PROGRAMMABLE LOGIC CONTROLLER

Dr. Ir. Yeffry Handoko Putra, M.T

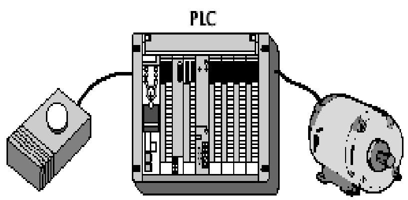
PLC TERMINOLOGY AND APPLICATIONS

Programmable Logic Controller, also called a ***PLC*** or ***programmable controller***, is a computer-type device used to control equipment in an industrial facility.

PLC Applications: Process Control, HVAC Control, Conveyor Systems, Food Processing Machinery, Auto Assembly Lines, etc. (Regulatory Control as well as Sequential Control Systems)

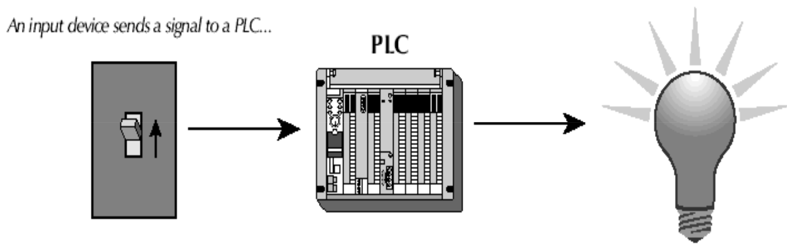
SYSTEM WIRING COMPARISON

Traditionally hardwired system

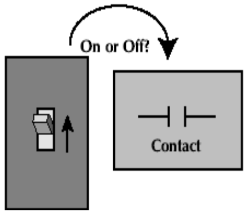


PLC System Wiring

PLC SYSTEM INPUT/OUTPUT

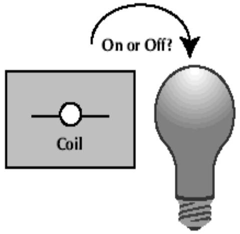


RELAY AND COIL

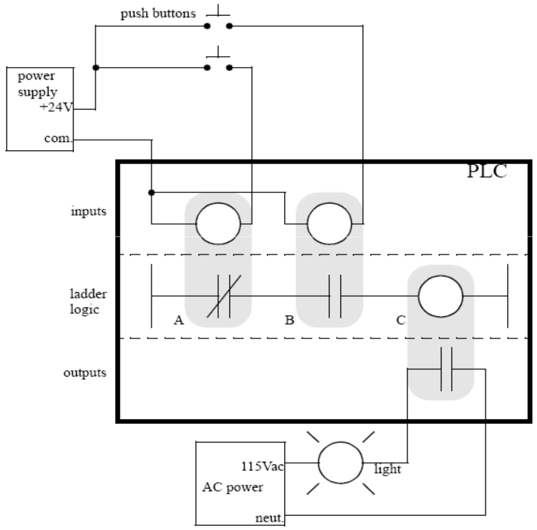


A contact is a computer code that monitors the status of an input…

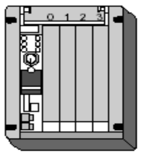
…A coil is a computer code that monitors the status of an output.



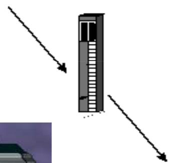
PLC SYSTEM WITH RELAYS



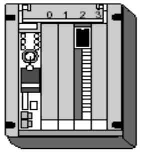
SYSTEM RACK COMPONENTS



A rack is an enclosure with slots…



…into which I/O modules…

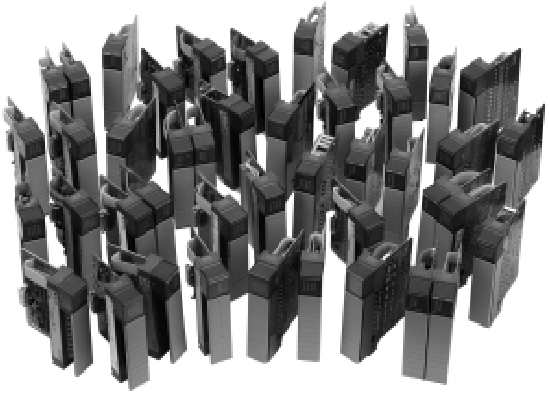


…are installed.

