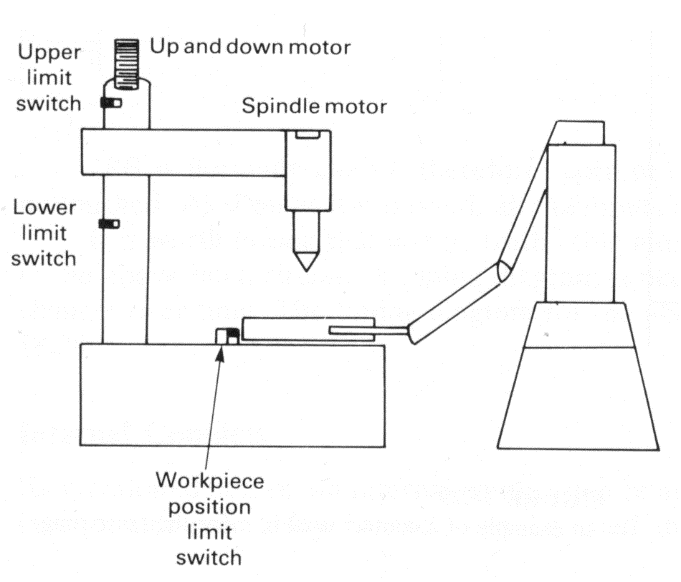
**ME 576 Lab Assignment Session 8**

This week's laboratory experiment involves programming of PLCs. Generate PLC programs for two examples covered in the class: i) drilling operation and ii) conveyor line with bin bags.

For the first case, the following functions need to be added. After drilling is completed (lower limit switch E7 is touched), there will be 2 seconds dwelling time before the drill begins to retract. Also, when the drill arm finishes retracting (upper limit switch E8 is touched), the spindle motor must stop. Assume that the spindle motor is in operation only while the output to A1 is high.

In both cases, make the PLC programs more comprehensive by adding more functions to make the operations more robust.

Generate a truth table for each case and verify the table with the simulation results on the PLC. The report must include ladder diagrams, truth tables and programs. Use the following input and output memories:



A2(up), A3(down)

**Problem 1:**

E7 Lower Limit Switch 1

A1

E8 Lower Limit Switch 2

E9 Lower Limit Switch 3

EA Lower Limit Switch 4

EA

A0 Part Loading

A1 Spindle Motor

A2 Up Motor

A3 Down Motor

A4 Part Unloading

E7

A0

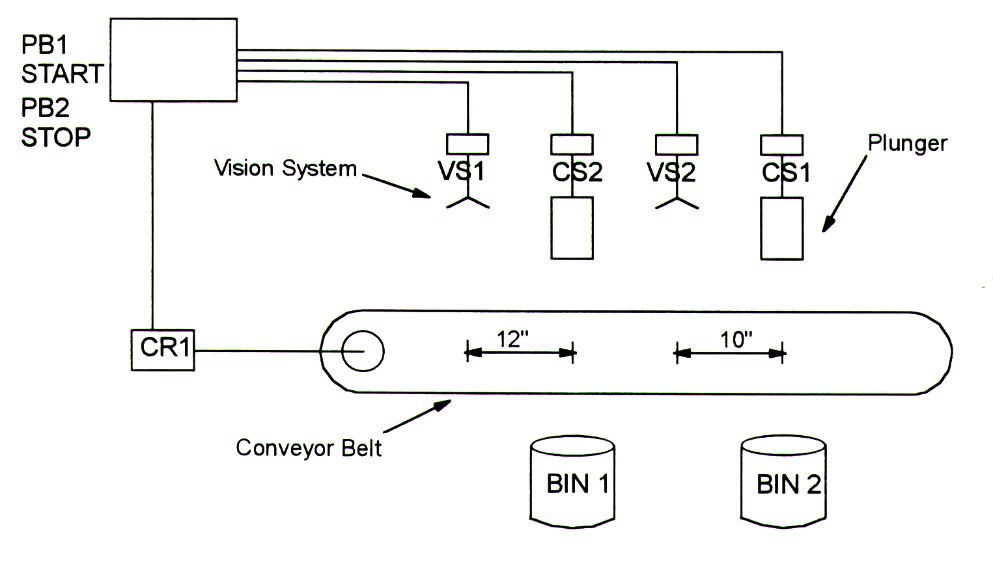
part loading

A4

part

unloading

**Problem 2:**



E0 Start (PB1)

E1 Stop (PB2)

E2 Visual System VS1

E3 Visual System VS2

E4 Bin 1 Overfill Sensor

E5 Bin 2 Overfill Sensor

A0 Conveyor Motor

A0 Plunger CS1

A1 Plunger CS2

Note: The following operators are helpful for this lab assignment:

**CAL**

IEC Operator: Calling a function block or a program

Use CAL in **IL** to call up a function block instance. The variables that will serve as the input variables are placed in parentheses right after the name of the function block instance.

Example:

Calling up the instance Inst from a function block where input variables Par1 and Par2 are 0 and TRUE respectively.

CAL INST(PAR1 := 0, PAR2 := TRUE)

**TON**

Timer function block, implements a turn-on delay. When the input gets TRUE, first a certain time will run through until also the output gets TRUE.

**Inputs:**

IN : BOOL; Rising edge starts counting up ET.

PT : TIME; Upper limit for counting up ET (delay time).

**Outputs**:

Q : BOOL; Gets a rising edge as soon as ET has reached the upper limit PV (delay time is over).

ET : current state of delay time.

Example in IL:

