PowerLogic P3

Protection and control device

Communication User Manual

P3/EN/CM/30-208A

03/2025

www.se.com

Legal Information

The information provided in this document contains general descriptions, technical characteristics and/or recommendations related to products/solutions.

This document is not intended as a substitute for a detailed study or operational and site-specific development or schematic plan. It is not to be used for determining suitability or reliability of the products/solutions for specific user applications. It is the duty of any such user to perform or have any professional expert of its choice (integrator, specifier or the like) perform the appropriate and comprehensive risk analysis, evaluation and testing of the products/solutions with respect to the relevant specific application or use thereof.

The Schneider Electric brand and any trademarks of Schneider Electric SE and its subsidiaries referred to in this document are the property of Schneider Electric SE or its subsidiaries. All other brands may be trademarks of their respective owner.

This document and its content are protected under applicable copyright laws and provided for informative use only. No part of this document may be reproduced or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), for any purpose, without the prior written permission of Schneider Electric.

Schneider Electric does not grant any right or license for commercial use of the document or its content, except for a non-exclusive and personal license to consult it on an "as is" basis.

Schneider Electric reserves the right to make changes or updates with respect to or in the content of this document or the format thereof, at any time without notice.

To the extent permitted by applicable law, no responsibility or liability is assumed by Schneider Electric and its subsidiaries for any errors or omissions in the informational content of this document, as well as any non-intended use or misuse of the content thereof.

Table of Contents

Safety information

Important information

Read these instructions carefully, and look at the equipment to become familiar with the device before trying to install, operate, service, or maintain it. The following special messages may appear throughout this documentation or on the equipment to warn of potential hazards or to call attention to information that clarifies or simplifies a procedure.

The addition of this symbol to a "Danger" or "Warning" safety label indicates that an electrical hazard exists which will result in death or serious injury if the instructions are not followed.

This is the safety alert symbol. It is used to alert you to potential personal injury hazards. Obey all safety messages that follow this symbol to avoid possible injury or death.

DANGER

DANGER indicates a hazardous situation which, if not avoided, **will result in** death or serious injury.

WARNING

WARNING indicates a hazardous situation which, if not avoided, **could result in** death or serious injury.

CAUTION

CAUTION indicates a hazardous situation which, if not avoided, **could result in** minor or moderate injury.

NOTICE

NOTICE is used to address practices not related to physical injury.

User qualification

Electrical equipment should be installed, operated, serviced, and maintained only by qualified personnel. No responsibility is assumed by Schneider Electric for any consequences arising out of the use of this material.

A qualified person is one who has skills and knowledge related to the construction, installation, and operation of electrical equipment and has received safety training to recognize and avoid the hazards involved.

Use the password protection feature in order to help prevent untrained person and non-authorized personnel interacting with the PowerLogic P3.

About the document

Document scope

The purpose of this document is to describe the configuration and use of the Modbus client and the Modbus server for the PowerLogic P3 protection devices.

NOTE: This document assumes that the reader has some previous knowledge of the Modbus protocol.

Validity note

This document is valid for PowerLogic P3 with firmware version 30.208 and eSetup Easergy Pro version 4.9.1 or later.

General cybersecurity information

In recent years, the growing number of networked machines and production plants has seen a corresponding increase in the potential for cyber threats, such as unauthorized access, data breaches, and operational disruptions. You must, therefore, consider all possible cybersecurity measures to help protect assets and systems against such threats.

To help keep your Schneider Electric products secure and protected, it is in your best interest to implement the cybersecurity best practices as described in the Cybersecurity Best Practices document.

Schneider Electric provides additional information and assistance:

- Subscribe to the Schneider Electric security newsletter.
- Visit the Cybersecurity Support Portal web page to:
  - Find Security Notifications.
  - Report vulnerabilities and incidents.
- Visit the Schneider Electric Cybersecurity and Data Protection Posture web page to:
  - Access the cybersecurity posture.
  - Learn more about cybersecurity in the cybersecurity academy.
  - Explore the cybersecurity services from Schneider Electric.

Environmental data

For product compliance and environmental information, refer to the Schneider Electric Environmental Data Program.

Related documents

Table 1 - Related documents

Title of documentation	Document number
PowerLogic P3U20 and P3U30 User Manual	P3U/EN/M/30-208A
PowerLogic P3F30 User Manual	P3F/EN/M/30-208A

Table 1 - Related documents (Continued)

Title of documentation	Document number
PowerLogic P3G30 and P3G32 User Manual	P3G/EN/M/30-208A
PowerLogic P3L30 User Manual	P3L/EN/M/30-208A
PowerLogic P3M30 and P3M32 User Manual	P3M/EN/M/30-208A
PowerLogic P3T32 User Manual	P3T/EN/M/30-208A
PowerLogic P3 Modbus Profile[1]	P3/EN/MB/30-208A

To find documents online, visit the Schneider Electric download center (www.se.com/ww/en/download/).

Information on non-inclusive or insensitive terminology

As a responsible, inclusive company, Schneider Electric is constantly updating its communications and products that contain non-inclusive or insensitive terminology. However, despite these efforts, our content may still contain terms that are deemed inappropriate by some customers.

