

Setting Up a Redundant Controller Architecture with UC20 Controllers and Modbus TCP I/O

Abstract:

This application note describes the setup of a redundant controller architecture using two UC20-M4000 controllers and distributed UR20 Modbus TCP fieldbus couplers.

It covers the configuration of the CODESYS controller redundancy, the redundant Modbus TCP communication, and an RSTP-based Ethernet network to provide a highly available control system.

Finally, the document demonstrates the validation of the complete setup through controller switchover and network failover tests.

Hardware reference

No.	Component name	Article No.	Hardware / Firmware version
1	UC20-M4000	2839160000	HW 01.00.00
2	UR20-FBC-MOD-TCP-V2	2476450000	HW 01.05.00
3	IE-SW-AL05LM-5TX	2682250000	HW 01.00 FW 01.19

Software reference

No.	Software name	Article No.	Software version
1	u-OS		2.5.0
2	CODESYS Development System		3.5 SP21 Patch 4 - 3.5.21.40
3	UR20-FBC-MOD-TCP-V2 Firmware		02.16.00
4	CODESYS Control SL for Weidmüller u-OS Runtime App		4.21.0.0-1
5	CODESYS Control SL for Weidmüller u-OS Package		4.21.0.0
6	IoDrvUR20Control		2.1.0.1
7	RedundancyReferenceApplication		3.0.0.0
8	CODESYS Redundancy Configuration		4.3.0.0
9	libWiUtils		3.0.0.0
10	libWilpAlias		2.0.0.0
11	Runtime_identification		3.5.22.20
12	Redundancy Implementation		3.5.22.20

Contact

Weidmüller Interface GmbH & Co. KG
 Klingenbergstraße 26
 32758 Detmold, Germany
www.weidmueller.com

For any further support please contact your local sales representative:
<https://www.weidmueller.com/countries>

Content

1	Warning and Disclaimer.....	4
2	Introduction.....	5
3	System Architecture.....	6
3.1	System Overview.....	6
3.2	Network Topology.....	7
3.3	CODESYS Redundancy Principle (System Redundancy).....	8
3.4	Media Redundancy.....	9
4	Set up the fieldbus coupler UR20-FBC-MOD-TCP-V2.....	10
5	Setting Up CODESYS Controller Redundancy	18
5.1	Preparing the UC20-M4000 Controllers.....	18
5.2	Installing the CODESYS runtime on u-OS	23
5.3	Creating the CODESYS Project.....	26
5.4	Configuring Redundancy Settings.....	29
5.5	Configuring the Redundancy Settings.....	33
5.6	Set objects which should be synchronized.....	37
5.7	Write the redundancy settings to the controllers	38
5.8	Downloading the Application.....	40
6	Setting Up Modbus TCP Communication in CODESYS.....	41
6.1	Adding the Ethernet Adapter.....	41
6.2	Adding the Modbus TCP Client.....	44
6.3	Adding the Modbus TCP Server	46
6.4	Configuring the Modbus TCP Couplers in CODESYS.....	48
6.5	Downloading the Application.....	50
7	Set up a minimal application	51
7.1	Add the Modbus TCP Redundancy Manager.....	55
7.2	Downloading the Application.....	57
8	Validation.....	58
8.1	Verify the Initial System State	58
8.2	Verify Controller Switchover.....	61
8.3	Verify the Redundant Modbus TCP Communication	61

1 Warning and Disclaimer

Warning

Controls may fail in unsafe operating conditions, causing uncontrolled operation of the controlled devices. Such hazardous events can result in death and / or serious injury and / or property damage. Therefore, there must be safety equipment provided / electrical safety design or other redundant safety features that are independent from the automation system.

Disclaimer

This Application Note / Quick Start Guide / Example Program does not relieve you of the obligation to handle it safely during use, installation, operation and maintenance. Each user is responsible for the correct operation of his control system. By using this Application Note / Quick Start Guide / Example Program prepared by Weidmüller, you accept that Weidmüller cannot be held liable for any damage to property and / or personal injury that may occur because of the use. The names *CODESYS Development System* is a registered trademark of CODESYS.

Note

The given descriptions and examples do not represent any customer-specific solutions, they are simply intended to help for typical tasks. The user is responsible for the proper operation of the described products. Application notes / Quick Start Guides / Example Programs are not binding and do not claim to be complete in terms of configuration as well as any contingencies. By using this Application Note / Quick Start Guide / Example Program, you acknowledge that we cannot be held liable for any damages beyond the described liability regime. We reserve the right to make changes to this application note / quick start guide / example at any time without notice. In case of discrepancies between the proposals Application Notes / Quick Start Guides / Program Examples and other Weidmüller publications, like manuals, such contents have always more priority to the examples. We assume no liability for the information contained in this document. Our liability, for whatever legal reason, for damages caused using the examples, instructions, programs, project planning and performance data, etc. described in this Application Note / Quick Start Guide / Example is excluded.

Security notes

In order to protect equipment, systems, machines and networks against cyber threats, it is necessary to implement (and maintain) a complete state-of-the-art industrial security concept. The customer is responsible for preventing unauthorized access to his equipment, systems, machines and networks. Systems, machines and components should only be connected to the corporate network or the Internet if necessary and appropriate safeguards (such as firewalls and network segmentation) have been taken.

2 Introduction

This application note describes the individual steps required to set up a redundant controller system using two UC20-M4000 controllers communicating with two UR20 Modbus TCP fieldbus couplers.

The setup uses the CODESYS Redundancy functionality to provide controller redundancy and managed Ethernet switches configured with Rapid Spanning Tree Protocol (RSTP) to provide network redundancy.

The document covers the hardware setup, controller configuration, network configuration, and validation procedures required to establish and verify redundant communication between the controllers and the distributed Modbus TCP I/O devices.



Note:

The setup, configuration steps and test results described in this application note have been validated exclusively using the hardware and software components listed in the Hardware Reference and Software Reference sections of this document.

The use of different hardware, firmware versions, software versions, network infrastructure components, or additional third-party devices may influence the behavior of the system and could lead to results that differ from those described herein. Therefore, any deviations from the documented setup must be evaluated and validated individually.

3 System Architecture

3.1 System Overview

The system presented in this application note demonstrates a redundant controller setup using two UC20 controllers and distributed Modbus TCP I/O. The objective is to maintain application execution and communication to the fieldbus level in defined failure scenarios.

The controller redundancy is provided by the CODESYS Redundancy Configuration functionality. In normal operation, one controller operates as an active PLC, while the second controller operates as the standby PLC. Both controllers execute the same application and are synchronized via a dedicated redundancy link. If the active controller becomes unavailable, the standby controller can take over application execution at runtime.

The distributed I/O is connected via two UR20-FBC-MOD-TCP-V2 Modbus TCP fieldbus couplers. For Modbus TCP communication in a redundant controller setup, the CODESYS Modbus TCP Client redundancy mechanism is used. This allows the Modbus TCP communication to be managed according to the current redundancy state of the controllers.

The fieldbus network is built as a redundant Ethernet topology using two managed Ethernet switches. Rapid Spanning Tree Protocol (RSTP) is used to prevent network loops during normal operation and to provide an alternative communication path in case of a cable interruption.

The overall setup combines controller redundancy and network redundancy. Controller redundancy ensures that the application can continue running on the second controller if the active controller fails. Network redundancy ensures that communication to the Modbus TCP fieldbus couplers can continue if one Ethernet connection in the fieldbus network is interrupted.

prioritized for network communication. If higher network or fieldbus traffic is expected, it is therefore recommended to use these interfaces for communication-intensive connections, such as the Modbus TCP fieldbus network.

Symbol	Description
	CODESYS Redundancy Link – Synchronization channel between the primary and secondary UC20 controller. Used for redundancy management and data synchronization.
	Controller Ethernet Connection – Ethernet connection between the UC20 controller and the local managed switch.
	Modbus TCP Network (RSTP Ring) – Redundant Ethernet network used for communication between the controllers and the UR20 Modbus TCP fieldbus couplers. Rapid Spanning Tree Protocol (RSTP) provides automatic path redundancy in case of a network failure.

3.3 CODESYS Redundancy Principle (System Redundancy)

The controller redundancy described in this application note is based on the CODESYS Redundancy Configuration. Two UC20 controllers (of the same type) execute the same application, while only one controller operates as the **active PLC**. The second controller remains in **standby mode**, continuously synchronizing its application data with the active controller and monitoring its operational state.

The official documentation of the “CODESYS Redundancy Configuration” can be found under the following link:

[CODESYS Redundancy Configuration](#)

The synchronization is performed via a dedicated IP-based redundancy link. CODESYS uses both **UDP** and **TCP** communication for the redundancy mechanism. UDP is used for controller status monitoring, while TCP is used for additional redundancy services such as registered area synchronization, visualization redundancy and acyclic data exchange.

Understanding this principle is important because the Modbus TCP communication described in this application note is directly linked to the redundancy state of the controllers. While CODESYS manages the controller switchover, the RSTP-based Ethernet network ensures that the communication path to the distributed Modbus TCP fieldbus couplers remains available in the event of a network failure. Together, these mechanisms provide a high available control system.

3.4 Media Redundancy

In addition to controller redundancy, the communication network must also tolerate failures. If the active controller remains operational but loses the communication path to the distributed I/O, the application can no longer exchange process data with the Modbus TCP fieldbus couplers. Therefore, network redundancy is an essential part of a highly available control system.

In this application, media redundancy is implemented by connecting the managed Ethernet switches and the **UR20-FBC-MOD-TCP-V2 Modbus TCP fieldbus couplers** in a redundant Ethernet ring topology (see Chapter 3.2). Both the managed switches and the fieldbus couplers support the **Rapid Spanning Tree Protocol (RSTP)** and therefore actively participate in the redundant network.

Under normal operating conditions, RSTP blocks one redundant communication path, preventing Ethernet loops while maintaining a backup path. If a cable or communication path fails, the RSTP topology is recalculated and the previously blocked path is enabled. This restores the communication path to the affected fieldbus couplers without requiring manual intervention.

It is important to distinguish between **controller redundancy** and **media redundancy**. CODESYS Redundancy ensures that application execution can continue on the standby controller if the active controller fails. Media redundancy, on the other hand, ensures that the Ethernet communication path between the controllers, switches and Modbus TCP fieldbus couplers remains available after a network failure. Both mechanisms complement each other and are required to achieve a highly available automation system.

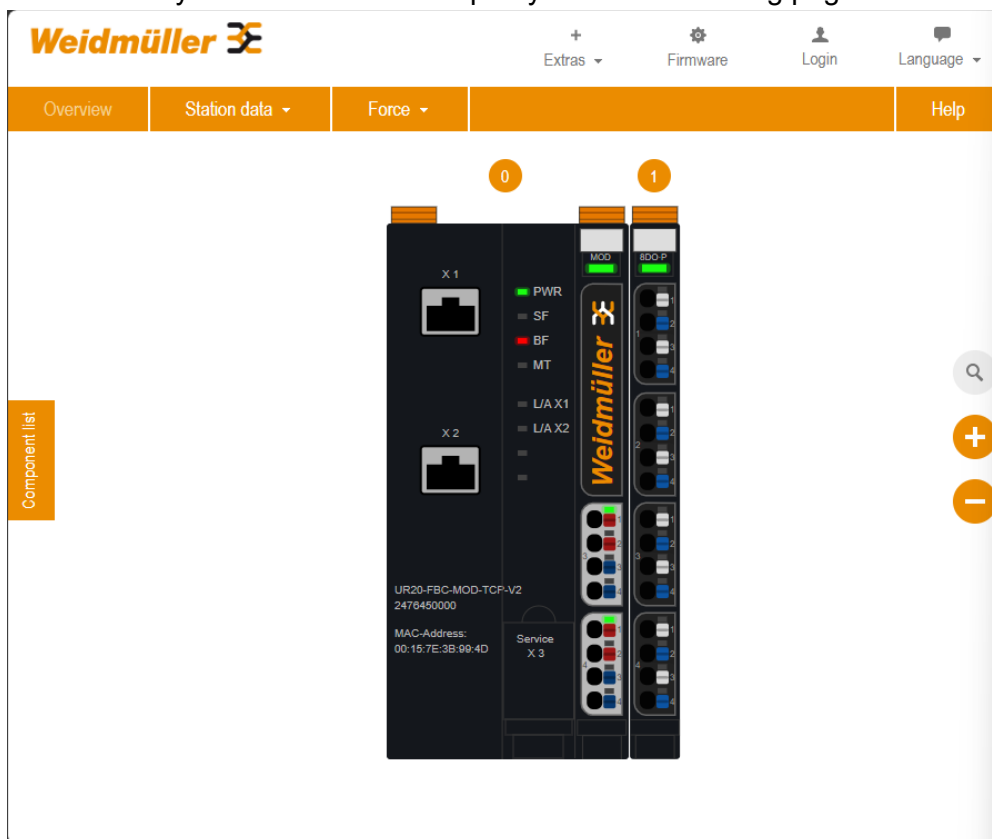


Note:

This application note focuses on the configuration of CODESYS controller redundancy. The configuration of the RSTP network, including the managed Ethernet switches and the RSTP configuration of the UR20-FBC-MOD-TCP-V2 fieldbus couplers, is described in a separate application note AN0145 – Setting Up Media Redundancy using RSTP. The network shown in this document is therefore assumed to be correctly configured and fully operational before commissioning the controller redundancy. Nevertheless, an understanding of the underlying media redundancy concept is important, as the overall availability of the system depends on the interaction between controller redundancy and network redundancy.

4 Set up the fieldbus coupler UR20-FBC-MOD-TCP-V2

1. To connect to the fieldbus coupler connect it via the USB-Service interface to the development computer.
2. Open a web browser and type in the address 192.168.1.102
3. If successfully connected to the coupler you will see starting page

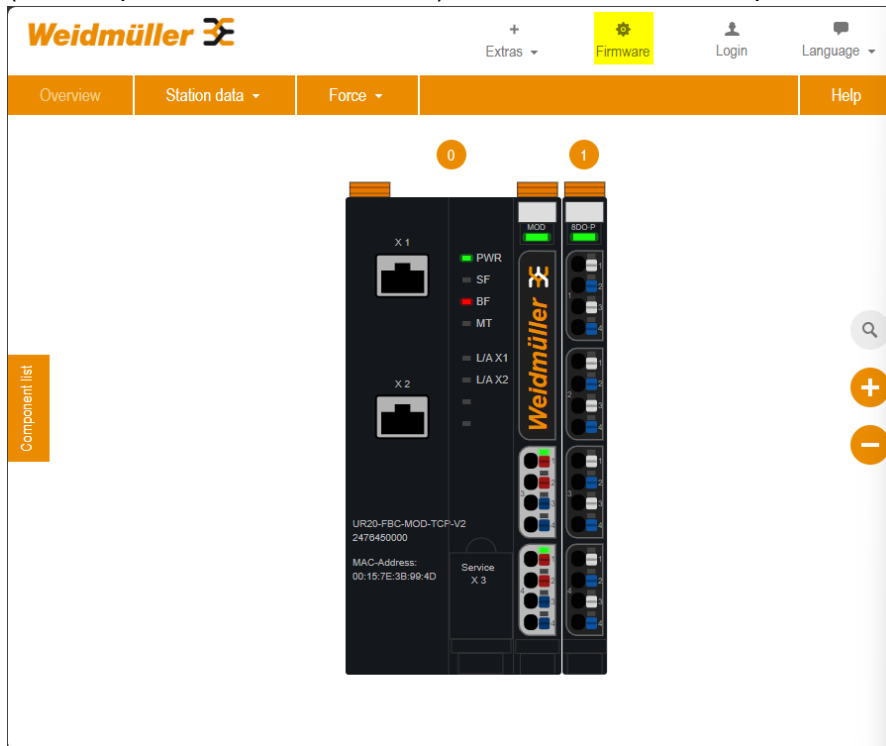


Note:

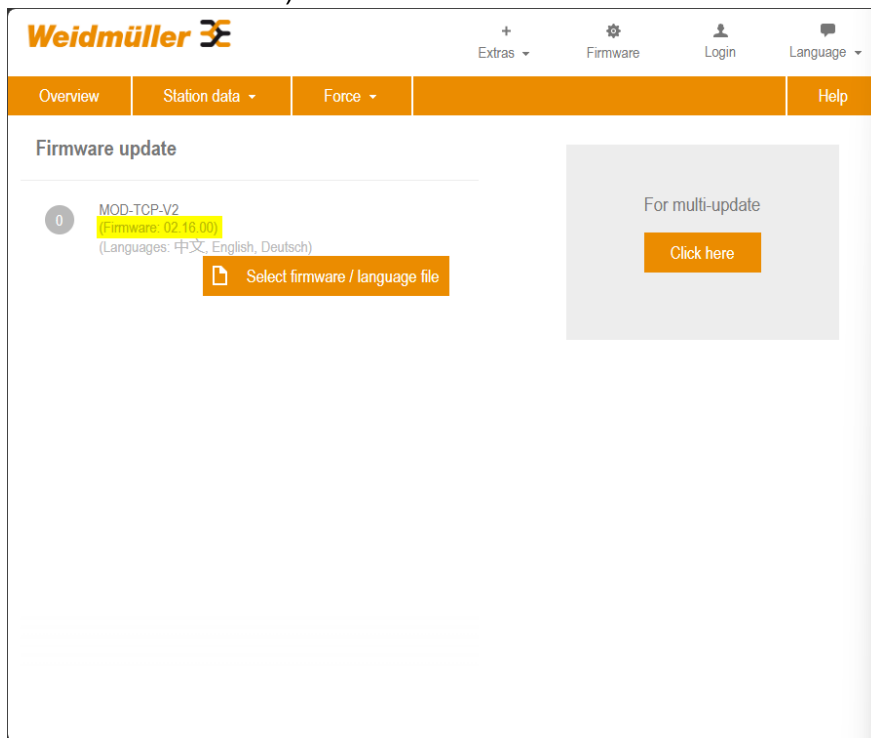
If any IO-cards were added they appear automatically in the web-Ui. If not check if the backplane bus is powered correctly (both power supply path) and make sure card has a tight connection with the fieldbus coupler.

Setting Up a Redundant Controller Architecture with UC20 Controllers and Modbus TCP I/O

4. First it must be checked if the current firmware fits the one required by the application note (see chapter software references). Click on Firmware to open the overview.



5. Check if the firmware version suits the one required by this application note (see chapter software references).



- If the firmware version deviates from the one required by this application note go to the Weidmueller product catalog and download the corresponding version.

Link: [UR20-FBC-MOD-TCP-V2 | Feldbuskoppler](#)

The screenshot shows the 'Software Support' page for the UR20-FBC-MOD-TCP-V2 fieldbus coupler. On the left, there are navigation links: 'Produkt Downloads' (Anwenderdokumentation, Zertifikat, Konformitätsdokument, Kataloge), 'Engineering Support' (Engineering-Daten, WSCAD Universe, Eplan Data Portal, CADENAS eCATALOG), 'Software Support' (Software), and 'Support Center'. The main content area is titled 'Software Support' and lists several software items:

- WI-Manager, DTM-Library for online installation** (ZIP, Größe 232 MB)
- USB driver** (ZIP, Driver, Größe 7 KB)
- Archiv Firmware files Modbus V2 coupler** (ZIP, Firmware, Größe 34 MB)
- Release-Notes - Firmware and Supported modules list** (ZIP, Release Note, Größe 244 KB)
- Current Firmware file Modbus V2 coupler** (BSC, Firmware, Größe 3 MB)

- Install the firmware by clicking on “Select firmware / language file” and select the corresponding firmware file you want to install.

