ME576 Laboratory 10 Assignment

Introduction

For this lab, the conveyor trainer will be operated using the Siemens S5 PLC. The conveyor system is supposed to sort the metal pegs from rings on the moving conveyor and assemble them. The metal pegs are detected by an inductive sensor (I20.0) for sorting. The peg and ring assembly will be inspected for correct assembly and the defective ones shall be ejected into the reject area.

First, familiarize yourself with the appended schematic diagram with inputs and outputs. You may also want to familiarize yourself with the following Siemens ladder elements that will be needed for this assignment: extended pulse timer, on delay timer, counter (up and down). A “dummy” contact of F.001 can be used for the end of a rung with no output (as with a timer or counter). Timer and counters (i.e. T1, C8) can be used as logic contacts in ladder logic programming.

Part Assembly

Activate the chain and belt conveyors. Using the two sensors in the sort area (metal peg detect I20.0 and sort area detect I20.1), activate the sort solenoid to have the rings go to the ~~assembly hopper~~ assembly hopper queue (rings need to wait in the hopper queue to get into assembly hopper). A component (either a metal peg or a plastic ring) passing through the sorting area may produce multiple pulses from the sort area detect sensor. Introduce necessary coding into your program so that further pulses generated by the sensor will be ignored. Increment a queue counter when a ring is sent to the assembly hopper. When the ring hopper is empty and there is a ring in the queue, activate the rotary solenoid to dispense a ring into the hopper and decrement the queue counter.

<u>Procedure</u>

1. Make the chain and belt start when the start button is pressed and keep them running until the stop button is pressed. Note that the stop button is a normally closed switch, i.e., I20.7=1 when not pressed. Latching logic will need to be utilized.
2. Sense whether a component at the sorting area on the chain conveyor is a plastic ring (not a metal peg) by utilizing I20.0 and I20.1. Note that I20.0 is a bit ahead of I20.1. So for a metal peg, I20.0 will turn down if the metal peg leaves its range while I20.1 is still high. This may cause unexpected activations of the sorting solenoid. *Hint: prevent unexpected activations by utilizing a latching logic for I20.0*.
3. If the component is a plastic ring and the assembly hopper queue is not full (number of rings in the queue is less than 5), activate the sorting solenoid for 0.50 seconds for the ring to be dispensed into the ~~assembly hopper~~ assembly hopper queue. A 3 second timer (referred to as the component pass delay) should be employed to ensure that the sensed component has cleared the area. This should prevent multiple activations of the solenoid for the same component. *Hint: use an extended pulse timer for both timers.*
4. Increment a counter (use C8) that indicates the number of rings in the assembly hopper queue when the sort solenoid is activated.
5. If there is at least one ring in the assembly hopper queue and the assembly hopper is empty, initiate a 2 second timer (referred to as dispense delay) to give a delay before activating the rotary solenoid. This gives a delay for the rings in the queue to settle against the index cam and for an assembled component to clear the assembly hopper.
6. When the 2 second dispense delay is complete, activate the rotary solenoid to index a ring into the assembly hopper.

7. When the rotary solenoid is activated, initiate a 0.5 second timer (referred to as rotary solenoid on delay) to give a delay before de-activating the solenoid. *Hint: use an on delay timer for both timers.*
8. When activation of the rotary solenoid is complete, decrement the assembly hopper queue counter.
9. Comment on any mechanical variation that kept your program from working perfectly. What measures (mechanical or with additional sensors or programming) could you take to prevent production problems.

Part Inspection

If both a metal peg and a plastic ring are detected, the parts are assembled correctly and should not be rejected. Otherwise, when the part reaches the reject area the reject solenoid should be activated to reject the part. Note: The information on whether a part is assembled correctly will need to be latched until it reaches the reject area. When the part has left the reject area, this information will need to be cleared so the next part can be processed.

