TECHNOLOGY SOLUTIONS

TURNING YOUR PRODUCT VISION INTO REALITY
www.oem.net.au

OEM TECHNOLOGY SOLUTIONS PTY LTD | ABN 40 113 389 852
UNIT 13, 82 RESERVE ROAD, ARTARMON NSW 2064 AUSTRALIA
T: + 61 2 9966 9424 | F: +61 2 9966 9429

ISAGRAF V3.5 OVERVIEW

Document Number: TD-0004
Issue Status: 1

Prepared By:
OEM TECHNOLOGY SOLUTIONS PTY LTD
UNIT 13, 82 RESERVE ROAD
ARTARMON NSW 2064
AUSTRALIA

TABLE OF CONTENTS

1. WHAT IS ISAGRAF?

With the increased demand for standard hardware and programming systems, international standards such as IEC 61131-3 have been developed to meet this demand. The IEC 61131-3 outlines the requirements for uniformity of industrial programming languages, specifically targeted at PLC's (Programmable Logic Controllers).

ISaGRAF was the first Windows based development environment to fully support all five of the PLC languages:

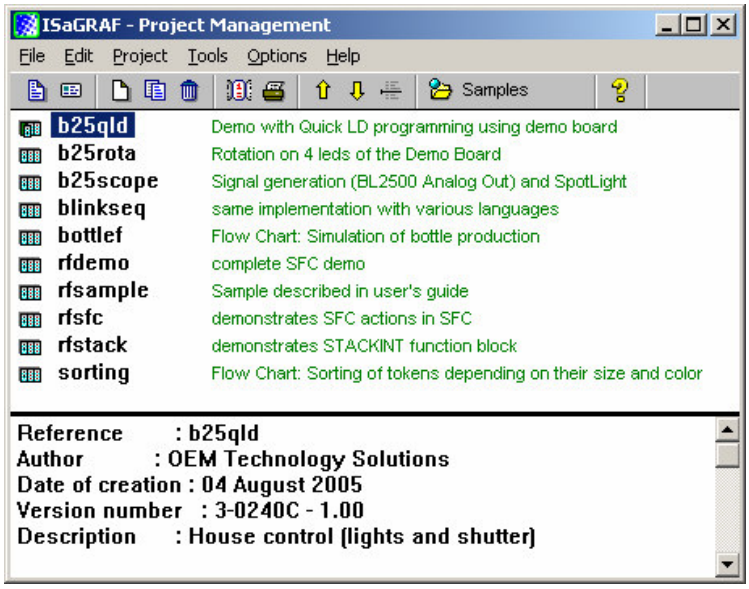
- Sequential Function Chart (SFC)
- Function Block Diagram (FBD)
- Ladder Diagram (LD)
- Structured Text (ST)
- Instruction List (IL)

Additionally, for the ultimate in power and flexibility, ISaGRAF supports functions and function blocks written in C and/or IEC 61131-3 languages. The ISaGRAF workbench provides a fully featured set of tools for programming Rabbit Semiconductor controllers providing editing, debugging, code generation, documentation, library management, archiving, on-line monitoring, off-line simulation and on-line change of projects made for execution by the ISaGRAF runtime kernel. The workbench runs on Windows 3.1, 3.11, 95 or NT, OS/2 and all Windows emulation packages. The ISaGRAF workbench and documentation are available in English, French, German and Japanese, with other languages soon to be available.

