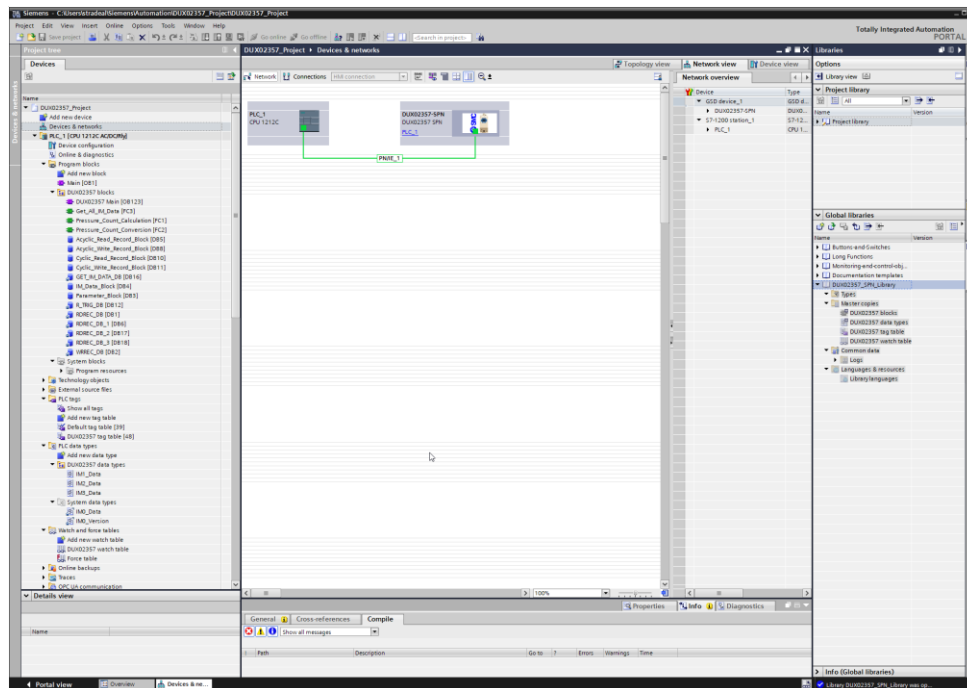




# PROFINET ITV (DUX02357) TIA Portal™ Library Manual



**SMC Corporation**



Revision History		
Revision	Description	Date
1.0	Initial release (DRAFT).	06/17/2021

# Table of Contents

<b>1. About this Manual.....</b>	<b>- 10 -</b>
<b>2. Getting Started.....</b>	<b>- 10 -</b>
2.1. General Information.....	- 10 -
<b>3. Terminology .....</b>	<b>- 10 -</b>
<b>4. Importing the Global Library.....</b>	<b>- 11 -</b>
<b>5. Adding the Library Items to the TIA Portal Project.....</b>	<b>- 13 -</b>
<b>6. Compiling the Library .....</b>	<b>- 14 -</b>
<b>7. Downloading the Library .....</b>	<b>- 16 -</b>
<b>8. Verify Library Operations.....</b>	<b>- 17 -</b>
8.1. Going Online .....	- 17 -
8.2. Checking Operations .....	- 21 -
<b>9. Program Blocks .....</b>	<b>- 22 -</b>
9.1. Organization Blocks.....	- 23 -
9.1.1. DUX02357 Main.....	- 23 -
9.1.1.1. Network 1 - Display Actual Pressure in Engineering Units (Cyclic Read).....	- 23 -
9.1.1.2. Network 2 - Calculate Pressure Count and Set ITV to Desired Pressure (Cyclic Write) .....	- 24 -
9.1.1.3. Network 3 - Send Acyclic Read command .....	- 25 -
9.1.1.4. Network 4 - Send Acyclic Write command.....	- 26 -
9.1.1.5. Network 5 - Read all I&M data (I&M0, I&M1, I&M2, and I&M3) .....	- 28 -
9.2. Functions.....	- 28 -
9.2.1. Reading Identification and Maintenance Data (Get_All_IM_Data) .....	- 28 -
9.2.2. Calculate Pressure Counts (Pressure_Count_Calculation) .....	- 32 -
9.2.3. Convert Pressure Counts (Pressure_Count_Conversion) .....	- 33 -
9.3. Data Blocks.....	- 33 -
9.3.1. Acyclic_Read_Record_Block.....	- 34 -
9.3.2. Acyclic_Write_Record_Block.....	- 34 -
9.3.3. Cyclic_Read_Record_Block.....	- 34 -
9.3.4. Cyclic_Write_Record_Block.....	- 35 -
9.3.5. GET_IM_DATA_DB .....	- 35 -
9.3.6. IM_Data_Block.....	- 35 -
9.3.7. Parameter_Block.....	- 36 -
9.3.8. R_TRIG_DB.....	- 37 -
9.3.9. RDREC_DB, RDREC_DB_1, RDREC_DB_2, and RDREC_DB_3.....	- 37 -
9.3.10. WRREC_DB.....	- 37 -
9.4. Function Blocks.....	- 37 -
<b>10. PLC Tags .....</b>	<b>- 38 -</b>
10.1. DUX02357 Tag Table and User Constants .....	- 38 -
10.1.1. Tag Table.....	- 38 -
10.1.2. User Constants.....	- 39 -
<b>11. PLC Data Types .....</b>	<b>- 40 -</b>
11.1. IM0_Data and IM0_Version.....	- 40 -
11.2. IM1_Data .....	- 41 -
11.3. IM2_Data .....	- 41 -
11.4. IM3_Data .....	- 41 -
<b>12. Watch and Force Tables .....</b>	<b>- 41 -</b>
12.1. DUX02357 Watch Table.....	- 41 -



## List of Tables

Table 1 – Definitions and Terminology .....	- 11 -
Table 2 – Acyclic_Read_Record_Block Structure .....	- 34 -
Table 3 – Acyclic_Write_Record_Block Structure.....	- 34 -
Table 4 – Cyclic_Read_Record_Block Structure.....	- 35 -
Table 5 – Cyclic_Write_Record_Block Structure .....	- 35 -
Table 6 – IM_Data_Block Structure .....	- 36 -
Table 7 – Parameter_Block Structure .....	- 37 -
Table 8 – Tag Table.....	- 39 -
Table 9 – User Constants .....	- 40 -
Table 10 – IM0_Data and IM0_Version Structures.....	- 40 -
Table 11 – IM1_Data Structure .....	- 41 -
Table 12 – IM2_Data Structure .....	- 41 -
Table 13 – IM3_Data Structure .....	- 41 -
Table 14 – Watch Table.....	- 43 -
Table 15 – Force Table .....	- 43 -

## List of Figures

Figure 1 – Open Global Library Menu Item .....	- 11 -
Figure 2 – Global Library File Type Pulldown Menu .....	- 12 -
Figure 3 – Selecting the PROFINET ITV Global Library .....	- 12 -
Figure 4 – Selecting the TIA Portal Project.....	- 12 -
Figure 5 – PROFINET ITV Global Library Contents.....	- 13 -
Figure 6 – PROFINET ITV Global Library Contents.....	- 14 -
Figure 7 – Compiling the Hardware Configuration .....	- 15 -
Figure 8 – Compiling Icon .....	- 15 -
Figure 9 – Compile Status Window .....	- 15 -
Figure 10 – System Blocks.....	- 16 -
Figure 11 – System Data Types.....	- 16 -
Figure 12 – Downloading the Hardware Configuration .....	- 16 -
Figure 13 – Compiling Button .....	- 17 -
Figure 14 – Load Preview Window .....	- 17 -
Figure 15 – Load Preview Load Button.....	- 17 -
Figure 16 – Load Preview Finish Button .....	- 17 -
Figure 17 – Placing the PLC Online .....	- 18 -
Figure 18 – Go Online Button .....	- 18 -
Figure 19 – Setting the Subnet.....	- 19 -
Figure 20 – Start Search Button .....	- 19 -
Figure 21 – Placing the PLC Online .....	- 20 -
Figure 22 – PLC Online.....	- 20 -
Figure 23 – Selecting DUX02357 Organization Block.....	- 21 -
Figure 24 – Enable Monitoring.....	- 21 -
Figure 25 – Setting Parameters for Reading the Supply Voltage.....	- 21 -
Figure 26 – Reading the Supply Voltage .....	- 22 -
Figure 27 – Supply Voltage Displayed .....	- 22 -
Figure 28 – Program Blocks.....	- 23 -
Figure 29 – Cyclic Read Ladder Logic .....	- 24 -
Figure 30 – Cyclic Write Ladder Logic.....	- 25 -
Figure 31 – Acyclic Read Ladder Logic.....	- 26 -
Figure 32 – Acyclic Write Ladder Logic .....	- 27 -
Figure 33 – Identification and Maintenance Read Ladder Logic .....	- 28 -
Figure 34 – Get_All_IM_Data Parameters and Variables .....	- 29 -
Figure 35 – Get_All_IM_Data SCL.....	- 32 -

Figure 36 – Pressure Count Calculation Parameters and Variables..... - 32 -  
Figure 37 – Pressure Count Calculation SCL..... - 33 -  
Figure 38 – Pressure Count Conversion Parameters and Variables ..... - 33 -  
Figure 39 – Pressure Count Conversion SCL..... - 33 -

# **End-User License Agreement ("Agreement")**

Please read this End-User License Agreement ("Agreement") carefully before downloading or using the DUX02357 TIA Portal Library ("Application").

SMC Corporation provides and maintains this Application as a convenience to you and other users. Although SMC Corporation reserves the right at any time and for any reason to refuse access to or edit or remove content from this Application, you acknowledge and agree to accept sole responsibility and liability for any Application content posted, transmitted, downloaded, or used by you. SMC Corporation has no obligation to monitor Site content and does not endorse or guarantee the accuracy or completeness of the information contained therein.

By downloading or using the Application, you are agreeing to be bound by the terms and conditions of this Agreement.

If you do not agree to the terms of this Agreement, do not download or use the Application.

## **License**

SMC Corporation of America grants a revocable, non-exclusive, limited license to download, install and use the Application in accordance with the terms of this Agreement.

The copyright in material downloadable or accessible this Application is held by SMC Corporation or by the original creator of the applicable material. You may not delete or alter or attempt to delete or alter any attribution, legal notice, trademark or copyright notice appearing on any Application content and such notices or marks must be included on any copy you make. Furthermore, you will not reproduce, redistribute or repost on other web sites materials downloaded from this Site unless expressly authorized to do so by the relevant material copyright owner.

You must take your own precautions to ensure that whatever content you select from this Application is free of computer viruses or Trojan horses that may interfere with or damage the functionality of this Site or the operations of your or any third party computer system.

## **Modifications to Application**

SMC Corporation of America reserves the right to modify, suspend or discontinue, temporarily or permanently, the Application or any service to which it connects, with or without notice and without liability to the user.

## **Term and Termination**

This Agreement shall remain in effect until terminated by you or SMC Corporation of America.

SMC Corporation of America may, in its sole discretion, at any time and for any or no reason, suspend or terminate this Agreement with or without prior notice.

This Agreement will terminate immediately, without prior notice from SMC Corporation of America, in the event that you fail to comply with any provision of this Agreement.

Upon termination of this Agreement, you shall cease all use of the Application and delete all copies of the Application from your device(s).

## **Severability**

If any provision of this Agreement is held to be unenforceable or invalid, such provision will be changed and interpreted to accomplish the objectives of such provision to the greatest extent possible under applicable law and the remaining provisions will continue in full force and effect.

## **Exemption of Application**

SMC Corporation is not responsible for any damage incurred in the use of this software.

## **Amendments to this Agreement**



SMC Corporation of America reserves the right, at its sole discretion, to modify or replace this Agreement at any time. In the event of a revision to the material, we will provide at least thirty (30) days notice prior to any new terms taking effect. What constitutes a material change will be determined at our sole discretion.

**Contact Information**

If you have any questions about this Agreement, please contact us.

SMC Corporation of America

10100 SMC Boulevard

Noblesville, IN 46060

Phone: 1-800-SMC-SMC1 (1-800-762-7621)

URL: [www.smcusa.com](http://www.smcusa.com)

## 1. About this Manual

The purpose of this manual is to provide the user with necessary information to import and use the PROFINET ITV (DUX02357) TIA-Portal® library.

## 2. Getting Started

### 2.1. General Information

- This manual is supplied by SMC Corporation of America As-Is and is to be used as reference only.
- This manual is to be used with TIA-Portal® version 16 to demonstrate importing of the library.
- This version of the library will not be able to be imported by other versions of TIA-Portal®.
- The PLC used in this manual is a Siemens S7-1212C and the folder structure may be different for other models of PLCs.
- Other configuration of the system may be needed (ID of the ITV, cyclic memory locations, etc.).
- The PLC and ITV (along with the GSDML file) must already be installed for the example in this document. Please see the manufacturer's documentation for detailed instructions on the setup and use of your Siemens or other PLC. Also refer to the "Operation Manual - PROFINET ITV - IN22335.docx" for the setup of the ITV and the installation of the GSDML file. These instructions will vary with other configurations.

