Universidad Zaragoza
1542

Trabajo Fin de Master

Master Universitario en Ingeniería Industrial

Sistemas Electrónicos

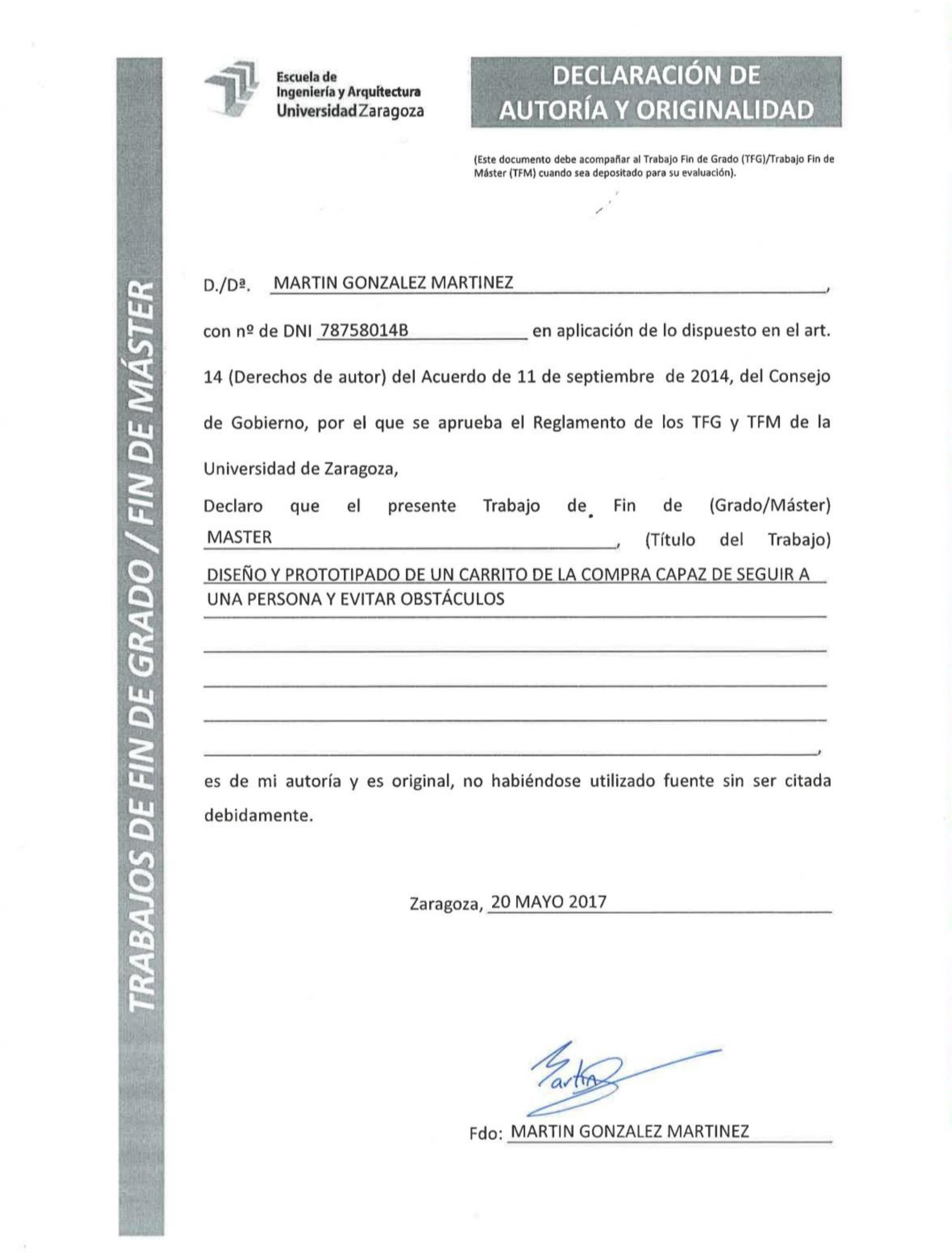
# Diseño y prototipado de un carrito de la compra capaz de seguir a una persona y evitar obstáculos

Autor:
Martin Gonzalez Martinez

Director:
Kathleen Meehan

Ponente:
Eduardo Aznar Colino

Universidad de Zaragoza
Junio 2017



TRABAJOS DE FIN DE GRADO / FIN DE MÁSTER

Escuela de
Ingeniería y Arquitectura
Universidad Zaragoza

# DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD

(Este documento debe acompañar al Trabajo Fin de Grado (TFG)/Trabajo Fin de Máster (TFM) cuando sea depositado para su evaluación).

D./Dª. MARTIN GONZALEZ MARTINEZ,

con nº de DNI 78758014B en aplicación de lo dispuesto en el art. 14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de (Grado/Máster) MASTER, (Título del Trabajo)

DISEÑO Y PROTOTIPADO DE UN CARRITO DE LA COMPRA CAPAZ DE SEGUIR A UNA PERSONA Y EVITAR OBSTÁCULOS,

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada debidamente.

Zaragoza, 20 MAYO 2017

Martin

Fdo: MARTIN GONZALEZ MARTINEZ

# ABSTRACT

## Shopper’s Assistant

The aim of this Final Year Project was to design a shopping trolley that is able to navigate autonomously following a human within a 3 feet distance, transport a load up to 20 kg and avoid collisions. Due to the large weight of the load, it has been decided to build a smaller prototype and then choose the components needed to build the real size trolley.

One infrared transmitter is attached to the body of the target person and several infrared phototransistors are placed on the robot to calculate the relative position robot-person and the distance between them. A collision avoidance algorithm has been implemented using ultrasonic sensors for detecting the nearest obstacles.

An ARM mbed NXP LPC1768 microcontroller coordinates every sensor measure and control calculation. It also controls the power stage designed for driving the motors.

# Acknowledgements

I would like to express my gratitude and a few words of thanks to all the people that, in some different ways, have helped me on the accomplishment of this project, because it would not have been possible without them.

First of all, I would like to thank my family, my brother and specially my parents, for the help not only during this project but also during these years as a university student and for all the values they taught me.

I would like to express my sincere gratitude to my supervisor Dr Kathleen Meehan, for her guidance, motivation, encouragement and continuous support. Her guidance helped me not only during my research and lab work but also when writing this report.

I would also like to acknowledge the contribution of Andrew Phillips, the person in charge of the electronics laboratory where I have been working at University of Glasgow. He helped me a lot during this project, especially at the very beginning stage, when I had just arrived and was more lost. I have also appreciated Alan Yuill and Shona Ballantyne’s help, who also work in the laboratory and have contributed to the development of this project as well.

I can’t finish without saying thank you to my friends, not only the ones I met during my studies in Spain and here in Scotland, but also my lifelong friends, that have influenced me in a very positive way at a personal level.

# List of Content

# List of Figures

# 1 Introduction

Technological innovations are present every day in our society. The retail sector is not unfamiliar with the technological innovation that has been going on in the past decades. The way customers shop is changing very fast. Some of these innovations are outlined below.

In December 2016, Amazon unveiled “Amazon Go”, a high-tech retail store with no check-out lines. The system logs any item the shopper takes from the shelves which eliminates the need of a traditional check-out line. When the shopper leaves the store, his virtual cart is checked out automatically and charged to his Amazon account [1], [2].

During Expo Milano 2015, Carlo Ratti Associati presented their *Supermarket of the Future* project. A few months later, Coop Italia opened their first supermarket incorporating this technology, including “augmented labels” where each product can communicate its nutritional properties, its origin or the presence of allergens when a shopper puts his hand close to it [3].

Nevertheless, if we have a look at shopping carts, they are not being improved with any modern technology. In 2014, Uplift, a folding shopping cart created by Yen Le Lofting, a young industrial designer from University of Houston, was named the Year’s Best American Invention by James Dyson Foundation [4]. It consisted of “*a personal utility cart with the ability to collapse and directly load itself into a vehicle’s storage compartment to eliminate the effort of repetitive loading/unloading and heavy lifting*” [5].

Last year, the American multinational chain of hypermarkets Walmart secured a patent for the self-driving shopping cart [6]. This patent shows a robot positioned under the cart steering it according to the shopper's direction. However, like other patents, it could take years before the development of this technology.

This is why at University of Glasgow it has been considered convenient to develop a prototype of an autonomous shopping cart as a Final Year Project.

## 1.1 Project framework

It should be noted that this project has been carried out within the framework of the University of Glasgow Undergraduate Final Year Project course *Individual Project 4 ENG4110P*, with the supervision of two university lecturers and researchers.

The bulk of the work has been carried out at the electronics laboratory located in the Rankine Building of the University of Glasgow. In this laboratory, every Final Year Project student has access to all the available electronic instrumentation such as power supplies, oscilloscopes, wave generators and soldering stations. There are also several technicians working in this laboratory that contribute with their extensive experience and support students with their projects.

## 1.2 Objectives

The main objective of this project is to design a semi-autonomous robot trolley that is able to follow a human within a distance of 3 feet, transport a load up to 20 kg and avoid collisions.

Regarding the large load the trolley has to transport, it would be easier to build a smaller working prototype and then choose the components needed to build the real size trolley.

The objectives of the project can be summarised as:

- **Design a power stage to control the motors.**

  Design a power circuit to drive the motors properly including all the necessary protection components using power electronics theory.

- **Design a person tracking system and implement a person following algorithm.**

  Design a system that permits the robot to follow the target person including the development of a target recognition and relative positioning system and implement an algorithm that allows the robot to follow the target person within a fixed distance.

- **Design and implement a collision avoidance algorithm.**

  Choose suitable sensors to detect possible obstacles along with the implementation of an algorithm that makes the robot deal with these obstacles, avoid collisions, and damage to the shopping trolley or other people.

- **Choose the appropriate components to build a real size shopping trolley.**

  After building the small-size shopping trolley prototype, the components for the real size trolley have to be selected, specifically the H-bridge, motors and battery.

## 1.3 Report description

This report is divided into nine chapters:

- Chapter 1 is dedicated to introducing the project, giving a brief outline of its aims along with an initial idea of its framework.
- Chapter 2 describes the main features and specifications of the prototype giving an idea of the components used.
- Chapter 3 explains how the power stage that drives the motors has been designed, underlining the main features of the components along with qualitative and quantitative indications for justifying their choice.
- Chapter 4 covers the person tracking system design and justifies the choice of the technology used along with an explanation of both the emitter and receiver devices' design.
- Chapter 5 describes the code implementation of the main program and the interrupts for measuring the sensors along with the characterization of the robot's movement. It focuses on the person following algorithm.
- Chapter 6 covers the collision avoidance algorithm implementation and explains the procedure and the sensors used for detecting obstacles.
- Chapter 7 presents the advantages of printed circuit boards over prototyping boards and shows the layout of the PCB that has been created for the trolley.
- Chapter 8 covers the design of the real size trolley according to the prototype development, explaining which will be the main differences between the prototype and a real-sized trolley and which components could remain on the real-sized trolley.
- Chapter 9 outlines the conclusions of the project, recalling the objectives of the project and making a point about potential further studies and improvements.

# 2 Prototype description and specifications

As it has been mentioned in the previous chapter, a small-size prototype of an autonomous shopping trolley has been built during this project. During this report, whenever a reference to the shopping trolley prototype is made, it will be referred to by any of these names: (shopping) cart, (shopping) trolley, prototype or robot.

The main features of the components used are listed below:

- **Robot kit** shown in Figure 1:
    - Rectangular aluminium chassis (150 mm x 160 mm x 50 mm)
    - 4x Ø 120 mm x 60 mm wheels.
    - 4x 12 V DC motors.

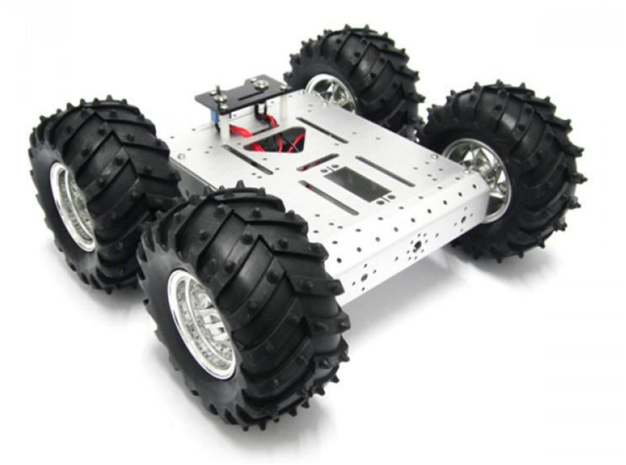


*Figure 1. Aluminium four-wheeled drive robot kit* [7]

- **Microcontroller:** Taking into account that the robot has to deal with several sensor inputs and must be able to perform a wide variety of control actions depends on these sensor measurements, it needs a brain for computing all these inputs and making the calculations of the outputs. This brain is a microcontroller, and an ARM mbed NXP LPC1768 has been used for this project, whose main features are [8] :
    - High performance ARM® Cortex™-M3 Core.
    - 96 MHz, 32 KB RAM, 512 KB FLASH.
    - 40-pin 0.1" pitch DIP package, 54x26 mm.

  - 5 V USB or 4.5-9 V supply.
  - High level C/C++ SDK.
  - Cookbook of published libraries and projects.
- **Power supply:**
  - 12 V 2800 mAh NiMH battery.
  - 12 V 1.0 A / 2.0 A charger with an overheating detection system installed by one of the technicians in the laboratory of the University.
- **Power stage:**
  - Double H-bridge: Two complete H-bridge modules of four transistors that allows to drive two different loads, DC motors, in whether one direction or the other.
  - Eight protection diodes: They allow a low-impedance path for discharge the motor current.
  - Six resistors for protecting the microcontroller outputs.
- **Person tracking system:**
  - Infrared (IR) transmitter attached to the body of the target person.
  - IR phototransistors placed on the robot for measuring the intensity of the signal from the IR transmitter and the relative position of the target person.
- **Collision avoidance system:**
  - Ultrasonic sensors for measuring the distance to the nearest objects in order to avoid collisions.

Every component present on the prototype is outlined on Appendix D.

# 3 Power stage design

Among the several electronic solutions for driving DC motors [9], those converters that use transistors as switches are the most efficient choice. This is because Bipolar Junction Transistor (BJT) only operate by switching between saturation and cut-off modes and Metal-oxide-semiconductor field-effect transistors (MOSFET) only operate by switching between ohmic and cut-off modes. In this way, power losses associated with the converter are small in comparison with those whose switches work in active mode in the case of BJT transistors, or in the saturation mode in the case of MOSFET.

As it has been mentioned in the paragraph above, this type of converter uses switches that work exclusively in switching between *on* and *off* modes. The *on* mode is characterized by a low voltage drop across the switch even if there is a big current passing through it. The *off* mode is characterized by a low current flow even if a large voltage is applied.

## 3.1 Power supply

The motors are the component that need the greatest voltage among all of the electronic components within the prototype. Four 12 V motors are being used so the power supply for the whole system should be 12 V at least. That is why a 12 V battery has been used.

The mbed NXP LPC1768 microcontroller can be powered between 4.5 and 9 V and for safety reasons it is powered with 5 V using a ST L7805CV voltage regulator that supplies 5V from any voltage between 5 and 35 V.

Having a look at power consumption of the components:

- The motors have a rated current of 0.52 A each when driven at full power.
- The ultrasonic sensors used for collision avoidance draw 15 mA each.
- The microcontroller draws around 200 mA.
- The IR sensors use 3.3 V and large current-limiting resistors so they draw almost no current.

Therefore, Equation (3.1) shows the maximum current value drawn by the whole system.

$$I_{max} = 4 \cdot 0.52\ A + 2 \cdot 0.015\ A +\ 0.2\ A\ \cong 2.2\ A \tag{3.1}$$

As the motors are not always working at full power, an average factor can be applied to the motor current for calculating an estimation of the average current consumption of the system. This average factor has been set to 0.7. Taking into account that the robot is not always moving during its operation, another factor of 0.7 has been added to consider the times that the motors are stopped. Therefore, the estimated average current calculation is shown in Equation (3.2).

$$I_{avg} = 4 \cdot 0.52 \cdot 0.7 \cdot 0.7\ A + 2 \cdot 0.015\ A +\ 0.2\ A\ \cong 1.25\ A \tag{3.2}$$

It was decided to use a 2800 mAh 12 V NiMH battery available in the laboratory store, that with this average current consumption it will have an estimated battery life of 2 hours and 15 minutes as calculated by Equation (3.3).

$$Battery\ life = \frac{Battery\ capacity}{Average\ current} = \frac{2.8\ Ah}{1.25\ A} = 2.24\ hours \cong 2\ h\ 15\ min \tag{3.3}$$

## 3.2 H-Bridge

An H-Bridge is a DC-DC converter with four transistors, as shown in Figure 2, that are switched on and off two by two, allowing the application of a voltage across a load in either direction. They are widely used for driving motors because they allow changing the direction of rotation.

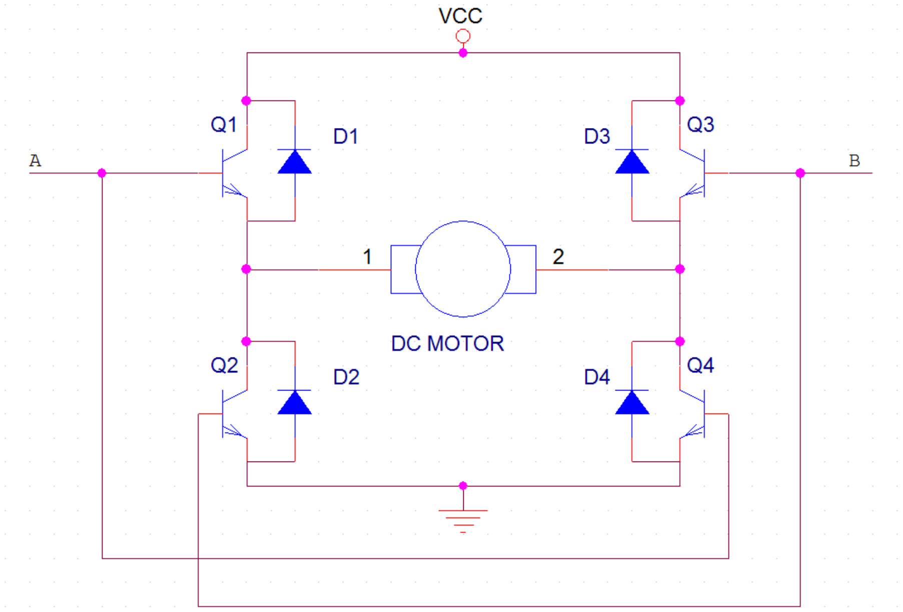


Figure 2. H-Bridge configuration

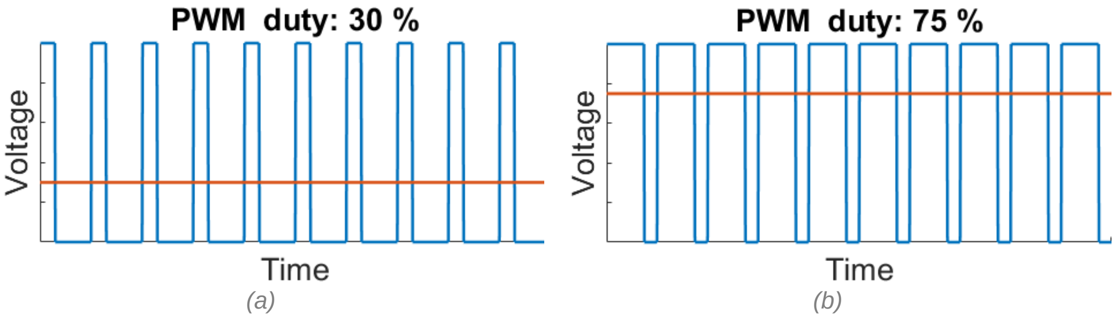
A STMicroelectronics L298n H-bridge has been used for the development of the prototype. Its main features are [10]:

- High voltage and high current dual full-bridge driver (Up to 46 V and 4 A).
- Dual Full-Bridge: it has two bridges in the same IC so it can drives two motors independently at the same time.
- The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the connection of an external sensing resistor.
- It uses additional supply input for making the logic work at a lower voltage.

Figure 2 also shows that there is one diode in anti-parallel with each transistor to permit a discharge path for inductive current of the motor.

## 3.3 PWM Modulation and microcontroller interfacing

The H-bridge allows controlling the speed of the motor using Pulse Width Modulation (PWM), which uses high frequency DC square wave signals. By changing the duty cycle, which is the proportion of the square wave period that the pulse is high, the average voltage given to the motor changes as it is shown in Figure 3 along with the motor's velocity. As the square wave has high frequency, the load behaves as if it was driven with a constant supply with the average value as described with the red line on the figure.



(a) (b)

Figure 3. PWM square waves examples. (a) Duty 30%. (b) Duty 75%.