2. ISAGRAF PRODUCT OVERVIEW

2.1 PROJECT MANAGER

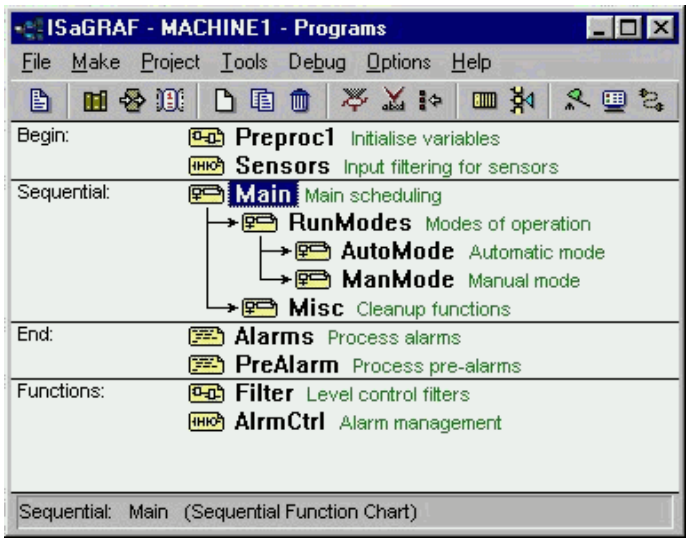
ISaGRAF begins with the preparation of a detailed project specification. This step is common to all PLC programming techniques and is usually the result of a thorough analysis of the project and consultation between the integrator and the customer. Distributed applications can easily be represented as several ISaGRAF projects grouped together in the project list. These applications can be edited, simulated or debugged on the same Window screen.



2.2 PROGRAM MANAGER

With the program manager the application specification is divided into smaller functional modules. The exact operation that is to be performed within each module is defined at that stage.

ISaGRAF's program management facilities allow the user to define each of these modules, their operations and their interaction to form the complete application.



2.3 VARIABLE DEFINITION

All variables are declared or imported in the ISaGRAF dictionary. When this step is achieved, during programming, a mouse click will insert the variable in the program.

Any external database can be imported to build the ISaGRAF dictionary. A quick declaration allows many variables to be declared with one command and an easy to use mapping tool is provided for Modbus-based SCADA systems such as Indusoft®.

2.4 FUNCTIONAL MODULE PROGRAMMING

The next step in the ISaGRAF methodology is the actual programming of the various functional modules. This can be done using any one of the six supported languages:

- Sequential Functional Chart (SFC)
- Function Block Diagram (FBD)
- Ladder Diagram (LD)
- Structured Text (ST)
- Instruction List (IL)
- Flow Chart (FC)

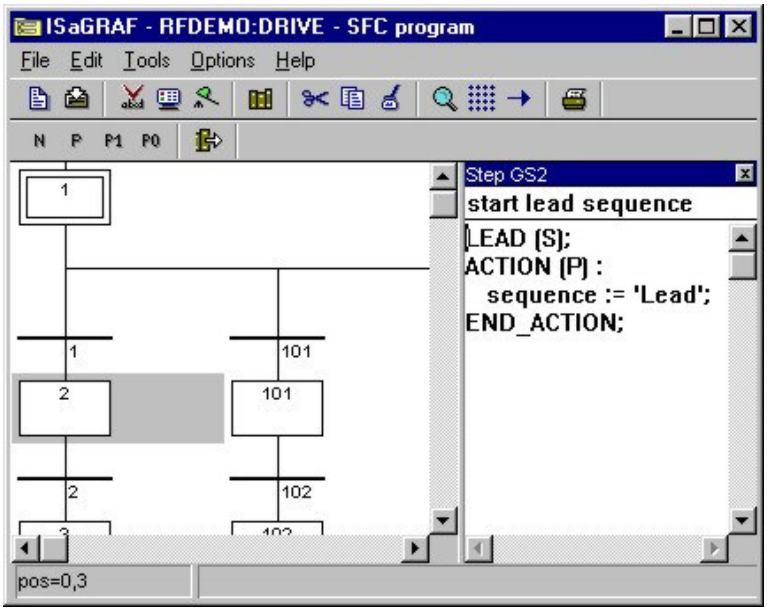
2.5 IEC 61131-3 LANGUAGE EDITORS

In February 1993, responding to the need for standards to reduce training costs and guaranteed portability, the IEC issued the IEC 61131-3 standard: a specification of five PLC programming languages that can be freely mixed to define automation and control procedures.