## 3. Terminology

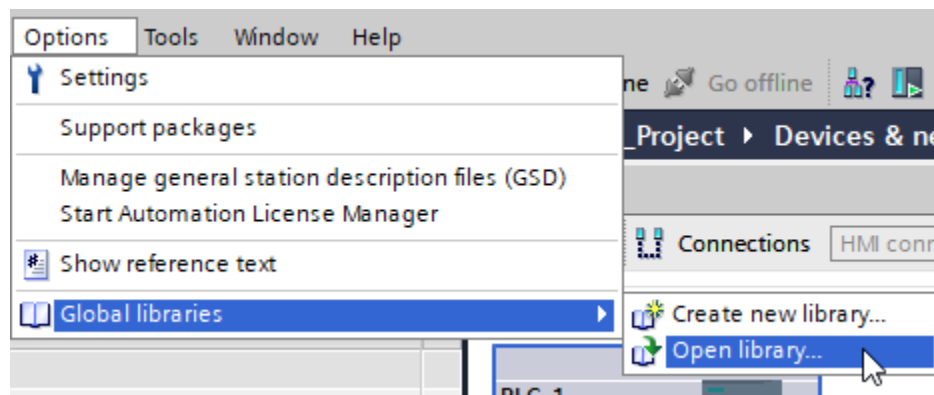
	Terms	Meaning
A	A/D	Analog to Digital
B	BF	Bus Fault
	Bus In / Bus Out	M12 Ethernet communication connectors located on top of the ITV.
C	Counts	The ITV controls and reports pressure or vacuum internally in Counts. The Counts are used by hardware devices to control and measure the pressure or vacuum regulated by the ITV.
	Current consumption	The current necessary to operate each unit or device using 24 VDC.
D	Device Name	The Device Name (or Station Name) - used for unique identification of the device in the network.
	Downstream	Relative to your ITV, a downstream device receives power and/or communication signals which originate <i>from</i> or are <i>passed through</i> your ITV. This also refers to pneumatic devices controlled by the ITV outlet pressure.
	DUX02357-SPN	PROFINET™ ITV
F	Fieldbus	The protocol that uses digital communication to exchange signals between field equipment (instruments and actuators).
G	GSD	General Station Description - Device description for configuring IO devices within the engineering system.
	GSDML	General Station Description Markup Language - An XML-based language, the GSDML file is used as the language for a device description file. The file contains settable attribute information for a device (each parameter's object size, etc.). It is used by Siemens PLC TIA Portal™ software to permit faster, more consistent PLC setup.
I	I&M	Identification and Maintenance - general information functions about devices (for example manufacturer, version, ordering data, etc.)
	IO	Input/Output
	IOC	Input/Output Controller
	IOD	Input/Output Device
	IP address	A 32-bit digit sequence which is assigned to identify devices which are connected to the network. This address must be unique for each device to ensure proper operation of the network.
L	LED	Light Emitting Diode - Indicator for signaling device states.
M	MAC	Media Access Control

	MAC address	A unique number inherent to all devices which are connected to Ethernet.
P	PC	Personal Computer.
	PI	PROFINET International – PROFINET governing body.
	PLC	Programmable Logic Controller - a digital computer/controller used for automation of electromechanical processes.
	PNIO	PROFINET Input/Output
	Power Cycle	A Power Cycle involves briefly removing power from the ITV and then restoring power.
	PROFINET™	The specific Fieldbus (network) communication protocol used by the ITV to communicate with the PLC, and any other connected devices. PROFINET is a well-documented, standard communications protocol.
	PROFINET™ ITV	DUX02357-SPN
	PWR	Power
S	SCL	Structured Control Language – programming language used for Siemens PLCs.
	Setpoint	The specified pressure or vacuum value which the ITV has been directed to maintain.
	SF	System Fault
	SI unit	Abbreviation for Serial Interface unit. A unit connected to a device to communicate input and output data. The ITV connection to another device is a Serial Interface Unit (SI Unit).
	Station Name	The Station Name (or Device Name) - used for unique identification of the device in the network.
	Subnet Address	The 3 <sup>rd</sup> octet of an IP Address. Also called Subnet.
T	TIA Portal™	Totally Integrated Automation Portal (TIA Portal) - software developed by Siemens that combines a wide range of Siemens software to create automation solutions as well as tools for testing.
	Topology	Connection configuration of the network.
U	Upstream	Relative to your ITV, an upstream device provides power and/or communication signals to your ITV.

**Table 1 – Definitions and Terminology**

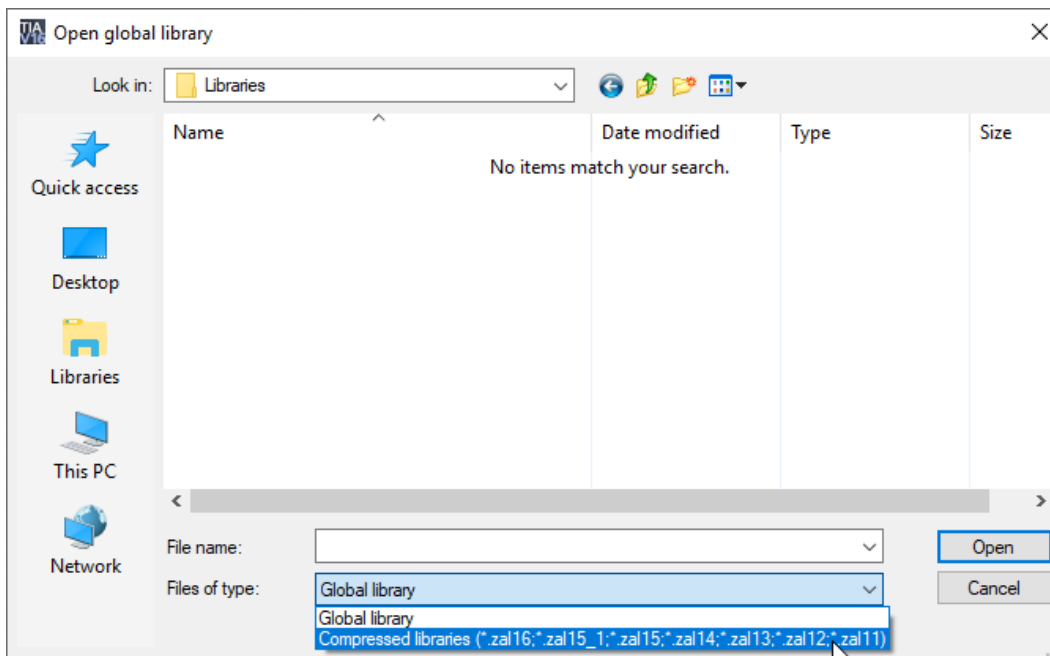
## 4. *Importing the Global Library*

Before using the DUX02357 Global Library, it must be imported into the current project. This can be done by opening the library by clicking on the “Open library...” from the Main Menu “Options...Global libraries...Open library...”.



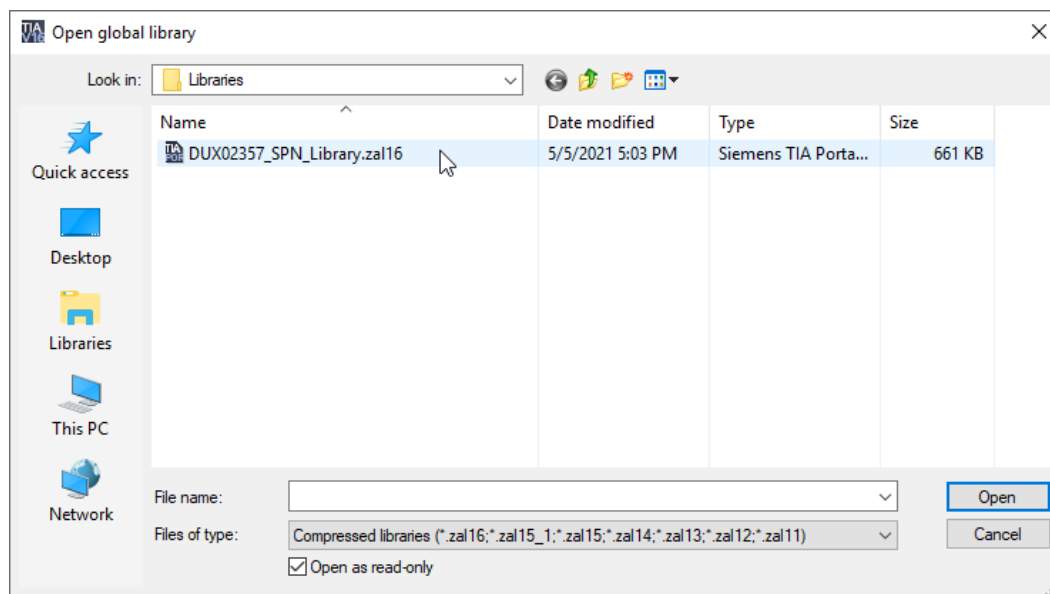
**Figure 1 – Open Global Library Menu Item**

Next, change the File Type to “Compressed libraries” by clicking on the “Files of type” pulldown menu.



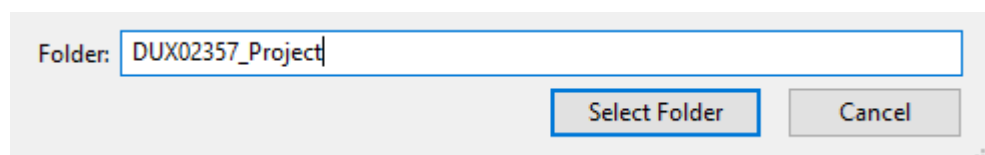
**Figure 2 – Global Library File Type Pulldown Menu**

Now, navigate to where the DUX02357-SPN library has been stored. Select the DUX02357-SPN global library by double clicking on it.



**Figure 3 – Selecting the PROFINET ITV Global Library**

Next, enter the folder name of the current TIA Portal project and then click on the “Select Folder” button to decompress the global library.



**Figure 4 – Selecting the TIA Portal Project**

The DUX02357 Global Library is now added to the global library list.

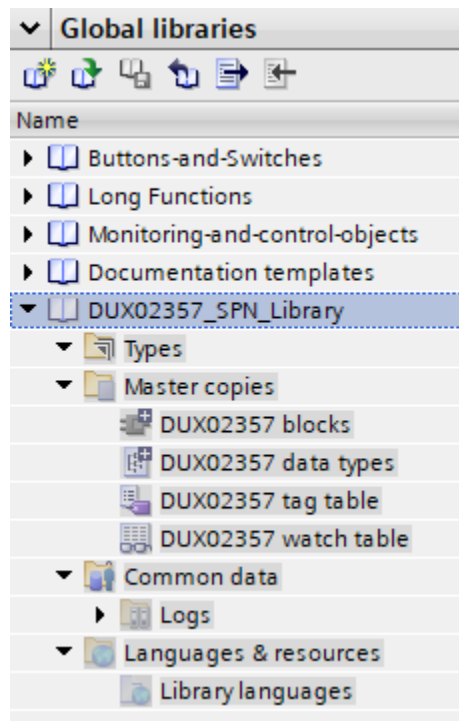


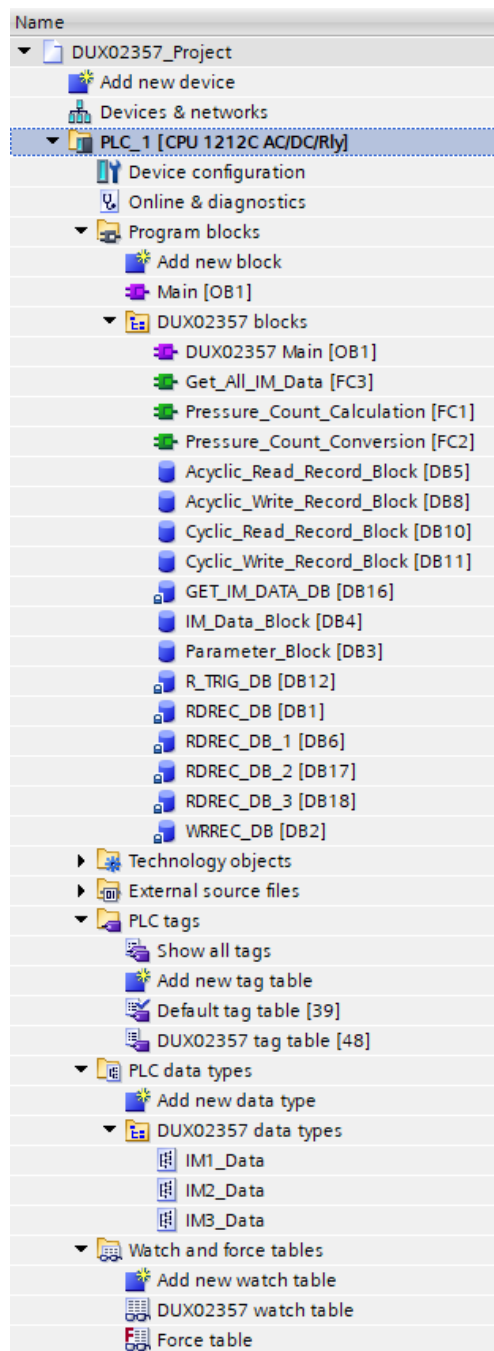
Figure 5 – PROFINET ITV Global Library Contents

## 5. *Adding the Library Items to the TIA Portal Project*

The library now needs to be added to the existing TIA Portal project. Begin this process by dragging and dropping each item of the following four (4) sections of the library into the corresponding PLC section of the project:

- 1) DUX02357 blocks → Program blocks
- 2) DUX02357 data types → PLC data types
- 3) DUX02357 tag table → PLC tags
- 4) DUX02357 watch table → Watch and force tables

Once everything has been integrated, the PLC section should look like the following:

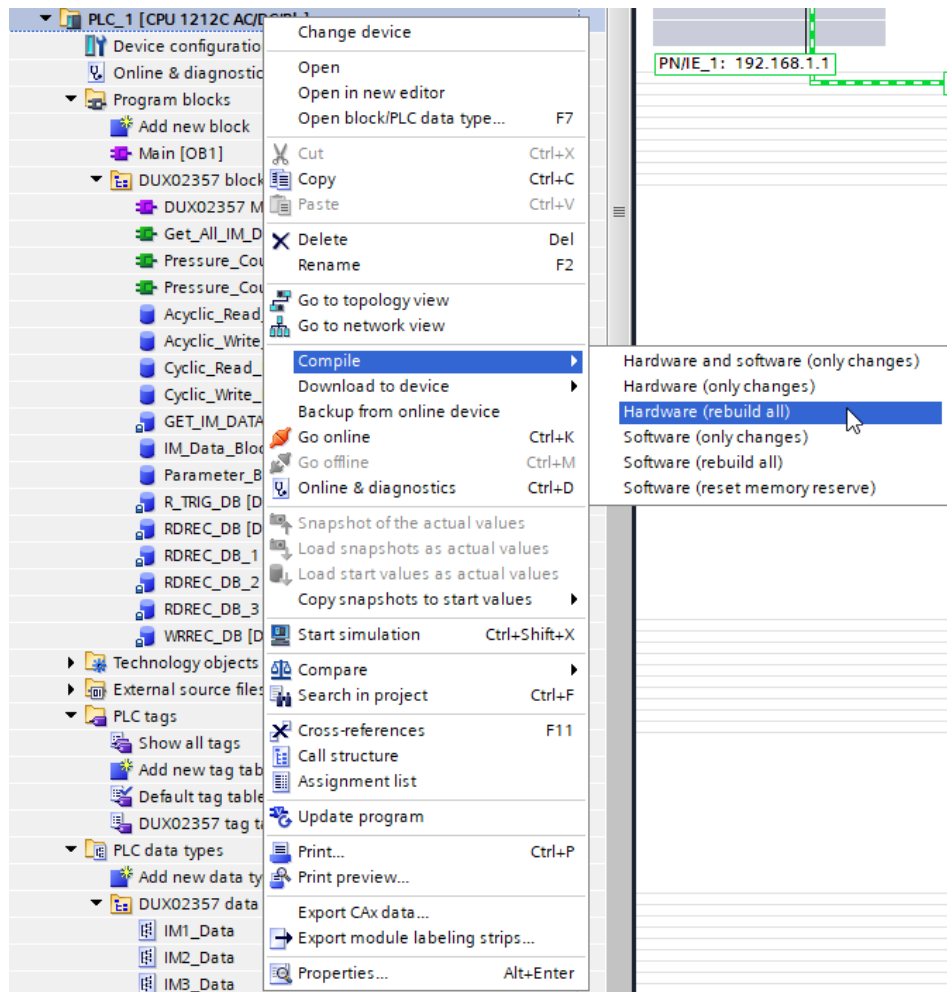


**Figure 6 – PROFINET ITV Global Library Contents**

Items can now be moved to different folders and folders can be renamed if needed.

