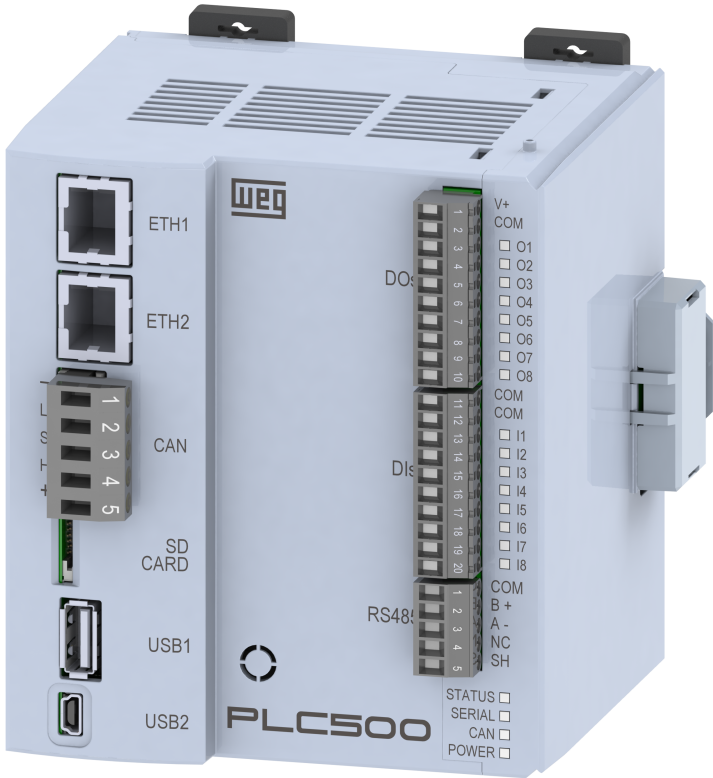


# Modbus/TCP - Network Slave

## PLC500

### Application Note





# **Slave Modbus/TCP - Application Note**

Series: PLC500

Language: English

Document: 10010165911 / 00

Publication Date: 10/2022

The information below describes the reviews made in this manual.

<b>Version</b>	<b>Revision</b>	<b>Description</b>
-	R00	First edition

## Contents

<b>1 INTRODUCTION</b> .....	<b>5</b>
1.1 MODBUS/TCP NETWORK COMPONENTS .....	5
1.2 ETHERNET INTERFACE .....	5
1.3 NETWORK ARCHITECTURE .....	5
<b>2 MODBUS/TCP SETTINGS</b> .....	<b>7</b>
2.1 CREATING A PROJECT IN CODESYS .....	7
<b>3 SLAVE MODBUS/TCP SETTINGS</b> .....	<b>9</b>
3.1 NETWORK SETTINGS .....	9
3.2 I/O SETTINGS .....	9
<b>4 MONITORING</b> .....	<b>12</b>
4.1 VARIABLE MONITORING .....	12
4.2 COMMUNICATION ERRORS .....	12

# 1 INTRODUCTION

This application note describes how to operate the PLC500 to communicate in Modbus/TCP network. Here, we will show the settings for **network slave**. For the PLC500 product settings, follow the steps described in this document to configure the Modbus/TCP network properly.

**ATTENTION!**

This application manual is intended for professionals trained in industrial networks. The devices must be installed and configured according to manufacturer's manual.

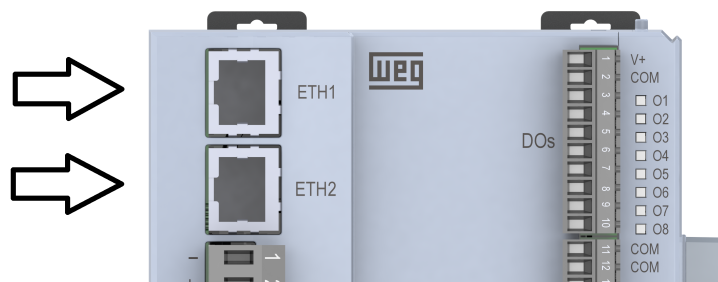
## 1.1 MODBUS/TCP NETWORK COMPONENTS

For the network passive components — cables, connectors, power supply, switches, hubs — it is recommended to use only components certified for industrial applications. See the product documentation for information on the proper installation of the Ethernet network.

