4. Sample Task Function Block for Conveyor Control

When blocks are to be generated that are working in any program like a "Black Box" as it were, they have to be programmed by using variables. In this case, the following rule applies: that in these blocks, no absolute-addressed inputs/outputs, flags etc. must be used. Within the block, only variables and constants are used.

In the example below, a function block is to be generated with a variable declaration containing a conveyor control that is dependent on the operating mode.

With button 'S1', the operating mode 'Manual' and with button 'S2' the operating mode 'Automatic' can be selected.

In the operating mode 'Manual', the motor is switched on as long as button 'S3' is operated, whereby button 'S4' must not be operated.

In the operating mode 'Automatic', the conveyor motor is switched on with button 'S3' and switched off with button 'S4' (break contact).

Assignment list:

Symbol	Comment
S1	Button operating mode Manual S1 NO
S2	Button operating mode Automatic S2 NO
S3	On button S3 NO
S4	Off button S4 NC
M1	Conveyor motor M1
	Symbol S1 S2 S3 S4 M1

Note: The Off button S4 is a break contact here in order to ensure wire break safety. That means: if there is a wire break at this button, the system stops automatically. Otherwise, it could not stop if there were a wire break. For that reason, in control engineering all Stop buttons, Off buttons/switches have to be designed as break contacts.

5. Programming the Conveyor Control for the SIMATIC S7-1200

The project is managed and the components are programmed with the software **'Totally Integrated Automation Portal'**.

Here, under a uniform interface, the components such as controller, visual display and networking of the automation solution are set up, parameterized and programmed. Online tools are provided for error diagnosis

In the steps below, a project can be set up for the SIMATIC S7-1200 and the solution for a task can be programmed:

1. The central tool is the 'Totally Integrated Automation Portal' that we call here with a double click (\rightarrow Totally Integrated Automation Portal V11)



 Programs for the SIMATIC S7-1200 are managed in projects. Such a project is now set up in the portal view → Create new project → FB_conveyor → Create)

VA Siemens			- a x
			Totally Integrated Automation PORTAL
Start 🦓		Create new project	
Devices & Antonia Contraction of Antonia Contractiono Contractio Contraction of Antonia Con	 Open existing project Create new project Migrate project Close project Close project Welcome Tour First steps Installed software Help User interface language 	Project name: Path: Author Comment:	FB_conveyor C:DATAIDo_TA4Portal
Project view			

3. **'First Steps'** are suggested regarding the configuration. First, we want to **'Configure a device'**. (\rightarrow First Steps \rightarrow Configure a device)



4. Next, we 'Add new device' with the 'Device name conveyor control'. To this end, we select from the catalog the 'CPU1214C' with the matching order number. (\rightarrow Add new device \rightarrow conveyor control \rightarrow CPU1214C \rightarrow 6ES7 \rightarrow Add)



5. Now, the software automatically switches to the project view with the opened hardware configuration. Here, additional modules from the hardware catalog (to the right) can be added and in the **'Device overview'**, the addresses of the inputs and outputs can be set. The integrated inputs of the CPU have the addresses %I0.0 to %I1.5 and the integrated outputs have the addresses %Q0.0 to %Q1.1 (\rightarrow Device overview \rightarrow DI14/DO10 \rightarrow 0...1)



6. So that the software later accesses the correct CPU, its IP address and the subnet mask have to be set.

 $(\rightarrow$ Properties \rightarrow General \rightarrow PROFINET interface \rightarrow IP address: 192.168.0.1 \rightarrow Subnet mask: 255.255.255.0)



7. Since modern programming is not carried out with absolute addresses but with variables, the **global PLC tags** have to be specified here.

These global PLC variables are descriptive names with comments for those inputs and outputs that are used in the program. Later, during programming, this name is used to access the global PLC tags.

These global tags can be used in the entire program, in all blocks.

