Zebra RFID – PROFINET

User Guide

ZEBRA

MN-004128-01EN Rev. A

Terms of Use

Proprietary Statement

This manual contains proprietary information of Zebra Technologies Corporation and its subsidiaries ("Zebra Technologies"). It is intended solely for the information and use of parties operating and maintaining the equipment described herein. Such proprietary information may not be used, reproduced, or disclosed to any other parties for any other purpose without the express, written permission of Zebra Technologies.

Product Improvements

Continuous improvement of products is a policy of Zebra Technologies. All specifications and designs are subject to change without notice.

Liability Disclaimer

Zebra Technologies takes steps to ensure that its published Engineering specifications and manuals are correct; however, errors do occur. Zebra Technologies reserves the right to correct any such errors and disclaims liability resulting therefrom.

Limitation of Liability

In no event shall Zebra Technologies or anyone else involved in the creation, production, or delivery of the accompanying product (including hardware and software) be liable for any damages whatsoever (including, without limitation, consequential damages including loss of business profits, business interruption, or loss of business information) arising out of the use of, the results of use of, or inability to use such product, even if Zebra Technologies has been advised of the possibility of such damages. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to you.

Revision History

Change	Date	Description
MN-004128-01EN Rev. A	March 2021	Initial Release.

Contents

List of Figures **6**

List of Tables **8**

About This Guide **10**
Introduction 10
Chapter Descriptions 10
Related Documents and Software 10
Notational Conventions 11

FX9600 Reader Configuration **12**
Update Reader Software 12
Setting Up PROFINET 12
Install PROFINET License 12
Select the Required Virtual Device Access Point (DAP) for the PROFINET App 13
Install and Run the PROFINET App 14
Configure the Device Name Using the PROFINET Commander Tool 15

Siemens TIA Portal Configuration for the FX9600 RFID Reader **16**
Configuring the Siemens TIA Portal 16
Import the General Station Description (GSDML) Associated with the FX9600 RFID Reader 16
Browse the Catalog to Find the FX9600 Reader 16
Configure Device Name 17
Configure IP and Subnet Mask 17
Configure IO Cycle Time 19
Configure Various Modules on the FX9600 Reader 19
Create User Defined Data Types (UDT) 20

Contents

Module Parameters Configuration 22
Accessing the Module Parameters Configuration Page 22
Antenna Configuration Using Module Parameters 24
Pre-Filter Configuration using Module Parameters 24
Trigger Configurations Using Module Parameter 25

Perform RFID Inventory Operation 27
Create Data Blocks and Variables to Use with Inventory Operation 27
Inventory Command Ladder Diagram 28
Inventory Response Ladder Diagram 28
Simple Ladder Logic to Start and Stop Inventory 30
Inventory Command Parameter Structure 31
Inventory Response Parameter Structure 31
Standard EPC Parameter Structure 31
Extended Standard EPC Parameter Structure 32
Long EPC Parameter Structure 33
Extended Long EPC Parameter Structure 34
Sample Inventory Response Output 36

Perform RFID Access Operation 37
Create Data Blocks and Variables to Use With Access Operation 37
Access Command Ladder Diagram 38
Access Response Ladder Diagram 38
Simple Ladder Diagram to Perform Access Operation 39
Access Command Parameter Structure 39
Access Response Parameter Structure 41
Sample Access Response Output 42

RFID SET/GET Parameter 43
RFID Parameters 43
RFID Capabilities 44
Set/Get Parameters 44
Profile 44
Antenna Config 45
Delete Pre-Filter 45
Pre-Filter Config 46
Post-Filter Config 46
Trigger Config 47
GPIO Config 48
Event Report 50

Module Parameter Error Status 51
Sample Ladder Diagram to Set/Get RFID Parameters 51
Ladder Diagram to Read Parameter from the FX9600 Reader 52
Ladder Diagram to Write Parameter to PROFINET Devices 53

Alarms, Event Mask and Error Codes 55
Alarms 55
Antenna Disconnect Alarm 55
RFID Reader Exception 55
RFID Disconnect 55
Event Mask 55
RFID Error Codes 56

Appendix 59
Pre-Filter Settings Example 59
Report Tags That Match Pattern 59
Ignore Tags That Match Pattern 60
Reading Only Unique EPC Tags 61
Memory Estimation for Cyclic Data Exchange 62

List of Figures

Figure 1: Reader Web-Console 12
Figure 2: PROFINET License Version 13
Figure 3: Instantiate DAP 17
Figure 4: Device Name 17
Figure 5: IO Cycle 19
Figure 6: Standard EPC Response Modules 19
Figure 7: Adding a New External File 20
Figure 8: Creating UDTs 20
Figure 9: Module Parameter Configuration Page 22
Figure 10: Access Command Module 23
Figure 11: Antenna Configuration 24
Figure 12: Pre-Filter Configuration 25
Figure 13: Antenna 2 25
Figure 14: Trigger Configurations 26
Figure 15: DPWR_DAT Instruction - Data Sent from PLC to the FX9600 Reader 28
Figure 16: DPRD_DAT Instruction - Data Received from the FX9600 Reader to the PLC 29
Figure 17: Ladder Networks 4 and 5 - Receive Tag Data from Standard EPC Module_2 of FX9600 30
Figure 18: Ladder Logic to Start and Stop Inventory 30
Figure 19: Example Tag Inventory Response - Watch Table TIA Portal 36
Figure 20: Data_block_2 37
Figure 21: DPWR_DAT Iinstruction 38
Figure 22: Data_block_2.ACCRSP and Data_block_2.ACCRSP.Status.SeqNo 38
Figure 23: Sample Ladder Diagram 39
Figure 24: Sample Access Response (Read Operation) Output -

List of Figures

Watch Table TIA Portal 42
Figure 25: Antenna Config Parameter 52
Figure 26: Ladder Diagram Example - Read Parameter from the FX9600 Reader 53
Figure 27: Ladder Diagram Example - Write Parameter from the FX9600 Reader 54
Figure 28: Memory Estimate Example 62

List of Tables

Table 1: Inventory Command Parameter Structure 31
Table 2: Standard EPC Parameter Structure 32
Table 3: Extended Standard EPC Parameter Structure 33
Table 4: Long Inventory Response 34
Table 5: Extended Long EPC Parameter Structure 35
Table 6: Access Command Parameter Structure 40
Table 7: Access Response Parameter Structure 41
Table 8: RFID Parameters 43
Table 9: Reader Capability 44
Table 10: Set Active Profile 44
Table 11: Get Active Profile Name 44
Table 12: Get Profile Name 45
Table 13: Antenna Config 45
Table 14: Delete Pre-Filter 45
Table 15: Pre-Filter Config 46
Table 16: Post-Filter Config 46
Table 17: Trigger Config 47
Table 18: GPIO Config 48
Table 19: Event Report 50
Table 20: Module Parameter Error Status 51
Table 21: Event Mask Descriptions 55
Table 22: RFID Error Codes 56
Table 23: Antenna Config Parameters - Report Tags 59
Table 24: Pre-Filter Config Parameters - Report Tags 59
Table 25: Antenna Config Parameters - Ignore Tags 60

List of Tables

Table 26: Pre-Filter Config Parameters - Report Tags 60
Table 27: Trigger Config - Read Only Unique EPC Tags 61
Table 28: Module and Sub-module Sizes 62

About This Guide

Introduction

This guide describes how to install the PROFINET app on the FX Series RFID Reader and how to configure it using Siemens TIA Portal to perform RFID operations.

IMPORTANT: If you have a problem with your equipment, contact Zebra Global Customer Support for your region. Contact information is available at: www.zebra.com/support.

Chapter Descriptions

Topics covered in this guide are as follows:

- FX9600 Reader Configuration provides steps to install and run the PROFINET app.
- Siemens TIA Portal Configuration for the FX9600 RFID Reader explains steps to configure the device to communicate with PLC.
- Module Parameters Configuration explains steps to set basic RFID configuration parameters at the time of connection setup with PLC.
- Perform RFID Inventory Operation includes steps to read RFID tag EPCs.
- Perform RFID Access Operation explains steps to read or write various memory banks of the RFID tag.
- RFID SET/GET Parameter explains how to read or write various RFID configuration parameters like antenna configuration, etc.
- Alarms, Event Mask and Error Codes lists details on various events and error codes specific to the RFID operation.
- Appendix includes Pre-Filter settings examples.

Related Documents and Software

The following documents provide more information.

- FX Series RFID Fixed Reader Integration Guide
- FX Series RFID Fixed Reader FX Connect Licensing Management User Guide
- RFID Reader Software Interface Control Guide

For the latest version of this guide and all guides, go to www.zebra.com/support.

Notational Conventions

The following conventions are used in this document:

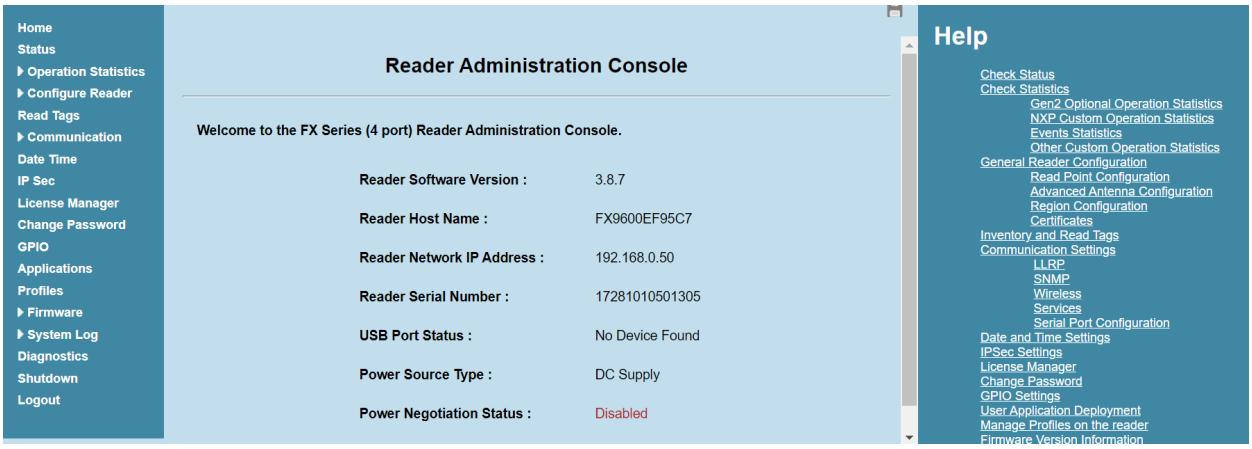
- **Bold** text is used to highlight the following:
  - Dialog box, window and screen names
  - Drop-down list and list box names
  - Check box and radio button names
  - Icons on a screen
  - Key names on a keypad
  - Button names on a screen.
- Bullets (•) indicate:
  - Action items
  - Lists of alternatives
  - Lists of required steps that are not necessarily sequential.
- Sequential lists (such as those that describe step-by-step procedures) appear as numbered lists.

FX9600 Reader Configuration

Update Reader Software

Update the reader software to use the PROFINET functionality. The FX9600 Reader must run firmware version 3.8.7, or later. Verify the current running version from the FX9600 Reader web-console shown in Figure 1.

Figure 1 Reader Web-Console



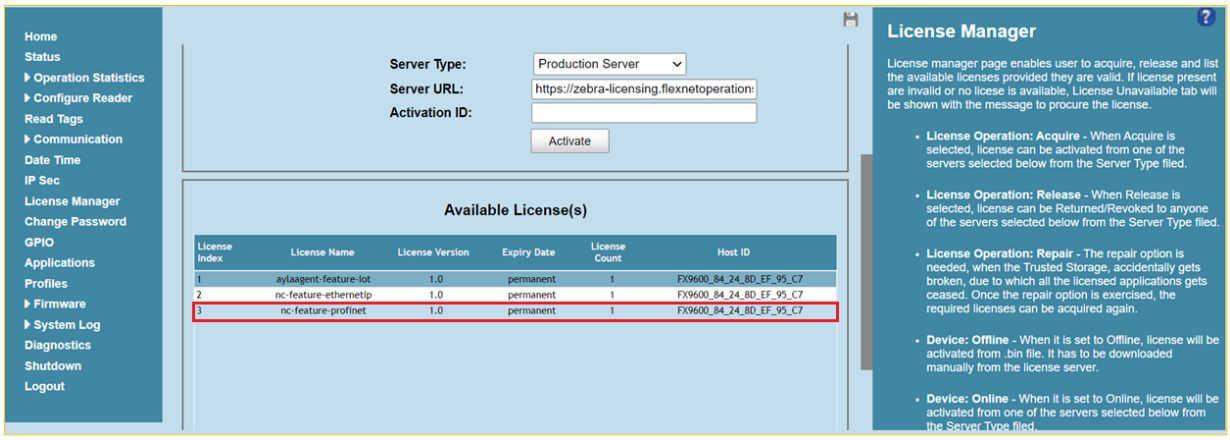
If the reader is running an older software version, upgrade the reader with new version. Download the FX Series RFID Fixed Reader Integration Guide and follow the instructions in the section Firmware Upgrade.

Setting Up PROFINET

Install PROFINET License

Install the PROFINET license from the FX9600 Reader web-console. Follow the instructions in the FX Series RFID Fixed Reader FX Connect Licensing Management User Guide to request, obtain, and install the license for Zebra FX Series readers. When the license is installed, the licensing information can be viewed from the web-console shown in Figure 2.

Figure 2 PROFINET License Version

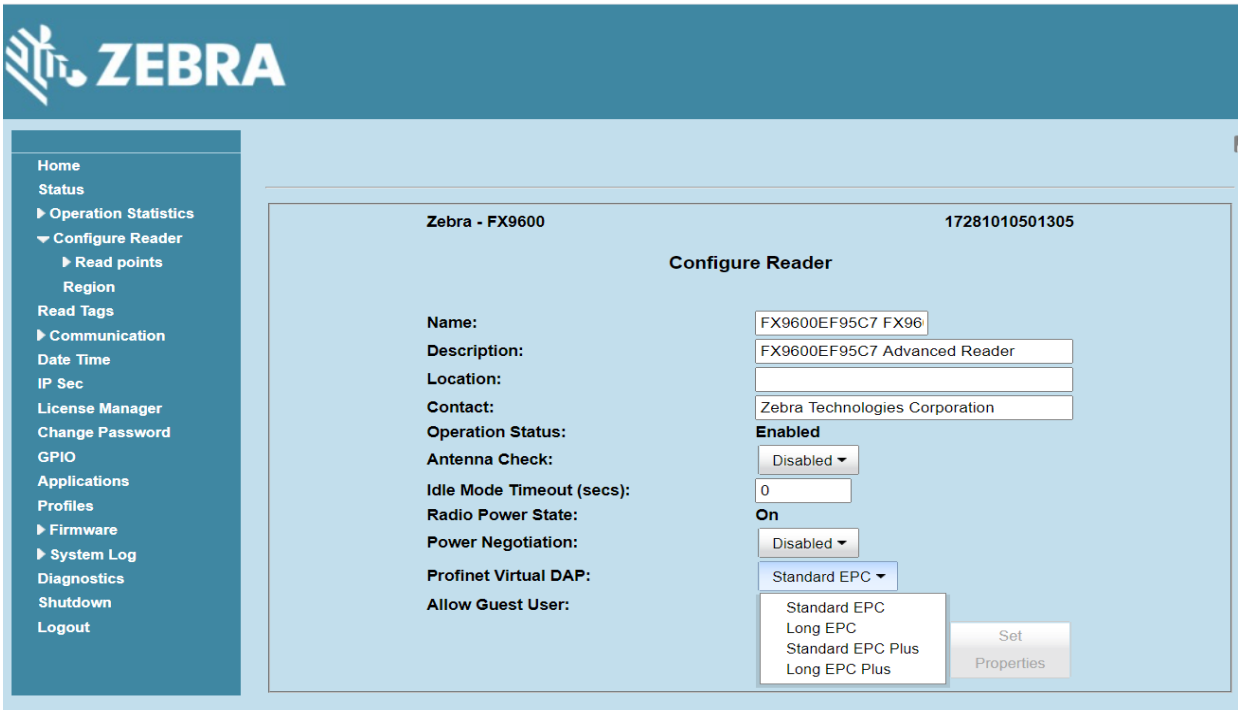


Select the Required Virtual Device Access Point (DAP) for the PROFINET App

PROFINET on FX9600 Readers supports four virtual DAPs:

- Standard EPC - Reads 12-byte EPCs.
- Standard EPC Plus - Reads 12-byte EPCs with extended tag information.
- Long EPC - Reads 64-byte long EPCs.
- Long EPC Plus - Reads 64-byte long EPCs with extended tag information.

1. Select the required virtual DAP, then click Set Properties.



2. Reboot the device to complete the selection/update.
3. Navigate to the Shutdown page to reboot the device.

Install and Run the PROFINET App

The PROFINET app can be installed in one of two ways.

It can be installed along with the license by enabling the Install and Run application check-box. Note that with this method, the PROFINET app does not start up automatically.

To run the PROFINET app:

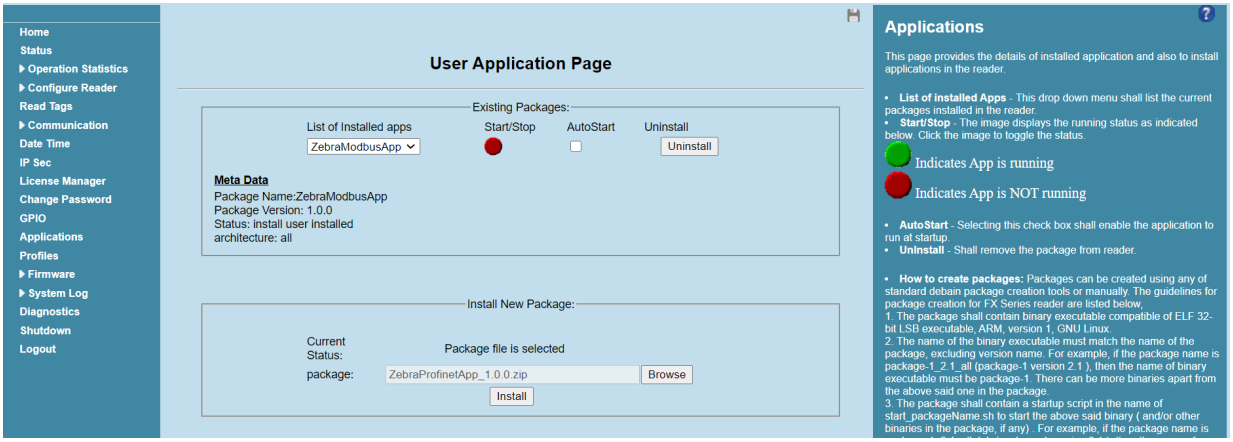
1. Navigate to the User Application page.
2. From the List of Installed apps drop-down menu, select PROFINET app.
3. Select AutoStart to enable the PROFINET app to start up automatically during power cycle on the reader.
4. To run the app, click the red Start/Stop button. The button turns green to show the app status as running.

or

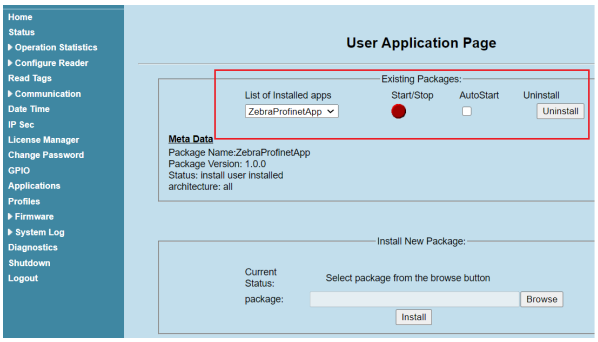
It can be installed by using the web-console to install the app zip package.

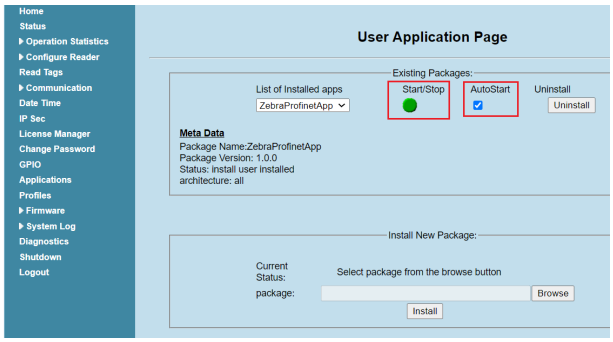
To install the app as a Debian package:

1. Navigate to the User Application page and browse for the downloaded application zip package.
2. Click Install. When the app completes installation, it is available in the List of Installed apps drop-down list under Existing Packages.



3. Select AutoStart to enable the PROFINET app on the reader to start up automatically during the power cycle.
4. To run the app, click the red Start/Stop button. The button turns green to show the app status as running.

