

Modbus/RTU - Network Slave

PLC500, PLC500ED, PLC500MC
PLC410

Application Note



Application Note

PLC410, PLC500, PLC500ED, PLC500MC

Document: 10010165520

Revision: 01

Publication Date: 12/2024

The information below describes the reviews made in this manual.

Version	Revision	Description
1.4.0	R01	Information update and inclusion of PLC models PLC410, PLC500ED and PLC500MC.
1.0.0	R00	First edition.

CONTENTS

1	INTRODUCTION	1-1
1.1	TERMS AND DEFINITIONS USED	1-1
1.2	NUMERIC REPRESENTATION	1-1
1.3	MODBUS/RTU DOCUMENTS	1-2
1.4	IMPORTANT NOTICE ABOUT CYBERSECURITY AND COMMUNICATIONS	1-2
1.5	TRADEMARKS	1-2
2	RS485 INTERFACE	2-1
2.1	RS485 SERIAL INDICATION	2-1
2.2	MODBUS/RTU NETWORK INSTALLATION	2-2
2.3	MODBUS/RTU NETWORK ARCHITECTURE EXAMPLE	2-3
3	PROJECT IN CODESYS	3-1
3.1	CREATING THE PROJECT	3-1
3.2	ADDING THE SLAVE DEVICE	3-2
4	MODBUS/RTU CONFIGURATION	4-1
4.1	TERMINATION RESISTORS	4-1
4.2	NETWORK CONFIGURATION	4-1
4.3	MODBUS I/O DATA CONFIGURATION	4-2
5	MONITORING	5-1
5.1	MONITORING VARIABLES	5-1
5.2	COMMUNICATION ERRORS	5-2

1 INTRODUCTION

This Application Note aims to assist in the use of the **Modbus/RTU** protocol on WEG PLCs, models PLC410, PLC500, PLC500ED, and PLC500MC. Throughout this document, the PLC500 is used as an example. However, the information presented is applicable to the other PLC models described earlier.

This document presents information about the Modbus/RTU protocol, description of the RS485 interface, installation recommendations, and a guide to establish Modbus/RTU communication between the PLC500 (slave) and the PLC300 (master) in **CODESYS®**.

For more information about the hardware, interfaces, and communication protocols, please refer to the User Manual of the respective product, available on the [WEG](#) website. For a more in-depth and detailed description of the operation of the Modbus/RTU network, access the online help at [CODESYS Online Help](#).

**ATTENTION!**

This application note is intended for professionals trained in industrial networks. The installation and configuration of devices should be performed according to the manufacturer's manual.

**NOTE!**

This application note is intended for the CODESYS programming tool. It is recommended to use **CODESYS V3.5 SP19** or higher. It is also recommended to use the latest available version of the configuration libraries for the Modbus/RTU network.

1.1 TERMS AND DEFINITIONS USED

CODESYS: Programming software for the PLC500. A programming platform that allows the development, configuration, and monitoring of solutions for industrial automation and system integration.

kbps: Kilobits per second.

m: meter.

ms: millisecond = 0.001 seconds.

s: second.

USB: stands for “*Universal Serial Bus*”, a type of serial communication interface designed to operate based on the “*Plug and Play*” concept.

1.2 NUMERIC REPRESENTATION

Decimal numbers are represented using digits without any prefix. Hexadecimal numbers are represented with the prefix '0x'.

1.3 MODBUS/RTU DOCUMENTS

It is recommended to consult the documents related to the Modbus/RTU network presented in [Table 1.1](#) on [page 1-2](#).

Table 1.1: Reference Documents.

Document	Version	Source
MODBUS Application Protocol Specification, December 28th 2006	V1.1b	MODBUS.ORG
MODBUS over Serial Line Specification, December 20th 2006.	V1.02	MODBUS.ORG
MODBUS Protocol Reference Guide, June 1996.	Rev. J	MODICON

1.4 IMPORTANT NOTICE ABOUT CYBERSECURITY AND COMMUNICATIONS

WEG PLCs, models PLC410, PLC500, PLC500ED, and PLC500MC, have the ability to connect and exchange information through networks and communication protocols. While they have been designed and tested to ensure proper operation with other automation systems using the protocols mentioned in this manual, it is essential that the customer understands the responsibilities associated with information and cybersecurity when using this equipment.

Therefore, it is the customer's full responsibility to adopt defense-in-depth strategies and implement policies and measures to ensure the security of the system as a whole, including communications sent and received by the equipment. These measures include, but are not limited to, the installation of firewalls, antivirus and antimalware programs, data encryption, authentication control, and physical access control for users.

WEG and its affiliates are not responsible for damages or losses arising from cybersecurity breaches, including, but not limited to, unauthorized access, intrusion, data leakage and/or theft, denial of service, or any other form of security violation. The use of this product in conditions for which it was not specifically designed is not recommended and may cause damage to the product, network, or automation system.

In this regard, it is crucial that the customer understands that external interventions through third-party programs, such as sniffers or programs with similar actions, have the potential to cause interruptions or restrictions in the functionality of the equipment.

1.5 TRADEMARKS

All other registered trademarks are the property of their respective owners.

2 RS485 INTERFACE

Modbus/RTU communication is performed through the RS485 interface, as shown in [Figure 2.1 on page 2-1](#) for the PLC500 and PLC410. The pinout of the interface is described in [Table 2.1 on page 2-1](#).

