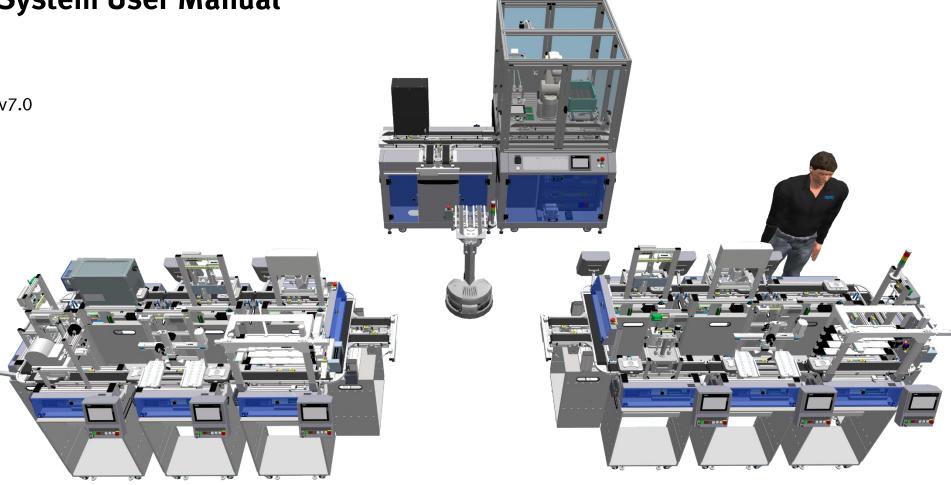


CIROS Festo CP-System User Manual

Based on CIROS v7.1.7

Can be used for v6.4.6 and v7.0 with few variations.





1. Introduction to CIROS

(→)

- Overview
- Installation
- License
- Application Scenarios
- Help Menu
- Keyboard Shortcuts
- Options Setting
- Window's Size

2. Introduction to CIROS Model (\rightarrow)

- Collaborative Working
- Model's Structure
- Elements and Coordinate Systems
- Create a Model
- Window's Layout
- Toolbar
- View and Edit Mode
- Standard Views
- View Used when Creating CIROS Model
- Snapping into Place
- Floor and Background
- Import and export data
- Teacher mode
- Simulation Control in Model Window



3. Introduction to Festo CP-System

 (\rightarrow)

• Building Blocks

- Product
- Terms and Definitions
- Standard Part Numbers
- Group and Utilities
- Carriers
- Resources and Buffers

4. CP-System Model Libraries (\rightarrow)

- Open Model Libraries Window
- Model Library Festo CP System
- Configuration in Properties Section CP System
- Sources and Sinks
- Adding New Libraries to CIROS



5. Virtual Commissioning with MES4 (\rightarrow)

- Steps to Virtual Commission with MES4
- Steps to Virtual Commission with FactoryViews
- Steps to Virtual Commission with Robotino
- Synchronise CIROS Parts in Storage with MES4 Buffers
- Running CIROS and MES4 on Different PCs
- Running CIROS and Fleet Manager on Different PCs
- Terms and Definitions in MES4
- MES4 Communication Interface
- Terminology in MES4 Messages
- MES4 Service Requests
- Festo MES4 Interface
- Use Case: Update Resource Status with MES Controller
- Message Request from CP System to MES4
- MES Communication Flow Chart
- Message Request from CIROS to Fleet Manager

6. Virtual Commissioning with Soft PLC (\rightarrow)

- Scenario Overview
- Process Summary
- Preparing a CIROS Model
- Starting a PLCSIM Instance
- Creating the Hardware Configuration and IO Tags in TIA Portal
- Configuring the Interface
- Common Issues
- Remote Connection between CIROS and PLCSIM Advanced



7. Simulation (\rightarrow)

- Simulation Kernel
- Reduce Simulation Computing Requirement
- Code Sequence Trace
- Visualising Sensor Data
- Data Logging
- Simulation Control in CP System

8. Python (\rightarrow)

- Python in Model Libraries
- Python Installed but Not Working
- Python Scripts in CIROS
- Built-In Function List
- CP System Construction Helper
- Use Case: Common TCP/IP Communication



9. OPC UA Interface (\rightarrow)

• CIROS as OPC UA Client

11. Robot Programming (1) (\rightarrow)

- Mitsubishi Industrial Robot
- Layout and Windows
- CP-F-RASS
- Steps to Configure CP-F-RASS for Simulation
- Steps to Simulate CP-F-RASS
- Simulate Real Robot Program in CP-F-RASS Model
- CP-F-RASS Robot Programming
- Move Robot Manually
- Mount and Release a Gripper Manually
- TCP Tracking
- View TCP Coordinate
- Robot Workspace
- Collision Detection
- Connect to Robot Controller
- Online Information from Robot Controller



11. Robot Programming (2) (\rightarrow)

- Create / Load Robot Controller Backup
- Upload / Download Robot Programs
- Online Teach-In
- Get Actual Robot Data with Built-In Python Function

12. VR (→)

- Setting Up VR Glasses
- Interact with Model



13. Advanced (\rightarrow)

- Move I/O Address
- Export as High-Resolution Images
- Multiple View Windows
- CIROS Starter
- Model Analysis
- CIROS Part Number for CP System
- Steps to Create Own Part
- Steps to Create Own Model Library
- Steps to Create Own Virtual Machine Communicating with MES4

14. Troubleshooting (\rightarrow)

• External document "CIROS-CP_Troubleshoot_EN_v7.1_xxxxx.pdf"

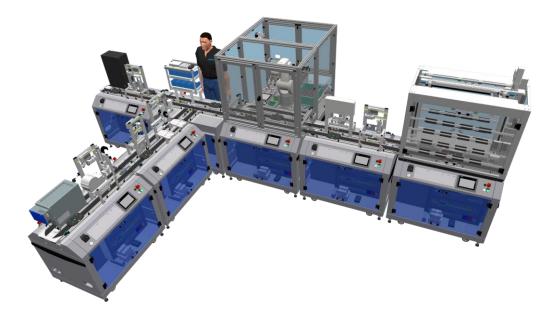


Introduction to CIROS



CIROS – Computer Integrated RObot Simulation

- Powerful, kinematic real-time 3D simulation
 - Time-discrete simulation kernel (update every 40ms)
 - Large model library: not only CP Lab / Factory, but also >1000 robots from various manufacturers
 - CAD import for user-defined modules / kinematics
 - Several interfaces, e.g. to MES4, Fleet manager, Matlab, Python, VR glasses, Mitsubishi robots, ...
 - User interaction during simulation
 - Collision analysis
 - Fault injection / simulation
 - PLC and robot programming
 - Online help with introductory examples





Studio vs. Education

- CIROS Studio
 - Full version including all features
 - Designing and saving models from scratch
 - RCI explorer interface to Mitsubishi robot controllers (download/upload of programs, individual step tracking)
 - Fits perfect for preparation of teaching scenarios to be analyzed by students

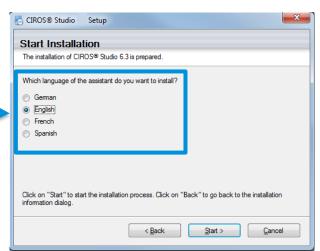
CIROS Education

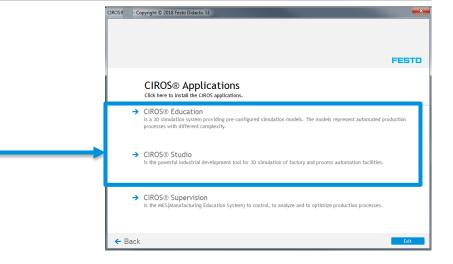
- Limited functionality
- Already existing models can be analyzed & modified, but not saved
- No RCI explorer interface to Mitsubishi robot controllers
- Fits perfect for scenarios, in which CIROS studio models have to be opened, analyzed, and modified only

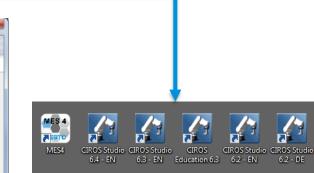


Language packs & software releases

- CIROS Studio and Education are two different software products.
- Depending on the license key one is allowed to start CIROS Studio and/or Education.
- Installation of different CIROS releases & language packs on a single PC possible.
- Unfortunately, the desired language must be set during installation (there is no option to change the language during runtime) .









Hardware requirements

- Either USB port for USB dongle or network access for server-based licensing
- Officially supported operating system: Windows 10
- Hardware requirements
 - High-performance CPU, i.e. Intel i5/i7
 - At least 8 GB main memory
 - At least 4 GB SSD memory
 - NVIDIA graphics card with OpenGL 4.5 support and 4 GB dedicated memory
- When using CIROS in combination with other software (e.g., MES4, PLCSIM Advanced) two screens are highly recommended!
- It is also possible to run CIROS and other software (e.g., MES4, PLCSIM Advanced) on different PCs

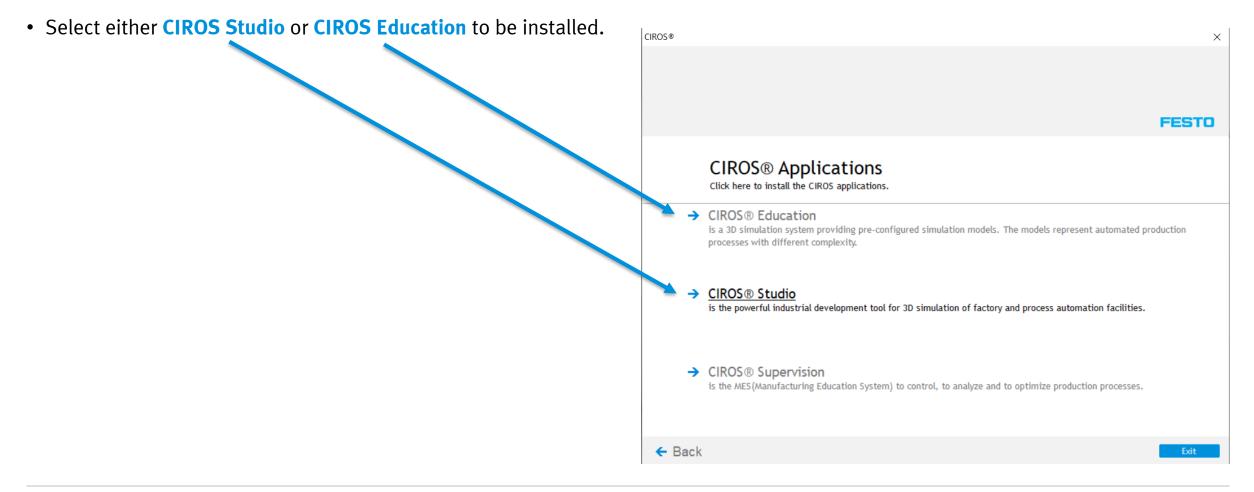


Installation

 Open Start.exe and select CIROS Applications. CIROS® \times CIROS MES4 Tutorial 6.4.1 2019_05_22
 Software
 CIROS FESTO 🖬 Öffnen reigeben für 🔻 Brennen Neuer Ordner eren 🔻 -. Name **CIROS**® riten The whole virtual learning environment of CIROS® sktop Addons → CIROS® Applications wnloads CIROS Language Click here to install the CIROS applications. Select your language here letzt besucht Documentation eDrive Tools EN - English autorun.inf → Additional Applications License Agreement Click here to install MES4 and Python. top festo.ico Make sure to read the attached license bliotheken Net_Inst_CM.pdf agreement carefully hubert, Tobias a NetzInst_CM.pdf → Online Service mputer Start.exe Links to CIROS® 32-bit version, Adobe Reader, Festo Didactic and CIROS® PDF C:) OSDisk Start.ini D.) DVD DW/ Lauf-

FESTO

Installation





License

License is managed by CodeMeter.

List of USB dongles currently attached to the system.

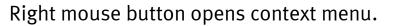


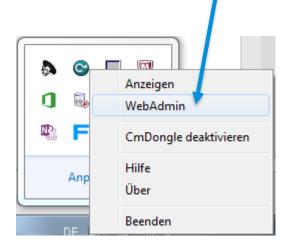
		2
CodeMeter Kontrollzentrum		
<u>D</u> atei A <u>k</u> tion <u>A</u> nsicht <u>H</u> ilfe		
Lizenz Ereignisse		
CmStick 3-4588346	Name: CmStick	
	Serien-Nr.: 3-4588346	
	Version: CmStick 4.10	ଭ
	Kapazität: 99 % free (313120 Bytes)	
	Status: 🔘 🚭 Deaktiviert	
	🔘 😔 Aktiviert solange angeschlossen	
	Aktiviert	
	Lizenzaktualisierung Auswerfen Kennwort ändern	
CodeMeter ist gestartet.		WebAdmin



View local licenses in CodeMeter WebAdmin

Option 1





Option 2

Contraction Contra	odeMeter Kontrollzer A <u>k</u> tion <u>A</u> nsicht				
0	Lizenz importieren		1		
Ð	WebAdmin Protokollierung Einstellungen	Ctrl+W Ctrl+L	Name: CmS Serien-Nr.: 3-45		
×	<u>B</u> eenden	Ctrl+Q	Version: CmS	tick 4.10	ାର୍ଯ
			Kapazität: 99 9	% free (313120 Bytes)	
			Status: 🔘	🞯 Deaktiviert	
			\odot	🮯 Aktiviert solange angeschlossen	
			۲	😋 Aktiviert	
			Lizenzaktualisierung	uswerfen Kennwort ändern	
Cod	eMeter ist gestarte	t.			WebAdmin



CodeMeter WebAdmin

Note: All CIROS related licenses should be placed in the same container.

List of licenses stored on the attached USB dongle.

SYSTEMS			CodeM	eter WebAdr	nin		C
	Container ~	License Monitoring ~	Diagnosis ~	Configuration ~ Info			
All Container	<no name=""></no>	(3-4588346)				?	English (US) ~
(le	<no name=""></no>			3-4588346		CmStick	4.10
E Q	▲ Licenses	✓ CmContainer Info	♥ User Data	✤ Backup and Restore			
∧ 10114	42 Festo [Didactic					
Product Code	Name			Unit Counter	Valid Until	License Quantity	Feature Map
191000	MES4			n/a	n/a	1	0x1
▲ 10234	44 Festo I	Didactic					
Product Code	Name			Unit Counter	Valid Until	License Quantity	Feature Map
400	CIROS 6 Educati	on		n/a	n/a	1	0x1
500	CIROS 6 Supervi	sion		n/a	n/a	200	n/a
▲ 6000	122 RIF e.\	<i>.</i>					0
Product Code	Name			Unit Counter	Valid Until	License Quantity	Feature Map
100	CIROS 6.4 Studio	MB		o n/a	n/a	1	0x1
104	CIROS 6.4 Studio	Industry Version		n/a	n/a	1	0x1



Allowing CodeMeter WebAdmin to act as a server.

SYSTEMS	CodeMeter WebAdmin	C*
Dashboard Container ~ License Monitoring	g ~ Diagnosis ~ Configuration ~ Info	
Server Configuration Server Access	2 Basic >	😢 📑 English (US) 🗸
Server Access License Access Permis	sions Server > Server Access	
Network Server	Advanced License Access Permis	sions
DisableEnable		
Network Port *: 22350		
CmWAN Server		
 Disable 		
O Enable		
(*) Changes only take effect after restarting CodeM	3 Apply Restore Defaults	



Connecting to a CodeMeter WebAdmin server. (1)

SYSTEMS	CodeM	eter Web	Ad	lmin	C∽
Dashboard	Container ~ License Monitoring ~ Diagnosis ~	Configuration ~	Inf		*
🔅 Basic Con	figuration Server Search List	Basic	>	Server Search List	? English (US) 🗸
Server	r Search List Proxy WebAdmin Backup	Server	>	Proxy	
Serve	er Search List	- Advanced		WebAdmin	
1. SI	DET2220 2			Backup	
	add new Server	/ Restore Defau	lts		



Connecting to a CodeMeter WebAdmin server. (2)

SYSTEMS	CodeMet	ter WebAdmin	C
Dashboard		Configuration ~ Info	🕜 뺄 English (US) ~
Serve 1. SI Enter t 172.21.0	Search List Proxy WebAdmin Backup r Search List DET2220 3 he Server's name or IP-Address: 199 Add Cancel Apply	Restore Defaults	

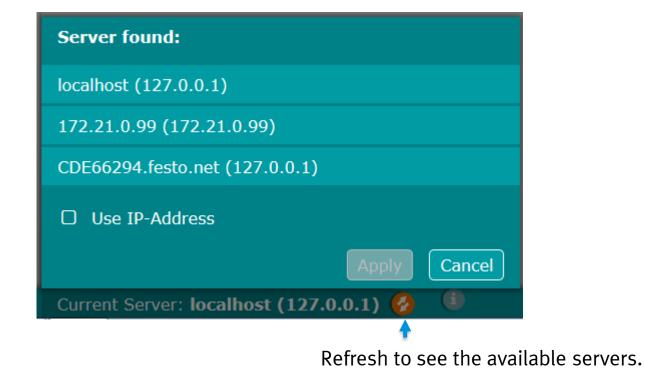


Connecting to a CodeMeter WebAdmin server. (3)

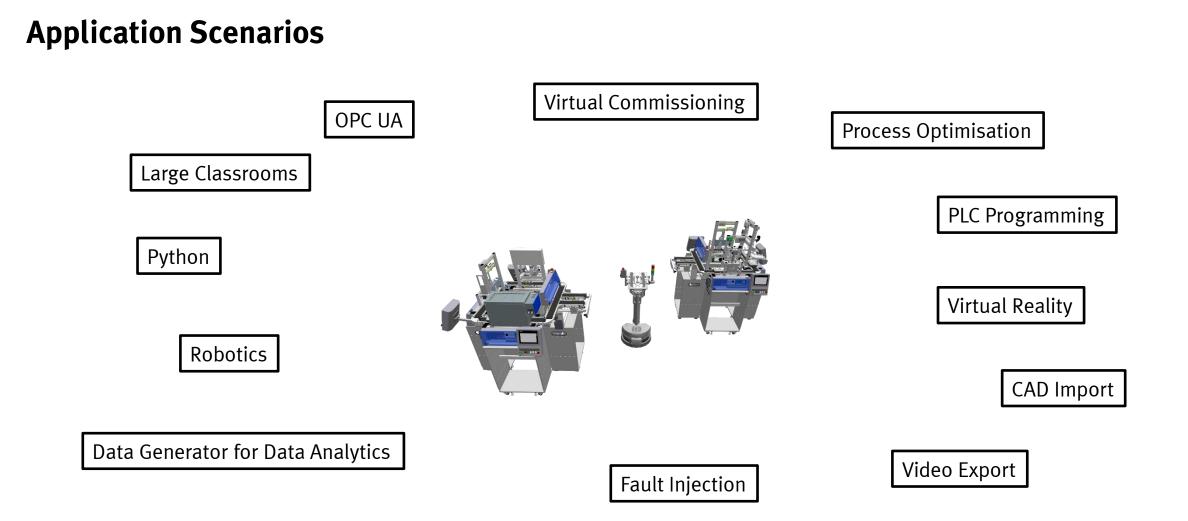
			CodeM	eter Web	Admin		С
Dashboard		License Monitoring ~ Search List	Diagnosis ~	Configuration ~	Info	0	English (US) ∽
		Proxy WebAdmin	Backup				
Serve	r Search List-						
1. SE	DET2220			(
2. 17	2.21.0.99			٢	0		
•	add new Server		_			•	
			Apply	Restore Defa	ilts		



Connecting to a CodeMeter WebAdmin server. (4)





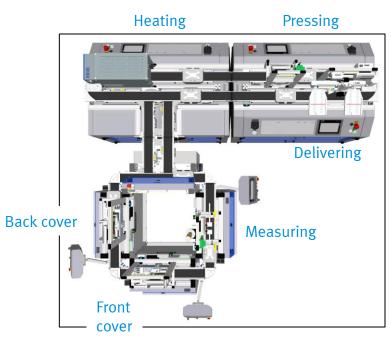


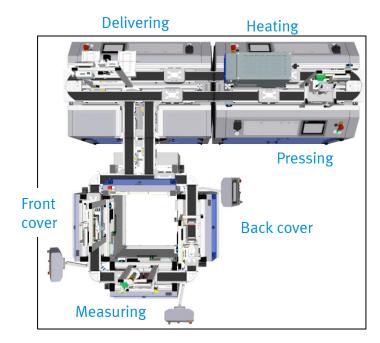


Application Scenarios

Process optimisation

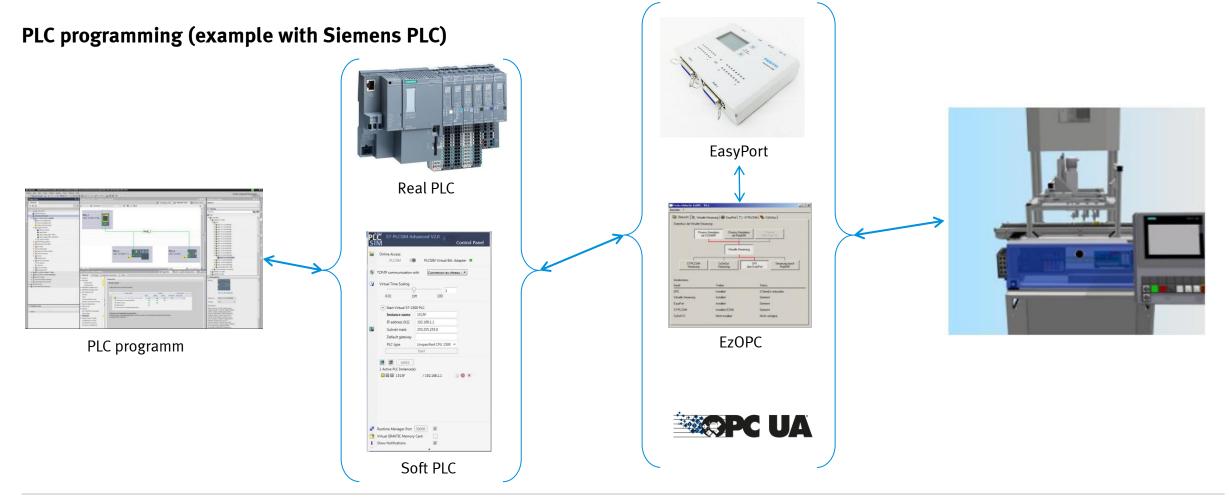
- Assume following process: Front cover \rightarrow Measuring \rightarrow Back cover \rightarrow Pressing \rightarrow Heating \rightarrow Delivering
- Which of the two configurations below is more efficient?







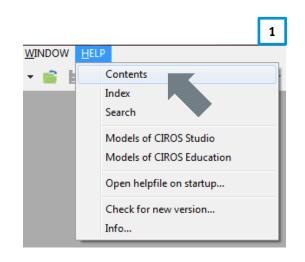
Application Scenarios

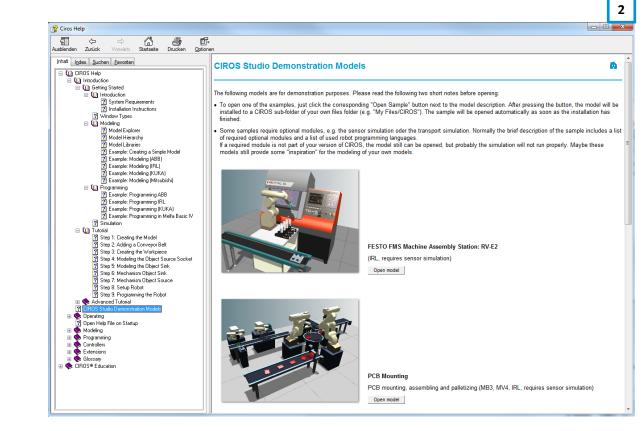




Help Menu

• Detailed help system with a couple of introductory examples, but not focusing on CP Lab / Factory explicitly.







Keyboard Shortcuts

SHIFT + left mouse button	Move user perspective in view mode
CTRL + left/right mouse button	Rotate user perspective in view mode
+ / -	Zoom in / zoom out
V / H	Front / rear view
A	Top view
R	Right view
L	Left view
SHIFT + O	Full screen
F5	Start / stop simulation
CTRL + F5	Reset simulation
CTRL + E	Toggle between view (default cursor) and edit mode (crosshair cursor)
CTRL+./CTRL+,	Rotate selected object by +/-90° in edit mode
CTRL + T	Show model explorer
CTRL + F9	Compile robot / PLC programs
CTRL + SHIFT + M	Show model libraries



Options Setting

- CIROS has two option settings, application options and model options. There are for different configurations
- Application options configure the whole CIROS application, regardless of the models.
- All changes made in model options are only applied to the active model.



Application Options

File \rightarrow Application options

- Define data import and export
- General display settings
 - Frequently used to reduce computational load.
- Editor settings
- Warning options for modelling and transport

- ORL
- Position and paths
- Programming tools settings
- VR devices configuration
- Workspaces

		Application Options		?	×
		> Data export > Data import	Workspaces		
-		✓ Display GUI OpenGL Text Rendering	Configuration Update workspace while saving the model		
FILE	EDIT VIEW MODELING PROGRAMM New > Open Ctrl+0	IN Stereo Editor ✔ Modeling	Update workspace while closing the model Save toolbar layout		
8	Close Save Ctrl+S Save as	Transport ORL Paths	Menu entries Workspaces]
er I	Save all Ctrl+Shift+S	Programming tools Save Teach-In	Coperation Connection Monitor Generation Generation Generation Generation Generation Generation		
	Export Application options Ctrl+Shift+1	VR devices Workspaces	Teacher Mode General Teacher Mode		
	Print Ctrl+P Recent files > Recent models >				
	Exit				

ОК



Model Options

Settings \rightarrow Model options

- Collision detection
- Model display settings
 - Background
 - Floor
 - Sensors
 - Etc.
- Fault simulation in teacher mode
- Interfaces
 - MES4
 - Fleet Manager
- Data logging
- Model explorer settings
- Modelling
- Online management for Mitsubishi robots ٠
- ORL •
- Overlays
- Model Simulations
- Video recorders

SETT	INGS	WINDOW	HELP	
		sion detectio	n	
2		server		
	Simu	lation film s	cript	
	Mod	el options		Ctrl+I
	Mod	el options		Ctrl+I
	Mod	el options		Ctrl+1

Model Options

Collision detection	
> Display	Simulation
Fault simulation	Model computation
> General > Interfaces	Simulation increment: 0.04 s model time
> Logging Model Explorer	Relationship model time / real time
> Modeling	• Progress preferably in real time
> Online management ORL Overlays	The progress of the model time is restricted to the real time. To achieve this, waiting times may be inserted. Additionally, you may want the model time to keep pace with the real time:
> Peripherals	Allow the skipping of model visualizations
> Simulation Video recorder	Model time may catch up on the real time but the model visualization may show jumps. Force model visualization after each 1s of elapsed model time Allow the skipping of waiting times Model time may catch up on the real time but the model visualization may show strongly accelerated sequences. Progress faster than real time (if possible) Model visualization after 10 simulation increments (after each 0.4 s model time)
	OK Cancel Apply

Apply



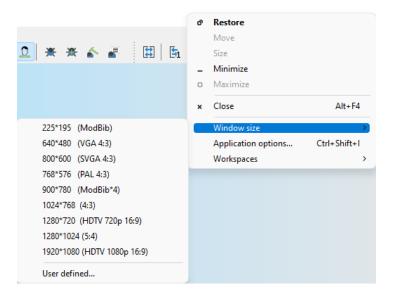
Where to look for the options?

- Structure and elements of models
 - Modell Explorer: Modelling → Model Explorer or Ctrl+T
 - Often used
- Properties of elements in the model
 - Properties: Modelling → Properties or Alt+Enter
 - Assistant
 - For example Settings → Collision detection
 - Often used
- Properties of model
 - Model options: Settings → Model options or Ctrl+I
 - Used fairly
- Properties of CIROS program
 - Application options: Files → Application options or Ctrl+Shift+I
 - Seldom used



Window's Size

• Windows size for Application Window and Modell Window can be adjusted.



Enter Wind	low Size	×
Window	size	
Width	1922	Pixel
Height	1161	Pixel
	ОК	Cancel



Introduction to CIROS Model



CIROS Model

- Preliminary remark: Each CIROS model not only consists of the modx/ini files but also the folders CF and Textures, storing the internal PLC programs and textures
- It is highly recommended, to store each CIROS model in a separate folder!
- Important: Do not copy the modx/ini files only, but the entire folder containing the subfolders CF and Textures, too!

A CIROS project folder:	Änderungsdatum	Тур	Größe
CF	09.04.2019 10:29	Dateiordner	orose
I Textures	09.04.2019 10:29	Dateiordner	
CIROS.ini CIROS.modx	09.04.2019 10:33 09.04.2019 10:33	Konfigurationseinstellungen CIROS Model	61 KB 88.236 KB



Collaborative Working

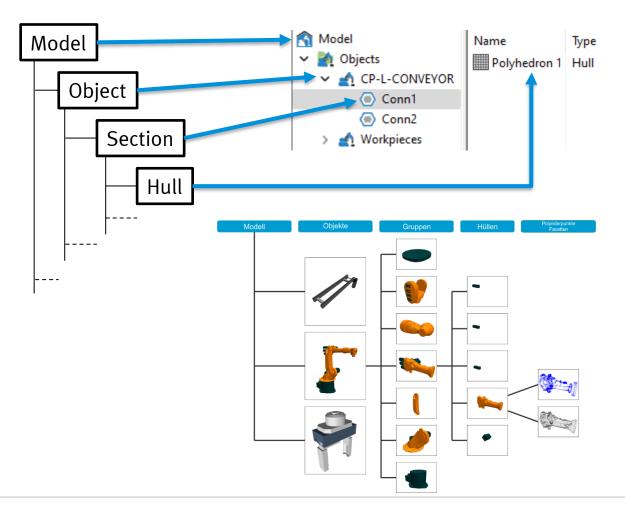
- Several users can work on the same model at the same time.
- When a user changes the model, other users will receive a notification.
- However, simultaneous changes and changes that crossed over time cannot be merged together.



Model's Structure

Elements in a model

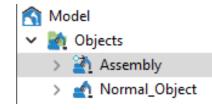
- Structures of elements
 - Model / Environment
 - Objects: Logical unit
 - Sections: Static body
 - Hulls: Geometries
- Positions based on coordinate system
- Hulls
 - Geometric primitives
 - Box, sphere, etc.
 - Polyhedron
 - Vertex, Facet

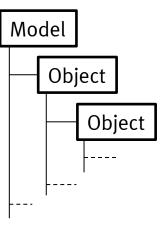




Model's Structure

- Objects in the structure tree
 - Placed on the highest hierarchy in the model or
 - Placed under an object
- Object's nomenclature
 - Parent object: superordinate object
 - Child object: child of a parent object
- Usage
 - For a clear structure
 - Definition of static assembly
- Moving child objects
 - During modelling: always
 - During simulation: only when the object is an object assembly
 - Select object \rightarrow right click \rightarrow Edit \rightarrow Assembly

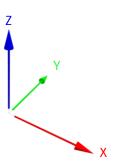






Elements and Coordinate Systems

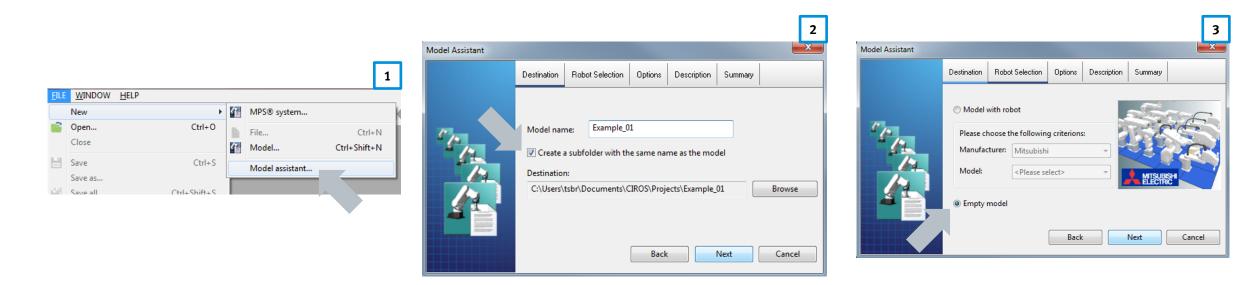
- Different coordinate systems, each might has same or different origins
 - World : based on world coordinate system
 - Object : based on coordinate system of the parent object
 - Section : based on coordinate system of the section it belongs to
 - Hull : based on coordinate system of the hull
- In model window, origin of three axes of coordinate system are shown in different colours.





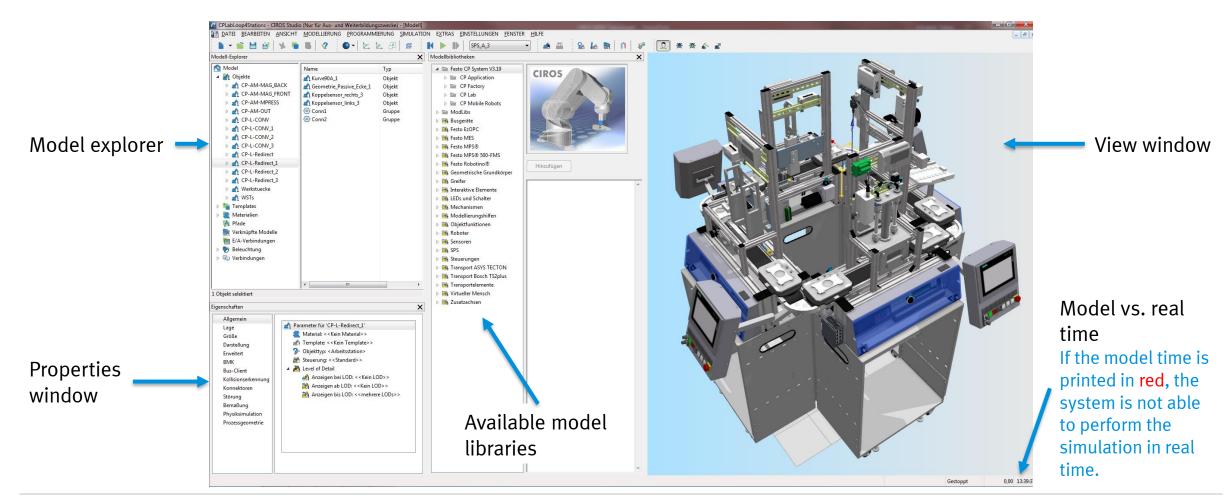
Create a CIROS Model

- Recommended way of defining a new CIROS Model
 - 1. Choose FILE \rightarrow New \rightarrow Model assistant
 - 2. Specify the model's name and enable Create a subfolder with the same name as the model.
 - 3. Important: Do not select a robot, these ones are not the ones integrated within CP Lab / Factory! Choose Empty model instead!





Window's Layout





Toolbar can be configured

Standard

View

✓ Controller

Programming

Modeling extensions
 Collision detection

✓ I/O monitors

PLC Switch

Fault simulation

3D marker

Mitsubishi online connection

OPC server

OPC UA server

Python

RAPID programming tools

VR

Simulation film

Video recorder

Hide all

Save as default

Use default

Reset default



View and Edit Mode

View mode

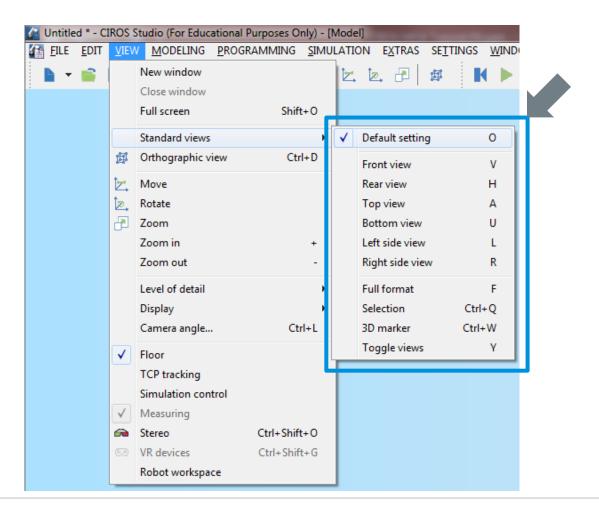
- Change user perspective onto scene
- Default cursor:

Edit mode

- Place, move, rotate objects within scene
- Crosshair cursor:



Standard Views





Standard Views



Default setting



Front view



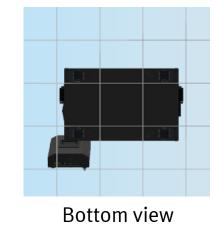
Left side view



Right side view



Rear view





Top view

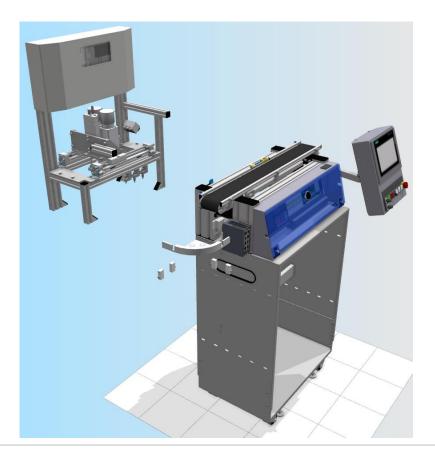


View Used when Working with CIROS Model

- With respect to the z-axis all CP Lab / Factory modules within the library have been prepared in such a way that
 - Conveyor belts, CNC milling stations, robot assembly stations, warehouses, and Robotinos are placed on the floor (z = 0mm)
 - Carriers, deflections, sources, sinks, and application modules flush with the conveyor belts' upper edges (z = 975mm)

• Important

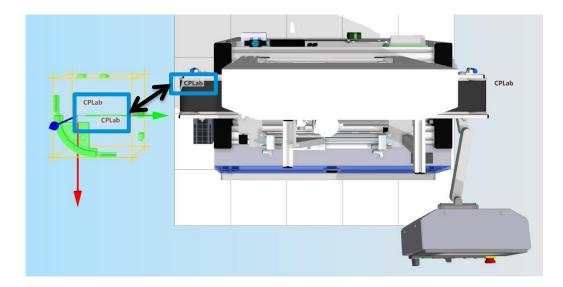
- Switching to Top View ensures that the z-values remain constant when moving components within the scenery!
- Snapping into place of modules works well in Top view only!

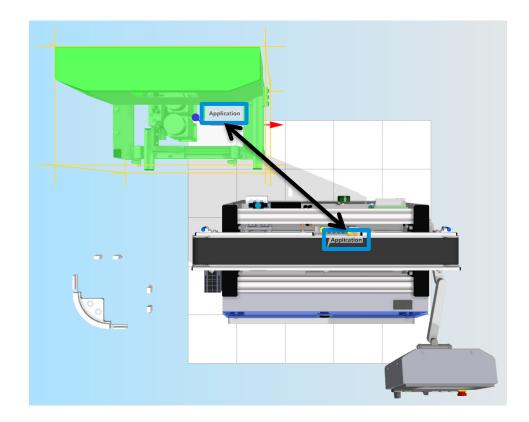




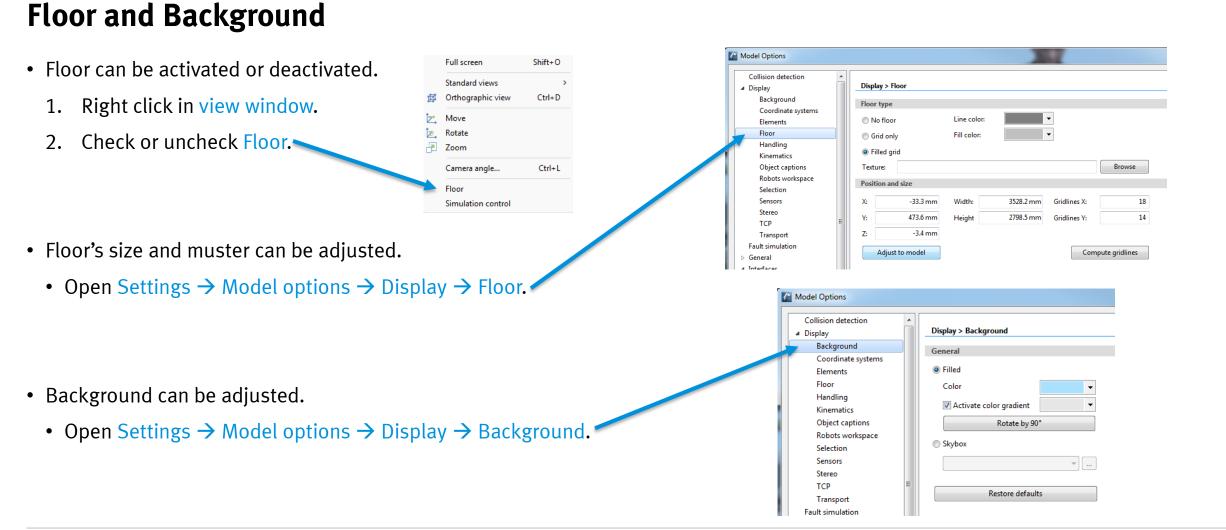
Snapping into Place

- When placing application modules onto conveyor belts, it is not necessary to adjust them as precisely as possible.
- Just putting the phrases Application, CPLab, Modul, etc. on top of each other is all one has to do!
- Same holds for mounting deflections, docking kits, and so on.





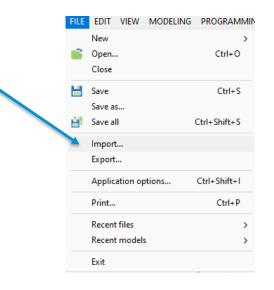






Import Data

- It is possible to import a model from CAD data by selecting File \rightarrow Import.
- Supported formats are as follow:
 - 3ds Max
 - AutoCAD DXF
 - Autodesk
 - Blender
 - Collada
 - IGES
 - PointCloud
 - STEP
 - STL
 - VRML
 - Wavefront Object





Export Data

- It is possible to export a model or selected objects in the model to CAD or picture.
- 1. To export a whole model, in Model Explorer, select Objects. To export objects in model, select the objects in Model Explorer.
- 2. Click File \rightarrow Export.
- Supported export formats are:
 - AutoCAD
 - For viewing in a web browser (HTML)
 - PNG
 - IGES
 - POV-Ray scene description
 - RT toolbox file
 - STEP
 - STL
 - VRML
 - Windows bitmap



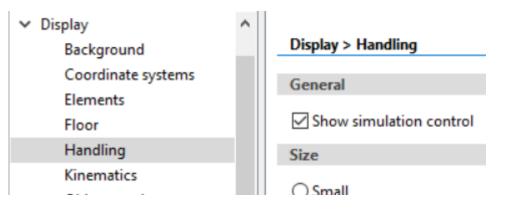
Teacher Mode

- Teacher mode can be activated to configure fault simulation in "Fault Setting " and "Fault Log" windows.
 - Extras \rightarrow Fault Simulation \rightarrow Teacher Mode
 - 0
- Password for Teacher Mode is "didactic".



Simulation Control in Model Window

- Simulation control buttons can be shown in model window.
- This is to allow simulation control in full screen mode or in VR.
- 1. Go to "Settings \rightarrow Model option".
- 2. Select "Display \rightarrow Handling".
- 3. In section "General", activate "Show simulation control".
- 4. Click on "ok".