Abbreviations and terms

Table 2 - Abbreviations used in the document

Acronyms	Indication
COMTRADE	Common format for transient data exchange. File format for storing oscillography and status data related to transient power system disturbances.
DHCP	Dynamic host configuration protocol
EF	Earth fault
HR	Holding register
NTP	Network time protocol
OC	Overcurrent
RTC	Real-time clock
RTU	Remote terminal unit
SNTP	Simple network time protocol
TCP	Transmission control protocol

Table 3 - Terms used in the document

Terms	Definition
Modbus client (historically known as Modbus master)	A device that issues commands and receives responses from Modbus server devices. Serial Modbus networks can only have one Modbus client per network.
Modbus RTU	Serial Modbus protocol format; Modbus RTU transmits data using binary. Server and client devices must use the same format.
Modbus register/ address	Identifies the data to read or write. Modbus register maps are available for Modbus client devices and detail the information available from the client device. More than one register may be used to store a single value.

1. To view this file, please download PowerLogic P3 Protection and Control Device Communication User Manual to your local device and open it using Adobe Reader. The PowerLogic P3 P3 Modbus Profile is part of the PowerLogic P3 Protection and Control Device Communication User Manual. For access, navigate to the **Attachments** area.

Table 3 - Terms used in the document (Continued)

Terms	Definition
Modbus server (historically known as Modbus slave)	A device that responds to Modbus commands and performs actions or provides information back to the Modbus client. Most Modbus networks contain multiple server devices.
Modbus TCP	The Ethernet Modbus protocol format.

Modbus overview

Modbus is a client-server communications protocol where the client initiates transactions and the server or servers respond with the requested information or action. For more information on the Modbus protocol, visit www.modbus.org.

Modbus was originally designed for RS-232 and RS-485 physical interfaces, supporting two serial communication versions: Modbus RTU and Modbus ASCII. The PowerLogic™ P3 devices utilize the Modbus RTU version, which is suitable for serial communication. In addition, the Modbus TCP version enables communication over Ethernet and TCP/IP interfaces.

The Modbus RTU and the Modbus TCP can be used to transfer the following types of data:

- events
- status information
- measurements
- control commands
- clock synchronization
- protection settings
- disturbance recordings

The Modbus protocol defines four different categories of data. The mapping of data to these categories is implementation dependent. In the PowerLogic P3 devices, all data is mapped to the holding registers.

Table 4 - Modbus data organization

Primary table	Object type	Access	Comments
Discrete input	Single bit	Read-only	This type data can be provided by an I/O system.
Coils	Single bit	Read-write	This type of data can be altered by an application program.
Input registers	16-bit word	Read-only	This type of data can be provided by an I/O system.
Holding registers	16-bit word	Read-write	This type of data can be altered by an application program.

NOTICE

POTENTIAL COMPROMISE TO CYBERSECURITY

- For transmitting data over an internal network, physically or logically segment the network. Restrict the access to the internal network by using standard controls, such as firewalls, and other relevant features supported by your device, such as IP Table allowlist.
- For transmitting data over an external network, encrypt protocol transmissions over all external connections using an encrypted tunnel, TLS wrapper or a similar solution.

Failure to follow these instructions can increase the risk of unauthorized access.

The actual mapping of particular data items to holding registers in PowerLogic P3 devices can be checked with the setting tool by selecting the items under the **Modbus main configuration** setting view.

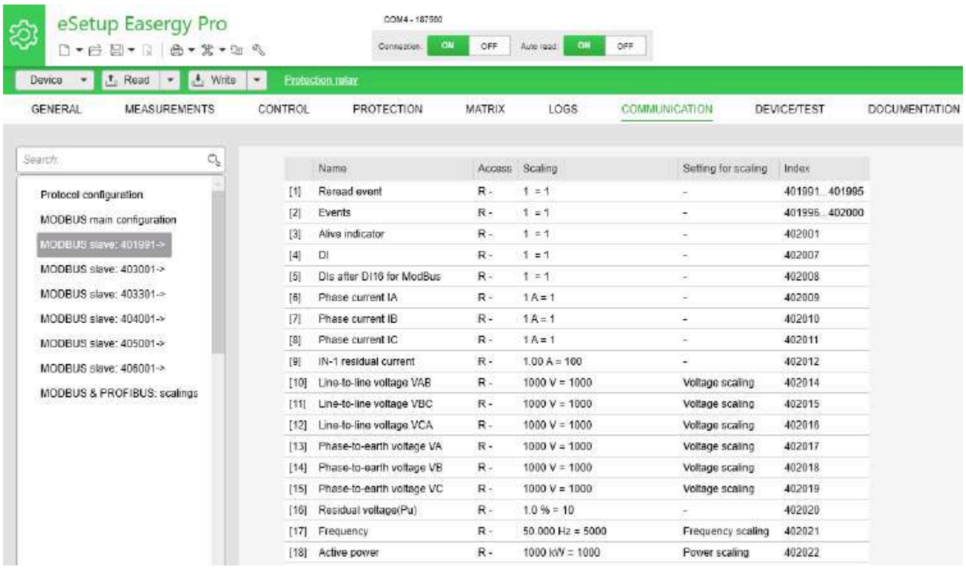
Figure 1 - Modbus main configuration setting view

NOTE:

- The data mappings are visible only when the Modbus protocol is activated. For instructions on how to activate the protocol, see Chapter .
- The values in the Address column (for example *40xxxx*) consist of two parts: *40* and *xxxx*, where the prefix *40* denotes a reference to Read/ Write Output or Holding Registers and *xxxx* denotes the address of a particular register.
- The holding register address sent via Modbus is one less than that indicated by a device. For example, if the Alive Indicator, holding register address 402001, is sent over a Modbus data link, the frame indicates that the address of the holding register is 402000.

The bit rate of Modbus RTU is typically 9600 bps and for the transmission of frames, a parity check mode must be selected (even, odd or none parity checking).

