

PLC software manual

X-Flow90 Device

Version 1.0

Flowsort B.V.

Chisinau,

Moldova

20/03/2024



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

Tel. +373 22 996 170

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

ISD and the ISD logo are protected by ISD.

All Rights Reserved.

© 2008 - 2021 ISD

Possession, use, duplication, or dissemination of this documentation is authorized only pursuant to a valid written license from the owner of the copyright.

This document is confidential and copyrighted by ISD. It is an unpublished work protected under the international copyright laws.



Table of contents

1.	Changelog	4
2.	Introduction	5
2.1	Introduction	5
3.	Hardware	6
3.1	Controller connection	6
3.2	Hardware configuration integration	7
4.	Software	10
4.1	Integration	10
4.2	Function Block Inputs	16
4.3	Function Block outputs	21
4.4	Flow chart	24
4.5	Function Block: explanation	25
4.6	Program integration for ALLEN BRADLEY STUDIO 5000 V35	34
4.7	Program integration for OMRON SYSMAC STUDIO V1.58.0	38
4.8	Program integration for MITSUBISHI GX WORKS 2 OR HIGHER	41
4.9	Program integration for BECKHOFF TWINCAT 3.4.3147.18	43
5.	IO and Functional tests	47
5.1	IO test	47
5.2	Functional test	47
<mark>6.</mark>	Tips and tricks/lessons learned	50





1. Changelog

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



2. Introduction

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

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

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

2.1 Introduction

Flowsort

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

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

ISD

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



3. Hardware

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

- ConveyLinx-Ai2 controller
- Two inductive sensors for detecting the position of the mechanism.
- A Pulseroller PGD motor for lifting the mechanism up:
 - https://www.pulseroller.com/products/geared-drives/geared-drives-ai/pulse-geared-drive-ai-24v-48v
- A Pulseroller Senergy-Ai motor driven roller that drives the transport belts:
 https://www.pulseroller.com/products/motor-rollers/motor-rollers-ai/senergy-ai-24v

3.1 Controller connection

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

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







General description of the module:

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

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

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

3.2 Hardware configuration integration

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

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

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

Pro	oject Edit View Insert Online	Options Tools Window Help
Ľ	🕴 🎦 🔚 Save project 📑 🐰 🗐 🛙	Settings
	Project tree	Support packages
	Devices	Manage general station description files (GSD) – Start Automation License Manager
6		Show reference text
, m	🕨 📜 PLC tags	🛄 Global libraries 🔹 🕨



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.

_ # #×	Hardware catalog 🛛 🗐 🗉		
vice view	Options		
			H
^	× Catalog	_	rd w
			are
=		m	8
	Filter Profile: <all></all>	UÌ,	5
	Controllers	^	ß
	🕨 🫅 HMI		
	PC systems		8.
- 110	Drives & starters		0
- 11	Image: Interview of the second s		LE:
- 11	Detecting & Monitoring		le
	Distributed I/O		00
- 6	Power supply and distribution		l <u>s</u>
	Field devices		
, ŝ	▼ ☐ Other field devices		Þ
- 5	Additional Ethernet devices		Ta
	▼ 📑 PROFINETIO		sks
	Controllers		
- 11	Drives		
	Encoders		5
- 11	Gateway		bra
	General	≡	Tie
- 11	▼ 10		l "
	Euchner GmbH + Co. KG		F
	 Industrial Software Co. 		A
	▼ Lin ConveyLinx		L [‡]
~	Conveyor Control with EasyRoll configuration		s
	Conveyor Control with standard configuration		-
	Conveyor Control with topology and full PL		
	ConveyLinx Al in PLC mode		
	ConveyLinx Al In PLC mode with ConveyLo		
·	Conveyting Ai in reduced PLC mode		
	ConveyLinx AI In reduced PLC mode (24 0		
=	Convey inx Ai in 7PA mode		
	Convey inv Ai Merger		
	Conveychtx Al Merger		

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

CLAiPLC ConveyLinx Ai i	eyLinx Al2	PLC_1 CPU 1515F-2 PN
Not assigned	Change device	
	Add subnet	
	Assign to new subnet	CLAiPLC ConveyLinx Ai i
	Disconnect from subnet	
	Add IO system	

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



The following parameters should be set in the properties:

	Left side										
General											
Ethernet addresses		Left Pin 2 invert									
 Advanced options 		Left Pin 2 PUSH/PULL									
Identification & Maintenance		Left Pin 4 invert									
Module parameters		Left Pin 4 PUSH/PULL									
		Keverse motor									
	Motor slaving	Motor is not a slave									
	Motor mode	ECO mode(3A start / 2.8A continuous current limit)									
	Speed[mm/s]	1000									
	Brake mode	Servo brake method									
	Accelimm	30									
	• Decelimm	30									
-											
	Right side										
		Right Pin 2 invert									
		Right Pin 2 PUSH/PULL									
		Right Pin 4 invert									
		Bight Pin 4 PUSHPULL									
		Beverse motor									
	Motor claving	Motoric pot a clava									
	Wotor slaving	WHICH IS HIDE & SHARE									
	Motor mode	BUDS I mode (5A start / 3.6A continuous current limit)									
	Speed[mm/s]	420									
	Brake mode	Free brake method									
	Accel[mm]	30									
	Decelimm1	30									