## 6. *Compiling the Library*

The next step is to compile the library. The first thing to do is right click on the PLC under the “Project tree...Devices” and navigate down the menu tree to select “Compile...Hardware (rebuild all)”.



**Figure 7 – Compiling the Hardware Configuration**

Repeat the previous step but this time select “Compile...Software (rebuild all)” to compile the software. You can also use the icon to compile.



**Figure 8 – Compiling Icon**

There should be no errors or warnings in the Compile window.

!	Path	Description	Go to	?	Errors	Warnings	Time
✓	PLC_1		↗		0	0	3:17:33 PM
✓	PLC data types		↗		0	0	3:17:35 PM
✓	DUX02357 data types		↗		0	0	3:17:40 PM
✓	Program blocks		↗		0	0	3:17:36 PM
✓	DUX02357 blocks		↗		0	0	3:17:36 PM
✓	Main (OB1)	Block was successfully compiled.	↗				3:17:37 PM
✓		Compiling finished (errors: 0; warnings: 0)					3:17:44 PM

**Figure 9 – Compile Status Window**

After compiling, there will be additional TIA Portal library items under “System data types” and the “System blocks...Program resources”. These additional items are brought in from the internal TIA Portal library and are used by the DUX02357 Global Library.

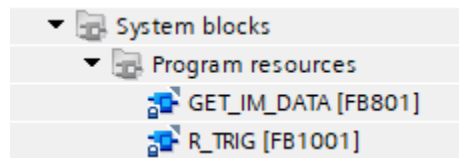


Figure 10 – System Blocks

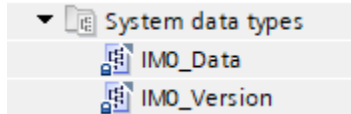


Figure 11 – System Data Types

## 7. *Downloading the Library*

The configuration and software now needs to be downloaded to the PLC. Right click on the PLC under the “Project tree...Devices” and navigate down the menu tree to select “Download to device...Hardware configuration”.

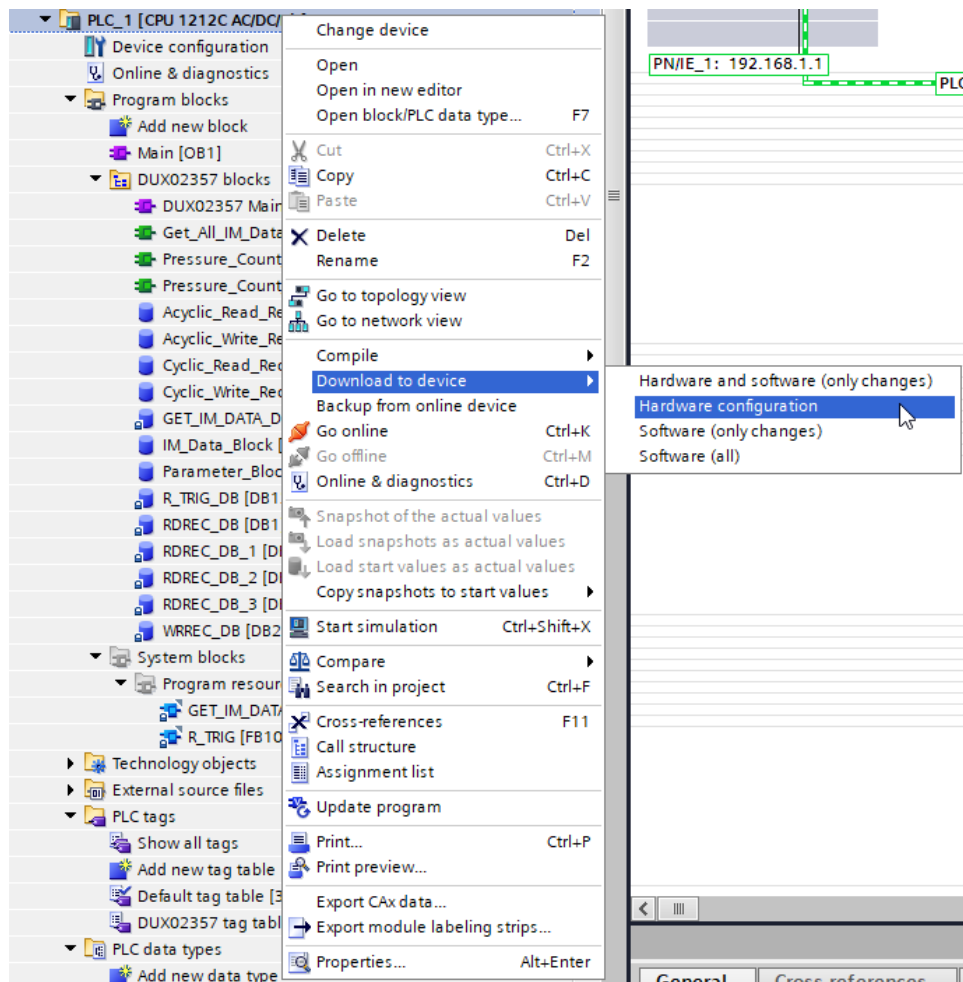


Figure 12 – Downloading the Hardware Configuration

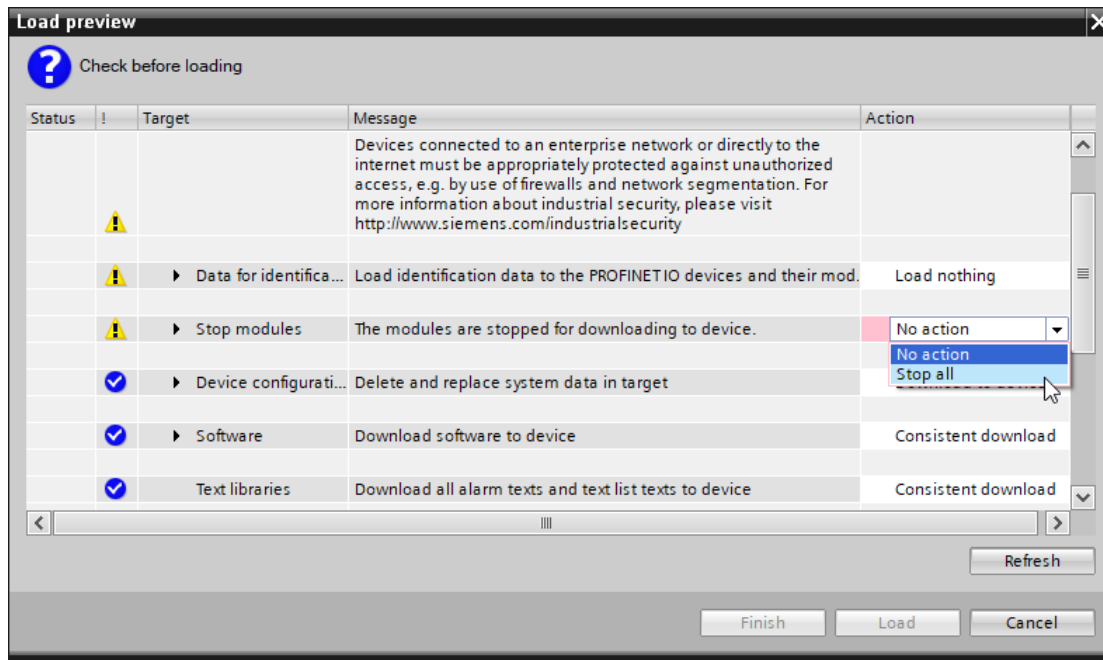
You can also use the download icon.





**Figure 13 – Compiling Button**

The “Load preview” window will appear and if needed, you may need to stop the module by clicking on the “Stop all” item in the “Stop modules” pulldown menu.



**Figure 14 – Load Preview Window**

Next, click on the Load button.



**Figure 15 – Load Preview Load Button**

After the configuration has been downloaded, click on the Finish button



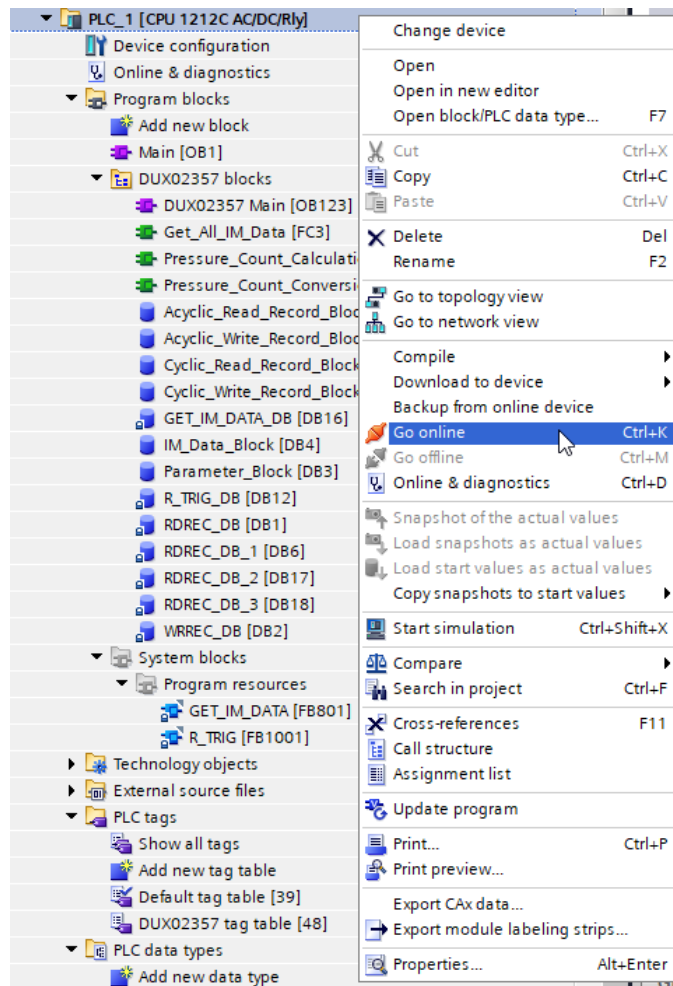
**Figure 16 – Load Preview Finish Button**

Repeat the previous step but this time select “Download to device...Software (all)”. Again, the “Load preview” window will appear and click on the Load button and once TIA Portal is finished then click on the Finish button.

## 8. *Verify Library Operations*

### 8.1. **Going Online**

To verify the operations of the library, the PLC must first be placed online. To start this process, right click on the PLC under the “Project tree...Devices” and navigate down the menu tree to select “Go online”.



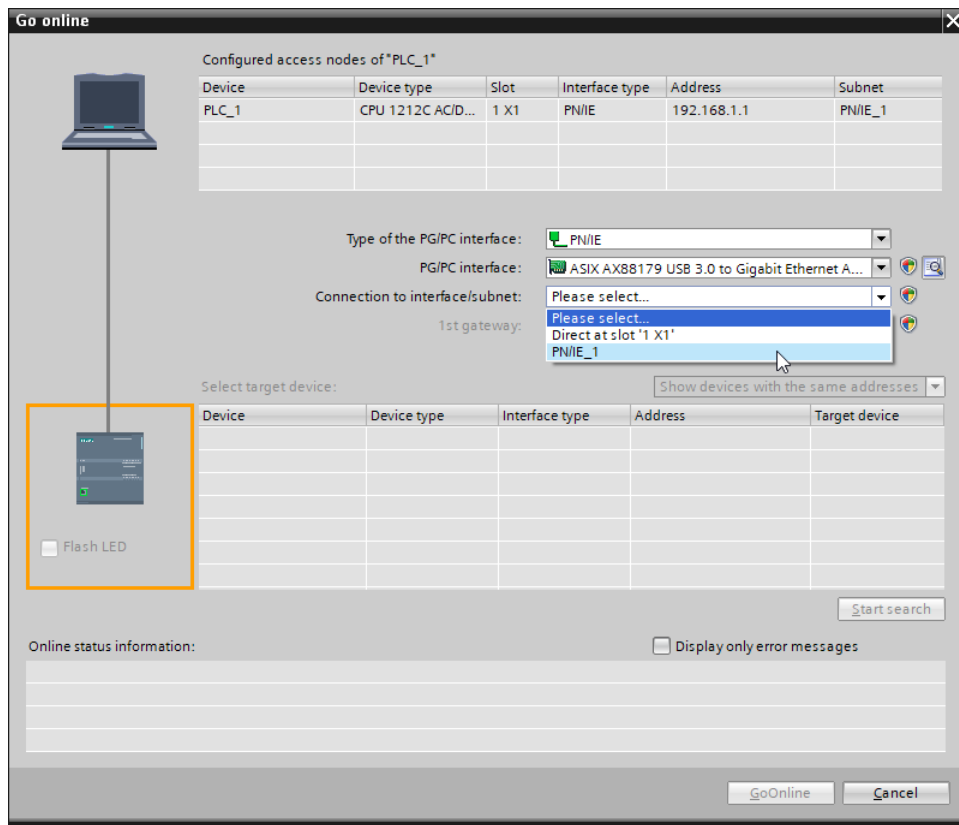
**Figure 17 – Placing the PLC Online**

You can also use the “Go online” icon.