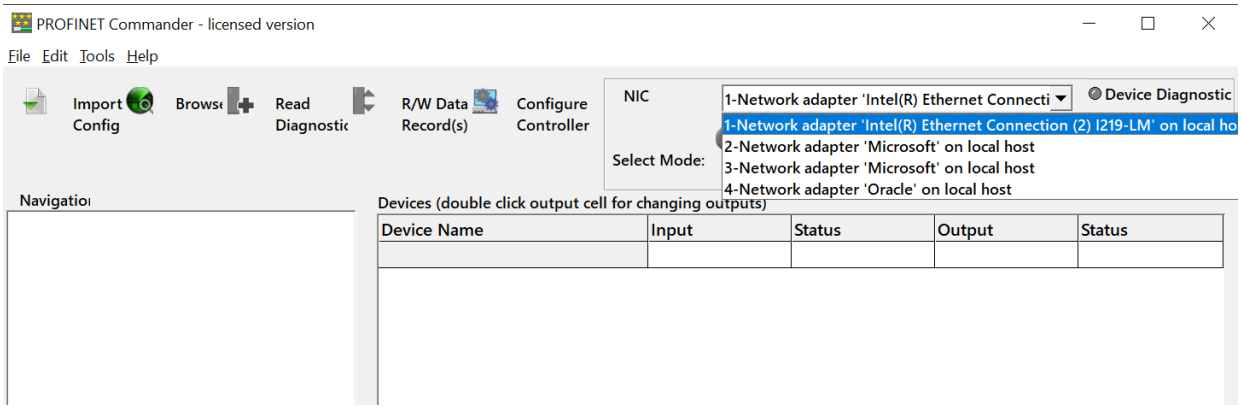




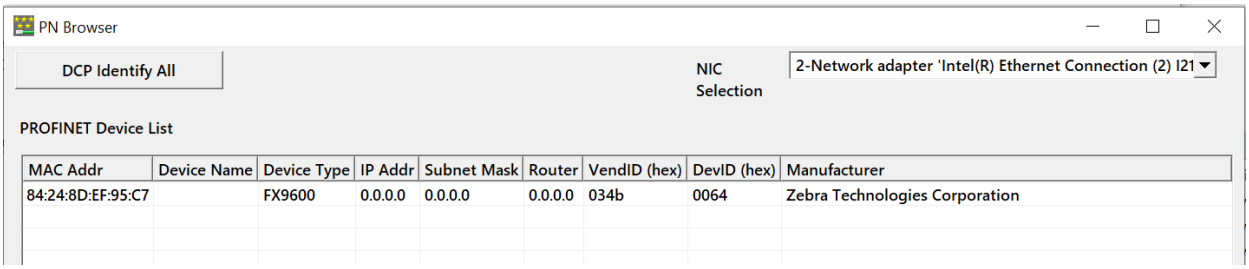
NOTE: The web-console session logs out upon app start-up. The reader is not accessible until the Name and IP are set using the PROFINET Commander, or similar tools.

Configure the Device Name Using the PROFINET Commander Tool

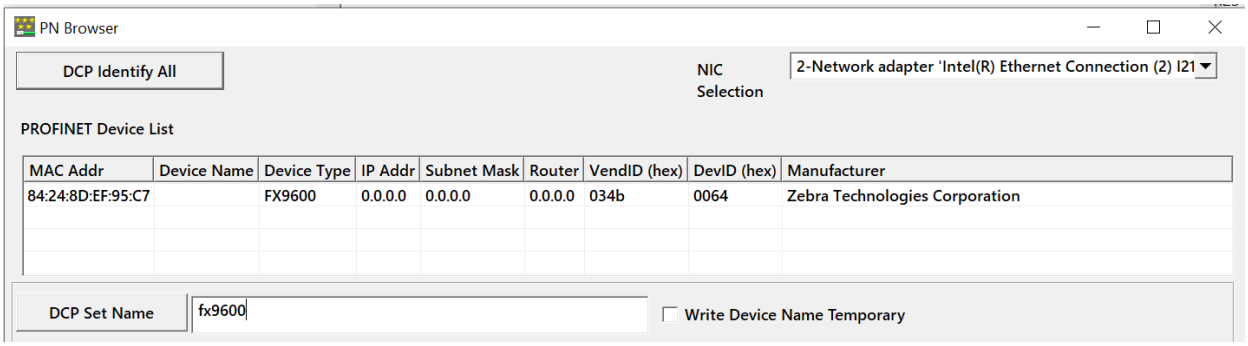
1. Download the free PROFINET Commander version from https://PROFINETcommander.com/.
2. Ensure the FX9600 Reader running the PROFINET app is connected to the same network as the PC running the PROFINET Commander tool.
3. Choose the appropriate NIC and select Browse.



4. Select DCP Identify All. The FX9600 Reader is detected as shown.



5. Enter the Name in the DCP Set Name text box and click DCP Set Name.



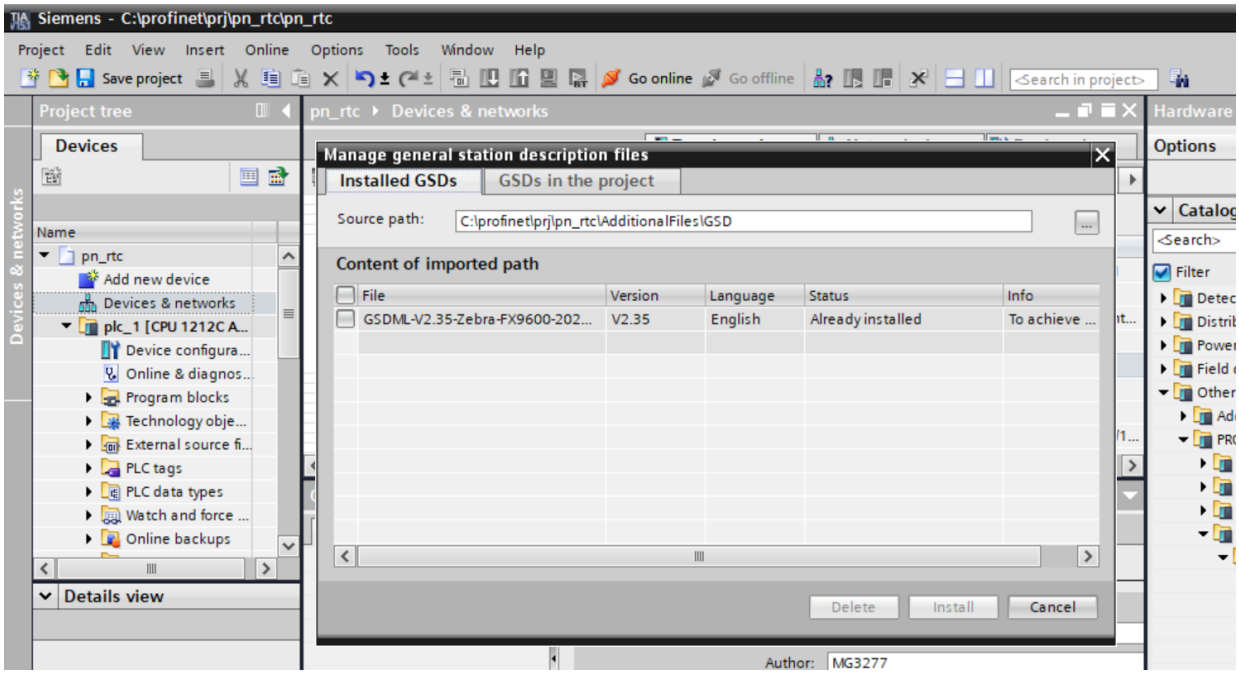
6. The FX9600 is now ready to be connected to the PROFINET network containing the PLC/PROFINET Controller. PLC/PROFINET Controller is able to set the IP address using the device name.

Siemens TIA Portal Configuration for the FX9600 RFID Reader

Configuring the Siemens TIA Portal

Import the General Station Description (GSDML) Associated with the FX9600 RFID Reader

1. Open the TIA portal (V15.1 or V16).
2. From the options menu, select Manage General Station Description files (GSD).
3. Browse and select the GSDML-V2.35-Zebra-FX9600-20201215.xml.



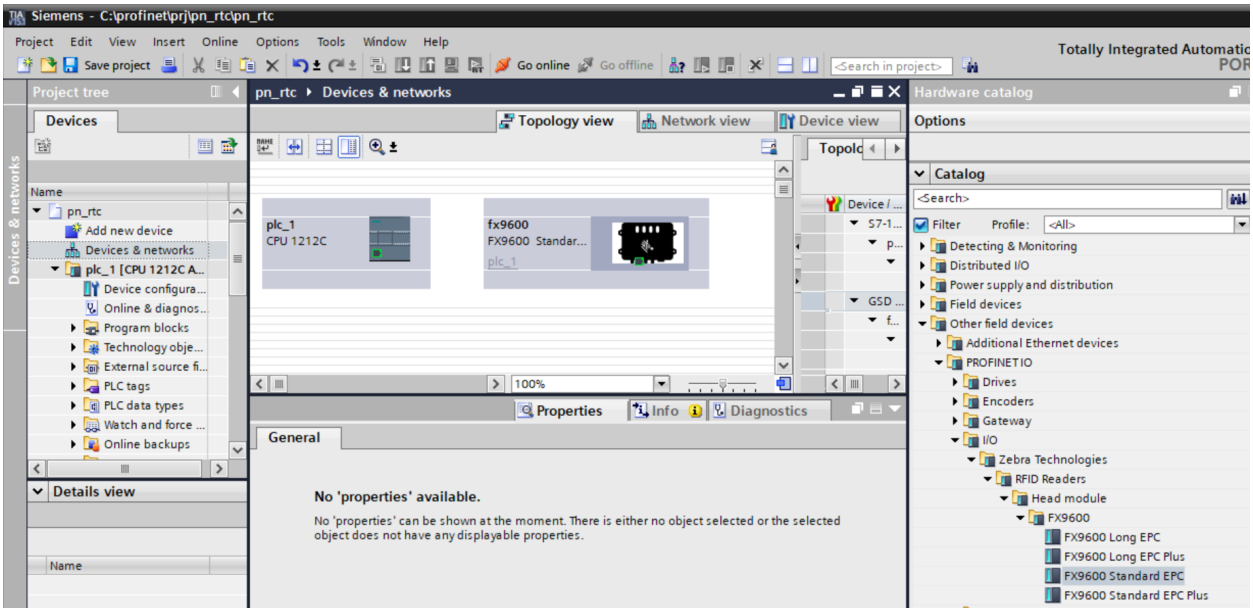
4. Click Install.

Browse the Catalog to Find the FX9600 Reader

Select the same DAP set on the FX9600 Reader. See Select the Required Virtual Device Access Point (DAP) for the PROFINET App on page 13.

NOTE: The FX9600 Reader supports four virtual DAPs and any of them can be used at any point. However, if the virtual DAP selected on the FX9600 Reader is different from the one selected in the TIA portal, the connection cannot be established and a Hardware Mismatch error is reported by the PLC.

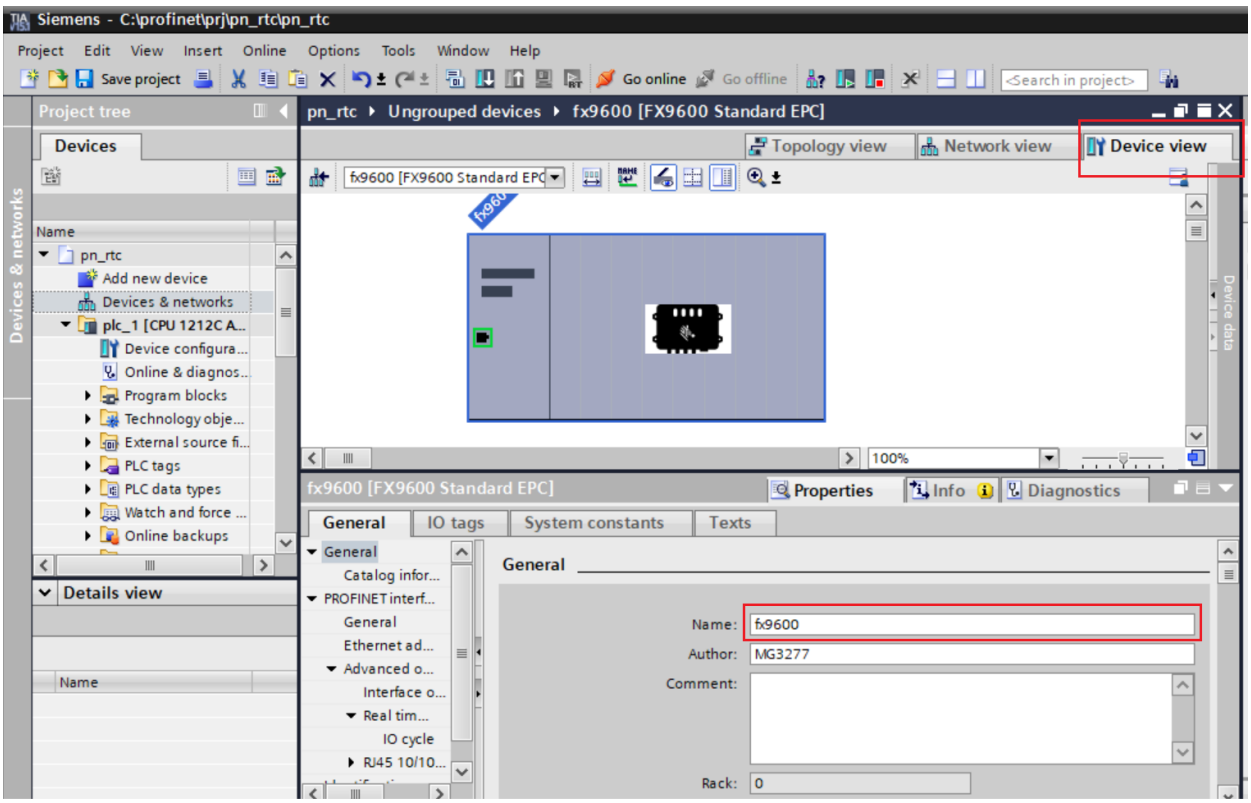
Figure 3 Instantiate DAP



Configure Device Name

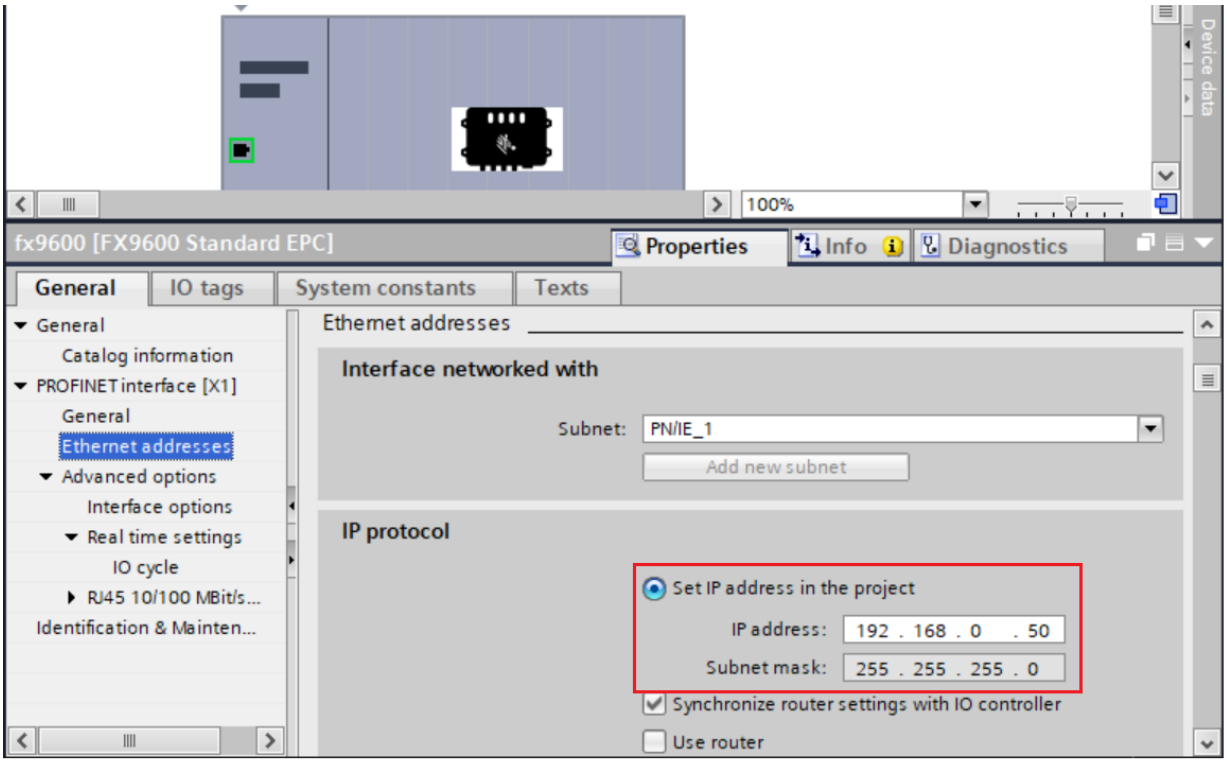
Enter the same device Name set in Configure Device Name.

Figure 4 Device Name

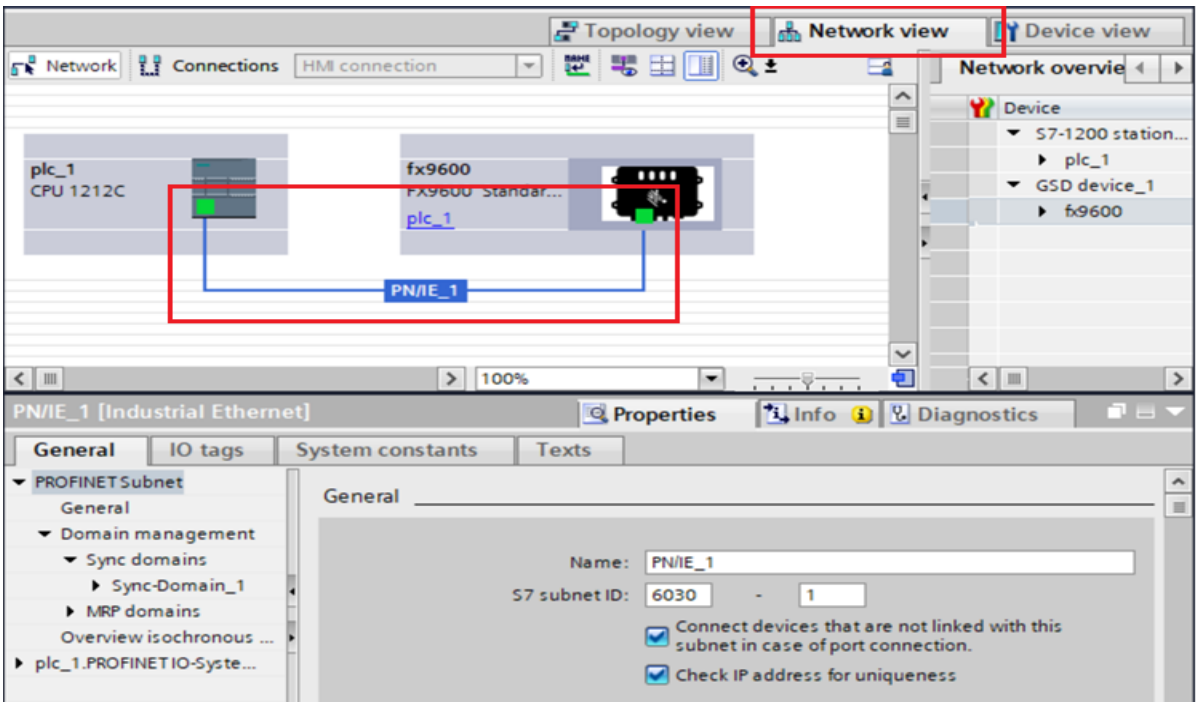


Configure IP and Subnet Mask

1. Assign the IP and Subnet Mask, as necessary, to enable communication with the PLC.



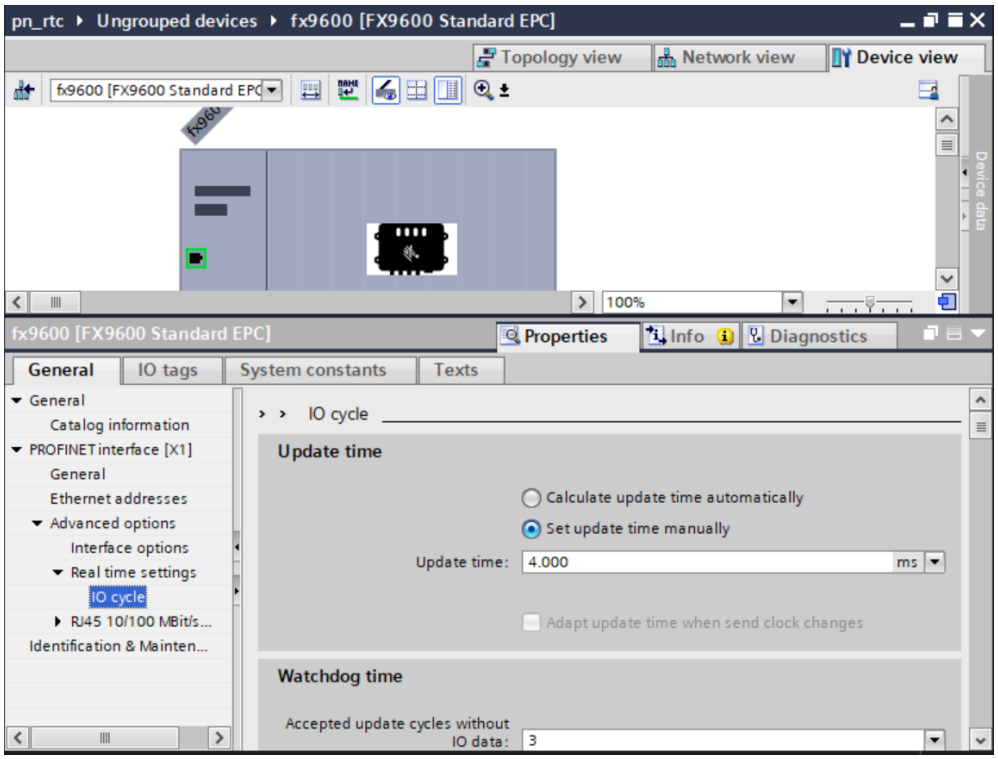
2. Connect the PLC and FX9600 by drawing the line between them in Network view.



