Robotics: Science and Systems

Practicals - Final Report

Design, mechanics, and logic of an autonomous robot made out of LEGO parts

Henrique Manuel Martins Ferrolho
s1683857 - henrique.ferrolho@gmail.com

November 21, 2016

Design, mechanics, and logic of an autonomous robot made out of LEGO part

Robotics: Science and Systems - Practicals Final Report

Henrique Ferrolho

Abstract

You MUST preface the report with a 100-200 word summary of what it contains. This is usually easier to write when you have finished the report. It should briefly explain the task, the approach used, the results and the conclusions drawn. Avoid making entirely generic statements that could apply to almost anything, e.g., (BAD) "This report describes the construction of a robot to perform a task. We describe the design decisions and outline the control program, then explain the results and possible improvements". Instead make it specific to what you have done, e.g., (GOOD) "We have built a robot capable of searching for and recognising special locations in a lab environment. It uses two IR sensors to avoid obstacles, and a low cost camera to recognise resources and target locations, as well as a sonar sensor for navigation. We implement a subsumption control architecture. The robot was tested in five time trials and was able to locate an average of 4 resources and 3 target locations within that time. The main limitation was that our robot was unable to reliably plan its route to the next location but relied on random search".

Contents

1 Introduction

The presented task was to build a robot capable of autonomously navigating through an arena, avoiding obstacles and walls, looking for three unique textured cardboard cubes, and delivering each cube to its respective base - each of them belonging to one and only one base. The entire task should be fulfilled in under 5 minutes.
The arena layout was static, and contained two bases marked with a black rectangle on the floor. The assignment of each cube resource to its base can be easily configured in the robot's source code - as it should, since such assignment were to be changed just before the final practical demonstrations.

The entire project was divided into two major milestones. Each milestone was approached by dividing it into other small sub-tasks. Each sub-task was then completed in a sequential manner, from the most simple to the most daunting. The sub-tasks required for the first major milestone had a bigger priority, and thus were approached first.

It is important to note that the task was not approached by building the entire physical layout of the robot first, and only then programming it. On the contrary, the robot was built iteratively: the physical layout got slightly tweaked each time a sub-task required it. The final physical layout of the robot still used some elements of the initial design, but overall it changed completely.
The same thing can be said about the source code: the architecture was iteratively adapted to integrate new modules. Each module was separately tested before being integrated into the main program.
There were times when the source started to have code smells which required a minor (and some times a major) refactoring. Close to the end of the project a major refactoring had to be done in order to implement a proper State Machine. In retrospective, the State Machine should have been one of the first things to have been outlined and implemented - it would have saved a considerable amount of time throughout all the development process.

2 Methods

2.1 Essential components

This section contains lists of all the components required by the final robot.

Power and logic boards:

- Fit PC
- Power board
- DC motor control board
- Servo control board

Sensors and camera:

- Light sensors (x3)
- IR sensors (x2)
- Camera
- Sonar
- Hall effect sensor

Actuators and battery:

- DC motors (x2)
- Battery

Up until the first major milestone, the robot also had two whisker sensors, but they have since been removed. Their initial purpose and the reason why they were removed are documented on section Sensing: Whiskers

2.2 Physical architecture

The entire structure of the robot is made out of LEGO parts. The robot has two large rubber LEGO wheels on the front, and one LEGO steel ball caster on the back. The robot is driven by the two large wheels on the front, and the caster wheel is just for support.
This design was preferred from the beginning because it allows the robot to rotate on itself, i.e., it can rotate any amount of degrees without actually moving relatively to the arena.

