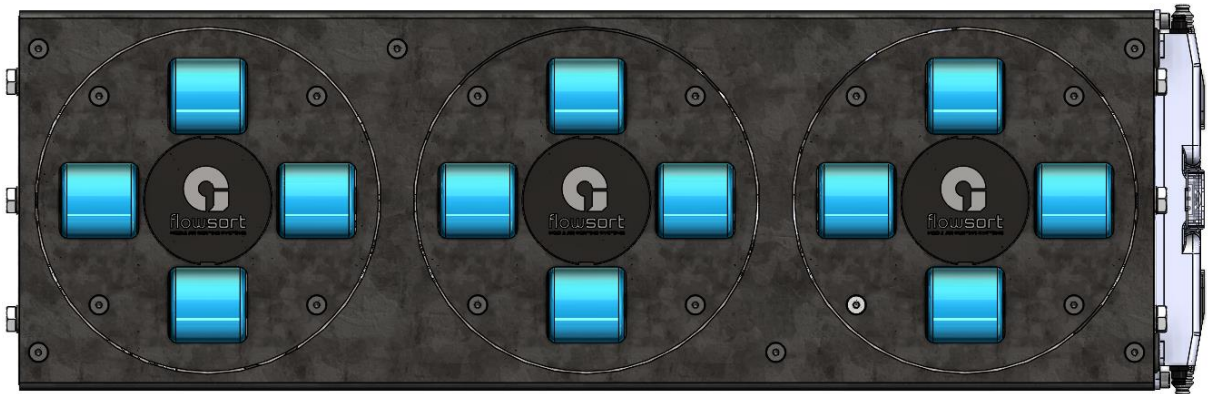


Integration Manual

Integration Instructions

- SLD - Single Line Diverter 24V**
- DLD - Double Line Diverter 24V**
- ZZ – ZigZag Diverter 24V**



Manufacturer's address:

Flowsort Kft.

Gyár u. 2.

2040 Budaörs

Hungary

T: +36 23 880-977

W: www.flow-sort.com

E: info@flow-sort.com

Version: 3.1.2

16-08-2024

The copyright of these installation and operating instructions remains with Flowsort BV. The installation and operating instructions contain technical regulations and drawings which may not be reproduced partially or in full, transmitted by any means, utilized without permission for competitive purposes or disclosed to third parties.

TABLE OF CONTENTS

TABLE OF CONTENTS.....	- 3 -
Electrical Specifications.....	- 5 -
1.1 Motor Specifications	- 5 -
1.2 Controller Connection	- 5 -
1.3 Connection Diagram.....	- 5 -
1.4 Bus Specification	- 5 -
Controller Information.....	- 6 -
2.1 Controller Information	- 6 -
2.2 Controller Usage.....	- 7 -
2.3 Disk Motor Control (Left PGD)	- 7 -
2.4 Roller Motor Control (Right PGD).....	- 9 -
2.5 Homing Sequence Disks	- 10 -
Example Function Block	- 12 -
3.1 General Information.....	- 12 -
3.2 General Functioning	- 12 -
3.2.1 Input Variables	- 13 -
3.2.2 Output Variables	- 15 -
3.3 Program integration for Siemens Tia Portal V15.1 or Higher (ProfiNET).....	- 17 -
3.3.1 Hardware.....	- 17 -
3.3.2 Software	- 19 -
3.3.3 Controller Link Software to Hardware	- 20 -
3.4 Program integration for Siemens Tia Portal V15.1 or Higher (ModBus).....	- 23 -
3.4.1 Hardware.....	- 23 -
3.4.2 Software	- 23 -
3.4.3 Controller Link Software to Hardware Receive data	- 25 -
3.4.4 Controller Link Software to Hardware Send data.....	- 26 -
3.5 Program integration for Siemens Simatic Manager V5.6 or Higher (ProfiNET)	- 27 -
3.5.1 Hardware.....	- 28 -
3.5.2 Software (ProfiNET Onboard)	- 31 -
3.5.3 Software (ProfiNET CP-Card).....	- 32 -
3.5.4 Controller Link Software to Hardware (ProfiNET Onboard)	- 33 -
3.5.5 Controller Link Software to Hardware (ProfiNET CP-Card).....	- 34 -
3.6 Program integration for Beckhoff TwinCAT V3.1.4024.0 or Higher.....	- 35 -
3.6.1 MotionLinx-Ai Controller Hardware.....	- 35 -
3.6.2 MotionLinx-Ai Controller Software	- 37 -
3.6.3 MotionLinx-Ai Controller Link Software to Hardware	- 38 -
3.6.4 Conveylinx-Ai2 Controller Hardware	- 38 -
3.6.5 Conveylinx-Ai2 Controller Software.....	- 40 -
3.6.6 Conveylinx-Ai2 Controller Link Software to Hardware.....	- 42 -
3.7 Program integration for Allen Bradley Studio 5000 V32 or Higher	- 43 -
3.7.1 Hardware.....	- 43 -
3.7.2 Software	- 46 -
3.7.3 Controller Link Software to Hardware	- 47 -

3.8	Program integration for Omron Sysmac Studio 1.40.0 or Higher	- 48 -
3.8.1	Hardware and Link to Software	- 48 -
3.8.2	Software	- 50 -
3.9	Program integration for Mitsubishi GX Works 2 or Higher	- 53 -
3.9.1	Hardware	- 53 -
3.9.2	Controller Link Software to Hardware Receive data	- 54 -
3.9.3	Software	- 55 -
3.10	Program integration for Omron CX-Programmer Version 9	- 56 -
3.10.1	Hardware and link to the software	- 57 -
3.10.2	Software	- 61 -
Appendix A – Updating the firmware		- 63 -

ELECTRICAL SPECIFICATIONS

1.1 MOTOR SPECIFICATIONS

The motors used in the diverters are small and robust motors. They work on 24 voltage, they have a nominal output of 50W and the nominal current is 2.5A. The starting current can be around 3A.



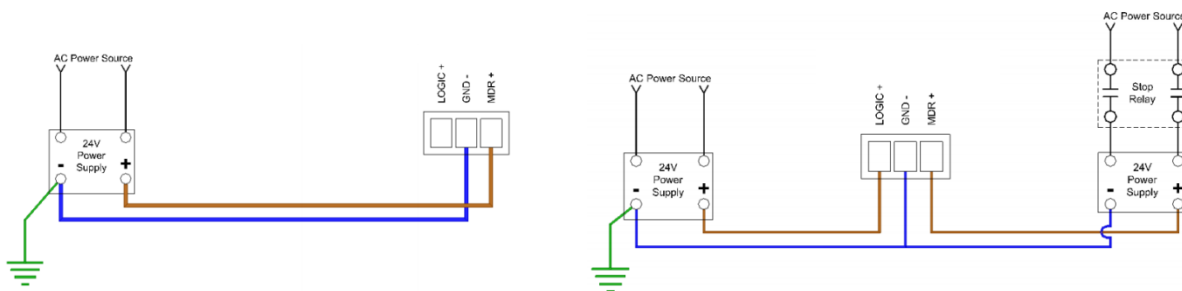
1.2 CONTROLLER CONNECTION

The controllers used in the diverters are advanced intelligence, high end networked motor drive controllers.



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 – 2,5mm ² (28 - 12AWG)
Network Link	Link Left and Link Right – RJ-45 style Ethernet network connection between modules including LED Indicators

1.3 CONNECTION DIAGRAM



1.4 BUS SPECIFICATION

Baud Rate	10 Mbps / 100 Mbps
Network Protocols	PROFINET, Ethernet I/P, Modbus/TCP, EtherCAT**
PLC type	All major PLC manufacturers (Siemens, Beckhoff, Allen Bradley, ...)

** specify bus protocol when ordering module

CONTROLLER INFORMATION

2.1 CONTROLLER INFORMATION

The SLD / DLD / ZZ modules come standard with Conveylinx-Ai2 controllers. They support the following protocols: PROFINET, Ethernet I/P, Modbus / TCP. On request, you can also choose the Motionlinx-Ai controller when ordering, which works on the EtherCAT bus protocol.

In the following document, a basic principle will be explained which variable is required for your chosen controller type. However, the method of control can be chosen entirely according to the programmer's wishes and can be expanded with additional functions depending on the need.

For the full set of variable and possibilities you can read the manuals of the specific controller at <https://www.pulseroller.com/downloads/> Here you can also find specific documents on how to connect to specific PLC brands.

ConveyLinx-Ai Family Complete Guide:

<https://www.flow-sort.com/images/files/pulseroller/conveylinx-ai-product-family-complete-guide.pdf>

MotionLinx Complete Guide:

<https://www.flow-sort.com/images/files/pulseroller/motionlinx-product-family-complete-guide.pdf>

For some PLC brands you also need optional files. ESI, GSDML, ... files. These can be found per controller in the software and firmware download section.

Profinet GSDML file:

https://www.flow-sort.com/images/files/pulseroller/Profinet_GSDML_files.zip

EtherCAT ESI file:

https://www.flow-sort.com/images/files/pulseroller/MotionLinx-Ai_ESI.xml.zip

Ethernet I/P EDS and AOI files:

https://www.flow-sort.com/images/files/pulseroller/Ethernet-IP_EDS_and_AOI_Dateien.zip

CC-Link IEFB file:

https://www.flow-sort.com/images/files/pulseroller/CC-Link_IEFB_ConveyLinx-Ai.zip

Step7 UTP file:

https://www.flow-sort.com/images/files/pulseroller/UDTs_for_STEP7.scl.zip

TIA Portal UTD file:

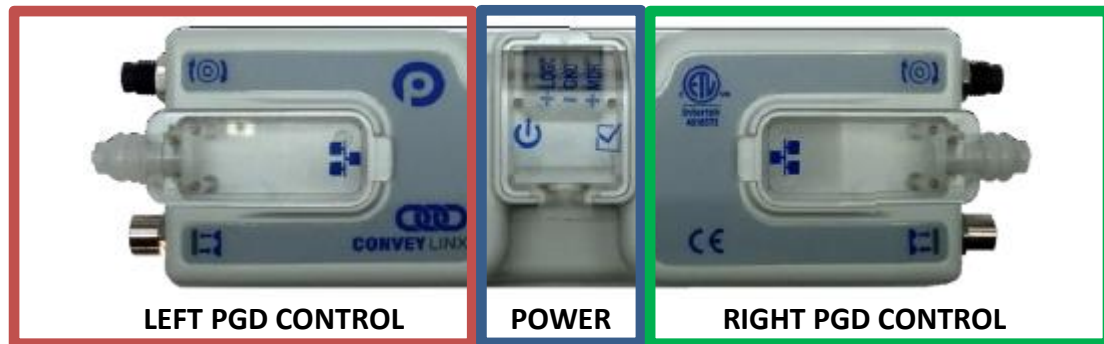
https://www.flow-sort.com/images/files/pulseroller/UDTs_for_TIA_Portal.zip

Always check on the website for the latest version!

2.2 CONTROLLER USAGE

An SLD, DLD or ZZ module comes standard with one controller per row. A controller thus operates a row for turning as well as for the transport rollers.

From the factory, all sorting movements are connected on the left and the transport rollers on the right. The links that must be made in the software therefore relate to the left and right side of the controller.



We recommend setting the Conveylinx-Ai2 controller in Full PLC mode via the EasyRoll program. This gives you the advantage that the controller works completely as an IO-module and full control can be done by the PLC. The setting for this mode can also be found in the developer manual of the manufacturer.

The Motionlinx-Ai controller is a standard IO-module without internal logic and should therefore not be set. Here, the settings are made exclusively at PLC level.

2.3 DISK MOTOR CONTROL (LEFT PGD)

The left side of the controller is used to control the rotation of the sorter module. An homing sensor is also provided here to detect the zero point. It is recommended to perform the rotational movement with standard commands as well as the extended servo commands for positioning the discs.

The PGD-motor for the rotary movement is standard delivery of the type with gear ratio 15. This means that a full 360° rotation of the motor (30 * 15) gives 450 pulses / 360°.

The formula below can be used to calculate the angle for the number of pulses that the motor has to move for a desired angle. The mechanical ratio of the gears is 1:2.

Formula: Pulses = (desired angle * 30 * gear ratio * 2) / 360

Example for 45°: Pulses = (45 * 30 * 15 * 2) / 360 = 112.50

Since the value to be set is an INT, the result should be rounded to a integer value.

For the controls the following output parameters are used:

OUTPUTS	ESD (Ethercat)	Offset (Profinet)	Min	Max
Left Motor Speed Reference	SpeedLeft	BYTE 20(HI)/21(LO)	380	3870
Left Motor Acceleration Ramp	AccelLeft	BYTE 24(HI)/25(LO)	10	10000
Left Motor Deceleration Ramp	DecelLeft	BYTE 26(HI)/27(LO)	10	10000
Left Motor Brake Method	BrakeModeLeft	BYTE 10(HI)/11(LO)	3 (See guide for detail)	3 (See guide for detail)
Left Motor Run / Reverse	MDRCtrlLeft	BYTE 8(HI)/9(LO)	(See guide for detail)	(See guide for detail)
Clear Motor Error	ClearMotorError	BYTE 32(HI)/33(LO)	(See guide for detail)	(See guide for detail)
Left Motor Servo Command Word	ServoCtrl	BYTE 48(HI)/49(LO)	(See guide for detail)	(See guide for detail)
Left Motor Servo Command Distance	ServoLeft	BYTE 46(HI)/47(LO)	(See guide for detail)	(See guide for detail)

For the controls the following input parameters are used:

INPUTS	ESD (Ethercat)	Byte (Profinet)	Min	Max
Sensor Port Inputs	AllSensors	BYTE 2(HI)/3(LO)	BIT 00	(See guide for detail)
Left Motor Servo Status	ServoStateL	BYTE 42(HI)/43(LO)	(See guide for detail)	(See guide for detail)
Left Motor Status	DiagnosticLeft	BYTE 14(HI)/15(LO)	(See guide for detail)	(See guide for detail)

Value of speed, voltages, temperatures, etc. as a feedback from the controller can be added in the code. All these values can be found under the chapter "PLC I/O MODE ASSEMBLY INPUTS" of the developers guide. The integrator is free to choose to use these values.

2.4 ROLLER MOTOR CONTROL (RIGHT PGD)

The right side of the controller is used to control the blue transport rollers. There are only a few parameters used to be controlled. As standard it is equipped with a PGD motor type gear ratio 11. This motor has a minimum of 52 RPM and a maximum of 528.9 RPM. Some projects with diverters could be delivered with another gear ratio.

The wheel diameter of the transport rollers is 58 mm. The speed settings in the controllers should be multiplied by 10 to use in the controls. The actual speed on the rollers can be calculated with the following formula.

The minimum speed (RPM x 10) = 520 pulse which results in 0.15 m/s on the blue wheels
The maximum speed (RPM x 10) = 5289 pulse which results in 1.55 m/s on the blue wheels

Formula: $v = (\pi * 0.058 * \text{RPM}) / 60$

Example (maximum speed): $v = (3.14 * 0.058 * 528.9) / 60 = 1.606 \text{ m/s}$

We calculate with a mechanical loss of max 5% so that the aforementioned value is lower than the calculated value.

For the controls the following output parameters are used:

OUTPUTS	ESD (Ethercat)	Byte (Profinet)	Min	Max
Right Motor Speed Reference	SpeedRight	BYTE 22(HI)/23(LO)	520	5289
Right Motor Acceleration Ramp	AccelRight	BYTE 28(HI)/29(LO)	(See guide for detail)	(See guide for detail)
Right Motor Deceleration Ramp	DecelRight	BYTE 30(HI)/31(LO)	(See guide for detail)	(See guide for detail)
Right Motor Brake Method	BrakeModeRight	BYTE 16(HI)/17(LO)	1 (See guide for detail)	1 (See guide for detail)
Right Motor Run / Reverse	MDRCtrlRight	BYTE 14(HI)/15(LO)	(See guide for detail)	(See guide for detail)
Clear Motor Error	ClearMotorError	BYTE 32(HI)/33(LO)	(See guide for detail)	(See guide for detail)

Value of speed, voltages, temperatures, etc. as a feedback from the controller can be added in the code. All these values can be found under the chapter "PLC I/O MODE ASSEMBLY INPUTS" of the developers guide. The integrator is free to choose to use these values.

2.5 HOMING SEQUENCE DISKS

For the correct functioning of the disks it is mandatory for the system to know in which position the disk is. The disks can be turned manually when there's no power on the controller.

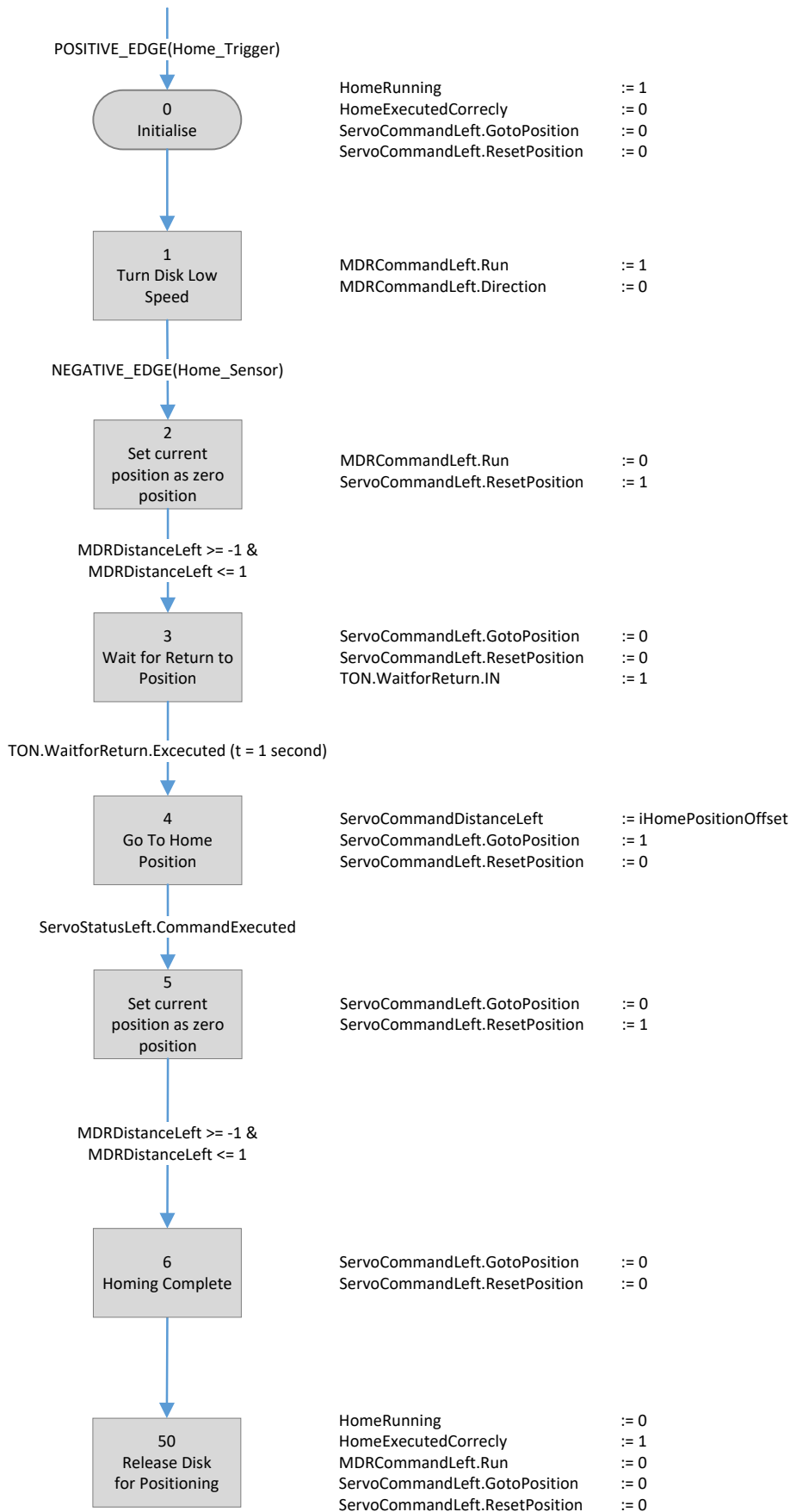
The controller doesn't register this movement so it is mandatory to do a homing sequence for the disk to correct its position. In the flowchart shown below the homing sequence of the disks is described. This is the homing sequence which is implemented in the example function block.

