

# Get Together for Robotics 2025

**Tommaso Fochi**

Principal Engineer

Nov 2025



1

Rockwell  
Automation

2

Robotics:  
Where we  
come from

3

System Control  
Convergence

4

Unified Robot  
Control

5

Questions?

# About Us

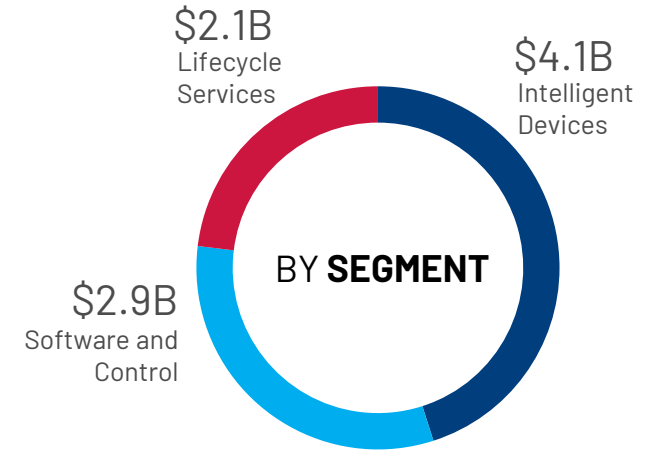
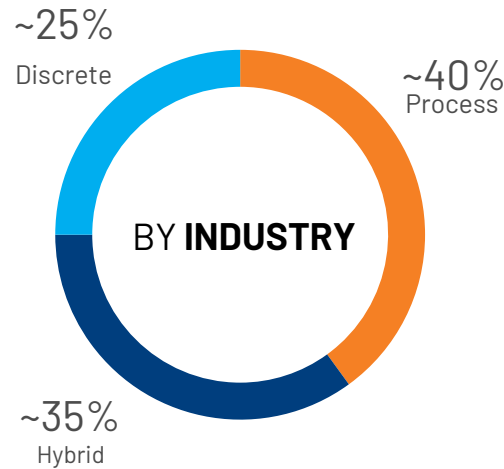
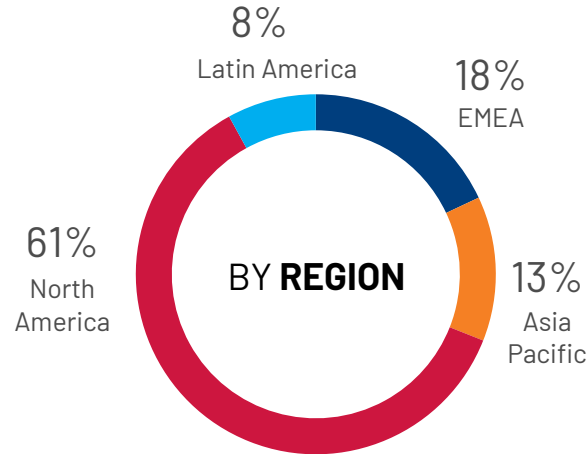


expanding human possibility<sup>®</sup>

# Rockwell Automation At A Glance

2024 GLOBAL SALES

**8.3**  
BILLION  
USD



**100+**

Number of countries



**27k**

Employees: more than half outside the U.S.



World's Most Ethical Companies

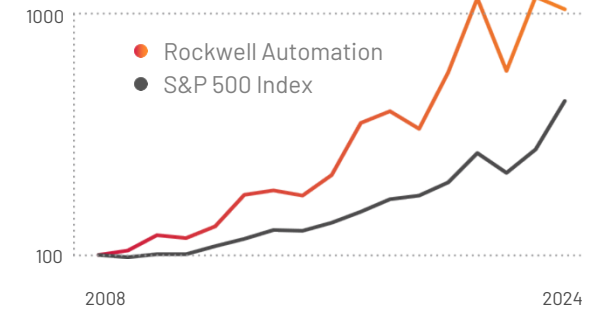


SCIENCE  
BASED  
TARGETS

DRIVING AMBITIOUS CORPORATE CLIMATE ACTION

Committed: January 2024

## Total shareowner return



Serving customers for **122 years**

# 120 + YEARS OF HISTORY

**1962**

Four-sided clock tower is first illuminated in Milwaukee, WI.



**1903**

Lynde Bradley, Dr. Stanton Allen and Harry Bradley fund Rheostat Company and in 1909, the company is reincorporated as the Allen-Bradley® Company.



**1923**

Octagon logo debuts as the Allen-Bradley brand trademark.



**1969**

NASA chooses Allen-Bradley components and went to the moon with Apollo 11 mission.



**1985**

Rockwell International® purchases the Allen-Bradley Company.

**2001**

Company renamed to Rockwell Automation.

**1992**

First Automation Fair event in Philadelphia, PA.

**2011**

Rockwell Automation acquires **Lektronix**, a leading industrial automation repairs and services provider.

**2013**

First cloud-based mobile access to plant floor data via smartphone.

**2016**

Rockwell Automation acquires **Maverick Technologies**, a leading systems integrator with expertise in key process and batch applications.

**2019**

Rockwell Automation and Schlumberger co-found **Sensia** as a joint venture to overcome the performance challenges of oil & gas companies.

**2020**

**Microsoft** and Rockwell Automation expand its partnership to simplify industrial transformation.

**2020**

Rockwell Automation purchases **Fix Ins., Oylo, Kalypso and Avnet** increasing its eco-system of acquisitions, industrial automation repairs and services.

**2021**

Rockwell Automation purchases **Plex System**, a cloud-native smart manufacturing platform operating at scale.

**2022**

Rockwell Automation completes the acquisition of **Cubic**, a company that specializes in modular systems for the construction of electrical panels.

**2023**

Rockwell Automation purchases **Knowledge Lens**, a services and solutions provider that turns data into useful information, combining digital technologies; and **Verve**, a cybersecurity software and services company that focuses specifically on industrial environments.

**Present**

Bringing **The Connected Enterprise®** to Life.

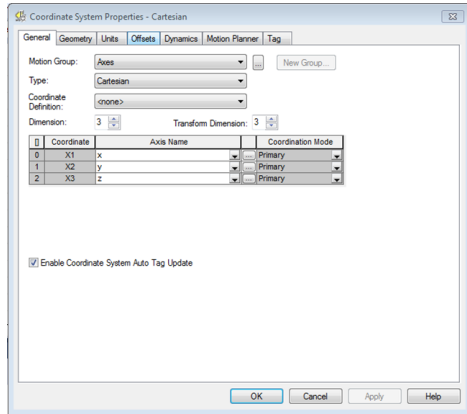


# A quick look at the past





# It's a Long Way to the Top (If You Wanna Rock 'N' Roll)

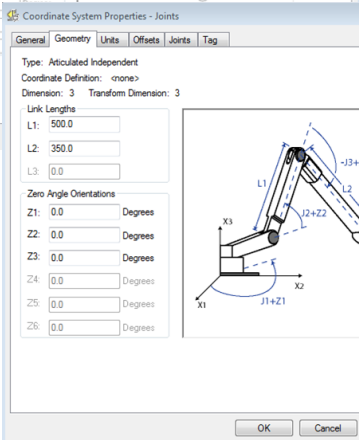
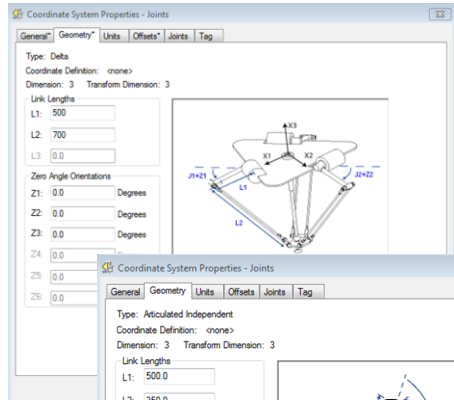


2003

- RSLogix5000 V12
- **Cartesian Coordinate system**
- Coordinated motion instructions
- Path Blending Between Motion Segments (termination types)

- RSLogix5000 V16
- **Kinematics support:** Articulated Robot, SCARA, Delta

2007



- RSLogix5000 V18
- **Motion over Ethernet**



2010



MagneMotion

2015

- **iTRAK:** Intelligent Track System

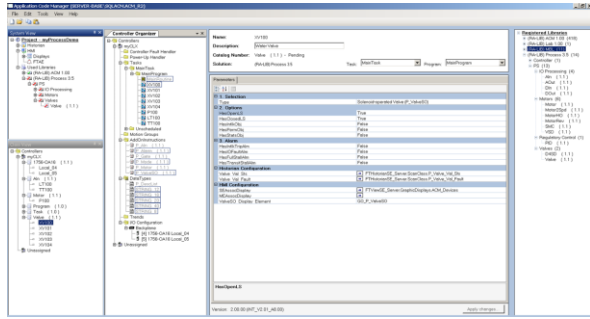
2016

- Intelligent Conveyor System





# It's a Long Way to the Top (If You Wanna Rock 'N' Roll)

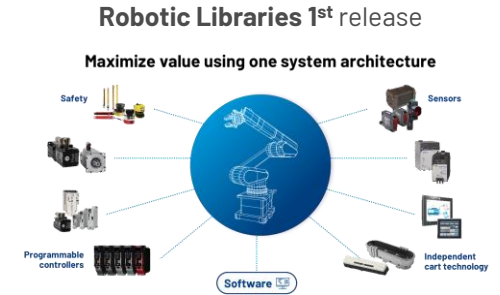


Studio 5000 Logix Designer V31

**Increased Kinematics** support in:

- 6 axis linear interpolation
- 3-5 axis Delta geometry support
- Tool Center Point (TCP) programming with dynamic tool offsets

**Advanced Safety** for Kinetix Drives



2018

- **Application Code Manager:** Engineering design productivity tool focused on rapid automation application development leveraging

2019

2019

2022

2023

Coordinate System Properties - Cartesian

Coordinate	Axis Name	Coordination Mode
0	X	Primary
	Y	Primary
	Z	Primary

Coordinate System Properties - Joints

Type: Articulated Dependent  
Coordinate Definition: J1J2J3J6  
Dimension: 4 Transform Dimension: 4

Link Lengths

L1: 1000.0  
L2: 750.0  
L3: 0.0

Zero Angle Orientations

Z1: 0.0 Degrees  
Z2: 0.0 Degrees  
Z3: 0.0 Degrees  
Z4: 0.0 Degrees  
Z5: 0.0 Degrees  
Z6: 0.0 Degrees