When programming the mbed microcontroller used for the development of the prototype, a PWM output can be easily defined in the compiler, what is very helpful during the coding process. Once you have defined the pin where you want to have your PWM output, that has to be one of the PWM outputs of the microcontroller, you can easily set a period for the square wave using different functions depending if you want to use seconds, milliseconds or microseconds. For the duty cycle, it allows to define it using either amount of time or a percentage of the period. And as if all that were not enough, on the mbed microcontroller developer site, there is a free-to-use library called *Motor.h* that has everything pre-set for driving a DC motor using an H-Bridge [11]. It is only needed to define those pins that the motor is connected to (pwm, forward and reverse) and then easily set the speed whenever necessary.

An example of how to use this library is shown in Figure 4, where it can be seen that the speed is set as a percentage of the maximum, what is the same as defining the duty cycle of the PWM wave. In this example the motor will be supplied with an average voltage of 70% of the power supply voltage.

```
#include "mbed.h"
#include "Motor.h"
Motor m(p23, p6, p5);   // pwm, fwd, rev
int main() {
    float s = 0.7;
    m.speed(s);
}
```

*Figure 4. Motor.h library example algorithm*

As the prototype has four motors, there are several different forms to drive them, but it has been decided to drive each side separately as if there were only two motors. It can be thought that this is not the best way to do it, but it has been sufficient for the prototype demonstration and it made the coding and the electronics easier than driving each motor separately. In case every single motor is drive separately, four h-bridge will be needed instead of two and, also, four speeds will need to be set within the coding stage so that is why it has been decided to drive each side separately instead of each motor individually.

## 3.4 Protection components

Although, in previous sections, some protection components have been named, this section aims to recap what has been said before and introduce some other protection components that have not been named yet but are also present on the prototype electronics.

- Taking into account that DC motors are inductive loads, four diodes are used for protect the motors on each side from the current on them. When the transistors shut off, the current that must still flow thought the motor, will flow through the diodes.

- A 100 kΩ resistor is used for protect each microcontroller output pin connected to the H-bridge from the possible transient current coming from the bridge when transistors are turned *on* and *off,* as the laboratory technician recommended.
- Two fuses have been used when designing the stage for protecting against excess currents:
    - A 2.5 A fuse has been placed for protecting the motors considering the maximum current value calculated on Equation (3.1).
    - A 500 mA fuse protects the microcontroller and the sensors. This fuse value was chosen following the laboratory technician's advice and considering the available fuses at the laboratory.

Every single component described in the sections above has been implemented on the electronic power stage designed for the prototype. After the prototyping period, a printed circuit board has been designed to allocate not only all the power electronics components but also all the sensors and the microcontroller as described in Chapter 7.

# 4 Person tracking system

One of the main features of autonomous vehicles and autonomous mobile robots is that they are able to sense their environment and navigate without human inputs [12]. For sensing the environment, they can use different technologies and sensors.

Person tracking is a challenging task when talking about mobile robots. It has been an active research field in the last decades due to its wide range of application [13]–[17]. A wide variety of sensors can be used for target tracking in mobile robots and it is not easy to select the right one for each application. When choosing a sensor for any kind of application it is important to know certain features of each sensor such as type of sensor, accuracy, range, environmental conditions or cost.

## 4.1 Types of sensors

At the beginning of this project, many different sensors were considered for the person tracking system, whose main features are described below.

- **Camera or image sensor:** They are the most popular sensors for tracking and detection in mobile robots. Combined with software like OpenCV, they are very powerful for human detection. In order to use this computer vision technology in this project it would be necessary a microcomputer like Raspberry Pi [18] or a whole computer mounted in the prototype.

  Some of these image sensors, such as Kinect or Leap Motion, also have a depth sensor that can be combined with the correct software for human tracking or gesture recognition [19], [20].

  However, the implementation of these sensors can be much more difficult than using other simpler technology.

- **Ultrasonic distance measurer:** These sensors use ultrasonic waves for measuring distance to the nearest object. Even if there are some cheap ultrasonic modules (less than £5 each) with enough range and accuracy for this task, they are not the best choice for target tracking because they are not able to differentiate between objects. However, they can be an interesting choice for other tasks like collision avoidance as can be seen in Chapter 6.

- **Infrared transmitter-receiver system:** An IR transmitter and some IR receivers can be used to know the relative position between them. It is inexpensive, easy to interface with the mbed microcontroller and suitable for indoor applications considering that sunlight is composed by not only visible and ultraviolet light but also infrared. However, as this trolley is going to be used indoors, this technology is very appropriate.
- **GPS module:** Global Positioning System (GPS) modules use satellites that transmit radio signals to determine their location. However, this technology cannot work indoors because it is not able to pass through solid structures.

## 4.2 Simplifying assumption and sensor choice justification

After the stage of first literature research along with a time to think about how to address every single challenge that this project poses, and considering the limited time available, an assumption was made to reduce the scope of the problem. The robot and the target person are going to be close enough that nothing should interfere so that if a transmitter-receiver system is used for tracking.

At the very beginning of the project, it has been considered to use a Raspberry Pi and a PiCamera Module with OpenCV software. However, due to its programming complexity in comparison with other suitable sensors and microcontrollers, it was decided to develop the prototype using an mbed microcontroller and an IR transmitter-receiver tracking system.

## 4.3 IR Tracking system

Infrared radiation is electromagnetic radiation that extends from visible light spectrum at 700 nm to 1 mm of wavelength. It has longer wavelengths than those of visible light, so it is invisible for humans and does not disturb anyone.

As it is said in the previous section, the idea of the IR tracking system consists in an IR transmitter attached to the target person and some IR receivers placed on the robot (Figure 5) that can sense not only the intensity of emission that gives an idea of the distance to the transmitter but also the relative position.

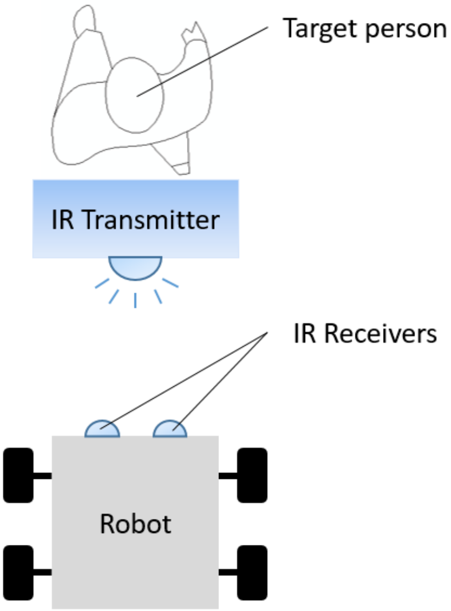


Figure 5. IR Tracking system diagram

For that, receivers have to be sensible to the wavelength of the signal that the transmitter uses. After several laboratory tests, it has been decided to use 940 nm as wavelength for both transmitter and receivers.

### 4.3.1 IR transmitter

Figure 6 shows the device that has being used as IR Transmitter during the development of the prototype and its electric circuit. As it has to be portable because the target person should attach it to his or her body, it is powered by two AA batteries and has a switch to turn it on and off. Taking into account that IR light is not detectable by human eyes, a red LED has been incorporated to give feedback of the device status to the user. The fundamental part of the device are three infrared LEDs that work as transmitters.

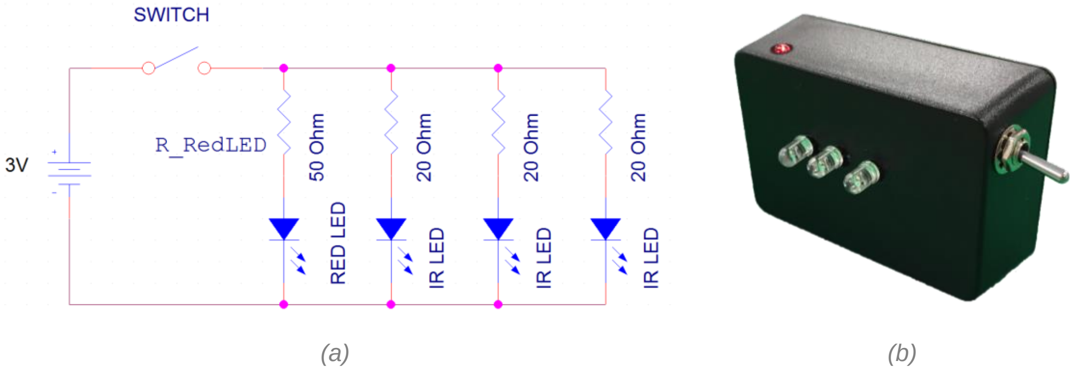


Figure 6. Infrared transmitter. (a) Circuit (b) Prototype IR transmitter device

Both infrared LEDs and the red LED have a resistor in series for limiting the current flowing through them. As two AA alkaline batteries are being used, 3 V is the voltage across each group of series resistor and LED.

The red LED is a Kingbright L-7113IT 5 mm 2 V Red LED supplied by Rapid Electronics. Its forward voltage $V_F$ is 2 V and its maximum forward current $I_{Fmax}$ is 30 mA [21]. It has been determined that there should be a 20 mA current flowing through this LED. Equation (4.1) shows the calculation of the series resistor required for reach these specifications.

$$R_{Red\ LED} = \frac{Vcc - V_{F\ Red\ LED}}{I_{F\ Red\ LED}} = \frac{3\ V - 2\ V}{20\ mA} = 50\ \Omega \tag{4.1}$$

The IR LEDs used are the Stanley Electric AN5307B supplied by RS. They come in a 5mm through hole package and have a peak wavelength of 940 nm and a half intensity angle of 38º as shown in its datasheet. They have a forward voltage $V_F$ = 1.35 V and a maximum forward current $I_F$ = 100 mA [22].

As the maximum forward current that this IR LEDs can draw is 100 mA, a resistor in series of each one is needed for limit the current flowing through them. 20 Ω resistors have been used as the current flowing is not only less than the maximum permitted value but also enough for the LEDs to emit the signal properly. (Equation (4.2) and (4.3))

$$I = \frac{U}{R} = \frac{3\ V - 1{,}35\ V}{R} = \frac{1.65\ V}{R} < 100\ mA \qquad \therefore\ R > 16.5\ \Omega \tag{4.2}$$

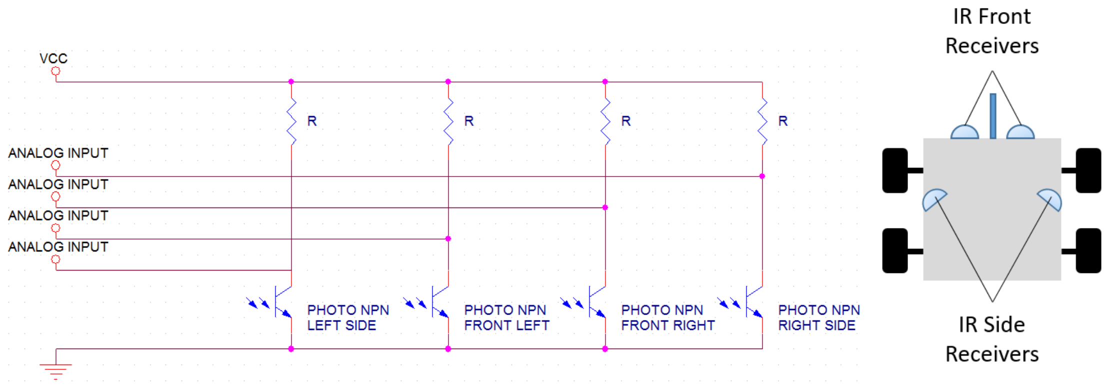
$$R = 20\ \Omega\ \rightarrow\ I = 82.5\ mA \tag{4.3}$$

This IR transmitter battery life can be calculated considering that, as it has been said before, the red LED draws 20 mA and the IR LEDs draw 82.5 mA each. The total current supplied by the batteries is 267.5 mA. Equation (4.4) shows the calculation of the battery life using alkaline batteries, which is around 9 hours and 20 minutes.

$$Battery\ life = \frac{Battery\ mAh}{Load\ mA} = \frac{2500\ mAh}{267.5\ mA} = 9.34\ h \approx 9h\ 20min \tag{4.4}$$

### 4.3.2 IR receivers design

Once it has been explained how the IR transmitter works, it is easy to understand how the IR receivers work. Figure 7 shows both the electric circuit (Figure 7.a) and the diagram of how the IR receivers are placed on the robot (Figure 7.b).



(a) (b)

Figure 7. IR receiver (a) Electric circuit. (b) Prototype diagram

Figure 7.a shows that one IR phototransistor and one resistor compose each IR receiver. The system operation is very easy, the more IR light the phototransistor receive, the more current draws from its collector to its transmitter and then, the more voltage across the resistor, that can be measured with an analogue input pin on the microcontroller. The sensitivity of the system can be adjusted by changing the value of the resistor.

Taking into account that the voltage across the resistor should be read by the microcontroller, the value of this voltage should be greater than 0 V but less than 3.3 V that is the range of the ADC [8]. That is why the value of $V_{cc}$ for the IR receivers has been set to 3.3 V.

The IR phototransistor use are the Vishay TEFT4300 supplied by RS. They come in standard 3 mm (T-1) through-hole packages and their main features are high radiant sensibility, a daylight-blocking filter matched with 940 nm transmitters and fast response time [23].

Considering that a different value of resistor in series with the phototransistor changes the circuit sensibility to light, a meticulous analysis has been carried out for choosing the resistors values. Figure 8 shows the value of the analogue input (in Volts) of one sensor versus de distance to the transmitter using different resistor values. Nine different values of resistors have been analysed using two sensors. For each resistor value, seventeen different distances have been evaluated, what results in more than 300 measurements.

As the phototransistor draws current depending on the amount of IR light it receives, a value of 0 V means that the transmitter is very close to the receiver and a value of 3.3 V means that there is no signal received by the receiver.

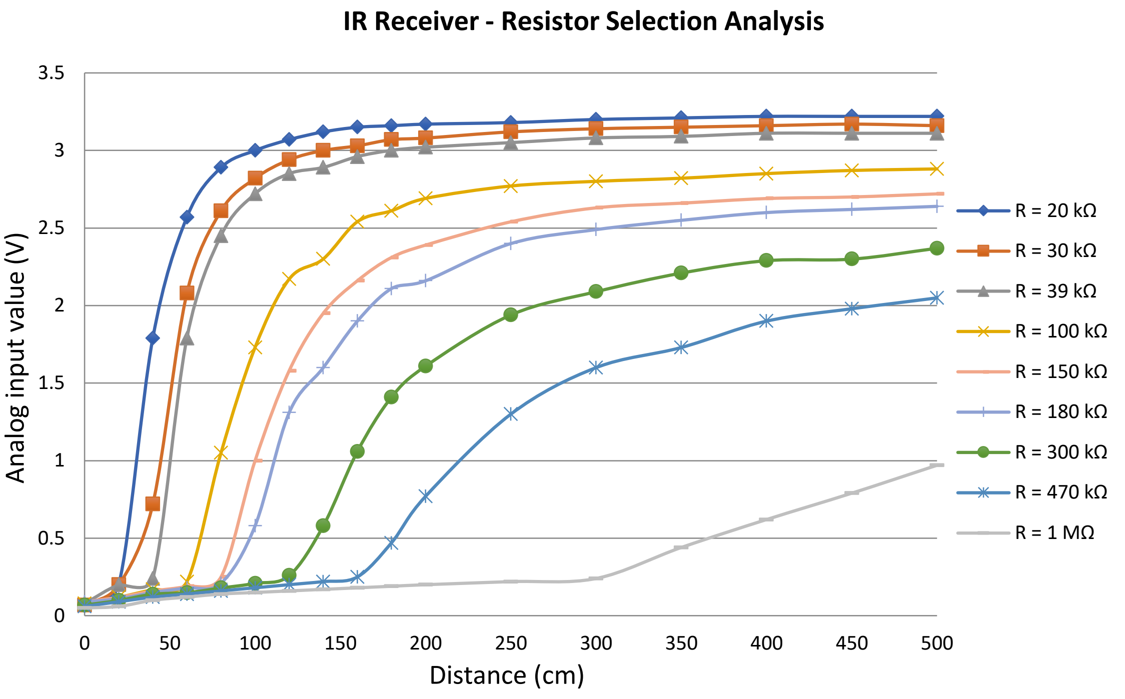


*Figure 8. IR receiver calibration: influence of resistor*

By having a look at Figure 8, it is clear that none of these resistor values return a linear behaviour. It has been considered a better idea to make the system more sensitive (approximately linear behaviour) within the working distance range at the expense of a bigger maximum emitting sensible distance. Thus, it can be guaranteed that when the trolley is in the set point distance range, the IR receiver will give the robot accurate measurements.

As the distance between the trolley and the person (set point) has been set around 3 feet, what is a little bit less than 1 meter, it would be necessary that the robot could differentiate if it is whether near or far from this distance and also if it is even closer than this distance.

Taking all this into consideration, the optimal value for the resistor is either 150 kΩ or 180 kΩ, because they both not only permit to distinguish if the robot is closer than the set point clearly but also permit to differentiate how far robot and person are in a reasonable range of distances. Thus, a value of 150 kΩ has been chosen.

### 4.3.3 IR Receivers implementation

At the beginning of the project, in the first working versions of the software, the robot was able to navigate using only two IR receivers that were located on its front instead of having two on the front and one on each side as in the final prototype.

As it can be seen in Figure 7.front sensors are physically separated by a sort of wall that works as a barrier and allows the robot to know the relative position of the target person as Figure 9 shows.

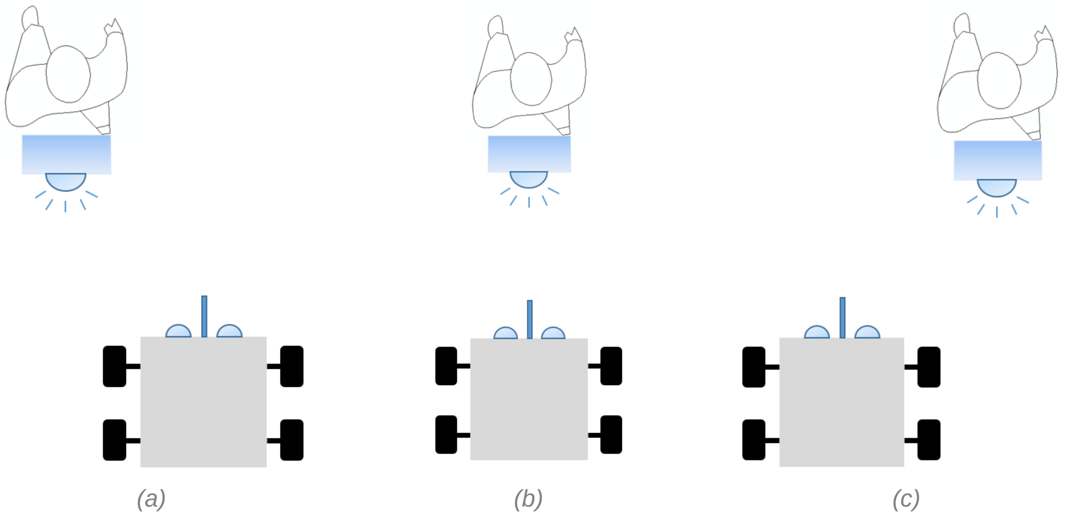


Figure 9. Relative positioning with two IR receivers. (a) The target person is on the left side of the robot. (b) The target person is in front of the robot. (c) The target person is on the right side of the robot.

The working principle of the system is based on the differences between output voltages of both front sensors. One of the biggest challenges in this part of the project is that due to the phototransistors' manufacturing process and calibration and the resistors' tolerances, output voltages slightly changed from one receiver.

If left receiver measurement is substantially greater than the right receiver measurement, the target person is on the left side of the robot (Figure 9.a). If both sensors measure almost the same value, there are the same voltage drop both across both resistors and the target person is in front of the robot (Figure 9.b). Finally, if right receiver measurement is substantially greater than left receiver measurement, the target person is on the right side of the robot (Figure 9.c).

Even if this version worked in an acceptable way, it was thought that by adding two more IR receivers, one on each side of the robot, the relative positioning would be better, as illustrated in Figure 10, and it would help when the collision avoidance algorithm turn up later on. These two new sensors are placed on both sides of the robot and they permit to locate the target person when he or she is not in the field of vision of the front sensors

(Figure 10.a and Figure 10.e). They allow the robot to catch the target person by turning on itself.