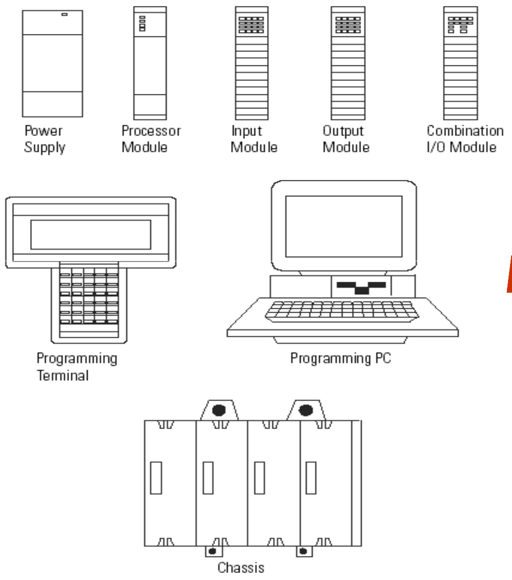
PLC MODULES

I/O MODULES

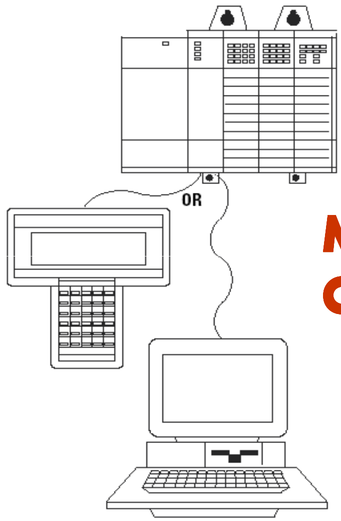
PROCESSOR



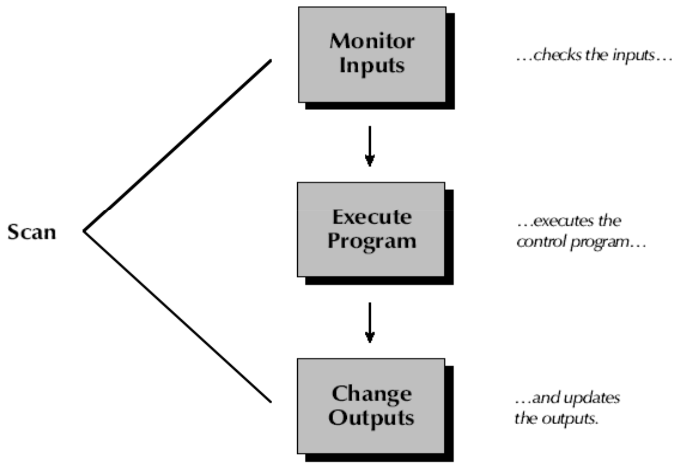
PLC MODULES



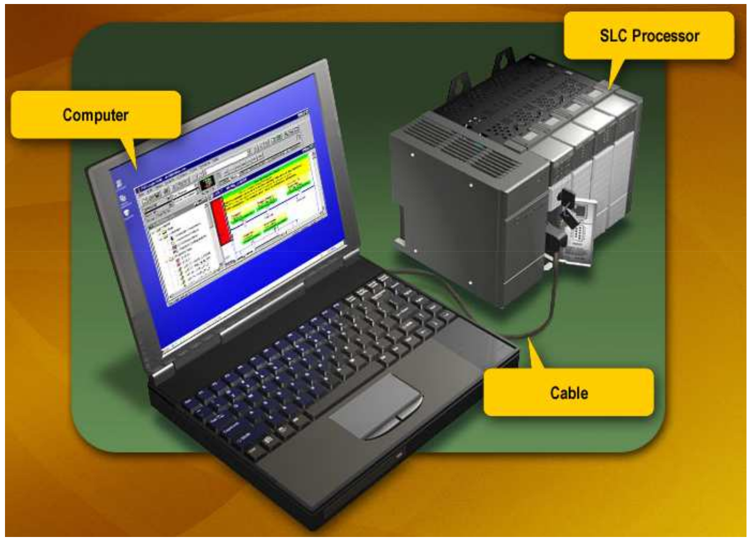
MODULAR CONTROLLER



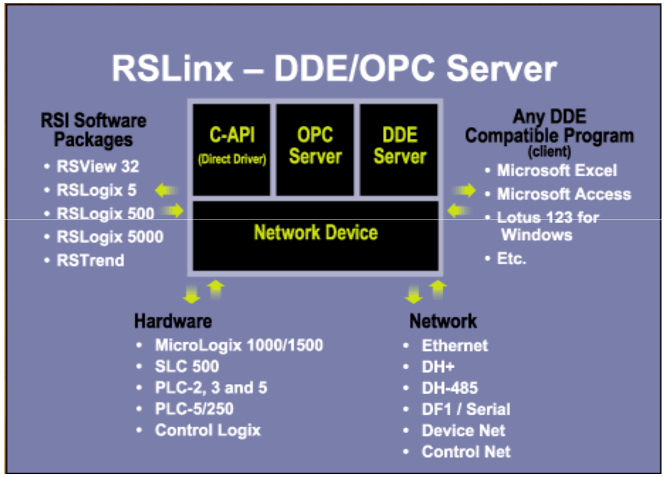
PLC PROGRAM FLOW



DIRECT PC-PLC LINK



PC-PLC LINK SOFTWARE TOOL



PC-PLC LINK SOFTWARE TOOL

PLC PROGRAM UP-LOAD

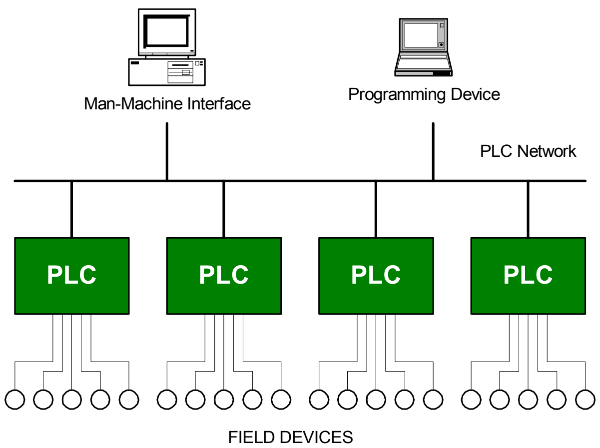