For a detailed description of the network operation Modbus/TCP and its settings, visit Codesys webpage at: <https://help.codesys.com/>.

## 1.2 ETHERNET INTERFACE

As indicated in Figure 1.1, Ethernet connections are used for Modbus/TCP communication. Initially, each port has the IPv4 address contained in Table 1.1, which can be changed at any time using the *Codesys software* or via website.



**Figure 1.1:** Ethernet interface connector.

**Table 1.1:** Start address for each Ethernet port.

Connection	Default IPv4 address
ETH1	192.168.1.10
ETH1	192.168.2.10

## 1.3 NETWORK ARCHITECTURE

In figure 1.2, the devices are programmed by the computer and communicate with each other through the Ethernet network switch. For the PLC500 product settings, follow the steps described in this document to configure the Modbus/TCP network properly.

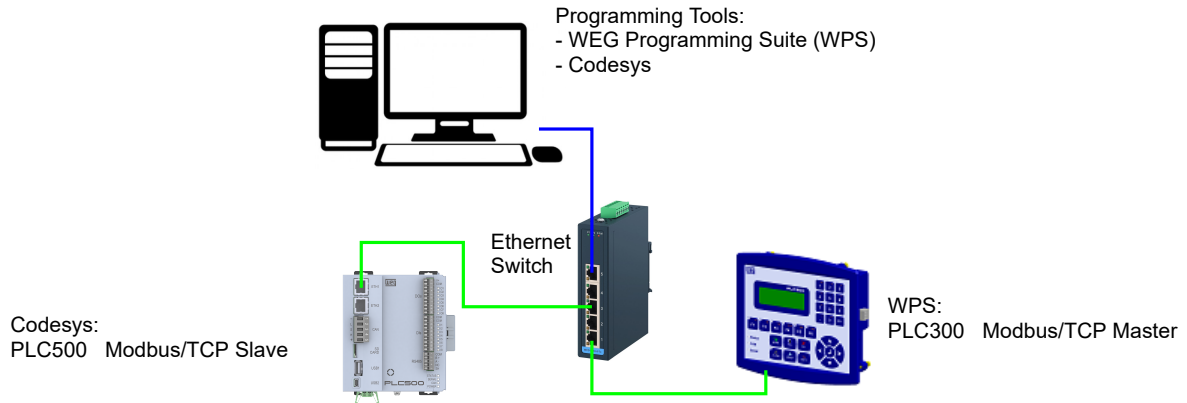


Figure 1.2: Network components.



