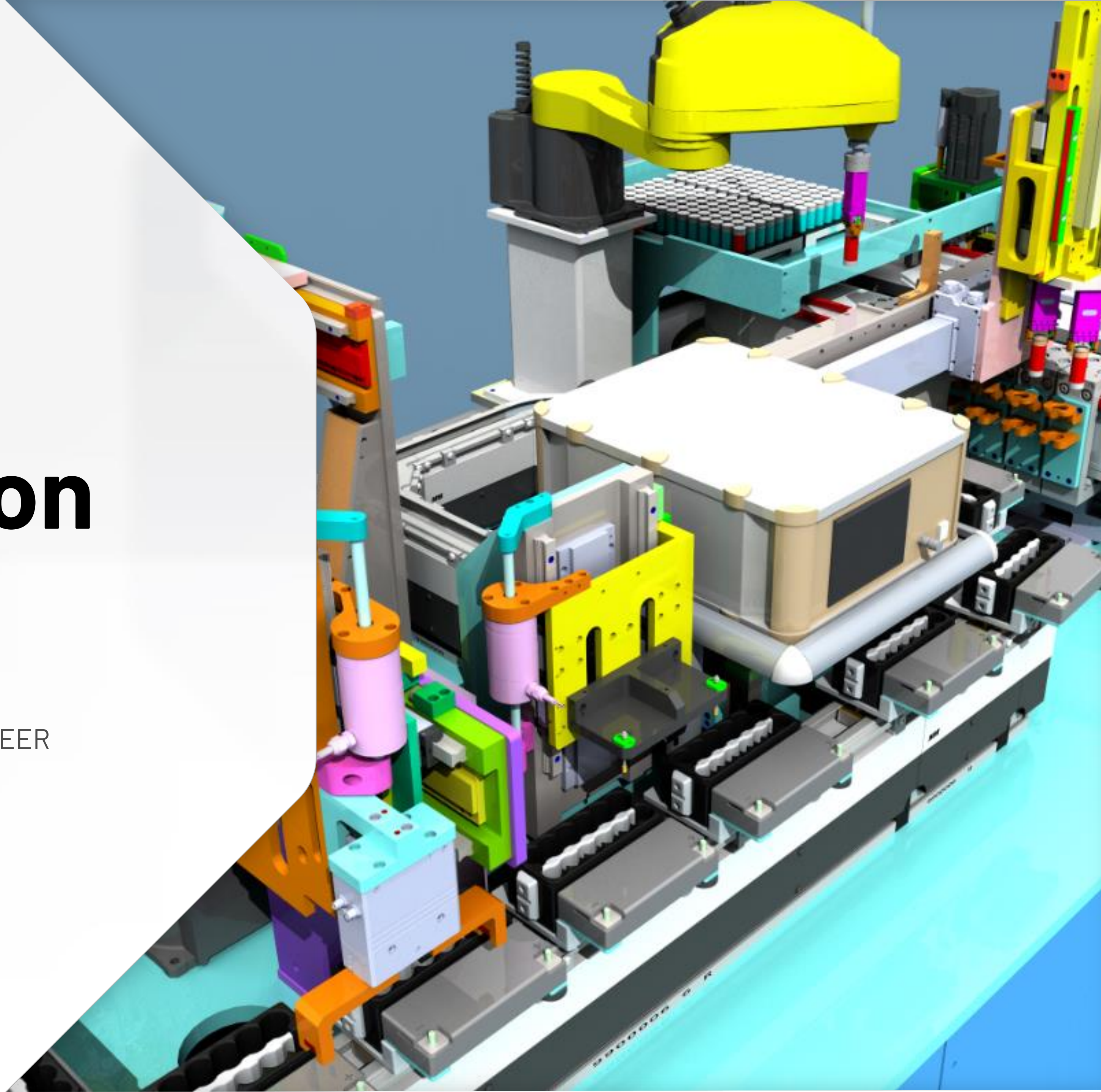
Device Emulation

THEODOROS ASLANIDIS • SENIOR COMMERCIAL ENGINEER
SEBASTIAN ZAJAC • EMULATE3D SUPPORT ENGINEER

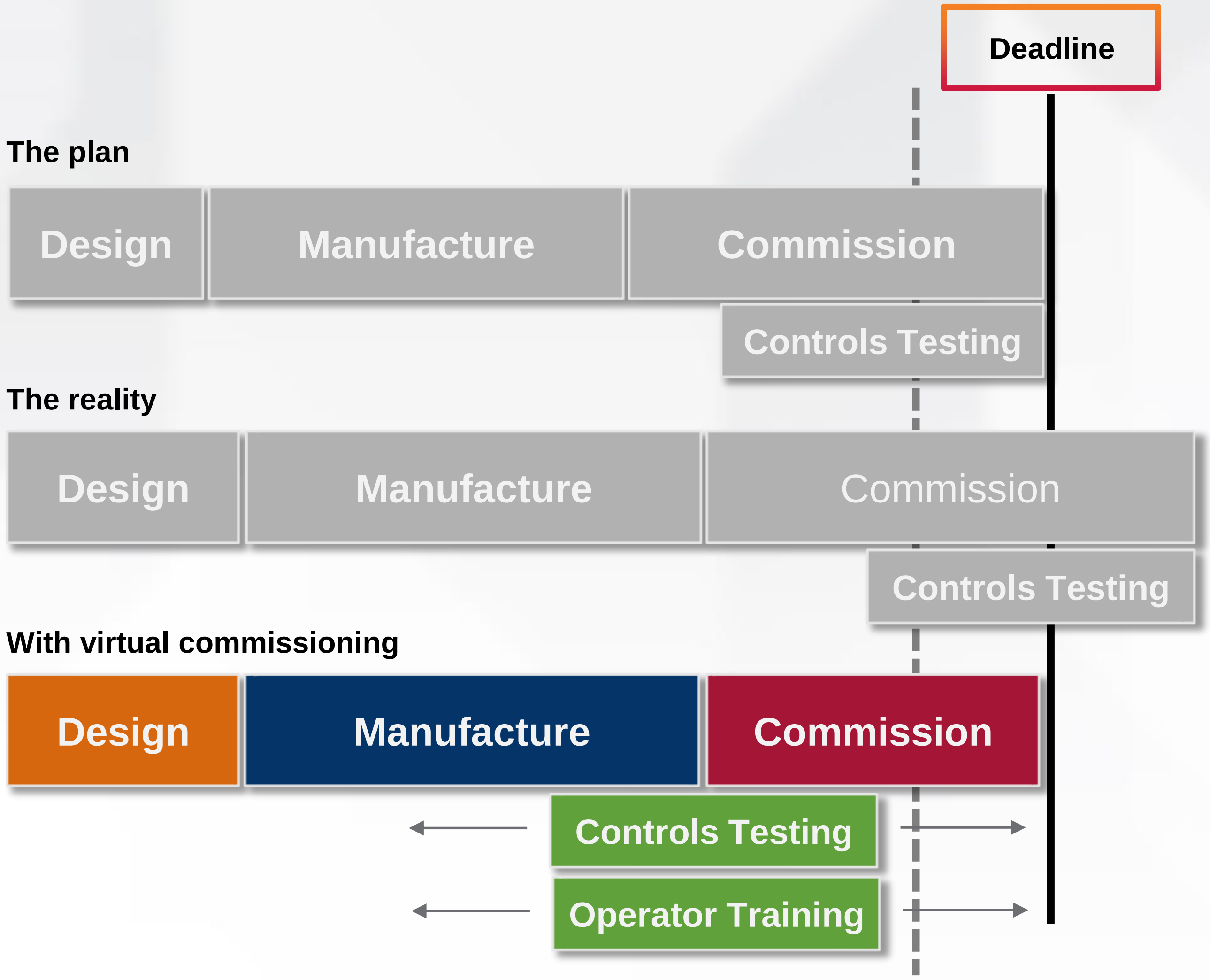
expanding **human possibility**®



PUBLIC

Why Controls Testing with Emulate3D?