PLC PROGRAM DOWN-LOAD

ON-LINE EDITING

ON-LINE EDITING FUNCTIONS

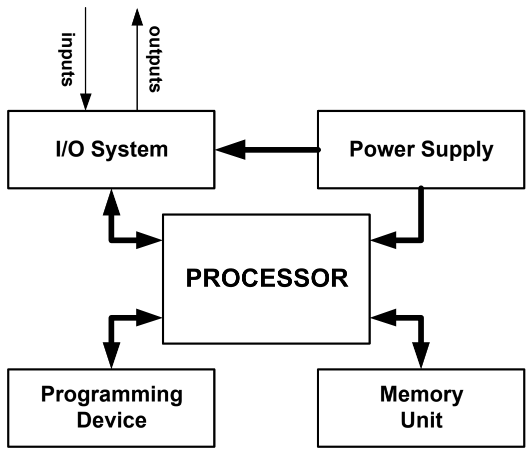
- Inserting Rungs
- Replacing Rungs
- Deleting Rungs
- Documenting Program
- Modifying Addresses or Instruction Types

A Picture of PLC System



Basic Components of PLC

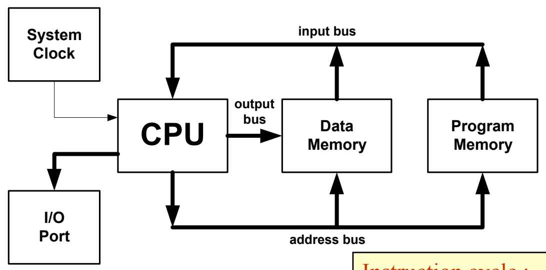
- Processor
- Memory
- I/O Interface
- Programming Device & Languages
- Power Supply



How does a PLC work ?

