CODESYS® Tech Talk Fall 2025

From Deployment to Optimization: EtherCat Motion Control with IPCs and CODESYS.

Eddie Chau | Business Development, NODKA

Agenda

01
About NODKA

02
PLC to IPC/PAC

03
In the Field

04
Optimization

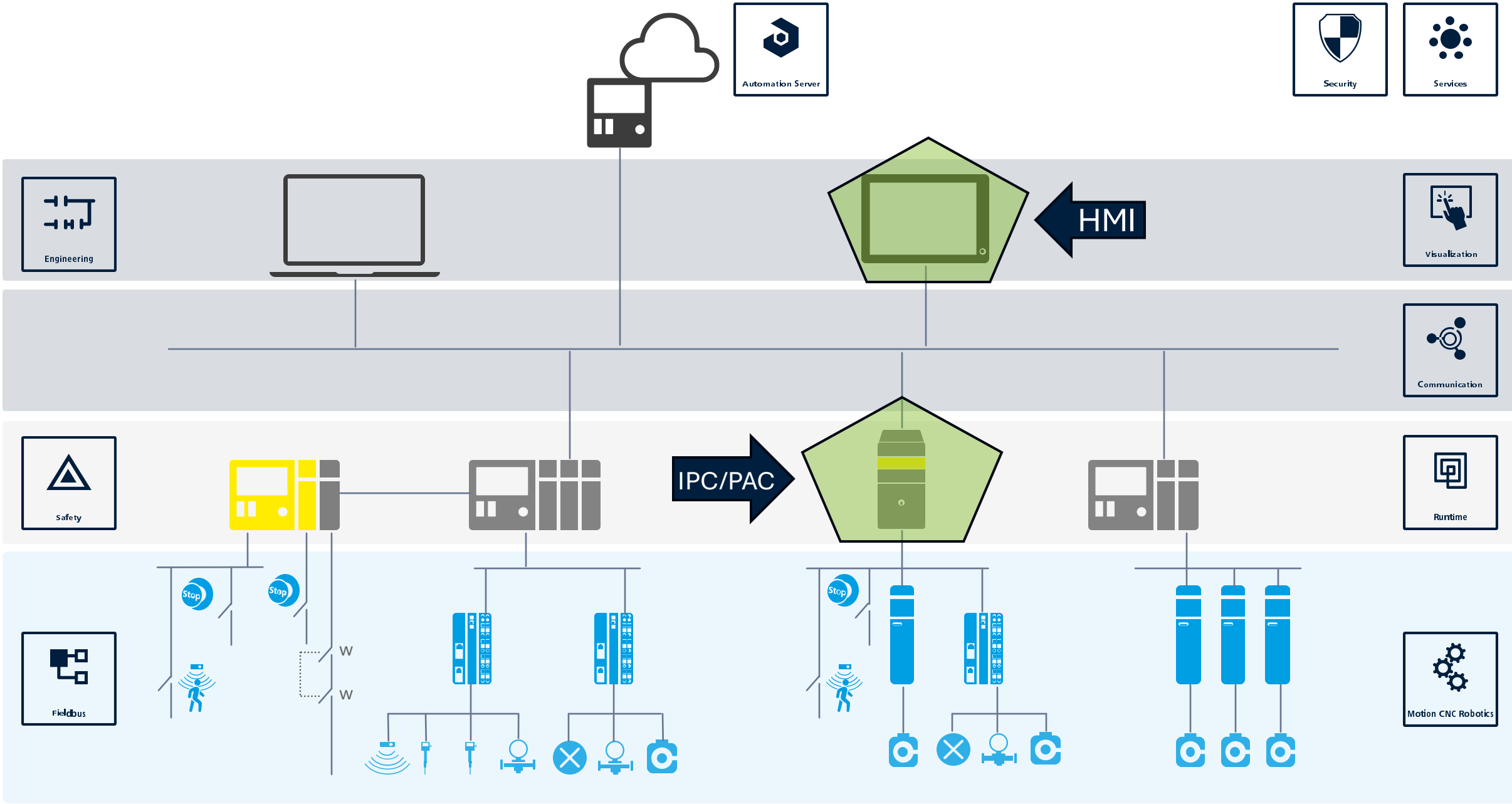
05
Deployment

06
Conclusion

About NODKA

One-stop IPC, Panel PC and HMI Design and Manufacturing

- Specialized IPC platforms focused on Industrial Automation:
- High performance computing power for demanding applications.
- Optimized for software platforms like CODESYS
- We help partners build real-time controllers and application specific by providing customizable hardware.


Automation Server
Security
Services
Engineering
HMI
Visualization
Communication
Safety
IPC/PAC
Runtime
Fieldbus
Motion CNC Robotics
CODESYS®

From PLC to IPC/PAC: A Shift in Control Systems

Traditional PLCs are evolving towards more flexible IPC/PAC solutions

Traditional PLC Setup

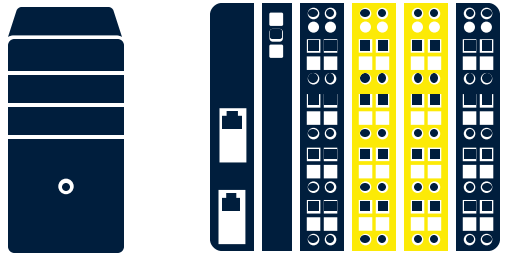
Logic-only focus

Proprietary, closed ecosystems

Single protocol communications

Requires an external gateway for Cloud integration

Modern IPC/PAC Setup (w/ CODESYS)



All-in-one computing and control

IPCs/PACs with CODESYS; Open, standard-based platforms

Multi-protocol, IT/OT ready

Built-in cloud and enterprise connectivity

Application Story: A Foundation for Real-Time Control

The Challenge

Motion controller requires extreme precision, high speed/throughput, high-level language programming

The IPC/PAC

High performance Core I processor

BIOS optimized for real-time control

The Result

CODESYS

EtherCAT

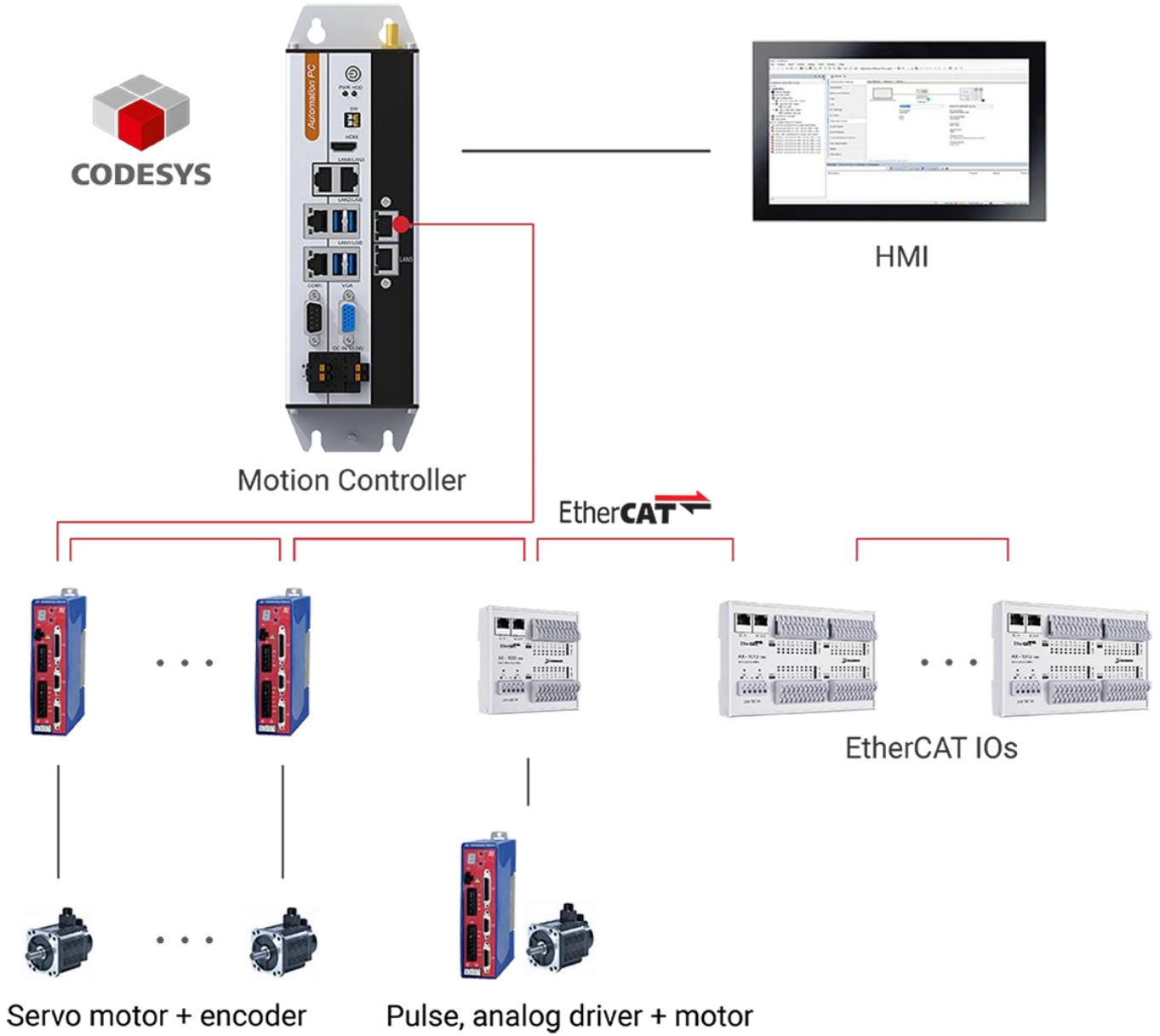
A EtherCAT Master Motion Controller with CODESYS Run-Time

