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

Modicon, **or**

Allen-Bradley ControlLogix, **or**

Allen-Bradley PLC-5/SLC-500, **or**

Siemens S7-300/400, **or** S7-200, **or** S7-1200, **or**

GE

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. **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 variables/symbols/tags in the ladder logic. Use instructions consistent with the PLC processor.
4. Table listing additional internal memory (variables/symbols/tags) used and a brief description of their use. For the Modicon, Allen-Bradley ControlLogix and PACSystemprocessors, list the internal variables/tags and the data type. For the other processors, list the internal variables/symbols and the associated memory address.

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. Rail Welding Station Control. Using the function chart approach, implement the program for the following station that welds railroad rails together.

Figure P6.1 shows the general layout of a station that welds 40-foot lengths of railroad rail into a 1/4-mile length of rail (called a string). In summary, individual rails are fed into a track, a ram pushes the rail into the welder, the robot welder welds the rail onto the current "string" of welded rail, and then the entire string is moved 40 feet to be in position to weld on the next piece. The operation is described in the following manner:

When initially started, assume there is a rail piece in position 1 in Figure P6.1.

Before continuing with the operation, two conditions must be verified: (1) There must be a rail piece in the feeder (sensed by PROX374); and (2) the

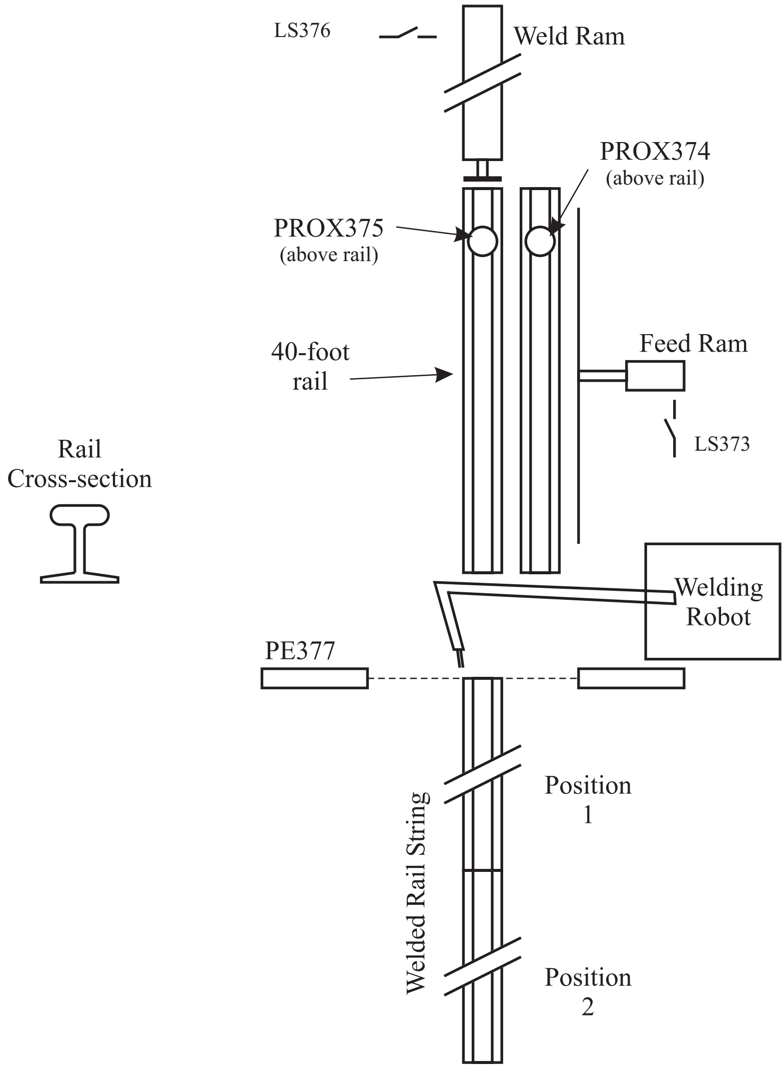

Figure P6.1. Rail welding station.

WELD_ENABLE internal coil must be **on**. Both of these conditions must be met before the next piece can be moved onto the welding track.

