PLC Training for Beginners
**Contents**

PLC Training for Beginners ........................................................................................................... 1  
Part 1: PLC Basics ......................................................................................................................... 4  
  Part 1: PLC Basics ....................................................................................................................... 4  
  History of the PLC ....................................................................................................................... 5  
  PLC Architecture ......................................................................................................................... 6  
  Rockwell Automation PLCs ........................................................................................................ 7  
    ControlLogix ............................................................................................................................... 7  
    CompactLogix ........................................................................................................................... 8  
    MicroLogix ............................................................................................................................... 9  
    Micro800 .................................................................................................................................. 10  
    And the old ones PLC5, SLC ................................................................................................. 11  
final thoughts .................................................................................................................................. 12  
What’s Now ................................................................................................................................. 12  
Part 2: PLC Hardware Components ............................................................................................ 13  
  Part 2: PLC Hardware Components .......................................................................................... 13  
  PLC CPU ..................................................................................................................................... 15  
  PLC Power Supply ...................................................................................................................... 15  
  PLC Digital and Analog IO .......................................................................................................... 16  
  PLC Communication ................................................................................................................... 17  
  Choose the best suitable PLC hardware for your application ..................................................... 19  
  What’s Now ............................................................................................................................... 19  
Part 3: PLC Programming ............................................................................................................. 20  
  Part 3 - PLC Programming ......................................................................................................... 20  
    Studio 5000 Introduction .......................................................................................................... 20  
    Studio 5000 Installation and Activation .................................................................................... 21  
    Get Started with Studio 5000 ................................................................................................... 21  
      Open Studio 5000 and Create New Project .......................................................................... 21  
      Studio 5000 Main View orientation .................................................................................... 23  
      Project Hierarchy ................................................................................................................... 25  
      Controller tags vs Program tags .......................................................................................... 25  
      Ladder Diagram Basic Instructions ..................................................................................... 26  
      IO Configuration .................................................................................................................... 27
PLCynergy

What’s Now ................................................................................................................................. 28
Part 4: PLC practical training .................................................................................................. 29
Part 4 - PLC practical training ............................................................................................... 29
Let’s develop a basic program .................................................................................................. 29
SOO - Sequence of Operation .................................................................................................. 29
P&ID - Piping and instrumentation diagram .......................................................................... 30
Code ........................................................................................................................................ 30
Connect your laptop to the PLC (Or Emulator) ..................................................................... 32
Configure the controller in Studio 5000 ................................................................................ 32
RSLink Classic .......................................................................................................................... 33
Download/Upload or edit Online ............................................................................................. 34
Part 5 Bonus: PLC practical training ...................................................................................... 36
Upload ...................................................................................................................................... 36
Edit Online ................................................................................................................................. 36
Part 1: PLC Basics

Welcome to our PLC training for beginners! It is the perfect place to start your journey into the industrial automation world.

Though this is general PLC training for beginners, we will be focused on Rockwell automation PLCs and software in the examples.

Part 1: PLC Basics

- **History of the PLC**
- **PLC Architecture**
- **Rockwell automation PLCs** – ControlLogix, CompactLogix, MicroLogix, Micro 800.

The programmable logic controller, as shown in figure 1, otherwise known as the PLC, is a compact industrial computer designed to control system processes from one location.

Other definitions define it as a special purpose computer with no display, keyboard, printer, or hard drive. It usually hides in the control panel on the factory floor or as a digital computer used to control machinery by constantly monitoring inputs and output devices.

Thus, whenever there is a need to control devices either locally, distributed, or remotely located, the PLC is the go-to solution for a flexible way to tie every component in the control system together.
The PLC is a computer means it shares similar features and capabilities with typical everyday computers, such as a central processing unit, input units, output units, software, memory units, and communication abilities.

The PLC is mainly divided into three core areas

- The central processing unit
- The power pack and rack
- The Input/output (IO) unit

**History of the PLC**

Prior to the emergence of the PLC, industrial control was mainly done in hardwired mode using relays and timers, where large banks of hardwired relays and timers, which took up large amounts of space, were used to control a single machine.