The Application

A high-speed, high-precision Robot for Semiconductor

The Application: A Unified Solution

Motion Control Solution: operating on a three-layer architecture:



BIOS-Level Optimization for Real-Time

Standard IPC

- Not real-time system out-of-the box

VS

Motion Control IPC

- Real-time optimized for CODESYS runtime.
- EtherCAT master enabled

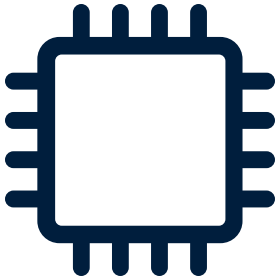
The key feature for this is **Intel's TCC Mode** in the BIOS.

How TCC Mode Helps CODESYS Runtime

Stable CPU Performance

- Disables Power Management
- Disables Turbo Boost

Resource Isolation



- Disables Hyper-Threading
- Configures Cache Allocation Technology (CAT)

System-Level Optimization

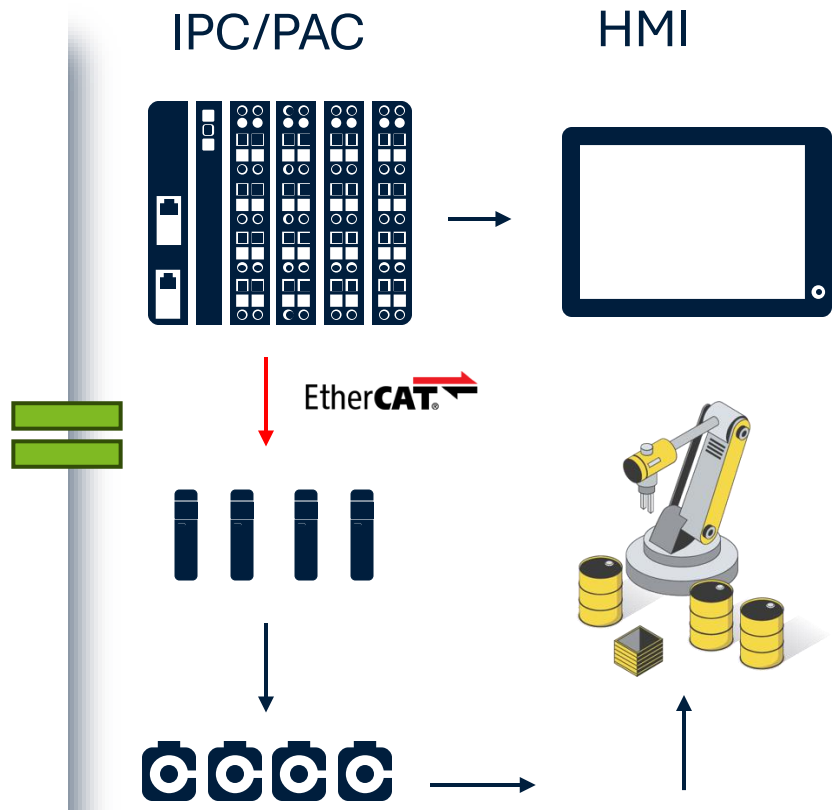
- Minimizes Interrupts
- Configures PTM for EtherCAT

Conclusion: Deployment to Optimization

Real-time optimized Motion Controller

CODESYS

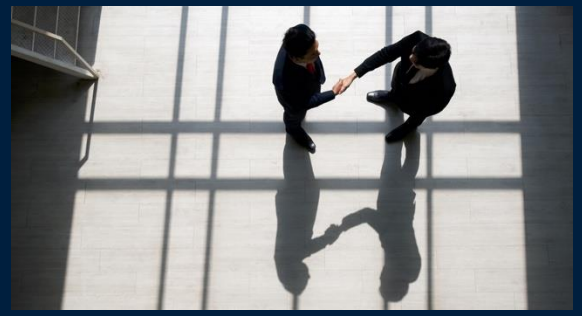
Flexible, standards-based software environment



Complete, optimized motion control solution

The Result: Motion Controller Solution

Embedded multi-axis
EtherCAT network
Motion Controller



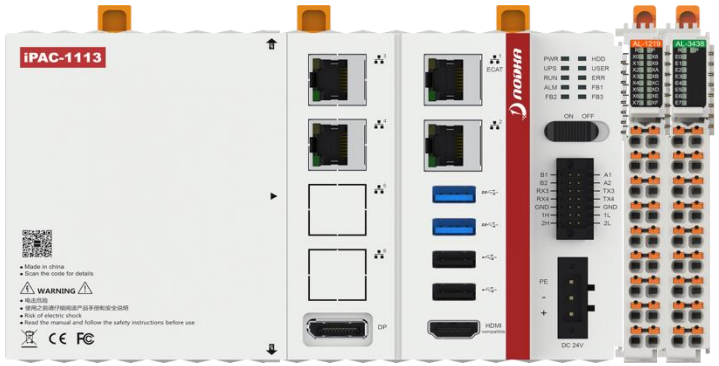
The **integration** of a pre-optimized hardware platform and real-time software provides a **deterministic foundation**.

CODESYS provides the flexible, standardized software for this seamless **integration**.

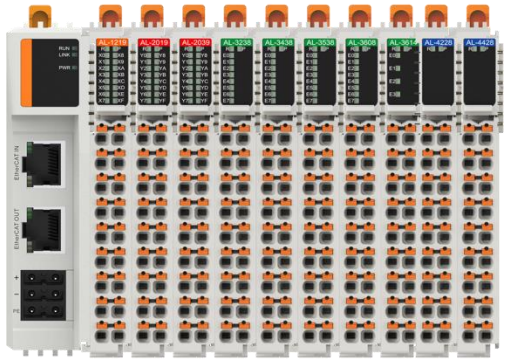
This **integrated approach** enables high-performance EtherCAT motion control, leading to streamlined development and optimized performance.