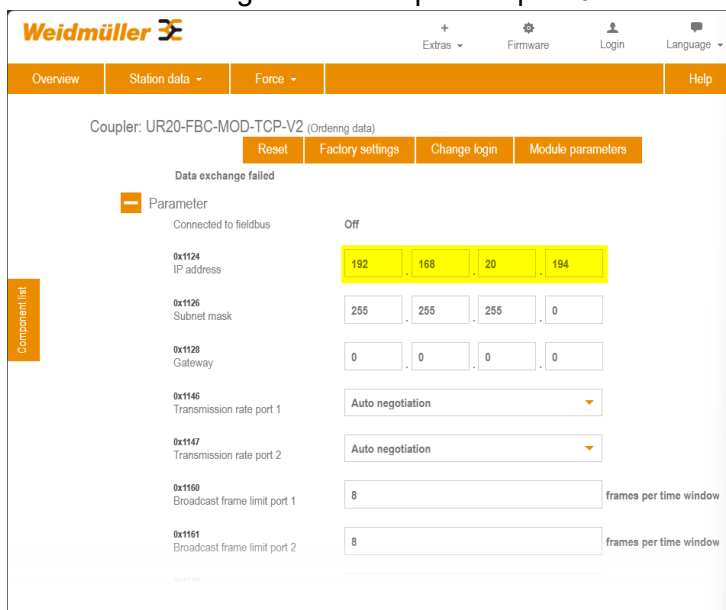
The screenshot shows the Weidmüller Web Interface (WMI) with the 'Firmwareupdate' section active. The top navigation bar includes 'Overview', 'Station data', 'Force', and 'Help'. The 'Firmwareupdate' section displays a list of available firmware files for the MOD-TCP-V2 (Firmware: 02.16.00, Languages: 中文, English, Deutsch). A yellow box highlights the 'Select firmware / language file' button. To the right, there is a 'For multi-update' section with a 'Click here' button.

Setting Up a Redundant Controller Architecture with UC20 Controllers and Modbus TCP I/O

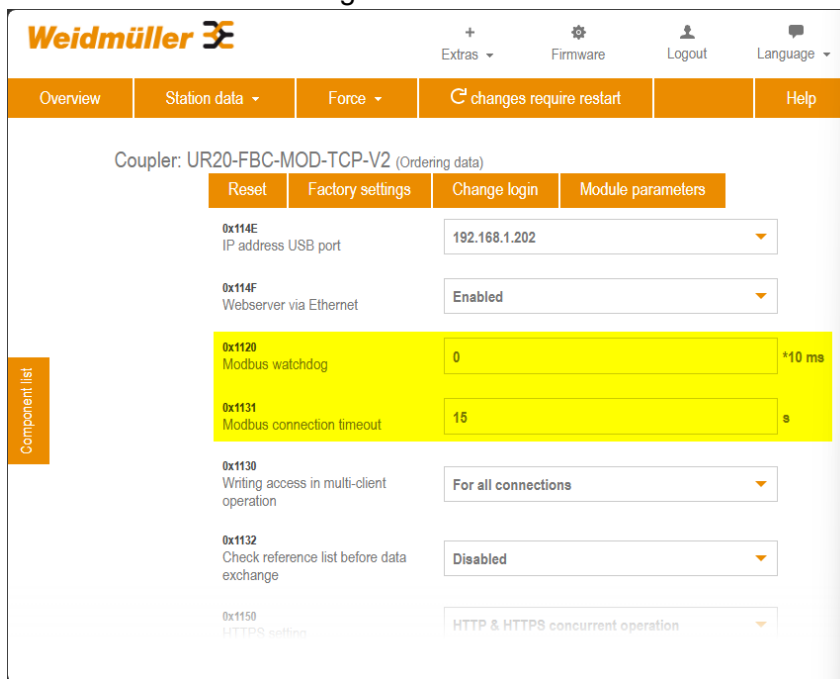
8. Start the installation and wait until the fieldbus coupler restarted.
9. Start parameterizing the coupler by clicking on the station head indicated by the number 0.



10. Open the parameter list and start with entering the IP address. Make sure the configuration matches the configuration set up in chapter 6.4.



11. Set the “Modbus watchdog” to 0 and the “Modbus connection timeout” to 15.



The screenshot shows the Weidmüller web interface for configuring a UR20-FBC-MOD-TCP-V2 coupler. The interface includes a top navigation bar with 'Overview', 'Station data', 'Force', 'changes require restart', and 'Help'. A sidebar on the left contains a 'Component list'. The main content area shows the coupler's configuration options:

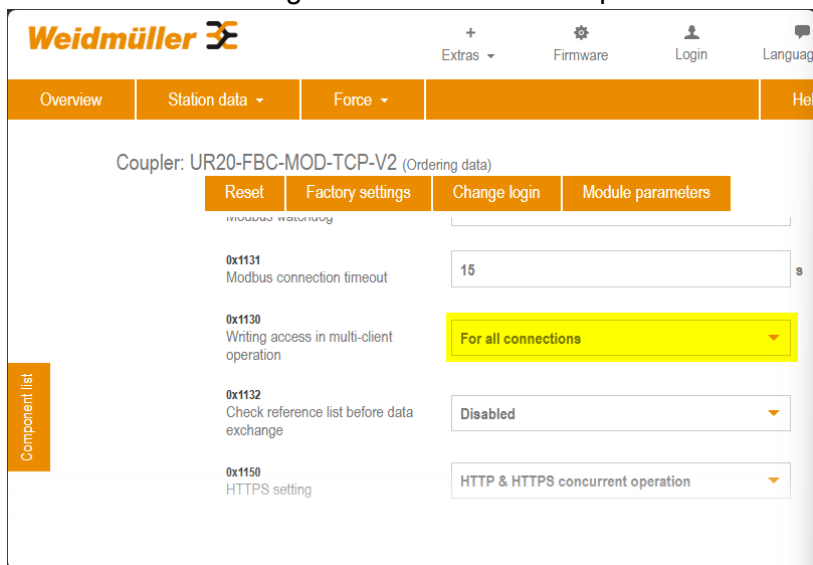
- Reset**, **Factory settings**, **Change login**, **Module parameters** (Buttons)
- 0x114E** IP address USB port: 192.168.1.202
- 0x114F** Webserver via Ethernet: Enabled
- 0x1120** Modbus watchdog: 0 (highlighted in yellow)
- 0x1131** Modbus connection timeout: 15 (highlighted in yellow)
- 0x1130** Writing access in multi-client operation: For all connections
- 0x1132** Check reference list before data exchange: Disabled
- 0x1150** HTTPS setting: HTTP & HTTPS concurrent operation



Note:

If the watchdog is < 0 it requires the application connecting to the fieldbus coupler to reset the watchdog periodically within the configured time frame otherwise the watch will trigger an error and the coupler must be reset. For this use case it is recommended to not enable the watchdog at all.

12. Make sure the “Writing access in multi-client operation” is set to “For all connections”.



The screenshot shows the Weidmüller web interface for configuring a UR20-FBC-MOD-TCP-V2 coupler. The interface includes a top navigation bar with 'Overview', 'Station data', 'Force', and 'Help'. A sidebar on the left contains a 'Component list'. The main content area shows the coupler's configuration options:

- Reset**, **Factory settings**, **Change login**, **Module parameters** (Buttons)
- 0x1131** Modbus connection timeout: 15
- 0x1130** Writing access in multi-client operation: For all connections (highlighted in yellow)
- 0x1132** Check reference list before data exchange: Disabled
- 0x1150** HTTPS setting: HTTP & HTTPS concurrent operation

13. Apply changes by clicking on “Apply changes”.

Weidmüller

Overview Station data Force changes require restart Help

Coupler: UR20-FBC-MOD-TCP-V2 (Ordering data)

Reset Factory settings Change login Module parameters

Diagnoses
Data exchange failed

Parameter
Connected to fieldbus Off

0x1124
IP address 192 . 168 . 20 . 194

0x1126
Subnet mask 255 . 255 . 255 . 0

0x1128
Gateway 0 . 0 . 0 . 0

Apply changes Restore

14. To write the changes to the memory of the coupler and make them stay indefinitely click on “Module Parameters”.

Component list

Coupler: UR20-FBC-MOD-TCP-V2 (Ordering data)

Reset Factory settings Change login **Module parameters**

0x1186 RSTP port 1 path cost	200000
0x1188 RSTP port 2 priority	128
0x1189 RSTP port 2 path cost	200000
0x1140 Modbus transport protocol	TCP
0x114E IP address USB port	192.168.1.202
0x114F Webserver via Ethernet	Enabled

15. Save the parameters.

Module parameters

The coupler can store the currently set module parameters, or set them back to default values.

No values are stored currently.

Restore default values restores the default values and deletes possibly stored values.
Save module parameters stores the currently set module parameters and overwrites possibly stored values.

Restore default values **Save module parameters** Cancel

16. By restarting the coupler, the parameter changes will be activated.

Reset	Factory settings	Change login	Module parameters
0x1186	RSTP port 1 path cost	200000	
0x1188	RSTP port 2 priority	128	
0x1189	RSTP port 2 path cost	200000	
0x114D	Modbus transport protocol	TCP	
0x114E	IP address USB port	192.168.1.202	
0x114F	Webserver via Ethernet	Enabled	

17. Repeat steps one to eighteen for the second fieldbus coupler. Make sure that when executing step 10. not setting the same IP address as for coupler 1.

Expl.:

Coupler 1: 192.168.20.194

Coupler 2: 192.168.20.196

5 Setting Up CODESYS Controller Redundancy

5.1 Preparing the UC20-M4000 Controllers

Before configuring the controller redundancy in CODESYS, both UC20-M4000 controllers must be prepared. To ensure a consistent software environment and avoid configuration conflicts from previous installations, it is recommended to start with a factory reset.

The preparation consists of the following steps:

1. **Perform a Factory Reset**

Reset both controllers to their factory default configuration.

1. Click on “u-OS Control Center”

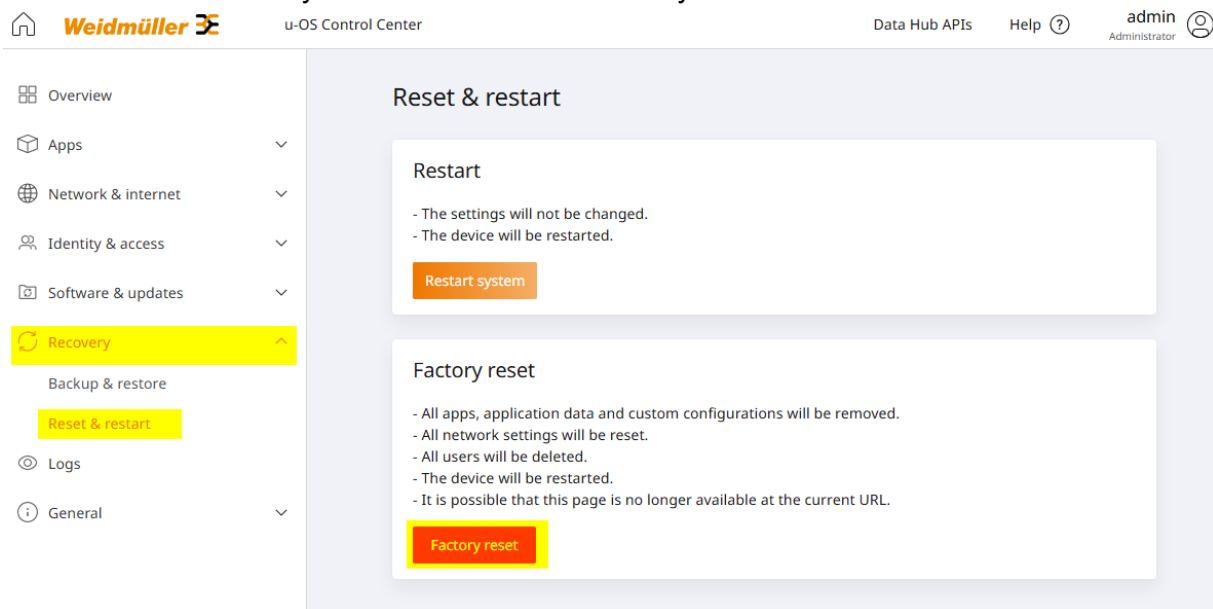
The screenshot displays the Weidmüller UC20-M4000 u-OS web interface. At the top, the header includes the Weidmüller logo, the model name 'UC20-M4000', and navigation links for 'Data Hub APIs', 'Help', and an 'admin' user profile. The main content area begins with a 'Welcome to u-OS' message, indicating the last update was 24 seconds ago with a 'Refresh' button. Below this is a 'Device information' section featuring a product image and a table of details:

Product name	Order number	Serial number	u-OS version
UC20-M4000	2839160000	AX9T14PC7500001	2.8.0

Additional information includes the host name 'wm-uc20-m4000-ax9t14pc7500001' and a list of network interfaces: eth-x4 (disabled), eth-x5 (disabled), eth-x6 (192.168.20.10), eth-x7 (192.168.0.10), and usb-x1 (disabled). The interface also presents four system management tiles: 'u-OS Control Center' (highlighted in yellow, for configuring the system), 'u-OS Data Hub APIs' (for API overview), 'u-OS Data Hub Browser' (for seamless data exchange), and 'u-OS IO Configurator' (for configuring remote devices).

Setting Up a Redundant Controller Architecture with UC20 Controllers and Modbus TCP I/O

2. From there click on Recovery → Reset & restart → Factory reset

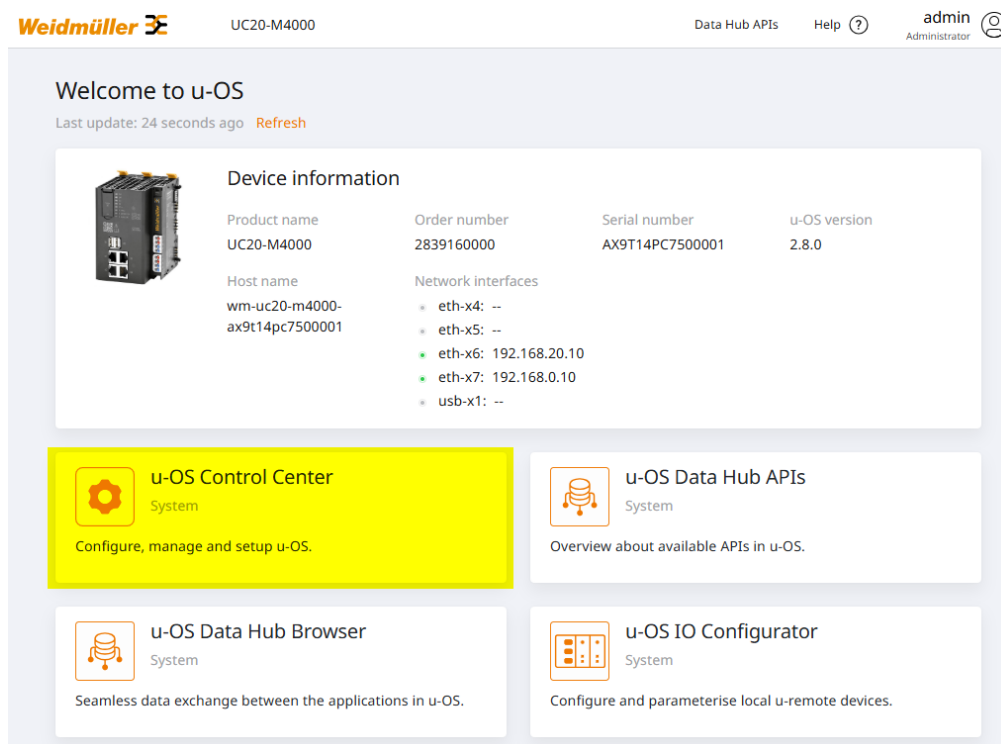


3. Wait until the factory reset is done and u-OS can be accessed again

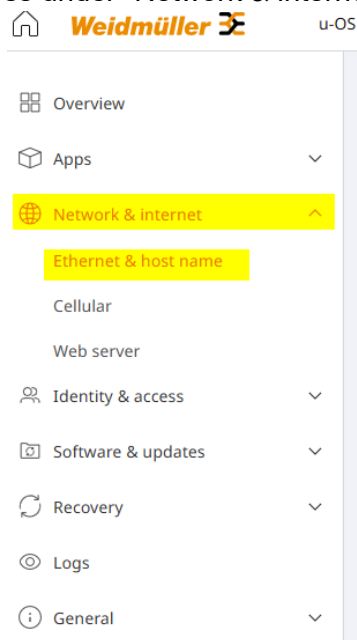
2. Configure the Network Settings

Configure the required IP addresses for the Ethernet interfaces in u-OS. These addresses are later used for the CODESYS engineering connection, the controller redundancy link and the Modbus TCP network.

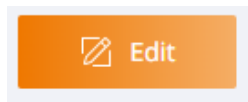
1. Click on “u-OS Control Center”



2. Access under “Network & internet” the “Ethernet & host name”



3. Click on edit and start editing the IP-Addresses



4. For this example, the IP-Addresses were chosen as follows:

1. Controller 1 (Default Active)

Network configuration		Network configuration	
eth-x4	eth-x5	eth-x4	eth-x5
Enabled	☒	Enabled	☒
Dynamic (DHCP)	☐	Dynamic (DHCP)	☐
IP address	192.168.20.10	IP address	192.168.0.10
Subnet mask	255.255.255.0	Subnet mask	255.255.255.0
Default gateway		Default gateway	
DNS servers		DNS servers	
Cancel Apply		Cancel Apply	

2. Controller 2 (Default Standby)

Network configuration

eth-x4 eth-x5 eth-x6 eth-x7 usb-x1

Enabled ☒

Dynamic (DHCP) ☐

IP address 192.168.20.11

Subnet mask 255.255.255.0

Default gateway

DNS servers

Cancel

Apply

Network configuration

eth-x4 eth-x5 eth-x6 eth-x7 usb-x1

Enabled ☒

Dynamic (DHCP) ☐

IP address 192.168.0.11

Subnet mask 255.255.255.0

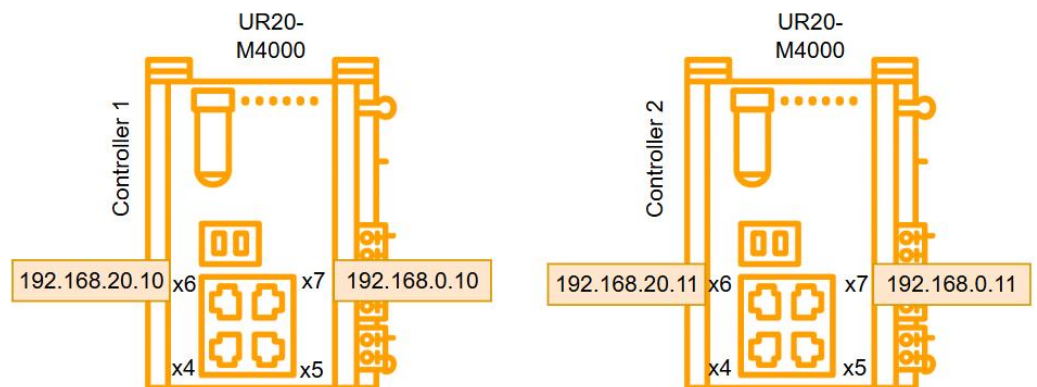
Default gateway

DNS servers

Cancel

Apply

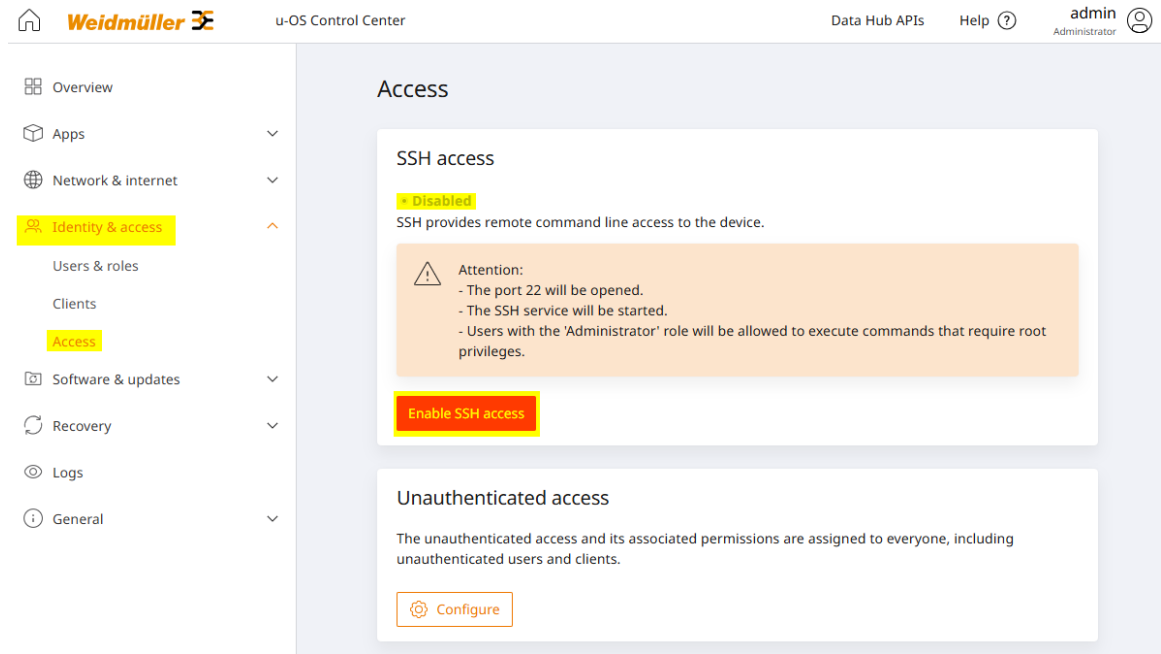
3. This results in the following IP-address assignment



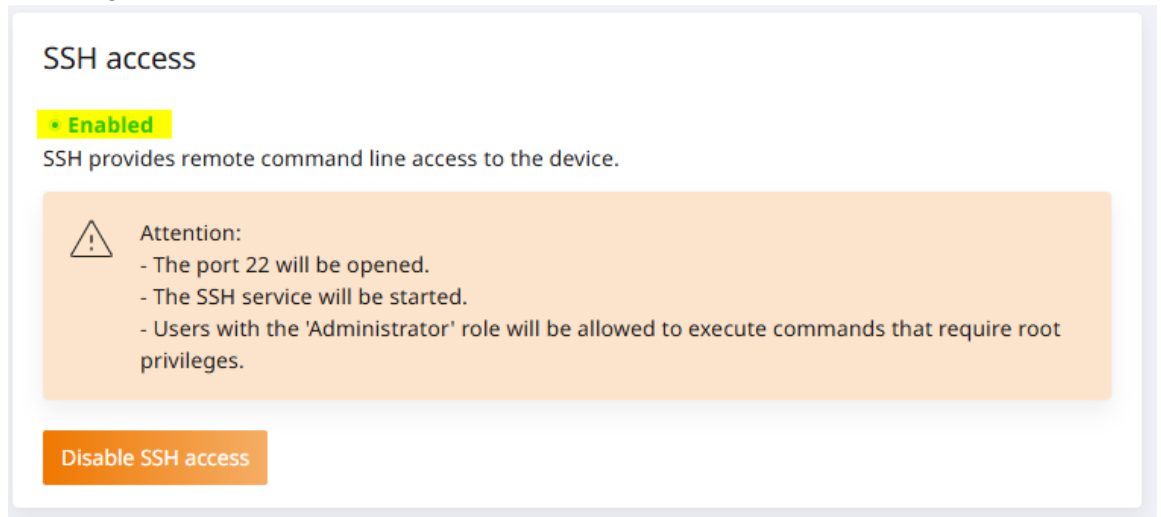
3. Enable SSH Access

Enable the SSH service in u-OS. SSH access is required for the initial installation of the CODESYS runtime from the CODESYS Development System.

