PROBLEMS

General instructions for the problems:

Using the function chart approach, write a ladder logic program for the application. Implement it for one of the following PLC ladder logic languages

ControlLogix/CompactLogix, **or**

MicroLogix/SLC-500, **or**

Siemens S7-1200/1500, **or** S7-300/400 (Portal or Classic), **or** S7-200, **or**

Modicon, **or**

Emerson PACSystems **or** VersaMax

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

The physical inputs, physical outputs, and internal variables for each problem are given in the problem.Note that for the problems, the ControlLogix addresses assume 1756-, 1734-, 1769-, or 1794-series I/O modules. If using 5069- or 5094-series modules, the address tags will need to be modified. **DO NOT** assign any more physical inputs!

Your solution should include the following:

1. Function chart of the process, showing the transitions between steps and the outputs active (or **on**) during each step.
2. Specify the PLC processor used.
3. Ladder logic diagram (with comments). For consistency among the different PLCs, use only tags/variables/symbols in the ladder logic. Use instructions consistent with the PLC processor.
4. Table listing additional internal memory (tags/variables/symbols) used and a brief description of their use. For the ControlLogix, Modicon, and PACSystem processors, also indicate the data type for the internal memory. For MicroLogix/SLC-500, indicate the memory address and for S7, indicate the memory address and data type.

Note to instructor: Break each problem into two assignments. For the first assignment, the students draw the function chart. The second assignment implements the ladder logic. For the second assignment, the students are allowed to use the correct function chart or their function chart if it is close to the correct solution. This approach will save the instructor from needing to grade many different ladder logic solutions.

P6-1. Using the function chart approach, implement the program for the following machine that transfers product to a carton blank and pushes it into a machine that finishes sealing the carton.

Figure P6.1 shows the layout of a station that transfers product (3" x 5" x 1/4" thick) from an infeed conveyor to the top of a carton blank and then pushes the product/carton down into the carton former that folds up the sides. Your program only does this part of the carton formation. Another part of the machine (and program) finishes forming the carton

around the product and sealing it. Your program is also responsible for feeding in the carton blank, which is a thin cardboard sheet printed and having the fold lines scored. Note that this problem is a simplified version of an actual machine.

Upon initial startup, assume that the transfer mechanism is up and left, the lift cylinder is down, and there is no sheet in the station. The operation steps are first described, followed by the details. When the machine is initially turned **on**, the following steps must be executed:

Wait for product on the infeed conveyor (IN_PROX senses).

Move horizontal pickup to the right position (HORIZ_RLS senses).

Simultaneously, activate the lift to move the product off the infeed conveyor to contact the vacuum pickup (LIFT_ULS senses) ***and*** feed carton blank into position. Pulses from an encoder attached to a pinch roller are used to gauge the blank carton position (60 pulses).

Activate the vacuum (VAC_OK senses)

Move horizontal pickup to the left position (HORIZ_LLS senses).

Simultaneously, deactivate the lift cylinder (LIFT_DLS senses) ***and*** extend the vertical cylinder to push the product on top of the carton blank and the carton blank down into the part of the machine that will finish folding the carton (VERT_DLS senses). Pushing the product/carton down will fold up the sides of the carton.

Release the vacuum (VAC_OK senses)

Retract the vertical cylinder until fully up (VERT_ULS senses).

The process repeats. Do not combine steps.

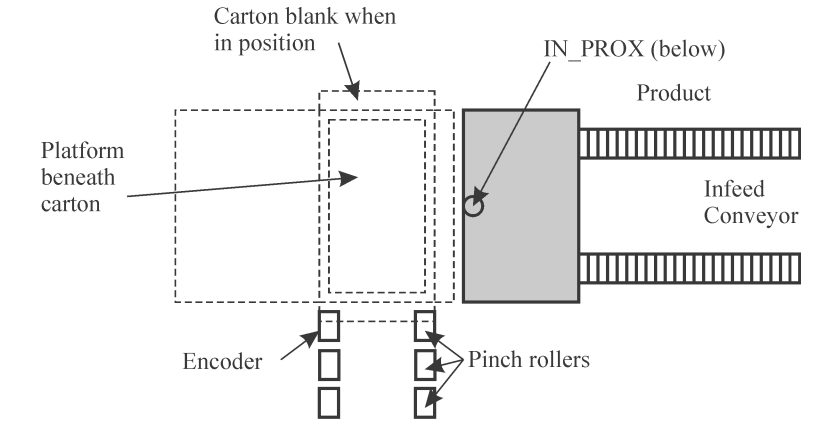
This part of the program has no control over the infeed conveyor, so assume it is always running. When the product is lifted off the infeed, new product is prevented from entering the station.

