2025

MINT V2.0.1

GUIDELINE REV2.4

MICHEAS GOETHALS

PHOENIX CONTACT

Table of Contents

1 Terminology 4
2 Required knowledge 5
3 MINT configuration preparation 6
3.1 Download MINT configuration 6
3.2 Creating a MINT project in CodeCreator 9
3.3 Importing the “Code Generator” (.zip) 14
3.4 Creating a MINT Project 17
4 Preparing the PLC 25
4.1 Firmware update 25
4.2 System services 27
4.3 Firewall settings 27
4.4 NTP synchronisation 28
4.5 Adding MQTT Certificate to the PLC 28
4.6 Checking the PLC via Linux 30
4.6.1 Check the firmware 31
4.6.2 Checking the NTP server 32
5 MULTI PLC 33
5.1 Download site configuratie 33
5.2 MULTI PLC certification 33
5.2.1 Trust Stores 33
5.2.2 Identity Stores 35
5.3 Firewall settings 37
5.4 Changes in PLCnext Engineer 38
5.4.1 MAIN PLC 38
5.4.2 SUB PLC 1 “Sub_1” 38
5.4.3 MAIN PLC 2 “Sub_2” 38
5.5 MUTLI PLC / MINT Licentie 39
5.6 MUTLI PLC / MQTT Credentials 39
6 Starting a MINT project 40
6.1 Starting the MINT Project / MULTI PLC 41
7 Verifying the MINT Project 42
7.1 MINT Licence Active 42

7.2 MQTT Connection Active 42
7.3 Collector data 43
7.4 Charging Station Data 43
7.5 MINT Scaling 44

Table of Revisions

Date	Version	Description of Changes	Status	Author
06-03-2025	1.0	Concept	Draft	Micheas Goethals
07-03-2025	2.0	Addition of MULTI PLC	Draft	Micheas Goethals
11-03-2025	2.1	Update to V2.0.1	Draft	Micheas Goethals
07-04-2025	2.2	Expansion of descriptions	Draft	Femke Parthoens
06-10-2025	2.3	Firmware change for PLC AXC 2152	Done	Micheas Goethals
20-11-2025	2.4	Basis firewall settings change	Done	Micheas Goethals

1 Terminology

- **MINT Core**; A MINT Core system is the control system responsible for continuously monitoring all defined limits.

- **MINT Advanced**; As soon as an additional feature is included in a MINT installation, it is referred to as MINT Advanced. These features may include generating optimised charging recommendations followed by the PLC, or visualisation. The MINT Core is always present and continuously monitors the fuse.

- **Multi PLC**; When the number of assets exceeds the limits for a single PLC and multiple PLCs are required for one MINT installation, this is referred to as a Multi PLC system. One PLC will act as the Main PLC, while the others are Sub PLCs.

- **Main**; Refers to the primary PLC in a Multi PLC system. This PLC is configured with the full installation to be monitored and will calculate all asset setpoints and forward them to the required Sub PLCs.

- **Sub**; Refers to PLCs subordinate to the Main PLC. A Sub PLC handles communication with part of the field devices, sends data to the Main PLC, and executes the calculated commands received from the Main PLC. A Sub PLC does not perform any calculations itself.

- **Advice**; A recommendation generated by an optimiser to control assets in an optimal way. A simple example is postponing charging sessions until solar intensity increases later in the day, resulting in lower charging costs and reduced CO_2 impact.

- **NTP (Network Time Protocol)**; NTP defines the behaviour of systems within the same network that compare their configured time settings to ensure each device maintains accurate time. A standalone device may drift over time. Specialised NTP servers maintain accurate time and can be used for synchronisation.

- **Firewall**; A device's firewall is a network security system that determines whether specific network traffic is allowed to reach the device. In the case of a private firewall, such as that of a PLC, it can be configured directly on the device. It may also be necessary to adjust settings in the overarching network firewall in which the PLC operates to realise the project.

2 Required knowledge

To create MINT projects, experience with PLCnext Engineer and Code Creator, along with familiarity with firewalls, Modbus, MQTT, Linux, and the principles of electricity, is ideal.

However, using this step-by-step plan (or guideline) and the described tools, a large part of the preparations can be carried out with minimal prior knowledge.

For final verification, it is strongly recommended that someone with insight into the electrical network reviews the preparations.

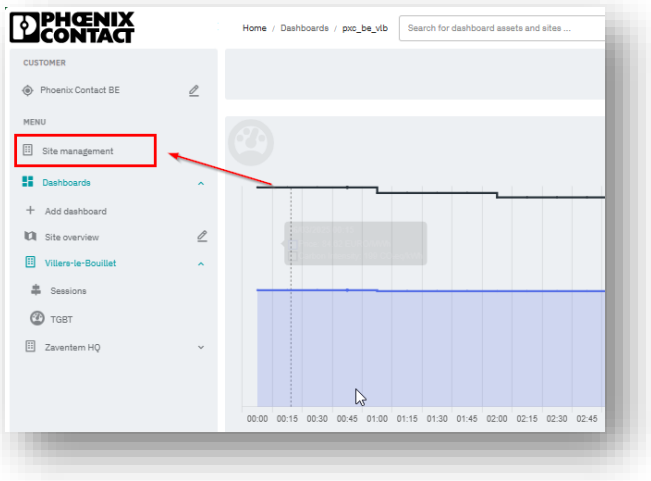
3 MINT configuration preparation

3.1 Download MINT configuration

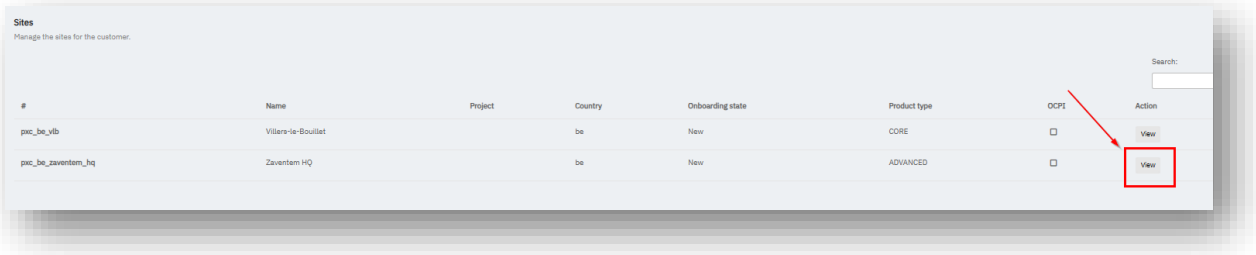
The site configuration can be downloaded via the MINT Portal.
Follow the steps below:

Go to the following URL to access the MINT Portal:
[| Phoenix Contact Mint]

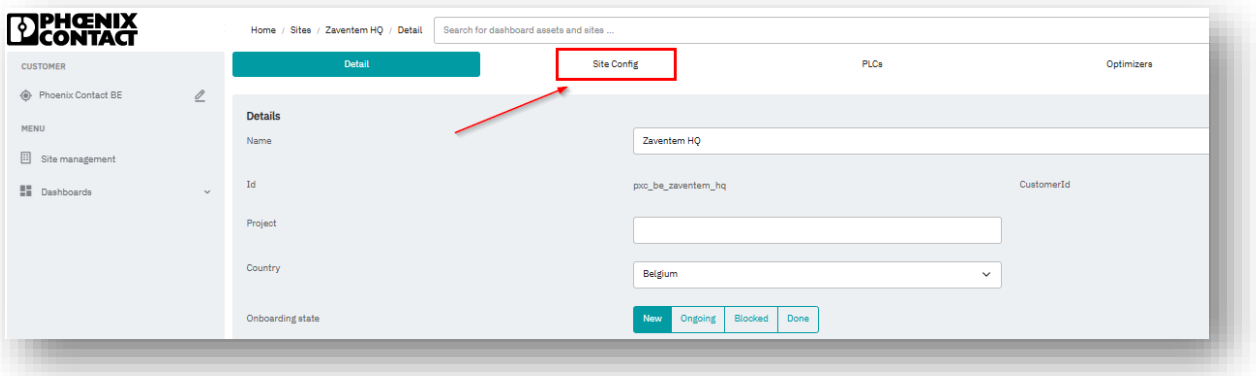
Navigate to “site management”.



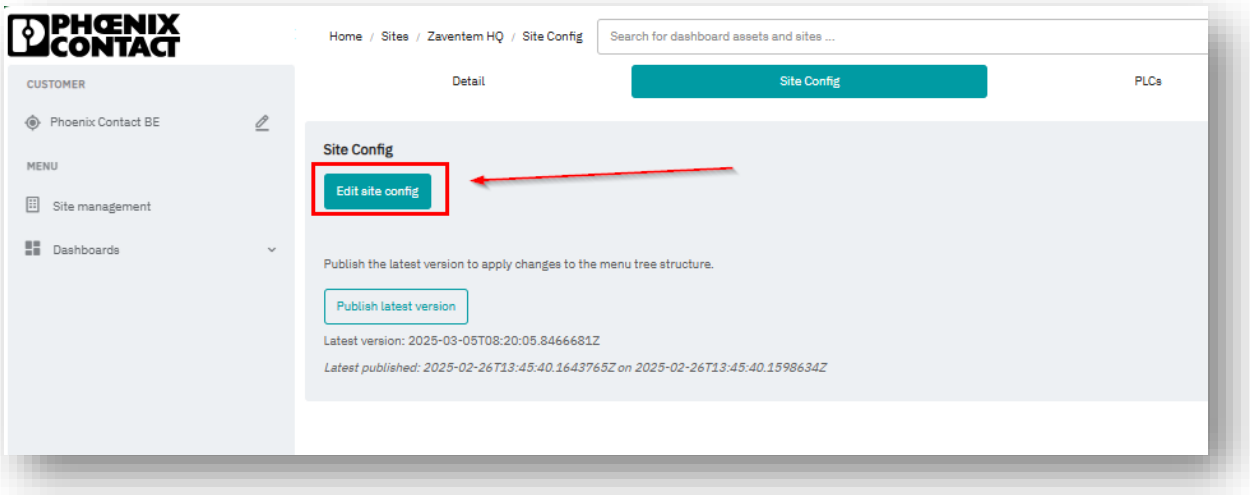
Navigate to “view”.



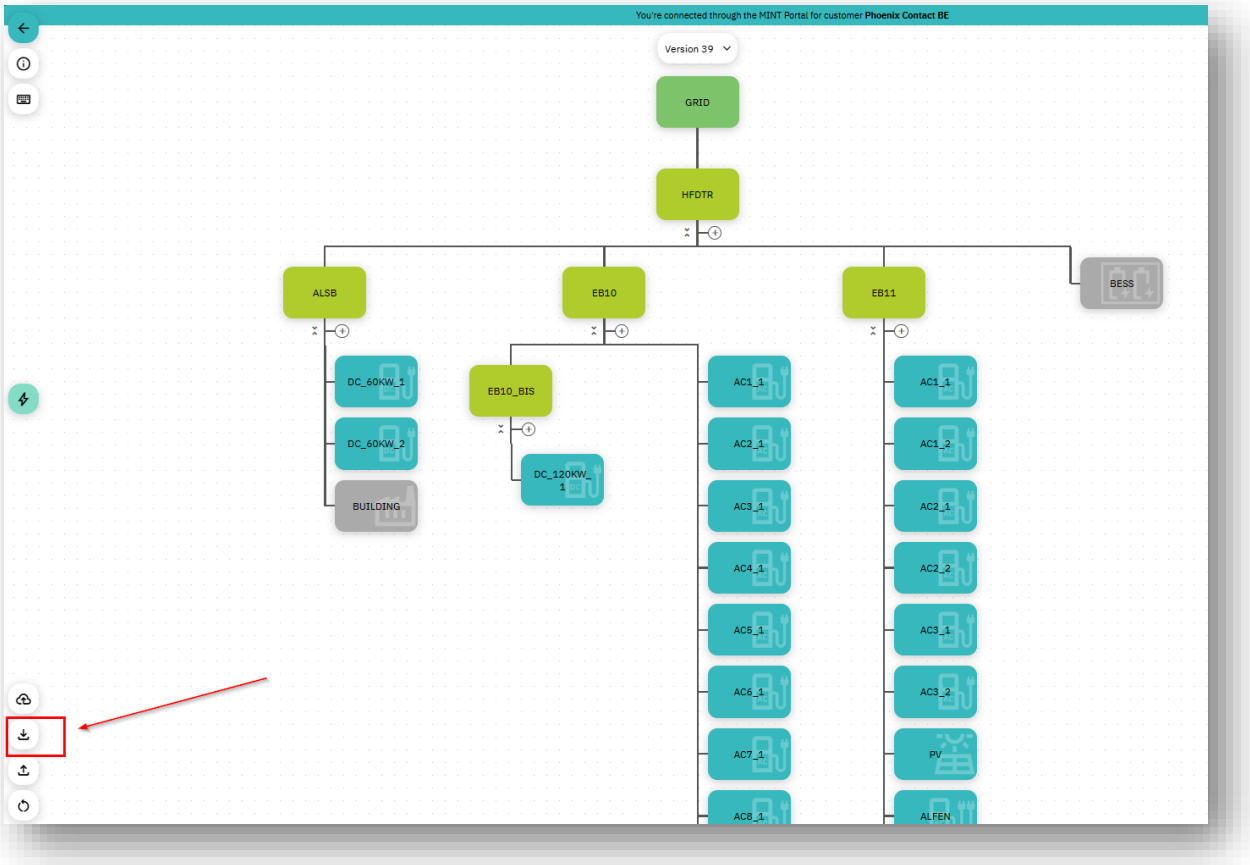
Navigate to “Site Config”



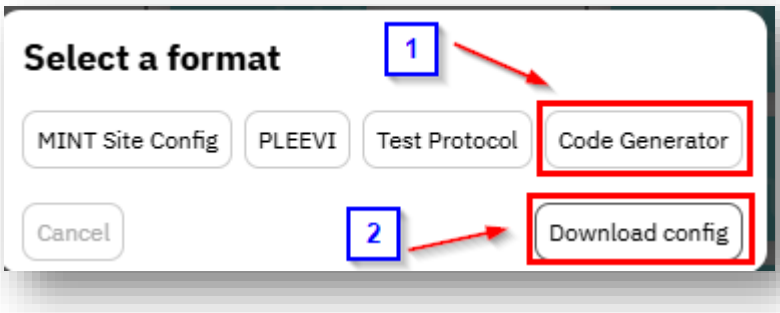
Navigate to “Edit site config”

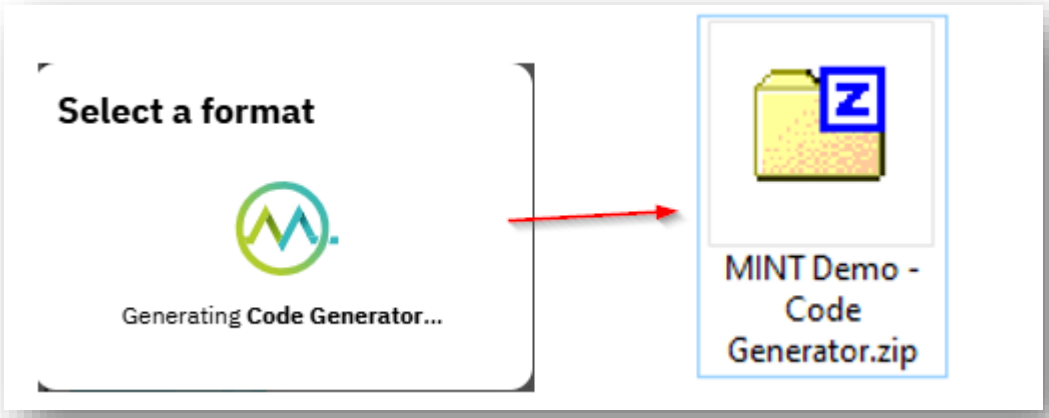


Click on the download icon.



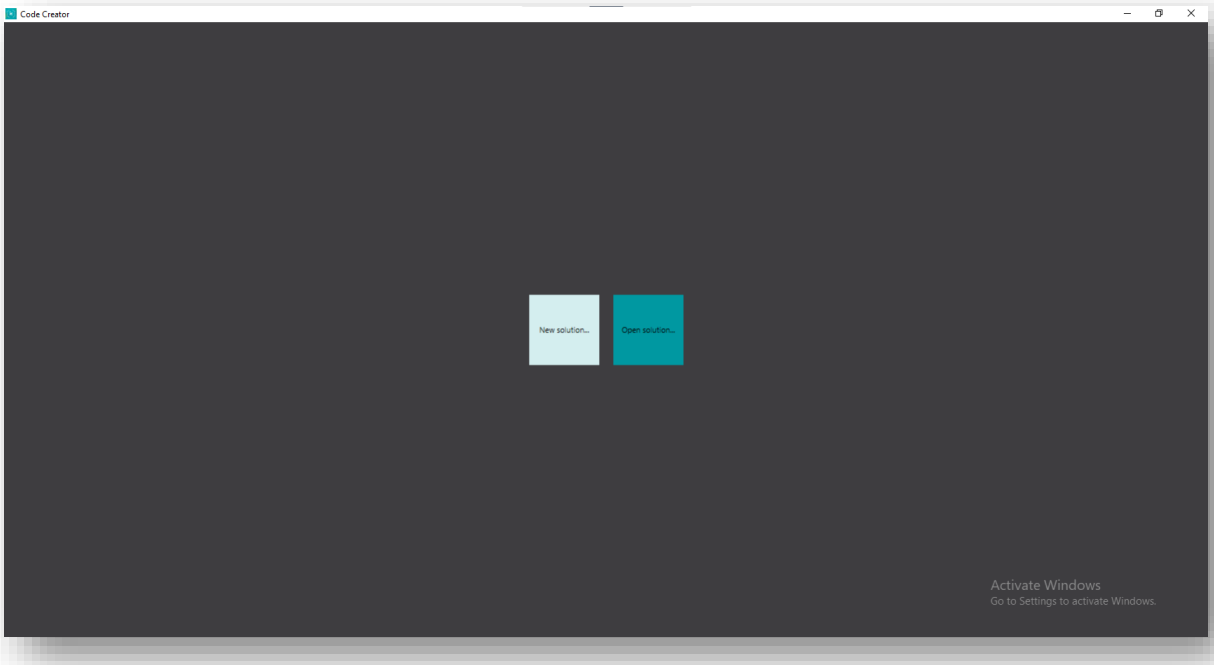
Select “Code Generator”, followed by “Download Config”