Due to the play in the gear box and the mechanical movements, the discs could not be aligned perfectly to the desired 0° position. The angle can be adjusted by setting a small offset in degrees to the homing sequence. This will be showed in the example below and it is written in the function blocks as well.

The homing sequence should be executed multiple times during operation to be sure that the discs are aligned properly during the sortation. Ignoring the homing sequence during the operation hours is on the customers on responsibility and could affect the succession of the amount of sortation. Flowsort recommends to execute the homing sequence after the following interval:

	Low speed (0 – 0.75m/s)	High speed (0.75 – 1.55m/s)
Mostly light products (0 – 10kg)	100 sortation movements	85 sortation movements
Mostly heavy products (10 – 25kg)	75 sortation movements	50 sortation movements

Flowchart of the homing sequence:



EXAMPLE FUNCTION BLOCK

3.1 GENERAL INFORMATION

The Flowsort function block can be used to control the SLD/DLD/ZZ 24V diverters. In the following PLC programming packages, the Flowsort function block is available.

- Siemens TIA V15.1 or higher with the ConveyLogix-Ai2 Controller
- Siemens Simatic Manager STEP 7 V5.6 + SP2 AND SIMATIC-SCL Professional 2017 V5.6 with the ConveyLogix-Ai2 Controller
- Allen Bradley Studio 5000 V32 or higher with the ConveyLogix-Ai2 Controller
- Beckhoff TC v3.1.4024.0 or higher with the Motionlinx-Ai Controller or with the ConveyLogix-Ai2 Controller
- Omron Sysmac Studio 1.40.0 or higher with the ConveyLogix-Ai2 Controller
- Mitsubishi GX Works 2 or higher with the ConveyLogix-Ai2 Controller

This document is designed to show how it is possible to control the diverter by different systems. The function block can be used in a multi-instance structure.

It is also possible to use the controller on other brands PLC's. When structured text programming language is available in the programming package, the code from the example function block can be copied and used to the project. However it is mandatory for the PLC to have one of the following network protocols available on the controller:

- ProfiNET
- Modbus/TCP
- Ethernet I/P
- EtherCAT

3.2 GENERAL FUNCTIONING

The function block is an example of the basic usage of the diverter. With the function block it is possible to send the diverter to 3 custom positions. Also a homing functionality is implemented to show how to make a reference travel to make sure the disks always start at the correct middle position.

The calculations to the desired angle from degrees to pulses are also implemented in the example function block. The functionality of the function block will be explained by the inputs and outputs of the function block.

!! Two different versions of the function block exist in the example software for the different PLC brands, The Version 3.x Function block has a better repeatable homing position due to firmware changes in the Conveylinx controller. The Version 3.x Function block however can only be used with Firmware version 6.06.0 and Higher (not 6.64.0). With lower firmware versions on the Conveylinx controller the homing function doesn't work anymore, so first upgrade the firmware of the Conveylinx controller before implementing the Version 3.x Function block!!

3.2.1 INPUT VARIABLES

Input	Datatype	Description	Additional Information
bEnable	Bool	Enable the module roller MDR start rolling and disc function Enable	Enable signal to enable the module
bHoming_Trigger	Bool	The Homing cycle is activated by a positive edge at this input	Only rising edge of signal is evaluated
bSortPos_Execute	Bool	Execute go to selected position	Only rising edge of signal is evaluated, Disks should be homed first!
bRoller_Execute	Bool	Start Rollers	Start Rollers
iGearRatio_Disk	Int	Gear ratio of the PGD Motor pulse rotation disk	Standard modules are supplied with engine type 15. Only adjust if another motor type is being installed.
iMechanicRatio_Disk	Int	Mechanical ratio of the of the system rotation disk	Mechanical gear ratio. Is fixed by the manufacturer. (1:2)
iGearRatio_Roller	Int	Gear ratio of the PGD Motor Pulse Roller	Standard modules are supplied with engine type 11. Only adjust if another motor type is being installed.
iMechanicRatio_Roller	Int	Mechanical ratio of the system roller	Mechanical gear ratio. Is fixed by the manufacturer. (1:1)
iRotationSpeed	Int	Rotational speed sort discs 0 - 100%	Calculated value in relation to the installed MDR type. Adjustable between 0 (min. Speed, still turning) and 100% (max. Speed).
iRollerSpeed	Int	Rotational speed rollers 0 - 100%	Calculated value in relation to the installed MDR type. Adjustable between 0 (min. Speed, still turning) and 100% (max. Speed).
iSortPos_MID	Int	Sorting Position Middle	Angle in degrees for the middle position of the disks
iSortPos_LEFT	Int	Sorting Position Left	Angle in degrees for the left sorting position of the disks
iSortPos_RIGHT	Int	Sorting Position Right	Angle in degrees for the right sorting position of the disks
iHomePositionOffset	Int	Offset Home Sensor to zero position	Angle in degrees for the offset of the home position to the zero position of the disks
iSortPos_Location	Int	Disc position to sort (0 = MID, 1 = LEFT, 2 = RIGHT)	Next sort position. Value 0 = Middle position, 1 Left Position, 2 Right Position.

3.2.1.1. Enable

The enable signal must always be enabled before any movement is possible. When the enable is **True** the release is given for the motors to move. When the enable is **False** all movements are stopped immediately and also the homing program will be aborted.

3.2.1.2. Homing Trigger

The Homing trigger will start the homing sequence described in chapter 2.5. The homing sequence will only be started on the rising edge of this signal. When the Homing trigger stays **True**, the homing sequence will not be started a second time. For the homing sequence to start the Enable must be **True**!

3.2.1.3. SortPos Execute

The Sort Position Execute signal will start the disk to go to the requested position. The Sort Position Execute will only be started on the rising edge of this signal. When the Sort Position Execute trigger stays **True**, the homing sequence will not be started a second time. For the Sortpos Execute to start the Enable must be **True** and the Disks must be Homed first!

3.2.1.4. Roller Execute

The Roller Execute signal will start the rollers to start running. When the Roller Execute signal changes to **False**, the rollers will stop running. . For the rollers to start running the Enable must be **True**!

3.2.1.5. GearRatio Disk

The gear ratio from the disks is a given value with the configuration Flowsort has delivered. It can be found in the documentation. This value will be used to convert the angle in degrees to a value in pulses which will be send to the controller.

3.2.1.6. MechanicalRatio Disk

The mechanical ratio from the disks is a given value with the configuration Flowsort has delivered. It can be found in the documentation. This value will be used to convert the angle in degrees to a value in pulses which will be send to the controller.

3.2.1.7. GearRatio Roller

The gear ratio from the Rollers is a given value with the configuration Flowsort has delivered. It can be found in the documentation. This value will be used to convert the angle in degrees to a value in pulses which will be send to the controller.

3.2.1.8. MechanicalRatio Roller

The mechanical ratio from the Rollers is a given value with the configuration Flowsort has delivered. It can be found in the documentation. This value will be used to convert the angle in degrees to a value in pulses which will be send to the controller.

3.2.1.9. Rotation Speed

The rotation speed is the rotating speed of the disks. The input value must be between 0% and 100%. When entering 0% the lowest possible speed for the motor will be send to the controller. When entering 100% the highest possible speed for the motor will be send to the controller. When the homing sequence is carried out the function block will automatically send the lowest possible speed to the controller.

3.2.1.10. Roller Speed

The roller speed is the speed of the rollers. The input value must be between 0% and 100%. When entering 0% the lowest possible speed for the motor will be send to the controller. When entering 100% the highest possible speed for the motor will be send to the controller.

3.2.1.11. Sortpos MID

The middle sorting position is basically the normal position of the disks. In most cases this will be the position of the disks when packages are going straight forward. The position can be given in degrees. The position is calculated according to the home position.

3.2.1.12. Sortpos LEFT

The Left sorting position is to sort out a package to the left track. The position can be given in degrees. The position is calculated according to the home position.

3.2.1.13. Sortpos RIGHT

The Right sorting position is to sort out a package to the right track. The position can be given in degrees. The position is calculated according to the home position.

3.2.1.14. Home Position Offset

The Home position offset is an offset that will be added to the home position. Normally this value will be 0, but when it's not possible to get the home position exactly in the middle this offset can be used to get the home position exactly in the middle.

3.2.1.15. Sortpos Location

The sort position location is the location to which the disks should be send to. There are three possibilities. The middle position is 0, the left position is 1 and the right position is 2.

For the discs to execute the position change a rising edge on the SortPos Execute is necessary!

3.2.2 OUTPUT VARIABLES

Output	Datatype	Description	Additional Information
bSTAT_Ready_Roller	Bool	State Rollers Ready to Start	Status of Rollers Ready to Start
bSTAT_Ready_Disk	Bool	State Disk Ready for Positioning	Status of Disk Ready for Positioning
bSTAT_CommOK	Bool	State Communication to Module OK	Status Communication to Conveylinxmodule OK
bSTAT_HomingOK	Bool	State Homing of Disk executed Correctly	Status Homing of Disks executed Correctly
bSTAT_InPosition_Disk	Bool	State Disk is in Position	Status Disks in Requested Position
wSTAT_DiagnosticDisk	Word	Diagnostic Status of Disk motor	Diagnostic Status of Disk motor. More info on controller specifically.
wSTAT_DiagnosticRoller	Word	Diagnostic Status of Roller motor	Diagnostic Status of Disk motor. More info on controller specifically.

3.2.2.1. Ready Roller

The Ready Roller signal is a status signal for the rollers. The status signal is **True** when there are no motor faults on the roller motor, the communication with the controller is OK and when the Enable signal (input) is **True**. When one of these conditions is not met, the signal will be **false**.

3.2.2.2. Ready Disk

The Ready Disk signal is a status signal for the disks. The status signal is **True** when there are no motor faults on the disk motor, the communication with the controller is OK, the homing sequence is executed correctly and when the Enable signal (input) is **True**. When one of these conditions is not met, the ready signal will be **false**.

3.2.2.3. Communication OK

The communication OK signal is **True** when the heartbeat received from the controller keeps on toggling. When the heartbeat signal not changes status in 5 seconds then a communication fault occurred and the signal changes to **False**.

3.2.2.4. Homing OK

The homing OK signal is the signal which shows the status the homing sequence was executed correctly. The homing OK signal is always **false** after a restart or new power-up of the PLC. Also when a new homing sequence is started the signal will return to **False**. When the homing sequence is executed correctly the signal will change to **True**. When the homing sequence is aborted in the middle of the sequence the signal will stay **false**.

3.2.2.5. In Position Disk

The signal In Position Disk shows the status of the positioning action of the disk. When a positioning of the disks is started (rising edge on the SortPos Execute), the In Position signal changes to **False** and the positioning will start. When the controller gives a ready signal that the action was carried out, the In Position signal changes to **True**.

3.2.2.6. Diagnostic Disk

The Diagnostic disk status is a combination variable which shows the status of the disk motor. This diagnostic can be used to check a motor error. Each bit in the Word describes a diagnostic status of the Disk:

Bit	Description
Bit 0	Motor Status (see below)
Bit 1	Motor Status (see below)
Bit 2	Port in Digital Mode
Bit 3	Reserved
Bit 4	Reserved
Bit 5	Reserved
Bit 6	Over-Voltage
Bit 7	Low Voltage
Bit 8	Overheated
Bit 9	At Maximum Torque
Bit 10	Short Circuit
Bit 11	Motor Not Connected
Bit 12	Overloaded
Bit 13	Motor Stalled
Bit 14	Hall Sensor Error
Bit 15	Motor Not Used

Motor Status:

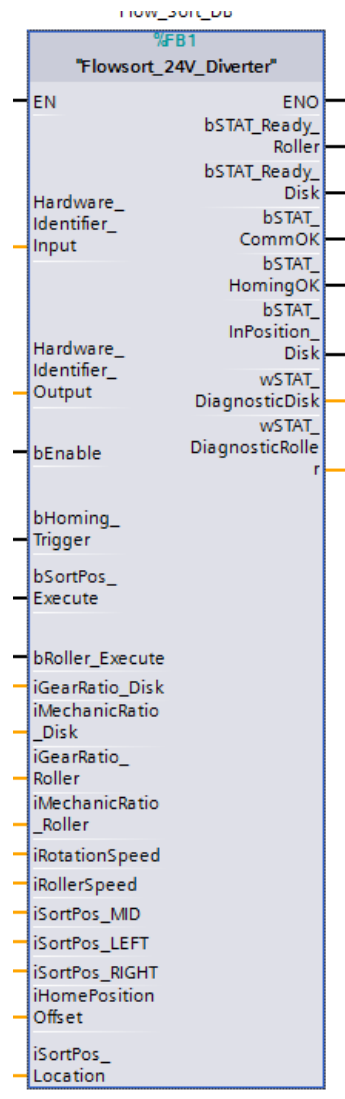
Bit 1	Bit 0	Description
0	0	Motor not Running, Standard or servo braking Applied
0	1	Motor Running in CCW Direction
1	0	Motor Running in CW Direction
1	1	Motor not Running and no braking applied (free to spin)

3.2.2.7. Diagnostic Roller

The Diagnostic roller status is a combination variable which shows the status of the roller motor. This diagnostic can be used to check a motor error. The description of the signals can be in paragraph 3.2.2.6.

3.3 PROGRAM INTEGRATION FOR SIEMENS TIA PORTAL V15.1 OR HIGHER (PROFINET)

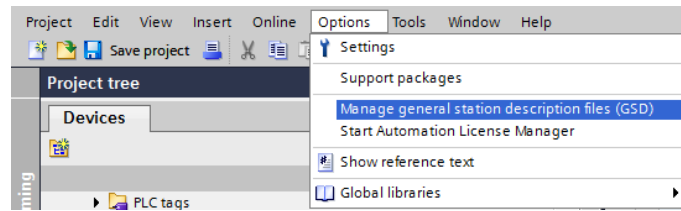
For the use of the Flowsort 24V Diverter with a Siemens Controller (CPU1200 – CPU1500) via ProfiNET, follow the steps described below.



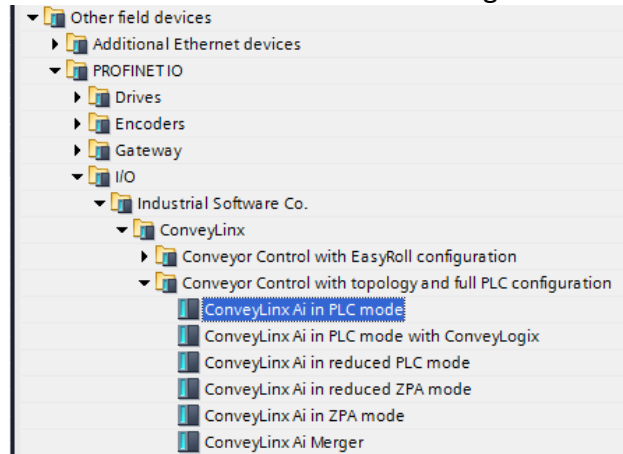
3.3.1 HARDWARE

To integrate the Flowsort 24V Diverter in a Siemens 1200 or 1500 Controller, first the GSDML file of the controller must be downloaded. This can be done on the Pulserollers website: <https://pulseroller.com/downloads/>. Go to Software And Firmware downloads -> PLC Connectivity -> PLC – ProfiNET GSDML files -> Conveylinx AI2 -> download the file called: GSDML.

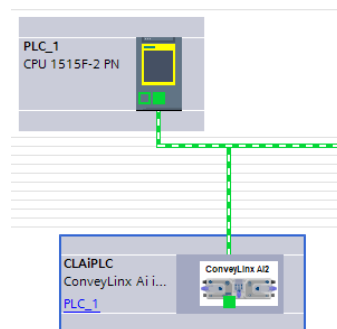
First unzip the GSDML file, then import the GSDM file into the project. This can be done in the options menu of Tia portal under Manage General station description files.



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.

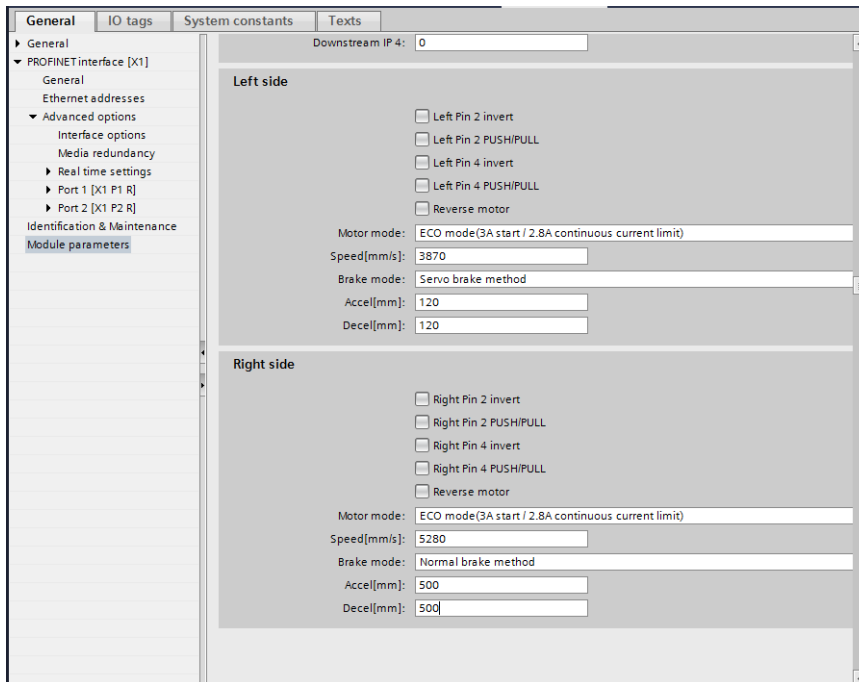


Drag and drop the ConveyLinX Ai in PLC mode into the network. Assign the Conveylinx object to the controller.



Now the system integrator can give the module an ProfiNET name and IP-adress.

The following Module parameters must be set in the properties of the Conveylinx-AI.

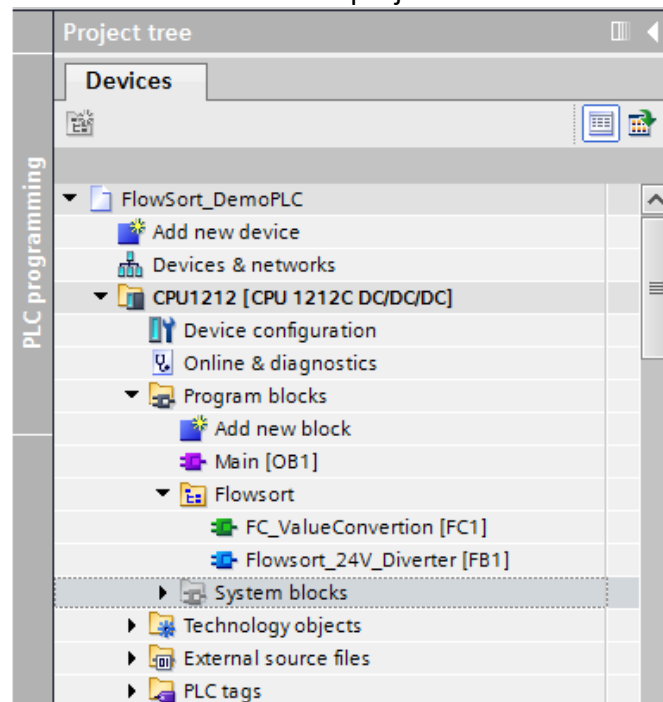


The speed and acceleration can be adapted to the application of the diverters. These settings are the settings that the controller uses after a communication fault.

