PLC & SCADA FOR PROCESS AUTOMATION AND DATA AQUISITION

Shubhangi Sudhir Landge

[1] *Assistant Professor, Electrical Engineering Department, AISSMS'S, IOIT, Pune, Maharashtra, India*

ABSTRACT

Automation generally refers to the science and technology of process control and includes the control of chemical and petrochemical plants, Oil refineries, iron and steel plant, Power plants, cement mills, Paper pulp and paper mills water and waste water treatment plants and many like all this. The basic objective of automation is identifying the information flow and manipulates the material and energy flow as given process in a desired way.

PLC & SCADA plays an important role in atomizing industrial system. They work individually or together to satisfy the purpose. The acquisition of data, the processing of those data for use by the operator ,and the operator control of remote devices are the fundamental building blocks upon which all modern utility control systems are based. The systems to accomplish these functions are known as supervisory control and data acquisition (SCADA) systems.

Keyword: - *SCADA , PLC, Automation, Digital Signals, remote control.*

1. PROGRAMMABLE LOGIC CONTROLLERS

Automation of many different processes, such as controlling machines or factory assembly lines, is done through the use of small computers called as ***Programmable Logic Controller (PLC).*** This is actually a control device that consists of a programmable microprocessor, and is programmed using a specialized computer language.

Programmable logic controllers were invented as less expensive replacement for older automated systems that would use hundreds or thousands of relays and cam timers. PLC was first created to serve the automobile industry, and the first PLC project was developed in 1968 for General Motors. The programmable logic controller has made a significant contribution to factory automation. Earlier automation systems had to use Thousands of individual relays and cam timers, but all the relays and timers within a factory system can often be replaced with a single programmable logic controller.

Today, programmable logic controllers deliver a wide range of functionality, including basic relay control, motion control, process control, and complex networking, as well as being used in Supervisory Control and Data Acquisition Systems and Distributed Control Systems.

1.1 PLC System Block Diagram

A programmable controller is a specialized computer, hence it has all the basic component part contained in any other computer- a central processing unit, memory, input interfacing, and output interfacing. A typical programmable controller block diagram is shown below.

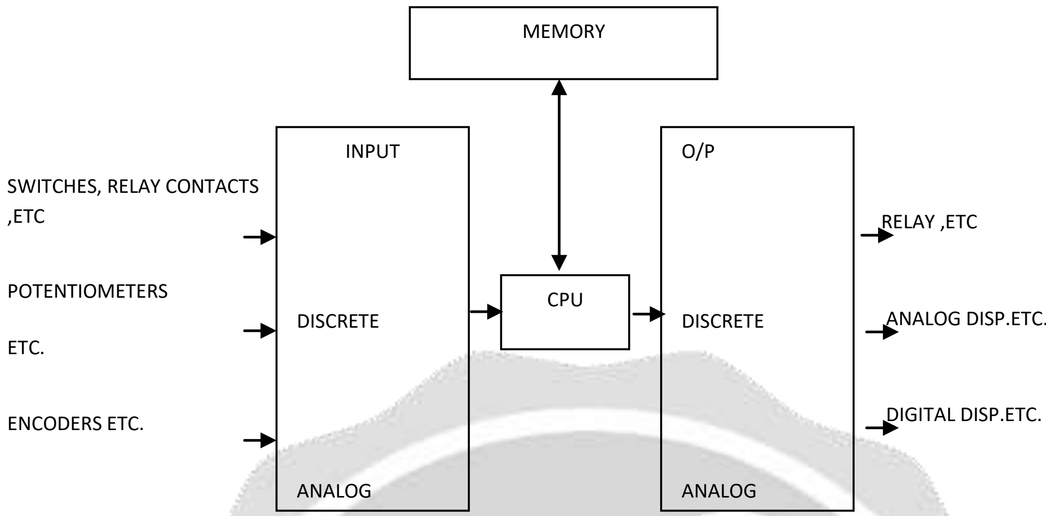


Fig -1 PLC Block Diagram

The Central Processing unit is the control portion of the PLC. It interprets the program commands retrieved from memory and acts on those commands, in present days PLC the unit is microprocessor based systems.

