CHAPTER 7 SOLVED PROBLEMS

General instructions for problems:

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

ControlLogix, **or** 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 situation 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 appropriately.

Assume the analog input channel values corresponding to the lowest and highest sensor values for the various processors are as specified before text P7-1. Also assume the analog output channel values corresponding to the lowest and highest values of the output control device for the various processors are as specified before text P7-1.

Your solution should include the following:

1. Function chart (if appropriate)
2. Specify the PLC processor used.
3. Ladder logic diagram (with comments). For consistency among the different PLCs, use only tags/symbols/tags in the ladder logic. Use instructions and function blocks consistent with the PLC processor.
4. Table listing additional internal memory (symbols/tags) used and a brief description of their use. For the ControlLogix, CompactLogix, and Micro800 processors, list the internal tags/variables and the data type. For the MicroLogix/SLC-500 processors, list the internal variables/symbols and the associated memory address.

SP7-1. A pressure sensor measures vessel pressures between 15 and 100 pounds per square inch (psi). The output of the pressure transmitter is a 4 to 20 mA signal that is connected to an analog input module channel whose channel value corresponds to the lowest and highest pressure values as specified before text P7-1. The tag/variable/symbol for the channel value is PT101_Raw. The tag for the conversion result is PT101_PSI, a real.:

Program ladder logic to do the conversion for

(a) ControlLogix/CompactLogix with 1756/5069/5094-series input module
(b) CompactLogix with 1769-series input module
(c) Micro800 with plug-in analog input module

(d) Micro800 with expansion analog input module
(e) MicroLogix with 1762-series input module

SP7-2. Day Tank Level Control. Implement the following calculation and comparison operations for a day tank level control.

A day tank has about one 8-hour shift worth of material to feed into a process. A day tank is located close to the process and is fed from a much larger bulk tank, located some distance from the process to be conveniently loaded from a tanker truck or railcar. The day tank allows much finer control of the material feed into the process and saves energy since the bulk tank pump runs only occasionally. The day tank is only fed from the bulk tank when the day tank level is low (for example, 10%). When the day tank has been filled from the bulk tank to a high level (for example, 90%), the bulk tank pump is shut off.

The PLC implements a simple level control for the T-304 day tank in Figure SP7.2. The PLC controls the P-303 pump and XV303 block valve to keep the liquid level in T-304 between two limits, T304_HI_LVL and T304_LO_LVL:

The valve is opened and the pump is turned **on** when

Measured Level < T304_LO_LVL

The valve is closed and the pump is turned **off** when

Measured Level > T304_HI_LVL

The T304_HI_LVL and T304_LO_LVL values, in inches, are already stored in internal memory locations. The level measurement transmitter, LT304, is calibrated to measure from 0 to 45 inches. LT304 is connected to an analog input module having an ADC whose output value (LT304_MEAS) corresponds to the lowest and highest levels as specified in the general instructions. The LT304_MEAS value should be converted into a level in inches (a real number) and placed in T304_INCH.

Also, in order to protect against illegal values entered by the operator, make sure the T304_HI_LVL is no higher than 40 inches and that the T304_LO_LVL is no lower than 10 inches. Also, T304_HI_LVL should be at least 10 inches higher than T304_LO_LVL.

The T304_ENABLE internal coil enables the day tank control. When T304_ENABLE is **on**, the T-304 level is controlled as described above. The valve is closed and the pump does not run when T304_ENABLE is **off**.

Assume the following physical inputs, physical outputs and internal tags/symbols:

Tag/Var./Symbol	Description
P303_RUN	P-303 pump control, **on** runs pump.
XV303_SOL	XV303 valve control, **on** opens valve, **off** closes valve.
LT304_MEAS	Raw T-304 level measurement, represents 0 to 45 inches.
T304_ENABLE	Enable day tank control; **on** allows day tank control to operate. When **off**, the valve is closed and the pump is off.
T304_LO_LVL	Minimum tank level, in inches (a REAL).

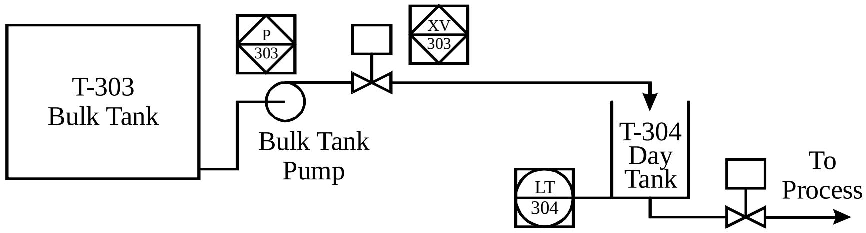


Figure SP7.2. Day tank system.

T304_HI_LVL	Maximum tank level, in inches (a REAL).
T304_INCH	Current tank level, in inches (a REAL).

The addresses associated with the physical inputs and outputs are:

Tag/Var./Symbol	ControlLogix	Micro800	MLogix	SLC-500
P303_RUN	Local:2:O.Data.0	_IO_EM_DO_00	O:0/0	O:2/0
XV303_SOL	Local:2:O.Data.1	_IO_EM_DO_01	O:0/1	O:2/1
LT304_MEAS	Local:3:I.Ch0Data	_IO_P1_AI_00	I:1.0	I:3.0

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

Tag/Var./Symbol	CLogix Data Type	Micro800 Data Type	MLogix/SLC Address
T304_ENABLE	BOOL	BOOL	B3/20
T304_LO_LVL	REAL	REAL	F12:0
T304_HI_LVL	REAL	REAL	F12:1
T304_INCH	REAL	REAL	F12:2

SP7-3. Day Tank Alarms. Add low and high alarms to the day tank of problem SP7-2 described as follows:

The alarm indicators consist of high and low level alarm lights (T304_HIALM and T304_LOALM) and an alarm horn, ALM_HORN. There are two low alarm levels and two high alarm levels:

Low-low alarm	5 inches
Low alarm	T304_LO_LVL – 3
High alarm	T304_HI_LVL + 3
High-high alarm	44 inches

The alarm light actions depend on the relationship of the current level with the alarm values:

Level ≤ Low-low alarm	Low alarm flashes at 2 Hz
Low-low < Level ≤ Low	Low alarm steady on
Low < Level < High	No alarms
High ≤ Level < High-high	High alarm steady on
High-high ≤ Level	High alarm flashes at 2 Hz

The horn is handled in the following manner. The horn is turned on (steady) whenever the level ≤ low-low alarm or whenever the level ≥ high-high alarm. The horn remains on until the operator has acknowledged the alarm by pressing the HORN_ACK button. When the HORN_ACK button is pressed, the horn should be silenced and remain silenced when the button is released. If a level ≤ low-low alarm turns on the horn, the horn should remain **on** until acknowledged by the operator, even if the level rises above the low-low alarm value. If a level ≥ high-high alarm turns on the horn, the horn should remain **on** until acknowledged by the operator, even if the level falls below the high-high alarm value.

Assume the following additional physical inputs and outputs.

Tag/Var./Symbol	Description
HORN_ACK	Horn acknowledge push button switch; **on** when pushed, **off** when released. Operation described above.
T304_HIALM	High level alarm indication, high-high when flashing.
T304_LOALM	Low level alarm indication, low-low when flashing.
ALM_HORN	Alarm horn, **on** to sound horn.

The addresses associated with the physical inputs and outputs are:

Tag/Var./Symbol	ControlLogix	Micro800	MLogix	SLC-500
HORN_ACK	Local:1:I.Data.0	_IO_EM_DI_00	I:0/0	I:1/0
T304_HIALM	Local:2:O.Data.2	_IO_EM_DO_02	O:0/2	O:2/2
T304_LOALM	Local:2:O.Data.3	_IO_EM_DO_03	O:0/3	O:2/3
ALM_HORN	Local:2:O.Data.4	_IO_EM_DO_04	O:0/4	O:2/4

SP7-4. Plant Waterer Improvements. Modify the plant waterer of problem SP5-6 so that the 8-hour interval can be changed as desired by the user and the number of hours to the next watering check is calculated and displayed.

The desired watering interval (WAT_INTERVAL) is an integer, specified by an HMI package. Make sure WAT_INTERVAL is in the range of 0 – 15, inclusive. Do not write to this location, except to make sure it is within range.

Calculate REMAIN_TIME, the number of hours remaining to the next watering interval. Note that REMAIN_TIME is an integer. This number is to be displayed by an HMI package. The number of remaining hours is also displayed as a binary pattern on four discrete outputs. However you do the calculation, the other bits in the discrete output word must not be changed by the result of your calculation.

Assume the following additional physical output and internal memory assignments:

Tag/Var./Symbol	Description
WAT_INTERVAL	Desired watering interval, in hours. Maximum of 15. DO NOT write to this location.
REMAIN_TIME	Remaining hours to the next watering interval.
REMAIN0	Least significant bit (bit 0) of 4-bit display of hours remaining.
REMAIN1	Bit 1
REMAIN2	Bit 2
REMAIN3	Most significant bit (bit 3) of 4-bit display of hours remaining.

The addresses associated with the physical inputs and outputs are:

Tag/Var./Symbol	ControlLogix	Micro800	MLogix	SLC-500

REMAIN0	Local:2:O.Data.2	_IO_EM_DO_02	O:0/2	O:2/2
REMAIN1	Local:2:O.Data.3	_IO_EM_DO_03	O:0/3	O:2/3
REMAIN2	Local:2:O.Data.4	_IO_EM_DO_04	O:0/4	O:2/4
REMAIN3	Local:2:O.Data.5	_IO_EM_DO_05	O:0/5	O:2/5

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

Tag/Var./Symbol	CLogix Data Type	Micro800 Data Type	MLogix/SLC Address
WAT_INTERVAL	INT	DINT	N7:0
REMAIN_TIME	INT	DINT	N7:1

SP7-5. Flow Meter Accumulator. Use ladder logic to implement an accumulator that converts the reading from a flow meter (gal/min) to the volume of material (gallons) that passes through the flow meter. *Do not implement a function chart*.

The accumulator converts flow into volume by implementing a simple integrator that integrates the flow rate converting it into a volume. In order to do the integrator operation in the PLC, the flow rate is sampled at 1-second intervals and the volume is updated accordingly. Given samples of the flow, the general equation for the volume at the ***i***-th time interval is

$$\text{Volume}_i = T\sum_{k=0}^{i}\text{Flow}_k \qquad \text{(SP7-5.1)}$$

where *T* is the sample period (1 second in this case). In words, the volume can be calculated by summing all previous flow samples and multiplying this sum by *T*. However, it is not necessary to keep all previous values of the flow, because the volume can be updated with each new sample of the flow rate by the formula

$$\text{Volume}_i = \text{Volume}_{i-1} + T \times \text{Flow}_i \qquad \text{(SP7-5.2)}$$

In words, the volume is updated by adding to it the product of the current flow rate and the sample period.

What your code needs to do:

(1) Implement equation (SP7-5.2) every 1 second. In order to make the units consistent, *T* = 1/60 = 0.01667 (since volume is gallons and the flow rate units are gallons/minute). Just update the current value of VOLUME. Do not save the previous value of VOLUME.

(2) When the RESET_ACCUM internal coil is ON, reset the accumulator, that is, set the volume to zero.

The FTXXX_VAL value has already been converted from the raw analog input channel value into a real value with units of gallons/minute.

All time intervals must be accurate to within 0.01 second.

Assume the following internal memory assignments.

Tag/Var./Symbol	Description
FTXXX_VAL	Current flow rate, in gallons/minute (a REAL).

VOLUME	Volume of material that has passed through flow meter since the accumulator has been reset, in gallons (a REAL).
RESET_ACCUM	**On** to reset accumulator (set VOLUME to zero).

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

Tag/Var./Symbol	CLogix Data Type	Micro800 Data Type	MLogix/SLC Address
FTXXX_VAL	REAL	REAL	F8:6
VOLUME	REAL	REAL	F8:7
RESET_ACCUM	BOOL	BOOL	B3/201

P7-6. Weigh Scale Station Control. Using the function chart approach, implement the program for the following weigh scale station that weighs pallets and prints the weight.