Figure 2 - Modbus data mappings



Your device can function as a Modbus client or Modbus server, depending on how it is configured.

Related information

Configuring the Modbus TCP

Configuring the Modbus RTU, page 13

Modbus client, page 16

Modbus server

Your device is capable of serving as a Modbus server, which has the ability to:

- supply the requested data to the client upon request.
- execute actions that are specified in the client's query.

In the context of Modbus RTU, each server on a single data link, which refers to a communication pathway is assigned a unique address ranging from 1 to 247. This ensures that each server can be individually addressed and communicated with by the client. Additionally, address 0 is designated for broadcast messages, allowing the client to send a query to all servers on the network simultaneously.

Function description

Clock synchronization

The internal clock of the PowerLogic P3 can be synchronized via the Modbus protocol. However, this is not a native feature of the Modbus protocol, but a PowerLogic-specific system. The accuracy of the clock synchronization is in the scale of 10 milliseconds.

The clock can be synchronized either by all fields (seconds, minutes, hours, days, month and year) or by only the minutes, which in turn sets the seconds and milliseconds to zero.

An example of how minute synchronization can be done: when the reference clock (the clock assumed to be correct) is exactly seven minutes past (any hour), a minute synchronization is performed. The result is that the internal clock of the device is set to *HH:07:00.000* (Hours:Minutes:Seconds.Milliseconds) *HH* is not changed.

These two ways of synchronizing the clock are denoted *Set RTC*, where RTC stands for real-time clock and *Synchronize Minutes* in the data map. The holding register address of the minute synchronization is *402502*.

Table 5 - Description of holding registers allocated to *Set RTC* synchronization

Holding register	Content
402504	Lower byte: seconds. Milliseconds are zero
402505	Upper byte: minutes Lower byte: hours
402506	Upper byte: day Lower byte: month
402507	Year

Clock synchronization is recommended to be performed hourly.

For Modbus TCP, clock synchronization can also be done by using simple network time protocol (SNTP). This requires an NTP server, the address of which is set in the **Protocol configuration > Ethernet** setting view in eSetup Easergy Pro.

Events

The event buffer of the PowerLogic P3 can be read via the Modbus RTU by reading one event at a time, from holding registers 402101…402105. The event registers contain the latest event, and are cleared when they are read. The registers are then updated to contain the following event from the event buffer.

Table 6 - Description of Events in holding registers

Holding register	Content
402101	Event code
402102	Event timestamp Bits 15-6 = milliseconds Bits 5-0 = seconds
402103	Event timestamp Upper byte = minute Lower byte = hour
402104	Event timestamp Upper byte = day Lower byte = month
402105	Event timestamp, year

If an error occurs when reading an event from registers 402101…402105, the previously read event is available in registers 402490…402494 for re-reading.

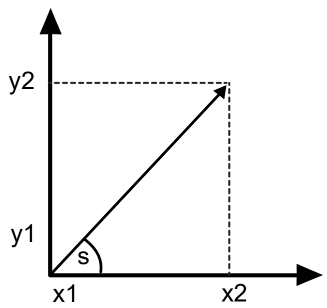
NOTE: Events can also be read from holding registers 401996…402000.

Events are coded with a numbering starting from 0. To obtain the meaning of these event codes, run the GETSET-command `g eventcodes` in the terminal software (for example PuTTY).

Scaling

Holding registers are 16 bits in size, so they can directly represent 2^{16} = 65535 different values, which might not be enough to describe the values of some physical quantity such as voltage or power. Thus, values transmitted over a Modbus data link must be scaled to account for this.

Figure 3 - Line defined by two points in a two-dimensional space



The scaling is determined by the equation of the line connecting two points in a two-dimensional space, (x_1, y_1) and (x_2, y_2). That is, the trajectory of the line and the offset from origin. Note that there is no offset in the example if the point (x_1, y_1) is assumed to lie at the origin.

It is common to use scaling factors with base ten (10, 100, 1000…) since, in such cases, the original measurements only lose decimals and such values are easy to read and re-scale to actual values on the client side after transmission. Note that the scaled measurements get rounded.

Different settings for scaling can be used for the power, power factor, tan phi, voltage and frequency scaling. These can be set in the **Communication > Modbus & Profibus scalings** setting view item in eSetup Easergy Pro.

A short example: The frequency is internally (in the device) stored as an integer value which also holds three decimal places, that is, 50.000 Hz is represented as

50000. This is a value too large to be represented with 16 bits (signed integer), however, frequency is by default scaled with the points: $(x_1, y_1) = (0, 0)$ and $(x_2, y_2) = (10, 1)$, enabling it to be sent over Modbus.

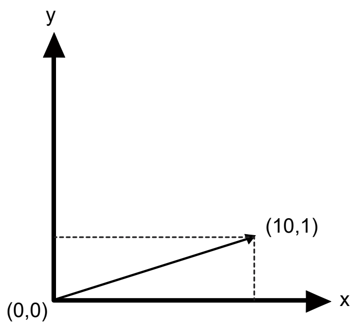
The slope is 0.1 and there is no offset.

$$k = \frac{y_2 - y_1}{x_2 - x_1} = \frac{1-0}{10-0} = 0.1$$

Thus, the value on the receiving side (the Modbus value) is:

$$value_{modbus} = k \times value_{internal} = 0.1 \times 50000 = 5000$$

Figure 4 - Default frequency scaling



To make sure that a scaling has been set as intended, you can check it in eSetup Easergy Pro by viewing the **Scaling** column for each holding register in the Modbus data mapping lists, for example in **Communication** > **Modbus slave 401991** .