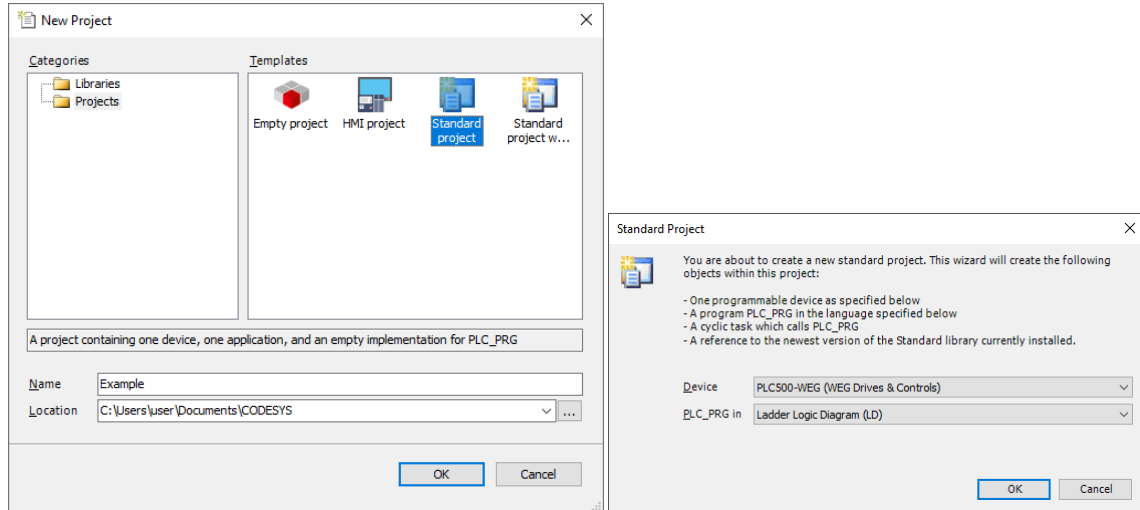
**NOTE!**

This application manual is intended for the PLC500 and the **Codesys** programming tool, we recommend using the **Codesys V3.5 SP18 or higher**. If you need more information about the Modbus/TCP communication protocol, see its manuals.

## 2 MODBUS/TCP SETTINGS

### 2.1 CREATING A PROJECT IN CODESYS

In the Codesys *software*, create a new project and choose the directory and application name. Then select the PLC500-WEG device and the desired programming language, as shown in Figure 2.1.



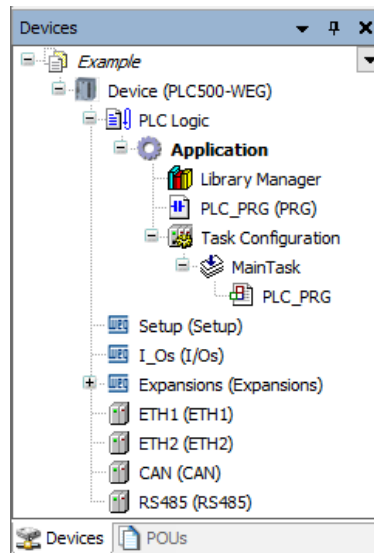
**Figure 2.1:** Project configuration in Codesys.



**NOTE!**

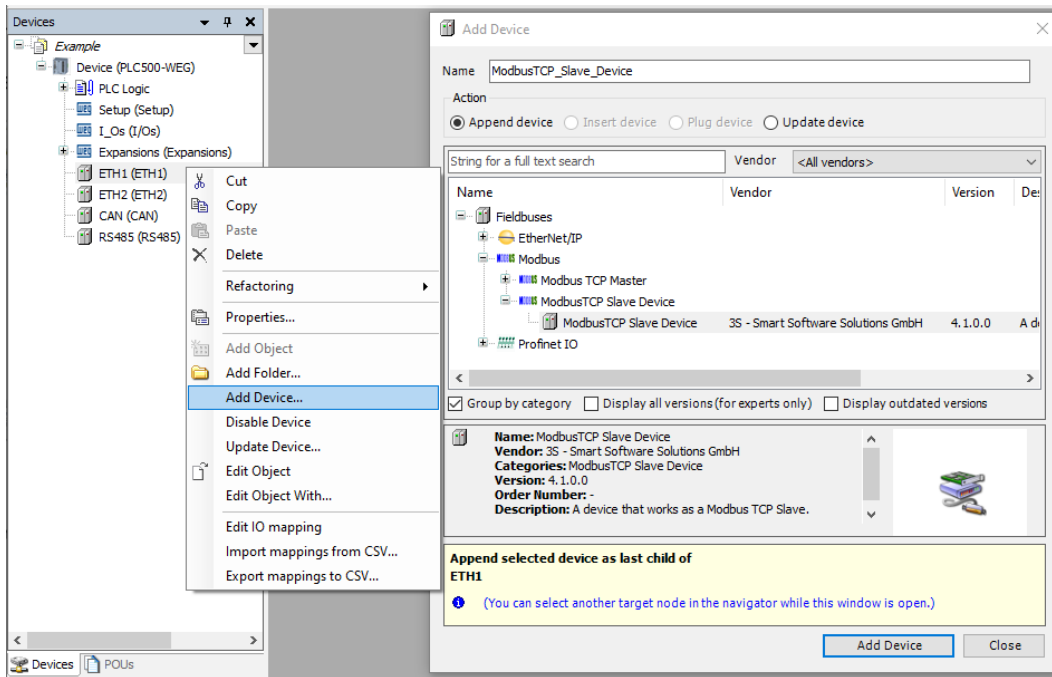
In case the PLC500 device is not available in the Codesys options yet, you must download and install the configuration file. See the **Product Manual** for the necessary steps and settings.

