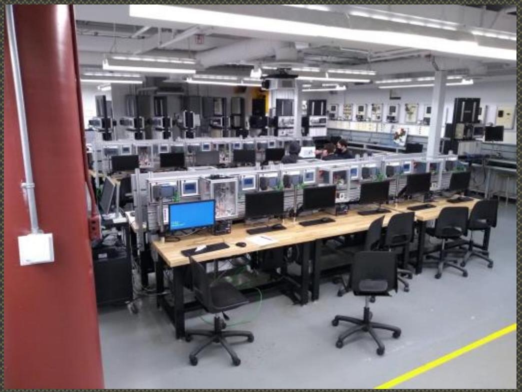


PROGRAMMABLE LOGIC CONTROLLERS
MENG 3500

HUMBER

ControlLogix Controllers

- Automation companies have introduced a new class of industrial controllers known as programmable automation controllers (PACs).
- PAC combines PLC ruggedness with PC functionality.
- These devices resemble PLCs in physical appearance but integrate advanced features such as communication, data logging, signal processing, motion control, process control, and machine vision into a unified programming environment.
- A PLC mixes scan-based and event-driven program execution, whereas PAC software is typically event-driven.

Figure 15-1 Programmable automation controllers (PACs).
Source: Image Courtesy of Rockwell Automation, Inc.

HUMBER

(Frank D. Petruzella Programmable Logic Controllers 6th edition

ControlLogix Controllers

- PLC (Scan-Based + Event-Driven) Example: A conveyor system in a factory moves products and stops when a sensor detects an object. The PLC continuously scans input sensors, checks conditions, and controls the motor.
- PAC (Event-Driven) Example: A smart manufacturing system adjusts machine parameters based on real-time data from multiple sensors. Instead of scanning all logic repeatedly, the PAC processes events like temperature changes or remote commands to optimize performance dynamically.
- In short: PLCs are ideal for simple, repetitive control, while PACs handle complex, data-driven automation.

Figure 15-1 Programmable automation controllers (PACs).
Source: Image Courtesy of Rockwell Automation, Inc.

HUMBER

(Frank D. Petruzella Programmable Logic Controllers 6th edition

ControlLogix Controllers
PAC vs PLC

- PLCs are traditionally designed for discrete control applications, excelling in tasks that involve on/off control logic.
  - They are well-suited for applications in manufacturing, where precise timing and sequencing are crucial.
- On the other hand, PACs offer a more comprehensive approach by integrating both discrete and process control functionalities.
  - PACs are equipped with a broader range of capabilities, supporting not only sequential and logic control but also complex data handling and communication in continuous and batch processes.
  - Batch processes, such as pharmaceutical, food production, and chemical processing are examples of precise control systems that play a pivotal role in ensuring quality and consistency in the production of discrete units or batches.
    - Batch processes are involved in mixing specific quantities of raw materials, chemical reactions, and precise control over temperature and pressure to ensure the quality and consistency of the final product.

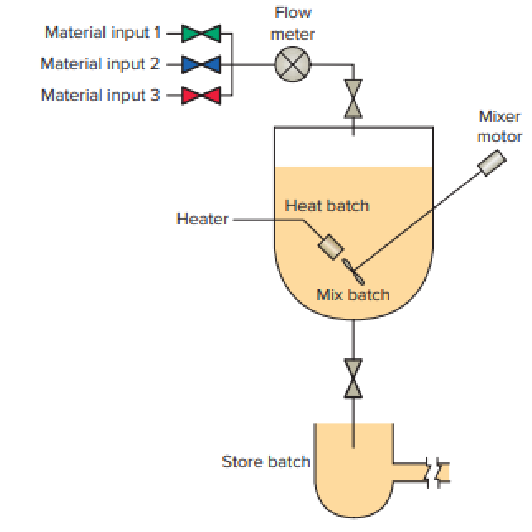


Figure 14-2 Batch process.

HUMBER

(Frank D. Petruzella Programmable Logic Controllers 6th edition

ControlLogix Controllers
Size and Application

- When classifying PLCs, we consider what they can do, how many inputs and outputs they have, their cost, and size, with the number of inputs and outputs being the most crucial factor.
- There are three major types of PLC application: single-ended, multitask, and control management.
  - A single-ended or stand-alone PLC application means one PLC controls one process, operating independently without communication with other computers or PLCs.
  - Multitask PLC application means one PLC handles multiple processes, needing enough I/O capacity. If it's part of a bigger system and needs to talk to a central PLC or computer, a data communications network is essential.
  - In a control management PLC application, one big PLC oversees others, needing a sizable processor for communication with other PLCs and maybe a computer. It supervises by sending programs to instruct other PLCs, requiring the ability to connect and communicate with any of them through proper addressing.

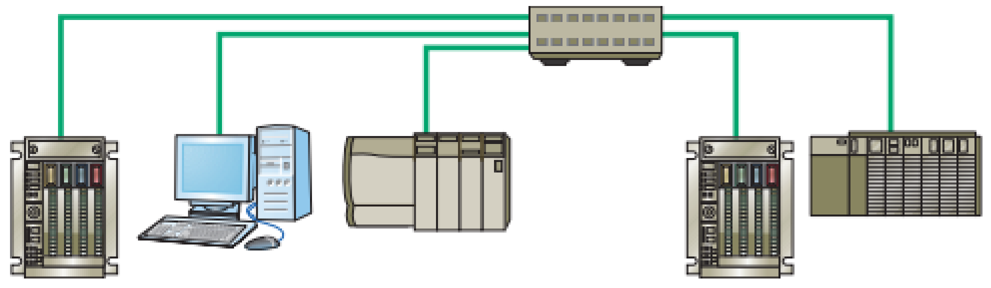


Figure 1-23 Control management PLC application.

(Frank D. Petruzella Programmable Logic Controllers 6th edition

ControlLogix Controllers
Memory

- PLC memory stores data, instructions, and the control program.
- Memory size is usually expressed in K or M values: 1 K, 16 K, 8 M, 16 M, etc.
  - 1 K means 1024, measured in the binary number system ($2^{10} = 1024$).
- A PLC's instructions set presents the various type of supported instructions.
- Memory needs in a PLC vary based on the application. Factors influencing the required memory size include:
  - Number of I/O points
  - Size of control program
  - Data-collecting requirements
  - Supervisory functions required
  - Future expansion

Table 1-1 Typical PLC Instructions

Instruction	Operation
XIC (Examine ON)	Examine a bit for an ON or 1 condition
XIO (Examine OFF)	Examine a bit for an OFF or 0 condition
OTE (Output Energize)	Turn ON a bit (nonretentive)
OTL (Output Latch)	Latch a bit (retentive)
OTU (Output Unlatch)	Unlatch a bit (retentive)
TOF (Timer Off-Delay)	Turn an output ON or OFF after its rung has been OFF for a preset time interval
TON (Timer On-Delay)	Turn an output ON or OFF after its rung has been ON for a preset time interval
CTD (Count Down)	Use a software counter to count down from a specified value
CTU (Count Up)	Use a software counter to count up to a specified value

HUMBER

(Frank D. Petruzella Programmable Logic Controllers 6th edition

ControlLogix Controllers Configuration

- In a ControlLogix controller, there is no set memory for specific data or I/O.
- Users configure the internal memory layout using the programming software during project creation.
- The software needs hardware details like the processor and I/O modules.
- The communication software is used to set up a communications link between the programming software and the PLC hardware.

Figure 15-2 RSLogix 5000 screen.
Source: Image Courtesy of Rockwell Automation, Inc.

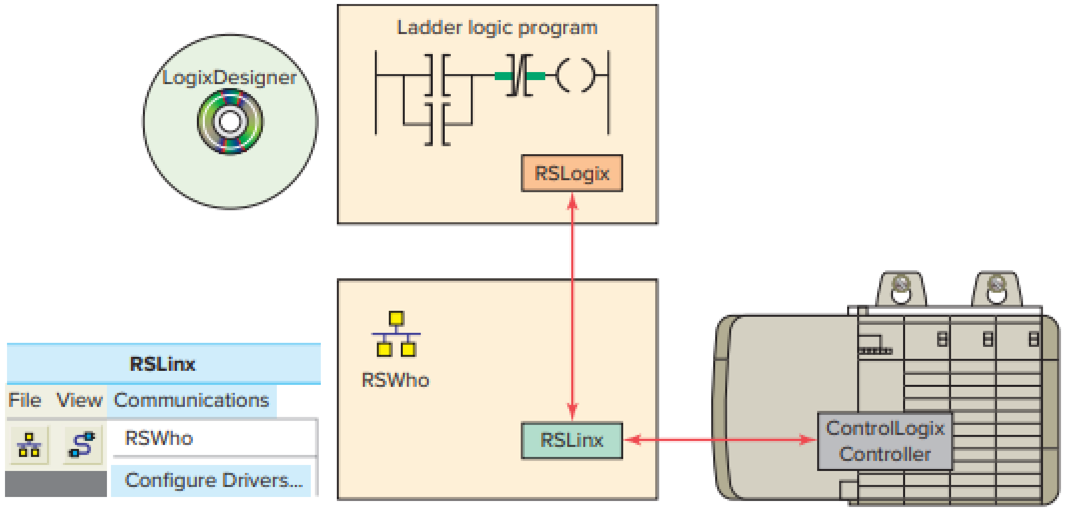


Figure 15-3 RSLinx and LogixDesigner software.

HUMBER

(Frank D. Petruzella Programmable Logic Controllers 6th edition

ControlLogix Controllers
Project

- RSLogix software stores a controller's programming and configuration information in a file called a project.
- The main components of the project file are tasks, programs, and routines.
- A controller can hold and execute only one project at a time.
- Global vs Local Data, System Shared Data (Produced & Consumed tags)

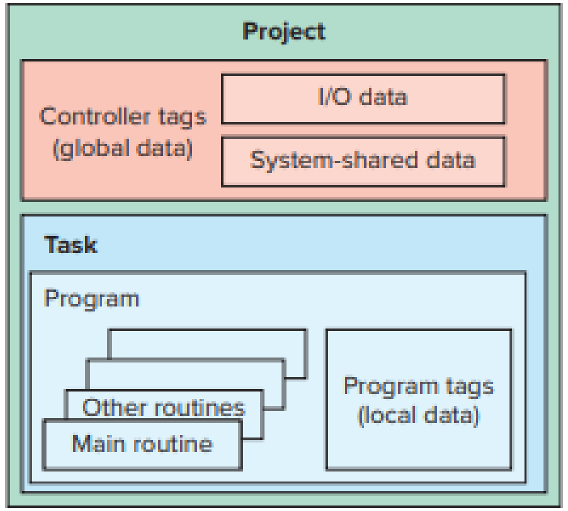


Figure 15-5 ControlLogix processor program file.

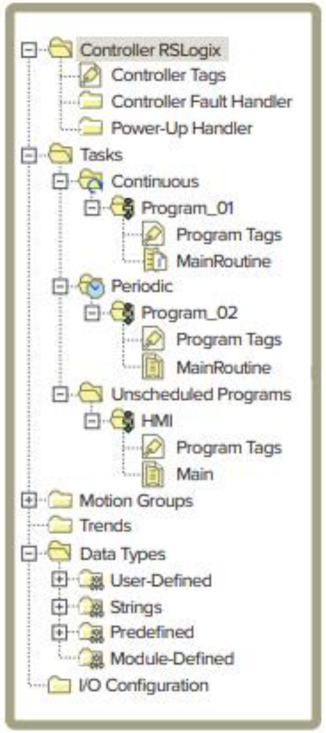


Figure 15-6 Controller organizer tree.

(Frank D. Petruzella Programmable Logic Controllers 6th edition

ControlLogix Controllers
Tasks

- Tasks, the first level of scheduling in a project, are groups of scheduled programs.
- When a task runs, its programs execute in order.
- Only one task runs at a time. The number of tasks depends on the specific controller.
- The main types of tasks include:
  - Continuous tasks execute nonstop but are always interrupted by a periodic task.
    - Lowest priority.
    - Named Main Task.
  - Periodic tasks act like timed interrupts.
    - They interrupt the continuous task, running for a fixed duration at specific intervals.
  - Event tasks also act as interrupts.
    - It differs from timed interrupts as it's triggered by an occurrence or failure to occur.

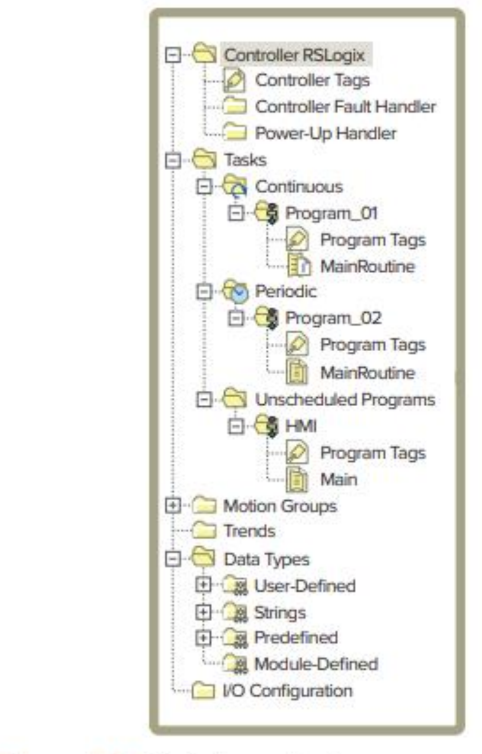


Figure 15-6 Controller organizer tree.

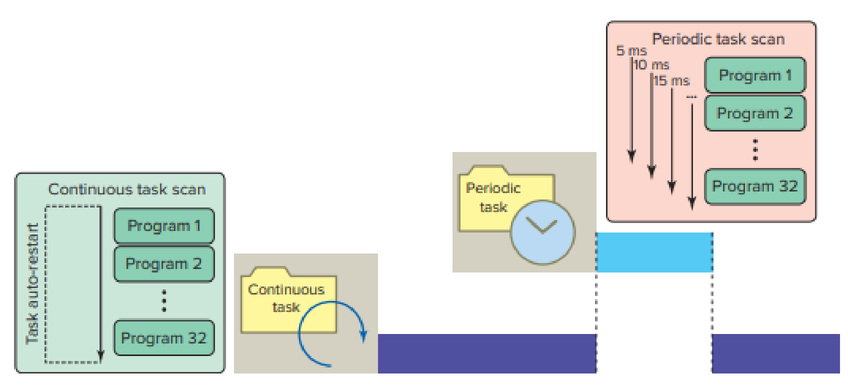


Figure 15-7 Continuous and periodic tasks.

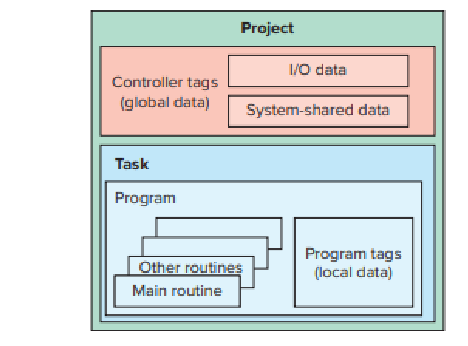


Figure 15-5 ControlLogix processor program file.

(Frank D. Petruzella Programmable Logic Controllers 6th edition

HUMBER

ControlLogix Controllers
Tasks

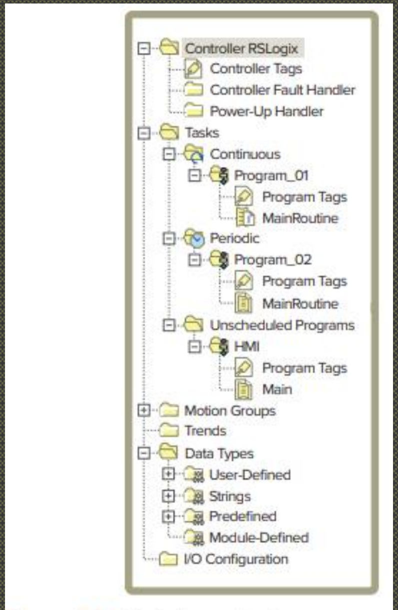
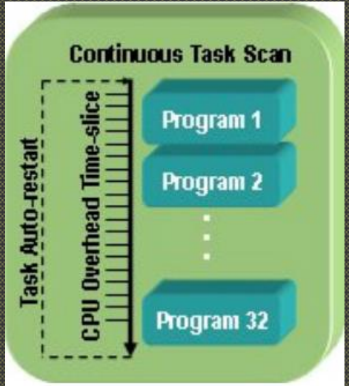
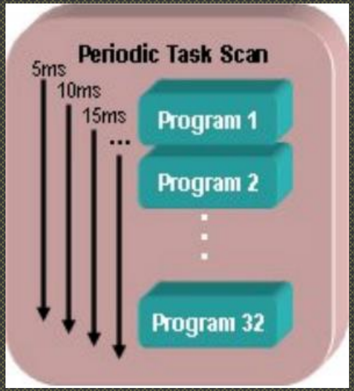


Figure 15-6 Controller organizer tree.

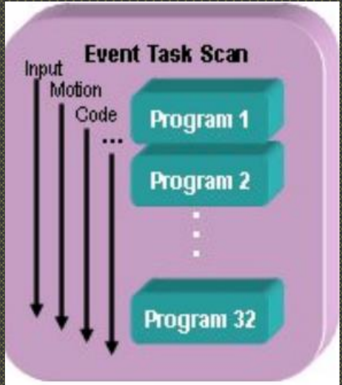
In a continuous task, the controller scans the code top to bottom and then repeats the process.

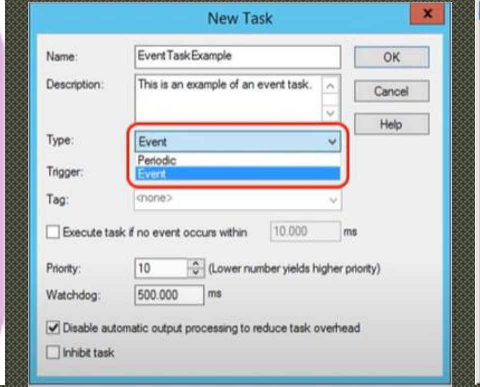


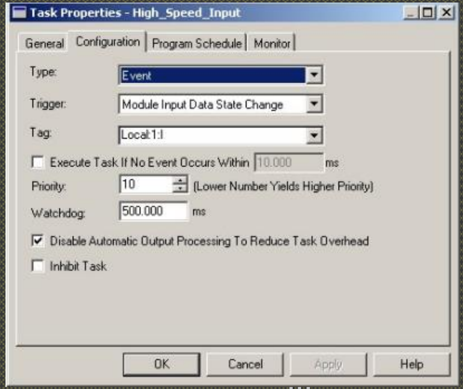
A periodic task runs top to bottom like a continuous task but waits a set time before restarting.



An event task runs only when triggered, interrupting lower-priority tasks, executing once, and then resuming the previous task.







(Frank D. Petruzella Programmable Logic Controllers 6th edition

https://literature.rockwellautomation.com/idc/groups/literature/documents/wp/logix-wp003_-en-p.pdf

ControlLogix Controllers
Programs - Routines

- Programs are the second level of scheduling within a project.
- The folders under Main Task specify the order in which programs should execute.
- Programs that are not assigned to a task are unscheduled.
  - Unscheduled programs, once downloaded to the controller, stay inactive until they are needed.

- Routines, at the third scheduling level, provide the project's executable code.
- Each routine contains a set of logic elements for a specific programming language.
  - Main routine and subroutines.

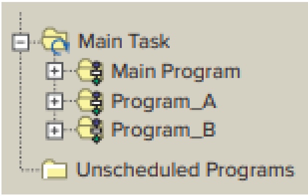


Figure 15-8 Order of execution of programs.

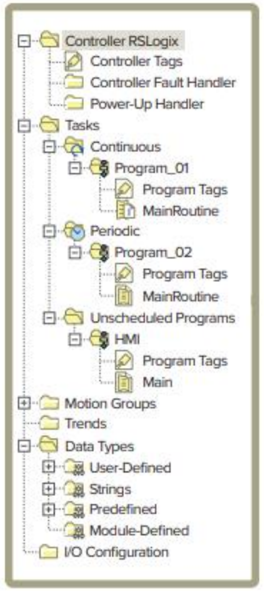


Figure 15-6 Controller organizer tree.

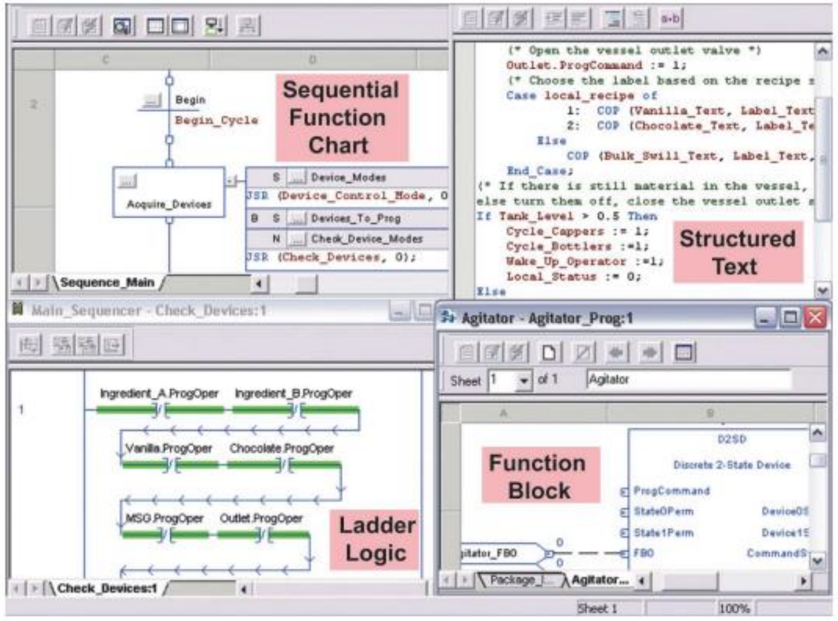
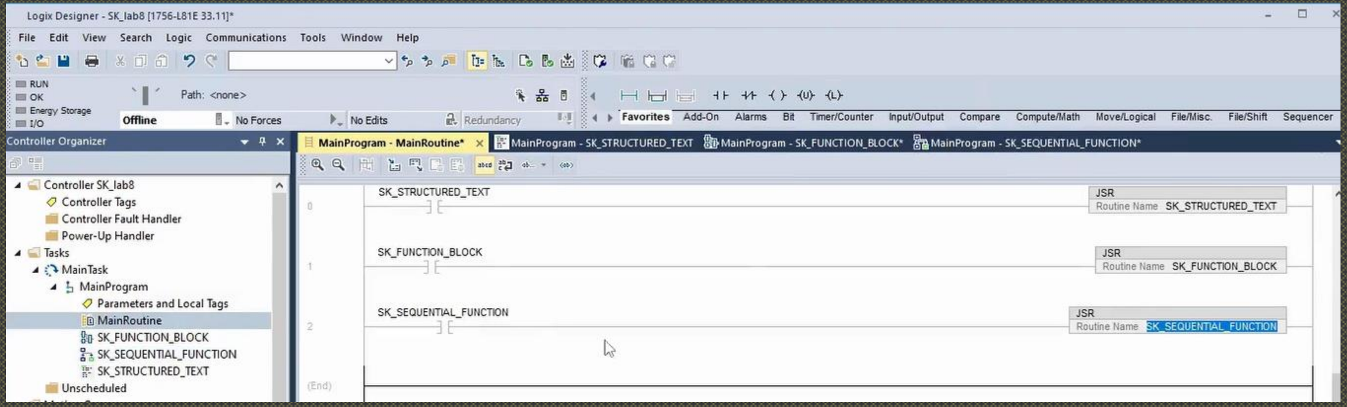


Figure 15-9 Each routine contains a set of logic elements for a specific programming language.
Source: Image Courtesy of Rockwell Automation, Inc.



HUMBER

(Frank D. Petruzella Programmable Logic Controllers 6th edition

ControlLogix Controllers Tags

- ControlLogix uses a tag-based addressing structure.
- Tags are names that describe a specific application.
- A tag represents and identifies data locations in the controller's memory.
- To group data, create an array – a collection of similar tags.
- Scope determines which programs can access a tag.
  - A program tag contains data accessible only by routines within its specific program (local data).
    - Routines in other programs can't access program-scoped tags from another program.
  - A controller tag contains data accessible by all routines within a controller (global data). I/O tags are automatically created as controller scoped tags

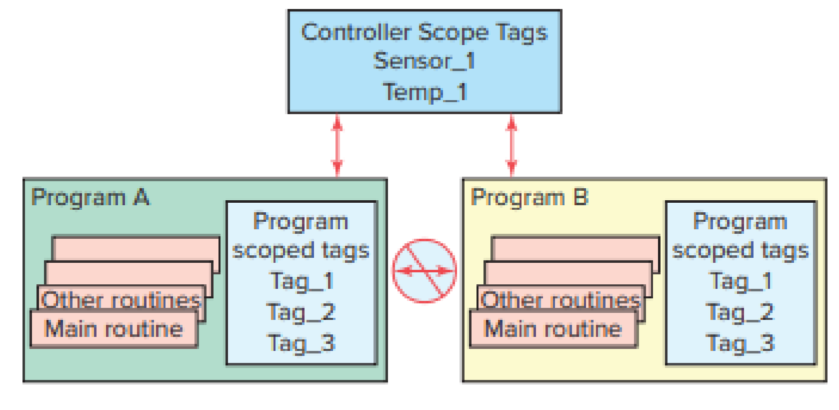


Figure 15-11 Program scoped and controller scoped tags.