For an example of scaling change, see the figures as follows.

Figure 5 - Voltage scaling changed from default (x_2 changed to 10 from 1)

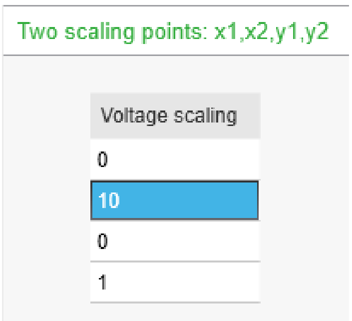


Figure 6 - Result of the change in voltage scaling

[6]	Phase current IA	R -	1 A = 1	-	402009
[7]	Phase current IB	R -	1 A = 1	-	402010
[8]	Phase current IC	R -	1 A = 1	-	402011
[9]	IN-1 residual current	R -	1.00 A = 100	-	402012
[10]	Line-to-line voltage VAB	R -	1000 V = 100	Voltage scaling	402014
[11]	Line-to-line voltage VBC	R -	1000 V = 100	Voltage scaling	402015
[12]	Line-to-line voltage VCA	R -	1000 V = 100	Voltage scaling	402016
[13]	Phase-to-earth voltage VA	R -	1000 V = 100	Voltage scaling	402017
[14]	Phase-to-earth voltage VB	R -	1000 V = 100	Voltage scaling	402018
[15]	Phase-to-earth voltage VC	R -	1000 V = 100	Voltage scaling	402019

Compare the scaling in Result of the change in voltage scaling, page 12 and Default scaling for some holding registers, page 13 .

Figure 7 - Default scaling for some holding registers

[6]	Phase current IA	R -	1 A = 1	-	402009
[7]	Phase current IB	R -	1 A = 1	-	402010
[8]	Phase current IC	R -	1 A = 1	-	402011
[9]	IN-1 residual current	R -	1.00 A = 100	-	402012
[10]	Line-to-line voltage VAB	R -	1000 V = 1000	Voltage scaling	402014
[11]	Line-to-line voltage VBC	R -	1000 V = 1000	Voltage scaling	402015
[12]	Line-to-line voltage VCA	R -	1000 V = 1000	Voltage scaling	402016
[13]	Phase-to-earth voltage VA	R -	1000 V = 1000	Voltage scaling	402017
[14]	Phase-to-earth voltage VB	R -	1000 V = 1000	Voltage scaling	402018
[15]	Phase-to-earth voltage VC	R -	1000 V = 1000	Voltage scaling	402019

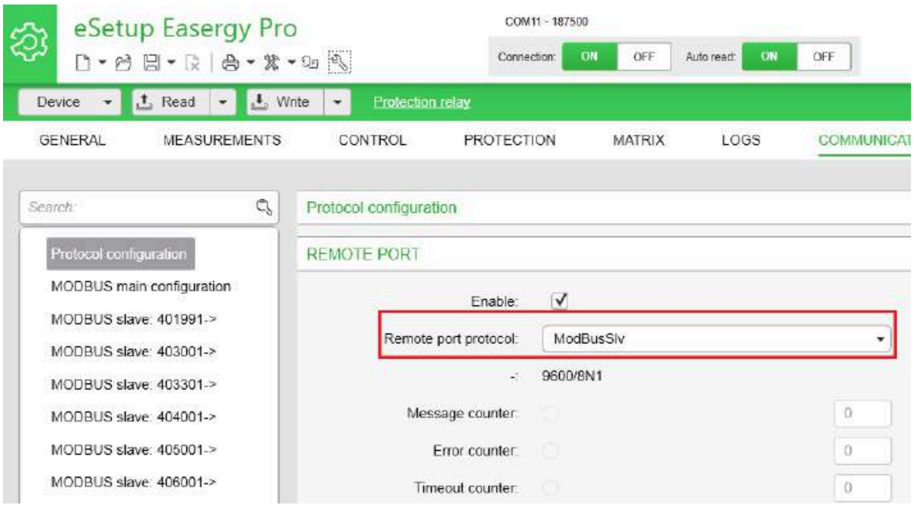
NOTE: To avoid overflow, scale values should be kept in the interval 0~32768.

Configuring the Modbus RTU

To configure the Modbus RTU:

1. In the eSetup Easergy Pro, go to **COMMUNICATION** > **Protocol configuration**.
2. Under **Remote port**, select **ModBusSlv** from the **Remote port protocol** drop-down menu.

Figure 8 - Selecting the Remote port protocol



NOTE: The protocol can also be enabled via the local panel of the device. You must enter the password with the configurator access level via the local panel before this is possible.

3. Reboot the device to ensure changes are applied.

4. Go to **COMMUNICATION > Modbus main configuration**, set the **Slave number**, **Modbus bit rate** and **Parity**.

Figure 9 - Modbus main configuration setting view

NOTE: The parity and bit rate must be set to the same value on all devices connected to the same data link.

