5CCS2RGP

Robotic Group Project

Tomas Vitek, Oluwasheun Adebari, Josh Manuncia

King's College London

2012/2013

The team

- Tomas Vitek

  the team leader; chief programmer; hardware designer

- Oluwasheun Adebari

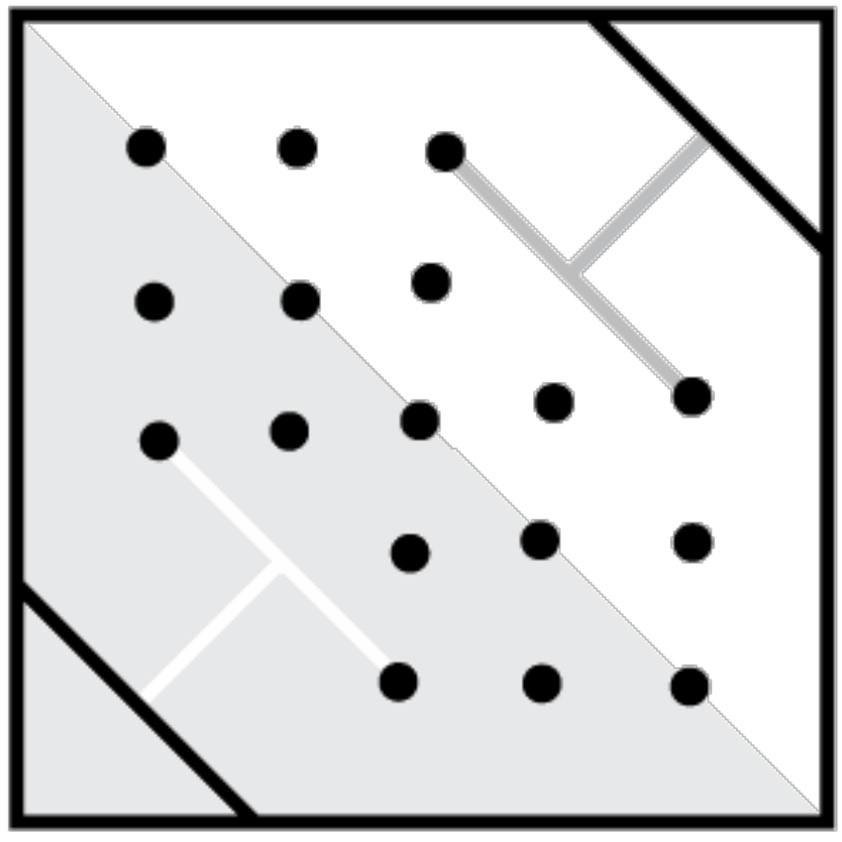
  the team secretary; programmer; hardware designer

- Josh Manuncia

  programmer; hardware designer

The Problem: Ball Collection

- Design a robot using NXT LEGO Mindstorms
- Create a program in NXC to collect more balls than opponent
- Solution needs to work only on given map with known ball positions



Team Cooperation

- We have set a schedule during our 1. meeting and we have been able to keep up with it
- Both hardware and software parts of the project are products of teamwork
- Every member has done research at home, results then brought together in a group in the lab on meetings every week
- Team management - democratic spirit, overall project organisation by the team leader

Our Schedule

- **6th Feb:** Hardware Design Ready
- **13th Feb:** Testable Ball Collecting Mechanism
- **20th Feb:** Final Version of Ball Collecting Mechanism
  Working on Unloading Mechanism
- **27th Feb:** Working on Search and Turning Algorithm
- **6th Mar:** Search and Turning Algorithm Completed
- **13th Mar:** Final Version of Hardware and Software Done
- **20th Mar:** Presentation and Robot Ready for Competition

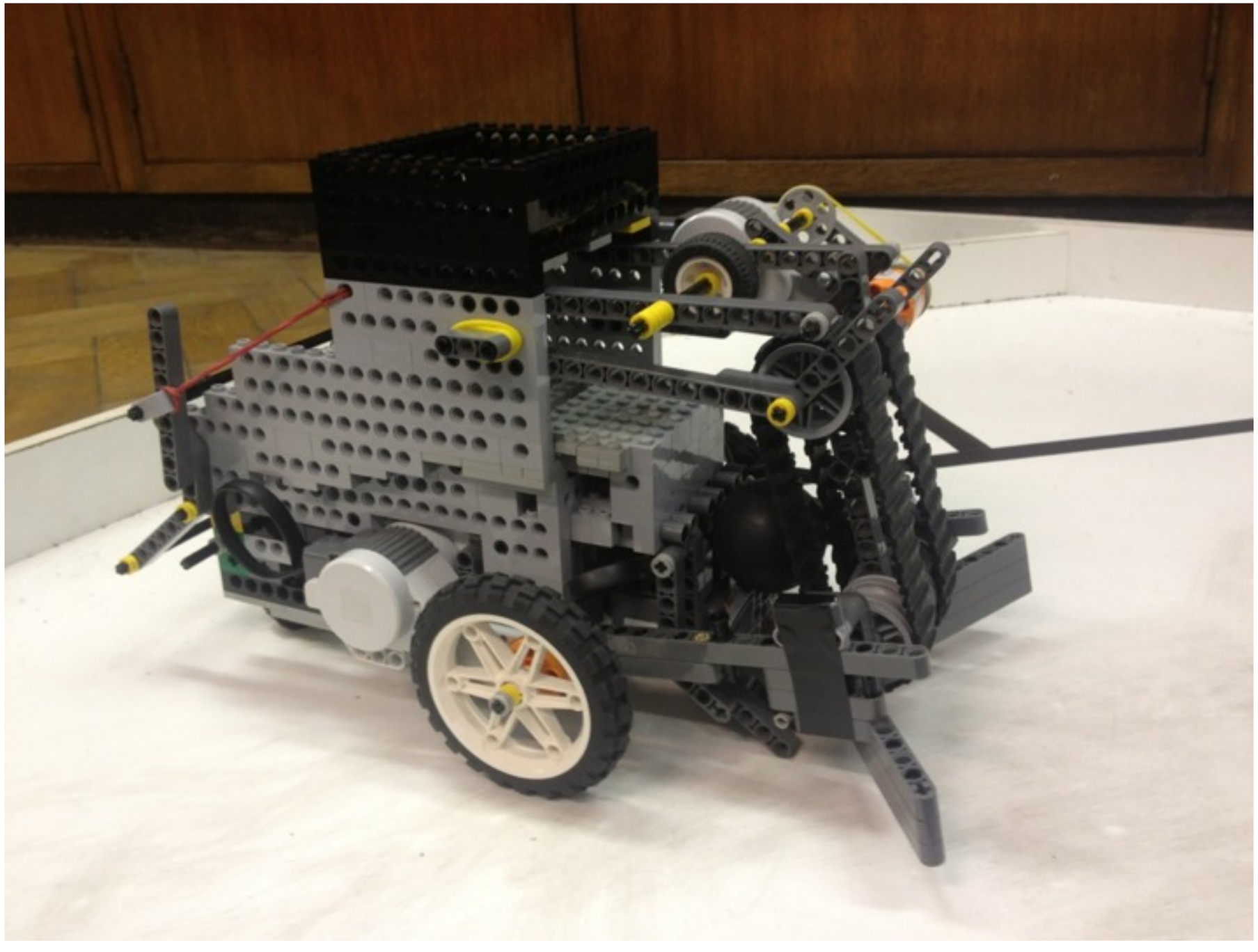
Our Solution

Hardware

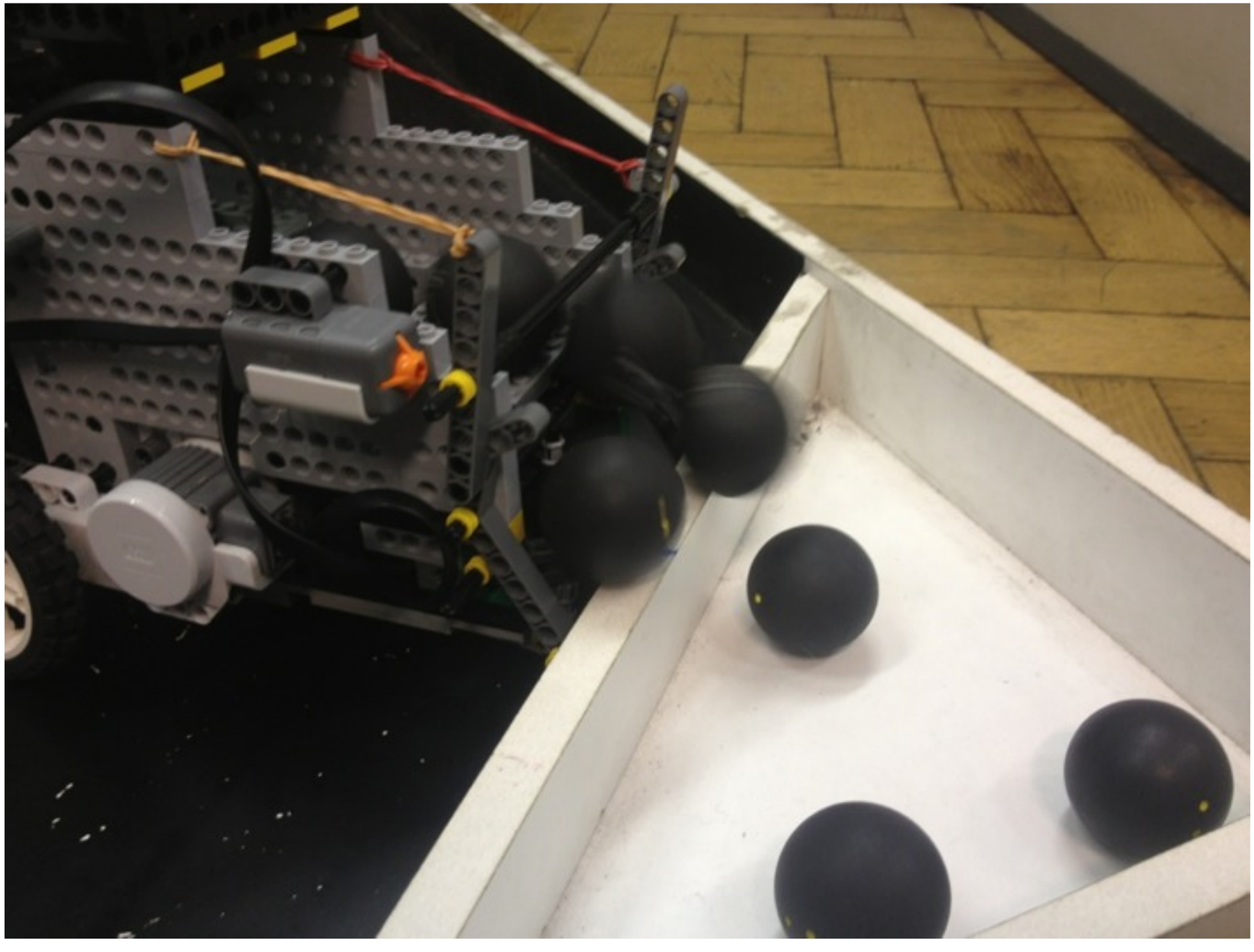
Three main goals:

- Fast and reliable ball collecting
- Easy ball unloading
- Accurate and precise

Ball Collecting



Ball Unloading



Software

We have researched and tried two approaches:

- Adaptive Ball Searching

  Searching for balls using ultrasonic sensor

- Preprogrammed Route

  Using precise movement control and predefined route to collect balls as fast as possible

Software

We have researched and tried two approaches:

- ~~Adaptive Ball Searching~~

  ~~*Searching for balls using ultrasonic sensor*~~

- Preprogrammed Route

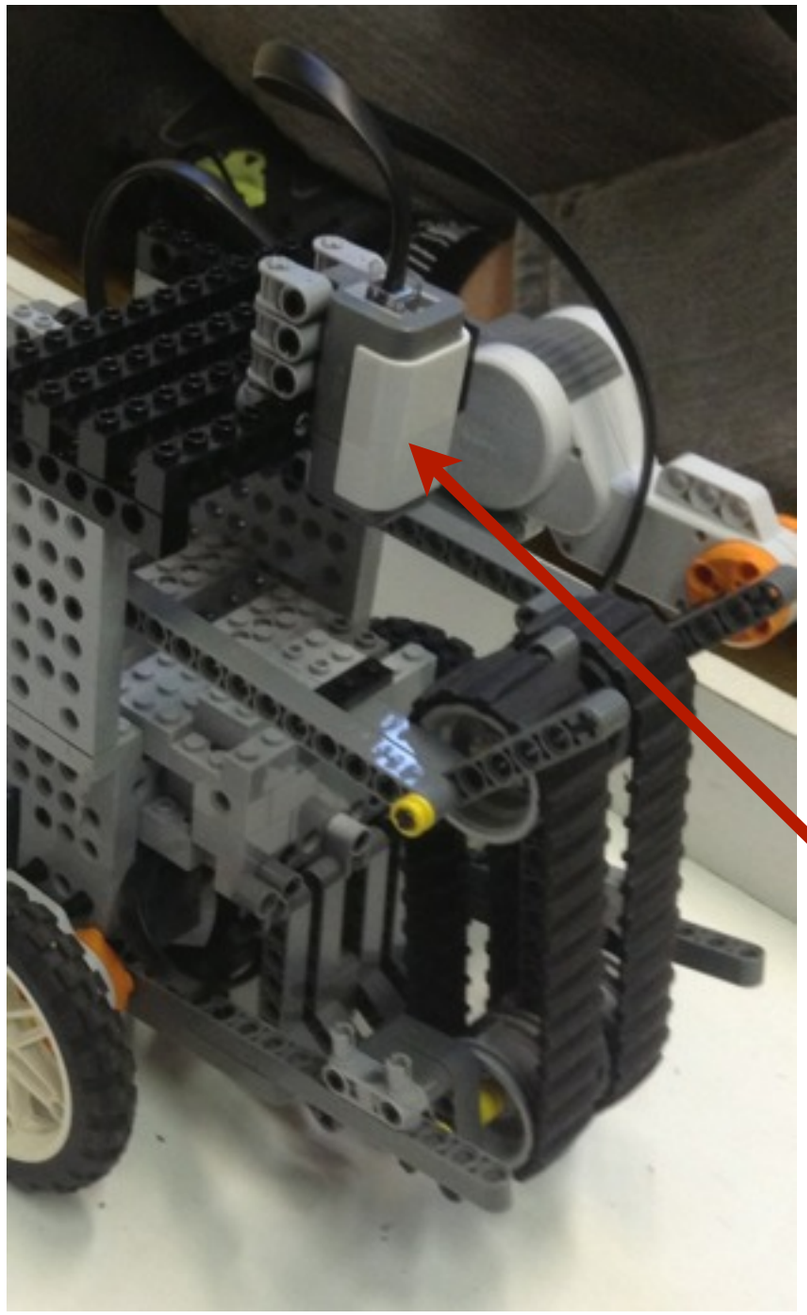
  Using precise movement control and predefined route to collect balls as fast as possible

Software

Preprogrammed Route

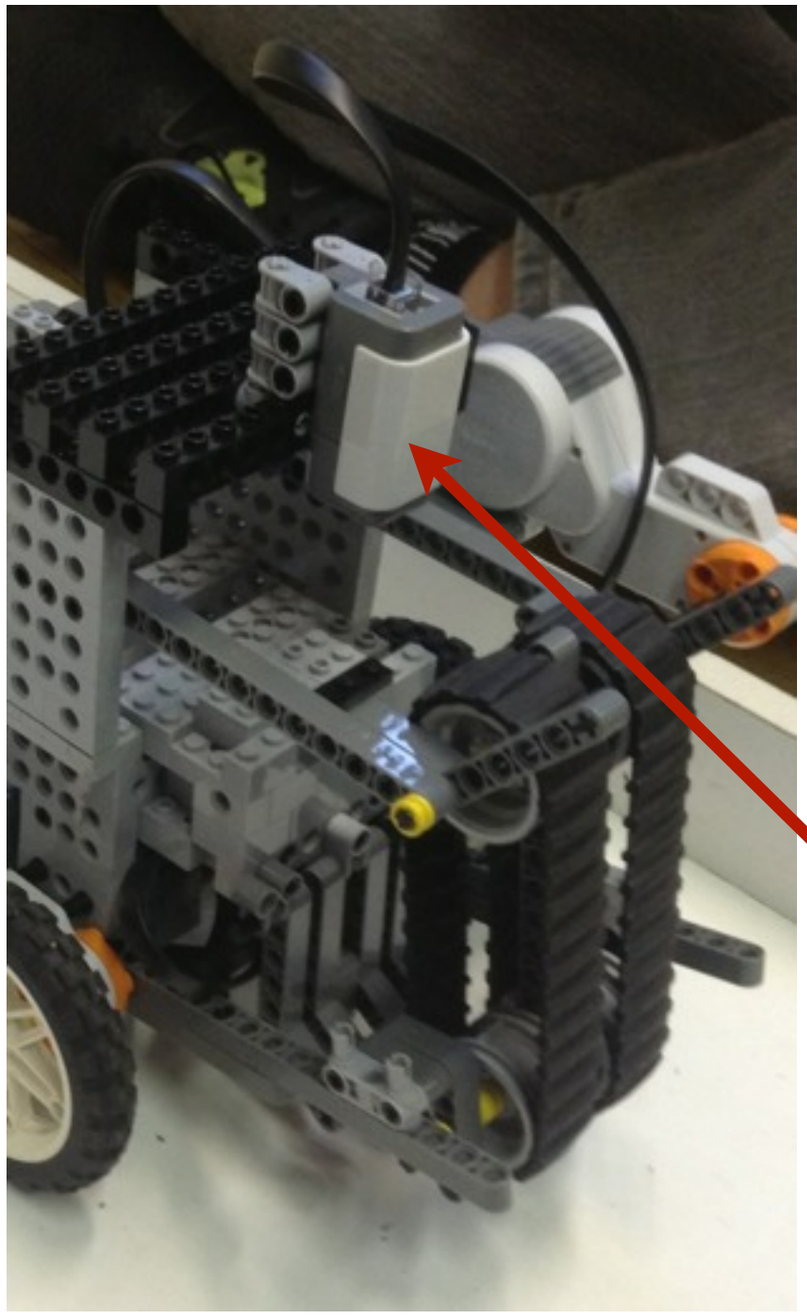
- This solution is heavily dependent on reliability of control subroutines (turning, distance measuring, ...)
- We have therefore focused on each subroutine separately to try to make it as good as possible
- Each member has done research on different control subroutines

