8

Programmable Logic Controller

8.1 INTRODUCTION

A programmable logic controller (PLC) is a special-purpose computer dedicated to controlling the operation of a machine and processes. The digital electronic device provides the control logic for industrial and manufacturing machinery, equipment or for complete processes. An entire manufacturing plant may also be controlled through a single master PLC. Specialized programming skills are not essential for writing the programs that provide detailed instructions for every aspect during the entire sequence of operations. Shop personnel can write program instructions into the computer system.

The turbo-machinery industry has played a leading role in originating and refining the PLC. Design criteria for the first logic controllers were initially developed in the 1950's and 1960's. The objective was to eliminate replacing the inflexible and hard-wired relay panels with a computer that can survive in the industrial environment. Initially the units were little more than elaborate switches, but their capabilities and sophistication grew rapidly.

From its modest beginnings, the modern PLC has developed into far greater than a substitute for relay logic panels. Individualized modules that execute a host of functions such as logic, timing, counting, sequencing and proportional-integrating-derivative (PID) control are included in the PLC's. The computers have the added advantage of carrying out arithmetical calculations, analyse data, and exchange data with other programmable controllers and host computers.

PLC's have expanded to include hierarchical systems for integrated manufacturing. A supervisory computer communicates with individual PLC's to coordinate their activities. The intent is to reduce the complexity in the computer program that is required to control a centralized system with one large PLC. Splitting the process into smaller and more manageable portions simplifies the software, and also permits use of smaller PLCs.

8.2 PLC ARCHITECTURE

PLCs range in size from small units performing the job of eight relays to large units for accommodating thousands of inputs and outputs. The controllers are assembled from a number of essential and some special-purpose components. The necessary components are the power supply, processor module, input and output modules and programming unit. The PID controller and data communicator unit are found only in certain PLC designs. Figure 8.1 illustrates the basic components.

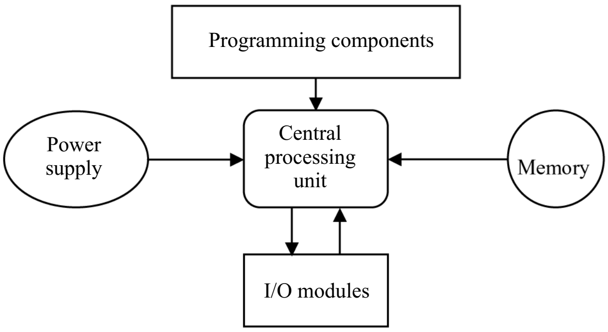


Fig. 8.1 Programmable Logic Controller Architecture

A PLC needs two power supplies. An external power supply is connected to a 120 V or a 240 V AC for the input devices and output loads. The other power source is an internal one for the central processor unit.

Most input devices are of the discrete type for bringing in data from pushbutton contacts and limit and pressure switches. The input and output modules may have up to 32 input terminals, plus a common and a ground terminal. Each input data is converted into a logic voltage. Input of analog signals is also used for control and data acquisition. The analog input module converts the data to a digital form with an analog-to-digital converter. Discrete output devices provide on or off signals for lamps, relays, solenoids and motor starters, among others. Analog outputs require 8 to 16 bits from the output section of the memory. A digital-to-analog converter converts the binary number into an analog voltage.

Programming of instructions may be done with a hand-held programmer, CRT terminal and personal computer. The programming unit allows the operator to examine, enter and edit programs residing in the memory. It also lets monitoring the status of input and output commands, display data and counter values.

8.3 CENTRAL PROCESSING UNIT

The central processing unit, or the CPU, forms the core portion of the PLC. The microprocessor-based unit contains the logic and control algorithm for the controller.

The CPU scans the total information package stored in the memory and the input and output devices continuously. During the scan the CPU also executes instructions based on input data, sends appropriate output responses to the output devices, updates data acquisition systems, and indicates condition changes.

Scan time for larger units depends on the size of the memory and on the configuration of the system. Smaller PLC's have longer scan times than larger units, and have less memory and reduced capacity for special instructions and peripheral equipment.

8.4 MEMORY

The memory in a PLC stores the digital control logic, the process program and the necessary instructions to operate the system. Stored in units called 'bytes', the memory may retain the information (called 'non-volatile memory') or may lose it when electrical power is turned off (called 'volatile memory'). User interface with the PLC mostly works through the volatile memory.