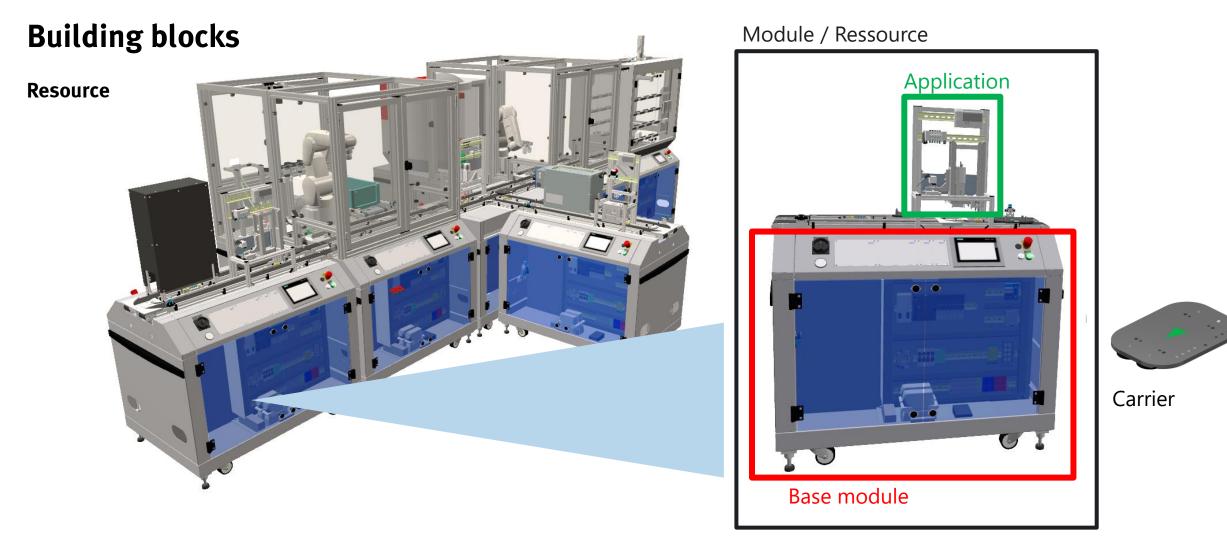






Introduction to Festo CP-System

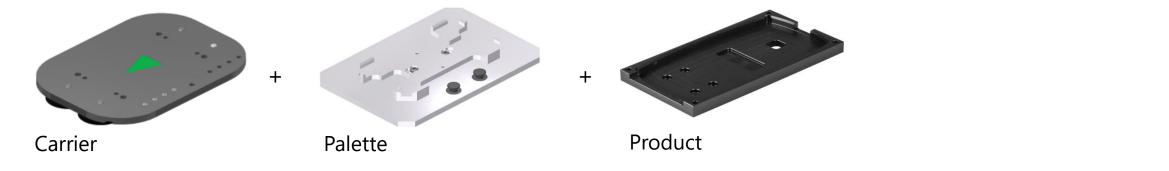






Building Blocks

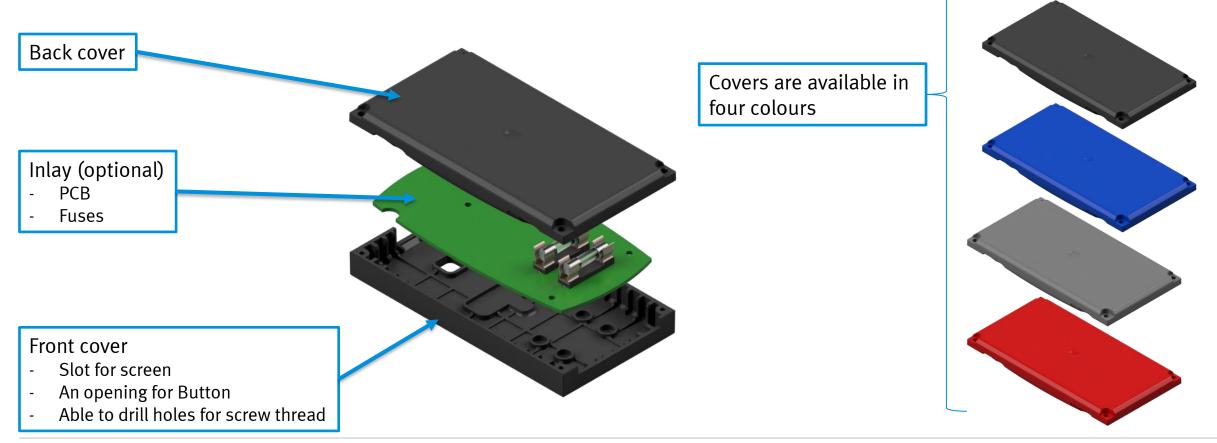
Carrier and product







Product (Smartphone)



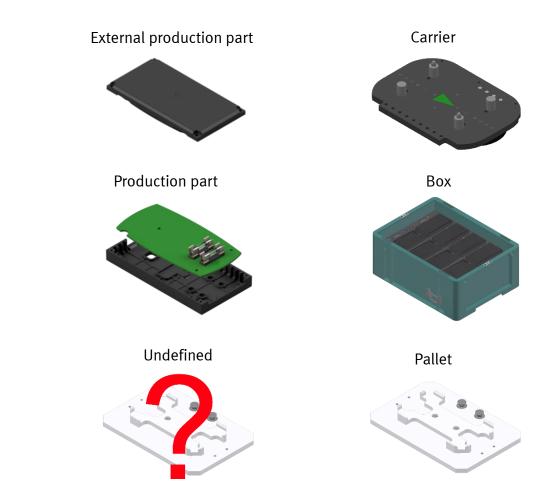


Terms & Definitions

Parts

- Set of all objects, except Robotinos, that are moving around within a CP Lab / Factory.
- Parts get identified by a unique part number (PNo).
- Six different subclasses.

External production parts	Parts not produced by CP Lab / Factory
Production parts	Parts produced by CP Lab / Factory
Boxes	Transport workpieces from one production facility to another
Pallets	Mounted on top of the carriers
Carriers	Move (external) production parts on the conveyor belts.
Undefined	Containing all unknown parts or objects, typically used to represent faulty parts.





Standard Part Numbers

Supported part numbers

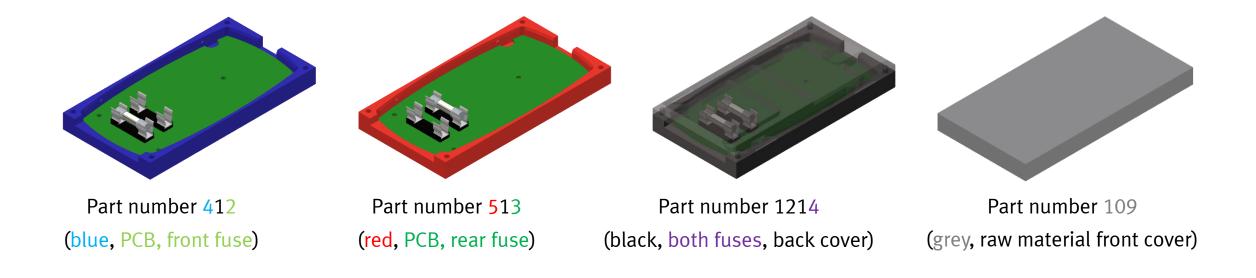
25 31	Pallet Carrier
110 / 109 / 108 / 107	Raw material front cover black / grey / blue / red
111 / 112 / 113 / 114	Back cover black / grey / blue / red
210 / 310 / 410 / 510	Front cover black / grey / blue / red
x11 / x12 / x13 / x14 (x = 2, 3, 4, 5)	Front cover in black / grey / blue / red including PCB / front fuse / rear fuse / both fuses
1y11 / 1y12 / 1y13 / 1y14 (y = 2, 3, 4, 5)	Front and back cover in black / grey / blue / red including PCB / front fuse / rear fuse / both fuses
26 / 120 / 130	Unknown workpiece / PCB / fuse

- Supported part numbers are parts that can be booked into a high-bay warehouse at simulation time t=0s.
- All other part numbers in particular user-defined parts can be generated and stored into a high-bay warehouse during production, but not replicated in a warehouse at simulation startup at t=0s.



Standard Part Numbers

Some examples





Groups and Utilities

- Groups define sets of parts somehow belonging to each other.
- Besides the list of parts belonging to a group each group also contains a unique ID and a description.
- There is no restriction on the number of different types of parts a group consists of.
- Parts can be attached to an arbitrary number of groups or not attached to any group at all.
- Typically, groups are used to define zones and restrictions wrt. buffer positions of a high-bay warehouse.
- Utilities defines a special subclass or group of parts, containing all carriers, boxes, and pallets, which are the ones that are responsible for transport purposes.





Carriers

- In CIROS two different kinds of carriers are supported
- CP-F-CARRIER: Carriers without Pallets



• CP-L-CARRIER: Carriers including Pallets



- Carriers without Pallets must be used in cases in which a CP Factory high-bay warehouse is part of the CIROS model, since the workpieces will be stored and released together with pallets in this situation!
- In all other cases Carriers including Pallets should be used
- Rule of thumb according to the number of carriers used within a CIROS model: One carrier per conveyor belt.



Resources and Buffers

- Resources are the production facilities (modules) of the CP Lab / Factory.
- Each resource is represented by a unique ID.
- Periodically, each resource sends a status update to MES4 and gets process data back when initiating a corresponding service call.
- Robotino is a special kind of a mobile resource.
- Some resources contain buffers, like the high-bay warehouse, the robot assembly station, Robotinos, and the branches.
- Each buffer consists of at least one buffer position to store parts, one per buffer position.
- MES4 could be configured in such a way that, based on zones and restrictions, buffer positions are allowed to store specific types of parts only.



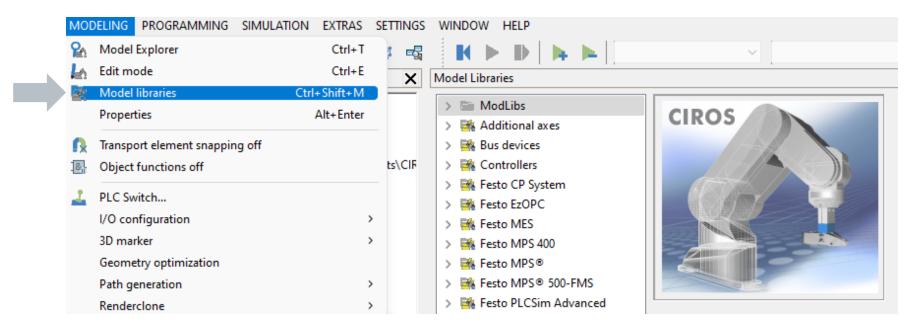


CP-System Model Libraries



Open Model Libraries Window

Modelling \rightarrow Model libraries

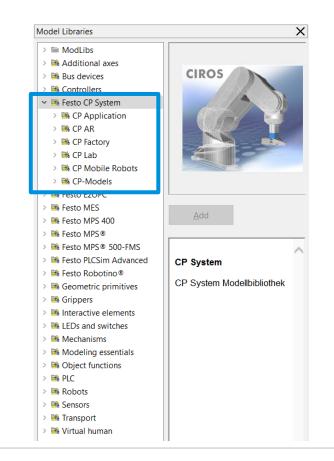




• There are four groups containing CP Lab / Factory modules

Name	Description
CP-Application	Application modules
CP AR	AR Marker for Festo Didactic Augmented Reality
CP Factory	CP-Factory based modules and stations
CP Lab	CP-Lab based modules
CP Mobile Robots	Robotino related modules
CP-Models	CP-Lab standard systems with configured MES4 v1 database

• For each module there is a brief description and a tiny image.





CP Application

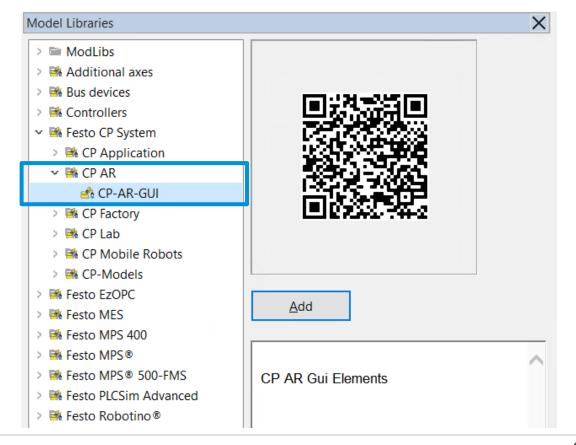
Module	Description
CP-AM-CAM	Camera inspection
CP-AM-DRILL	Drilling
CP-AM-HEAT	Heating tunnel
CP-AM-IDRILL	Drilling module with own PLC
CP-AM-iPICK	Pick by light with own PLC
CP-AM-LABEL	Labelling printer
CP-AM-MAG_BACK	Back cover magazine
CP-AM-MAG_FRONT	Front cover magazine
CP-AM-MANUAL	Manual working place
CP-AM-MEASURE	Analog measurement
CP-AM-MPRESS	Muscle press
CP-AM-OUT	Workpiece output
CP-AM-PRESS	Pressing
CP-AM-TURN	Workpiece flipping

🗸 🏽 Festo CP System	
🗸 🏽 CP Application	
📣 CP-AM-CAM	
📣 CP-AM-DRILL	
📣 CP-AM-HEAT	
📣 CP-AM-iDRILL	
📣 CP-AM-iPICK	
📣 CP-AM-LABEL	
📣 CP-AM-MAG_BACK	A.44
📣 CP-AM-MAG_FRONT	Add
📣 CP-AM-MANUAL	
📣 CP-AM-MEASURE	
📣 CP-AM-MPRESS	CP System
📣 CP-AM-OUT	
📣 CP-AM-PRESS	CP System Modellbibliothek
📣 CP-AM-TURN	
> 🏽 CP AR	
> 🏽 CP Factory	
> 🏽 CP Lab	
> 🏽 CP Mobile Robots	
> i CP-Models	



CP AR

Module	Description
CP-AR-GUI	CP-AR QR-Code to the AR server.





CP Factory

Module	Description
CP-F-ASRS20-B	Automated Storage Retrieval System for boxes
CP-F-ASRS32-P	Automated Storage Retrieval System for parts and pallets
CP-F-BRANCH	Branch base module
CP-F-BUF-B	Manual working place with automated box transfer
CP-F-BUFROB-B	Part transfer between box and CP-Factory line with robot
CP-F-BUFROBM-B	CNC milling application attached to robot and box transfer
CP-F-BYPASS	CP-Factory base module with bypass belt
CP-F-CARRIER	15 carriers for CP-Factory base modules without pallet
CP-F-DEFLECTION180	Passive 180° deflection
CP-F-FEEDROBM	CNC milling application attached to robot and CP-F linear
CP-F-LINEAR	CP-Factory base module with two parallel conveyor belts
CP-F-RASS	Robot assembly station
CP-F-SINK	Sink to remove carriers, pallets, workpieces
CP-F-SOURCE	Source to generate carriers, pallets, workpieces

🐝 Festo CP System
> 📑 CP Application
> 🏽 CP AR
✓ ➡ CP Factory
💰 CP-F-ASRS20-B
💰 CP-F-ASRS32-P
🛃 CP-F-BRANCH
💰 CP-F-BUF-B
📣 CP-F-BUFROB-B
📣 CP-F-BUFROBM-B
🛃 CP-F-BYPASS
📣 CP-F-CARRIER
🛃 CP-F-DEFLECTION180
💰 CP-F-FEEDROBM (Mitsubishi)
📣 CP-F-LINEAR
💰 CP-F-RASS (Mitsubishi)
💰 CP-F-SINK
🛃 CP-F-SOURCE
> 🃑 CP Lab
> Mobile Robots
> 🏽 CP-Models

v



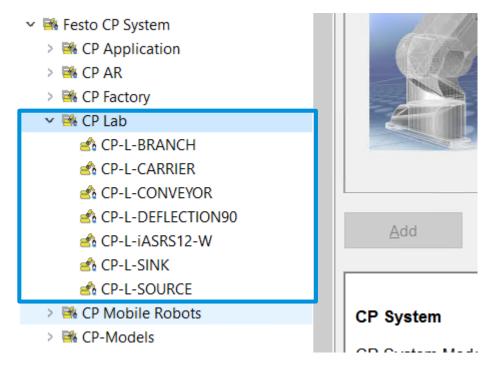
CP System

CP System Modellbibliothek



CP Lab

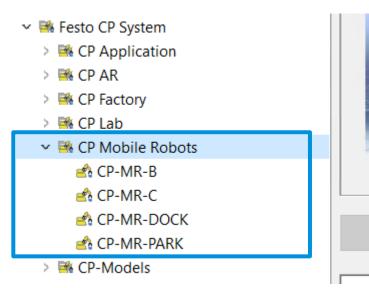
Module	Description
CP-L-BRANCH	Branch base module
CP-L-CARRIER	15 carriers for CP-Lab base modules with pallet
CP-L-CONVEYOR	CP-Lab base module with a conveyor belt
CP-L-DEFLECTION90	Passive 90° deflection
CP-L-iASRS12-W	Automated Storage Retrieval System for parts
CP-L-SINK	Sink to remove carriers, pallets, workpieces
CP-L-SOURCE	Source to generate carriers, pallets, workpieces