1. After configuring the IP-Addresses you directly change to “Identity & access” and the click on “Access”



2. After enabling SSH access it is indicated as follows:



After completing these steps, both controllers are ready for the installation of the CODESYS runtime and the subsequent configuration of the controller redundancy.

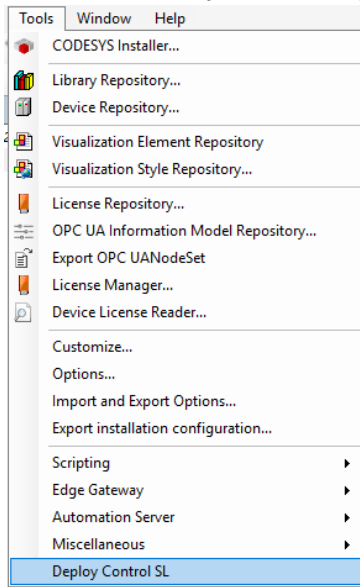
5.2 Installing the CODESYS runtime on u-OS



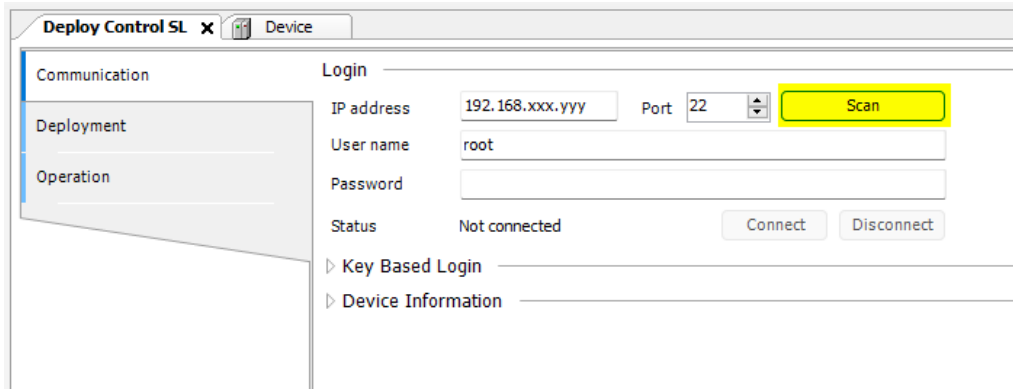
Note:

SSH access must be enabled on the device to successfully deploy the CODESYS runtime from the CODESYS development system.

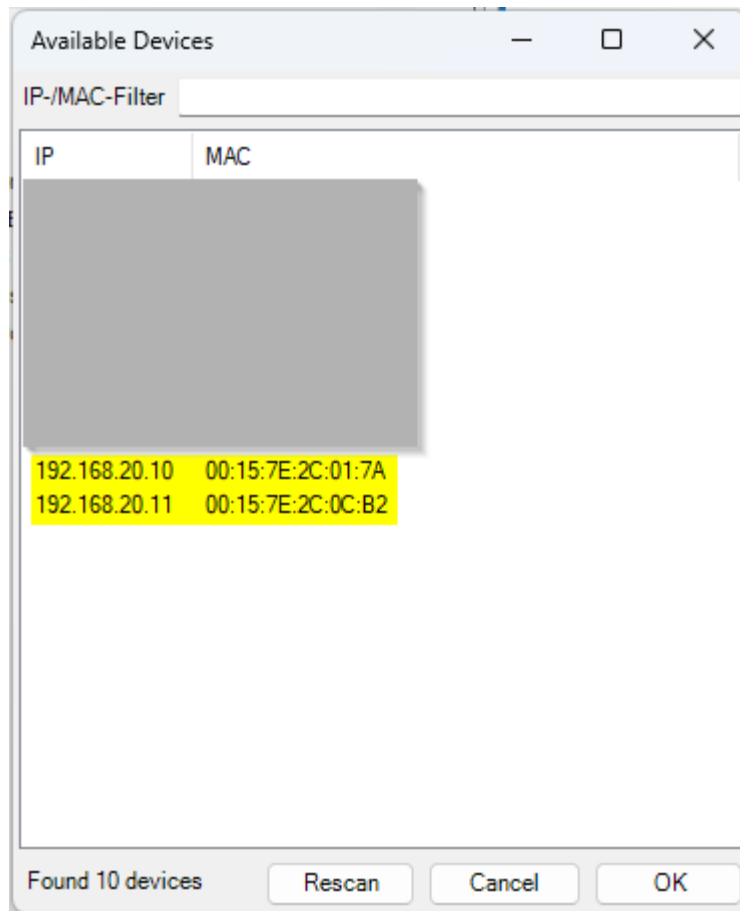
3. Open the deployment dialog under “Tools”→”Deploy Control SL”



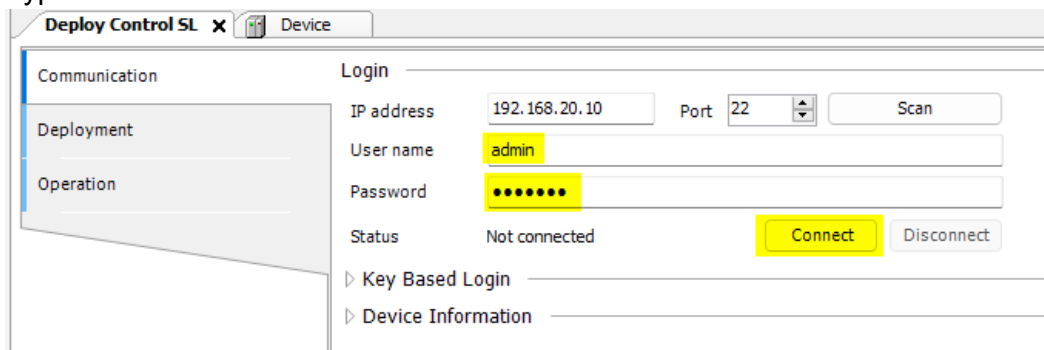
4. Click scan to scan all the devices within the network.



5. If successfully performed the devices are displayed in the browsing dialog.



4. Type in the SSH credentials and connect to the controller.

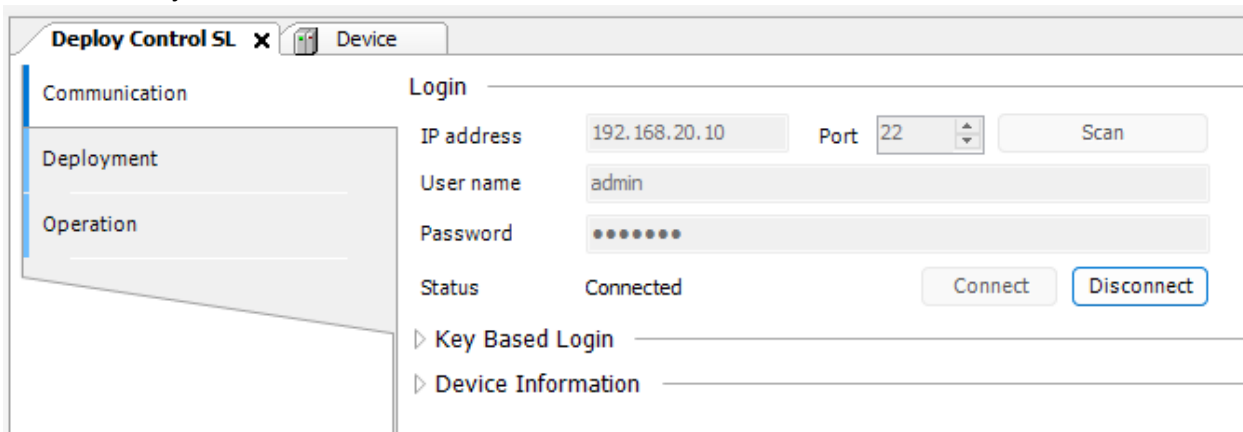


Note:

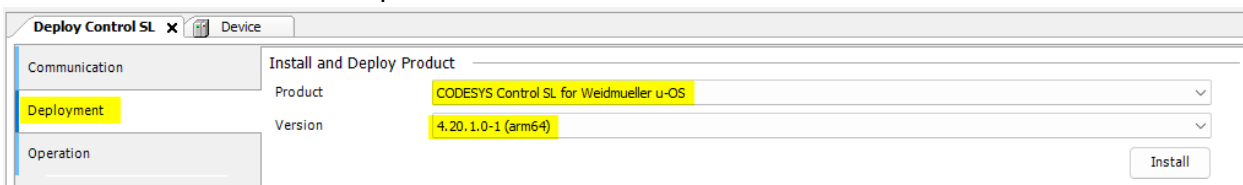
The default credentials chosen whenever a controller was set up within the Weidmueller organization are as follows:

Username: admin
Password: Detmold

5. If successfully connected the “Status” should indicate it.



6. Select the deployment tab and choose under “Product” CODESYS Control SL for Weidmueller u-OS and under “Version” the corresponding version you want to install. Click install to start the installation process.



Note:

The CODESYS Runtime is an Application which can be is part of the “CODESYS Control SL for Weidmueller u-OS Package Version 4.xx.x.x”. Both the Application and the Package can be found as separate files on the Weidmueller Support center.

Link: [Weidmüller Support Center](#)

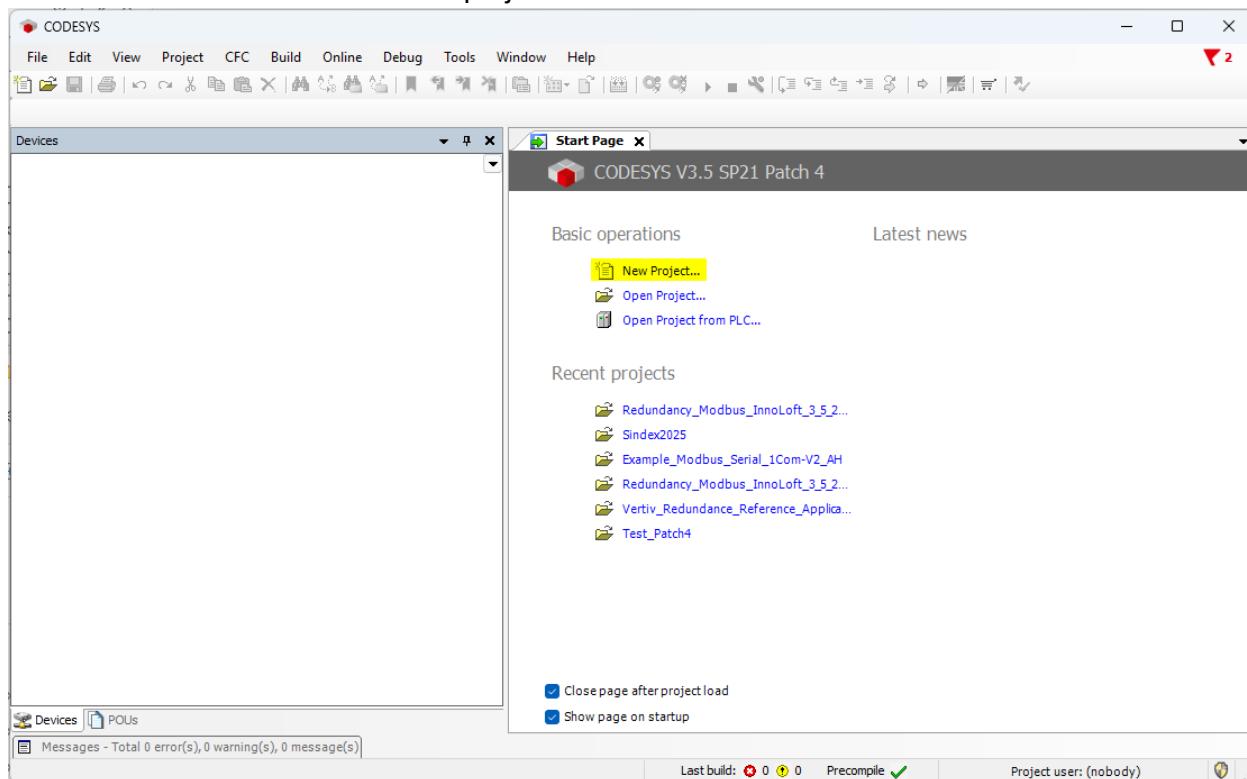
7. After the installation repeat step 2-6 to reconnect with the controller
8. Check in the Deployment tab if the package was installed successfully



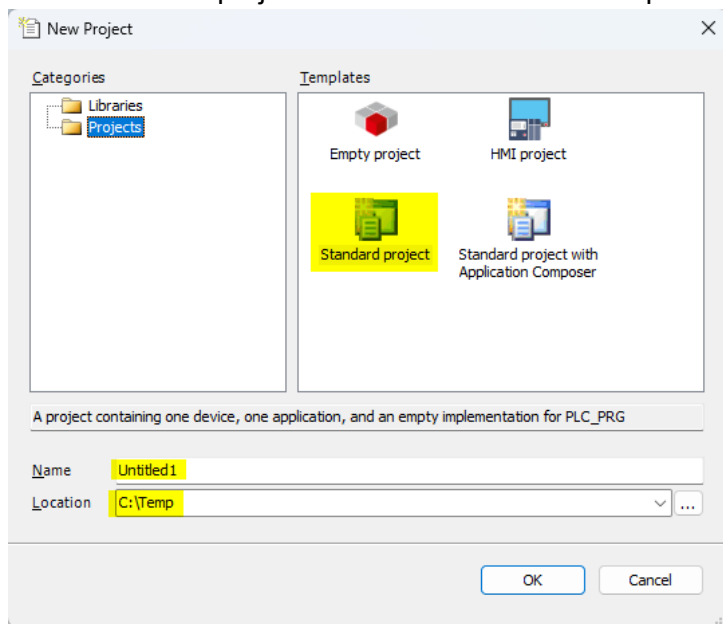
9. Repeat the same process for Controller 2.

5.3 Creating the CODESYS Project

1. Start CODESYS and create a new project.

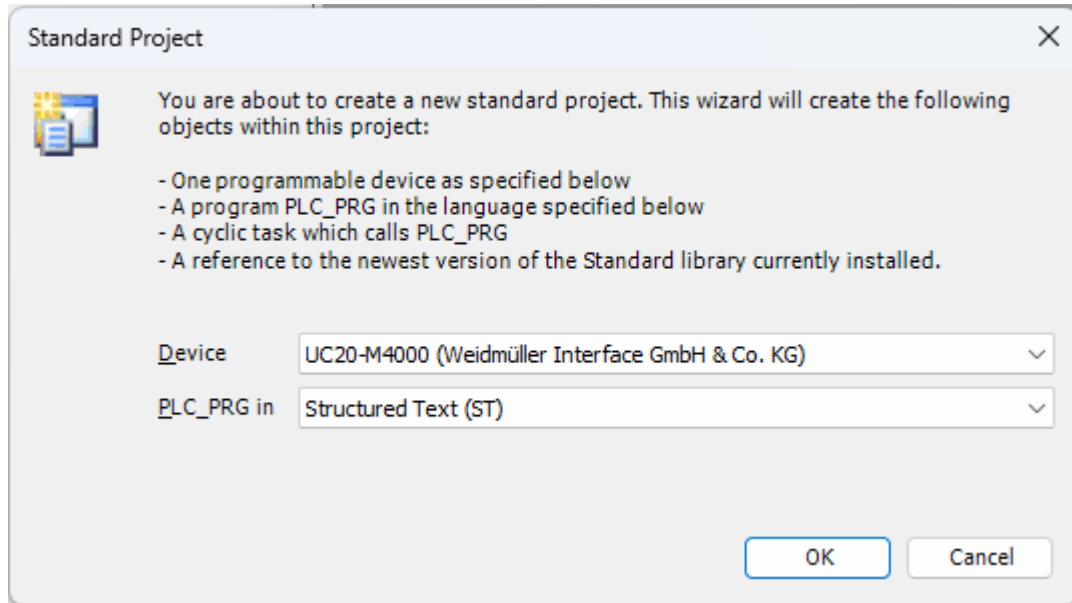


2. Click “Standard project” and choose a name and path where the project is stored:

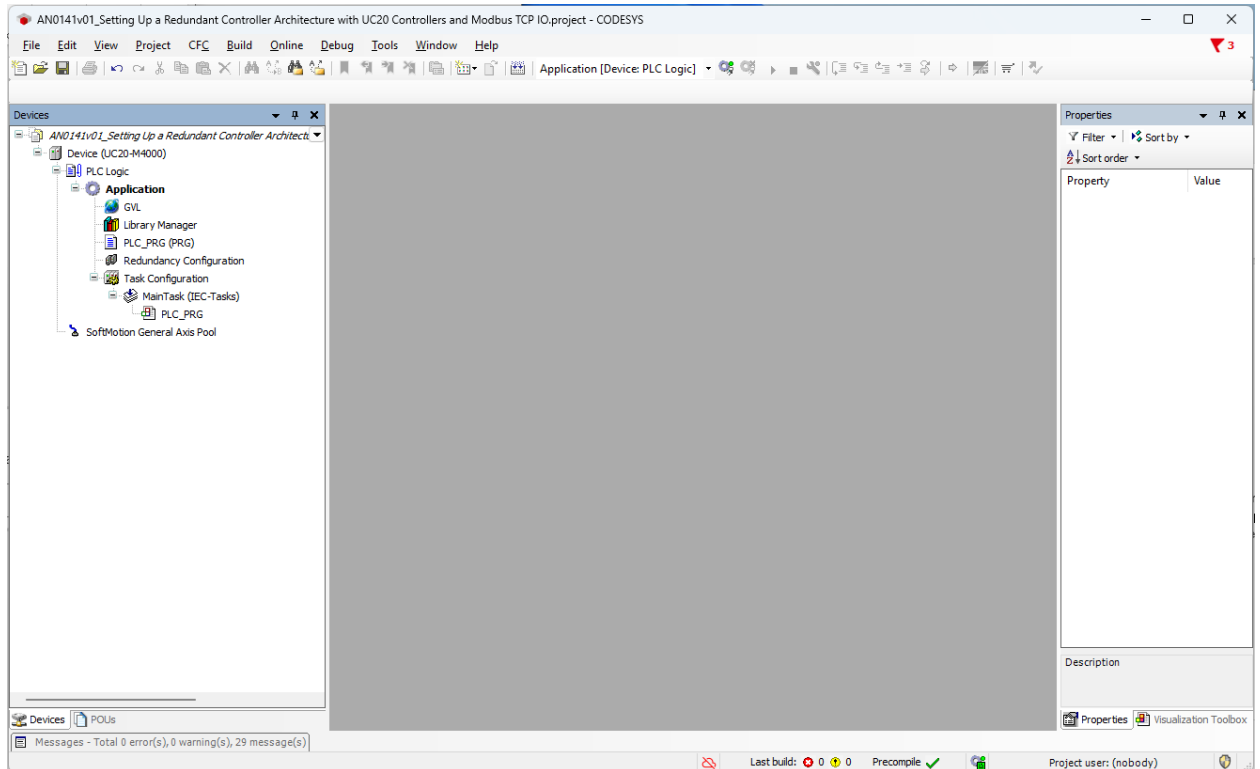


Setting Up a Redundant Controller Architecture with UC20 Controllers and Modbus TCP I/O

3. Choose the UC20-M4000 Controller and set the default programming language to “Structured Text”.



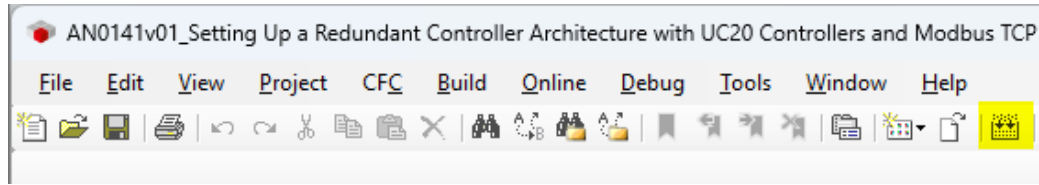
10. A basic project structure should appear if the project is created correctly.