Memory in the system generally consists of ROM and RAM. RAM memory is backed-up through an on-board battery. This battery can be a standard dry cell or rechargeable nickel-cadmium type. Newer PLC units are now available with EEPROM, which does not require battery.

PLC contains a variable number of Input/ Output ports. Digital or discrete signals behave as switches, yielding simply an ON or OFF signal pushbuttons, limit switches, and photo-eyes are examples of devices providing a digital signal. Digital signals are judges using either voltage or current. A PLC might use 24 VDC I/O, with values above 22VDC representing ON and values below 2 VDC representing off. Analog signals are like volume controls, with a range of values between zero and full-scale. Pressure, temperature and weight are often analog signals. Analog signals can again be either be voltage or current. Different versions of PLC are available depending on the number of I/P and O/P e.g. 64 point GE Fanuc Versamax micro PLC which means it has 40 Digital inputs and 24 Digital output. Output can be a transistorized or potential free i.e. relay o/p.

1.2 PLC Programming

Previously programmable logos controller were programmed in ladder logic, which is similar to a schematic of relay logic. Modern programmable logic controller is usually programmed in any on of several languages, ranging from ladder logic to Basic or C. Typically the program is written in a development environment on a personal computer, and then is downloaded onto the programmable logic controller directly through a cable connection.

The International standard IEC 61131-3 has defined 5 programming languages for programmable control systems:

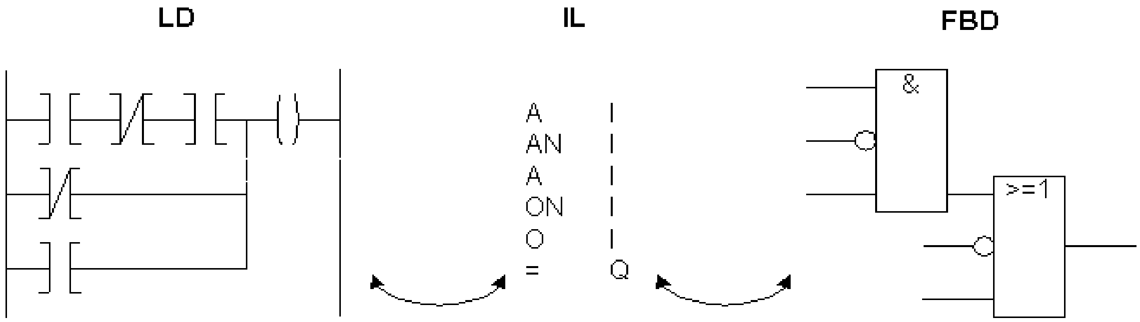
FBD (Function Block Diagram),

LD (Ladder Diagram),

ST (Structured Text, Pascal type language.)

IL (Instruction List)

SFC (Sequential function chart)



1.3 PLC Operation & Working Principle

A PLC works by continually scanning a program. The scan cycle consists of three important steps. There are typically more than three but we focus on the important parts.

- Check Input Status: First the PLC takes a look at each input to determine if it is on or off. In other words, is the sensor connected to the first input on? How about the second input? And so on it checks all the connected inputs. It records this data into its memory to be used during the next step.
- Execute Program: Next the PLC executes your program, one instruction at a time.
- Update output Status: Finally the PLC updates the status of the outputs based on which inputs were on during the first step and the results of executing your program during the second step.

2. SUPERVISORY CONTROL AND DATA ACQUISITION

SCADA- Supervisory Control and Data Acquisition can mean many things to many people. The acquisition of data, the processing of those data for use by the operator, and operator control of remote devices are the fundamental building blocks upon which all modern utility control systems are based. The system to accomplish these functions is known as SCADA system. 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, or other commercial hardware modules.

Patents for remote control and remote indication were filed as early as 1890s. These very early systems were intended for either remote control or remote indication but not both. In the 1920 and 1930s various commercial systems evolved employing concepts of check-before-operate and with the capability of conveying status of multiple points. These early systems were based on electromechanical logic which largely evolved from telephone system technology.