- Examine the status of inputs and outputs
- Controls some process or machine through outputs using some control logic
- This control logic is executed periodically by the processor in a predetermined sequential order
- User can change the control logic using a programming language and it is stored in the program memory

Processor & Memory



- 80386/80486 microprocessor based
- Random Access Memory (RAM)
- Electrically Erasable Programmer Read Only Memory (EEPROM)

Instruction cycle :
- Fetch the next instruction from program memory
- Place it in the instruction register in the CPU
- Increment the program counter by one
- Execute the instruction

Input/Output Systems

- Discrete Inputs/Outputs
- Analog Inputs/Outputs
- Special Purpose Modules
- Intelligent Modules
- Communication Modules

Discrete Signal Types

- Selector switches
- Temperature switches
- Flow switches
- Level switches
- Valve position switches
- Starter auxiliary contacts
- Pushbuttons
- Motor starter contacts
- Limit switches
- Pressure switches
- Hand switches
- Proximity switches
- Relay contacts
- Photoelectric sensors

- Anunciators
- Alarm lights
- Electric control relays
- Electric fans
- Indicating lights
- Electric valves
- Alarm horns
- Solenoid valves
- Motor starters
- Heater starters

Discrete Input Modules

- **AC and DC Discrete Input Module**
  - Electrical isolation between the field device (power) and the controller (logic)
  - Common return line connection
  - ACI-XX where XX is voltage (120 and 220 VAC)
  - DCI-XX where XX is voltage (12, 24, and 48 VDC)
- **TTL**
  - TTL-compatible devices (solid-state controller and sensing instruments)
  - external +5 V dc power supply
- **Isolated Input Module**
  - Separated return lines
  - IACI-XX or IDCI-XX

Discrete Output Module

- **AC Output Module**
  - AC loads (using Triac/ SCR as switch)
  - RC snubber protection
  - Peak voltage limiter (metal oxyde varistor - MOV)
  - Fuse protection
  - External switching voltage
- **DC Output Module**
  - DC loads (using power transistors)
  - Fuse protection
- **TTL Output Module**
  - TTL-compatible output devices (seven segment LED, IC, and +5 Vdc logic-based devices)
  - External +5 Vdc power supply
- **Isolated AC Output Module**
- **Dry Contact**
  - Normally open (NO)
  - Normally closed (NC)

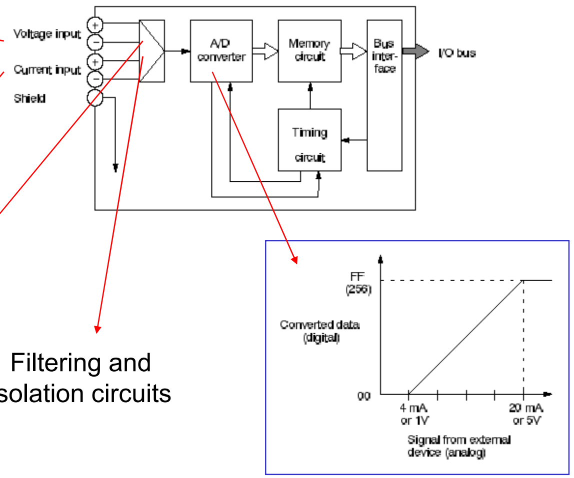
Analog Devices

- Flow transmitters
- Pressure transmitters
- Temperature transmitters
- Analytical transmitters
- Position transmitters
- Potentiometers
- Level transmitters
- Speed instruments
- Electric motor drives
- Analog meters
- Chart data recorders
- Process controllers
- Current-to-pneumatic transducers
- Electrical-operated valve
- Variable speed drives

Analog Input Module

Single-ended
Differential

Current (0-20 mA)
Voltage (1-5 V)



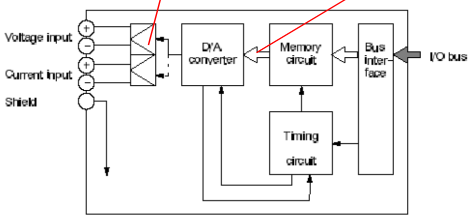
Very high input impedance to prevent signal loading

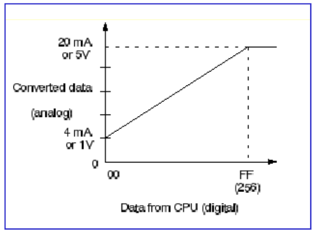
Filtering and isolation circuits

Analog Output Module

Isolation between the output circuit and the logic circuit is provided through optical couplers

Receive data from the CPU which is translated into a proportional voltage/current to control an analog field device





Special Purpose Modules

- Binary Coded Decimal (BCD) Modules
  - Provide parallel communication between the processor and input/output devices
- Encoder/Counter Input Module
  - Used for operations that require direct high speed encoder input into a counter
- Pulse Counter Input Module
  - Used to interface with field instruments that generate pulse such as positive displacement (PD) flowmeters and turbine type flowmeters

Intelligent Module

- Can perform complete processing functions, independent of the CPU and the control program scan
- Thermocouple Input Module
  - Designed to accept inputs directly from a thermocouple
  - Provides cold junction compensation
- Stepping Motor Module
  - Generates a pulse train that is compatible with stepping motor translators that represent distance, speed, and direction commands
- Control Loop Module (PID module)
  - Used in continuous closed-loop where the proportional-integral-derivative (PID) control algorithm is required

Communication Module

- **ASCII**
  - Used to send and receive alphanumeric data between peripheral equipment and the controller
- **Universal Remote I/O Link**
  - Allows I/O subsystems to be remotely located from the processor (1000 ft to several miles)
- **Serial Communications Module**
  - Used to communicate between the programmable controller and an intelligent instrument with a serial output
  - RS-232C, RS-422, or RS-485 communication link
- **PCMCIA Interface Card**
  - Allows communications between PLC or data highway and notebook PC
- **Ethernet**
  - Designed to allow a number of PLC and other computer-based devices to communicate over a high speed plant local area network
- **Fiber Optic Converter**
  - Transform electrical signals and transmit these signals through fiber optic cables

Designing I/O Systems : Electrical

- **Input voltage rating**
  - Lists the magnitude and type of signal the module will accept
- **Input current rating**
  - The minimum input current required at the module's rated voltage that the field device must be capable of supplying to operate the input module circuit
- **Input threshold voltage**
  - The voltage at which the input signal is recognized as being ON/true
- **Output voltage rating**
  - The magnitude and type of voltage that can be controlled within a stated tolerance
- **Output current rating**
  - The maximum current that a single output circuit in a module can safely carry under load
- **Output power rating**
  - The maximum total power that an output module can dissipate with all output energized
- **Backplane current requirements**
  - Lists the current demand that a particular I/O module internal circuitry places on the rack power supply

Designing I/O Systems : Mechanical

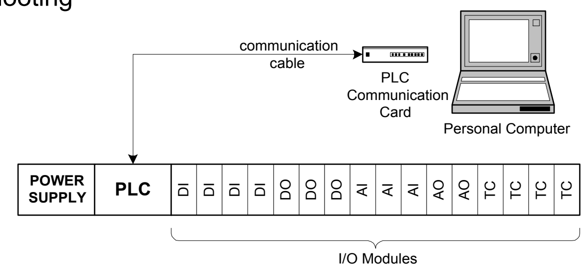
- Number of I/O points per module
- Number of wires
- Wire size specification
  - Number of conductors
  - Largest wire gage that the I/O terminal points will accept
- Ambient temperature rating
  - Based on the heat dissipation characteristics of the circuit components inside the I/O module
  - 0 - 60 C typical
- Humidity rating
  - 5 - 95 % typical relative humidity

IEC1131-3 Standard Languages

- Ladder Diagram (LD)
  Uses a standardized set of ladder logic symbols
- Sequential Function Charts (SFC)
  Blocks connected together like a circuit diagram
- Functional Block Diagram (FBD)
  A graphical language used to describe sequential operations
- Instruction List (IL)
  Low level language
- Structured Text (ST)
  High level structured language designed for automation processes

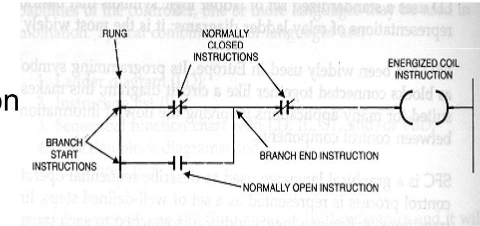
Programming Devices

- Used to enter, store, and monitor the PLC software
- PC-based system
- Connected to the PLC only during :
  - programming & testing
  - startup
  - troubleshooting



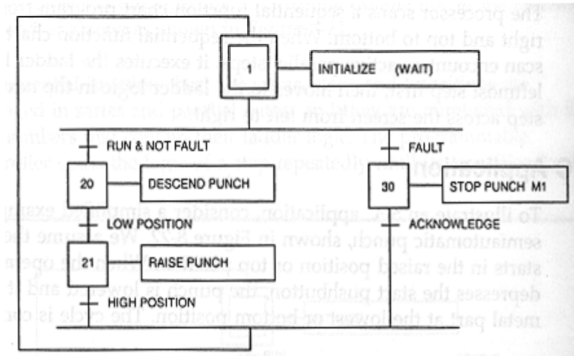
Ladder Diagram (LD)

- Representations of relay ladder diagrams
- The most widely used
- Composed of six categories of instructions
  - Relay type
  - Timer/counter
  - Data manipulation
  - Arithmetic
  - Data transfer
  - Program control



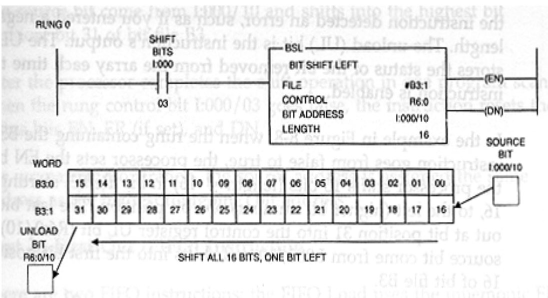
Sequential Function Charts (SFC)

- Represented as a set of well-defined steps, linked by transitions
- Actions within the steps are detailed by using the Structured Text language (ST)



Functional Block Diagram (FBD)

- Describes a relationship or function between input and output variables
- Well suited for many applications involving the flow of information or data between control components
- Allows the programmer to build complex control procedures
- Extensive function blocks library is provided



Instruction List (IL)

- Very effective for small simple applications or for optimizing parts of an application
- A list of low level instructions
- Consists of standardized operators
  - Data manipulation
  - Arithmetic
  - Branch instruction

Example :

Start :	LD	IX1	(* load input IX1, start pushbutton *)
	ANDN	MX5	(* AND with NOT of MX5)
	ST	QX2	(* store output QX2 to start motor *)

Structured Text (ST)

- Used mainly to implement complex procedures
- Default language for the description of the actions within steps and conditions attached to the transitions of the SFC
- Statement types :
  - assignment
  - subprogram or function call
  - “C” function block call
  - Selection (if, then, else, case, etc)
  - Iteration (for, while, repeat, etc)
  - Control (return, exit, etc)
  - Special

```
imax:=max_ite;
cond:=X12;
if not(cond(*alarm*))
then return;
end_if;

for i:=1 to max_ite do
if i<>2 then
SPcall();
end_if;
end_for;
```

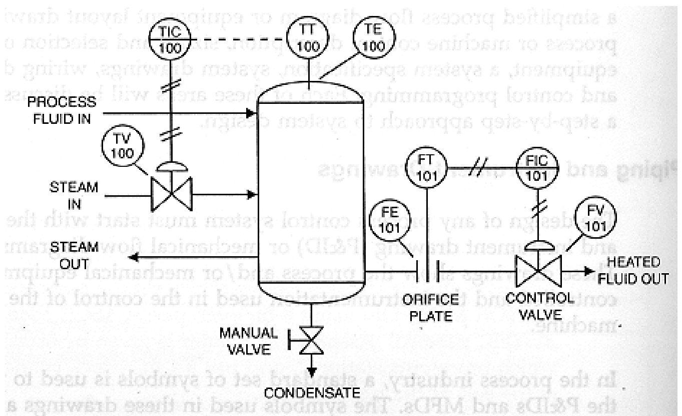

PLC System Design

1. Process Description
2. I/O Sizing
3. Memory Sizing
4. Selecting programming language
5. Peripheral requirements
6. System drawing and I/O wiring diagrams
7. System programming

Process Description

- States the purpose and the steps of the process/plant operation
- Process Description is the most important step in the design process
- Bridge of communications between the user and the designer
  - Piping and Instrument Diagram (P&ID) and Mechanical Flow Diagrams (MFDs)
  - Simplified drawing that shows only the equipment and instrumentation controlled or measured by the PLC is required
  - This drawing will be used to show the status of the process in each step or state to aid in the programming of the system

Piping & Instrument Diagram (P&ID)



I/O Sizing (1)

- Estimate the number of input/output (I/O) required to control the process
  - Obtain the number of device from P&ID diagram
  - Add the number of I/O points from each devices to obtain total I/O points
- Consider different types of I/O :
  - Discrete AC/DC : Limit switches, push buttons, selector switches, solenoid, etc.
  - TTL : solid state displays and electronic instrumentation
  - Analog : Level transmitter, pressure transmitter, etc.
  - Encoders

I/O Sizing (2)

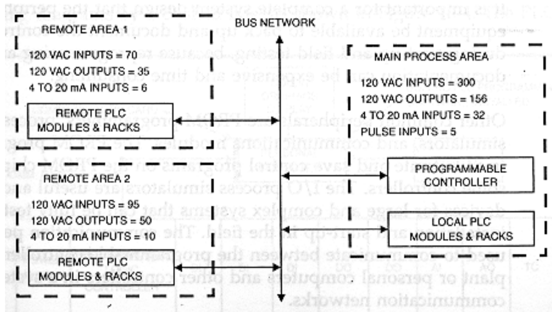
- Select PLC size :
  - *Micro* : up to 32 I/O points
  - *Small* : 32 - 256 I/O points
  - *Medium* : 256 - 1024 I/O points
  - *Large* : > 1024 I/O points
- Estimate the number of I/O module
  - Classify each type of I/O points (discrete, analog, isolated, TTL, dry contact)
  - Select suitable I/O modules from PLC manual and obtain the number of I/O points per module
  - The number of each I/O module type required = *[Total I/O points] / [Number of I/O points per module]*
- Add spare and future expansions (10-20% spare capacity)

Memory Sizing

- The amount of memory required depends on :
  - control program complexity
  - the number of I/O points
- Precise (*almost impossible*) method to determine memory size :
  - Write out the control program
  - Count the number of instructions used
  - Multiply this count by the number of words used per instructions (obtained from PLC programming manual)
  - Add the amounts of memor used by executive programs and the processor overhead
- **Practical method** :

  Total memory = 10 x [the number of I/O points]