- Take Commissioning Off the Critical Path
- Perform More Testing, Sooner
- Perform Testing that May be Otherwise Impossible
- Empower Multiple Engineering Disciplines to Work on the Same Systems in Parallel



Objective for Emulate3D Controls Testing

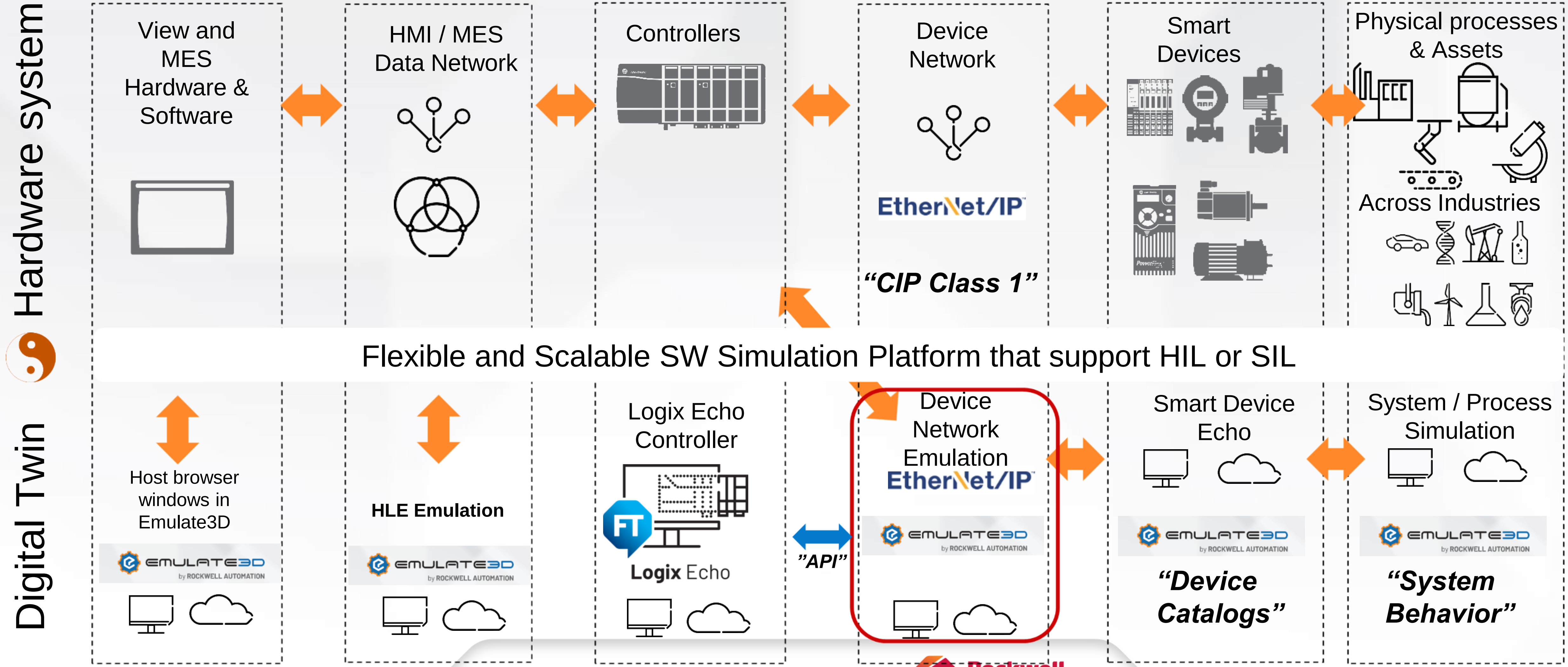
Virtual Commissioning system fidelity is a key factor to success with Emulate3D

- **Simplify** the method of connecting a Controller to a Controls Testing model in Emulate3D while improving the Emulation fidelity of the system by providing Device Echo capabilities
- Mimic the **local intelligence of IO modules** and **smart devices** as part of their control system. Rockwell devices are configured via **AOP**'s
- Include the physical wiring system for testing by using the **Wiring Diagram** feature
- Implement the **Diagnostics and Fault Handling** of the devices.
- Minimum to zero **PLC program modifications** - same code for digital twin and real hardware.
- Deliver **IO and RA basic device echo capability**.
- Provide the **Device Echo API** to the customers or business units who want to develop high fidelity emulations – particularly for complex systems like Motor Control Devices and advanced Smart Device Components like Sensors.
- **Add-On Profiles (AOP)** determine the module behavior as well as the connection type, thus defining the IO Mapping in the controller, smart functions.

I/O and Components

Digital Twins deployment

Most elements of a system can be emulated with various levels of fidelity based on specific objectives.
The current focus is on “controls logic testing”, not on detailed “digital design prototyping”



Emulate3D Connectivity Overview

Real-time Controls Testing with Multi IO-Slave Emulation - Hardware In the Loop

Logix

EtherNet/IP

CIP Class 1

Rockwell Automation

S7/1500

PROFINET

PROFIBUS

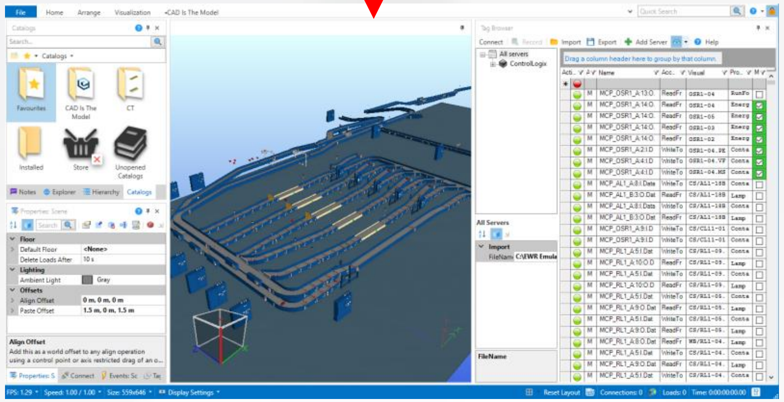
Simit Unit

SIEMENS

WMS / Internet / External

TCP / UDP
SOAP

Telegrams
SAP EWM / MQTT



Controller Messaging
Lower bandwidth connections

EtherNet/IP

CIP Class 3

Logix

Micro800

PCCC / CSP

SLC 5/05

MicroLogix

Rockwell Automation

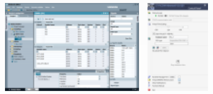
Controller co-simulation - Software In the Loop API

Logix Echo

Logix Echo

Rockwell Automation

PLCSIM Advanced



SIEMENS

GXWorks 2

MITSUBISHI ELECTRIC

Gateway Messaging

OPC UA

Robot Controllers

FANUC

ABB

COMAU

DENSO robotics

CIP Class 3	CX, NJ Series	OMRON AUTOMATION & SAFETY
S7 Functions Fetch/Write	Simotion, Simatic, PLCSim	SIEMENS
Modbus	Modbus compatible platforms	Schneider Electric; SEW EURODRIVE; UNITRONICS
SLMP MX Component	Q Series, iQ-R Series, FX, GX Works 2	MITSUBISHI ELECTRIC
PVI		B&R
ADS	TwinCAT 3	BECKHOFF

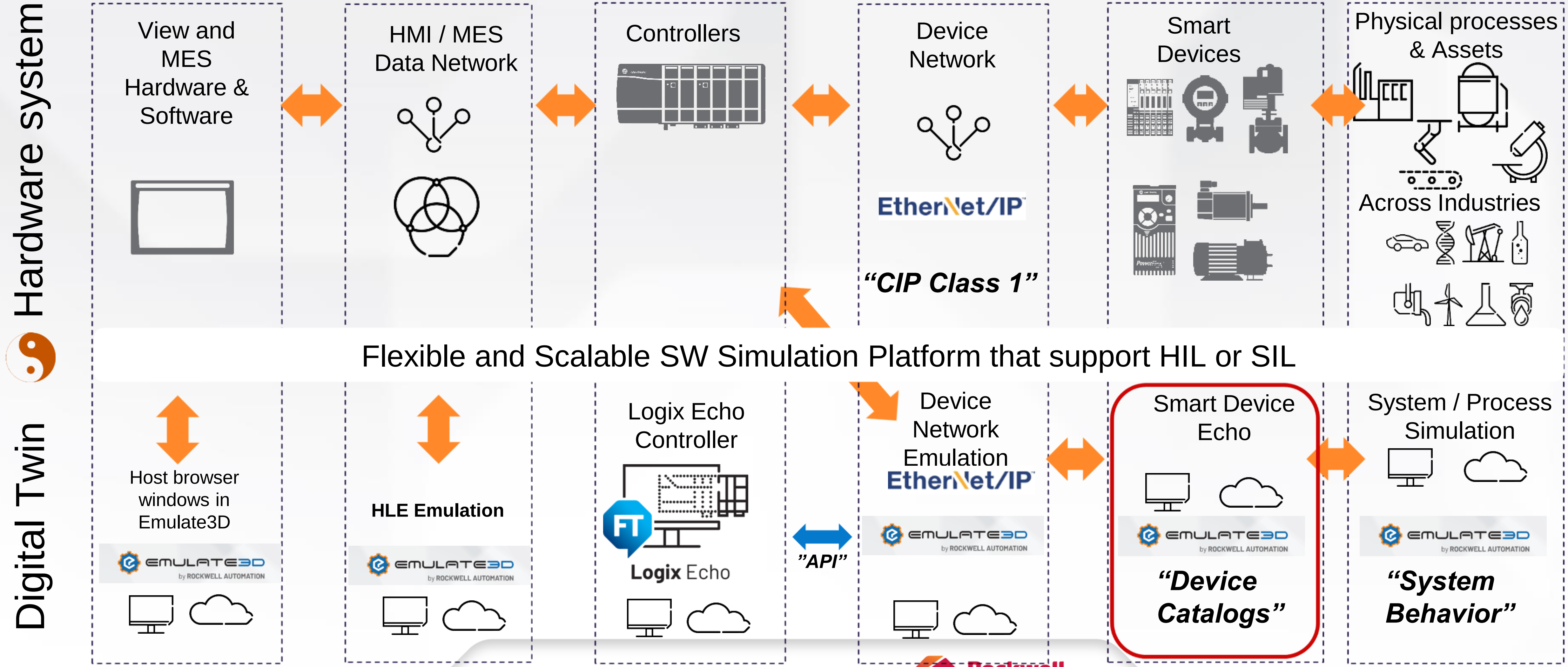
IO Connectivity – EtherNet/IP CIP class 1 to Logix

