Engineering Note

IND360 Precision

PROFINET PLC

METTLER TOLEDO

Contents

1. Overview

This Engineering Note is based on integration of Mettler Toledo's Compact Automation Transmitter/ Terminal IND360 Precision with a Profinet PLC. Go to www.mt.com/ind-ind360-downloads to download all the necessary files and documents.

Note: The configuration used in this sample code is based on the default settings：

Siemens TIA Portal V14 SP1

SAI data format: 2-Block format

Device Name: (configured by Profinet PLC)

IP Address: (configured by Profinet PLC)

IND360 device firmware version： 1.05.0012_4.3.0.5;

GSDML file for IND360 Precision version： GSDML-V2.35-MT-IND360-20200527

It is recommended to integrate one IND360 Precision into the PLC Profinet network and go through the sample code to understand the functionality of each Function Block. To add more IND360 Precision into the Profinet network, follow the steps listed in Chapter 6. Steps to Add New IND360s.

2. Setup of Project Development Environment

2.1. Hardware Integration

There are two industrial Ethernet ports (X1.1 and X1.2) on the IND360 Precision. Either port can be used to connect to the PLC. Multiple IND360 Precision's can be connected in one daisy chain network.

2.2. LLDP Function

LLDP (Link Layer Discovery Protocol) is a protocol used for topology discovery in the Siemens Profinet IO systems. It provides the option of communicating data between neighboring devices (e.g. device name, port, MAC address). IND360 POWERCELL Profinet models support this protocol.

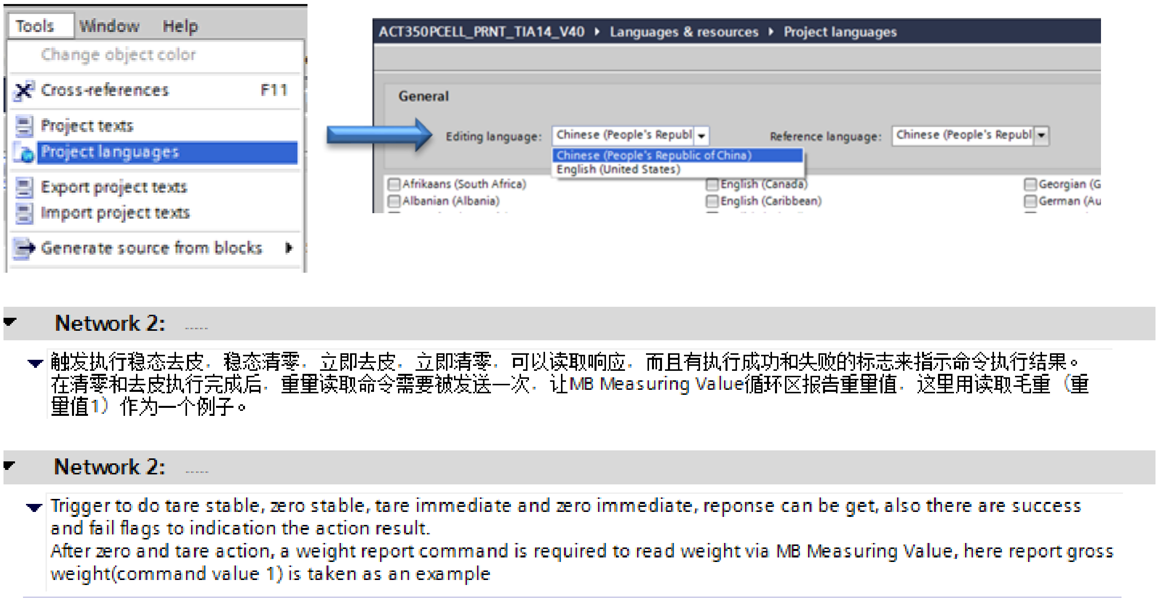
With LLDP, the downtime for IND360 POWERCELL replacement can be minimized. There is no need to reconfigure the device's IP Address and Device Name, as long as the new device is connected to the Profinet network via the same physical network port as the previous device.

2.3. Open the Sample Code

To open and use this sample code "IND360_PRECISION_PN.ap14", you need to use Siemens TIA Portal version 14 SP1 or higher. All the required GSDML files will be installed automatically when opening the sample code.

2.4. Switching Project Languages

Under Tools -> Project Languages -> Editing Language, choose the preferred language for your project. Selections are English (United States) and Chinese (People's Republic of China).



Editing language: Chinese (People's Republ

Chinese (People's Republic of China)
English (United States)

触发执行稳态去皮，稳态清零，立即去皮，立即清零，可以读取响应，而且有执行成功和失败的标志来指示命令执行结果。
在清零和去皮执行完成后，重量读取命令需要被发送一次，让MB Measuring Value循环区报告重量值，这里用读取毛重（重量值1）作为一个例子。

Trigger to do tare stable, zero stable, tare immediate and zero immediate, reponse can be get, also there are success and fail flags to indication the action result.
After zero and tare action, a weight report command is required to read weight via MB Measuring Value, here report gross weight(command value 1) is taken as an example

Figure 2-1: Switching Project Languages between English and Chinese

2.5. Select the correct controller model

There are three projects included in one sample code, each project uses different Siemens PLC model:

1. "S7-300" uses S7-300 series PLC with IND360 Precision Profinet;

2. "S7-1200" uses S7-1200 series PLC with IND360 Precision Profinet;
3. "S7-1500" uses S7-1500 series PLC with IND360 Precision Profinet

Choose the most relevant project according to your PLC type to download into the PLC.

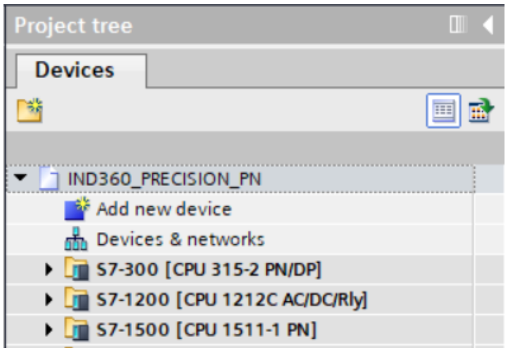


Figure 2-2： Three projects included in one sample code

To change the PLC model: Go to Device Configuration under the project folder, right click on the current controller, select "Change Device" and choose the new controller as well as its firmware version.

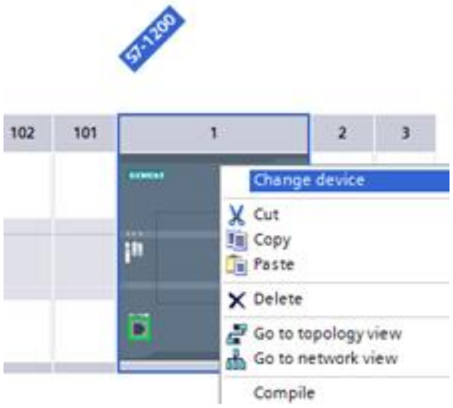
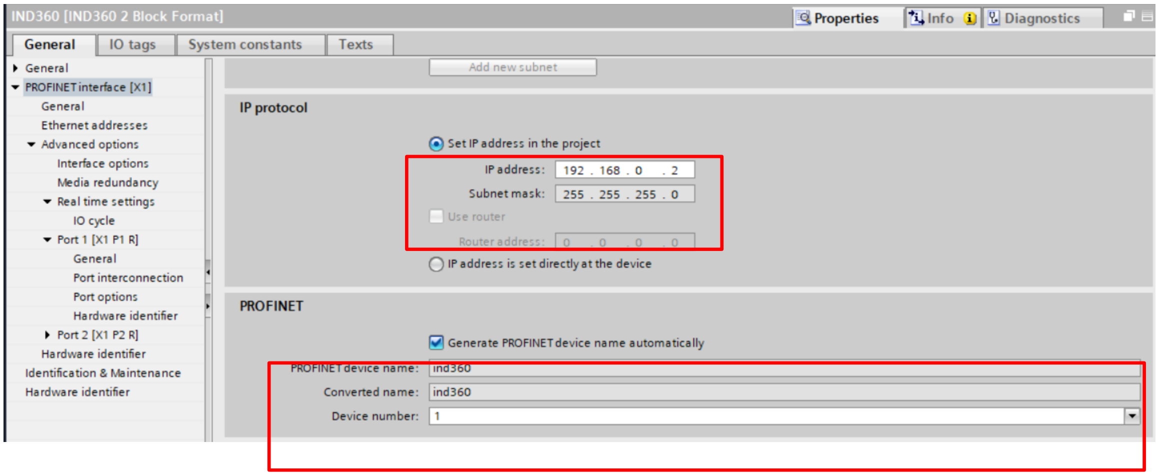


Figure 2-3: Change the controller type

Compile and download the project into the controller.

The IND360 Precision Device Name is set to be generated automatically and IP Address as "192.168.0.2". If different Device Name is required, please change and assign it accordingly. As for the PLC, its Device Name will be assigned according to its model while its IP Address is "192.168.0.10".



