PLC software manual

X-Flow90 Device

Flowsort B.V.

Version 1.0

Chisinau,

Moldova

20/03/2024

Bulgara 33/1,
2001MD Chisinau,
Republic of Moldova

Tel. +373 22 996 170

info@isd-soft.com
www.isd-soft.com

Table of contents

1. Changelog

REVISION HISTORY			
DATE	VERSION	DESCRIPTION	AUTHOR
20.03.2024	1.0	Initial version	Adrian Lulascu
10.04.2024	1.1	Terminology update	Adrian Lulascu
09.07.2024	1.2	Added program integration description for Allen-Bradley, OMRON, MITSUBISHI, BECKHOFF	Adrian Lulascu

2. Introduction

In this document will be described the PLC software program for 90° transfer machine called X-Flow90.

This device is intended to transfer packages from one conveyor lane to another under a 90° angle.

It consists of one Pulseroller PGD motor for lifting the mechanism by excentric shafts, there are two inductive sensors for detecting the state of the lifting mechanism: the mechanism is in top or bottom position. For diverting it is used a Pulseroller Senergy-Ai motor driven roller that drives the transport belts. All mentioned components are connected to a ConveyLinx-Ai2 module or optionally a MotionLinx-Ai module.

2.1 Introduction

Flowsort

Flowsort is an experienced company specializing in intralogistics automation technology, offering high-quality solutions with a focus on modular standard systems and open interfaces.

Flowsort plug and play high-speed sorting diverters are best in the market and can be implemented at low costs. Flowsort diverters are suitable for a wide range of sorting applications, easily incorporated into any conveyor system.

ISD

ISD is an innovative software outsourcing provider with various skills, focusing primarily on software development and maintenance. For over 15 years, ISD has been delivering enterprise level software solutions in various verticals.

3. Hardware

In this chapter will be described the hardware part used in the transfer. The module consists of:

- ConveyLinx-Ai2 controller
- Two inductive sensors for detecting the position of the mechanism.
- A Pulseroller PGD motor for lifting the mechanism up:

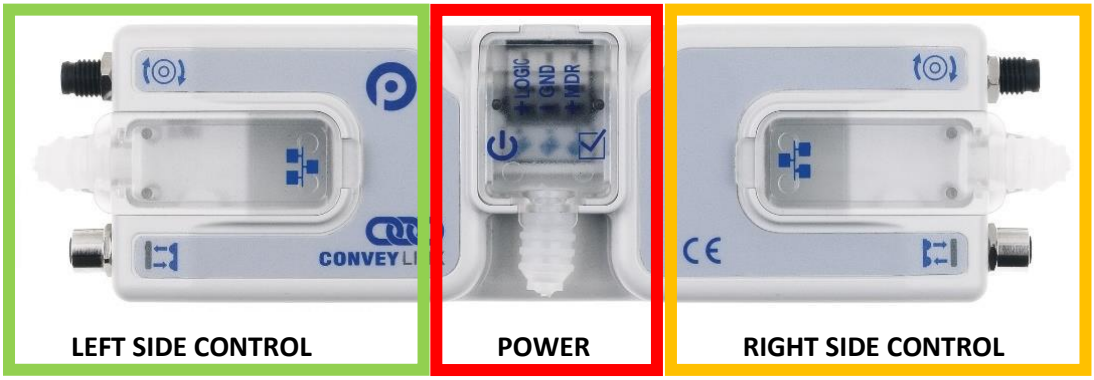
  https://www.pulseroller.com/products/geared-drives/geared-drives-ai/pulse-geared-drive-ai-24v-48v
- A Pulseroller Senergy-Ai motor driven roller that drives the transport belts:

  https://www.pulseroller.com/products/motor-rollers/motor-rollers-ai/senergy-ai-24v

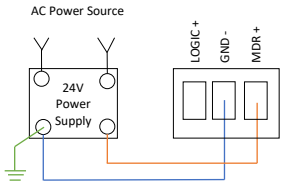
3.1 Controller connection

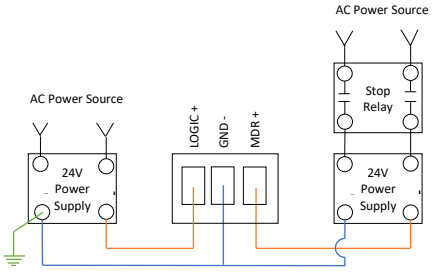
The X-Flow90 diverter comes standard with ConveyLinx-Ai2 controllers. The connection of the hardware is the following: the motors are connected at the upper connector and sensors at the bottom connectors.

- Left side: PGD motor and sensor top position.
- Right side: Pulseroller Senergy-Ai motor and sensor bottom position.



Power connection diagram:





General description of the module:

MDR (motor) Connection	M8 4 pin Female
Sensor Connection	M8 4 pin Male
Power Terminals	24VDC Power Terminals with separate connections for Logic and Motors
Power requirement	18V-30V – 120 mA Logic – 6A MDR (with 2 MDR connected)
Power conductors	0,2 – 2,5mm² (28 - 12AWG)
Network Link	Link Left and Link Right – RJ-45 style Ethernet network connection between modules including LED Indicators

For the full set of variable and possibilities you can read the manuals of the specific controller at https://www.pulseroller.com/downloads/.

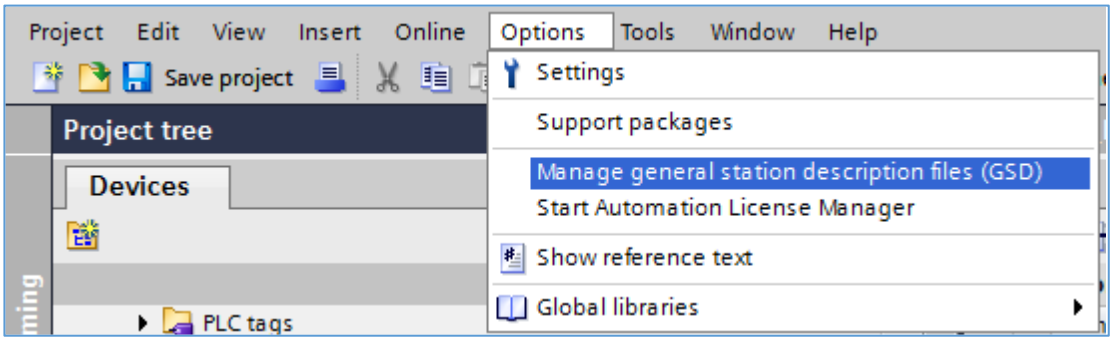
More information can be found on ConveyLinx-Ai Family Complete Guide: https://www.pulseroller.com/files/EU/Control%20Literature%20&%20Drawings/ConveyLinx%20Ai/Users%20Manual%20and%20Specifications/ConveyLinx-Ai%20Family%20Complete%20Guide.pdf

3.2 Hardware configuration integration

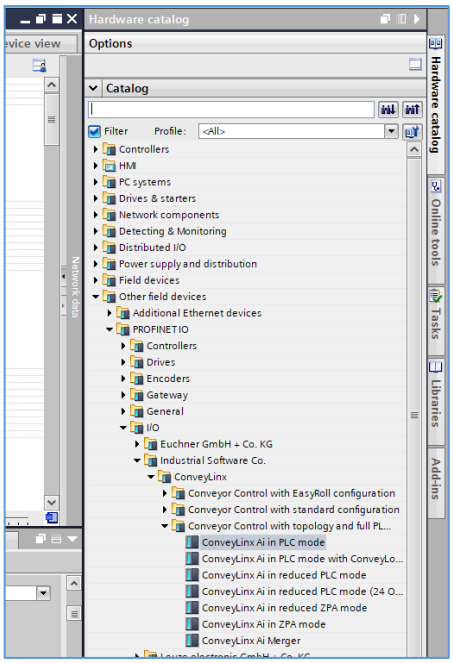
To integrate the X-Flow90 Diverter in a Siemens 1200 or 1500 Controller, first the GSDML file of the controller must be downloaded. This can be done on the Pulseroller's website: https://www.pulseroller.com/downloads/.

Go to Firmware & Software downloads -> PLC-Files -> ConveyLinx-Ai -> Profinet GSDML -> download the file called: Profinet GSDML files.

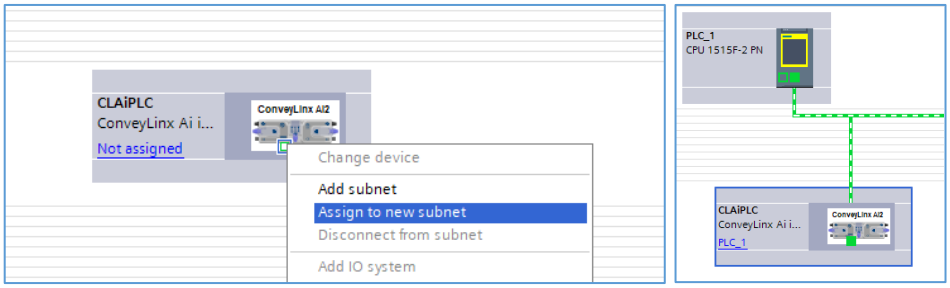
First unzip the GSDML file, then import the GSDM file into the project. **Note**: make sure only one TIA instance is opened. This can be done in the options menu of TIA portal under Manage General station description files:



In the following popup select the path of where the GSDML file from the diverter is downloaded and install the GSDML file. When the installation was done correctly, the device is now available in Devices & Networks -> Hardware Catalog -> Other field devices.

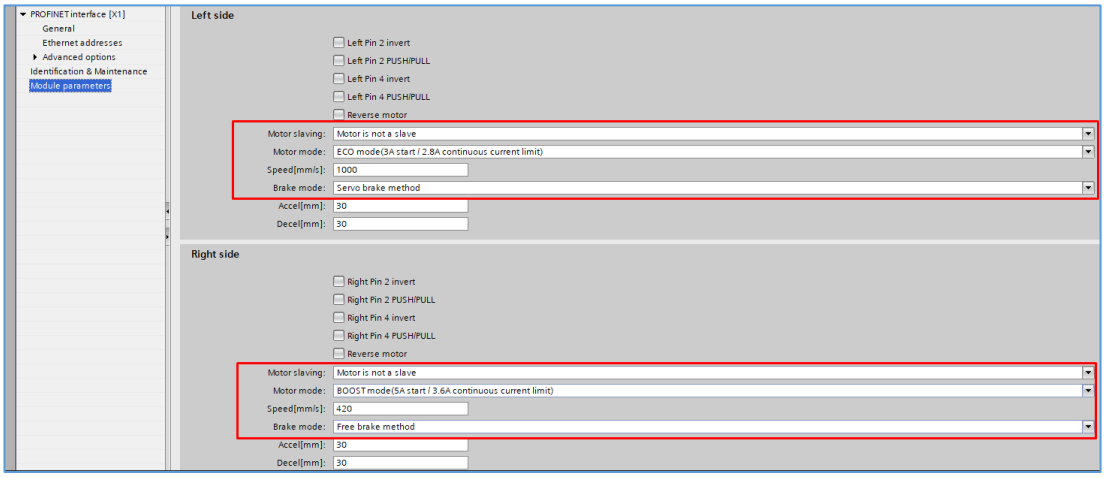


Drag and drop the ConveyLinx-Ai2 in PLC mode into the network. Assign the ConveyLinx-Ai2 object to the controller.



Now the system integrator can give the module an ProfiNET name and IP-address (the device should be in the same sub-network).

The following parameters should be set in the properties:

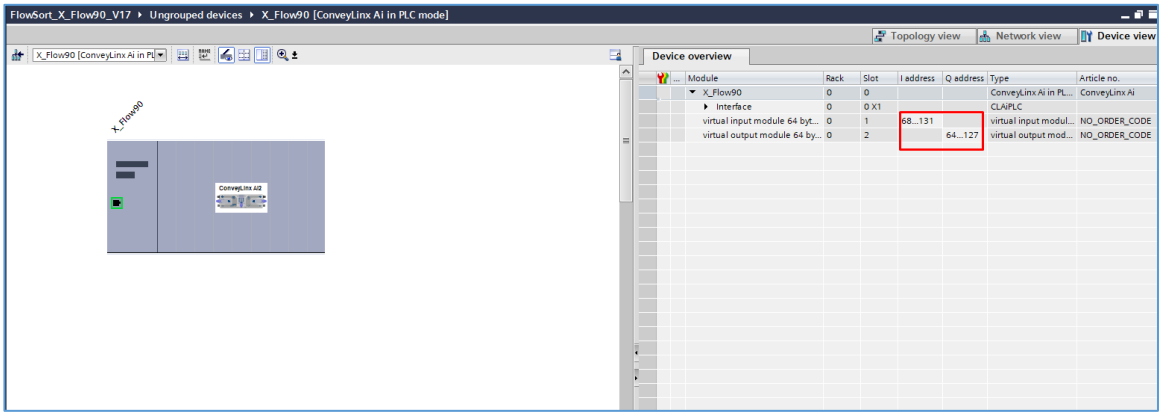


The speed and acceleration can be adapted on the inputs at the function block. These settings are the settings that the controller uses after a communication fault.

After these settings are set the configuration can be downloaded to the controller.

3.2.1 Integration of the addresses

After assigning the ConveyLinx-Ai2 to the network, the PLC will assign the next free addresses:



These addresses will be necessary for creating tags, that will be explained in chapter 4.

4. Software

In this chapter the software part of the X-Flow90 diverter will be explained.

4.1 Integration

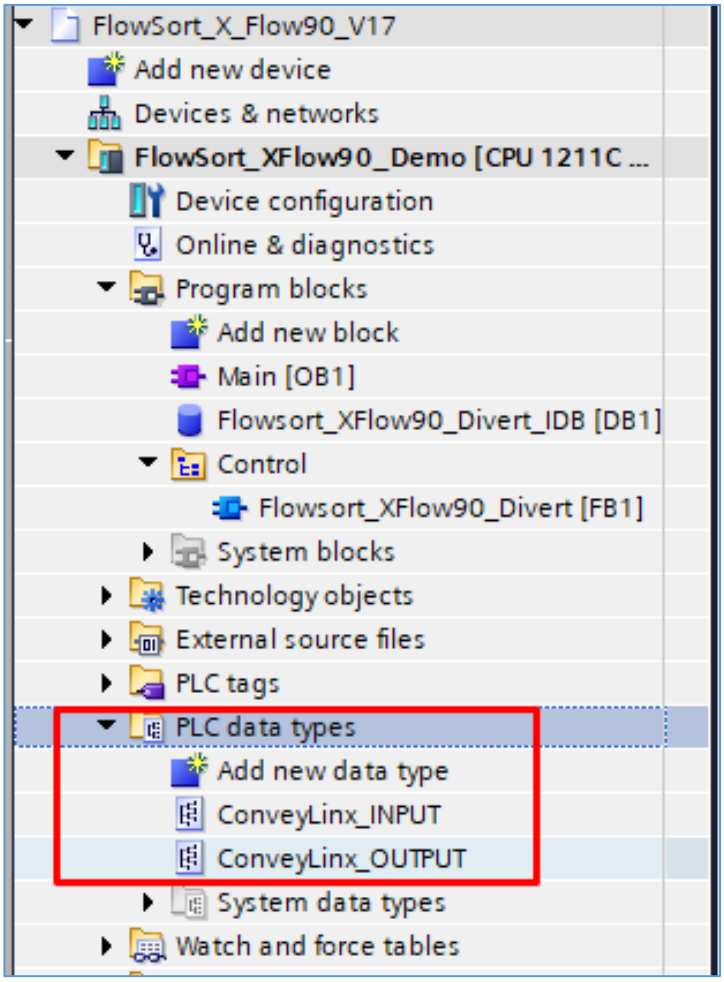
For the integration the following parts should be copied to the integrator project:

- PLC Data types
- PLC tags
- Flowsort_XFlow_Divert Function Block

Integration of each will be explained next:

PLC Data types:

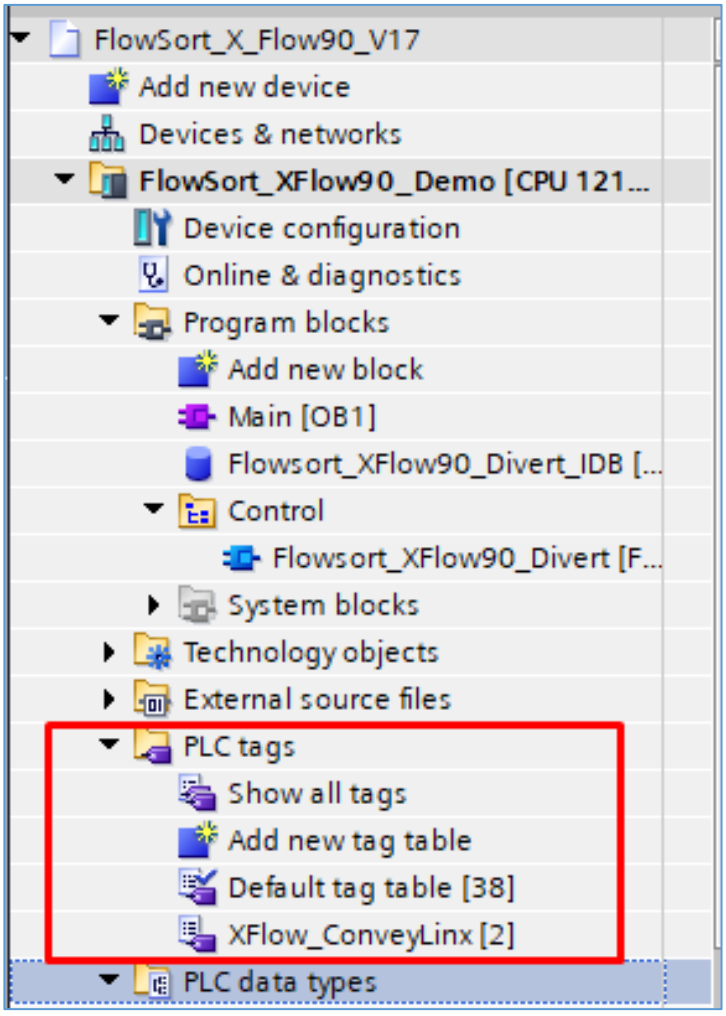
From the project tree -> PLC data types: CL_INPUT and CL_OUTPUT should be copied to the same folder on the integrators project.