With the PLC500 device selected, you will have a project with the available networks already preset, as shown in Figure 2.2.



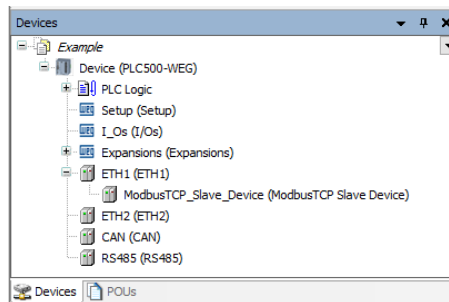
**Figure 2.2:** Network interfaces for the PLC500.

On the **ETH1** or **ETH2** interface, add **"Modbus TCP Slave"**, as shown in Figure 2.3.



*Figure 2.3: Codesys - Settings Modbus/TCP Step 1.*

At this moment, the chosen **ETH** interface should have the items indicated in Figure 2.4.



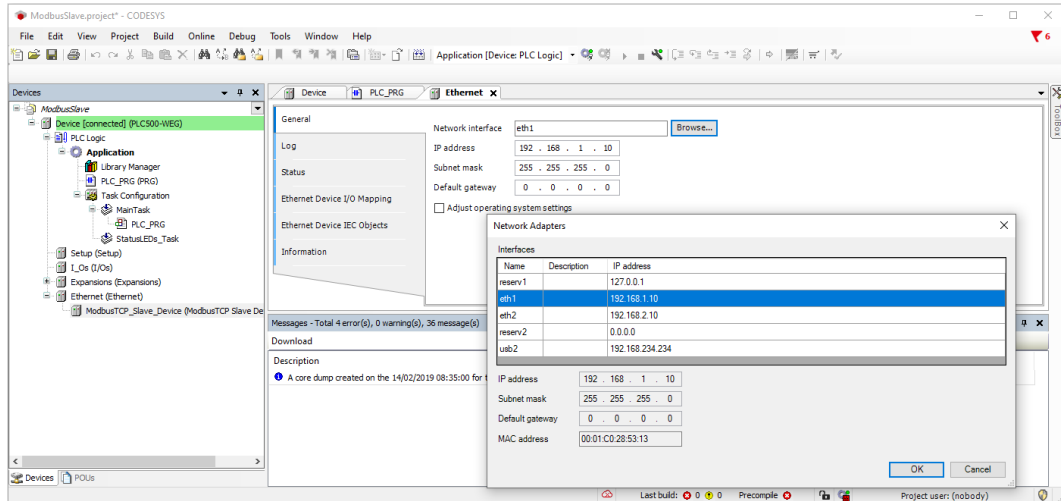
*Figure 2.4: Codesys - Settings Modbus/TCP Step 2.*



## 3 SLAVE MODBUS/TCP SETTINGS

### 3.1 NETWORK SETTINGS

Define a unique addresses for each slave and select the allowed IPv4 address range; otherwise, the network will have problems. In Figure 3.1, the master device settings are defined, selecting the interface to configure the IPv4 address and the other requirements.



**Figure 3.1:** Network settings for the device programmed in Codesys.

After this step, the network is configured for the slave device. Each one must have its own address and be within the available IPv4 address range. The other settings, such as baud rate, will be managed automatically by the network.

