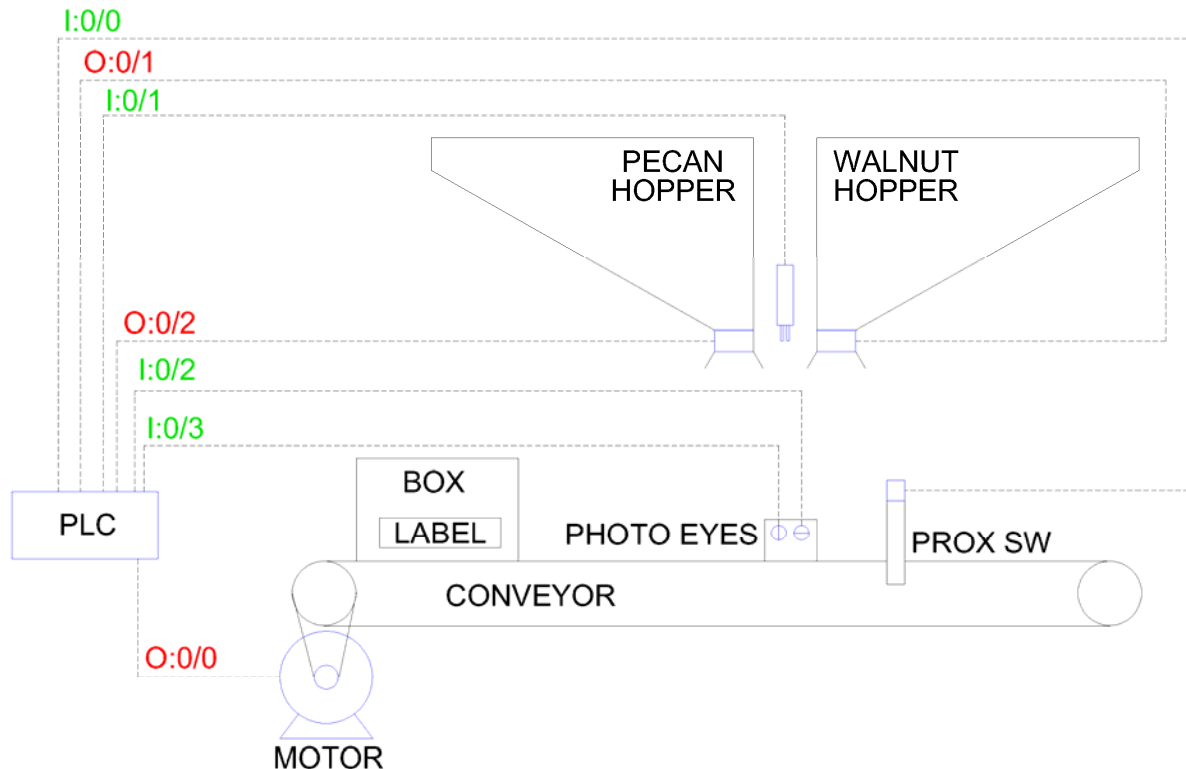


# Project 2

## PROCESS:



## SUMMARY:

A conveyor belt carries boxes with colored labels to our filling station and beyond. The proximity switch closes when a box arrives, and either a red or blue photo eye tells us which label is on the box. Red labeled boxes get filled with pecans and blue labels are for walnuts. A level sensor tells us when the box is full and ready to send along.

## IO / ASSIGNED MEMORY:

- I:0/0 - Proximity switch (closes when a box is near)
- I:0/1 - Level sensor (closes when the box is full)
- I:0/2 - Red photo eye (closes when a red label is in front of it)
- I:0/3 - Blue photo eye (closes when a blue label is in front of it)
- O:0/0 - Conveyor motor (makes the conveyor move forward when closed)
- O:0/1 - Walnut hopper (when closed, solenoid opens allowing contents to fall from the hopper)
- O:0/2 - Pecan hopper (when closed, solenoid opens allowing contents to fall from the hopper)

## TEST CRITERIA:

To start, run your program on Emulate. The conveyor motor should start immediately but both hoppers should be off.

Next, force only the proximity switch on (closed). The conveyor motor should shut off, and both hoppers should remain deenergized.

Third, leave the proximity switch closed and force the red photo eye on as well. The conveyor motor and the walnut hopper should remain off, but the pecan hopper should energize.

Fourth, leave the proximity switch closed and force the red photo eye off and the blue photo eye on. The conveyor motor should remain off, but the pecan hopper should deenergize and the walnut hopper should energize.

Next step, force the level sensor on. The hoppers should both deenergize and the conveyor should start back up to move the box forward.

Finally, force the proximity switch, the level sensor and both photo eyes off. Both hoppers should remain deenergized and the conveyor should keep running.

Bonus test: with the proximity switch and level sensor deenergized, force both photo eyes on. Neither hopper should energize. (We don't want to release product when there isn't a box to catch it in.)

## NOTES:

There is a lot of IO in this scenario, but it's all digital, so this isn't QUITE as hard as it looks, but it's still a solid challenge. The trick is to envision this process as a cycle and manage it accordingly in your logic.

Ask yourself "When does the pecan hopper open?" "When does it need to close?" "When can I start the conveyor?" "What makes it stop?" What you don't want to happen is for the box to run away and you pour a bunch of pecans all over the floor. You also don't want to overfill the box or fill it and make it just sit there forever. Your program needs to keep everything working continuously – nothing wasted, and no delays.

There's one place where I expect everybody to get stuck on this. You'll fill your box and ask, "Now how do I start the conveyor again?" When you get into the testing, you'll see what I mean. 😊

Okay, make it work. The people want their NUTS!