**FRC Robotics 2018**

**Software Engineering Team Log**

Team Members:

AM

Abby Drane: [adrane5274@vbtc.net](mailto:adrane5274@vbtc.net)

Austin Holt: [aholt5457@vbtc.net](mailto:aholt5457@vbtc.net)

PM

Matthew Bryant: [mbryant7135@vbtc.net](mailto:mbryant7135@vbtc.net)

Gavin Buskirk: [gbuskirk5148@vbtc.net](mailto:gbuskirk5148@vbtc.net)

Anthony Conner: [aconner7202@vbtc.net](mailto:aconner7202@vbtc.net)

Noah Fraser: [nfraser5321@vbtc.net](mailto:nfraser5321@vbtc.net)

Cody Hetler: [chetler7427@vbtc.net](mailto:chetler7427@vbtc.net)

https://ssl.gstatic.com/ui/v1/icons/mail/images/cleardot.gifRobbie Phillips: [rphillips5743@vbtc.net](mailto:rphillips5743@vbtc.net)

The purpose of this document is to facilitate collaboration and communication between team members on this project.

Initially, the contents are the daily robotics-related individual work log entries from the team members. Other, relevant content from any team member can also be added here.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

12-Jan-2018

Abby:

We found out what height and configuration the wheels have to be at for it to move and make it  up the ramp with the vertical flattened out part at the bottom.. The height of the frame has to be 3in. off the ground for clearing, and the bumpers have to be 7in. off the ground to make it. We also reconfigured where the wheels go ( bottom example ) To have the ability to move up this ramp because the last one ( top example ) couldn’t move upwards on it.



Austin:

In the robotics room we worked on the frame of the bot and tested out different ways the wheels can be arranged. We also found out that the ramp the bot must go up in going to be squared off making it impossible for our first idea to work. Our new design will be able to go up the ramp with no problems and will have at least 2 inches of space to clear the top of the ramp and the bottom of the bots frame. The frame will have to be 3-3.5 inches off the ground and the bumpers must sit on 7 inches. The wheels we’re using are completely omni and they will be arranged with 2 verticals in the front and back and 2 horizontal in the middle. This will allow the bot to clear the ramp and the wheels in the middle will no longer catch the ramp. Compared to our original design we added 2 more omni wheels and changes the arrangement of the wheels.



Matthew:

Today I communicated with Mr. Kooyers about what needs to be known about the robot for the programmers.

I had also sent out a group email telling the group

- the different parts that need programed

- parts of the bot that they need to know about

-how to get github onto their computers

-where to find the game manual

-where to find the release videos

And I downloaded eclipse.

Gavin:

I have been working on Robotics; I installed C++ for Eclipse and got the toolchain supplied by FRC. I have also installed NI Vision for future usage in the same robotics stuff I have previously mentioned. I have also researched things I did not know about mechanics such as the talons and other motor-related mechanical pieces.

Anthony:

Installed NI Vision Software and assisted others in installing and the like and went over the game details with other members.

Noah:

Went to robotics and talked about the code we will be using and GitHub.

Cody:

(Absent)

Robbie:

I followed the instructions Matthew left and downloaded GitHub and Eclipse for C++.

J. M. Hinckley:

This is a good start. The clock is running, however. We are one week down and have five to go. I have been mentoring the FIRST team 1254 for many years (more that I’ve been teaching Software Engineering here). The common experience is that each year we come down to the wire and have to rush the software and don’t have enough time to really test. My advice is this:

1. Master the software. Mechanical, electrical, CAD stuff is important, but this thing doesn’t go anywhere without software. In the end, all eyes will be on you to make it work. And the pressure will be high. So, while you stay in touch with the other build teams on different aspects of the robot, prepare yourselves and concentrate on mastering the software.
2. Write down, in a file here in the repository so that everyone has access, the specifications and design decisions for the software. Don’t rely on keeping this information on paper in a notebook. Those things get misplaced and are messy. Take the time to scan documents as needed and upload them. We have 21st century tools, let’s use them. If you need to, make drawings on paper, but then scan and upload them. Take the time to do neat work—other people have to understand your information.
3. Figure out how to write a basic robot program in the specified language. It looks like you are going to use C++ this year. So, figure out, as a team, how to write the equivalent of a “Hello World” program for the RoboRIO. This is generally a tank-drive kind of program. You need to be able to at least get a robot, perhaps last year’s robot, to move using the joysticks.
4. If last year’s robot is not available for code practice, at least get a RoboRIO (there are at least two of them around) and connect it on the test panel to a motor and a pneumatic valve. You must learn how to control a motor with a joystick and to switch a pneumatic valve with one of the buttons on the joystick. Figure this out sooner rather than later. You do not want to have to be learning this in the last week of the build. As the robot design evolves, add corresponding hardware components to your test setup and test your code for the new hardware. Start writing and testing simple code as soon as possible.
5. Make every day count. You don’t have enough of them to waste any. Work efficiently. There are eight of you on the software team. Divide the research and development work, delegating specific work to specific team members. In the same way that the entire robotics team is divided into subteams (e.g. drive team, climbing team, etc.), the software team should be organized with specific people responsible for specific things, such as tool configuration, test code, test panel system, hardware interfacing, code architecture, code test and debug, etc.. Don’t duplicate each other’s work. Be clear about who is doing what. This will require organization and communication. As each of you figures out how to do something, create a written, step by step document and upload it to this repository. The team should not have to figure out how to do something more than once.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_