



Corso di Automazione industriale

Lezione 2

PLC - Introduzione

Cos'è un PLC

PLC: Programmable Logic Controller

Unità di elaborazione in grado di pilotare degli attuatori a fronte di variazioni acquisite attraverso sensori ad esso connessi



Cos'è un PLC

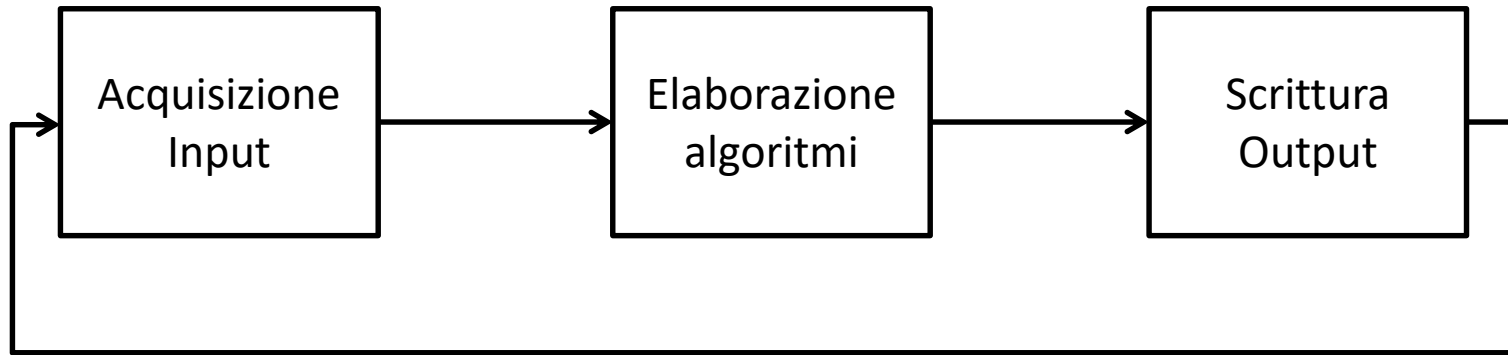
PLC: Programmable Logic Controller

Per funzionare questo dispositivo ha bisogno di funzionare con una frequenza fissata, per questo motivo deve essere real-time

Cosa significa real-time?

Significa avere noto il tempo massimo di esecuzione di una porzione di codice

Ciclo PLC



N.B.: Solitamente il tempo di rilancio del ciclo è definito; tuttavia, con alcuni PLC (ad esempio Siemens), se non viene definito un blocco funzionale con un tempo di campionamento predefinito, l'esecuzione avviene in un «ciclo infinito» (in un main con `while true`), quindi senza un tempo di rilancio noto

Real-time

Ogni produttore di PLC personalizza uno o più sistemi operativi real-time per consentirne l'integrazione con i propri dispositivi

I sistemi operativi più utilizzati nell'ambito dell'automazione industriale sono:

- VXWorks
- QNX
- Windows embedded (o CE)

Real-time

Come funziona un S.O. real-time?

Un sistema operativo real-time è un sistema deterministico, cioè in grado di garantire, a priori, il massimo tempo di esecuzione di un programma.

Definizioni:

- Release time: istante in cui il programma è disponibile per l'esecuzione
- Deadline: istante entro cui l'esecuzione deve essere terminata
- Completion time: istante di terminazione dell'esecuzione

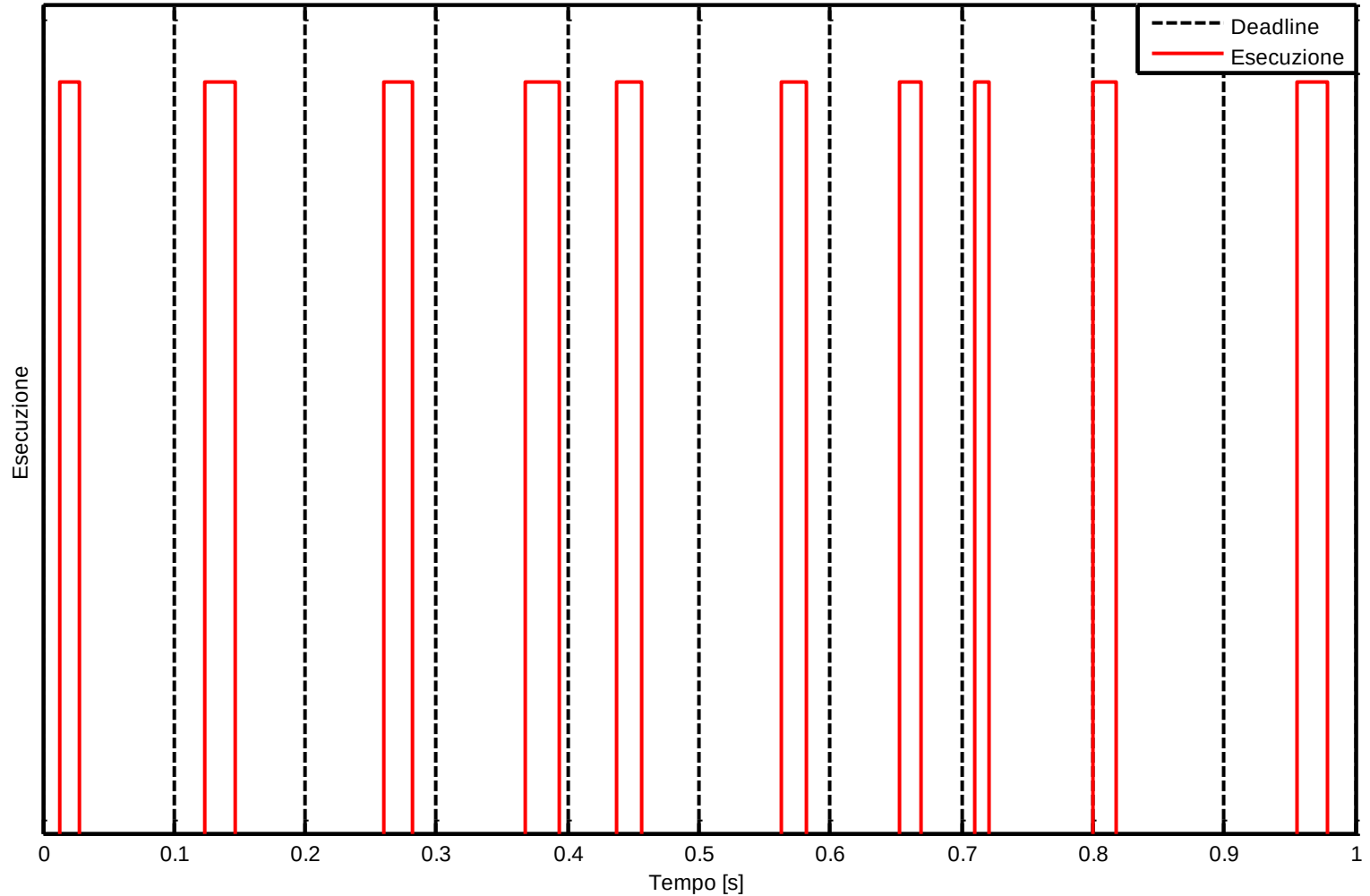
Real-time

Nel caso dei PLC solitamente:

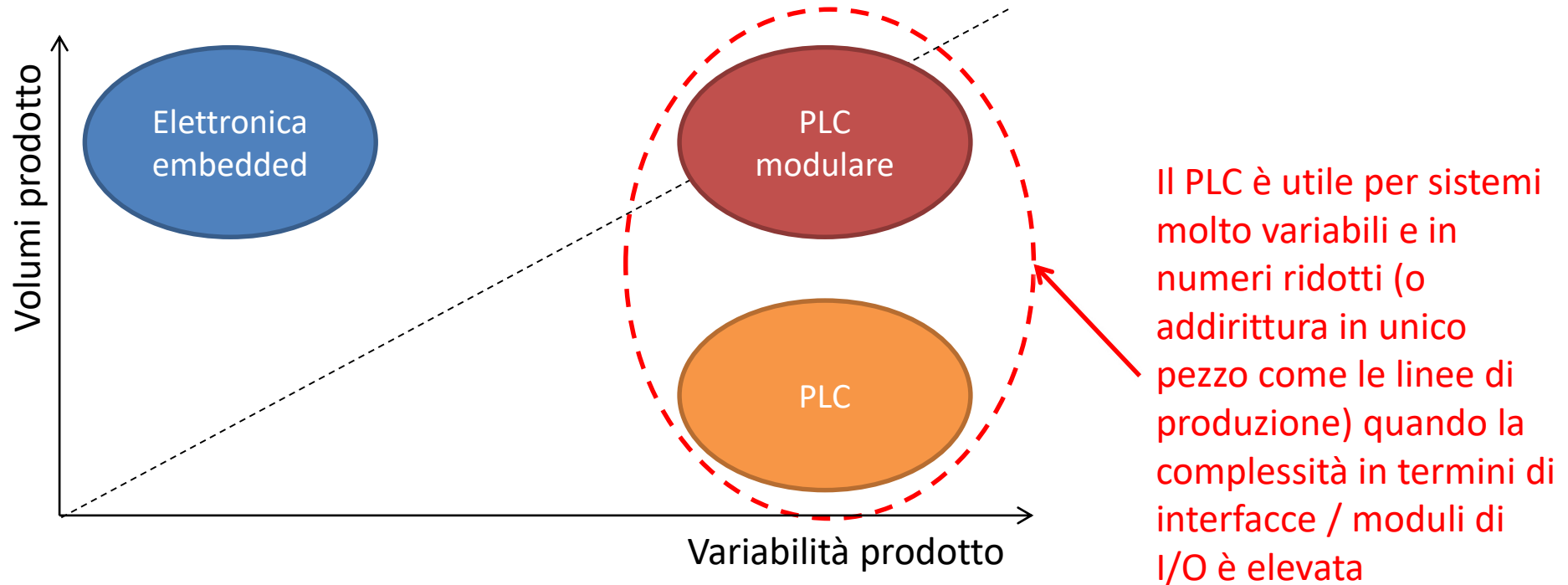
- Release time: il programma è disponibile per l'esecuzione al termine della deadline successiva
- Deadline: equivale al tempo di rilancio del programma
- Completion time: dipende dal carico della CPU su cui viene eseguito il programma