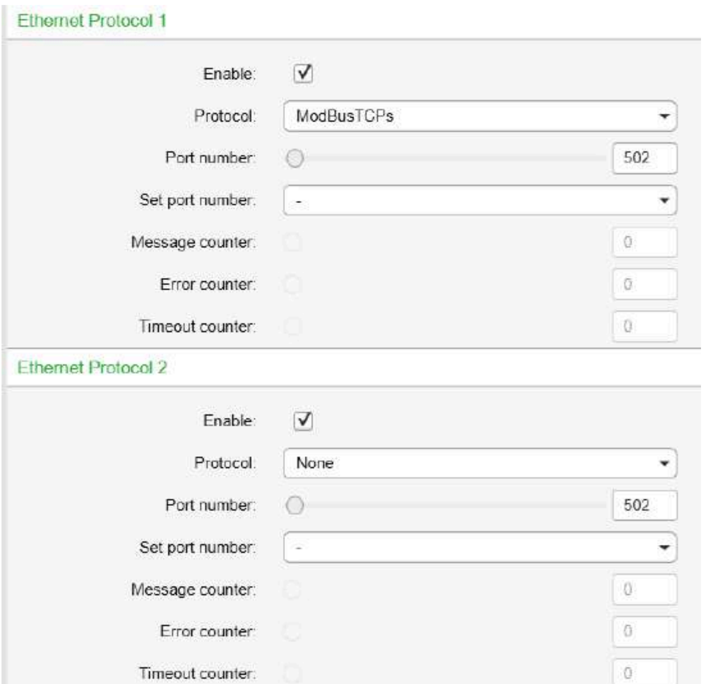
Configuring the Modbus TCP

NOTE: The PowerLogic P3 devices support dynamic host configuration protocol (DHCP) but usually a static IP address is used. Consider this before connecting a device to an existing network, so that no conflicts occur.

To configure the Modbus TCP:

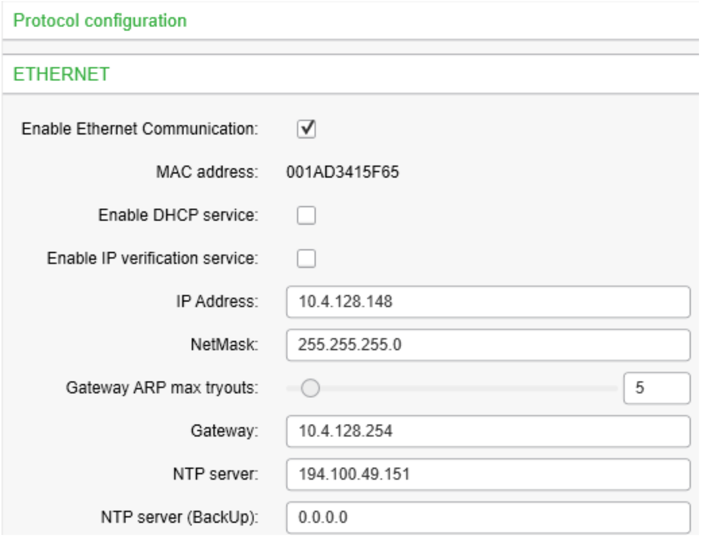
1. In the eSetup Easergy Pro, go to **COMMUNICATION > Protocol configuration > ETHERNET PROTOCOL 1**, select the **ModbusTCPs** from the **Protocol** drop-down menu.

Figure 10 - Setting Ethernet port protocol to ModbusTCPs



2. Scroll up to **ETHERNET** area, set the **IP Address**, **NetMask** (subnet mask) and **Gateway**.

Figure 11 - Ethernet setting view



3. Reboot the device to apply the changes.

NOTE:

- There are two different Ethernet protocols, Ethernet protocol 1 and 2. Modbus TCP can be selected to only one of them. IEC 61850 serves as an example of another protocol.
- Two Modbus TCP clients can communicate with the device simultaneously.
- Serial Modbus RTU can be used at the same time as Modbus TCP clients.

Modbus client

Your device is capable of functioning as a Modbus client, depending on how it is configured. The PowerLogic P3 device can only function as a client using the serial version of the protocol (Modbus RTU, only supports one server).

A PowerLogic P3 device functioning as a Modbus client needs to have a server ID defined, which identifies the server from which the client will request data, such as current and voltage measurements

The bit rate of the Modbus RTU is typically 9600 bps and for the transmission of frames, a parity check mode must be selected (for example, even, odd or none parity checking).

Configuration

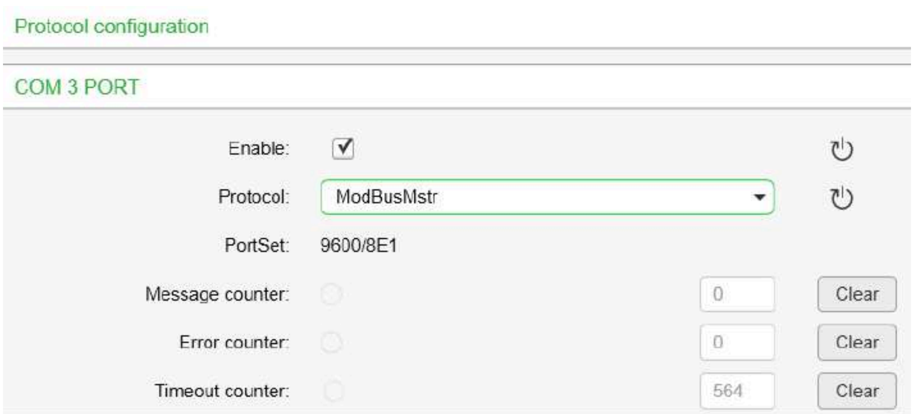
This section will describe how to configure a PowerLogic P3 device to use the Modbus RTU as a client on the bus.

NOTE: The configuration and features will vary between different PowerLogic P3 relay devices, such as what scaling settings there are and what data is available.

General configuration

1. To activate Modbus client, go to **COMMUNICATION > Protocol configuration** of the eSetup Easergy Pro.
2. Under **COM 3 PORT**, select *ModBusMstr* from **Protocol** drop-down menu.

