Application Note

Applicative System Redundancy for PLCnext Control AXC F 2152

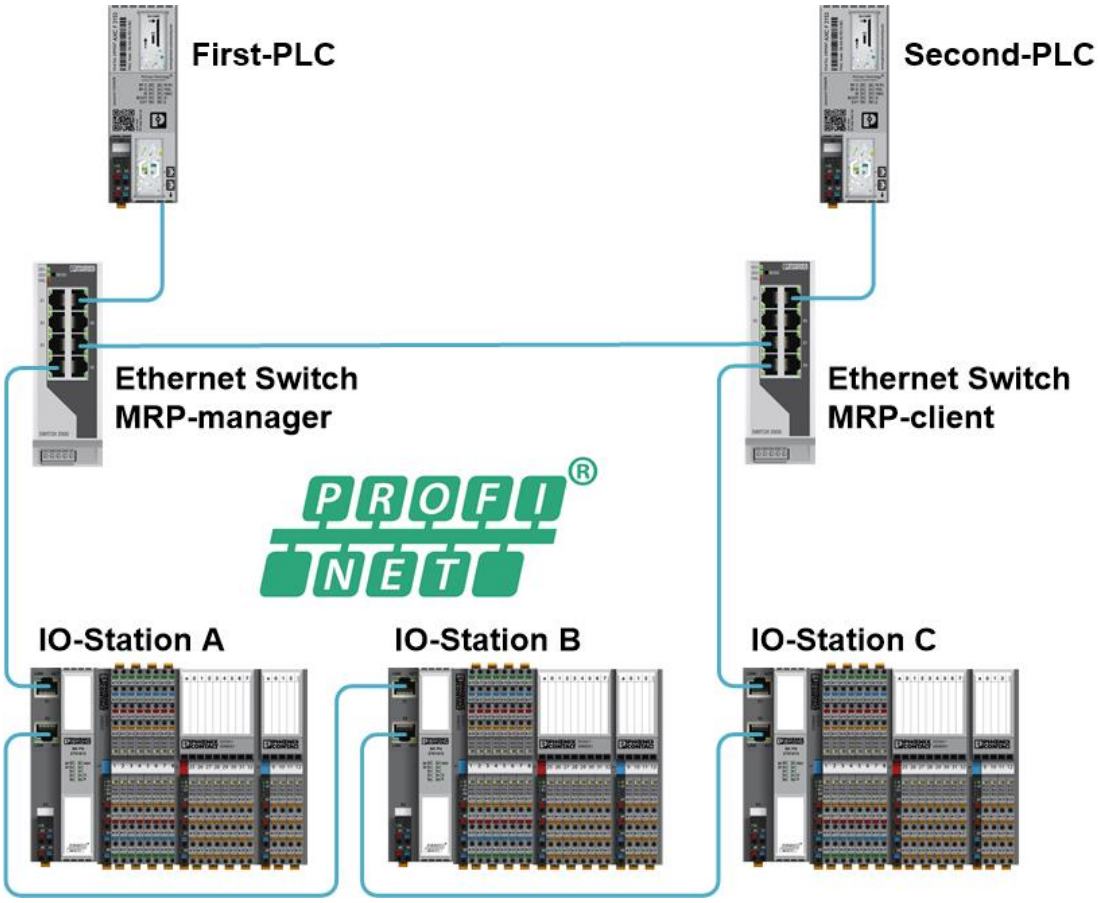
Established by:
Phoenix Contact Electronics GmbH
Friedrich Hackl
Phone: +49 52 81 9 46-2264
E-mail: fhackl@phoenixcontact.com

Revision Documentation			
Date	**Author**	**Comments**	**Revision**
27.01.20	Friedrich Hackl	Preliminary version	**00**
29.04.20	Friedrich Hackl	First released version	**01**
19.02.21	Friedrich Hackl	Documentation adapted to library version 2	**02**

Table of Contents

1 Introduction

The Applicative System Redundancy (ASR) is realized by providing two redundant PLCs (First and Second) in the same network together with the responding IO-stations. The communication between the PLCs and the IO-stations is based on the Profinet RT protocol.



On both controllers (First- and Second-PLC) the identical application program is in operation. One of the PLCs is the active controller (serves as Primary controller) and the other is the standby controller (serves as Backup controller). The Primary controller controls the process. The redundancy role (Primary or Backup) is specified by the application software. Therefore, the responding function block needs to be implemented and parameterized in the application program. If the Primary controller fails or some other user defined switchover condition occurs, the Backup controller takes over the process control.

The use of function blocks for data synchronization allows during normal operation (as long as both controllers are available) selected data to be continuously synchronized between the Primary and the Backup controller. By holding and processing identical application data, a seamless switchover and a short take over time is possible from the Primary to the Backup controller in case of malfunction of the Primary controller.

The Applicative System Redundancy is based on the mechanism of the PROFINET System Redundancy. This PROFINET standard describes several types of System Redundancy. The PLCnext Technology control system supports PROFINET System Redundancy type S2 according to that PROFINET specification.

PROFINET IO-devices used in such a system must support PROFINET system redundancy type S2, including the option “RT_InputOnBackupAR_Supported” activated. This option ensures that the PROFINET IO-devices provide valid input data also for the Backup controller. PROFINET IO-devices that do not support this option cannot be integrated into the PROFINET bus configuration when the PROFINET system redundancy type S2 is activated.

The IO-Stations are directly integrated into a network ring structure via the Media Redundancy Protocol (MRP) by using two ethernet switches. One switch must be configured as MRP-manager and the other as MRP-client. In such a network ring structure with Media Redundancy Protocol, maximum network recovery time of 200ms can be expected.

First- and Second-PLC (PROFINET IO-Controller):
AXC F 2152 with firmware version ≥2020.0 LTS

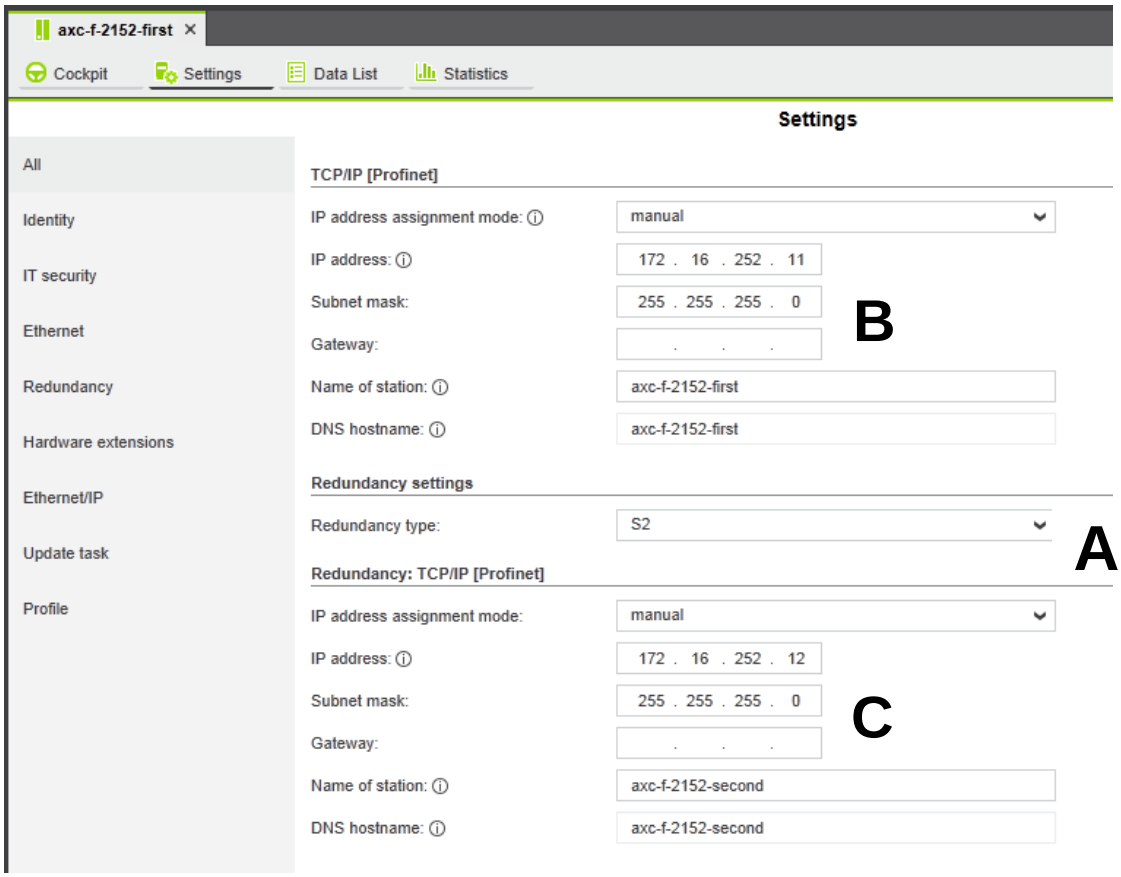
Ethernet switch MRP-client:
FL SWITCH 22xx