To this end, select in Project Navigation the 'controller_conveyorCPU1214C DC/DC/DC]' and then 'PLC tags'. Open the table 'PLC tags' with a double click and enter the names for the inputs and outputs as shown below (\rightarrow controller_conveyor[CPU1214C DC/DC/DC]' \rightarrow PLC tags \rightarrow PLC tags)

W	Siemens - FB_conveyor										_ # X
P	roject Edit View Insert Online Options Ti	ools Window Help							Totall	Integrated Automatio	n
	📑 🎦 🔚 Save project 📕 🐰 💷 🛅 🗙 🏷 🗄	(#± 🖬 🖥 🖪 🖬 🔡	📮 💋 Go online 🖉 Go offi	line 🛵 🖪	🖪 🗶 🚽				Totali	POR	TAL
	Project tree 🛛 🕄 🖣	FB_conveyor ► controller	_conveyor [CPU 1214C D	C/DC/DC] •	PLC tags 🕨	Default 1	tag tabl	e [18]		- * 1	XK
	Devices							🕣 Tags	User constants	System constants	5
	1300 B	🥩 🥐 🖶 🕾 🗽									Tas
		Default tag table									ks
j.	▼ T FB conveyor	Name	Data type	Address	Retain	Visible	Acces	Comment			_
Ē	Add new device	1 💶 S1	Bool	%E0.0				pushbutton man	ual mode (no contact)		
1 B	B Devices & networks	2 4 52	Bool	%E0.1				pushbutton auto	matic mode (no contact)		Lib
	controller_conveyor [CPU 1214C DC/DC/DC]	3 📲 S3	Bool	%E0.2				pushbutton conv	eyor ON (no contact)		rari
Ľ	The vice configuration	4 💶 \$4	Bool	%E0.3				pushbutton conv	eyor OFF (nc contact)		es
	Q. Online & diagnostics	5 📲 M1	Bool	3 %A0.2				motor conveyor l	pelt M1	1	
	Program blocks	6 <add new=""></add>					4			1	
	Technology objects										
	External source files										
	🕶 🌄 PLC tags										
	a Show all tags										
	Add new tag table										
	🗳 Default tag table [18]	M1						O Proper	ties tilleto (1)	Diagnostics	
	C data types							Shoper		Diagnostics	1000
	Watch and force tables	General									
	Program info	Tag	Ten								^
	Text lists		rag						_		=
	Local modules		General								
	Common data										
	Documentation settings			Name:	M1						
	Languages & resources			Data type:	Bool						
	Online access		1	Address	8.40.2						
	SIMATIC Card Reader			Audress.	7640.2						
			,		Retained						
		1		Comment:	motor conve	yor belt					
	✓ Details view		Time stamp								
					5/45/2042						
	Name Details		D	ate created:	5/15/2012 12	18 PM					~
		Default tag t	d:	et modified	5/15/2012 12	22.814					
		Derault tag t							The project FB_	conveyor was saved suc	

8. To generate the function block FB1, first select 'controller_conveyor[CPU1214C DC/DC/DC]' in project navigation, and then 'Program blocks'. Now, double click on 'Add new block' (\rightarrow controller_conveyor[CPU1214C DC/DC/DC]' \rightarrow Program blocks \rightarrow Add new block)

