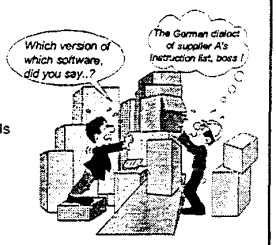
The application of IEC 61131-3 to industrial control

Tuesday March 30th 1999

Tony Ciardiello
Schneider Electric

Current programming environments

- Large variety of programming techniques
- Large number of different programming dialects
- No software portability between different platforms
- Lack of additional programming tools
- Inefficiencies in multiple projects through incompatibility
- Vendor driven technology may disregard user needs
- Large dependency on current supplier(s)



IEC 1131-3

The International Standard for Automation Controller Programming Languages

Major Topics

- Transcendence
- Minimising duplication of effort
- Improving software quality
- Practicalities
- Next Steps
- Open Discussion,Questions and Answers

Transcendence

- Transcends Regional, Cultural and Philosophical barriers
- Transcends the "Technology Generation Gap"
- Transcends the "Comfort Factor" barriers
- Transcends the H/W and S/W platform arguments
- Transcends automation "Fashions"

The IEC 1131-3 Standard

- Programming language syntax & behavior document
- Brings many common practices and proven techniques together with new, structural, high-productivity (development) tools
- Provides a well defined suite of inter linked languages for solving a large variety of industrial control problems
- Enhances the quality of application software through well-structured design, data and control code encapsulation and information hiding
- Provides proven solutions to improve productivity and re-usability of code

What is the Benefit of such a Standard ?

- Combining harmoniously different components from different locations, companies, countries or projects 'Groupware' engineering
- Reduced waste of human resources (in training, debugging, maintenance)
- Creating a focus to problem solving via software re-usability (reduced application investment and supplier dependency)
- Reduced misunderstandings and errors
- Programming techniques usable in more environments (general industrial control)
- Additional software tools via independent suppliers

Advantages to Users

- Well structured, 'top-down' or 'bottom-up' program development (via Program Organisation Units, POU)
- Strong data typing prohibiting programming errors
- Support for full execution control
  different parts at different times, rates and in parallel
- Support for complex sequential organisation description (SFC)
- Data structures for easy exchange of associated data elements
- Flexible language selection
  (3 graphical and 2 textual inter linked languages)
- Vendor independent software development

More Advantages to Users

Many IEC programming systems offer:

- graphical programming screens
- support for multiple windows
- mouse operation
- pull-down menu's
- built-in hypertext help function
- software verification during design

(vs. character based systems with cryptic commands)

Benefits of the standard for vendors

- Product recognition and acceptance through standards
- Use of third party modules and tools decrease development cost and improve software quality and time-to-market
- Ability to offer plug-and-play modular products
- Increased total available market (TAM) thru wider application coverage
- Ability to concentrate on differentiating added value

An Automation Supplier's Perspective

- A non-constrictive, self-extensible standard
- Well implemented, 1131-3 methodology is an excellent S/W development productivity enabler
- Provides plenty of room to differentiate offer
- No prejudicial financial entry level
- Encourages innovation?

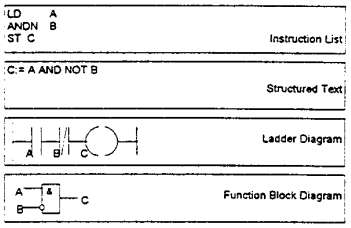
Highlights of IEC 1131-3

- Definition of syntax & behavior of five inter linked editors
- Textual languages (IL, ST) as well as graphical languages (LD and FBD) for application oriented programming
- Sequential Function Chart, SFC, for well structured program design via chains of parallel and sequential actions
- Instruction set in English - internationally accepted
- Symbolic programming ensures easy to read programs
- Choice of standard data types for optimal use of memory and efficient data processing

More highlights of IEC 1131-3

- Structured programming and high level languages to develop and maintain large programs efficiently
- Re-use of tested program modules reduces bugs
- Datacom function blocks reduce effort for external data transfers
- Mathematical functions (trigonometry, exponents)

Representation in different languages



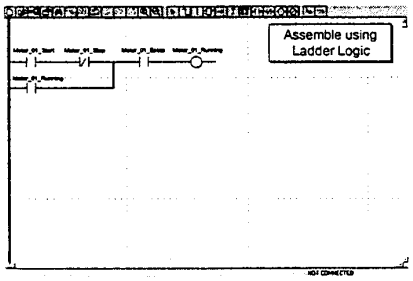
IEC 1131-3 Standard Datatypes

- Bit string types (BOOL, BYTE, WORD, DWORD, WORD)
- Integer types (SINT, INT, DINT, LINT)
- Unsigned integer types (USINT, UINT, UDINT, ULINT)
- Real types (REAL, LREAL)
- Time types (TIME, DATE, TIME_OF_DAY, DATE_AND_TIME)
- Vendor and user defined data types
  - Single element,
  - Array, structure