The advent of minicomputer in the 1960s prompted dramatic changes in the design and use of supervisory control systems. Electromechanical systems which preceded the introduction of computer based systems were largely intended for remote control and simple indication of status. The acquisition of large numbers of status indications and analog values was not practical. The early systems were generally referred to as simply “Supervisory Control”. In the late 1960s, as new minicomputer-based systems began to emerge, the possibilities for vastly increasing data acquisition became apparent, and the expression “Supervisory Control and Data Acquisition” or SCADA came into being as more appropriate description of the system.

2.1 Basic SCADA Elements

There are four major elements to a SCADA system, master terminal unit (MTU), communications, and remote terminal unit (RTU). The operator exercises control through information that is depicted on a video display unit

(VDU). Input to the system normally initiates from the operator via the master terminal unit's keyboard. The MTU monitors information from remote sites and displays information for the operator. The relationship between MTU and RTU is analogous to master and slave. SCADA systems are capable of communicating using a wide variety of media such as fiber optics, dial-up, or dedicated voice grade telephone lines, or radio.

Master Terminal Unit

At the heart of the system is the master terminal unit (MTU). The master terminal unit initiates all communications, gathers data, stores information, sends information to other systems, and interfaces with operators. The major difference between the MTU and RTU is that the MTU initiates virtually all communications by its programming and people. Almost all communication is initiated by the MTU. The MTU also communicates with other peripheral devices in the facility like monitors, printers or other information systems. The primary interface to the operator is the monitor that portrays a representation of valves, pumps, etc. As incoming data changes the screen is updated. Figure shows the examples of inputs from the MTU and field devices.

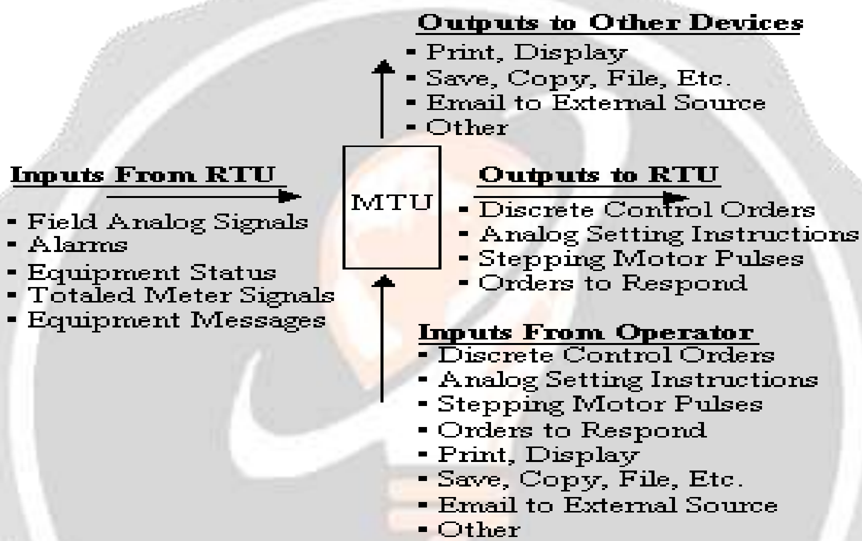


Fig -2: Inputs & Outputs for MTU

Remote Terminal Unit

Remote terminal units gather information from their remote site from various input devices, like valves, pumps, alarms, sensors, meters, etc. Essentially, data is either analog (real numbers), digital (on/off),or pulse data(e.g., counting revolutions of meters). Many remote terminal units hold the information gathered in their memory and wait for a request from the MTU to transmit the data. Other more sophisticated remote terminal units have microcomputers and PLC that perform direct control over a remote site without the direction of the MTU. The RTU central processing unit receives a binary data stream in accordance with the communication protocol. Protocols can be open, like Transmission Control Protocol and Internet Protocol (TCP/IP) or proprietary. Figure shows an example of outputs of the RTU to MTU and field devices.