The relay needed to be wired according to a defined logic for the machine to function consistently.

When one amongst the swaths of relays malfunctions, as they were always prone to do, the entire system would be shut down, and troubleshooting the problem could take hours or even days to diagnose and correct.
Unfortunately, this posed a great challenge, as the hardwired system wasn’t very flexible with respect to the need for change and thus slowed down optimization endeavors.

The emergence of the PLC in the ’60s, specifically in 1968, revolutionized the industrial sphere, as it meant that a single device could replace the banks of hardwired relays and timers that the industry had gotten accustomed to.

The evolution of the PLC, starting from the days of the standard machine controller introduced by MODICON in 1968 to the 084 model in 1969, the 1774 PLC family introduced by Allen Bradley in 1970, the MODICON 184 model in 1971, the emergence of the Profibus/Ethernet in the ’90s.

There is evidence that the technology has grown in leaps and bounds and still has open possibilities for further growth, thanks to advancements in technologies such as artificial intelligence, analytics, machine health monitoring systems, and optimization.

PLC’s are now offered in different sizes, dependent on the application of usage. It is not unheard of to see a pocket-size PLC, which could be used for limited control purposes, while larger ones are most applicable for larger and more complex control applications.

**PLC Architecture**

Figure 3 depicts the overall architecture for a typical PLC connection with direct connections to the actuators, communication modules, Input and output modules, communication channels such as the Profibus, analog sensors, etc.

Different protocols guide communication within a PLC network: **Profibus**, ModbusRTU, Devicenet, **EthernetIP**, Hart, **ModbusTCP**, and Profinet.
The main components of a PLC hardware consist of a programming device, input and output modules, power supply, and the central processing unit.

Its software consists of the diverse distinct manufacturer software development platforms used for programming the PLC, using different languages such as structured text, ladder logic, and function blocks.

In part 2, we will focus on every component in detail.

**Rockwell Automation PLCs**

There are dozens of different PLCs vendors, but as mentioned at the beginning of the training, we will focus on Rockwell automation PLC.

The reason for picking Rockwell automation PLC is that it’s the most common brand in the USA and a leading vendor worldwide.

Although we in PLCynergy are familiar with all the leading PLC vendors, we are experts in Rockwell automation PLCs and software.

**ControlLogix**

ControlLogix is Rockwell’s flagship controllers’ family, also recognized with the catalog number 1756. It’s not a specific PLC rather a family of related hardware built around the L6X-L8X CPUs.
ControlLogix Controllers

The ControlLogix family includes CPUs, Power Supply, Rack, IO Modules, Communication Modules, and all hardware required to implement a redundant system if needed.

The ControlLogix is a high-performance PLC series with a modular and rack-based architecture allowing flexibility and scalability. Therefore, more appropriate for large-scale applications.

The PLC programming software of the ControlLogix is Studio5000 or RSLogix5000 (For firmware less than V20). Those software’s are practically the same. The Studio5000 is just a new branding for the RSLogix5000.

To summarize, It’s the PLC with the best capabilities you will ever find in the market, and its price is accordingly.

CompactLogix

The CompactLogix family is an excellent PLC series, also recognized with catalog number 1769 or 5069.
CompactLogix Controllers

The CompactLogix family is also modular but with few models that are integrated with a power supply, IOs, and communication in one box. Therefore, it is more suitable for small to medium applications.

Studio5000 or RSLogix5000 are the programming software also for CompactLogix. The support of two different PLC families is very beneficial to the organization that holds ControlLogix and CompactLogix on the plant floor.

To summarize, CompactLogix is a perfect PLC for a single machine or production line, and it’s much more affordable than ControlLogix.

MicroLogix

MicroLogix is a pretty old PLCs family. Part of the family doesn’t manufacture anymore. The only two PLCs still active is the MicroLogix 1400,1100.
Figure 6: MicroLogix Controllers

The MicroLogix is an excellent choice for small applications since it has an integrated power supply, built-in IO with an option to extend with additional IO modules, and communication ports.