Configure IO Cycle Time

Choose IO cycle time. 4 ms is the fastest rate supported when using the FX9600 for tag Inventory operations. If using the FX9600 Reader for Access operations, it is recommended to use 32 ms, or a lower rate, as Access operations tend to be slower compared to Inventory operations.

Figure 5 IO Cycle



Configure Various Modules on the FX9600 Reader

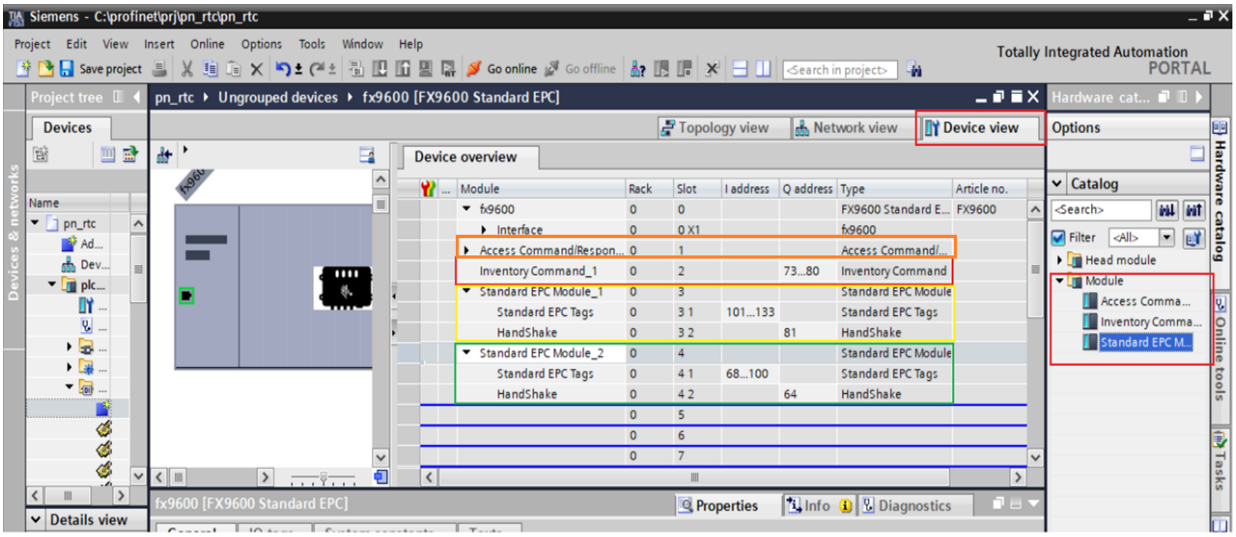
Access Command/Response module can only be configured at slot 1. This has 2 submodules one for command and one for response. These are used for the Memory Access operation on the RFID tag.

Inventory Command module can only be configured at slot 2. This module is used to start/stop the tag Inventory operation.

Up to 10 Standard EPC Response modules can be configured between slots 3 and 12. Figure 6 shows two Standard EPC Response modules configured at slot 3 and 4, respectively.

Each Response module can fetch 1 EPC tag in each IO cycle from the FX9600 Reader to PLC. Depending on the tag Read Rate required and memory availability on the PLC, configure as many Standard Response modules as needed.

Figure 6 Standard EPC Response Modules



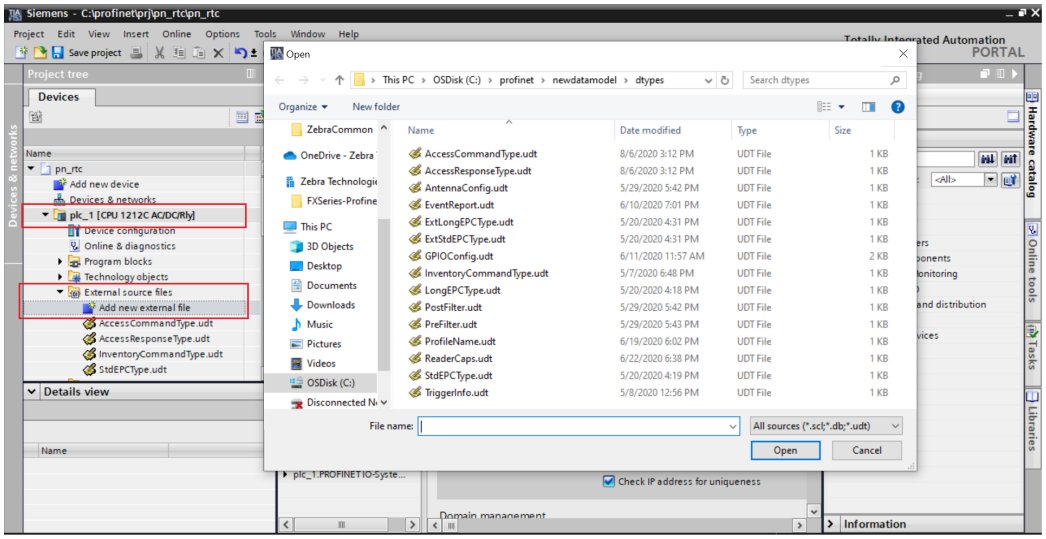
NOTE: With IO cycle time set to 4 msec, it is possible to fetch a maximum of 62 tags each second. If more than 62 tags need to be read each second consider another Inventory Response module. A maximum of 10 Response modules can be configured and more than 600 tags can be read each second.

In addition, whenever the AB_FLIP option is selected for Target Parameter in Antenna Config, it is required to have several Inventory Response modules. This is because AB_FLIP configures the radio to read the RFID tags in both state A and state B. This results in a very high tag read rate. If the number of Response modules are inadequate, then a buffer overflow occurs on the FX9600 Reader as it can only hold a certain amount of tags in its memory for the PLC to read. When buffer overflow occurs, the oldest tag info is discarded on the FX9600. This may result in loss of valuable data.

Create User Defined Data Types (UDT)

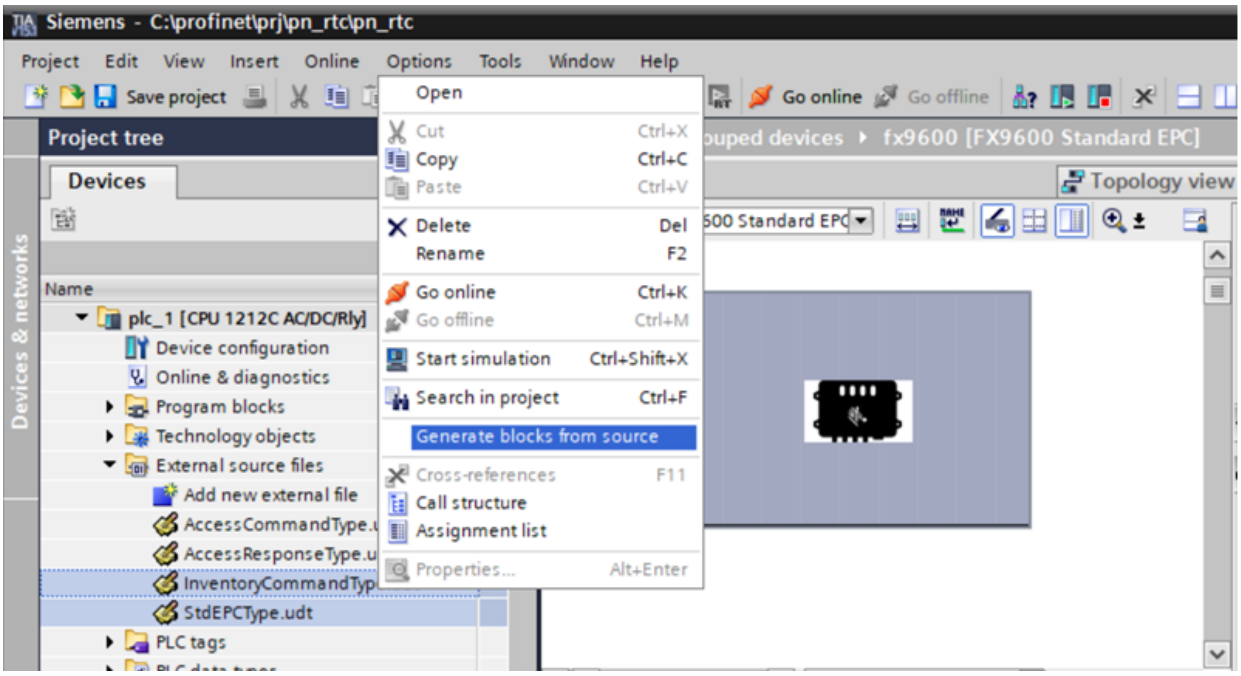
Add necessary UDTs supplied by Zebra for the FX9600 Reader by selecting Add new external file, shown in Figure 7.

Figure 7 Adding a New External File



After adding the necessary UDT files, create UDTs by right clicking and selecting Generate blocks from source option, shown in Figure 8.

Figure 8 Creating UDTs


Siemens - C:\profinet\prj\pn_rtc\pn_rtc
Project Edit View Insert Online Options Tools Window Help
Save project
Go online
Go offline
Project tree
Devices
Devices & networks
Name
plc_1 [CPU 1212C AC/DC/Rly]
Device configuration
Online & diagnostics
Program blocks
Technology objects
External source files
Add new external file
AccessCommandType.u
AccessResponseType.u
InventoryCommandTyp
StdEPCType.udt
PLC tags
Open
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Delete Del
Rename F2
Go online Ctrl+K
Go offline Ctrl+M
Start simulation Ctrl+Shift+X
Search in project Ctrl+F
Generate blocks from source
Cross-references F11
Call structure
Assignment list
Properties... Alt+Enter
fx9600 [FX9600 Standard EPC]
Topology view

Module Parameters Configuration

RFID parameters like Profile, Antenna, Pre-Filter and Trigger Configurations can be set at the time of connection establishment between the PLC and device.

Accessing the Module Parameters Configuration Page

The Module Parameters Configuration page can be accessed from the Inventory Commnd_1 module and Access Command module.

Figure 9 Module Parameter Configuration Page

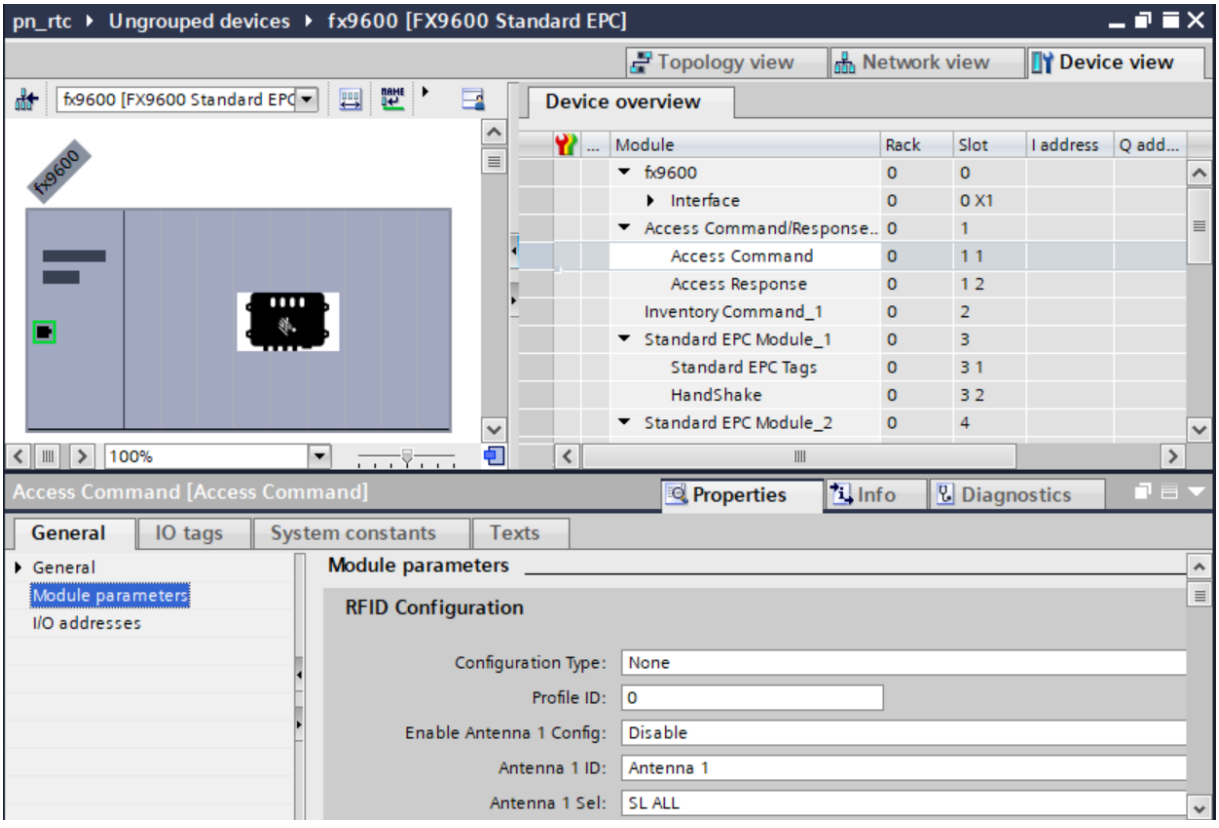
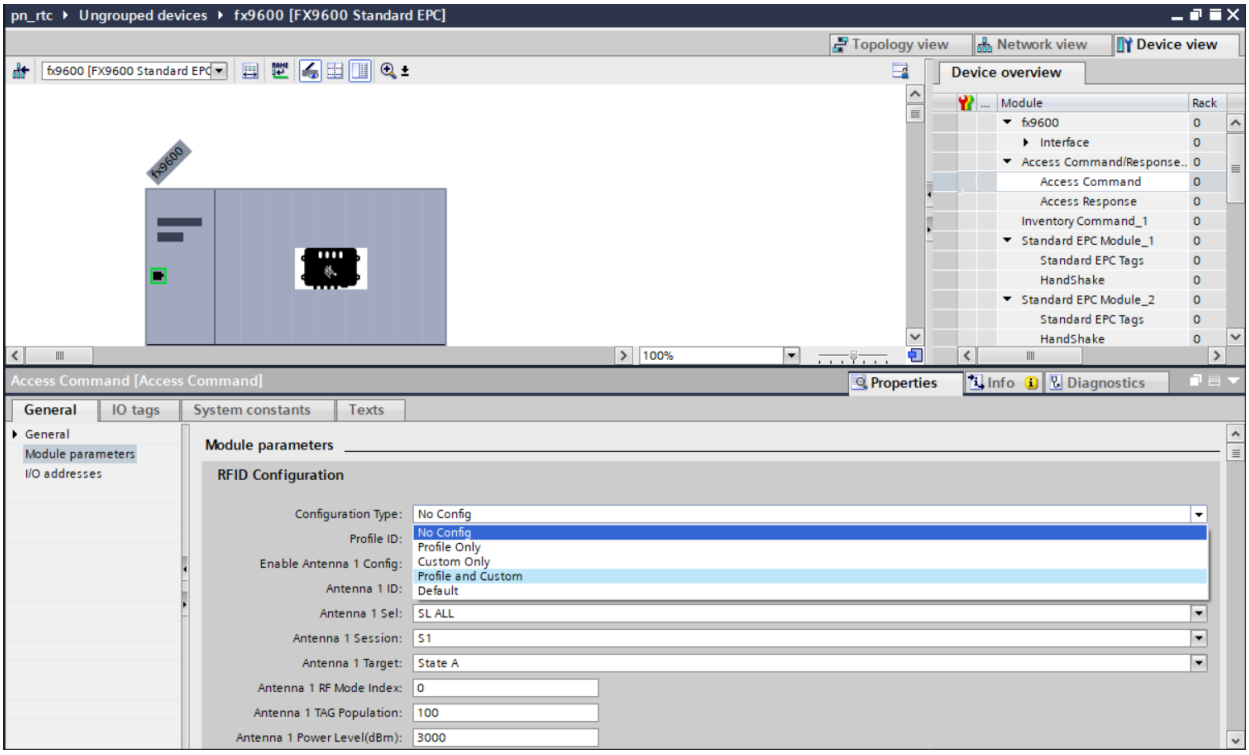


Figure 10 Access Command Module



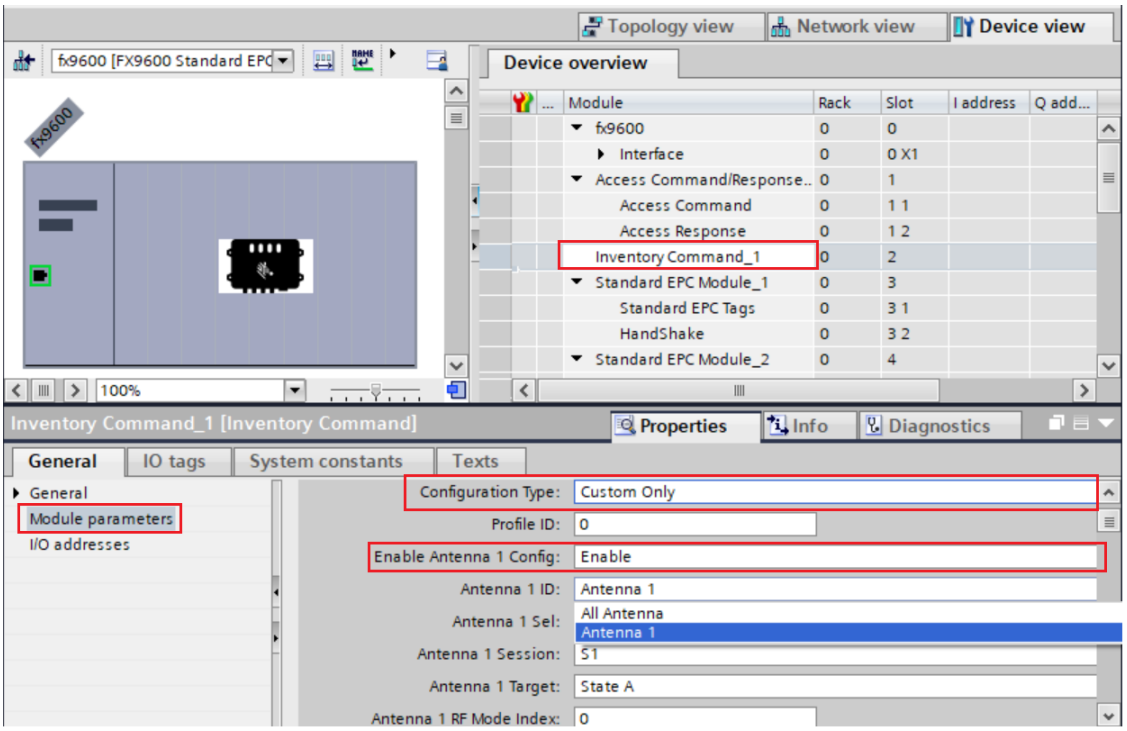
Configuration Type:

- No Config - Module parameters are not processed.
- Profile Only - Select profile on the reader using Index.
- Custom Only - To override default antenna, Pre-Filter and Trigger configurations.
- Profile and Custom - Select Profile and Override default antenna, Pre-Filter and Trigger configurations.
- Default - Default values are set for Profile, Antenna configuration, Pre-Filter ID 1 and Trigger configuration. Any previous settings for these options are discarded.