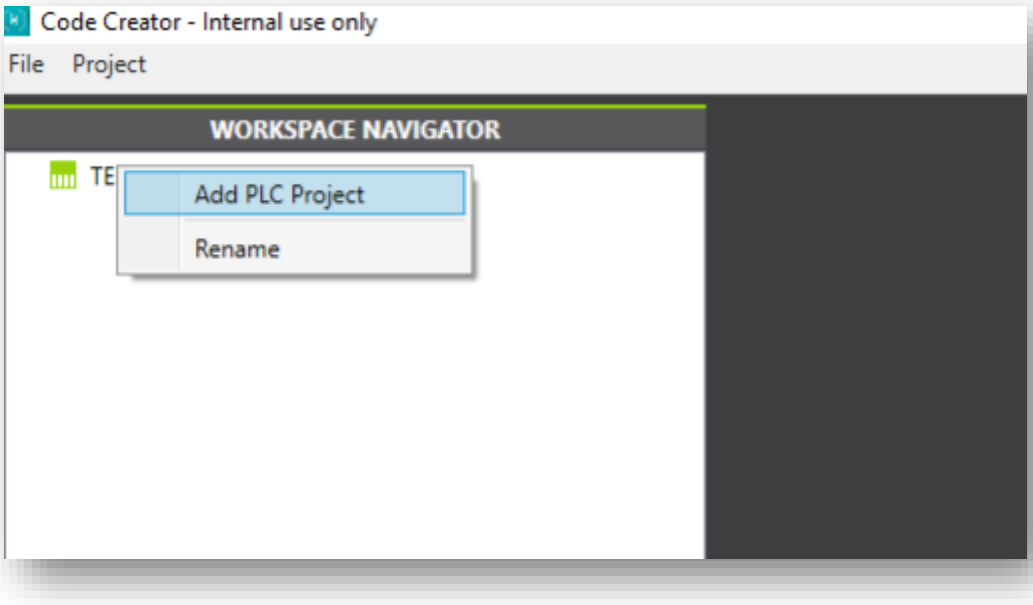


3.2 Creating a MINT project in CodeCreator

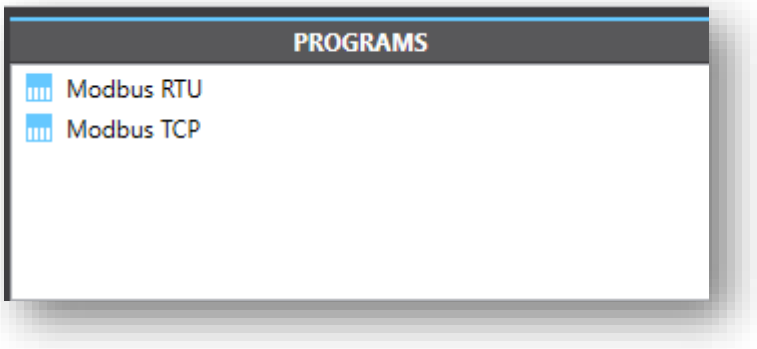
Open a new project using ‘New Solution…’



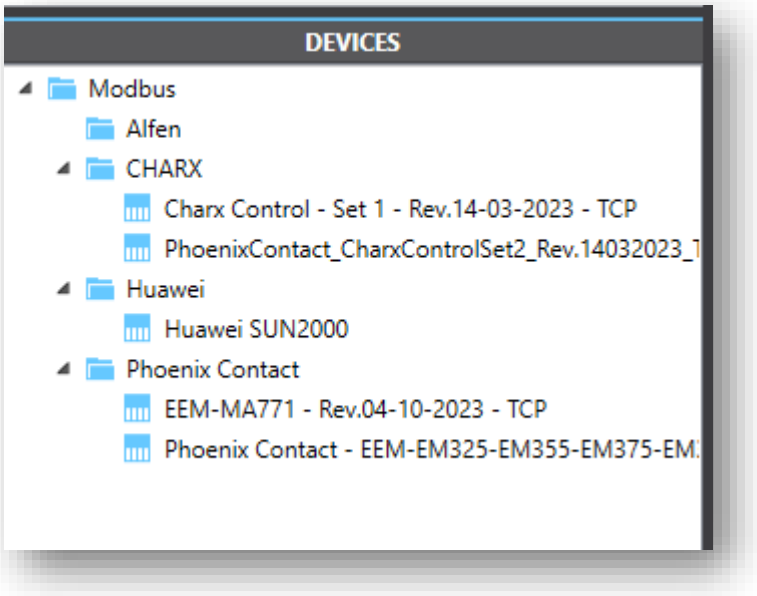
Right-click on the project name and choose ‘Add PLC Project’



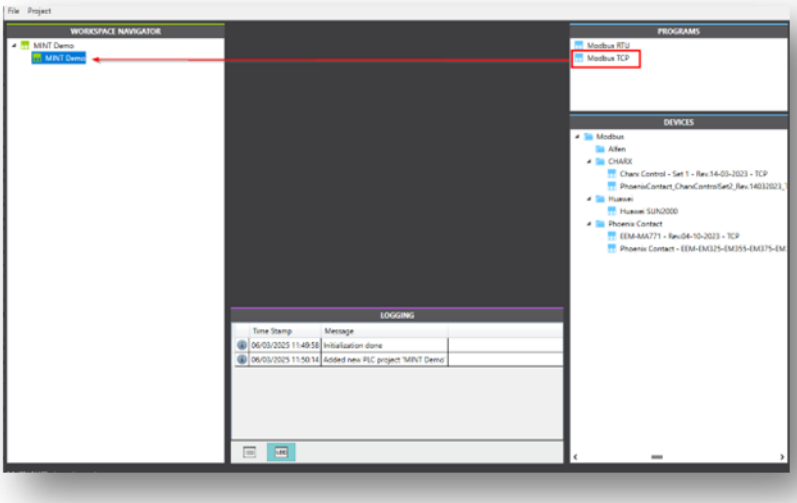
Import the programmes listed below.



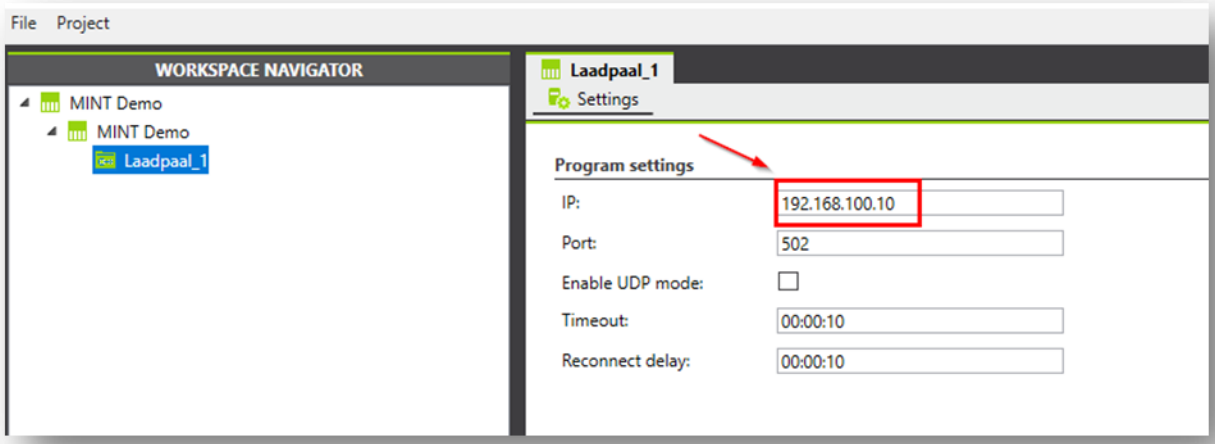
As well as the required device packages.



Add the necessary Modbus communication programme for each physical device.

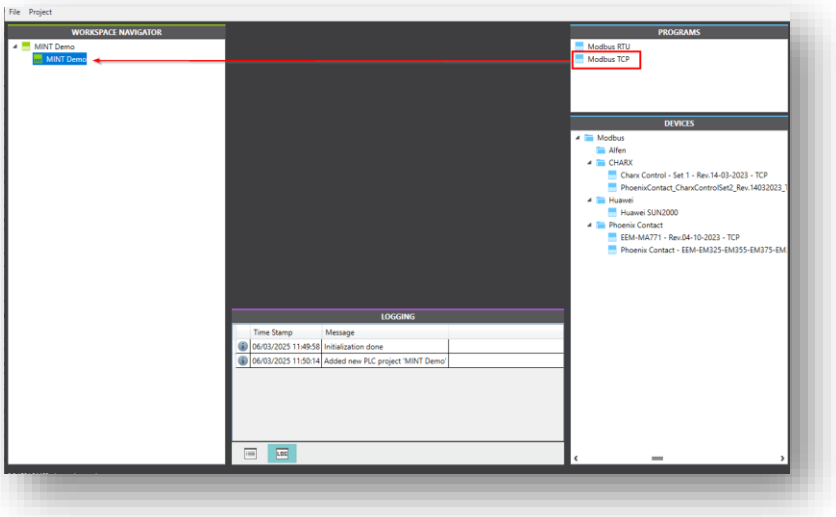


For communication via Modbus TCP, add one ‘Modbus TCP’ programme per IP address and enter the corresponding address.

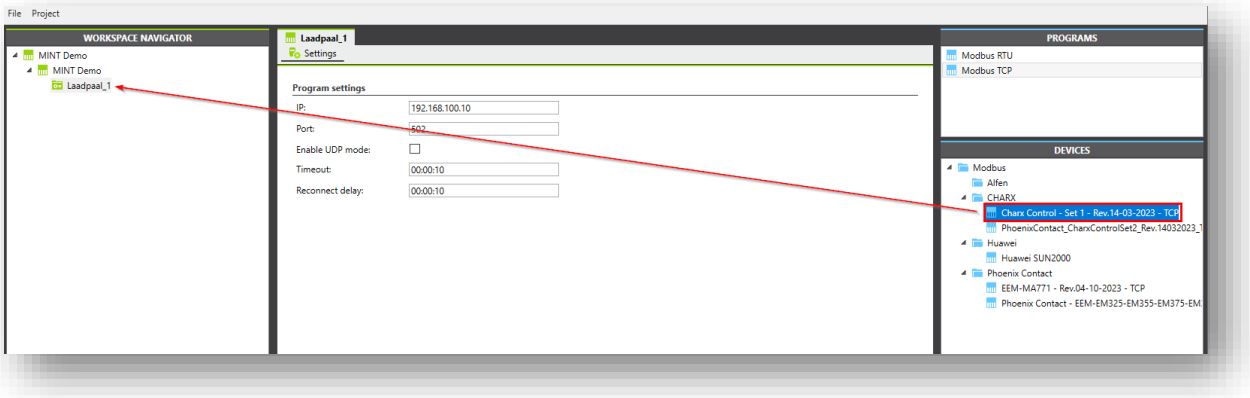


For communication via Modbus RTU, add one ‘Modbus RTU’ programme per serial connection. Choose the option that matches the hardware card connected to the PLC.

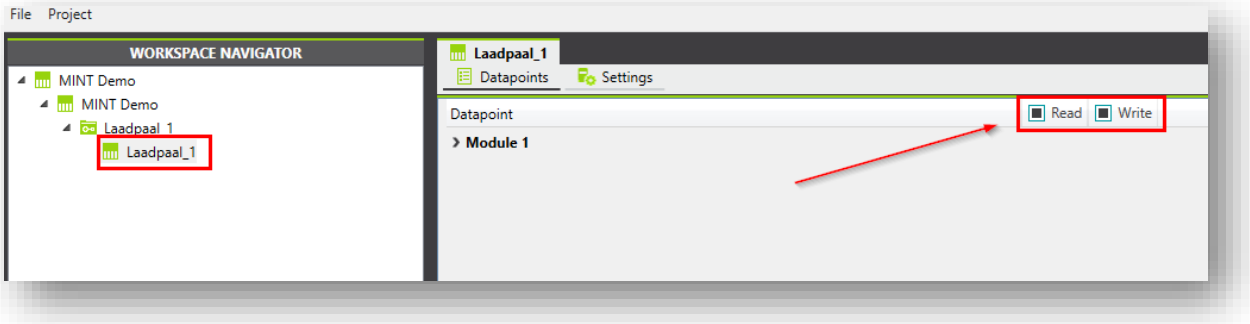
The names assigned to these blocks will be the names of the communication programmes in the PLC project. Pay close attention to naming conventions and ensure there are no spaces in the names.



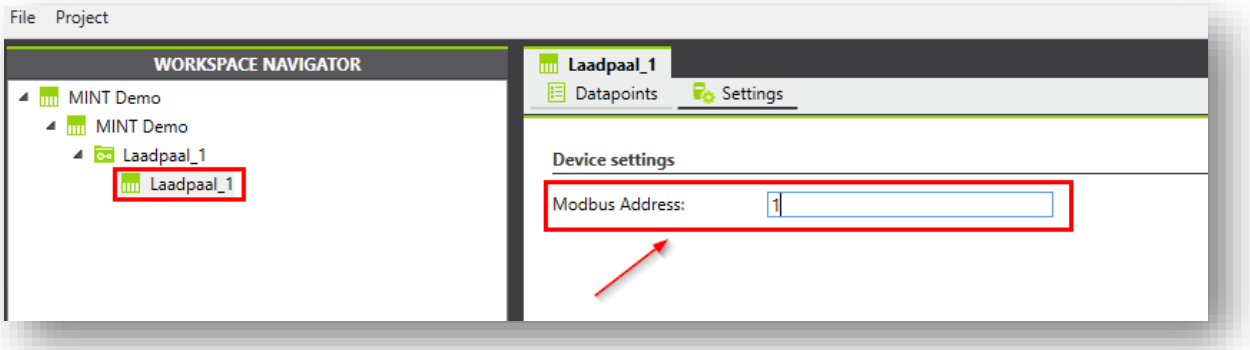
Next, add the required devices to the corresponding communication programme.



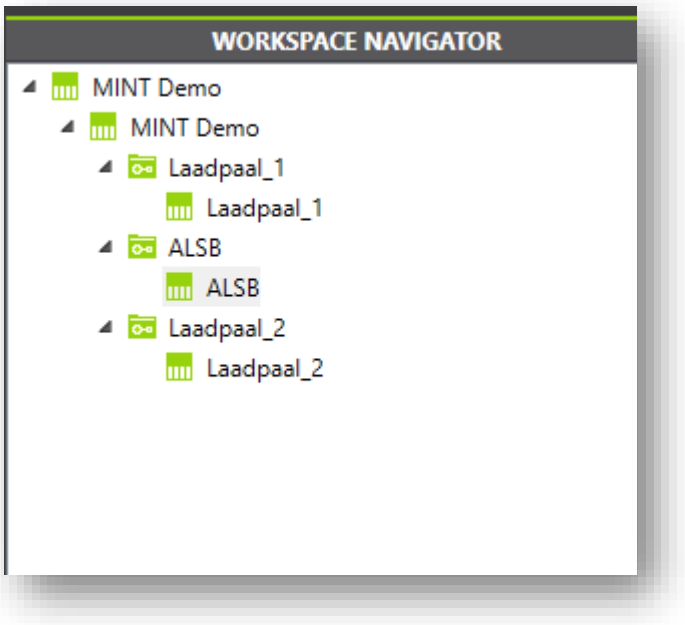
Indicate the necessary Read and Write registers.



In the Settings for each device, fill in the Modbus Address field with the corresponding unique identifier. If none exists, enter 1.

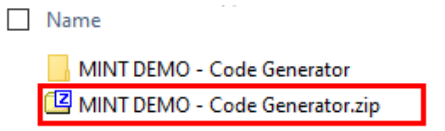


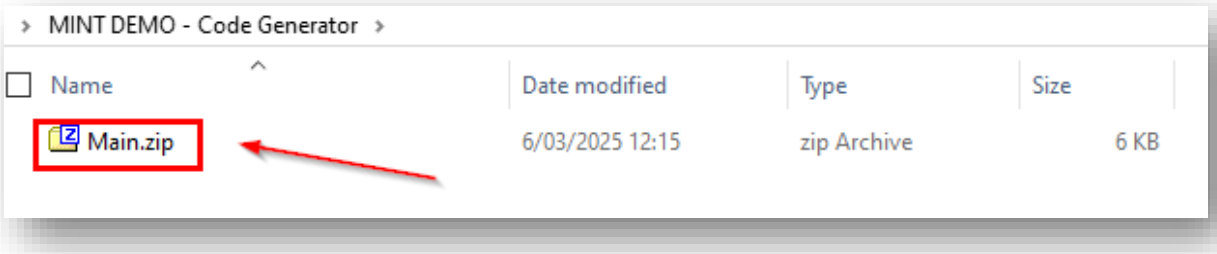
The final result should look like this:

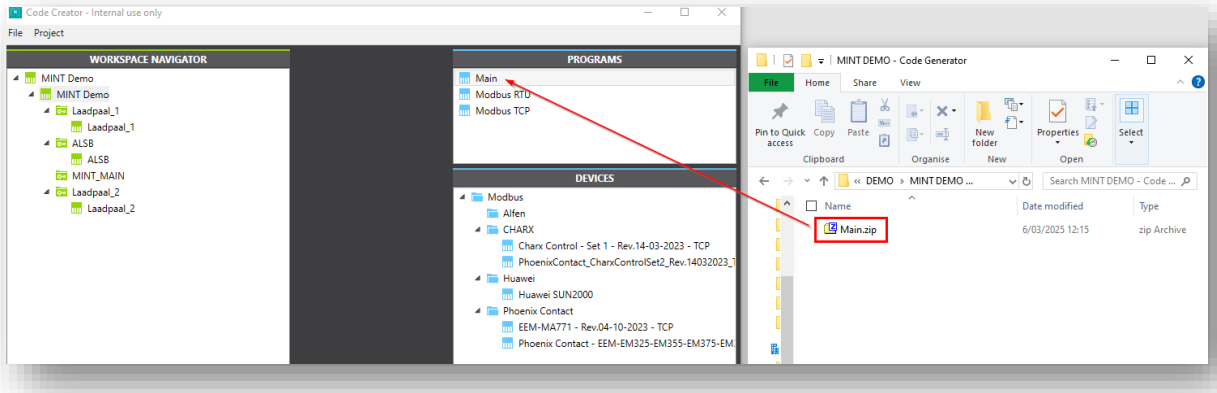


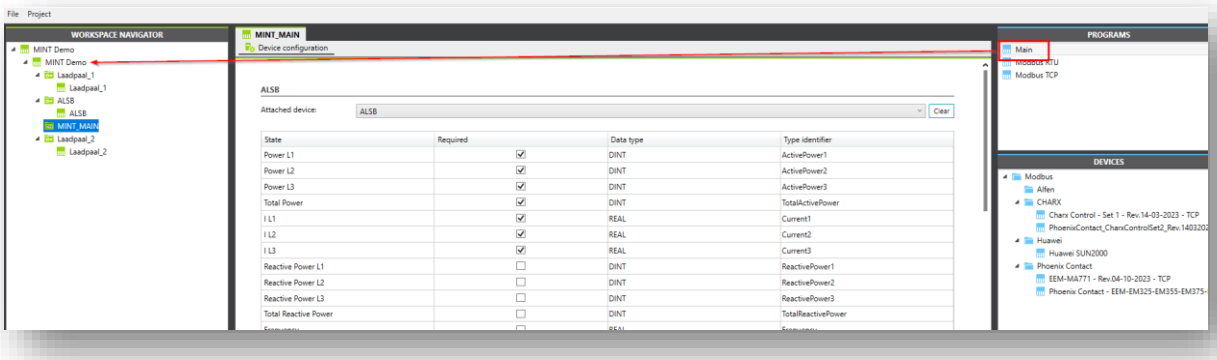
3.3 Importing the “Code Generator” (.zip)

Once all steps from section 1.2 have been completed for the current setup, you can import the downloaded “Code Generator” into CodeCreator.
In the download folder, there will be one zipped folder for each required PLC project (naming convention: Main.zip, Sub X.zip).
Unpack the download folder and use these .zip folders as input files for CodeCreator.