### 3.2 I/O SETTINGS

Before declaring the network variables, select the **Offset** value of the addresses. The addresses can be defined independently for each function, as you can see in **Start Addresses** inside the red rectangle in Figure 3.2.

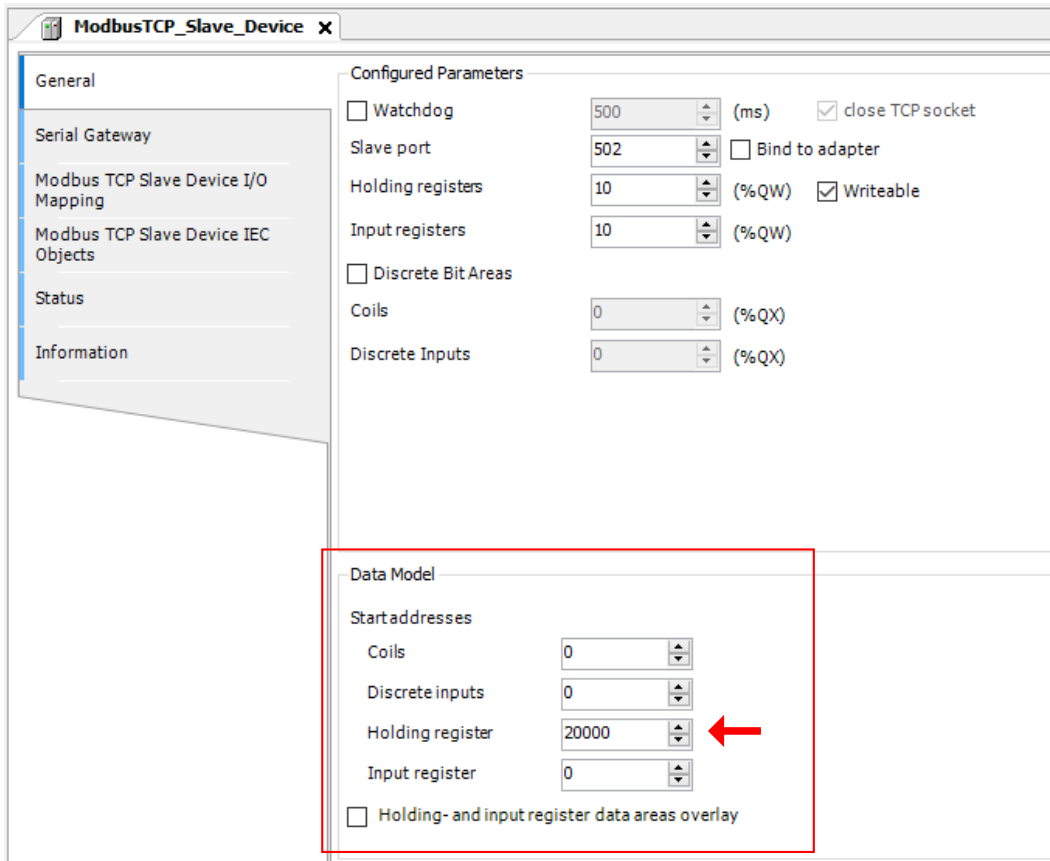


Figure 3.2: Declaration of variables transmitted by the network slave device.

In this step, we declare the memory addresses reserved for the transmission variables, reading a value in the network slave register and writing in another. Figure 3.3 below shows the Modbus functions for reading and writing these data to the slave device.