M Siemens - FB_conveyor										-	_ # X
Project Edit View Insert Online Options T	ools Window Help								Tetall	Integrated Automation	
📑 🛅 🔚 Save project 🏭 🐰 🗐 🗐 🗙 🌖 🗄	et 🖬 🗄 🗵 🖬 🖳 🛱	🖌 💋 Go online 💋 Go d	offline	III ×					rotany	PORT	AL
Project tree	FB_conveyor > controller_	conveyor [CPU 1214C	DC/DC/DC] >	PLC tags	▶ Def	ault ta	g table	e [18]		_ # =	× (
Devices								🕣 Tags	User constants	System constants	1
800 3	🥩 🔗 🖶 약 ûx									E	Tas
2	Default tag table										ks
FB_conveyor	Name	Data type	Address	Re	tain Vi	isible.	Acces	Comment			
Add new device	1 🚭 S1	Bool	%E0.0					pushbutton man	ual mode (no contact)		L
Devices & networks	2 43 52	Bool	%E0.1					pushbutton auto	matic mode (no contact)		Lib
controller_conveyor [CPU 1214C DC/DC/DC]	3 😋 \$3	Bool	%E0.2					pushbutton conv	evor ON (no contact)		rar
Device configuration	4 43 54	Bool	%E0.3					pushbutton conv	eyor OFF (nc contact)		es
V. Online & diagnostics	5 📲 M1	Bool	₩A0.2	-				motor conveyor b	elt M1		
	6 <add new=""></add>						V				
Add new block											
- Main [OB1]											
Technology objects											
External source files											
PLC tags											
PLC data types	MI							Propert	ine 🐴 Info (D) 0	Disgnostics	
Watch and force tables	Contract of the second s									Diagnostics	
Program info	General										
Text lists	Tag										^
Local modules		Tag							_		=
Common data		General									
 Documentation settings 											
Languages & resources			Name:	M1					1		
Online access			Data type:	Bool				10			
SIMATIC Card Reader		•		A 10 D					2		
			Address:	%A0.2					1		
		1		Retain	ned						
			Comment:	motor co	onveyor b	belt M1	_				
	-	Time stamn									
Details view		. Inc stamp									
			Date created:	5/15/2012	2 12:18 PI	M					
Name			Lass modified	E/1E/2012	12.22.8	4.4		1			~
< Portal view 🔛 Overview	Default tag t								💙 The project FB_r	conveyor was saved suc	

9. In the selection, select 'Function block (FB)' and assign the name 'conveyor'. As programming language, we specify function block diagram 'FBD'. Enumeration is automatic. Since this FB1 is called later with the symbolic name anyhow, this number is no longer that important. Accept the input with 'OK'. (\rightarrow Function block (FB1) \rightarrow conveyor \rightarrow FBD \rightarrow OK)

Name: conveyor Image: FBD Image: FBD Number: 1 Image: Image: Image: </th <th></th>	
Language: FBD Number: 1 Organization block Block access: Optimized Function block Description: Function blocks are code blocks that s]
Automatic Block access: Optimized Standard - Standard - Description: Function blocks are code blocks that s	 ▼
Description: Eunction blocks are code blocks that s Function blocks are code blocks that s	compatible with S7-300/400
Function	tore their values permanently in instance data fter the block has been executed.
Data block More	
	OK Const

10. The block '**conveyor[FB1]**' will be opened automatically. But before we can write the program, we have to declare the block's interface.

When the interface is declared, the local variables -known only in this block- are specified.

The variables are divided into two groups:

• Block parameters that generate the interface of the block for the call in the program.

Туре	Name	Function	Available in
Input parameters	Input	Parameters whose values the block reads	Functions, function blocks and some types of organization blocks
Output parameters	Output	Parameters whose values the block writes	Functions and function blocks
In/out parameters	InOut	Parameters whose value the block reads when called, and after processing writes to the same parameter	Functions and function blocks

• Local data that is used for storing intermediate results

Туре	Name	Function	Available in
Temporary local data	Temp	Variables that are used for storing temporary intermediate results. Temporary data is retained for one cycle only.	Functions, function blocks and organization blocks
Static local data	Static	Variables that are used for storing static intermediate results in instance data blocks. Static data is retained -also over several cyclesuntil it is written anew.	Function blocks

11. To declare l	ocal variables, the following variables are needed for our example.
Input:	
manual	Here, the signal for selecting the operating mode Manual is entered
automatic	Here, the signal for selecting the operating mode Automatic is entered
on	Here, the start signal is entered
off	Here, the stop signal is entered
Output:	
motor	Here, the output signal for the output conveyor motor is written

Static (exists only in the function blocks FB):

memory_automatic Here, the preselected operating mode is stored memory_motor Here, we store when the motor was started in the Automatic mode

All variables are of the type 'Bool'; that means binary variables that only can have the status '0' (false) or '1' (true).

In this example, it is important to note that the status of the two variables 'memory_automatic' and 'memory_motor' has to be stored over a longer period of time. For that reason, the variable type '**Static'** has to be used here. This variable type in turn exists only in a function block FB. For the sake of clarity, all local variables should also be provided with a sufficient comment.

