

How PLCs Work

Archived: 2026-04-05 14:55:11 UTC



How PLCs Work

Operations and connections

How PLCs work, or to use its full name, how Programmable Logic Controllers work is what we're about to discover.

Let's put the focus on the why and how PLCs tick.

The PLC is a specialist type of computer so the why and how PLC's work is needed to utilise them correctly and to your best advantage. They are used extensively but not solely in industry, to control machines and processes.

Being a computer, it shares common terms with the typical PC (Personal Computer) you'd have in your home. Terms like CPU (central processing unit), RAM & ROM (random access memory & read only memory), also software, hardware and communications (Comms).

That's about where similarities end though as unlike a PC, the PLC is designed to be used in harsh and rugged industrial conditions. Also the PLC needs to be and is, very flexible in how it can interface with inputs and outputs and other computers in the real world.

- BTW -

Also in the interest of how PLCs work, you may like to see the [PLC History](#) or the [Smallest PLC](#) pages on this

website.

Or maybe you would like to register for our self teach [PLC programming course](#) which is in the making. Just a thought.

- BTW -

HOW PLCs WORK.

The components that make a modular PLC work can be divided into four core areas.

- **The backplane or *rack* and power supply.**
- **The central processing unit (CPU).**
- **The input/output (I/O) section.**
- **The Program section.**

PLCs come in so many makes and different shapes and sizes. They can be so small as to fit in your pocket but also the more involved controls systems can be as large as a wardrobe with several interconnected large PLC racks.

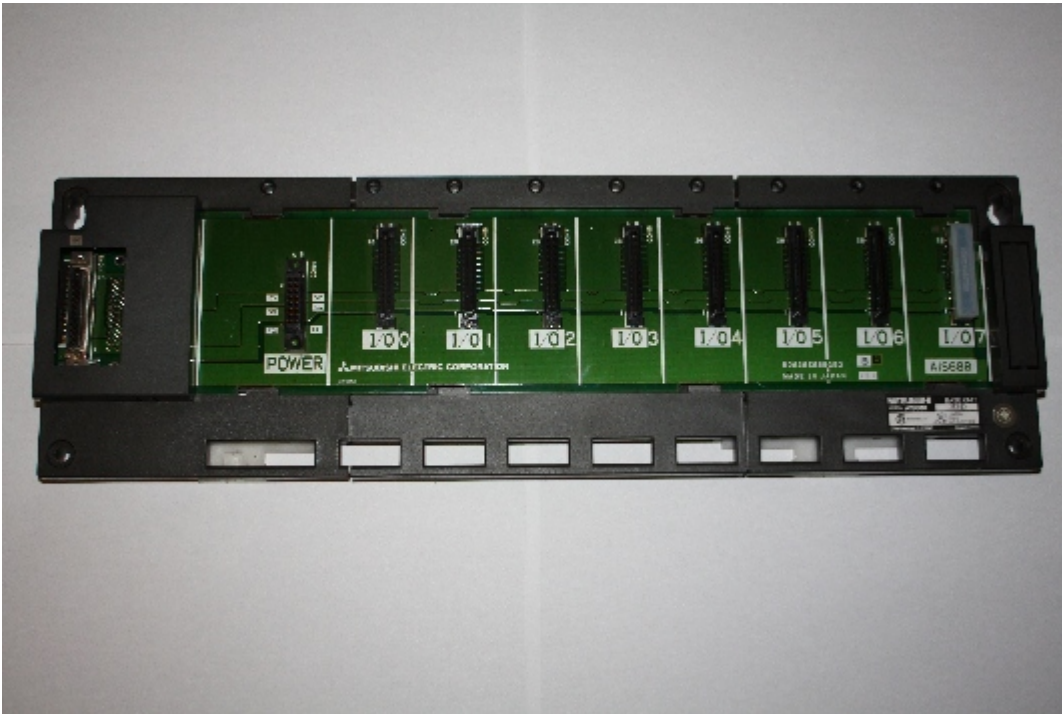
The smaller PLCs, also known as a *brick* type, are typically designed with fixed I/O points as low as 6 but also up to 256.

For our purposes of consideration here, we will be looking at the modular rack based systems as they have easily definable parts. They are called “modular” because the backplane rack can accept many different configurations and types of I/O modules that simply slide into the rack and plug in.

Consequently a modular PLC can be custom built with whatever configuration of I/O is best suited to the job in hand. Along with a multitude of configurations that also means they can made as small or big as needed too.

The backplane - Rack and Power Supply.

So lets start off with the rack, this part provides the physical assembly for the PLC I/O as well as the bus connections between all the parts. The *bus* refers to data-bus which is the Comms link between all the I/O under the PLCs CPU control.



The rack above is a Mitsubishi 'A' series rack with eight I/O slots and a power supply unit (PSU) slot. You may think it doesn't do much and you'd be right, well on its own anyway. But, the rack is an important part of how PLCs work.



Here it is again with the PSU slotted in place.

OK, I said just now the rack is an important part of how PLCs work. It provides all the slotted in I/O modules with power from the PSU.

More importantly, it also provides communications between the I/O modules installed and the PLC CPU. Now if you look closely at the rack above it does not have a slot for the CPU.

That's because the rack pictured is what's known as an expansion rack. It's provides an I/O expansion to the PLC CPU rack, below is how it would look in a typical installation.



Can you see the rack data cable connecting the expansion rack (on top) to the CPU rack (below)? It's the white cable snaking along the top of trunking. Well it started off white, this **is** a picture of a working panel.

If you look very closely you can see some of the input and output LED (Light Emitting Diode) lights are actually on! As you may now realise this is quite typical of how a PLC CPU rack and Expansion rack set-up would look in real working panel. The PLC in this panel is controlling an escalator for London's Underground no less.

Given the millions of people that use the London Underground Tube railway, gives an indication of just how PLCs work very reliably day in and day out. Otherwise they wouldn't be in use there I can tell you.

The rack is the PLC component that actually holds everything together. Depending on the I/O needs of the PLC control system being produced, it can be ordered in various different sizes to hold more modules. Also as you've seen here, it can be *daisy chained* together with additional racks called expansion racks.

Expansion racks offer two advantages over just having one huge long PLC rack for large I/O capable PLC systems.

One, while expanding the I/O they are separate to the CPU rack so can be stacked. This give the advantage of keeping the width to a manageable size.

Two, while able to be positioned at heights within the cabinet and dividing up the wiring, they give almost limitless expansion possibilities to the PLC.



Like the spine in the human body, the rack has a backplane at the rear which gives the physical support needed to the PLC modules/cards. As well as allowing all the modules/cards to communicate with the CPU via the back plane databus. This databus is a very important part of how PLCs work, allowing the CPU direct access to each individual module.

The power supply plugs into the rack as well (one for each rack) and supplies regulated DC power to all the modules that plug into that rack.

The most popular power supplies in use are either the universal AC supply type (pictured here) which work with anywhere from 100V AC to 240V AC or 24V DC sources.

How PLCs work with the inputs and outputs is generally using 24V DC to give inputs and switch output devices on.

That's why you can see in the PSU picture the 24V DC output as well as the 5V DC for the PLCs internal power and the universal AC input terminals.

The 24V DC supplied by a PLCs supply is generally quite low capacity and is only used to power the inputs for the system. A separate PSU would be used more the more demanding output supply as these could be quite an amount of Watts (Volts X Amps measurement).

The CPU - Central Processing Unit.

How PLCs work is down to the CPU, the CPU is the PLC part which is the brain of the whole PLC. This module typically lives in the slot beside the power supply or is incorporated with a power supply. PLC Manufacturers do offer different types of CPUs and CPU/PSU combinations based on the complexity needs for the system.

The CPU is made up of several components, a microprocessor, memory chip or chips, I/O interfacing and other integrated circuits (IC's) to control logic, monitoring and communications. CPU's vary in speed of operation, the price going up along with the speed of the CPU of course.

The CPU itself has a few different operating modes.

- **Programming Mode.**
- **Run Mode.**
- **Stop Mode.**
- **Reset Mode.**

In **Programming mode** it accepts the downloaded program logic from the laptop or PC you would use to write the controlling program. The CPU is then placed in run mode so that it can execute the program and operate the desired process.

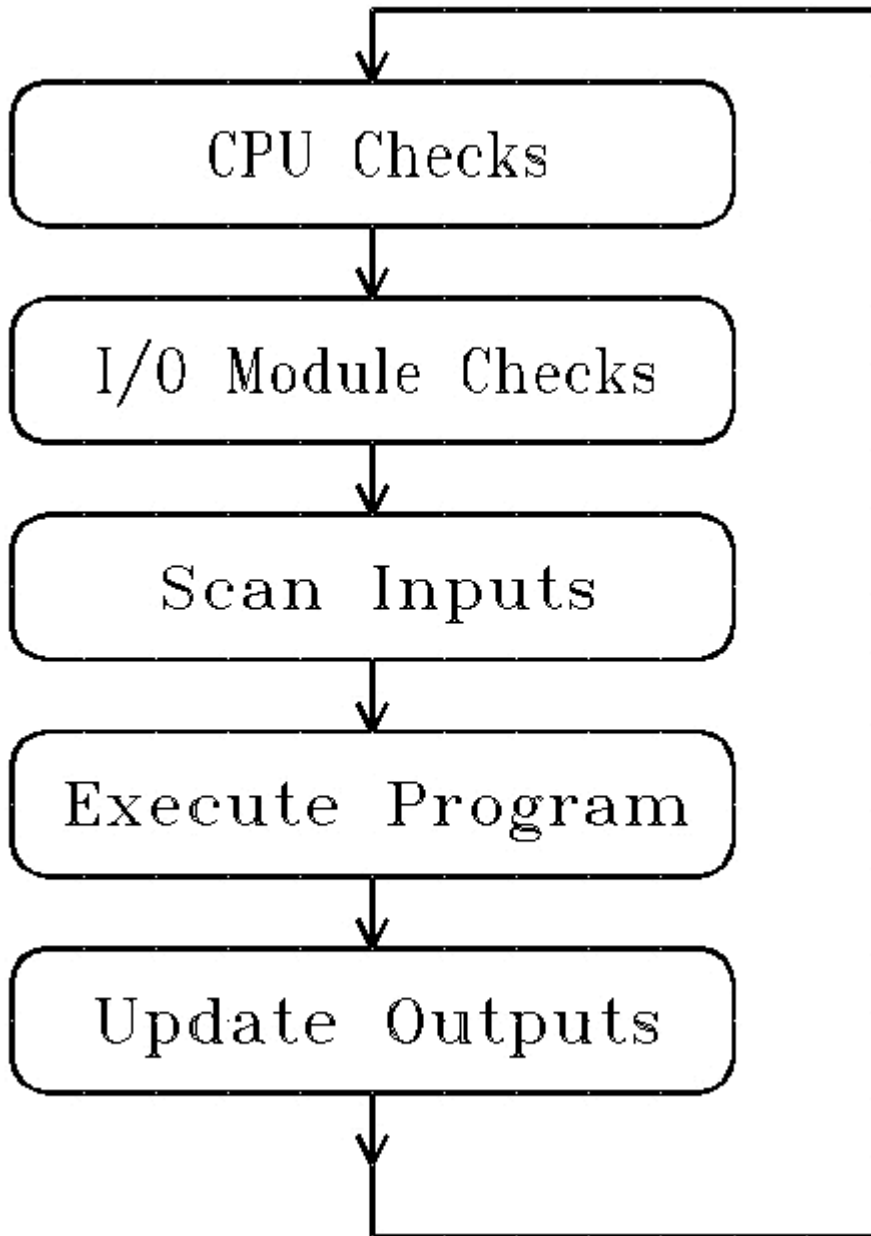
In **Run mode** the PLC is in full operation, doing all its self checks and operating governed by the program held in it. Reading the inputs and setting the outputs accordingly. Even *conversing* with other units via the RS232, Profibus, Scada or CC-Link.

In **Stop mode** it's as it says, stopped. Now you might not think this is a mode as such but since some programming can be done while running, while some PLC programming functions require it to be stopped. Putting a PLC into Stop mode also turns off all the outputs. Handy for checking inputs with causing anything to happen out of sequence.

In **Reset mode** is also as it says, it resets the PLC from operating conditions back to switch on position. When this is done without resetting any data memory registers, this is called a **warm reset**. If the reset performed is full, for example resetting all I/O and data registers, it's called a **cold reset**.

The Scan-time

Below is the basic flow chart of how PLCs work through and process the I/O, their program operation and error checks, know as the **Scan** process. The Scan-time is how long it takes the PLC to loop around the whole process back to the start, and can change depending on how much the Scan process has to do.



The scan-time speed is very quick, and has to be as they're dealing with real time situations. The scan-time of a PLC happens in the order of 1/1000th's of a second. Sometimes it can even vary on a scan by scan basis depending on program loops being switched in and out by inputs and/or program equations.

Most of how PLCs work through their operational processes is similar to any CPU, a PLC program scan would first perform self-checks. Such as error check its own memory and integrated circuits to verify the CPU and associated circuitry is actually operating correctly.

The next step is to error check any I/O modules fitted into the rack(s). This would include inputs, outputs, RS232 modules and in fact any and all modules fitted. Once it has a green light for all these will it then start on the control program sequence.

Before it executes the user program, the PLC will scan the input modules, after that's complete it will execute the user program. But since a PLC is a dedicated controller, it will only process this one program, it will go through

the whole program once per scan.

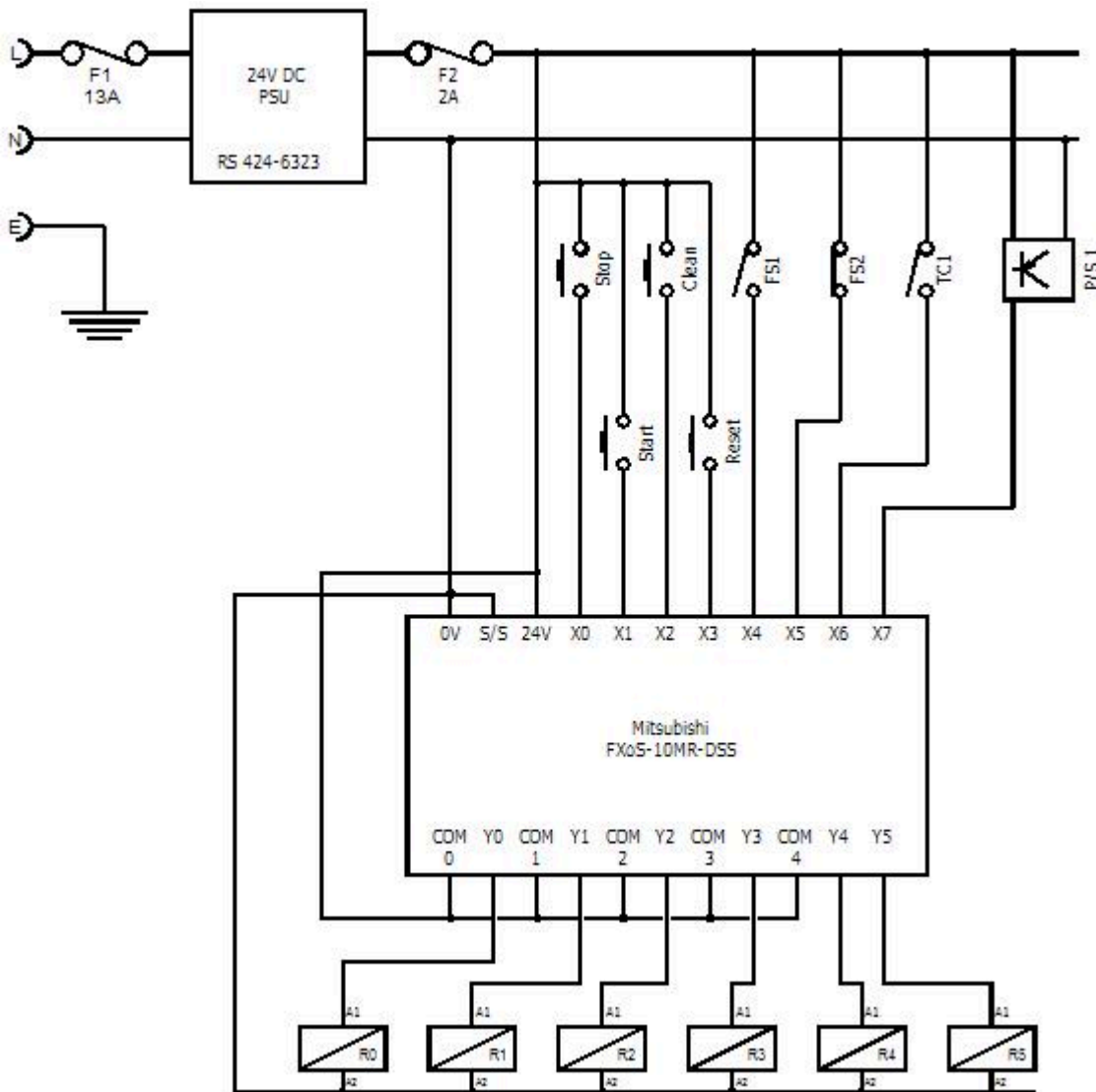
The memory in the CPU stores the program in non-volatile RAM, which means it won't lose the program if the power is lost. It uses volatile and non-volatile RAM for holding the status of the I/O and providing a means to store values. Some are kept at power down some are not.

Then the PLC will update the outputs according to the condition of the inputs and the program logic instructions. Then the PLC repeats this process over and over again.

The finer points of how PLCs work does vary with different manufacturers, but this basic process outlined here holds good for all of them.

The input/output (I/O) section.

The I/O connectors on the PLC system provides the physical connection between the equipment and the PLC. Opening the little plastic cover doors on an I/O card reveals a terminal strip which is where the devices connect to.



Above is a simple PLC wiring diagram (called a schematic), in this example you can see the PLC is a 24V DC supply type. To show how PLCs work the PLC is shown just as a box. Sometimes not even that, more often then not there's just a terminal with a reference to where it connects on the PLC I/O.

There are many different types and kinds of input and output cards which are picked depending on what type of control system is needed. How PLCs work can be customised by the type of input and output cards picked so the CPU can use them for it's logic control.

It's simply a matter of defining your PLC specification for which kind of inputs, outputs and Comms cards are needed. Filling the rack with the appropriately picked cards and then addressing them correctly from within the CPUs program.

This addressing is easily done within the PLC parameters area of the ladder program these days, this tell the PLC which cards are fitted in the rack and at what position. Having said that, it is still possible to address them all directly from the ladder program, as the earlier PLC versions had to.

The Inputs.

A PLC input device means anything that can give an input to the PLC, that can influence the programs operation. These can consist of digital, analogue, switches, sensors, intelligent devices and even Comms modules.



You would choose a digital input card to handle discrete devices such as push-buttons, micro-switches, selector switches, photocells and proximity sensors which give a signal that has only two states. They are either on or off.

This type of device is what's called a bit device. This is because the full scale of their signal range, (called Full Scale Deflection or FSD for short), can be represented by one bit. There are eight bits in a byte, computers *talk* in bytes or multiples of bytes. To help to understand more on how PLCs work see [32-Bit](#) and [64-Bit](#) computer databus.

PLC digital input cards that handle discrete devices are available with anywhere from 8 to 128 inputs on a single card that slot into the rack. However any more than 16 inputs on a card usually means having a breakout connector as it's just not possible to connect that many wires onto the top of an input card on the PLC with built in screw terminals.

A breakout connector is merely a means of *fanning* out the physical connection from the input or output card. This is achieved by using a multi-pole connector on top of the card which connects via a data cable to another card with all the screw terminals on it. It's this card that the discrete bit devices will connect to.

Analogue devices can also be input devices to PLCs, but due to how PLCs work, they need a special card to translate their infinitely variable signal (which could be voltage or current) into something the CPU can understand. Analogue signals need to be, what's called *digitised* before the CPU can deal with them.

This simply means converting say a 0V to 10V DC signal that you might get from load cell or speed control potentiometer into a value. For example if 10V DC equalled a value of 4000 then 2.5V DC would equal 1000 and 5V 2000. Also the FSD for this device would then be 4000 (just testing).

Examples of analogue devices that you may come across are those like the previously mentioned reference setting potentiometers and load cells, as well as pressure transducers, flow meters and thermocouples for temperature readings. There are many more of course, than just the ones I've mentioned here.

I should also mention Comms modules, as they could be used as input devices, they are capable of conveying either type of signal as they communicate in bits and bytes, or in other words - values. These values could be representing levels from previously digitised analogue devices, or lots of individual bit devices.

Lastly we'll give intelligent devices a mention too as, like Comms modules they too talk in values. Intelligent devices would include things like DC and AC drives for motors, other PLCs, HMI screens (Human Machine Interface), remote I/O stations as well as sophisticated sensors such as cameras and position arrays.

The Outputs.

Now, furthering how PLCs work it may not surprise you to know output devices can consist of digital (bit devices) and/or also analogue devices (value devices). Also the a fore mentioned Comms modules and intelligent devices are bi-directional devices and part of how PLCs work).



Bi-directional devices, by their definition talk in both directions, in other words they are simultaneously inputs and outputs at the same time. If you look at the [RS232 pinouts](#) you will see a Tx pin and an Rx pin, which stands for transmit and receive respectively. More basically, this translates into talk (Tx) and listen (Rx).

Perhaps this is a good place to say as a prelude to the outputs, one of the reasons how PLCs work so well is because they are able to convert signals in both directions, in and out. It makes sense really, I mean if you have a voltage or current reference input, sooner or later you're going to need a voltage or current output.

These input converting cards or modules are called [A/D Converters](#), the 'A/D' bit stands for **A**nalogue and **D**igital, complimenting the A/D converter input cards are the output converting cards called D/A Converters. As you can see from the name the process is reversed.

This pair of PLC modules give us the the ability to convert an analogue signal into a digitally represented value, process it within the PLC program and then convert it back to an analogue signal again.

An analogue output card will convert a digital value or number sent held in a memory location by the CPU into a real world voltage or current. Typical outputs signals can range from 0-10V DC, -10V to +10V DC or 4-20mA and are used to control servo drives and positioning controls as well as pressure regulators and level control systems. This type of system is called 'Closed Loop' control.

The digital output card is the compliment to the digital input card and turns a (bit) device on or off such as lights, LEDs, small motors, solenoids (electromagnets), and relays. Digital output cards are available with 8 up to 128 output per card, but again like the input cards any more than 16 would need a connector breakout card because of the physical space needed for the wire screw connections.

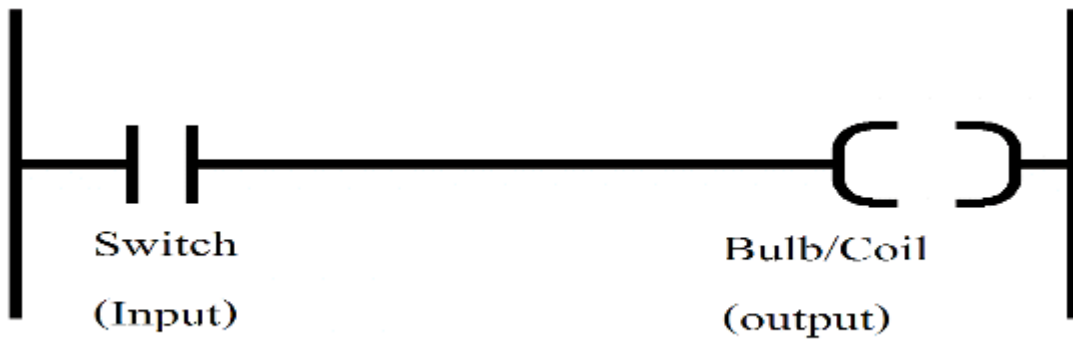
The Program.

Programming a PLC these days calls for a PC, specially dedicated software from the manufacturer of the PLC and more often than not, a special connecting or programming lead. This pretty much covers the minimum that must be obtained and used to program a PLC.

The most favoured and widely used form of programming language is called ladder logic. Ladder logic programming uses specific symbols instead of actual words. This came about as an emulation of the real world hard wired relay logic control in use long ago (and still). In a bid to make it easy for the technicians to see how PLCs work and be able to program the PLCs.

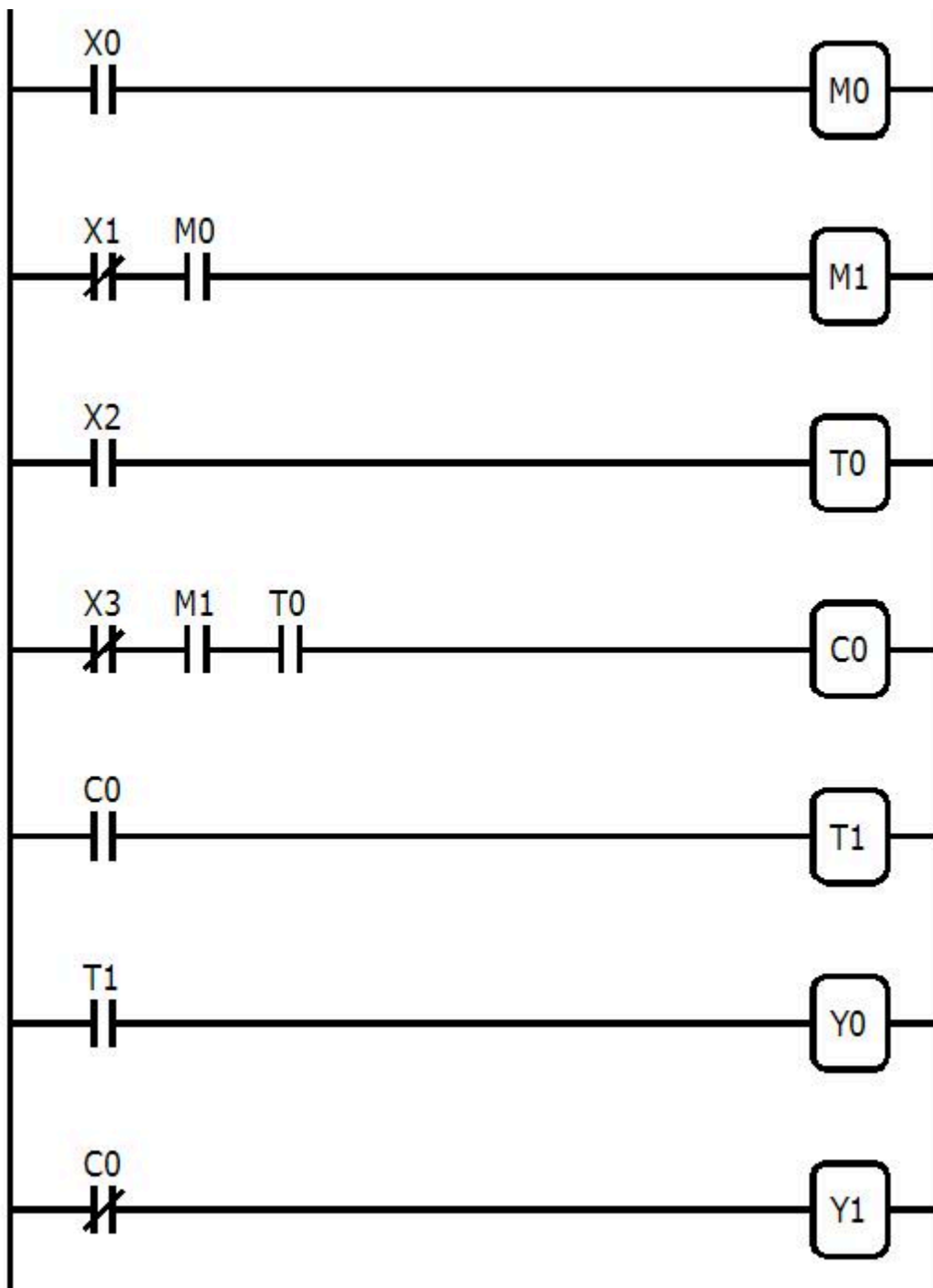
Although sometimes considered by some as a relic from the PLCs history. It does still mean it's relatively easy for anyone with an understanding of relay type control wiring to program PLCs with little guidance. Or at the very least, to understand how PLCs work and program them on the basic level anyway. As due to the advancements in PLCs over the years, there are now some very complex instructions that may be used.