Figure 12 - *ModBusMstr* selected as port protocol



NOTE:

a. The *ModBusMstr* protocol cannot be active on two different ports simultaneously. eSetup Easergy Pro will indicate this with an error message on one of the ports.

b. Setting a protocol on any port will require a reboot of the device before the changes will take effect. eSetup Easergy Pro will prompt for a device reboot after making such a change.

3. Under **MODBUS main configuration**, configure the Modbus bit rate and parity if needed.

Figure 13 - The Modbus main configuration menu in eSetup Easergy Pro

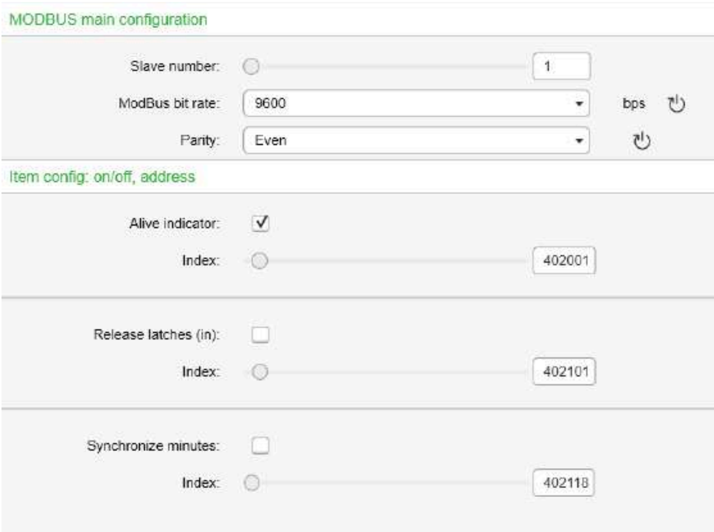


Table 7 - Description of the Modbus main configuration parameters

Param-eter	Range	Description
Slave number	1...247	The ID of the slave to which the device will perform data writes.
Modbus bit rate	1200 2400 4800 9600 19200	The bit rate used on the bus must be the same on all devices connected to the bus.
Parity	Even None Odd	The type of parity bit used for (rudimentary) error checking on the bus.

The additional settings visible in this menu are related to status and control of the device. These settings have two parameters which can be set, which are explained in following table.

Table 8 - Description of data item parameters

Param-eter	Range	Description
<Item name>	On	The data item is in use, written by the client device to the specified address on the client.
	Off	The data item is not in use.
Address	400001...465535	The address to which the client writes the value of the data item if enabled.

NOTE: Data items whose name contains a remark “(in)” refers to values that the client device reads from the server and acts upon if needed, such as the *Release latches (in)* item visible in figure The Modbus main configuration menu in eSetup Easergy Pro, page 17.

Data configuration

The data and control configuration in the *ModBusMstr* protocol is divided into six categories:

1. Measurements
2. Energy
3. DI and object status
4. Object control
5. Events
6. Running hours

Additionally, there are some control and status configuration settings in the main configuration menu (described in the previous subsection), as well as the scaling setting in their own menu.

The data items in these categories are defined in the same way as explained in the Table Description of the Modbus main configuration parameters, page 17. Data items can be enabled and their address (the address to where their values are written) can be changed.

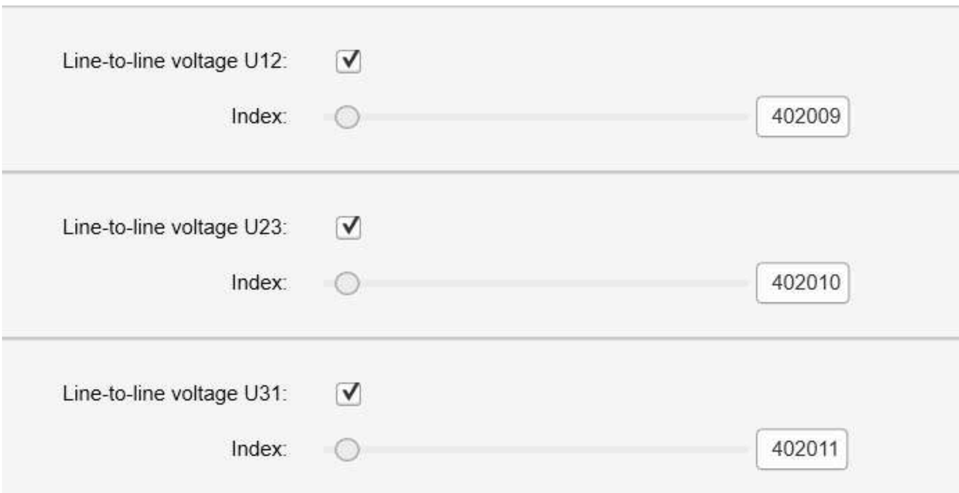
Some data items also have a deadband that needs to be defined. This deadband indicates the change in value needed before the device writes a new value to the server.

An example:

The default deadband setting for voltage is 20 V (refer to Figure Default voltage deadband of the *ModBusMstr* protocol, page 18). Let us consider the Line-to-line voltage U12, which has been observed at 12340 V and written to the server as the 'current' measurement value. If the measurement changes by more than 20 V, for example, to 12400 V, this change exceeds the deadband threshold. Consequently, the new measurement is written to the server, replacing the previous value and becoming the new 'current' measurement value.

Figure 14 - Default voltage deadband of the *ModBusMstr* protocol

Figure 15 - An example of a few voltage measurements



Modbus registers for protection, fault current and disturbance recordings

Registers for reading and writing protection settings

The protection settings can be read and written using Modbus registers starting from address 405001.

