

Project 3

SUMMARY:

Our department manages three raw materials: “widgets,” “doodads” and “wockies.” These all have part numbers in our system of 123, 456, and 789 respectively. People will use a barcode scanner which will write the following string to our PLC: “PPP-QQ:D” (without quotes) which will contain part number (PPP), quantity (QQ) and direction (1 - incoming, 2 - outgoing). Store the on-hand quantity of each.

IO / ASSIGNED MEMORY:

ST9:0 - Barcode scan into PLC from barcode scanner

N7:0 - Part #123 quantity on-hand

N7:1 - Part #456 quantity on-hand

N7:2 - Part #789 quantity on-hand

TEST CRITERIA:

To start, run your program on Emulate. Nothing should be happening. Check N7:0, N7:1 and N7:2 and make sure all three are equal to 0.

Next, change the value of ST9:0 to “123-10:1” without quotes. Then check the N7 data file and N7:0 should be equal to 10. N7:1 and N7:2 should be 0 still. The value of ST9:0 should be automatically reset to a blank / empty / null / default value IMMEDIATELY after each scan (or each time we manually set the value).

Third, set the value of ST9:0 to “123-10:1” AGAIN. You shouldn’t have to manually change the value to anything else between last step and this test. Then check the N7 data file and N7:0 should be equal to 20. N7:1 and N7:2 should be 0 still.

Fourth, set the value of ST9:0 to “456-15:1”. Then check the N7 data file and N7:0 should be equal to 20. N7:1 should be 15 and N7:2 should be 0 still.

Next step, set the value of ST9:0 to “789-30:1”. Then check the N7 data file and N7:0 should be equal to 20. N7:1 should be 15 and N7:2 should be 30. Now we know that incoming scans work!

Now, set the value of ST9:0 to “123-19:2”. Then, immediately after, set it to “456-14:2”. Then, change it once more to “789-29:2”. When we check the N7 data file and N7:0, N7:1 and N7:2 should all be equal to 1. Now we know that outgoing scans work!

Bonus test: set the value of ST9:0 to “149-1:1”. Nothing should change in our N7 data file, and we shouldn’t have any errors / interruptions in the flow of the program. It should simply ignore the unrecognized part number.

NOTES:

There is no machine, there are no outputs, and we don't even have a traditional input. What the hell?!

Sometimes, clients will want / need to use a PLC for something that could and often SHOULD actually be done on a computer. And that's fine by us – more money for you and me! This project is based on a project I've actually done which was sure enough – a PLC-based inventory control system. To this day I don't know why they wanted to use a PLC, but they made several similar systems, so there's obviously a demand for it.

So in this application, a barcode scanner is going to write a string right into our data file. How does the PLC know when that has happened? How can we work with that string and pull out the information we need?

And what's a wockie anyway? Is that really a thing, or did you just make it up? 😊