Group Robotics project

Marek Kujawa, Jack Rome and Shayne Shaw

Contents

[Introduction & Objectives 2](#_Toc481106787)

[System Elements 3](#_Toc481106788)

[Chassis 3](#_Toc481106789)

[Version 1.0 3](#_Toc481106790)

[Version 1.1 3](#_Toc481106791)

[Version 2.0 3](#_Toc481106792)

[Version 3.0 4](#_Toc481106793)

[Pin allocations 4](#_Toc481106794)

[Motor Shield 4](#_Toc481106795)

[Software 5](#_Toc481106796)

[Final Performances 6](#_Toc481106797)

[Light Following 6](#_Toc481106798)

[Line Following 6](#_Toc481106799)

[Collision Avoidance 6](#_Toc481106800)

[Conclusions 7](#_Toc481106801)

[Light Following 7](#_Toc481106802)

[Line Following 7](#_Toc481106803)

[Collision Avoidance 7](#_Toc481106804)

[Overall 8](#_Toc481106805)

[Appendix 9](#_Toc481106806)

[Wiring Guide 9](#_Toc481106807)

[Chassis Measurements 10](#_Toc481106808)

# Introduction & Objectives

The aim of the project was to construct and code a small autonomous robot that consisted of three sub-systems. This report is a collaborative discussion of the three sub-systems as well as the construction, and design choices, of the entire robot.

The first sub-system was the light following. The aim for this was to implement a light following system based around 2 light dependent resistors which would enable the robot to navigate autonomously towards the brightest source of light that it was exposed to.

The next sub-system was constructed to follow a line track marked with black tape on the floor as reliably and as fast as possible. This will be achieved using a circuit of three IR sensors soldered onto a Veroboard piece.

The final sub-system was collision avoidance. The aim of this sub-system is to provide the robot with the ability to autonomously roam without crashing into any obstacles. This is achieved by using an ultrasonic distance sensor as the only input to the Arduino board.

Background theory on the inner workings of the sub-system circuits, the physics supporting the sensors are given as well as explanations of the programs used by the Arduino board to control each of the sub-systems. This report will also include discussions on the experimental design of the robot and the steps taken to achieve the final design.

# System Elements

## Chassis

### Version 1.0

The vehicle was built around the original single layer chassis which was provided. Once the motors had been attached the battery was added to the top of the chassis using blue tack to form a stable, but impermanent, mounting. The Arduino and breadboard were then mounted on top of the battery, also using blue tack. While this was not ideal it did prove a stable configuration for sensor testing as well as allowing for further testing with alternative chassis layouts for each of the components.

### Version 1.1

It was determined that while the LDR’s could be tested using purely their own wires as a means of support a more stable mounting would be needed for further testing. As such a basic mounting was created using perforated metal strips to hold the LDR’s more securely and to maintain a constant distance between the sensors. The Arduino and breadboard also needed a more secure attachment and this was also built using perforated metal strips to suspend the two components on a board above the battery. The added advantage of this design was that the battery could be moved or changed without disturbing the boards as well as the struts serving as a stable mounting for the power regulation board. Lastly an underslung mounting was also created for the infra-red sensors again using the perforated metal strips for the support struts as well as Veroboard to which the sensors and associated resistors were themselves attached.

### Version 2.0

While version 1.1 did resolve some issues, it resulted in untidy wiring and support structures which could not be expanded upon. It was decided that using standardized mountings would be the best solution to this issue. A mounting kit was purchased and the parts were incorporated into the chassis build. A clear layer of Perspex was also cut to size and, using the mountings, a second layer was added to the chassis. All other support struts which were previously made from perforated metal strips were also replaced with standardized mountings. Where necessary these we plastic to ensure that no short circuits were created where they came into contact with Veroboard.

This new layered design allowed for most of the wiring from the sensors to be routed under the second layer next to the battery creating a neater more compact system which prevented the wires from accidental snags during testing and/or handling. At this stage the mounting for the ultrasonic sensor was also added. This was composed of a Veroboard base to which the sensor and necessary wires were soldered.