NOTE: When using both Access and Inventory commands, it is necessary to use module parameter configuration on only one of these modules. On the other module, Configuration Type should be set to the default value (None). This avoids overwriting configurations.

Antenna Configuration Using Module Parameters

Figure 11 Antenna Configuration



Antenna Configuration is applied when the Configuration Type is set to Custom Only or Profile and Custom.

Antenna configuration can be applied to individual Antenna ID from 1 – 8 by setting the Config Parameter to Enable.

Additionally, if all the Antennas need to have the same configuration, Antenna 1 ID configuration can be applied to all Antennas by selecting All Antenna in the drop-down menu, shown in Figure 11. If the Antenna configurations differ for each antenna, configurations can be applied to the respective Antenna IDs.

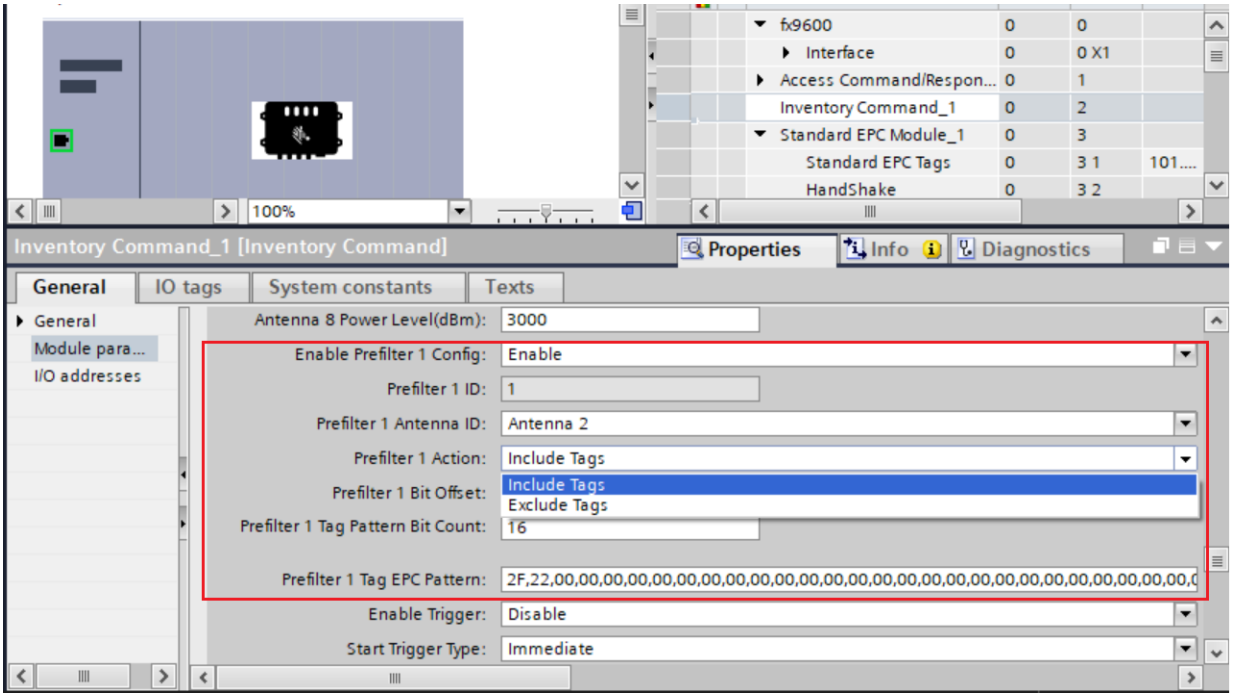
Power Level Range, RF Mode Index supported depends on the Region selected at time of initial Reader configuration.

Antenna Config with RF Mode Index beyond 39 can be set using Profinet Acyclic Write method (see Set/Get Parameters on page 44).

Pre-Filter Configuration using Module Parameters

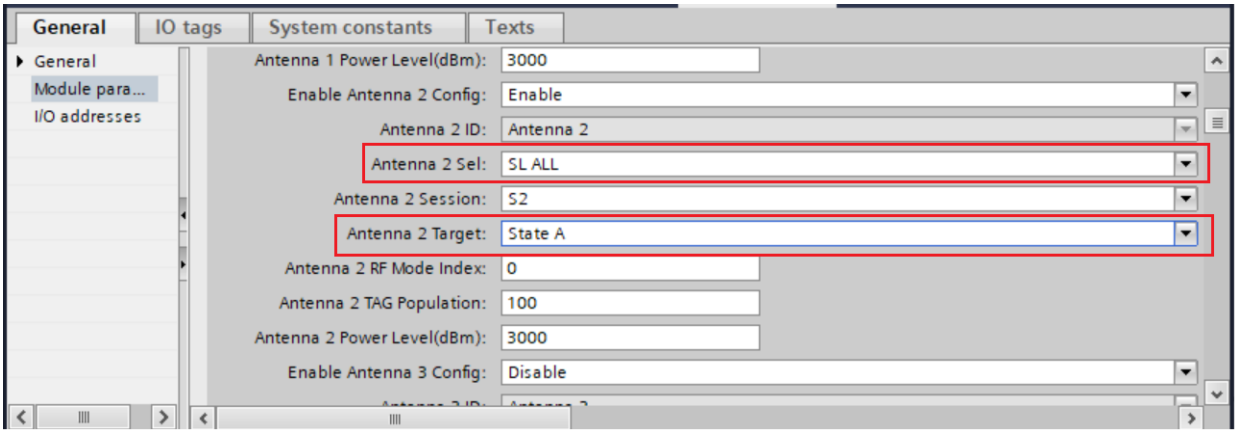
Pre-Filter can be configured to Include/Exclude tags based on the TAG EPC Pattern/ID.

Figure 12 Pre-Filter Configuration



In the example, tags with matching EPC Pattern 2F 22 at offset 0 are only included while reporting the tags found during inventory.

Figure 13 Antenna 2

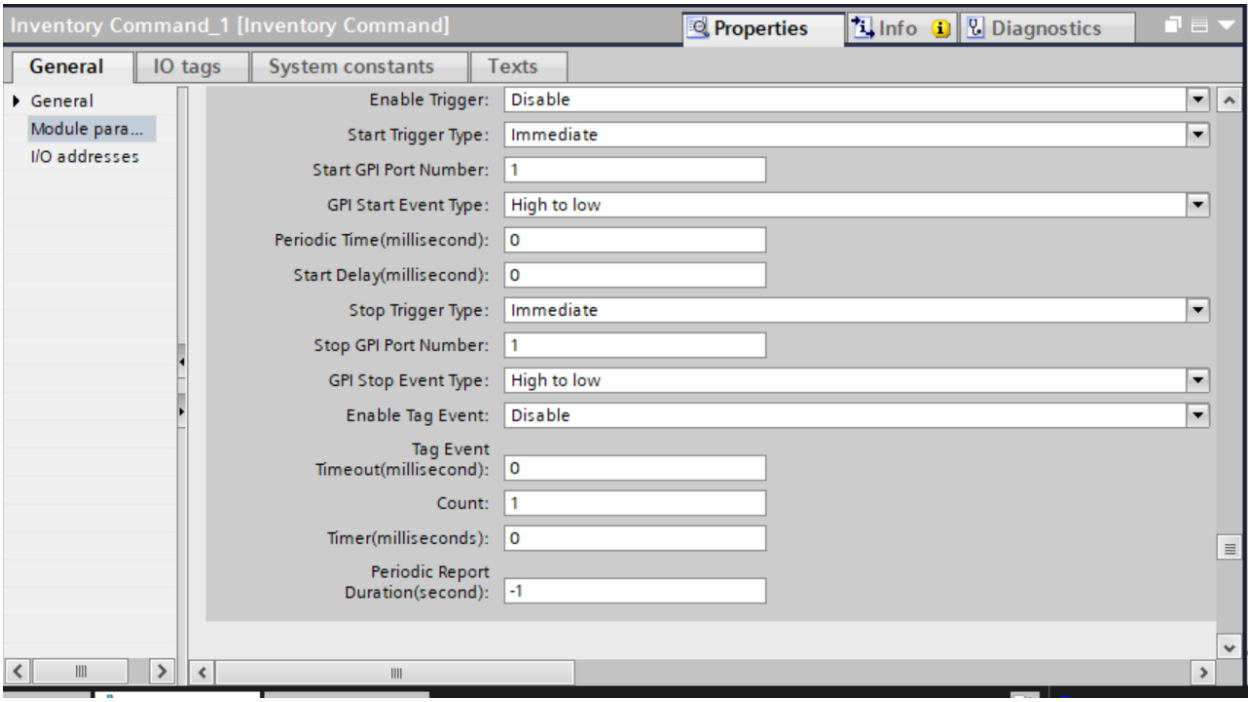


NOTE: In the selected Antenna ID of the Pre-Filter, Antenna 2 Sel must be set to SL ALL and Antenna 2 Target must be set to State A only. In the example shown in Figure 13, Pre-Filter is set in Antenna 2 ID, the Antenna 2 ID configuration parameters Sel and Target are set to SL ALL and State A, respectively.

Trigger Configurations Using Module Parameter

Trigger configurations can also be set using module parameters on the Inventory Command module as shown in Figure 14.

Figure 14 Trigger Configurations



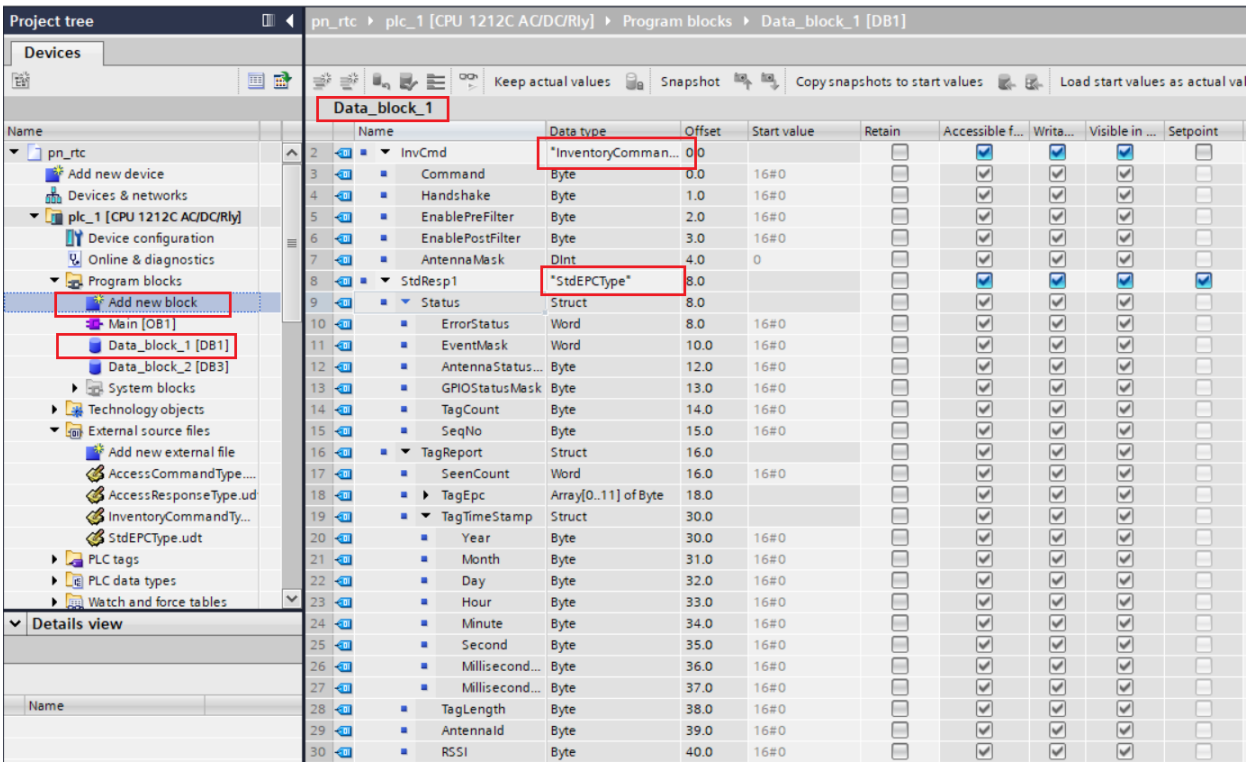
NOTE: Errors reported due to invalid values for any of these parameters can be read using an Acyclic read of Record Parameter Index 80 as described in Module Parameter Error Status on page 51.

Perform RFID Inventory Operation

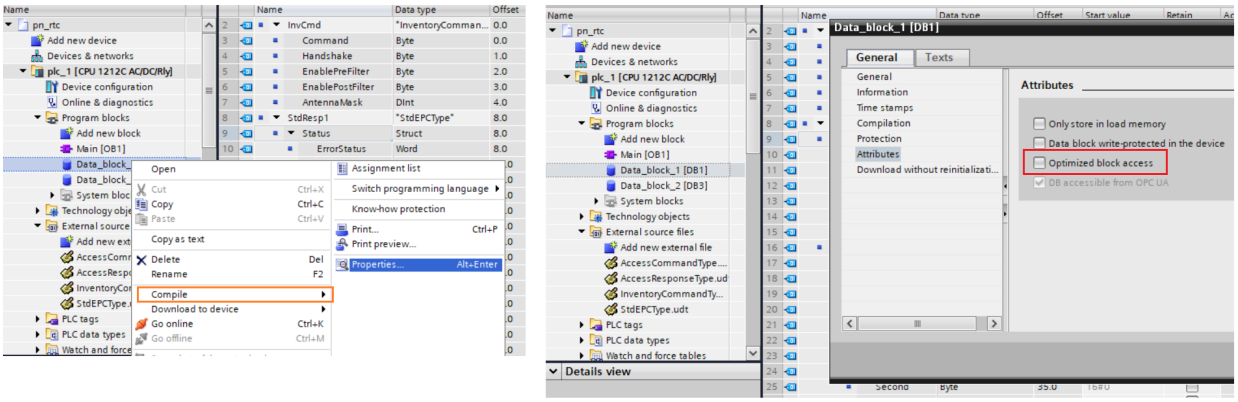
Siemens S7-1200 PLC and TIA Portal V15.1 are used to demonstrate the RFID Inventory start/stop operation.

Create Data Blocks and Variables to Use with Inventory Operation

1. Create s Data Block (Data_block_1) by selecting Add new block.
2. Create variables to hold Inventory Command and Standard EPC Response data and choose the appropriate Data Type, shown below. These variables are used in the Ladder Diagram to exchange data between PLC and the device.
   a. InvCmd - Variable is defined of type InventoryCommandType. It holds the data related to the Inventory command that is sent from the PLC to the device in each cycle.
   b. StdResp1 - Variable is defined of type stdEPCType. It holds the data related to 1 TAG EPC that is sent from the device to the PLC.



3. From the Properties page of the newly created Data Block, uncheck the Optimized block access.



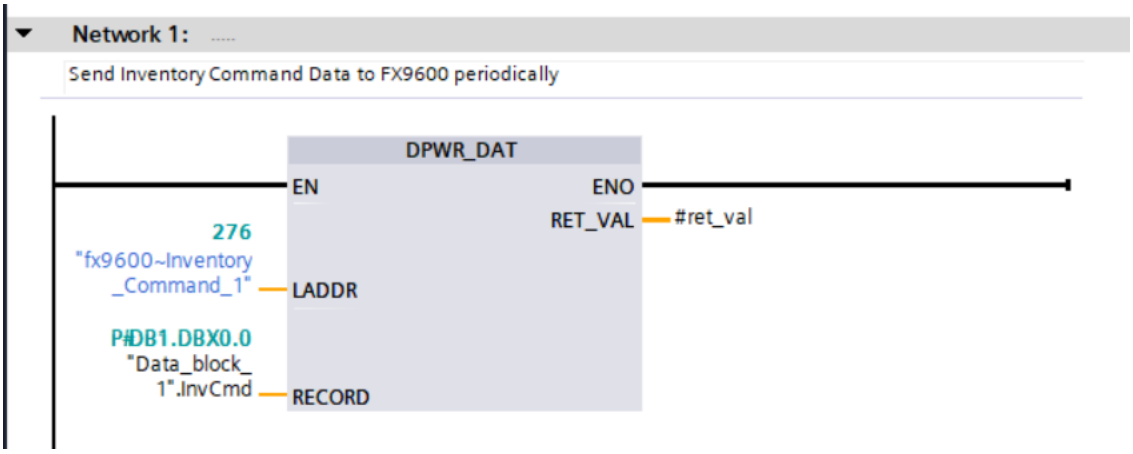
4. Compile the Data Block to use in the Ladder Diagram.

Inventory Command Ladder Diagram

DPWR_DAT instruction is used to send data from the PLC to the FX9600 Reader.

Figure 15 shows the Ladder Network1: Data_block_1.InvCmd variable holding the inventory command data sent from the PLC to the Inventory Command_1 module of the FX9600 Reader.

Figure 15 DPWR_DAT Instruction - Data Sent from PLC to the FX9600 Reader

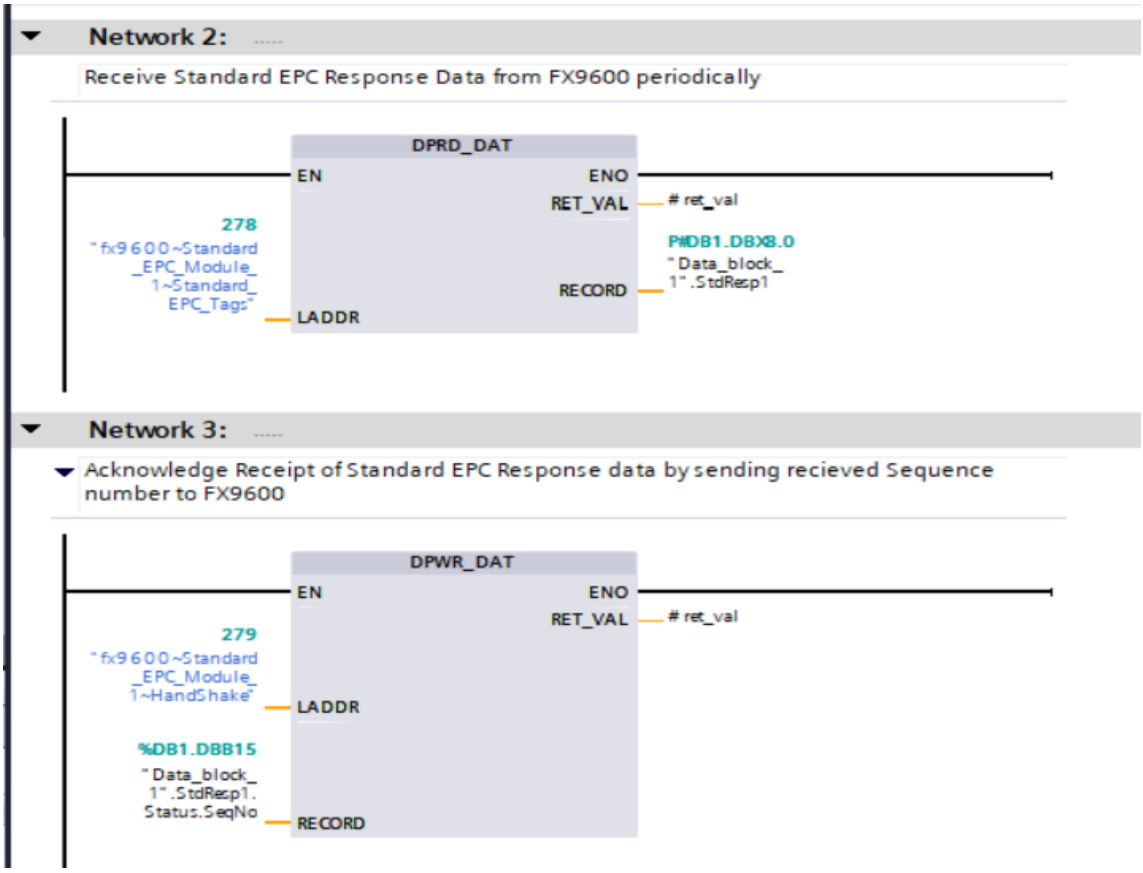


Inventory Response Ladder Diagram

DPRD_DAT instruction is used to receive data from the FX9600 Reader to the PLC.

In Figure 16, Ladder Network 3: Data_block_1.StdResp1.Status.SeqNo, the value received during Ladder Network 2 is sent back using the corresponding Handshake submodule to the FX9600 Reader as an acknowledgment of the receipt of standard EPC tag data.