The input symbols are representations of real world normally open and normally closed switches, the output symbols represents relays and lights connected by lines as though to show a wire. So the flow of current through the switch and relay or light, like relay contacts and coils can be viewed like a real circuit, like this.



This above is the simplest of programs that can be written for a PLC, that I might add, will actually work. A digital input, like a push-button connected to the first input position on the card. When the input is made (the gap bridged) it turns on the output, which supplies power to energize an indicator light outside the PLC.

The completed program pages look like a ladder, which **is** why it's called Ladder Logic. The left and right rails, the ladder uprights, indicate the positive and negative or ground of the power supply.



The rungs of the ladder are the *wiring* between the different symbols or operations, which are (within the PLC) all in the virtual world of the PLCs CPU. So if you can understand how basic electrical circuits work (as do technicians and engineers) then you can understand ladder logic and therefore how PLCs work.

The completed program (which would be much longer than this one in practice) is downloaded from the PC to the PLC using the special programming or connection cable. It connects between a serial port on the PC to the programming connector on the front of the PLC CPU and is usually [RS232 wiring](#). When *downloaded* the CPU is then put into run mode so that it can start scanning the inputs, program logic and controlling the outputs.

The PLC Program Symbols.

The program symbols used within the PLC determine how PLCs work and what functions the PLC carries out and

when. A different PLC program symbol used can completely change the operation of identical *rungs* of a program ladder.

In this following paragraph I will be referring specifically to Mitsubishi PLCs as this is the make I am most familiar with. But, having said that the comments here will transpose to other makes of PLC quite happily.

As a very quick example to start with, take the ladder rung at the very top of this page. If the output symbol [Y0] was change to an internal relay symbol [M0], this program rung would then be **no** longer controlling a relay or valve in the real world.

It would only be switching an internal relay (memory bit) within the PLC. This rung would only then be able to affect a change in the real world if this M0 bit was subsequently used in a rung using or affecting a [Y0] to [Yn] output. Incidentally, Yn is an expression used to indicate any **number** of available outputs.

* more to follow *

That's about it for how PLCs work, at the more basic end of course, no point bombarding you with too much at once.

I hope our 'How PLCs Work' page has helped you in your quest for knowledge.

Please, choose one of the links below for more related information.

[Go from How PLCs Work to our PLC Tutorial page.](#)

[Have s look-see about the Smallest PLC page.](#)

[Return from How PLCs Work to the PLC page.](#)

[Return from here to the Home page.](#)

PLEASE feel free to use this button and donate [You Are Secure!](#)

Source: http://www.machine-information-systems.com/How_PLCs_Work.html