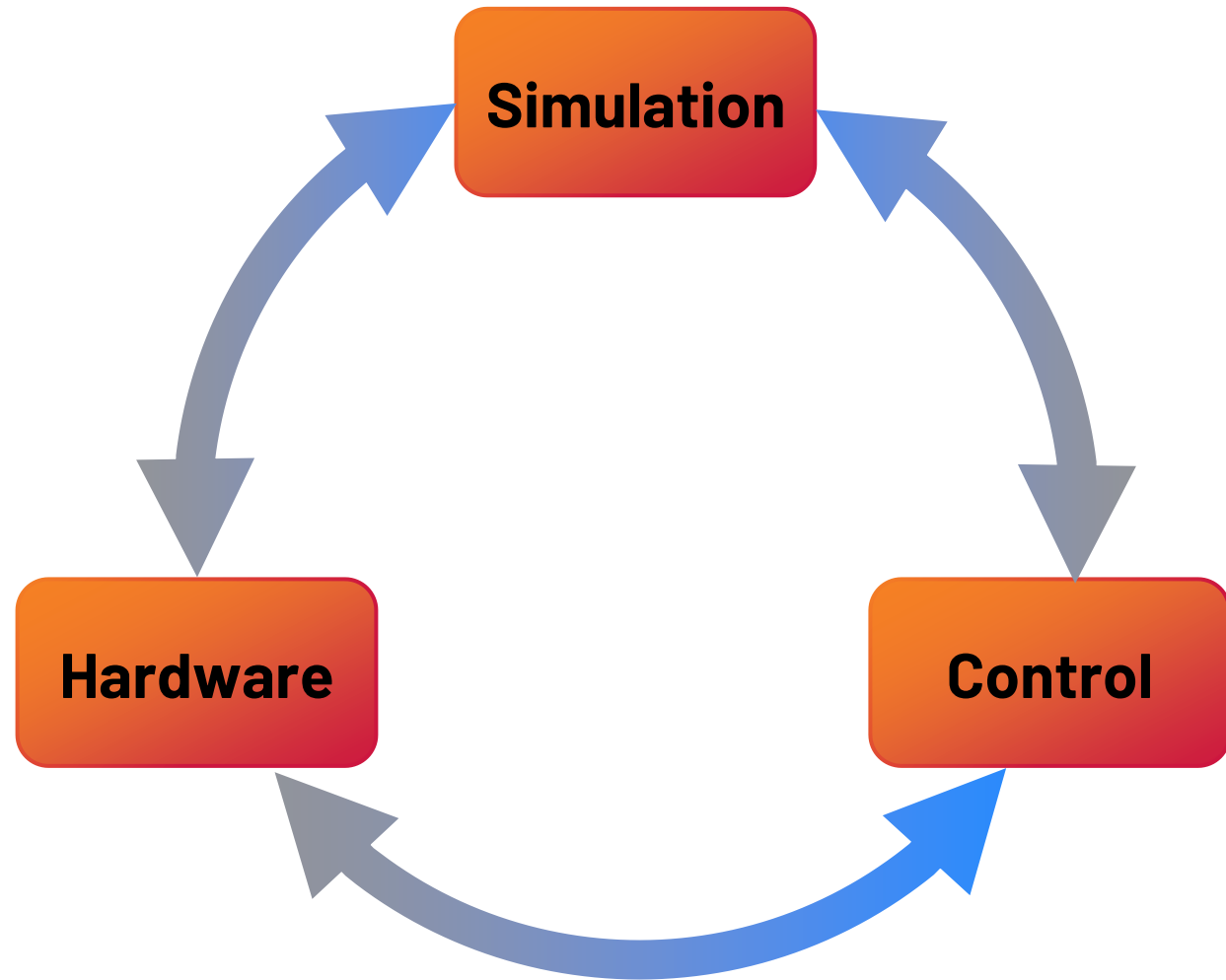


# Simplifying Customer Experience and Unlocking Value with Digital Tools



# | Hardware and Software Convergence

Convergence between hardware and software refers to the integration of physical components (hardware) and programs (software) to create more advanced, efficient systems with improved functionality.



# RA | Hardware and Software Convergence

Subtitle goes here or gets deleted

## **HARDWARE:**

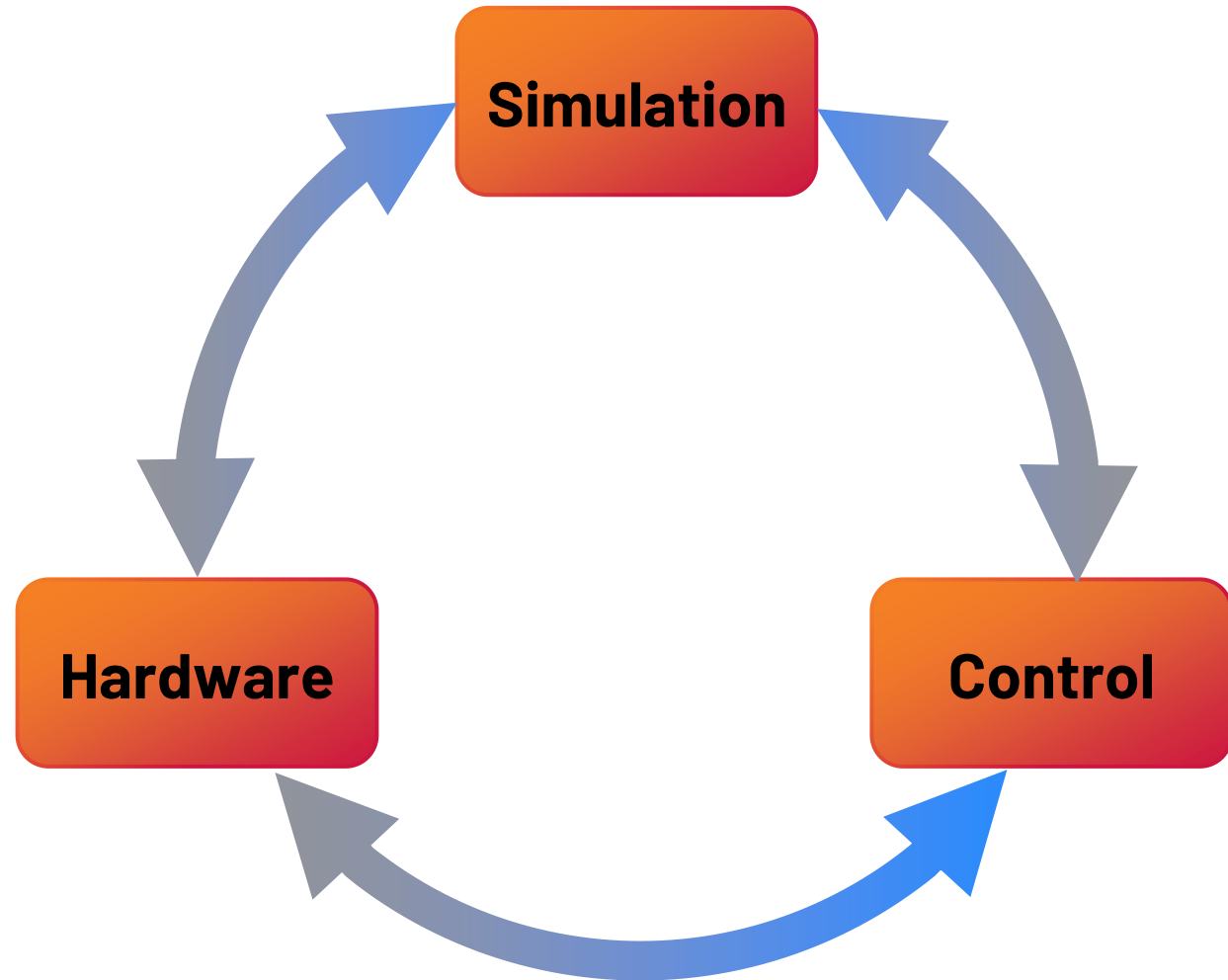
It can be a single component (an IO module for example) scaling up to an entire mechatronic system

## **CONTROL:**

Corresponding Control Module (Software) that address the hardware module

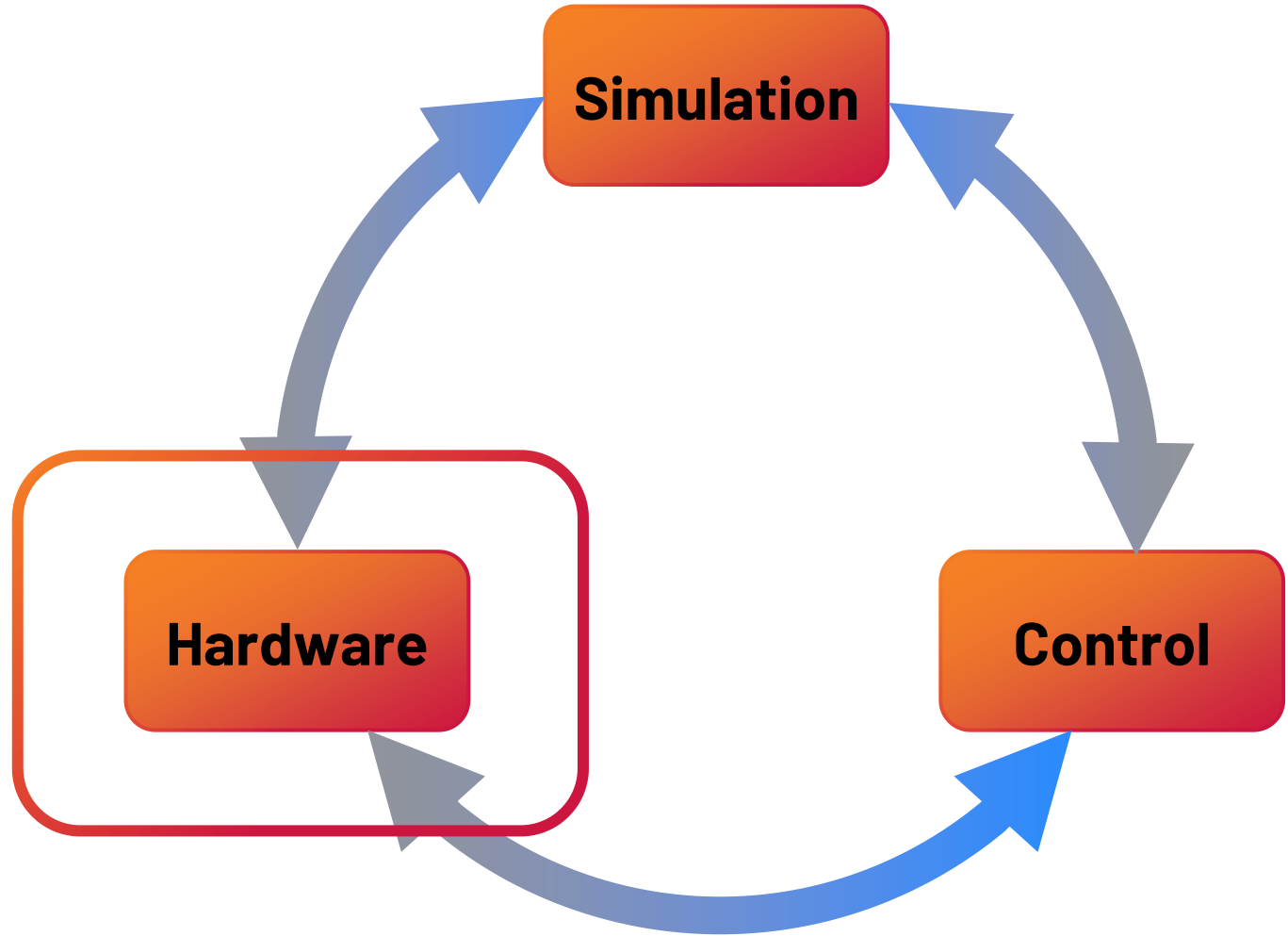
## **SIMULATION:**

Set of Tools to allow from device simulation up to system simulation, including controller emulation