No parallel steps. When lifting the product off the infeed conveyor and feeding in the blank carton, make no assumptions about which movement will finish last. ***Both*** must be complete before advancing to the next step in the operation. When deactivating the lift cylinder so it is in position for the next piece and pushing the carton down, make no assumptions about which movement will finish last. ***Both*** must be complete before advancing to the next step in the operation.

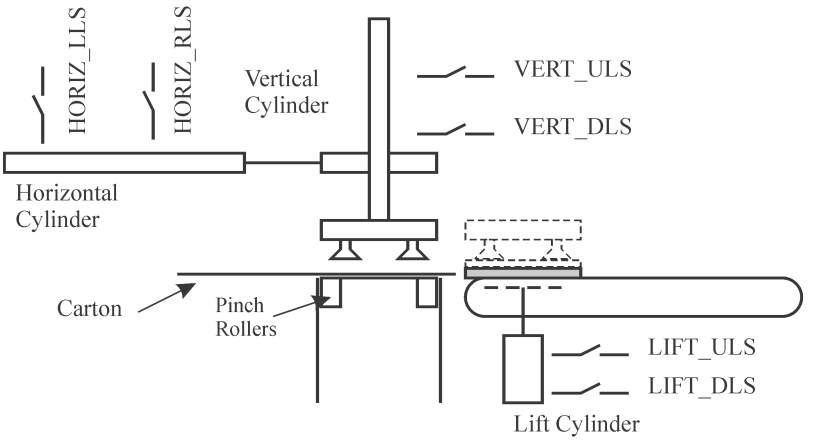
The carton blank is fed into the station by activating (turning **on**) the PINCH_ON output. This output activates a series of **pinch rollers** that engage the top and bottom of the blank, moving it onto a platform. One of the pinch rollers has an encoder on its shaft to measure the movement of the carton blank. The proper distance is 60 pulses of the PINCH_ENC physical input. The pulses are slow enough that they can be counted by the processor.

The IN_PROX **proximity sensor** is a photoelectric sensor that senses the leading edge of product in position on the infeed conveyor. When product is in position, IN_PROX turns **on**. IN_PROX remains **on** as the product is lifted from the conveyor and turns **off** sometime while the product is being moved horizontally left.

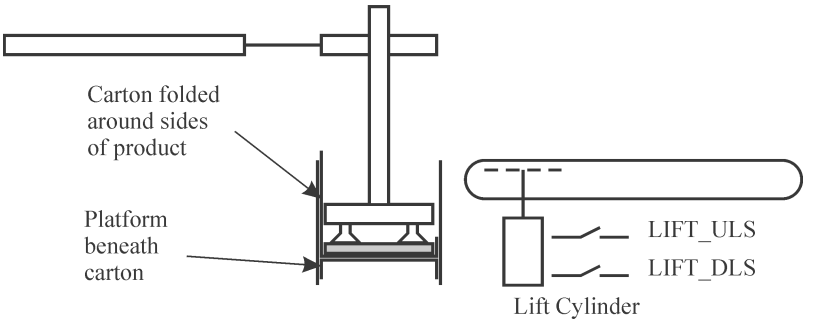
Lift Cylinder: The cylinder that controls the up/down movement of the mechanism that lifts the product off the infeed conveyor and contacts the vacuum cups on the transfer plate is a single-action pneumatic cylinder, controlled by the LIFT_UP output. When LIFT_UP is energized (turned **on**), the cylinder extends and lifts the product. When LIFT_UP is **off**, the cylinder retracts. The cylinder has two limit switches, LIFT_DLS,



(a)



(b)



(c)

Figure P6.1. Product transfer machine: *(a)* top view with product in station; *(b)* side view, right position shown as dashed lines; *(c)* product

which is **on** when the cylinder is retracted (down) and LIFT_ULS, which is **on** when the cylinder is extended (up).