The Siemens - FB_conveyor									
Project Edit View Insert Online Options T	īools	W	indow Help						
📑 📑 🔚 Save project 昌 🐰 🗉 🚡 🗙 🌖	6	1 ±	₮ % 🖸 🖸 🗎 🗯	ダ Go online 🚀 Go o	ffline 🔥 🖪 🛛	🖡 🗶 🖃 💷			
Project tree 🔲 🖣	FB	_co	nveyor 🕨 controller_con	veyor [CPU 1214C	DC/DC/DC] + F	Program blocks	• conve	yor [FB1] 🛛 🗕 🗖	∎×
Devices	Γ								
		ай н	X 🥩 🧈 🐁 🛱 🚍 🕞	22 ± 22 = 50	ر ه که '=				
5	1	Int	erface						-
▼] FB_conveyor			Name	Data type	Default value	Retain	Visible in	Comment	
Add new device	1	-	✓ Input						^
Devices & networks	2	-00	 manual 	Bool	false	Non-retentive		signal select manual mode	=
controller_conveyor [CPU 1214C DC/DC/DC]	3	-	 automatic 	Bool	false	Non-retentive		signal select automatic mode	
Device configuration	4	-	 on 	Bool	false	Non-retentive		start signal	
🖳 😟 Online & diagnostics	5	-	 off 	Bool	false	Non-retentive		stop signal	
🕶 🔜 Program blocks	6	-	✓ Output						
Add new block	7	-00	 motor 	Bool	false	Non-retentive		write signal to motor conveyor	r
Hain [OB1]	8	-	✓ InOut						
conveyor [FB1]	9	-00							
Technology objects	10	-	mem_automatic	Bool	false	Non-retentive		memory bit mode selection	
External source files	11	-	mem_motor	Bool	false	Non-ret		memory bit motor conveyor O	N
PLC tags	12	-	▼ Temp						
PLC data types	13	8	Add new>						*
Watch and force tables		<				1			>
Program info				1-1 (0) (0)					
Text lists	Ľ	•	>=1 [[]0 ->	1-1 -1×F					
Local modules	-	Blo	ck title:						^
🕨 🙀 Common data		Com	ment						20
Documentation settings	1								=
Languages & resources	-		Network 1:						

12. After the local variables have been declared, the program can now be entered by using the variable names (variables are identified with the symbol '#'). For the example in FBD, it could look like this:

Program in function block diagram (FBD):

▼ BI	ock title: conve	eyor control					
Con	nment						
•	Network 1: n	nemory bit mo	de selection	n			
		#mem	_automatic				
			SR				
	#automa	atic — S					
	#111d1		Q.	-			
•	Network 2: n	nemory bit sta	rt motor cor	nveyor in automatic			
					#mem_mot	or	
			>=1		SR		
	#mem_autom	#off — atic — 0 »*		#or	n — S	0-	
	#mem_autom	a die 🛶 🦗				Q -	
•	Network 3:	motor convey	or				
			&				
	#	≠on —			>=1		#motor
		#off 0		#mem_moto	r —		=
	#mem_automa	atic 🗝 🔆					-

Program in ladder diagram (LAD):



13. Next, right-click on the block 'Main[OB1]'.

Then, under 'Switch programming language', select the function block diagram 'FBD'.



14. Now, the block "conveyor" has to be called from the program block Main[OB1]. Otherwise, the block would not be processed.

A double click on 'Main[OB1]' opens this block. (\rightarrow Main[OB1])



15. Now, you can drag the block **"conveyor[FB1]**" with Drag&Drop to Network 1 of the block Main[OB1]. (\rightarrow conveyor[FB1])