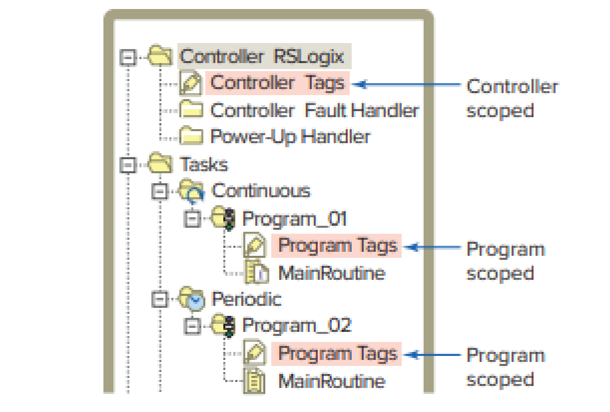


Figure 15-12 Listing of program and controller scoped tags.

HUMBER

(Frank D. Petruzella Programmable Logic Controllers 6th edition

ControlLogix Controllers Tags

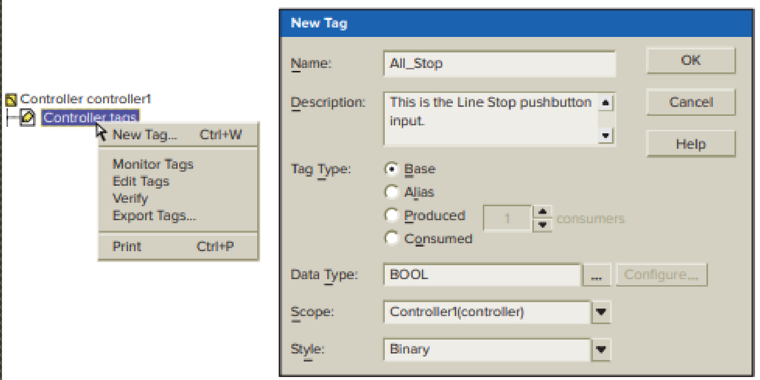


Figure 15-21 Controller scope base tag.
Source: Image Courtesy of Rockwell Automation, Inc.

- There are four different tag types: base, alias, produced, and consumed tags.
  - A base tag stores different data types for project logic use.
  - An alias tag provides an alternate name for a tag.
    - The alias tag is used to create a tag name for a physical input or output.
  - Produced/consumed tags share information between devices on a network.
    - A produced tag sends data while a consumed tag receives data.
    - Produced tags are always controller scoped.

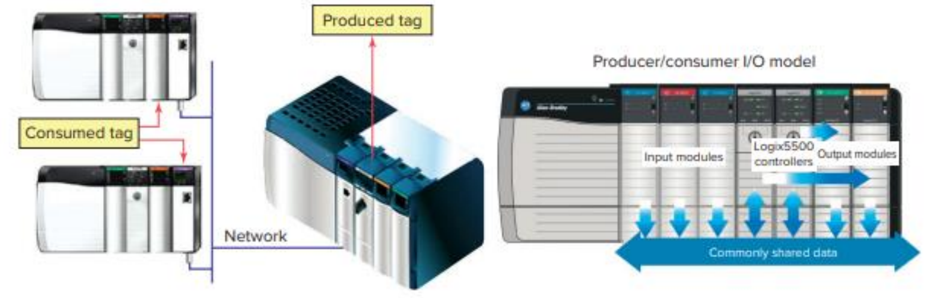


Figure 15-15 Produced/consumed tags used to share information.
Source: Image Courtesy of Rockwell Automation, Inc.

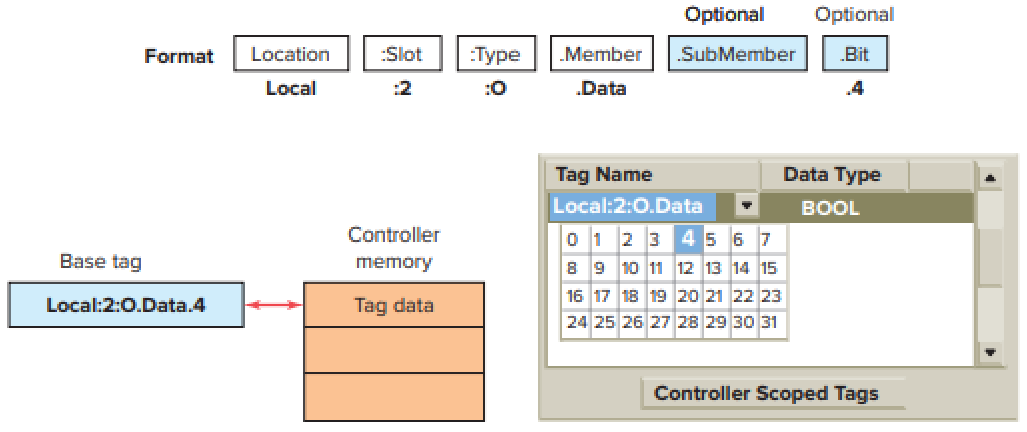


Figure 15-13 Base tag.

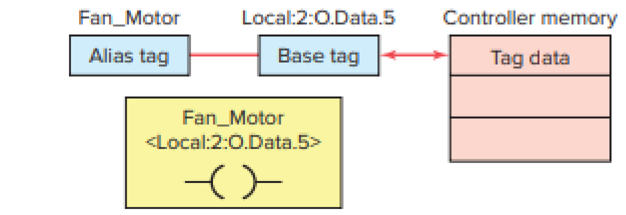


Figure 15-14 Alias tag linked to a base tag.

(Frank D. Petruzella Programmable Logic Controllers 6th edition

ControlLogix Controllers
Tags

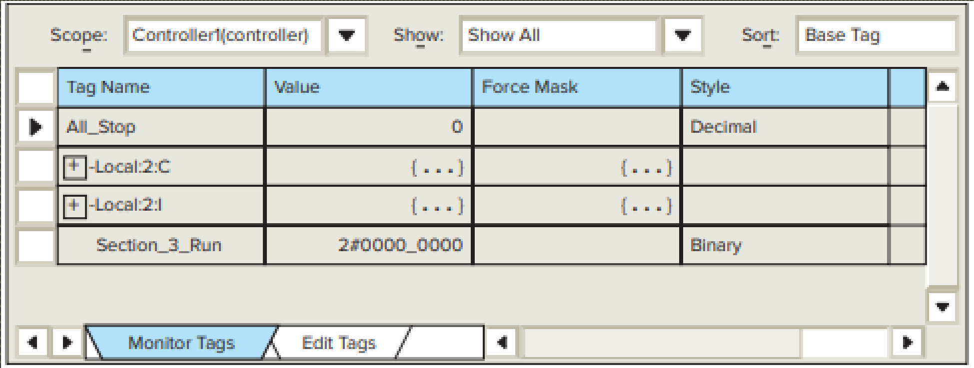


Figure 15-22 Monitor Tags window.

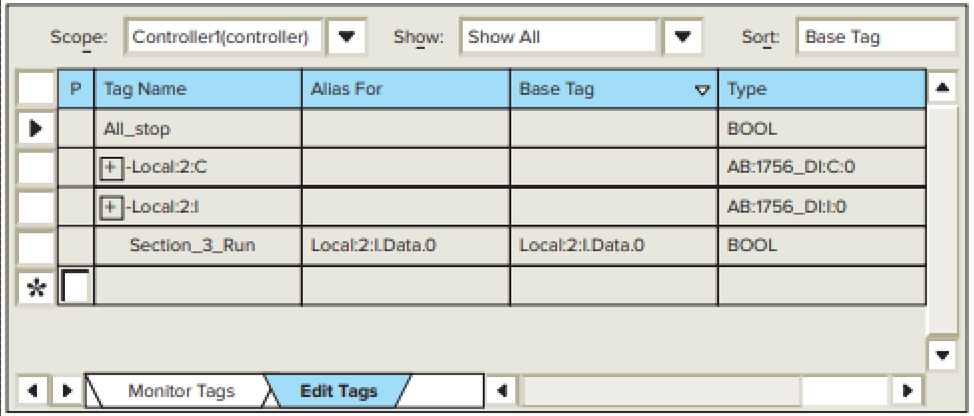


Figure 15-23 Edit Tags window.

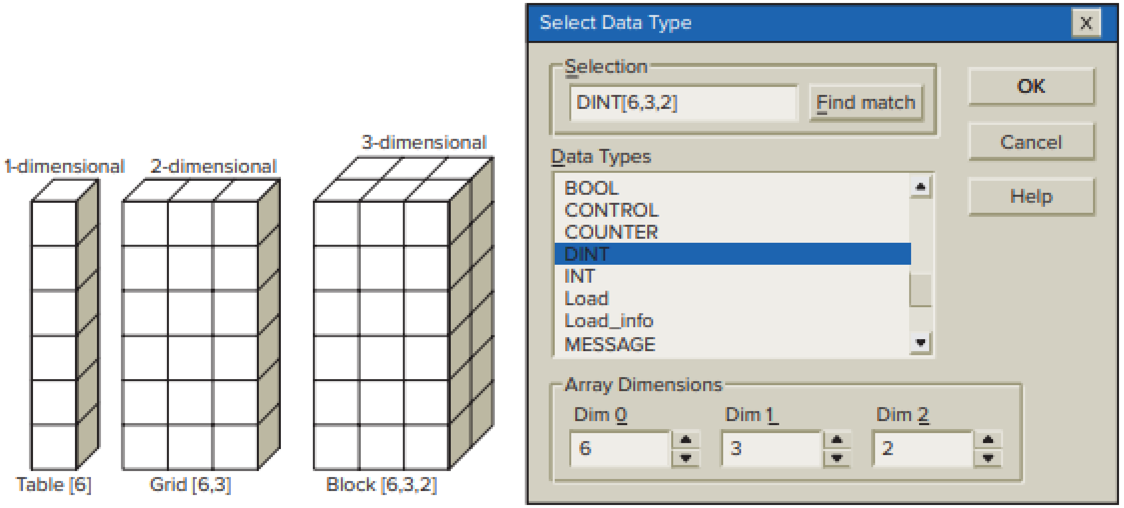


Figure 15-24 Types of arrays.
Source: Image Courtesy of Rockwell Automation, Inc.

Array - Temp
Data Type - INT[5]

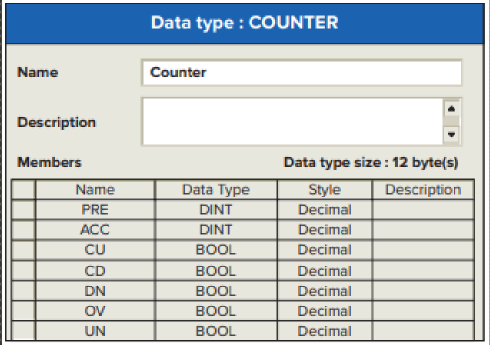
Temp[0]	297
Temp[1]	200
Temp[2]	180
Temp[3]	120
Temp[4]	100

Figure 15-25 Memory layout for a one-dimensional array.

(Frank D. Petruzella Programmable Logic Controllers 6th edition

ControlLogix Controllers
Structures – Additional Data Type

- Logix controllers are based on 32-bit operations. The types of data that can be a base tag are BOOL, SINT, INT, DINT, and REAL.
- In a ControlLogix controller, there are three structure types: predefined, module-defined, and user-defined.
  - The controller automatically creates predefined structures, including timers, counters, messages, and PID instructions.
  - Module-defined structures are automatically created when configuring I/O modules for the system.
  - A user-defined structure complements predefined structures, allowing the creation of custom structures to store and handle data as a group.



Data type : COUNTER

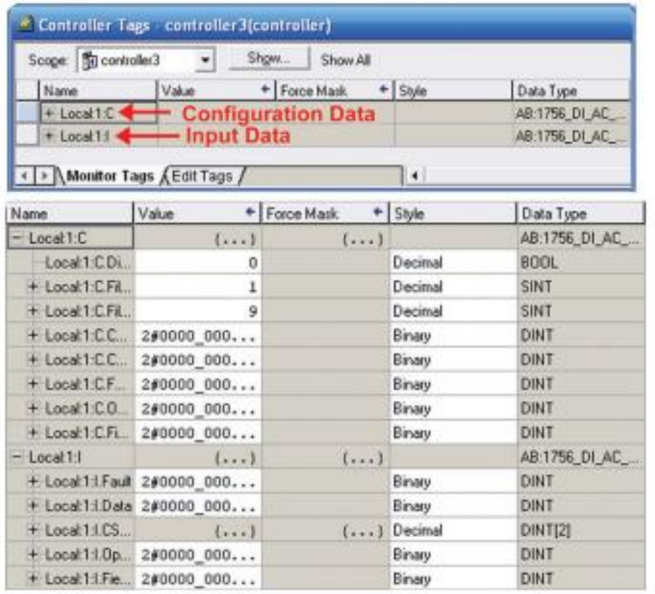
Name: Counter

Description:

Members — Data type size : 12 byte(s)

Name	Data Type	Style	Description
PRE	DINT	Decimal	
ACC	DINT	Decimal	
CU	BOOL	Decimal	
CD	BOOL	Decimal	
DN	BOOL	Decimal	
OV	BOOL	Decimal	
UN	BOOL	Decimal	

Figure 15-18 Predefined structure.



Controller Tags - controller3(controller)

Scope: controller3 Show... Show All

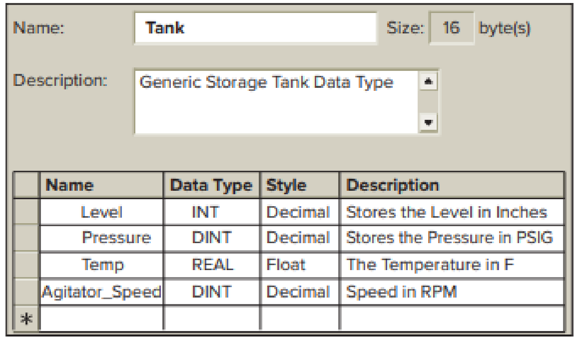
Name	Value	Force Mask	Style	Data Type
+ Local:1:C ← Configuration Data				AB:1756_DI_AC_...
+ Local:1:I ← Input Data				AB:1756_DI_AC_...

Monitor Tags / Edit Tags

Name	Value	Force Mask	Style	Data Type
− Local:1:C	{...}	{...}		AB:1756_DI_AC_...
Local:1:C.Di...	0		Decimal	BOOL
+ Local:1:C.Fil...	1		Decimal	SINT
+ Local:1:C.Fil...	9		Decimal	SINT
+ Local:1:C.C...	2#0000_000...		Binary	DINT
+ Local:1:C.C...	2#0000_000...		Binary	DINT
+ Local:1:C.F...	2#0000_000...		Binary	DINT
+ Local:1:C.O...	2#0000_000...		Binary	DINT
+ Local:1:C.Fi...	2#0000_000...		Binary	DINT
− Local:1:I	{...}	{...}		AB:1756_DI_AC_...
+ Local:1:I.Fault	2#0000_000...		Binary	DINT
+ Local:1:I.Data	2#0000_000...		Binary	DINT
+ Local:1:I.CS...	{...}	{...}	Decimal	DINT[2]
+ Local:1:I.Op...	2#0000_000...		Binary	DINT
+ Local:1:I.Fie...	2#0000_000...		Binary	DINT

Figure 15-19 Module-defined structure for a digital input module.

Source: Image Courtesy of Rockwell Automation, Inc.



Name: Tank Size: 16 byte(s)

Description: Generic Storage Tank Data Type

Name	Data Type	Style	Description
Level	INT	Decimal	Stores the Level in Inches
Pressure	DINT	Decimal	Stores the Pressure in PSIG
Temp	REAL	Float	The Temperature in F
Agitator_Speed	DINT	Decimal	Speed in RPM
*			

Figure 15-20 User-defined storage tank structure.

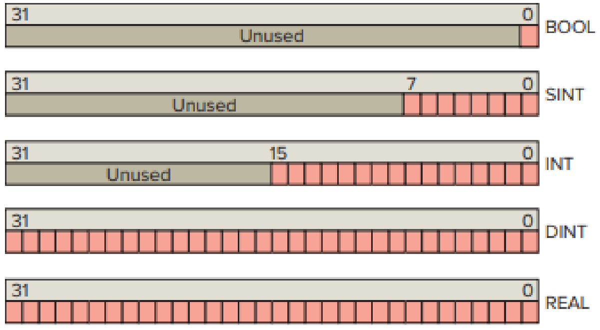


Figure 15-16 Types of base tag data.

HUMBER

(Frank D. Petruzella Programmable Logic Controllers 6th edition

ControlLogix Controllers Addresses

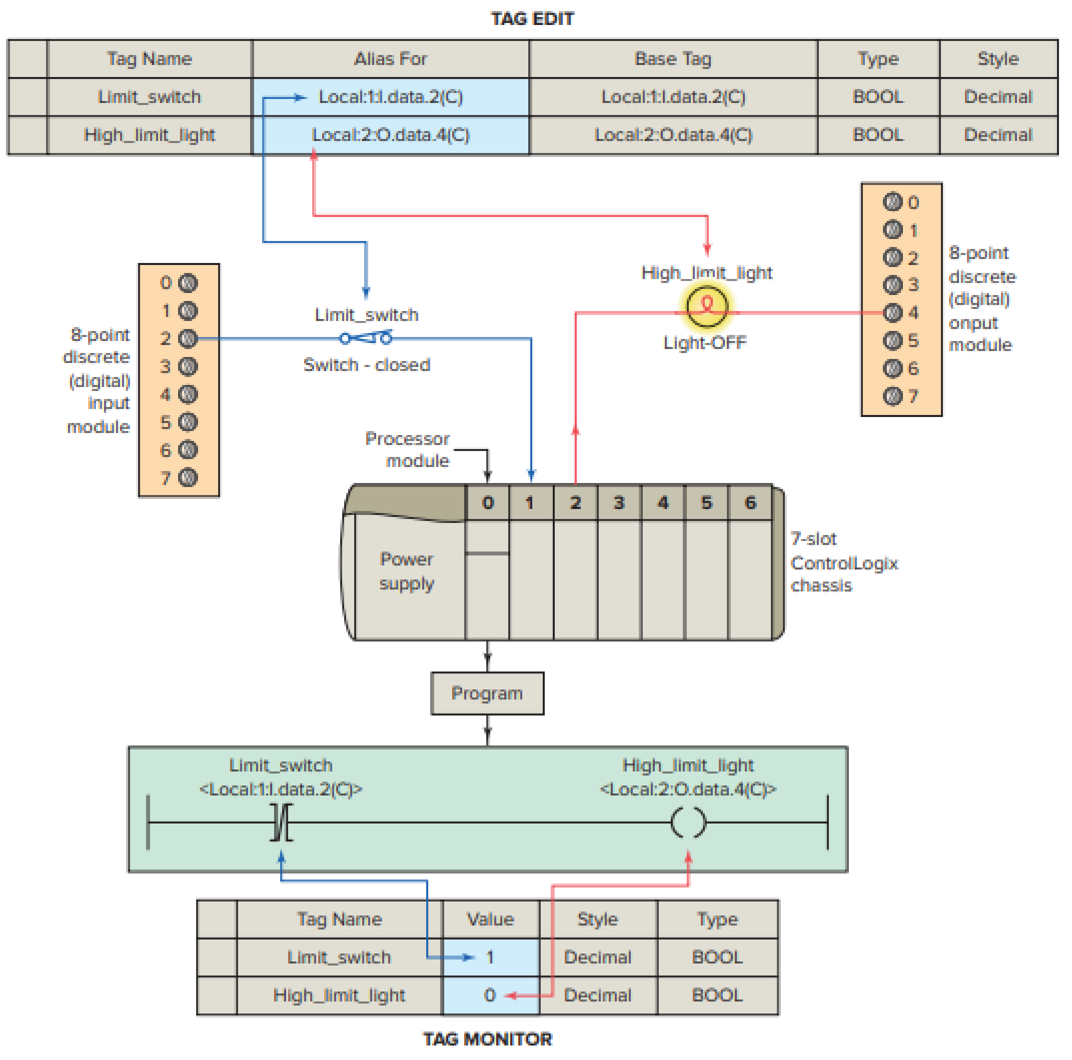


Figure 15-30 Tag-based address implementation.

(Frank D. Petruzella Programmable Logic Controllers 6th edition

ControlLogix Controllers Program – Motor Control

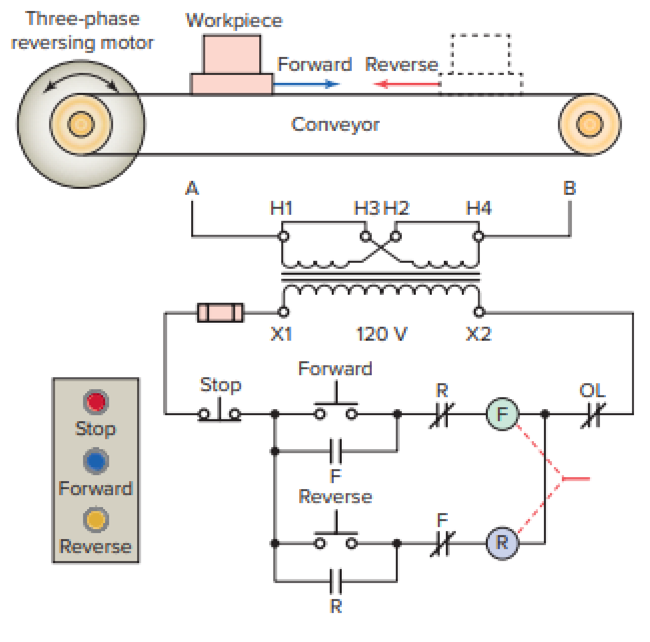


Figure 15-36 Hardwired reversing conveyor motor control circuit.

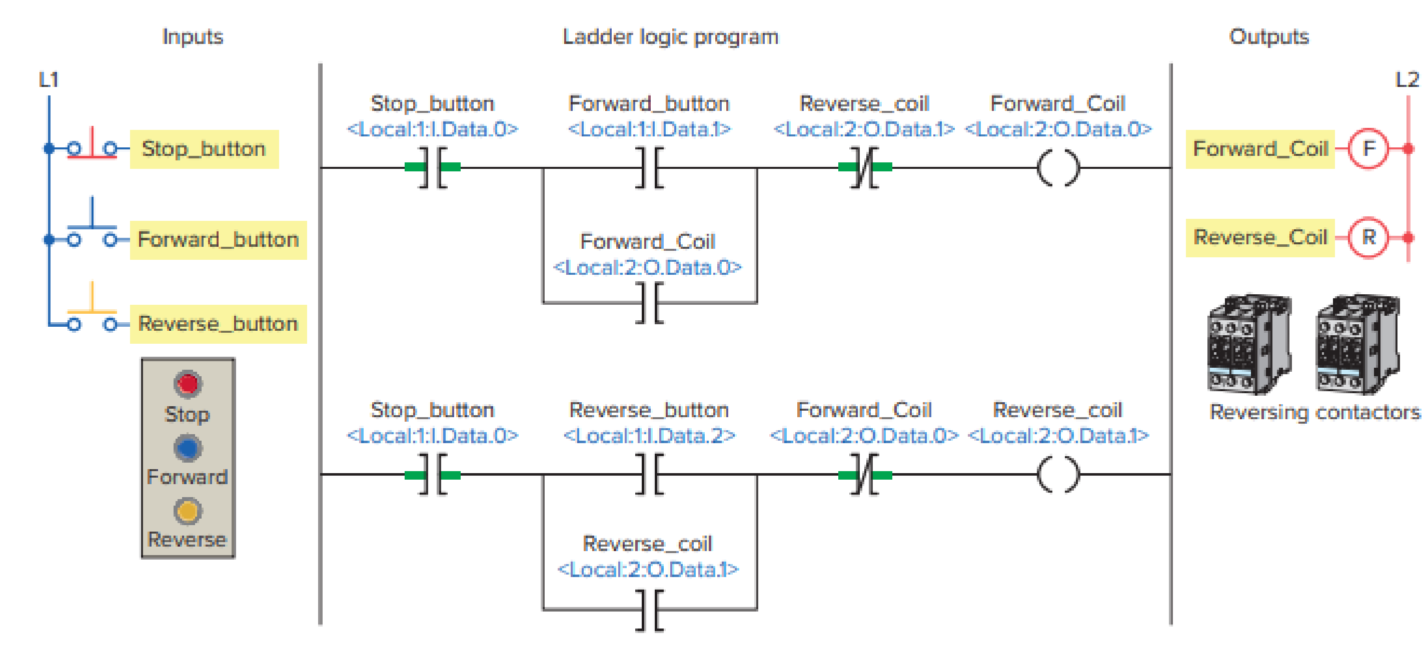


Figure 15-37 Programmed reversing conveyor motor control circuit.

