Development of the *LAN Interface Card* with Modbus

Katsuhiro Yoshizawa Yutaka Kato Kiyoshi Mizuguchi

Kenji Higuchi Yuki Hara Naoki Takemoto

1. Introduction

IoT has been incorporated in production lines at an ever-increasing pace, and it has become common practice to monitor the operating status of industrial equipment remotely.

In such an environment, demand is increasing for a single monitoring system that can batch monitor industrial equipment and uninterruptible power supplies (UPS) used for backing up this equipment.

We developed the *SANUPS LAN Interface Card* (hereinafter, *LAN Interface Card*) as a UPS optional product. It is used for monitoring UPSs remotely, and for shutting down computers in data centers and server rooms automatically during power failures. However, with the increased popularity of virtual servers, there is growing demand to use LAN Interface Cards in equipment with comparatively complicated shutdown procedures.

To respond to the above demands, we have added or enhanced the following features in the newly developed *LAN Interface Card*.

- We added Modbus, a widely used communication protocol between industrial equipment, to enable use in industrial environments.
- We enhanced functions to enable use in information systems environments for automatic shutdown of virtual servers that require complex suspension procedures.

This article provides an overview of the newly developed *LAN Interface Card* (new model) and introduces its features.

2. Product Overview

The new models maintain all functions of our current product and feature an additional Modbus communication function.

There are two forms of Modbus communication: the Modbus TCP protocol, which communicates over networks, and the Modbus RTU protocol, which uses serial communication.

The new models are divided into two groups: one that supports the Modbus TCP protocol, and another that supports both Modbus TCP and Modbus RTU protocols.

Figure 1 shows PRLANIF021A, which only supports the Modbus TCP protocol.

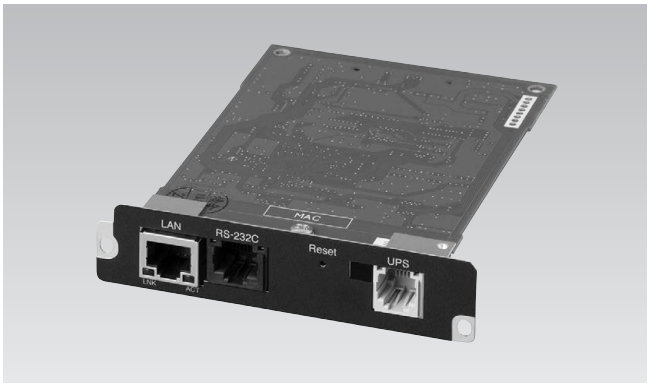

Fig. 1 PRLANIF021A

Figure 2 shows PRLANIF023A, which supports both Modbus TCP and Modbus RTU protocols. When connecting to a device that uses the Modbus RTU protocol, a communication cable is connected to the EXT port on the front panel.

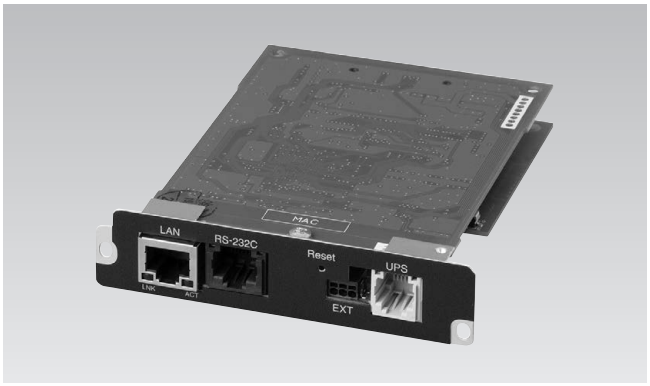

Fig. 2 PRLANIF023A

We also added a model with a cover around the PCB that protects from impacts so the *LAN Interface Card* can be transported while mounted.

Figure 3 shows PRLANIF024A, a model that supports both Modbus TCP and Modbus RTU protocols and has a protective cover.

Fig. 3 PRLANIF024A

3. Features

3.1 Modbus master/slave functions

The Modbus protocol uses a master/slave communication method whereby the master issues a request to the slave, which then returns a response.

The new models support both Modbus master and Modbus slave communication methods.

Figure 4 shows a visualization of the Modbus communication method.

When installing a UPS in an environment with a PLC or a monitoring control system such as a supervisory control and data acquisition (SCADA) operating as a Modbus master, the *LAN Interface Card* is set to Modbus slave communication. The UPS can be remotely managed from Modbus masters, enabling users to check UPS operating status and control the UPS (battery test, UPS stop, UPS start).

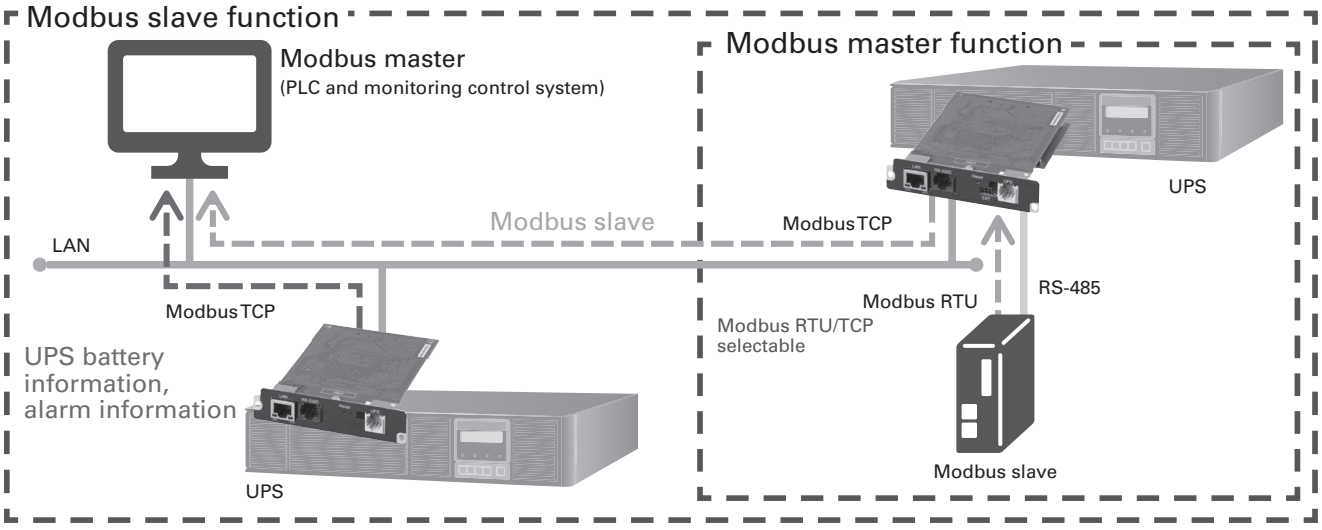

Fig. 4 Visualization of Modbus communication method

When connecting a UPS to measuring instruments such as wattmeters or I/O units via Modbus slave communication and reading measurement data or status information, the *LAN Interface Card* is set to Modbus master communication.

Modbus-related settings are performed using the Web Management Tool, the same as for our current models.

Figure 5 depicts the Web Management Tool screen for setting the Modbus connection method.

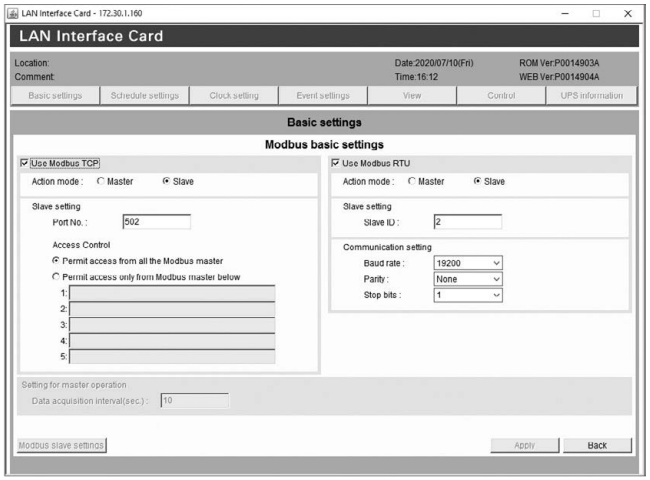

Fig. 5 Modbus basic settings screen

3.1.1 Modbus master/slave settings

On the new models, users can select either master or slave communication method for each communication protocol. For new models that support both Modbus TCP and Modbus RTU protocols, users can combine communication methods, such as slave for Modbus TCP and master for Modbus RTU. However, master cannot be set for both Modbus TCP and Modbus RTU at the same time.