The Steps: CODESYS + IPC

IPC/IPAC



EtherCAT I/O



CODESYS
Runtime

EtherCAT
Master

EtherCAT
Slave

CODESYS® Tech Talk Fall 2025

Part 2: CODESYS deployment on IPC

A Hands-On Walkthrough

Cesar Hernandez | Technical Support, NODKA

Step-by-Step Overview

01
Installing the Real-Time Ethernet Driver

02
Creating the Project

03
Importing Hardware Descriptions

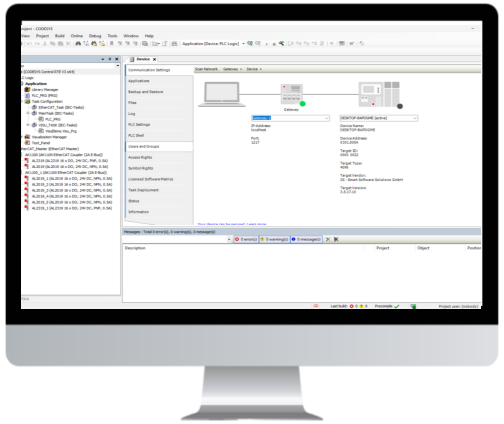
04
Device Discovery

05
Programming the I/O

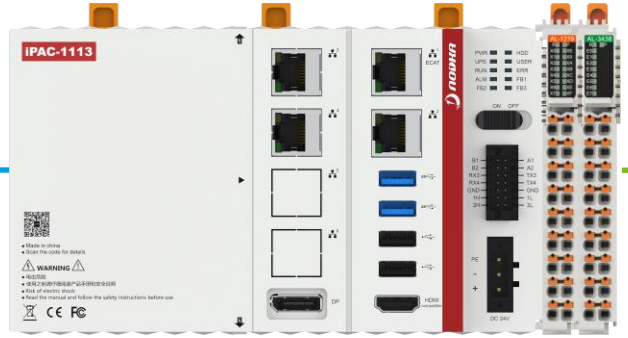
06
Sample Output

The Setup: CODESYS + IPC

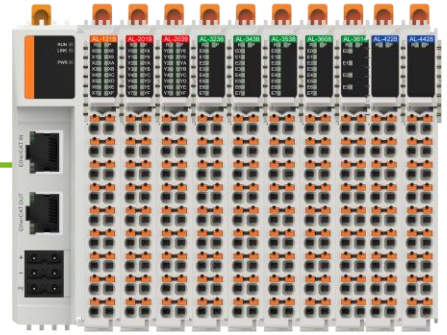
CODESYS IDE



IPC/PAC



EtherCAT I/O



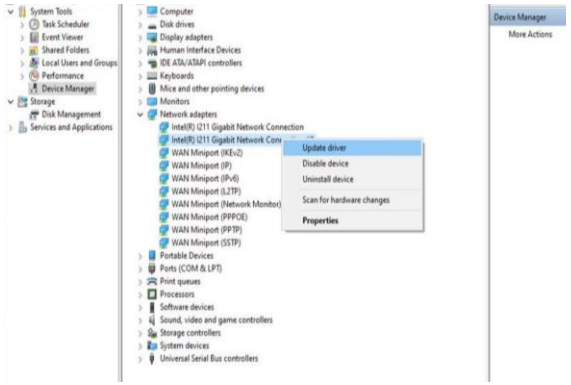
CODESYS Runtime

EtherCAT Master

EtherCAT Slave

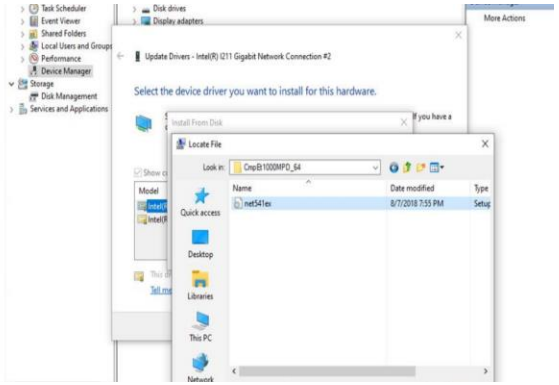
Installing the Real-Time Ethernet Driver

Update Existing Driver



Bypass the Windows network stack for real-time access

Install the CODESYS-specific driver



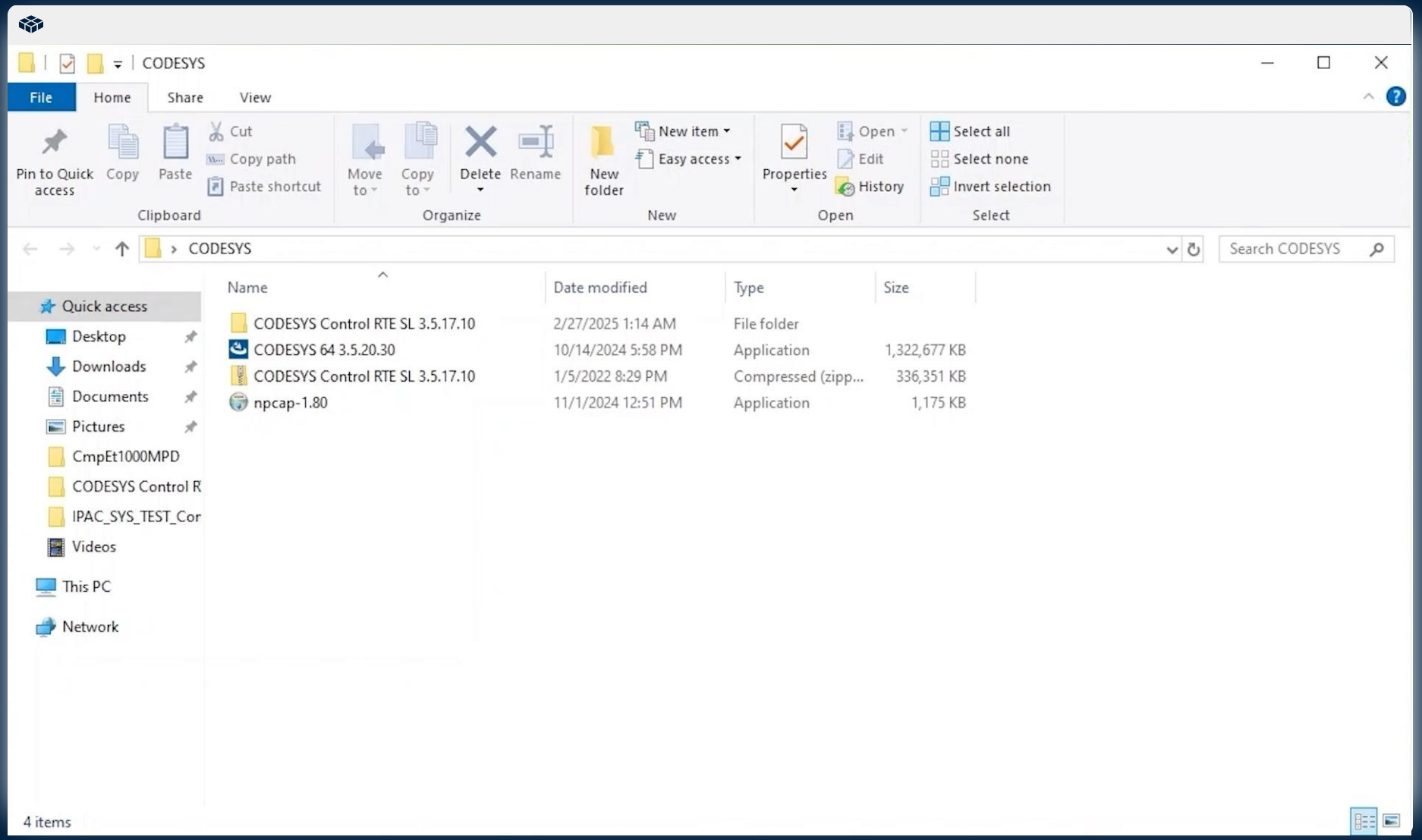
Gain direct access to the hardware via EtherCAT