Horizontal **Cylinder:** The horizontal cylinder controls the right/left movement of the transfer mechanism and is a double-action hydraulic cylinder, controlled by the HORIZ_RGT and HORIZ_LFT outputs. When HORIZ_RGT is energized (turned **on**), the cylinder extends and moves toward the right. When HORIZ_LFT is energized (turned **on**), the cylinder retracts and moves toward the left. When HORIZ_RGT and HORIZ_LFT are both **off**, the cylinder stops at the current position. When HORIZ_RGT and HORIZ_LFT are both **on**, the cylinder operation is not consistent (it may stop or move in either direction slowly). Therefore, HORIZ_RGT and HORIZ_LFT must not be **on** simultaneously. The cylinder has two limit switches, HORIZ_LLS, which is **on** hen the cylinder is retracted (left position) and HORIZ_RLS, which is **on** when the cylinder is extended (right position).

Vertical Cylinder: The vertical cylinder controls the down/up movement of the transfer mechanism and is a double-action hydraulic cylinder, controlled by the VERT_DN and VERT_UP outputs. When VERT_DN is energized (turned **on**), the cylinder extends and moves downward. When VERT_UP is energized (turned **on**), the cylinder retracts and moves upward. When VERT_DN and VERT_UP are both **off**, the cylinder stops at the current position. When VERT_DN and VERT_UP are both **on**, the cylinder operation is not consistent (it may stop or move in either direction slowly). Therefore, VERT_DN and VERT_UP must not be **on** simultaneously. The cylinder has two limit switches, VERT_ULS, which is **on** when the cylinder is retracted (up position) and VERT_DLS, which is **on** when the cylinder is extended (down position).

The **vacuum pickup** within the transfer mechanism consists of 4 cups and is controlled by the VACUUM_ON output. The VACUUM_ON output must remain **on** to attach and hold the product to the transfer mechanism. This output must be **off** to release the product. The VAC_OK input is **on** when the vacuum pressure is such that the product is attached and being held by the transfer mechanism. VAC_OK is **off** otherwise. When the VACUUM_ON output is turned **off** to release the product, the VAC_OK must be sensed **off** before retracting the vertical cylinder.

When the start switch is pressed (turned **on**) for the first time only, the station assumes there is no product on the infeed conveyor and the cylinders are in their initial position stated earlier in the problem. When the stop switch is pressed (turned **off**) the operation should pause. **Do not advance** to the next step when paused. When paused, all physical outputs **except** for the lift cylinder control (LIFT_UP) and the vacuum control (VACUUM_ON) must be turned **off**. Pressing the start switch while the operation of the station is paused causes the station to resume its suspended step. **DO NOT** retain a timer accumulator when paused. If paused in a timed step, the timer must continue to accumulate time.

A separate reset switch, RESET_PB, is provided which your part of the program uses to restore the system to its initial state. When the RESET_PB is pressed, the horizontal cylinder must be moved left, the vertical cylinder must be moved up, and the lift must be moved down. The reset operation is not complete until the horizontal cylinder is left, the vertical cylinder is up, and the lift cylinder is down. **Do not implement a function chart for the reset** operation. There is no mechanical interference. Assume that an operator is responsible for actually removing any material from the machine after a reset. The reset switch should be ignored if the machine is running. The start switch should be ignored when the reset operation is in progress. If the station is paused and RESET_PB is **on**, the operator

must release the RESET_PB before the start pushbutton switch can be used to restart the station.

Do not add any more timed steps to those explicitly stated in the problem. In other words, do not put a timer in a step unless it is stated that the step duration is a specific time. Assume the tolerance on all timer values is at most ±0.01 second.

Assume the following physical input and output tags/variables/symbols.

Tag/Var,/Symbol	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.
IN_PROX	Product-sensing proximity switch**, on** when product in position to be transferred. **Off** when product is being transferred or when the next one is not in position.
PINCH_ENC	Pinch roller encoder pulses to count to determine movement of blank carton
LIFT_DLS	Limit switch that closes (**on**) when the lift cylinder is retracted and in the down position.
LIFT_ULS	Limit switch that closes (**on**) when the lift cylinder is extended and in the up position.
HORIZ_RLS	Limit switch that closes (**on**) when the horizontal cylinder is extended and in the right position.
HORIZ_LLS	Limit switch that closes (**on**) when the horizontal cylinder is retracted and in the left position.
VERT_DLS	Limit switch that closes (**on**) when the vertical cylinder is extended and in the down position.
VERT_ULS	Limit switch that closes (**on**) when the vertical cylinder is retracted and in the up position.
VAC_OK	Limit switch that closes (**on**) when vacuum is sensed and therefore the suction cups have sufficient vacuum to pick up the product. **Off** otherwise.
PINCH_ON	Blank carton feed pinch roller control, **on** to activates a series of pinch rollers to move carton blank onto the platform, **off** stops the movement.
LIFT_UP	Lift cylinder extend control, **on** to extend cylinder, moving it up, **off** retracts cylinder, moving it down.
HORIZ_RGT	Horizontal cylinder extend control, **on** to extend cylinder, moving transfer mechanism right, **off** stops movement.
HORIZ_LFT	Horizontal cylinder retract control, **on** to retract cylinder, moving transfer mechanism left, **off** stops movement.
VERT_DN	Vertical cylinder extend control, **on** to extend cylinder, moving transfer mechanism down, **off** stops movement.
VERT_UP	Vertical cylinder retract control, **on** to retract cylinder, moving transfer mechanism up, **off** stops movement.

VACUUM_ON Product pickup vacuum control. **On** to turn on the vacuum generator. **Off** releases vacuum.

The addresses associated with the physical inputs and outputs are:

Tag/Var./Symbol	ControlLogix	MLogix	Siemens	Modicon	Emerson
START_PB	Local:1:I.Data.0	I:0/0	%I0.0	%I0.2.0	%I1
STOP_PB	Local:1:I.Data.1	I:0/1	%I0.1	%I0.2.1	%I2
RESET_PB	Local:1:I.Data.2	I:0/2	%I0.2	%I0.2.2	%I3
IN_PROX	Local:1:I.Data.3	I:0/3	%I0.3	%I0.2.3	%I4
PINCH_ENC	Local:1:I.Data.4	I:0/4	%I0.4	%I0.2.4	%I5
LIFT_DLS	Local:1:I.Data.5	I:0/5	%I0.5	%I0.2.5	%I6
LIFT_ULS	Local:1:I.Data.6	I:0/6	%I0.6	%I0.2.6	%I7
HORIZ_RLS	Local:1:I.Data.7	I:0/7	%I0.7	%I0.2.7	%I8
HORIZ_LLS	Local:1:I.Data.8	I:0/8	%I1.0	%I0.2.8	%I9
VERT_DLS	Local:1:I.Data.9	I:0/9	%I1.1	%I0.2.9	%I10
VERT_ULS	Local:1:I.Data.10	I:0/10	%I1.2	%I0.2.10	%I11
VAC_OK	Local:1:I.Data.11	I:0/11	%I1.3	%I0.2.11	%I12
PINCH_ON	Local:2:O.Data.0	O:0/0	%Q4.0	%Q0.3.0	%Q1
LIFT_UP	Local:2:O.Data.1	O:0/1	%Q4.1	%Q0.3.1	%Q2
HORIZ_RGT	Local:2:O.Data.2	O:0/2	%Q4.2	%Q0.3.2	%Q3
HORIZ_LFT	Local:2:O.Data.3	O:0/3	%Q4.3	%Q0.3.3	%Q4
VERT_DN	Local:2:O.Data.4	O:0/4	%Q4.4	%Q0.3.4	%Q5
VERT_UP	Local:2:O.Data.5	O:0/5	%Q4.5	%Q0.3.5	%Q6
VACUUM_ON	Local:2:O.Data.6	O:0/6	%Q4.6	%Q0.3.6	%Q7

P6-2. Using the function chart approach, implement the program for the following machine that places a rivet into a hole.

Figure P6.2 shows the layout of a station that places a rivet into a hole of a part assembly. This station is one station in a series of stations that the assembly moves through. The assembly is moved into position, a rivet is placed into the pivot arm, the arm is moved to place the rivet into the hole and then the arm is moved back to its initial position. Your program only does this station which is part of a longer manufacturing line. An upstream station places the two pieces together and a downstream station uses a hydraulic press to flatten the bottom of the rivet against the assembly, thus securing it. Note that this problem is a simplified version of an actual machine.