Table 1 shows available communication method combinations.

When using Modbus master and slave communications together, as shown in Figure 4, the Modbus RTU protocol can be used as a Modbus master connection to gather information from Modbus slave units, and then pass that information to a host Modbus master unit via Modbus slave communication.

Table 1 Available communication method combinations

By product type/ combination pattern		Modbus TCP		Modbus RTU	
		Master	Slave	Master	Slave
Modbus TCP model	1	○	-		
	2	-	○		
Modbus TCP/RTU model	1	○	-	Disabled	Disabled
	2	-	○	Disabled	Disabled
	3	Disabled	Disabled	○	-
	4	Disabled	Disabled	-	○
	5	○	-	-	○
	6	-	○	○	-
	7	-	○	-	○

(○: Can be set, Disabled: Function is set as disabled)

3.1.2 Modbus slave function

If used with the Modbus slave function enabled, the new models can manage the UPS from the system's Modbus master unit in the same way as other Modbus slave devices.

UPS operational information can be checked and UPSs can be controlled from Modbus master units monitoring customers' production equipment, such as PLCs and SCADA systems. As such, there are expectations toward applying this technology to production line management and monitoring.

Table 2 shows UPS information and control functions available from Modbus master units.

Table 2 Acquirable information and UPS control functions

Major acquirable information	· UPS status information · UPS measurement value information · UPS output status · UPS battery information · UPS profile information (serial numbers, product names, rated capacities, etc.) · LAN Interface Card event log (up to 10 recent events) · Alarm status · LAN Interface Card connection unit information · Measurement deviation information · Modbus slave unit information
UPS control functions	· Battery test (start/cancel) · UPS stop · UPS start · UPS reset (UPS stop → start)

Note: Information that can be acquired differs by UPS model.

3.1.3 Modbus master function

When connected to a Modbus slave unit such as an I/O unit or measuring instrument, the new models can monitor status and measurement data of up to 16 points.

When importing status data, the user can set the UPS to treat a situation as abnormal when the signal status changes from off to on, or vice-versa.

When acquiring measurement data from Modbus devices, measurement values may include decimal numbers or be several digits long, depending on the device. The new models can be set to import measurement value data from a variety of devices. Moreover, a threshold can be set for the measurement values being imported. If readings deviate from the threshold value, an email notification is sent to a preset email address and the event is recorded in the data log.

Figure 6 is an example of a screen for registering a Modbus slave unit to acquire measurement values.

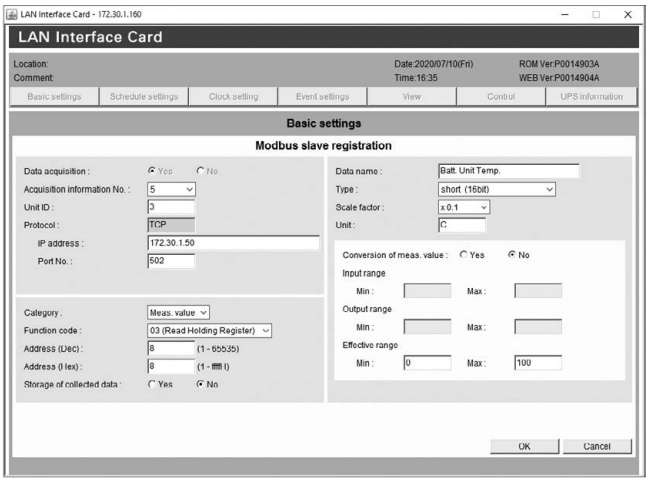

Fig. 6 Example screen
for registering Modbus slave units

Status information and measurement values retrieved from the Modbus slave unit can be checked from the Web Management Tool.

Figure 7 depicts an example of a screen for displaying status and measurement values from the Modbus slave unit.