Figure P7.6 shows the general layout of a station that weighs and prints the weight of a pallet that comes down the conveyor. The infeed and outfeed conveyors are on at all times and are not controlled by another part of the ladder logic. The weigh scale has its own small conveyor belt that can be stopped to weigh the pallet. The operation is described in the following manner:

The weigh scale belt runs continuously until the weigh scale senses at least 5 pounds. The presence of the pallet is sensed by the weigh scale and not by any limit switches or photoelectric “eyes."

When the weigh scale detects a weight of greater than 5 pounds, then the conveyor belt is stopped.

After a pause of 4 seconds, to allow the weight to stabilize, the spray printer is activated to spray the weight on the pallet. In order to keep the problem simple, you will not need to know how to send the weight to the printer. All your ladder needs to do is to make sure the weight, in pounds, is in the proper memory location and turn **on** the DO_PRINT internal coil. Another internal coil, called PRINT_DONE, is set by another part of the ladder to signal that the weight has been printed.

After the printer has sprayed the pallet, then the weigh scale conveyor is activated until the weight has been less than 5 pounds for 2 seconds to move the pallet out of the weigh station and onto the outfeed conveyor.

After the pallet has been moved off, the weigh scale station repeats its operation, running the conveyor until the next pallet is processed. Assume the pallets are spaced far enough apart so that the next pallet arrives after the pallet has left the station.

Upon initial startup, assume there are no pallets in the station, and the conveyor runs until a pallet is detected as detailed above. When the stop push button is pressed (turned **off**) the operation should pause and the conveyor must be turned **off**. Pressing the start switch while the operation of the station is paused causes the station to resume its suspended operation.

There is a reset push button that when pressed, resets any internal steps so that when the start switch is pressed, no pallets are assumed to be in the station. The reset switch should be ignored if the station is running. Assume the pallets are removed before RESET_PB is pressed.

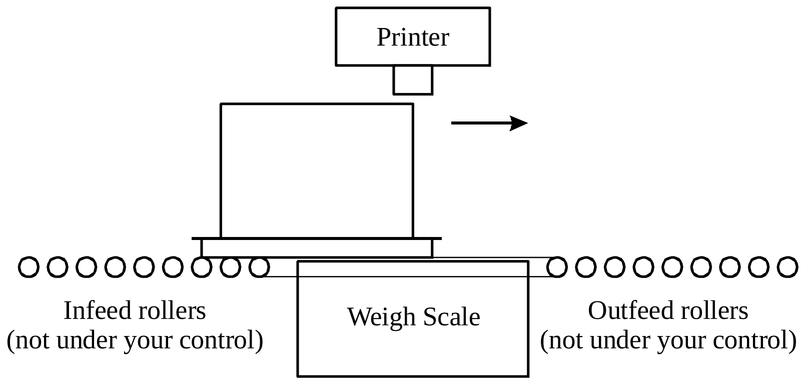


Figure SP7.6. Weigh scale station.

The weight transducer is calibrated to measure from 0.0 to 900.0 pounds and is connected to an analog input module having an ADC whose output value (WT132_MEAS) corresponds to the lowest and highest weights as specified in the general instructions. The WT132_MEAS value should be converted into a weight, in pounds (a real number) and placed in the WT132_VAL tag/symbol.

Assume the following physical inputs, physical outputs, and internal tags/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.
CONV_MOTOR	Conveyor motor control, **on** to move conveyor.
WT132_MEAS	Raw pallet weight measurement, represents 0 to 900 lbs.
WT132_VAL	Current weigh scale measurement, in pounds (a REAL).
PRINT_DONE	Indicates that printer has finished spraying weight on box. **On** when printing done. Controlled by another part of the ladder.
DO_PRINT	**On** to activate the spray printer to spray weight on box. Controlled by this part of the ladder.

The addresses associated with the physical inputs and outputs are:

Tag/Var./Symbol	ControlLogix	Micro800	MLogix	SLC-500
START_PB	Local:1:I.Data.0	_IO_EM_DI_00	I:0/0	I:1/0
STOP_PB	Local:1:I.Data.1	_IO_EM_DI_01	I:0/1	I:1/1
RESET_PB	Local:1:I.Data.2	_IO_EM_DI_02	I:0/2	I:1/2
CONV_MOTOR	Local:2:O.Data.0	_IO_EM_DO_00	O:0/0	O:2/0
WT132_MEAS	Local:3:I.Ch0Data	_IO_P1_AI_00	I:1.0	I:3.0

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

Tag/Var./Symbol	CLogix Data Type	Micro800 Data Type	MLogix/SLC Address
WT132_VAL	REAL	REAL	F12:0
PRINT_DONE	BOOL	BOOL	B3/101
DO_PRINT	BOOL	BOOL	B3/102

SP7-7. Width Check Station Control. Using the function chart approach, implement the program for the following station that checks the width of parts.

Figure SP7.7 shows the general layout of a station that checks the width of parts as they come down the conveyor. The section of conveyor at the station is on at all times the station is running. Not every part is checked. Only every tenth part is checked. In summary, when a part is to be checked, a clamp is activated to hold the part and check the width with an LVDT. Delays must be incorporated into the operation to allow the measurement to settle and the part to clear when the part is unclamped. The operation is described in the following manner:

The presence of a part is sensed by the PART_PRESENT reflective infrared sensor.

After 9 parts have passed, the tenth part is checked (when it is present), by activating the clamp cylinder. When the part is clamped, the conveyor is stopped and a gate is also activated to prevent more parts from interfering with the measurement.

After a pause of 2 seconds, to allow the measurement to stabilize, the width is measured. The width measurement is described below.

After the measurement is completed, the clamp is released, the gate is deactivated, the conveyor is restarted, and then there is a 2 second delay to allow the conveyor to restart and move the part out of the station. The NEW_MEAS internal coil must be **on** in this step, signaling to another part of the ladder that there is a new width measurement.

After the 2 second wait, the station restarts the operation, waiting for the tenth part.

The width of the part is measured using a linear variable differential transformer (LVDT), calibrated to measure from 0.0 to 70.0 mm. The LVDT is connected to an analog input module having an ADC whose output value (WIDTH_MEAS) corresponds to the lowest and highest widths as specified in the general instructions. The WIDTH_MEAS value should be converted into units of mm and stored in WIDTH_VAL (a real number). The width is only valid at the end of the 2-second period that allows it to be stabilized. DO NOT continuously calculate WIDTH_VAL.

When the start push button is pressed (turned **on**) for the first time only, the station assumes that no parts have passed the infrared detector. When the stop switch is pressed (turned **off**) the operation should pause and only the conveyor must be turned **off**. When paused, retain the delay that allows the part to move out and do not advance to the next step. Pressing the start switch while the operation of the station is paused causes the station to resume its suspended operation.

There is a RESET_PB switch that when **on**, restarts the operation and retracts the two cylinders. When RESET_PB is **on**, the internal state is set so that the ladder logic program that no parts have passed the infrared detector. The RESET_PB switch must have no effect unless the operation is already paused.

All delays must be accurate to within 0.1 seconds.

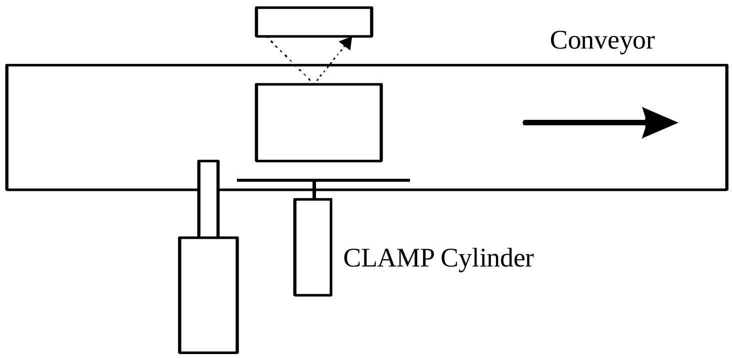


Figure SP7.7. Width measurement station.

Assume the following physical input, physical output, and internal memory assignments:

Tag/Var./Symbol	Description
START_PB	Start push-button switch, **on** (closed) to start.
STOP_PB	Stop push-button switch, **off** (open) to stop.
RESET_PB	Reset push button switch, **on** (closed) to reset.
PART_PRESENT	Proximity sensor, **on** when part present.
CONV_MTR	Conveyor motor, **on** to run conveyor at station.
CLAMP	Clamp cylinder control, **on** to clamp part.
GATE	Gate cylinder control, **on** to activate gate that prevents more parts from interfering with the measurement.
NEW_MEAS	Internal coil, set **on** for 2 seconds by your ladder, to indicate that a new width has been measured.
WIDTH_MEAS	Raw width measurement, represents 0 to 70 mm.
WIDTH_VAL	Width of current (or last) part, in mm (a REAL).

The addresses associated with the physical inputs and outputs are:

Tag/Var./Symbol	ControlLogix	Micro800	MLogix	SLC-500
START_PB	Local:1:I.Data.0	_IO_EM_DI_00	I:0/0	I:1/0
STOP_PB	Local:1:I.Data.1	_IO_EM_DI_01	I:0/1	I:1/1
RESET_PB	Local:1:I.Data.2	_IO_EM_DI_02	I:0/2	I:1/2
PART_PRESENT	Local:1:I.Data.3	_IO_EM_DI_03	I:0/3	I:1/3
CONV_MTR	Local:2:O.Data.0	_IO_EM_DO_00	O:0/0	O:2/0
CLAMP	Local:2:O.Data.1	_IO_EM_DO_01	O:0/1	O:2/1
GATE	Local:2:O.Data.2	_IO_EM_DO_02	O:0/2	O:2/2
WIDTH_MEAS	Local:3:I.Ch0Data	_IO_P1_AI_00	I:1.0	I:3.0

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

Tag/Var./Symbol	CLogix Data Type	Micro800 Data Type	MLogix/SLC Address
NEW_MEAS	BOOL	BOOL	B3/140
WIDTH_VAL	REAL	REAL	F12:0

SP7-8. Stamping Station Control. Using the function chart approach, implement the program for the following station that stamps a piece of metal.

Figure SP7.8 shows the general layout of a station that stamps an impression into a metal piece. Not every piece is stamped. Only those having a particular bar code are stamped. The piece is pushed off the conveyor into the stamping station, where it is stamped before being returned to the conveyor. The required stamping pressure is not constant and is stored in an internal memory location. The actual stamp pressure is measured by a pressure transducer connected to an analog input channel.

The overall operation of the station is first described, followed by the details. The sequence of operation for the station follows these steps:

Metal piece coming down conveyor has its bar code read by the bar code reader. If the bar code indicates the piece is to be stamped, then the following steps are done. Otherwise, the station just waits for the next piece.

When PROX turns **on**, the piece is in position to be pushed into the stamper.

Extend PCYL1 to push piece to stamping position. LS2 turns **on** when in position. When in position, the piece is also against the end of PCYL2

The hydraulic ram, HCYL3 is moved down until the stamping pressure reaches the desired value.

Move HCYL3 up to fully up position (allow 3 seconds)

Simultaneously, PCYL2 is extended and PCYL1 is retracted to push piece onto conveyor (LS1 senses)

Retract PCYL2 (allow 2 seconds). It requires 4 seconds to fully retract PCYL2, but 2 seconds is sufficient time to allow it to clear the conveyor and allow a conveyor restart.