# RA | Hardware and Software Convergence

A basic I/O Module



# RA | Hardware and Software Convergence

Device Handler as a Software Commodity

Faster Delivery

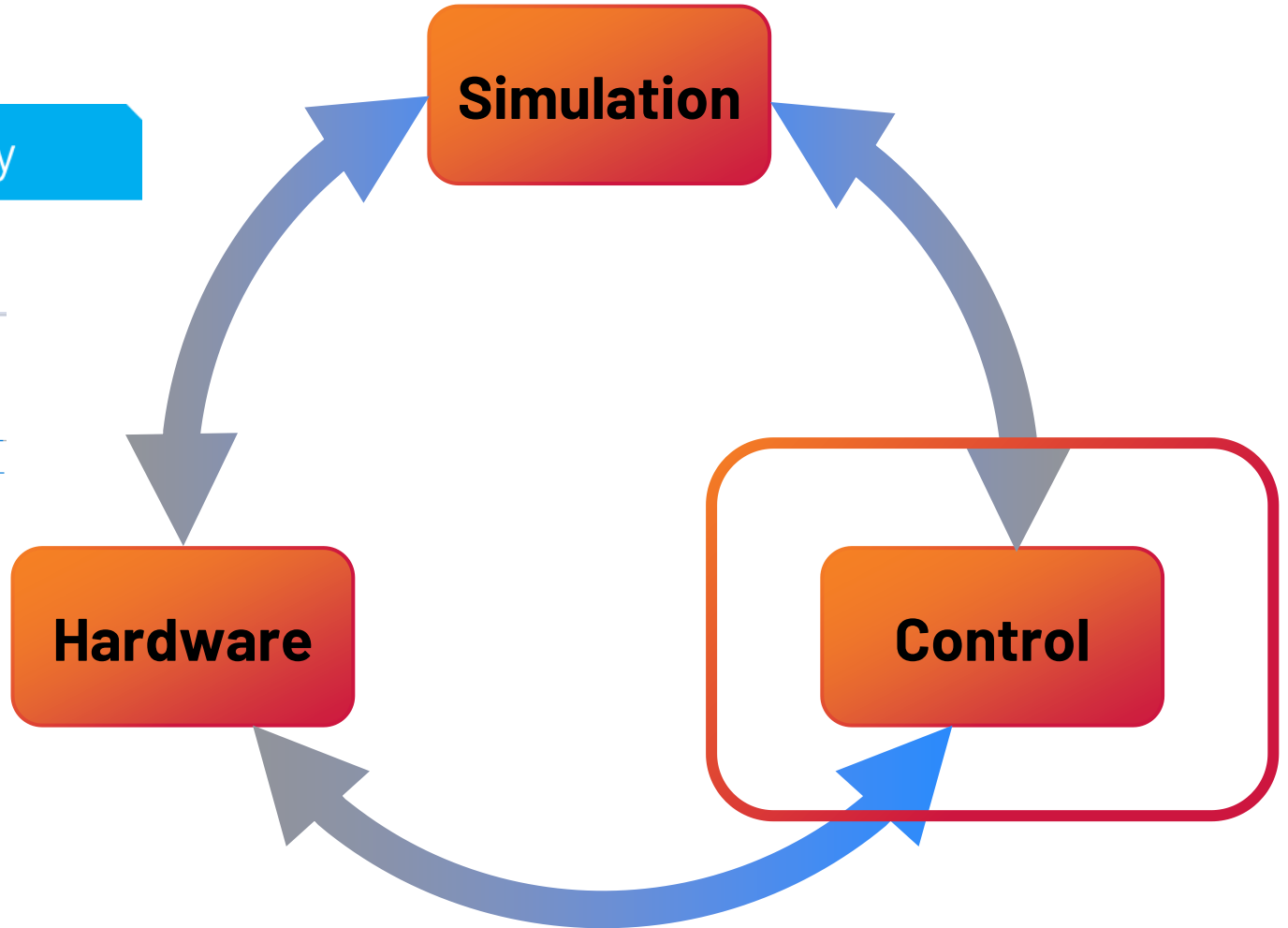
Improve Consistency

Supported

Device Object for  
1732E Series Digital  
16-channel Input  
Modules with  
Diagnostics (Example  
1732E-IB16M12DR)

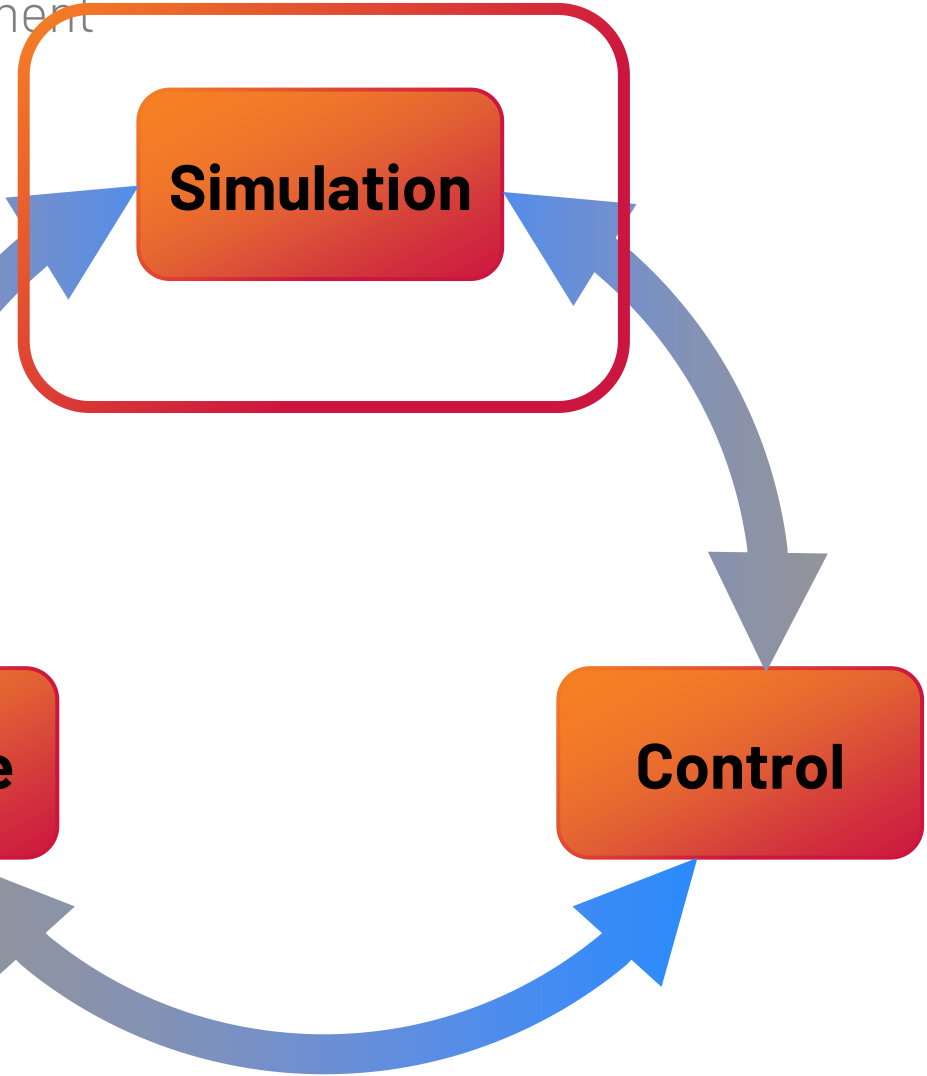
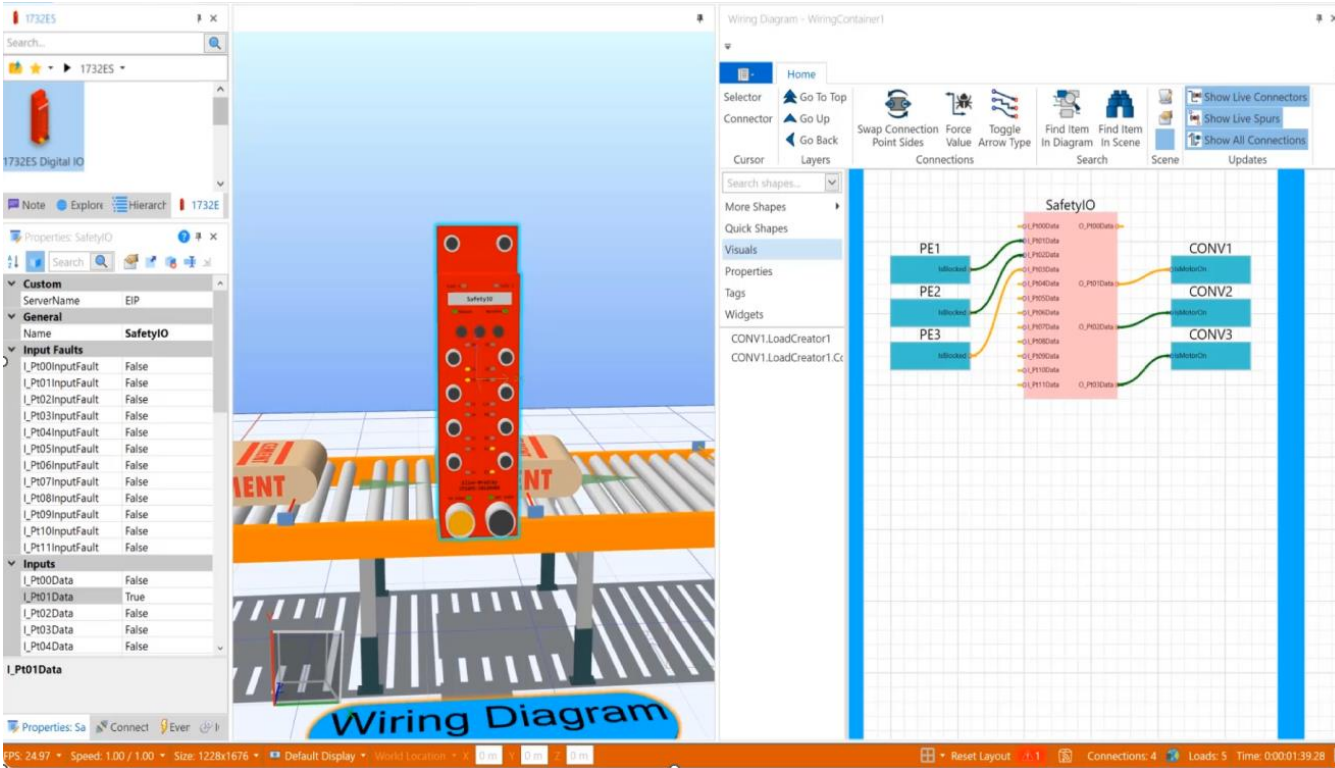
raM_Dvc_1732EIB16_Diag			
raM_Dvc_1732EIB16_Di..._1732ES_Safety	Sts_bNotReady	2#0000_0000	<Sts_Virtual>
Ref_Module	_1732ES		<Sts_Ch00>
Ref_Ctrl_Set	_1732ES_Ctrl_Set		<Sts_Ch01>
Ref_Ctrl_Cmd	_1732ES_Ctrl_Cmd		<Sts_Ch02>
Ref_Ctrl_Sts	_1732ES_Ctrl_Sts		<Sts_Ch03>
Inf_Lookup	_1732ES_infLookup		<Sts_Ch04>
Ref_Ch00	_1732_Ch[0]		<Sts_Ch05>
Ref_Ch01	_1732_Ch[1]		<Sts_Ch06>
Ref_Ch02	_1732_Ch[2]		<Sts_Ch07>
Ref_Ch03	_1732_Ch[3]		<Sts_Ch08>
Ref_Ch04	_1732_Ch[4]		<Sts_Ch09>
Ref_Ch05	_1732_Ch[5]		<Sts_Ch10>
Ref_Ch06	_1732_Ch[6]		<Sts_Ch11>
Ref_Ch07	_1732_Ch[7]		<Sts_Ch12>
Ref_Ch08	_1732_Ch[8]		<Sts_Ch13>
Ref_Ch09	_1732_Ch[9]		<Sts_Ch14>
Ref_Ch10	_1732_Ch[10]		<Sts_Ch15>
Ref_Ch11	_1732_Ch[11]		
Ref_Ch12	_1732_Ch[12]		
Ref_Ch13	_1732_Ch[13]		
Ref_Ch14	_1732_Ch[14]		
Ref_Ch15	_1732_Ch[15]		

