SYSTEM DOCUMENT

Project: ECSE 211 Final Design Project – Team 6
Document Version Number: 3.0
Creation Date: 14/10/2017
Original Author: Justin Tremblay
Edit History:
[14/10/2017] Justin Tremblay: First draft.
[15/10/2017] Justin Tremblay: Filled in most of the partsrelating to software.
[16/10/2017] Alex Hale: added TODO notes for sections we can't fill in yet; edited some writing; changed version system to WEEK.EDIT (e.g. it is currently week 1, edit 3 => 1.3)
[22/10/2017] Alex Hale: transferred to Word document for easier formatting; minor editing
[23/10/2017] Alex Hale: added electromechanical limitations
[30/10/2017] Alex Hale: updated state machine and class hierarchy diagrams

1.0 TABLE OF CONTENTS

2.0 SYSTEM MODEL

TODO: block diagram of basic functions that are needed - should gradually be updated as implications of client requirements become clearer

2.1 HARDWARE AVAILABILITY AND CAPABILITY

The hardware for this project will be the LEGO Mindstorms kit, including one EV3 brick, an assortment of motors and sensors, and a wide variety of connecting pieces for construction. Since the team is made up of three laboratory groups, up to three kits will be available, but it is unlikely that more than two EV3 bricks will be used.

The capacity of the battery is limited, and as the battery depletes, the supplied voltage drops dramatically. The lower voltage affects the performance of the motors in an unpredictable fashion, so it is very important to always test the robot with a charged battery.

The EV3 brick contains a low-power ARM processor, so our computing speed is quite limited. Our system memory is stored on a micro SD card, which is fast enough that it won't bottleneck our speed before the processor.

2.2 SOFTWARE AVAILABILITY AND CAPABILITY

This project will be written in Java, making use of the lejOS library for the LEGO Mindstorms EV3brick.

Java is a high-level object-oriented programming language with features like multi-threading and garbage collection. Its advantages are that it is easy to use and multi-platform. Its disadvantages are that, since it runs on a virtual machine, it adds a lot a overhead and makes it hard to optimize code on platforms such as the EV3, which has very limited resources. Java's memory management model is also a big disadvantage: since it uses garbage collection, it doesn't allow much control over the memory usage of the program and adds a lot of overhead compared to using pointers and manual memory management in languages such as C and C++. The last big disadvantage is how the language is compiled. When Java is compiled, it is converted to class files that are then compiled using JIT (just in time) compilation at run time, which limits the optimization of code, often making it suboptimal.

3.0 COMPATIBILITY

3.1 HARDWARE COMPATIBILITY

The physical robot design is limited to the items available in the LEGO EV3 Mindstorm kit. This kit has hundreds of pieces, but parts can only be connected in a limited number of ways, limiting the design possibilities. It may be possible to produce custom parts using 3D printing – this option will be investigated later if necessary.

3.2 SOFTWARE COMPATIBILITY

Every piece of software that was developed during the semester has been written in Java, meaning that every piece code we have can easily be adapted to work with one another. Every tool we are using is multi-platform so it accommodates people using Windows, Linux and MacOS. Tools like Git also take care of the different text formatting of the different operating systems (line endings) and converts it all to the same format, making it very easy to work with multiple platforms.

4.0 REUSABILITY

4.1 HARDWARE REUSABILITY

Not much of the hardware design will be carried forward from the labs, because the robots in the labs made many assumptions. For example, the labs required the robot to have only one or two sensors and two or three motors equipped, whereas the project robot will have at least four of each (or more). The project robot will need to have an arm to

grab the zipline, and in general will need to be larger and more stable to accommodate more sensors and motors.

The motor structure is mirrored on either side of the robot. Aside from the general tri-bot design, little is kept from the original labs. Most of the structure of the robot had to be changed to support more sensors and more motors.

4.2 SOFTWARE REUSABILITY

As mentioned above, since every piece of code from the laboratories is written in Java, everything that has been done during the semester is reusable for the final project. Software tools such as the odometer, the navigation and the localization will be essential to this project and will be taken from the previous laboratories and improved using the implementations of the multiple groups in the team.

5.0 STRUCTURES

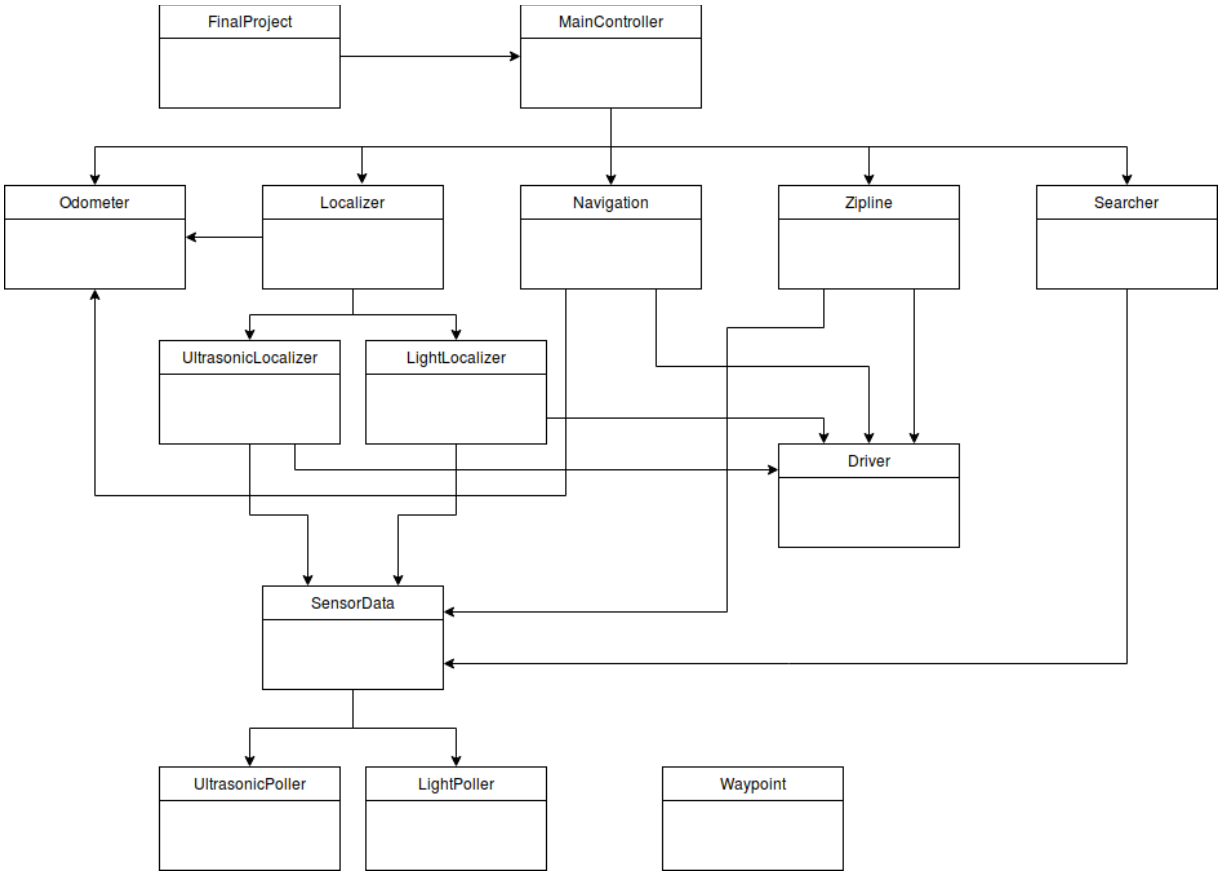
5.1 HARDWARE STRUCTURE

TODO: reasons for hardware choices
TODO: reasons for electrical choices

5.2 SOFTWARE STRUCTURE

Figure 1 displays the project's class structure, while Figure 2 shows the project's state machine.

Figure 1 – Project Class Structure


FinalProject
MainController
Odometer
Localizer
Navigation
Zipline
Searcher
UltrasonicLocalizer
LightLocalizer
Driver
SensorData
UltrasonicPoller
LightPoller
Waypoint

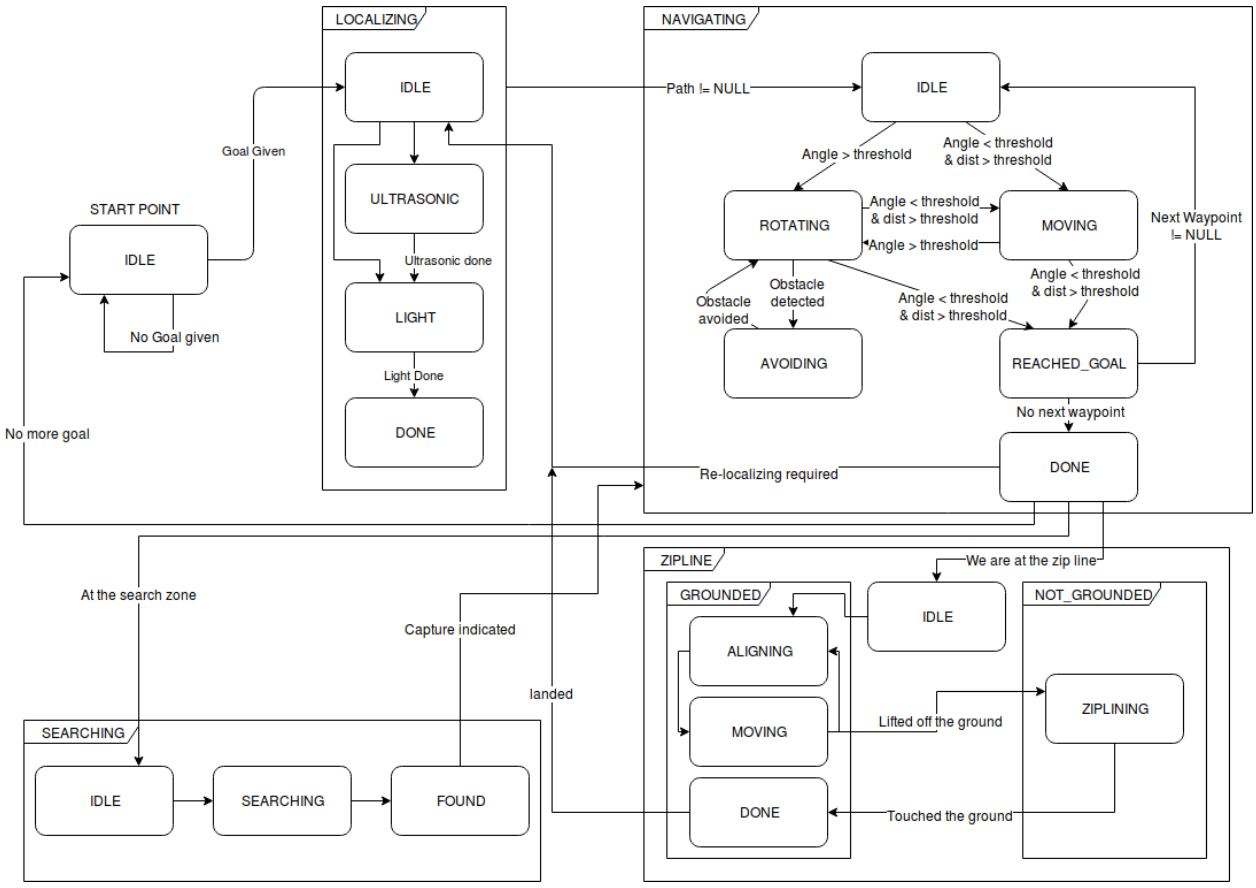


Figure 2 – Project State Machine

6.0 METHODOLOGIES

6.1 HADWARE METHODOLOGY

TODO: list of candidate solutions for parts of the problem – solutions come out of idea generation and allow for critical analysis before final design is performed

6.2 SOFTWARE METHODOLOGY

Software is going to be developed using the iterative development process. We are going to be constantly testing the software to find issues and adjust the design accordingly.

7.0 TOOLS

- Git / GitHub
- Eclipse
- Google Drive
- Slack
- leJOS
- Lego Mindstorms

8.0 GLOSSARY OF TERMS

None required