International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622
National Conference on Emerging Trends in Engineering & Technology (VNCET-30 Mar'12)

DEVELOPING THE RELATIONSHIP BETWEEN DCS & PLC USING MODBUS TECHNOLOGY

Mr.Prafulla Patil Mr.Vishal Pande Mr.Vaibhav Sarfare Mr.Swaroop Hangal*
prafullavpatil@gmail.com vishalp_2000@yahoo.com vaibhav.inst09@gmail.com

Vidyavardhini's College of Engineering & Technology
Department of Instrumentation
K.T. Marg , Vasai (W) , Dist : Thane, Maharashtra-401201
***Research Scholar, Department of Aerospace Engineering, IIT-B, Powai, Mumbai, 400076**

Abstract : -

This paper explores the changing roles of traditional distributed control systems (DCS) and programmable logic controllers (PLC) used to automate manufacturing processes. The two technologies initially served two different control requirements. However, improvements in microprocessor-based controllers created conditions for two technologies to merge. The shift toward commercial, off-the-shelf automation technology, software-based control verses hard control and use of non-proprietary networks has created a new class of systems called hybrid process automation systems.

In addition, the role of the plant control system has been expanded from just process visualization and control to include process optimization, plant asset management, energy management, and inventory control.

Control system vendors have adopted unique terms to differentiate their systems from the traditional definition of a DCS or PLC. Some have expanded their process control systems to offer vertical integration with IT systems and horizontal integration with intelligent drives, motors, process instrumentation and discrete control technologies. This paper addresses detail communication setup of DCS HC900 C30 (Master) and PLC AB1200C (Slave) using MODBUS, as a result of these developments and identifies technologies that offer the best solution to meet the challenges faced by Process Industry.

Index Terms: -

DCS, PLC, Master, Slave, MODBUS, Hybrid Control.

1. Introduction :-

Modern Trends in manufacturing, such as mass customization, lot size one, company spanning supply chains, and supplier networks, demand effective production management and control systems, including efficient product data management and data applications in control. Conventional centralized approaches run into difficulties when it comes to adapting to these new requirements where flexibility and Scalability of the system are often more important than finding the global optimum of the production. Centralized control systems are efficient when the physical layout of the plant as well as the product portfolio produced are static and will not be changed during the life cycle of the plant. They give a direct control to the system operator and a view of the whole plant which serves as a basis for scheduling.

The growing size and complexity of the manufacturing processes have placed new demands on automation system vendors. By understanding these requirements and the control and automation technologies available today, we can effectively select the best platform for the automation of applications in the process industry.

2. Control Strategies :-

There are two strategies to implement overall control in process plants. The distributed approach uses individual PLCs to control auxiliary tasks. These auxiliary systems are usually controlled through dedicated PLCs supplied with mechanical equipment. These auxiliary systems are, in turn, controlled by the plant's main control system that is typically a DCS type system.

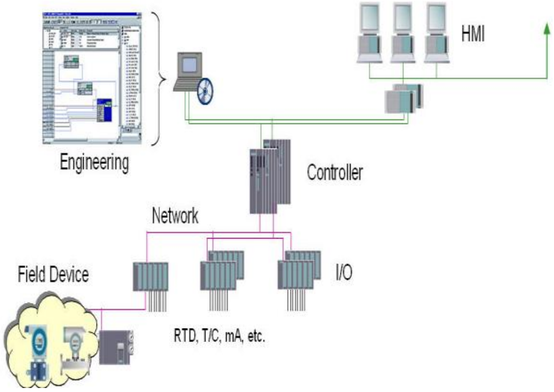
The main plant control system interfaces to these individual PLCs via network interfaces or hardwiring and provides the overall command and process control environment. This control approach is used when the plant control system cannot meet specific auxiliary applications requirements. This approach is also partly driven by the need of equipment suppliers to guarantee performance. They must provide control systems with their equipment so that the control logic functions are properly implemented to ensure correct operation and control of the equipment. The suppliers usually develop their solutions based on the product capabilities of a single automation vendor. It is cost effective for the process OEM to use a single PLC vendor, because it does not require them to develop, maintain, and support their programs on multiple PLC platforms.

