



Mars Mining Robot Team

C.E.R.E.S.

Computerized and Enhanced Robotic Excavation System



Milestone 6

- Networking - We were successful in passing the standard input to the python program controlling the arduinos. This allowed us to remotely send controls to the robot without being directly connected to the robot. We ran into a new problem with the standard output when we have the server start when we turn the Jetson TX1 on. Even though the standard output is being directed to the python program, there is no communication between the two programs. This will be fixed prior to competition. We also noticed a considerable time difference between sending the robot commands and the robot actually implementing these commands.
- Software to interact with hardware - We successfully developed a python program that interacts with the arduinos via I2C. This code, depending on the signal it receives from the server, will tell which arduino to do either: turning of the bot, forward movement, reverse movement, stop, lift the bin up/down, or spin the auger (the digging tool). The startup of the robot can cause a little too much torque from all the power given to the machine. Therefore, we need to test and modify each of the controls so that operating the machine runs more smoothly.
- Autonomous Functionality - Now that we have a functioning robot and can enter the competition, we have been focusing a little more on the autonomy of the robot. We still have a lot to do before we head to competition in May. We have been working on implementing code that will control the robot autonomously. We are looking into using python, java, C/C++, and bash in order to accomplish this task. We will then a fix cameras onto the vehicle in order for the robot to monitor its movement.

Milestone 6

Tasks	Completion	Catherine	Nicholas
Finish implementing and testing Networking functions	90%	90%	10%
Finish implementing and testing Software that controls hardware	90%	30%	70%
Continue/finish design, implement, and test autonomous functionality	30%	50%	50%

Lessons Learned

- **Do not always trust the documentation for software. Sometimes it is wrong.**
- **Sometimes the difference in sending/receiving time for a network can be a large problem.**
- **Sometimes the best way to learn a lesson is to devote a large amount of time and effort into something, and ultimately, find out it was not a good idea or have it fail.**
- **Do not underestimate something that at first glance seems easy and quick. It might actually turn out to be time consuming and/or difficult.**
- **Do not always trust your teammates to complete what they need to do.**

