Intro to Programmable Logic Controller

Created by Tom Wylie, 5/12/17

Introduction to Programmable Logic Controllers

After viewing this document, the student should be able to:

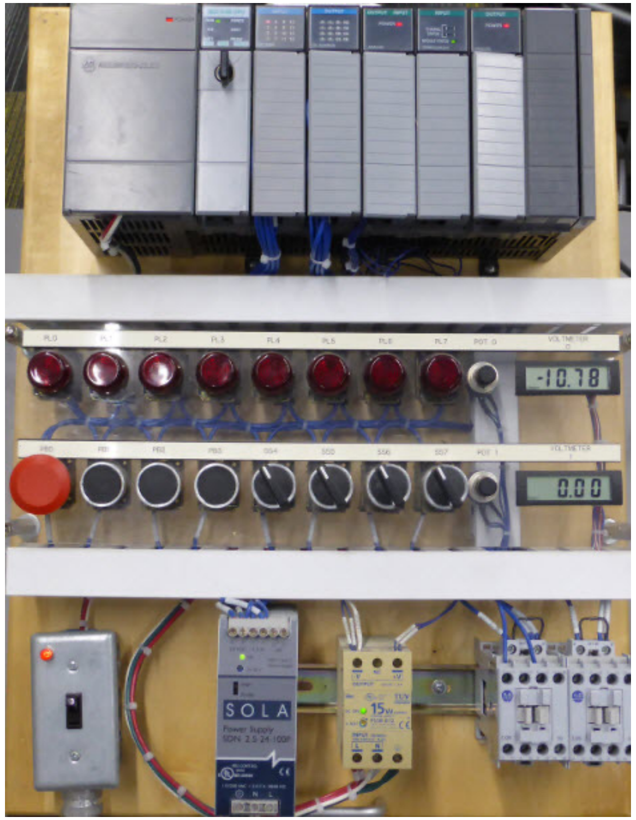
1. Identify what PLCs are used in the PLC course.
2. Explain what software is needed to make a computer, a PLC program panel
3. Identify the 4 sections of a PLC
4. Explain the difference between digital and analog I/O.
5. Explain the difference between a fixed I/O and modular I/O PLC system
6. Explain how signals flow from input modules to input image tables, and output image table to output modules
7. Explain the purpose of the PLC program (ladder logic), and where it is stored.
8. Explain the scanning sequence of a PLC processor
9. Explain the purpose of an electrical print and how to find the output address of an output

Please Read This

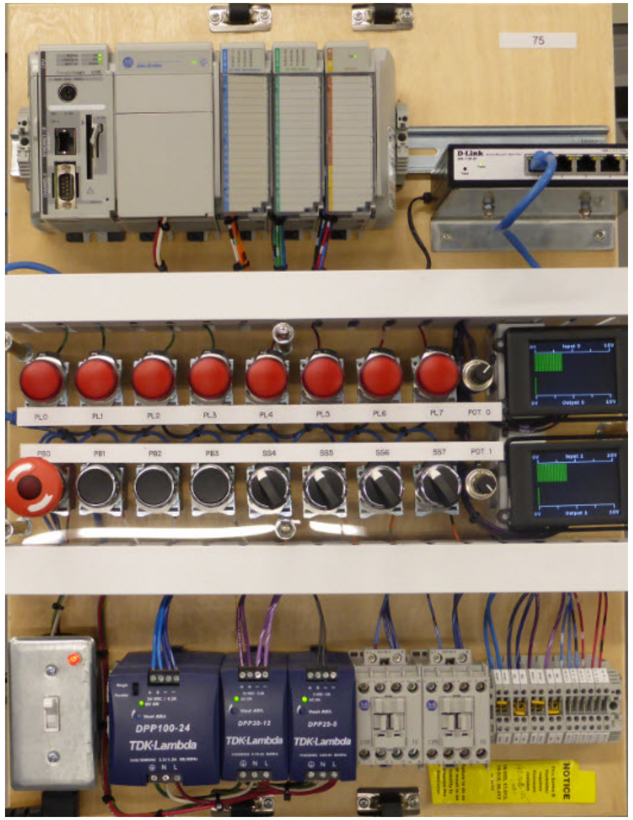
This document is to overview the operation of a PLC system, based on the hardware the student will use in the PLC lab.

The intent of the document is the student to get a good understanding of the hardware used in the course, as well as the software that will be required to program and troubleshoot the PLC.

PLC Training Units used in the PLC200 course:



Allen Bradley SLC-500 Training Unit



Allen Bradley CompactLogix Training Unit

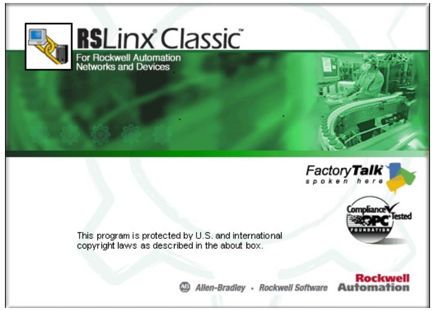
In the Industrial Electricity II course, you should have been introduced to the Allen Bradley MicroLogix 1200 PLC. That unit was a fixed I/O type of PLC.

In the PLC200 course, the course will focus on the Allen Bradley SLC-500 modular I/O PLC, and the popular CompactLogix PAC. The SLC-500 is at least 20 years old, but with the large installed base, the employers still request that we include these PLCs in our coursework. These are fairly simple units to understand and learn our PLC programming and troubleshooting on. 25% of the course is on the CompactLogix. The CompactLogix is a baby version of the ControlLogix. Each of these controllers are considered PAC (Programmable Automation Controllers). This is a somewhat new term, but means there is enhanced data communications (networking), multiple programming languages, and a more deterministic method of communication between the processor and the I/O modules.

Computer needed for PLC communications

Desktop monitor, Keyboard and mouse

Computer tower sitting on floor



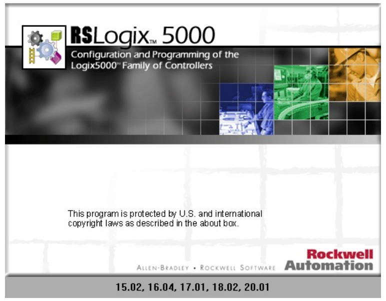
RSLinx Classic

For Rockwell Automation Networks and Devices

This program is protected by U.S. and international copyright laws as described in the about box.

FactoryTalk spoken here
Compliance Tested OPC Foundation
Allen-Bradley · Rockwell Software
Rockwell Automation

RSLogix 500



RSLogix 5000

Configuration and Programming of the Logix5000 Family of Controllers

This program is protected by U.S. and international copyright laws as described in the about box.

15.02, 16.04, 17.01, 18.02, 20.01

An important peripheral to the PLC system will be the programming panel. This is basically a Windows based computer with PLC software loaded on it that will be used for programming, communications and troubleshooting. The three software packages that will be used in PLC200 is as follows:

RSLinx – The communications software. This software is required anytime the user wishes to perform a function with the PLC. RSLinx ties the PLC communications (through RSLinx drivers) to the Windows environment, so any Windows applications can communicate with the PLC.

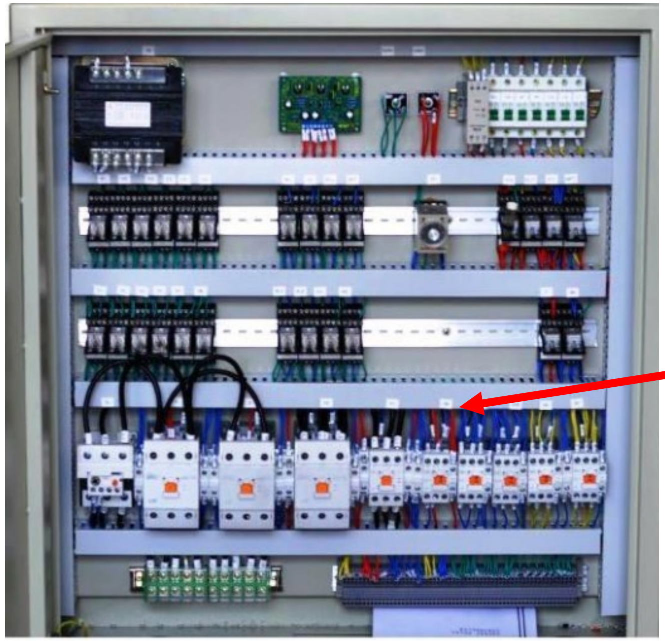
RSLogix500 – The programming and troubleshooting software for the SLC-500. This software typically retails for about $2,000.

RSLogix5000 – The programming and troubleshooting software for the Compact/ControlLogix processor. This software typically retails for about $2,000.

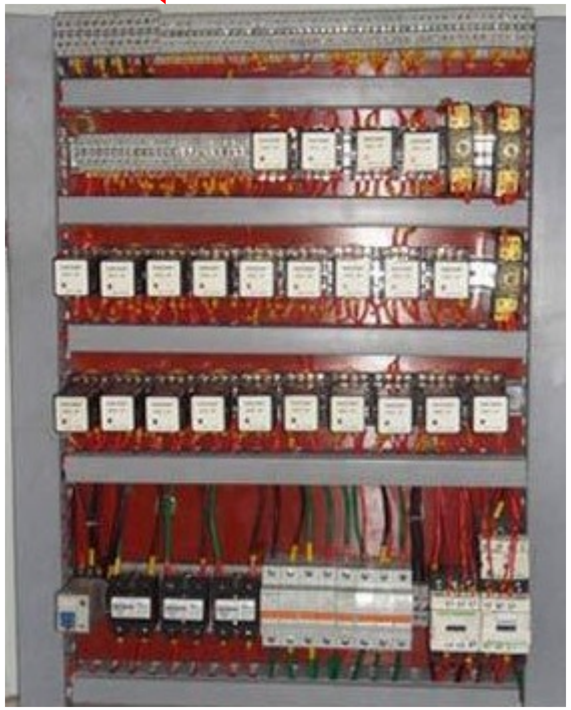
All of these software packages will be found on the computer in the PLC lab. Students will also have full access to this software from home, using a standard Windows computer to connect to their Virtual Machine.

The top two graphics shows the desktop/tower computers found in the PLC lab.

Just a few things about PLCs



Relay Panels – relays and timers are wired together to create the logic to run a machine.

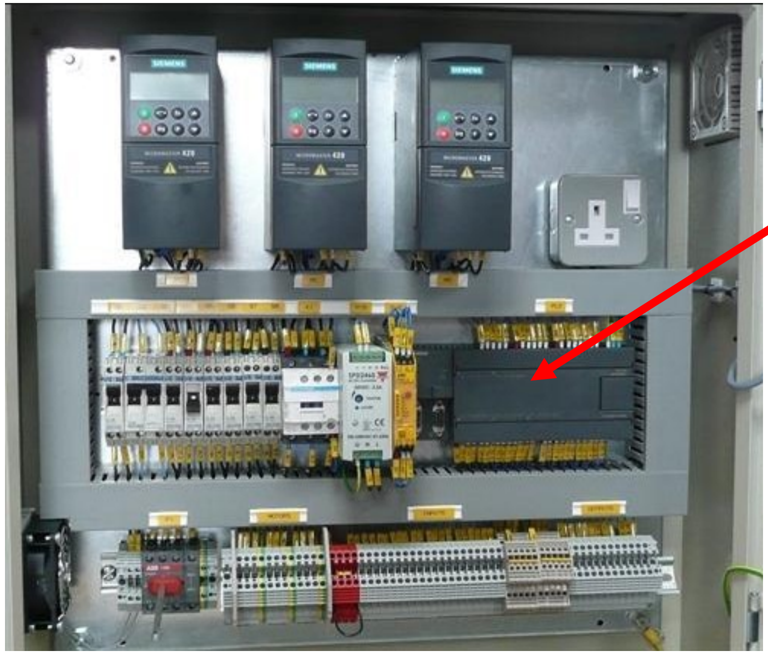


PLCs (Programmable Logic Controllers) are industrial hardened computers that control machinery in a manufacturing plant, but are also used in many commercial installations.

Originally PLCs were many times called "Relay Replacers". Before PLCs, control panels used electro-mechanical relays and timers to create the logic to run a machine. The problem is not only the maintenance of these devices that eventually wear out, but also, to change the logic to run a different part on a machine, the maintenance personnel would have to rewire the panel, to get a new logical sequence or operation.

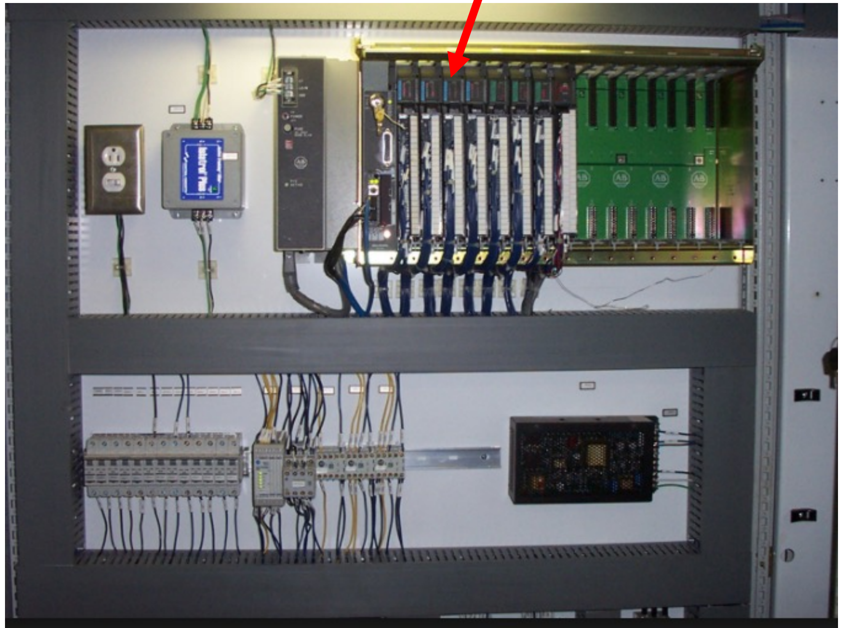
These two illustrations show relay panels, which is a term to indicate that they were installed before PLCs were used. Seldom will you find any relay panels left. Most all control panels have been moved to PLCs, or other methods of solid-state control.

Just a few things about PLCs, cont.



Siemens S7 PLC

Allen Bradley PLC-5, PLC

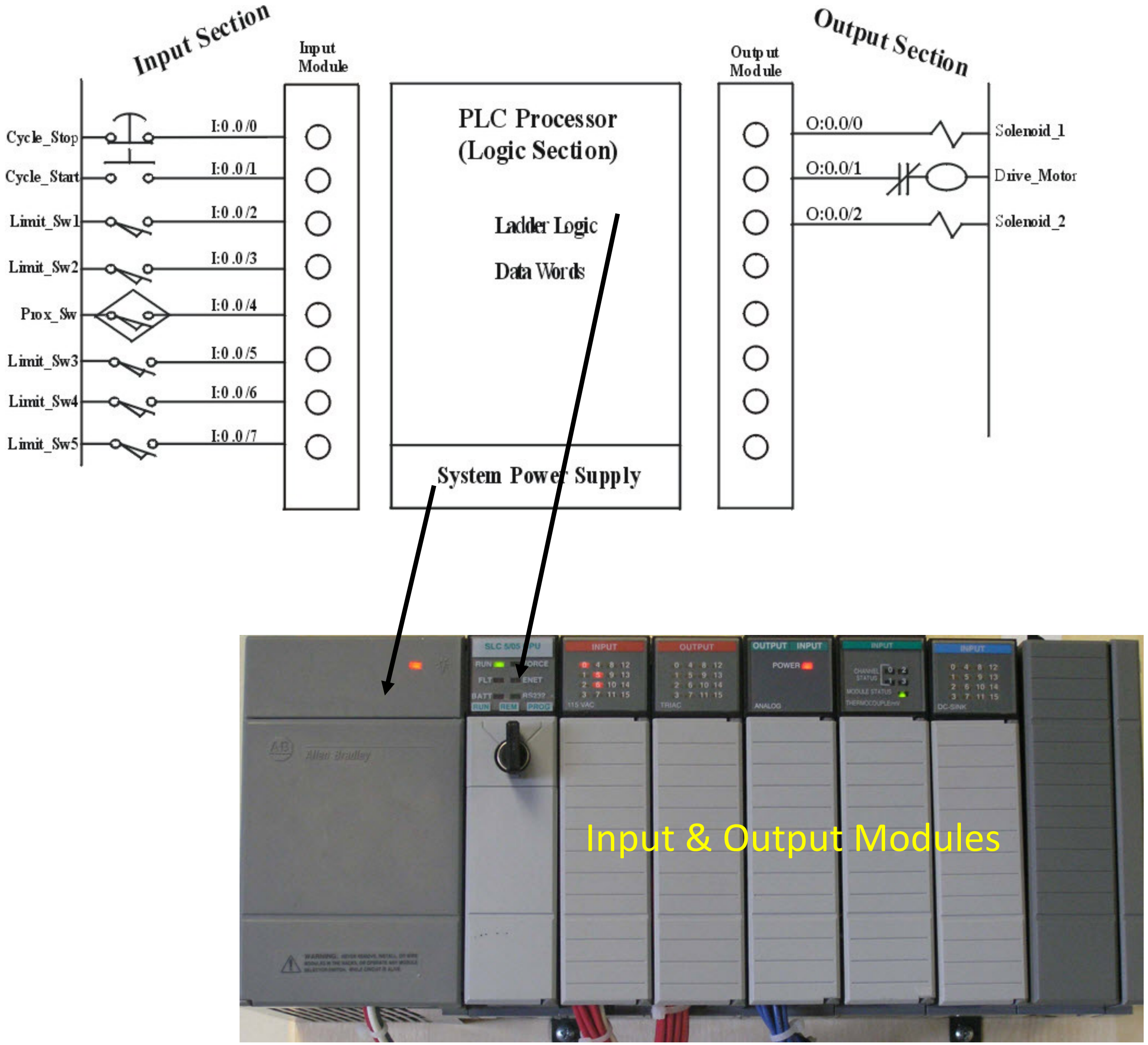


These two graphics will show the inside of a PLC control panel. The first thing to notice is how clean and organized the panel is. Realize that all the logic to run a machine is done in the ladder logic program within the PLC processor, thus there are very few relays, and not a lot of messy wires run between relays and timers.

In the top graphic, a Siemens S7 processor is controlling a machine, and also the 3 variable frequency drive units in the top of the panel.
The lower graphic shows an Allen Bradley PLC-5 PLC that is controlling a production machine.

PLCs have programmable memories, thus to do a changeover to a new machine sequence, will simply be switching to another program inside the PLC processor, or downloading a new program into the processor. There are also fewer moving parts, so there is less wear and tear, which means less maintenance. The PLC and modules are mostly solid state. Solid state devices seldom wear out, but will go bad if subjected to excessive heat or electrical spikes.

Sections of a PLC



- The 4 sections of a PLC are:
  - Processor
  - Input section
  - Output section
  - Power supply

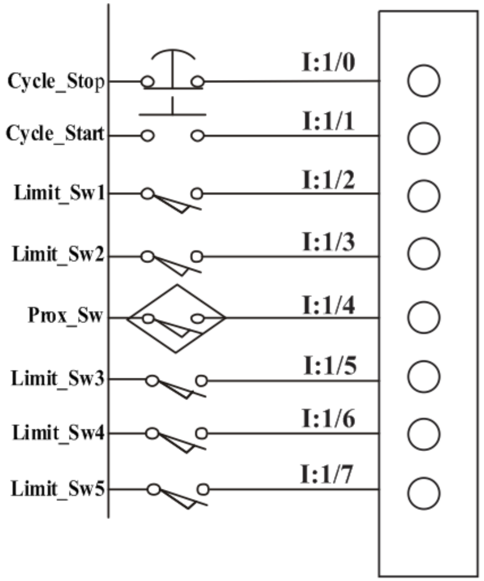
- In this example, the power supply is located on the left edge of the I/O chassis.

- The processor is located in the first slot of the I/O chassis. The processor is the brains of the system. This is where the program is stored that actually controls the I/O of the PLC system.

- The input section is where the input modules are that will bring signals into the PLC processor to analyze.

- The output section is where the output modules are that turn real outputs on and off, based on the design of the PLC program.

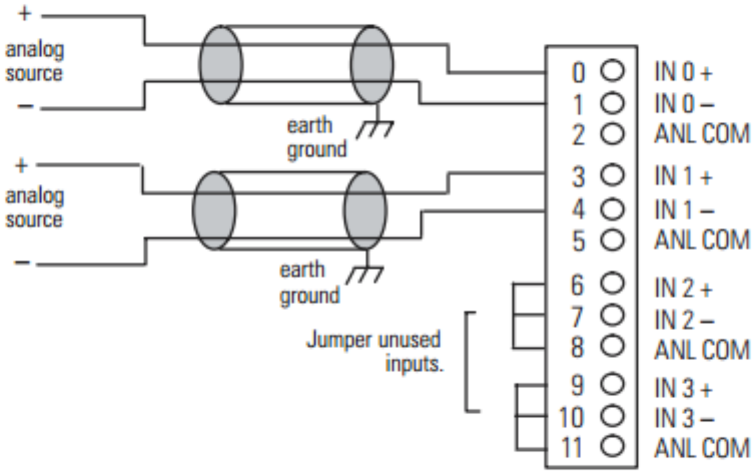
Discrete versus Analog I/O



Discrete Input Module

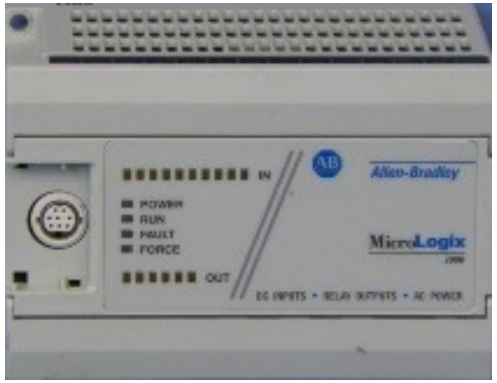
This illustration shows the difference between a discrete and analog input module. Discrete I/O means on/off I/O, or sometime called digital I/O. The state of the input or output device is on or off. An example of discrete inputs are pushbuttons, limit switches, proximity switches, pressure switches, etc. An example of a discrete output is a motor starter coil, pilot light, contactor coil, and solenoid coil. There is also a status indicator for each discrete input or output on a PLC I/O module.

Analog Input Module

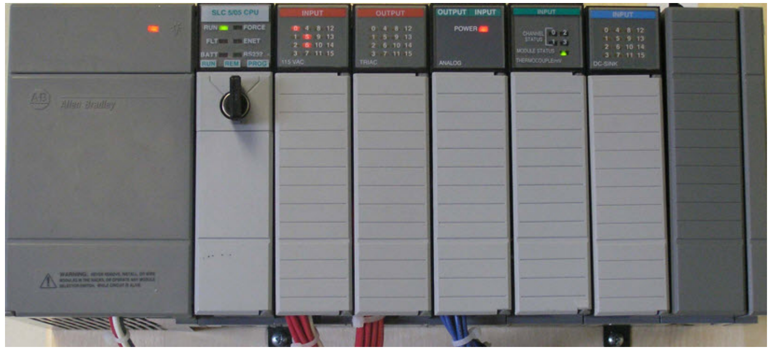


An analog input is fed with an analog input signal, which will typically be a 4-20 mA, or 0-10Vdc signal. This is a signal that varies according to a sensor. As an example, if a thermocouple sensor was sensing an oven that was from 100 to 300 degrees, the system would be setup to send a 4 mA signal to the analog input module at 100 degrees, and a 20 mA signal when the oven is 300 degrees. There are also analog output devices which will be covered in other courses.

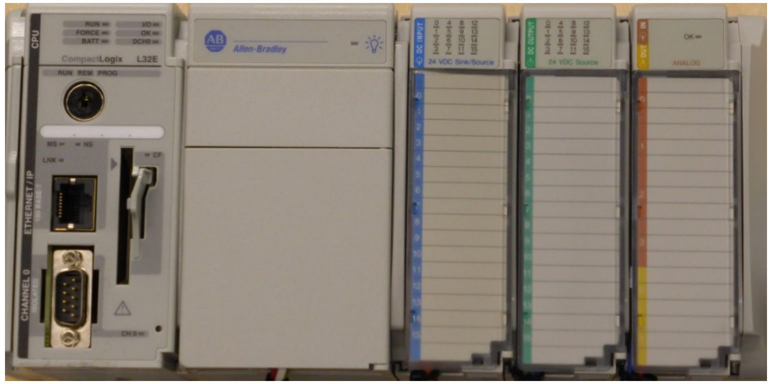
Fixed I/O versus modular I/O



Fixed I/O type of PLC



Modular I/O system
Mounted in an I/O chassis



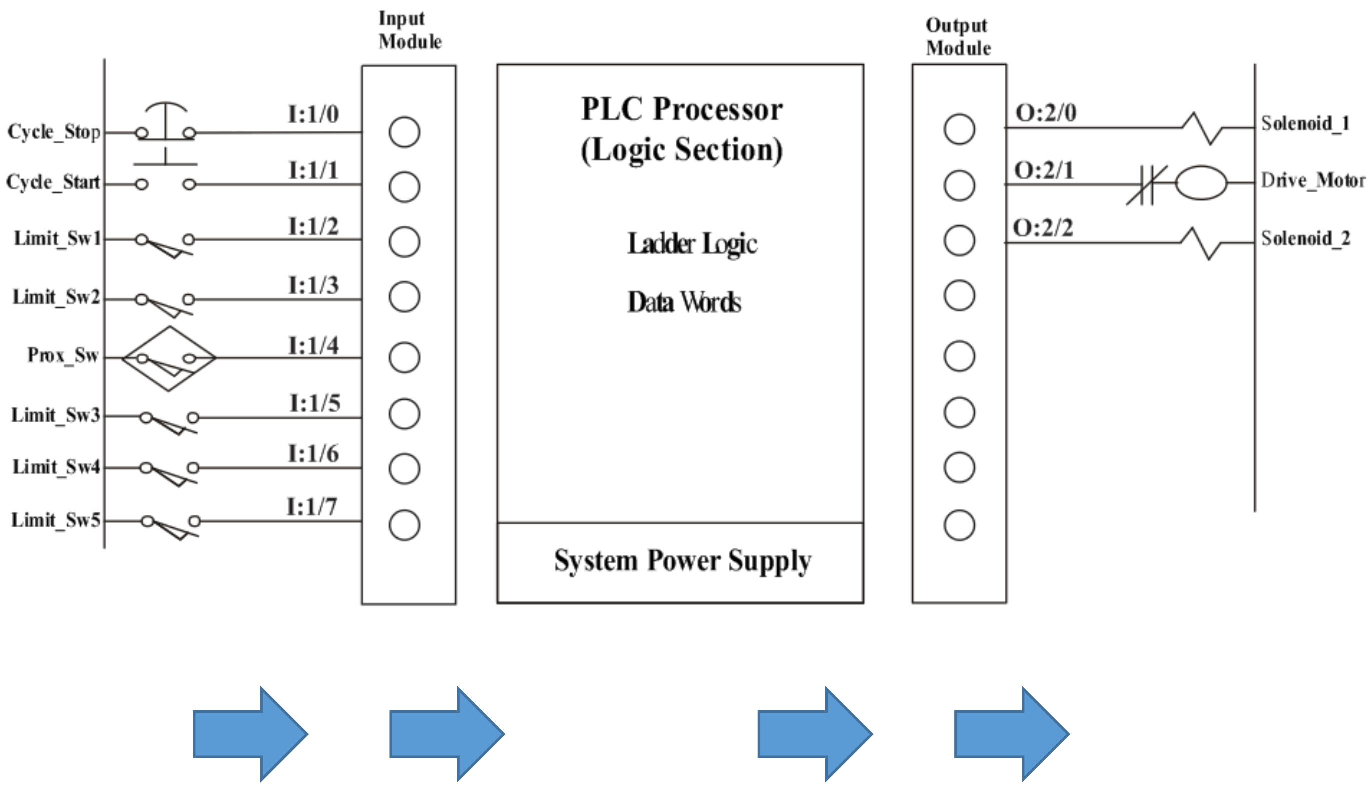
Modular I/O system
Mounted on a DIN rail

This illustration shows different types of I/O systems. The top graphic is a fixed I/O unit. These are set I/O which has the processor and power supply all mounted into one unit. The advantage of this unit is that it has low cost. The problem is when one I/O goes bad, the whole unit must be replaced.

The second graphic is the SLC-500 system, that has an I/O chassis (sometimes called a rack). The power supply feeds power to the backplane of the rack, and the processor is in modular format and plugs into the chassis.

The lower graphic is the CompactLogix system, which is a system that has the modules plug together from one to another, and they are mounted on a DIN rail. The processor plugs into one side of the power supply, and the I/O modules plug into the other side of the power supply.

Basic operation of a PLC system:

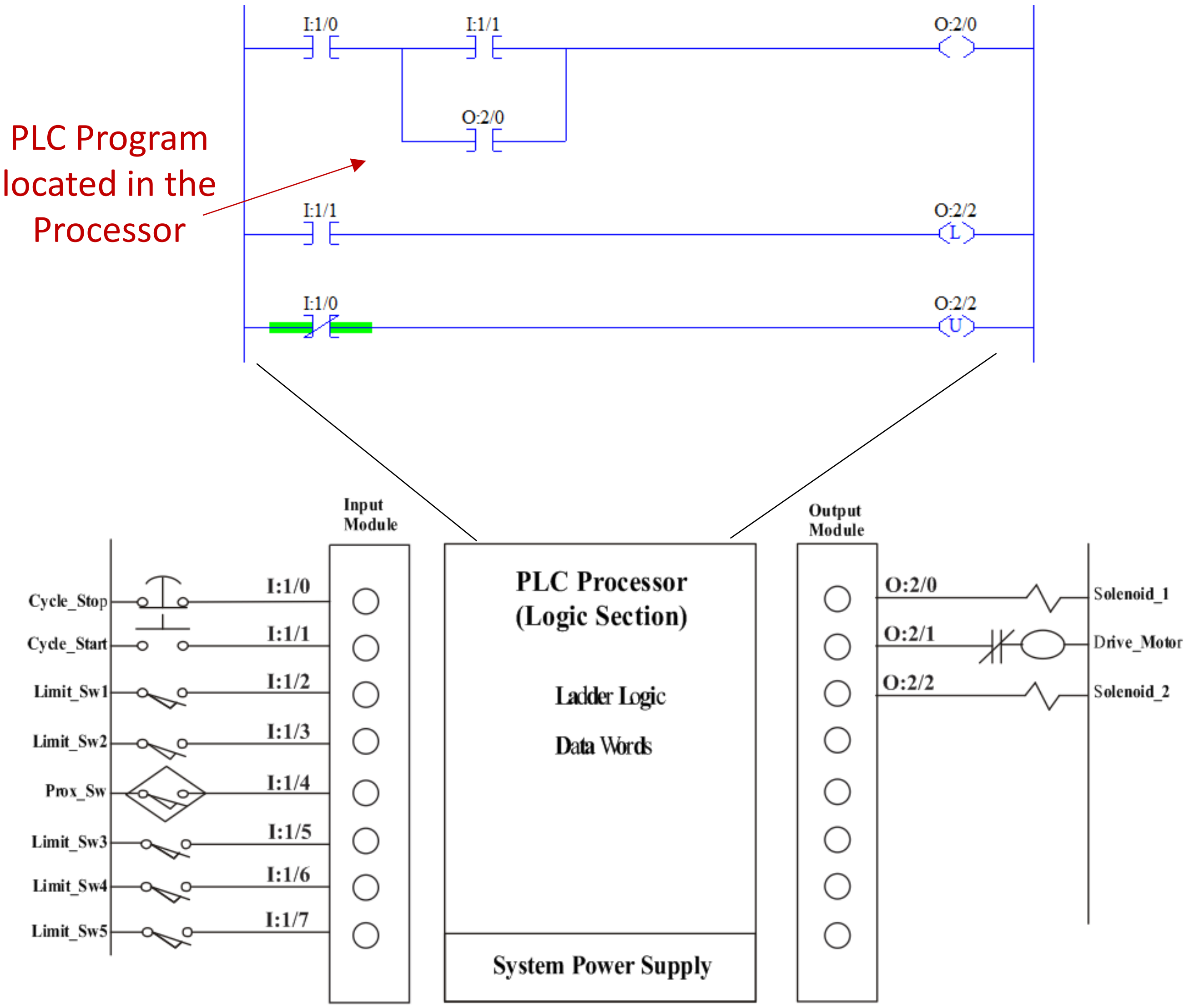


Signals flow from the left to the right on a PLC system. Signals come into the input modules from input devices, then the information comes into the processor, then the processor turns outputs on/off according to the program in the PLC.

This illustration the operation of the PLC system, and the signal flow. This is an important concept to understand, especially when troubleshooting a system. Signals flows conceptually from the left to the right. Input devices are turned on, which in turn sends a signal into the PLC input module (the status light for that input will come on). The input status then controls the logic flow in the ladder logic program within the processor. The PLC program then turns on the output instructions within the program, based on logic power flow (which will be explained later). These output instructions turn on the real outputs (solenoid coil, motor starter coils, etc. The output status light on the output module will also be on if an output is supposed to be turned on.

I/O status lights are used to assist the user in troubleshooting a faulty system. Actuating an input switch (such as a limit switch) will turn on the input status light. The output status light on a discrete output module is turned on by the output instruction within the PLC program.

Basic operation of a PLC system, cont.



PLC Program located in the Processor

The PLC program is designed to control the operation or sequence of a machine controlled by the PLC system.

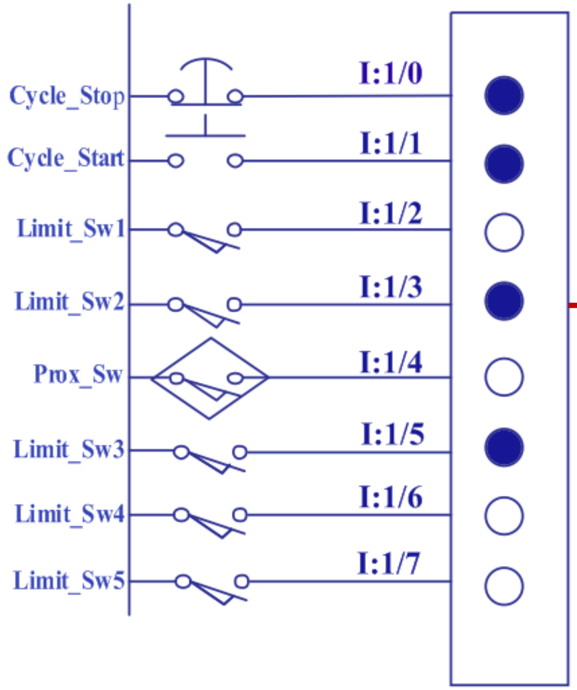
The PLC program turns on specific outputs, based on the condition of inputs on the machine

The PLC program is created on a program panel (computer with the PLC programming software), and can be viewed with the program panel.

The PLC program is typically created by the company that built the machine the PLC controls. This is typically called an Original Equipment Manufacturer (OEM). If an older machine is retrofitted with a new control system, the company doing this is sometimes called a Systems Integrator. They do primarily control systems design and installation.

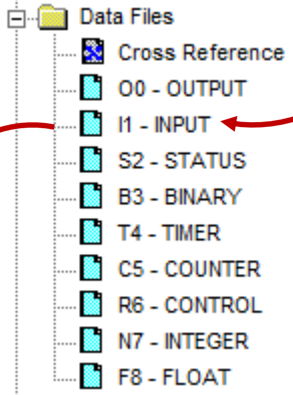
It is not common to have an Engineer within a company design the control system or the PLC program. This can create a liability issue for the individual and the company.

Discrete Input Data:



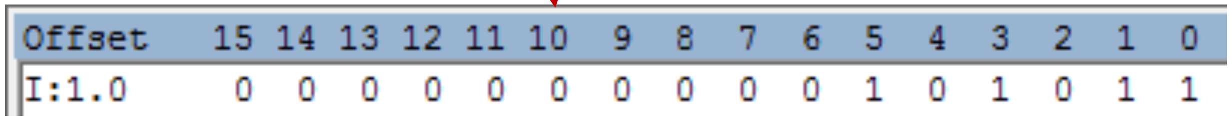
120VAC
Input Module

120VAC Input
data is converted
to a logic level (5V)
and sent to processor



- Data Files
  - Cross Reference
  - O0 - OUTPUT
  - I1 - INPUT
  - S2 - STATUS
  - B3 - BINARY
  - T4 - TIMER
  - C5 - COUNTER
  - R6 - CONTROL
  - N7 - INTEGER
  - F8 - FLOAT

The Input Image Table

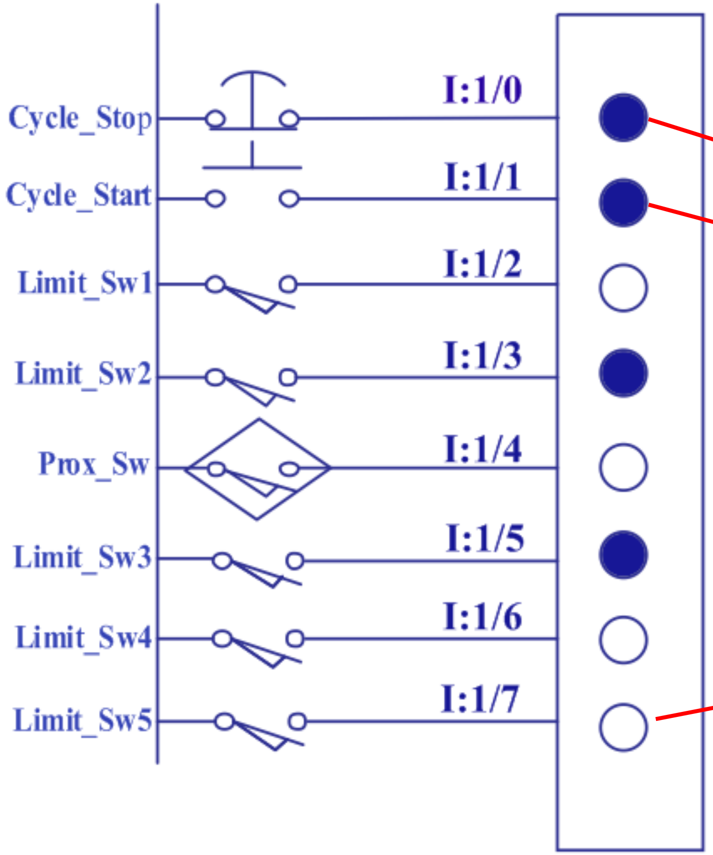


Offset	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I:1.0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1

This illustration shows the flow of data from the input module to the input image table. This example shows 8 input points (sometimes called bits) that have input devices wired to them. A darkened indicator light means the input light is “on”, which means that a signal is coming into the input point. As an example, input I:1/5 indicator light is on, so Limit_Sw3 is activated (made), since it is a normally open limit switch. The input module then converts the signal to lower level of voltage that the processor can handle (typically 5Vdc). The processor actually goes out to get this data from the input module during the processor scan sequence. We will discuss the scan sequence more in depth later, but for now just realize that the information from the input module is stored in a memory location within the processor as “1s” and “0s”. If a status light on an input point is “on”, it will store a “1” in the corresponding address within the processor. If the status light on the input bit is “off”, it will store a “0” in the corresponding bit address within the processor. This data is stored in the input data file of the processor. For many years this was termed the input image table, since it stored an image of the indicator lights in a data file.

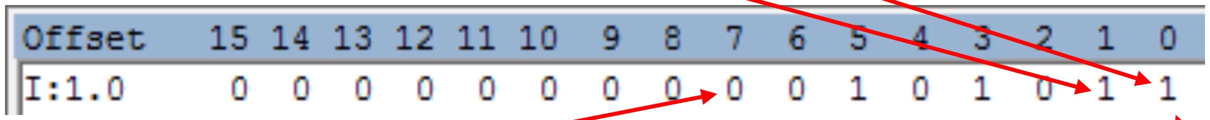
Discrete Input Data, cont.

This illustration shows a more in depth look at the data that goes into the input image table. If the indicator light is on, that means there is an input signal (in this case, it is 120V at the input terminal), thus the processor will write a "1" in the corresponding bit in the input image table.



When the input indicator is on, a "1" is written into the corresponding bit in memory

Input Image Table located in the processor



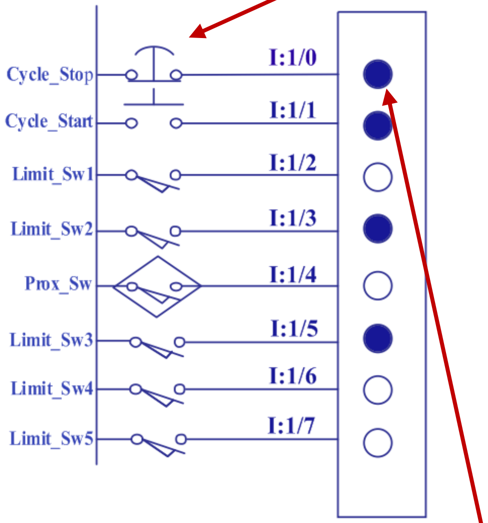
Offset	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I:1.0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1

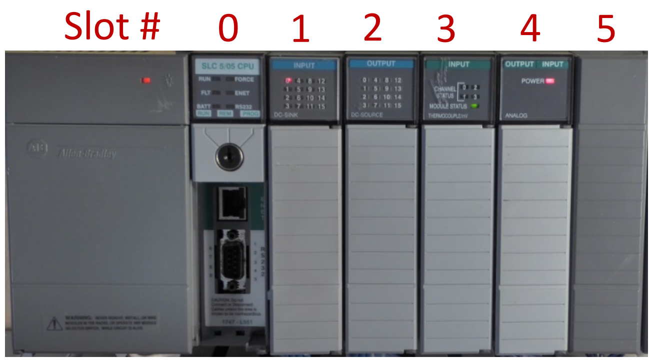
When the input indicator is off, a "0" is written into the corresponding bit in memory

Input I:1/0 in the PLC memory

Discrete Input Data, cont.

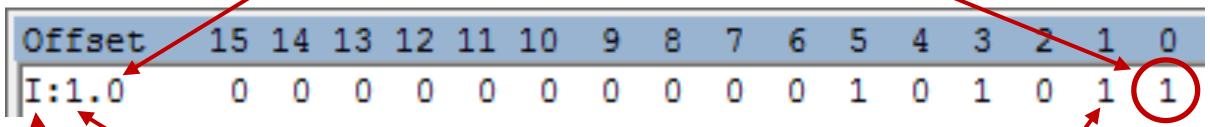
Normally Closed Stop Pushbutton





Bit Address: **I:1/0**

Word Number



Offset	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I:1.0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1

Input File

Slot Number

Bit Number

Though we will discuss I/O addressing a little later in this document, it is important to understand the addressing on the input module and the addressing within the input data file and how they correlate. The input bit address is based on location of the input module in the I/O chassis. The chassis has slots in it. The processor is always slot "0". On the SLC-500 training units, the discrete input module was placed in slot 1. If we focus on the Cycle_Stop pushbutton, it is wired to input terminal 0 of the input module that is located in slot 1, thus the input address is I:1/0.

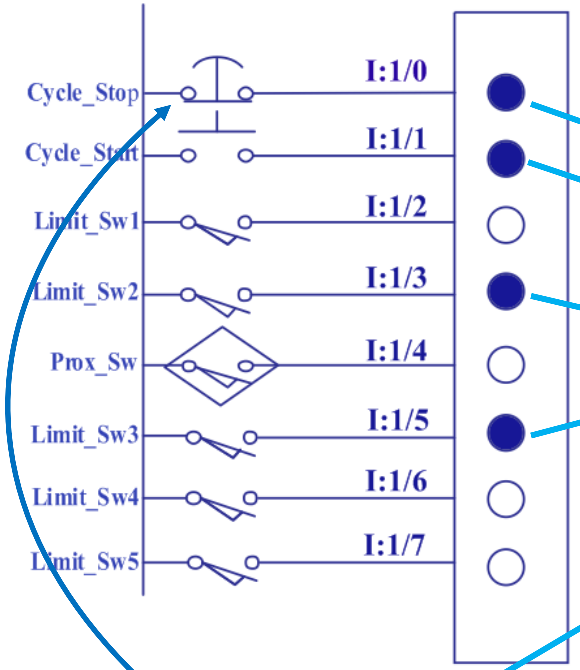
In the input image table, the Word address (located under the word Offset) is I:1.0. This is a little confusing, but is simply means I for input, 1 for slot number, and 0 for Word 0. A word has 16 bits, thus in some cases there may me more than 16 bits in a slot number.

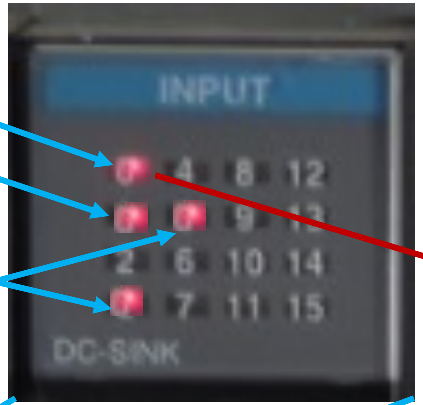
Notice that there are 16 bits to the right of this slot address.

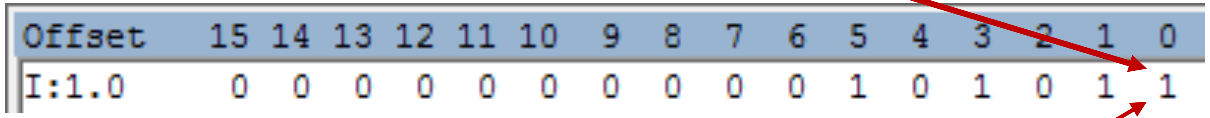
In this illustration, the stop pushbutton passes power, brings a high signal to the input terminal, which turns on the input status light, then writes a "1" into the corresponding bit in the input image table.

Discrete Input Data, cont.

This illustration shows a more in depth look at the data that goes into the input image table. If the indicator light is on, that means there is an input signal (in this case, it is 120V at the input terminal), thus the processor will write a “1” in the corresponding bit in the input image table.

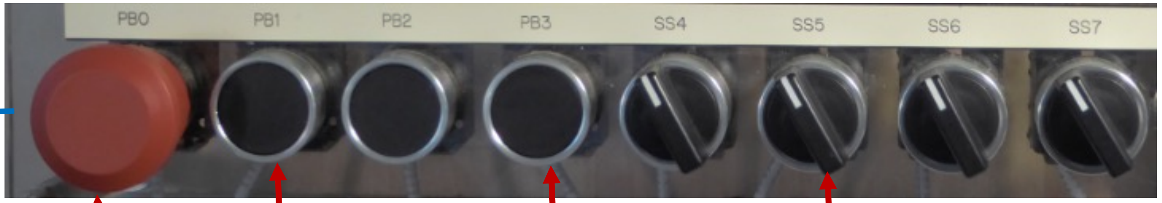






```
Offset   15 14 13 12 11 10  9  8  7  6  5  4  3  2  1  0
I:1.0     0  0  0  0  0  0  0  0  0  0  1  0  1  0  1  1
```

Input I:1/0 in the PLC memory



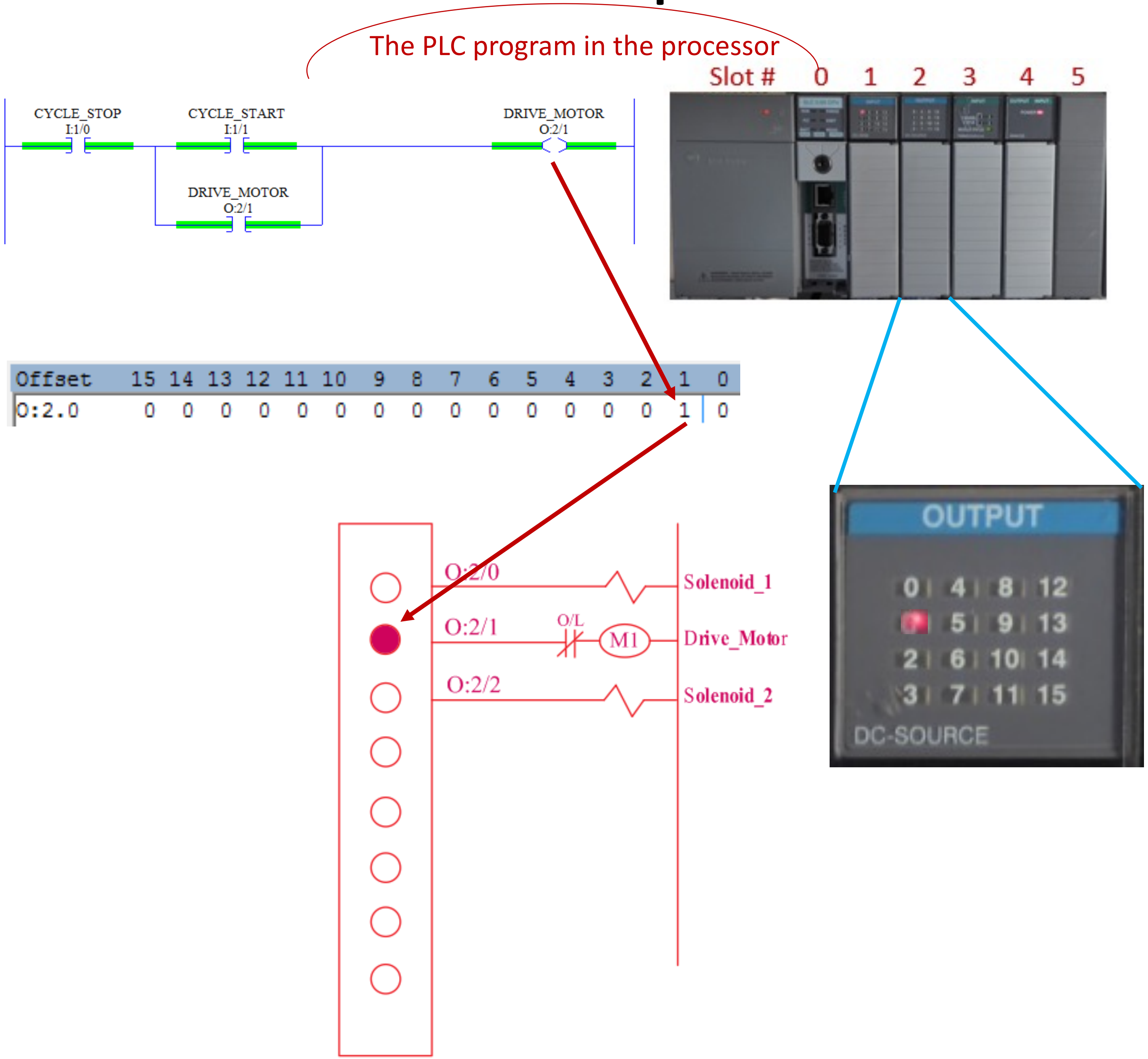
I:1.0
Not
Actuated

I:1.1
Actuated

I:1.3
Actuated

I:1.0
Actuated

A discrete PLC output:

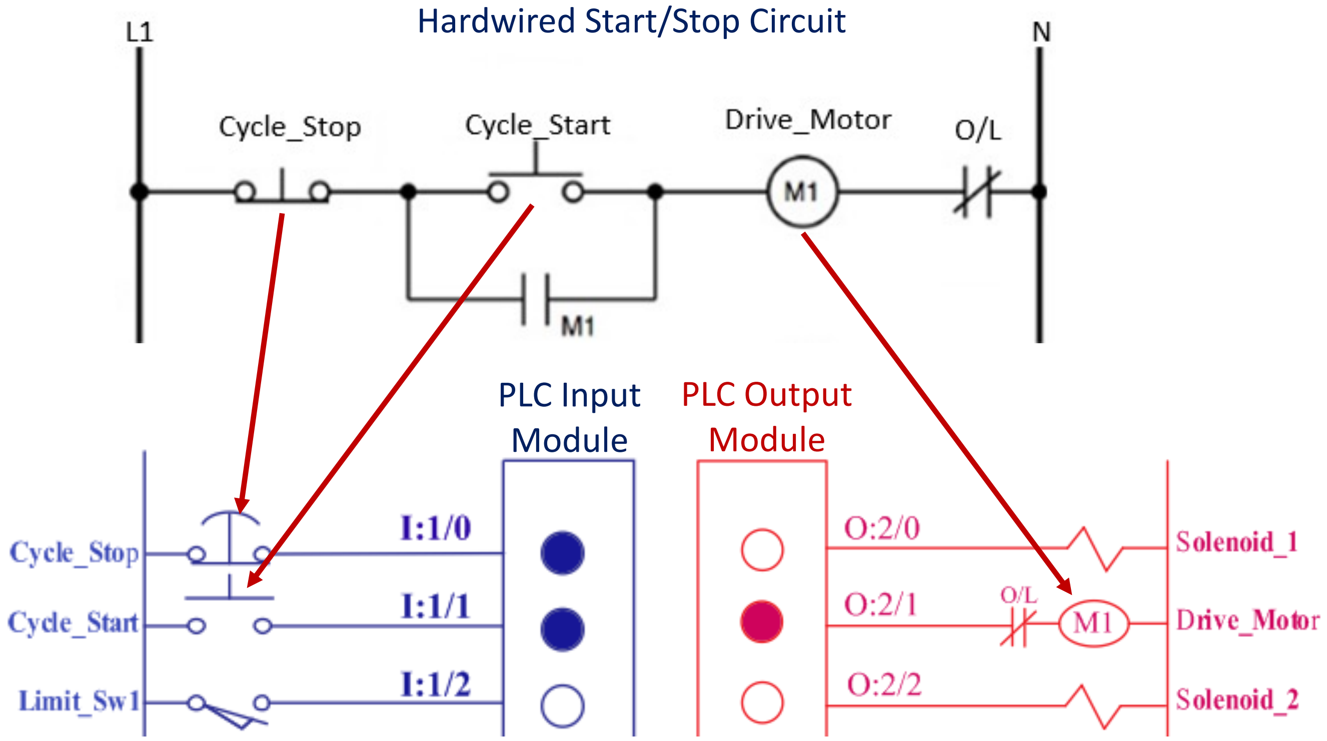


The PLC program in the processor

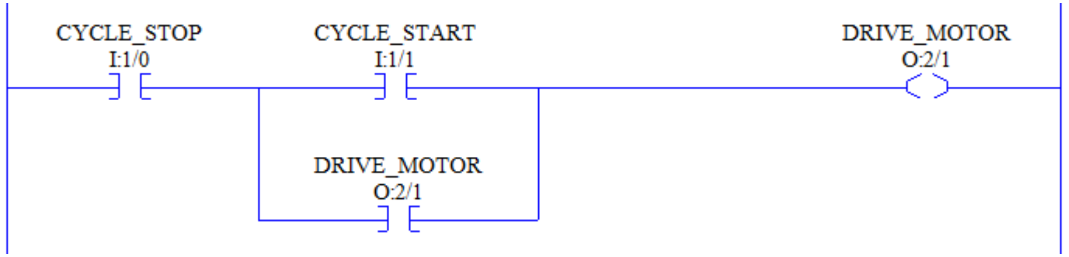
Offset	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
O:2.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

This illustration show how the PLC program controls the outputs on a PLC system. Input signal are brought into the input module, read into the input image table (as shown in previous slides). The processor scans the ladder logic and references any ladder logic instructions with an input address, to determine if the instruction has logic power flow (highlight). As you can see in the

Hardwired circuit to PLC circuit & program



The PLC program in the processor

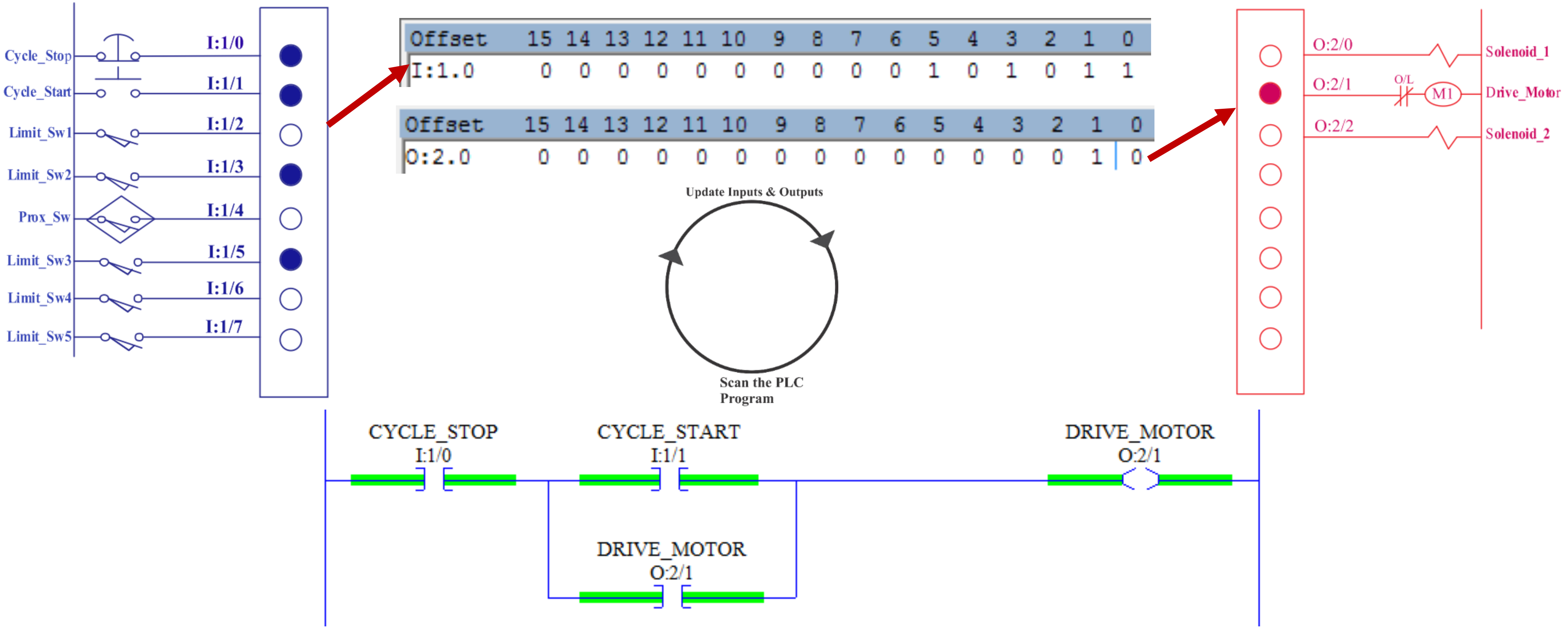


This illustration shows a start/stop, 3-wire control circuit, that all students would be familiar with. The next step is to wire the input devices and output device to the input & output modules. Notice that each device then is assigned an I/O address. The input device addresses start with an I, the output device address starts with an O.

The lower diagram is the ladder logic program, that will be found in the processor. Notice that the addresses assigned to the instruction (N.O., N.C, Coil) in the ladder logic program, align with the addresses on the I/O modules.

Also notice that the device names found on the wiring diagram in the top graphic, is also assigned to the addresses within the program. We will look at all of this information in depth.

The Processor scanning sequence

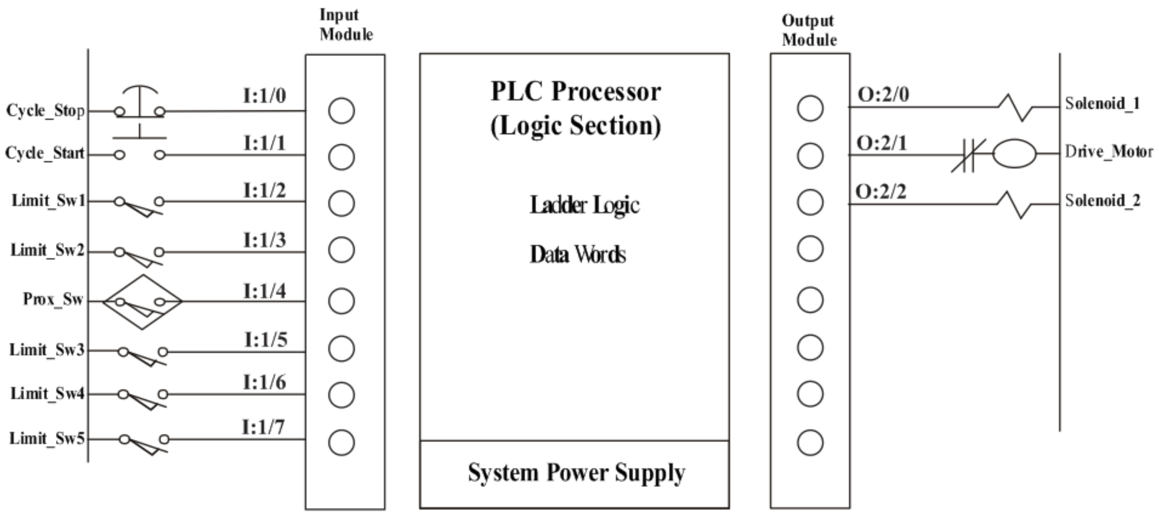


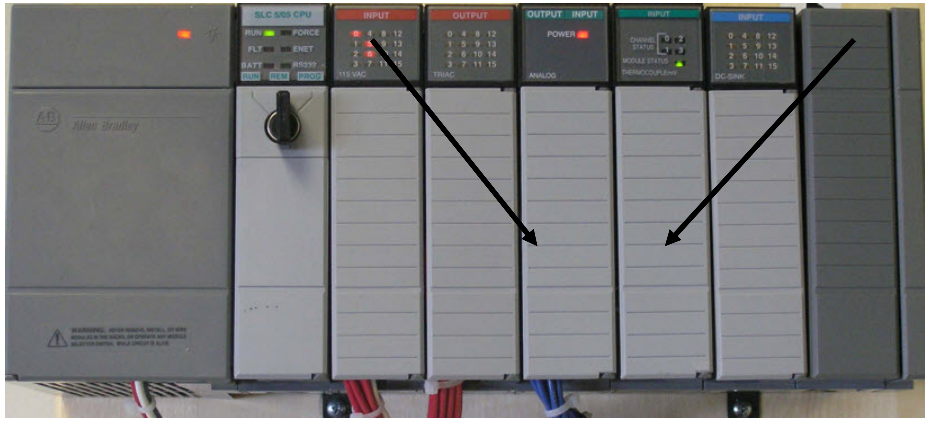
Offset	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I:1.0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1

Offset	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
O:2.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

This illustration shows the processor scanning sequence. Realize that this is primarily the way a traditional PLC (SLC-500 & PLC-5) works, but the PACs function a little differently. The processor updates the inputs & outputs, which means the processor goes to the input modules and writes the 1s & 0s into the input image table, and then goes to the output image table, and writes the 1s & 0s out to the discrete output module, which will affect the real output devices. A “1” will turn on an output device. The processor then goes to the PLC program (ladder logic) and scans/analyzes the instructions compared to information in the input image table and other data within the data files, then turns on the output instructions, which will then write the information into the output image table.

Input and Output Modules

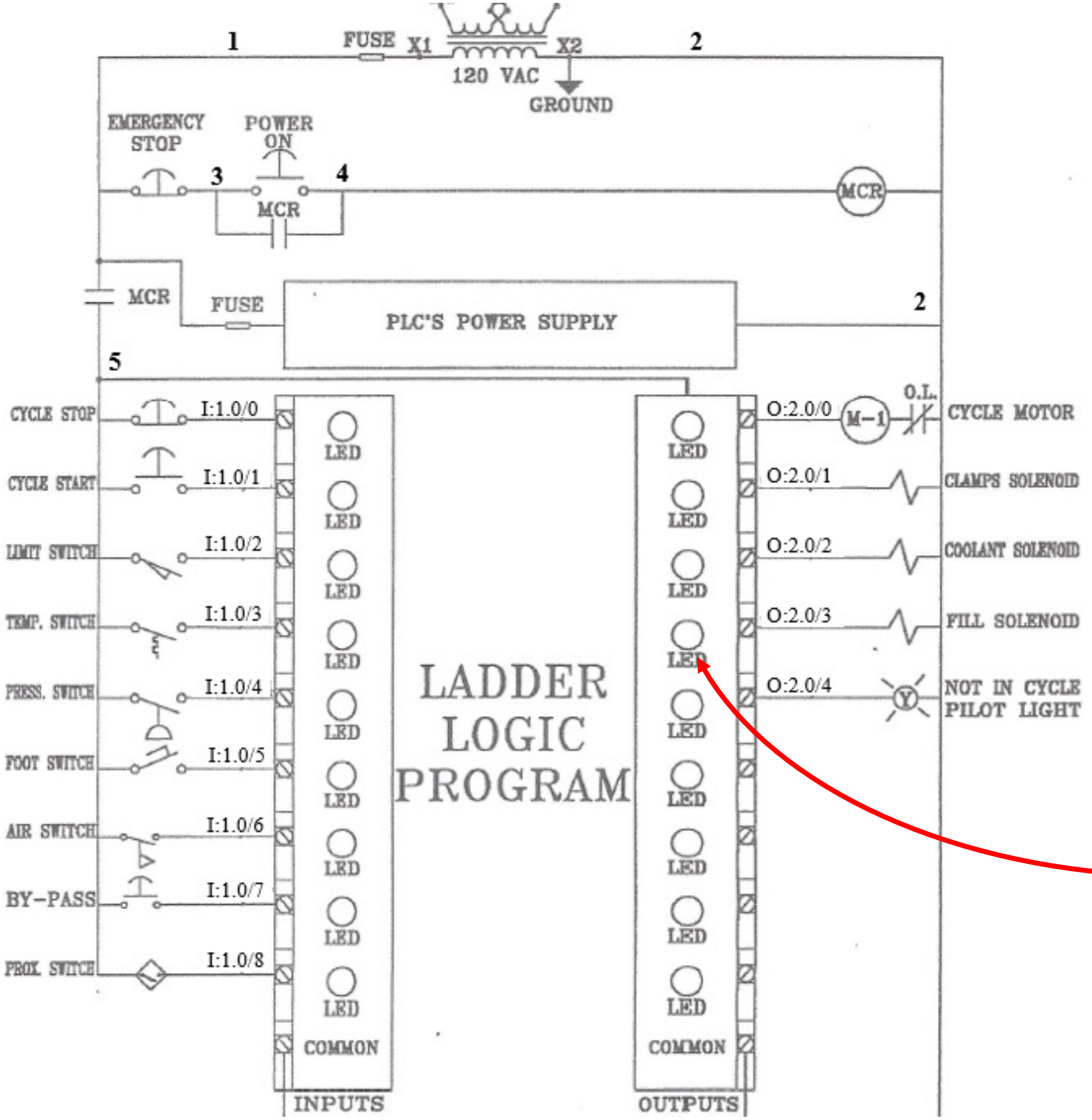


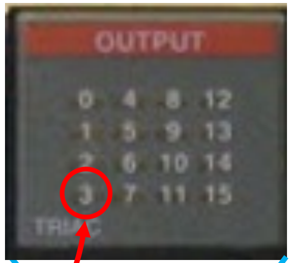


This illustration shows the input and output modules in the I/O chasses of an SLC-500 system.

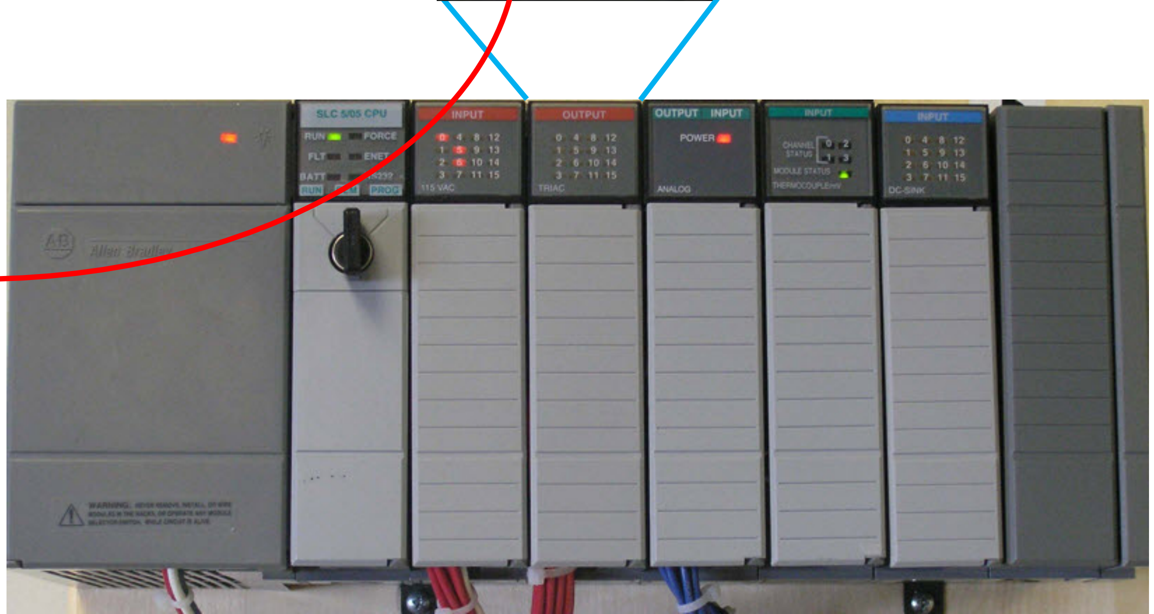
Electrical Print & I/O Addressing

This illustration shows the input and output modules in the I/O chasses of an SLC-500 system, and an electrical diagram of a PLC system.   In this example, if the FILL SOLENOID quits working on a machine, the Technician needs to look at the electrical print, to see the output address. In this case it is O:2.0/3. The Technician would then go to the output module, find the status light for the output to see if it is on. This is important to understand for troubleshooting.

