CHAPTER 5 SOLVED PROBLEMS

General instructions for the problems:

Write a ladder logic program for the application and implement it for one or more of the following PLC ladder logic languages:

ControlLogix/CompactLogix, **or**

Micro800 **or**,

MicroLogix, **or**

SLC-500

If any part of the operation is ambiguous, write down your additional assumptions.

The physical inputs, physical outputs, and internal tags/symbols for each problem are given in the problem. **DO NOT** assign any more physical inputs!

Your solution should include the following:

1. Specify the PLC processor used.
2. Ladder logic diagram (with comments). For consistency among the different PLCs, use only tags/symbols in the ladder logic. Use instructions consistent with the PLC processor.
3. Table listing additional internal memory (variables/symbols/tags) used and a brief description of their use. For the ControlLogix, CompactLogix, and Micro800 processors, list the internal tags and the data type. For the MicroLogix/SLC processors, list the internal symbols and the associated memory address.

SP5-1. Develop a ladder logic program that will turn **on** air cylinder CYL101 0.7 seconds after proximity switch PROX101 is turned **on**. Air cylinder CYL102 should turn **on** 2.5 seconds after CYL101 is turned **on**. Pilot light PL105 should turn **on** 3.0 seconds after CYL102 is turned **on**. Activating (turning **on**) limit switch LS110 will reset all timers and turn off all cylinders and the pilot light. The program must ensure that turning **on** LS110 is ignored if PL105 is **off**. Assume that PROX101 is **on** for at least one scan, but no longer than 1 second. Use no more than two internal coils.

Assume the following input and output assignments:

Variable	Description
PROX101	Proximity switch, **on** when closed.
LS110	Limit switch, **on** when closed.
CYL101	Air cylinder control, **on** to extend cylinder
CYL102	Air cylinder control, **on** to extend cylinder
PL105	Pilot light

The addresses associated with the physical I/O are:

Tag/Symbol	ControlLogix	Micro800	MLogix	SLC-500
PROX101	Local:2:I.Data.0	_IO_EM_DI_00	I:0/0	I:0/0
LS110	Local:2:I.Data.1	_IO_EM_DI_01	I:0/1	I:0/1

CYL101	Local:4:O.Data.0	_IO_EM_DO_00	O:0/0	O:4/0
CYL102	Local:4:O.Data.1	_IO_EM_DO_01	O:0/2	O:4/1
PL105	Local:4:O.Data.2	_IO_EM_DO_02	O:0/3	O:4/2

SP5-2. Develop a ladder logic program to produce a 20-second one-shot pulse. When the START_PB is pressed, the LA1 lamp should light for 20±0.1 seconds and then turn **off**. Pressing START_PB should have no effect when the lamp is **on** (the light should not go out, nor should the light remain on any longer). When the lamp turns **off**, another press of START_PB should start the pulse over. If START_PB is held down, the lamp should light for twenty seconds and turn **off**. The lamp should not turn **on** until the start switch is released and pressed again. Use only one timer.

Assume the following input and output assignments:

Variable	Description
START_PB	Start push button switch, **on** when pressed.
LA1	Pilot light

The addresses associated with the physical I/O are:

Tag/Symbol	ControlLogix	Micro800	MicroLogix	SLC-500
START_PB	Local:1:I.Data.7	_IO_EM_DI_07	I:0/7	I:1/7
LA1	Local:3:O.Data.9	_IO_EM_DO_03	O:0/5	O:3/11

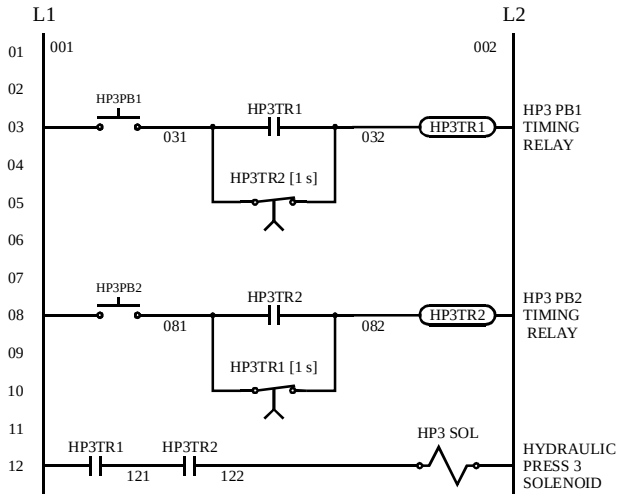
SP5-3. Implement the relay logic of Figure SP5.3*a* in ladder logic. The description of the operation is below. The connections to the PLC are shown in Figure SP5.3*b*.