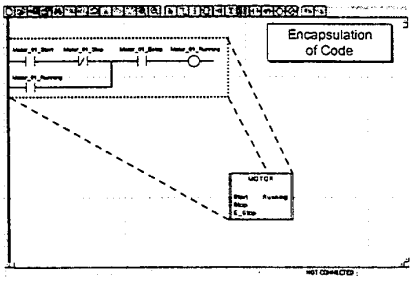
Function Blocks - Software ICs

- Basic building block in packaged form (like hardware IC) or software objects that represent specialised control functions
- Highly re-usable in (different parts of) same program, different programs or projects
- Can be written in any IEC language
- Well defined interface (and so data type) (hidden internals)
- Ability to store data as well as the algorithm
- Describes both the behaviour of data and the data structure
- Base for vendor-independent (neutral) software libraries

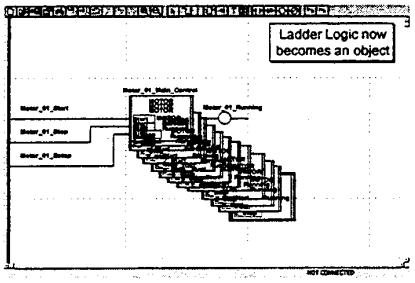
Devices as objects greatly simplifies programming



Devices as objects greatly simplifies programming



Devices as objects greatly simplifies programming



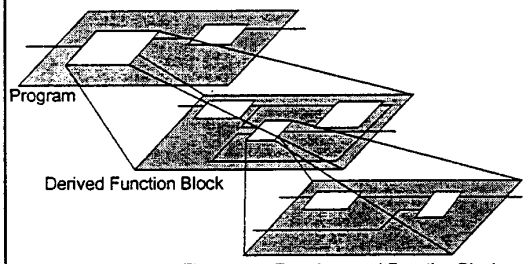
Hierarchical Design

- A program can be defined as a network of Function Blocks and Functions
  - A function Block type definition can be defined using instances of other function block types
    - and so on.....

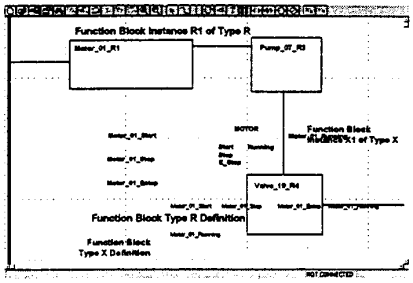
In this way complex programs can be broken down in to

- large function blocks (major areas of functionality)
  - which can be broken down into smaller blocks (focussed functionality)
    - ... till at the bottom FB's are defined using functions, textual statements or provided by standard libraries

Hierarchical Software Structure



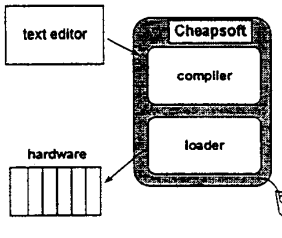
Abstraction



Vendor added value

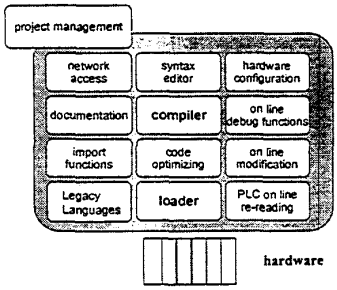
- Implementation options
  - size of pages, rungs
- Associated services
  - debug, searching
  - documentation
  - export
  - languages edition
- Simulation
- Hardware specific functions
- Libraries

Basic IEC implementation system



- This is a compliant system but no services included
- To be compliant is not a quality certification

Enhanced IEC implementation system



This also is an IEC 61131-3 implementation

Marketing Caveman's Decision

"We've decided to postpone inventing the wheel until we agree on what colour it should be!"

Benefits

IEC1131

•Choice of editors provides optimum tool for application
•Reusable functions speed program development
•Simplified program structure reduces commissioning and debugging time

= greater productivity

Next Steps

- Hierarchical structural and functional decomposition
- Abstraction - isolating the user from the underlying magic
- Structured, real-time information,object programming IEC 1499
- Emergence of automation controllers as "content providers" to business systems
- Adoption of commercially available technologies
- Ethernet as the de facto *industrial* fieldbus
- Automation objects embedded intelligence

Any Questions?

Thank You!