It’s worth mentioning that the MicroLogix 1400 and 1100 have an Ethernet port that supports both EthernetIP and ModbusTCP protocols. And also, RS485 serial ports for older protocols like DF1, Modbus RTU/ASCII.

The prices of the MicroLogixs are low compared with the other Rockell PLCs and compared to competitors’ PLCs.

The programming software of this PLCs family is RSLogix500.

To summarize, MicroLogix is an old PLC series, but it is still active and manufactured today. We love using it in tiny projects with a tight budget or even as a communication gateway.

MicroLogixs PLCs had their time, and the use of them reduced in the last few years. Rockwell seems to develop the next PLCs family (Micro800) to replace the MicroLogixs in the future.

**Micro800**

Micro800 is the heir of MicroLogix. It has almost the same features include support in different serial and ethernet protocols.
The Micro800 is more affordable than the MicroLogix, so there is no sense in choosing MicroLogix over Micro800.

The programming software for this PLCs family is CCW – Connected Components Workbench, and it’s free in its standard version. The Pro version is pretty cheap and supports online edits.

**And the old ones PLC5, SLC.**

Nowadays, it is hard to find a PLC5 in a production environment. Although it is a workhorse PLC, it respectfully represented the Rockwell automation brand for a few decades.

The ControlLogix replace the PLC5 family in the late ’90s.

SLC500 is the middle-range PLC family of the past, and the CompactLogix family has replaced it.
Programmable logic controllers (PLC) offer a flexible way to effect control action. The ability to handle complex applications makes a must-have for critical operations such as manufacturing, distributed control applications, remote control applications, and adaptive control applications.

Its strength comes from the fact it is programmable and thus allows for quick changes to program logic in record time, unlike with the previous hardwires relay and timer logic.

The PLC has come a long way indeed, as it revolutionized industrial control, and with the emergence of more advanced technologies and techniques, we can rest assured that the PLC is here to stay for good.

**What’s Now**

Congrats! You finished the first part of our PLC training. The next part of this training is Part 2: PLC Hardware Components. In this part, we will dive in and specify the hardware parts of a PLC.
Part 2: PLC Hardware Components

Welcome to the second part of our PLC training for beginners! In this part, we will specify the hardware parts of a PLC in detail.

Though this is general PLC training for beginners, it will be focused on Rockwell automation PLCs and software in the examples.

Part 2: PLC Hardware Components

- **Central Processing Unit- CPU**
- **Power Supply**
- **Digital And Analog IO**
- **Communication**
- **Choose suitable PLC hardware for your application**

To proceed to the practical training, you first have to learn to construct a basic PLC component list with all of its components.

In part 1, we briefly described the PLC architecture. Then, here, we will break up every component.
The scan starts from the field, a measurement performed by the digital and analog sensors.

Examples of analog sensors are temperature, level, flow, pressure, vibration.

Examples of digital sensors are pushbuttons, limit switches, motors feedback, valves feedback.

The Input modules on the rack receive the signals from the sensors as electric voltage or current value.

The CPU process the inputs signals and runs a code (or logic) with the inputs values and values stored on a memory.

The result of the code is reflected as digital and analog signals that send back to the field devices (motors activate, valve open) through the outputs modules.

The scan is cyclic and will begin again automatically.

There are a few more processes in addition to the primary flow we described:

- **Diagnostic** – every scan, the CPU performs a diagnostic check for all components.
- **Communication** – The CPU with the communication modules handles the communication to more devices like other PLCs, VFDs, etc.
- **SCADA** – A connection to a SCADA is also running in the background, Inputs, and setpoints from the user.
The power supply provides power to all components on the specific rack.

**PLC CPU**

The CPU of a PLC is like a typical CPU for a personal computer, which controls all the functions of the computer.

However, different CPUs differ with respect to their handling speeds and storage.

The memory unit stores the PLC control programs and other numerical data electronically.

The CPU operates on 5V, where the holdup time for a PLC refers to how long it can bear a loss of power, typically 10ms to 3seconds.

The CPU of a PLC might contain more than one processor, with all individual processors having their own memory to be able to function independently.