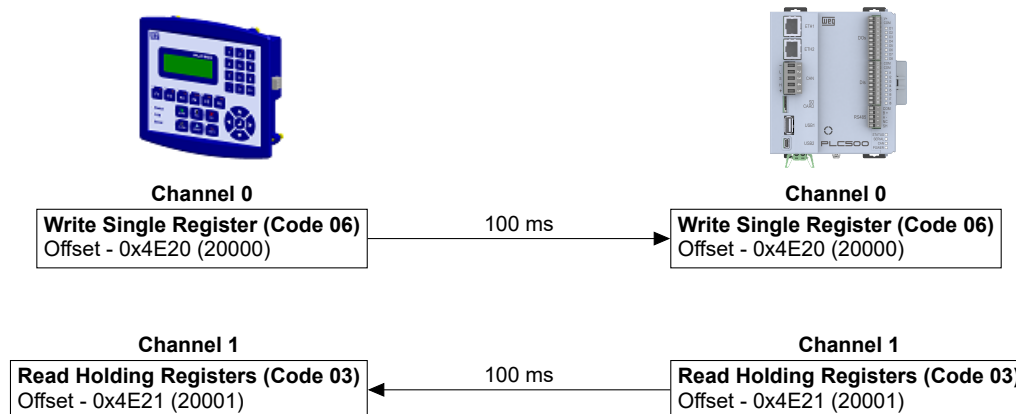


Figure 3.3: Transmission Channel Settings.

The number of registers and coils declared in Figure 3.3 are indicated on the page in Figure 3.4. Take into account the difference between the **registers** and **coils**—the coils are in separate addresses and must be enabled with **Discrete Bit Areas**.

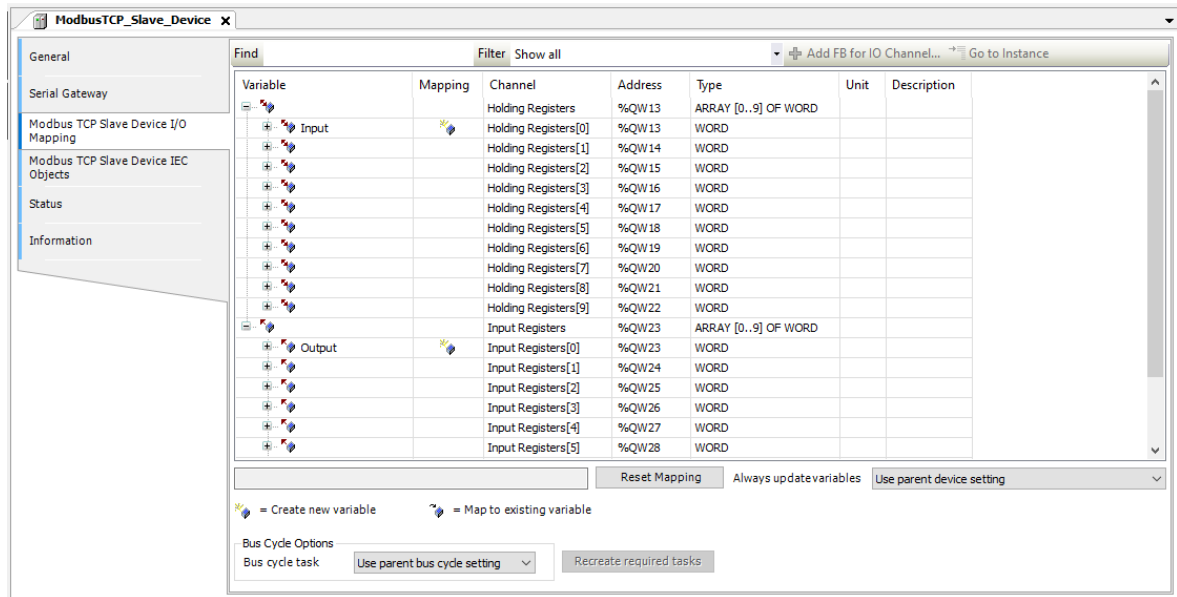


Figure 3.4: Declaration of addresses transmitted by the network slave device.

The variables declared in the network can be monitored through two methods: first, to add the variables to the **Codesys** program and monitor their values online; second, to enable **Always Update Variables** at the bottom of the previous page, as indicated by the red arrow in Figure 3.5.