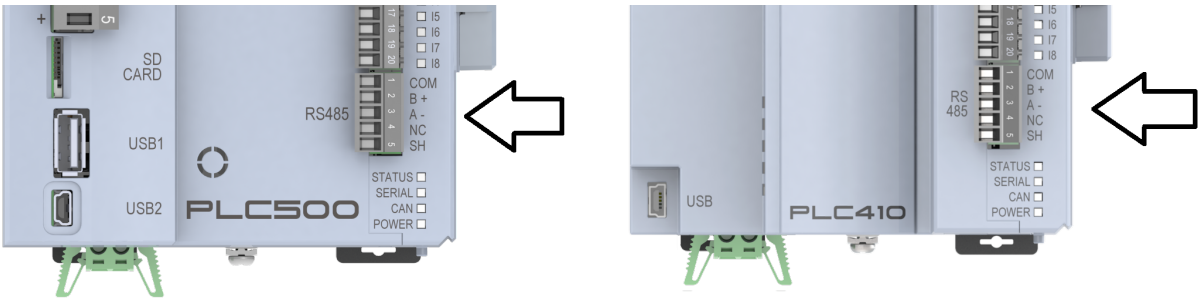


Figure 2.1: RS485 connector pinout on the PLC500 and PLC410.

Table 2.1: RS485 interface description.

Pin	Label	Function
1	COM	Common (connected to the negative terminal of the Modbus/RTU network)
2	B+	B+ communication signal
3	A-	A- communication signal
4	NC	Not connected (may receive the positive terminal of the Modbus/RTU network)
5	SH	Cable shielding

The RS485 interface is isolated with its own internal power supply, meaning the pin 4 (NC) does not need to be connected. However, it is recommended that pin 1 (COM) be connected to the master and other slaves to maintain the same voltage reference for the Modbus/RTU network.

The PLC410, PLC500, PLC500ED, and PLC500MC allow connection of up to 246 Modbus/RTU slaves. However, for connecting more than 32 devices to the RS485 interface, a repeater must be used.

2.1 RS485 SERIAL INDICATION

The PLC500 and PLC410 have LEDs on the front panel used to indicate the status of the RS485 (SERIAL) interface. The LED color indications shown in [Figure 2.2 on page 2-1](#) are described in [Table 2.2 on page 2-1](#).

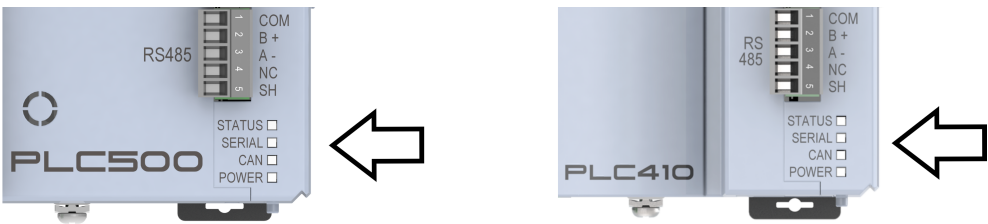


Figure 2.2: LED indicators of the PLC500 and PLC410.

Table 2.2: Operation of the RS485 interface indicator LED.

SERIAL LED	STATUS	DESCRIPTION
OFF	No configuration.	There is no Modbus/RTU communication configuration in the application.
GREEN	Modbus/RTU communication without failure.	Modbus/RTU communication configured in the application and all elements running without errors/failures.
ORANGE	Modbus/RTU communication with partial failure.	Modbus/RTU communication configured in the application with some elements having errors/failures.
RED	Modbus/RTU communication with total failure.	Modbus/RTU communication configured in the application and all elements having errors/failures.

2.2 MODBUS/RTU NETWORK INSTALLATION

The Modbus/RTU network, like many industrial communication networks, due to its frequent use in harsh environments with high electromagnetic interference, requires certain precautions to ensure a low communication error rate during its operation.



ATTENTION!

For passive components of the network such as cables, connectors, termination resistors, and power supplies, it is recommended to use only components certified for industrial applications.

The RS485 interface of the PLC500 programmable logic controller can communicate using the rates defined in [Table 2.3 on page 2-2](#).

Table 2.3: Supported RS485 communication rates.

Communication Rate
1200 kbps
2400 kbps
4800 kbps
9600 kbps
19200 kbps
38400 kbps
57600 kbps
115200 kbps

For connecting the **A-** and **B+** signals, shielded twisted pair (STP) cables should be used. For equipment requiring external power for the RS485 interface, the positive and negative pair should also be twisted.

The maximum cable length depends on the communication rate. For a rate of 9600 kbps and an AWG 26 (or larger) wire gauge, the maximum length is 1000 meters. An AWG 24 wire is sufficient for Modbus data in most cases. Branches should be no longer than 20 meters.

For interconnecting the various network nodes, it is recommended to connect the equipment directly to the main line, without using branches. During cable installation, the cables should not be routed near power cables, as this increases the likelihood of transmission errors due to electromagnetic interference.

An adequate connection to the grounding system is essential to reduce electromagnetic interference in industrial environments. It is important to avoid grounding the cable at multiple points, especially where there are grounds with different voltage potentials.



ATTENTION!

Grounding the negative terminal of the PLC500 power supply and pin 1 of the RS485 interface may cause damage to the device due to loss of isolation in the power supply.



DANGER!

Defective grounding installations can lead to death and malfunction of the Modbus/RTU network.

2.3 MODBUS/RTU NETWORK ARCHITECTURE EXAMPLE

In this application note, an example is presented with a PLC500 as a network slave and a PLC300 as a network master. The [Figure 2.3 on page 2-3](#) shows the components and architecture of the Modbus/RTU network. In this scenario, a computer programs the PLC500 and PLC300 via USB cables. The configuration of the PLC500 is done in CODESYS, while the PLC300 configuration is carried out in the **Weg Programming Suite (WPS)**.