To bring in the next 40-foot rail piece, the feed hydraulic ram control, FRAM_EXT is turned **on** to push a rail piece onto the welding track (sensed by PROX375).

Then the WRAM_EXT cylinder control is energized to extend the welding hydraulic ram to push the 40-foot rail piece into the welding position (sensed by PE377 **off**).

The welding ram cylinder controls are then turned **off** so the new rail piece is held in the welding position (butting against the end of the existing "string" of welded rail).

The welding robot is then activated (with ROBOT_START) to actually produce the weld. The welding robot turns **on** the ROBOT_FINI input to the PLC to signal that the welding operation is complete.

The string of welded rail is then pulled forward by activating the STRING_FWD output. When the proximity sensor PE377 turns **on**, the string is in position to weld on the next 40-foot piece.

The WRAM_RET cylinder control is energized to move the welding hydraulic ram back into position, ready to push the next piece into the welder. The LS376 limit switch turns **on** when the ram is retracted.

The operation then repeats (starting with verifying the two conditions).

The WELD_ENABLE internal coil signals that the current welded rail string is less than 1320 feet (1/4 mile) and thus the next 40-foot piece in the feeder can be welded onto the current string. When the string is 1320 feet long, the WELD_ENABLE internal coil is turned **off** and a new string is started by moving one rail into position 1. **Your part of the ladder does not move this first rail piece into position**. Another part of the PLC program does this part of the operation.

The feeder area has two proximity sensors, PROX374 and PROX375. When PROX 374 is **on**, a 40-foot rail piece is present in the feeder. You are not concerned with the mechanism that places the piece into the feeder area. You just need to sense that the rail is present before activating the FRAM_EXT control to push the 40-foot rail piece into the weld track. PROX375 is **on** when the rail has been pushed by the feed ram onto the welding rail.

The weld position is sensed by the photoelectric eye, PE377. When the active end (end to be welded) of the welded string is moved into the proper position, PE377 transitions **on**. When the 40-foot piece is moved in to contact the active end, PE377 turns **off**.

The feed ram is a single-action hydraulic cylinder, controlled by FRAM_EXT. Once the FRAM_EXT output is energized (**on**), the ram extends and pushes the 40-foot rail piece onto the welding track. When FRAM_EXT is **off**, the ram retracts. The ram retracts very quickly and so you may assume that when FRAM_EXT is turned **off** after the rail piece is placed onto the weld track, the ram has retracted sufficiently far so that it does not interfere with the weld ram. In normal operation, you do not need to detect that it has retracted (if the ram does not retract within a certain time period an alarm is activated, which is **not** part of the problem). However, during reset, you do need to detect when the feed ram is retracted. LS373 is **on** when the feed ram is retracted.

The welding ram is a double-action hydraulic cylinder with two controls. While the WRAM_EXT control is energized, the ram extends (moves out) as long as power is applied (WRAM_EXT is **on**). While the WRAM_RET control is energized, the ram retracts (moves in) as long as power is applied (WRAM_RET is **on**).The ram stops at its current position when either control is turned **off**. The ram will not move if both WRAM_EXT and WRAM_RET are energized simultaneously. There are mechanical stops to prevent extension or retraction beyond certain limits.

The LS376 limit switch is **on** when the welding ram is fully retracted.

The welding robot is a piece of equipment separate from the PLC. It is programmed separately to move a welding head along a programmed path and thus weld the two rail pieces together. The PLC only commands the welding robot to start its operation (with ROBOT_START) and receives an indication (ROBOT_FINI) that the welding operation is complete. The ROBOT_START output must remain **on** during the robot's operation (unless paused). If ROBOT_START is turned **off**, the robot pauses its preprogrammed operation. The ROBOT_FINI input to the PLC is turned **on** when the welding robot has finished its operation.

The STRING_FWD output commands a mechanism that moves the welded string (up to 1320' long) to place the active end in a position where the next piece can be welded. You are not concerned about the details of the mechanism. It basically activates a roller conveyor to move the string. The STRING_FWD output must be **on** to move the welded rail string.