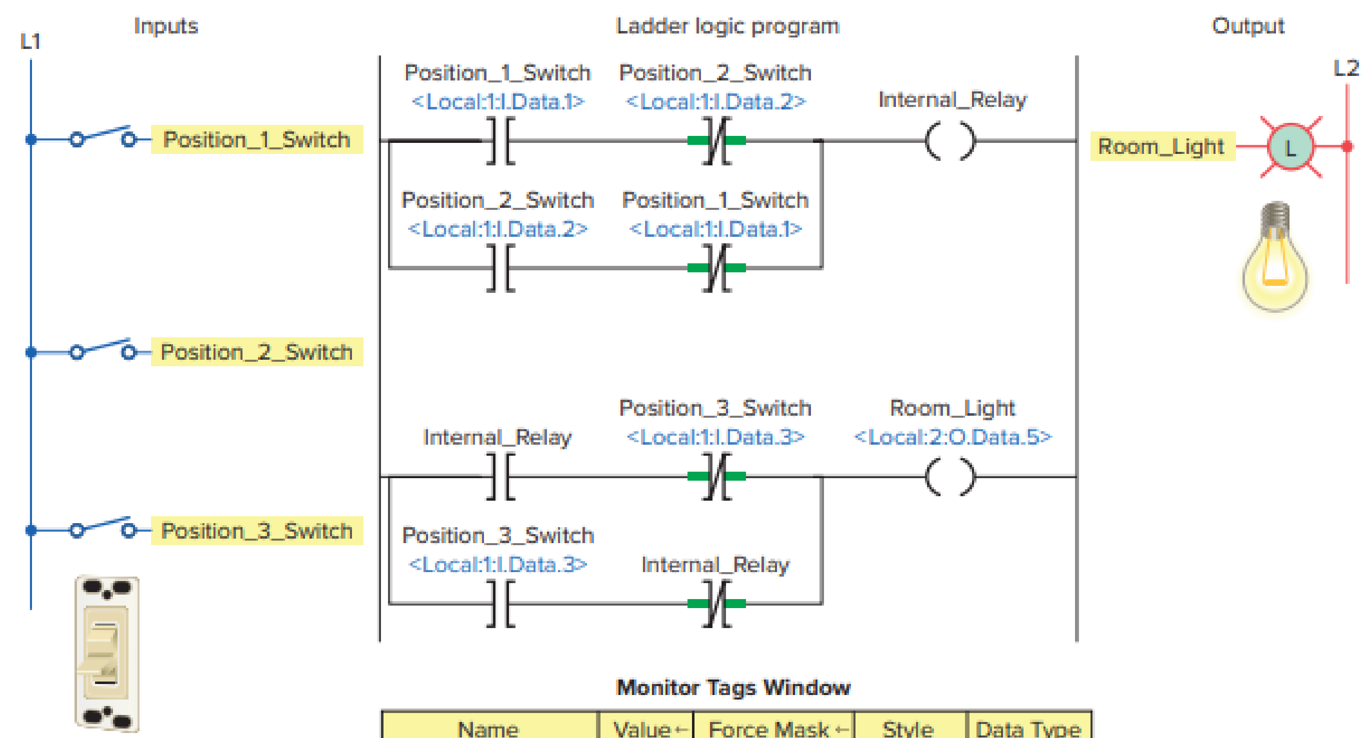
To reverse the motor using this control circuit, the operator needs to press the stop button first. This action de-energizes the coil and closes the normally closed contact again.

HUMBER

(Frank D. Petruzella Programmable Logic Controllers 6th edition

ControlLogix Controllers
Program – Light Control

Discuss the program that uses an internal relay to implement on/off control of a room light from three different entrances.



Monitor Tags Window

Name	Value ←	Force Mask ←	Style	Data Type
Internal_Relay	0		Decimal	BOOL
Position_1_Sw...	0		Decimal	BOOL
Position_2_Sw...	0		Decimal	BOOL
Position_3_Sw...	0		Decimal	BOOL
Room_Light	0		Decimal	BOOL

Figure 15-38 Internal relay to implement on/off control of a room light from three different entrances.

(Frank D. Petruzella Programmable Logic Controllers 6th edition

ControlLogix Controllers Program – Car Wash

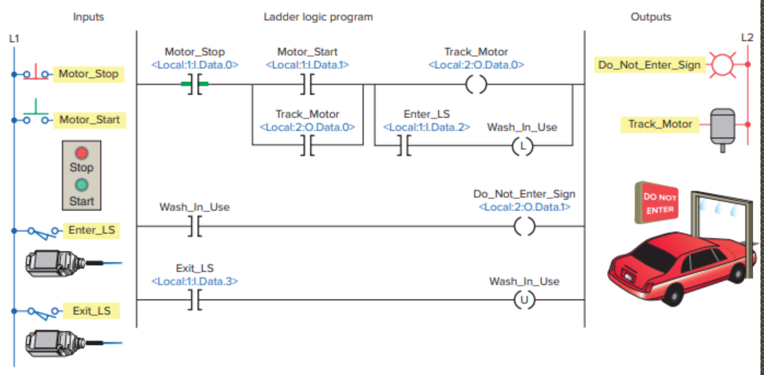


Figure 15-42 Latch/unlatch instruction used as part of a car wash program.

The operation of the program can be summarized as follows:

- The car wash only washes one car at a time.
- The operator controls the ON/OFF operation of the track motor with the track motor Stop and Start PB.
- When the Motor_Start button is closed the Track_Motor is energized and performs a seal-in function that keeps the motor operating when the Motor_Start button is released.
- As the car enters the car wash, the Enter_LS contact momentarily closes to energize the Wash_In_Use latch instruction.
- This, in turn, energizes the Do_Not_Enter_Sign to indicate that the car wash is in use.
- When the car exits the car wash, the Exit_LS contact momentarily closes and energizes the Wash_In_Use unlatch instruction.
- This, in turn, de-energizes the Do_Not_Enter_Sign to indicate the car wash is not in use.
- The track motor remains running ready to restart the process once another car enters.
- Once running, the track motor can be stopped at any time by operating the Motor_Stop button to de-energize the Track_Motor.

HUMBER

(Frank D. Petruzella Programmable Logic Controllers 6th edition

ControlLogix Controllers Program – Car Wash

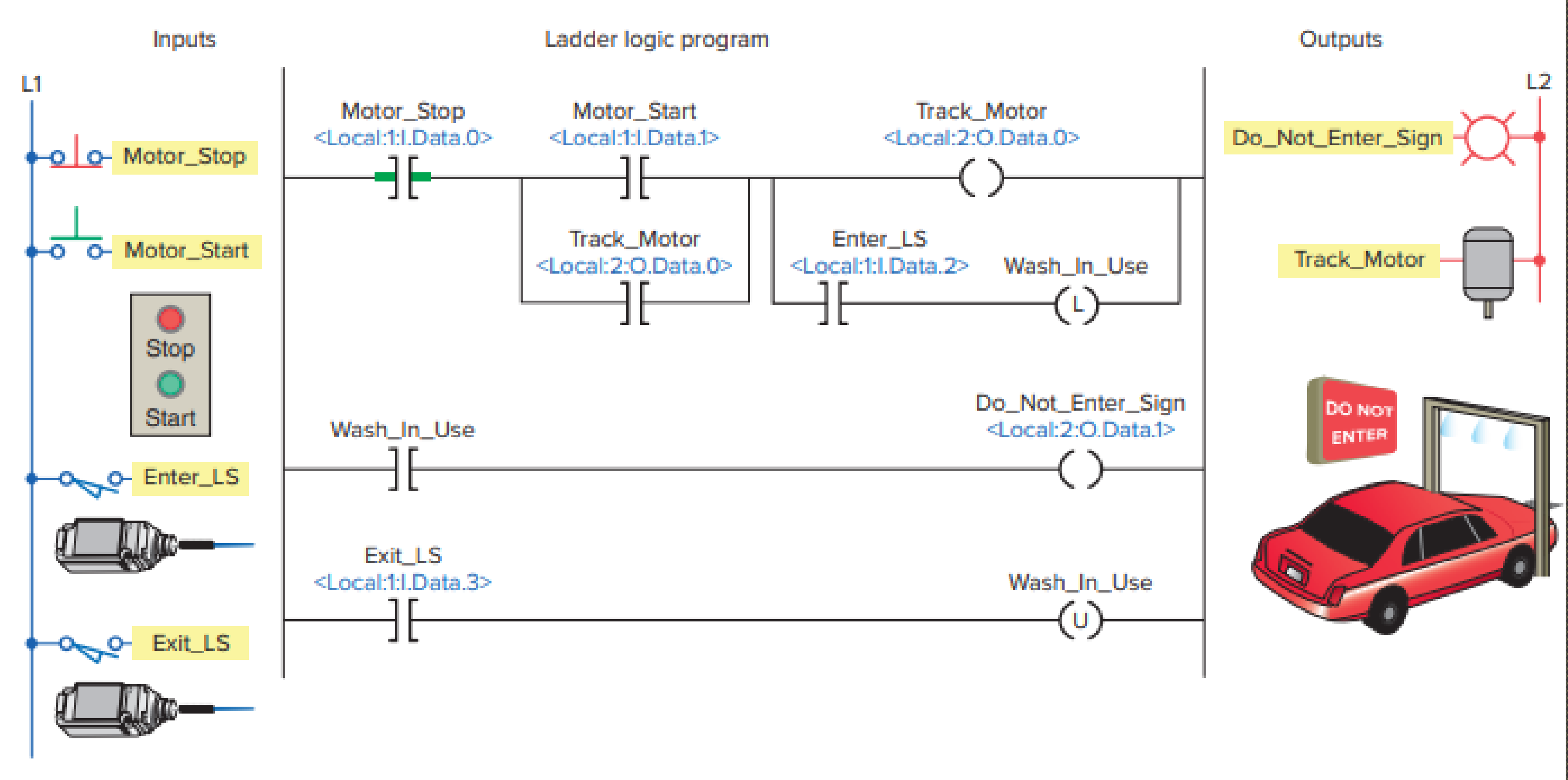


Figure 15-42 Latch/unlatch instruction used as part of a car wash program.

HUMBER

(Frank D. Petruzella Programmable Logic Controllers 6th edition

ControlLogix Controllers
Program – Conveyor Speed Control

The system comes equipped with a weight detector and three-speed motor controller.
It is designed to move the conveyor belt at a certain speed when a specific value of weight is on the conveyor.

- If the weight exceeds the preset value, the conveyor speed increases to compensate for the increase in weight.
- If the weight falls below the preset value, the conveyor speed is reduced to compensate for the decrease in weight.

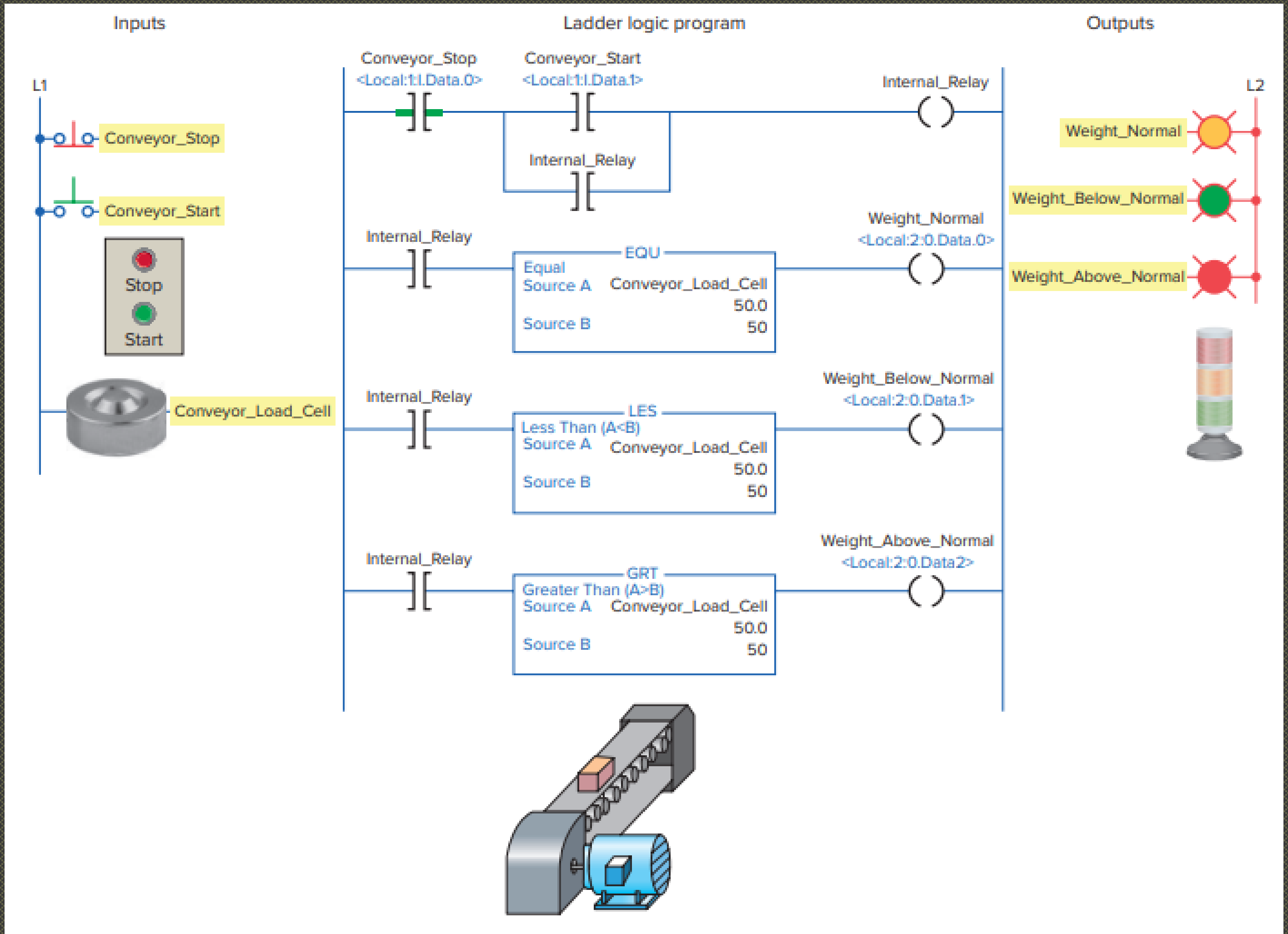


Figure 15-93 PLC program for three-speed control of a conveyor system.

(Frank D. Petruzella Programmable Logic Controllers 6th edition

Analog Modules

- Analog input and output modules are used for continuous control processes, as opposed to digital ON/OFF types which are discrete.
- Analog input devices like temperature sensors, potentiometers, and ultrasonic sensors detect/read, while analog output devices like control valves, meters, and VFDs control/write in a control system.
- The Bulletin number is a four-digit identifier for Logix controllers. ControlLogix starts with 1756, SoftLogix with 1789, and CompactLogix with 1769 or 1768.

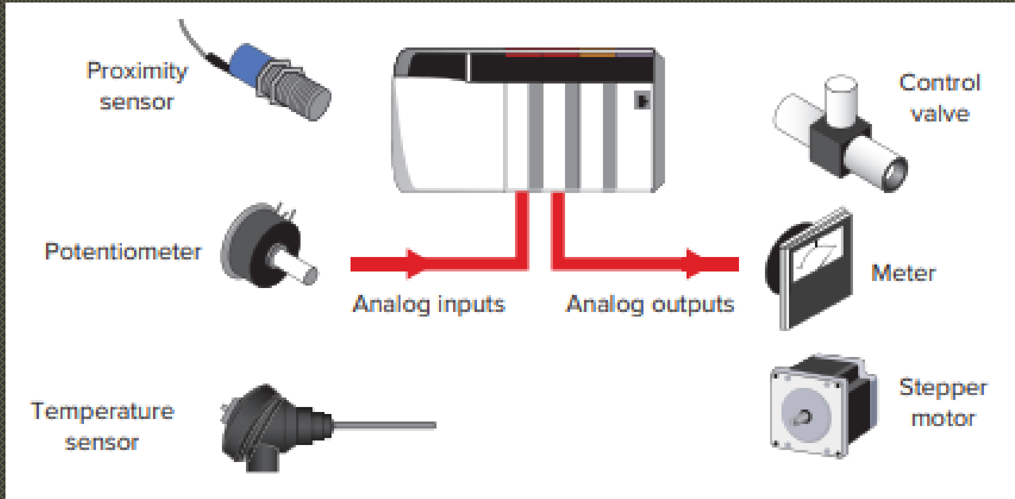


Figure 2-17 Analog input and output devices.

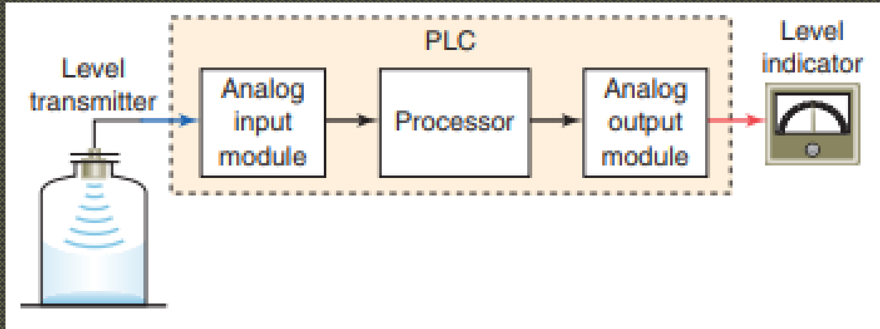


Figure 2-18 Analog input and output to a PLC.

HUMBER

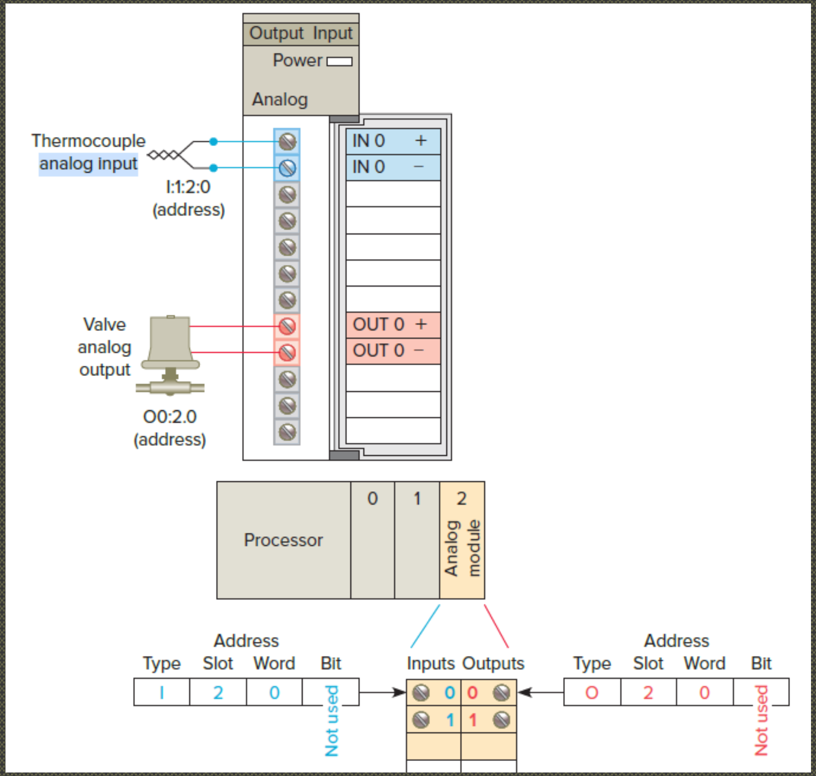
(Frank D. Petruzella Programmable Logic Controllers 6th edition

Analog Input Modules

- An analog signal represents a physical quantity that can have an infinite number of values.
- There are two basic types of analog input modules: voltage sensing and current sensing.

- Analog sensors measure a physical quantity, producing a corresponding voltage or current signal over a specific range.
  - temperature, speed, level, flow, weight, pressure, and position

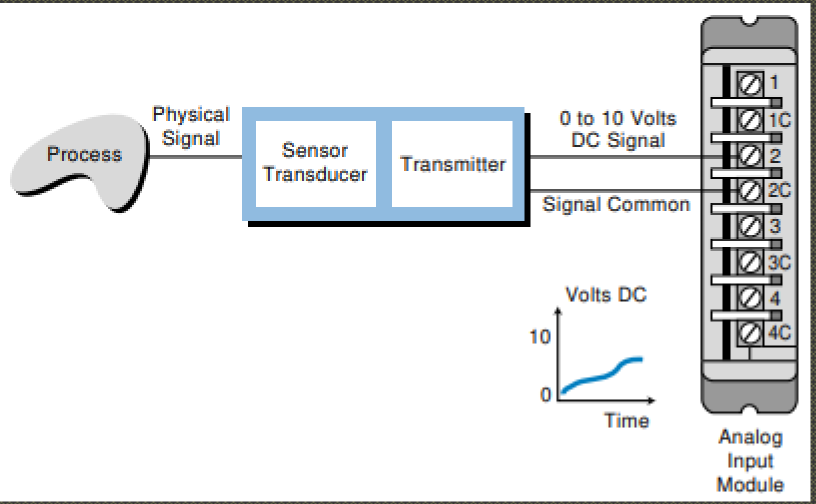
- Typical analog signal ranges are:
  - 1 ÷ 5 VDC
  - 0 ÷ 5 VDC
  - 0 ÷ 10 VDC
  - - 5 ÷ + 5 VDC
  - - 10 ÷ + 10 VDC
  - 4 ÷ 20 mA
  - 0 ÷ 20 mA
  - - 20 ÷ 20 mA
- The most common range is 4 ÷ 20 mA



HUMBER

Analog Modules - Inputs

- Analog input modules receive the measured quantity (voltage or current) from transmitters.
- A transducer turns a field device's variable (like pressure) into a low-level electric signal. This signal can be amplified by a transmitter and input into the analog module.
- Common physical quantities measured by a PLC analog input module are:
  - Position
  - Speed
  - Acceleration
  - Weight
  - Flow
  - Level
  - Pressure
  - Temperature
  - Distance



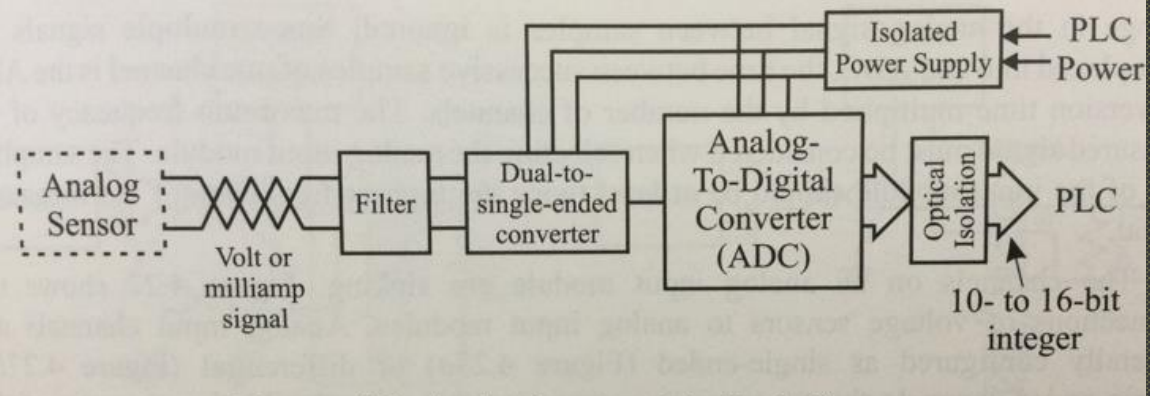


Figure 4.25. Analog input module block diagram.

L.A. Bryan & E.A. Bryan, Programmable Controllers: Theory and Implementation 2nd edition

HUMBER

Analog Modules - Inputs

- The ADC converts the analog voltage into an integer number, which has a finite resolution.
- The resolution of an analog input channel is the smallest change that can be sensed and corresponds to the width of each step.
- To explain the concept of resolution, in the figure below, the output integer is assumed to be 4-bit.
- The ADC requires a finite time to perform the analog-to-digital conversion – analog signal is sampled continuously, and the conversion is performed on each sample.
- Any change in analog signal between samples is ignored.