PLC’s CPUs have a key switch, which provides functions

- **RUN position** – implementation of ladder logic program and operation of output instruments.
- **PROG position** – for editing special properties and firmware updates. The logic is not running in this state, and outputs are not active (by default).
- **REM position** – remote RUN or remote PROG will be controlled from the software.

PLC CPUs can interact with existing SCADA networks, complex decision-making aptitudes, equipped with extended commands for quick operation, execution of complex tasks such as calculation, indication control, latching, and other complicated mathematical endeavors.

**PLC Power Supply**

The PLC power supply converts a line voltage, typically 120V AC or 220/240V AC, into a usable 24V DC, used to power the PLC and its components.
The power supply typically has three stages: the step-down transformer stage, the rectification stage, and the filtering stage.

On certain PLC types, such as shown in Fig 3, the power supply also comes as part of the rack.

Thus the power supply provides power to all other connected components in the rack through a BUS system in the rack.

However, this is not standard as some modules do not have a power system integrated and must be wired individually to components requiring a power supply.

The current rating for PLCs can range from 2 to 10A for smaller PLCs to up to 50A for more powerful and more significant programmable logic controllers.

The mains typically provide power to the PLCs, but there usually is a reserve backup battery as well, as this will provide power to the memory of the PLC in the event of a power outage.

![Figure 3: PLC Power Supply](image)

**PLC Digital and Analog IO**

The most common PLC input and output type are the discrete I/O, which typically means signals with only two logical states (1 or 0)/ ON or OFF. Light switches, pushbuttons, proximity switches are all sources of discrete I/O.

However, we also have the analog I/O, which refers to signals with a range of values greater than just 1 or 0.

I/O modules are either monolithic or modular (rack-based)

The rack-based input/ output module has the advantage of being easier to troubleshoot and replace as opposed to the monolithic, which also suffers from having limited I/O ports.

The I/O modules are always protected from the power supply by optocoupler circuits which require isolation.

Discrete output cards can have up to 4, 8, 16, or 32 channels connected to current sourcing or current sinking devices.
Different signal types can be interfaced to an analog IO module, such as voltages between 0-10V, currents between 0-20mA or 4-20mA, thermocouples, RTD’s, and strain gauges.

![Figure 4: PLC Digital and Analog IO](image.png)

**PLC Communication**

The communication rule in the PLC architecture is increasingly important.

Different protocols are used for various purposes. Rockwell’s PLCs mainly depend on the EthernetIP protocol for communication with other RA PLCs, RemoteIOs, VFDs, HMIs, etc.

Most communication modules support DLR – Device-level ring topology, which provides a single-fault tolerant ring network.

In the following figure, you can see an example of a typical EthernetIP network.
If a different protocol is needed, like ModbusTCP or IEC81650, vendors like Prosoft are partnered with Rockwell and a few more brands and manufacturing communication modules for different protocols.
Choose the best suitable PLC hardware for your application

In every project, the first step will be choosing the most suitable hardware for the task from the application size and budget point of view.

Now you are familiar with the component required for your project. Next, you can go to Rockwell’s Selection Guides to explore the properties of every element.

- ControlLogix Selection Guide
- CompactLogix Selection Guide

What’s Now

Congrats! You finished the second part of our PLC training. The next part of this training is Part 3: PLC Programming. In this part, we will explore the different programming languages and the studio5000 PLC programming software.
Part 3: PLC Programming

Welcome to the third part of our PLC training for beginners! In this part, we will do an introduction to Studio 5000 programming software to program our controller, and focus on Ladder Diagram.

Though this is general PLC training for beginners, it will be focused on Rockwell automation PLCs and software in the examples.

Part 3 - PLC Programming

- Studio 5000 Introduction
- Studio 5000 Installation and Activation
- Get Started with Studio 5000
- Ladder Diagram Basic Instructions
- IO Configuration

In parts 1-2, we were focused on the hardware side. In this part, we will do the software side. Studio 5000 is Rockwell’s leading PLC programming software, used to program ControlLogix and CompactLogix controllers.