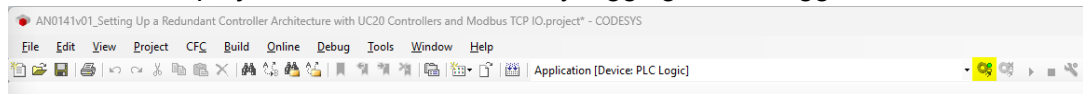
Setting Up a Redundant Controller Architecture with UC20 Controllers and Modbus TCP I/O

11. Download the project to each controller to make sure that an application is running when later configuring and writing the redundancy configuration to the controllers.

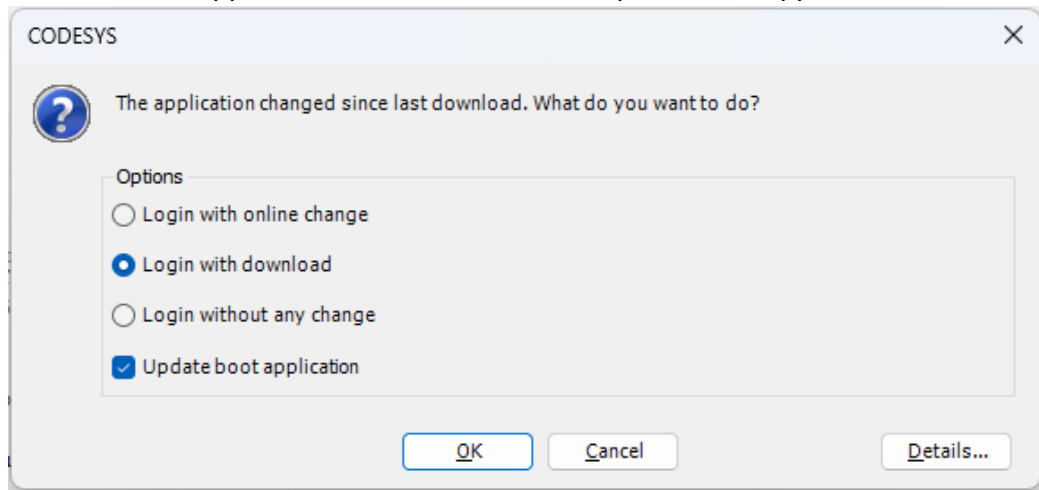
1. Compile the project.



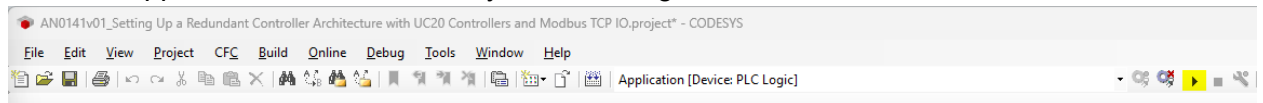
2. Download the project on to the controller by logging in and trigger the download.



3. Download the application and make sure the Update boot application field is enabled.



4. Start the application after successfully downloading.



5. Repeat this process for the second controller

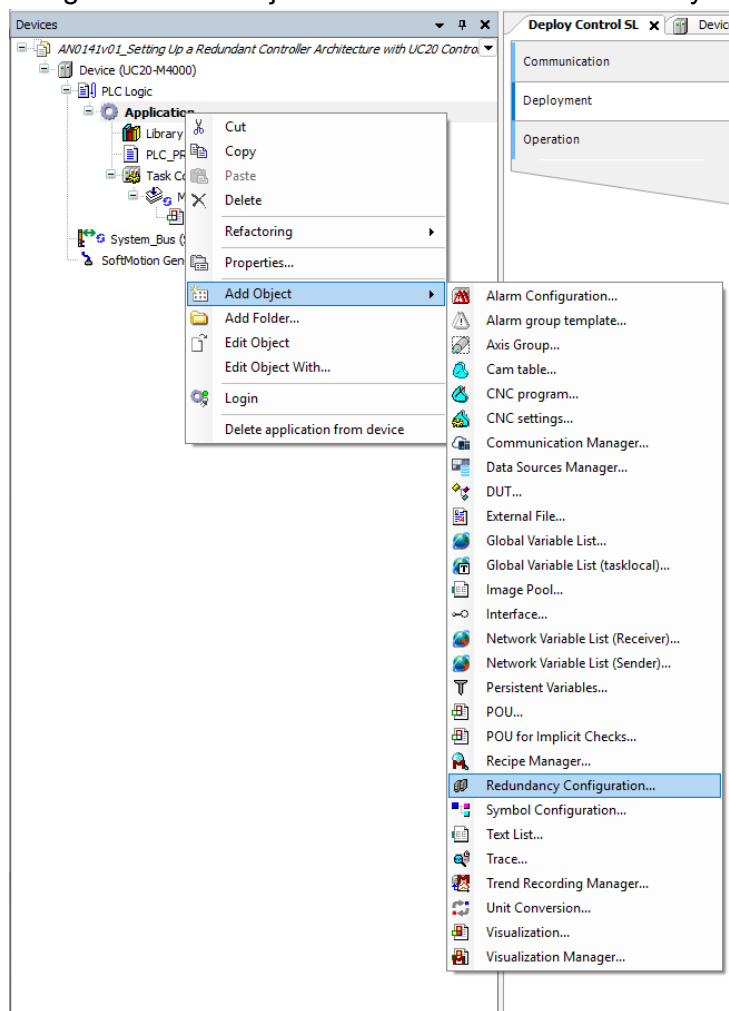
5.4 Configuring Redundancy Settings



Note:

The redundancy package must be installed to successfully take the following steps.

1. Add the redundancy configuration by right clicking on “Application” in the device tree, navigate to “Add Object” and search for “Redundancy Configuration”.

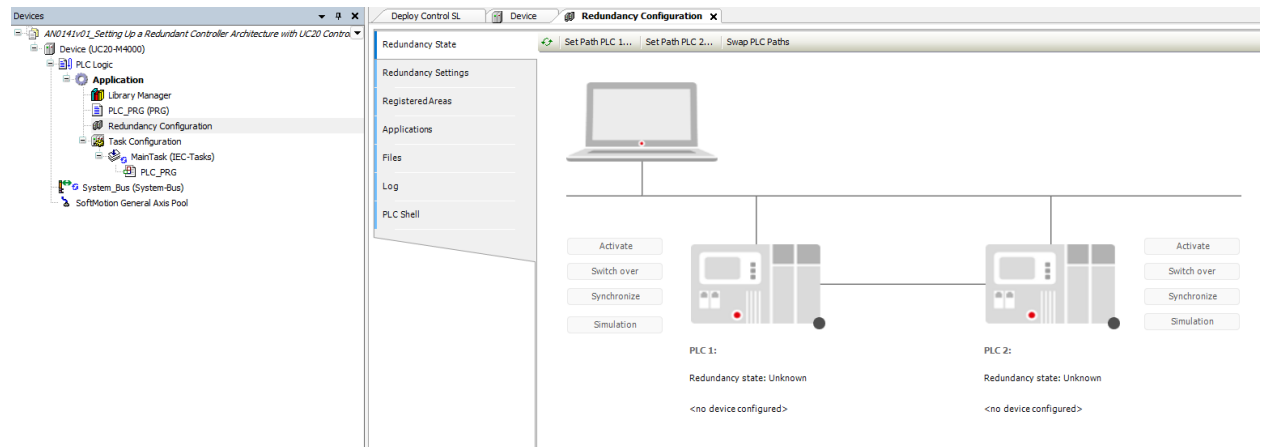


Note:

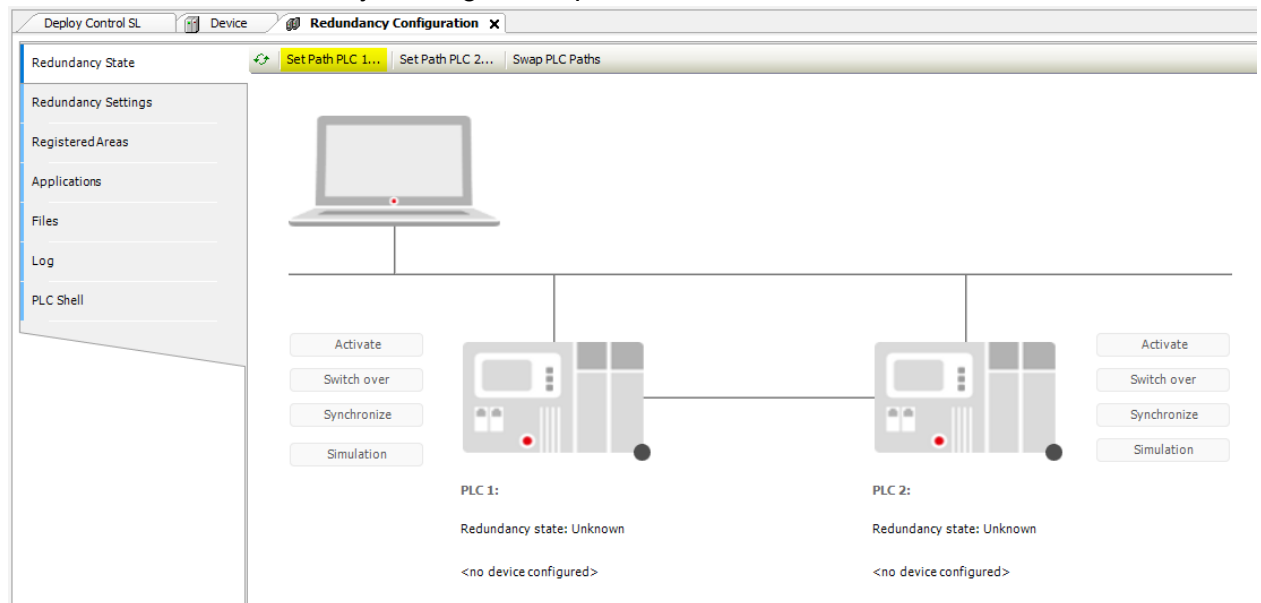
The Redundancy Configuration object is added only once to the engineering project. It represents the complete redundant controller pair and therefore contains the communication settings for both PLCs. During configuration, the engineering project is first connected to Controller 1 (PLC1) and afterwards to Controller 2 (PLC2) to assign both devices to the same redundancy configuration. Do not create a separate Redundancy Configuration object for the second controller.

Setting Up a Redundant Controller Architecture with UC20 Controllers and Modbus TCP I/O

6. After successfully adding the “Redundancy Configuration” it appears in the device tree and can be opened.

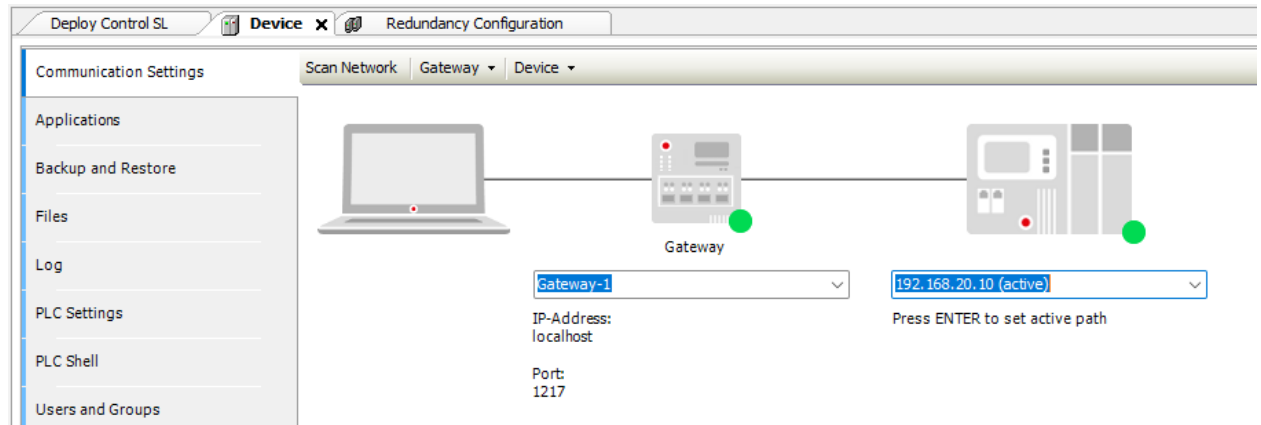


7. Connect to the Controller by clicking on set path to PLC1

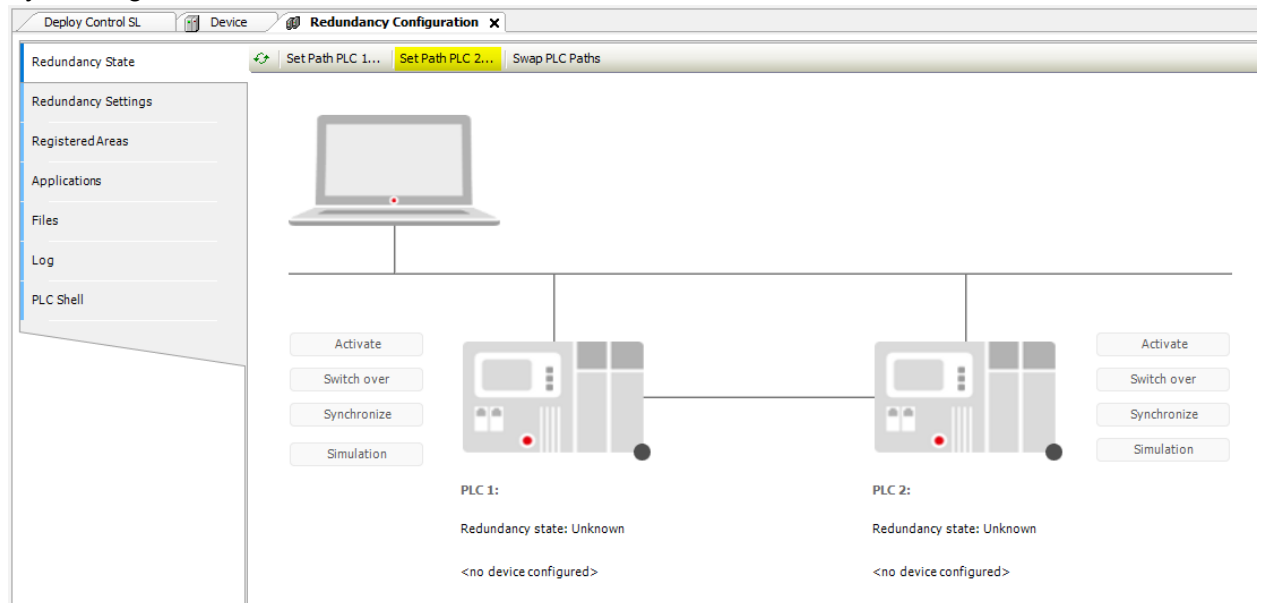


Setting Up a Redundant Controller Architecture with UC20 Controllers and Modbus TCP I/O

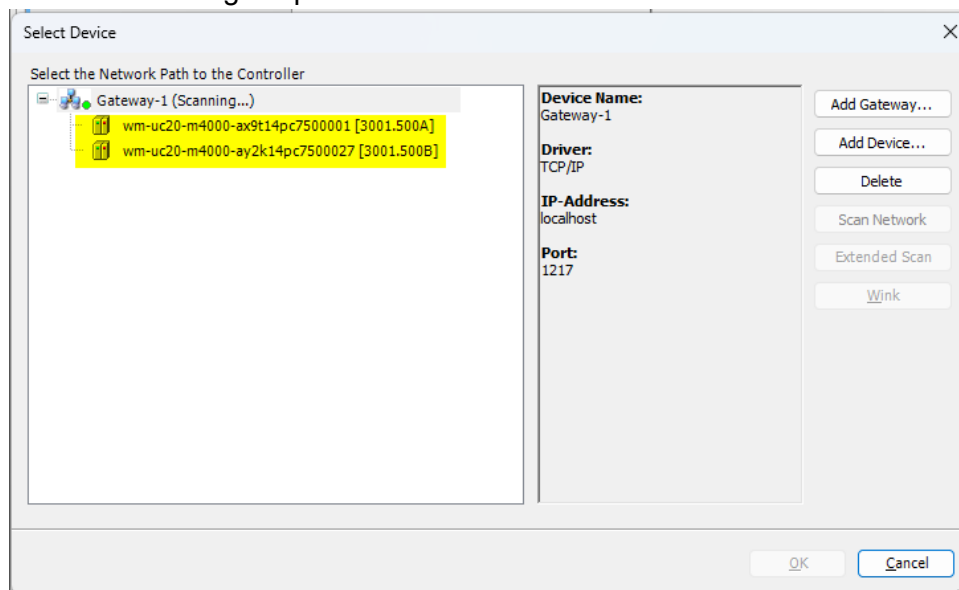
8. The “Device” dialog will be opened. Type the IP-Address of the controller 1 and press enter. If the connection is successful, the dot on the right corner will be green.



