

LOGIC LAB

AXEL

SOLUZIONI SOFTWARE PER L'AUTOMAZIONE INDUSTRIALE



 un progetto italiano

[www.axelsw.it](http://www.axelsw.it)



## La garanzia e l'economia degli standard più diffusi sul mercato

**LogicLab** è un ambiente di sviluppo **IEC 61131-3**, il più diffuso standard per la programmazione di controllori industriali, progettato per soddisfare le esigenze e adattarsi alle competenze di chi sviluppa applicazioni industriali.

Con **LogicLab** è inoltre semplice riutilizzare, come blocchi di libreria disponibili al programmatore, componenti software scritti in linguaggio C/C++, standard de facto nella programmazione di sistemi embedded.

Insomma, **LogicLab** consente di **preservare il valore del software già prodotto dalla Vostra azienda** e di usufruire delle competenze già acquisite dal Vostro personale.

## Un unico strumento per l'intera gamma di prodotti

**LogicLab** si accompagna ad un run-time estremamente parsimonioso nell'uso delle risorse hardware e facilmente portabile su qualunque sistema. **LogicLab** è stato già portato con successo **su numerose architetture a 32, 16 e persino 8 bit.**

Questa versatilità dà la possibilità alla Vostra azienda di fornire un unico strumento di programmazione per un'intera gamma di prodotti, rendendo così più economica per i Vostri clienti l'adozione di ulteriori elementi, poiché potranno lavorare con uno strumento già familiare e riutilizzare le applicazioni scritte per i prodotti acquistati in precedenza.

## Massima efficienza per le applicazioni più esigenti

Il compilatore converte le applicazioni scritte secondo lo standard IEC 61131-3 direttamente in **linguaggio macchina**, evitando la necessità di un interprete run-time e rendendo l'esecuzione dell'applicazione il più efficiente possibile.

Questa caratteristica rende **LogicLab** adatto anche per le **applicazioni time-critical** (ad esempio, motion control) e può essere un valido supporto al Vostro tentativo di raggiungere nuove fasce di mercato.



# AXEL

## Le funzionalità del software rendono inutile l'acquisto di hardware aggiuntivo

LogicLab integra un **potente sistema di debug**, che permette il campionamento di segnali in rapido cambiamento direttamente sul sistema target, garantendo accuratezza ed affidabilità dell'informazione senza richiedere l'acquisto di hardware dedicato.

È inoltre dotato di un **simulatore integrato**, per verificare la propria applicazione anche in assenza dell'hardware.

## La flessibilità di un software su misura per i Vostri prodotti

L'integrazione di un browser Web e la disponibilità di una potente interfaccia di programmazione rendono semplice l'estensione di LogicLab con **funzionalità dedicate al Vostro prodotto**.

È dunque possibile corredare l'ambiente di sviluppo con tutti gli strumenti di semplificazione disponibili in una **soluzione software custom**, appoggiandosi però su di un prodotto aderente agli standard e con un'evoluzione indipendente dai propri investimenti.

## Il cardine di una suite omogenea di strumenti software che copre tutte le esigenze di automazione

LogicLab si integra perfettamente con gli altri prodotti Axel, inclusi un software per la costruzione di interfacce uomo-macchina (PageLab), strumenti di configurazione e di configurazione di rete, un simulatore (SimuLab) e un ambiente di esecuzione Soft PLC (LLExec), offrendo una **suite software completa** capace di soddisfare tutti i requisiti del Vostro sistema di automazione.



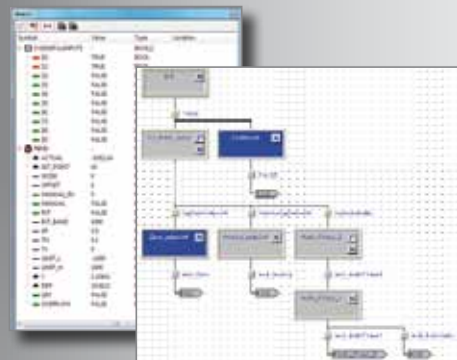
LogicLab viene utilizzato con successo in numerosi campi applicativi, dall'automotive alla climatizzazione passando per il settore energetico, l'automazione industriale e altro ancora.