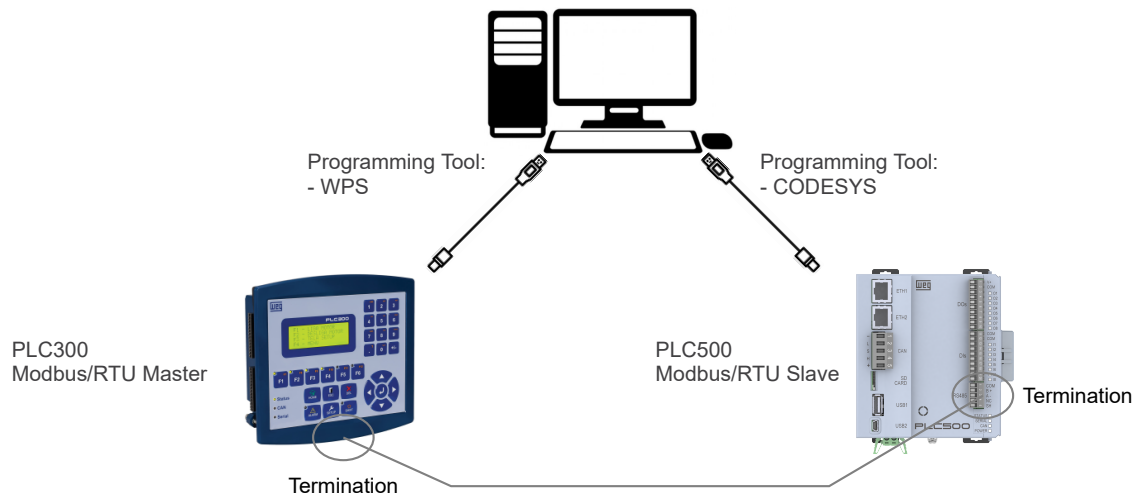


Figure 2.3: Components of the Modbus/RTU network example.



NOTE!

For more information on programming the PLC300 and using WPS, refer to the product documentation available on the [WEG](#) website.

3 PROJECT IN CODESYS

In this chapter, the steps for creating a project and adding the Modbus/RTU network slave device in CODESYS are presented.

3.1 CREATING THE PROJECT

To configure and define a Modbus/RTU network, you must first create the project and include the PLC500 programmable logic controller. In the CODESYS software, create a new project using the **Standard project** template, choose the directory and the application name. Then, select the PLC500 device and the desired programming language, as shown in [Figure 3.1 on page 3-1](#).

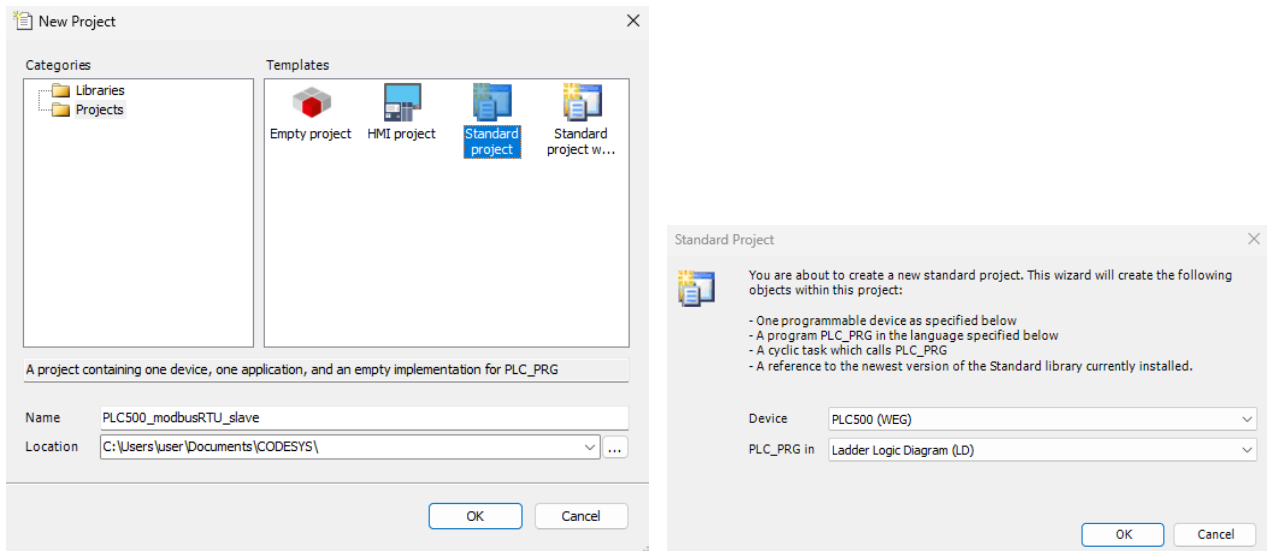


Figure 3.1: Creating the project in CODESYS.



NOTE!

If the PLC410, PLC500, PLC500ED, or PLC500MC devices are not available in the CODESYS options, you must download and install the configuration file. Please check the **Product Manual** available on the [WEG](#) website for the necessary steps and configurations.

With the PLC500 device selected, the project tree will appear as shown in [Figure 3.2 on page 3-1](#).