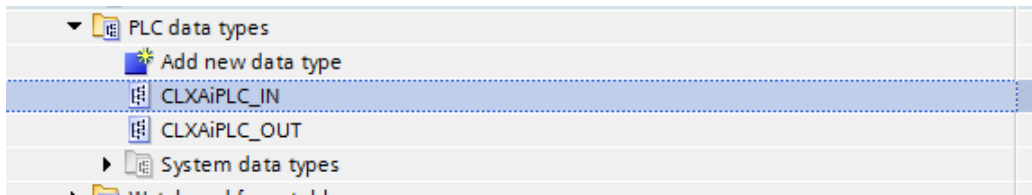
After these settings are set the configuration can be downloaded to the controller. When the configuration is in the controller, the ProfiNET name needs to be assigned to the device before operation is possible.

3.3.2 SOFTWARE

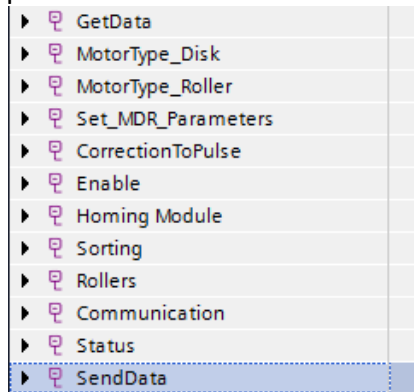
In the Demo program, the following blocks are available. For the integration of the Flowsort function block, copy the Flowsort folder to the project.



Also copy the datatypes in the Demo program:



The Flowsort function block is freely accessible for the integrator to adapt. The example function block is divided in 12 parts.



The **Get Data** parts reads out the data from the ProfiNET Device with the given Hardware Identifier

The **MotorType_Disk** part scales the speed values from the disk from 0 – 100% to pulses to send to the controller

The **MotorType_Roller** part scales the speed values from the rollers from 0 – 100% to pulses to send to the controller

The **set_MDR_Parameters** part sets some default settings to the controller

The **CorrectionToPulse** part recalculates the angles on the input of the functionblock to the value in pulses to send to the controller

The **Enable** part stops all the movements of the motors when the enable is not “TRUE”.

The **Homing Module** part is a step program which makes the disks do a reference travel and set a new zero position.

The **Sorting** part is the part where the positioning of the disks is handled.

The **Rollers** part is where the starting and stopping of the rollers is handled.

The **Communication** part checks if the communication with the Conveylinx module is still online.

The **Status** part checks the status of the disk and the rollers and outputs them on the functionblock.

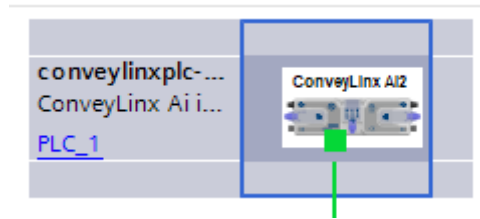
The **Send Data** parts gets the data from the function block and sends it to the ProfiNET Device with the given Hardware Identifier

For the exact functioning of the function block, the example code is supported with comments on what certain parts of the code do. The code is just an example on how to start with the Flowsort diverter and is freely accessible for the integrator to adjust.

3.3.3 CONTROLLER LINK SOFTWARE TO HARDWARE

The link from the software to the hardware can be done by linking the variable of the hardware identifier or by linking the constant number of the hardware identifier to the

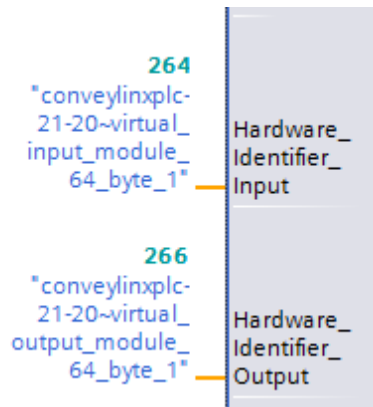
function block. The hardware identifiers of the module can be found under the “Devices & Network” on the device itself.



In the Properties window open the “System constants”. Here the Hardware identifiers of the input and output can be found.

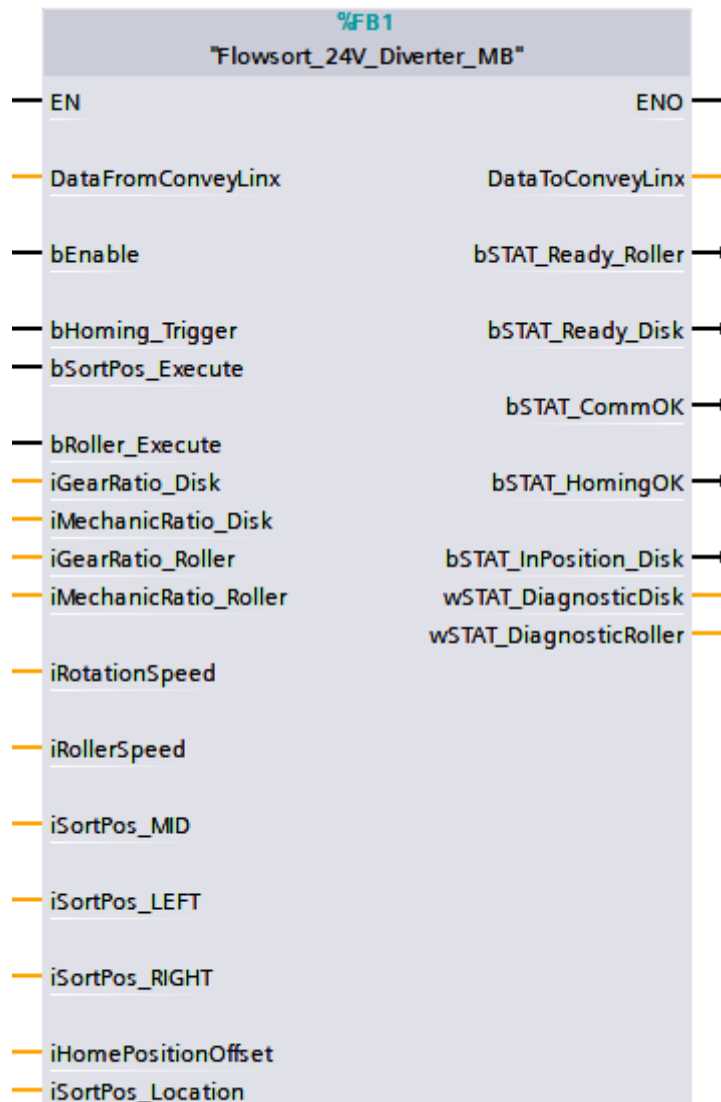
conveylinxplc-21-20 [ConveyLinx Ai in PLC mode]					
General		IO tags	System constants	Texts	
Show hardware system constant ▾					
Name	Type	Hardware identi.	Used by	Comment	
conveylinxplc-21-20-Interface-Port_1	Hw_Interface	260	PLC_1		
conveylinxplc-21-20-Interface-Port_2	Hw_Interface	261	PLC_1		
conveylinxplc-21-20-Interface	Hw_Interface	259	PLC_1		
conveylinxplc-21-20-Proxy	Hw_SubModule	258	PLC_1		
conveylinxplc-21-20-Head	Hw_SubModule	262	PLC_1		
conveylinxplc-21-20-virtual_input_module_64_byte_1	Hw_SubModule	264	PLC_1		
conveylinxplc-21-20-virtual_output_module_64_byte_1	Hw_SubModule	266	PLC_1		

The conv(.....)~virtual_input_module_64_byte_1 must be linked to the “Hardware identifier Input” and the conv(.....)~virtual_output_module_64_byte_1 must be linked to the “Hardware identifier Output”. The linking can be done by selecting the tag and dragging it to the input of the function block you want to link this module to. It can also be done by just typing the number which is in the hardware identifier column at the input of the function block.



3.4 PROGRAM INTEGRATION FOR SIEMENS TIA PORTAL V15.1 OR HIGHER (MODBUS)

For the use of the Flowsort 24V Diverter with a Siemens Controller (CPU1200 – CPU1500) via Modbus, follow the steps described below.

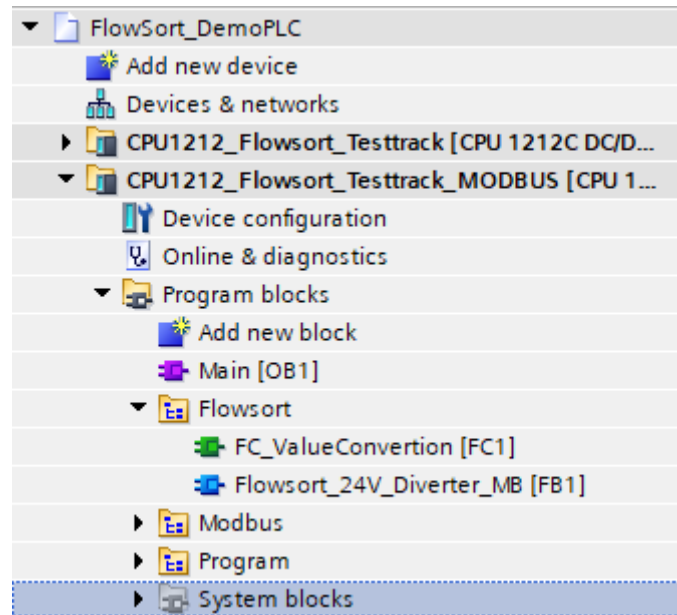


3.4.1 HARDWARE

To integrate the Flowsort 24V Diverter in a Siemens 1200 or 1500 Controller via modbus, no hardware has to be configured. The only thing that should be configured is that the IP-adress and subnetmask that is configured in the PLC should be in range of the IP-adresses of the conveylinx modules.

3.4.2 SOFTWARE

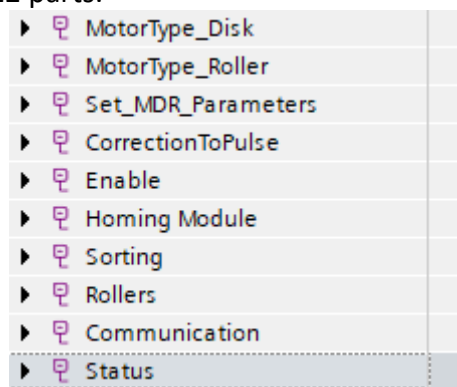
In the Demo program, the following blocks are available. For the integration of the Flowsort function block, copy the Flowsort folder to the project.



Also copy the datatypes in the Demo program:



The Flowsort function block is freely accessible for the integrator to adapt. The example function block is divided in 12 parts.



The **MotorType_Disk** part scales the speed values from the disk from 0 – 100% to pulses to send to the controller

The **MotorType_Roller** part scales the speed values from the rollers from 0 – 100% to pulses to send to the controller

The **set_MDR_Parameters** part sets some default settings to the controller

The **CorrectionToPulse** part recalculates the angles on the input of the functionblock to the value in pulses to send to the controller

The **Enable** part stops all the movements of the motors when the enable is not “TRUE”.

The **Homing Module** part is a step program which makes the disks do a reference travel and set a new zero position.

The **Sorting** part is the part where the positioning of the disks is handled.

The **Rollers** part is where the starting and stopping of the rollers is handled.

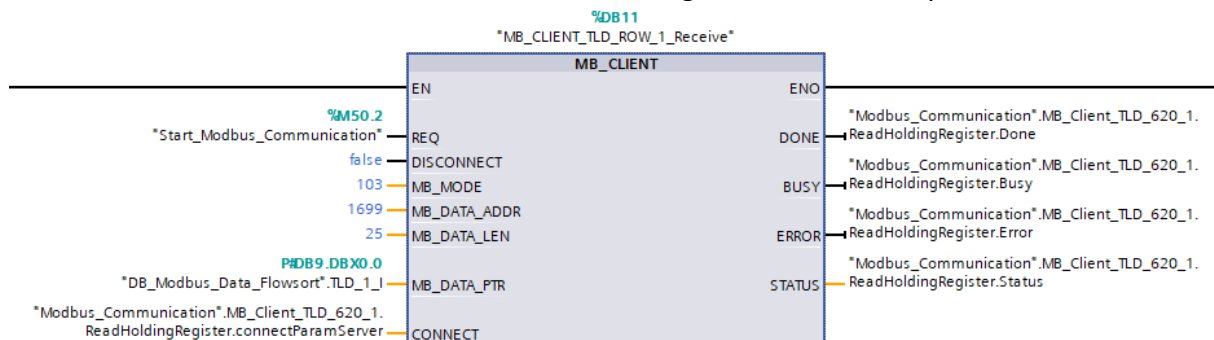
The **Communication** part checks if the communication with the Conveylinx module is still online.

The **Status** part checks the status of the disk and the rollers and outputs them on the functionblock.

For the exact functioning of the function block, the example code is supported with comments on what certain parts of the code do. The code is just an example on how to start with the Flowsort diverter and is freely accessible for the integrator to adjust.

3.4.3 CONTROLLER LINK SOFTWARE TO HARDWARE RECEIVE DATA

The link from the software to the hardware can be done by using Modbus communication function blocks. These functionblocks must be configured with certain parameters.



For receiving modbus data, function block **MB_CLIENT** is used. To get the right data from the conveylinx module the inputs:

- **MB_MODE** must be set to **103**
- **MB_DATA_ADDR** must be set to **1699**
- **MB_DATA_LEN** must be set to **25**

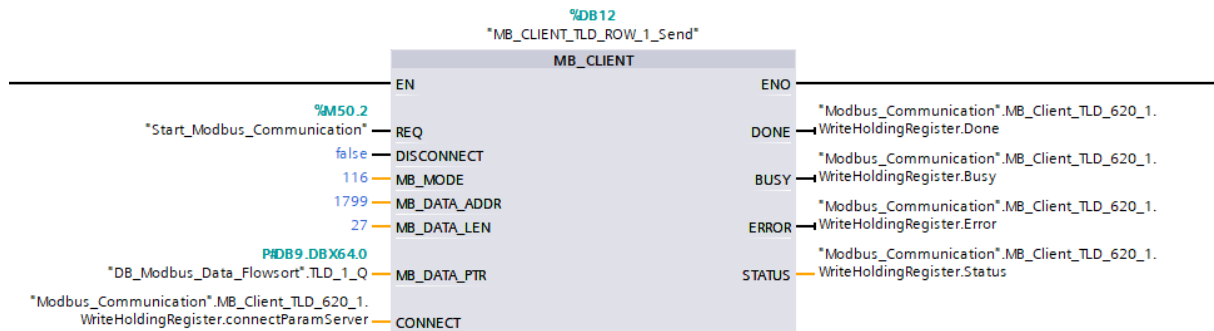
The data **MB_DATA_PTR** is the data that is received from the modbus client. In the example an data area is assigned to it with datatype "**CLXAiPLC_IN**". This is exactly the correct size. The other important input of the functionblock is the **CONNECT** input. This should be connected with a datatype "**TCON_IP_v4**" and should be configured in this way:

connectParamServer	TCON_IP_v4			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
InterfaceId	HW_ANY	64		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	HW-identifier of IE-interface submodule
ID	CONN_OUC	16#0002		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	connection reference / identifier
ConnectionType	Byte	16#0B		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	type of connection: 11=TCP/IP, 19=UDP (17=TCP/IP)
ActiveEstablished	Bool	TRUE		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	active/passive connection establishment
RemoteAddress	IP_V4			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	remote IP address (IPv4)
ADDR	Array[1..4] of Byte			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	IPv4 address
ADDR[1]	Byte	16#C0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	IPv4 address
ADDR[2]	Byte	16#A8		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	IPv4 address
ADDR[3]	Byte	16#15		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	IPv4 address
ADDR[4]	Byte	16#B6		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	IPv4 address
RemotePort	UInt	502		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	remote UDP/TCP port number
LocalPort	UInt	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	local UDP/TCP port number

The **interfaceID** is the hardware identifier of the IE-Interface of the configured PLC. The **ID** should be an unique ID for every connection. The **ConnectionType** should be 11 (HEX: 0B). The **ActiveEstablished** should be TRUE. In the **ADDR** the IP-Adress should be set in this example: 192.168.21.182 (HEX: C0,A8,15,B6). The **RemotePort** should be 502 and the **LocalPort** should be 0.

3.4.4 CONTROLLER LINK SOFTWARE TO HARDWARE SEND DATA

For sending data via modbus the functionblocks must be configured with certain parameters.



For sending modbus data, function block **MB_CLIENT** is used. To send the data to the right place in the conveylinx module the inputs:

- **MB_MODE** must be set to **116**
- **MB_DATA_ADDR** must be set to **1799**
- **MB_DATA_LEN** must be set to **27**

The data **MB_DATA_PTR** is the data that is send to the modbus client. In the example an data area is assigned to it with datatype "**CLXAIPLC_OUT**". This is exactly the correct size. The other important input of the functionblock is the **CONNECT** input. This should be connected with a datatype "TCON_IP_v4" and should be configured in this way:

connectParamServer	TCON_IP_v4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
InterfaceId	HW_ANY	64	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	HW-identifier of IE-interface submodule
ID	CONN_OUC	16#0003	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	connection reference / identifier
ConnectionType	Byte	16#0B	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	type of connection: 11=TCP/IP, 19=UDP (17=TCP/IP)
ActiveEstablished	Bool	TRUE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	active/passive connection establishment
RemoteAddress	IP_V4		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	remote IP address (IPv4)
ADDR	Array[1..4] of Byte		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	IPv4 address
ADDR[1]	Byte	16#C0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	IPv4 address
ADDR[2]	Byte	16#A8	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	IPv4 address
ADDR[3]	Byte	16#15	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	IPv4 address
ADDR[4]	Byte	16#B6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	IPv4 address
RemotePort	UInt	502	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	remote UDP/TCP port number
LocalPort	UInt	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	local UDP/TCP port number

The **interfaceID** is the hardware identifier of the IE-Interface of the configured PLC. The **ID** should be an unique ID for every connection. The **ConnectionType** should be 11 (HEX: 0B). The **ActiveEstablished** should be TRUE. In the **ADDR** the IP-Adress should be set in this example: 192.168.21.182 (HEX: C0,A8,15,B6). The **RemotePort** should be 502 and the **LocalPort** should be 0.

Make sure when using the **MB_CLIENT** Function block multiple times that each call has an unique instance and also every call has a unique ID!

For more information on the parameters for the modbus communication use the Help function on the **MB_CLIENT** functionblock.

3.5 PROGRAM INTEGRATION FOR SIEMENS SIMATIC MANAGER V5.6 OR HIGHER (PROFINET)

For the use of the Flowsort 24V Diverter with a Siemens Controller (S7-300/S7-400) via ProfiNET, follow the steps described below.