M Siemens - FB_conveyor			
Project Edit View Insert Online Options To	Fools Window Help • C* ± 🖬 🗟 🖪 🕅 🛱 🛤 🗛	🖌 Go online 🦨 Go offline 🔥 🖪 🖪 🥵 🗙 🚽 🗍	
Project tree	FB_conveyor → controller_com	veyor [CPU 1214C DC/DC/DC]	_ # = ×
Devices			
B 0 0 B	ki ki 🖻 👘 🍬 🗖 🚍 💬	28±28⊟1266491⊑1=&™	
5	Interface		
▼ □ FB conveyor	Name	Data type Comment	
Add new device	1 - Temp		
Devices & networks	2		
➡ ☐ controller_conveyor [CPU 1214C DC/DC/DC]			
Device configuration			
😨 Online & diagnostics			
🕶 🛃 Program blocks	· · · · · · · · · · ·	J-1 - (-> - (->	
Add new block			
Hain [OB1]	▼ Block title: *Main Program Sv	weep (Cycle)*	
Conveyor [FB1]	Comment		
Technology objects			
External source files	Network 1:		
PLC tags			
PLC data types			
Watch and force tables	10- GOA	nveyor [FB1]	
Program info			
Text lists			
🕨 🧊 Local modules			

16. Since we are dealing with a function block, it has to be provided with memory. In SIMATIC S7-1200, data blocks are provided as memory. Such an assigned data block is called **Instance Data block**.

Here, it is to be specified and generated ' automatically '. (\rightarrow Automatic \rightarrow	OK)
---	-----

Call options	×
Single	Data block Name <u>conveyor_DB</u> Number Manual Automatic The called function block saves its data in its own instance data block. <u>More</u>
	OK Cancel

Highlight the default tag table.

VA Si	emens - FB_conveyor														
Proje	ct Edit View Insert	Online Options	То	ols	W	índov	N H	elp							
	📑 🔚 Save project 📑	X 🗉 🖬 🗙 🛰	D±	C	±				6 回	G	🥑 Go	online	Go Go	offline	A2 🖪 🖪
P	roject tree		•	FB_	_co	nvey	/or ▶	COI	itrolle	er_cor	iveyo	or [CPL	J 1214(DC/DC] + Pro
ſ	Devices														
	0.0	-	F.	1.3	š.	X ≡	ik ⊒ik	80				+ 👳	- 87	60	6. 13 ' '-
				1.00	Int	orfa		1 .00		- 12			- 04		*** • : = -
Ĕ.				-	mu	ena	ce					Dete			Comment
	controller_conveyor [CPU 1214C DC/DC	2		-	Nam	ne Ta una					Data t	уре		Comment
<u>e</u>	Device configurati	on	26 -	1			iemp								a
2	Voline & diagnost	ics		2		-	40</td <td>ia ne</td> <td>W></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>0</td>	ia ne	W>						0
Ū.	Program blocks														
ā.	Add new block														
	- Main [OB1]			_	-	_				1	1.	-	-		hasted have
		011		8		> = 1	??	н	-01	\rightarrow	-[=]	-(s)-	-(B)-		
	Conveyor_DB [L			-						-	1000	15 1.3			
	 External source fill 	5	-	•	DIC	OCKI	iue:	Ma	in Prog	gram S	weep	(Cycle))		
		15			.om	men									
	Chausell to an			_		Mat	work	1.							
	Add new tag ta	bla				Net	WUIK	••				200			
	Add new tag ta		-									co	nveyor_[DB"	
		GIID	~							EN			conveyor		-
<		>						1	alse 🗕	man	ual				
~	Details view								alse -	auto	matio	:			100
									alse -	off					FNO
	Name	Details	1												
-	M1	%A0.2	~												
E	OB Main	1		•		Net	work	2:							
j.	PROFINET_interface_1	64													
	Pulse_1[PTO/PWM]	266													
e	Pulse_2[PTO/PWM]	267													
-	S1	%E0.0	=												
-00	S2	%E0.1													
-	53	%E0.2													

17. In OB1, we now connect the input variables and the output variable with the PLC tags shown here.

To this end, just drag the PLC tags to the block variables.

By clicking on Save project, the project is saved. (\rightarrow "S1" \rightarrow "S2" \rightarrow "S3" \rightarrow

"S4" \rightarrow "M1" \rightarrow \square Save project



Important!