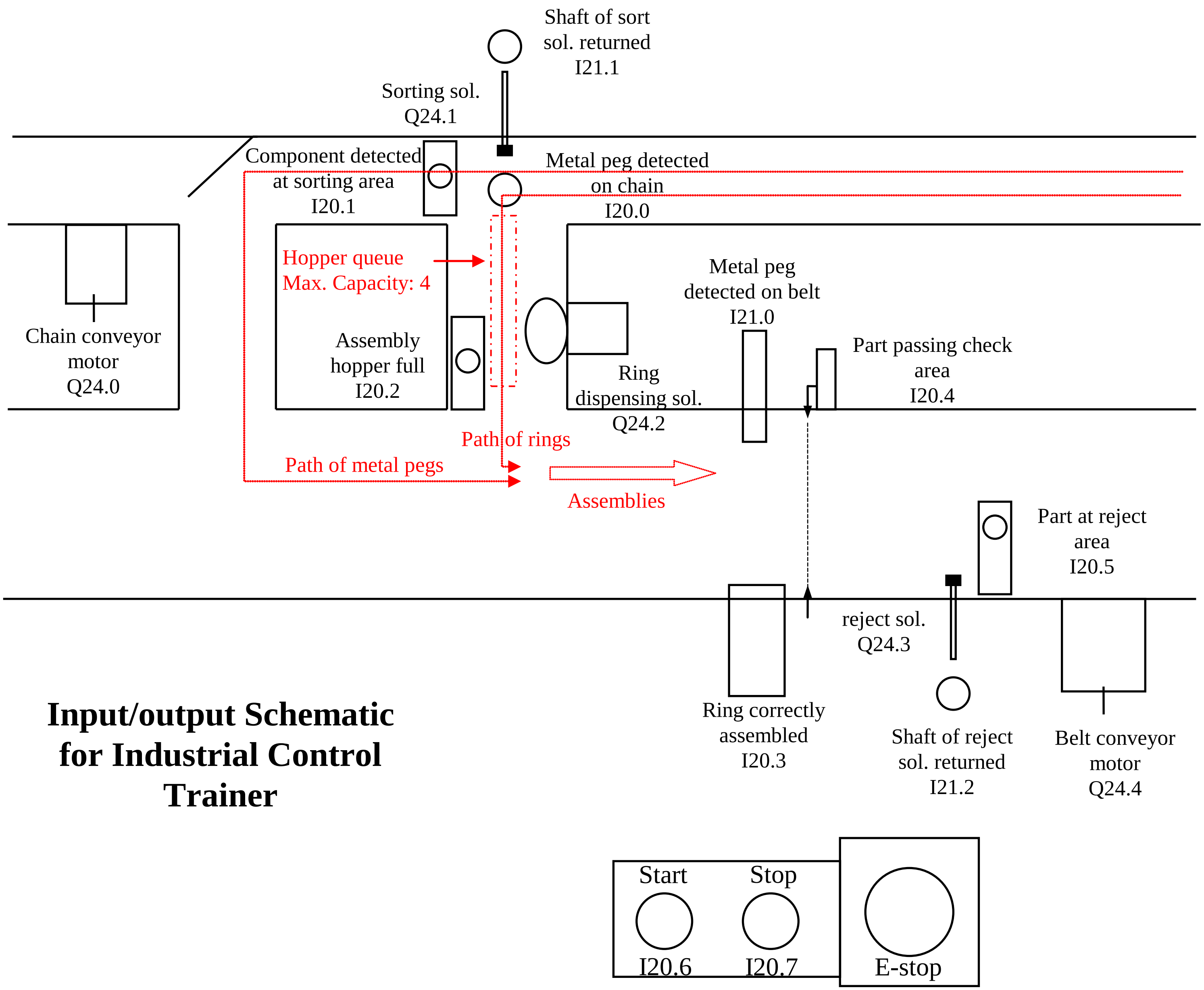
For this part of the lab, latching contacts (or flags) will need to be utilized to hold information about whether parts are on the belt, assembled correctly, etc. For these, use F62.2 - F64.7 as the contacts. A reset flag will also be needed to reset all the flags when the part is off the belt. Pay close attention to whether the sensors are normally open or closed.