For safety reasons, an operator is required to actuate two pushbuttons in order to close a stamping press. The intention is to ensure that both of the operator's hands are out of the press when it closes. The press is equipped with the relay logic circuit shown in Figure SP5.3*a*, which incorporates timing relays that prevent the operator from taping one pushbutton down and making the press faster but unsafe to operate.

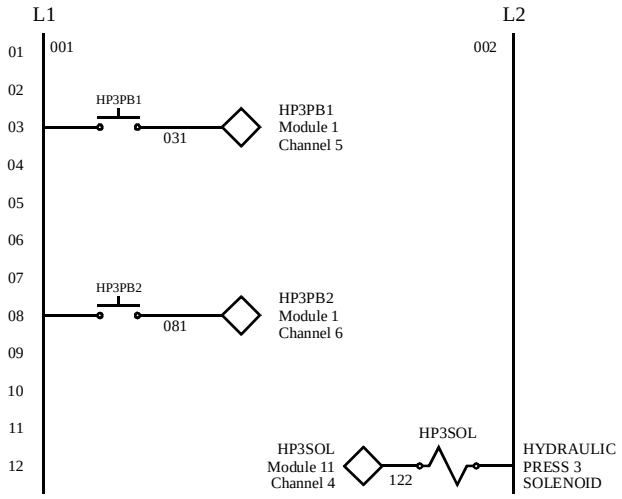
The normally-closed timing relay contacts labeled HP3TR1[1 s] and HP3TR2[1 s] in Figure SP5.3*a* open 1 second after the relay coil is energized. These contacts close immediately when the coil is de-energized. The normally-open contacts labeled HP3TR1 and HP3TR2 close and open immediately. A solenoid labeled HP3SOL is used to actuate the hydraulic press. The pushbuttons are HP3PB1 and HP3PB2. The numbers beneath the lines are the wire numbers used in the installation.

The relay logic operates as follows. Assume both pushbuttons are open and HP3PB1 is pressed first.

1. An operator presses (closes) HP3PB1, power flows through the HP3TR2[1 s] contact and energizes the HP3TR1 coil.
2. The HP3TR1 contact on the first rung closes, sealing the HP3TR1 coil **on** as long as HP3PB1 is held down.
3. One second after HP3TR1 is energized, the HP3TR1[1 s] contact on the second rung opens.



(a)



(b)

Figure SP5.3. Press operator safety push buttons: *(a)* relay ladder logic; *(b)* PLC I/O connections.

4. If HP3PB2 is pressed within one second after HP3PB1 is pressed, power flows through the HP3TR1[1 s] contact, energizing the HP3TR2 coil.
5. When both HP3TR1 and HP3TR2 relay coils are energized, both contacts on the third rung close, energizing the HP3SOL solenoid.
6. If HP3PB2 is pressed more than one second after HP3PB1 is pressed, then the HP3TR2 coil is **not** energized because the HP3TR1[1 s] contact has opened,

blocking power flow to the HP3TR2 coil. In this case, the second contact on the third rung remains open and the solenoid does not turn **on**.

Assume the following physical input and output assignments:

Variable	Description
HP3PB1	N.O. push button
HP3PB2	N.O. push button
HP3SOL	Press solenoid; **on** to run press

The addresses associated with the physical I/O are:

Tag/Symbol	ControlLogix	Micro800	MLogix	SLC-500
HP3PB1	Local:1:I.Data.4	_IO_EM_DI_04	I:0/4	I:1/4
HP3PB2	Local:1:I.Data.5	_IO_EM_DI_05	I:0/5	I:1/5
HP3SOL	Local:6:O.Data.3	_IO_EM_DO_03	O:0/3	O:6/3

SP5-4. Develop a ladder logic program that will latch **on** an output, CYL410, after a proximity switch, PROX412, indicates that 100 parts have passed. When the count of 100 is reached, the logic should reset the counter. Push button PB421 is pressed to unlatch CYL410. The counter should be allowed to count while CYL410 is **on** (but not yet unlatched). Assume the following input and output assignments:

Variable	Description
PROX412	Proximity switch, **on** when part passes
PB421	Push button switch that is **on** when pressed
CYL410	Pneumatic cylinder control

The addresses associated with the physical I/O are:

Tag/Symbol	ControlLogix	Micro800	MLogix	SLC-500
PROX412	Local:1:I.Data.3	_IO_EM_DI_03	I:0/3	I:1/3
PB421	Local:1:I.Data.5	_IO_EM_DI_05	I:0/5	I:1/5
CYL410	Local:2:O.Data.7	_IO_EM_DO_03	O:0/7	O:2/7

SP5-5. Develop a ladder logic program that provides an indication when a certain push button has been pressed 20 times. Latch **on** an output, PL612, after an input, PB151, has been pressed 20 times. When the count of 20 is reached, the logic should reset the counter. Pushbutton PB613 is pressed to unlatch PL612. The counter should be allowed to count while PL612 is **on** (but not yet unlatched). Assume the following input and output assignments:

Variable	Description
PB151	Push button switch that is **on** when pressed.
PB613	Push button switch that is **on** when pressed.
PL612	Pilot light

The addresses associated with the physical I/O are:

Tag/Symbol	ControlLogix	Micro800	MLogix	SLC-500

PB151	Local:1:I.Data.4	_IO_EM_DI_04	I:0/4	I:1/4
PB613	Local:6:I.Data.2	_IO_EM_DI_02	I:0/2	I:6/2
PL612	Local:7:O.Data.2	_IO_EM_DO_02	O:0/2	O:7/2

SP5-6. Develop a ladder logic program for an automatic plant waterer (Figure SP5.6) with the following specifications:

1. Every 8 hours, water is added to the container, by turning on P103, up to the desired level (LS102 closes). When the desired level is reached, P103 should be shut off. At the end of the 8-hour interval, the 8-hour clock should be reset.
2. If LS102 is still closed at the end of the 8-hour interval, assume the level switch is stuck on, and turn on the AH104 alarm. Do not add any water, but still reset the 8-hour clock.
3. If P103 remains on for 25 seconds and LS102 has not turned on, then assume either LS102 is stuck off or the reservoir is empty and turn on the AH104 alarm.
4. When the alarm is on, the pump will be shut off and the 8-hour timer is paused.
5. SW101 must be on for the waterer to operate. If SW101 is off, the 8-hour timer must be paused (but not reset).
6. OVERRIDE is an override/reset switch. When this switch is pressed, the action of the program is as if the end of the 8-hour interval is reached (attempt to fill container and reset 8-hour timer). Also, the OVERRIDE switch turns off the

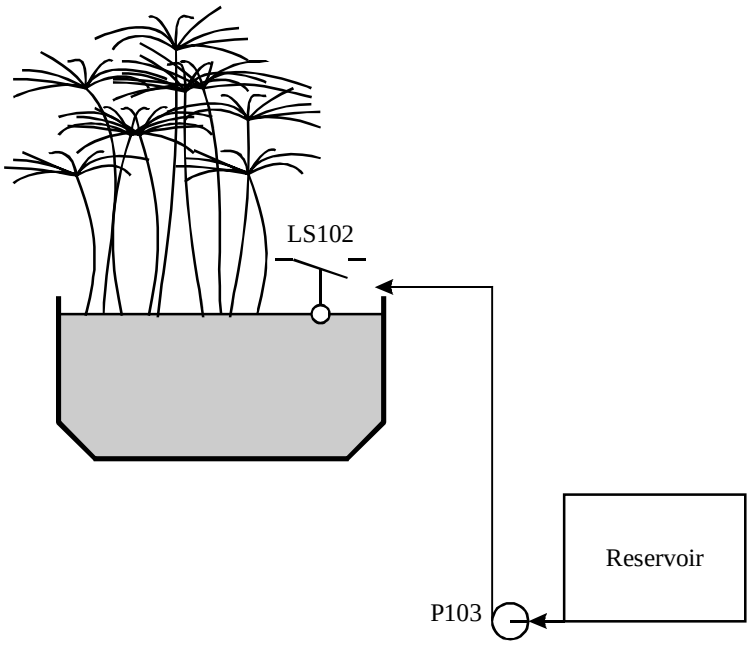


Figure SP5.6. Plant waterer.

alarm. Holding down the override switch should not change its operation, even if held down for more than 8 hours.

7. The OVERRIDE switch is the only item that can turn off the AH104 alarm.
8. As part of your program construct an 8-hour clock that keeps track of seconds, minutes, and hours. Do not use a timer with an 8-hour preset time.

Assume the following physical input and output assignments:

Variable	Description
SW101	Toggle switch, **on** (closed) when enabling plant waterer.
LS102	Level switch, N.O., closed when water at correct level.
OVERRIDE	Override push button, **on** to override the timer operation (described above).
P103	Pump, **on** to pump water into plant container.
AH104	Alarm horn; **on** to sound alarm.

The addresses associated with the physical I/O are:

Tag/Symbol	ControlLogix	Micro800	MLogix	SLC-500
SW101	Local:1:I.Data.0	_IO_EM_DI_00	I:0/0	I:1/0
LS102	Local:1:I.Data.1	_IO_EM_DI_01	I:0/1	I:1/1
OVERRIDE	Local:1:I.Data.2	_IO_EM_DI_02	I:0/2	I:1/2
P103	Local:2:O.Data.0	_IO_EM_DO_00	O:0/0	O:2/0
AH104	Local:2:O.Data.1	_IO_EM_DO_01	O:0/1	O:2/1