CP Mobile Robots

Module	Description
CP-MR-B	Robotino for boxes
CP-MR-C	Robotino for carriers
CP-MR-DOCK	Docking kit to be mounted on branches to enable (un)docking maneuvers by a Robotino
CP-MR-PARK	Robotino parking position
~~~~	

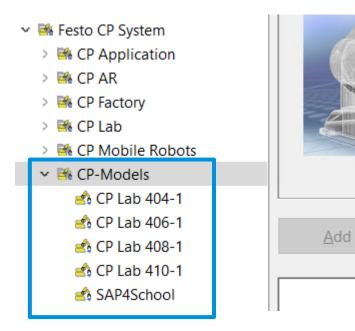






#### **CP-Models**

Module	Description
CP Lab 404-1	<ul> <li>Standard CP Lab system with four stations and configured MES4 v1</li> <li>1. Front cover magazine</li> <li>2. Analog measurement</li> <li>3. iDrill</li> <li>4. Workpiece Output</li> </ul>
CP Lab 406-1	<ul> <li>Standard CP Lab system with six stations and configured MES4 v1</li> <li>All modules in CP Lab 404-1</li> <li>Back cover magazine</li> <li>Pressing</li> </ul>
CP Lab-408-1	<ul> <li>Standard CP Lab system with six stations and configured MES4 v1</li> <li>All modules in CP Lab 406-1</li> <li>Pick by light</li> <li>Labelling</li> </ul>
CP Lab 410-1	<ul> <li>Standard CP Lab system with six stations and configured MES4 v1</li> <li>All modules in CP Lab 408-1</li> <li>Camera inspection</li> <li>Turning</li> </ul>
SAP4School	Model for virtual commissioning of SAP4School



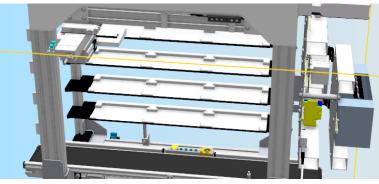


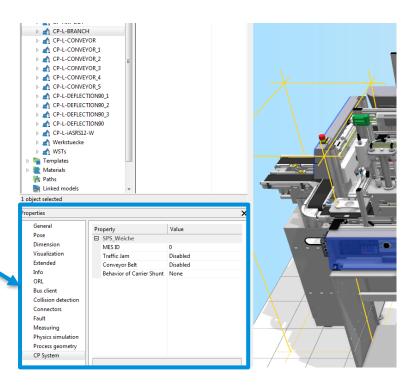


### **Configuration in Properties Section CP System**

- For almost all components there is a section CP System as being part of the corresponding Properties menu.
- Represents the options that can be defined at the HMIs of a real CP Lab / Factory
  - MES ID
  - Traffic jam control
  - Energy saving (stopping the belts whenever possible)
  - Behavior of branches
- Additionally, one can configure some CIROS internal parameters not available on a real CP Lab / Factory.

eneral	Pre	operty	Value
	Ξ	SPS_A_1	
nsion		MES ID	0
zation		Traffic Jam	Disabled
ded		Conveyor Belt	Disabled
		Init_LabLager	
		Part Number	110
lient	Ξ	Replicator_PartNr	
ision detection		Percentage Faulty Fuses Rear	0.00
nnectors		Percentage Faulty Fuses Front	0.00
t		Percentage Rotated Workpieces	0.00
uring			
simulation			
ocess geometry			
/stem			





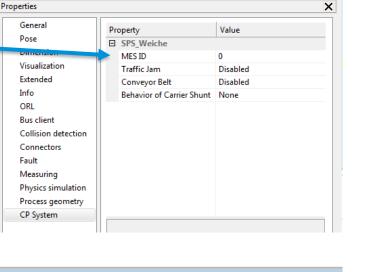


### **Configuration in Properties Section CP System**

#### **Define MES ID (1)**

- MES ID = 0  $\rightarrow$  Default mode without using MES4
- MES ID > 0  $\rightarrow$  MES4 mode
- Note, that the default mode of CIROS is not equal to the default mode of a real CP Lab / Factory!
- If the MES ID of at least one component is greater 0, running the CIROS simulation without MES4 results in an error message!

Messages 10061 -> Connection refused MES connection: Connect error #10061 10061 -> Connection refused MES connection: Connect error #10061 10061 -> Connection refused MES connection: Connect error #10061 10061 -> Connection refused



Info

ORL



# **Configuration in Properties Section CP System**

### Define MES ID (2)

- Typically, MES ID is defined with one of the two ideas below:
  - Based on the MES IDs of a similar real CP Lab / Factory.
  - According to the default process to be performed, starting with MES ID = 1 for the component which is executing the first step of the process.
- Constraints
  - MES IDs must be unique throughout the entire model .
  - Each MES ID must be greater 0.
  - Definition of IDs within CIROS and MES4 must match each other.

ieneral			
	Pr	operty	Value
Pose	Ξ	SPS_Weiche	
Dimension		MES ID	0
/isualization		Traffic Jam	Disabled
Extended		Conveyor Belt	Disabled
nfo		Behavior of Carrier Shunt	None
ORL	_		
Bus client			
Collision detection			
Connectors			
Fault			
Measuring			
Physics simulation			
Process geometry			
CP System			

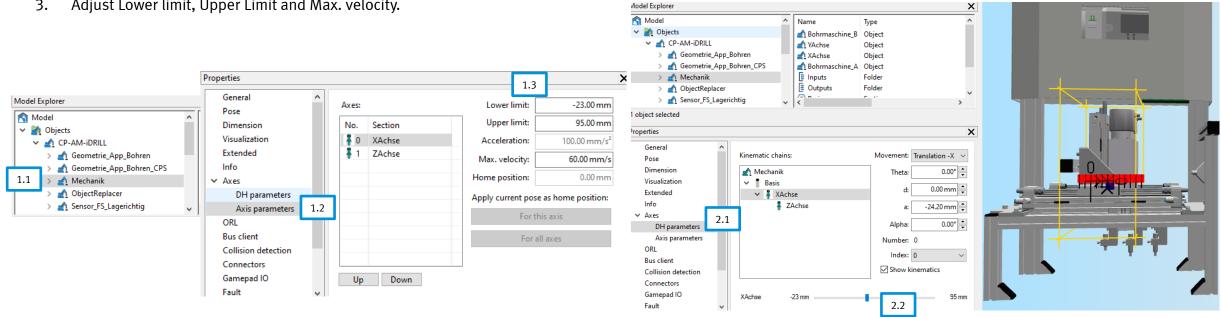


### **CP-AM-iDRILL**

#### **Configure x-axis linear drive**

- 1. Configure axis lower limit, upper limit and maximum velocity.
  - In Model Explorer, select CP-AM-iDRILL  $\rightarrow$  Mechanik. 1.
  - In Properties, select Axes  $\rightarrow$  Axis parameters. 2.
  - 3. Adjust Lower limit, Upper Limit and Max. velocity.

- 2. It is possible to see the position of the drill bit in model window.
  - In Properties, select Axes  $\rightarrow$  DH parameters. 1.
  - By dragging the scale XAchse at bottom, the drill bit will move accordingly 2.



# **CP-AM-iDRILL**

#### **Object Mechanik (1)**

- The object Mechanik has type Mechanism. It is a double acting cylinder.
- Object type double acting cylinder in CIROS has following attributes, which can be assigned to I/O. By default, following attributes are connected.
- The attributes can be viewed in CP-AM-iDRILL → Mechanik → Section XAchse → Properties → Mechanism. 
   Model



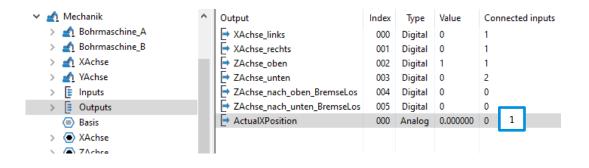
Туре	Cylinder, double-acting				
Attri	bute	Value			
Retra	act [l/O]	XAchse_nach_links			
Exter	nd [I/O]	XAchse_nach_rechts			
Spee	d [input]				
Retra	acted [output]	XAchse_links			
Exter	nded [output]	XAchse_rechts			
Actu	al position [output]				
Mov	ement characteristics (.csv)				
Min.	limit defined by grip relation [gpp]				
Max.	limit defined by grip relation [gpp]				
Press	sure switch retracting [output]				
Press	sure switch extending [output]				



# **CP-AM-iDRILL**

#### **Object Mechanik (2)**

- It is possible to assign own I/O to the attribute.
- In this example, An analog output ActualXPosition is assign to the attribute Actual position.
- 1. Add a new analog output with name ActualXPosition in object Mechanik.
- 2. In Model Explorer, select CP-AM-iDRILL → Mechanik → Section XAchse.
- 3. In Properties, select Mechanism.
- 4. In Attribute table, select the row Actual position [output].
- 5. Click on the drop down list at the bottom, select ActualXPosition.
- 6. Click Apply.



	General Pose	Type Cylinder, double-acting	~	
	Dimension	Attribute Va	lue	
	Visualization	Retract [I/O] XA	XAchse_nach_links	
	Extended	Extend [I/O] XA	XAchse_nach_rec	
	Info	Speed [input]		
3	Mechanism	Retracted [output] XA	Achse_links	
	Measuring	Extended [output] XA	Achse_rechts	
	Process geometry	Actual position [output] Act	ActualXPosition	
		Movement characteristics (.csv)		
		Min. limit defined by grip relation		
		Max. limit defined by grip relation		
		Pressure switch retracting [output]		
		Pressure switch extending [output]		
		<nothing> 5 V W <nothing> ActualXPosition</nothing></nothing>	fizard Apply ?	

# **CP-AM-iPICK**

Property	Description
Working time (s)	Working time required by the worker to finish an operation (valid for CIROS and CIROS/MES4).
Target number	Part number of the part to be replicated by the worker (valid for CIROS default mode only).

Linked models Linked models I/O connections Lighting Graph Connections 1 object selected				
Properties			×	
General Pose Dimension Visualization Extended Info ORL Bus client Collision detection Connectors Gamepad IO Fault Measuring Physics simulation Process geometry CP System	Property CP-AM-iPICK Working time (s) Target number	Value 5.00 210		

# **CP-AM-MAG_BACK**

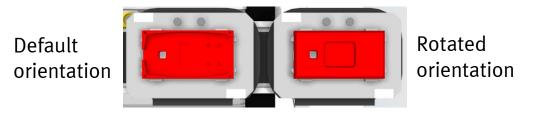
Properties	Description
Automatic Mode	In "Automatic Mode" the magazine will be filled automatically during simulation.
Colour Workpiece	Defines the color of the back covers stored in the magazine. By enabling "Manual Override" the user can specify the color of each individual cover to be replicated.

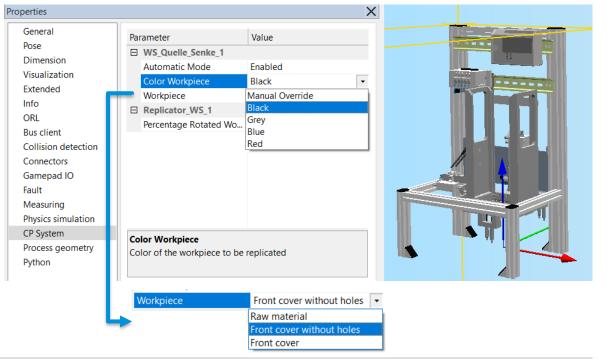
operties			X	
General	Property	Value		
Pose	WS_Quelle_Senke			
Dimension	Automatic Mode	Enabled		
Visualization	Color Workpiece	Black	•	
Extended		Manual Override		
Info		Black		
ORL		Grey		
Bus client		Blue Red		
Collision detection				
Connectors				
Gamepad IO				-34
Fault				
Measuring				
Physics simulation				
Process geometry				
CP System	Color Workpiece			
	Color of the workpiece to	be replicated		



## **CP-AM-MAG_FRONT**

Properties	Description
Automatic Mode	In "Automatic Mode" the magazine will be filled automatically during simulation.
Colour Workpiece	Defines the color of the front covers stored in the magazine. By enabling "Manual Override" the user can specify the color of each individual cover to be replicated.
Workpiece	Defines the front cover type, is it a raw plastic block, front cover without drilled holes or final front cover.
Percentage Rotated Workpieces	Fault injection option, defines percentage of faulty part filled to the magazine.



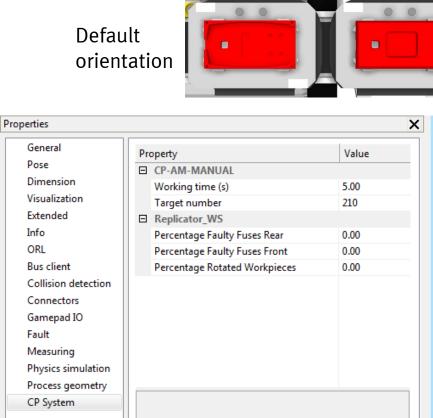




# **CP-AM-MANUAL**

### **Configuration in properties section CP System**

Properties	Description
Working time (s)	Working time required by the worker to finish an operation (valid for CIROS and CIROS/MES4).
Target number	Part number of the part to be replicated by the worker (valid for CIROS default mode only).
Percentage Faulty Fuses	Fault injection option. In case the worker must replicate workpieces with fuses, one can specify the percentage of "faulty" fuses assembled (faulty fuses are highlighted in red).
Percentage Rotated Workpieces	Fault injection option. Percentage of workpieces placed in a rotated orientation (upside down).





Rotated

orientation



Faulty fuses are shown in red

# **CP-AM-MEASURE**

urement Variation Fault injection option, adds some random
"noise" with Gaussian filter to the measured value.       General       Property       Value         Bose       Messen       Messen       Messen
Dimension   Visualization   Extended   Info   ORL   Bus client   Collision detection   Collision detection   Connectors   Gamepad IO
Fault       Measuring       Physics simulation       Process geometry       CP System

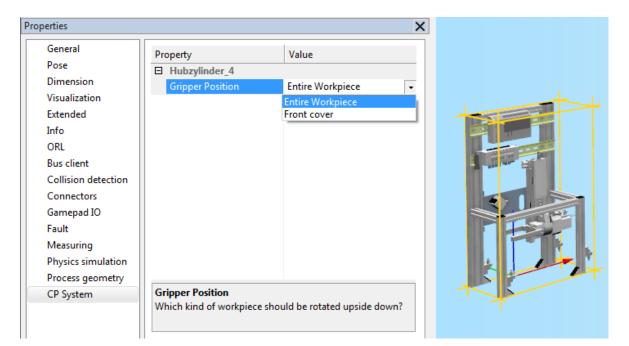
# **CP-AM-OUT**

Properties	Description
Automatic Mode	In "Automatic Mode" workpieces will be removed from the ramps automatically during simulation. Otherwise, one might run into a traffic jam due to slides full!

roperties			×	
General	Property	Value		
Pose	WST_Senke			
Dimension	Automatic Mode	Enabled		
Visualization				
Extended				
Info				
ORL				
Bus client				
Collision detection				
Connectors				
Gamepad IO				
Fault				
Measuring				
Physics simulation				
Process geometry				
CP System				

### **CP-AM-TURN**

Properties	Description
Gripper Position	Before starting the simulation, one must specify whether front covers or entire workpieces (i.e., front & back covers pressed together) should be rotated upside down.





# **CP-F-BRANCH / CP-L-BRANCH**

Properties	Description
Traffic Jam	Stop carrier if conveyor belt is occupied.
Conveyor Belt	Stop conveyor belt while application is running.
Behaviour of Carrier Shunt	Percentage of carriers without an order turning to the right.

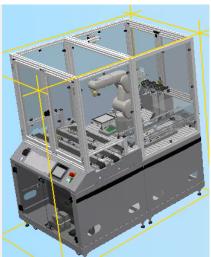
General	Parameter	Value	
Pose	SPS_Weiche		
Dimension	MES ID		
Visualization	Traffic Jam	Disabled	
Extended	Conveyor Belt	Disabled	
Info ORL	Behavior of Carrier Shunt	50%	
Bus client			
Collision detection			
Connectors			
Fault			
Measuring			
Physics simulation			
CP System			
Process geometry			
Python			



# **CP-F-RASS**

Properties	Description
Traffic Jam	Stop carrier if conveyor belt is occupied.
Conveyor Belt	Stop conveyor belt while application is running.
Behaviour of Carrier Shunt	Percentage of carriers without an order turning to the right.
Automatic Mode	In automatic mode PCBs in the box will be created automatically regardless of buffer in MES4.
ID for the box	If the box is not created by MES, this will be the ID of the box.
Percentage Faulty Fuses	Percentage of "faulty" fuses assembled by the robot assembly station. Faulty fuses are highlighted in red.

General	Parameter	Value
Pose Dimension Visualization Extended Info ORL Bus client Collision detection	SPS_Roboter  MES ID  Traffic Jam  Conveyor Belt  Automatic Mode  Buffer_RASS_1_1  ID for the box (if not cre  Werkzeug-Sicherung	0 Disabled Disabled Enabled
Connectors Fault Measuring Physics simulation CP System	Percentage Faulty Fuses	10





# **CP-MR-C / CP-MR-B**

### **Configuration in properties section CP System**

Properties	Description
RFM ID	Unique ID, mandatory for the Festo Fleet Manager to get access to a particular Robotino .

#### Note:

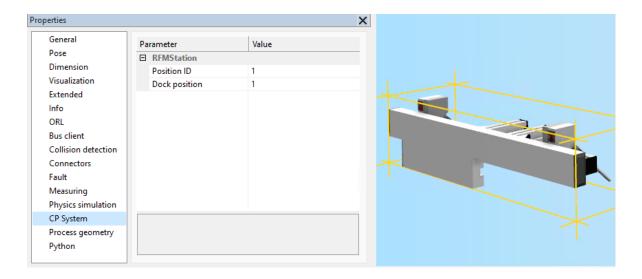
- CP-MR-C stands for Cyberphysical-Mobile Robot-Carrier.
- CP-MR-B stands for Cyberphysical-Mobile Robot-Box.
- **RFM** stands for Robot Fleet Manager.

perties			×
General	Property	Value	*
Pose	RFMController		
Dimension	RFM ID	1	
Visualization			
Extended			
Info			
ORL			
Bus client			
Collision detection			-
Connectors			
Gamepad IO			
Fault			
Measuring			
Physics simulation			
Process geometry			
CP System			



### **CP-MR-DOCK**

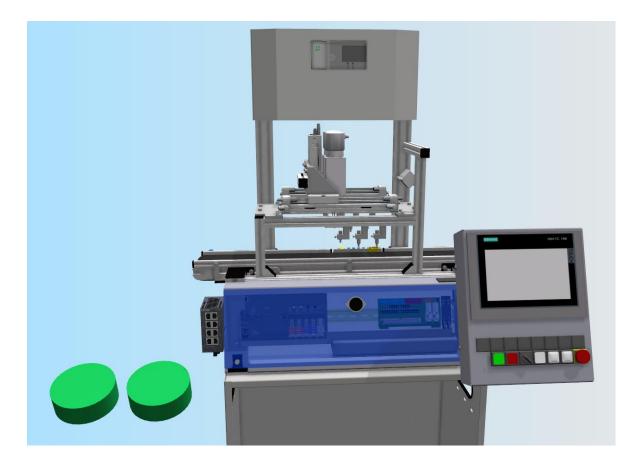
Properties	Description
Station ID	Unique ID, mandatory for the Festo Fleet Manager to differ between the various docking positions a Robotino could dock to within a model.
Dock position	Number of conveyor belts available for exchanging workpieces (always 1 for carrier Robotinos).





### **Sources and Sinks**

- Sources and Sinks can be used to dynamically replicate and remove workpieces during the simulation, either via pushing the corresponding button or automatically controlled by CIROS.
- There are different versions of sources and sinks, one set to be applied to CP Lab conveyor belts and one set to be used in combination with CP Factory components.
- Typically, sources and sinks come into play whenever one wants to model a single conveyor belt & application module.

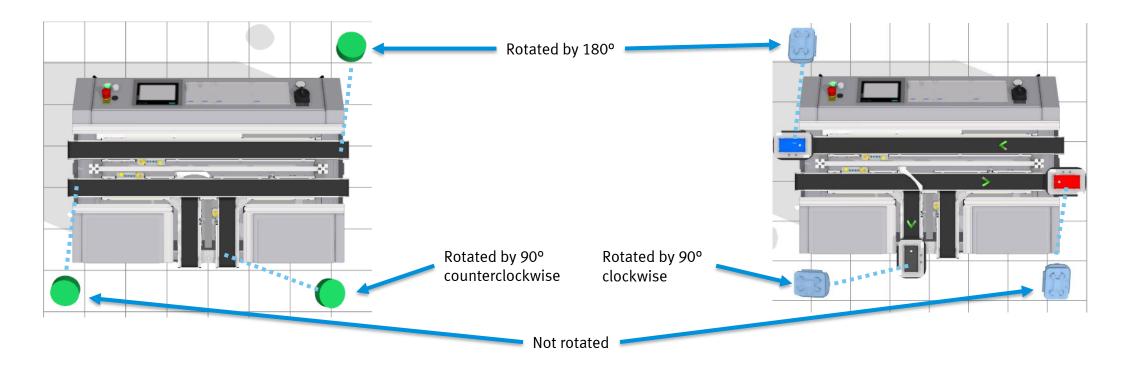




### **Sources and Sinks**

#### Remarks

• Depending on the model, sources and sinks may have to be rotated to operate correctly!





# **Sources and Sinks**

Properties	Description
Automatic Mode	If enabled carriers / pallets / workpieces within the range of the sink will be removed automatically during simulation.

Properties		
General Pose	Property □ CP-F-Sink	Value
Dimension Visualization Extended Info ORL Bus client Collision detection Connectors Gamepad IO Fault Measuring Physics simulation Process geometry CP System	Automatic Mode	Enabled

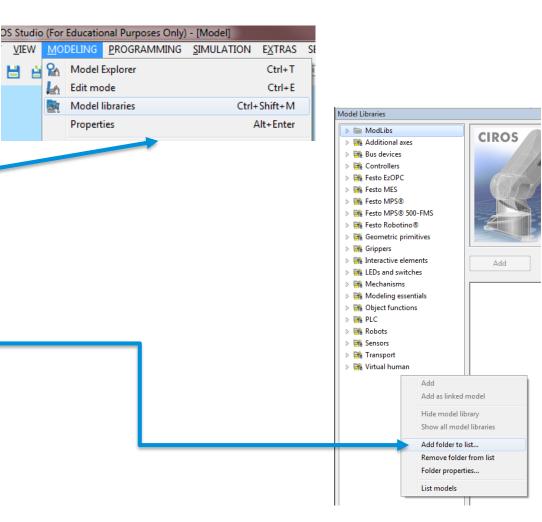


# **Adding New Libraries to CIROS**

Open an already existing model or create a new CIROS model like 1. shown before.

Note: Model libraries window can only be opened when a CIROS model is opened or created!

- Open the model libraries window 2.
  - MODELING → Model libraries or
  - CTRI + SHIFT + M
- Click on the right mouse button within the tree view part of the model 3. libraries to open the corresponding context menu
- Execute Add folder to list... 4.
- Select the folder in which the model library to be added is stored and 5. press OK



VIEW

ы

2

Lo.



# **Virtual Commissioning with MES4**

**Basic knowledge in Festo MES4 and Fleet Manager is required.** 



#### MES4 v2 and below

1. Create a new CIROS project.

Video tutorial: 10_CreateNewCIROSModel.mp4

- Build a CIROS model from model libraries.
   Video tutorial: 11_BuildAModelFromModelLibraries.mp4
- 3. Configure MES4.

Video tutorial: 12_MES4ForCirosModel.mp4

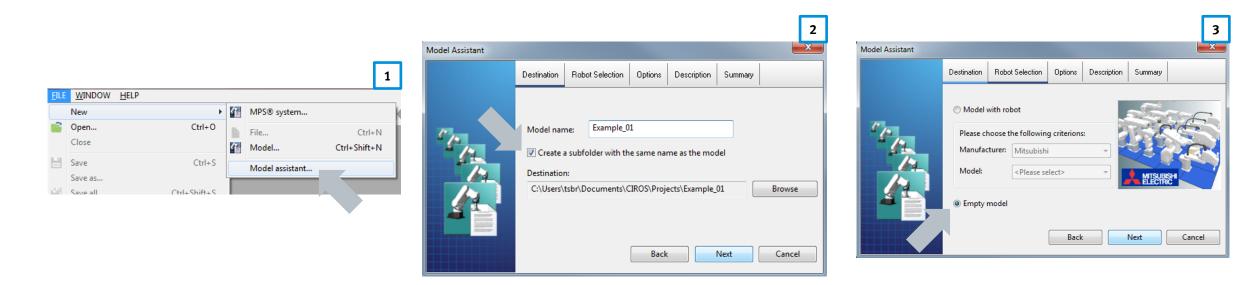
- 4. Configure CIROS model respective to the MES4 configured. <u>Video tutorial</u>: 13_ConfigureCirosModelForMES4.mp4
- 5. Run the simulation

Video tutorial: 14_SimulateAnMES4Order.mp4



#### **1. Create a new CIROS project.**

- 1. Choose FILE  $\rightarrow$  New  $\rightarrow$  Model assistant
- 2. Specify the model's name and enable Create a subfolder with the same name as the model.
- 3. Important: Do not select a robot, these ones are not the ones integrated within CP Lab / Factory! Choose Empty model instead!





#### 2. Build a CIROS model from model libraries.

- 1. Switch to Top View within the list of views.
- 2. Open the model libraries.
- Select all modules needed and add them to the model.
   Note: take care, that components properly snap into place.
- 4. After adding all modules, select each passive deflection and snap it into place again!
- 5. Depending on the model,
  - 1. Integrate sources and sinks if required.
  - 2. Adjust floor and background.
  - 3. Disable shadow simulation

### **3. Configure MES4.**

- 1. According to the CIROS model add all resources in MES4.
  - Application modules and Robotino.
  - CP Lab branches (Remark: CP Factory branches are defined implicitly by the MES4 topology) .
  - Define the MES4 ID, IP address, type of PLC (i.e. Siemens) of each resource.
  - Specify the system's topology.
- 2. Start the CIROS simulation and check in MES4 Production Control  $\rightarrow$  Resources whether all resources are available and active.
- 3. Define all parts that are required by the various work plans.
- 4. Add work plan(s).
- 5. In case a high-bay warehouse is part of the model, specify the initial load of the corresponding buffers.
- 6. Clear all other buffers (branches, Robotino!).
- 7. Start a production or customer order and check that everything works fine.



Picture source: MES4 v2 or lower



4. Configure CIROS model respective to the MES4 configured.

- 1. Configure the CP System options for each component.
  - 1. MES ID in CIROS should be the same as Resource ID in MES4.



### Virtual Commissioning with FactoryViews^[1]

#### Import Model from Python Script

- Python script to generate model with configured MES4 resource ID can be generated from Festo Factory View, which is the successor of Festo MES4 Software.
- The script can be executed in CIROS to configure the CP System setup for virtual commissioning.
- With this option, time to insert the models and to configure the resource IDs is saved.
- However, the carriers are not generated from the code. Thus, they have to be added manually from Festo CP System Model Library.



### Virtual Commissioning with FactoryViews^[1]

#### **Guide to import model from python script**

- 1. Create a CIROS model.
- 2. Select Extras  $\rightarrow$  Python  $\rightarrow$  Execute.
- 3. Select the python script generated, for example, from Factory View, and click open.

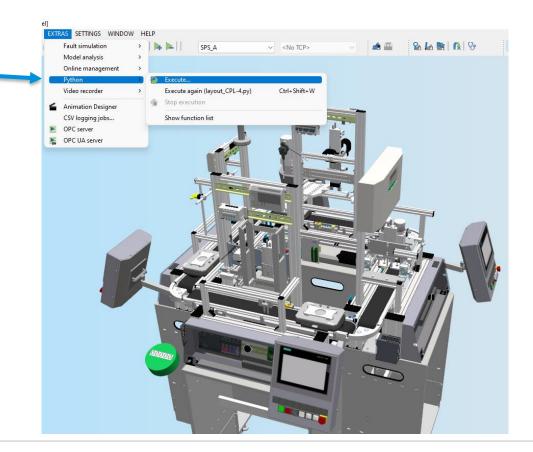
Note: Last python script can be executed again by selecting Extras  $\rightarrow$  Python  $\rightarrow$  Execute Again or shortcut key Ctrl+Shift+W.

- 4. Go to top view or shortcut key A.
- 5. Open Model Library or shortcut key Ctrl+Shift+M.
- 6. Insert carriers and delete extra carriers.
- 7. Snap the remaining carriers in place.
- 8. The model is ready.

#### Hint:

- In the python script exported from FactoryViews, the modules are snapped in place and MES ID of the resources are set.
- To reduce graphic power consumption and avoid crash, close all windows and minimize Model Window before loading the script.

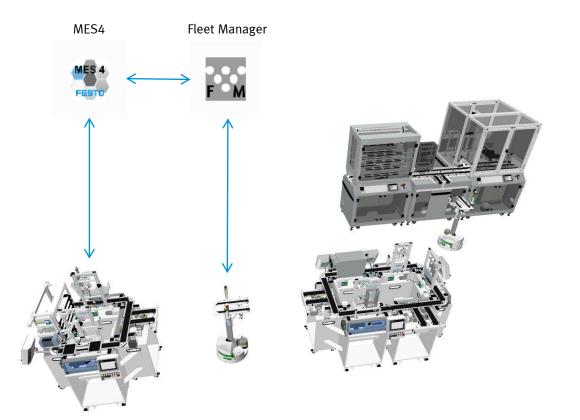
[1] FactoryViews is the new software bundle for MES and web based services. It contains MES4 v3.



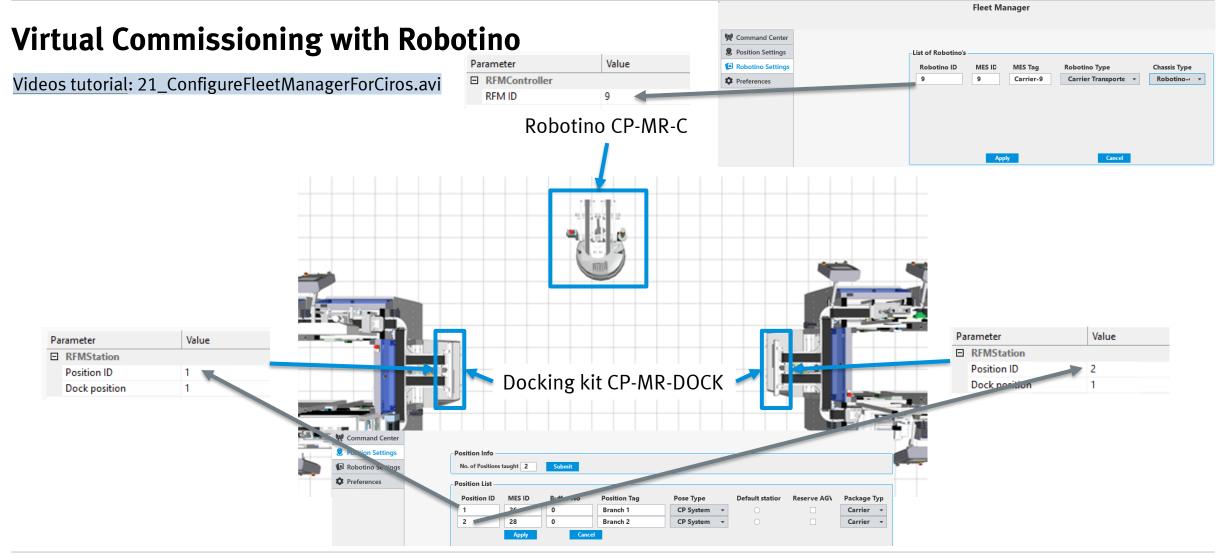


### Virtual Commissioning with Robotino

- Note: MES4 is not communicating with Robotino(s) directly!
- Communication is carried out via the Fleet Manager.
- MES4 is just sending transportation orders like "Go to position A, grab a workpiece, move to position B, and release the workpiece over there" to the Fleet Manager
- Fleet Manager itself selects one of the available Robotino(s) and sends commands like "dock to position A" to the Robotino to fulfill the MES4 order.





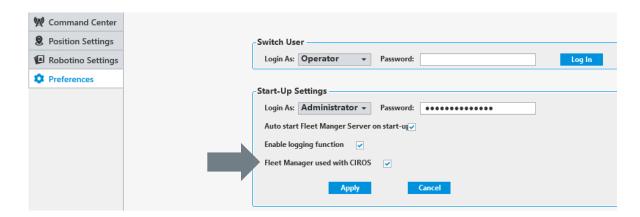




### Virtual Commissioning with Robotino

#### Fleet Manager v3 and above

• Option Fleet Manager used with CIROS must be activated.



Fleet Manager (CIROS-MODE only)				
	real and the second	ESTO		
👷 Command Center				
Position Settings	Info For Fleet Manager MES System Information			
Robotino Settings	No. of Robotino's: 1 MES System IP Address: 127.0.0.1			
Preferences	Fleet Manager Port No:         13000         Start Server         Stop Server         MES System Port No:         2000         Connect         Disconnect			



### **Virtual Commissioning with Robotino**

### Simulation with CIROS, MES4 and Fleet Manager

- Start simulation
  - 1. Start MES4, CIROS, and Fleet Manager in any order, but **do not** start the CIROS simulation yet.
  - 2. In Fleet Manager, if the server is not started, start the Robotino server via the Start Server button.
  - 3. Start the CIROS simulation.
  - 4. Fleet manager: Select all available Robotinos and switch them to Automatic.
  - 5. Place your MES4 orders.

#### Videos tutorial:

01_StartUpRobotino.mp4 22_SimulationOfRobotino.avi

- Stop simulation
  - 1. Stopping simulation in CIROS.
  - 2. Reset the CIROS simulation to t=0s.
  - 3. Fleet manager: Stop the Robotino server via Stop Server.

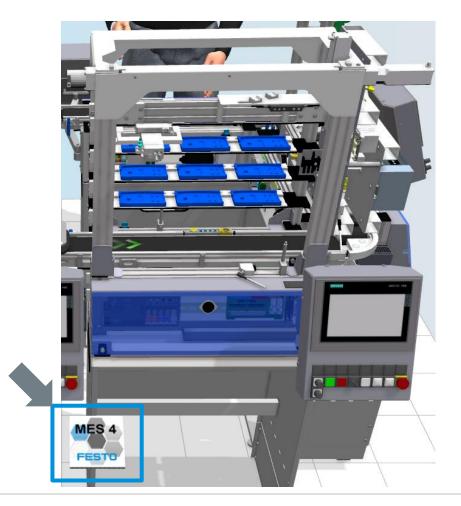
#### **Note: Server must be stopped!**

4. MES4: Clear all Robotino-related buffers.



### **Synchronise CIROS Parts in Storage with MES4 Buffers**

- During the initialization phase of a CIROS simulation run, CIROS will ask for the buffer contents of high-bay warehouses automatically.
- But, whenever the buffer of a high-bay warehouse has changed within the MES4 while the CIROS simulation is running, one must click the MES 4 button in CIROS view window to transmit these changes to CIROS.





# **Running CIROS and MES4 on Different PCs**

### $CIROS \rightarrow Settings \rightarrow Model Options$

Model Options				? ×	
Collision detection Display Fault simulation	Interfaces > MES				
> General	Connection				
✓ Interfaces	Don't activate conne	ction			Address = IP-Address of PC
MES	Address:	127 . 0 . 0 . 1			
RFM > Logging	Timeout:	15 s			which MES4 is running.
Model Explorer > Modeling	Message server				
> Online management	Port:	2000			
ORL Overlays	Field definition file:	<ciros>\HeaderSend.xml</ciros>			
> Peripherals	State server				
> Simulation Video recorder	Port:	2001			
	Cycle time:	1000 ms			
	Status- und Fehlermele	lungen (JSON)			
	Aktivieren				
	Port:	8989			
		2000 ms			
	Zykluszeit:	2000 ms			
·	I I		ОК	Cancel Apply	



# **Running CIROS and Fleet Manager on Different PCs**

#### $CIROS \rightarrow Settings \rightarrow Model Options$

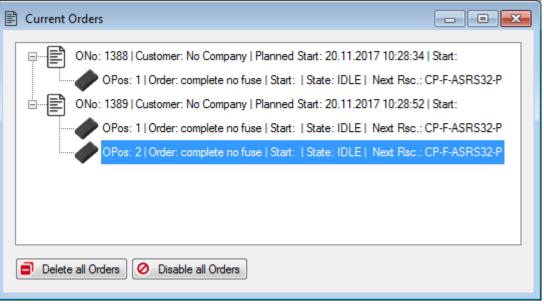
Model Options			? ×
Collision detection > Display Fault simulation > General V Interfaces MES RFM	Interfaces > RFM Connection Address: Timeout:	127 . 0 . 0 . 1 15s	 Address = IP-Address of PC which Fleet Manager is running.
<ul> <li>&gt; Logging Model Explorer</li> <li>&gt; Modeling</li> <li>&gt; Online management ORL Overlays</li> <li>&gt; Peripherals</li> <li>&gt; Simulation Video recorder</li> </ul>	Message server Port:	13000	



### **Terms & Definitions in MES4**

#### Customers, orders, order number & order position

- MES4 maintains a list of registered customers which are allowed to place customer orders.
- Each order has a unique order number (ONo) and may consist of a couple of different products and parts to be produced.
- Each production part within an order has its own order position (OPos), ranging from 1 up to the total number of parts of a particular order.
- (ONo, OPos) is a unique representation of an individual part.



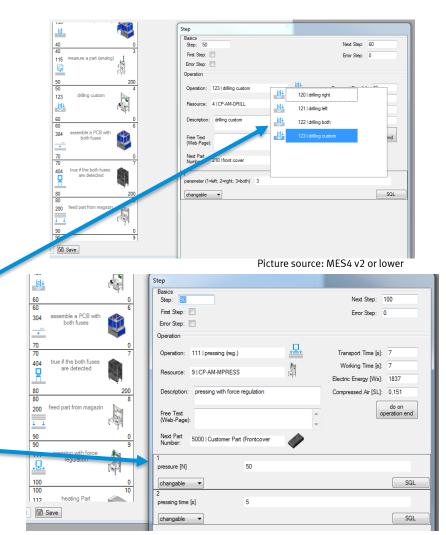
Picture source: MES4 v2 or lower

#### FESTO

## **Terms & Definitions in MES4**

#### **Operations and parameters**

- Operations define the functionality of a production step and are executed by resources.
- But, MES4 separates between operations and resources, since there might be several resources able to perform the same type of operation, i.e., operations are not sub-objects of resources but allocated to them.
- Vice versa, some resources can execute more than just one 
  operation.
- Each operation has its own unique ID (OpNo) and might have no, one or even quite several parameters to adjust the production step.



Picture source: MES4 v2 or lower



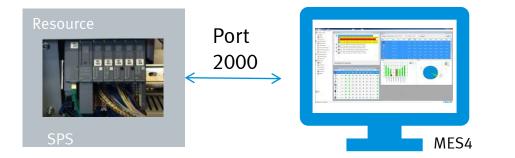
## **MES4** Communication Interface

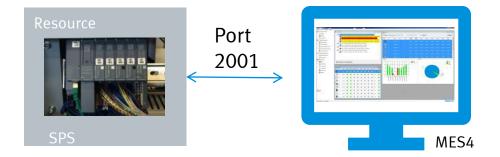
### Cyclic status message

• The resources send a status update to MES4 in every second.

#### **Service requests**

• Resources or other applications query data from MES4 or write data to MES4.







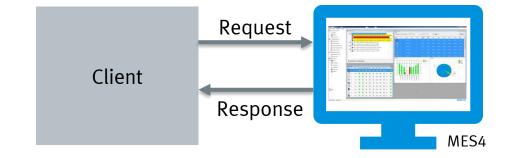
## **Terminology in MES4 Messages**

Term	Meaning
BoxID	Box ID
BoxPNo	Box part number
BoxPos	Position in box
BufNo	Buffer number
BufPos	Buffer position
ONo	Order number
OPos	Order position
Ор	Operation
OpEnd	Operation end
OpNo	Operation number

FESTO

### **MES4 Service Requests**

- MES4 offers many services that are required for the operation of a plant.
- None of the plants use all services, but all services are available at any time and can be called not only by PLCs but also by other business applications via a TCP/IP connection. In addition, a skilled operator can implement additional services, if they can be mapped to an SQL query of theMES4 database.
- Service calls always follow the request response paradigm, i.e., a client sends a call and MES4 response back.
- Internal controller of CIROS model is the virtual representation of PLC. Thus, it communicates the same way as a real PLC.





#### Message classification

- Services are uniquely identified by two characteristics:
  - MClass (service class)
  - MNo (service number)
- The MNo is only unique within an MClass. This means that the service with MClass 100, MNo 6 is different from MClass 150, MNo 6.
- They also have a name, but this is of no relevance to the client or to MES4. It is only used for recognition by humans.
- All the messages available can be seen in MES4
  - Tools  $\rightarrow$  Com. Simulator
- The messages can be edited
  - MES4  $\rightarrow$  Tools  $\rightarrow$  Config SQL
  - <u>Note:</u> Only Administrator can access Config SQL
  - Administrator's password is SolutionCenter

#### Example:

MClass	MNo	Name
100	6	GetOpForONoOPos
100	33	GetStepDescription
101	20	OpEnd
150	5	GetBufPos



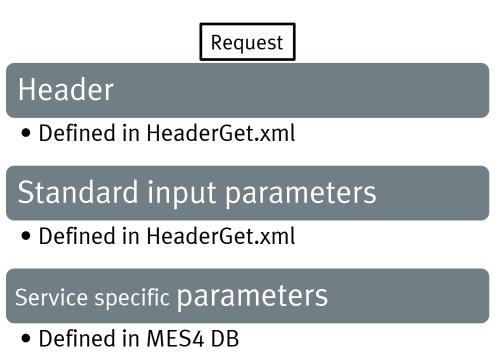
### Classes

• Service classes on the right are frequently used. For user defined services, any other classes can be used.

MClass	Description
100	Get information from orders and work plans
101	Write information to orders and work plans
110	Request topology related data
150	Request buffers and utilities (incl. boxes) status
151	Write buffers and utilities (incl. boxes) status
200	Request logistic and Robotino information
201	Write logistic and Robotino information



#### Message packet overview



### Response

## Header

• Defined in HeaderSend.xml

## Standard output parameters

• Defined in HeaderSend.xml

### Service specific parameters

• Defined in MES4 DB



### Data coding

- MES4 allows two different encoding procedures for service requests and responses.
- The first three or four bytes of each packet are TCP Ident header, which indicates which method the packet uses.

#### • Binary coding

- Binary coding is primarily used for communication with PLCs on which it is fundamentally easier to handle fixed-address binary data than strings.
- In the binary procedure, a distinction is still made between the Siemens format and the CODESYS format. MES4 also responds to each binary-coded request in the binary procedure of the same format.

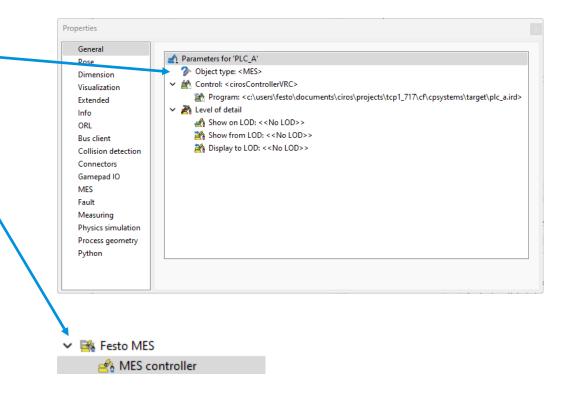
#### • String coding

- Well suited for implementation in high-level languages or for manual tests. Parameter names and values are transmitted in human-readable form.
- The string procedure also has two forms.
  - The complete format can be used for both calls and responses.
  - The abbreviated format is only used in MES4 responses if this was requested in the call. was requested.

FESTO

## **Festo MES4 Interface**

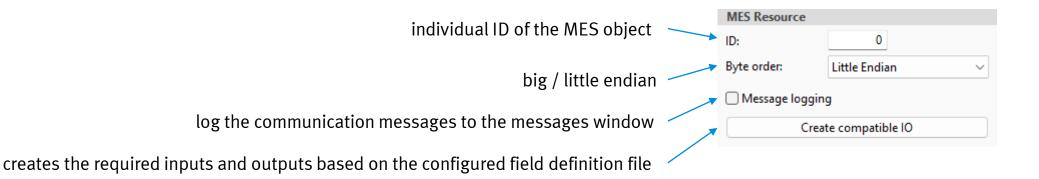
- The communication between the MES4 and the CIROS model requires an object of type MES in the CIROS model.
- A corresponding object named MES Controller can be found in the CIROS model library Festo MES.
- The MES object from the model library already has all required communication inputs and outputs. Those inputs and outputs shall are only be used for external communication and should not be connected to other inputs or outputs in CIROS.





## **Festo MES4 Interface**

- The communication with the MES object inside CIROS is exclusively done via IRL functions. These functions are provided by a program named MES_commands.irl which is automatically added to a Programs subfolder of your model folder when you add the MES object. The required functions can be made accessible in your own IRL program by stating FROM MES_commands IMPORT ALL;
- By default, the MES4 interface uses field definitions from the HeaderSend.xml file in the CIROS root directory. If you need own field definitions, these can be configured in the model options. After configuring field definitions, go to the MES property page of the MES object and press the Create compatible IO button to create the required object inputs and outputs.





- At the end of this tutorial, user is able to update to MES4 that the resource is in Automatic, MES mode when simulation starts.
- 1. Create a CIROS project.
- 2. In Model Libraries, insert following:
  - 1. Festo MES \ MES controller.
  - 2. Controllers \ Simulation controller
- 3. Open Project Management window.
  - 1. Create a new project in Industrial Robot Language (IRL) in location <project folder>\Programs. Name it ResourceState.prjx.
  - 2. Assign the project to controller MES_Controller.
  - 3. Add MES_commands.irl in the project.
  - 4. Create a new program. Name it ResourceState.irl. Make it main program.
  - 5. Program ResourceState.irl as in next page.
  - 6. Save and compile the program.
  - 7. Save the project.
  - 8. Run simulation and observe in MES4 Production Control \ Resources.



ResourceState.irl

PROGRAM ResourceState;

FROM MES_commands IMPORT ALL;

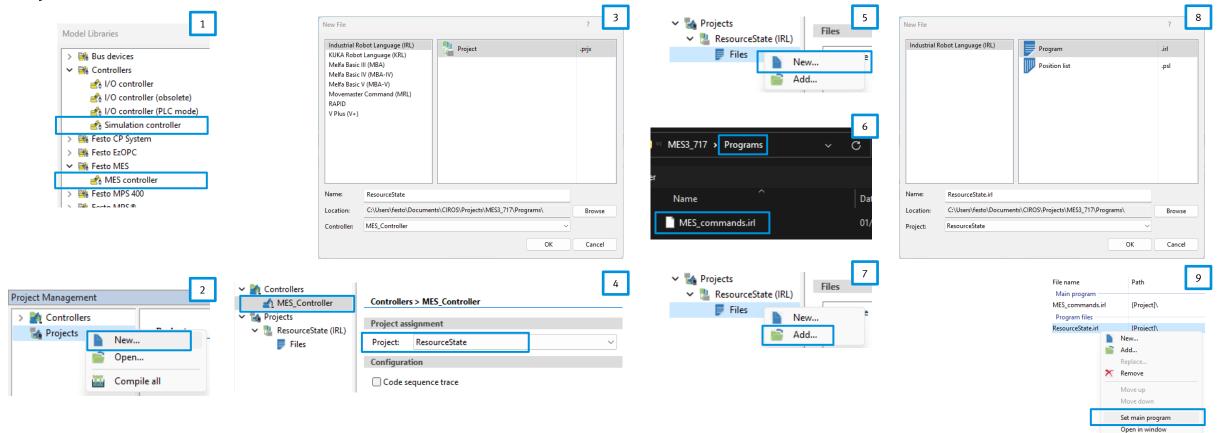
BEGIN

{Initiate MES status}
setStateMESMode(true);
setStateAuto(true);
setStateReset(false);
setStateError(-1);

ENDPROGRAM



### **Steps in screenshots**

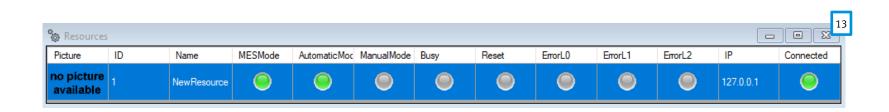




#### **Steps in screenshots**

2	💾 🔐   🐝 🖷 📕   🕐 -   🔤
-L	
[ME	S_Controller] .\Programs\ResourceState.irl * PROGRAM ResourceState;
2	
3 4	FROM MES_commands IMPORT ALL;
5	BEGIN {Initiate MES status}
7	<pre>setStateMESMode( true );</pre>
8 9	<pre>setStateAuto( true ); setStateReset( false );</pre>
10	<pre>setStateError( -1 );</pre>
11 12	ENDPROGRAM;
13	ENDEROGRAFI,







### **Message Request from CP System to MES4**

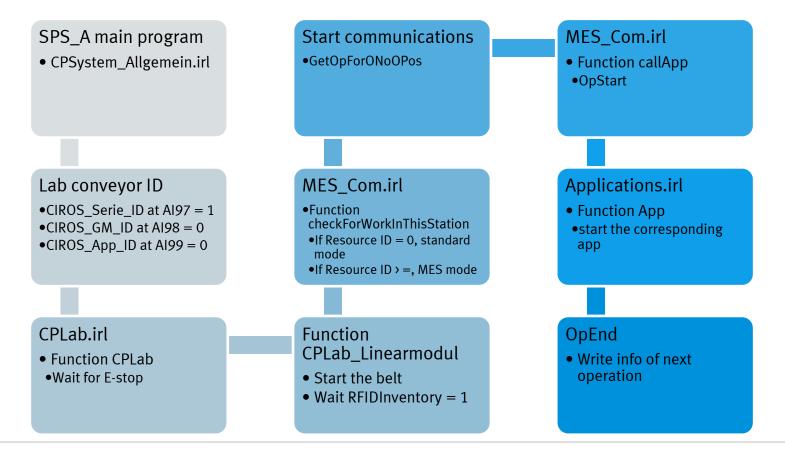
- CIROS behaves the same as PLC in terms of MES4 communication. Thus a bit coding is used.
- The messages are written in IRL format and can be found in Project Management.
  - 1. In toolbar, open Programming \ Project Management.
  - 2. In Project Management window, if not yet exist, add following project.
    - <project folder\CF\CPSystems\CPSystem_Allgemein.prjx>
  - 3. Open Projects \ CPSystem_Allgemein(IRL) \ Files.

Project Management			
🗸 🛃 Controllers			
🛃 SPS_A	Projects > CPSystem_Allge	mein (IRL) > Files	
🛃 SPS_B	<b>F</b> 'l		
🗸 🍓 Projects	Files		
✓ L CPSystem_Allgemein (IRL)	File name	Path	
🗾 Files	Main program		
	CPSystem_Allgemein.irl	[Project]\	
	Program files		
	MES 200.irl	[Project]\Shared\	
	RFID_100.irl	[Project]\Shared\	
	Applikation_Lager.irl	[Project]\Shared\	
	Applikation_LabLager.irl	[Project]\Shared\	
	Applikation_Roboter.irl	[Project]\Shared\	
	Applikationen.irl	[Project]\Shared\	
	MES_Com.irl	[Project]\Shared\	
	CPLab.irl	[Project]\Shared\	
	CPLabBridge.irl	[Project]\Shared\	
	CPLabLager.irl	[Project]\Shared\	
	CPFactory.irl	[Project]\Shared\	
	CPFactoryWeiche.irl	[Project]\Shared\	
	CPFactoryByPass.irl	[Project]\Shared\	
	CPFactoryLager.irl	[Project]\Shared\	
	CPFactoryRoboter.irl	[Project]\Shared\	
	CPBoxLager.irl	[Project]\Shared\	
	CPBoxBuffer.irl	[Project]\Shared\	
	CPBoxBuffer.irl	[Project]\Shared\	



## **MES** Communication Flow Chart

#### **Example: CP-L-CONVEYOR**





### **Message Request from CIROS to Fleet Manager**

- Like PLC, CIROS Robotino model is the virtual representation of real Robotino. Thus it communicates with Fleet Manager the same way as a real Robotino.
- The communication messages are written in IRL format and can be found in CIROS  $\rightarrow$  Programming  $\rightarrow$  Project Management  $\rightarrow$  Projects  $\rightarrow$  RFM_MR.

Project Management		
<ul> <li>Controllers</li> <li>RFMController</li> <li>SPS_A</li> <li>SPS_B</li> </ul>	Projects > RFM_MR (IR Files	L) > Files
<ul> <li>Projects</li> <li>CPSystem_Allgemein (IRL)</li> </ul>	File name	Path
V NRFM_MR (IRL)	Main program	[Project]\
	Program files RFM_150.irl	[Project]\