- **The entire IO tree is available to your application:**
  - All Device connection information
  - Looks for tags / IO modules "in use" in the simulation.
  - For those, identify the IP address of the parent adapter node and attempt to emulate the IP address.
    - If the same node is on the physical network, the DNE app will back off.
  - Multiple IP nodes are emulated at once – utilizes multi-homing of
    the node IP addresses on the Windows host
    - When the app exits, the IP addresses are released again
  - On Change events indicate updates from/to the controller
  - IO in the local rack is excluded from the class 1 connection but can be reached over Class 3.

- **Pre-requisites:**
  - Administrative rights on E3D computer
  - CIP Class 1 connection
  - Physical Logix controller,
    or
  - FactoryTalk Logix Echo API (req. for Safety IO)
  - Firewall exception
    - Add public exception for IPCHost.exe
  - RSLinx configuration
    - Disable "Accept UDP Messages on Ethernet Port"
  - FactoryTalk Linx configuration
    - Disable "Listen on Ethernet/IP encapsulation ports" for all Drivers
  - We do not do Class 1 to other tags – e.g. P/C tags.
    => We do support a 1756-Module emulator (~480 bytes)
    Note that this requires modifying PLC code to remap / copy controller data to/from that IO datablock.

Smart Device Echo

Our initial focus is on various IO components, but intelligent devices are equally important



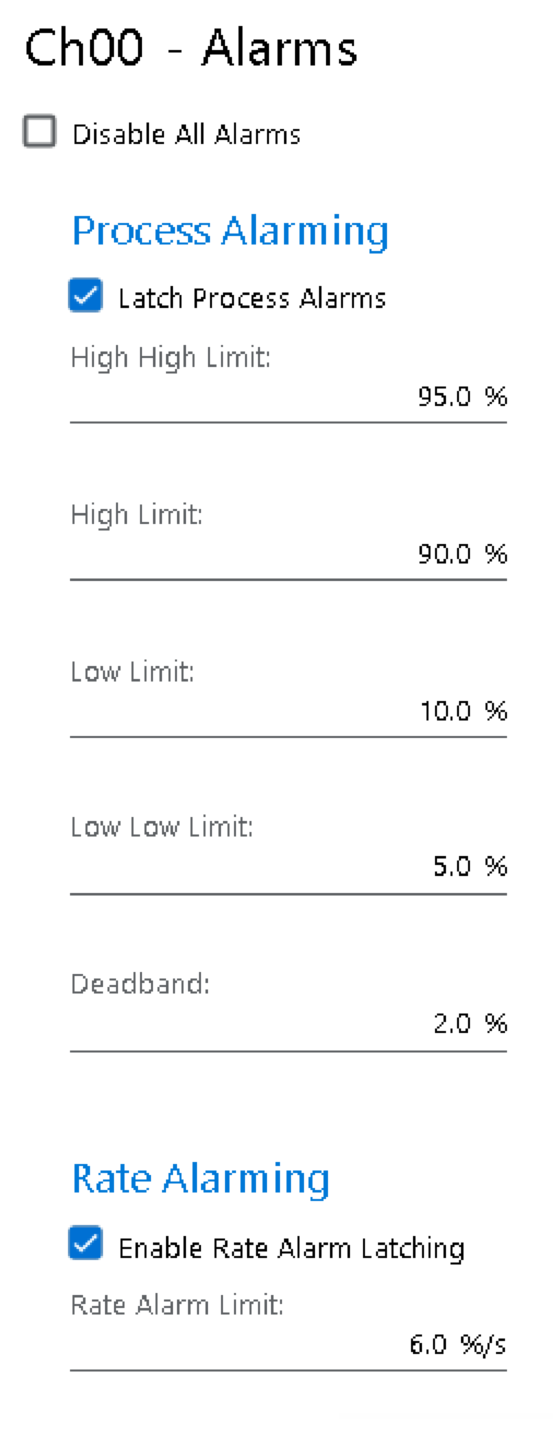
Device Echo API for Rockwell Devices and I/O

Enables emulating the **smart logic behavior** of the device

- **Alarms** – Input value too high, too low, excessive rate of change
- **Outputs** - scheduled outputs, program mode, safe state behavior
- **Faults** – blown fuse, discrepancy faults, fault latching
- **Power supply** – module power, field power
- **Network faults** – module unplugged, cable unplugged
- **Input filters** – de-bouncing – example of a Hardware Firmware function which is not emulated, but the setpoint value is transferred to the model for validation

- And much more…

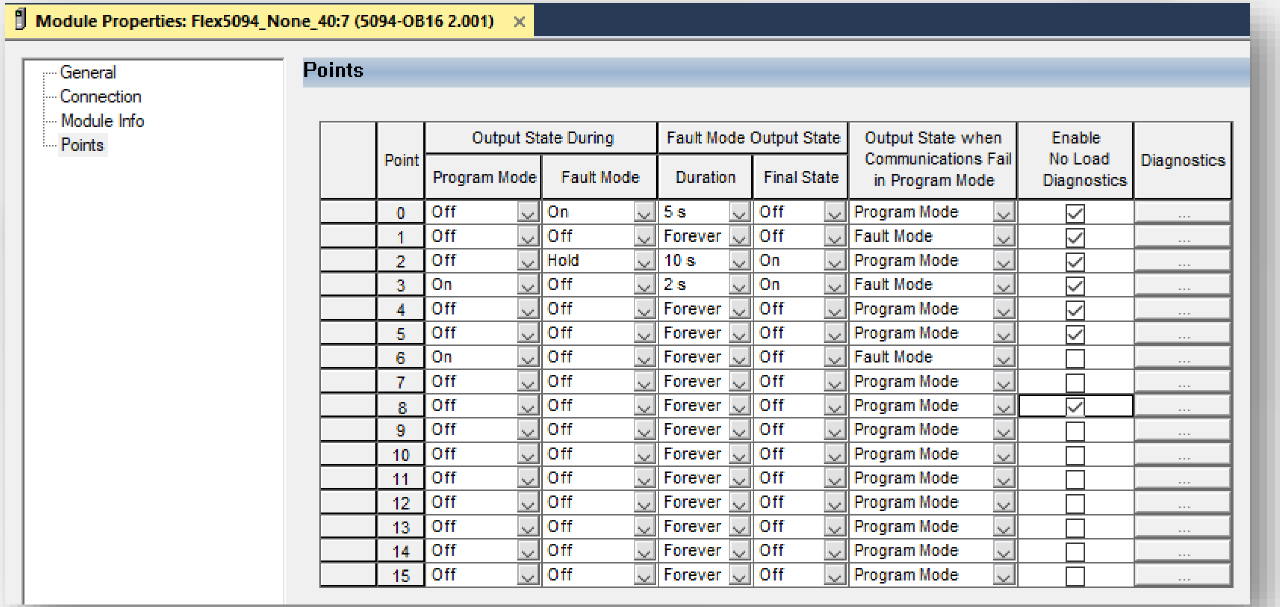
- All these devices are configured via an AOP in Logix Designer and thus a core part of the control system/code.