Select the physical network controller on the IPC

Dedicates a physical port for the EtherCAT Master

Installing the Real-Time Ethernet Driver

Installing the CODESYS Control RTE

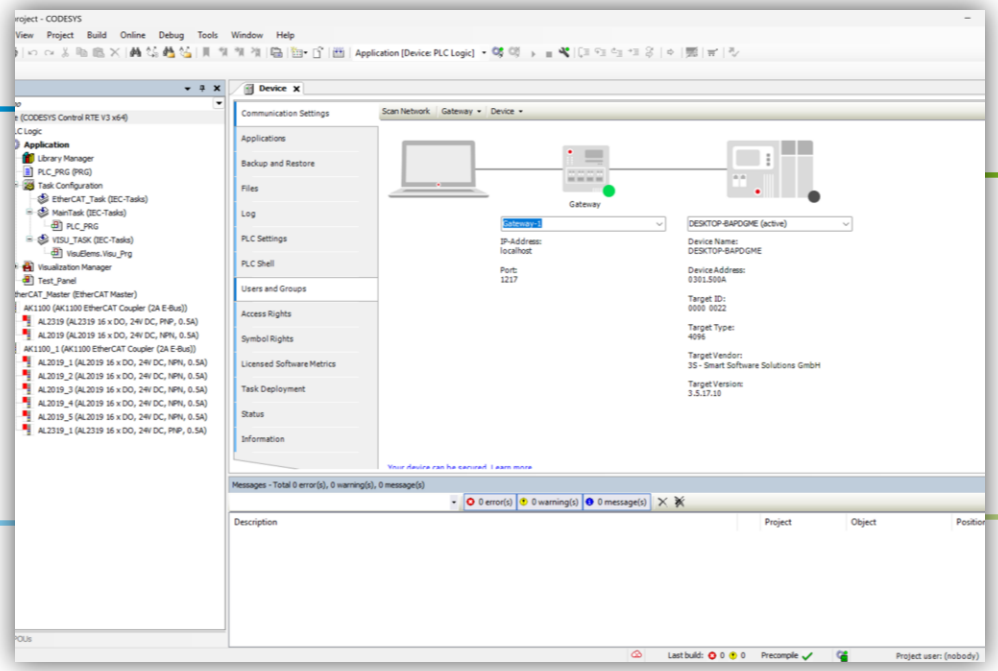


Creating the Project

01

Create the Project

Create a new project on the CODESYS IDE



02

CODESYS IDE → IPC

Connect the CODESYS IDE to the IPC Controller

03

Configure Target

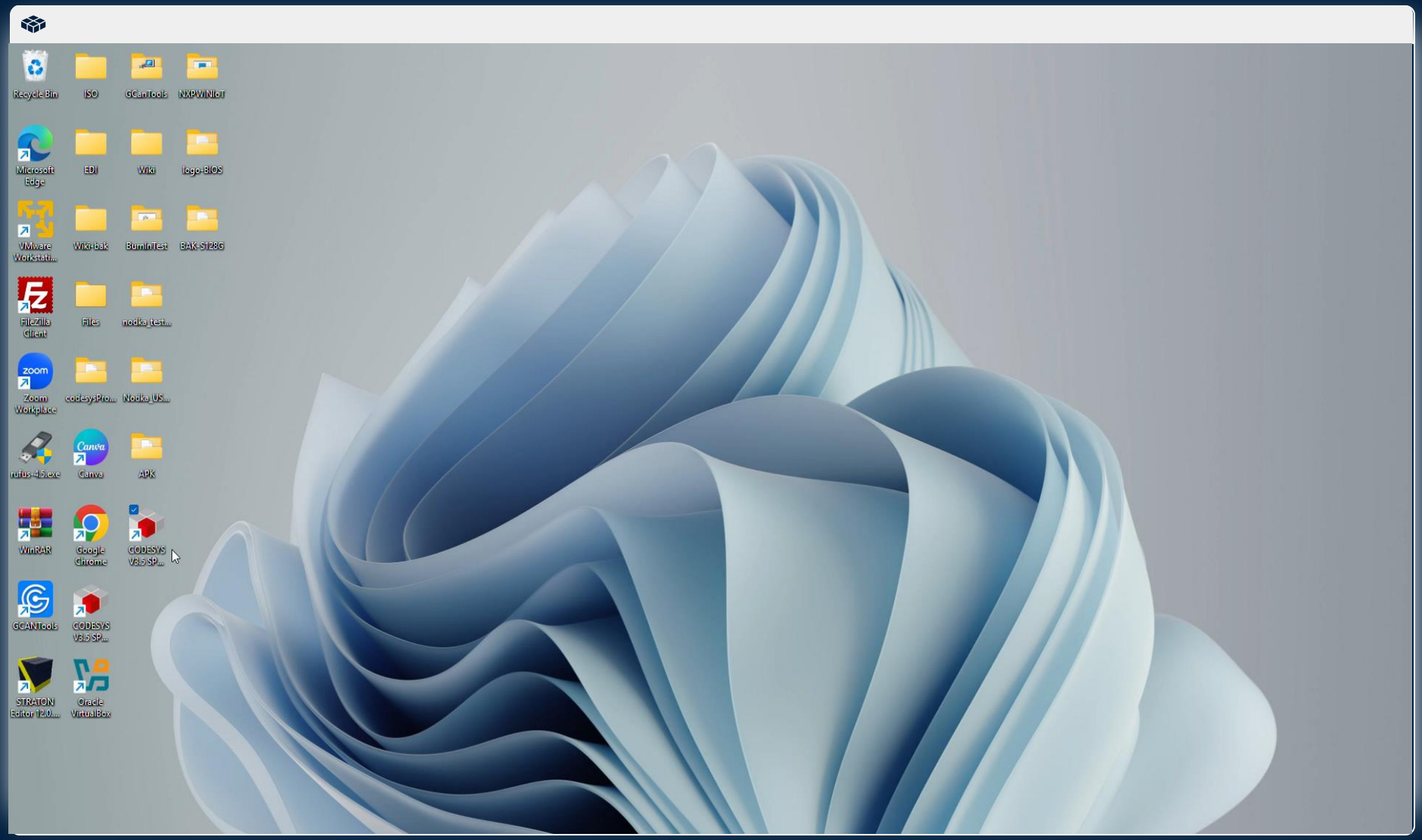
Configure the target device with the newly created project

