PC-PLC BASED VACUUM CONTROL SYSTEM FOR SUPERCONDUCTING CYLOTRON AT VECC

Tapas Samanta*, Dr. D. Sarkar[@], Mr. S. Dasgupta[$],
VECC, 1/AF, Bidhan Nagar, Kolkata, India
* tsamanta@veccal.ernet.in [@] dsarkar@veccal.ernet.in [$] sdg@veccal.ernet.in

Abstract

The superconducting cyclotron's vacuum control system has been developed at Variable Energy Cyclotron Center (VECC) indigenously. The s7 series of siemens plc system controls the whole process. A pc serves as the interface to the system for monitoring and control. The pc communicates with the plc system via serial port to the communication controller at plc side. There are two distinct part of the pc software. The first one is the vacuum server for the real-time communication with the plc using 3964r protocol and a TCP/IP server to communicate with the second part, the vacuum client. The vacuum client is the control and monitoring interface for actual operation. It communicates with the vacuum server. The client gets master control depending upon IP address & the level of user authentication. The complete system starting from the fabrication of the Motor control Center (MCC) to the design of the monitoring & control software for pc and the user software for the plc have been developed in house. The heart of the system is plc-pc real-time communication protocol 3964r that needed to be implemented in c/c++ on real-time platform. The windows platform gives a very good graphical environment for human interface at the same time imposes lots of difficulties in implementing the real-time features. However, the protocol has been implemented with a few precautionary measures that windows platform ensures. Therefore, only a soft-real-time system could be implemented. The experience is quite interesting, specially developing the plc-pc real-time communication protocol under windows environment.

Introduction

PLC Overview

Programmable **L**ogic **C**ontrollers (PLCs) are diskless compact computers including all the necessary software and hardware interfaces to the process. They are generally used for automation control application either standalone or connected to distributed inputs/outputs, to other PLCs and/or to supervision PCs. The connections are established by means of field buses such as PROFIBUS or Ethernet.

A typical PLC consists of

- A power supply;
- A CPU where the user program runs;
- Input/output modules;
- Optional communication modules.

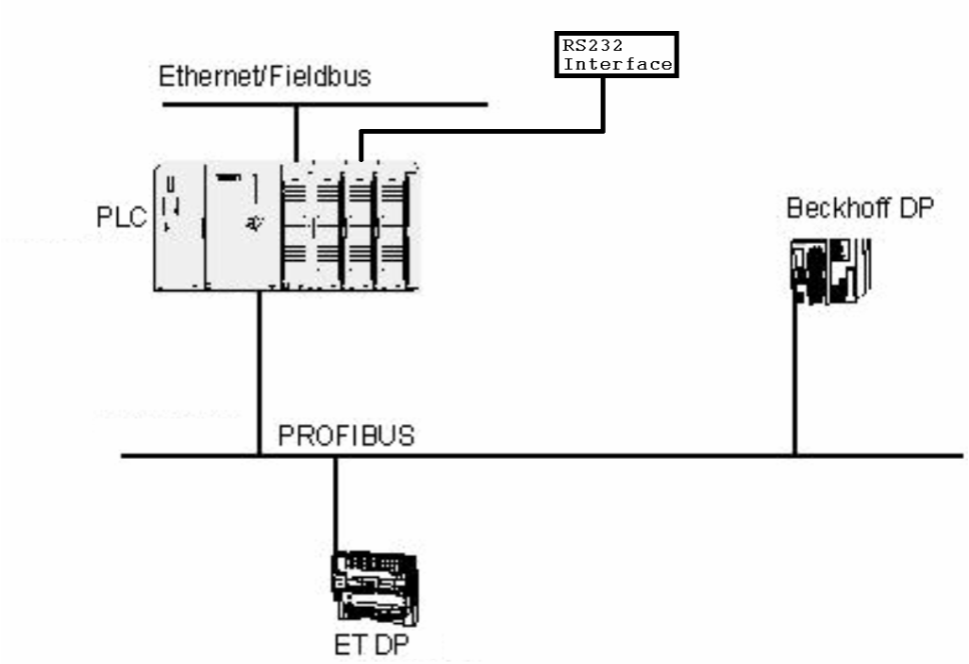

Figure.1 Typical PLC BUS communication

The available I/O modules support a wide range of electrical interfaces

- Analog module (+/- 10V, +/- 1V, 4-20mA, resistor, etc.)
- Temperature measurement (pt100, Ni 100, etc).
- Digital module (+- 24V, 220V, etc.)
- TTL module (Beck Hoff I/O module, etc.)
- RS 232 module
- Ethernet Modules

PLC-based solutions are well adapted to two-level control architectures where the front-end layer has to be autonomous and independent from the supervision layer. The process control does not depend on the network neither on a remote computer; it is more secure.

Project Overview

The Vacuum Control System consists of a PLC with MCC (Motor Control Centre) and a Supervisory PC connected in between the PLC & the control LAN (Local Area Network). PLC take cares the process control part with instruction from the supervisory PC. The Supervisory PC gets the commands from remote Clients over the LAN. PLC communicates with supervisory PC over a real-time protocol 3964R, which is basically a proprietary protocol of Siemens. The Protocol has been implemented on Windows platform. This implementation was the main hurdle of the Software project. Only a soft-real-time version of the protocol could be implemented.

