LOGIC LAB

AXEL

SOFTWARE SOLUTIONS FOR THE INDUSTRIAL AUTOMATION

an Italian project

www.axelsw.it

LOGIC LAB

From simple to the most complex automation.

The guarantee and the convenience of the most widespread standards in the market

LogicLab is an **IEC 61131-3** development environment, **the most widespread standard** for programming industrial controls.
It has been designed in order to satisfy the needs and meet the skills of people who develop industrial applications.
LogicLab makes easy to reuse software components in C/C++ language (de facto standard in embedded systems programming) as library blocks which are available to the programmer.
In short, **LogicLab** allows you to **preserve the value of the software you have already produced** and to benefit from the skills which your staff has acquired.

One tool for the whole range of products

LogicLab comes with a lightweight PLC run-time, which makes an exceptionally reduced use of hardware resources and which is easily portable on any system. Indeed, **LogicLab** has already been successfully ported on **several 32-, 16-, and even 8-bit architectures**.
Such a flexibility allows you to supply a single programming tool for the whole range of products, making cheaper the adoption of further elements for your Customers, as they will be able to work with a tool they already know and reuse the applications which were written for the products they previously purchased.

Highest efficiency for the most demanding applications

The compiler translates applications written according to the IEC 61131-3 standard directly into **machine code**, avoiding the need for a run-time interpreter, thus making the program execution as efficient as possible.
This feature makes **LogicLab** suitable also for **time-critical applications** (e.g., motion control) so it can support you effectively in reaching new markets.

No need to buy additional hardware

LogicLab is endowed with a **powerful set of debugging tools** which allows the sampling of fast changing data directly on the target environment, ensuring the information is accurate and reliable, without the need to purchase dedicated hardware.
Moreover, **LogicLab** is equipped with an **embedded simulator** in order to test, debug, and validate the application also without the physical device.

The flexibility of software tailored for your products

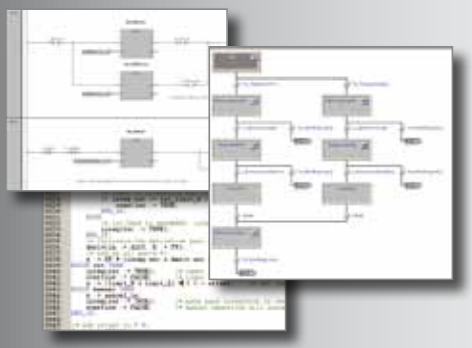
The integration of a Web browser and the availability of a powerful programming interface make easy to extend **LogicLab** with new **features dedicated to your product**.
So it is possible to equip the development environment with all the simplification tools usually available in a **custom software solution**, but built on a standard-compliant product evolving independently with respect to your investments.

The foundation of a homogeneous software suite covering all the automation needs

LogicLab smoothly integrates with other Axel products, including a human-machine interface builder (PageLab), device and network configuration tools, an emulator (SimuLab), and a Soft PLC execution environment (LLExec), providing a **comprehensive software suite** able to meet the most challenging requirements of your automation system.

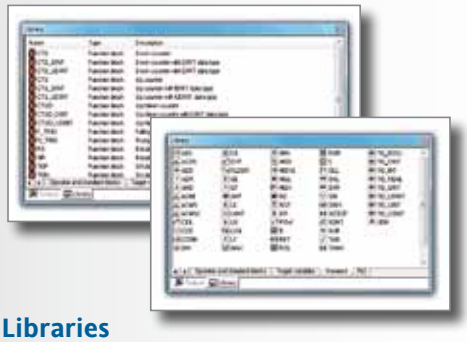



LogicLab is used successfully in several sectors, including: automotive, HVAC/R, energy, industrial automation and many more.



IEC 61131-3 language

Support of all IEC 61131-3 programming languages, standard data types, structures, arrays, and user-defined data types.
Extensions for system programming in IEC 61131-3, including pointers and macros.