9. Return to the “Redundancy Configuration” and repeat the same process for controller 2 by clicking “Set Path to PLC2”.



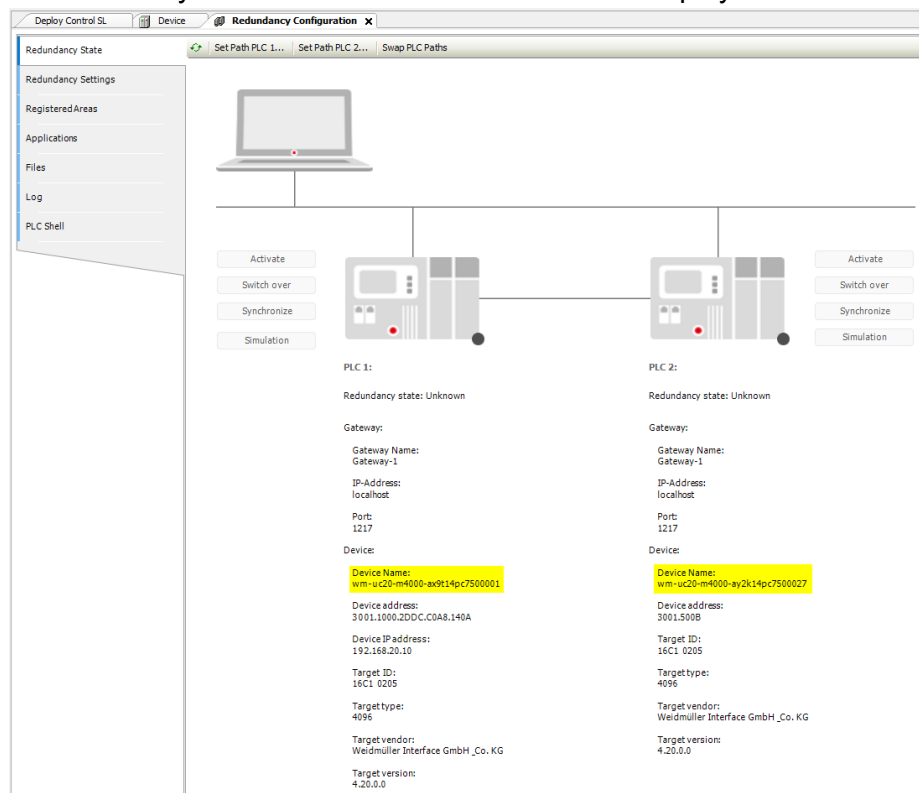
10. A selection dialog is opened. Select controller 2 and click ok.



Note:

Since only the host name is displayed in the selection dialog it can be hard to identify the device. If you don't know which device is controller 2 there is a wink function included. Select one device and click in "Wink". The "CTRL1" status LED will blink on the corresponding device.

11. If successfully conducted both controllers will be displayed in the redundancy state.



5.5 Configuring the Redundancy Settings

1. Open the Redundancy Settings tab.

The screenshot shows the 'Redundancy Configuration' window with the 'Redundancy Settings' tab selected. The left sidebar contains a tree view with 'Redundancy Settings' highlighted. The main area has three tabs: 'General', 'Fieldbus Settings', and 'Visualization'. The 'General' tab is active, showing the following settings:

- Application**
 - Redundancy task name: MainTask
 - Timeout (ms): 10
 - Synchronisation timeout (ms): 0
 - Bootup timeout (ms): 0
- Runtime Settings**
 - ☒ Auto sync
 - ☐ Data sync always
 - ☐ Debug messages
 - ☐ Record the needed sync time in system trace
- Redundancy Link**
 - IP address PLC 1: 0 . 0 . 0 . 0
 - IP address PLC 2: 0 . 0 . 0 . 0
 - Port: 1205
 - ☐ Use two redundant links

- Set the parameters “Timeout”, “Synchronization timeout” and “Bootup timeout” as followed.



Note:

The selected timeout values provide a good balance between fast fault detection and stable operation for the hardware and network topology described in this application note.

The Timeout parameter determines how quickly the standby controller detects the loss of communication with the active controller. Reducing this value decreases the switchover time but increases the risk of unnecessary transitions to standalone mode caused by temporary communication delays, task jitter or increased network latency. Increasing the value improves robustness but delays the detection of a controller failure.

The Synchronization timeout defines the maximum time allowed for transferring the synchronized application data between both controllers. Applications containing large global data areas, extensive visualization projects, retained variables or large online changes may require a higher synchronization timeout. If this value is too low, synchronization may be aborted before completion, preventing the standby controller from entering the synchronized state.

The Bootup timeout specifies how long the active controller waits for the standby controller to complete its startup and initial synchronization. Larger applications or systems requiring longer initialization times may require this value to be increased. Setting the value unnecessarily high only increases the startup time before a timeout is reported.

The timeout values presented in this application note were validated using the hardware and software versions listed in the *Hardware Reference* and *Software*

Reference sections. In particular, the selected Bootup timeout was determined empirically during system validation by identifying the minimum value that allowed reliable startup and synchronization of the redundant controller pair under all tested conditions.



Note:

The value configured for Redundancy task name must correspond to the name of the CODESYS Task Configuration object, not to the name of an individual cyclic task.

In the reference application, the Task Configuration object is named MainTask. The individual application tasks are configured and assigned within this Task Configuration. If a different name is used for the Task Configuration object, the Redundancy task name parameter must be adapted accordingly.

3. Set up the redundancy links

1. Click on the “Browse” button to assign an IP address for PLC 1 (controller 1).

Redundancy Link

IP address PLC 1: 0 . 0 . 0 . 0 [Browse...]

IP address PLC 2: 0 . 0 . 0 . 0 [Browse...]

Port: 1206

2. Select eth-x7 and click ok.

Network Adapters

Name	Description	IP address
lo		127.0.0.1
sit0		0.0.0.0
eth-x6		192.168.20.10
eth-x7		192.168.0.10
eth-x4		0.0.0.0
eth-x5		0.0.0.0

IP address: 172 . 17 . 0 . 1

[OK] [Cancel]

3. Make sure the Port number is set to the default 1205.



Note:

The default redundancy communication port is 1205. CODESYS uses this UDP port for the cyclic redundancy communication between the controllers. Additional TCP ports are automatically derived from this base port.

Link: [Tab: Redundancy Settings: General](#)

4. Repeat the process for PLC 2. If successful both IP addresses should be shown as follows.

Redundancy Link	
IP address PLC 1	192 . 168 . 0 . 10 Browse...
IP address PLC 2	192 . 168 . 0 . 11 Browse...
Port	1205



Note:

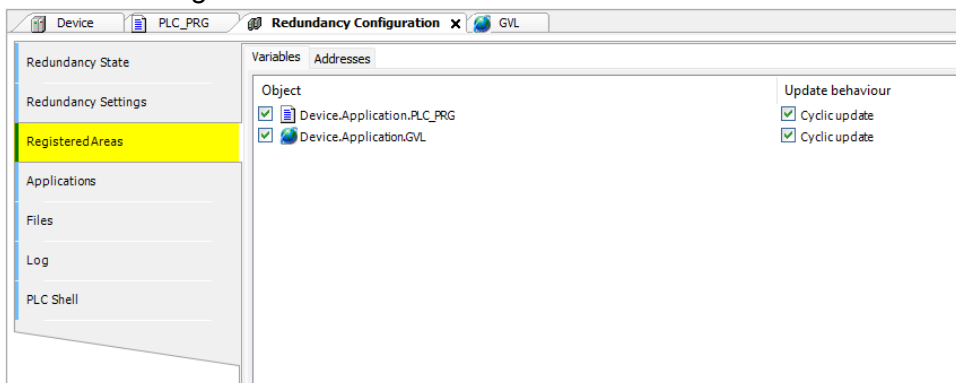
Do not enter the IP addresses manually. Always use the brows button otherwise this can lead to problems when writing the configuration to the controllers.

1. Enable Auto Sync

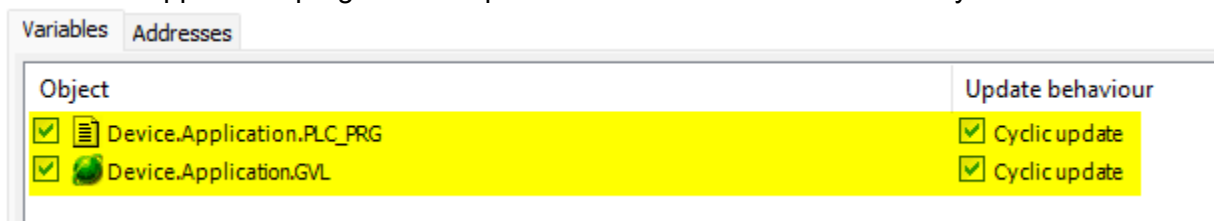
Runtime Settings		
☒ Auto sync	☐ Data sync always	☐ Debug messages
☐ Record the needed sync time in system trace		

5.6 Set objects which should be synchronized

1. Select the “Registered Areas” tab.



2. Select the Application programs and process variables which should be synchronized

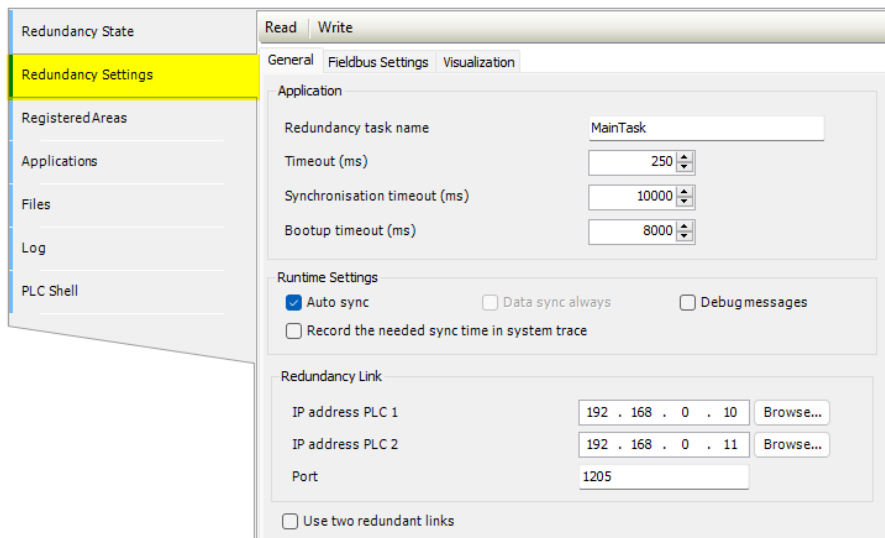


Note:

If “Cyclic update” is not enabled the Object is only initially synchronized.

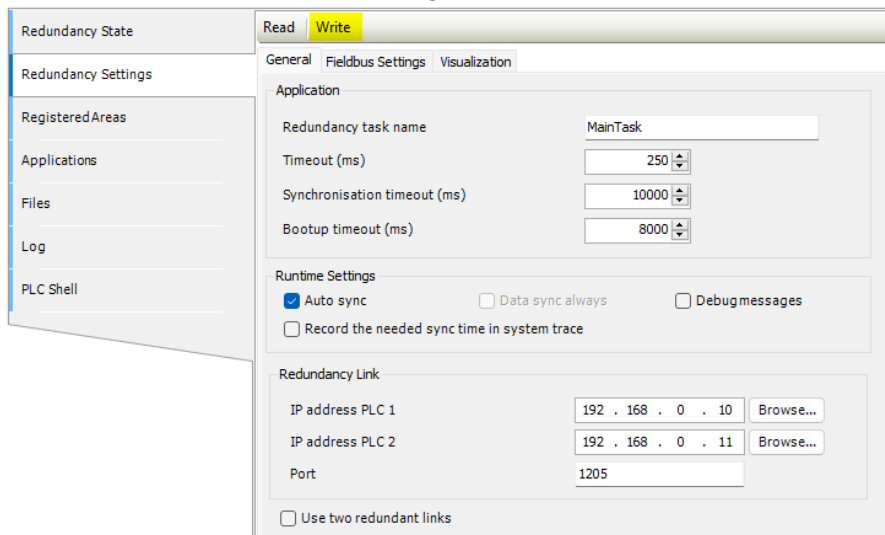
5.7 Write the redundancy settings to the controllers

1. Open the “Redundancy Settings” tab



The screenshot shows the 'Redundancy Settings' tab selected in the left sidebar. The main window has tabs for 'Read' and 'Write', with 'Read' currently active. The 'General' sub-tab is selected. The settings are organized into sections: 'Application' with fields for 'Redundancy task name' (MainTask), 'Timeout (ms)' (250), 'Synchronisation timeout (ms)' (10000), and 'Bootup timeout (ms)' (8000); 'Runtime Settings' with checkboxes for 'Auto sync' (checked), 'Data sync always', 'Debug messages', and 'Record the needed sync time in system trace'; and 'Redundancy Link' with fields for 'IP address PLC 1' (192.168.0.10), 'IP address PLC 2' (192.168.0.11), and 'Port' (1205). There are 'Browse...' buttons next to the IP address fields and an unchecked checkbox for 'Use two redundant links' at the bottom.

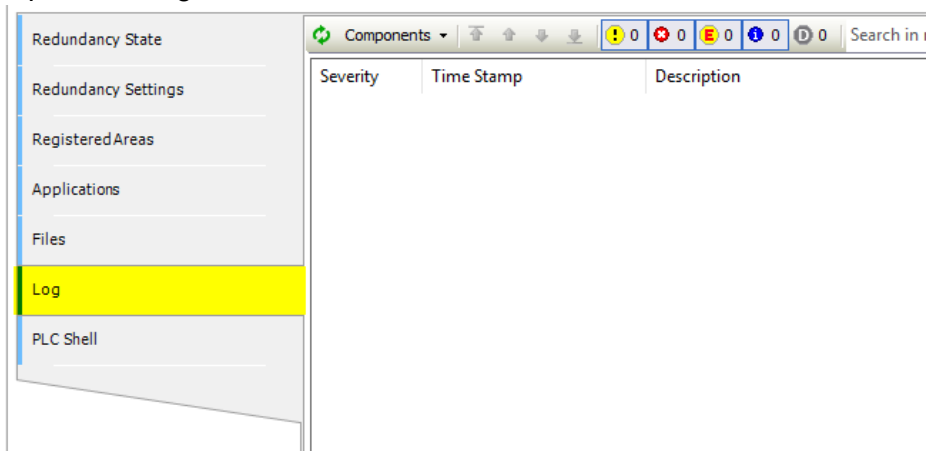
2. Click on “Write” to write the configuration to the controller.



This screenshot is identical to the previous one, but the 'Write' tab in the top navigation bar is now highlighted in yellow, indicating it has been selected. The 'Read' tab is no longer active.

Setting Up a Redundant Controller Architecture with UC20 Controllers and Modbus TCP I/O

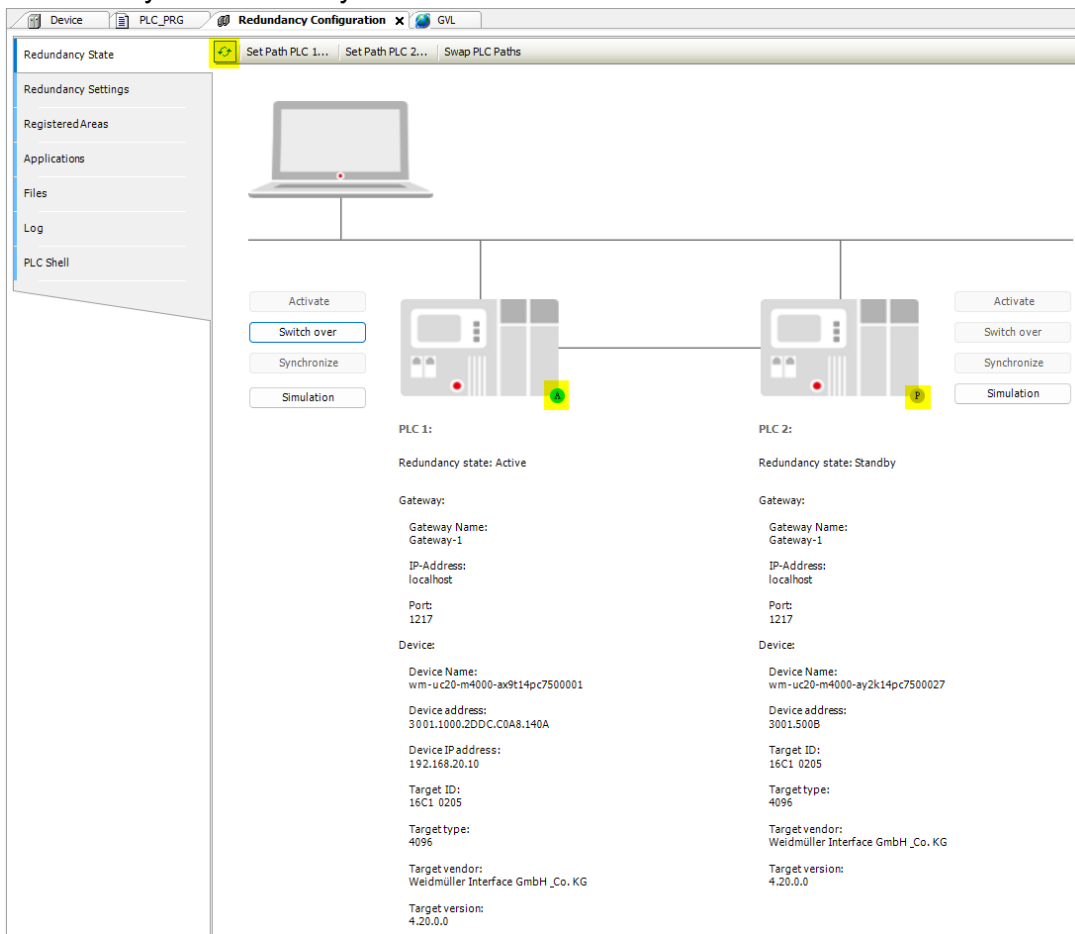
- Open the “Log” tab.



- Check for the following log entry and restart both controllers accordingly.

Severity	Time Stamp	Description	Component
Information	30.06.2026 14:44:05	Redundancy settings written. Restart plc to activate.	CmpRedundancy

- After both controllers were restarted the synchronization is executed automatically. If redundancy is successfully executed the state should be as follows.



A: Active Controller, P: Passive controller

5.8 Downloading the Application

After completing the controller redundancy configuration, the application must be downloaded to both controllers again so that the updated redundancy configuration becomes part of the running application.

Repeat the download procedure described in **Chapter 5.3 – Creating the CODESYS Project**.



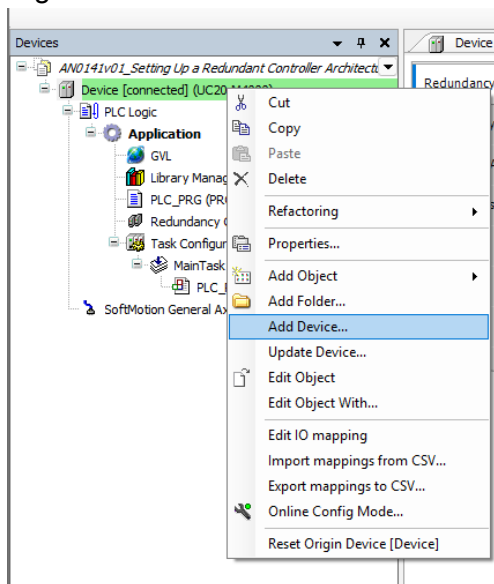
Note:

Ensure that the Update Boot Application option is enabled when downloading the application to both controllers.