Low-Code  
Reusable Content



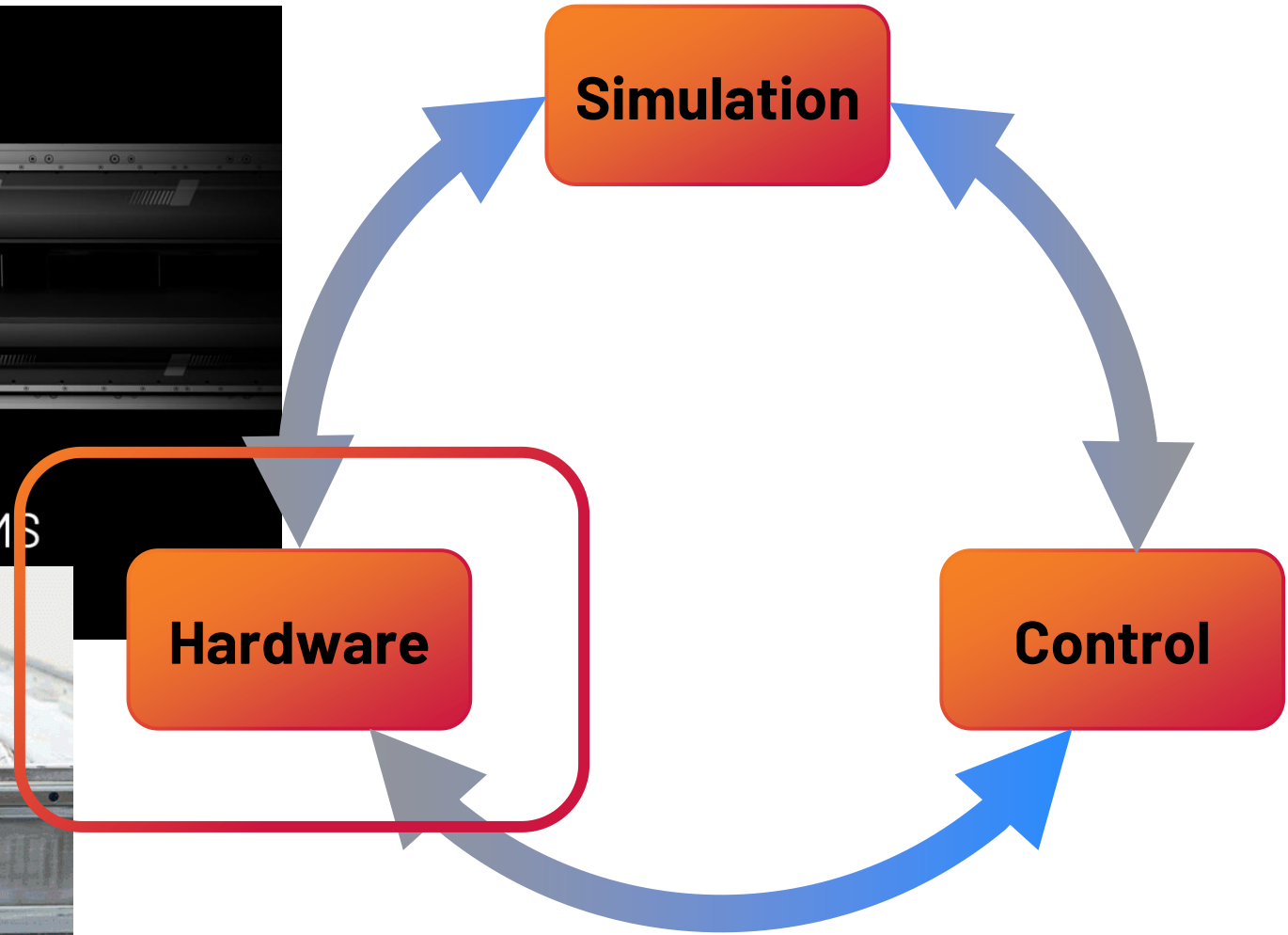
# RA | Hardware and Software Convergence

Simulation of the component in its working environment



# RA | Hardware and Software Convergence

From a device to a mechatronic system



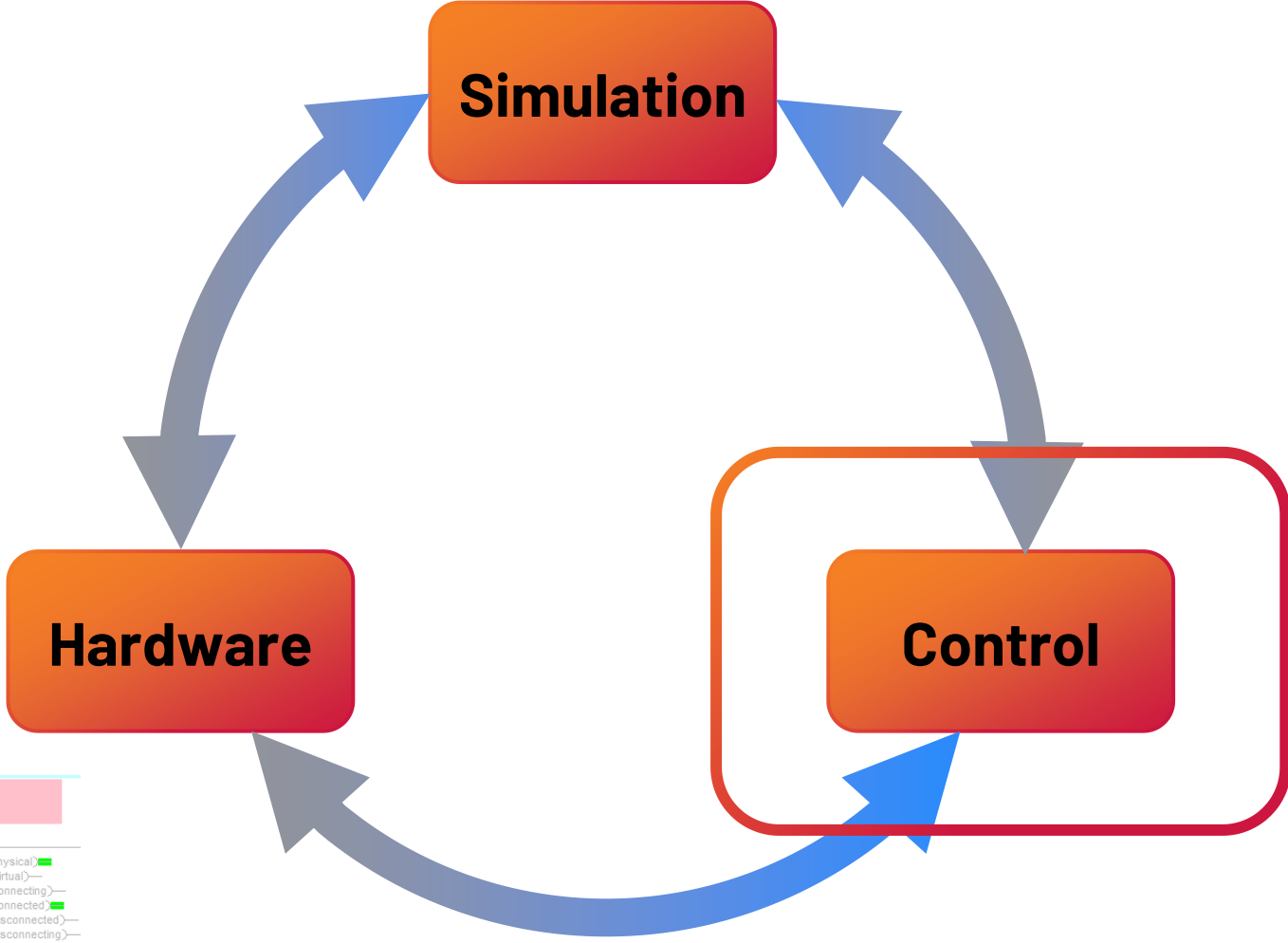
# RA | Hardware and Software Convergence

Software counterpart: configuration vs programming

## System Device Handler

```
ITRAK5750_DH - _00_X_DeviceObjectManagement x
69 /**
70  * Verify the state of the Device Objects
71  */
72
73
74 //Hndl.bEnergized := 0;
75 _bAnyEnergized := 0;
76 //Sections
77 _nAllAvailable := 0; //Check for all available
78 _nAllEnergized := 0; //Check for AllMovers and AllSections Energized status
79 _nAllIdle := 0; //Check for all idle
80 _nAllConnected := 0; //Check for all connected
81 _bAllVirtual := 1;
82 _bAllPhysical := 1;
83
84 for _iSection := 0 to nMaxNumberDoSections - 1 do
85   if Section[_iSection].Status.Energized then
86     _nAllEnergized := _nAllEnergized + 1;
87     _bAnyEnergized := 1; //If any device is energized, notify the bEnergized.
88   end_if;
89   if stSectionDeviceObjects[_iSection].Sts_Available then
90     _nAllAvailable := _nAllAvailable + 1;
91   end_if;
92   if stSectionDeviceObjects[_iSection].Sts_Connecting then
93     _nAllIdle := _nAllIdle + 1;
94   end_if;
95 end_for;
```

