SYSTEM DOCUMENT

GROUP 4

2nd December 2016
Version 3

Contents

1 BACKGROUND

1.1 Edit History

Benjamin Willms: 2016-10-23, Initial setup and work on section 3

Mamoun Benchekroun: 2016-10-23, Work on section 6

Benjamin Willms: 2016-10-23, Work on section 8

Mamoun Benchekroun: 2016-10-23, Work on section 4

Benjamin Willms: 2016-10-23, Work section 10

Mamoun Benchekroun: 2016-10-23, Work on section 10 and 7

Quentin Norris: 2016-10-24, Work on section 5 and 7

Benjamin Willms: 2016-10-24, Work on section 2

Jake Zhu: 2016-10-24, Added table of contents

Benjamin Willms: 2016-10-24, Work on section 9

Benjamin Willms: 2016-10-24, Final touches

Jake Zhu: 2016-10-28, Ported to LaTeX

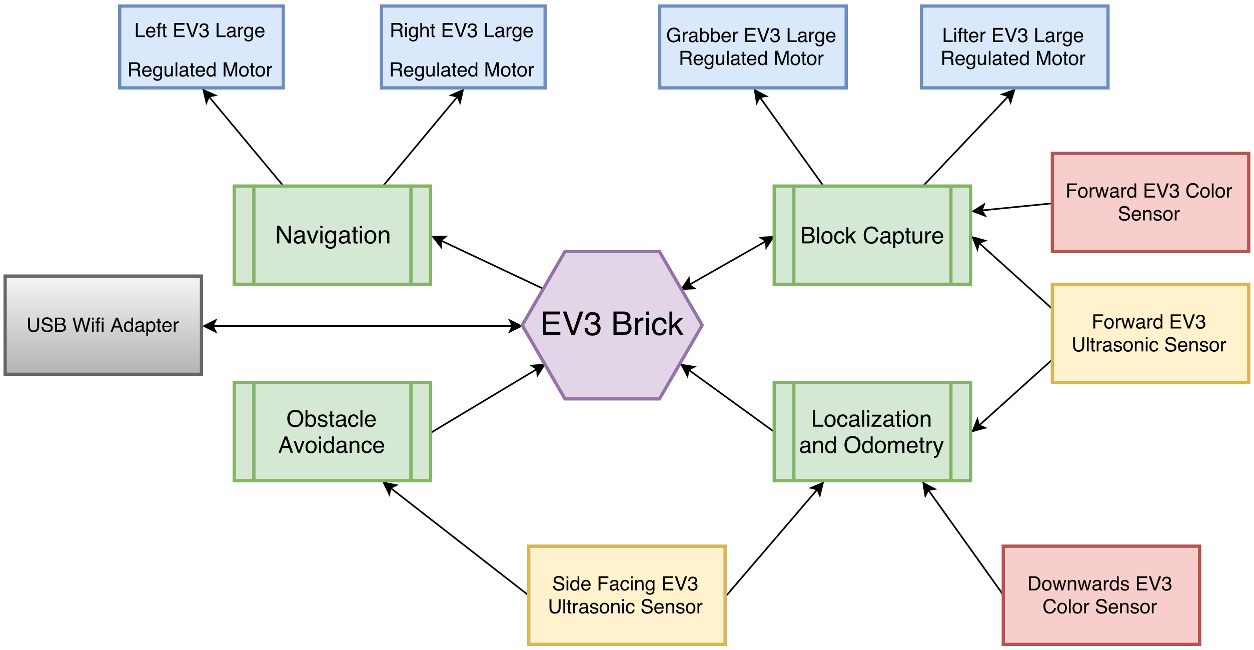
Benjamin Willms: 2016-10-31, Reworked section 8

Benjamin Willms: 2016-10-31, Edited section 10, and converted section 1 to the numbering system adopted in the port to LaTeX

Benjamin Willms: 2016-11-04, Edited section 8 to include the gravity fed hybrid design

Kareem Halabi: 2016-11-26 Updated systems model and performed general edits on document

2 SYSTEM MODEL



3 HARDWARE AVAILABLE AND CAPABILITIES

Lego EV3 kits:
These kits include various structural pieces, connection pieces, bars, bar caps, wheels, gears, tires, tracks, 12 large motors, 3 medium motors, 3 ultrasonic sensors, 3 rechargeable batteries, 3 colour sensors, 3 gyroscopes, 6 touch sensors, 3 EV3 bricks, 3 USB cables, numerous RJ12 cables, 3 chargers, and various other specialized Lego pieces.

3.1 Mechanical Components

Due to the limitations of Lego, the robot must follow a very specific structure, limited in discrete sizes by steps of a single connection hole. The placement of these holes will limit the possible locations of the sensors, motors, and each structural component. The number and sizes of gears available limit the complexity and placement of the possible mechanical systems.

The strength of the structure is limited by the strength of each Lego piece, and the structural integrity (shifting of the pieces, bending, etc.) is also limited by the Lego format. Lego is made of plastic, and can flex when under the forces exerted during the robots motion and simply from the weight of the robot itself (as seen in the labs). The connections between pieces are not perfect, and can have a small amount of give, causing small shifts in the structure.

3.2 Electromechanical Components

The RJ12 cable lengths and flexibility will limit the positions of motors and sensors in relation to the EV3 brick, or bricks. These cables and the brick(s) will also limit the structure of the robot, insofar as needing to avoid blocking the ports, and needing to leave space for the wires to snake through the structure.

The motors are limited in their torque output. The batteries charge could also affect the torque available. This will limit the weight of the robot, the extra friction allowed between the robot and the ground (when pushing/pulling a block, or blocks, as the design may require), and the possible weight to be lifted when stacking the blocks.

3.3 Electronics/Processor

The EV3 brick runs an ARM9 chip at 300MHz. From the labs completed, it has been seen that this processor can run code sufficiently fast enough. The labs used basic filters, and fewer threads than our design will no doubt use, and so further tests must be done to determine whether the processors capabilities will limit our code in any significant manner other than limiting the travel speed to allow for the processing of enough samples from the sensors.

The EV3 brick also has enough memory that we do not currently see any problems arising from insufficient memory.

4 SOFTWARE AVAILABLE AND CAPABILITIES

Java : See Constraints Document, Section 4.

LeJOS: LeJOS allows Java code to be run on the EV3 brick, and adds tools and libraries of code (for controlling the robot).

Eclipse: The most used Java IDE. Has an integrated interface for the EV3 brick that facilitates communication and send instructions reliably, by USB or Bluetooth.

Slack: Group messaging and communication service for teams using a predisposed site for each team with chat rooms relevant to each part of the project.

GanttProject : Basic project management software that makes use of the features of Gantt charts. Handles days only and has many features such as PDF and HTML reports or can generate PERT charts. Cannot handle hour-based operations, which may be crucial since our budget is 9 hours a week.

Github: Web based Git repository for managing source code and distributing it quickly and efficiently between team members.

LaTeX: Document preparation language, useful for clear reports and documentation but requires to learn a specific language. Only some members know it currently.

Overleaf : Web based editing software for documents using LaTeX. Members of the team unfamiliar with LaTeX will encounter difficulties at first.

5 COMPATIBILITY

5.1 Connections of Multiple bricks

For this project we are planning to utilize two bricks to complete our final task. This brings up the question of how do we go about connecting these bricks to work in an efficient, effective manner without mitigating development time and contributing to development issues. There are three methods to connecting the bricks: wifi, bluetooth, and a USB cable. From my research the best way to go about connecting the EV3 bricks is via wifi, in which to do so you use the interface on the EV3 brick to connect to a wifi network. Then you simply unlock the brick so it accepts a TCP/IP connection, and from here the user may send messages from one brick to another through a designated port

5.2 Implementation of code from previous labs

Labs 3, 4, 5 of the semester make up the essential building blocks to our final project. Since all four groups have completed these labs, its imperative that we utilize the code (even if it is just for inspiration) from these projects to aid in perfecting the basic functions necessary to begin working on our final project. However, it is important to take into account that the code for each of these labs was designed with the intention of solving the specific problem in place for that lab. Adaptations will have to be made in order to use it for our final project, for example in lab 5 the localization is designed for a small travelling area, with one possible starting square, and the need for the robot to be placed on the diagonal. In the project, there are 4 possible starting squares, and the robot need not be placed on the diagonal.

6 REUSABILITY

6.1 Knowledge from previous labs

For this project we could reuse most of the features developed during the labs. The localization from Labs 4 and 5 will be the key to our task completion efficiency, as its performance will greatly affect how fast the robot can localize and complete the task during the demo. The traveling methods in Lab 3 that helped us identify and avoid obstacles will also be an important structure to reutilize. While we want to reuse the code developed in past labs, it will have to be adapted to the situations of the final project.

6.2 Motors and sensors

Furthermore, motors and sensors can be used in many ways to accomplish different tasks. However, we have a limited amount of each and have to manage the reusability of these components. As we have various design possibilities, each one will make a different use of these components. For example we can have a 2-use motor design that completes the tasks of rotating the sensor and grabbing/releasing the blocks. We are essentially reusing the sensor after the first task to complete the second one. The previous design is much better than using three motors : one for a clamp, one to raise the block and one for the ultrasonic sensor rotation. These designs always take into account the use of two motors for the wheels to rotate.

7 STRUCTURES

7.1 Mechanical structures

Wheels :

We are using the ball as a third wheel at the present time. However the final demo will be on an assembled 12x12 tile wooden structure made of 4 4x4s. This means there might be some small space between the border of the 4x4s so it could be beneficial to replace the ball by treads in order to account for that possibility. The treads also offer less slippage going in a straight line but possibly more slippage while turning.

Crane :

An idea our group is considering implementing is that of a crane - a structure that is used to pick up and drop the blue styrofoam blocks. There are several types of crane designs that we are comparing. The main type of claw we are considering is a motor attached to a string with a grabber on the end. As the motor pulls the string up the grabber will close over the block and lift it. We are considering using a crane to pick up the blocks as we have decided it is very possible to implement utilizing various hardware and software we have available, in addition to making the second part, the stacking of the blocks much more feasible as the blocks will be lifted in the air. This makes them easier to stack

Cage :

A method of storing the blocks picked up would be through that of constructing a cage. The preliminary design ideas we have so far is a skeleton tower consisting of the legos combined from the four EV3 kits placed over the two EV3 bricks. The purpose of the cage is to hold the blocks picked up by the crane and grabber and store them until we encounter more blocks. Once the tower is full there would be a function to release or detach the tower from the cage, or detach the cage entirely. The rationale for using a cage to carry the blocks instead of stacking them is to save time and making the stacking of the blocks more efficient. It is also less software intensive to do the stacking via a hardware component, which mitigate the potential for error.

7.2 Electrical structures

The brick :

We have two choices currently : use one brick to centralize communication or use two bricks to have functionalities at the cost of communication quality. If we use one brick, we will be restricted to 8 total ports for components which means implementing designs similar to the 2-use motor design where each motor can do multiple tasks. However we are not yet sure how to fit all the required tasks into the one brick model. On the other hand, a second brick will allow us to have more functionalities. Doubling the amount of ports will help us improve the vision of the robot by having multiple sensors that rotate and more motors to support the crane/clamp movements. Thus, a second brick with more sensors could be a potential design. Nevertheless, this increase in functionality comes at the price of potentially reducing communication quality; but this may be solved by connecting the bricks using a USB cable.

8 METHODLOGIES

8.1 Hardware

Each hardware section will have multiple ideas brainstormed. We will choose 1-2 of the ideas from each section, and build each of them. After each one is built, or determined to be either impossible to build, or too complicated to be worth the possible performance increase, the finished designs will be tested to determine which ones make it into the final design.

Collector Ideas:

- Two arms that swing into a shut position, fixing the orientation of the block to be manipulated. The arms could swing from two positions: on the side of the robot or on the front of the robot.
- Two arms that slide into position, instead of swing.
- Two arms placed in an angled orientation, to funnel the block to the correct position, and orientation.

Grabber Ideas:

- A pinching claw design for grabbing the blocks. A secondary enclosure would lower onto the claw, forcing it to clamp down on the block. The entire system would then be raised to the correct height, where the enclosure would then be raised, releasing the claw.

- A partial platform design that relies on friction between two blocks, or the block and the floor, to release each block.
- A motorized grabber. This could take many different forms including a claw design, a clamp design, an ”excavator” design, etc.

Lifter Ideas:

- A tower structure with one motor driving up and down it to raise and lower the grabber.
- A tower structure that guides the grabber, with a single motor raising and lowering it with cables.
- A crane structure which uses cables for lifting.

Hybrid Ideas:

- A cage that lifts and holds the blocks inside, creating a stack that the robot will carry around until the time limit nears completion, or the cage hits its capacity (this would require a specialized collector or grabber to function properly, and is thus included in the Hybrid section).
- A 2-use motor design that uses a single motor for both rotating the ultrasonic sensor, and grabbing/releasing the blocks (reusing the same motor for two different tasks). This would be used with a crane, or tower lifting structure.
- A collector that squeezes tight enough to be able to hold onto the block when lifting. This would nullify the need for a grabber, as the collector would function as the grabber.
- A gravity fed grabber/collector which will use downward motion, coupled with upward force from the block, to close a grabbing device, and a lever system to release the mechanism.

Locomotion Ideas:

- The common design from the labs: two wheels and a metal ball bearing at the back.
- Two wheels and small, rotating gear wheel at the back.
- Two tracks, like a tank.

Localization Idea:

- A bumper on the front, or back of the robot, allowing quick and easy orientation to the y or x axis. We could drive into a wall and then that will localize the angle.

Ultrasonic Mount Ideas:

NOTE: All ultrasonic sensors must be mounted close enough to the ground to detect the styrofoam blocks.

- A large motor mounted on one side of the robot, to avoid obstruction by a collected block.

- A medium motor mounted in the center (in the case of the collector/crane being in the back), or on top of the cranes grabber (in the case of the 2-use motor idea for the crane).
- A medium motor with a gear system placed underneath the brick, in the center, so that the ultrasonic does not extend past the front of the robot, but can still rotate as much as necessary.

Placement of the Stacking Mechanism:

- The whole mechanism could be placed on the front of the robot.
- The mechanism could be placed at the back of the robot.

8.2 Software

Each section of the software will be planned out in detail before beginning the coding, to ensure a coherent, efficient design, and to save as much time as possible in the coding process. Multiple possibilities for each section will be discussed, and (barring unforeseen problems) the most appropriate solution will be implemented.

Localization Ideas:

NOTE: The colour sensor/s will use the finite differences filter, to nullify the dependency on ambient light. The ultrasonic sensor should use a moving window of values for a median filter.

- The same idea as in Lab 4, except improved to successfully localize within 30 seconds.
- A falling edge detector with avoidance of the corner, coupled with the mechanical bumper idea. This would orient the robot along one axis. The robot would then travel to detect two lines to finish the localization.

Odometry Correction Ideas:

NOTE: The colour sensor/s will use the finite differences filter, to nullify the dependency on ambient light.

- Use a single colour sensor to detect grid lines and correct the x and y values, as in lab 2.
- Use 2 colour sensors to correct the orientation of the robot when crossing lines sufficiently far away from line intersections.
- Periodically (for example, every 1 minute), travel to a specific location, and use a light sensor and the lines to correct the orientation, x, and y values.

Block Identification Ideas:

NOTE: The ultrasonic sensor should use a moving window of values for a median filter.

- Use a single colour sensor to determine the color of the block, and so, determine the whether it is a target block, or an obstacle.
- If a single colour sensor isnt reliable enough for identification, an ultrasonic could also be used to scan the block and determine its dimensions (height may be the best indicator).

Obstacle Avoidance Ideas:

- Once the object is determined to be an obstacle, the robot could back up to a safe distance, and start a bang-bang or p-controller, using the ultrasonic sensor, to avoid the obstacle.
- Once the object is determined to be an obstacle, the robot could have a pre-determined avoidance path that it calls, which would consists of traveling in 3 steps around the obstacle, with the possibility of being interrupted by block identification.

Mapping Idea:

- The robot could store the positions of obstacles it finds, mapping the area so as to be able to be able to avoid obstacles without the use of the ultrasonic sensor when it is indisposed (possible depending on the placement of the sensor, and the crane/collector design).

8.3 Testing

Testing will mostly follow the process of requesting a test of certain capabilities, the design of the test, and then the running of the test itself.

9 TOOLS

ALL TOOLS DESCRIBED IN SECTION 3.0

10 GLOSSARY OF TERMS

Collector (section 8.1): The hardware component that ensures that a styrofoam block to be picked up has the same orientation as the robot, and is in position for the grabber to pick it up.

Lifter (section 8.1): The hardware component that raises and lowers the grabber, allowing the stacking of blocks.

Grabber (Section 8.1): The piece of hardware that secures the block for lifting.

PERT chart (section 3) : Program Evaluation and Review Technique or PERT is a tool for analyzing and representing tasks in a statistical structure.