5094-OB16 Configuration example

A mainstream digital output module with a typical feature set for output fault handling

Channel configuration main dialogs



Module Properties: Flex5094_None_40:7 (5094-OB16 2.001)

General
Connection
Module Info
Points

Points

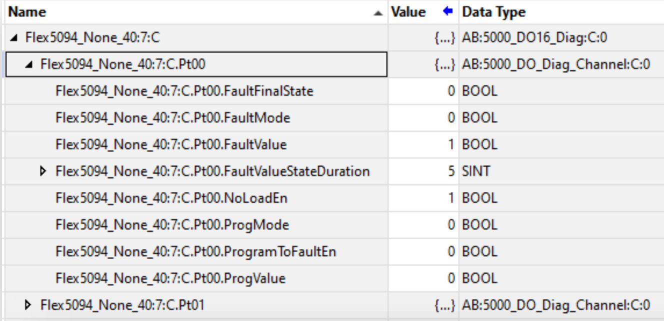
Point	Output State During: Program Mode	Output State During: Fault Mode	Fault Mode Output State: Duration	Fault Mode Output State: Final State	Output State when Communications Fail in Program Mode	Enable No Load Diagnostics	Diagnostics
0	Off	On	5 s	Off	Program Mode	☑	...
1	Off	Off	Forever	Off	Fault Mode	☑	...
2	Off	Hold	10 s	On	Program Mode	☑	...
3	On	Off	2 s	On	Fault Mode	☑	...
4	Off	Off	Forever	Off	Program Mode	☑	...
5	Off	Off	Forever	Off	Program Mode	☑	...
6	On	Off	Forever	Off	Fault Mode	☐	...
7	Off	Off	Forever	Off	Program Mode	☐	...
8	Off	Off	Forever	Off	Program Mode	☑	...
9	Off	Off	Forever	Off	Program Mode	☐	...
10	Off	Off	Forever	Off	Program Mode	☐	...
11	Off	Off	Forever	Off	Program Mode	☐	...
12	Off	Off	Forever	Off	Program Mode	☐	...
13	Off	Off	Forever	Off	Program Mode	☐	...
14	Off	Off	Forever	Off	Program Mode	☐	...
15	Off	Off	Forever	Off	Program Mode	☐	...

The "points" dialogue shows all relevant configuration and configurable diagnostic data which needs to be simulated / verified. The Configuration comes in 5000_DO_DIAG_Channel and all status data is sent to Channel_DO_Diag.

- The IO **Configuration** is done In **Logix Designer** during controller programming, not in Emulate3D.
- The **Virtual Commissioning process** will need to validate that all fault and diagnostic data are configured as needed for the customer's application and behave accordingly when the signals are raised in the controller input image.
- A script can mimic the behavior of the module into Logix and **allows for fault injections**.

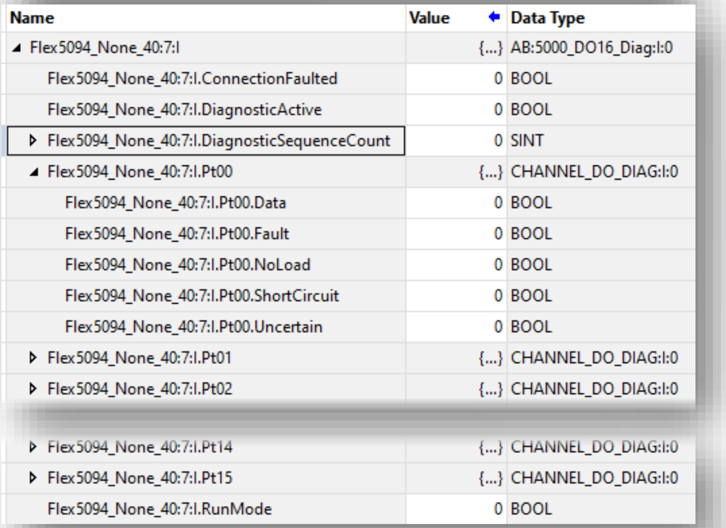
Module config Data, Input back to Logix and output from the controller

Configuration data



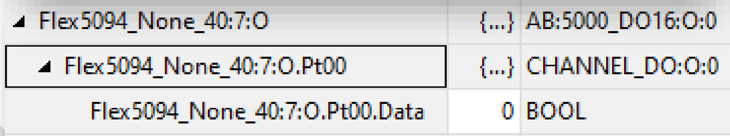
Name	Value	Data Type
Flex5094_None_40:7:C	{...}	AB:5000_DO16_Diag:C:0
Flex5094_None_40:7:C.Pt00	{...}	AB:5000_DO_Diag_Channel:C:0
Flex5094_None_40:7:C.Pt00.FaultFinalState	0	BOOL
Flex5094_None_40:7:C.Pt00.FaultMode	0	BOOL
Flex5094_None_40:7:C.Pt00.FaultValue	1	BOOL
Flex5094_None_40:7:C.Pt00.FaultValueStateDuration	5	SINT
Flex5094_None_40:7:C.Pt00.NoLoadEn	1	BOOL
Flex5094_None_40:7:C.Pt00.ProgMode	0	BOOL
Flex5094_None_40:7:C.Pt00.ProgramToFaultEn	0	BOOL
Flex5094_None_40:7:C.Pt00.ProgValue	0	BOOL
Flex5094_None_40:7:C.Pt01	{...}	AB:5000_DO_Diag_Channel:C:0

Diagnostic feedback data



Name	Value	Data Type
Flex5094_None_40:7:I	{...}	AB:5000_DO16_Diag:I:0
Flex5094_None_40:7:I.ConnectionFaulted	0	BOOL
Flex5094_None_40:7:I.DiagnosticActive	0	BOOL
Flex5094_None_40:7:I.DiagnosticSequenceCount	0	SINT
Flex5094_None_40:7:I.Pt00	{...}	CHANNEL_DO_DIAG:I:0
Flex5094_None_40:7:I.Pt00.Data	0	BOOL
Flex5094_None_40:7:I.Pt00.Fault	0	BOOL
Flex5094_None_40:7:I.Pt00.NoLoad	0	BOOL
Flex5094_None_40:7:I.Pt00.ShortCircuit	0	BOOL
Flex5094_None_40:7:I.Pt00.Uncertain	0	BOOL
Flex5094_None_40:7:I.Pt01	{...}	CHANNEL_DO_DIAG:I:0
Flex5094_None_40:7:I.Pt02	{...}	CHANNEL_DO_DIAG:I:0
Flex5094_None_40:7:I.Pt14	{...}	CHANNEL_DO_DIAG:I:0
Flex5094_None_40:7:I.Pt15	{...}	CHANNEL_DO_DIAG:I:0
Flex5094_None_40:7:I.RunMode	0	BOOL

Actual Output signal



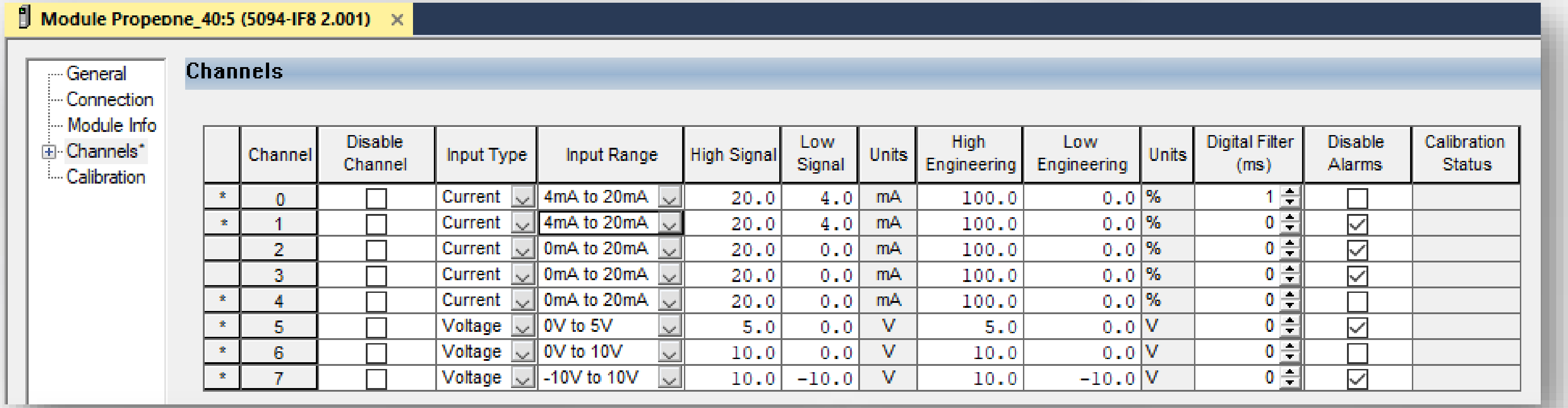
Flex5094_None_40:7:O	{...}	AB:5000_DO16:O:0
Flex5094_None_40:7:O.Pt00	{...}	CHANNEL_DO:O:0
Flex5094_None_40:7:O.Pt00.Data	0	BOOL