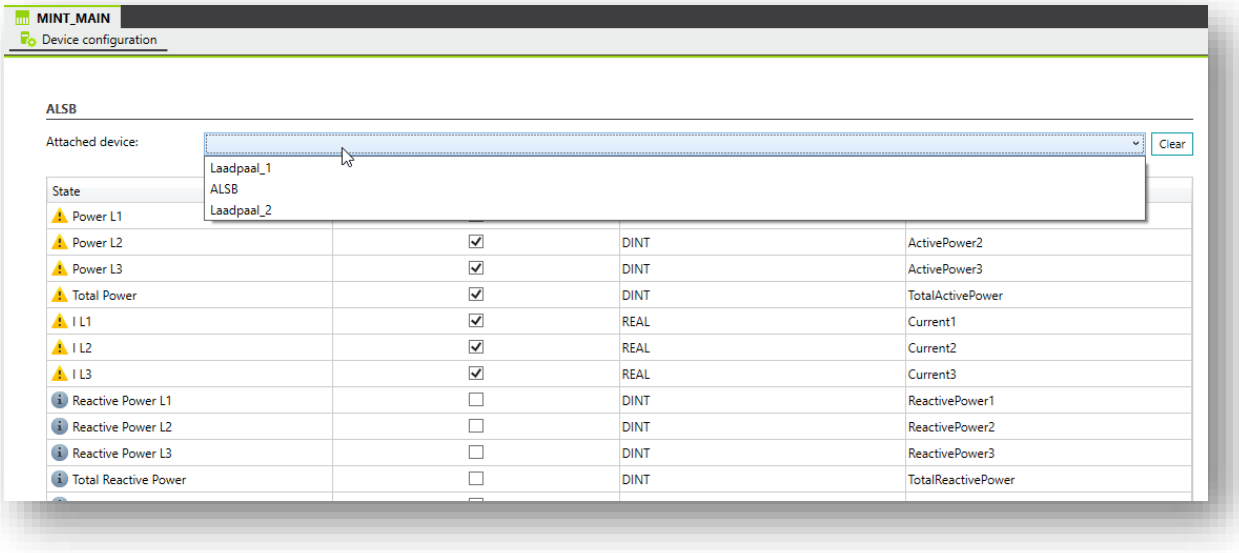


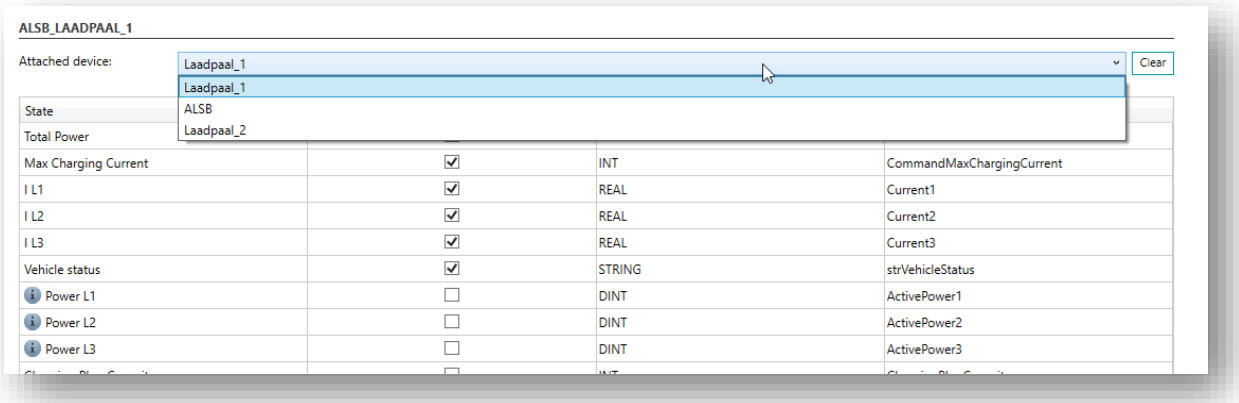


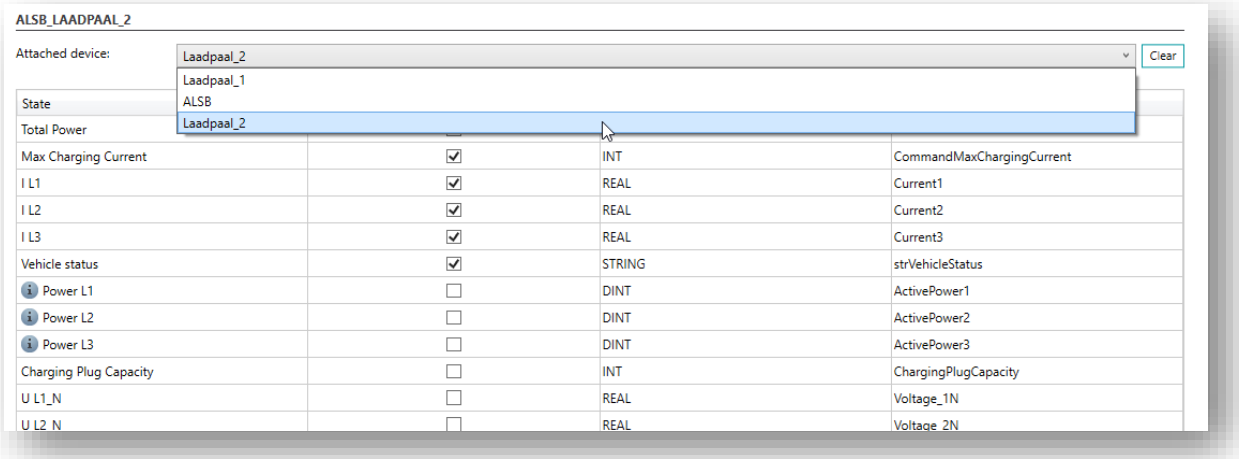


Establish a link between the Code Generator input and the devices in the Workspace Navigator.

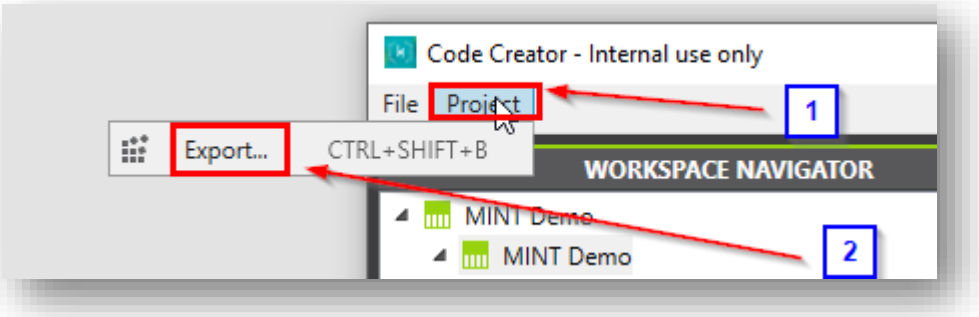
Note: It is not necessary to tick or untick any boxes in this view; they are purely informative for this application of CodeCreator.

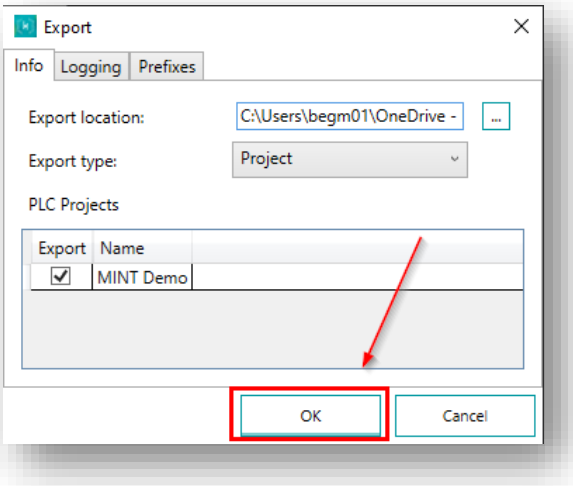




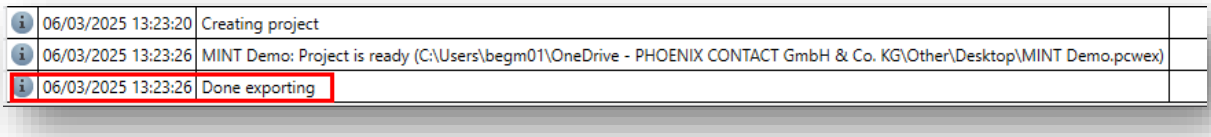


After completing all the steps above, export the code.

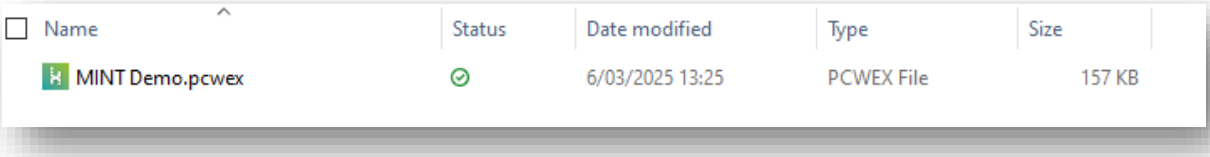




If the following “Message” appears, CodeCreator is ready.

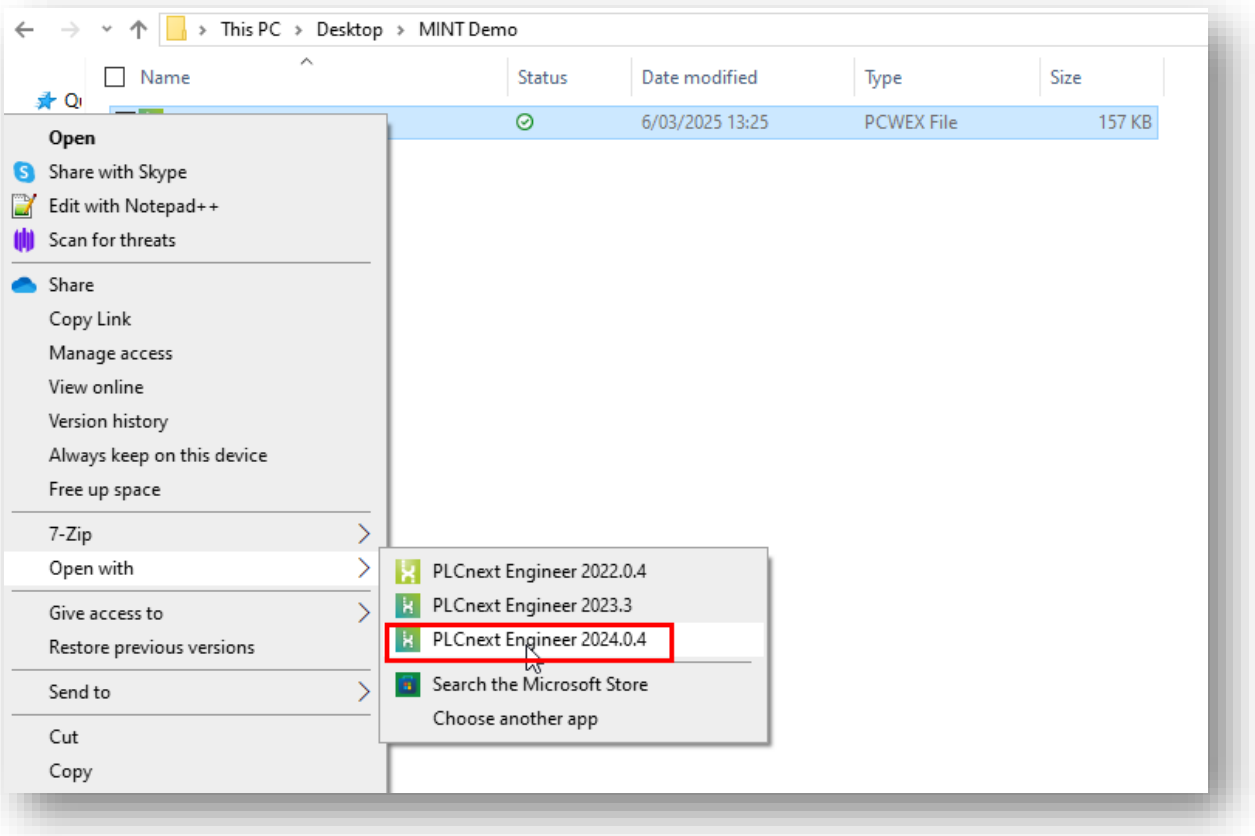


Upon creation, a .pcwex file will be generated.

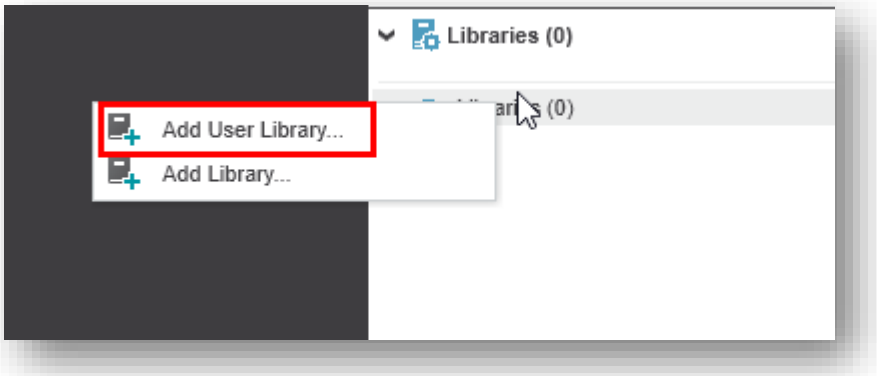


3.4 Creating a MINT Project

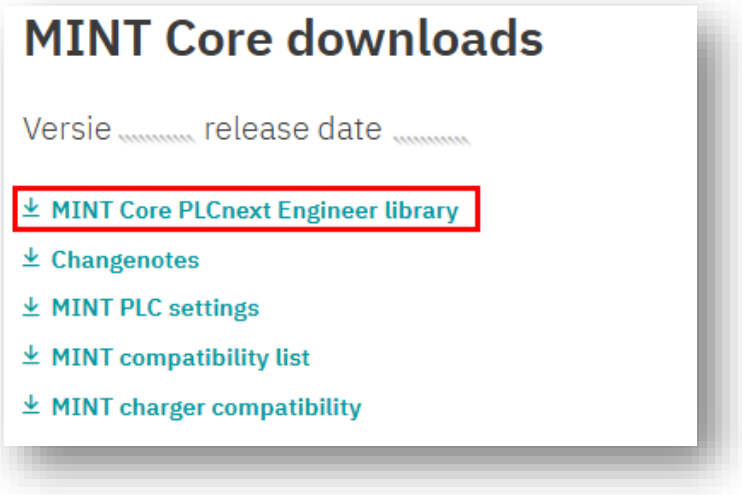
Open the generated file via CodeCreator using **PLCnext Engineer 2024.0.4 LTS**.



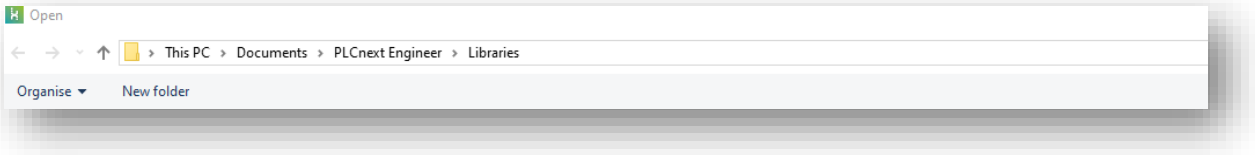
Import the correct libraries.



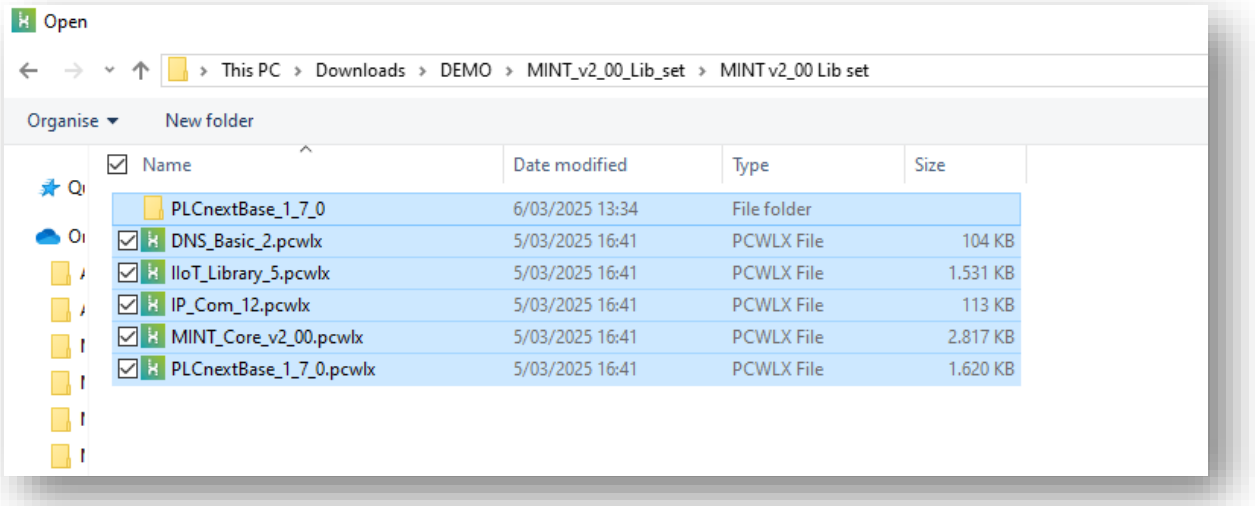
The required libraries for **MINT V2.0** can be downloaded via:
[MINT energiebeheer met AI | Phoenix Contact]



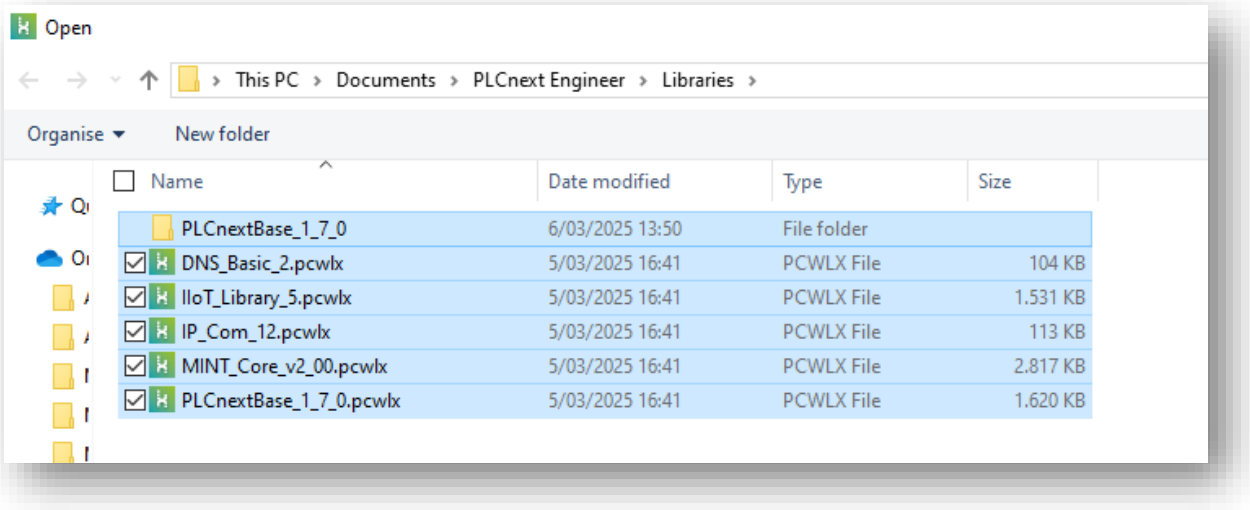
Copy the path to the libraries from PLCnext Engineer.