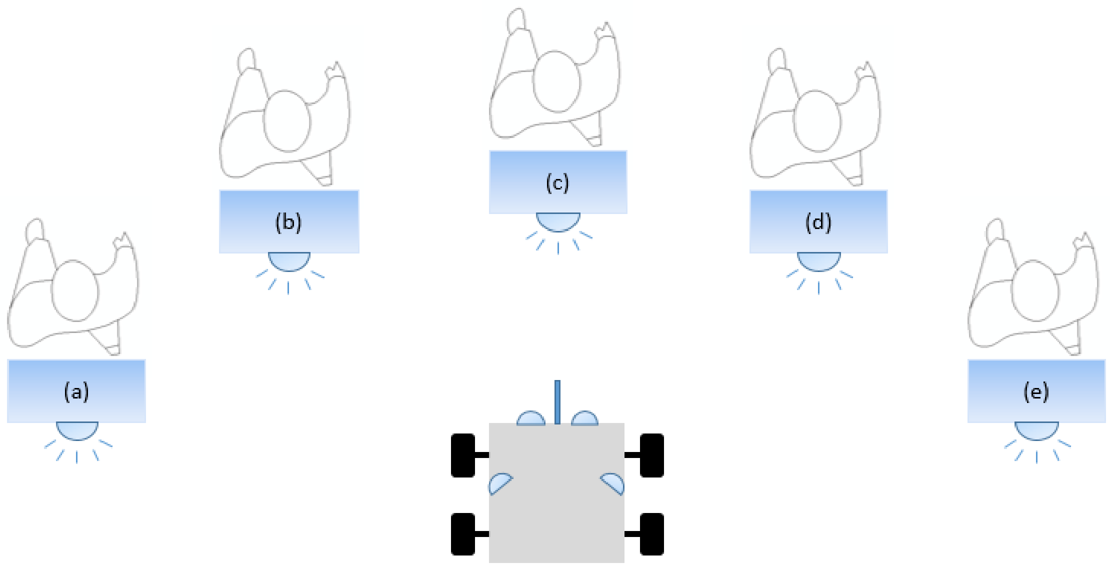


*Figure 10. Relative positioning using four IR receivers. (a) The target person is on the left side of the robot and not in the field of vision of front sensors. (b) The target person is on the left side of the robot. (c) The target person is in front of the robot. (d) The target person is on the right side of the robot. (d) The target person is on the right side of the robot but not in the field of vision of front sensors.*

# 5 Person following algorithm

With the progress of the society and the development of science and technology, autonomous robots have become more and more usual. An example would be the worldwide known robotic vacuum cleaner, Roomba, introduced in 2002 by the American company iRobot, which is an autonomous vacuum cleaner that is able to clean an entire level of a home and it can change direction when it finds obstacles [24], [25].

In recent years, target tracking and following is one of the most interesting fields of study when talking about autonomous robots. There are a few examples about vehicles able to autonomously follow a person [26], [27]. Relating these research works with this project, there are some algorithms and following schemes that can be taken into account when developing the person following algorithm for the purpose of this project [28].

In this chapter, the person following algorithm that has been created based on the person tracking system described in the prior chapter, is explained in detail along with the movement strategies developed at the very early stages of the project.

## 5.1 Movement characterization

Once the power stage was completely designed, several motor tests were carried out in order to establish which strategies were going to be used for moving the robot. The aim of these tests were to determine how the robot moved depending on the control actions send by the microcontroller. This section describes the process conducted to define the microcontroller actions needed for achieve an appropriate behaviour of the robot.

### 5.1.1 Forward movement characterization

First, a simple constant speed experiment was carried out. It was aimed at establishing the minimum duty cycle value necessary for moving the robot at reduced speed along with the maximum duty cycle value for a reasonably safe movement.

The experiment concluded that the minimum duty cycle for moving the robot, expressed as a percentage of the power supply value, was 25% and the maximum 100%. For safety reasons, this maximum value will be changed to 95%. From now on, if a speed value is described it will be simply defined as its duty cycle value.

### 5.1.2 Turns characterization

When talking about turns on a mobile robot, a distinction can be drawn between smooth or slight turns and sharp turns or rotations.

- **Slight turns**:

The robot moves forward and turns at the same time. It is done by setting different speed values on each side. After several tests, the best movements were achieved when setting speed values between 80% and 95% to the outer curve wheel and values between 25% and 30% to the inner curve wheel. If different values from those described above are used, the robot does not turn properly and only goes forward because there is not enough speed difference between both sides.

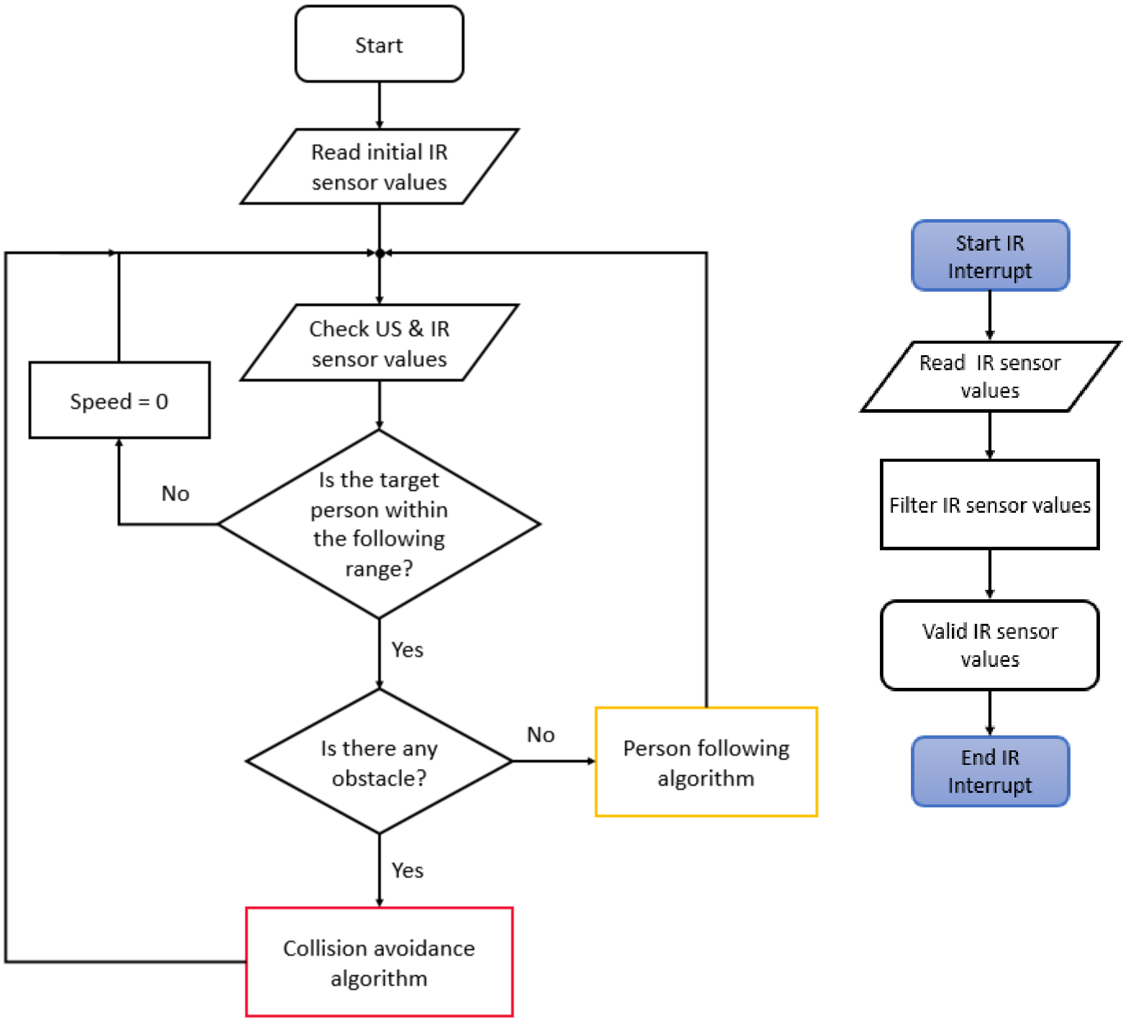
- **Rotate:**

The robot remains in the same place and rotates changing its direction. It is done by setting the same speed on both sides but a different direction of rotation on each side. After some tests, the best movements were achieved when using speed values between +70% and +90% for one set of motors and between -70% and -90% for the other set of motors. When using higher speed values, turns were too sharp, and when using lower speed values, turns were too slow to catch the target.

## 5.2 IR sensors measurement interrupts & main program

Figure 11.a shows the flowchart of the main program, which is the core of the robot's operation. It can be noticed that within this program, sensor measurements only appear at the initial stage, which are for initializing the IR sensors with the ambient IR light values in such a way that the robot is able to distinguish whether there is a target person to follow or not.

IR sensor measurements have been implemented in the microcontroller through the use of its timer's interrupts. To implement the interrupts, the Ticker class included in mbed.h library has been used. These interrupts use Timer3 on Cortex M3 microprocessor to pause the main program execution periodically, and run the function attached to this Ticker object by the user. A Ticker object called *Person_Tracking* has been defined for the IR sensor measurements with a 100 Hz frequency and with a function attached that reads the values of the sensors and applies a filter to the data. This frequency has been selected to ensure that the system works not only when a person walks at a regular pace but also when he or she moves away rapidly. The operation of this function is shown on Figure 11.b.


Start
Read initial IR sensor values
Check US & IR sensor values
Speed = 0
No
Is the target person within the following range?
Yes
Is there any obstacle?
No
Person following algorithm
Yes
Collision avoidance algorithm
Start IR Interrupt
Read IR sensor values
Filter IR sensor values
Valid IR sensor values
End IR Interrupt

(a) (b)

*Figure 11. Algorithms' flowcharts (a) Main program (b) IR sensors measurement interrupts*

The main program consists of a loop in which the robot is checking the values of the sensors, previously measured using interrupts, and decides whether the target person is within the following range or not and whether there is an obstacle or not. "Within the following range" means that the target person is farther than the reference distance, which the robot has to be separated from the person. If the target person is within the following range and there are no obstacles, the robot performs the *person following algorithm*, explained in Section 5.3. However, if there is a target person within the range but there are any obstacles, the robot performs the *collision avoidance algorithm* explained in Chapter 6. But instead, if the target person is not within the range, that means that the person is close to the robot, the robot will remain on its place waiting for the person to move away.

Figure 11.b shows the procedure for obtaining the IR sensor values that feed the main loop. During this procedure, a filtering process is mentioned. This filter aims to remove erroneous noise from the data obtained from the sensors diluting the impact of spurious values. However, this impact would never be completely removed.

The filter used is a *simple moving average filter*, which is the unweighted mean of the previous n data for each sensor values [29], [30]. In this case, the amount of previous data taken for the *moving average filter* is five.

## 5.3 Person following algorithm flowchart & behaviour

Once both the movement characterization and the person tracking system design were terminated, the design of an algorithm for following the target person was carried out. The flowchart in Figure 12 demonstrates the operation of this algorithm, which operates within the main program (Figure 11.a).

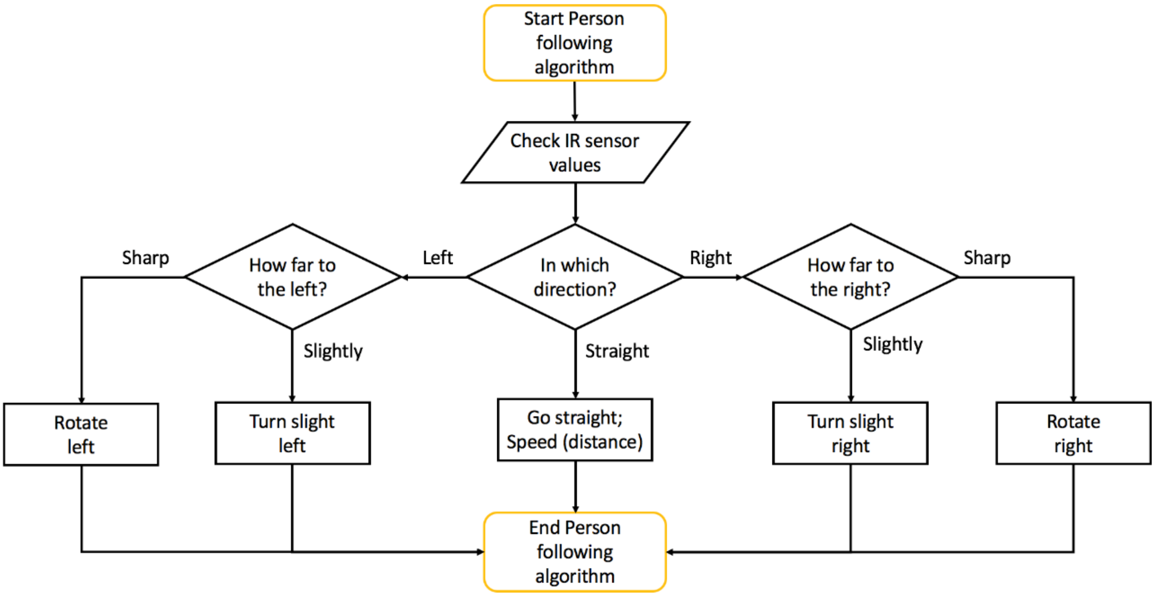
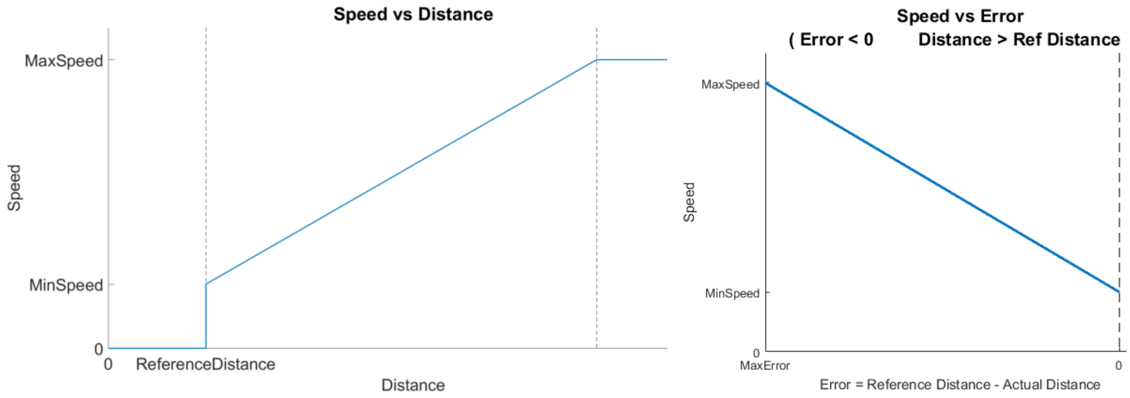


Figure 12. Person following algorithm flowchart

The main idea of this algorithm is to position the robot pointing to the target person at a fixed distance. At the beginning of the execution, the robot makes some initial measurements in order to quantify the IR environmental light and establish the idle values of the sensors. Once these initial measurements are done, the robot starts to check periodically the sensors in order to locate the target person. The algorithm distinguish between six different states depending on the sensor measurements, the five described in Section 4.3.3, which are the five dealt directly by the algorithm, and one more when the target person is too close to the robot, that finishes within the main loop:

- **The target person is too close to the robot**: If this happens, the robot remains stopped until the target person move away. Both sides' speed is set to 0%.
- **The target person is in front of the robot:** The robot goes straight in order to maintain the fixed distance with the target person. A proportional controller is used for calculate the speed of both sides' wheels only if the target person is farther than the reference distance as shown in Figure 13.



(a) (b)

*Figure 13. Person following algorithm graphs (a) Speed vs distance (b) Speed vs Error*

Figure 13.a shows the desired behaviour of robot's speed when it goes straight depending on the distance to the target person. Whenever it is closer than the pre-set *ReferenceDistance* value, it must remain stopped. This may happen when the shopper wants to place some goods in the trolley. If the distance is greater than *ReferenceDistance*, the speed is proportional to the difference between *ReferenceDistance* and the actual sensor-measured distance as shown in Figure 13.a, Figure 13.b, Equations (5.1) - (5.4).

$$Speed = Min_{Speed} + K \cdot error \qquad (error < 0) \tag{5.1}$$

$$error = (reference_{distance} - actual_{distance}) \tag{5.2}$$

$$K = \frac{Min_{Speed} - Max_{Speed}}{0 - Max_{error}} = \frac{Min_{Speed} - Max_{Speed}}{0 - (Max_{Distance} - reference_{distance})} \tag{5.3}$$

$$\therefore\ K = \frac{Max_{Speed} - Min_{Speed}}{Max_{Distance} - reference_{distance}} \tag{5.4}$$

For calculating the constant K, the minimum and maximum speed values for forward movement mentioned in Section 5.1.1 come into play as shown in Equations (5.3) and (5.4). The *MaxDistance* value set on these equations refers to the minimum distance value that the designer wants the robot to move with *MaxSpeed* and it is a designed parameter.

This speed control works just like the *adaptive cruise control* present in some cars, which automatically adjusts the vehicle speed to maintain a safe distance from vehicles ahead.

- **The target person is on the left side of the robot:** The algorithm split this case into two subcases:
    - **In the field of vision of front sensors:** If the target person is in the field of vision of front sensors, it will make a slight left for catching the direction of the person. It is done by setting the speed of left side wheels to 30% and the right side wheels' speed to 95% as stated in Section 5.1.2.
    - **Out of the field of vision of front sensors:** If the target person is out of the field of vision of the front sensors, it means that the person is located on the left side of the robot. The robot performs an anti-clockwise rotation.

- **The target person is on the right side of the robot**: Just like in the case above, the algorithm splits this case into two subcases. Both cases and movements are symmetrical to the ones described above:
    - **In the field of vision of front sensors:** If the target person is in the field of vision of front sensors, it will make a slight right by setting the speed of right side wheels to 30% and the left side wheels' speed to 95% as stated in Section 5.1.2 and in the previous case.
    - **Out of the field of vision of front sensors:** If the target person is out of the field of vision of the front sensors, it means that the person is located on the right side of the robot. The robot performs a clockwise rotation.

# 6 Collision avoidance

The majority of actual autonomous vehicles are able to detect if there are any obstacles in their way using a wide variety of sensors and technologies. They use this data can avoid collisions using complex control algorithms that are able to modify the vehicle's path safely [31]–[33].

There are many examples of implementation of these technologies in consumer products nowadays, such as Tesla Autopilot, which Tesla Motors describes as a full self-driving hardware that is installed on all their cars that is equipped with all these sensors [34]:

- Eight surround cameras that provide 360 degrees of visibility around the car at up to 250 meters of range.
- Twelve ultrasonic sensors for complementing this vision allowing for detection of hard and soft objects.
- A forward-facing radar that provides additional data.

They ensure that this system, present in all their cars, has a safety level substantially greater than that of a human driver.

Another example is the new DJI Phantom 4 drone, which is equipped with a revolutionary *Vision Positioning System* that uses dual cameras and ultrasonic sensors and allows the robot to be operated either by remote control or autonomously. When operating autonomously, it has an operating mode called *ActiveTrack* that allows the aircraft to track a moving subject without a separate GPS tracker and uses its Obstacle Sensing System that constantly scans for obstacles in front of it, allowing it to avoid collisions by going around, over or hovering [35].

## 6.1 Obstacle detection system

In contrast to the examples described above, the shopping trolley developed is characterized by simple, easy to program and low-cost electronics and sensors. That is why two ultrasonic ranging HC-SR04 modules, manufactured by *Multicomp* and supplied by *Farnell* at a cost of £2.49 each, are used for detecting obstacles.

The main features of these sensors are [36]:

- Low-cost precise and non-contact distance measurement.

- Perform measurements between moving or stationary objects.
- Simple pulse in/out communication.
- Distance 2-400 cm measurement range, 3mm accuracy.
- 40 kHz working frequency.
- 5 V supply

They are pointing 45 degrees in both directions , where 0 degrees is the front of the robot. In this way, they can sense obstacles on both sides as it is getting closer to them. However, one of the limitations of this configuration is that they are not able to detect obstacles just in front of the robot. This is why the assumption that was made in Section 4.2 is referred again. It stated that the robot and the target person were going to be close enough so it can be assumed that there is not going to be any obstacle just in front of the robot. This assumption could be ignored by using a computer-vision based tracking system, which would be able to locate the target person and obstacles at the same time.

## 6.2 Ultrasonic sensor measure function

HC-SR04 modules have four pins. Two of them are for power supply pins, 5 V and ground. The other two are called *trigger* and *echo*: *Trigger* is used for applying a pulse that starts the measurement procedure and *echo* is used to measure the pulse that the module generates depending on the distance sensed [36], [37].