Figure 16 DPRD_DAT Instruction - Data Received from the FX9600 Reader to the PLC

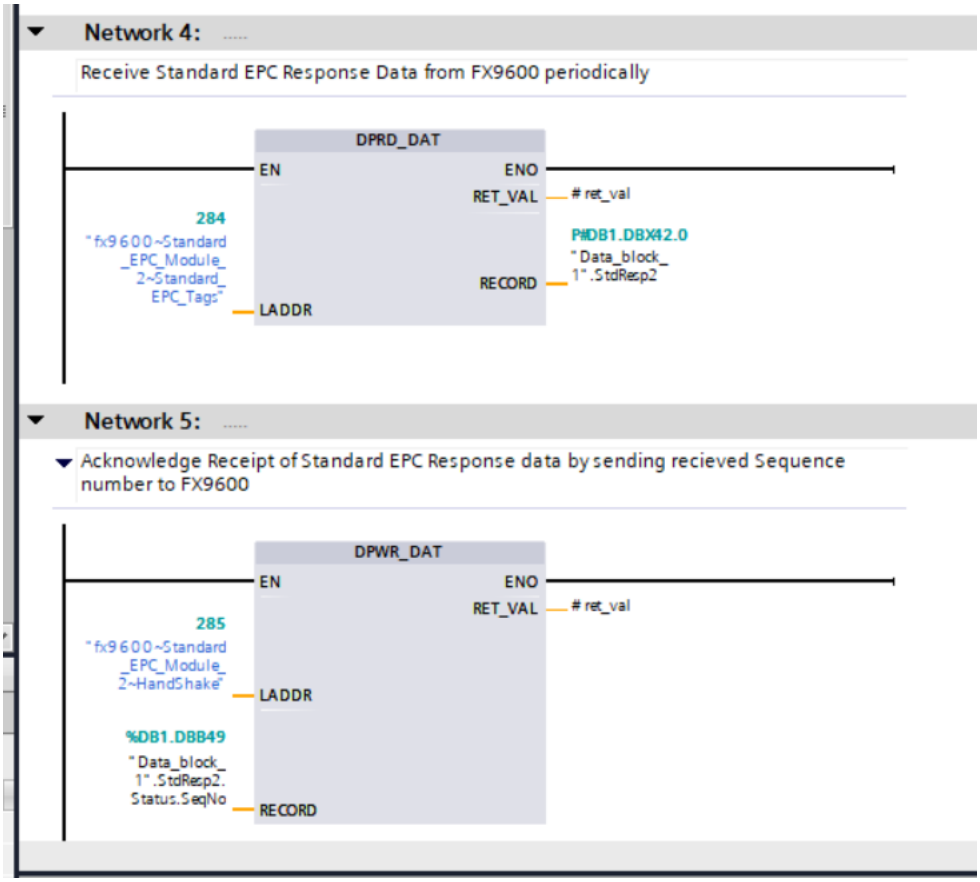


NOTE: The FX9600 Reader does not send the next tag data until it receives an acknowledge from the PLC for the previous tag data, shown in the Network 3 Ladder diagram above.

In Figure 17, Ladder Networks 4 and 5 are used to receive tag data from the **Standard EPC Module_2** of the FX9600 Reader into **Data_block_1.StdResp2** and to send the **Data_block_1.StdResp2.Status.SeqNo** as an acknowledgment by using the corresponding Handshake submodule to the FX9600.

These two ladder diagrams must be added for each Inventory Response module configured on the FX9600 Reader.

Figure 17 Ladder Networks 4 and 5 - Receive Tag Data from Standard EPC Module_2 of FX9600

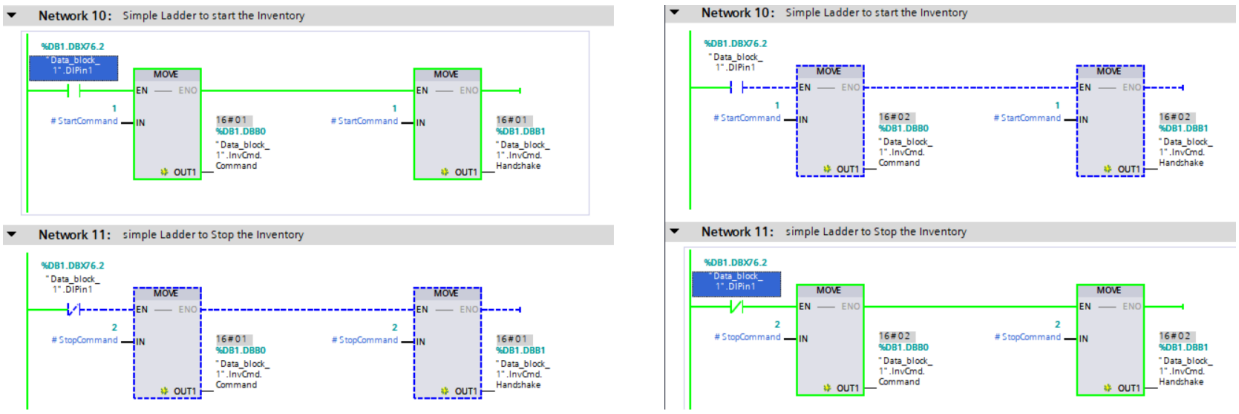


Simple Ladder Logic to Start and Stop Inventory

Tag Inventory can be started by setting Data_block_1.InvCmd.Command to 1 and modifying the Data_block_1.InvCmd.Handshake value to any number between 0 – 127.

Tag Inventory can be stopped by setting Data_block_1.InvCmd.Command to 2 and modifying the Data_block_1.InvCmd.Handshake value to any number between 0 – 127.

Figure 18 Ladder Logic to Start and Stop Inventory



NOTE: Data_block_1.InvCmd.Handshake value must be changed from the previous value to recognize it as a new command by the FX9600 Reader.

In Figure 18, both Data_block_1.InvCmd.Command and Data_block_1.InvCmd.Handshake are set to Value 1 (start Inventory) when Data_block_1.DIPin1 value is high to start the Inventory operation.

Also in Figure 18, both Data_block_1.InvCmd.Command and Data_block_1.InvCmd.Handshake are set to Value 2 (stop inventory) when Data_block_1.DIPin1 value is low to stop the Inventory operation.

Inventory Command Parameter Structure

Table 1 Inventory Command Parameter Structure

Inventory Command	Data Type	Offset	Comments
Command	Byte	0	This field is used to specify the inventory commands. It can be either 1 (Start) or 2 (Stop).
Handshake	Byte	1	For each new command, Handshake value must be modified to instruct FX9600 Reader to process the command. Handshake parameter can take any value between 0-127. one way to use is to increment each time a new command is to be processed and roll it over when it reaches 127 to 0.
EnablePreFilter	Byte	2	Used to specify whether the user wants to use pre-filter with this inventory command. Non-zero value is treated as TRUE and pre-filter takes effect.
EnablePostFilter	Byte	3	Used to specify whether post-filter is used. Criteria is the same as pre-filter.
AntennaMask	DInt	4	Use this mask to specify the antenna IDs on which this inventory operation should perform. Each bit in the antenna mask, from 0 bit, represents one antenna. 0 bit for antenna ID 1, and so on. If the AntennaMask is 0, then inventory is performed with all available antennas.

Inventory Response Parameter Structure

The FX9600 Reader supports four virtual DAPs (see Select the Required Virtual Device Access Point (DAP) for the PROFINET App on page 13). Only one DAP can be active at any time.

Each DAP supports Identical Access Command/Response and Inventory Command modules and a unique Inventory Response module.

Standard EPC Parameter Structure

This parameter is used to interpret Inventory Response when the FX9600 Standard EPC DAP is chosen at the time the FX9600 Reader is configured (see Configure Various Modules on the FX9600 Reader on page 19).

Table 2 Standard EPC Parameter Structure

Standard Inventory Response	Data Type	Offset	Comments
Status	Struct	0	
ErrorStatus	Word	0	RFID errors are reported in this field.
EventMask	Word	2	RFID events are reported in this field.
AntennaStatusMask	Byte	4	Currently connected antennas. Bit n = Antenna ID n
GPIOStatusMask	Byte	5	Current GPIO pin status
TagCount	Byte	6	1
SeqNo	Byte	7	SeqNo is incremented with each new report. Note that same tag may be read several times and is reported each time; SeqNo is also incremented to indicate that tag was newly read.
TagReport	Struct	8	Tag info data.
SeenCount	Word	8	Number of times the tag was read.
TagEpc	Byte[12]	10	EPC ID of the tag.
TagTimeStamp	Struct	22	Date and time of the tag read.
Year	Byte	22	Year 2020 is reported as 20 (after subtracting 2000).
Month	Byte	23	Month.
Day	Byte	24	Day.
Hour	Byte	25	Hour.
Minute	Byte	26	Minute.
Second	Byte	27	Second.
MillisecondMSB	Byte	28	MSB value of millisecond.
MillisecondLSB	Byte	29	LSB value of millisecond.
TagLength	Byte	30	Length of EPC ID of the tag.
AntennaId	Byte	31	Antenna ID on which tag was read.
RSSI	Byte	32	RSSI value of the tag read.

NOTE: This parameter structure is supplied as StdEPCType.udt file which can be imported while using Siemens TIA portal. For other PLC vendors this User Defined Data type needs to be created manually.

Extended Standard EPC Parameter Structure

This parameter is used to interpret Inventory Response when the FX9600 Standard EPC Plus DAP is chosen at the time the FX9600 Reader is configured (see Configure Various Modules on the FX9600 Reader on page 19).

NOTE: This parameter structure is supplied as ExtStdEPCType.udt file which can be imported while using the Siemens TIA portal. For other PLC vendors this User Defined Data type needs to be created manually.

Table 3 Extended Standard EPC Parameter Structure

Extended Standard Inventory Response	Data Type	Offset	Comments
Status	Struct	0	
ErrorStatus	Word	0	RFID errors are reported in this field.
EventMask	Word	2	RFID events are reported in this field.
AntennaStatusMask	Byte	4	Currently connected antennas. Bit n = Antenna ID n.
GPIOStatusMask	Byte	5	Current GPIO pin status.
TagCount	Byte	6	1.
SeqNo	Byte	7	SeqNo is incremented with each new report. Note that same tag may be read several times and is reported each time; SeqNo is also incremented to indicate that tag was newly read.
TagReport	Struct	8	Tag info data.
SeenCount	Word	8	Number of times the tag was read
TagPC	Word	10	PC of the tag read
TagCRC	Word	12	CRC of the tag read
ChannelIndex	Word	14	Channel On which tag was read
PhaseInfo	Word	16	Phase angle of the tag Read
AccessStatus	Word	18	Results of Access operation
TagEpc	Byte[12]	20	EPC ID of the tag
TagTimeStamp	Struct	32	Date and time of the tag read.
Year	Byte	32	Year 2020 is reported as 20 (after subtracting 2000)
Month	Byte	33	Month
Day	Byte	34	Day
Hour	Byte	35	Hour
Minute	Byte	36	Minute
Second	Byte	37	Second
MillisecondMSB	Byte	38	MSB value of millisecond
MillisecondLSB	Byte	39	LSB value of millisecond
TagLength	Byte	40	Length of EPC ID of the tag
AntennaId	Byte	41	Antenna ID on which tag was read
RSSI	Byte	42	RSSI value of the tag read
TagEvent	Byte	43	Tag Event Info

Long EPC Parameter Structure

This parameter is used to interpret Inventory Response when the FX9600 Long EPC DAP is chosen at the time the FX9600 Reader is configured (see Configure Various Modules on the FX9600 Reader on page 19).

Table 4 Long Inventory Response

Long Inventory Response	Data Type	Offset	Comments
Status	Struct	0	
ErrorStatus	Word	0	RFID errors are reported in this field.
EventMask	Word	2	RFID events are reported in this field.
AntennaStatusMask	Byte	4	Currently Connected Antennas. Bit n = Antenna ID n.
GPIOStatusMask	Byte	5	Current GPIO pin status.
TagCount	Byte	6	1.
SeqNo	Byte	7	SeqNo is incremented with each new report. Note that same tag may be read several times and is reported each time; SeqNo is also incremented to indicate that tag was newly read.
TagReport	Struct	8	Tag info data.
SeenCount	Word	8	Number of times the tag was read.
TagEpc	Byte[64]	10	EPC ID of the tag.
TagTimeStamp	Struct	74	Date and time of the tag read.
Year	Byte	74	Year 2020 are reported as 20 (after subtracting 2000).
Month	Byte	75	Month
Day	Byte	76	Day
Hour	Byte	77	Hour
Minute	Byte	78	Minute
Second	Byte	79	Second
MillisecondMSB	Byte	80	MSB value of millisecond.
MillisecondLSB	Byte	81	LSB value of millisecond.
TagLength	Byte	82	Length of EPC ID of the tag.
AntennaId	Byte	83	Antenna ID on which tag was read.
RSSI	Byte	84	RSSI value of the tag read.

NOTE: This parameter structure is supplied as LongEPCType.udt file which can be imported while using the Siemens TIA portal. For other PLC vendors this User Defined Data type needs to be created manually.

Extended Long EPC Parameter Structure

This parameter is used to interpret Inventory Response when the FX9600 Long EPC Plus DAP is chosen at the time the FX9600 Reader is configured (see Configure Various Modules on the FX9600 Reader on page 19).

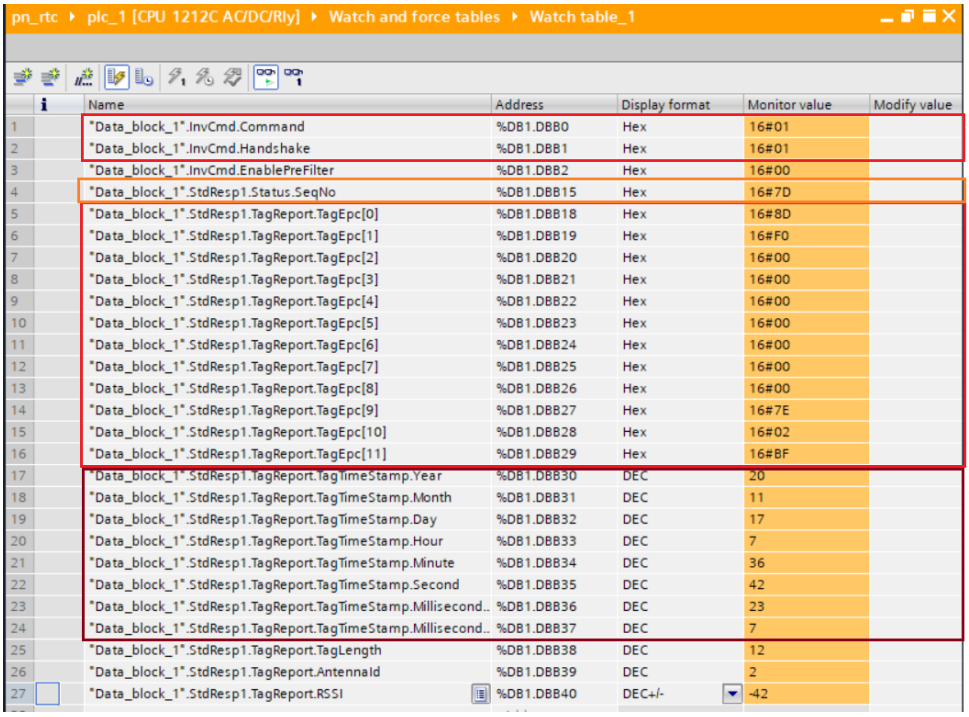
Table 5 Extended Long EPC Parameter Structure

Extended Long Inventory Response	Data Type	Offset	Comments
Status	Struct	0	
ErrorStatus	Word	0	RFID errors are reported in this field.
EventMask	Word	2	RFID events are reported in this field.
AntennaStatusMask	Byte	4	Currently Connected Antennas. Bit n = Antenna ID n.
GPIOStatusMask	Byte	5	Current GPIO pin status.
TagCount	Byte	6	1
SeqNo	Byte	7	SeqNo is incremented with each new report. Note that same tag may be read several times and is reported each time; SeqNo is also incremented to indicate that tag was newly read.
TagReport	Struct	8	Tag info data.
SeenCount	Word	8	Number of times the tag was read.
TagPC	Word	10	PC of the tag Read.
TagCRC	Word	12	CRC of the tag Read.
ChannelIndex	Word	14	Channel On which tag was read.
PhaseInfo	Word	16	Phase angle of the tag Read.
AccessStatus	Word	18	Results of Access operation.
TagEpc	Byte[64]	20	EPC ID of the tag.
TagTimeStamp	Struct	84	Date and time of the tag read.
Year	Byte	84	Year 2020 are reported as 20 (after subtracting 2000).
Month	Byte	85	Month
Day	Byte	86	Day
Hour	Byte	87	Hour
Minute	Byte	88	Minute
Second	Byte	89	Second
MillisecondMSB	Byte	90	MSB value of millisecond.
MillisecondLSB	Byte	91	LSB value of millisecond.
TagLength	Byte	92	Length of EPC ID of the tag.
AntennaId	Byte	93	Antenna ID on which tag was read.
RSSI	Byte	94	RSSI value of the tag read.
TagEvent	Byte	95	Tag Event Info.

NOTE: This parameter structure is supplied as ExtLongEPCType.udt file which can be imported while using Siemens TIA portal. For other PLC vendors this User Defined Data type needs to be created manually.

Sample Inventory Response Output

Figure 19 Example Tag Inventory Response - Watch Table TIA Portal



	Name	Address	Display format	Monitor value	Modify value
1	"Data_block_1".InvCmd.Command	%DB1.DBB0	Hex	16#01	
2	"Data_block_1".InvCmd.Handshake	%DB1.DBB1	Hex	16#01	
3	"Data_block_1".InvCmd.EnablePreFilter	%DB1.DBB2	Hex	16#00	
4	"Data_block_1".StdResp1.Status.SeqNo	%DB1.DBB15	Hex	16#7D	
5	"Data_block_1".StdResp1.TagReport.TagEpc[0]	%DB1.DBB18	Hex	16#8D	
6	"Data_block_1".StdResp1.TagReport.TagEpc[1]	%DB1.DBB19	Hex	16#F0	
7	"Data_block_1".StdResp1.TagReport.TagEpc[2]	%DB1.DBB20	Hex	16#00	
8	"Data_block_1".StdResp1.TagReport.TagEpc[3]	%DB1.DBB21	Hex	16#00	
9	"Data_block_1".StdResp1.TagReport.TagEpc[4]	%DB1.DBB22	Hex	16#00	
10	"Data_block_1".StdResp1.TagReport.TagEpc[5]	%DB1.DBB23	Hex	16#00	
11	"Data_block_1".StdResp1.TagReport.TagEpc[6]	%DB1.DBB24	Hex	16#00	
12	"Data_block_1".StdResp1.TagReport.TagEpc[7]	%DB1.DBB25	Hex	16#00	
13	"Data_block_1".StdResp1.TagReport.TagEpc[8]	%DB1.DBB26	Hex	16#00	
14	"Data_block_1".StdResp1.TagReport.TagEpc[9]	%DB1.DBB27	Hex	16#7E	
15	"Data_block_1".StdResp1.TagReport.TagEpc[10]	%DB1.DBB28	Hex	16#02	
16	"Data_block_1".StdResp1.TagReport.TagEpc[11]	%DB1.DBB29	Hex	16#BF	
17	"Data_block_1".StdResp1.TagReport.TagTimeStamp.Year	%DB1.DBB30	DEC	20	
18	"Data_block_1".StdResp1.TagReport.TagTimeStamp.Month	%DB1.DBB31	DEC	11	
19	"Data_block_1".StdResp1.TagReport.TagTimeStamp.Day	%DB1.DBB32	DEC	17	
20	"Data_block_1".StdResp1.TagReport.TagTimeStamp.Hour	%DB1.DBB33	DEC	7	
21	"Data_block_1".StdResp1.TagReport.TagTimeStamp.Minute	%DB1.DBB34	DEC	36	
22	"Data_block_1".StdResp1.TagReport.TagTimeStamp.Second	%DB1.DBB35	DEC	42	
23	"Data_block_1".StdResp1.TagReport.TagTimeStamp.Millisecond..	%DB1.DBB36	DEC	23	
24	"Data_block_1".StdResp1.TagReport.TagTimeStamp.Millisecond..	%DB1.DBB37	DEC	7	
25	"Data_block_1".StdResp1.TagReport.TagLength	%DB1.DBB38	DEC	12	
26	"Data_block_1".StdResp1.TagReport.AntennaId	%DB1.DBB39	DEC	2	
27	"Data_block_1".StdResp1.TagReport.RSSI	%DB1.DBB40	DEC+/-	-42	