The digital memory of modern microprocessors is classified into four categories: random access memory (or RAM), read only memory (or ROM), programmable read only memory (or PROM), and erasable programmable read only memory (or EPROM). Only the RAM has a volatile memory, which is preferred since it is easy to program, provides large memory to store the programs and data, and is easy to access. ROM memory cells store information permanently, and is used to store the operating program of the computer to permit easy startup. They may also be used for permanent and unalterable process control. PROM's are manufactured with a full array of memory cells, each of which is electronically tied to a signal grid by a fusible link. Programming requires physically melting the fusible link to disable specific memory cells. EPROM's are more convenient to use than ROM or PROM because they can be erased and reprogrammed.

8.5 INPUT AND OUTPUT MODULES

Input and output modules provide interfacing between the CPU and the devices that provide and receive electrical signals. Digital input devices include toggle switches, pushbuttons, limit switches and pressure switches to provide discrete on-off signals. Touch screens that allow the operator to control functions are widely used for process control. Analog devices such as pressure sensors, thermocouples and accelerometers provide continuously changing signals that represent the state of the system.

Some level of signal conditioning may be assigned to the input module to ensure a usable signal is available for the CPU. Circuit protection capabilities will also be necessary in the module to protect the CPU circuitry from faults, surges or spikes in the incoming signal.

The output module reads digital signals transmitted from the CPU, and operates motor starters, solenoids and status monitors. The module essentially functions as a switching device, and provides on-off signals to the associated devices. If the output module is incapable of handling the power needed by the device, it sends a signal to an external relay, which connects with the actual power circuit.

Input and output devices are placed outside the PLC, with connecting wires attached to the terminal strips on the PLC. There are no connections between the input and the output. Each input and output terminal has a unique address. Where solid-state devices are employed as input or output signals for DC circuits, it is imperative that the correct device type be selected.

8.6 PLC PROGRAMMING AND OPERATION

The LD programming language used for a PLC has many similarities with the ladder diagram for relay logic. Contact and coil symbols are used to construct the diagrams. Contact and output symbols have a number written above or below it to identify the location in the RAM process memory where its image is stored.

A number of restrictions are placed on ladder diagram programs. Typical limitations include: (*i*) output must be at the right of the rung, (*ii*) power must flow from left to right, up or down, (*iii*) power must never flow from right to left, (*iv*) number of series contacts is limited to 11, (*v*) number of parallel contacts is limited to 11, (*vi*) a rung can have only one output, (*vii*) a single output must be on the top line in a rung. Sneak paths are not permitted in programming a PLC, which simplifies the programming. But adjustments are needed when converting a relay panel to a PLC program.

Figure 8.2 shows an example of the schematic and ladder diagrams of a relay logic circuit for controlling the action of a pneumatic cylinder. When the PLC is in the operating mode, the processor repeats the four-step cycle of: (*i*) scanning and storing the new input conditions, (*ii*) scan the program and derive new output conditions, (*iii*) transfer output information to output device, and (*iv*) communicate and carry out related routine tasks.

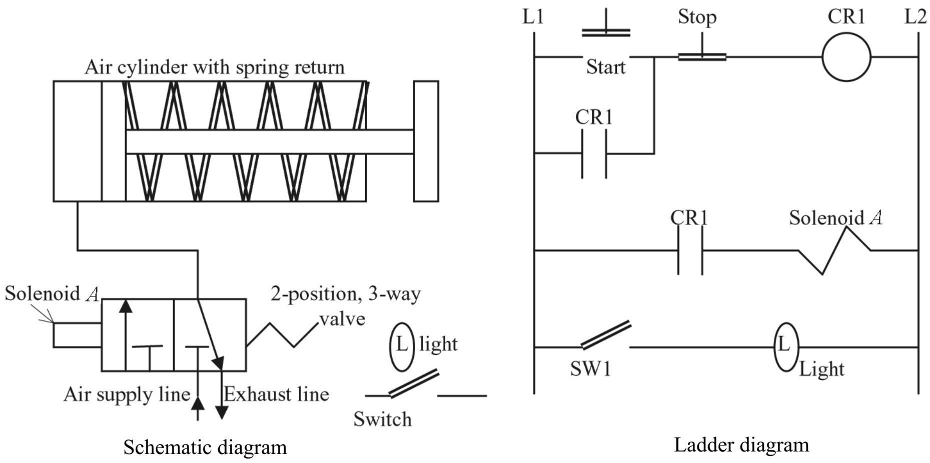


Fig. 8.2 Ladder Diagram Control Circuit of Pneumatic Cylinder for PLC Implementation

It may be observed that the order of operations is of significance, since the PLC evaluates the program one rung at a time. Also, a time delay occurs between the input action and the resulting change in output conditions, because the PLC takes time to execute one cycle.

8.7 PROGRAMMING FUNCTIONS

The open and closed contact symbols are instructions in the LD programming language. The instructions for event-driven processes with on and off inputs and outputs are the NO contact, the NC contact and the output instructions. Some control applications require the counting function, and others need the ability to compare and manipulate numbers with arithmetic functions. Control of a process variable requires the PID control function. All advanced functions are available in the PLC.

The 'timer' is frequently used in the PLC, and the 'time delay on' and 'time delay off' are the most common timing functions. Two types of timers are used, the one-input non-retentive timer and the two-input retentive timer.

The non-retentive timer has one output and a preset delay time. Thus, if the preset delay time is 12 s, the timer runs for 12 s following the turning on of the input before the output turns on. The output stays on until the input is turned off. The timer resets every time the input is turned on.

The retentive timer has two inputs (Enable/Reset and Run lines), one output (Done line), and two status bits (Enable and Run lines) that can be used as internal outputs. The timer runs only after turning on both the Enable/Reset and the Run lines, but stops when the delay time is satisfied. The timer is reset only when the Enable/Reset line is turned off.

The 'counter' of a PLC keeps track of the number of times a contact opens. The objective is to count the number of occurrences of an event in a control system. The counting sequence may run from zero to a preset value or may be from a preset value to zero. A memory register can store both the preset and the counted values. Multiple counters may be employed in a PLC if the incidence of more than one event is to be counted. For instance, in the filling of a storage tank with a fluid, one counter counts the number of gallons entering the tank at the inlet and another counter counts the number of gallons leaving the tank at the discharge. The two counters can use the same memory register for recording. For each gallon that enters or leaves the tank, the counter adds 1 to the count register for entering and subtracts 1 for leaving. Thus, at any given time the count register provides the number of gallons held in the tank.

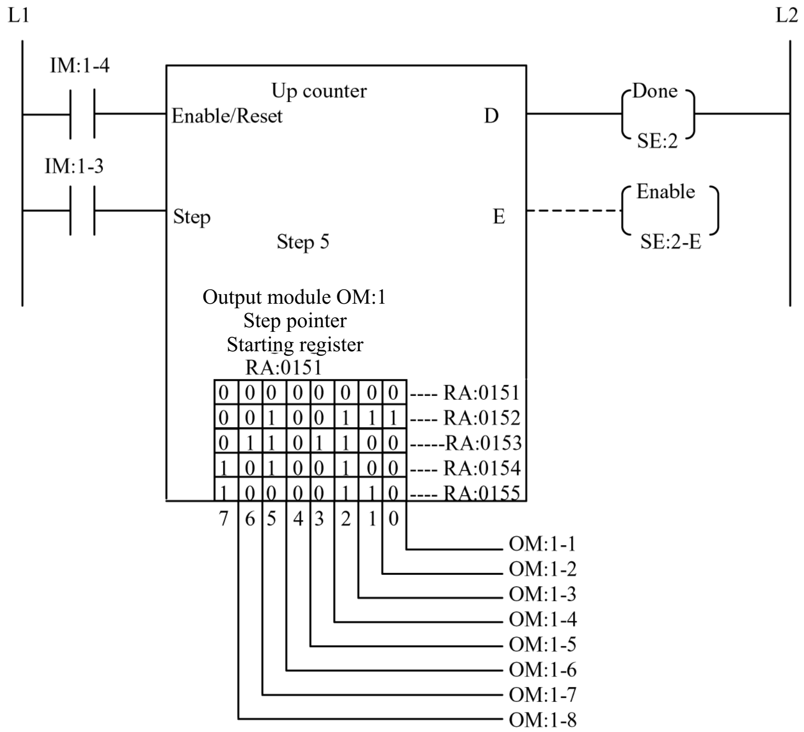
The 'sequencer' function executes a series of steps to produce specific conditions in a number of on or off outputs at each step. The specified output pattern for each step is stored in the memory. The number of outputs that a sequencer can control depends on the word size of the PLC.

