

## NSLS CONTROL SYSTEM INTERFACE TO MODICON PLC

S.Ramamoorthy, S.Buda, J.Oliva, G.Ramirez, BNL, N.Y. 11973, USA

### Abstract

The hardware engineering efforts at the National Synchrotron Light Source facility are currently geared towards the use of programmable logic controllers (PLCs) for the control of machine hardware. The PLC provides a cost-effective solution combined with easy maintenance and upgrade. Modicon control products from Schneider Automation Inc. have been chosen since a wide range of PLC's, inexpensive I/O modules and networking options are offered by the company. The NSLS Control Monitor software uses the Modbus/TCP which is a variant of Modbus protocol to communicate with the PLCs. This paper describes the software interface to the Ethernet attachment on the PLCs.<sup>1</sup>

### 1 INTRODUCTION

The PLCs have been extensively used in industry to automate a variety of control and monitoring functions for many years. They are flexible, versatile and are built to withstand harsh factory conditions. However, their role was confined to industries due to their slow response, slow processing power and lack of communication standards. Because of recent evolution towards PLCs with fast processors, a variety of compatible I/O products and communication adapters that support TCP/IP network, the PLCs have made a quantum leap into the arena of distributed-control systems for large accelerator facilities.

The control system at the NSLS has a two-level distributed architecture consisting of HP/c8000 series workstations for the high-level (the operator level) and 80 VME-based microprocessor subsystems (referred to as micros) for the lower level. The communication link is a high-speed Ethernet. The micros are responsible for the control and monitoring of the storage ring hardware. The equipment to be controlled, dictates the types of I/O boards for a micro. The I/O peripherals include ADC, DAC and digital IO cards in the micro crate.

Though fast and reliable, VME I/O cards are very expensive and may not be required when controlling and monitoring devices with slow response. PLC was considered as an alternative. The hardware engineers also preferred to replace the traditional digital circuit

hardware with PLC. This will allow them to implement logic at the equipment level without the help from the software group. Future upgrades and maintenance will be very easy.

### 2 DEVELOPMENT OF PLC-BASED CONTROLS AT NSLS

The decision factors when selecting the Modicon products were cost, technical support, availability of a wide range of inexpensive I/O modules and networking solutions. The introduction of MODBUS-TCP protocol combined with the networking hardware by the Schneider Corporation, has greatly simplified the interface between the control system and the PLCs.

At NSLS two types of processors are used. One is the TSX Quantum Controller. The logic is stored and run in the Quantum controller. The processor communicates with I/O module using the Modbus Plus protocol. An Ethernet module connected to the processor via quantum back-plane provides the link to the control system. The other processor is from the family of TSX Momentum products which include processor adapters, communication adapters, option adapters (to provide the processor with additional networking capability), and a variety of I/O modules. The modular design of the adapters and the simple plug-in and wiring of the products allow easy integration of a system that meets the NSLS requirements.

The Concept Software is used for PLC program development. This complies with the Microsoft Windows GUI interface and the IEC 1131-3 standards for PLC programming. The Concept software provides a great development and debugging environment and generates documentation for the programmers.

### 3 INTEGRATION OF PLCs TO THE NSLS CONTROL SYSTEM

The operations at the NSLS facility rely heavily on a set of high-level programs that run on workstations. These programs access the various analog and/or digital hardware parameters and software variables by meaningful names (for example Uvrf1trip, Xraycurrent, etc).

The complexity and diversity of various hardware used in the facility make each micro unique at the

<sup>1</sup> Work performed under the U.S. DOE Contract No: DE-AC02-98CH10886

hardware control level. However, the real-time software (referred to as NSLS Control Monitor [1] and is executed by all the micros), presents a standard interface to the high-level programs. The monitor views all I/O signals in any micro as a set of devices with definite types. The types define a device as analog IN/OUT or Digital IN/OUT or a combination of analog and digital or as an array. A standard set of commands (READ/SET) has been defined for access by high-level programs. The application modules in the micro interpret the commands and operate on the hardware.

### *3.1 PLC as a remote Slave I/O*

To fit into the framework of NSLS controls, the PLC is treated as a remote slave I/O to a micro, the physical link being the Ethernet. Using the TCP/IP protocol, the micro communicates with the server that resides in the firmware of the PLC's Ethernet interface. The server uses the registered port 502. Once the client makes the connection to the server, it can periodically send messages to the PLCs. The server automatically closes the connection if it is inactive for more than 10 seconds. The server is normally designed to support multiple connections (5 to 8) by the vendors.

### *3.2 Data transaction between PLC and Micro*

The underlying protocol for the data transaction is called Modbus Protocol. It uses a query-response cycle between a master and the slave devices. The query is normally initiated by the micro. Every query gets a response from the PLC. The request contains the address of a register, a function code and data items, if any. The basic function codes are Read or Write a single register or multiple registers. The response function code indicates whether the reply has valid data or an error code. The protocol used over the Ethernet is called Modbus TCP/IP. It basically inserts a MODBUS frame into a TCP frame and sends it as a message. The information for packing messages is available in MODBUS/TCP specification manual [2].

The PLC logic is responsible for data acquisition and control of the hardware in response to messages from the micro. It allocates a block of 16-bit wide registers (referred to as 4x registers in Modbus PLC language) for data exchange with the micro. During the execution of its logic cycle, the digitized data are stored in the registers. All discrete inputs are packed into 16-bit words and stored. The micro can read a maximum of 125 registers in one message transaction. This reduces the network traffic considerably. To write set-point data or On/Off control commands to a hardware the micro sends the data or command to a register assigned by the PLC logic. The PLC logic executes the commands, which can be "setting a DAC" or "turning a supply ON/OFF" etc.