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

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

3.2.1 Integration of the addresses

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

FlowSort X_Flow90_V17 > Ungrouped devices > X_Flow90 [ConveyLinx Ai in PLC mode] 🖬												
					2	Fopology v	iew 🚮	Network view	🛐 Device view			
🔐 🗶 Flow90 [ConveyLinx Ai in Pl 💌 🔛 🔛 🔛 🗒 🕰 🖽 🖽 🕲	4	Devic	e overview						-			
	^	- **	Module	Rack	Slot	I address	Q address	Туре	Article no.			
			 X_Flow90 	0	0			ConveyLinx Ai in PL	ConveyLinx Ai			
			Interface	0	0 X1			CLAIPLC				
1 Store			virtual input module 64 byt	0	1	68131		virtual input modul	NO_ORDER_CODE			
	=		virtual output module 64 by	0	2		64127	virtual output mod	NO_ORDER_CODE			
Convegianz A/2												
	-											
	•											

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



4. Software

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

4.1 Integration

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

- PLC Data types
- PLC tags
- Flowsort_XFlow_Divert Function Block

Integration of each will be explained next:

PLC Data types:

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





PLC tags:

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

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



Adjustment of the addresses:

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

Project tree	< 🕕 FlowSo	rt_X_Flow90_	V17 → Ungrouped devices →	X_Flow	v90 [Con	weyLinx A	i in PLC m	ode]	_ # = ×	×	=V17 FlowSort_XFlow90_De	no [CPU 1211C	DC/DC/DC]	PLC tags	 XFlow 		yLinx [2]
Devices				a Te	opology	view d	Networ	k view 🛛 🕅 Devi	ce view	1						🕢 Ta	gs 🔳 U
11 II II I	2 dt 1	Der	vice overview							9	ê 🕑 📴 📴 🕫 😭						
		^ v	Module	Back	Clar	Laddreer	Onddrage	Tune	Articlano		XFlow_ConveyLinx						
 FlowSort_X_Flow90_V17 	^		X Elow90	0	0		Q BODIESS	Converting Ai in Pl	Converti		Name	Data type	Address	Retain	Acces	Write	Visibl Co
Add new device			 Jeterface 	0	0.11			CLAIRIC	conrege	1	A Flow90ConveyLinxInput	*Converting	₩68.0				
devices & networks			virtual input module 64 but		1	68 131		control input modul	NO ORDE	2	XFlow90ConveyLinxOutput	Converties O	► %Q64.0				
FlowSort_XFlow90_Demo (CPU 121			virtual autout module of by		-	00	44.122	virtual input mount.	. 110_0101		<add new=""></add>				Image: A start and a start	Image: A start and a start	V
Device configuration		-	Virtual output module 64 b	y 0	-		04127	and the opport mod	NO_ONDE								
Q Online & diagnostics																	
Program blocks																	
Add new block																	
Main [OB1]																	
Flows ort_XFlow90_Divert_IDB [
🕶 🔚 Control	10 C																
Flowsort XFlow90 Divert (F																	
System blocks																	
Technology objects																	
External source files																	
PLC tags																	
Show all tags																	
Add new tag table																	
Sefault tag table [38]																	
S XFlow_ConveyLinx [2]		•															
PLC data types																	



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

Creating sensor tags:

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

FlowSort_X_Flow90_V17	^			Name		Data type	Address	Retain	Acces	Writa	Visibl	Comment
💕 Add new device		1	-00	- x	Flow90ConveyLinxInput	"ConveyLinx_INPUT"	%168.0					
Devices & networks		2	-00	-	Sensors	Struct	%168.0		V	V		All sensor inputs
FlowSort_XFlow90_Demo [CPU 1211		3	-00		Spare1	Bool	%168.0		 Image: A start of the start of	V		Not in use
Device configuration		4	-00		Spare2	Bool	%168.1		V	V		Not in use
😼 Online & diagnostics		5	-00		Spare3	Bool	%168.2		 Image: A start of the start of	V		Not in use
🕶 🛃 Program blocks		6	-00		Spare4	Bool	%168.3		V	V		Not in use
Add new block		7	-00		Spare5	Bool	%168.4		Image: A start and a start	1		Not in use
se Main [OB1]		8	-00		Spare6	Bool	%168.5		1	V		Not in use
Flowsort_XFlow90_Divert_IDB [D	_	9	-00		Spare7	Bool	%168.6		V	V		Not in use
▼ E Control	=	10	-00		Heartbeat	Bool	%168.7		V	V		This bit toggles every 2 seconds
Flowsort_XFlow90_Divert [FB1]		11	-00		LeftAdditional	Bool	%169.0		V	V		Left sensor port state (Pin2)
System blocks		12	-00		Spare8	Bool	%169.1		Image: A start and a start	V		Not in use
Technology objects		13	-00		RightAdditional	Bool	%169.2		 Image: A start of the start of	v		Right sensor port state (Pin2)
External source files		14	-00		Spare9	Bool	%169.3		V	V		Not in use
🔻 🎑 PLC tags		15	-00		Left	Bool	%169.4		 Image: A start of the start of	×		Left sensor port state (Pin4)
🍇 Show all tags		16	-00		Spare10	Bool	%169.5		V	V		Not in use
📑 Add new tag table		17	-00		Right	Bool	%169.6		v	~		Right sensor port state (Pin4)
🎬 Default tag table [38]		18	-00		Spare11	Bool	%169.7		 Image: A start of the start of	>		Not in use
XFlow_ConveyLinx [2]		19	-00	•	SensorsPresent	Struct	%170.0		V	V		Sensor detection
 PLC data types 		20	-00		MotorTemperatureLeft	Byte	%1872		Image: A start and a start	V		The temperature of the Left motor [°C]
💣 Add new data type		21	-00	•	MotorDiagnosticsLeft	Struct	%174.0		 Image: A start of the start of	V		Left motor diagnostics
ConveyLinx_INPUT		22	-00		MotorTemperatureRight	Byte	%IB76		V	V		The temperature of the Right motor [°C]
ConveyLinx_OUTPUT		23	-00	•	MotorDiagnosticsRight	Struct	%178.0		V	V		Right motor diagnostics
 System data types 		24	-00	•	MotorStatusLeft	Struct	%180.0		 Image: A start of the start of	V	V	This is the status when the left motor is co
 Watch and force tables 		25	-00	•	MotorStatusRight	Struct	%182.0		V	V	V	This is the status when the right motor is c.
💣 Add new watch table		26	-00	► X	Flow90ConveyLinxOutput	"ConveyLinx_OUTPUT"	%Q64.0					
Force table		27		<	Add new>				V	V	V	
2001												