Ethernet switch MRP-manager:
FL SWITCH 22xx + FL SD FLASH/MRM

IO-Stations (PROFINET IO-Device):
AXL F BK PN TPS with firmware version ≥1.31
+ Axioline F IO-modules

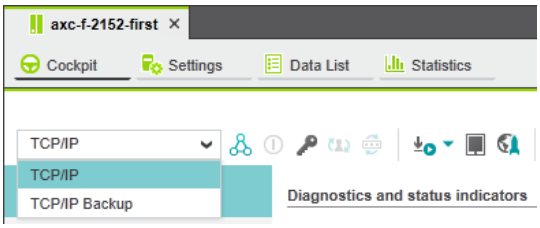
1.1 Engineering software PLCnext Engineer for redundant systems

The PROFINET System Redundancy needs to be activated and parameterized. For a detailed description of how to do this, please have a look at the online help of PLCnext Engineer.



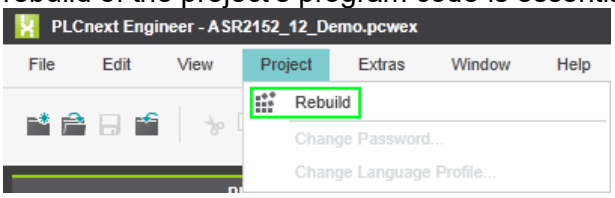
A…Activation of the PROFINET System Redundancy
B…IP-address and PROFINET name of the First-PLC
C…IP-address and PROFINET name of the Second-PLC

Afterwards the selection with which controller you would like to connect to for the purpose of program download or debugging is subsequently possible via the appropriate dropdown menu in the Cockpit:

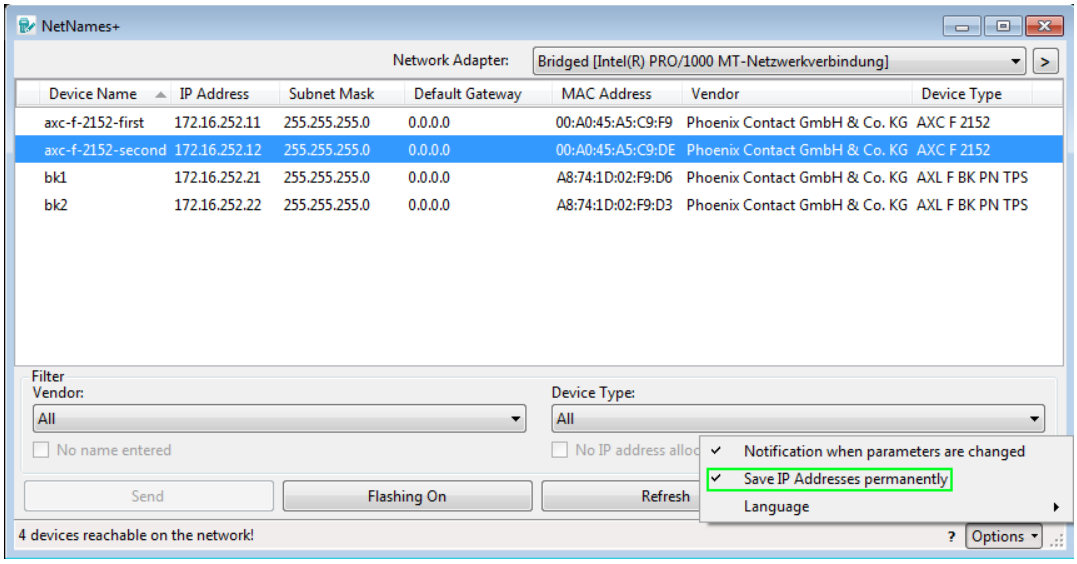


TCP/IP…	Connect online services to the First-PLC
TCP/IP Backup	Connect online services to the Second-PLC

Note:
If you switch the communication path between the First- and Second-PLC, a rebuild of the project's program code is essential:



For the adjustments of the IP-address and the PROFINET name on the PLCs, it is preferable to use the stand-alone software tool NetNames+. If the parameters are to be stored on the device permanently, the corresponding option must be selected:

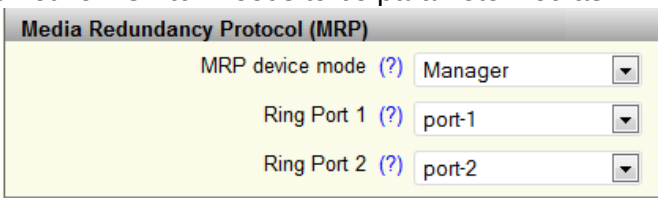


1.2 Configuration of the Media Redundancy network ring structure

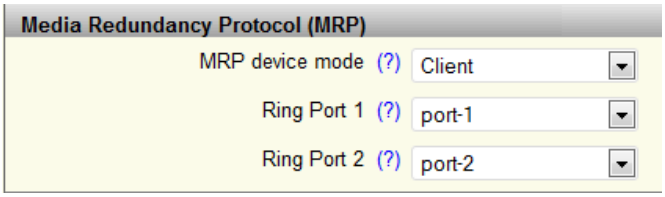
1.2.1 Network Switches

The network switches used in the MRP ring structure of the Applicative System Redundancy cannot be operated in PROFINET mode, but as a standard ethernet switch. Therefore, the operating mode "universal" (default setting) needs to be selected for these network switches. For this reason, the parameters for MRP are set via the web-based management.

One network switch needs to be parameterized as MRP-manager:



All other network switches need to be parametrized as MRP-client:

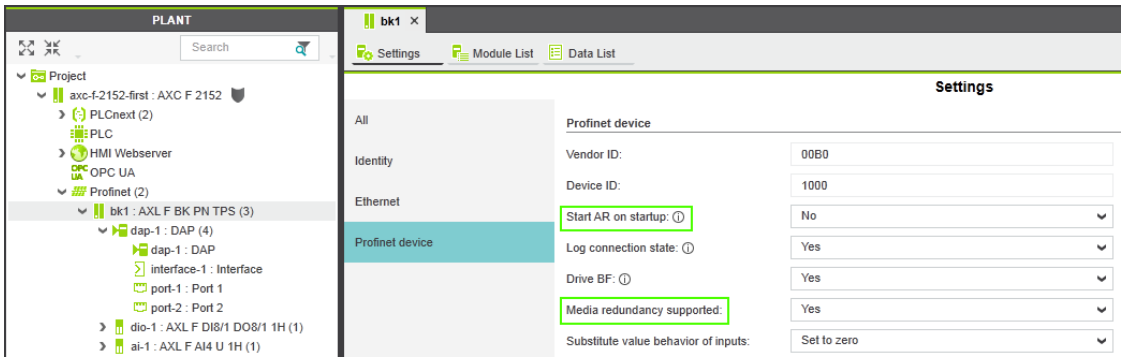


The ring ports for the MRP-manager and the MRP-client must be selected according to the actual situation of the network topology.