Hardware Configuration

The hardware configuration is shown in the figure 2. The PLC is connected to the Process i.e. Vacuum System via Motor Control Centre (MCC).

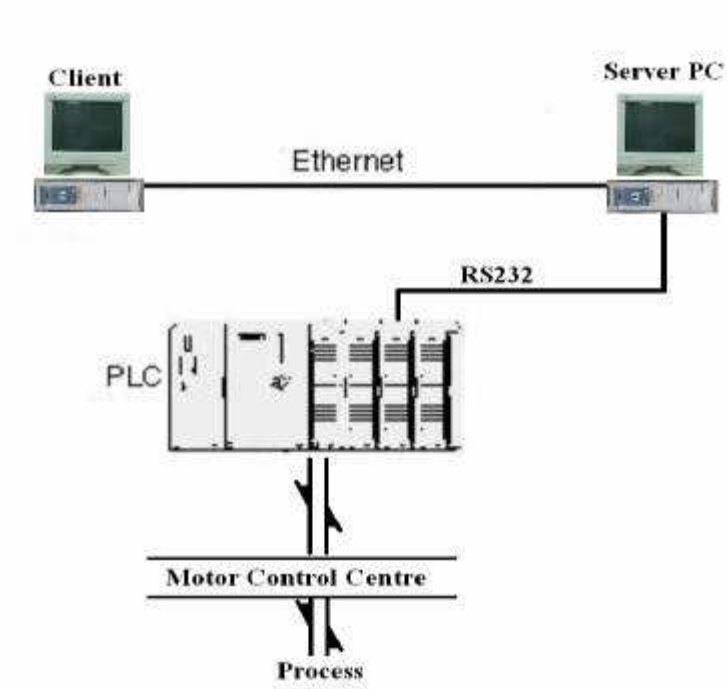

Figure 2. The Hardware Interface

The MCC hosts all power contactors, relays and related components. PLC is connected to the MCC via I/O modules. The hardware interface between PC and PLC is through RS232 interface. On PLC side a communication module is used for this and on the PC side it is standard COM port. The user interface with the server PC is through a remote PC over LAN on standard 10/100 BS.

Software Configuration

The project is divided into two basic parts.

1. The PLC user programming
2. The PC based supervisory Control.

The PLC user programming takes care of the process control part (closed loop control) and the data/command communication part with the PC through a communication module.

The PC based supervisory control has been developed in two parts.

1. Soft-real-time communication with PLC and Server taking care of command and control part.
2. The Vacuum Client part taking care of the Human Interface with the command and control of the Process.

Part one run with real-time-critical priority over a real-time priority class thread on Windows Platform.

Performance

The system has been tested in the Laboratory. The critical response time was 150-200ms when the Client runs on the same Server machine. However, the performance degrades to 0.5 to 1 Sec, depending on the network traffic over the LAN.

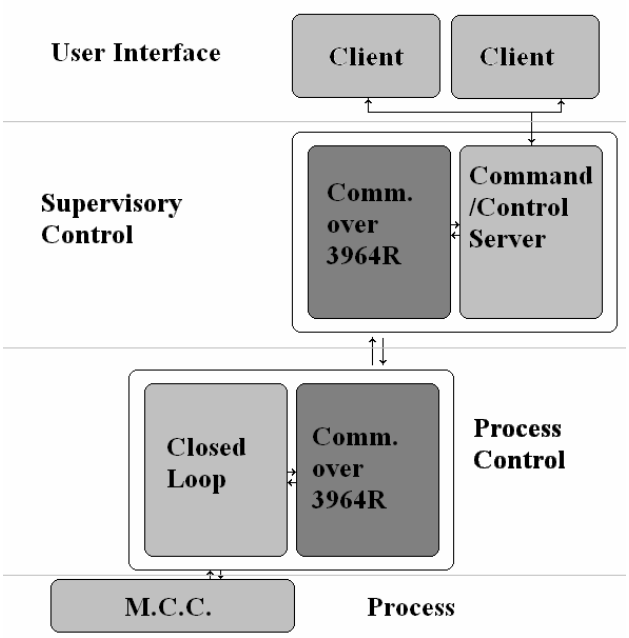

Figure 3 Software Interfacing

Testing

Both functional and system testing has been done in the Laboratory Environment. The time-criticality of the real-time communication protocol 3964R could be well understood and implemented in the test phase only. Due to this reason, the supervisory control was separated into two parts i.e. Communication part and Server part. Otherwise, even multi threading didn't help much.

Challenges

The most challenging part of the project was to implement the real-time communication protocol 3964R. Though the requirement originally was a hard-real-time Operating system, still the soft-real-time OS has served the purpose with some compromise on the initial conflict of the handshaking of the protocol. Since the Process to be controlled is a slow one, this doesn't matter for the performance.

Conclusion

- Testing is the most important phase of any software development.
- A Soft-real-time system may be developed over the Windows Platform for a slow Process
- To fix bugs, the product must under go a long testing phase may be for weeks together.
- Network Traffic does matter to the performance of a system, which works even over a LAN.
- PLC-PC combination gives a good hardware environment for closed loop process control.

References

[1] http://itcofe.web.cern.ch/itcofe/Services/PLC/WhatIsPLC/welcome.html.
[2] Siemens S7 series PLC programming guide for PLC user programming.
[3] VC++ online help for developing the VC++ projects.