The main disadvantage with this approach is that there is always a level of ambiguity between the auxiliary systems and the main plant control. One In the centralized control approach, the control system must meet the overall plant control requirements, such as plant asset management, energy management and advanced process control, and interface with CMMS and ERP systems. The system must also be able to meet the process control requirements of the analog control and be able to execute portions of the logic in a faster cycle to meet the speed control requirements of the sub controls. The system must be able to support various types of programming languages including function blocks, structured programming languages, and ladder logic. The systems that have these capabilities combine the features of both DCS and PLC, sometimes called hybrid control systems. In the central approach, the main plant control system provides the overall logic for the main process and the auxiliary systems. The auxiliary systems have remote I/Os that are connected to the main plant control system; the logic for the control of the auxiliary system resides in the central controller.

2.1 DCS Overview : -

DCS evolved within the process control industry. DCS are known for regulatory control capability and Redundancy. The DCS architectures were centrally orientated with main control room as a meeting point for long cables for the input and output. They were based on proprietary components - operating systems, networks, hardware, and configuration tools with all the hardware located centrally in a controlled environment. Traditional DCS were not designed with open interfaces thus restricting the communication within the system boundaries. Communication interfaces with other vendors equipment requires expensive and time consuming hardware and software development process.

disadvantage is that maintenance and control personnel treat these systems as black boxes, since only the equipment vendor knows the why and how of the logic inside the PLC, or if the control is based on a dedicated controller. In most cases the tools and software used to implement these systems are very different from main plant control software, requiring the maintenance people to learn several different programming packages from different vendors. Managing the different programs and versions of software packages and keeping them updated over the life cycle of the plant is Very costly.



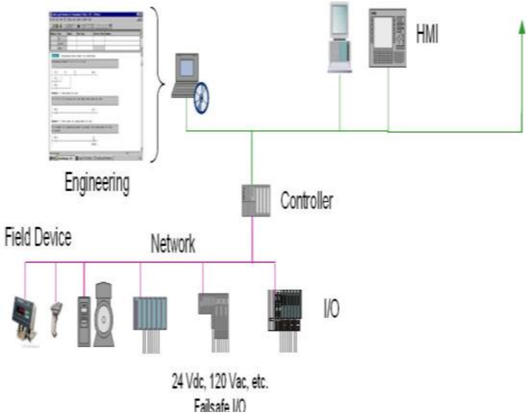
The earlier days of process control was mainly isolated to a certain number of loops or PID controllers (Proportional Integral and Derivative) to control temperature, pressure, steam, and flow. As the electronics technology developed and the PLC-based systems became pervasive, it becomes possible to program a number of PID loops in a single controller in a rack away from the field. The PID controllers that were initially made of dedicated electronic hardware in the earlier DCS systems can now be implemented in a cost-effective way in a single, microprocessor-based controller. Some of the field bus based technologies, such as Foundation Field bus, still support the concept of "Control in the Field," Where a single, independent control loop can be implemented in the field away from main controller.

The programming environment for the traditional DCS was based on function blocks commonly understood in the process control industry. There is not a big need to program the complicated logic operations that required high speed program execution, since most of the temperature and pressure loop's response time is in 100ms or higher. In process plants, motors and drives were programmed and controlled via the PLC.

As the microprocessor became faster and cheaper, it became possible to implement some of the PID loops in the software, since software-based controllers were able to meet the input to output cycle time requirements of the overall process control loop. High speed control loops, such as those used in metal processing, still use dedicated control hardware due to performance reasons, but most of the process loops in a process plant can be programmed in a single controller/PLC using software.