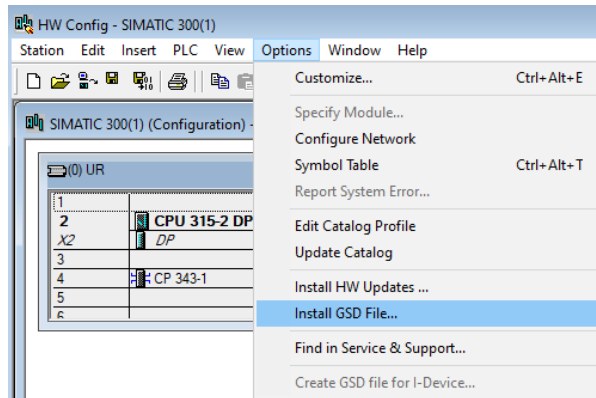
DB10	
"iDE_ Flowsort_ SLD1"	
FB1	
"Flowsort_24V_ Diverter"	
EN	ENO
LADDR_ Input	bSTAT_ Ready_ Roller
LADDR_ Output	bSTAT_ Ready_ Disk
bEnable	bSTAT_ CommOK
bHoming_ Trigger	bSTAT_ HomingOK
bSortPos_ Execute	bSTAT_ InPositio n_Disk
bRoller_ Execute	wSTAT_ Diagnosti cDisk
iGearRati o_Disk	wSTAT_ Diagnosti cRoller
iMechanic Ratio_ Disk	
iGearRati o_Roller	
iMechanic Ratio_ Roller	
iRotation Speed	
iRollerSp eed	
iSortPos_ MID	
iSortPos_ LEFT	
iSortPos_ RIGHT	
iHomePosi tionOffse t	
iSortPos_ Location	

3.5.1 HARDWARE

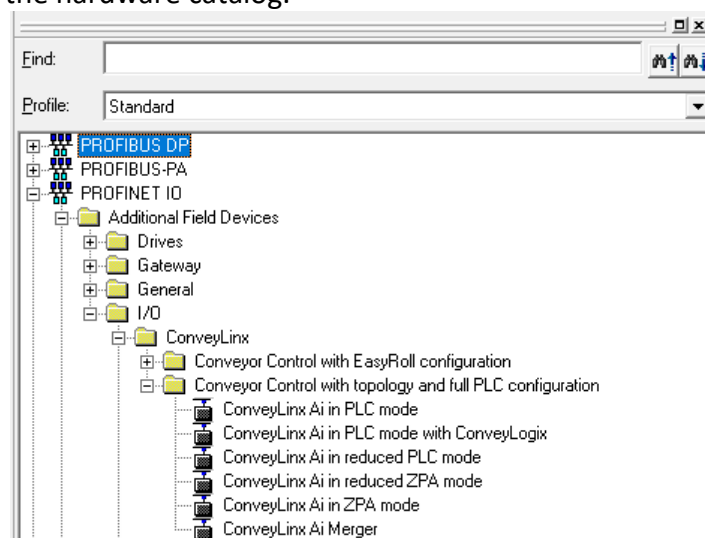
To integrate the Flowsort 24V Diverter in a Siemens S7-300 or S7400, first the GSDML file of the controller must be downloaded. This can be done on the Pulseroller website:

<https://pulseroller.com/downloads/>. Go to Software And Firmware downloads -> PLC Connectivity -> PLC – ProfiNET GSDML files -> Conveylinx Ai2 -> download the file called: GSDML.

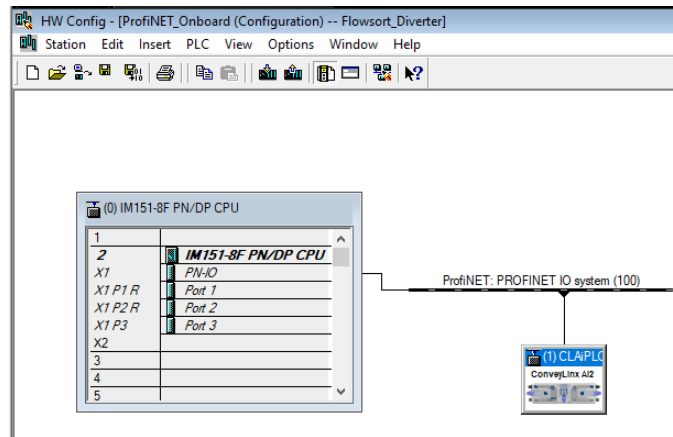
First unzip the GSDML file, then import the GSDM file into the project. This can be done in the options menu of the Hardware configuration in Simatic Manager:



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 the hardware catalog.

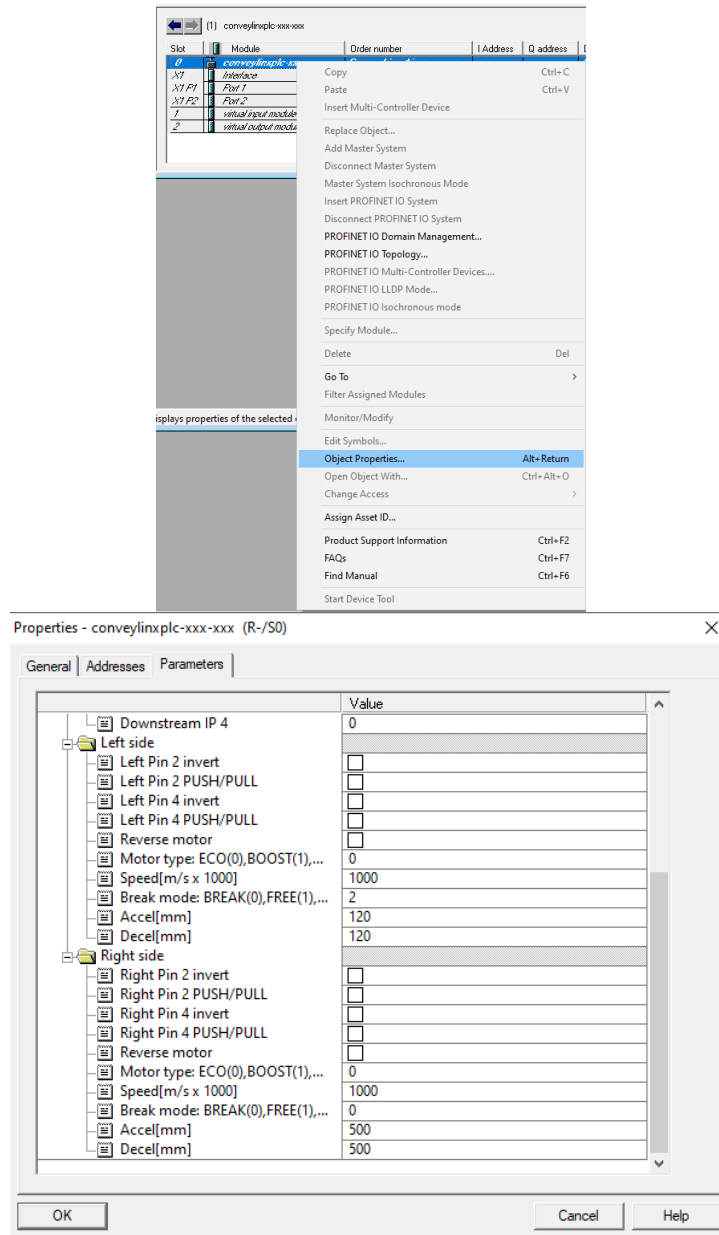


Drag and drop the **ConveyLinx Ai in PLC mode** into the network and assign the Conveylinx object to the controller.



Now the system integrator can give the module an ProfiNET name and IP-adress.

The following Module parameters must be set in the properties of the Conveylinx-AI. Right click on Slot 0 of the module and go to the Parameters tab.



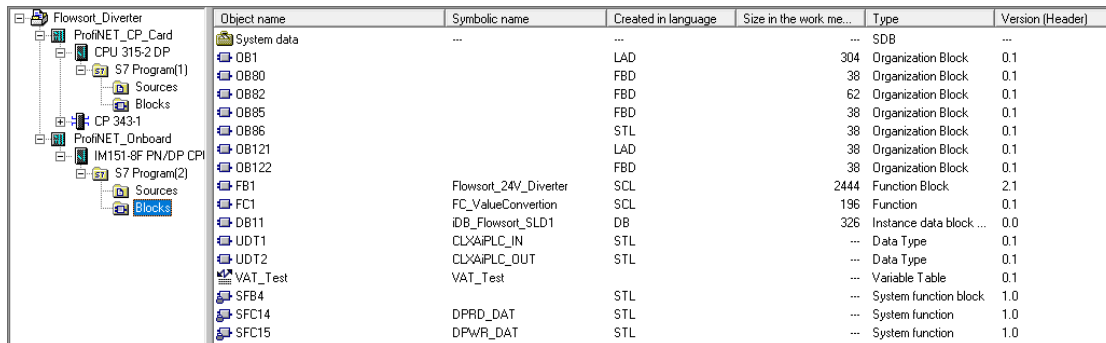
The speeds and ramps (Accel/Decel) are send in the program so it is not necessary to set the speed in the configuration of the controller. The values used here are only used after a communication failure as “initial values”.

After these settings are set the configuration can be downloaded to the controller. When the configuration is in de controller, the ProfiNET name needs to be assigned to the device before operation is possible.

3.5.2 SOFTWARE (PROFINET ONBOARD)

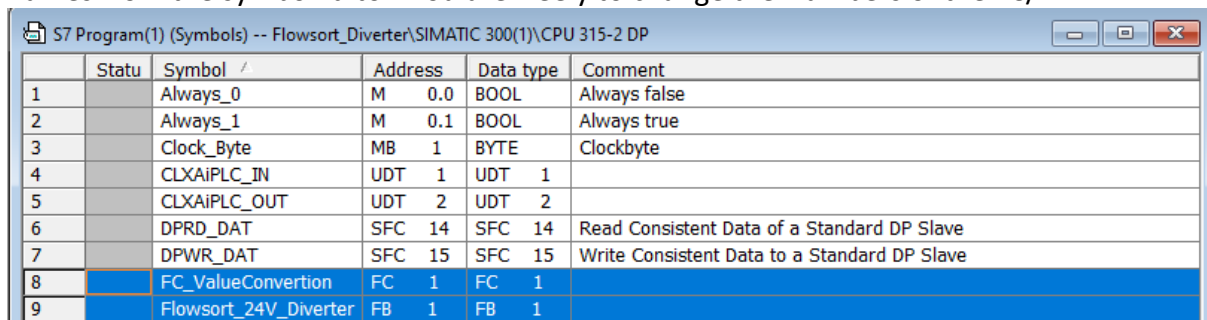
In the Demo program, there are two PLC's available. If a PLC with a ProfiNET interface onboard is used (where the diverters are connected to), use the blocks/UDT's/Sources in the **ProfiNET_Onboard** Project. For the integration of the Flowsort function block copy first the UDT's:

- **UDT1: CLXAIPLC_IN**
- **UDT2: CLXAIPLC_OUT**



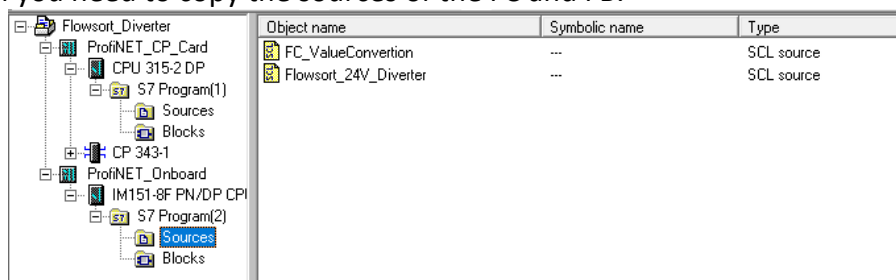
Object name	Symbolic name	Created in language	Size in the work me...	Type	Version (Header)
System data	---	---	---	SDB	---
OB1	---	LAD	304	Organization Block	0.1
OB80	---	FBD	38	Organization Block	0.1
OB82	---	FBD	62	Organization Block	0.1
OB85	---	FBD	38	Organization Block	0.1
OB86	---	STL	38	Organization Block	0.1
OB121	---	LAD	38	Organization Block	0.1
OB122	---	FBD	38	Organization Block	0.1
FB1	Flowsort_24V_Diverter	SCL	2444	Function Block	2.1
FC1	FC_ValueConversion	SCL	196	Function	0.1
IDB_Flowsort_SLD1	---	DB	326	Instance data block ...	0.0
UDT1	CLXAIPLC_IN	STL	---	Data Type	0.1
UDT2	CLXAIPLC_OUT	STL	---	Data Type	0.1
VAT_Test	VAT_Test	---	---	Variable Table	0.1
SFB4	---	STL	---	System function block	1.0
SFC14	DPRD_DAT	STL	---	System function	1.0
SFC15	DPWR_DAT	STL	---	System function	1.0

After this want to copy the FC and FB. Because of the SCL language we first need symbol names from the symbolEditor. You are freely to change the Numbers of the FC/FB.



Statu	Symbol /	Address	Data type	Comment
1	Always_0	M 0.0	BOOL	Always false
2	Always_1	M 0.1	BOOL	Always true
3	Clock_Byte	MB 1	BYTE	Clockbyte
4	CLXAIPLC_IN	UDT 1	UDT 1	
5	CLXAIPLC_OUT	UDT 2	UDT 2	
6	DPRD_DAT	SFC 14	SFC 14	Read Consistent Data of a Standard DP Slave
7	DPWR_DAT	SFC 15	SFC 15	Write Consistent Data to a Standard DP Slave
8	FC_ValueConversion	FC 1	FC 1	
9	Flowsort_24V_Diverter	FB 1	FB 1	

On addition you need to copy the sources of the FC and FB.



Object name	Symbolic name	Type
FC_ValueConversion	---	SCL source
Flowsort_24V_Diverter	---	SCL source

After the copy you need to open and compile first the FC and then FB. When all this is finished without error's there is generated a FC and FB in the Blocks folder.

Note: If you copy the FC and FB from the example project from the Blocks instead of the Sources it's easy to monitor and modify the block. Still it should work.

The Flowsort function block is freely accessible for the integrator to adapt. The example SCL source function block is divided in 12 parts.

The **Get Data** part reads out the data from the ProfiNET Device with the given Hardware Identifier

The **MotorType_Disk** part scales the speed values from the disk from 0 – 100% to pulses to send to the controller

The **MotorType_Roller** part scales the speed values from the rollers from 0 – 100% to pulses to send to the controller

The **set_MDR_Parameters** part sets some default settings to the controller

The **CorrectionToPulse** part recalculates the angles on the input of the functionblock to the value in pulses to send to the controller

The **Enable** part stops all the movements of the motors when the enable is not “TRUE”.

The **Homing Module** part is a step program which makes the disks do a reference travel and set a new zero position.

The **Sorting** part is the part where the positioning of the disks is handled.

The **Rollers** part is where the starting and stopping of the rollers is handled.

The **Communication** part checks if the communication with the Conveylinx module is still online.

The **Status** part checks the status of the disk and the rollers and outputs them on the functionblock.

The **Send Data** part gets the data from the function block and sends it to the ProfiNET Device with the given Hardware Identifier

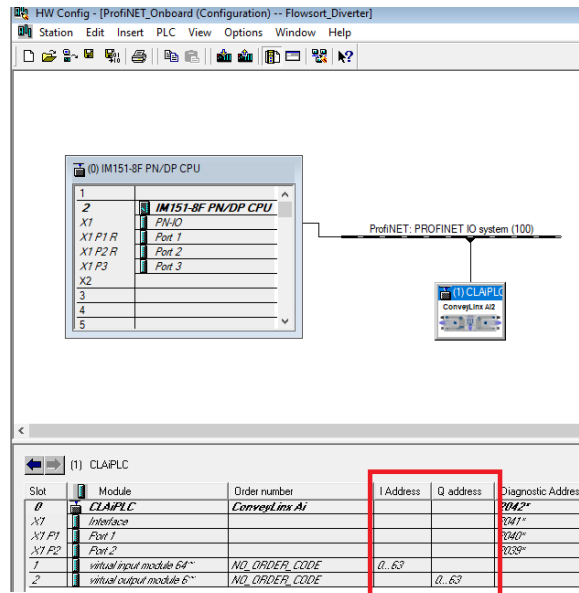
For the exact functioning of the function block, the example code is supported with comments on what certain parts of the code do. The code is just an example on how to start with the Flowsort diverter and is freely accessible for the integrator to adjust.

3.5.3 SOFTWARE (PROFINET CP-CARD)

In the Demo program, there are two PLC's available. If a PLC with a ProfiNET interface with a CP-card is used (where the diverters are connected to), use the blocks/UDT's/Sources in the **ProfiNET_CP_Card** Project. The rest of the steps are identical to the steps described in paragraph 3.4.2.

3.5.4 CONTROLLER LINK SOFTWARE TO HARDWARE (PROFINET ONBOARD)

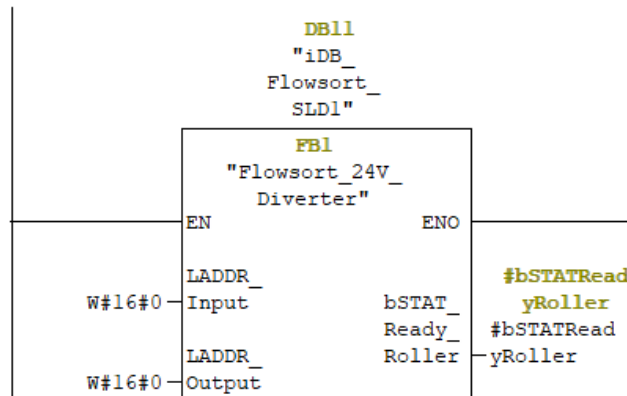
The link from the software to the hardware can be done the start address of input and output find in the hardware configuration.



In this example are the input and output start address 0. The "LADDR_Input" and the "LADDR_Output" needs to be linked with these start addresses in hexadecimal format. (0 Decimal = 0 HEX but 100 decimal = 64 HEX.)

Network 3: SLD1

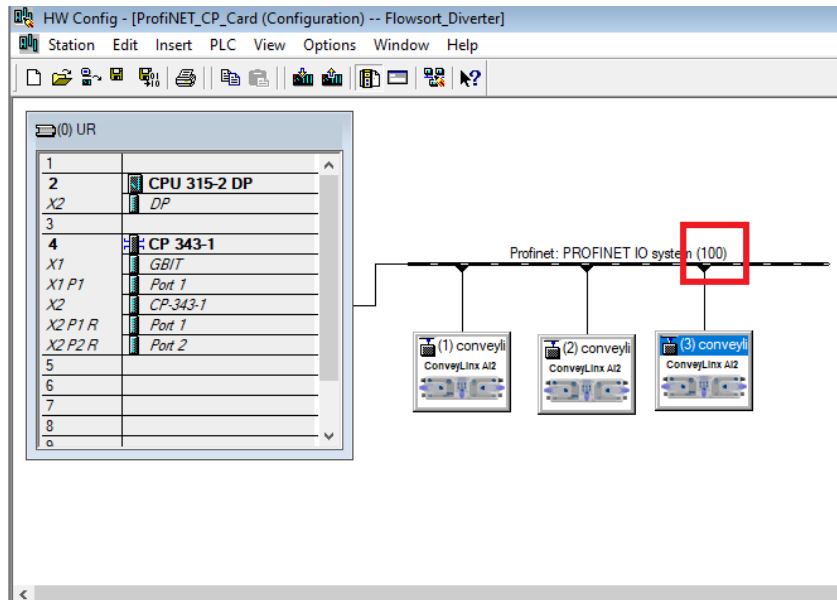
0 (DEC) = W#16#0 (HEX)



3.5.5 CONTROLLER LINK SOFTWARE TO HARDWARE (PROFINET CP-CARD)

When a CP-card is used the data from the devices must be read and write by the PNIO_RECV and PNIO_SEND blocks from Siemens. With this function block all data from the CP-card is read and write.