IP protocol

● Set IP address in the project

IP address: 192 . 168 . 0 . 2

Subnet mask: 255 . 255 . 255 . 0

☐ Use router

Router address: 0 . 0 . 0 . 0

○ IP address is set directly at the device

PROFINET

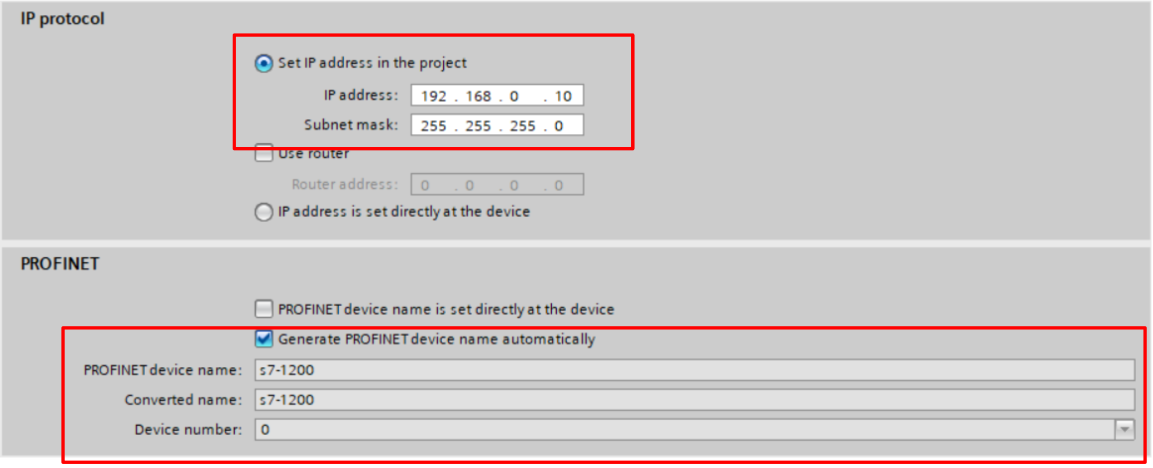
☑ Generate PROFINET device name automatically

PROFINET device name: ind360

Converted name: ind360

Device number: 1

Figure 2-4: IND360 Device Properties



IP protocol

● Set IP address in the project

IP address: 192 . 168 . 0 . 10

Subnet mask: 255 . 255 . 255 . 0

☐ Use router

Router address: 0 . 0 . 0 . 0

○ IP address is set directly at the device

PROFINET

☐ PROFINET device name is set directly at the device

☑ Generate PROFINET device name automatically

PROFINET device name: s7-1200

Converted name: s7-1200

Device number: 0

Figure 2-5: PLC Device Properties

Select the "MT_IND_Application" program, click on "Go Online" button to start using the sample code.

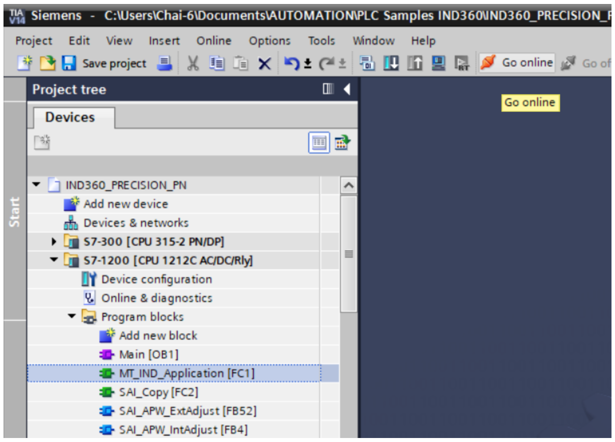
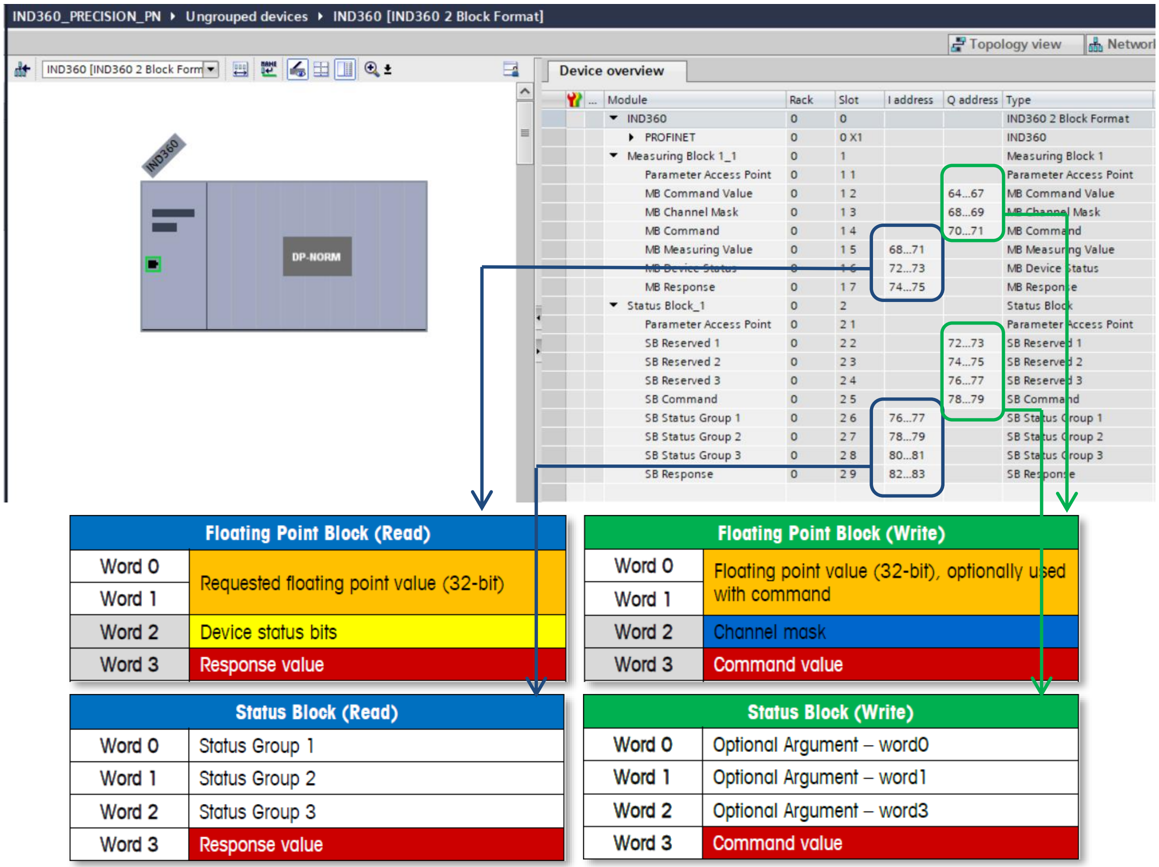


Figure 2-6: go online with MT_IND_Application

3. SAI Data Structure in Device Overview

In the Device Overview, the SAI input and output data structure has been assigned with the respective I and Q addresses as shown below. For more details on SAI data structure, please refer to the Standard Automation Interface user manual, which is downloadable from the IND360 Download Page www.mt.com/ind-ind360-downloads.


IND360_PRECISION_PN ▸ Ungrouped devices ▸ IND360 [IND360 2 Block Format]
Topology view
Networ
IND360 [IND360 2 Block Form
IND360
DP-NORM
Device overview

Module	Rack	Slot	I address	Q address	Type
IND360	0	0			IND360 2 Block Format
PROFINET	0	0 X1			IND360
Measuring Block 1_1	0	1			Measuring Block 1
Parameter Access Point	0	1 1			Parameter Access Point
MB Command Value	0	1 2		64...67	MB Command Value
MB Channel Mask	0	1 3		68...69	MB Channel Mask
MB Command	0	1 4		70...71	MB Command
MB Measuring Value	0	1 5	68...71		MB Measuring Value
MB Device Status	0	1 6	72...73		MB Device Status
MB Response	0	1 7	74...75		MB Response
Status Block_1	0	2			Status Block
Parameter Access Point	0	2 1			Parameter Access Point
SB Reserved 1	0	2 2		72...73	SB Reserved 1
SB Reserved 2	0	2 3		74...75	SB Reserved 2
SB Reserved 3	0	2 4		76...77	SB Reserved 3
SB Command	0	2 5		78...79	SB Command
SB Status Group 1	0	2 6	76...77		SB Status Group 1
SB Status Group 2	0	2 7	78...79		SB Status Group 2
SB Status Group 3	0	2 8	80...81		SB Status Group 3
SB Response	0	2 9	82...83		SB Response

Floating Point Block (Read)	
Word 0	Requested floating point value (32-bit)
Word 1	
Word 2	Device status bits
Word 3	Response value

Floating Point Block (Write)	
Word 0	Floating point value (32-bit), optionally used with command
Word 1	
Word 2	Channel mask
Word 3	Command value

Status Block (Read)	
Word 0	Status Group 1
Word 1	Status Group 2
Word 2	Status Group 3
Word 3	Response value

Status Block (Write)	
Word 0	Optional Argument – word0
Word 1	Optional Argument – word1
Word 2	Optional Argument – word3
Word 3	Command value

Figure 3-1: SAI Data Structure as shown in the Device Overview

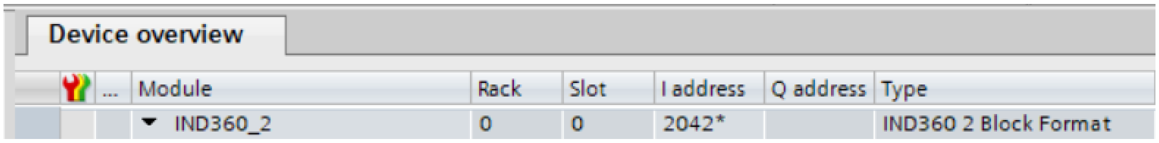
These I and Q addresses shown above will be used as input parameters in [4. Function Blocks](#4-function-blocks)

4. Function Blocks

About the "ID" input parameter for all the acyclic communication function blocks:

For all the function blocks which involve acyclic communication between the PLC and the weighing transmitter, the "ID" input parameter is required. Examples of function block with acyclic communication are zero adjustment, span adjustment and condition monitoring etc.