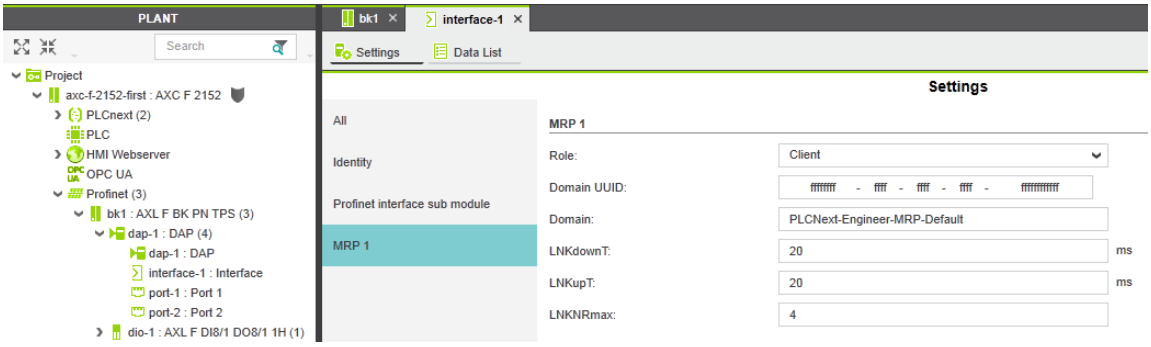
1.2.2 IO-Stations (PROFINET IO-devices)

The IO-stations can be directly integrated into the MRP network ring structure. In this case, the MRP-client on the bus coupler must be activated and parameterized. For these devices, the necessary settings of the MRP-client are directly done in PLCnext engineer.

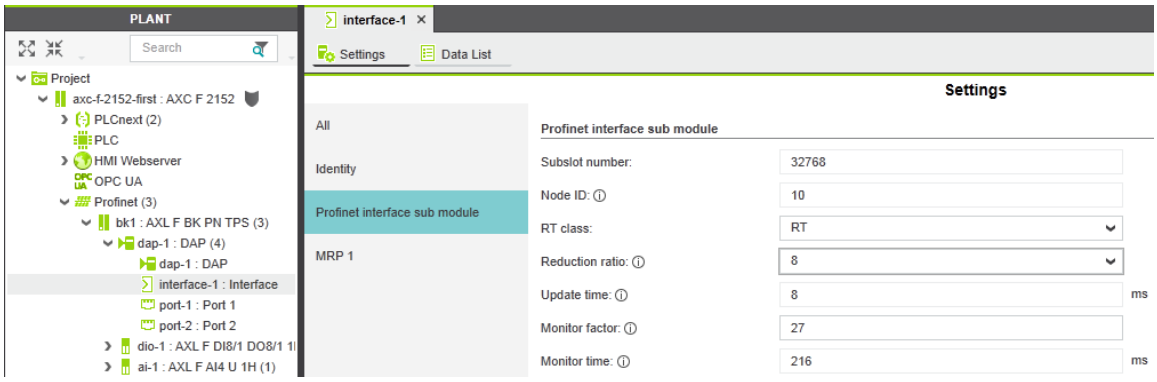
At the category “Profinet device” of the settings for the node of the PROFINET IO-device the media redundancy support is activated and the parameter “Start AR on startup” has to be set deactivated:



At the settings of the node “interface” the parameters for MRP can be adjusted. For the use case of the Applicative System Redundancy the default values can be maintained:



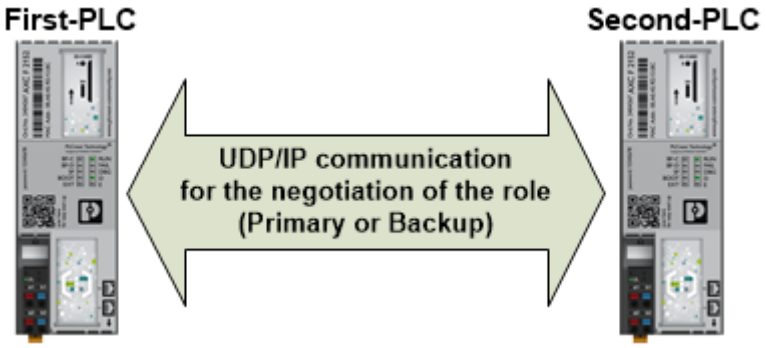
In general, PROFINET System Redundancy is independent of the network topology and, conversely, media redundancy is independent of the PROFINET System Redundancy. Therefore, both redundancy levels can be combined or used independently of each other. However, if both redundancy mechanism are combined, the network recovery time of the Media Redundancy Protocol must be considered in the adjustments of the PROFINET System Redundancy. Specifically, this concerns the value of the so-called “Monitor time” for the PROFINET IO-devices. This parameter must be set in that way that a switchover in the ring network does not result in a loss of communication between PLCs and IO-Stations. This means that the value of the “Monitor time” must be greater than the maximum network reconfiguration time to expect. The value for the “Monitor time” cannot be set directly but is the product of “Monitor factor” and “Update time”. Since the “Update time” can be adjusted individually, according to the needs of the application, for each PROFINET IO-device, the “Monitor factor” must also be adjusted to the proper value for each device in the settings of the node “interface”:



Since the maximum of the network recovery time for MRP is 200ms, at the example of an “Update time” of 8ms the “Monitor factor” needs to be set to a minimum of 27.

2 Function block for the Controller-Redundancy

2.1 General



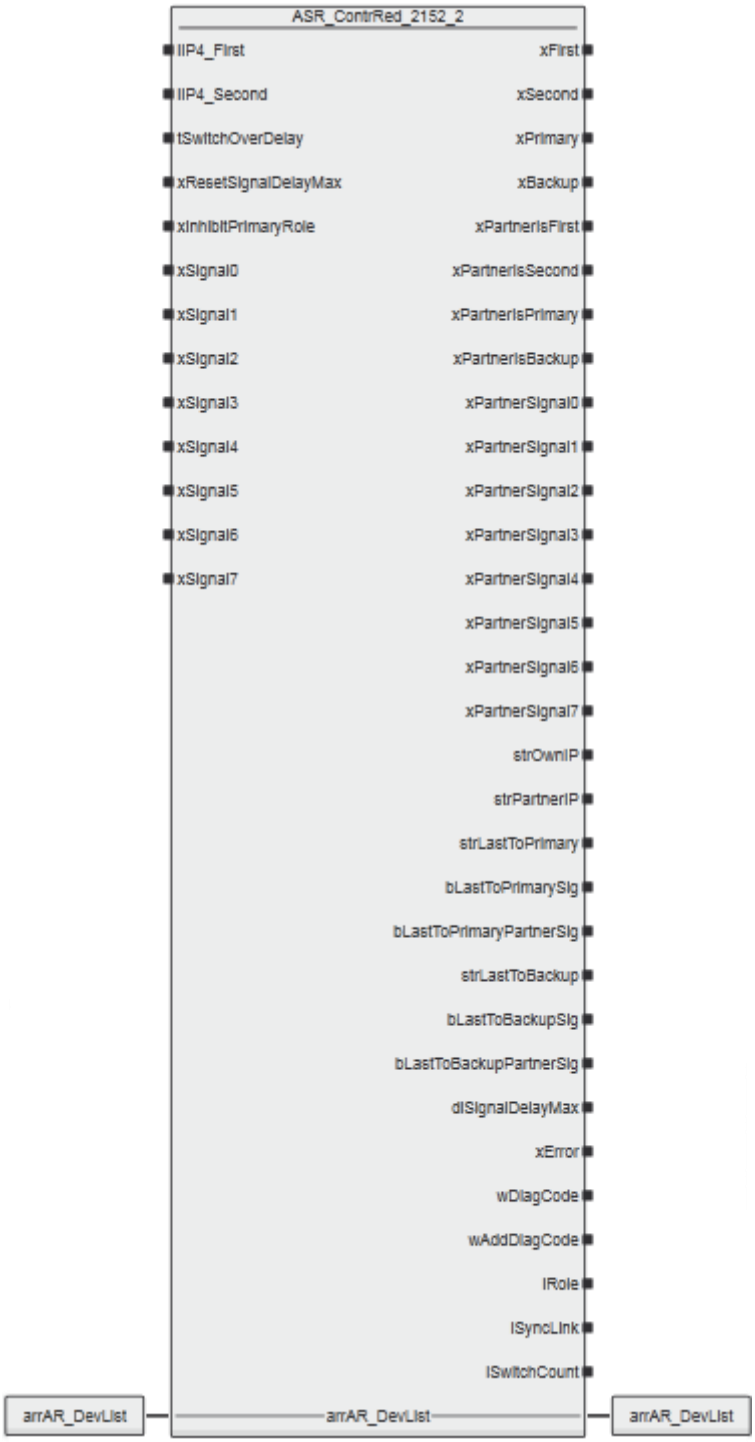
The communication between the two PLCs necessary for the Controller-Redundancy is realised via an UDP/IP communication protocol. For this purpose, the following rules for the assignment of the IP-addresses must be considered:

- The IP-addresses of the First- and Second-PLC must use the same Net-IDs.
- The difference between the IP-addresses of the First- and the Second-PLC must be limited to the fourth octet only.

Example:

	First-PLC	Second-PLC
IP Address	192.168.0.91	192.168.0.92
Subnet Mask	255.255.255.0	255.255.255.0

2.2 Function block “ASR_ContrRed_2152”



On the two PLCs the same program is in operation. Therefore, it is necessary to determine during the runtime of the program, which PLC is the First- and which PLC is the Second-Controller. For this purpose, the fourth octet of the IP-addresses (Host-ID) of the First- and of the Second-Controller needs to be stored as a constant at the corresponding input parameters of this function block in the PLCnext Engineer program. During the start-up of the program the own IP-address is read out by executing a Firmware-Service. As a result of the comparison of the own IP-address with these constants, it can be determined, whether the Controller is the First- or the Second-PLC.

Only one controller (either First or Second) controls the process. The PLC which controls the process is the Primary, the other PLC has the role of the Backup. The determination which PLC is the Primary respectively Backup, is done by a comparison of the value of own Current-State-Signals with Current-State-Signals of the Partner-PLC.

Therefore, this function block compares cyclically the Current-State-Signals of the Partner-PLC (xPartnerSignal...) with the own Current-State-Signals (xSignal...). The values of the signals are compared in ascended order beginning with 0. If a difference of the values at a determined bit-number x is perceived, the values of the Current-State-Signals with higher bit-numbers (>x) are not compared anymore. Therefore, the priority of a switch over condition depends on the bit-number used for this Current-State-Signal.

Possible switch over conditions:

- Switching over the state of a PLC from Backup to Primary is carried out, if the value of an own Current-State-Signal is TRUE and the value of the corresponding Current-State-Signal from the Partner-PLC is FALSE.

- Switching over the state of a PLC from Primary to Backup is carried out, if the value of an own Current-State-Signal is FALSE and the value of the corresponding Current-State-Signal from the Partner-PLC is TRUE.

2.2.1 Description of inputs and outputs:

Name	Data type	Data direction	Description
iIP4_First	INT	Input	Fourth octet of the IP-address of the First-PLC.
iIP4_Second	INT	Input	Fourth octet of the IP-address of the Second-PLC.
tSwitchOverDelay	TIME	Input	Delay time for switching over the PLCs in case of a switchover condition occurs.
xResetSignalDelayMax	BOOL	Input	A rising edge at this input signal resets the output signal "uiSignalDelayMax".
xInhibitPrimaryRole	BOOL	Input	With a TRUE the takeover of the Primary role is inhibited. For example, if the PLC has no connection to the network at all.
xSignal...	BOOL	Input	Current-State-Signals (0...highest priority, 7...lowest priority).
arrAR_DevList	ASR_ARR_STR_1_50	Input	“Name of station” list of all PROFINET IO-devices.
xFirst	BOOL	Output	Indicates, whether the PLC is the First-PLC.
xSecond	BOOL	Output	Indicates, whether the PLC is the Second-PLC.
xPrimary	BOOL	Output	Indicates, whether the PLC has the redundancy role Primary.
xBackup	BOOL	Output	Indicates, whether the PLC has the redundancy role Backup.
xPartnerIsFirst	BOOL	Output	Indicates, whether the Partner-PLC is the First-PLC.
xPartnerIsSecond	BOOL	Output	Indicates, whether the Partner-PLC is the Second-PLC.
xPartnerIsPrimary	BOOL	Output	Indicates, whether the Partner-PLC has the redundancy role Primary.
xPartnerIsBackup	BOOL	Output	Indicates, whether the Partner-PLC has the redundancy role Backup.
xPartnerSignal...	BOOL	Output	Current-State-Signals of the Partner-PLC (0...highest priority, 7...lowest priority).

strOwnIP	STRING	Output	Own IP-address.
strPartnerIP	STRING	Output	IP-address of the Partner-PLC.
strLastToPrimary	STRING	Output	Indicates time and date for the last switchover of the PLC from Backup to Primary.
bLastToPrimarySig	BYTE	Output	Indicates the state of the signals, when the last switchover from Backup to Primary was performed.
bLastToPrimaryPartnerSig	BYTE	Output	Indicates the state of the signals from the Partner-PLC, when the last switchover from Backup to Primary was performed.
strLastToBackup	STRING	Output	Indicates time and date for the last switchover of the PLC from Primary to Backup.
bLastToSlaveSig	BYTE	Output	Indicates the state of the signals, when the last switchover from Primary to Backup was performed.
bLastToSlavePartnerSig	BYTE	Output	Indicates the state of the signals from the Partner-PLC, when the last switchover from Primary to Backup was performed.
diSignalDelayMax	DINT	Output	Maximum measured delay time for the transmission of the Current-State-Signals. This output parameter can be reset by a rising edge at the input signal “xResetSignalDelayMax”.
xError	BOOL	Output	This output indicates with a rising edge that an error has occurred. You can read the corresponding error codes at the output-signals "wDiagCode" and "wAddDiagCode" during the cycle in which the rising edge occurs.
wDiagCode	WORD	Output	Indicates the cause of the error.
wAddDiagCode	WORD	Output	Indicates additional information to the cause of the error.
iRole	INT	Output	Indicates the current role of the PLC: 0…. Undefined 1…. Primary 2…. Backup
iSyncLink	INT	Output	Indicates the status for the transmission of the Current-State-Signals: 0…. Undefined 1…. Transmission of the signals is OK 2…. Transmission of the signals is not OK
iSwitchCount	INT	Output	Number of switchovers of the PLC from Backup to Primary.

