NASA Rock Mining Competition Software Team

Catherine Grover - cgrover2014@my.fit.edu
Nicholas Persing - npersing2015@my.fit.edu

Faculty Advisor - Eraldo Ribeiro - eribeiro@cs.fit.edu

Progress of current Milestone (progress matrix)

Tasks	Completion	Catherine	Nicholas	To Do
Finish implementing and testing Networking functions	90%	90%	10%	Find a way for the server that begins on startup to send output to the python code that controls the robot's movements. Lower the amount of "lag" between a command being sent and the robot receiving the message.
Finish implementing and testing Software that controls hardware	90%	30%	70%	Refine the controls of the robot allowing ease of control for the person controlling the robot (or autonomously)
Continue/finish design, implement, and test autonomous functionality	30%	50%	50%	Implement sensors to monitor the robot's surroundings and use this information to develop programs to

				control the robot autonomously.
Develop a User Manual	100%	50%	50%	N/A
Develop a Demo Video	100%	50%	50%	N/A

Discussion of each accomplished task (and obstacles) for the current Milestone:

- 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.
- Develop a User Manual - A user manual was developed so that someone could easily operate the robot.
- Develop a Demo Video - The demo video was completed by the whole NASA Robotic Mining Team.

Discussion (at least a few sentences, ie a paragraph) of contribution of each team member to the current Milestone:

- Catherine Grover - Catherine has been working on the network for the robot. She is currently trying to minimize the time difference between sending the robot a

command and the robot moving. She is also working on fixing the issue with the server sending standard output to the python program that controls the robot on start up of the machine. She has also has helped Nick with the I2C functions of the robot.

- Nicholas Persing - Nick has been working with the other team to finish up our development of the robot. When he isn't busy helping them, he has been working to fix the issues with I2C and helping Catherine with the server. Currently, he has set his sights on making the vehicle autonomous.

Lessons Learned:

- **Do not always trust the documentation for software. Sometimes it is wrong**. We learned this while attempting to implement I2C with Bash Script. The command's man page on Linux stated that the format of the command was to include a field for the location of the memory on the arduino. Instead, we found out after testing the command, that that particular field was not for the memory location, as was stated, but it was used to state what was to be sent to the arduino.
- **Sometimes the difference in sending/receiving time for a network can be a large problem.** This sounds like a pretty logical conclusion, but we did not realize that one as large as 3-4 seconds delay could be a momentous fault. We learned this at senior design show case when our robot decided to "run away" into the crowd watching it. Specifically, it aimed at the judges who were watching.
- **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.** We learned this lesson countless times throughout the project when we would have to change paths after spending hours on it, or believing something would work when it actually failed in practice.
- **Do not underestimate something that at first glance seems easy and quick. It might actually turn out to be time consuming and/or difficult.** We learned this on a few occasions when we procrastinated something that seemed easy, but ended up taking us a long time to complete. Our best example would have to be our python code. What seemed like an easy task, parse input from a java program and when it receives a certain command break out of the while loop, ended up taking us several days due to inexperience with python.
- **Do not always trust your teammates to complete what they need to do.** Thankfully, neither Nicholas or Catherine had to deal with this with each other, but we did learn this when dealing with the other engineering members of the team.

Sponsor feedback on each task for the current Milestone

Sponsor Signature: _____ Date: _____

Sponsor Evaluation

- Sponsor: detach and return this page to Dr. Chan (HC 322)
- Score (0-10) for each member: circle a score (or circle two adjacent scores for .25 or write down a real number between 0 and 10)

Catherine Grover	0	1	2	3	4	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10
Nicholas Persing	0	1	2	3	4	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10

Sponsor Signature: _____ Date: _____