Engineering Note

IND360 POWERCELL

PROFINET PLC

METTLER TOLEDO

Menu

1. Overview

This Engineering Note is based on integration of Mettler Toledo's Industrial Weighing Automation Terminal IND360 POWERELL 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: (empty); IP Address: (empty)

GSDML file： GSDML-V2.35-MT-IND360-20200527.xml;

It is recommended to integrate one IND360 POWERCELL into the PLC Profinet network and go through the sample codes to understand the functionality of each Function Block. To add more IND360 POWERCELL into the Profinet network, follow the steps listed in Chapter 5. Add New IND360 POWERCELL.

2. Setup of Project Development Environment

2.1. Hardware Integration

Connect the Ethernet cable from the PLC Ethernet port to IND360 POWERCELL industrial Ethernet port (X1.1 or X1.2).

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_POWERCELL_PN.ap14", you need to use Siemens TIA Portal version 14 SP1 or higher. All the required GSDML files will be installed automatically while 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).

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 POWERCELL weighing terminal;
2. "S7-1200" uses S7-1200 series PLC with IND360 POWERCELL weighing terminal;
3. "S7-1500" uses S7-1500 series PLC with IND360 POWERCELL weighing terminal;

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

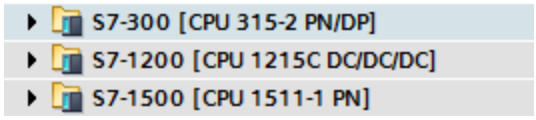


Figure 2-2: three projects in the sample code

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.

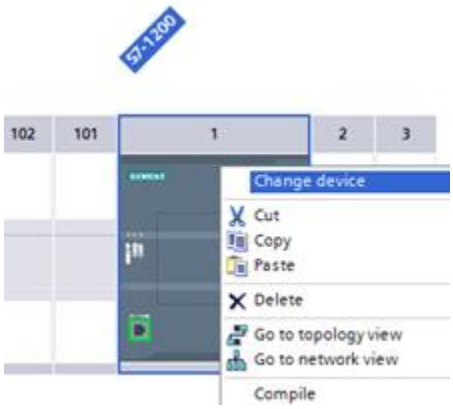


Figure 2-3: Change controller type

Compile and download the project into the controller.

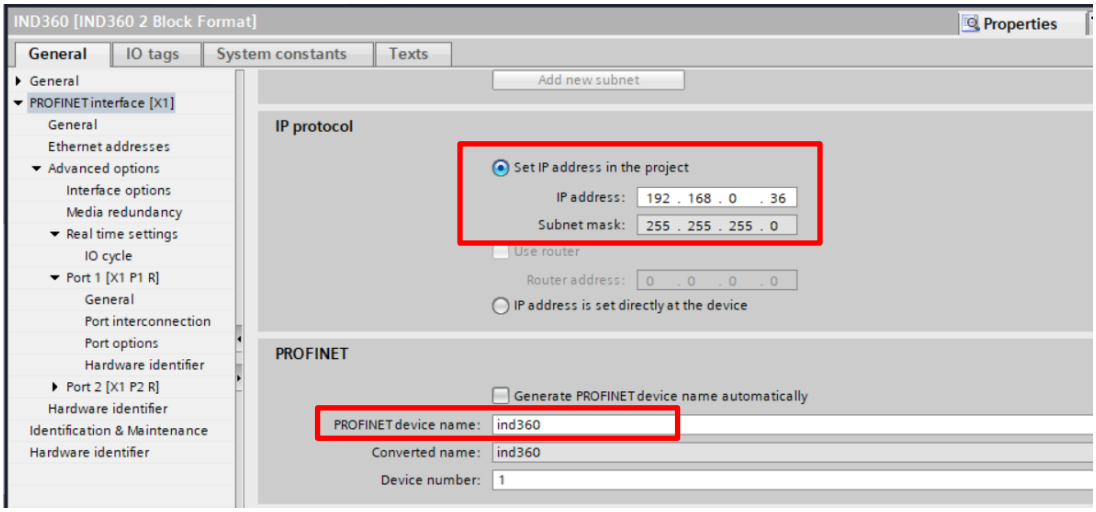


Figure 2-4: IND360 POWERCELL Device Properties – Ethernet Addresses

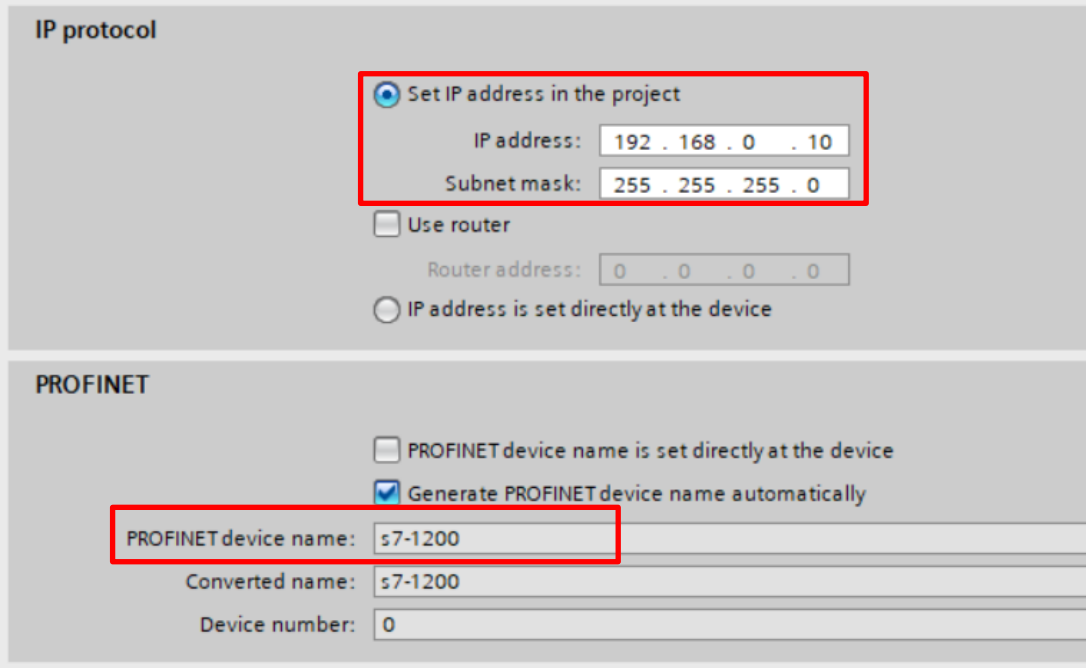


Figure 2-5: PLC Device Properties – Ethernet Addresses

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 User Manual: Standard Automation Interface: IND360 POWERCELL Transmitters English, which is downloadable from the IND360 Download Page www.mt.com/ind-ind360-downloads.