PLC tags:

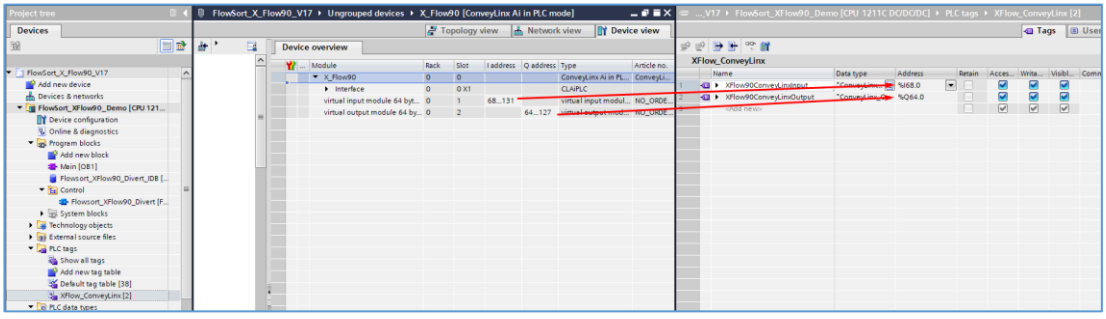
At this point, the addresses from Chapter 3.2.1 will be used:

Open project tree -> PLC tags -> XFlow_ConveyLinx should be copied to the same folder in the integrator's project: (Note: keep the data type as CL_INPUT and CL_OUTPUT)



Adjustment of the addresses:

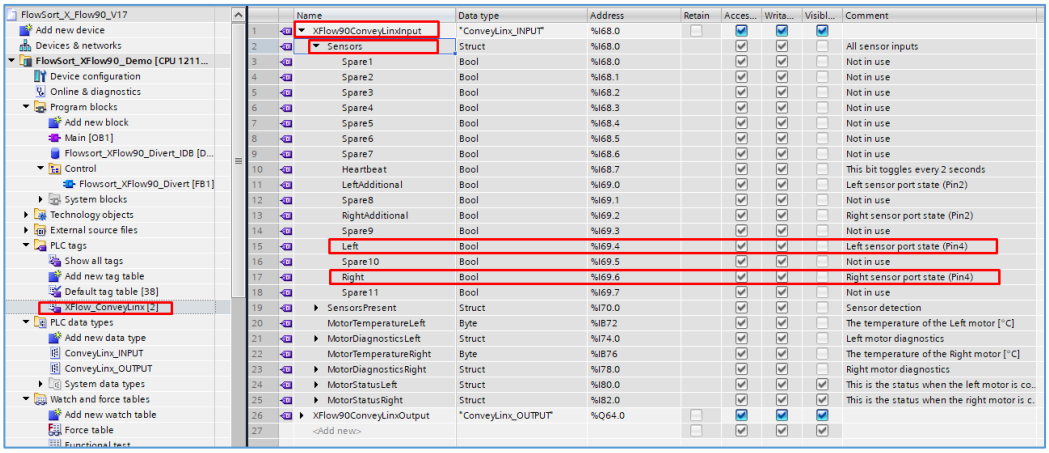
The address from the PLC tags should be the same as in Chapter 3.2.1; from the Devices & networks -> Device overview -> virtual input module -> I address and virtual output module -> Q address.



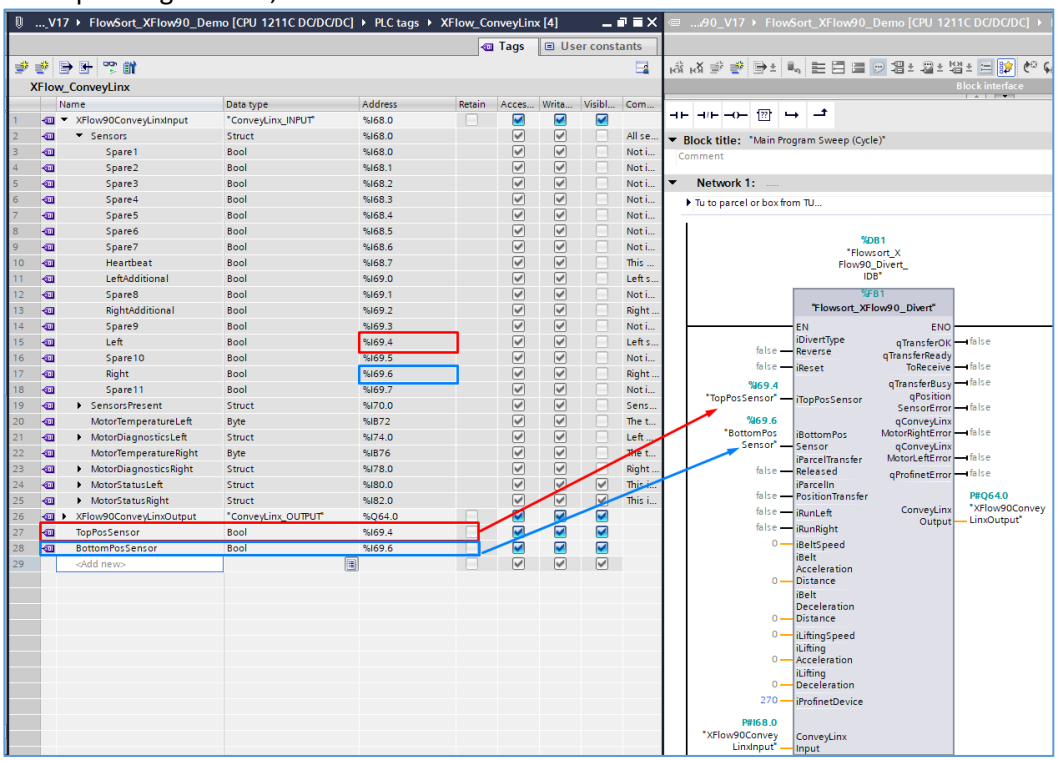
Note: the first bit should be used X.0.

Creating sensor tags:

The sensors addresses are extracted from the: PLC tags -> XFlow_ConveyLinx -> "ConveyLinx_INPUT" -> Sensors. In the list of the sensors find Left and Right:

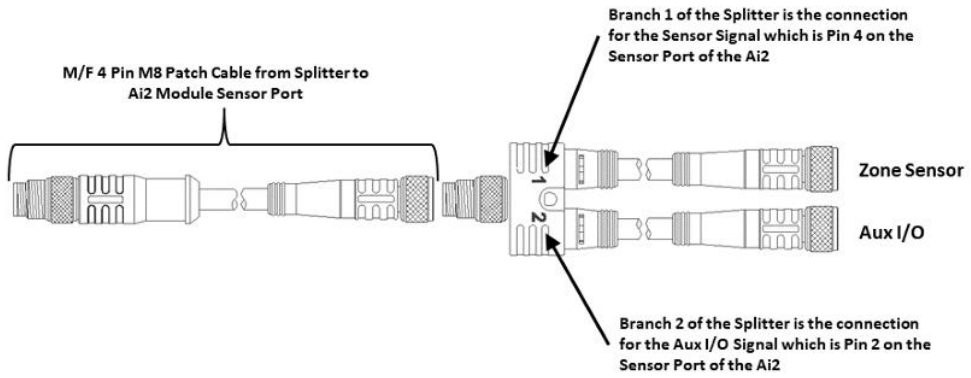


The left sensor is the top positioning sensor, the right sensor is the bottom positioning sensor. It is necessary, for convenience to create new tags for the sensors, use the addresses from the corresponding sensors, in this case:



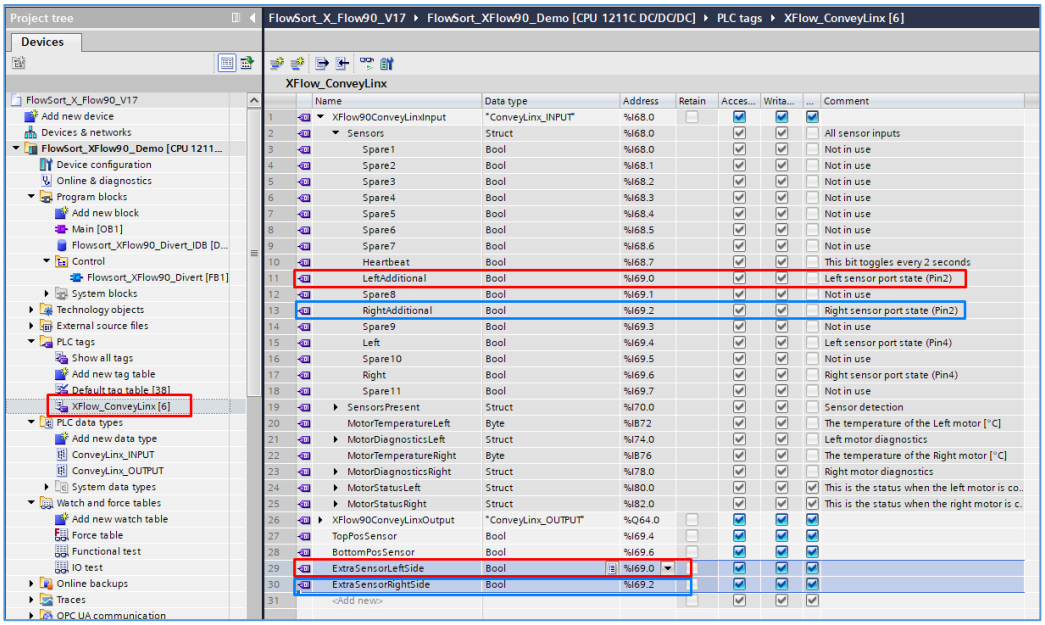
Drag and drop to the corresponding inputs to the function block.

It is possible to add one extra sensor to each side of the ConveyLinx-Ai2 module, in this case it will be necessary to use a splitter:



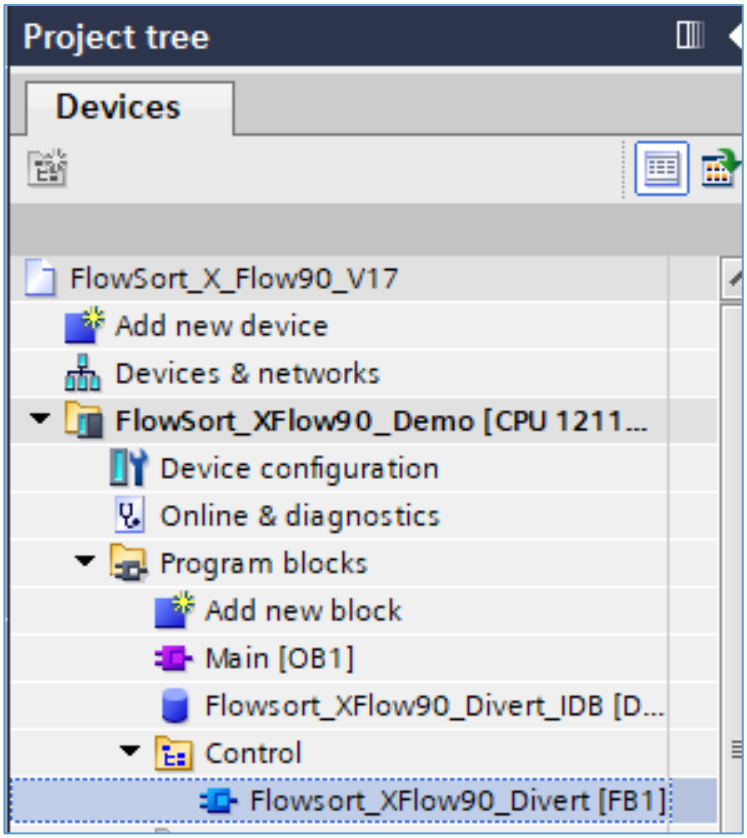
Typical Parallel Splitter Cable Usage

Connect physically the existing sensor to splitter port 1 and the extra sensor to splitter port 2. It will be necessary to create extra tags for each sensor:



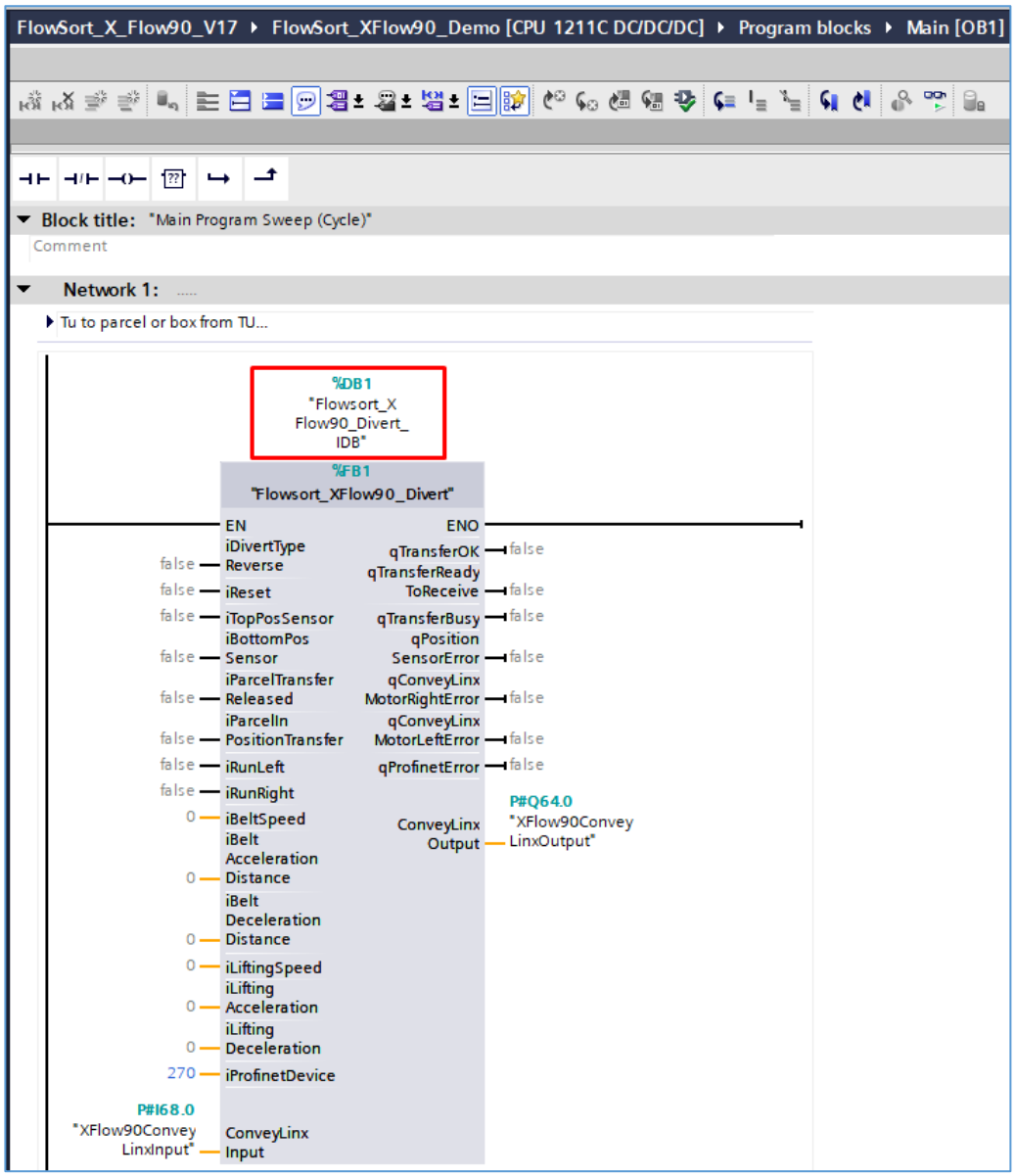
Flowsort_XFlow90_Divert Function Block:

From the project tree -> Program blocks -> Control -> Flowsort_XFlow_Divert.



The number of the FB (currently is 1) can be adjusted, if it will be occupied, TIA will assign the next free number.