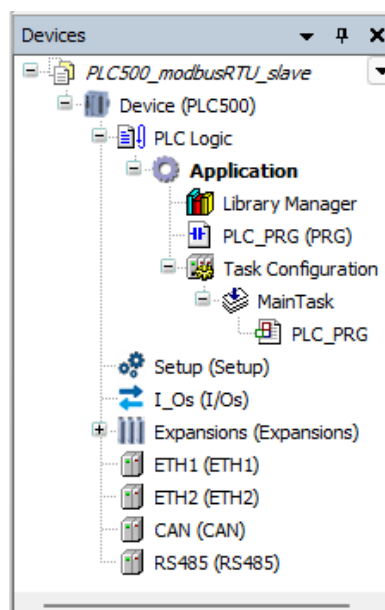


Figure 3.2: Initial project tree of the PLC500.

3.2 ADDING THE SLAVE DEVICE

The next step is to add the **Modbus_Serial_Device**. Right-click on the **RS485** icon and select **Add Device**. A new window will open. Expand the icons **Fieldbuses**, **Modbus**, and **Modbus Serial Device** by clicking on the + symbol until the **Modbus_Serial_Device** appears, as shown in [Figure 3.3 on page 3-2](#). With the device selected, click on **Add Device**.

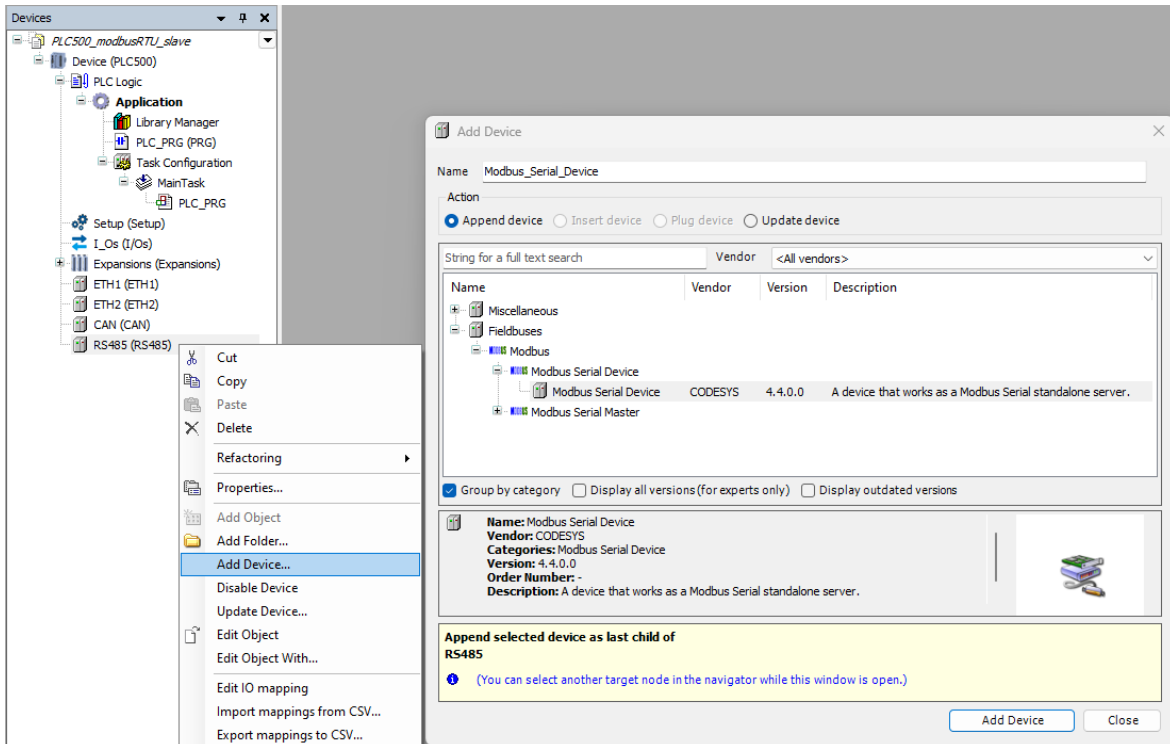


Figure 3.3: Adding the Modbus Serial Device.

The device will be added to the project tree just below the **RS485** interface, as demonstrated in [Figure 3.4 on page 3-2](#).

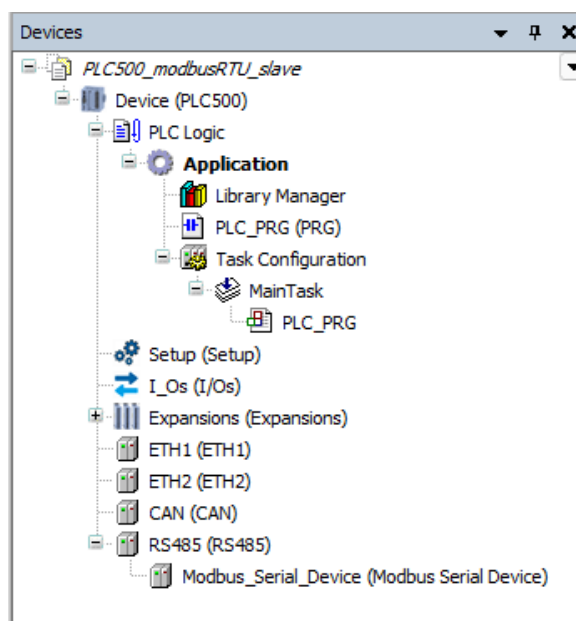


Figure 3.4: Modbus_Serial_Device added to the project tree.

4 MODBUS/RTU CONFIGURATION

In this chapter, the steps for configuring a Modbus/RTU network with a PLC500 slave in CODESYS are presented.