For an S7-300, ID can be found under the Device overview -> Diagnostics Address of Rack 0, Slot 0. In the example below the ID is "2042".



Device overview

Module	Rack	Slot	I address	Q address	Type
▾ IND360_2	0	0	2042*		IND360 2 Block Format

Figure 4-1: the ID parameter for S7-300

For S7-1200 and S7-1500 PLCs, the ID is the Hardware Identifier which can be identified as "(Device name)~Head".

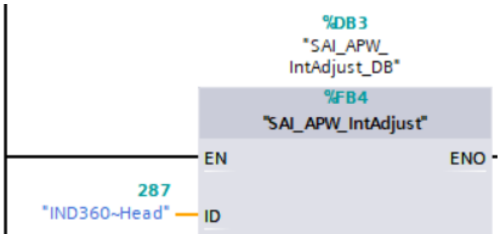


Figure 4-2: the ID parameter for S7-1200 and 1500

4.1. Cyclic Weight Data Processing

This function block reads in all the important real-time, cyclical weighing data such as weight value, Data OK bit, Motion bit, Net mode bit and critical alarm bit.

Set the scale command bit one at a time to trigger different commands such as tare stable, zero stable, tare immediate, zero immediate, preset tare and clear tare. A successful execution of a scale command will set the Done bit on, else the Error bit will be set on instead.

The cyclic weight data can be reported automatically right after any scale command. The type of weight data (gross, net, or tare) being reported depends on the setting for WeightCmd. By default, the WeightCmd is decimal "3" and the function block will return a net weight value every time after any scale command such as tare or zero. Similarly, if the WeightCmd parameter is configured as decimal "0" or "1" the function block will then return a gross weight after any scale command.

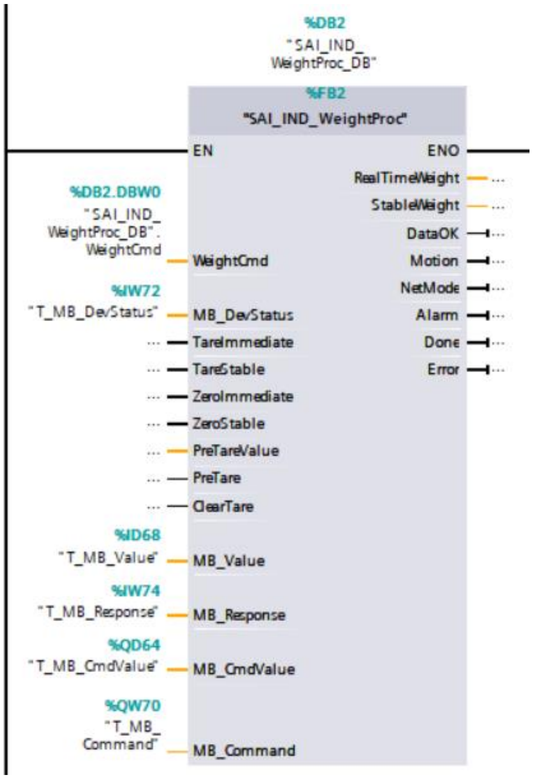


Figure 4-3: SAI_IND_WeightProc Function Block

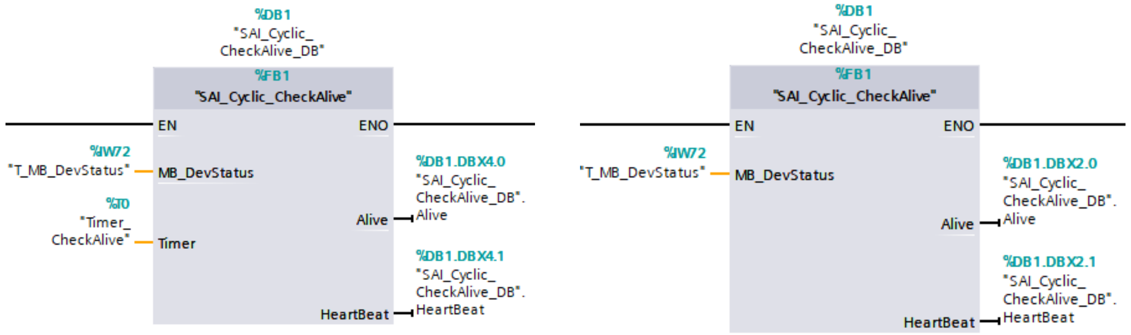
Table 4-1: SAI_IND_WeightProc Function Block Parameters

Input Parameters	Data Type	Values	Description
WeightCmd	Word	0, 1	Report gross weight value
		2	Report tare weight value
		3 (default)	**Report net weight value**
		5	Report gross weight value (with internal resolution)
		6	Report tare weight value (with internal resolution)
		7	Report net weight value (with internal resolution)
MB_DevStatus	Word		Refer to Device Overview, input address of MB Device Status
TareImmediate	Bool		Trigger this bit to perform immediate tare command. This tare command doesn't check for stability criteria. Upon completion of this command, the input bit will be reset.
TareStable	Bool		Trigger this bit to perform stable tare command. This tare command requires the weight value to remain stable within the stability criteria (configurable in web browser or APW-Link) for a predefined timeout range, failing which, the command will return an error. Upon completion of this command, the input bit will be reset.
ZeroImmediate	Bool		Trigger this bit to perform immediate zero command. The zero command can only be executed when the weight value is within the zero range. Else, the command will return an error. Upon completion of this command, the input bit will be reset.
ZeroStable	Bool		Trigger this bit to perform a stable zero command. This zero command requires the weight value to remain stable within the stability criteria (configurable in web browser or APW-Link) for a

			predefined timeout range. Furthermore the weight value has to be in the zero range to trigger this command, failing either condition; the command will return an error. Upon completion of this command, the input bit will be reset.
PreTareValue	Real		The preset tare value which has to be configured before issuing the PreTare command. Valid PreTare value is between scale's zero point up to maximum capacity.
PreTare	Bool		Trigger this bit to perform a preset tare command. The PreTareValue has to be configured prior to issuing this PreTare command. Upon completion of this command, the input bit will be reset.
ClearTare	Bool		Trigger this bit to perform a clear tare command. This command removes the tare and brings the scale into gross mode. Upon completion of this command, the input bit will be reset.
MB_Value	Real		Refer to Device Overview, input address of MB Measuring Value
MB_Response	Word		Refer to Device Overview, input address of MB Response
MB_CmdValue	Real		Refer to Device Overview, output address of MB Command Value
MB_Command	Word		Refer to Device Overview, output address of MB Command
Output Parameters	**Data Type**	**Values**	**Description**
RealTimeWeight	Real		Real-time weight value, can be gross, tare or net weight
StableWeight	Real		Stable weight value, the last real-time weight during Motion = 0
DataOK	Bool	0	This bit gets set to 0 when the device is still operational but the value being reported cannot be guaranteed to be valid. The following conditions cause the Data Okay bit to be set to 0: • Device is powering up • Device is in setup mode • Device is in test mode • Over capacity condition occurs - When the A/D converter is at its limit - Product dependent over capacity that occurs when the device determines it cannot trust the weight • Under capacity condition occurs - When the A/D converter is at its limit - Product dependent under capacity that occurs when the device determines it cannot trust the weight
		1	Weight data is normal, valid
Motion	Bool	0	Weight value is stable
		1	Weight value is in motion
NetMode	Bool	0	Weighing is in gross mode
		1	Weighing is in net mode
Alarm	Bool	0	No alarm
		1	Also called the RedAlert alarm. If this bit is true it is an indication that the control device should stop until the source of the alarm is evaluated and corrected. The control system should use a Field Value command or evaluate the RedAlert status block to determine the nature of the alarm.
Done	Bool	0	Zero, tare or clear tare command is in process, or failed
		1	Zero, tare or clear tare command is successful
Error	Bool	0	Zero, tare or clear tare command is in process, or succeeded
		1	Zero, tare or clear tare command is not completed due to error

4.2. Device Heart Beat Monitoring

This function block monitors the Heart Beat bit of the weighing transmitter and outputs an "Alive" flag.