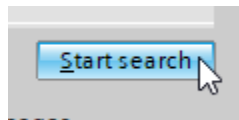
**Figure 18 – Go Online Button**

You may need to configure the system by first selecting the subnet in the “Go online” window.



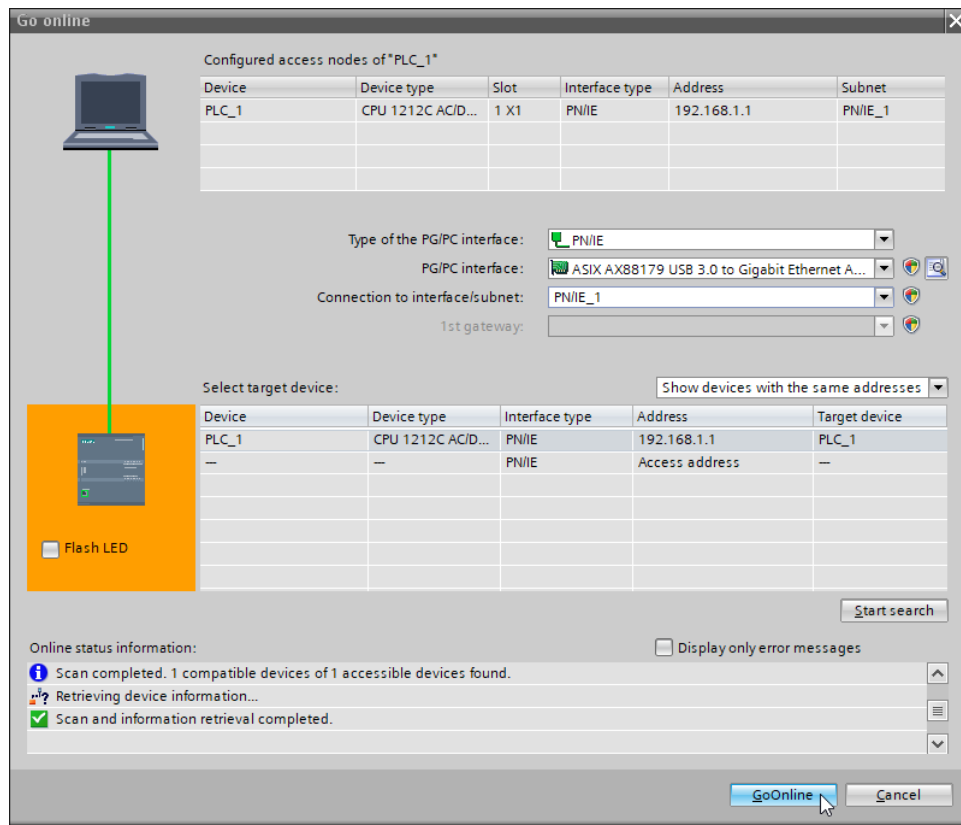
**Figure 19 – Setting the Subnet**

You will then need to start the search by clicking the “Start search” button.



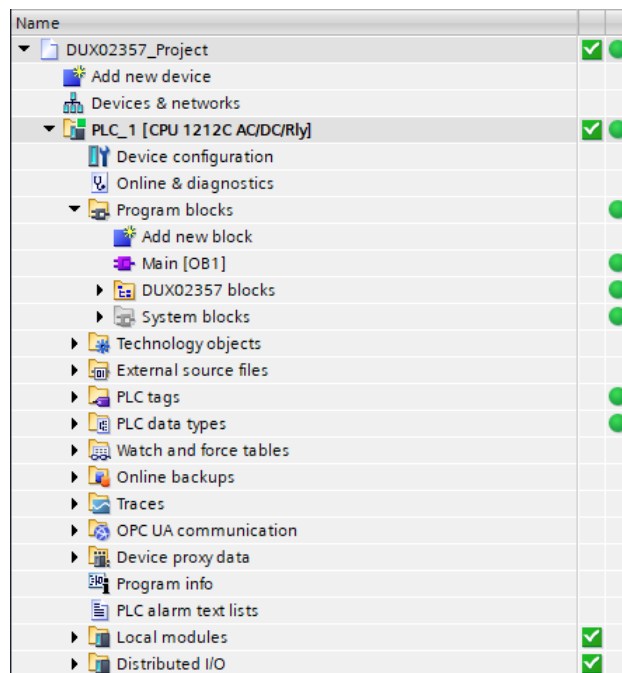
**Figure 20 – Start Search Button**

The PLC should be displayed in the “Select target device” window and you can then go online by clicking the “GoOnline” button. If the PLC is not displayed, then you may need to adjust the network settings.



**Figure 21 – Placing the PLC Online**

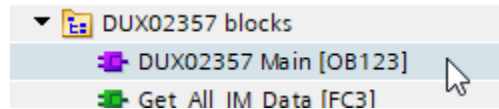
If everything is working correctly then there should be green checkmarks by the PLC and the overall system in the "Project tree...Devices" window.



**Figure 22 – PLC Online**

## 8.2. Checking Operations

The functionality of the PROFINET ITV can be checked by examining the Supply Voltage acyclic message. To check, double click on the DUX02357 Organization Block to bring up the ladder logic.



**Figure 23 – Selecting DUX02357 Organization Block**

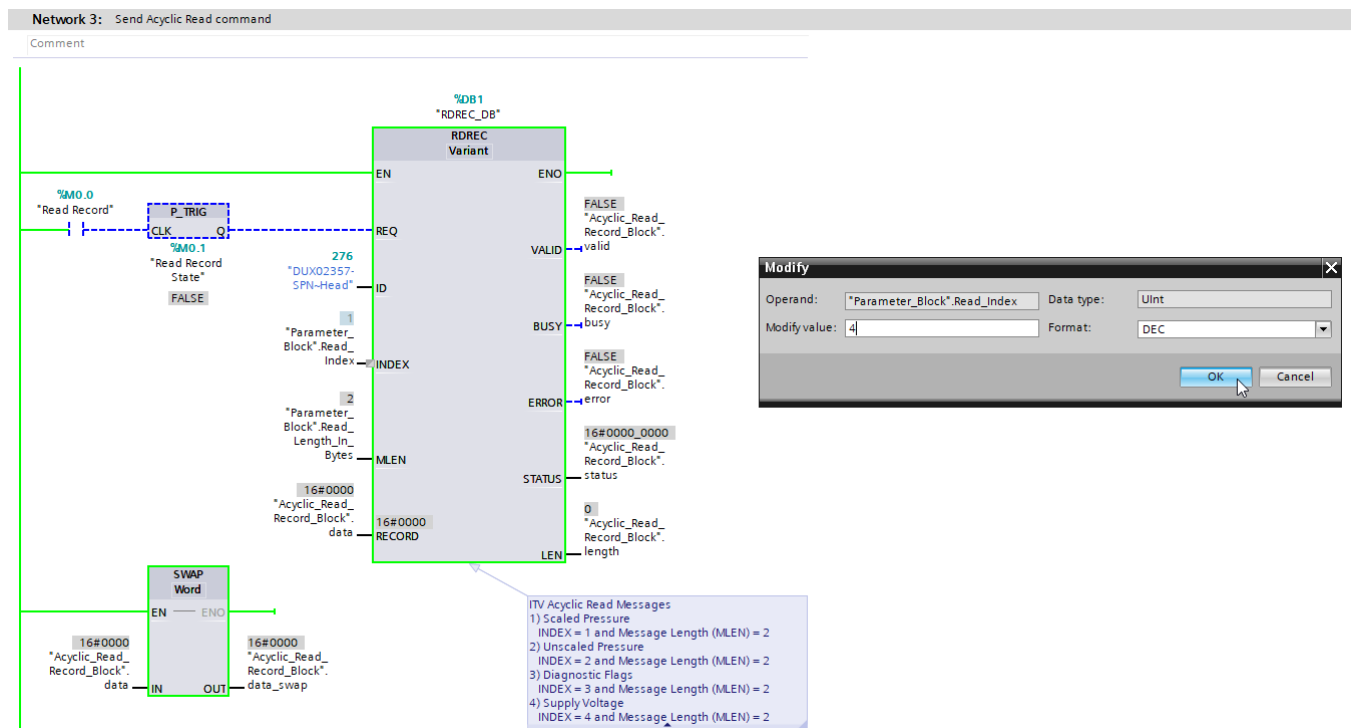
Once the ladder logic is displayed, enable the monitoring by clicking on the “Monitoring on/off” icon.



**Figure 24 – Enable Monitoring**

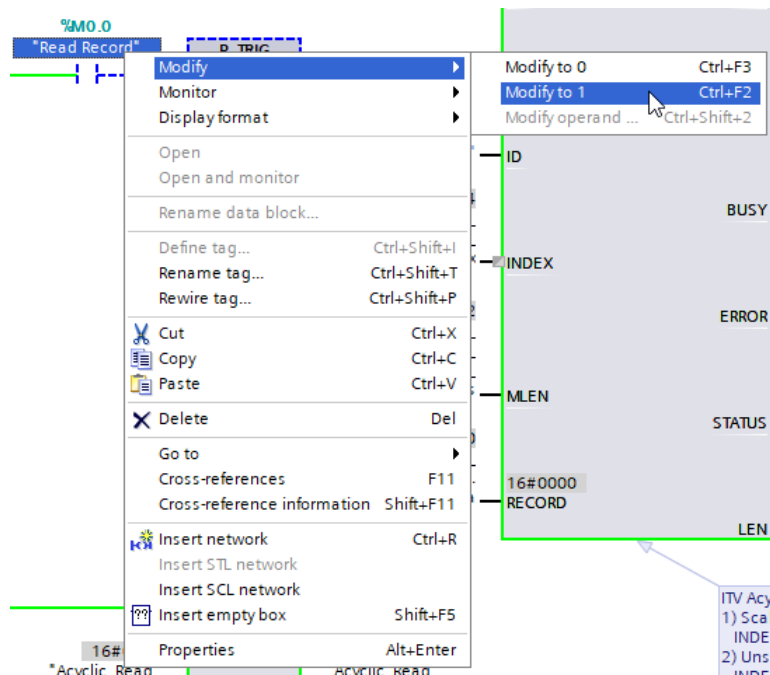
If the Ladder Logic is not highlighted in green once the button is pressed, then make sure that the PLC is in the Run mode.

Next, scroll down until “Network 3 – Send Acyclic Read command” is shown. Double click on the value for “Parameter\_Block.Read\_index”. Change this value to 4 and then click on the OK button. The “Parameter\_Block.Read\_Length\_in\_Bytes” value should be 2 by default so it should not need to be changed.



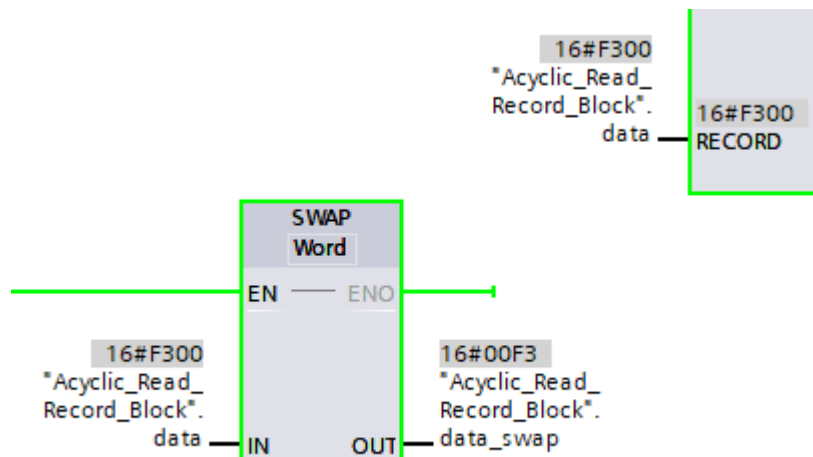
**Figure 25 – Setting Parameters for Reading the Supply Voltage**

Next, send the message to the PROFINET ITV by toggling the “Read Record” bit. You can toggle the bit by right clicking on the “Read Record” and navigating down the menu and clicking on the “Modify...Modify to 1”.



**Figure 26 – Reading the Supply Voltage**

The block output data should have changed.



**Figure 27 – Supply Voltage Displayed**

In this case the data returned was 0x00F3 which is 243. The voltage can be calculated by dividing this number by 10 which will give you 24.3 volts.

## 9. Program Blocks

The Program Blocks control the operations of the PROFINET ITV. The following sections describe the different blocks.