2.2 PLC Overview : -

Programmable logic controllers initially emerged as a replacement of the relay control systems. Before PLC, control, sequencing, and safety interlock logic for manufacturing processes was accomplished using hundreds of relays, cam timers, drum sequencers, and dedicated closed-loop controllers. The process for updating such facilities to make changes to the process was very time consuming and expensive, as the relay systems needed to be rewired by skilled electricians. PLCs eliminated the need to physically rewire the relay systems as it can now be done by modifying the software. Hence software in the PLC became the brain of the manufacturing process.



Typical PLC Architecture

The basic concepts of PLC programming are common to all manufacturers. Differences in I/O addressing, memory organization, and instruction sets mean that PLC programs are never perfectly interchangeable between different vendors. Even within the same product line of a single manufacturer, different models offer different levels of programming capability, resulting in programs being non-interchangeable.

The IEC 61131-3 specification is an effort to standardize the PLC programming languages and define common instruction sets and programming languages. IEC 61131-3 currently defines five programming languages for programmable control systems: FBD (function block diagram), LAD (ladder diagram), SCL (structured text, similar to the Pascal programming language), STL (instruction list, similar to assembly language), and SFC (sequential function chart). These techniques emphasize logical organization of operations with a program.

In PLC-based systems, logic control and programming works in the same ways as relay logic. PLCs are programmed in ladder logic, which strongly resembles a schematic diagram of relay logic. Modern PLCs can be programmed in a variety of ways, from ladder logic, to more traditional structured programming languages, C and function blocks. It is no wonder that ladder logic is the preferable form of programming language for electricians and technicians responsible for the maintenance of manufacturing facilities. Since not much change in the thinking process is required to program ladder logic, it was the natural first step for the electricians who were familiar and had experience working with relays.

The elements used in PLC programming, such as inputs, are called contacts. Outputs are called coils, as in relay coil. This kind of language is found in almost all the PLC programming software. Programs are divided into multiple rungs of ladder. Each rung of ladder language typically has one coil at the far right. Some manufacturers may allow more than one output coil on a rung.--( )-- a regular coil, true when its rung is true or normally open relay contact close

--(\)-- a "not" coil, false when its rung is true or normally close relay contact opens

--[ ]-- A regular contact, true when its coil is true (normally false)

--[\]-- A "not" contact, false when its coil is true (normally true)

The coil may represent a physical output that operates a device, such as a motor contactor, connected to the PLC output. It may also represent an internal storage bit to use elsewhere in the program. The limitation with the PLC way of thinking is that it gives a programmer a very limited view of the overall plant sequence control. It also makes the logic very complicated to troubleshoot if a large number of contacts and coils are used. This is also known as the bottoms up approach to programming. It is very easy to look at the individual pieces of the equipment and their interlocks. However, it is very difficult to understand the complete plant control logic, because the ladder logic quickly becomes very complicated as the number of devices and their interactions increase.

Ladder logic does not provide the kind of interface that a process control engineer can easily understand, because his view of the plant is in the form of groups, routes, and individual objects such as motors, PID, and diverters.

Most technicians and maintenance people do not understand why a particular piece of logic was implemented. They will, however, use software and hardware jumpers to bypass the logic. This can damage the equipment and pose serious safety risks to workers. Some equipment control programs for equipment (circular stackers and grate hydraulic control) are very complicated and require mathematical computations. These calculations must be performed in the specific amount of time (10ms or less). Such stringent execution requires controllers with high speed computation capability. Once these sub-controls are programmed, their logic rarely needs to be modified or changed; only modification in the process parameters is required.

DCS PLC Communication : -

From above Introduction it's clear that both PLC& DCS are designed to meet two different application requirement,but attributes of both the systems are required to provide effective control of the various systems in process industries, Therefore for the optimal result its required to establish the communication between DCS & PLC.

3.1 DCS MODBUS Communication : -

This implementation is designed to provide a popular data exchange format connecting the HC900 to both Honeywell and foreign master devices via the RS232 and RS485 communication ports. The Modbus RTU allows the instrument to be a citizen on a data link shared with other devices.The Modbus RTU specification is respected in the physical and data link layers. The message structure of the Modbus RTU function codes is employed and standard IEEE 32-bit floating point and integer formats are used. Data register mapping is unique to the HC900 and other Honeywell instruments.