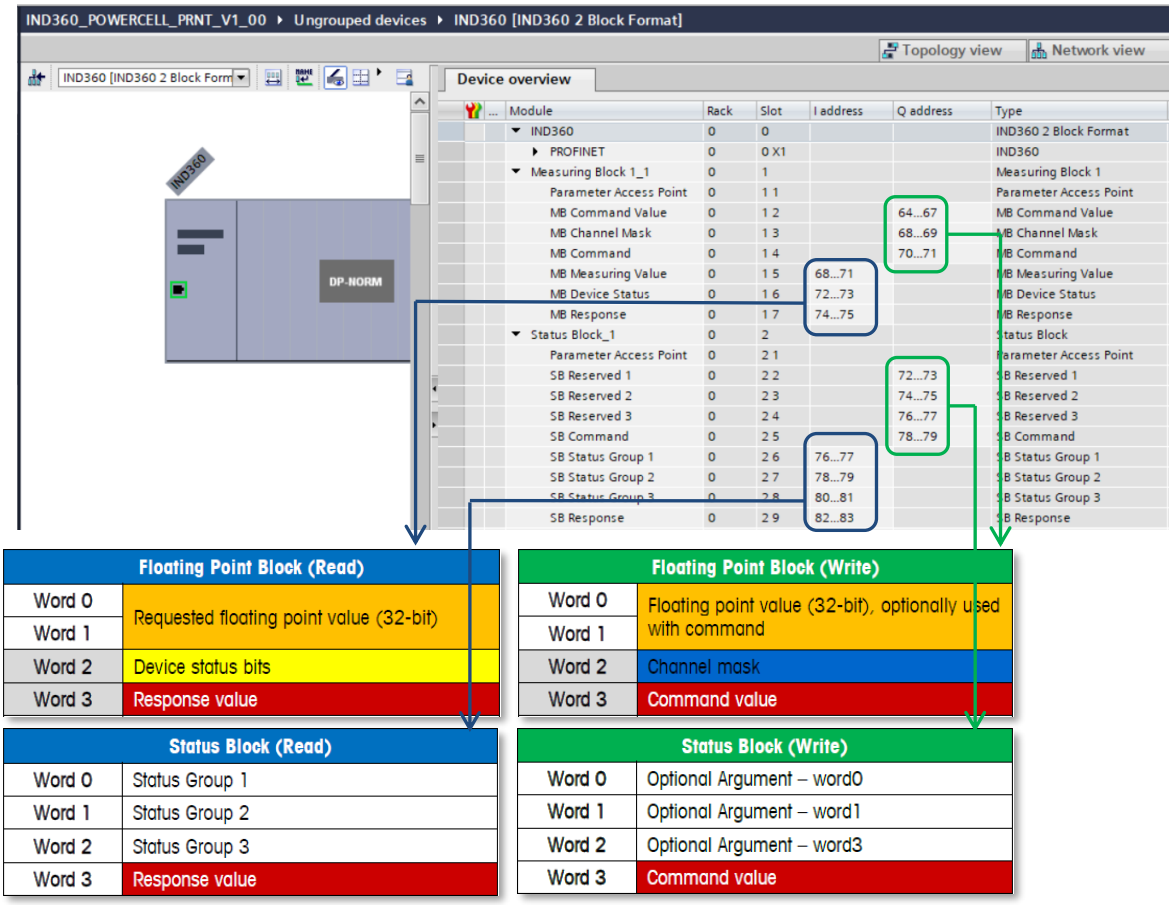


Figure 3-1: SAI Data Structure as shown in the Device Overview

The I and Q addresses above will be used as input parameters in 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".

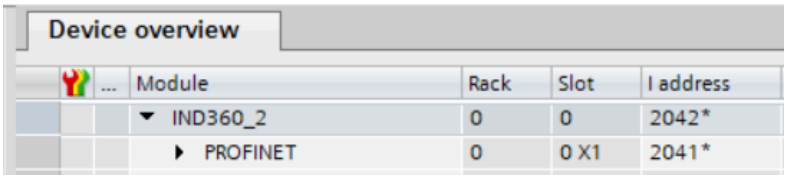


Figure 4-1: the ID parameter for S7-300

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

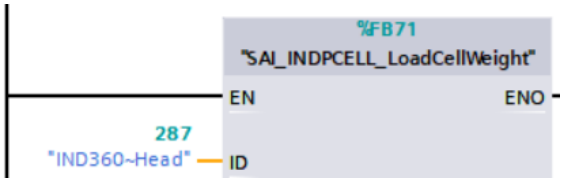


Figure 4-2: the ID parameter for S7-1200 and 1500

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 right 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.

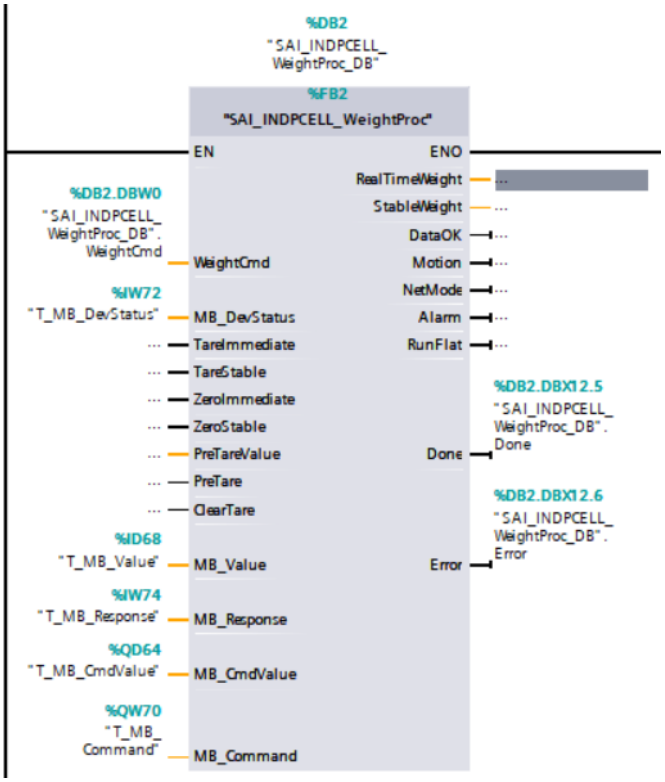


Figure 4-3: SAI_INDPCELL_WeightProc Function Block

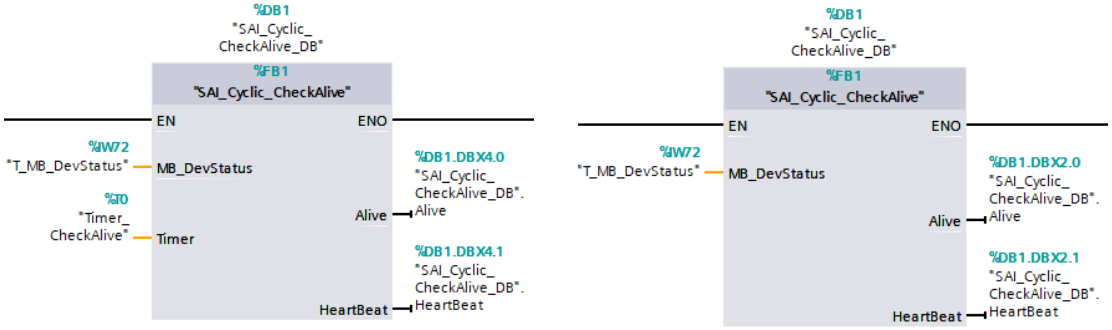
Table 4-1: SAI_INDPCELL_WeightProc Function Block Parameters

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 (+-1d within 0.3 second) for a predefined timeout range (3 seconds by default), 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 (+-2% by default). 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 (+-1d within 0.3 second) for a predefined timeout

			range (3 seconds by default). 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.
RunFlat	Bool	0	RunFlat is inactive
		1	RunFlat is active
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 和 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. Diagnostic Status Monitoring

This function block reads in all the critical real-time diagnostic data from Powercell™ load cells.

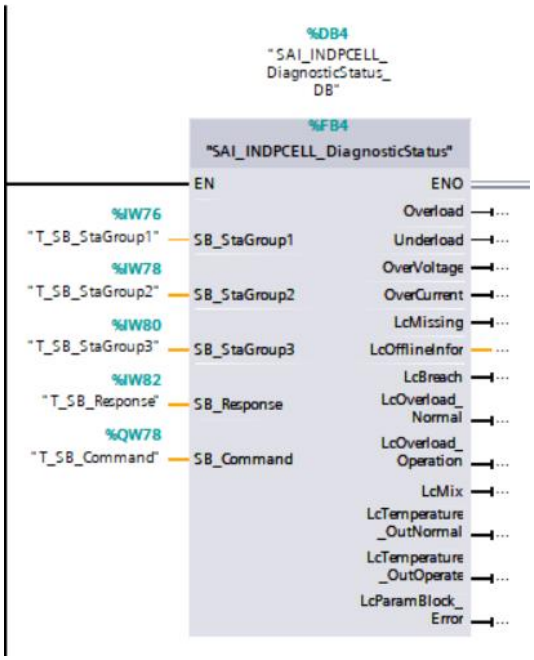


Figure 4-5: SAI_INDPCELL_DiagnosticStatus Function Block

Table 4-3: SAI_INDPCELL_DiagnosticStatus Function Block Parameters

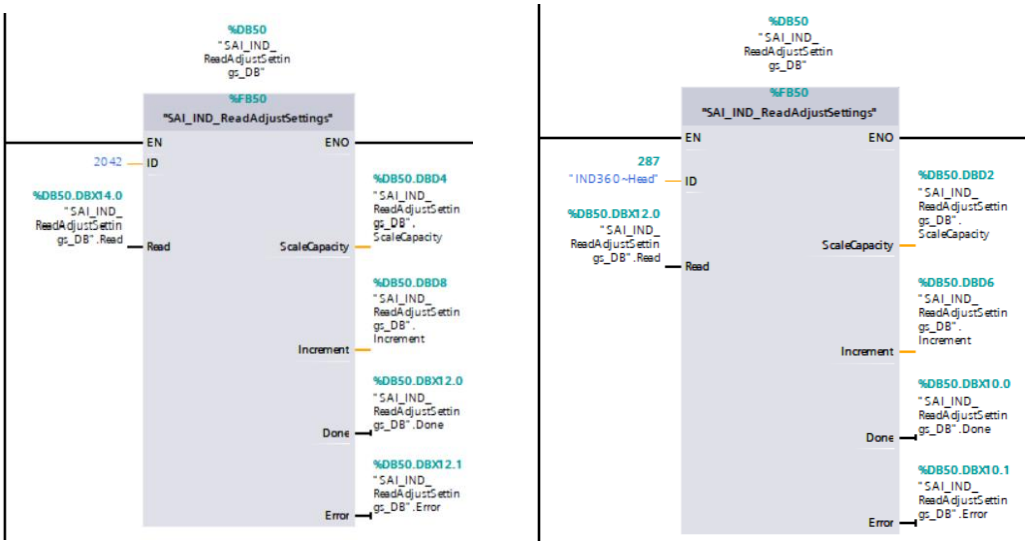
Input Parameters	Data Type	Values	Description
SB_StaGroup1	Word		Refer to Device Overview, input address of SB Status Group 1
SB_StaGroup2	Word		Refer to Device Overview, input address of SB Status Group 2
SB_StaGroup3	Word		Refer to Device Overview, input address of SB Status Group 3
SB_Response	Word		Refer to Device Overview, input address of SB Response
SB_Command	Word		Refer to Device Overview, output address of SB Command
Output Parameters	Data Type	Values	Description
Overload	Bool	0	Scale is not overload
		1	The weight is equal to or greater than a "customer-programmed" limit either on the scale (multi-sensor system) or individual sensor's capacity
Underload	Bool	0	Scale is not underload
		1	The weight is under the "customer-programmed" limit on the scale / sensor (under zero but still within A/D range)
OverVoltage	Bool	0	Operating voltage is normal
		1	Operating voltage out of range
OverCurrent	Bool	0	Operating current is normal
		1	Operating current is out of range
LcMissing	Bool	0	All load cells are communicating normally
		1	One or multiple load cell has lost communication
LcOfflineInfor	Word		A word of 16 bits, each bit represents the communication status of the Powercell™ load cell. Bit status "1" means the load cell has lost communication, while status "0" means no comm. lost. The IND360 POWERCELL™ supports up to 14 digital load cells hence only 14 bits (bit 0 – bit 13) are relevant in this Word. Word: 00XX XXXX XXXX XXXX

			|.....| LC#14 LC#1
LcBreach	Bool	0	No load cell enclosure breach
		1	the sensors enclosure has been compromised and therefore vulnerable to outside influences such as moisture / water – in most cases a failure will occur if the breach is not corrected or the sensor replaced
LcOverload _Normal	Bool	0	The load cell's individual weight is not within 101%-150% of its Rated Capacity
		1	The load cell's individual weight falls within 101%-150% of its Rated Capacity
LcOverload _Operation	Bool	0	The load cell's individual weight has not exceeded 150% of its Rated Capacity
		1	The load cell's individual weight exceeds 150% of its Rated Capacity
LcMix	Bool	0	All the connected load cells are of the same model
		1	There is at least a mix of different load cell model in a scale
LcTemperature _OutNormal	Bool	0	Load cell temperature is normal
		1	Load cell temperature is out of compensated range
LcTemperature _OutOperate	Bool	0	Load cell temperature is within the operating range
		1	Load cell temperature is out of operating range
LcParamBlock _Error	Bool	0	Load cell parameters are normal
		1	At least one of the load cell parameters has error

4.4. Read Scale Adjustment Settings

This function block reads the current scale capacity and increment values from the connected weighing transmitter. Set the "Read" input parameter on to start the reading process. Upon completion of the read process, this "Read" bit will be reset.

It is useful to know the current scale settings before performing any scale adjustment procedure.



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

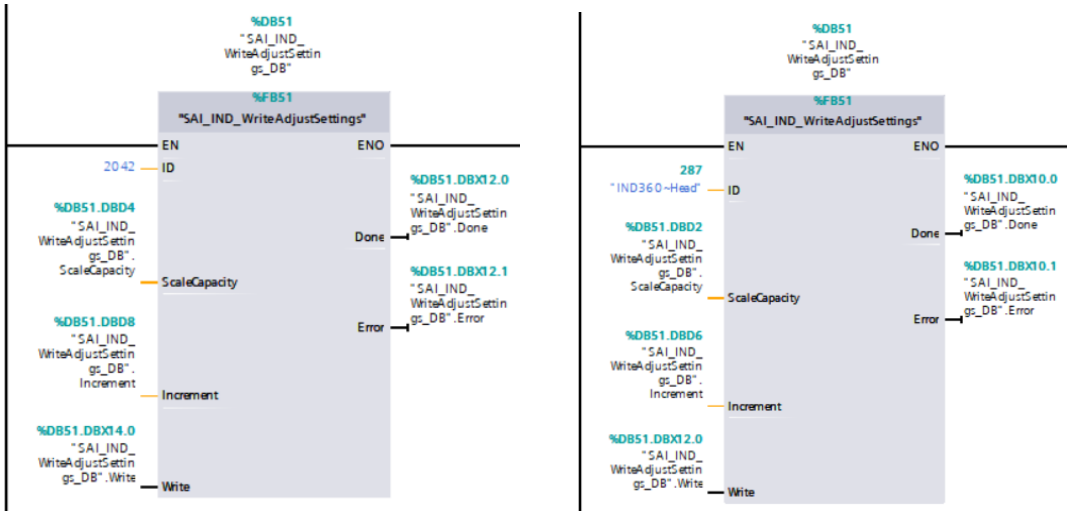
Figure 4-6: SAI_IND_ReadAdjustSettings Function Block

Table 4-4: SAI_IND_ReadAdjustSettings parameter descriptions

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"	
Read	Bool	1, 0	Trigger this input bit to start the reading process.
Output Parameters	**Data Type**	**Values**	**Description**
ScaleCapacity	REAL (32 bits)	Example: "3000.0"	Current scale capacity value
Increment	REAL (32 bits)	Example: "0.1"	Current scale increment value
Done	Bool	1	Read process is completed successfully
		0	Read process is not completed
Error	Bool	1	An error has occurred during the read process
		0	No error

4.5. Write Scale Adjustment Settings

This Function Block configures the new settings of scale capacity and increment value onto the weighing transmitter. Even though all IND360 weighing transmitters now support scale configuration through its built-in web server, the PLC can also overwrite these scale settings.
The scale resolution (scale capacity/ increment) has to be within the range of 500 – 100 000.



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

Figure 4-7: SAI_IND_WriteAdjustSettings Function Block

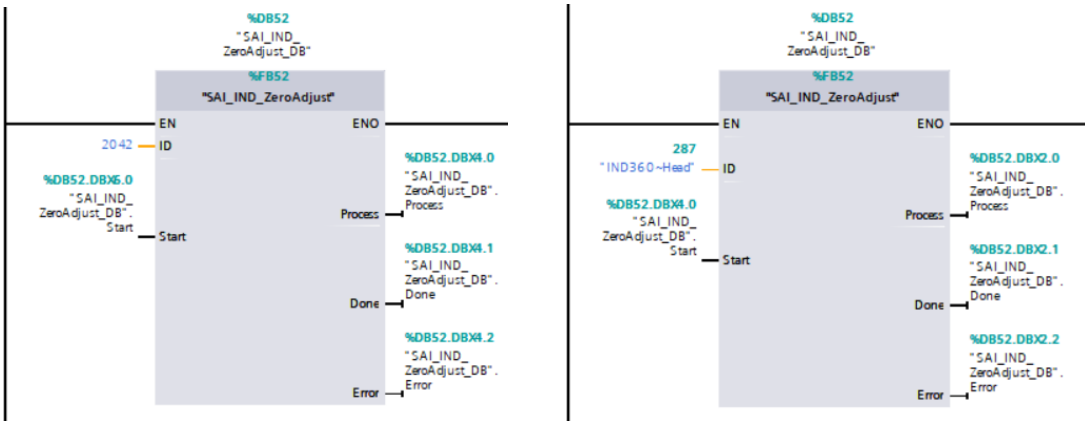
Table 4-5: SAI_IND_WriteAdjustSettings parameter descriptions

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"	
ScaleCapacity	REAL (32 bits)	Example: "3000.0"	New scale capacity value
Increment	REAL (32 bits)	Example: "0.1"	New scale increment value
Write	Bool	1, 0	Trigger this input bit to start the writing process.
Output Parameters	Data Type	Values	Description
Done	Bool	1	Write process is completed successfully
		0	Write process is not completed
Error	Bool	1	An error has occurred during the write process
		0	No error

4.6. Zero Adjustment

Zero calibration has to be performed first before CalFree+ or span calibration. Make sure the scale is empty before starting this zero calibration procedure.

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



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

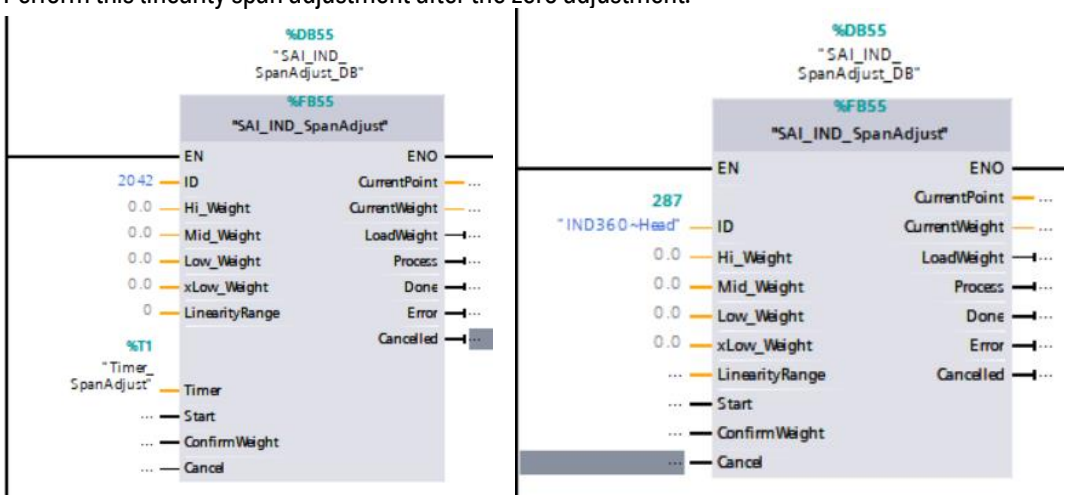
Figure 4-8: SAI_IND_ZeroAdjust Function Block

Table 4-6: SAI_IND_ZeroAdjust 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 calibration 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

4.7. Span Adjustment

Perform this linearity span adjustment after the zero adjustment.



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

Figure 4-9: SAI_IND_SpanAdjust Function Block

If only 2 points adjustment (zero, span) is required, only configure the highest reference weight (span) into this Function Block. In this case, the span is the second linearity point.
The first reference point is always the zero reference which has to be adjusted prior to this.

If linearity adjustment is required, up to 4 points can be set-up. The table below shows all the possible selection of linearity adjustment and the required input parameters for this Function Block.

LinearityRange settings:	Required reference weight(s), cannot be zero:
"0", 2-point (zero, span)	Hi_Weight
"1", 3-point linearity	Hi_Weight, Mid_Weight
"2", 4-point linearity	Hi_Weight, Mid_Weight, Low_Weight
"3", 5-point linearity	Hi_Weight, Mid_Weight, Low_Weight, xLow_Weight

Table 4-7: SAI_IND_SpanAdjust Linearity Range Settings

Notes:

- The Function Block will return an error if the reference weights are not configured according to the linearity range setting.
- The Function Block will return an error if the required reference weight(s) is zero or not in the correct ascending order when starting the adjustment process.

The flow chart below explains the linearity adjustment process flow according to different selection of linearity range:

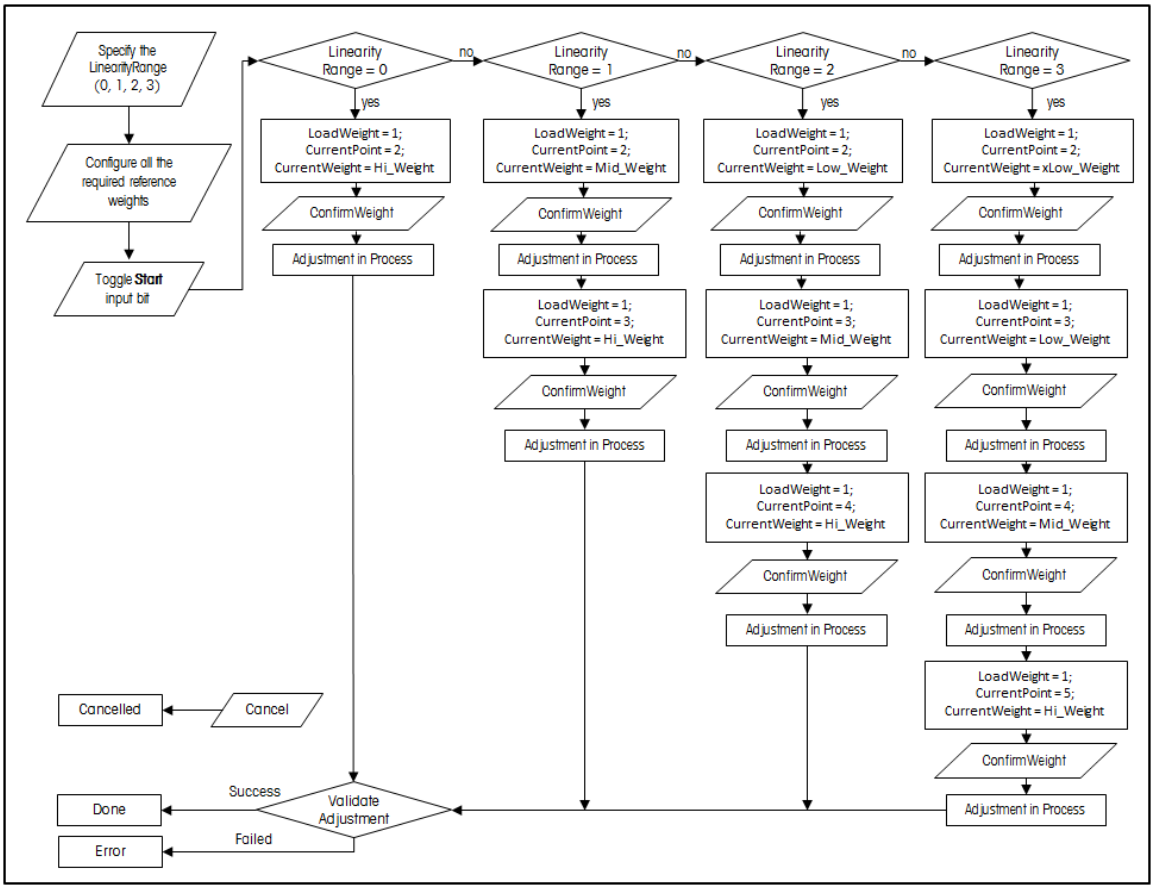


Figure 4-10: SAI_IND_SpanAdjust Flow Chart

Configure the required Linearity Range and all the respective reference weights. Set the Start bit on to run the adjustment process. Wait for the LoadWeight output bit to turn on and then load the reference weight according to CurrentWeight value. After the new reference weight has been loaded, set the ConfirmWeight bit on to proceed with adjustment. Repeat the same sequence for the rest of the reference weights until the adjustment process is completed. The adjustment process can be cancelled at any point of time after started.

Table 4-8: SAI_IND_SpanAdjust 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"	
Hi_Weight	REAL (32 bits)	Example: "800.00"	The highest reference weight in linearity calibration. For a 2-point calibration, this is the span value.
Mid_Weight	REAL (32 bits)	Example: "600.00"	For a 5-point calibration, this is the 4th reference point. For a 3-point calibration, this is the 2nd reference point.
Low_Weight	REAL (32 bits)	Example: "400.00"	For a 5-point calibration, this is the 3rd reference point. For a 4-point calibration, this is the 2nd reference point.

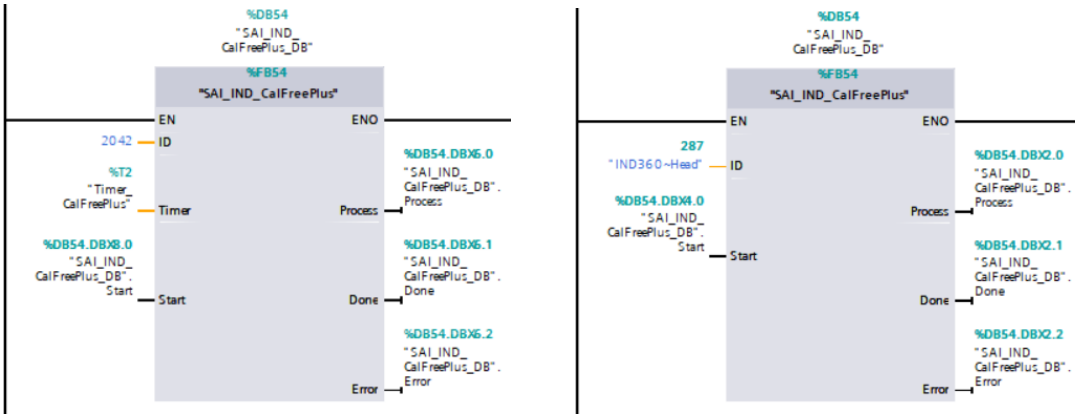
xLow_Weight	REAL (32 bits)	Example: "200.00"	The lowest reference weight value in linearity calibration. Only used when the linearity range is configured to "3" – 5-point linearity.
LinearityRange	INT	0, 1, 2, 3	Decimal "0" – 2-point; Decimal "1" – 3-point; Decimal "2" – 4-point; Decimal "3" – 5-point
Start	Bool	1, 0	Trigger this input bit to start the calibration process.
ConfirmWeight	Bool	1, 0	User has to trigger this input bit after loading the "CurrentWeight" onto the scale. This bit serves as an acknowledgement flag for the Function Block to proceed to next steps. The Function Block will reset this bit automatically.
Cancel	Bool	1, 0	Trigger this input bit to cancel/ abort the calibration process after being started.
Output Parameters	Data Type	Values	Description
CurrentPoint	INT	Example: "2"	The Function Block updates the current reference point here.
CurrentWeight	REAL (32 bits)	Example: "400.00"	The Function Block updates the required reference weight here.
LoadWeight	Bool	1	User has to load a new reference weight according to the value displayed in CurrentWeight.
		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

4.8. CalFree+

The IND360 POWERCELL transmitter provides a method to calibrate a scale without using test weights. This is based on the POWERCELL load cell rated capacity and count value. This method can be used for initial check-out and testing of systems or when a large structure is used as the weighing vessel and it is not possible to apply test weights to the structure.

METTLER TOLEDO highly recommends that the test weights or RapidCal™ method be used whenever possible as these methods provide the most accurate calibration accuracy.

Set the Start bit on to run the CalFree+ adjustment. Upon completion of the adjustment process, this Start bit will be reset.



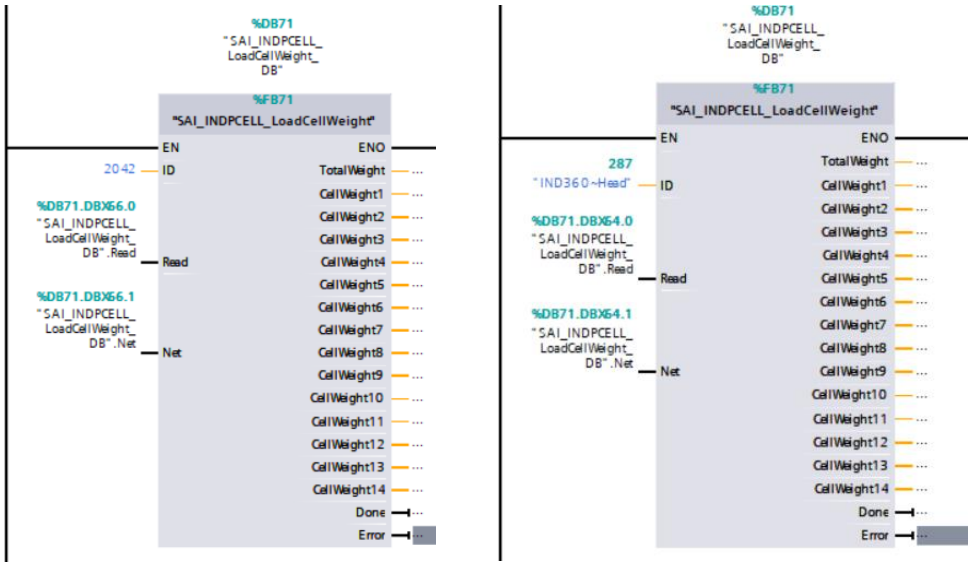
S7-300 **S7-1200 & S7-1500**

Figure 4-11: SAI_IND_CalFreePlus Function Block

Table 4-9: SAI_IND_CalFreePlus 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 calibration process. Upon completion of the calibration (succeeded or failed) this input bit will be reset by the Function Block itself.
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

4.9. Read Individual Load Cell Weight Value (gross or net)



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

Figure 4-12: SAI_INDPCELL_LoadCellWeight Function Block

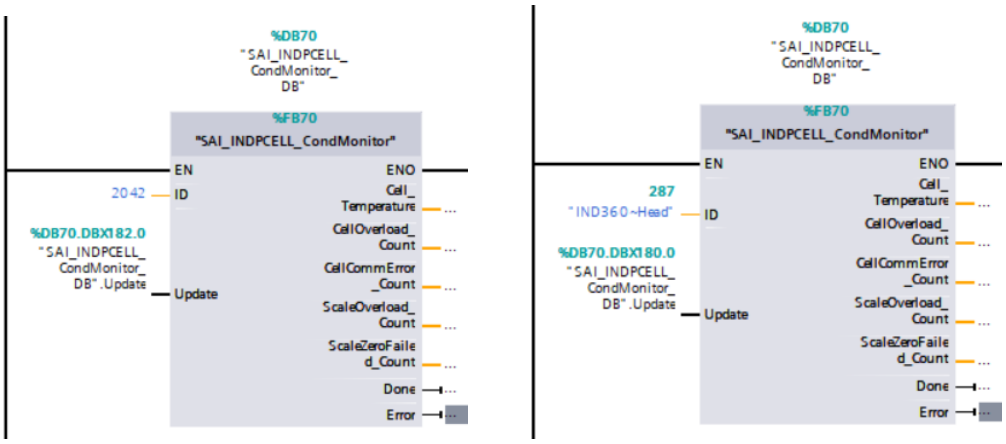
Only in a POWERCELL® scale, the control system is able to read individual load cell weight value.
The availability of these individual load cell weight values can be used to monitor the tank/ platform load distribution. Uneven load distribution might be caused by permanent feeder and machinery attached to the scale or when weighing powder or granules. Nonetheless, the load distribution for each load cell should not differ too much from the others. When huge disparity of load distribution is detected, there could be mechanical structure failure on the scale.

Input Parameters	Data Type	Description
ID	HW_IO	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	
Read	Bool	Trigger this input bit to start the read process.
Net	Bool	Net = 0; Read individual load cell gross weight Net = 1; Read individual load cell net weight
Output Parameters	**Data Type**	**Description**
TotalWeight	REAL (32 bits)	The total gross or net weight of all the load cells combined
CellWeight1	REAL (32 bits)	The load cell #1's gross or net weight
CellWeight2	REAL (32 bits)	The load cell #2's gross or net weight
CellWeight3	REAL (32 bits)	The load cell #3's gross or net weight
CellWeight4	REAL (32 bits)	The load cell #4's gross or net weight
CellWeight5	REAL (32 bits)	The load cell #5's gross or net weight
CellWeight6	REAL (32 bits)	The load cell #6's gross or net weight
CellWeight7	REAL (32 bits)	The load cell #7's gross or net weight

CellWeight8	REAL (32 bits)	The load cell #8's gross or net weight
CellWeight9	REAL (32 bits)	The load cell #9's gross or net weight
CellWeight10	REAL (32 bits)	The load cell #10's gross or net weight
CellWeight11	REAL (32 bits)	The load cell #11's gross or net weight
CellWeight12	REAL (32 bits)	The load cell #12's gross or net weight
CellWeight13	REAL (32 bits)	The load cell #13's gross or net weight
CellWeight14	REAL (32 bits)	The load cell #14's gross or net weight
Done	Bool	1; The read process is completed successfully
		0; The read process is in process or there is an error
Error	Bool	1; There is an error during the read process
		0; No error

Table 4-10: SAI_INDPCELL_LoadCellWeight Function Block Parameters

4.10. POWERCELL® Condition Monitoring



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

Figure 4-13: SAI_INDPCELL_CondMonitor Function Block

The IND360 Industrial Transmitter in combination with the POWERCELL® system including POWERCELL-based tank, silo and PowerDeck™ scales offers Condition Monitoring features such as scale overload/underload, missing load cell, network failure and individual load cell overload monitoring.

On this Function Block, trigger the "Update" input bit to start gathering all the diagnostic data from both the weighing transmitter and the connected POWERCELL® load cells.

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"	
Update	Bool	1, 0	Trigger this input bit to start updating all the monitoring data.
Output Parameters	Data Type		Description

Cell_Temperature	ARRAY [0..13] of REAL	Example: [23.5, 23.5, 23.4, 23.7, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]	This is an array of 14 load cells' temperature value. Temperature plays an important part in load cell overall accuracy as it has effect on the load cell's minimum dead load output and sensitivity. Always keep the load cell operating temperature within the Compensated Temperature Range.
CellOverload_Count	ARRAY [0..13] of DINT	Example: [0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	This is an array of 14 load cells' overload counts. When a load cell gross weight falls within 100-150% of its rated capacity, this overload count will be incremented. High count of load cell overload will shorten its life span.
CellCommError_Count	ARRAY [0..13] of DINT	Example: [2, 1, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	This is an array of 14 load cells' communication lost count. When a load cell lost its communication with the weighing transmitter, this count will be incremented once. This communication lost could be due to the load cell cable or a degrading load cell.
ScaleOverload_Count	DINT	Example: "5"	This is the scale overload count. When the scale gross weight overshoots its user-defined capacity range, this overload count will be incremented.
ScaleZeroFailed_Count	DINT	Example: "11"	This is the number of time the push button zero command has failed, due to out of zero range or scale in motion.
Error	Bool	0, 1	This output bit will be set ON if an error occurred during the read process
Cancelled	Bool	0, 1	This output bit will be set ON if the read process is completed successfully.

Table 4-11: SAI_ACTPCELL_CondMonitor Function Block Parameters

4.11. 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.

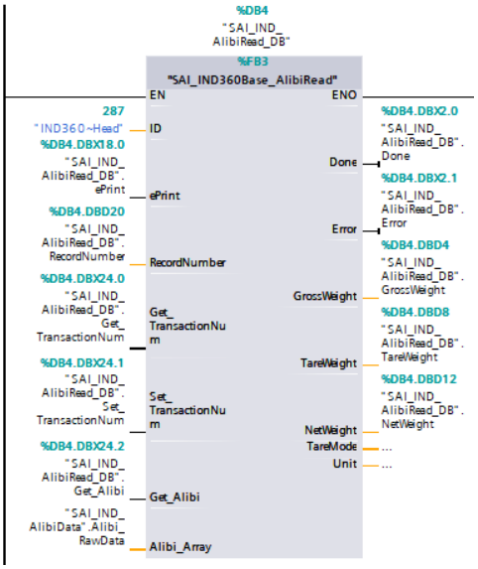


Figure 4-14: SAI_IND360Base_alibiRead Function Block

Table 4-12: 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) an IND360 2P 2 Block Format.

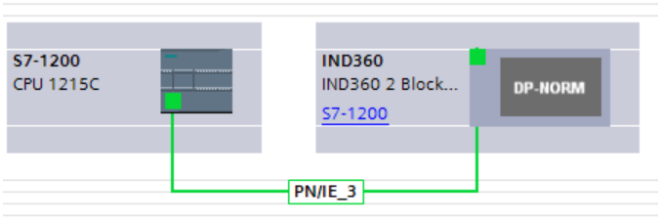


Figure 5-1: Add a Profinet device in the Network view

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

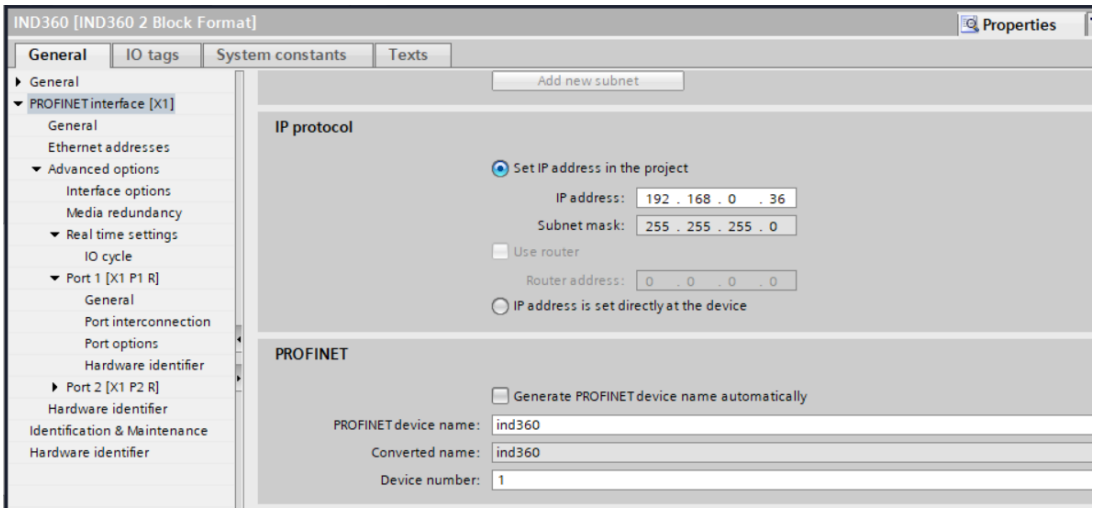


Figure 5-2: Profinet device name and IP address

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.

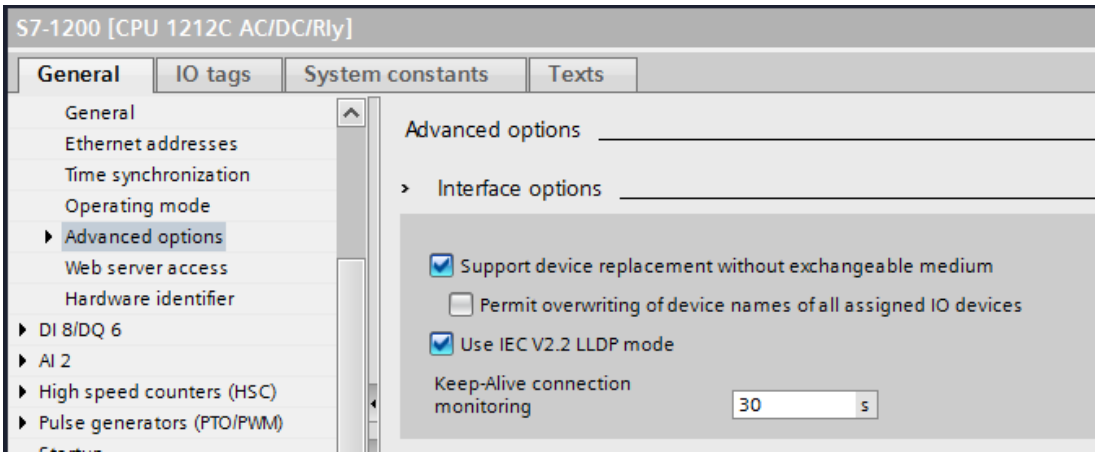


Figure 5-5: the LLDP feature

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).

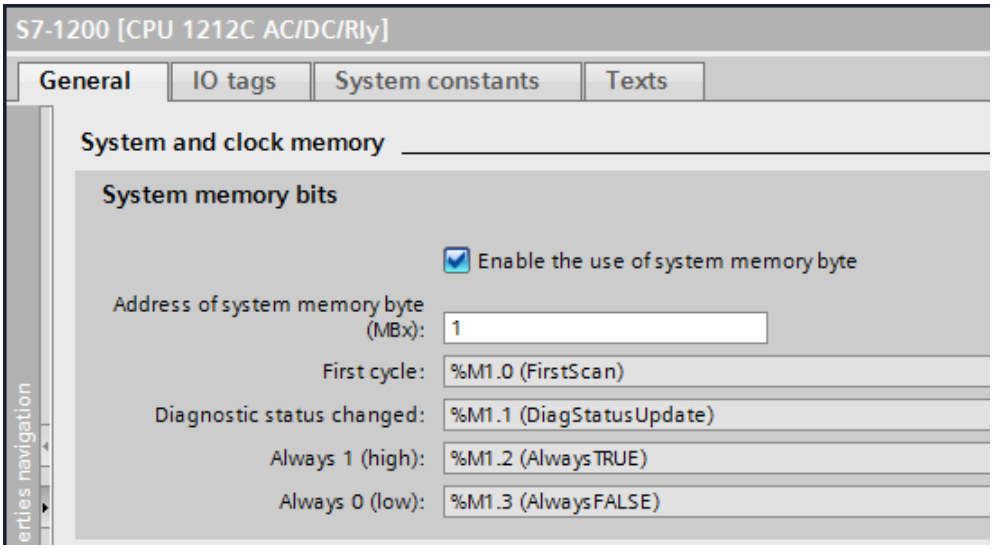


Figure 5-6: Enable system memory byte

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_INDPCELL_WeightProc(FB), SAI_INDPCELL_WeightProc_DB

   d) SAI_Cyclic_CheckAlive(FB), SAI_Cyclic_CheckAlive_DB

   e) SAI_Buffer(DB600)， **do not modify this Data Block's number as other Function Blocks are referring directly to its DB number.**

   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.

   f) SAI_IND_CalFree(FB), SAI_IND_CalFree_DB

   a) SAI_IND_ZeroAdjust(FB), SAI_IND_ZeroAdjust_DB

   b) SAI_IND_SpanAdjust(FB), SAI_IND_SpanAdjust_DB

   c) SAI_IND_WriteAdjustSettings(FB), SAI_IND_WriteAdjustSettings_DB

   d) SAI_IND_ReadAdjustSettings(FB), SAI_IND_ReadAdjustSettings_DB

   The other function blocks can be added into the programming if required.

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

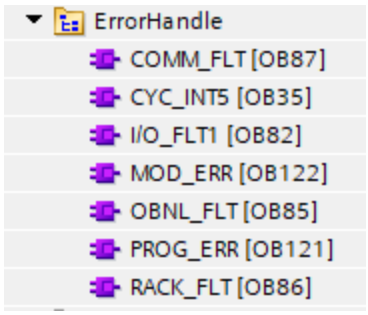


Figure 5-7: Error handle programs of S7-300

2) Delete the other unused program blocks in MT_ACT_Application.

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

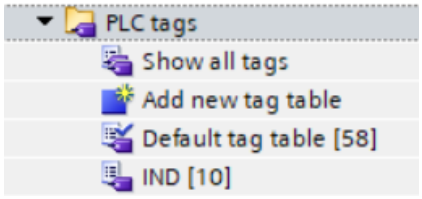


Figure 5-8: Duplicate the PLC tags

4) Duplicate all the PLC data types.

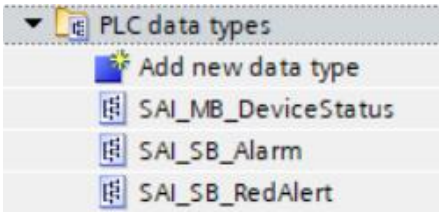


Figure 5-9: Duplicate the PLC data types

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

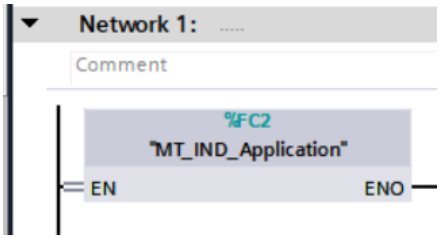


Figure 5-10: Call up "MT_IND_Application" in the Main OB

6. Add New IND360 POWERCELL

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

1) In Devices and networks -> Network View, add another IND360 2 Block Structure.

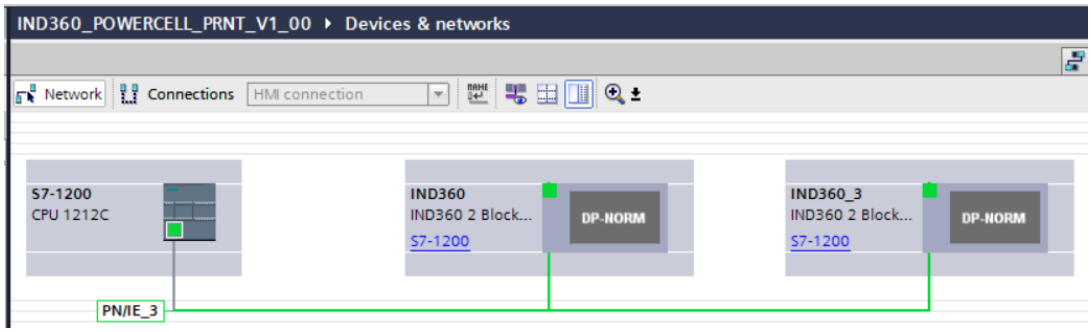


Figure 6-1: Add another IND360 into the network

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

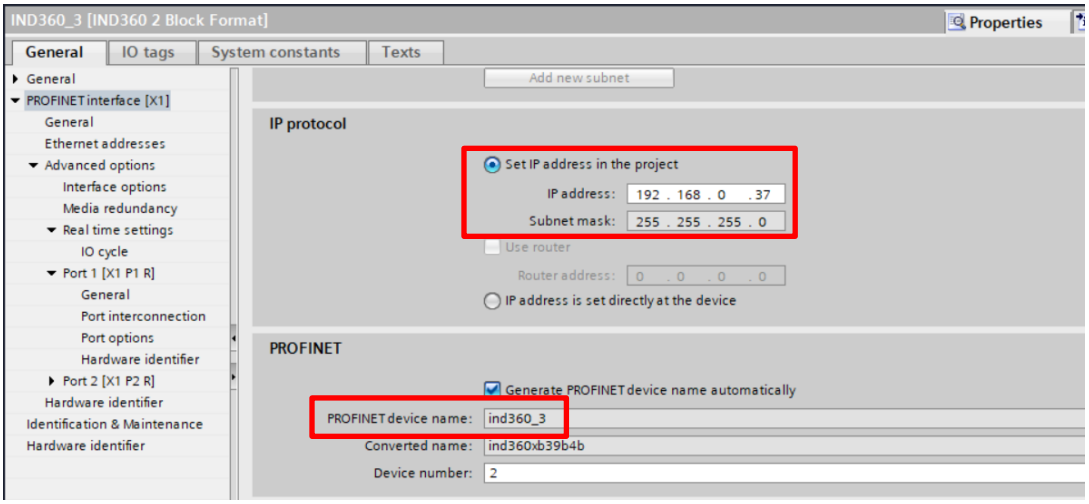


Figure 6-2: PROFINET device name and IP address

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.

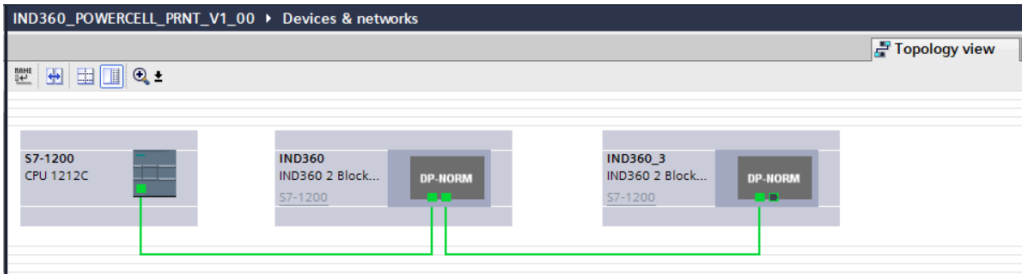


Figure 6-3: Connecting multiple IND360 in PROFINET network

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 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		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	88...89		MB Device Status
MB Response	0	1 7	90...91		MB Response
Status Block_1	0	2			Status Block
Parameter Access Point	0	2 1			Parameter Access Point
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

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_Cyclic_CheckAlive function blocks but both FBs are assigned with different DBs which are SAI_Cyclic_CheckAlive_DB (DB1) and SAI_WeightProc_DB2 (DB3)

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.

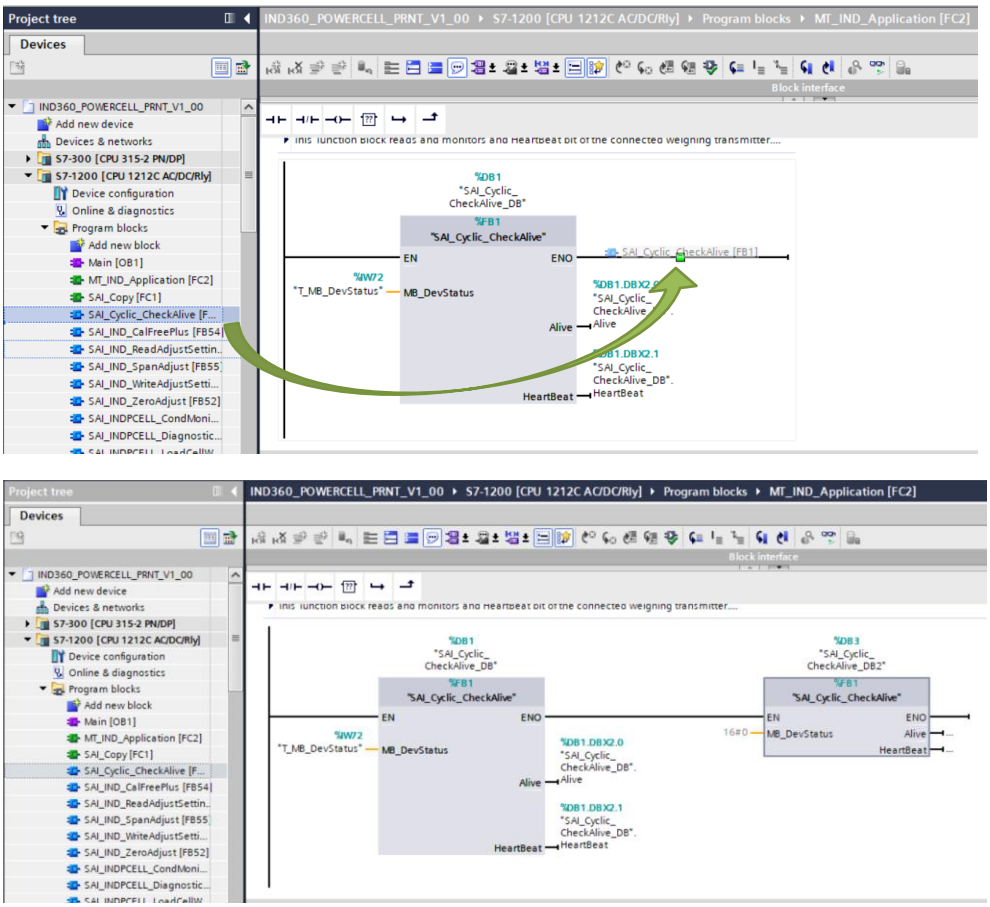


Figure 6-5: Two function blocks of the same type, but different data blocks

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

7) Download the project into the PLC.

7. Frequently Asked Questions

1. Q: I have duplicated the SAI_INDPCELL_WeightProc function block and SAI_INDPCELL_WeightProc_DB data block into another project, but I was not able to read the weight data.

   A: Make sure the device I and Q addresses are assigned accordingly between the Device overview and the function block assignment. If it is an S7-300 PLC, there is a need to edit the default cyclic data range (128 byte) to cover the device I and Q address range. In this sample code, the PLC's cyclic data range has been configured to 512 bytes.

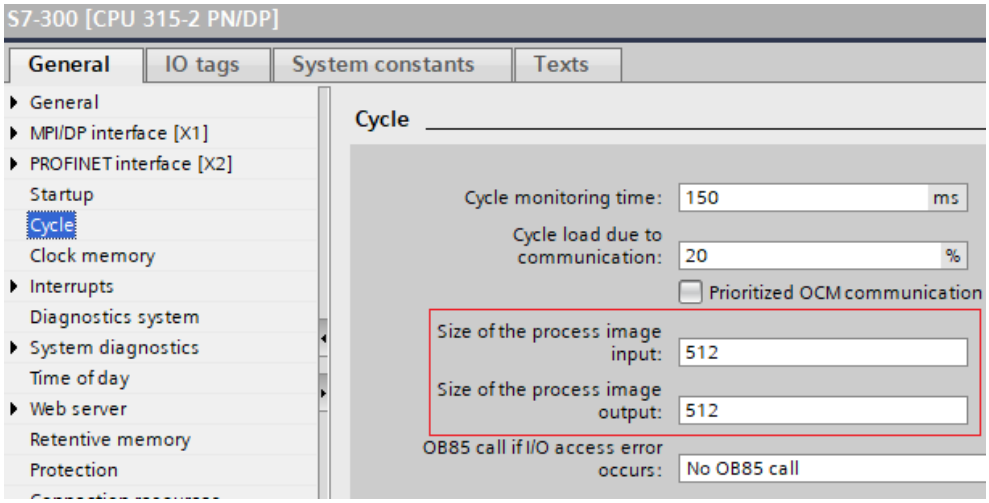


Figure 7-1: Edit the S7-300 PLC cyclic data range

2. Q: How to read the gross, tare or net weight?

   A: The PLC command to read gross weight is decimal "0" or "1", decimal "2" to read tare weight and decimal "3" to read net weight. Insert one of these decimal command values into the "WeightCmd" input parameter of SAI_INDPCELL_WeightProc function block, after a tare or zero command the function block will then return the required weight data accordingly.

3. Q: How to know when the scale is overloaded or underloaded?

   A: Refer to the Overload and Underload bits of SAI_INDPCELL_DiagnosticStatus function block.

4. Q: After I managed to integrate the IND360 to the PLC, why is the IND360's IP address showing 0.0.0.0?

   A: This is due to the PROFINET protocol. By choosing "Set IP address in the project", while booting up the PLC will assign the IP address to the Profinet device according to the Device name. Hence with this option, the IND360 will not display its assigned IP address. If the second option "IP address is set directly at the device" is chosen, the PLC will not assign any IP address to the device. With this option, the IND360 will display its own IP address (see below).

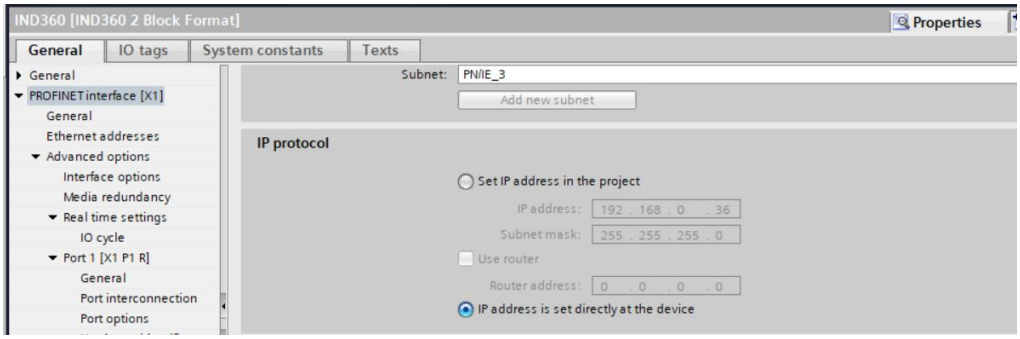


Figure 7-2: IND360 POWERCELL will display its own IP address with this setting

5. Q: The PLC has activated its LLDP function, but the newly connected IND360 POWERCELL cannot communicate automatically.

   A: Under Online access, expand the active Ethernet interface, click on Update accessible devices. Look for the newly connected IND360 POWERCELL's MAC address, check whether it says "Accessible device [MAC address can be found on the device label]" as shown in Image 7-3. If the new device has been assigned with Device name and IP address previously, click on Online & diagnostics, then reset the device to factory settings.

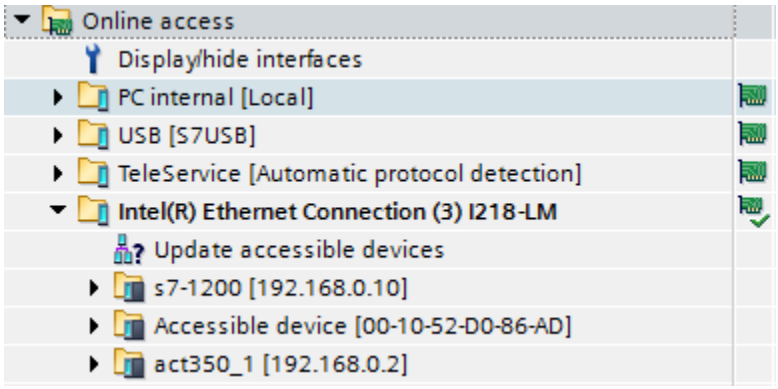


Figure 7-3: new device appears as Accessible device

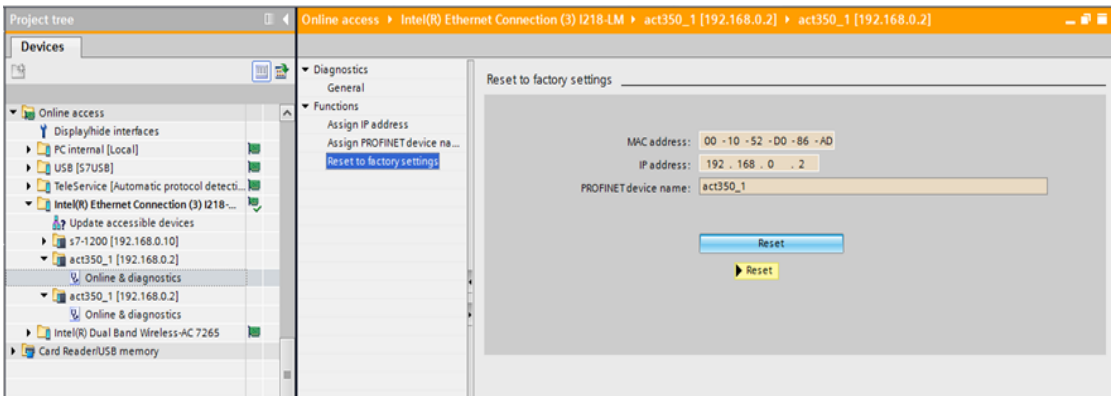


Figure 7-4: reset the IND360 POWERCELL to factory settings, no Device name and IP address

6. Q: With a network switch, is the LLDP function still available?

A: The network switch has to support LLDP function. After imported the device's GSDML file, configure the Ethernet connection in Devices & networks -> Topology view.