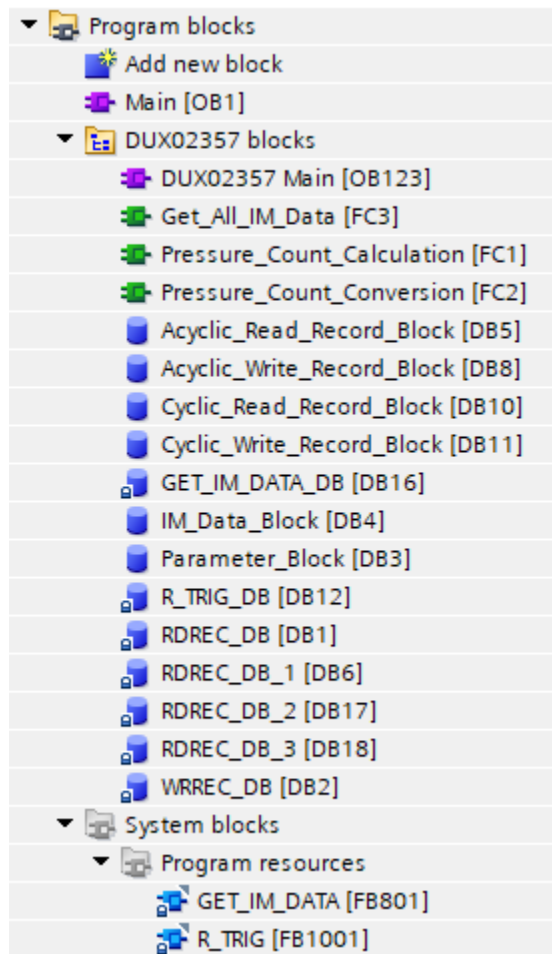


Figure 28 – Program Blocks

## 9.1. Organization Blocks

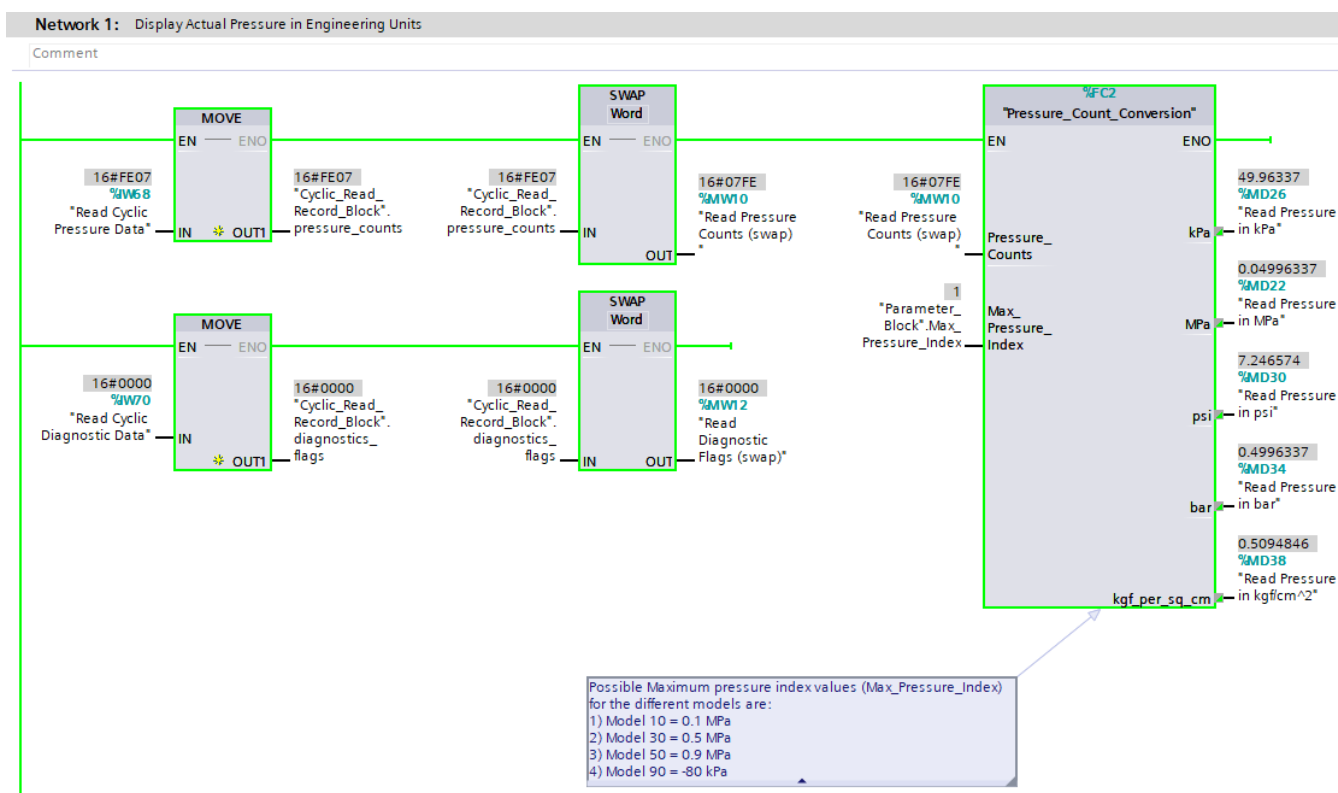
The Organization Blocks contain specific user programs that control the system as well as providing the interface between the operating system and the user program.

### 9.1.1. DUX02357 Main

The DUX02357 Main programs provide software for reading and writing both cyclic and acyclic messages to and from the PROFINET ITV. The software uses both Ladder Logic and SCL in controlling of the PLC.

#### 9.1.1.1. Network 1 - Display Actual Pressure in Engineering Units (Cyclic Read)

The Network 1 software takes the pressure and diagnostic cyclic data and manipulates it. First, the software performs byte swap on both sets of data in order that the data can be used for other programming needs. Next, the software takes the pressure data and converts it from raw counts to engineering units that are easily understood. The units that are calculated are kPa, MPa, psi, bar, and kgf/cm<sup>2</sup>. The user needs to provide the index that represents which model of ITV is being used so that the pressure can be scaled accordingly for the engineering unit calculations. Please refer to the **Convert Pressure Counts** section for more information.



**Figure 29 – Cyclic Read Ladder Logic**

#### 9.1.1.2. Network 2 - Calculate Pressure Count and Set ITV to Desired Pressure (Cyclic Write)

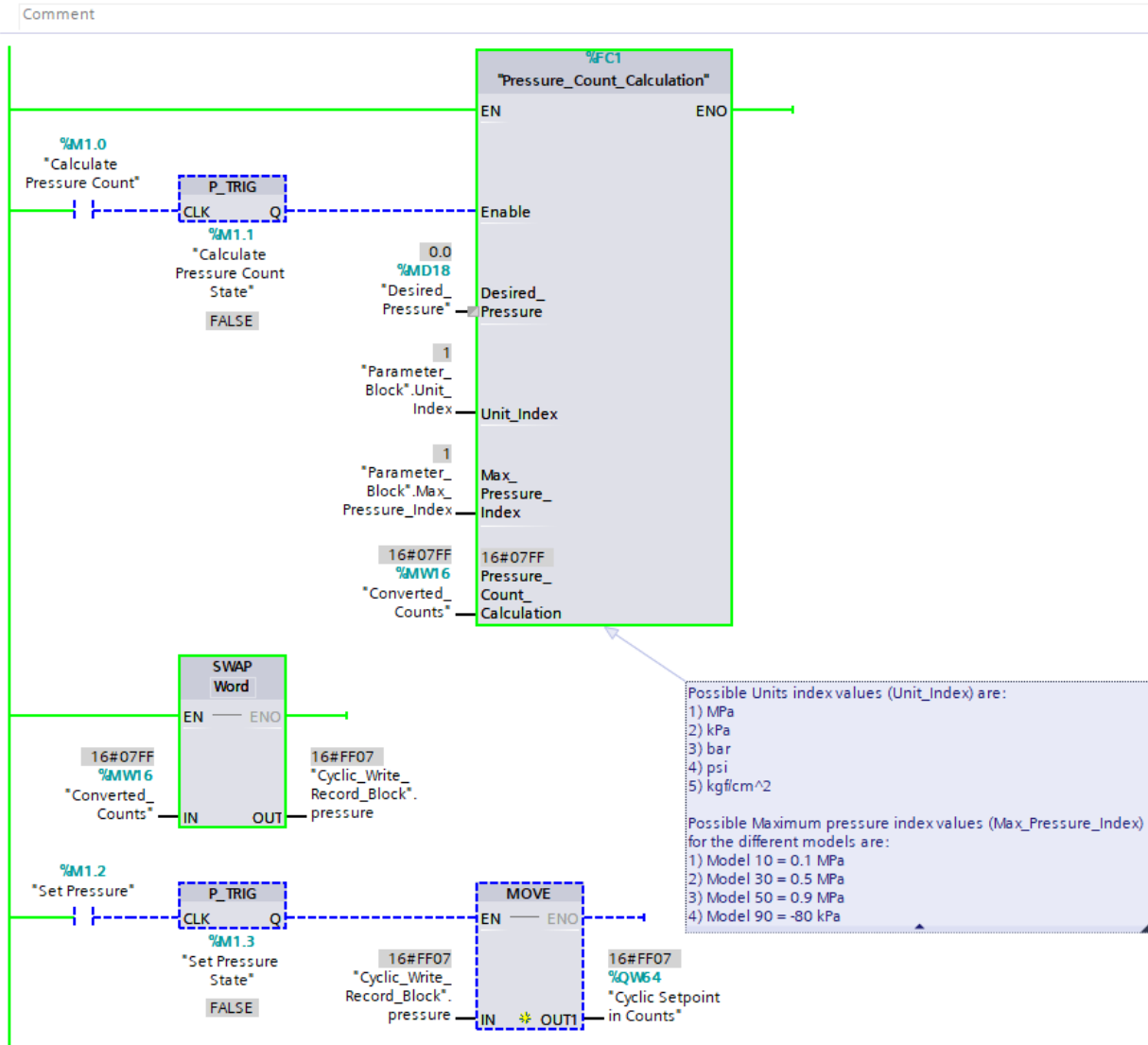
The Network 2 software takes an entered pressure and sets the ITV to the requested pressure. The user needs to provide the following:

- 1) Pressure – decimal value of the desired pressure.
- 2) Unit Index - index value that represents the units of the requested pressure (see **User Constants** section).
- 3) Model Index - index that represents which model of ITV is being used so that the pressure can be scaled accordingly for the count calculation (see **User Constants** section).

Separate logic signals are provided to perform the count calculation as well as the sending of the data to the ITV. Also, logic is provided to perform the byte swap operation on the sent data. Please refer to the **Calculate Pressure Counts** section for more information.



## Network 2: Calculate Pressure Count and Set ITV to Desired Pressure



**Figure 30 – Cyclic Write Ladder Logic**

### 9.1.1.3. Network 3 - Send Acyclic Read command

The Network 3 software performs all acyclic read operations. The user needs to provide the following information:

- 1) ID – the ID of the ITV to communicate to.
- 2) Read Index – index value that represents the appropriate data to read (see below). The Read\_Index member of the Parameter\_Block is used to hold this value (see **Parameter\_Block** section).
- 3) Message Length – length of returned data in bytes (see below). The Read\_Length\_in\_Bytes member of the Parameter\_Block is used to hold this value (see **Parameter\_Block** section).

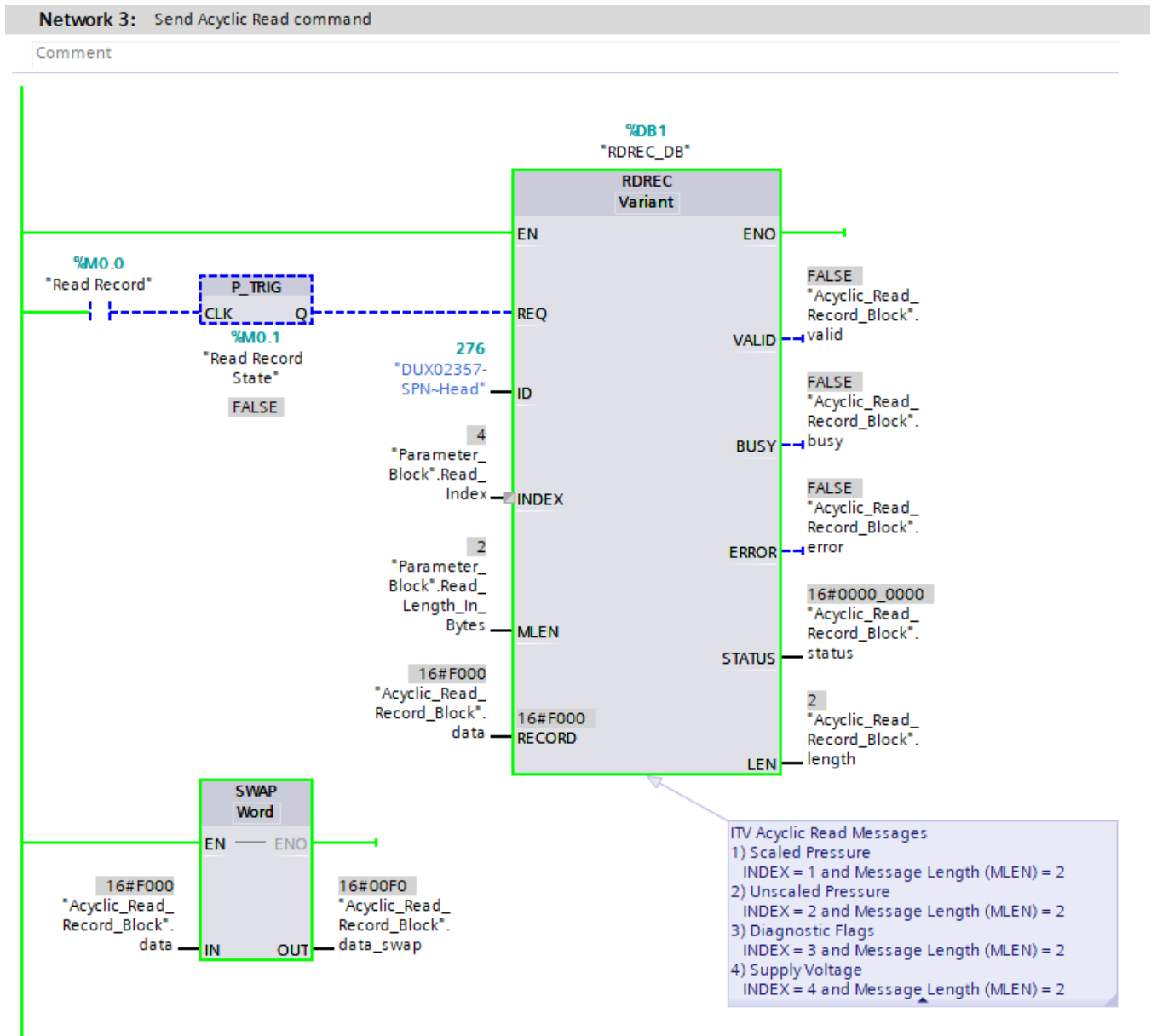
The returned data will be available in the “data” and “data\_swap” members of Acyclic\_Read\_Record\_Block. The number of bytes returned as well as the status of the operation will also be available in Acyclic\_Read\_Record\_Block. See the **Acyclic\_Read\_Record\_Block** section for additional information.

The following index values are used to read information from the ITV:

- 1) Scaled Pressure with a message length of 2

- 2) Unscaled Pressure with a message length of 2
- 3) Diagnostics Flags with a message length of 2
- 4) Supply Voltage with a message length of 2

A logic signal is provided to perform the read operation. Logic is also provided to perform the byte swap operation on the returned data. The DUX02357 Library uses the internal TIA Portal Library function RDREC to perform the actual operation.



**Figure 31 – Acyclic Read Ladder Logic**

#### 9.1.1.4. Network 4 - Send Acyclic Write command

The Network 4 software performs all acyclic write operations. The user needs to provide the following information:

- 1) ID – the ID of the ITV to communicate to.
- 2) Write Index – index value that represents the appropriate data to write (see below). The Write\_Index member of the Parameter\_Block is used to hold this value (see **Parameter\_Block** section).