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

Û	₋ V1	7 → FlowSort_XFlow90_De	emo [CPU 1211C DC/DC/	DC] + PLC 1	ags 🕨 XFlow_Co	onveyLir	ıx [4]	-	∎∎×	@90_V17 → FlowSort_XFlow90_Demo [CPU 1211C DC/DC/I	DC] ▶
					4	Tags	🗉 Us	er cons	tants		
\$	🕐 (e 🕆 🕆 🖬								🕸 1월 2월 2월 🔄 📰 🗄 📰 🖉 2월	¢0 (
)	Flow	_ConveyLinx								Block interfa	
-		Name	Data type	Address	Retain	Acces	. Writa	Visibl	Com		
1	-00	 XFlow90ConveyLinxInput 	"ConveyLinx_INPUT"	%168.0							
2	-01	 Sensors 	Struct	%168.0		1	V		All se	 Block title: "Main Program Sweep (Cycle)" 	
3	-00	Spare1	Bool	%168.0		 Image: A start of the start of	V		Not i	Comment	
4	-00	Spare2	Bool	%168.1		V	V		Not i		
5	-00	Spare3	Bool	%168.2		1	V		Not i	 Network 1: 	
6	-	Spare4	Bool	%168.3		V	V		Not i	Tu to parcel or box from TU	
7	-00	Spare5	Bool	%168.4		1	V		Not i		
8	-00	Spare6	Bool	%168.5		1	V		Not i	No. 1	
9	-00	Spare7	Bool	%168.6			Image: A start and a start		Not i	Televrort V	
10	-00	Heartbeat	Bool	%168.7		V	V		This	Flow90_Divert_	
11	-00	LeftAdditional	Bool	%169.0		 Image: A start of the start of	Image: A start and a start		Left s	IDB"	
12	-00	Spare8	Bool	%(69.1		 Image: A start of the start of	Image: A start and a start		Not i	%FB1	
13	-00	RightAdditional	Bool	%169.2		1	Image: A start and a start		Right	"Flowsort_XFlow90_Divert"	
14	-00	Spare9	Bool	%169.3		 Image: A start of the start of			Not i	EN ENO	
15	-01	Left	Bool	%169.4					Left s	iDivertType gTransferOK — false	
16	-00	Spare10	Bool	%169.5					Not i	taise Reverse qTransferReady	
17	-00	Right	Bool	%169.6					Right	false iReset ToReceive Italse	
18	-00	Spare11	Bool	%169.7					Not i	%169.4 qTransferBusy false	
19	-00	SensorsPresent	Struct	%170.0					Sens	"TopPosSensor" TopPosSensor Prostion	
20	-00	MotorTemperatureLeft	Byte	%IB72					The t	%69.6 gConveyLinx	
21	-00	MotorDiagnosticsLeft	Struct	%174.0					Left	"BottomPos iBottomPos MotorRightError Halse	
22	-00	MotorTemperatureRight	Byte	%IB76					The t	Sensor GConveyLinx	
23	-00	MotorDiagnosticsRight	Struct	%178.0				1	Right	false Palaared	
24	-0	MotorStatusLeft	Struct	%(80.0					This	iParcelln	
25	-0	MotorStatusRight	Struct	% 82.0					This i	false PositionTransfer P#Q64.0	
26	-01	XElow90ConveyLinxOutput	"ConveyLinx OUTPUT"	%064.0						false	Convey
27	-01	TopPosSensor	Bool	%169.4						false iRunRight	
28	-01	BottomPosSensor	Bool	% 69.6						0 — iBeltSpeed	
29		<add new=""></add>								iBelt	
							0	0		Acceleration Distance	
										iBelt	
										Deceleration	
										0 — Distance	
										0 iLiftingSpeed	
										iLifting Acceleration	
										il iffing	
										0 — Deceleration	
										270 iProfinetDevice	
										DHIGR O	
										*XFlow90Convey ConveyLinx	
										LinxInput" Input	