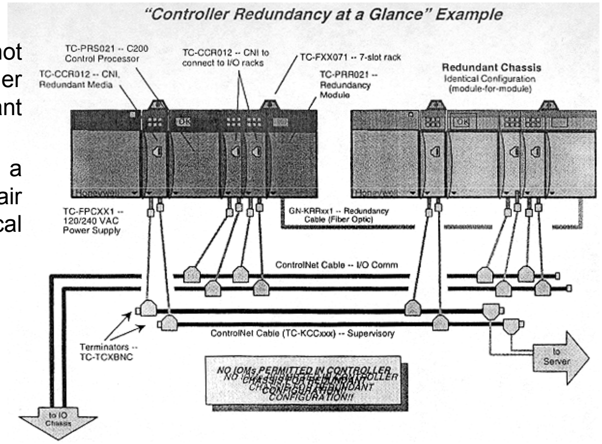
I/O and Memory Sizing Example



- I/O Points
  - Remote Area 1 : I/O = 70 + 35 + 6 = 111
  - Remote Area 2 : I/O = 95 + 50 + 10 = 155
  - Main Process Area : I/O = 300 + 156 + 32 + 5 = 493
  - Total I/O points = 759
  - Spare points = 10% x 759 = 76
- PLC size = Medium-sized (1024 I/O points max.)
- Memory size = 10 x 759 = 7590 or 8K

Redundant Configuration

- I/O modules are not permitted in either chassis of a redundant chassis pair
- Both chassis of a redundant chassis pair must have identical slot-for-slot configurations



Selecting Programming Language

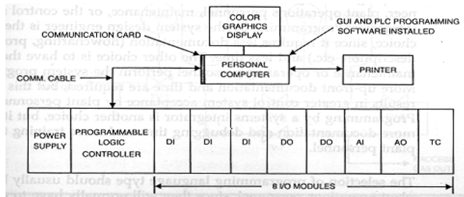
- Most PLCs offer the basic ladder logic instructions plus a combination of the other types of languages
- Programming language selections is depends on :
  - Complexity of the control system
  - Background knowledge of the control system programmers and operators

Peripheral Requirements

- Peripheral = other equipment in the PLC system that is not directly connected to field I/O devices
  - Compact portable programming device from PLC manufacturer
  - Portable PC with PLC software
  - Magnetic tape storage unit to store control program
  - PROM Programmer
  - Process I/O simulators
  - Communications modules
    - Depends on plant network design
    - Extra modules is required for integration within different brands
  - Operator interfaces
    - Hard-wired local and main control panels
    - GUI software runs on a personal computer
    - Intelligent peripheral devices such as touch screen
    - Industrial PC with function keys and GUI software

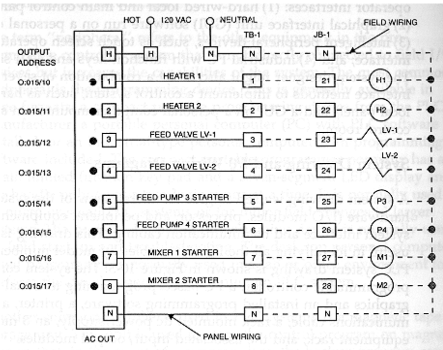
System Drawing

- Gives an overall view of the system hardware
  - I/O modules
  - processor
  - peripheral equipment
  - system interface
  - communication cabling
- Useful in identifying all the interface cables by model number



Example of I/O Wiring Diagrams

- Field wiring is normally indicated by a dashed line
- PLC output addresses are given on the left-hand side
- TB = Terminal Box
- JB = Junction Box

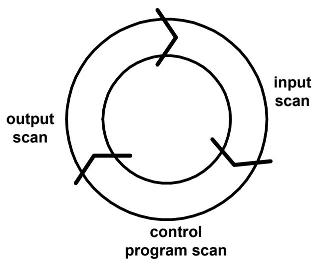


System Programming

- Person involved in system programming :
  - System design engineer
  - Plant operations personnel
  - Maintenance personnel
  - Control system integrator
- Programming by system design engineer takes less time and require less documentation (flowcharting, process description, etc.)
- Selection of programming language type should usually be left to plant operations personnel for easier maintenance and troubleshooting

Real Time Consideration

- The period required to examine inputs, perform the control logic, and execute the outputs is called "*scan time*"
- Scan time must be determined correctly to achieve sufficient real-time performance
- The numbers of I/O points and complex control algorithms affects scan time



Installation

- Control Panel Design
  - Layout
  - Heating Consideration
  - Enclosure Standards (NEMA)
- Maintenance Features
- Panel Duct and Wiring Design
- Power Distribution Design
- Grounding Considerations
- Electrical Interference Considerations
- I/O Module Installation and Wiring
- Equipment Layout Design