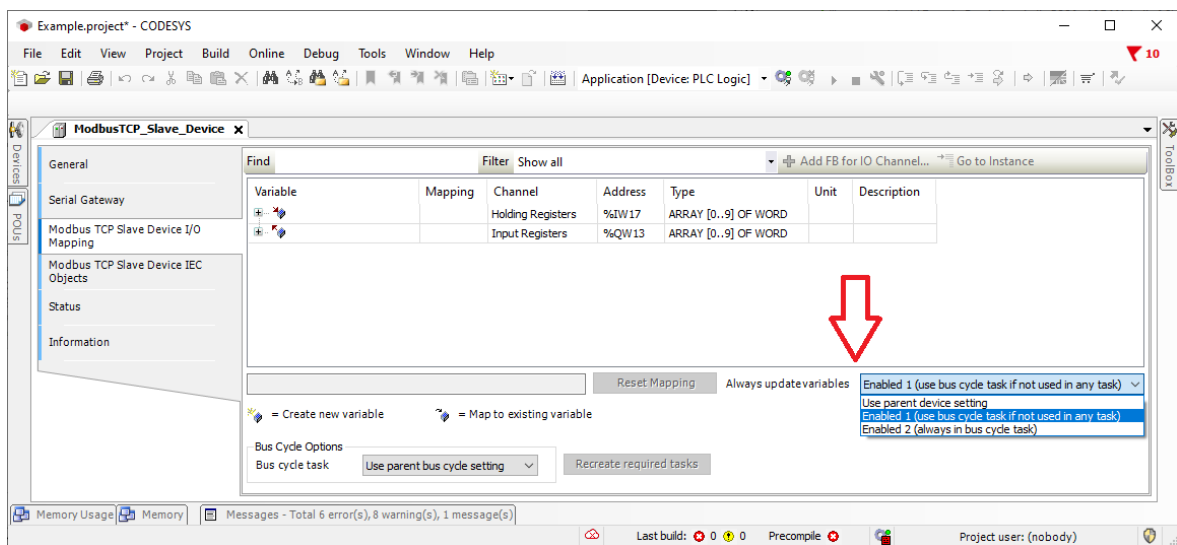


Figure 3.5: Option to show updated values of variables all the time.

## 4 MONITORING

### 4.1 VARIABLE MONITORING

After setting the Modbus/TCP network and declaring the transmission addresses, we can monitor and control the data transmitted by the devices. Finally, we must declare the variables that we want to assign to the addresses. On the *online* mode, **Mapping** tab, it is possible to check and write the values in the network variables.