Ball Counting



- First approach using light sensor
- Change of reading values = ball collected

Ball Counting

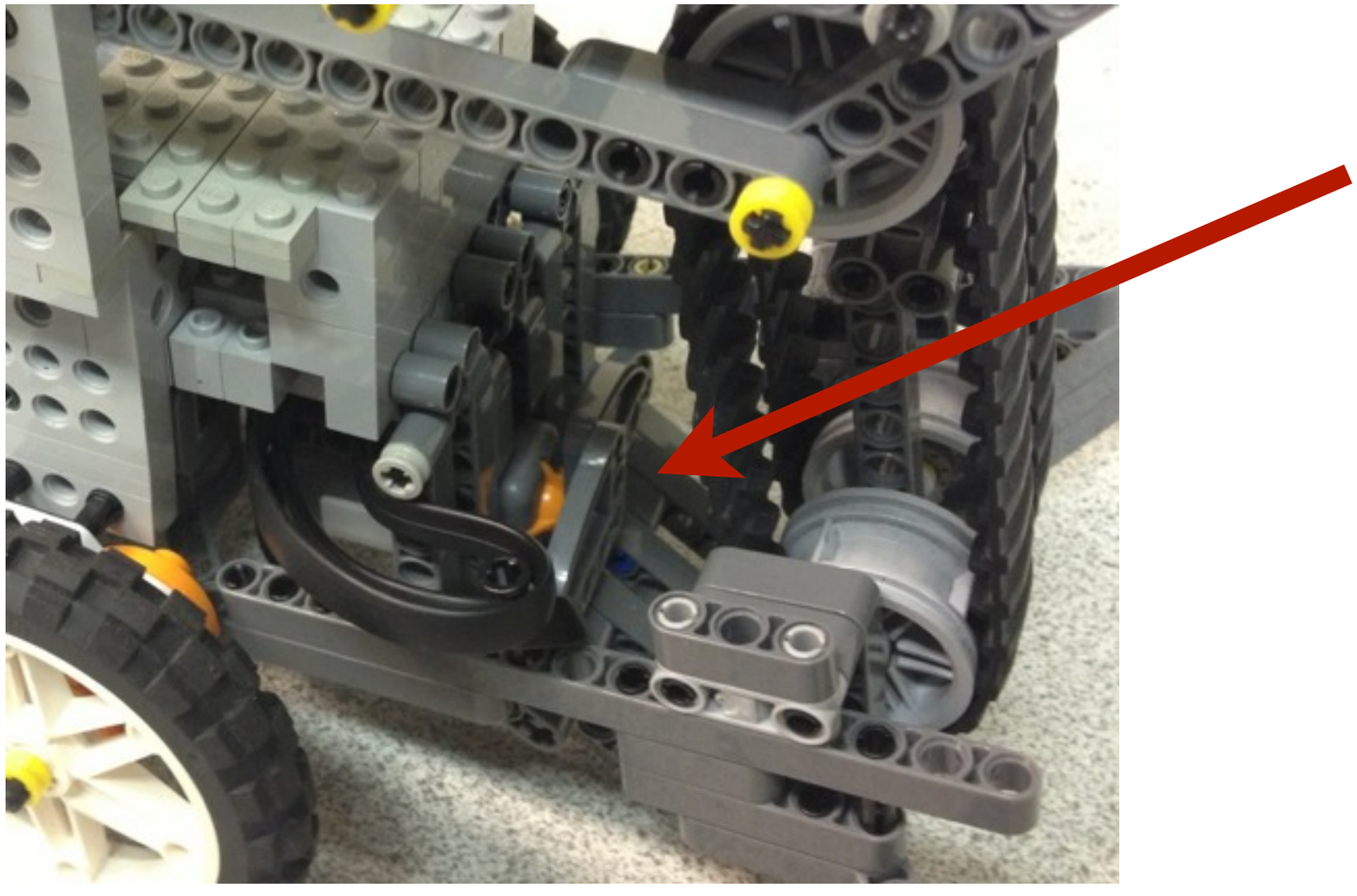


- First approach using light sensor
- Change of reading values = ball collected
- Unreliable (shadows, lights on/off, etc.)

Ball Counting

- Second approach using touch sensor
- When sensor touched - ball counter incremented and then sensor waits for a while
- Almost 100% reliable
- Recognises balls sooner

Ball Counting



Ball Counting: Limitations

- In unlikely event of ball getting stuck, stops working properly
- Even though the collected ball is being recognised sooner than when using light sensor, there is still short pause (~300 ms) after a ball is caught by the belt and before the sensor identifies the ball

Ball Unloading

- Simple use of kinetic energy to push the robot against the container wall
- This causes the barrier to move and release the balls
- (Touch sensor used to sense when the barrier has been opened)
- Repeat

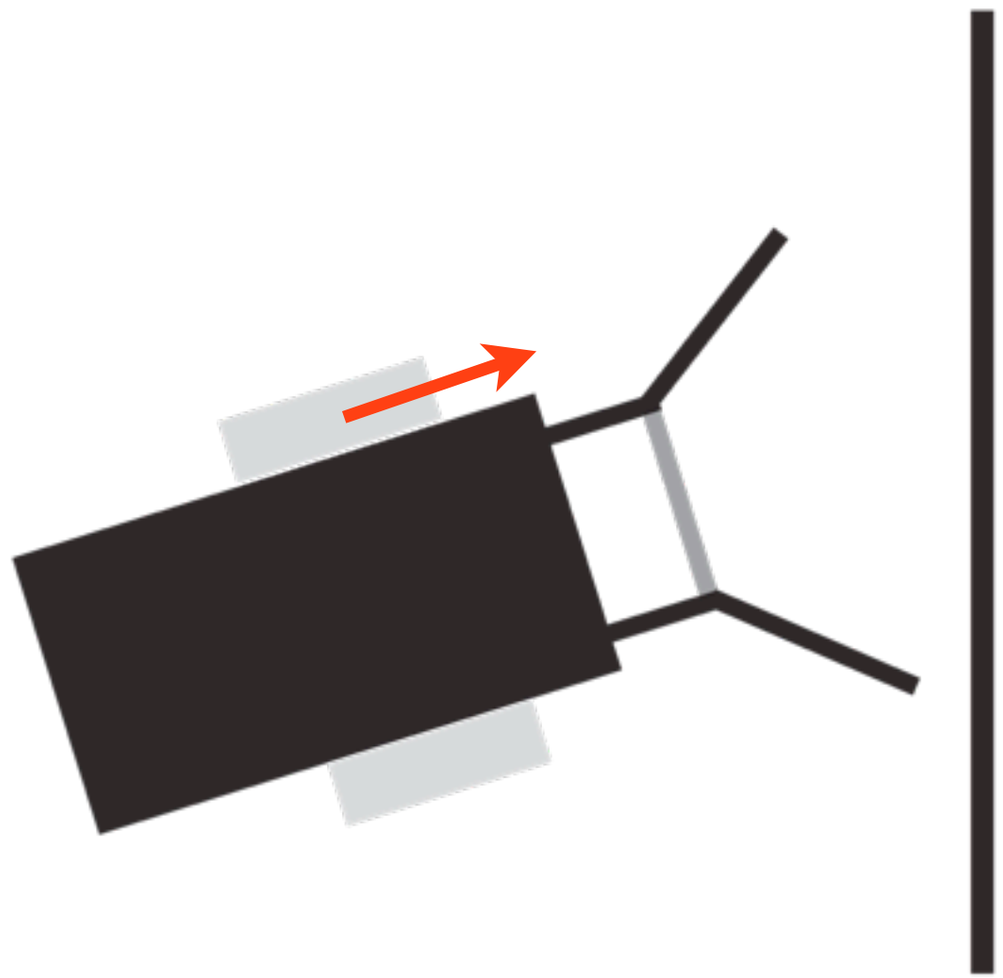
Ball Unloading: Limitations

- The mechanism relies on rubber bands to bring the barrier back in place - when both rubber bands break, the mechanism stops working
- When the robot is too fast, the barrier can slide over the wall and the mechanism won't work

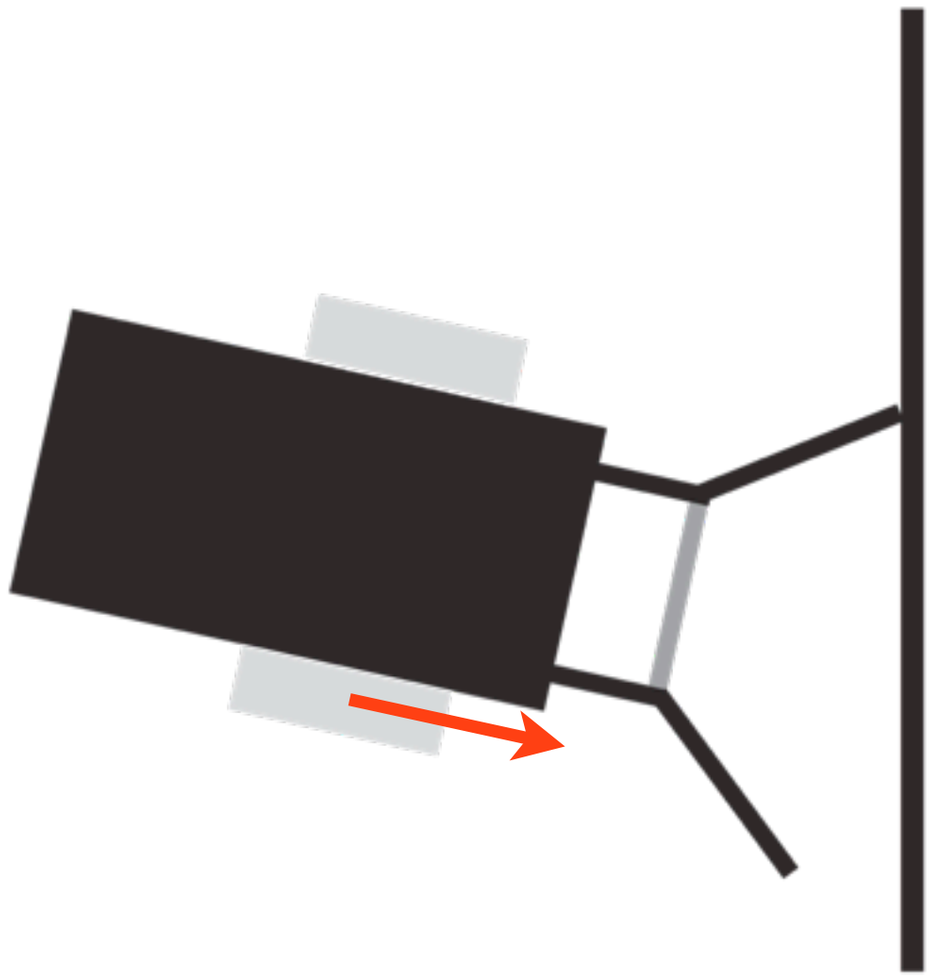
Wall Aligning

- Used to fix position and orientation deviation during the preprogrammed route
  - Left wheel turned on for a while, then stopped
  - Right wheel turned on for a while, then stopped
  - This results in the robot aligned to a wall