The generic sequencer is an event-driven function, where the event is the occurrence of an 'off' to an 'on' change on the step input line. Combined with a timer, it becomes a time-driven function. The generic sequencer shown in figure 8.3 controls eight outputs in module OM:1 in a five-step sequence. The desired output conditions are stored in memory cells RA:0151 through RA:0155, the contents of which are shown in the figure. Also shown are the lines connecting memory bits 0 through 7 to the eight terminals in the output module OM:1. When the step pointer selects terminal OM:1, the actual selection is memory cell RA:0150. Combined with a timer, as shown in figure 8.4, the sequencer becomes a time-driven function. When OM:1-1 is turned on, the timer output T-8 produces an off-on-off pulse every 30 seconds.

Most PLCs have the ability to compare two operands. The operands may be numbers stored in the memory, or one may be a constant and the other a memory register. A 'compare' function is programmed as a rung in the ladder diagram program with a contact that specifies the operands.

'Arithmetic' functions perform arithmetic calculations of one or two operands and store the result in a destination memory register. Addition, subtraction, multiplication, division and square root may be determined with the aid of the arithmetic function. This function is programmed as a rung in the ladder diagram program with a contact that specifies the operands.

A 'control' function changes the method of scanning the program. The function may be split into four separate operations. The 'skip' function permits the PLC to pass over a portion of the program as it is being scanned. The 'master control relay' (MCR) function is similar to the skip function in that a specified number of rungs immediately following the MCR rung are affected. The 'jump' function causes the PLC scan to go immediately to the next rung in the program. The 'jump to subroutine' (JSR) function causes the scan to go to another section of the program that ends with a rung that has a return (RET) label.



0	0	0	0	0	0	0	0
0	0	1	0	0	1	1	1
0	1	1	0	1	1	0	0
1	0	1	0	0	1	0	0
1	0	0	0	0	1	1	0

Fig. 8.3 Generic Sequencer

Medium and large PLCs have the ability to copy data from one place and place it in another area. A 'move' label copies data from the source register to the destination register. A 'block transfer' (BT) function copies data from a block of source registers into a block of destination registers.

'Bit manipulation' functions change the bits in a memory register in a specified manner. The functions permit changing the status of individual bits, moving and rotating the bits in a register to the left or to the right, and to perform logical operations on corresponding bits in two operands stored in the memory registers.

Many PLCs have the PID control function built into the input/output module. Other PLCs have the PID function programmed as a rung in the ladder diagram program. A generic PID controller function is shown in figure 8.5.

8.8 ANALOG CONTROL

Electronic 'analog control' calls for an error detector and a control mode unit. An analog controller requires a single operational amplifier as a function generator and some resistors and capacitors to implement the transfer function of the desired control mode or combination of modes.

The analog proportional controller uses three resistors to form an inverting amplifier. The circuit has two inputs, an error and an output offset. The proportional gain is obtained by dividing the feedback resistor by the error input resistor. The offset resistor must equal the feedback resistor

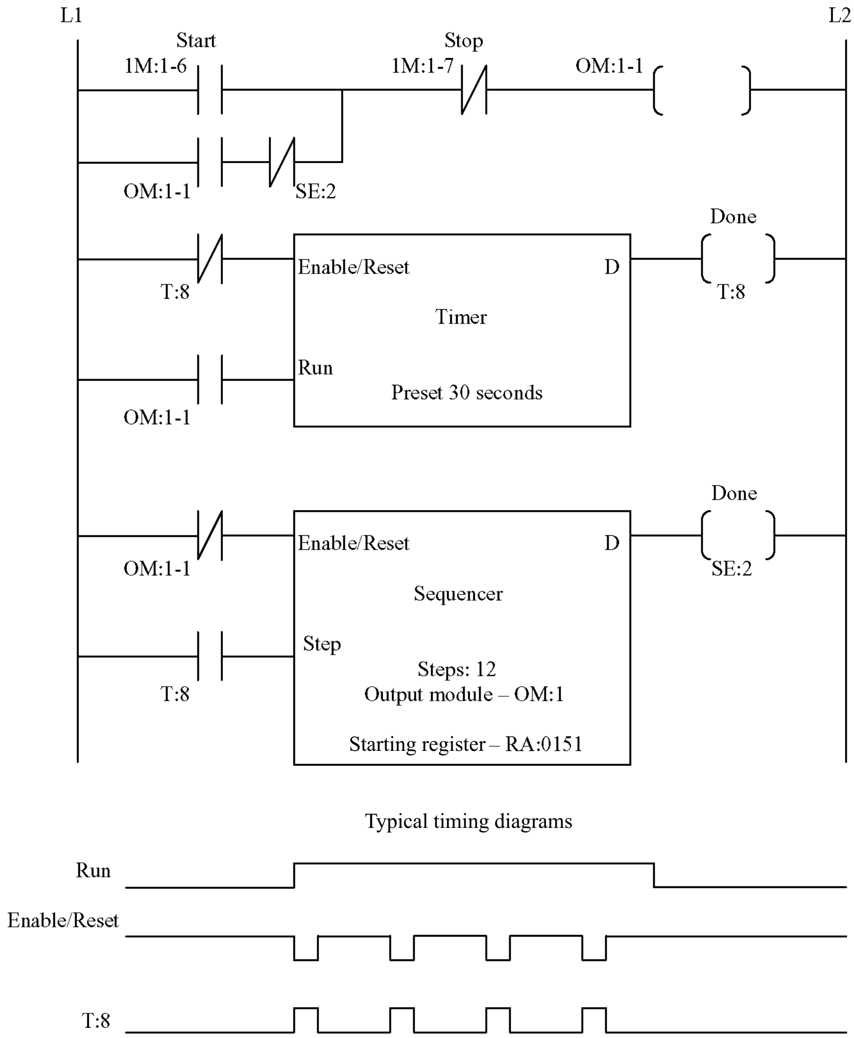


Fig. 8.4 Generic Sequencer with Timed Steps

to satisfy the time domain equation. The output lines may be reversed to make the output either positive or negative with respect to the error.

The proportional plus integral controller needs two resistors and a capacitor to implement the PI transfer function. The capacitor is connected in series with the feedback resistor. The gain is given by dividing the feedback resistor by the input resistor. The integral action rate is equal to the reciprocal of the product of the input resistor and the capacitor.

The proportional plus derivative controller uses four resistors and a capacitor to implement the PD transfer function. The circuit is a proportional controller with a parallel combination of a resistor and a capacitor placed in series with the input resistor.

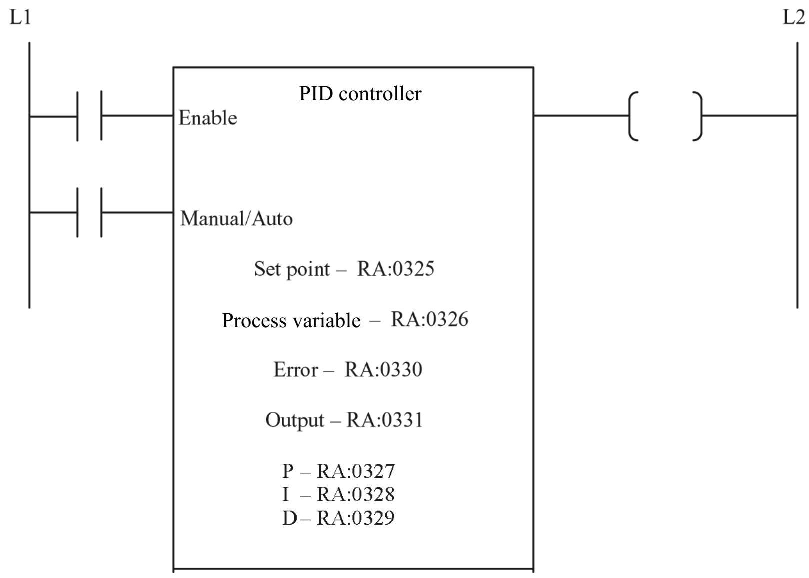


Fig. 8.5 Generic PID Controller Function

8.9 DIGITAL CONTROL

Microprocessor based digital controllers are widely used in industrial control systems since advanced tuning, multi-variable control and expert systems are available. Digital controllers in the closed-loop usually implement the PI, PD or PID modes.

A digital controller measures the controlled variable after specific intervals of time, called the sampling time, Δt. The measurement is converted to a binary number for input to a digital computer. The computer subtracts each reading of the sample from the set point value to obtain a set of error samples.

At the end of each error sample calculation, the digital PID controller follows a procedure called the PID algorithm to calculate the controller output based on the error samples. The positional version of the algorithm determines the actuator position, while the incremental version of the algorithm determines changes in the position of the actuator. The positional version of the algorithm has the advantage that the controller has the actuator position retained in its memory. The incremental version is suitable for output devices such as stepper motors.

The integral mode sometimes presents computational difficulties that can lead to unsatisfactory results. When the value of $PI\Delta t$ is less than 1, it is convenient to work with the reciprocal of $PI\Delta t$, which could be stored in the computer as an integer. If the value of $PI\Delta t$ is very small, the computer may ignore relatively large errors because of insufficient resolution. The change would then be lost unless a special provision is made to include the change in the calculations for the next sample. The end result is an error offset that the integral mode is unable to eliminate. One solution is to increase the precision by increasing the word length in the computer. A 16-bit word length, for example, has a precision of 1 part in 65,536. Another solution is to add the unused portion of the sum of the error samples to the current error sample before computing the integral mode change.

The derivative mode can also present computational difficulties that can lead to unsatisfactory results. A slowly changing signal, for instance, results in a fluctuating derivative action mode. The derivative mode estimates the rate of change of the error, de/dt. Since Δt is fixed by the sampling rate, attention is focused on the change in the error. To obtain a better estimate of Δe, estimators of previous samples may be used to smoothen the derivative term and thus improve the estimate.

8.10 REVIEW PROBLEMS

Example problem # 8.1: A batch blending of water and acid is to be controlled by a timer- and event-driven sequential control system, as shown in figure 8.6. Three levels are specified in the mixing tank. The process initiates with the drain valve open, water and acid valves closed, mixer motor is off, and tank is empty. The design criteria specifies the following steps:

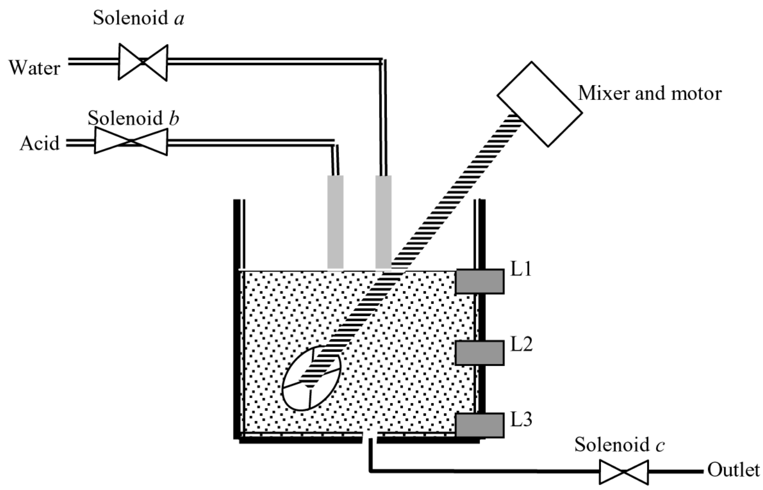


Fig. 8.6 Batch Mixer

Step 1 Begin when Start button is pressed, fill tank with water up to level 2, and ending with level switch L2 closed.

Step 2 Following step 1, fill tank with acid up to level 3, tank then fill with 60 per cent water and 40 per cent acid, close level switch 3. Turn on mixer motor.

Step 3 Mixer runs for 5 minutes, then stop mixer motor.

Step 4 Open switch L1 and drain tank.

Step 5 When tank is drained, close switch L1 and restart process for another batch.

Draw a PLC ladder diagram program for the process.

Solution: The PLC program has six rungs (figure 8.7). Rung 1 turns on output OM:1-4 when Start is pressed, until tank is empty at end of step 5. Rung 2 closes normally open drain valve, before timer T:4 activates. Rung 3 energizes solenoid *a*, L2 is closed and de-energizes it. Rung 4 energizes solenoid *b* when L2 closes, de-energizes it and closes L3. Rung 5 enables timer when cycle light is on, starts timer when L3 closes, turns on output T:4 when timer goes off. Rung 6 turns mixer on during steps 2 and 3.

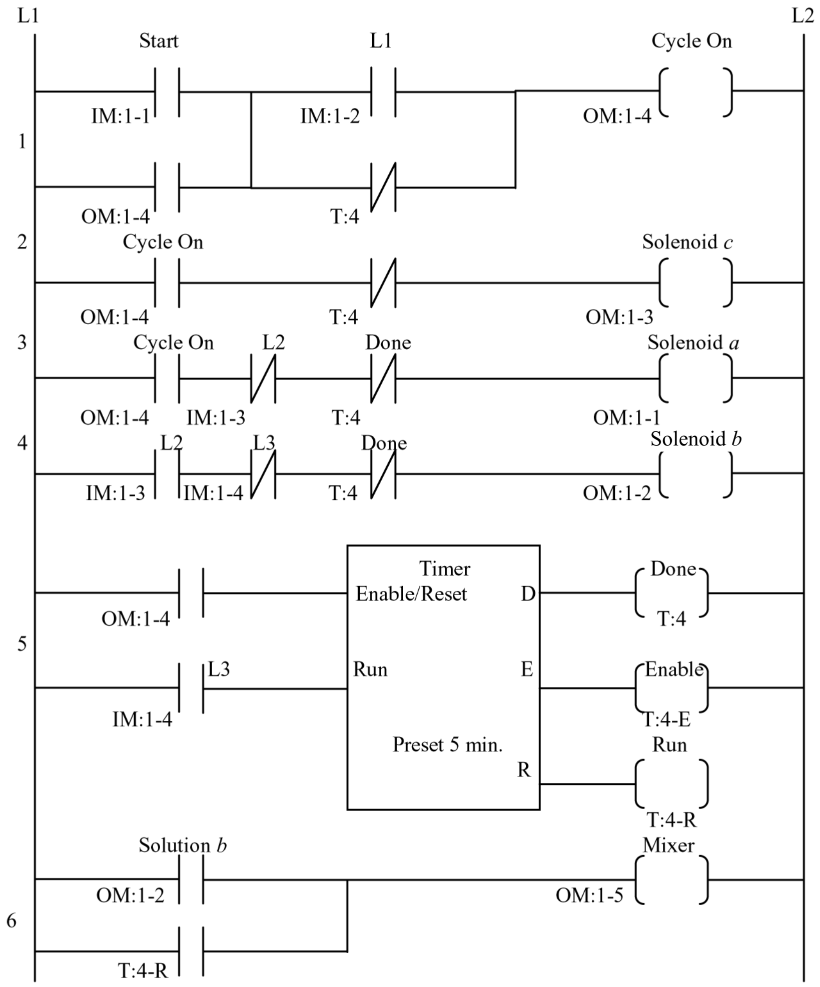


Fig. 8.7 PLC Ladder Diagram

Example problem # 8.2: Determine the values of R_1 and R_i for the electronic proportional plus integral controller shown in figure 8.8. Gain $P = 1.8$, and integral action rate $I = 0.03\ s^{-1}$. Use a 10 μF capacitor for C_i, and determine the transfer function.

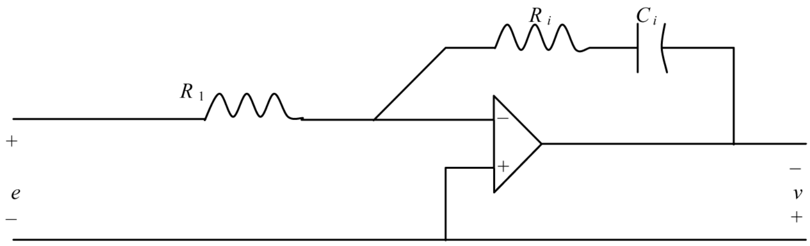
Solution: Integral $I = 1/(R_i\ C_i)$

Hence $R_i = 1/(IC_i) = (1/0.03) \times (1/10^{-5}) = 3.33$ MΩ

Gain $P = R_i/R_1$.

Or $R_1 = R_i/P = (3.33 \times 10^6)/1.8 = 1.85$ MΩ

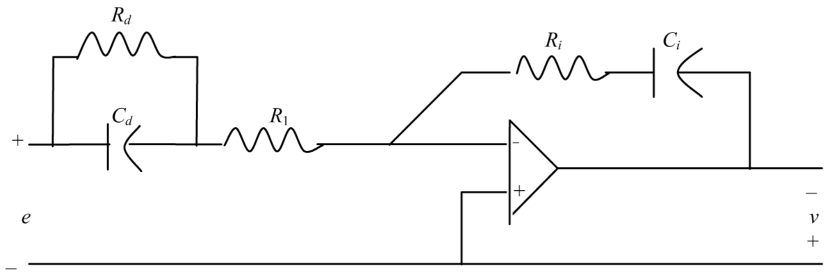
Transfer function = $V/E = 1.8 \times (1 + 33.33s)/(33.33s)$



Transfer function = $V/E = P[I + S]/S$, $I = 1/R_iC_i$, gain $P = R_i/R_1$

Fig. 8.8 Analog Proportional Plus Integral Controller

Example problem # 8.3: Determine the value of R_1, R_i, R_d and C_d for the PID analog controller of figure 8.9. Gain $P = 3.5$, derivative action time constant $D = 0.4$ s, derivative limiter coefficient $\alpha = 0.1$, and integral action rate $I = 1/6\ s^{-1}$. Use a 10 μF capacitor for C_i, and determine the transfer function.