Copy all these files (libraries).
The example below uses version V2.0.0 — always use the latest version (!)

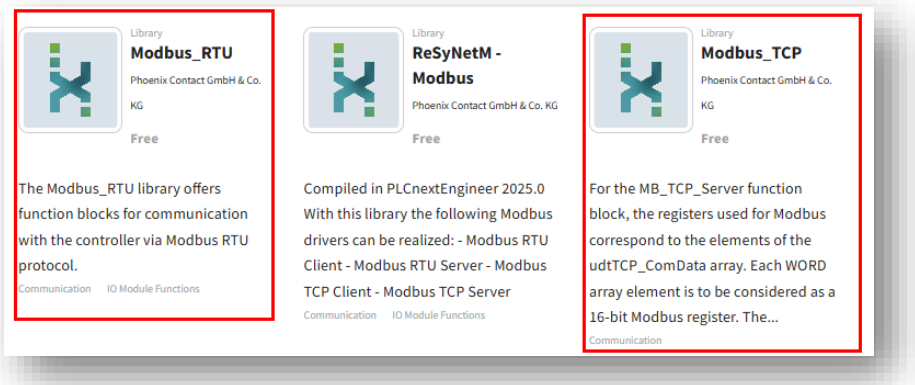


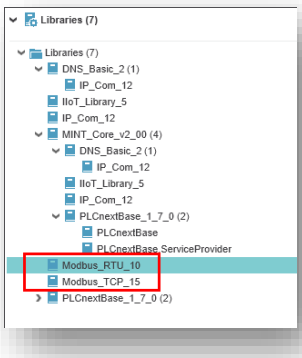
Return to the library path in PLCnext Engineer and place the libraries there.
Again, the example uses V2.0.0 — always use the latest version (!)



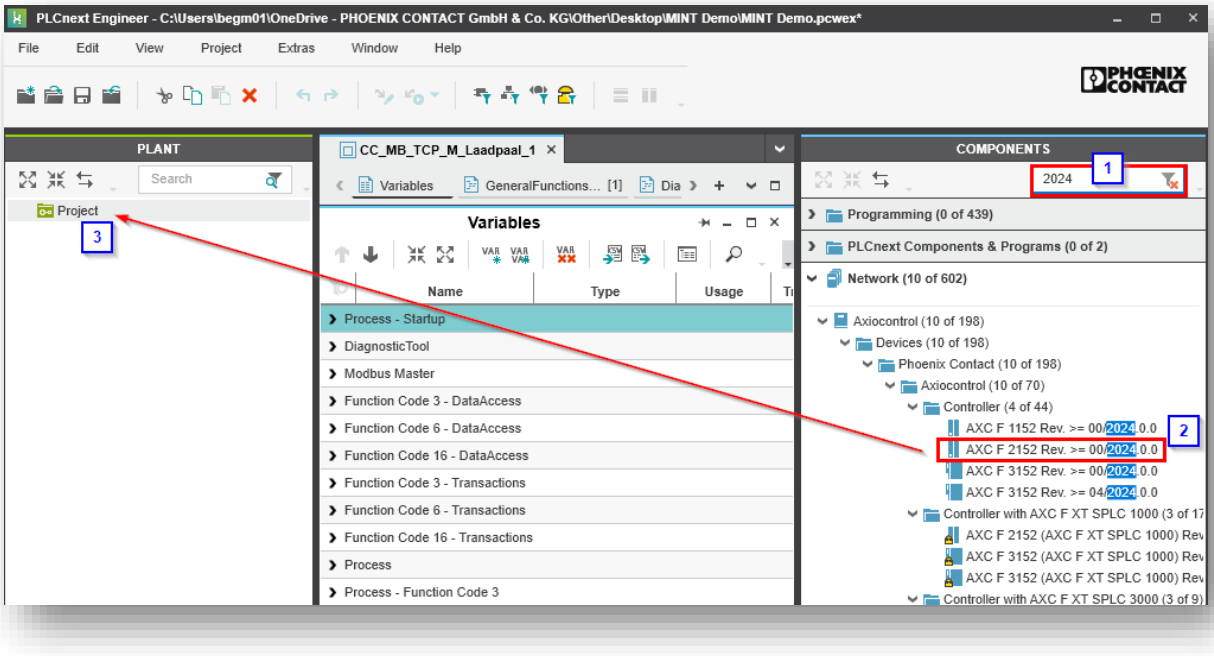
It is also necessary to download the correct communication libraries via the PLCnext Store (PLCnext Store | The open software store for automation), such as Modbus_TCP and Modbus_RTU.

- Modbus_TCP 15 has been successfully tested.
- Modbus_RTU 10 has been successfully tested.



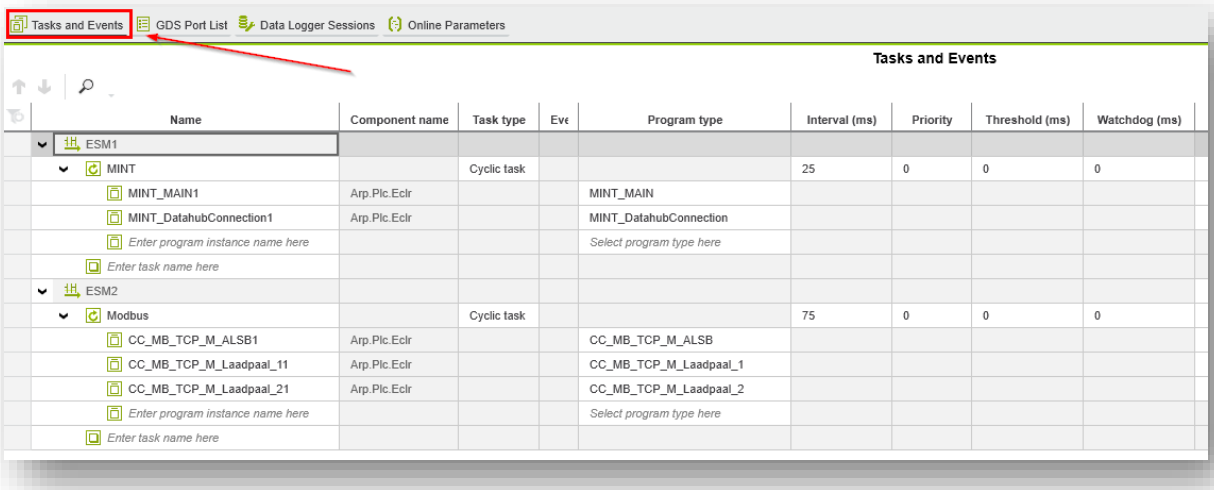


Adding a PLC to the project;
Use the search function to look for “2024”, then drag AXC F 2152 Rev. ≥ 00/2024.0.0 into the project.



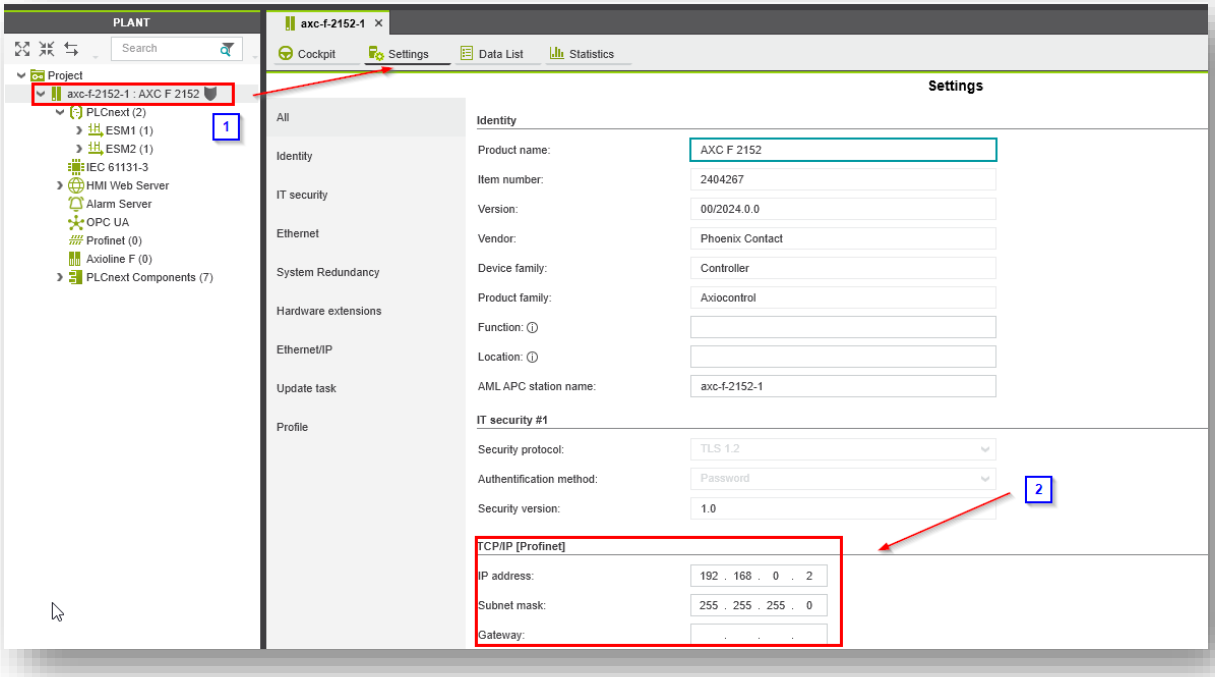
Creating Tasks and Events
For the MINT section, a task must be created with a cycle time of 25 ms.
Add the generated MINT programme and the MINT_DatahubConnection programme to this task to ensure they are properly aligned. Disable the watchdog by setting it to 0.

Add the communication programmes to the second core, choose a suitable cycle time, and disable the watchdog.

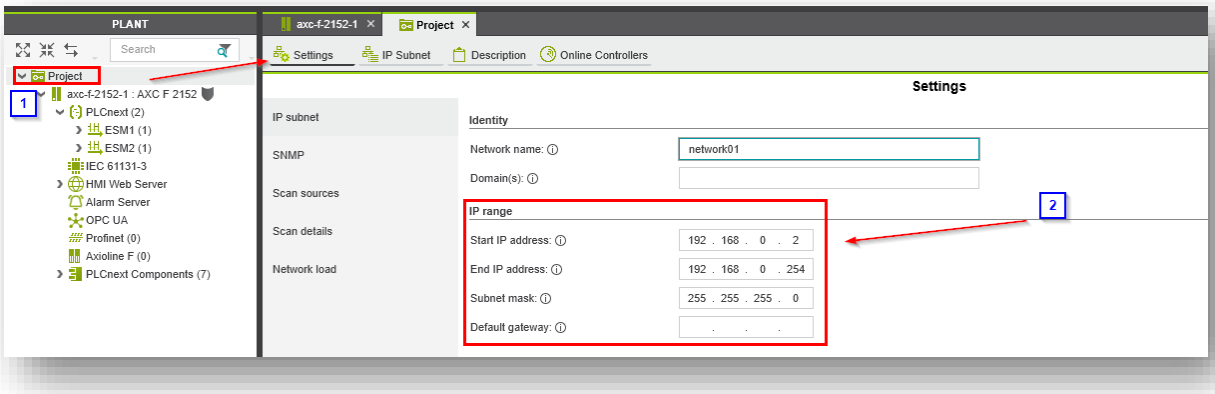


Name	Component name	Task type	Eve	Program type	Interval (ms)	Priority	Threshold (ms)	Watchdog (ms)
ESM1								
MINT		Cyclic task			25	0	0	0
MINT_MAIN1	Arp.Plc.Eclr			MINT_MAIN				
MINT_DatahubConnection1	Arp.Plc.Eclr			MINT_DatahubConnection				
Enter program instance name here				*Select program type here*				
Enter task name here								
ESM2								
Modbus		Cyclic task			75	0	0	0
CC_MB_TCP_M_ALSB1	Arp.Plc.Eclr			CC_MB_TCP_M_ALSB				
CC_MB_TCP_M_Laadpaal_11	Arp.Plc.Eclr			CC_MB_TCP_M_Laadpaal_1				
CC_MB_TCP_M_Laadpaal_21	Arp.Plc.Eclr			CC_MB_TCP_M_Laadpaal_2				
Enter program instance name here				*Select program type here*				
Enter task name here								

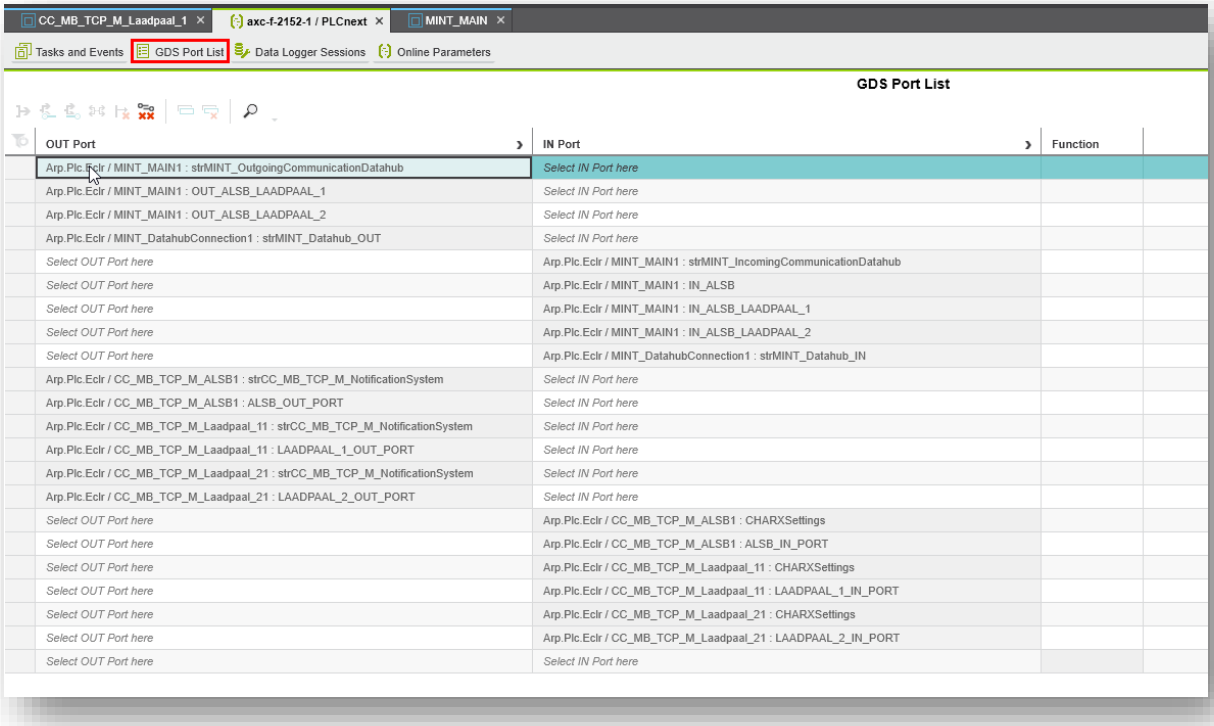
Adjusting the PLC IP address

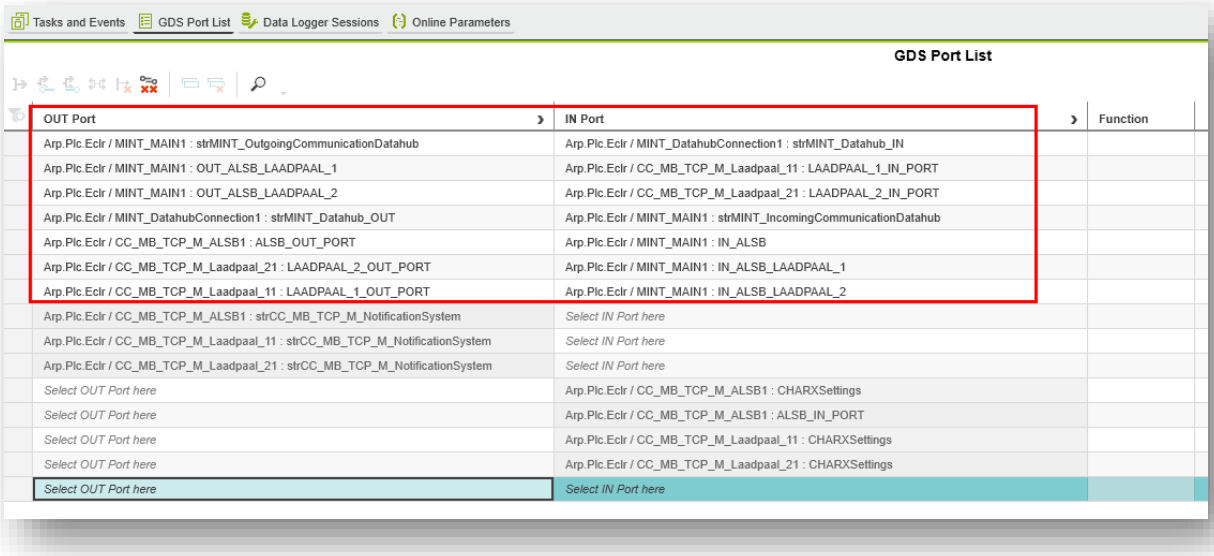


The IP range must also be updated in the project to reflect the actual situation.

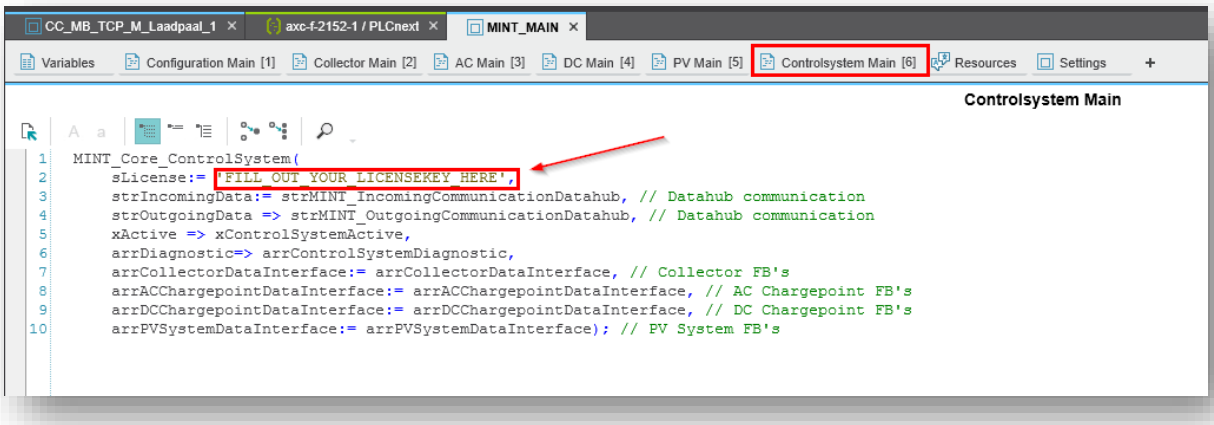


In the GDS Port List, link the IN PORTS and OUT PORTS of the MINT programme to their counterparts in the communication programmes

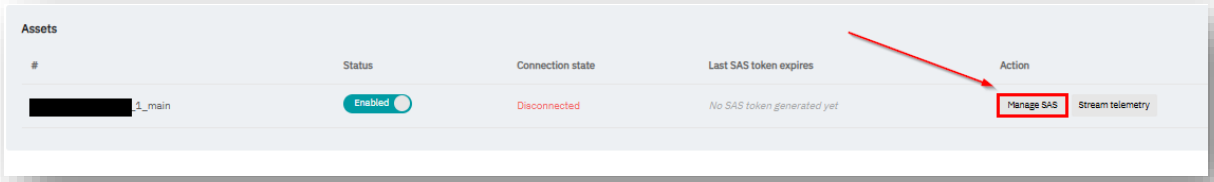


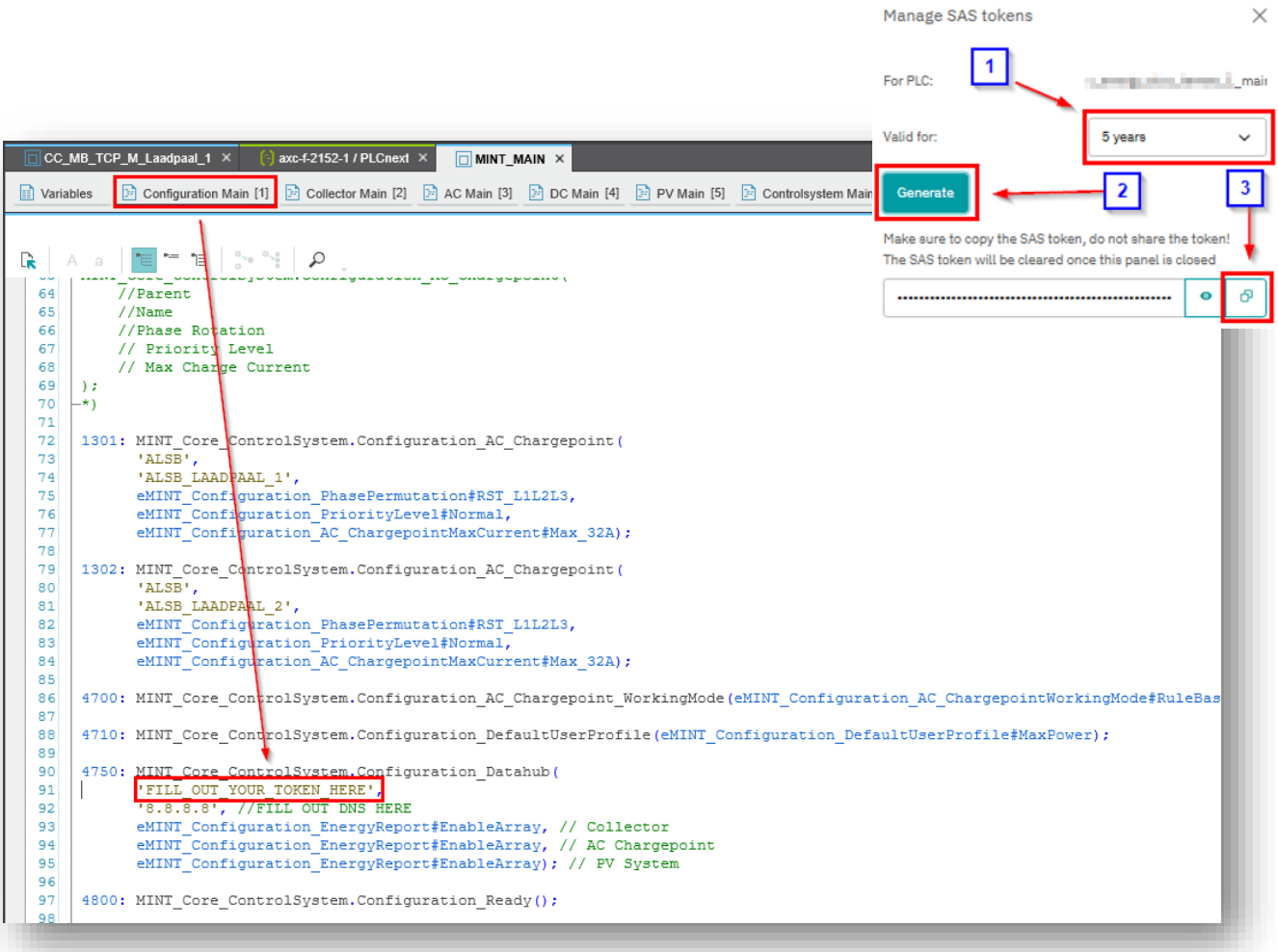


Adding the MINT licence, this can be obtained via Phoenix Contact.

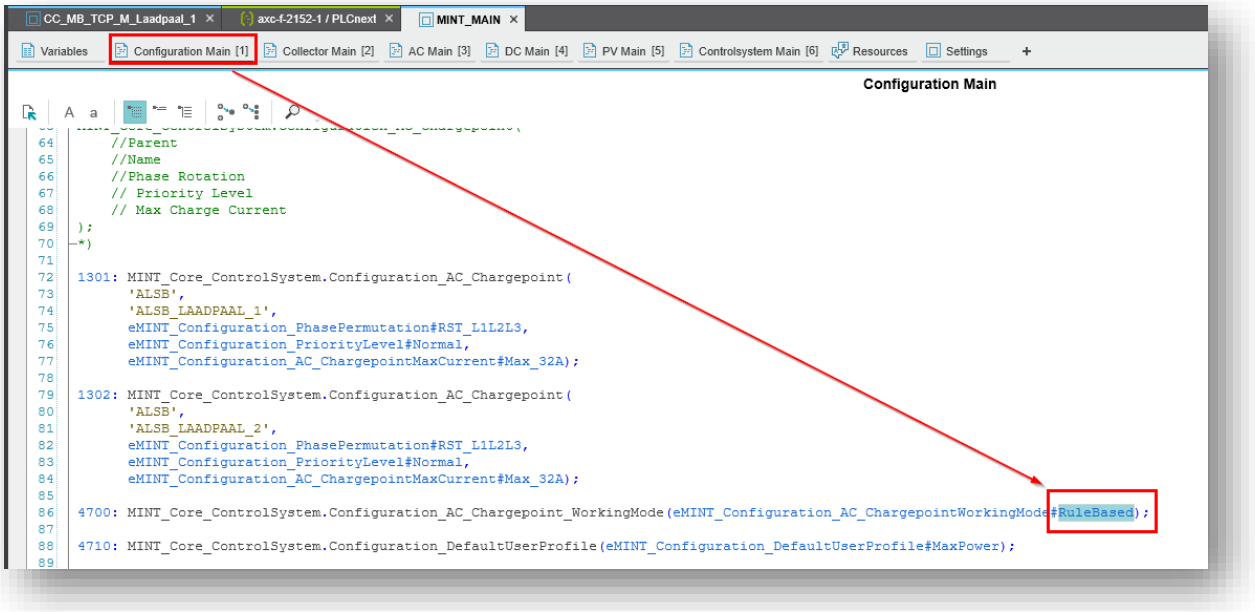


Adding the MQTT SAS Token, this can be obtained via Phoenix Contact or via the MINT Portal. It is required to set up outbound communication from the PLC.

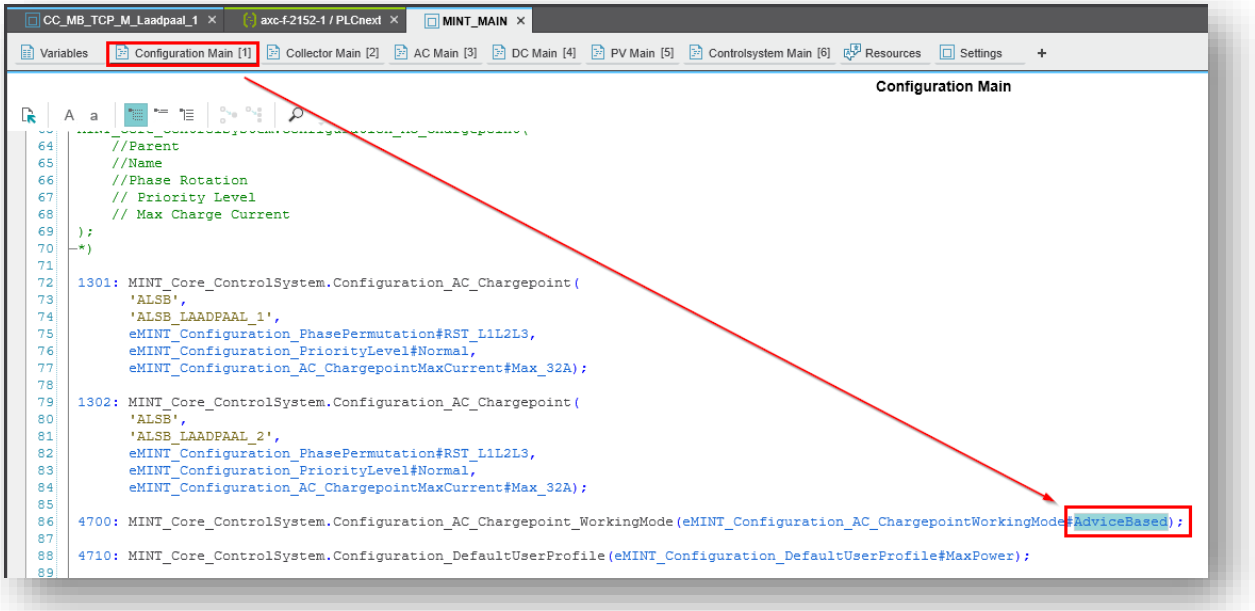




For a MINT Core, the working modes of the asset types must be set to #RuleBased.
(See example below for AC charging stations)



For a MINT Advanced with optimised charging schedules, the working modes of the asset types must be set to #AdviceBased.
(See example below for AC charging stations)

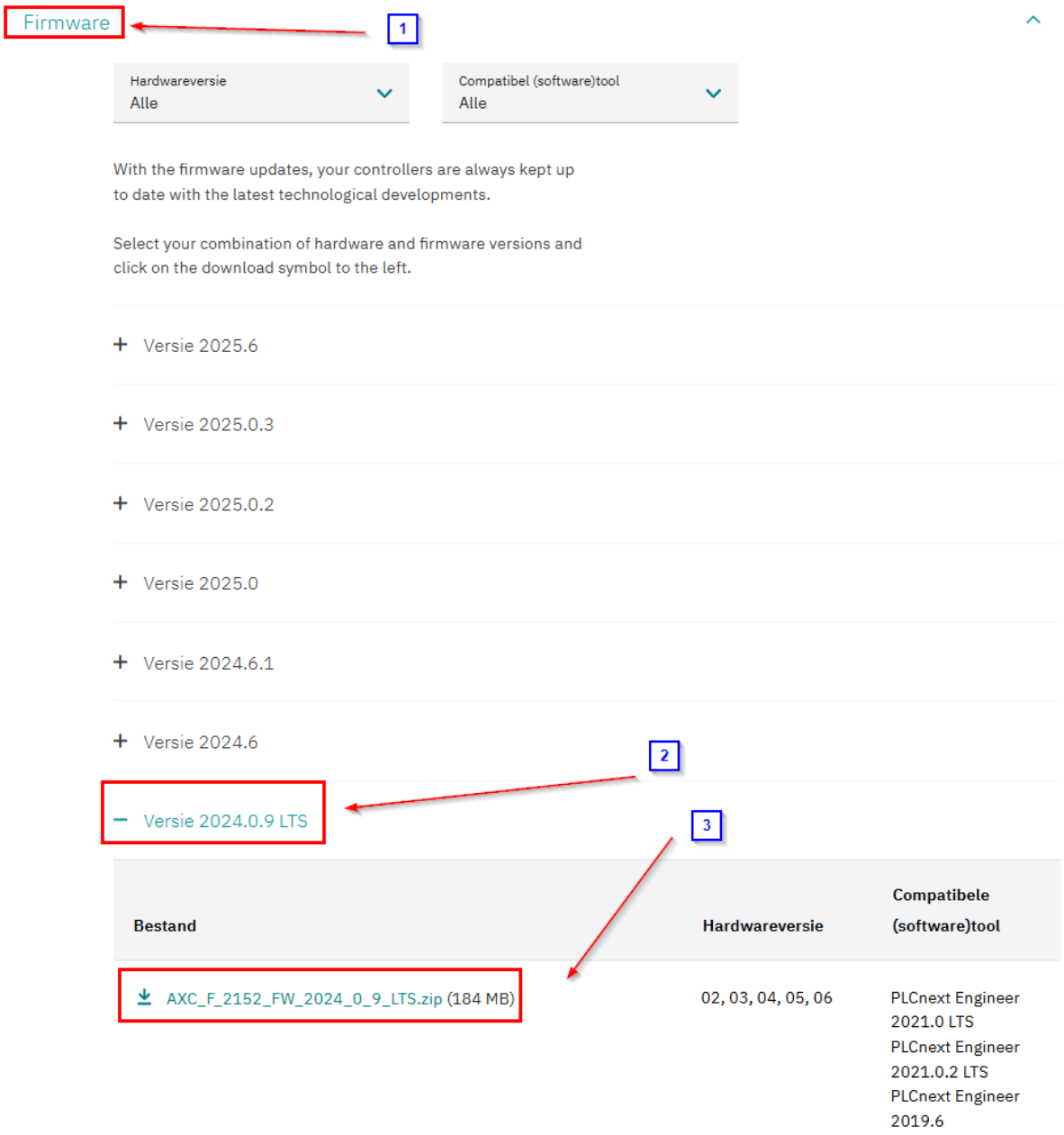


4 Preparing the PLC

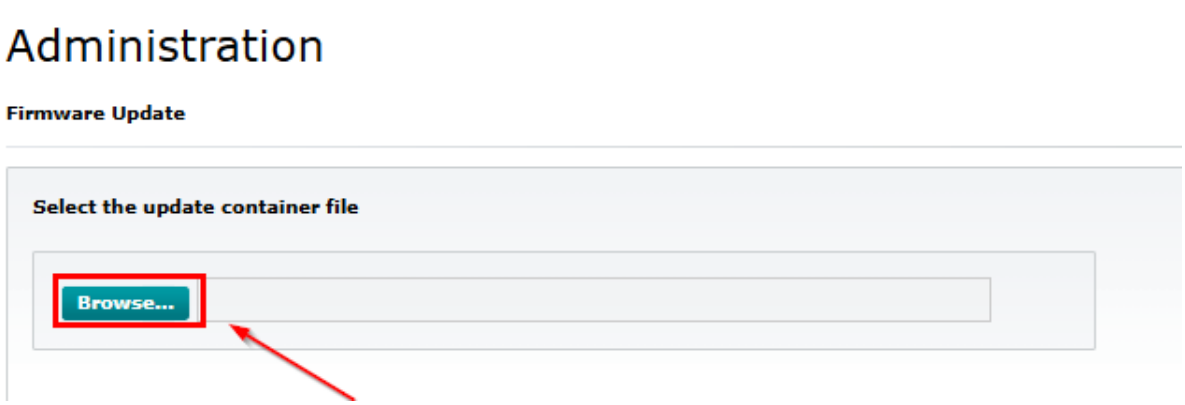
For the MINT V2.0.1 update, the PLC must also undergo several updates and adjustments. These are explained below.

4.1 Firmware update

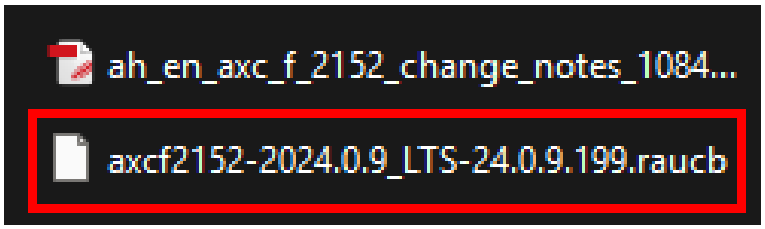
The PLC (or PLCs) must be updated to version 2024.0.9 LTS.
This version can be found on the Phoenix Contact website: (AXC F 2152 - Besturing - 2404267 | Phoenix Contact)



Navigate to the WBM (Web-Based Management) interface of the PLC and go to “Firmware Update”. **(!) This update must be performed twice (!)**

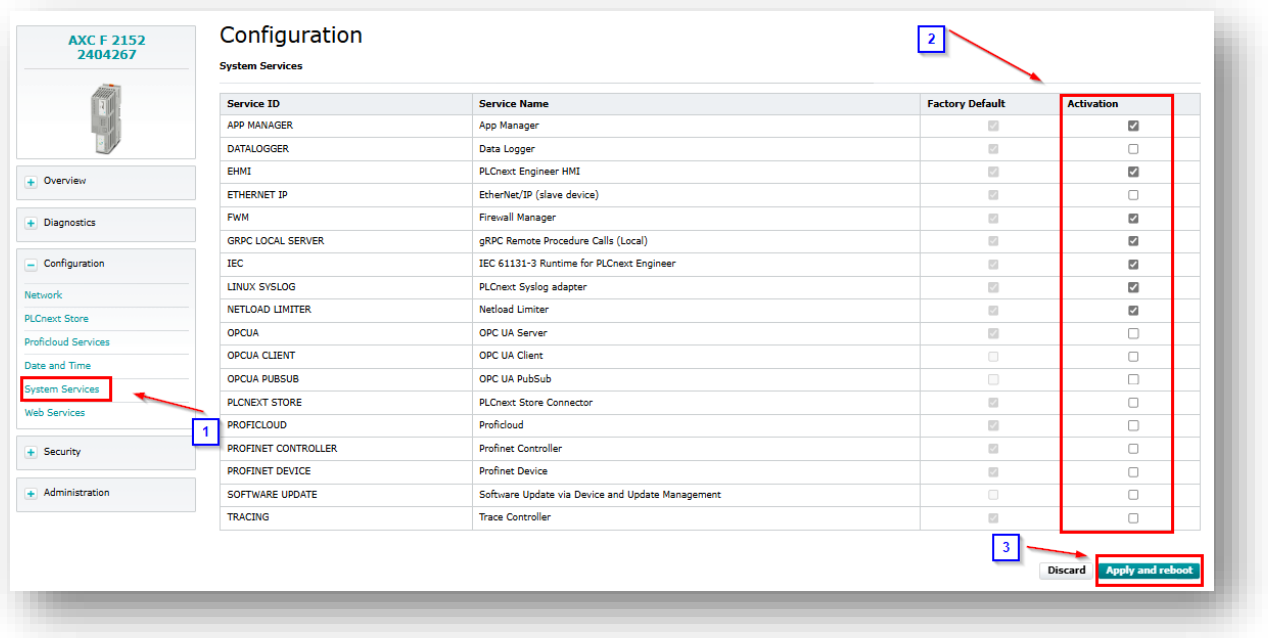


Select the .raucb file.

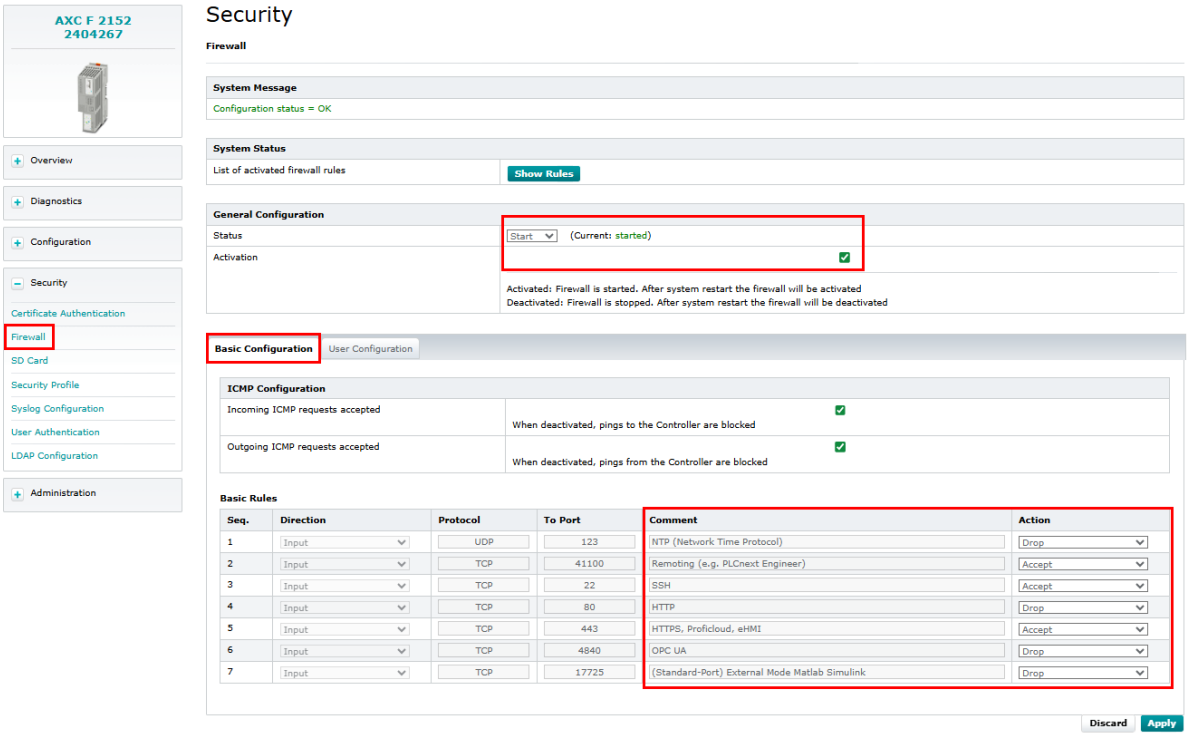


4.2 System services

Navigate to “System Services” in the PLC’s WBM.
Ensure all settings match the configuration shown below.



4.3 Firewall settings

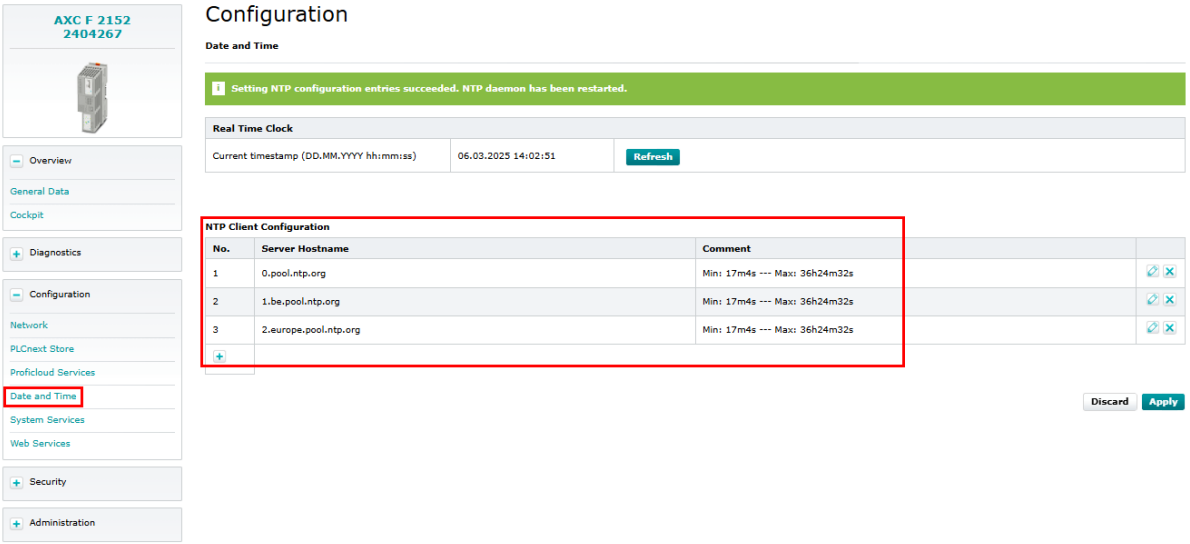


4.4 NTP synchronisation

Add the following NTP servers:

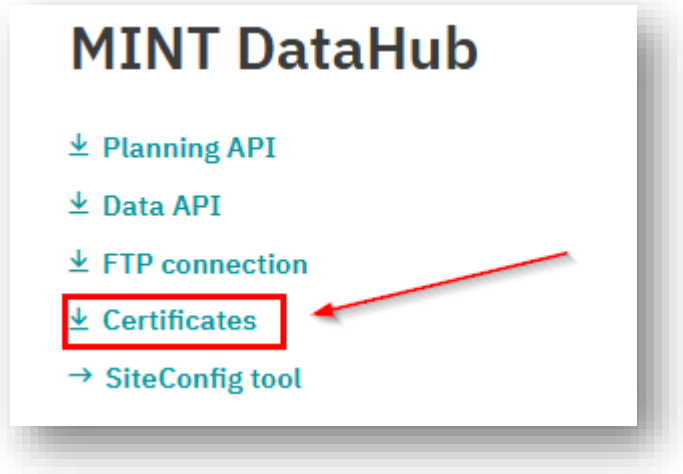
- 0.pool.ntp.org
- 1.be.pool.ntp.org
- 2.europe.pool.ntp.org

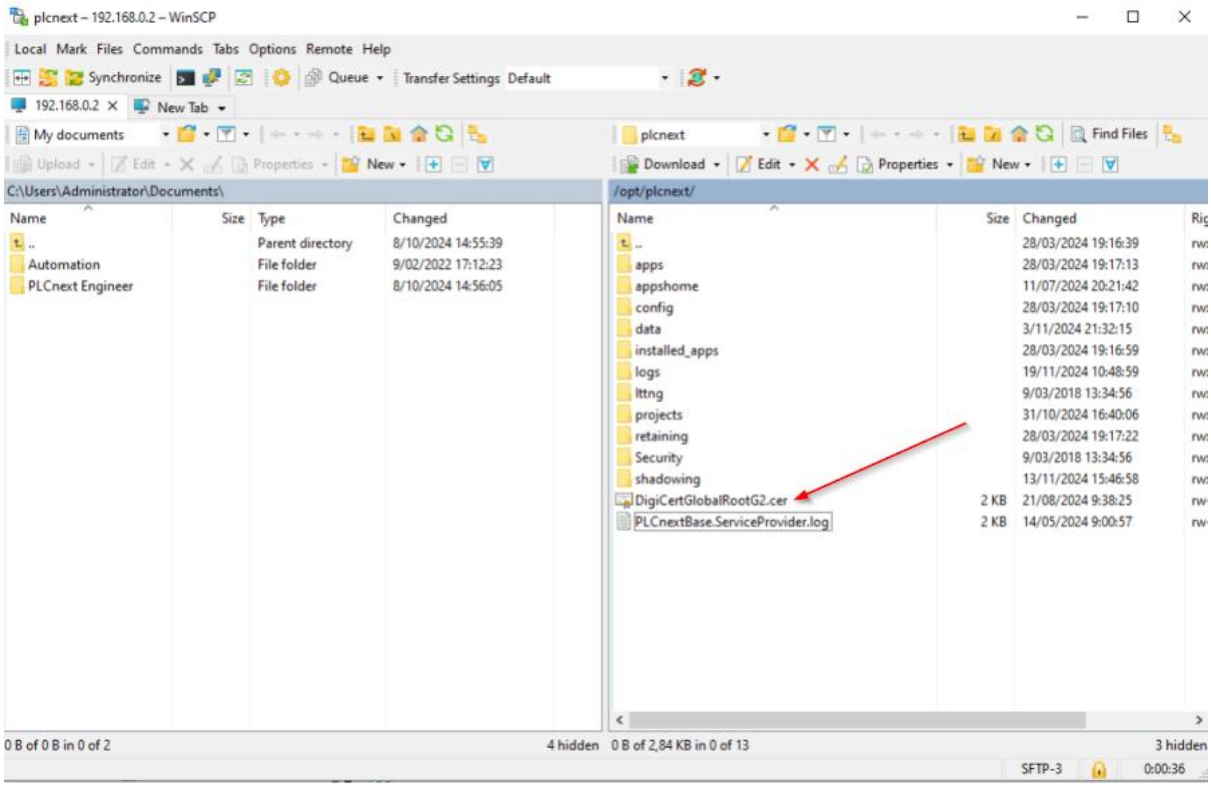
min. polling van: 17m4s
max. polling van: 36h24m32s



4.5 Adding MQTT Certificate to the PLC

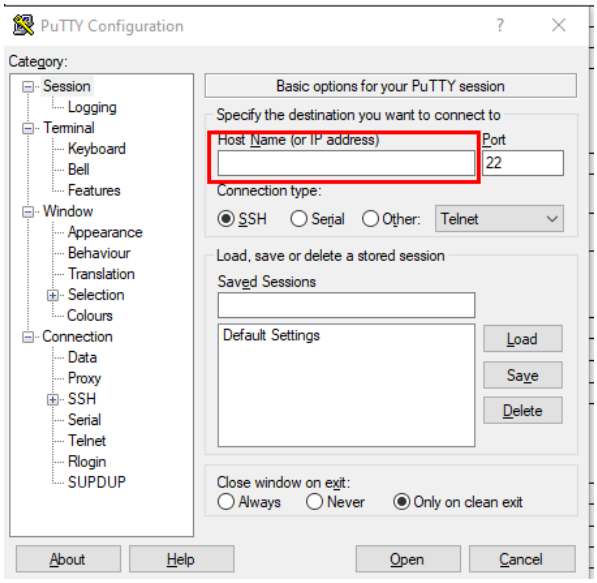
Use the tool WinSCP to log in using the PLC's username and password.
The certificate **"DigiCertGlobalRootG2.cer"** must be uploaded to the PLC.
This certificate can be downloaded via: MINT energiebeheer met AI | Phoenix Contact.



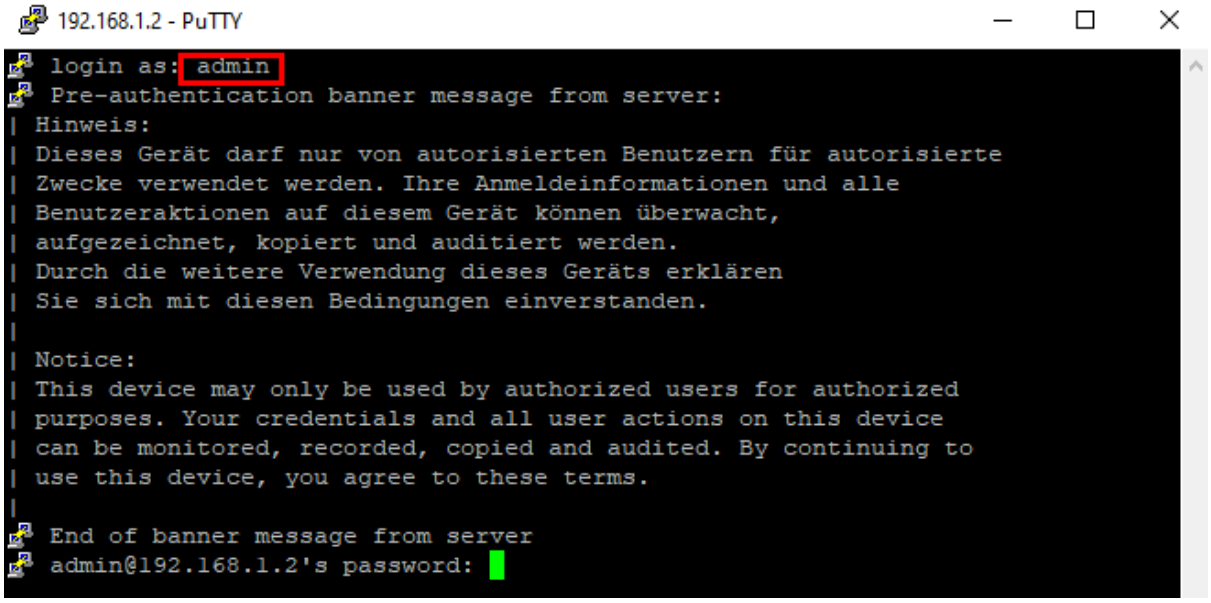

plcnext – 192.168.0.2 – WinSCP
Local Mark Files Commands Tabs Options Remote Help
Synchronize
Queue
Transfer Settings Default
192.168.0.2
New Tab
My documents
Upload
Edit
Properties
New
C:\Users\Administrator\Documents\
Name Size Type Changed
.. Parent directory 8/10/2024 14:55:39
Automation File folder 9/02/2022 17:12:23
PLCnext Engineer File folder 8/10/2024 14:56:05
0 B of 0 B in 0 of 2
4 hidden
plcnext
Download
Edit
Properties
New
Find Files
/opt/plcnext/
Name Size Changed
.. 28/03/2024 19:16:39
apps 28/03/2024 19:17:13
appshome 11/07/2024 20:21:42
config 28/03/2024 19:17:10
data 3/11/2024 21:32:15
installed_apps 28/03/2024 19:16:59
logs 19/11/2024 10:48:59
lttng 9/03/2018 13:34:56
projects 31/10/2024 16:40:06
retaining 28/03/2024 19:17:22
Security 9/03/2018 13:34:56
shadowing 13/11/2024 15:46:58
DigiCertGlobalRootG2.cer 2 KB 21/08/2024 9:38:25
PLCnextBase.ServiceProvider.log 2 KB 14/05/2024 9:00:57
0 B of 2,84 KB in 0 of 13
3 hidden
SFTP-3
0:00:36

4.6 Checking the PLC via Linux

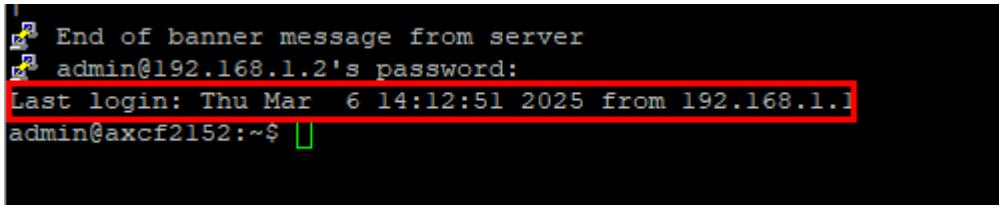
Use the tool ‘PuTTY’.
Enter the IP address of the PLC.



Log in as: ‘admin’



Enter the PLC password.



4.6.1 Check the firmware

Enter the command: “rauc status --detailed”

```
admin@axcf2152:~$ rauc status --detailed
=== System Info ===
Compatible:  axcf2152_v1
Variant:
Booted from: rootfs.1 (B)
```

```
=== Slot States ===
x [rootfs.1] (/dev/mmcblk0p3, ext4, booted)
        bootname: B
        mounted: /media/rfs/ro
        boot status: good
      slot status:
          bundle:
              compatible=axcf2152_v1
              version=24.0.8.183
              description=Update container for axcf2152
              build=20241113101132
              hash=e59ba2934155c641f18dcf7d3eae4159b968c5db46f71f2e0d2c996db5973193
```

```
o [rootfs.0] (/dev/mmcblk0p2, ext4, inactive)
        bootname: A
        boot status: good
      slot status:
          bundle:
              compatible=axcf2152_v1
              version=24.0.8.183
              description=Update container for axcf2152
              build=20241113101132
```

4.6.2 Checking the NTP server

Enter the command: “date”
This should return the time in UTC. The time must be accurate, otherwise outbound communications will fail. It is possible that the clock is still synchronising, but do not leave the system unattended if this has not yet occurred.

```
admin@axcf2152:~$ date
Thu Mar  6 14:23:58 UTC 2025
```

Enter the command “ntpq -p”
This table must be filled. It is important that the system has NTP synchronisation capabilities. Over time, the clock will begin to drift, and outbound communication will eventually fail.

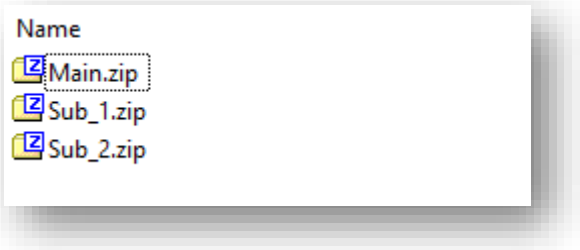
```
admin@axcf2152:~$ ntpq -p
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
*LOCAL(0)        .LOCL.          14 l   21   64  377    0.000   +0.000   0.002
 time.cloudflare 10.78.8.4        3 u  201 1024    3   31.809   -2.254   0.281
 212.129-177-91. 45.87.78.35      3 u  205 1024    3   33.882   -0.517   1.752
 91.109.118.94 ( 185.57.191.229   2 u  154 1024    3   38.936   -1.173   1.849
admin@axcf2152:~$
```

5 MULTI PLC

When more than one PLC is present in the project, it is considered a **MULTI-PLC project**. This consists of one **main** PLC and one or more **sub** PLCs.