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



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



Typical Parallel Splitter Cable Usage

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

Project tree 🔲 🖣	Flo	wSort_	_X_Flow90_V17 → FlowSor	t_XFlow90_Demo [CPU '	1211C DC/D	DC] ▶	PLC tag	s ▶ XF	low	_ConveyLinx [6]
Devices										
1 I I I I I I I I I I I I I I I I I I I	1	#	⇒ ₩ °° 🛍							
		XFlow	ConveyLinx							
FlowSort_X_Flow90_V17		1	lame	Data type	Address	Retain	Acces	Writa		Comment
Add new device	1	-00	 XFlow90ConveyLinxInput 	"ConveyLinx_INPUT"	%168.0					
Devices & networks	2	-00	 Sensors 	Struct	%168.0					All sensor inputs
FlowSort_XFlow90_Demo [CPU 1211	3	-00	Spare 1	Bool	%168.0					Not in use
Device configuration	4	-	Spare2	Bool	%168.1		×	 Image: A start of the start of		Not in use
Q Online & diagnostics	5	-00	Spare3	Bool	%168.2					Not in use
🔻 🙀 Program blocks	6	-00	Spare4	Bool	%168.3		×	 Image: A start of the start of		Not in use
💕 Add new block	7	-00	Spare 5	Bool	%168.4		 Image: A start of the start of			Not in use
Main [OB1]	8	-00	Spare 6	Bool	%168.5					Not in use
Flowsort_XFlow90_Divert_IDB [D	9	-00	Spare7	Bool	%168.6					Not in use
▼ E Control	10	-00	Heartbeat	Bool	%168.7					This bit toggles every 2 seconds
Flowsort_XFlow90_Divert [FB1]	11	-01	LeftAdditional	Bool	%169.0					Left sensor port state (Pin2)
System blocks	12	-01	Spare8	Bool	%169.1		×	~		Not in use
Technology objects	13	-0	RightAdditional	Bool	%169.2		 Image: A start of the start of	 Image: A start of the start of		Right sensor port state (Pin2)
External source files	14	-00	Spare9	Bool	%169.3		 Image: A start of the start of	 Image: A start of the start of		Not in use
🔻 🌄 PLC tags	15	-00	Left	Bool	%169.4		×	 Image: A start of the start of		Left sensor port state (Pin4)
🗞 Show all tags	16	-00	Spare 10	Bool	%169.5		 Image: A start of the start of			Not in use
🗳 Add new tag table	17	-00	Right	Bool	%169.6					Right sensor port state (Pin4)
🔀 Default tao table [38]	18	-00	Spare 11	Bool	%169.7					Not in use
🛬 XFlow_ConveyLinx [6]	19	-00	SensorsPresent	Struct	%170.0		1	V		Sensor detection
PLC data types	20	-	MotorTemperatureLeft	Byte	%IB72		1	 Image: A start of the start of		The temperature of the Left motor [°C]
🚔 Add new data type	21	-	MotorDiagnosticsLeft	Struct	%174.0		1	 Image: A start of the start of		Left motor diagnostics
E ConveyLinx_INPUT	22	-	MotorTemperatureRight	Byte	%IB76		×	 Image: A start of the start of		The temperature of the Right motor [°C]
ConveyLinx_OUTPUT	23	-00	MotorDiagnosticsRight	Struct	%178.0		×	 Image: A start of the start of		Right motor diagnostics
System data types	24	-00	MotorStatusLeft	Struct	%180.0		×	 Image: A start of the start of		This is the status when the left motor is co
 Watch and force tables 	25	-00	MotorStatusRight	Struct	%182.0		V	 Image: A start of the start of		This is the status when the right motor is c.
🚔 Add new watch table	26	-00	XFlow90ConveyLinxOutput	"ConveyLinx_OUTPUT"	%Q64.0					
E. Force table	27	-00	TopPosSensor	Bool	%169.4					
Eq. Functional test	28	-00	BottomPosSensor	Bool	%169.6					
oo, IO test	29	-01	ExtraSensorLeftSide	Bool	%169.0					
Dnline backups	30	1	ExtraSensorRightSide	Bool	%169.2					
🕨 📴 Traces	31	-	<add new=""></add>				1	 Image: A start of the start of		
No. and the second s										



Flowsort_XFlow90_Divert Function Block:

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



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



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

FlowSort_X_Flow90_V17 ▶ FlowSor	rt_XFlow90_Demo [CPU 1211C DC/DC/DC] + Program blocks + Main [OB1
юй най 🖈 📽 🐛 🗮 🚍 💬 🕏	3 * * * * = 😥 🕫 🐝 🕷 🕫 🗣 🐂 📢 🚱 🌚 🔒
⊣⊢⊣⊢⊸⊢ ↦ ᅼ	
 Block title: "Main Program Sweep (Cy 	cle)*
Comment	
▼ Network 1:	
▶ Tu to parcel or box from TU	
9 "Flo	4DB1 wsort X
Flow9	0_Divert_ IDB"
9	AFB1
"Flowsort_X	Flow90_Divert"
EN iDivertType	GTransferOK — false
false — Reverse	qTransferReady
false — iTopPosSensor	qTransferBusy — false
iBottomPos false — Sensor	qPosition SensorError → false
iParcelTransfer false — Released	qConveyLinx MotorRightError — false
iParcelln false — PasitionTransfo	qConveyLinx
false — iRunLeft	qProfinetError — false
false — iRunRight	P#Q64.0
0 — iBeltSpeed iBelt	ConveyLinx "XFlow90Convey Output — LinxOutput"
Acceleration 0 — Distance	
iBelt Deceleration	
0 — Distance	
0 — iLiftingSpeed iLifting	
0 — Acceleration	
0 — Deceleration	
270 — iProfinetDevice	
P#I68.0 "XFlow90Convey ConveyLinx	
LinxInput" — Input	

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

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



4.2 Function Block Inputs

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





iDivertTypeReverse:

There are two methods to divert the transport unit:

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



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

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

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

iReset:

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

iTopPosSensor and iBottomPosSensor:

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



iTuTransferReleased:

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

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

iTuInPositionTransfer:

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

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

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

iRunLeft/Right:

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

iBeltSpeed:

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

iAcceleration Distance and iDeceleration distance:

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

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

iLiftingSpeed:

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



iLiftingAcceleration and iLiftingDeceleration:

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

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

iProfinetdevice:

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

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

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

Project tree 🔲 🖣	FlowSort_X_Flow90_V17 ▶ FlowS	ort_XFlow90_[0emo [CPU 1211C	DC/DC/DC] → PLC tags _ 🖬 🖬 🗙	
Devices			📲 Tags	User constants x System constants	
1 I I I I I I I I I I I I I I I I I I I	ł –				ふ 秋 地 美 (学) - 「 三 三 三 三 二 (2 + 2 + 12 + 二)) で (
	PLC tags				Block interface
FlowSort_X_Flow90_V17	Name	Data type	Value	Comment	
Add new device	1 J None	Pip	65535		⊣⊢⊣⊢⊞ → ┛
Devices & networks	2 📮 Automatic update	Pip	0		▼ Block title: "Main Program Sween (Curle)"
FlowSort_XFlow90_Demo [CPU 1211	3 🖉 PIP 1	Pip	1		Comment
Device configuration	4 PIP 2	Pip	2		
Online & diagnostics	5 💭 PIP 3	Pip	3		 Network 1:
🔻 🛃 Program blocks	6 👷 PIP 4	Pip	4		Tu to parcel or box from TU
💕 Add new block	7 👳 PIP OB Servo	Pip	32768		
🖀 Main [OB1]	8 💭 Local-MC	Hw_SubModule	51		No. 1
Flowsort_XFlow90_Divert_IDB [D	9 💭 Local-Common	Hw_SubModule	50		74081 "Flowsort X
- Ei Control	= 10 🖉 Local-Device	Hw_Device	32		Flow90_Divert_
Flowsort_XFlow90_Divert [FB1]	11 Local-Configuration	Hw_SubModule	33		IDB"
System blocks	12 Local-Exec	Hw_SubModule	52		%FB1
Technology objects	13 🚛 Local	Hw_SubModule	49		"Flowsort_XFlow90_Divert"
External source files	14 Local-PROFINET_interface_1	Hw_Interface	64		EN ENO
PLC tags	15 Local~PROFINET_interface_1~Port_1	Hw_Interface	65		iDivertType qTransferOK — folse
🍇 Show all tags	16 Local-HSC_1	Hw_Hsc	257		Taise — Reverse qTransferReady
📑 Add new tag table	17 Local-HSC_2	Hw_Hsc	258		Taise - iReset ToReceive - Taise
🍇 Default tag table (38)	18 Local-HSC_3	Hw_Hsc	259		Talse TopPosSensor gTransferBusy Talse
🔩 XFlow_ConveyLinx [2]	19 Local-HSC_4	Hw_Hsc	260		false — Sensor SensorError — false
PLC data types	20 Local-HSC_5	Hw_Hsc	261		iParcelTransfer qConveyLinx
💕 Add new data type	21 Local-HSC_6	Hw_Hsc	262		false — Released MotorRightError — false
E ConveyLinx_INPUT	22 Local-Al_2_1	Hw_SubModule	263		false PeritionTransfer MotorLeftError Infalse
E ConveyLinx_OUTPUT	23 Local-DI_6_DQ_4_1	Hw_SubModule	264		false iPunt aft aProfinatError - false
 G System data types 	24 Local-OPC_UA	Hw_SubModule	117		false - iPunDiabt
Watch and force tables	25 Local-Pulse_1	Hw_Pwm	265		P#Q64.0
Image:	26 Local-Pulse_2	Hw_Pwm	266		iBelt Output LinxOutput
🕨 🔯 Traces	27 Local-Pulse_3	Hw_Pwm	267		Acceleration
DPC UA communication	28 E Local~Pulse_4	Hw_Pwm	268		0 — Distance
Device proxy data	29 CB_Main	OB_PCYCLE	1		Deceleration
🔠 Program info	30 Local-PROFINET_IO-System	Hw_loSystem	269		0 — Distance
PLC alarm text lists	31 January X_Flow90~Proxy	Hw_SubModule	272		0 — iLiftingSpeed
Local modules	32 Z Flow90~IODevice	Hw_Device	270		iLifting
Distributed I/O	33 X_Flow90~Interface	Hw_Interface	273		0 — Acceleration
Ungrouped devices	34 🖉 X_Flow90~Interface~Port_1	Hw_Interface	274		0 — Deceleration
Security settings	35 January X_Flow90~Interface~Port_2	Hw_Interface	275		270 iProfinetDevice
Cross-device functions	36 X_Flow90~Head	Hw_SubModule	276		
🕨 🙀 Common data	37 X_Flow90~virtual_input_module	Hw_SubModule	277		P#I68.0
Documentation settings	38 X_Flow90~virtual_output_module	Hw_SubModule	278		Linxinput" Input
 De transmissión d'una companya 					, mpar



CLInputs:

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



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



4.3 Function Block outputs



qTransferOK:

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

qTransferReadyToReceive:

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



qTransferBusy:

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

Sensor error:

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

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

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

qCLMotorRightError:

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

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

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

qCLMotorLeftError:

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

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

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



CLOutput:

Project tree 🔲 🖣	◎V17 + FlowSort_XFlow90_Demo [CPU 1211C DC/DC/DC] + PLC tags + XFlow_ConveyLinx [2] ■ ■ ■ X	@90_V17 FlowSort_XFlow90_Demo [CPU 1211C DC/DC/DC] F
Devices	Tags User constants	
	XFlow ConveyLinx	Block interface
FlowSort_X_Flow90_V17	Name Data type Address Retain Acces Writa Visibl Comment	
💕 Add new device	1 💶 🕨 XFlow90ConveyLinkinput *ConveyLink 🗒 %l68.0 🔍 🗹 🗹	┤┥┝╴┥/┝╶┥ ┝╴ ╴Ѽ╴┕→ ╶┛
A Devices & networks	2 TowyoConveyLinxOutput *ConveyLinx_0 %Q64.0	Riock title: "Main Program Sweep (Cycle)"
FlowSort_XFlow90_Demo [CPU 1211	3 <add new=""></add>	Comment
Device configuration		Commune.
😵 Online & diagnostics		 Network 1:
 Program blocks 		Tu to parcel or box from TU
💕 Add new block		
🖀 Main [OB1]		
Flowsort_XFlow90_Divert_IDB [D		*OB1 "Elevisiont X
		Flow90_Divert_
Flowsort_XFlow90_Divert [FB1]		IDB*
System blocks		%FB1
Technology objects		"Flowsort_XFlow90_Divert"
External source files		EN ENO
🔻 🚂 PLC tags		iDivertType qTransferOK —Ifalse
line and the second sec		alse – Reverse qTransferReady
💕 Add new tag table		Taise
🍇 Default tag table [38]		international in
Schow_ConveyLinx [2]		false — Seasor SensorError — false
👻 💽 PLC data types		iParcencansfer qConveyLinx
📑 Add new data type		false — Released MotorRightError — false
ConveyLinx_INPUT		false — PositionTransfer MotorLeftError — false
ConveyLinx_OUTPUT		false — iRuni eft gPrometError — Ifalse
 System data types 		felse — iBunBinht
Watch and force tables		0 - iBeltSpeed
Online backups		iBelt Output — LinxOutput"
Traces		Acceleration
OPC UA communication		0 — Distance
Device proxy data		Deceleration
Program info		0 — Distance
PLC alarm text lists		0 — iLiftingSpeed
Local modules		iLifting
Distributed I/O		U — Acceleration
Ungrouped devices		0 — Deceleration
Security settings		270 - iProfinetDevice
 Cross-device functions 		Desce o
Common data		*XFlov/90Convey Convey inv
 Image: Documentation settings 		Linxinput" Input