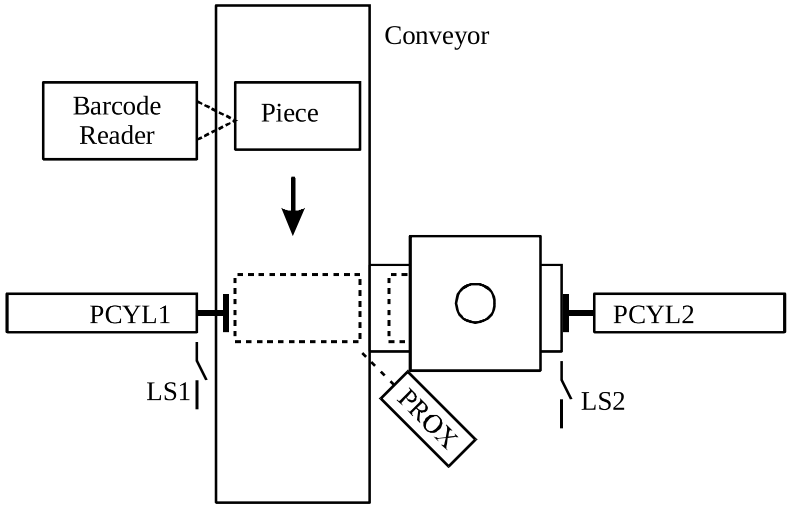
Move piece out of station

The operation then repeats.

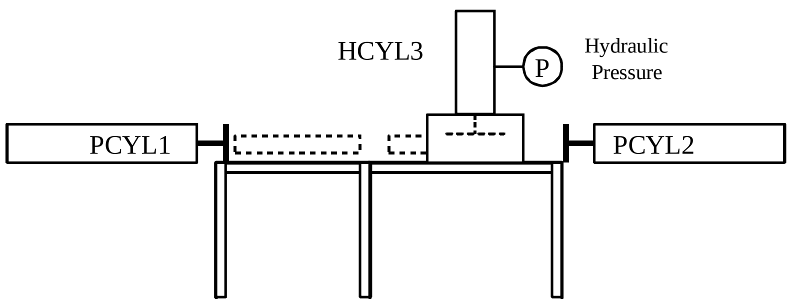
The conveyor must be stopped while the part is being pushed off or on the conveyor and while the metal piece is being stamped. Assume pieces are far enough apart on the conveyor so that each piece can be handled individually.

A bar code reader is used to scan the bar code on the metal piece. All you know is that the bar code (a part number) is placed in an internal memory location, called BARCODE, and that an internal coil, NEW_CODE, is **on** when a new bar code has been read and placed into BARCODE. Assume that NEW_CODE is **on** only for the scan when a new barcode is read (it is the output of a rung driven by a one-shot instruction). DES_BARCODE contains the bar code of the parts that must be stamped, and is set by a host PLC or computer.

PCYL1 is pneumatic ram powered by a double action cylinder. When the PCYL1_EXTEND output is **on**, the PCYL1 ram extends. When the PCYL1_RETRACT output is **on**, the PCYL1 ram retracts. When neither PCYL1 output or both outputs are **on**, the PCYL1 ram stops moving and remains at its present position. However,



(a)



(b)

Figure SP7.8. Stamping station: *(a)* top view; *(b)* side view.

PCYL1_EXTEND output must remain **on** to hold the piece in position while it is being stamped (even when paused).

PCYL2 is a pneumatic ram powered by a single action cylinder with a spring return. When the PCYL2 output is **on**, the ram extends until it reaches the fully extended position. When the power is removed (PCYL2 is **off**), the ram retracts until it is in its fully retracted position. There is no limit switch that indicates that the ram powered by PCYL2 is fully retracted or extended. When turning off the PCYL2 output, assume that at least 4 seconds are required to go from the fully extended position to the fully retracted position.

The ram that moves the stamp down and up, HCYL3, is a hydraulic ram powered by a double action cylinder (like PCYL1). When the STAMP_DOWN output is **on**, the stamping press moves down. When the STAMP_UP output is **on**, the press moves up. When neither stamp ram output or both outputs are **on**, the cylinder stops moving vertically and remains at its present position. There is no limit switch indicating when HCYL3 is fully up. When moving the press up, assume that at least 3 seconds are required to go from the fully down (extended) position to the fully up (retracted) position.

PROX is used to sense when the metal piece is in position to be pushed into the stamping press. When it is **on** the piece is in position.

The DES_PRESS internal memory location indicates the desired hydraulic stamping pressure in pounds per square inch (psi). Assume this value has been placed in the PLC memory from a supervisory PLC or computer. The hydraulic pressure is measured by a pressure transducer is calibrated to measure from 2000 to 5000 psi and is connected to an analog input module having an ADC whose output value (PT214_MEAS) corresponds to the lowest and highest pressures as specified in the general instructions. The PT214_MEAS value should be converted into a pressure, in psi (a real number) and placed in the PT214_VAL tag/symbol.

Use the pressure measurement, converted to the proper units to determine when to stop the hydraulic stamping ram. The ram pressure increases as the stamping ram extends (stamp lowers) and contacts the metal plate. (Hint: remember, there may be some inaccuracy in the measurement, so using equality to stop the ram is not wise.)

When the start switch is pressed (turned **on**) for the first time only, the station assumes there is no piece in the station and waits for the first piece to arrive. When the stop switch is pressed (turned **off**) the operation should pause and all outputs (except PCYL1 when stamping taking place) must be turned **off**. The operation cannot be paused when the hydraulic stamping press is being moved down (to make sure impression is correct). If the stop switch is pressed while the press is being lowered, the operation should continue to the next step and and pause in that step. Pressing the start switch while the operation of the station is paused causes the station to resume its suspended operation.

There is a RESET_PB switch that when **on**, restarts the operation. When RESET_PB is **on**, the internal step is set so that the ladder logic program waits for the next metal piece (but does not actually turn any outputs **on**). Also, retract all rams, including those that are double-acting when RESET_PB is momentarily pressed. All rams must be retracted before START_PB has any effect. START_PB must be pressed to actually restart the station. The RESET_PB switch must have no effect unless the operation is already paused and not pressing. 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.

All delays must be accurate to within 0.1 seconds.

Assume the following input, output, and internal memory assignments.

Tag/Var./Symbol	Description
START_PB	Start push button switch, **on** (closed) to start.
STOP_PB	Stop pushbutton switch, **off** (open) to stop.
RESET_PB	Reset push button switch, **on** (closed) when resetting.
LS1	Limit switch, **on** (closed) when PCYL1 is fully retracted and piece is back on conveyor.
LS2	Limit switch, **on** (closed) when metal piece is in position to be stamped (under stamping press).
PROX	Proximity switch, **on** when piece in position to be moved into stamper.

CONV_MTR	Conveyor motor, **on** to run conveyor.
PCYL1_EXTEND	PCYL1 extend ram control; **on** to extend ram, **off** stops ram movement (but does not retract ram).
PCYL1_RETRACT	PCYL1 retract ram control; **on** to retract ram, **off** stops ram.
PCYL2	PCYL2 ram control; **on** to extend ram, **off** retracts.
STAMP_DOWN	**On** to move stamp ram down, **off** stops movement (but does not retract ram).
STAMP_UP	**On** to move stamp ram up, **off** stops movement.
PT214_MEAS	Raw stamping ram pressure measurement, represents 2000 to 5000 psi.
NEW_CODE	**On** when a new barcode has been read and placed into BARCODE. This part of ladder DOES NOT control it.
PT214_VAL	Current stamping ram pressure, in psi (a REAL).
DES_PRESS	Desired stamping ram pressure, in psi (a REAL).
BARCODE	Bar code read from piece. Only valid when NEW_CODE is ON.
DES_BARCODE	Barcode of those parts that are to be stamped (integer). This part of ladder DOES NOT control it.

The addresses associated with the physical inputs and outputs are:

Tag/Var./Symbol	ControlLogix	Micro800	MLogix	SLC-500
START_PB	Local:1:I.Data.0	_IO_EM_DI_00	I:0/0	I:1/0
STOP_PB	Local:1:I.Data.1	_IO_EM_DI_01	I:0/1	I:1/1
RESET_PB	Local:1:I.Data.2	_IO_EM_DI_02	I:0/2	I:1/2
LS1	Local:1:I.Data.3	_IO_EM_DI_03	I:0/3	I:1/3
LS2	Local:1:I.Data.4	_IO_EM_DI_04	I:0/4	I:1/4
PROX	Local:1:I.Data.5	_IO_EM_DI_05	I:0/5	I:1/5
CONV_MTR	Local:2:O.Data.0	_IO_EM_DO_00	O:0/0	O:2/0
PCYL1_EXTEND	Local:2:O.Data.1	_IO_EM_DO_01	O:0/1	O:2/1
PCYL1_RETRACT	Local:2:O.Data.2	_IO_EM_DO_02	O:0/2	O:2/2
PCYL2	Local:2:O.Data.3	_IO_EM_DO_03	O:0/3	O:2/3
STAMP_DOWN	Local:2:O.Data.4	_IO_EM_DO_04	O:0/4	O:2/4
STAMP_UP	Local:2:O.Data.5	_IO_EM_DO_05	O:0/5	O:2/5
PT214_MEAS	Local:3:I.Ch0Data	_IO_P1_AI_00	I:1.0	I:3.0

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

Tag/Var./Symbol	CLogix Data Type	Micro800 Data Type	MLogix/SLC Address
NEW_CODE	BOOL	BOOL	B3/140
PT214_VAL	REAL	REAL	F8:40
DES_PRESS	REAL	REAL	F12:0
BARCODE	SINT	SINT	N7:45
DES_BARCODE	SINT	SINT	N13:0

SP7-9. Valve Leak Check Station Control. Using the function chart approach, implement the program for the following valve leak check station.

Figure SP7.9 shows the layout of a station that checks each assembled valve for leaks. The station pressurizes the valve and then checks that the pressure has not dropped too far within a certain time period.

Another part of the ladder controls the roller conveyor, so assume it is always moving.

Each valve is on a carrier, which ensures that the PROX sensor will turn off between valves, even if there is no spacing between the carriers.

There is no START_PB and STOP_PB because there is an overall start and stop for the line. Instead, there is a RUN internal coil that is **on** when this station is to be running. The station is running as long as valves are being produced.

Upon initial turning on of the RUN internal coil, assume there are no valves in the station. To pressure-check a valve the following happens:

A new valve is sensed (PROX turns **on**).

The HD_DOWN cylinder control is activated in order to move the measuring head down into position. Since this station must work with different valve sizes, the head is moved down until HD_HGT is less than or equal to VLV_HGT. Simultaneously, the LIFT_SOL is activated to raise the valve off the rollers. Assume LIFT_SOL is fast-acting and has finished moving the valve up long before the head is down in position. Do not construct a branch to do this operation.

The valve is pressurized to the desired pressure, DES_PRES. Turn on AIR_VLV to pressurize the valve under test.

Wait 30 seconds. If the valve has a leak, then the measured pressure at the end of this interval will be smaller than DES_PRES. The details about the pressure check appear below.

Activate the HD_UP cylinder control to move measuring head up. UP_LS is **on** when the head is fully up. Simultaneously, deactivate, the LIFT_SOL to let the valve fall back onto the conveyor. Assume LIFT_SOL is fast-acting and has finished moving the valve down on to the conveyor long before the head is fully up. Do not construct a branch to do this operation.

The operation then repeats.

The conveyor rollers are not controlled by this section of the ladder, so assume they are always **on**.

The proximity sensor is a reflective infrared sensor (like many of the proximity sensors in the conveyor lab exercise). PROX senses the leading edge of the valve when it is in the proper position. Assume that PROX remains **on** as the valve is being checked for leaks and as it is being lifted off the conveyor.