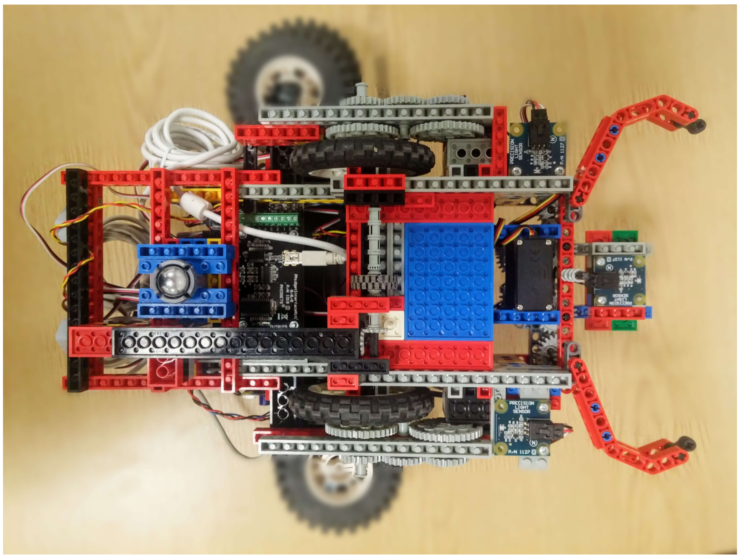


Figure 1: Bottom view of the robot. The caster wheel can be seen on the left of the image, and the two large rubber wheels on the center. The other wheels on the blurry background are not part of the robot and were used just to support the upside down robot while this shot was taken.

To take advantage of the quite heavy battery of the robot, it was placed as close to the front wheels as possible. Doing this added pressure on the tyres of the wheels, increasing the friction between them and the ground of the arena - which is considerably slippy from all the dust.

The power boards and the Fit PC were mounted on the back of the robot, between the two front wheels and the caster wheel. Doing this, and placing the battery close to the front wheels ensured that the center of mass would stay between all the wheels, approximately evenly distributed by them. This was a crucial element of the robot's design.

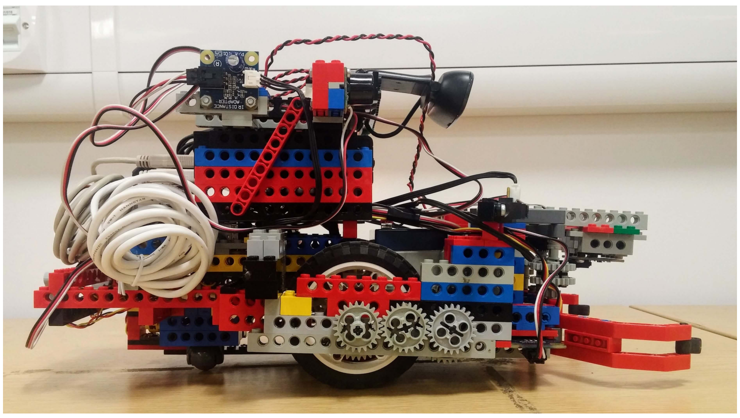
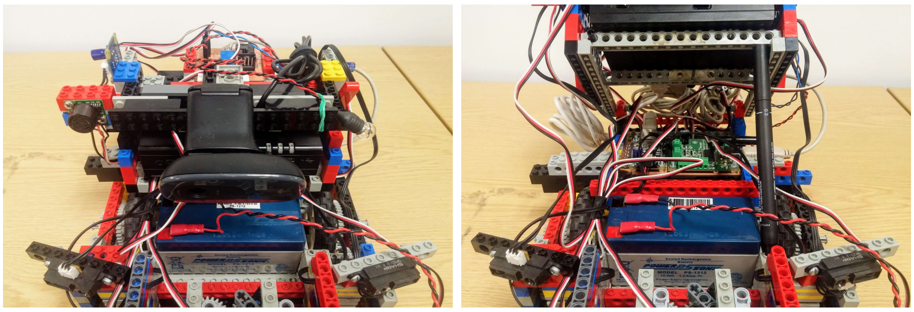


Figure 2: View of the right side of the robot.

It was decided from the very start that the main Power Board should be easily accessible to plug the AC adaptor and to turn On/Off the DC motor power board. For that reason, the Power Board was placed on the very top of the robot, just behind the camera.

The need to constantly plug and unplug sensors to the Phidgets Boards made it clear that there should be an easy way to get to those sensor slots. This was even more important with the constant battery switching when they ran out of power. To solve this problem, a *Hop On - Hop Off* with hinges was devised. To gain access to the boards and to the battery slot, one needed only to lift the hop.



(a) Hop on (b) Hop off

Figure 3: Lifting the hop reveals the Phidgets boards and enables easy access to the battery slot.

2.3 Base detector and gripper

The gripper had to be placed on the front in order to catch the cubes when the robot drove to them. The precision servo motor was used to open or close the gripper handles.

One light sensor was mounted facing down on top of the gripper dock, at such a height that a cube would fit just underneath it. Since the cubes are textured on the sides, but are solid black on the top and bottom, placing the light sensor in such position made it very easy for the robot to detect whenever there was a cube on its gripper dock or not.

To complement the gripper, two more light sensors were placed at the front of the robot, one in each side, just hovering the floor and facing down.
These constituted the base detector for the black rectangular bases on the arena, and they functioned very similarly to the cube detector on the gripper: they measured the amount of light reflected by the gray floor of the arena, and triggered whenever black was being sensed instead of gray - this meant that that sensor was on one of the bases.
When the robot moved forward, carrying a cube and suddenly both light sensors from the base detector got triggered, it meant the robot had arrived to the base and it could release the cube to deliver it.

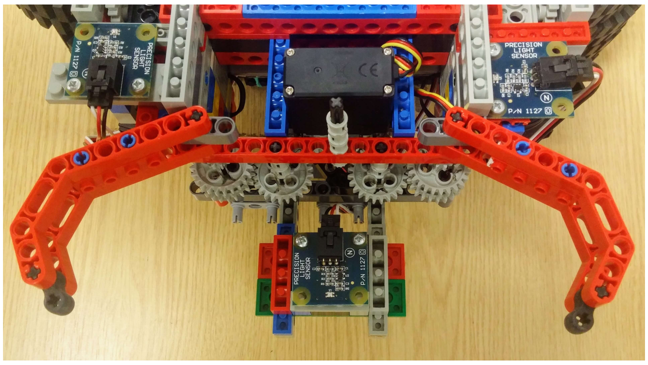


Figure 4: The base detector light sensors can be seen on the left and right top corners of this image. On the center, there is the precision servo motor, the gear train to open or close the gripper, and finally, on the middle bottom, the cube detection light sensor.

2.4 Actuators and gearing

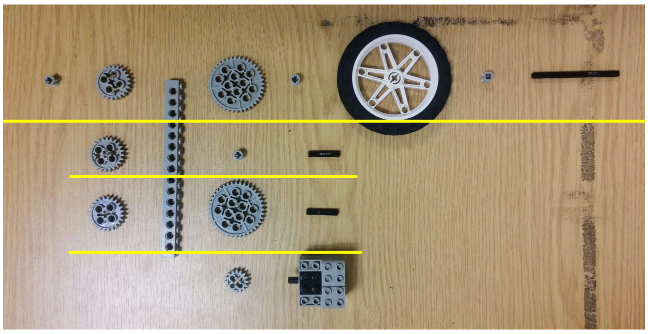


Figure 5

2.5 Sensing

2.5.1 IR and Sonar

2.5.2 Camera

2.5.3 Hall effect sensor

2.5.4 Whiskers

For the control program you should provide a flow diagram or pseudo-code description, and again explain the reasoning that led to this solution.

This is likely to be the longest section of the report. Do not include code except for short snippets that help explain a crucial part of the program you created.

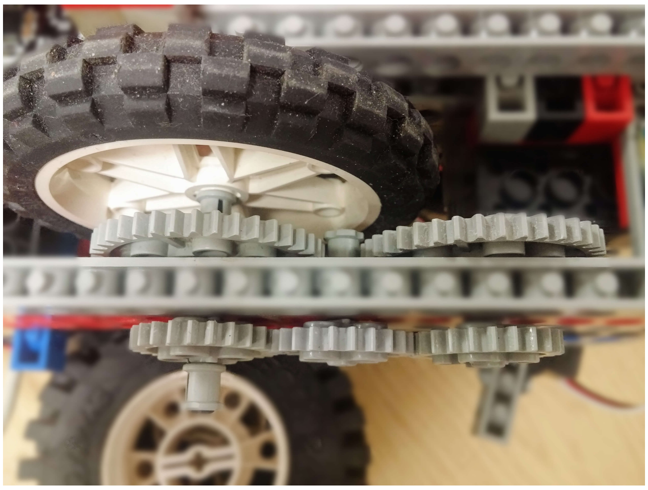


Figure 6

Avoid repetition and refer to other peoples' work instead of describing well known algorithms. (1400 words)

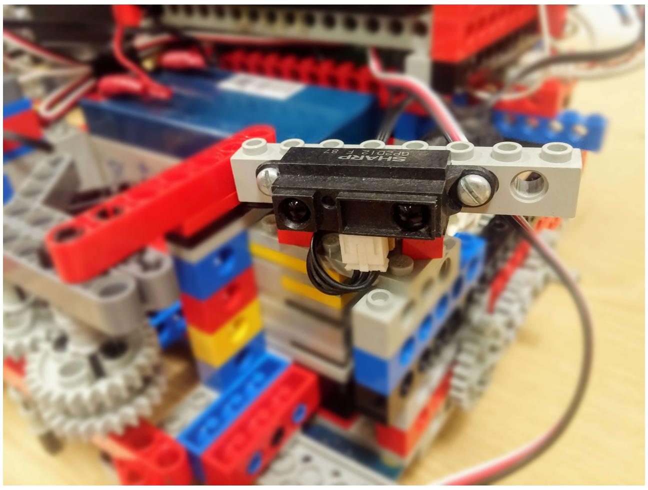


Figure 7

3 Results

3.1

This should contain some quantitative evaluation of the robot performance. For example: that it can find a resource site from a disance of x metres, and recognise and leave within t seconds; etc.

If your robot is not capable of doing the final task, you should evaluate what it does do correctly, and try to analyse what it does wrong.

The reader should be left with an accurate understanding of exactly what your robot is capable of, even if this is not as good as you hoped. Bad results are results too. You get marked based on how you approached the problem and how you evaluated the results. (800 words)

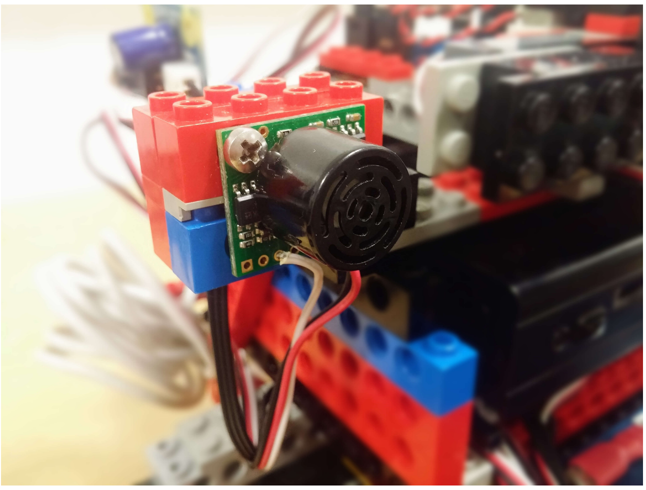


Figure 8

4 Discussion

4.1

Start by summarising the results, and giving your evaluation of how well it works. Explain what you think were the most successful elements of your approach, and what was less successful. Include ideas about how the system could be improved. (200 words)

4.2

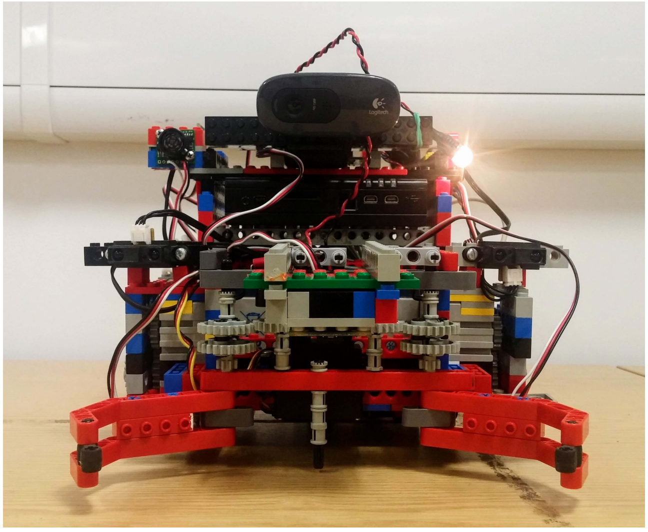


Figure 9

5 Sources

Python 2.7 Docs
https://docs.python.org/2.7/

OpenCV 3.0.0 Docs
http://docs.opencv.org/3.0.0/index.html

The Art of LEGO Design
http://www.cs.tufts.edu/comp/150IR/artoflego.pdf

LEGO Design
https://www.clear.rice.edu/elec201/Book/legos.html

Gears, Pulleys, Wheels, and Tires
http://www.ecst.csuchico.edu/~juliano/csci224/Slides/03%20-%20Gears%20Pulleys%20Wheels%20Tires.pdf

LEGO Gear Ratio Calculator
http://gears.sariel.pl/

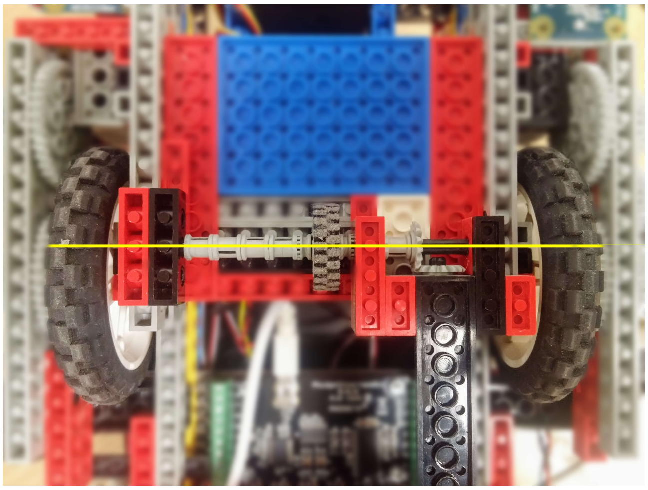


Figure 10

A Appendix

A.1 Extra images

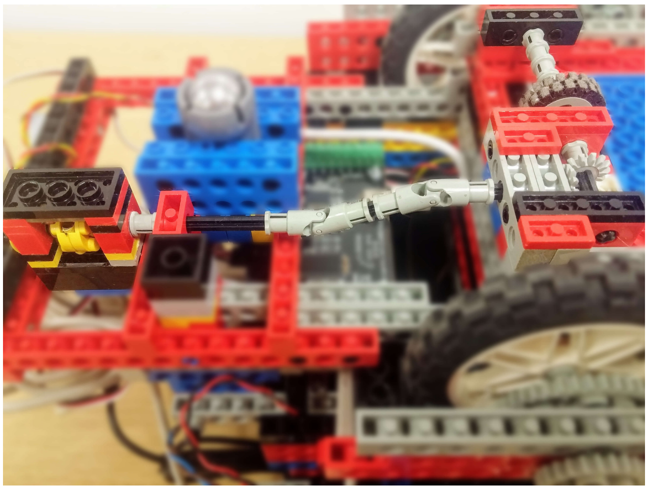


Figure 11

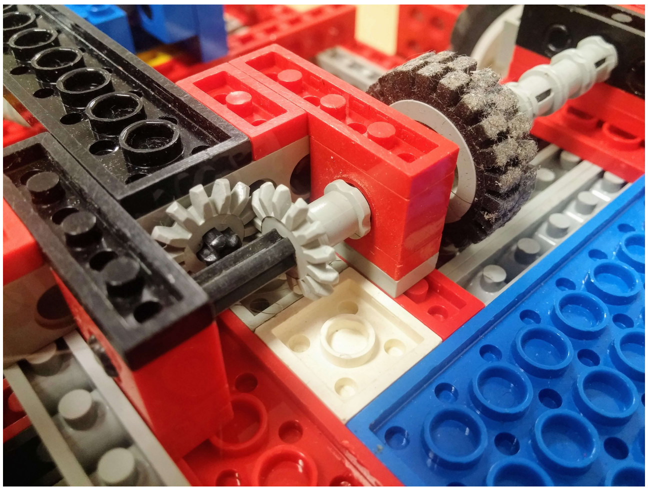


Figure 12

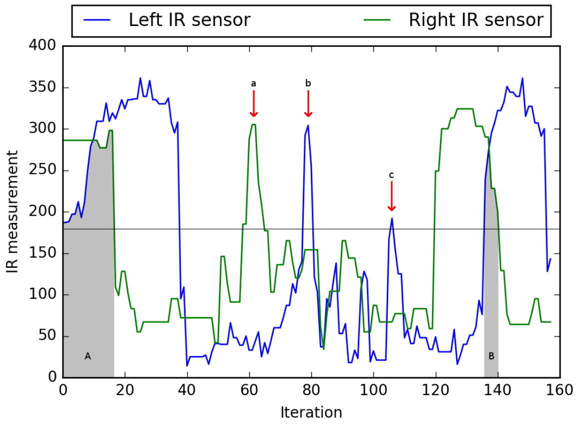
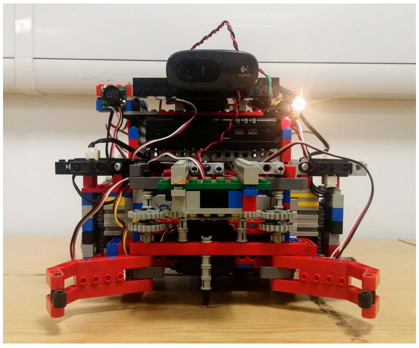
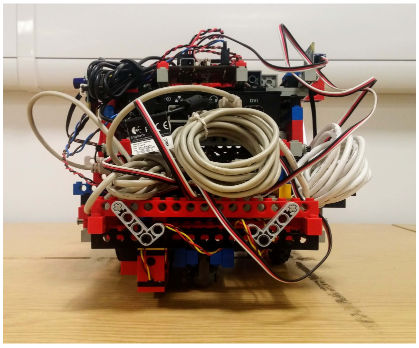


Figure 13

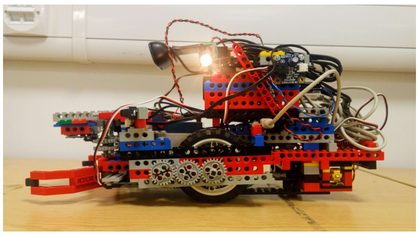


(a)

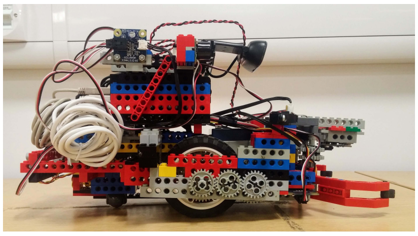


(b)

Figure 14

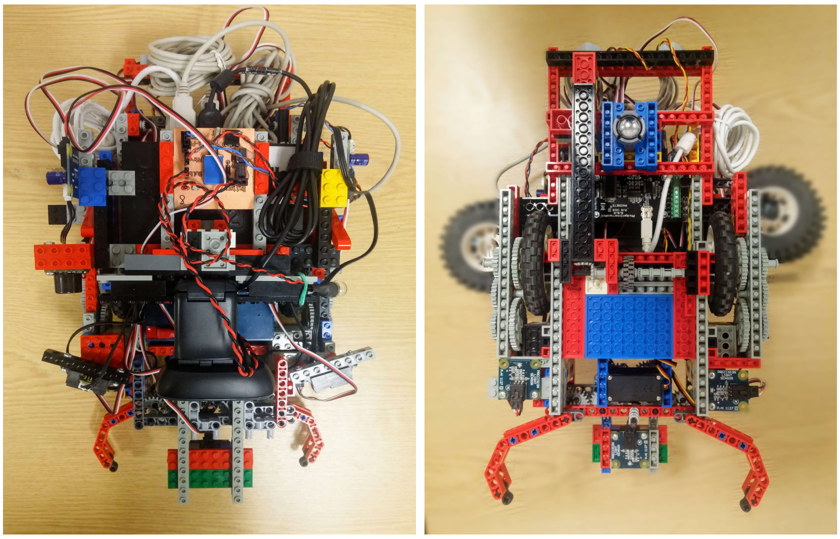


(a)



(b)

Figure 15



(a) (b)

Figure 16

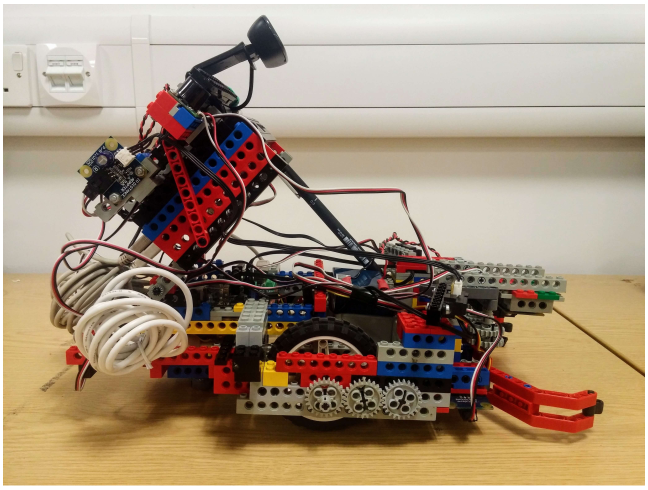


Figure 17: Hop off, side view.

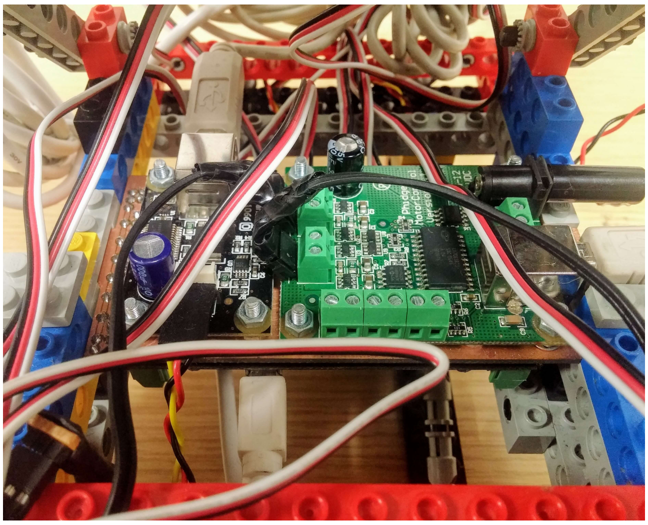


Figure 18

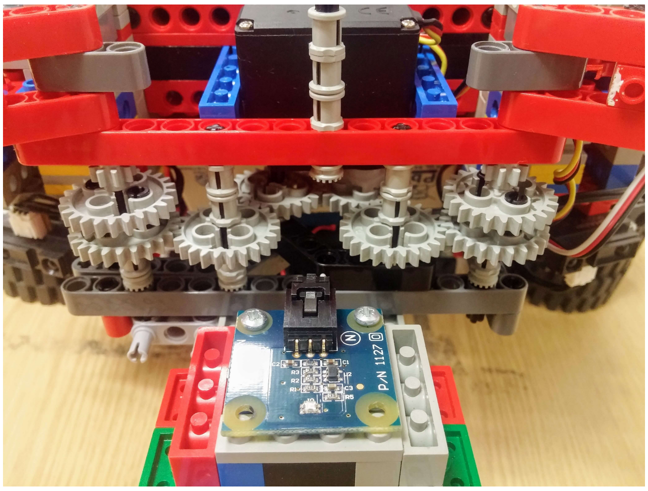


Figure 19

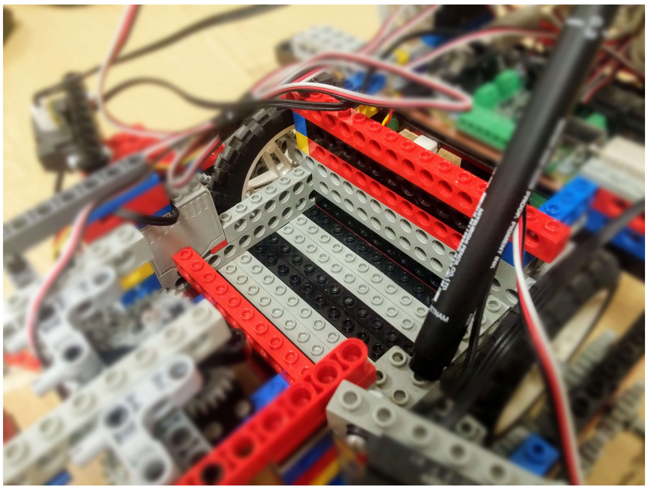


Figure 20

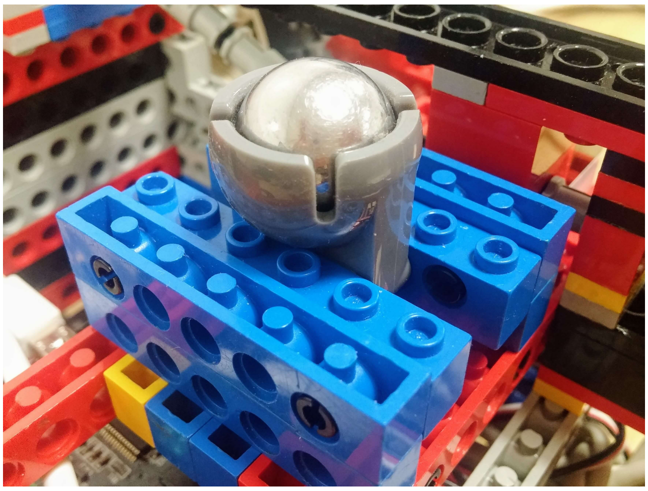


Figure 21