Rockwell Automation

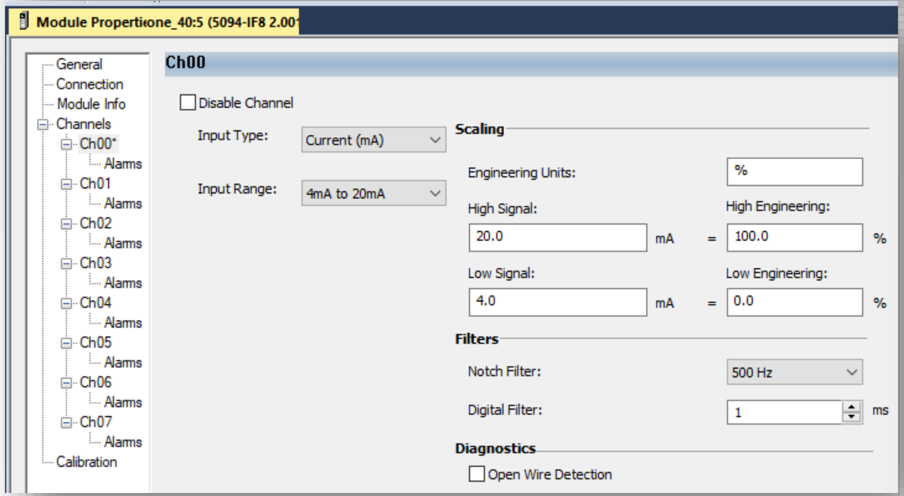
5094-IF8 Configuration Options example

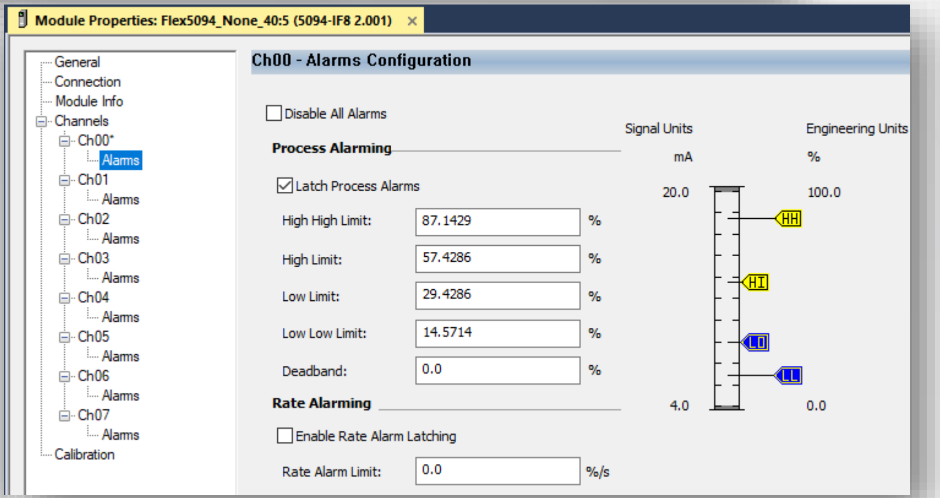
A mainstream analog input module with a typical feature set for signal pre-processing.

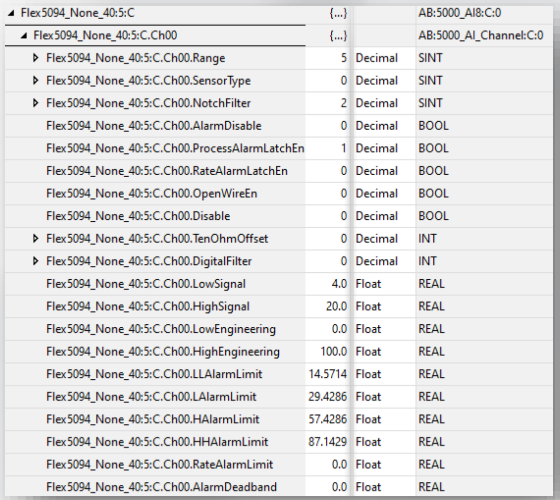
Channel configuration main dialogs



	Channel	Disable Channel	Input Type	Input Range	High Signal	Low Signal	Units	High Engineering	Low Engineering	Units	Digital Filter (ms)	Disable Alarms	Calibration Status
*	0	☐	Current	4mA to 20mA	20.0	4.0	mA	100.0	0.0	%	1	☐	
*	1	☐	Current	4mA to 20mA	20.0	4.0	mA	100.0	0.0	%	0	☑	
	2	☐	Current	0mA to 20mA	20.0	0.0	mA	100.0	0.0	%	0	☑	
	3	☐	Current	0mA to 20mA	20.0	0.0	mA	100.0	0.0	%	0	☑	
*	4	☐	Current	0mA to 20mA	20.0	0.0	mA	100.0	0.0	%	0	☐	
*	5	☐	Voltage	0V to 5V	5.0	0.0	V	5.0	0.0	V	0	☑	
*	6	☐	Voltage	0V to 10V	10.0	0.0	V	10.0	0.0	V	0	☐	
*	7	☐	Voltage	-10V to 10V	10.0	-10.0	V	10.0	-10.0	V	0	☑	



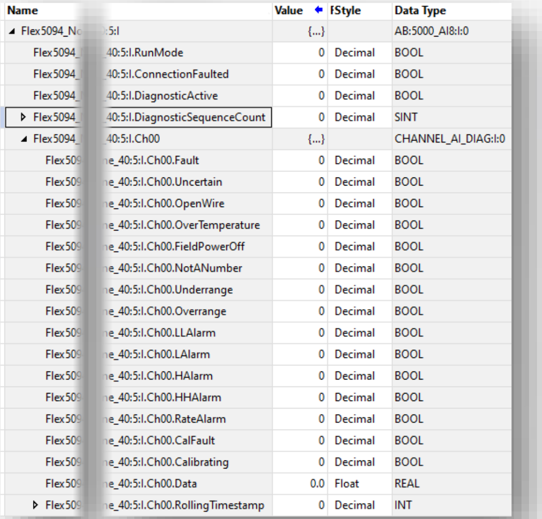




Flex5094_None_40:5:C	{...}		AB:5000_AI8:C:0
Flex5094_None_40:5:C.Ch00	{...}		AB:5000_AI_Channel:C:0
Flex5094_None_40:5:C.Ch00.Range	5	Decimal	SINT
Flex5094_None_40:5:C.Ch00.SensorType	0	Decimal	SINT
Flex5094_None_40:5:C.Ch00.NotchFilter	2	Decimal	SINT
Flex5094_None_40:5:C.Ch00.AlarmDisable	0	Decimal	BOOL
Flex5094_None_40:5:C.Ch00.ProcessAlarmLatchEn	1	Decimal	BOOL
Flex5094_None_40:5:C.Ch00.RateAlarmLatchEn	0	Decimal	BOOL
Flex5094_None_40:5:C.Ch00.OpenWireEn	0	Decimal	BOOL
Flex5094_None_40:5:C.Ch00.Disable	0	Decimal	BOOL
Flex5094_None_40:5:C.Ch00.TenOhmOffset	0	Decimal	INT
Flex5094_None_40:5:C.Ch00.DigitalFilter	0	Decimal	INT
Flex5094_None_40:5:C.Ch00.LowSignal	4.0	Float	REAL
Flex5094_None_40:5:C.Ch00.HighSignal	20.0	Float	REAL
Flex5094_None_40:5:C.Ch00.LowEngineering	0.0	Float	REAL
Flex5094_None_40:5:C.Ch00.HighEngineering	100.0	Float	REAL
Flex5094_None_40:5:C.Ch00.LLAlarmLimit	14.5714	Float	REAL
Flex5094_None_40:5:C.Ch00.LAlarmLimit	29.4286	Float	REAL
Flex5094_None_40:5:C.Ch00.HAlarmLimit	57.4286	Float	REAL
Flex5094_None_40:5:C.Ch00.HHAlarmLimit	87.1429	Float	REAL
Flex5094_None_40:5:C.Ch00.RateAlarmLimit	0.0	Float	REAL
Flex5094_None_40:5:C.Ch00.AlarmDeadband	0.0	Float	REAL

Config Data sent to Emulate3D

Input Data received from Emulate3D which Logix needs to Respond to / Fault Handle.