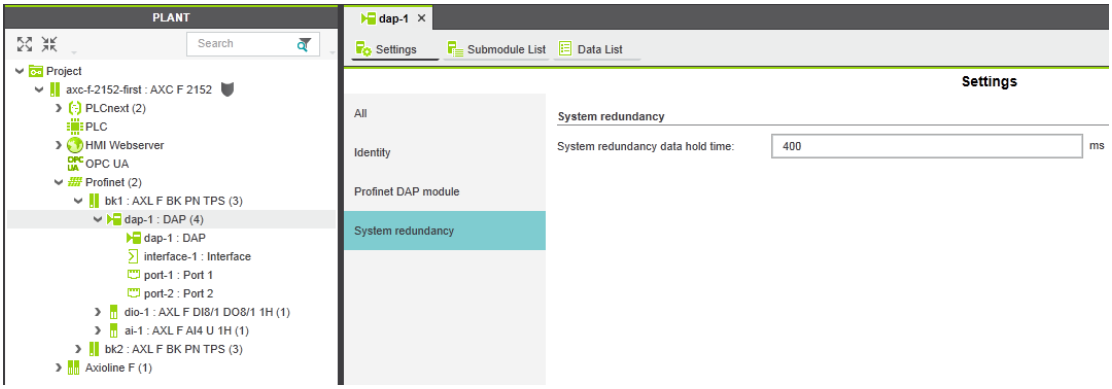
The valid range for the value of the input parameter “tSwitchOverDelay” is between 300ms and 1000ms.
The achievable switchover time of the system depends on the following different parameters of the overall application and cannot be generalized:

- Task interval in which this function block is used.
- Overall CPU load of the PLC.
- Network redundancy protocol which is used for the application specific network layout.

However, the following table shows the recommended minimum switchover time which can be adjusted depending on the adjusted task interval in which the function block is used:

Task interval	**Minimum value for the input variable „tSwitchOverDelay“**
20ms	300ms

The switchover time of the system must also be parameterized for the PROFINET IO-devices. This is done at the settings of the node “DAP” via the so-called “System redundancy data hold time”. The value entered here must not be smaller than the switchover time set at the function block:



2.2.2 Error Codes

Error Code	Description
C401	Indicates that an error while trying to stop the PROFINET ARs. The output parameter “wAddDiagCode” indicates the relevant error code from the responding function block.
C402	Indicates that an error while trying to start the PROFINET ARs. The output parameter “wAddDiagCode” indicates the relevant error code from the responding function block.
C403	Indicates that an error occurred while reading the adjustments of the IP-address. The output parameter “wAddDiagCode” indicates the relevant error code from the responding function block.
C404	Indicates that this PLC is neither the First- nor the Second-PLC, because the fourth octet of the IP-address neither is equal with the value of the input parameter “iIP4_First” nor with “iIP4_Second”. The output parameter “wAddDiagCode” contains the fourth octet of the adjusted IP-address.
C405	Indicates that during the start up of the internal state machine a timeout occurred. The output parameter indicates the relevant step at which the timeout was detected.

2.2.3 Additional program instances in the ESM

In order for the controller redundancy to work properly, the following program instances must be created in addition to the cyclic call of the function block:

- Program instance for the system event task “Cold Start”:

Name	Component name	Task type	Event name	Program type	Interval (ms)	Priority	Threshold (ms)	Watchdog (ms)
ESM1								
COLDSTART		Event task	Cold start			0	0	100
ProgColdstart	Arp.Plc.Eclr			ASR_EventStart_1				

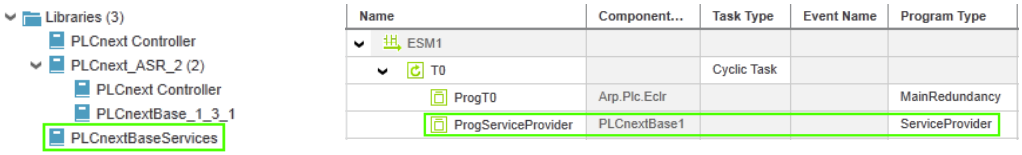
- Program instance for the system event task “Warm Start”:

Name	Component name	Task type	Event name	Program type	Interval (ms)	Priority	Threshold (ms)	Watchdog (ms)
ESM1								
WARMSTART		Event task	Warm start			0	0	100
ProgWarmStart	Arp.Plc.Eclr			ASR_EventStart_1				

- Program instance for the system event task “Stop”:
  Note that the watchdog of this system event task must be set to 2000ms.

Name	Component name	Task type	Event name	Program type	Interval (ms)	Priority	Threshold (ms)	Watchdog (ms)
ESM1								
STOP		Event task	Stop			0	0	2000
ProgStop	Arp.Plc.Eclr			ASR_EventStop_1				

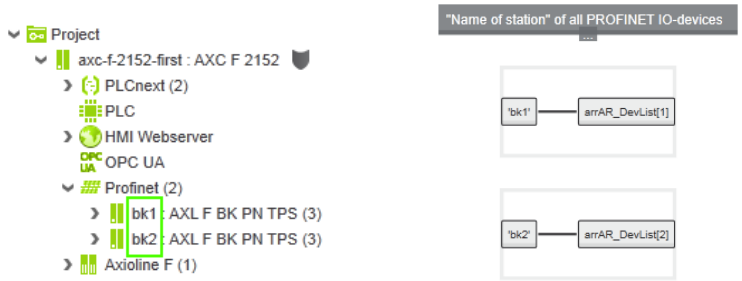
The function block “ASR_ContrRed_2152” uses internally some firmware services. To make these services available, it is necessary to include the library “PLCnextBaseServices” into the PLCnext Engineer project and to create a program instance of the program “ServiceProvider” in one cyclic task:



2.2.4 Recommendations for the configuration of the function block

InOut parameter “arrAR_DevList”:

At the start-up the function block performs some PROFINET specific services on the PROFINET IO-devices and therefore the function block needs to know the name of all connected IO-devices. For this purpose, the “Name of Station” of all PROFINET IO-devices (bus couplers) must be stored at the InOut parameter “arrAR_DevList”:

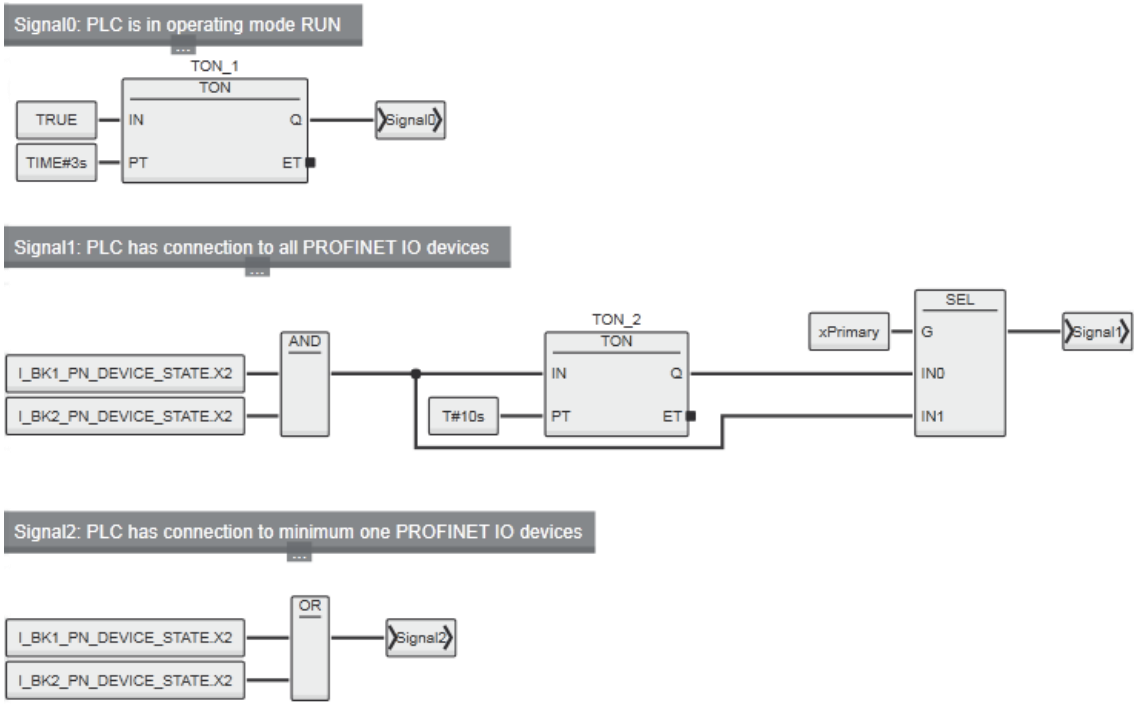


At this example program two PROFINET IO-devices with the names “bk1” and “bk2” are used. If there are more IO-devices connected, their names must be also added to this list.