Perform RFID Access Operation

Create Data Blocks and Variables to Use With Access Operation

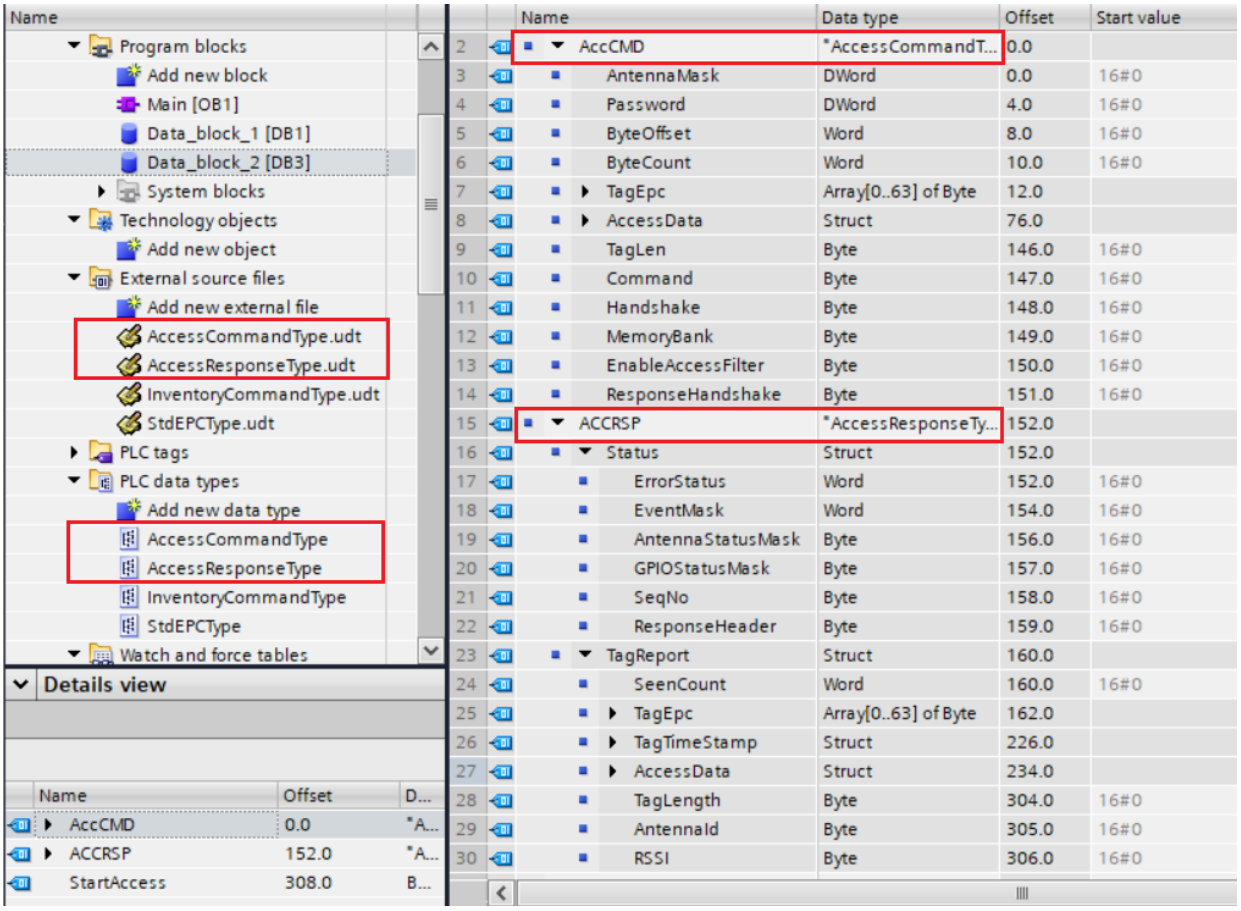
Add Zebra Supplied AccessCommandType.udt and AccessResponseType.udt external files and generate data types to use them with Data blocks.

In Data_block_2 (Figure 20), AccessCMD variable is created with data type AccessCommandType and AccessRSP variable with AccessResponseType.

Make sure to uncheck the optimized bloc access attribute and compile Data_block_2 before using in the Ladder diagrams.

The user can set default values various parameter as needed.

Figure 20 Data_block_2

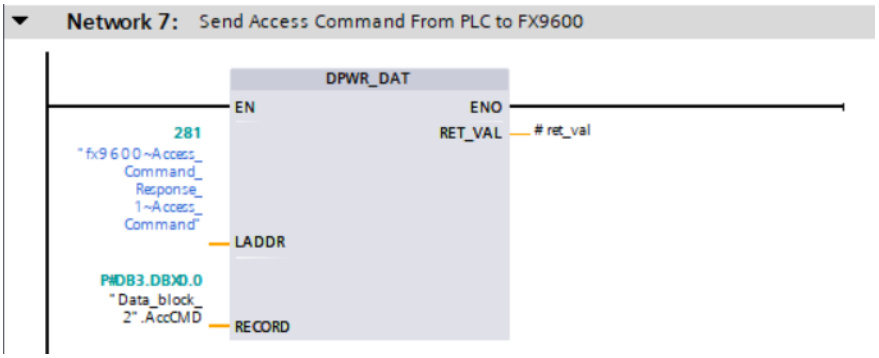


Access Command Ladder Diagram

The DPWR_DAT instruction is used to send data from the PLC to the FX9600 Reader.

In Figure 21, the Ladder Network7: Data_block_2.AccCMD variable holding the Access command data is sent from the PLC to the Access Command module of the FX9600 Reader.

Figure 21 DPWR_DAT Iinstruction



Access Response Ladder Diagram

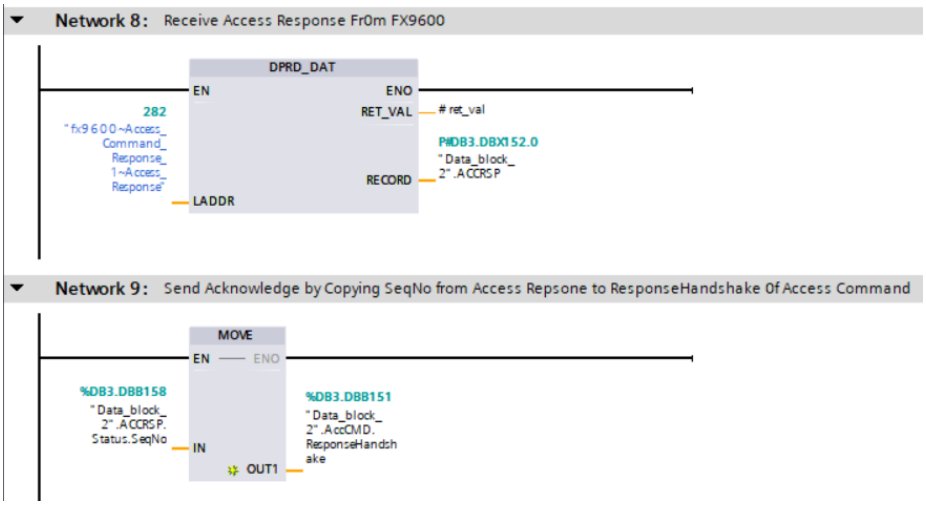
In Figure 22, Network 8, the Ladder Network 8: Data_block_2.ACCRSP variable holds the received tag data in the Access Response module from the the FX9600 in the current cycle.

In Figure 22, Network 9, the Ladder Network 9: Data_block_2.ACCRSP.Status.SeqNo value, received during the Ladder Network 8, is sent back in the Data_block_2.AccCMD. The ResponseHanshake parameter uses the Access Command module in the next cycle of execution to the FX9600 Reader as an acknowledgment of the Receipt access response data.

NOTE: Without receiving the acknowledgment from the PLC, the FX9600 Reader cannot send the next response data with the updated SeqNo.

NOTE: Unlike the Inventory Operation, the Access Operation is done for only one round and the result is reported to the PLC. To repeat the Access operation, the Handshake parameter in the Access Command must be incremented to do another round of Access operation. The Handshake value has a range of 0-127.

Figure 22 Data_block_2.ACCRSP and Data_block_2.ACCRSP.Status.SeqNo



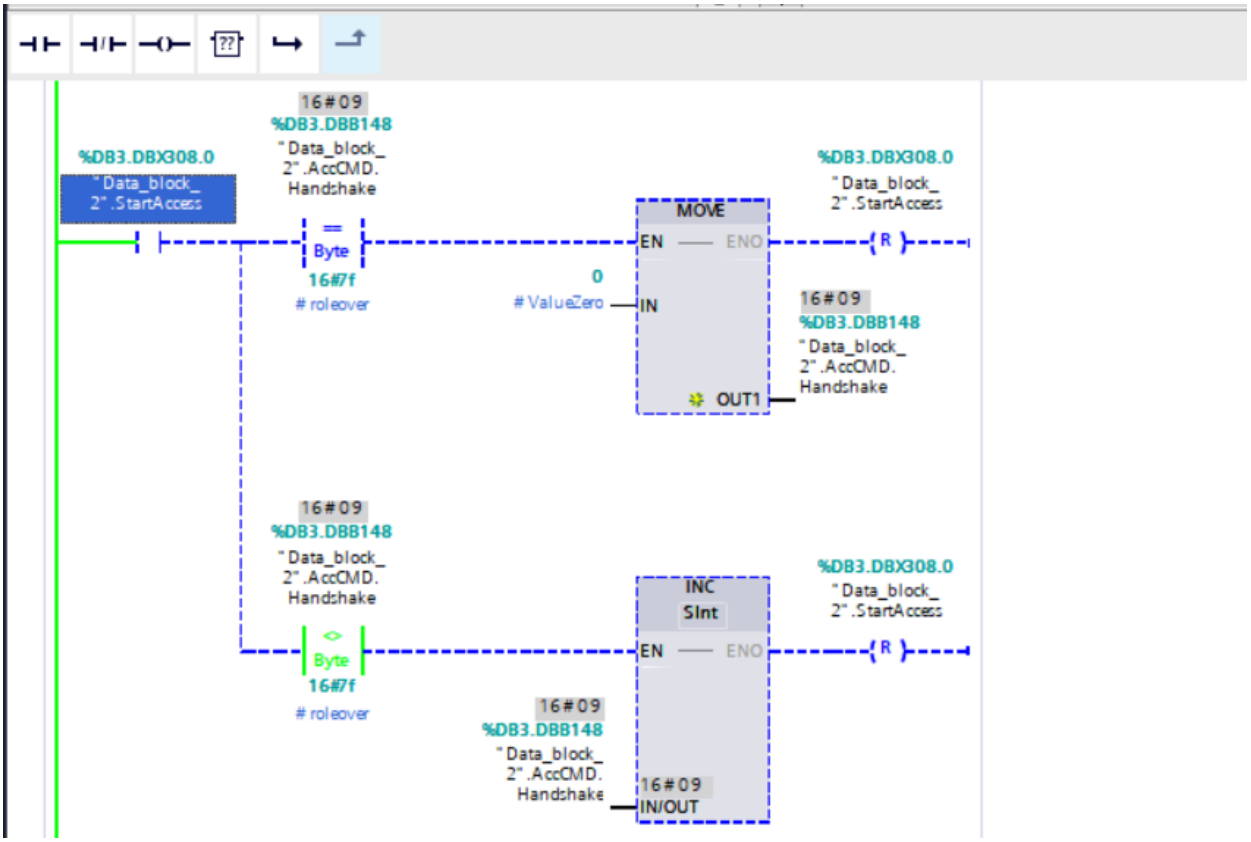
Simple Ladder Diagram to Perform Access Operation

Figure 23 shows a sample Ladder diagram to increment the Handshake parameter when the access operation needs to be performed by the reader. This Ladder diagram rolls-over the Handshake parameter when it reaches 127 (0x7f).

Data_block_2.StartAccess (bool) variable can be set to True to initiate one round of the Access operation. Data_block_2.StartAccess is set to False at the end of operation, as shown.

Each time the Access operation needs to be performed, Data_block_2.StartAccess can be set to True.

Figure 23 Sample Ladder Diagram



Access Command Parameter Structure

NOTE: This parameter structure is supplied as AccessCommandType.udt file which can be imported while using the Siemens TIA portal. For other PLC vendors this User Defined Data type needs to be created manually.

Table 6 Access Command Parameter Structure

Access Command	Data Type	Offset	Comments
AntennaMask	DWord	0	Use this mask to specify the antenna IDs on which this inventory operation should perform. Each bit in the antenna mask, from 0 bit, represents one antenna. 0 bit for antenna ID 1, and so on. If the AntennaMask is 0, then inventory is performed with all available antennas.
Password	DWord	4	Password if the memory bank needs a password for access.
ByteOffset	Word	8	Byte offset from where the memory needs to access.
ByteCount	Word	10	Number of bytes to read. 0 indicates all the bytes available.
TagEpc	Byte[64]	12	EPC of the tag on which the access operation is performed. This field can be used to provide EPC pattern as well. For example, all tags starting with 2F 22.
AccessData	Struct	76	Access data info.
PageIndex	DWord	76	0
PageDataLen	Word	80	Length of access data bytes to write when using Access Write.
PageData	Byte[64]	82	Access data bytes to write when using Access Write.
TagLen	Byte	146	Length if EPC tag/pattern provided.
Command	Byte	147	1- Access Read 2 - Access Write
Handshake	Byte	148	For each new command, the Handshake value must be modified to instruct the FX9600 Reader to process the command. The Handshake parameter can take any value between 0-127. One way to use it is to increment each time a new command is processed and roll it over when it reaches 127 to 0.
MemoryBank	Byte	149	0 - Reserved 1 - EPC 2 - TID 3 - User memory
EnableAccessFilter	Byte	150	Non-zero value is treated as TRUE and Access Filter takes effect.
ResponseHandshake	Byte	151	SeqNo received in the Access Response must be copied here to be sent to the FX9600 Reader as acknowledge, upon which the FX9600 sends the next available data.

Access Response Parameter Structure

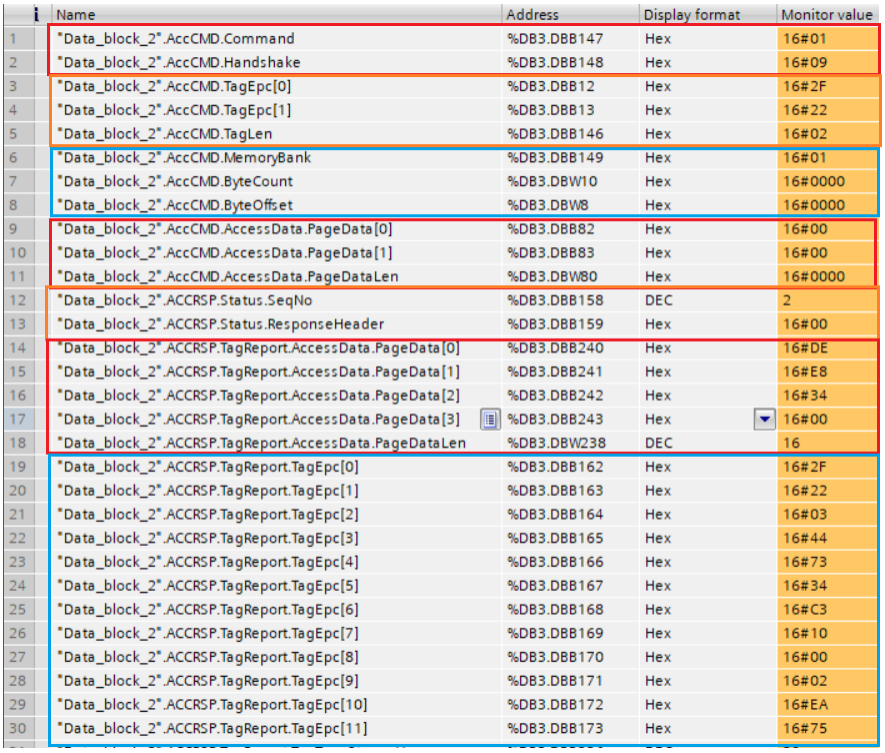
NOTE: This parameter structure is supplied as AccessResponseType.udt file which can be imported while using the Siemens TIA portal. For other PLC vendors this User Defined Data type needs to be created manually.

Table 7 Access Response Parameter Structure

Access Response	Data Type	Offset	Comments
Status	Struct	0	
ErrorStatus	Word	0	RFID errors are reported in this field.
EventMask	Word	2	RFID events are reported in this field.
AntennaStatusMask	Byte	4	Currently Connected Antennas. Bit n = Antenna ID n.
GPIOStatusMask	Byte	5	Current GPIO pin status.
TagReport	Struct	6	Tag info data.
SeenCount	Word	6	Number of times tag was read.
TagEpc	Byte[64]	8	EPC ID of the tag.
TagTimeStamp	Struct	72	Date and time of the tag read.
Year	Byte	72	Year 2020 is reported as 20 (after subtracting 2000).
Month	Byte	73	Month
Day	Byte	74	Day
Hour	Byte	75	Hour
Minute	Byte	76	Minute
Second	Byte	77	Second
MillisecondMSB	Byte	78	MSB value of millisecond.
MillisecondLSB	Byte	79	LSB value of millisecond.
AccessData	Struct	80	Contents of the Memory Bank accessed.
PageIndex	DWord	80	PageiIndex of the memory.
PageDataLen	Word	84	Length memory data accessed.
PageData	Byte[64]	86	Contents of the Memory Bank.
TagLength	Byte	150	Length of EPC ID of the tag.
AntennaId	Byte	151	Antenna ID on which tag was read.
RSSI	Byte	152	RSSI value of the tag read.
SeqNo	Byte	153	SeqNo is incremented with each new report. Note that the same tag may be read several times and is reported each time; SeqNo is also incremented to indicate that tag was newly read.
ResponseHeader	Byte	154	0 - Success 1 - Tag found but Access operation failed. 2 - Tag not found and Access operation failed.

Sample Access Response Output

Figure 24 Sample Access Response (Read Operation) Output - Watch Table TIA Portal



	Name	Address	Display format	Monitor value
1	"Data_block_2".AccCMD.Command	%DB3.DBB147	Hex	16#01
2	"Data_block_2".AccCMD.Handshake	%DB3.DBB148	Hex	16#09
3	"Data_block_2".AccCMD.TagEpc[0]	%DB3.DBB12	Hex	16#2F
4	"Data_block_2".AccCMD.TagEpc[1]	%DB3.DBB13	Hex	16#22
5	"Data_block_2".AccCMD.TagLen	%DB3.DBB146	Hex	16#02
6	"Data_block_2".AccCMD.MemoryBank	%DB3.DBB149	Hex	16#01
7	"Data_block_2".AccCMD.ByteCount	%DB3.DBW10	Hex	16#0000
8	"Data_block_2".AccCMD.ByteOffset	%DB3.DBW8	Hex	16#0000
9	"Data_block_2".AccCMD.AccessData.PageData[0]	%DB3.DBB82	Hex	16#00
10	"Data_block_2".AccCMD.AccessData.PageData[1]	%DB3.DBB83	Hex	16#00
11	"Data_block_2".AccCMD.AccessData.PageDataLen	%DB3.DBW80	Hex	16#0000
12	"Data_block_2".ACCRSP.Status.SeqNo	%DB3.DBB158	DEC	2
13	"Data_block_2".ACCRSP.Status.ResponseHeader	%DB3.DBB159	Hex	16#00
14	"Data_block_2".ACCRSP.TagReport.AccessData.PageData[0]	%DB3.DBB240	Hex	16#DE
15	"Data_block_2".ACCRSP.TagReport.AccessData.PageData[1]	%DB3.DBB241	Hex	16#E8
16	"Data_block_2".ACCRSP.TagReport.AccessData.PageData[2]	%DB3.DBB242	Hex	16#34
17	"Data_block_2".ACCRSP.TagReport.AccessData.PageData[3]	%DB3.DBB243	Hex	16#00
18	"Data_block_2".ACCRSP.TagReport.AccessData.PageDataLen	%DB3.DBW238	DEC	16
19	"Data_block_2".ACCRSP.TagReport.TagEpc[0]	%DB3.DBB162	Hex	16#2F
20	"Data_block_2".ACCRSP.TagReport.TagEpc[1]	%DB3.DBB163	Hex	16#22
21	"Data_block_2".ACCRSP.TagReport.TagEpc[2]	%DB3.DBB164	Hex	16#03
22	"Data_block_2".ACCRSP.TagReport.TagEpc[3]	%DB3.DBB165	Hex	16#44
23	"Data_block_2".ACCRSP.TagReport.TagEpc[4]	%DB3.DBB166	Hex	16#73
24	"Data_block_2".ACCRSP.TagReport.TagEpc[5]	%DB3.DBB167	Hex	16#34
25	"Data_block_2".ACCRSP.TagReport.TagEpc[6]	%DB3.DBB168	Hex	16#C3
26	"Data_block_2".ACCRSP.TagReport.TagEpc[7]	%DB3.DBB169	Hex	16#10
27	"Data_block_2".ACCRSP.TagReport.TagEpc[8]	%DB3.DBB170	Hex	16#00
28	"Data_block_2".ACCRSP.TagReport.TagEpc[9]	%DB3.DBB171	Hex	16#02
29	"Data_block_2".ACCRSP.TagReport.TagEpc[10]	%DB3.DBB172	Hex	16#EA
30	"Data_block_2".ACCRSP.TagReport.TagEpc[11]	%DB3.DBB173	Hex	16#75

1: read, 2: write Command
Handshake to signal start of operation

Tags starting with 2F 22

Read Memory Bank EPC at offset 0

Used to supply the data while using Access write

SeqNo incremented with new data
ResponeHeader 0: Access success

Access Data read form the memory bank of the Tag

EPC of the Tag on which Access read operation was performed

RFID SET/GET Parameter

RFID Parameters

RFID parameters can be read from or written to the FX9600 Reader using the PROFINET Acyclic Write feature.