5.1 Download site configuratie

The MINT Portal (| Phoenix Contact Mint) also provides downloads for MULTI PLC projects. This download includes a separate configuration file for each PLC.
See example below:

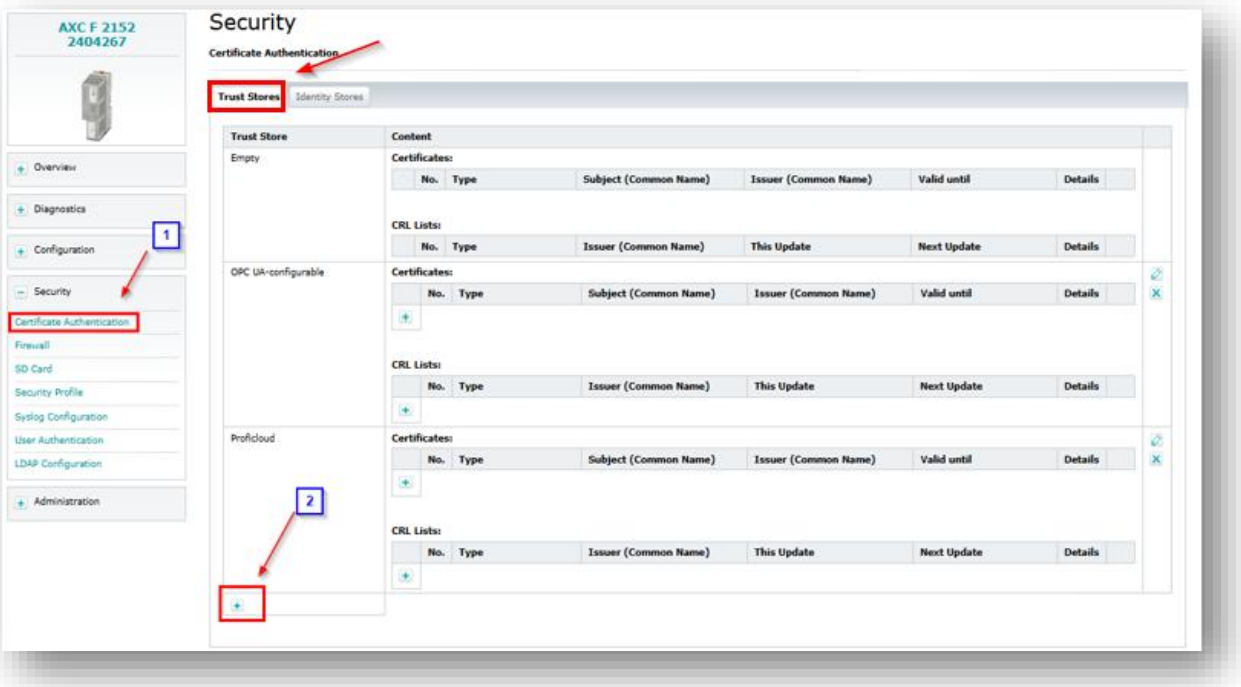


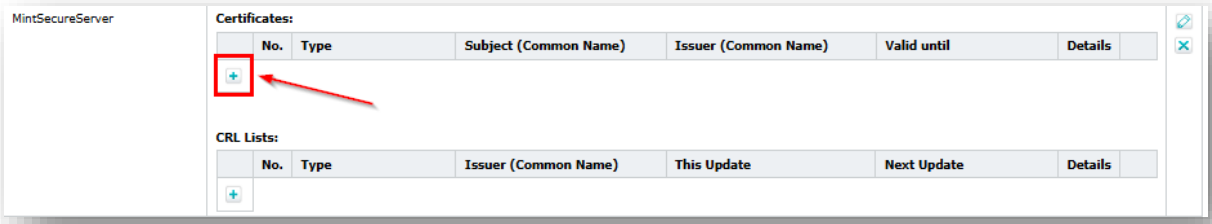
Each PLC must go through the steps outlined in sections **1.2** and **1.3**.

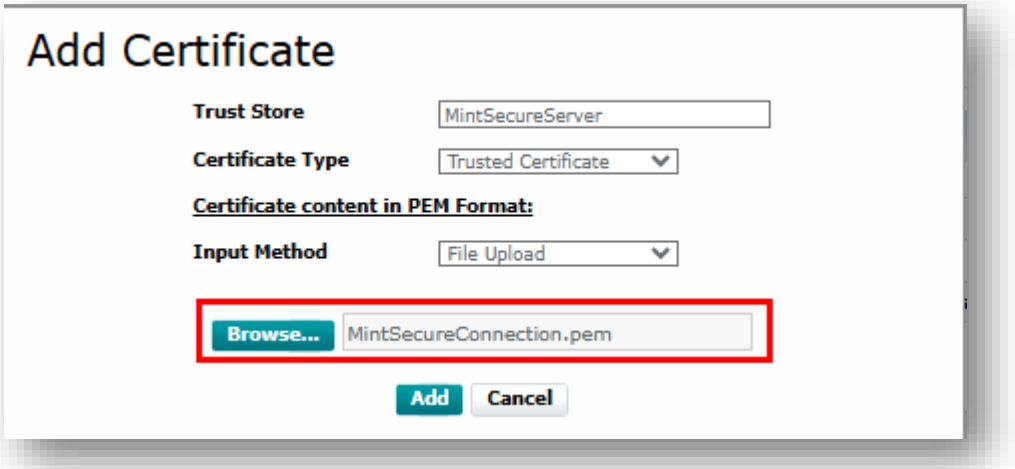
5.2 MULTI PLC certification

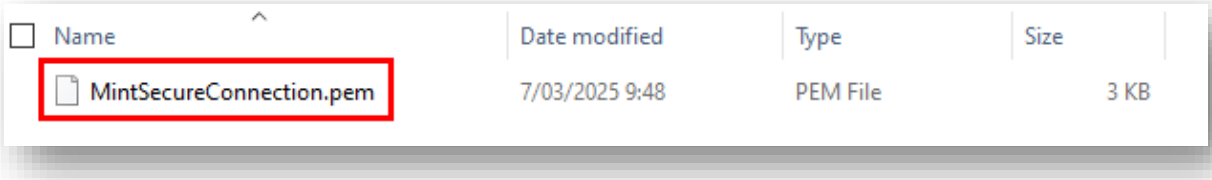
Navigate to the **WBM** (Web-Based Management) interface of all PLCs in the project.

5.2.1 Trust Stores

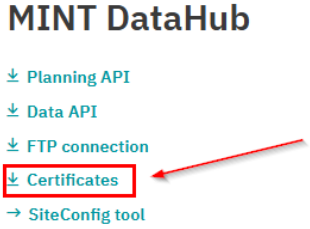




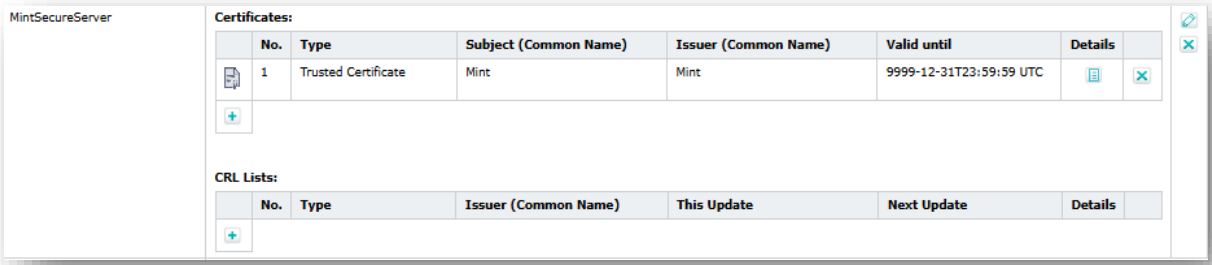




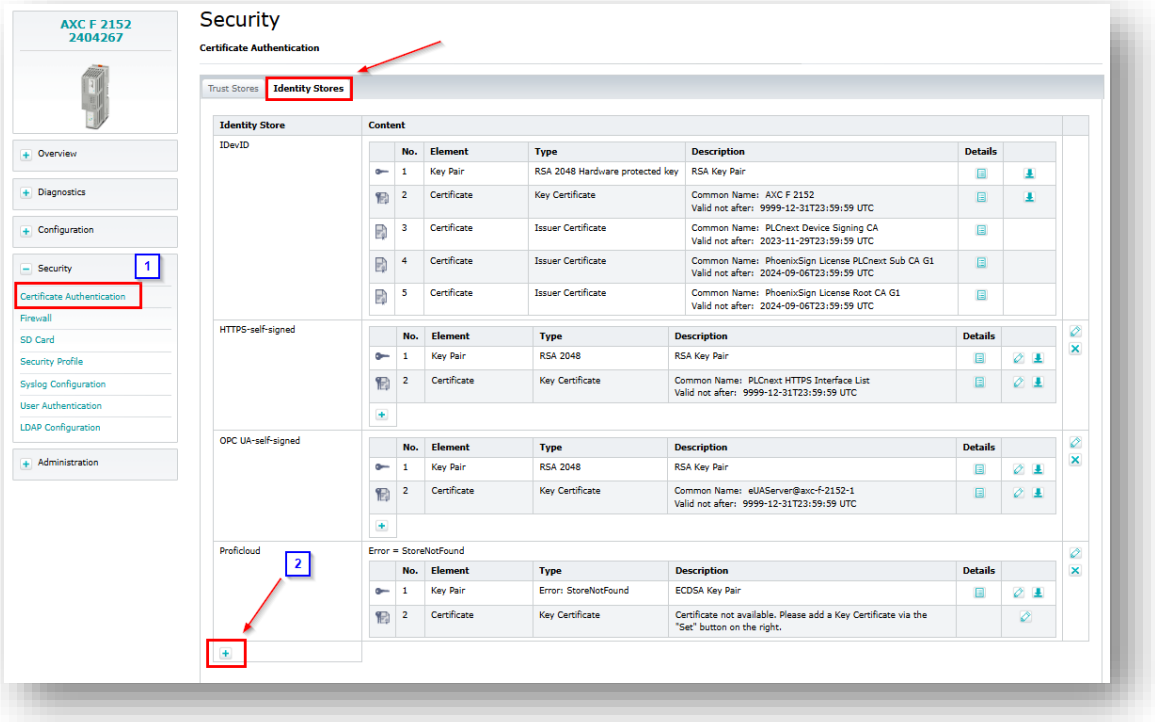
Upload the certificate, which can be downloaded via [MINT energiebeheer met AI | Phoenix Contact] under “MINT DataHub”.

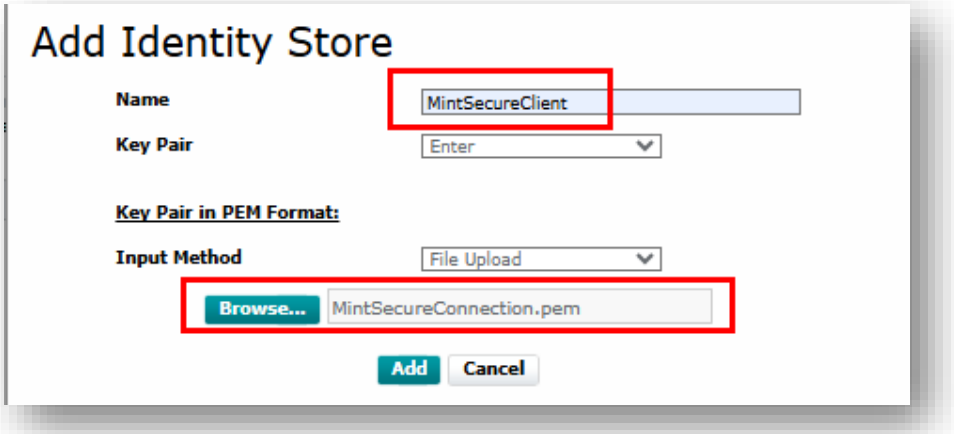


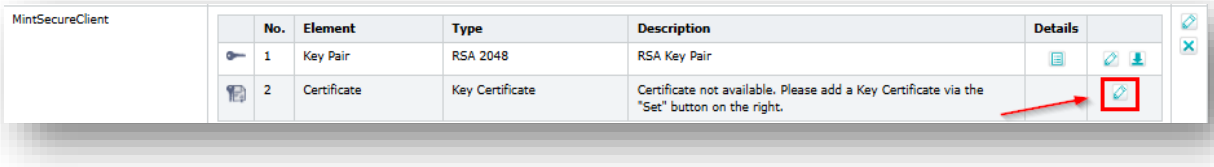
It should appear as shown below.

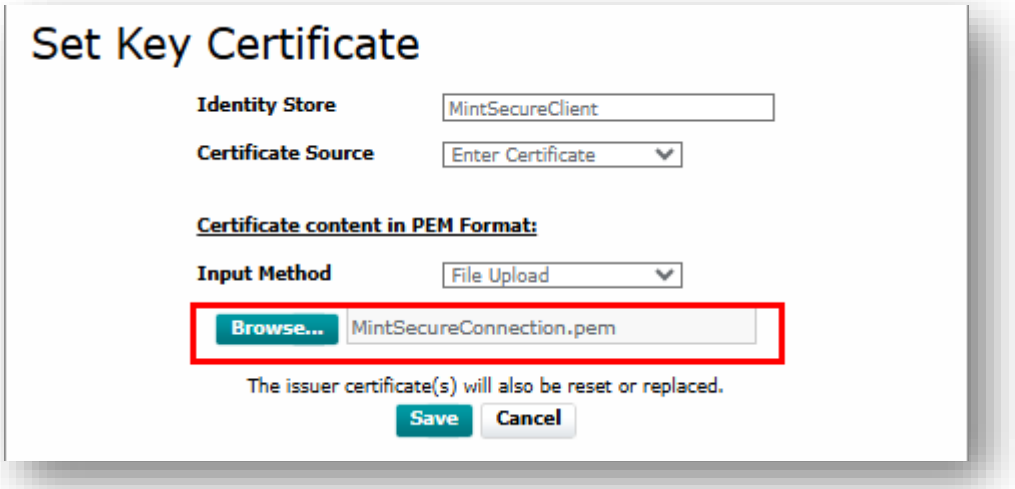


5.2.2 Identity Stores

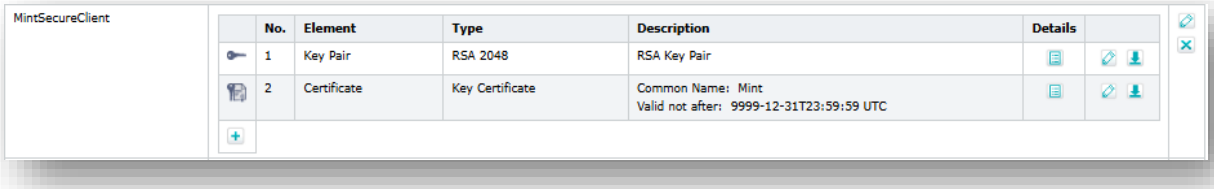






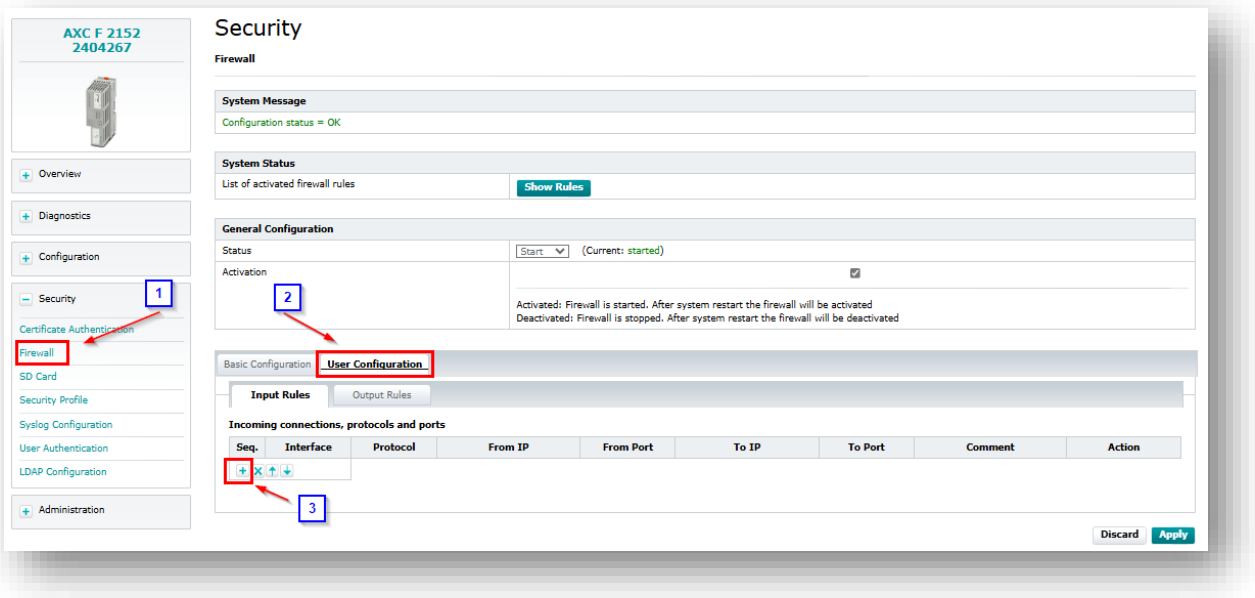


It should appear as shown below.



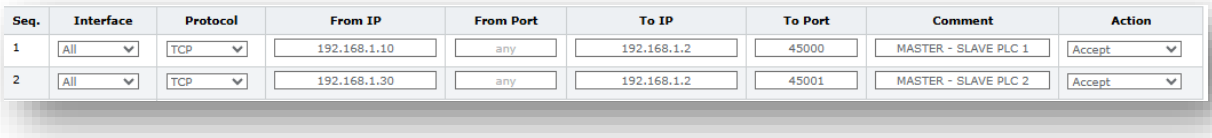
5.3 Firewall settings

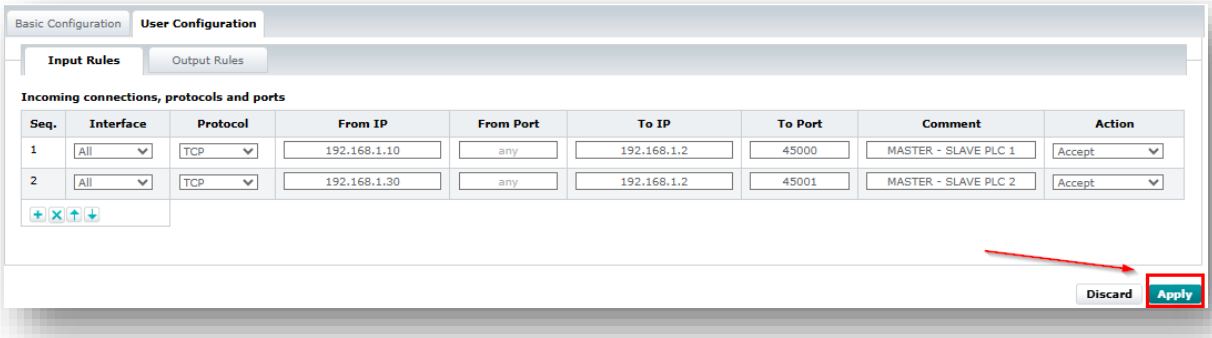
To enable communication between PLCs, the following firewall settings must be adjusted on the **main PLC**.



In the example below, we assume:

- MASTER PLC = 192.168.1.2
- SLAVE PLC 1 = 192.168.1.10 (Port 45000)
- SLAVE PLC 2 = 192.168.1.30 (Port 45001)





5.4 Changes in PLCnext Engineer

When building a MULTI PLC project, an additional line (or lines) will appear under the **“Configuration”** tab.
These lines refer to the MASTER and SLAVE PLCs.

5.4.1 MAIN PLC

- SUB PLC 1 “sub_1”

MINT_Core_ControlSystem.Configuration_Connection('Sub_1',**'xx.xx.xx.xx'**,UINT#45000);
Replace xx.xx.xx.xx with the corresponding IP address.

- SUB PLC 2 “sub_2”

MINT_Core_ControlSystem.Configuration_Connection('Sub_2',**'xx.xx.xx.xx'**,UINT#45001);
Replace xx.xx.xx.xx with the corresponding IP address.

```
3000: MINT_Core_ControlSystem.Configuration_Connection('Sub_1','xx.xx.xx.xx',UINT#45000);

3001: MINT_Core_ControlSystem.Configuration_Connection('Sub_2','xx.xx.xx.xx',UINT#45001);
```

5.4.2 SUB PLC 1 “Sub_1”

MINT_Core_ControlSystem.Configuration_Connection('Main',**'xx.xx.xx.xx'**,UINT#45000);
Replace xx.xx.xx.xx with the IP address of the MASTER PLC.

```
1: MINT_Core_ControlSystem.Configuration_Connection('Main','xx.xx.xx.xx',UINT#45000);
```

5.4.3 MAIN PLC 2 “Sub_2”

MINT_Core_ControlSystem.Configuration_Connection('Main',**'xx.xx.xx.xx'**,UINT#45001);
Replace xx.xx.xx.xx with the IP address of the MASTER PLC.