S7-300 **S7-1200 & S7-1500**

Figure 4-4: SAI_Cyclic_CheckAlive Function Block

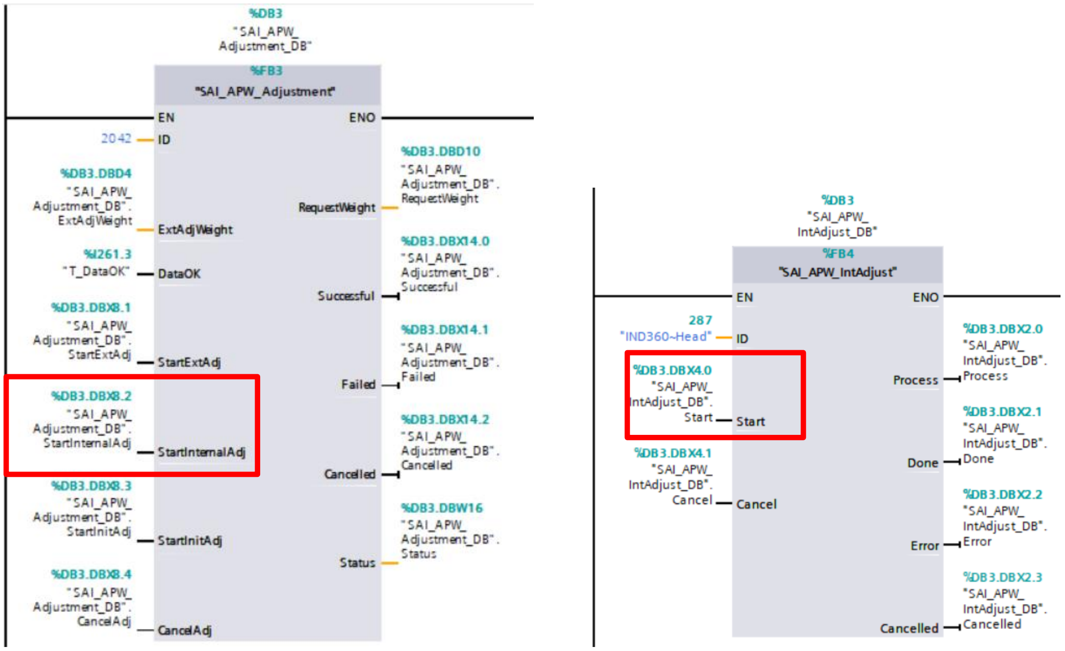
Table 4-2: SAI_Cyclic_CheckAlive Function Block Parameters

Input Parameters	Data Type	Values	Description
MB_DevStatus	Word		Refer to Device Overview, input address of MB Device Status
Timer (S7-300)	Timer		Timer, use independent timer for each function block, do not replicate.
Output Parameters	Data Type	Values	Description
Alive	Bool	0	Device has lost communication
		1	Device is communicating OK
HeartBeat	Bool		To insure that the device is working as expected and updating data in Words 0, 1 and 2, this heart beat bit is toggled between off and on states. The frequency is dependent on the specific device's ability to cycle this bit. For example, a 1 second heart beat would be sufficient for most applications.

4.3. Internal Adjustment

Automated Precision Weighing "APW" module with internal adjustment feature can be adjusted without applying any external weight.

Trigger the "Start" input bit to start the internal adjustment process. Upon completion of the adjustment process, this "Start" bit will be reset.


%DB3
"SAI_APW_Adjustment_DB"
%FB3
"SAI_APW_Adjustment"
EN
ENO
2042
ID
%DB3.DBD4
"SAI_APW_Adjustment_DB".ExtAdjWeight
ExtAdjWeight
%I261.3
"T_DataOK"
DataOK
%DB3.DBX8.1
"SAI_APW_Adjustment_DB".StartExtAdj
StartExtAdj
%DB3.DBX8.2
"SAI_APW_Adjustment_DB".StartInternalAdj
StartInternalAdj
%DB3.DBX8.3
"SAI_APW_Adjustment_DB".StartInitAdj
StartInitAdj
%DB3.DBX8.4
"SAI_APW_Adjustment_DB".CancelAdj
CancelAdj
%DB3.DBD10
"SAI_APW_Adjustment_DB".RequestWeight
RequestWeight
%DB3.DBX14.0
"SAI_APW_Adjustment_DB".Successful
Successful
%DB3.DBX14.1
"SAI_APW_Adjustment_DB".Failed
Failed
%DB3.DBX14.2
"SAI_APW_Adjustment_DB".Cancelled
Cancelled
%DB3.DBW16
"SAI_APW_Adjustment_DB".Status
Status
%DB3
"SAI_APW_IntAdjust_DB"
%FB4
"SAI_APW_IntAdjust"
EN
ENO
287
"IND360~Head"
ID
%DB3.DBX4.0
"SAI_APW_IntAdjust_DB".Start
Start
%DB3.DBX4.1
"SAI_APW_IntAdjust_DB".Cancel
Cancel
Process
%DB3.DBX2.0
"SAI_APW_IntAdjust_DB".Process
Done
%DB3.DBX2.1
"SAI_APW_IntAdjust_DB".Done
Error
%DB3.DBX2.2
"SAI_APW_IntAdjust_DB".Error
Cancelled
%DB3.DBX2.3
"SAI_APW_IntAdjust_DB".Cancelled

S7-300 **S7-1200 & S7-1500**

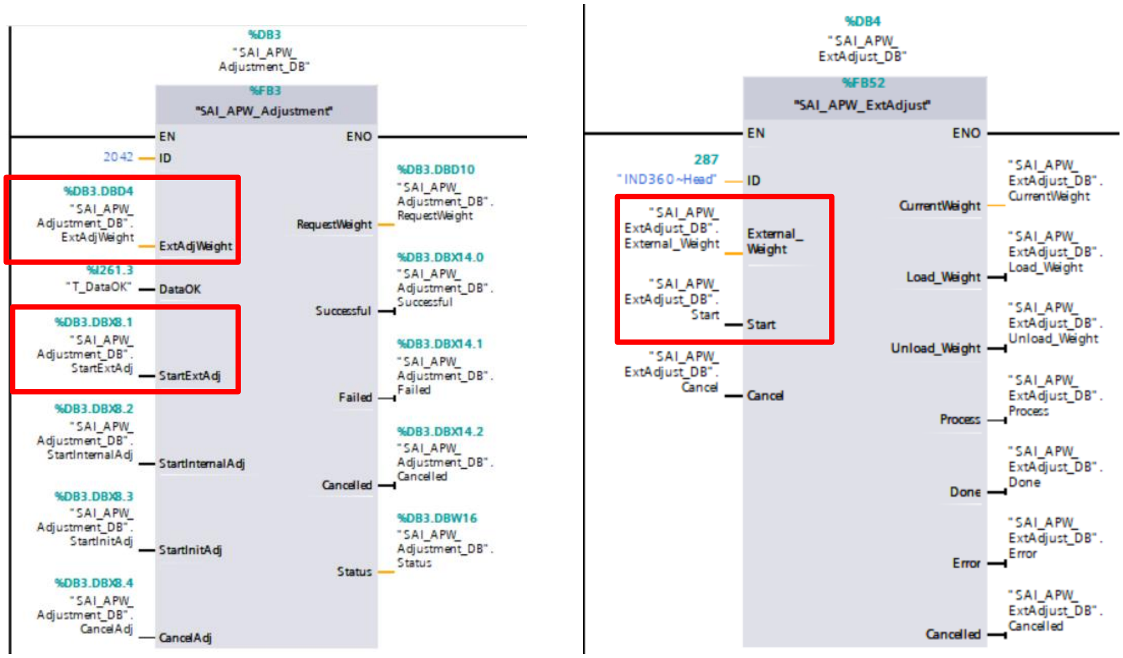
Figure 4-5: SAI_APW_IntAdjust Function Block

Table 4-3: SAI_APW_IntAdjust Function Block Parameters

Input Parameters	Data Type	Values	Description
ID (for S7-1200 and S7-1500)	HW_IO	Example: "IND360~Head"	ID parameter to select the module for which a data record is to be written. Use only the hardware identifier (HW ID) of the module for the ID parameter. In this sample program, the ID parameter of the device can be found under Device Properties > Slot 0 Hardware Identifier.
ID (S7-300)	DWORD	"2042"	
Start	Bool	1, 0	Trigger this input bit to start the adjustment process.
Cancel	Bool	1, 0	Trigger this input bit to cancel the adjustment process.
Output Parameters	Data Type	Values	Description
Process	Bool	1	Adjustment is started and in process
		0	Adjustment is not started
Done	Bool	1	Adjustment is completed successfully
		0	Adjustment is in process or in error state
Error	Bool	1	Adjustment failed due to error
		0	No error
Cancelled	Bool	1	Adjustment is cancelled successfully
		0	-

4.4. External Adjustment

Use this Function Block to perform scale adjustment with external weight.