Each of these RFID Parameter Structures are supplied as User Defined Data Type (UDT) files with the Zebra PROFINET Package for FX9600. These UDT files can be used to create the Data Types in the Siemens TIA portal as shown in Perform RFID Inventory Operation. For other PLC vendors, see Table 9 and define data types accordingly.

Table 8 RFID Parameters

Index	Size	Name	Comments
0		Record Parameter	Configuration at the time of first connect
1	115	Reader Capability<XREF>	Reader capability read only
2	2	Set Active Profile	Write only
2	66	Get Active Profile Name	Read (use profile name Data Type)
3-34	66	Get Profile Name	Read only
35 - 42	11	Antenna Config	R/W
43	1	Delete Pre-Filter	Write only
44 - 75	45	Pre-Filter	R/W
76	164	Post-filter	R/W
77	27	Trigger Config	R/W
78	24	GPIO Config	R/W
79	8	Event Report	R/W
80	22	Module Parameter Error Status	Read only

RFID Capabilities

Table 9 Reader Capability

Reader Capability	Data Type	Offset	Comments
ReaderID	Byte[32]	0	Reader ID in (ASCII String)
FirmwareVersion	Byte[32]	32	Firmware version (ASCII String)
ModelName	Byte[32]	64	Model name (ASCII String)
CountryCode	Word	96	Country of the code
CommunicationStandard	Word	98	Communication standard
NoOfRFModes	Word	100	No of supported modes
MinPower	Word	102	Minimum power supported
MaxPower	Word	104	Maximum power supported
StepPower	Word	106	Step size of power
NoOfAntennas	Byte	108	No of antenna supported
NoOfGPI	Byte	109	No of GPI
NoOfGPO	Byte	110	No of GPO
MaxNoOfPreFilters	Byte	111	No of Pre-Filters supported by the Reader
ReaderIDLen	Byte	112	Reader ID string length
FirmwareVersionLen	Byte	113	Firmware version string length
ModelNameLen	Byte	114	Model name string length
Record Parameter Index 1 **Length of Record Parameter: 115 (Bytes)**			

Set/Get Parameters

Profile

Table 10 Set Active Profile

Set Active Profile	Data Type	Offset	Comments
Profile Index	Word	0	Profile Index (can be obtained from FX9600 Web pages)
Record Parameter Index 2 **Length of Record Parameter: 2 (Bytes)**			

Table 11 Get Active Profile Name

Profile Name	Data Type	Offset	Comments
ProfileNameLen	Word	0	Length of Profile Name String
ProfileName	Byte[64]	2	Profile Name String
Record Parameter Index 2 **Length of Record Parameter: 66 (Bytes)**			

Table 12 Get Profile Name

Profile Name	Data Type	Offset	Comments
ProfileNameLen	Word	0	Length of Profile Name String
ProfileName	Byte[64]	2	Profile Name String
Record Parameter Index 3 - 34 Length of Record Parameter: 66 (Bytes)			

Antenna Config

Table 13 Antenna Config

Antenna Config	Data Type	Offset	Comments	Default
PowerLevel	Word	0	1000 - 3000 (dBm)	3000
TagPopulation	Word	2	0 - 32767	100
TariValue	Word	4	Depends on RF Mode	0
RFModeIndex	Byte	6	0 - 39	0
Target	Byte	7	State A=0, State B=1, AB FLIP=2	0
Session	Byte	8	S0=0, S1=1 , S2=2, S3=3	1
Sel	Byte	9	Asserted=0, Deasserted=1, SL ALL=2	2
AntennaId	Byte	10	0-8, 0=All Antenna	0
Antenna Config Parameter can be Read from or Written to FX9600 Reader. Record Parameter Index 35 - 42 Length of Record Parameter: 11 (Bytes)				

Delete Pre-Filter

Table 14 Delete Pre-Filter

Delete Pre-Filter	Data Type	Offset	Comments
Pre-Filter ID	Byte	0	Delete Pre-Filter (1 -32)
Record Parameter Index 43 Length of Record Parameter: 1 (Bytes)			

Pre-Filter Config

Table 15 Pre-Filter Config

Pre-Filter	Data Type	Offset	Comments
TagPatternBitCount	Word	0	Length tag pattern in number of bits
BitOffset	Word	2	Bit offset from where tag pattern begins
Tagpattern	Byte[36]	4	Tag pattern
Action	Byte	40	0 - 7 Refer to Select Command Section under C1G2 Specification (https://www.gs1.org/sites/default/files/docs/epc/uhfc1g2_1_2_0-standard-20080511.pdf.)
Target	Byte	41	SL=0, S0=1, S1=2, S2=3, S3=4
MemoryBank	Byte	42	EPC=1, TID=2, USER=3
AntennaID	Byte	43	0-8, 0=All Antenna
Pre-FilterID	Byte	44	1 to 32

Record Parameter Index 44 – 75 (1 for each 32 Pre-Filter IDs)
Length of Record Parameter: 45 (Bytes)

Post-Filter Config

Table 16 Post-Filter Config

Post-Filter	Data Type	Offset	Comments
TagPatternA	Struct	0	Pattern A parameters.
Bitoffset	Word	0	Bit offset from where tag pattern begins.
TagPatternBitCount	Word	2	Length tag pattern in number of bits.
TagMaskBitCount	Word	4	Length tag mask in number of bits.
TagPattern	Byte[36]	6	Tag pattern.
TagMask	Byte[36]	42	Tag mask.
MemoryBank	Byte	78	Reserved = 0 EPC = 1 TID = 2 USER = 3
pad	Byte	79	
TagPatternB	Struct	80	Pattern B parameters.
Bitoffset	Word	80	Bit offset from where tag pattern begins.
TagPatternBitCount	Word	82	Length tag pattern in number of bits.

Record Parameter Index 76
Length of Record Parameter: 164 (Bytes)

NOTE:
Post Filter applied takes affect when the Inventory operation is performed with EnablePostFilter set to a non-zero value in the Inventory Command parameter.

Table 16 Post-Filter Config (Continued)

Post-Filter	Data Type	Offset	Comments
TagMaskBitCount	Word	84	Length tag mask in number of bits.
TagPattern	Byte[36]	86	Tag pattern.
TagMask	Byte[36]	122	Tag mask.
MemoryBank	Byte	158	Reserved = 0 EPC = 1 TID = 2 USER = 3
pad	Byte	159	
MatchPattern	Byte	160	A_AND_B = 0 NOTA_AND_B = 1 NOTA_AND_NOTB = 2 A_AND_NOTB = 3
PeakRSSILowerLimit	Byte	161	RSSI Filter Lower Limit.
PeakRSSIUpperLimit	Byte	162	RSSI Filter Higher Limit.
PeakRSSIMatchRange	Byte	163	WITHIN_RANGE = 0 OUTSIDE_RANGE = 1 GREATER_THAN_LOWER_LIMIT = 2 LOWER_THAN_UPPER_LIMIT = 3
Record Parameter Index 76 **Length of Record Parameter: 164 (Bytes)** **NOTE:** **Post Filter applied takes affect when the Inventory operation is performed with EnablePostFilter set to a non-zero value in the Inventory Command parameter.**			

Trigger Config

Table 17 Trigger Config

Trigger Config	Data Type	Offset	Comments	Default
PeriodicTime	DWord	0	Used when start trigger is periodic (millisecond).	0
StartDelay	DWord	4	Used when start trigger is periodic (millisecond).	0
Timer	DWord	8	When stop trigger type is Duration or as Timeout (millisecond).	0
PeriodicReportDuration	DWord	12	-1 = Disable Periodic reporting 0 = Report at the end of inventory n = Report once every n second	-1
TagEventTimeOut	Word	16	Tag Event Timeout (millisecond).	0
Record Parameter Index 77 **Length of Record Parameter: 27 (Bytes)**				

Table 17 Trigger Config (Continued)

Trigger Config	Data Type	Offset	Comments	Default
Count	Word	18	Used when stop trigger is tag observations or n attempts.	1
StartTriggerType	Byte	20	0 = Immediate 1 = Periodic 2 = GPI	0
StartGPIPortNumber	Byte	21	GPI pin number	1
StartEventType	Byte	22	0 = High to low 1 = Low to high (GPI pin state transition)	0
StopTriggerType	Byte	23	0 = Immediate 1 = Duration 2 = GPI with timeout 3 = Tag observation 4 = n attempts	0
StopGPIPortNumber	Byte	24	GPI PIN number.	1
StopEventType	Byte	25	0 = High to low 1 = Low to high (GPI pin state transition)	0
EnableTagEvent	Byte	26	Enables tag Events (like Report TagEvents in the Extended standard and Long Inventory Response modules).	0
Record Parameter Index 77 **Length of Record Parameter: 27 (Bytes)**				

GPIO Config

Table 18 GPIO Config

GPIO Config	Data Type	Offset	Comments
ConfigGPI1	Struct	0	
Enable	Byte	0	R: Shows status W: Enable GPIO events
State	Byte	1	R: Shows state W: Ignore
Set	Byte	2	W: 0 ignore this update W: 1 set this update
Pad	Byte	3	W: Ignore
ConfigGPI2	Struct	4	
R: Implies while Reading **W: Implies while Writing** **Record Parameter Index 78** **Length of Record Parameter: 24 Bytes (12 words)**			

Table 18 GPIO Config (Continued)

GPIO Config	Data Type	Offset	Comments
Enable	Byte	4	R: Shows status W: Enable GPIO events
State	Byte	5	R: Shows state W: Ignore
Set	Byte	6	W: 0 ignore this update W: 1 set this update
Pad	Byte	7	W: Ignore
ConfigGPI3	Struct	8	
Enable	Byte	8	R: Shows status W: Enable GPIO events
State	Byte	9	R: Shows state W: Ignore
Set	Byte	10	W: 0 ignore this update W: 1 set this update
Pad	Byte	11	W: Ignore
ConfigGPI4	Struct	12	
Enable	Byte	12	R: Shows status W: Enable GPIO events
State	Byte	13	R: Shows state W: Ignore
Set	Byte	14	W: 0 ignore this update W: 1 set this update
Pad	Byte	15	W: Ignore
ConfigGPO1	Struct	16	
State	Byte	16	R: Shows current state W: New state
Set	Byte	17	W: 0 ignore this update W: 1 set the new state
ConfigGPO2	Struct	18	
State	Byte	18	R: Shows current state W: New state
Set	Byte	19	W: 0 ignore this update W: 1 set the new state
ConfigGPO3	Struct	20	
State	Byte	20	R: Shows current state W: New state

R: Implies while Reading
W: Implies while Writing

Record Parameter Index 78
Length of Record Parameter: 24 Bytes (12 words)

Table 18 GPIO Config (Continued)

GPIO Config	Data Type	Offset	Comments
Set	Byte	21	W: 0 ignore this update W: 1 set the new state
ConfigGPO4	Struct	22	
State	Byte	22	R: Shows current state W: New state
Set	Byte	23	W: 0 ignore this update W: 1 set the new state
R: Implies while Reading **W: Implies while Writing** **Record Parameter Index 78** **Length of Record Parameter: 24 Bytes (12 words)**			

Event Report

EventMask is reported in Inventory Response and Access Command Response data, periodically. See Inventory Response Parameter Structure on page 31 Access Response Parameter Structure on page 41.

Each bit in EventMask is used to identify the corresponding EventType (see Event Mask on page 55). For example: EventMask value 4 corresponds to bit number 3 (the bit number set). In Table 21 on page 55, bit number 3 corresponds to Antenna_Event and its EventType value is 3.

Set the Event Report parameter to FX9600 with required EventType parameter for which detailed report needs to be read. For example, EventMask value 4 corresponds to bit number 3 being set. By Referring to Event Mask Table, bit number 3 corresponds to Antenna Event and its EventType value is 3.

The Get Event Report parameter from the FX9600 Reader, and the values corresponding to the EventType set above, are reported if the event occurred on the FX9600 Reader.

Table 19 Event Report

Event Report	Data Type	Offset	Comments
EventType	Byte	0	W: Select appropriate event
GpiPortNumber	Byte	1	R: Event reported on GPI PIN Number
GpiValue	Byte	2	R: GPI state
AntennaID	Byte	3	R: Event Reported on Antenna ID
AntennaStatus	Byte	4	R: Antenna Status
TemperatureSource	Byte	5	R: Source of Temperature Event
TemperatureAlarm	Byte	6	R: Temperature Alarm Level
TemperatureValue	Byte	7	Temperature Value
Record Parameter Index 79 **Length of Record Parameter: 8 (Bytes)**			

Module Parameter Error Status

Module parameters are set during the establishment of the connection described in Module Parameters Configuration. This record parameter can be read to check if there were any errors reported during the configuration.

NOTE: Module properties are set on Inventory Command and the Access Command module. Select the correct module name in the ID parameter of the RDREC function block.

Table 20 Module Parameter Error Status

Module Parameter Error Status	Data Type	Offset	Comments
ActiveProfileConfigStatus	Word	0	Profile Config Error Status
AntennaConfigStatus_0_1	Word	2	Antenna 0 (all) or 1 Config Error Status
AntennaConfigStatus_2	Word	4	Antenna 2 Config Error Status
AntennaConfigStatus_3	Word	6	Antenna 3 Config Error Status
AntennaConfigStatus_4	Word	8	Antenna 4 Config Error Status
AntennaConfigStatus_5	Word	10	Antenna 5 Config Error Status
AntennaConfigStatus_6	Word	12	Antenna 6 Config Error Status
AntennaConfigStatus_7	Word	14	Antenna 7 Config Error Status
AntennaConfigStatus_8	Word	16	Antenna 8 Config Error Status
PreFilterConfigStatus	Word	18	Pre-Filter ID 1 Config Error Status
TriggerConfigStatus	Word	20	Trigger Config Error Status
Record Parameter Index 80 **Length of Record Parameter: 22 (Bytes)**			

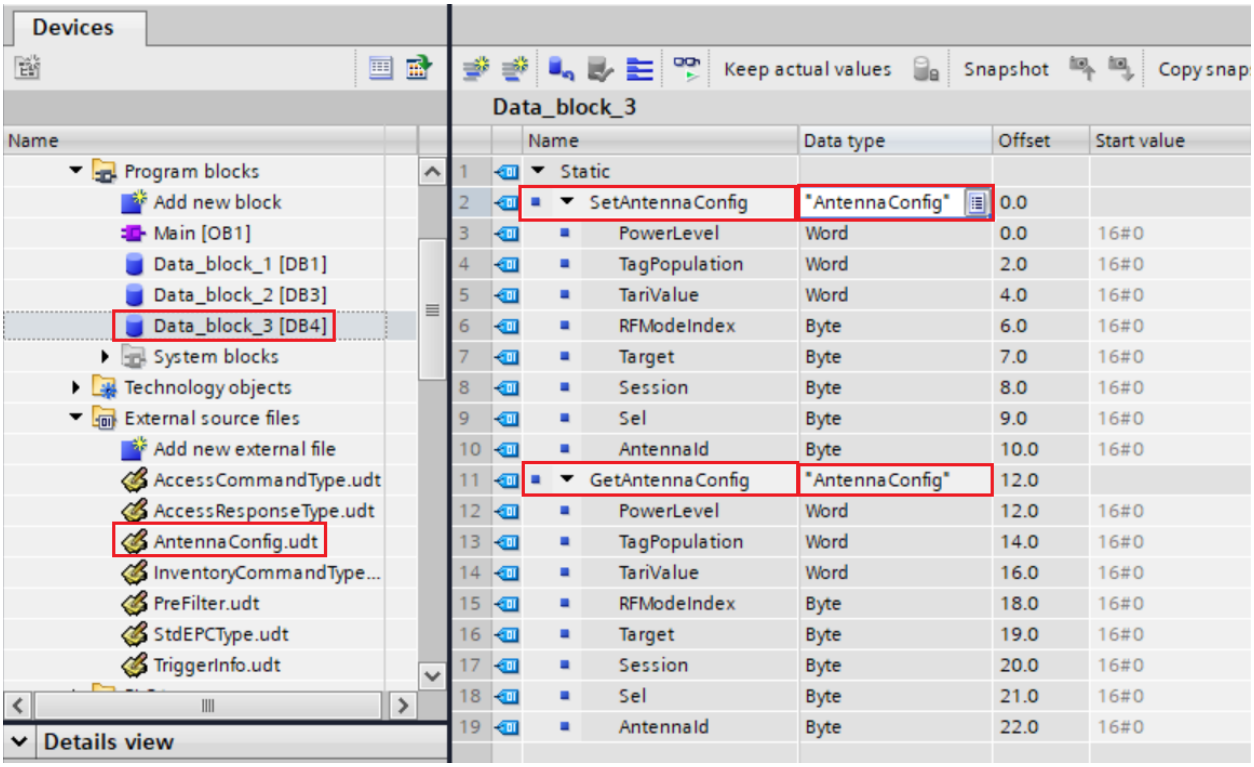
Sample Ladder Diagram to Set/Get RFID Parameters

Figure 25 shows and example Antenna Config parameter.

To create the data type, add the external AntennaConfig.udt data type file supplied by Zebra; right click and select Generate from blocks from source.

Create the data block and add variables for the Set and Get Antenna Config operations as shown in the below.

Figure 25 Antenna Config Parameter



Ladder Diagram to Read Parameter from the FX9600 Reader

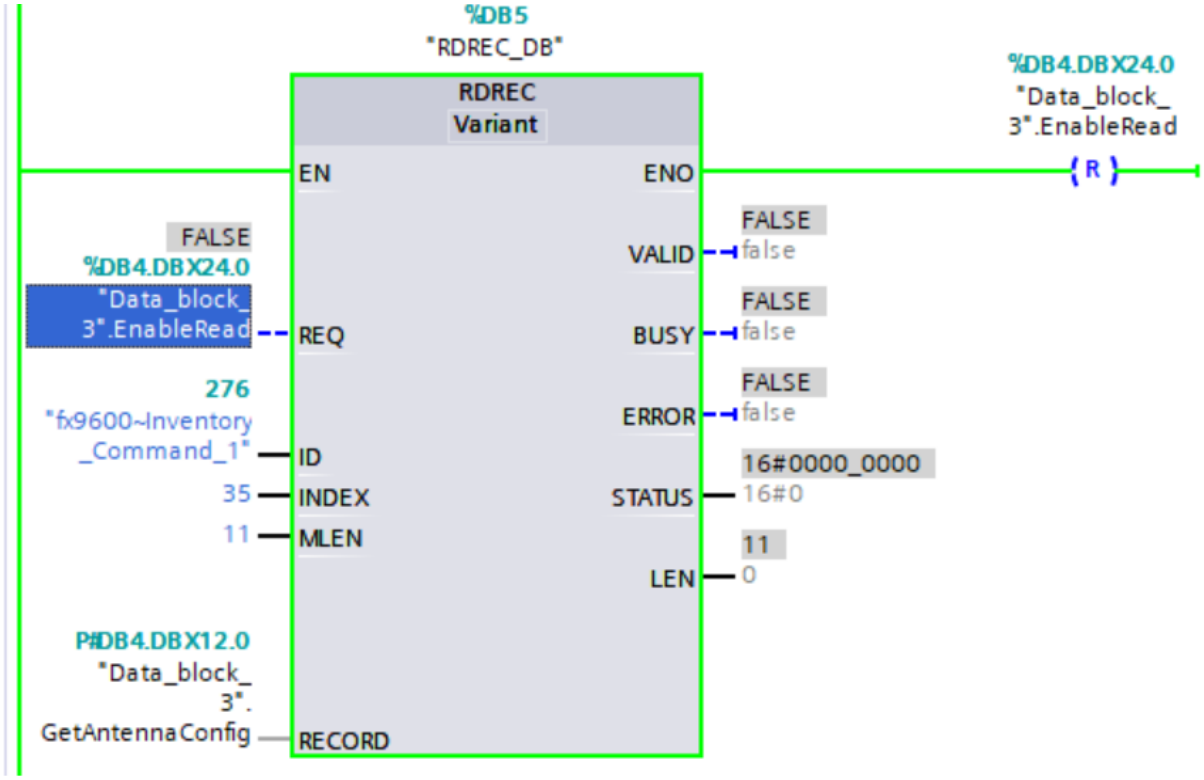
The RDREC function block is used to read parameters from PROFINET devices.

