

Lessons Learned

- Parts not fitting well together, gap was too big. Making the gap width smaller let us use one O-Ring, which decreased the friction, making it easier for the robot to assemble the parts.
- Pusher was not pushing the object straight down like we wanted. Using the Approach/Depart commands, the robot would go diagonally rather than vertically. The robots use the Z-axis of the 6th joint with those commands, so with our design, this could not work. Instead we used the Move command and used one location above the assembly and the other when it was assembled.
- The springs we bought were not stiff enough to give enough pressure for the push finger. Using a stiffer spring, the finger could easier press down on the parts.
- We needed to find a position above the assembly area so that the robot would not hit any of the parts. If we did not have this, the robot would slide over at risk of breaking our end effector or damaging the robot.
- The gripper will not work without the use of the airline and ensuring the power is on.
- IP protocols only work if you make one robot the server and one the client. Using these settings attached to a port, you can code the robots to send a signal to each other.
- Delimiter errors were fixed by changing the delimiter setting on the controller for the server. Changing this from CRLF to CR enabled the code to work.
- In order to transmit the code to the controller, Ethernet cables must be plugged in to each as well as the computer. Using WINCAPS and entering the IP addresses manually, the computer is able to connect to each robot.
- Each robot has to have its own code. We cannot use the same code for both.
- You can only run the code when the controller is on AUTO mode.
- To connect the Arduino via Ethernet, a computer needs to be wired to the Arduino in order to tell whether there is IP connection or not.
- In order to separate the male and female parts, we have to use an angled “ceiling” on the top of the ramp to mechanically separate the two parts.
- The male will have to be grabbed first. Due to the orientation of the tool on Robot 1, it cannot be flipped. This means the tool on Robot 2 needs to have a larger diameter to push the female onto the male.
- The robot cannot send a signal to the Arduino, so the scanner must be implemented using Robot 2. This will send the signal to the Arduino in order for the conveyor belt to move.

- The conveyor belt “zipper” struggles at each of the bearings. In order for it to move through that obstacle, a slower speed must be used as well as the highest torque allowed by the controller.
- The conveyor belt moves to one side of the bearings, at times moving over the top of the depth which was turned out of the bearing. A further depth was required to ensure the belt stays on the track.