Modbus holding registers are 16 bits wide. However, some of the parameters which have been mapped to these registers in the device have 32-bit values. These parameters are mapped to two consecutive 16-bit holding registers, so that the register with the smaller address represents the Least Significant Bit (LSB) 16-bit word for the parameter and the second register represents the Most Significant Bit (MSB) 16-bit word for the parameter.

When the value of such parameter is updated through Modbus, the client must perform two write operations: one for the LSB word and the other for the MSB word. Each write operation updates the value of the parameter, so that only the word being written to is modified. The intermediate state for the parameter where only one word has been updated may represent an invalid value for the parameter. For example, the value may be out of allowed range. In such case the first write operation gives an error message, indicating that the value of the parameter was not updated. The client can nevertheless continue with the second write operation. This second write operation modifies the incomplete intermediate state of the parameter. If the intermediate state then reaches a valid 32-bit value for the parameter, the second write operation should indicate a success.

The client can, for example, first write the LSB 16-bit word for the parameter and then write the MSB 16-bit word for the parameter, and the parameter should have the new 32-bit value after the second write operation. The two 16-bit words can be written in any order. These two write operations should occur within two seconds of each other.

Registers for fault current

Here you can find explanations for fault current registers.

Table 9 - Modbus fault current registers

Register	Explanation
402110	Last fault current from any dir/undir OC stage (can be cleared)
402111	Fault current from I> stage
402112	Fault current from I>> stage
402113	Fault current from I>>> stage
402130	Last fault I_o current from any undirectional EF stage
402131	Fault current from I_o> stage
402132	Fault current from I_o>> stage
402133	Fault current from I_o>>> stage
402134	Fault current from I_o>>>> stage
402135	Fault current from I_o>>>>> stage
405505…405506	IL1 fault current
405507…405508	IL2 fault current
405509…405510	IL3 fault current
405511…405512	Io1 fault current

Table 9 - Modbus fault current registers (Continued)

Register	Explanation
405513…405514	Io2 fault current
405515…405516	IoCalc fault current
406550…406551	Last fault current
406552…406553	Fault current I>
406554…406555	Fault current I>>
406556…406557	Fault current I>>>
406558…406559	Last fault Io current
406560…406561	Fault current Io>
406562…406563	Fault current Io>>
406564…406565	Fault current Io>>>
406566…406567	Fault current Io>>>>
406568…406569	Fault current Io>>>>>

Registers for disturbance recording

COMTRADE is a standard file format for exchanging disturbance recordings. These are (oscillography) measurements of power system variables, such as (instantaneous) currents and voltages, over an event like a transient fault.

One recording consists of two files: a CFG file describes what was recorded in the recording, and a DAT file contains the recorded values.

In P3 devices, following registers mentioned in the Modbus registers for disturbance recordings, page 20 table is used to issue commands to the COMTRADE reading mechanism.

Table 10 - Modbus registers for disturbance recordings

Register	Content

405400 (Command register)	• MSB byte: Number of the recording to be downloaded, so that value zero means the oldest record, value one means the second oldest record, and so on.
	• LSB byte: A set of flags
	  ◦ b5: Trigger to remove oldest record
	  ◦ b4: Trigger to request the next chunk
	  ◦ b3: Flag for auto-increment the chunk
	  ◦ b2: Flag for load DAT file
	  ◦ b1: Flag for load CFG file
	• (LSB) b0: Trigger start of download when value 1 is written to this bit

405401 (Status register 1)	• MSB byte: Number of recording available
	• LSB byte: A set of flags
	  ◦ b5: Trigger for error in download
	  ◦ b4: Trigger for last chunk of data for this file
	  ◦ b3: Flag for chunk auto-increment in use
	  ◦ b2: Flag for DAT file now being loaded
	  ◦ b1: Flag for CFG file now being loaded

Table 10 - Modbus registers for disturbance recordings (Continued)

Register	Content
	• (LSB) b0: ◦ MSB byte: Number of the recording to be downloaded, so that value zero means the oldest record, value one means the second oldest record, and so on. ◦ LSB byte: A set of flags – b5: Trigger to remove oldest record – b4: Trigger to request the next chunk – b3: Flag for auto-increment the chunk – b2: Flag for load DAT file – b1: Flag for load CFG file ◦ (LSB) b0: Flag for valid data in data registers which implies that a download is in progress.
405402 (Status register 2)	• MSB byte: Number of bytes of data available in data registers • LSB byte: Approximate progress downloading current file in percentage
405403 to 405498 (96 Data registers)	Content (fragment) for the file being downloaded

NOTE: The maximum length of a Modbus message is 253 bytes. It is thus possible to read the two status registers and all of the data registers in a single command.

Reading disturbance records

Disturbance recordings (COMTRADE files) can be read via registers starting from *405400*.

1. Open Modbus connection to the device.
2. Read status register 1 (*405401*) and check that there is at least one recording available.
3. Construct a value for the command which starts the download process.

   For example, value *271 (0x10F)* starts the download of the second oldest record.

   MSB byte: *0x01*

   Number of the recording to be downloaded.