Studio 5000 Introduction
Studio 5000 is the new name for RSlogix 5000 that existed from the early days when the ControlLogix was born in 1997. The CompactLogix was released in 2006 and was also supported by RSlogix 5000.

As described in part 2, we have a CPU, IO modules, and communication modules in the PLC system. The program running and controlling all this hardware is programmed by Studio 5000 and can be downloaded to the PLC and modified in the future.

The latest version of Studio 5000 is V32. In the last years, Rockwell updated the versions of Studio5000 rapidly, adding functionality and fixing bugs, but the main features of Studio 5000 don’t change between the versions.

**Studio 5000 Installation and Activation**

In this training, we will not explain the installation and activation of Studio 5000 in detail. We assume that you already have it ready for use.

**Get Started with Studio 5000**

Ok, Let’s start being more practical.

**Open Studio 5000 and Create New Project**

Open Studio 5000 from the start button:

The Studio 5000 will open with this quick start screen, press New Project:
Chose the PLC type, here you will find all the controllers supported by Studio 5000. Of course, it will be all ControlLogix and CompactLogix controllers.

Give the new project a name, in our case it will be “Test_Project”:

New, we will choose a few hardware-related properties according to the type of controller we chose. Here the ControlLogix L85E controller was chosen, and the properties are:

- Revision – The controller firmware has to match this version.
- Chassis – Chassis size 4,7,10,13.
- Slot – The slot of the CPU in the chassis.
For other properties, we will leave the default values.

**Studio 5000 Main View orientation**

Finally, you will see the main view of the Studio 5000. Don’t be overwhelmed, I will explain the necessary properties required for this training.
**Project name** – as the name you assigned earlier to the project. It can be changed later from the controller properties settings.

**Project Hierarchy** – Tasks -> Programs -> Routines

Controller Tags – Tags and IO database, Controller Tags is the global level of tags, the local level is Program Tags.

**IO Configuration** – Here, the type of controller, IO Modules, And Communication modules are configured. This is the connection to the actual hardware.

**Controller Status** – Online/Offline/Program. Program: A state where the logic isn’t running and outputs are disabled

**Controller Key Mode** –

- Run – Logic is scanned, outputs are energized
- Prog – Logic isn’t scanned, outputs are disabled
- REM – Remote – The software controls the mode

**Ladder Diagram Instructions** – Relay Instructions, Timers, Counters, etc. Drag and Drop into The Rung.

Rungs – Ladder Diagram Canvas, Drag and Drop From The Instructions Tab. Stored In The MainRoutine.
Communication Tab – From there, you can download and upload the program.

Project Hierarchy

Routine – the logic code is saved as a routine. In our case, only the “MainRoutine” exists.

Program – a program contains the routines and the Program tags (those are local tags, in this training, we will use only Controller tags that are global to the project). You can have multiple routines with different names in one program. In our case, the program “MainProgram” contains one routine named “MainRoutine”.

Task – a task contains one program or more. In our case, the task “MainTask” includes one program named “MainProgram”. there are three types of tasks:

- Continues – Task will run all the programs and routine cyclically.
- Periodic – Task will run all the programs and routines every period (in milliseconds). Most suggested since it enables control of the scan timing.
- Event – Task will run all the programs and routine when a condition is met like tag change, or EVENT Instruction is scanned in the logic.

Controller tags vs Program tags

Tags are the controller’s database, and it includes IO tags (created automatically when adding hardware in the IO configuration section) and memory tags (created manually by the user).

Controller Tags are global to all tasks, programs, and routines. This means that you can program code wherever you want and reference those tags.

Program tags, Unlike Controller Tags, the Program tags are local for the program. Therefore, you can only reference those tags from a code in the same program.

In the beginning, it’s better to create Controllers Tags only. It will help you avoid confusion.
Ladder Diagram Basic Instructions

If you want to read about different PLC programming languages visit our blog post – [Popular PLC Programming Languages](https://plcynergy.com).