- 3) Message Length – number of bytes to write (see below). The Write\_Length\_in\_Bytes member of the Parameter\_Block is used to hold this value (see **Parameter\_Block** section)
- 4) .
- 5) Message – message to be sent to the ITV. The Write\_Value member of the Parameter\_Block is used to hold this value (see **Parameter\_Block** section). The byte swap is automatically performed on the data before the message is sent.

The status of the operation will be available in Acyclic\_Write\_Record\_Block. See the **Acyclic\_Write\_Record\_Block** section for additional information.

The following index values are used to write information to the ITV:

- 1) Pressure Setpoint with a message length of 2
- 2) Application Configuration with a message length of 2
- 3) User Gain with a message length of 2
- 4) User Sensitivity with a message length of 2

A logic signal is provided to perform the write operation. Logic is also provided to perform the byte swap operation on the sent data. The DUX02357 Library uses the internal TIA Portal Library function WRREC to perform the actual operation.

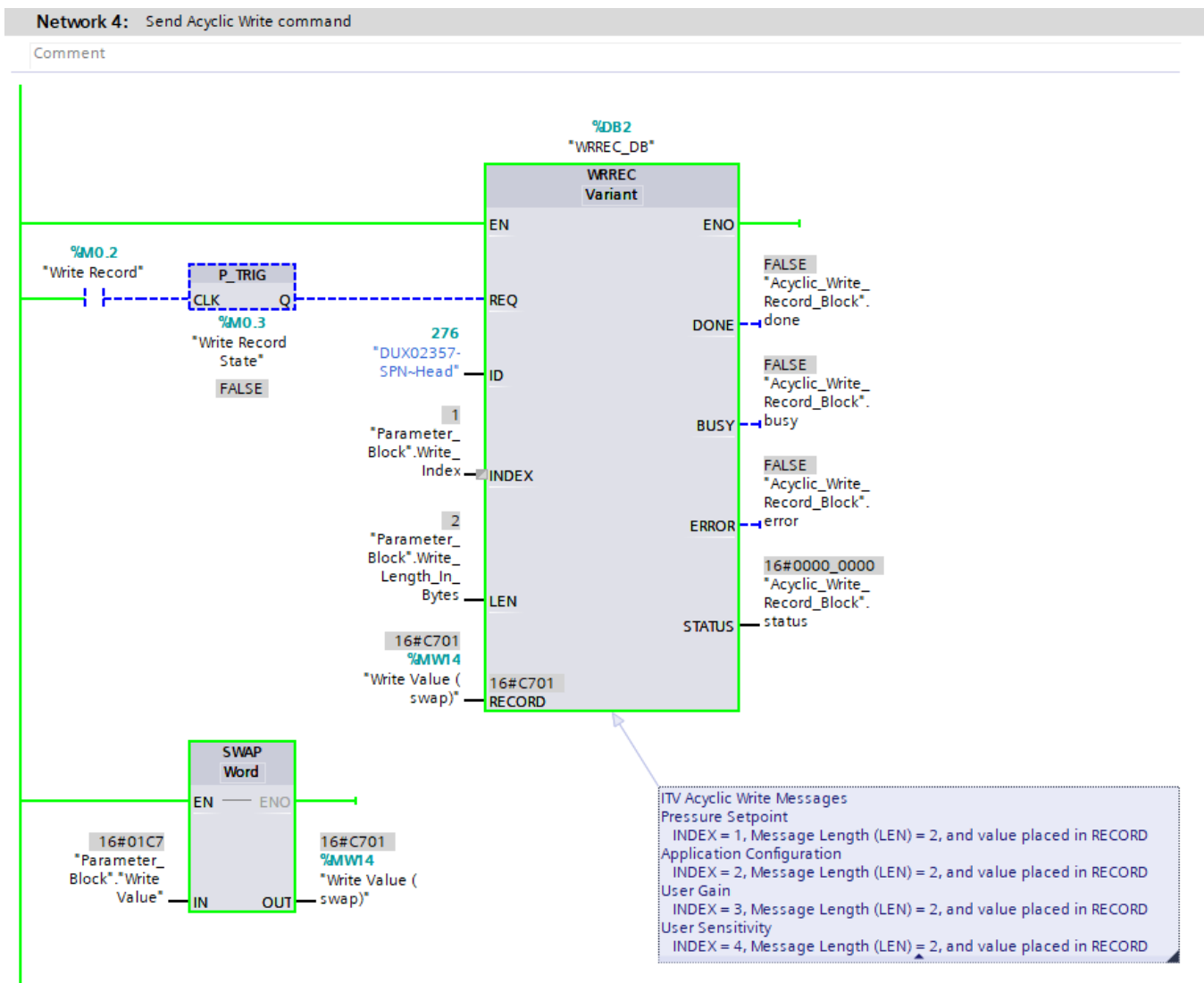


Figure 32 – Acyclic Write Ladder Logic

#### 9.1.1.5. Network 5 - Read all I&M data (I&M0, I&M1, I&M2, and I&M3)

The Network 5 software reads the Identification and Maintenance data from the ITV. A custom data block (IM\_Data\_Block) is defined to contain the address ID of the device to read from, the data returned from the ITV, and the status of the last read operation. The user needs to provide the ID (address) of the ITV to read from. A logic signal is provided to perform the I&M read operation. Please refer to the **Reading Identification and Maintenance Data** section for more information.

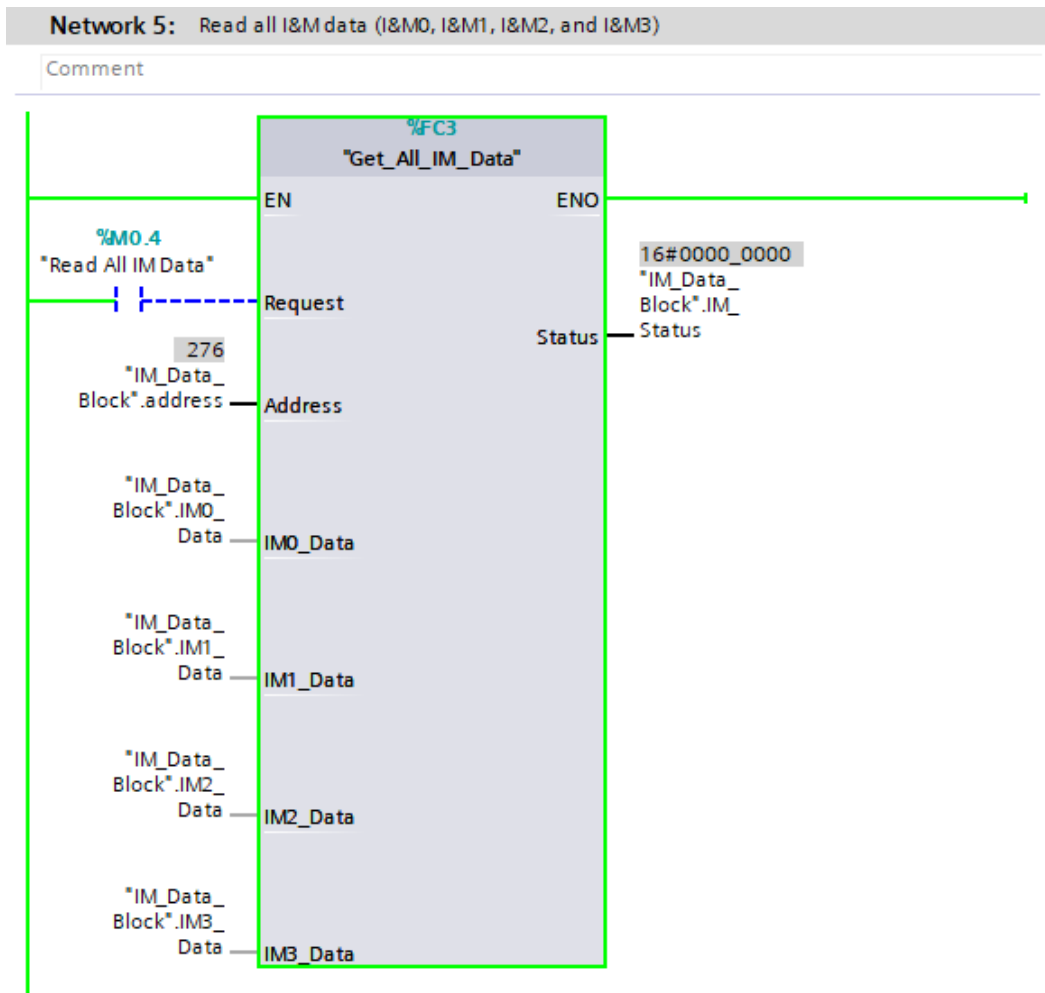


Figure 33 – Identification and Maintenance Read Ladder Logic

## 9.2. Functions

Functions are code blocks without memory but can use global data blocks to store data permanently. The following functions are developed in SCL and provide specific user programs that are used by the Organization Blocks.

### 9.2.1. Reading Identification and Maintenance Data (Get\_All\_IM\_Data)

The Get\_All\_IM\_Data function performs the actual reading of I&M0 to I&M3 data. The software checks to see if the request state has changed and is currently being requested then the software requests the I&M0 thorough I&M3 data from the ITV. After each request, if there is an error condition then the software will exit the function and return the status.

Get_All_IM_Data			
	Name	Data type	Default value
1	▼ Input		
2	Request	Bool	
3	Address	HW_IO	
4	▼ Output		
5	Status	DWord	
6	▼ InOut		
7	▶ IMO_Data	IMO_Data	
8	▶ IM1_Data	"IM1_Data"	
9	▶ IM2_Data	"IM2_Data"	
10	▶ IM3_Data	"IM3_Data"	
11	▼ Temp		
12	i	UInt	
13	Output_Length	UInt	
14	▶ Data	Array[0..59] of Byte	
15	Valid	Bool	
16	Busy	Bool	
17	Error	Bool	
18	Done	Bool	
19	Status_Word	Word	
20	Loop_Counter	UInt	
21	Req_Flag	Bool	
22	▼ Constant		
23	HEADER_LENGTH	UInt	6
24	FUNCTION_LENGTH	UInt	32
25	LOCATION_LENGTH	UInt	22
26	DATE_LENGTH	UInt	16
27	DESCRIPTOR_LENGTH	UInt	54
28	BUFFER_LENGTH	UInt	60
29	MAX_LOOPS	UInt	5000
30	IM1_INDEX	DInt	16#AFF1
31	IM2_INDEX	DInt	16#AFF2
32	IM3_INDEX	DInt	16#AFF3
33	▼ Return		
34	Get_All_IM_Data	Void	

**Figure 34 – Get\_All\_IM\_Data Parameters and Variables**

```
// Reading I&M0 will be performed with using the Get_IM_Data function and the
// IM1, IM2, and IM3 will be performed with the RDREC function. Using the
// Get_IM_Data function to read the IM1, IM2, and IM3 data using index 11,
// 12, and 13 does not seem to work. */
```

```
"R_TRIG_DB"(CLK := #Request,
             Q => #Req_Flag);
```

```
IF #Req_Flag = TRUE THEN
```

```
    // IMO
    #Done := FALSE;
    #Busy := TRUE;
    #Loop_Counter := 0;
    #Status_Word := 0;
```

```
    WHILE #Busy = TRUE AND #Loop_Counter < #MAX_LOOPS DO
```

```
        "GET_IM_DATA_DB"(LADDR := #Address,
                         IM_TYPE := 0,
```

```

        DONE => #Done,
        BUSY => #Busy,
        ERROR => #Error,
        STATUS => #Status_Word,
        DATA := #IM0_Data);

    #Loop_Counter := #Loop_Counter + 1;

END_WHILE;

#Status := WORD_TO_DWORD(#Status_Word);

IF #Error = TRUE OR #Done = FALSE OR #Loop_Counter >= #MAX_LOOPS THEN
    #Req_Flag := FALSE;
    RETURN;
END_IF;

// IM1
#Output_Length := 0;
#Valid := FALSE;
#Busy := TRUE;
#Loop_Counter := 0;

FOR #i := 0 TO (#BUFFER_LENGTH - 1) DO
    // Clear bytes
    #Data[#i] := B#20;
END_FOR;

WHILE #Busy = TRUE AND #Loop_Counter < #MAX_LOOPS DO

    "RDREC_DB_1"(REQ:=#Req_Flag,
        ID:=#Address,
        INDEX:=#IM1_INDEX,
        MLEN:=#BUFFER_LENGTH,
        VALID=>#Valid,
        BUSY=>#Busy,
        ERROR=>#Error,
        STATUS=>#Status,
        LEN=>#Output_Length,
        RECORD:=#Data);

    #Loop_Counter := #Loop_Counter + 1;

END_WHILE;

IF #Error = TRUE OR #Valid = FALSE OR #Loop_Counter >= #MAX_LOOPS THEN
    #Req_Flag := FALSE;
    RETURN;
END_IF;

IF #Valid = TRUE THEN

    // Copy over character array to appropriate character array
    FOR #i := 0 TO (#FUNCTION_LENGTH - 1) DO
        #IM1_Data.Function[#i] := #Data[#HEADER_LENGTH + #i];
    END_FOR;

    // Copy over character array to appropriate string
    Chars_TO_Strg(Chars := #Data,
        pChars := #HEADER_LENGTH,
        Cnt := #FUNCTION_LENGTH,
        Strg => #IM1_Data.Function_str);

    // Copy over character array to appropriate character array
    FOR #i := 0 TO (#LOCATION_LENGTH - 1) DO
        #IM1_Data.Location[#i] := #Data[#HEADER_LENGTH + #FUNCTION_LENGTH + #i];
    END_FOR;

    // Copy over character array to appropriate string
    Chars_TO_Strg(Chars := #Data,
        pChars := #HEADER_LENGTH + #FUNCTION_LENGTH,
        Cnt := #LOCATION_LENGTH,
        Strg => #IM1_Data.Location_str);

END_IF;