   LSB byte: *0x0F*

   - b5: (0) Trigger to remove the oldest record
   - b4: (0) Trigger to request the next chunk
   - b3: (1) Flag to auto-increment the chunk
   - b2: (1) Flag to load the DAT file
   - b1: (1) Flag to load the CFG file
   - (LSB) b0: (1) Trigger start of download when value 1 is written to this bit

   Write this value to the command register (*405400*). This command indicates which recording you want to read and whether you want to download both CFG file and DAT file, or only either one at a time. This enables *chunk auto-incrementing*, which means that whenever we read the value of status register 1, the contents of the data registers are updated to contain the next fragments of data for the file which is currently downloading.
4. Make a read request that starts from status register 1 (*405401*) and continues until the end of the data registers (registers *405403...405498*).

5. Inspect the value of status register 1.

Table 11 - Register *405401* (status register 1)

Byte / bit	Meaning	Explanation
MSB byte	Number of recording available	If this value changes in the middle of a download, a new recording has been made. It is recommended that in this case the download is aborted and restarted, by returning to step 3 or step 1.
(LSB) b5	Flag for error in download	If this flag is set, there was an internal error in the device, and the download is automatically aborted by the device. Contents of all other registers are invalid.
(LSB) b4	Flag for last chunk of data for this file	If this flag is set, the data registers contain the last pieces of data for the file which we are currently downloading. When next chunk is triggered, we get the first pieces of data for another file. (With auto-increment, this triggering happens when the status register 1 is read.)
(LSB) b3	Flag for chunk auto-increment in use	Indicates whether auto-increment is in use for this download. If this is not used, a separate write is needed to the command register, with correct flags, to get the next pieces of data in the data registers.
(LSB) b2	Flag for DAT file now being loaded	If set, indicates that we are now downloading the DAT file.
(LSB) b1	Flag for CFG file now being loaded	If set, indicates that we are now downloading the CFG file.
(LSB) b0	Flag for valid data in data registers	If this flag is set, the data registers DO contain valid data for the files being downloaded. Absence of this flag indicates either an error or that the download is complete.

6. Assuming that you now have valid data for the download, inspect the value of status register 2 (*405402*).

   MSB byte: Number of bytes of data available in data registers.

   LSB byte: Approximate progress downloading current file in percentage.

   > **NOTE:** The data registers hold 16-bit values (two bytes), while the amount of data is reported in bytes. The data registers should be filled with data (giving value *192* here), unless the last pieces of content for the file are now in the data registers.

   You can always read all data registers, and then ignore those which do not contain valid data, or first read the two status registers, and then, with a separate read command, read only those data registers which actually contain data. Also, you may choose to read only at most M registers at a time, so that M is smaller than the total number of data registers (96). These options may be useful if the used communication link is slow or expensive.

   If the link is prone to transfer errors, you can choose to read a given data register several times, and use the value which occurs most commonly. In this use case, it is recommended not to use the auto-increment feature.

7. Copy data from the data registers to the file which is currently being downloaded.

   Note the actual amount of (valid) data in the registers that was discovered in step 5. If this is the last chunk of data for this file, then after the data has been written to the file, close the file. When the next pieces of data are received, you can open a new file. Flags on status register 1 indicate which file is then being downloaded.

8. Start to read the next chunk by returning to step 4.

Client IP address filter

The Client IP address filter feature can be used to define the IP addresses that are accepted or rejected by the PowerLogic P3 when **IP firewall** is enabled in the eSetup Easergy Pro. The IP address filter is implemented for all communication protocols through TCP/IP.

Up to 10 IP addresses for the IP filter can be configured and enabled each rule individually on the eSetup Easergy Pro. Select **COMMUNICATION > Protocol configuration**, go to the **IP firewall** section, configure the filter rule as needed. More information, refer to the Configuring the IP filter section from *PowerLogic P3 Universal Relays P3U20 and P3U30 User Manual (P3U/EN/M/30-208A)*.

There are four types of IP address configurations supported in the **IP address** column.

Table 12 - IP address for the IP address filter

IP address	Description
Any	By writing a dash or value zero in this column, the rule is set to match any source IP address. The column shows a dash.
Single IP address	If a single IP address (such as 192.168.0.10) is written here, the packets (or connections) must originate from this IP address to match the rule.
IP subnet	If all IP addresses in a subnet should match this rule, write the subnet here using the CIDR (Classless Inter-Domain Routing) notation. For example, notation 192.168.0.0/24 matches all IP addresses in the range 192.168.0.0 – 192.168.0.255.
IP address range	If a range of IP addresses (for example, 192.168.0.20–192.168.0.30) is written here, packets from these addresses match the rule. Both end points of this range are inclusive.

NOTE: If the matching range of IP addresses can be expressed using the CIDR notation, the range is expressed in this format, regardless of how the range was entered into the configuration. As a result, the presentation format of the configuration as it is read from the device might not match the format in which it was entered. This may cause problems with eSetup Easergy Pro because this tool expects the presentation format to match exactly. To work around this issue, select the Reset and read current view command in eSetup Easergy Pro after writing the configuration. This is required to handle the large number of different input formats supported.

For each IP address, define the corresponding action in the **Action** column as needed. More information, refer to the Table *Actions for IP filter* of Configuring the IP filter section from *PowerLogic P3 Universal Relays P3U20 and P3U30 User Manual (P3U/EN/M/30-208A)*.

Revision history

Document version	Description	
P3/EN/CM/30-208A	Firmware version	V30.208
2025-03	Configuration tool	eSetup Easergy Pro 4.9.1 or later.
	Description of changes	Initial version

Schneider Electric
35 rue Joseph Monier
92500 Rueil Malmaison
France

+ 33 (0) 1 41 29 70 00

www.se.com

As standards, specifications, and design change from time to time, please ask for confirmation of the information given in this publication.

© 2025 – Schneider Electric. All rights reserved.

P3/EN/CM/30-208A