ECSE 211 DESIGN PROJECT

SYSTEMS DOCUMENT

Version *1.04*

02/25/2018

ECSE 211 TEAM 11

VERSION HISTORY

Title	Systems Document			
Description	Week 1 iteration of this document			
Created By	Luka Jurisic, Documentation Manager			
Date Created	19th February 2018			
Maintained By	Luka Jurisic			
Version Number	**Modified By**	**Modifications Made**	**Date Modified**	**Status**
1.00	Luka Jurisic	Created the document. Set out the overall structure that the document should follow	19th February	Initial work done
1.01	Luka Jurisic	Completed section 1.0.	20nd February	All other sections remain. Section 1.0 complete but needs sketches
1.02	Luka Jurisic	Completed section 2.0-5.0	21nd February	Sections 6.0-8.0 remain
1.03	Luka Jurisic/ Volen Mihaylov	Luka: Completed section 8.0 Volen: Designed the sketches	22nd February	Sections 6.0 and 7.0 remain
1.04	Luka Jurisic	Completed the Document. Created a Title page and perfected the presentation of the document	25th February	Final Version

SYSTEMS

1. SYSTEM MODEL

Localization: Initially, the robot will be placed at any point, in any orientation, within it's designated zone. The robot must be able to localize itself and traverse to the starting point before it begins its mission to steal the enemy flag.

Navigation: The robot must be able to navigate through both the bridge and tunnel obstacles without crashing or falling off into the water. In the remote chance of having a possible collision with the enemy robot, it must be able to avoid this as well.

Object Detection: The robot will be able to detect the presence of the enemy flag through its distinct colour. Hence, it must be able to differentiate between different colors within its surrounding.

Flag Capture. Once detecting the presence of the flag, it must be able to capture it by effectively utilizing the hardware components that make up its design.

Flag Delivery: The robot must be able to traverse through one of the obstacles again and return the captured flag to its zone.

Our designs are currently very preliminary. We are still deciding between utilizing a single brick or using two of them and creating a master/slave communication. Below, figures 1 and 2 illustrate a very basic design of our robot in its current iteration.

Figure 1: Back-left top view

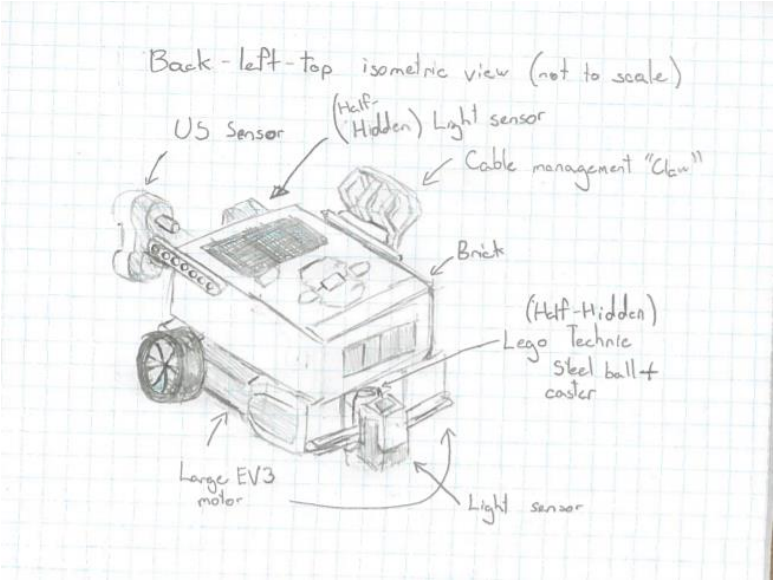
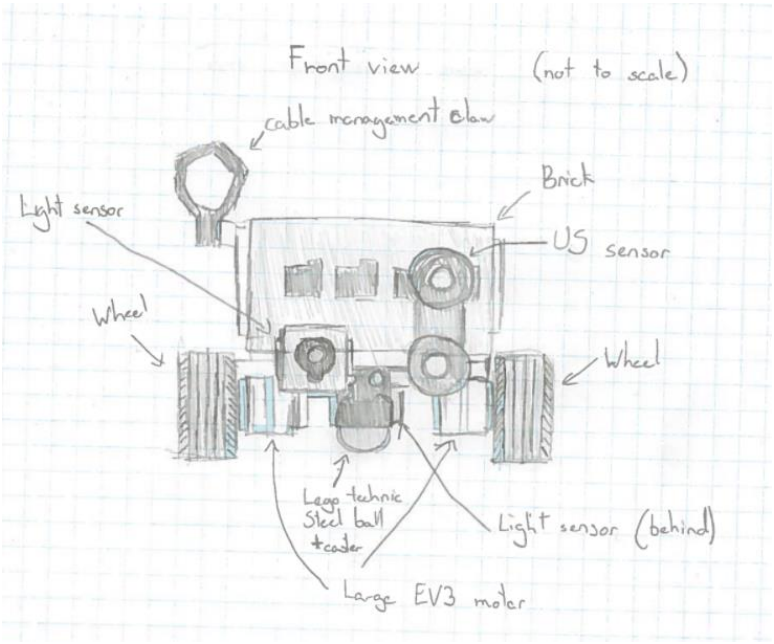


Figure 2: Front View



2. HARDWARE AVAILABLE AND CAPABILITIES

Refer to sections 1.3 and 3. in the requirements and constraints documents respectively for the hardware constraints. It is very important to note that the batteries provided by the Mind-storm Kits have limiting power provision. The final hardware design coupled with an intrinsic software implementation will degrade battery life quite quickly, leading to a decrease in overall peak system performance during competition day. In terms of capabilities, the NXT Brick is a 48MHz processor, has a maximum of 256KB of flash memory and contains 64KB of ram. To effectively utilize this, every double will be converted to a float. Floating point numbers take the same or less clock-cycles in execution. Conjointly, floats take half the size of a double, and this smaller memory footprint will allow efficient cache-usage.

3. SOFTWARE AVAILABLE AND CAPABILITIES

Refer to sections 1.3 and 4. in the requirements and constraints documents respectively for the software constraints. The use of eclipse over other well-known IDE's allows for a wide range of capabilities. It is preferred as it is completely compatible with LeJOS. Eclipse has been the IDE of choice by many of the professors at McGill that have taught the current ECSE 211 group of students. All 6 group members have grown accustomed to using Eclipse ever since ECSE 202, a prerequisite for this course.

Withal, eclipse allows for very good error detection, vital for a large software project such as this one. These advantages facilitate an efficient development process.

4. COMPATABILITY

Perhaps the most integral part of this design project is the fundamental components that make up the robot. The given Lego constituents can only put together in a limiting number of ways, thus everything else within the design must adhere to this constraint; everything must be compatible with the Lego pieces. In terms of previous development, the initial 6 weeks of the courses allowed the familiarization of the LeJOS environment, and the software developed during this time is not only compatible, but indispensable to this design project. The navigator, odometry and odometry are all integral classes that will be utilized during this design project. Likewise, 3rd party communication through wi-fi will continue to be used as it was during the labs. However, the hardware designs that were constructed for the labs cannot be used for this design project as they do not adhere to the specifications. A new design will be required.

5.0 REUSABILITY

Refer to section 3.1 in the Requirements document.

6. STRUCTURES

The robot will have three ultrasonic sensors mounted on each side and the front. The front sensor will allow for localization at the beginning and will allow for obstacle avoidance during the run. The side sensors will permit the completion of the search of the enemies “flag”. A light sensor pointing onto the grid will allow for further error reduction of localization and another light sensor point forward will allow for confirmation of the detection of the flag. The robot will be using a front-wheel drive to ease the calculations. Furthermore, a single brick will be used to lighten the load to allow for better performance. Finally, the robot will make use of the rubber tires instead of the Lego Technic Steel Ball to allow for more traction to be able to pass the bridge obstacle.

7. METHODOLOGIES

Refer to section 6 above

8.0 TOOLS

The table below outlines the tools that will be integral to the success of this project. Their respective advantages and disadvantages are summarized.

TOOL	ADVANTAGES	DISADVANTAGES
Eclipse	Refer to section 3.0	Large application size

Gannt Chart	Allows for excellent schedule management. Task division is visualized easier.	Does not differentiate the relative sizes of tasks.
Lego Digital Designer	Provides excellent visual representation of our hardware designs	The user interface is poor which leads to tedious time consumption.
Dropbox	Large storage capacity. Very easy to file share. Client approved.	Unreliable in tracking version changes.
Slack	Professional equivalent to Facebook. Excellent for promoting subtask communication and integration.	
Google Drive	Allows for documents to altered by multiple members.	Cannot replace meeting.
ObjectAid Eclipse Plugin	Automatically generates Javadoc and class diagrams from the existing code.	Difficult to customize and organize the class diagrams.