Applying a 10 μs duration pulse to the *trigger* pin of the ultrasonic module makes the module send eight ultrasonic pulses at 40 kHz. At this moment, the module set the *echo* pin to *HIGH* until an *echo* signal is detected. If there is no *echo* signal, it means that there are no obstacles within the range (0-4000 mm). By measuring when the *echo* pin goes *HIGH* and *LOW* the distance to the obstacle can be known using the sound speed, which is around 340 m/s, as describes in Equation (6.1) , Equation (6.2) and Figure 14.

$$distance = \frac{time\left(echo_{pulse}\right)}{2} \cdot sound_{speed} \tag{6.1}$$

$$distance(mm) = \frac{time\left(echo_{pulse}\right)\,(\mu s)}{2 \cdot 10^{3}} \cdot 340\,\left(\frac{m}{s}\right) \tag{6.2}$$

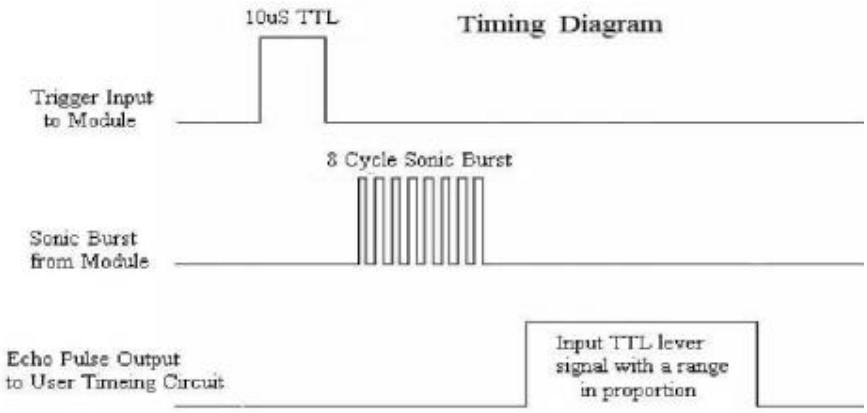

10uS TTL
Timing Diagram
Trigger Input to Module
8 Cycle Sonic Burst
Sonic Burst from Module
Echo Pulse Output to User Timeing Circuit
Input TTL lever signal with a range in proportion

*Figure 14. HC – SR04 operation: Trigger and echo pulses graph*

*(Source: Ultrasonic Ranging Module HC - SR04 datasheet)*

Figure 15 shows the flowchart of the interrupt used for doing the measurements (Figure 15.a) and the ultrasonic (US) measurement function flowchart used within the interrupt (Figure 15.b).

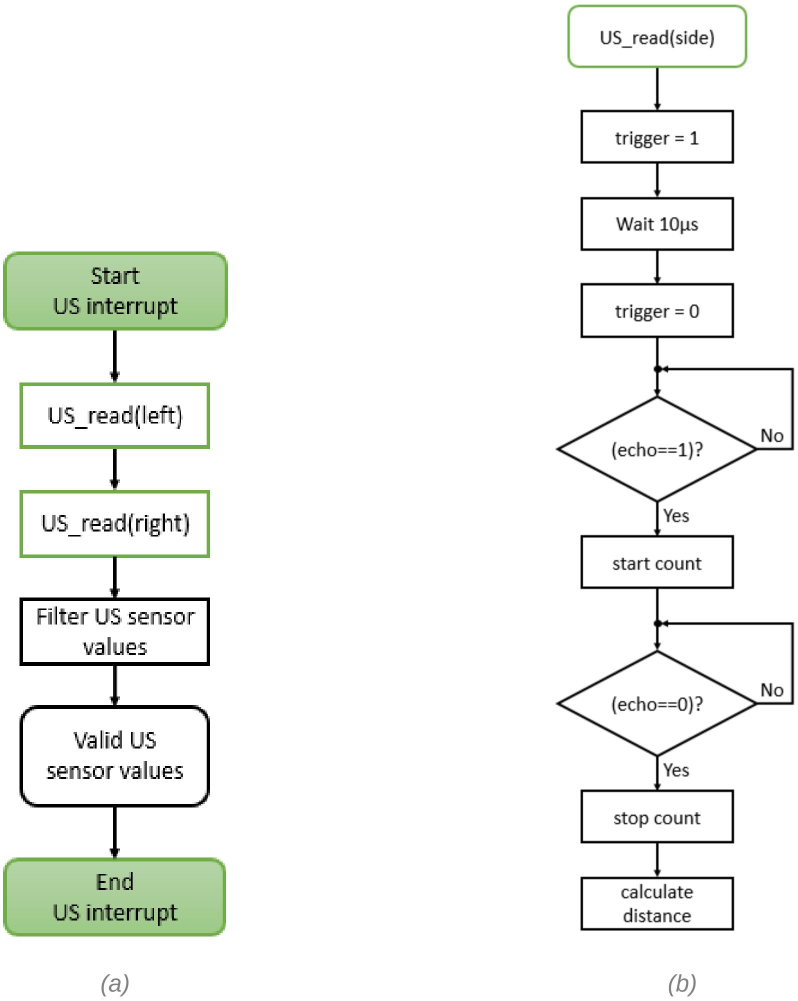


*Figure 15. Ultrasonic (US) sensor measurements (a) Trigger and echo pulses graph (b) Interrupt flowchart (c) Ultrasonic measurement function flowchart*

It can be noticed that within the interrupt a data filtering process takes place. This filtering process uses exactly the same *simple moving average filter* algorithm described in Section 5.2.

The interrupt that used for interfacing the ultrasonic sensors has been implemented in the microcontroller by using timer interrupts as with the IR sensor interrupt described in Section 5.2. A Ticker object called *Obstacle_avoidance* has been defined for the US sensor measurements with a 20 Hz frequency and with a function attached that reads the values of the sensors and applies a filter to the data. This frequency ensures a proper system behaviour as it is not only slow enough for allowing the microcontroller to do all the calculations within the interrupts but also fast enough for detecting fast-approaching obstacles.

Additionally, a voltage divider has been implemented using two resistors for converting the 5 V *echo* pulse to a pulse compatible with the microcontroller: 3 V, for example. The value of these resistors have been calculated as shown in Equations (6.3) and (6.4).

$$V_{out} = V_{in} \cdot \frac{R2}{R1 + R2} \qquad 3\,V = 5\,V \cdot \frac{R2}{R1 + R2} \tag{6.3}$$

$$\therefore\ R1 = \frac{5}{3}R2 - R2 = \frac{2}{3}R2 \quad \rightarrow \quad R2 = 1.5\,k\Omega \quad \therefore\ R1 = 1\,k\Omega \tag{6.4}$$

## 6.3 Collision avoidance algorithm

As mentioned at the beginning of this chapter, the majority of autonomous vehicles include a collision avoidance system for safety reasons and for preventing accidents. In Section 5.2, the main program flowchart was shown and it stated that, whenever an obstacle was detected while the target person was in the following range, the robot ran a collision avoidance algorithm.

This statement means that the collision avoidance algorithm is only performed when there is a person to follow. Taking into account that the ultrasonic sensors are pointing 45 degrees from the source of the robot, if there is a potential collision to avoid on both sides, the target has to be almost in front of the robot. It means that the target cannot be on the side of the robot and the possible target locations are reduced to three as when there were only two front infrared sensors.

Figure 16 shows the flowchart of the collision avoidance algorithm, which explains its operation graphically. At the beginning of the algorithm, the values of both ultrasonic and infrared sensors are checked and the robot performs different actions depending on these values.

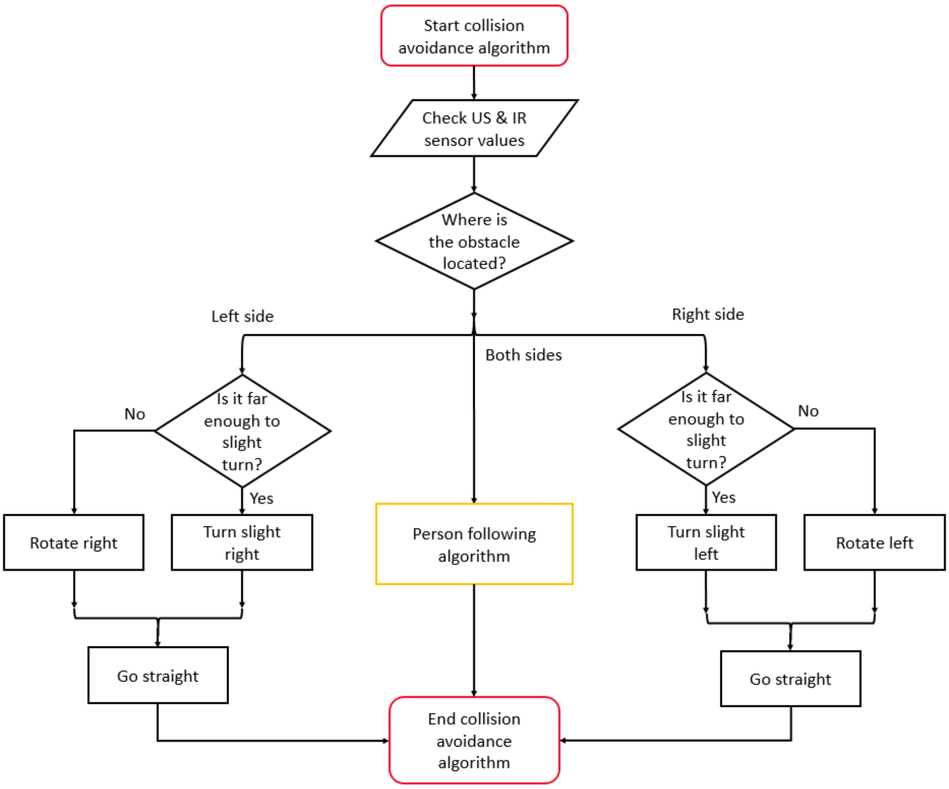

Start collision avoidance algorithm
Check US & IR sensor values
Where is the obstacle located?
Left side
Right side
Both sides
Is it far enough to slight turn?
No
Yes
Is it far enough to slight turn?
No
Yes
Rotate right
Turn slight right
Person following algorithm
Turn slight left
Rotate left
Go straight
Go straight
End collision avoidance algorithm

*Figure 16. Collision avoidance algorithm: Flowchart*

Depending on the ultrasonic sensors' values, the algorithm is able to differentiate between three states:

- **There is an obstacle on both sides:** As stated before, if the robot detects obstacles on both sides, the target person has to be almost in front of the robot, in its vision field. Therefore, the robot performs the person following algorithm taking into account that it can only make a slight turn depending on the target person location but not a sharp rotation.
- **There is an obstacle on the left side:** Whenever the robot detects that there is an obstacle on the left side or there are obstacles on both sides but the one on the left side is significantly closer than the one on the right side. In order to avoid the collision, it first tries to make a slight right if there is enough distance between the obstacle and itself. However, if there is not enough space for it, the robot will rotate to the right. After performing one of the two turns described, the robot goes straight for a short period of time for finishing this avoiding manoeuvre.
- **There is an obstacle on the right side:** It performs a symmetrical manoeuvre to the one described before.

# 7 PCB design

A printed circuit board, PCB, is a board that has lines and pads that electrically connects various points together. It allows signals and power to be routed between physical devices without using wires. On a PCB, solder has two functions. The first one is to create an electrical connection between a component and one or more paths on the board. The second one is to mechanically attach the component to the board [38].

The main advantages of using a PCB instead of a prototyping board circuit are:

- As the robot will be moving around during all its life, the components remain attached to the board even if it moves sharply.
- On the PCB, components are placed in such a way as to guarantee that the electronic noise, which can create a performance degradation of the system, is minimized.
- Components are uniformly distributed on the board and it looks more elegant and professional than having wires on the board.

After having designed the power stage for controlling the motors along with all the sensors, which act as inputs for calculating the speed of these motors, it has been considered to design a PCB for containing all the electronic components present on the robot. For that purpose, a schematic of the whole system has been created using *Cadence Capture CIS* software as shown in Appendix A. This schematic file has been transferred into *Cadence PCB Editor* software for designing the layout of the printed circuit board, whose production files can be found in Appendix A.

Furthermore, Figure 18.a shows how the PCB has been implemented on the robot in comparison with the previous configuration using a stripboard and wires (Figure 18.b), which is very helpful when prototyping because it permits to easily solder and unsolder components but looks less professional.

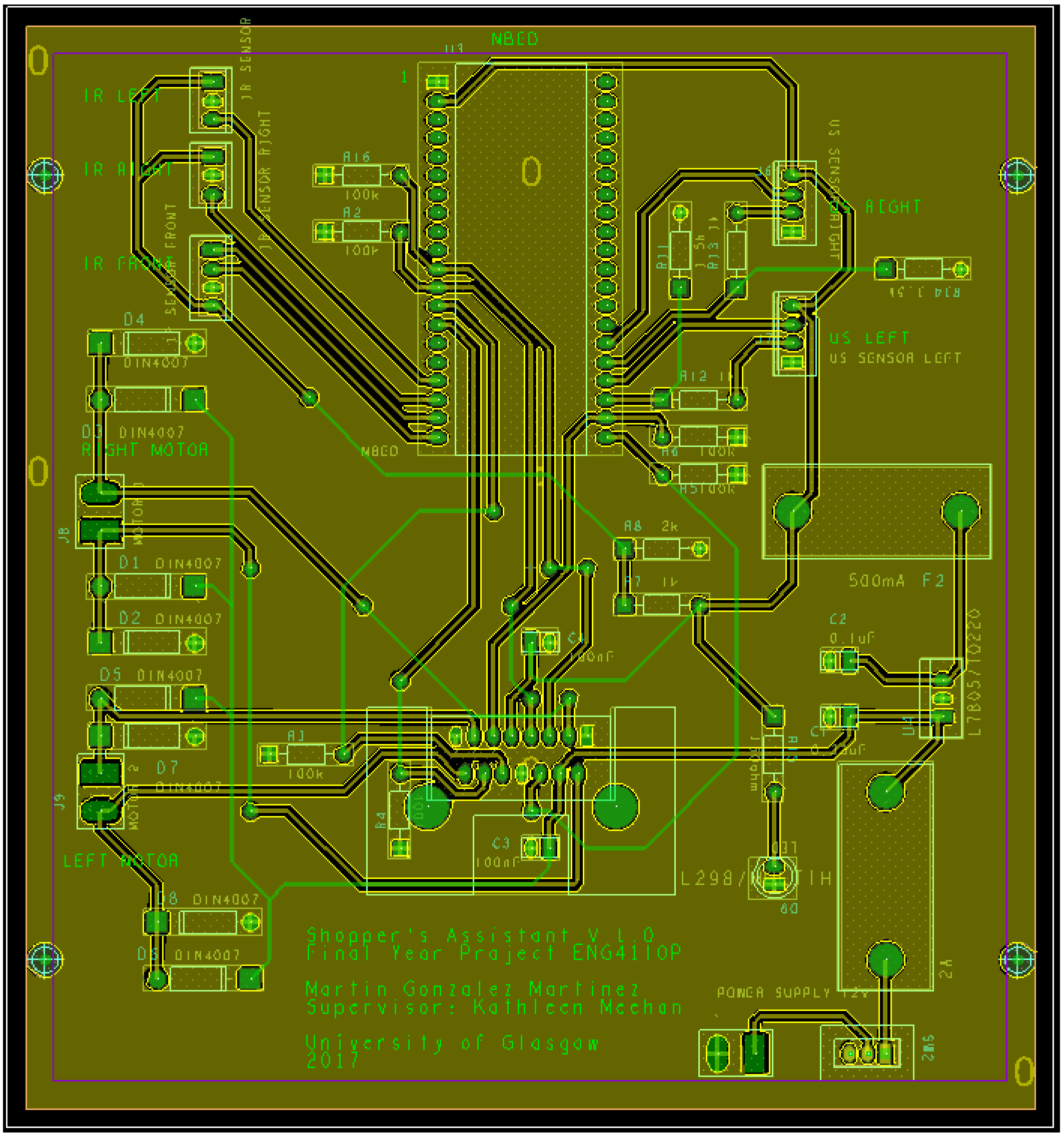
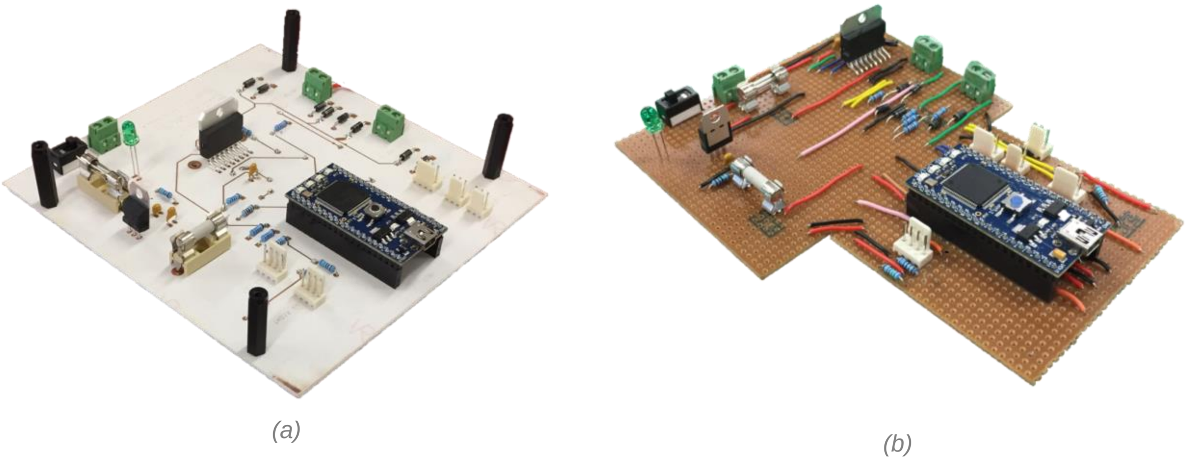


*Figure 17. PCB Layout designed using Cadence PCB Editor*



*(a)*

*(b)*

*Figure 18. Power stage implementation (a) PCB-based power stage (b) Stripboard-based power stage*

# 8 Real size shopping trolley design

Now that the prototype has been successfully designed, it is time to come back to the initial scope of the project, which was to design a semi-autonomous robot trolley able to follow a human, transport a load up to 20 kg and to avoid obstacles.

In the previous chapters, a design process has been carried out for developing the trolley prototype. However, the aim of the project was to design a real size trolley and not a prototype, and this is what this chapter is aimed at.

## 8.1 Design specifications

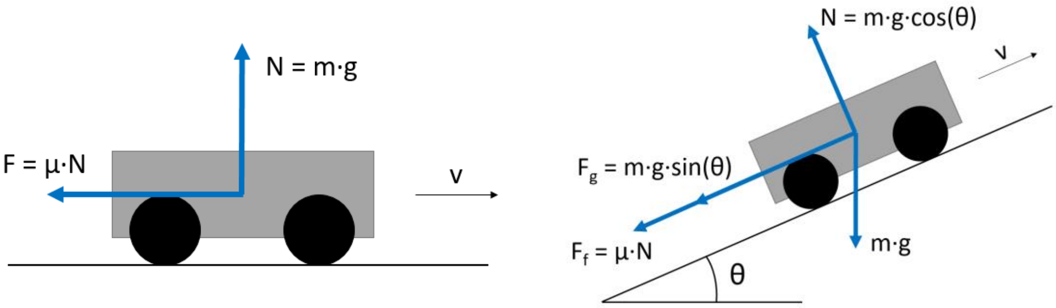
For starting the components' dimensioning, the design specifications of the trolley have to be defined:

- **Trolley mass:** It corresponds to the mass of the structure and all the parts excluding the load mass. It has been estimated to be 10 kg.
- **Maximum load mass:** As it was said in Chapter 1, the trolley has to be able to transport a load up to 20 kg.
- **Maximum speed:** As the trolley has to follow a person walking, the maximum speed has been set to 2 m/s, approximately 7 km/h, as the typical walking speed is 1.44 m/s [39], [40].
- **Maximum acceleration:** To ensure a behaviour in accordance with the trolley's aim, a value of 0.75 m/s$^2$ has been established as maximum acceleration based on previous person-following studies [41].
- **Maximum slope angle:** Considering the application scope of the trolley, which is going to be used in supermarkets that typically have a flat surface.
- **Overall efficiency:** It corresponds to the overall efficiency of the motor, transmission and other components participating on the movement
- **Wheel diameter:** These values have to be defined beforehand, because the motors' selection is directly linked to it. Following the trend of actual shopping trolleys in supermarkets, the diameter of the wheels has been defined to a value of 20 cm.

- **Battery life:** Considering the average time that a shopper spends in a supermarket along with the possibility that there is not enough time to charge the trolley between two customers, it has been estimated a required battery life of around 1.5 hours, which permit to finish one shopping and even a second one if it has some charge time.

## 8.2 Motors

When a motorised trolley moves on a surface, it has to overcome the friction of the wheels with the ground along with the internal friction inside the motor. In addition, if the trolley starts to climb a hill, it also has to overcome the gravity force as shown in Figure 19.



(a)

(b)

Figure 19. Forces that the shopping trolley need to overcome (a) Flat surface. (b) Sloping surface

If the trolley moves on a flat surface, the friction force it has to overcome is described in Equation (8.1), where μ is the coefficient of friction between the wheel and the surface and N is the normal force.

$$F = \mu \cdot N \tag{8.1}$$

However, if the trolley moves over inclined plane, both gravity and friction forces must be considered as shown in Equations (8.2) and (8.3).

$$F_f = \mu \cdot m \cdot g \cdot \cos(\theta) \tag{8.2}$$

$$F_g = m \cdot g \cdot \sin(\theta) \tag{8.3}$$

Equation (8.4) shows the required torque to overcome the friction, where T is the torque required and R is the wheel radius.

$$T = F_f \cdot R \qquad \therefore F_f = \frac{T}{R} \tag{8.4}$$

When the trolley moves forward, the frictional force direction is forward at the contact point of the wheel with the ground, just opposite to the direction of the velocity of this portion of the wheel, which is backwards as shown in Figure 20.