# Virtual Commissioning with Soft PLC

Basic knowledge in PLC programming and TIA Portal is required.



## **Scenario Overview**

- Program your PLC against a virtual mechatronic model
- No risk to your hardware if students make mistakes in program code
- Program modules that you don't physically own or let dozens of students program the same module even if you only own it once



## **Scenario Overview**

### All on single PC



#### **Software on different PCs**





### **Process Summary**

- 1. Prepare a CIROS model with the hardware you want to program
- 2. Create your hardware configuration and I/O tags in TIA Portal
- 3. Create a PLCSIM Advanced instance and download the hardware configuration
- 4. Configure the interface between CIROS and your instance
- 5. Start programming!

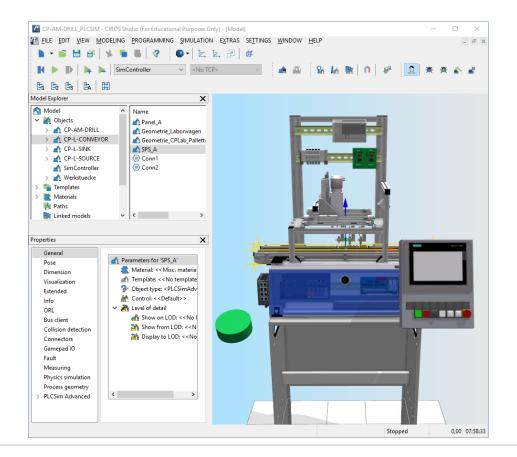
#### **Important:**

- CIROS v7.1 is only compatible with Siemens **PLCSIM Advanced v3.0** or above!
- CIROS v7.0 or below is only compatible with Siemens PLCSIM Advanced v2.0 or below.



## **Preparing a CIROS Model**

- Two approaches are possible:
- 1. Create a model from scratch
  - Maximum flexibility
  - Program any CP station you like
- 2. Load a premade model from the model library
  - Get started quickly with minimum effort
  - Limited selection of CP systems available

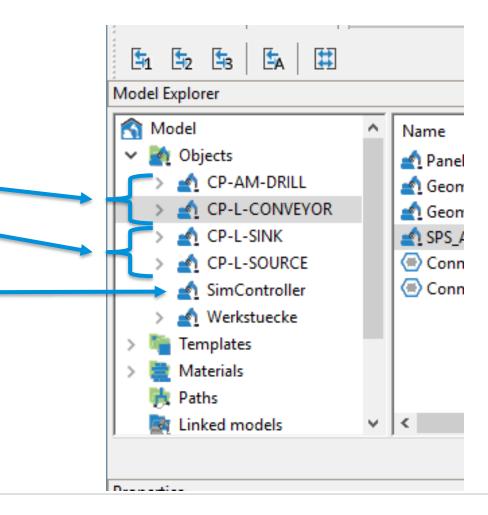




## **Preparing a CIROS Model**

Your model usually needs three basic elements to serve for virtual commissioning with PLCSIM Advanced:

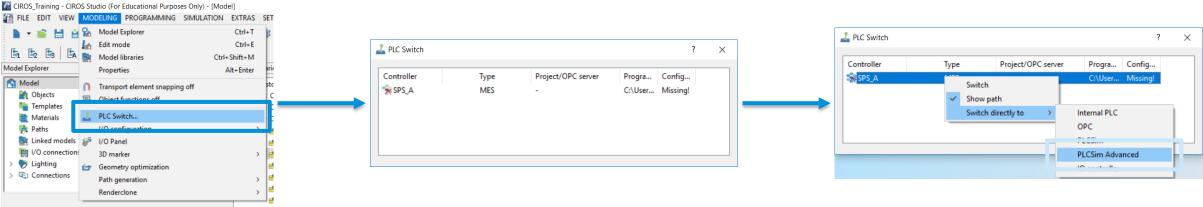
- The mechatronic system you want to program
- A source and sink to generate and remove carriers with parts
- A SimController so CIROS is able to simulate your model





### Preparing a CIROS model from scratch

- 1. Create a new empty model.
- 2. Add your mechatronic system from the model library. For this exercise, add a CP-L-CONV.
- 3. Add a source and sink from the model library that matches your system. For this exercise, add a CP Lab source and sink.
- 4. Connect the source and sink to your CP Lab module.
- 5. Switch the PLC in your CP Lab module (it's named SPS_A) to PLCSIM Advanced mode.





## **Starting a PLCSIM Instance**

Feel free to choose any PLCSIM Advanced settings that work for you. The only setting relevant to CIROS is the instance name. Choose one you like and remember it. You'll need it later.

#### Some recommendations:

- For Online Access, choose PLCSIM unless your simulated PLC needs to communicate over the network. This mode makes the connection to TIA Portal effortless
- Leave Time Scaling off. CIROS has its own time scale and will make sure the PLC keeps track if you speed up the simulation beyond real-time
- Choose ET 200SP for PLC type as that matches the physical PLC in most CP hardware systems

	- 🗆 X
ΡĹ	S7-PLCSIM Advanced V2.0 SP1
SI	M Control Panel
	Online Access  PLCSIM  PLCSIM Virtual Eth. Adapter
•	
Q	Virtual Time Scaling
	0.01 Off 100
	Start Virtual S7-1500 PLC
•	Instance name CIROS_Training
	PLC type Unspecified ET 200SP V
	Start
	MRES MRES
	No Active PLC Instance
	ria
	Drop Instances Here
<b>a</b>	Runtime Manager Port 50000
Ň	Virtual SIMATIC Memory Card
i	Show Notifications
?	Function Manual
8	Exit

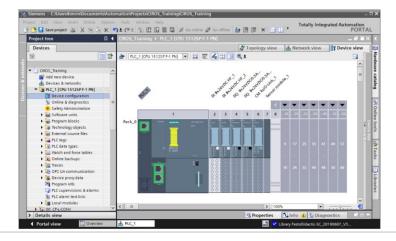


### **Creating the Hardware Configuration and IO Tags in TIA Portal**

#### Hardware configuration

You can configure your PLC in any way you like.

Ideally, it should have at least the number of digital and analog I/Os that the physical PLC inside your chosen CP system has.



#### IO tags

You can freely name your inputs and outputs, as long as the address and the type of an input/output is correct.

If you like you can skip the inputs and outputs that are not connected to anything in your CP system.

Refer to the Festo Didactic Infoportal (<u>https://ip.festo-</u> <u>didactic.com</u>) for an I/O listing of your CP system. Alternatively, find the relevant information in your manual or circuit diagram.

Yoject tree     Xove project ▲ X 1 →      Xove project ▲ X 1 →      Xove project tree     Yoject tree			raining > PLC_1 [CPU				_			- 6	7 8
Devices								- Tags	O U	ser consta	ants
19	1 😰 🗄	0 -0	⇒ ₩ ?? #								В
		myTa									-
CIROS_Training			Name	Data type	Address	Retain	Acces	Write	Mathi	Supervis	Com
Add new device	<u></u>		xStartButton	Bool	Sil1.0	Ketain	Acces.	. write		supervis	Com.
Devices & networks			StopButton	Bool	51.2		ă	Ĩ	×.		
<ul> <li>Devices a networks</li> <li>PLC_1 [CPU 15125P F-1 PN]</li> </ul>		-	<add news<="" td=""><td></td><td>1611.2</td><td></td><td></td><td></td><td></td><td></td><td></td></add>		1611.2						
Device configuration			<pre><pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre>&gt;<pre><pre< td=""><td></td><td></td><td></td><td><b>V</b></td><td>(V)</td><td>V</td><td></td><td></td></pre<></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre>				<b>V</b>	(V)	V		
Conline & diagnostics											
Safety Administration											
Software units											
Program blocks											
Technology objects											
External source files	- 16										
<ul> <li>PLC tags</li> </ul>											
Show all tags											
Add new tag table											
Default tag table [53]											
myTags [2]											
PLC data types											
Watch and force tables											
Online backups	- 16										
Traces											
OPC UA communication											
Device proxy data	- 1 F										
Program info		<									



### Creating the hardware configuration

Detailed instructions how to do the hardware configuration in TIA Portal are beyond the scope of this document. Refer to the courseware 'Device configuration' if you're having trouble.

For this exercise, we're configuring the PLC as we would a real CP-L-CONV module with a Siemens IO-Link 1.1 conformant RFID device.

- 1. Create a new TIA project
- 2. Add a S7-1512SP F-1 PN PLC to your project (6ES7 512-1SK01-0AB0)
- 3. Add two DI 8x24VDC HF (6ES7 131-6BF00-0CA0)
- 4. Add two DQ 8x24VDC/0.5A HF (6ES7 132-6BF00-0CA0)
- 5. Add a CM 4xIO-Link (6ES7 137-6BD00-0BA0)
- 6. Add a server module (6ES7 193-6PA00-0AA0)
- 7. Set the IO-Link master's input/output type to 64/64 and shift the starting I/O addresses to address 10



Creating the hardware configuration





### Setting up the I/O tags

- 1. Find the list of I/O addresses for the CP-L-CONV at <u>https://ip.festo-</u> <u>didactic.com/InfoPortal/CPFactoryLab/hardware/base/datasheet.php?model=CP-L-CONV&lang=en</u>
- 2. Create a new tag table
- 3. Enter all tags listed on the Infoportal into your tag table

#### Note:

Depending on the revision of a physical CP-L-CONV, any address listed on the Infoportal in byte 18 might require to be shifted to byte 42. This is only relevant if you plan to download this TIA project to a real CP-L-CONV. In CIROS the absolute I/O addresses don't matter.



### Setting up the I/O tags

PLOD CONTREPORT (CP-L-CONV)       Detailed       Decomposition       Notes         Cinputs & courtputs       PLC       P	Project Edit View Insert Online Options Tools > Totally Integrated	d Automa P
Call Control (CP-L-CUNV)       Decome Table M       Decome Table M <thdecome m<="" table="" th="">       Decome Table M</thdecome>	CIROS_Training → PLC_1 [CPU 1512SP F-1 PN] → PLC tags → myTags [36]	- 1
Product & coutiputs         ingital inputs         Description       Reference top	Tags 🗉 Us	ser const
Note the addresses       Note the addresses       Note the addresses       Note the addresses         Note - Carrier ident code bit 1 detected       +G1-BG1       %11.2       G4	🚽 🛫 🔮 🕑 🕑 🛗	
Apple Inputs       PLC       Reference to 0       PLC       Reference to 0       Address         TRUE = Start button pushed       +P1.5F1       %11.0       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %       %		
Description       Reference og		
Description         Reference 0         Reference 0 <threference 0<="" th=""> <threference 0<="" th=""></threference></threference>		
Description         Reference to Q         Pic of address           TRUE = Start button pushed         +P1-SF1         %11.0           FALSE = Stop button pushed (n.c.)         +P1-SF2         %11.1           Select operation mode (FALSE = Setup mode, TRUE = A+P1-SF3         %11.2           Automatic mode)         %11.3           TRUE = Carrier detected at stopper position (corresponds to P1-SF5         %11.5           FALSE = Setup mode, TRUE = Carrier detected at conveyor entry         +G1-BG5         %11.6           TRUE = Carrier ident code bit 0         461-BG2         %11.8.1           TRUE = Carrier ident code bit 0 detected         +G1-BG1         %11.8.0           TRUE = Carrier ident code bit 0 detected         +G1-BG1         %11.8.0           TRUE = Carrier ident code bit 0 detected         +G1-BG2         %11.8.1           TRUE = Carrier ident code bit 0 detected         +G1-BG2         %11.8.0           TRUE = Carrier ident code bit 0 detected         +G1-BG2         %11.8.0           TRUE = Carrier ident code bit 0 detected         +G1-BG2         %11.8.0           TRUE = Carrier ident code bit 0 detected         +G1-BG2         %11.8.0           TRUE = Carrier ident code bit 1 detected         +G1-BG2         %11.8.0           TRUE = Carrier idetetcod at conveyor entit         +G1-BG2 <t< td=""><td></td><td></td></t<>		
TRUE = Start button pushed       +P1-SF1       %11.0         FALSE = Stop button pushed (n.c.)       +P1-SF2       %11.1         Select operation mode (FALSE = Setup mode, TRUE =       +P1-SF3       %11.2         Automatic mode)       +P1-SF3       %11.2         TRUE = Carrier detected at stopper position (corresponds to bit 0       +G1-BG5       %11.6         TRUE = Carrier detected at conveyor enty       +G1-BG5       %11.6         TRUE = Carrier detected at conveyor enty       +G1-BG5       %11.6         TRUE = Carrier detected at conveyor enty       +G1-BG5       %11.6         TRUE = Carrier detected at conveyor enty       +G1-BG5       %11.6         TRUE = Carrier detected at conveyor enty       +G1-BG5       %11.6         TRUE = Carrier detected at conveyor enty       +G1-BG5       %11.6         TRUE = Carrier detected at conveyor enty       +G1-BG5       %11.6         TRUE = Carrier ident code bit 0       %11.8       %11.8         RUE = Carrier ident code bit 0       %11.8       %11.8         RUE = Carrier ident code bit 0       %11.8       %11.8         RUE = Carrier ident code bit 0       %11.8       %11.8         RUE = Carrier ident code bit 1       %11.8       %11.8         RUE = Carrier ident code bit 3       %11.8	2 4 4 xSF4 Bool %11.3	1
TRUE = Start button pushed       +P1-SF1       %11.0         FALSE = Stop button pushed (nc.)       +P1-SF2       %11.1         Select operation mode (FALSE = Setup mode, TRUE =       +P1-SF3       %11.2         Automatic mode       +P1-SF4       %11.3         TRUE = Carrier detected at stopper position (corresponds to b 461-861       %11.6         TRUE = Carrier detected at conveyor entry       +61-865       %11.6         TRUE = Carrier ident code bit 3 detected       +61-863       %11.6         TRUE = Carrier ident code bit 3 detected       +61-863       %11.6         TRUE = Carrier ident code bit 3 detected       +61-863       %11.6         TRUE = Carrier ident code bit 3 detected       +61-863       %11.6         TRUE = Carrier ident code bit 3 detected       +61-863       %11.6         TRUE = Carrier ident code bit 3 detected       +61-863       %11.6         TRUE = Carrier ident code bit 3 detected       +61-863       %11.6         TRUE = Carrier ident code bit 3 detected       +61-863       %11.6         TRUE = Carrier ident code bit 3 detected       +61-863       %11.6         TRUE = Carrier ident code bit 3 detected       +61-863       %11.6         TRUE = Carrier ident code bit 3 detected       +61-863       %11.6         TRUE = Carrier ident cod	5 400 x8G1 Bool %11.4 🗌 🗹 🗹	
FALSE = Stop button pushed (n.c)       +P1-SF2       %11.1         Select operation mode (FALSE = Setup mode, TRUE =       +P1-SF3       %11.2         Automatic mode)       +P1-SF4       %11.3         TRUE = Carrier detected at stopper position (corresponds to ind et code bit 0       +P1-SF5       %11.4         FALSE = Setup mode, TRUE =       +P1-SF3       %11.2         Automatic mode)       +P1-SF4       %11.3         TRUE = Carrier detected at stopper position (corresponds to ind et code bit 0       +P1-SF5       %11.4         FALSE = Carrier detected at conveyor entry       +C1-BG5       %11.6         TRUE = Carrier ident code bit 0 detected       +G1-BG1       %11.8         TRUE = Carrier ident code bit 0 detected       +G1-BG1       %11.6         TRUE = Carrier ident code bit 0 detected       +G1-BG1       %11.6         TRUE = Carrier ident code bit 0 detected       +G1-BG1       %11.6         TRUE = Carrier ident code bit 1 detected       +G1-BG3       %11.6         TRUE = Carrier ident code bit 1 detected       +G1-BG3       %11.8         TRUE = Carrier ident code bit 2 detected       +G1-BG3       %11.8         TRUE = Carrier ident code bit 2 detected       +G1-BG3       %11.8         TRUE = Carrier ident code bit 2 detected       +G1-BG3       %11.8 <t< td=""><td></td><td></td></t<>		
FALSE = Stop button pushed (n.c)       +P1-SF2       %11.1         Select operation mode (FALSE = Setup mode, TRUE =       +P1-SF3       %11.2         TRUE = Reset button pushed       +P1-SF4       %11.3         TRUE = Carrier detected at stopper position (corresponds to 0)       +G1-BG1       %11.4         FALSE = Energency stop button pushed (n.c.)       +P1-SF5       %11.5         TRUE = Carrier detected at conveyor exit       +G1-BG1       %11.8         TRUE = Carrier ident code bit 0 detected       +G1-BG1       %11.8         TRUE = Carrier ident code bit 1 detected       +G1-BG1       %11.8         TRUE = Carrier ident code bit 2 detected       +G1-BG1       %11.8         TRUE = Carrier ident code bit 1 detected       +G1-BG1       %11.8         TRUE = Carrier ident code bit 1 detected       +G1-BG1       %11.8         TRUE = Carrier ident code bit 2 detected       +G1-BG1       %11.8         TRUE = Carrier ident code bit 3 detected       +G1-BG1       %11.8         TRUE = Carrier ident code bit 3 detected       +G1-BG1       %11.8         TRUE = Carrier ident code bit 3 detected       +G1-BG1       %11.8         TRUE = Carrier ident code bit 3 detected       +G1-BG1       %11.8         TRUE = Carrier ident code bit 3 detected       +G1-BG1       %11.8	7 40 x8G5 Bool %41.6 ✔ ✔ ✔ 8 40 x8G6 Bool 441.7 ✔ ✔ ✔ ✔	
Select operation mode (FALSE = Setup mode, TRUE =       +P1-SF3       %11.2         Automatic mode)       +P1-SF4       %11.3         TRUE = Carrier detected at stopper position (corresponds to vice)       +G1-BG1       %11.4         FALSE = Emergency stop button pushed (n.c.)       +P1-SF5       %11.5         FRUE = Carrier detected at conveyor ent       +G1-BG5       %11.6         TRUE = Carrier ident code bit 0 detected       +G1-BG1       %11.6         TRUE = Carrier ident code bit 1 detected       +G1-BG1       %11.6         TRUE = Carrier ident code bit 3 detected       +G1-BG1       %11.6         TRUE = Carrier ident code bit 3 detected       +G1-BG1       %11.6         TRUE = Carrier ident code bit 3 detected       +G1-BG1       %11.6         TRUE = Carrier ident code bit 3 detected       +G1-BG3       %11.6         TRUE = Carrier ident code bit 3 detected       +G1-BG3       %11.6         TRUE = Carrier ident code bit 3 detected       +G1-BG3       %11.6         TRUE = Carrier ident code bit 3 detected       +G1-BG3       %11.6         TRUE = Carrier ident code bit 3 detected       +G1-BG3       %11.6         TRUE = Carrier ident code bit 3 detected       +G1-BG3       %11.6         TRUE = Carrier ident code bit 3 detected       +G1-BG3       %11.6	9 40 x8G1_8CD0 8ool %42.0	
Select operation mode (N2) = 3 etup mode, IRUE = 4P1-51-3       %11.2         TRUE = Reset button pushed       +P1-51-4       %11.3         TRUE = Reset button pushed       +P1-54       %11.4         TRUE = Carrier detected at tooper position (corresponds to 10)       +G1-BG1       %11.5         TRUE = Carrier detected at conveyor exit       +G1-BG5       %11.6         TRUE = Carrier detected at conveyor exit       +G1-BG6       %11.7         TRUE = Carrier ident code bit 0 detected       +G1-BG1       %11.8         TRUE = Carrier ident code bit 1 detected       +G1-BG2       %11.8         TRUE = Carrier ident code bit 2 detected       +G1-BG3       %11.8         TRUE = Carrier ident code bit 3 detected       +G1-BG3       %11.8         TRUE = Carrier ident code bit 3 detected       +G1-BG3       %11.8         TRUE = Carrier ident code bit 3 detected       +G1-BG3       %11.8         TRUE = Carrier ident code bit 3 detected       +G1-BG3       %11.8         TRUE = Carrier ident code bit 3 detected       +G1-BG3       %11.8         TRUE = Carrier ident code bit 3 detected       +G1-BG3       %11.8         TRUE = Carrier ident code bit 3 detected       +G1-BG3       %11.8         TRUE = Carrier ident code bit 3 detected       +G1-BG3       %11.8         TRUE		
TRUE = Reset button pushed       +P1-SF4       %11.3         TRUE = Carrier detected at stopper position (corresponds to )       +G1-BG1       %11.4         FALSE = Emergency stop button pushed (n.c)       +P1-SF5       %11.5         TRUE = Carrier detected at conveyor enty       +G1-BG5       %11.6         TRUE = Carrier detected at conveyor exit       +G1-BG5       %11.6         TRUE = Carrier ident code bit 0 detected       +G1-BG5       %11.6         TRUE = Carrier ident code bit 1 detected       +G1-BG3       %11.6         TRUE = Carrier ident code bit 2 detected       +G1-BG3       %11.6         TRUE = Carrier ident code bit 3 detected       +G1-BG3       %11.6         TRUE = Carrier ident code bit 3 detected       +G1-BG3       %11.6         TRUE = Carrier ident code bit 3 detected       +G1-BG3       %11.6         TRUE = Carrier ident code bit 3 detected       +G1-BG3       %11.6         TRUE = Carrier ident code bit 3 detected       +G1-BG3       %11.6         TRUE = Carrier ident code bit 3 detected       +G1-BG3       %11.6         TRUE = Carrier ident code bit 3 detected       +G1-BG3       %11.6         TRUE = Carrier ident code bit 3 detected       +G1-BG3       %11.6         TRUE = Carrier ident code bit 3 detected       +G1-BG3       %11.6       W1	11 🚛 xBG3_BCD2 Bool %42.2	
IKUE = Keest button pushed       +P1-S4       \$\frac{1}{5}\$1.3         TRUE = Carrier detected at stopper position (corresponds to 0)       +G1-BG1       \$\frac{1}{5}\$1.4         FALSE = Emergency stop button pushed (n.c.)       +P1-SF5       \$\frac{1}{5}\$1.6         TRUE = Carrier detected at conveyor entry       +G1-BG5       \$\frac{1}{5}\$1.6         TRUE = Carrier ident code bit 0 detected       +G1-BG6       \$\frac{1}{5}\$1.7         TRUE = Carrier ident code bit 1 detected       +G1-BG6       \$\frac{1}{5}\$1.6         TRUE = Carrier ident code bit 1 detected       +G1-BG3       \$\frac{1}{5}\$1.6         TRUE = Carrier ident code bit 2 detected       +G1-BG4       \$\frac{1}{5}\$1.6         TRUE = Carrier ident code bit 3 detected       +G1-BG3       \$\frac{1}{5}\$1.6         TRUE = Carrier ident code bit 3 detected       +G1-BG4       \$\frac{1}{5}\$1.6         TRUE = Carrier ident code bit 3 detected       +G1-BG4       \$\frac{1}{5}\$1.6         TRUE = Carrier ident code bit 3 detected       +G1-BG4       \$\frac{1}{5}\$1.6         TRUE = Carrier ident code bit 3 detected       +G1-BG4       \$\frac{1}{5}\$1.6         Switch S1 on 'X: TRUE = Encoder A signal detected       +G1-BG4       \$\frac{1}{5}\$1.6         Si on 'X: FRUE = Encoder A signal idetected       +G1-BG4       \$\frac{1}{5}\$1.6       \$\frac{1}{5}\$1.6       \$\frac{1}{5}\$1.6       <	12 🚛 xBG4_BCD3 Bool %42.3 💽 🖌 🖌	
TRUE = Carrier detected at stopper position (corresponds to 1)       +G1-BG1       %11.4         FALSE = Emergency stop button pushed (n.c.)       +P1-SF5       %11.5         TRUE = Carrier detected at conveyor entry       +G1-BG5       %11.6         TRUE = Carrier ident code bit 0)       +G1-BG6       %11.7         TRUE = Carrier ident code bit 0 detected       +G1-BG1       %11.8.1         TRUE = Carrier ident code bit 1 detected       +G1-BG3       %11.8.1         TRUE = Carrier ident code bit 3 detected       +G1-BG3       %11.8.1         TRUE = Carrier ident code bit 3 detected       +G1-BG3       %11.8.1         TRUE = Carrier ident code bit 3 detected       +G1-BG3       %11.8.1         TRUE = Carrier ident code bit 3 detected       +G1-BG3       %11.8.1         TRUE = Carrier ident code bit 3 detected       +G1-BG3       %11.8.1         TRUE = Carrier ident code bit 3 detected       +G1-BG3       %11.8.1         TRUE = Carrier ident code bit 3 detected       +G1-BG3       %11.8.1         TRUE = Carrier ident code bit 3 detected       +G1-BG3       %11.8.1         TRUE = Carrier ident code bit 3 detected       +G1-BG3       %11.8.1         TRUE = Carrier ident code bit 3 detected       +G1-BG4       %11.8.1         TRUE = Carrier ident code bit 3 detected       +G1-BG4	13 🗠 xG1_BG7_KG1 Bool %i42.4 🗌 🗹 🗹	
INDE = Carrier ident code bit 0   FRLE = Carrier ident code bit 1 detected   +G1-BG1   %11.8.2   TRUE = Carrier ident code bit 1 detected   +G1-BG3   %11.8.2   TRUE = Carrier ident code bit 2 detected   +G1-BG3   %11.8.2   TRUE = Carrier ident code bit 2 detected   +G1-BG3   %11.8.2   TRUE = Carrier ident code bit 3 detected   +G1-BG7   %11.8.4 %11.8.4 Note the addresses are shifted from %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4 %11.8.4		
ident code bit 0)     FALSE = Emergency stop button pushed (n.c)   +P1-SF5   %11.5   TRUE = Carrier detected at conveyor entry   +G1-BG5   %11.7   TRUE = Carrier ident code bit 0 detected   +G1-BG1   %11.8.7   TRUE = Carrier ident code bit 1 detected   +G1-BG3   %11.8.3   Switch S1 on 'A': TRUE = Encoder A signal detected   *G1-BG4    *G1-BG4   *G1-BG4   *G1-BG4   *G1-BG4   *G1-BG4   *G1-BG4   *G1-BG4   *G1-BG4   *G1-BG4   *G1-BG4   *G1-BG4   *G1-BG4   *G1-BG4   *G1-BG4   *G1-BG4   *G1-BG4   *G1-BG4   *G1-BG4 <	15	
FALSE = Emergency stop button pushed (n.c.) $+P1-SF5$ $9k11.5$ TRUE = Carrier detected at conveyor entry $+G1-BG5$ $9k11.6$ TRUE = Carrier detected at conveyor exit $+G1-BG6$ $9k1.7$ TRUE = Carrier ident code bit 0 detected $+G1-BG1$ $9k18.0$ TRUE = Carrier ident code bit 1 detected $+G1-BG2$ $9k18.1$ TRUE = Carrier ident code bit 2 detected $+G1-BG2$ $9k18.2$ TRUE = Carrier ident code bit 2 detected $+G1-BG3$ $9k18.2$ TRUE = Carrier ident code bit 3 detected $+G1-BG3$ $9k18.2$ TRUE = Carrier ident code bit 3 detected $+G1-BG3$ $9k18.2$ TRUE = Carrier ident code bit 3 detected $+G1-BG3$ $9k18.2$ TRUE = Carrier ident code bit 3 detected $+G1-BG3$ $9k18.2$ TRUE = Carrier ident code bit 3 detected $+G1-BG3$ $9k18.2$ TRUE = Carrier ident code bit 3 detected $+G1-BG3$ $9k18.2$ TRUE = Carrier ident code bit 3 detected $+G1-BG3$ $9k18.2$ TRUE = Carrier ident code bit 3 detected $+G1-BG3$ $9k18.2$ Switch S1 on 'A': TRUE = Encoder A signal detected $*G1-BG3$ $8k18.4$ Sol to 'S 042.2 $G'$ $G'$ Sol to 'S 042.2 $G'$ Sol to 'S	17 ta byłnalogin0 Byte	
FILSE = Entregency stop buttoh pushed (n.c.)       +P1-sh5       %11.5         TRUE = Carrier detected at conveyor entry       +G1-BG5       %11.6         TRUE = Carrier ident code bit 0 detected       +G1-BG1       %11.8.0         TRUE = Carrier ident code bit 0 detected       +G1-BG2       %11.8.1         TRUE = Carrier ident code bit 1 detected       +G1-BG3       %118.2         TRUE = Carrier ident code bit 2 detected       +G1-BG3       %118.3         Switch S1 on 'A: TRUE = Encoder A signal detected       +G1-BG7       %118.4         Stor R: FALSE = Coupling signal right detected       +G1-BG7       %118.4	18 - Qu byAnalogin1 Byte %IB48 🛛 🗹 🗹	
TRUE = Carrier detected at conveyor exit       +G1-BG3       %11.8         TRUE = Carrier ident code bit 0 detected       +G1-BG1       %11.8         TRUE = Carrier ident code bit 0 detected       +G1-BG2       %11.8.1         TRUE = Carrier ident code bit 1 detected       +G1-BG2       %118.1         TRUE = Carrier ident code bit 2 detected       +G1-BG3       %118.2         TRUE = Carrier ident code bit 3 detected       +G1-BG3       %118.3         Switch S1 on 'A: TRUE = Encoder A signal detected       +G1-BG7       %118.4    Note the addresses are shifted from	19 🚛 xPF1 B, %Q1.0 🗹 🗹 🗹	
TRUE = Carrier detected at conveyor exit       +G1-BG6       %11.7         TRUE = Carrier ident code bit 0 detected       +G1-BG1       %118.0         TRUE = Carrier ident code bit 1 detected       +G1-BG2       %118.1         TRUE = Carrier ident code bit 2 detected       +G1-BG3       %118.2         TRUE = Carrier ident code bit 2 detected       +G1-BG4       %118.3         Switch S1 on 'A': TRUE = Encoder A signal detected       +G1-KG1 oder + G1-BG7       %118.4		
TRUE = Carrier ident code bit 0 detected       +G1-BG1       %11.0         TRUE = Carrier ident code bit 0 detected       +G1-BG2       %118.1         TRUE = Carrier ident code bit 1 detected       +G1-BG3       %118.2         TRUE = Carrier ident code bit 2 detected       +G1-BG4       %118.3         Switch S1 on 'A': TRUE = Encoder A signal detected       +G1-KG1 oder       %118.4         S1 on 'R: FALSE = Coupling signal right detected       +G1-BG7       %118.4		
TRUE = Carrier ident code bit 0 detected       +G1-BG1       %118.0         TRUE = Carrier ident code bit 1 detected       +G1-BG2       %118.1         TRUE = Carrier ident code bit 2 detected       +G1-BG3       %118.2         TRUE = Carrier ident code bit 3 detected       +G1-BG4       %118.3         Switch S1 on 'A': TRUE = Encoder A signal detected       +G1-BG7       %118.4         Stor 'S FLASE = Coupling signal right detected       +G1-BG7       %118.4		
TRUE = Carrier ident code bit 0 detected       +G1-BG2       %118.1         TRUE = Carrier ident code bit 1 detected       +G1-BG3       %118.2         TRUE = Carrier ident code bit 3 detected       +G1-BG4       %118.3         Switch S1 on 'A': TRUE = Encoder A signal detected       +G1-BG7       %118.4         Store files Coupling signal right detected       +G1-BG7       %118.4		
TRUE = Carrier ident code bit 1 detected       +G1-BG3       %118.1         TRUE = Carrier ident code bit 2 detected       +G1-BG3       %118.2         TRUE = Carrier ident code bit 3 detected       +G1-BG4       %118.3         Switch S1 on 'A: TRUE = Encoder A signal detected       +G1-BG7       %118.4         Si on 'R: FALSE = Coupling signal right detected       +G1-BG7       %118.4		
RUE       Califier ident code bit 2 detected       Hold       Mole         TRUE = Carrier ident code bit 2 detected       +G1-BG3       %118.2         TRUE = Carrier ident code bit 3 detected       +G1-BG4       %118.3         Switch S1 on 'A: TRUE = Encoder A signal detected       +G1-KG1 oder       %118.4         1 on 'R: FALSE = Coupling signal right detected       +G1-KG1 oder       %118.4         Stor 'R: FALSE = Coupling signal right detected       +G1-KG1 oder       %118.4		
TRUE = Carrier ident code bit 3 detected       +G1-BG4       %118.3         Switch S1 on 'A': TRUE = Encoder A signal detected       +G1-KG1 oder       %118.4         S1 on 'R': FALSE = Coupling signal right detected       +G1-BG7    Note the addresses          are shifted from	🗤 xPH2_A %Q42.0	
TRUE = Carrier ident code bit 3 detected       +G1-BG4       %I18.3       Note the addresses         Switch S1 on 'A': TRUE = Encoder A signal detected       +G1-KG1 oder       %I18.4       Bool       %Q42.4       W W         S1 on 'R: FALSE = Coupling signal right detected       +G1-BG7       %I18.4       Bool       %Q42.5       W W       W W	28 🚛 xPH2_B Bool %Q42.1 🗌 🗹 🗹	
Switch S1 on 'A': TRUE = Encoder A signal detected +G1-KG1 oder %118.4 +G1-BG7 %118.4 +G1-BG7 %118.4		
Switch S1 on 'A': TRUE = Encoder A signal detected +G1-KG1 oder %618.4 S1 on 'R': FALSE = Coupling signal right detected +G1-BG7 All All All All All All All All All Al	addresses         Rev xPH2_D         Bool         %Q42.3         Image: Control of the second sec	
S1 on 'R': FALSE = Coupling signal right detected +G1-BG7 are shifted from		
	nitted trom	<u> </u>
With Char (b) TOUS Francha B signal data tak Chara (19) TOUS Contraction (19) TOUS CONTR		
Switch S2 on 18': TRUE = Encoder B signal detected; S2 on +G1-KG2 oder %118.5 1': FALSE = Coupling signal left detected +G1-BG8 1': FALSE +G1-BG8 1':		



### **Enabling simulation support**

- 1. Open the project properties.
- 2. On the Protection tab, check Support simulation during block compilation.
- 3. Compile and download it again. It should work without a problem now

Project tree ■ Devices ■ ■ ■ ■ ■ ■ ■ ■	<ul> <li>CIROS_Tr</li> <li>Add n</li> <li>Add new device</li> <li>Add new group</li> <li>Open block/PLC data</li> <li>Open block/PLC data</li> <li>Cut</li> <li>Copy</li> <li>Paste</li> <li>Compile</li> <li>Go online</li> <li>Go offline</li> </ul>	a type F7 Ctrl+X Ctrl+C Ctrl+V ► Ctrl+K Ctrl+M	General Protection Protection	Protection Note that the know-how protection of blocks can be weakened by a simulation.
t a	Pri Compile			
Aug new gevice	🕨 🍃 PL 🏣 Search in project	Ctrl+F		
		F11 Ctrl+P		
	Print preview	Alt+Enter		< Ⅲ > OK Cancel

CIROS Training [Project]



#### **Downloading project to PLC instance**

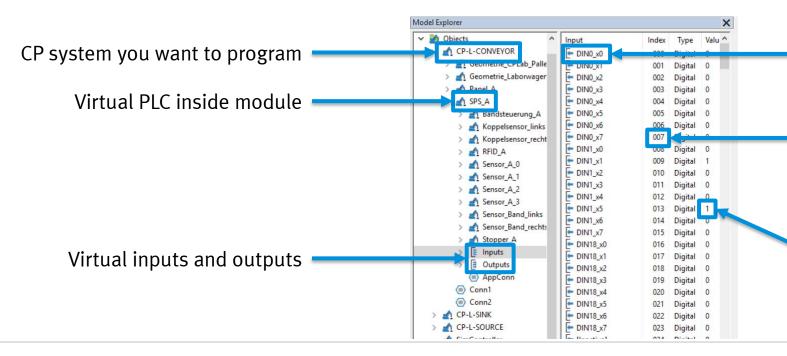
- 1. Compile the project.
- 2. Download it to your simulated PLC. If Online Access in PLCSIM Advanced is set to PLCSIM mode, this is almost fully automatic.

Status	1	Target	Message	Action
<b>+</b> ∎	0	▼ PLC_1	Ready for loading.	Load 'PLC_1'
	0	Simulated module	The loading will be performed from a simulated PLC.	
	0	Software	Download software to device	Consistent download
	0	Text libraries	Download all alarm texts and text list texts to device	Consistent download
<				



## **Configuring the Interface**

The virtual PLC in CIROS has a large number of inputs and outputs, most of which are unused or used for internal processes inside CIROS. A few virtual I/Os correspond to the I/Os of the PLC inside a real CP system, though. These are named DINO_x0 to DIN18_x7 and DOUT0_x0 to DOUT18_x7, after the absolute addresses of the real PLC's I/Os.



Name of a virtual I/O, corresponding to %I0.0 in the real CP module.

I/O address inside CIROS.Completely irrelevant to your TIA project.

Current value of a virtual I/O. For inputs these come from simulated sensors. For outputs, they control a simulated actuator.



## **Configuring the Interface**

You can connect each virtual PLC in your CIROS model to exactly one PLCSIM Advanced instance.

You have to configure our CIROS PLC to connect to the right instance and to hook up the virtual CIROS I/Os to the correct TIA I/Os.

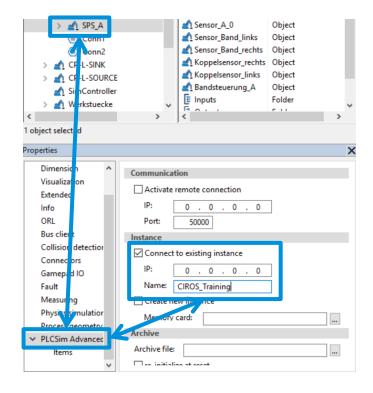
Model Explorer							2			
V 🛃 Objec	ts		^				Ƴ sta	art Virtual S7-1500 PLC		
	P-L-CONVEY	OR				ſ	. [	MRES		
>	Geometrie	CPLab P	alle							
_	Geometrie					1	Activ	e PLC Instance(s):		
_	Danel A		-					CIROS_Training	/ 192.168.0.1	) 🖒 🗙
× 4	SPS_A									
put	Index	Туре	Valu ^			► C	ROS_	Training 🕨 PLC_1 [CF	PU 1512SP F-1 PN] → I	PLC tags 🕨 m
DIN0_x0	000	Digital	0							
DIN0_x1	001	-	0				16 _J6			
DIN0_x2	002	_	0			3	_	🖻 🛃 📽 🛍		
DIN0_x3	003	Digital	0			2	myT	ags		
DIN0_x4	004	Digital	0					Name	Data type	Address 🔺
DIN0_x5	005	Digital	0					xSF1	Bool	%11.0
DIN0_x6	006	Digital	0			8 2	-	XSF2	BOOI	%11.1
DIN0 x7	007	Digital	0		_	<u></u> 3	-	xSF3	Bool	%11.2
DIN1_x0	008	Digital	0			<b>H</b> 4	-	xSF4	Bool	%11.3
DIN1_x1	009	Digital	1		•	5	-	xBG1	Bool	%11.4
DIN1_x2	010	Digital	0			6	-	xSF5	Bool	%11.5
DIN1_x3	011	Digital	0		•	7	-	xBG5	Bool	%11.6
DIN1_x4	012	Digital	0			8	-	xBG6	Bool	%11.7
DIN1_x5	013	Digital	1			9	-	xBG1_BCD0	Bool	%I42.0
DIN1_x6	014	Digital	0		_	10		xBG2_BCD1	Bool	% 42.1
DIN1_x7	015	Digital	0					xBG3_BCD2	Bool	%142.2
DIN18_x0	016	Digital	0			1	-	XBG4_BCD3	BOOI	%142.3
DIN18 v1	017	Digital	٥			13	-	xG1_BG7_KG1	Bool	%142.4
DIN18_x2	018	Digital				14		xG1_BG8_KG2	Bool	% 42.5



### **Configuring the interface**

Configure your CP-L-CONV model to connect to your instance. All of this is done in CIROS. TIA doesn't know anything about the CIROS interface.

- 1. Open the properties of your virtual PLC (SPS_A)
- 2. On the PLCSIM Advanced page, select Find instance by name.