Here will focus on the basic Ladder logic instructions:

Those are the most basic instructions:

- **New rung** – simply adding a new rung to the routine.
- **Branch** – add a branch when you need two conditions in parallel.
- **Branch level** – like a regular branch, it will add a new branch aligned with the previews branch.
- **XIC** – **Examine on** – a Boolean condition. If the bool tag in the XIC instruction is equal to ‘true’, then the action after (from right) the XIC will be executed.
- **XIO** – **Examine off** – a Boolean condition but the opposite of XIC. If the bool tag in the XIO instruction is equal to ‘false’, then the action after (from right) the XIO will be executed.
- **OTE** – **Output energize** – an output action, if the conditions before the OTE are ‘true’ then the bool tag in the OTE instruction will be equal to ‘true’, if the conditions before the OTE are ‘false’ then the bool tag in the OTE instruction will be equal to ‘false’.
- **OTL** – **Output latch** – an output action, if the conditions before the OTL are ‘true’ then the bool tag in the OTL instruction will be equal to ‘true’, it will keep the value of ‘true’ until an OTU instruction with the same tag will be executed.
- **OTU** – **Output unlash** – an output action. If the conditions before the OTU are ‘true’ then the bool tag in the OTL instruction will be equal to ‘false’. It will keep the value of ‘false’ until an OTL instruction with the same tag will be executed.
Look at the picture above:

**Rung 0** – two bool tags (Bool1, Bool2) are a parallel condition (XIC, XIO) for the bool tag Output1. Then, if Bool1 is equal to ‘true’ or Bool2 is equal to ‘false’, the Output1 tag will be equal to ‘true’. Otherwise, Output1 will be equal to ‘false’.

**Rung 1** – one bool tag (Bool3) is an XIC condition for the Output2 tag. If Bool3 is equal to ‘true’ then Output2 will be equal to ‘true’. Output2 will be equal to ‘true’ until an OTU instruction will be executed.

**Rung 2** – one bool tag (Bool3) is an XIO condition for the Output2 tag. If Bool3 is equal to ‘false’ then Output2 will be equal to ‘false’. Output2 will be equal to ‘false’ until an OTL instruction will be executed.

Note: here, the tag Bool3 will always be equal to the tag Output2.

**IO Configuration**

In the IO configuration section, we will configure the actual hardware on the PLC. It may include communication modules, Digital IOs, Analog IOs, and many more modules for different capabilities.

You can see an example of ControlLogix 7 slots chassis with L85E CPU, EN2T Communication modules, and 4 IO modules (DI, DO, AI, AO).

The IO configuration must be configured accurately, Otherwise, unexpected results may occur?
What’s Now

Congrats! You finished the third part of our PLC training. The next part of this training is Part 4: PLC practical training. In this part, we will take everything we learned in this training and implement a simple program, including downloading the code to a controller.
Part 4: PLC practical training

Welcome to the final part of our PLC training for beginners! In this part, we will exercise Studio 5000 programming software and download our Ladder code to the controller.

Though this is general PLC training for beginners, we will be focused on Rockwell automation PLCs and software in the examples.

Part 4 - PLC practical training

- Let's develop a basic program...
- Connect your laptop to the PLC (Or Emulator)
- Download/Upload or edit Online

Let’s take everything we’ve learned until now and implement a real-life application.

Let's develop a basic program...

SOO - Sequence of Operation

The SOO is the story of the process. It will specify the equipment involved with the process (sensors, motors, valves) and its rule in the story.
The story of our process: Level control of a water tank. Two Level switches will produce indication of the level in the tank, an on/off (digital) valve will be open and close according to the level switch.