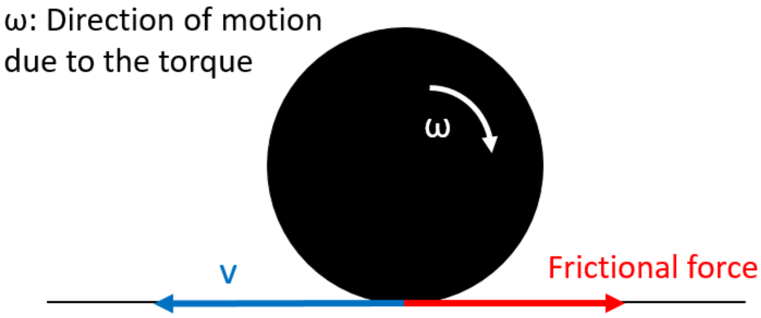


Figure 20. Forces at the contact point of the wheel with the ground

The complete forces equation is shown in Equation (8.5). It is easy to reorder it to isolate the total torque needed for move the trolley (Equation (8.6)).

$$\frac{T}{R} - m \cdot g \cdot \sin(\theta) = m \cdot a \tag{8.5}$$

$$T = \big(a + g \cdot \sin(\theta)\big) \cdot m \cdot R \tag{8.6}$$

Considering that four motors are used for moving the trolley, the motor torque required to drive the trolley will be equal to quarter of the total required torque when it is moving forward. However, if there is a turn, two of the motors do not contribute much and it can be assumed that the maximum motor torque required to drive the trolley is equal to half the total required torque. In addition, the internal friction present on the motor must be considered and introduced into the required torque equation (Equation (8.7), where $\eta$ is the overall efficiency of the motor, transmission and other components participating on the movement).

$$T = \frac{1}{\eta} \cdot \frac{\big(a + g \cdot \sin(\theta)\big) \cdot m \cdot R}{2} \tag{8.7}$$

Replacing the variables values in the equation with the design parameters described in the previous section, the torque required by each motor is calculated in Equation (8.8).

$$T = \frac{1}{0.8} \cdot \frac{\left(0.75\ \frac{m}{s^2} + 9.81\ \frac{m}{s^2} \cdot \sin(10°)\right) \cdot 30\ kg \cdot 0.10\ m}{2} = 4.6\ N \cdot m \qquad (8.8)$$

However, motors cannot be chosen yet. A speed constrain is needed, the maximum angular speed of the motor has to be calculated considering the maximum speed of the trolley as shown in Equations (8.9) and (8.10).

$$\omega \left(\frac{rad}{s}\right) = \frac{v\left(\frac{m}{s}\right)}{2 \cdot \pi \cdot R\ (m)} \qquad (8.9)$$

$$\omega\ (rpm) = \frac{v\left(\frac{m}{s}\right)}{2 \cdot \pi \cdot R\ (m)} \cdot \frac{1\ (rev)}{2 \cdot \pi\ (rad)} \cdot \frac{60\ (s)}{1\ (\min)} \qquad (8.10)$$

Replacing the variables values in Equation (8.10) with the design parameters described in the previous section, the maximum angular speed of the motor can be calculated (Equation (8.11)).

$$\omega = \frac{2\frac{m}{s}}{2 \cdot \pi \cdot 0.1\ m} \cdot \frac{1\ rev}{2 \cdot \pi\ rad} \cdot \frac{60\ s}{1\ \min} = 30.4\ rpm \qquad (8.11)$$

Considering these two parameters that have been calculated, the maximum angular speed and the torque required, the required usable power output needed is calculated using Equation (8.12) [42], [43].

$$P(W) = \frac{2 \cdot \pi}{60} \cdot T\ (N \cdot m) \cdot \omega\ (rpm) \qquad (8.12)$$

$$P = \frac{2 \cdot \pi}{60} \cdot 4.6\ \cdot 30.4 = 14.64\ W \qquad (8.13)$$

The motors selected are the *CROUZET AUTOMATION 80807020 Geared DC Motor, Solid Metal Gears, 15.6 W, 62 rpm, 5 N-m* supplied by Farnell, whose main features are [44]:

- Output speed: 62 rpm.
- Maximum torque: 5 Nm.
- Gearbox shaft: Ø 8 mm.
- Maximum usable power: 15.6 W.
- Weight: 800 g.
- Voltage: 24 V.
- Rated current: 0.81 A.

## 8.3 Battery

Considering the power consumption of the components stated on Chapter 3, and updating the value of current drawn by the motors, Equations (8.14) and (8.15) show the maximum and the estimated average currents of the trolley.

$$I_{max} = 4 \cdot 0.81\ A + 2 \cdot 0.015\ A + \ 0.2\ A\ \cong 3.5\ A \tag{8.14}$$

$$I_{avg} = 4 \cdot 0.81 \cdot 0.7 \cdot 0.7\ A + 2 \cdot 0.015\ A + \ 0.2\ A\ \cong 1.8\ A \tag{8.15}$$

With these values in mind, the estimated battery capacity needed to achieve the battery life goal can be calculated as shown in Equations (8.16) and (8.17).

$$Battery\ life = \frac{Battery\ capacity}{Average\ current} = \frac{Battery\ capacity}{I_{avg}} \tag{8.16}$$

$$\therefore\ Battery\ capacity = I_{avg}\ \cdot Battery\ life = 1.8A \cdot 1.5h = 2.7Ah \tag{8.17}$$

Comparing the available batteries on some online electronic components' suppliers, the *ENIX ENERGIES AMH9080 24 V 3.2 Ah Ni-MH Battery*, supplied by Farnell, and whose main features are [45], [46]:

- Voltage: 24 V.
- Capacity: 3.2 Ah.
- Dimensions: 87 mm x 120 mm x 97 mm.
- Weight: 1.4 kg.

The estimated battery life can be calculated as shown in Equation (8.18).

$$Battery\ life = \frac{Battery\ capacity}{Average\ current} = \frac{3.2\ Ah}{1.8\ A} = 1.78\ hours \cong 1\ h\ 45\ min \tag{8.18}$$

It has been considered the idea of adding a battery indicating system that permit the user to know the remaining capacity. It could be easily designed using operational amplifiers, some voltage divider resistors and some LEDs for indicating the status.

## 8.4 Other electronic components

Some other components need to be updated due to the new changes introduced in the electronic components.

- The H-bridge do not need to be updated as the L298n has a maximum current output of 2 A per channel (2 x 0.81 A = 1.62 A < 2 A) and a maximum input voltage of 46 V (46 V < 24 V).
- The protection diodes have to be changed. 1N4007 diodes, which have a maximum average forward rectified current of 1.0 A, were used for the prototype, but now more current is needed. Instead of the 1N4007, eight *Vishay BYW54-TR* should be used for the real-size trolley. These diodes permit a maximum forward current of 2 A and a maximum reverse voltage of 600 V.
- A 4 A fuse should replace the 2.5 A fuse on the PCB.
- The L7805CV voltage regulator do not need to be replaced, as it can be supplied with up to 35 V.
- The microcontroller, its protection resistors and all the sensors can remain in the real-size trolley exactly as they were placed on the prototype.

## 8.5 Real-size trolley implementation

### 8.5.1 Structure

The structure of the trolley will be a redesign of the actual shopping trolleys that can be found in a supermarket, with the addition of a compartment on their lower part for the electronic components.

### 8.5.2 Sensors

On the real size trolley, the sensors will be placed at different heights than on the prototype. This placement will permit a better interfacing with the human, in the case of IR sensors, and a better obstacle detection when talking about ultrasonic sensors.

The ultrasonic sensors would be placed at a lower height than the infrared ones. This will allow the target person to place the IR transmitter at a higher height and the signal would not be interrupt as it happened sometimes with the prototype version. The ultrasonic sensors will be placed at a similar height than on the prototype, permitting the trolley to detect small obstacles such as boxes or kids.

All the components composing the real-sized trolley are outlined on Appendix D.

# 9 Conclusions

In this chapter, the objectives stated at the beginning of the project are going to be recalled along with an analysis of the results achieved and a description of potential improvements that could be carried out during further studies.

## 9.1 Conclusions

The main objective of the project was to design an autonomous robot trolley that is able to follow a human within a distance of around 3 feet, transport a load up to 20 kg and avoid collisions. Regarding the large load the trolley had to transport, this objective was changed soon and it was decided to build a smaller prototype and then choose the components needed to build the real size trolley.

To do this, a power stage has been designed using a L298 dual full h-bridge for driving each side of the motors of the robot independently and some protection components such as diodes, fuses and protection resistors for the microcontroller. The prototype is supplied using a 12 V NiMH battery that allows it to work for more than 2 hours when fully charged.

A person tracking system and a person following algorithm have been implemented using infrared. For that, an infrared transmitter device, which has to be carried by the person that the robot has to follow, has been developed along with several infrared sensors using infrared placed on the robot. Two HC-SR04 ultrasonic modules are used for detecting obstacles and avoid collisions using the especially designed collision avoidance algorithm. An ARM mbed NXP LPC1768 microcontroller coordinates every sensor measure and control calculation. At the end of the designing stage, a PCB has been designed to contain all the electronic parts on a professional-looking board.

Once the prototype was designed and tested, the necessary components for building the real-size shopping trolley have been selected, thus concluding the design stage and the project itself.

The completion of this project has had a significant personal impact, not only because I had the opportunity to carry out my Final Year Project how at a foreign university and gain an awareness of different cultures and working methods but also because I had the opportunity to accomplish the wide variety of challenges that this project involved.

## 9.2 Further studies

Even if the objectives of the project have been achieved successfully, due to the time constraints that these type of academic projects have, there are many improvements and add-ons that can be done to the prototype develop on this Final Year Project.

In view of the latest technological improvements introduced in the supermarket industry, the following improvements have been identified as potential aims of further studies:

- Implement a combined *person tracking* and *obstacle detection* system using Computer Vision. As it was mentioned in Chapter 4, person tracking system can be implemented using a Raspberry Pi and a Pi Camera module. In Chapter 6, it was stated that obstacles could be detected using computer vision. Then, it can be developed a combined system using only cameras for detection of both target person and obstacles. It would also mitigate sunlight sensibility of the system as the sun emits some infrared light and it has some influence on the robot's behaviour.

  During the final stage of the project, some tests have been done using Raspberry Pi and OpenCV software. However, due to the short time allocated to the project and the simplicity of the infrared-based tracking system, it has not been implemented on the prototype.

- Implement a speed control for the motors, including speed sensors and control loops to ensure proper functioning of the trolley in all load and environmental conditions.

- Design a new power stage to drive the four motors independently instead of driving them as couples. As stated in Chapter 3, each side motors are driven as if there were only one. It would be interesting to test the performance of the trolley when all the four motors are driven independently.

- Add some type of communication between the trolley and the person. It can be implemented either with gestures, if using a camera, or with a button on the transmitter and a wireless communication if using the actual system. It would add a wide new variety of applications to the trolley.

# 10 References

[1] G. Lise Puerini, D. Kumar, and S. Kessel, ‘Transitioning Items From a materials handling facility’, US Patent Number US 2015/0012396 A1, 2015.

[2] GeekWire, ‘How “Amazon Go” works: The technology behind the online retailer’s groundbreaking new grocery store’. [Online]. Available: http://www.geekwire.com/2016/amazon-go-works-technology-behind-online-retailers-groundbreaking-new-grocery-store/. [Accessed: 09-Jan-2017].

[3] Carlo Ratti Associati website, ‘Supermarket of the Future’. [Online]. Available: http://www.carloratti.com/project/supermarket-of-the-future/. [Accessed: 15-Feb-2017].

[4] Gizmodo.com, ‘This Smart Cart Was Just Named the Year’s Best American Invention’. [Online]. Available: http://gizmodo.com/dyson-just-crowned-this-smart-cart-as-the-years-best-am-1635914745. [Accessed: 10-Nov-2016].

[5] James Dyson Award, ‘Uplift’. [Online]. Available: http://www.jamesdysonaward.org/projects/uplift/).

[6] M. D. Atchley, D. R. High, and D. C. Winkle, ‘Shopping facility assistance systems, devices and methods’, US Patent Number US 2016/0260161 A1, 2016.

[7] ‘Alsrobot Web Page’. [Online]. Available: www.alsrobot.com.

[8] ARM mbed, ‘mbed NXP LPC1768 Getting Started’. [Online]. Available: https://developer.mbed.org/handbook/mbed-NXP-LPC1768-Getting-Started. [Accessed: 07-Oct-2016].

[9] N. Mohan, T. M. Undeland, and W. P. Robbins, *Power Electronics. Converters, Applications and Design*, 2nd ed. 1995.

[10] ST Microelectronics, ‘L298 Dual full-bridge driver Datasheet’, *Current*, no. October. pp. 1–13, 2000.

[11] Mbed Cookbook website, ‘Motor.h library’. [Online]. Available: https://developer.mbed.org/cookbook/Motor. [Accessed: 15-Oct-2016].

[12] G. Vitale, D. Avola, G. L. Foresti, L. Cinque, and C. Massaroni, ‘A Multipurpose Autonomous Robot for Target Recognition in Unknown Environments’, *IEEE 14th*

*Int. Conf. Ind. Informatics*, pp. 766–771, 2016.

[13] J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale, and S. Shafer, ‘Multi-Camera Multi-Person Tracking for Easy Living’, *Proc. IEEE Int. Work. Vis. Surveill.*, 2000.

[14] A. Kapusta and P. Beeson, ‘Person Tracking and Gesture Recognition in Challenging Visibility Conditions Using 3D Thermal Sensing’, *25th IEEE Int. Symp. Robot Hum. Interact. Commun.*, pp. 1112–1119, 2016.

[15] A. Ess, B. Leibe, K. Schindler, and L. Van Gool, ‘A mobile vision system for robust multi-person tracking’, *IEEE Conf. Comput. Vis. Pattern Recognition, 2008. CVPR 2008.*, 2008.

[16] B. Hepp, N. Tobias, and O. Hilliges, ‘Omni-directional person tracking on a flying robot using occlusion-robust ultra-wideband signals’, *IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2016.

[17] A. Leigh, J. Pineau, N. Olmedo, and H. Zhang, ‘Person tracking and following with 2D laser scanners’, *Proc. - IEEE Int. Conf. Robot. Autom.*, no. June, pp. 726–733, 2015.

[18] M. S. Hassan, A. F. Khan, M. W. Khan, M. Uzair, and K. Khurshid, ‘A Computationally Low Cost Vision Based Tracking Aigorithm for Human Following Robot’, *2nd Int. Conf. Control. Autom. Robot.*, pp. 62–65, 2016.

[19] L. Min, Y. Yang, Y. Liu, and Y. Leng, ‘Robust 3D Human Tracking based on Kinect’, *34th Chinese Control Conf.*, pp. 4615–4619, 2015.

[20] G. Marin, F. Dominio, and P. Zanuttigh, ‘HAND GESTURE RECOGNITION WITH LEAP MOTION AND KINECT DEVICES Department of Information Engineering , University of Padova’, *IEEE Int. Conf. Image Process.*, pp. 1565–1569, 2014.

[21] Kingbright, ‘L-7113IT HIGH EFFICIENCY RED LED Datasheet’, *Boards*, vol. 4, pp. 2–5, 2005.

[22] Stanley Electric, ‘AN5307B Through-hole IRED Datasheet’. 2012.

[23] Vishay Semiconductors, ‘TEFT 4300 Datasheet’, 2014.

[24] Wikipedia, ‘Roomba’. [Online]. Available: https://en.wikipedia.org/wiki/Roomba. [Accessed: 11-Mar-2017].

[25] ‘iRobot’. [Online]. Available: http://www.irobot.co.uk/. [Accessed: 11-Mar-2017].

[26] F. Vasconcelos and N. Vasconcelos, ‘Person-following UAVs’, *2016 IEEE Winter Conf. Appl. Comput. Vision, WACV 2016*, 2016.

[27] R. Bartak and A. Vykovsky, ‘Any object tracking and following by a flying drone’, *Proc. - 14th Mex. Int. Conf. Artif. Intell. Adv. Artif. Intell. MICAI 2015*, pp. 35–41, 2016.

[28] B. S. Choi, J. W. Lee, and J. J. Lee, ‘Effective target-following schemes for indoor mobile robots’, *2010 IEEE Int. Conf. Inf. Autom. ICIA 2010*, no. 2, pp. 666–671, 2010.

[29] S. Haykin, *Adaptive Filter Theory*, 3rd ed. 1996.

[30] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*. Prentice-Hall, 1985.

[31] D. K. D. Kim, S. M. S. Moon, J. P. J. Park, H. J. Kim, and K. Y. K. Yi, ‘Design of an Adaptive Cruise Control / Collision Avoidance with lane change support for vehicle autonomous driving’, *2009 ICROS-SICE Int. Jt. Conf.*, pp. 2938–2943, 2009.

[32] C.-F. Lin, J.-C. Juang, and K.-R. Li, ‘Active collision avoidance system for steering control of autonomous vehicles’, *IET Intell. Transp. Syst.*, vol. 8, no. 6, pp. 550–557, 2014.

[33] B. M. Sommer, N. N. Wang, T. R. Oriol, and J. P. Gasselin de Richebourg, ‘Collision Avoidance Of Arbitrary Polygonal Obstacles’, US Patent Number US20160358485A1, 2016.

[34] ‘Tesla Autopilot website’. [Online]. Available: https://www.tesla.com/en_GB/autopilot. [Accessed: 15-Jan-2017].

[35] ‘DJI Phantom 4’. [Online]. Available: http://www.dji.com/phantom-4.

[36] ‘Ultrasonic Ranging Module HC - SR04 Datasheet’. pp. 1–4, 2013.

[37] Cytron Technologies Sdn. Bhd., ‘HC-SR04 User ’ s Manual’. pp. 1–10, 2013.

[38] Learn.Sparkfun.com, ‘PCB Basics’. [Online]. Available: https://learn.sparkfun.com/tutorials/pcb-basics. [Accessed: 13-Mar-2017].

[39] R. V Levine and A. Norenzayan, ‘The Pace of Life in 31 Countries’, *J. Cross. Cult. Psychol.*, vol. 30, no. 2, pp. 178–205, 1999.

[40] Highways England, ‘Design Manual for Roads and Bridges (DMRB)’, 1992.

[41] S. Shaker, J. J. Saade, and D. Asmar, ‘Fuzzy Inference-Based Person-Following Robot’, *Int. J. Syst. Appl. Eng. Dev.*, vol. 2, no. 1, pp. 29–34, 2008.

[42] F. Mora, *Maquinas Electricas*, 5th ed. S.A. MCGRAW-HILL.

[43] Robot and machines design, ‘How to select a motor’. [Online]. Available: http://www.robot-and-machines-design.com/en/Articles/Mechanics/Tips-Guides/264-Motor-Selection-How-To-Select-A-Motor-How-to-Choose-A-Motor.html. [Accessed: 20-Mar-2017].

[44] CROUZET AUTOMATION, ‘CROUZET AUTOMATION 80807020 Geared DC Motor Datasheet’. pp. 1–4.

[45] ENIX ENERGIES, ‘AMH9080 24V 3.2Ah Ni-MH Battery Datasheet’. .

[46] Farnell, ‘ENIX ENERGIES AMH9080 24V 3.2Ah Ni-MH Battery’. [Online]. Available: http://onecall.farnell.com/enix-energies/amh9080/battery-ni-mh-24v-3-2ah-makita/dp/BT06140.

# APPENDICES

# Appendix A. Prototype final aspect

This appendix is aimed to show the final aspect of the prototype. Figure A.1 shows the final aspect of the prototype.

