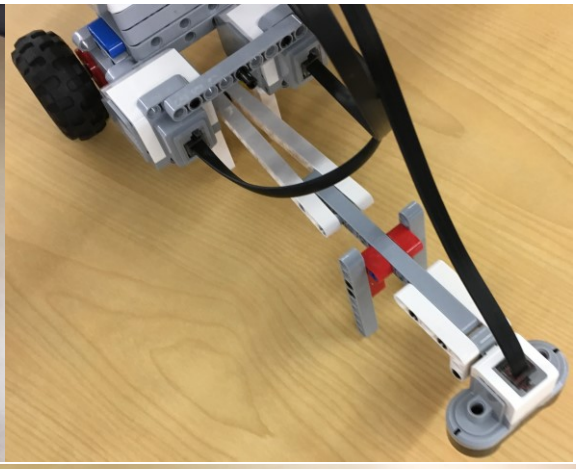
LabView ICA 16 – Write-up

1. Sensor Calibration

We calibrated the ultrasonic sensor for the EV3 by writing a simple VI that output the distance (which we predetermined to be cm) between the sensor and the next object to a numeric indicator. Then, while the robot was tethered to the computer, we moved the robot from the center of the table to the edge and recorded the safest threshold. We originally had the threshold set to 6 cm, but with changes in the design of the robot, the final threshold ended up being 3.8 cm.

2. Robot build

I'm not sure exactly what you're looking for, but here are some pictures of our robot. Hopefully that's what you're asking ☺

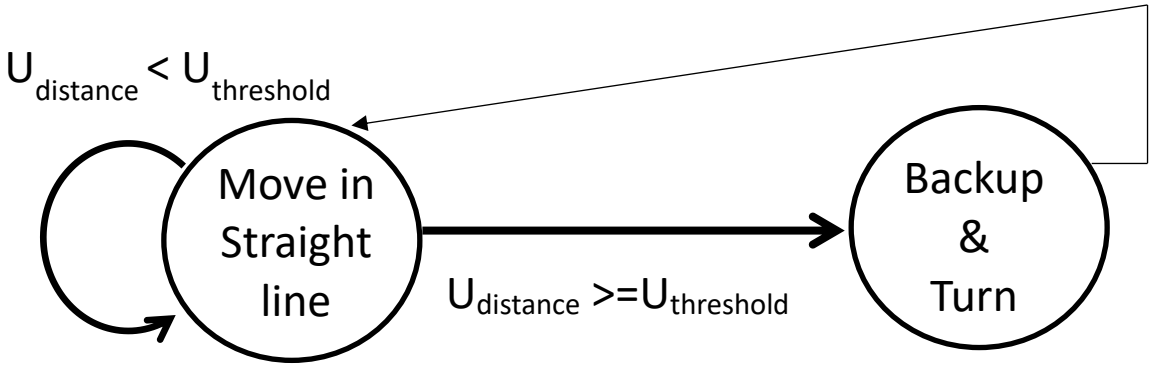


This is the final iteration of the robot. The EV3 was mounted to the motors using beams that were connecting using non-rotating connector pegs. The motors were connected to the layer of beams using the 5x11 module. The motors were connected to Port A and B. The ultrasonic sensor was connected to the motor using an axle with bushings everywhere. We used a beam to prop the robot up in 2 places – the middle of the ultrasonic sensor (so the sensor wouldn't be flush with the table), and the base of the motor (so the motor wouldn't be dragged along; we could have

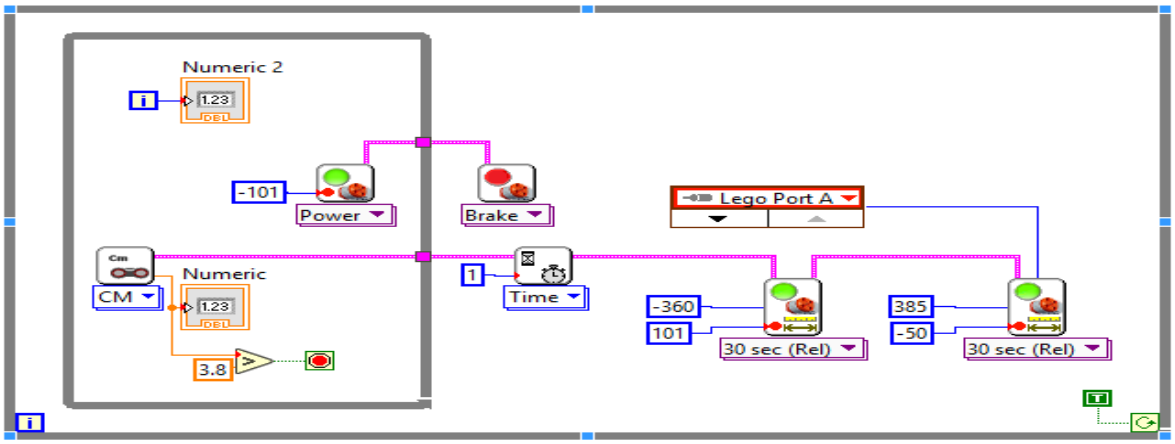
used a wheel, but we were low on time and hadn't planned on using one originally). The ultrasonic sensor is connected to the robot using a variety of rods of the maximum possible length to increase the distance between the wheels and the table – that way when the ultrasonic sensor approaches the edge, the robot will have enough time to back up and turn, as well as the robot having enough time to stop if it approaches the edge at a very acute angle. The ultrasonic sensor was connected to Port 1

3. Robot States & State Diagram

\We decided that the robot should have 2 states – moving, and adjusting. The moving stage was when the robot moved in a straight line, and the adjusting stage was when the ultrasonic sensor reached the threshold and reversed as well as turned another direction. We decided that we wouldn't need to check if the sensor was over the table after the backup and turn because the consecutive actions made it highly unlikely to happen.



4. VI – Image



I'm assuming you want an explanation for what I did, so here it goes. Everything is encased In a while true loop, since EV3 doesn't have a run continuously option. As in the state diagram, there are 2 main parts – the moving straight, and the adjusting. The moving straight part is a while false loop – while the threshold of the ultrasonic sensor (3.8 cm) is bigger than the current reading of the

ultrasonic sensor, send power to the motor ports (which are A and B, but automatically configured by LabVIEW). Once it breaks out of the loop, we apply the brake, and wait for 1 second (this ensures that the robot is fully stopped and gives us a chance to grab it if something unfortunate happens. Then, move the motors forward (because of the change in design everything is inverse) 1 rotation and move only the motor plugged into Port A (it doesn't matter which one, it really just needs to turn) at half power 385 degrees (which is equivalent to the entire robot turning around 90 degrees). Repeat until stopped.

5. Modifications & the Design Process

Our initial design looked a lot like the final design. However, the motors were mounted along the sides of the NXT, and the ultrasonic sensor was mounted closer to the source. Due to the original distance between the robot and the sensor, the robot was susceptible to fall if the angle between the robot and the table was extremely acute (~ < 30 degrees). We tried to resolve the bug by angling the sensor (thus theoretically increasing the distance the robot stops), but the readings were all over the place and the problem was still not fully resolved. Our final test was increasing the distance between the sensor and the back wheel. We decided to redesign the base in the middle to increase stability, and ended up flipping the motors.