

Fully Integrated Modbus Communication Functions on STARDOM PLC/RTU

Industry: Oil & Gas, Water and Waste Water, Power, Environment Monitoring

Product: STARDOM FCN/FCJ Intelligent RTUs, FAST/TOOLS SCADA Package

Introduction

With its open and simple specifications, the Modbus protocol is widely used to control device communications in industrial applications. As indicated in the following table, Modbus modes and functions differ depending on cable type.

Cable type	Serial		Ethernet
Mode	ASCII	RTU	TCP
Functions	Master/slave	Master/slave	Client/server

Project Overview

Yokogawa Engineering Asia Pte. Ltd. was engaged in an upstream supervisory control and data acquisition (SCADA) system project where there was a requirement for Modbus systems that could operate simultaneously as master and slave nodes. As STARDOM field control node (FCN) autonomous controllers allow both Modbus master and slave functions to run simultaneously, they were selected for this project.

While functioning as a Modbus master mode in this application, the FCN polls the fire and gas system (FGS) to obtain data. As a Modbus slave node, it then transmits FGS data to the Yokogawa FAST/TOOLS SCADA system via a Modbus gateway and a WAN. This gateway converts the Modbus RTU protocol to Modbus TCP protocol for communication over the WAN. The serial modules are duplexed for network redundancy.

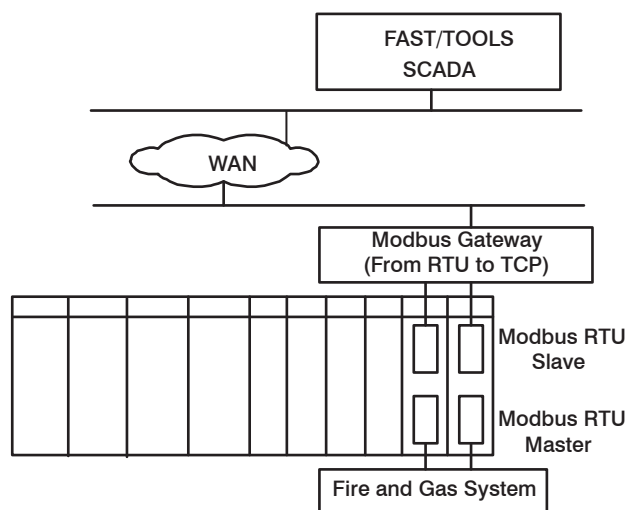


Figure: System Configuration

Expected benefits

The NFLR121 FCN serial communications module has two ports for either RS-422 or RS-485 communications, and each can be configured as slave or master. By installing two of these modules, the second module is made to be a backup.

With STARDOM, the following benefits are expected:

- Reduced installation costs
- Simplified system configuration
- Increased system reliability

STARDOM Engineering

STARDOM supports the IEC61131-3 programming language. The following two engineering tools are available:

- Resource Configurator: for hardware settings
- Logic Designer: for application logic

By keeping the application logic separate from the physical addresses, engineers can write programs for specific applications without having to consider the hardware configuration. This allows programs to be reused and improves engineering efficiency. The following screenshot from the Resource Configurator tool shows how the virtual name for a physical port is defined. In this case, SERIAL 1 is set for port1 and SERIAL 2 is set for port2 of the NFLR121 module.

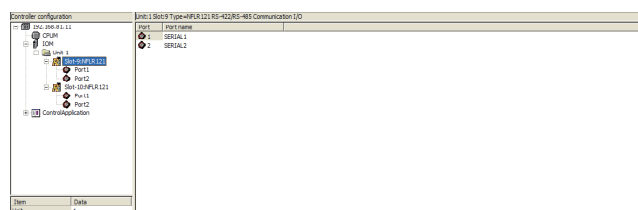


Figure: Port name definition in Resource Configurator

By setting a virtual name for the program organization unit (POU) parameter, it is possible to link the physical hardware and the application logic. In the following figure, SERIAL 1 is set for SD_CMDBSM_BM_OPEN, which executes a Modbus master task. This specifies that port1 on an NFLR121 module is used as the communications port.