The IDB of the FB will be generated on the integrator's project when the FB will be added to the FC or to OB1:

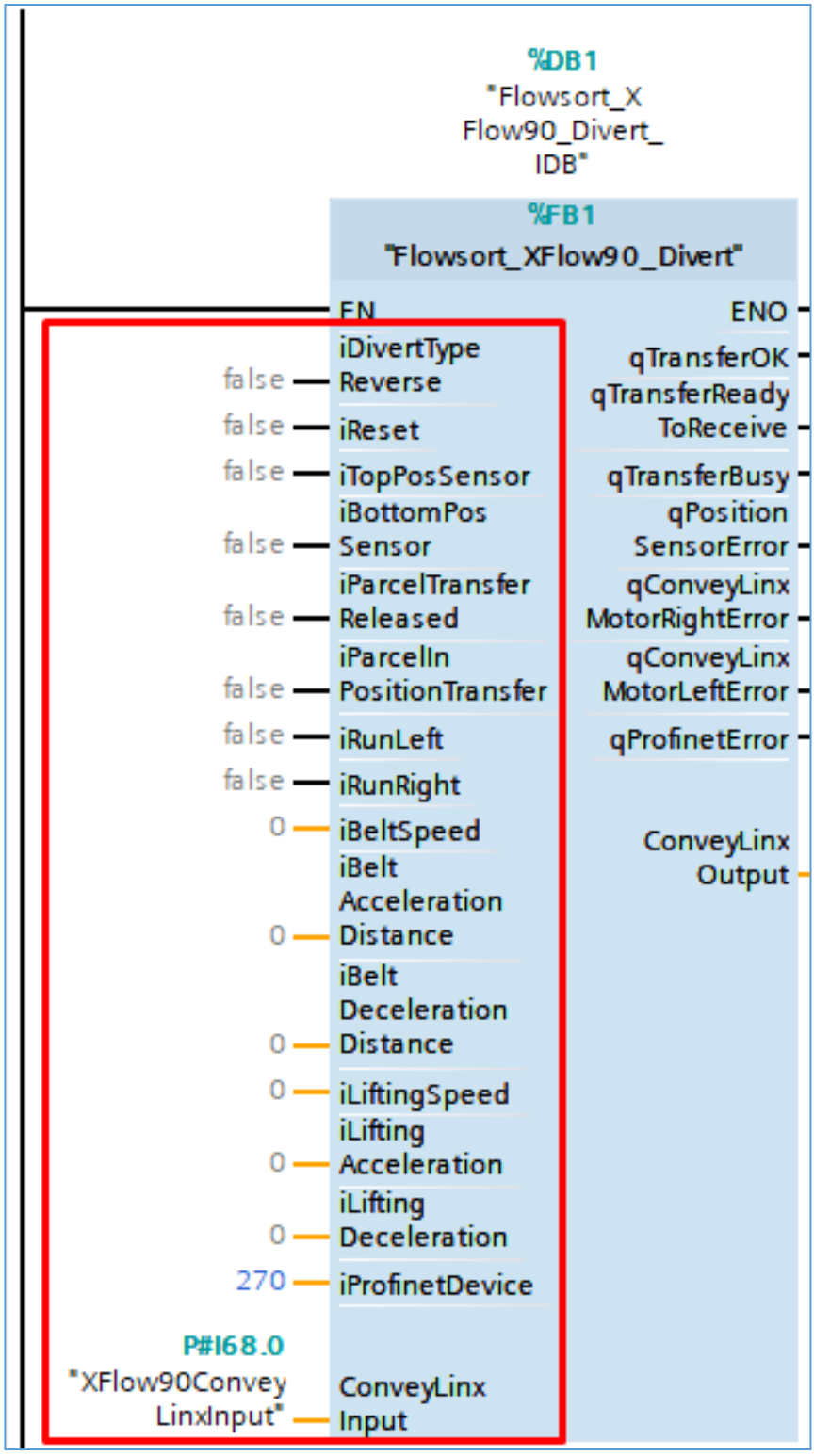


The name of the IDB will be generated by TIA, if necessary, it can be changed according to the standard of the integrator.

If all the steps above are complete, next all inputs and outputs of the FB will be explained.

4.2 Function Block Inputs

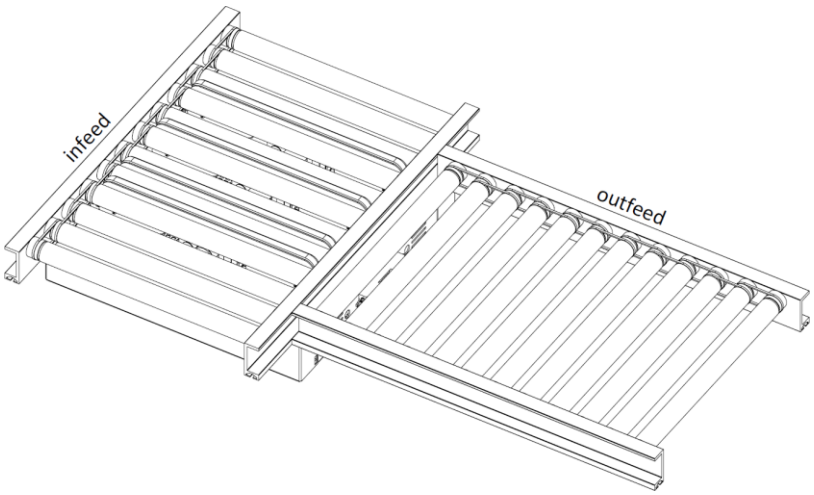
In this chapter will be explained all the inputs, where is it coming from and how to use it.



iDivertTypeReverse:

There are two methods to divert the transport unit:

- Straight: the package is approaching from the infeed when it reaches the transfer position it will be lifted and transferred to the outfeed. In this case the input should be set to FALSE.
- Reverse: the package is approaching from the outfeed, to the infeed, the transfer moves up and is ready to accept the package, after it is moved onto the transfer position it goes down and moved to the infeed. In this case the input should be set to TRUE.



Note: the type can be hot changed, the following conditions should be met for changing it during operation:

- X-Flow90 should be ok
- X-Flow90 should not be busy
- X-Flow90 should not be in error
- X-Flow90 should be in initial position

This input will be used more in this document to explain certain logic based on the used divert type.

iReset:

The reset is used to reset or acknowledge certain errors that can be encountered during operation. This signal should come from the integrator and best practice is to connect the reset button located in the field.

iTopPosSensor and iBottomPosSensor:

The addresses created for the positioning sensors (explained in chapter 4.1 PLC Tags) should be added to these inputs.

iTuTransferReleased:

This input is used only in Straight divert type (iDivertTypeReverse is FALSE). It indicates that a new package was released onto the transfer position, as a result the qTransferReadyToReceive output of the function will be set to false, as it already busy.

The signal should come from the integrator, it should be activated only when the package was released onto the transfer position. The signal active length should be at least 1 second and can be reset up to when the next input will be activated: iTuInPositionTransfer. If the signal will not change it state when the next package will be released, the signal will be ignored.

iTuInPositionTransfer:

The signal should come from the integrator. The duration is at least 1 second and up to finishing the divert process.

When using Reverse divert type (iDivertTypeReverse is TRUE) this signal should be set to TRUE when the package is waiting on the outfeed to be transported onto the transfer position. The signal will activate the lifting motor, the transfer will be set to busy: qTransferBusy = TRUE and will be ready to receive: qTransferReadyToReceive = TRUE. This will be the indicators that the package can be released onto the transfer position.

When using straight divert type (iDevertTypeReverse is FALSE) this input is used to indicate that the package is in position for lifting. Activate the signal only when the package is in position to be lifted.

iRunLeft/Right:
The signal should come from the integrator and will activate the movement of the transfer belt to the left or right. The signal should be activated after the iTuInPositionTransfer and should be deactivated when the package left the transfer position in case of straight divert type and is on transfer position completely in case of reverse transfer type.

iBeltSpeed:

It is the input for setting the speed of the transfer belts running left or right. The value is in m/s and the range that can be used is minimum 0,05 m/s, max speed: 0,42 m/s.

iAcceleration Distance and iDeceleration distance:

It is the acceleration and deceleration distance of the transfer belts running left and right. The value is in mm and according to the motor specification it can be:

- Acceleration: from 30 to 10000
- Deceleration: from 0 to 10000.

iLiftingSpeed:

It is the input for setting the transfer lifting speed. The value is in RPM and the range that can be used is minimum 8 RPM maximum 86 RPM.

iLiftingAcceleration and iLiftingDeceleration:

It is the acceleration and deceleration of the transfer lifting. The value is in pulses and according to the motor specification it can be:

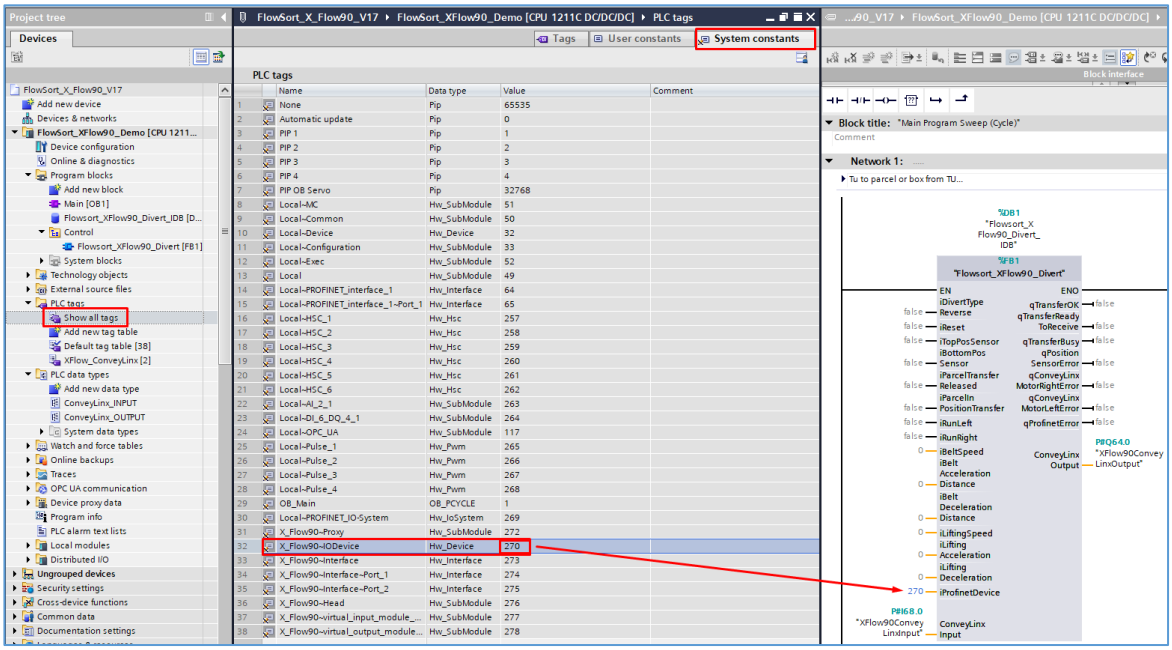
- Acceleration: from 30 to 10000
- Deceleration: from 0 to 10000.

iProfinetdevice:

It is the hardware device number of the ConveyLinx-Ai2 module. It is used to detect the profinet connection error.

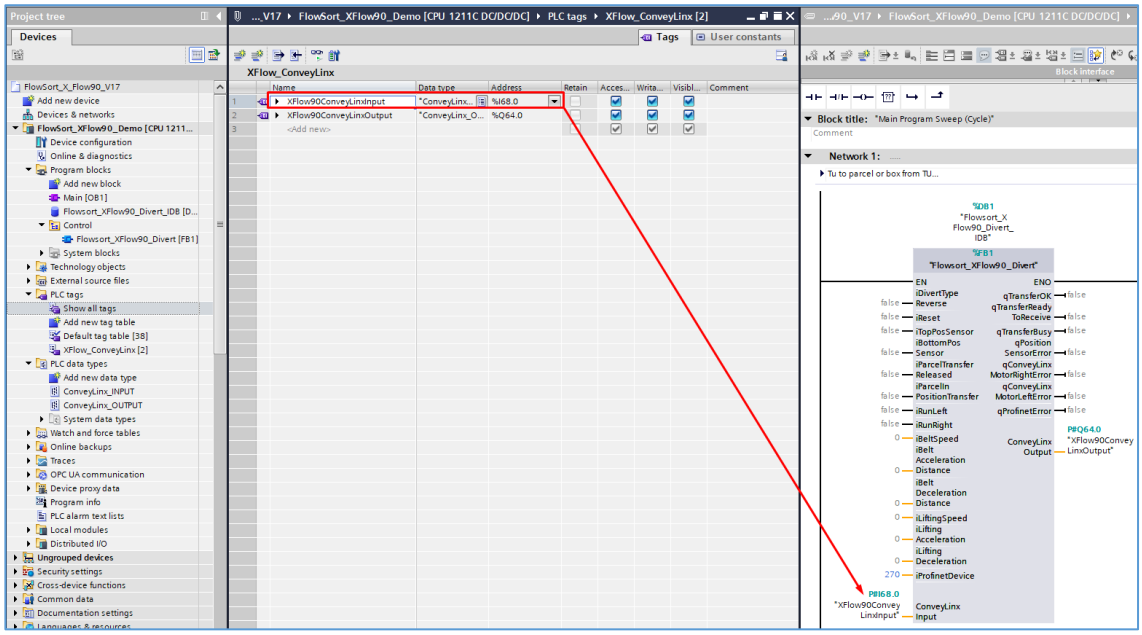
The number can be found in the System constants: Open project tree -> PLC tags -> Show all tags -> System constants.

In the Tag list check for ConveyLinx-Ai2 name (assigned in device & networks) and Data type: Hw_Device.



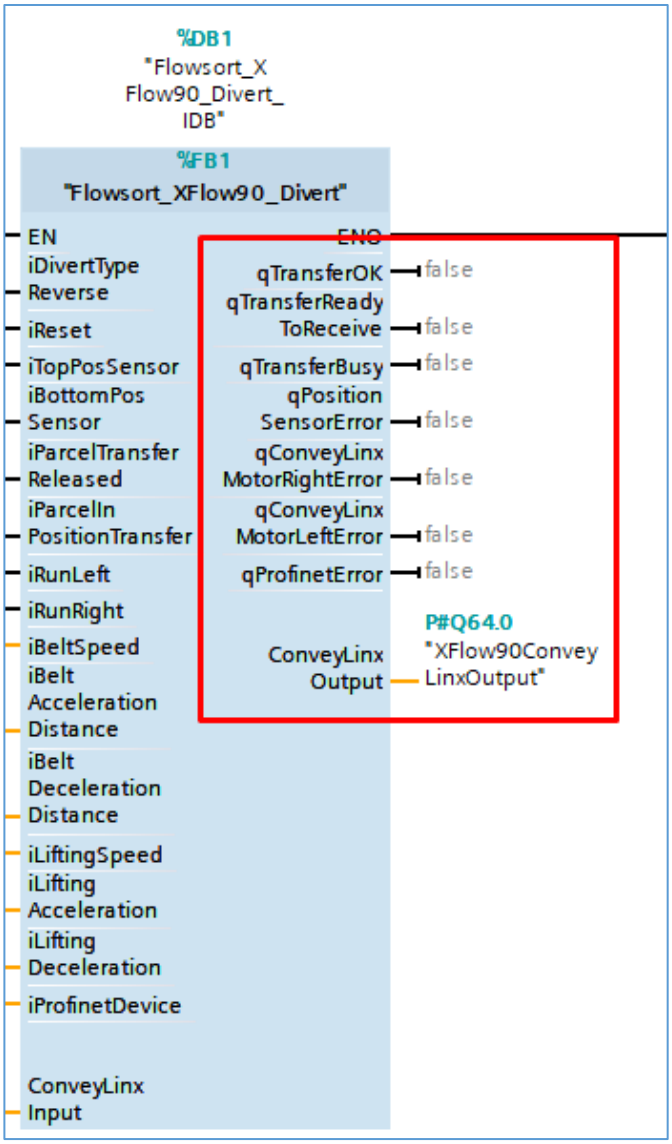
CLInputs:

The inputs from the ConveyLinx-Ai2. It can be found in the tags created in chapter 4.1 PLC Tags.



Drag and drop the tags to the output of the function block.

4.3 Function Block outputs



qTransferOK:

The output indicates that the transfer position has no errors. The signal should be used by the integrator for releasing a new package onto the transfer position. If this signal is false, the transfer is not ready to accept any units.

qTransferReadyToReceive:

Indicates that the transfer is ready to accept a new package. It should be used by the integrator together with qTransferOK to release a new package.

qTransferBusy:

Indicates that there is a package moving and the transfer process is not finished. Can be also used by the integrator as an indicator that there is already a package on the divert position.

Sensor error:

Indicates a hardware error on the sensor. The sensor was not triggered in time during initialization, moving up or moving down.

Possible causes faulty sensor, faulty sensor adjustment, the lift motor did not work.

If this error is TRUE, a visual inspection should be performed.

qCLMotorRightError:

Indicates a hardware issue on the motor connected to the right side of the ConveyLinx-Ai2, in this case the motor moving left/right the belts. The following reasons can trigger the error:

- Overheat
- Short circuit
- Overload
- Stalled
- Over voltage
- Low voltage

If this error is high a visual inspection should be performed.

qCLMotorLeftError:

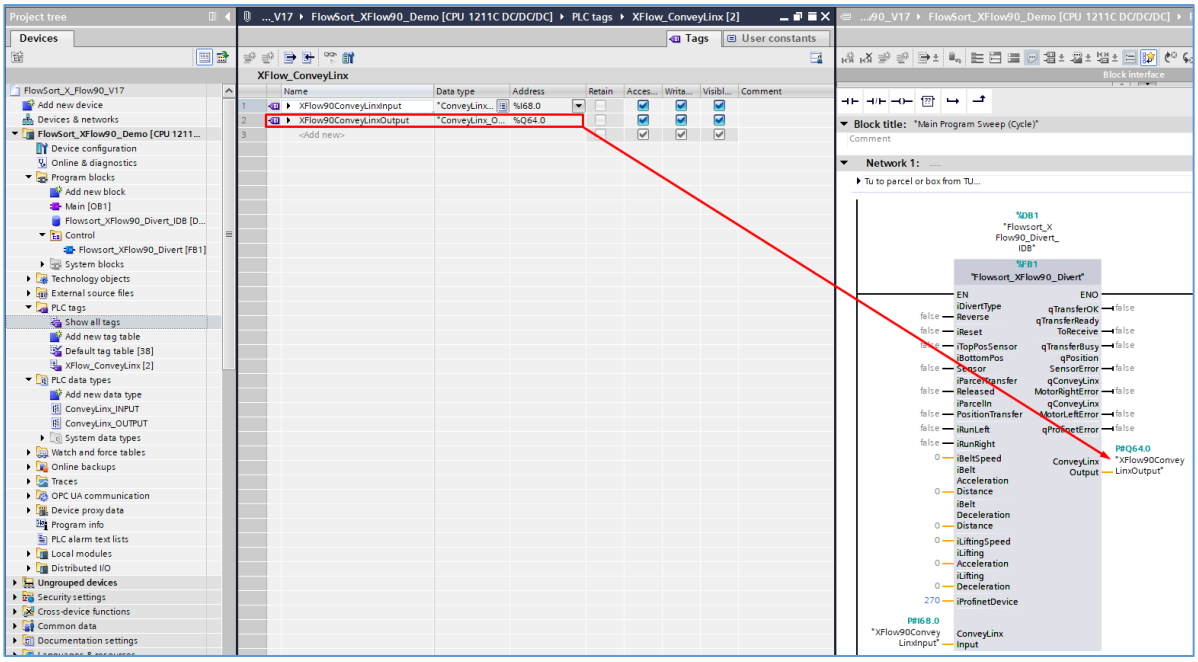
Indicates a hardware issue on the motor connected to the left side of the ConveyLinx-Ai2, in this case the motor of the lifting mechanism. The following reasons can trigger the error:

- Overheat
- Short circuit
- Overload
- Stalled
- Over voltage
- Low voltage

If this error is high a visual inspection should be performed.