The Off button S4 is a break contact (NC) and has to be negated at the block during wiring; i.e., the Off function in the block is pending when the Off button S4 is operated and thus no signal is pending at terminal %I0.3.

18. To load your entire program to the CPU, first highlight the folder 'controller conveyor' and then click on the symbol \square Load to device. (\rightarrow controller_conveyor \rightarrow \square)



19. If you omitted specifying the PG/PC interface beforehand, a window is displayed where you can still do this.

 $(\rightarrow PG/PC \text{ interface for loading} \rightarrow Load)$

	Device	Device type	Туре	Add	ress	Subnet	
	controller_conveyor	CPU 1214C DC/D	PN/IE	192.	168.0.1		
		Tyr	oe of the PG/PC PG/PC Connection 1 1st	interface: interface: to subnet: t gateway:	PN/IE Realtek PC (local) PN/IE	Cle GBE Family C	• • • • •
					_		
	Accessible devices in	target subnet:	Tupe	Ad		Show all accessib	ole devi
	Accessible devices in Device controller press	target subnet: Device type CPU 1214C DC/D	Type . PN/IE	Ad	Idress	Show all accessib Target device controller pr	ole devi
Flash LED	Accessible devices in Device controller_press 	target subnet: Device type CPU 1214C DC/D —	Type PN/IE PN/IE	Ad 19 Ac	Idress 12.168.0.1 cess address	Show all accessib Target device controller_pro 	ole devi
Flash LED	Accessible devices in Device controller_press 	target subnet: Device type CPU 1214C DC/D -	Type PN/IE PN/IE	Ad 19 Ac	idress 12.168.0.1 cess address	Show all accessit	ess <u>R</u> efresh

20. Click on 'Load' once more. In a window, the status is indicated during loading. (\rightarrow Load)

tatus	1	Target	Message	Action
† []	0		Ready for loading.	
	0	- Stop modules	All modules will be stopped for downloading to device.	Stop all
	0		Download to device is not possible as long as the module "controller_conveyor" is in RUN.	
	0	 Device configurati 	Delete and replace system data in target	Download to device
	0		Delete and replace system data for "controller_conveyor" in the target system?	
	0	👻 Software	Download software to device	Consistent downlo
	0	Download to d	Objects that do not exist online.	
	0	Overwrite onli	Objects that exist online and will be overwritten.	
	0	Delete online	Objects that only exist online and will be deleted.	

21. If loading was successful, it is shown in a window. Click on 'Finish'. (\rightarrow Finish)

itatus		Target	Message Downloading to device completed without error.	Action
•	4	Start modules	Start modules after downloading to device.	🛃 Start all

22. Now, start the CPU by clicking on the symbol

🗿 🎦 🔒 Save project 🔳 🐰 💷 🗊	× 5± (** 🖬 🖥 🛄 🖬 🖢	🛛 🙀 💋 Go online 🧋	🛿 Go offline 🕌 🌆 🖪 🔛	
Project tree	💷 🖣 F	B_conveyor → control	ler_conveyor [CPU 1	214C DC/DC/DC] Program blocks Main [O	B1] _ 🖬 🖬 🔪
Devices				Surreit	
B 0 0	B	ый на 🛒 👻 🏍 🔚	들 🗩 📲 ± 🖓 🗄	🗊 🍋 💊 '= '= 🐣 🥸	3
				Block interface	
FB_conveyor Add new device	^	& >=1 [??]	ı <u>⊷</u> -{=] -(s)		
Devices & networks		Block title: "Main Pr	ogram Sweep (Cycle)*		[
controller_conveyor [CPU 1214C Device configuration	DC/DC	Comment			
Q Online & diagnostics		Network 1			
- 🙀 Program blocks		Network I		223	
Add new block	=		*conv	DB1 ever DB"	
- Main [OB1]			%	FB1	
conveyor [FB1]			"cor	iveyor"	
conveyor_DB [DB1]		9/50.0	EN		
Technology objects		"S1"	— manual		
External source files		%E0.1			
🕶 🚂 PLC tags		"S2"	automatic		
a Show all tags		%E0.2	0.0	9440.2	
Add new tag table		%E0.3	011	motor - "M1"	
💥 Default tag table [18]		"S4"	o off	ENO —	
PLC data types					
Watch and force tables		-	NCO O	11.00 1.1.4	
Sine Descent info		▼ 51 "co"	%E0.0	pushbutton manual mode (no contact)	
	/	"53"	%E0.2	pushbutton conveyor ON (no contact)	
Details view		"M1"	%A0.2	motor conveyor belt M1	
		"\$4"	%E0 3	pushbutton conveyor OFE (nc contact)	

