Weekly Report - Tuesday, April 16, 2019

Auburn University IEEE SoutheastCon 2019 Hardware Competition Team

---

Tasks Remaining

Task Name	Assignee
Compile Notebook	Nia
Document Electrical Hardware	Joe
Document Software	Josh/Nia/Matthew
Document Mechanical Hardware	Matthew/Joe
Move Software from private Gitlab to SPARC's public Github	Matthew
Poster	Matthew
Cleanup and Capture ROS Graph	Josh
Create a highlight video compiling photos and videos	Joe
Create final report	Full Team
Create final presentation	Full Team
Put together a live demonstration for the senior design fair	Full Team

Achievements, Obstacles, and Risks

Unfortunately, we were unable to get the robot to work properly as we would have liked. We were really close but we were unable to compete with it. There were many late nights spent attempting to get it to work. We ended up scrambling together an Arduino program in the last 20 minutes before competing.

The robot scored 5 points for leaving the home base, 10 points for returning, and 25 points for raising a flag for a total of 40 points. Between the two rounds, half of us kept trying to use ROS while the rest of the team strapped the broken test bot back together and create a higher scoring Arduino program that could score points by making circles around the center structure. The ROS group ended up almost getting circles around the center to work. When it came time to compete, we had to load the Arduino program. However, we never tested it on the main bot before loading it and competing. It didn't work quite the same as the test platform but managed to score points and almost make it back to the home base for a score of 40 points again. Our total points scored was 80 points placing us 16th in the competition against 40 other teams. All in all, we competed, we had a robot that moved, scored points, and we learned a ton about robotics and ROS. Several of us are going to keep working on the project for the remainder of the semester and potentially in the summer to see if we can make a proper ROS operated robot. The results of the competition can be found here: https://docs.google.com/spreadsheets/d/1DqehVuSWq2uOKsn9gIgF807Y_UsugKH4_x3prgCqvh8/edit#gid=165236288

What Went Well

- The mechanical platform was rock solid. It was very sturdy. The design was fairly adaptable even with cramming large amounts of additional electronics. The motors were

a good choice as they were very powerful and had a lot of potential. The 3D printed parts were fairly sturdy despite our fears of parts constantly breaking. The only major part to break was the gate flap when we hit a spacetel in testing. The 8020 frame was a great decision due to its modularity which came in very useful when we added additional electrical components.

- The electrical hardware system worked well. The motor controller was a great choice. It was powerful enough to provide the high amperage of the motors. It is not very difficult to connect it to an Arduino via serial communication. Having the main battery for the motors and an auxiliary battery for the five-volt devices was very reliable and made debugging and testing much easier.
- Building a second robot was essential. We had a full set of spare parts as well as a platform to run tests on. We also had the ability to make some very necessary changes.
- Computer vision worked great. It was fairly accurate at determining the positions of the objects with respect to the robot at a decent framerate considering the hardware utilized.

What We Learned

- ROS is very difficult but we have a working understanding of ROS. We started development with ROS only 3 months ago. If we starting using ROS over 6 months ago, we would have had a better grasp of the process of integration for electronics that require custom nodes. We also kept running into a lack of processing power issues when it was too late to replace the processor. Strapping a second Raspberry Pi on the bot to increase processing power further increased the integration complexity and caused other issues
- We overestimated the competition. Our robot was by far the most complex and most ambitious design software wise.
- Lack of sleep and ROS programming do not go very well together.
- Despite our lack of success, we learned alot more about robotics, sensors, and control systems than many of the other teams due to our ambitious goals.
- Having a back-up plan is important even if it takes a little time away from the main plan.

Approaches of Other Universities

- The University of Alabama utilized 8 IR sensors with home brew algorithm in C++. They did not implement ROS like we did but were very successful.
- Clemson had a really unique object sorting mechanism.
- The University of Tennessee Chattanooga used fans to push the balls into the corners.
- Virginia Military College used adhesive strips to collect the debris.
- We saw many teams localize using static distance sensors like IR and Ultrasonic. There were also a few teams with LIDAR's.

- We were one of the few teams to implement ROS. Many teams we talked to considered it but decided against it due to its complexity.

How can we improve if we were to compete next year

- Always create a backup bot. If you can create a robot that can score simple points with an Arduino, you will do much better in competition than a complicated robot that cannot move. You should always try to build the more complex bot in hopes that it will work better than the Arduino controlled one. Whichever one can score more points the day before the competition would compete in the main competition and the other in the open competition.
- Start simple. In terms of the competition, a simple bot that can score points is better than a complex one that cannot move.
- Create backup programs with various levels of complexity so that you have options when issues arise on competition day.