The LIFT_SOL is a fast-acting single action pneumatic cylinder. When the LIFT_SOL output is energized, the valve on its carrier is raised off the conveyor. The carrier remains up as long as LIFT_SOL is **on**. As soon as LIFT_SOL is turned **off**, the carrier (and valve) fall to the rollers and are conveyed out of the check station. There are limit switches, LU_LS and LD_LS that indicate when the LIFT_SOL is up or down, respectively. However, LIFT_SOL is fast-acting and you are to assume that the pressurizing head (which moves at the same time) is slower. Therefore, do not use LU_LS or LD_LS in your ladder. They

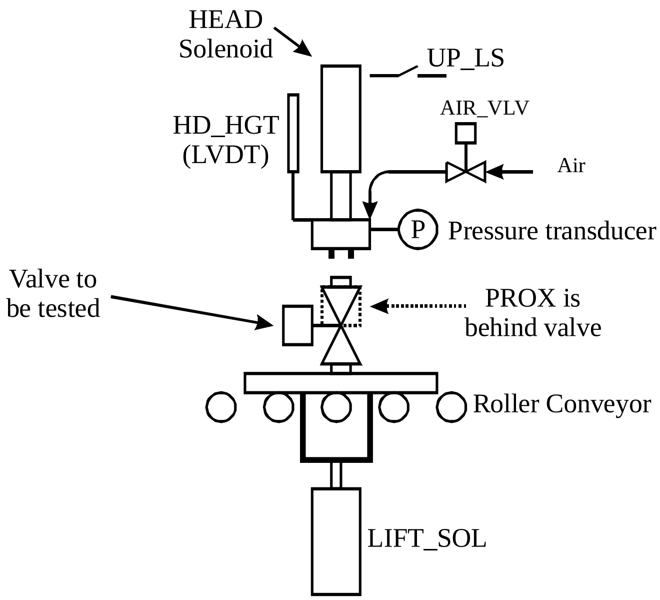


Figure SP7.9. Leak check station.

would be used for error checking to indicate that the cylinder did not move and is outside the scope of this problem.

The mechanism used to lower and raise the pressurizing head consists of a double-action linear pneumatic cylinder. When the HD_DOWN output is energized, the mechanism moves down and continues to move down as long as power is applied (turned on). When the HD_UP output is energized, the mechanism 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. UP_LS is **on** when the pressurizing mechanism is in the fully "up" position.

When AIR_VLV is **on**, air is admitted to pressurize the valve. The valve should be turned **off** when the target pressure, DES_PRES is reached. There are no limit switches indicating that the valve is open or closed.

If the valve pressure (VLV_PRES) drops more than 0.1 psi below DES_PRES during the step while the pressure is being checked, the STAT20_REJ bit is set (latched). This bit is part of a shift register that tracks the good/bad status of the valve as it moves down the line. Do not turn this bit **off**. The instructions used to generate the shift register will move this bit to the STAT21_REJ bit as the STAT19_REJ bit is shifted into the STAT20_REJ bit.

The height of the pressurizing head is measured using a linear variable differential transformer (LVDT) connected to an analog input module having an ADC whose output value (HGT_MEAS) corresponds to the lowest (75.0 mm) and highest (150.0 mm) heights as specified in the general instructions. The HGT_MEAS value should be converted into units of mm and stored in HD_HGT (a real number).

The pressure on the valve is measured with a pressure transducer connected to an analog input module having an ADC whose output value (PRES_MEAS) corresponds to the lowest (0.0 psi) and highest (100.0 psi) pressures as specified in the general instructions.

The PRES_MEAS value should be converted into units of psi and stored in VLV_PRES (a real number).

When the RUN internal coil is turned **on** for the first time only, the station assumes there are no valves in the station and waits for the first one. When the RUN internal coil is turned **off** the operation should pause at its current step, except when the pressure is being checked. If the RUN is turned off while the pressure is being checked, the timer must run to completion, and the program must advance to the next step. When the operation is paused, LIFT_SOL must remain **on**. All other outputs must be **off** when paused. If RUN is turned **on** while the operation of the station is paused, the station should resume its suspended step. Assume the conveyor is not running when RUN is **off**.

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.

There is a RESET internal coil that when **on**, restarts the operation. When RESET is **on**, the pressurizing head must be raised and the internal steps are set so that the ladder logic program assumes there are no valves in the check station. After a reset, RUN must be turned **on** to actually restart the station. In other words, after RESET is turned **on** and then back **off**, the next time RUN turns **on**, it is treated as the first time RUN turns **on**. The RESET coil must have no effect unless the operation is already paused. RUN must be ignored while the pressurizing head is being raised during the reset operation.

Assume the following internal tag/symbol assignments.

Tag/Var./Symbol	Description
RUN	When **on**, allow pressure check station to run. When **off**, pause (explained above).
RESET	When on resets operation of station (explained above).
HD_HGT	Pressurizing head height, in mm (a REAL). Your part of the ladder needs to calculate this value.
VLV_PRES	Pressure, in psi (a REAL). Your part of the ladder needs to calculate this value.
VLV_HGT	Height of valve, desired height of pressurizing head to do pressure check (a REAL). Set by another part of the ladder.
DES_PRES	Desired test pressure, in psi (a REAL). Set by another part of the ladder.
STAT20_REJ	Set if valve is to be rejected because it will not hold pressure. Cleared by another part of the ladder.

Assume the following physical input and physical output assignments.

Tag/Var./Symbol	Description
PROX	Reflective proximity switch that is **on** when valve is in position to be pressure-checked.
UP_LS	Limit switch that closes (**on**) when pressurizing head is fully up.
HGT_MEAS	Measurement of pressurizing head height, represents 75 to 150 mm.
PRES_MEAS	Measurement of pressure, represents 0 to 100 psi.
HD_DOWN	When **on**, moves the pressurizing head down. When **off**, the head does not change position.

HD_UP	When **on**, moves the pressurizing head up. When **off**, the head does not change position
LIFT_SOL	**On** to move carrier (and valve) up and off the conveyor and retain it there. When **off**, the valve and carrier falls into conveyor.
AIR_VLV	Opens air valve pressurize tested valve. **On** to open valve (pressurize), **off** to close.

The addresses associated with the physical inputs and outputs are:

Tag/Var./Symbol	ControlLogix	Micro800	MLogix	SLC-500
PROX	Local:1:I.Data.0	_IO_EM_DI_00	I:0/0	I:1/0
UP_LS	Local:1:I.Data.1	_IO_EM_DI_01	I:0/1	I:1/1
HD_DOWN	Local:2:O.Data.0	_IO_EM_DO_00	O:0/0	O:2/0
HD_UP	Local:2:O.Data.1	_IO_EM_DO_01	O:0/1	O:2/1
LIFT_SOL	Local:2:O.Data.2	_IO_EM_DO_02	O:0/2	O:2/2
AIR_VLV	Local:2:O.Data.3	_IO_EM_DO_03	O:0/3	O:2/3
HGT_MEAS	Local:3:I.Ch0Data	_IO_P1_AI_00	I:1.0	I:3.0
PRES_MEAS	Local:3:I.Ch1Data	_IO_P1_AI_01	I:1.1	I:3.1

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

	CLogix	Micro800	MLogix/SLC
Tag/Var./Symbol	Data Type	Data Type	Address
RUN	BOOL	BOOL	B33/20
RESET	BOOL	BOOL	B34/20
HD_HGT	REAL	REAL	F8:50
VLV_PRES	REAL	REAL	F8:51
VLV_HGT	REAL	REAL	F24:15
DES_PRES	REAL	REAL	F24:16
STAT20_REJ	BOOL	REAL	B21/20

SP7-10. Hole-drilling Station Control. Using the function chart approach, write a ladder logic program for the following station that drills a hole into a piece of metal.

Figure SP7.10 shows the general layout of a station that drills a hole into a metal piece. The horizontal X and Y location of the hole is not the same for every piece and so there are two internal memory locations that indicate the desired hole location. The actual x-horizontal displacement and y-horizontal displacement are measured by LVDTs (linear variable differential transformers) that are connected to analog input channels and are used to control the two horizontal ram positions.

The overall operation of the station is first described, followed by the details. The sequence of operation for the station follows these steps:

Metal piece coming down inbound conveyor contacts the stop at the end of the conveyor, turning **on** LS1.

Extend XCYL to push piece to desired x-horizontal drilling position.

Extend YCYL to push piece to desired y-horizontal position.

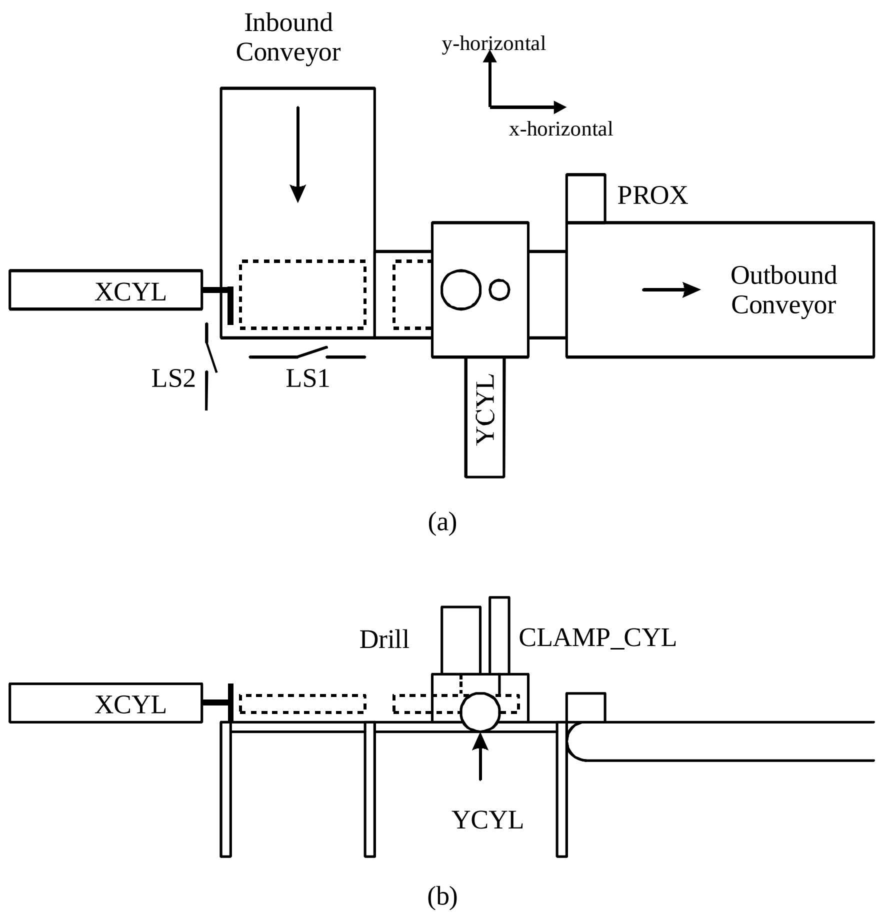


Figure SP7.10. Drilling station: *(a)* top view; *(b)* side view.

Extend CLAMP_CYL to clamp piece into drilling position. Simultaneously, retract YCYL. Assume 1 second is required to complete this operation.

Move drill down to fully extended position, pause 1.5 second, and then move drill up to fully retracted position.

Extend XCYL to push piece onto outbound conveyor (PROX senses).

Fully retract XCYL. LS2 turns **on** when it is fully retracted.

The operation then repeats. The inbound and outbound conveyors are both **on** while the station is running. These conveyors are controlled by this station.

XCYL is a pneumatic rams powered by a double action cylinder. When the XCYL_EXTEND output is **on**, the XCYL ram extends. When the XCYL_RETRACT output is **on**, the XCYL ram retracts. When neither XCYL output or both outputs are **on**, the XCYL ram stops moving and remains at its present position.

YCYL and CLAMP_CYL are pneumatic rams powered by single action cylinders with a spring return. For example, when the YCYL output is **on**, the YCYL ram extends until it reaches the fully extended position. When the power is removed (YCYL is **off**), the ram retracts until it is in its fully retracted position. There is no limit switch that indicates that these rams are fully retracted or extended. Assume 1 second is required to go from the fully extended position to the fully retracted position, or vice versa. CLAMP_CYL may be retracted at the same time the part is being pushed out of the station.

The ram that moves the drill down and up is a pneumatic ram powered by a double action cylinder (like XCYL). When the DRILL_DOWN output is **on**, the drill moves down. When the DRILL_UP output is **on**, the drill moves up. When neither drill ram output or both outputs are **on**, the drill stops moving vertically and remains at its present position. Limit switch LS3 indicates when the drill is fully up. Limit switch LS4 indicates when the drill is fully down.

The DRILL_MOTOR output must be turned **on** when the metal piece is being moved into its y-horizontal position so that it is turning at its maximum speed when it is starting to move down. The DRILL_MOTOR output is not turned **off** until the drill is fully up.

The PROX sensor is used to sense when the metal piece is fully on the outbound conveyor. As the piece is pushed out of the drilling station, PROX is turned **on** when the piece is at PROX, and then PROX is turned **off**. Only when PROX changes from **on** to **off**, can one be assured that the piece is fully on the outbound conveyor. When the metal piece is in any of the possible horizontal positions for the hole, the PROX is **off**.

The DES_X and DES_Y internal memory locations indicates the desired x-horizontal position and the desired y-horizontal position, respectively, in millimeters. Assume these values have been placed in the PLC memory from a supervisory PLC or computer.

The actual horizontal displacements are measured by linear variable differential transformers (LVDTs) that are connected to analog input channels. The range of the X-horizontal measurement is 150.0 to 300.0 mm and the range of the Y-horizontal measurement is 0.0 to 110.0 mm. For the measurements, the ADC output values (X_MEAS and Y_MEAS) correspond to the lowest and highest values as specified in the general instructions. The X_MEAS and Y_MEAS values should be converted into units of mm and stored in X_VAL and Y_VAL, respectively (both real numbers).

Use the appropriate LVDT measurement, converted to the proper units to determine when to stop the ram (x-horizontal or y-horizontal) at the desired position. Assume the LVDT measurement increases as the ram extends. (Hint: remember, there may be some inaccuracy in the measurement.)

When the start switch is pressed (turned **on**) for the first time only, the station assumes there is no piece in the station and waits for the first piece to arrive. When the stop switch is pressed (turned **off**) the operation should pause and all outputs (except CLAMP_CYL and DRILL_MOTOR) must be turned **off**. The operation cannot be paused when the drill motor is being moved down (to make sure the hole is correct). If the stop switch is pressed while the drill is being lowered, the drill should continue to be lowered and the operation should be paused at the end of the step. Pressing the start switch while the operation of the station is paused causes the station to resume its suspended operation.

There is a RESET_PB switch that when **on**, restarts the operation. When RESET is **on**, the internal step is set so that the ladder logic program waits for the next metal piece (but does not actually turn any outputs **on**). Also, retract all rams, including those that are double-acting when RESET_PB is momentarily pressed. If the drill must be retracted during reset, the drill motor must be **on**. All rams must be retracted before START_PB has any effect. START_PB must be pressed to actually restart the station. The RESET_PB switch must have no effect unless the operation is already paused. The RESET_PB switch must also have no effect while the drill is moving down or paused in the down position.

All delays must be accurate to within 0.1 seconds.

Assume the following input, output, and internal memory assignments.

Tag/Var./Symbol	Description
START_PB	Start push button switch, **on** (closed) to start.
STOP_PB	Stop push button switch, **off** (open) to stop.
RESET_PB	Reset push button switch, **on** (closed) when resetting.
LS1	Limit switch, **on** (closed) when metal piece is at the stop at the end of the conveyor.
LS2	Limit switch, **on** (closed) when XCYL ram is fully retracted.
LS3	Limit switch, **on** (closed) when drill is fully up.
LS4	Limit switch, **on** (closed) when drill is fully down.
PROX	Proximity switch, **on** as piece moves on outbound conveyor, **off** when it is fully on conveyor.
IN_CONV	Inbound conveyor motor control, **on** to run conveyor.
OUT_CONV	Outbound conveyor motor control, **on** to run conveyor.
XCYL_EXTEND	X ram extension control; **on** to extend XCYL ram, **off** stops ram movement (but does not retract ram).
XCYL_RETRACT	X ram retraction control; **on** to retract ram, **off** stops ram.
YCYL	Y ram control; **on** to extend ram, **off** retracts.
CLAMP_CYL	Clamp ram control, **on** to extend ram that clamps piece for drilling, **off** retracts.
DRILL_DOWN	Drill cylinder extension control; **on** to move drill down, **off** stops movement (but does not retract drill ram).
DRILL_UP	Drill cylinder retraction control; **on** to move drill up, **off** stops movement.
DRILL_MOTOR	Drill motor control; **on** to cause drill to turn, **off** stops drill.
X_MEAS	X-horizontal position measurement, represents 150 to 300 mm.
Y_MEAS	Y-horizontal position measurement, represents 0 to 110 mm.
X_VAL	Actual x-horizontal position, in millimeters (a REAL).
Y_VAL	Actual y-horizontal position, in millimeters (a REAL).
DES_X	Desired x-horizontal position, in millimeters (a REAL).
DES_Y	Desired y-horizontal position, in millimeters (a REAL).

The addresses associated with the physical inputs and outputs are:

Tag/Var./Symbol	ControlLogix	Micro800	MLogix	SLC-500
START_PB	Local:1:I.Data.0	_IO_EM_DI_00	I:0/0	I:1/0
STOP_PB	Local:1:I.Data.1	_IO_EM_DI_01	I:0/1	I:1/1
RESET_PB	Local:1:I.Data.2	_IO_EM_DI_02	I:0/2	I:1/2
LS1	Local:1:I.Data.3	_IO_EM_DI_03	I:0/3	I:1/3
LS2	Local:1:I.Data.4	_IO_EM_DI_04	I:0/4	I:1/4
LS3	Local:1:I.Data.5	_IO_EM_DI_05	I:0/5	I:1/5
LS4	Local:1:I.Data.6	_IO_EM_DI_06	I:0/6	I:1/6

PROX	Local:1:I.Data.7	_IO_EM_DI_07	I:0/7	I:1/7
IN_CONV	Local:2:O.Data.0	_IO_EM_DO_00	O:0/0	O:2/0
OUT_CONV	Local:2:O.Data.1	_IO_EM_DO_01	O:0/1	O:2/1
XCYL_EXTEND	Local:2:O.Data.2	_IO_EM_DO_02	O:0/2	O:2/2
XCYL_RETRACT	Local:2:O.Data.3	_IO_EM_DO_03	O:0/3	O:2/3
YCYL	Local:2:O.Data.4	_IO_EM_DO_04	O:0/4	O:2/4
CLAMP_CYL	Local:2:O.Data.5	_IO_EM_DO_05	O:0/5	O:2/5
DRILL_DOWN	Local:2:O.Data.6	_IO_EM_DO_06	O:0/6	O:2/6
DRILL_UP	Local:2:O.Data.7	_IO_EM_DO_07	O:0/7	O:2/7
DRILL_MOTOR	Local:2:O.Data.8	_IO_EM_DO_08	O:0/8	O:2/8
X_MEAS	Local:3:I.Ch0Data	_IO_P1_AI_00	I:1.0	I:3.0
Y_MEAS	Local:3:I.Ch1Data	_IO_P1_AI_01	I:1.1	I:3.1

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

Tag/Var./Symbol	CLogix Data Type	Micro800 Data Type	MLogix/SLC Address
X_VAL	REAL	REAL	F8:50
Y_VAL	REAL	REAL	F8:51
DES_X	REAL	REAL	F12:0
DES_Y	REAL	REAL	F12:1

SP7-11. Part Width Sorter Control Using the function chart approach, write a ladder logic program to control the following station that sorts parts into one of three bins, depending on the size of the part.

A top view of the system is shown in Figure SP7.11. As parts proceed down the conveyor, the width is measured and then the part is ejected into one of three bins. To eject the part into a bin, the corresponding eject cylinder control must be **on** for 1 sec (when the part is at the appropriate position). The parts are sorted into bins as follows:

Width	**Bin**
1 ± 0.1"	Bin 1
2 ± 0.1"	Bin 2
All others	Bin 3

The conveyor does not need to be stopped while ejecting the part. The eject mechanisms are each controlled by a single-acting air cylinder. Once an eject control is energized, the eject bar is extended, causing the part to be pushed into the bin. The eject bar remains extended as long as power is applied (turned **on**). The eject bar retracts when power is removed (turned **off**).

The width is measured by using the two ultrasonic distance transducers UX1 and UX2. When PROX is ON (indicating part is in position to be measured), the width is determined by subtracting the sum of the two transducer, UX1 and UX2, readings from 16 (UX1 and UX2 are 16 inches apart):

Part width (inches) = 16 – (UX1 + UX2)

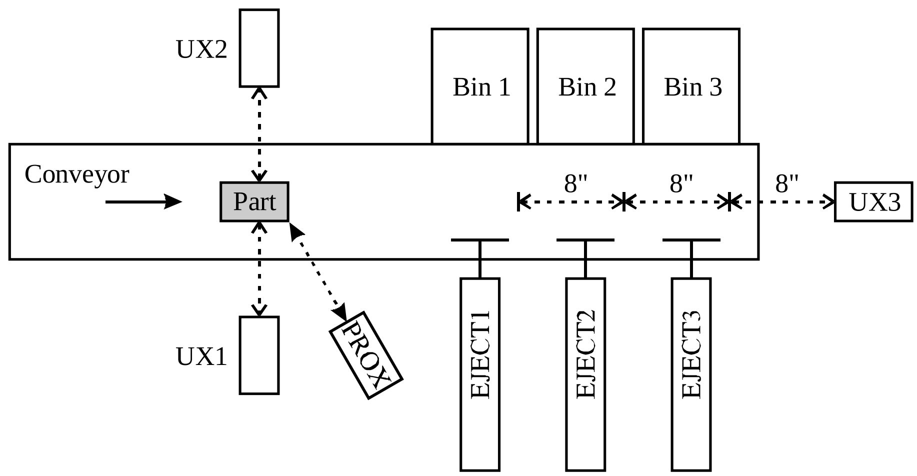


Figure SP7.11. Width sorting station.

The reading and conversion to part width must be done instantaneously and does not require that the conveyor belt be stopped.

The position at which a part is to be ejected into a bin is a certain distance from the UX3 ultrasonic distance transducer:

Bin	Distance from UX3
Bin 1	24"
Bin 2	16"
Bin 3	8"

Note that as soon as an eject cylinder control is turned **on**, the UX3 reading is invalid because the part is being moved off the conveyor. The actual distances are measured by ultrasonic transducers that are connected to analog input channels. The highest and lowest values of all three distance measurements are 4.0 and 30.0 inches, respectively. The ADC output values (UX1_MEAS, UX2_MEAS, and UX3_MEAS) correspond to the lowest and highest values as specified in the general instructions. The distance values should be converted into units of inches and stored in internal memory locations (real numbers).

The conveyor is **on** whenever the station is running.

A RUN internal coil that operates as a start/pause. When RUN is **on**, the station operates. When it is **off**, the station operation is paused. When it is turned **on** for the first time, no parts are assumed on the conveyor from PROX1 to UX3.

If the station is paused during the time that the part is being ejected into a bin, the step should **complete**. The operation **must not** pause when any of the eject cylinder controls are **on**. If the RUN coil is turned **off** during the time the piece is being ejected, the step should be completed (and advanced to the next step) before pausing. When the RUN internal coil turns **on** when the operation is paused, the station should resume the suspended step. When paused, **it is permissible to advance** to the next step as long as equipment will not be damaged. When the station is paused, the conveyor should be turned off.

A separate INT_RESET internal coil is provided which resets any internal states so that when the RUN internal coil is turned **on**, the conveyor has already been cleared of any items from PROX1 to UX3. You may assume that the INT_RESET coil can be ON only when RUN is OFF.

The RUN and INT_RESET internal coils are **inputs**. Your ladder logic cannot control them and thus **must** not appear as an output on any rung!

Assume the tolerance on all timer values is 0.1 seconds.

Assume the following input, output, and internal memory assignments.

Tag/Var./Symbol	Description
PROX	Proximity sensor, **on** when part is in position for width measurement.
UX1_MEAS	Distance sensor raw measurement, represents 4 – 30 inches.
UX2_MEAS	Distance sensor, same format and range as UX1_MEAS.
UX3_MEAS	Distance sensor, same format and range as UX1_MEAS.
RUN	When **on**, allow system to operate. When **off**, pause (explained above).
INT_RESET	When on resets operation of station (explained above)
CONV_MTR	Conveyor motor control; **on** to run conveyor belt.
EJECT1	**On** to operate cylinder to eject part into bin 1, must remain **on** for 1 sec to eject part.
EJECT2	**On** to operate cylinder to eject part into bin 2. Operates in similar manner as EJECT1.
EJECT3	**On** to operate cylinder to eject part into bin 3. Operates in similar manner as EJECT1.

The addresses associated with the physical inputs and outputs are:

Tag/Var./Symbol	ControlLogix	Micro800	MLogix	SLC-500
PROX	Local:1:I.Data.0	_IO_EM_DI_00	I:0/0	I:1/0
CONV_MTR	Local:2:O.Data.0	_IO_EM_DO_00	O:0/0	O:2/0
EJECT1	Local:2:O.Data.1	_IO_EM_DO_01	O:0/1	O:2/1
EJECT2	Local:2:O.Data.2	_IO_EM_DO_02	O:0/2	O:2/2
EJECT3	Local:2:O.Data.3	_IO_EM_DO_03	O:0/3	O:2/3
UX1_MEAS	Local:3:I.Ch0Data	_IO_P1_AI_00	I:1.0	I:3.0
UX2_MEAS	Local:3:I.Ch1Data	_IO_P1_AI_01	I:1.1	I:3.1
UX3_MEAS	Local:3:I.Ch2Data	_IO_P1_AI_02	I:1.2	I:3.2

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

	CLogix	Micro800	MLogix/SLC
Tag/Var./Symbol	Data Type	Data Type	Address
RUN	BOOL	BOOL	B33/20
INT_RESET	BOOL	BOOL	B34/20

SP7-12. Part Height Sorter Control. Using the function chart approach, write a ladder logic program to control the following station that sorts parts onto one of four outbound conveyors, depending on the height of the part.

Top and side views of the station are shown in Figure SP7.12. There are multiple conveyor sections. Parts come into the station on the inbound conveyor, CONV_1. Assume that the inbound and outbound conveyors are controlled by another PLC. A short conveyor, CONV_2, is not much longer than the part and holds the part as it is being measured. This conveyor must be stopped while the part height is being measured. Following the measurement conveyor, the eject conveyor, CONV_3 conveys the part to the appropriate place where the part is ejected onto one of four outbound conveyors. To eject the part onto a conveyor, the corresponding eject cylinder control must be **on** for 1 second (when the part is at the appropriate eject position). The parts are sorted onto the conveyors as follows:

Height	Outbound Conveyor
60 ± 4 mm	OUTCONV_1
75 ± 4 mm	OUTCONV_2
90 ± 4 mm	OUTCONV_3
All others	OUTCONV_4

The sequence of measuring each part and ejecting it onto the proper conveyor is as follows:

Wait for next part coming down conveyor, sensed by PROX

CONV_2 is stopped, the gate cylinder is extended, and the MEAS_RAM is extended to measure the part. Since the part height is variable, the extension of MEAS_RAM must be timed. Allow 2 seconds for the maximum extension. The pressure exerted by the ram will not damage the part.

Measure the height and display it on the 2-digit LED display.

Retract the MEAS_RAM. LS1 is **on** when the ram is fully retracted.

Move the part onto the eject conveyor and to the proper eject position.

When the part is at the proper eject position, CONV_3 is stopped and the part is ejected onto the proper outbound conveyor.

The GATE cylinder is retracted and the operation is repeated.

The parts are on carriers, so that even when the carriers are next to each other on the inbound conveyor, there is sufficient space between parts for the GATE ram to extend and allow the first part into the station and keep all others out. The PROX sensor is in such a position that activating the gate will prevent the next part from entering the measuring station. When PROX is **on**, the part is also in position to be measured.

The CONV_2 conveyor should be **off** when measuring the part (including extending and retracting the ram). The CONV_3 conveyor should only be **on** when moving the part to the eject position. CONV_3 must be stopped while ejecting the part onto one of the outbound conveyors.

The GATE and MEAS_RAM control single-acting pneumatic cylinders. Once a control output is energized, the cylinder extends. The cylinder remains extended as long as the control is **on**. The cylinder retracts when the control is **off**. The GATE cylinder extension/retraction does not need to be timed.

The eject mechanisms are powered by single-acting pneumatic cylinders. Once a control output is energized, the eject bar extends, causing the part to be pushed onto the

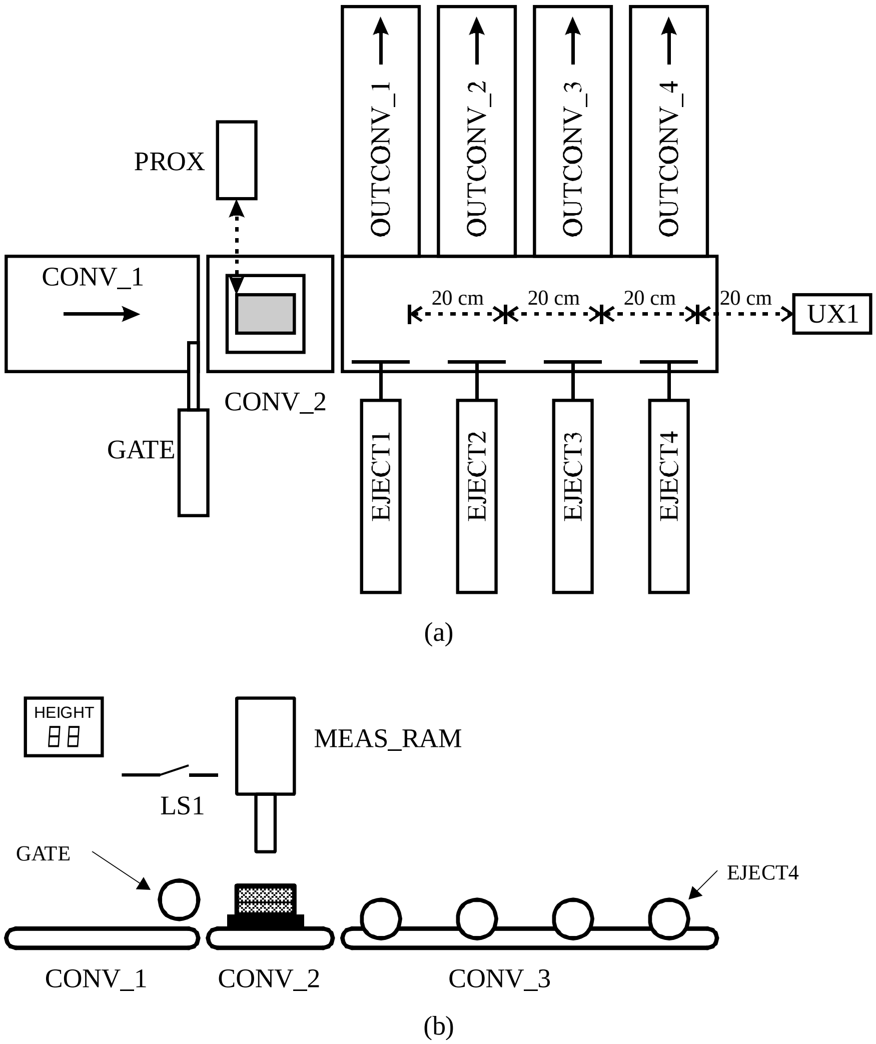


Figure SP7.12. Height sorting station: *(a)* top view; *(b)* side view.

outbound conveyor. The eject bar remains extended as long as the control is **on**. The eject bar retracts when the control is **off**.

The height is measured by a linear variable differential transformer (LVDT) attached to the measuring ram. The LVDT measures 0 – 100 mm, but it is not the part height. When MEAS_RAM is fully retracted (LVDT is measuring 0 mm), the end of the ram is actually calibrated to be 150 mm above the part carrier. Therefore, when the ram is fully extended (LVDT is measuring 100 mm) the end of the ram is 50 mm above the part carrier.

The LVDT measurement range is 0.0 to 100.0 mm and is connected to an analog input channel whose ADC output value (HGT_MEAS) corresponds to the lowest and highest values as specified in the general instructions. The height value should be converted into units of mm and stored in the HGT_VAL internal memory location (real number).

The height measurement, in mm, should be placed into the HGT_HMI internal tag/variable for display on an HMI. HGT_HMI is only changed when a new measurement is made. At all other times, it shows the last measurement.

The position at which a part is to be ejected onto an outbound conveyor is a certain distance from the UX1 ultrasonic distance transducer:

Outbound Conveyor	**Distance from UX1**
OUTCONV_1	80 cm
OUTCONV_2	60 cm
OUTCONV_3	40 cm
OUTCONV_4	20 cm

Note that as soon as an eject cylinder control is turned **on**, the UX1 reading is invalid because the part is being moved off the conveyor. The actual distance is measured by an ultrasonic transducer connected to an analog input channel. The highest and lowest value of the distance measurement is 15.0 and 100.0 cm, respectively. The analor input channel value (UX1_MEAS) corresponds to the lowest and highest values as specified in the general instructions. The distance value should be converted into units of cm and stored in an internal memory location (real number).

The RUN internal coil operates as a start/pause. Assume it is controlled by another part of the ladder. When RUN is **on**, the station operates. When RUN is **off**, the station operation is paused. When it is turned **on** for the first time, no parts are assumed on the conveyor from PROX1 to UX1.

If the station is paused during the time that the part is being ejected onto an outbound conveyor, the step should **complete**. The operation **must not** pause when any of the eject cylinder controls are **on**. If the RUN coil is turned **off** during the time the part is being ejected, the step should be completed (and advanced to the next step) before pausing. When the RUN internal coil turns **on** when the operation is paused, the station should resume the suspended step. When paused, **it is permissible to advance** to the next step as long as equipment will not be damaged. When the station is paused, the conveyors should be turned **off**.

A separate INT_RESET internal coil is provided which retracts the rams and resets any internal states so that when the RUN internal coil is turned **on**, the conveyor has already been cleared of any items from PROX1 to UX1. You may assume that the INT_RESET coil can be **on** only when RUN is **off**.

The RUN and INT_RESET internal coils are **inputs**. Your ladder logic cannot control them and thus **must** not appear as an output on any rung!

Assume the tolerance on all timer values is 0.1 seconds.

Assume the following input, output, and internal memory assignments.

Tag/Var./Symbol	Description
PROX	Proximity sensor, **on** when part is in position for height measurement.
LS1	Limit switch, **on** when MEAS_RAM retracted.
HGT_MEAS	LVDT length measurement, represents 0 – 100 mm.
UX1_MEAS	Distance sensor raw measurement, represents 15 – 100 cm.

RUN	When **on**, allow system to operate. When **off**, pause (explained above).
INT_RESET	When on resets operation of station (explained above)
GATE	Gate ram control; **on** to extend ram, **off** retracts ram.
MEAS_RAM	Measuring ram control; **on** to extend ram, **off** retracts ram.
CONV_2	Short conveyor motor control; **on** to run conveyor belt.
CONV_3	Eject conveyor motor control; **on** to run conveyor belt.
EJECT1	**On** to operate cylinder to eject part onto OUTCONV_1, must remain **on** for 1 sec to eject part.
EJECT2	**On** to operate cylinder to eject part onto OUTCONV_2.
EJECT3	**On** to operate cylinder to eject part onto OUTCONV_3.
EJECT4	**On** to operate cylinder to eject part onto OUTCONV_4.
HGT_VAL	Part height, in mm (a REAL).
HGT_HMI	Part height, in mm (an INT).
UX1_VAL	Distance, in cm (a REAL).