- 3. Enter the name of your PLCSIM Advanced instance



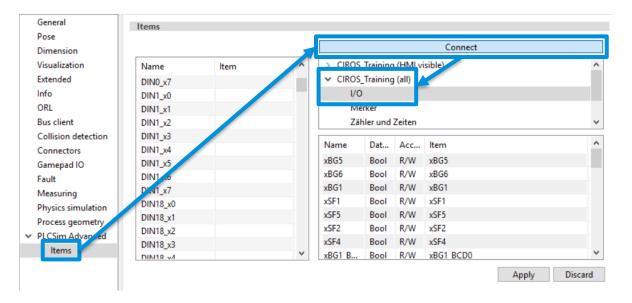
FESTO

### **Exercise**

#### **Configuring the interface**

- 1. Go to the subpage ltems
- 2. Click on Connect
- 3. Open either the entry that says all or HMI visible, depending on your preference. The latter only offers you I/O tags that have been declared as *Visible in HMI engineering* inside TIA
- 4. Under this entry, open I/O

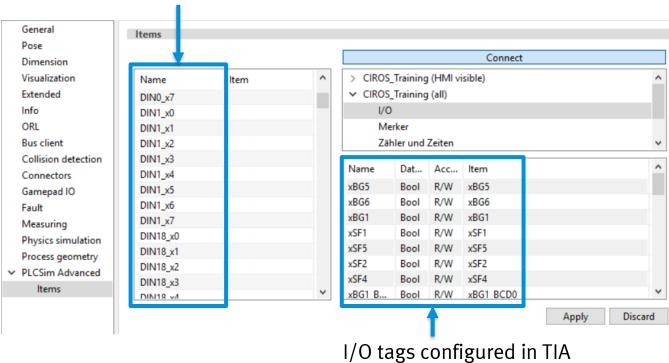
Note that you're also offered Memory, Counters and Timers and Data blocks. You can hook up CIROS I/Os to any of these but in this exercise, we'll only use I/Os.





### Exercise

#### **Configuring the interface**



#### Virtual I/Os of the CIROS PLC

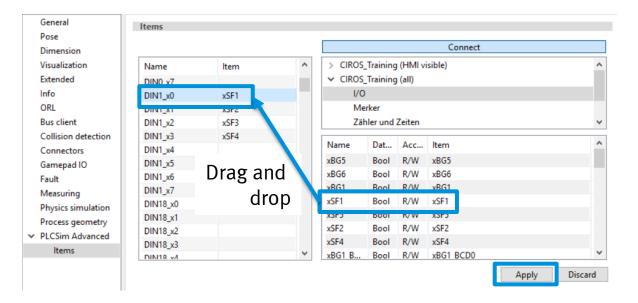
FESTO

### Exercise

#### **Configuring the interface**

- From the list of I/O tags on the right, drag and drop each I/O to the matching entry on the left.
   <u>Note:</u> Connect CIROS inputs with PLC inputs, and CIROS outputs with PLC outputs.
- 2. When done, click "Apply".
- 3. Optionally, click "Connect" again to disconnect.

**Hint:** You can select multiple I/Os from the right if they are in the right order and drag them to the left at once.



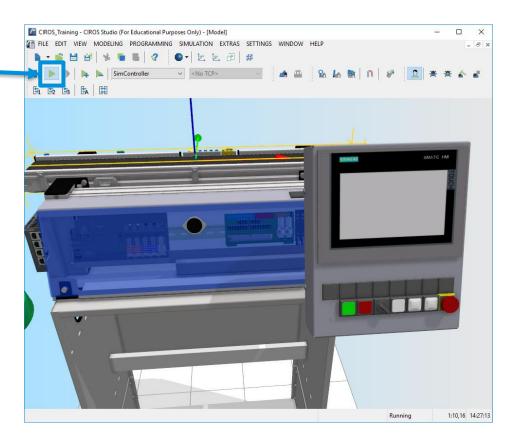


### Exercise

#### Run the simulation

- 1. Click the play button to start the simulation
- 2. Use TIA to go online / connect to your PLC instance

K Siemens - C:\Users\hmrm\Documents\Auton	nation\Projects\CIROS_Train_q\CIROS_Train	ning
Project Edit View Insert Online Options	Tools Window Help	
📑 📑 🖬 Save project 📑 🐰 🏥 🗎 🗙 🗠	) ± (🍽 ± 🗟 🗓 🔝 🔛 🔛 🙀 🎺 Go online	🖉 Go offline  🏭 🖪 🖪 🔛 🔭
Project tree 🔲 🖣	CIROS_Training ► PLC_1 [CPU 1512SP	F-1 PN] → PLC tags → myTags [36]
Devices		
🖬 🔲 🖬	🖆 😤 🔁 😤 🛍	
6	myTags	
🗧 🔻 🗋 CIROS_Training	Name	Data type Address 🔺 Retain Acces
🗧 📑 Add new device	1 📲 xSF1	Bool %11.0 🗌 🗹
통 🛗 Devices & networks	2 📲 xSF2	Bool 🔳 %11.1 💌 📃 🗹
PLC_1 [CPU 1512SP F-1 PN]	3 📲 xSF3	Bool %11.2
Device configuration	4 🚛 xSF4	Bool %11.3





### Exercise

#### Run the simulation

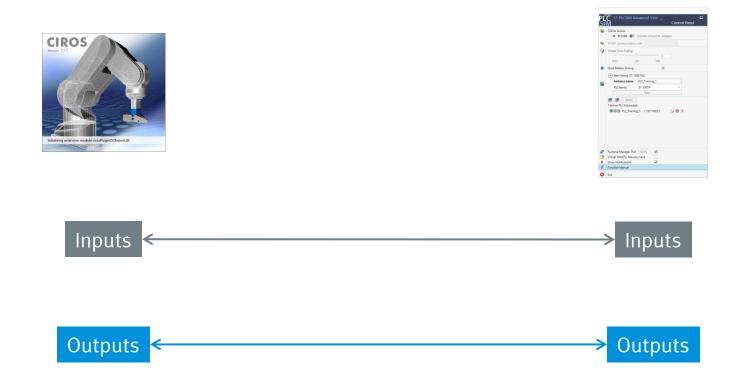
- 1. In TIA, open your tag list and monitor it
- 2. Test the connection by clicking on the virtual green start button inside CIROS. You should see the value of %11.0 change in TIA

CIROS, Training - CIROS Studio (For Educational Purposes Only) - [Model] – 🗆 🗙	K Siemens - C:\Users\hmrm\Documents\Auto	mation\Projects\CIROS_	Training\CIROS_Training						_ 🗆 X
The EDIT VIEW MODELING PROGRAMMING SIMULATION EXTRAS SETTINGS WINDOW HELP	Project Edit View Insert Online Options	Tools Window Hole							
▶ • 🖆 🖆   🛸 👅 🗸 🔍 ● • 🗠 🗁 🗇							Totally Int	tegrated Automa	ation
🔣 🕨 🕨 🛌 SimController 🗸 «NoTCP» 🗸 🏩 🏠 🛵 🕅 🖓 🚨 🛪 🐇 🕹	📑 🎦 🔚 Save project ا 🐰 🗎 🗎 🗙 🕨	) ± (ª ± 🛅 🔣 🔟	🖳 🔝 🔊 Go online 📝 Go o	offline 🚮? 📗 📑	* = 🗆			P	ORTAL
	Project tree 🔲 🖣	CIROS_Training > P	LC_1 [CPU 1512SP F-1 PN]	▶ PLC tags ▶ my	/Tags [36]				🖬 🖬 🗙 🔳
	Devices						🕣 Tags	🗉 User const	ants 🔋 🕏
		🔊 🔮 🖻 🖻 🚏	11				_	1	Tas
SMITC HM	5	yrags							ଜି
	🗧 🔻 🗋 CIROS_Training 🛛 💆 🔼	Name	Data type	Address .	Retain	Acces Writ	ta Visihl	Monitor value	
	🗧 📑 Add new device	1 🕣 xSF1	Bool	%11.0			✓	FALSE	지님
	B Devices & works	2 📲 xSF2	Bool	1 1011	- U			IRUE	▲ Libr
	🚊 👻 📴 💷 1 [CPU 1512SP F-1 PN] 🗹 🔵	3 🚛 XSE2	Bool	%11.2			✓	FALSE	≡ ari.
	P Device configuration	xSF4	Bool	%11.3				FALSE	es
	😵 Online & diagnostics	5 📲 xBG1	Bool	%11.4				FALSE	
	Be Software units	6 📲 xSF5	Bool	%11.5				TRUE	
	🔰 🕨 🕞 Program blocks	7 📲 xBG5	Bool	%11.6			Ā 🗖	FALSE	
	Technology objects	8 📲 xBG6	Bool	%11.7			Ā 🕅	FALSE	
	External source files	9 🕣 xBG1_BCD0	0 Bool	%142.0			Ā 🕅	FALSE	
	🔻 🔁 PLC tags 🛛 🔵	10 - xBG2_BCD		%142.1			Ā 🕅	FALSE	
	Show all tags	11 - xBG3_BCD2		%142.2			Ā 🕅	FALSE	
	Add new tag table	12 🚛 xBG4_BCD3		%142.3			Ā 🕅	FALSE	
	🛒 Default tag table [52]	13 🕣 xG1_BG7_k		%142.4			Ā 🕅	FALSE	
	🖳 myTags [36]	14 🕣 xG1_BG8_k		%142.5			Ā 🗖	FALSE	
	▶ 📺 PLC data types	4.5		2002 C					~
Running 1:10,16 1422713	Watch and force tables	<							>



## **More Information**

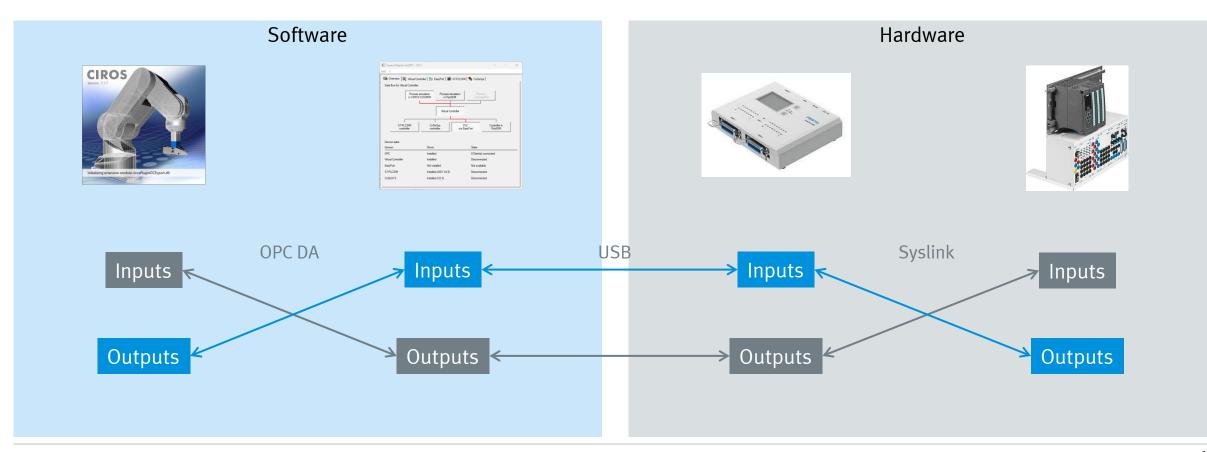
Mapping CIROS I/Os to PLCSIM Advanced instance I/Os





## **More Information**

#### Mapping CIROS I/Os to real PLC via EasyPort and EzOPC





### **Common Issues**

#### Can't download project to PLC instance anymore

Once CIROS has established a link to a PLCSIM instance, that instance is bound to the simulation. Only when the CIROS simulation is running, will the instance run as well.

Should TIA appear to be stuck when downloading to the instance that is likely because your CIROS simulation is paused. As soon as you start the simulation, the download will continue.



### **Remote Connection between CIROS and PLCSIM Advanced**

If you're running CIROS and PLCSIM Advanced on different machines, you need a little bit of extra configuration.

- 1. PLCSIM Advanced must use the Virtual Ethernet Adapter
- 2. The communication interface must be set to the network interface through which you're connecting to CIROS
- 3. The runtime manager port must be enabled. Note the port number written here

Online Access	
PLCSI	
TCP/IP comm	unication with <local> ~</local>
Virtual Time S	Scaling Off 100
Start Virt	ual S7-1500 PLC
1 Active PLC I	MRES instance(s):
Die Die Die Ch	ROS_Training / 192.168.0.1 🛛 🕀 🔘 💌
	ager Port 50000
	IC Memory Card



### **Remote Connection between CIROS and PLCSIM Advanced**

You also have to let CIROS know, where to find PLCSIM Advanced.

- 1. Go to the virtual PLC's properties
- 2. Open the PLCSIM Advanced page
- 3. Check Activate remote connection
- 4. Enter the IP address of the PC where PLCSIM Advanced is running
- 5. Enter the runtime manager port number that is configured in PLCSIM Advanced

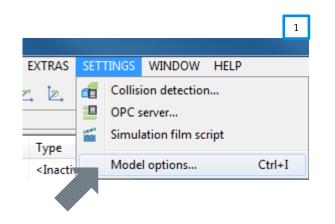
Properties		×
Properties Visualization Extended Info ORL Bus client Collision detec Connectors Gamepad IO Fault Measuring	tior	Communication ✓ Activate remote connection IP: 192.168.0.22 Port: 50000 Instance ✓ Connect to existing instance IP: 0.0.0.0 Name: CIROS_Training
Physics simula Process geometry > PLCSim Advar	etry	Create new instance Memory card:

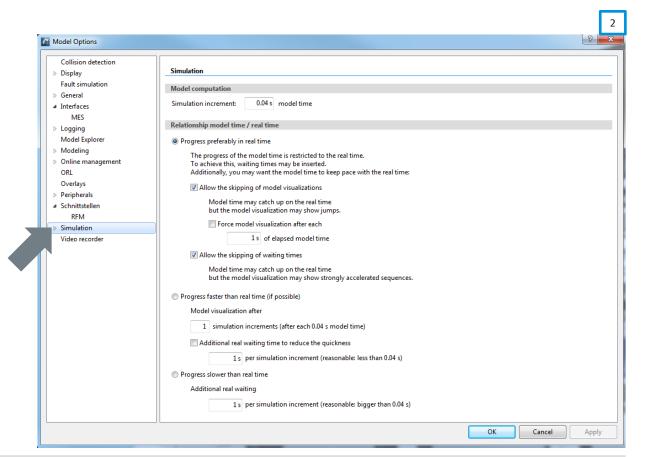


# Simulation



- Settings → Model options allows for configuring the way in which the simulation status will be updated during simulation.
- Mainly characterized by two parameters,
  - model computation
  - relationship model time / real time



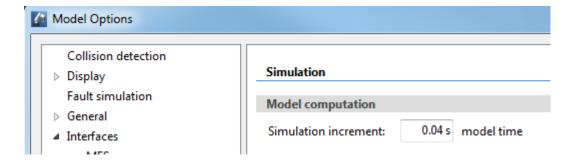




#### Model computation

- Simulation increment specifies the intervals in which the simulation status and its visualization will be updated.
- Default of 0.04s means 25 updates per second of model time.
- Provided that the computer CIROS is running on is powerful enough, this results in real-time behavior!
- Increasing the Simulation increment leads to fewer calculations of simulation states which might lead to some strange behavior.

**<u>Rule of thumb:</u>** Do not touch the default value for the simulation increment!





### Relationship model time / real time (1)

- Progress preferably in real time
  - Progress of model time is restricted to real time.
  - Default option, ensuring that a CIROS model behaves like a similar real CP Lab / Factory system (wrt. process times).
- Allow the skipping of model visualizations
  - By default, the visualization gets updated with every simulation increment.
  - Skipping some of these calculations might give CIROS the chance to keep track with real time.
- Allow the skipping of waiting times
  - To keep track with real time, one can also skip simulation updates when nothing has changed.

Model Options	<b>२</b> – ×
Collision detection Collision detection General General General General General General General General Codeling Conline management ORL Overlays Peripherals Schnittstellen RFM	Simulation         Model computation         Simulation increment:       0.04 s         model time         Relationship model time / real time         Progress preferably in real time         The progress of the model time is restricted to the real time.         To achieve this, waiting times may be inserted.         Additionally, you may want the model time to keep pace with the real time:         Image: Allow the skipping of model visualizations         Model time may catch up on the real time         but the model visualization may show jumps.
RFM  Simulation  Video recorder	<ul> <li>Force model visualization after each         <ul> <li>1s of elapsed model time</li> <li>Allow the skipping of waiting times</li> <li>Model time may catch up on the real time but the model visualization may show strongly accelerated sequences.</li> </ul> </li> <li>Progress faster than real time (if possible)         <ul> <li>Model visualization after</li> <li>1 simulation increments (after each 0.04 s model time)</li> <li>Additional real waiting time to reduce the quickness</li> </ul> </li> </ul>
	Additional real waiting time to reduce the quickness      Is per simulation increment (reasonable: less than 0.04 s)      Progress slower than real time      Additional real waiting      Is per simulation increment (reasonable: bigger than 0.04 s)      OK Cancel Apply



#### **Relationship model time / real time (2)**

- Progress faster than real time (if possible)
  - Allows CIROS to simulate as fast as the underlying hardware environment allows.
  - Skipping some visualizations results in even faster computations.
  - The speed can be set to adjust how fast the simulation should be in Model visualization after _____ simulation increments ...
    - For example, to simulate 5x faster than real time, configure it as follow: Model visualization after 5 simulation increments ...

Model Options	8 <mark>×</mark>
Collision detection Display Fault simulation General Interfaces MES Logging Model Explorer Modeling Online management	Simulation         Model computation         Simulation increment:       0.04 s         model time         Relationship model time / real time         @ Progress preferably in real time         The progress of the model time is restricted to the real time.         To achieve this, waiting times may be inserted.
ORL Overlays > Peripherals Schnittstellen RFM > Simulation Video recorder	Additionally, you may want the model time to keep pace with the real time:   Additionally, you may want the model visualizations  Model time may catch up on the real time but the model visualization may show jumps.  Force model visualization after each  1s of elapsed model time  Allow the skipping of waiting times  Model time may catch up on the real time but the model visualization may show strongly accelerated sequences.
	<ul> <li>Progress faster than real time (if possible)</li> <li>Model visualization after</li> <li>1 simulation increments (after each 0.04 s model time)</li> <li>Additional real waiting time to reduce the quickness</li> <li>1s per simulation increment (reasonable: less than 0.04 s)</li> <li>Progress slower than real time</li> </ul>
	Additional real waiting          1 s       per simulation increment (reasonable: bigger than 0.04 s)         OK       Cancel       Apply



### **Relationship model time / real time (3)**

- Progress slower than real time
  - Mainly for debugging purposes.

Model Options		? ×
Collision detection Display Fault simulation General Interfaces MES Logging Model Explorer Modeling Online management ORL Overlays Peripherals Schnittstellen RFM Simulation Video recorder	Simulation         Model computation         Simulation increment:       0.04 s         model time       model time         Relationship model time / real time         Progress preferably in real time         The progress of the model time is restricted to the real time. To achieve this, waiting times may be inserted. Additionally, you may want the model time to keep pace with the real time:         Additionally, you may want the model time to keep pace with the real time:         Additionally, you may want the model time to keep pace with the real time:         Additionally, you may want the model time to keep pace with the real time:         Image: State the model visualization may show jumps.         Image: State the model visualization after each         Image: State the model visualization may show strongly accelerated sequences.         Image: Progress faster than real time (if possible)         Model visualization after         Image: State than real time (if possible)         Model visualization after         Image: State than real waiting time to reduce the quickness         Image: State than real waiting time to reduce the quickness         Image: State than real waiting time to reduce the quickness         Image: State than real time (if personable: less than 0.04 s)         Image: State than real time to reduce the quickness         Image: State than real waiting time to reduce the quicknes	
	Additional real waiting           1 s         per simulation increment (reasonable: bigger than 0.04 s)	
	OK Cancel	Apply



#### Remarks

- Timestamps in CIROS are based on model time, while in MES4 timestamps are based on real time!
- Example of what could happen if model time differs from real time.
  - Assume that a single step of an MES4 workplan takes 10s of real time on a corresponding real application module.
  - If CIROS is keeping track with real time (model time = real time), the step will be simulated in 10s of real time, too.
  - If CIROS is running faster, this step still requires 10s of model time, but CIROS can simulate that 10s of model time in (to give an example) 2s of real time. MES4 will record that the operation required only 2s of real time!
- Therefore, if CIROS is simulating faster/slower than real time the process times the MES4 is measuring are no longer reliable!
- Due to the different ways of measuring time, it is not possible to simulate the annual production of a CP Lab / Factory within a few hours!



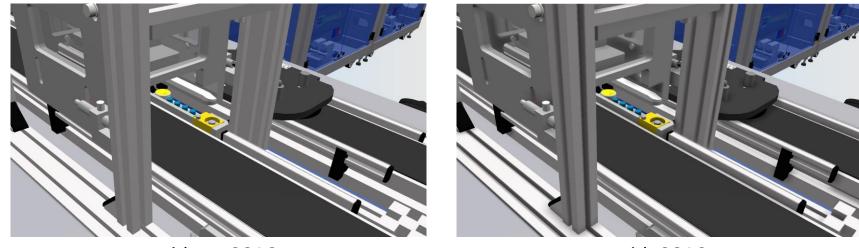
#### **Disabling shadow simulation**

- Calculating shadows during simulation allows for a more natural appearance of the model but requires a lot of CPU/GPU performance. Simulation might slow down significantly!
- For less powerful hardware environments disabling shadows improves overall simulation performance.
  - Screen Space Ambient Occlusion (SSAO)
  - Shadow light sources



### Screen Space Ambient Occlusion (SSAO) (1)

- SSAO is a computer graphics technique for efficiently approximating the ambient occlusion effect, caused by ambient lightning, in real time.
- While the implementation in principle is quite fast, it nevertheless requires substantial computation power.

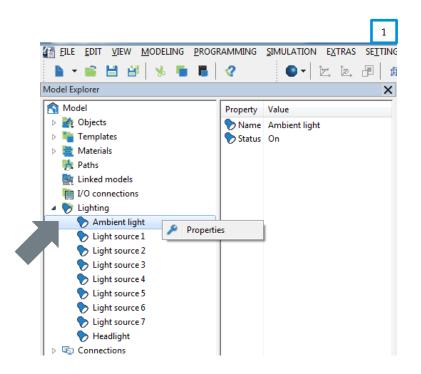


without SSAO



#### Screen Space Ambient Occlusion (SSAO) of a model (2)

• Use Model Explorer  $\rightarrow$  Model  $\rightarrow$  Lightning  $\rightarrow$  Ambient light  $\rightarrow$  Properties to disable or enable Screen Space Ambient Occlusion.

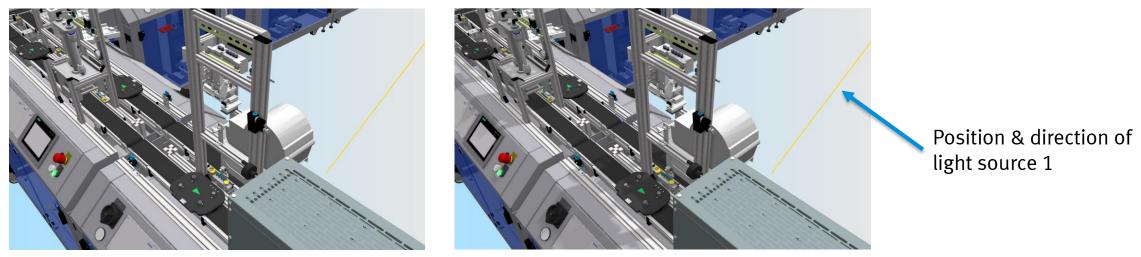


General		
Shadow (SSAO)	🔽 Enable	
	Intensity	
	Quality	
	Range	
		Restore defaults



#### Shadow light source 1

- Usually, light source 1 is causing shadows, while light sources 2 to 7 and headlight do not have that option
- If enabled the simulation of that shadow depends on the position of light source 1 and its properties



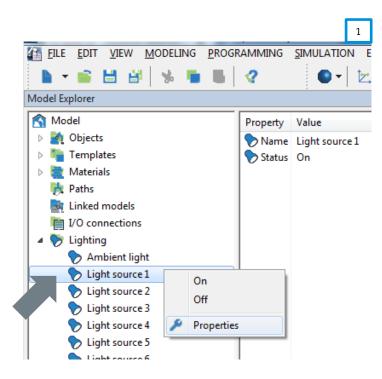
without shadows caused by light source 1

including shadows



#### Shadow light source 1 of a model

• Use Model Explorer  $\rightarrow$  Model  $\rightarrow$  Lightning  $\rightarrow$  Light source 1  $\rightarrow$  Properties to disable or enable shadow simulation.



		2	
Properties			
General			
Shadow	🔽 Enable		
	Intensity		
	Resolution	2048 🗸	
	Filter	Soft 🔹	



#### SSAO and shadow source of CIROS application.

- Disabling shadows like shown on the slides before is valid for a particular model only!
- To disable or enable shadows regardless of the model use

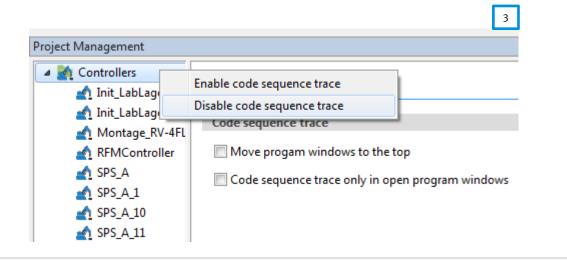
File  $\rightarrow$  Application options  $\rightarrow$  Display.

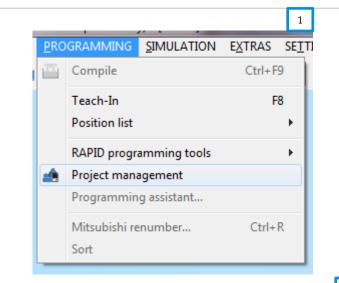
Application Options		?	×
<ul> <li>&gt; Data export</li> <li>&gt; Data import</li> <li>&gt; Display</li> <li>Editor</li> <li>&gt; Modeling</li> <li>ORL</li> <li>Paths</li> <li>Programming tools</li> <li>Save</li> <li>Teach-In</li> <li>VR devices</li> <li>Workspaces</li> </ul>	Display         Shadows         O Deactivate shadows         Activate shadows only for first directional light source         Activate shadows for all light sources         Activate shadows for all light sources and shadow casting of transparent hulls         General         Deactivate SSAO         Deactivate warning 'Texture not found'		

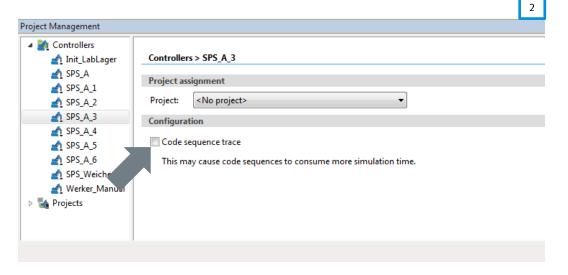
#### FESTO

### **Code Sequence Trace**

- Simulation is controlled by the chosen controller in the project and its code, whether PLC controller or robot program, can be traced during the simulation.
- 1. The function can be activated in Programming  $\rightarrow$  Project Management.
- 2. To enable or disable tracing of a particular controller, check the option Code sequence trace for the controller.
- 3. To enable or disable tracing of all controllers in the project, right click on Controllers and choose the option.





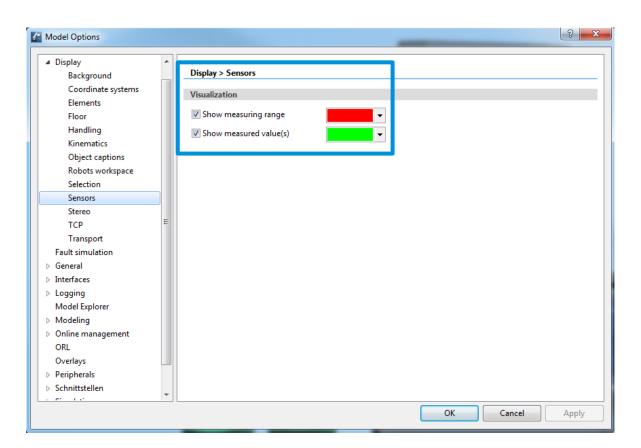




### **Visualising Sensor Data**

- Depending on the model options, measured range and values are not visualized by default in view window.
- Displaying sensors can be achieved by configuring Settings → Model options → Display → Sensors
- Measured value depends on the sensor type

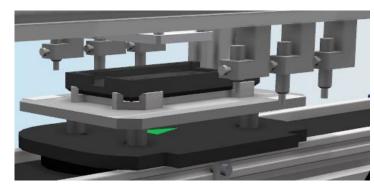
Light barrier	Obstacle detected
Distance sensor	Distance to obstacle
Colour sensor	Colour



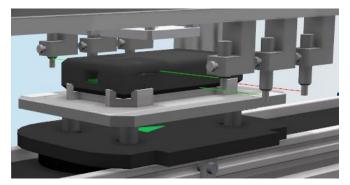


# **Visualising Sensor Data**

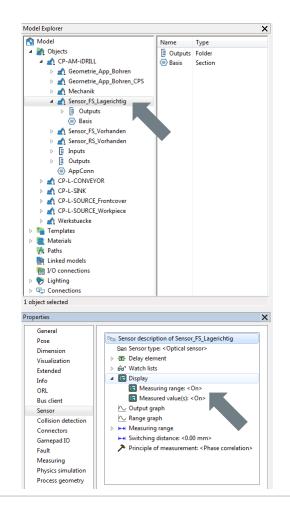
• In rare cases one must enable the visualization of sensor data by modifying the properties of each individual sensor.



Sensor visualisation off



Sensor visualisation on





? X

Display

Assigned profiles: Simulation performance

>

### **Data Logging**

- Data logging in CIROS allows for storing and visualization of values of inputs, outputs, variables, etc. during simulation.
- Example shown here: Visualization of the time needed for performing a status update.
- Configuration of data logging is available at Settings  $\rightarrow$  Model options  $\rightarrow$  Logging.

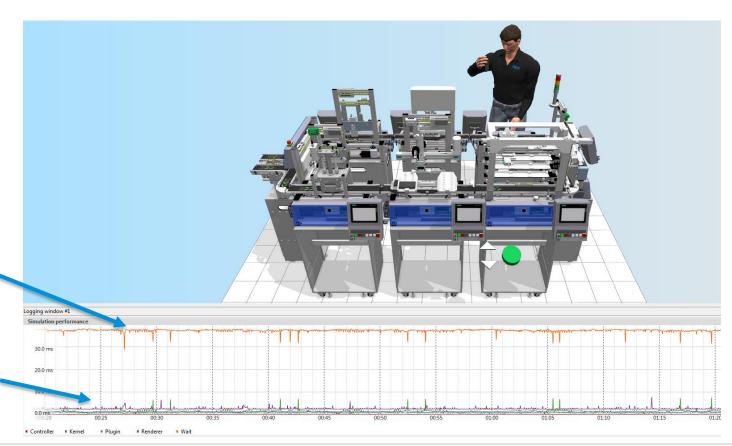
Collision detection Display Fault simulation General	_Logging General	Collision detection <ul> <li>Display</li> <li>Fault simulation</li> <li>General</li> </ul>	Logging General
Interfaces     MES     Interfaces     MS     Profiles     Modeling     Online management     ORL     Overlays     Peripherals     Schnittstellen     RFM     Simulation     Video recorder	Target: Logging window #1  Available profiles:  Real-time graph Simulation performance XB0:360 Controller 1 XB0:380 Controller 2 XB0:360 Controller 2 XB0:360 Controller 4 Show sub-cycle values: never	Johefräces     MES     Logging     Profiles     Model Explorer     Model Explorer     Modeling     Online management     ORL     Overlays     Periphereals     Schnittstellen     RFM     FM     Simulation     Video recorder	Target: Logging window ≠1 (1) Available profiles: Real-time graph XB0:360 Controller 1 XB0:360 Controller 2 XB0:360 Controller 3 XB0:360 Controller 4 Show sub-cycle values:
	Activation Object: • I/O: •		Activation Object: UO:



### **Data Logging**

#### Simulation speed restricted to real time

Almost all the 40ms the simulation is waiting to avoid being faster than real time.



Time required for updating the simulation status is almost zero in this setup.



### **Data Logging**

Simulation speed as fast as possible without manual working place

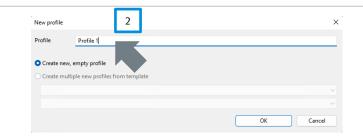


#### FESTO

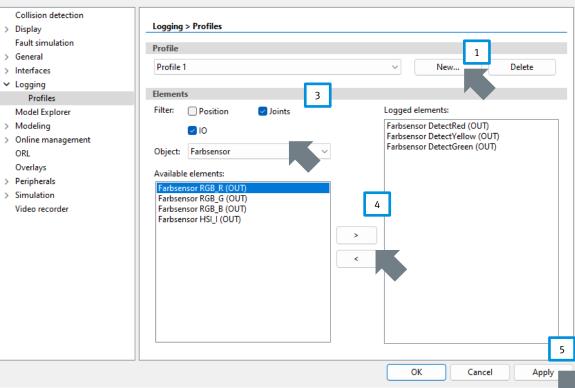
### **Data Logging**

#### Create new profile

- Apart from default data logging profiles, it is possible to log own data to simulate the model built.
- To log own data, a logging profile containing desired data must be created, it can be created in Settings → Model Options → Logging → Profiles.
- 1. In section Profile, select New...
- 2. Give a Name.
- 3. Use the filter option to filter the selection.
  - Possible data are robot positions, joint coordinates and IOs of the objects.
- 4. Use the arrow buttons to select and deselect data.
- 5. Finally, click Apply to create the profile.
- Profile created can then be added to logging window in Settings → Model Options → Logging, section General.



#### Model Options



? ×



# Simulation Control in CP System

### Industrial Robotic Language (IRL)

- Located in <project file>\CF\CPSystems.
- Scripts can be opened in CIROS in project management window.
  - Programming  $\rightarrow$  Project management
- Main simulation controller.
- Communicates with MES4 and Fleet Manager.
- Control the model movements.
- Read input values and write the output values.

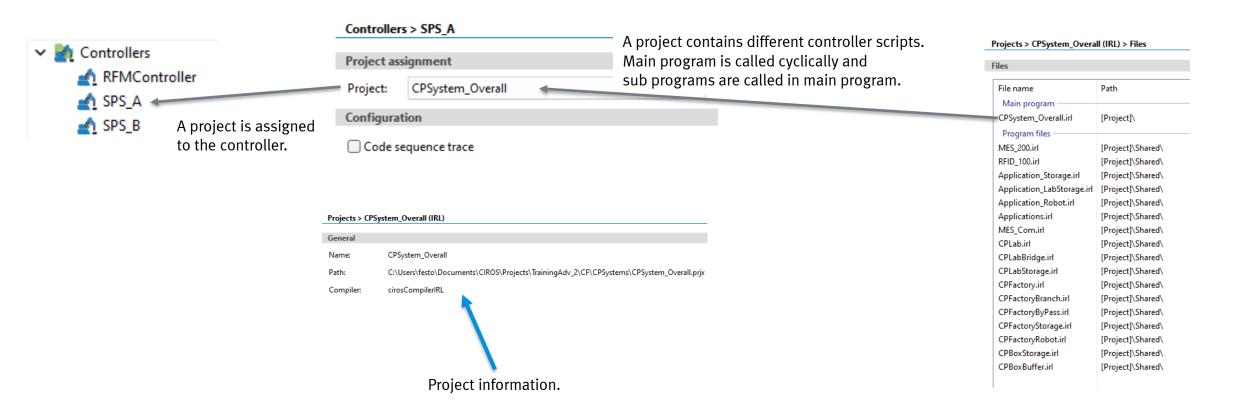
### Python

- Located in <project file>\CF\py.
- Translate MES4 part number to CIROS part number.
- Replicates workpieces and boxes.
- Send string requests to MES4

• <u>Note</u>: RFID data structure in CIROS is different from in real system.



### **CP System Simulation Controller**





# Python



# **Python in Model Libraries**

- Python 3.7 or higher is required for the support of CP Lab / Factory model library.
- Replication of CP Lab / Factory workpieces within CIROS is based on various Python scripts, compared to previous versions this change within the CIROS kernel simplifies the integration of user defined workpieces.
- If not already installed, CIROS installation wizard will install Python and add it to Windows path during the setup in silence mode.
- In case Python in removed...
  - Due to various reasons, like uninstallation of other applications in the same PC, Python can be uninstalled or removed from Windows path.
  - In this case, CIROS will throw an error, most commonly cirospluginpython.dll not loaded.
  - When this happens, user should check whether the correct Python version is installed and is it in the Windows path.



### **Python in Model Libraries**

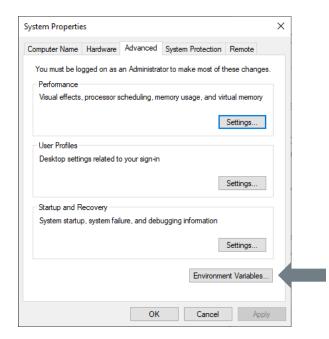
### Adding python to Windows path (1)

- To define Python in System Environment Variable Path in Windows 10:
  - 1. Open System Properties.

Menu  $\rightarrow$  Settings  $\rightarrow$  System  $\rightarrow$  About  $\rightarrow$  Advanced System Settings

ධ Home	Abou		
			This page has a few new settings
Find a setting	P Your P		Some settings from Control Panel
ystem	See detai		have moved here, and you can copy your PC info so it's easier to share.
Notifications & actions	Device		
Notifications & actions			Related settings
D Focus assist	OptiPle Device na		BitLocker settings
) Power & sleep	Processor	Hz 3.00 GHz	Device Manager
J Power & sleep	Installed I		Remote desktop
⊐ Storage	Device ID	29668	
	Product II		System protection
문 Tablet	System ty Pen and t	cessor iis display	Advanced system settings
H Multitasking	Copy	iis display	Rename this PC (advanced)
	cop)		
Projecting to this PC	Rename		Get help
Shared experiences			Give feedback
	Windo		
Clipboard	Edition		
C Demote Dealter	Version		
Remote Desktop	Installed		
D About	OS build		
	Experienc	212.3920.0	

- 2. Open Environment Variables.
  - Advanced  $\rightarrow$  Environment Variables





### **Python in Model Libraries**

#### Adding python to Windows path (2)

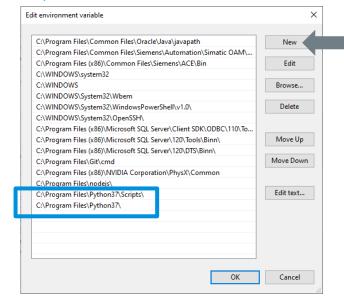
3. Edit Variable Path in System Variables.

#### System Variables $\rightarrow$ Path $\rightarrow$ Edit

Variable	Value
OneDrive	C:\Users\festo\OneDrive
Path	C:\Users\festo\AppData\Local\Programs\Python\Python38\Scripts
TEMP	C:\Users\festo\AppData\Local\Temp
TMP	C:\Users\festo\AppData\Local\Temp
at an an in blac	New Edit Dele
/stem variables	
Variable	Value
Variable NUMBER_OF_PROCESSORS	Value 6
Variable NUMBER_OF_PROCESSORS OS	Value 6 Windows_NT
Variable NUMBER_OF_PROCESSORS OS Path	Value 6 Windows_NT C:\Program Files\Common Files\Oracle\Java\javapath;C:\Program
Variable NUMBER_OF_PROCESSORS OS Path PATHEXT	Value 6 Windows_NT <u>C:\Program Files\Common Files\Oracle\Java\javapath;C:\Program</u> .COM;.EXE;.BAT;.CMD;.VBS;.VBE;JS;JSE;.WSF;.WSF;.WSC;.PY;.PYW
Variable NUMBER_OF_PROCESSORS OS Path PATHEXT PROCESSOR_ARCHITECTURE	Value 6 Windows_NT C< <u>Program Files\Common Files\Oracle\Java\javapath;C\Program</u> .COM;.EXE;.BAT;.CMD;.VBS;.VBE;JS;JSE;.WSF;.WSF;.WSH;.IMSC;.PY;.PYW AMD64
Variable NUMBER_OF_PROCESSORS OS Path PATHEXT	Value 6 Windows_NT <u>C:\Program Files\Common Files\Oracle\Java\javapath;C\Program</u> .COM;.EXE;.BAT;.CMD;.VBS;.VBE;JS;JSE;.WSF;.WSF;.WSC;.PY;.PYW
Variable NUMBER_OF_PROCESSORS OS Path PATHEXT PROCESSOR_ARCHITECTURE PROCESSOR_IDENTIFIER	Value 6 Windows_NT C:\Program Files\Common Files\Oracle\Java\javapath;C:\Program .COM;EXE;:BAT;.CMD;.VBS;.VBE;.JS;JSE;.WSF;.WSH;.MSC;.PY;.PYW AMD64 Intel64 Family 6 Model 158 Stepping 10, GenuineIntel

4. Insert the path to Python and Python Scripts folder here.

For example: C:\Program Files \Python 38\ and C:\Program Files \Python 38\Scripts\

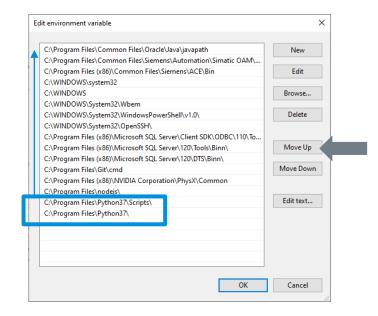


5. Restart the computer.



## Python Installed but Not Working

- When different python versions are installed in the same computer, CIROS always selects the version on top of the Windows environment path list.
- When the version on top of the list is not supported by CIROS, there will be an error.
- Typical error message is "cirosPluginPython.dll not loaded".
- Steps to solve this problem
  - 1. Open Windows Environment Variable.
  - 2. Edit System variables.
  - 3. Move the python with correct version up to the top.





## **Python Scripts**

- CIROS v7 and above works with Python scripts. The scripts can be called for following purposes
  - Creation and modification of model.
  - Controller for simulation or components.
  - Define user defined commands in context menu.
- There is an integrated python module for CIROS, "Ciros"
  - Overview of the functions in the module can be called from Menu  $\rightarrow$  Extras  $\rightarrow$  Python  $\rightarrow$  Show function list
  - In CIROS 7.1, there are currently 19 classes and 275 commands
  - Example models can be found in C:\Program Files\Festo Didactic\CIROS 7.0\CIROS Studio\Example Models\Help\Python
- Requirements
  - Python 3 is installed
    - Supported versions are 3.6, 3.7, 3.8 and 3.9
    - The used version will be chosen automatically
    - First line in python function list states the version used.

#### Note:

- Python must be defined in windows PATH-variable.
- The script cannot contain endless loop.



### **Python Scripts**

#### Python scripts can be executed by following methods

- 1. Manually
  - Application script (Concurrent or blocking)
  - With or without simulation
  - Menu Extras  $\rightarrow$  Python  $\rightarrow$  Execute
- 2. By a trigger during simulation
  - Simulation script (Blocking)
  - Parameters
    - Element: Specify the element that calls the script.
    - Trigger: Specify event that trigger the call.
    - File: The script to be called.
    - Function call: Function in the script to be called. Optional.
    - Output: Defines whether outputs should be written in message window or not.

Python script

Element:

Trigger

Execute:

Output:

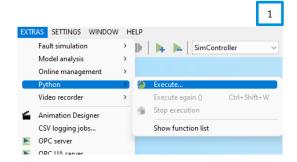
File:

Object "Assembly"

simulation start

Allowed

- 3. Python controller
  - Controller script (Concurrent)
  - A controller object, e.g. simulation controller is required.



2

× ...

Cancel

Collision detection Connectors

Info

ORL Bus client

Fault Measuring Physics simulation

CP System Process geometry

Python

ОК



### **Python Scripts**

#### **Execution type**

- 1. Concurrent
  - The python script is executed while CIROS is running.
  - This execution type is required for scripts containing GUI elements.
- 2. Blocking
  - First, CIROS is interrupted. Then, the python script is executed up to its end. Finally, CIROS is resumed.
  - This execution type is required for scripts containing special commands e.g. 'muteGUI'.

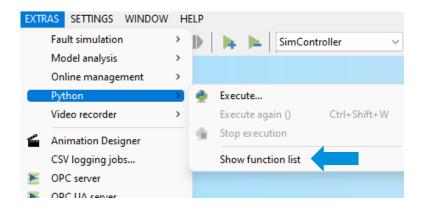
#### Call type

- 1. Simulation script
  - Invoked by a trigger during simulation.
  - The execution type is 'blocking'.
- 2. Controller script
  - Invoked upon simulation start. Acts as a controller for a specific object.
  - The execution type is 'concurrent'.
- 3. Application script
  - Invoked by a menu command or a command line argument. The execution type can be chosen by the user.
  - Please use *#CIROS pragma: blocking* or *#CIROS pragma: concurrent* in the first line of your script.



#### **Built-In Function List**

- 1. Show the function list by clicking Menu  $\rightarrow$  Extras  $\rightarrow$  Python  $\rightarrow$  Show function list.
- 2. Copy the function list and paste it in an editor, for example 'Notepad++'.



#### Duthon 3 9 1

#### Execution Type of a Script

Concurrent: The python script is executed while CIROS is running. This execution type is required for s Blocking: First, CIROS is interrupted. Then, the python script is executed up to its end. Finally, CIRC

#### Call type of a Script

Simulation script: Invoked by a trigger during simulation. The execution type is 'blocking'. Controller script: Invoked upon simulation start. Acts as a controller for a specific object. The execu Application script: Invoked by a menu command or a command line argument. The execution type can be chr #CIROS pragma: blocking

#CIROS pragma: concurrent

in the first line of your script.

#### Optional Parameters

Some function parameters are optional. If an optional parameter is omitted, the first listed alternativ

#### Element ID

or

Syntax: <element type>:<namel>|<name2>|... Element types: ENVIRONMENT|OBJECT|SECTION|HULL|LINKEDMODEL|MATERIAL|GPP|GP|PATH|PATHNODE|INFUT|OUTPUT|E Examples: OBJECT:HyObject, SECTION:HyObject[MySection, OUTFUT:HyObject|Output], GPP:HyObject[MyGrippet]

#### Robot Configurations

A robot configuration is represented by a configuration index. For a 6-axis robot with standard Kinemat

#### CirosUtil.py

A collection of additional CIROS-specific functions can be found in the module CirosUtil.py. After inst

Module 'Ciros2

#### Class: Application

This object can only be instantiated in application scripts.



### **CP System Construction Helper**

• A python library contains of four classes.

Class	Explanation
CPFactoryConstructionHelper	Constructs a single line of CP-Factory base modules from left to right.
CPLabConstructionHelper	Constructs a single island of CP-Lab modules from left to right, base modules and corners must be added in the correct order, build order is from left to right.
CPRobotinoConstructionHelper	Construct Robotino and related modules
CPSystemConfigurationHelper	This can be used to change the configuration of certain modules. Please note that u need to give the correct object name as it appears in the model, for example CP-L-BRANCH_1 for the second added branch module.

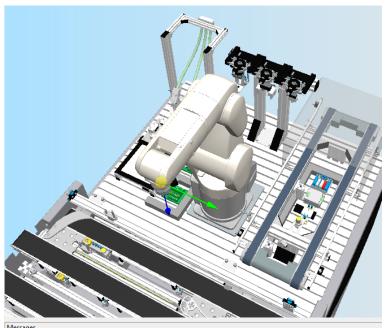
• Note: This library does not come in default with the installation. It might be found in the CIROS project configured by Festo Didactic in folder python. If it is not found, please contact us to request for it.



### **Use Case: Common TCP/IP Communication**

#### Tutorial: Send robot TCP to a TCP/IP server.

- At the end of this tutorial, user is able to get a robot TCP in joint coordinate and send it to an external TCP/IP server using common TCP/IP communication protocol with a python script.
- The example will be demonstrated with a CP-F-RASS_Mitsubishi modell.



WE3395 [335.0000898367876, -2.974001961338694e-08, 624.9998903862756] sent! Received b'Coordinate received'



### **Use Case: Common TCP/IP Communication**

- 1. Insert CP-F-RASS_Mitsubishi model in CIROS.
- 2. Create a python script named TcplpClient.py.
- 3. Program the script as shown in next page.
- 4. Start a TCP/IP server.
  - Example: With open source software 'Hercules'.
- 5. In CIROS, select Extra \ Python \ Execute.
- 6. Select the python script.
- 7. Cartesian coordinate is received in the server.
- 8. Send a reply to close the connection.



### **Use Case: Common TCP/IP Communication**

#### **Python script**