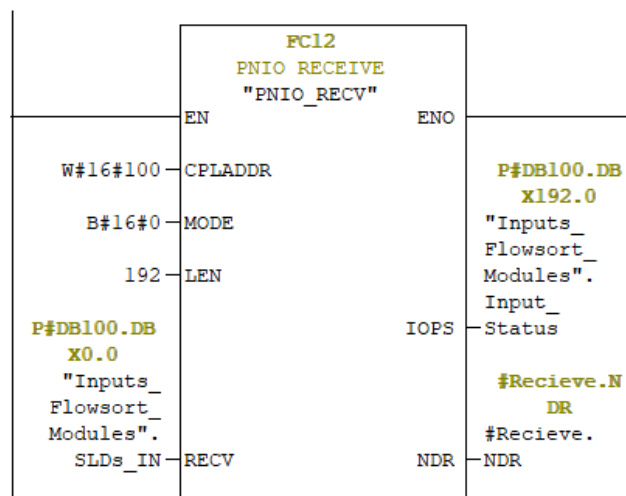
The Adress from the PNIO_Send/PNIO_Receive function blocks can be read out in the Hardware configuration:



In this example the CPLADDR is **W#16#100**. Use Mode **B#16#0**. The LEN parameter depends on the number of Modules needs to be read out. 1 diverter needs 64 bytes, in this example 3 diverters are configured so $3 \times 64 = 192$ bytes. The Data area for the data of the diverters is put at the RECV IO from the function block. This data area needs to be linked to the Flowsort Function block.

Network 3: Read data SLD's

256(DEC) = W#16#100 (HEX)



3.6 PROGRAM INTEGRATION FOR BECKHOFF TWINCAT V3.1.4024.0 OR HIGHER

The Flowsort 24V Diverter 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.

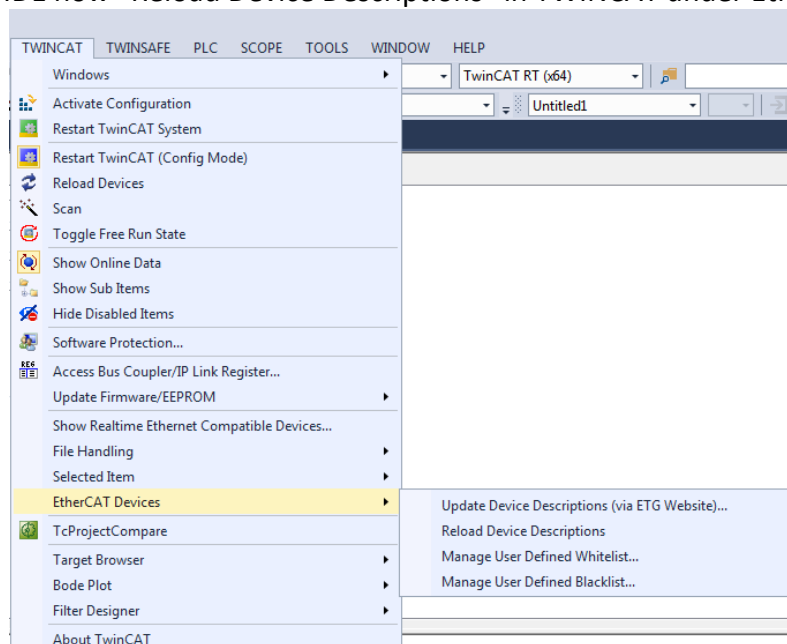
FlowSort_24V_Diverter_ProfiNET	
bEnable	bSTAT_Ready_Roller
bHoming_Trigger	bSTAT_Ready_Disk
bSortPos_Execute	bSTAT_CommOK
bRoller_Execute	bSTAT_HomingOK
iGearRatio_Disk	bSTAT_InPosition_Disk
iMechanicRatio_Disk	wSTAT_DiagnosticDisk
iGearRatio_Roller	wSTAT_DiagnosticRoller
iMechanicRatio_Roller	OMDR_Hardware
iRotationSpeed	
iRollerSpeed	
iSortPos_MID	
iSortPos_LEFT	
iSortPos_RIGHT	
iHomePositionOffset	
iSortPos_Location	
IMDR_Hardware	

3.6.1 MOTIONLINX-AI CONTROLLER HARDWARE

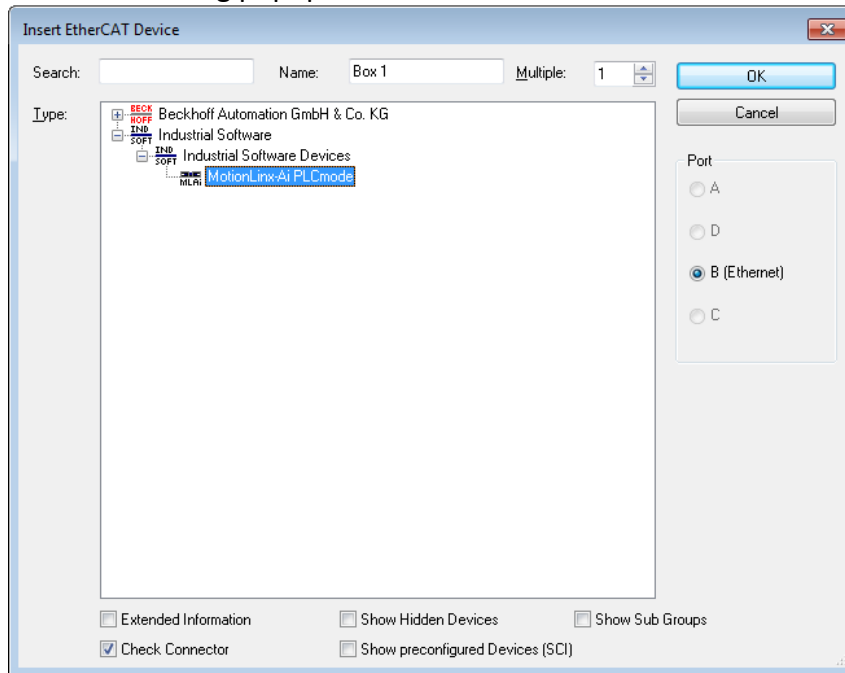
To integrate the Flowsort 24V Diverter in a Beckhoff Controller, first the ESI file of the controller must be downloaded. This can be done on the Pulserollers website: <https://pulseroller.com/downloads/>. Go to Software and Firmware Downloads -> Plc Connectivity-> Plc- Ethercat ESI Files -> MOTIONLINX AI and download the file called: ESI

Now the ESI file must be placed into the installation folder of TwinCat. Normally this should be: *C:\TwinCAT\3.1\Config\Io\EtherCAT* . But can be different when TwinCAT is installed in another folder. Extract the ESI file and copy the XML file into this folder.

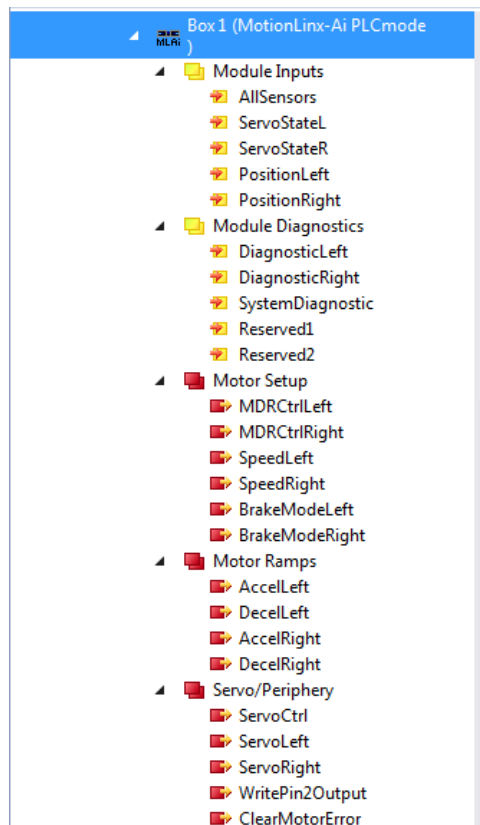
In the TwinCAT IDE now “Reload Device Descriptions” in TWINCAT under EtherCAT Devices.



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

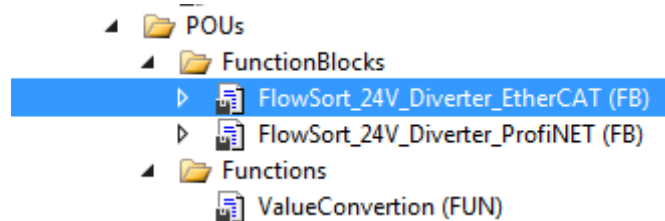


Select the MotionLinx-Ai PLCmode and click OK. Now the MotionLinx-Ai Controller is added to the project. In the Solution Explorer the MotionLinx-Ai Controller is available with all the data.

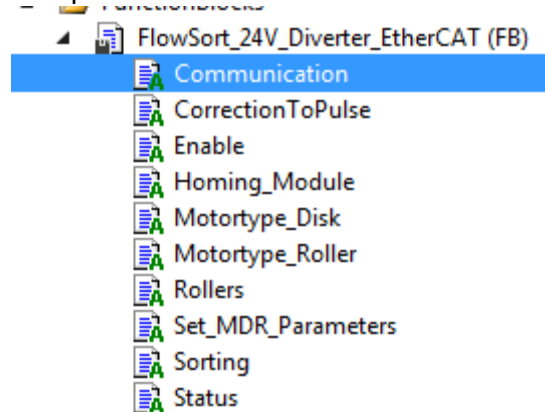


3.6.2 MOTIONLINUX-AI CONTROLLER SOFTWARE

In the Demo program, the following blocks are available. For the integration of the Flowsort function block, copy the FlowSort_24V_Diverter_EtherCAT and the ValueConversion into the project.



The Flowsort function block is freely accessible for the integrator to adapt. The example function block is divided in 12 parts.



The **MotorType_Disk** part scales the speed values from the disk from 0 – 100% to pulses to send to the controller

The **MotorType_Roller** part scales the speed values from the rollers from 0 – 100% to pulses to send to the controller

The **set_MDR_Parameters** part sets some default settings to the controller

The **CorrectionToPulse** part recalculates the angles on the input of the functionblock to the value in pulses to send to the controller

The **Enable** part stops all the movements of the motors when the enable is not “TRUE”.

The **Homing Module** part is a step program which makes the disks do a reference travel and set a new zero position.

The **Sorting** part is the part where the positioning of the disks is handled.

The **Rollers** part is where the starting and stopping of the rollers is handled.

The **Communication** part checks if the communication with the Conveylinx module is still online.

The **Status** part checks the status of the disk and the rollers and outputs them on the functionblock.

For the exact functioning of the function block, the example code is supported with comments on what certain parts of the code do. The code is just an example on how to start with the Flowsort diverter and is freely accessible for the integrator to adjust.

3.6.3 MOTIONLIX-AI CONTROLLER LINK SOFTWARE TO HARDWARE

The link from the software to the hardware can be done directly or by a global variable. In the example project all “hardware” variables are firstly linked to a global variable, this variable is then linked to the input of the function block.

To link the variables directly, go to the controller and right-click on variable and click on change link. Here you can directly select the instance to which you want to attach the “hardware” variable to. In the table below the input/output variables with their “hardware” variable links are shown.

Input	Datatype	Hardware Variable
bMDR_Homing_Sensor_Disc	Bool	Module Inputs_AllSensors.0
bMDR_Heartbeat	Bool	Module Inputs_AllSensors.15
sMDR_DiskDiagnostic	Uint	Module Diagnostic_DiagnosticLeft
sMDR_RollerDiagnostic	Uint	Module Diagnostic_DiagnosticRight
iMDR_DistanceDisc	Uint	Module Inputs_PositionLeft
wMDRStatusDisk	Usint	Module Inputs_ServoStateL
wMDRStatusRoller	Usint	Module Inputs_ServoStateR
Output	Datatype	Hardware Variable
iMDR_Speed_Disc	Uint	Motor Setup_SpeedLeft
iMDR_BrakeMode_Disc	Usint	Motor Setup_BrakeModeLeft
iMDR_Acc_Disc	Uint	Motor Ramps_AccelLeft
iMDR_Dec_Disc	Uint	Motor Ramps_DecelLeft
iMDR_Speed_Roller	Uint	Motor Setup_SpeedRight
iMDR_BrakeMode_Roller	Usint	Motor Setup_BrakeModeRight
iMDR_Acc_Roller	Uint	Motor Ramps_AccelRight
iMDR_Dec_Roller	Uint	Motor Ramps_DecelRight
iMDR_ControlMotor_Disc	Usint	Motor Setup_MDRCtrlLeft
iMDR_ServoCTRL_Disc	Uint	Servo/Periphery_ServoCtrl
iMDR_ControlDistance_Disc	Uint	Servo/Periphery_ServoLeft
iMDR_ControlMotor_Roller	Usint	Motor Setup_MDRCtrlRight

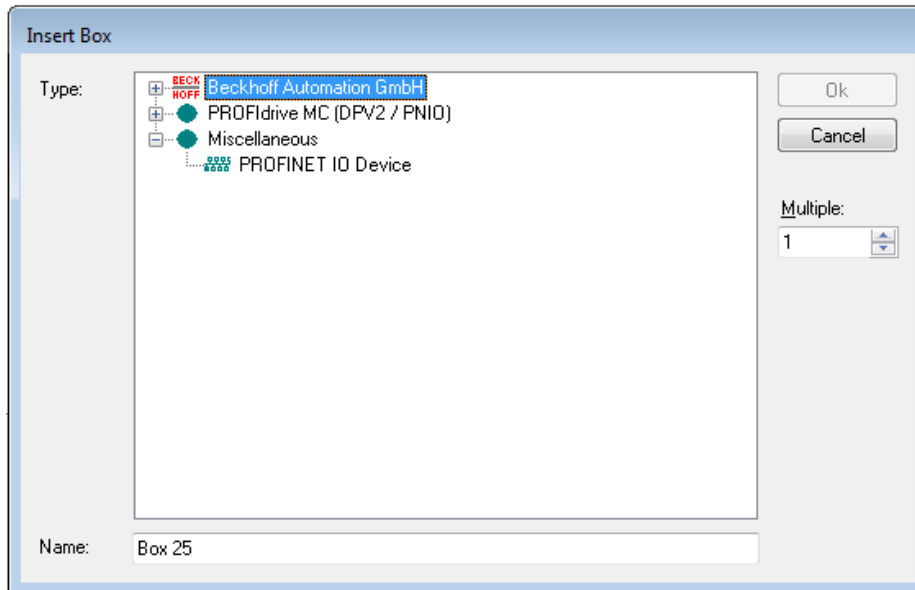
For the correct functioning of the function block it is mandatory that the variables are linked according to this table.

3.6.4 CONVEYLIX-AI2 CONTROLLER HARDWARE

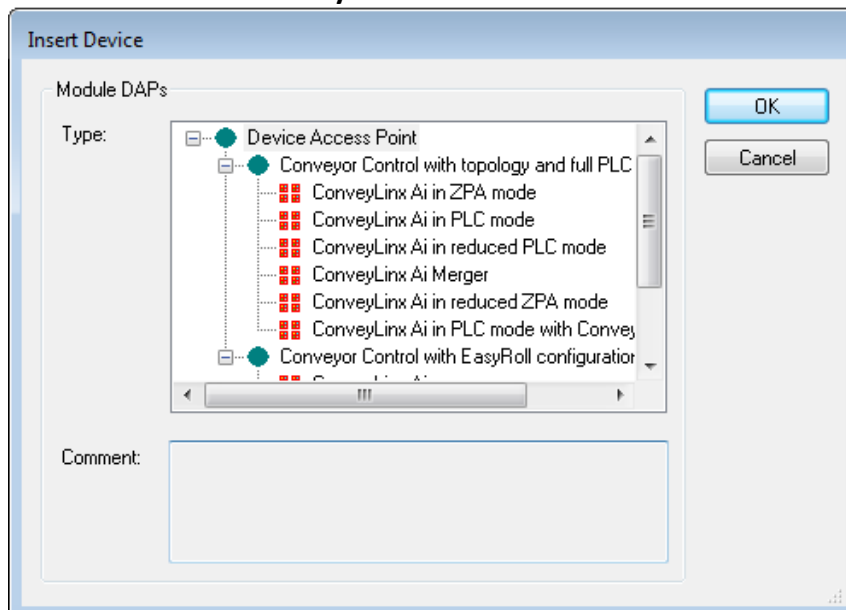
To integrate the Flowsort 24V Diverter in a Beckhoff Controller, , first the GSDML file of the controller must be downloaded. This can be done on the Pulserollers website:

<https://pulseroller.com/downloads/>. Go to Software And Firmware downloads -> PLC Connectivity -> PLC – ProfiNET GSDML files -> Conveylix AI2 -> download the file called: GSDML.

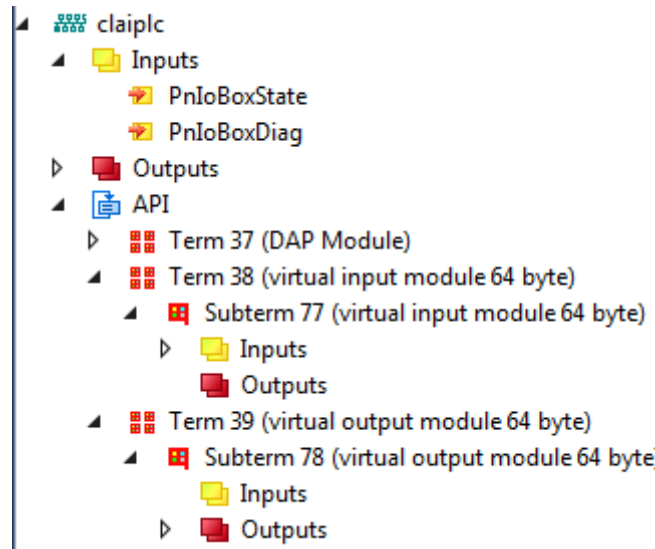
Now the ConveyLinX-Ai2 controller can be added to the project. In the project under I/O -> Devices, go to the ProfiNET master at which the Conveylinx-Ai2 controller is connected to and then push insert. The following popup will be shown:



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 Ai 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 AI 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

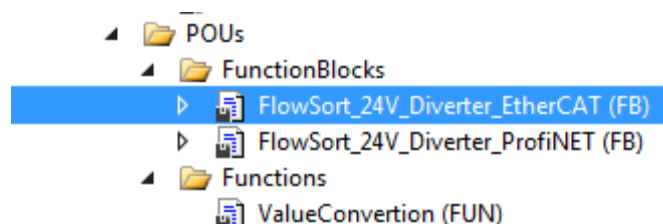
Right side_Motor mode -> ECO mode

Right side_Brake mode -> Normal brake method

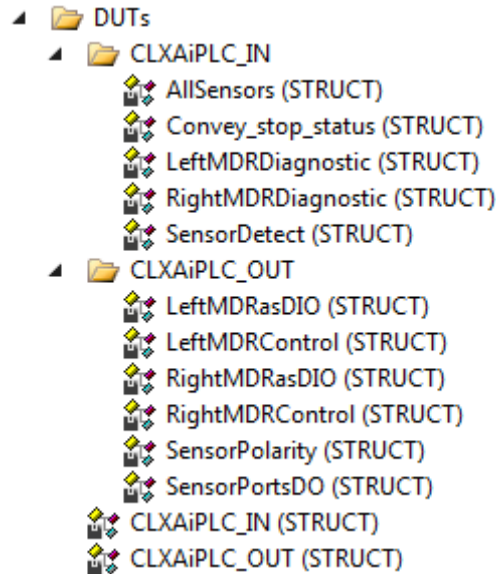
The speeds and ramps are send in the program so it is not necessary to set the speed in the configuration of the controller. The values used here are only used after a communication failure as "initial values".