Our SOO is as follow:

Equipment list:

<table>
<thead>
<tr>
<th>#</th>
<th>Equipment</th>
<th>Name</th>
<th>IO</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Water Tank</td>
<td>C101</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Level Switch</td>
<td>LS102</td>
<td>Input: Digital switch</td>
</tr>
<tr>
<td>3</td>
<td>Level Switch</td>
<td>LS103</td>
<td>Input: Digital switch</td>
</tr>
<tr>
<td>4</td>
<td>Valve</td>
<td>XV104</td>
<td>Output: Digital on/off</td>
</tr>
</tbody>
</table>

Operation list:

<table>
<thead>
<tr>
<th>#</th>
<th>Condition</th>
<th>Delay</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LS102 digital input is 'Off'</td>
<td>5 Sec</td>
<td>Open XV104</td>
</tr>
<tr>
<td>2</td>
<td>LS103 digital input is 'On'</td>
<td>10 Sec</td>
<td>Close XV104</td>
</tr>
<tr>
<td>3</td>
<td>LS103 digital input is 'On'</td>
<td>120 Sec</td>
<td>Alarm - High Level</td>
</tr>
<tr>
<td>4</td>
<td>LS102 digital input is 'Off'</td>
<td>180 Sec</td>
<td>Alarm - No Supply</td>
</tr>
</tbody>
</table>

P&ID - Piping and instrumentation diagram

Code

Take the time and try to write the logic for our SOO. My Code in hidden here:

Let’s tell the story:
Rung 0 – If level switch LS102 is off (level under 50%) for 5 seconds, then open valve XV104 to fill the tank.

Rung 1 – If level switch LS103 is on (level is above 80%) for 10 seconds, then stop filling the tank by closing XV104.

Rung 2 – If level switch LS103 is on (level is above 80%) for 120 seconds, then it means that the valve is closed for two minutes, and still the tank level isn’t going down. Turn on an alarm for “High Level In Tank C101”.

Rung 2 – If level switch LS102 is off (level is under 50%) for 180 seconds, then it means that the valve is open for three minutes and still the tank level isn’t rising. Turn on an alarm for “No Supply To Tank C101”.

Look at the following trends. Do they tell the same story?

The same SOO ladder code can be written in ten different ways, and a code can continuously be improved. So check my code and make sure you didn’t forget anything.
Connect your laptop to the PLC (Or Emulator)

Great! We have the code ready to download, but first we have to configure the communication to the controller.

Configure the controller in Studio 5000
As described in part 3, we have to configure the IO Configuration with the actual hardware. So now, we only need to configure the communication:

For **Emulator** – Only change the type of the controller to Emulator, and change the controller slot according to the emulator slot.

For **Real Controller** – Configure the IP address to the actual controller IP (if the controller has no IP configured yet, you should assign IP via the BootP tool).

**RSLinx Classic**

It’s time to configure the driver that allow us to communicate with the controller:

For **Emulator** – Choose the Configure Driver icon, and pick the “Virtual Backplan” Driver.
For **Real Controller** – Choose the Configure Driver icon and Choose the “Ethernet Devices” Driver. Then, assign the same IP address as the IP of the actual controller.

**Download/Upload or edit Online**

Until now we were programed in Offline state. After we configured our controller in RSLinx we only need to download the program.
Now the controller is in RUN mode and the logic are scanned:
Part 5 Bonus: PLC practical training

In this bonus part we will continue with Upload and online edit training.

Upload

Most of the upload process is similar to download. As previously, the IP in the communication tab should be equal to the actual controller IP.

![Upload and download interface](https://plcynergy.com)

Edit Online

Okey, we downloaded the code to the controller, but be aware that you will need to change the code in the future. When you download the code, the controller will be in Prog mode and in this state the code is not running, and outputs are disabled.

In most cases the production process will not allow to download and stop the process, for this reason you should use Online Edit.

When you online go to the ladder diagram in one of the routines and double click the rung you want edit. The rung will duplicate itself; the upper rung represents the edited code and there you will make the changes. The second duplication represents the online code running in the controller.
After you make the necessary code changes, you have two options to insert your changes to the running program.

**Option 1** – Finalize all edits in program

This will finalize all the edits in the code, even if they are in different routine, the be carful with this button. Remember the process is running and we usually want to do the edits in a controlled manner, this way option 2 is preferable.

**Option 2** – Accept pending program edits

It will accept the only edits of rungs that you marked; this way you can insert changes one by one.

After accepting edits, you should click the Test Accepted Program Edits. In this state the edited code replaces the online code, but you can still regret and go back.

And finally click the Assemble Accepted Program Edits to make the code permanent: