Assume the following physical inputs and outputs.

Variable	Description
START_PB	Start push button, N. O., **on** when starting.
STOP_PB	Stop push button, N. C., **off** when stopping.
RESET_PB	Reset push button, N. O., **on** when restoring station to initial state.
BOX_PRESENT	Photoelectric sensor, **on** (closed) when box is present at gate.
LS1	Limit switch that closes (**on**) when ram is fully retracted
LS2	Limit switch that closes (**on**) when ram is fully extended
GATE_SOL	Cylinder control to drop gate, **on** to hold down gate. When **off**, gate is held up by a spring.
EXTEND_SOL	Pneumatic ram extension solenoid, **on** to extend ram, **off** has no effect on ram position
RETRACT_SOL	Pneumatic ram retraction solenoid, **on** to retract ram, **off** has no effect on ram position
MOTOR1	Inbound conveyor motor control, **on** to move inbound conveyor
MOTOR2	Outbound conveyor motor control, **on** to move outbound conveyor

The addresses associated with the physical inputs and outputs are:

Variable	Modicon	PLC-5	ControlLogix	Siemens	GE Fanuc
START_PB	100017	I:02/00	Local:2:I.Data.0	I8.0	%I87
STOP_PB	100018	I:02/01	Local:2:I.Data.1	I8.1	%I88
RESET_PB	100019	I:02/02	Local:2:I.Data.2	I8.2	%I89
BOX_PRESENT	100021	I:02/04	Local:2:I.Data.4	I8.4	%I91
LS1	100022	I:02/05	Local:2:I.Data.5	I8.5	%I92
LS2	100023	I:02/06	Local:2:I.Data.6	I8.6	%I93
GATE_SOL	000033	O:03/00	Local:3:O.Data.0	Q12.0	%Q13
EXTEND_SOL	000034	O:03/01	Local:3:O.Data.1	Q12.1	%Q14
RETRACT_SOL	000035	O:03/02	Local:3:O.Data.2	Q12.2	%Q15
MOTOR1	000036	O:03/03	Local:3:O.Data.3	Q12.3	%Q16
MOTOR2	000037	O:03/04	Local:3:O.Data.4	Q12.4	%Q17

P6-8. Hole Drilling Station 1 Control. Using the function chart approach, implement the program for the following station that drills a hole in each part on a conveyor.

Figure P6.8 shows the layout of a station that drills a hole in each part that comes down the conveyor. The part completely fills the area enclosed in the dashed lines. This station is only one is a series of stations along this conveyor. You are implementing ladder logic for this station only. You have no control over the conveyor, so assume it is always moving. This particular line is asynchronous, that is, each station processes parts at its own speed and does not coordinate its operation with any other station. Because this is an asynchronous line, each station

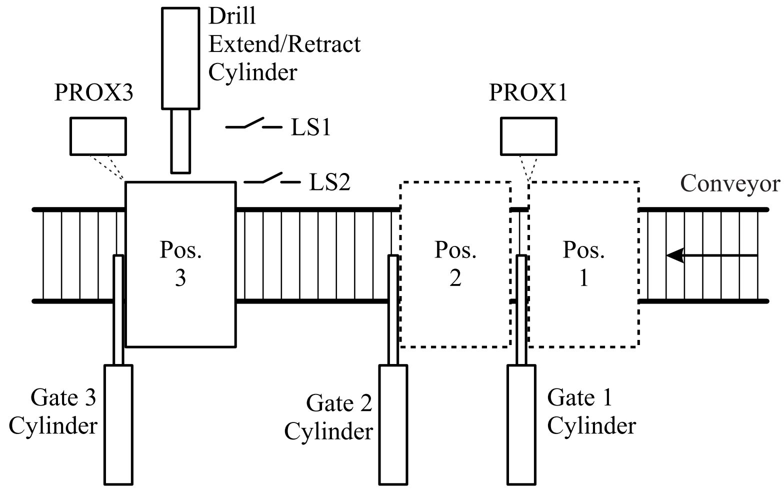

Figure P6.8. Drilling station.

contains a series of two gates that control access to the station and allow parts to queue up before the station.

Upon initial startup, assume that there are no parts waiting at gate 1. When a part is detected at Gate 1 (by PROX1), the following major steps are executed:

Sequence Gate 1 and Gate 2 to allow only one part to move into the drilling position (against Gate 3).
Drill is turned and is extended (moves out) to the correct depth,
Drill is retracted,
Open Gate 3 to allow drilled part to move out.

The operation then repeats. Assume the conveyor is on at all times. The conveyor slides beneath the parts as they are held against a gate or being drilled.

To move only one part into the station, Gates 1 and 2 are sequenced as follows (assume both gates are closed at the start): Gate 1 is opened to allow a part to move into position 2 (sensed by PROX2). Then Gate 1 is closed and Gate 2 is opened. As far as your ladder logic is concerned, assume that Gate 1 is closed and Gate 2 is opened at the same time. Physically, Gate 1 closes much faster than Gate 2 opens, so any part in position 1 is prevented from moving when Gate 2 is open enough to allow the part to move from position 2 to position 3. Gate 2 is closed when the part is in position 3 (sensed by PROX3).

A single-action air cylinder drives each gate. Once a cylinder control is energized, the gate is opened and remains open as long as power is applied (turned **on**). The gate closes when power is removed (turned **off**).

The drill extension/retraction is driven by a double-action air cylinder with two direction controls. Once a direction control is energized, the drill mechanism moves and keeps moving as long as power is applied (turned **on**). The mechanism stops at its current position when power is removed (turned **off**). The mechanism will not move if both opposing directions are energized simultaneously (e.g., extension and retraction). DRILL_MOTOR must be **on** whenever the drill is being extended or retracted.

Proximity sensor, PROX1, is **on** when a part is in position 1, meaning there is a part to be processed. PROX3 is **on** when a part is in position 3, ready to be drilled. When PROX3 is **off**, the part has passed Gate 3 and moved out of the station.

The drill position is indicated by limit switches. LS1 is **on** when the drill is fully retracted. LS2 is **on** when the drill is extended to the proper hole depth.

The start/stop switches are only for the station. They do not control any other stations or the conveyor. Upon initial startup, assume there are no parts present in any of the positions (1-3). If the stop switch is pressed at any time, the station operation should pause, except during drilling. The operation **must not** pause when the drill is being extended or retracted (otherwise it may jam or ruin the hole). When the start switch is pressed while the operation is paused, the station should resume the suspended step. If paused when not drilling, **do not advance** to the next step. If you advance to the next step when paused, you will have problems with the gate operations. When the station is paused, the drill motor and the drill extension/retraction solenoids should not be affected. Any open gates must not be closed when paused (or parts may be knocked off).

A separate reset switch is provided that when pressed, a drill not fully retracted is retracted, and the process step is set as if the process is waiting for the next part. Note that the drill motor must remain **on** while a drill is retracting. When the start switch is pressed, no items are assumed present at position 1. To keep the problem simple, do not implement a function chart for reset; just retract the drill (motor **on**) until finished. The reset switch should have no effect unless the operation is already paused.

Assume the tolerance on all timer values is ±0.1 seconds.

Assume the following physical inputs and outputs.

Variable	Description
START_PB	Start push button, N. O., **on** when starting.
STOP_PB	Stop push button, N. C., **off** when stopping.
RESET_PB	Reset push button, N. O., **on** when restoring station to initial state.
PROX1	Proximity sensor, **on** when part in position 1
PROX3	Proximity sensor, **on** when part in position 3
LS1	Limit switch that closes (**on**) drill is fully retracted
LS2	Limit switch that closes (**on**) when the drill is extended to the proper hole depth.
GATE_1	Gate 1 cylinder control, **on** to open gate 1, **off** closes gate.
GATE_2	Gate 2 cylinder control, **on** to open gate 2, **off** closes gate.
GATE_3	Gate 3 cylinder control, **on** to open gate 3, **off** closes gate.
DRILL_EXTEND	Drill extension cylinder control, **on** to extend drill.
DRILL_RETRACT	Drill retraction cylinder control, **on** to retract drill
DRILL_MOTOR	Drill motor control, **on** to cause drill to rotate.

The addresses associated with the physical inputs and outputs are:

Variable	Modicon	PLC-5	ControlLogix	Siemens	GE Fanuc
START_PB	100017	I:01/00	Local:1:I.Data.0	I4.0	%I97
STOP_PB	100018	I:01/01	Local:1:I.Data.1	I4.1	%I98
RESET_PB	100019	I:01/02	Local:1:I.Data.2	I4.2	%I99
PROX1	100021	I:01/04	Local:1:I.Data.4	I4.4	%I101
PROX3	100022	I:01/05	Local:1:I.Data.5	I4.5	%I102
LS1	100023	I:01/06	Local:1:I.Data.6	I4.6	%I103
LS2	100024	I:01/07	Local:1:I.Data.7	I4.7	%I104
GATE_1	000001	O:02/00	Local:2:O.Data.0	Q8.0	%Q1
GATE_2	000002	O:02/01	Local:2:O.Data.1	Q8.1	%Q2
GATE_3	000003	O:02/02	Local:2:O.Data.2	Q8.2	%Q3
DRILL_EXTEND	000004	O:02/03	Local:2:O.Data.3	Q8.3	%Q4
DRILL_RETRACT	000005	O:02/04	Local:2:O.Data.4	Q8.4	%Q5
DRILL_MOTOR	000006	O:02/05	Local:2:O.Data.5	Q8.5	%Q6

P6-9. Part Oiler Station Control. Using the function chart approach, implement the program for the following station that squirts oil onto each part that passes on a conveyor.

Figure P6.9 shows the layout of a station that squirts oil onto every part that comes down the conveyor. The station is only one in a series of stations along this conveyor. You are implementing ladder logic for this station only. You have no control over the conveyor, so assume it is always moving. This particular line is asynchronous, that is, each station processes parts at its own speed and does not coordinate its operation with any other station. Assume that the parts are spaced far enough apart that this operation can complete before the next part comes down the conveyor.

The conveyor consists of two parallel tracks. The part sensing and capturing mechanism is located between the two tracks. The part actually is attached to and rides upon an aluminum platform.

Upon initial startup, assume there are no parts in the station. When a part is detected by the proximity detector, PROX, the following steps are executed:

Capture part by activating ENGAGE_SOL and waiting for 4±0.1 seconds.

The oiler tip is lowered into position.

Open the oiler valve (OIL_VALVE) for 0.5±0.01 seconds, which squirts oil onto a certain place on the part.

The oiler is raised.

The ENGAGE_SOL is released and the part moves out of the station.

The operation then repeats.

The proximity sensor is inductive. PROX senses the platform before the platform reaches the engage position. You must assume that when the part is captured by the engaging mechanism, PROX remains **on**.

The engaging mechanism is driven by a single-action pneumatic cylinder, controlled by ENGAGE_SOL. When ENGAGE_SOL is energized, the “hook”

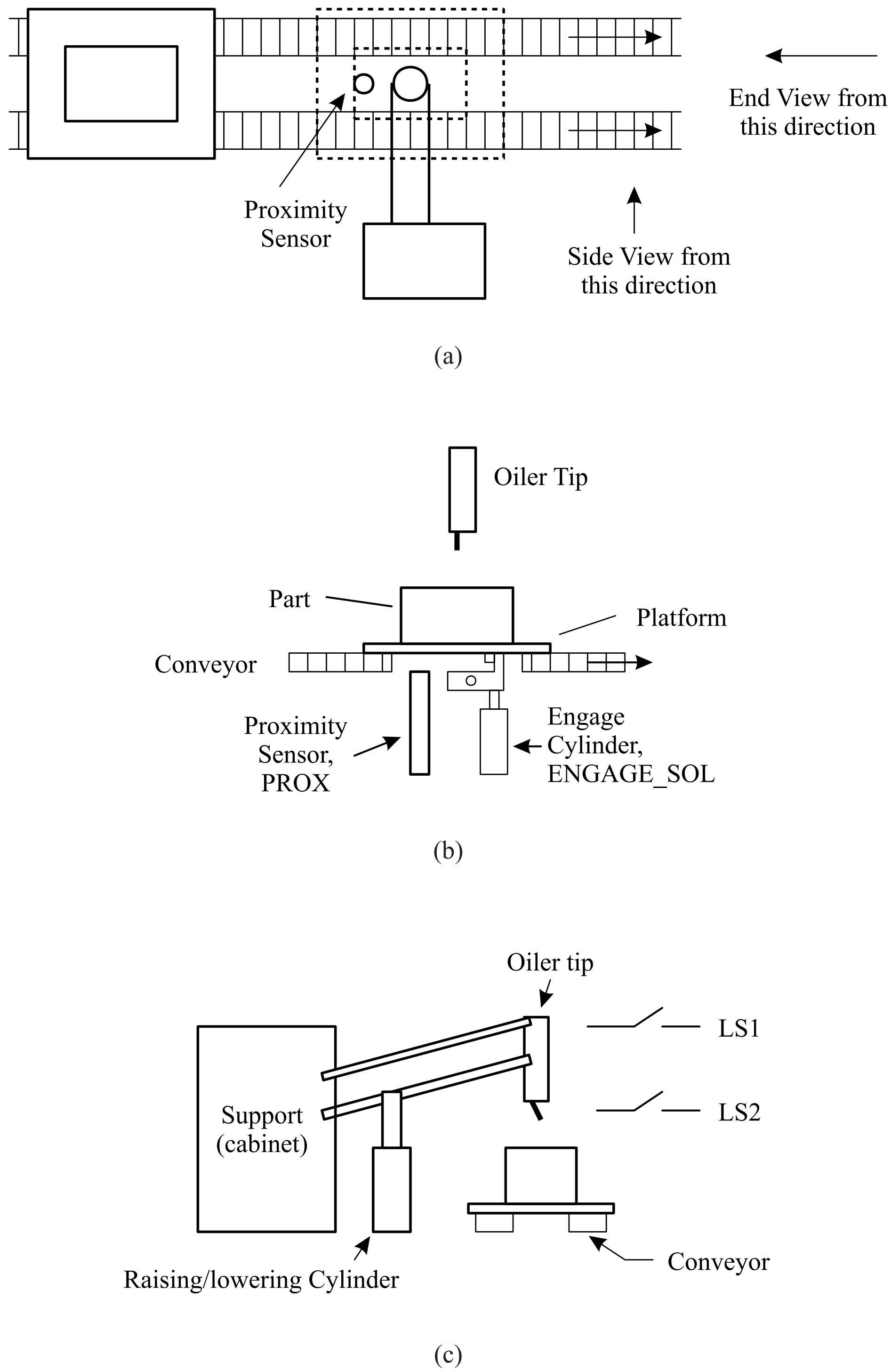

Figure P6.9. Oiler station: *(a)* top view; *(b)* side view; *(c)* end view.

moves up and remains in the "up" position as long as power is applied (turned **on**). The "hook" moves down when power is removed (turned **off**).

The mechanism used to lower and raise the oiler tip is driven by a double-acting pneumatic cylinder. When the OILER_DOWN output is energized, the oiler tip moves down and continues to move down as long as power is applied (turned **on**). When the OILER_UP output is energized, the oiler tip moves up and continues to move up as long as power is applied (turned **on**). The mechanism stops if neither output is **on** or if they are energized simultaneously. LS1 is **on** when the oiler tip is in the fully "up" position. LS2 is **on** when the oiler tip is in the fully "down" position.

When the start switch is pressed (turned **on**) for the first time only, the station assumes there is no part in the station and waits for the first part to arrive. When the stop switch is pressed (turned **off**), the operation should pause **except** when the oiler valve is opened to squirt the oil. If STOP_PB is pressed during the time the oil is being squirted, the step should be completed before pausing. Do not ignore STOP_PB during the step when oil is being squirted. When the operation is paused all outputs (except ENGAGE_SOL and OIL_VALVE) must be turned **off**. Pressing the start switch while the operation of the station is paused causes the station to resume its suspended operation.

If a timed step is paused, you need to determine if it is permissible to advance to the next step. It is permissible to advance to the next step if the timing interval has expired and no equipment will be damaged as a result of advancing to the next step.

There is a RESET_PB switch that when **on**, restarts the operation. When RESET_PB is **on**, the oiler is raised (if not in the "up" position), and the internal step is set so that the ladder logic program waits for the next part (but does not actually turn any outputs **on**). After a reset, START_PB must be pressed to actually restart the station. In other words, after RESET_PB is pressed and then released, the next press of the start switch is treated as the first time the start switch is pressed. However, the reset operation must be completed (oiler raised) before pressing START_PB has any effect. The RESET_PB switch must have no effect unless the operation is already paused.

Assume the following physical inputs and outputs.

Variable	Description
START_PB	Start push button, N. O., **on** when starting.
STOP_PB	Stop push button, N. C., **off** when stopping.
RESET_PB	Reset push button, N. O., **on** to restore station to initial state.
PROX	Proximity switch, **on** when platform is in station.
LS1	Limit switch, **on** (closed) when oiler tip is in raised position.
LS2	Limit switch, **on** (closed) when oiler tip is in lowered position.
ENGAGE_SOL	**On** to move up hook to engage platform in station, **off** releases platform to move down conveyor.
OILER_DOWN	**On** to lower oiler tip, **off** has no effect.
OILER_UP	**On** to raise oiler tip, **off** has no effect.
OIL_VALVE	**On** to open valve and squirt oil.

The addresses associated with the physical inputs and outputs are:

Variable	Modicon	PLC-5	ControlLogix	Siemens	GE Fanuc
START_PB	100001	I:01/00	Local:1:I.Data.0	I4.0	%I81
STOP_PB	100002	I:01/01	Local:1:I.Data.1	I4.1	%I82
RESET_PB	100003	I:01/02	Local:1:I.Data.2	I4.2	%I83
PROX	100017	I:02/00	Local:2:I.Data.0	I8.0	%I97
LS1	100018	I:02/01	Local:2:I.Data.1	I8.1	%I98
LS2	100019	I:02/02	Local:2:I.Data.2	I8.2	%I99
ENGAGE_SOL	000010	O:03/11	Local:3:O.Data.9	Q13.1	%Q10
OILER_DOWN	000011	O:03/12	Local:3:O.Data.10	Q13.2	%Q11
OILER_UP	000012	O:03/13	Local:3:O.Data.11	Q13.3	%Q12
OIL_VALVE	000013	O:03/14	Local:3:O.Data.12	Q13.4	%Q13

P6-10. Pressing Station Control. Using the function chart approach, implement the program for the following station that presses a pattern into each wood piece that passes on a conveyor.

Figure P6.10 shows the layout of a station that presses a pattern into each wood piece that passes on the conveyor. The station is only one in a series of stations along this conveyor. You are implementing ladder logic for this station only. You have control over the infeed conveyor. However, you have no control over the outfeed conveyor, so assume it is always moving. This particular line is asynchronous; that is, each station processes woods pieces at its own speed and does not coordinate its operation with any other station. Assume that the pieces are spaced far enough apart that this operation can complete before the next piece comes down the conveyor.

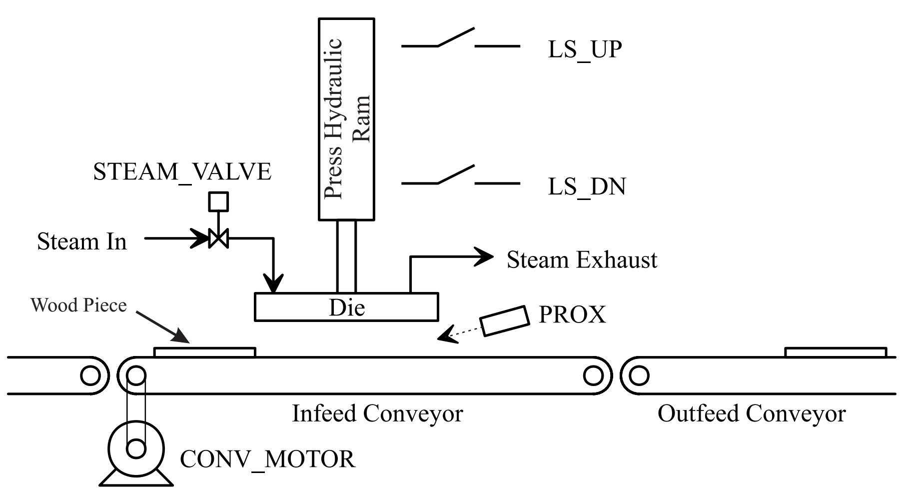

Figure P6.10. Pressing station (side view).