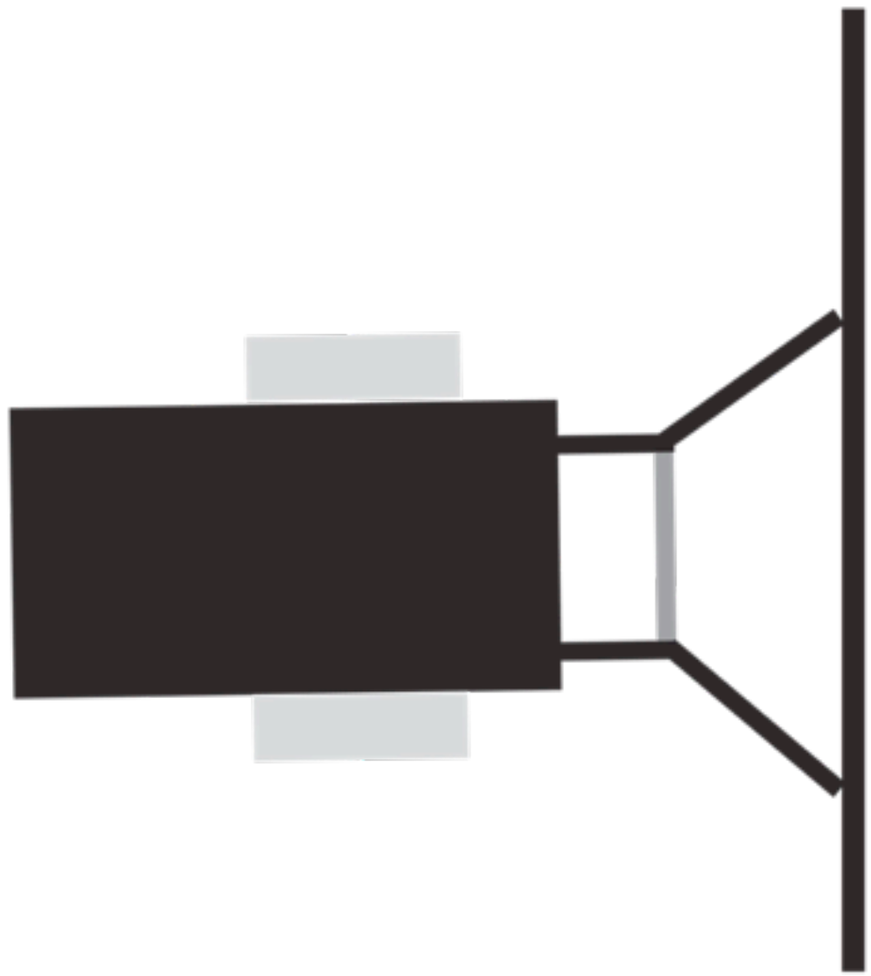
Wall Aligning



Wall Aligning



Wall Aligning



Wall Aligning: Limitations

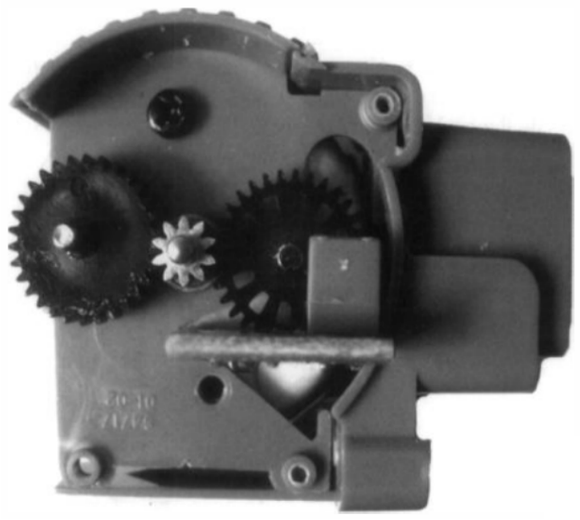
- Time consuming - when not necessary only wastes time
- Works only against a flat wall
- If the robot (by some error) is not close to a wall, does not work properly

Tomas Vitek: Results Achieved

- Designed and developed quite reliable ways how to *Collect*, *Count* and *Unload* balls.
- Helped to implement and bring together solutions from other members into the final program.
- Led the team, made decisions what and when should be done, organised all meetings as well as made sure that our work was on schedule.

Motor Synchronisation

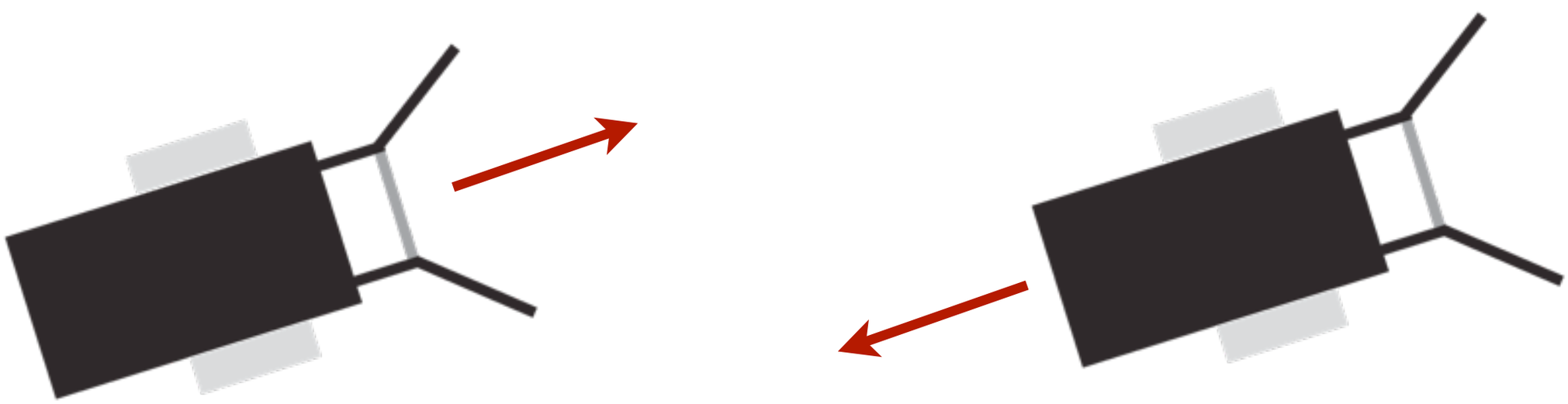
- Each LEGO Motor includes a built in encoder that can be used to program the robot to move around in a very precise manner.
- The encoders or rotation sensors in the motors can be monitored by the NXT program with the idea being that when one wheel falls behind the speed of the other will be adjusted to compensate causing the robot to move in a straight path regardless of terrain.



Motor Synchronisation

Use of motor syncing in our robot:

- Traveling in a straight path when going both forward and in reverse

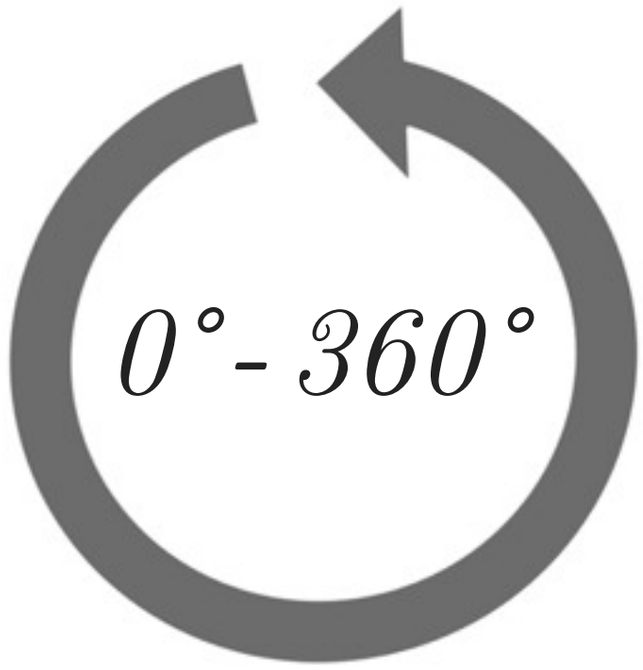


Motor Synchronisation: Limitations

- Works properly only when both wheels are touching the surface
- Becomes less accurate when the battery levels are lower (however still OK)

Turning

- We have implemented a function which allows the robot to turn any degree of angle.
- The robot uses a turning method which allows it to turn on the spot by rotating one wheel certain degree and rotating the other wheel in the other direction.



Turning: Limitations

- Does need to be calibrated on different surfaces
- Accuracy dependant on battery levels
- It is not 100% reliable because of dirt and uneven friction on a surface

Olu Adebari:
Results Achieved

- Provided support to the team as the team secretary. Documented all work in minutes from every meeting.
- *Motor Synchronisation* is very important part our solution
- Worked together with others to improve the hardware design of the robot

Ball Searching

- Utilizes Ultrasonic Sensor (Sonar)
- Records distance between robot and objects at set intervals
- Sensor stores the closest distance within a 5 second window

Ball Searching: Methodology

- Variables are initialised for recording distance.
- Robot spins while recording distance at half-second intervals.
- Records minimum distance measured then proceeds to approach nearest object.
- Subsequent runs of ball searching can be made to ensure robot travels in the correct direction.

Ball Searching: Limitations

- The NXT US (ultrasonic sensor) has a range of 255 cm and accuracy within 3 cm.
- The US cannot accurately detect small spherical objects via sonar due to the shape.
- The US requires precise positioning.
- The US fails to differentiate between objects (i.e. wall and ball)

Moving Accurate Distance

- Utilizes motor rotation in conjunction with circumference of wheel as constant.
- Can travel distance input in centimetres relatively accurately.

```
int degrees = 360 * distance * 10 / WHEEL_CIRCUMFERENCE;
RotateMotorEx(BOTH_MOTORS, FORWARD_SPEED_MEDIUM, degrees, 0, true, true);
```

Moving Accurate Distance: Limitations

- Total battery power greatly affects accuracy of the action.
- Motors stall if robot is unable to completely travel total distance that was input.