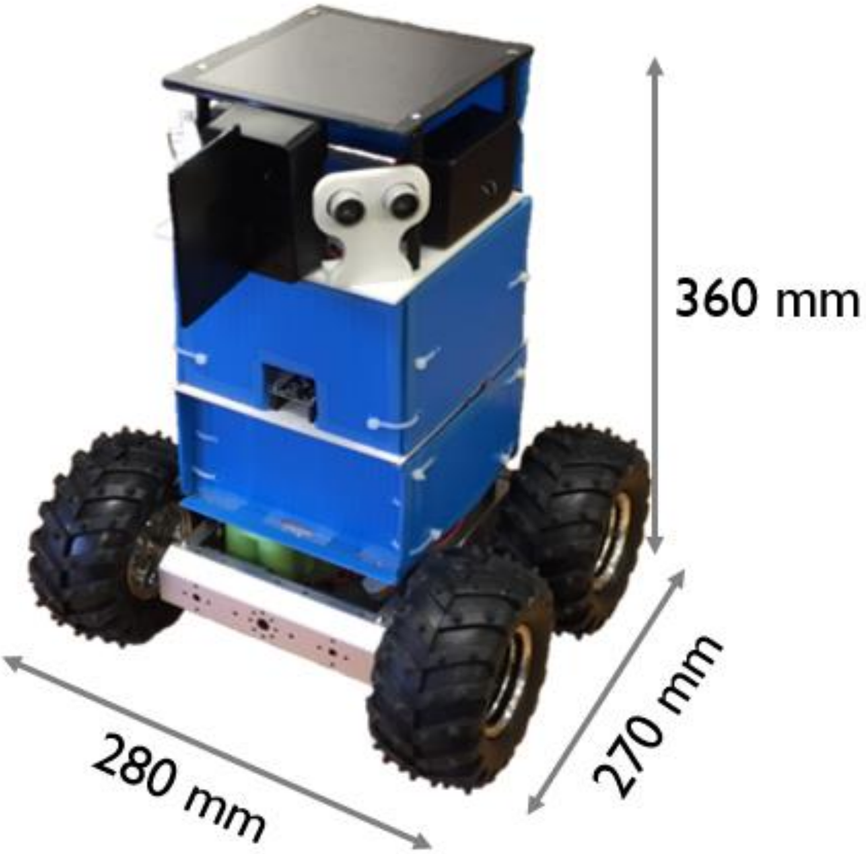
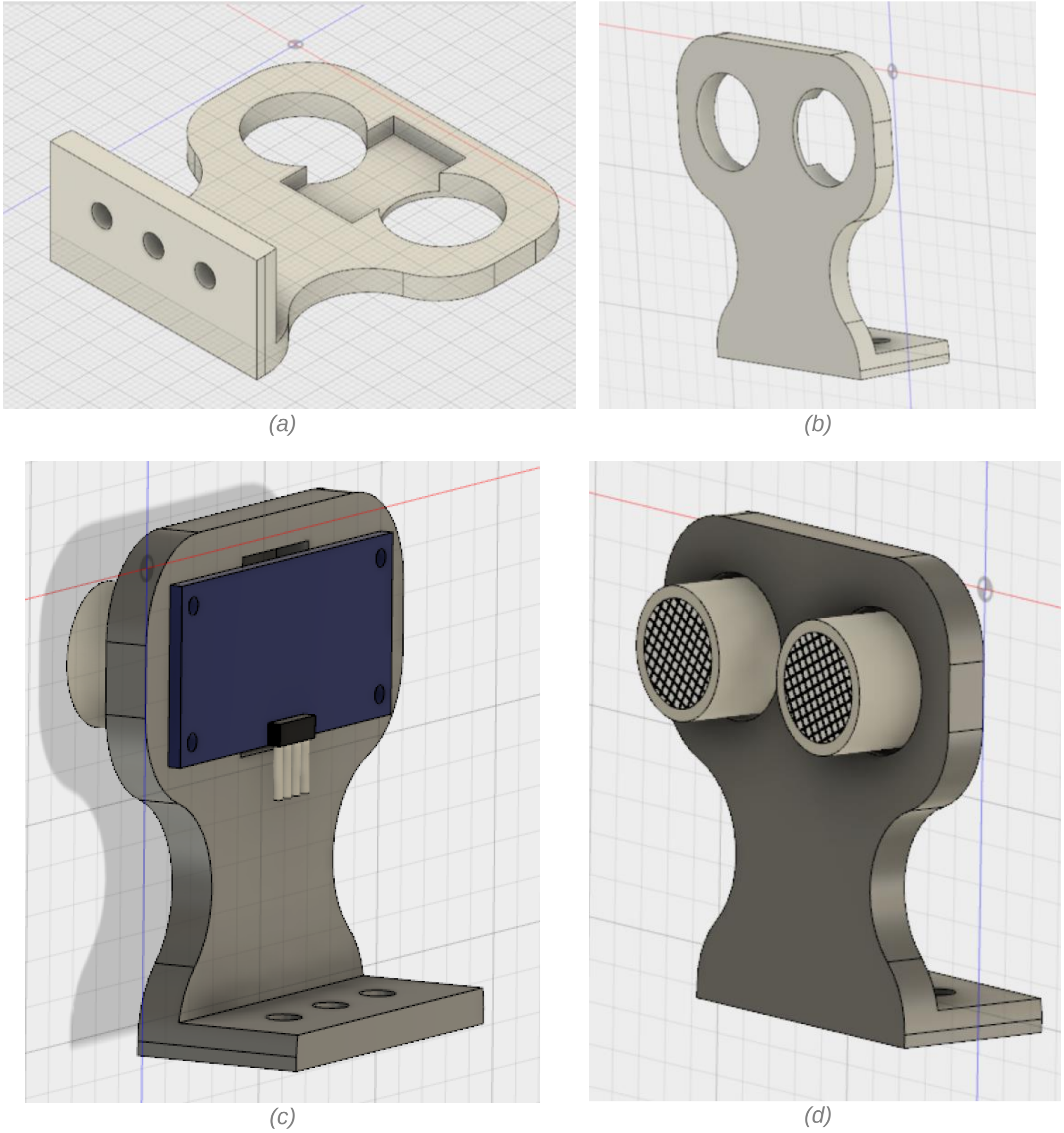


Figure A.1. Prototype final aspect

**Ultrasonic sensors’ support design**

As mentioned in Chapter 3, ultrasonic sensors are pointing 45 degrees in both directions from the source of the robot. In this way, they can sense obstacles on both sides as it is getting closer to them. However, it has not been mentioned how these sensors are placed on the robot. For that, two supports have been designed using the CAD software *Autodesk Fusion 360* (Figure A.2) and 3D printed using a *BQ Prusa i3 Hephestos* 3D Printer (Figure A.3).



(a)

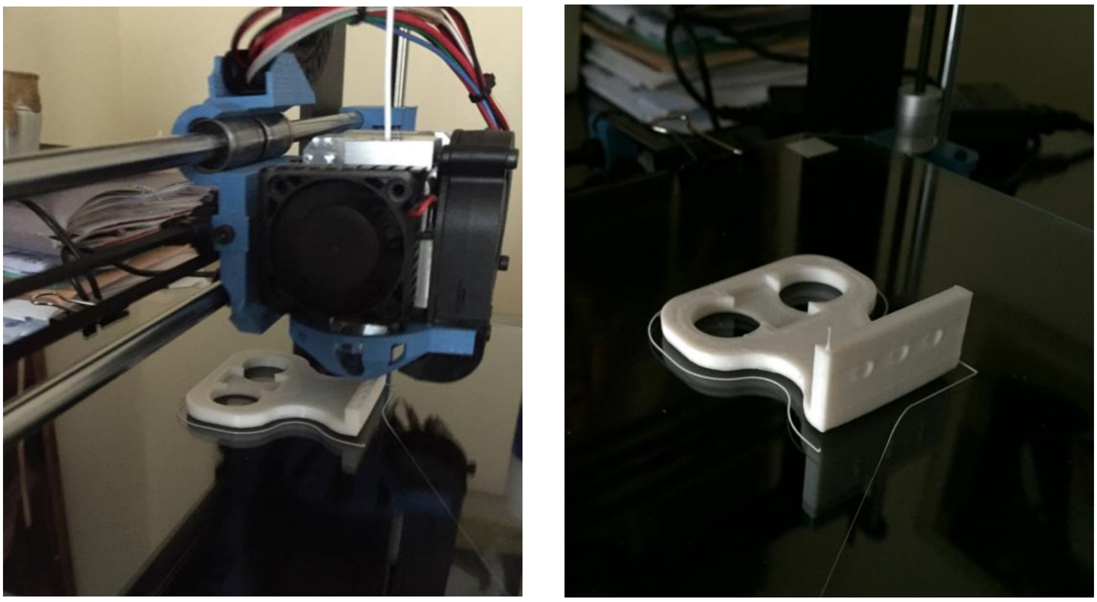
(b)

(c)

(d)

*Figure A.2. Ultrasonic sensor supports' CAD files*

Figure A.3 shows the manufacturing process with the 3D printer and Figure A.4 shows the final aspect of the supports with the sensors.



(a)

(b)

Figure A.3. 3D Printing process

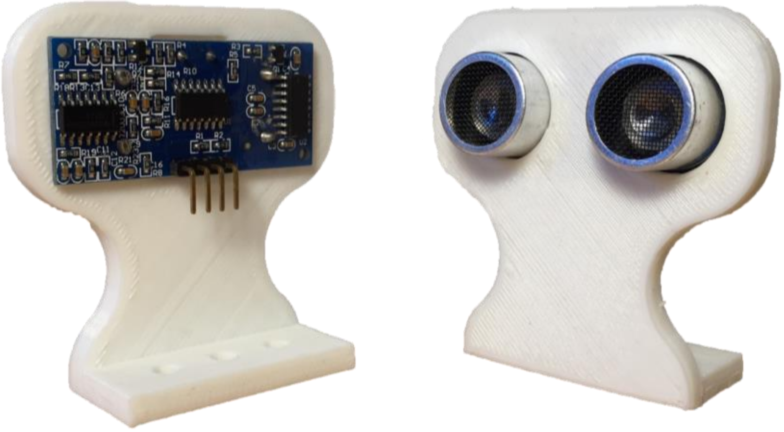


Figure A.4. Final aspect of the ultrasonic sensors' sup

# Appendix B. Schematic & PCB production files

This appendix contains the schematic files of the electronic components used for developing the small-size prototype (Figure B.1) along with the PCB production files (Figure B.2 and Figure B.3). The aim of this appendix is to not only clarify graphically how the power stage has been implemented on the PCB but also make available the files for anyone interested on building its own one. The PCB has a dimension of 140 mm x 150 mm.