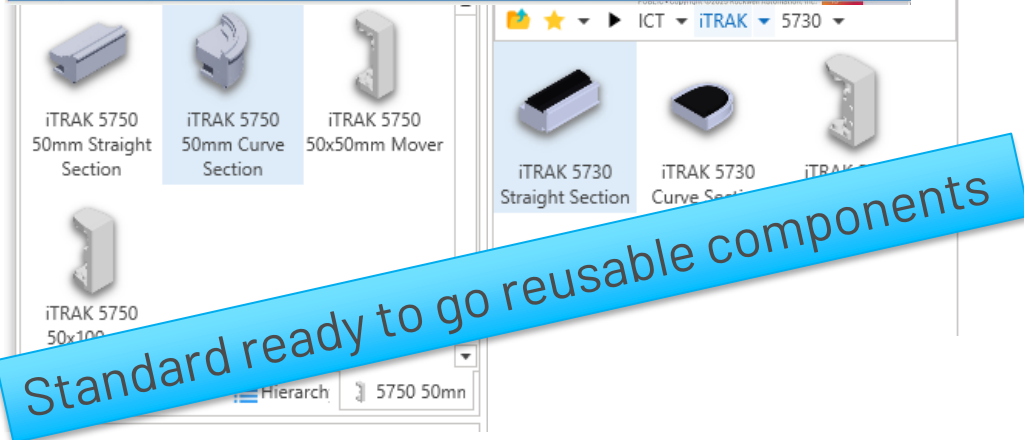
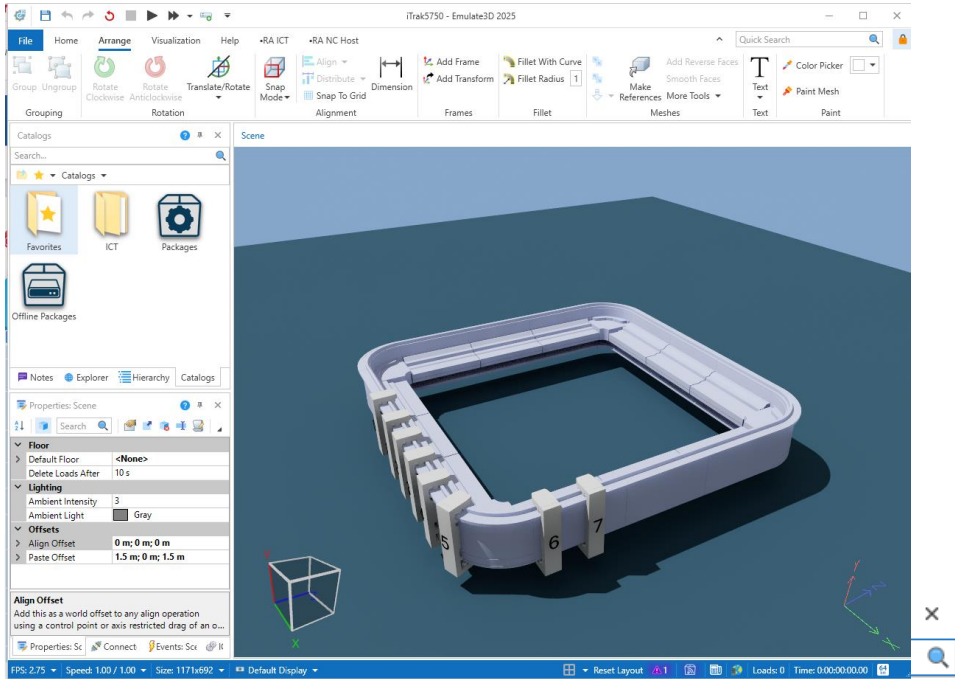
Low-Code Reusable Content



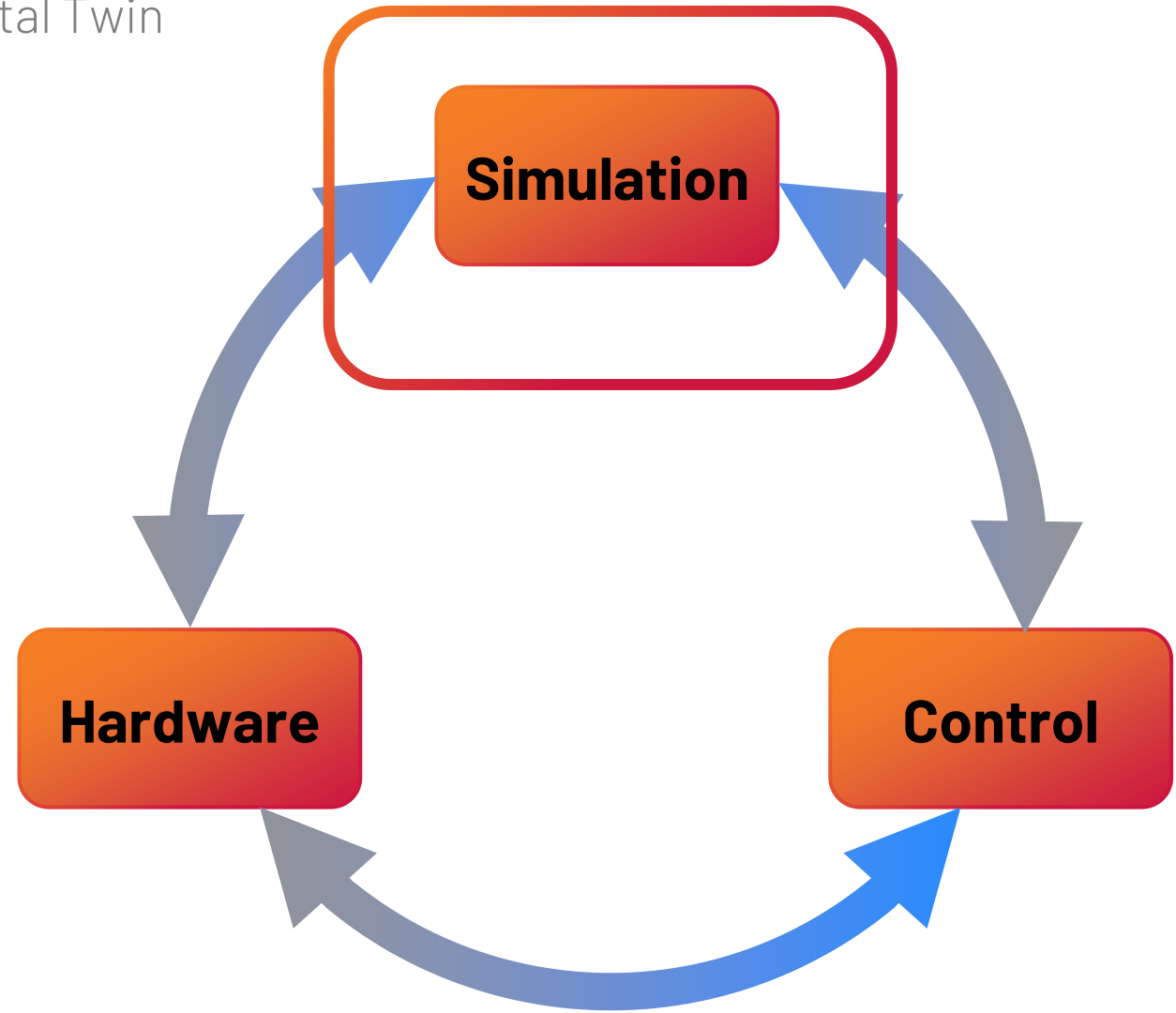
## Sub-Systems Device Handlers

# RA | Hardware and Software Convergence

Design and test application using Digital Twin

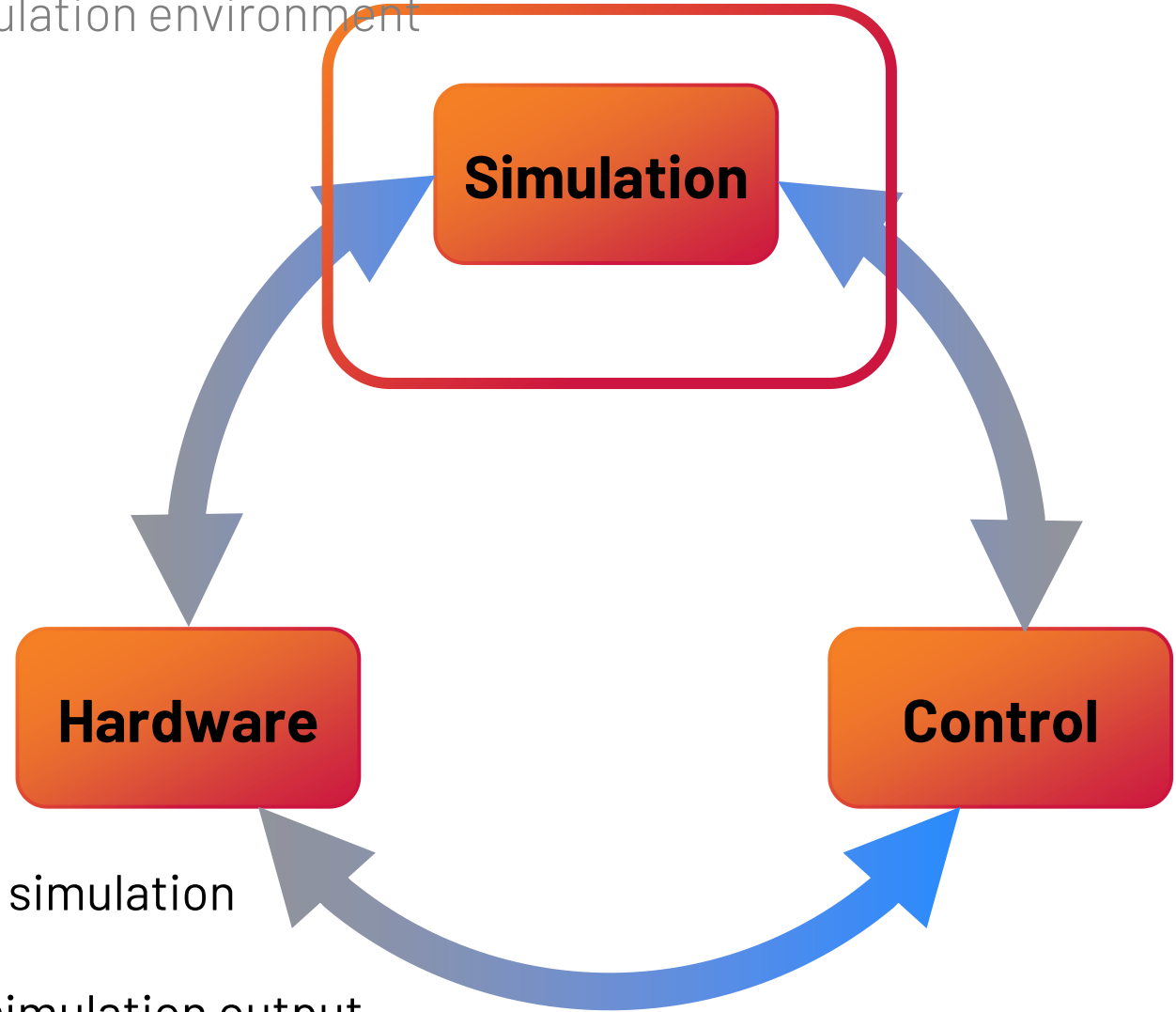
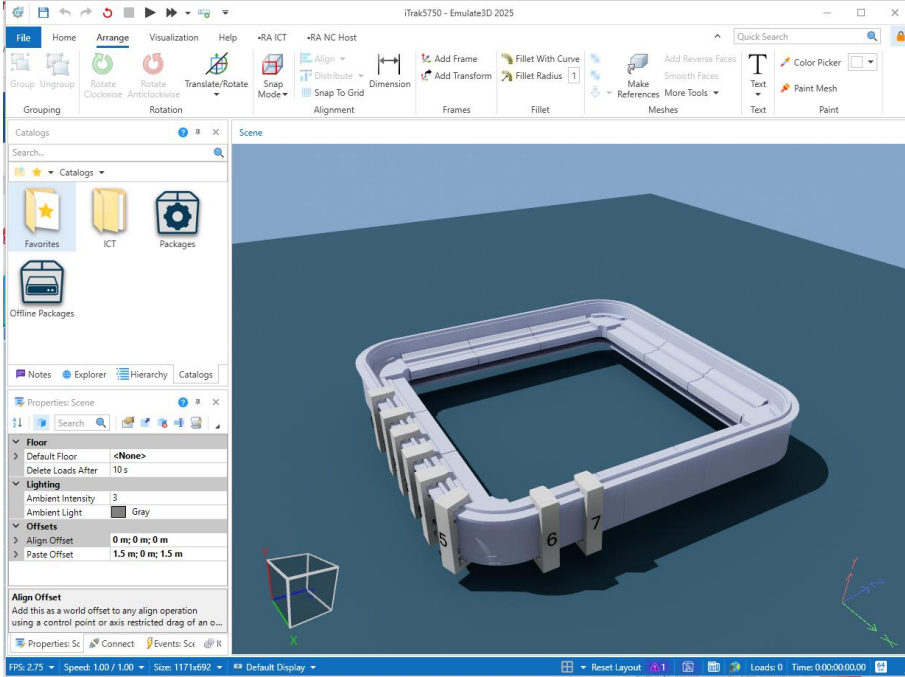