### Linguaggio IEC 61131-3

Supporto dei 5 linguaggi IEC 61131-3, di tutti i tipi di dato standard, di dati strutturati, di array e dei meccanismi di definizione di tipi di dato derivati.

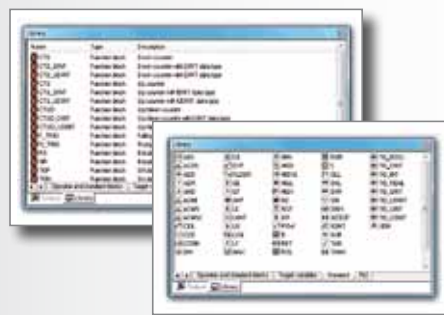
Estensioni orientate alla programmazione di sistema, con puntatori e macro.



### Live debug

Animazione di schemi LD, con illuminazione dei contatti, ed SFC, con illuminazione delle fasi attive.

Visualizzazione del valore attuale delle variabili per tutti i linguaggi. Watch window con funzione di forzatura valori e supporto dei tipi di dati complessi.



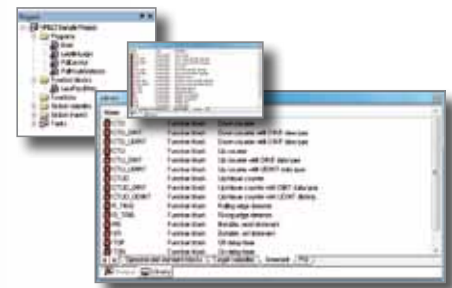
### Librerie

Possibilità di creazione di librerie di oggetti (POU), di variabili e di definizioni di dati. Inclusione nel progetto di un numero illimitato di librerie con possibilità di visione del codice sorgente (se non protetto) e di debug all'interno dei blocchi. Import/export di parti di progetto, import di codice sorgente testuale proveniente da altri sistemi di sviluppo e/o editor.



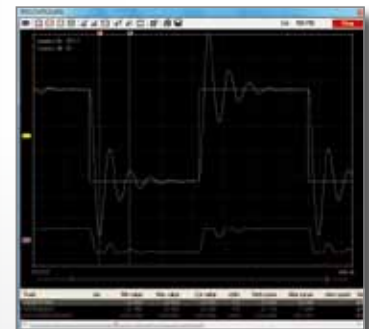
### Trigger e breakpoint

Funzione di trigger che permette di visualizzare il valore di più variabili in un preciso punto di esecuzione individuato sul sorgente, senza arrestare l'applicazione. Disponibilità di breakpoint in numero illimitato.



### Codice sorgente e crittografia

Funzione di upload/download del codice sorgente, eventualmente criptato, sul sistema target. Crittografia dell'intero progetto o di parti di esso. Possibilità di creare librerie crittografate per la distribuzione a clienti finali di codice protetto.



### Trace real-time

Campionamento eseguito dal sistema target dell'evoluzione del valore di un massimo di 8 variabili in un preciso punto del programma. I dati acquisiti sono poi visualizzati in forma grafica con possibilità di zoom, misura e salvataggio su file.

## PROGRAMMARE CON I LINGUAGGI IEC 61131-3

Lo standard IEC 61131-3 definisce 5 linguaggi di programmazione con caratteristiche e finalità distinte, ma che possono essere usati insieme all'interno di un'applicazione. In questo modo, è sempre possibile utilizzare il linguaggio più adatto al problema di programmazione che si sta affrontando.