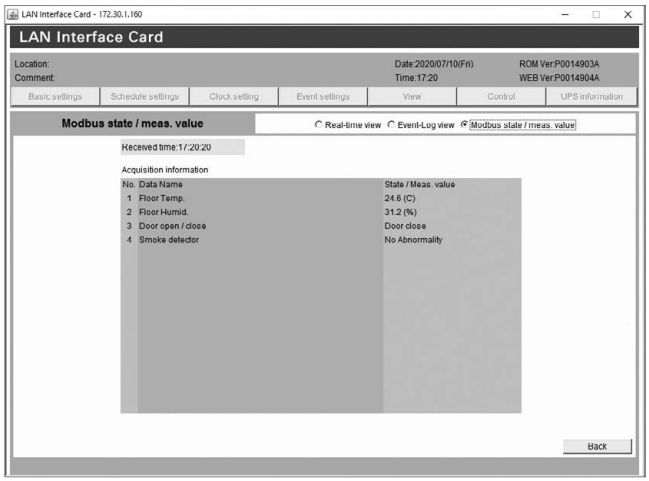

Fig. 7 Example screen
for displaying Modbus slave unit information

3.2 Enhanced shutdown functions

With the WScript function, even our current LAN Interface Cards models can safely shut down computers that are backed up by UPSs in the event of a power failure.

When a server needs to shut down in a power outage, with server shutdown commands registered to the *LAN Interface Card*, this function uses the SSH/Telnet protocol to remotely operate the server, log in, and then use the registered commands to shut down the server.

In conventional server environments, servers, data storage devices, network devices, and other infrastructure all used dedicated hardware. But in recent years, there has been an increasing use of virtualized servers that integrate and operate multiple OSs within a server and hyper-converged infrastructure (HCI) that virtualizes infrastructure in software.

For the new models, we have enhanced the shutdown function of virtualization platforms such as virtualized OSs and HCI.

3.2.1 Expansion of WScript function setting capacity

When shutting down virtualization platforms, several operations such as shutting down the guest OS, disabling the storage function, and disabling the management controller must be performed in a certain order.

With current models, there was a limit to the registration capacity for WScript commands, and sometimes it was not possible to set all of the procedures necessary to shut down virtualization platforms.

On the new models, the command registration capacity was increased from 598 to 10,238 bytes: sufficient space for complex virtualization platform shutdown procedures.

3.2.2 Additional commands to the WScript function

Table 3 lists the commands of the current models' WScript function that can be used to shut down a server.

When shutting down a virtualization platform, all guest OSs must shut down before disabling the storage function.

The current models lack a command to check whether or not a guest OS has shut down, so after the operation to shut down a guest OS executes, they must wait the maximum time for a guest OS to shut down using the sleep command. As such, even if the guest OS shuts down early in the wait period, the storage function will only stop after waiting for the set period. Therefore, it sometimes took longer than necessary to shut down systems.

Table 4 shows a list of the commands newly added to the WScript function.

We have prepared a wait_dev command and delay_dev command to proceed to the next process even within the stop/start wait time of the guest OS without waiting for the set wait time to finish if it can be confirmed that the designated unit has either stopped or started.

Furthermore, some virtualization platform administrators expressed a desire to use script files they had prepared themselves to stop or start a system.

With the latest development, if the user sets a shell script they have prepared together with our commands, the contents of that shell script will be sent to a server and directly executed on the server side.

Using the new commands, it is now possible to shut down even complex systems such as virtualization platforms within the UPS battery backup time, satisfying our customers' requests.

Table 3 List of commands used with the current model

Command	Description
Send	Set the character string (command) to be sent to the WScript side. Example: send=shutdown
Wait	Set the character string (such as a prompt) to be received on the UPS side. Example: wait=login
sleep	Set the wait time until the next process. Unit: Seconds Example: sleep=90
delay	Delay script start by the designated time. Unit: Seconds Example: delay=120

Table 4 List of additional commands of the new models

Command	Description	Example
wait_dev_on	Designate name (address) of devices to check activation status and have script processing wait until all devices are activated.	wait_dev_on=[192.168.1.1 192.168.1.2]
wait_dev_off	Designate name (address) of devices to check shutdown state and have script processing wait until all devices are shut down.	wait_dev_off=[192.168.1.1 192.168.1.2]
delay_dev_on	Designate registered device names (addresses) and delay running of the script until all devices are activated.	delay_dev_on=[192.168.1.1 192.168.1.2]
delay_dev_off	Designate registered device names (addresses) and delay running of the script until all devices are shut down.	delay_dev_off=[192.168.1.1 192.168.1.2]
<begin_shell_script> <end_shell_script>	Execute shell script created by user. When shell script file content is entered between "<begin_shell_script>" and "<end_shell_script>," that content is sent to the server to be executed.	<begin_shell_script> #!/bin/sh ⋮ <end_shell_script>

3.3 UPS measurement storage function

The *LAN Interface Card* acquires measurement data from the UPS approximately every 10 seconds.