During testing of the collision avoidance, it was observed that the robot regularly rammed into objects at full speed. To prevent damage to the sensors pieces of the perforated metal strips (salvaged from the version 1.1 design) were attached to the front of the vehicle. The force of the impact could now be borne by the chassis and not the more delicate sensors and wiring.

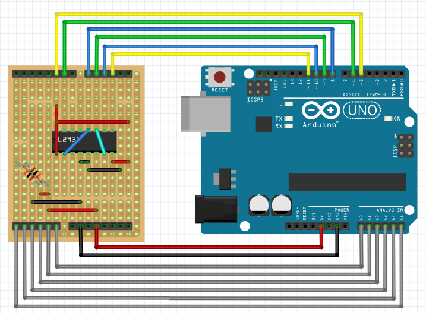
### Version 3.0

The multi layered approach used for version 2.0 was expanded upon for the third and final variant of the chassis. A third layer was added to the top of the forward section. It had a dual purpose. The first was to act as a mounting space for the newly acquired LCD display as well as to replace the sensor protectors used in version 2.0. Although this final variant was significantly higher than the original it’s centre of gravity was kept low due to the position of the battery on the original 1st layer. By comparison it contributed more mass on its own than the rest of the additional components combined.

## Pin allocations

The pin allocations for the different inputs and outputs for the system were reorganized so that there would be easier with wiring the hardware. Details of this new configuration can be found in the appendix.

## Motor Shield



The initial approach for the project was to use a breadboard as it is perfect for prototyping and testing. However, breadboards are best suited for simple circuits and, as the project grew, the network of connections quickly became dense and complicated making it difficult to trace wires and debug the system.

This is why the idea of creating a purpose-built shield appeared as a perfect solution. A shield, in Arduino jargon (also Raspberry Pi) means an extension module board that plugs into GPIO Interface of Arduino. There are many Arduino shields available. They come in different sizes, functionality and purposes. They also have the following advantages:

* small size
* lower amount and length of wires
* quicker installation and uninstallation
* smaller chance of breaking connections (most connections are solid)

The shield was made with a piece of Veroboard (also known as stripboard). In order to transfer the circuits from breadboard to Veroboard-based shield, the connection diagram was drawn. Then, it was carefully analysed and matched against GPIO pins in Arduino. It was critical to assign the correct types of pins as different components accept different signal types. For instance, the enable pins of H-bridge had to be connected to PWM pin. The process of building and testing the shield exposed a few issues with its design. One of them was regarding the pins connections. As the used Veroboard was single-sided, the connection pins could be installed only on one side. This made sensor and motor inputs difficult to incorporate into the board. A solution to this could be to use longer pin sockets that go through the Veroboard serving as sockets for sensors or motors inputs on the top side and as the pins for Arduino GPIO connection on the other side. Additionally, it would let a user add another shield on the top.

Another problem comes from the Veroboard construction. It provides a way to create connection only along a line either vertically or horizontally. Those lines often have to be broken and occasionally an additional wire is needed to provide a connection with a different component.

As an enhancement h-bridge could be connected via a socked instead of being soldered directly onto the board. This would allow for its replacement in case of a failure.

## Software

The Arduino programming environment was used to code the ‘Arduino Uno’ board that was supplied. The version that was used was 1.6.8. The default set of libraries were used for all of the sub-systems and the LCD library was added so that the LCD screen could be used for the line following and the collision avoidance sub-system. This environment was also used for testing the inputs of the sensors using the ‘Serial Monitor’.

## Sensors

# Final Performances

## Light Following

The light following sub-system performed as expected. The robot was able to navigate towards the strongest light source as required and, therefore, a light source was able to be used to guide the robot around the set tracks on the floor. The robot was adequately able to navigate both the round course and the sharp corners of the rectangular course.

Once moving, however, the sub-system performed as mentioned previously and it was found that the sub-system was better controlled without a blinder between the LDRs.

## Line Following

The line following sub-system met its requirements. In practice, the robot was able to very accurately follow the line track on the ground. This included both the round track and the sharp-cornered track. Although very accurate, the robot did not complete a lap of each track in an impressive time.

