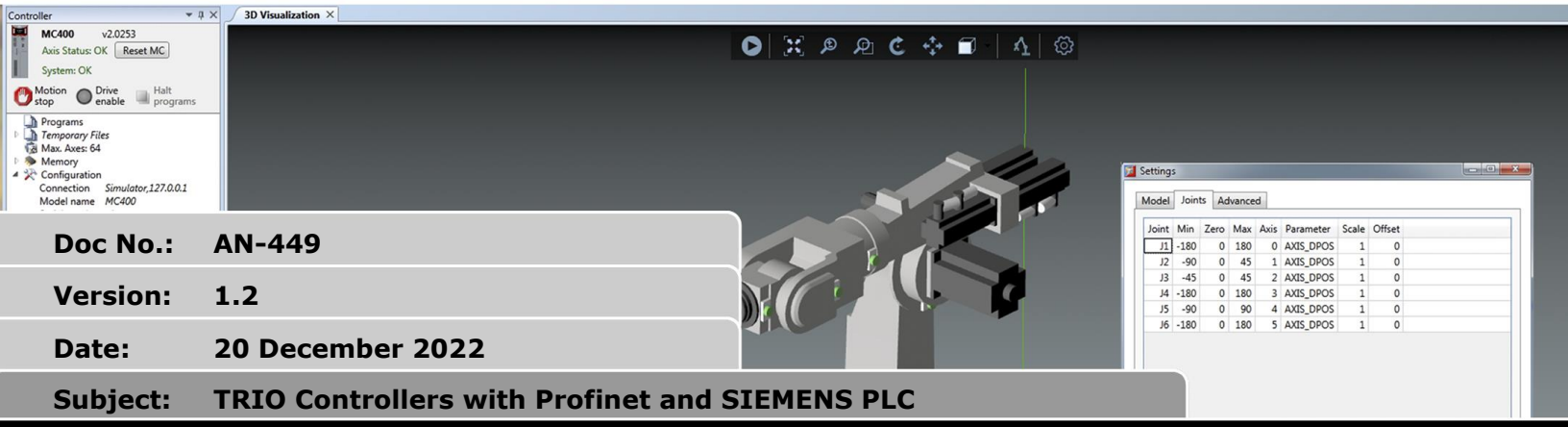
Trio Motion Technology Ltd.,
Shannon Way,
Tewkesbury,
Gloucestershire,
GL20 8ND
United Kingdom
Tel : +44 (0)1684-292333

Trio Motion Technology LLC.,
187 Northpointe Blvd,
Suite 105 Freeport,
PA 16229,
USA
Tel : +1 724 472 4100

Trio Motion Technology (Shanghai) Co. Ltd.,
A1104 Yunding International Commercial Plaza,
800 Chengshan Rd,
Pudong New Area,
Shanghai,
CHINA
Tel : +86 21 587 976 59

Trio Motion Technology,
Teerth Business Center,
3rd Floor, Unit No. 7, Block EL - 15,
MIDC, Bhosari,
Pimpri-Chinchwad,
Pune, 411026, INDIA
Tel : +91 827 506 5446



Doc No.: AN-449

Version: 1.2

Date: 20 December 2022

Subject: TRIO Controllers with Profinet and SIEMENS PLC

APPLICATION NOTE

1. Introduction

This application notes describes the supported functions in Profinet that is part of TrioBasic command.

Profinet support is part of firmware 2.0319 release.

This application note is written using the following controllers: TRIO F6Nano and SIEMENS CPU 1211C AC/DC/Rly.

2. Requirements

The functionalities described in this Application Notes are part of 2.0319 firmware release and therefore a minimum of firmware version 2.0319 is required.

3. TrioBasic Commands

3.1. PROFINET - Function 0

This function is to define or display module mapping of TRIO controller. To display configured module mapping, use the first 2 parameters only as in Example 1. To reset the entire module map, use the first 3 parameters as in Example 2.

3.1.1. Syntax

```
PROFINET(0, slot [, pn_api, pn_slot, pn_subslot, datasource [, index, type, count, direction [,endian]]])
```

3.1.2. Parameters

Name	Description
slot	Module slot number (default: -1)
pn_api	Application Process Identifier (default: 0)
pn_slot	User slots (1..63). Note: slot 0 is the interface
pn_subslot	Sub slot (1..7)

datasource	-1: delete mapping 0: VR 1: TABLE
index	Start position (i.e., the base address) in data source
type	Size and type of the data sent across the bus 1: signed 8-bit-integer 2: signed 16-bit-integer 3: signed 32-bit-integer
count	Number of datatypes mapped
direction	Direction of data movements from master point of view (in this example, SIEMENS PLC) 0: input (sent by TRIO controller) 1: output
endian	0: default 1: swap

3.1.3. Return Value

-1: OK

0: error

3.1.4. Examples

3.1.4.1. Example 1 - display configured module mapping

```
>>?profinet(0,-1)
0 1 (1) : 1 0 10 2 1 0 1
0 2 (1) : 1 0 11 2 1 1 1
-1
```

3.1.4.2. Example 2 - reset entire module mapping

```
>>profinet(0,-1,-1)
>>?profinet(0,-1)
-1
```

3.1.4.3. Example 3 - configure full module mapping

```
>>'Module mapping configuration - Sending to SIEMENS PLC
>>'function = 0
>>'slot = -1
>>'pn_api = 0
>>'pn_slot = 1
>>'pn_subslot = 1
>>'datasource = 0
>>'index_send = 10
>>'datatype = 2
>>'count = 1
>>'direction = 0
>>'endian = 1
>>profinet(0,-1,0,1,1,0,10,2,1,0,1)
```

```
>>?profinet(0,-1)
0 1 (1) : 0 0 10 2 1 0 1
-1
>>'Module mapping configuration - Receiving from SIEMENS PLC
>>'pn_slot = 2
>>'direction = 1
>>'index = 11
>>profinet(0,-1,0,2,1,0,11,2,1,1,1)
>>?profinet(0,-1)
0 1 (1) : 0 0 10 2 1 0 1
0 2 (1) : 0 0 11 2 1 1 1
-1
```

3.2. PROFINET - Function 1

This function starts the Profinet protocol.

3.2.1. Syntax

PROFINET(1, slot)

3.2.2. Parameters

Name	Description
slot	Module slot number (default: -1) Note: • For TRIO Controller to be visible under Profinet network, run function 1 in an autorun program that gets executed on every power cycle. See 4.2.1 sample program.

3.2.3. Return Value

No return value

3.2.4. Examples

3.2.4.1. Example 1

```
>>profinet(1,-1)
```

3.3. PROFINET - Function 2

This function stops the running Profinet protocol.

3.3.1. Syntax

PROFINET(2, slot)

3.3.2. Parameters

Name	Description
slot	Module slot number (default: -1)

3.3.3. Return Value

No return value

3.3.4. Examples

3.3.4.1. Example 1

```
>>profinet(2,-1)
```

3.4. PROFINET - Function 5

This function defines which messages generated by the Profinet stack are written out to the command line.

3.4.1. Syntax

```
PROFINET(5, slot, level)
```

3.4.2. Parameters

Name	Description
`slot`	Module slot number (default: -1)
`level`	Type of messages to be printed out on command line 0x00: none 0x01: errors 0x02: warning 0x04: info 0x08: debug

3.4.3. Return Value

No return value

3.4.4. Examples

3.4.4.1. Example 1 - print out all information on command line

```
>>profinet(5,-1, $f)
```

3.4.4.2. Example 2 - print out errors only on command line

```
>>profinet(5,-1, 1)
```

3.5. PROFINET - Function 8

This function sets TRIO Controller's Profinet name.

3.5.1. Syntax

```
PROFINET(8, device_name)
```

3.5.2. Parameters

Name	Description
device_name	Configured Profinet device name Note: • underscore and space are not allowed • execute this function in an autorun program that gets executed on every power cycle. See 4.2.1 sample program.

3.5.3. Return Value

No return value

3.5.4. Examples

3.5.4.1. Example 1 - set ProfiNET device name

```
>>profinet(8,"trio-192-168-0-240")
```

3.6. AUTO_PROFINET

Controls the action of the system software on power up. If present, the ProfiNET network is initialised automatically on power up or soft reset (EX). If this is not required, then setting AUTO_ETHERCAT to OFF will prevent the ProfiNET from being set up and it is then up to the programmer to start the ProfiNET network from a BASIC program.

ProfiNET name of the controller will be assigned the default name, i.e.: *trio-<controller name>-<5 digit serial number>*. For example, *trio-f6nano-00260*.

This command should not be used in a TrioBASIC program. You must use it in the special MC_CONFIG script which runs automatically on power up. This parameter is NOT stored in FLASH.

3.6.1. Syntax

```
AUTO_PROFINET=value
```

3.6.2. Values

Value	Description
0	ProfiNET network does not initialise on power up
1	ProfiNET network starts on power up

3.6.3. Examples

3.6.3.1. Example 1 - set ProfiNET network to start on power up

```
'MC_CONFIG script file
AUTO_PROFINET = ON
```

3.6.3.2. Example 2 - prevent ProfiNET network to start on power up

```
'MC_CONFIG script file
AUTO_PROFINET = OFF
```

4. Sample Application

4.1. IP Addresses

Set the TRIO controller and SIEMENS PLC on the same subnet addresses. In this example, Trio controller is set to 192.168.0.240 with subnet mask 255.255.255.0 and SIEMENS PLC to 192.168.0.238 with subnet mask 255.255.255.0.

4.2. Motion Perfect Sample Program

4.2.1. START.BAS (autorun)

```
' Rename to specified in 2nd argument
' Default name: trio-<controller name>-<5 digit serial number>.
' For example, trio-f6nano-00260.
' Underscore and space are not accepted.
'PROFINET(8,"tom-cruise-maverick")
PROFINET(8,"trio-192-168-0-240")

' Start Profinet so it's detectable by SIEMENS PLC
PROFINET(1,-1)

WA(5000)
PRINT "Profinet started after power up in START_PN"
```

4.2.2. MAIN.BAS

```
ON BASICERROR GOTO errorhandling

func = 0 ' 0 = define or display mapping; 1 = start; 2 = stop
slot_= -1 ' Only -1
pn_api = 0 ' Only 0
pn_slot = 1
pn_subslot = 1
datasource = 0 ' -1 = delete; 0 = VR; 1 = TABLE
index_send = 10 ' VR/TABLE index to send to master
index_receive = 11 ' VR/TABLE index to receive from master
datatype = 2 ' 2 = signed 16bit integer
count = 1 ' total data
direction = 0 ' 0 = input (to master); 1 = output (to master)
endian = 1 ' 0 = default; 1 = swap
index_missedheartbeat = 0
index_interval = 1
index_totaltransfer = 2
index_missedpercentage = 3
index_transmissionstarted = 4


' Send to SIEMENS (master)
PROFINET(func,slot_,pn_api,pn_slot,pn_subslot,datasource,index_send,datatype,count,
direction,endian)
WA(1000)

' Receive from SIEMENS (master)
pn_slot = 2
direction = 1 ' 0 = input (to master); 1 = output (to master)
PROFINET(func,slot_,pn_api,pn_slot,pn_subslot,datasource,index_receive,datatype,cou
nt,direction,endian)
WA(1000)


' Reset values
VR(index_send) = 0
VR(index_receive) = 0
VR(index_totaltransfer) = 0
VR(index_missedheartbeat) = 0
```

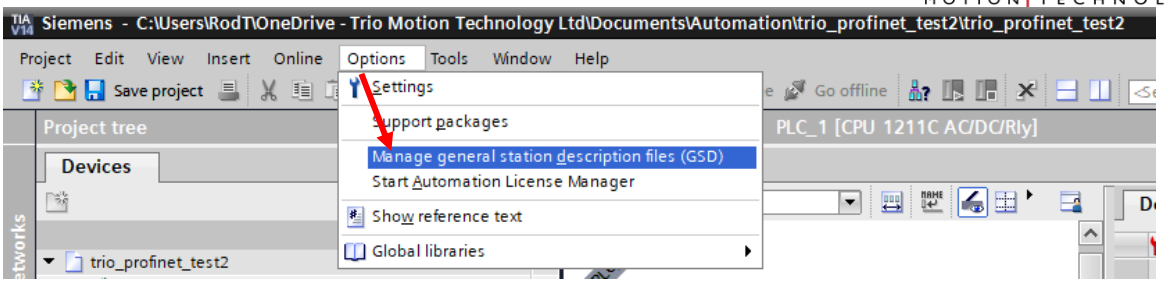
```
VR(index_missedpercentage) = 0
VR(index_transmissionstarted) = 0


' Stop PROFINET
func = 2
PROFINET(func,slot_)
WA(5000) ' 5 sec delay


' Start PROFINET
func = 1
PROFINET(func,slot_)
WA(5000) ' 5 sec delay



' Program
WHILE TRUE
    VR(index_transmissionstarted)=1 ' Indicator that program starts transmitting
    IF(VR(index_totaltransfer) <> 0) THEN
        VR(index_missedpercentage) = VR(index_missedheartbeat)/
VR(index_totaltransfer)*100
    ENDIF
    VR(index_send) = VR(index_send) + 1
    VR(index_totaltransfer) = VR(index_totaltransfer) + 1
    WA(VR(index_interval))

    ' Detect missed heartbeat
    IF (VR(index_send) <> VR(index_receive)) THEN VR(index_missedheartbeat) =
VR(index_missedheartbeat) + 1

    ' Reset heartbeat
    IF VR(index_send) > 100 THEN VR(index_send) = 0
WEND



errorhandling:
IF RUN_ERROR=31 THEN
    VR(index_transmissionstarted)=0
    ' Stop PROFINET
    func = 2
    PROFINET(func,slot_)

    PRINT "total transfers: ";STR(VR(index_totaltransfer),0)
    PRINT "total missed: ";STR(VR(index_missedheartbeat),0);" (";_
        STR((VR(index_missedheartbeat)/VR(index_totaltransfer)*100),2);"%";")"
ELSE
    PRINT "RUN_ERROR = ";STR(RUN_ERROR,0)
ENDIF
```

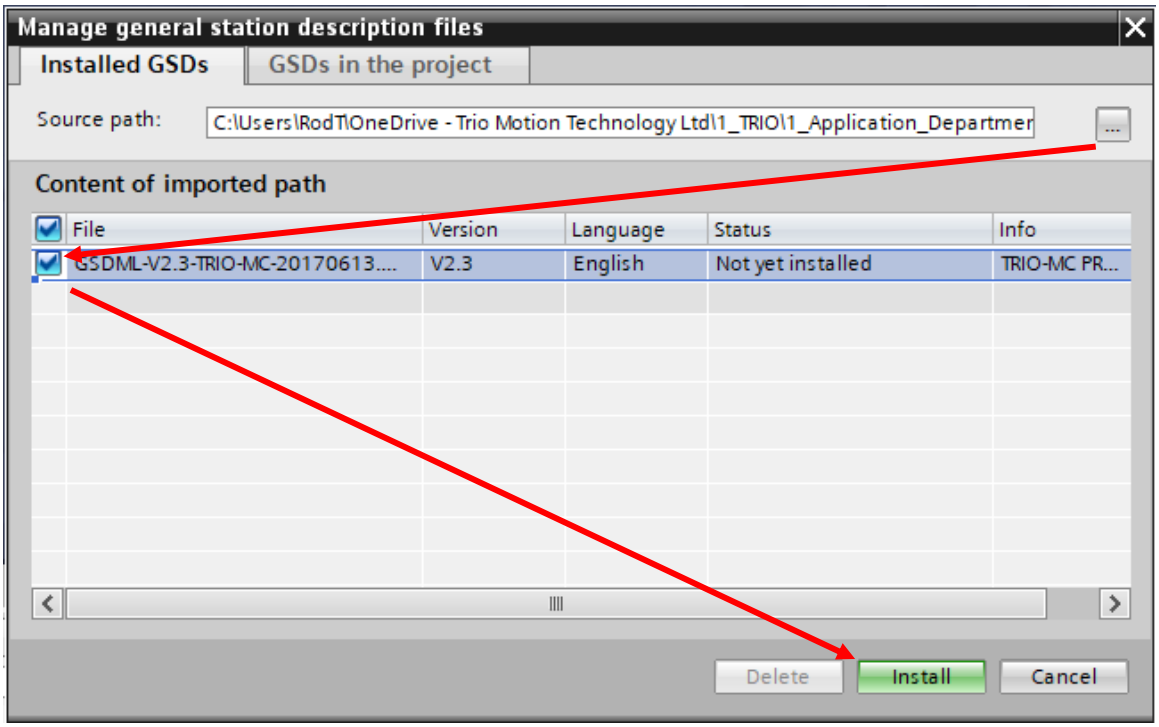
4.3. TIA Portal Sample Program

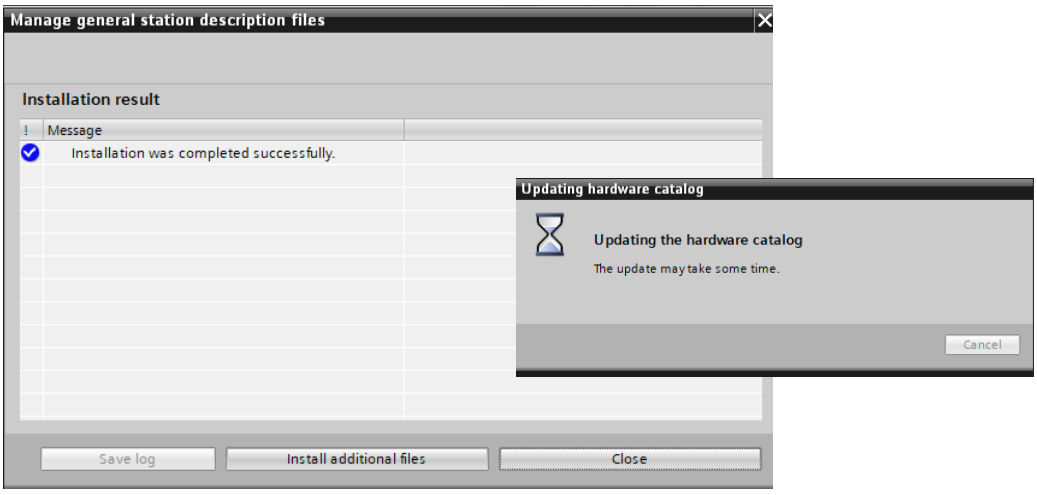
Any P824 application which uses either one servo axis (+/-10V output and encoder input) or 2 servo axes cannot use the MC404-Z as an alternative.

4.3.1. TRIO GSDML Installation

To allow TRIO controllers to appear in TIA Portal hardware catalogue, install the GSDML for Trio controllers as shown below.

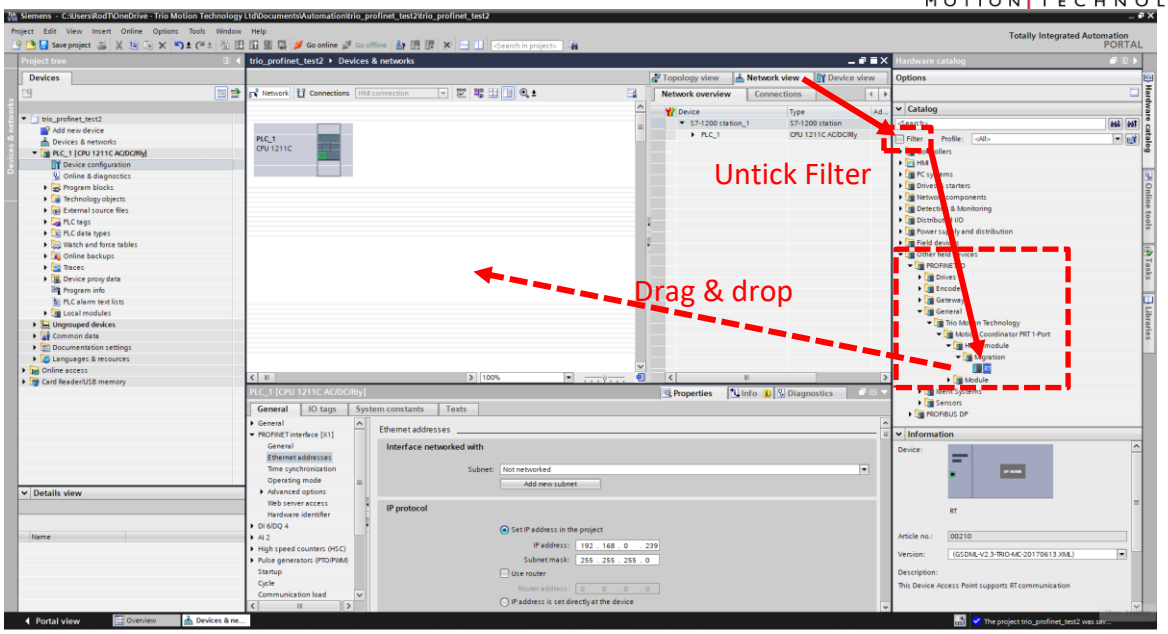


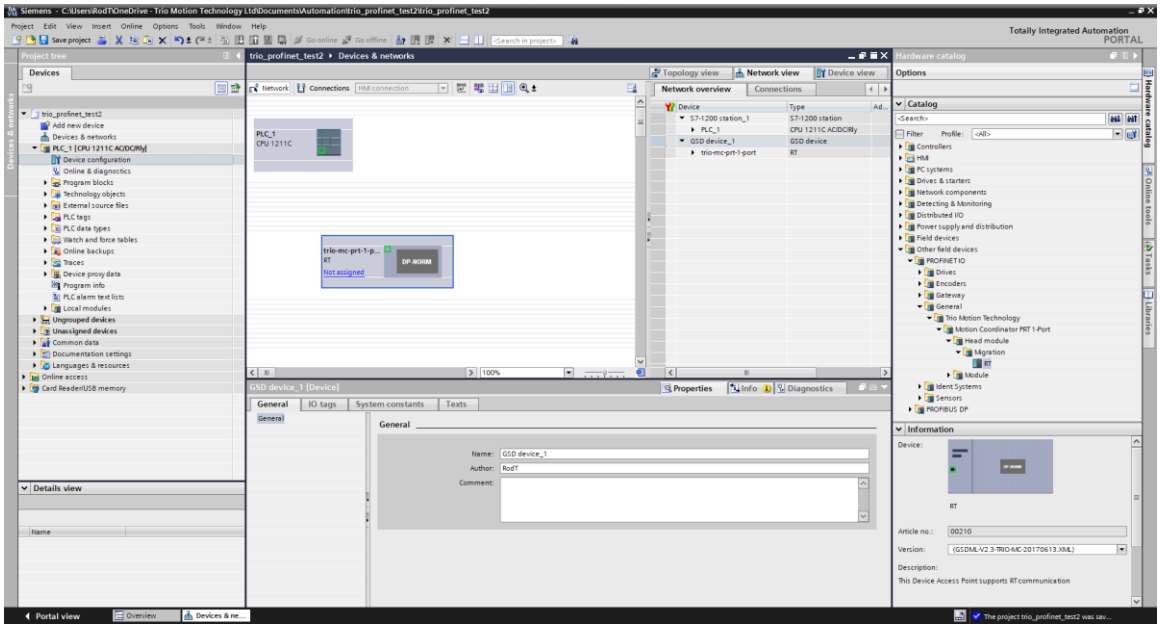




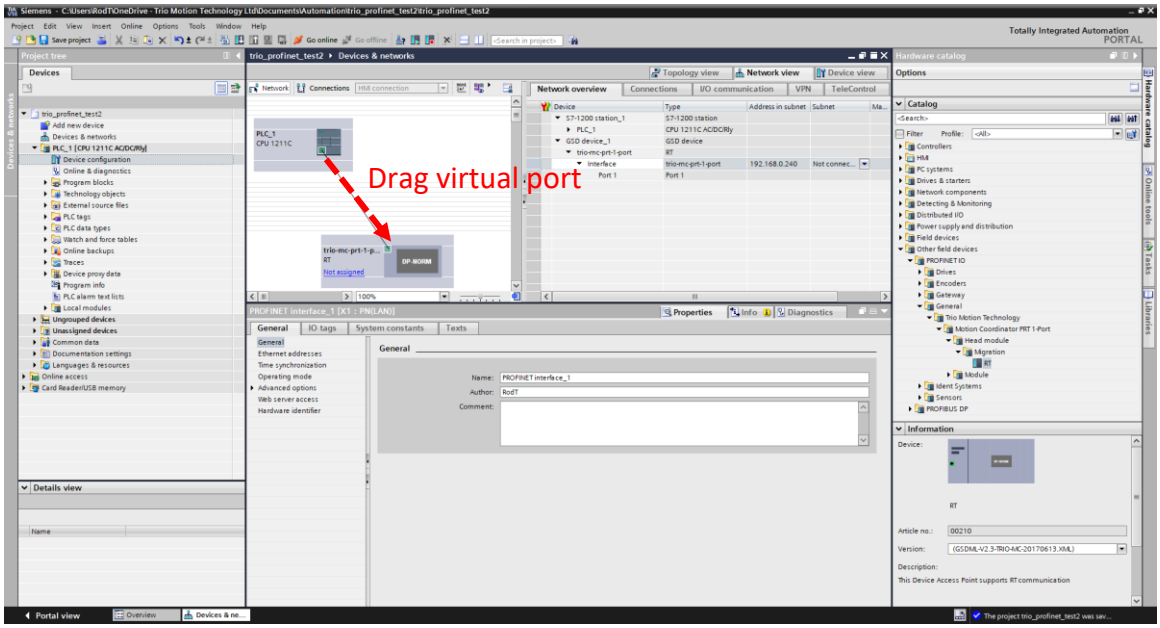
4.3.2. Adding TRIO Controller into Project

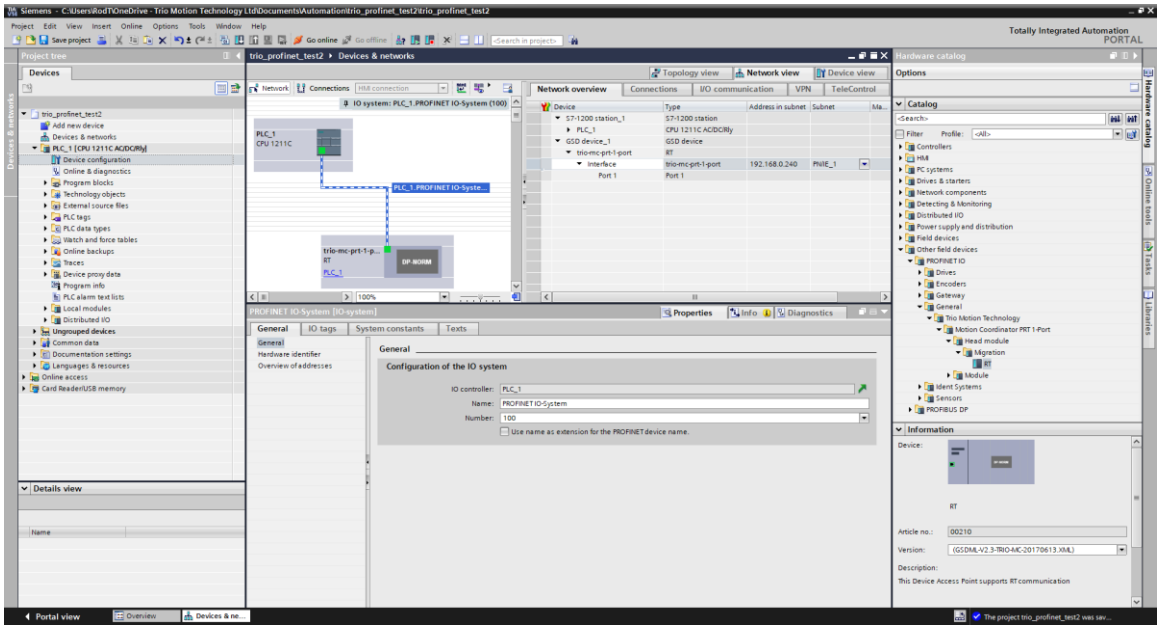
Follow the steps below to add TRIO controller into your TIA Portal project.


Untick Filter
Drag & drop



4.3.3. Creating Network Interface between TRIO Controller and SIEMENS PLC

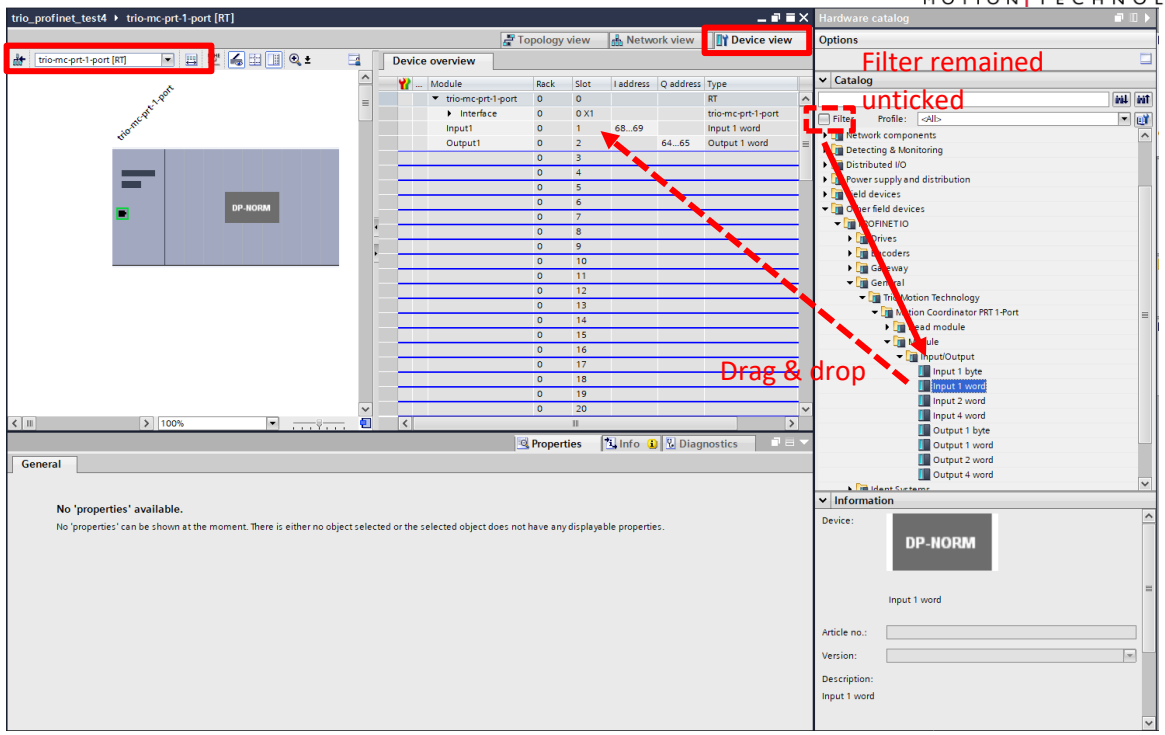




4.3.4. TRIO Module Setup

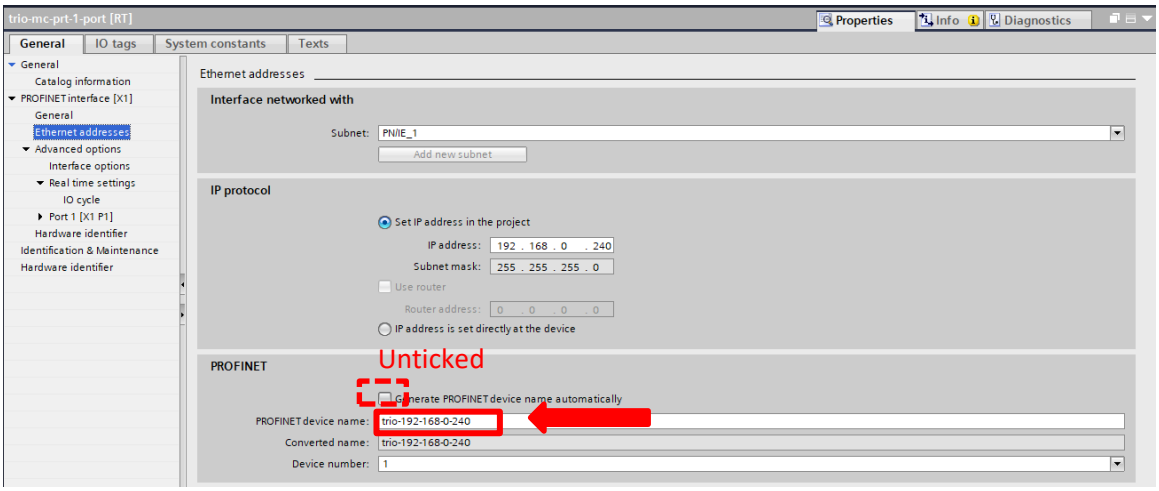
Once Trio GSDML is installed, you should be able to drag and drop the available IO modules into the configuration.

In this example, we will only use 1 word input and 1 word output.

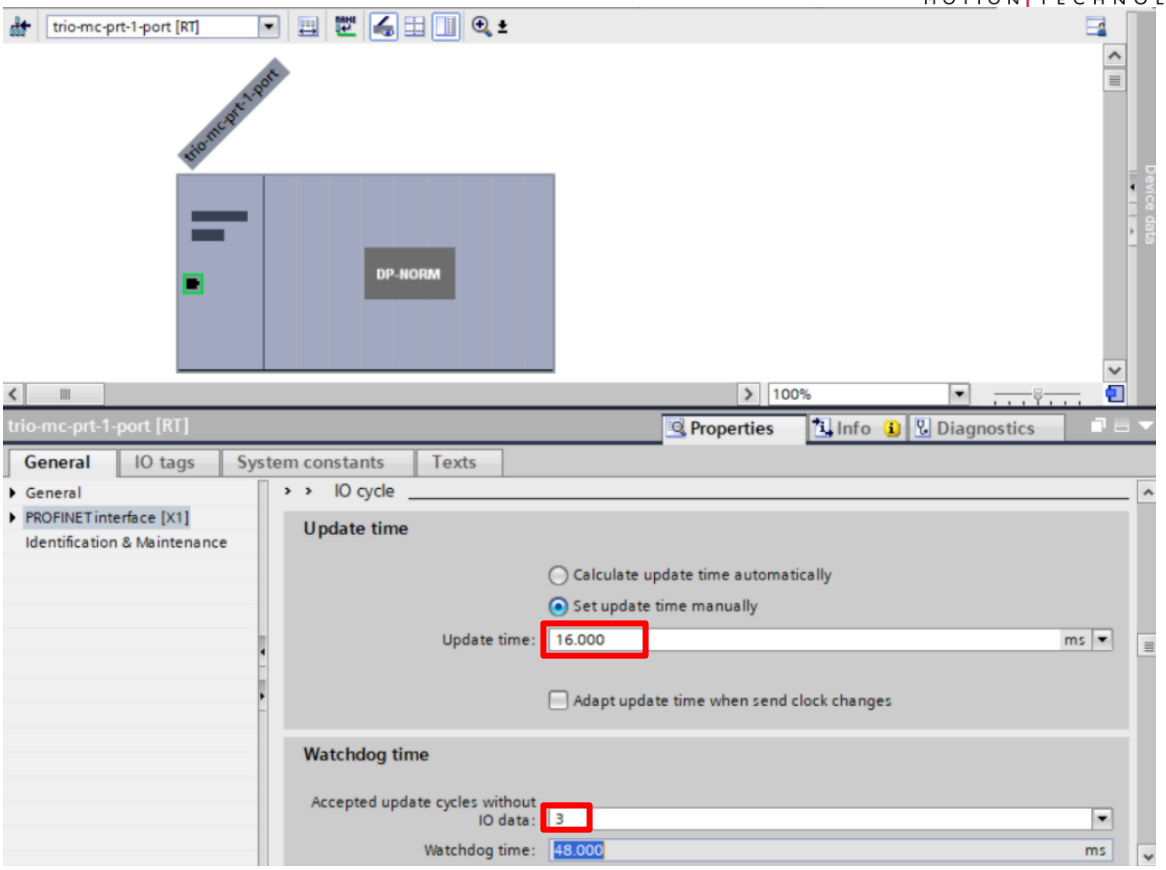


4.3.5. TRIO Module Ethernet Properties

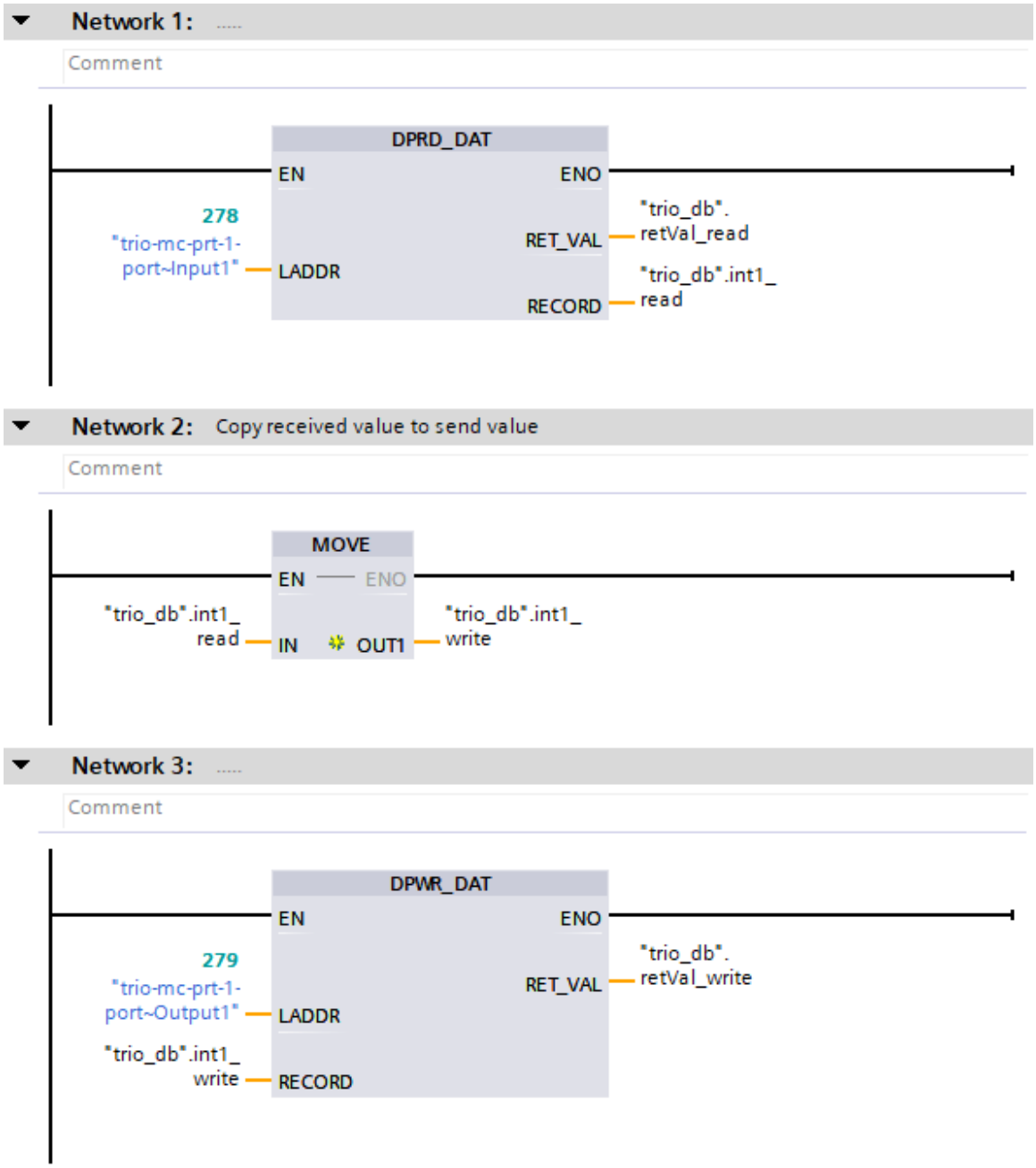
Trio controller has a default Profinet name in the following format: *trio-<controller name>-<5 digit serial number>*. For example, trio-f6nano-00260. **Note**: profinet name needs to be identical with function 8 defined in an autorun program in Motion Perfect project. See 4.2.1 sample program.



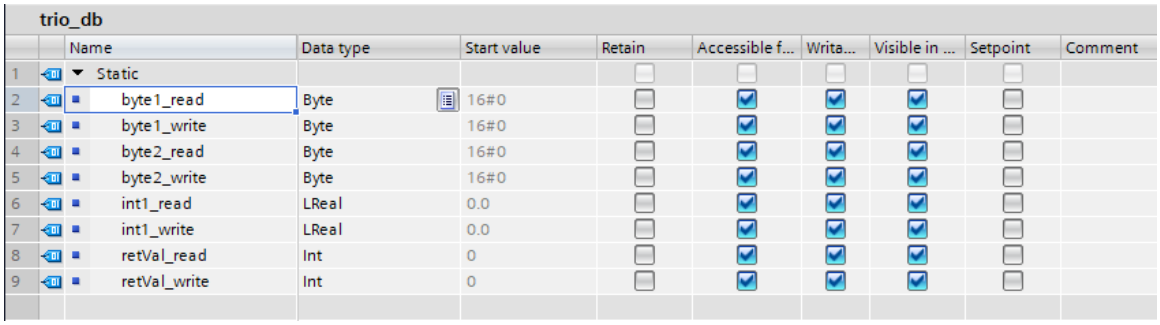
Assign Update Time and Accepted Update Cycles without IO data to be 16 and 3 respectively.


trio-mc-prt-1-port [RT]
trio-mc-prt-1-port
DP-NORM
100%
trio-mc-prt-1-port [RT]
Properties
Info
Diagnostics
General
IO tags
System constants
Texts
General
PROFINET interface [X1]
Identification & Maintenance
IO cycle
Update time
Calculate update time automatically
Set update time manually
Update time:
16.000
ms
Adapt update time when send clock changes
Watchdog time
Accepted update cycles without IO data:
3
Watchdog time:
48.000
ms

4.3.6. Main [OB1]


Network 1:
Comment
DPRD_DAT
EN
ENO
278
"trio-mc-prt-1-port~Input1"
LADDR
RET_VAL
"trio_db".retVal_read
RECORD
"trio_db".int1_read
Network 2: Copy received value to send value
Comment
MOVE
EN
ENO
"trio_db".int1_read
IN
OUT1
"trio_db".int1_write
Network 3:
Comment
DPWR_DAT
EN
ENO
279
"trio-mc-prt-1-port~Output1"
LADDR
RET_VAL
"trio_db".retVal_write
"trio_db".int1_write
RECORD

4.3.7. Trio_db [DB1]



trio_db

		Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Comment
1		Static			☐	☐	☐	☐	☐	
2		byte1_read	Byte	16#0	☐	☑	☑	☑	☐	
3		byte1_write	Byte	16#0	☐	☑	☑	☑	☐	
4		byte2_read	Byte	16#0	☐	☑	☑	☑	☐	
5		byte2_write	Byte	16#0	☐	☑	☑	☑	☐	
6		int1_read	LReal	0.0	☐	☑	☑	☑	☐	
7		int1_write	LReal	0.0	☐	☑	☑	☑	☐	
8		retVal_read	Int	0	☐	☑	☑	☑	☐	
9		retVal_write	Int	0	☐	☑	☑	☑	☐	