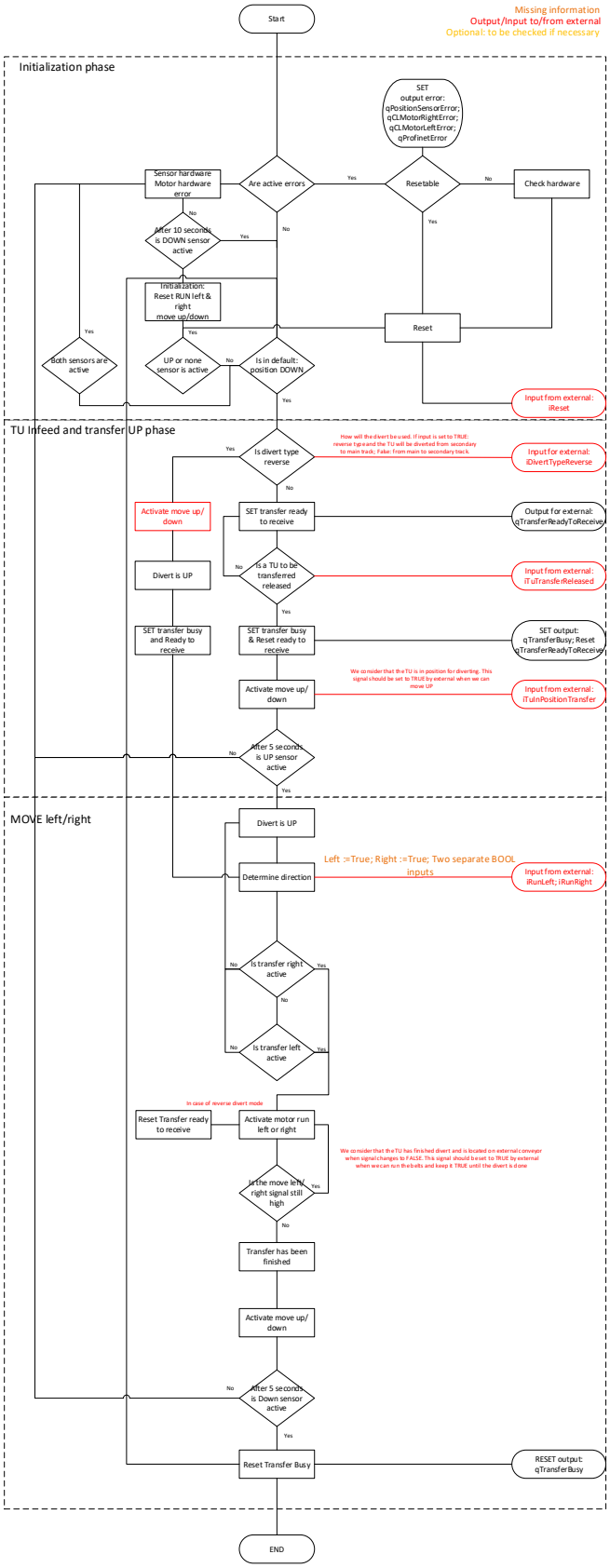
CLOutput:

Same as CLInput, the outputs to control the ConveyLinx-Ai2, It can be found in the tags created in chapter 4.1 PLC Tags.



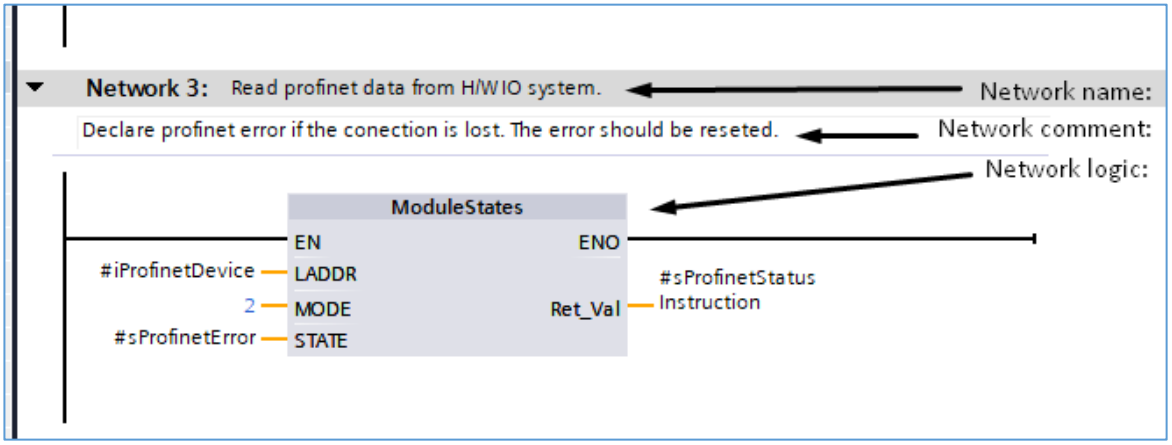
Drag and drop the tags to the output of the function block.

4.4 Flow chart

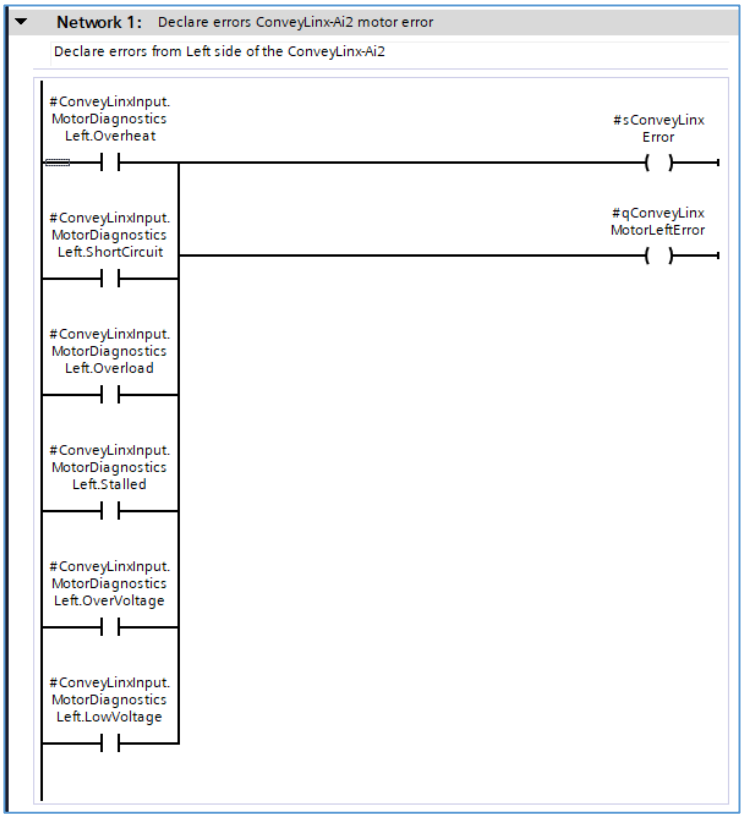


4.5 Function Block: explanation

In this chapter the logic inside the function block Flowsort_XFlow90_Divert will be explained. The function block consists of 16 networks, each of them has a name and a comment, where the logic is shortly explained:

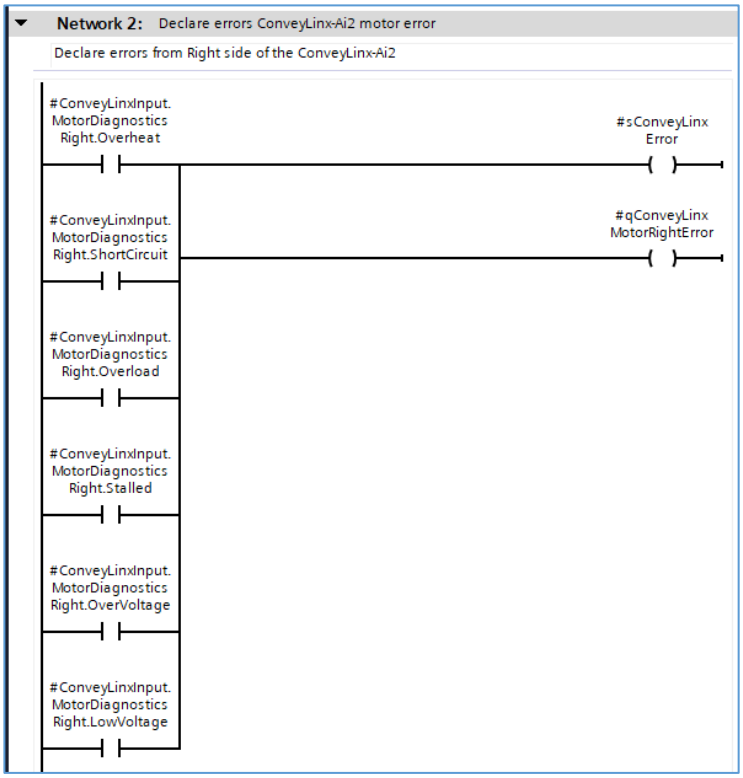


Network 1: Declare errors ConveyLinx-Ai2 motor error.



In network 1, the motor errors connected to the left side of the ConveyLinx-Ai2 module are declared. If one of the errors from the module will be active, the ConveyLinxError will be triggered and the output of the function block qConveyLinxMotorLeftError will be set to TRUE to indicate the error.

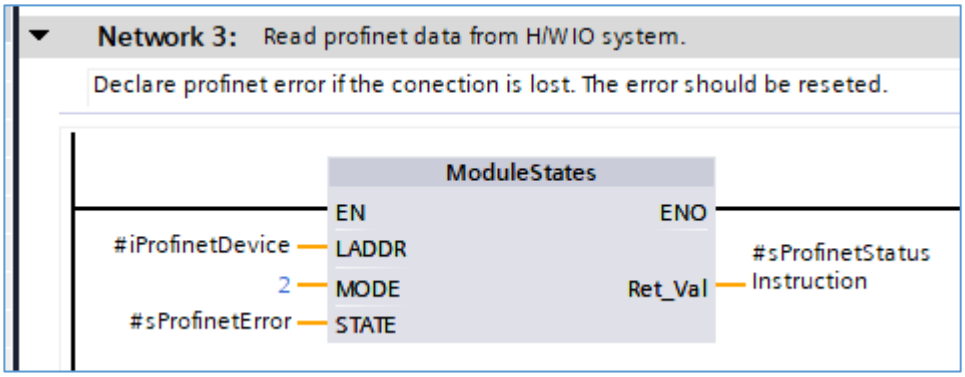
Network 2: Declare errors ConveyLinx-Ai2 motor error.



In network 2, the motor errors connected to the right side of the ConveyLinx-Ai2 module are declared. If one of the errors from the module will be active, the ConveyLinxError will be triggered and the output of the function block qConveyLinxMotorRightError will be set to TRUE to indicate the error.

In case of qConveyLinxMotorRightError and/or qConveyLinxMotorLeftError a physical error on the motor is triggered. Possible causes are the wiring and the motor, it requires a physical intervention. Depending on the error type after solving the issue, it can be acknowledged using reset or the error is not required to be acknowledged. When the error is solved and acknowledged, the output of the block qConveyLinxMotorRightError will be set to FALSE, indicating there is no more active error on the motor.

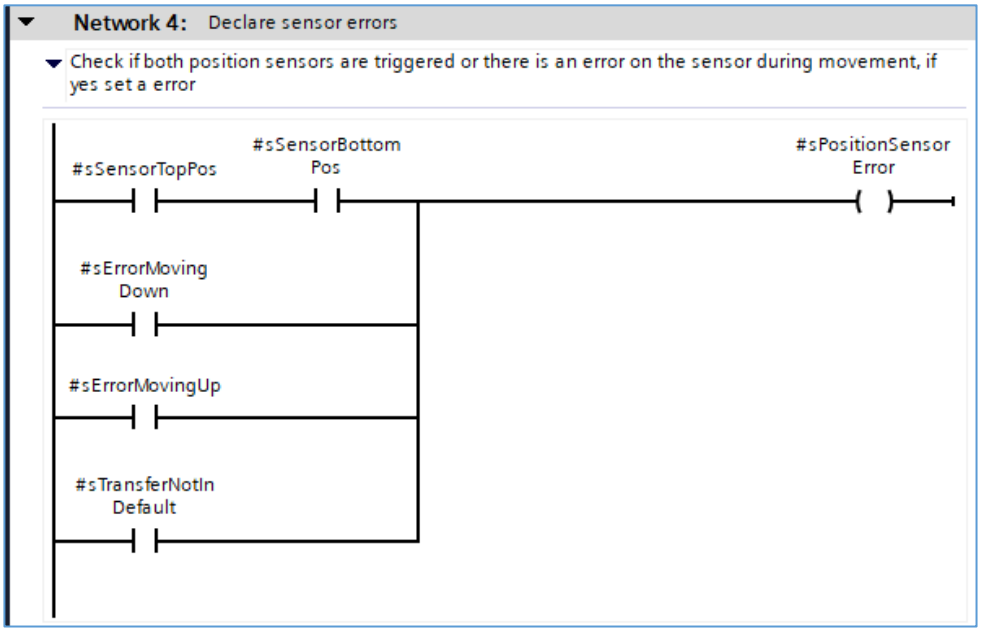
Network 3: Read profinet data from H/W IO system.



In network 3 the profinet connection of the ConveyLinx-Ai2 module is being read. This error can be triggered by the faulty ethernet cable, the module is off/faulty and not communicating or the iProfinetDevice input number was wrongly assigned (explained in chapter 4.2).

When the issue is fixed and the communication is reestablished, the error should be acknowledged using reset.

Network 4: Declare sensor errors.



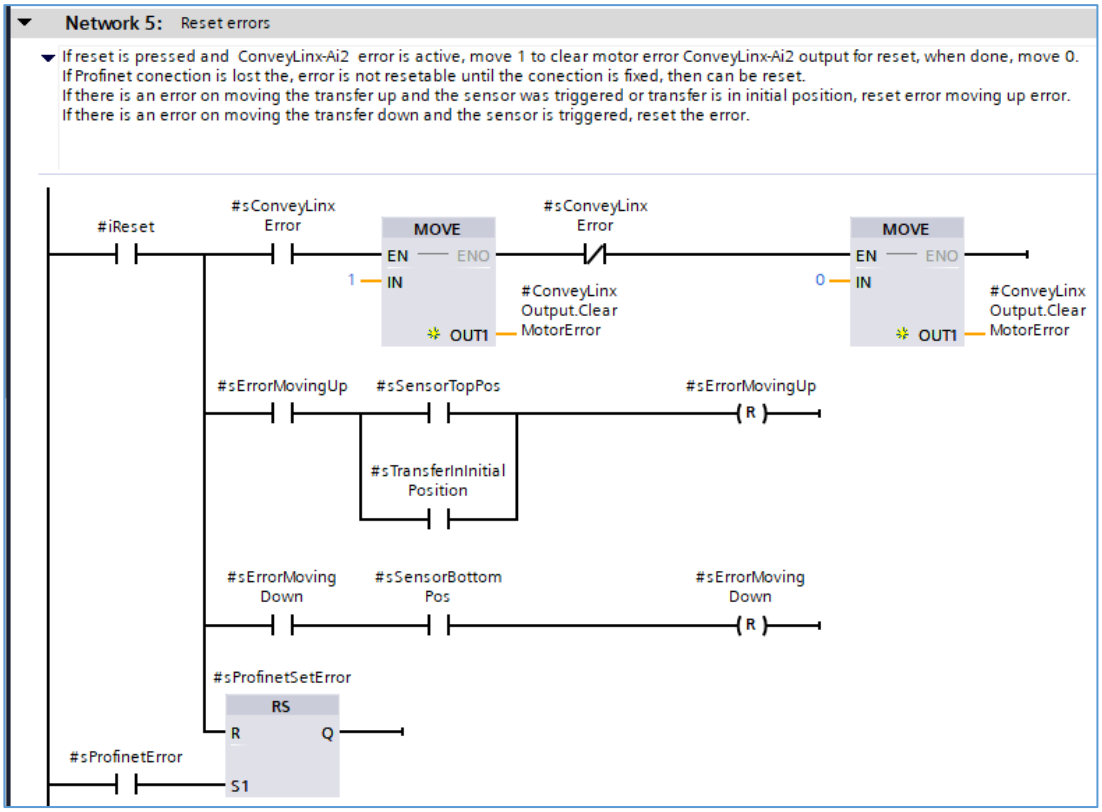
In network 4 the sensor errors are defined. When at least one of the following situation arises, the qPositionSensorError output of the function block will be triggered, indicating a sensor error:

- When both positioning sensors are triggered. If both sensors have signal, the position should be adjusted as of, when the lifting mechanism is down, the bottom sensor only should be triggered, when the mechanism is up, only the top detection sensor should be triggered.