## Collision Avoidance

The collision avoidance sub-system performed very well and achieved its aims. The robot was able to roam around the lab completely autonomously and could be left unmonitored for a while and still not have collided with any object.

However, there was a problem with the different performances of the motors. The robot would often drift to the right when a straight path was desired. An attempt to correct this was by adjusting the speed of the left motor slightly and having the speed set to as close to maximum as possible. This was an improvement but a slight drift would occur if the robot travelled a long path.

# Conclusions

## Light Following

The robot light following subsystem did perform to the desired specifications. The start sequence was not activated in ambient light and the robot only started to move and follow the light source after it was directed onto the sensors. Once the start sequence had been activated the robot was successfully guided around the track, performing well on the sharp turns when required and never losing acquisition of the light source that was being used to guide it.

The light following sub-system could be improved if the characterisation of the sensors was better implemented. Testing the LDR circuits would be more accurate if a calibrated light source was used. The sensors could then be tested under identical conditions and a response curve plotted for each sensor individually. From the curve an equation could be derived which would describe each sensors response in detail throughout its range.

In addition, more sensors could be used to increase light sensitivity and/or increase field of view.

## Line Following

The line following sub-system completed a lap the tape courses with impressive accuracy. Unfortunately, this accuracy was at the sacrifice of speed as the robot completed a lap of the courses in a rather unimpressive time. This is because the robot would often overcompensate any error and, therefore, would take longer than necessary to get back on track. It is believed that a better pace could be achieved with more amendments to the line following code.

The sensors worked without a problem during the performance. After judging the optimal spacing between the sensors and the ground, and the position of the IR sensors, the Arduino board was able to receive perfect readings from the sensors.

This sub-system was difficult to program to implement due to the fact that a vast number of variables need to be considered. For example, motor speed, angle of robot, angle and shape of the track ahead, etc. It became evident that this situation became increasingly difficult to gauge simply because it was moving too fast to analyse.

To improve the code, then the behaviour of the robot in this type of scenario is to be studied in more depth. This will give a better understanding of what the correct responses should be on a given place on the track.

## Collision Avoidance

Overall, the collision avoidance sub-system performed very well and achieved its aims. The robot was able to roam around the lab completely autonomously and could be left unmonitored for a while and still not have collided with any object.

However, there was a problem with the sensor not detecting narrow objects. The sensor’s user manual recommends that the surface of the object in range should be more than 0.5m2. Therefore, the robot, as mentioned above, often collided with the legs of the laboratory stools.

The fact that there was only one sensor was problematic. This meant that the robot would often have trouble with objects that were out with its 30O range and, subsequently, would sometimes collide with objects while turning. This could be solved by adding more sensors in different positions on the robot. These could then be used to obtain a better decision on the best direction to travel.

Alternatively, a different improvement that could be made with this sub-system would be the inclusion of a servo motor. This would provide the ability to scan an area and select the best path rather than the robot’s current method of rotating until the first free path is found. In this scenario, the sensor would need to be moved to the top of the robot, with the Veroboard it is soldered on being mounted onto the servo motor.

## Overall

Overall, the robot met all of its given objectives. The shield constructed for the Arduino board made it easier to switch between the three configurations and the LCD screen allowed for feedback data whilst testing the different sub-systems.

The plate plastic helped to organise some of the wiring as it allowed for the groups of wires from the sensory components to be fed through the robot, between the first layer and the added second layer of plastic, and to come out the back to be plugged into the Arduino shield when needed.

To improve the robot’s performance in all of the sub-systems, the best solution would be to replace the wheel setup. The current motors are cheap and so may have some differences subtle between them (as mentioned in the collision avoidance sub-system). This could be corrected by either replacing the motors or by compensating for the difference using resistor circuits.

Another way to improve the driving of the robot is by replacing the round wheels with tacked wheels. This would make turning easier and would avoid the skidding/slipping on the floor of the lab by increasing friction.