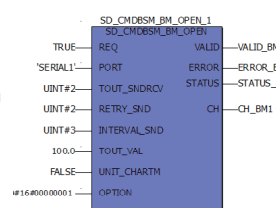
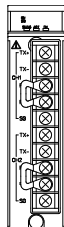


Figure: Setting of virtual name for port in POU

Application Overview

The NFLR121 FCN serial communications module has two ports for either RS-422 or RS-485 communications that support communication speeds between 300 bps and 115.2 kbps. In this project, one port was set as a Modbus master and the other as a Modbus slave. For redundancy purposes, two modules are installed in the following side-by-side configuration:



NFLR121

PSU Power Supply Unit	PSU Power Supply Unit	CPU	CPU	Spare	AI	DI	DO	COM	COM
--------------------------------	--------------------------------	-----	-----	-------	----	----	----	-----	-----

Figure: Unit Configuration

For this application, the following tasks were carried out in the sequence shown:

1. Port configuration
2. Modbus RTU master and slave programming for the same module
3. Redundant Modbus RTU slave programming

Port Configuration

With the Resource Configurator, the following communication settings were made for each port:

- Port name
- Wiring method (2 wire/ 4 wire connection)
- Half duplex / full duplex
- Baud rate
- Data bits
- Parity setting

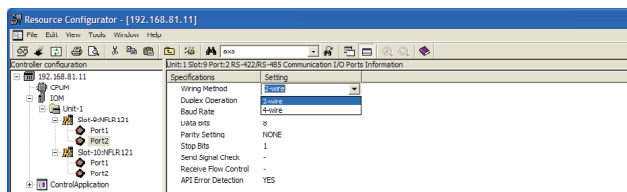


Figure: Setting of wiring method for port 2 on NFLR121

Programming the Same Module as Modbus RTU Master and Slave

For this application, port name SERIAL 1 was set to port 1 and port name SERIAL 2 was set to port 2 for the NFLR121 module in slot 9.

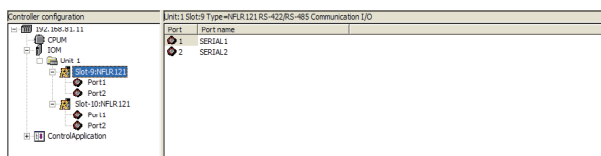
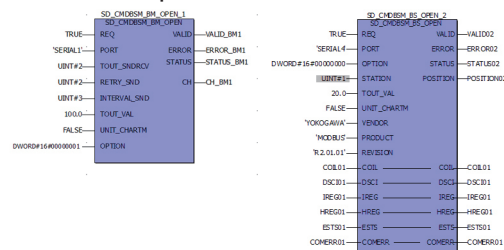


Figure: Port name definition with Resource Configurator

As shown below, two POU were used in this application: SD_CMDBSM_BM_OPEN for the Modbus RTU master node and SD_CMDBSM_BS_OPEN for the slave node. Port 1 on the NFLR121 module was used for communication as a Modbus RTU master node and port 2 was used as a slave node.



Redundant Modbus RTU Slave Programming

NFLR121 modules were installed in slots 9 and 10. In this application, port name SERIAL 3 was set to port 1 and SERIAL 4 was set to port 2 for the redundant NFLR121 module in slot 10.

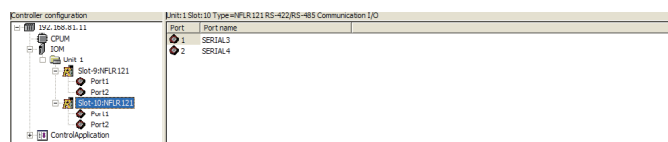
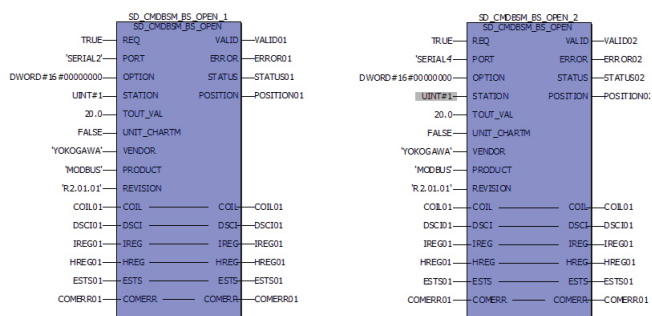


Figure: Port name definition with Resource Configurator

The two SD_CMDBSM_BS_OPEN POU's shown below were used in this application. Port 2 of the NFLR121 module in slot 9 and port 2 of the NFLR121 module in slot 10 were used for communication as a Modbus RTU slave node. In the event of a failure of port 2 on the NFLR121 module in slot 9, communication would not be disrupted.



Conclusion

Each of the communications modules on a STARDOM FCN autonomous controller can simultaneously function as Modbus master and slave nodes without affecting the other settings in the customer site. The use of two modules makes communications fully redundant.

In this application, the FCN Modbus master node simultaneously polls data from the FGS while the Modbus slave node transmits the FGS data to FAST/TOOLS. This keeps installation costs to a minimum. Furthermore, the use of the redundant module configuration ensures that there will be no interruption in the communications between FAST/TOOLS and the FCN autonomous controller.