PLC Programming

Using RSLogix 500 &

Real World Applications

Learn Ladder Logic Concepts Step by Step

with Real Industrial Applications

By

A. B. Lawal

PLC Programming Using RSLogix 500 & Real World Applications

Copyright © AB Prominent Publisher

ISBN: 9788835349488

Published in the United States

PLC is a trademark of the Allen-Bradley Company. The following are trademarks of Rockwell Automation Inc. : Allen-Bradley®, MicroLogix™, PanelView™, RSLinx®, RSLogix™, RSLogix™ 500, SLC™ 500.

The author or the publisher of this book is in no way affiliated with Rockwell Automation, Inc.

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews. Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. The author/publisher, its dealers and distributors will not be held liable for any damages caused or alleged to be caused directly or indirectly by this book. The author/publisher has endeavored to provide trademark information about all of the companies and products mentioned in this book. However, he cannot guarantee the accuracy of this information.

Table of Contents

How this Book can Help You

This book covers the fundamental knowledge you need if you want to start writing your very first PLC program. It also covers some advanced knowledge of PLCs you need to become an expert in programming PLCs. So, after reading this book, you should have a clear understanding of the structure of ladder logic programming and be able to apply it to real world industrial applications.

The best way to master PLC programming is to use real world situations. The real-world scenarios and industrial applications developed in this book will help you to learn better and faster many of the functions and features of the RSLogix 500 platform.

The methods presented in this book are those that are usually employed in the real world of industrial automation, and they may be all that you will ever need to learn. The information in this book is very valuable, not only to those who are just starting out, but also to any other skillful PLC programmer, no matter their skill level.

Merely having a PLC user manual or referring to the help contents is far from sufficient in becoming a skillful PLC programmer. Therefore this book is extremely useful for building PLC programming skills. First, it will give you a big head start if you have never programmed a PLC before. Then it will teach you more advanced techniques you need to learn, design and build anything from simple to complex programs on the RSLogix 500 platform.

One of the questions I get quite often is, where can I get a free download of RSLogix 500 to practice? I provide later in this book links to a **free version of RSLogix 500** and a **free version of the RSLogix Emulate 500**. So you don't even need to buy a PLC to learn, run and test your ladder logic programs.

I do not only show you how to **get these important Rockwell Automation software for free and without hassle**, I also show with crystal-clear screenshots how to install, configure, navigate and use them to write ladder logic programs.

1. Introduction to PLCs

Almost all the industrial equipment you find in modern manufacturing facilities share one thing in common: **control**. A control system is a means of controlling a process by monitoring the inputs and, depending on the value, setting or manipulating the outputs.

Control systems have been in existence for many years. The early controllers consisted of hardware with physical links such as wiring or pipework but no solid state electronics are involved at all.

The most commonly used controller is the **PLC** (**P**rogrammable **L**ogic **C**ontroller), using a programming language called **Ladder Logic**. See a few different types of PLCs in Fig. 1.1.

Fig. 1.1. Different types of PLC's. Machine Design.

Ladder logic language was developed to make programming easy for anyone who already understood how switches, relay contacts and coils work. Its format is similar to the electrical style of drawing known as the "ladder diagram".

Originally, there were only a few functions available in this language, but as times have progressed, many higher-level functions have been introduced. We are going to look at the basic commonly used functions first, and follow with the advanced ones. There are various types of PLCs and differences among PLCs, but what is discussed here should be common to all types. Let's get started now, shall we?

1.1 Difference between PLC & Microprocessor

A PLC is a miniature computer that is designed specifically for industrial applications whereas a microprocessor is the central processing unit (CPU) of a computer. To be more specific, a PLC is often used for the automation of industrial electromechanical processes, for example, for controlling machinery on factory assembly lines, food processing, amusement rides, power plants, oilfields and even vehicles.

A microprocessor is just one component of an electronic device and it requires additional circuits, memory and software or firmware before it can be useful. A PLC on the other hand is a complete computer with its own microprocessor.

You can program or reprogram a PLC to control different types of devices like motors and lights, by using relatively simple programming languages such as Ladder Logic, which resembles an electrical circuit diagram.

Some popular manufacturers of PLCs are Allen Bradley (**A-B**), Siemens, Mitsubishi, Honeywell, Motorola, Hitachi, General Electric, Modicon, Schneider Electric, and Panasonic.

PLCs are designed for multiple arrangements of analogue and digital inputs and outputs with extended temperature ranges, good immunity to electrical noise, and a resistance to impact and vibration. Basically a PLC will consist of two sections: the central processing unit (CPU) and the input/output (**I/O**) interface system. See Fig. 1.1.1.