```

```

// IM2
#Output_Length := 0;
#Valid := FALSE;
#Busy := TRUE;
#Loop_Counter := 0;

FOR #i := 0 TO (#BUFFER_LENGTH - 1) DO
    // Clear bytes
    #Data[#i] := B#20;
END_FOR;

WHILE #Busy = TRUE AND #Loop_Counter < #MAX_LOOPS DO

    "RDREC_DB_2"(REQ := #Req_Flag,
        ID := #Address,
        INDEX := #IM2_INDEX,
        MLEN := #BUFFER_LENGTH,
        VALID => #Valid,
        BUSY => #Busy,
        ERROR => #Error,
        STATUS => #Status,
        LEN => #Output_Length,
        RECORD := #Data);

    #Loop_Counter := #Loop_Counter + 1;

END_WHILE;

IF #Error = TRUE OR #Valid = FALSE OR #Loop_Counter >= #MAX_LOOPS THEN
    #Req_Flag := FALSE;
    RETURN;
END_IF;

IF #Valid = TRUE THEN

    // Copy over character array to appropriate character array
    FOR #i := 0 TO (#DATE_LENGTH - 1) DO
        #IM2_Data.Installation_Date[#i] := #Data[#HEADER_LENGTH + #i];
    END_FOR;

    // Copy over character array to appropriate string
    Chars_TO_Strg(Chars := #Data,
        pChars := #HEADER_LENGTH,
        Cnt := #DATE_LENGTH,
        Strg => #IM2_Data.Installation_Date_str);

END_IF;

// IM3
#Output_Length := 0;
#Valid := FALSE;
#Busy := TRUE;
#Loop_Counter := 0;

FOR #i := 0 TO (#BUFFER_LENGTH - 1) DO
    // Clear bytes
    #Data[#i] := B#20;
END_FOR;

WHILE #Busy = TRUE AND #Loop_Counter < #MAX_LOOPS DO

    "RDREC_DB_3"(REQ := #Req_Flag,
        ID := #Address,
        INDEX := #IM3_INDEX,
        MLEN := #BUFFER_LENGTH,
        VALID => #Valid,
        BUSY => #Busy,
        ERROR => #Error,
        STATUS => #Status,
        LEN => #Output_Length,
        RECORD := #Data);

    #Loop_Counter := #Loop_Counter + 1;

```

```

END_WHILE;

IF #Error = TRUE OR #Valid = FALSE OR #Loop_Counter >= #MAX_LOOPS THEN
    #Req_Flag := FALSE;
    RETURN;
END_IF;

IF #Valid = TRUE THEN

    // Copy over character array to appropriate character array
    FOR #i := 0 TO (#DESCRIPTOR_LENGTH - 1) DO
        #IM3_Data.Descriptor[#i] := #Data[#HEADER_LENGTH + #i];
    END_FOR;

    // Copy over character array to appropriate string
    Chars_TO_Strg(Chars := #Data,
                  pChars := #HEADER_LENGTH,
                  Cnt := #DESCRIPTOR_LENGTH,
                  Strg => #IM3_Data.Descriptor_str);

END_IF;

END_IF;

```

**Figure 35 – Get\_All\_IM\_Data SCL**

## 9.2.2. Calculate Pressure Counts (Pressure\_Count\_Calculation)

The Pressure\_Count\_Calculation function takes the entered decimal pressure, converts the value to MPa, and then calculates the corresponding count value associated with the given ITV model.

Pressure_Count_Calculation				
	Name	Data type	Default value	Comment
1	▼ Input			
2	■ Enable	Bool		
3	■ Desired_Pressure	LReal		
4	■ Unit_Index	UInt		Possible units: MPa, kPa, bar, psi, or kgf/cm^2
5	■ Max_Pressure_Index	UInt		Maximum pressure for the different models are: 10 = 0.1 MPa, 30 = 0.5 MPa, 50 = 0.9 MPa, and 90 = -80 kPa
6	▼ Output			
7	■ <Add new>			
8	▼ InOut			
9	■ Pressure_Count_Calculation	UInt		
10	▼ Temp			
11	■ Desired_Pressure_Convert.	LReal		
12	▼ Constant			
13	■ <Add new>			
14	▼ Return			
15	■ Ret_Val	Void		

**Figure 36 – Pressure Count Calculation Parameters and Variables**

```

IF #Enable = 1 THEN

    CASE #Unit_Index OF
        "UNIT_MPA": // MPa
            #Desired_Pressure_Converted := #Desired_Pressure;
        "UNIT_KPA": // kPa
            #Desired_Pressure_Converted := #Desired_Pressure / "KPA_PER_MPA";
        "UNIT_BAR": // bar
            #Desired_Pressure_Converted := #Desired_Pressure / "BAR_PER_MPA";
        "UNIT_PSI": // psi
            #Desired_Pressure_Converted := #Desired_Pressure / "PSI_PER_MPA";
        "UNIT_KGF_PER_SQ_CM": // kgf/cm^2
            #Desired_Pressure_Converted := #Desired_Pressure / "KGF_PER_SQ_CM_PER_MPA";
        ELSE
            #Desired_Pressure_Converted := 0.0;
    END_CASE;

    CASE #Max_Pressure_Index OF
        "MAX_PRESSURE_0_1_MPA_INDEX": // +0.1 MPa
            #Pressure_Count_Calculation := LREAL_TO_UINT((#Desired_Pressure_Converted / 0.1) *
                UINT_TO_LREAL("MAX_PRESSURE_COUNTS"));
        "MAX_PRESSURE_0_5_MPA_INDEX": // +0.5 MPa
    END_CASE;

```



```

        #Pressure_Count_Calculation := LREAL_TO_UINT((#Desired_Pressure_Converted / 0.5) *
UINT_TO_LREAL("MAX_PRESSURE_COUNTS"));
        "MAX_PRESSURE_0_9_MPA_INDEX": // +0.9 MPa
        #Pressure_Count_Calculation := LREAL_TO_UINT((#Desired_Pressure_Converted / 0.9) *
UINT_TO_LREAL("MAX_PRESSURE_COUNTS"));
        "MAX_PRESSURE_NEG_8_0_KPA_INDEX": // -80 kPa
        #Pressure_Count_Calculation := LREAL_TO_UINT((#Desired_Pressure_Converted / -0.08) *
UINT_TO_LREAL("MAX_PRESSURE_COUNTS"));
        ELSE
        #Pressure_Count_Calculation := 0;
        END_CASE;
    END_IF;

```

**Figure 37 – Pressure Count Calculation SCL**

### 9.2.3. Convert Pressure Counts (Pressure\_Count\_Conversion)

The Pressure\_Count\_Conversion function reads the pressure counts from the ITV, converts the value to MPa that is associated with the given ITV model, and then converts the MPa value to kPa, psi, bar, and kgf/cm<sup>2</sup>.

Pressure_Count_Conversion				
	Name	Data type	Default value	Comment
1	▼ Input			
2	Pressure_Counts	UInt		
3	Max_Pressure_Index	UInt		Maximum pressure for the different models are: 10 = 0.1 MPa, 30 = 0.5 MPa, 50 = 0.9 MPa, and 90 = -0.08 kPa
4	▼ Output			
5	kPa	LReal		
6	MPa	LReal		
7	psi	LReal		
8	bar	LReal		
9	kgf_per_sq_cm	LReal		
10	▼ InOut			
11	<Add new>			
12	▼ Temp			
13	<Add new>			
14	▼ Constant			
15	<Add new>			
16	▼ Return			
17	Pressure_Count_Conversion	Void		

**Figure 38 – Pressure Count Conversion Parameters and Variables**

```

#MPa := 0.0;

CASE #Max_Pressure_Index OF
    "MAX_PRESSURE_0_1_MPA_INDEX": // +0.1 MPa
        #MPa := (UINT_TO_LREAL(#Pressure_Counts) / UINT_TO_LREAL("MAX_PRESSURE_COUNTS")) * 0.1;
    "MAX_PRESSURE_0_5_MPA_INDEX": // +0.5 MPa
        #MPa := (UINT_TO_LREAL(#Pressure_Counts) / UINT_TO_LREAL("MAX_PRESSURE_COUNTS")) * 0.5;
    "MAX_PRESSURE_0_9_MPA_INDEX": // +0.9 MPa
        #MPa := (UINT_TO_LREAL(#Pressure_Counts) / UINT_TO_LREAL("MAX_PRESSURE_COUNTS")) * 0.9;
    "MAX_PRESSURE_NEG_8_0_KPA_INDEX": // -80 kPa
        #MPa := (UINT_TO_LREAL(#Pressure_Counts) / UINT_TO_LREAL("MAX_PRESSURE_COUNTS")) * -0.08;
END_CASE;

#kPa := #MPa * "KPA_PER_MPA";
#psi := #MPa * "PSI_PER_MPA";
#bar := #MPa * "BAR_PER_MPA";
#kgf_per_sq_cm := #MPa * "KGF_PER_SQ_CM_PER_MPA";

```

**Figure 39 – Pressure Count Conversion SCL**

## 9.3. Data Blocks

Data blocks contain variable data that is used by the user program. Global data blocks store data that can be used by all blocks. The Direction column in the following sections is from the point of view of the user. Please refer to the Siemens TIA Portal documentation regarding the data types.

### 9.3.1. Acyclic\_Read\_Record\_Block

The Acyclic\_Read\_Record\_Block Data Block contains the information associated with performing the generic acyclic read operation. The data is populated from the user and RDREC\_DB. The number of bytes to read needs to be stored in “length” and the returned data can be read from either “data” or “data\_swap” depending on which endian is needed. The status of the operation can be determining by monitoring the “valid”, “busy”, “error”, and “status” parameters. Please consult Siemens documentation for additional information regarding the RDREC instruction.

Parameter	Type	Direction	Description
data	Word	Out	Returned data
data_swap	Word	Out	Returned data with bytes swapped
valid	Bool	Out	Indicator that data was received and valid (TRUE)
busy	Bool	Out	Indicator that transmission is not yet complete (TRUE)
error	Bool	Out	Indicator that an error has occurred (TRUE)
status	DWord	Out	Status information or error condition
length	UInt	In	Number of bytes to read

Table 2 – Acyclic\_Read\_Record\_Block Structure

### 9.3.2. Acyclic\_Write\_Record\_Block

The Acyclic\_Write\_Record\_Block Data Block contains the information associated with performing the generic acyclic write operation. The data is populated by WRREC\_DB. The transmitted data is currently defined in the Parameter\_Block Data Block to have one place for acyclic initialization data (see **Parameter\_Block** section). The status of the operation can be determining by monitoring the “done”, “busy”, “error”, and “status” parameters. Please consult Siemens documentation for additional information regarding the WRREC instruction.

Parameter	Type	Direction	Description
data	Word	In	Data to be sent (not used)
data_swap	Word	Out	Data to be sent with bytes swapped (not used)
done	Bool	Out	Indicator that data has been transmitted if no errors (TRUE)
busy	Bool	Out	Indicator that transmission is not yet complete (TRUE)
error	Bool	Out	Indicator that an error has occurred (TRUE)
status	DWord	Out	Status information or error condition

Table 3 – Acyclic\_Write\_Record\_Block Structure

### 9.3.3. Cyclic\_Read\_Record\_Block

The Cyclic\_Read\_Record\_Block Data Block contains the information associated with performing the generic cyclic read operation. The data is populated by using internal library MOVE instructions to move the data from internally assigned memory to the respective data block member. Associated with this data are two internal library SWAP instructions that swap the data to provide the data in both endian forms for “Read Pressure Counts (swap)” and “Read Diagnostic Flags (swap)”.

Parameter	Type	Direction	Description
pressure_counts	Word	Out	Pressure counts read from the ITV (see Operation Manual for additional information)
diagnostics_flags	Word	Out	Diagnostic Flags read from the ITV (see Operation Manual for additional information)

**Table 4 – Cyclic\_Read\_Record\_Block Structure**

#### 9.3.4. Cyclic\_Write\_Record\_Block

The Cyclic\_Write\_Record\_Block Data Block contains the information associated with performing the generic cyclic write operation. First, the data is calculated using the “Pressure\_Count\_Calculation” function (see **Calculate Pressure Counts** section). The calculated data is then moved to the data block by an internal library SWAP instruction.

Parameter	Type	Direction	Description
pressure	Word	In	Pressure counts to be sent to the ITV (see Operation manual for additional information)

**Table 5 – Cyclic\_Write\_Record\_Block Structure**

#### 9.3.5. GET\_IM\_DATA\_DB

The GET\_IM\_DATA\_DB Data Block is provided as a TIA Portal internal Data Block and Function. THE PROFINET ITV can use GET\_IM\_DATA\_DB to read all the Identification and Maintenance information of a device. However, the PROFINET ITV library only uses this Data Block and Function to obtain the I&M0 data. The function does not appear to work for the other I&M data. At the time of the development of this software, the internal library did not work when reading I&M1 through I&M3 if the IM\_TYPE value were 1, 2, and 3 OR 11, 12, and 13 as stated in the documentation. For the I&M 1 through I&M3 the RDREC\_DB internal Function is used instead. Please consult Siemens documentation for additional information.

#### 9.3.6. IM\_Data\_Block

The IM\_Data\_Block Data Block stores the Identification and Maintenance data and is populated by the Get\_All\_IM\_Data Function (see **Reading Identification and Maintenance Data** section). This data includes the I&M0 through I&M3 data.

Parameter	Type	Direction	Description
Address	HW_IO	In	The address (ID) of the PROFINET ITV
IM_Data_Raw	Array of Char	Out	Sixty (60) bytes of temporary storage that is used for each read of the individual I&M data.
IM0_Data	IM0_Data	Out	Storage for the I&M0 data and is provided by the TIA Portal (Please consult Siemens documentation and the <b>IM0_Data and IM0_Version</b> section for additional information).
IM1_Data	IM1_Data	Out	Storage for the I&M1 data (see <b>IM1_Data</b> section for additional information).
IM2_Data	IM2_Data	Out	Storage for the I&M2 data (see <b>IM2_Data</b> section for additional information).
IM3_Data	IM3_Data	Out	Storage for the I&M3 data (see <b>IM3_Data</b> section for additional information).
IM_Status	DWord	Out	Status information on the last I&M read operations. (Please consult Siemens documentation for additional information)

**Table 6 – IM\_Data\_Block Structure**

### 9.3.7. Parameter\_Block

The Parameter\_Block stores initialization values for other blocks to use. Additional information about the values corresponding to the members of the block can be found in the Operations Manual.