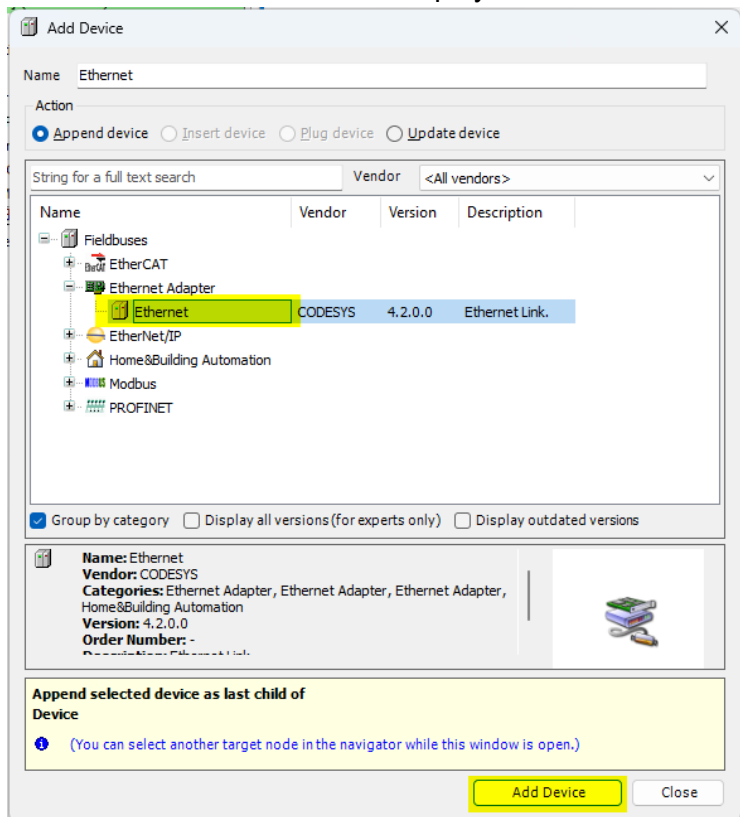
6 Setting Up Modbus TCP Communication in CODESYS

6.1 Adding the Ethernet Adapter

1. Right click on the “Device” in the device tree and then on “Add device”.

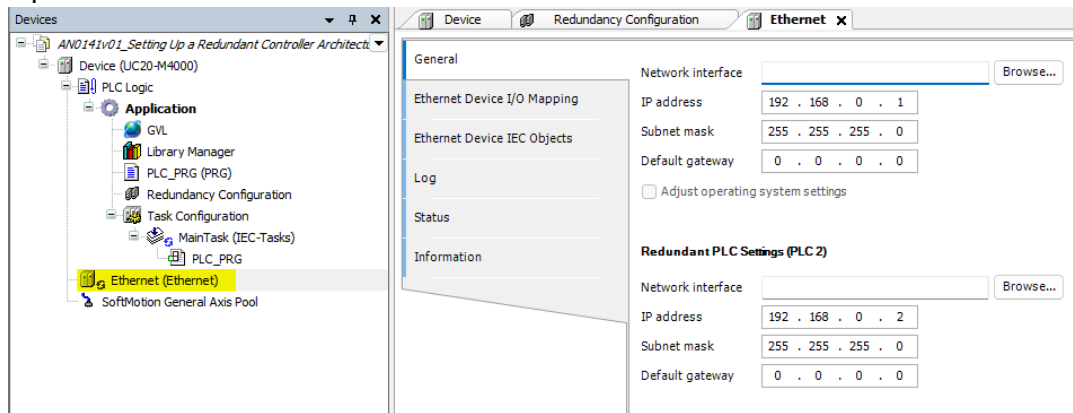


2. Add the Ethernet Interface as displayed below.

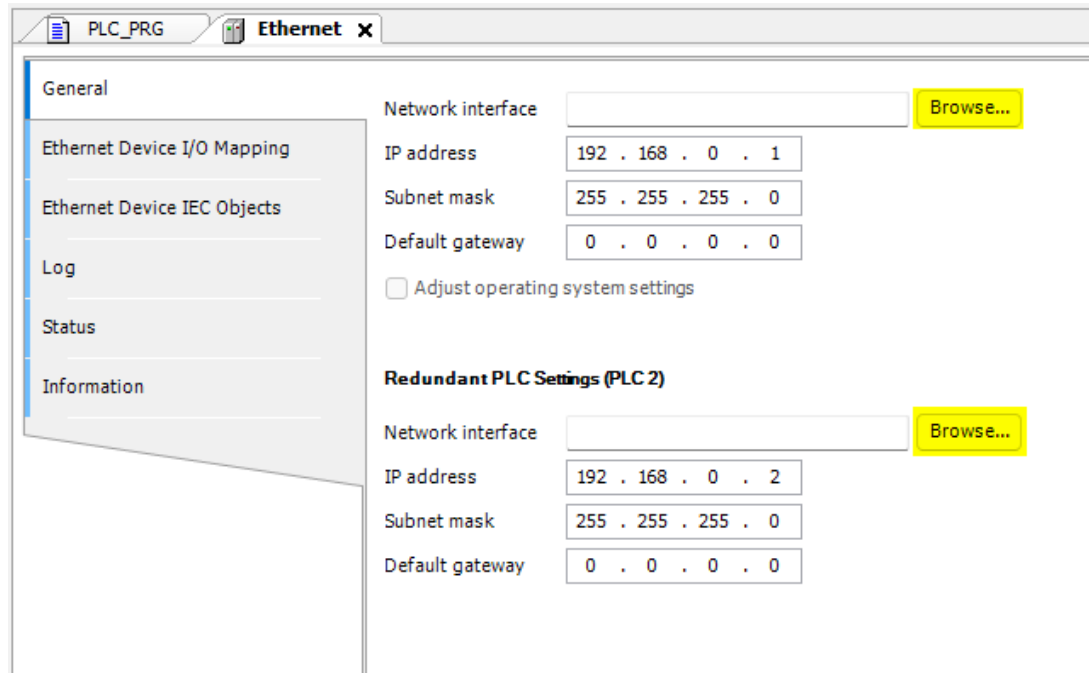


Setting Up a Redundant Controller Architecture with UC20 Controllers and Modbus TCP I/O

3. Open the Ethernet Interface in the device tree

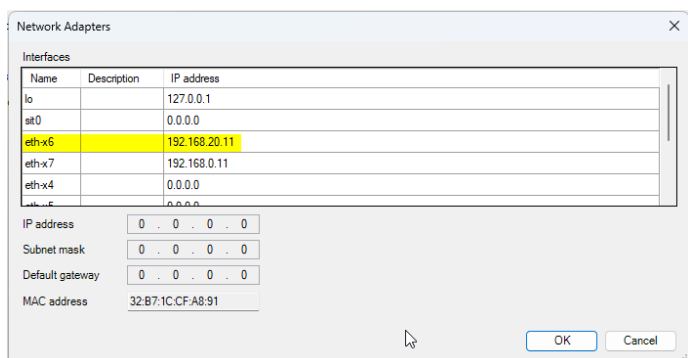
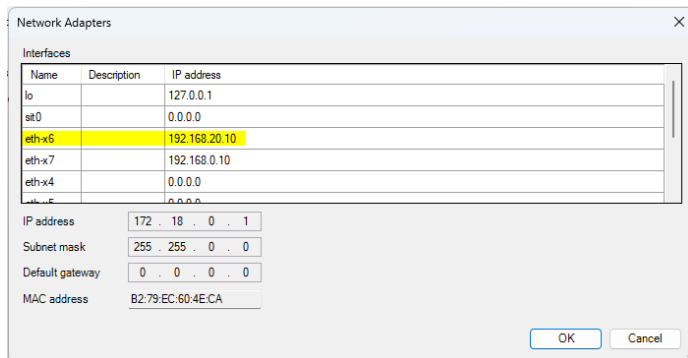
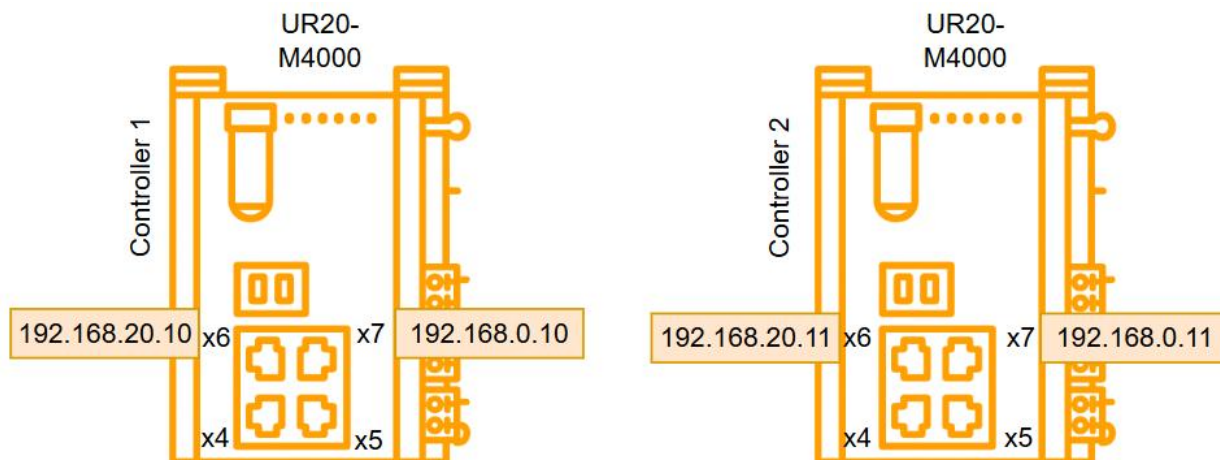


4. Assign the IP address for each controller by browsing the available interfaces



Setting Up a Redundant Controller Architecture with UC20 Controllers and Modbus TCP I/O

5. As shown in chapter 3.2 choose the Network interface **eth-x6** for both controllers



Note:

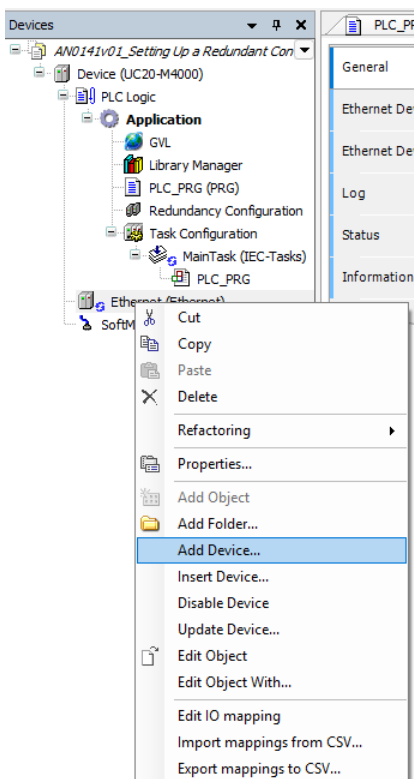
Do not enter the IP addresses manually. Always use the brows button otherwise this can lead to problems when writing the configuration to the controllers.

6. If correctly executed both IP addresses will be shown as follows:

The screenshot shows the 'General' tab of a configuration window. On the left is a sidebar with tabs: General, Ethernet Device I/O Mapping, Ethernet Device IEC Objects, Log, Status, and Information. The main area is divided into two sections. The top section has fields for 'Network interface' (eth-x6), 'IP address' (192 . 168 . 20 . 10), 'Subnet mask' (255 . 255 . 255 . 0), and 'Default gateway' (0 . 0 . 0 . 0). There is a checkbox for 'Adjust operating system settings'. The bottom section is titled 'Redundant PLC Settings (PLC 2)' and has identical fields for 'Network interface' (eth-x6), 'IP address' (192 . 168 . 20 . 11), 'Subnet mask' (255 . 255 . 255 . 0), and 'Default gateway' (0 . 0 . 0 . 0). Each IP address field is highlighted in yellow.

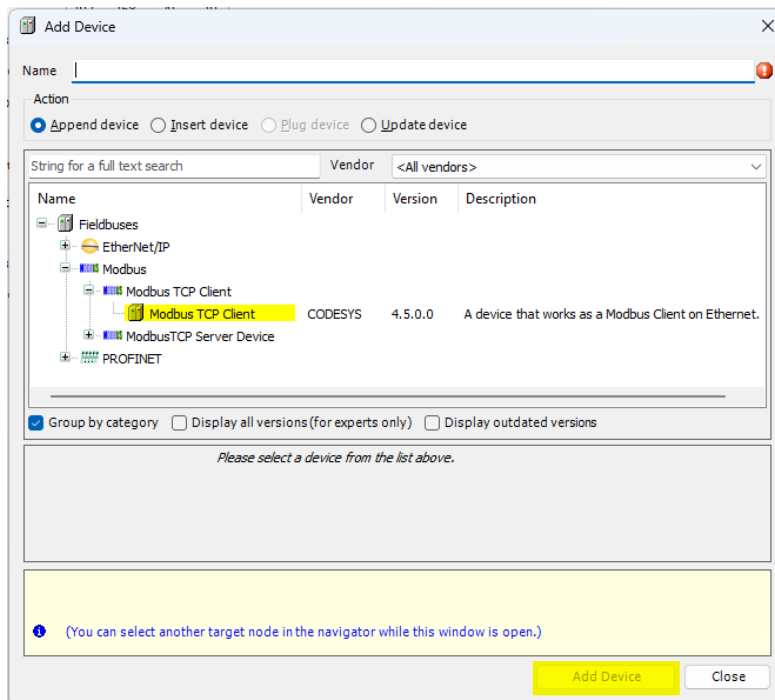
6.2 Adding the Modbus TCP Client

1. Add a Modbus TCP client by right clicking on the newly added Ethernet interface and on add device.

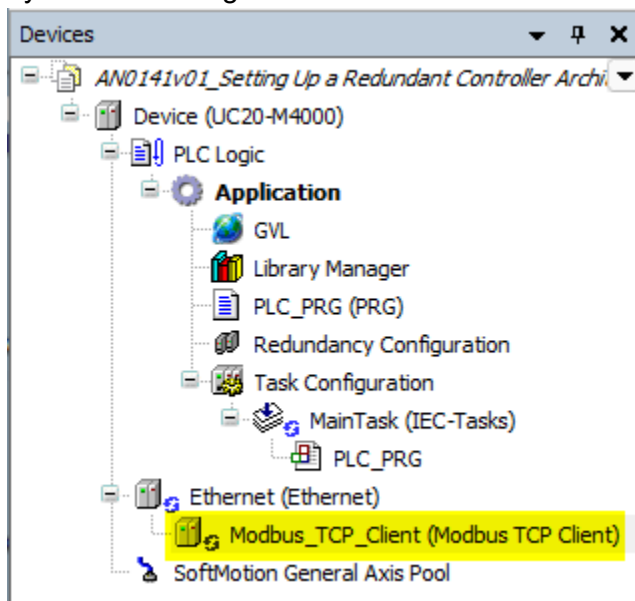


Setting Up a Redundant Controller Architecture with UC20 Controllers and Modbus TCP I/O

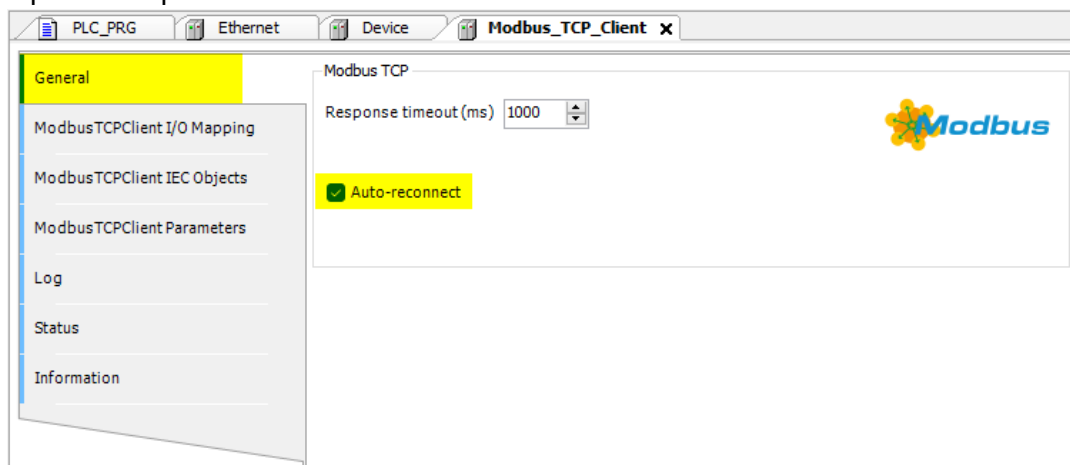
2. Select the Modbus TCP Client and click on “Add Device”.



3. The Modbus TCP Client will appear in the hardware tree on the left. Open the configuration by double clicking.

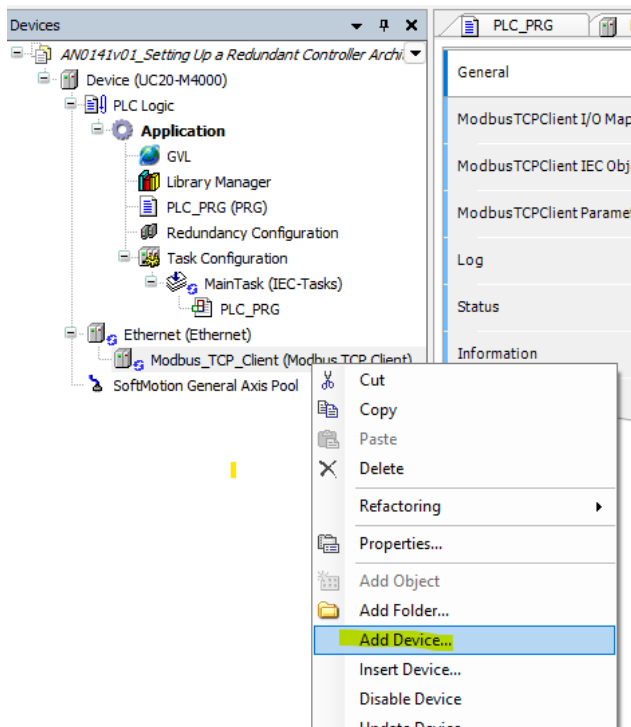


4. Open the top tab and make sure that “Auto-reconnect” is selected.

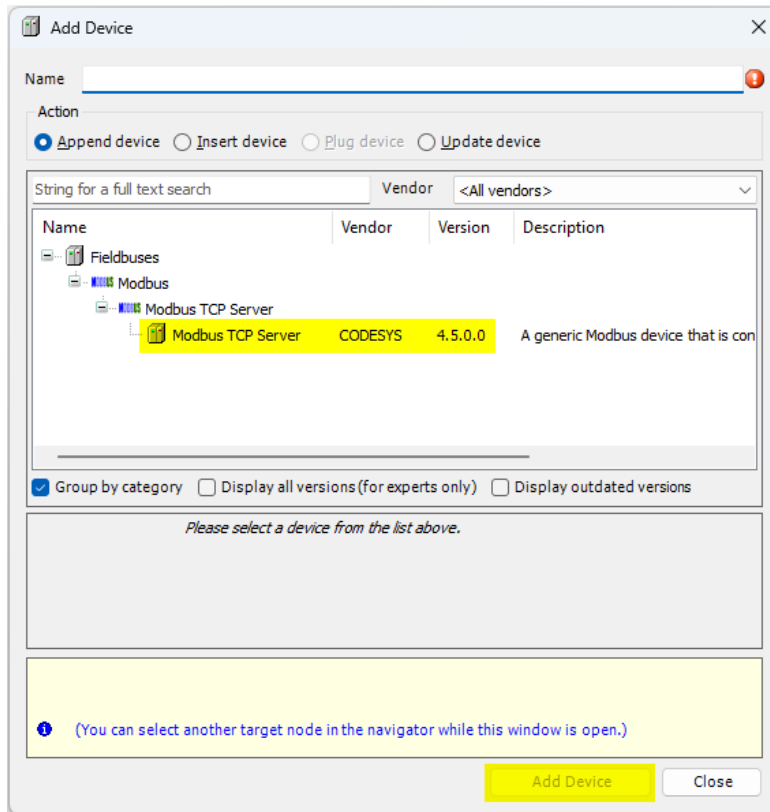


6.3 Adding the Modbus TCP Server

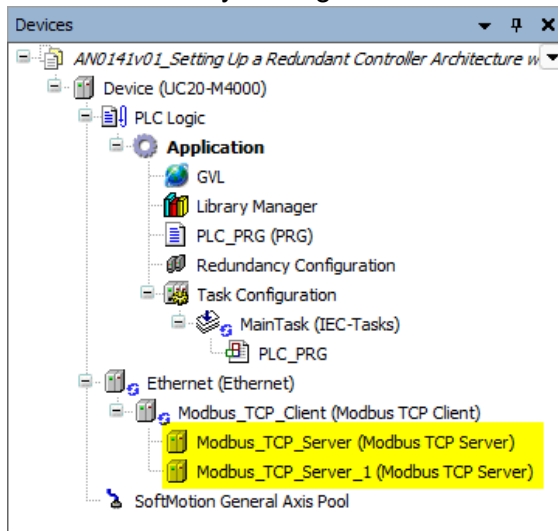
1. Right click on the newly added Modbus TCP client in the hardware tree and on “Add device”



2. Select the Modbus TCP Client and click on “Add Device”.

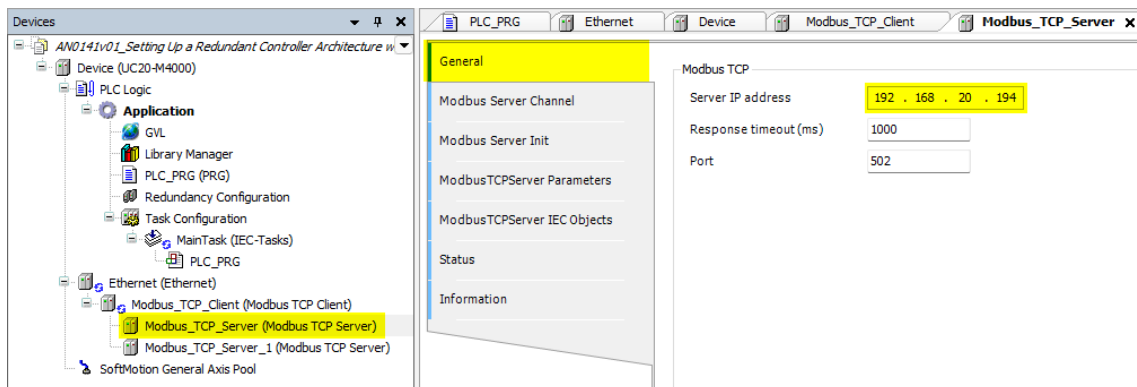


3. Since we have two Modbus TCP fieldbus couplers (see chapter 3.4 Network Topology) the process must be repeated for the second one.
4. After successfully adding both Modbus TCP Servers they will appear in the hardware tree.