### Ecco i nostri consigli

#### Instruction List (IL)

Il linguaggio IL si colloca a basso livello, con una struttura simile ai linguaggi assembler. È ideale per programmi caratterizzati da un numero limitato di punti di decisione, condizioni e cambiamenti nel flusso di esecuzione. Il suo uso è consigliato ove il tempo di esecuzione sia un fattore critico.

```

01  sigm
02  sigm
03  sigm
04  sigm
05  sigm
06  sigm
07  sigm
08  sigm
09  sigm
10  sigm
11  sigm
12  sigm
13  sigm
14  sigm
15  sigm
16  sigm
17  sigm
18  sigm
19  sigm
20  sigm
21  sigm
22  sigm
23  sigm
24  sigm
25  sigm
26  sigm
27  sigm
28  sigm
29  sigm
30  sigm
31  sigm
32  sigm
33  sigm
34  sigm
35  sigm
36  sigm
37  sigm
38  sigm
39  sigm
40  sigm
41  sigm
42  sigm
43  sigm
44  sigm
45  sigm
46  sigm
47  sigm
48  sigm
49  sigm
50  sigm
51  sigm
52  sigm
53  sigm
54  sigm
55  sigm
56  sigm
57  sigm
58  sigm
59  sigm
60  sigm
61  sigm
62  sigm
63  sigm
64  sigm
65  sigm
66  sigm
67  sigm
68  sigm
69  sigm
70  sigm
71  sigm
72  sigm
73  sigm
74  sigm
75  sigm
76  sigm
77  sigm
78  sigm
79  sigm
80  sigm
81  sigm
82  sigm
83  sigm
84  sigm
85  sigm
86  sigm
87  sigm
88  sigm
89  sigm
90  sigm
91  sigm
92  sigm
93  sigm
94  sigm
95  sigm
96  sigm
97  sigm
98  sigm
99  sigm
100 sigm

```

#### Structured Text (ST)

Il linguaggio ST è ad alto livello, con una sintassi simile al linguaggio Pascal.

Ha una vasta gamma di costrutti per l'assegnamento di valori alle variabili, per la creazione di espressioni, per la valutazione di condizioni (IF, CASE) e l'implementazione di iterazioni (FOR, WHILE, REPEAT).

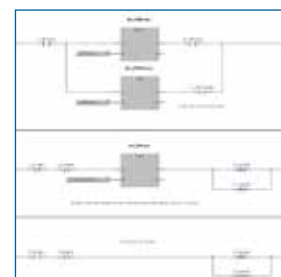
```

// Il linguaggio ST è ad alto livello, con una sintassi simile al linguaggio Pascal.
// Ha una vasta gamma di costrutti per l'assegnamento di valori alle variabili,
// per la creazione di espressioni, per la valutazione di condizioni (IF, CASE)
// e l'implementazione di iterazioni (FOR, WHILE, REPEAT).

```

#### Ladder Diagram (LD)

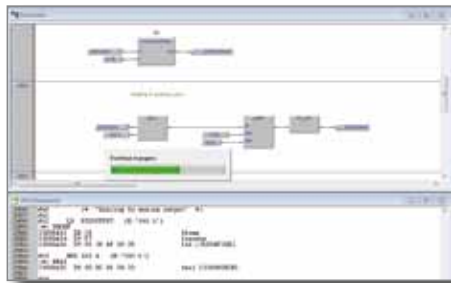
La rappresentazione di una sequenza logica mediante il linguaggio LD nasce dalla progettazione della logica a relé nell'ambito dell'ingegneria degli impianti elettrici. È una rappresentazione particolarmente idonea all'implementazione di operazioni su segnali digitali o variabili booleane.





### Framework

Supporto alla customizzazione dell'ambiente di sviluppo al fine di supportare le caratteristiche specifiche dei target programmabili. Possibilità di inserire e gestire componenti grafici (HTML, VB, C++ ecc.), script (JavaScript) e strutture dati in formato XML. Interazione con LogicLab attraverso l'interfaccia di automazione OLE dedicata.