Transfer function = $V/E = P[(I + S)/S] \times [(1 + DS)/(1 + \alpha DS]$
$I = 1/ R_iC_i$, gain $P = R_i/(R_1 + R_d)$, $D = R_dC_d$, $\alpha = R_1/(R_1 + R_d)$

Fig. 8.9 Analog Proportional Plus Integral Plus Derivative (PID) Controller

Solution: Integral $I = 1/(R_iC_i)$

Hence $R_i = 1/(IC_i) = 6 \times (1/10^{-5}) = 0.6\ \text{M}\Omega$

$\alpha = R_1/(R_1 + R_d)$

Hence $R_d = R_1(1 - \alpha)/\alpha = R_1(1 - 0.1)/0.1 = 9R_1$

$P = R_i/(R_1 + R_d) = R_i/(10R_1)$

Hence $R_1 = R_i/(10P) = 0.6\ \text{M}\Omega/(10 \times 3.5) = 17.1\ \text{k}\Omega$

And $R_d = 9R_1 = 9 \times 17.1 = 154.26\ \text{k}\Omega$

$D = R_dC_d$

Hence $C_d = D/R_d = 0.4/154.26 = 2593\ \mu\text{F}$

Transfer function $= V/E = 3.5 \times [1/6 + (1 + 0.4/6)s + 0.4s^2]/(s + 0.04s^2)$

$= (1 + 6.4s + 2.4s^2)/(1.714s + 0.0685s^2)$

8.11 EXERCISES

8.1 Draw a PLC ladder diagram for a system to turn on a light when the count reaches 25 and then switches off the light when the count reaches 45.

8.2 Draw a PLC ladder diagram for a system to turn on a solenoid actuated valve when count X reaches 20 or when count B reaches 30.

8.3 Draw a ladder diagram program for two comparators AB and BC to produce 5 comparisons as follows:

Rung 1 is TRUE when op1 equals op2 (AB).

Rung 2 is TRUE when op1 is not equal to op2 (NB).

Rung 3 is TRUE when op1 is greater than or equal to op2 (BC).

Rung 4 is TRUE when op1 is less than op2 (LT).

Rung 5 is TRUE when op1 is greater than op2 (CT).

8.4 An analog proportional plus integral mode controller of the type shown in figure 8.8 uses 50 kΩ resistor for R_1. Determine the value of R_i if the gain (*i*) $P = 0.35$, (*ii*) $P = 0.85$, (*iii*) $P = 2.5$, (*iv*) $P = 7.5$.

8.5 The analog PID controller of the type shown in figure 8.9 uses 80 kΩ resistor for R_i and $R_1 = 25$ kΩ. Determine the value of R_d if the gain (*i*) $P = 0.15$, (*ii*) $P = 0.75$, (*iii*) $P = 3.5$, (*iv*) $P = 8.5$.

References & Bibliography

Eckman, D.P., “Industrial Instrumentation”, John Wiley & Sons, New York, 1980.

Floyd, T.L., “Digital Fundamentals”, Macmillan Publishing Co., New York, 1990.

Herman, S.L. and Alerich, W.N., “Industrial Motor Control”, Delmar Publishers, Albany, NY, 1985.

Hordeski, M.F., “Design of Microprocessor, Sensor and Control Systems”, Reston Publishing Co., Englewood Cliffs, NJ, 1985.

Johnson, C.D., “Process Control Instrumentation Technology”, Prentice Hall, Upper Saddle River, NJ, 1997.

Jones, C.T. and Bryan, L.A., “Programmable Controllers: Concept and Applications”, International Programmable Controls, Inc., Atlanta, 1983.

Kilian, C.T., “Modern Control Technology Components and System”, West Publishing Co., St. Paul, MN, 1996.

Stein, D.H., “Introduction to Digital Data Communications”, Delmar Publishers, Albany, NY, 1985.

Webb, J.W. and Reis, R.A., “Programmable Logic Controllers”, Prentice Hall, Upper Saddle River, NJ, 1995.