Modbus RTU Link Layer

The link layer includes the following properties/behaviors:

- Slave address recognition,
- Start / End of Frame detection,
- CRC-16 generation / checking,
- Transmit / receive message time-out,
- Buffer overflow detection,
- Framing error detection,
- Idle line detection.

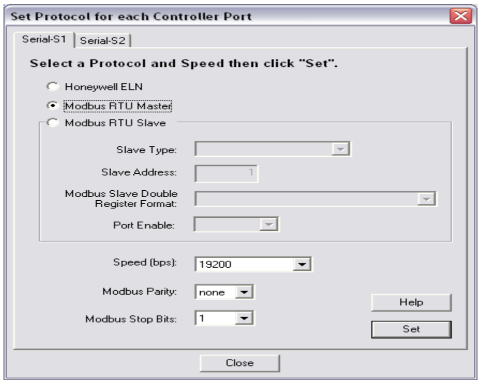
Errors detected by the physical layer in messages received by the slave are ignored and the physical layer
automatically restarts by initiating a new receive on the next idle line detection.

Modbus RTU Data Layer

The data layer includes:

- Diagnostic loopback,
- Function code recognition / rejection,
- Data error code generation
- Busy / repoll,

Errors detected by the data layer are rejected and the slave responds to the polling device with a Modbustype status exception error. Modbus RTU Exception Codes.

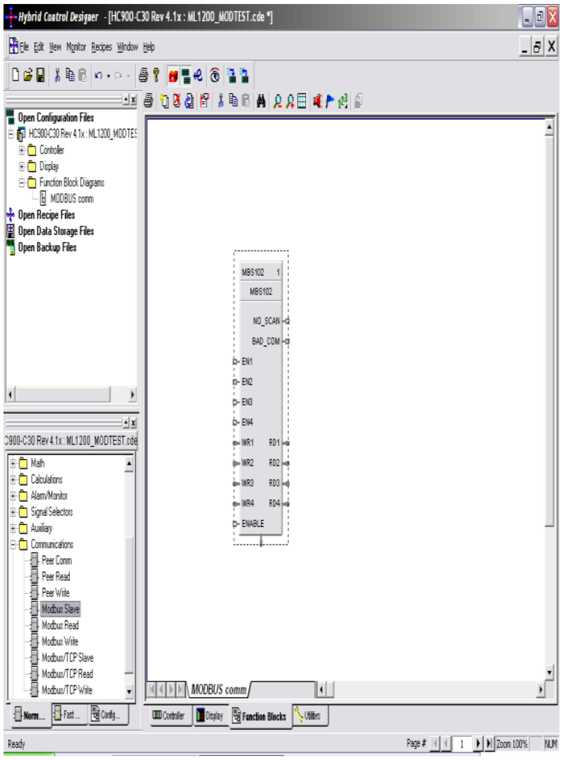


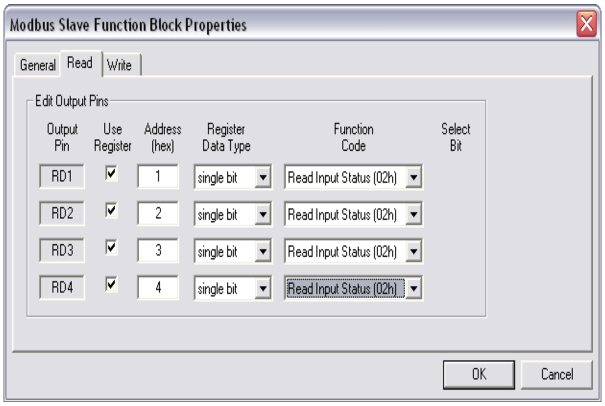
Select a Protocol and Speed then click "Set".