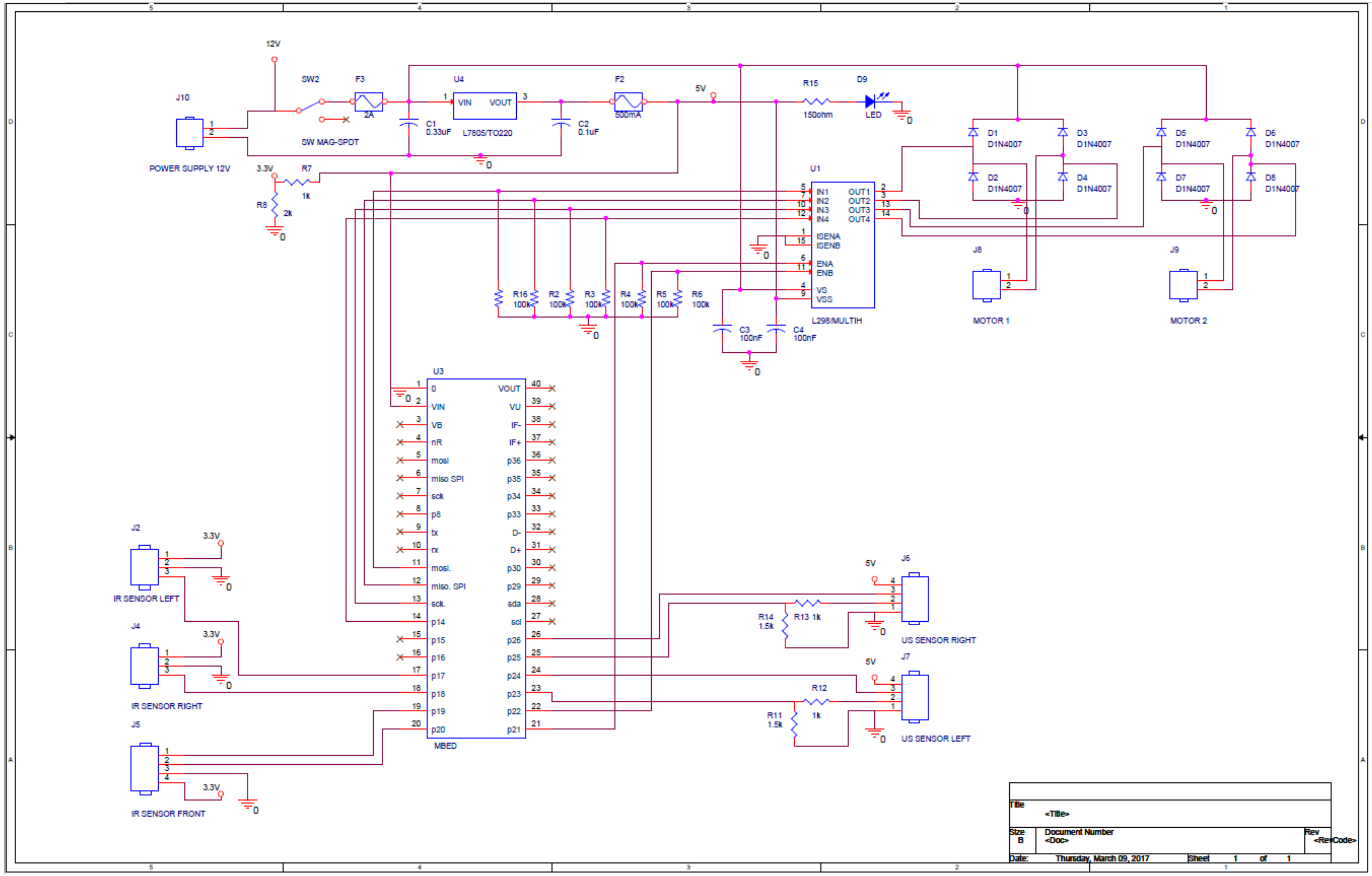


*Figure B.1. Printed Circuit Board schematic file*

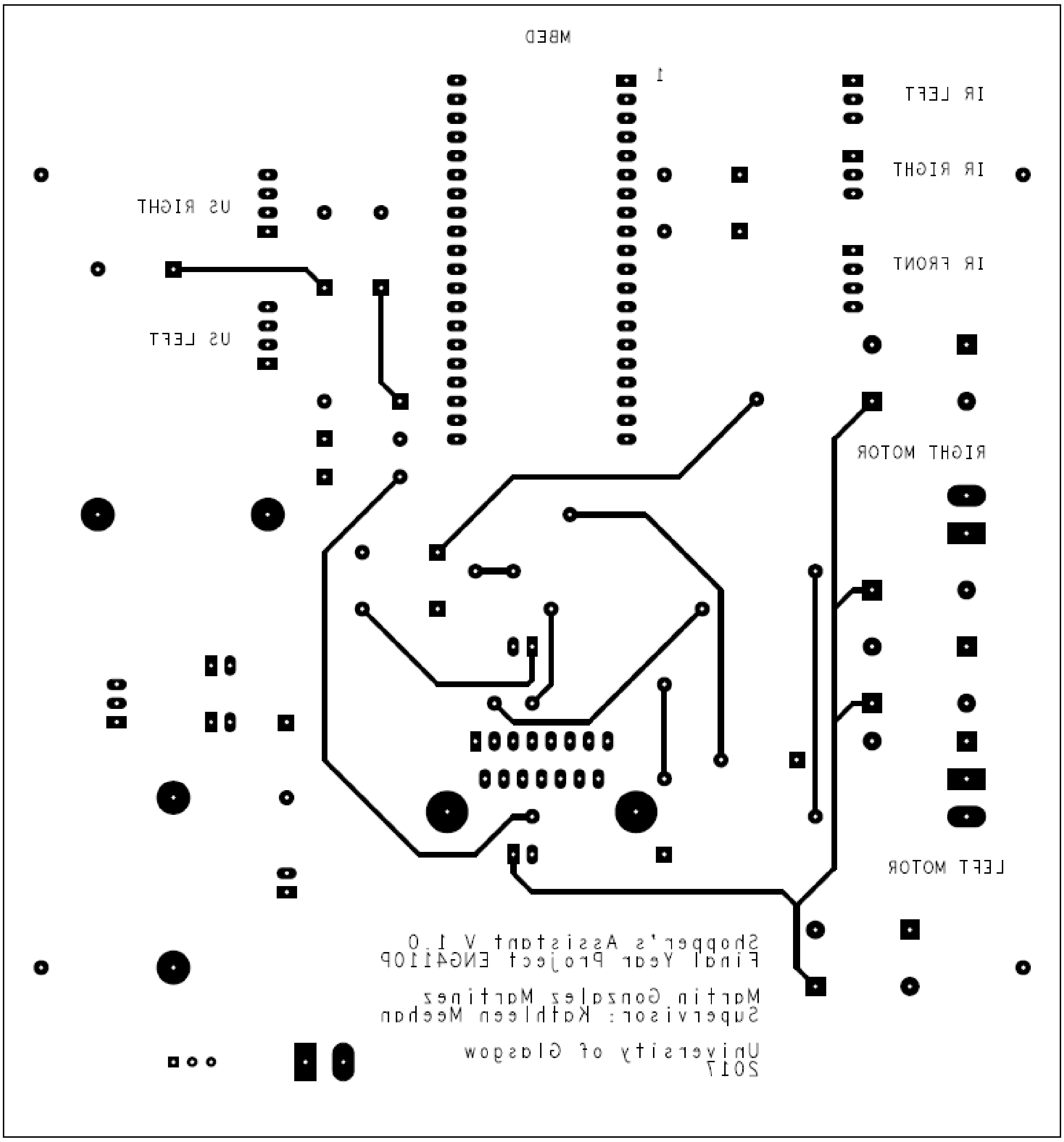


*Figure B.2. PCB top layer production file*

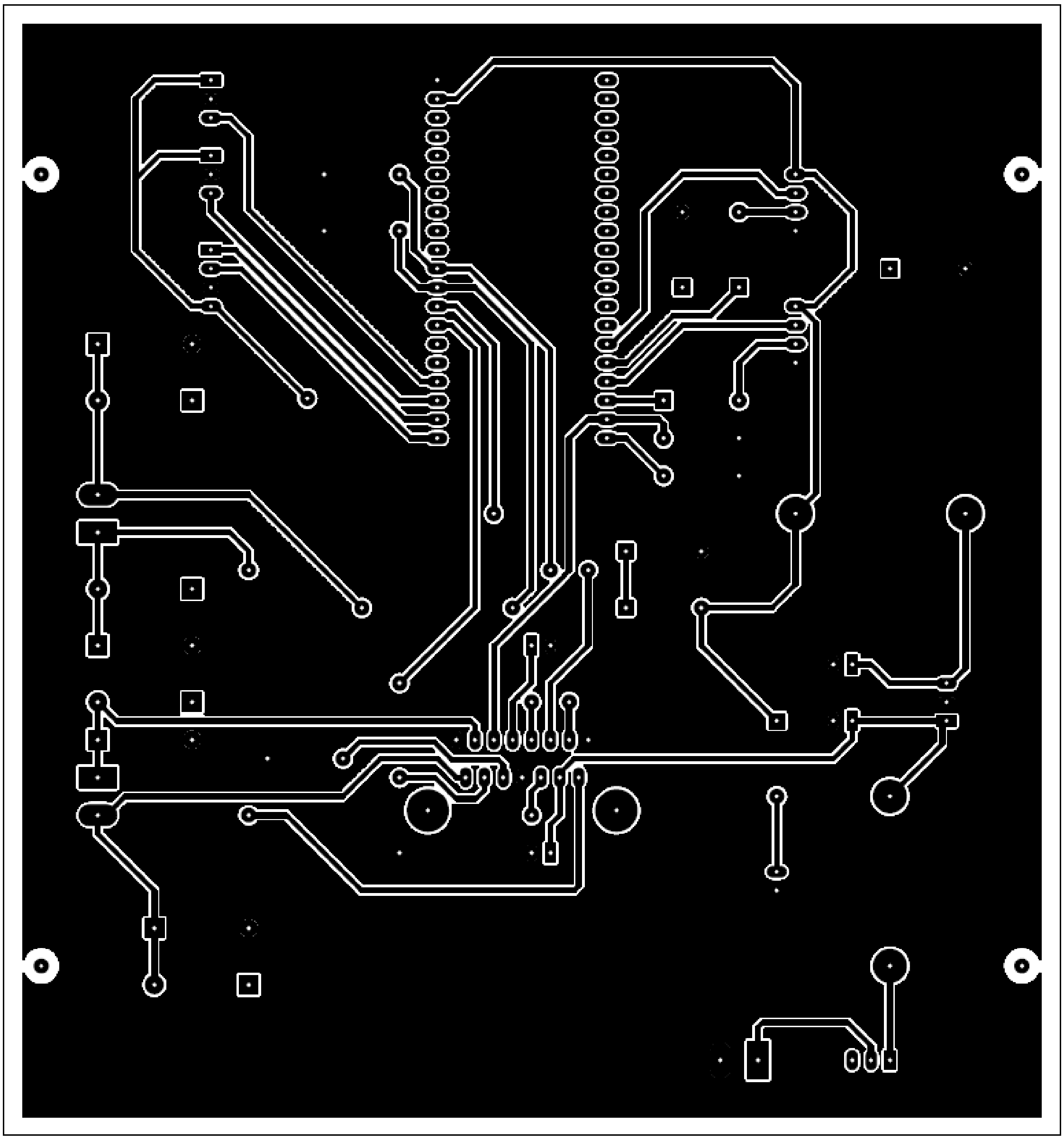
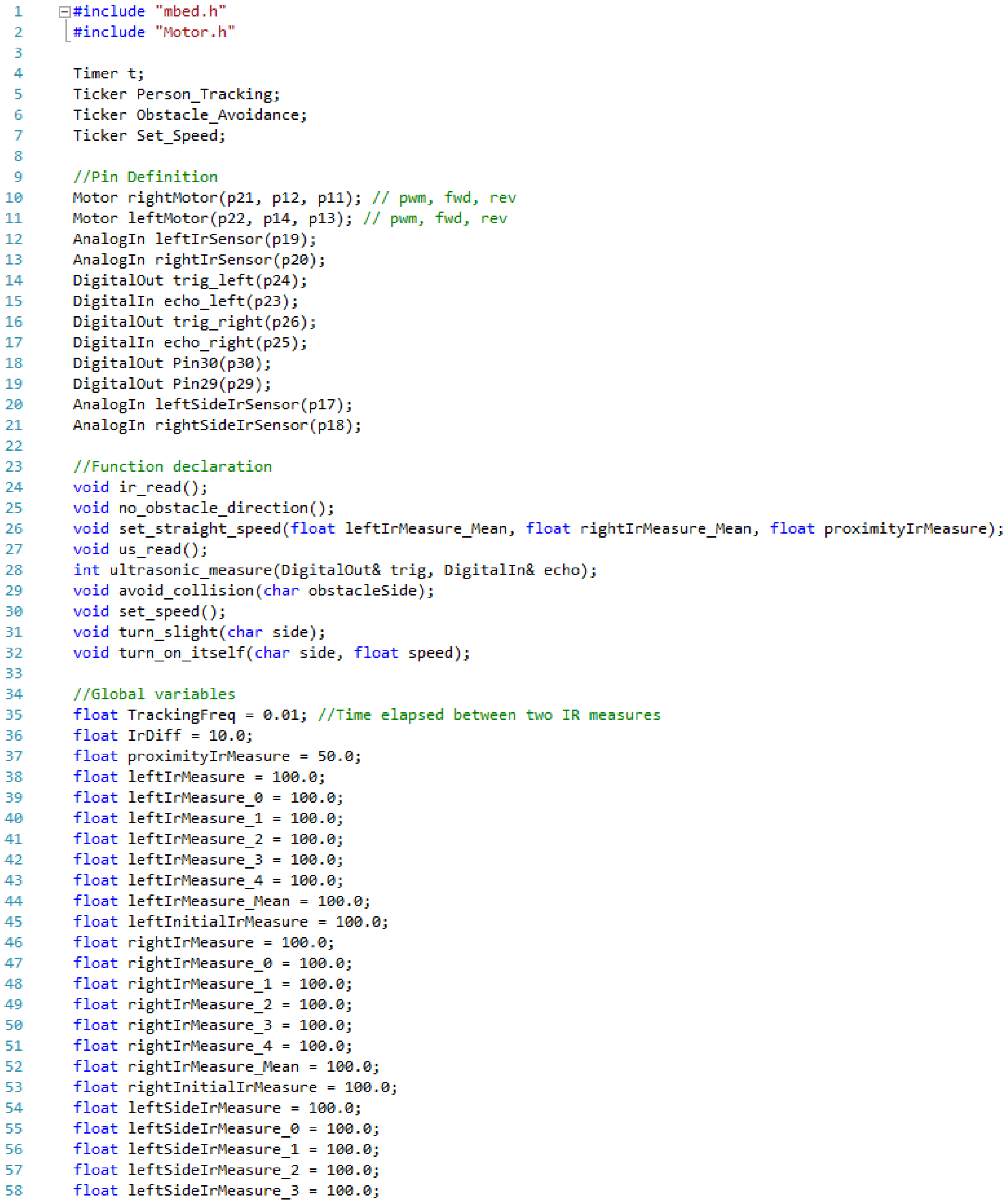
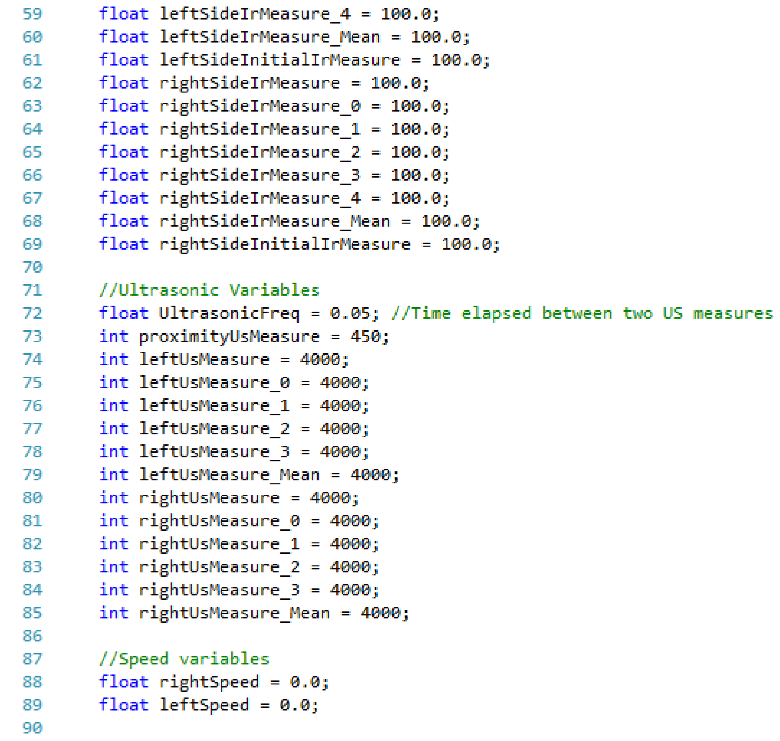


*Figure B.3. PCB bottom layer production file*

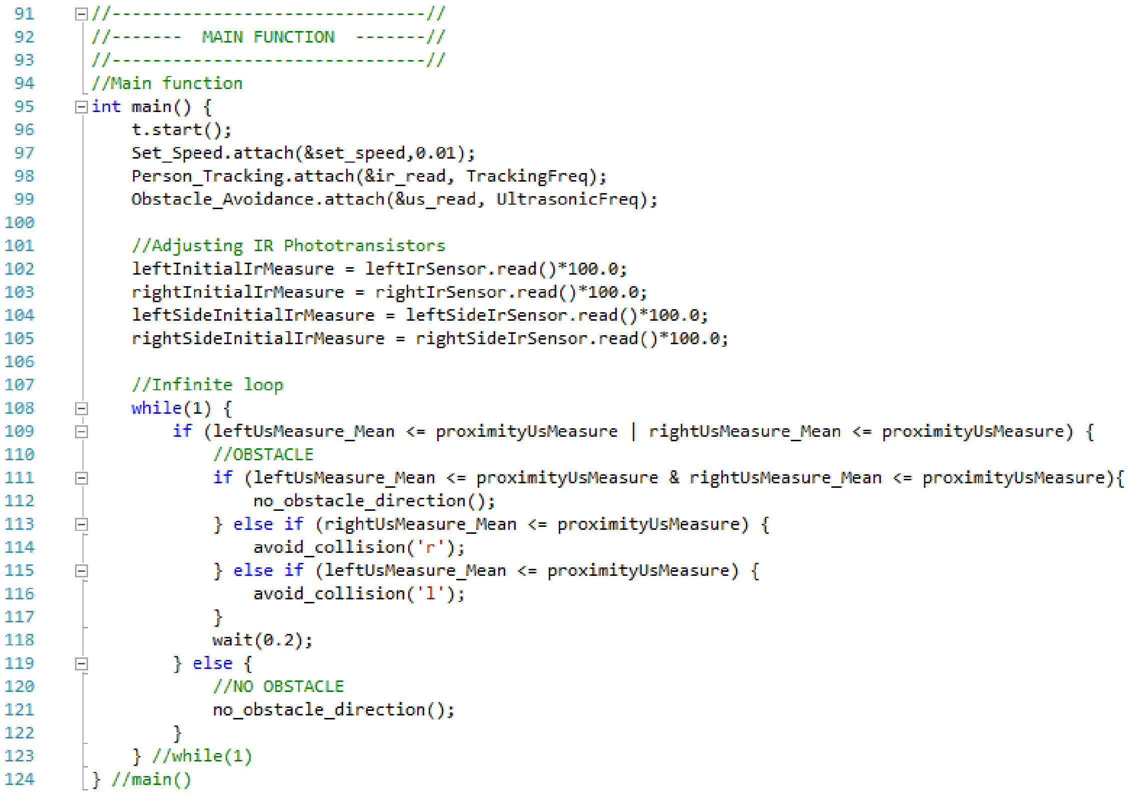
# Appendix C.   Mbed microcontroller code



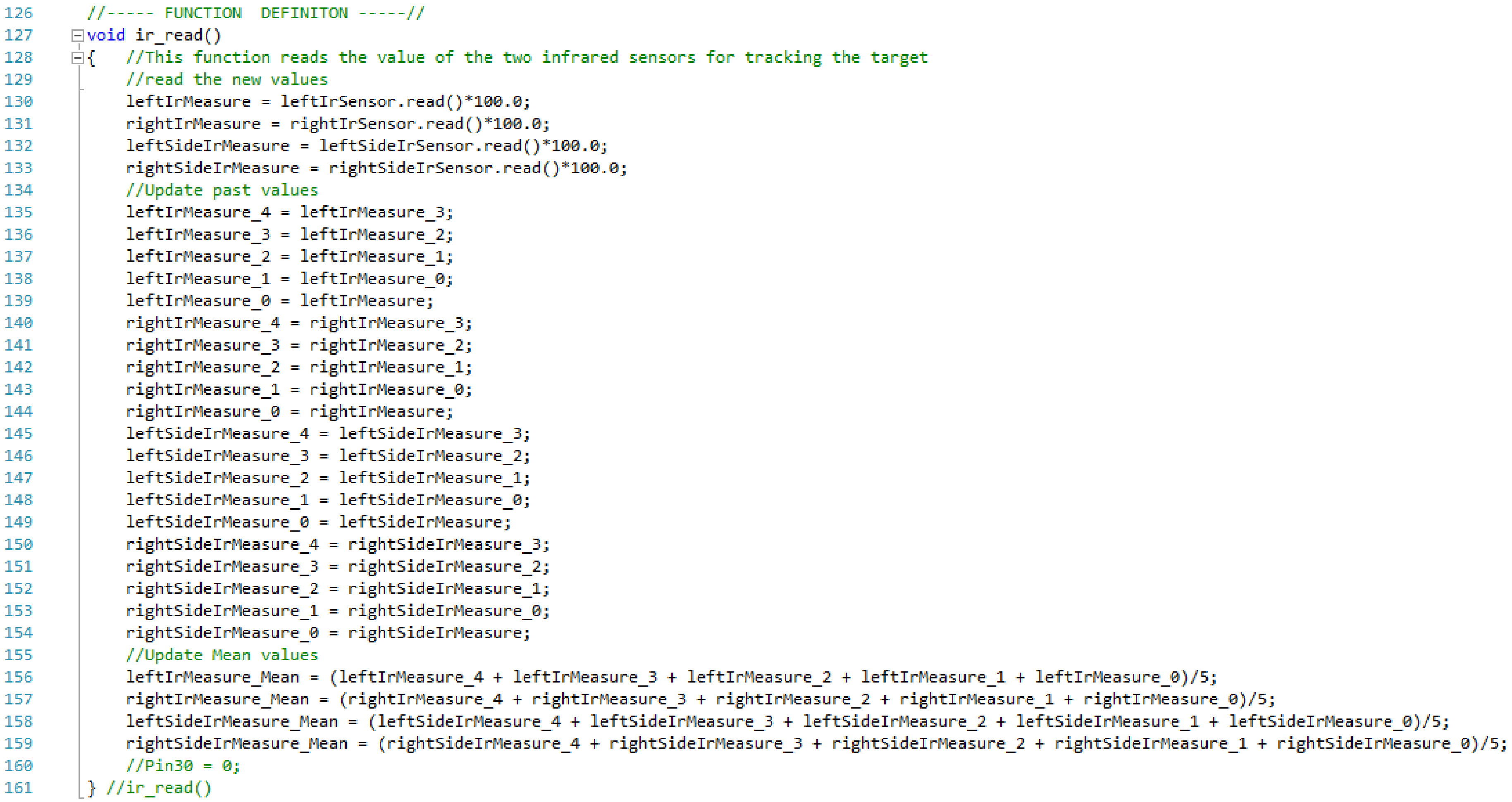
```
#include "mbed.h"
#include "Motor.h"

Timer t;
Ticker Person_Tracking;
Ticker Obstacle_Avoidance;
Ticker Set_Speed;

//Pin Definition
Motor rightMotor(p21, p12, p11); // pwm, fwd, rev
Motor leftMotor(p22, p14, p13); // pwm, fwd, rev
AnalogIn leftIrSensor(p19);
AnalogIn rightIrSensor(p20);
DigitalOut trig_left(p24);
DigitalIn echo_left(p23);
DigitalOut trig_right(p26);
DigitalIn echo_right(p25);
DigitalOut Pin30(p30);
DigitalOut Pin29(p29);
AnalogIn leftSideIrSensor(p17);
AnalogIn rightSideIrSensor(p18);

//Function declaration
void ir_read();
void no_obstacle_direction();
void set_straight_speed(float leftIrMeasure_Mean, float rightIrMeasure_Mean, float proximityIrMeasure);
void us_read();
int ultrasonic_measure(DigitalOut& trig, DigitalIn& echo);
void avoid_collision(char obstacleSide);
void set_speed();
void turn_slight(char side);
void turn_on_itself(char side, float speed);

//Global variables
float TrackingFreq = 0.01; //Time elapsed between two IR measures
float IrDiff = 10.0;
float proximityIrMeasure = 50.0;
float leftIrMeasure = 100.0;
float leftIrMeasure_0 = 100.0;
float leftIrMeasure_1 = 100.0;
float leftIrMeasure_2 = 100.0;
float leftIrMeasure_3 = 100.0;
float leftIrMeasure_4 = 100.0;
float leftIrMeasure_Mean = 100.0;
float leftInitialIrMeasure = 100.0;
float rightIrMeasure = 100.0;
float rightIrMeasure_0 = 100.0;
float rightIrMeasure_1 = 100.0;
float rightIrMeasure_2 = 100.0;
float rightIrMeasure_3 = 100.0;
float rightIrMeasure_4 = 100.0;
float rightIrMeasure_Mean = 100.0;
float rightInitialIrMeasure = 100.0;
float leftSideIrMeasure = 100.0;
float leftSideIrMeasure_0 = 100.0;
float leftSideIrMeasure_1 = 100.0;
float leftSideIrMeasure_2 = 100.0;
float leftSideIrMeasure_3 = 100.0;
```



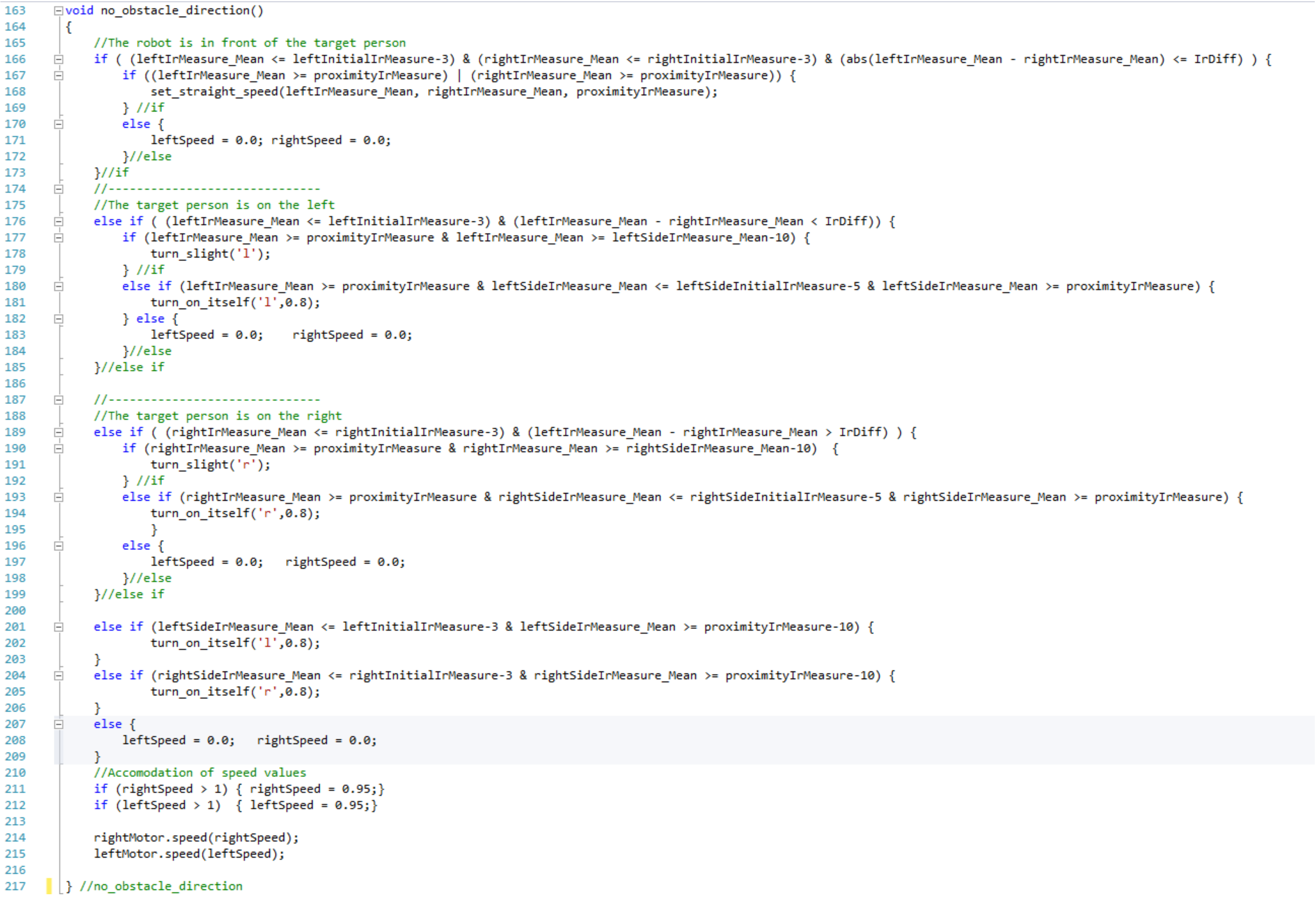
```
float leftSideIrMeasure_4 = 100.0;
float leftSideIrMeasure_Mean = 100.0;
float leftSideInitialIrMeasure = 100.0;
float rightSideIrMeasure = 100.0;
float rightSideIrMeasure_0 = 100.0;
float rightSideIrMeasure_1 = 100.0;
float rightSideIrMeasure_2 = 100.0;
float rightSideIrMeasure_3 = 100.0;
float rightSideIrMeasure_4 = 100.0;
float rightSideIrMeasure_Mean = 100.0;
float rightSideInitialIrMeasure = 100.0;

//Ultrasonic Variables
float UltrasonicFreq = 0.05; //Time elapsed between two US measures
int proximityUsMeasure = 450;
int leftUsMeasure = 4000;
int leftUsMeasure_0 = 4000;
int leftUsMeasure_1 = 4000;
int leftUsMeasure_2 = 4000;
int leftUsMeasure_3 = 4000;
int leftUsMeasure_Mean = 4000;
int rightUsMeasure = 4000;
int rightUsMeasure_0 = 4000;
int rightUsMeasure_1 = 4000;
int rightUsMeasure_2 = 4000;
int rightUsMeasure_3 = 4000;
int rightUsMeasure_Mean = 4000;

//Speed variables
float rightSpeed = 0.0;
float leftSpeed = 0.0;