Josh Manuncia: Results Achieved

- US Sensor was far too unreliable when measuring distance between robot and squash balls.
- Decided not to use *Ball Searching* and simply rely on pre-planned route.
- *Moving Accurate Distance* was very successful and features regularly in our coding.

Team: Results Achieved

- We have done quite a lot of research and have been able to find appropriate methods for our project
- We were able to design and build very reliable hardware solution for the given task
- Our solution is quite fast and has a low error occurrence
- We have over the time managed to develop very good team coordination and we're able to work very well together
- We have been able to keep up with the set schedule as well as record meeting minutes for every single week

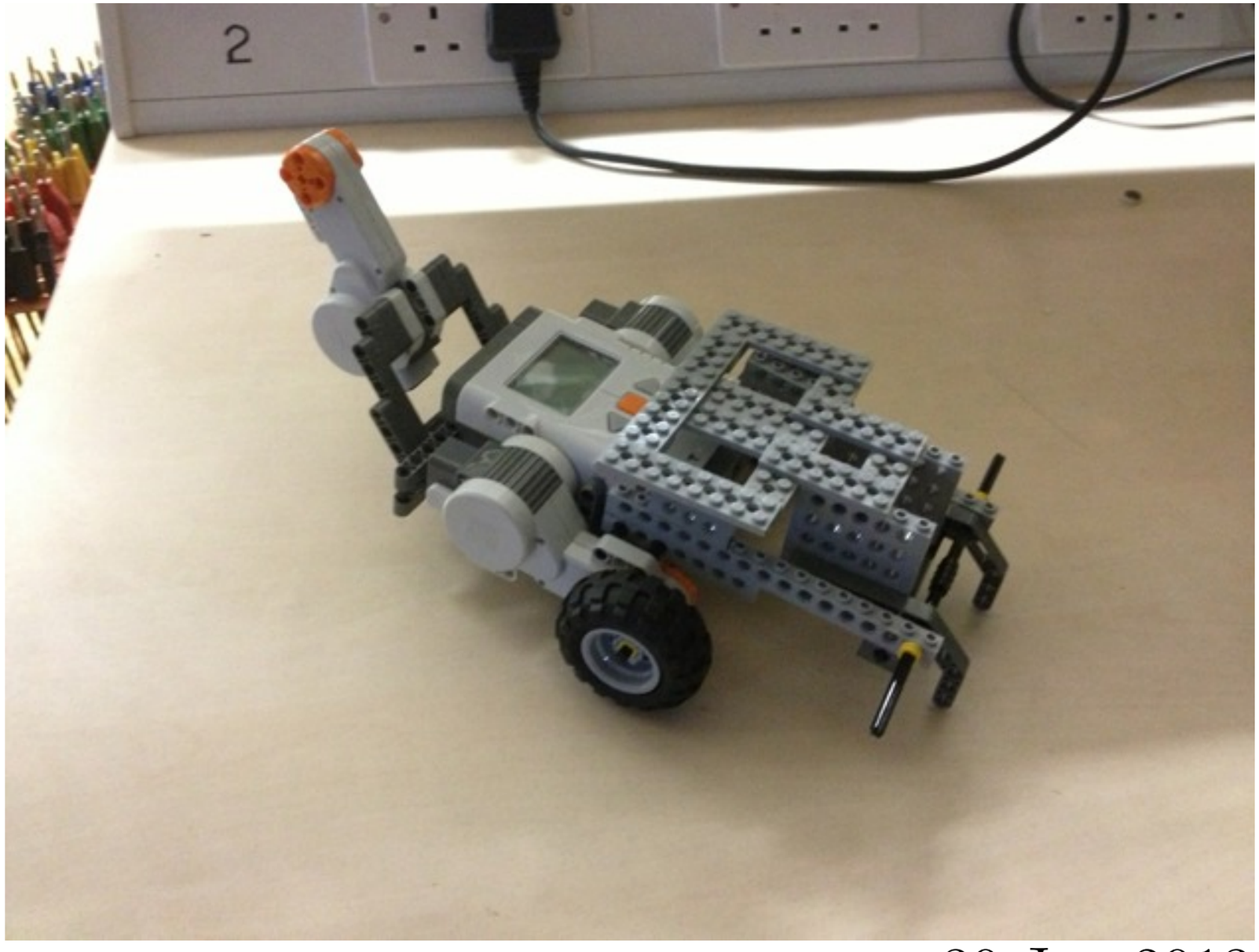
Future Work

- *Developing adaptive software for our robot:*
  - Incorporating additional sensors, which we couldn't use *(such as compass, combination of more sonar sensors)*
  - Creating mapping subroutines to be able to collect balls even on unknown maps
- *Adding ability to communicate and cooperate with similar ball collecting robots*

Hardware Design:

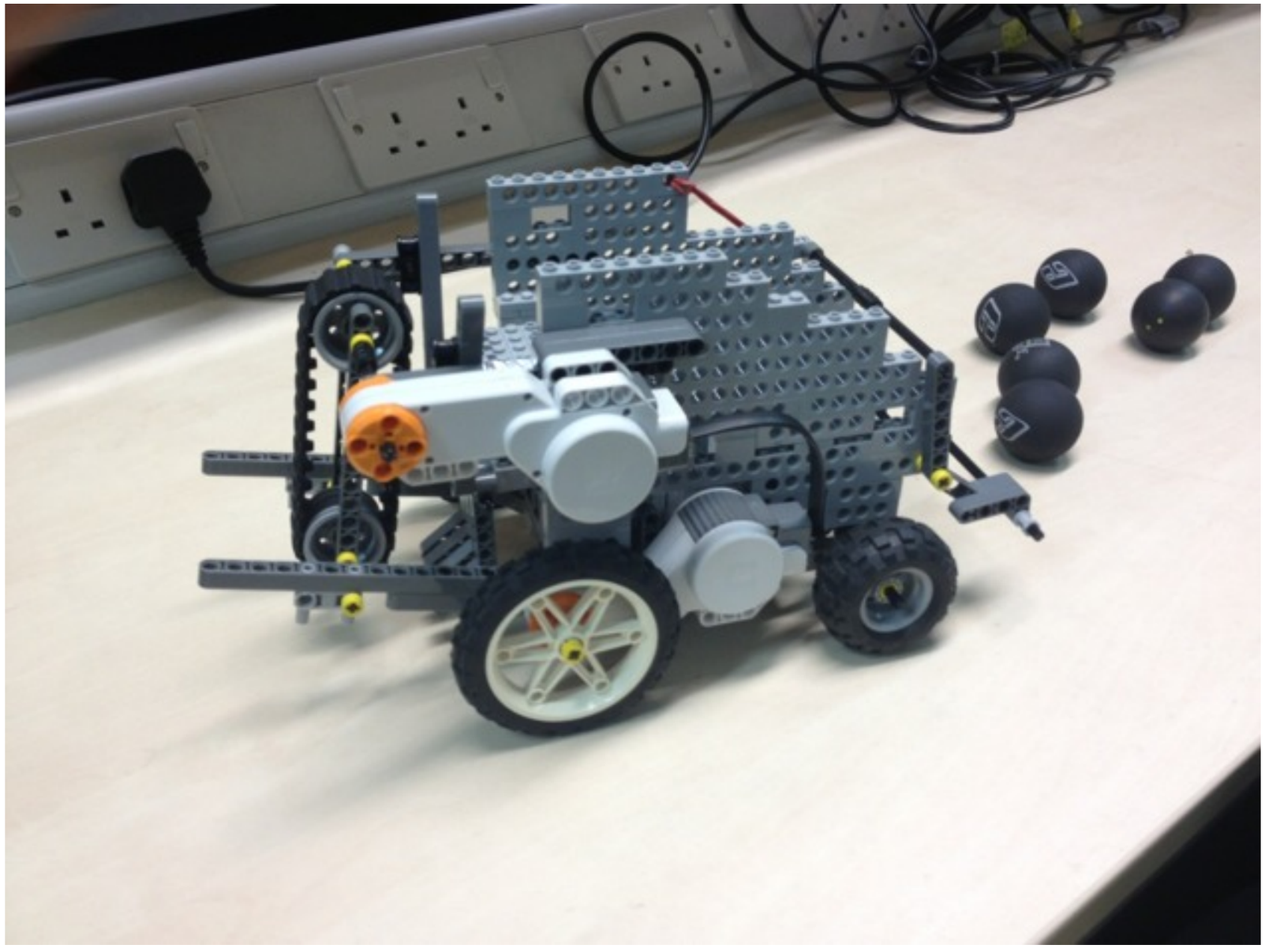
Time-lapse

Hardware Design



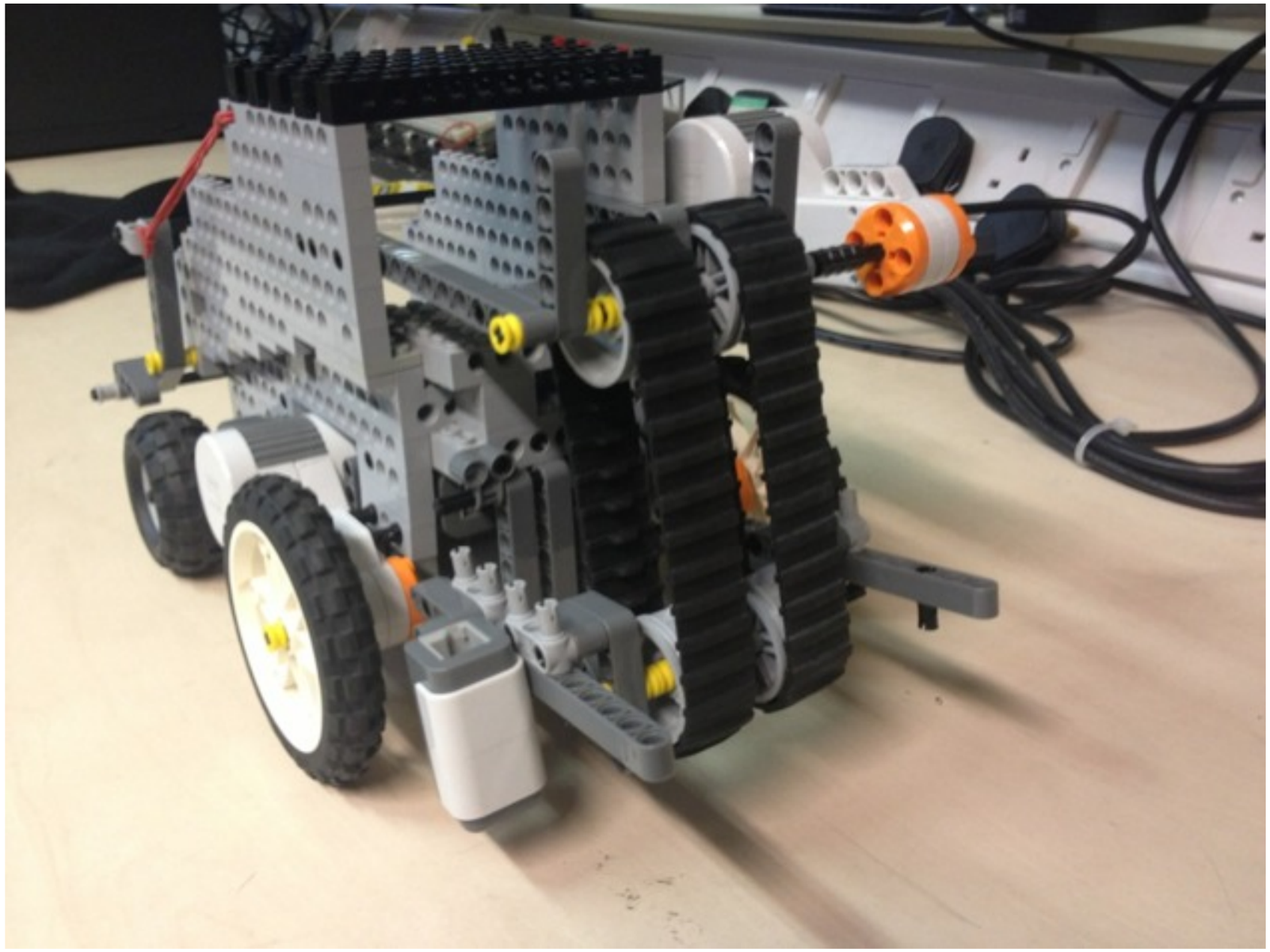
30 Jan 2013

Hardware Design



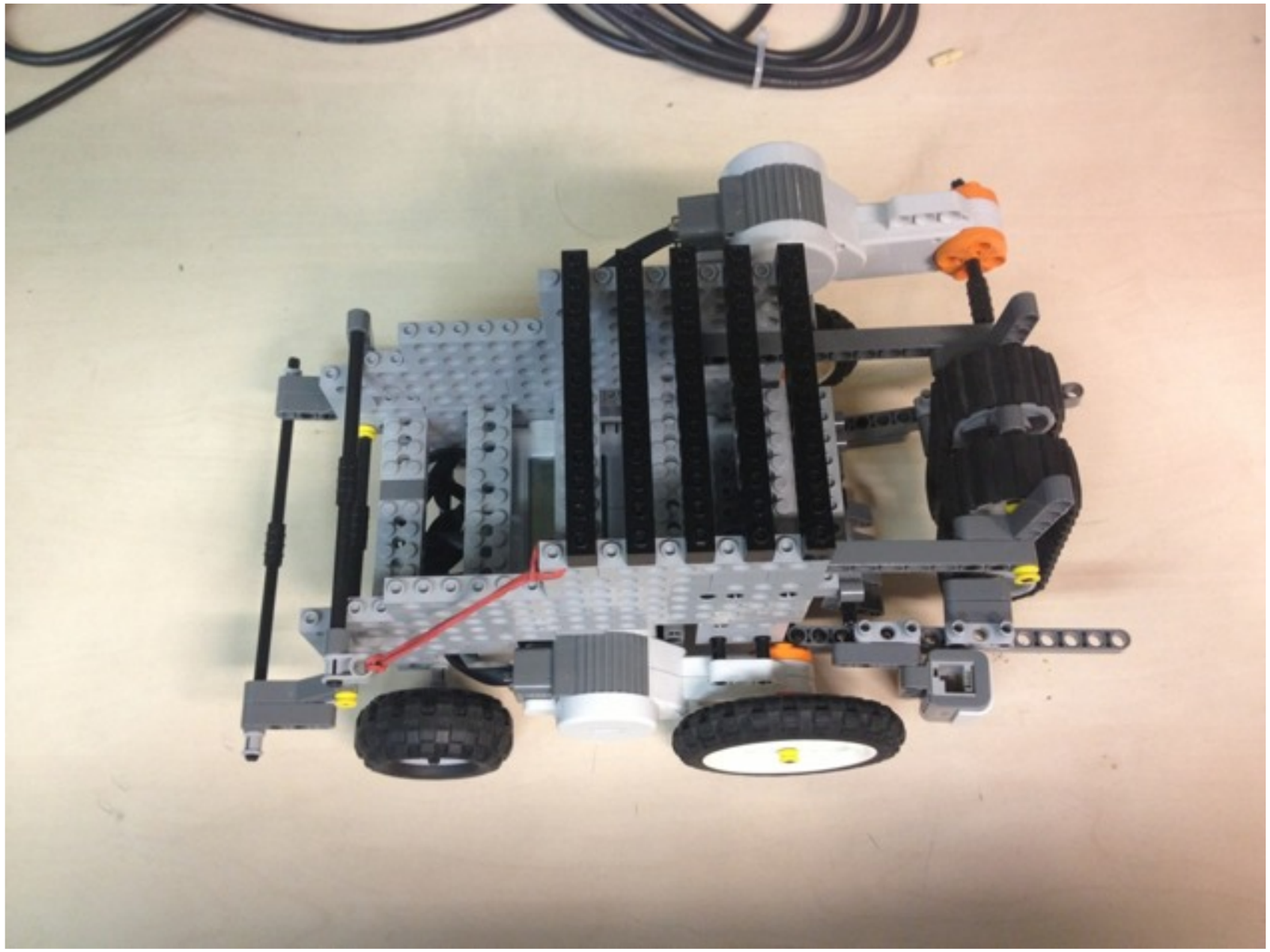
5 Feb 2013

Hardware Design



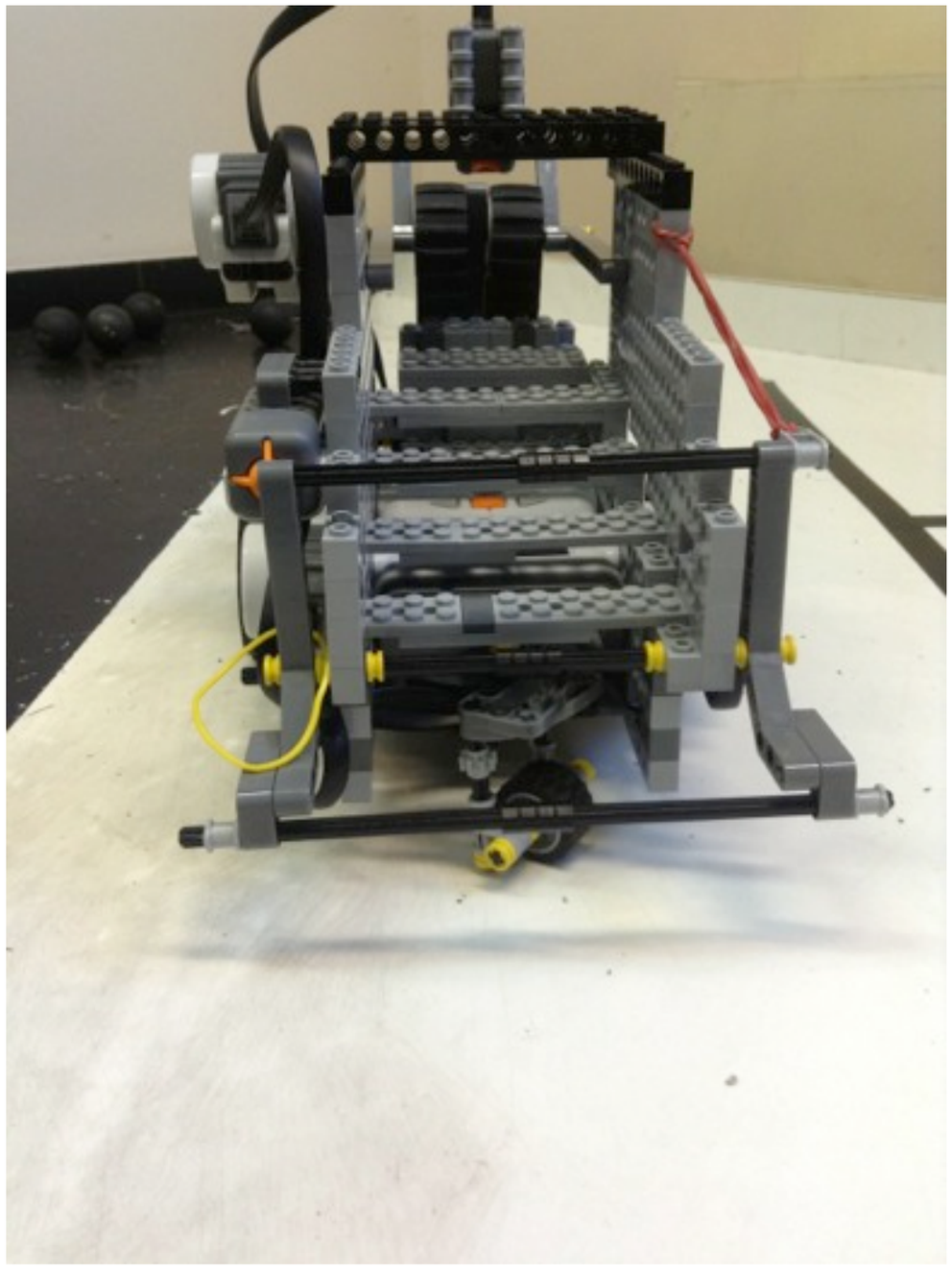