3.6.5 CONVEYLINX-AI2 CONTROLLER SOFTWARE

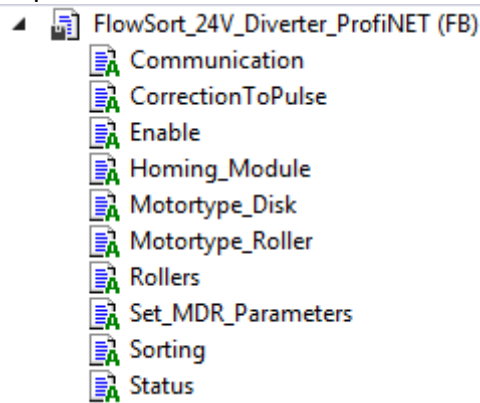
In the Demo program, the following blocks are available. For the integration of the Flowsort function block, copy the FlowSort_24V_Diverter_ProfiNET and the ValueConversion into the project.



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



The Flowsort function block is freely accessible for the integrator to adapt. The example function block is divided in 12 parts.



The **MotorType_Disk** part scales the speed values from the disk from 0 – 100% to pulses to send to the controller

The **MotorType_Roller** part scales the speed values from the rollers from 0 – 100% to pulses to send to the controller

The **set_MDR_Parameters** part sets some default settings to the controller

The **CorrectionToPulse** part recalculates the angles on the input of the functionblock to the value in pulses to send to the controller

The **Enable** part stops all the movements of the motors when the enable is not “TRUE”.

The **Homing Module** part is a step program which makes the disks do a reference travel and set a new zero position.

The **Sorting** part is the part where the positioning of the disks is handled.

The **Rollers** part is where the starting and stopping of the rollers is handled.

The **Communication** part checks if the communication with the Conveylinx module is still online.

The **Status** part checks the status of the disk and the rollers and outputs them on the functionblock.

For the exact functioning of the function block, the example code is supported with comments on what certain parts of the code do. The code is just an example on how to start with the Flowsort diverter and is freely accessible for the integrator to adjust.

3.6.6 CONVEYLINX-AI2 CONTROLLER LINK SOFTWARE TO HARDWARE

The link from the software to the hardware can be done directly or by a global variable. In the example project all “hardware” variables are firstly linked to a global variable, this variable is then linked to the input of the function block.

To link the variables directly, go to the controller and right-click on variable and click on change link. Here you can directly select the instance to which you want to attach the “hardware” variable to. The array of 64 Bytes in the hardware can be linked directly to the data unit type’s CLXAIPLC_IN and CLXAIPLC_Out while they are the same size.

For the linking make sure the checkbox “Matching Size” in “Show Variable Types” is checked. Then all the instances of the Function block will be shown. Link the Inputs and the Outputs to the function block.

Input	Datatype	Hardware Variable
iMDR_Hardware	CLXAIPLC_IN	Inputs
Output	Datatype	Hardware Variable
OMDR_Hardware	CLXAIPLC_OUT	Outputs

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”. Do this also for the outputs!

For the correct functioning of the function block it is mandatory that the variables are linked according to this table.

3.7 PROGRAM INTEGRATION FOR ALLEN BRADLEY STUDIO 5000 V32 OR HIGHER

For the use of the Flowsort 24V Diverter with a Allen Bradley CompactLogix/ControlLogix, follow the steps described below.

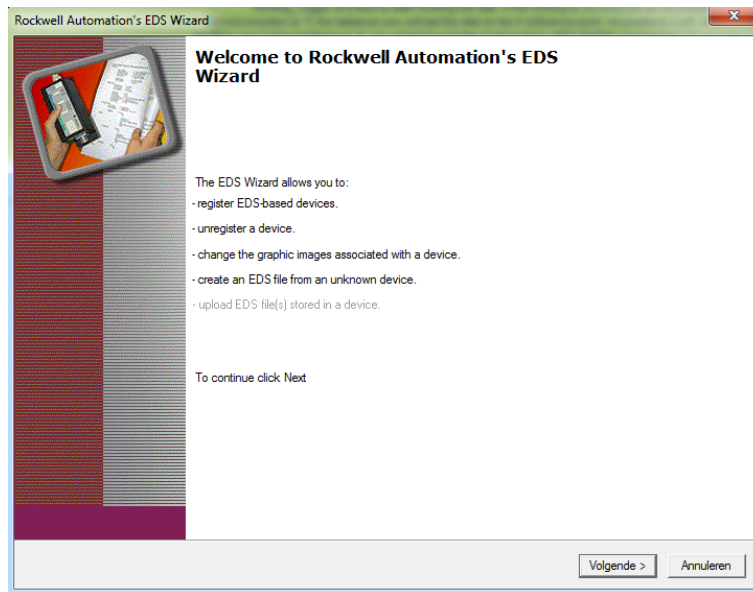
SLD functionblock

Flowsort_24V_Diverter		
Flowsort_24V_Diverter	FB_Control_Module_10_20	...
bEnable	1	(bStat_Ready_Roller)
bHoming_Trigger	BIT_3_HomingStart	(bStat_Ready_Disc)
bSortPos_Execute	Sorting_Execute	(bStat_CommOK)
bRoller_Execute	BIT4_StartRollers	(bStat_HomingOK)
iGearRatio_Disk	M1020_DiskType	(BStat_InPositionDisc)
iMechanicRatio_Disk	15	
iGearRatio_Roller	M1020_RollerType	
iMechanicRatio_Roller	11	
iRotationSpeed	M1020_DiskSpeed	
iRollerSpeed	M1020_RollerSpeed	
iSortPos_MID	Sort_Pos_Middle	
iSortPos_LEFT	Sort_Pos_Left	
iSortPos_RIGHT	Sort_Pos_Right	
iHomePositionOffset	Pos_Correction_Servo	
iSortPos_Location	Sorting_Posto_Execute	
I_Hardware	Module_10_20:1	
O_Hardware	Module_10_20:01	
wSTAT_DiagnosticDisc	0	
wSTAT_DiagnosticRoller	0	

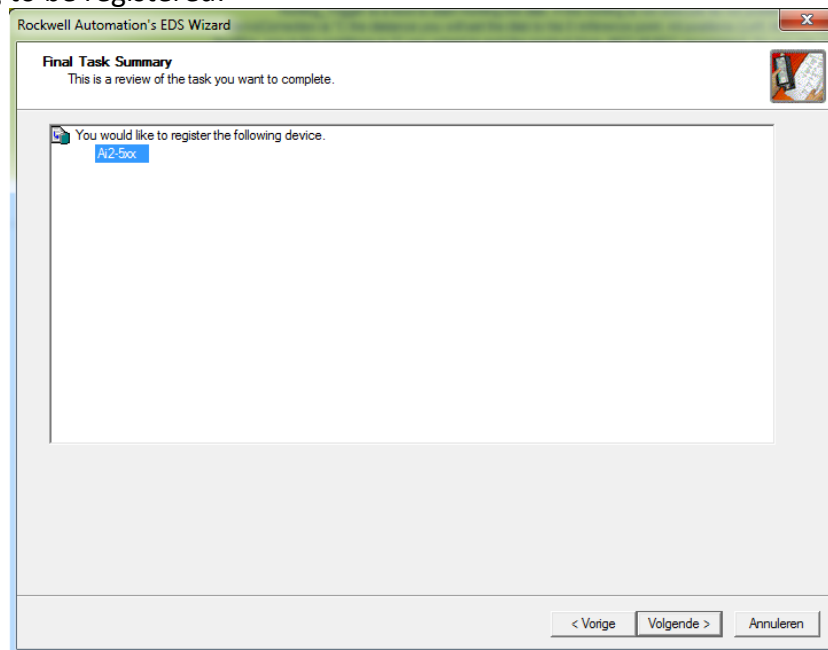
3.7.1 HARDWARE

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 Pulserollers 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:

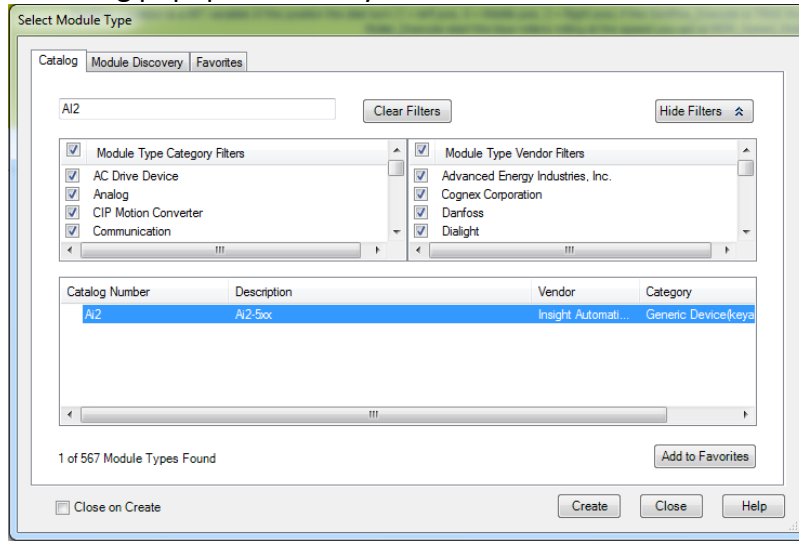


Click Next in this Popup. In the next window Select “**Register an EDS file**” and click next. In the next window select “**Register a single file**” and Browse to the directory where the unpacked EDS file is located and click Open. When the directory is chosen click next. When the path was OK the Installation 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:



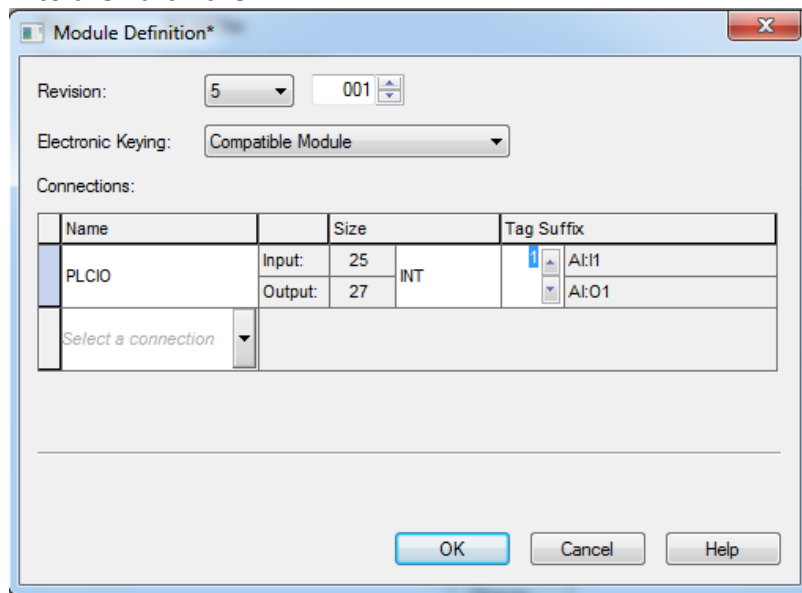
Again click next. After this a confirmation of the successfully registered EDS file will popup. Click Finish to complete the procedure.

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 module can be searched:

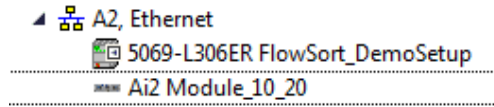


Choose the **AI2** and click on **Create**. The “New Module” popup will be shown. Here the name and IP address can be set by the integrators wishes.

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.

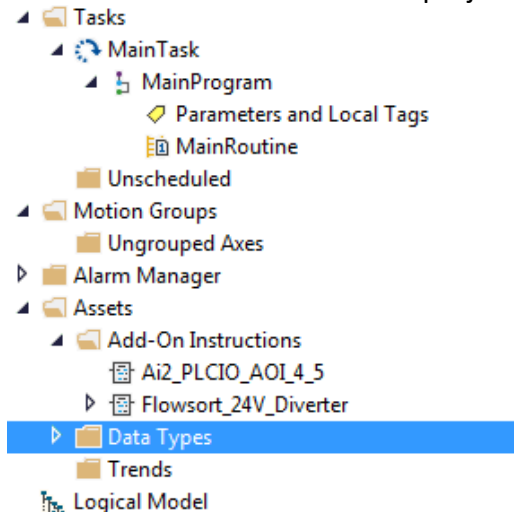


When the settings are correct, click **OK**. A Confirmation popup will be shown that the module definitions will be changed, Click **Yes**. In the New Module popup also Click **OK**. Now the module is available in the I/O configuration.



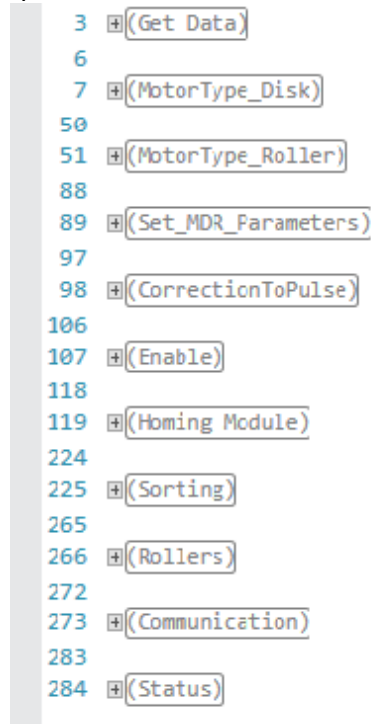
3.7.2 SOFTWARE

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 Flowsort function block is freely accessible for the integrator to adapt. The example function block is divided in 11 parts.



The **Get Data** part reads out the data from the Ethernet/IP Device with the given Hardware data and convert the data to values.

The **MotorType_Disk** part scales the speed values from the disk from 0 – 100% to pulses to send to the controller

The **MotorType_Roller** part scales the speed values from the rollers from 0 – 100% to pulses to send to the controller

The **set_MDR_Parameters** part sets some default settings to the controller

The **CorrectionToPulse** part recalculates the angles on the input of the functionblock to the value in pulses to send to the controller

The **Enable** part stops all the movements of the motors when the enable is not “TRUE”.

The **Homing Module** part is a step program which makes the disks do a reference travel and set a new zero position.

The **Sorting** part is the part where the positioning of the disks is handled.

The **Rollers** part is where the starting and stopping of the rollers is handled.

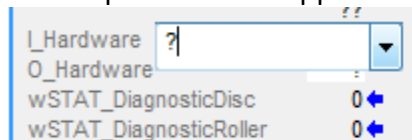
The **Communication** part checks if the communication with the Conveylinx module is still online.

The **Status** part checks the status of the disk and the rollers and outputs them on the functionblock.

For the exact functioning of the function block, the example code is supported with comments on what certain parts of the code do. The code is just an example on how to start with the Flowsort diverter and is freely accessible for the integrator to adjust.

3.7.3 CONTROLLER LINK SOFTWARE TO HARDWARE

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



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

Name	Data Type	Usage
DLD_24V_002:11	_055C:Ai2_1F7F96C5:1:0	<controller>
SLD_24V_001:11	_055C:Ai2_1F7F96C5:1:0	<controller>

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

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

3.8 PROGRAM INTEGRATION FOR OMRON SYSMAC STUDIO 1.40.0 OR HIGHER

For the use of the Flowsort 24V Diverter with a Omron CPU Unit, follow the steps described below.

FlowSort_24V_Diverter	
bEnable	bStat_CommOK
bHoming_Trigger	bStat_HomingOK
bSortPos_Execute	bStat_InPositionDisc
bRoller_Execute	bStat_Ready_Disc
iGearRatio_Disk	bStat_Ready_Roller
iMechanicRatio_Disk	O_Hardware
iGearRatio_Roller	wSTAT_DiagnosticDisc
iMechanicRatio_Roller	wSTAT_DiagnosticRoller
iRotationSpeed	
iRollerSpeed	
iSortPos_MID	
iSortPos_LEFT	
iSortPos_RIGHT	
iHomePositionOffset	
iSortPos_Location	
I_Hardware	

3.8.1 HARDWARE AND LINK TO SOFTWARE

To integrate the Flowsort 24V Diverter in a Omron Controller with Sysmac Studio, first the EDS file of the controller must be downloaded. This can be done on the Pulserollers 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

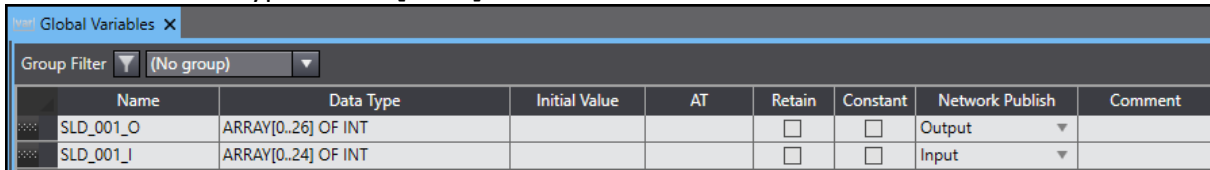
Now the EDS file must be placed into the installation folder of Sysmac Studio. Normally this should be:

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**.



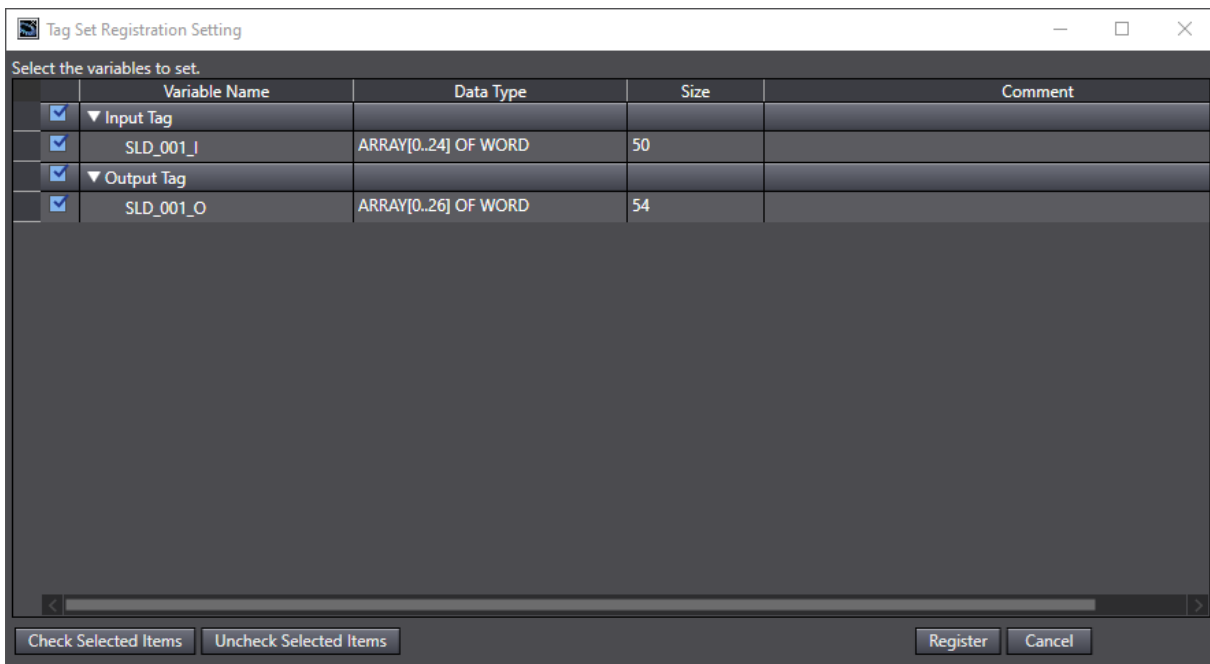
Name	Data Type	Initial Value	AT	Retain	Constant	Network Publish	Comment
SLD_001_O	ARRAY[0..26] OF INT			<input type="checkbox"/>	<input type="checkbox"/>	Output	
SLD_001_I	ARRAY[0..24] OF INT			<input type="checkbox"/>	<input type="checkbox"/>	Input	

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”**.