## **4 PLC DRIVER SOFTWARE**

The driver supports multiple clients. The maximum number of clients has been arbitrarily chosen as 5. It can be extended if required. The multi-client driver is useful for the following. (1) The maximum number of registers for one Modbus transaction is 125. If the number of I/O signals requires more than 125 registers one can use multiple clients for data access from different register segments. (2) In some micro systems, more than one PLC unit with different IP addresses may be used. The micro can concurrently communicate with all the PLC units by setting up multi-clients. The driver provides three utility functions to the micro application module. All the functions return a status code and the data, if any. A non-zero status code indicates failure. The value of the code describes the type of the error.

### *4.1 Init Function*

The application should first invoke this function with the following parameters: the upper and lower limit for the registers that will be used for data exchange, IP address of the PLC and the polling period. The application program specifies the frequency at which the "read register" command is to be sent to the PLC. This is based on the response time of the hardware and the PLC logic cycle time. The function checks the validity of the parameters. A successful **init** call will spawn a client task with the same priority as the application module.

### *4.2 Read Function*

The application module uses the **Read** function to retrieve the data from any register range so long as it is within the declared set. The application should first test the status code for error before updating the data field within the device record. There is no need to call this function faster than the polling rate.

### *4.3 Write Function*

When a high-level program sends a command (analog or digital or an array) to a device, the application module encodes the command in an appropriate format and invokes the write function to post the information to the client task. The first register number, number of registers and the data items should be specified.

### *4.4 Client Task*

The client task opens a TCP connection to the specified PLC server. If the connection is successful, the client will periodically send a "Read Multiple

Register" request for the specified registers. The response and error status, if any, will be stored in the memory accessible by the application module. In order to keep the connection alive, the maximum delay should not exceed 5 seconds. The task checks whether any new SET message (ON/OFF, set-point etc) has been posted by the application module. This is checked at a frequency of 50 Hz. The message is formatted using the specified register number and the data and sent to the PLC. Set messages always take precedence over Read messages. The client uses the "select call" with 5 seconds time-out before reading the socket. If the PLC is turned off or the physical link is lost the task closes the connection and retries again.

## 5 MICROS WITH PLCS

1. The UV beam line status micro continuously monitors the status of the safety shutters, photon mask, vacuum valves etc. for the 16 beam lines in the UV ring. It generates a real-time display on a local CATV and reports alarm conditions to the Alarm Handler. The interlock system for the beam line user stations consists of PLCs to operate the vacuum and fast valves, the photon mask and the safety shutter. The sequence of operation and interlocking of these components is controlled by PLC logic executed in a local PLC-CPU. The local processor for each beam line communicates with others via Modbus Plus communication bus. The Modbus Plus bus is also connected to a master PLC that controls the master shutter and the RF permit for operation. Information from all the beam line components is passed to the master PLC using peer cop communication through the Modbus Plus bus. The data is available in 16 bit registers in the master PLC. The master PLC processor is equipped with an Ethernet port for communication with the control system. Its registers contain information such as the open/closed status of the shutters, vacuum valves and safety shutters. The TSX momentum family products are used in this system.

2. The Booster-UV transport micro controls the operation of the transport power supplies and the UV trim micro controls the trim power supplies. For the ON/OFF control and status read-backs of these power supplies four non-intelligent I/O modules each with 16-bit inputs are used for reading the status of the power supplies. One non-intelligent I/O module with 16 inputs and 16 outputs is used to control the main and auxiliary contactors for the power supplies. The quantum PLC

processor communicates with the IO modules via modbus plus ports. The quantum controller is coupled to an Ethernet module for communication with the VME micros.

3. Four micros control the RF systems for the UV and X-ray rings. The logic in the VME systems checks for various fault conditions before turning the amplifiers and the RF system on. A backup hardware logic board is also used in each system. The future plan is to eliminate the hardware and control the amplifier and RF system turn on/off sequences with PLCs. A PLC system for amplifier controls has been implemented and is ready for integration with the micro system. The micro will still implement DAC controls that require fast response while ramping. The PLC system for the amplifier controls has a TSX momentum processor and 2 I/O bases for analog read-backs and one I/O base for digital inputs/outputs. The PLC processor communicates with the I/O modules via I/O bus. The PLC program residing in the processor's flash memory implements all the amplifier turn on/off sequences, monitoring of fault status and digitization of analog signals. These data are available in 16-bit registers in the PLC memory. The Ethernet on the processor adapter communicates with the control system. One more PLC unit will be added to the system for RF system control.

## 6 CONCLUSIONS

The Modicon quantum processors and the TSX momentum family products and the Modbus/TCP protocol have facilitated seamless integration between PLCs and the micros. The software can be easily incorporated in any operational micros. The integrated systems have been stable and reliable.

## 7 ACKNOWLEDGEMENTS

The authors wish to thank R.Biscardi, E.Blum, M.Fulkerson, N.A.Towne J.Vaughn and E.Morello for their discussions and assistance

## REFERENCES

- [1] S. Ramamoorthy et al, "NSLS Control Monitor Upgrade", Proc IEEE, 1849,PAC 1993.
- [2] A. Swales, "Open Modbus/TCP Specification", <http://www.modicon.com/openmbus/eio/enetio.html>, March 1999.