Upon initial startup, assume that the push cylinder is retracted, the pivot arm is in the clockwise position, in the up position, and in the retracted position and there is no assembly in the station. The operation steps are first described, followed by the details. When the machine is initially turned **on**, the following steps must be executed:

Wait for the INDEX_DN internal coil to turn **on**. This is the signal that a new assembly has moved into the station.

Extend the pusher cylinder to push a new rivet into the slot on the pivot arm.

Simultaneously, retract the pusher cylinder (PUSH_RLS senses) ***and*** rotate the pivot arm counterclockwise (also called anticlockwise) (ROT_CCWLS senses). Note that these TWO actions are ONE step.

Extend the horizontal cylinder to place the rivet over the hole (HORIZ_ELS senses).

Retract the vertical cylinder to place the rivet into the hole (VERT_DLS senses).

Retract the horizontal cylinder, leaving the rivet in the hole (HORIZ_RLS senses).

Simultaneously, extend the vertical cylinder to move the pivot arm up (VERT_ULS senses) ***and*** rotate the pivot arm clockwise to its initial position (ROT_CWLS senses). Note that these TWO actions are ONE step.

The process repeats. **Do not combine steps**.

No parallel steps. When retracting the pusher cylinder and rotating the pivot arm counterclockwise, make no assumptions about which movement will finish last. ***Both*** must be complete before advancing to the next step in the operation. When extending the vertical cylinder and moving the pivot clockwise, make no assumptions about which movement will finish last. ***Both*** must be complete before advancing to the next step in the operation.

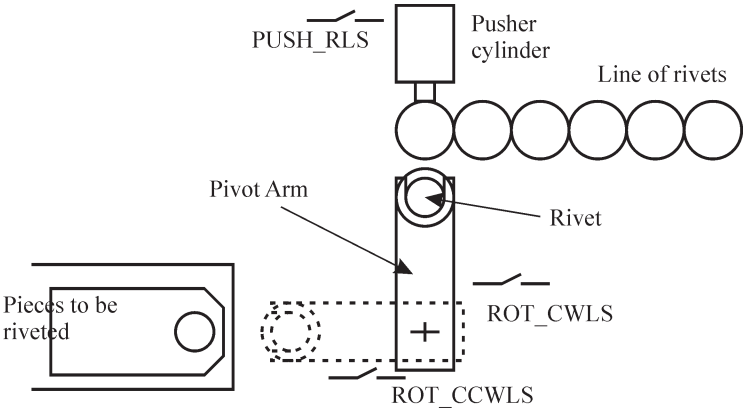
Your ladder logic does not control the mechanism that moves assemblies into and out of the station.

The INDEX_DN internal coil transitions **on** when a new assembly has been moved in this station and is ready to receive a rivet. This internal coil is controlled by another part of the ladder logic and thus your part of the program **must not** have any coil that refers to INDEX_DN. This internal coil will be **off** by the time the pivot arm is moved back.

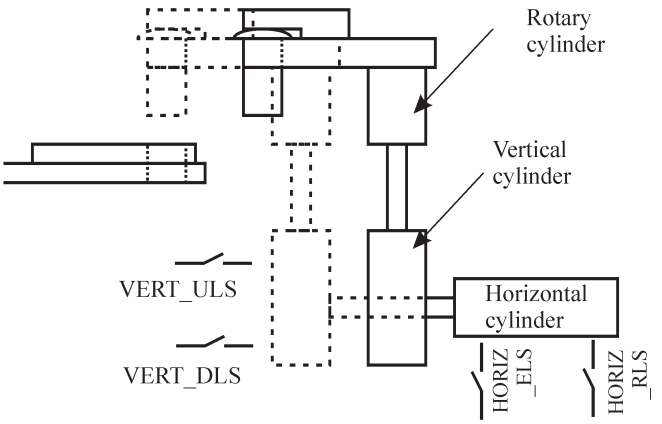
Pusher Cylinder: The cylinder that pushes the rivets into the pivot arm is a single-action pneumatic cylinder, controlled by the PUSH_EXT output. When PUSH_EXT is energized (turned **on**), the cylinder extends and pushes the leftmost rivet in the line into the pivot arm. When PUSH_EXT is **off**, the cylinder retracts. The cylinder has one limit switch, PUSH_RLS, which is **on** when the cylinder is retracted. There is no limit switch indicating when the pusher is extended. So, the extension of this cylinder must be timed. Assume that when PUSH_EXT is **on** for one second, a rivet has been pushed into the pivot arm. When the cylinder is retracted, the line of rivets advances to place the next one in position. But, that function is not a part of your ladder.