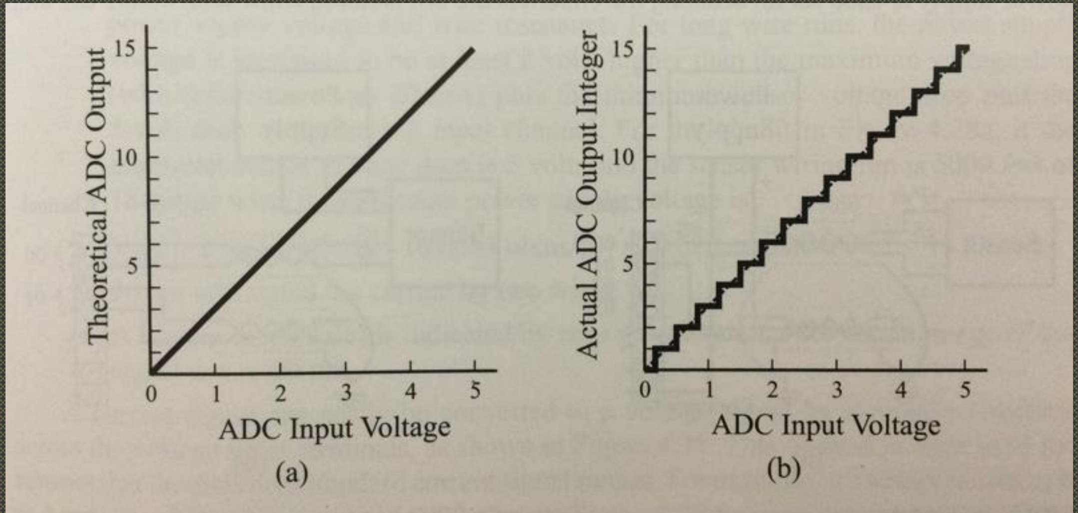


Figure 4.26. ADC input/output relationship: *(a)* theoretical output; *(b)* actual output integer.

- The maximum frequency of the measured signal must be considered when selecting the sampling rate.
- The sampling rate of the input module should be at least twice the highest frequency of the measured signal.

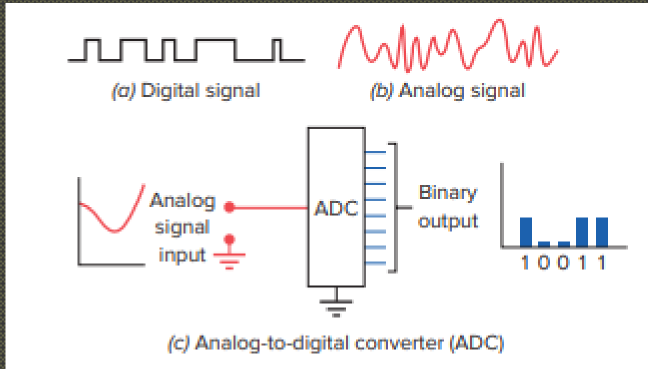


Figure 5-50 Digital and analog signals.

(Frank D. Petruzella Programmable Logic Controllers 6th edition

Analog Modules - Inputs

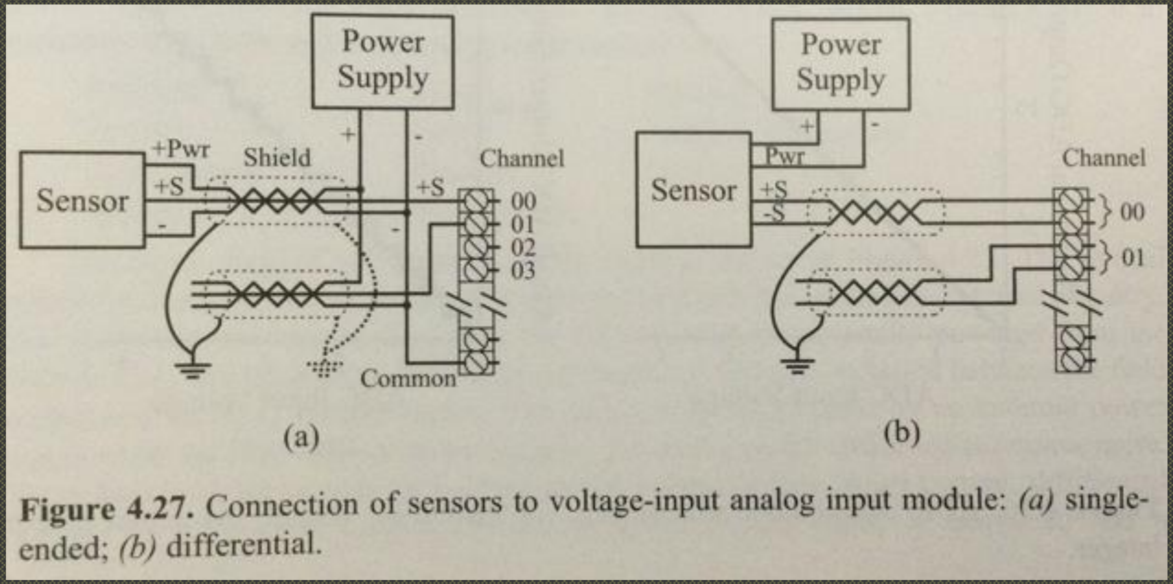


Figure 4.27. Connection of sensors to voltage-input analog input module: *(a)* single-ended; *(b)* differential.

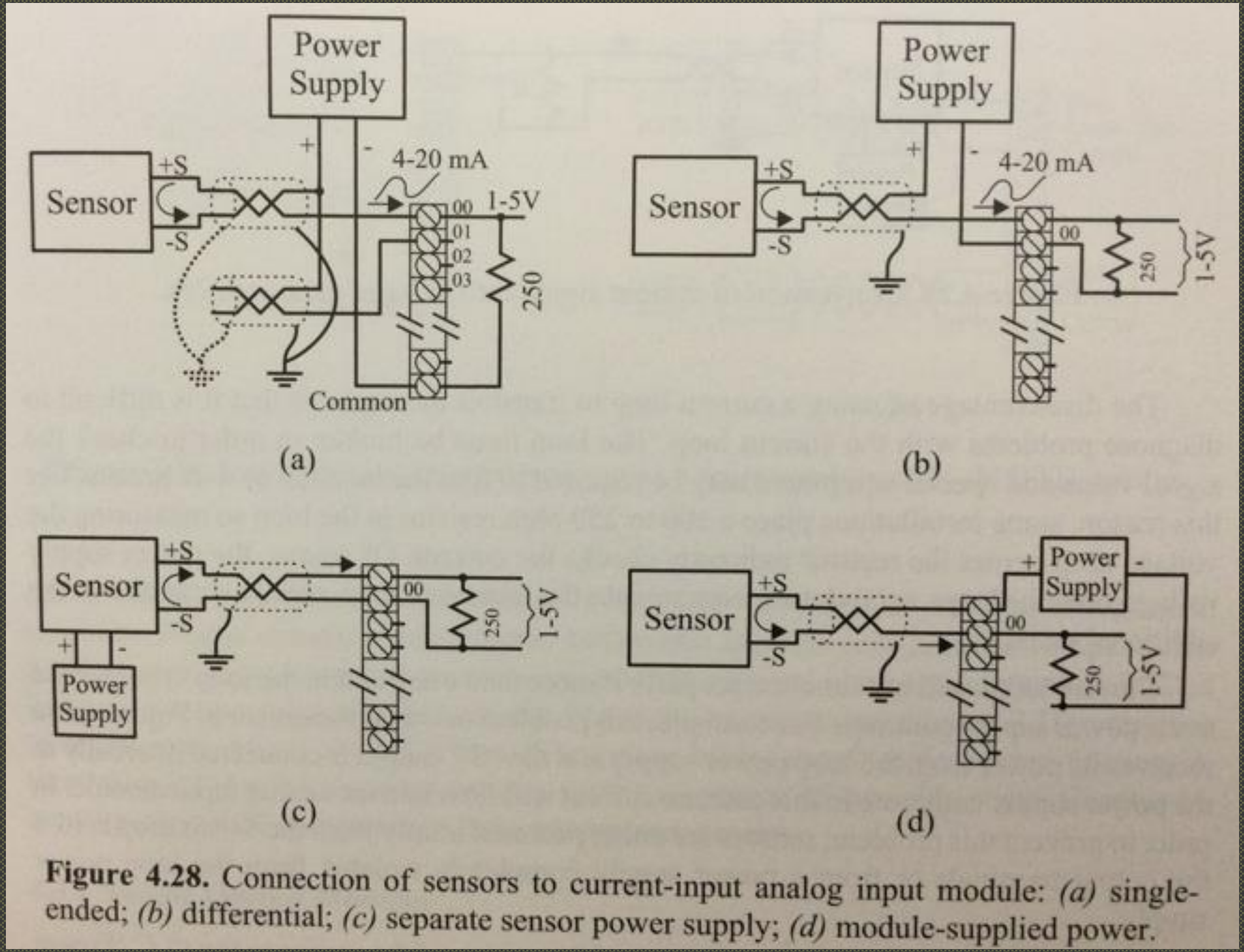


Figure 4.28. Connection of sensors to current-input analog input module: *(a)* single-ended; *(b)* differential; *(c)* separate sensor power supply; *(d)* module-supplied power.

Two disadvantages of using voltage as analog signal:

- Easily pick up electromagnetic noise
- Limited distance because of voltage drop

The advantage of using current as analog signal are:

- Less susceptible to EMI
- Signal transmitted longer distance
- A broken wire is indicated as 0 mA
- Power and signal are carried by two wires

Analog Modules - Inputs

- Easily converted in voltage signal by connecting a resistor across the voltage terminals of the analog module.
- The disadvantage of using current loop to transmit the signal is challenging when troubleshooting.
- A separate visual indicator, useful for diagnostics, may be part of the wiring of the sensor to the PLC.

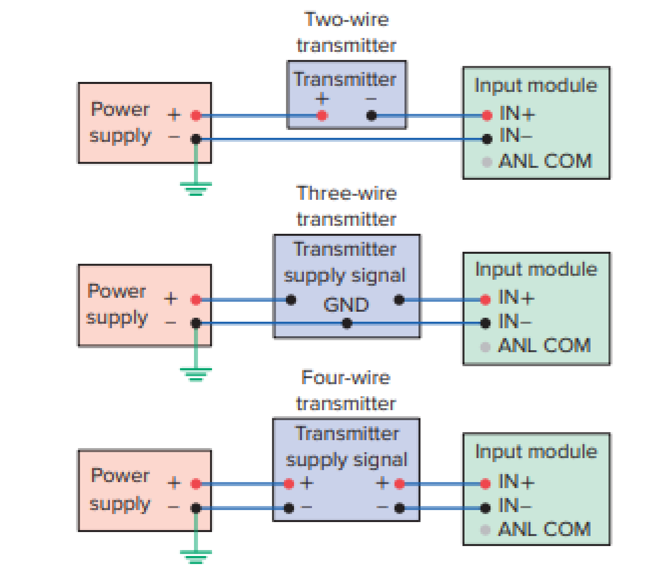


Figure 2-22 Wiring two-, three-, and four-wire sensors to an analog input module.

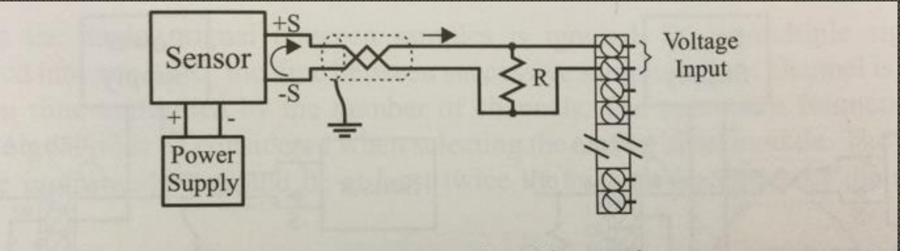


Figure 4.29. Conversion of current signal into voltage measurement.

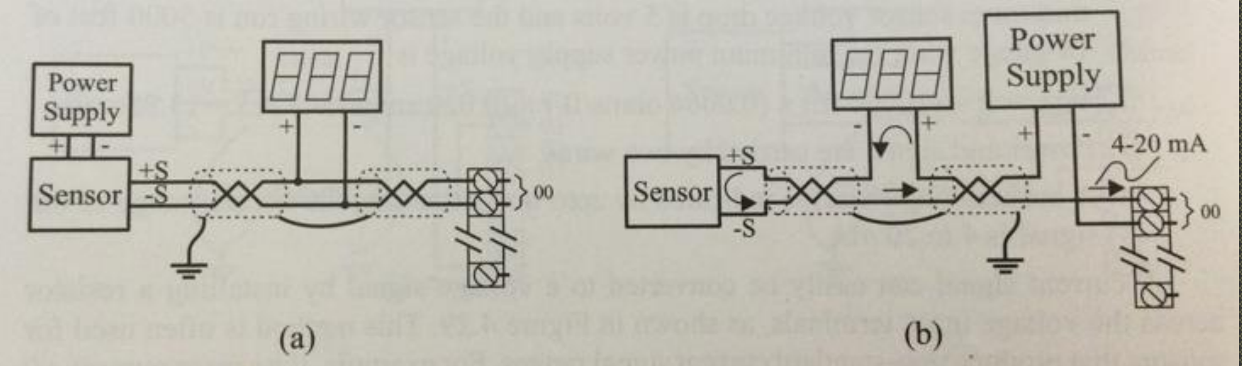


Figure 4.30. Connection of indicator to analog input signal: *(a)* voltage sensor; *(b)* current sensor.

(Frank D. Petruzella Programmable Logic Controllers 6th edition

HUMBER

Analog Modules - Outputs

- The analog output module gets digital data from the processor and converts it to a proportional voltage or current to control an analog device in the field.
  - The generated voltage or current signal drives other device to manipulate a physical quantity.
- Common devices controlled by a PLC analog output module are:
  - Variable speed motor
  - Current-to-pneumatic converter (pneumatic regulating valve position)
  - Analog meter
  - Chart recorder

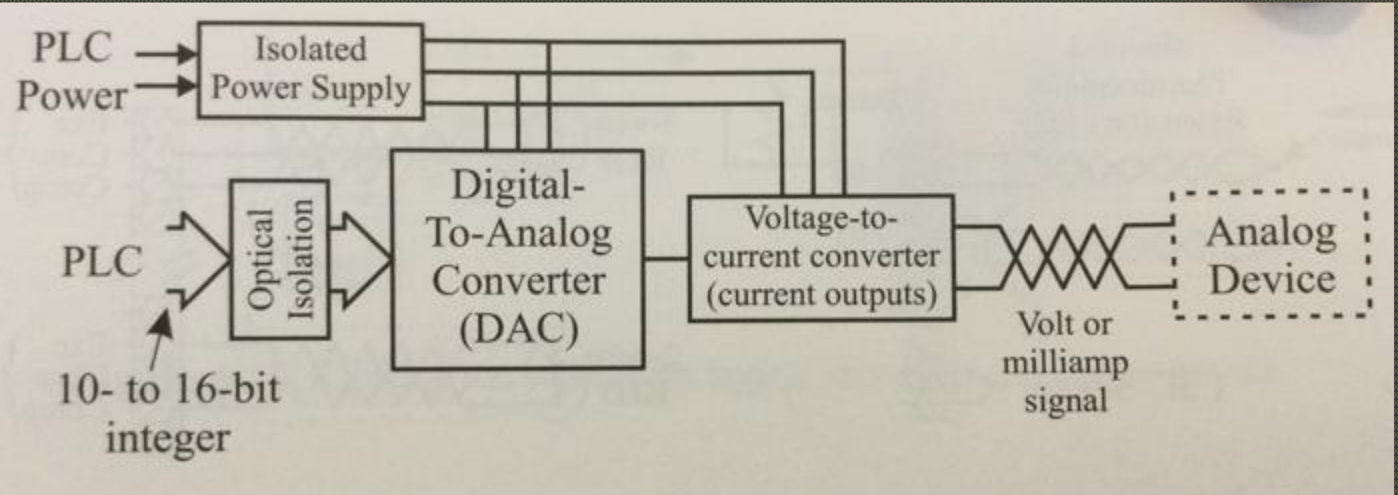


Figure 4.32. Analog output module block diagram.

Typical analog signal ranges are:
1 ÷ 5 VDC
0 ÷ 5 VDC
0 ÷ 10 VDC
- 5 ÷ + 5 VDC
- 10 ÷ + 10 VDC
4 ÷ 20 mA
0 ÷ 20 mA
- 20 ÷ 20 mA
The most common range is 4 ÷ 20 mA

HUMBER

Analog Modules - Outputs

The channels on an analog output module are sourcing.

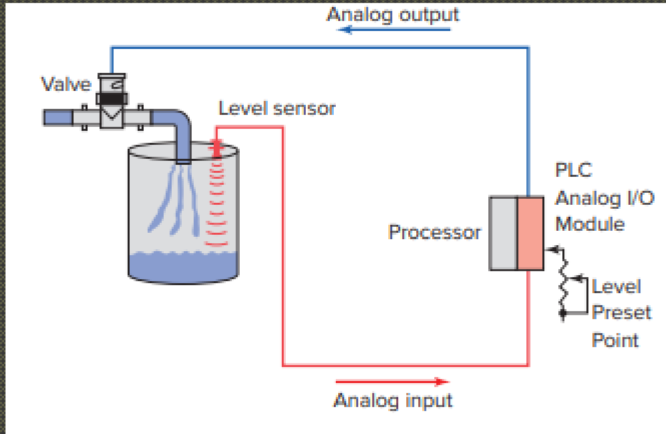


Figure 5-51 Analog control for a tank-filling process.

(Frank D. Petruzella Programmable Logic Controllers 6th edition

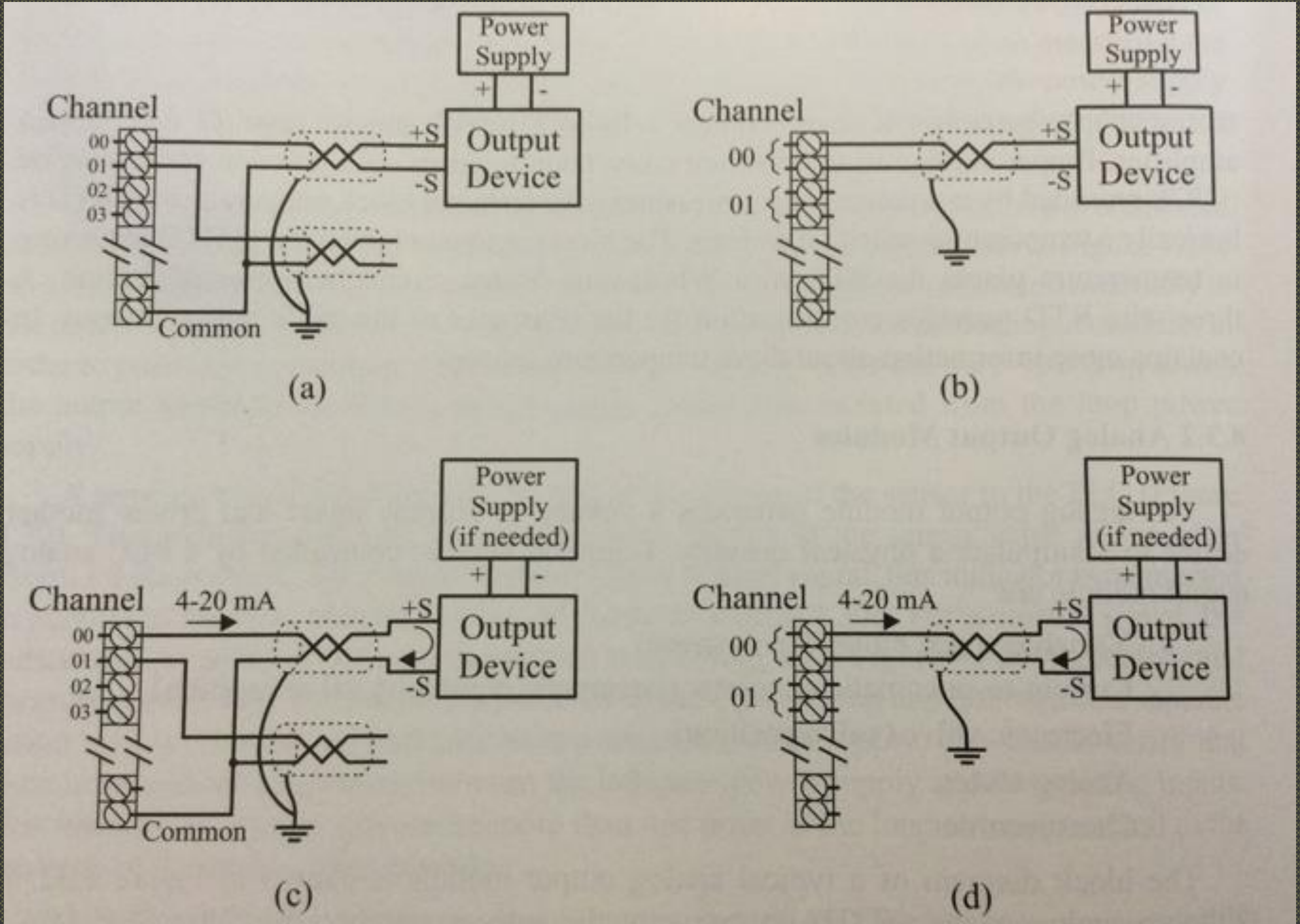
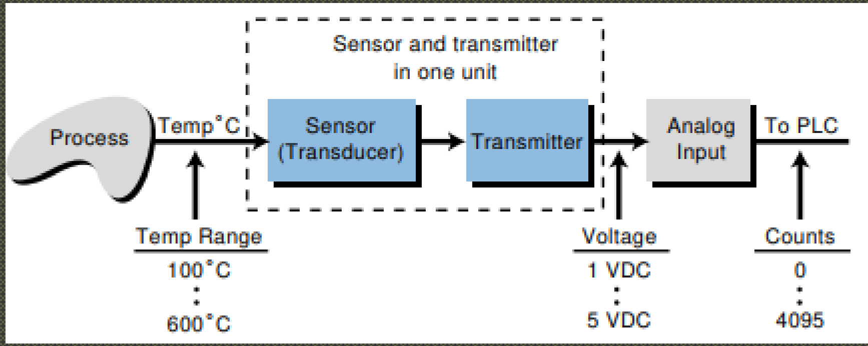


Figure 4.33. Connection of devices to analog output module: *(a)* single-ended voltage; *(b)* differential voltage; *(c)* single-ended current; *(d)* differential current.

HUMBER

ANALOG MODULES – EXAMPLE 1

An input module, which is connected to a temperature transducer, has an A/D with a 12-bit resolution. When the temperature transducer receives a valid signal from the process (100 to 600°C), it provides, via a transmitter, a +1 to +5 VDC signal compatible with the analog input module.



a. Find the equivalent voltage change for each count change (the voltage change per degree Celsius change) and the equivalent number of counts per degree Celsius, if the input module transforms the data into a linear 0 to 4095 counts.
b. find the same values for a module with a 10-bit resolution.

ANALOG MODULES – EXAMPLE 1

a. The changes (Δ) in temperature, voltage, and input counts are 500°C, 4 VDC, and 4095 counts.

- Therefore, the voltage change for a 1°C temperature change is:

  Δ 500 °C = Δ 4 VDC

  1° C = 4 VDC / 500 = 8. 0 mVDC

- The change in voltage for each input count is:

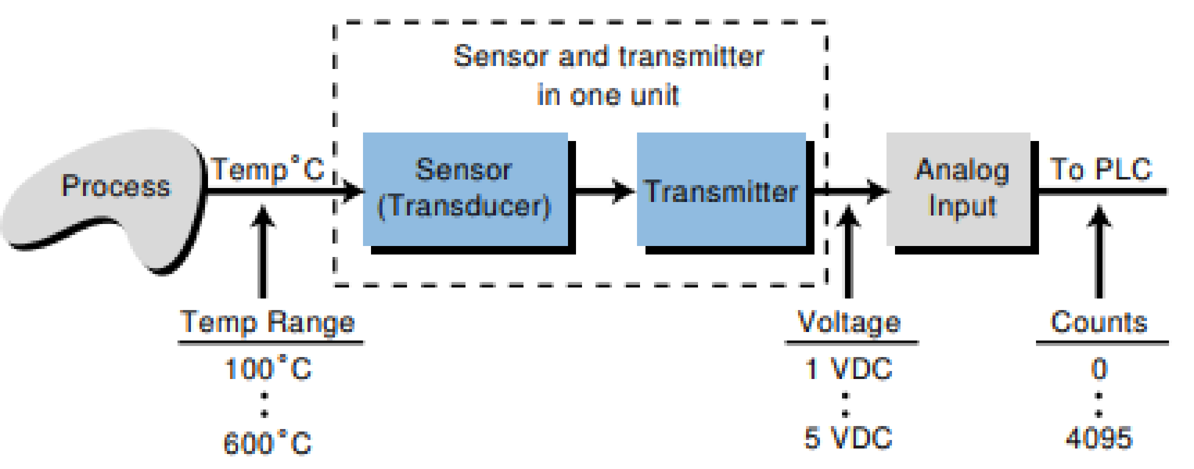
  Δ 4095 counts = Δ 4 VDC

  1 count = 4 VDC / 4095 = 0.9768 mVDC

- Therefore, the corresponding number of counts per degree Celsius is:

  Δ 500 °C = Δ 4095 counts

  1° C = 4095 counts / 500 = 8.19 counts



ANALOG MODULES – EXAMPLE 1

b. The changes (Δ) in temperature, voltage, and input counts are 500°C, 4 VDC, and 1023 counts.