When the start switch is pressed (turned **on**) for the first time only, the station assumes there is one 40-foot rail in position 1. When the stop switch is pressed (turned **off**) the operation should pause and all outputs (except STRING_FWD) must be turned **off**. Pressing the start switch while the operation of the station is paused causes the station to resume its suspended step. The STRING_FWD output must remain **on** if the station is paused in the step where the rail string is moved in order to accurately place the active end.

With one exception, **do not advance** to the next step when paused. If the operation is paused when STRING_FWD is **on**, the welded rail string must continue to be moved into position and then advance to the next step when the string is in position.

A separate reset switch, RESET_PB, is provided which resets any internal states so that when the start switch is pressed, no rails are assumed present in the station. Obviously, the operator must clear everything out of the welding area before resuming operation after a reset operation. Also, the reset operation should cause the welding and feed rams to retract and the reset operation is not complete until both rams are retracted. Make no assumption about which ram requires the longest time to retract. The reset switch should be ignored if the station is running or when the string is moving forward. 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 0.01 seconds.

Assume the following physical inputs and outputs. DO NOT assign any more inputs !!

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.
LS373	Limit switch that closes (**on**) when feeder ram is retracted.
PROX374	Proximity sensor, **on** (closed) when 40-foot rail piece in feeder area; **off** as rail is pushed onto weld rail or when no rail piece is present.
PROX375	Proximity sensor, **on** (closed) when 40-foot rail piece on weld track, ready to be pushed into welder; **off** as rail is pushed into welder or when no rail piece is present.
LS376	Limit switch that closes (**on**) when weld ram is retracted.
PE377	Photo-electric eye that senses active end of welded rail. See above for description.

ROBOT_FINI	Indication from welding robot that is **on** when operation is complete.
FRAM_EXT	Feed ram extend control, **on** to extend ram, **off** causes ram to retract.
WRAM_EXT	Weld ram extend control, **on** to extend ram.
WRAM_RET	Weld ram retract control, **on** to retract ram.
ROBOT_START	Command to welding robot to start operation. Welding robot senses an **off**-to-**on** transition to start. It may remain **on** while welding is occurring.
STRING_FWD	Welded string movement control, **on** to move welded string forward and down roller conveyor.

Assume the following internal coil:

WELD_ENABLE	Signals that the next 40-foot piece of rail can be welded onto current string. **On** when next 40-foot piece of rail can be welded onto current string. **Set by another part of the ladder (MUST NOT appear as an output in your part of the ladder)**

The addresses associated with the physical inputs and outputs are:

Variable	Modicon	ControlLogix	PLC-5	Siemens	GE
START_PB	%I0.1.0	Local:1:I.Data.0	I:01/00	I0.0	%I1
STOP_PB	%I0.1.1	Local:1:I.Data.1	I:01/01	I0.1	%I2
RESET_PB	%I0.1.2	Local:1:I.Data.2	I:01/02	I0.2	%I3
LS373	%I0.1.3	Local:1:I.Data.3	I:01/03	I0.3	%I4
PROX374	%I0.1.4	Local:1:I.Data.4	I:01/04	I0.4	%I5
PROX375	%I0.1.5	Local:1:I.Data.5	I:01/05	I0.5	%I6
LS376	%I0.1.6	Local:1:I.Data.6	I:01/06	I0.6	%I7
PE377	%I0.1.7	Local:1:I.Data.7	I:01/07	I0.7	%I8
ROBOT_FINI	%I0.1.8	Local:1:I.Data.8	I:01/10	I1.0	%I9
FRAM_EXT	%Q0.2.0	Local:2:O.Data.0	O:02/00	Q4.0	%Q1
WRAM_EXT	%Q0.2.1	Local:2:O.Data.1	O:02/01	Q4.1	%Q2
WRAM_RET	%Q0.2.2	Local:2:O.Data.2	O:02/02	Q4.2	%Q3
ROBOT_START	%Q0.2.3	Local:2:O.Data.3	O:02/03	Q4.3	%Q4
STRING_FWD	%Q0.2.4	Local:2:O.Data.4	O:02/04	Q4.4	%Q5

The data type or address associated with the internal variable is:

	Modicon	CLogix	PLC-5	Siemens	GE	
Variable	Data Type	Data Type	Addr.	Addr.	Type	Addr.
WELD_ENABLE	BOOL	BOOL	B35/241	M382.1	BOOL	%M2025

P6-2. Lumber Cutter Station Control. Using the function chart approach, implement the program for the following station that cuts a long piece of lumber into shorter pieces.