Name	Value	Style	Data Type
Flex5094_No :5:I	{...}		AB:5000_AI8:I:0
Flex5094_ 40:5:I.RunMode	0	Decimal	BOOL
Flex5094_ 40:5:I.ConnectionFaulted	0	Decimal	BOOL
Flex5094_ 40:5:I.DiagnosticActive	0	Decimal	BOOL
Flex5094_ 40:5:I.DiagnosticSequenceCount	0	Decimal	SINT
Flex5094_ 40:5:I.Ch00	{...}		CHANNEL_AI_DIAG:I:0
Flex509 ne_40:5:I.Ch00.Fault	0	Decimal	BOOL
Flex509 ne_40:5:I.Ch00.Uncertain	0	Decimal	BOOL
Flex509 ne_40:5:I.Ch00.OpenWire	0	Decimal	BOOL
Flex509 ne_40:5:I.Ch00.OverTemperature	0	Decimal	BOOL
Flex509 ne_40:5:I.Ch00.FieldPowerOff	0	Decimal	BOOL
Flex509 ne_40:5:I.Ch00.NotANumber	0	Decimal	BOOL
Flex509 ne_40:5:I.Ch00.Underrange	0	Decimal	BOOL
Flex509 ne_40:5:I.Ch00.Overrange	0	Decimal	BOOL
Flex509 ne_40:5:I.Ch00.LLAlarm	0	Decimal	BOOL
Flex509 ne_40:5:I.Ch00.LAlarm	0	Decimal	BOOL
Flex509 ne_40:5:I.Ch00.HAlarm	0	Decimal	BOOL
Flex509 ne_40:5:I.Ch00.HHAlarm	0	Decimal	BOOL
Flex509 ne_40:5:I.Ch00.RateAlarm	0	Decimal	BOOL
Flex509 ne_40:5:I.Ch00.CalFault	0	Decimal	BOOL
Flex509 ne_40:5:I.Ch00.Calibrating	0	Decimal	BOOL
Flex509 ne_40:5:I.Ch00.Data	0.0	Float	REAL
Flex509 ne_40:5:I.Ch00.RollingTimestamp	0	Decimal	INT

Module Emulator Framework

Tightly integrated into the Demo3D build environment

- Reduced IO Browser bindings – one line per module
- Realistic module visuals – dedicated catalog
- Support for the Wiring Diagram

Scripting API framework for writing Module Emulators

✓ CIP Class 1 – high fidelity emulation

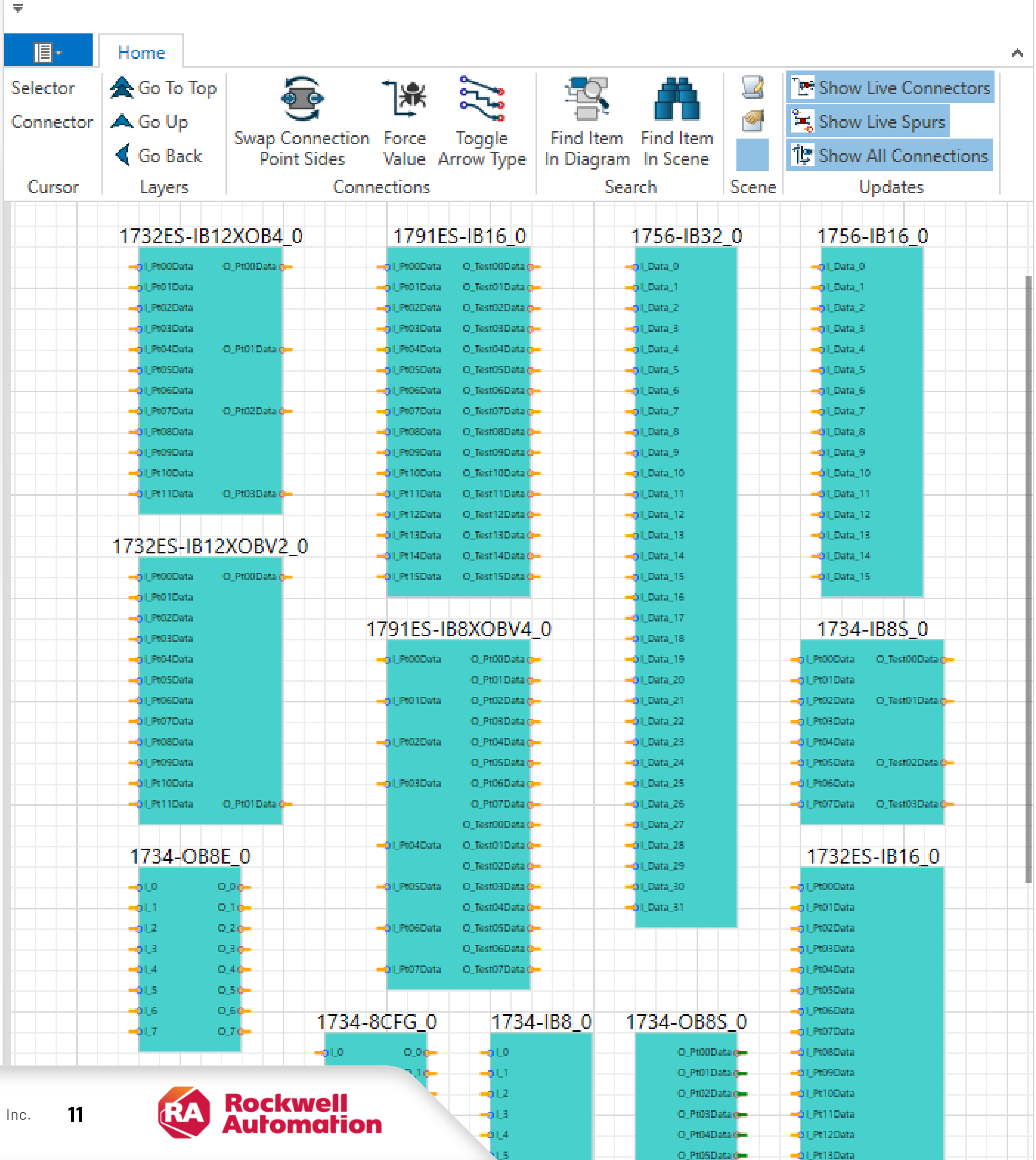
✓ CIP Safety – no 'gateway' Controller, no code changes!
– **Only for GuardLogix Echo controllers**

✓ CIP Sync – time-sync with Controller

× CIP Class 3 – CIP 1 isn't always appropriate

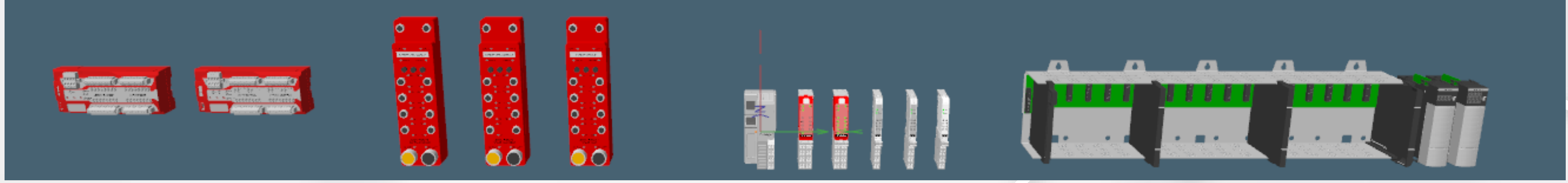
× Logix Echo – we really want full co-simulation