- Therefore, the voltage change for a 1°C temperature change is:

  Δ 500 °C = Δ 4 VDC

  1° C = 4 VDC / 500 = 8. 0 mVDC

- The change in voltage for each input count is:

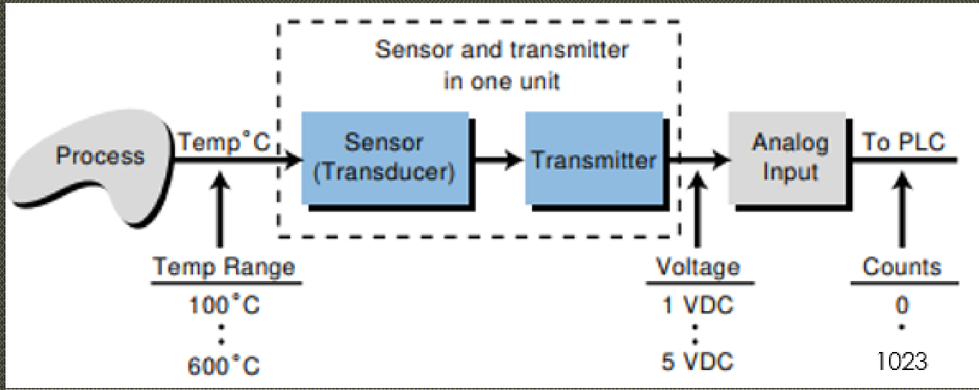
  Δ 1023 counts = Δ 4 VDC

  1 count = 4 VDC / 1023 = 3.91 mVDC

- Therefore, the corresponding number of counts per degree Celsius is:

  Δ 500 °C = Δ 1023 counts

  1° C = 1023 counts / 500 = 2.046 counts

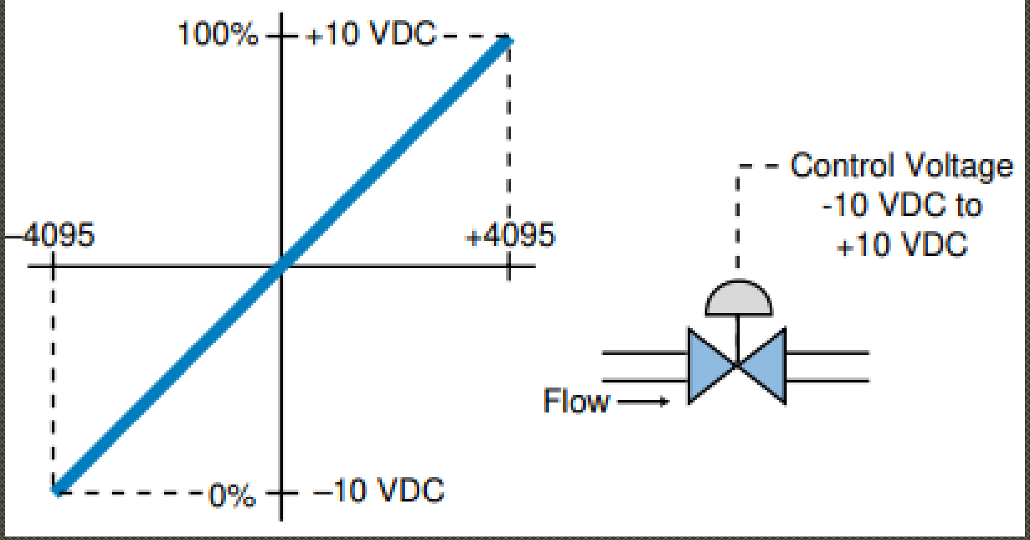


Temperature	Voltage Signal	Input Counts
100°C	1 VDC	0
.	.	.
.	.	.
.	.	.
600°C	5 VDC	1023

ANALOG MODULES – EXAMPLE 2

- A transducer connects an analog output module with a flow control valve capable of opening from 0% to 100% of total flow.
- The percentage of opening is proportional to a –10 to +10 VDC signal at the transducer's input.
- The bipolar output module has a 12-bit D/A (binary) with an additional sign bit that provides polarity to the output swing.

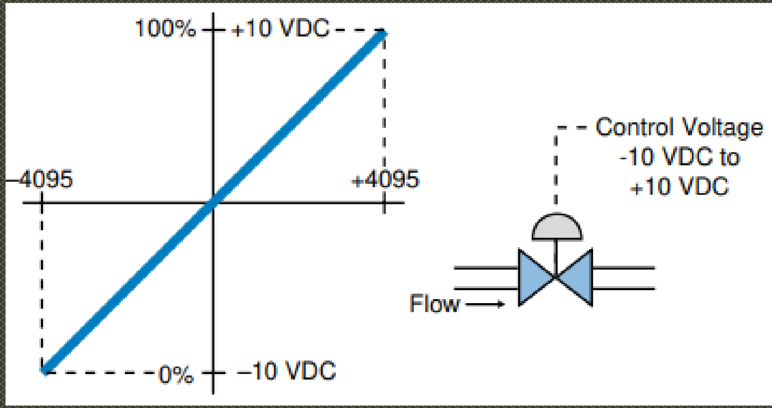
- Tabulate the relationship between percentage opening, output voltage, and counts for the output module in increments of 10% (i.e., 10%, 20%, etc.).



ANALOG MODULES – EXAMPLE 2

- To create the table, find equivalent values for each variable. Express the solution in percentage increments by calculating the changes accordingly.

ΔPercentage	ΔVoltage (–10 to + 10)	ΔCounts (–4095 to + 4095)
100	20	8190



- 1% change as function of voltage = 20 VDC / 100 = 0.2 VDC
- 1% change as function of counts = 8190 counts / 100 = 81.90 counts

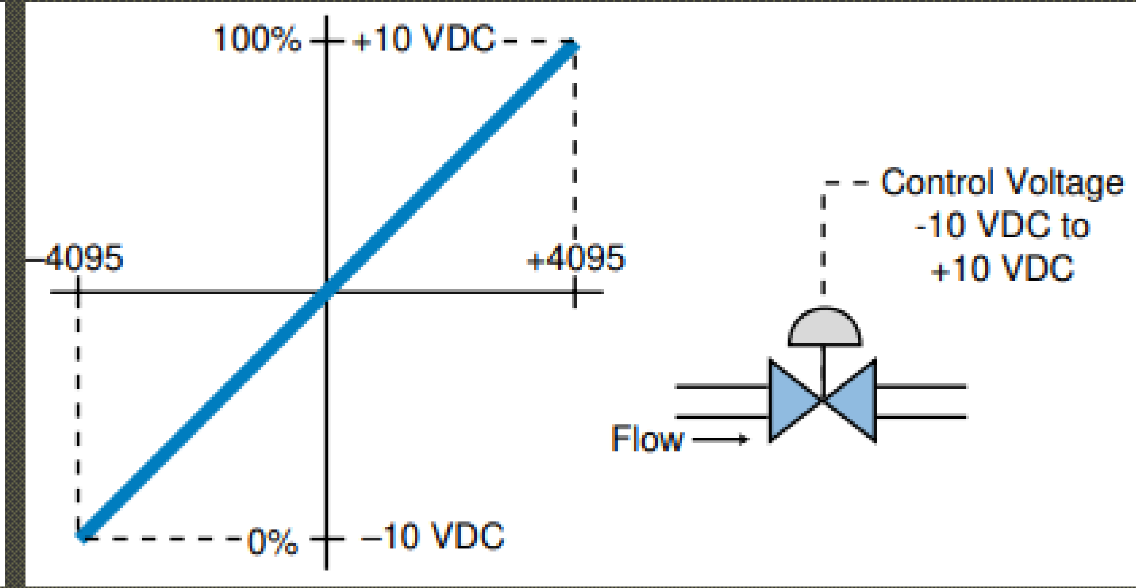
- These computations are magnitude changes. To implement the table, the offset values for the voltage and counts must be added, taking into consideration the bipolar effect of the module and the negative-to-positive changes in counts.
- Therefore, to obtain the voltage and count equivalents per percentage change, add the offset voltage and count values when the percentage is at 0%.
    - Percentage as function of voltage = (0.2 × P) − 10 VDC
    - Percentage as function of counts = (81.9 × P) − 4095 counts

  where P is the percentage to be used in the table

ANALOG MODULES – EXAMPLE 2

The PLC's program calculates output counts using a set algorithm. The output values are expressed in engineering units in a range from 0000 to 9999 (binary or BCD). These need to be converted to counts, ranging from -4095 to +4095.

Percentage Opening	Output Voltage	Counts
0%	–10 VDC	–4095
10	–8	–3276
20	–6	–2457
30	–4	–1638
40	–2	–819
50	0	0
60	+2	+819
70	+4	+1638
80	+6	+2457
90	+8	+3276
100	+10	+4095



L.A. Bryan & E.A. Bryan, Programmable Controllers: Theory and Implementation 2nd edition

COMPACT LOGIX CONFIGURATION

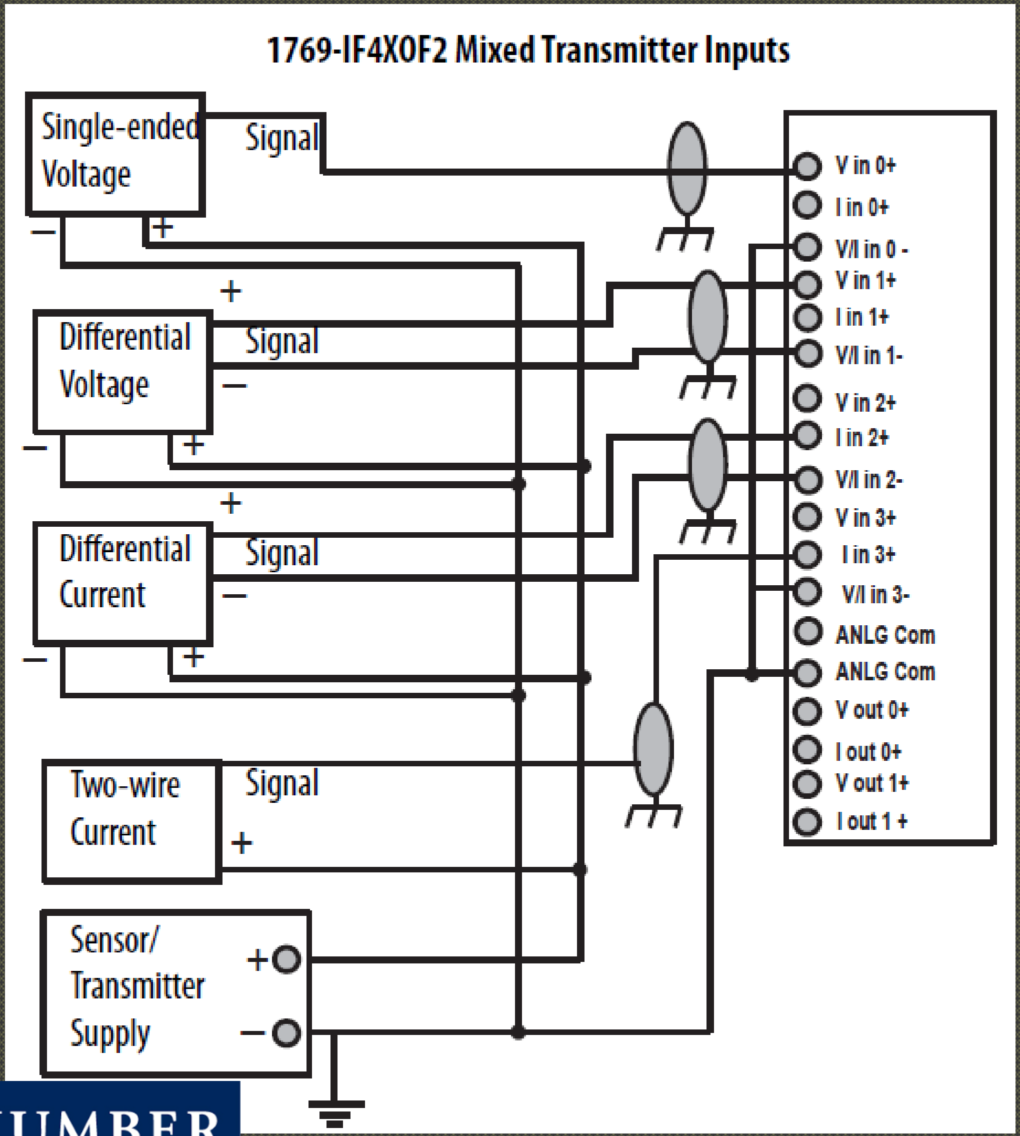
Example

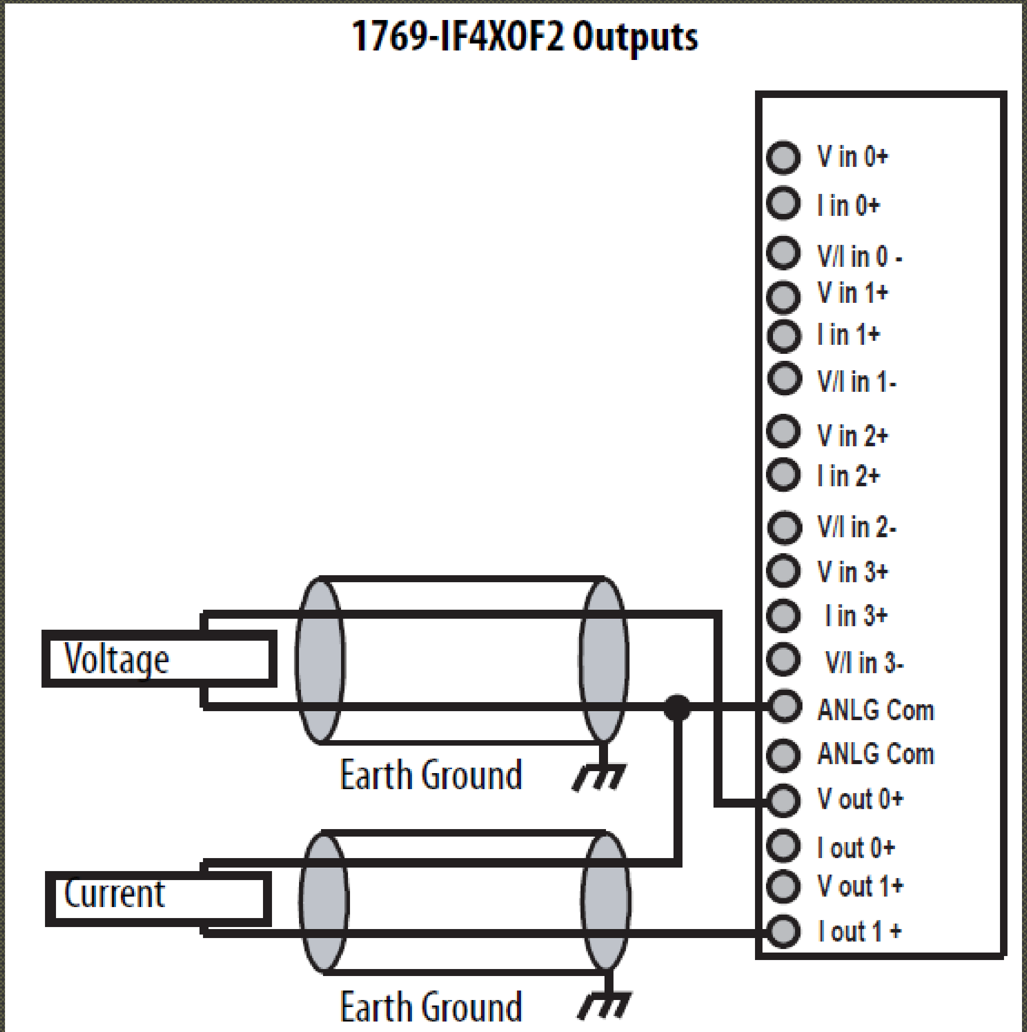
1769 – IF4XOF2 ANALOG 4 IN/4 OUT - MODULE 3

HUMBER

1769 – IF4XOF2 ANALOG 4 IN/4 OUT - MODULE 3

Example





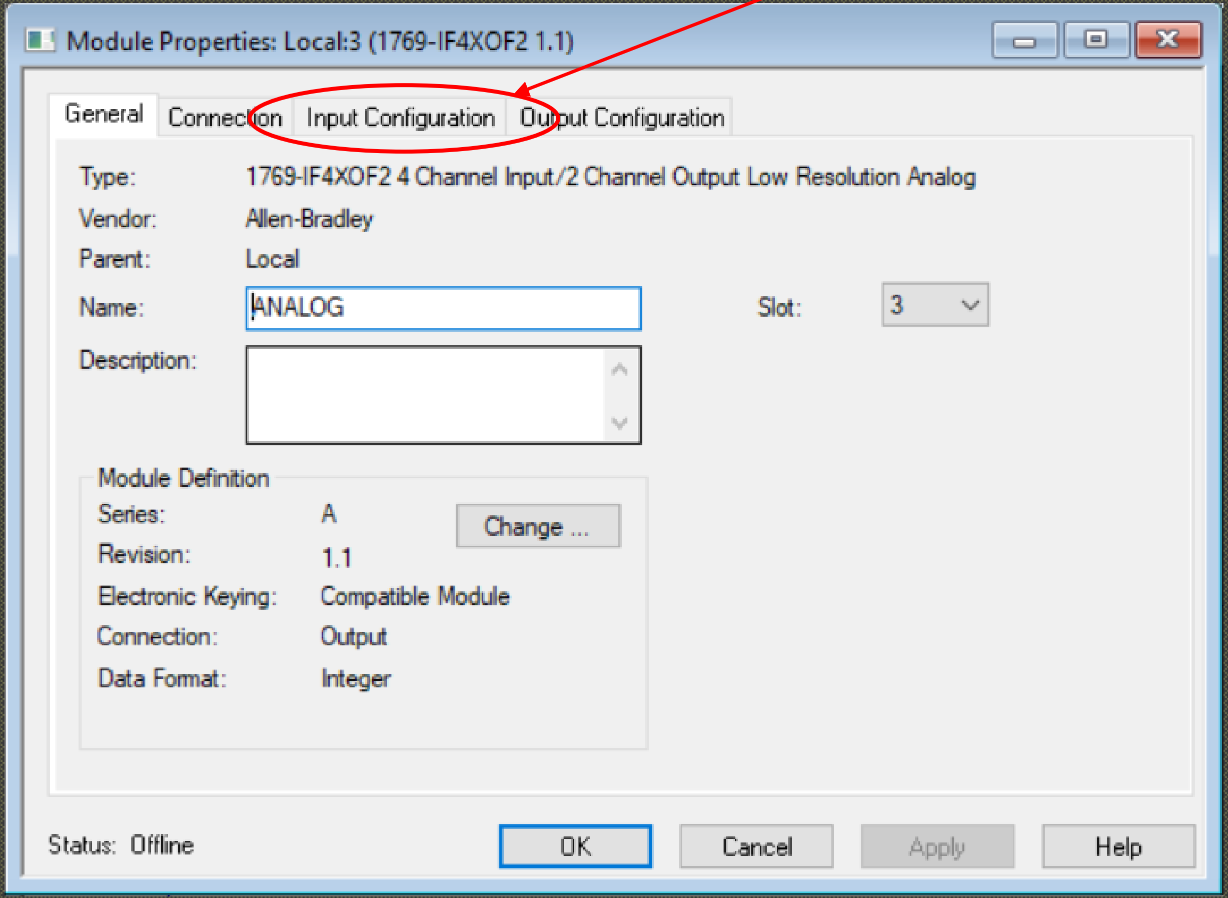
HUMBER

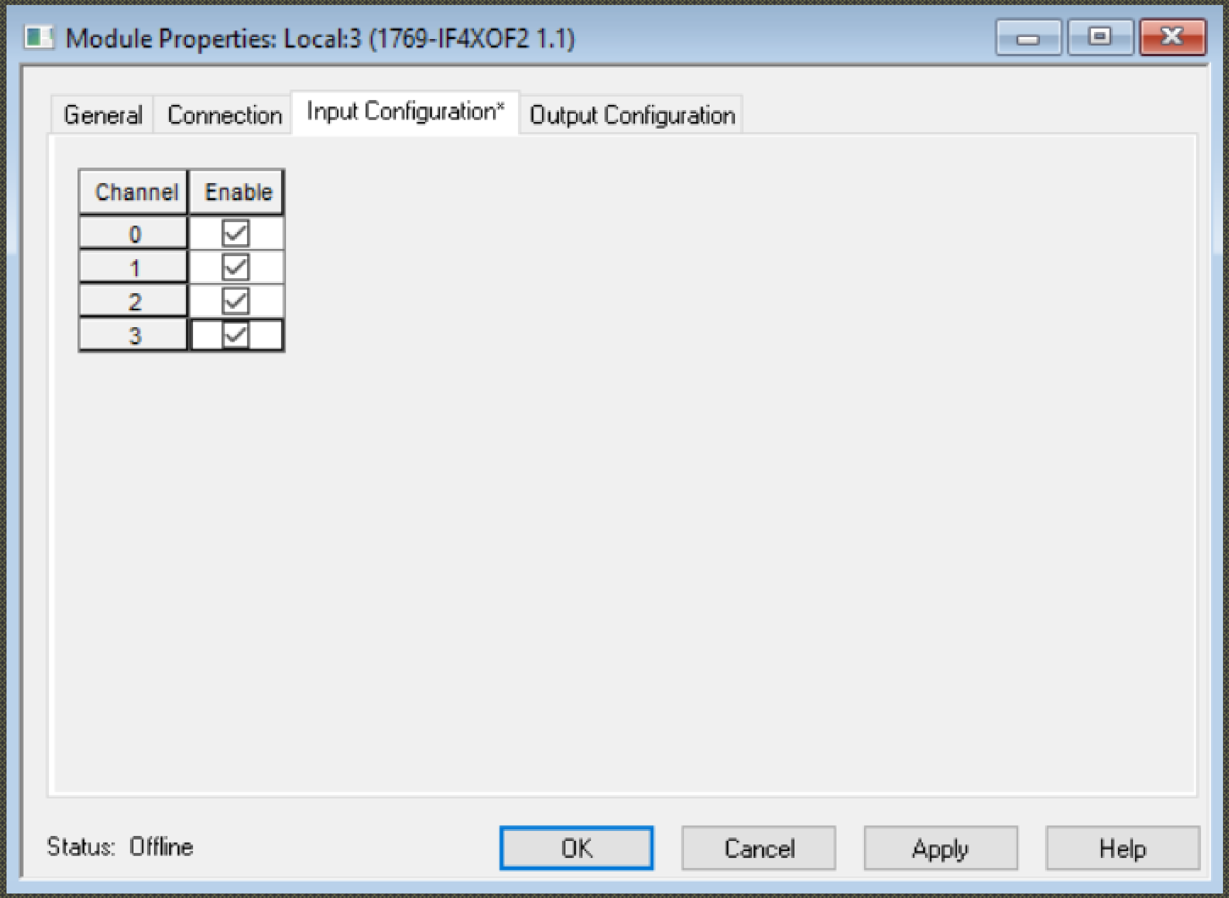
ANALOG INPUT AND OUTPUT CHANNELS ACTIVATION

Example

- Open Input Channels configuration

- Check mark channels you use
- Click on OK



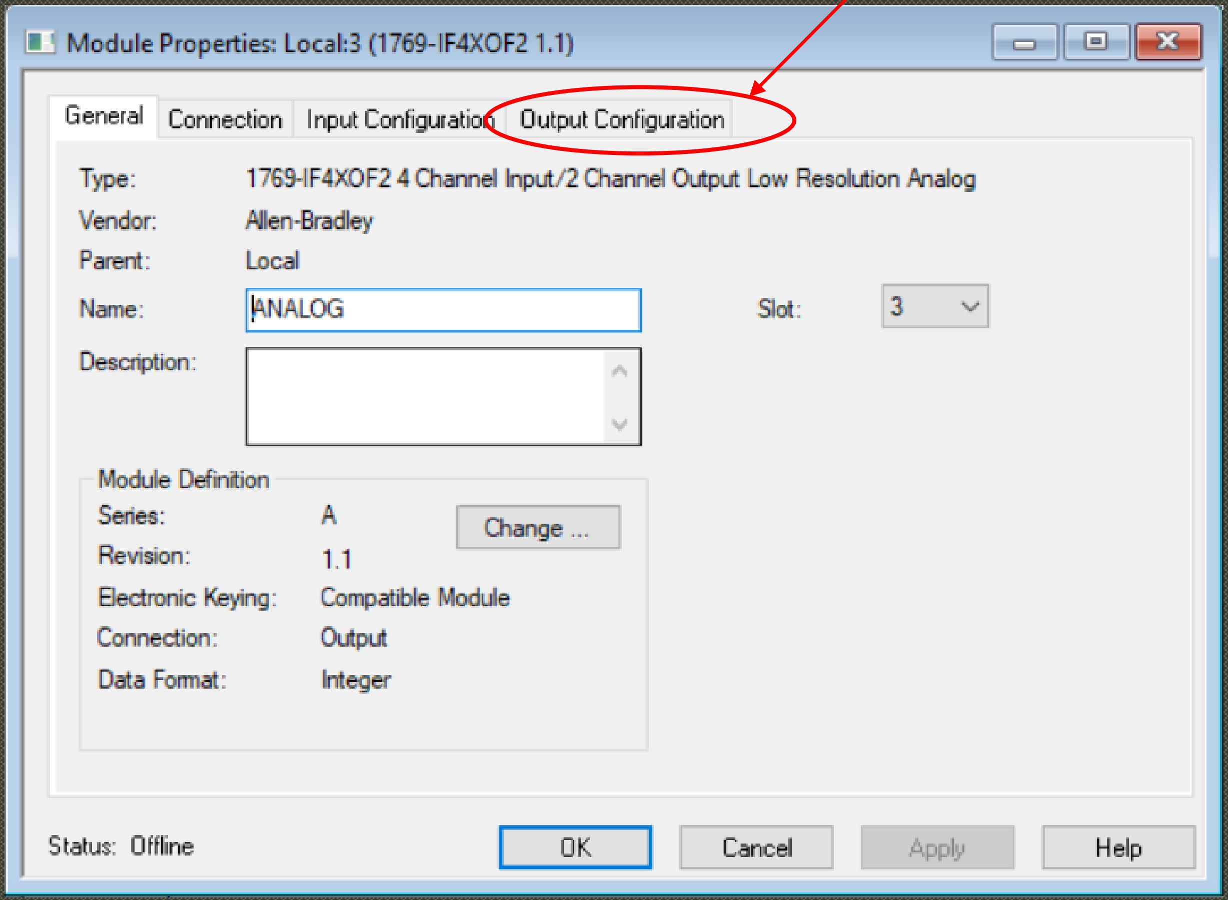


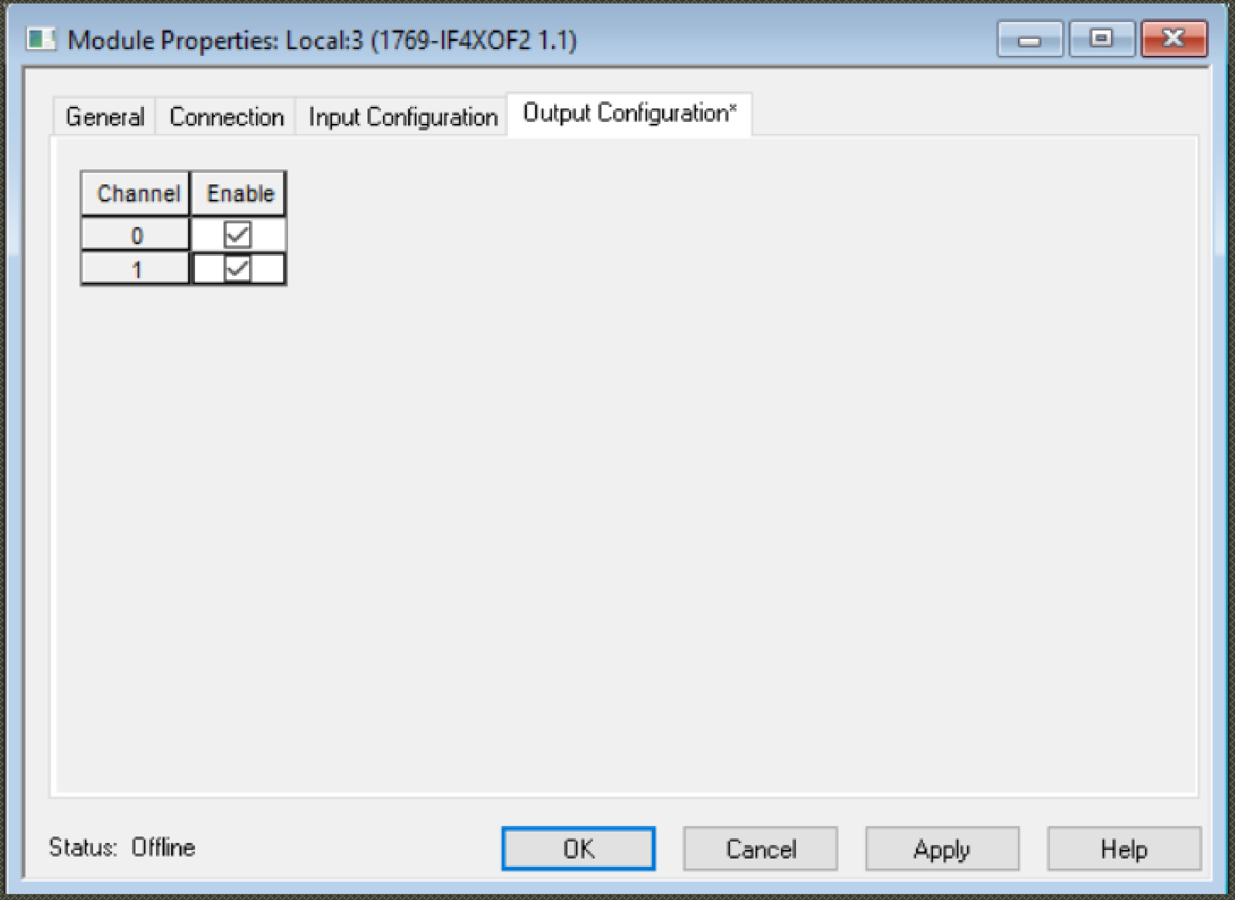
ANALOG INPUT AND OUTPUT CHANNELS ACTIVATION

Example

- Open Output Channels configuration

- Check mark channels you use
- Click on OK





1769 – IF4XOF2 ANALOG 4 IN/4 OUT - MODULE 3

Example

- The analog input channels convert and digitally store analog data for the controller. The module supports connections from any combination of up to four voltage or current analog sensors.

- The output channels provide two single-ended analog outputs, each individually configurable for voltage or current.

Normal Operating Range	Full Module Range
0 to +10V dc	0.0 to +10.5V dc
0 to 20 mA	0 to 21 mA

Input Data File

The input data file provides access to input data for use in the control program, over-range indication for the input and output channels, and output data feedback as described below.

Word	Bit Position															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	SGN	Analog Input Data Channel 0								0	0	0	0	0	0	0
1	SGN	Analog Input Data Channel 1								0	0	0	0	0	0	0
2	SGN	Analog Input Data Channel 2								0	0	0	0	0	0	0
3	SGN	Analog Input Data Channel 3								0	0	0	0	0	0	0
4	Not Used[(1)]												I3	I2	I1	I0
5	Not Used	H0	Not Used	H1	Not Used[(1)]								E1	E0	O1	O0
6	SGN	Output Data Echo/Loopback for Output Channel 0								0	0	0	0	0	0	0
7	SGN	Output Data Echo/Loopback for Output Channel 1								0	0	0	0	0	0	0

(1) All unused bits are set to 0 by the module.

Output Data File

The output data file applies only to output data from the module as shown in the table below.

Word	Bit Position															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	SGN	Analog Output Data Channel 0								0	0	0	0	0	0	0
1	SGN	Analog Output Data Channel 1								0	0	0	0	0	0	0

HUMBER

1769 – IF4XOF2 ANALOG 4 IN/4 OUT - MODULE 3

Example

- Scaling creates a linear relationship between input and scaled values to produce an output.
- It lets you convert analog input from a sensor into the necessary engineering units for the application.

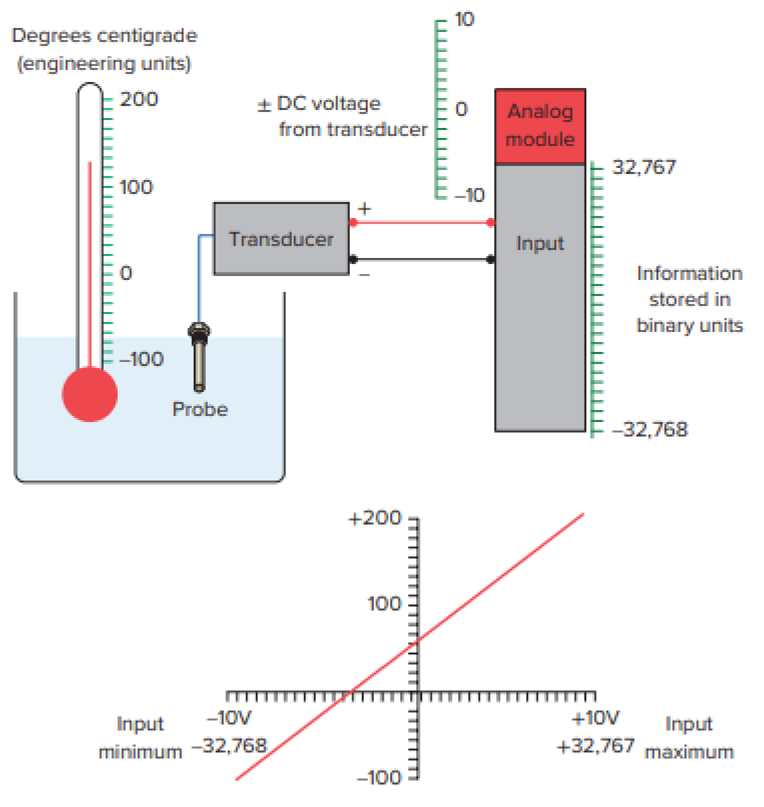
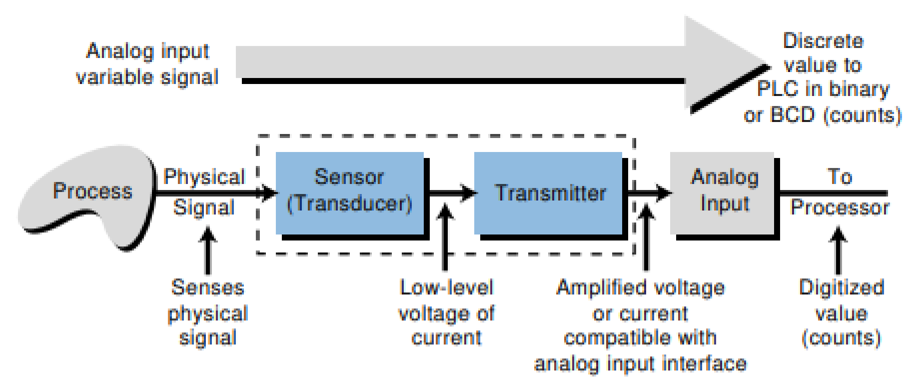


Figure 2-25 Measuring temperature.

(Frank D. Petruzella Programmable Logic Controllers 6th edition

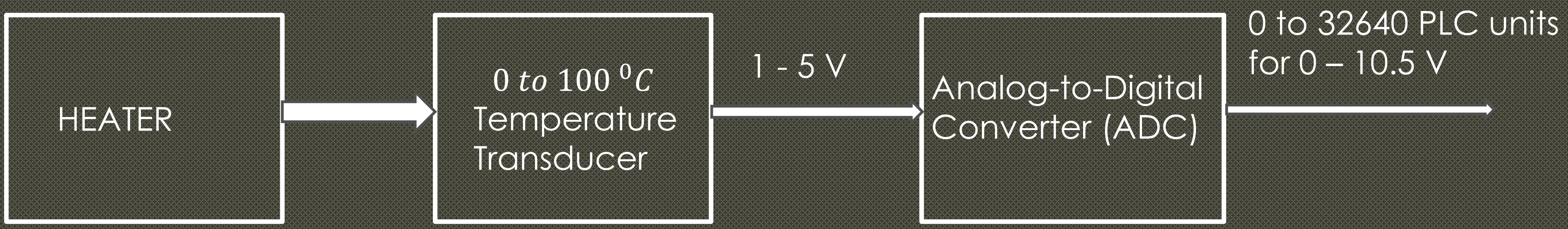


L.A. Bryan & E.A. Bryan, Programmable Controllers: Theory and Implementation 2nd edition

Conversion of Physical Quantity

Example

The transducer measures temperature from 0 to 100°C and transmits a 4-20 mA signal through a 250 Ω resistor (1 V – 5 V). The analog input module has an ADC with an output range of 0 to 32640 corresponding to a 0 to 10.5 VDC input. Find the transmitted voltage and ADC output for temperatures of 0, 25, 50, and 78 degrees.



For voltage-type analog input, we convert the 4-20 mA range to voltage. At 4 mA, there is a 1V drop across the 250 Ω resistor, and at 20 mA, it is a 5V drop across the same resistor.

4 mA ⇒ 1 V
20 mA ⇒ 5 V

HUMBER

1769 – IF4XOF2 I/O ANALOG CHANNELS

Example

4 INPUT ANALOG CHANNELS:

- CH 0 0 V – 10.5 V
- CH 1 0 V – 10.5 V
- CH 2 0 V – 10.5 V
- CH 3 0 V – 10.5 V

2 OUTPUT ANALOG CHANNELS:

- CH 0 0 V – 10.5 V
- CH 1 0 V – 10.5V

All input and output analog channels have decimal range from 0 to 32640 plc units corresponding to 0 – 10.5 volts

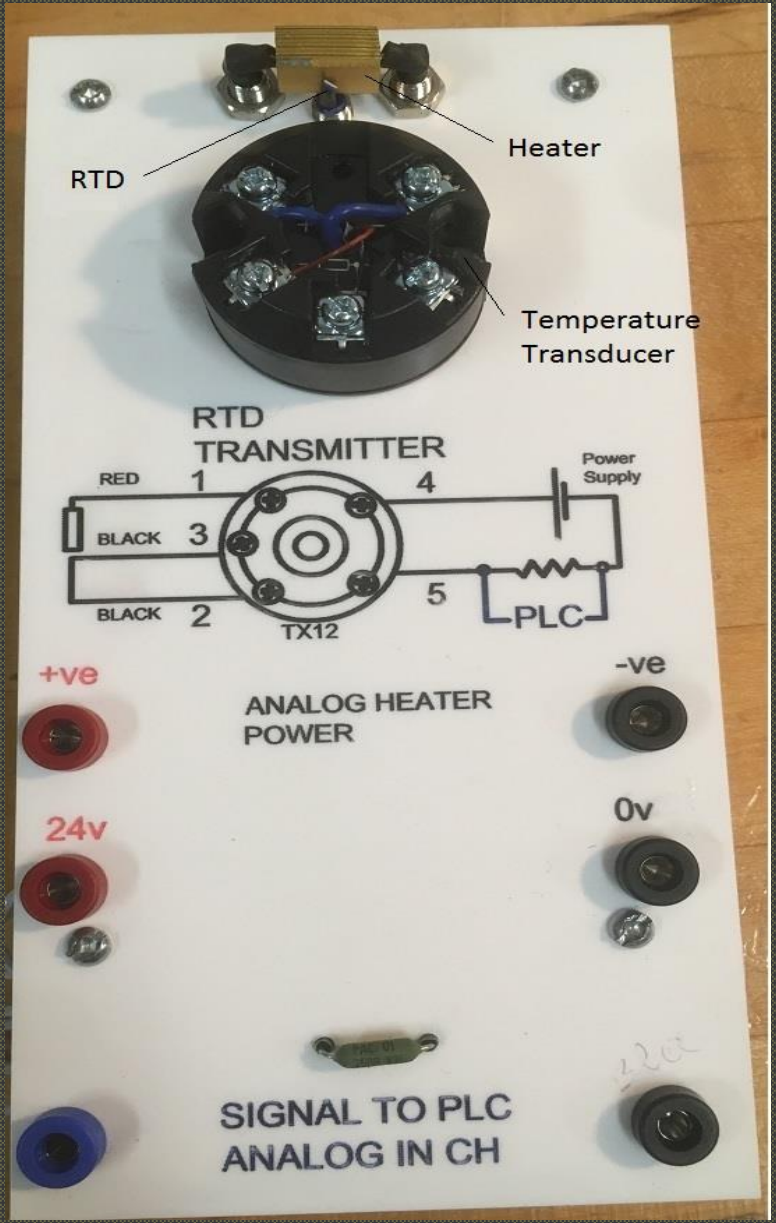
BIT 15	BIT 14	BIT 13	BIT 12	BIT 11	BIT 10	BIT 9	BIT 8	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	
SIGN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DECIMAL 0

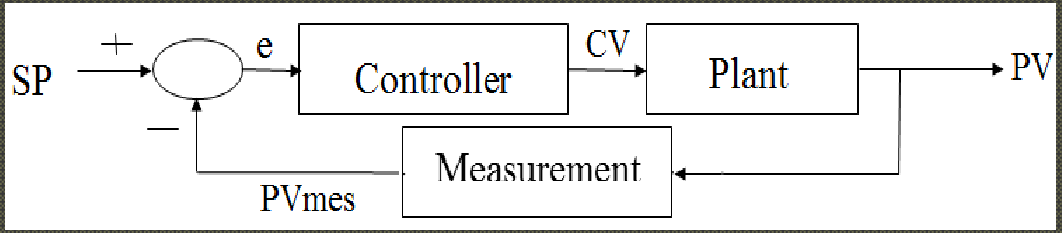
BIT 15	BIT 14	BIT 13	BIT 12	BIT 11	BIT 10	BIT 9	BIT 8	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	
SIGN	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	DECIMAL 32640

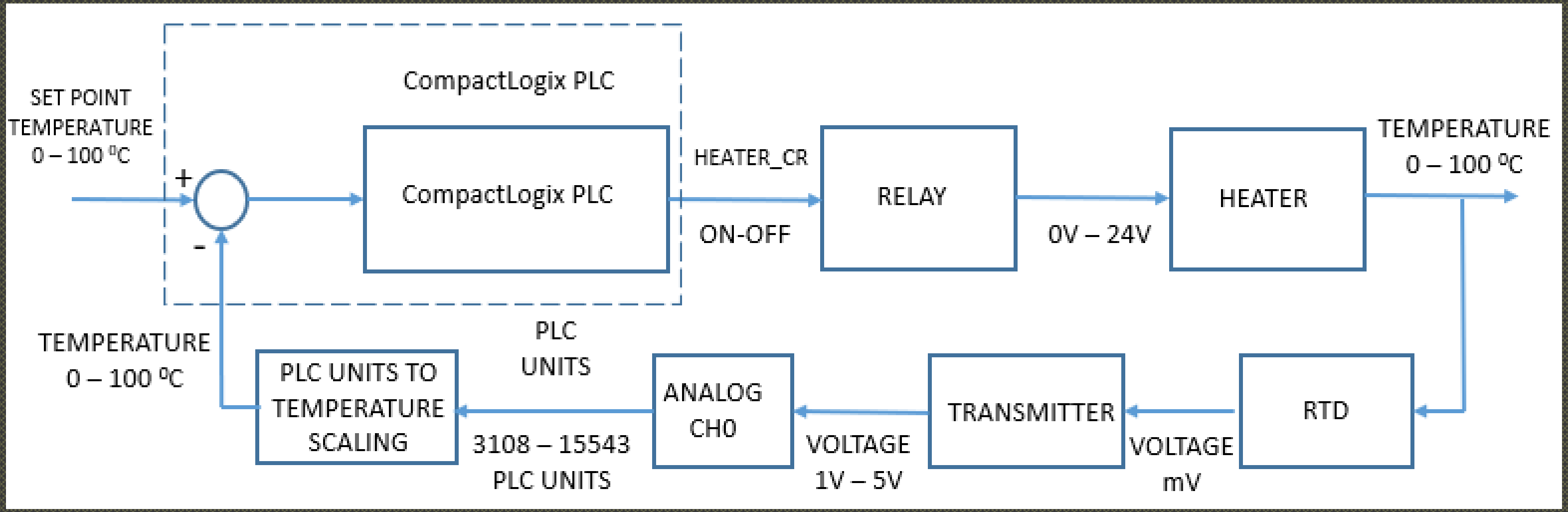
TEMPERATURE ON-OFF CONTROL EXAMPLE

CLOSED-LOOP SYSTEM

Example

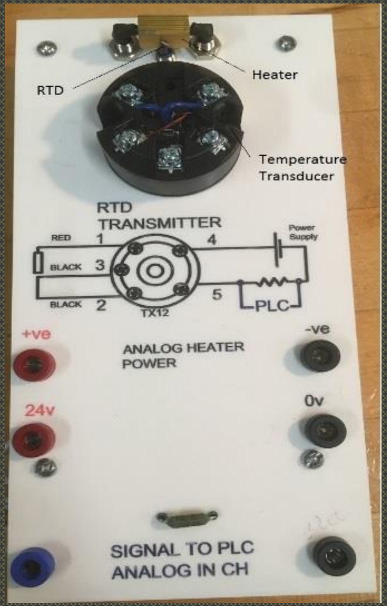


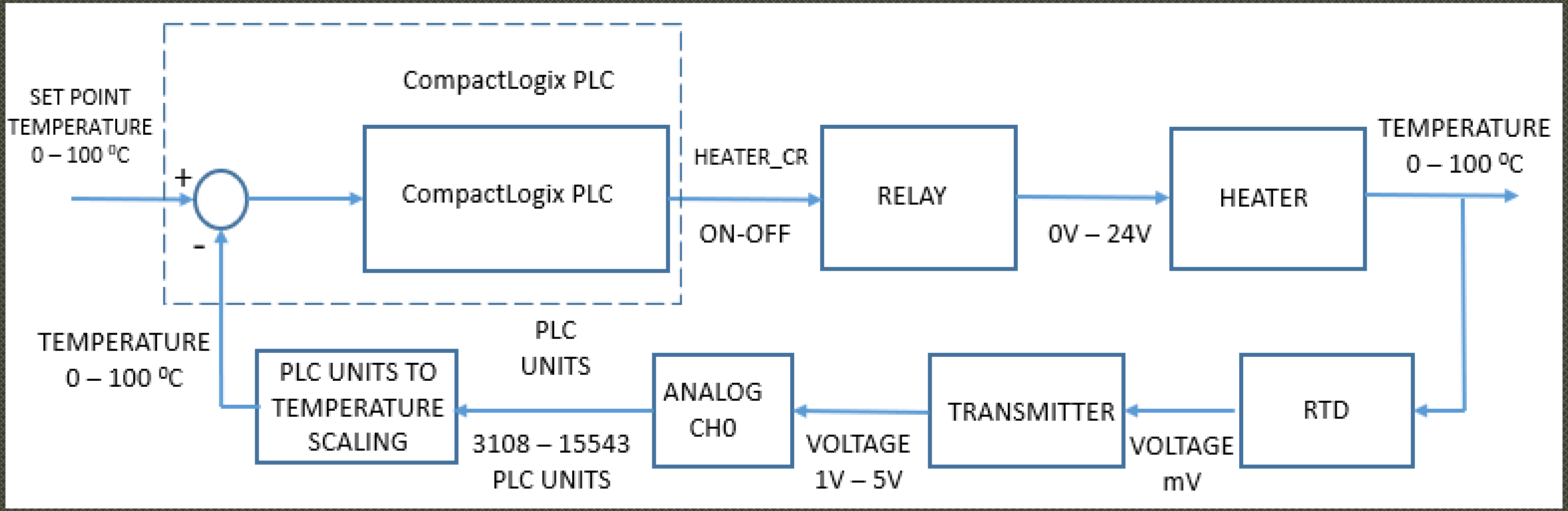




TEMPERATURE ON-OFF CONTROL EXAMPLE

Example





TRANSMITTER OUTPUT TO THE INPUT CHANNEL 0: 1 V – 5 V IS CONVERTED TO 3108 – 15543 PLC UNITS RANGE

The true readings from the calculations below: 24 * 128 = 3072; 25 * 128 = 3200; 121 * 128 = 15488; 122 * 128 = 15616

0 ⁰C –> 1 V

10.5 V – 32640 PLC UNITS

1 V –> X PLC UNITS

X = 1 * 32640 / 10.5 = 3108

1 V –> 3108 PLC UNITS

0 ⁰C – 3108 PLC UNITS

100 ⁰C – 5 V

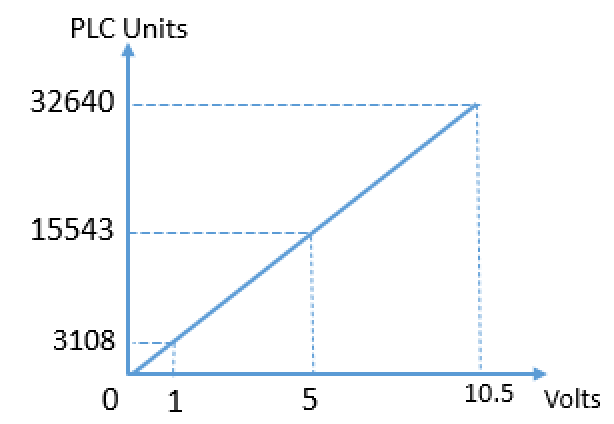
10.5 V – 32640 PLC UNITS

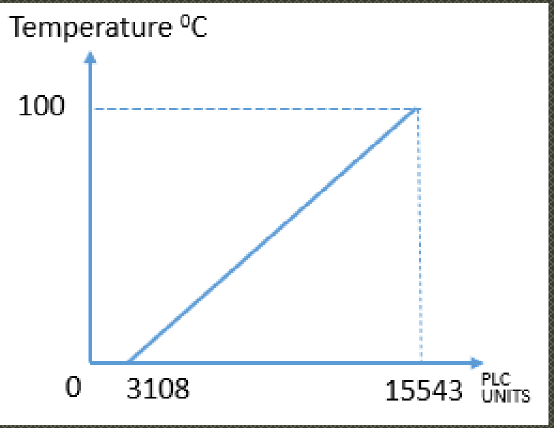
5 V –> X PLC UNITS

X = 5 * 32640 / 10.5 = 15543

5 V –> 15543 PLC UNITS

100 ⁰C – 15543 PLC UNITS





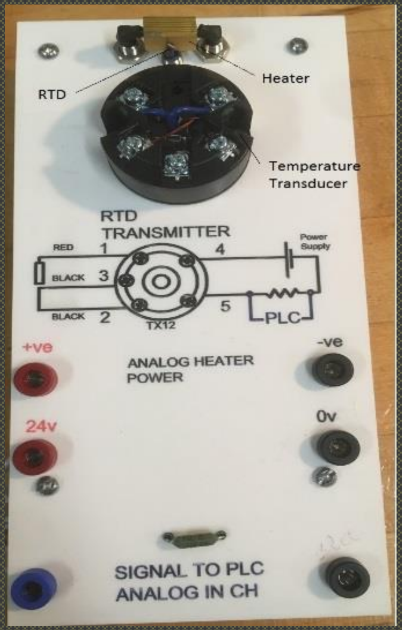
k = 100/(15543-3108) = 0.008 b = 100 – 15543*0.008 = -24.8 Temperature = k*Ch0 + b; k = 0.008; b = -24.86

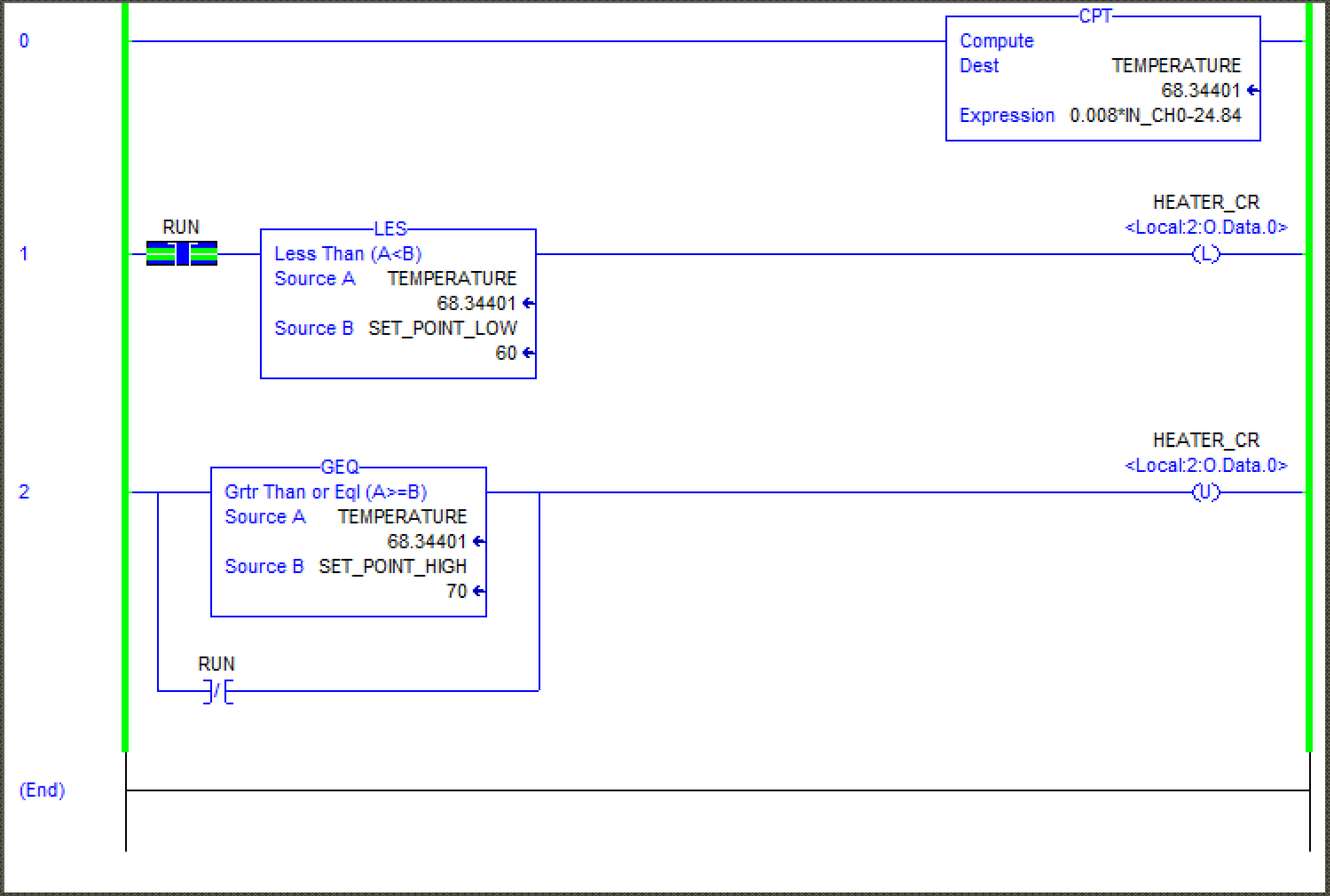
2 SETPOINTS ON-OFF CONTROL

Example

Temperature = k*Ch0 + b; k = 0.008; b = -24.86

$\boxed{\text{Temperature} = 0.008*\text{Ch0} - 24.86}$

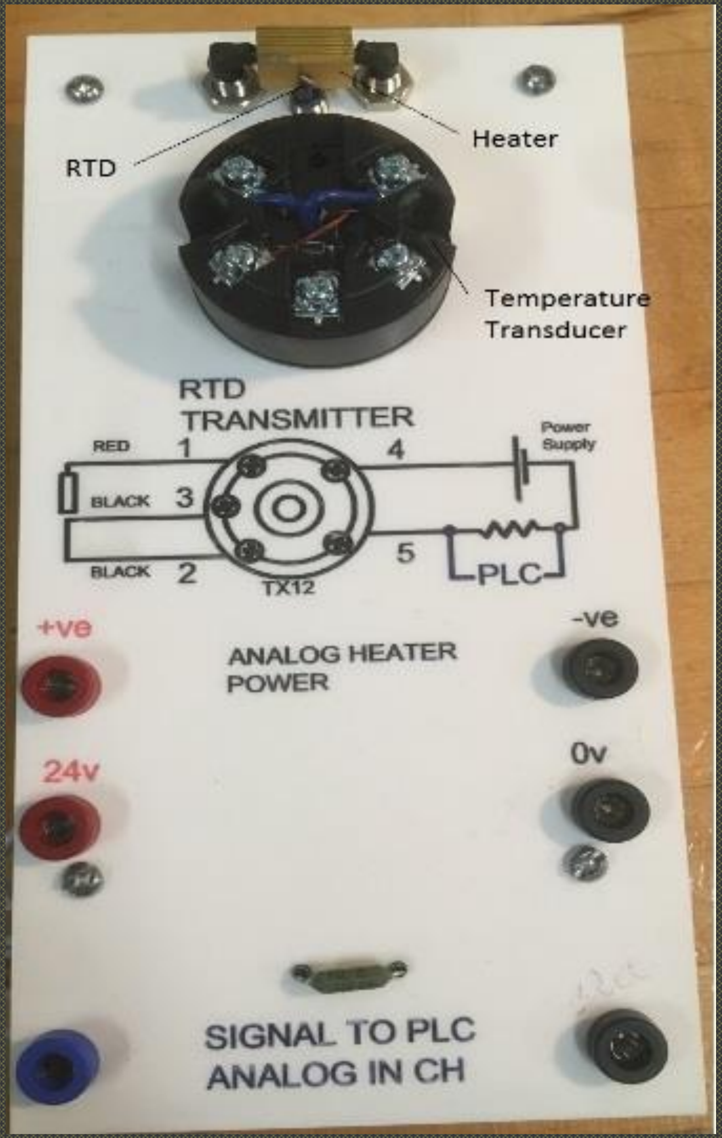


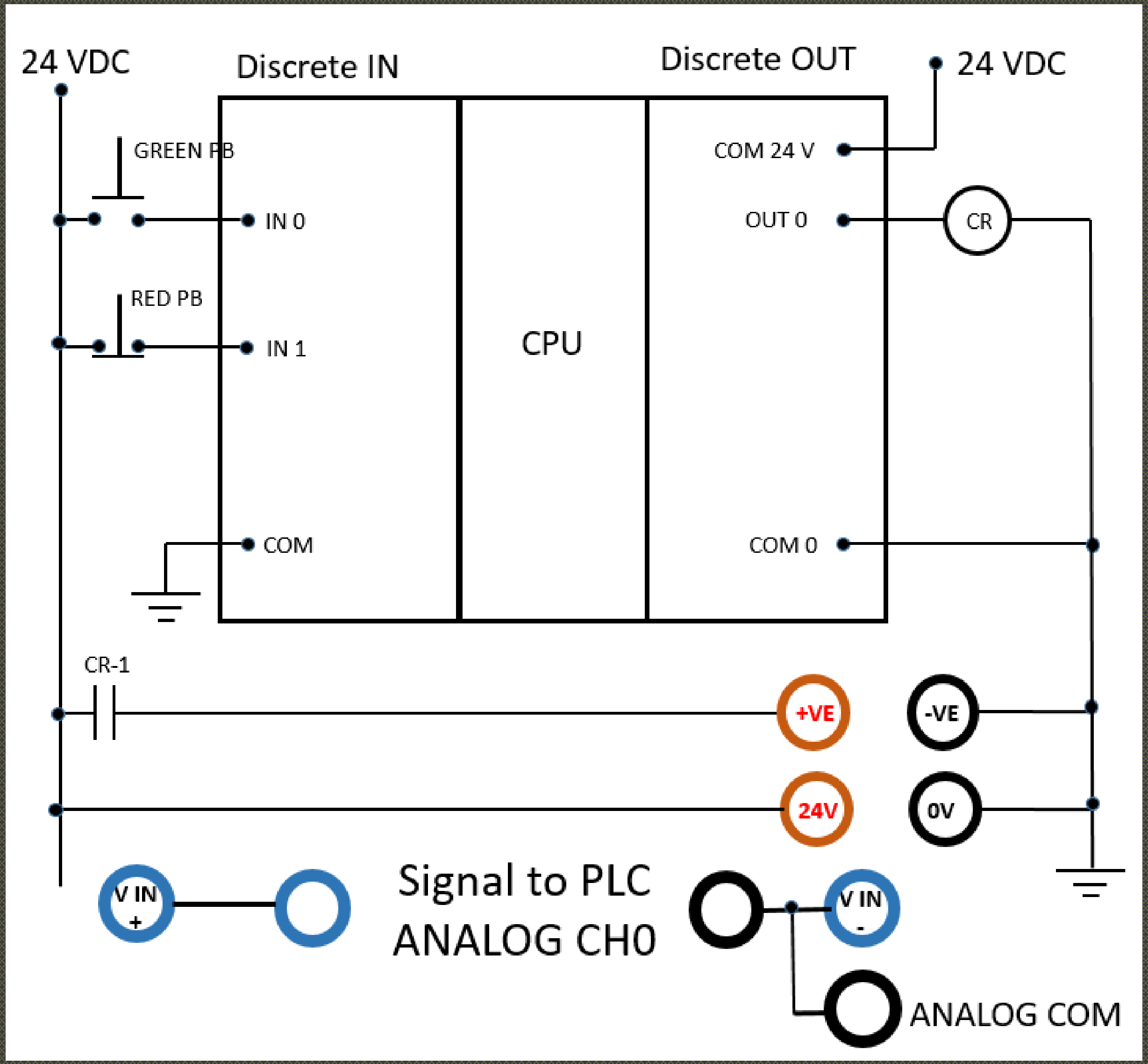


HUMBER

Wiring Diagram

Example



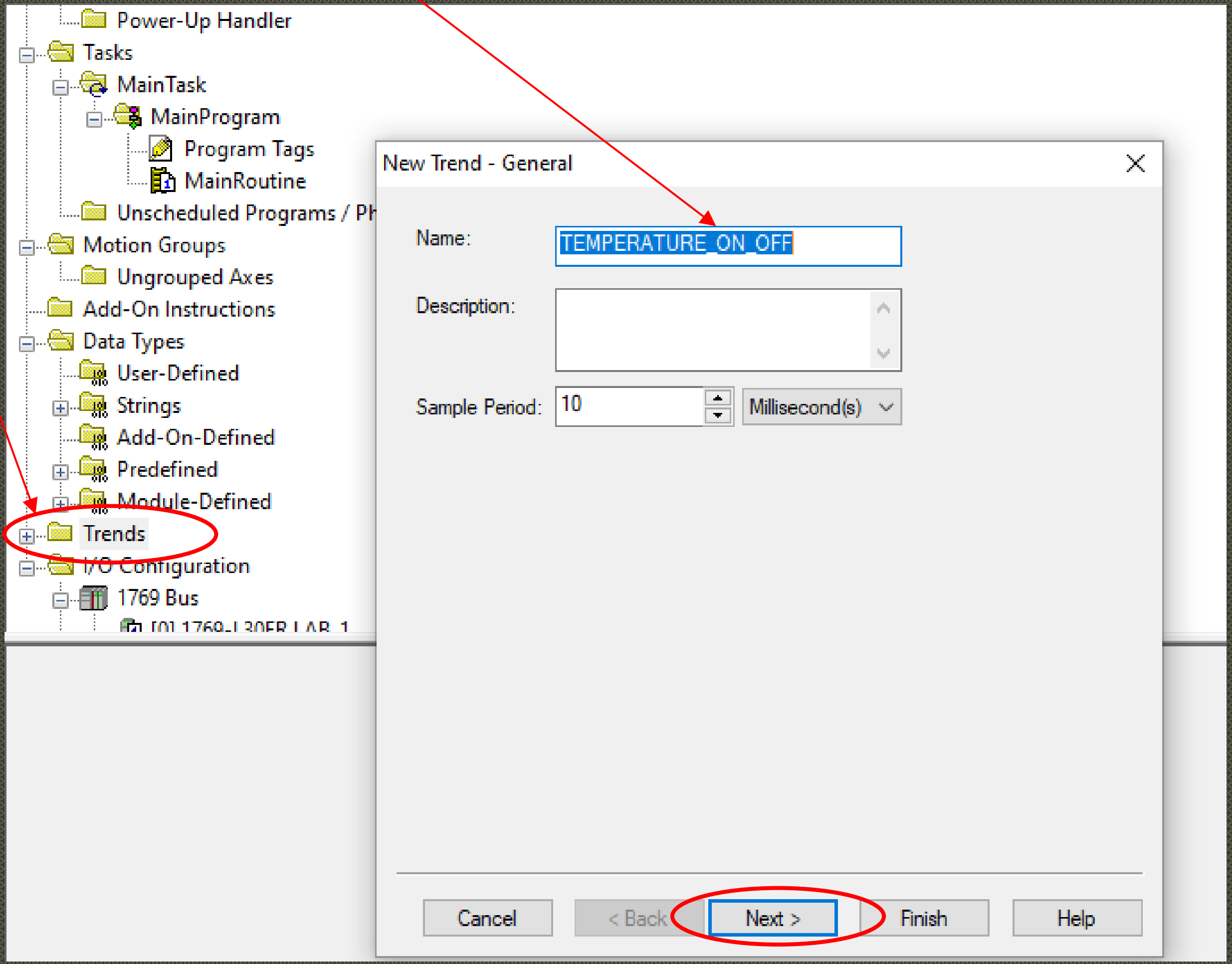


HUMBER

TRENDS

Trends are helpful to analyse the system's behaviour and make corrections of its performance. In order to create trends for the speed control system in this assignment using RSLogix 5000, follow the steps below:

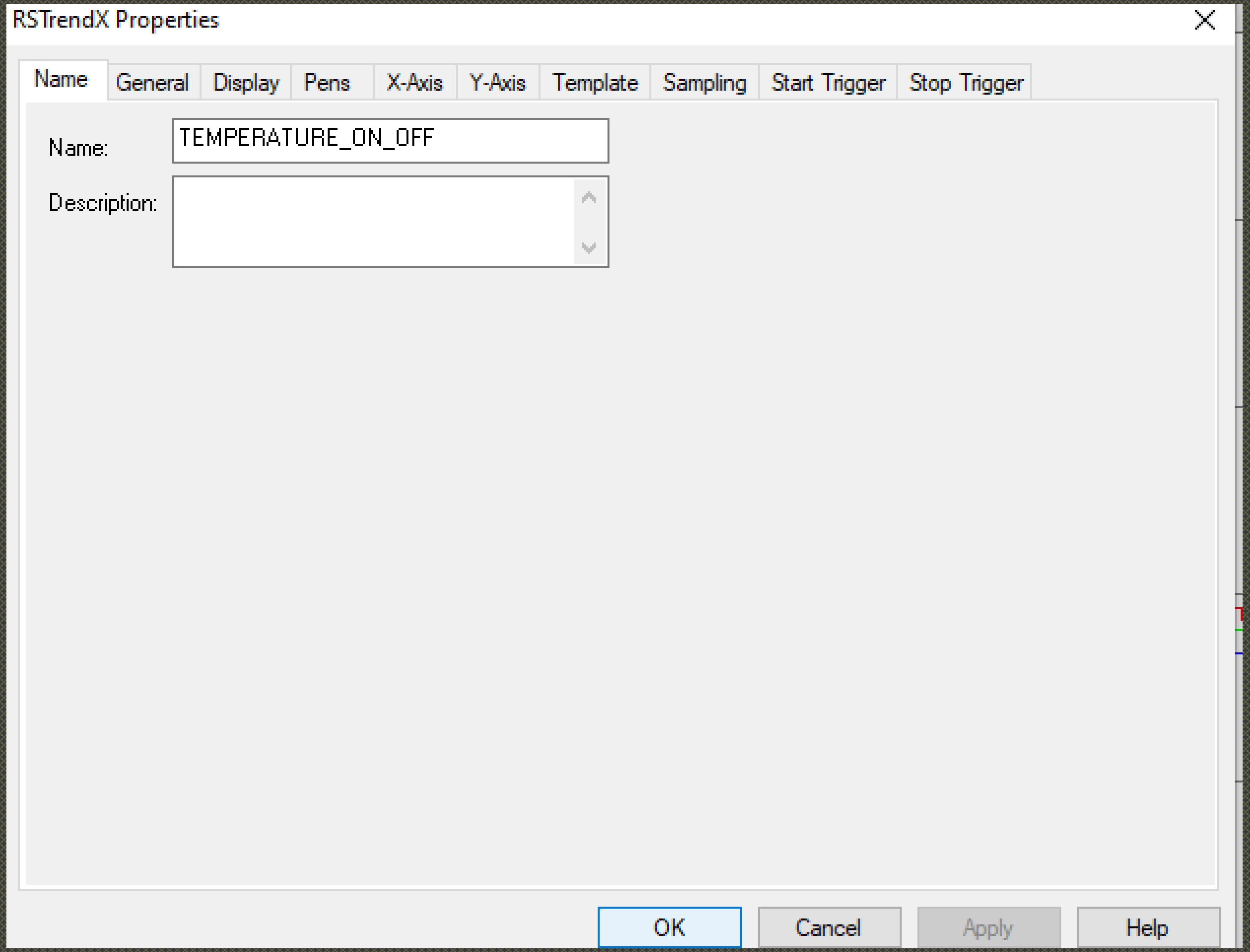
- Create a new **TEMPERATURE_ON_OFF** trend to monitor the Actual Temperature and both Set points. Right click on Trends and type the Trend Name. Click on NEXT.



HUMBER

TRENDS

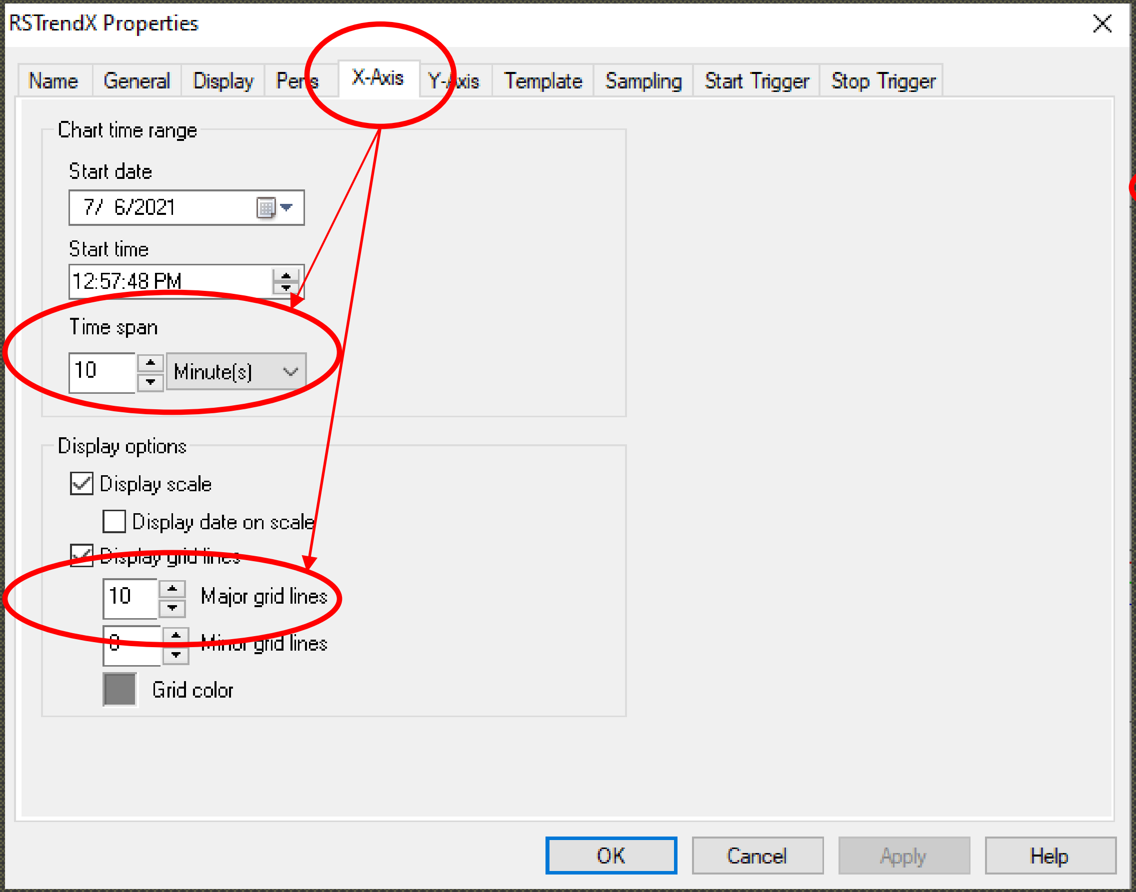
- Right click on the chart and select Chart Properties;



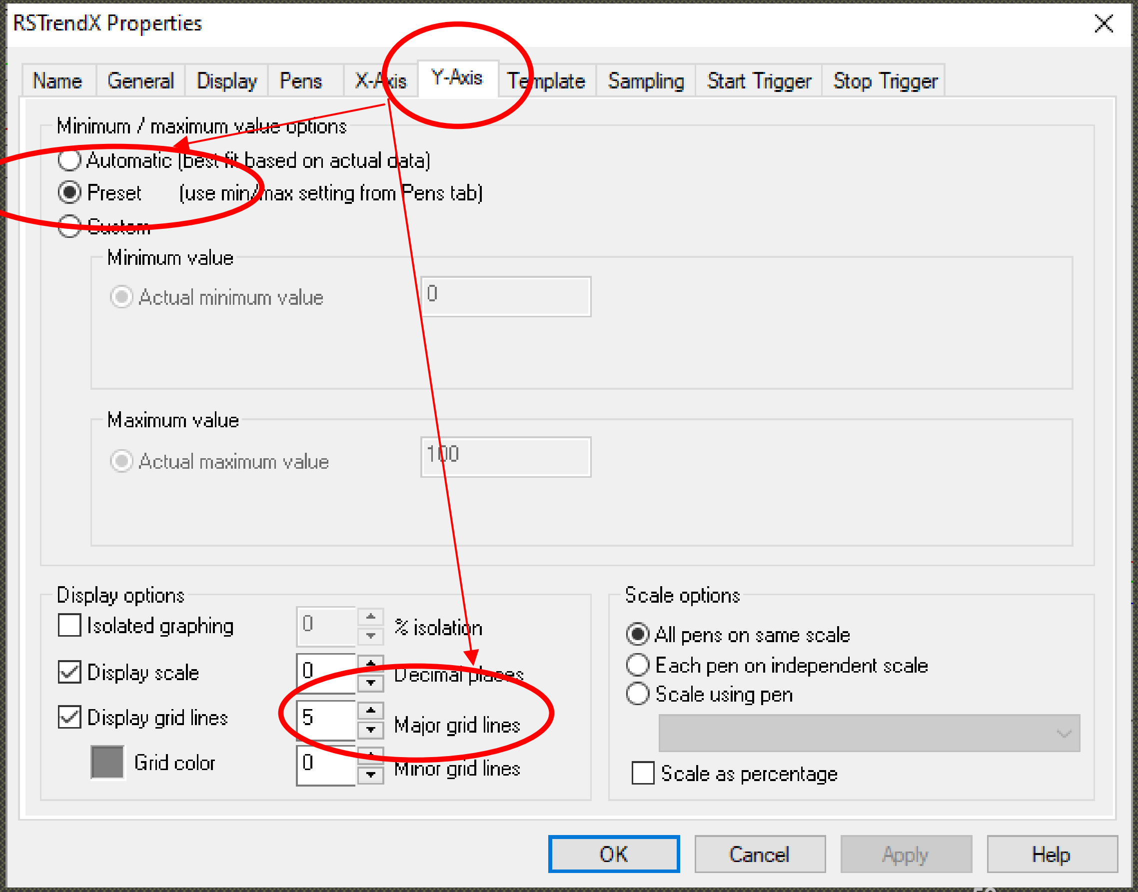
HUMBER

TRENDS

- For **X – Axis**;

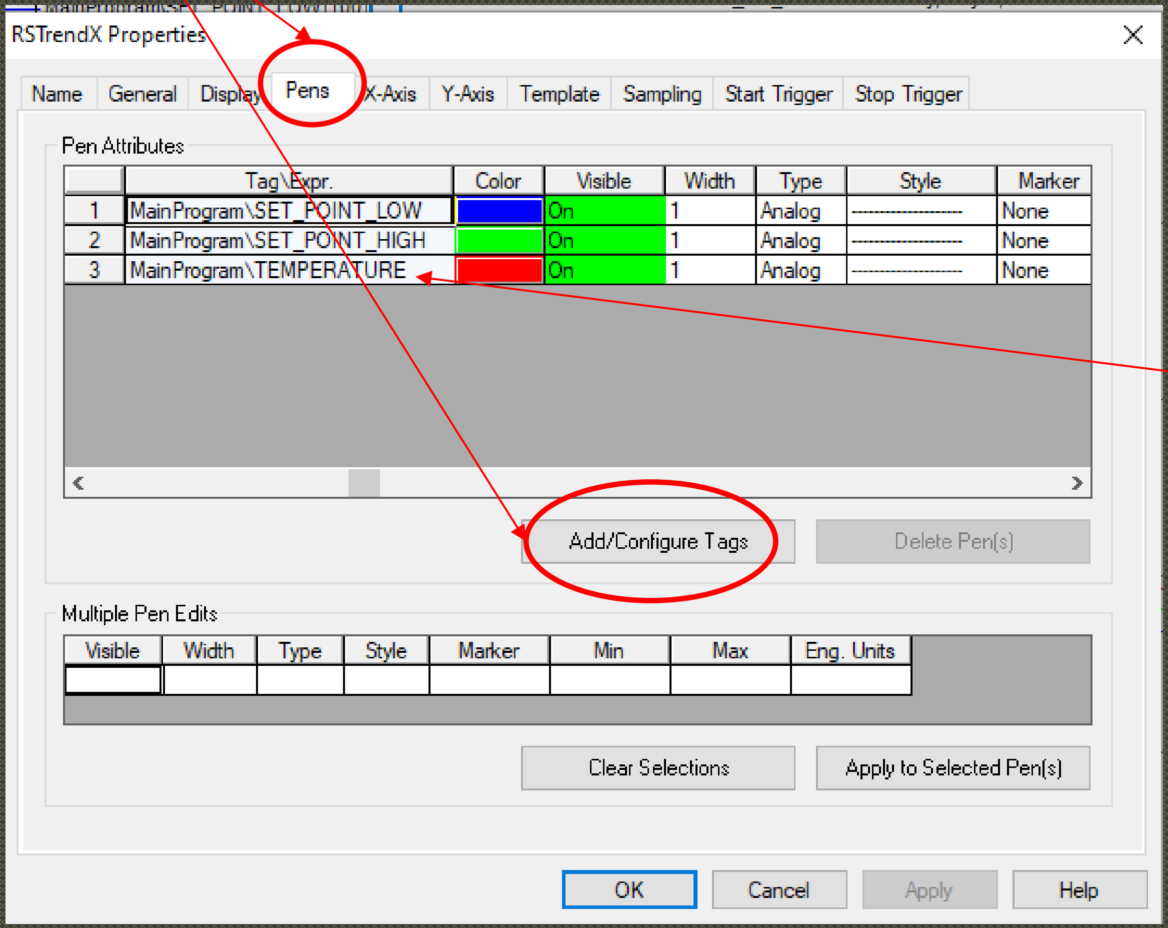


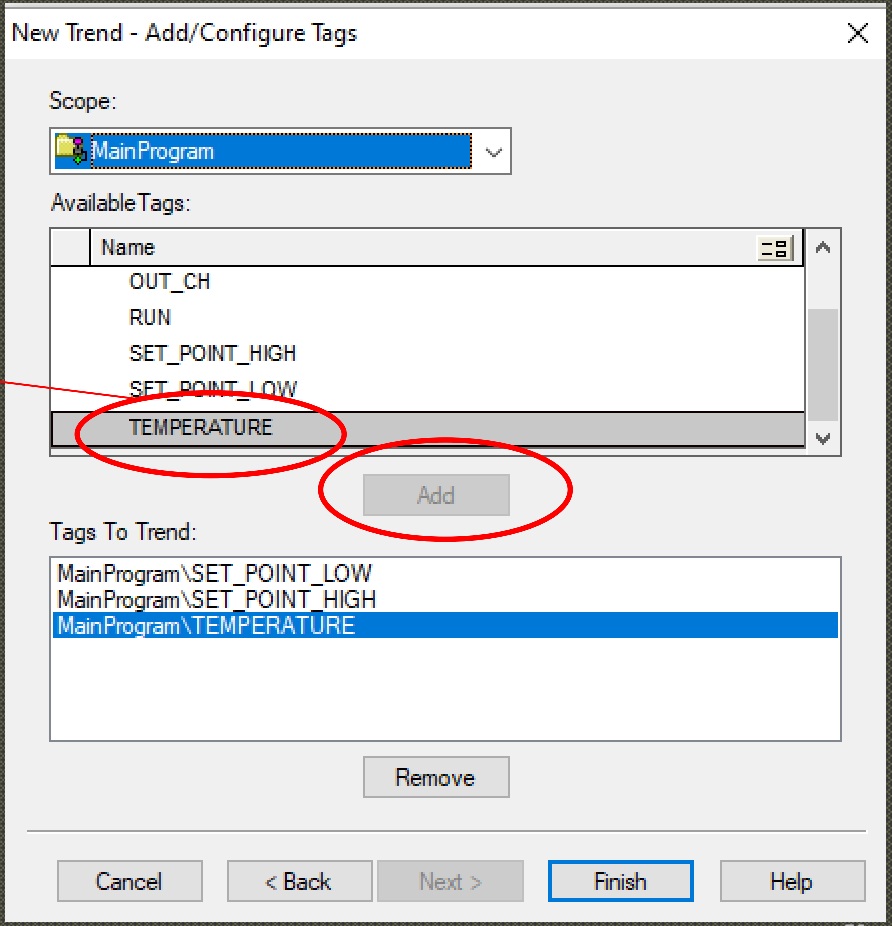
- For **Y – Axis**;



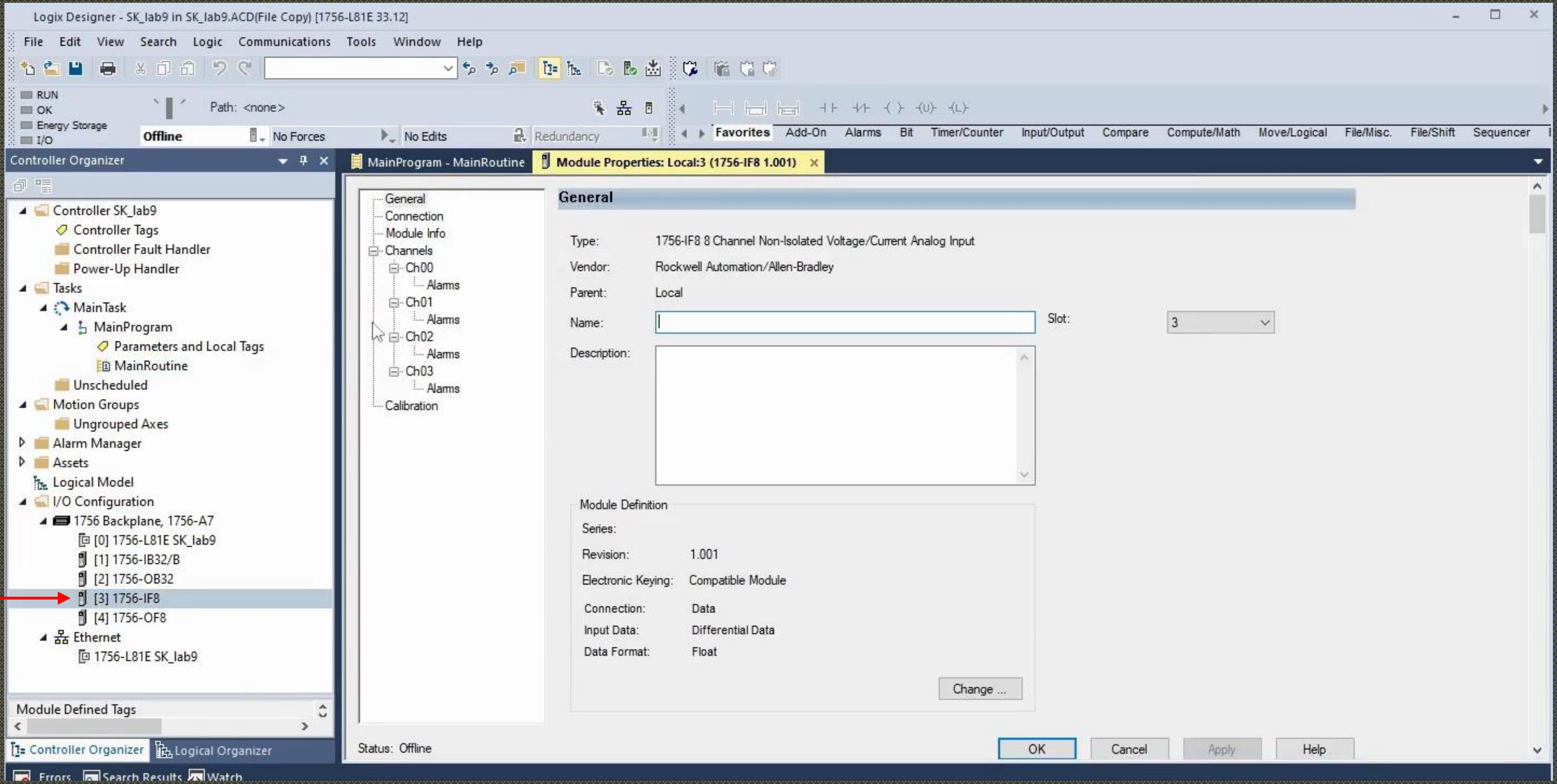
TRENDS

- For **Pens**, Use Add/Configure Tags button to create three tags: **SET_POINT_LOW**, **SET_POINT_HIGH** and **TEMPERATURE**.



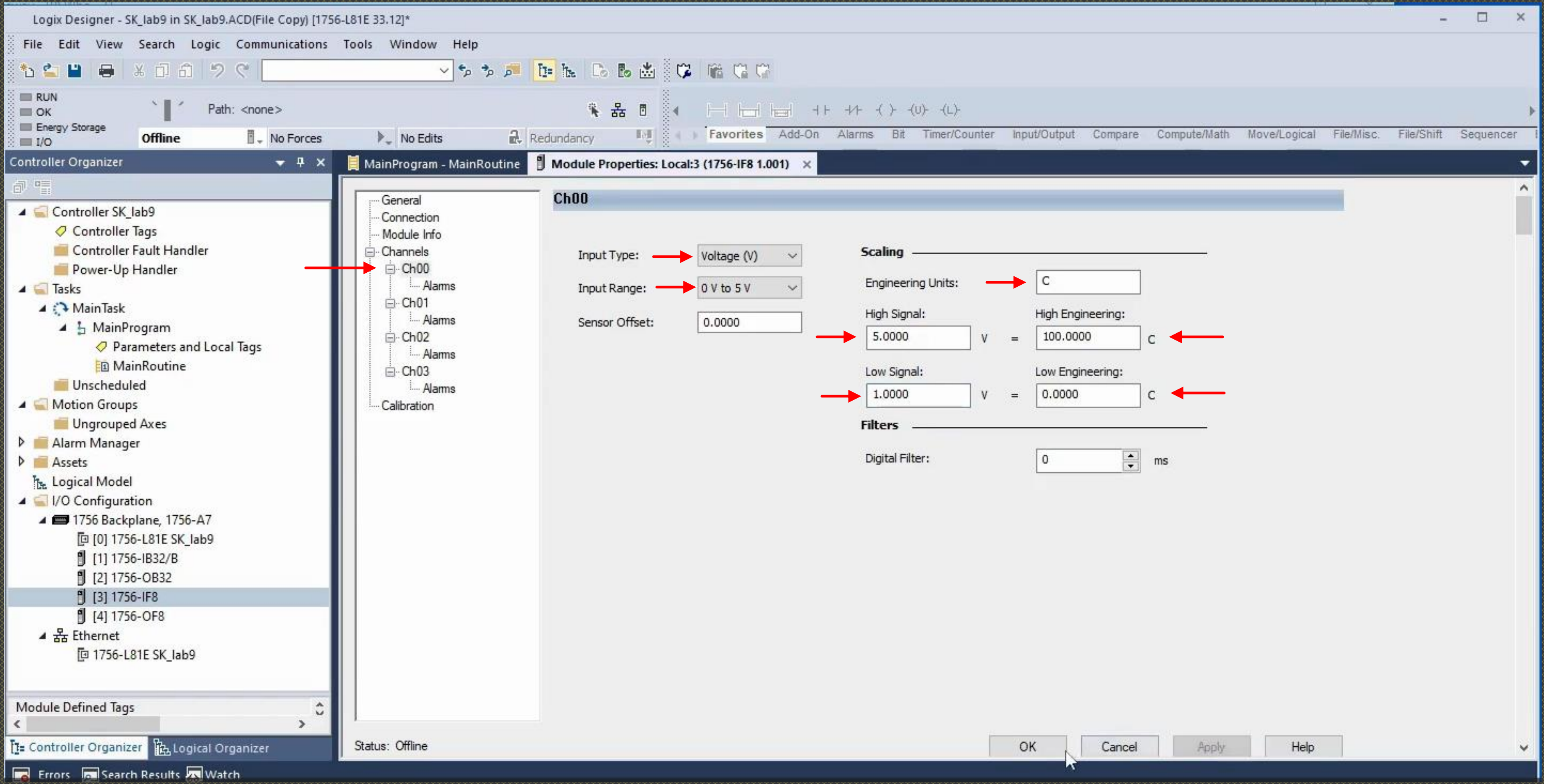


1756 – IF8



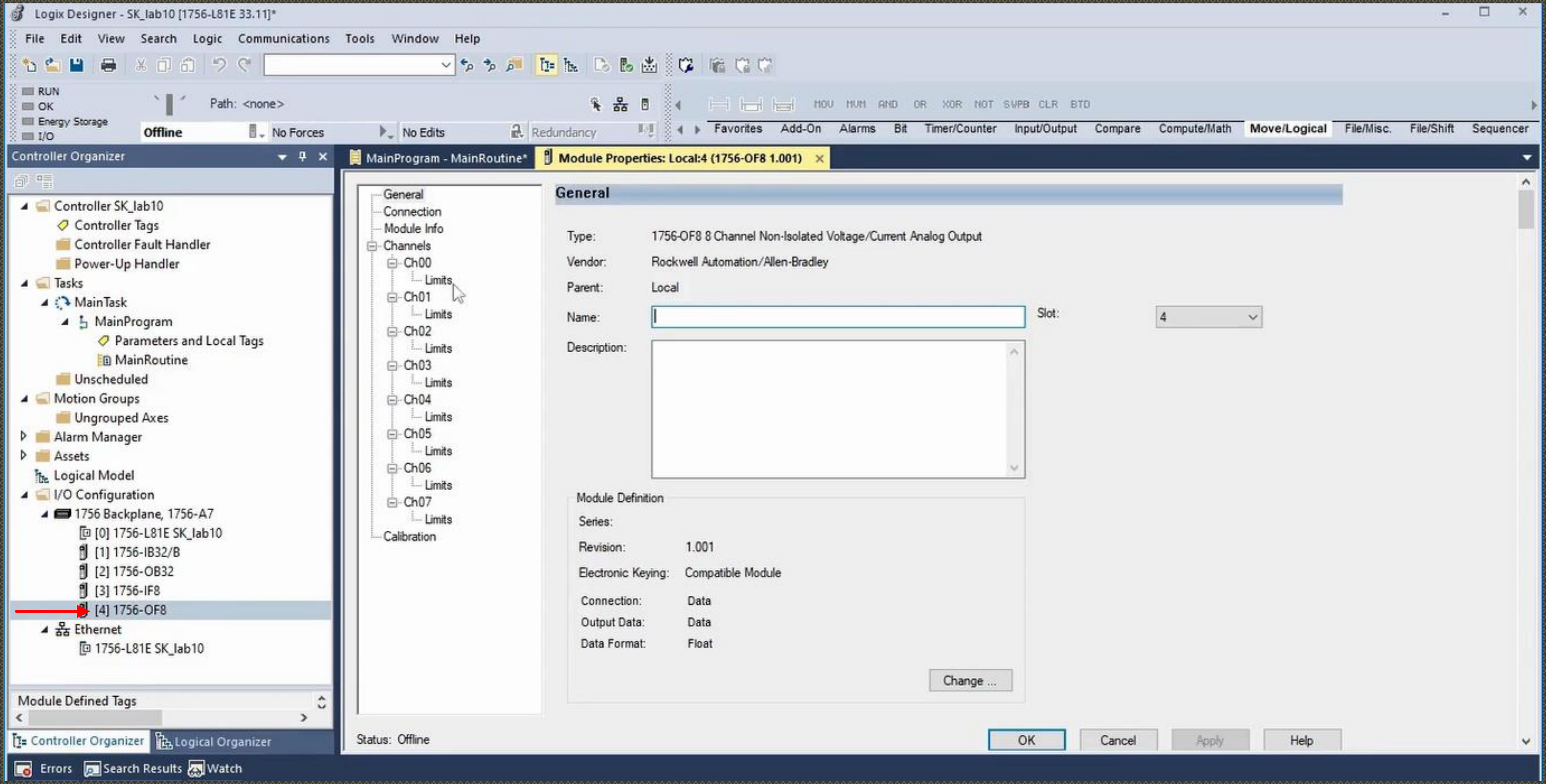
HUMBER

1756 – IF8

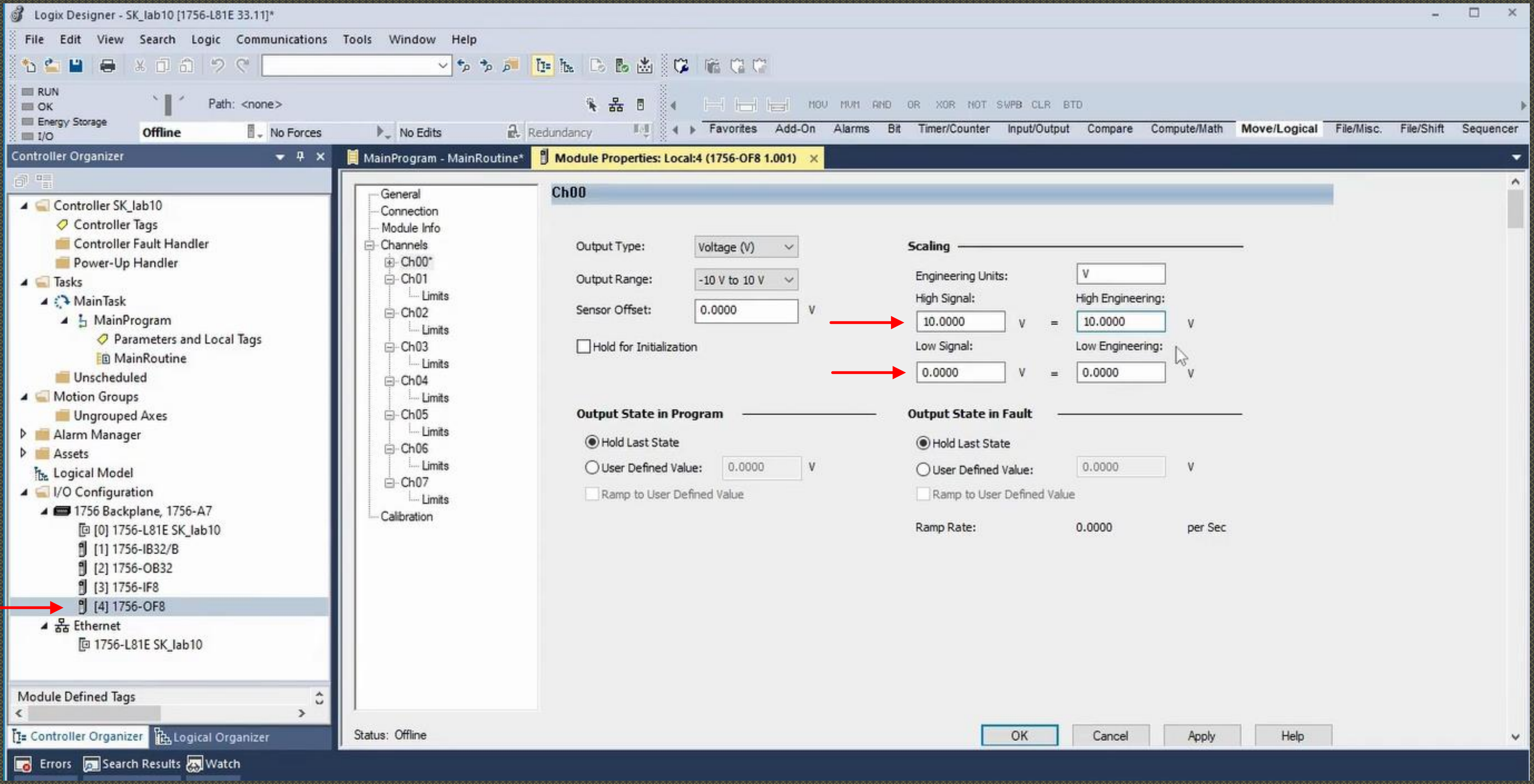


HUMBER

1756 – OF8


Logix Designer - SK_lab10 [1756-L81E 33.11]*
File Edit View Search Logic Communications Tools Window Help
RUN
OK
Energy Storage
I/O
Path: <none>
Offline
No Forces
No Edits
Redundancy
Favorites Add-On Alarms Bit Timer/Counter Input/Output Compare Compute/Math Move/Logical File/Misc. File/Shift Sequencer
Controller Organizer
Controller SK_lab10
Controller Tags
Controller Fault Handler
Power-Up Handler
Tasks
MainTask
MainProgram
Parameters and Local Tags
MainRoutine
Unscheduled
Motion Groups
Ungrouped Axes
Alarm Manager
Assets
Logical Model
I/O Configuration
1756 Backplane, 1756-A7
[0] 1756-L81E SK_lab10
[1] 1756-IB32/B
[2] 1756-OB32
[3] 1756-IF8
[4] 1756-OF8
Ethernet
1756-L81E SK_lab10
Module Defined Tags
Controller Organizer
Logical Organizer
Errors
Search Results
Watch
MainProgram - MainRoutine*
Module Properties: Local:4 (1756-OF8 1.001)
General
Connection
Module Info
Channels
Ch00
Limits
Ch01
Limits
Ch02
Limits
Ch03
Limits
Ch04
Limits
Ch05
Limits
Ch06
Limits
Ch07
Limits
Calibration
General
Type: 1756-OF8 8 Channel Non-Isolated Voltage/Current Analog Output
Vendor: Rockwell Automation/Allen-Bradley
Parent: Local
Name:
Slot: 4
Description:
Module Definition
Series:
Revision: 1.001
Electronic Keying: Compatible Module
Connection: Data
Output Data: Data
Data Format: Float
Change ...
Status: Offline
OK
Cancel
Apply
Help

1756 – OF8



HUMBER

PROGRAMMABLE LOGIC CONTROLLERS

Analog Signal Processing, Scaling

Erickson, K. (2016) Programmable logic controllers: An emphasis on design and application (3rd edition). Rolla MO: Dogwood Valley Press.

Chapter 4.3
Chapter 7.2

HUMBER

Thank you!

Discussions?