# Appendix

## Wiring Guide

The wires were grouped/bundled by component and the colour coding of these groupings are as follows:

Motors

1. Red x 2 - live
2. Black x 2 - ground

LDR’s

1. Red x 2 – live
2. Black x 1 – Common ground for both resistors

Infra-red

1. Red x 1 – common live
2. Black x 1 – common ground
3. Green x 1 – right sensor
4. Orange x 1 – middle sensor
5. Blue x 1 – left sensor

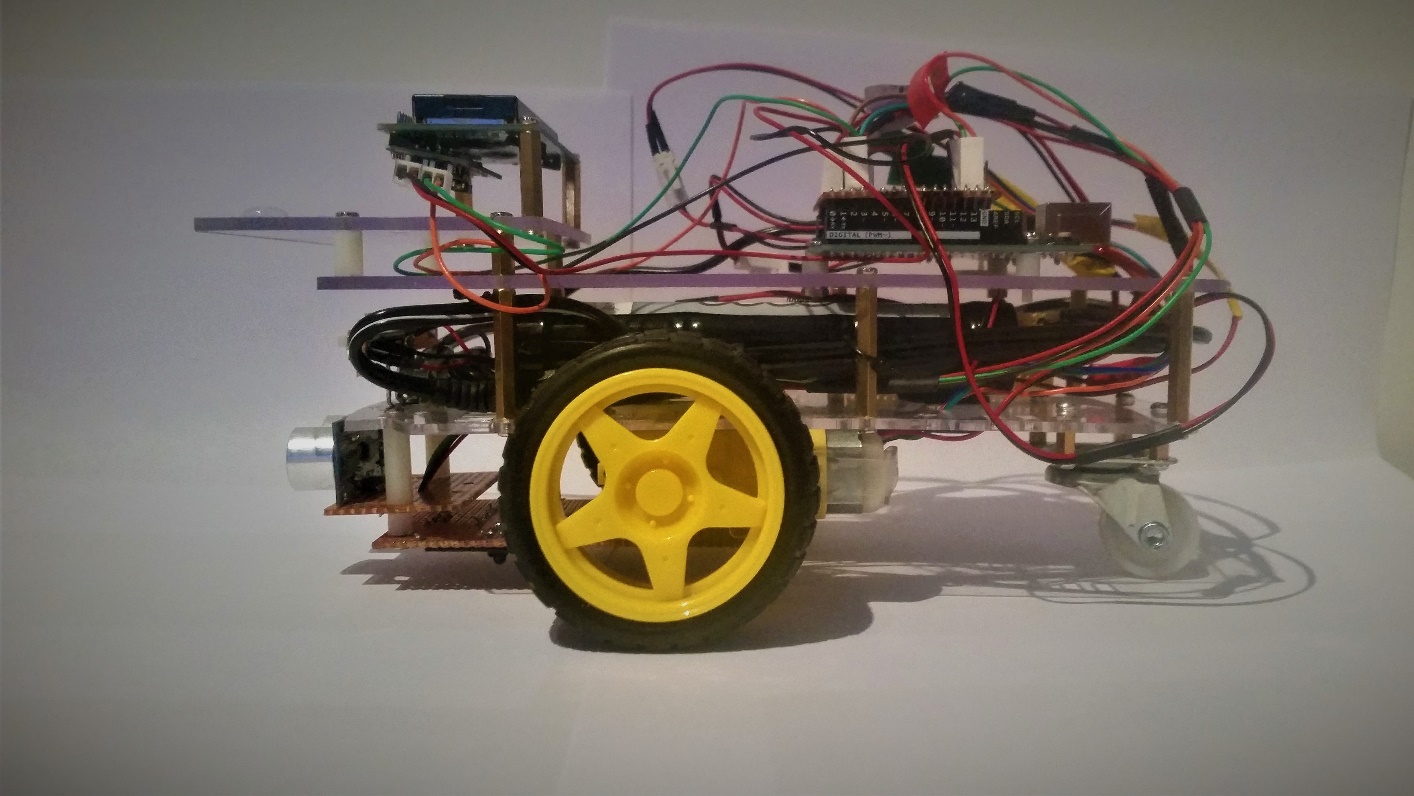
Ultrasonic

1. Red x 1 – live
2. Black x 1 – ground
3. Orange x 1 – trigger
4. Green x 1 – echo

LCD

1. Red x 1 – live
2. Black x 1 – ground
3. Orange x 1 – SCL
4. Green x 1 - SDA

## Chassis Measurements



118 mm

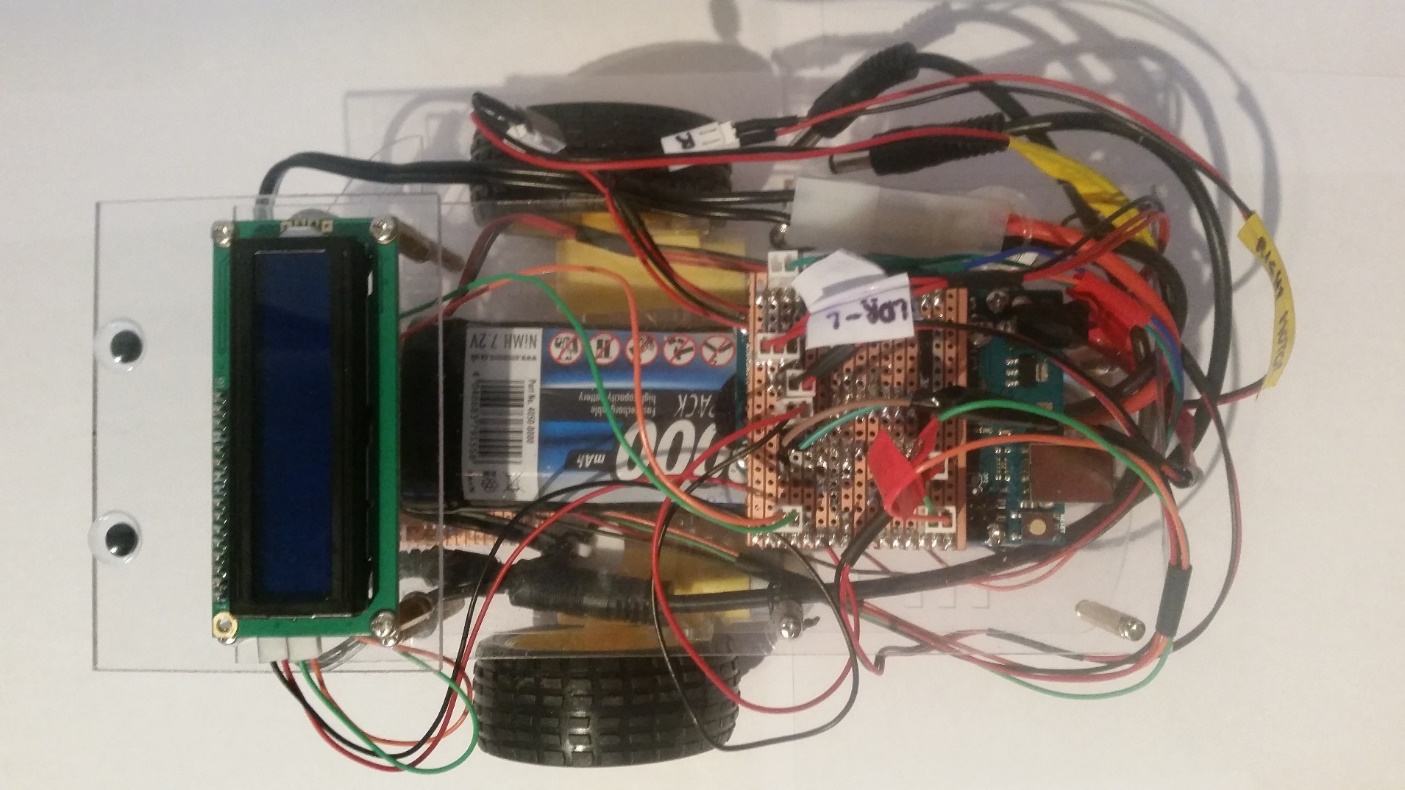
90 mm

78 mm

20 mm

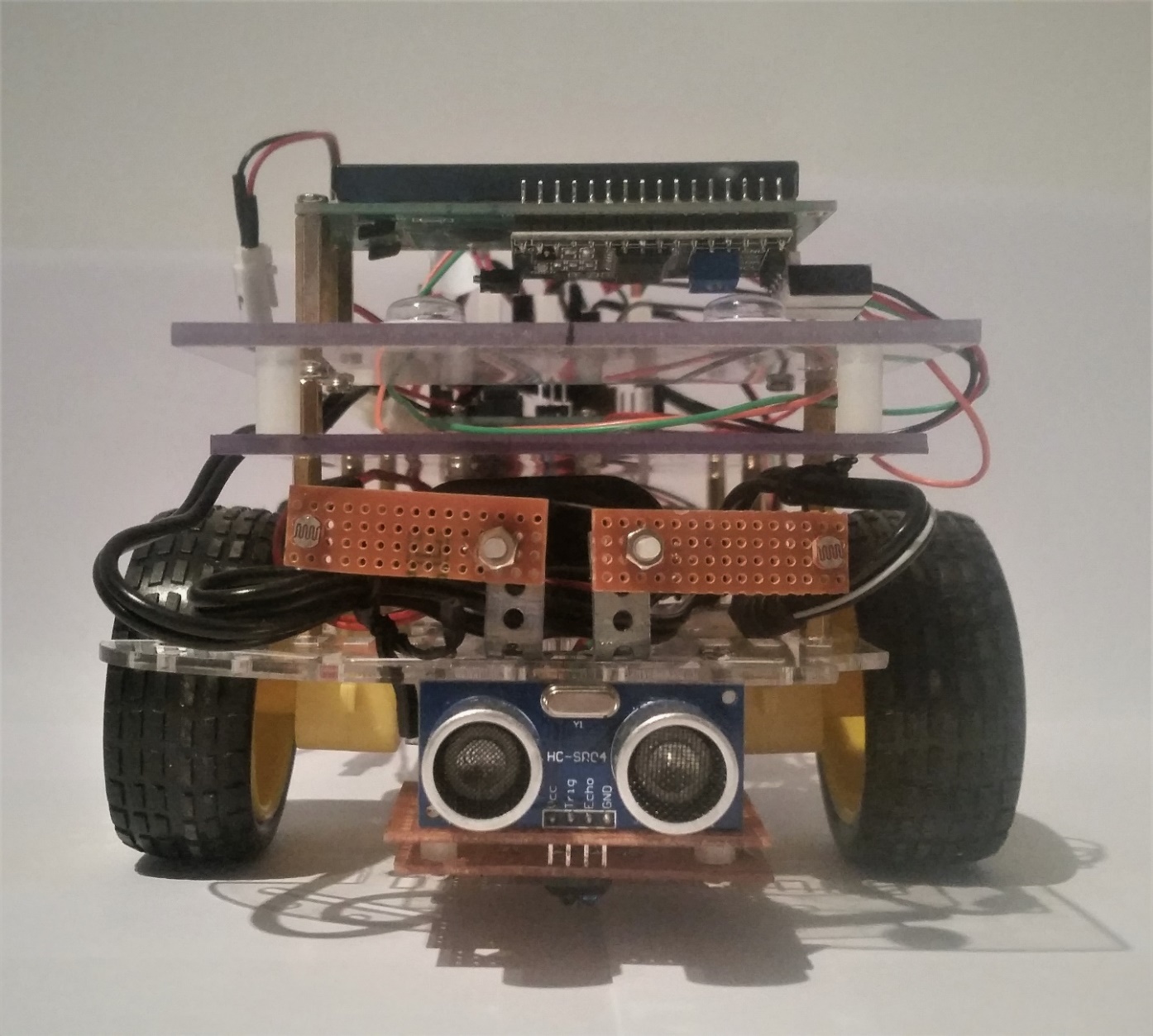
5 mm

45 mm



100 mm

230 mm



100 mm

118 mm