} Planned



Rockwell Automation

Module Emulators

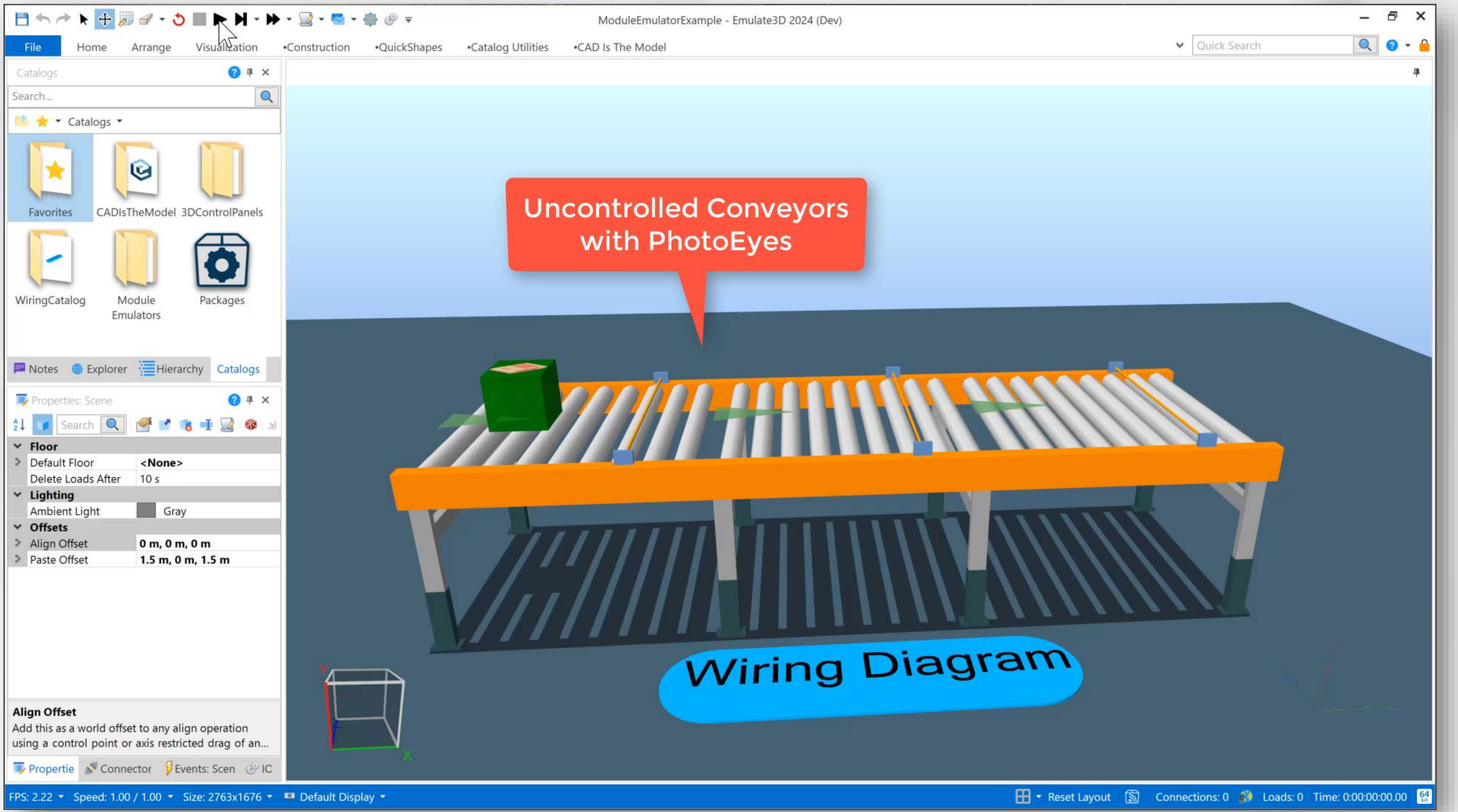


Some prototype emulators already written and available to be used as example code for the API*:

- 1732ES Safety IO
  - IB16, IB12XOBV2, IB12XOB4, IB8XOBV4, IB8XOB8
- 1791ES Safety IO
  - IB16, IB8XOB4
- 1756 IO
  - IA16I, IB16I, IA16, IB16, IC16, IH16I, IM16I, IN16, IV16, IG16, IB16D, IA32, IB32, IV32
  - OA8, OB8, ON8, OC8, OH8I, OX8I, OB8I, OB8EI, OA16I, OB16I, OW16I, OG16, OA16, OB16E, OV16E, OB32
- 1734 IO
  - IA2, IB2, IM2, IV2, IA4, IB4, IM4, IV4, IB8, IV8
  - OA2, OB2, OW2, OX2, OB2E, OB2EP, OV2E, OA4, OB4, OW4, OB4E, OV4E,
  - OB8, OB8E, OV8E,
  - IB8S, OB8S - Safety
- Add your own... – it's an open framework

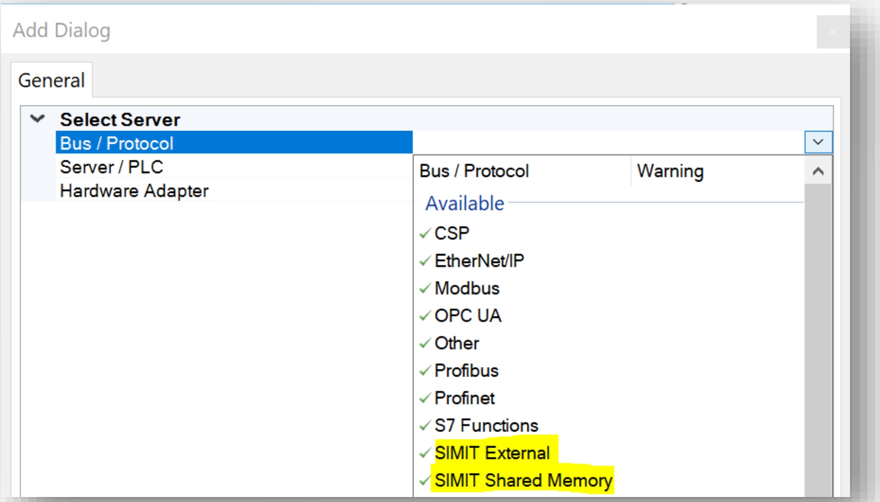
*CIP 3 Reset functions not fully implemented for the Diagnostic or Electronically Fused modules.

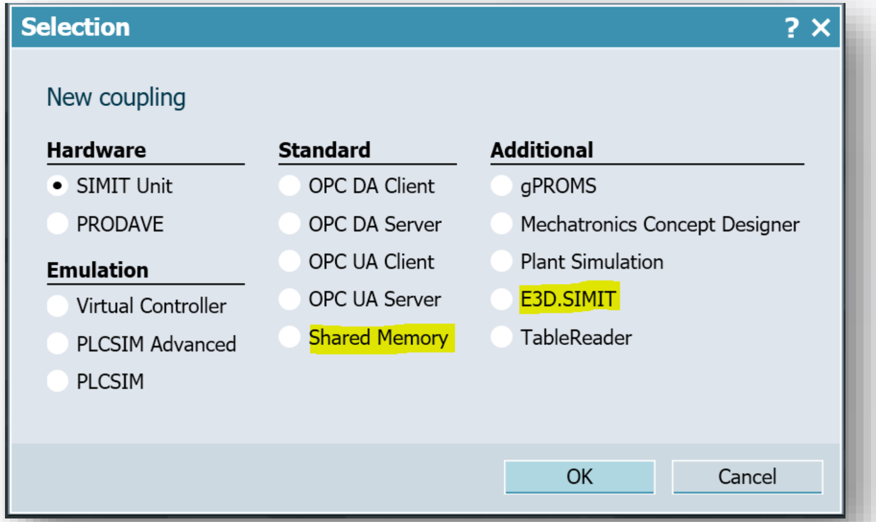
Module Emulators Prototypes Example



Emulate3D – SIMIT communications enhancements

- Support for communications between E3D and the SIMIT simulation platform is now possible through new tag server protocols:
- SIMIT Shared Memory
- SIMIT External Coupling
- These protocols correspond to couplings in the SIMIT software, which are in charge of signal exchange with external partners (software or hardware)