23. With 'OK', confirm the question whether you actually want to start the CPU. (\rightarrow OK)

Run			×
	Are you sure CPU to RUN I	you want to ch node?	ange the
		ок	Cancel

24. By clicking on the symbol P Monitoring On/Off, you can, during the program test, observe the status of the input and output variables at the block "conveyor", but also the program execution in

the block "conveyor" (\rightarrow conveyor[FB1] \rightarrow

Options Tools			
V D+C			Totally Integrated Automation
	conveyor > controller_conveyor [CPU 1214C DC/DC/DC] > Program blocks + conveyor [FB1]	_ # = ×	Testing
			Options
a	(x ≠ ≠ + = = = = = = = = + + + + + + + + +		=
	Block interface		✓ CPU operator panel
M 🗠 N	trigger applied.	14	controller_conveyor [CPU 1214C D.
	>=1 [77] -I -oI L→ -[+] -(5)(8)-		RUN / STOP RUN
: DC/		0	ERROR STOP
-	Network 1: memory bit mode selection		MAINT MRES
•	BUE SR #automatic SR #automatic SR #memory bit start motor conveyor in automatic BUE SR #ord SR #ord SR #mem_sutomatic SR BUE SR #mem_sutomatic SR #mem_sutomatic SR BUE SR #ord SR #mem_sutomatic SR	_	< III > ✓ ✓ Call environment ✓ Call environment ✓ Call environment ✓ Change _ ✓ Breakpoints
× ×	FALSE FALSE FOR	nostics	Fast mode Fast mode Fastie outrout in run PLC register Call hierarchy Main (OB1) - NW1
			Image: applied. Image: applied. Image: applied. Image: applied.

25. Since our block "conveyor" was generated according to the rules for standard blocks (no use of global variables within the block!!!!!), it can now be used and called any number of times.

Below, an expanded PLC tag table is shown, with the inputs and outputs for two conveyors.

1)efa	ult tag table						
	-	Name	Data type	Address	Retain	Visible	Acces	Comment
1		S1_CONVEYOR1	Bool	%E0.0				conveyor1 pushbutton manual mode (no contact)
2	-	S2_CONVEYOR1	Bool	%E0.1				conveyor1 pushbutton automatic mode (no contact)
3	-	S3_CONVEYOR1	Bool	%E0.2				conveyor1 pushbutton conveyor ON (no contact)
4	-	S4_CONVEYOR1	Bool	%E0.3				conveyor1 pushbutton conveyor OFF (nc contact)
5	-	M1_CONVEYOR1	Bool	%A0.2				conveyor1 motor conveyor belt M1
6		S1_CONVEYOR2	Bool	%E0.4				conveyor2 pushbutton manual mode (no contact)
7	-	S2_CONVEYOR2	Bool	%E0.5				conveyor2 pushbutton automatic mode (no contact)
8	-	S3_CONVEYOR2	Bool	%E0.6				conveyor2 pushbutton conveyor ON (no contact)
9		S4_CONVEYOR2	Bool	%E0.7				conveyor2 pushbutton conveyor OFF (nc contact)
10		M1_CONVEYOR2	Bool	%A0.3				conveyor2 motor conveyor belt M1
11		<add new=""></add>						

26. Now, the block **"conveyor**" can also be called twice in OB1, with different wiring respectively. For each call, another instance data block is specified.

Program in function block diagram (FBD):



Program in ladder diagram (LAD):



Network 2: call program conveyor 2



Two conveyors separated from each other can now be controlled by means of the same conveyor block. Only another instance data block has to be assigned for each call.