Input parameters “xSignal0” to “xSignal7”:

The switchover conditions are defined by logical operations which are assigned to the input parameters “xSignal0” to “xSignal7”.
If there are no special requirements related to the switchover conditions the following configuration is recommended in order to trigger a switchover in case of “standard failure”

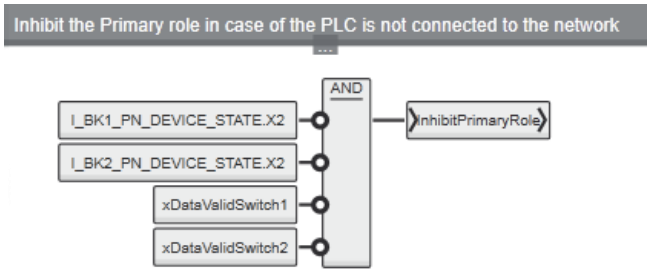
situations like switching off the power supply of a PLC, setting a PLC in operating mode Stop or disconnecting the ethernet connection:



The variables “I_BKx_PN_DEVICE_STATE” are to be defined as External Variables and the responding Global Variables needs to be connected to the Process Data Item “PN_DEVICE_STATE” of the PROFINET bus couplers.
At this example program two bus couplers are used. If there are more bus couplers in the system, the “AND” and the “OR” function needs to be extended accordingly.

Input parameter “xInhibitPrimaryRole”:

This input parameter is used to prevent the PLC from taking over the Primary role. This is usually useful when the PLC has no connection to the ethernet network at all. In this case the PLC doesn’t know the current status of the Partner-PLC and therefore it should not takeover the Primary role.
In the following example two bus couplers and two ethernet switches are used in the system. Thus, if the controller has neither a connection to one of the bus couplers nor to one of the switches, it can be assumed that the controller is not connected to the network and therefore the takeover of the Primary role would not make sense:

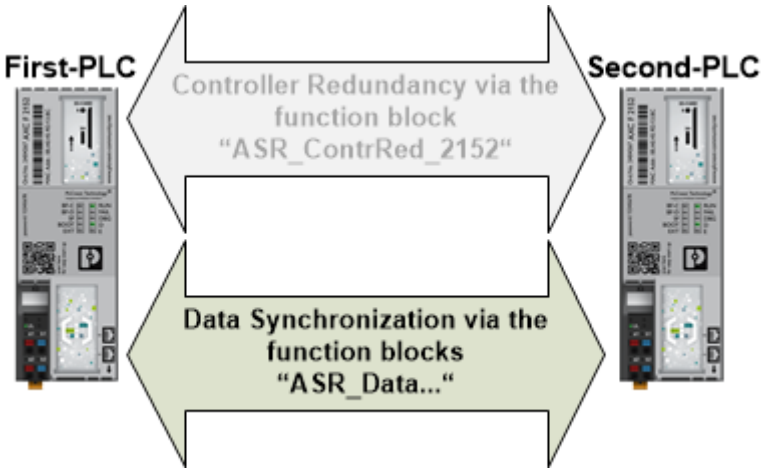


2.2.5 OPC UA variables

In order to make it possible to display the current status of the redundant controllers on a visualization system, the function block provides corresponding information in the form of OPC UA variables via the embedded OPC UA server on the PLCs.

Name	Data type	Description
iRole	INT	Indicates the current role of the PLC: 0…. Undefined 1…. Primary 2…. Backup
iSwitchCount	INT	Number of switchovers of the PLC from Backup to Primary.
iSyncLink	INT	Indicates the status for the transmission of the Current-State-Signals: 0…. Undefined 1…. Transmission of the signals is OK 2…. Transmission of the signals is not OK
uiPlcInRun	UINT	Cycle counter for the “PLC in Run” monitoring. The value of this counter is incremented each program cycle by 1 as soon as the application program is in operating mode RUN.
siFirstPlcCurrentRole	SINT	Indicates the current role of the First-PLC: 0…. Undefined 1…. Primary 2…. Backup
siFirstPlcState	SINT	Indicates the current state of the First-PLC: 0…. Undefined 1…. Error 2…. Run
siSecondPlcCurrentRole	SINT	Indicates the current role of the Second-PLC: 0…. Undefined 1…. Primary 2…. Backup
siSecondPlcState	SINT	Indicates the current state of the Second-PLC: 0…. Undefined 1…. Error 2…. Run

3 Function blocks for the Data Synchronization



At the Applicative System Redundancy on the two redundant PLCs (First- and Second-PLC) the same application program is in operation, but only one PLC (either First or Second) controls the process. By the means of the function block “ASR_ContrRed_2152” the PLCs negotiate among themselves who controls the process. The PLC, which controls the process is the Primary, the other PLC has the role of the Backup.

The task of the following described function blocks is to synchronize values of selected variables between the PLCs. The library provides function blocks for common standard data types. For this purpose, there are two different types of function blocks:

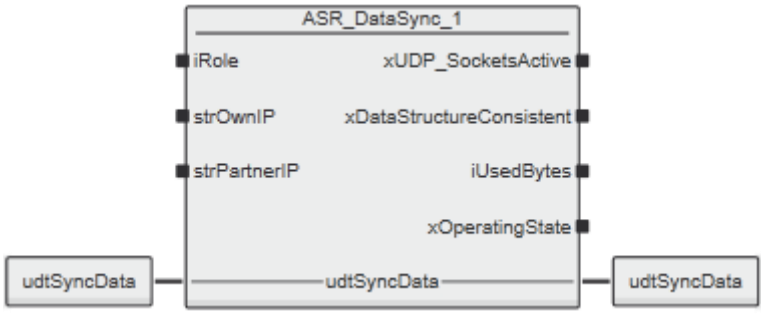
- Main function block for establishing the communication link between the PLCs.
- Function blocks for the synchronization of standard data types.

NOTE:

- These function blocks are designed to be used only in one common cyclic task.

- The call of the function block must be active in each program cycle. This means that these function blocks must not be used inside any IF-, ELSE-, ELSIF-, REPEAT-, CASE-, WHILE-Statement or FOR-Loops.

3.1 Main function block “ASR_DataSync”



The input parameters "iRole", strOwnIP and "strPartnerIP" must be connected to the corresponding output parameters of the function block "ASR_ContrRed_2152". Via the output parameter “xUDP_SocketsActive”, the function block indicates whether the UDP/IP sockets, necessary for the data synchronization, were successfully opened.

The output parameter “xDataStructureConsistent” indicates with a TRUE that the number and the sequence of calls of the function blocks for the synchronization of the standard data types are identical on both PLCs. Only if this bit indicates TRUE, the variables between the two controllers will be synchronized. Via the output parameter “xOperatingState” it is shown whether the communication between the two PLCs is in operation.

The output parameter “iUsedBytes” shows the number of bytes which are necessary for the synchronization of the variables (standard data types). The system supports a maximum of 32767 bytes for the data synchronization of standard data types.