Rotary Cylinder: A rotary cylinder controls the rotational movement of the pivot arm mechanism and is a double-action pneumatic cylinder, controlled by the ROT_CW and ROT_CCW outputs. When ROT_CW is energized (turned **on**), the cylinder rotates the pivot arm clockwise. When ROT_CCW is energized (turned **on**), the cylinder rotates the pivot arm counterclockwise (anticlockwise). When ROT_CW and ROT_CCW are both **off**, the cylinder stops at the current position. When ROT_CW and ROT_CCW are both **on**, the cylinder operation is not consistent (it may stop or rotate in either direction slowly). Therefore, ROT_CW and ROT_CCW must not be **on** simultaneously. The cylinder has two limit switches, ROT_CWLS, which is **on** when the cylinder is in the clockwise position (end of arm at place to receive rivets) and ROT_CCWLS, which is **on** when the cylinder is in the counterclockwise position (rivet above hole).

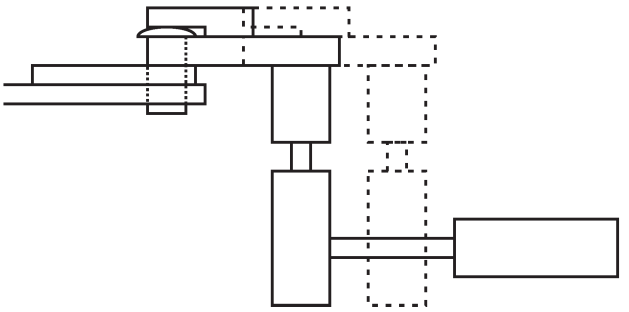
Horizontal Cylinder: The horizontal cylinder controls the in/out movement of the pivot arm mechanism and is a double-action pneumatic cylinder, controlled by the HORIZ_EXT and HORIZ_RET outputs. When HORIZ_EXT is energized (turned **on**), the cylinder extends and moves the pivot arm mechanism to place the rivet over the hole. When HORIZ_RET is energized (turned **on**), the cylinder retracts. When HORIZ_EXT and



(a)



(b)



(c)

Figure P6.2. Rivet insertion station: *(a)* top view, pivot arm clockwise and counterclockwise (dashed lines); *(b)* side view, arm counterclockwise up and horizontal retracted and horizontal extended (dashed lines); *(c)* side view, arm counterclockwise down, horizontal extended and retracted (dashed lines).

HORIZ_RET are both **off**, the cylinder stops at the current position. When HORIZ_EXT and HORIZ_RET are both **on**, the cylinder operation is not consistent (it may stop or move in either direction slowly). Therefore, HORIZ_EXT and HORIZ_RET must not be **on** simultaneously. The cylinder has two limit switches, HORIZ_RLS, which is **on** when the cylinder is retracted (inward position) and HORIZ_ELS, which is **on** when the cylinder is extended (outward position).

Vertical Cylinder: The vertical cylinder controls the down/up movement of the pivot arm mechanism and is a double-action pneumatic cylinder, controlled by the VERT_DN and VERT_UP outputs. When VERT_DN is energized (turned **on**), the cylinder retracts and moves downward. When VERT_UP is energized (turned **on**), the cylinder extends and moves upward. When VERT_DN and VERT_UP are both **off**, the cylinder stops at the current position. When VERT_DN and VERT_UP are both **on**, the cylinder operation is not consistent (it may stop or move in either direction slowly). Therefore, VERT_DN and VERT_UP must not be **on** simultaneously. The cylinder has two limit switches, VERT_ULS, which is **on** when the cylinder is extended (up position) and VERT_DLS, which is **on** when the cylinder is retracted (down position).

The station has START_PB and STOP_PB pushbuttons. When the start switch is pressed (turned **on**) for the first time only, the station assumes there is no assembly in the station and the cylinders are in their initial position stated earlier in the problem. When the stop switch is pressed (turned **off**) the operation should pause. **Do not advance** to the next step when paused. When paused, all physical outputs **except** for the pusher cylinder control (PUSH_EXT) must be turned **off**. When paused, any timers associated with a timed step must continue to run. Pressing the start switch while the operation of the station is paused causes the station to resume its suspended step.

A separate reset switch, RESET_PB, is provided which your part of the program uses to restore the system to its initial state. When the RESET_PB is pressed, the pusher cylinder must be retracted, the horizontal cylinder must be retracted, the vertical cylinder must be moved up, and the arm must be rotated clockwise. The reset operation is not complete until the pusher cylinder is retracted, the horizontal cylinder is retracted, the vertical cylinder is up, and the arm is clockwise. **It is not necessary to implement a function chart for the reset** operation. The only mechanical interference is that the horizontal cylinder must be retracted before the pivot arm is moved up and/or rotated. Assume that an operator is responsible for actually removing any material from the machine after a reset. The reset switch should be ignored if the machine is running. The start switch should be ignored when the reset operation is in progress. If the station is paused and RESET_PB is **on**, the operator must release the RESET_PB before the start pushbutton switch can be used to restart the station.

Do not add any more timed steps to those explicitly stated in the problem. In other words, do not put a timer in a step unless it is stated that the step duration is a specific time. Assume the tolerance on all timer values is at most ±0.01 second.

Assume the following physical input and output tags/variables/symbols.

Tag/Var./Symbol	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.

HORIZ_ELS	Limit switch that closes (**on**) when the horizontal cylinder is extended.
HORIZ_RLS	Limit switch that closes (**on**) when the horizontal cylinder is retracted.
PUSH_RLS	Pusher cylinder retracted limit switch**,** closes (**on**) when cylinder is retracted.
ROT_CWLS	Limit switch that closes (**on**) when the pivot arm in clockwise position.
ROT_CCWLS	Limit switch that closes (**on**) when the pivot arm in counterclockwise (anticlockwise) position.
VERT_DLS	Limit switch that closes (**on**) when the vertical cylinder is in the down position.
VERT_ULS	Limit switch that closes (**on**) when the vertical cylinder is in the up position.

HORIZ_EXT	Horizontal cylinder extend control, **on** to extend cylinder, **off** stops movement.
HORIZ_RET	Horizontal cylinder retract control, **on** to retract cylinder, **off** stops movement.
PUSH_EXT	Pusher cylinder control, **on** to extend, **off** retracts.
ROT_CW	Rotary cylinder clockwise control, **on** to rotate clockwise, **off** stops movement.
ROT_CCW	Rotary cylinder counterclockwise control, **on** to rotate counterclockwise, **off** stops movement.
VERT_DN	Vertical cylinder down movement control, **on** to retract cylinder, moving down, **off** stops movement.
VERT_UP	Vertical cylinder up movement control, **on** to extend cylinder, moving up, **off** stops movement.

Assume the following internal tags/variables/symbols:

INDEX_DN	Assembly indexing into position done. When **on**, a new assembly is in the station. **Off** by the time the pivot arm is moved back. Controlled by another part of the ladder logic in the PLC. **You must not have any output coil referring to INDEX_DN.**

The addresses associated with the physical inputs and outputs are:

Tag/Var./Symbol	ControlLogix	MLogix	Siemens	Modicon	Emerson
START_PB	Local:1:I.Data.0	I:0/0	%I0.0	%I0.2.0	%I1
STOP_PB	Local:1:I.Data.1	I:0/1	%I0.1	%I0.2.1	%I2
RESET_PB	Local:1:I.Data.2	I:0/2	%I0.2	%I0.2.2	%I3
HORIZ_ELS	Local:1:I.Data.3	I:0/3	%I0.3	%I0.2.3	%I4
HORIZ_RLS	Local:1:I.Data.4	I:0/4	%I0.4	%I0.2.4	%I5
PUSH_RLS	Local:1:I.Data.5	I:0/5	%I0.5	%I0.2.5	%I6
ROT_CWLS	Local:1:I.Data.6	I:0/6	%I0.6	%I0.2.6	%I7
ROT_CCWLS	Local:1:I.Data.7	I:0/7	%I0.7	%I0.2.7	%I8