants to enhance security.

PID Control Theory Tutorial

The **P** stands for proportional control, **I** for integral control and **D** for derivative control. This is also what is called a three term controller.

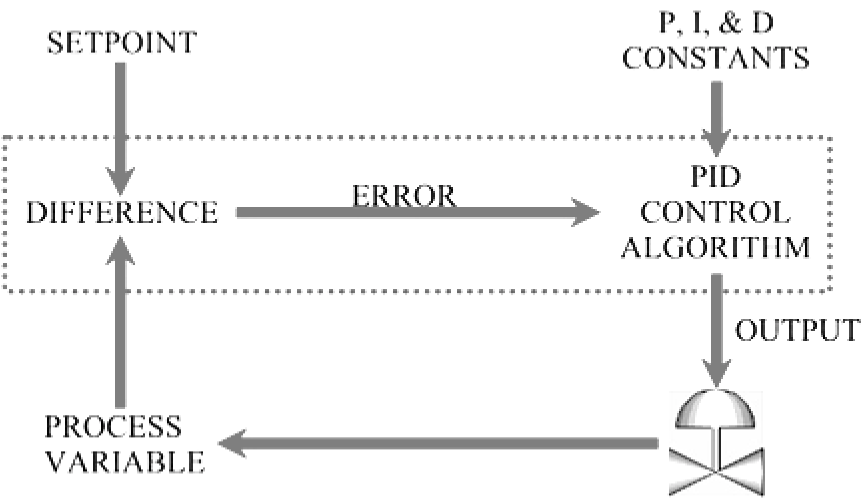
The basic function of a controller is to execute an algorithm (electronic controller) based on the control engineer's input (tuning constants), the operators desired operating value (setpoint) and the current plant process value. In most cases, the requirement is for the controller to act so that the process value is as close to the setpoint as possible. In a basic process control loop, the control engineer utilises the PID algorithms to achieve this.

The PID control algorithm is used for the control of almost all loops in the process industries, and is also the basis for many advanced control algorithms and strategies. In order for control loops to work properly, the PID loop must be properly tuned. Standard methods for tuning loops and criteria for judging the loop tuning have been used for many years, but should be reevaluated for use on modern digital control systems.

While the basic algorithm has been unchanged for many years and is used in all distributed control systems, the actual digital implementation of the algorithm has changed and differs from one system to another.

How a PID Controller Works

The PID controllers job is to maintain the output at a level so that there is no difference (error) between the process variable (PV) and the setpoint (SP).



In the diagram shown above the valve could be controlling the gas going to a heater, the chilling of a cooler, the pressure in a pipe, the flow through a pipe, the level in a tank, or any other process control system.

What the PID controller is looking at is the difference (or "error") between the PV and the SP. It looks at the absolute error and the rate of change of error. Absolute error means -- is there a big difference in the PV and SP or a little difference? Rate of change of error means -- is the difference between the PV or SP getting smaller or larger as time goes on.

When there is a "process upset", meaning, when the process variable or the setpoint quickly changes - the PID controller has to quickly change the output to get the process variable back equal to the setpoint. If you have a walk-in cooler with a PID controller and someone opens the door and walks in, the temperature (process variable) could rise very quickly. Therefore the PID controller has to increase the cooling (output) to compensate for this rise in temperature.

Once the PID controller has the process variable equal to the setpoint, a good PID controller will not vary the output. You want the output to be very steady (not changing). If the valve (motor, or other control element) are constantly changing, instead of maintaining a constant value, this could case more wear on the control element.

So there are these two contradictory goals. Fast response (fast change in output) when there is a "process upset", but slow response (steady output) when the PV is close to the setpoint.

Note that the output often goes past (over shoots) the steady-state output to get the process back to the setpoint. For example, a cooler may normally have it's cooling valve open 34% to maintain zero degrees (after the cooler has been closed up and the temperature settled down). If someone opens the cooler, walks in, walks around to find something, then walks back out, and then closes the cooler door -- the PID controller is freaking out because the temperature may have raised 20 degrees! So it may crank the cooling valve open to 50, 75, or even 100 percent -- to hurry up and cool the cooler back down -- before slowly closing the cooling valve back down to 34 percent.

****
For more details please see the following files at "Fieldbus Class room "

Proportional Integral Derivative PID Controls 67kB

****