4.1 TERMINATION RESISTORS

For the correct operation of the Modbus/RTU network, it is essential to use termination resistors. You can choose to use an external resistor or enable the internal resistor directly through CODESYS. In the PLC500, the internal resistor can be configured through the **Setup** → **Setup Parameters** → **Termination Resistors** → **RS485** object, as shown in [Figure 4.1 on page 4-1](#).

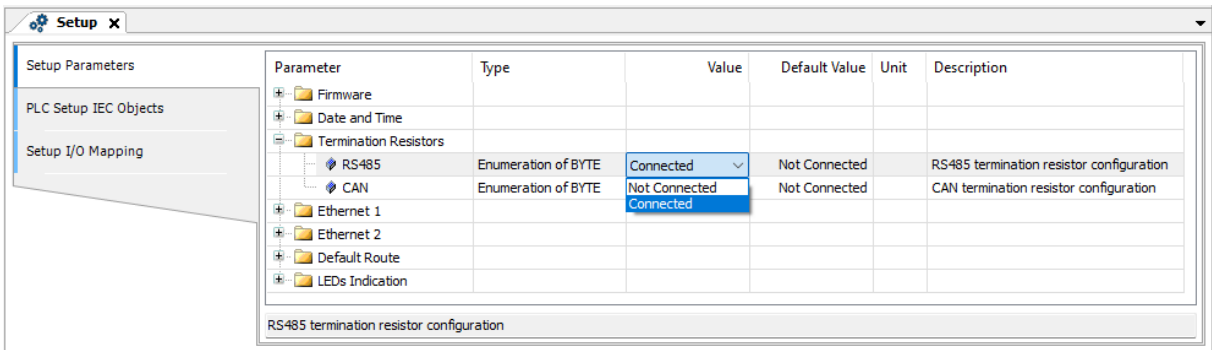


Figure 4.1: Termination resistors configuration page.



NOTE!

In the PLC300, the internal resistor can be added through switch S1. For more information, refer to the product documentation available on the [WEG](#) website.

4.2 NETWORK CONFIGURATION

The network must be configured on the master device to ensure that the slave addresses are correctly addressed. It is necessary to define the parameters and addresses that will be transmitted from the master. Each slave device must be configured with a unique address and must operate at the same baud rate; otherwise, communication failures may occur on the network. Double-clicking on **RS485** will define the slave device configurations, as shown in [Figure 4.2 on page 4-1](#).

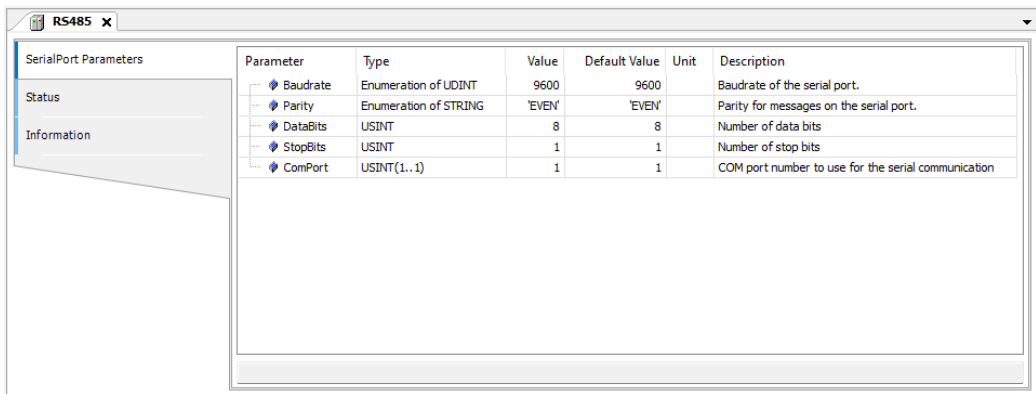


Figure 4.2: Slave device communication settings.



ATTENTION!

Only use **ComPort 1** for the PLC500; using another port will cause a communication error.

4.3 MODBUS I/O DATA CONFIGURATION

For each application, the transmission addresses and relevant data can be defined. As an example, the PLC300 writes a value to the digital outputs and reads the value from the digital inputs of the PLC500 every 100 ms. In [Figure 4.3 on page 4-2](#), the schematic representation of this network configuration is shown.

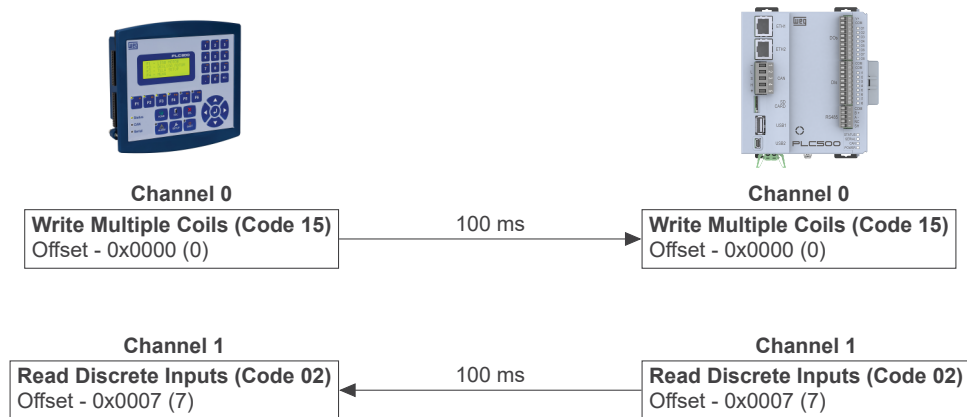


Figure 4.3: Schematic representation of Modbus channel configuration.