The addresses associated with the physical inputs and outputs are:

Tag/Var./Symbol	ControlLogix	Micro800	MLogix	SLC-500
PROX	Local:1:I.Data.0	_IO_EM_DI_00	I:0/0	I:1/0
LS1	Local:1:I.Data.1	_IO_EM_DI_01	I:0/1	I:1/1
GATE	Local:2:O.Data.0	_IO_EM_DO_00	O:0/0	O:2/0
MEAS_RAM	Local:2:O.Data.1	_IO_EM_DO_01	O:0/1	O:2/1
CONV_2	Local:2:O.Data.2	_IO_EM_DO_02	O:0/2	O:2/2
CONV_3	Local:2:O.Data.3	_IO_EM_DO_03	O:0/3	O:2/3
EJECT1	Local:2:O.Data.4	_IO_EM_DO_04	O:0/4	O:2/4
EJECT2	Local:2:O.Data.5	_IO_EM_DO_05	O:0/5	O:2/5
EJECT3	Local:2:O.Data.6	_IO_EM_DO_06	O:0/6	O:2/6
EJECT4	Local:2:O.Data.7	_IO_EM_DO_07	O:0/7	O:2/7
HGT_MEAS	Local:3:I.Ch0Data	_IO_P1_AI_00	I:1.0	I:3.0
UX1_MEAS	Local:3:I.Ch1Data	_IO_P1_AI_01	I:1.1	I:3.1

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

	CLogix	Micro800	MLogix/SLC
Tag/Var./Symbol	Data Type	Data Type	Address
RUN	BOOL	BOOL	B33/20
INT_RESET	BOOL	BOOL	B34/20
HGT_VAL	REAL	REAL	F8:50
HGT_HMI	INT	INT	N11:2
UX1_VAL	REAL	REAL	F8:51

SP7-13. Batch Reactor Control. Using the function chart approach, implement the program for the following batch process.

This application deals with the automation of a batch process. Figure SP7.13 shows the process and its associated equipment. There are two ingredients, A and B that will be mixed in the reactor tank. The two ingredients react as they are agitated to form a thicker product. Therefore, the agitator speed is controlled throughout the reaction time and then the product is dumped into a product tank, called a QA tank. The overall operation of the batch cycle is first described, followed by the details. The batch cycle sequence follows these major steps:

Wait for READY signal to be **on** (When on, it signifies that the QA tank has enough room for this batch, and other permissives)

Fill desired amount of ingredient A

Fill desired amount of ingredient B

Agitate at constant speed (500 rpm) for 10 minutes

Agitate at decreasing speed (500 - 260 rpm) over a period of 40 minutes (2400 seconds)

Dump into QA tank (agitate at 260 rpm)

The operation then repeats. The READY signal is controlled by another part of the ladder and thus must not appear as an output on any rung of your solution.

Ingredient A is added to the tank by opening the VALVE_A solenoid valve. Ingredient B is added to the tank by opening the VALVE_B solenoid valve. The desired amount (gallons) of A and B to be loaded into the reactor are in the DES_A and DES_B integer locations, respectively. The values in these locations are from a recipe and are therefore read-only (do not write into these locations).

Ingredient B may only be added after the desired amount of A has been added to the tank.

The A and B inlet flows and the outlet flow are monitored by pulse-type flow meters. These meters output a pulse for every gallon that passes through the flow meter. Therefore, to measure the amount of material that flows in the stream, one must count the flow meter pulses (PULSE_A, PULSE_B, or PULSE_OUT). The flows are low enough that regular discrete inputs are used. No high-speed counter modules are used in this application.

The AGIT_ON output turns the agitator motor on/off. The agitator speed (0 to 1000 rpm) is controlled by an analog output channel from the PLC. The AGIT_AO input integer value to the DAC of the analog output module corresponds to the lowest and highest speeds as specified in the general instructions. The desired agitator speed in rpm should be stored in AGIT_RPM (a real number) before being converted to the DAC integer, AGIT_AO.

The agitator must be run at 100 rpm while ingredient B is flowing into the tank. The step during which the agitator speed is decreased is done because the mixture thickens and the agitator motor will overheat if the speed is kept constant. The agitator speed decrease corresponds to a change of 0.1 rpm/sec during the time of this step. The agitator must be **off** when waiting for ready and while adding ingredient A. The agitator must be run at a constant 260 rpm while emptying the tank.

Use the flow meter on the outlet flow to determine when the tank is empty. Assume that the product mixture has the same volume (in gallons) as the amount of A and B combined.

The agitator motor current is monitored during the batch cycle and the following alarms are generated:

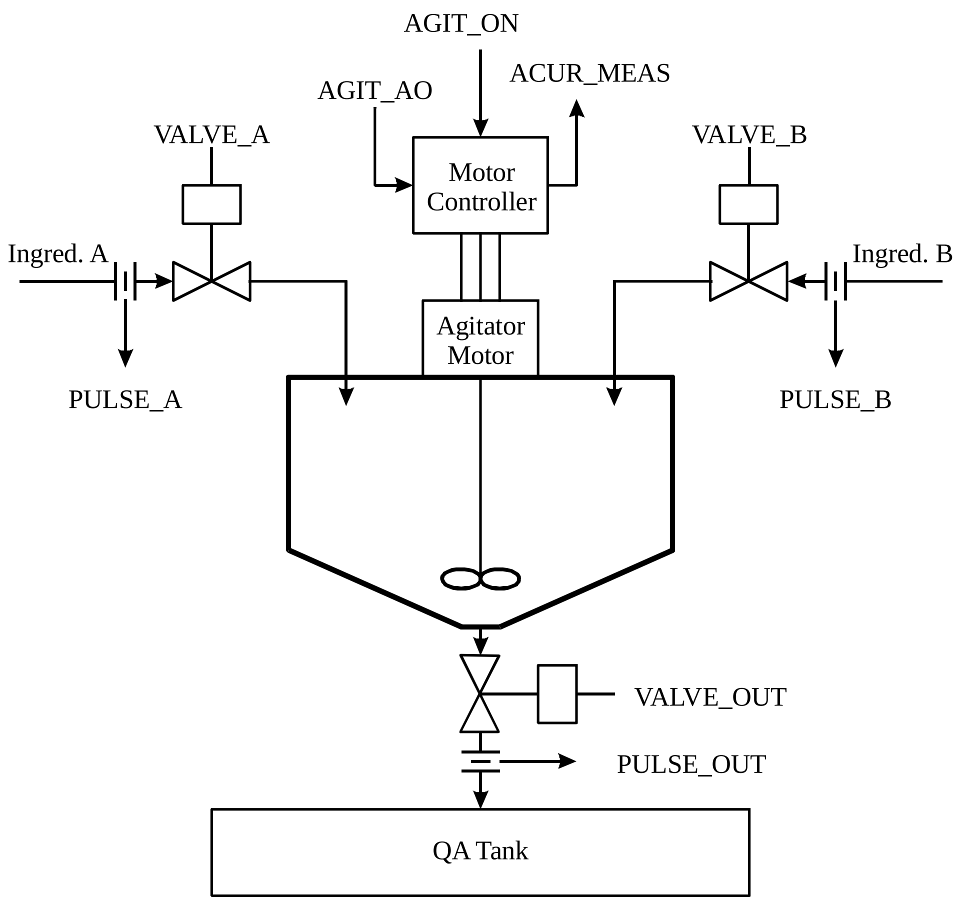


Figure SP7.13. Batch process.

CURR_WARN	**On** when agitator current is ≥ 15 amps, but < 18 amps
CURR_ALARM	**On** when agitator current is ≥ 18 amps

Do not latch these alarms.

The agitator current is measured by a current transformer connected to an analog input channel. The current transformer measures current in the range of 0.0 to 20.0 amperes. The ADC output value (ACUR_MEAS) corresponds to the lowest and highest values as specified in the general instructions. The measurement values should be converted into units of amperes and stored in AGIT_CURR (a real number).

When the start switch is pressed (turned **on**) for the first time, the batch process goes to the step that waits for the READY signal. When the stop switch is pressed (turned **off**) the operation should pause and all outputs must be turned **off**. When paused, all timer values must be retained. The agitator speed is ignored when the AGIT_ON output is **off**, and so the agitator speed command does not change when paused. The operation can be paused in any step. Pressing the start switch while the batch operation is paused causes the batch operation to resume its suspended operation.

There is no provision for resetting the batch operation.

All time delays must be accurate to within 1 second.

Assume the following input, output, and internal memory assignments.

Tag/Var./Symbol	Description
START_PB	Start push button switch, **on** (closed) to start.
STOP_PB	Stop push button switch, **off** (open) to stop.
PULSE_A	Used to indicate amount of ingredient A that is flowing past flow meter. Each pulse indicates that 1 (one) gallon has flowed past.
PULSE_B	Used to indicate amount of ingredient B that is flowing past flow meter. Each pulse indicates that 1 (one) gallon has flowed past.
PULSE_OUT	Used to indicate amount of outlet material that is flowing past flow meter. Each pulse indicates that 1 (one) gallon has flowed past.
VALVE_A	**On** to open solenoid valve to allow ingredient A to flow into mixing vessel. **Off** closes valve.
VALVE_B	**On** to open solenoid valve to allow ingredient B to flow into mixing vessel. **Off** closes valve.
VALVE_OUT	**On** to open solenoid valve to allow mixture to flow out of mixing vessel. **Off** closes valve.
AGIT_ON	**On** to run agitator motor, when OFF, agitator speed command is ignored.
CURR_WARN	**On** when agitator current is 15 amps, but < 18 amps.
CURR_ALARM	**On** when agitator current is 18 amps.
ACUR_MEAS	Raw agitator motor current measurement, represents 0 – 20 amps.
AGIT_AO	Agitator motor controller speed command, represents 0 – 1000 rpm.
READY	Ready signal, ON when permissives are satisfied and batch cycle may begin by adding ingredient A.
DES_A	Desired amount of ingredient A to add, in gallons (a REAL).
DES_B	Desired amount of ingredient B to add, in gallons (a REAL).
AGIT_RPM	Commanded agitator speed, in rpm (a REAL).
AGIT_CURR	Agitator motor current, in amps (a REAL).

The addresses associated with the physical inputs and outputs are:

Tag/Var./Symbol	ControlLogix	Micro800	MLogix	SLC-500
START_PB	Local:1:I.Data.0	_IO_EM_DI_00	I:0/0	I:1/0
STOP_PB	Local:1:I.Data.1	_IO_EM_DI_01	I:0/1	I:1/1
PULSE_A	Local:1:I.Data.2	_IO_EM_DI_02	I:0/2	I:1/2
PULSE_B	Local:1:I.Data.3	_IO_EM_DI_03	I:0/3	I:1/3

PULSE_OUT	Local:1:I.Data.4	_IO_EM_DI_04	I:0/4	I:1/4
VALVE_A	Local:2:O.Data.0	_IO_EM_DO_00	O:0/0	O:2/0
VALVE_B	Local:2:O.Data.1	_IO_EM_DO_01	O:0/1	O:2/1
VALVE_OUT	Local:2:O.Data.2	_IO_EM_DO_02	O:0/2	O:2/2
AGIT_ON	Local:2:O.Data.3	_IO_EM_DO_03	O:0/3	O:2/3
CURR_WARN	Local:2:O.Data.4	_IO_EM_DO_04	O:0/4	O:2/4
CURR_ALARM	Local:2:O.Data.5	_IO_EM_DO_05	O:0/5	O:2/5
ACUR_MEAS	Local:3:I.Ch0Data	_IO_P1_AI_00	I:1.0	I:3.0
AGIT_AO	Local:4:O.Ch0Data	_IO_P2_AO_00	O:2.0	I:4.0

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

Tag/Var./Symbol	CLogix Data Type	Micro800 Data Type	MLogix/SLC Address
READY	BOOL	BOOL	B33/20
AGIT_RPM	REAL	REAL	F8:50
AGIT_CURR	REAL	REAL	F8:51
DES_A	REAL	REAL	F12:0
DES_B	REAL	REAL	F12:1

SP7-14. Multi-tank Batch Control. Using the function chart approach, implement the program for the following batch process.

A plant consists of four tanks with valves to transfer the materials through the system (Figure SP7.14). Each tank is fitted with a load cell that measures weight, and thus determines the level. Tank 3 has an electric heating element wrapped around it and an associated temperature sensor. Tank 4, the reaction vessel, is equipped with a stirrer to mix the materials while they are being transferred into the reaction vessel and the reaction is taking place. The reaction tank has the capacity of tanks 1, 2, and 3 combined.