Figure P6.2 shows the layout of a station that cuts a long piece of lumber into shorter pieces. This station is only one in a series of stations along this conveyor. The station

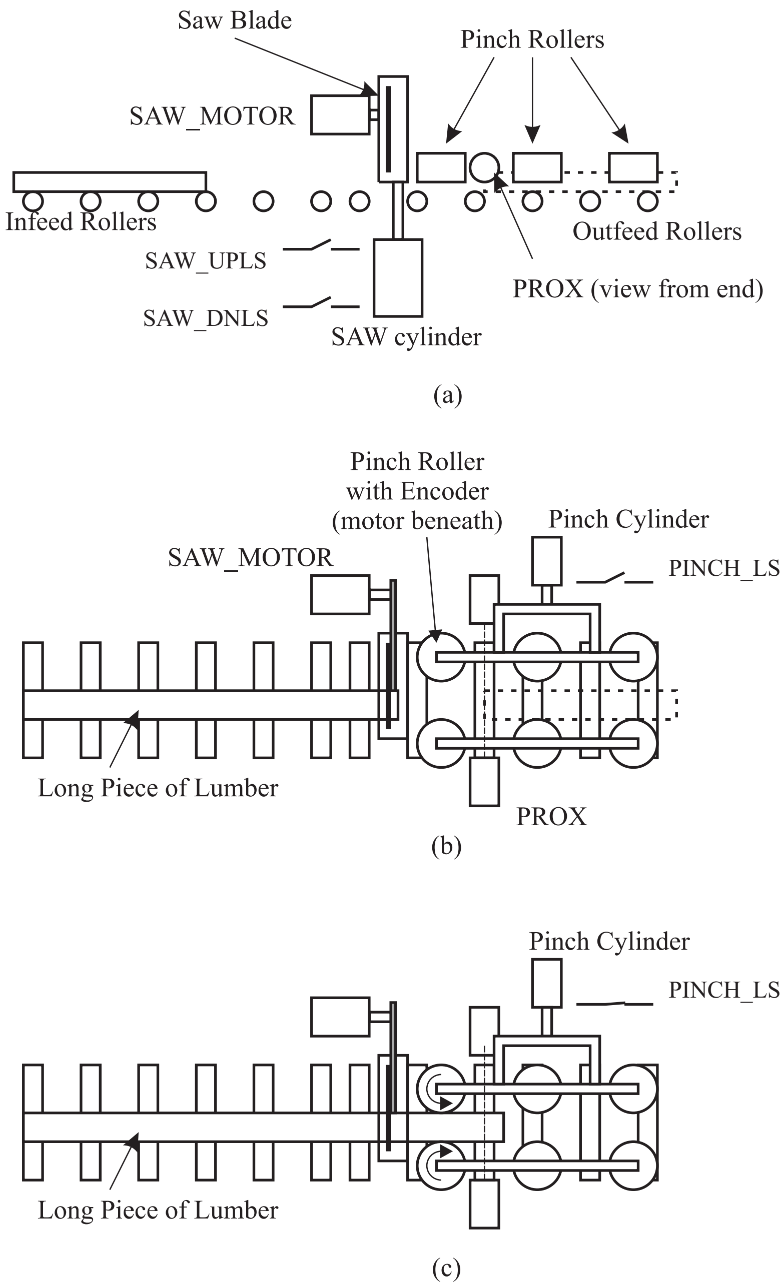

Figure P6.2. View of lumber cut station: *(a)* from top; *(b)* from side; (c) from side with pinch rollers feeding long piece.

upstream of this station disassembles banded stacks of the long lumber pieces and feeds them to this station one at a time. You are implementing ladder logic for this station only. However, your logic controls this section of the conveyor. This particular line is asynchronous, that is, each station processes parts at its own speed and does not directly coordinate its operation with any other station.