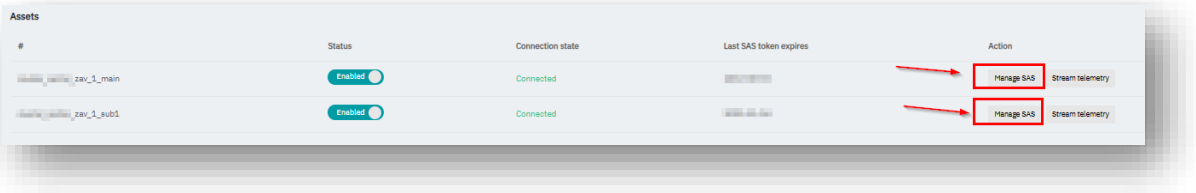
```
1: MINT_Core_ControlSystem.Configuration_Connection('Main','xx.xx.xx.xx',UINT#45001);
```

5.5 MUTLI PLC / MINT Licentie

Each PLC in a MULTI PLC project must have its own MINT licence.
The MASTER PLC will hold a licence for all assets and collectors.
The SLAVE PLCs will receive a SLAVE MINT licence.

5.6 MUTLI PLC / MQTT Credentials

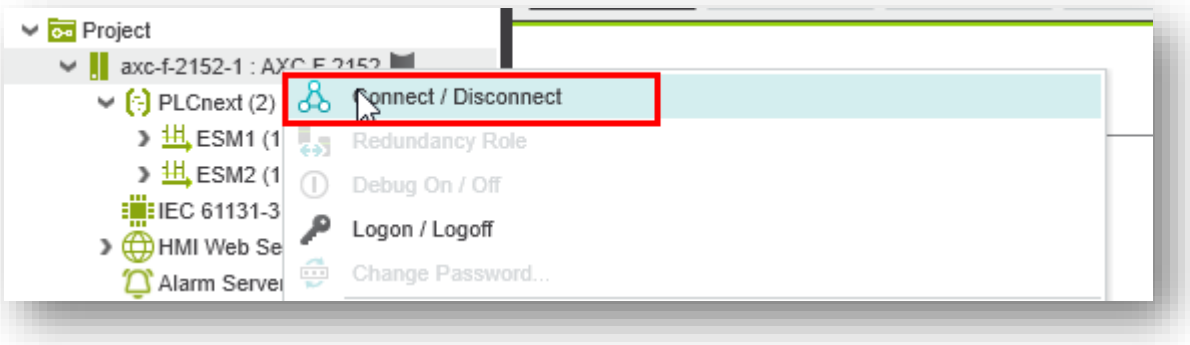
Each PLC in a MULTI PLC project must have its own MQTT connection (if applicable).
The SAS token can be generated per PLC.



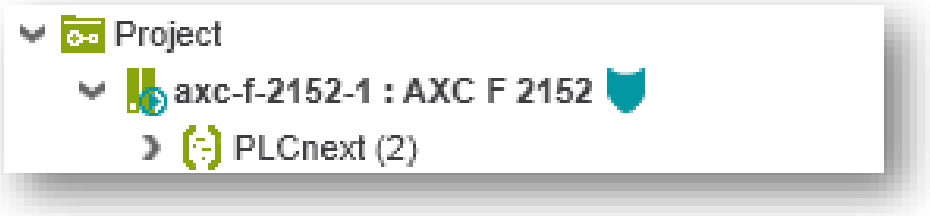
6 Starting a MINT project

Once all preparations are complete, the MINT software can be deployed on-site.
This begins by opening PLCnext Engineer 2024.0.4 LTS with the prepared MINT software.

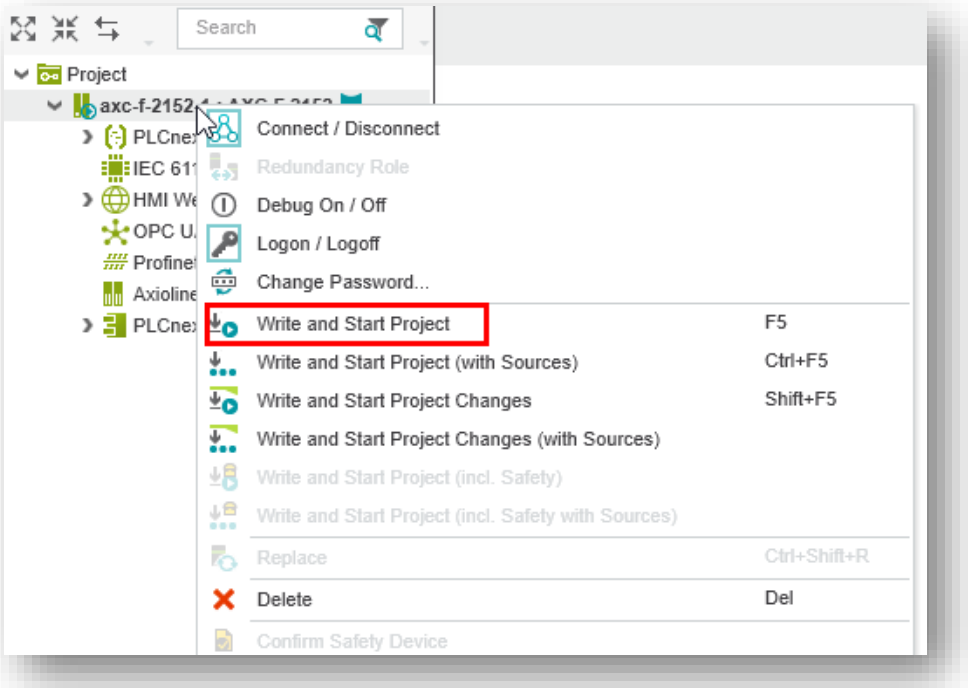
Right-click on “axc-f-2152-1 : AXC F 2152” and select “Connect / Disconnect”.



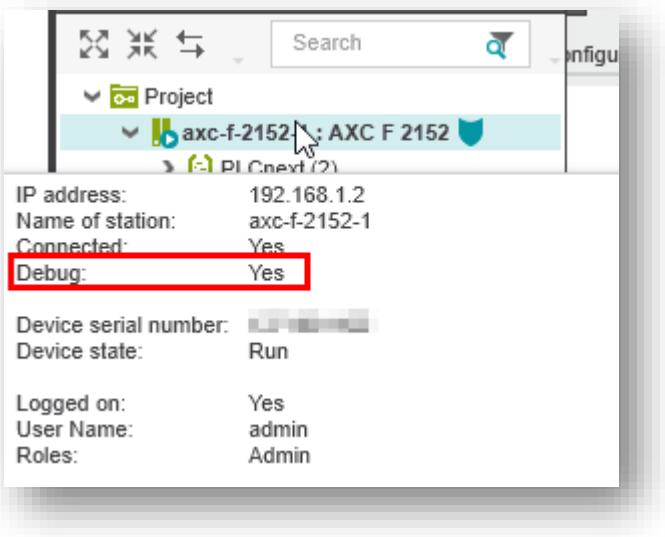
Log in using the PLC’s username and password.
If the connection to the PLC is successful, it will appear as follows:



Write and Start Project



If the build is successful, the PLC will automatically switch to debug mode.
You can confirm the debug status by hovering over “axc-f-2152-1 : AXC F 2152”.



6.1 Starting the MINT Project / MULTI PLC

For best practice, we recommend starting the **SLAVE PLCs first**, followed by the **MASTER PLC**.

7 Verifying the MINT Project

Nadat de MINT software is gedownload naar de PLC, dienen onderstaande checks uitgevoerd te worden om de software te valideren naar correcte werking.

7.1 MINT Licence Active

xControlSystemActive must be set to “TRUE”.

```
MINT_Core_ControlSystem(
    sLicense:= '[redacted]',
    strIncomingData:= strMINT_IncomingCommunicationDatahub(...), // Datahub communication
    strOutgoingData => strMINT_OutgoingCommunicationDatahub(...), // Datahub communication
    xActive => xControlSystemActive TRUE,
    arrDiagnostic=> arrControlSystemDiagnostic(...),
    arrCollectorDataInterface:= arrCollectorDataInterface(...), // Collector FB's
    arrACChargepointDataInterface:= arrACChargepointDataInterface(...), // AC Chargepoint FB's
    arrDCChargepointDataInterface:= arrDCChargepointDataInterface(...), // DC Chargepoint FB's
    arrPVSystemDataInterface:= arrPVSystemDataInterface(...)); // PV System FB's
```

7.2 MQTT Connection Active

The variable xConnect must be set to “TRUE”.

```
MINT_FB_Datahub_Interface(  xActivate := strMINT_Datahub_IN.strSettings.xConnect TRUE,
                            sHost := strMINT_Datahub_IN.strSettings.sHost[redacted],
                            sUsername := strMINT_Datahub_IN.strSettings.sUserName[redacted],
                            sPassword := strMINT_Datahub_IN.strSettings.sPassword[redacted],
                            sPort := strMINT_Datahub_IN.strSettings.sPort[redacted],
                            sClientID :=  strMINT_Datahub_IN.strSettings.sDatahubId[redacted],
                            sPersistDir := '',
                            diBufferMessages := DINT#100,
                            strConnectOptions := strConnectOptions(...),
                            xConnected => xConnected TRUE,
                            xTimeout => xTimeout FALSE,
                            xError => xError FALSE,
                            strDiagnose => strDiagnose(...),
                            xReadyToReceive => xReadyToReceive TRUE,
                            xNewDataAvailable => xNewDataAvailable FALSE);
```

7.3 Collector data

Are the energy values from the collector being received correctly?

If no vehicles are currently charging, the collector may not show current values. However, voltage should always be present.

```
// *** Collector *** HFDTR ********************************************************************************

Collector_001.Input_Measurements(           TO_DINT(IN_HFDTR.strRead.strInstantaneous.strData.strP.rScaledValue          19069.81),
                                            TO_DINT(IN_HFDTR.strRead.strInstantaneous.strData.strP1.rScaledValue          6797.94),
                                            TO_DINT(IN_HFDTR.strRead.strInstantaneous.strData.strP2.rScaledValue          6298.95),
                                            TO_DINT(IN_HFDTR.strRead.strInstantaneous.strData.strP3.rScaledValue          5972.93),
                                            IN_HFDTR.strRead.strInstantaneous.strData.strI1.rScaledValue          36.83,
                                            IN_HFDTR.strRead.strInstantaneous.strData.strI2.rScaledValue          28.57,
                                            IN_HFDTR.strRead.strInstantaneous.strData.strI3.rScaledValue          28.64);
Collector_001.Input_TotalPowerFactor(       IN_HFDTR.strRead.strInstantaneous.strData.strPF.rScaledValue          0.86);
Collector_001.Input_TotalReactivePower(     TO_DINT(IN_HFDTR.strRead.strInstantaneous.strData.strQ.rScaledValue          -11467.81));
Collector_001.Input_TotalEnergy(            TO_ULINT(IN_HFDTR.strRead.strMeterReadings.strData.strTotalActiveEnergyDemand.rScaledValue          8.1050541
                                            TO_ULINT(IN_HFDTR.strRead.strMeterReadings.strData.strTotalActiveEnergyDelivery.rScaledValue          99
Collector_001.Input_ConductorVoltage(       IN_HFDTR.strRead.strInstantaneous.strData.strU12.rScaledValue          413.26,
                                            IN_HFDTR.strRead.strInstantaneous.strData.strU23.rScaledValue          412.18,
                                            IN_HFDTR.strRead.strInstantaneous.strData.strU31.rScaledValue          413.87);
Collector_001.Input_Voltage(                IN_HFDTR.strRead.strInstantaneous.strData.strU1.rScaledValue          239.14,
                                            IN_HFDTR.strRead.strInstantaneous.strData.strU2.rScaledValue          238.11,
                                            IN_HFDTR.strRead.strInstantaneous.strData.strU3.rScaledValue          238.27);
Collector_001.Input_ReactivePower(          TO_DINT(IN_HFDTR.strRead.strInstantaneous.strData.strQ1.rScaledValue          -5600.56),
                                            TO_DINT(IN_HFDTR.strRead.strInstantaneous.strData.strQ2.rScaledValue          -2566.74),
                                            TO_DINT(IN_HFDTR.strRead.strInstantaneous.strData.strQ3.rScaledValue          -3300.52));
Collector_001.Input_PowerFactor(            IN_HFDTR.strRead.strInstantaneous.strData.strPF1.rScaledValue          0.77,
                                            IN_HFDTR.strRead.strInstantaneous.strData.strPF2.rScaledValue          0.93,
                                            IN_HFDTR.strRead.strInstantaneous.strData.strPF3.rScaledValue          0.88);
Collector_001.Input_Frequency(              IN_HFDTR.strRead.strInstantaneous.strData.strF.rScaledValue          49.98);

Collector_001(sName:= 'HFDTR',
xCommunicationValid:= IN_HFDTR.strDiag.iSecondsSinceLastSuccessfulTransaction          1 < 30,
arrDataInterface:= arrCollectorDataInterface[...]);
```

```
Collector_001.Input_ConductorVoltage(       IN_HFDTR.strRead.strInstantaneous.strData.strU12.rScaledValue          413.26,
                                            IN_HFDTR.strRead.strInstantaneous.strData.strU23.rScaledValue          412.18,
                                            IN_HFDTR.strRead.strInstantaneous.strData.strU31.rScaledValue          413.87);
Collector_001.Input_Voltage(                IN_HFDTR.strRead.strInstantaneous.strData.strU1.rScaledValue          239.14,
                                            IN_HFDTR.strRead.strInstantaneous.strData.strU2.rScaledValue          238.11,
                                            IN_HFDTR.strRead.strInstantaneous.strData.strU3.rScaledValue          238.27);
```

7.4 Charging Station Data

Is data from the charging stations being received correctly?

If there is no active charging session at a station, no current or power will be visible. However, voltage should still be present.

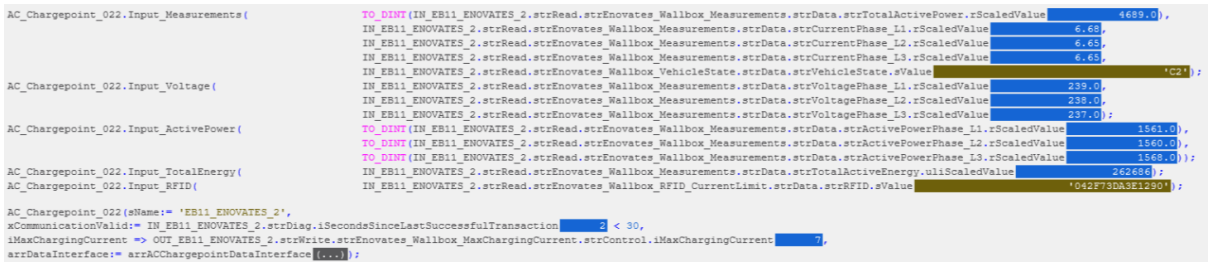
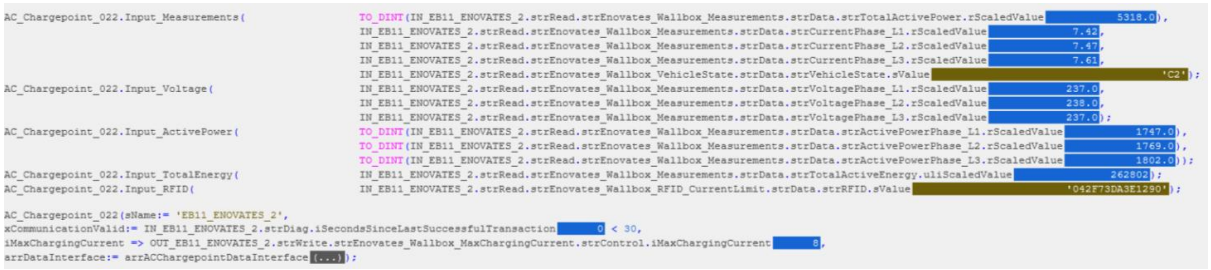
```
// *** AC Chargepoint *** EB11_AC1_1 ***********************************************************************

AC_Chargepoint_014.Input_Measurements(          TO_DINT(IN_EB11_AC1_1.strRead.strModule_1_Measurement.strData.strTotalActivePower.rScaledValue          0.0),
                                                IN_EB11_AC1_1.strRead.strModule_1_Measurement.strData.strCurrentPhase_L1.rScaledValue          0.0,
                                                IN_EB11_AC1_1.strRead.strModule_1_Measurement.strData.strCurrentPhase_L2.rScaledValue          0.0,
                                                IN_EB11_AC1_1.strRead.strModule_1_Measurement.strData.strCurrentPhase_L3.rScaledValue          0.0,
                                                IN_EB11_AC1_1.strRead.strModule_1_Measurement.strData.strVehicleStatus.sValue          'A1');
AC_Chargepoint_014.Input_Voltage(               IN_EB11_AC1_1.strRead.strModule_1_Measurement.strData.strVoltagePhase_L1_N.rScaledValue          238.6,
                                                IN_EB11_AC1_1.strRead.strModule_1_Measurement.strData.strVoltagePhase_L2_N.rScaledValue          239.53,
                                                IN_EB11_AC1_1.strRead.strModule_1_Measurement.strData.strVoltagePhase_L3_N.rScaledValue          238.73);
AC_Chargepoint_014.Input_ChargingPlugCapacity(  IN_EB11_AC1_1.strRead.strModule_1_Measurement.strData.strChargingPlugCapacity.iScaledValue          0);
AC_Chargepoint_014.Input_TotalEnergy(           IN_EB11_AC1_1.strRead.strModule_1_Measurement.strData.strTotalActiveEnergy.uliScaledValue          5734407);
AC_Chargepoint_014.Input_RFID(                  IN_EB11_AC1_1.strRead.strModule_1_Measurement.strData.strBadgeRFID.sValue          'FAEBBBF5');

AC_Chargepoint_014(sName:= 'EB11_AC1_1',
xCommunicationValid:= IN_EB11_AC1_1.strDiag.iSecondsSinceLastSuccessfulTransaction          1 < 30,
iMaxChargingCurrent => OUT_EB11_AC1_1.strWrite.strModule_1_MaxChargingCurrent.strControl.iMaxChargingCurrent          6,
arrDataInterface:= arrACChargepointDataInterface[...]);
```

```
AC_Chargepoint_014.Input_Voltage(               IN_EB11_AC1_1.strRead.strModule_1_Measurement.strData.strVoltagePhase_L1_N.rScaledValue          238.6,
                                                IN_EB11_AC1_1.strRead.strModule_1_Measurement.strData.strVoltagePhase_L2_N.rScaledValue          239.53,
                                                IN_EB11_AC1_1.strRead.strModule_1_Measurement.strData.strVoltagePhase_L3_N.rScaledValue          238.73);
```

7.5 MINT Scaling

The MINT software will scale up the allowed setpoint of the charging stations if sufficient power is available. See the example below:

Setpoint of 7A:



Setpoint of 8A: