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**
Micro800, **or**
MicroLogix/SLC-500

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

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

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, CompactLogix, and Micro800, also indicate the data type for the internal memory. For MicroLogix and SLC-500, indicate 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. Using the function chart approach, implement the program for the following machine that injects ink into a highlighter marker.

Figure P6.1 shows the layout of a machine that injects ink into the felt "inside" of two highlighter markers. The markers are already mounted in a "belt" carrier and are moved in pairs through the stations that insert the felt, inject the ink, place the plug in the bottom, and then add the cap. You are only responsible for the station that injects the ink. The other aspects of the operation are controlled by other parts of the PLC program.

Also, assume that when the prox senses that the markers are indexed into position, both markers are present. Your solution will not be concerned with how to handle the case where no markers are present (for example, when starting a production run).

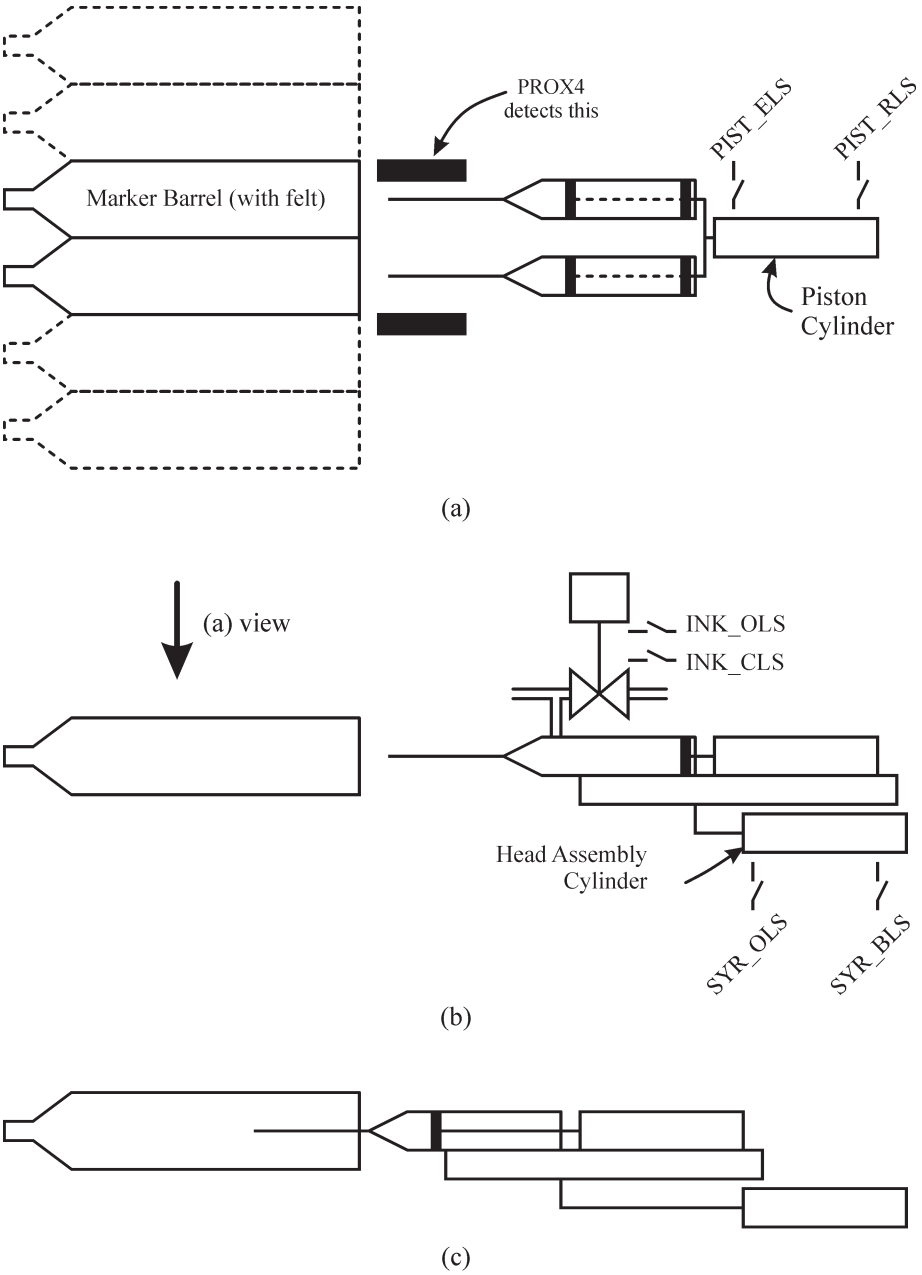


Figure P6.1. Marker ink injection station: *(a)* top view; *(b)* side view, head back; *(c)* side view, head out and piston extended.

Upon initial startup, wait for the prox to sense the pin on the belt carrier. In summary, after the prox senses the presence of the markers, the syringe head moves out, the syringe piston is extended, there is a small delay, the syringe head moves back, and the syringe

piston is retracted. The initial states of the main mechanical controls are: syringe head back, syringe piston retracted, and ink valve closed.

The following steps are executed:

Wait for PROX4 to turn **on** to indicate that the belt has indexed the markers into position. The markers are now in position to be filled with ink.

Move syringe head out to insert the syringe needles into the felt inside the marker barrel. SYR_OLS senses when the syringe head is in position.

Extend syringe piston to inject ink into the marker felt. PIST_ELS senses when the piston is in its fully extended position. Note that the ink valve must be closed during this step (or the ink will be simply driven back to the reservoir).

Delay 0.5 seconds to allow the ink to finish flowing into the felt.

Move syringe head back. SYR_BLS senses when the syringe head assembly is fully retracted and the needles are out of the marker barrel.

Retract syringe piston (PIST_RLS senses). The ink valve must be open in this step to fill the syringe with ink for the next markers.

The operation then repeats.

PROX4: The PROX4 proximity switch senses a pin on the belt holding the markers and indicates that the markers are in position. When **on**, the markers are in position. When **off**, the markers are not in position (in the process of being moved).

Syringe Head Cylinder: The cylinder that controls the out/back movement of the syringe head assembly is a double-action pneumatic cylinder, controlled by the SYR_OUT and SYR_BACK outputs. When SYR_OUT is energized (turned **on**), the cylinder extends and moves the head assembly out, inserting the syringe needles into the marker barrel. When SYR_BACK is energized (turned **on**), the cylinder retracts, moving the head assembly back, retracting the needles. When SYR_OUT and SYR_BACK are both **off**, the cylinder stops at its present position. When SYR_OUT and SYR_BACK are both **on**, the cylinder operation is not consistent (it may stop or move in either direction slowly). The cylinder has two limit switches, SYR_OLS, which is **on** when the cylinder is extended and the needles are fully inserted into the marker barrels and SYR_BLS, which is **on** when the cylinder is retracted (fully back).

Syringe Piston Cylinder: The cylinder that controls the extend/retract movement of the syringe piston is a double-action pneumatic cylinder, controlled by the PIST_EXT and PIST_RET outputs. When PIST_EXT is energized (turned **on**), the syringe piston extends, injecting ink into the marker. When PIST_RET is energized (turned **on**), the syringe piston retracts. When PIST_EXT and PIST_RET are both **off**, the piston stops at its present position. When PIST_EXT and PIST_RET are both **on**, the piston operation is not consistent (it may stop or move in either direction slowly). The cylinder controlling the piston has two limit switches, PIST_ELS, which is **on** when the piston is extended and the ink has been injected into the marker and PIST_RLS, which is **on** when the piston is retracted (fully back).

The **ink valve** is basically a single-action solenoid controlled by the INK_VLV output. The INK_VLV physical output must remain **on** to open the valve. This output must be **off** to close the valve. The INK_OLS input is **on** when the valve is open, and the INK_CLS input is **on** when the valve is closed. The ink valve **must be open** when the syringe piston is **being retracted**, to bring ink into the syringe for the next marker. The ink valve must be closed all other times. If the piston is in its retracted position, the valve must be closed.

There is no start/stop switch for your part of the ladder. Instead, there is an internal coil, I_RUN that is **on** when the operation is to happen. When I_RUN is **off**, the operation should pause at its current step. **Do not advance** to the next step when paused. Also, when I_RUN is **off**, the **on** transition of PROX4 should be ignored (this is how the system disables ink injection at the start of a production run).

When paused, all outputs should be **off except** for the following:

1. The ink valve control, INK_VLV must not be turned **off** (if it is **on**).

When I_RUN turns back **on** while the operation of the station is paused, the station should resume its suspended step. Note that it is not necessary to retain the timer accumulator when paused.

The start and stop switches are absent from your part of the ladder because they control the overall operation of the marker production line. In order to reliably start/stop/pause the operation of the ink injection, the belt carrier and other production line stations must be controlled. These parts of the operation are outside the scope of this problem. The other part of the logic controls I_RUN. **You must not have any output coil referring to I_RUN.**

A separate reset internal coil, I_RESET, is provided which restores the production line to its initial state. The reset operation should cause the syringe head to be moved back, and the syringe piston retracted (valve open while retract). The reset operation is not complete until the head is back, the syringe piston is retracted and the ink valve is closed. There is no mechanical interference between the head and syringe piston. If the syringe piston is in the retracted position, the ink valve must be closed during the reset operation. Assume that an operator is responsible for actually removing any markers from the station after a reset. I_RESET should be ignored if I_RUN is **on**. The I_RUN internal coil used to restart the machine should be ignored when the reset operation is in progress. **You must not have any output coil referring to I_RESET.**

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 seconds.

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

Tag/Var./Symbol	Description
PROX4	Belt pin-sensing inductive proximity switch, **on** when markers in position, **off** otherwise.
SYR_OLS	Limit switch that closes (**on**) when the syringe head assembly is in the out position therefore the syringe needles are inserted into the marker barrel. **Off** otherwise.
SYR_BLS	Limit switch that closes (**on**) when the syringe head assembly is in the back position. **Off** otherwise.
PIST_ELS	Limit switch that closes (**on**) when the syringe piston is fully extended. **Off** otherwise.
PIST_RLS	Limit switch that closes (**on**) when the syringe piston is retracted. **Off** otherwise.
INK_OLS	Limit switch that closes (**on**) when ink valve is open. **Off** otherwise.
INK_CLS	Limit switch that closes (**on**) when ink valve is closed. **Off** otherwise.

SYR_OUT	Syringe head assembly cylinder out control, **on** to move assembly out. **Off** has no effect.
SYR_BACK	Syringe head assembly cylinder back control, **on** to move assembly back. **Off** has no effect.
PIST_EXT	Syringe piston cylinder extend control, **on** to extend piston and inject ink into marker. **Off** has no effect.
PIST_RET	Syringe piston cylinder retract control, **on** to retract piston. **Off** has no effect.
INK_VLV	Ink valve control. **On** to open valve and allow ink to be sucked into syringe. **Off** will close the valve. The control must be **on** while the syringe piston is being retracted. The control must be **off** while the syringe piston is being extended.

Assume the following internal tags/variables/symbols:

Tag/Var./Symbol	Description
I_RUN	Run/stop/pause control for the ink injection station; **on** to run machine, **off** to stop/pause. Controlled by another part of the ladder logic in the PLC. **You must not have any output coil referring to I_RUN.**
I_RESET	Reset control, **on** to restore station to initial state. Controlled by another part of the ladder logic in the PLC. **You must not have any output coil referring to I_RESET.**

The addresses associated with the physical input and output tags/variables/symbols are:

Tag/Var./Symbol	ControlLogix	Micro800	MLogix	SLC-500
PROX4	Local:1:I.Data.3	_IO_EM_DI_03	I:0/3	I:1/3
SYR_OLS	Local:1:I.Data.4	_IO_EM_DI_04	I:0/4	I:1/4
SYR_BLS	Local:1:I.Data.5	_IO_EM_DI_05	I:0/5	I:1/5
PIST_ELS	Local:1:I.Data.6	_IO_EM_DI_06	I:0/6	I:1/6
PIST_RLS	Local:1:I.Data.7	_IO_EM_DI_07	I:0/7	I:1/7
INK_OLS	Local:1:I.Data.8	_IO_EM_DI_08	I:0/8	I:1/8
INK_CLS	Local:1:I.Data.9	_IO_EM_DI_09	I:0/9	I:1/9
SYR_OUT	Local:2:O.Data.0	_IO_EM_DO_00	O:0/0	O:2/0
SYR_BACK	Local:2:O.Data.1	_IO_EM_DO_01	O:0/1	O:2/1
PIST_EXT	Local:2:O.Data.2	_IO_EM_DO_02	O:0/2	O:2/2
PIST_RET	Local:2:O.Data.3	_IO_EM_DO_03	O:0/3	O:2/3
INK_VLV	Local:2:O.Data.4	_IO_EM_DO_04	O:0/4	O:2/4

The addresses and/or data types associated with the internal tags/variables/symbols are:

Tag/Var./Synbol	CLogix Data Type	Micro800 Data Type	MLogix/SLC Address
I_RUN	BOOL	BOOL	B3/101
I_RESET	BOOL	BOOL	B3/102

P6-2. Using the function chart approach, implement the program for the following machine that unloads a part from a weld die.

Figure P6.2 shows the layout of a machine that removes a part from a weld die. The part is being removed from a rotary table. The part is placed onto a holder on the table, the table is indexed (turned one position), a piece is placed on the part, the table is indexed, the part is welded, the table is indexed, and then the part is removed. You are only programming the part of the operation that removes the welded part and places it in a bin. The other aspects of the operation are controlled by another part of the PLC program. Also, assume that when commanded to remove the part, the part is present. Your solution will not be concerned with how to handle the case where no part is present on the die.

Upon initial startup, assume a part is present at the "unload" position of the rotary table. In summary, the arm is swiveled to its CCW position, the arm is moved down, the piece is clamped, the arm is moved up, swiveled to its CW position, and the part is released. The initial states of the main mechanical controls are: arm CW (over bin), arm up, and the clamp released. The following steps are executed:

Wait for the INDEX_DN internal coil to turn **on**. This is the signal that the rotary table has indexed one position and a new part is ready to be unloaded.

Rotate the arm counter-clockwise (CCW) to place it over the part. ARM_CCWLS senses when the arm is in position over the part.

Move the arm down (ARM_DNLS senses).

Clamp the part (PART_PROX senses). Note that PART_PROX must be on for 0.5 seconds to reliably indicate the clamp has closed and grips the part.

Move the arm up (ARM_UPLS senses).

Rotate the arm clockwise (CW) to locate it over the part bin. ARM_CWLS senses when the arm is in position over the bin.

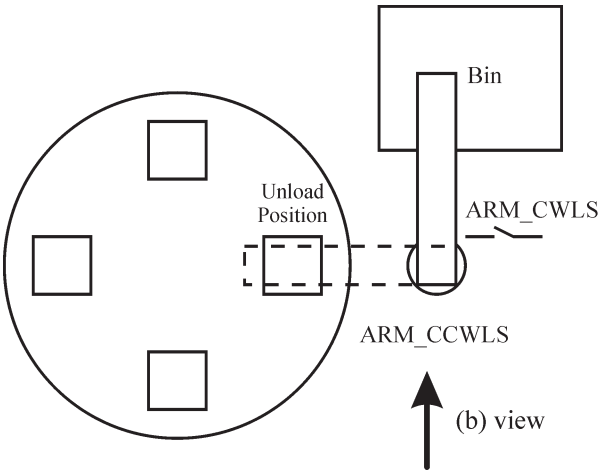
Release the part (CLAMP_OPLS and PART_PROX sense).

The operation then repeats.

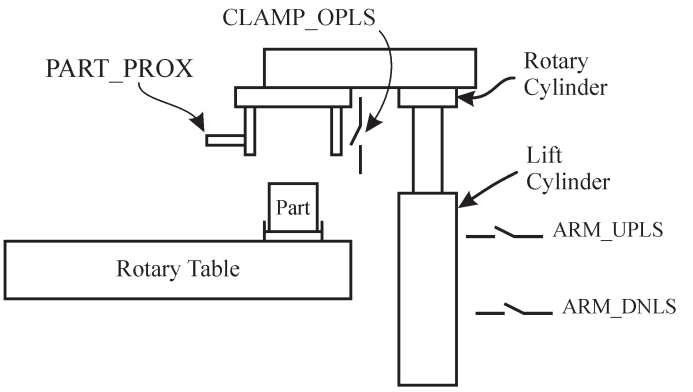
The INDEX_DN internal coil transitions **on** when the rotary table has indexed one position and a new part is ready to be removed. This internal coil is controlled by another part of the ladder logic and thus your part of the program must not have any coil that refers to INDEX_DN. You may assume this internal coil is **off** by the time the part is released from the clamp.

Rotary Cylinder: The rotary cylinder that controls the swiveling of the arm is a double-action pneumatic cylinder, controlled by the ARM_CW and ARM_CCW outputs. When ARM_CW is energized (turned **on**), the arm rotates to the bin position, where the end of the arm is over the bin. When ARM_CCW is energized (turned **on**), the arm rotates to the part position, where the end of the arm is over the position to pick up the welded part. When ARM_CW and ARM_CCW are both **off**, the arm stops at its present position. When ARM_CW and ARM_CCW are both **on**, the rotary cylinder operation is not consistent (it may stop or move in either direction slowly). The cylinder has two limit switches, ARM_CWLS, which is **on** when the arm is over the bin position and ARM_CCWLS, which is **on** when the arm is over the part position.

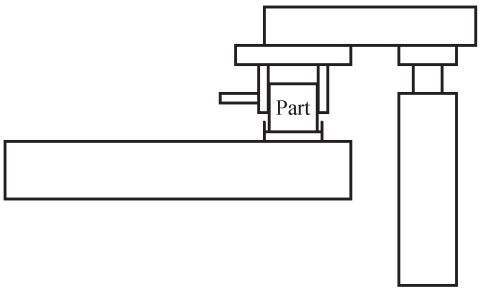
Lift Cylinder: The cylinder that controls the up/down of the arm is a double-action pneumatic cylinder, controlled by the ARM_UP and ARM_DN outputs. When ARM_DN is energized (turned **on**), the cylinder retracts and moves the arm mechanism down. When ARM_UP is energized (turned **on**), the cylinder extends, moving the arm up. When



(a)



(b)



(c)

Figure P6.2. Weld die unload station: *(a)* top view; *(b)* side view, arm up; *(c)* side view, arm down and part clamped.

ARM_DN and ARM_UP are both **off**, the arm stops at its present position. When ARM_DN and ARM_UP are both **on**, the cylinder operation is not consistent (it may stop or move in either direction slowly). The cylinder has two limit switches, ARM_DNLS, which is **on** when the arm is in the position to pick up the part and ARM_UPLS, which is **on** when the cylinder is extended (fully up).

The **part clamp** is controlled by a single-action cylinder controlled by the CLAMP_CLOS output. The CLAMP_CLOS physical output must remain **on** to grip and hold the part. This output must be **off** to release the part. The CLAMP_OPLS input is **on** when the clamp is open, not holding the part and **off** otherwise.

The PART_PROX **capacitive sensor** senses the presence/absence of a part in the clamp. However, since the sense distance is a few millimeters, PART_PROX must be **on** for 0.5 seconds to reliably indicate the clamp has closed and grips the part. Also, PART_PROX is not a reliable indicator that the part is completely released and so it and the CLAMP_OPLS sensor must be used together to sense that the part has been released by the clamp and has fallen into the bin. Of course, PART_PROX remains **on** when the clamp is holding the part.

There is no start/stop switch for your part of the ladder. Instead, there is an internal coil, W_RUN that is **on** when the operation is to happen. When W_RUN is **off**, the operation should pause at its current step. **Do not advance** to the next step when paused. When paused, all outputs should be **off except** for the following exception:

1. The clamp control, CLAMP_CLOS must not be turned **off** (if it is **on**).

When W_RUN turns back **on** while the operation of the station is paused, the station should resume its suspended step. Note that it is not necessary to retain the timer accumulator when paused.

The start and stop switches are absent from your part of the ladder because they control the overall operation of the rotary table. In order to reliably start/stop/pause the operation of the machine, the rotary table and other stations around the perimeter must be controlled. These parts of the operation are outside the scope of this problem. The other part of the logic controls W_RUN. **You must not have any output coil referring to W_RUN.**

A separate reset internal coil, W_RESET, is provided which restores the rotary table machine to its initial state. The reset operation should cause the arm to be fully raised, the arm rotated so that it is over the bin, and the clamp released. The reset operation is not complete until the arm is up and over the bin (clockwise) and the clamp is released. There is some mechanical interference between the clamp and the part holder on the wheel. So, the arm should not be rotated nor the clamp released until the arm is in its fully up position. Assume that an operator is responsible for actually removing any parts from the station after a reset. W_RESET should be ignored if W_RUN is **on**. The W_RUN internal coil used to restart the machine should be ignored when the reset operation is in progress. **You must not have any output coil referring to W_RESET.**

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 seconds.

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

Tag/Var./Symbol	Description
PART_PROX	Part-sensing capacitive proximity switch, **on** when part in clamp, **off** otherwise. However, because of the sensing distance, there must be a delay before reliably detecting the part presence. Description above.
CLAMP_OPLS	Limit switch that closes (**on**) when part clamp is fully open and therefore cannot grip the part. **Off** otherwise.
ARM_CWLS	Limit switch that closes (**on**) when the arm is in the clockwise position therefore over the bin where the part is to be released. **Off** otherwise.
ARM_CCWLS	Limit switch that closes (**on**) when the arm is in the counter-clockwise position therefore in position to pick up the part. **Off** otherwise.
ARM_DNLS	Limit switch that closes (**on**) when the arm mechanism is in its down position. **Off** otherwise.
ARM_UPLS	Limit switch that closes (**on**) when the arm mechanism is in its fully up position. **Off** otherwise.
CLAMP_CLOS	Part clamp control. **On** to close the clamp and grip the part. **Off** will open the clamp, dropping the part from the clamp.
ARM_CW	Arm rotary cylinder clockwise control, **on** to move arm mechanism clockwise to bin position. **Off** has no effect.
ARM_CCW	Arm rotary cylinder counter-clockwise control, **on** to move arm mechanism counter-clockwise to part position. **Off** has no effect.
ARM_DN	Arm linear cylinder down control, **on** to move arm mechanism down. **Off** has no effect.
ARM_UP	Arm linear cylinder up control, **on** to move arm mechanism up. **Off** has no effect.

Assume the following internal tags/variables/symbols:

Variable	Description
W_RUN	Run/stop/pause control for the unload station (also for rotary table); **on** to run machine, **off** to stop/pause. Controlled by another part of the ladder logic in the PLC. **You must not have any output coil referring to W_RUN.**
W_RESET	Reset control, **on** to restore unload station (and rotary table) to initial state. Controlled by another part of the ladder logic in the PLC. **You must not have any output coil referring to W_RESET.**
INDEX_DN	Rotary index done. When **on**, table has indexed one position and a new part is ready to be removed. 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 input and output tags/variables/symbols are:

Tag/Var./Symbol	ControlLogix	Micro800	MLogix	SLC-500
PART_PROX	Local:1:I.Data.3	_IO_EM_DI_03	I:0/3	I:1/3
CLAMP_OPLS	Local:1:I.Data.4	_IO_EM_DI_04	I:0/4	I:1/4
ARM_CWLS	Local:1:I.Data.5	_IO_EM_DI_05	I:0/5	I:1/5
ARM_CCWLS	Local:1:I.Data.6	_IO_EM_DI_06	I:0/6	I:1/6
ARM_DNLS	Local:1:I.Data.7	_IO_EM_DI_07	I:0/7	I:1/7
ARM_UPLS	Local:1:I.Data.8	_IO_EM_DI_08	I:0/8	I:1/8
CLAMP_CLOS	Local:2:O.Data.0	_IO_EM_DO_00	O:0/0	O:2/0
ARM_CW	Local:2:O.Data.1	_IO_EM_DO_01	O:0/1	O:2/1
ARM_CCW	Local:2:O.Data.2	_IO_EM_DO_02	O:0/2	O:2/2
ARM_DN	Local:2:O.Data.3	_IO_EM_DO_03	O:0/3	O:2/3
ARM_UP	Local:2:O.Data.4	_IO_EM_DO_04	O:0/4	O:2/4

The addresses and/or data types associated with the internal tags/variables/symbols are:

Tag/Var./Symbol	CLogix Data Type	Micro800 Data Type	MLogix/SLC Address
W_RUN	BOOL	BOOL	B3/101
W_RESET	BOOL	BOOL	B3/102
INDEX_DN	BOOL	BOOL	B3/151

P6-3. Using the function chart approach, implement the program for the following station that places a barcode label on a screwdriver.

Figure P6.3 shows the layout of a station that places a barcode label on a screwdriver. This station is only one in a series of stations along this conveyor. The stations upstream of this station manufacture the screwdrivers and each screwdriver comes to this station in a carrier. You are implementing ladder logic for this station only. You have no control over the conveyor so assume it is always moving. As a carrier is captured, the conveyor slides underneath. This particular line is asynchronous, that is, each station processes screwdrivers at its own speed and does not directly coordinate its operation with any other station.

Upon initial startup, assume that there are no carriers (with screwdrivers) in the station. The initial states of the main mechanical controls are: catch mechanism retracted, rotate head up and label ram retracted. The following steps are executed:

Rotate the label takeup reel until the label is in position (sensed by PX362). The catch cylinder must also be extended during the rotation.

Wait for next carrier (continue to extend the catch mechanism). When a carrier has been captured, PX361 senses.

Lower the rotate head into position (CY368_RLS senses). As the head is lowered, it engages the top of the screwdriver.

Extend the label cylinder to move the labels (on a backing tape) into contact with the screwdriver shaft (CY364_ELS senses).

Activate the motor on the rotate head to rotate the screwdriver so that the label is wrapped around the screwdriver shaft. The action also pulls the labels such that PX362 is turned **off**. Pulses from an encoder attached to the rotate motor are used to gauge the amount of rotation, which is 225 encoder pulses.