%DB3
"SAI_APW_Adjustment_DB"
%FB3
"SAI_APW_Adjustment"
EN
ENO
2042
ID
%DB3.DBD4
"SAI_APW_Adjustment_DB".ExtAdjWeight
ExtAdjWeight
%I261.3
"T_DataOK"
DataOK
%DB3.DBX8.1
"SAI_APW_Adjustment_DB".StartExtAdj
StartExtAdj
%DB3.DBX8.2
"SAI_APW_Adjustment_DB".StartInternalAdj
StartInternalAdj
%DB3.DBX8.3
"SAI_APW_Adjustment_DB".StartInitAdj
StartInitAdj
%DB3.DBX8.4
"SAI_APW_Adjustment_DB".CancelAdj
CancelAdj
RequestWeight
%DB3.DBD10
"SAI_APW_Adjustment_DB".RequestWeight
Successful
%DB3.DBX14.0
"SAI_APW_Adjustment_DB".Successful
Failed
%DB3.DBX14.1
"SAI_APW_Adjustment_DB".Failed
Cancelled
%DB3.DBX14.2
"SAI_APW_Adjustment_DB".Cancelled
Status
%DB3.DBW16
"SAI_APW_Adjustment_DB".Status
%DB4
"SAI_APW_ExtAdjust_DB"
%FB52
"SAI_APW_ExtAdjust"
EN
ENO
287
"IND360~Head"
ID
"SAI_APW_ExtAdjust_DB".External_Weight
External_Weight
"SAI_APW_ExtAdjust_DB".Start
Start
"SAI_APW_ExtAdjust_DB".Cancel
Cancel
CurrentWeight
"SAI_APW_ExtAdjust_DB".CurrentWeight
Load_Weight
"SAI_APW_ExtAdjust_DB".Load_Weight
Unload_Weight
"SAI_APW_ExtAdjust_DB".Unload_Weight
Process
"SAI_APW_ExtAdjust_DB".Process
Done
"SAI_APW_ExtAdjust_DB".Done
Error
"SAI_APW_ExtAdjust_DB".Error
Cancelled
"SAI_APW_ExtAdjust_DB".Cancelled

S7-300 **S7-1200 & S7-1500**

Figure 4-6: SAI_APW_ExtAdjust Function Block

Configure the "External_Weight" according to the adjustment weight to be used. The weight unit is gram by default. Trigger the Start input bit to run the adjustment procedure.

Table 4-4: SAI_APW_ExtAdjust Function Block Parameters

Input Parameters	Data Type	Values	Description
ID (for S7-1200 and S7-1500)	HW_IO	Example: "IND360~Head"	ID parameter to select the module for which a data record is to be written. Use only the hardware identifier (HW ID) of the module for the ID parameter. In this sample program, the ID parameter of the device can be found under Device Properties > Slot 0 Hardware Identifier.
ID (S7-300)	DWORD	"2042"	
External_Weight	REAL (32 bits)	Example: "200.00"	This is the span weight value for the sensitivity adjustment.
Start	Bool	1, 0	Trigger this input bit to start the adjustment process.
Cancel	Bool	1, 0	Trigger this input bit to cancel/ abort the adjustment process after being started.
Output Parameters	**Data Type**	**Values**	**Description**
CurrentWeight	REAL (32 bits)	Example: "200.00"	The required reference weight here is shown here.
Load_Weight	Bool	1	User has to load the external weight according to the value displayed in CurrentWeight.
		0	No action required from the user
Unload_Weight	Bool	1	User has to unload the external weight
		0	No action required from the user
Process	Bool	1	Adjustment is started and in process
		0	Adjustment is not started

Done	Bool	1	Adjustment is completed successfully
		0	Adjustment is in process or in error state
Error	Bool	1	Adjustment failed due to error
		0	No error
Cancelled	Bool	1	Adjustment is cancelled successfully
		0	No cancellation

4.5. Alibi Record Access

This Function Block is used to execute ePrint to register the latest transaction into the alibi log. Also it provides user interface to read one alibi record at a time.

To read out a certain record number from the alibi log:

1. Write in the 'RecordNumber' to specify which alibi transaction entry to be read.

2. Trigger the 'Set_TransactionNum' bit to set the alibi record number to be read out.

3. Trigger the 'Get_Alibi' bit to read out the alibi transaction record specified by "RecordNum". Alibi record is read out as an array of 56 bytes, written into "Alibi_Array" data block.

If you don’t know the exact transaction number of the alibi record you want to read, executing step 3 without step 1 and 2. The latest entry in the alibi memory will be read.

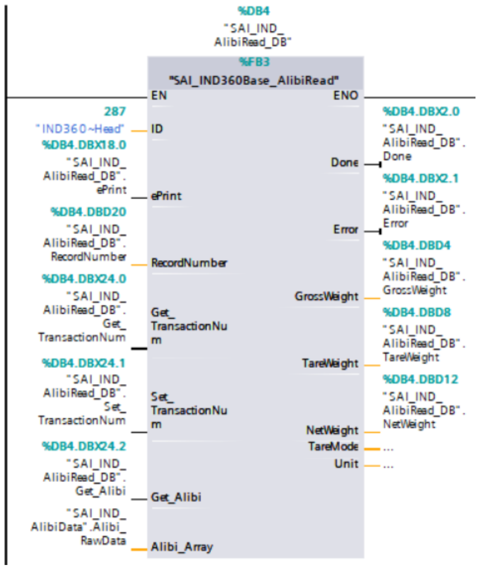

%DB4
"SAI_IND_ AlibiRead_DB"
%FB3
"SAI_IND360Base_AlibiRead"
EN
ENO
287
"IND360~Head"
ID
%DB4.DBX18.0
"SAI_IND_ AlibiRead_DB". ePrint
ePrint
%DB4.DBD20
"SAI_IND_ AlibiRead_DB". RecordNumber
RecordNumber
%DB4.DBX24.0
"SAI_IND_ AlibiRead_DB". Get_ TransactionNum
Get_ TransactionNum
%DB4.DBX24.1
"SAI_IND_ AlibiRead_DB". Set_ TransactionNum
Set_ TransactionNum
%DB4.DBX24.2
"SAI_IND_ AlibiRead_DB". Get_Alibi
Get_Alibi
"SAI_IND_ AlibiData".Alibi_ RawData
Alibi_Array
%DB4.DBX2.0
"SAI_IND_ AlibiRead_DB". Done
Done
%DB4.DBX2.1
"SAI_IND_ AlibiRead_DB". Error
Error
%DB4.DBD4
"SAI_IND_ AlibiRead_DB". GrossWeight
GrossWeight
%DB4.DBD8
"SAI_IND_ AlibiRead_DB". TareWeight
TareWeight
%DB4.DBD12
"SAI_IND_ AlibiRead_DB". NetWeight
NetWeight
TareMode
...
Unit
...

Figure 4-7: SAI_IND360Base_alibiRead Function Block

Table 4-5: SAI_IND360Base_alibiRead Function Block Parameters

Input Parameters	Data Type	Values	Description

ID	HW_IO	Example: "IND360~Head "	ID parameter to select the module for which a data record is to be written. Use only the hardware identifier (HW ID) of the module for the ID parameter. In this sample program, the ID parameter of the device can be found under Device Properties > Slot 0 Hardware Identifier.
ePrint	Bool		Trigger this bit to register the latest weighing transaction into the alibi log memory. Transaction number will be incremented.
RecordNumber	DINT		The alibi transaction record number which is to be read out from the weighing indicator. Specify the record number first before setting and reading alibi record.
Get_TransactionNum	Bool		Trigger this bit to get the previously read alibi record number. The alibi record number will be updated at the "RecordNum" field. This bit will be toggled off by the FB after reporting its execution status (DONE/ ERROR).
Set_TransactionNum	Bool		Trigger this bit to set the alibi record number which is to be read out from the weighing indicator. Ensure that the "RecordNum" has been specify before this step. This bit will be toggled off by the FB after reporting its execution status (DONE/ ERROR).
Get_Alibi	Bool		Trigger this bit to read out the alibi transaction record specified by "RecordNum". Alibi record is read out as an array of 56 bytes, written into "Alibi_Array" data block. This bit will be toggled off by the FB after reporting its execution status (DONE/ ERROR).

Alibi_Array	Array of Byte [0..55]		The array of 44 bytes to store the raw data of alibi record being read from the weighing indicator. This array is following of structure below:
			Struct(44 Bytes)
			{
			Byte[20]: Date & time; //ASCII String =>Byte0-19
			Long: Transaction Number; =>Byte20-23
			Float32: Gross Weight; =>Byte24-27
			Float32: Net Weight; =>Byte28-31
			Float32: Tare Weight; =>Byte32-35
			Byte: Tare Mode; =>Byte36
			Byte: Unit; =>Byte37
			Byte[6]: Not Used; =>Byte38-43
			}

Output Parameters	Data Type	Values	Description
Done	Bool	0	Get/ Set_TransactionNum or Get_Alibi command is not completed
		1	Get/ Set_TransactionNum or Get_Alibi command is completed
Error	Bool	0	Get/ Set_TransactionNum or Get_Alibi command gives no error
		1	Get/ Set_TransactionNum or Get_Alibi command is completed, with error
GrossWeight	REAL		This is the gross weight field extracted from the latest alibi record read. This variable is in double precision floating point format
TareWeight	REAL		This is the tare weight field extracted from the latest alibi record read. This variable is in double precision floating point format
NetWeight	REAL		This is the net weight field extracted from the latest alibi record read. This variable is in double precision floating point format
Unit	SINT		The weight unit used according the weighing indicator's setting.

			Refer to SAI Reference Manual – Secondary Scale Status unit bits for more details. Combination of 4 bits represent different weight unit [b4/b3/b2/b1], for e.g. [0/0/0/0] is unit 'g'; [0/0/0/1] is unit 'kg'; [0/0/1/0] is unit 'lb'

5. Sample Code Migration

5.1. Hardware Configurations

1) Under Devices & networks > Network view, add (or drag over) a new IND360 2 Block Format. Another option is the 8 Block Format, which offers more weight data to the PLC.

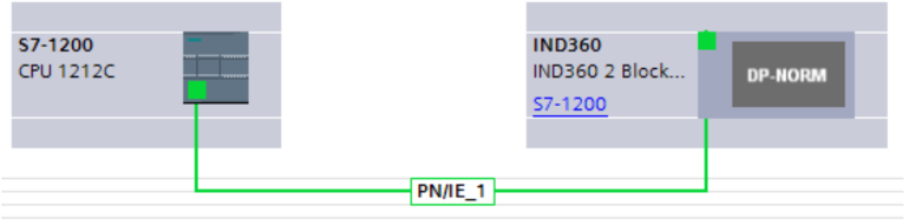
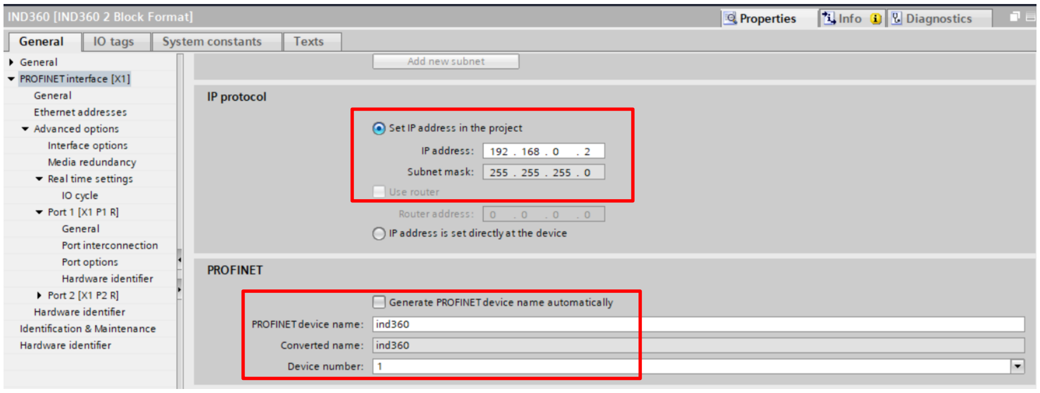


Figure 5-1: Add a Profinet device in the Network view

2) Assign the independent Profinet device name and IP address for the new device.



IP protocol

● Set IP address in the project

IP address: 192 . 168 . 0 . 2

Subnet mask: 255 . 255 . 255 . 0

☐ Use router

Router address: 0 . 0 . 0 . 0

○ IP address is set directly at the device

PROFINET

☐ Generate PROFINET device name automatically

PROFINET device name: ind360

Converted name: ind360

Device number: 1

Figure 5-2: Profinet device name and IP address

3) Under Devices & networks -> Topology view, link up the PLC and the IND360's network port 1 (left, NW1).

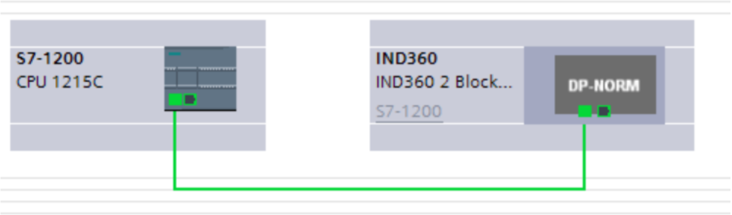
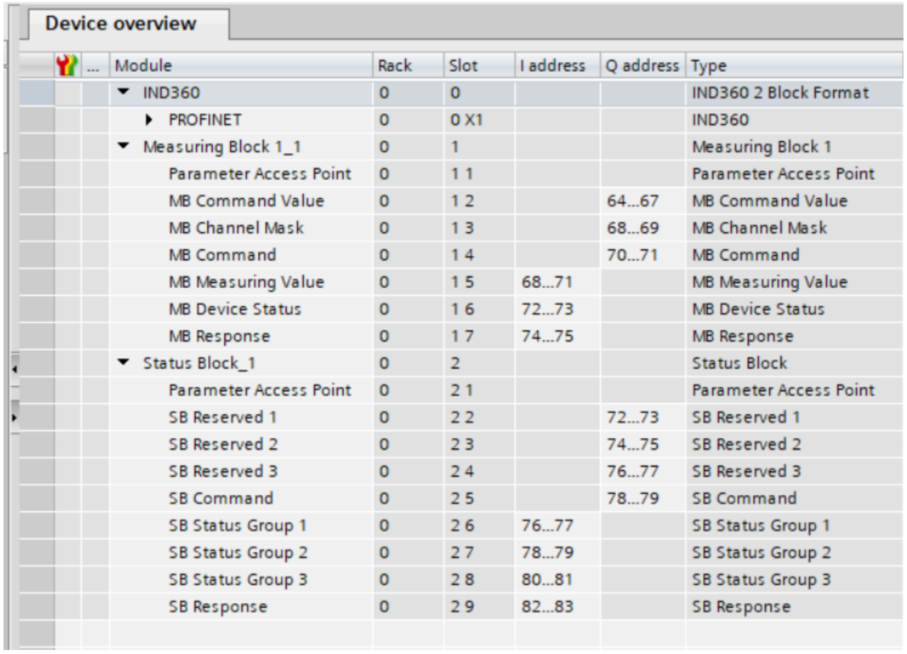


Figure 5-3: Devices & networks, Topology view

4) The sample code is following the default I and Q addresses assignment as shown below. To minimize the modification to the code, consider sticking to the same I and Q address assignment.



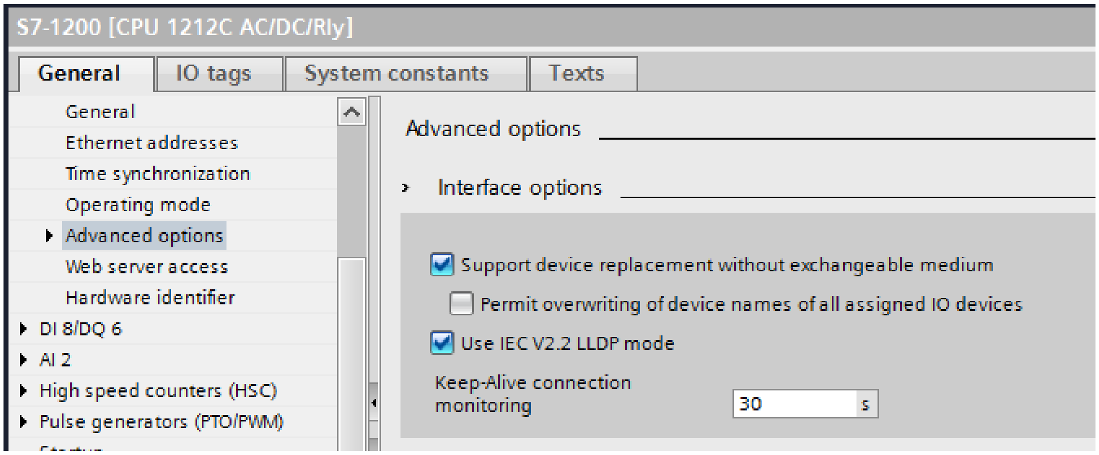
Device overview

Module	Rack	Slot	I address	Q address	Type
IND360	0	0			IND360 2 Block Format
PROFINET	0	0 X1			IND360
Measuring Block 1_1	0	1			Measuring Block 1
Parameter Access Point	0	1 1			Parameter Access Point
MB Command Value	0	1 2		64...67	MB Command Value
MB Channel Mask	0	1 3		68...69	MB Channel Mask
MB Command	0	1 4		70...71	MB Command
MB Measuring Value	0	1 5	68...71		MB Measuring Value
MB Device Status	0	1 6	72...73		MB Device Status
MB Response	0	1 7	74...75		MB Response
Status Block_1	0	2			Status Block
Parameter Access Point	0	2 1			Parameter Access Point
SB Reserved 1	0	2 2		72...73	SB Reserved 1
SB Reserved 2	0	2 3		74...75	SB Reserved 2
SB Reserved 3	0	2 4		76...77	SB Reserved 3
SB Command	0	2 5		78...79	SB Command
SB Status Group 1	0	2 6	76...77		SB Status Group 1
SB Status Group 2	0	2 7	78...79		SB Status Group 2
SB Status Group 3	0	2 8	80...81		SB Status Group 3
SB Response	0	2 9	82...83		SB Response

Figure 5-4: Device I and Q addresses

5.2. PLC Settings

1) Under the PLC device properties -> Advanced options, tick the below two options to support LLDP feature.



☑ Support device replacement without exchangeable medium

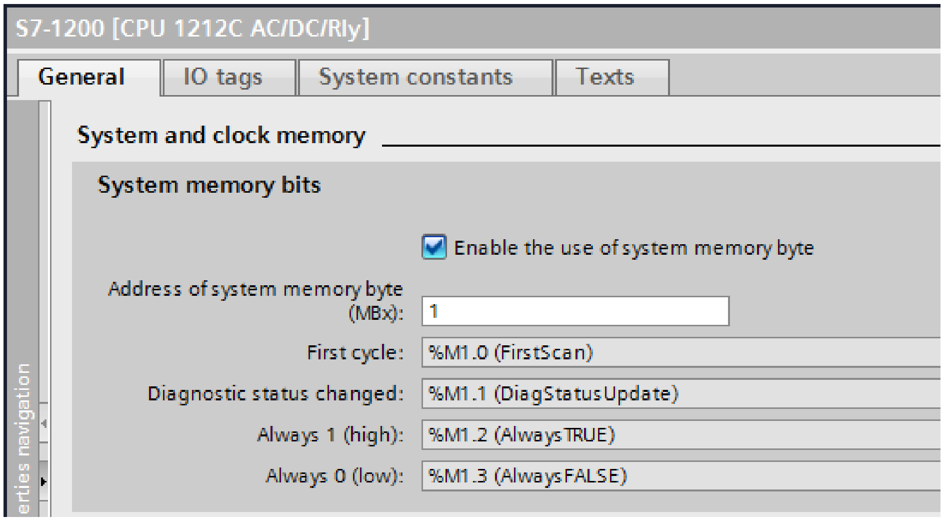
☐ Permit overwriting of device names of all assigned IO devices

☑ Use IEC V2.2 LLDP mode

Keep-Alive connection monitoring: 30 s

Figure 5-4: the LLDP feature

2) Under the PLC device properties -> System and clock memory, tick "Enable the use of system memory byte" (this feature is not available in the S7-300 series PLC).



☑ Enable the use of system memory byte

Address of system memory byte (MBx): 1

First cycle: %M1.0 (FirstScan)

Diagnostic status changed: %M1.1 (DiagStatusUpdate)

Always 1 (high): %M1.2 (AlwaysTRUE)

Always 0 (low): %M1.3 (AlwaysFALSE)

Figure 5-6: Enable system memory byte

5.3. Duplicate Programming Files

1) The required program blocks:

   a) MT_IND_Application(FC)

   b) SAI_Copy(FC)（for S7-1200 and S7-1500， not for S7-300）

   c) SAI_IND_WeightProc(FB), SAI_IND_WeightProc_DB

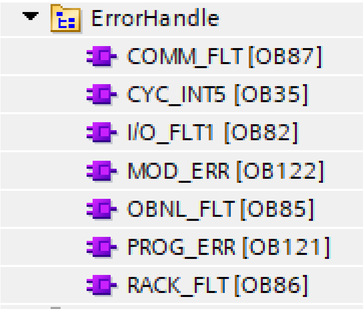
   d) SAI_Cyclic_CheckAlive(FB), SAI_Cyclic_CheckAlive_DB

   The function blocks below are used to perform scale adjustment from the PLC. All variants of IND360 now support scale adjustment via built-in web browser.

   a) SAI_APW_IntAdjust(FB), SAI_APW_IntAdjust_DB

   b) SAI_APW_ExtAdjust(FB), SAI_APW_ExtAdjust_DB

   For S7-300, need to add COMPLETE RESTART(OB100) and error handle programs as below, to support PROFINET auto reconnection feature.

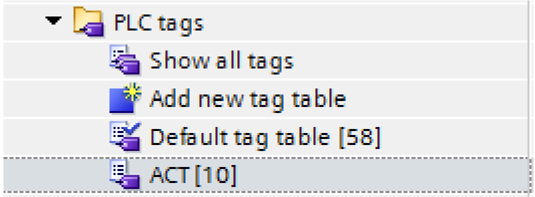


ErrorHandle
COMM_FLT [OB87]
CYC_INT5 [OB35]
I/O_FLT1 [OB82]
MOD_ERR [OB122]
OBNL_FLT [OB85]
PROG_ERR [OB121]
RACK_FLT [OB86]

Figure 5-7: Error handle programs of S7-300

2) Delete the other unused program blocks in MT_IND_Application.

3) Duplicate the "ACT" under the PLC tags.



PLC tags
Show all tags
Add new tag table
Default tag table [58]
ACT [10]

Figure 5-8: Duplicate the PLC tags

4) Duplicate all the PLC data types.

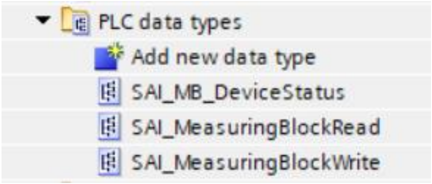


Figure 5-9: Duplicate the PLC data types

5) Lastly, in the Main (OB1) call up the function "MT_IND_Application".

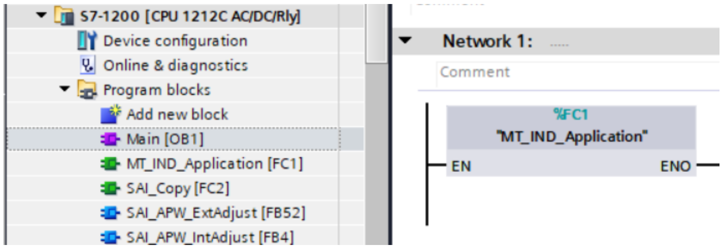


Figure 5-10: Call up "MT_IND_Application" in the Main OB

6. Add New IND360

In a Profinet system, each Profinet device is identified with different individual Device Name, the same rule applies to a network of multiple IND360 Precision.

1) In Devices and networks > Network View, add another IND360 2P 2 Block/ 8 Block structure.

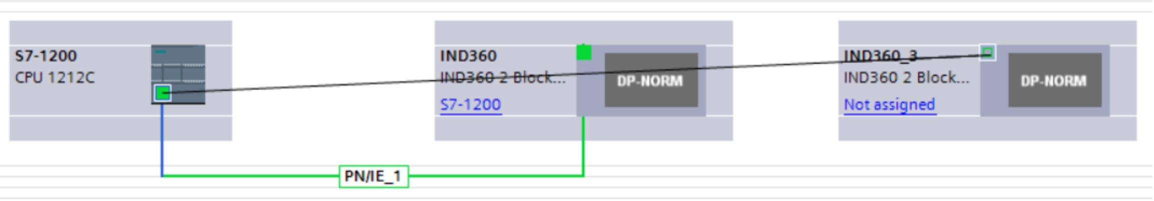
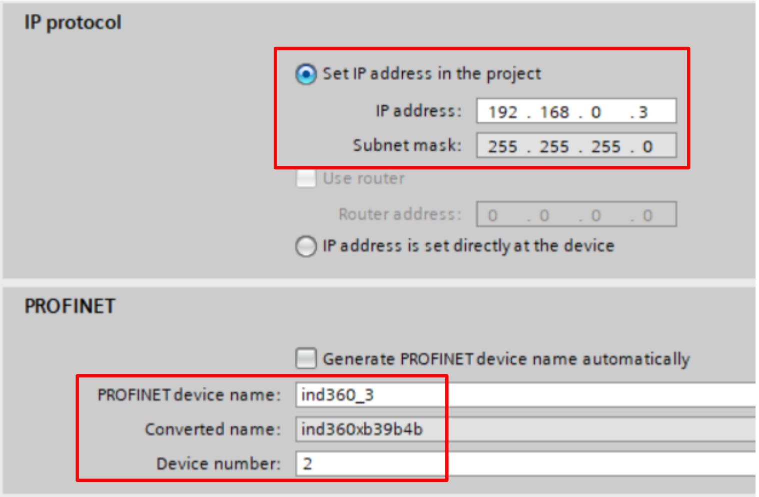


Figure 6-1: Add another IND360 into the network

2) Configure a dedicated PROFIENT device name and IP address to the new IND360. **Only use lower case letters for the device name**.



IP protocol

● Set IP address in the project

IP address: 192 . 168 . 0 . 3

Subnet mask: 255 . 255 . 255 . 0

☐ Use router

Router address: 0 . 0 . 0 . 0

○ IP address is set directly at the device

PROFINET

☐ Generate PROFINET device name automatically

PROFINET device name: ind360_3

Converted name: ind360xb39b4b

Device number: 2

Figure 6-2: PROFINET device name and IP address

3) In Devices and networks > Network overview, connect the device IND360's second Ethernet port NW2 to device IND360_3's first Ethernet port NW1.