04

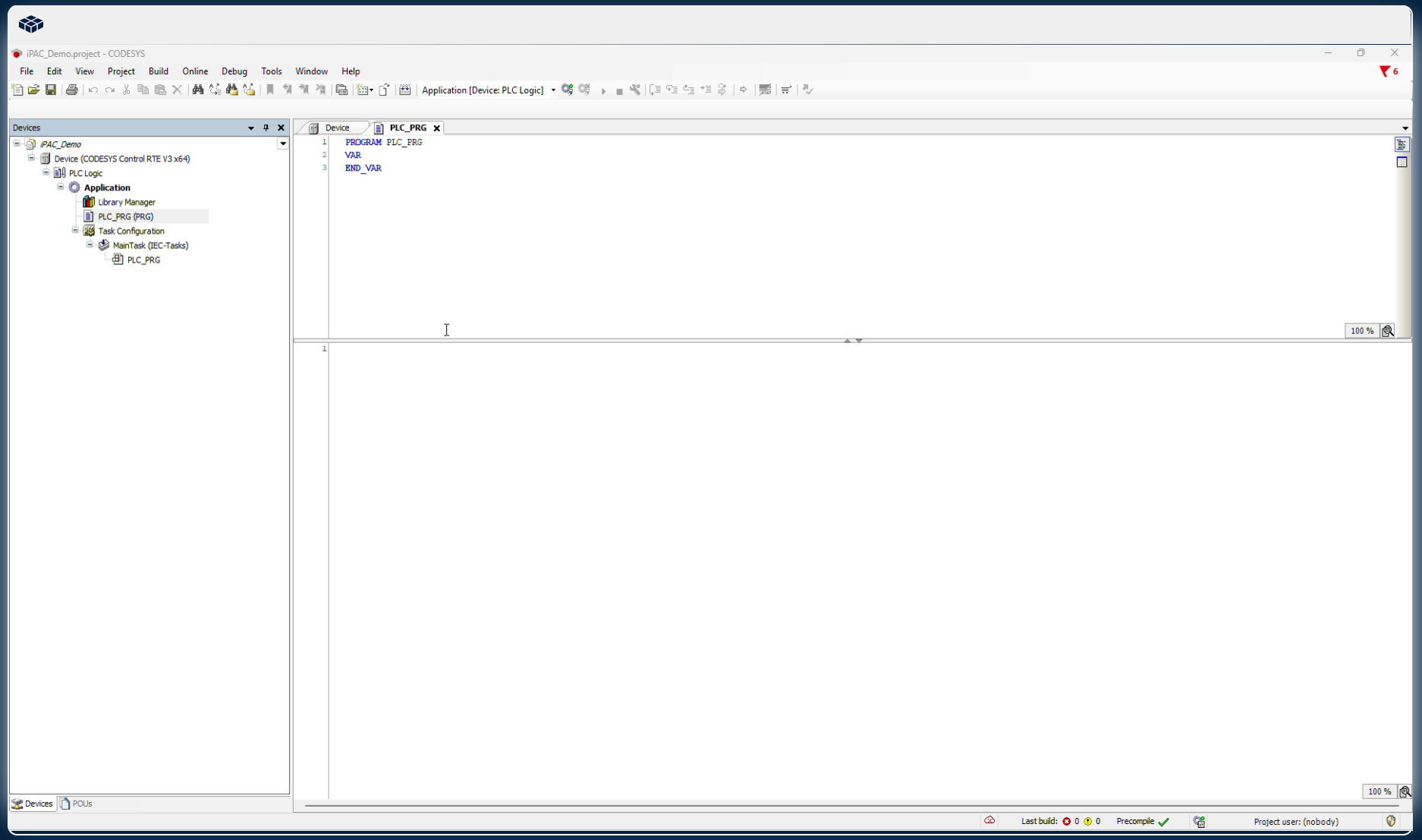
Build/Deploy Application

Build, deploy and run the application the IPC

Creating the Project



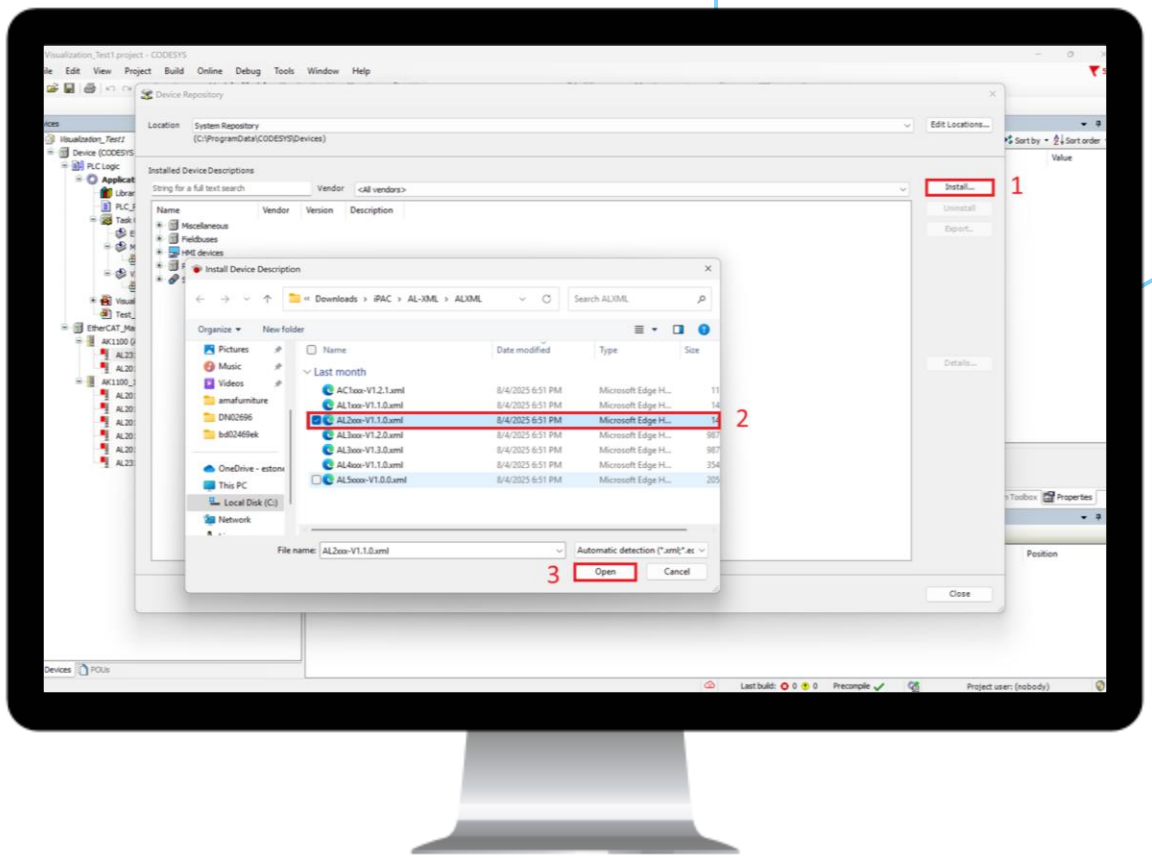
CODESYS → IPC


iPAC_Demo.project - CODESYS
File Edit View Project Build Online Debug Tools Window Help
Application [Device: PLC Logic]
Devices
iPAC_Demo
Device (CODESYS Control RTE V3 x64)
PLC Logic
Application
Library Manager
PLC_PRG (PRG)
Task Configuration
MainTask (IEC-Tasks)
PLC_PRG
Device
PLC_PRG
PROGRAM PLC_PRG
VAR
END_VAR
100 %
100 %
Devices
POUs
Last build: 0 0
Precompile
Project user: (nobody)

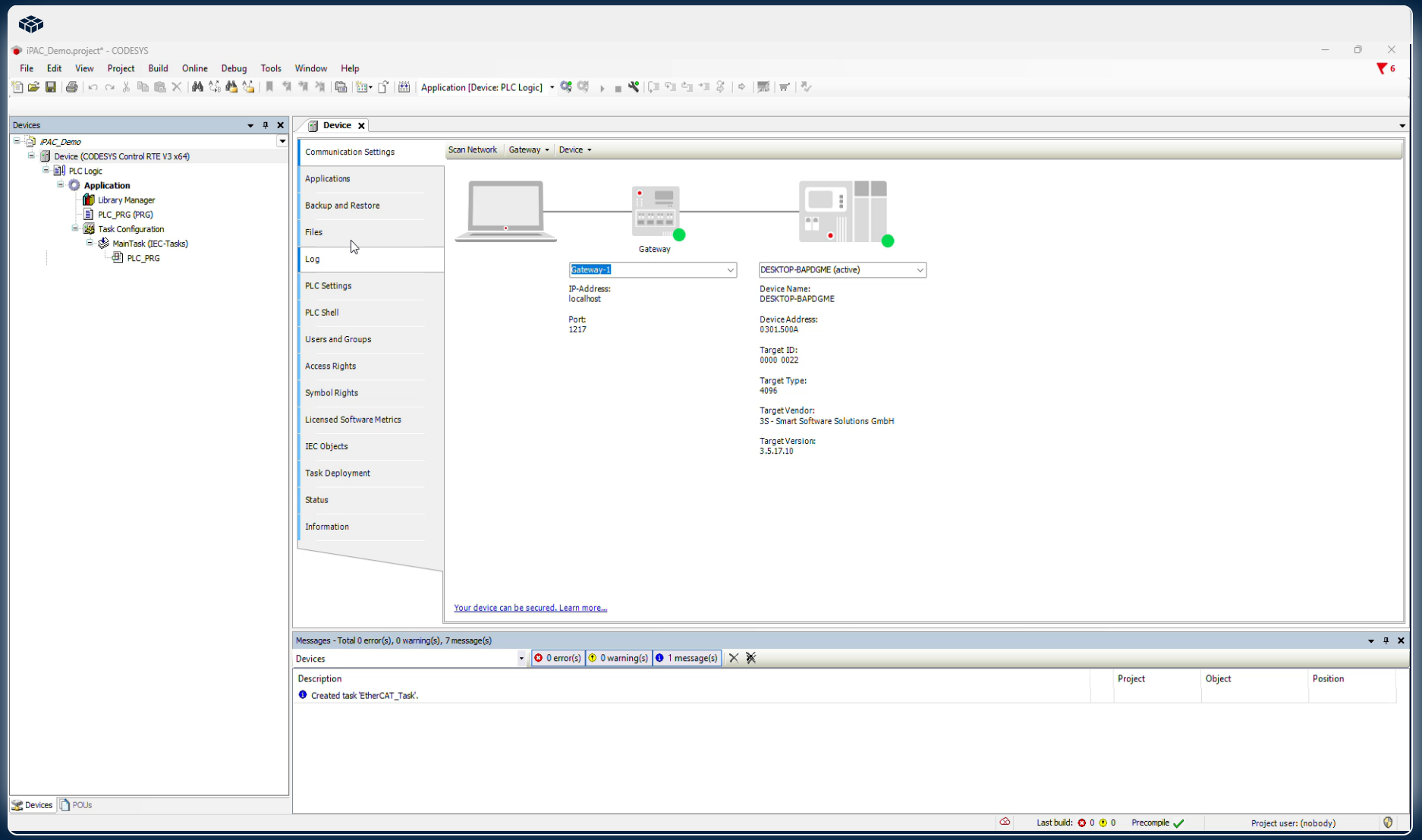
Importing Hardware Descriptions

- Import the device description files (.XML) provided by NODKA
- These files contain the digital blueprint for our modular I/O and EtherCAT modules