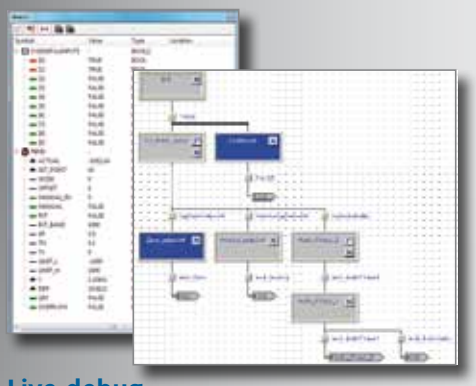


Libraries

Management of libraries of POUs, type definitions, and variables. The project links an unlimited number of libraries, allowing the developer to view and debug the source code (if not protected). Import/Export of single project elements and import of textual source code from other development systems and/or editors is also supported.

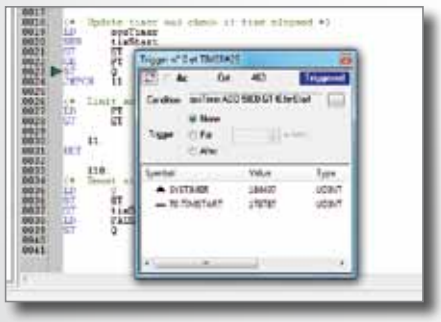
Source code and encryption

Upload/Download of (potentially encrypted) source code from/to the target system.
Encryption of either the whole project or single project elements. Creation of encrypted libraries for distributing the protected code to final Customers.



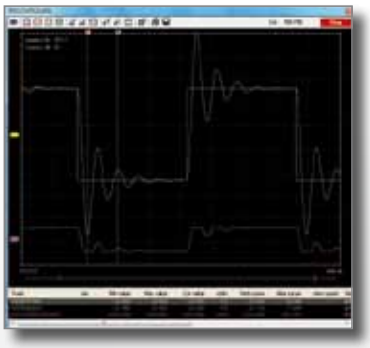
Live debug

Animation of LD schemata, with contact/coil highlighting and SFC schemata with active steps highlighting. Display of the current value of the variables for all the languages. Watch window that allows the forcing of watch values and supports complex data types.



Triggers and breakpoints

Trigger function which allows to display the value of several variables in a precise execution point on the source code, without stopping the application. An unlimited number of breakpoints is available.



Trace real-time

Sampling, directly on the target environment, of up to 8 variables in a precise execution point on the source code.
Acquired data is then displayed in a graphic which allows zooming, measures and saving to file.

PROGRAMMING WITH IEC 61131-3 LANGUAGES

IEC 61131-3 standard defines 5 programming languages which have different features and purposes, but can be used together within an application. In this way, you can always use the most suitable language for the programming problem you are facing.

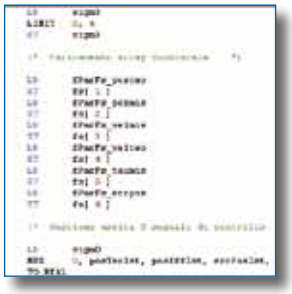
Here are our advices

Structured Text (ST)

ST language is a higher-level language, whose syntax resembles that of Pascal.
It provides a wide range of language constructs: value assignment, expressions of arbitrary complexity, selection (IF, CASE) and iteration (FOR, WHILE, REPEAT) statements.

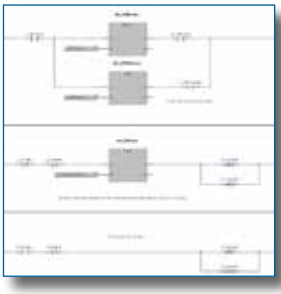
Instruction List (IL)

IL language is a low-level assembly-like programming language. It is ideal for programs with few decision points, conditions, and changes in the execution flow. Its use is recommended when the execution time is considered a critical element.