Create the Ladder diagram shown in Figure 26 to read the Antenna Config parameter from the FX9600.

The ID parameter can be set to either the Inventory Command or Access Command module. From the RFID Parameters on page 43, Index is set to 35 (Antenna ID 0), the MLEN parameter is set to 11 (length in bytes for Antenna Config parameter) and Record parameter is set to the Data_block_3.GetAntennaConfig variable as created in previous step.

When the Data_block_3.EnableRead is set to True, RDREC function block reads the parameter from the FX9600 Reader and copies the contents to the Data_block_3.GetAntennaConfig variable and Data_block_3. EnableRead is set to False by Reset function.

Figure 26 Ladder Diagram Example - Read Parameter from the FX9600 Reader



Ladder Diagram to Write Parameter to PROFINET Devices

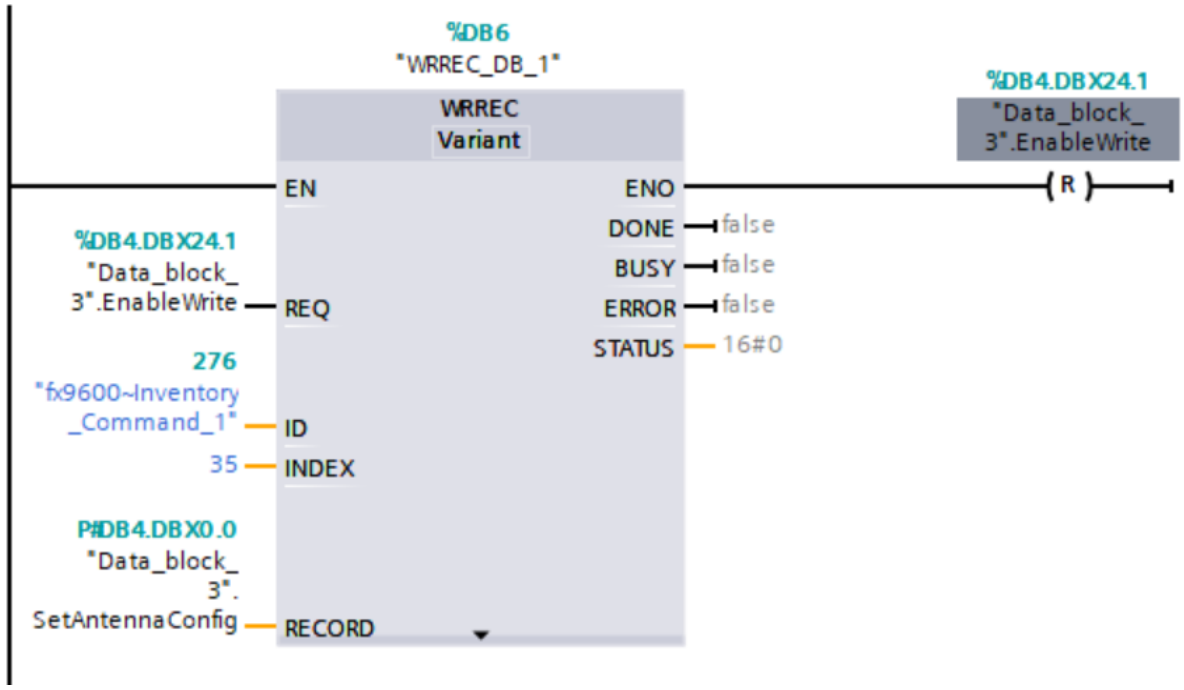
WRREC function block is used to write parameters to the PROFINET devices.

Create the Ladder diagram shown Figure 27 to write the Antenna Config parameter from FX9600

ID parameter can set to either Inventory Command or Access Command module. From the RFID Parameters on page 43 Index is set to 35 (Antenna ID 0) and Record parameter is set to the Data_block_3.SetAntennaConfig variable created above.

When the Data_block_3.EnableWrite is set to True, contents of Data_block_3.SetAntennaConfig variable is sent to the FX9600 Reader by the WRREC function block and Data_block_3. EnableWrite is set to False by the Reset function.

Figure 27 Ladder Diagram Example - Write Parameter from the FX9600 Reader



Alarms, Event Mask and Error Codes

Alarms

Antenna Disconnect Alarm

The FX9600 Reader reports an Antenna Disconnect alarm when one of the antennas connected to the device is disconnected, faulty or loose.

The alarm returns to normal when the required antenna is connected firmly back to device.

RFID Reader Exception

When the RFID radio in the FX9600 Reader malfunctions, an RFID Reader Exception alarm is reported.

The FX9600 Reader must be power cycled (restarted) to recover from this fault to continue with the operation.

RFID Disconnect

The FX9600 Reader reports RFID Disconnect alarm when the RFID radio malfunctions.

The Reader must be power cycled (restarted) to recover from this fault to continue with the operation.

Event Mask

Event Mask Is reported as part of Inventory Response and Access Command Response data.

Each bit in the Event Mask corresponds to an Event. EventType value could be used to fetch additional details from the FX9600 by using the Event Report RFID parameter.

Table 21 Event Mask Descriptions

Event Mask Bit Number	Event Name	EventType
0	GPI_EVENT	0
1	TAG_DATA_EVENT	1
2	BUFFER_FULL_WARNING_EVENT	2
3	ANTENNA_EVENT	3
4	INVENTORY_START_EVENT	4

Table 21 Event Mask Descriptions

Event Mask Bit Number	Event Name	EventType
5	INVENTORY_STOP_EVENT	5
6	ACCESS_START_EVENT	6
7	ACCESS_STOP_EVENT	7
8	DISCONNECTION_EVENT	8
9	BUFFER_FULL_EVENT	9
10	NXP_EAS_ALARM_EVENT	10
11	READER_EXCEPTION_EVENT	11
13	DEBUG_INFO_EVENT	13
14	TEMPERATURE_ALARM_EVENT	14

RFID Error Codes

Table 22 RFID Error Codes

Error Code	Error Type
0	RFID_API_SUCCESS
1	RFID_API_COMMAND_TIMEOUT
2	RFID_API_PARAM_ERROR
3	RFID_API_PARAM_OUT_OF_RANGE
4	RFID_API_CANNOT_ALLOC_MEM
5	RFID_API_UNKNOWN_ERROR
6	RFID_API_INVALID_HANDLE
7	RFID_API_BUFFER_TOO_SMALL
8	RFID_READER_FUNCTION_UNSUPPORTED
9	RFID_RECONNECT_FAILED
10	RFID_API_DATA_NOT_INITIALISED
11	RFID_API_ZONE_ID_ALREADY_EXITS
12	RFID_API_ZONE_ID_NOT_FOUND
100	RFID_COMM_OPEN_ERROR
101	RFID_COMM_CONNECTION_ALREADY_EXISTS
102	RFID_COMM_RESOLVE_ERROR
103	RFID_COMM_SEND_ERROR
104	RFID_COMM_RECV_ERROR
105	RFID_COMM_NO_CONNECTION
106	RFID_INVALID_SOCKET
107	RFID_READER_REGION_NOT_CONFIGURED
108	RFID_READER_REINITIALIZING

Table 22 RFID Error Codes (Continued)

Error Code	Error Type
109	RFID_SECURE_CONNECTION_ERROR
110	RFID_ROOT_SECURITY_CERTIFICATE_ERROR
111	RFID_HOST_SECURITY_CERTIFICATE_ERROR
112	RFID_HOST_SECURITY_KEY_ERROR
200	RFID_CONFIG_GET_FAILED
201	RFID_CONFIG_SET_FAILED
202	RFID_CONFIG_NOT_SUPPORTED
300	RFID_CAP_NOT_SUPPORTED
301	RFID_CAP_GET_FAILED
400	RFID_FILTER_NO_FILTER
401	RFID_FILTER_INVALID_INDEX
402	RFID_FILTER_MAX_FILTERS_EXCEEDED
403	RFID_NO_READ_TAGS
404	RFID_NO_REPORTED_EVENTS
405	RFID_INVENTORY_MAX_TAGS_EXCEEDED
406	RFID_INVENTORY_IN_PROGRESS
407	RFID_NO_INVENTORY_IN_PROGRESS
420	RFID_TAG_LOCATING_IN_PROGRESS
421	RFID_NO_TAG_LOCATING_IN_PROGRESS
422	RFID_NXP_EAS_SCAN_IN_PROGRESS
423	RFID_NO_NXP_EAS_SCAN_IN_PROGRESS
500	RFID_ACCESS_IN_PROGRESS
501	RFID_NO_ACCESS_IN_PROGRESS
502	RFID_ACCESS_TAG_READ_FAILED
503	RFID_ACCESS_TAG_WRITE_FAILED
504	RFID_ACCESS_TAG_LOCK_FAILED
505	RFID_ACCESS_TAG_KILL_FAILED
506	RFID_ACCESS_TAG_BLOCK_ERASE_FAILED
507	RFID_ACCESS_TAG_BLOCK_WRITE_FAILED
508	RFID_ACCESS_TAG_NOT_FOUND
510	RFID_ACCESS_SEQUENCE_NOT_INITIALIZED
511	RFID_ACCESS_SEQUENCE_EMPTY
512	RFID_ACCESS_SEQUENCE_IN_USE
513	RFID_ACCESS_SEQUENCE_MAX_OP_EXCEEDED
514	RFID_ACCESS_TAG_RECOMMISSION_FAILED
515	RFID_ACCESS_TAG_BLOCK_PERMALOCK_FAILED
516	RFID_ACCESS_NXP_TAG_SET_EAS_FAILED

Table 22 RFID Error Codes (Continued)

Error Code	Error Type
517	RFID_ACCESS_NXP_TAG_READ_PROTECT_FAILED
518	RFID_ACCESS_FUJITSU_CHANGE_WORDLOCK_FAILED
519	RFID_ACCESS_FUJITSU_CHANGE_BLOCKLOCK_FAILED
520	RFID_ACCESS_FUJITSU_READ_BLOCKLOCK_FAILED
521	RFID_ACCESS_FUJITSU_BURST_WRITE_FAILED
522	RFID_ACCESS_FUJITSU_BURST_ERASE_FAILED
523	RFID_ACCESS_FUJITSU_CHANGE_BLOCK_OR_AREA_GROUPPASSWORD_FAILED
524	RFID_ACCESS_FUJITSU_AREA_READLOCK_FAILED
525	RFID_ACCESS_FUJITSU_AREA_WRITELOCK_FAILED
526	RFID_ACCESS_FUJITSU_AREA_WRITELOCK_WOPASSWORD_FAILED
527	RFID_ACCESS_NXP_CHANGE_CONFIG_FAILED
528	RFID_ACCESS_IMPINJ_QT_READ_FAILED
529	RFID_ACCESS_IMPINJ_QT_WRITE_FAILED
530	RFID_ACCESS_G2V2_AUTHENTICATE_FAILED
531	RFID_ACCESS_G2V2_READBUFFER_FAILED
532	RFID_ACCESS_G2V2_UNTRACEABLE_FAILED
533	RFID_ACCESS_G2V2_CRYPTO_FAILED
601	RFID_RM_INVALID_USERNAME_PASSWORD
602	RFID_RM_NO_UPDATION_IN_PROGRESS
603	RFID_RM_UPDATION_IN_PROGRESS
604	RFID_RM_COMMAND_FAILED
605	RFID_NXP_BRANDID_CHECK_IN_PROGRESS
606	RFID_NO_RF_SURVEY_OPERATION_IN_PROGRESS
607	RFID_RFSURVEY_IN_PROGRESS
700	RFID_INVALID_ERROR_CODE

Appendix

Pre-Filter Settings Example

The FX9600 Reader supports 32 Pre-Filters. Pre-Filter 1 is provided through Module Parameters Configuration. Remaining Pre-Filters from 2 - 32 can only be accessed using PROFINET Acyclic Read/Write commands.

Report Tags That Match Pattern

In Table 23 and Table 24, Antenna config parameter values and Pre-Filter Config parameter values are shown to apply a filter to the report only tag EPCs that begins with 2F 22.

Table 23 Antenna Config Parameters - Report Tags

Antenna Config	Data Type	Value	Comments
PowerLevel	Word	1000 - 3000	Choose required power
TagPopulation	Word	100	Assuming 100 tags in field
TariValue	Word	0	Default
RFModeIndex	Byte	0	Default
Target	**Byte**	**0 (State A)**	**State A**
Session	**Byte**	**S1**	**Session S1**
Sel	**Byte**	**2**	**Select SL_ALL**
AntennaId	Byte	0 (All Antenna)	All antennae

Table 24 Pre-Filter Config Parameters - Report Tags

Pre-Filter	Data Type	Value	Comments
TagPatternBitCount	Word	16	2F 22 is 2 bytes of 16 bits
BitOffset	Word	32	Tag EPC has PC (2 bytes) and CRC (2 bytes) at the beginning and the Pattern 2F 22 only starts after these 4 Bytes. Therefore, offset is 32.
Tagpattern	Byte[36]	2F, 22, 00, 00, . .	Tag EPC begins with pattern 2F, 22
Action	**Byte**	**0**	**0 (INV_A_NOT_INV_B refer 0 – 7 Refer Select Command Section under C1G2 Specification) to Report tags with this pattern.**

Table 24 Pre-Filter Config Parameters - Report Tags (Continued)

Pre-Filter	Data Type	Value	Comments
Target	**Byte**	**2**	**As Session S1 is selected in Antenna Config**
MemoryBank	Byte	1	EPC Memory Bank
AntennaID	Byte	0	Apply this filter on all antennae
PreFilterID	Byte	2	Using Pre-Filter Id 2

Ignore Tags That Match Pattern

In Table 25 and Table 26, Antenna config parameter values and Pre-Filter Config parameter values are shown to apply a filter to **not report** tag EPCs that begins with 8D F0.

Table 25 Antenna Config Parameters - Ignore Tags

Antenna Config	Data Type	Value	Comments
PowerLevel	Word	1000 - 3000	Choose required power
TagPopulation	Word	100	Assuming 100 tags in field
TariValue	Word	0	Default
RFModeIndex	Byte	0	Default
Target	**Byte**	**0**	**State A**
Session	**Byte**	**S1**	**Session S1**
Sel	**Byte**	**2**	**Select SL_ALL**
AntennaId	Byte	0 (All Antenna)	All Antenna

Table 26 Pre-Filter Config Parameters - Report Tags

Pre-Filter	Data Type	Value	Comments
TagPatternBitCount	Word	16	8D F0 is 2 bytes of 16 bits
BitOffset	Word	32	TAG EPC has PC (2 bytes) and CRC (2 bytes) at the beginning and the Pattern 2F 22 only starts after these 4 bytes. This is why offset is 32
Tagpattern	Byte[36]	8D F0 00 00 . .	Tag EPC begins with pattern 8D, F0
Action	**Byte**	**4**	**4 (INV_B_NOT_INV_A refer 0 – 7 Refer Select Command Section under C1G2 Specification) to Report tags with this Pattern.**
Target	**Byte**	**2**	**As Session S1 is selected in Antenna Config**
MemoryBank	Byte	1	EPC Memory Bank
AntennaID	Byte	0	Apply this filter on all Antenna
PreFilterID	Byte	2	Using Pre-Filter Id 2

NOTE: The user can customize this example by keeping the parameter values in bold text as shown and the remaining parameter values can be changed. For example, choose a memory bank other than EPC, select a particular antenna or change the Power or RF Mode Index.

The applied Pre-Filter takes effect when the Inventory operation is performed with EnablePreFilter set to a non-zero value in the Inventory Command parameter.

Reading Only Unique EPC Tags

With the default trigger setting, the FX9600 Reader reports each time tag EPC is read. The tag which is closer to the antenna is likely read a greater number of times. However, there are situations where it is necessary to read all the tags in visibility of the antenna only once.

The FX9600 Reader can be configured to report once every few seconds, minutes or hours using the Trigger Config. Using this feature, it is possible to read only the unique tags.

Consider a scenario where RFID tagged items are moving on a conveyer belt and are in the field of antenna visibility for about 30 seconds. In this scenario it is possible to set the trigger settings to report all tags only once, every 30 seconds. Since the tagged item is moving and goes out of antenna visibility within 30 seconds, each tag is reported only once.

Table 27 Trigger Config - Read Only Unique EPC Tags

Trigger Config	Data Type	Offset	Default Values	Comments
PeriodicTime	DWord	0	0	Used when start trigger is "periodic" (millisecond)
StartDelay	DWord	4	0	Used when start trigger is "periodic" (millisecond)
Timer	DWord	8	0	When stop trigger type is "Duration" or as "Timeout" (millisecond)
PeriodicReportDuration	**DWord**	**12**	**30**	**30 = report once every 30 second**
TagEventTimeOut	Word	16	0	Tag Event Timeout (millisecond)
Count	Word	18	0	Used when stop trigger is "tag Observations" or "n attempts"
StartTriggerType	Byte	20	0	0 = Immediate
StartGPIPortNumber	Byte	21	0	GPI pin number
StartEventType	Byte	22	0	0 = High to low
StopTriggerType	Byte	23	0	0 = Immediate
StopGPIPortNumber	Byte	24	0	GPI pin number
StopEventType	Byte	25	0	0 = High to low
EnableTagEvent	Byte	26	0	Enables tag events (Reports TagEvents in the extended standard and long Inventory Response modules)

Memory Estimation for Cyclic Data Exchange

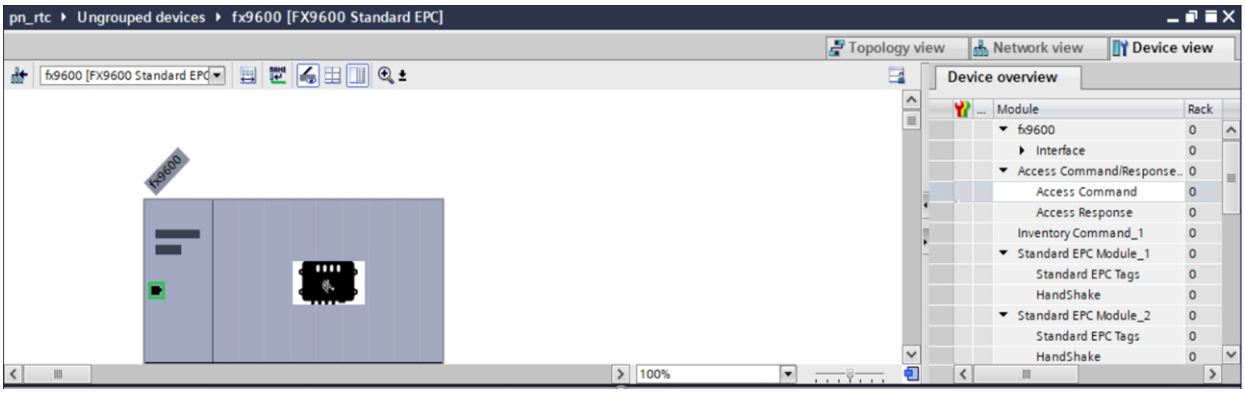
Table 28 describes the sizes of various modules and sub-modules available with the FX9600 device for cyclic data exchange with PLC.

Table 28 Module and Sub-module Sizes

Module Name	Sub Module Name	Size (Bytes)	Data Direction
Inventory Command		8	PLC to Device
Standard EPC MODULE	Standard EPC Tags	33	Device to PLC
	Handshake	1	PLC to Device
Extended Standard EPC MODULE	Extended Standard EPC Tags	44	Device to PLC
	Handshake	1	PLC to Device
Long EPC MODULE	Long EPC Tags	85	Device to PLC
	Handshake	1	PLC to Device
Extended Long EPC MODULE	Extended Long EPC Tags	96	Device to PLC
	Handshake	1	PLC to Device
Access Command/Response	Access Command	152	PLC to Device
	Access Response	155	Device to PLC

To estimate memory requirements on PLC, see the example in Figure 28 where Access Command/Response, Inventory Command and 2 Standard EPC Modules are configured.

Figure 28 Memory Estimate Example



Total Output Memory (Q Addresses) can be calculated as follows:

Total Q Address = Access Command + Inventory Command + 2 * Handshake

Total Q Address = 152 + 8 + 2 * 1

Total Q Address = 162

Total Input Memory (I Addresses) can be calculated as follows:

Total I Address = Access Response + 2 * Standard EPC Tags

Total I Address = 155 + 2 * 33

Total I Address = 221

Ensure that the required total Q Address and I Address are free and available for use on the PLC to use required module configuration.