Standard ready to go reusable components



# RA | Hardware and Software Convergence

The other way around: Start from simulation environment



- Start from requirements
- Use simulation to **validate** application
- **Generate Application Code** directly from simulation environment
- **Select Appropriate Hardware** based on simulation output

# Unified Robot Control



expanding human possibility<sup>®</sup>

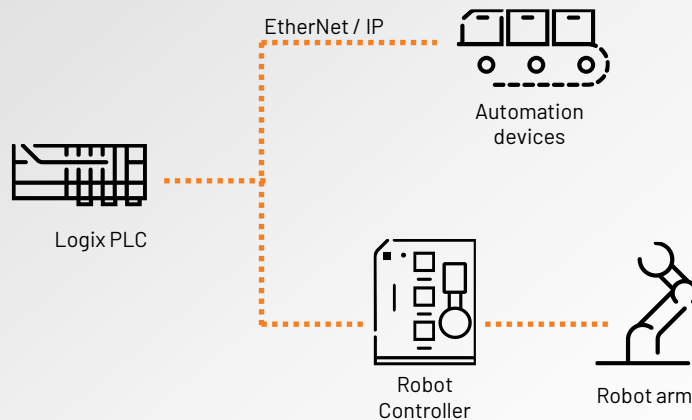


# Two paths to getting more from your robots

## Integrated Robotics

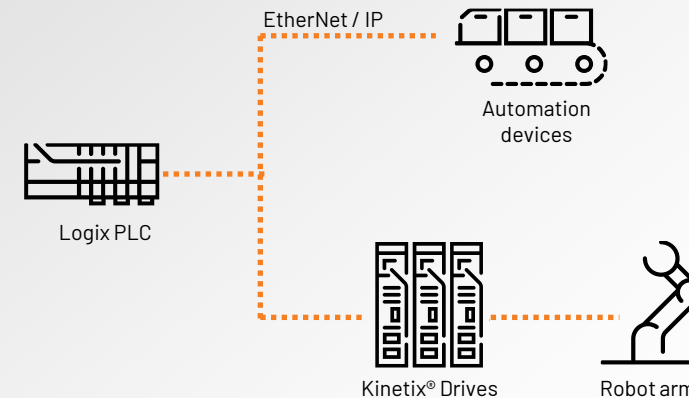
### Ethernet/IP Connected

Connecting a dedicated robot controller to a Logix controller via the standard, unmodified Ethernet protocol. Robot programs are written in either the robot vendor's programming environment or via Studio 5000® robot integration features.



### Unified Robot Control

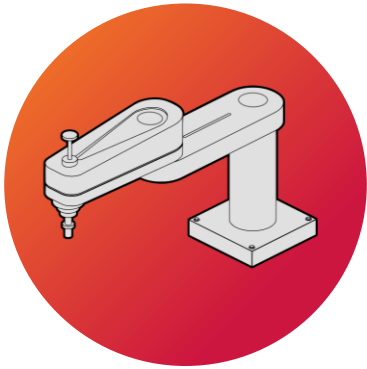
- Use of a Logix controller and Kinetix® drives to directly control a robot arm.
- The Logix controller hosts the robot kinematics, and full program is made in Studio 5000® to direct all robot movement without the use of a dedicated robot controller.





# Supported Geometries

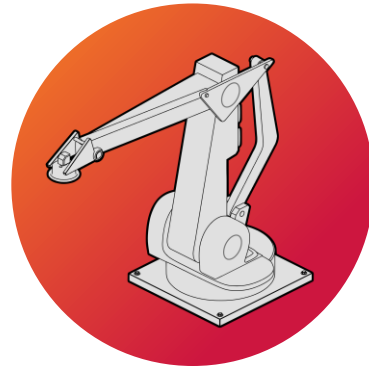
With Robotics Libraries



**SCARA**

## Typical Applications

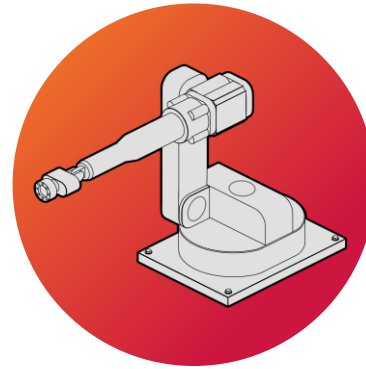
- Primary packaging
- Assembly
- Kitting
- Inspection



**Articulated  
Dependent**

## Typical Applications

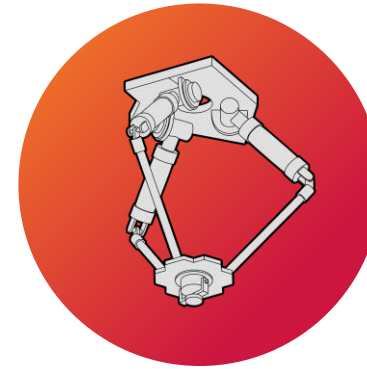
- Palletizing
- Depalletizing



**Articulated  
Independent**

## Typical Applications

- Machine tending
- Secondary packaging
- Material handling
- Body assembly
- Welding



**Delta**

## Typical Applications

- Packaging
- Packing
- Kitting and assembly
- Picking and placing



**Cartesian**

## Typical Applications

- Plastic molding
- Sealing
- Laser welding
- Pressing

# Partnership network



Regions:

- Asia Pacific

Supported architectures:

- Unified robot control



Regions:

- North America
- Europe, Middle East, Africa

Supported architectures:

- Unified robot control



Regions:

- Global

Supported architectures:

- Unified robot control
- Ethernet/IP connected



Regions:

- North America

Supported architectures:

- Ethernet/IP connected



Regions:

- North America
- Latin America
- Europe, Middle East, Africa

Supported architectures:

- Ethernet/IP connected



Regions:

- Global

Supported architectures:

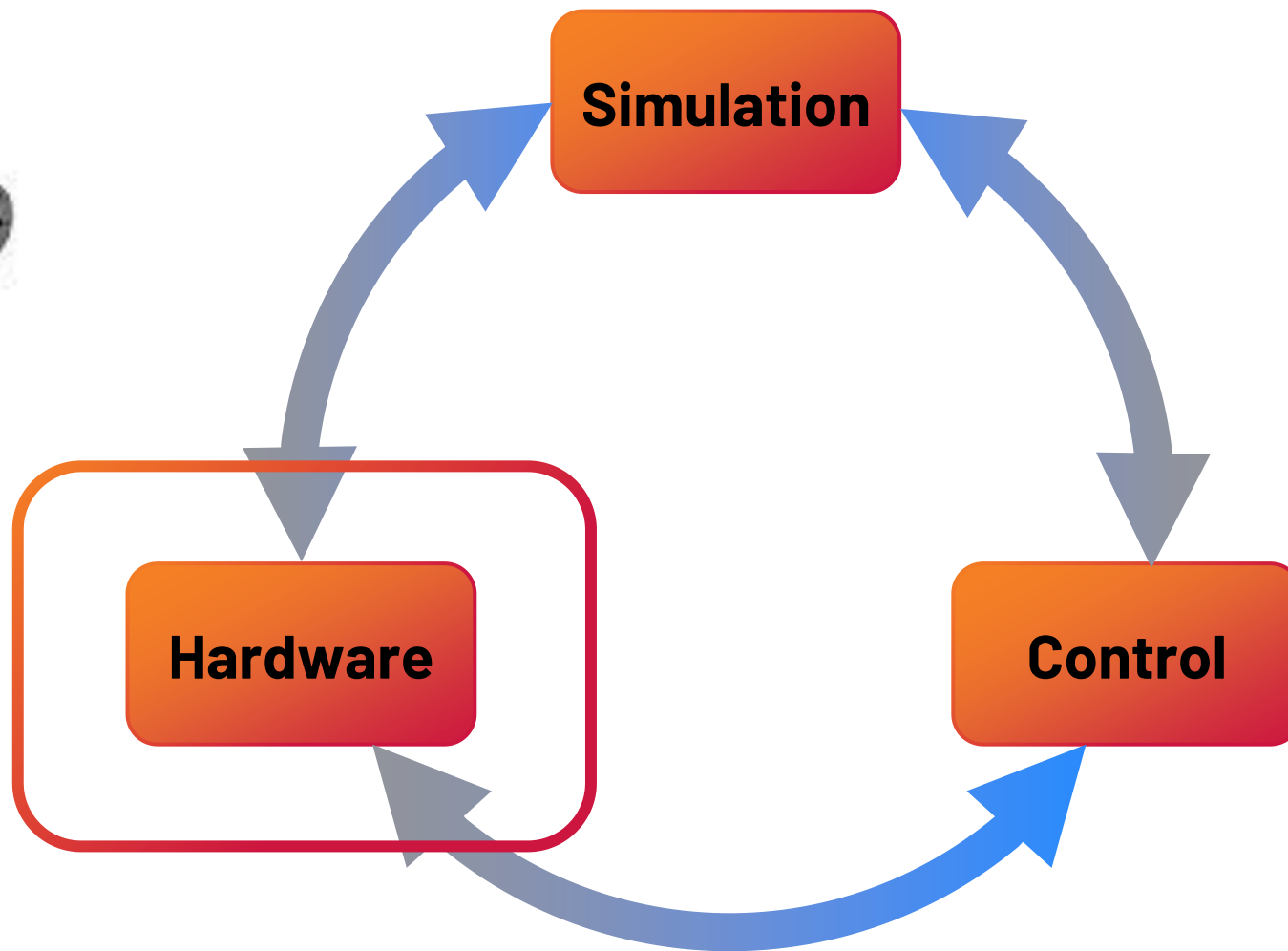
- Ethernet/IP connected

# RA | Hardware and Software Convergence

From a device to a mechatronic system



Select a specific Hardware from Partners Catalog



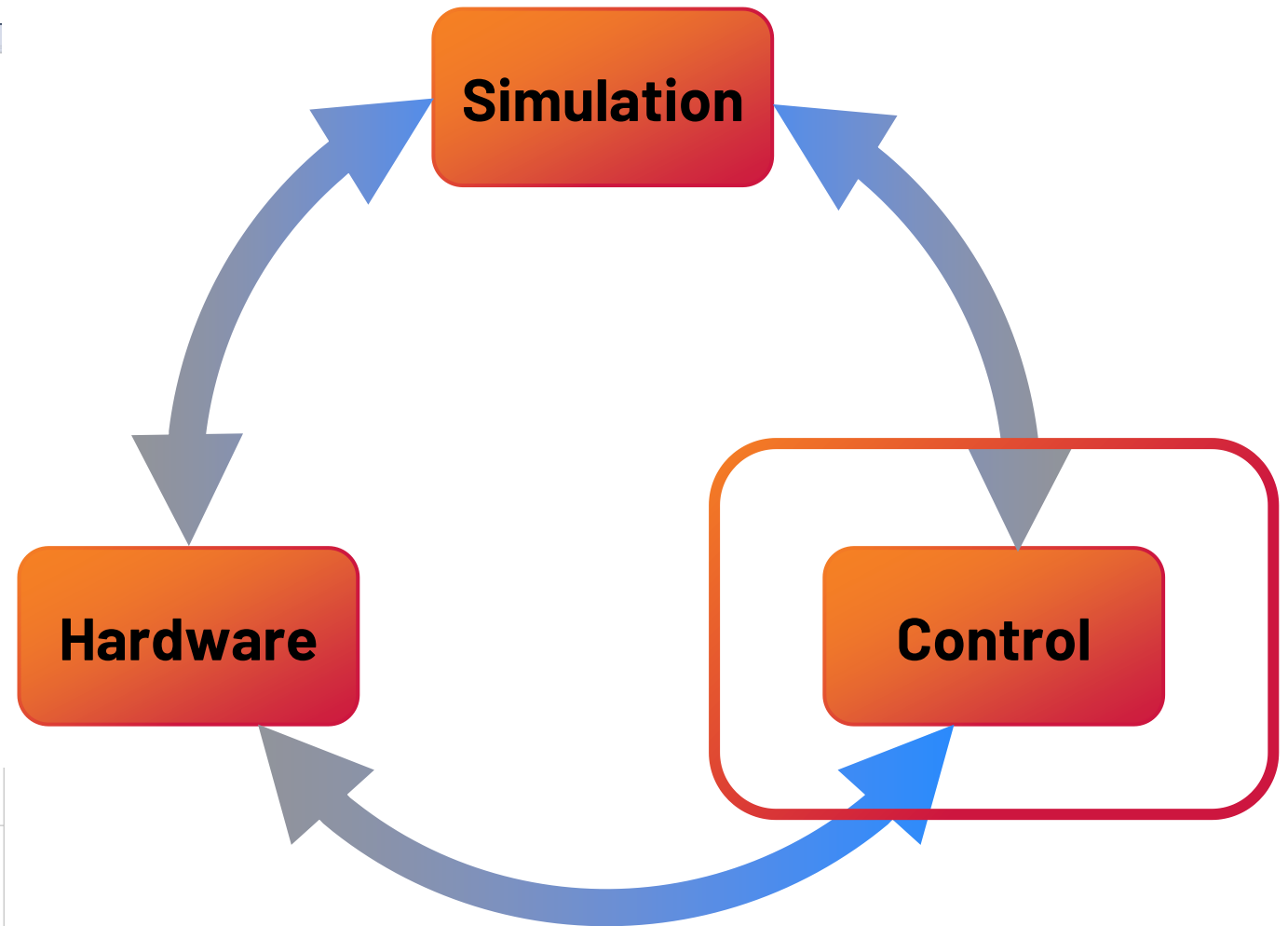
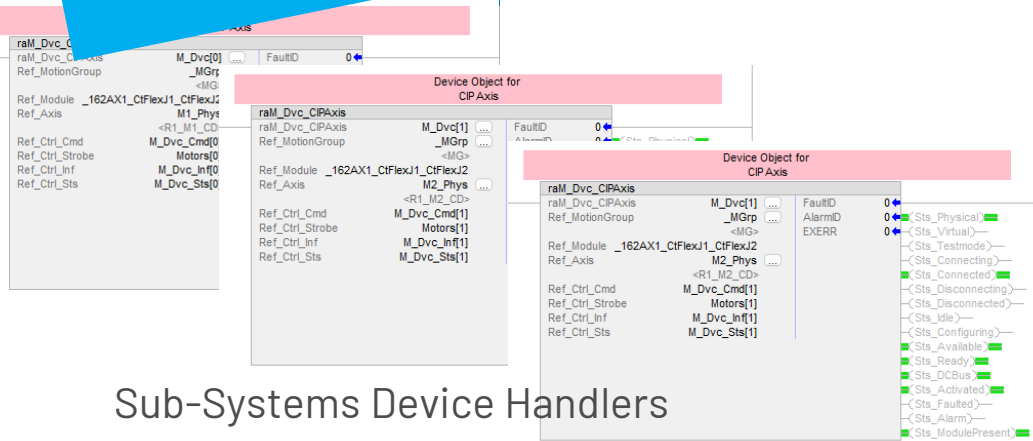
# RA | Hardware and Software Convergence

## Device Handler as a Software Commodity

```
1 // If the system is in a specific state (FS) or requires reinitialization and the hardware is not energized, reset variables
2 if(SIFS or (Cmd.Reinitialize and not HndI.bEnergized)) then
3   stStateMachine.State := 0; // Reset state machine state
4   _nStateLast := 1; // Set last state to 1
5   HMI_Cmd_Mode := 1; // Set HMI command mode to 1
6   _TimerStateMachine.PRE := 500; // Set the timer preset value
7   _nOneshot := 0; // Reset one-shot trigger
8   _nLocalHandlerID := 0; // Reset handler ID
9
10 // Determine the size of the motors and devices arrays
11 SIZE(Motors,0,_nNDINT);
12 SIZE(M_Dvc,0,_nMaxDevices);
13
14 // Ensure the max devices count does not exceed the motor count
15 if _nNDINT < _nMaxDevices then
16   _nMaxDevices := _nNDINT;
17 end_if;
18
19 // Calculate the maximum motor ID
20 _nMotorMaxID := _nMaxDevices - 1;
21 end_if;
22
23 // Clear the reinitialize command
24 Cmd.Reinitialize := 0;
25
26 // Initialize flags for hardware and axes statuses
27 _bAllHardwareModulesConnected := 1;
28 _bAllHardwareModulesReady := 1;
29 _bAllHardwareModulesAvailable := 1;
30 _bAllAxesEnergized := 1;
31 _bAllAxesDeenergized := 1;
32 _bStandstill := 1;
33 _bAllRequestMotorsHomed := 1;
34 _bAllAxesSafetyEnabled := 1;
35 _bAllAxesSafe := 1;
36
37
38
39
40
41
42
43
```

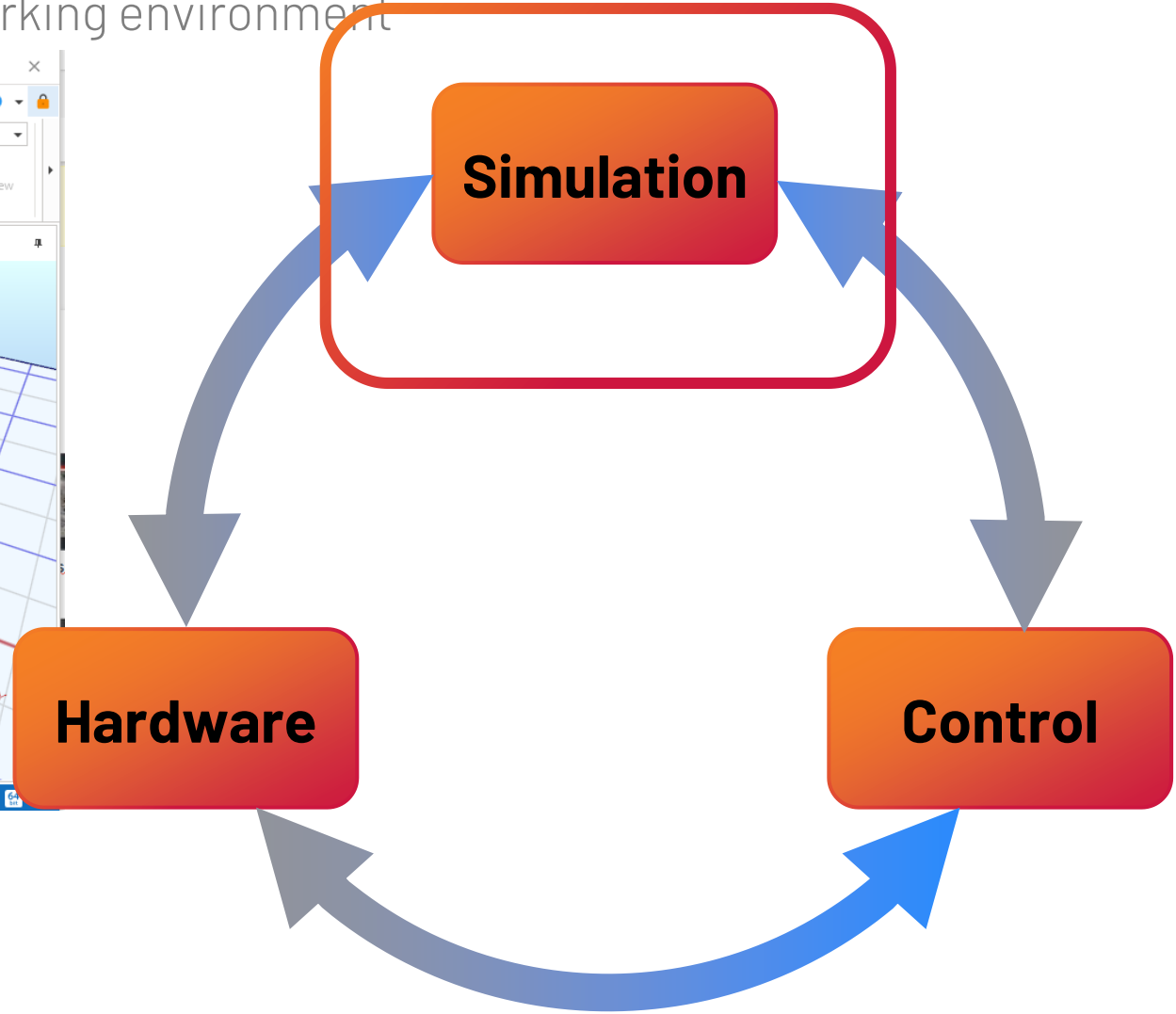
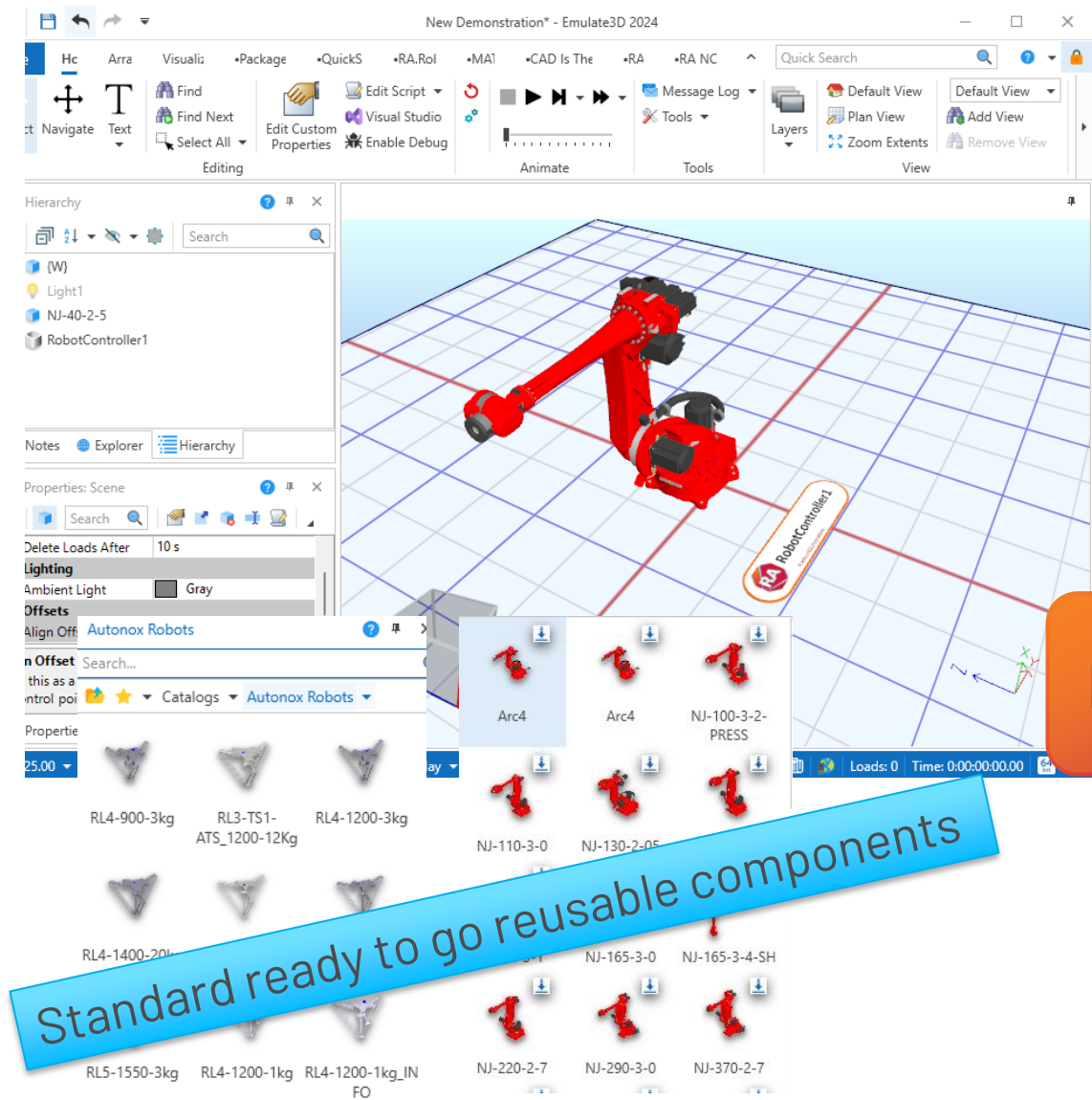
System Device Handler

Low-Code Reusable Content



# RA | Hardware and Software Convergence

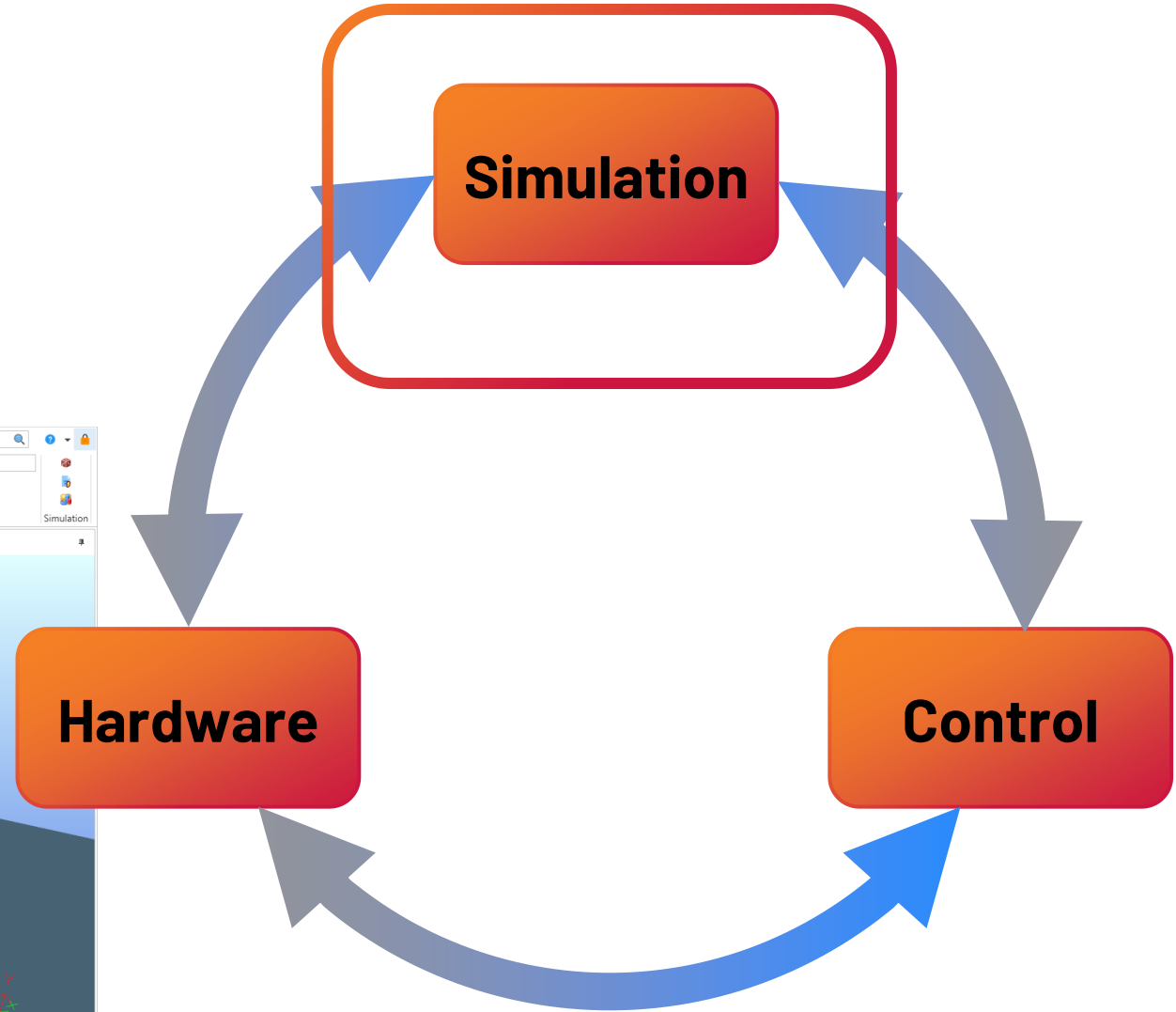
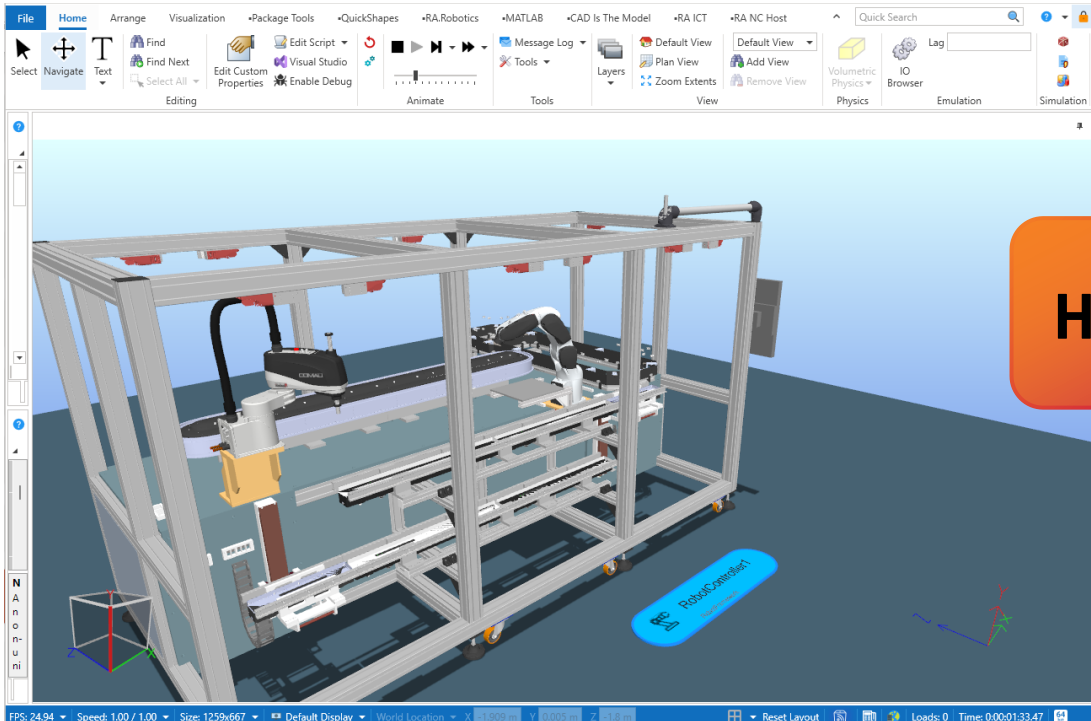
Simulation of the component in its working environment



# RA | Hardware and Software Convergence

Let's place them all together

Combine Virtual and Physical environment  
Leverage Standard Reusable Components  
Validate Application  
Develop tested & robust code  
Save precious time!



# QUESTIONS?



expanding human possibility<sup>®</sup>



Thank you

[www.rockwellautomation.com](http://www.rockwellautomation.com)