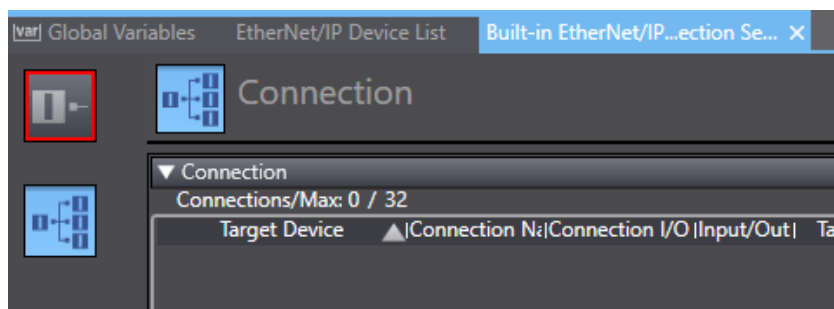
In the opened window select the In and Output tags you want to register, and click **“Register”**.



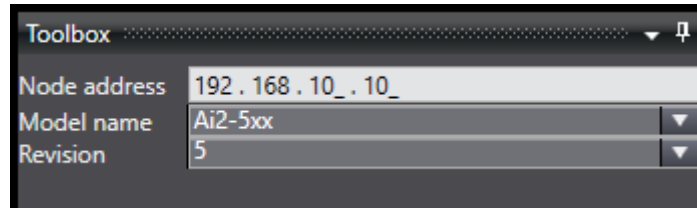
Variable Name	Data Type	Size	Comment
▼ Input Tag			
SLD_001_I	ARRAY[0..24] OF WORD	50	
▼ Output Tag			
SLD_001_O	ARRAY[0..26] OF WORD	54	

Buttons: Check Selected Items, Uncheck Selected Items, Register, Cancel

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



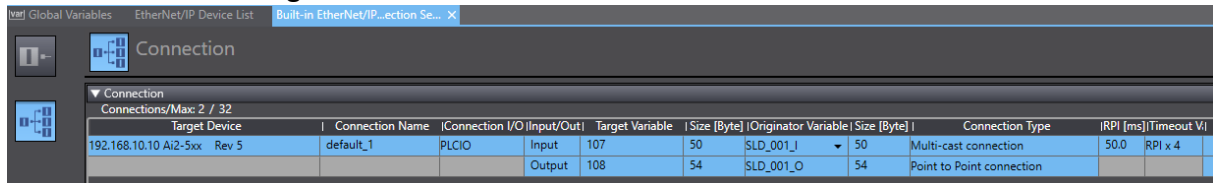
In the Connection window at the right hand side the toolbox is located. In this window click on the “+” to add a new target device. In the window fill in the Node address of the Diverter and select the model name and revision. The model name should be:



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.

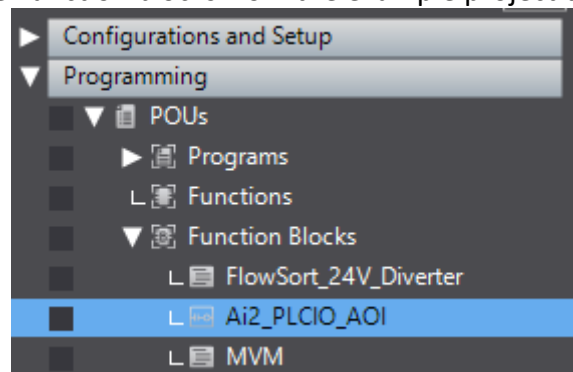
As Connection I/O Select “PLCIO” as target variable select **107** for the Input and **108** for the Output. As Originator Variable select the Tag sets that are registered. When all the settings are filled in correctly everything will have a light blue color. When a setting is pink that means that the 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.

3.8.2 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.

Structures	Name	Base Type	Offset Type	Offset Byte
Union	Ai2_Module	STRUCT	NJ	
Enumerated	Ai2_Module_In_Raw	STRUCT	NJ	
	Ai2_Module_Out_Raw	STRUCT	NJ	
	Ai2_Module_In_UDT	STRUCT	NJ	
	Ai2_Module_Out_UDT	STRUCT	NJ	

The Flowsort function block is freely accessible for the integrator to adapt. The example function block is divided in 11 parts.

```

1 // Flowsort by 24V Diverter Function Block
2 // Release 03/03/2021 V2.1
3
4 // Get/Send Data
5 [F AlwaysOn THEN...
10
11 // MotorType_Disk
12 [F AlwaysOn THEN...
56
57 // MotorType_Roller
58 [F AlwaysOn THEN...
98
99 // Set_MDR_Parameters
100 [F AlwaysOn THEN...
106
107 //CorrectionToPulse
108 [F AlwaysOn THEN...
116
117 //Enable
118 [F NOT bEnable THEN...
126
127 // Homing Module
128 [F AlwaysOn THEN...
225
226 // Sorting
227 [F AlwaysOn THEN...
265
266 // Rollers
267 [F AlwaysOn THEN...
271
272 // Commuication
273 [F AlwaysOn THEN...
279
280 // Status
281 [F AlwaysOn THEN...

```

The **Get/Send Data** part reads out the data from the Ethernet/IP Device and sends the data to the Ethernet/IP device with the given Hardware data and convert the data to values .

The **MotorType_Disk** part scales the speed values from the disk from 0 – 100% to pulses to send to the controller

The **MotorType_Roller** part scales the speed values from the rollers from 0 – 100% to pulses to send to the controller

The **set_MDR_Parameters** part sets some default settings to the controller

The **CorrectionToPulse** part recalculates the angles on the input of the function block to the value in pulses to send to the controller

The **Enable** part stops all the movements of the motors when the enable is not “TRUE”.

The **Homing Module** part is a step program which makes the disks do a reference travel and set a new zero position.

The **Sorting** part is the part where the positioning of the disks is handled.

The **Rollers** part is where the starting and stopping of the rollers is handled.

The **Communication** part checks if the communication with the Conveylinx-Ai2 module is still online.

The **Status** part checks the status of the disk and the rollers and outputs them on the function block.

For the exact functioning of the function block, the example code is supported with comments on what certain parts of the code do. The code is just an example on how to start with the FlowSort diverter and is freely accessible for the integrator to adjust.

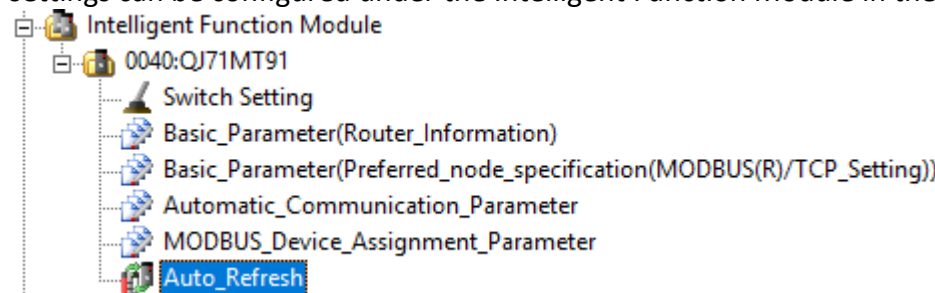
3.9 PROGRAM INTEGRATION FOR MITSUBISHI GX WORKS 2 OR HIGHER

For the use of the Flowsort 24V Diverter with a Mitsubishi Q Series Controller via Modbus, follow the steps described below.

		Flowsort_1		
		Flowsort		
M5000	bEnable		bStat_CommOK	M5004
M5001	bHoming_Trigger		bStat_HomingOK	M5005
M5002	bRoller_Execute		BStat_InPositionDisc	M5006
M5003	bSortPos_Execute		bStat_Ready_Disc	M5007
.15	iGearRatio_Disk		bStat_Ready_Roller	M5008
.11	iGearRatio_Roller		wSTAT_DiagnosticDisc	D15001
.0	iHomePositionOffset		wSTAT_DiagnosticRoller	D15002
.2	iMechanicRatio_Disk		iMDR_Speed_Disc	D6010
.1	iMechanicRatio_Roller		iMDR_BrakeMode_Disc	D6005
.80	iRollerSpeed		iMDR_Acc_Disc	D6012
.80	iRotationSpeed		iMDR_Dec_Disc	D6013
.60	iSortPos_LEFT		iMDR_Speed_Roller	D6011
D15000	iSortPos_Location		iMDR_BrakeMode_Roller	D6008
.0	iSortPos_MID		iMDR_Acc_Roller	D6014
.60	iSortPos_RIGHT		iMDR_Dec_Roller	D6015
D1001.0	bMDR_Homing_Sensor_Disk		iMDR_ControlMotor_Disc	D6004
D1001.F	bMDR_Heartbeat		iMDR_ServoCTRL_Disc	D6024
D1007	sMDR_DiscDiagnostic		iMDR_ControlDistance_Disc	D6023
D1011	sMDR_RollerDiagnostic		iMDR_ControlMotor_Roller	D6007
D1019	iMDR_DistanceDisc			
D1021	wMDRStatusDisk			
D1022	wMDRStatusRoller			

3.9.1 HARDWARE

To integrate the Flowsort 24V Diverter 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-adress of the device depends on the configuration of the conveylinx 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 Address is not the address that is used in the PLC! This address is in the Auto Refresh parameter which is explained later. 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 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 module.

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

3.9.2 CONTROLLER LINK SOFTWARE TO HARDWARE RECEIVE DATA

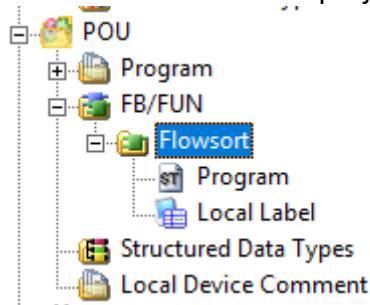
The Auto_Refresh settings are customer specific:

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 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 modules).

3.9.3 SOFTWARE

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



The Flowsort function block is freely accessible for the integrator to adapt. The example function block is divided in 12 parts.

The **MotorType_Disk** part scales the speed values from the disk from 0 – 100% to pulses to send to the controller

The **MotorType_Roller** part scales the speed values from the rollers from 0 – 100% to pulses to send to the controller

The **set_MDR_Parameters** part sets some default settings to the controller

The **CorrectionToPulse** part recalculates the angles on the input of the functionblock to the value in pulses to send to the controller

The **Enable** part stops all the movements of the motors when the enable is not “TRUE”.

The **Homing Module** part is a step program which makes the disks do a reference travel and set a new zero position.

The **Sorting** part is the part where the positioning of the disks is handled.

The **Rollers** part is where the starting and stopping of the rollers is handled.

The **Communication** part checks if the communication with the Conveylinx module is still online.

The **Status** part checks the status of the disk and the rollers and outputs them on the functionblock.

For the exact functioning of the function block, the example code is supported with comments on what certain parts of the code do. The code is just an example on how to start with the Flowsort diverter and is freely accessible for the integrator to adjust.

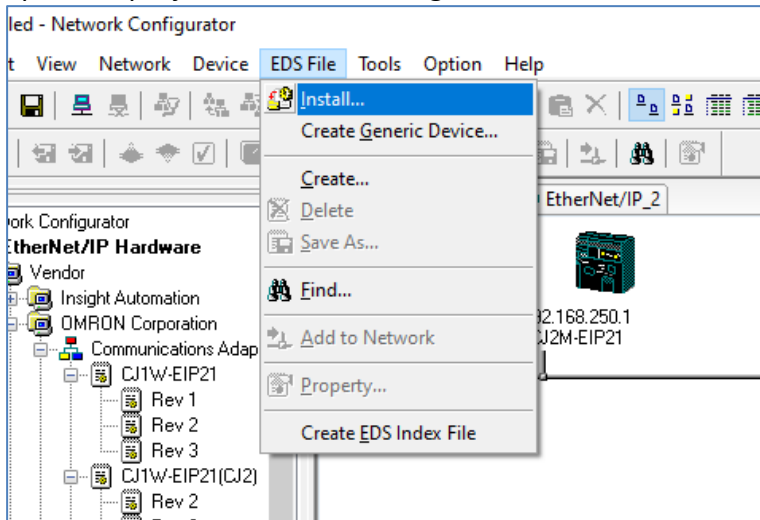
3.10 PROGRAM INTEGRATION FOR OMRON CX-PROGRAMMER VERSION 9

For the use of the FlowSort 24V Diverter with a Omron CPU Unit, follow the steps described below.

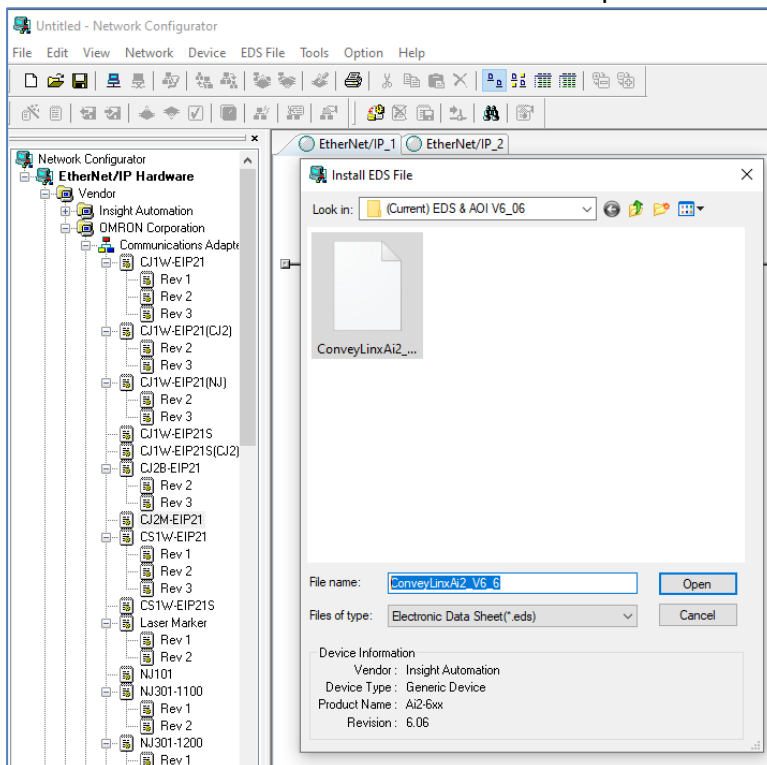
FlowSort24V	
FlowSort_24V_Diverter	
P_On	(BOOL) EN
Always ON FL...	(BOOL) ENO
Homing_Trig	(BOOL) bHoming_Trigger
SortPosExe...	(WORD) wSTAT_DiagnosticRoller
RollerExecute	(WORD) wSTAT_DiagnosticDisc
SLD_001_O[4]	(BOOL) bStat_Ready_Roller
RotationSpeed	(BOOL) bStat_Ready_Disc
SLD_001_O[0]	(BOOL) bStat_InPositionDisc
RotationSpeed	(BOOL) bStat_HomingOK
RotationSpeed	(INT) iRotationSpeed
RollerSpeed	(INT) iRollerSpeed
SortPos_Mid	(INT) iSortPos_MID
SortPos_Left	(INT) iSortPos_LEFT
SortPos_Right	(INT) iSortPos_RIGHT
HomePosOf...	(REAL) iHomePositionOffset
SortPos_Lo...	(INT) iSortPos_Location
SLD_001_I	(INT[25]) I_Hardware
SLD_001_O	(INT[27]) O_Hardware
CommOK	(BOOL) bStat_CommOK
DiagnosticR...	(WORD) wSTAT_DiagnosticRoller
DiagnosticD...	(WORD) wSTAT_DiagnosticDisc
Ready_roller	(BOOL) bStat_Ready_Roller
Ready_Disk	(BOOL) bStat_Ready_Disc
InPositionDisk	(BOOL) bStat_InPositionDisc
HomingOK	(BOOL) bStat_HomingOK
SLD_001_I	(INT[25]) I_Hardware
SLD_001_O	(INT[27]) O_Hardware

3.10.1 HARDWARE AND LINK TO THE SOFTWARE

To integrate the FlowSort 24V Diverter in an Omron Controller with CX-Programmer, 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 Firmware & Software Downloads -> PLC-Files -> ConveyLinX-Ai -> Ethernet IP Files -> Latest, and Download the file: EDS & AOI Vx x
Now the EDS file must be installed in the hardware configuration, for that it is necessary to open the projects Network Configurator. In the menu 'EDS File' select 'Install'.

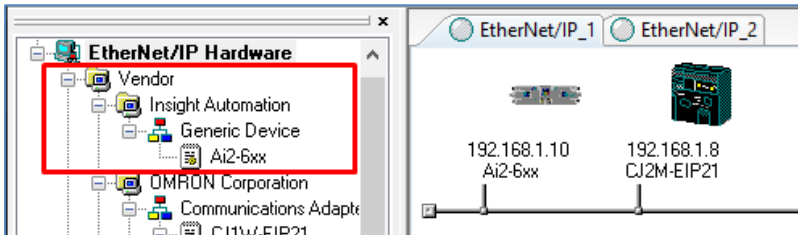


Browse to the downloaded file and click on 'Open':

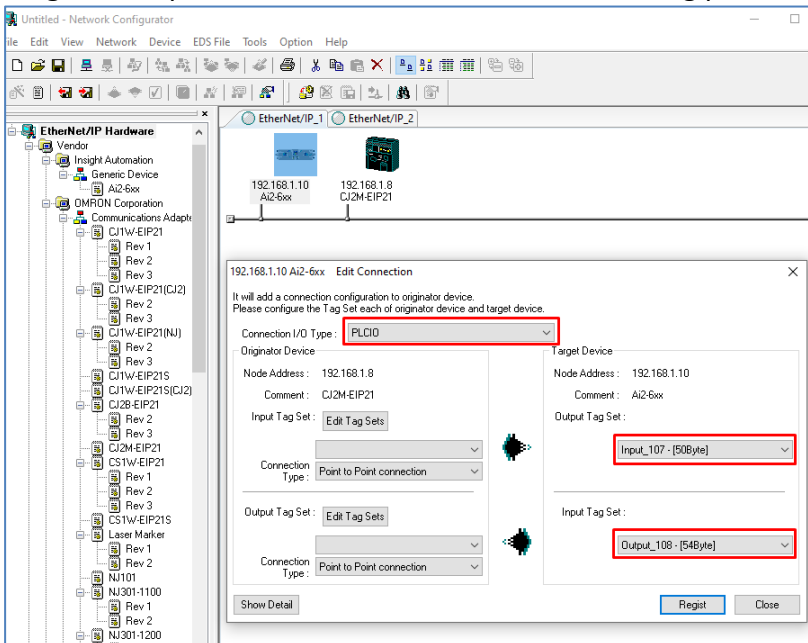


After the EDS was installed, it is necessary to integrate it in the hardware configuration of the existing project.