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

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



4.4 Flow chart





4.5 Function Block: explanation

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



Network 1: Declare errors ConveyLinx-Ai2 motor error.



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





Network 2: Declare errors ConveyLinx-Ai2 motor error.

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

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



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



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

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

Network 4: Declare sensor errors.



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

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



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

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



Network 5: Reset errors.

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

If iReset input is activated and:

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



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

Network 6: Declare position OK.



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





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



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

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

Network 8: Transfer is ready to receive new parcel.



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

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

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

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



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



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

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

Network 10: Parcel in position for lifting the divert.



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

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

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

Network 11: Run belts.



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

Network 12: Transfer has been finished.





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

Network 13: Move transfer down.



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

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



Network 14: Set speed limits.



In network 14, the speed limits are declared:

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

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

Network 15: Motor speed, acceleration and deceleration.

The values for conveylinx:						
1	<pre>#ConveyLinxOutput.MotorSpeedRight := #sBeltSpeedLimit * 1000;</pre>					
2	<pre>#ConveyLinxOutput.MotorSpeedLeft := #sLiftingSpeedLimit / 10;</pre>					
3	<pre>#ConveyLinxOutput.MotorAccelerationRight := #iBeltAccelerationDistance;</pre>					
4	<pre>#ConveyLinxOutput.MotorDecelerationRight := #iBeltDecelerationDistance;</pre>					
5	<pre>#ConveyLinxOutput.MotorAccelerationLeft := #iLiftingAcceleration;</pre>					
6	<pre>#ConveyLinxOutput.MotorDecelerationLeft := #iLiftingDeceleration;</pre>					

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

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

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

Network 16: Add static to output.

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



4.6 Program integration for ALLEN BRADLEY STUDIO 5000 V35

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

Flowsort_XFlow90_Divert	
Flowsort_XFlow90_Divert	Flowsort_XFlow90_Divert
iDivertTypeReverse	false_bool
iTopPosSensor	false_bool
iBottomPosSensor	false_bool
iReset	false_bool
iParceITransferReleased	false_bool
iParcelInPositionTransfer	false_bool
iRunLeft	false_bool
iRunRight	false_bool
iBeltSpeed	false_REAL
iBeltAccelerationDistance	false_INT
iBeltDecelerationDistance	false_INT
iLiftingSpeed	false_INT
iLiftingAcceleration	false_INT
iLiftingDeceleration	false_INT
ConveyLinxInput	X_Flow:I1
qTransferOK	false_bool
qTransferReadyToReceive	false_bool
qTransferBusy	false_bool
qPositionSensorError	false_bool
qConveyLinxMotorRightErro	r false_bool
qConveyLinxMotorLeftError	false_bool
qProfinetError	false_bool
ConveyLinxOutput	X_Flow:01

4.6.1 ALLEN BRADLEY STUDIO 5000 Hardware

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

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

Rockwell Automation's EDS Wo	Welcome to Rockwell Automation's EDS Wizard
	The EDS Wizard allows you to: -register EDS based devices.
	- change the graphic images associated with a device.
	-create an EDS file from an unknown device. - upload EDS file(a) stored in a device.
	To continue click Next
	Volgende > Annuleren



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:

This is a review of the task you want to complete.		A.
You would like to register the following device.		

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

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

AI2			C	lear	Filter	s	Hide Fi	lters 🛠
V	Module Type Category	Filters				Module Type Vendor Filters		*
V	AC Drive Device				V	Advanced Energy Industries, Inc.		
V	Analog				V	Cognex Corporation		
V	CIP Motion Converter					Danfoss		
	Communication			-		Dialight		*
_								
Ca	atalog Number	Description				Vendor	Category	
	Ai2	Ai2-5xx				Insight Automati	. Generic D	evice(keya
1								ħ

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.

Module Definition*				×
Revision: 5				
Electronic Keying: Compa	•			
Connections:				
Name		Size		Tag Suffix
PI CIO	Input:	25	INT	1 🔺 Al:11
FLCIO	Output:	27	1141	× AI:01
Select a connection 👻				
			ОК	Cancel Help

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.