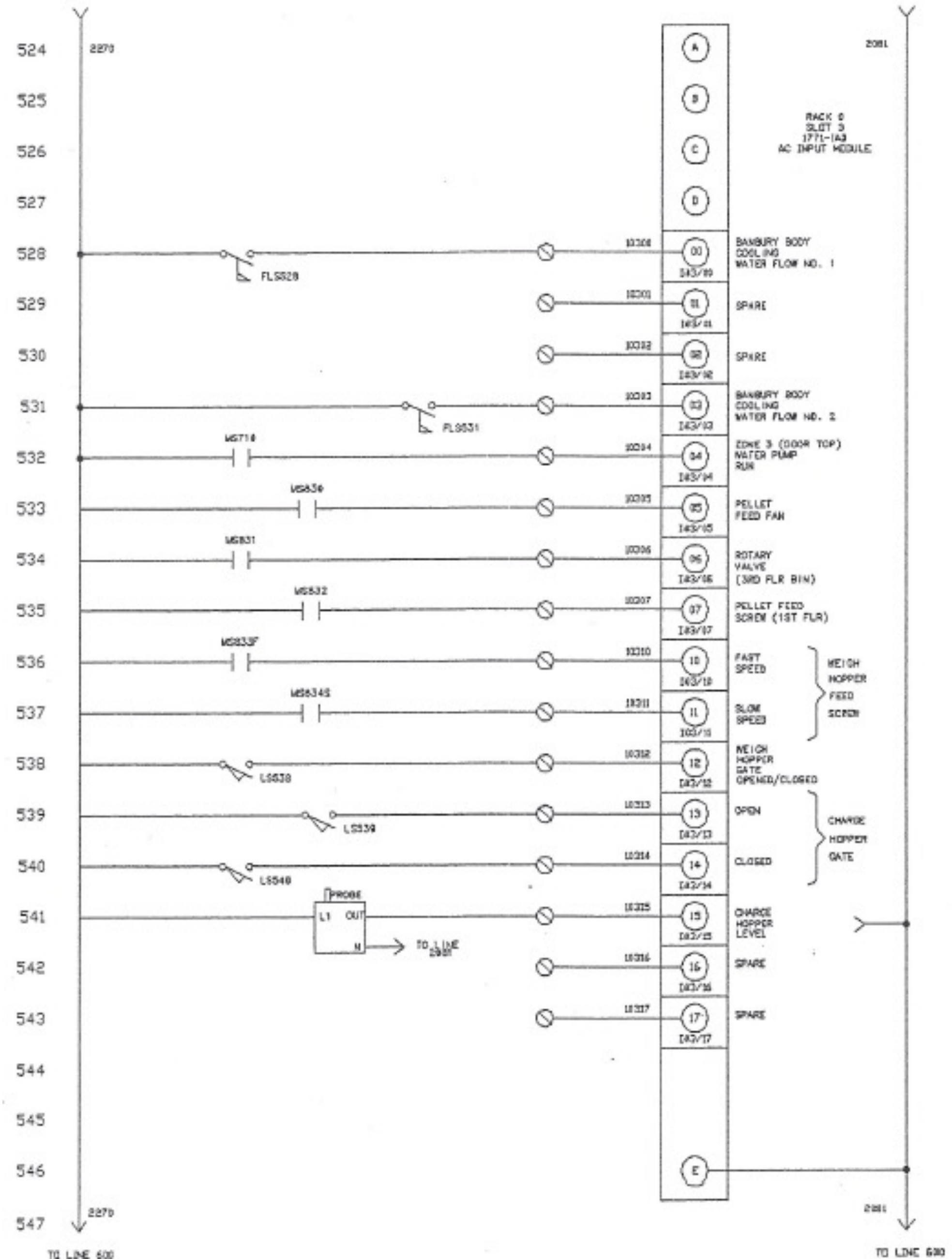




Status light 3 on a discrete output module in slot 2, would be address: O:2.0/3



Allen Bradley PLC-5 Discrete Input Module Wiring

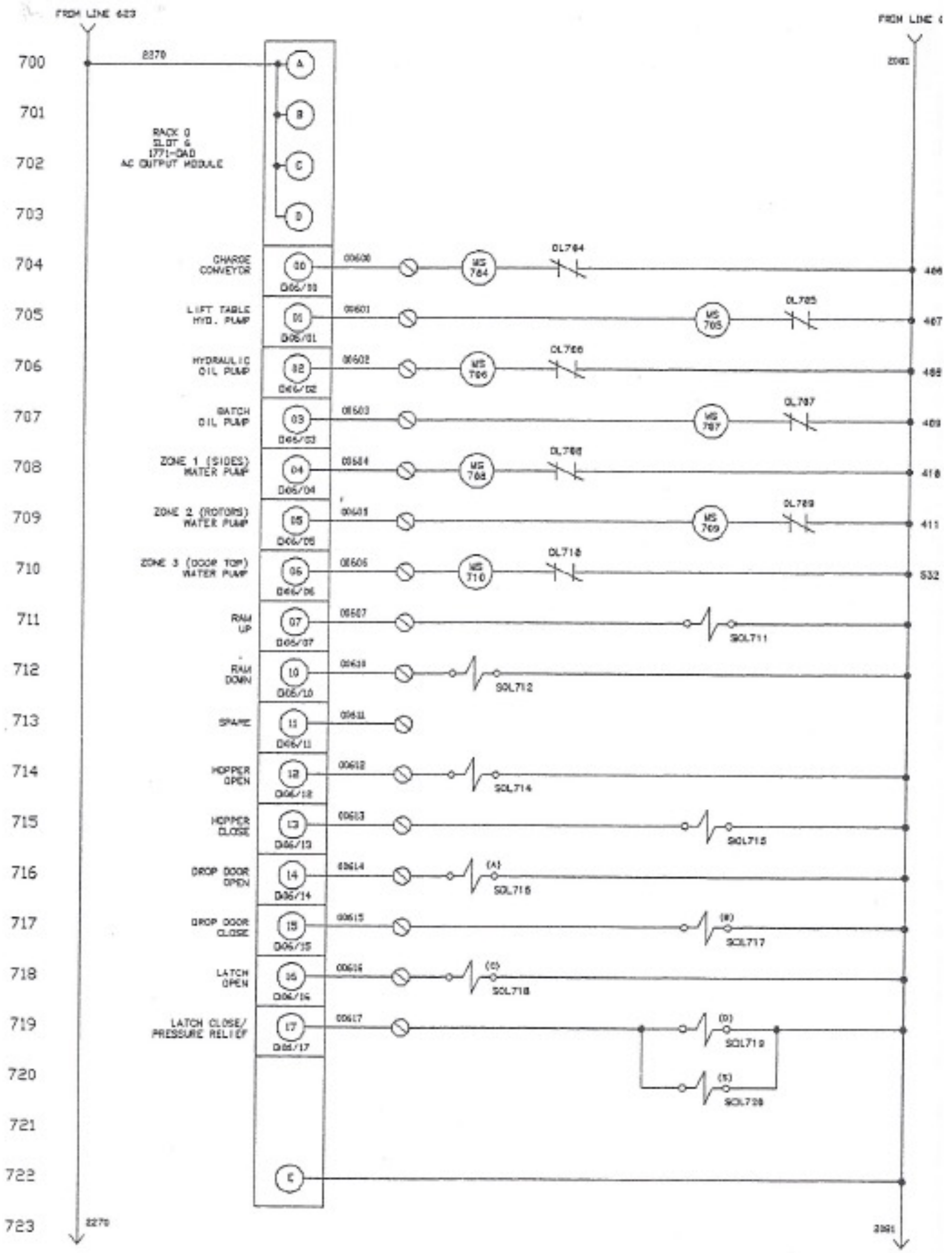


This illustration shows the input module wiring on an Allen Bradley PLC-5 system. Notice there are line numbers on the left hand side of the print, such as 528. This means page 5, line 28 of the set of electrical prints.

Notice there is a description to the right of each of the discrete inputs.

This print also shows the module wiring.

Allen Bradley PLC-5 Discrete Output Module Wiring



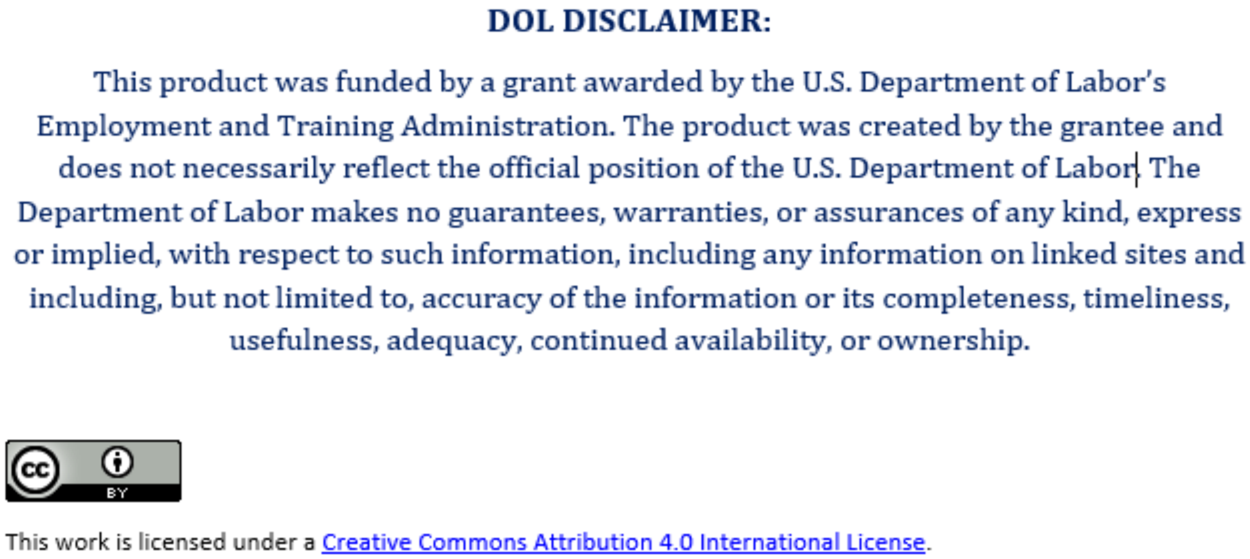
This illustration shows the output module wiring on an Allen Bradley PLC-5 system. Notice there are line numbers on the left hand side of the print, such as 716. This means page 7, line 16 of the set of electrical prints.

Notice there is a description to the left of each of the discrete outputs.

This print also shows the module wiring.

Though this electrical print is hard to read, it is important to understand the purpose of these prints. If there is a problem on a machine and an output fails, the Technician must determine the output address of the failed output. It is important to read the descriptions to determine the address, then locate the status light. If the status light is “on”, the output voltage should be checked. If the status light for the output is “on”, the processor is telling it to be on. If there is no output voltage to the output, then the output module may be faulty, and will need to be replaced.

This Concludes this Instructional Document



DOL DISCLAIMER:

This product was funded by a grant awarded by the U.S. Department of Labor's Employment and Training Administration. The product was created by the grantee and does not necessarily reflect the official position of the U.S. Department of Labor. The Department of Labor makes no guarantees, warranties, or assurances of any kind, express or implied, with respect to such information, including any information on linked sites and including, but not limited to, accuracy of the information or its completeness, timeliness, usefulness, adequacy, continued availability, or ownership.