○ Honeywell ELN
● Modbus RTU Master
○ Modbus RTU Slave

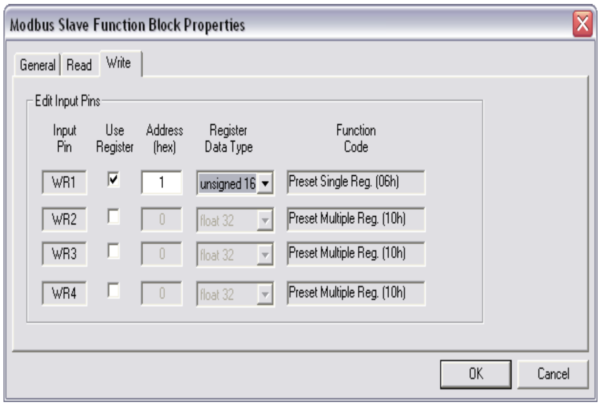
Slave Type:
Slave Address: 1
Modbus Slave Double Register Format:
Port Enable:

Speed (bps): 19200
Modbus Parity: none
Modbus Stop Bits: 1

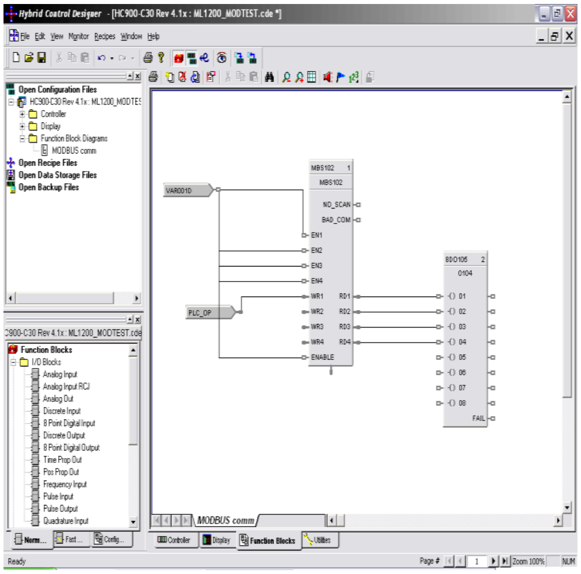


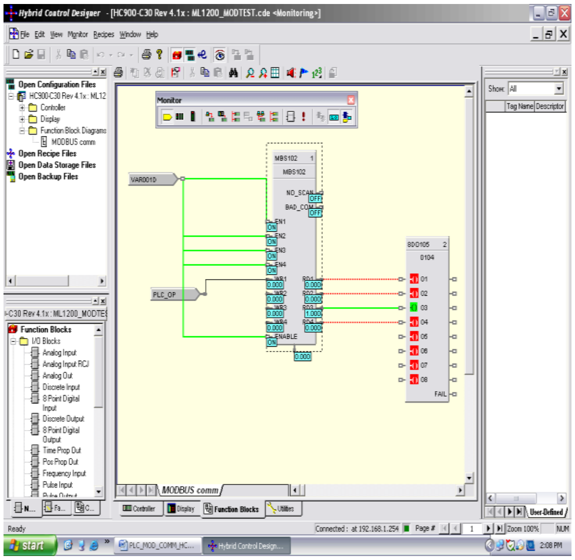


Output Pin	Use Register	Address (hex)	Register Data Type	Function Code	Select Bit
RD1	☑	1	single bit	Read Input Status (02h)	
RD2	☑	2	single bit	Read Input Status (02h)	
RD3	☑	3	single bit	Read Input Status (02h)	
RD4	☑	4	single bit	Read Input Status (02h)	



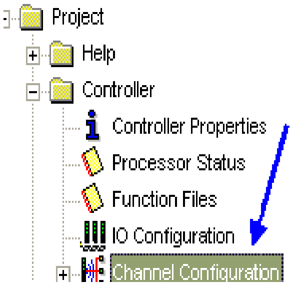
Input Pin	Use Register	Address (hex)	Register Data Type	Function Code
WR1	☑	1	unsigned 16	Preset Single Reg. (06h)
WR2	☐	0	float 32	Preset Multiple Reg. (10h)
WR3	☐	0	float 32	Preset Multiple Reg. (10h)
WR4	☐	0	float 32	Preset Multiple Reg. (10h)