Double-clicking on the **Modbus_Serial_Device** icon opens the window shown in [Figure 4.4 on page 4-2](#). In this window, the slave addressing can be changed using the **Unit ID** parameter. The **Watchdog** option is activated with a 500 ms timeout. Next, enable the **Discrete Bit Areas** box and set the number of **Coils** and **Discrete Inputs** to 8. In the **Start Addresses** settings, the coil address is set to start at 0, while the discrete input address is configured to start at 7.

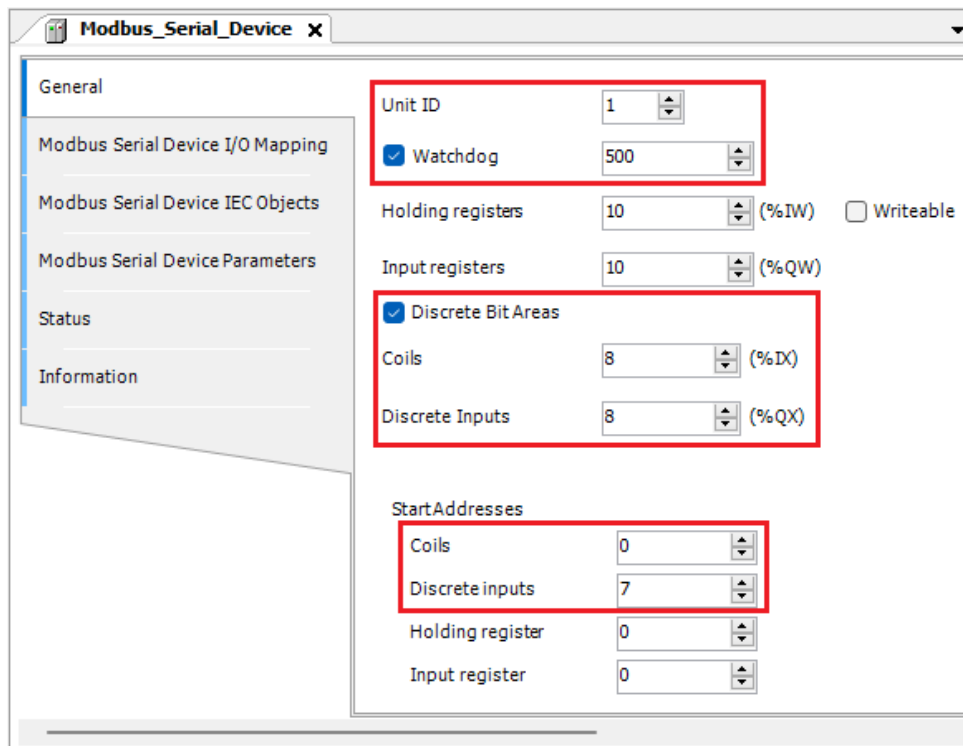


Figure 4.4: I/O data configuration for the Modbus/RTU slave device.

The number of coils and discrete inputs declared can be viewed in the **Modbus Serial Device I/O Mapping** tab. In the case of the PLC500, the I/Os do not have fixed Modbus addresses. Therefore, to access the digital inputs and outputs of the device, variable mapping using auxiliary variables in the application must be done. In [Figure 4.5 on page 4-3](#), the mapping of coils and discrete inputs using these auxiliary variables is shown.