Parameter	Type	Start Value	Description
Max_Pressure_Index	UInt	1	The maximum pressure index value corresponds to the model of the ITV being used. This value is used in the “Pressure_Count_Conversion” and the “Pressure_Count_Calculation” functions. There are no enumerations in TIA Portal, but appropriate values can be found in the <b>User Constants</b> section of this manual. 1 = 0.1 MPa (Model 10) 2 = 0.5 MPa (Model 30) 3 = 0.9 MPa (Model 50) 4 = -80 kPa (Model 90)
Unit_Index	UInt	1	The unit index value corresponds to the units associated with the pressure value entered. This value is used in the “Pressure_Count_Calculation” function. There are no enumerations in TIA Portal, but appropriate values can be found in the <b>User Constants</b> section of this manual. 1 = MPa 2 = kPa 3 = bar 4 = psi 5 = kgf/cm <sup>2</sup>

Read_Index	UInt	1	The read index value corresponds to what information should be read from the ITV. This value is used in the acyclic read function (RDREC_DB). 1 = Scaled Pressure 2 = Unscaled Pressure 3 = Diagnostic Flags 4 = Supply Voltage
Read_Length_In_Bytes	UInt	2	Currently, all acyclic read messages have the length of 2.
Write_Index	UInt	1	The write index value corresponds to what information should be written to the ITV. This value is used in the acyclic write function (WRREC_DB). 1 = Pressure Setpoint 2 = Application Configuration 3 = User Gain 4 = User Sensitivity
Write_Length_In_Bytes	UInt	2	Currently, all acyclic write messages have the length of 2.
Write_Value	Word	0	Initialize write value to 0.

**Table 7 – Parameter\_Block Structure**

### 9.3.8. R\_TRIG\_DB

The R\_TRIG\_DB Data Blocks is provided as a TIA Portal internal Data Block and Function. The library software uses this Data Block and Function to detect a positive trigger when determining when to perform a I&M data read (see **Reading Identification and Maintenance Data** section). Please consult Siemens documentation for additional information.

### 9.3.9. RDREC\_DB, RDREC\_DB\_1, RDREC\_DB\_2, and RDREC\_DB\_3

The RDREC\_DB Data Blocks are provided as a TIA Portal internal Data Block and Function. The library software uses this Data Block and Function to perform acyclic reads from a device. The RDREC\_DB\_1, RDREC\_DB\_2, and RDREC\_DB\_3 are used to read the I&M1, I&M2, and I&M3 data respectively from the device (see **Reading Identification and Maintenance Data** section) while the RDREC\_DB is used to perform the generic acyclic read operation (see **Network 3 - Send Acyclic Read command** section). Please consult Siemens documentation for additional information concerning the operations of the RDREC instruction.

### 9.3.10. WRREC\_DB

The WRREC\_DB Data Blocks is provided as a TIA Portal internal Data Block and Function. The library software uses this Data Block and Function to perform acyclic writes to a device. The WRREC\_DB is used to perform the generic acyclic write operation (see **Network 4 - Send Acyclic Write command** section). Please consult Siemens documentation for additional information concerning the operations of the WRREC instruction.

## 9.4. Function Blocks

The only Function Blocks that are used in the library are located under the “System blocks...Program resources”. The GET\_IM\_DATA and R\_TRIG functions are used in the other blocks and are provided by the TIA Portal system library.

## 10. PLC Tags

A PLC Tag Table for the PROFINET ITV is provided to easily reference variables. Tags are the method for assigning and referencing memory locations and are simply an alphanumeric string that points to a memory location.

### 10.1. DUX02357 Tag Table and User Constants

The DUX02357 Tag Table is provided to reference all variables needed for the library and the User Constants are provided for the functions. The referenced memory locations may need to be changed if TIA Portal automatically reassigns the variables when imported or may need to be manually changed if there is a conflict with existing system components.

#### 10.1.1. Tag Table

Name	Type	Address	Description
Read Record	Bool	%M0.0	Enables the acyclic read operation.
Read Record State	Bool	%M0.1	Stores the previous Read Record state.
Write Record	Bool	%M0.2	Enables the acyclic write operation.
Write Record State	Bool	%M0.3	Stores the previous Write Record state.
Read All IM Data	Bool	%M0.4	Enables the operation to read the I&M0 through I&M3 data from the ITV.
Calculate Pressure Count	Bool	%M1.0	Enables the Pressure Count Calculation.
Calculate Pressure Count State	Bool	%M1.1	Stores the previous Calculate Pressure Count state.
Set Pressure	Bool	%M1.2	Enables the setting of the ITV pressure by writing to the cyclic pressure memory value.
Set Pressure State	Bool	%M1.3	Stores the previous Set Pressure state.
Read Pressure Counts (swap)	Word	%MW10	Cyclic pressure count with bytes swap to be more easily readable.
Read Diagnostic Flags (swap)	Word	%MW12	Cyclic diagnostic flags with bytes swap to be more easily readable.
Write Value (swap)	Word	%MW14	Acyclic write data with bytes swap to convert for proper format for ITV.
Converted_Counts	Word	%MW16	Calculated pressure counts corresponding to user input.
Desired_Pressure	Real	%MD18	User entered desired pressure.
Read Pressure in MPa	Real	%MD22	Pressure in MPa from the Pressure Count Conversion function.
Read Pressure in kPa	Real	%MD26	Pressure in kPa from the Pressure Count Conversion function.
Read Pressure in psi	Real	%MD30	Pressure in psi from the Pressure Count Conversion function.
Read Pressure in bar	Real	%MD34	Pressure in bar from the Pressure Count Conversion function.
Read Pressure in kgf/cm <sup>2</sup>	Real	%MD38	Pressure in kgf per cm <sup>2</sup> from the Pressure Count Conversion function.
Cyclic Setpoint in Counts	Word	%QW64	Pressure counts sent to the ITV.

Read Cyclic Pressure Data	Word	%IW68	Pressure counts read from the ITV.
Read Cyclic Diagnostic Data	Word	%IW70	Diagnostic flags read from the ITV.
Setpoint Error	Bool	%I70.0	Setpoint Values out of range - If Setpoint greater than 120% full scale
Clear Zero Error	Bool	%I70.1	Not going zero within a timely manner
Pressure Error	Bool	%I70.2	Output Pressure Greater the 120% full scale
Non-volatile Data Error	Bool	%I70.3	Configuration data was corrupt and needed to restore defaults
Pressure Window Error	Bool	%I70.4	Not within setpoint and sensitivity in timely manner
Initializing Status	Bool	%I70.5	Not Used
Integrity Error	Bool	%I70.6	Memory Corruption Check
Not Used	Bool	%I70.7	
System Fault	Bool	%I71.0	
Bus Fault	Bool	%I71.1	
Voltage Fault	Bool	%I71.2	Voltage out of range (Nominal is 24 VDC $\pm 10\%$ )
Miswiring Fault	Bool	%I71.3	Not Used

**Table 8 – Tag Table**

### 10.1.2. User Constants

Constant	Type	Value	Description
KPA_PER_MPA	Real	1000.0	Conversion factor from kPa to MPa.
BAR_PER_MPA	Real	10.0	Conversion factor from bar to MPa.
PSI_PER_MPA	Real	145.03773800722	Conversion factor from psi to MPa.
KGF_PER_SQ_CM_PER_MPA	Real	10.197162129779	Conversion factor from kgf/cm <sup>2</sup> to MPa.
MAX_PRESSURE_COUNTS	UInt	4095	Pressure counts that correspond to 100% pressure.
MAX_PRESSURE_0_1_MPA_INDEX	UInt	1	Index value for pressure calculation of a Model 10 with a maximum pressure of 0.1 MPa.
MAX_PRESSURE_0_5_MPA_INDEX	UInt	2	Index value for pressure calculation of a Model 30 with a maximum pressure of 0.5 MPa.
MAX_PRESSURE_0_9_MPA_INDEX	UInt	3	Index value for pressure calculation of a Model 50 with a maximum pressure of 0.1 MPa.

MAX_PRESSURE_NEG_8_0_KPA_INDEX	UInt	4	Index value for pressure calculation of a Model 90 with a vacuum of -80 kPa.
UNIT_MPA	UInt	1	Index value for pressure units are entered in MPa.
UNIT_KPA	UInt	2	Index value for pressure units are entered in kPa.
UNIT_BAR	UInt	3	Index value for pressure units are entered in bar.
UNIT_PSI	UInt	4	Index value for pressure units are entered in psi.
UNIT_KGF_PER_SQ_CM	UInt	5	Index value for pressure units are entered in kgf/cm <sup>2</sup> .

**Table 9 – User Constants**

## 11. PLC Data Types

Several PLC Data Types are defined for parsing of the I&M data. The IM0\_Data and IM0\_Version Data Type are provided by the TIA Portal while the IM1\_Data, IM2\_Data, and IM3\_Data are provided by the DUX02357 Library.

### 11.1. IM0\_Data and IM0\_Version

Parameter		Type	Bytes	Description
Manufacturer_ID		UInt	2	Manufacturer ID
Order_ID		String	20	Order number
Serial_Number		String	16	Serial number
Hardware_Revision		UInt	2	Hardware revision
Software_Revision		IM0_Version	4	Firmware revision
	Type	Char	1	
	Functional	USInt	1	
	Bugfix	USInt	1	
	Internal	USInt	1	
Revision_Counter		UInt	2	Revision counter
Profile_ID		UInt	2	Profile
Profile_Specific_Type		UInt	2	Device class
IM_Version		Word	2	I&M version
I&M_Supported		Word	2	I&M data (I&M0-I&M4) supported at the device end

**Table 10 – IM0\_Data and IM0\_Version Structures**



## 11.2. IM1\_Data

Parameter	Type	Bytes	Description
Function	Char Array	32	Unique identification of a device function or task within a plant.
Function_str	String	Varying	String representation of the Function parameter.
Location	Char Array	22	Unique identification of a device location within a plant.
Location_str	String	Varying	String representation of the Location parameter.

**Table 11 – IM1\_Data Structure**

## 11.3. IM2\_Data

Parameter	Type	Bytes	Description
Installation_Date	Char Array	16	Date of installation or commissioning of the device. The format of the data is “YYYY-MM-DD HH:MM”.
Installation_Date_str	String	Varying	String representation of the Installation_Date parameter.

**Table 12 – IM2\_Data Structure**

## 11.4. IM3\_Data

Parameter	Type	Bytes	Description
Descriptor	Char Array	54	Stores additional information and annotation concerning the device.
Descriptor_str	String	Varying	String representation of the Descriptor parameter.

**Table 13 – IM3\_Data Structure**

## 12. Watch and Force Tables

The DUX02357 Library contains both a predefined Watch Table as well as a Force Table. The Watch Table allows the TIA Portal user to monitor memory locations or predefined Tags. The included watch table contains most of the predefined Tags as well as the Parameter\_Block Data Block. The Force Table allows the user to change data for a given memory location or Tag. Included in the Force Table is the cyclic pressure setpoint.

### 12.1. DUX02357 Watch Table

Name	Address	Display Format	Description
Read Record	%M0.0	Bool	Enables the acyclic read operation.
Read Record State	%M0.1	Bool	Stores the previous Read Record state.
Write Record	%M0.2	Bool	Enables the acyclic write operation.
Write Record State	%M0.3	Bool	Stores the previous Write Record state.

Read All IM Data	%M0.4	Bool	Enables the operation to read the I&M0 through I&M3 data from the ITV.
Calculate Pressure Count	%M1.0	Bool	Enables the Pressure Count Calculation.
Calculate Pressure Count State	%M1.1	Bool	Stores the previous Calculate Pressure Count state.
Set Pressure	%M1.2	Bool	Enables the setting of the ITV pressure by writing to the cyclic pressure memory value.
Set Pressure State	%M1.3	Bool	Stores the previous Set Pressure state.
Read Pressure Counts (swap)	%MW10	Hex	Cyclic pressure count with bytes swap to be more easily readable.
Read Diagnostic Flags (swap)	%MW12	Hex	Cyclic diagnostic flags with bytes swap to be more easily readable.
Write Value (swap)	%MW14	Hex	Acyclic write data with bytes swap to convert for proper format for ITV.
Converted_Counts	%MW16	Hex	Calculated pressure counts corresponding to user input.
Desired_Pressure	%MD18	Floating-point number	User entered desired pressure.
Read Pressure in MPa	%MD22	Floating-point number	Pressure in MPa from the Pressure Count Conversion function.
Read Pressure in kPa	%MD26	Floating-point number	Pressure in kPa from the Pressure Count Conversion function.
Read Pressure in psi	%MD30	Floating-point number	Pressure in psi from the Pressure Count Conversion function.
Read Pressure in bar	%MD34	Floating-point number	Pressure in bar from the Pressure Count Conversion function.
Read Pressure in kgf/cm <sup>2</sup>	%MD38	Floating-point number	Pressure in kgf/cm <sup>2</sup> from the Pressure Count Conversion function.
Cyclic Setpoint in Counts	%QW64	Hex	Pressure counts sent to the ITV.
Read Cyclic Pressure Data	%IW68	Hex	Pressure counts read from the ITV.
Read Cyclic Diagnostic Data	%IW70	Hex	Diagnostic flags read from the ITV.
Setpoint Error	%I70.0	Bool	Setpoint Values out of range - If Setpoint greater than 120% full scale.
Clear Zero Error	%I70.1	Bool	Not going zero within a timely manner.
Pressure Error	%I70.2	Bool	Output Pressure Greater the 120% full scale.
Non-volatile Data Error	%I70.3	Bool	Configuration data was corrupt and needed to restore defaults.
Pressure Window Error	%I70.4	Bool	Not within setpoint and sensitivity in timely manner.
Initializing Status	%I70.5	Bool	Not Used.

Integrity Error	%I70.6	Bool	Memory Corruption Check.
Not Used	%I70.7	Bool	
System Fault	%I71.0	Bool	
Bus Fault	%I71.1	Bool	
Voltage Fault	%I71.2	Bool	Voltage out of range (Nominal is 24 VDC $\pm 10\%$ ).
Miswiring Fault	%I71.3	Bool	Not Used.
Parameter_Block			See <b>Parameter_Block</b> section.
Max_Pressure_Index		DEC	
Read_Index		DEC	
Read_Length_In_Bytes		DEC	
Unit_Index		DEC	
Write Value		Hex	
Write_Index		DEC	
Write_Length_In_Bytes		DEC	

**Table 14 – Watch Table**

## 12.2. Force Table

Name	Address	Display Format	Description
Cyclic Setpoint in Counts	%QW64:P	Hex	Pressure counts sent to the ITV.

**Table 15 – Force Table**

SMC Corporation of America

10100 SMC Boulevard

Noblesville, IN 46060

Phone: 1-800-SMC-SMC1 (1-800-762-7621)

URL: [www.smcusa.com](http://www.smcusa.com)