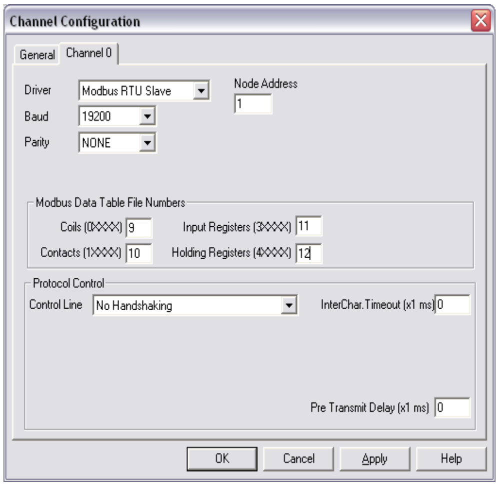


3.2PLC MODBUS Communication : -

ABµlogix1200 PLC is one of many devices on the market today that utilize the Modbus RTU protocol. It is important to keep in mind that this paper and the steps that follow focus specifically on the RUI/Gateway as it is used with the AB MicroLogix 1200 PLC. It should also be noted that it is assumed by the writer that the reader has an understanding of RSLogix 500 software and its embedded tools to create projects, files, etc…for these steps are intentionally omitted to minimize the size of this paper. The following screen shots capture some of the more significant steps to successfully establish communications between the PLC (channel 0) and the Gateway running Modbus RTU. Using RSLogix 500, open up a project or create a new one. Once opened or created, proceed to double click on "Channel Configuration" as shown below.



Click on the tab identified as "Channel 0" and set it as shown. The configuration shown below reflects defaults. Change the parameters to meet your network requirements or if acceptable as is simply click OK to finish.



Driver: Modbus RTU Slave
Node Address: 1
Baud: 19200
Parity: NONE

Modbus Data Table File Numbers
Coils (0XXXX): 9
Input Registers (3XXXX): 11
Contacts (1XXXX): 10
Holding Registers (4XXXX): 12

Protocol Control
Control Line: No Handshaking
InterChar.Timeout (x1 ms): 0
Pre Transmit Delay (x1 ms): 0

4. Conclusion :

The paper has presented the authors experience with DCS HC9WC30 & Allen Bradley µlogix1200 PLC MODBUS Communication.

A variety of MODBUS communication setting has been discussed in this paper. Various diagrams and communication settings is valuable information for understanding most common issues of MODBUS communication.

Future work will concentrate on improving communication between DCS & PLC and making it more illustrative.Also under development is the idea of setting up a communication between more than one PLC & DCS.

Ultimate aim of this paper is to provide knowledge PLC DCS communication to final year students of Engineering.

5. References:

[1]. Distributed Automation: PABADIS versus HMS (2005) IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS

[2]. Cementing the relationship between DCS and PLC: A review of emerging trends in plant control systems; Moin Shaikh ,Siemens Energy & Automation, Inc.

[3]. N. Duffie and R. Piper, "Nonhierarchical control of manufacturing systems," J. Manuf. Syst., vol. 5, no. 2, 1986.

[4]. R. A. Brooks, "Robust layered control systems for a mobile robot," IEEE J. Robot. Automat., vol. RA-2, no. 1, pp. 14–23, 1986.

[5]. A. Lüder, J. Peschke, S. Deter, and A. Bratoukhine, "Interaction of software agents and real-time industrial control system within a PABADIS system," in Proc. 14th Euromicro Int. Conf. Real-Time Systems, Work in Progress Session, Vienna, Austria, Jun. 2002, pp. 21–25.

[6].Manual for Hybrid Controller HC900 C30 by Honeywell.

[7] Manual for µLogix 1200PLC by Allen Bradely.