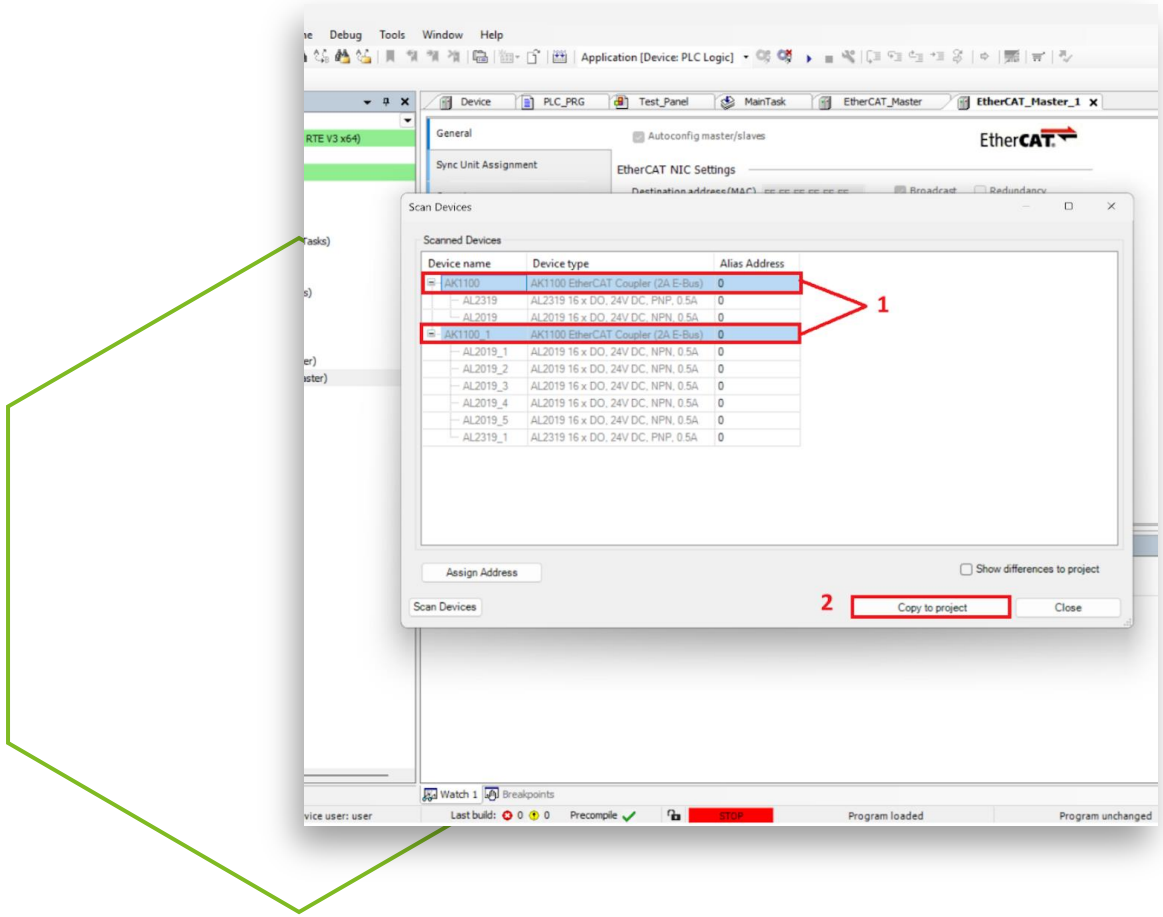
XML Files



Importing Hardware Descriptions

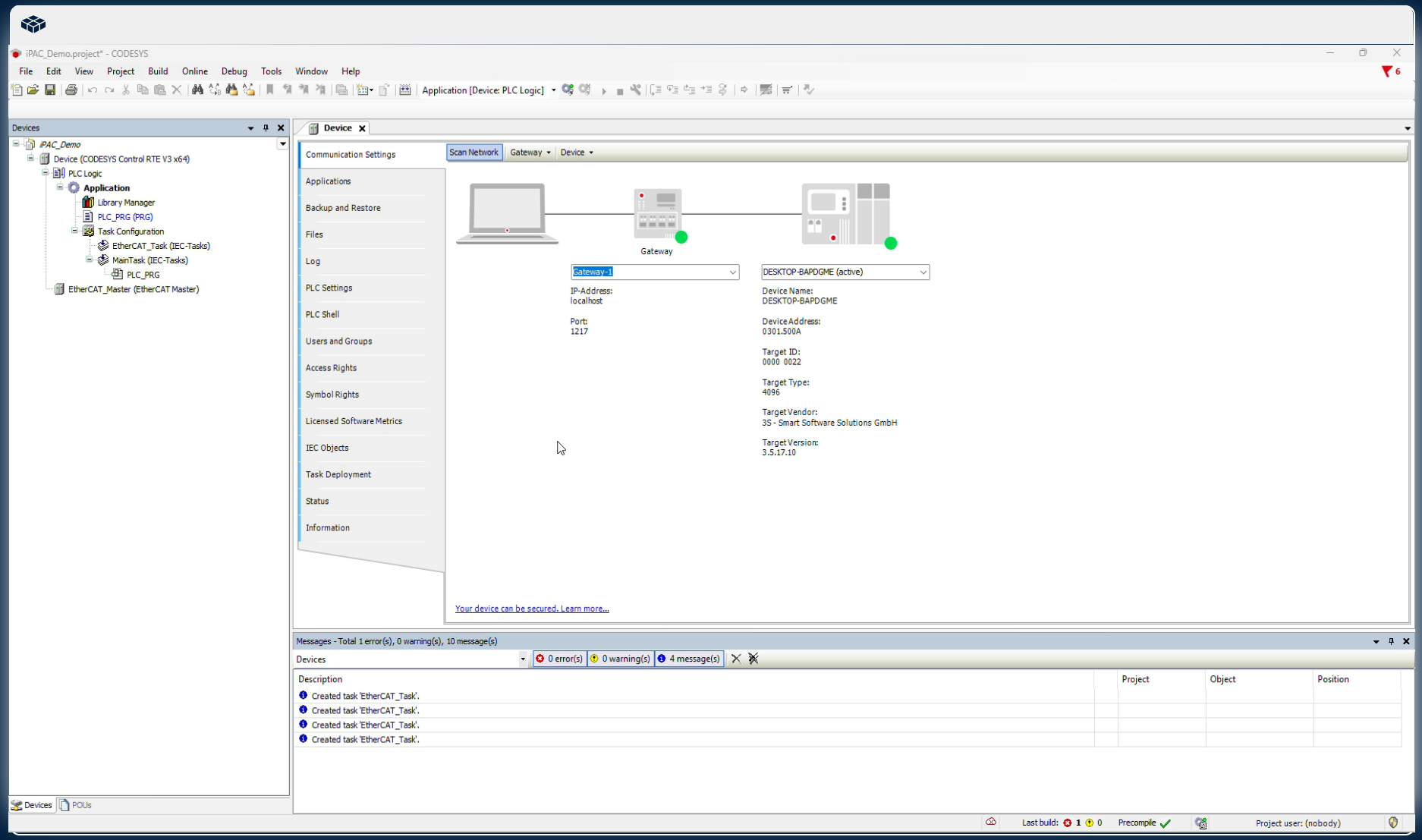


Device Discovery



- Add the EtherCAT Master to the project
- Use the *“Scan for Devices”* button to detect the devices (I/O Modules)
- CODESYS automatically finds and adds all connected drives and I/O modules

Device Discovery



Programming the Sample Application

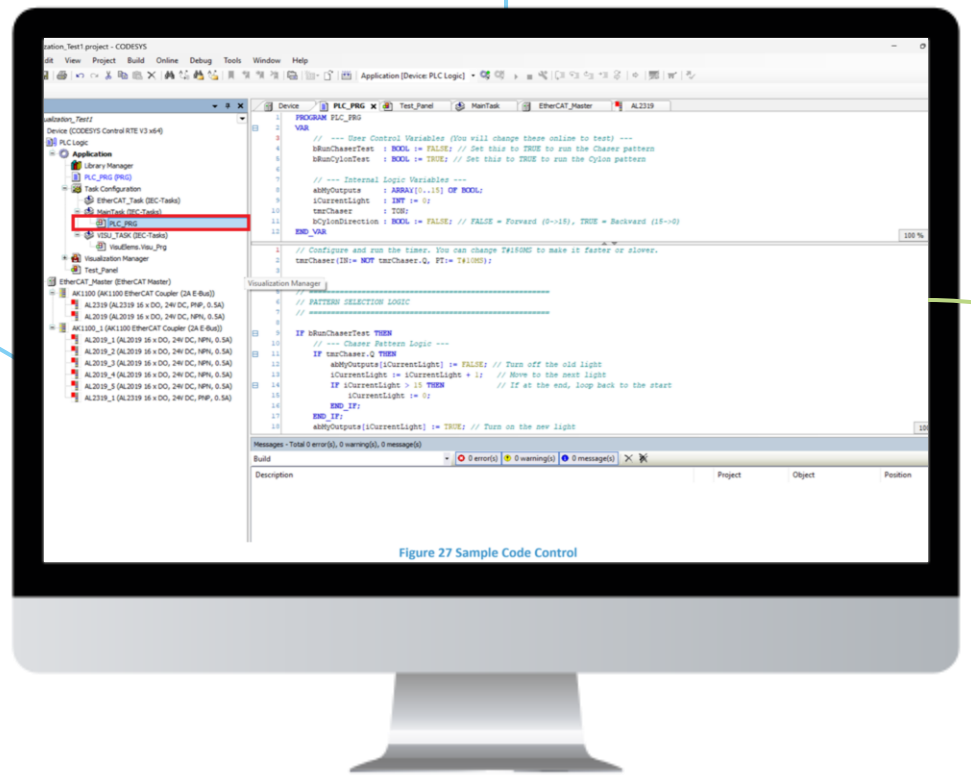
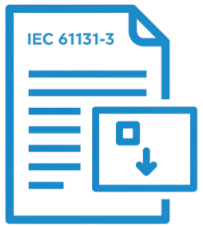
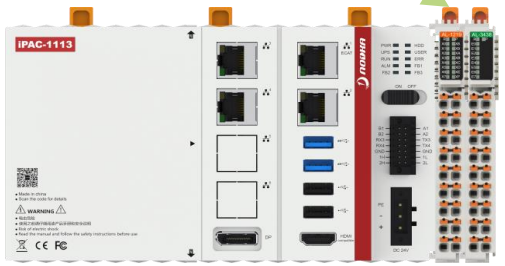


Figure 27 Sample Code Control

- Program the application logic using standard IEC 61131-3 language (Structured Text)
- Implement the logic for I/O Modules control





Connecting Hardware to Software

Mapping Physical I/O to Program Variables

1 Identify Available I/O Channels

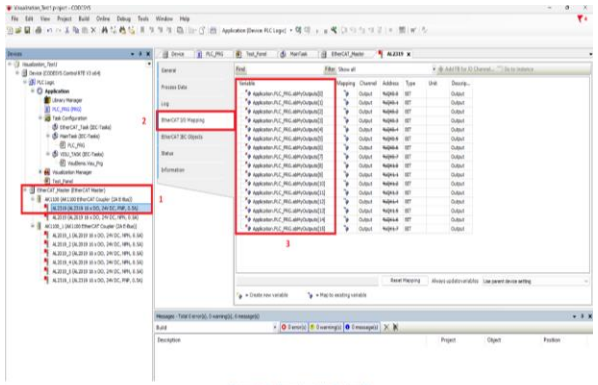
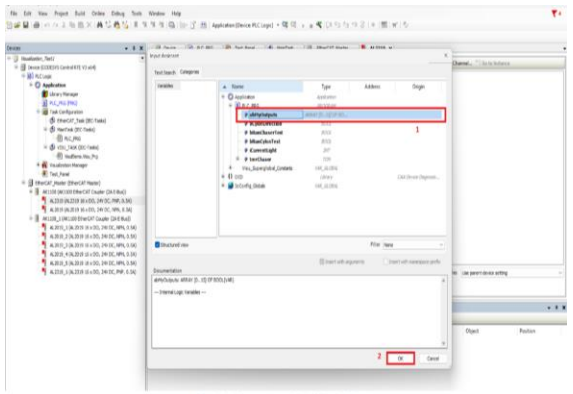


Figure 28 Configuration of the Device I/O

Map physical hardware to your program variables

2 Assign Program Variables



Creates the link between the IPC and the program logic

3 Completed Mapping Verification

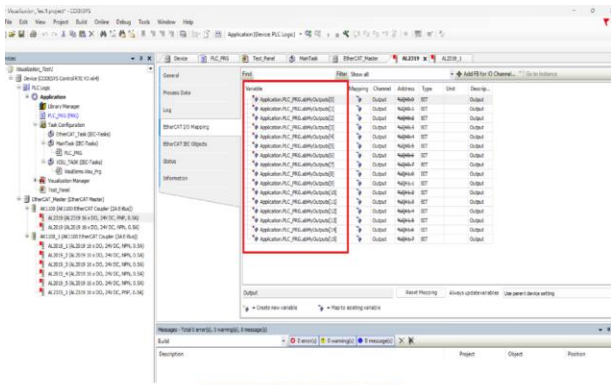
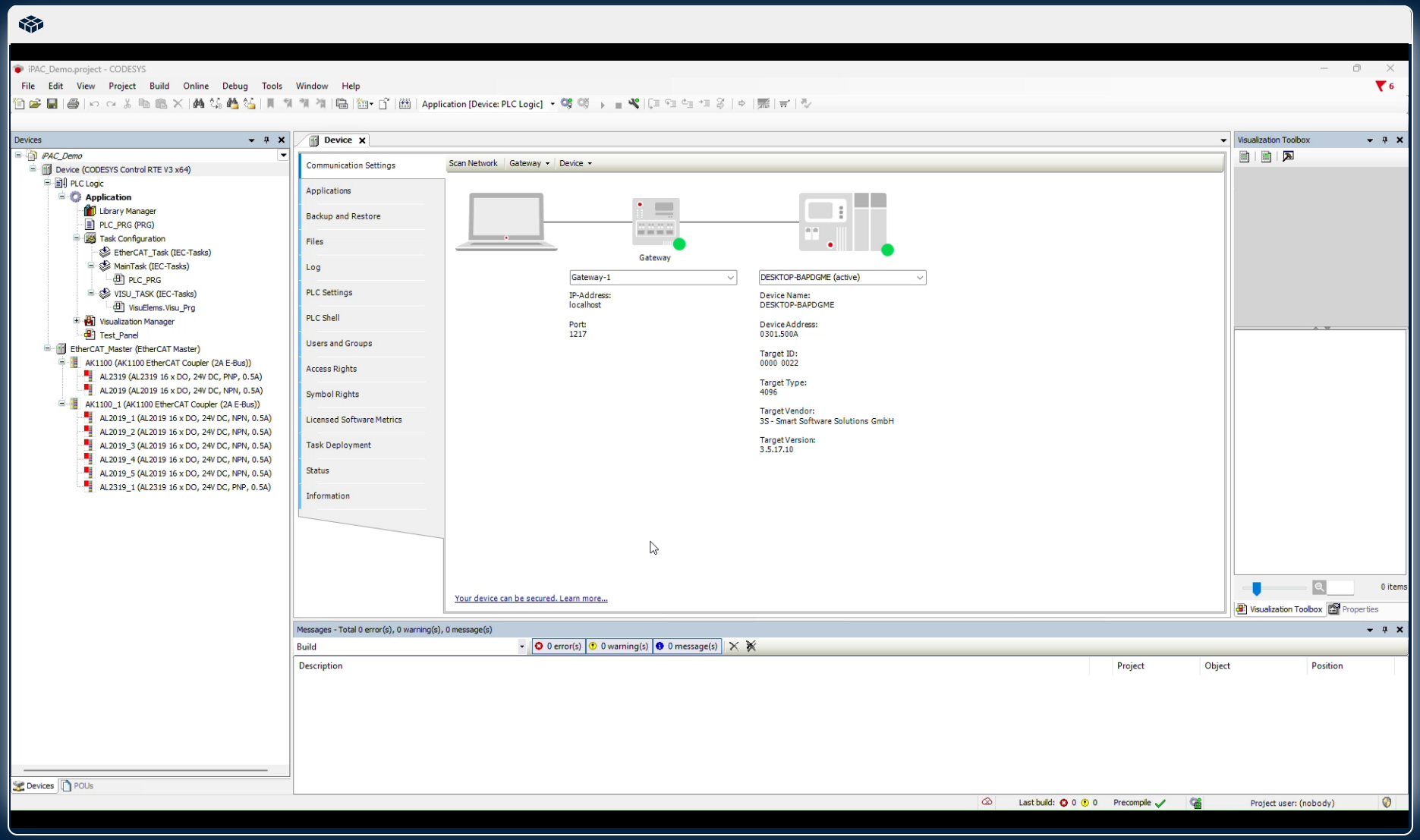


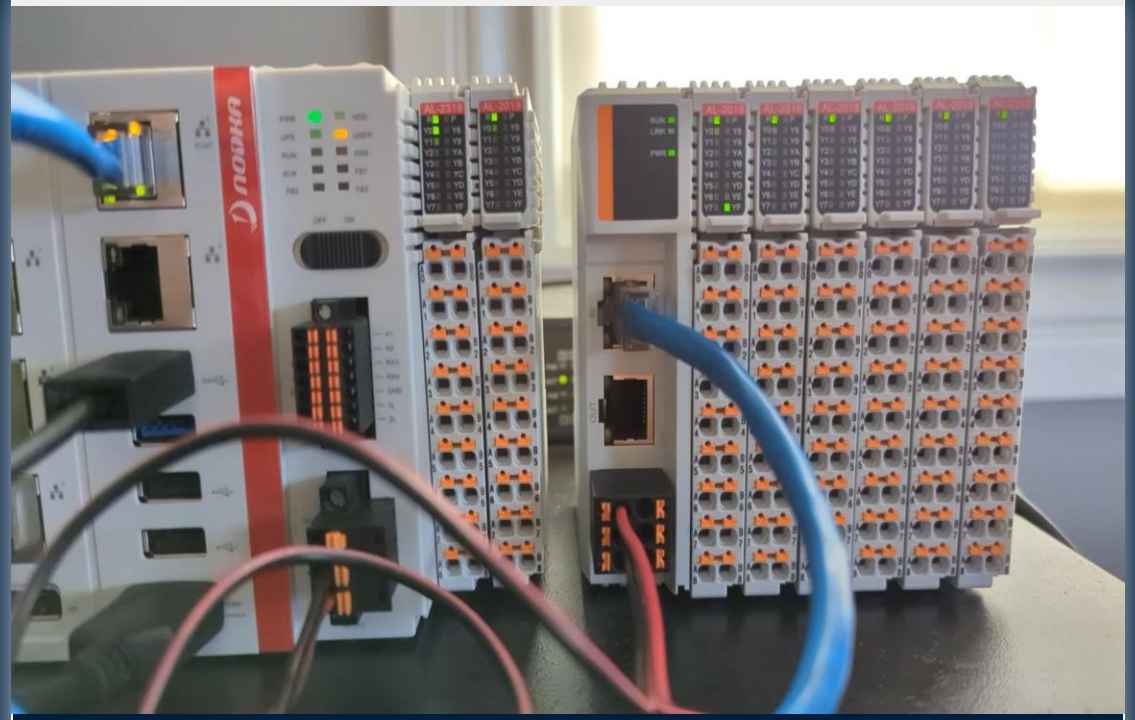
Figure 31 Apply Array [0-15] to all Variables 16 Output Channels

This ensures your software is correctly linked to the IPC

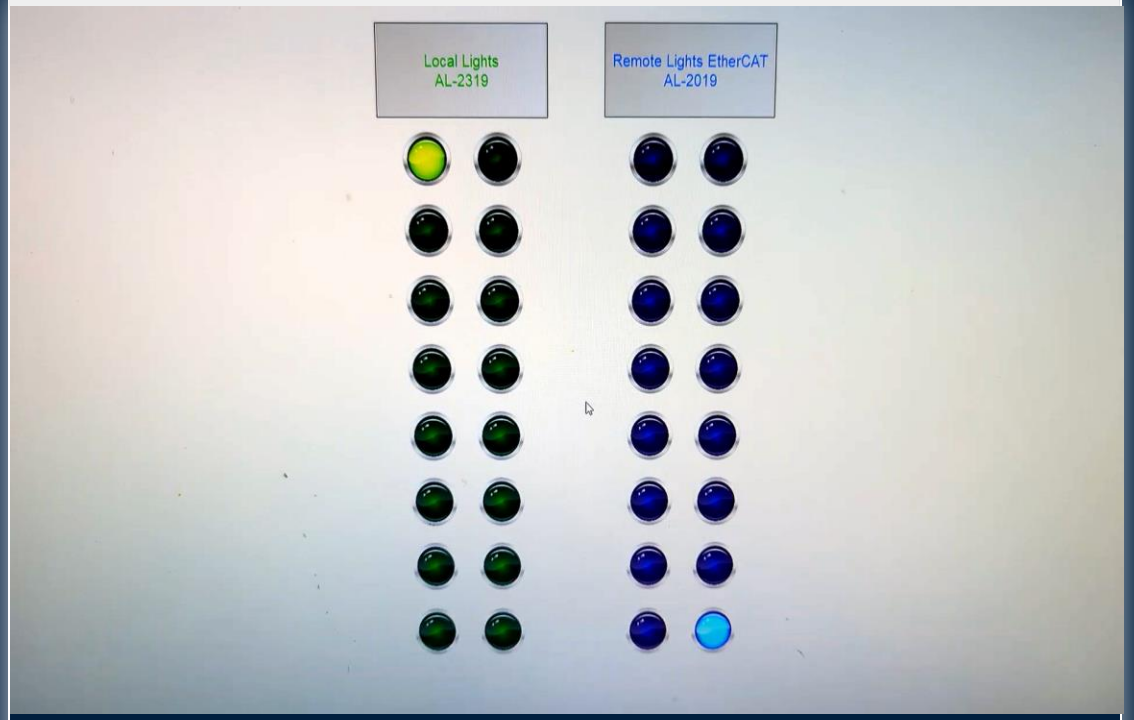
Programming the Logic



Application Output



✓ Device Application Output



✓ CODESYS Application Output

Key Takeaways

✓ Establish Foundation

- Installing the **Real-Time Ethernet Driver** to gain direct, deterministic control over the hardware, bypassing the standard Windows network stack.

✓ Define System

- Create the project in the CODESYS IDE, import the **XML hardware descriptions** (the digital blueprint), and use **Device Discovery** to automatically scan and add all connected I/O modules.

✓ Bridge Software and Hardware

- Programming the logic using standard IEC 61131-3 (Structured Text) and, most importantly, **mapping the physical I/O** to your program variables. Define System

✓ Final Application Output

- The result is a fully deployed application where the logic written directly controls the physical device, as seen in the final **Application Output**.

CODESYS®

Tech Talk Fall 2025

October 13 - 17

CODESYS® is a registered trademark. Subject to misprints, errors, and technical changes. Reproduction (even in parts) and distribution only with prior permission. Please note: Not all CODESYS functions are available in all countries. For further information on these country-specific restrictions, please contact sales@codesys.com.

Data Sheets

https://nodka.com/automation-pcs/

For More Information

https://nodka.com/solutions/automation-pc-for-motion-control/

North American Sales & Service

+1 909-594-7630 | sales@nodka.us

User Guide

https://nodka.com/technology/step-by-step-guide-configuring-an-ethercat-controller-with-codesys-on-an-ipc/

Your contact

CODESYS North America

NorthAmerica@codesys.com