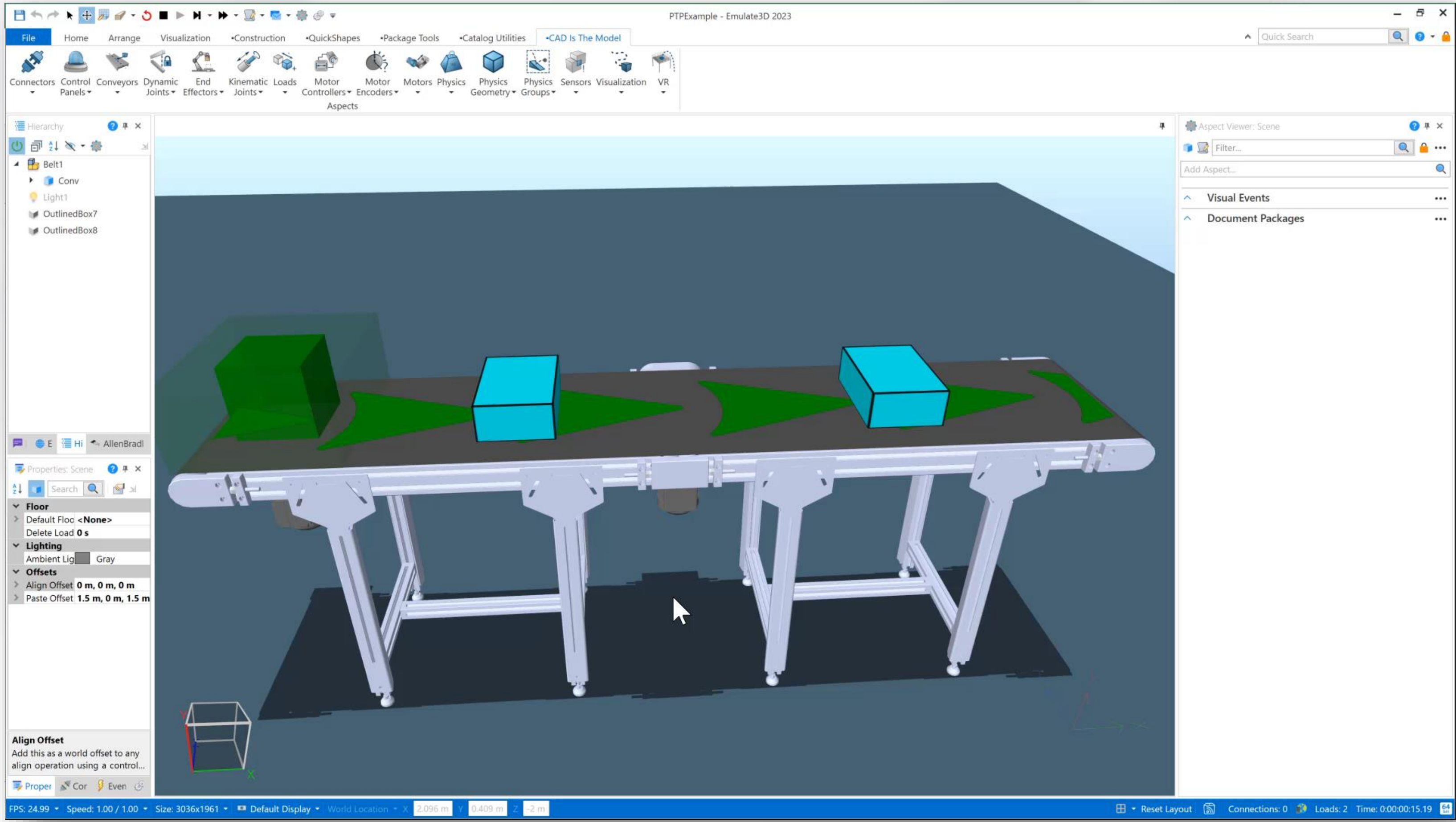




Allen Bradley Velocity Drives Catalog – PowerFlex PTP Mode

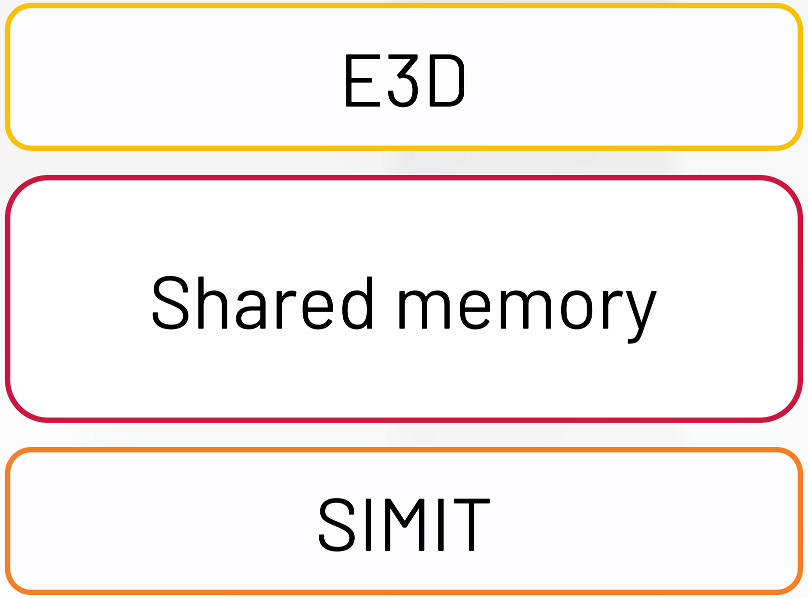
This catalog was expanded with the following features

- Added Point To Point Mode (PTP) to PowerFlex 755 & 755T Drives
- Control Mode (Velocity, PTP) is user selectable
- Aspect reads various Parameters from PLC Memory
- IO Bindings reconfigured based upon Control Mode selection
- All PTP Modes implemented (Absolute, Index, Immediate)
- Commission test your logic using the emulated drives



Shared Memory Coupling for SIMIT

- SIMIT communicates with external software, like Emulate3D, over shared memory
- For general purpose data exchange with high performance
- SIMIT allows multiple partners to access the shared memory, not just Emulate3D



Siemens SIMIT External Coupling

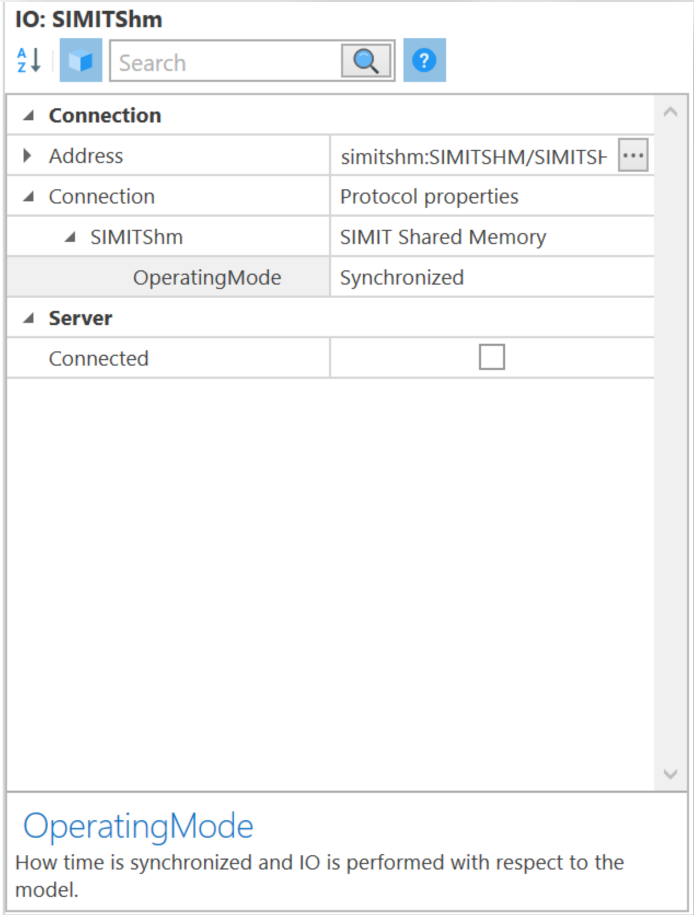
- SIMIT communicates with external partners through a custom implementation of the coupling API, specific to that partner only. A built-in example provided is the TableReader coupling
- Allows for E3D-specific functionality and for future features to be added easily
- Two types of signal exchange are supported:
  - Indirect signal exchange – More suited to older projects that already contain other external couplings which only use indirect signal exchange
  - Direct signal exchange – **Better performance and recommended for new projects**

Co-simulation (SIMIT RCI)

- SIMIT also allows external partners to assume control of the simulation through the RCI interface, which is combined with the existing E3D co-simulation framework to allow data exchanges to be synchronized with respect to model time.
- The effect of this functionality means that the user does not manually start the SIMIT simulation from within the SIMIT software and then have E3D connect to a running simulation. Instead, E3D automatically starts the SIMIT simulation upon connection. Likewise, when E3D disconnects from SIMIT, then the simulation is automatically stopped.

Co-simulation (SIMIT RCI)

- Similar to existing protocols that allow co-simulation (PLCSIM Advanced, Logix Echo), functionality is controlled in the IO Browser properties
- Scheduled (co-sim disabled) - Time is synchronized with real time. IO is performed on a real time schedule.
- Synchronized (co-sim enabled) - Time is virtual time and is tightly synchronized with co-simulation partners. IO is synchronized with the model, happening at specific communication points.



CTQA – Controls Testing Quality Assurance

Automated testing of Emulate3D builds for Controls Testing features – initial focus on Communications performance

- CTQA is our automated testing platform for Emulate3D's Controls Testing features.
- Previous testing for Controls Testing was an entirely manual process. Automating the tests provides us with multiple advantages, but especially helps to ensure consistent and reliable testing.
- The end goal is to run automated tests to verify the functionality of **every protocol** supported by Emulate3D, in **every version** of the software.
- CTQA will also be used to measure the characteristics of each protocol, to give an indication of the expected performance, and to track performance as development continues to ensure performance does not degrade.

CTQA

Protocols being tested:

- CIP (Class 3, Class 1)
- CSP
- PCCC
- Logix Echo
- S7P
- SIMIT (Unit, Shared Memory, External)
- OPC UA
- TwinCAT Ads
- Modbus
- Mitsubishi MX Component
- Telegrams (TCP, UDP, MQTT)
- And more...

Types of tests being ran:

- Connectivity tests
- Round trip tests
- Protocol characterization tests
  - Time spent at every stage of communications pipeline.
  - Latency (early, average, late)
  - Packets sent/received

- Tests are executed on a successfully unit tested build, using Emulate3D's command line, to replicate an end user's configuration.

THANK YOU

Rockwell Automation

expanding **human possibility**®

www.rockwellautomation.com