```
#CIROS pragma: concurrent
import Ciros2 as ciros
import CirosUtil
import socket
def TcplpClient(msg):
   HOST = "127.0.0.1"
   PORT = 9000
   with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
        s.connect((HOST, PORT))
        s.sendall(msg.encode('utf-8'))
        data = s.recv(1024)
   print(f"Received {data!r}")
```

#### def robotTCP():

rass = ciros.Object('Montage_RV-4FL') #Find the object
flange = rass.getGripperpoint('Flange') #Find the gripper
coord = flange.getFrame(origin='OBJECT').getTranslation() #Get translation
coordinate of gripper / TCP of robot
return str(coord)

TcpIpClient(robotTCP())

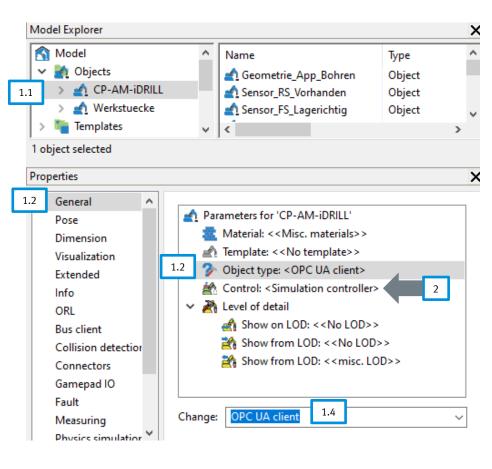


# **OPC UA Interface**

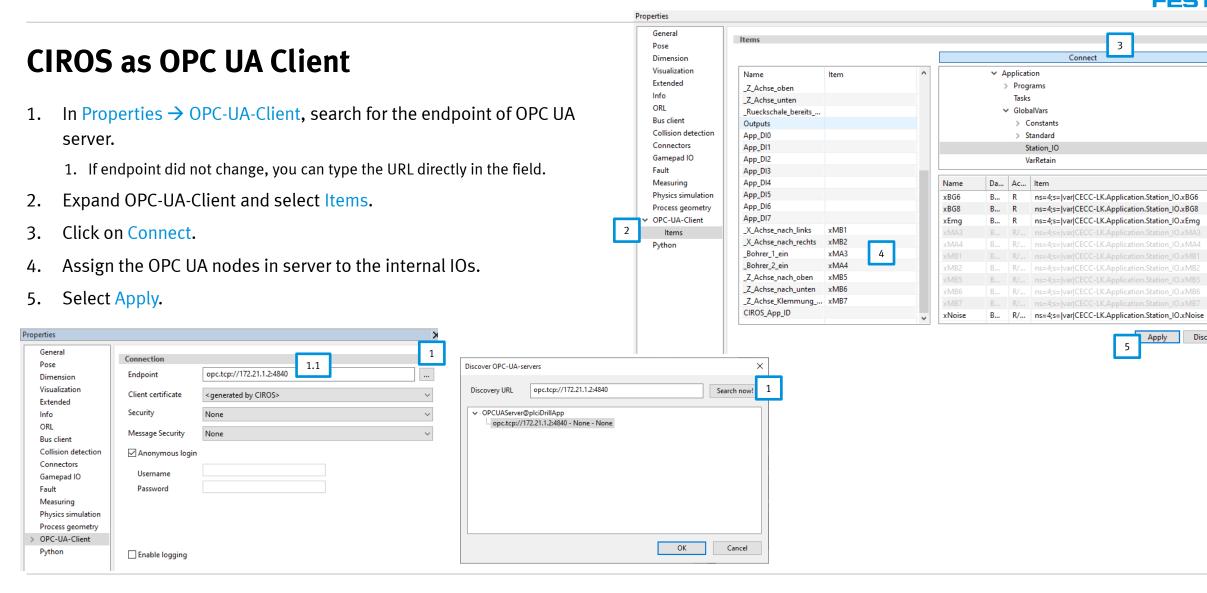
#### FESTO

## **CIROS** as OPC UA Client

- 1. Change controller object type to OPC UA client.
  - 1. Select the object in Model Explorer.
  - 2. In Properties window, select General  $\rightarrow$  Object type.
  - 3. First, change object type to inactive object to clear filter.
  - 4. Then, all the available object types will be shown, select OPC UA client.
- 2. Change controller control to Simulation controller.







#### FESTO

3

Discard

Apply

5



# **Robot Programming**

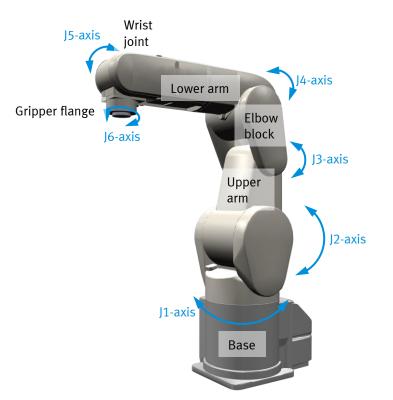
Mitsubishi



## Mitsubishi Industrial Robot

#### Electric RV-4FL robotic arm

Туре	Articulated robot
Number of axes	6
Ultimate load	4 kg
Maximum reach radius	649 mm
Movement range	480° / 240° / 164° / 400° / 240° / 720°
Maximum composite speed	9048 mm/s
Cycle time	0.36 s
Position repeatability	± 0.02 mm
Weight	41 kg
Tool wiring	8 1/0
Protection rating	IP67





## Mitsubishi Industrial Robot

#### Robot controller CR750-D

Programming language	MELFA-BASIC-V
Number of programs	512
Positions / program	3900
Programming	Teach box / PC
Power supply	Single-phase 180 – 253 V AC
Interface	RS422 / ethernet / USB / digital I/O
Dimensions (H x W x D)	430 mm x 425 mm x 174 mm
Weight	16 kg
Protection rating	Ground position / IP20





## Mitsubishi Industrial Robot

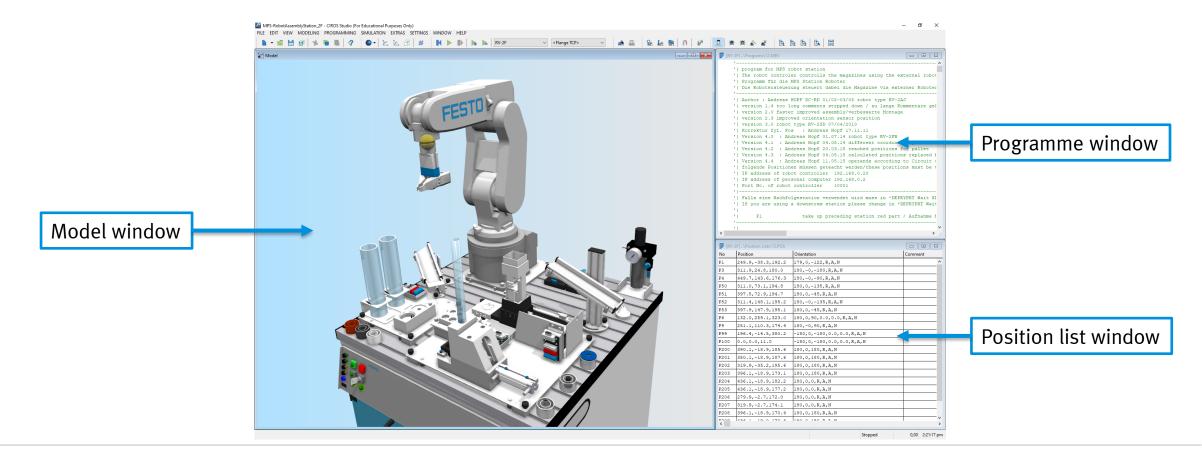
#### Teach box R56TB

Menu navigation (language)	German, English, French, Italian
Features	Operating, programming and monitoring all robot features
Programming and Monitoring	Reading out information even during the running system; Programming using a virtual keyboard; Display of up to 14 lines of programming code; I/O Monitoring of up to 256 inputs and 256 outputs; Maintenance display of service intervals; Trouble indication of the last 128 alarms
Display	Touchscreen with background lighting 6,5 " TFT display (640 x 480 pixel), 65536 colours
Interface	USB, combined RS422 and ethernet interface
Connection	Direct connection to the robot controller, cable length 7m
Protection rating	IP65
Weight	1,25 kg





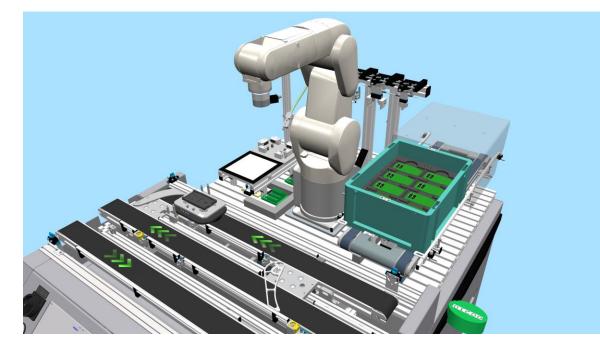
### **Layout and Windows**





#### **Robot simulation**

- RASS stands for Robot ASsembly Station.
- It is possible to simulate the Mitsubishi robot program in CIROS environment.
- The robot program requires several input parameters, which usually comes from MES4 or PLC. For standalone robot simulation in CIROS, user has to provide the input parameters manually.
- Following are the input parameters are required for standalone CIROS robot simulation:
  - 1. Program number.
    - Program 1 : Assemble PCB
    - Program 2 : Assemble PCB and front fuse
    - Program 3 : Assemble PCB and rear fuse
    - Program 4 : Assemble PCB and both fuses
  - 2. Position in box.





#### **General information**

- <u>Attention</u>: General knowledge of Mitsubishi robot and robot programming are required to proceed in this chapter. Please be informed about Mitsubishi robot program structure, programming language, and I/O connections before proceeding.
- The robot programs are written in Melfa Basic V (MBA5) with extension .mb5, which is the programming language of Mitsubishi robot.
- Positions are stored in the position list with extension .pos.
- There is a robot project in CP-F-RASS model, which have the same programs as in actual robot. However, the I/O channels and tool changing mechanism are modified to suit the modelling environment.
- In CIROS simulation, a main program RobotSystemDriver.mb5 is used to simulate the slot allocation in actual robot controller.
- Generally, a program name can be any string characters. However, PLC only calls the program by integers. To allow the new program to be able to function in real robot, the program name should be numbers.

#### FESTO

## **CP-F-RASS**

#### Programs list

		MONITORPALWS.mb5
RobotSystemDriver.mb5	Main program. Assign subprograms to slot.	MOUNTBOTFUSE.mb5
999.mb5	Reset robot	MOUNTPCB.mb5
1.mb5	Assemble PCB to front cover	MOUNTTOPFUSE.mb5
2.mb5	Assemble PCB and front fuse to front cover	MOVHOME.mb5
3.mb5	Assemble PCB and rear fuse to front cover	PCBTRAYCNTRL.mb5
4.mb5	Assemble PCB and both fuses to front cover	PICKFRMSTOPR.mb5
5.mb5	Demo program. Assemble and disassemble PCB in front cover.	PICKFRMVISION.mb5
123.mb5	Check all positions in PCB box. Assemble and disassemble all PCBs in box.	PICKFUSFRMAG.mb5
234.mb5	Camera test program. Pick front cover from stopper, place to vision field and run camera. Repeat four times.	PICKNEWTOOL.mb5
255.mb5	Calculate positions in box based on four positions taught	PICKPCBFRPAL.mb5
ENERGSAVEVACU.mb5	Switch on vacuum gripper when workpiece is loose	PICKWPFROMASS.mb5
GETCAMRESULT.mb5	Get camera result	PLACETOSTOPR.mb5
GETCURTOOLNO.mb5	Get current tool number	PLACETOVISION.mb5
GETFUSEMAGNO.mb5	Get fuse magazine number	SENSORCHECK.mb5
GRPCLOSE.mb5	Close gripper	SENSORCHECK1.mb5
GRPLOCK.mb5	Lock the gripper to robot arm flange	SENSORCHECK6.mb5
GRPOPEN.mb5	Open gripper	UBP.mb5
GRPRELEASE.mb5	Release gripper from robot arm flange	

GRPVACOFF.mb5	Switch off vacuum gripper
GRPVACON.mb5	Switch on vacuum gripper
INITIALIZE.mb5	Initialize input and output variables
MONITORHOME.mb5	Check if robot arm is at home position
MONITORPALWS.mb5	Check if robot arm is at PCB box position
MOUNTBOTFUSE.mb5	Mount rear fuse
MOUNTPCB.mb5	Mount PCB to front cover
MOUNTTOPFUSE.mb5	Mount front fuse
MOVHOME.mb5	Move robot arm to home position
PCBTRAYCNTRL.mb5	Check PCB box lock signal and PCB position count
PICKFRMSTOPR.mb5	Pick workpiece from stopper
PICKFRMVISION.mb5	Pick workpiece from vision field
PICKFUSFRMAG.mb5	Pick a fuse from fuse magazine
PICKNEWTOOL.mb5	Pick a new gripper
PICKPCBFRPAL.mb5	Pick PCB from PCB box
PICKWPFROMASS.mb5	Pick workpiece from assembly position
PLACETOSTOPR.mb5	Place workpiece to stopper.
PLACETOVISION.mb5	Pick workpiece from stopper to vision field
SENSORCHECK.mb5	Check sensors at stopper, assembly position, fuse magazines and input parameter for PCB position in box.
SENSORCHECK1.mb5	Check sensors at stopper, assembly position and input parameter for PCB position in box.
SENSORCHECK6.mb5	Check sensors at stopper position
UBP.mb5	User base program. Global program contains all global variable, flags and positions.



#### **Controllers and I/Os**

- Station CP-F-RASS has two controllers,
  - PLC controller
  - Robot controller
- In CIROS, the station has two controllers as well.
  - SPS_Roboter is the virtual representation of PLC controller
  - Montage_RV-4FL is the virtual representation of robot controller
- The I/Os are linked internally in CIROS model.

Montage_RV-4FL	Index	Туре	SPS_Roboter	Index	Туре	Description
I_Stop	100	DI	DOUT_100_x0	40	DO	Stop robot program
I_Start	101	DI	DOUT_100_x1	41	DO	Start a robot program
IDATA_(0-15)	116 – 131	16 Bit DI	AOUT_W102	002	AO	Program number in binary
DI_PCBPalletNo_(0-7)	172 – 179	8 Bit DI	AOUT_B109	004	AO	Position in PCB box
HOpen_1	900	DO				Open gripper
HClose_1	901	DO				Close gripper
HOpen_3	904	DO				Release gripper
HClose_3	905	DO				Lock gripper



#### **Grippers' tool number**

- Each gripper has a tool number. It is important to select the right tool as different TCP is used for different tools.
  - Tool 1 : Vacuum gripper.
    - Z-offset relative to flange TCP 205 mm.
    - C-rotation relative to flange TCP 33.5°.
  - Tool 2 : Parallel gripper for front cover.
    - Z-offset relative to flange TCP 170 mm.
    - C-rotation relative to flange TCP 33.5°.
  - Tool 3 : Parallel gripper for fuse.
    - Z-offset relative to flange TCP 151.5 mm.
    - C-rotation relative to flange TCP 33.5°.
  - Tool 4 : No gripper.
    - Z-offset relative to flange TCP 0 mm.
    - C-rotation relative to flange TCP 0°.



#### List of default positions in CIROS environment for reference

9 10

1 8

No	Definition	Position	
1	P_AssemblePCB	(-140.03,-367.01,109.01,-179.73,0.09,89.95)(7,0)	
2	P_PCBPalletOrigin	(93.00,276.00,213.00,-180.00,0.00,90.00)(7,0)	
3	P_PCBPaletXDir	(-140.10,276.00,213.00,-180.00,0.00,90.00)(7,0)	
4	P_PCBPaletYDir	(93.00,398.00,213.00,-180.00,0.00,90.00)(7,0)	
5	P_PCBPaletXYDir	(-140.10,398.00,213.00,-180.00,0.00,90.00)(7,0)	
6	P_CarrierStop1	(402.50,-267.00,171.50,179.73,-0.09,90.05)(7,0)	
7	P_Vision	(30.00,-350.00,101.50,180.00,-0.00,90.00)(7,0)	
8	P_AssembleWp	(-139.00,-362.00,114.00,-180.00,-0.00,90.00)(7,0)	<b>9</b>
9	P_AssembleFuse1	(-138.97,-401.18,127.13,-180.00,0.00,180.00)(7,0)	
10	P_AssembleFuse2	(-138.97,-388.18,127.13,180.00,-0.00,180.00)(7,0)	
11	P_FuseMagazine1	(-300.65,-430.30,160.30,177.86,-42.56,-177.96)(7,0)	
12	P_FuseMagazine2	(-299.24,-360.21,159.46,177.87,-42.57,-177.96)(7,0)	
13	P_FuseMagazine3	(-299.44,-290.40,159.99,177.87,-42.57,-177.96)(7,0)	
14	P_GrpStorageVac	(-395.65,-125.00,484.90,-180.00,-0.00,-0.00)(7,1)	
15	P_GrpStorageWp	(-395.65,0.00,484.90,-180.00,0.00,0.00)(7,1)	
16	P_GrpStorageFuse	(-395.65,125.00,484.90,-180.00,-0.00,-0.00)(7,0)	





### Steps to configure CP-F-RASS for simulation

- 1. Create a new CIROS project.
- 2. Insert and snap following from CP System model libraries.
  - 1. CP-F-RASS (Mitsubishi)
  - 2. CP-F-SOURCE
  - 3. CP-F-SINK
- 3. Optional: Hide safety glass.
  - 1. In Model Explorer, choose Objects → CP-F-RASS_Mitsubishi → Geometrie → Geometrie_Umhausung
  - 2. In Properties, select Visualization.
  - 3. Click Invisible.
- 4. Open robot program in project management.
  - 1. In Project Management, right click on Projects and select Open.
  - 2. Select <project folder>\CF\Rob_Montage\RV-4FL\Montage_RV_4FL.prjx.
  - 3. In Project Management, select Controllers → Montage_RV_4FL
  - 4. Assign project Montage_RV_4FL to the controller.

- 5. Assign required I/Os to IO monitor.
  - 1. Open an I/O monitor window, it can be any I/O monitor.
  - 2. Drag the Outputs from Model Explorer to I/O monitor window.
  - 3. Required Outputs are in Objects → CP-F-RASS_Mitsubishi → SPS_Roboter → Outputs
    - 1. AOUT_W102 (analogue 002): robot program
    - 2. AOUT_B109 (analogue 004) : position in PCB box
- 6. Change source part number to 210.
  - 1. In Model Explorer, select CP-F-SOURCE.
  - 2. In Properties  $\rightarrow$  CP System, change Part Number to 210.

**Note:** The configured model can be saved as a template which can be opened with CIROS Studio and Education in robot programming lessons.

Video tutorial: 51_ConfigureRASSForRobotProgramSimulation.mp4



## **Steps to simulate CP-F-RASS**

- 1. Optional: To reduce computing power, close all windows except model window and I/O monitor, e.g. Model Explorer, Properties, Project Management, etc.
- 2. Start simulation.
- 3. Give a robot program and position in box in I/O monitor and activate override.
  - 1. For example: robot program 1 and box position 3.
    - AOUT_W102 = robot program  $\rightarrow$  Value = 1
    - AOUT_B109 = position in box  $\rightarrow$  Value = 3
- 5. Press on green source button in model window.
- 6. Observe the program.
- To run another robot program, repeat step 3 to 6.
- To restart simulation, repeat step 2 to 6.

Video tutorial: 52_SimulateRASSRobotProgram.mp4



### Simulate Real Robot Program in CP-F-RASS Model

- It is possible to simulate real CP-F-RASS robot program in CIROS.
- However, the program has to be modified to suit the simulation environment, for example, the I/Os address and tool changing mechanism.
- Besides, the simulated program does not connect to a camera. Thus, the subprogram which connects with camera has to be commented out.
- The modified robot program can be saved as a template project and be used repeatably in robot programming classes as it is portable with both CIROS Studio and CIROS Education.

**<u>Ready to use project:</u>** CP-F-RASS_RobotProgramming_v717



### Simulate Real Robot Program in CP-F-RASS Model

#### Steps to modify robot program (1)

- 1. Create a new CIROS project.
- 2. Load and snap following models in place from CP System model libraries.
  - 1. CP-F-RASS (Mitsubishi)
  - 2. CP-F-SOURCE
  - 3. CP-F-SINK
- 3. Change Part Number of CP-F-SOURCE to 210.
- 4. Place all the real robot programs in a single folder and place the folder in CIROS project folder.
- 5. In the folder, delete following files.
  - 1. Files with type .bak.
  - 2. Files with type .prjx.

- 6. Copy following files from <project folder>\CF\Rob_Montage\RV-4FL to the robot program folder.
  - 1. RobotSystemDriver.mb5
  - 2. Montage_RV_4FL.prjx
  - 3. ENRGSAVEVACU.mb5, if not exist
  - 4. MonitorHome.mb5, if not exist
  - 5. MonitorPalWS.mb5, if not exist
  - 6. PCBTrayCntrl.mb5, if not exist
- In CIROS, open Project Manager, open the copied project Montage_RV_4FL.prjx in robot program folder and assign it to controller Montage_RV-4FL.
- 8. In Project Manager, open Projects  $\rightarrow$  Montage_RV_4FL (MBA5)  $\rightarrow$  Files and delete the files which do not exist.



### Simulate Real Robot Program in CP-F-RASS Model

#### Steps to modify robot program (2)

9. In CIROS, change the following in all files in the project.

Bits and bytes in I/O definitions							
From	То		From	То		From	То
2032	132		2072	172		2148	248
2033	133		2144	244		2150	250
2040	140		2147	247		2151	251
2064	164						

2. Tool changing mechanism

1.

From	То
<pre>M_Tool = m_GripperFuse</pre>	Tool P_tGripperFuse
<pre>M_Tool = m_GripperNone</pre>	Tool P_tGripperNone
<pre>M_Tool = m_GripperVac</pre>	Tool P_tGripperVac
<pre>M_Tool = m_GripperWP</pre>	Tool P_tGripperWP
HOpen 6	HOpen 3
HClose 6	HClose 3

- 3. Make following changes.
  - 1. Add following lines in 999.mb5. P_tGripperVac=(0,0,205,0,0,33.50) P_tGripperWP=(0,0,170,0,0,33.50) P_tGripperFuse=(0,0,151.5,0,0,33.50) P_tGripperNone=(0,0,0,0,0,33.50)
  - Comment out or delete all position declarations in UBP.mb5 and add following lines. Def Pos P_tGripperVac Def Pos P_tGripperWP Def Pos P_tGripperFuse Def Pos P_tGripperFuse
  - 3. Change following positions to reference positions in UBP.pos. The location of tool magazine in CIROS model is different from actual robot. Thus, the offsets are too large to be ignored.

P_GrpStorageVac = (-395.65,-125.00,484.90,-180.00,-0.00,-0.00)(7,1) P_GrpStorageWp = (-395.65,0.00,484.90,-180.00,0.00,0.00)(7,1) P_GrpStorageFuse = (-395.65,125.00,484.90,-180.00,-0.00,-0.00)(7,0)

- Comment out or remove all lines calling camera related subprograms, for example the line calling GetCamResult in program 1 to 5.mb5.
   REM CallP "GetCamResult", CamPrgNumber%
- 10. Save all and compile the project.



### **CP-F-RASS** Robot Programming

#### Robot programming example 'gripper test' (1)

- The program is independent of all the other robot programs, but uses positions in UBP.pos.
- It is written in MBA V and does the following.
  - 1. Move robot arm to home position.
  - 2. Move robot to parallel workpiece gripper magazine.
  - 3. Mount the gripper.
  - 4. Remove gripper from magazine.
  - 5. Open gripper.
  - 6. Close gripper.
  - 7. Store gripper back to magazine.
  - 8. Move robot arm back to home position.

Note: For more programming example, see tutorial video: 50_RASS-Programming.mp4.



### **CP-F-RASS Robot Programming**

#### Robot programming example 'gripper test' (2)

- 1. Create a CIROS project and load CP-F-RASS from CP System model libraries.
- 2. Make sure a project is assigned to controller Montage_RV-4FL and UBP.pos is in the project.
- 3. In Project Management, right click on Projects → <project name> → Files and choose new.
- 4. Create a Melfa Basic V program and name it RASS-GripperTest.mb5.
- 5. With the programming window being the active window, select Programming  $\rightarrow$  Programming assistant.
- 6. Uncheck Declare inputs and outputs and click OK.
- 7. In the programming window, add the lines shown in right.
- 8. Save the program.
- 9. In Project Management, right click on the program and select Set main program.
- 10. Compile the project.
- 11. Run simulation.

#### ' TODO add your code here

' Move to home position MOV P_Home DLY 1

Mount parallel workpiece gripper
MOV P_PCBPalletHelp
MOV P_GrpStorageWp, -30
JOVRD 50
MVS P_GrpStorageWp
HClose 3
MVS P_GrpStorageWp + P_ToolX80
DLY 1

' Open and close gripper HClose 1 HOpen 1 DLY 1 HOpen 1 HClose 1 DLY 1 ' Store gripper back to magazine MVS P_GrpStorageWP HOpen 3

' Move back to home position MVS P_GrpStorageWp, -30 JOVRD 100 MOV P_PCBPalletHelp MOV P_Home

END



### **Move Robot Manually**

- There are several ways to move robot manually in CIROS.
- 1. Move the robot directly to a position on position list.
  - Double click on the position.

🗾 [R\	/-2F]\Position Lists\12.POS		
No	Position	Orientation	Comment
Pl	249.9,-38.3,192.2	179,0,-122,R,A,N	^
P3	311.9,24.8,180.0	180,-0,-180,R,A,N	
P4	449.7,143.6,176.3	180,-0,-90,R,A,N	
P50	311.0,73.1,194.8	180,0,-135,R,A,N	
P51	397.5,72.9,194.7	180,0,-45,R,A,N	
P52	311.4,148.1,195.2	180,-0,-135,R,A,N	
P53	397.9,147.9,195.1	180,0,-45,R,A,N	
P6	132.0,259.1,323.0	180,0,90,0.0,0.0,R,A,N	

2. Double click on any location in model window.

#### **FESTO**

## **Move Robot Manually**

- 3. Move the robot with Teach-In panel.
  - Gripper can be controlled in section Gripper output.

PROGRAMMING SIMULATI		SIMULATION	EXTRAS	SETT	
	Compile	Ctrl+F9			
	Teach-In		F	-8	
	Position list			>	
<u>_</u>	RAPID progra	amming tools		>	
	Project mana	agement			
	Programmin	g assistant			
	Mitsubishi re	Ctrl+R			
	Sort				

Joint coordinates (	Cartesian coordii	nates
Move		Joint values
Waist	← →	0.00 • -240.00 to +240.00
Shoulder	← →	0.00 • -120.00 to +120.00
Elbow	← →	90.00 • +0.00 to +160.00
Twist	← →	0.00 • -200.00 to +200.00
Pitch	← →	90.00 • -120.00 to +120.00
Roll	← →	0.00 • -360.00 to +360.00
Hold TCP pose	2	Apply
Speed override		Gripper output
-	10 %	HCLOSE1 V Close



Teach-In	x					
Joint coordinates Cartesian coordinates						
Move	TCP pose					
X-axis 🛏 🗕 👅 🕗	x 270.00 mm R 180.00 �					
Y-axis 🔶 🔶 🕑	y 0.00 mm P 0.00 �					
Z-axis 🛏 🗕 🐧 🕗	z 505.00 mm Y 180.00 �					
Translation Rotation	Apply					
Reference system	Reference system					
Robot base $\sim$	Robot base $\checkmark$					
тср	Configuration					
<flange tcp=""> ~</flange>	R, A, N $\sim$					
Speed override Gripper output						
10 %	HCLOSE1 V Close					
Move incrementally						
25.00 mm 5.00 🚸						



### Mount and Release a Gripper Manually

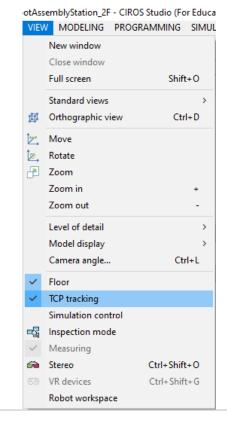
- In some system, for example, CP-F-RASS, TCP changes relative to the gripper.
- Steps to mount a gripper manually with example CP-F-RASS model.
  - 1. Select < Automatic TCP>.
  - 2. Move the robot arm to the tool position.
  - 3. In Teach-In panel  $\rightarrow$  Gripper output, close HClose_3.
  - 4. Start simulation (F5).
  - 5. Stop simulation (F5).
  - 6. The gripper is mounted.

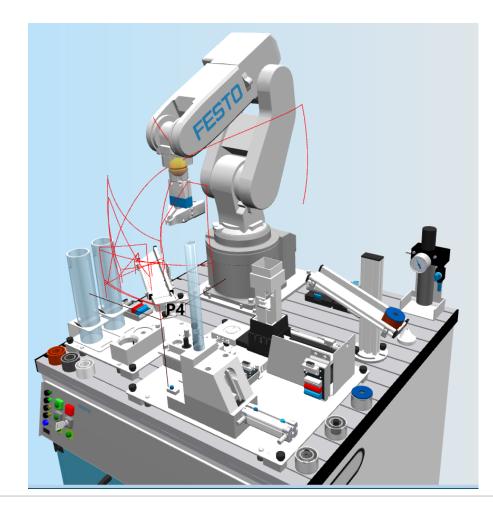
- Steps to release a gripper manually with example CP-F-RASS model.
  - 1. Move the robot arm to the desired position.
  - 2. In Teach-In panel  $\rightarrow$  Gripper output, open HClose_3.
  - 3. Start simulation (F5).
  - 4. Stop simulation (F5).
  - 5. The gripper is released.



## **TCP** Tracking

- TCP path of robot movement can be monitored.
- Active TCP tracking from View  $\rightarrow$  TCP tracking.







## **View TCP Coordinate**

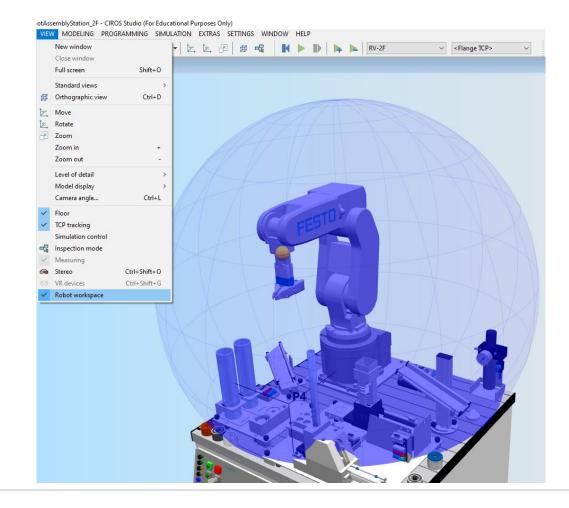
• Joint coordinate and cartesian coordinate of the active TCP can be monitored.

WIN	DOW HELP				Hanna		~	
	Logging	>	CP>	· · · · ·			2	k
	Robot position	>		Show joint coordinat	ec		E7	,
	Workspaces	>		Show robot coordina		Sh	ift+F7	
#	I/O connection monitor			Joint Coordinates				
	Messages			Waist	0.0	Deg		
	Cascade			Shoulder Elbow	0.0	Deg		
	Tile vertically			Twist		Deg Deg		
	Tile horizontally			Pitch		Deg		
	Arrange icons			Roll	0.0	Deg		
	Close all			Robot Coordinates		×		
	Close all position and program windows			x:	270.0	mm		
	1 [RV-2F]\\\\\\PROGRAM FILES\\Position Lists\12.POS			y:		mm		
	2 Model				505.0 180.0			
•				• P:		Deg		
3 [RV-2F]\\\\Program Files\\Programs\12.MB5 *				<ul> <li>Y:</li> <li>Right, Above, No-Flip</li> </ul>	180.0	Deg		



### **Robot Workspace**

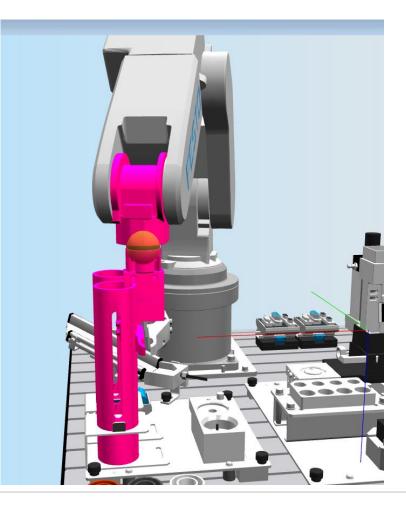
- The room that TCP can reach.
- Activate at View → Robot workspace.





## **Collision Detection**

- Collision detection can be activated.
- In collision detection mode the movements are always incremental.
- It is useful in testing a new robot program to avoid collision.





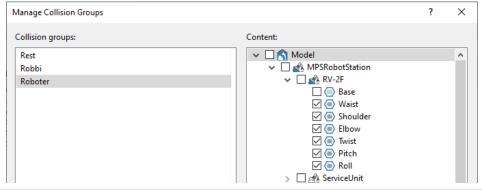
### **Collision Detection**

### **Configuration (1)**

- 1. Assign group. Collision is only detected when elements in deferent groups cross each other.
  - 1. Select Settings → Collision detection.

	đ	Collision detection						
i	=	OPC server						
	$\leq$	Simulation film script						
		Mode	el options		Ctrl+I			

- 2. In Collision Detection window, select Manage collision groups.
- 3. Assign content to Roboter group.



4. Assign content to Rest group.

Manage Collision Groups		?	×
Collision groups:	Content:		
Rest Robbi Roboter	✓       Model         ✓       MPSRobotStation         >       RV-2F         >       ServiceUnit         ✓       ModelRobotHandling         >       ModulRobotAssembly         ✓       ModulRobotAssembly         ✓       Springs         ✓       Pistons         ✓       GUI         ✓       C-Interface_2         ✓       AdapterPlate         ✓       Trolley700         ✓       SafetyHousing         ✓       Multigrip         ✓       Templates		
		Clo	se



### **Collision Detection**

### **Configuration (2)**

- 2. Assign the collision group pair.
  - 1. Close Manage collision groups window.
  - 2. Select Roboter as group 1 and Rest as group 2.
  - 3. Move the pair to the right to activate it.
  - 4. Select OK.

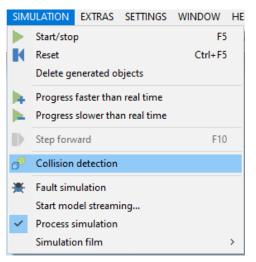
Collision Detection		?	×
Collision group 1 Roboter Rest Robbi <all objects=""> <all robots=""> <all templates=""> Collision group 2 Roboter Rest Kobbi <all objects=""> <all robots=""> <all templates=""></all></all></all></all></all></all>	Test pairs of collision groups         Roboter <-> Rest         <		
Manage collision groups	Cancel	OK	



### **Collision Detection**

#### Activate simulation

1. Select Simulation  $\rightarrow$  Collision detection.



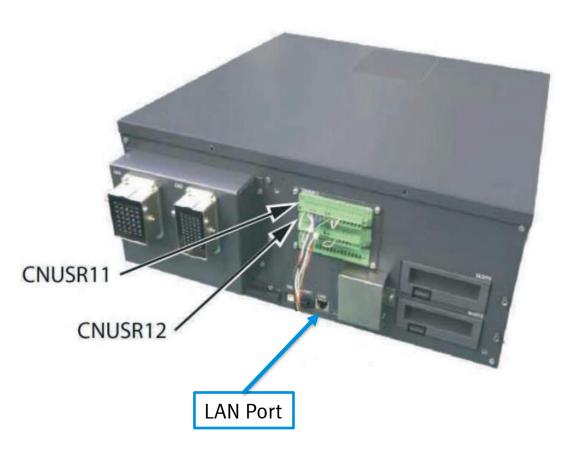
2. Move the robot against an object and observe the simulation.



## Only CIROS Studio Connect to Robot Controller

#### **Communication interface**

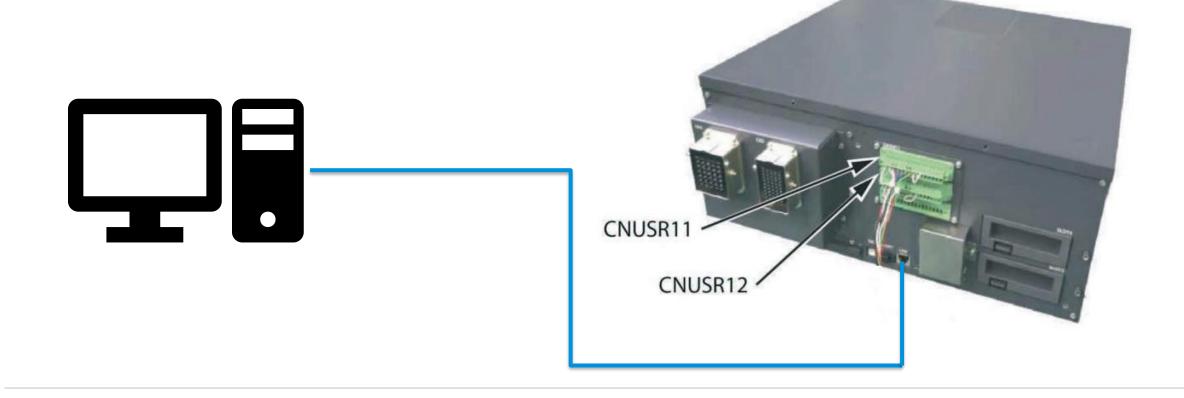
- Ethernet cable
- TCP/IP protocol
- IP-Address and Port





# Only CIROS Studio Check the Ethernet Cable

Please check the ethernet connection between robot controller and the computer.





### Only CIROS Studio Connect to Robot Controller

#### Find the IP-address from R32TB

1. The IP-address can be read from robot teach box (TB).



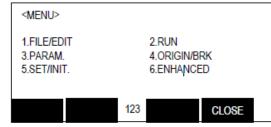
2. Exit everything until home page.

MELFA CRnD-7xx	/er. P2T
RV-6SDL	
COPYRIGHT (C) 2008 MITSUBISHI TRIC CORPORATION ALL RIGHTS RVED	ELEC RESE

3. Press F1 to enter menu.



4. Select **3.PARAM.** 



5. Search for parameter **NETIP**. Read the IP-address.

-						
	<pre><parameter></parameter></pre>		NAME (NETIP)			
	DATA			ELE()		
	(192.16	8.0.20				
			ABC		CLOSE	
			1.00		OLOOL	



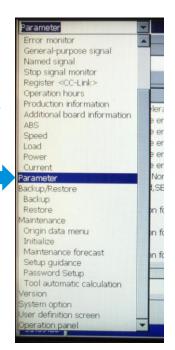
### Only CIROS Studio Connect to Robot Controller

#### Find the IP-address from R56TB

1. The IP-address can be read from robot teach box (TB).



2. Select "Parameter" from the menu.



Video Tutorial: 53_FindMitsulpR56TB.mp4

3. Click on "Parameter name" to search for parameter "NETIP".



4. Key in "NETIP" and click "Enter".



5. IP Address is shown.



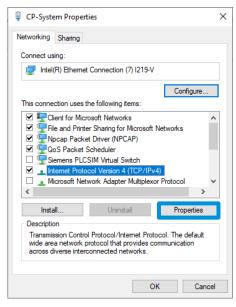


### **Only CIROS Studio**

### **Connect to Robot Controller**

#### **Change the IP-address of the computer to the same network**

1. Open network adapter properties.



2. Select Internet Protocol Version 4 (TCP/IPv4).

- 3. Select Properties.
- 4. Change the IP address to the same network as robot

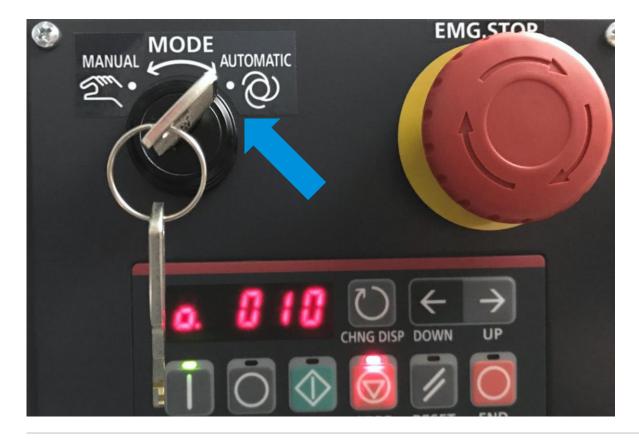
#### controller.

Internet Protocol Version 4 (TCP/IPv4) Properties								
General								
You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.								
Obtain an IP address automatically								
• Use the following IP address:								
IP address:	172 . 21 . 0 . 90							
Subnet mask:	255 . 255 . 192 . 0							
Default gateway:								
Obtain DNS server address autom	natically							
Use the following DNS server add	resses:							
Preferred DNS server:								
Alternative DNS server:								
Validate settings upon exit Advanced								
OK Cancel								



### Only CIROS Studio Connect to Robot Controller

Set robot controller to Automatic and switch of teach mode.







#### **Only CIROS Studio**

### **Connect to Robot Controller**

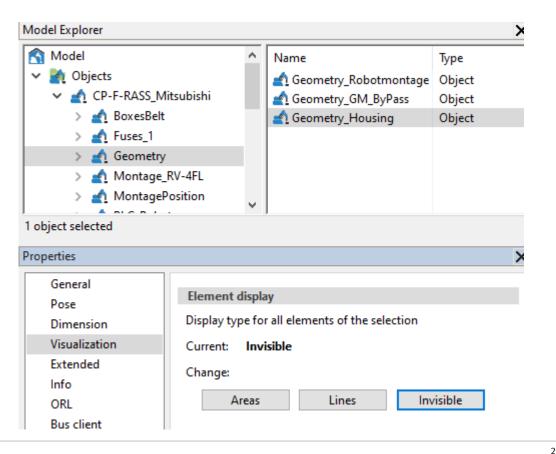
#### Insert CP-F-RASS model

- 1. Create a new CIROS project.
- 2. Insert "CP-F-RASS_Mitsubishi" from Festo CP System model library.

Video tutorial: 54_InsertCP-F-RASS.mp4

 Optional: To hide robot housing, in Model Explorer, select "Geometrie\Geometrie_Umhausung". In Properties\Visualization, select "Invisible".

Video tutorial: 56_HideHousing.mp4

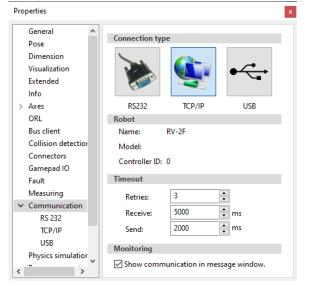




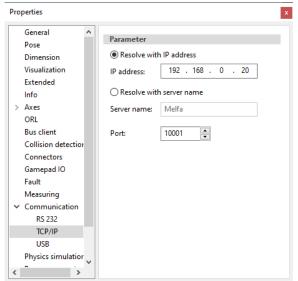
### Only CIROS Studio Connect to Robot Controller

#### **Configure communication setting**

- 1. In Model Explorer, select "CP-F-RASS_Mitsubishi\Montage_RV-4FL".
- 2. In properties window, select **Communication**.
- 3. Select **TCP/IP** as connection type.



- 4. Expand Communication in properties window and select **TCP/IP**.
- 5. Enter the IP-address of the robot controller.
- 6. Port = 10001

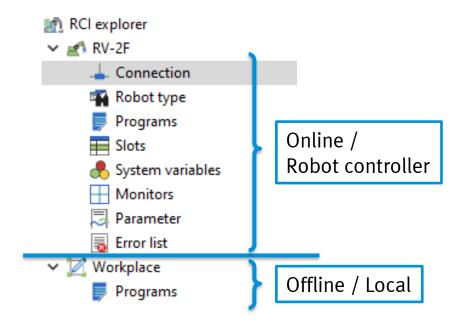




### Only CIROS Studio Connect to Robot Controller

### **RCI-Explorer**

- RCI = Robot Control Interface
- Allow user to read information, program and control the robot controller in CIROS.
- Can create / load robot controller backup.
- Edit the program by uploading the robot program into local workspace.



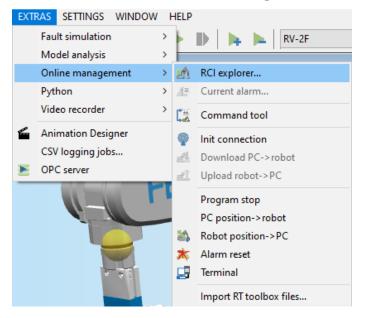


### **Only CIROS Studio**

### **Connect to Robot Controller**

#### **Open RCI-Explorer**

• Select Extras > Online management > RCI explorer...





### **Only CIROS Studio**

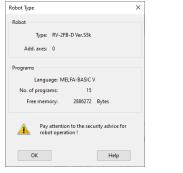
### **Connect to Robot Controller**

#### **Build the connection**

1. In RCI Explorer, right click on Connection and select **Connect**.



- 2. Please pay attention to the security advice and environment of real robot!
- 3. Select **OK** when the window pops up.



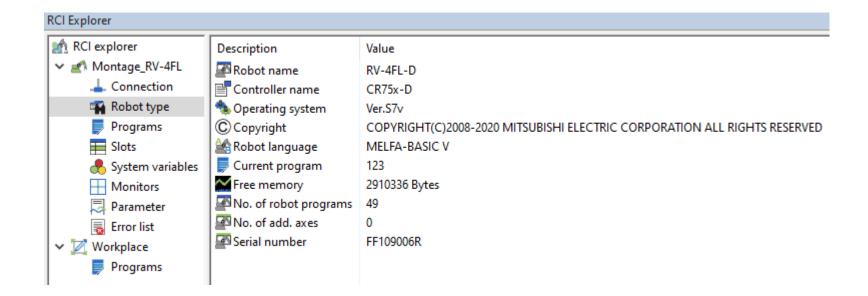
4. Connection is established.

n RCI explorer ✓ 🛃 RV-2F	Description	Value
Connection	Connection type	TCP/IP
Robot type	Lange State	Connection to robot established
Programs		
E Slots		
🐣 System variables		
📮 Parameter		
👼 Error list		
🖌 📝 Workplace		
Programs		

Video tutorial: 57_ConnectMitsuRobot.mp4



Robot type





#### **Robot programs**

CI Explorer				
🔬 RCI explorer	File name	Size	Saved at	Lines
✓ 🛃 Montage_RV-4FL	1	3285 Bytes	22-11-30	91
📥 Connection	2	3495 Bytes	22-11-30	102
🖷 Robot type	5 3	3507 Bytes	22-11-30	102
📮 Programs	<b>5</b> 4	3770 Bytes	22-11-30	114
Slots	5	3028 Bytes	22-11-30	84
롽 System variables	5 💭 6	1974 Bytes	22-11-30	50
Monitors	5 11	962 Bytes	22-11-30	28
Parameter	5 12	1312 Bytes	22-11-30	26
<b>Error list</b>	5 13	1115 Bytes	22-11-30	25
✓ ☑ Workplace	14	1117 Bytes	22-11-30	25
Programs	123	5086 Bytes	22-11-30	141
	234	3140 Bytes	22-11-30	83
	255	1240 Bytes	22-11-30	18
	900	1871 Bytes	22-11-30	37
	999	3305 Bytes	22-11-30	63
	UBP	7393 Bytes	22-12-01	112
	GRPLOCK	765 Bytes	22-11-30	20
	GRPOPEN	1090 Bytes	22-11-30	26
		977 Bytes	22-11-30	23
	GRPCLOSE	1092 Bytes	22-11-30	26
	🛛 💭 GRPVACON	806 Bytes	22-11-30	20

_			
MOUNTPCB	1399 Bytes	22-11-30	36
📑 GRPVACOFF	798 Bytes	22-11-30	19
📕 GRPRELEASE	792 Bytes	22-11-30	20
📑 INITIALIZE	1481 Bytes	22-11-30	30
📕 CAMERACALIB	9420 Bytes	22-11-30	240
S MONITORHOME	2243 Bytes	22-11-30	53
PICKNEWTOOL	5361 Bytes	22-11-30	178
PLROTWP2STP	3122 Bytes	22-11-30	84
SENSORCHECK	3747 Bytes	22-11-30	85
<b>DEMOMOUNTPCB</b>	4746 Bytes	22-11-30	135
📑 ENRGSAVEVACU	1459 Bytes	22-11-30	35
📕 GETCAMRESULT	7994 Bytes	22-11-30	157
📕 GETCURTOOLNO	2953 Bytes	22-11-30	61
📕 GETFUSEMAGNO	2716 Bytes	22-11-30	68
MONITORPALWS	2268 Bytes	22-11-30	53
S MOUNTBOTFUSE	1217 Bytes	22-11-30	34
MOUNTTOPFUSE	1211 Bytes	22-11-30	34
PCBTRAYCNTRL	1341 Bytes	22-11-30	31
PICKFRMSTOPR	3008 Bytes	22-11-30	83
PICKFRMVISON	1493 Bytes	22-11-30	39
📕 PICKFUSFRMAG	2892 Bytes	22-11-30	79
PICKPCBFRPAL	3897 Bytes	22-11-30	97
PICKWPFRMASS	1767 Bytes	22-11-30	42
	2102 Buter	22-11-20	25



#### Robot programs in slots

CI Explorer						
🔬 RCI explorer	Slot no.	Program name	Condition	Mode	Priority	State
∨ 🛃 Montage_RV-4FL	E Slot 1	123	START	сүс	1	Stop
📥 Connection	Slot 2	MONITORHOME	ALWAYS	REP	1	Run
🌇 Robot type	Slot 3	ENRGSAVEVACU	ALWAYS	REP	1	Run
Programs	E Slot 4	MONITORPALWS	ALWAYS	REP	1	Run
E Slots	E Slot 5	PCBTRAYCNTRL	ALWAYS	REP	1	Run
Å System variables	ESlot 6		START	REP	1	Empty
- Monitors	ESlot 7		START	REP	1	Empty
👼 Parameter	E Slot 8		START	REP	1	Empty
a Error list						
✓ ☑ Workplace						
Programs						



#### System variables

RCI Explorer							
🔬 RCI explorer	Name	Value					
∨ 🛃 Montage_RV-4FL	₽_CURR	(-393.54, +109.36, +523.69, +179.90, +0.66, -0.19)(7, 15)					
📥 Connection	₿J_CURR	(-195.59,+13.35,+92.42,+0.28,+73.62,-51.98)					
🖷 Robot type	🛃 J_ECURR	(-652813120.00, + 55691384.00, + 555274048.00, + 787244.00, + 177462512.00, -105353128.00)					
Programs	🛃 J_FBC	(-195.59,+13.35,+92.42,+0.28,+73.62,-51.98)					
E Slots	🐣 P_FBC	(-393.54, +109.36, +523.69, +179.90, +0.66, -0.19)(7,15)					
🐣 System variables	🐣 M_CMPDST	+0					
Monitors	P_TOOL	(+0.00,+0.00,+0.00,+0.00,+36.50)(0,0)					
👼 Parameter	🐣 P_BASE	(+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)					
👼 Error list	P_NTOOL	(+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)					
✓ ☑ Workplace	P_NBASE	(+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)					
Programs	M_HNDCQ						
	M_OPOVRD	+10					
	M_OVRD	+10					
	M_JOVRD	+100					
	M_NOVRD	+100					
		+100					
		+1					
		+0					
		+100					
	M_RDST	+0					
	🐣 M_RSPD						

🔥 M_SPD	+10000
🐣 M_NSPD	+10000
M_ACL	+100
M_DACL	+100
M_NACL	+100
& M_NDACL	+100
M_ACLSTS	+0
🔥 M_RUN	+0
🐣 M_WAI	+1
🔥 M_PSA	+0
AM_CYS	+1
& M_CSTP	+0
M_ECURR	
C_PRG	"123"
🔥 с_сом	
🐣 M_ERR	+0
M_ERRLVL	+0
🔥 M_ERRNO	+0
🛃 M_SVO	+1
🛃 M_UAR	+0
A_IN(0)	+0
A_INB(0)	+0
A_INW(0)	+0
	+0



#### Parameters

RCI Explorer				CPRCE	11
🛃 RCI explorer	Name	Value	Description		
✓      Montage_RV-4FL			Number of Multitask	CBUFE	
📥 Connection	RETPATH		Use resumed buffer 2:ON(MVS)/1:ON(MOV)/0:OFF	CBAUE	
🖷 Robot type			Stop input signal normally open(0)/close(1)		
Programs			Continue function Disable/Enable (0/1)		
Slots	TBCAN			CSTOP	
System variables	TBBZR		T/B Buzzer (1:ON, 0:OFF)	CTERM	
Monitors	COMDEV		Definition of device for COM:1-8		
Parameter	CDTR422		DTR control[0:OFF,1:ON]		
Error list	CBAU232		Baud rate		
✓ ☑ Workplace	CLEN232		Length[8,7]		
Programs	CPRTY232		Parity[0:None,1:odd,2:even]		
· · · · · · · · · · · · · · · · · · ·	CSTOP232		Stop bit[1,2]	CSTOP	
	CTERM232		Termination[0:CR 1:CR+LF]		
	CPRC232		Protocol[0:Non, 1:Process , 2:Data link]		
	CDTR232		DTR control[0:OFF,1:ON]		
	CBUF232		Receive buffer mode (0:Ring, 1:Fix)		
	CBAUE11		Baud rate		
	CLENE11		Length[8,7]		
	CPRTYE11		Parity[0:None,1:odd,2:even]		
	CSTOPE11		Stop bit[1,2]		
	CTERME11		Termination[0:CR 1:CR+LF]		

Protocol[0:Non, 1:Process, 2:Data link]
DTR control[0:OFF, 1:ON]
Receive buffer mode (0:Ring, 1:Fix)
Baud rate
Length[8,7]
Parity[0:None,1:odd,2:even]
Stop bit[1,2]
Termination[0:CR 1:CR+LF]
Protocol[0:Non,1:Process,2:Data link]
DTR control[0:OFF,1:ON]
Receive buffer mode (0:Ring, 1:Fix)
Baud rate
Length[8,7]
Parity[0:None,1:odd,2:even]
Stop bit[1,2]
Termination[0:CR 1:CR+LF]
Protocol[0:Non,1:Process,2:Data link]
DTR control[0:OFF,1:ON]
Receive buffer mode (0:Ring, 1:Fix)
Termination[0:CR 1:CR+LF]
Protocol[0:Non,1:Process,2:Data link]
Receive buffer mode (0:Ring, 1:Fix)
Termination[0:CR 1:CR+LF]
Protocolf0:Non 1:Process 2:Data link1

#### **FESTO**

# Only CIROS Studio Read Live Parameter Value

#### CI Explorer

			· · · · · · · · · · · · · · · · · · ·			
🛃 RCI explorer	Name	Value	Description			
∨ 🛃 Montage_RV-4FL	FSEST06		Data used in the force sensor calibration #6			
📥 Connection	FSEST07		Data used in the force sensor calibration #7			
🖷 Robot type	FSEST08		Data used in the force sensor calibration #8			
Programs	FSEST09		Data used in the force sensor calibration #9			
E Slots	<b>FSHAND</b>		Selection of left/right-handed coordinate system for			
롽 System variables	SIFVER		Force sensor I/F unit S/W version			
Monitors	A MVTERM		Specification of movement in 'CNT 0' (0/1/2 = VEL o			
Parameter	CNTSPEC		Specification of initial setting for CNT/END (1:New/0			
Error list	JACLADJ		Acceleration adjustment for Mov (ON=1/OFF=0)			
✓ [™] Workplace	A MEINST		Install bit pattern (1:end)			
Programs	A MEINSZ		Pulse data for install			
	A MEOFFZ		Single turn pulse for install			
	A MEJINS		Pulse of single turn for JRC			
	A MEINSD		Check data for install			
	JRCEXE		JRC Enable 1/0 = Enable/Disable			
	JRCQTT		Position Shift Quantity of JRC			
	JRCORG		Origin of JRC 0			
	JOGSPMX		JOG maximum speed (under 250[mm/sec])			
	JOGJSP		Joint JOG speed(High/Low/JOG OVRD)			
	JOGPSP		POSE JOG speed(High/Low/JOG OVRD)			
	JOGELBMD		JOG speed restriction mode of elbow of joi 🌽 Properties			
			Selected work coordinate number (JOG)			
	METWM2		Compensation for flexure ON/OFF = 1/0			
	APMODE 🖓		MAP ACC High Acceleration ON/OFF = 1/0			
	LJOGMD		Speed adjustment of 3-axis JOG (1:Enable/0:Disable)			
			Error of origin [doa] dl1 dl2 dl2 dl4 dl5 dl6			

Properties of JO	GJSP	$\times$
Parameter		
Current value:	0.10, 0.01, 5.00	
New value:	0.10, 0.01, 5.00	
Parameter desc	ription:	-
	ed(High/Low/JOG OVRD)	
	~	
Last refresh:	November 30, 2022 16:43:27	
	OK Cancel Help	



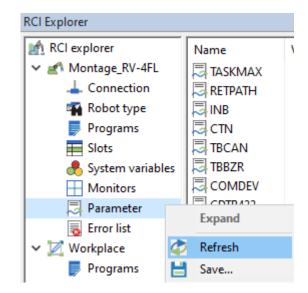
#### **Error list**

👘 RCI explorer	Date	Time	Error no.	Message
✓ 🛃 Montage_RV-4FL	A 22-12-01	00:19:22	2602	DSTN pos. exceeds the limit
📥 Connection	A22-12-01	00:16:22	2602	DSTN pos. exceeds the limit
🖷 Robot type	A22-12-01	00:16:22	2602	DSTN pos. exceeds the limit
Programs	<u>A</u> 22-12-01	00:16:08	2602	DSTN pos. exceeds the limit
E Slots	122-12-01	00:16:08	2602	DSTN pos. exceeds the limit
🐣 System variables	122-12-01	00:16:06	2602	DSTN pos. exceeds the limit
Monitors	122-12-01	00:15:54	2602	DSTN pos. exceeds the limit
🔜 Parameter	122-12-01	00:15:50	2602	DSTN pos. exceeds the limit
5 Error list	22-11-30	23:46:38	40	Door Switch Signal is Input
V 🕅 Workplace	22-11-30	23:46:26	40	Door Switch Signal is Input
Programs	22-11-30	23:46:20	40	Door Switch Signal is Input
<b>W</b>	22-11-30	23:46:08	40	Door Switch Signal is Input



#### **Refresh window**

• Sometimes, the window is not updated. In this case, the windows can be refreshed.



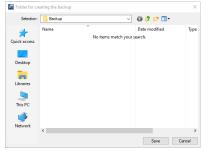


### Only CIROS Studio Create Robot Controller Backup

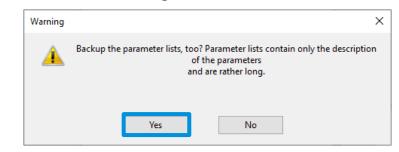
1. Right click on the robot and select Create backup\All.

鮒 RCI explorer		Description		Value		
✓	Reduce	1	Folder Folder			
🙀 Rc 🕨	Start (CYC)	Start (CYC)				
📕 Pr 🜔	Start (REP)		L .	Folder		
📰 Sli 💼	Stop		able	s Folder		
🐣 Sy 🗍						
Ш М	Servos on			Folder		
🔜 Pa	Servos off		Folder			
👼 Er 🕂	Current alarm		boar			
🗸 🔀 Work	Reset		ls	1		
Pr	Get position (robot - Set position (PC -> ro			-		
p	Properties					
	Create backup	>	D	All		
	Load backup	>	RG	Programs		
	Restart controller		6	Parameter		
			6	System programs		

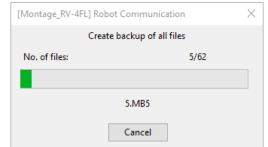
2. Create the backup in an empty folder.



#### 2. Confirm the warning.



3. Creating backup...



Video tutorial: 59_CreateRobotBackup.mp4



# Only CIROS Studio Robot Controller Backup Folder

• By default, backup is located in <project folder>\Backup.

Name	Date modified	Туре	Size
📙 Backup	30/11/2022 16:36	File folder	
CF	30/11/2022 16:23	File folder	
Montage_RV-4FL	30/11/2022 16:40	File folder	
Textures	30/11/2022 16:24	File folder	
RASS2_717.err	30/11/2022 16:33	ERR File	6 KB
📓 RASS2_717.ini	30/11/2022 16:24	Configuration sett	24 KB
🕋 RASS2_717.modx	30/11/2022 16:24	CIROS Model	35.471 KB
🗋 RASS2_717.par	30/11/2022 16:32	PAR File	1 KB
澷 translate.py	30/11/2022 15:10	Python File	9 KB

> This PC > Documents > CIROS > Projects > RASS2_717 > Backup							
Name	Date modified	Туре	Size				
1.MB5	30/11/2022 16:35	MB5 File	4 KB				
* 2.MB5	30/11/2022 16:35	MB5 File	4 KB				
🤺 📄 З.МВ5	30/11/2022 16:35	MB5 File	4 KB				
* 1.MB5	30/11/2022 16:35	MB5 File	4 KB				
* 5.MB5	30/11/2022 16:35	MB5 File	3 KB				
* 6.MB5	30/11/2022 16:35	MB5 File	2 KB				
🖈 📄 11.MB5	30/11/2022 16:35	MB5 File	1 KB				
12.MB5	30/11/2022 16:35	MB5 File	2 KB				
13.MB5	30/11/2022 16:35	MB5 File	2 KB				
14.MB5	30/11/2022 16:35	MB5 File	2 KB				
123.MB5	30/11/2022 16:35	MB5 File	5 KB				
234.MB5	30/11/2022 16:35	MB5 File	4 KB				
255.MB5	30/11/2022 16:35	MB5 File	2 KB				
900.MB5	30/11/2022 16:35	MB5 File	2 KB				
999.MB5	30/11/2022 16:35	MB5 File	4 KB				
AError.log	30/11/2022 16:35	Text Document	57 KB				
BACKUP.COS	30/11/2022 16:37	COS File	1 KB				
CAMERACALIB.MB5	30/11/2022 16:35	MB5 File	10 KB				
CError.log	30/11/2022 16:35	Text Document	4 KB				
COMMON.LST	30/11/2022 16:36	LST File	160 KB				
COMMON.PRM	30/11/2022 16:35	PRM File	2 KB				
DEMOMOUNTPCB.MB	<b>30/11/2022 16:36</b>	MB5 File	5 KB				
ENRGSAVEVACU.MB5	30/11/2022 16:36	MB5 File	2 KB				
GETCAMRESULT.MB5	30/11/2022 16:36	MB5 File	8 KB				
GETCURTOOLNO.MB5	30/11/2022 16:36	MB5 File	3 KB				
GETFUSEMAGNO.MB5	30/11/2022 16:36	MB5 File	3 KB				

20/11/2022 16:25

MP5 Eilo

iS.

s ts

5

ts s

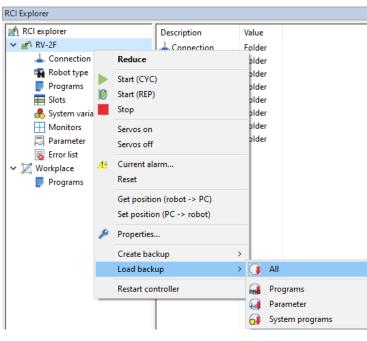
CPDCLOSE MP5

2 40



## Only CIROS Studio

1. Right click on the robot and select **Load backup > All**.



- 2. Select backup folder.
- 3. Restart the controller after loading complete.





# Only CIROS Studio Upload Robot Programs

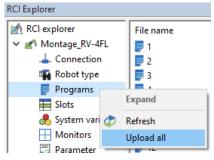
- Robot programs can be uploaded to the computer.
- Uploaded programs are listed in Workplace\Programs.
- The programs can then be edited.

📩 RCI explorer	File name	Size	Туре	Saved at
✓ 🛃 RV-2F	RV-2F			
📥 Connection	🔲 11.POS	3 KB	POS	22.09.2021 13:53
🖷 Robot type	15.POS	2 KB	POS	22.09.2021 13:54
Programs	3.POS	3 KB	POS	22.09.2021 13:53
E Slots	5.POS	1 KB	POS	22.09.2021 13:53
🐣 System variables	6.POS	1 KB	POS	22.09.2021 13:53
	99.POS	1 KB	POS	22.09.2021 13:54
🔜 Parameter	A1.POS	1 KB	POS	22.09.2021 13:54
a Error list	A2.POS	1 KB	POS	22.09.2021 13:54
🗸 🔀 Workplace	A3.POS	1 KB	POS	22.09.2021 13:54
📮 Programs	A4.POS	1 KB	POS	22.09.2021 13:54
	A5.POS	1 KB	POS	22.09.2021 13:54
	APP1.POS	1 KB	POS	22.09.2021 13:54
	T1.POS	0 KB	POS	22.09.2021 13:54
	= 11.MB5	39 KB	MB5	22.09.2021 13:53
	= 15.MB5	1 KB	MB5	22.09.2021 13:54
	3.MB5	19 KB	MB5	22.09.2021 13:53
	5.MB5	16 KB	MB5	22.09.2021 13:53
	5.MB5	11 KB	MB5	22.09.2021 13:53
	99.MB5	1 KB	MB5	22.09.2021 13:54
	F A1.MB5	2 KB	MB5	22.09.2021 13:54
	📕 A2.MB5	5 KB	MB5	22.09.2021 13:54
	📕 A3.MB5	5 KB	MB5	22.09.2021 13:54
	📕 A4.MB5	4 KB	MB5	22.09.2021 13:54
	📕 A5.MB5	4 KB	MB5	22.09.2021 13:54
	APP1.MB5	1 KB	MB5	22.09.2021 13:54
	📕 T1.MB5	1 KB	MB5	22.09.2021 13:54



## Only CIROS Studio Upload Robot Programs

1. In RCI Explorer, right click on Programs, select Upload all.



2. When this window pops out, click ok.

[Montage_	RV-4FL] -> PC	
Program		
🗹 Complete prog	gram	
🗹 Delete all befor	re downloading	
From line:	1	
To line:	9999	
Positions		
All positions		
From position:	1	
To position:	900	
Name		
1		
Load each line s	eparately	
Show dialog		

3. Programs which are running in slots cannot be uploaded. Click ok and continue uploading the missing programs.

Error	>	<	Warning		$\times$
8	Command failed, because a program is running. Stop the running program and try again.		<u> </u>	Error during upload. Do you want to continue to upload the missing programs?	
	ОК			Yes No	

Video tutorial: 60_UploadRobotPrograms.mp4



## Only CIROS Studio Robot Program Folder

- By default, uploaded robot programs are located in <project folder>\<robot controller name>.
  - For example,
    - CIROS project : RASS2_717
    - Robot name : Montage_RV-4FL
  - Uploaded robot programs are located in ..\RASS2_717\Montage_RV-4FL.

Name	Date modified	Туре	Size
📙 Backup	30/11/2022 16:36	File folder	
CF	30/11/2022 16:23	File folder	
	30/11/2022 16:40	File folder	
Textures	30/11/2022 16:24	File folder	
RASS2_717.err	30/11/2022 16:33	ERR File	6 KB
RASS2_717.ini	30/11/2022 16:24	Configuration sett	24 KB
TASS2_717.modx	30/11/2022 16:24	CIROS Model	35.471 KB
RASS2_717.par	30/11/2022 16:32	PAR File	1 KB
📄 translate.py	30/11/2022 15:10	Python File	9 KB

#### This PC > Documents > CIROS > Projects > RASS2_717 > Montage_RV-4FL

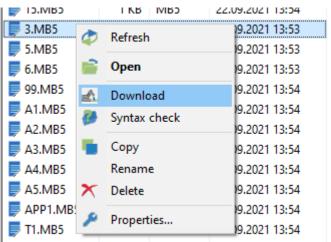
*

Name	Date modified	Туре	Size
1.MB5	30/11/2022 16:39	MB5 File	3 KB
1.POS	30/11/2022 16:39	POS File	1 KB
2.MB5	30/11/2022 16:39	MB5 File	4 KB
 2.POS	30/11/2022 16:39	POS File	1 KB
3.MB5	30/11/2022 16:39	MB5 File	4 KB
3.POS	30/11/2022 16:39	POS File	1 KB
4.MB5	30/11/2022 16:39	MB5 File	4 KB
4.POS	30/11/2022 16:39	POS File	1 KB
5.MB5	30/11/2022 16:39	MB5 File	3 KB
5.POS	30/11/2022 16:39	POS File	1 KB
6.MB5	30/11/2022 16:39	MB5 File	2 KB
6.POS	30/11/2022 16:39	POS File	0 KB
11.MB5	30/11/2022 16:39	MB5 File	1 KB
11.POS	30/11/2022 16:39	POS File	0 KB
12.MB5	30/11/2022 16:39	MB5 File	1 KB
12.POS	30/11/2022 16:39	POS File	1 KB
13.MB5	30/11/2022 16:39	MB5 File	1 KB
13.POS	30/11/2022 16:39	POS File	1 KB
14.MB5	30/11/2022 16:39	MB5 File	1 KB
14.POS	30/11/2022 16:39	POS File	1 KB
123.MB5	30/11/2022 16:39	MB5 File	5 KB
123.POS	30/11/2022 16:39	POS File	1 KB
234.MB5	30/11/2022 16:39	MB5 File	3 KB
234.POS	30/11/2022 16:39	POS File	1 KB
255.MB5	30/11/2022 16:39	MB5 File	1 KB
255 DOS	20/11/2022 16:20	DOS File	1 KR



## Only CIROS Studio Download Program

- Programs in Workspace can be downloaded into robot controller.
- 1. Select the MB5 program in workspace.
- 2. Right click and select Download.



#### FESTO

## Only CIROS Studio Online Teach-In

- Once connected, it is possible to activate online teaching in Teach-In panel.
- Online teaching mode allows
  - Simulation of real time robot position in CIROS
  - Move the real robot in CIROS
  - Track the real robot coordinates in CIROS
- 1. Activate online teaching.
- 2. Observe the change in model window.
- 3. Move Roll coordinate of the robot incrementally 5°.
- 4. Deactivate online teaching.
- 5. Reset the model.

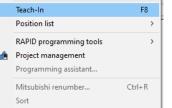
Teach-In	
Joint coordinates Cartesian coordin	nates
Move	Joint values
Waist 🔶 →	0.21 • -240.00 to +240.00
Shoulder 🔶 🔶	0.38 • -120.00 to +120.00
Elbow 🔶	89.94 • +0.00 to +160.00
Twist -	-50.17 • -200.00 to +200.00
Pitch 🔶	74.93 • -120.00 to +120.00
Roll 🔶 →	-0.76 • -360.00 to +360.00
Hold TCP pose	Apply
Speed override	Gripper output
10 %	HCLOSE1 ~ Close
Move incrementally	
25.00 mm 5.00 �	
Robot is connected.	Activate online teaching



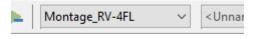
## Only CIROS Studio Online Teach-In

- 1. Make sure robot controller is online.
- 2. Open Teach-In panel.

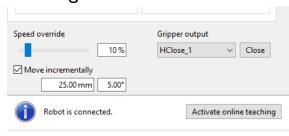




3. Select the robot as active controller.



4. At the bottom of Teach-In panel, click on Activate online teaching.



5. Click ok on pop-up window.



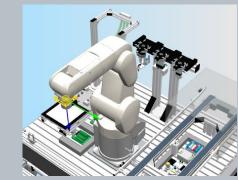


## Only CIROS Studio Online Teach-In

#### Notice the difference between before and after teach in is activated.

✓ Montage,RV         AD_Conv         AD_LONV         AD_CONV         AD_LONV         AD_	erter_88 erter_161 erter_161 erter_88 erter_161					
Properties						
Pose	Coordinate syst	m:		Set values	Actual	/alues
Dimension	Object	х: 🗘	Ŧ	335.00 mm	335	00 mm
Visualization	Only coorsy	у: 🗘	*	-0.00 mm	-0	00 mm
Extended		Z: 🔹	*	625.00 mm	625	00 mm
Info Measuring	Increments:	R: ‡	A V	180.00°		80.00°
Process geometry	100.00 mm	P: 🗘	<b>A</b>	-0.00°		-0.00°
	45.00°	Y: ‡	A	-180.00°		80.00°
				Арріу	U	

Joint coordinates Cartesian coordinates					
Move TCP pose					
X-axis 🗕 → し	x 335.00 mm R 143.50°				
Y-axis 🔶 🔶 🕑	y 0.00 mm P 0.00°				
Z-axis ← → し	z 625.00 mm Y -180.00°				
Translation Rotation	Apply				
Reference system Reference system					
Robot base $\sim$	Robot base $\sim$				
ТСР	Configuration				
<automatic tcp=""> $~~$ $~~$ R, A, N $~~~$</automatic>					



✓ Montage_RV-4         AD_Conve         AD_Conve         AD_Conve         AD_Conve         AD_Conve         AD_Conve         DA_Conve         DA_Conve         DA_Conve         DA_Conve         DA_Conve         JDA_Conve         JJDA_CONVE         JDA_CONVE         JDA_CONVE         JDA_CONVE         JJ1         JJ         JJ	rter_88 rter_88 rter_161 rter_88 rter_161					
Properties						×
General Pose	Coordinate sys	em:		Set values	Actua	values
Dimension	Object	/ X: 🔺	*	-393.56 mm	-39	.56 mm
Visualization	Only coors	у: 🗘	*	109.32 mm	10	.32 mm
Extended		Z: 🔺	*	523.70 mm	52	.70 mm
Info Measuring	Increments:	R: ‡	*	36.31°		36.31°
Process geometry	100.00 mm	P: 🗘	*	0.59°		0.59°
	45.00°	Y: 🗘	*	-179.69°		179.69°
				Apply	0	

Joint coordinates Cartesian coordin	ates
Move	TCP pose
X-axis 🔶 🔶 💍	x -393.56 mm R -0.19°
Y-axis 🔶 🔶 💍	y 109.32 mm P 0.665
Z-axis 🔶 🔶 💍	z 523.70 mm Y 179.90°
Translation Rotation	Apply
Reference system	Reference system
Robot base $\sim$	Robot base $\sim$
тср	Configuration
<automatic tcp=""> ~</automatic>	R, A, N 🗸





### Only CIROS Studio Get Actual Robot Data with Built-In Python Function

EXTRAS SETTINGS WINDOW HELP	Model	Name	Туре	RCI explorer	Name	Value 3
Fault simulation >	🙀 Objects	🛃 Montage_RV-4FL	Object	✓      Montage_RV-4FL	P_CURR	(-393.55, + 109.35, + 523.67, + 179.90, + 0.66, -0.19)(7, 15)
Model analysis >	✓ ▲ CP-F-RASS_Mitsubishi	PLC_Robot	Object	La Connection	SJ_CURR	(-195.58, +13.35, +92.43, +0.28, +73.62, -51.97)
Online management	> 🛃 BoxesBelt	🛃 BoxesBelt	Object	🖷 Robot type	SJ_ECURR	(-652805952.00,+55698888.00,+555280000.00,+786785.63,+177461536.00,-105352040.00)
	> 🛃 Fuses_1	🛃 MontagePosition	Object	Programs	<mark>₿J_FB</mark> C	(-195.58,+13.35,+92.43,+0.28,+73.62,-51.97)
Python > 🍨 Execute	> 🛃 Geometry	A Fuses_1	Object	E Slots	P_FBC	(-393.55, +109.35, +523.67, +179.90, +0.66, -0.19)(7, 15)
Video recorder > Execute again (screenshot.py) Ctrl+Shift+W	✓ A Montage_RV-4FL	🛃 Toolmagazin	Object	🐣 System variables	M_CMPDST	+0
Animation Designer Stop execution	> 🛃 AD_Converter_8Bit	and Tool-Platine	Object	Monitors	P_TOOL	(+0.00,+0.00,+0.00,+0.00,+36.50)(0,0)
	> 🛃 AD_Converter_8Bit	Tool-Workpiece	Object	Parameter	P_BASE	(+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
55 55	> 🛃 AD_Converter_16Bi	Tool-Fuse	Object	👼 Error list	P_NTOOL	(+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
CPC server ts tem variables	> 🛃 DA_Converter_8Bit	ToolChanger-Robot		V 🔀 Workplace	P_NBASE	(+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
CPC UA server	> 🛃 DA_Converter_16Bi	Ceometry	Object	Programs	M_HNDCQ	+0
, ,Jnitors	> 🗄 Inputs 🗸	Conn1	Section			+10
	< >	Conn2	Section		M_OVRD	+10 +100
	1 object selected					+100
	-					+100
C Open 2	Properties		×			+1
u open	General ^					+0
$\leftarrow \rightarrow \checkmark \uparrow$ $\square$ > This PC > Documents > CIROS > Projects > RASS2_717 > python $\checkmark$ $\circlearrowright$ $\land$	Pose		t values Actual values		M_RATIO	+100
	Dimension		0.00 mm 0.00 mm		M_RDST	+0
Organize 🔻 New folder	Visualization Onl	y coorsys y: 🔷 🗘 6	42.50 mm 642.50 mm		M RSPD	+0
TIA V17 🖈 ^ Name ^ Date modified Type Size	Extended	z; 🔶 🍝 8	45.00 mm 845.00 mm		& M_SPD	+10000
	Info				& M_NSPD	+10000
OneDrive Python File 2 KB	> Axes Increm	ents: R: 🔹 🗮	-90.00° -90.00°		M_ACL	+100
This PC	ORL 100.0	0 mm ≑ P: ≑ ≑	-0.00° -0.00°		M_DACL	+100
	Due allows				M_NACL	+100
	Collision detectior 🗸	45.00° 🜩 Y: 🔹 🔹	0.00° 0.00°		& M_NDACL	+100
	< >		Annly 0 🔚 🔳	86 object(s)	,	
	Messages			• • •		
	TCP pose cartesian = [-39	2 56200220022464	100 22270742200060	522 7016515080051		
	TCP pose cartesian - [-39 TCP pose rotational = [0.				8, 180.115463	027389041
						53200002, -0.907047612261]
	Forward kinematic = (Circ					
- · · · · · · · ·			93.56 ]			
<b>Ready to use project:</b> robotInfo.py						
ready to dod projecti reportinorpy	-		00 ]			
	, 0) Kinematic class = 3					
Video tutorial: 62_RobotInfoPython.mp4	Speed parameter = (5500.0	, 20000.0, 0.0, 0	.0)			



### **Virtual Reality**

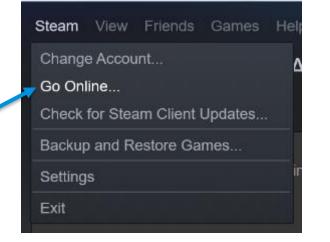


### Setting up VR Glasses

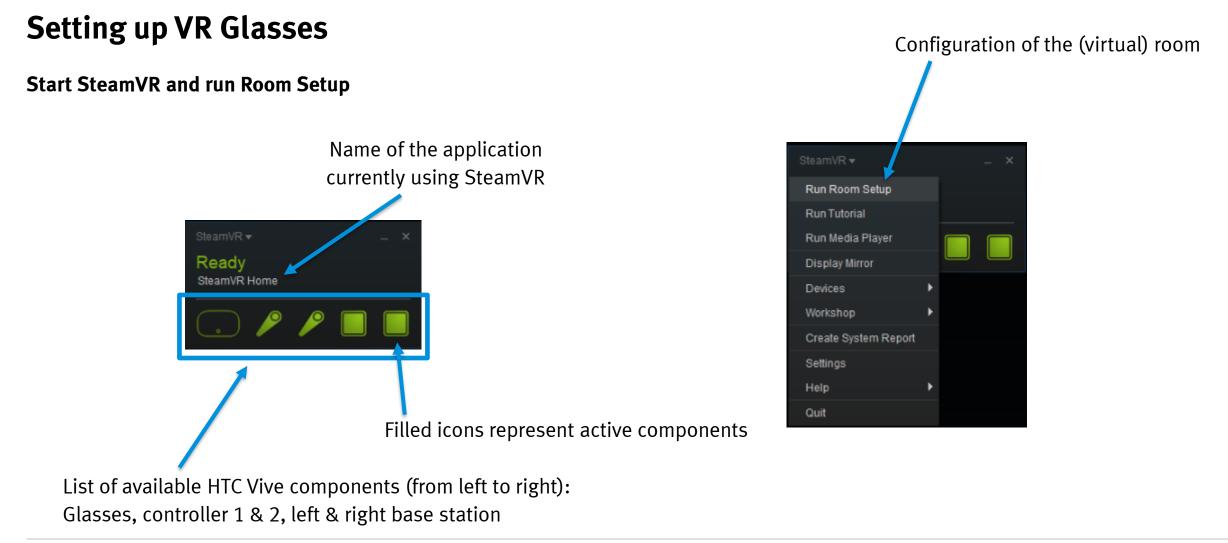
- CIROS can be directly connected to VR glasses like HTC Vive and Oculus Rift.
- Typical procedure
  - 1. Install Steam / SteamVR and create a user account (https://store.steampowered.com/)
  - 2. Start Steam and switch to "offline mode" in case that Steam should be used without internet access
  - 3. Start SteamVR and run room setup
  - 4. Configure the VR mode of CIROS
  - 5. Activate VR mode of CIROS and start the simulation

**Note:** HTC Vive is the only officially supported VR glasses.

Switching between Steam's online/offline mode 🗲









#### Start SteamVR and run Room Setup

Choose "Room-Scale" to be able to go around in the real and virtual room

#### Welcome to Room Setup!

#### Set up for Room-Scale

Play Room-Scale, Standing, and Seated VR experiences. Choose this if you have at least 2 meters by 1.5 meters, or around 6.5 by 5 feet.

#### Set up for Standing Only

Play Standing and Seated VR experiences. Choose this if you have limited space to walk around.



ROOM-SCALE

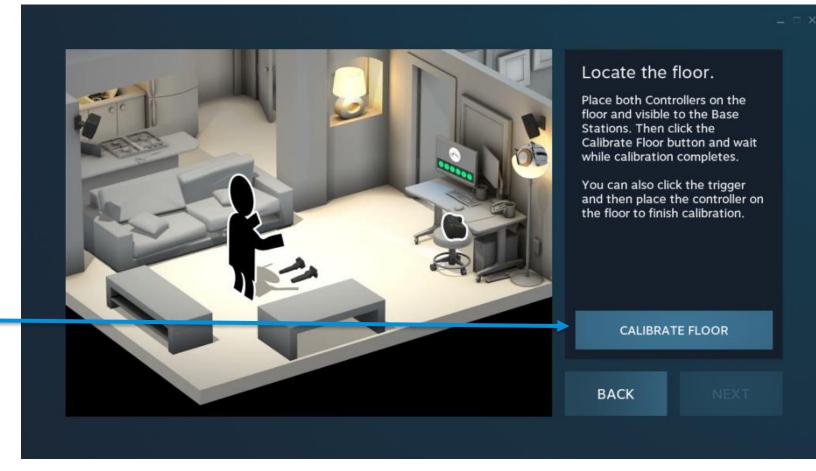


STANDING ONLY



Start SteamVR and run Room Setup

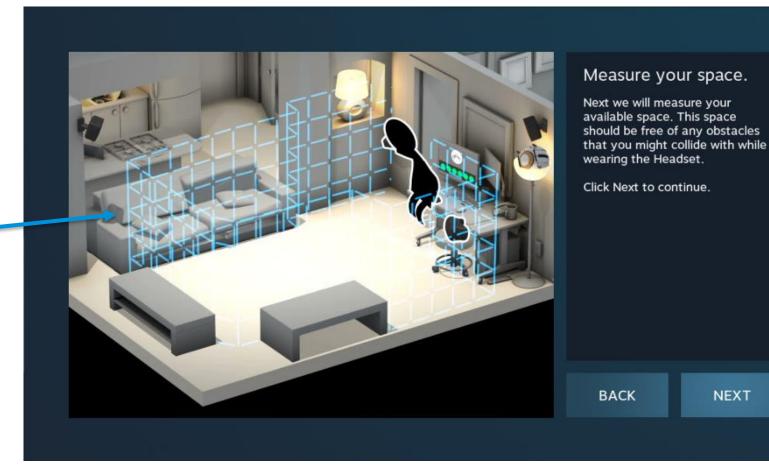
Calibrate real and virtual floor (ground level wrt. to z-axis)





Start SteamVR and run Room Setup

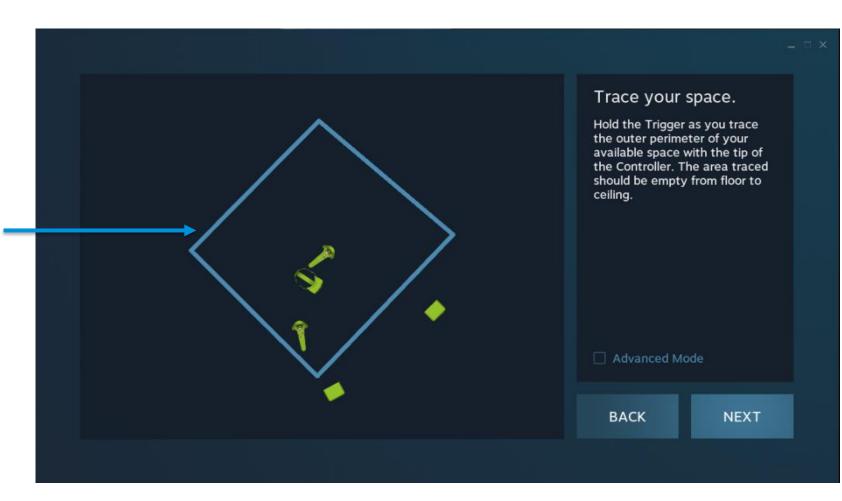
Configuration of the available physical space to be matched with the virtual room





Start SteamVR and run Room Setup

Specify the available physical space (its outer perimeter) by holding the trigger button of one of the two controllers



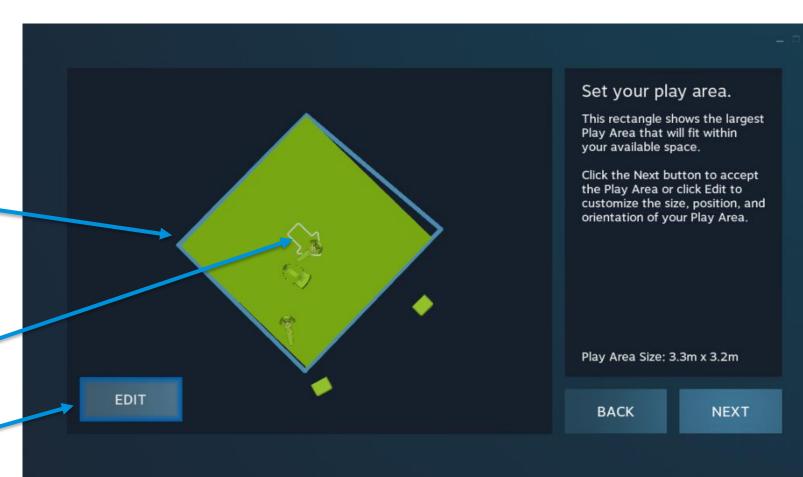


#### Start SteamVR and run Room Setup

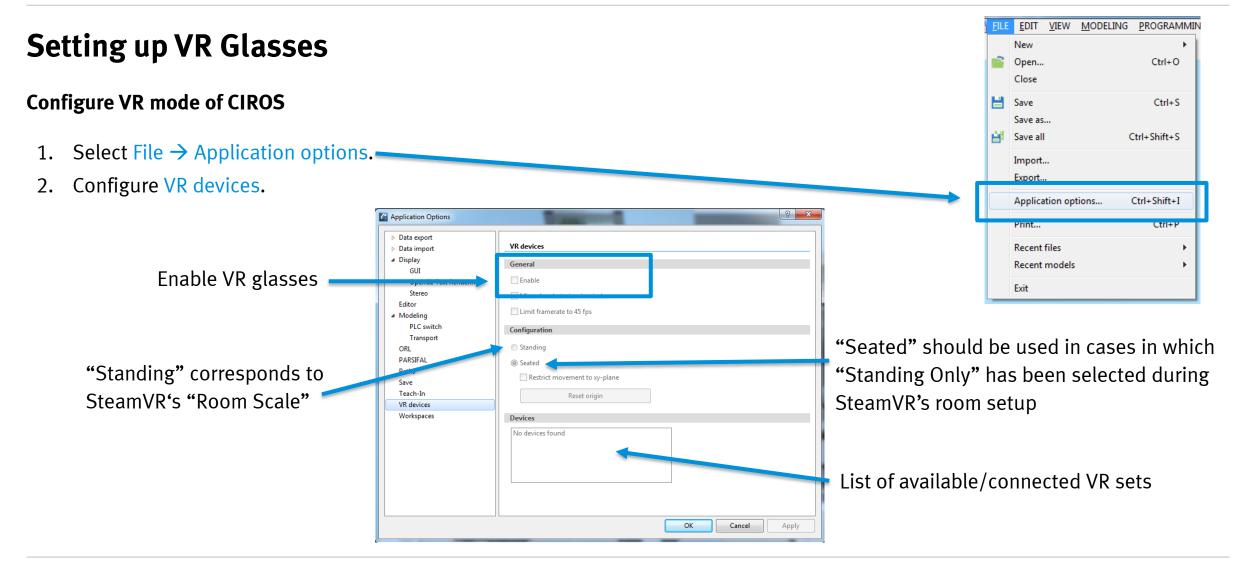
If the available space defined before fits the "Play Area" it will be highlighted in _____ green

Direction of the user perspective onto the

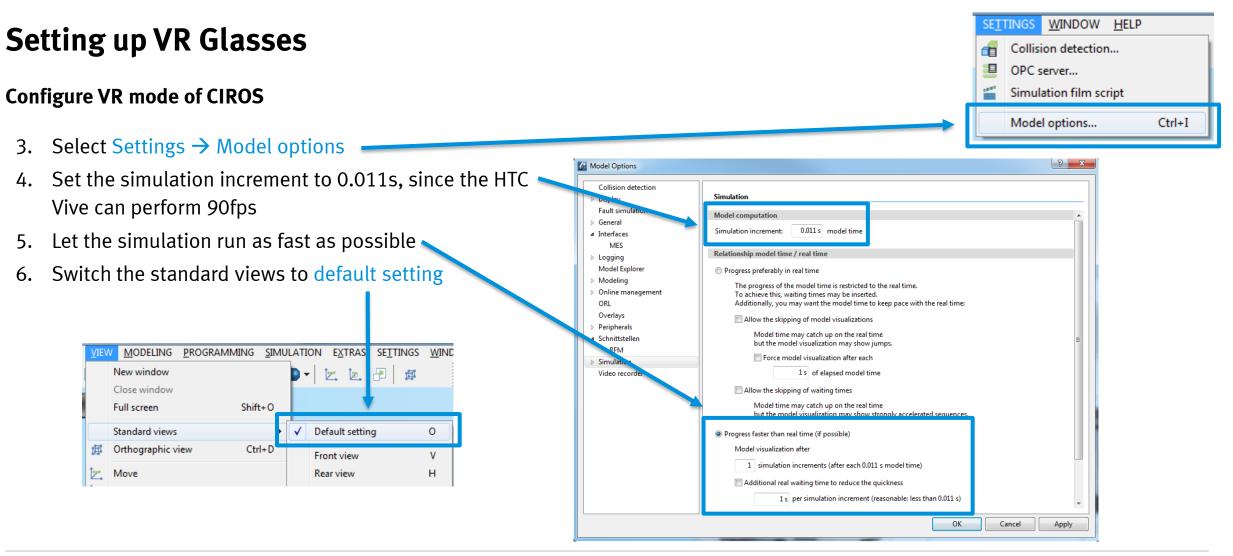
"EDIT" allows for optimizing position, " size, and user perspective manually







#### FESTO

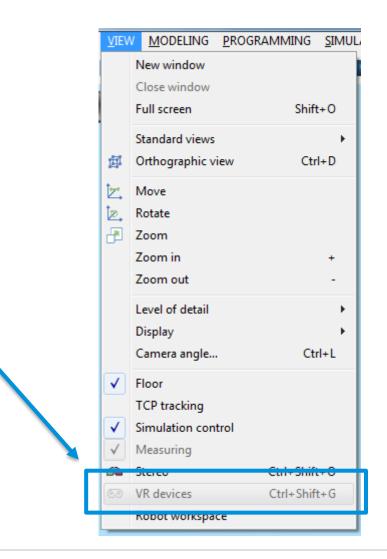


#### FESTO

### Setting up VR Glasses

Activate VR mode of CIROS and start the simulation

- Enable the connected VR device and start the simulation
- Remark: It is highly recommended to disable all shadow calculations to have enough computational power for a "smooth" visualization!

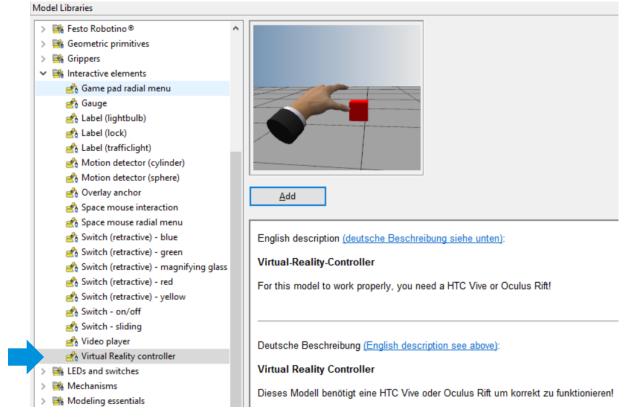




### **Interact with Model**

- To interact with the model, for example, click on the buttons, start and stop simulation, etc. An interactive element has to be added.
- 1. Go to "Modelling  $\rightarrow$  Model libraries".
- 2. Expand "Interactive elements".
- 3. Add "Virtual Reality controller".

• <u>Note:</u> "Simulation Control" has to be shown in "Model Window" to be able to control simulation in VR.





# Advanced



### **Move I/O Address**

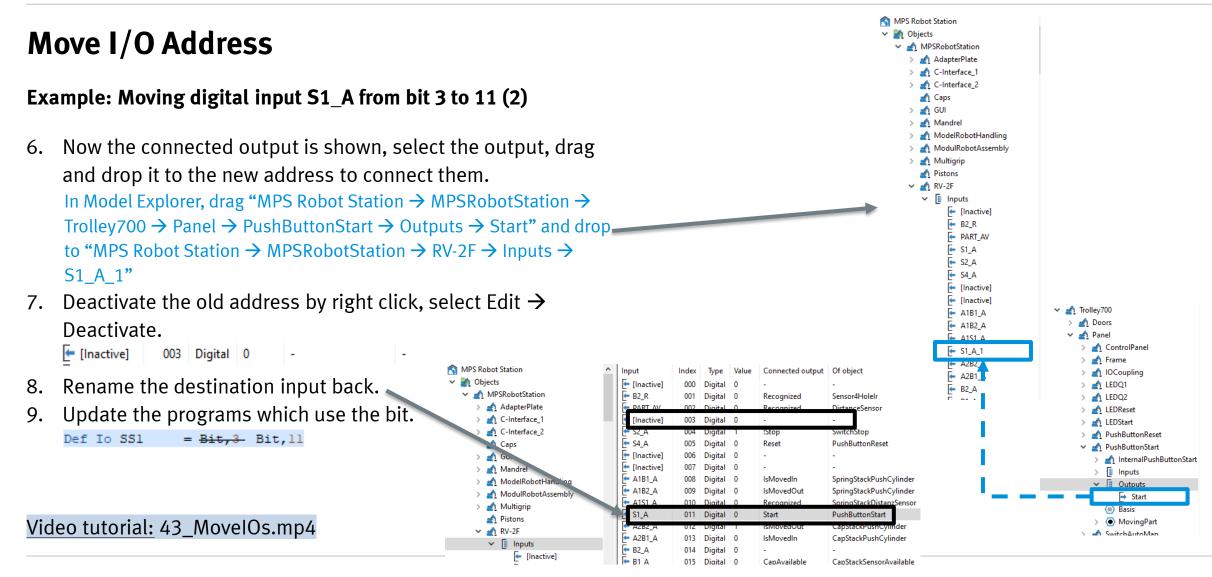
#### Example: Moving digital input S1_A from bit 3 to 11 (1)

- 1. Open "Model Explorer".
- 2. Go to the input to be moved. MPS Robot Station  $\rightarrow$  Objects  $\rightarrow$  MPSRobotStation  $\rightarrow$  RV-2F  $\rightarrow$  Inputs
- 3. Right click on destination address, and select "Rename".
- Assign a name, note that names of the cannot repeat in the whole model, add an extension such as "_1" at the end if needed.
   \$1_A_1
   011
   0
   -
- 5. If the input is connected to an output, it has to be reconnected. Double click on original address to open it, right click on "Connected output" and select "Follow connection".

	>	AssemblySocket	^	Property	Value			
	>	🛃 BoxStack_1	10	F Input	S1_A			
	>	🛃 BoxStack_2		- Index	003			
	>	🛃 Chute		🔄 Type	Digital			
	>	🔬 Mini-IO-Terminal_1		🔄 Value	0			
	>	🛃 SwitchesForNewParts		📑 Connected output	Start		Follow connection	
>		ModulRobotAssembly		Of object	PushButton		ronow connection	
>		Multigrip		🖶 Object path	MPSRobotSt		Remove connection	
		Pistons					Refresh	F6
$\sim$		RV-2F				_		
	$\sim$	Inputs				۶	Properties	
		[mactive]				_		
		듣 B2_R						
		PART_AV						
		두 S1_A						

lodel Explorer						
MPS Robot Station	Input	Index	Туре	Value	Connected output	Of object
🗸 🋃 Objects	[Inactive]	000	Digital	0	-	-
🗸 🔬 MPSRobotStation	- B2_R	001	Digital	0	Recognized	Sensor4HoleInBottom
> 🛃 AdapterPlate	- PART_AV	002	Digital	0	Recognized	DistanceSensor
> 🛃 C-Interface_1	← S1_A	003	Digital	0	Start	PushButtonStart
> 🛃 C-Interface_2	E SZ_A	004	Digital	1	!Stop	SwitchStop
🛃 Caps	🗲 S4_A	005	Digital	0	Reset	PushButtonReset
> 🛃 GUI	[mactive]	006	Digital	0	-	-
> 🛃 Mandrel	[mactive]	007	Digital	0	-	-
> 🖌 ModelRobotHandling	두 A1B1_A	008	Digital	0	lsMovedIn	SpringStackPushCylin
> A ModulRobotAssembly	두 A1B2_A	009	Digital	0	IsMovedOut	SpringStackPushCylin
> 🔬 Multigrip	두 A1S1_A	010	Digital	0	Recognized	SpringStackDistanzSer
Pistons	🗲 [Inactive]	011	Digital	0	-	-
V 🛃 RV-2F	[← A2B2_A	012	Digital	1	lsMovedOut	CapStackPushCylinde
> E Inputs	- A2B1_A	013	Digital	0	lsMovedIn	CapStackPushCylinde
> Outputs	🗲 B2_A	014	Digital	0	-	-
	🗲 B1_A	015	Digital	0	CapAvailable	CapStackSensorAvaila
Base	[mactive]	016	Digital	0	-	-
Waist	[mactive]	017	Digital	0	-	-
Shoulder	[mactive]	018	Digital	0	-	-
Elbow	[mactive]	019	Digital	0	-	-
<ul> <li>Twist</li> </ul>	[ [Inactive]	020	Digital	0	-	-
Pitch	[mactive]	021	Digital	0	-	-
> 💿 Roll	[ [Inactive]	022	Digital	0	-	-
SafetyHousing	두 [Inactive]	023	Digital	0	-	-

#### FESTO

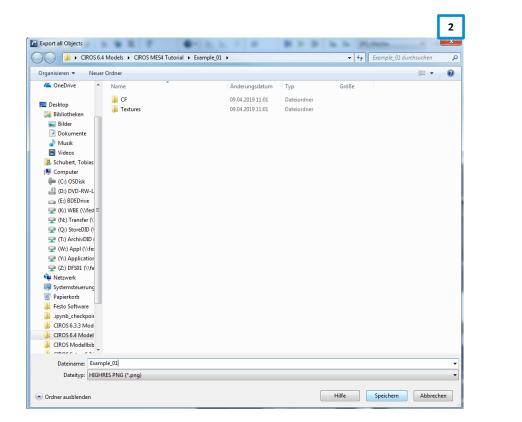




### **Export as High-Resolution Images**

• File  $\rightarrow$  Export enables the user to save the content of the so-called "first" view window as a high-resolution PNG file.

				1
FILE	<u>E</u> DIT	<u>V</u> IEW	MODELING	<u>P</u> ROGRAMMI
	New			+
6	Open			Ctrl+O
	Close			
Н	Save			Ctrl+S
	Save as			
÷.	Save all		C	Ctrl+Shift+S
	Import.			
	Export	•		
	Applica	tion o _l	ns	Ctrl+Shift+I
	Print		•	Ctrl+P
	Recent	files		+
	Recent	models		•
	Exit			

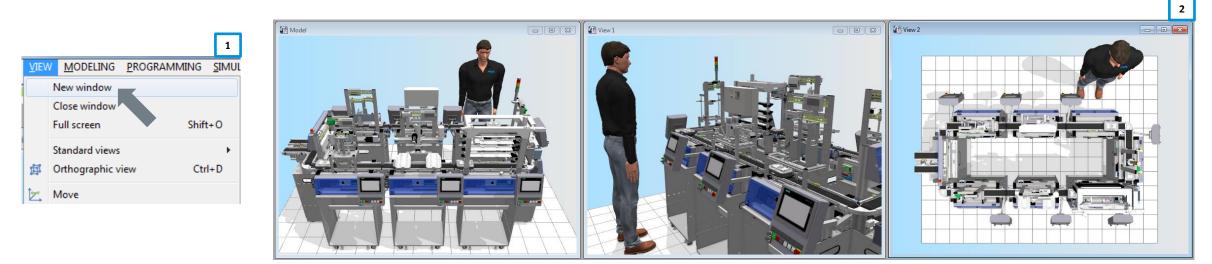


1	3 High-Resolution Export	
	High-Resolution Export	
	Width 2500 Height 1274 Pixels	
	Mask background	
	OK Cancel	
	OK Cancel	_
		4



### **Multiple View Windows**

- CIROS is not restricted to a single view window but offers the possibility to show a couple of view windows concurrently.
- To open a new window, select View → New window.
- Each view might have a different perspective onto the scene.



First / main view window

Additional view windows



### **CIROS Starter**

#### Calling CIROS model with URL

Possible action	Parameter	Explanation
Start simulation	StartSimulation	Starts the simulation.
Activate full screen	ActivateFullScreen	Runs CIROS in in full screen mode.
Use single instance	UseSingleInstance	CIROS uses an existing instance to open the new model. A new instance is only started if no instance of CIROS is running yet.

URL format: "StartCIROS: <path to model>?Parameter1&Parameter2"

Example:

"StartCIROS:Example Models\Small Demos for Learning\Robot Assembly with UR5\Robot Assembly with UR5.modx?ActivateFullScreen&UseSingleInstance"



## Model Analysis

#### Extras $\rightarrow$ Model analysis $\rightarrow$ Whole model

• Help in problem diagnosis and finding a solution

Mod	Modellanalyse										
	Beschreibung Objekt "Conveyor", Gruppe "CosTrans", Segment "Seg001": Die Gravitatio Objekt "Conveyor_1", Gruppe "CosTrans", Segment "Seg001": Die Gravitat Objekt "Conveyor_2", Gruppe "CosTrans", Segment "Seg001": Die Gravitat	Quelle Transportsimulation Transportsimulation Transportsimulation Transportsimulation									
	Träger-Objekt "Carrier_1": Der Träger hat keine sichtbare Geometrie. Träger-Objekt "Carrier_2": Der Träger hat keine sichtbare Geometrie.										
		Hilfe	F1								

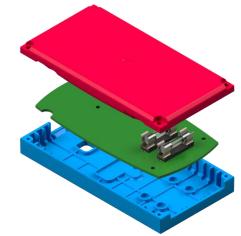
Note: Picture in German.



### **CIROS Part Number for CP System**

"PartNrReplikator.py" replicates the part in simulation. "PartNrTranslation.py" translates MES4 part number to CIROS part number.

Digit 6 Special parts		Digit 5 – 3 Parts		Digit 2 Fuses		Digit 1 Front cover	r colours	Digit 0 Back cover	colours
Pallet	1	Unprocessed front cover	1	Front fuse	1	Black	0	Black	0
Front cover, raw block	2	Front cover	2	Rear fuse	2	Grey	1	Grey	1
Back cover, not pressed on front cover	3	PCB	4			Blue	2	Blue	2
Turn part	4	Back cover	8			Red	3	Red	3
Boxes	5	Label on front cover	16			Not used	4	Not used	4
Not used	6	Label on back cover	32			Not used	5	Not used	5
Not used	7	Not used	64			Not used	6	Not used	6
Not used	8	Not used	128			Not used	7	Not used	7
Not used	9	Not used	256			Not used	8	Not used	8
Example: see picture	0	2+4+8		1+2		2	2		B = C



= 0014323 = **14323** 



#### Task: Add a MPS Part from CP-L-SOURCE

- There are some parts supported from CP-System model libraries. It is possible to create own part to work in CP environment in CIROS.
- User can import the CAD drawing of their parts and test it in virtual environment before implementing it on the real system.
- Steps to create own part will be explained with an example of adding a Festo MPS red housing with RFID and black MPS cap to a CP-L-CONVEYOR via a CP-L-SOURCE.





### MPS part numbers according to FactoryViews^[1]

Part number	Part	Image	Part number	Part	Image
3001	Red MPS housing with RFID		3020	Black cap	
3002	Black MPS housing with RFID		3080	Cap with integrated micro- controller	
3003	Silver MPS housing with RFID				

[1] FactoryViews is the new software bundle for MES and web based services. It contains MES4 v3.



#### Steps to create an own part (1)

- 1. Prepare CAD drawing of the parts in supported format.  $(\rightarrow)$
- 2. Define MES4 / FactoryViews part numbers. (→)
- 3. Define CIROS part numbers. (→)
- 4. In CIROS, create a new project and import the CAD drawing.  $(\rightarrow)$
- 5. If needed, optimize the geometry of the CAD drawing to reduce its size and rendering requirement in simulation.  $(\rightarrow)$
- 6. Rename the object and configure its properties, such as object type, gripper points and grip points.  $(\rightarrow)$   $(\rightarrow)$
- 8. Move the configured object to Templates by drag and drop.  $(\rightarrow)$
- 9. Insert following CP-System Models from Model Libraries and snap them to place.
  - CP-L-CONVEYOR, CP-L-SOURCE and CP-L-SINK
- 10. Modify following files to add new part numbers. <project>\CF\py\PartNrTranslator.py (→) and <project>\CF\py\PartNrReplikator.py (→)
- 11. Modify template "Palette" to allow it to take along new parts during transportation.  $(\rightarrow)$
- 12. Add following objects to "Objects  $\rightarrow$  Werkstuecke". These are the meta objects.
  - Housing, Pistons and Caps
- 13. Save all, close CIROS and restart CIROS. Then, open the project and test the new parts added.



#### **Prepare CAD drawing of the parts in supported format.**

- CIROS supported formats are as follow:
  - 3ds Max
  - AutoCAD DXF
  - Autodesk
  - Blender
  - Collada
  - IGES
  - PointCloud
  - STEP
  - STL
  - VRML
  - Wavefront Object



### **Define MES4 / FactoryViews part numbers.**

Part number	Part
1	MPS housing red
2	MPS housing black
3	MPS housing silver
3001	Red MPS housing with piston containing RFID chip
3002	Black MPS housing with piston containing RFID chip
3003	Silver MPS housing with piston containing RFID chip
3020	MPS cap
13001	Red MPS housing + piston + cap
13002	Black MPS housing + piston + cap
13003	Silver MPS housing + piston + cap



#### CIROS part number as defined in "PartNrTranslation.py"

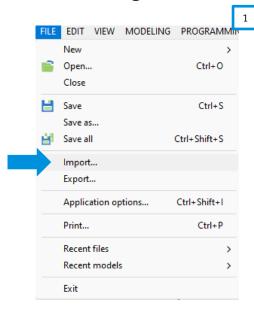
Digit 6 Special parts		Digit 5 – 3 Parts		Digit 2 Fuses		Digit 1 Front cover / housing colours		Digit 0 Back cover/cap colours	
Pallet	1	Unprocessed front cover	1	Front fuse	1	Black	0	Black	0
Front cover, raw block	2	Front cover	2	Rear fuse	2	Grey	1	Grey	1
Back cover, not pressed on front cover	3	РСВ	4			Blue	2	Blue	2
Turn part	4	Back cover	8			Red	3	Red	3
Boxes	5	Label on front cover	16			Not used	4	Not used	4
Not used	6	Label on back cover	32			Not used	5	Not used	5
Not used	7	MPS housing	64			Not used	6	Not used	6
Not used	8	MPS piston	128			Not used	7	Not used	7
Not used	9	MPS cap	256			Not used	8	Not used	8
Example: see picture	0	64+128+256		1 1 1 1 1 1 1 1 1	0		3		0

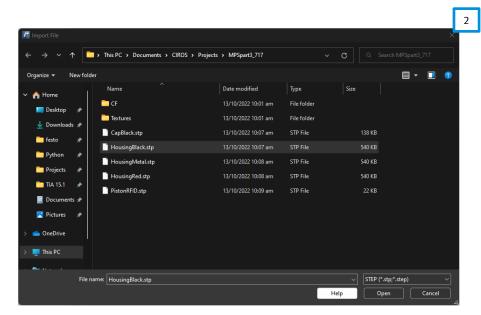


= 0 448 0 3 0 = **448030** 



#### Import CAD drawing





AD Import with Open(	_ascade			>
Source system				
Unit of length:	mm (millimeter)	~	1 mm	
Coordinate system:	Y-axis points up			
	Z-axis points up			
Additional options				
Object hierarchy:	Deep $\checkmark$	🗌 Import lin	es	
Surface color:	-			
Line color:	<b>•</b>			
Deflection:	1.0000			



#### **Optimize geometry**

- 1. Select the imported CAD object in Model Explorer.
- 2. Click Modeling  $\rightarrow$  Geometry optimization^[1].
- 3. Run the required optimizations.
- In the example, following optimizations were carried out.
  - Clean up geometry ^[2]
  - Merge hulls with matching material or area color^[3]
  - Simplify object hierarchy [4]

	MOE	PROGRAMMING	SIMULATION	EXTRAS	SE		Geor	metry o
	20	Model Explorer		Ctrl+T			+	Find p
		Edit mode Model libraries	Ctrl+	Ctrl+E Shift+M			÷	Fill ho
		Properties	A	Alt+Enter				Repla
	ß	Transport element snappi	ng off					Select
	8	Object functions off					00	Close
	4	PLC Switch						
		I/O configuration			>		-	Delete
ſ	1	3D marker Geometry optimization			>		<b>G</b> ≹	Delete
L	1	Path generation			>		•	Reduc
		Renderclone			>		-	Simpl
							-	Merg
						2	~	Clean
							-	Сору
							9	Optin
							+	Fit co
							•	Chan
						3	æ	Merg

G	eor	metry optimization	>
1	÷	Find problems	
4	r.	Fill holes on the surface	1
4	P	Replace by enveloping body	
Ę		Select identical elements	<b>▲</b> ⊙
-	~	Close drill holes	
-	•	Delete inside of primitive body	<b>≜</b> ⊙
Ģ	æ	Delete contained geometries	<b>1</b> 0
6	D	Reduce polygons	
,	-	Simplify rounded edges	
3	-	Merge vertices	11
	1	Clean up geometry	
2		Copy structure into elements	<b>A</b> O
6	3	Optimize for rendering	<u>1</u>
4	¥	Fit coordinate systems to hulls	
	D	Change cylinder approximation	
:	ľ	Merge hulls with matching material or area color	<b>▲</b> ⊙
	1	Create render clones for similar objects	<u>1</u>
<	D	Edit facets	
Ģ	¥	Simplify object hierarchy	1



#### **Objects' properties defined in example.**

Part name	Meta object	Object type	Gripper points	Grip points
MPS_Housing_rt ^[1]	Housing	WS	GripPistonRed GripCapRed	HousingRed
MPS_Housing_sw ^[2]	Housing	WS	GripPistonBlack GripCapBlack	HousingBlack
MPS_Housing_gr ^[3]	Housing	WS	GripPistonSilver GripCapSilver	HousingSilver
MPS_Piston	Pistons	WS		PPiston
MPS_Cap	Caps	WS		CapStackBlackCap



#### **Configure object's properties in CIROS (1)**

- Rename the object in Model Explorer. 1.
  - 🕎 Model 🗸 🏹 Objects > MPS_Housing_sw Templates
- Configure object type. 2.
  - In Properties, select General  $\rightarrow$  Object type. 1.
  - In Change field, type in 'WS'. 2.

General		_
Pose	Arameters for 'MPS_Housing_sw'	.
Dimension	🌮 Object type: <ws></ws>	
Visualization	K Control: < <default>&gt;</default>	
Extended	sa 28. ( and at also it	
Info	Change: WS	$\sim$

- Add gripper points. 3.
  - In Model Explorer, right click on MPS_Housing_sw  $\rightarrow$  Base and select New 1.

#### $\rightarrow$ Gripper point.

	🛃 MPS_Hou	sing_sw	Polyhedron 1	Hull		
	> > Base Templates	New		>	Ь	Grip point
ł	Materials	Edit		>	L.	Gripper point
ī	Paths					Geometric prin

2. Rename the gripper points.



4. Configure gripper points.

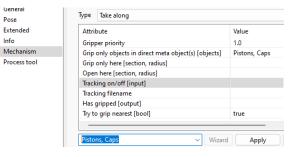
Pose

Info

- Select gripper point. In Properties  $\rightarrow$  Mechanism, select Type as Take along. 1.
- Set value of Grip only objects in direct meta object(s) to the object laying in 2. or on top of the part. Click Apply.

Example: Black MPS Housing, the value is Pistons, Caps. Because these parts lay in and on top of the housing.

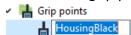
Hint: The value can be written directly in the field below.





#### **Configure object's properties in CIROS (2)**

- 5. Add grip point.
  - 1. Right click on Base, select New → Grip point. MPS_Housing_sw Folder Comparison of the point of the poin
  - 2. Rename the grip point.

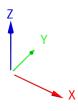


#### FESTO

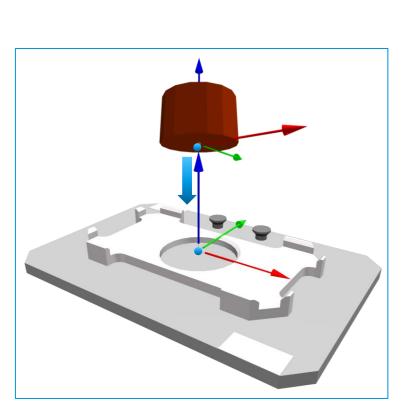
### **Create Own Part**

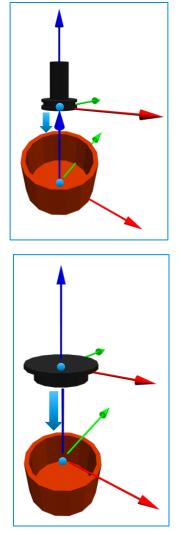
### Define coordinates of gripper points and grip points.

- The x, y, z-coordinates of the parts have to match, so that the parts replicated can be snapped in the right place.
- X, y, z-coordinates in CIROS are highlighted in different colours.



 Coordinates of gripper and grip points can be moved in Properties → Pose.

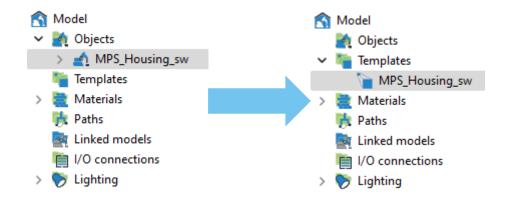






#### Move configured object to Templates.

• Objects can be moved between 'Objects' and 'Templates' in Model Explorer by drag and drop.





#### "PartNrTranslation.py" description

• "PartNrTranslation.py" translate MES4 part number to CIROS part number.

#### • 2D array "PN_by_color" sorts the part by colour.

<pre># MES Part number sorted by col PN by color = [</pre>	our [black, grey, blue, red]
[10, 0, 0, 0],	#cwlinder
	-
	#palette
[90, 91, 92, 93],	
[101, 102, 103, 104],	#raw block
[110, 109, 108, 107],	
[111, 112, 113, 114],	#back cover pressed
[120, 0, 0, 0],	#PCB
[121, 0, 0, 0],	#broken, do not use
[122, 0, 0, 0],	#broken, do not use
[123, 0, 0, 0],	#broken, do not use
[210, 310, 410, 510],	#front cover
[211, 311, 411, 511],	#front cover with PCB
[212, 312, 412, 512],	#front cover with PCB and front fuse
[213, 313, 413, 513],	#front cover with PCB and rear fuse
[214, 314, 414, 514],	#front cover with PCB and both fuses
[1210, 1310, 1410, 1510],	<pre>#pressed front and back cover</pre>
[1211, 1311, 1411, 1511],	<pre>#pressed front and back cover with PCB</pre>
[1212, 1312, 1412, 1512],	<pre>#pressed front and back cover with PCB and front fuse</pre>
[1213, 1313, 1413, 1513],	<pre>#pressed front and back cover with PCB and rear fuse</pre>
[1214, 1314, 1414, 1514],	<pre>#pressed front and back cover with PCB and both fuses</pre>
1	

• List "CirosPN_by_PNsw" translates MES part number to CIROS part number. Only part number of black part is needed here.

# Translate MES part number of black parts to CIROS part number

CirosPN by PNsw = { 10: 4000000, 25: 1000000, 90: 3000000, 101: 2000000, 110: 1000, 111: 8000, 120: 4000, 121: 4100, 122: 4200, 123: 4300, 210: 2000, 211: 6000, 212: 6100, 213: 6200, 214: 6300, 1210: 10000 1211: 14000, 1212: 14100, 1213: 14200, 1214: 14300 ъ

• <u>Attention:</u> Part row of "PN_by_color" must match with "CirosPN_by_PNsw" for the script to find the correct part!



#### Add new parts to "<project>\CF\py\PartNrTranslation.py"

#### 1. Add MES part numbers to "PN_by_color".

# MES Part number sorted by colour [black, grey, blue, red]
PN by color = [

N_by_color = [	
[2, 3, 0, 1],	''' NEW! ''' #MPS housing
[10, 0, 0, 0],	#cylinder
[25, 0, 0, 0],	#palette
[90, 91, 92, 93],	#back cover unpressed
[101, 102, 103, 104],	#raw block
[110, 109, 108, 107],	#undrilled front cover
[111, 112, 113, 114],	#back cover pressed
[120, 0, 0, 0],	#PCB
[121, 0, 0, 0],	#broken, do not use
[122, 0, 0, 0],	#broken, do not use
[123, 0, 0, 0],	#broken, do not use
[210, 310, 410, 510],	#front cover
[211, 311, 411, 511],	#front cover with PCB
[212, 312, 412, 512],	#front cover with PCB and front fuse
[213, 313, 413, 513],	#front cover with PCB and rear fuse
[214, 314, 414, 514],	#front cover with PCB and both fuses
[1210, 1310, 1410, 1510],	<pre>#pressed front and back cover</pre>
[1211, 1311, 1411, 1511],	<pre>#pressed front and back cover with PCB</pre>
[1212, 1312, 1412, 1512],	<pre>#pressed front and back cover with PCB and front fuse</pre>
[1213, 1313, 1413, 1513],	<pre>#pressed front and back cover with PCB and rear fuse</pre>
[1214, 1314, 1414, 1514],	<pre>#pressed front and back cover with PCB and both fuses</pre>
[3002, 3003, 0, 3001],	''' NEW! ''' #MPS housing with RFID piston
[13002, 13003, 0, 13001],	''' NEW! ''' #MPS housing with piston and cap
]	

#### 2. Add part numbers to "CirosPN_by_PNsw".

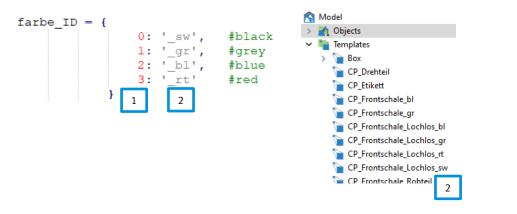
10: 4000000, 25: 1000000, 90: 3000000, 101: 2000000, 110: 1000, 111: 8000, 120: 4000, 121: 4100, 122: 4200, 123: 4300, 210: 2000, 211: 6000, 212: 6100, 213: 6200, 214: 6300, 1210: 10000, 1211: 14000, 1212: 14100, 1213: 14200, 1214: 14300, 3002: 192000, VI NEW! 13002: 448000 ''' NEW! ''' ł

 Note that the part rows of "PN_by_color" matches "CirosPN_by_PNsw".



#### "PartNrReplikator.py" description

- "PartNrReplikator.py" takes the translated CIROS part number and replicates the part in the model from templates.
- The script also configures gripper points and grip points.
- List "farbe_ID" matches colour ID^[1] in "PartNrTranslator.py" with colour add on at the end of templates name^[2].



• "bauteil_ID" contains detailed information of each CIROS ID.

# #		(Template name without colour addition	, Meta object	,GPP-Name in case none is given	,grippe which		
bauteil ID = {							
	1:	('CP_Frontschale_Lochlos'	, 'Frontschalen'	,'Bauchlage'	, []	),	#undrilled front cover
	2:	('CP Frontschale'	, 'Frontschalen'	,'Bauchlage'	<i>,</i> []	),	#front cover
	4:	('CP Platine'	'Platinen'	,'GPP Platine'	,[1,2]	),	#PCB
	8:	('CP Rueckschale'	, 'Rueckschalen'	,'GPP Rueckschale'	,[1,2]	),	#Back cover
	16:	('CP Etikett'	, 'Etiketten'	'GPP Etikett'	,[1,2]	),	#Label, sticked on front cove
	32:	('CP Etikett'	'Etiketten'	,'GPP Etikett'	,[8]	),	#Label, sticked on back cover
	64:	$('', \bar{'}', \bar{'}', 0)$					
	128:	('', '', ''', 0),					
	256:	('', '', '', 0),					
	512:	('CP Schmelzsicherung'	, 'Sicherungen'	,'GPP SicherungA'	, [4]	).	#Front fuse
		('CP Schmelzsicherung'	'Sicherungen'	'GPP SicherungB'	,[4]	5.	#Rear fuse

• "sonderteil_ID" contains detailed information of CIROS ID which are declared as special parts.

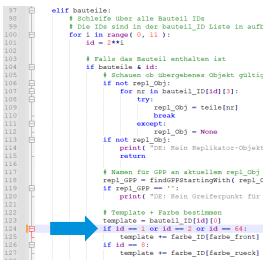
sondertell_	$TD = \{$						
	1:	('Palette'	, 'Paletten'	,'Bauchlage'	,[]	),	<pre>#empty palette</pre>
	2:	('CP_Frontschale_Rohteil'	, 'Frontschalen'	,'Bauchlage'	, []	),	#raw block
	3:	('CP_Rueckschale'	, 'Rueckschalen'	,'Rueckschale_aufgelegt_Palette	e',[]	),	<pre>#back cover, unpressed</pre>
	4 :	('CP_Drehteil'	, 'Drehteile'	,'Bauchlage'	, []	),	#cylinder
	5:	('Box'	, 'Boxen'	,'Box'	, []	),	#box
	}						



#### Add new parts to "<project>\CF\py\PartNrReplikator.py"

#### Add new parts to list "bauteil_ID". 1. bauteil_ID = { 1: ('CP Frontschale_Lochlos' , 'Frontschalen' ,'Bauchlage' ,[] ), #undrilled front cover ), #front cover 2: ('CP Frontschale' , 'Frontschalen' ,'Bauchlage' ,[] 4: ('CP Platine' , 'Platinen' ,'GPP Platine' ,[1,2] ), #PCB 8: ('CP Rueckschale' , 'Rueckschalen' ,'GPP Rueckschale' #Back cover ,[1,2]), 16: ('CP_Etikett' ,'GPP Etikett' , 'Etiketten' ), #Label, sticked on front cover ,[1,2] , 'Etiketten' ,'GPP Etikett' 32: ('CP Etikett' ), #Label, sticked on back cover ,[8] 64: ('MPS Housing , 'Housing' , [] ), #MPS housing , 'Pistons' ,'GripPiston' 128: ('MPS Piston' ,[64] #MPS piston with RFID chip ), , 'Caps' 256: ('MPS Cap' ,'GripCap' ,[64] ), #MPS cap ,'GPP SicherungA' 512: ('CP Schmelzsicherung' , 'Sicherungen' , [4] ), #Front fuse ), #Rear fuse 1024: ('CP Schmelzsicherung' , 'Sicherungen' ,'GPP SicherungB' , [4]

In function "replicatePartAt", under "elif bauteile:", add "or id == 64", to allow colour allocation of MPS housings. At default v7.1 script it is at line 123.





#### Modify template "Palette"

1.	Move	"Palette"	from	"Temp	olates"	to	"Objects"	,
----	------	-----------	------	-------	---------	----	-----------	---

😭 Model

- Objects
   CP-L-CONVEYOR
- > A CP-L-SINK
- > 🔬 CP-L-SOURCE
- > 🛃 Palette
- 2. Select "Palette  $\rightarrow$  _347490-d_Palette_110x164mm  $\rightarrow$  Gripper points  $\rightarrow$  Bauchlage_Palette".

🗸 🔬 Palette

- _014-105a_iCIM-Aufnahmebolzen
- _014-105a_iCIM-Aufnahmebolzen_2
- - 🗸 🕌 Gripper points

🛓 Bauchlage_Palette

- La Rueckschale_aufgelegt_Palette
- 🛓 Rueckenlage_Palette
- 🛓 Komplett_Rueckenlage_Palette
- s 🖬 🖓 🖬

3. In Properties window, change the value of "Grip only objects in direct meta object" to "Frontschalen, Ruechschalen, Housing" and click Apply.

**Note:** The value can be typed directly in the field below.

General Pose	Type Take along	
Extended	Attribute	Value
Info	Gripper priority	0.1
Mechanism	Grip only objects in direct meta object(s) [objects]	Frontschalen, Rueckschalen, Housin
Process tool	Grip only here [section, radius]	
	Open here [section, radius]	
	Tracking on/off [input]	
	Tracking filename	
	Has gripped [output]	
	Try to grip nearest [bool]	true
	Smooth grip correction [seconds]	

4. Move "Palette" back to "Templates".



### **Create Own Model Library**

- There are model libraries with a huge variety of models by default. However, these models are encrypted and not editable.
- It is possible to create own libraries, either from scratch or from modified standard model libraries. These libraries created can be edited from time to time and can also be used as a building block to build own models, just like the standard model libraries.
- In coming tutorials, steps to create own model libraries from modified standard model libraries containing a modified CP-L-CONVEYOR, CP-L-SOURCE and CP-L-SINK which supports MPS workpiece will be shown. Steps to create MPS workpiece in CP environment is shown in previous chapter.

• Important! If the library is made from modified standard library, avoid mixing new model library with standard library for an error free simulation.



#### Elements of a model library

- CF.7z
- Textures.7z
- Model
  - Model.modx
  - Model.png
  - Model.ini
  - Model.html
- InitModBib.modx
- InitModBib.ini
- InitModBib.fin



#### Steps to create own model library

- 1. Create CF.7z and Textures.7z.
- 2. Create InitModBib (*.fin, *.ini and *.modx).
- 3. Create the models.
- 4. Link the model library in CIROS.



#### **Create CP.7z and Textures.7z**

- 1. In a chosen window's path, create an empty folder. This will be the new model library folder.
- 2. In CIROS Studio, create a CIROS model in the folder created above. Name it InitModBib.modx.
- 3. Add any model from Festo CP System model library.
- 4. Delete all Materials in Model Explorer.
- 5. In <project folder>\CF, replace any modified python or irl scripts.
- 6. Compress following folders with 7-Zip:
  - 1. <project folder>\CF
  - 2. <project folder>\Textures

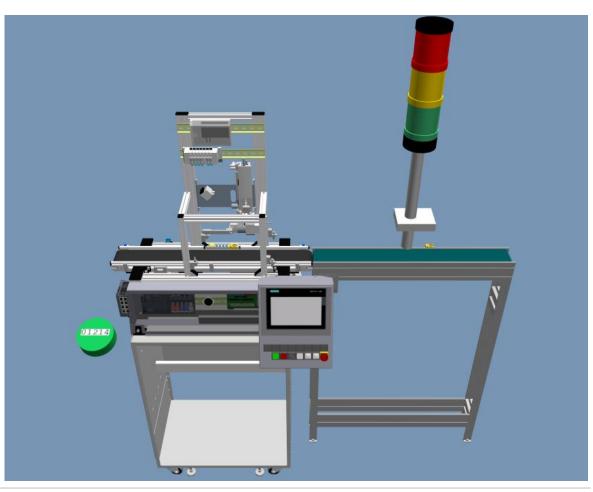


#### Create InitModBib (*.modx, *.ini and *.fin)

- 1. Go to CIROS project created (InitModBib.modx).
- 2. Make sure Model Explorer  $\rightarrow$  Templates contains all the parts required in the new model library. Note: Import the part as an object to CIROS model and convert it into template if required.
- 3. Make sure Model Explorer  $\rightarrow$  Objects  $\rightarrow$  Werkstuecke contains all the meta objects required in new model library.
- 4. In Model Explorer, delete all objects except "Werkstuecke".
- 5. In Model Explorer, delete all materials.
- 6. Save all.



- At the end of this tutorial, a resource consists of a belt with application module traffic light is created (see picture on the right) and is able to work together with a CP-L-CONVEYOR.
- The module can work in default mode and MES mode.
- In default mode, red, yellow and green LEDs will light up one after another.
- In MES mode, the LED configured in MES will light up.





#### Steps to create base module.

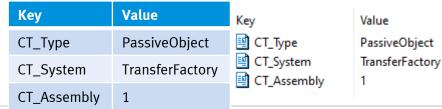
- 1. Insert CP-L-CONVEYOR from Festo CP System model library.
- 2. Insert ModLibs\Conveyors\ConveyorBelt.



3. Delete all Extended properties in Properties\Extended.

Properties			
General Pose Dimension Visualization	Key	Value	
Extended			
Info ORL			
D			

4. Add following extended properties.



- 5. Delete following components in ConveyorBelt.
  - LedBackwards
  - LedForwards
  - LedStopped
  - ProfilesTopX
  - All inputs
  - All outputs
- 6. Go to Properties\General, select object type as Inactive object.
- 7. Go to Modelling\Geometry optimization.
- 8. Select Merge hulls with matching material or area color to optimize ConveyorBelt.



#### Steps to create base module.

- 7. Change the size of ConveyorBelt in Properties\Dimension.
  - X = 700 mm
  - Y = 120 mm
  - Z = 974.50 mm



**Hint:** Use prepared CAD model "ConveyorBelt.stp" to skip step 2, 3, 5 and 7. Please request the CAD model from Festo Didactic.

- 10. Copy CP-L-CONVEYOR\SPS_A and paste it under object ConveyorBelt. The object is renamed to SPS_A_1.
- 11. Change position of SPS_A_1 in world coordinate.
  - X = 350 mm
  - Y = 54.55 mm
  - Z = 975 mm
- 12. Copy CP-L-CONVEYOR\Conn1 and Conn2 and paste them under ConveyorBelt.

M. A. Camurus Balt
✓ ▲ ConveyorBelt
> 🛃 SPS_A_1
🕒 Legs
ProfilesTopY
ProfilesTopX
ProfilesBottomY
ProfilesBottomX
Rolls
🛞 Belt
Orive
Conn1
Conn2



Trigger

### Steps to Create own Virtual Machine Communicating with MES4

#### Steps to create base module.

General

Visualization

Extended

Bus client

Connectors

Fault Measuring

Collision detection

Info ORL

Pose Dimension

13. Move ConveyorBelt\Conn1 and Conn2 to right and left end of ConveyorBelt respectively.

Object coordinate	X [mm]	Y [mm]	Z [mm]
Conn1	0	54.55	975
Conn2	700	54.55	975

Connectors

Element

Conn1

Conn2

Element

Туре

Pose

Pose

Triggers

Туре

Pose

14. Select object ConveyorBelt. Go to Properties\Connectors. Add following connectors and triggers.

Class

CPLab

CPLab

Class

ConveyorBelt CPLab Modeling

Active

					55			
	Туре:	Pose 🗸	Туре:	Pose 🗸	Туре: р	ose 🗸		
	CIROS element		CIROS element		CIROS element			
12 to right and left end of	Element:	Conn1 ~	Element:	Conn2 ~	Element: (	ConveyorBelt 🗸 🗸		
	General		General		General			
	Class:	CPLab $\checkmark$	Class:	CPLab ~	Active:	Modeling ~		
Z [mm]	Display text:	CPLab	Display text:	CPLab	Class:	CPLab ~		
975	Offset (X/Y):	10 Pixel / 20 Pixel	Offset (X/Y):	10 Pixel / 20 Pixel	Permanent conr	nection:		
515	Pose connector		Pose connector		Search			
975	Snapping: 🕑	X 🔽 Roll	Snapping: 🛛 🗸	Roll	Object:	<not chosen=""> ~</not>		
		Y Ditch	🗹 Y	Pitch	Meta object:	<not chosen=""> ~</not>		
anartias) Connectors Add		Z 🗌 Yaw	🔽 Z	🗌 Yaw	Nearest connect	tion: 🔽		
operties\Connectors. Add	Offset:	0.00 mm 0.00°	Offset:	0.00 mm 0.00°	Pose trigger			
		0.00° mm		0.00 mm 0.00°	Distance:	50 mm		
		0.00° mm		0.00 mm 0.00°		OK Cancel		
		OK Cancel		OK Cancel				
						<b>↑</b>		
Permanent								
g No								

 $\times$ 

Connector

Connector



#### Steps to create base module.

15. Go to SPS_A_1\Inputs, conne	ect DIN1_x5 to	1.				
•	✓ ▲ ConveyorBelt	E DIN0_x1	001 [	Digital	0	-
	✓ ▲ SPS_A_1	DIN0_x2		Digital		-
	> 🛃 Bandsteuerung_A_1	CIN0_x3	003 [	Digital	0	-
		DIN0_x4	004 [	Digital	0	-
	> 🛃 Koppelsensor_rechts_1	E DIN0_x5	005 [	Digital	0	-
	> 🛃 RFID_A_1	E DIN0_x6	006 [	Digital	0	-
	> 🛃 Sensor_A_4	E DIN0_x7	007 [	Digital	0	-
	> 🛃 Sensor A 5	E DIN1_x0	008 [	Digital	0	-
	> 🛃 Sensor_A_6	E DIN1_x1	009 [	Digital	0	-
	> 🛃 Sensor_A_7	E DIN1_x2	010 [	Digital	0	-
	> 🛃 Sensor_Band_links_1	E DIN1_x3	011 [	Digital	0	-
	> A Sensor_Band_rechts_1	E DIN1_x4	012 [	Digital	0	Erkannt
	> A Stopper_A_1	E DIN1_x5		2	<1>	<constant< td=""></constant<>
	> A Switch_Input18_4_6	DIN1_x6		Digital		Erkannt
	> E Inputs	DIN1_x7		Digital		Erkannt
	> I Outputs	DIN18_x0		Digital		Erkannt
16. Open Project Management.	AppConn	DIN18_x1		Digital		Erkannt
10. Open i toject management.	( Аррсони		010 1	Norital	0	Edvapart

- 17. Open project <project folder>\CF\CPSystems\CPSystem_Allgemein.prjx.
- 18. In Controllers\SPS_A_1, assign CPSystem_Allgemein. Do the same for SPS_A.

Project Management	
<ul> <li>Controllers</li> <li>SPS_A</li> </ul>	Controllers > SPS_A_1
SPS_A_1  Spicets  Spicets  Spiceter CPSystem_Allgemein (IRL)	Project assignment Project: CPSystem_Allgemein ~
Files	Configuration
	Code sequence trace

- 19. Right click on Projects\CPSystem_Allgemein(IRL) and compile the project. Make sure that all projects are compiled successfully.
- 20. Drag and snap ConveyorBelt to CP-L-CONVEYOR.
- 21. Base module is created.

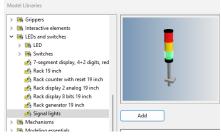
nt value>



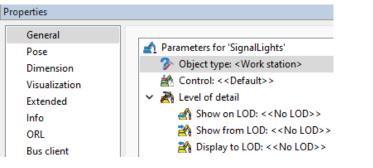


#### Steps to create application module.

1. In model libraries, insert LEDs and switches\Signal lights.

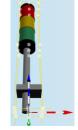


2. Select SignalLights, in Properties\General, change object type to Work station.

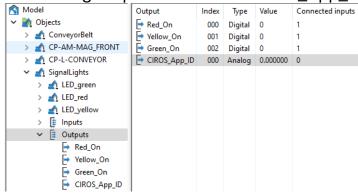


3. Move origin of SignalLights to middle bottom of the object.

Hint: With help of 3D marker.



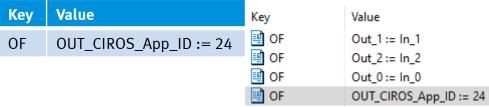
- 4. Change z-coordinate of SignalLights in world coordinate to 975 mm.
- 5. Add analog output and name it CIROS_App_ID.





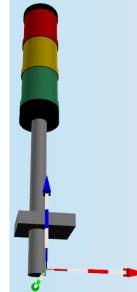
#### Steps to create application module.

6. Select SignalLights, add following extended property.



- 7. In CP System model library, add any application module, for example, CP-AM-MAG_FRONT.
- 8. Copy CP-AM-MAG_FRONT\AppConn and paste it under object SignalLights.
  - 🗸 🔬 SignalLights
    - > 🛃 LED_green
    - > 🛃 LED_red
    - > 🛃 LED_yellow
    - > 🗄 Inputs
    - > 🚦 Outputs
    - 🛞 Basis
    - AppConn

9. Move AppConn pose to front middle of object's bottom plane.





Step	s to c	reate	e appli	cati	on mod	ule.						Connector			×	Tag Edit	or I/O	Tag			×
												Type:	I/O		$\sim$	Type Outpo		Tag App_DI0			
												CIROS element				Outp	_	App_DI1			
0.	Select 9	Signa	ill ights.	. in Pi	roperties	s\Conn	ecto	rs, conf	igure co	onnect	ors		c: 11:1			Outpu		App_DI2 D CIROS_App_ID			-
		-	_		0 0 0 0 0 0 0			,	19610 00		010	Element:	SignalLigh	ts		Input	Red	App_DQ0			_
i	and trig	ggers	as follo	W.								General				Input	Yellow Green	App_DQ1 App_DQ2			-
		50										Class:	Applicatio	'n	~						_
												Display taxts									-
												Display text:			_						
	Connector			×								Offset (X/Y):	10 Pix	el /20	Pixel	Tag:	App_DI0			~ Apply	
	_											I/O connector								Close	
	Туре:	Pose	2	$\sim$								Release ex. conne	actions								
	CIROS element	nt												U				Tringer			×
	Element:	App	Conn	~								Method:	Tag		~			Trigger			^
	General					(	Connect	tors						Edit tags				Туре:	Pose		$\sim$
			l'anti-a				-	<b>F</b> 1 .	<i>C</i> 1									CIROS eleme	ent		
	Class:		lication	~			Туре	Element	Class									Element:	SignalLights		
	Display text:	Арр	lication				Pose	AppConn	Application									crementa	SignalLights		Ť
	Offset (X/Y):		10 Pixel /	20 Pixel			1/0	SignalLights	Application									General			
	Pose connecto												OK	Cancel				Active:	Modeling		$\sim$
		X	🖂 Roll															Class:	Application		$\sim$
		Ωγ	Pitch															Permanent c	onnection:	C	
		-	_			L.												Search			
		<b>Z</b>	🗹 Yaw			1	Triggers	;										Object:	<not chosen<="" td=""><td>1&gt;</td><td></td></not>	1>	
	Offset:	0.0	00 mm	0.00°		Г	-	<b>F</b> 1	<i>C</i> 1	A	<b>D</b> .										Ľ.
		0.0	00 mm	0.00°			Туре	Element	Class	Active	Permanent							Meta object:	<not chosen<="" td=""><td>&gt;</td><td>~</td></not>	>	~
		0.0		0.00°			Pose	SignalLights	Application	Modeling	No							Nearest conn	nection:	2	
				0.00														Pose trigger			
		ОК	Can	ncel														Distance:	200 m	ım	
1					1															_	
																			ОК	Cancel	



#### Steps to program virtual machine in IRL

- Add app ID to "Applikationen.irl". 1.
  - In "Project Management\Projects\CPSystem Allgemein(IRL)\Files", open 1. "Applikationen.irl".
  - 2. In Function "App()", add following line,