Add Ai2 device to the network with the corresponding IP address:



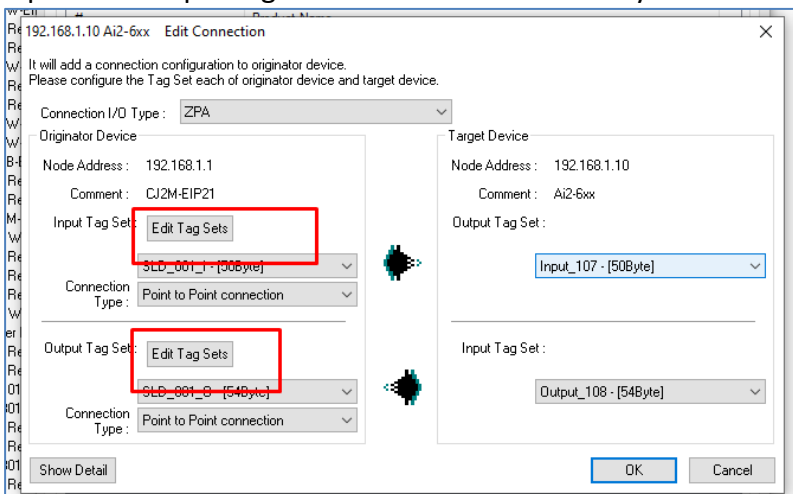
In the described case, the communication adapter CJ2M-EIP21 will be used. To integrate it drag and drop Ai2 module to CJ2M with the following parameters:



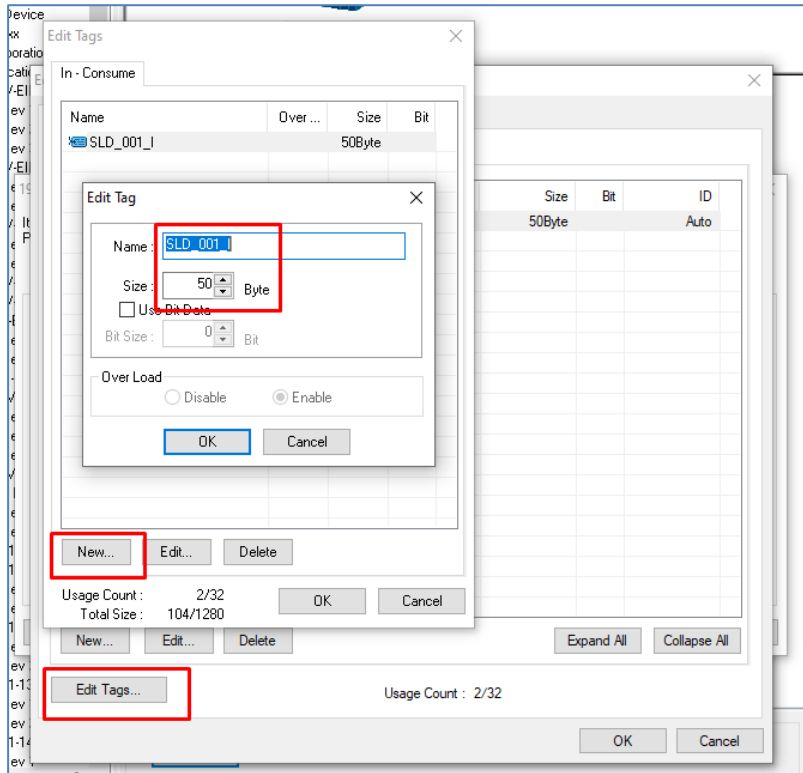
As Connection I/O Select: PLCIO

As target variable select 107 for the Input, and 108 for the Output.

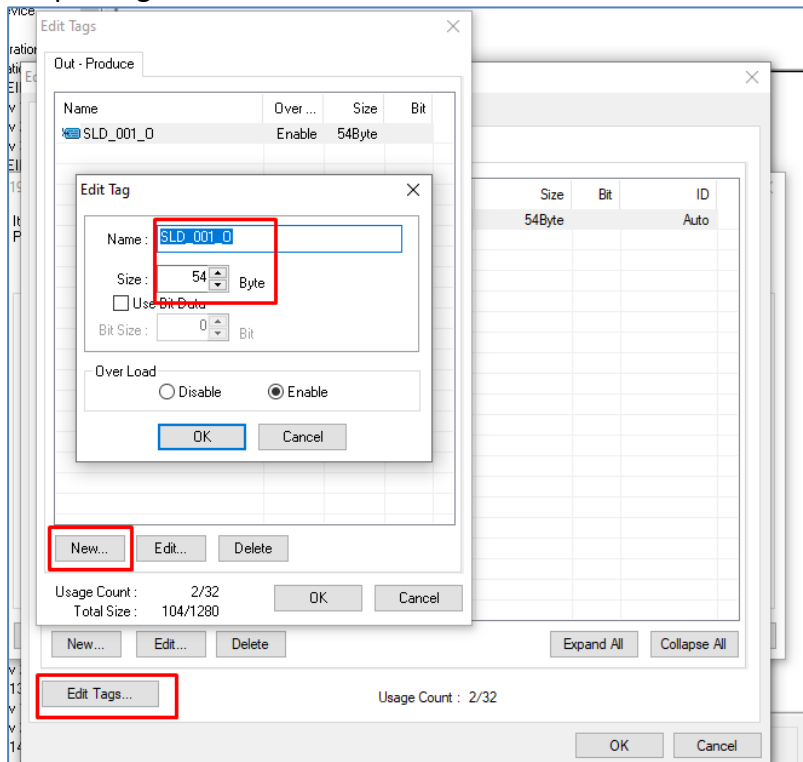
Input and Output tags have to be added manually in CJ2M:



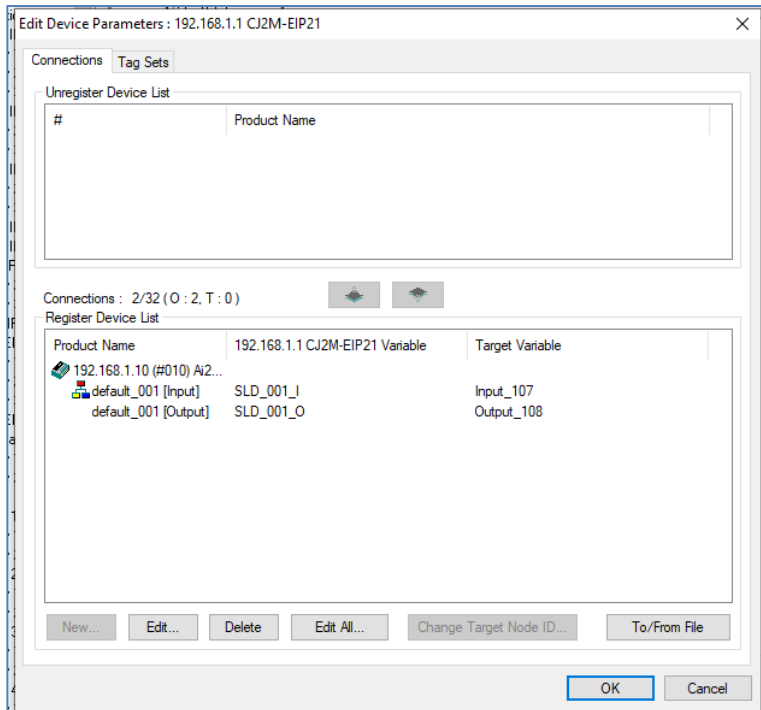
Input Tag set:



Output Tag set:



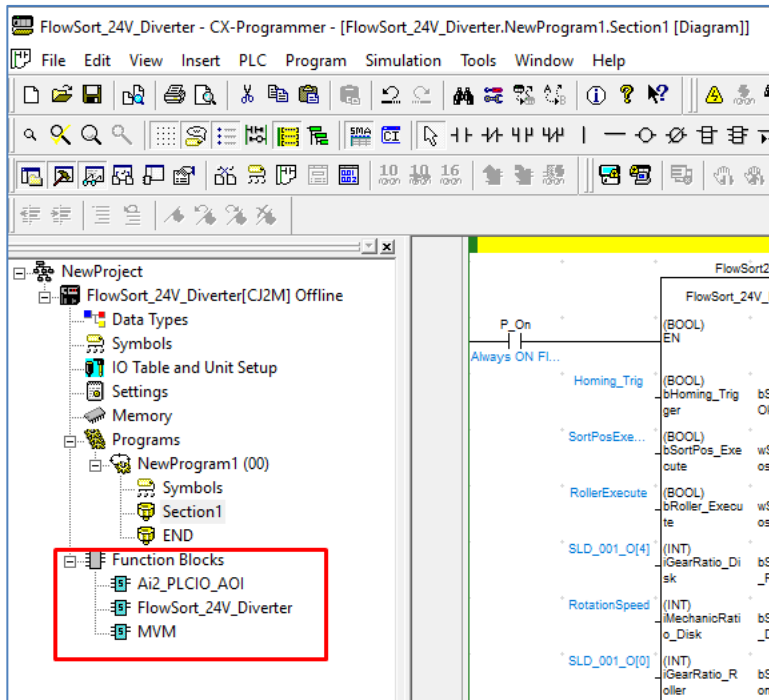
After the successful registration, the registered device of Ai2 in CJ2M should be:



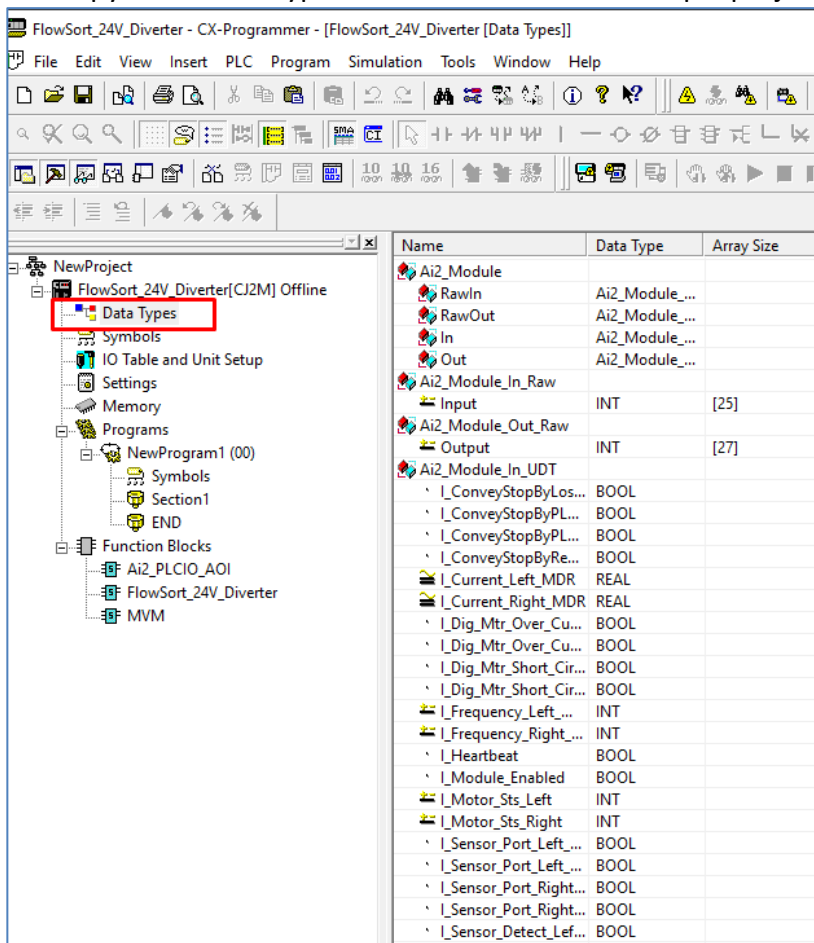
As Originator Variable select the Tag sets that are registered. After the device has been added and integrated, the configuration is necessary to be downloaded.

3.10.2 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 all the datatypes structures from the example project to the project.



The FlowSort function block is freely accessible for the integrator to adapt.

The example function block is divided in 11 parts:

- **Get/Send Data** parts reads out the data from the Ethernet/IP Device and sends the data to the Ethernet/IP device with the given Hardware data and convert the data to values .
- **MotorType_Disk** part scales the speed values from the disk from 0 – 100% to pulses to send to the controller
- **MotorType_Roller** part scales the speed values from the rollers from 0 – 100% to pulses to send to the controller
- **set_MDR_Parameters** part sets some default settings to the controller
- **CorrectionToPulse** part recalculates the angles on the input of the function block to the value in pulses to send to the controller
- **Enable** part stops all the movements of the motors when the enable is not “TRUE”.
- **Homing Module** part is a step program which makes the disks do a reference travel and set a new zero position. Integration Manual SLD/DLD/ZZ 24V
- **Sorting** part is the part where the positioning of the disks is handled.
- **Rollers** part is where the starting and stopping of the rollers is handled.
- **Communication** part checks if the communication with the Conveylinx-Ai2 module is still online.
- **Status** part checks the status of the disk and the rollers and outputs them on the function block.

For the exact functioning of the function block, the example code is supported with comments on what certain parts of the code do. The code is just an example on how to start with the FlowSort diverter and is freely accessible for the integrator to adjust.

APPENDIX A – UPDATING THE FIRMWARE

Flowsort has improved the movements of the controls in multiple ways. One of them is that the firmware for the Conveylinx-Ai2 controller is created especially for our 24V diverters. Along with the supplier of the controllers, Flowsort has developed a special firmware which will improve the start and stop movements of the discs.

The controllers on the diverters with the old firmware need to be upgraded manually. The new firmware has to be upgraded by the EasyRoll program. This can be done for the complete line of controller in a sequence or for one single separate controller. To upgrade the firmware, follow the following steps.

Step 1:

Download and open the EasyRoll program. It can be downloaded on the website of the supplier of the controllers: <https://www.pulseroller.com/downloads>. Go to SOFTWARE AND FIRMWARE DOWNLOADS -> FIRMWARE -> CONVEYLINX AI2 -> LATEST VERSION, Download the file: ConveyLinxAi x x x

Download the Flowsort firmware here or ask the Flowsort team for the newest files:
https://www.flow-sort.com/images/files/downloads/ConveyLinxAI_5_6_2-FLOWSORT.bin

Step 2:

Make a connection to the controllers by typing in the IP-address of the controller. Click on refresh to get the controller alive.

**Only upgrade the firmware on the controllers in the diverters!
Do not upgrade the firmware on the controller on other equipment
(i.e. controllers on roller conveyors / belt conveyors)**

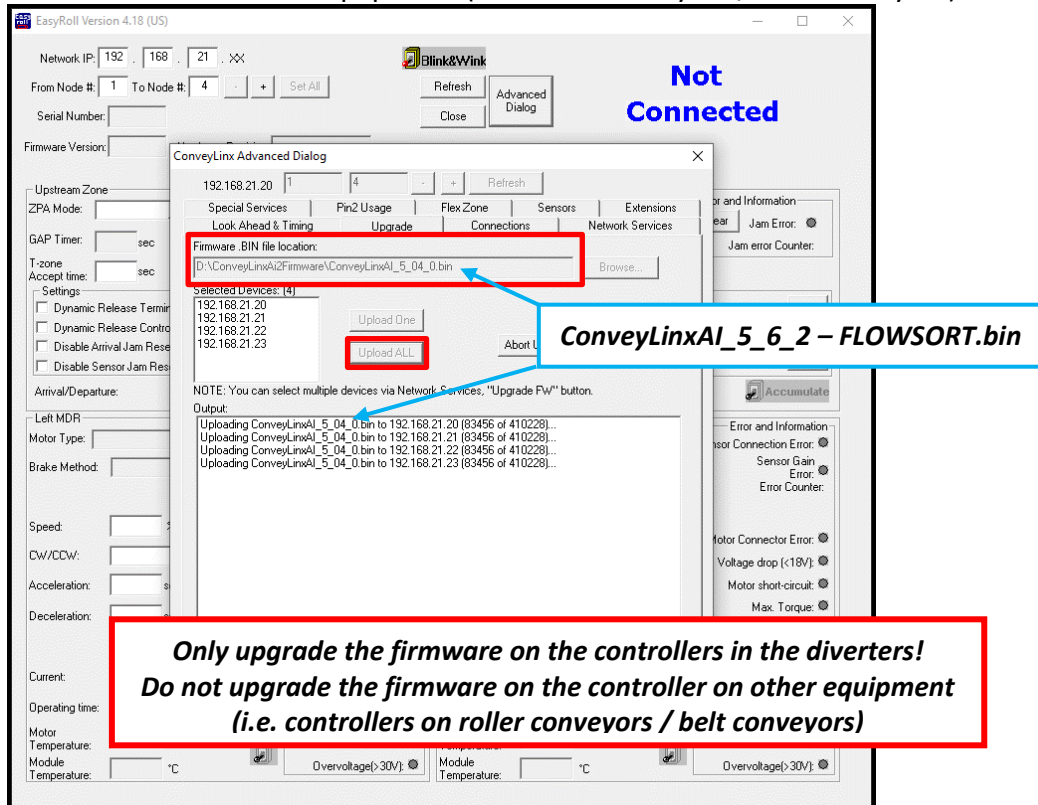
For the diverters delivered after 12-2020, the firmware is upgraded before delivery.

Step 3:

Click on Advanced Dialog, a new window will open. Go to the Upgrade tab. Click on Browse...

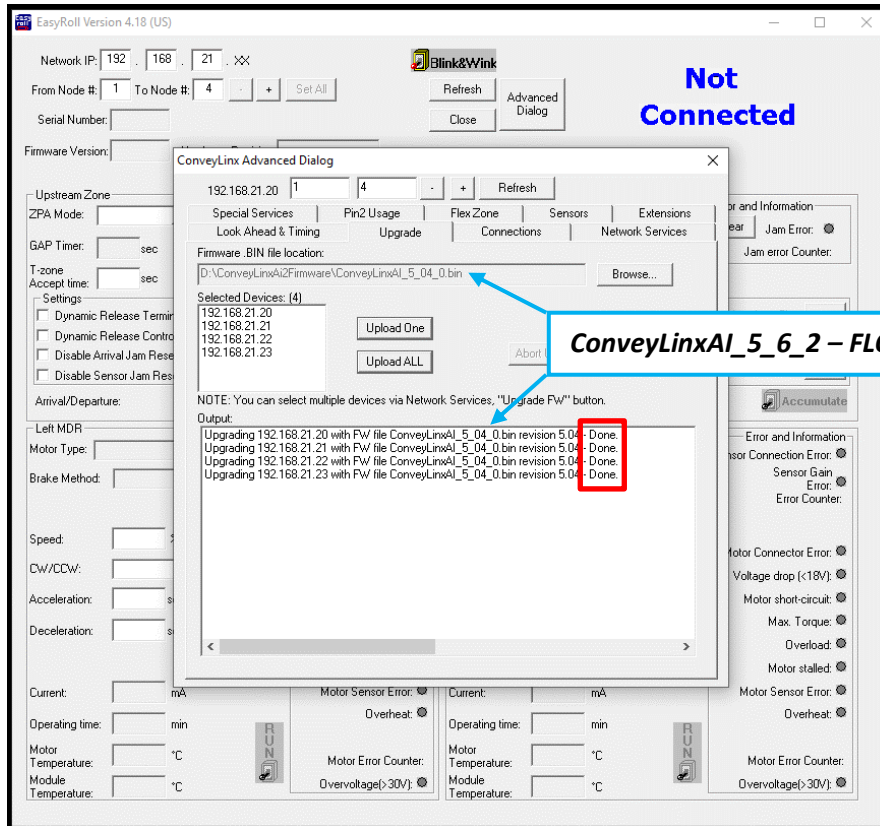
Step 4:

Select and open the Flowsort firmware (**ConveyLinxAI_5_6_2 – FLOWSORT.bin**). Click only on Upload ALL to upgrade all the controllers for the diverters in the sequence at once. Do not upgrade the controllers of other equipment (i.e. roller conveyors / belt conveyors)!



Step 5:

Wait for the program to show “Done” after every node. Now the upgrade to the Flowsort firmware has been succeeded.



Please note that if you upgrade firmware, all of each module's settings (motors, speeds, acceleration, deceleration, etc.) are preserved.

**Only upgrade the firmware on the controllers in the diverters!
Do not upgrade the firmware on the controller on other equipment
(i.e. controllers on roller conveyors / belt conveyors)**