The new *LAN Interface Card* models can store up to seven days' worth of measurement data.

If a fault occurs on the UPS, the stored measurement data can be used to analyze the cause of the fault.

This stored data can be acquired using the email function or FTP function.

Moreover, when using the Modbus master function to acquire measurement data from the Modbus slave unit, it is possible to save up to seven days' worth of measurement data in the same way as UPS measurement data. This stored data can also be acquired using the email function or FTP function.

4. Specifications

Figure 5 shows the specifications of the new models. PRLANIF021A and PRLANIF022A support only Modbus TCP, while PRLANIF023A and PRLANIF024A support both Modbus TCP and Modbus RTU.

Table 5 Specifications

Items	Ratings and standards			
Model	PRLANIF021A	PRLANIF022A (With cover)	PRLANIF023A	PRLANIF024A (With cover)
Dimensions	105 (W) × 125.5 (D) × 23.5 (H)			
Mass	80 g	120 g	110 g	150 g
Operating environment	Temperature: -25 to +60˚C Humidity: 0 to 90% RH (non-condensing)			
Power consumption	1.4 W		2.1 W	
LAN communication	Transmission speed: 100/10 Mbps (automatically detected) Transmission method (full-duplex/half-duplex): auto Auto-MDIX			
Function	· Automatic computer shutdown (multi-platform) · Shutdown of computers with redundant power supplies · Automatic computer startup at power restoration · Scheduled operation · UPS status display (Web Management Tool, Web Display Tool, SSH or Telnet) · SNMP agent (RFC1628, JEMA-MIB, SANYO DENKI private MIB) · Sending/receiving emails · NTP (Network Time Protocol) · Downloading and uploading UPS setting values · Test function (script execution, sending emails, sending SNMP traps, shutdown) · Notification of events to the Syslog server · Measurement deviation monitor (UPS internal information, Modbus measurements) · Statistical graph display function (UPS internal information) · Modbus master function / slave function · Status measurement of Modbus TCP slave device (max. of 16 points) · Storage of collected UPS/Modbus measurement data			
Modbus protocol	Modbus TCP		Modbus TCP, Modbus RTU	
Protocol	TCP/IP, UDP, DHCP, SNMP (v1, v2c, v3), HTTP, HTTPs, Telnet, SSH, FTP, FTPs, SMTP (over SSL/TLS), POP3 (over SSL/TLS), APOP, NTP			

5. Conclusion

This article has provided an overview of the *LAN Interface Card* with Modbus functions and introduced its features.

With the development of this product, UPSs can be managed in industrial networks, where demand has emerged recently, through the use of this *LAN Interface Card*.

Moving forward, we anticipate this product will be widely used not only for information systems, but also for industrial systems.

Moreover, with this development, we have enhanced the ability to shut down virtualization platforms, which are predicted to become even more widespread. Server management technology in information systems is rapidly advancing. For the *LAN Interface Card*, we will work to keep pace with the tide of technological innovation, swiftly develop products to satisfy market demands, and continue offering products that satisfy customers.

Author

Katsuhiro Yoshizawa
Power Systems Div., Design Dept.
Works on the development and design of power supplies and power management systems.

Yutaka Kato
Power Systems Div., Design Dept.
Works on the development and design of power supplies and power management systems.

Kiyoshi Mizuguchi
Power Systems Div., Design Dept.
Works on the development and design of power supplies and power management systems.

Kenji Higuchi
Power Systems Div., Design Dept.
Works on the development and design of power supplies and power management systems.

Yuki Hara
Power Systems Div., Design Dept.
Works on the development and design of power supplies and power management systems.

Naoki Takemoto
Power Systems Div., Design Dept.
Works on the development and design of power supplies and power management systems.