- Error moving up/down. This error is triggered when the move the lifting mechanism signal is triggered but the top or bottom sensor is not triggered in time. This is a result of a faulty sensor or cabling.
- Transfer not in default. This error is triggered when non or only the top position sensor is triggered and the transfer is not busy (transfer busy will be explained in network 9). In this case the initialization process is rolled out (will be explained in network 7).

For all the errors above, a visual/mechanical/electrical inspection of the sensors should be done.

Network 5: Reset errors.



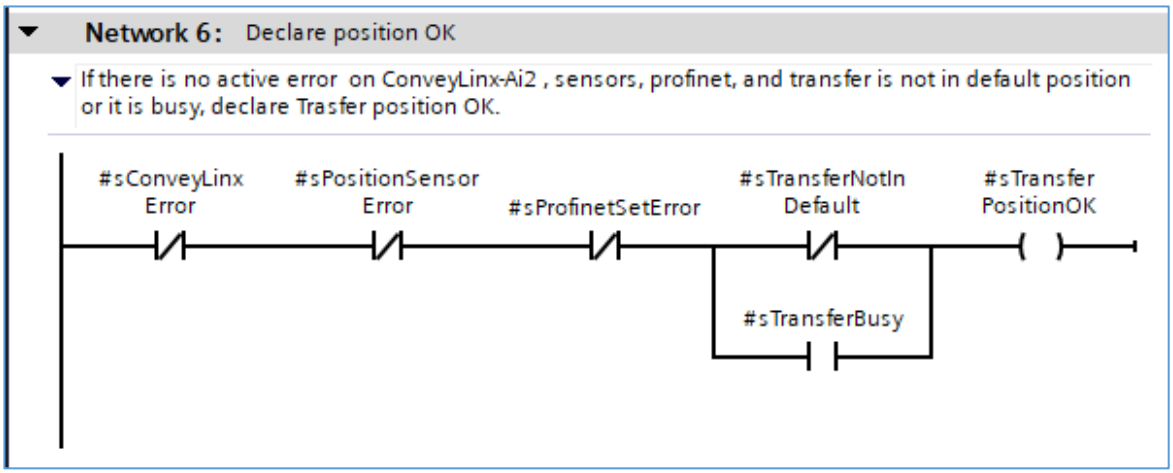
Network 5 is used for resetting the acknowledgeable errors. The logic is the following:

If iReset input is activated and:

- there is an active acknowledgeable error on one of the sides of the ConveyLinx-Ai2 connected motors, move 1 to #CLOutput.ClearMotorError for clearing the error, as soon as the error is not active anymore, set the value back to 0.
- an error moving up is active, the sensor indicating that the mechanism is in top position is triggered (the error on the sensor was fixed and the sensor works properly) or the transfer is in initial position (initialization was activated and passed successfully) then reset error moving up.
- an error moving down is active, the sensor indicating that the mechanism is in bottom position is triggered (the error on the sensor was fixed and the sensor works properly) then reset error moving down.

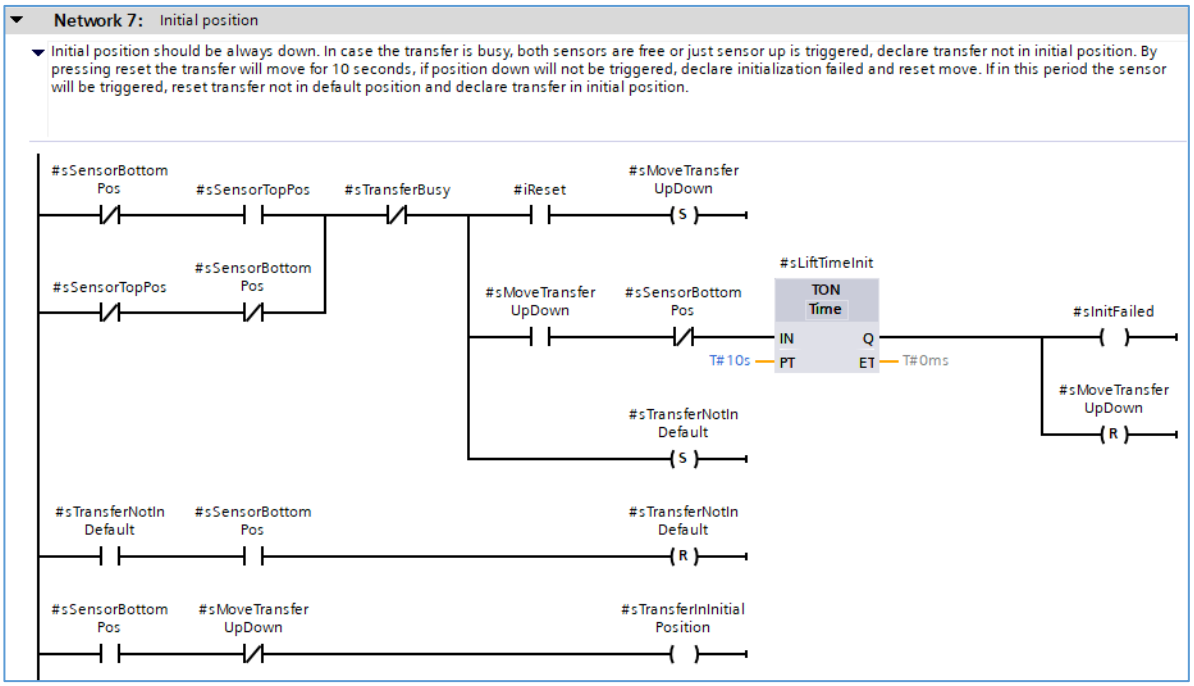
- the profinet error is active and the communication was reestablished then reset qProfinetError output.

Network 6: Declare position OK.



In network 6 the position ok is declared. If there no active errors on the device, then declare qTransferOK output TRUE.

Network 7: Initial position.

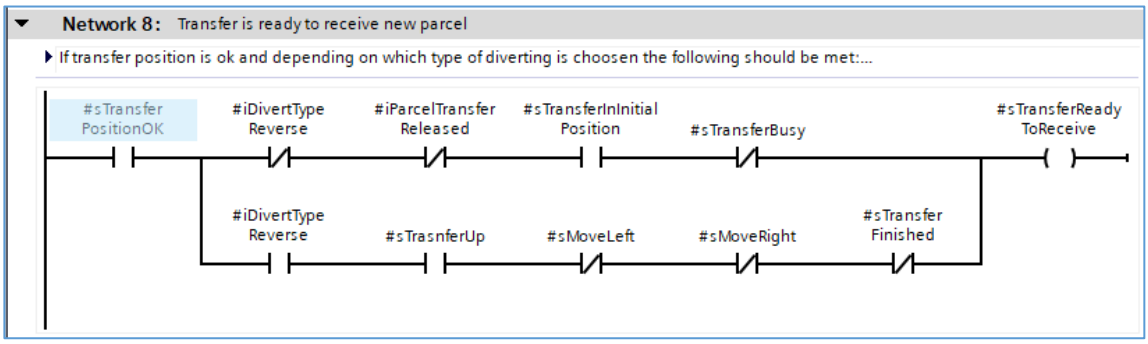


Initialization process: Initial position should be always down. In case the transfer is busy, both sensors are free or just the top positioning sensor is triggered then declare transfer not in initial position.

By pressing reset the transfer will move for 10 seconds, until it detects the bottom sensor, if it won't be detected withing this period, declare initialization failed and reset move lifting mechanism signal.

If within this period the sensor will be triggered, reset transfer not in default position and declare that the transfer is in initial position.

Network 8: Transfer is ready to receive new parcel.



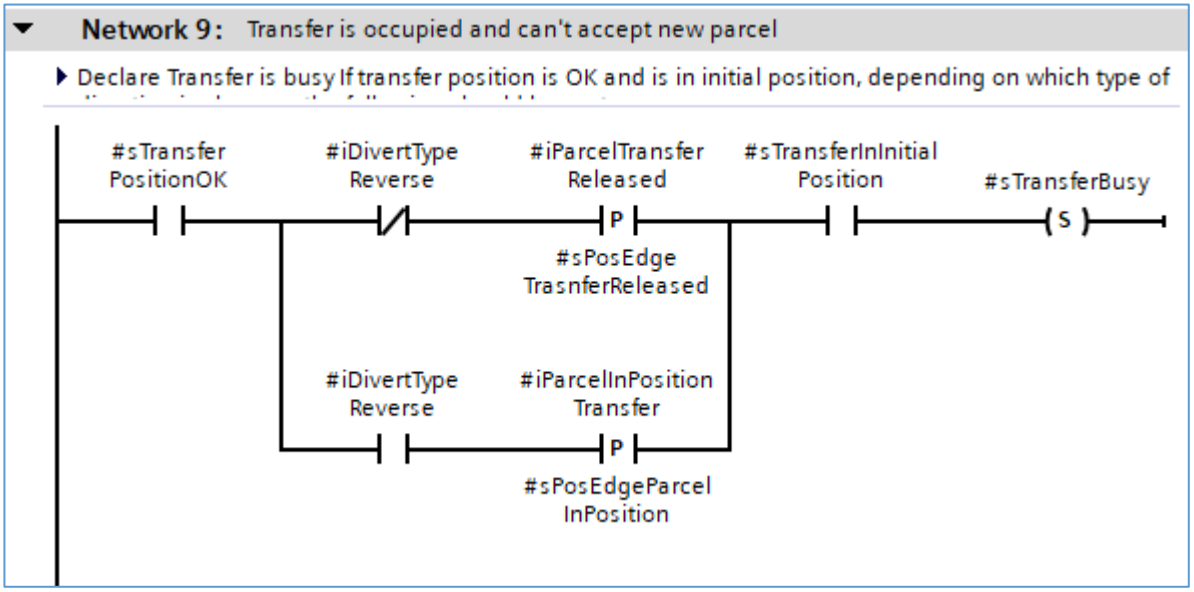
Network 8 is activating the ready to receive new parcel state. The logic is the following:

If transfer position is ok and depending on which type of diverting is chosen the following should be met:

- Divert type reverse is FALSE (the package is approaching from the infeed when it reaches the transfer position it will be lifted and transferred to the outfeed): input parcel released should be false, transfer should be in initial position and it should not busy then declare the transfer is ready to accept new parcel.

- Divert type reverse is TRUE (the package is approaching from the outfeed to the infeed): Transfer should be in up position, motor move left or right and transfer finished should be false, then declare transfer is ready to accept new parcel.

Network 9: Transfer is occupied and can't accept new parcel.

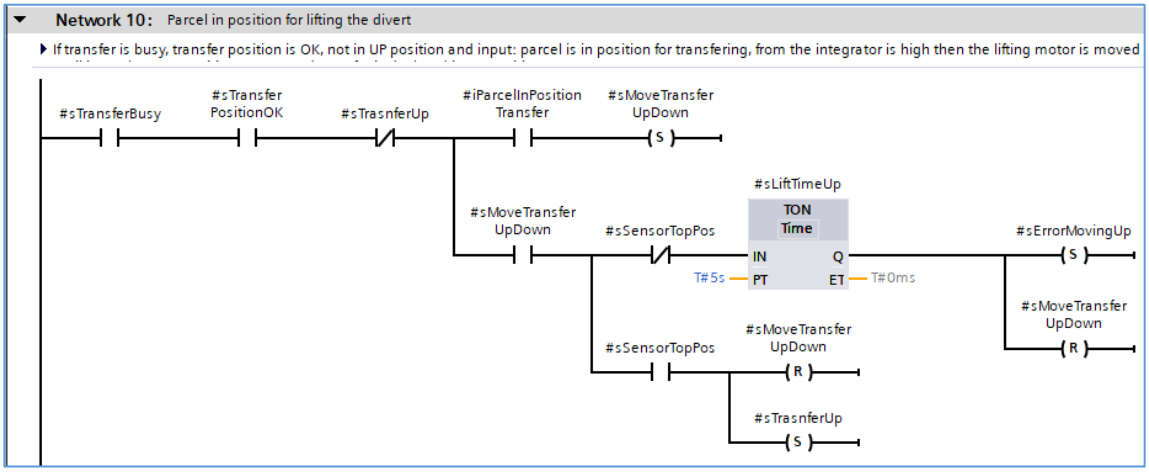


Declare Transfer is busy If transfer position is OK and is in initial position, depending on which type of diverting is chosen the following should be met:

- Divert type reverse is False: input iParcelTransferReleased should be activated.

- Divert type reverse is TRUE: input iParcelInPositionTransfer should be activated.

Network 10: Parcel in position for lifting the divert.

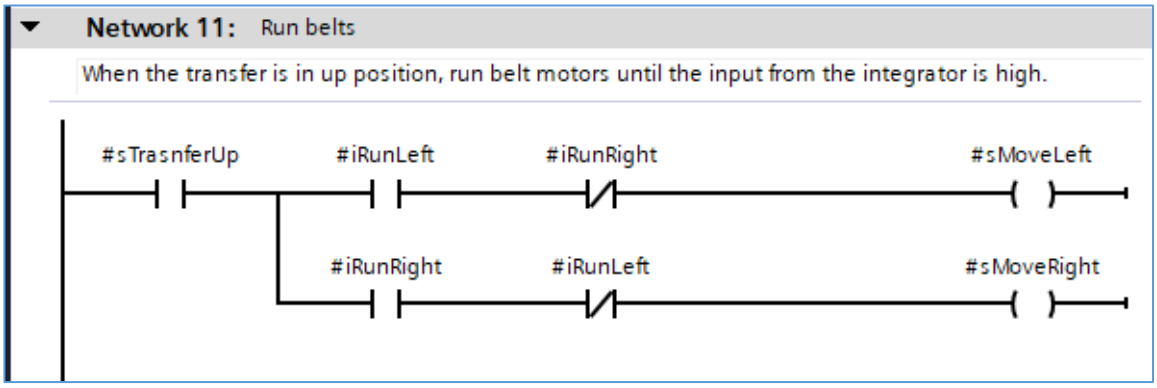


Network 10 is lifting the divert up. In order to lift, the following conditions should be met:

The transfer is busy, transfer position is OK, not in UP position and input: parcel is in position for transferring, from the integrator is high then the lifting motor is moved until it reaches top position sensor and transfer is declared in UP position.

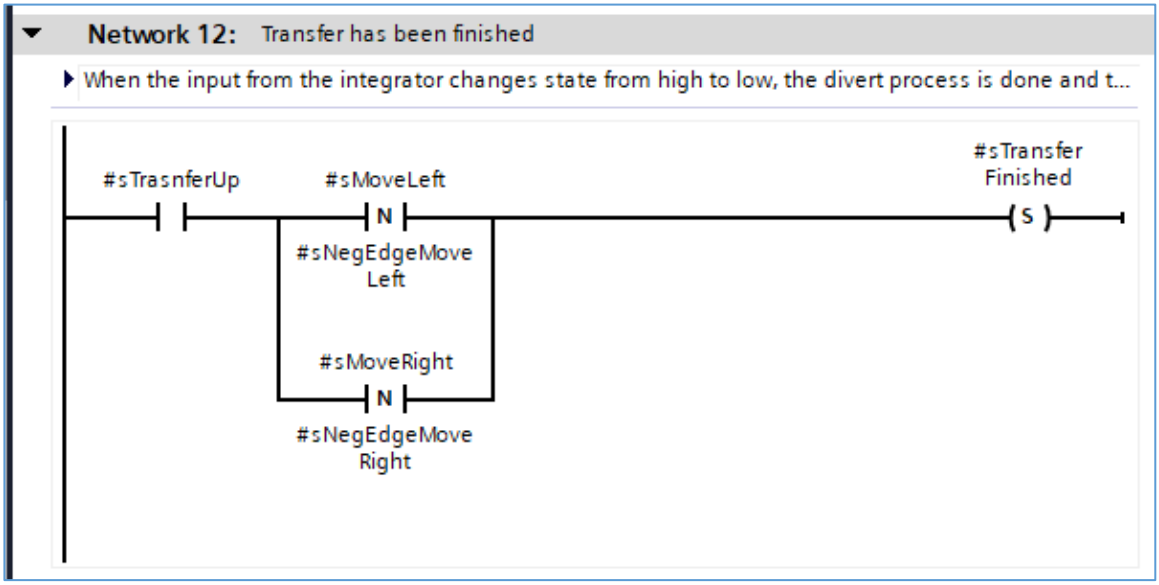
In case the top position sensor is not triggered withing 5 seconds, the movement is stopped and error moving up is declared (the error is described in network 4).

Network 11: Run belts.