The data exchange between the individual function blocks for the data synchronization is done by the means of the in-/out-parameter “udtSyncData”. Therefore, at each individual function block the same variable must be applied to this parameter.

3.1.1 Description of inputs and outputs:

Name	Data type	Data direction	Description
iRole	INT	Input	Current role of the PLC: 0...Undefined, 1...Primary, 2...Backup
strOwnIP	STRING	Input	Own IP-address. The proper syntax for this String is: ‘/IP=www.xxx.yyy.zzz’ (e.g. ‘/IP=192.168.0.91’)
strPartnerIP	STRING	Input	IP-address of the Partner-PLC. The proper syntax for this String is: ‘/IP=www.xxx.yyy.zzz’ (e.g. ‘/IP=192.168.0.92’)
xUDP_SocketsActive	BOOL	Output	"TRUE" indicates that the UDP/IP sockets were successfully opened.
xDataStructureConsistent	BOOL	Output	"TRUE" indicates that the sequential arrangement of the function blocks for the standard data types is identical on both PLCs.
iUsedBytes	INT	Output	Number of used bytes for the data exchange of the standard data types.

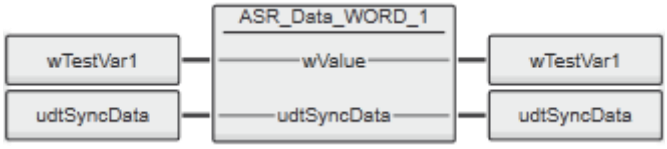
xOperatingState	BOOL	Output	"TRUE" indicates that the data exchange of the standard data types is in operation.
udtSyncData	ASR_UDT_ SYNC_DATA	In/Out	Parameter for the data exchange with the data type specific function blocks.

3.2 Function blocks for standard data types “ASR_Data_*”

By the means of these function blocks values of variables of standard data types can be synchronized between the PLCs. For this purpose, the active PLC (Primary) sends the values of the variables, which are applied to the in-/out-parameter “...Value”, to the passive PLC (Backup). The passive PLC receives these values and copies them to the variables connected to the in-/out-parameter.

‘*' is a placeholder for the supported data types. The FB is available for the data types BOOL, BYTE, WORD, DWORD, SINT, INT, DINT, UDINT, UINT, USINT, REAL and LREAL.

Example for the data type WORD:



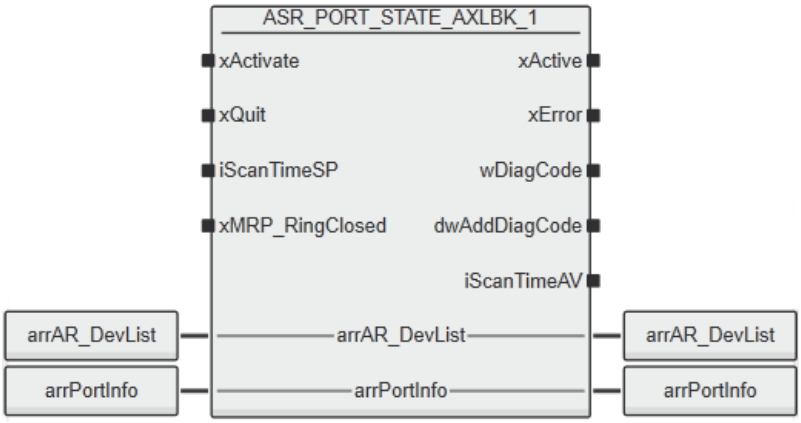
The variable, which should be synchronized between the PLCs, has to be applied to the in-/out-parameter “...Value”.
Via the in-/out-parameter “udtSyncData” the function block is connected to the responding main function block “ASR_DataSync”.

3.2.1 Description of inputs and outputs:

Name	Data type	Data direction	Description
...Value	BOOL, BYTE, WORD, DWORD, SINT, INT, DINT, USINT, UINT, UDINT, REAL, LREAL	In/Out	Variable which should be synchronized.
udtSyncData	ASR_UDT_ SYNC_DATA	In/Out	Parameter for the data exchange with the function block "ASR_DataSync".

4 Function block for gathering port states from Axioline PROFINET bus couplers

By the means of this function block the port states from PROFINET bus couplers of the type “AXL F BK PN TPS” can be read directly by the PLC via PROFINET acyclic services.



The PROFINET IO-device names of the bus couplers must be applied to the input parameter “arrAR_DevList”. This input parameter is an Array of String with 50 elements, so that a maximum number of 50 PROFINET IO-device names can be used.

A “TRUE” at the input parameter “xActivate” activates the function block. After the activation the function block starts with the initialization phase. During this initialization phase the applied PROFINET IO-device names are converted into unique IDs by the means of special PROFINET function blocks. This process can take several seconds. The successful termination of the initialization phase is indicated with a “TRUE” at the output parameter “xActive”. As soon as this output parameter is set to “TRUE”, the function block requests cyclically the port states from the specified bus couplers. A “FALSE” at the input parameter “xActivate” deactivates the function block and requesting of the port states is stopped.

The appearance of an error is indicated by the output parameter "xError". The output parameters "wDiagCode" and "dwAddDiagCode" display the cause of the error. The values are valid within the cycle, in which the rising edge of the output parameter "xError" occurred. The acknowledgment of an error message, which occurred during requesting the port states (output parameter “xActive” is “TRUE”), is carried out by a rising edge of the input parameter "xQuit". The acknowledgment of an error message during the initialization phase of the function block (output parameter “xActive” is “FALSE”) is only possible by deactivating the function block.

If the bus couplers are integrated into an MRP network ring structure and a signal is available indicating that the ring structure is closed, reading of the port states can be skipped by setting the input parameter “xMRP_RingClosed” to “TRUE”. As soon as this input parameter is set to “TRUE” the reading of port states via PROFINET acyclic services is stopped and the values of the elements “xLinkPort1” and “xLinkPort2” for all bus couplers are set to “TRUE”.

4.1 Description of inputs and outputs:

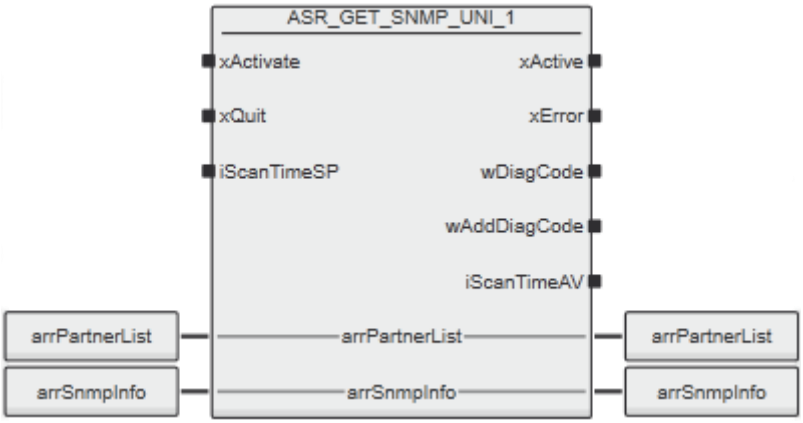
Name	Data type	Data direction	Description
xActivate	BOOL	Input	With a TRUE the function block is activated and reading of the port states is started. A FALSE deactivates the function block and reading of the port states is stopped.
xQuit	BOOL	Input	With a rising edge at this input parameter error messages, which occurred during requesting the port states, are acknowledged.
iScanTimeSP	INT	Input	Desired scanning cycle time in seconds.
xMRP_RingClosed	BOOL	Input	With a TRUE the reading of the port states via PROFINET acyclic services is stopped and the values of the elements “xLinkPort1” and “xLinkPort2” for all bus couplers are set to TRUE.
arrAR_DevList	ASR_ARR_STRING_1_50	Input	List of PROFINET IO-device names of the bus couplers from which the port states should be collected. This input parameter is an Array of String with 50 elements, so that a maximum number of 50 PROFINET IO-device names can be used.
xActive	BOOL	Output	A TRUE at this output parameter indicates that the initialization phase of the function block has been successfully terminated and the function block is reading now the port states cyclically.
xError	BOOL	Output	This output indicates with a rising edge that an error has occurred. You can read the corresponding error codes at the output parameters "wDiagCode" and "dwAddDiagCode" during the cycle in which the rising edge occurs.
wDiagCode	WORD	Output	Indicates the cause of the error.
dwAddDiagCode	DWORD	Output	Indicates additional information to the cause of the error.
iScanTimeAV	INT	Output	Actual scanning cycle time in seconds.
arrPortInfo	ASR_ARR_PORT_INFO	Output	Port states from the specified PROFINET bus couplers. This output parameter is an Array of Struct, in which each field contains the port states of a bus coupler corresponding to the PROFINET IO-device names at the input parameter “arrAR_DevList”. For each PROFINET bus coupler the following values are available: - xDataValid: This element indicates whether the last reading of the port states from the bus coupler was successful (TRUE) or not (FALSE). The following values are only valid, if this bit is TRUE. - xLinkPort1 and xLinkPort2: Link state of the port.