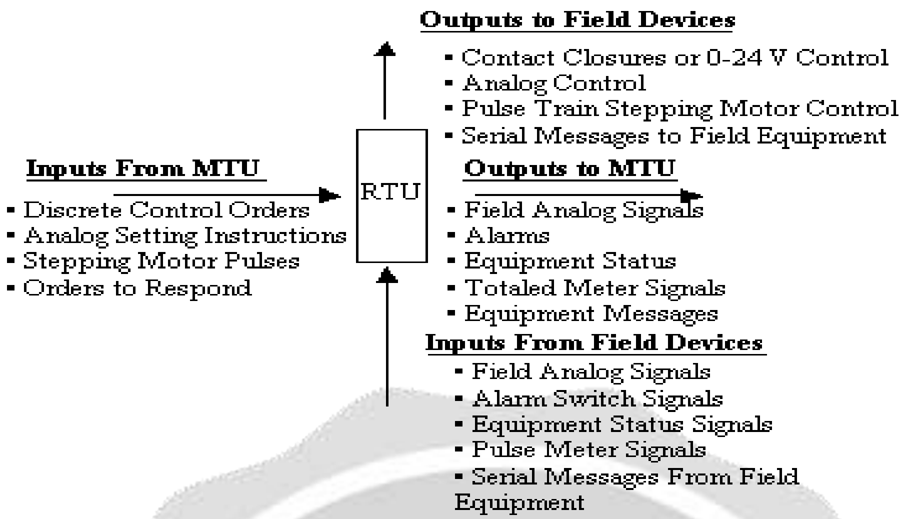


Fig -3: Inputs & Outputs for RTU

2.2 Need of SCADA System

Some basic needs are listed below

- **Save Time:** The time taken to travel remote sites to gather information or issue controls, to sift through manually entered data, write out a report or perform any of the functions that a SCADA system does as a matter of course is considerable. Time saving benefits go far beyond the man hours saved- timeliness of alarms, actions, control have high monetary value as well.
- **Avoid Trouble:** A SCADA system's primary purpose is to give advance warning of trouble. Hence action can be taken before it swallows the whole system and creates problem.
- **Achieve System wide Processes:**
  SCADA systems afford the user an opportunity to monitor and control processes that take place over a wide geographical area. E.g. A sewage treatment plant or energy management system. These applications can cause considerable financial savings in running cost and capital cost which can pay for the cost of SCADA in itself.
- **Save Manpower:** Before SCADA systems were implemented, remote sites such as substations and pumping stations were either manned or inspected frequently. The need for this was eliminated or greatly reduced by the implementation of wide area SCADA. This was the primary economic driver for SCADA system implementation in the first great wave of comprehensive systems in the 1970's and 1980's.

2.3 SCADA Functions

- **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.

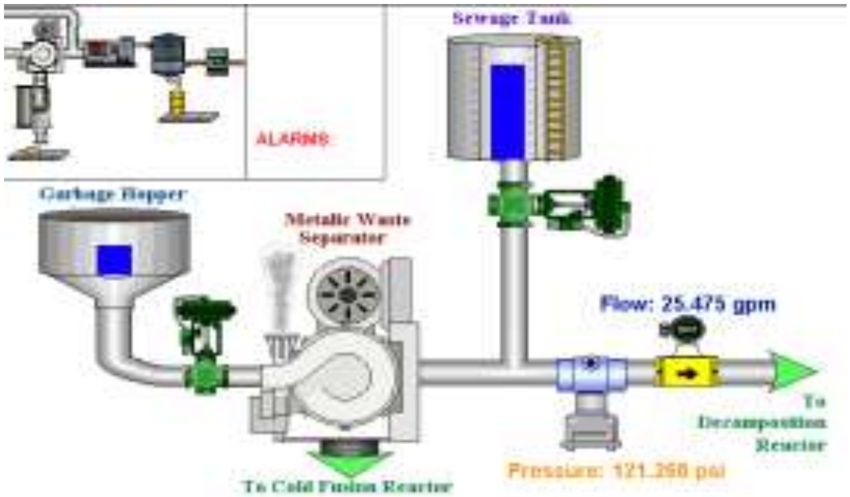


Fig -4 SCADA Functions.

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.