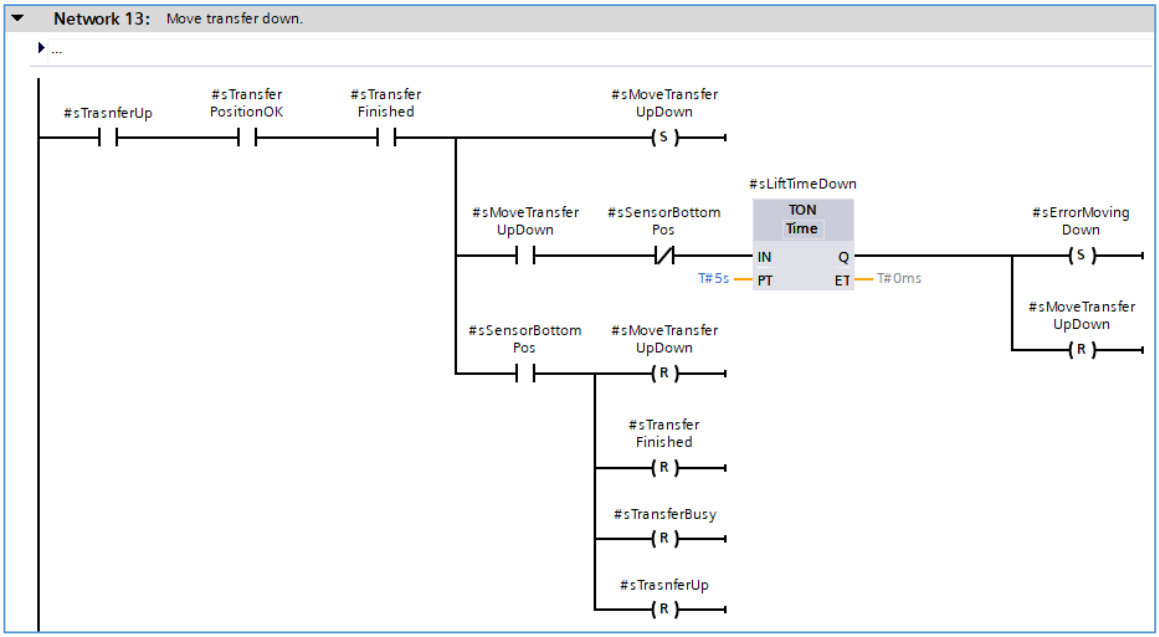
If transfer is up and the signal from the integrator run the belts left or right is high, then set the output for the motor until this signal is high from the integrator.

Network 12: Transfer has been finished.



On the negative edge of the run signal from the integrator and transfer is in up position, declare transfer is finished.

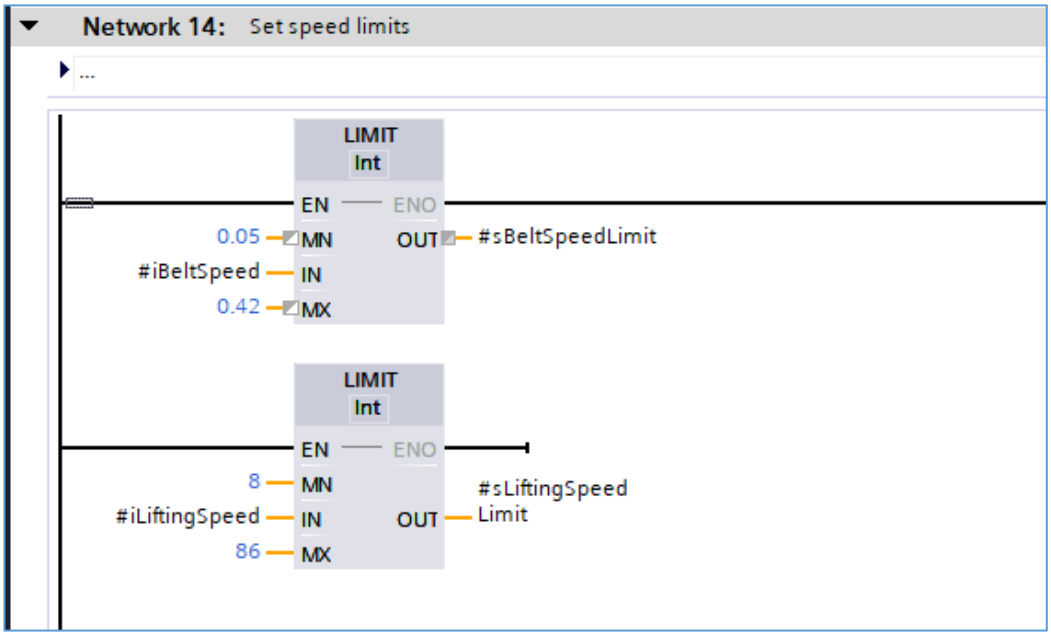
Network 13: Move transfer down.



If transfer is in up position, position is ok and transfer is finished, move the transfer down until it reaches bottom position sensor, then reset transfer finished, transfer busy and transfer up bits that were set in the previous networks. Transfer is not busy anymore and can accept new TU.

If during movement the bottom position sensor has not been triggered in 5 seconds, movement is stopped and error moving down is declared (the error is described in network 4).

Network 14: Set speed limits.

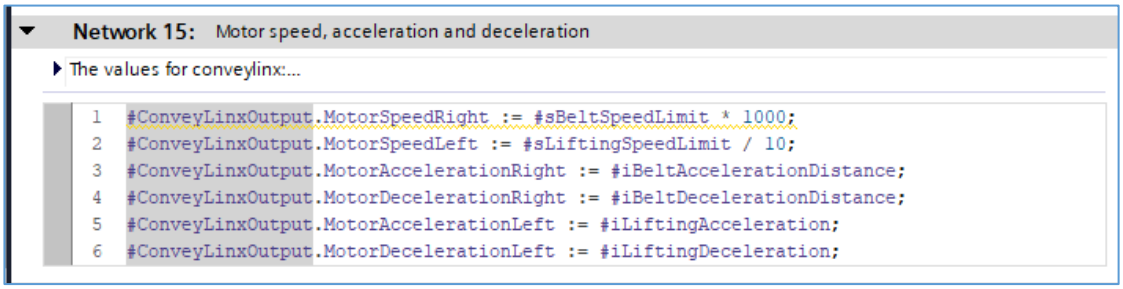


In network 14, the speed limits are declared:

- Belt speed limits: min 0.05 m/s; max 0.42 m/s.
- Lifting speed limits: min 8 RPM; max 86 RPM.

If the value on the input is outside these values, the maximum or minimum value will be applied for the motor.

Network 15: Motor speed, acceleration and deceleration.



The values the ConveyLinx-Ai2 module is operating are the following:

- For MDR value is in mm/s
- For PGD value is in RPM X 10

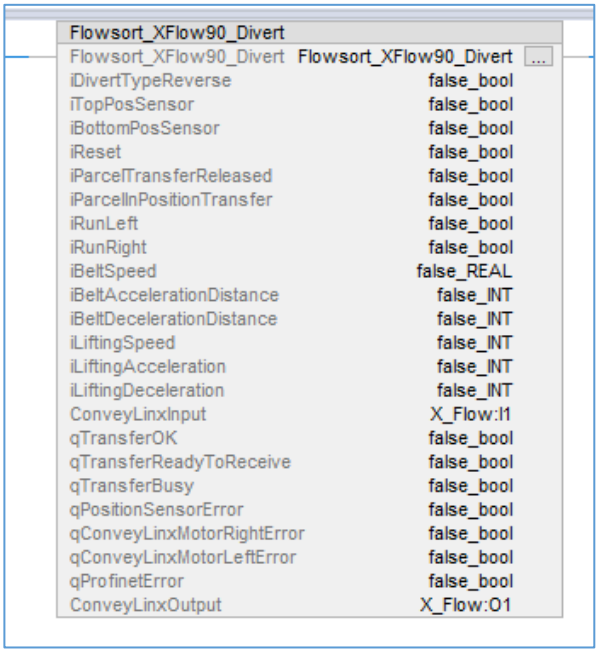
In network 15, the conversion to m/s and RPMs is applied.

Network 16: Add static to output.

The direct outputs were not used in the software, instead static variables were created. In Network 16, the static values are moved to the function block outputs.

4.6 Program integration for ALLEN BRADLEY STUDIO 5000 V35

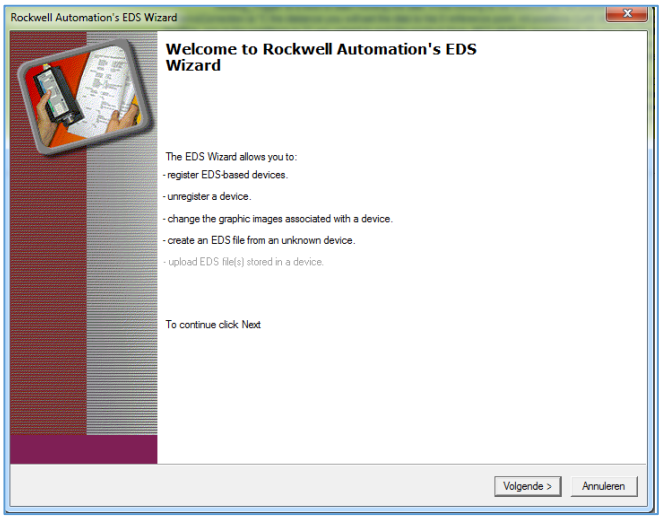
For the use of the Flowsort_XFlow90_Divert with a Allen Bradley CompactLogix/ControlLogix, follow the steps described below.



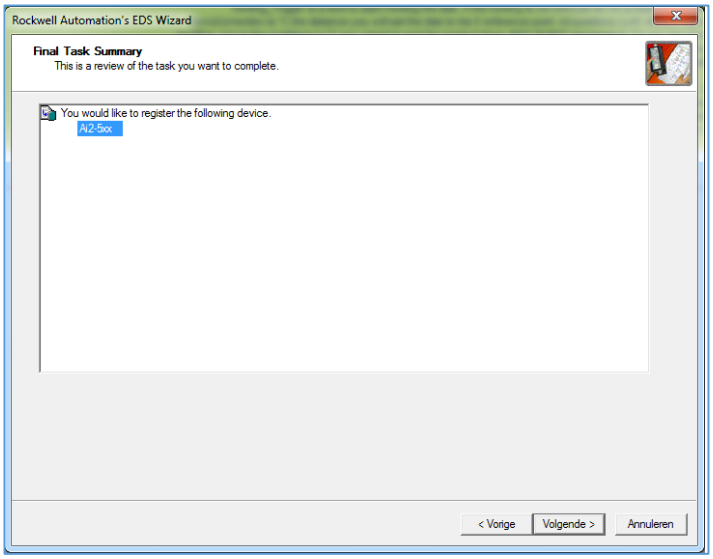
4.6.1 ALLEN BRADLEY STUDIO 5000 Hardware

To integrate the Flowsort 24V Diverter in a CompactLogix or ControlLogix Controller, first the EDS file of the controller must be downloaded. This can be done on the Pulseroller's website: https://pulseroller.com/downloads/. Go to Software and Firmware Downloads -> PLC Connectivity -> PLC – Ethernet IP Files -> ConveyLinx-Ai2 -> Latest, and Download the file: EDS & AOI Vx x

Now the EDS file must be imported into the project. This can be done in the Tools menu of the Logix Designer under EDS Hardware Installation Tool. The Following Popup will be shown:

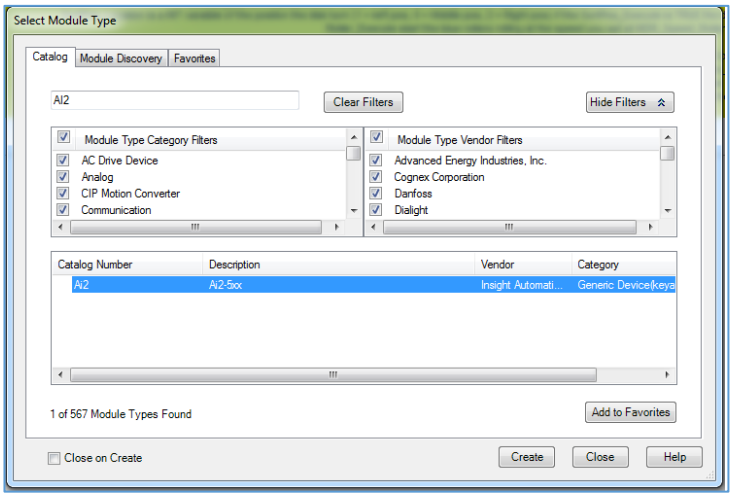


Click Next in this Popup. In the next window Select "**Register an EDS file**" and click next. In the next window select "**Register a single file**" and Browse to the directory where the unpacked EDS file is located and click Open. When the directory is chosen click next. When the path was OK the Installation Test Result will be good. Again click next in this popup. In the following popup it is possible to edit the image of the module. This is not necessary. Again click next in this window. In The final Task summary an overview is shown of the files that are going to be registered:



Again click next. After this a confirmation of the successfully registered EDS file will popup. Click Finish to complete the procedure.

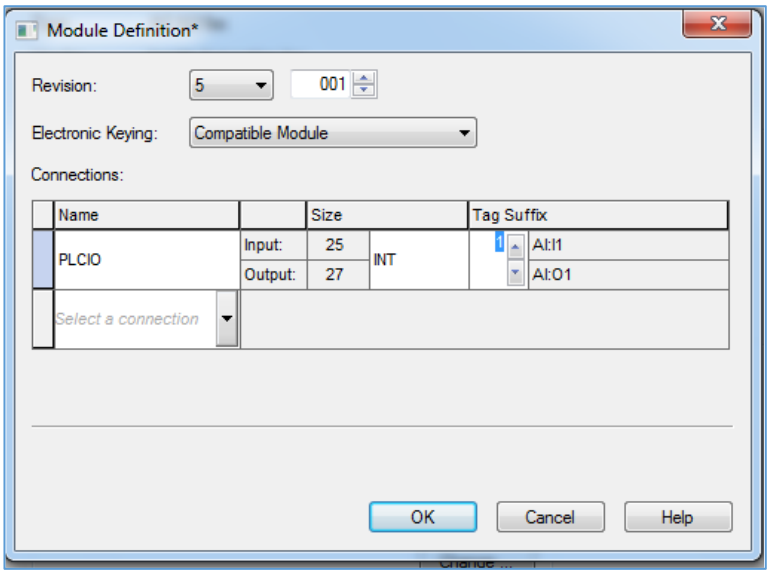
The Controller is now available to add to the ethernet/IP network. Click on the Ethernet/IP network where the Motionlinx controller is going to be connected to and click "**New Module**" In the following popup the ConveyLinx-Ai2 module can be searched:



Choose the AI2 and click on Create. The "New Module" popup will be shown. Here the name and IP address can be set by the integrators wishes.

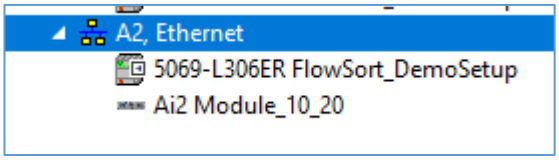
Now the correct connection parameters need to be set. This can be done in the module definition window, click Change. In the window Module Definition that will popup the connection needs to be

"PLCIO". The Sizes of the input and output are defined by the connection. However the Type needs to be changed to "INT". For the integrator it is possible to set a tag suffix to the hardware.



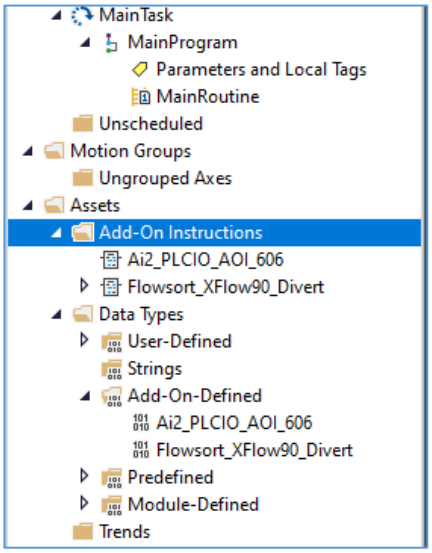
When the settings are correct, click OK. A Confirmation popup will be shown that the module definitions will be changed, Click Yes. In the New Module popup also Click OK.

Now the module is available in the I/O configuration.



4.6.2 ALLEN BRADLEY STUDIO 5000 Software

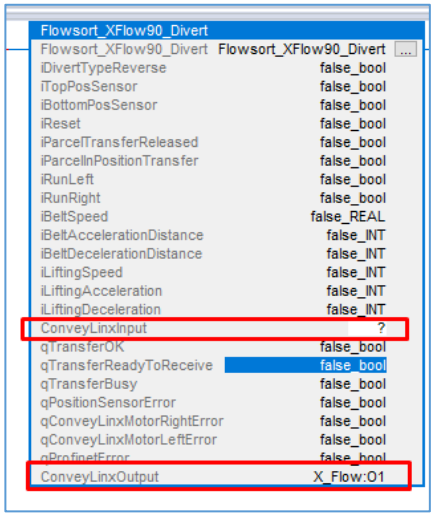
In the Demo program, the following blocks are available. For the integration of the Flowsort function block, copy the Add-On Instructions folder to the project.



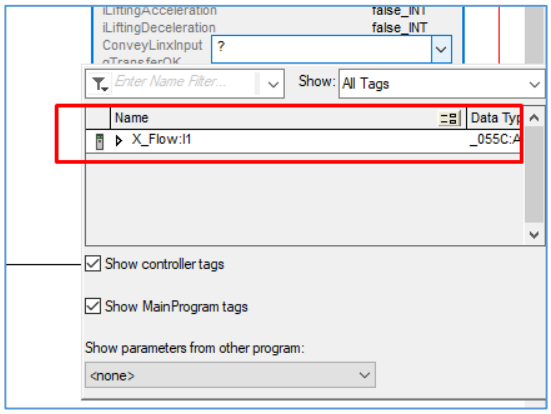
The In/Output datatype of the Function block will be automatically generated when the module is add in the hardware. The Inputs/Outputs and the logic is explained above in chapter 4.5 Function Block: explanation.

4.6.3 ALLEN BRADLEY STUDIO 5000 CONTROLLER LINK SOFTWARE TO HARDWARE

The link from the software to the hardware can be done by double clicking on the “ConveyLinxInput” or “ConveyLinxOutput”. A Dropdown menu appears.



Click on the arrow and a popup will appear where the “hardware” module can be selected.