Real-time

Esempio di esecuzione



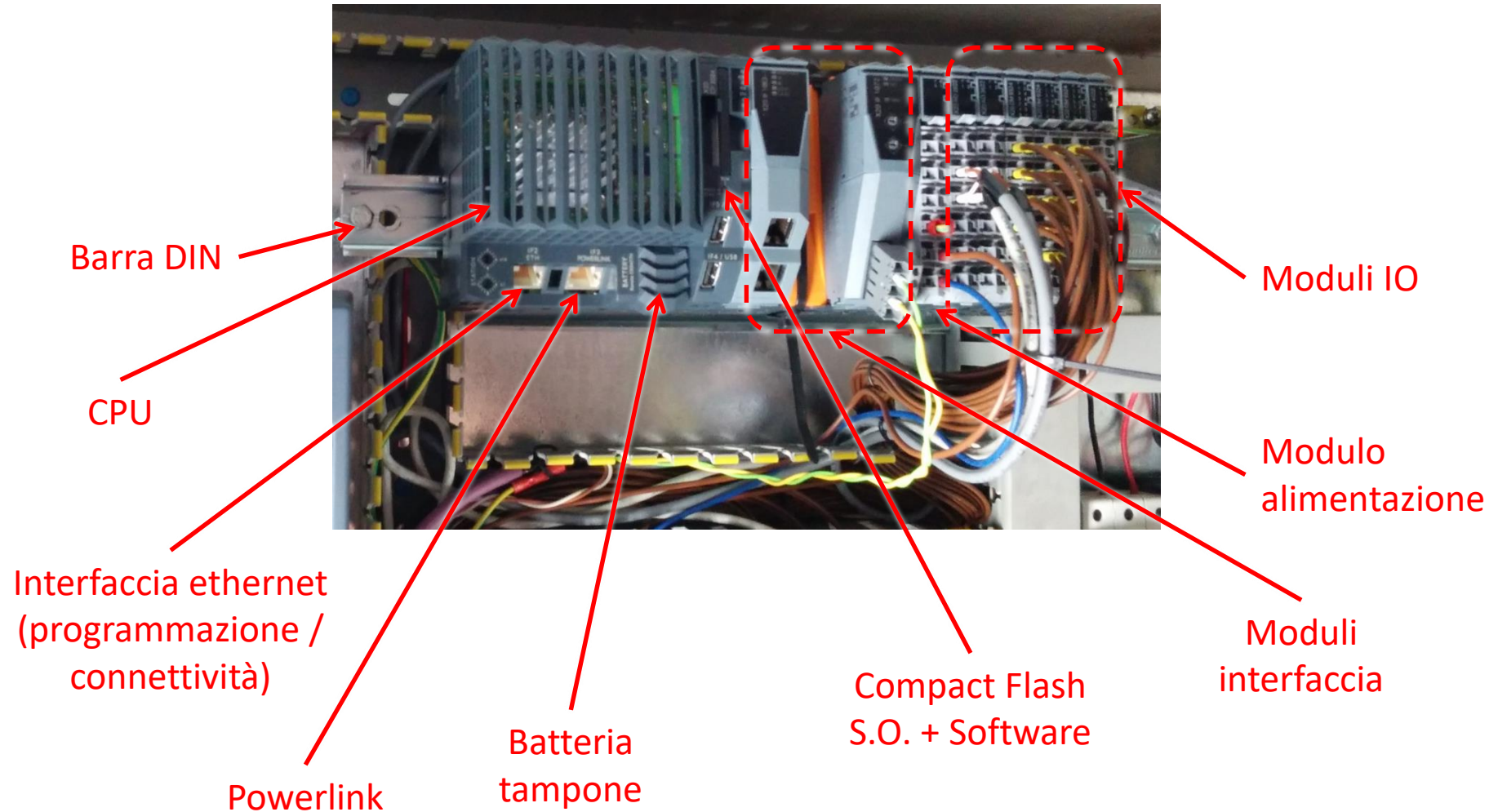
Quando serve un PLC



Tipologie

μPLC	PLC media dimensione	PLC grande dimensione
 In – Out ≤ 64 Da alcuni anni moduli di espansione Digitali / Analogici	 $64 < \text{In} - \text{Out} < 512$ La differenza tra queste due tipologie di PLC sta perdendo di significato visto l'utilizzo massivo di PLC modulari	 In – Out ≥ 512

Struttura



Architettura hardware

La tendenza degli ultimi anni è avvicinare l'architettura del PLC a quella di un normale x86.

In generale le memorie utilizzate sono:

- EEPROM su cui è caricato il S.O. e il programma utente (attualmente su dispositivi removibili come CF, SD, ecc...)
- RAM su cui viene allocato il software in esecuzione e vengono mantenute le variabili (in alcuni casi la RAM viene mantenuta attiva con la batteria tampone o supercap in caso di mancanza di alimentazione)

Architettura hardware

Esistono molte schede I/O, le più importanti (che la maggior parte dei produttori ha) sono:

- In / Out analogici (± 10 V, $-20 \div 20$ mA, $4 \div 20$ mA)
- In / Out digitali (5 V-DC, 24 V-DC, 240 V-AC)
- In / Out per controllo motore (encoder, PWM)
- In vari (termocoppie, celle di carico, ecc...)
- Interfacce per i bus di campo

Programmazione

Tutti i PLC sono programmabili attraverso un PC (solitamente con S.O. Windows)

Ormai tutte le marche utilizzano l'interfaccia Ethernet presente sul PLC

Quasi tutti utilizzano software di programmazione proprietario anche se alcuni sono compatibili con software di terze parti (ad esempio Codesys)

N.B.: In laboratorio useremo B&R Automation Studio 4.0

IEC 61131

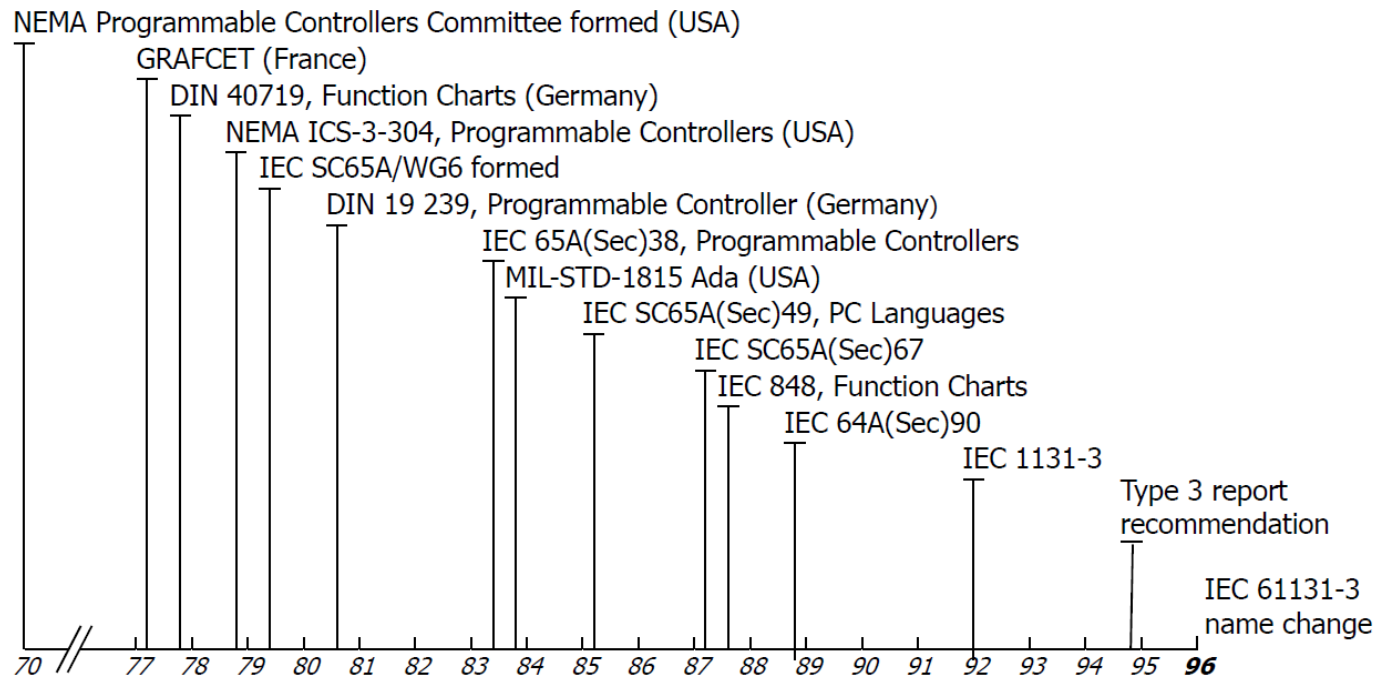
La IEC è una organizzazione no-profit e non-governativa nata allo scopo di definire e pubblicare standard relativi alle tecnologie elettriche ed elettroniche (e quelle ad esse correlate)

E' nata il 26 giugno 1906 dalla IEE inglese e AIEE americana

Era inizialmente a Londra ma nel 1948 è stata spostata a Ginevra

IEC 61131

Per quanto riguarda gli standard relativi ai PLC si fa riferimento alla norma IEC 61131 (che in passato era nota come IEC 1131, prima della variazione della numerazione IEC avvenuta nel 1996)



IEC 61131

Perché è così importante questa norma?

Perché è anche uno standard di fatto nell'industria

Perché l'80% dei PLC sul mercato si basano su di essa

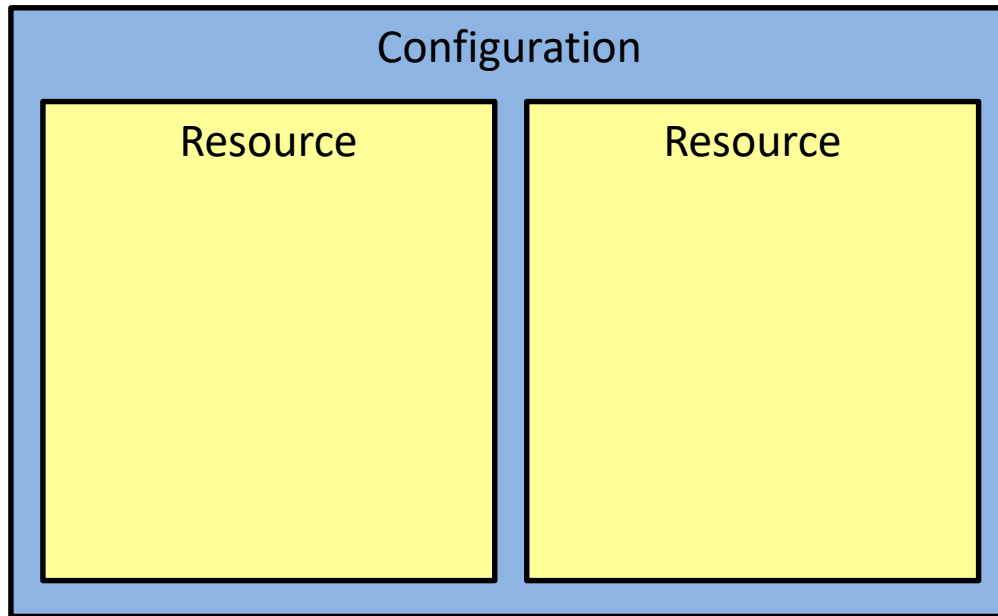
Perché aiuta a mantenere una struttura coerente con «il resto del mondo»

IEC 61131

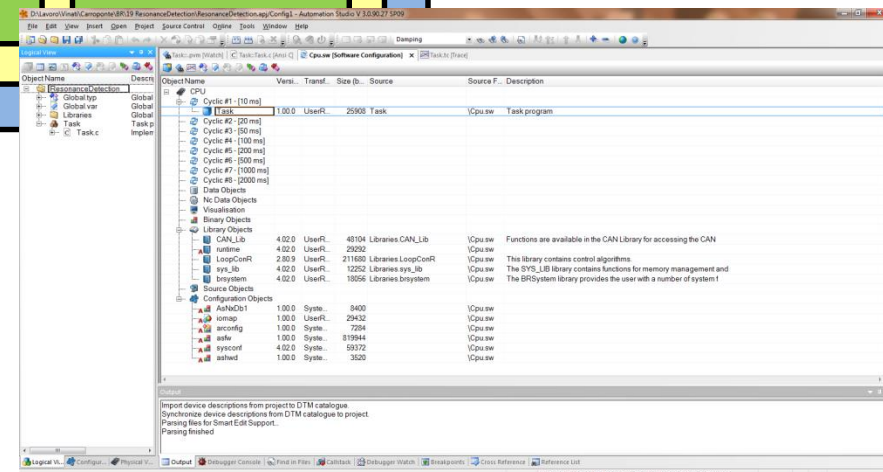
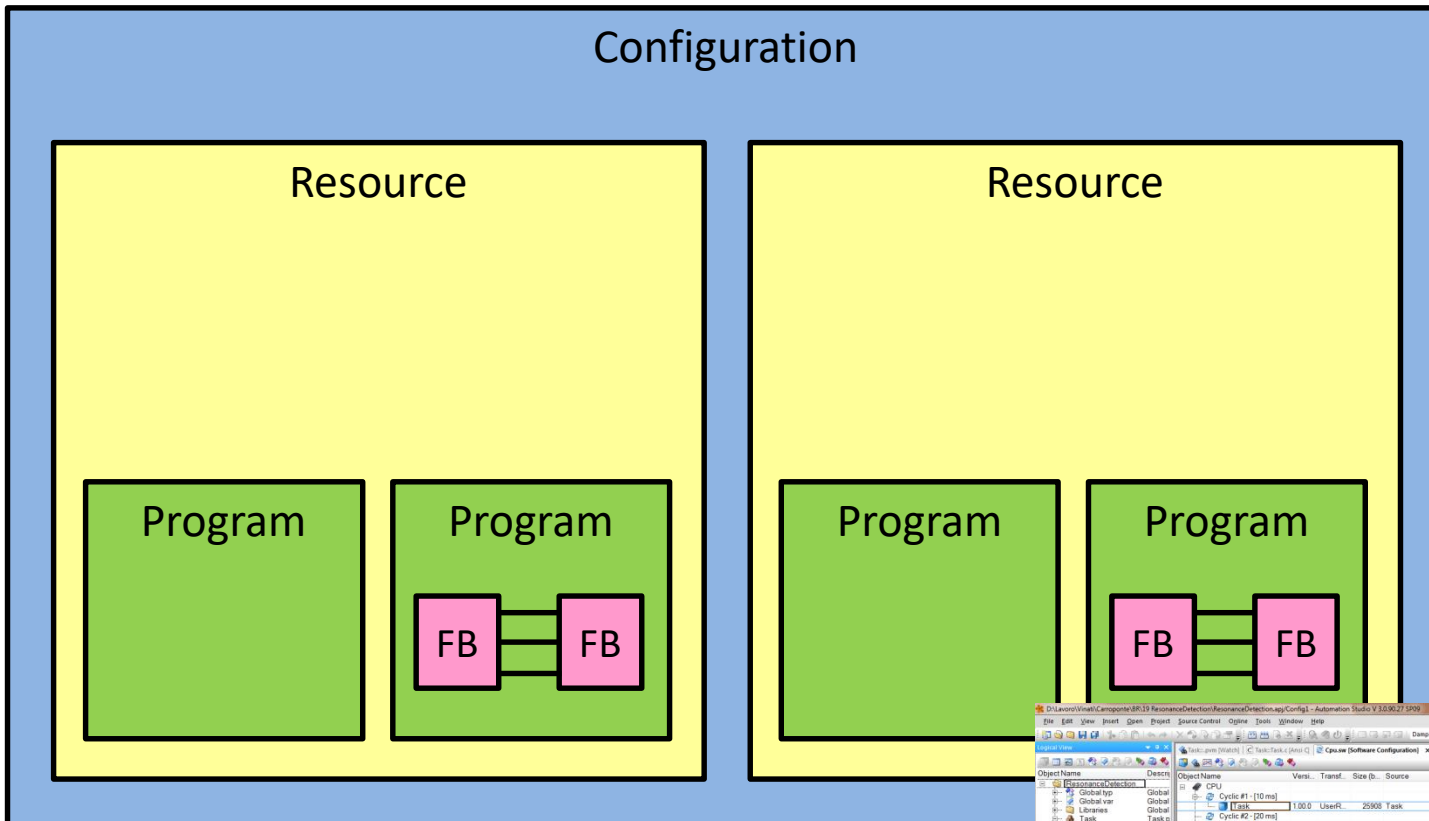
La norma è costituita dalle seguenti parti:

1. Overview generale, definizioni
2. Hardware
3. Linguaggi di programmazione
4. Linee guida per gli utenti
5. Comunicazione
6. Programmazione in logica booleana
7. Programmazione in logica fuzzy
8. Linee guida per l'implementazione

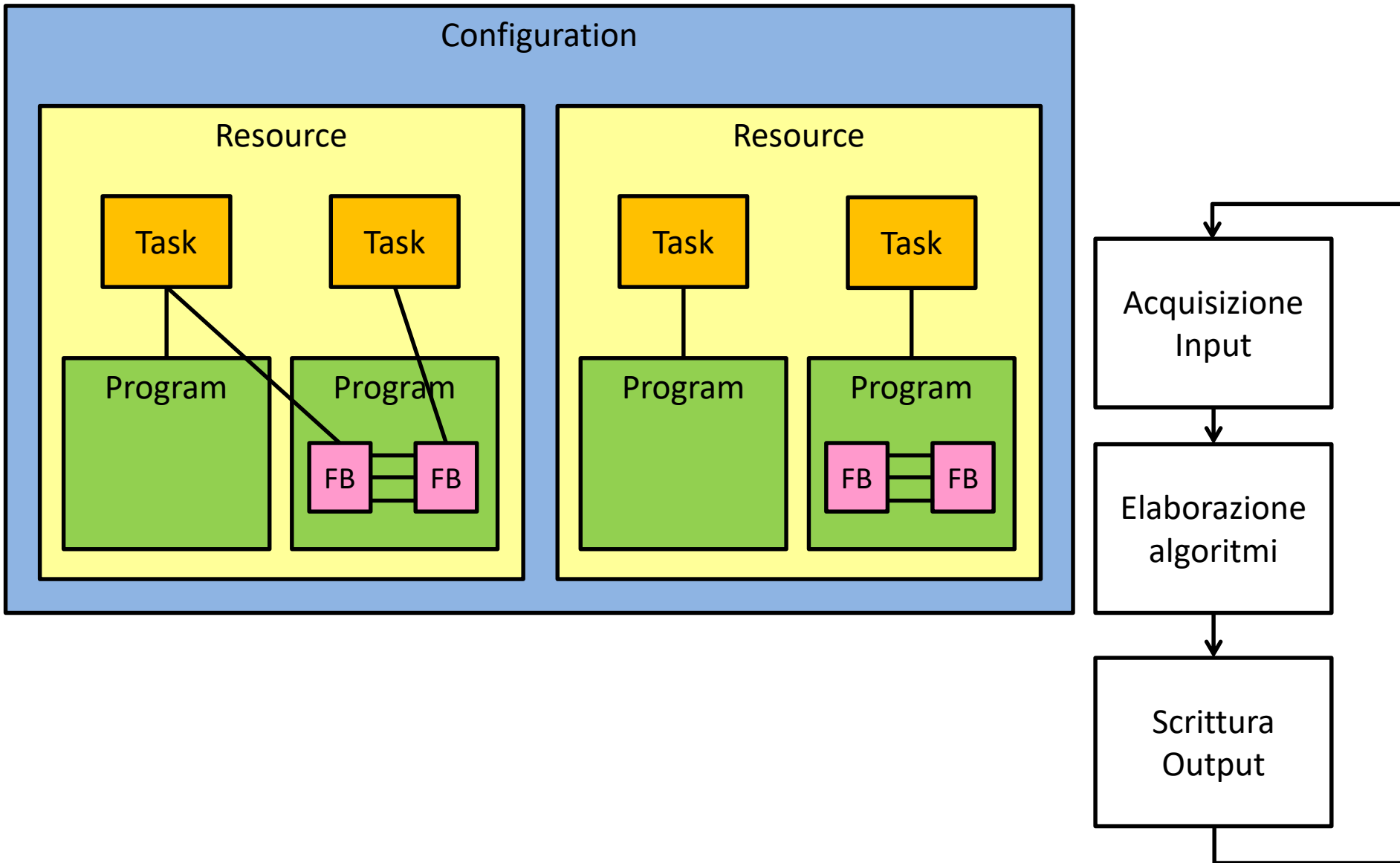
Struttura dell'applicazione



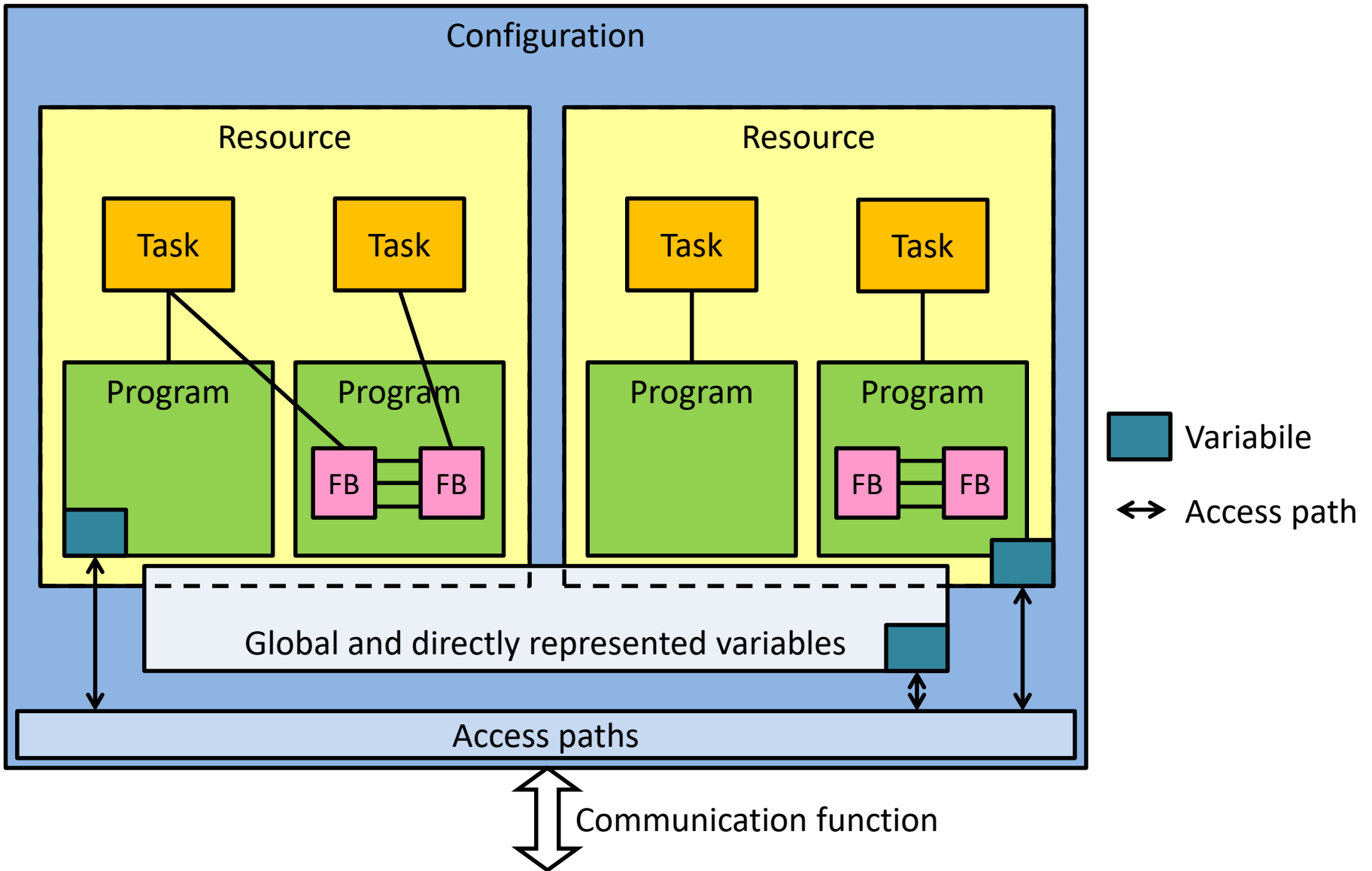
Struttura dell'applicazione



Struttura dell'applicazione



Struttura dell'applicazione



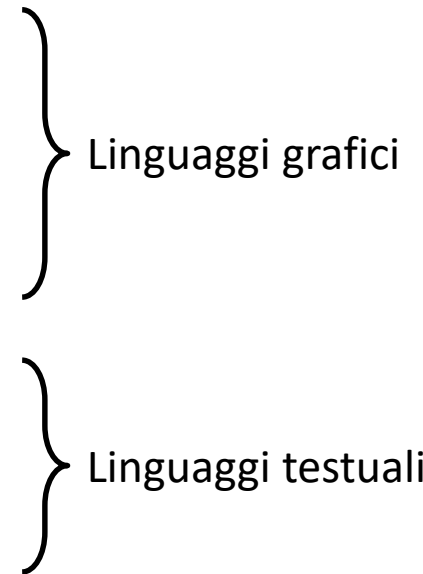
Struttura dell'applicazione

La comunicazione tra programmi può essere

- All'interno della stessa configurazione
 - Diretta se tra due function block dello stesso program
 - Attraverso variabili globali se tra due programmi
- Tra due configurazioni
 - Diretta tra function block di comunicazione
 - Accesso ad un access path condiviso

Linguaggi di programmazione

Nella parte 3 della norma (come mostrato in precedenza) vengono definiti 5 possibili linguaggi di programmazione:

- Ladder Diagram
 - SFC – Sequential Function Chart
 - FBD – Function Block Diagram
 - Instruction List
 - ST – Structured Text
- 
- The diagram uses two large curly braces on the right side to group the languages. The first brace groups 'Ladder Diagram', 'SFC – Sequential Function Chart', and 'FBD – Function Block Diagram' under the label 'Linguaggi grafici'. The second brace groups 'Instruction List' and 'ST – Structured Text' under the label 'Linguaggi testuali'.
- Linguaggi grafici
- Linguaggi testuali

N.B.: Per garantire l'esecuzione real-time questi linguaggi vengono tradotti in istruzioni di basso livello

Linguaggi di programmazione

Elementi comuni dei 5 linguaggi della norma IEC 61131:

- Identificatori
 - Primo carattere non può essere un numero
 - Non possono essere presenti due _ consecutivi
 - Non ci siano spazi
- Keywords
 - PROGRAM, FUNCTION, VAR, END_, ecc...
 - BOOL, BYTE, WORD, INT, REAL, TIME, STRING, prefissi S, D, L, U, ecc...
 - RETAIN, CONSTANT, ecc...

Linguaggi di programmazione

Elementi comuni dei 5 linguaggi della norma IEC 61131:

- Funzioni
 - ADD, SQRT, SIN, COS, GT, MIN, MAX, AND, OR, ecc...

Linguaggi di programmazione

Ladder Diagram

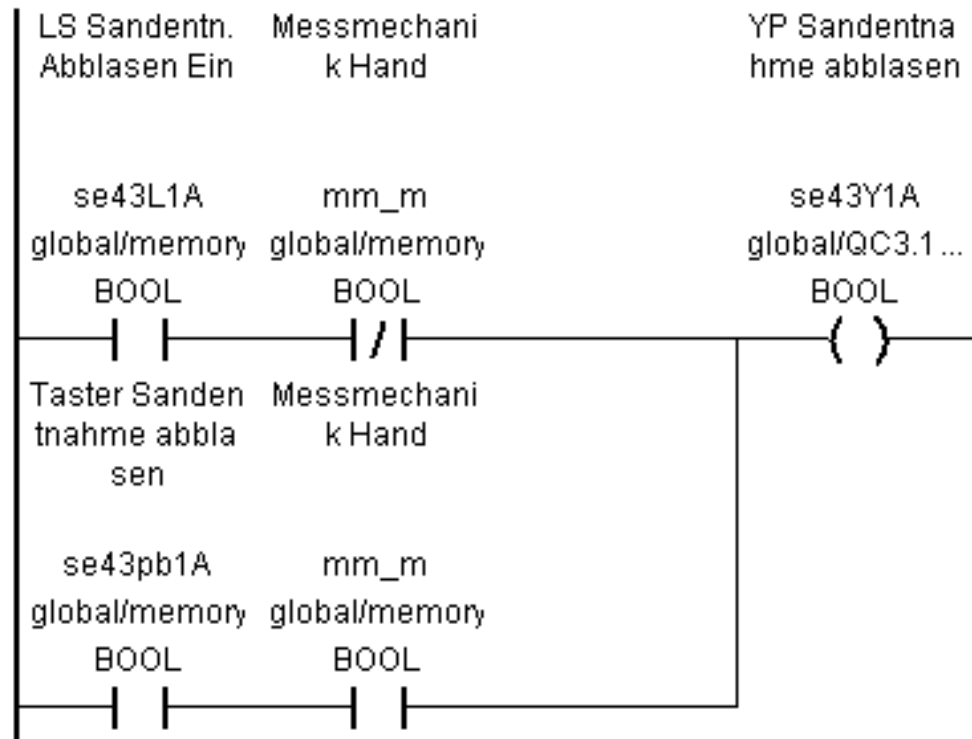
E' il linguaggio di programmazione più vecchio della IEC 61131

Si basa su simboli di provenienza elettrica: Binari di potenza, Contatti, Bobine

Viene chiamato Ladder a causa dell'aspetto grafico che hanno i programmi realizzati in questo linguaggio

Linguaggi di programmazione

Ladder Diagram



Linguaggi di programmazione

SFC – Sequential Function Chart

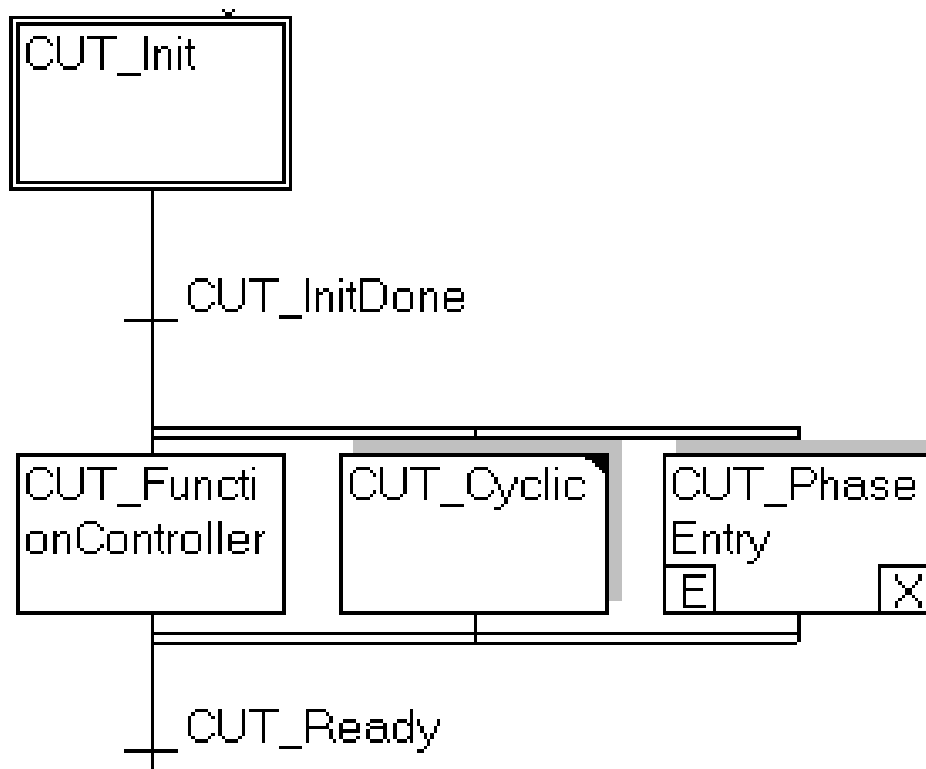
Linguaggio derivante dallo standard IEC 848

E' un linguaggio orientato al sequencing, per questo motivo è adatto ad un approccio di programmazione top-down

Si basa su: Step, Transizioni, Azioni, Archi orientati

Linguaggi di programmazione

SFC – Sequential Function Chart



Linguaggi di programmazione

FBD – Function Block Diagram

Linguaggio derivante dallo standard IEC 167

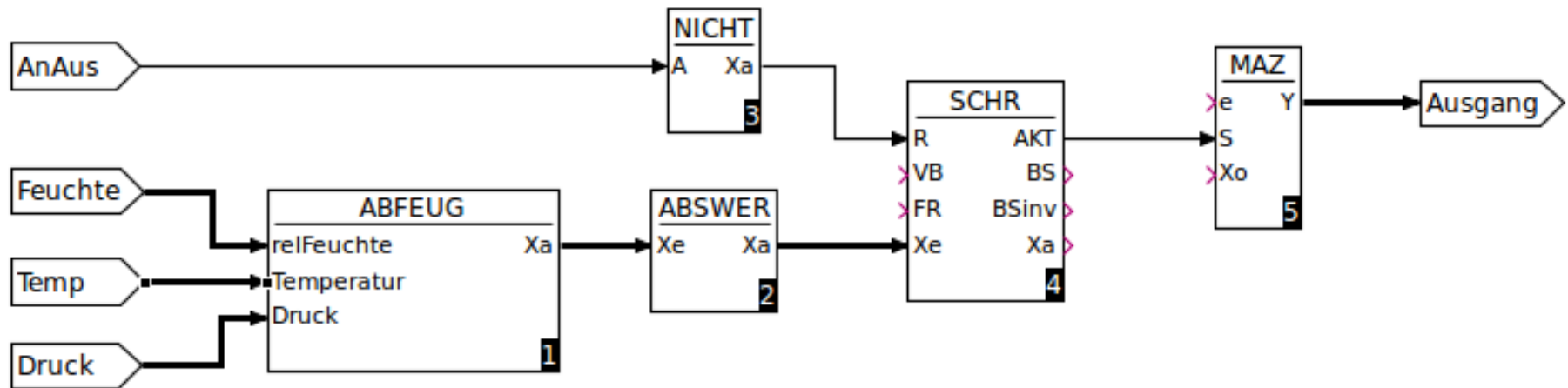
La rappresentazione grafica del FBD è simile a quella degli schemi a blocchi (visti in Fondamenti di Automatica)

Solitamente non viene utilizzato molto per il controllo logico

L'esecuzione dipende dalla posizione dei blocchi funzione

Linguaggi di programmazione

FBD – Function Block Diagram



Linguaggi di programmazione

Instruction List

Linguaggio più di basso livello presente nella norma IEC 61131

E' molto simile al codice assembly che sicuramente avete visto in passato

E' troppo di basso livello per essere utilizzato in ambito reale, anche se il «dialetto» Siemens (detto AWL) viene utilizzato

Linguaggi di programmazione

Instruction List

1.1	LD	I 0.0	Input Contact
1.2	MOVW	# 980 , VW200	Put 980 in VW200
1.3	LD	SM 0.5	Pulse generator
1.4	EU		Raising edge
1.5	SLW	VW200 # 1	Shift Left Word (1) bit
1.6	LD	SM1.1	Overflow =1 if last bit shifted output =1
1.7	=	Q0.0	Output Coil
1.8	MEND		End programming

Linguaggi di programmazione

ST – Structured Text

E' un linguaggio nato dai vecchi Pascal e Visual Basic

E' da considerarsi un linguaggio di alto livello rispetto all'Instruction List

Esistono dei tool per la generazione di testo strutturato a partire da implementazioni di alto livello (ad esempio PLC Coder di Mathworks)

Ogni produttore di PLC ha il suo «dialetto» ST

Linguaggi di programmazione

ST – Structured Text

```
FUNCTION_BLOCK FBControl
CASE ssMethodType OF
  0:
    (* Start for Enabled SubSystem: '<S1>/EnabledCase' *)
    (* Start for Enabled SubSystem: '<S3>/Dos' *)
    (* InitializeConditions for Atomic SubSystem: '<S6>/PID - antiwindup' *)

    (* InitializeConditions for UnitDelay: '<S12>/Unit Delay2' *)
    UnitDelay2_DSTATE := 0.0;

    (* InitializeConditions for UnitDelay: '<S10>/Unit Delay' *)
    UnitDelay_DSTATE_a := 0.0;

    (* InitializeConditions for UnitDelay: '<S12>/Unit Delay1' *)
    UnitDelay1_DSTATE := 0.0;

    (* InitializeConditions for UnitDelay: '<S11>/Unit Delay2' *)
    UnitDelay2_DSTATE_f := 0.0;

    (* InitializeConditions for UnitDelay: '<S11>/Unit Delay1' *)
    UnitDelay1_DSTATE_k := 0.0;
    (* End of InitializeConditions for SubSystem: '<S6>/PID - antiwindup' *)
    (* End of Start for SubSystem: '<S3>/Dos' *)
```

Linguaggi di programmazione

Nell'ambito del corso ci concentreremo su:

- Linguaggi grafici
 - Ladder Diagram
 - Sequential Function Chart
- Linguaggi testuali
 - Testo Strutturato

Tool-chain

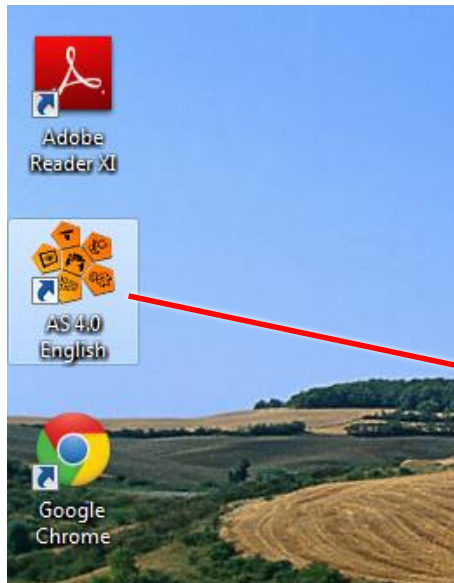
Come già anticipato nell'ambito del corso utilizzeremo l'ambiente di sviluppo per PLC B&R Automation Studio 4.0



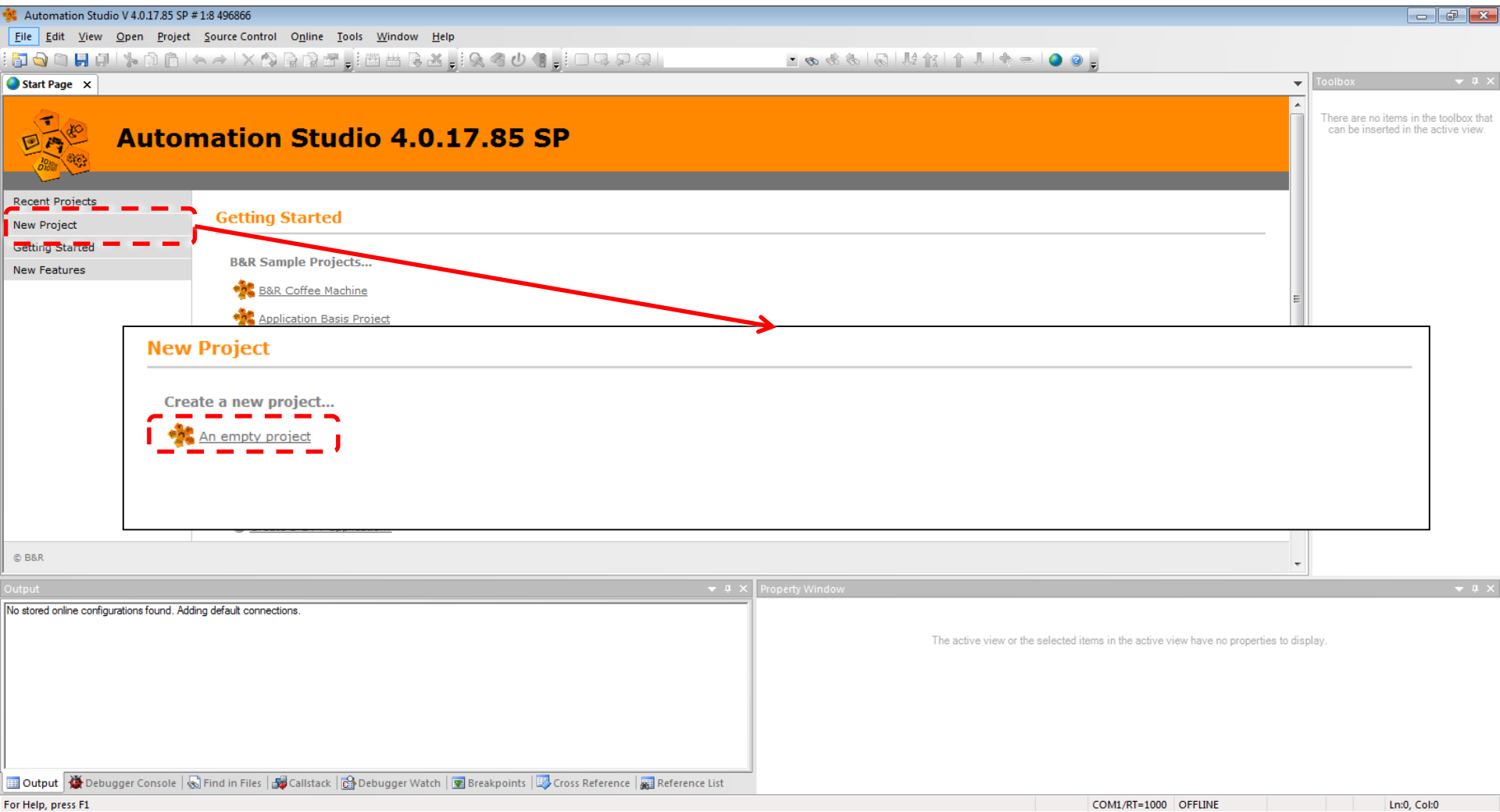
Prima di entrare nei dettagli dei linguaggi di programmazione della IEC 61131 diamo un'occhiata a come è strutturato il software

Automation Studio

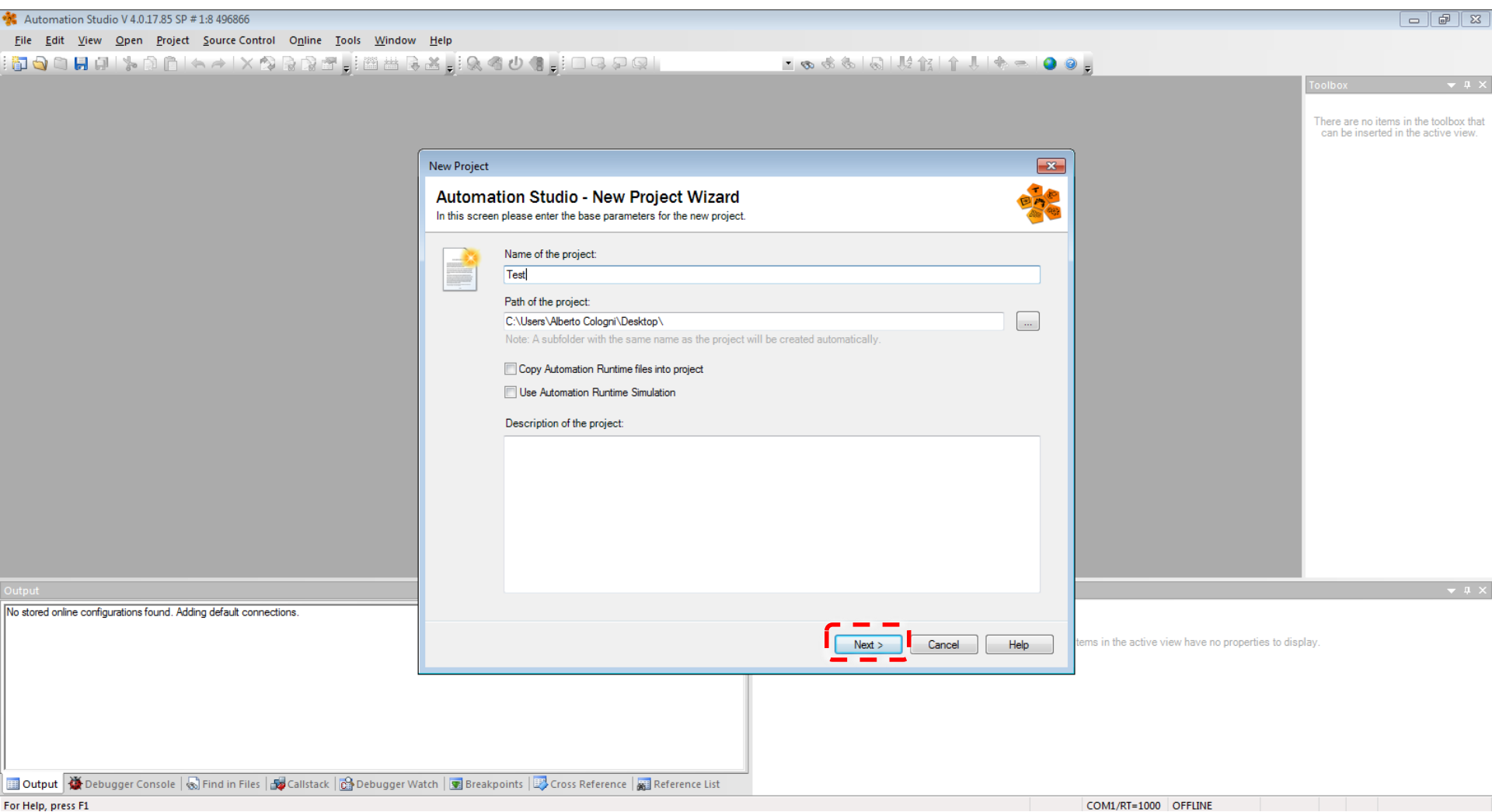
Trovate sulla pagina del corso il link per il download dell'ambiente di sviluppo (sono alcuni GB, se volete datemi la chiavetta e ve lo passo), abbiamo 30 giorni reali di utilizzo.



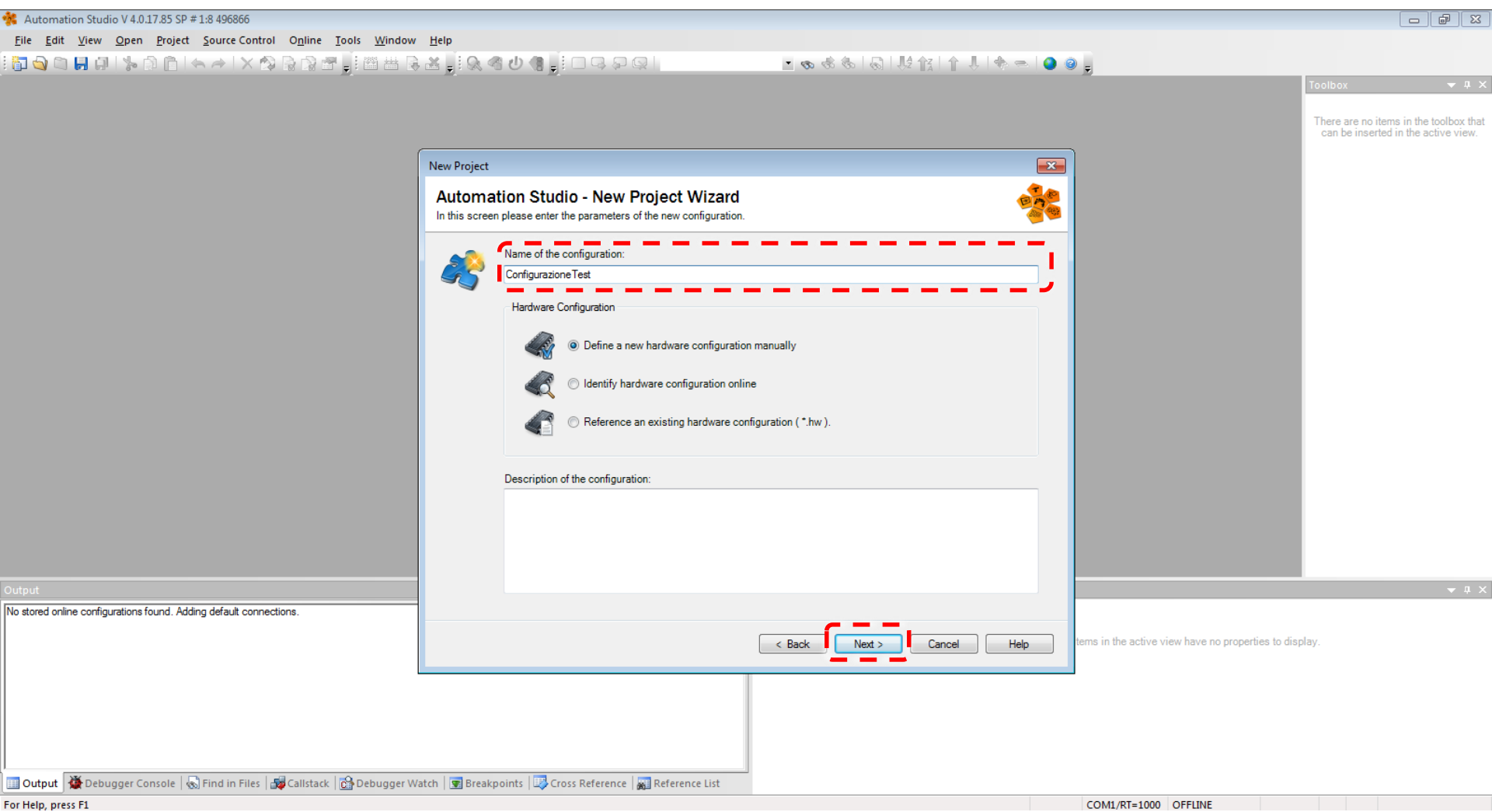
Automation Studio



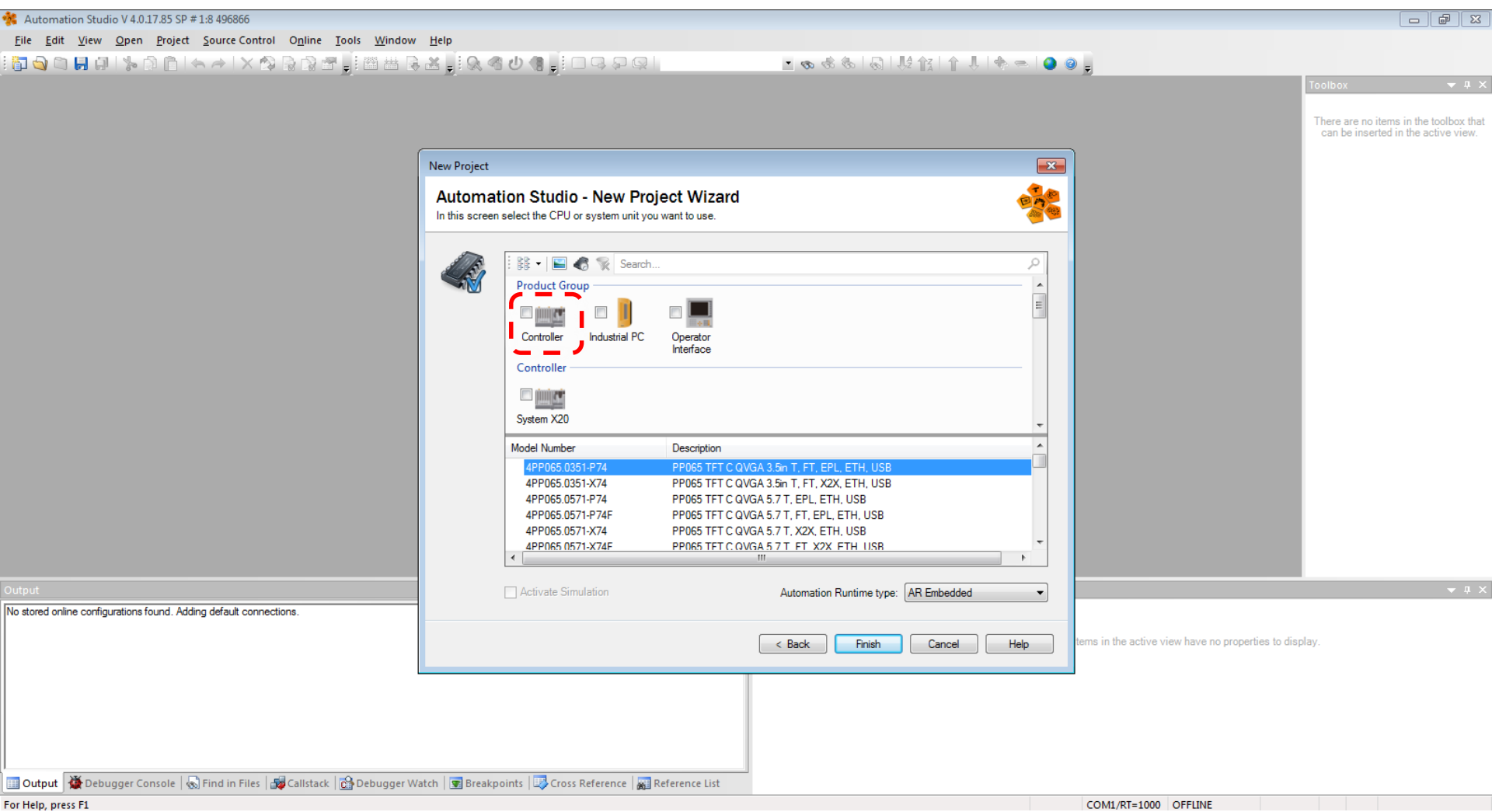
Automation Studio



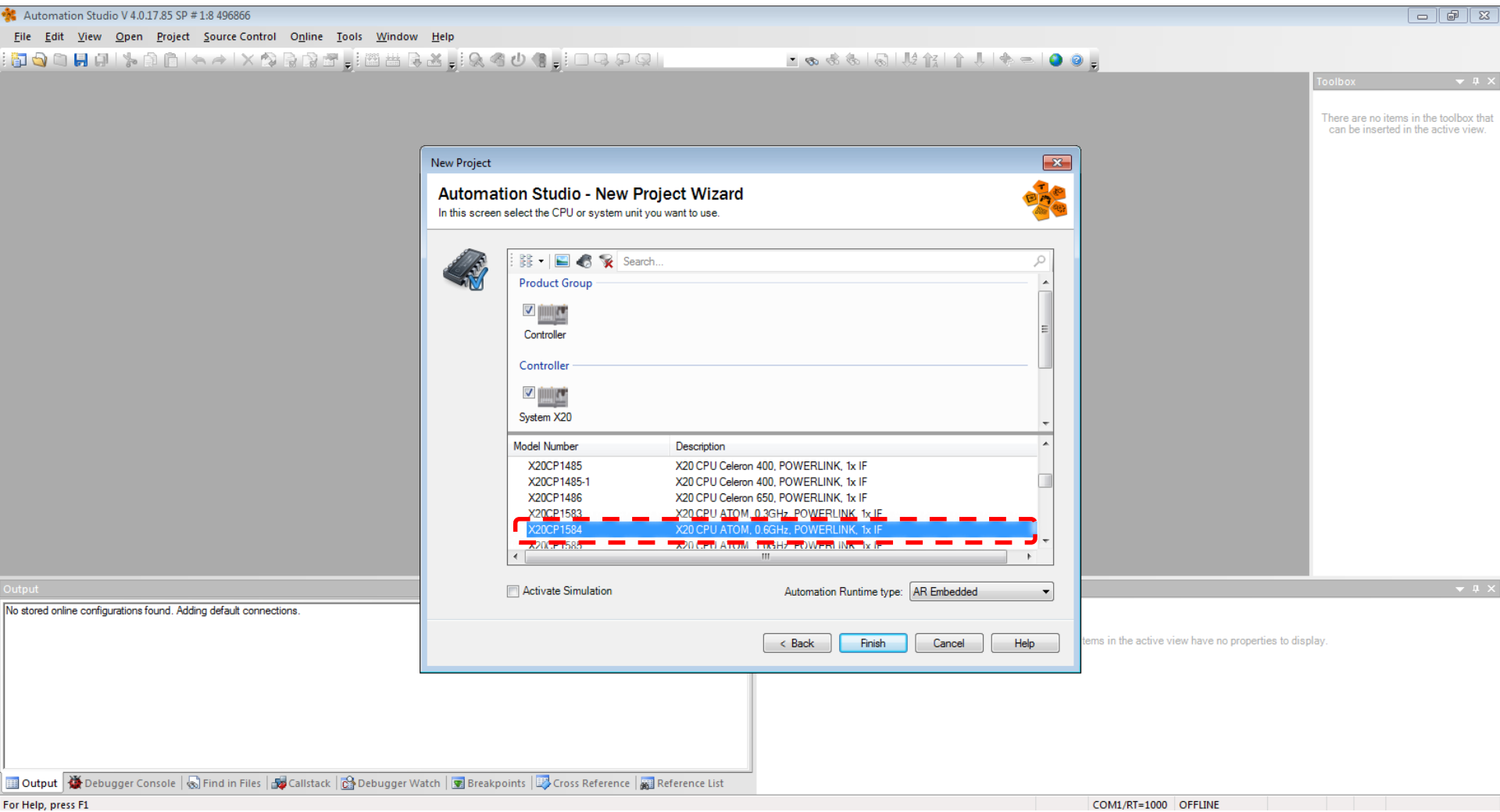
Automation Studio



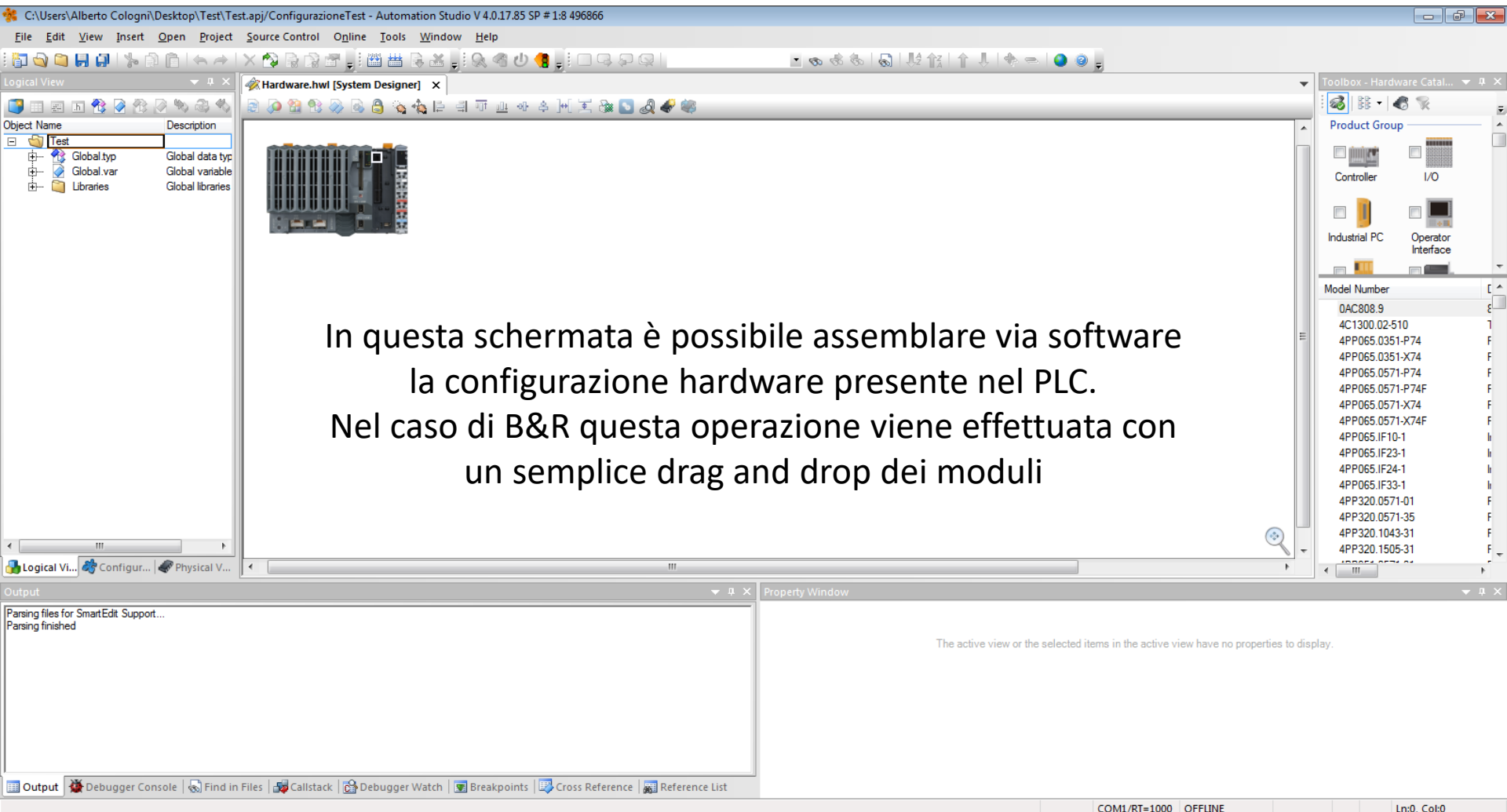
Automation Studio



Automation Studio



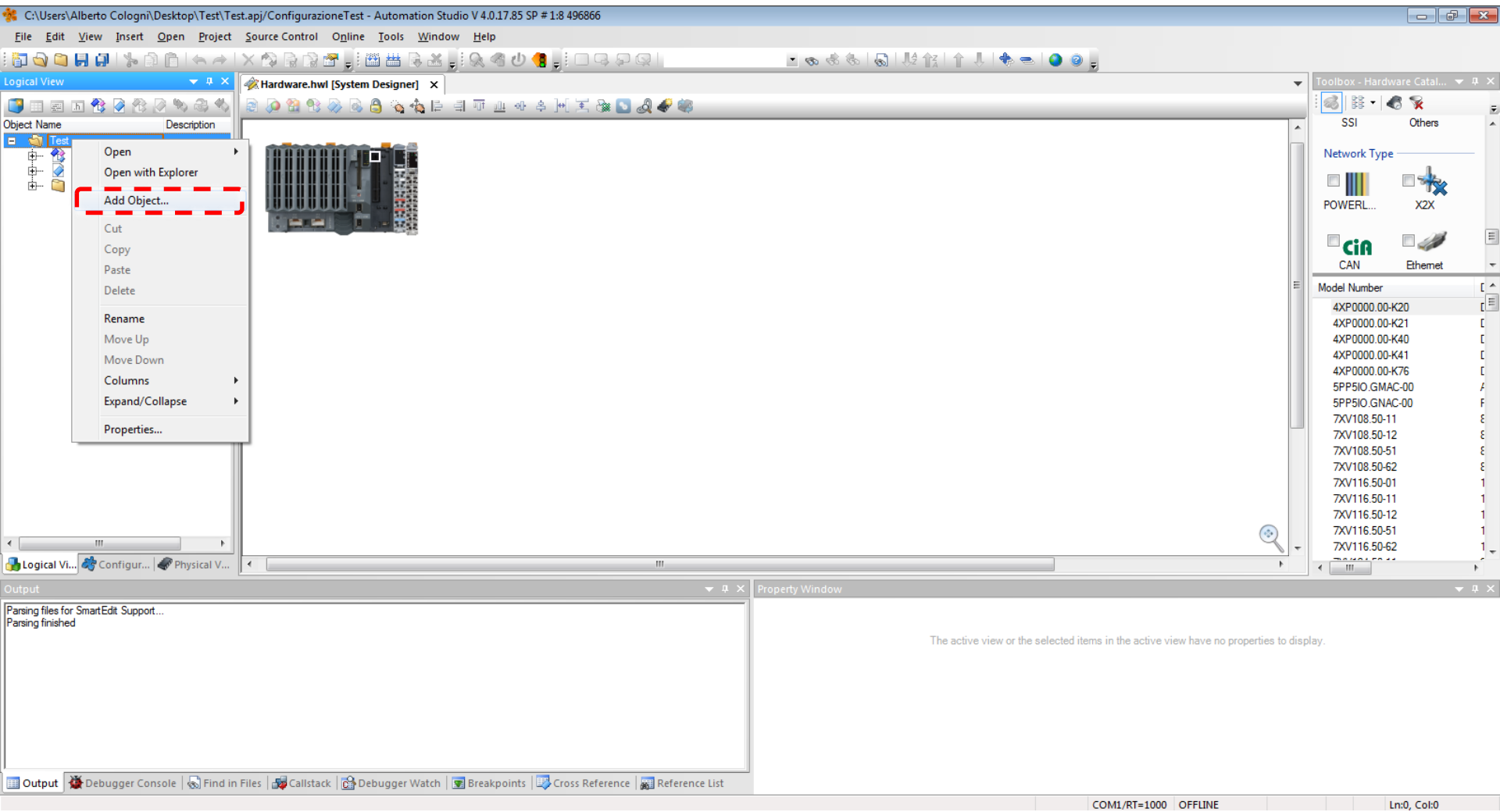
Automation Studio



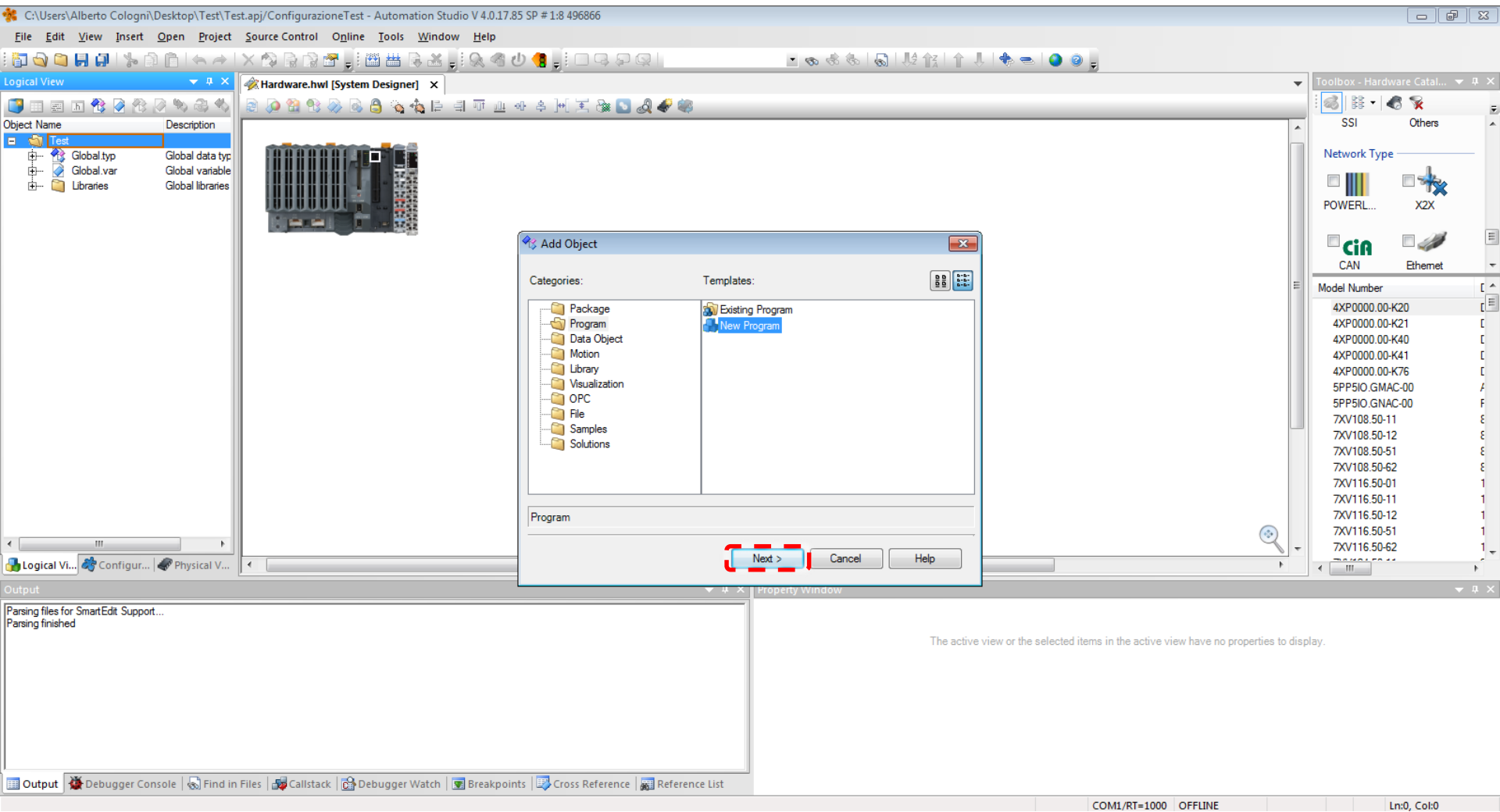
Automation Studio

Ad esempio andiamo ad inserire un modulo con due ingressi analogici 0 – 20 mA
Con il pulsante destro è possibile configurare i singoli moduli
N.B.: questi settaggi fanno parte della Configuration

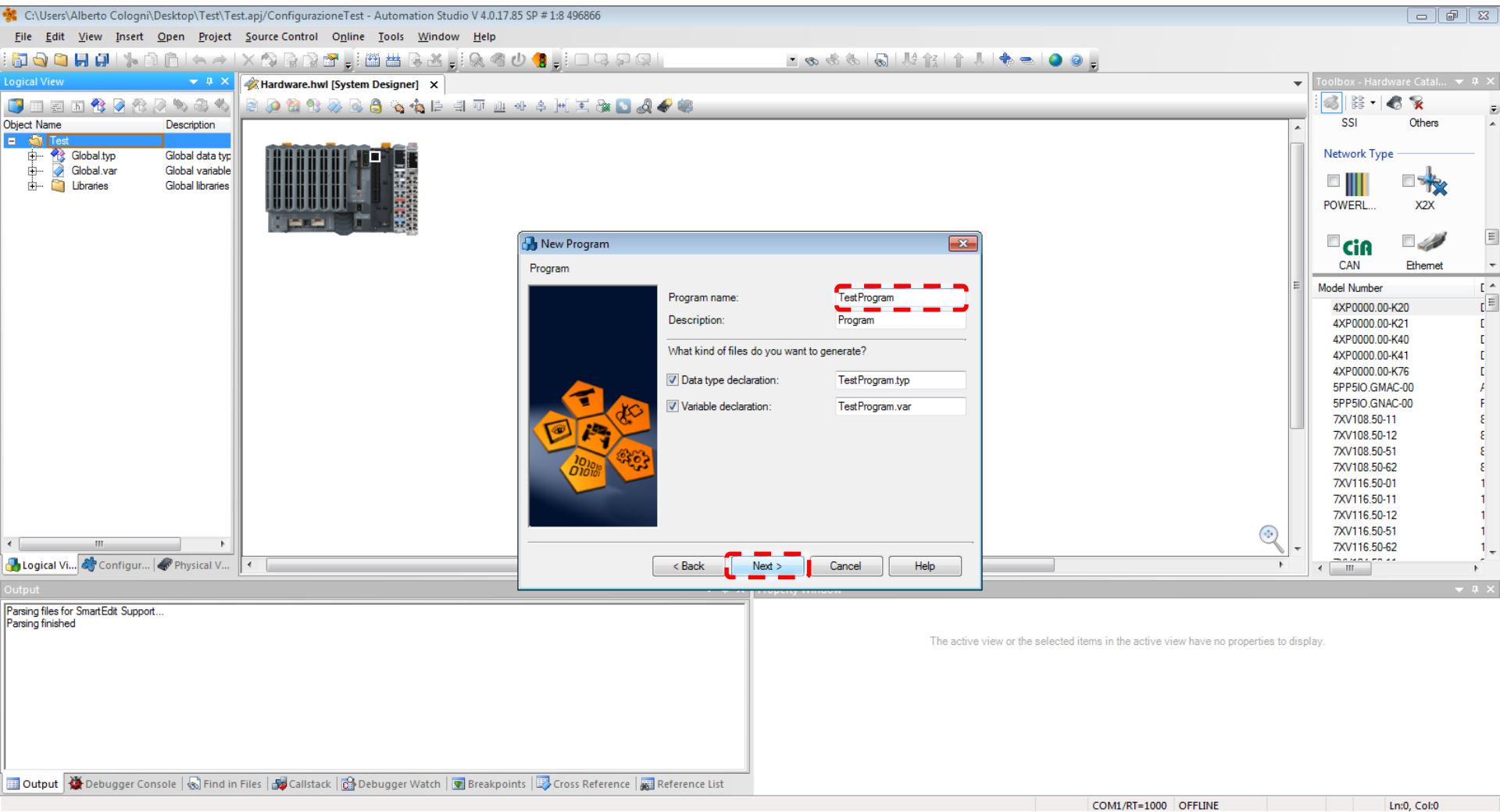
Automation Studio



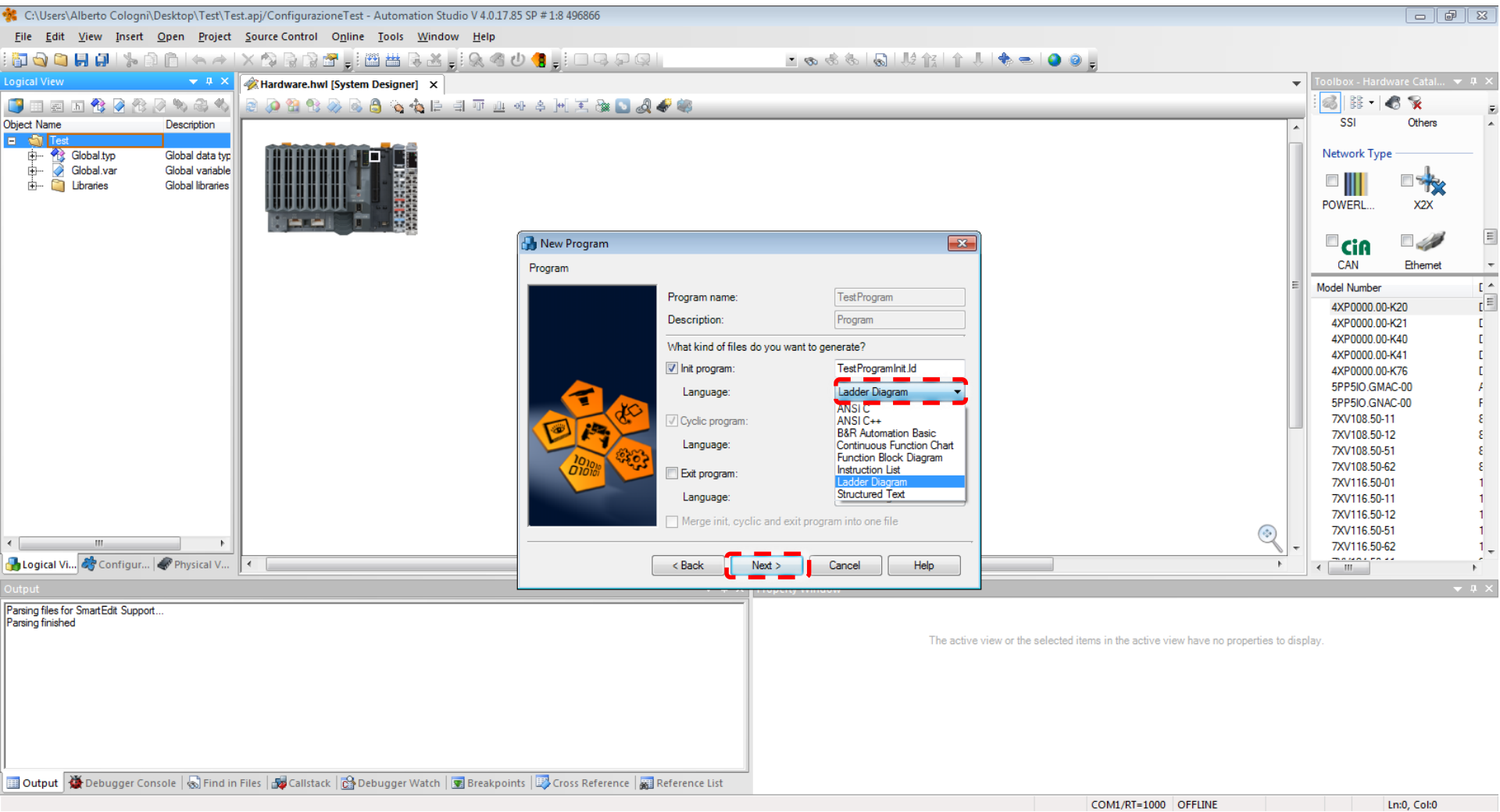
Automation Studio



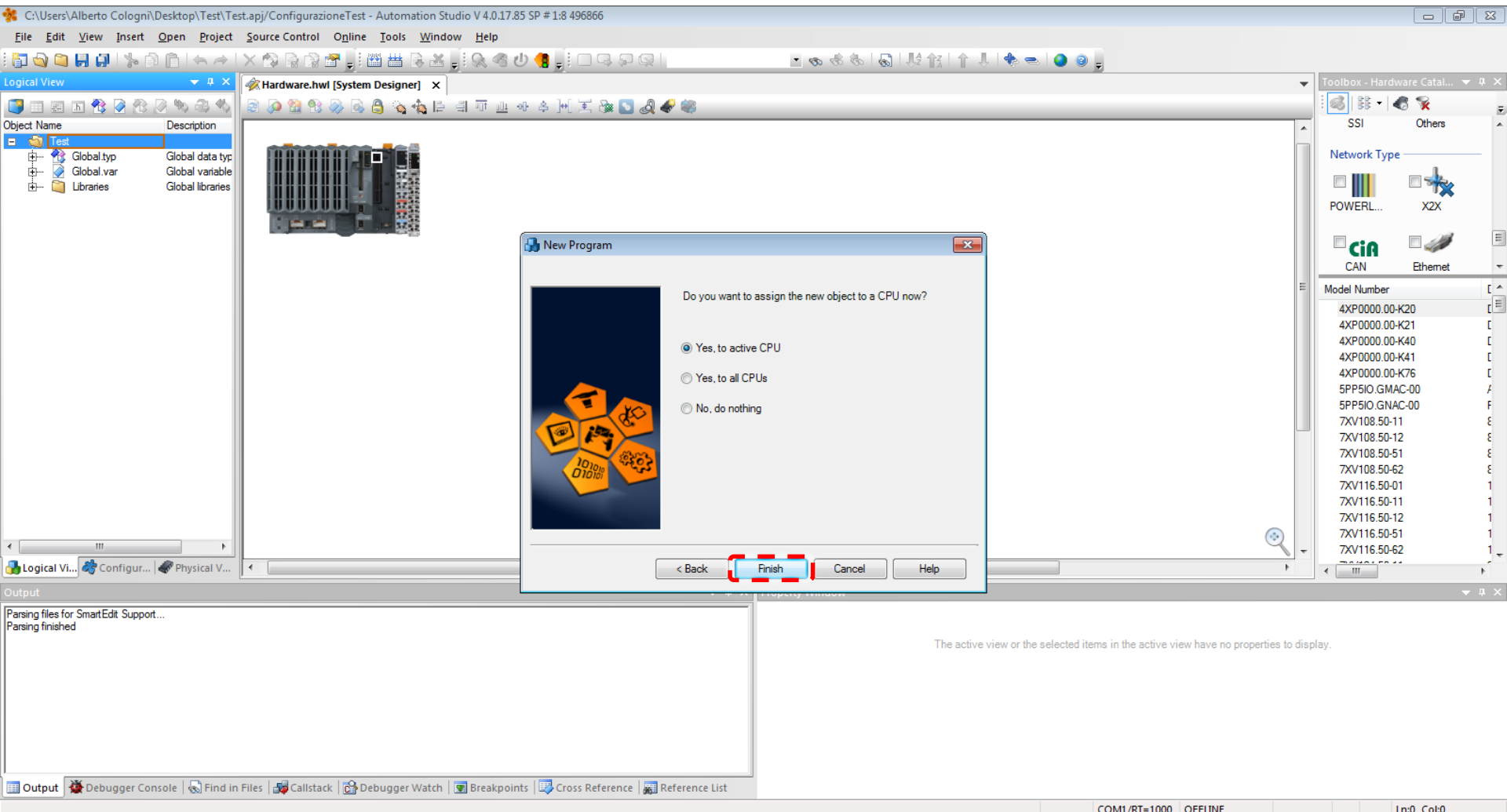
Automation Studio



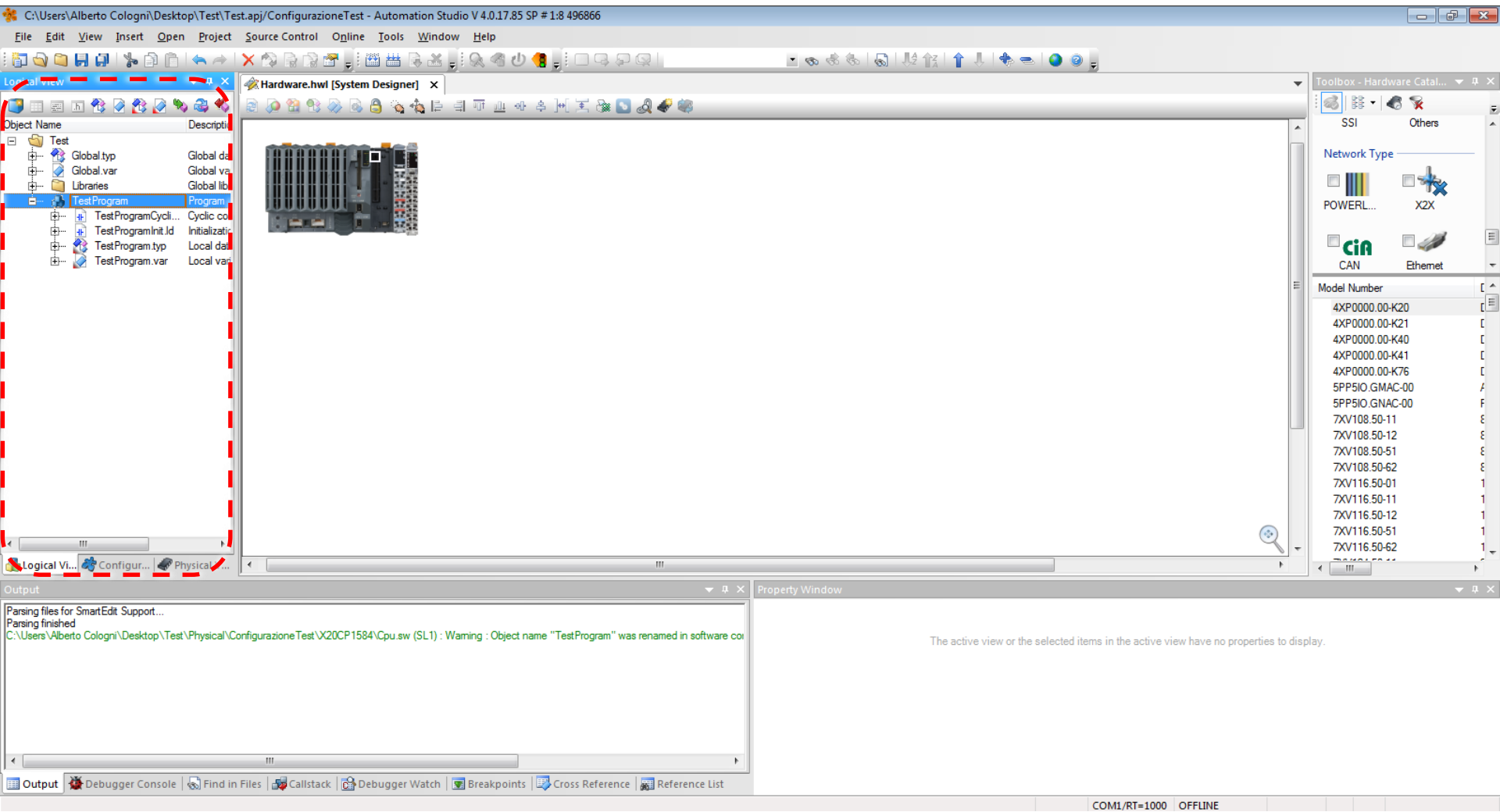
Automation Studio



Automation Studio



Automation Studio



Automation Studio

The screenshot displays the Automation Studio V 4.0.17.85 SP #1:8 496866 interface. The main window shows the 'Cpu.sw [Software]' project. The 'Object Name' table lists various cyclic tasks and data objects. A red dashed box highlights the 'TestProgram' task, which is scheduled at 100 ms. The 'Output' window at the bottom shows a warning message: 'Warning : Object name "TestProgram" was renamed in software co...'. The 'Property Window' at the bottom right indicates that the active view or selected items have no properties to display.

Object Name	Version	Transfer To	Size (bytes)	Source	Source File	Description
Cyclic #1 - [10 ms]						
Cyclic #2 - [20 ms]						
Cyclic #3 - [50 ms]						
Cyclic #4 - [100 ms]						
Cyclic #5 - [200 ms]						
Cyclic #6 - [500 ms]						
Cyclic #7 - [1000 ms]						
Cyclic #8 - [10 ms]						
Data Objects						
No Data Objects						
Visualization						
Binary Objects						
Library Objects						
Source Objects						
Configuration Objects						

In questa finestra è mostrato lo scheduler del PLC, com'è possibile notare il nuovo Task p stato posizionato in una ciclica a 100 ms. E' possibile modificare la posizione o la frequenza del task usando il pulsante destro sulla ciclica

Automation Studio