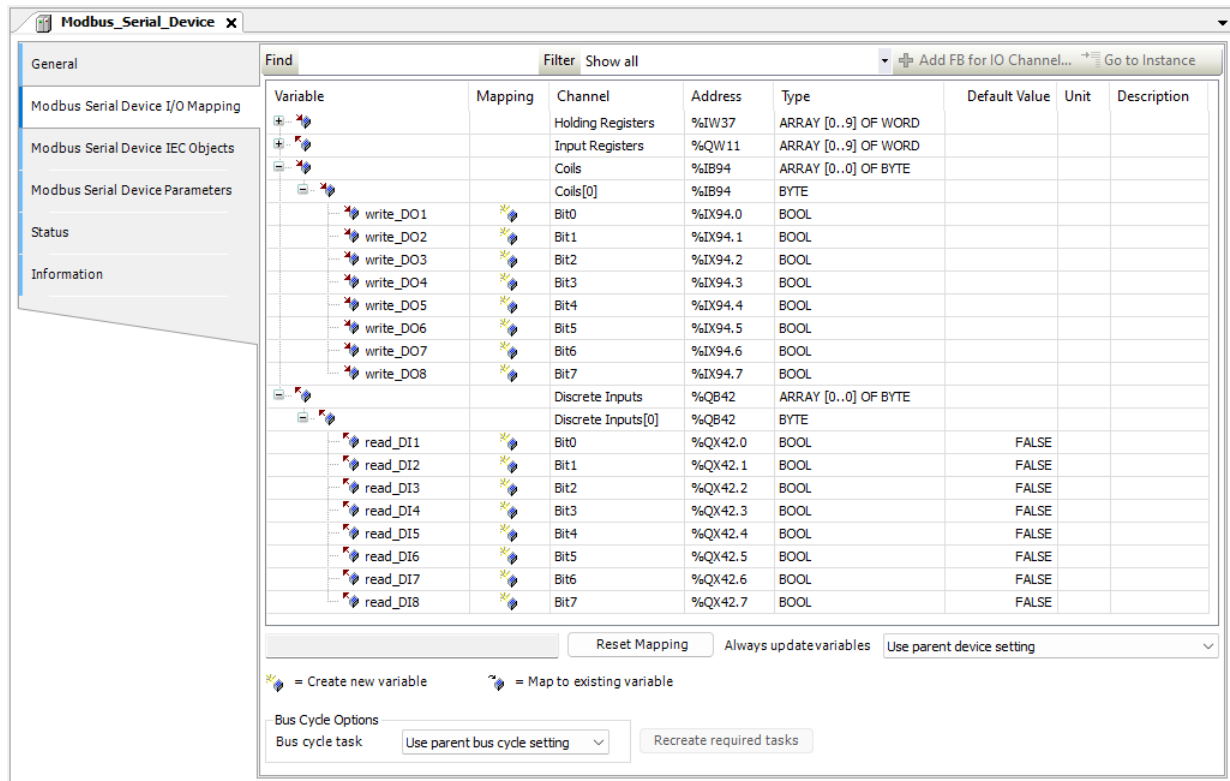


Figure 4.5: Declared and mapped variables for the Modbus/RTU slave device.

To link the Modbus variables with the I/Os mapped with their default names on the PLC500 (DO1, DO2 ... DI1, DI2 ...), a simple structured text code is shown in [Figure 4.6 on page 4-3](#).



ATTENTION!

Do not alter the default names of the I/O variables that were previously declared. When updating the device to a different model and/or version, **the variables will return to their original declaration.**

IO_mapping - Structured Text (ST)

```

read_DI1 := DI1;
read_DI2 := DI2;
read_DI3 := DI3;
read_DI4 := DI4;
read_DI5 := DI5;
read_DI6 := DI6;
read_DI7 := DI7;
read_DI8 := DI8;

DO1 := write_DO1;
DO2 := write_DO2;
DO3 := write_DO3;
DO4 := write_DO4;
DO5 := write_DO5;
DO6 := write_DO6;
DO7 := write_DO7;
DO8 := write_DO8;

```

Figure 4.6: IO_mapping program in structured text.



ATTENTION!

For larger applications, where the state of the digital inputs and outputs of the PLC500 is crucial, it is recommended that input variables be read at the start of the main program, while output variables should be written at the end. For more information, consult **Bus Cycle Task** in the online help at [CODESYS Online Help](#).

5 MONITORING

This chapter presents some methods for monitoring Modbus/RTU network variables and diagnosing communication errors in CODESYS.

5.1 MONITORING VARIABLES

The declared variables in the network can be monitored through two methods. The first method is to add them to the CODESYS program and monitor the variable values online. The second method only requires enabling **Always Update Variables** in the footer of the **Modbus Serial Device I/O Mapping** tab, as indicated by the red arrow in [Figure 5.1 on page 5-1](#).

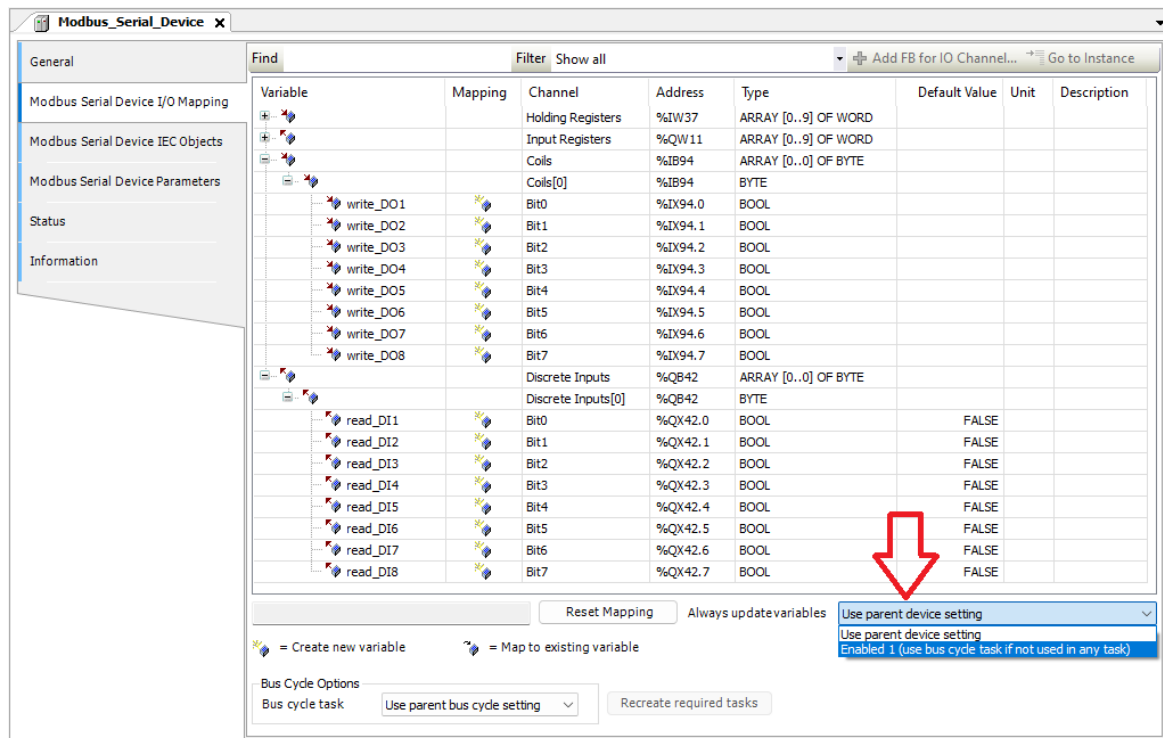


Figure 5.1: Option to always update Modbus/RTU variable values with bus cycle task.

[Figure 5.2 on page 5-1](#) shows the CODESYS variable visualization screen for the Modbus/RTU slave through online monitoring.