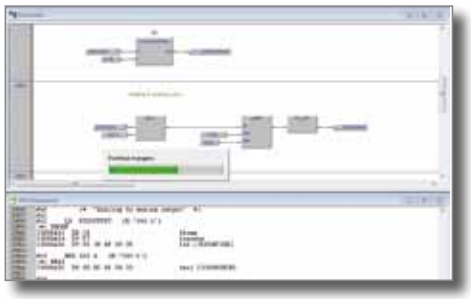
Ladder Diagram (LD)

The representation of a logical sequence by means of LD language starts from relay logic design within the engineering of electric installations.
This representation is particularly suitable for the implementation of operations on digital signals or boolean variables.



Framework

Support to the customization of the development environment in order to support specific features of the programmable targets. Chance to insert and manage graphic components (HTML, VB, C++, etc.), script (JavaScript), and data structures in XML format. Interaction with LogicLab through its OLE automation interface.



Compiler

Generation of optimized machine code for all the supported microprocessors. Generation of multi-tasking code with unlimited tasks.

Download hot-swap

Incremental compilation and code download without the need to restart the application and the target system.

Target variables and embedded blocks

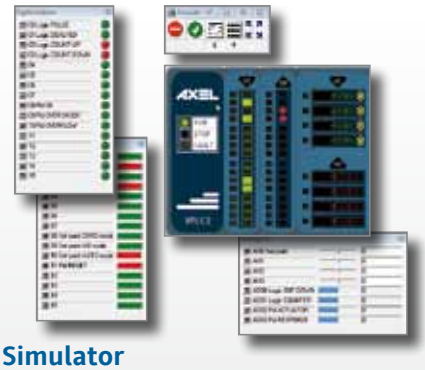
Direct access to system variables with or without process image. Direct interface with target system's functions (typically in C code).

Data and code allocation

The code may be executed either in RAM or in ROM. Data allocation in every kind of both physical and virtual memory (RAM, EEPROM, etc.) is supported.

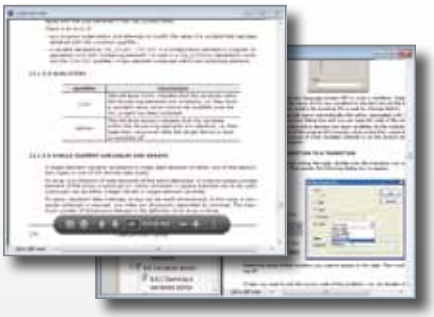
Interaction with on-line system

Embedded monitor of the PLC run-time execution, chance to check the execution of single tasks.



Simulator

It allows to execute and debug the whole application on the PC and it can be extended with standard Windows DLLs in order to reproduce exactly the system behaviour.



User Manuals

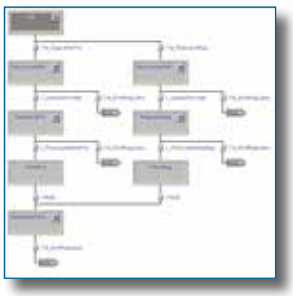
Up-to-date printable user manual in PDF format, integrated in the IDE as context-sensitive help. Support of specific documentation of applications, of libraries, and of target systems.

Function Block Diagram (FBD)

The basic idea of FBD language is the data flow. In this language values flow from entry to exit points through some blocks.
Programs behaviour is expressed in terms of interconnected graphic blocks, similarly to wiring diagrams or block diagrams of control systems.

Sequential Function Chart (SFC)

SFC language allows to describe the behaviour of a program in terms of states and transitions.
This language allows the development of an application by means of the top-down methodology. In fact the SFC schema makes up the structure of the control program, while the single actions and transitions are implemented in any of the IEC 61131-3 languages.



For further information

In the download area of our Web site you can freely download the latest LogicLab release. You can try the whole set of features of the development environment, including on-line ones, with a virtual target (soft PLC).

→ www.axelsw.it/download

For further information about porting LogicLab on your product, whether it is an industrial PC or a self-manufactured embedded system, we invite you to download the relative documentation from our Web site.

→ www.axelsw.it/logiclab

For any other information about LogicLab, please contact us directly by phone or by e-mail.

→ +39 0332 949600

→ info@axelsw.it

AXEL s.r.l.
SOFTWARE SOLUTIONS FOR THE INDUSTRIAL AUTOMATION

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