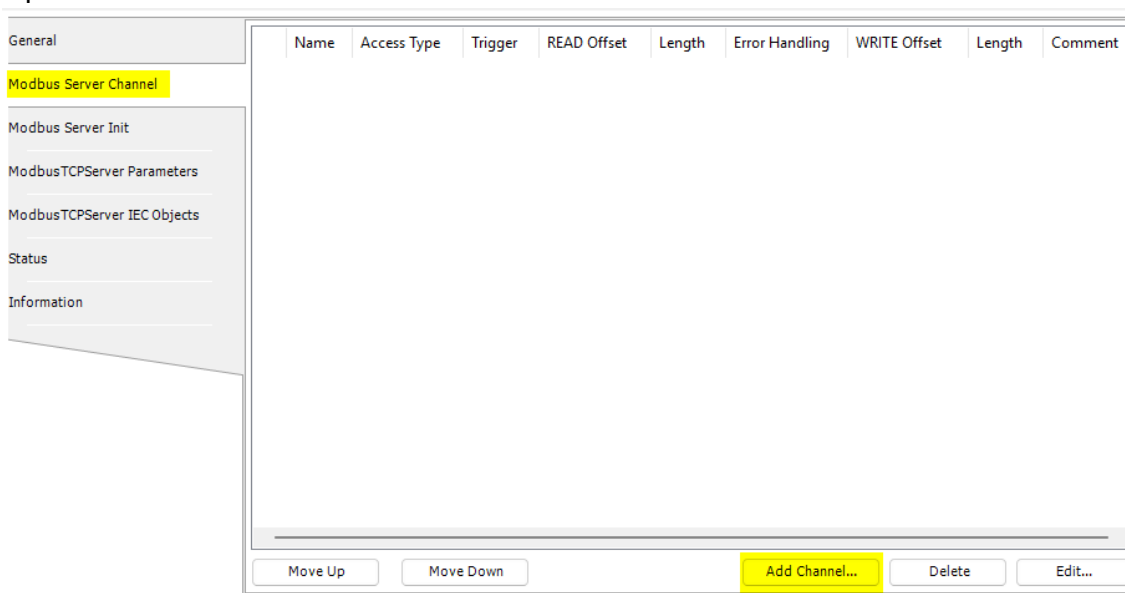


6.4 Configuring the Modbus TCP Couplers in CODESYS

1. Open the Configuration by double clicking on the first Modbus Server and open the first tab "General". Set the IP address to 192.168.20.194

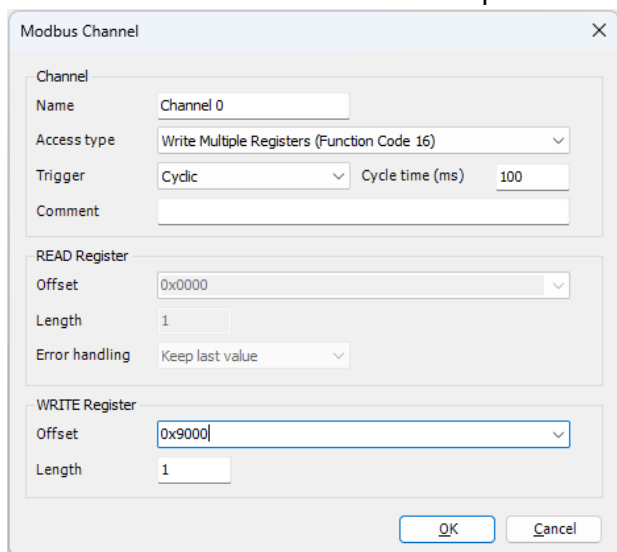


2. Open the second tab "Modbus Server Channel" and click on "Add Channel"



Setting Up a Redundant Controller Architecture with UC20 Controllers and Modbus TCP I/O

3. To write data to an UR20-8DO-P output card configure the channel as follows:



The 'Modbus Channel' dialog box is shown with the following settings:

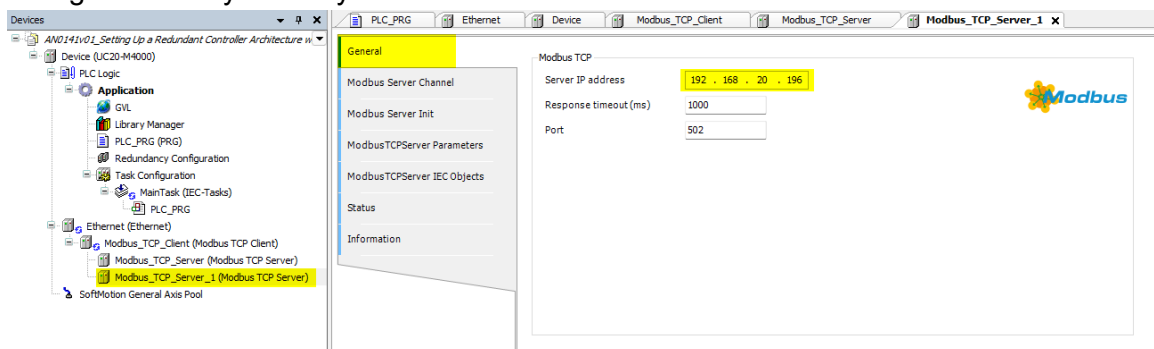
- Channel:**
 - Name: Channel 0
 - Access type: Write Multiple Registers (Function Code 16)
 - Trigger: Cyclic
 - Cycle time (ms): 100
 - Comment: (empty)
- READ Register:**
 - Offset: 0x0000
 - Length: 1
 - Error handling: Keep last value
- WRITE Register:**
 - Offset: 0x9000
 - Length: 1

Buttons: OK, Cancel

4. If successfully conducted the newly configured channel will appear as follows:

General										
Modbus Server Channel										
	Name	Access Type	Trigger	READ Offset	Length	Error Handling	WRITE Offset	Length	Comment	
0	Channel 0	Write Multiple Registers (Function Code 16)	Cyclic, t#100ms				16#9000	1		

5. Repeat step one to four for the second Modbus TCP Server. The IP address must be chosen differently. Use for this use case the IP address 192.168.20.196. The channel configuration stays exactly the same for the second Modbus TCP Server.



The screenshot shows the 'Modbus TCP Server' configuration window. The 'General' tab is selected, showing the following settings:

- Server IP address: 192.168.20.196
- Response timeout (ms): 1000
- Port: 502

The left sidebar shows the project tree with 'Modbus_TCP_Server_1 (Modbus TCP Server)' selected. The top bar shows the 'Modbus_TCP_Server' tab is active.

6.5 Downloading the Application

After completing the configuration, the application must be downloaded to controller again so that the updated configuration becomes part of the running application.

Repeat the download procedure described in **Chapter 5.3 – Creating the CODESYS Project**.



Note:

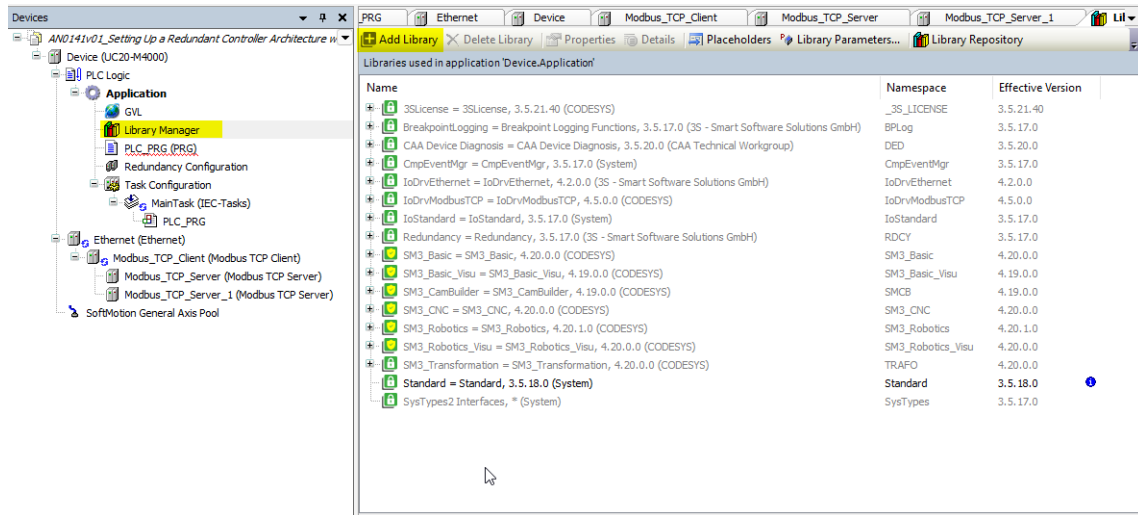
Ensure that the Update Boot Application option is enabled when downloading the application to both controllers.

After downloading the application to both controllers, verify that the applications are running and that the redundancy state is correctly established before proceeding with the Modbus TCP configuration.

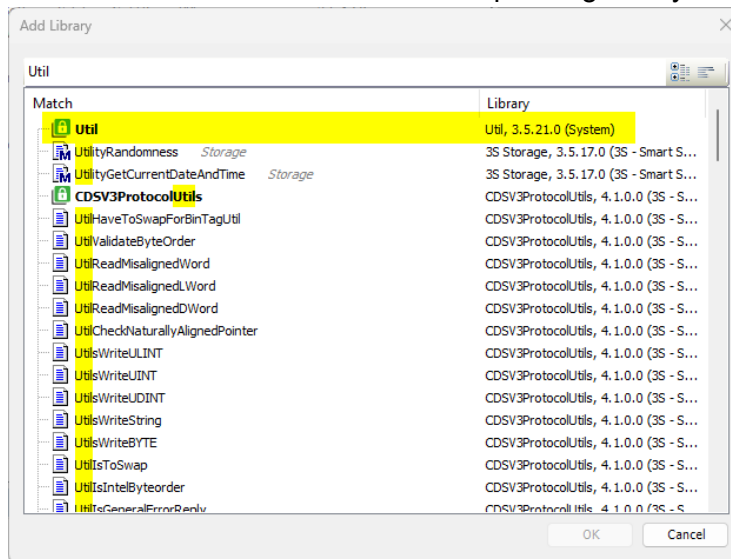
7 Set up a minimal application

To see that the Modbus TCP fieldbus couplers are being actively controlled it makes sense to write a small application to generate any output at all. The goal is to have chaser lights running each led on the UR20-8DO-P for and back.

1. Add the Util library by opening the library manager and click on add library.

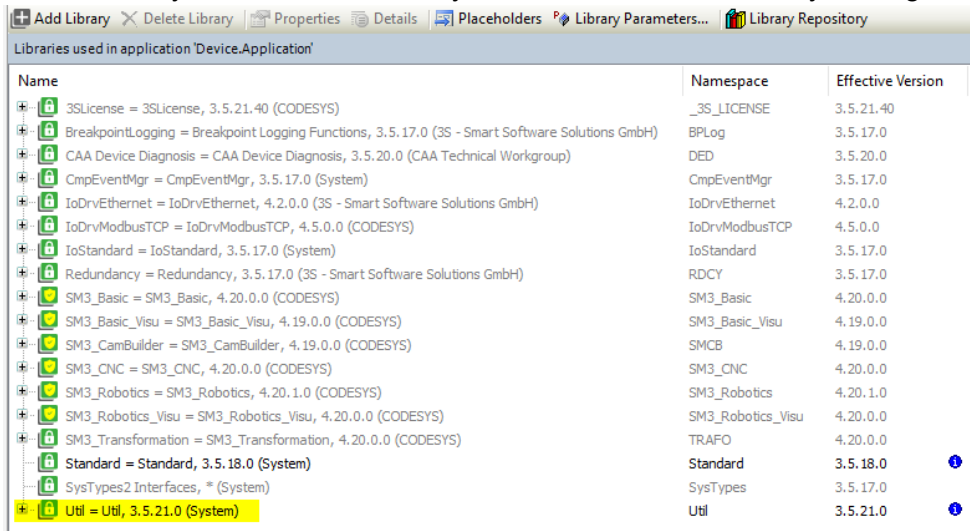


2. Search für “Util” and select the corresponding DeviceApplication library and click on ok.



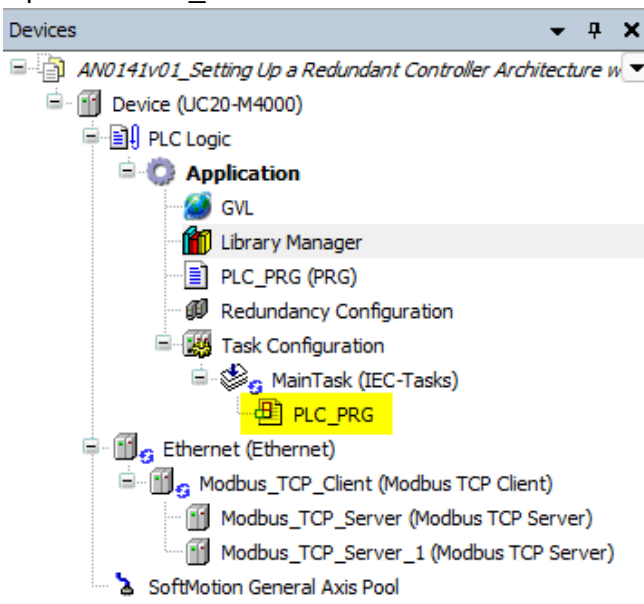
Setting Up a Redundant Controller Architecture with UC20 Controllers and Modbus TCP I/O

3. If successfully conducted the library will be visible in the Library Manager.



Name	Namespace	Effective Version
3SLicense = 3SLicense, 3.5.21.40 (CODESYS)	_3S_LICENSE	3.5.21.40
BreakpointLogging = Breakpoint Logging Functions, 3.5.17.0 (3S - Smart Software Solutions GmbH)	BPLog	3.5.17.0
CAA Device Diagnosis = CAA Device Diagnosis, 3.5.20.0 (CAA Technical Workgroup)	DED	3.5.20.0
CmpEventMgr = CmpEventMgr, 3.5.17.0 (System)	CmpEventMgr	3.5.17.0
IoDrvEthernet = IoDrvEthernet, 4.2.0.0 (3S - Smart Software Solutions GmbH)	IoDrvEthernet	4.2.0.0
IoDrvModbusTCP = IoDrvModbusTCP, 4.5.0.0 (CODESYS)	IoDrvModbusTCP	4.5.0.0
IoStandard = IoStandard, 3.5.17.0 (System)	IoStandard	3.5.17.0
Redundancy = Redundancy, 3.5.17.0 (3S - Smart Software Solutions GmbH)	RDCY	3.5.17.0
SM3_Basic = SM3_Basic, 4.20.0.0 (CODESYS)	SM3_Basic	4.20.0.0
SM3_Basic_Visu = SM3_Basic_Visu, 4.19.0.0 (CODESYS)	SM3_Basic_Visu	4.19.0.0
SM3_CamBuilder = SM3_CamBuilder, 4.19.0.0 (CODESYS)	SMCB	4.19.0.0
SM3_CNC = SM3_CNC, 4.20.0.0 (CODESYS)	SM3_CNC	4.20.0.0
SM3_Robotics = SM3_Robotics, 4.20.1.0 (CODESYS)	SM3_Robotics	4.20.1.0
SM3_Robotics_Visu = SM3_Robotics_Visu, 4.20.0.0 (CODESYS)	SM3_Robotics_Visu	4.20.0.0
SM3_Transformation = SM3_Transformation, 4.20.0.0 (CODESYS)	TRAFO	4.20.0.0
Standard = Standard, 3.5.18.0 (System)	Standard	3.5.18.0
SysTypes2.Interfaces, * (System)	SysTypes	3.5.17.0
Util = Util, 3.5.21.0 (System)	Util	3.5.21.0

4. Open the PLC_PRG in the device tree.



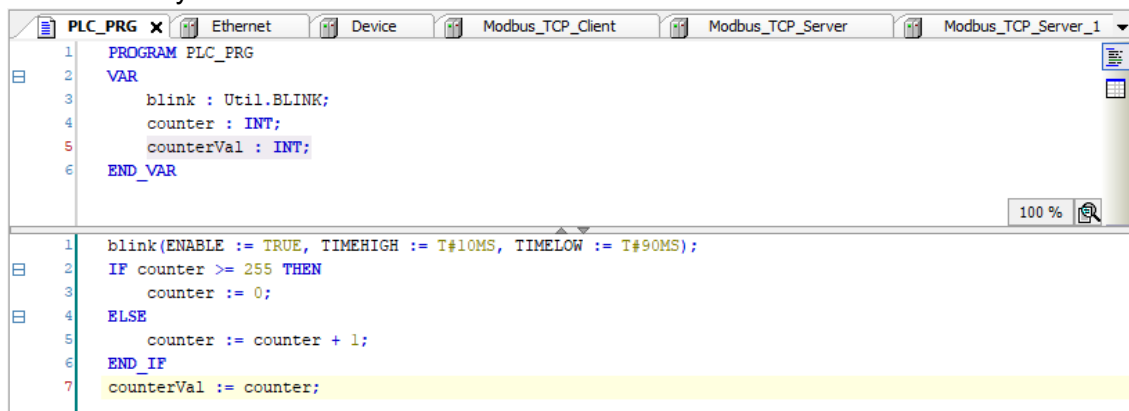
Setting Up a Redundant Controller Architecture with UC20 Controllers and Modbus TCP I/O

5. Copy the following code snippet into the program:

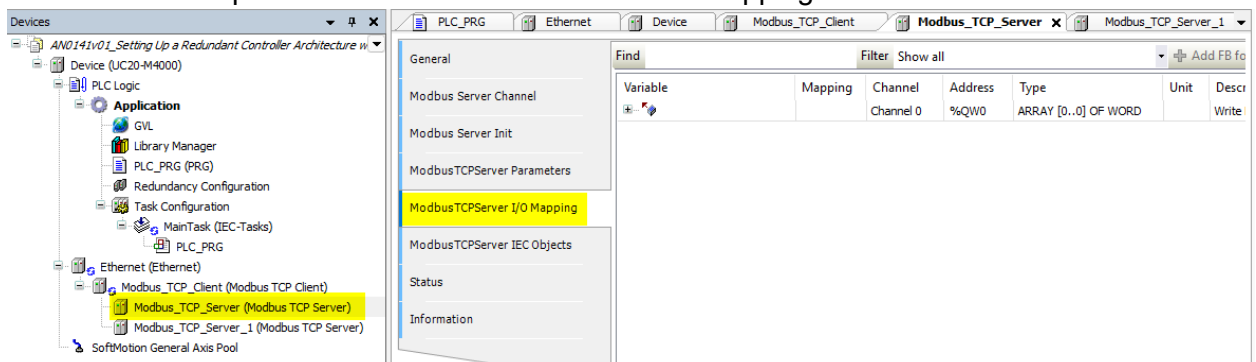
```
PROGRAM PLC_PRG
VAR
    blink : Util.BLINK;
    counter : INT;
    counterVal : INT;
END_VAR

blink(ENABLE := TRUE, TIMEHIGH := T#10MS, TIMELOW := T#90MS);
IF counter >= 255 THEN
    counter := 0;
ELSE
    counter := counter + 1;
END_IF
counterVal := counter;
```

6. If successfully conducted it will look like this.



7. Map the PV “counterVal” to the output card by opening the Modbus TCP Server in the device tree and open the tab “ModbusTCPServer I/O Mapping”.