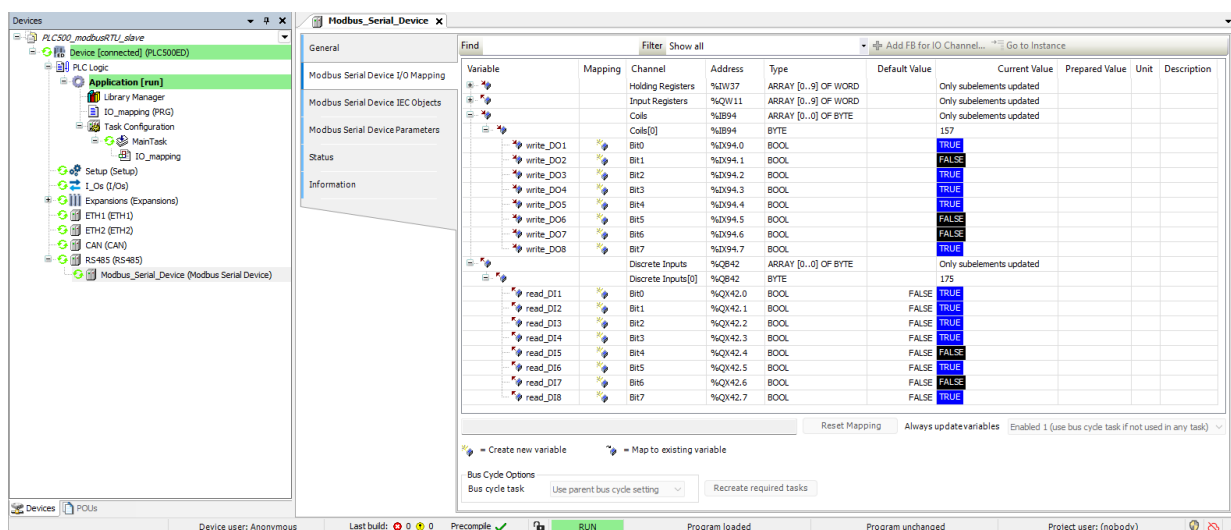


Figure 5.2: Monitoring Modbus/RTU variables online in CODESYS.

5.2 COMMUNICATION ERRORS

Network status monitoring can be done in **Devices**, indicating the status of each communication step and reporting the status (**Status**). When connection issues are encountered, as shown in [Figure 5.3 on page 5-2](#), check if the cables are properly connected, the state of the SERIAL LED, and review the configurations. In this case, the cable was not properly connected to the slave device.

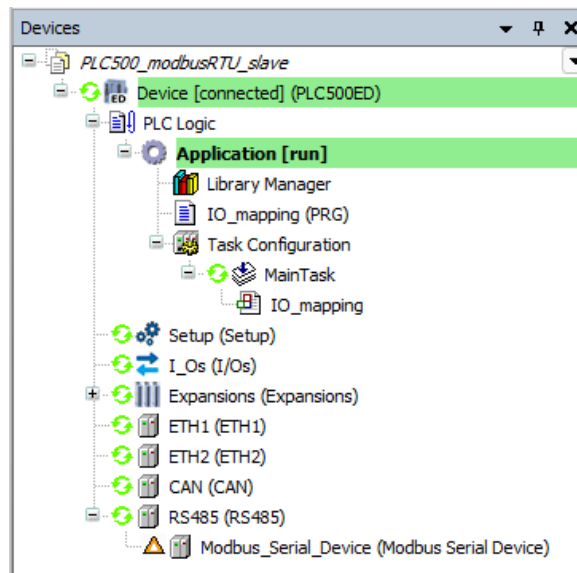


Figure 5.3: Indication of a communication error.

After reconnecting the cable and waiting for a few moments, the network communication is restored, and the device icons will be displayed in green, as shown in [Figure 5.4 on page 5-2](#).

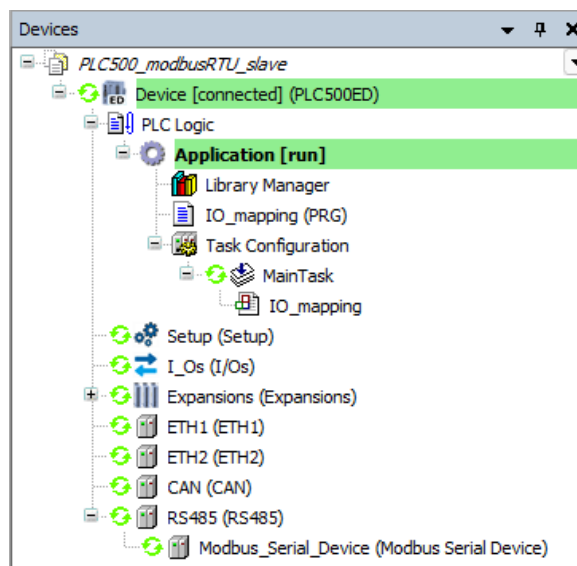


Figure 5.4: Correctly configured communication and devices communicating.



NOTE!

It is recommended to consult the documentation available in the **Library Manager**, or access the online help at [CODESYS Online Help](#), for more information on configuring the Modbus/RTU network and available functions.



BRAZIL

WEG DRIVES & CONTROLS - AUTOMAÇÃO LTDA.

Av. Prefeito Waldemar Grubba, 3000

89256-900 - Jaraguá do Sul - SC

Phone: 55 (47) 3276-4000

Fax: 55 (47) 3276-4060

www.weg.net/br