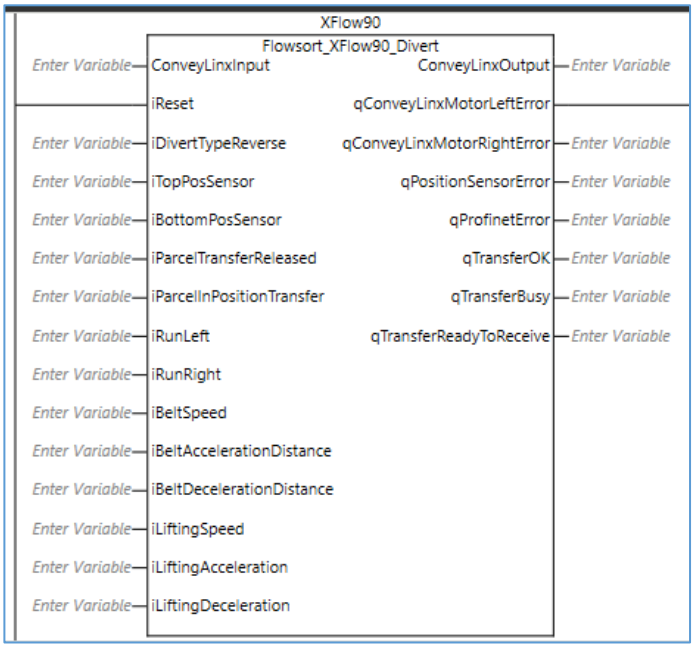


Select the module that needs to be linked to the function block. Do the same for the “ConveyLinxOutput” parameter.

Now the function block is linked to the module in the hardware configuration.

4.7 Program integration for OMRON SYSMAC STUDIO V1.58.0

For the use of the Flowsort_XFlow90_Divert with a Omron CPU Unit, follow the steps described below.



4.7.1 OMRON SYSMAC STUDIO HARDWARE AND LINK TO SOFTWARE

To integrate the Flowsort_XFlow90_Divert in a Omron Controller with Sysmac Studio, first the EDS file of the controller must be downloaded. This can be done on the Pulseroller's website: https://pulseroller.com/downloads/. Go to Software and Firmware Downloads -> PLC Connectivity -> PLC – Ethernet IP Files -> ConveyLinx-Ai2 -> Latest, and Download the file: EDS & AOI Vx x

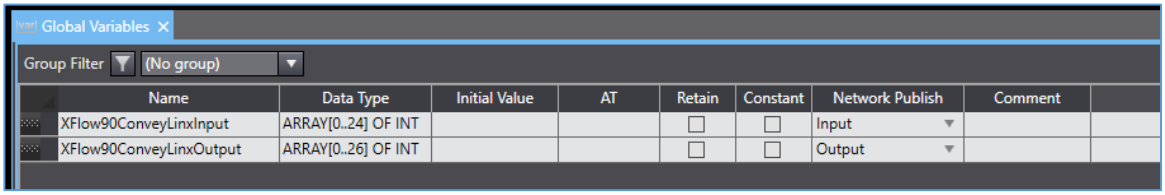
Now the EDS file must be placed into the installation folder of Sysmac Studio. Normally this should be:

C:\Program Files (x86)\OMRON\Sysmac Studio\IODeviceProfiles\EipConnection\Eds .

But can be different when Sysmac Studio is installed in another folder. Extract the Zip file and copy the EDS file into this folder.

In the Multiview Explorer go to Programming -> Data -> Global Variables.

In this window create an Input variable with datatype ARRAY[0..24] OF INT and an Output variable with datatype ARRAY[0..26] OF INT.

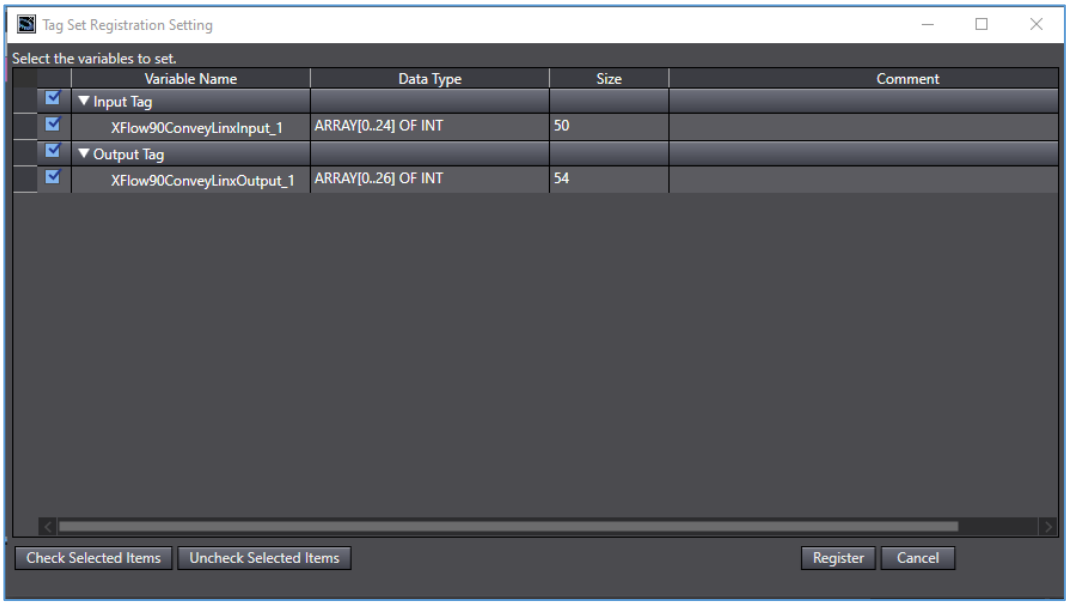


Open the Ethernet/IP connection settings via Tools -> Ethernet/IP connection settings.

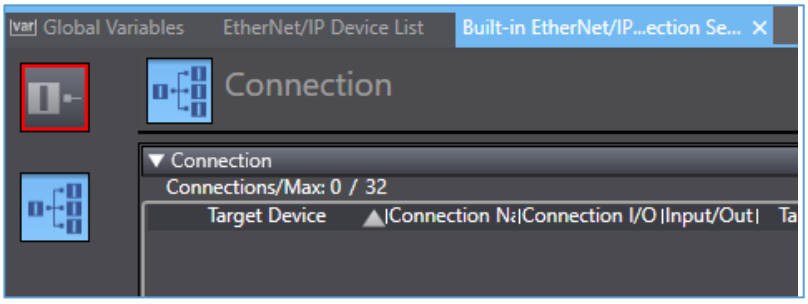
In the window that opens select the Ethernet/IP Device at which you want to connect the Diverter to. Right-click on the device and click “edit”.

In the opened Ethernet/IP connection settings window, Click on “Registration All”.

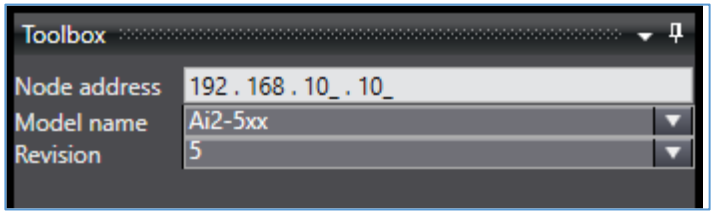
In the opened window select the In and Output tags you want to register, and click “Register”.



Now the tags are registered. Now go to the Connection window in the Ethernet/IP connection settings window.



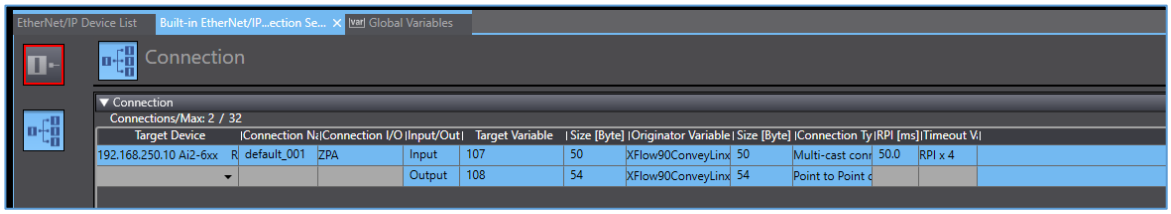
In the Connection window at the right hand side the toolbox is located. In this window click on the “+” to add a new target device. In the window fill in the Node address of the Diverter and select the model name and revision. The model name should be:



When the settings are filled in correctly click “Add” at the bottom of the toolbox.

Now the target device is added in the toolbox. Drag and drop the device in the connections table.

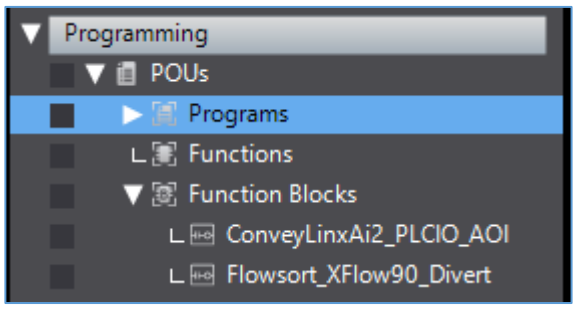
As Connection I/O Select “PLCIO” as target variable select 107 for the Input and 108 for the Output. As Originator Variable select the Tag sets that are registered. When all the settings are filled in correctly everything will have a light blue color. When a setting is pink that means that the setting is not correct.



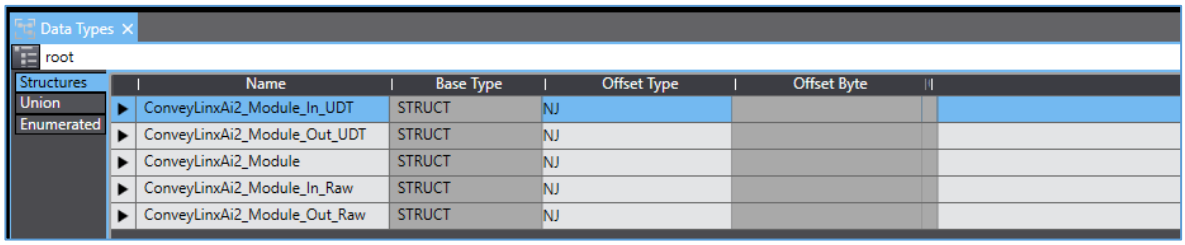
Now the connection is configured and linked to the software. Make sure the Node address (IP) is correctly configured on the controller of the diverter.

4.7.2 OMRON SYSMAC STUDIO Software

In the Demo program, the following blocks are available. For the integration of the Flowsort function block, copy the Function blocks from the example project to the project.



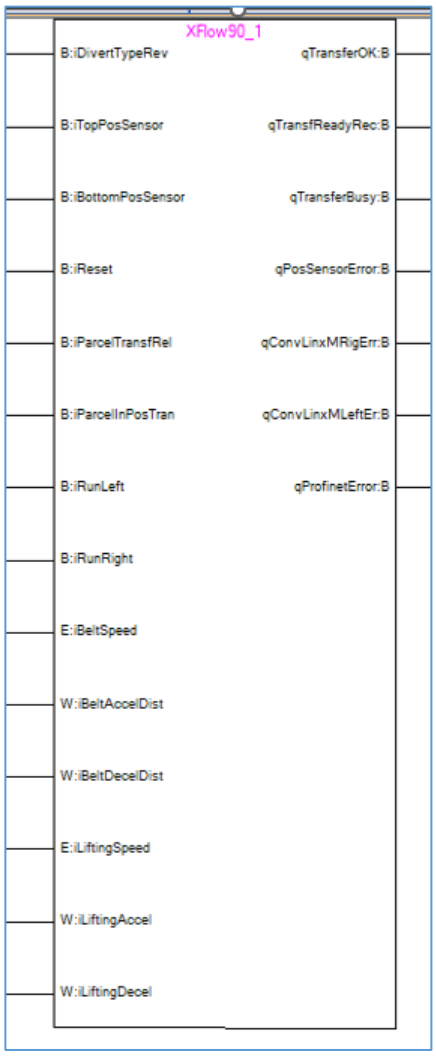
Also copy the datatypes structures from the example project to the project.



The Inputs/Outputs and the logic is explained above in chapter 4.5 Function Block: explanation.

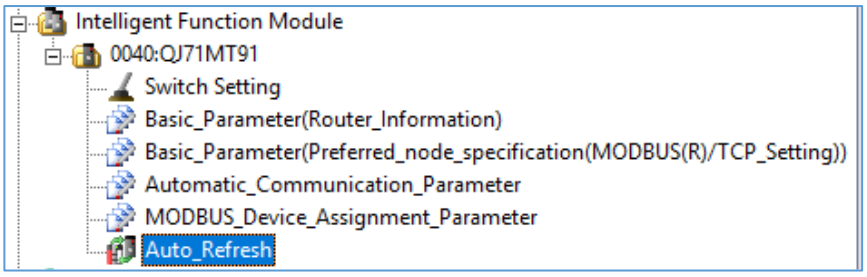
4.8 Program integration for MITSUBISHI GX WORKS 2 OR HIGHER

For the use of the Flowsort_XFlow90_Divert with a Mitsubishi Q Series Controller via Modbus, follow the steps described below.



4.8.1 MITSUBISHI GX WORKS Hardware

To integrate the Flowsort_XFlow90_Divert in a Mitsubishi Q-Series Controller via modbus the following settings should be configured (in this example we used a QJ71MT91 card). The settings can be configured under the Intelligent Function Module in the project tree.



The Basic_Parameters (Preferred_node_specification) should be configured as follows:

MODBUS/TCP Setting	**The parameter setting concerning the MODBUS/TCP setting.**
Local Slave Station Port No.	502
Target Slave Port No. for Automatic Communication Function	502
PLC Response Monitoring Timer Value	10

Then the automatic Communication Parameters should be configured as follows (per Device):

Automatic Communication Parameter	**Set the automatic communication parameters when using the automatic**
Automatic Communication Parameter 1	**The parameter setting concerning the automatic communication.**
Target Station IP Address	192.168.202.20
Module ID	255
Repetition Interval Timer Value	10
Response Monitoring Timer Value	0
Type Specification of The Target MODBUS Device	0505h:Read/Write Holding Registers
Read Setting	**The parameter setting concerning reading data from slave.**
Head Buffer Memory Address	1000 h
Target MODBUS Device Head Number	1699
Access Points	25
Write Setting	**The parameter setting concerning writing data to slave.**
Head Buffer Memory Address	3000 h
Target MODBUS Device Head Number	1799
Access Points	27

The IP-address of the device depends on the configuration of the ConveyLinx-Ai2 module.

The type specification of the Target Modbus device must be 0505h. The Head Buffer memory address of the Read settings and write settings can be set customer specific these are the memory addresses in the PLC buffer where the modbus data will be stored. This address must be unique within the Automatic Communication Parameters. This Adress is not the address that is used in the PLC! This address is in the Auto Refresh parameter. The Target MODBUS Device Head Number must be 1699 for the Read settings and 1799 for the Write settings. This is the address of the Modbus holding registers in the ConveyLinx-Ai2 module. The Access Points for the Read settings must be 25, and for the write settings this must be 27. These are the number of registers that are read/written in the ConveyLinx-Ai2 module.

For adding more ConveyLinx-Ai2 modules the same parameters with a different IP-Address and with different Head Buffer memory addresses.

4.8.2 MITSUBISHI GX WORKS LINK SOFTWARE TO HARDWARE RECEIVE DATA

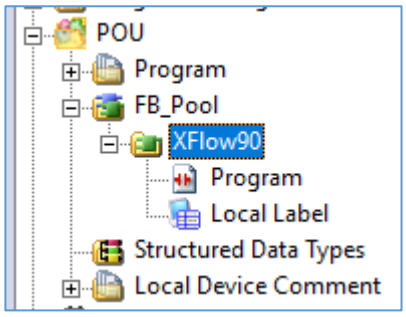
The Auto_Refresh settings are customer specific:

Item	
Transfer to PLC	**The data of the buffer memory is transmitted to the specified device.**
Auto Communication Function Buffer Input	D1000 (0,2150)
Auto Communication Function Operation Status Storage Area (Parameter 1 to 64)	
User Setting Area (Input)	
Transfer to Intelligent Function Module	**The data of the specified device is transmitted to the buffer memory.**
Auto Communication Function Buffer Output Area	D6000 (0,2150)
User Setting Area (Output)	

This means that the data configured in de Function Buffer Input is stored from PLC-address D1000 till address D3150 (these are the data read from the ConveyLinx-Ai2 modules). The data configured in the Function Buffer Output is stored from PLC-address D6000 till address D8150 (these are the data that is written to the ConveyLinx-Ai2 modules).

4.8.3 MITSUBISHI GX WORKS Software

In the Demo program, the following blocks are available. For the integration of the Flowsort function block, copy the XFlow90 function block to the project.



The Inputs/Outupts and the logic is explained above in chapter 4.5 Function Block: explanation.

4.9 Program integration for BECKHOFF TWINCAT 3.4.3147.18

The Flowsort_XFlow90_Divert can be controlled with a MotionLinx-Ai Controller or a ConveyLinx-Ai2 controller. For the ConveyLinx-Ai2 controller a ProfiNET IO Controller option must be added to the configuration.

