The Rosetta Stone of Control Systems - BBBB (Buy)
Programming Industrial Control Systems Using IEC 1131-3 by R. W. Lewis
Reviewed by Nick Sands

There is no way to read R. W. Lewis's book on IEC 61131-3 in an airport bar and not look like an engineer. But this book is worth reading. It is an excellent overview of the five programming languages and the principles on which the standard is based. This standard will change our future, change how control systems are programmed. Lewis has done an excellent job of making a readable book out of the programming standard. There is no information provided on R.W. Lewis himself, but it appears that he was a member of the IEC 61131-3 committee, originally IEC 1131-1 before a renumbering that added the 6.

Lewis starts with an overview of where 61131-3 fits in the bigger picture of 61131 and other IEC standards. Then he explains the concepts behind 61131-3; strong data typing, reusable program organizational units (POUs), such as functions and function blocks, hierarchical design, either top-down or bottom-up, flexibility for a range of equipment scales and architectures, and the possibility of transporting applications in whole or in part between systems.

Chapter 3 is on the common elements of programming, and it is the core of the book. This section describes the data various types; generic, standard, and derived, the variables types; input, output, and external, and the POUs; functions, function blocks, and programs that are the backbone of control system programming. There are excellent tips on programming style hidden in this chapter. Each of the five programming languages, structured text, function block diagram, ladder diagram, instruction list, and sequential function chart, is detailed in its own chapter.

The first language is structured text (ST), a generic text language similar to PASCAL. Lewis provides a tour of the key aspects of ST beginning with the basics like assignments, expressions, and calls. The more complicated structures of conditional statements (IF...THEN, IF... ELSE, IF...ELSIF...THEN, CASE) and iteration statements (FOR...DO, WHILE...DO, REPEAT...UNTIL) are also discussed, providing good advice on programming style. The advantages of ST are the strong data typing, the capability of complex equations, and iterative calculations. ST is a very flexible language, but flexibility can be also be a disadvantage when the documentation is poor.

Programming in the graphical Function Block Diagram (FBD) language is familiar to many DCS users, but may be new to PLC users. Function blocks process data via algorithms just like functions, but function blocks can retain internal values. A function will always return the same output given the same input (think COSine), but a function block may return a different value given the same input, depending on the internal values (think PID). This graphical language provides excellent documentation of an automation program, as well as consistency and efficiency. Lewis promotes function blocks as a key product of the standard. He explains the general rules for programming in FBD. Later in the book he illustrates how to create function blocks, which can be written in any of the five languages, and the small set of function blocks defined in the 61131-3 standard.

The most traditional of PLC programming languages is the graphical depiction of relay ladder logic called ladder diagram (LD). This section is an excellent overview of generic LD functionality. The section starts with a good explanation of each of the symbols used in the generic version of the LD language. Advantages of LD are the intuitive interpretation of LD by people familiar with electrical circuit diagrams and the excellent documentation. Calculations are often difficult in LD.

Instruction list (IL) is the most basic of the five control languages. This language is a generic form of the pneumonic assembly programming that some systems allow. Programming in IL requires the understanding of the accumulator or results register. Most of the IL commands move data into the accumulator or process the data in the register. IL is a good choice for optimizing code used for straightforward critical operations.

Sequential function charts (SFC) is the highest level programming languages in the standard. The IEC standard for SFC functionality is based on the French standard for Grafcet. Lewis explains the basic rules for programming SFCs. The basics are transitions, input conditions that must be met to proceed to the next action, and actions, output events caused when a transition is satisfied. The rules of SFCs allow for diverging and converging sequences and sequences that run in parallel. The graphical depiction of sequential operations makes SFC the preferred language for batch automation. The actions may be written in any of the five languages.

The final chapters include an example problem using the languages, a discussion on the features expected in the programming tool and a brief explanation of the communication between systems that is covered in IEC 61131-5. The appendixes have all of the amendments made since the IEC 61131-3 standard was first published in 1993.

This book is a Rosetta stone for automation professionals, defining and connecting the languages of control systems. The concepts of IEC 61131-3 and the tour through each of the five programming languages arm the reader with enough knowledge to decide which languages are best suited for certain problems, and how to translate between languages where possible. While further amendments may be made to the standard, this book is destined to be a classic reference. At $77 (from Amazon as the Institution of Electrical Engineers wanted pounds) I rate this book a solid buy.