ECET 4530

Industrial Motor Control

Programmable Logic Controllers

1

Programmable Logic Controllers

Programmable Logic Controllers, or **PLCs**, are real-time, event-driven, process-control computers that are used to automate industrial (and many other) processes.

Originally used by the automotive industry to replace hard-wired, relay-logic-based control systems that were difficult to alter or update, PLCs were designed to withstand the stresses that may be present in an industrial setting such as vibration, electrical noise, dust, high humidity, and extreme operating temperatures.

2

Programmable Logic Controllers

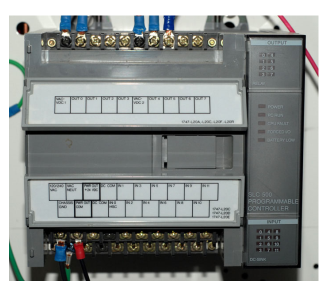
But unlike general purpose computers, whose operating systems are designed to multitask a wide variety of applications without a strong concern regarding latency issues, PLCs and their operating systems are streamlined to perform **real-time automation processes** for which even minor delays in response to changing input conditions could result in unintended or unacceptable operation.

3

Modular vs. Preconfigured PLCs

The different components may be manufactured individually and connected or linked together as part of a **modular system**, or they may be contained in a **preconfigured** enclosure that is assembled at the factory.

Allen-Bradley Compact-Logix Modular PLC



Allen-Bradley SLC-500 Preconfigured PLC

4

Modular PLCs

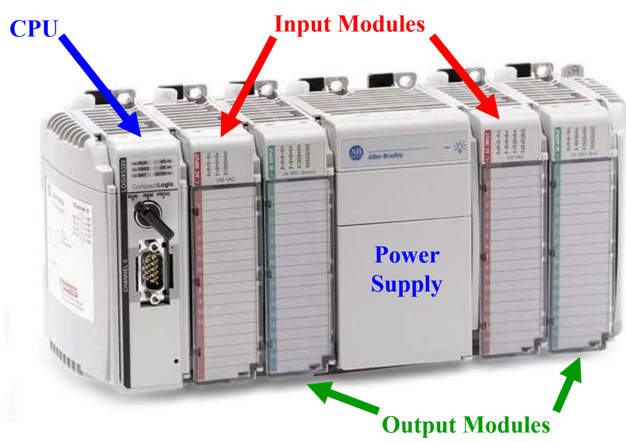
Modular PLCs can be configured using a variety of individual modules, each of which perform a specific function, the selection of which depends upon the needs of the system that the PLC will be used to control.

Allen-Bradley Compact-Logix PLC

Modular PLCs allow for easy expansion or reconfiguration in order to accommodate any changes that may be required in the operation of the control system.

5

Modular PLCs



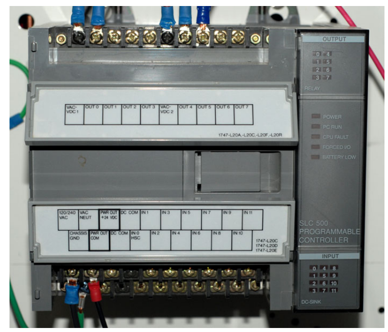
Shown above is a modular Allen Bradley CompactLogix PLC that contains: a CPU, a 12-port input module, a 16-port output module, a power supply, a 16-port input module, and another 16-port output module.

6

Preconfigured PLCs

Preconfigured PLCs are factory assembled with a predefined set of components.

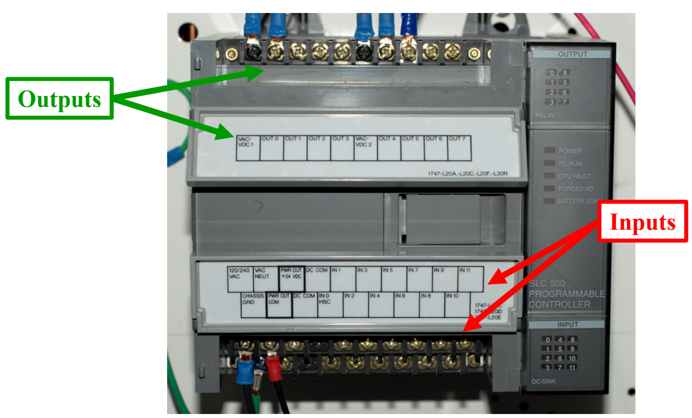
Although not as flexible as modular PLCs, preconfigured PLCs are often relatively inexpensive and come in a variety of configurations that are well-suited for many different applications.



Allen Bradley SLC-500 PLC

7

Preconfigured PLCs



Shown above is a preconfigured Allen Bradley SLC-500 PLC that contains: a CPU, a power supply, 12 input ports, 8 output ports, and an RS-485 serial communication port (not visible in figure).

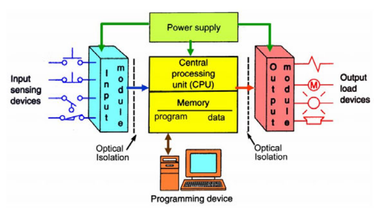
8

Components of a PLC

The fundamental components of a PLC include:

- **Central Processing Unit (CPU)**
- **Input Ports**
- **Output Ports**
- **Communication Interface** (RS-232, USB, Ethernet…)
- **Power Supply**

PLC System



9

PLC Components – CPU

The **CPU** typically contains a **microprocessor**, **memory** to store both the PLC's operating system and the user's program, along with the other hardware required to interface with the other peripheral components of the PLC.

Allen-Bradley Compact-Logix PLC

Some CPUs may include a discrete **programming port** that provides the point of connection required for a user to download a program into the PLC's memory, while others might utilize the PLC's **communication interface** for this purpose.

10

PLC Components – CPU

The PLC's overall operation is determined by a **program** that is typically down-loaded into the CPU's memory using an external computer that has the appropriate software and can either connect directly to the PLC's programming port or to transfer data remotely to the PLC via its communication interface.

Allen-Bradley Compact-Logix PLC

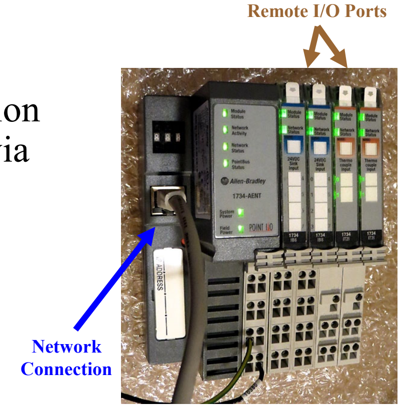
Note that, although a variety of programming languages exist, we will focus on **Ladder Logic programming** since it's well suited for engineers without a programming background.

11

PLC Components – Inputs & Outputs

Traditionally, PLCs communicate with the various control system components by means of a set of **input ports** and **output ports** that are directly connected to the PLC.

Newer PLCs often have communication or network ports that allow for communication with **remote (I/O) system components** via signals sent over a communication bus or via packets of data sent over a computer network.

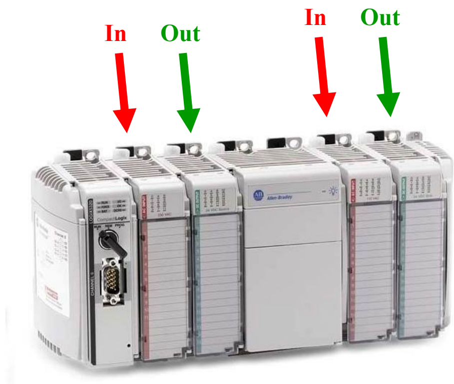


Allen-Bradley Point I/O

12

PLC Components – I/O Modules

Input modules and **output modules** contain ports that allow the PLC to interface with the other control system components.



Allen-Bradley Compact-Logix PLC

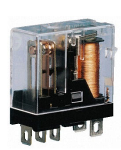
A **port** is an electrical interface that allows the PLC's CPU to either receive signals from external devices (such as push-buttons) or to send electrical signals to external devices (such as the field-coil of a contactor).

Input and Output Ports

The **input ports** provide the means for the CPU to receive control signals or other information from any switches, pushbuttons, sensors, detectors, or other input/logic devices connected to the system.


The **outputs ports** provide the means for the CPU to control or send signals to any external system devices such as relays, contactors, actuators, valves, or indicator lamps.





Input and Output Ports

A PLC used within a motor control system continuously monitors its **input ports** during operation and then, based upon their states and the PLC's program, adjusts the states of its **output ports** in order to control the operation of the motors or other system devices.

Example I: A "Start" button is pressed, temporarily supplying +24V_{DC} to one of the PLC's input ports.

⇓

The PLC detects the +24V_{DC} (high voltage) at the input port and responds by "activating" one of its output ports.

⇓

The activated output port supplies 120V_{AC} to the field-coil of a contactor, actuating closed the contactor's main contacts, in-turn supplying 208V_{AC} to an induction motor.

15

Input and Output Ports

A PLC used within a motor control system continuously monitors its **input ports** during operation and then, based upon their states and the PLC's program, adjusts the states of its **output ports** in order to control the operation of the motors or other system devices.

Example II: An overload relay trips, causing its NC contact to open, in-turn disconnecting +24V_{DC} from a different input port.

⇓

The PLC detects 0V_{DC} (low voltage) at the input port and responds by "deactivating" the output port that supplies the field coil and "activating" an output port that supplies a lamp.

⇓

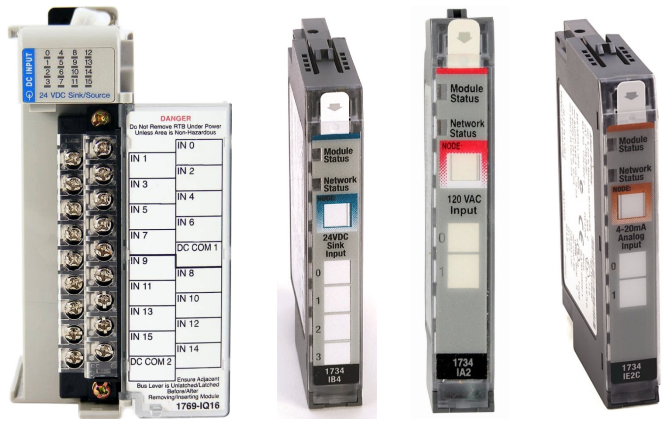
The motor is deenergized when the contactor drops out, and the lamp illuminates to indicate the overload condition.

16

Input Ports

A PLC's **inputs ports** are typically classified into two types:

Discrete Inputs and **Analog Inputs.**



17

Discrete Inputs

A PLC's **inputs ports** are typically classified into two types:

Discrete Inputs and Analog Inputs.

Discrete Inputs are used to detect signals that can have **two states** that are often described as:

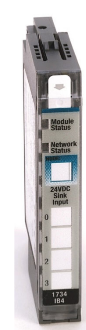
"**True/False**", "**On/Off**", "**High/Low**", "**1/0**"…

Many PLCs have discrete inputs that are rated for $\mathbf{24V_{DC}}$.

A $\mathbf{24V_{DC}}$ **discrete input** will:

- register **True** if $\mathbf{+24V_{DC}}$ **is applied** to its terminal, and
- register **False** if $\mathbf{+24V_{DC}}$ **is not applied** to its terminal.

(i.e. – **False** if $\mathbf{0V_{DC}}$ **is present** at its terminal)



18

Discrete Inputs

A PLC's **inputs ports** are typically classified into two types:

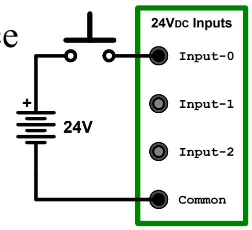
Discrete Inputs and Analog Inputs.

The state of a **discrete input** is often controlled by means of a logic device, such as a pushbutton or a contact.

For example:

if a NO pushbutton is wired between a $+24V_{DC}$ source and a $24V_{DC}$ discrete input, then the input will:

- register **True** while the **button is pressed**, and
- register **False** while the **button is not pressed**.



19

Analog Inputs

A PLC's **inputs ports** are typically classified into two types:

Discrete Inputs and **Analog Inputs**.

Analog Inputs are used to detect and measure the value of signals that can take-on a **range of values**, such as the signal provided by a temperature sensor that is designed to measure temperatures that range from 0° – 100°C.

Common analog input-signal ranges include:

0V to +10V, **–10V to +10V**, **4mA to 20mA**, …

20

Analog Inputs

A PLC's **inputs ports** are typically classified into two types:

Discrete Inputs and **Analog Inputs**.

Many **sensors** (temperature, pressure, flow, etc.) used within industrial control systems include a **transmitter** that converts the measured quantity to a standard analog signal (voltage or current).

For example:

a 4–20mA transmitter may be calibrated for use with a 0–100°C temperature sensor such that the transmitter outputs 4mA when the temperature is 0°C and 20mA when the temperature is 100°C, varying linearly as the temperature varies from 0–100°C.

21

Analog Inputs

A PLC's **inputs ports** are typically classified into two types:

Discrete Inputs and **Analog Inputs**.

Thus, a 4–20mA transmitter that is calibrated for use with a 0–100°C temperature sensor can be wired to a 4–20mA analog input port.

The analog input port will measure the current flowing into the port and convert the measured current to a numerical value that can then be referenced by the PLC's program as part of the conditional logic that is utilized to define the operation of a control system.

Note that the details regarding the wiring and operation of the various types of sensors, transmitters, and analog inputs are beyond the scope of this presentation.

22

Output Ports

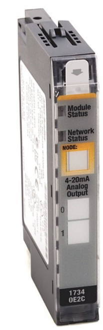
A PLC's **outputs ports** are typically classified into two types:

Discrete Outputs and **Analog Outputs**.

Discrete Outputs provide output signals that have **two states**;

"**On**" (True) or "**Off**" (False).

Analog Outputs provide output signals (voltages or currents) that can take on a continuous **range of values**.



23

Discrete Outputs

Discrete Outputs provide output signals that have two states;

"**On**" (True) or "**Off**" (False).

Discrete outputs can be further classified into two subcategories:

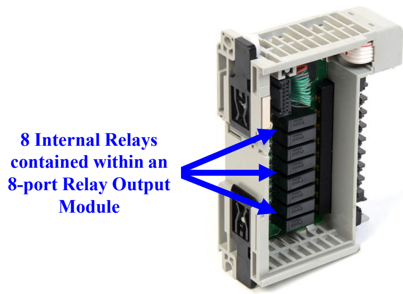
Relay Outputs and **Solid-State (Electronic) Outputs**.

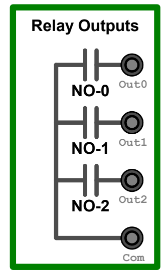
24

Relay-type Discrete Outputs

Relay Outputs are **discrete outputs** that utilize physical relays, each of which contain a **NO contact** that can be used to make or break any external circuits connected to their terminals.

The states of the NO contacts are controlled by the circuitry contained in the module.





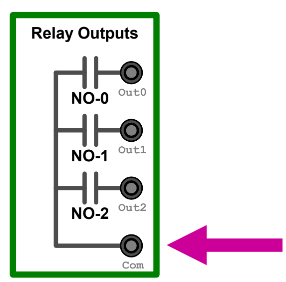
25

Relay-type Discrete Outputs

Relay Outputs are **discrete outputs** that utilize physical relays, each of which contain a **NO contact** that can be used to make or break any external circuits connected to their terminals.

The states of the NO contacts are controlled by the circuitry contained in the module.

The non-port sides of all the NO contacts are internally wired to a **common terminal** that can be connected externally to either the energized conductor of a power source or to ground depending on whether the ports are setup for ***sourcing*** or ***sinking*** operation**.**



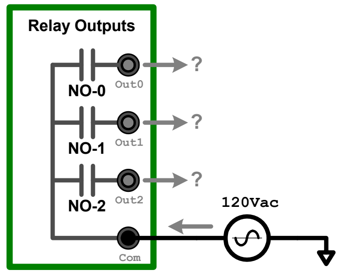
26

Relay Outputs – Sourcing Configuration

Relay Outputs are **discrete outputs** that utilize physical relays, each of which contain a **NO contact** that can be used to make or break any external circuits connected to their terminals.

Sourcing outputs are outputs that "source" or push current **out** of the ports in order to supply power to any connected loads.

In this configuration, the currents supplied by the individual ports are provided by a single **source** that is connected to the **common terminal**.

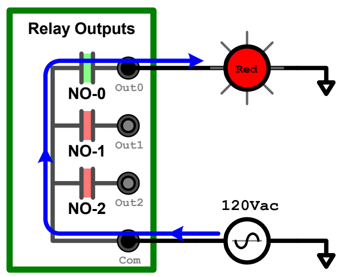


27

Relay Outputs – Sourcing Configuration

Relay Outputs are **discrete outputs** that utilize physical relays, each of which contain a **NO contact** that can be used to make or break any external circuits connected to their terminals.

Thus, when a NO contact is actuated closed, current will flow from the source, into the common terminal, through the closed contact and out of the port, energizing whatever load is connected to that port.

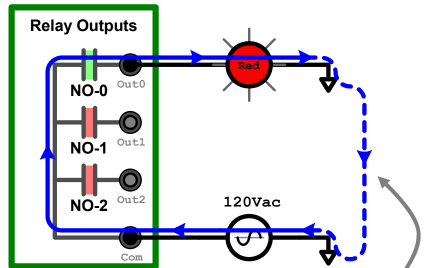


28

Relay Outputs – Sourcing Configuration

Relay Outputs are **discrete outputs** that utilize physical relays, each of which contain a **NO contact** that can be used to make or break any external circuits connected to their terminals.

Thus, when a NO contact is actuated closed, current will flow from the source, into the common terminal, through the closed contact and out of the port, energizing whatever load is connected to that port.



Note that, when multiple nodes are "grounded" within a circuit, they are considered to be connected to a "common ground" that provides a path for current to flow between any of the grounded nodes. The conductive paths between the commonly-grounded nodes are typically not shown in the figures.

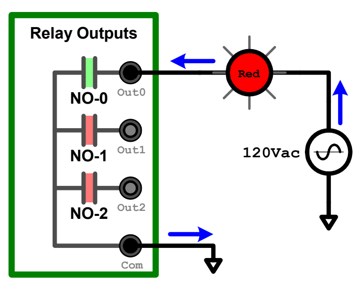
29

Relay Outputs – Sinking Configuration

Relay Outputs are **discrete outputs** that utilize physical relays, each of which contain a **NO contact** that can be used to make or break any external circuits connected to their terminals.

A **sinking output** is an output that "sinks" or pulls current **into** the port from the load.

For this type of output, the **common terminal** is **grounded,** and a source is connected directly to the load.



Sinking outputs allow for the simultaneous control of various loads that utilize different supply voltages provided that the different voltage sources can all be connected to a common ground.

30

Solid-State Discrete Outputs

Solid-state outputs function similar to relay-type outputs, but instead utilize **transistors** (DC) or **triacs** (AC) to provide the switching function necessary to control the operation of the connected loads.

Solid-state outputs operate much **faster** than relay-type outputs, but typically have much **lower current ratings**.

Additionally, solid-state outputs can have unique characteristics that need to be considered before choosing them for the certain applications.

31

Output Port Current Limitations

Operational Note – Output ports have **limited current** capabilities and thus are typically restricted to supplying control-type or other lower current devices such as indicator lamps or the field-coil from a relay or contactor.

Technical Specifications - 1769-OW16

Attribute	1769-OW16
Operating voltage range	5…265V AC 5…125V DC
Delay, on	10 ms
Delay, off	10 ms
Current per point, max	2.5 A
Current per module, max	20 A

These are the specifications for the output modules that are connected to the PLCs in the Q-215 lab.

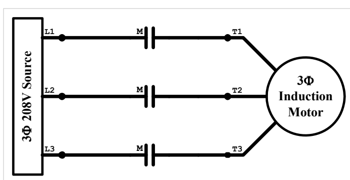
Thus, although an output module can not directly supply a large motor, an output port could be utilized to supply the field coil of a contactor, the main contacts of which could then be used to energize the motor.

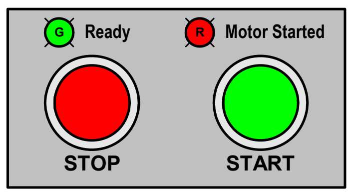
32

PLC-based Motor Controller Example

Design a **Start/Stop motor controller** that (by means of a contactor) energizes a motor when its "**START**" **button** is pressed and de-energizes the motor when its "**STOP**" **button** is pressed.

Include as part of the controller a **green indicator** that illuminates whenever the motor is stopped (Ready) and a **red indicator** that illuminates whenever the motor is operational (Started).





Note that the "power" portion of this system that contains both the main contacts and the motor will not be shown during the remaining parts of this example.

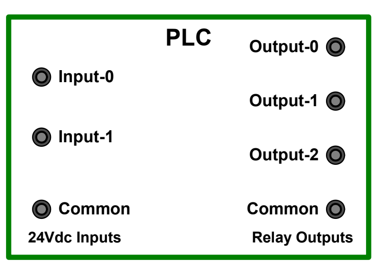
33

PLC-based Motor Controller Example

Since the **PLC** must interface with two input/logic devices (**STOP** and **START buttons**) and three output devices (a **field coil** and **two lights**), we will utilize a PLC that contains at least **two input ports** and **three output ports**.

The input ports will be **$24V_{DC}$ discrete inputs** and the output ports will be **relay-type outputs** that source a **$120V_{AC}$** supply.

Although the PLC would mostly likely contain more than two input ports and three output ports, for simplicity, the unused ports are not shown in the figures.



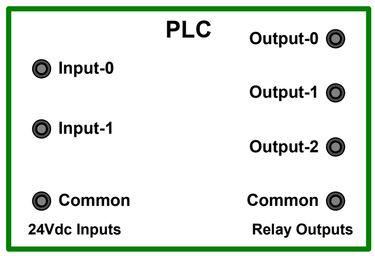
34

PLC-based Motor Controller Example

The **Logic States** of the inputs and outputs are defined as follows:

Input State: **True** ⇨ **+24V**$_{DC}$ detected at terminal
False ⇨ **0V**$_{DC}$ detected at terminal

Output State: **True** ⇨ **120V**$_{AC}$ present at terminal
False ⇨ **0V**$_{AC}$ present at terminal

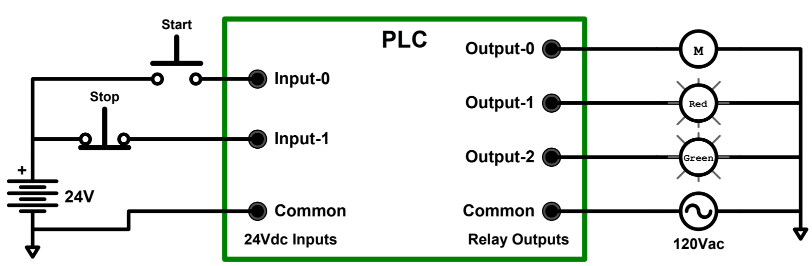


35

PLC-based Motor Controller Example

The **Input and Output Schedules** for the PLC are as follows:

Input	Device	Output	Device
0	NO Start Button	0	Field Coil
1	NC Stop Button	1	Red Lamp (Started)
		2	Green Lamp (Ready)



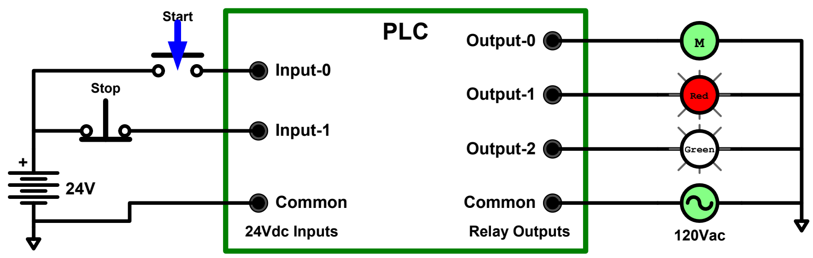
36

PLC-based Motor Controller Example

The **Operational Logic** with respect to **Input-0** is as follows:

Logic: If **Input-0 "True"** ⇨ Set/Hold **Output-0 "True"** (**coil ON**)
(**Start pressed**)
Set/Hold **Output-1 "True"** (**Red ON**)
Set/Hold **Output-2 "False"** (**Green OFF**)

If **Input-0 "False"** ⇨ **No change**.



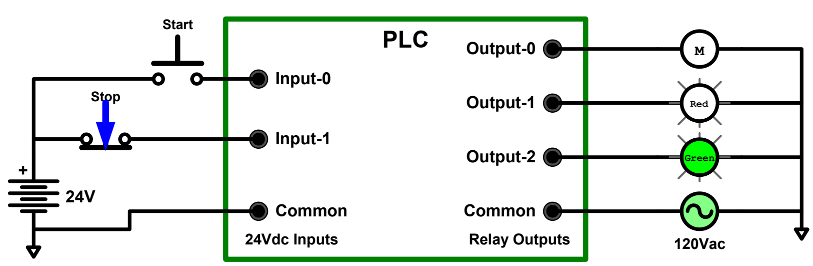
37

PLC-based Motor Controller Example

The **Operational Logic** with respect to **Input-1** is as follows:

Logic: If **Input-1 "False"** ⇨ Set/Hold **Output-0 "False"** (**coil OFF**)
(**Stop pressed**)
Set/Hold **Output-1 "False"** (**Red OFF**)
Set/Hold **Output-2 "True"** (**Green ON**)

If **Input-1 "True"** ⇨ **No change.**



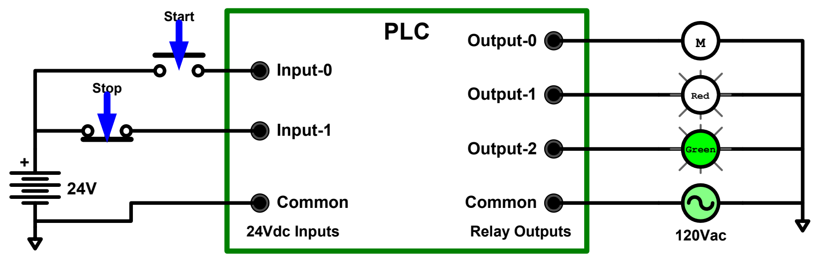
38

PLC-based Motor Controller Example

Other **Operational Logic** to consider:

Logic: If **Input-0 "True"** **<u>and</u>** **Input-1 "False"** **(<u>both</u> pressed)**

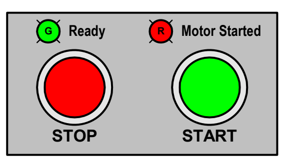
⇨ Set/Hold **Output-0 "False"** **(coil OFF)**
Set/Hold **Output-1 "False"** **(Red OFF)**
Set/Hold **Output-2 "True"** **(Green ON)**

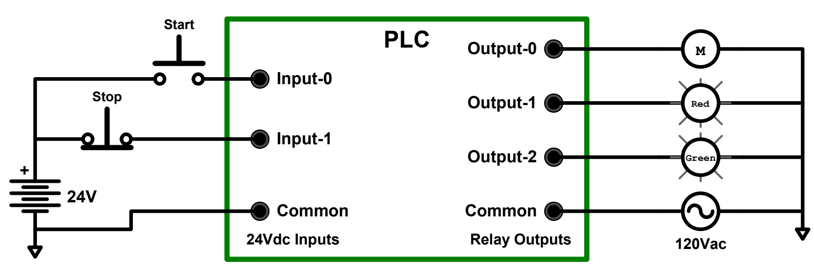


39

PLC-based Motor Controller Example

Note that the **programming** concepts required to implement the operational logic for the system are beyond the scope of this presentation.





40