4.2 Error Codes

Error Code	Description
C401	Indicates that an error occurred at the initialization phase during the conversion of the PROFINET IO-device names into unique IDs. One possible reason for such an error could be that the appropriate PROFINET IO-device name doesn't exist in the bus configuration of the project. The output parameter "dwAddDiagCode" indicates the error code of the conversion function (HighWord) and the index of the relevant bus coupler (Low-Word). The acknowledgement of this error is done by deactivating the function block.
C402	Indicates that the function block for reading the port states reported an error while a request was sent to a PROFINET bus coupler. The output parameter "dwAddDiagCode" contains the DiagCode of the function block (High-Word) and the index of the relevant bus coupler (Low-Word). The acknowledgment of this error message is carried out at a rising edge of the input parameter "xQuit".
C403	Time-out detection of the internal state machine (5s). The output parameter "dwAddDiagCode" indicates the step of the internal state machine (High-Word) and the index of the relevant bus coupler (Low-Word) at which this error has occurred. The acknowledgment of this error message is carried out at a rising edge of the input parameter "xQuit".

5 Function block for gathering SNMP values from managed switches

By the means of this function block SNMP values from managed switches with up to 8 ports can be read directly by the PLC via the SNMP protocol.



The IP addresses and the number of ports (maximum 16) of the managed switches must be applied to the input parameter “arrPartnerList”. This input parameter is an Array of Struct with 30 elements, so that a maximum number of 30 IP addresses can be used. The proper syntax for an IP address is: ‘/IP=www.xxx.yyy.zzz’ (e.g. ‘/IP=192.168.0.21’).

A “TRUE” at the input parameter “xActivate” activates the function block. After the activation, the function block requests cyclically the SNMP values from the indicated managed switches. The desired scanning time can be adjusted via the input parameter “iScanTimeSP”. The received SNMP values are stored at the output parameter “arrSnmpInfo”. A “FALSE” at the input parameter “xActivate” deactivates the function block and requesting the SNMP values is stopped.

The appearance of an error is indicated by the output parameter "xError". The output parameters "wDiagCode" and "wAddDiagCode" display the cause of the error. The values are valid within the cycle, in which the rising edge of the output parameter "xError" occurred. The acknowledgment of an error message is carried out by a rising edge of the input parameter "xQuit".

5.1 Description of inputs and outputs:

Name	Data type	Data direction	Description
xActivate	BOOL	Input	With a TRUE the function block is activated and reading of the SNMP values is started. A FALSE deactivates the function block and reading of the SNMP values is stopped.
xQuit	BOOL	Input	With a rising edge at this input parameter error messages, which occurred during requesting the SNMP values, are acknowledged.
iScanTimeSP	INT	Input	Desired scanning cycle time in seconds.
arrPartnerList	ASR_ARR_PARTNER_LIST1_30	Input	List of IP addresses and the number ports of the managed switches from which the SNMP values should be collected. This input parameter is an Array of Struct with 30 elements, so that a maximum number of 30 IP

			addresses can be used. The proper syntax for an IP addresses is: '/IP=www.xxx.yyy.zzz' (e.g. '/IP=192.168.0.21').
xActive	BOOL	Output	A TRUE at this output parameter indicates that the function block is requesting now the SNMP values cyclically.
xError	BOOL	Output	This output indicates with a rising edge that an error has occurred. You can read the corresponding error codes at the output parameters "wDiagCode" and "wAddDiagCode" during the cycle in which the rising edge occurs.
wDiagCode	WORD	Output	Indicates the cause of the error.
wAddDiagCode	DWORD	Output	Indicates additional information to the cause of the error.
iScanTimeAV	INT	Output	Actual scanning cycle time in seconds.
arrSnmpInfo	ASR_ARR_SNMP_INFO_1_30	Output	SNMP values from the specified managed switches. This output parameter is an Array of Struct, in which each field contains the SNMP values of a managed switch corresponding to the IP addresses at the input parameter “arrPartnerList”. The element “DataValid” indicates whether the last request of SNMP values to this managed switch was successful (TRUE) or not (FALSE). The meaning of the individual information read from the managed switches is explained separately in the table below.

Description of the elements of the structure “SnmpInfo”:

Most elements are available for each port and therefore presented in an Array.

Name	Data type	Description
SysUpTime	UDINT	Time in hundreds of seconds since the last reset of the managed switch.
PortSpeed	UDINT	Current bandwidth of the port in bits per second.
OperStatus	USINT	Current operational state of the port. Typical possible values: 1…LinkUp, 2…LinkDown, 3…Testing, 4…Unknown, 5…Dormant, 6…Not present
InOctets	UDINT	Total number of received octets.
InUcasts	UDINT	Total number of received unicast packets.
InNUcasts	UDINT	Total number of received broad and multicast packets.
InDiscards	UDINT	Total number of discarded inbound packets.
InErrors	UDINT	Total number of inbound packets with errors.
InUtilization	REAL	Calculated inbound utilization in percent.
InNUcastsPerSec	UINT	Calculated number of received broad and multicast packets per second.
InCorrupt	REAL	Calculated error rate of received packets in percent.
OutOctets	UDINT	Total number of transmitted octets.
OutUcasts	UDINT	Total number of transmitted unicast packets.
OutNUcasts	UDINT	Total number of transmitted broad and multicast packets.
OutDiscards	UDINT	Total number of discarded outbound packets.
OutErrors	UDINT	Total number of outbound packets with errors.
OutUtilization	REAL	Calculated outbound utilization in percent.
OutNUcastsPerSec	UINT	Calculated number of transmitted broad and multicast packets per second.
OutCorrupt	REAL	Calculated error rate of transmitted packets in percent.

5.2 Error Codes

Error Code	Description
C401	Indicates that the syntax of an applied string for the IP address at the input parameter "arrPartnerList" is wrong (proper syntax: '/IP=www.xxx.yyy.zzz'). The output parameter “wAddDiagCode” indicates the index of the relevant switch.
C402	Indicates that the SNMP-Client function block reported an error while a SNMP Request was sent to a managed switch. The output parameter "wAddDiagCode" contains the DiagCode of the SNMP-Client function block.
C403	Time-out detection of the internal state machine (3s). The output parameter "wAddDiagCode" indicates the step of the internal state machine (High-Byte) and the index of the relevant managed switch (Low-Byte) at which this error has occurred.

The acknowledgment of these error messages is carried out at a rising edge of the input parameter "xQuit".