In August 1996, ISaGRAF version 3.2 received the certificate of PLC open compliance class of IEC 61131-3 (base level IL).

2.5.1 Sequential Function Chart Editor

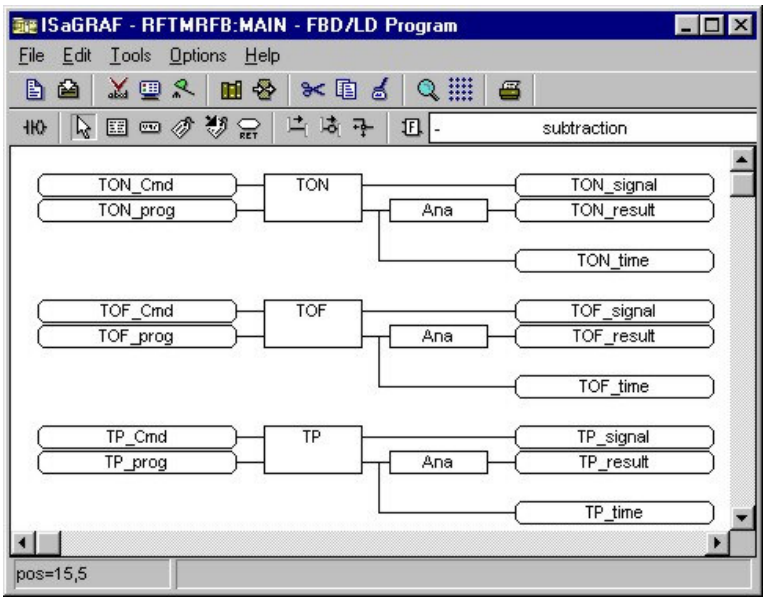
SFC divides the process cycle into a number of well defined steps, separated by transitions. SFC is the core language of the IEC 61131-3 standard. The other languages are used to describe the actions performed within the steps and the logical conditions for the transitions. Parallel processing can easily be described using SFC.



2.5.2 Function Block Diagram Editor

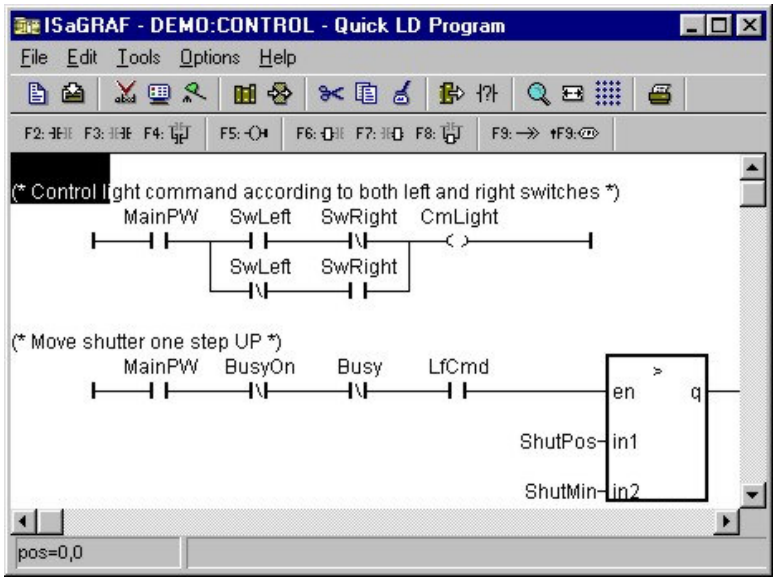
The FBD is a graphical language that allows the user to build complex procedures by taking existing function blocks from the ISaGRAF library and wiring them together on the screen.

ISaGRAF includes a library with more than 60 blocks ready to use. Users can enlarge this library by writing functions and function blocks in LD/FBD/ST/IL or C.