		_
🔺 🚣 A2,	Thernet	
	069-L306ER FlowSort_DemoSetup	
(MAN)	Ai2 Module_10_20	

4.6.2 ALLEN BRADLEY STUDIO 5000 Software

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





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

4.6.3 ALLEN BRADLEY STUDIO 5000 CONTROLLER LINK SOFTWARE TO HARDWARE

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

Flowsort_XFlow90_Divert	
Flowsort_XFlow90_Divert	Flowsort_XFlow90_Divert
iDivertTypeReverse	false_bool
iTopPosSensor	false_bool
iBottomPosSensor	false_bool
iReset	false_bool
iParcelTransferReleased	false_bool
iParcelInPositionTransfer	false_bool
iRunLeft	false_bool
iRunRight	false_bool
iBeltSpeed	false_REAL
iBeltAccelerationDistance	false_INT
iBeltDecelerationDistance	false_INT
iLiftingSpeed	false_INT
iLiftingAcceleration	false_INT
iLiftingDeceleration	false_INT
ConveyLinxInput	?
qTransferOK	false_bool
qTransferReadyToReceive	false_bool
qTransferBusy	false_bool
qPositionSensorError	false_bool
qConveyLinxMotorRightErr	or false_bool
qConveyLinxMotorLeftErro	r false_bool
dProfinetError	faise_hool
ConveyLinxOutput	X_Flow:01

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

iLittingAcceleration iLiftingDeceleration ConveyLinxInput	י י ?	false_INT	
Enter Name Filter	Show: All T	ags	~
Name		<u>=8</u>	Data Ty; A _055C:A
			~
Show controller tag	gs n tags		
Show parameters from	other program:	_	
<none></none>	~		

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

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



4.7 Program integration for OMRON SYSMAC STUDIO V1.58.0

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



4.7.1 OMRON SYSMAC STUDIO HARDWARE AND LINK TO SOFTWARE

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

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

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

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

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

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



iveri G	lobal Variables 🗙								
Gro	up Filter 🍸 (No group)	•							
	Name	Data Type	Initial Value	AT	Retain	Constant	Network Publish	Comment	
2000	XFlow90ConveyLinxInput	ARRAY[024] OF INT					Input 🔻		
5555	XFlow90ConveyLinxOutput	ARRAY[026] OF INT					Output 🔻		

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".

📓 Tag	- 🗆 ×							
Select the	Select the variables to set.							
	Variable Name	Data Type	Size		Comment			
	▼ Input Tag							
	XFlow90ConveyLinxInput_1	ARRAY[024] OF INT	50					
	▼ Output Tag							
	XFlow90ConveyLinxOutput_1	ARRAY[026] OF INT	54					
<								
Check	Selected Items Uncheck Selected	ltems		Register	Cancel			

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

var Global Vari	iables	EtherNet/IP De	vice List	Built-in Et	herNet/IP	ection Se	×
0-	¤€0	Connecti	on				
	▼ Conn	ection					
r0.	Conn	ections/Max: 0 /	/ 32				
oto:	Г	arget Device	▲ Connec	tion Na Co	nnection I/	O Input/Ou	ıtl Ta

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:



Toolbox 👻 🖡					
Node address Model name Revision	192 . 168 . 10 10_ Ai2-5xx 5		▼ ▼		

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

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

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



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

4.7.2 OMRON SYSMAC STUDIO Software

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

Programming	
🔻 📋 POUs	
Programs	
L 📰 Functions	
▼ I Function Blocks	
L 🔤 ConveyLinxAi2_PLCIO_AOI	
∟ 🔤 Flowsort_XFlow90_Divert	

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

🔁 Data Typ	Data Types ×								
root									
Structures		Name	Base Type	1	Offset Type	1	Offset Byte	11	
Union	►	ConveyLinxAi2_Module_In_UDT	STRUCT	NJ				Τ	
Enumerated	►	ConveyLinxAi2_Module_Out_UDT	STRUCT	NJ					
	►	ConveyLinxAi2_Module	STRUCT	NJ					
	►	ConveyLinxAi2_Module_In_Raw	STRUCT	NJ					
	►	ConveyLinxAi2_Module_Out_Raw	STRUCT	NJ					

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



4.8 Program integration for MITSUBISHI GX WORKS 2 OR HIGHER

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



4.8.1 MITSUBISHI GX WORKS Hardware



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



The Basic_Parameters (Preferred_node_specification) should be configured as follows:

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

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

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

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

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

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



4.8.2 MITSUBISHI GX WORKS LINK SOFTWARE TO HARDWARE RECEIVE DATA

The Auto_Refresh settings are customer specific:

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

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

4.8.3 MITSUBISHI GX WORKS Software

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



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

4.9 Program integration for BECKHOFF TWINCAT 3.4.3147.18

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



· .		
FlowSor	t_X_I	Flow90 MAIN* 🕫 🗙 XFlow90
1		PROGRAM MAIN
= 2		VAR
3		xflow90 : XFlow90;
4		
5		END_VAR
e		
□ 1		XFlow90(
2		iDivertTypeReverse:= TRUE,
3		iTopPosSensor:= ,
4		iBottomPosSensor:= ,
5		iReset:= ,
e		iParcelTransferReleased:= ,
7		iParcelInPositionTransfer:= ,
8		<pre>iRunLeft:= ,</pre>
9		<pre>iRunRight:= ,</pre>
10		iBeltSpeed:= ,
11		iBeltAccelerationDistance:= ,
12		<pre>iBeltDecelerationDistance:= ,</pre>
13		iLiftingSpeed:= ,
14		iLiftingAcceleration:= ,
15		iLiftingDeceleration:= ,
16		iProfinetDevice:= ,
17		ConveyLinxInput:= ,
18		qTransferOK=> ,
19		qTransferReadyToReceive=> ,
20		qTransferBusy=> ,
21		<pre>qPositionSensorError=> ,</pre>
22		<pre>qConveyLinxMotorRightError=> ,</pre>
23		<pre>qConveyLinxMotorLeftError=> ,</pre>
24		<pre>qProfinetError=> ,</pre>
25		ConveyLinxOutput=>);

4.9.1 **BECKHOFF TWINCAT Hardware**

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

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

Insert Box		
Туре:	Beckhoff Automation GmbH Greve PR0Fldrive MC (DPV2 / PNI0) Miscellaneous Miscellaneous PR0FINET IO Device	Ok Cancel
		Multiple:
Name:	Box 25	

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





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



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

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

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

Left side_Motor mode -> ECO Mode

Left side_Brake mode -> Servo brake method

Right side_Motor mode -> BOOST mode

Right side_Brake mode -> Free brake method

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

4.9.2 **BECKHOFF TWINCAT Software**



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



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



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

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

GVL_ConveyLinx.ConveyorLinx_DataIn.LeftMDRDiagnostic.Overheat	1	sConveyLinxError
GVL_ConveyLinx.ConveyorLinx_DataIn.LeftMDRDisgnostic.ShortCircuit		qConveyLinxMotorLeftError
GVL_ConveyLinx.ConveyorLinx_DataIn.LeftMDRDiagnostic.Overload		
GVL_ConveyLinx.ConveyorLinx_DataIn.LeftMDRDiagnostic.Stalled		
GVL_ConveyLinx.ConveyorLinx_DataIn.LeftMDRDiagnostic.OverVoltage		
GVL_ConveyLinx.ConveyorLinx_DataIn.LeftMDRDiagnostic.LowVoltage		



5. IO and Functional tests.

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

5.1 IO test.

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

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

	i	Name	Address	Display format	Monitor value	Modify value	9	Comment	Tag
FlowSort_X_Flow90_V17	1	"TopPosSensor"	169.4	Bool	•				
🚔 Add new device	2	"BottomPosSensor"	%169.6	Bool					
Devices & networks	з		<add new=""></add>						
FlowSort_XFlow90_Demo [CPU 1211									
Device configuration									
Q Online & diagnostics									
Program blocks									
Technology objects									
External source files									
PLC tags									
PLC data types									
 Watch and force tables 									
💕 Add new watch table									
Force table									
Functional test									
iO test									
Online backups									
Traces									

Go online and press the monitor all values:

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

1	≝ ≝ # II 9, 9, 9 1 9 1									
	i	Name		Display format	Monitor value	Modify value	4	Comment		
1		"TopPosSensor"		Bool	FALSE					
2		"BottomPosSensor"		Bool	TRUE					
3			<a< th=""><th></th><th></th><th></th><th></th><th></th></a<>							

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

If necessary, check the cabling and connector.

5.2 Functional test.

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

It is necessary to open the functional test watch table:



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

Project tree 🔲 🖣	FlowS	iort_X_Flow90_V17 FlowSort_XFlow90_Demo [CP	U 1211C DC/E	OC/DC] + Watch and	force tables 🕨 F	unctional test				
Devices										
191 🗉 🖬	· · · · · · · · · · · · · · · · · · ·									
	i 1	Name	Address	Display format	Monitor value	Modify value	9	Comment	Tag comment	
 FlowSort_X_Flow90_V17 	1	*XFlow90ConveyLinxOutput*.MotorControlLeft.Run	%Q71.0	Bool				Lift transfer up/down	motor Run = 1 / motor Stop = 0	
Add new device	2	*XFlow90ConveyLinxOutput*.MotorControlRight.Run	%Q73.0	Bool				Move belt right	motor Run = 1 / motor Stop = 0	
d Devices & networks	3	*XFlow90ConveyLinxOutput*.MotorControlRight.Direction	%Q72.0	Bool				Move belt left	0 = Configured direction /1 = opposite to	
FlowSort_XFlow90_Demo [CPU 1211	4	<u> </u>	<add new=""></add>							
Device configuration										
Online & diagnostics										
Program blocks										
Technology objects										
External source files										
PLC tags										
E PLC data types										
 Watch and force tables 										
Add new watch table										
Force table										
Functional test										
iO test										

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

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

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



Go online and press the monitor all values:

Project tree 🔲 🖣	FlowSort	_X_Flow90_V17 ► FlowSort_XFlow90_Demo [CF	PU 1211C DC/DC	C/DC] 🕨 Watch and	force tables Functional test		
Devices							
1	1	## 😼 🗓 🕫 1. 18 19 😤 😤 😭					
	i	Name	Address	Display format	Monitor value Modify value	9	Comment
 FlowSort_X_Flow90_V17 	1	"XFlow90ConveyLinxOutput".MotorControlLeft.Run	%Q71.0	Bool 💌			Life transfor unidoum
📫 Add new device	2	"XFlow90ConveyLinxOutput".MotorControlRight.Run	%Q73.0	Bool	Modity	· ·	Modify to 0 Ctri+P3
📥 Devices & networks	3	*XFlow90ConveyLinxOutput*.MotorControlRight.Direction	%Q72.0	Bool	Monitor all	Ctrl+T	Modify to 1 Clifer 2
FlowSort_XFlow90_Demo [CPU 1211	4		<add new=""></add>		Monitor now		Modifywith trigger Ctrl Shift F9
Device configuration					🔿 Insert row	Ctrl+Enter	Enable peripheral outputs
Online & diagnostics					Add row	Alt+Ins	~
Program blocks					A Insert comment line		
Technology objects					M		
External source files					X Cut	Ctrl+X	
PLC tags					E Copy	Ctrl+C	
PLC data types					aste	Ctn+v	
 Watch and force tables 					X Delete	Del	
🌁 Add new watch table					Rename	F2	
Force table					Go to definition	Ctrl+Shift+D	
Go, Functional test					Cross-references	F11	
UO test					Cross-reference information	n Shift+F11	



During monitoring, right click on monitor value of the

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

The next test is the belt motor, activate:

"XFlow90ConveyLinxOutput".MotorControlRight.Run -> Modify -> Modify to 1. The belt motor should run.

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

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

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

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

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



6. Tips and tricks/lessons learned

Add information of the files for the rest of the platforms