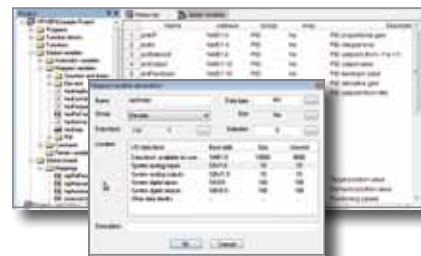


### Compilatore

Generazione di codice macchina ottimizzato per tutti i processori supportati. Generazione di codice multi-tasking con numero di task illimitato.

### Download hot-swap

Compilazione incrementale e funzione di download del codice "a caldo" senza dover riavviare l'applicazione e il sistema target.



### Variabili e blocchi di sistema

Accesso diretto alle variabili del sistema target con o senza immagine di processo. Interfacciamento diretto con funzioni del sistema target (tipicamente in codice C).

### Allocazione dati e codice

Il codice può essere eseguito indifferentemente in RAM o in ROM. È supportata l'allocazione dei dati in ogni tipo di memoria fisica (RAM, EEPROM ecc.) o virtuale.



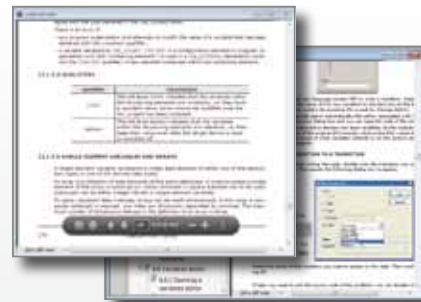
### Interazione con il sistema on-line

Monitor dell'esecuzione del run-time PLC, con possibilità di controllare l'esecuzione dei singoli task, integrato nell'ambiente di sviluppo.



### Simulatore

Permette di eseguire su PC l'intera applicazione e può essere esteso con DLL Windows standard per riprodurre fedelmente il comportamento del sistema. Dispone di pannelli per la simulazione dell'I/O del dispositivo reale.



### Manuali

Manuale aggiornato, in formato PDF stampabile e contestualizzato. Supporto della documentazione specifica delle applicazioni, delle librerie e dei sistemi target.

### Function Block Diagram (FBD)

Il concetto fondamentale nel linguaggio FBD è il flusso di dati. In questo linguaggio i valori scorrono dagli ingressi alle uscite, attraverso dei blocchi. Il comportamento dei programmi è espresso in termini di blocchi grafici interconnessi, in analogia agli schemi elettrici o agli schemi a blocchi dei sistemi di controllo.



### Sequential Function Chart (SFC)

Il linguaggio SFC permette di descrivere il comportamento di un programma in termini di stati e transizioni. Questo linguaggio consente lo sviluppo di un'applicazione con la metodologia top-down. Lo schema SFC costituisce infatti la struttura del programma di controllo, mentre le singole azioni e transizioni sono poi implementate in uno qualsiasi dei linguaggi IEC 61131-3.



## Per maggiori informazioni

Dall'area download del nostro sito Web, è possibile scaricare gratuitamente l'ultima versione di LogicLab. Potrete provare tutte le caratteristiche dell'ambiente di sviluppo, comprese quelle online, con un target virtuale (Soft PLC).

→ [www.axelsw.it/download](http://www.axelsw.it/download)

Per informazioni su che cosa significhi portare LogicLab sul Vostro prodotto, sia esso un PC industriale oppure un sistema embedded di Vostra produzione, Vi invitiamo a scaricare il materiale informativo dal nostro sito Web.

→ [www.axelsw.it/logiclab](http://www.axelsw.it/logiclab)

Per ogni altra informazione su LogicLab, vogliate contattarci direttamente per telefono o per e-mail.

→ +39 0332 949600

→ [info@axelsw.it](mailto:info@axelsw.it)

AXEL



**AXEL s.r.l.**  
**SOLUZIONI SOFTWARE PER L'AUTOMAZIONE INDUSTRIALE**

Via del Cannino, 3  
21020 Crosio della Valle (VA) Italy  
Tel. +39 0332 949600  
Fax. +39 0332 969315  
info@axelsw.it  
[www.axelsw.it](http://www.axelsw.it)