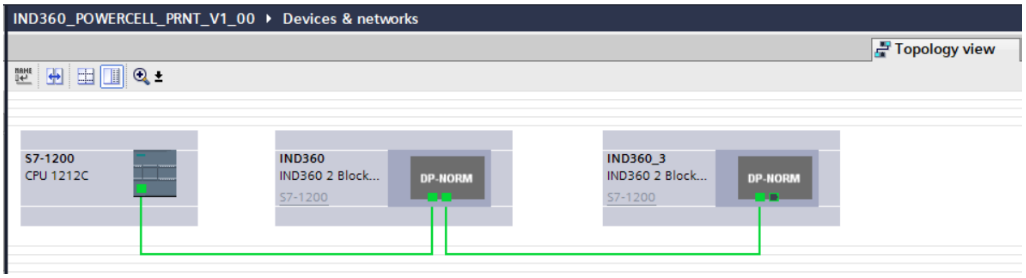
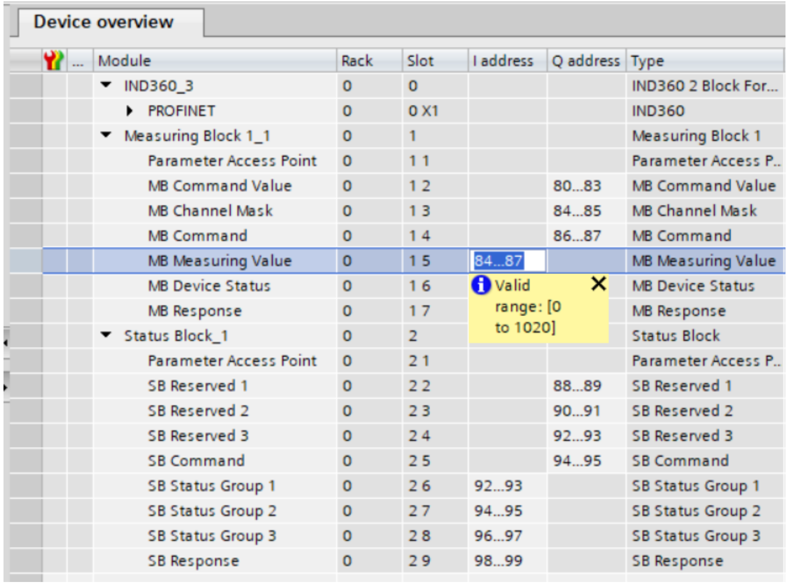


Figure 6-3: Connecting multiple IND360 in PROFINET network

4) When necessary, edit the automatically allocated I and Q addresses of the PROFINET device.



Device overview

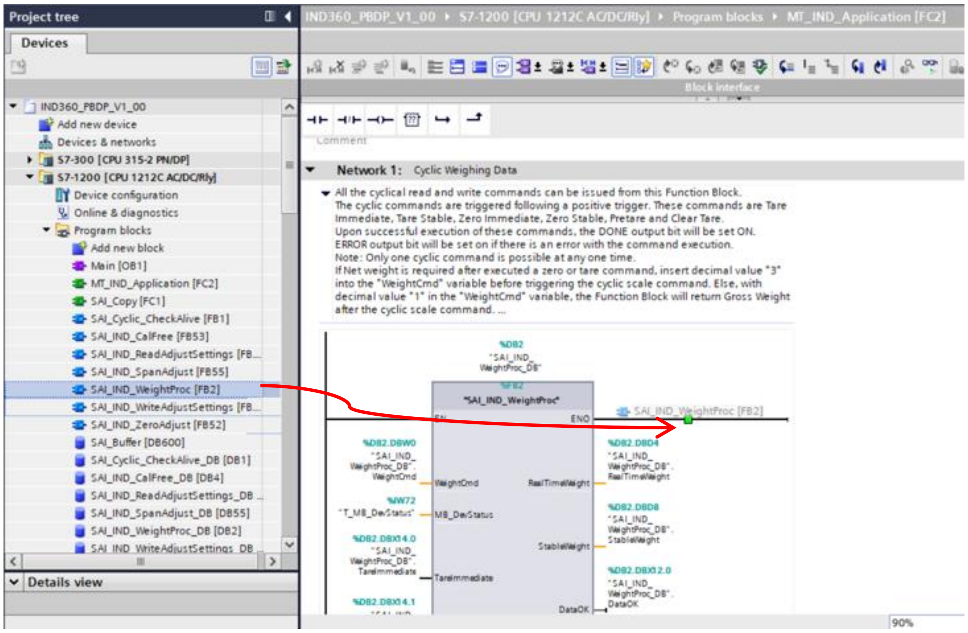
Module	Rack	Slot	I address	Q address	Type
▾ IND360_3	0	0			IND360 2 Block For...
▸ PROFINET	0	0 X1			IND360
▾ Measuring Block 1_1	0	1			Measuring Block 1
Parameter Access Point	0	1 1			Parameter Access P...
MB Command Value	0	1 2		80...83	MB Command Value
MB Channel Mask	0	1 3		84...85	MB Channel Mask
MB Command	0	1 4		86...87	MB Command
MB Measuring Value	0	1 5	84...87		MB Measuring Value
MB Device Status	0	1 6			MB Device Status
MB Response	0	1 7			MB Response
▾ Status Block_1	0	2			Status Block
Parameter Access Point	0	2 1			Parameter Access P...
SB Reserved 1	0	2 2		88...89	SB Reserved 1
SB Reserved 2	0	2 3		90...91	SB Reserved 2
SB Reserved 3	0	2 4		92...93	SB Reserved 3
SB Command	0	2 5		94...95	SB Command
SB Status Group 1	0	2 6	92...93		SB Status Group 1
SB Status Group 2	0	2 7	94...95		SB Status Group 2
SB Status Group 3	0	2 8	96...97		SB Status Group 3
SB Response	0	2 9	98...99		SB Response

Valid range: [0 to 1020]

Figure 6-4: I and Q Addresses

5) Duplicate the function blocks, and configure all the required input and output parameters. Each function block FB must have an independent data block DB. As shown below, there are two SAI_IND_WeightProc function blocks but both FBs are assigned with different DBs which are SAI_IND_WeightProc_DB (DB2) and SAI_IND_WeightProc_DB_1 (DB4)

A small trick can be used here to add adjacent function block, drag the function block from the Project Tree side window into the destination network.



Network 1: Cyclic Weighing Data

All the cyclical read and write commands can be issued from this Function Block.
The cyclic commands are triggered following a positive trigger. These commands are Tare Immediate, Tare Stable, Zero Immediate, Zero Stable, Pretare and Clear Tare.
Upon successful execution of these commands, the DONE output bit will be set ON.
ERROR output bit will be set on if there is an error with the command execution.
Note: Only one cyclic command is possible at any one time.
If Net weight is required after executed a zero or tare command, insert decimal value "3" into the "WeightCmd" variable before triggering the cyclic scale command. Else, with decimal value "1" in the "WeightCmd" variable, the Function Block will return Gross Weight after the cyclic scale command. ...

Figure 6-5: Duplicate function block for additional device

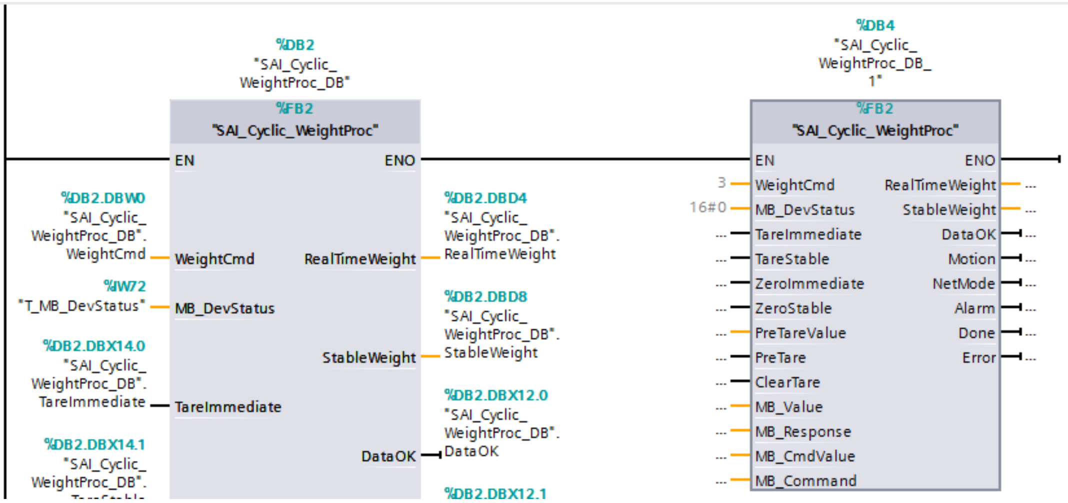


Figure 6-6: Two function blocks of the same type, but different data blocks

6) Repeat steps 1 – 5, until all the new IND360s have been integrated into the Profinet network.