6 Feb 2013

Hardware Design



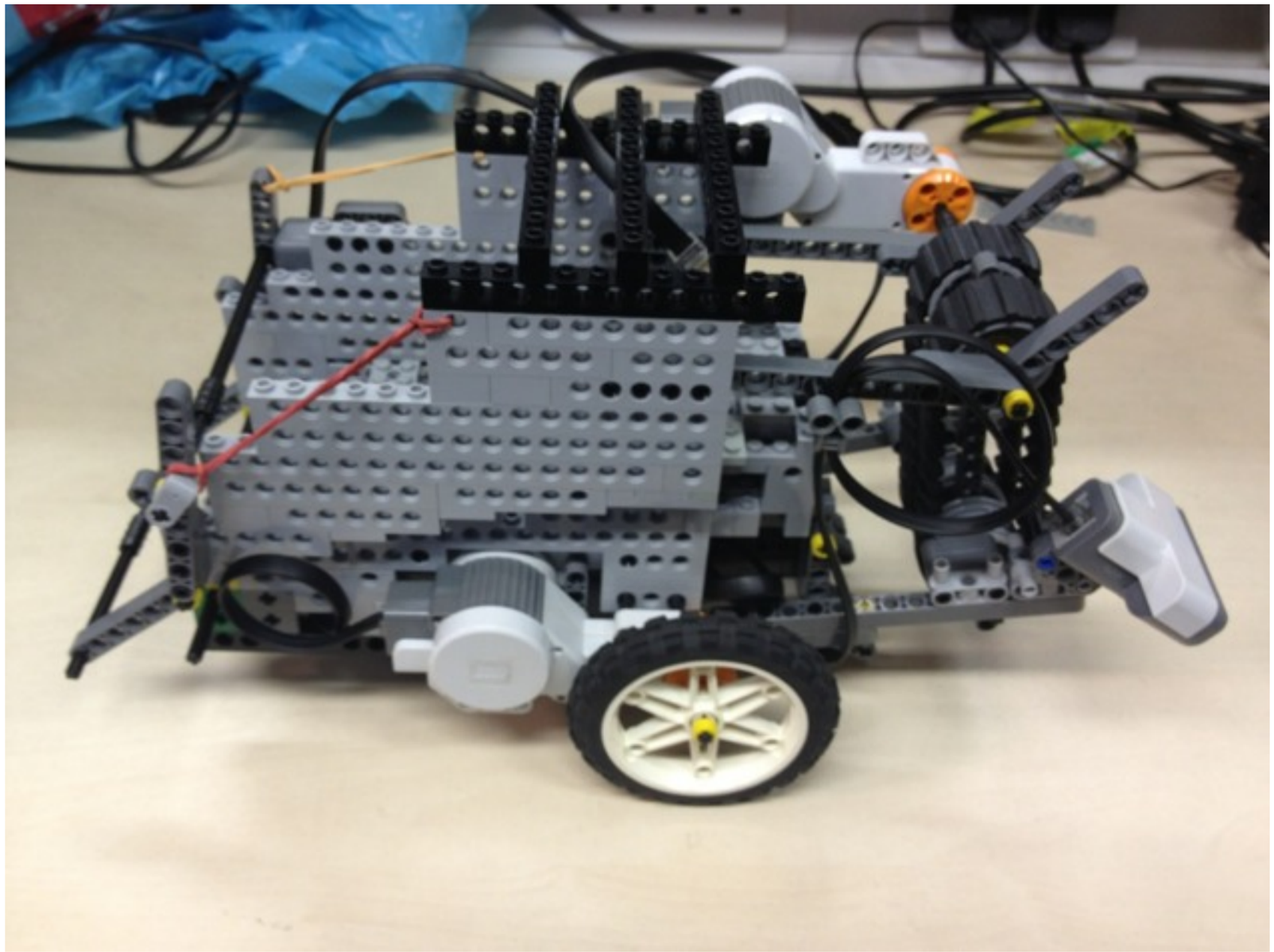
6 Feb 2013

Hardware Design



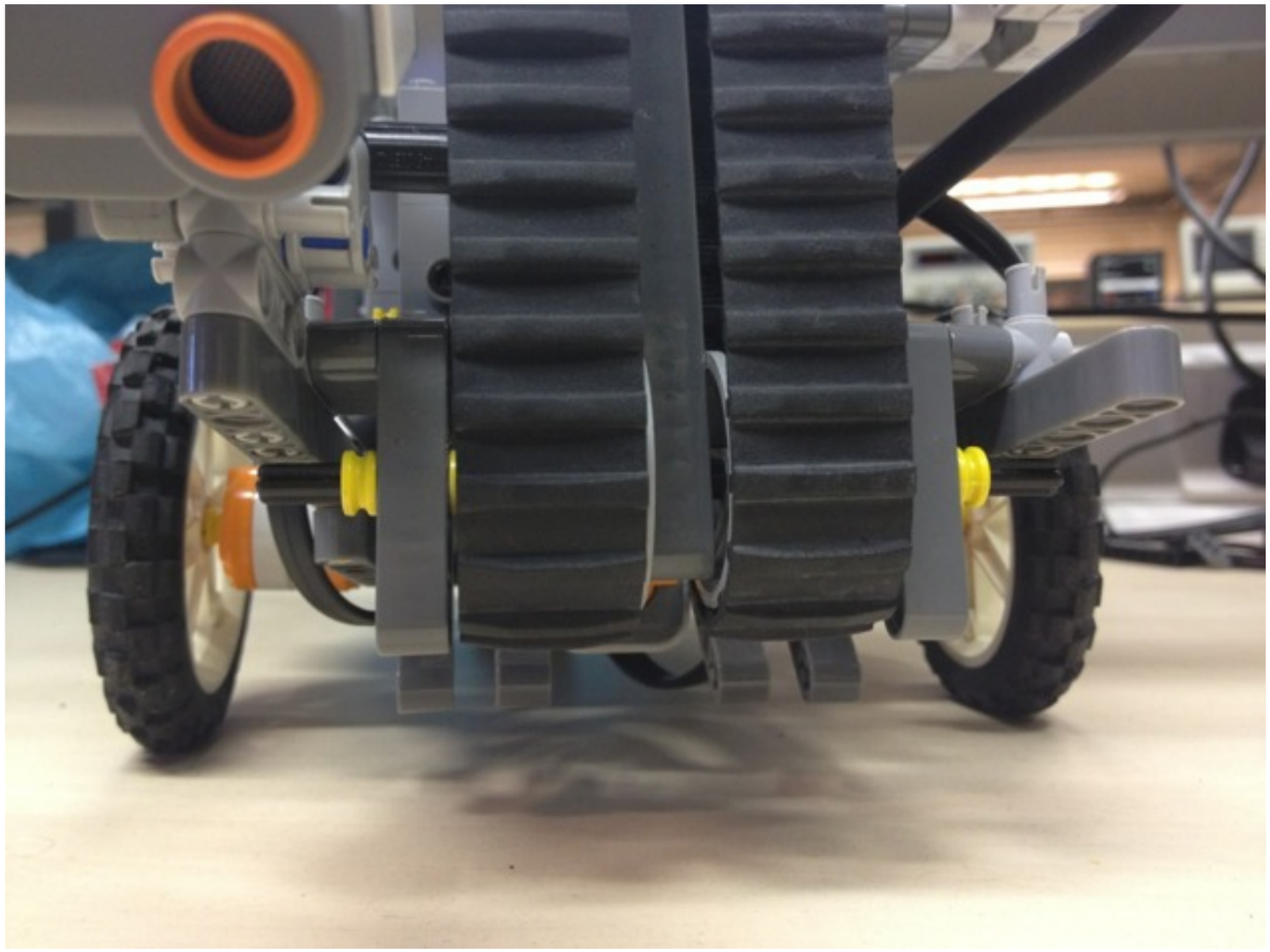
19 Feb 2013

Hardware Design



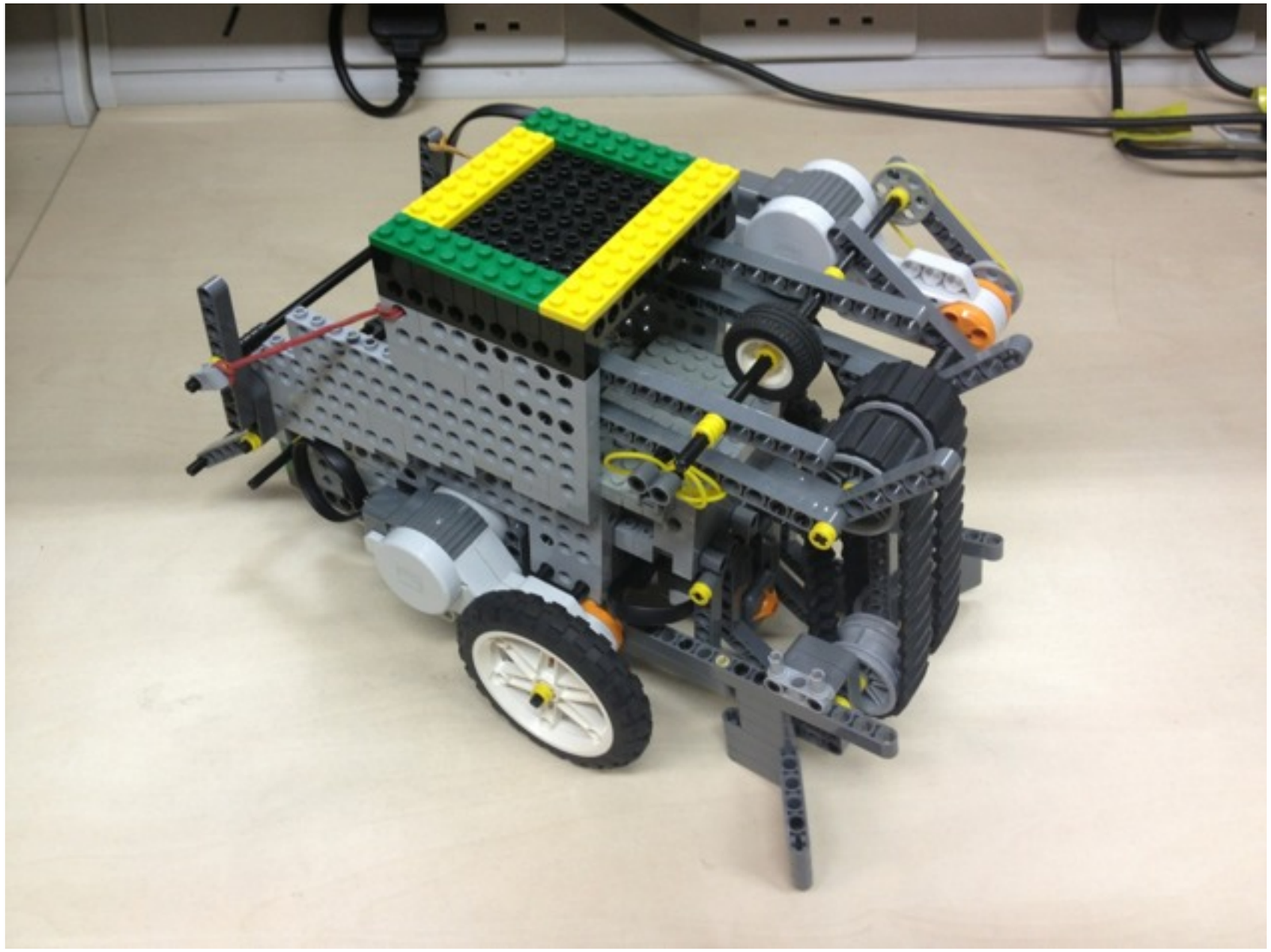
5 Mar 2013

Hardware Design



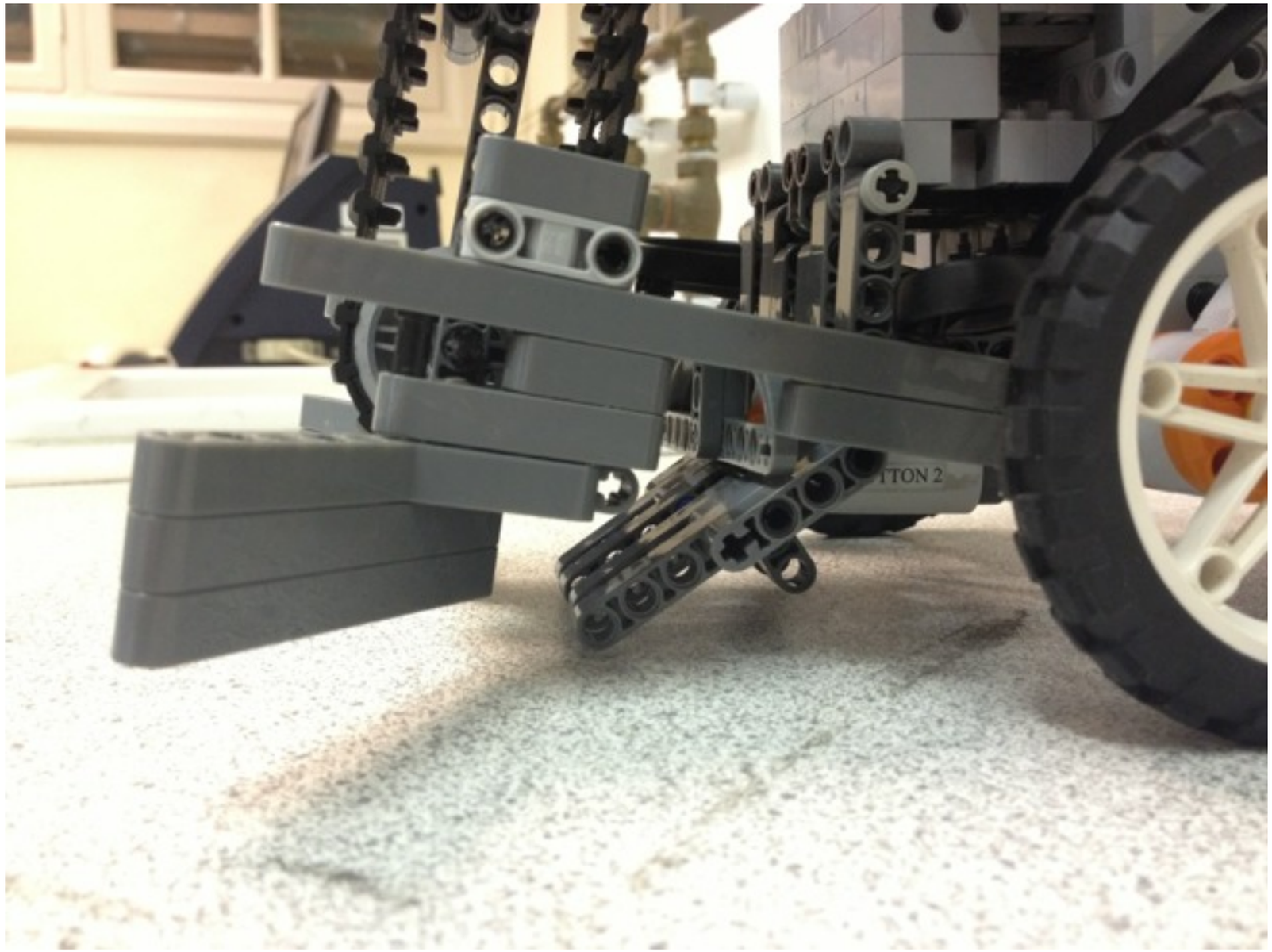
5 Mar 2013

Hardware Design



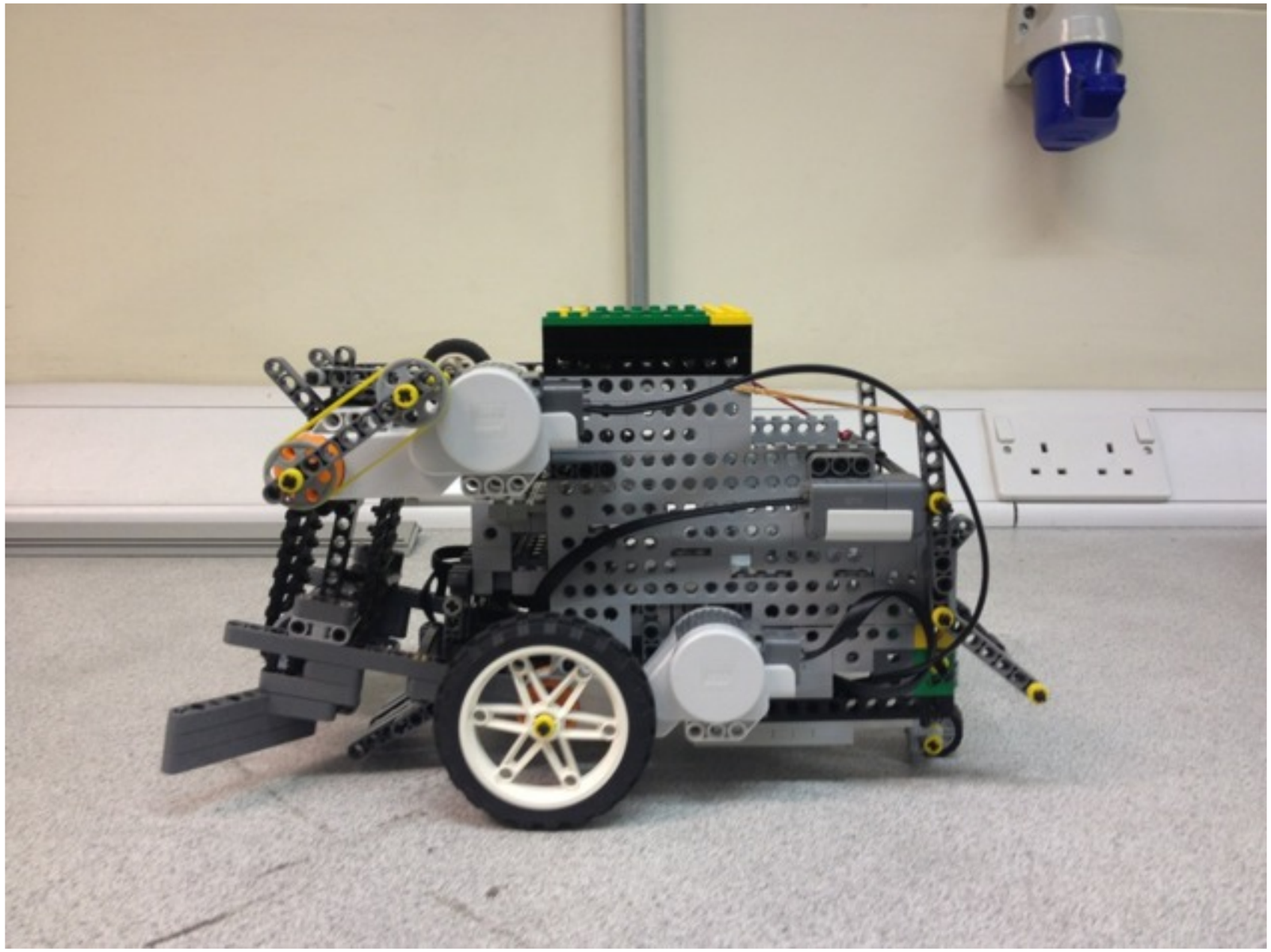
7 Mar 2013

Hardware Design



7 Mar 2013

Hardware Design



7 Mar 2013

Thank you for your attention.