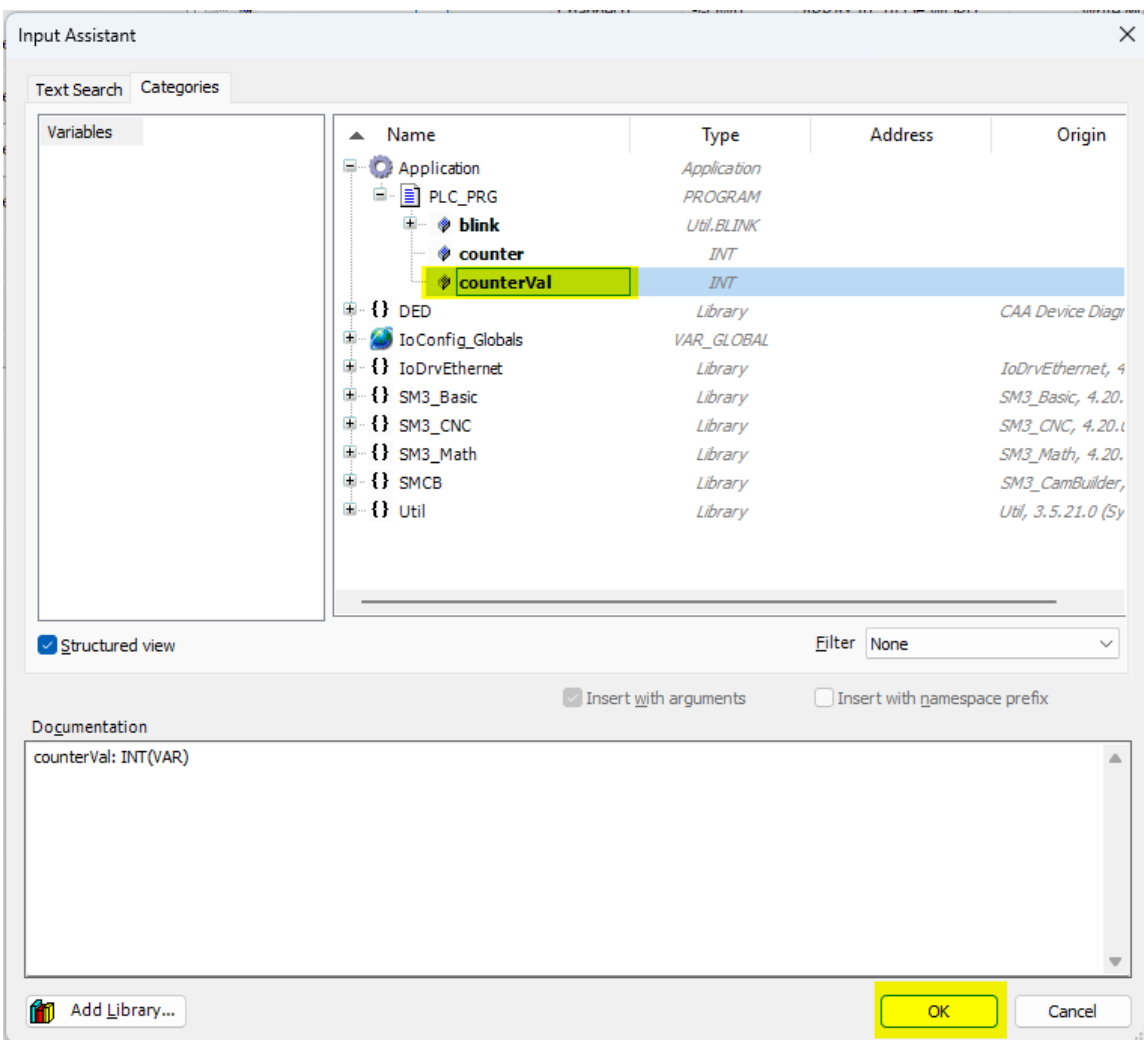


Setting Up a Redundant Controller Architecture with UC20 Controllers and Modbus TCP I/O

8. Map the PV by double click the first row in the most left field. Then click on the three dots which appear.

Find	Filter	Show all	Add FB for IO Channel...			
Variable	Mapping	Channel	Address	Type	Unit	Description
		Channel 0	%QW0	ARRAY [0..0] OF WORD		Write Multiple Registers
		Channel 0[0]	%QW0	WORD		0x9000

9. Search for the PV “counterVal” and click on ok.



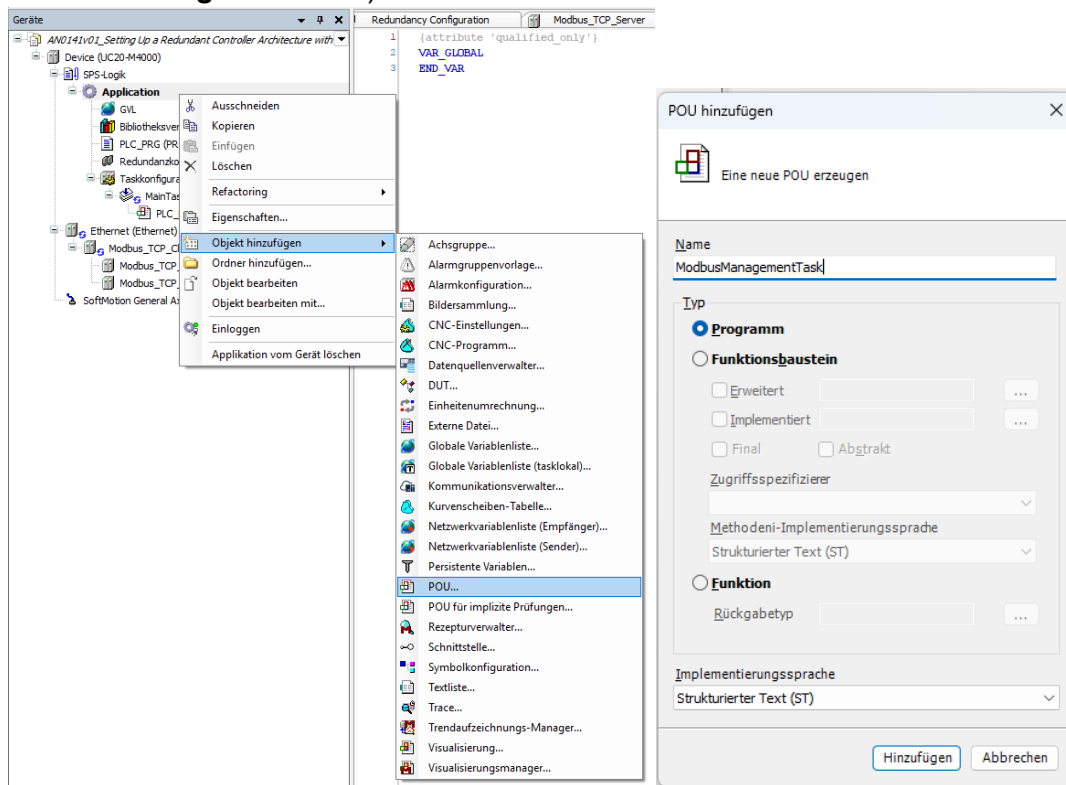
10. Repeat the steps seven to nine exactly for the second Modbus TCP Server.

7.1 Add the Modbus TCP Redundancy Manager

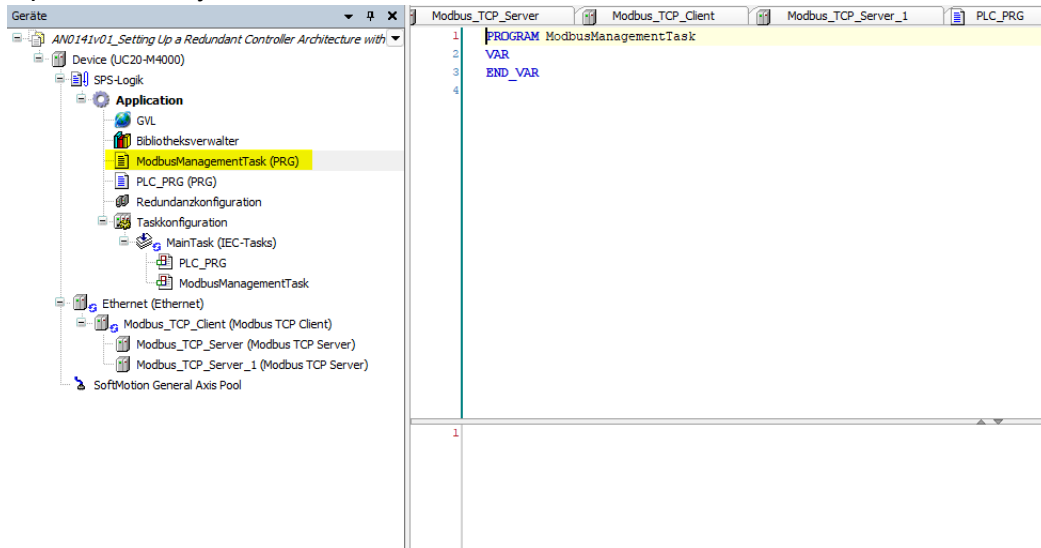
To support redundant Modbus TCP communication, the **RedundancyManager** function block must be integrated into the application. The function block supervises the redundancy state of the controller and determines which PLC is allowed to establish and maintain the Modbus TCP connections.

Unlike the application logic, the RedundancyManager must **not** execute in the cyclic synchronized application task. Instead, it should be executed in a dedicated task that is synchronized only once during startup. This allows both controllers to evaluate their redundancy state independently while the application data remains synchronized.

1. Create a new task in the CODESYS Task Configuration (for example **ModbusManagementTask**).



2. Open the newly created Task.



3. Add the following code to ModbusManagementTask.

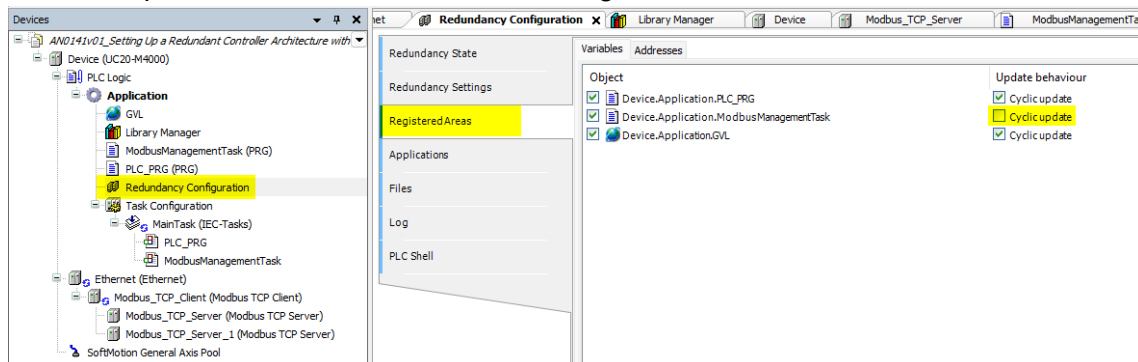
```
PROGRAM ModbusManagementTask
VAR
    fbModbusTcpMgn : RedundancyManager;
    tonTimeout      : TON;
    tPresetTime     : TIME := T#10S;
END_VAR

// Start the fallback timer if no Modbus redundancy role is nor in
// sync.
tonTimeout(
    IN := NOT (fbModbusTcpMgn.xSync),
    PT := tPresetTime
);

// Enable forced startup if no redundancy role has been detected
// within the configured timeout.
fbModbusTcpMgn.xForceStart := tonTimeout.Q;

// Call the Modbus TCP redundancy manager cyclically and assign
// the configured Modbus TCP Client.
fbModbusTcpMgn(
    client := Modbus_TCP_Client
);
```

4. Open the “Redundancy Configuration” and navigate to the tab “Registered Areas”. Make sure the Update behavior for the ModbusmanagementTask is disabled.



Note:

The Modbus TCP management logic described in Chapters 7.5 and 7.6 must be executed within the redundant application task. Ensure that PLC_PRG is included in the registered redundancy areas and assigned to the redundancy task as described in Chapter 5.6.1. Otherwise, the application state required for a coordinated Modbus TCP switchover may not be synchronized between the two controllers.



Note:

The timeout is configured to 10 seconds, which is intentionally greater than the configured Bootup timeout of 8 seconds. This ensures that the controllers have sufficient time to complete their startup and synchronization before the Modbus TCP management is forced to start. A shorter timeout could cause xForceStart to be activated during the normal boot process.

7.2 Downloading the Application

After completing the configuration, the application must be downloaded to controller again so that the updated redundancy configuration becomes part of the running application.

Repeat the download procedure described in **Chapter 5.3 – Creating the CODESYS Project**.



Note:

Ensure that the Update Boot Application option is enabled when downloading the application to both controllers.

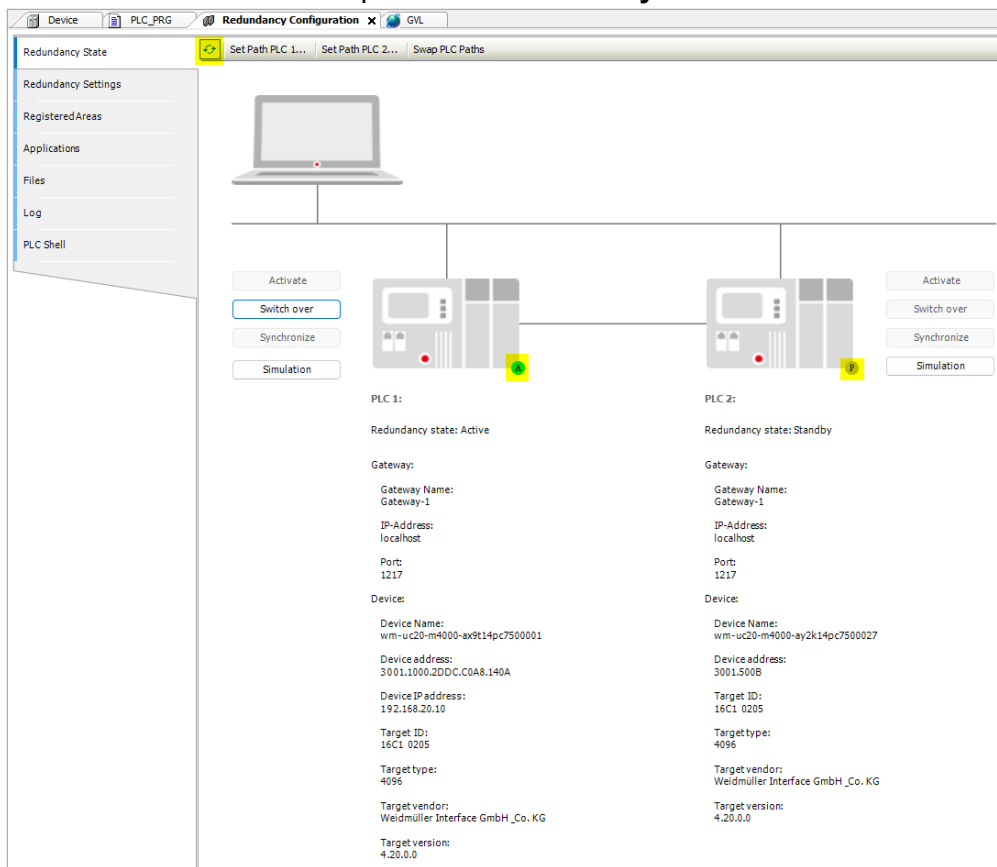
8 Validation

8.1 Verify the Initial System State

This chapter verifies that the complete redundant system has been commissioned successfully before any failure scenarios are introduced.

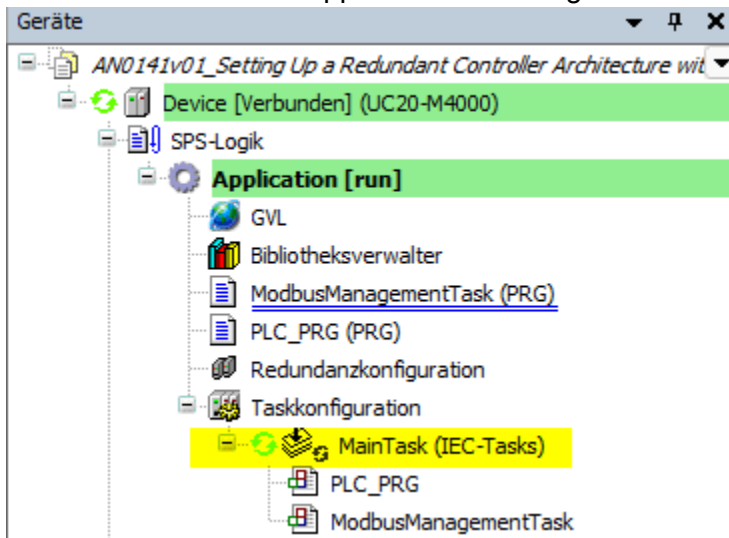
Verify that:

1. One controller operates as **Primary**.
2. The second controller operates as **Secondary**.

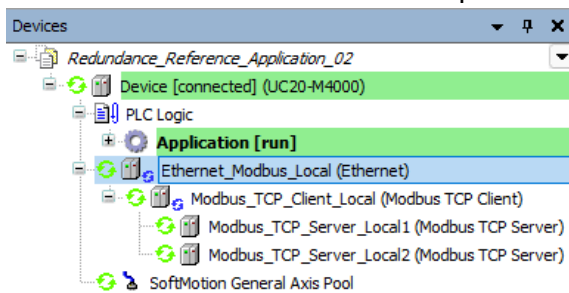


Setting Up a Redundant Controller Architecture with UC20 Controllers and Modbus TCP I/O

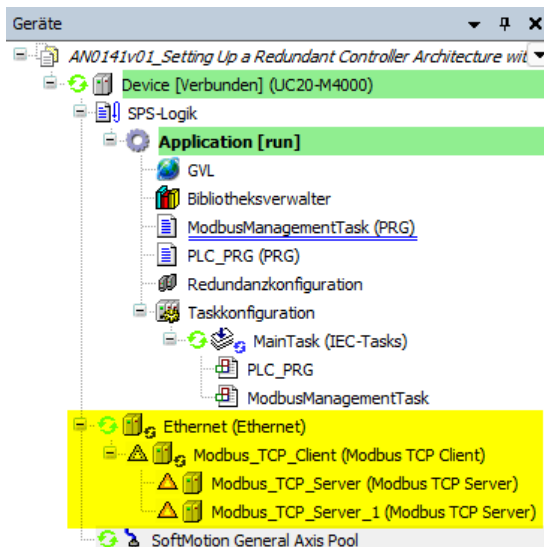
3. The demonstration application is running.



4. Both Modbus TCP fieldbus couplers are connected.



Active Controller



Passive Controller



Note:

The Modbus TCP fieldbus couplers are intentionally indicated differently on the Active and Standby controllers. On the Active controller, the Modbus TCP communication is enabled and the configured fieldbus couplers are shown as connected. On the Standby controller, the RedundancyManager keeps the Modbus TCP communication inactive, as the Standby controller must not establish and maintain the fieldbus connections simultaneously with the Active controller. The corresponding Modbus TCP devices may therefore be displayed as inactive or with a different status indication.

This behavior is expected and does not indicate a communication fault. After a controller switchover, the Modbus TCP communication is transferred to the new Active controller and the status indications of the two controllers change accordingly.

5. The digital outputs change according to the demonstration application.
6. No controller redundancy or Modbus TCP communication errors are reported.

8.2 Verify Controller Switchover

This chapter verifies that controller redundancy operates correctly during a controller failure.

1. Verify that the system is operating normally.
2. Disconnect the power supply of the Primary controller.
3. Observe the controller status.
4. Verify that the Secondary controller assumes the Primary role.
5. Verify that the demonstration application continues to run.
6. Verify that communication with both Modbus TCP fieldbus couplers is automatically re-established.
7. Restore power to the failed controller.
8. Verify that the restarted controller rejoins the redundancy pair as Secondary.

Expected Result

- Automatic controller switchover.
- Continuous execution of the demonstration application.
- Automatic recovery of the Modbus TCP communication.
- The restarted controller synchronizes and returns to the Secondary state.

8.3 Verify the Redundant Modbus TCP Communication

This chapter verifies that the redundant Modbus TCP communication behaves as expected during controller switchover.

Procedure:

1. Observe the digital outputs controlled via the Modbus TCP fieldbus couplers.
2. Trigger several controller switchovers by alternately powering down each controller.
3. Verify that the output sequence continues after each switchover.
4. Verify that no permanent communication errors occur.
5. Verify that both fieldbus couplers remain operational after each switchover.

Expected Result

- The Redundancy Manager transfers ownership of the Modbus TCP client to the active controller.
- Communication with the fieldbus couplers is automatically re-established after each controller switchover.
- The demonstration application continues to operate without manual intervention.