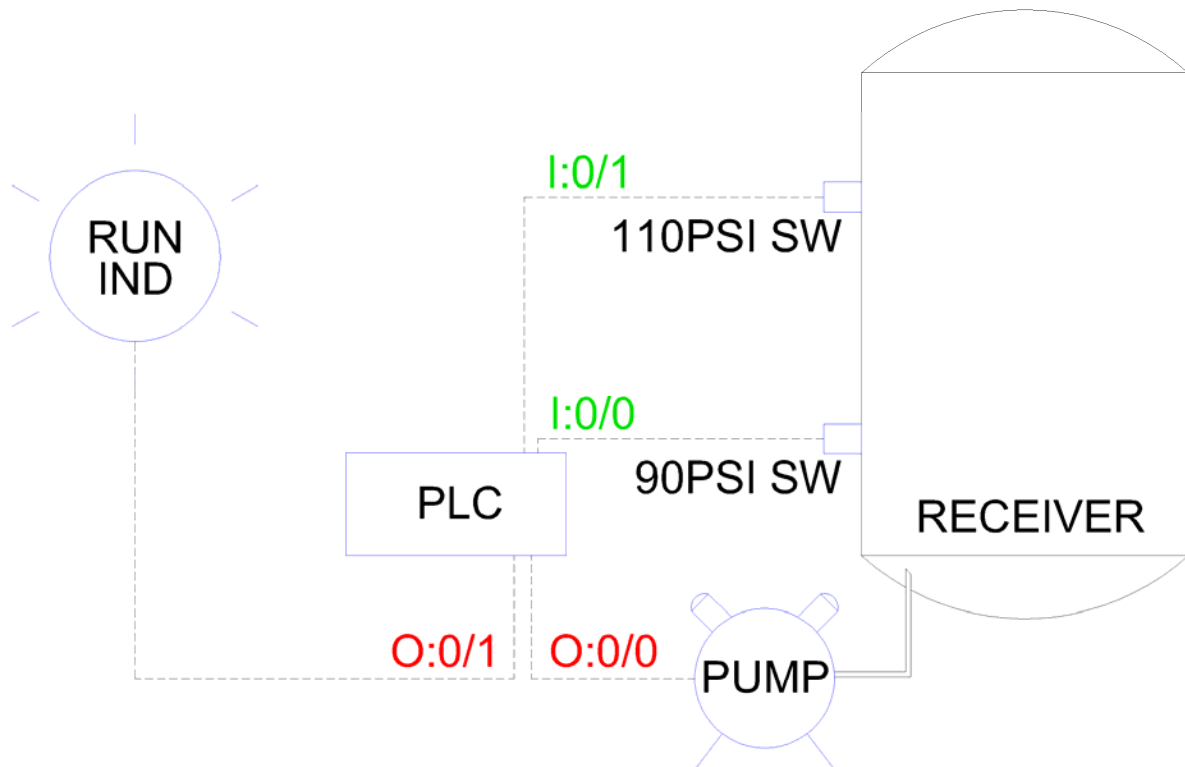


Project 1

PROCESS:



SUMMARY:

Today, we're going to be maintaining the pressure in a receiver on a compressor application. There are two pressure switches which close at 90 and 110psi (low and high). To control the pressure, we have one pump. Additionally, we want to illuminate an indicator light when the pressure is above 90psi.

IO / ASSIGNED MEMORY:

I:0/0 - Low pressure switch (closes at 90psi and above)
I:0/1 - High pressure switch (closes at 110psi and above)
O:0/0 – Pressure pump
O:0/1 – Pressure indicator light

TEST CRITERIA:

To start, run your program on Emulate. The pump should start immediately but the light should be off.

Next, force only the low pressure switch on (closed). The pump should remain energized and the light should now also energize.

Third, leave the low pressure switch closed and force the high pressure switch on as well. The pump should deenergize and the light should remain energized.

Fourth, leave the low pressure switch forced on and force off the high pressure switch. The pump should remain deenergized and the light should remain on.

Lastly, force both pressure switches off and verify that the pump starts back up and the light goes off.

NOTES:

This is the simplest application we'll be doing in this course. ALL of these should make you think a bit and may take some trial and error to get it working right. THAT'S OKAY! That's how real programming works. Try to create your own program. It doesn't HAVE to look just like mine – it just has to pass the test criteria. Remember that programmers all have different styles and approaches, and that's totally expected – there is always more than one "right" way to do something. What's important is what your client sees: a program that does what it's supposed to do.

When I approach these projects in the coming lectures, I won't be creating 'rehearsed' programs. You'll no doubt see me make a few mistakes which I won't catch until I try to emulate my program and realize it doesn't work as desired. This isn't because I'm a terrible programmer or because I was too lazy to go back and re-record my videos. ☺ The point here is that you are going to watch me create programs and approach problems the way I do every day. All programs have a bug or two in them. This is why it's important to TEST TEST TEST your work before you hand it over to your clients or run them on live machines.

The time to feel foolish and make mistakes is not when people are watching. Do that stuff alone, at your desk. The only thing you want the rest of the world to see is a PLC HERO!