<u>Procedure</u>

1. Latch the metal peg detected flag if a metal peg is detected on the belt. It should unlatch when the part is off the belt.
2. Latch a ring correctly assembled flag if a plastic ring is detected on the belt. It should unlatch when the part is off the belt.
3. Indicate that the part is assembled properly with a latched flag when both of the above flags are set. It should unlatch when the part is off the belt.
4. When a part (an assembly or an unassembled peg or ring) is sensed at the reject area, set a flag that is latched until the part is off the belt. As the part at reject area sensor (I20.5) is broken, use the part passing check area sensor (20.4) and a latched on-delay timer (4.3~4.5s delay) to set the flag.
5. When the part is detected at the reject area and the part is not assembled correctly, activate the reject solenoid for 0.5 seconds to push the part off the belt conveyor and into the reject bin. Hint: use an extended pulse timer.
6. When the part has left the reject area, set a flag (part off belt) to unlatch the metal peg detect, ring correctly assembled and part at the reject area flags. As I20.5 is broken, use the part passing check area sensor (20.4) and a latched on-delay timer (5s delay) to indicate that the part is off belt.
7. Are there any limitations that are put on the process by the program? How would you correct for these with either programming or additional hardware?

Once the program is complete, demonstrate the operation of the conveyor to the TA and print out the program.

The report must include the printouts of the ladder-logic diagram, the corresponding S5 STL program, and your comments on your logic and the use of timers and counters. It is recommended to declare the functions of flags clearly in the comments whenever you use flags in a segment to make your program more readable.



Input/output Schematic for Industrial Control Trainer

Input (to PLC)

Address used in trainer manual	**Address on Siemens PLC**	**Description**	**Pin**	**color**
32.0	I20.0	metal peg detected on the chain conveyor	P8	yellow
32.1	I20.1	component detected at the sorting area	P15	white
32.2	I20.2	assembly hopper full	P7	green
32.3	I20.3	ring correctly assembled	P14	purple
32.4	I20.4	part passing assembly checking area	P6	pink
32.5	I20.5	part at rejecting area	P13	cyan
32.6	I20.6	start switch (normally open - 0 volts)	P5	grey
32.7	I20.7	stop switch (normally closed - 24 volts)	P12	red/blue
33.0	I21.0	metal peg detected on the belt conveyor	P4	green/red
33.1	I21.1	shaft of the sorting solenoid returned	P11	yellow/red
33.2	I21.2	shaft of the reject solenoid returned	P3	orange

Normally Closed Inputs
The following is 24 volts when not depressed:
I20.7 - stop

The following are 24 volts when the shafts are back:
I21.1 - shaft of the sorting solenoid returned
I21.2 -shaft of the reject solenoid returned

The following are 24 volts when there is no component present:
I20.3 - ring correctly assembled
I20.4 - part passing assembly check area
I21.0 - metal peg detected on belt

Normally Open Inputs
The following is 0 volts when not depressed:
I20.6 - start

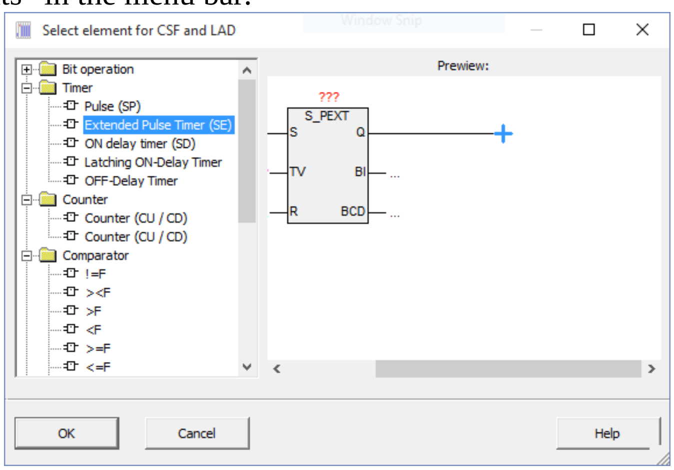
The following are 0 volts when there is no component present
I20.0 - metal peg detected on chain conveyor
I20.1 - component detected at sorting area
I20.2 - assembly hopper full
I20.5 - component at reject area

Output (from PLC)

Address used in trainer manual	Address on Siemens PLC	Description	Pin	color
32.0	Q24.0	chain conveyor motor	S8	yellow
32.1	Q24.1	sorting solenoid	S15	white
32.2	Q24.2	ring dispensing solenoid	S7	green
32.3	Q24.3	reject solenoid	S14	purple
32.4	Q24.4	belt conveyor motor	S6	pink

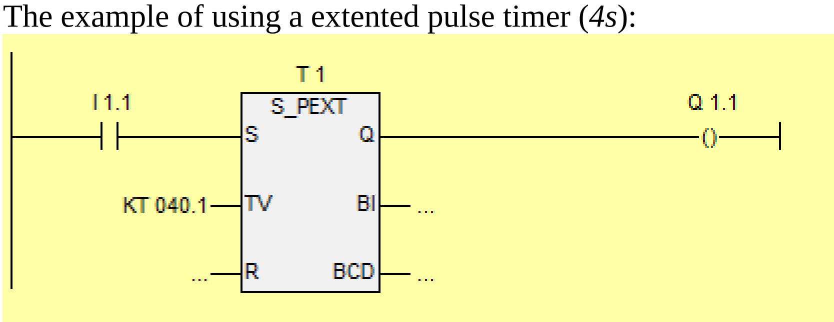
Timer, Counter and Comparator

To insert a timer, counter or comparator into your program, click “insert→select elements” in the menu bar.

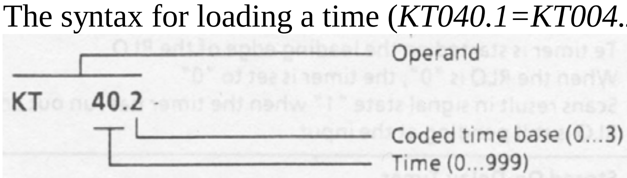


Timer:

The example of using a extented pulse timer (*4s*):



The syntax for loading a time (*KT040.1=KT004.2=4s*):



Key for Time Base:

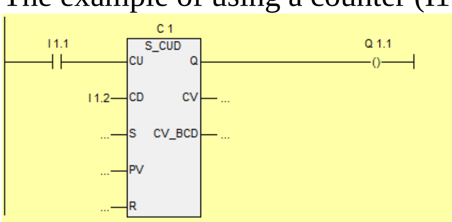
Base	0	1	2	3
Factor	0.01 sec.	0.1 sec.	1 sec.	10 sec.

Overview of different types of timers (***RLO*** refers to ***S*** and Scan results refers to ***Q***):

Table 7-4. Overview of Timer Operations

Operation	Operand		Meaning
SP	□	□	Pulse Timer The timer is started on the leading edge of the RLO. When the RLO is "0", the timer is set to "0". Scans result in signal state "1" as long as the timer is running.
SE	□	□	Extended Pulse Timer The timer is started on the leading edge of the RLO. When the RLO is "0," the timer is not affected. Scans result in signal state "1" as long as the timer is running.
~~SR~~ SD	□	□	On-Delay Timer Te timer is started on the leading edge of the RLO. When the RLO is "0", the timer is set to "0". Scans result in signal state "1" when the timer has run out and the RLO is still pending at the input.
SS	□	□	Stored On-Delay Timer The timer is started on the leading edge of the RLO. When the RLO is "0," the timer is not affected. Scans result in signal state "1" when the timer has run out. The signal state becomes "0" when the timer is reset with the "R" operation.
SF	□	□	Off-Delay Timer The timer is started on the trailing edge of the RLO. When the RLO is "1", the timer is set to its initial value. Scans result in signal state "1" as long as the RLO at the input is "1" or the timer is still running.

The example of using a counter (I1.1=1 to count up and I1.2=1 to count down):



The example of using a comparator (value of couter C1 <= 4 → Q1.2 is ture):