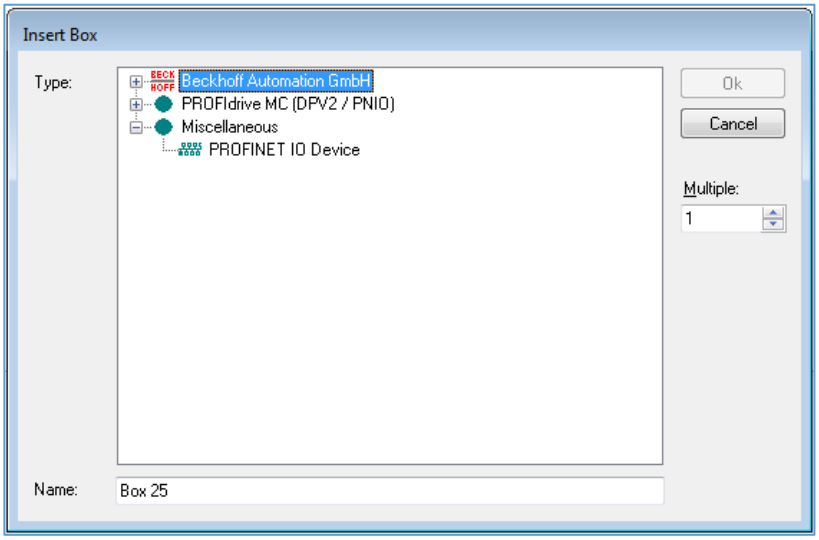
```
PROGRAM MAIN
VAR
    xflow90 : XFlow90;

END_VAR
```

```
XFlow90(
    iDivertTypeReverse:= TRUE,
    iTopPosSensor:= ,
    iBottomPosSensor:= ,
    iReset:= ,
    iParcelTransferReleased:= ,
    iParcelInPositionTransfer:= ,
    iRunLeft:= ,
    iRunRight:= ,
    iBeltSpeed:= ,
    iBeltAccelerationDistance:= ,
    iBeltDecelerationDistance:= ,
    iLiftingSpeed:= ,
    iLiftingAcceleration:= ,
    iLiftingDeceleration:= ,
    iProfinetDevice:= ,
    ConveyLinxInput:= ,
    qTransferOK=> ,
    qTransferReadyToReceive=> ,
    qTransferBusy=> ,
    qPositionSensorError=> ,
    qConveyLinxMotorRightError=> ,
    qConveyLinxMotorLeftError=> ,
    qProfinetError=> ,
    ConveyLinxOutput=> );
```

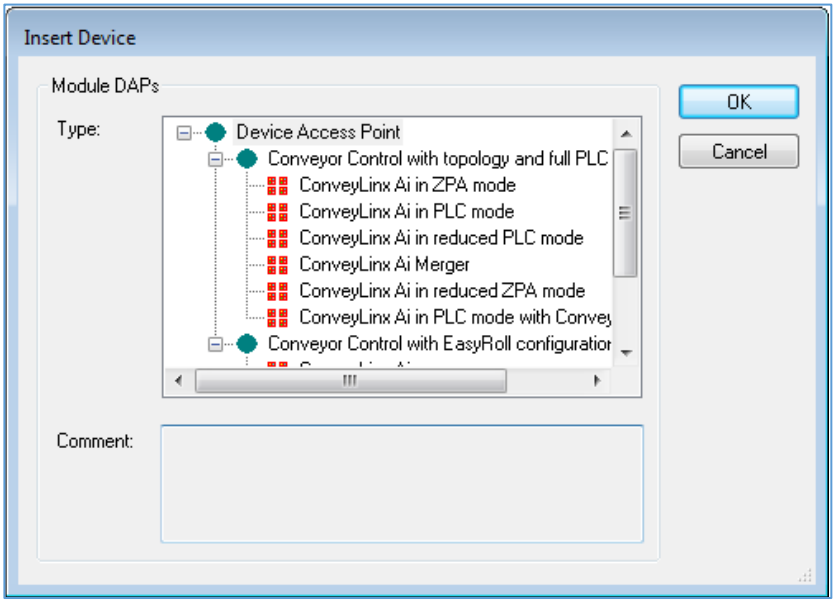
4.9.1 BECKHOFF TWINCAT Hardware

To integrate the Flowsort_XFlow90_Divert in a Beckhoff Controller, , first the GSDML file of the controller must be downloaded. This can be done on the Pulseroller's website: https://pulseroller.com/downloads/. Go to Software And Firmware downloads -> PLC Connectivity -> PLC – ProfiNET GSDML files -> ConveyLinx-Ai2 -> download the file called: GSDML.

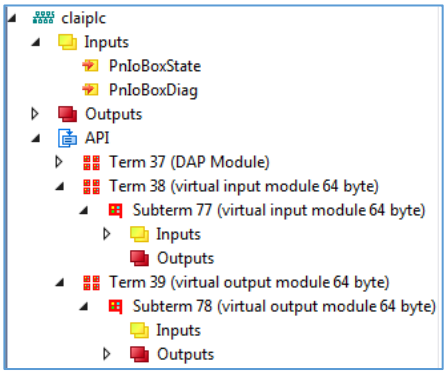
Now the ConveyLinx-Ai2 controller can be added to the project. In the project under I/O -> Devices, go to the ProfiNET master at which the ConveyLinx-Ai2 controller is connected to and then push insert. The following popup will be shown:



Select the PROFINET IO Device under miscellaneous and click OK. Now select the folder where the unpacked GSDML file is located. Select the .XML file and click Open. The following popup will be shown. Select the ConveyLinx-Ai2 in PLC mode and click OK.



Now the ConveyLinx-Ai2 Controller is added to the project. In the Solution Explorer the ConveyLinx-Ai2 Controller is available with all the data. The data is available in the API as an Array of 64 Bytes.



Make sure that the checkbox Swap LOBYTE and HIBYTE is checked! This can be found under the controller -> API -> Term (Virtual input module 64 byte) -> Inputs -> Inputs. Double click on the Inputs and go to the tab “Flags”. Check the checkbox Swap LOBYTE and HIBYTE. Do this also for the outputs!

Under the API go to Term (DAP Module) and go to the Subterm (ConveyLinx-Ai2 in PLC mode). Go to the tab “Parametrize Module”.

The following Module parameters must be set in the properties of the ConveyLinx-Ai2.

Left side_Motor mode -> ECO Mode

Left side_Brake mode -> Servo brake method

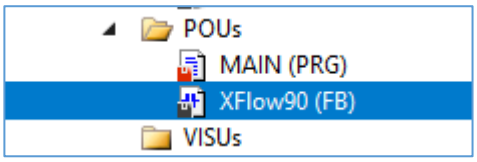
Right side_Motor mode -> BOOST mode

Right side_Brake mode -> Free brake method

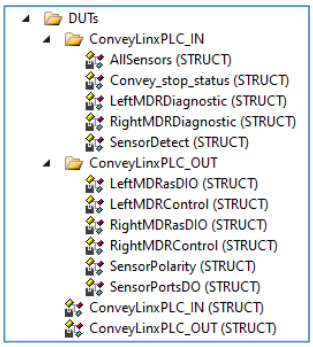
The speeds, accelerations and decelerations can be set MAIN (PRG) corresponding inputs.

4.9.2 BECKHOFF TWINCAT Software

In the Demo program, the following blocks are available. For the integration of the Flowsort_XFlow90_Divert, copy the Flowsort_XFlow90_Divert into the project.

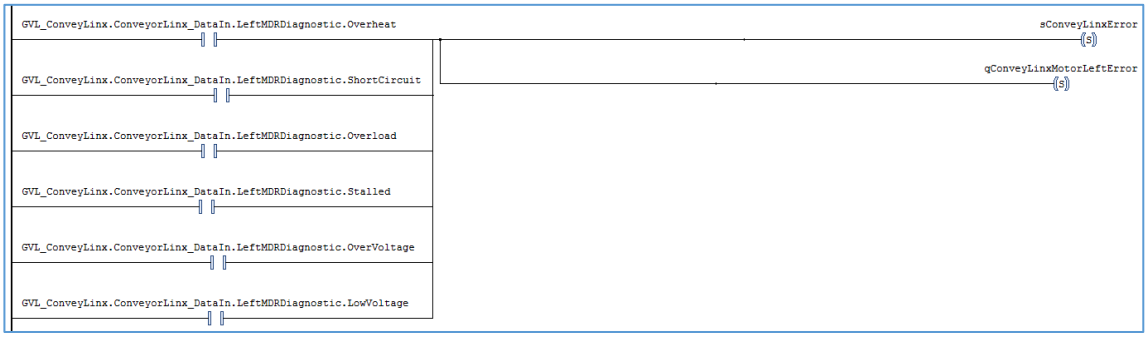


Also copy the data unit types from the Demo program to the project:



The Inputs/Outputs and the logic is explained above in chapter 4.5 Function Block: explanation.

The link from the software to the hardware are done by a global variable and are used in the function block:



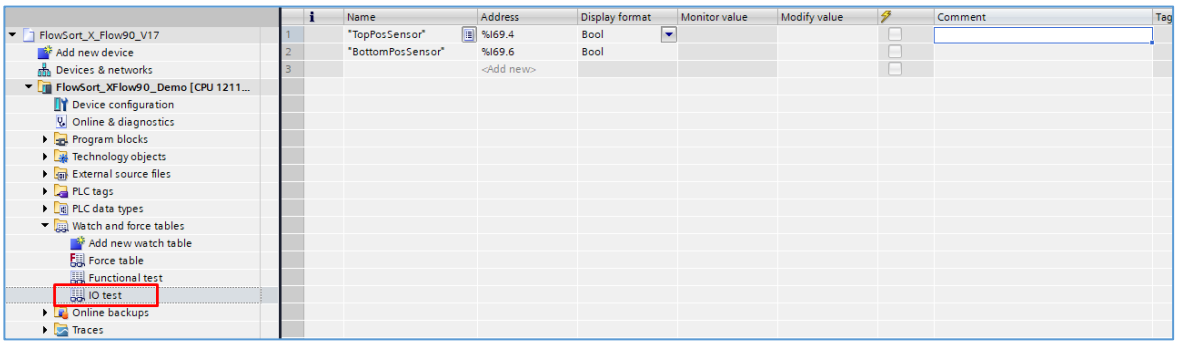
5. IO and Functional tests.

In order to verify that the hardware and electrical everything is fine, it is necessary to perform and IO and functional test before sending actual orders to the X-Flow90 device. In this chapter it will explained hot to commission the device.

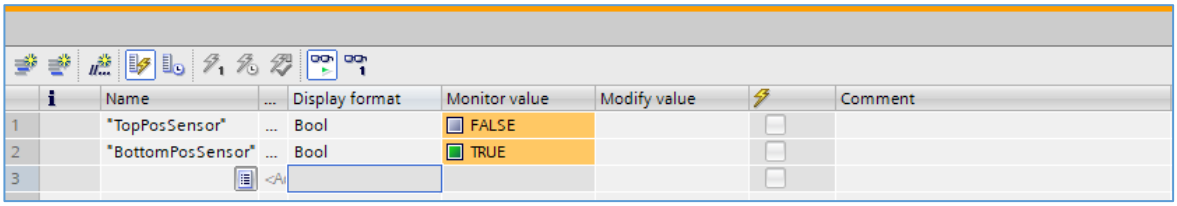
5.1 IO test.

The IO test contains the signal check of the top and bottom sensors, for this it is necessary to open the IO test watch table:

Open project tree -> Watch and force tables -> IO test



Go online and press the monitor all values:
It is necessary to trigger them and check the signal in the watch table, as these sensors are inductive it is necessary to use a steel material to trigger them:



In the picture above, the Bottom positioning sensor has been triggered. Make sure to trigger both and monitor the signal.

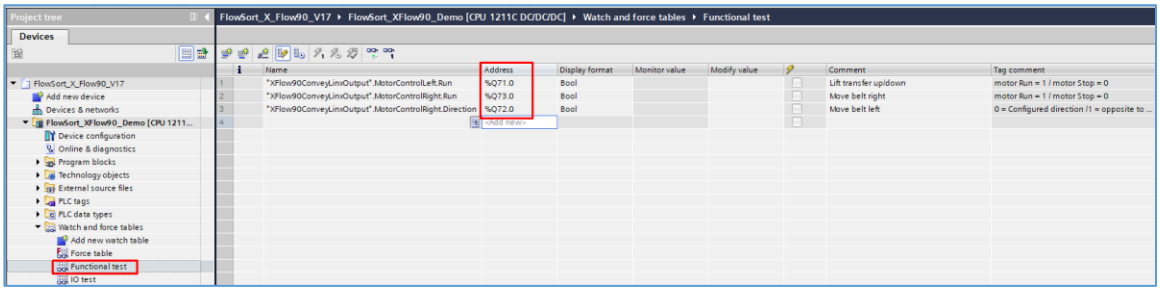
If necessary, check the cabling and connector.

5.2 Functional test.

During this test the motors of the device will be checked and run.

It is necessary to open the functional test watch table:

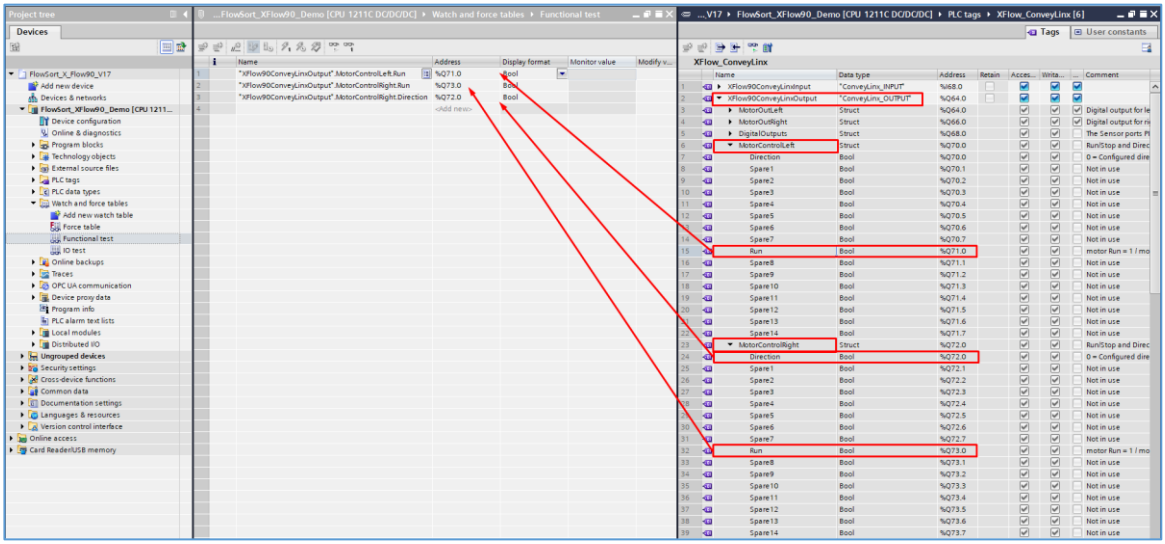
Open project tree -> Watch and force tables -> Functional test.



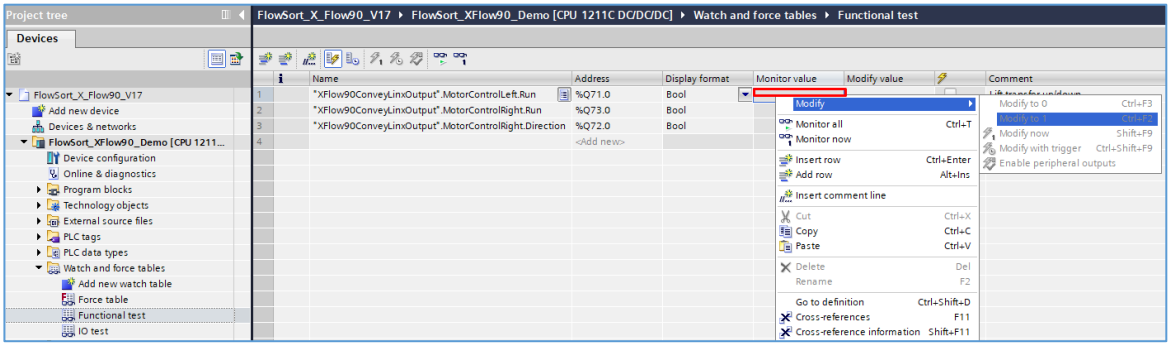
The address is different, depending on configuration of the ConveyLinx-Ai2 module but the name should be as following:

- "XFlow90ConveyLinxOutput".MotorControlLeft.Run for lifting up and down
- "XFlow90ConveyLinxOutput".MotorControlRight.Run for moving the belt
- "XFlow90ConveyLinxOutput".MotorControlRight.Direction for moving the belt in the opposite direction.

The addresses can be extracted from: PLC tags -> XFlow_ConveyLinx -> " XFlow90ConveyLinxOutput -> MotorControlLeft and MotorControlRight.



Go online and press the monitor all values:



During monitoring, right click on monitor value of the "XFlow90ConveyLinxOutput".MotorControlLeft.Run -> Modify -> Modify to 1. The lift motor should be activated (the lifting part of the device should go up and down). Stop the lift motor in up position by "XFlow90ConveyLinxOutput".MotorControlLeft.Run -> Modify -> Modify to 0 when the mechanism is in up position. Check if the "TopPosSensor" is triggered by the device, if necessary adjust.

The next test is the belt motor, activate: "XFlow90ConveyLinxOutput".MotorControlRight.Run -> Modify -> Modify to 1. The belt motor should run.

Activate "XFlow90ConveyLinxOutput".MotorControlRight.Direction -> Modify -> Modify to 1. The belt motor should change direction.

When all the test above are passed successfully, deactivate: "XFlow90ConveyLinxOutput".MotorControlRight.Run -> Modify -> Modify to 0.

"XFlow90ConveyLinxOutput".MotorControlRight.Direction -> Modify -> Modify to 0. The belt should stop running.

"XFlow90ConveyLinxOutput".MotorControlLeft.Run -> Modify -> Modify to 1 so the lift motor runs and deactivate when in down position. Check if "BottomPosSensor" is triggered, if necessary adjust.

If all the above steps are done, the device is ready for operation.

6. Tips and tricks/lessons learned

Add information of the files for the rest of the platforms