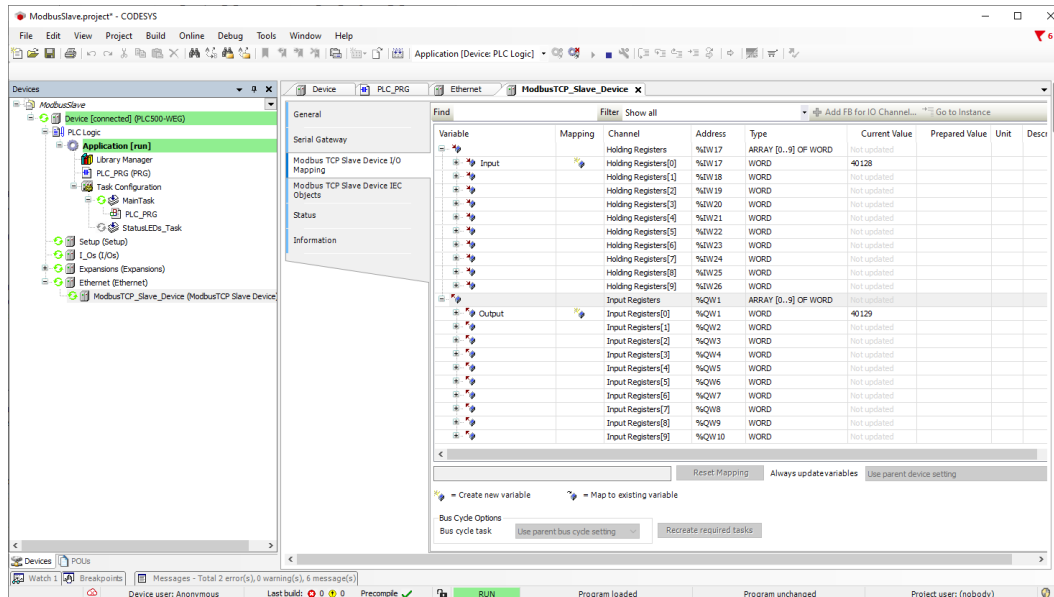


Figure 4.1: Declaration of variables transmitted by the network slave device.

### 4.2 COMMUNICATION ERRORS

The status of the networks in the Codesys software PLC500 can also be monitored in **Devices**, which indicates the status of each communication step and the **Status**. If you encounter connection problems, connect to the PLC500 and access the **Status** and **Log** tab within the created items; Codesys will report the problem hindering the communication.

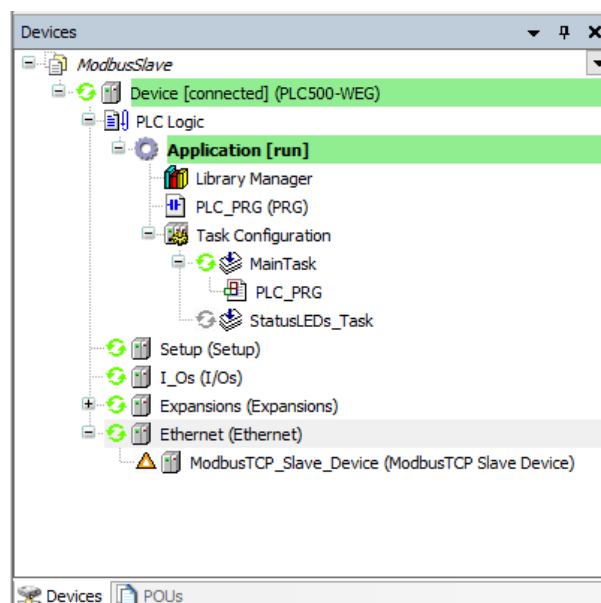


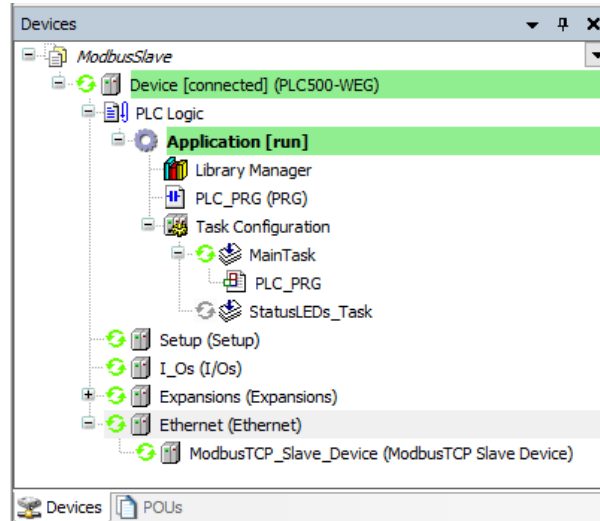
Figure 4.2: Communication error indication.



**NOTE!**

If you encounter problems, check that the cables are properly connected and that the respective network LED is turned on.

When the settings are correct and the devices identify each other, the application should show as in Figure 4.3.



**Figure 4.3:** Communication correctly configured and devices communicating.



WEG Drives & Controls - Automation LTDA.  
Jaraguá do Sul - SC - Brazil  
Phone 55 (47) 3276-4000 - Fax 55 (47) 3276-4020  
São Paulo - SP - Brazil  
Phone 55 (11) 5053-2300 - Fax 55 (11) 5052-4212  
[automacao@weg.net](mailto:automacao@weg.net)  
[www.weg.net](http://www.weg.net)