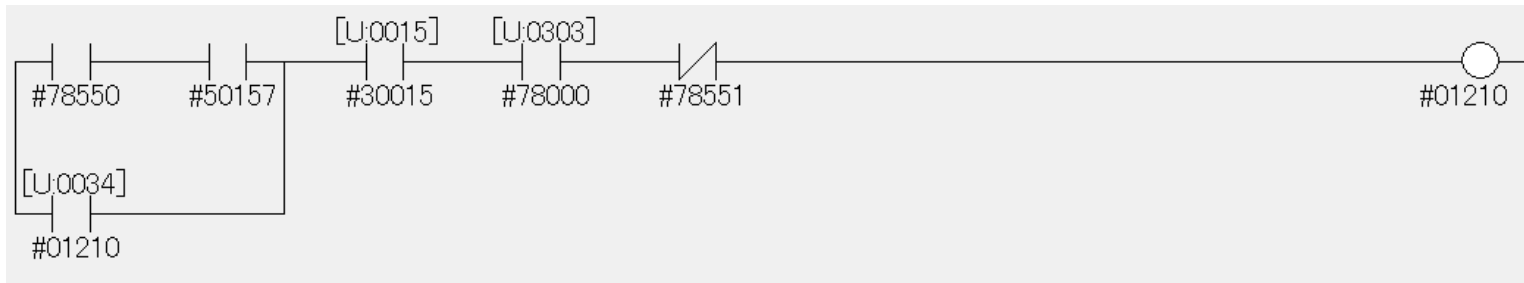


Implementing a Seal in Concurrent IO

This document explains how to implement seal logic in our concurrent IO program. A seal allows you to keep an output energized until the logic for that output becomes untrue. Below is an example of sealing in a universal input for a part ready to be processed.



```
STR #78550      // Side A Part Ready for Processing Button
STR #50157      // Robot at Safe Position for Processing
STR #01210      //IN #961 - Side A Part Ready for Processing SEAL
OR-STR
AND #30015      //Robot in Remote Mode
AND #78000      //Estops Ok
AND-NOT #78551 //Side B Not Ready for Processing
OUT #01210      //IN #961 - Side A Part Ready for Processing
```

The way this works: when the robots status is at safe positioning for processing “Interference Cube 64 #50157”, robot in remote mode “Dedicated External Output 30015”, robot is clear of any ESTOPS “Auxiliary relay 78000” and Side B not ready for processing “Auxiliary Relay 78551”. Once the above status is true and the Side A Part Ready for Processing Button is pressed “Auxiliary Relay 78550” this will then seal the output coil Side A Part Ready for Processing “Universal Input 961 #01210”. This will remain on until one of the above conditions become untrue.