```
if(applicationID = 24) then return(appSignalLights(mes.Parameter)); endif;
```

```
FUNCTION App( IN INT: applicationID; IN MESFields : mes ) : int;
BEGIN
```

```
if ( applicationID = 0 ) then return ( 0 ); endif;
        if ( applicationID = 1 ) then return ( appBohren ( mes.Parameter ) ); endif;
        if ( applicationID = 2 ) then return ( appWenden ( mes.Parameter ) ); endif;
        if (applicationID = 3 ) then return (appPresse (mes.Parameter )); endif;
        if (applicationID = 4) then return (appMuskelpresse(mes.Parameter)); endif;
        if ( applicationID = 5 ) then return ( appMagazin ( mes.Parameter ) ); endif; }
       if ( applicationID = 6 ) then return ( appKamera ( mes.Parameter ) ); endif;
        if (applicationID = 7 ) then return (appTunnelofen (mes.Parameter )); endif;
        if ( applicationID = 8 ) then return ( appEntnahme ( mes.Parameter ) ); endif;
        if (applicationID = 9) then return (appNacharbeit (mes.Parameter)); endif;}
        if ( applicationID = 10 ) then return ( appWerkstueckausgabe ( mes.Parameter ) ); endif;
        if (applicationID = 11 ) then return (appEtikettieren (mes.Parameter )); endif;
        if ( applicationID = 12 ) then return ( appMessenAnalog( mes ) ); endif;
        if (applicationID = 13 ) then return (appLager (mes.Parameter )); endif;
        if (applicationID = 14) then return (appRobMontage (mes.Parameter)); endif;
        if (applicationID = 15) then return (appRobCNCBeladen (mes.Parameter)); endif;
        if (applicationID = 16 ) then return (appLagerLab (mes.Parameter )); endif;
        if (applicationID = 17) then return (appManual (mes.Parameter)); endif;
        if ( applicationID = 18 ) then return ( appPickByLight ( mes.Parameter ) ); endif;
        if (applicationID = 19) then return (appMagazinFront (mes.Parameter)); endif;
        if (applicationID = 20) then return (appMagazinBack (mes.Parameter)); endif;
       if ( applicationID = 21 ) then return ( appBoxBufferManual ( mes ) ); endif;
        if ( applicationID = 22 ) then return ( appBoxRobotBvpass( mes ) ); endif;
        if (applicationID = 23 ) then return (appBoxRobotMill105 (mes ) ); endif;
        if ( applicationID = 24 ) then return ( appSignalLights ( mes.Parameter ) ); endif;
        return( 0 );
ENDFCT :
```



#### Steps to program virtual machine in IRL

2. Add function "appSignalLights()" to Applikationen.irl. FUNCTION appSignalLights(IN ARRAY[1..7] OF REAL: Parameter) : int; VAR

		470
	INPUT BOOL : RedIsOn	AT 0;
	INPUT BOOL : YellowIsOn	AT 1;
	INPUT BOOL : GreenIsOn	AT 2;
	OUTPUT <mark>BOOL</mark> : RedOn	AT-1;
	OUTPUT BOOL : YellowOn	AT 0;
	OUTPUT BOOL : GreenOn	AT 1;
	REAL : redMES;	
	REAL : yellowMES;	
	REAL : greenMES;	
BEGIN		
	RedOn := false;	
	YellowOn := false;	
	GreenOn := false;	
	,	
	if(MyResourceId = <mark>0</mark> ) then	
	{Standardmo	ode}
	RedOn := tru	ie:
	WAIT 1.0 SEC	
	RedOn := fai	-
	YellowOn :=	true;

#### else

redMES := Parameter[1]; yellowMES := Parameter[2]; greenMES := Parameter[3]; IF redMES = 1 THEN RedOn := true; ENDIF; IF yellowMES = 1 THEN YellowOn := true; ENDIF; IF greenMES = 1 THEN GreenOn := true; ENDIF;

WAIT 1.0 SEC:

YellowOn := false; GreenOn := true;

#### WAIT 1.0 SEC; RedOn := false; YellowOn := false; GreenOn := false;

return (0); ENDFCT;

endif;



#### Steps to program virtual machine in IRL

- 3. Save the program.
- 4. Compile the project and make sure that all projects are compiled successfully.
- 5. A resource with base and application module is created.



# **Troubleshoot in External Document**

CIROS-CP_Troubleshoot_EN_v7.1_xxxxx.pdf