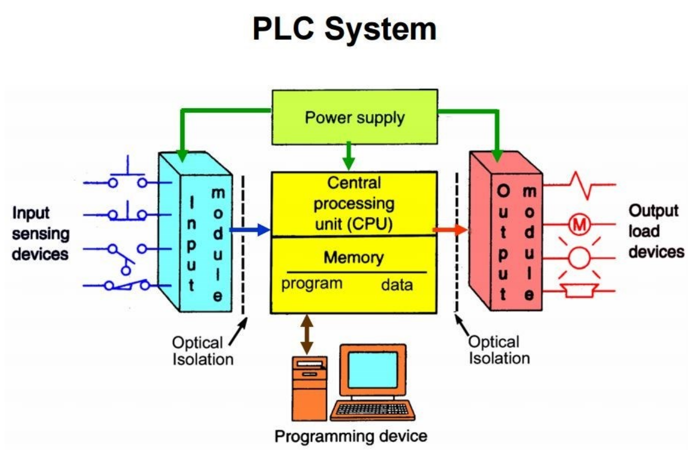

Fig. 1.1.1. Components of a PLC.

1.2 The 6 Main Components of PLC

PLCs are available in a wide range of sizes, but all of them contain 6 basic components:

1. processor, that is, the central processing unit or CPU;
2. mounting or rack;
3. input assembly;
4. output assembly;
5. power supply;

6. programming unit or device (a PC or software)

The Rack or Mounting Assembly

Many medium to large size PLCs are assembled in modules, that is, the individual components such as CPU, input/output, and power supply are mounted together within a rack.

However, in smaller PLCs, all of the above mentioned components are usually contained in a single “brick” or housing. Therefore such smaller PLCs are sometimes called “shoebox” PLCs or simply “bricks”.

The Power Supply

The power supply makes power available to the PLC system in the form of an internal DC current to operate the logic circuitry of the processor and the input/output assemblies. The most common power level used is either 24V DC or 120 VAC.

The Processor/CPU

The processor or CPU is the “brain” of a PLC. The type and size of the CPU determines the number of programming functions available, the size of the application logic available, the amount of memory available, and the processing speed.

Input and Output Assemblies

The input assembly sends input signals to the controller. Components of the input assembly may be switches or switch panels, push buttons, keyboards, pressure sensors, intelligent devices (such as robots) and so on. These can be perceived as the senses or sensors of a PLC.

The output assembly carries output signals from the controller to the outside world. It is made up of devices that the PLC uses to send signals to the outside world. An example is an actuator a PLC can change to control or adjust a process such as lamps, motors, pumps, relays, solenoid-controlled valves, etc.

There are many **types of inputs and outputs** that we can connect to a PLC. We can divide them into two large groups, namely **analog and digital**. Digital inputs and outputs are those that operate in a discrete or binary form, for example, ON or OFF, YES or NO. Analog inputs and outputs consists of variables that change continuously over a certain range, for example, temperature, pressure, potentiometer, etc.

Programming Device

A PLC is often programmed by a specialty programmer using some software on a computer, such as RSLogix 500, that can be used to load and change the logic inside. Many modern PLCs are now generally programmed using Allen-Bradley’s software such as RSLogix 500 or RSLogix

5000 installed on a PC or laptop computer. Older systems were programmed with custom programming devices.

1.3 What is the IEC 61131-3 Standard?

IEC stands for **International Electrotechnical Commission**. It is an international standards organization where standards for all electronic, electrical and related technologies (collectively known as "electrotechnology") are prepared and published.

The IEC 61131-3 standard represents the third (3) of 10 part of the open international standard used for PLCs. The IEC first published the first edition (IEC 61131 standard) in December 1993, and the current edition (IEC 61131-3 standard) was published in February 2013.

The IEC 61131-3 standard describes the basic software architecture and programming languages used for controlling a PLC. It gives three graphical and two textual programming language standards as follows:

1. Ladder diagram (LD), graphical
2. Function block diagram (FBD), graphical
3. Sequential function chart (SFC), graphical
4. Structured text (ST), textual
5. Instruction list (IL), textual

1.4 Difference between PLCs and PACs

PAC is an acronym for **Programmable Automation Controller**. Many major automation vendors also have PAC line, in addition to PLC. In this section I want to add some clarification to what a PAC is by explaining how it differs from the classic PLC.

Many people I’ve met in the automation industry have difficulty in distinguishing between the capabilities of the Programmable Logic Controller (PLC), and the newer Programmable Automation Controller (PAC).

First I have to admit that on several occasions I’ve felt guilty myself referring to PACs generically as PLCs just because I want to avoid confusing my clients. On certain occasions I even had to interrupt my presentation because I wanted to describe the somewhat intangible difference between PAC and PLC.

A lot of the confusion that arises from the difference between the capabilities of PACs and PLCs is the direct result of the **evolution from PLCs to PACs being a gradual one**.

For example, the attributes that vendors often use to describe their PACs today, such as the ability to control large quantities of distributed Input/output, or the capability of controlling process and batching systems, were previously been achieved with some higher end PLCs like the Allen-Bradley’s PLC-5.