Upon initial startup, assume that there are no lumber pieces in the station. The initial states of the main mechanical controls are: saw up and pinch rollers open. When the piece of lumber is moved into the station by the conveyor and detected by PROX, the following steps are executed:

Extend the pinch cylinder to capture the end of the long lumber piece (PINCH_LS senses).

Activate the pinch rollers to pull the lumber piece a certain amount through the station. Pulses from an encoder attached to the first pinch roller are used to gauge the length of the piece, which is 1270 encoder pulses.

Retract the saw cylinder to pull down the saw and cut the lumber (SAW_DNLS senses).

Extend the saw cylinder (SAW_UPLS limit switch senses) to move it out of the path of the lumber.

Allow the cut piece of lumber to move out by activating the conveyor drive motor. This action also initially moves in the long piece of lumber for the next cut. The outfeed rollers rotate faster than the infeed rollers, so there is guaranteed a gap between pieces that can be sensed by PROX.

The operation then repeats.

The PROX **proximity sensor** is a thru-beam photoelectric proximity sensor that senses the presence/absence of lumber between the first and second set of pinch rollers. PROX is **off** when lumber is present between the transmitter and receiver and **on** when there is no lumber between the transmitter and receiver. The outfeed rollers rotate faster than the infeed rollers, so there is guaranteed a gap between pieces. The PROX input turns **on** for a period of time before it turns **off** to sense the end of the long lumber piece.

The **infeed and outfeed rollers** are controlled by the FEED_MOTOR physical output. This output must be **on** to transport the end of the long lumber piece into the station and to move out the cut piece. The outfeed rollers rotate faster than the infeed rollers. The infeed and outfeed rollers must be off while the pinch rollers are pinching the lumber, while advancing the lumber, and while the saw is moved up and down.

The **pinch rollers** are controlled by the PINCH_ROLL physical output. This output must be **on** in order to advance the lumber piece to the proper cut position. The first pinch roll has an encoder on its shaft to measure the movement of the lumber piece. The length of the cut pieces is 1270 pulses of the PINCH_ENC physical input (plus the distance between the saw blade and the PROX beam).

Pinch Cylinder: The pinch cylinder is a single-action pneumatic cylinder, controlled by the PINCH_CYL output. When PINCH_CYL is energized, the cylinder extends and presses the pinch rollers to hold the lumber. When PINCH_CYL is de-energized, the cylinder retracts. The cylinder has only one limit switch, PINCH_LS, which is **on** when the cylinder is extended. There is no limit switch detecting when the cylinder is retracted. There is no need for a retract limit switch since the outfeed rollers will move out the cut piece when it is retracted.

The pinch cylinder control must remain **on** while the pinch roller control (PINCH_ROLL) advances the lumber piece and while the saw is moved up and down.

Saw Cylinder: The saw cylinder is a single-action pneumatic cylinder, controlled by the SAW_DOWN output. When SAW_DOWN is energized, the cylinder retracts and pulls the rotating saw blade down to cut the lumber. When SAW_DOWN is de-energized, the cylinder extends and the saw blade moves clear of the lumber path. The SAW_DNLS limit switch is **on** when the cylinder is retracted (saw down) and the SAW_UPLS limit switch is **on** when the cylinder is extended (saw up).

The **saw motor** is controlled by the SAW_MOTOR output and must be **on** while the station is running.

Since the last short piece will be shorter than desired, your part of the ladder logic must signal that a piece is too short, which is used by the next station to push it off the conveyor. This situation is detected by sensing the state of PROX while advancing the piece with the pinch rollers. If PROX is **on** during this part of the operation, set (latch) the SHORT_PIECE internal coil. Do not worry about resetting (unlatching) this internal coil. Assume another part of the ladder takes care of doing it. Also, do not alter the operation of the station if the piece is too short.