The process is operated as follows:

Tanks 1, 2 and 3 are to be filled simultaneously from supply reservoirs of chemicals, by opening valves XV101, XV102, and XV103. The inlet valve for each tank should be closed as soon as each tank is filled.

As soon as tank 3 is full, the heating element in tank 3 is activated, raising the liquid temperature to 70°C.

When all tanks are filled and the temperature of tank 3 is 70°C, valves XV104, XV105 and XV106 are opened and pumps P-105 and P-106 are run to transfer all the material in tank 1 and half of the contents of tanks 2 and 3 to the reaction vessel, tank 4. The vibrator must be turned **on** when tank 1 is being emptied. The stirrer must run when tank 4 has anything in it. When any tank becomes empty, its outlet valve must be closed and pump stopped (e.g., when tank 1 is empty, valve XV104 must be closed).

After tank 1 and half of tanks 2 and 3 have been emptied into tank 4, the stirrer is kept **on** for an additional 20 seconds.

After the end of this 20 second interval, a 250 second interval is started. During this interval, the remainder of tanks 2 and 3 are emptied into tank 4. Assume that it takes less than 250 seconds to empty the last half of tanks 2 and 3. The

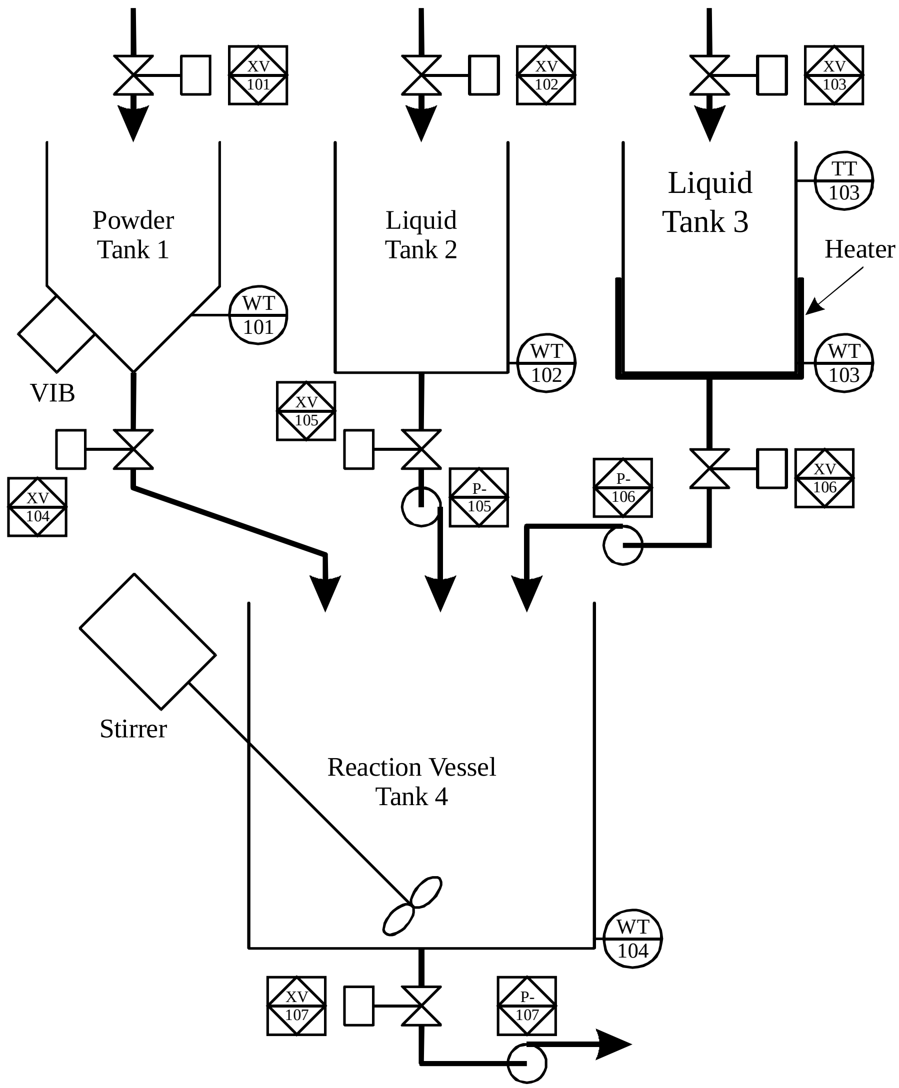


Figure SP7.14. Multi-tank batch process.

corresponding valve to empty a tank must be turned **off** as soon as the tank is empty and the heater for tank 3 must be turned off when tank 3 is empty (do not wait for the end of the 250 second interval).

When the stirrer has been **on** for the 250 second interval, valve XV107 is opened and pump P-107 is run to transfer the product to another tank.

Valve XV107 is closed and pump P-107 stopped once tank 4 is empty.

One cycle is thus completed, and the process should begin again without operator intervention (if the stop switch has not been pressed).

The heater is initially turned **on**, until the temperature reaches 70°C. After this point, the heater is turned **off** until the temperature is 68°C, at which time it is turned back **on** until the temperature reaches 70°C, when it is then turned **off** again. This operation continues until tank 3 is empty.

The operation may be stopped (paused) only when tanks 1, 2, and 3 are being filled or heated. If the stop pushbutton is pressed at any other time, the operation must continue until tank 4 is empty, and then it must pause until the start switch is pressed. The heater operation must remain **on**, if paused. A reset switch is not provided, nor needed for this application.

The only assumption you may make about the time to complete any of the operations that are to occur simultaneously is given above (emptying of last half of tanks 2 and 3). For example, you may not assume that the time to complete the filling and heating of tank 3 takes longer than the filling of tanks 1 and 2. Steps that are done simultaneously should be indicated as parallel tasks in the function chart and should be implemented as separate steps.

The tank weights are measured by load cells that are connected to analog input channels. All load cells are calibrated to measure 0.0 to 1000.0 pounds. For all four weight measurements, the ADC output values (WT101_MEAS, WT102_MEAS, WT103_MEAS, and WT104_MEAS) correspond to the lowest and highest values as specified in the general instructions. The weight values should be converted into units of pounds and stored in internal memory locations (real numbers).

The empty and full weights of the various tanks are:

	Empty (lbs)	**Full (lbs)**
Tank 1	200	300
Tank 2	150	400
Tank 3	200	350
Tank 4	300	800

A resistance temperature device (RTD) measures the temperature of tank 3. The RTD is connected to a transmitter whose output of 4 to 20 mA is calibrated to represent 0 to 100 C. The transmitter output is connected to an analog input module having an ADC whose output integer value (TT103_MEAS) corresponds to the lowest and highest temperatures as specified in the general instructions. The temperature measurement value should be converted into units of C and stored in an internal memory location (real numbers).

Assume the following physical input and output assignments:

Tag/Var./Symbol	Description
START_PB	Start push button switch, **on** (closed) to start.
STOP_PB	Stop push button switch, **off** (open) to stop.
WT101_MEAS	Tank 1 weight measurement, represents 0 – 1000 pounds.
WT102_MEAS	Tank 2 weight, same format and range as WT101_MEAS.
WT103_MEAS	Tank 3 weight, same format and range as WT101_MEAS.
WT104_MEAS	Tank 4 weight, same format and range as WT101_MEAS.
TT103_MEAS	Tank 3 temperature measurement, represents 0 – 100 °C; ADC integer range listed before SP7-2.
XV101_SOL	**On** to open XV101. **Off** closes valve.
XV102_SOL	**On** to open XV102. **Off** closes valve.
XV103_SOL	**On** to open XV103. **Off** closes valve.

XV104_SOL	**On** to open XV104. **Off** closes valve.
XV105_SOL	**On** to open XV105. **Off** closes valve.
XV106_SOL	**On** to open XV106. **Off** closes valve.
XV107_SOL	**On** to open XV107. **Off** closes valve.
P105_RUN	**On** to run pump P-105. **Off** stops pump.
P106_RUN	**On** to run pump P-106. **Off** stops pump.
P107_RUN	**On** to run pump P-107. **Off** stops pump.
STIR_ON	Tank 4 stirrer control, **on** to stir tank 4.
HEAT_RLY	Tank 3 heat relay control, **on** to heat tank 3.
VIB_ON	Tank 1 vibrator control, **on** to vibrate tank during emptying.
WT101_VAL	Tank 1 weight, in lbs. (a REAL).
WT102_VAL	Tank 2 weight, in lbs. (a REAL).
WT103_VAL	Tank 3 weight, in lbs. (a REAL).
WT104_VAL	Tank 4 weight, in lbs. (a REAL).
TT103_VAL	Tank 3 temperature, in °C (a REAL).

The addresses associated with the physical inputs and outputs are:

Tag/Var./Symbol	ControlLogix	Micro800	MLogix	SLC-500
START_PB	Local:1:I.Data.0	_IO_EM_DI_00	I:0/0	I:1/0
STOP_PB	Local:1:I.Data.1	_IO_EM_DI_01	I:0/1	I:1/1
XV101_SOL	Local:2:O.Data.0	_IO_EM_DO_00	O:0/0	O:2/0
XV102_SOL	Local:2:O.Data.1	_IO_EM_DO_01	O:0/1	O:2/1
XV103_SOL	Local:2:O.Data.2	_IO_EM_DO_02	O:0/2	O:2/2
XV104_SOL	Local:2:O.Data.3	_IO_EM_DO_03	O:0/3	O:2/3
XV105_SOL	Local:2:O.Data.4	_IO_EM_DO_04	O:0/4	O:2/4
XV106_SOL	Local:2:O.Data.5	_IO_EM_DO_05	O:0/5	O:2/5
XV107_SOL	Local:2:O.Data.6	_IO_EM_DO_06	O:0/6	O:2/6
P105_RUN	Local:2:O.Data.7	_IO_EM_DO_07	O:0/7	O:2/7
P106_RUN	Local:2:O.Data.8	_IO_EM_DO_08	O:0/8	O:2/8
P107_RUN	Local:2:O.Data.9	_IO_EM_DO_09	O:0/9	O:2/9
STIR_ON	Local:2:O.Data.10	_IO_P1_DO_00	O:0/10	O:2/10
HEAT_RLY	Local:2:O.Data.11	_IO_P1_DO_01	O:0/11	O:2/11
VIB_ON	Local:2:O.Data.12	_IO_P1_DO_02	O:1/0	O:2/12
WT101_MEAS	Local:3:I.Ch0Data	_IO_P2_AI_00	I:2.0	I:3.0
WT102_MEAS	Local:3:I.Ch1Data	_IO_P2_AI_01	I:2.1	I:3.1
WT103_MEAS	Local:3:I.Ch2Data	_IO_P2_AI_02	I:2.2	I:3.2
WT104_MEAS	Local:3:I.Ch3Data	_IO_P2_AI_03	I:2.3	I:3.3
TT103_MEAS	Local:3:I.Ch4Data	_IO_P3_AI_00	I:3.0	I:4.0

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

Tag/Var./Symbol	CLogix Data Type	Micro800 Data Type	MLogix/SLC Address
WT101_VAL	REAL	REAL	F8:4
WT102_VAL	REAL	REAL	F8:5
WT103_VAL	REAL	REAL	F8:6
WT104_VAL	REAL	REAL	F8:7
TT103_VAL	REAL	REAL	F8:8

SP7-15. Repeat problem SP7-11, but show the parallel paths (depending on part width) on your function chart. SHOW THE PARALLEL OPERATIONS ON YOUR FUNCTION CHART AS SEPARATE STEPS. Implement the revised function chart in ladder logic. Do not try to combine steps.

SP7-16. Repeat problem SP7-12, but show the parallel paths (depending on part height) on your function chart. SHOW THE PARALLEL OPERATIONS ON YOUR FUNCTION CHART AS SEPARATE STEPS. Implement the revised function chart in ladder logic. Do not try to combine steps.