2.4 Connections & Protocols

As mentioned earlier that PLC is used frequently in a SCADA system as the RTU. We need to through some light on the communication between the SCADA Computer and the PLC

There are several different types of connections between the SCADA computers and PLCs. For example, RS232, RS422, RS485, and Ethernet. Note that these are only the electrical connection and do not specify the protocol or language used on the connection. We can call this as the Hardware protocol.

The classical example is an Indian trying to talk to someone in France. There are several ways that you can connect to someone in France (telephone, e-mail, cell phone, fax machine, television, etc.) and these connections are the equivalent of RS232, RS422, RS485, and Ethernet. But just because you can call (connect with) someone in France does not mean that you can communicate. If the Indian does not speak French or the French person does not speak English they will not communicate. The language that PLCs speak is more commonly know as a "protocol". So it takes both -- a connection and a protocol.

Note that just because several PLCs all have Ethernet connections do not mean that they can talk to each other because they speak different languages (protocols)! It would be like saying you could talk to anyone in the world just because they have a telephone (although you do not speak their language). Likewise, PLCs with a RS232, RS422, or RS485 port cannot talk to each other unless they use the same protocol.

Each PLC manufacturer has their own protocol or language that the PLC speaks. For example, Allen-Bradley (Rockwell) PLCs talk DF1, Data Highway 485 (DH485), Data Highway (DH), Data Highway + (DH+), Remote I/O (RIO), Devicenet, Controlnet, and a few others. Talk about an identity complex! Modicon PLCs speak Modbus, Modbus Plus, and now Modbus TCP/IP (Modbus over Ethernet). Automation Direct PLCs speak Directed and K-sequence protocols while some of their PLCs still communicate using their old protocols. The old GE PLC protocol was CCM but they now use SNP and variants such as SNPX. Siemens PLCs work mainly with the Profibus protocol which is considered more of an open, rather than proprietary, protocol.

The most common "standard" protocols we see are DF1, Modbus, and Profibus. DF1 and Modbus are popular since they have been around longer than many of the people using them. Profibus is popular because, in our humble opinion, combines the best of simplicity (daisy chain, twisted pair), industrial ruggedness (RS485), speed (12

MBaud), adaptability (you can select from many different speeds), scalability (from simple COM port up to communications controllers), and function (most automation devices can talk Profibus). Typically all PLC manufacturers will have a PLC module that will talk these protocols.

Now that Ethernet is becoming the "most hyped" capability in automation there are several protocols over Ethernet. But remember -- just because an Allen-Bradley and a Siemens PLCs are both put on the same Ethernet cable -- they cannot talk to each other because they do not use the same protocol.

3 SCADA CONFIGURATIONS

Configuration in SCADA should be done sequentially only. This is having following sequence.

1. Protocol Configuration
2. Port Configuration.
3. Device Configuration
4. Tag Configuration
5. Role Configuration

4 APPLICATION OF SCADA

- Electric Utility
- Water / Waste Water Utilities
- Oil & Gas Transmission & Distribution
- Communication Networks
- Industrial Process Control

5. CONCLUSIONS

Automation is the need of today. In order to compete in the global market improved product quality along with reduced cycle time is required. This is achieved through automation by using PLC & SCADA. The paper deals with the basics of both PLC & SCADA, right from their history to their selection. The paper is written in such a manner that even a person without the knowledge of PLC & SCADA can understand it. The systems are considered based on the current knowledge of immerging technology.

6. REFERENCES

[1]. Miroslave Kover Dorco, "SCADA System creation by using Java application and PLC", Control Conference (ICCC) 2014, IEEE conference publication 2014, 264-267.

[2]. Mohamd Endi, Y Z Elhalwagy, Attalla Hashad, "Three layer PLC/SCADA system architecture in process automation and data acquisition", the second International conference on Computer and Automation Engineering (ICCAE), 2010, Vol.2, 774-779.

[3]. Lester Abbey ,"Evolution of SCADA systems", SCADA conference 2003, 75-80.

[4]. Duong Trung ,"The Design of next Generation SCADA systems", IEEE Transactions 1995, 120-125.

[5]. Samul C. Scacca, "Advanced SCADA Concept", IEEEE Transaction, 1995, 95-99.