When the start switch is pressed (turned **on**) for the first time only, the station assumes there is no lumber in the station. When the stop switch is pressed (turned **off**) the operation should pause except when the saw blade is moving down. For all other steps **do not advance** to the next step when paused. When paused, all outputs **except** for the pinch cylinder and saw cylinder controls must be turned **off**. If the stop switch is pressed while the saw blade is moving down, the saw motor 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 restores the system to its initial state. The reset operation should cause the pinch cylinder to retract and the saw cylinder to extend (which is what will happen when their controls are **off**) and the reset operation is not complete until the saw is up. Assume that an operator is responsible for actually removing any lumber from the station after a reset. The reset switch should be ignored if the station 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 0.01 seconds.

Assume the following physical inputs and outputs. DO NOT assign any more inputs!!

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	Lumber-sensing proximity switch, **off** when lumber interrupts beam, **on** otherwise.
PINCH_LS	Limit switch that closes (**on**) when the pinch cylinder is extended and therefore in contact with the lumber piece.
SAW_UPLS	Limit switch that closes (**on**) when the saw cylinder is extended (saw up).
SAW_DNLS	Limit switch that closes (**on**) when the the saw cylinder is retracted (saw down).

PINCH_ENC	Pinch roller encoder pulses to count to determine cut length.
FEED_MOTOR	Infeed and outfeed rollers, **on** to move lumber in/out of station.
PINCH_ROLL	Pinch roller motor, **on** to move lumber to cut position.
SAW_MOTOR	Saw blade drive motor, **on** to rotate saw. Must be **on** while station running.
PINCH_CYL	Pinch cylinder control, **on** to clamp lumber, **off** releases the clamp.
SAW_DOWN	Saw up/down cylinder control, **on** to retract (move saw down), **off** to extend (move saw up).

Assume the following internal coil:

Variable	Description
SHORT PIECE	Indicates current piece is short, set (latched) **on** by your ladder when current piece moving through the pinch rollers. YOUR PART OF THE LADDER MUST set (latch) this coil. Another part of the ladder will reset (unlatch) it.

The addresses associated with the physical inputs and outputs are:

Variable	Modicon	ControlLogix	PLC-5	Siemens	GE
START_PB	%I0.1.0	Local:1:I.Data.0	I:01/00	I0.0	%I1
STOP_PB	%I0.1.1	Local:1:I.Data.1	I:01/01	I0.1	%I2
RESET_PB	%I0.1.2	Local:1:I.Data.2	I:01/02	I0.2	%I3
PROX	%I0.1.3	Local:1:I.Data.3	I:01/03	I0.3	%I4
PINCH_LS	%I0.1.4	Local:1:I.Data.4	I:01/04	I0.4	%I5
SAW_UPLS	%I0.1.5	Local:1:I.Data.5	I:01/05	I0.5	%I6
SAW_DNLS	%I0.1.6	Local:1:I.Data.6	I:01/06	I0.6	%I7
PINCH_ENC	%I0.1.7	Local:1:I.Data.7	I:01/07	I0.7	%I8
FEED_MOTOR	%Q0.2.0	Local:2:O.Data.0	O:02/00	Q4.0	%Q1
PINCH_ROLL	%Q0.2.1	Local:2:O.Data.1	O:02/01	Q4.1	%Q2
SAW_MOTOR	%Q0.2.2	Local:2:O.Data.2	O:02/02	Q4.2	%Q3
PINCH_CYL	%Q0.2.3	Local:2:O.Data.3	O:02/03	Q4.3	%Q4
SAW_DOWN	%Q0.2.4	Local:2:O.Data.4	O:02/04	Q4.4	%Q5

The addresses or data types associated with the internal variable is:

	Modicon	CLogix	PLC-5	Siemens		GE
Variable	Data Type	Data Type	Addr.	Addr.	Type	Addr.
SHORT_PIECE	BOOL	BOOL	B3/68	M9.1	BOOL	%M137

P6-3. Yogurt Container Mover Control. Using the function chart approach, implement the program for the following system that places empty yogurt containers into a yogurt filler.

Figure P6.3 shows the layout of a station that moves a stack of empty yogurt containers from the place where a worker has placed them and drops them into the filling machine. This system is only a part of the overall operation of packaging yogurt. You are implementing ladder logic for this part of the packaging operation only. There is no START_PB and STOP_PB because there is an overall start and stop for the filling machine.