2.5.3 Ladder Diagram Editor

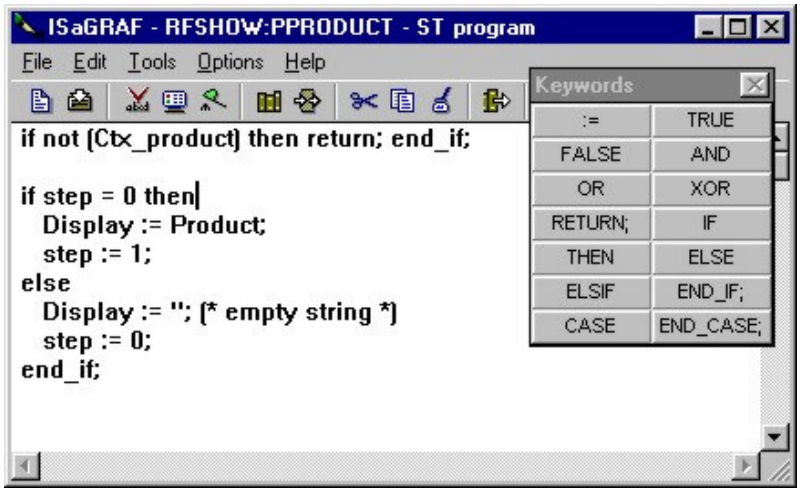
The ladder diagram is one of the most familiar methods of representing logical equations and simple actions. The ISaGRAF ladder diagram editor offers the best compromises between high level graphic capabilities and easy to use keyboard driven programming.



2.5.4 Structured Text Editor

Structured text is a high level structured language with a syntax similar to Pascal, but more intuitive to the automation engineer.

This language is mainly used to implement complex procedures that cannot be easily expressed with graphical languages (FOR, WHILE, etc.).

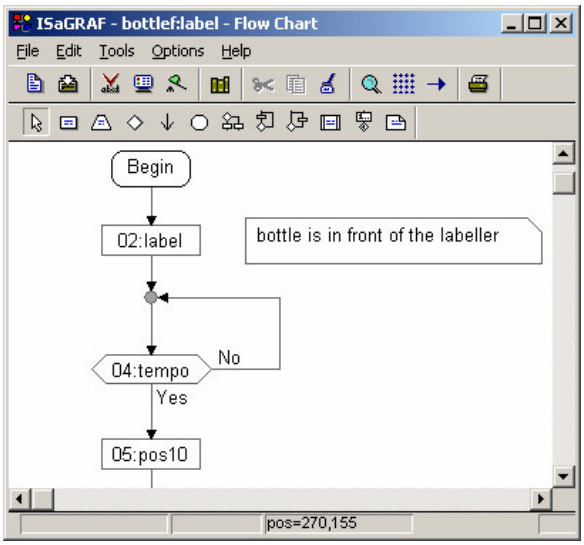


2.5.5 Instruction List Editor

Instruction list is a low level language similar to the simple textual PLC languages.

2.6 FLOW CHART EDITOR

In addition to the five IEC 61131-3 languages ISaGRAF implements another graphical language, Flow Chart. Flow Chart is a decision diagram, which can also be used to describe sequential operations.



2.7 ANSI C EDITOR

Additionally, for ultimate power and flexibility ISaGRAF supports ANSI C. Functions and function blocks written in C can be called directly from any of the six supported languages. The routines become an extension of the ISaGRAF languages.

2.8 DOCUMENT GENERATOR

ISaGRAF features a self-documentation capability that can automatically generate the project's most current project description, project architecture, history of modifications, I/O wiring lists, dictionaries and cross references.

2.9 SIMULATION

Without any target hardware platform, the programmer can validate the complete application in the office. With ISaGRAF's powerful simulator on the workbench, the user can perform structural and functional tests of each module separately or on the global application.

The simulator makes it easy to trace the program execution and to see the status of any internal variable. The I/O hardware can be fully simulated and internal status and variables manual forced by the user.

During simulation, editors can be opened in debug mode to see how programs are executed.