```



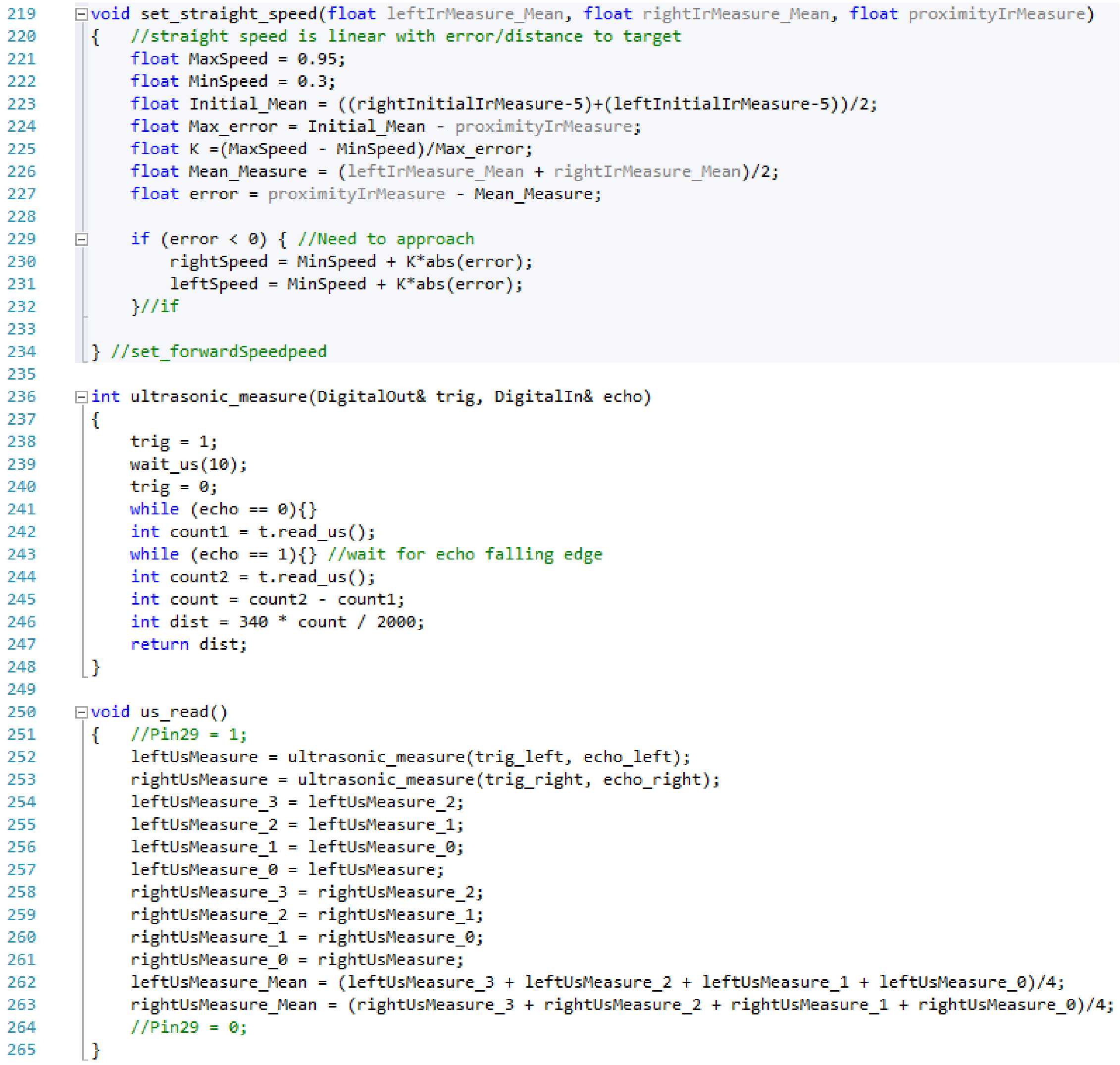
```
//------------------------------//
//-------  MAIN FUNCTION  -------//
//------------------------------//
//Main function
int main() {
    t.start();
    Set_Speed.attach(&set_speed,0.01);
    Person_Tracking.attach(&ir_read, TrackingFreq);
    Obstacle_Avoidance.attach(&us_read, UltrasonicFreq);

    //Adjusting IR Phototransistors
    leftInitialIrMeasure = leftIrSensor.read()*100.0;
    rightInitialIrMeasure = rightIrSensor.read()*100.0;
    leftSideInitialIrMeasure = leftSideIrSensor.read()*100.0;
    rightSideInitialIrMeasure = rightSideIrSensor.read()*100.0;

    //Infinite loop
    while(1) {
        if (leftUsMeasure_Mean <= proximityUsMeasure | rightUsMeasure_Mean <= proximityUsMeasure) {
            //OBSTACLE
            if (leftUsMeasure_Mean <= proximityUsMeasure & rightUsMeasure_Mean <= proximityUsMeasure){
                no_obstacle_direction();
            } else if (rightUsMeasure_Mean <= proximityUsMeasure) {
                avoid_collision('r');
            } else if (leftUsMeasure_Mean <= proximityUsMeasure) {
                avoid_collision('l');
            }
            wait(0.2);
        } else {
            //NO OBSTACLE
            no_obstacle_direction();
        }
    } //while(1)
} //main()
```



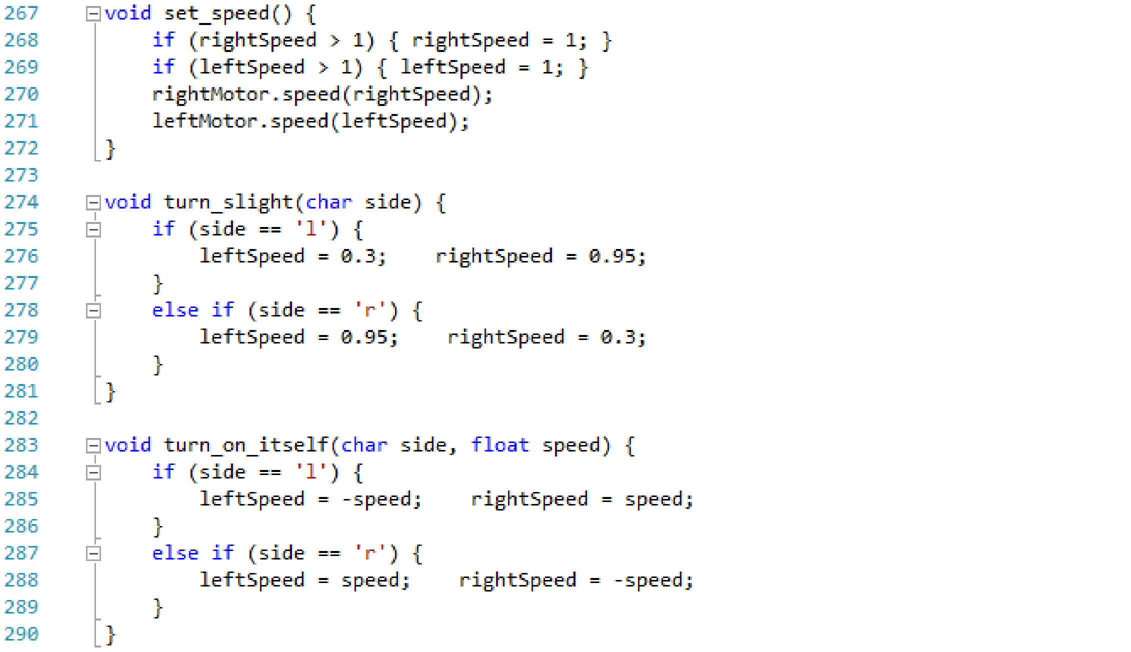
```
//----- FUNCTION  DEFINITON -----//
void ir_read()
{   //This function reads the value of the two infrared sensors for tracking the target
    //read the new values
    leftIrMeasure = leftIrSensor.read()*100.0;
    rightIrMeasure = rightIrSensor.read()*100.0;
    leftSideIrMeasure = leftSideIrSensor.read()*100.0;
    rightSideIrMeasure = rightSideIrSensor.read()*100.0;
    //Update past values
    leftIrMeasure_4 = leftIrMeasure_3;
    leftIrMeasure_3 = leftIrMeasure_2;
    leftIrMeasure_2 = leftIrMeasure_1;
    leftIrMeasure_1 = leftIrMeasure_0;
    leftIrMeasure_0 = leftIrMeasure;
    rightIrMeasure_4 = rightIrMeasure_3;
    rightIrMeasure_3 = rightIrMeasure_2;
    rightIrMeasure_2 = rightIrMeasure_1;
    rightIrMeasure_1 = rightIrMeasure_0;
    rightIrMeasure_0 = rightIrMeasure;
    leftSideIrMeasure_4 = leftSideIrMeasure_3;
    leftSideIrMeasure_3 = leftSideIrMeasure_2;
    leftSideIrMeasure_2 = leftSideIrMeasure_1;
    leftSideIrMeasure_1 = leftSideIrMeasure_0;
    leftSideIrMeasure_0 = leftSideIrMeasure;
    rightSideIrMeasure_4 = rightSideIrMeasure_3;
    rightSideIrMeasure_3 = rightSideIrMeasure_2;
    rightSideIrMeasure_2 = rightSideIrMeasure_1;
    rightSideIrMeasure_1 = rightSideIrMeasure_0;
    rightSideIrMeasure_0 = rightSideIrMeasure;
    //Update Mean values
    leftIrMeasure_Mean = (leftIrMeasure_4 + leftIrMeasure_3 + leftIrMeasure_2 + leftIrMeasure_1 + leftIrMeasure_0)/5;
    rightIrMeasure_Mean = (rightIrMeasure_4 + rightIrMeasure_3 + rightIrMeasure_2 + rightIrMeasure_1 + rightIrMeasure_0)/5;
    leftSideIrMeasure_Mean = (leftSideIrMeasure_4 + leftSideIrMeasure_3 + leftSideIrMeasure_2 + leftSideIrMeasure_1 + leftSideIrMeasure_0)/5;
    rightSideIrMeasure_Mean = (rightSideIrMeasure_4 + rightSideIrMeasure_3 + rightSideIrMeasure_2 + rightSideIrMeasure_1 + rightSideIrMeasure_0)/5;
    //Pin30 = 0;
} //ir_read()
```



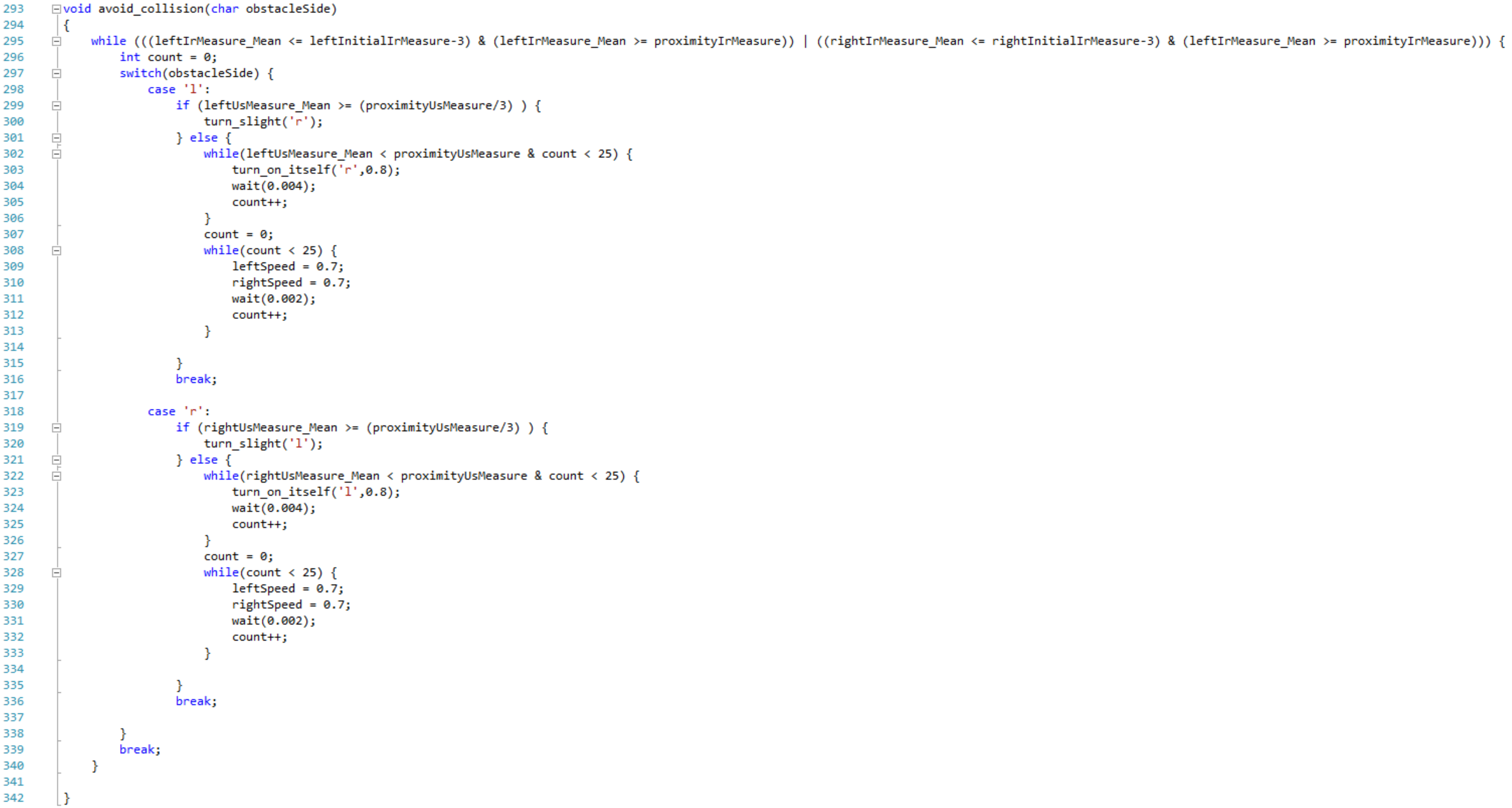
```
void no_obstacle_direction()
{
    //The robot is in front of the target person
    if ( (leftIrMeasure_Mean <= leftInitialIrMeasure-3) & (rightIrMeasure_Mean <= rightInitialIrMeasure-3) & (abs(leftIrMeasure_Mean - rightIrMeasure_Mean) <= IrDiff) ) {
        if ((leftIrMeasure_Mean >= proximityIrMeasure) | (rightIrMeasure_Mean >= proximityIrMeasure)) {
            set_straight_speed(leftIrMeasure_Mean, rightIrMeasure_Mean, proximityIrMeasure);
        } //if
        else {
            leftSpeed = 0.0; rightSpeed = 0.0;
        }//else
    }//if
    //-----------------------------
    //The target person is on the left
    else if ( (leftIrMeasure_Mean <= leftInitialIrMeasure-3) & (leftIrMeasure_Mean - rightIrMeasure_Mean < IrDiff)) {
        if (leftIrMeasure_Mean >= proximityIrMeasure & leftIrMeasure_Mean >= leftSideIrMeasure_Mean-10) {
            turn_slight('l');
        } //if
        else if (leftIrMeasure_Mean >= proximityIrMeasure & leftSideIrMeasure_Mean <= leftSideInitialIrMeasure-5 & leftSideIrMeasure_Mean >= proximityIrMeasure) {
            turn_on_itself('l',0.8);
        } else {
            leftSpeed = 0.0;    rightSpeed = 0.0;
        }//else
    }//else if

    //-----------------------------
    //The target person is on the right
    else if ( (rightIrMeasure_Mean <= rightInitialIrMeasure-3) & (leftIrMeasure_Mean - rightIrMeasure_Mean > IrDiff) ) {
        if (rightIrMeasure_Mean >= proximityIrMeasure & rightIrMeasure_Mean >= rightSideIrMeasure_Mean-10)  {
            turn_slight('r');
        } //if
        else if (rightIrMeasure_Mean >= proximityIrMeasure & rightSideIrMeasure_Mean <= rightSideInitialIrMeasure-5 & rightSideIrMeasure_Mean >= proximityIrMeasure) {
            turn_on_itself('r',0.8);
            }
        else {
            leftSpeed = 0.0;   rightSpeed = 0.0;
        }//else
    }//else if

    else if (leftSideIrMeasure_Mean <= leftInitialIrMeasure-3 & leftSideIrMeasure_Mean >= proximityIrMeasure-10) {
            turn_on_itself('l',0.8);
    }
    else if (rightSideIrMeasure_Mean <= rightInitialIrMeasure-3 & rightSideIrMeasure_Mean >= proximityIrMeasure-10) {
            turn_on_itself('r',0.8);
    }
    else {
        leftSpeed = 0.0;   rightSpeed = 0.0;
    }
    //Accomodation of speed values
    if (rightSpeed > 1) { rightSpeed = 0.95;}
    if (leftSpeed > 1)  { leftSpeed = 0.95;}

    rightMotor.speed(rightSpeed);
    leftMotor.speed(leftSpeed);

} //no_obstacle_direction
```



```
void set_straight_speed(float leftIrMeasure_Mean, float rightIrMeasure_Mean, float proximityIrMeasure)
{   //straight speed is linear with error/distance to target
    float MaxSpeed = 0.95;
    float MinSpeed = 0.3;
    float Initial_Mean = ((rightInitialIrMeasure-5)+(leftInitialIrMeasure-5))/2;
    float Max_error = Initial_Mean - proximityIrMeasure;
    float K =(MaxSpeed - MinSpeed)/Max_error;
    float Mean_Measure = (leftIrMeasure_Mean + rightIrMeasure_Mean)/2;
    float error = proximityIrMeasure - Mean_Measure;

    if (error < 0) { //Need to approach
        rightSpeed = MinSpeed + K*abs(error);
        leftSpeed = MinSpeed + K*abs(error);
    }//if

} //set_forwardSpeedpeed

int ultrasonic_measure(DigitalOut& trig, DigitalIn& echo)
{
    trig = 1;
    wait_us(10);
    trig = 0;
    while (echo == 0){}
    int count1 = t.read_us();
    while (echo == 1){} //wait for echo falling edge
    int count2 = t.read_us();
    int count = count2 - count1;
    int dist = 340 * count / 2000;
    return dist;
}

void us_read()
{   //Pin29 = 1;
    leftUsMeasure = ultrasonic_measure(trig_left, echo_left);
    rightUsMeasure = ultrasonic_measure(trig_right, echo_right);
    leftUsMeasure_3 = leftUsMeasure_2;
    leftUsMeasure_2 = leftUsMeasure_1;
    leftUsMeasure_1 = leftUsMeasure_0;
    leftUsMeasure_0 = leftUsMeasure;
    rightUsMeasure_3 = rightUsMeasure_2;
    rightUsMeasure_2 = rightUsMeasure_1;
    rightUsMeasure_1 = rightUsMeasure_0;
    rightUsMeasure_0 = rightUsMeasure;
    leftUsMeasure_Mean = (leftUsMeasure_3 + leftUsMeasure_2 + leftUsMeasure_1 + leftUsMeasure_0)/4;
    rightUsMeasure_Mean = (rightUsMeasure_3 + rightUsMeasure_2 + rightUsMeasure_1 + rightUsMeasure_0)/4;
    //Pin29 = 0;
}
```



```
void set_speed() {
    if (rightSpeed > 1) { rightSpeed = 1; }
    if (leftSpeed > 1) { leftSpeed = 1; }
    rightMotor.speed(rightSpeed);
    leftMotor.speed(leftSpeed);
}

void turn_slight(char side) {
    if (side == 'l') {
        leftSpeed = 0.3;    rightSpeed = 0.95;
    }
    else if (side == 'r') {
        leftSpeed = 0.95;    rightSpeed = 0.3;
    }
}

void turn_on_itself(char side, float speed) {
    if (side == 'l') {
        leftSpeed = -speed;    rightSpeed = speed;
    }
    else if (side == 'r') {
        leftSpeed = speed;    rightSpeed = -speed;
    }
}
```



```
void avoid_collision(char obstacleSide)
{
    while (((leftIrMeasure_Mean <= leftInitialIrMeasure-3) & (leftIrMeasure_Mean >= proximityIrMeasure)) | ((rightIrMeasure_Mean <= rightInitialIrMeasure-3) & (leftIrMeasure_Mean >= proximityIrMeasure))) {
        int count = 0;
        switch(obstacleSide) {
            case 'l':
                if (leftUsMeasure_Mean >= (proximityUsMeasure/3) ) {
                    turn_slight('r');
                } else {
                    while(leftUsMeasure_Mean < proximityUsMeasure & count < 25) {
                        turn_on_itself('r',0.8);
                        wait(0.004);
                        count++;
                    }
                    count = 0;
                    while(count < 25) {
                        leftSpeed = 0.7;
                        rightSpeed = 0.7;
                        wait(0.002);
                        count++;
                    }

                }
                break;

            case 'r':
                if (rightUsMeasure_Mean >= (proximityUsMeasure/3) ) {
                    turn_slight('l');
                } else {
                    while(rightUsMeasure_Mean < proximityUsMeasure & count < 25) {
                        turn_on_itself('l',0.8);
                        wait(0.004);
                        count++;
                    }
                    count = 0;
                    while(count < 25) {
                        leftSpeed = 0.7;
                        rightSpeed = 0.7;
                        wait(0.002);
                        count++;
                    }

                }
                break;

        }
        break;
    }

}
```

# Appendix D. Bill of Materials

| PROTOTYPE BILL OF MATERIALS | | |
|---|---|---|
| Subassembly title | Part | Qty |
| Structure | Aluminium chassis (150 x 160 x 50 mm) | 1 |
| | Black Nylon Hex Spacer M3 25mm | 40 |
| | 3D Printed sheet (140 x 150 x 1 mm) | 1 |
| | Ø 120mm x 60mm wheel | 4 |
| Electronics | PCB (Appendix B) | 1 |
| | ARM mbed NXP LPC1768 microcontroller | 1 |
| | Zheng K Gearbox Motor ZGA 25 RP 83 12 V | 4 |
| | 3-pin on/off switch | 1 |
| | Kingbright L-7113GD 5mm 2V Green LED | 1 |
| | L298n Dual H-bridge | 1 |
| | ST L7805CV 5 V voltage regulator | 1 |
| | 2.5 A fuse | 1 |
| | 500 mA fuse | 1 |
| | 20 mm fuse holder | 2 |
| | 1N4007 diode | 8 |
| | 1 kΩ resistor | 3 |
| | 1.5 kΩ resistor | 2 |
| | 2 kΩ resistor | 1 |
| | 100 kΩ resistor | 6 |
| | 150 kΩ resistor | 1 |
| | Molex Connector male 3-Pin | 2 |
| | Molex Connector male 4-Pin | 3 |
| | 2 pin 5mm PCB Screw Terminal Connector | 3 |
| | 0.33 µF capacitor | 1 |
| | 0.1 µF capacitor | 3 |
| Power supply | 12V 2800 mAh NiMH battery | 1 |
| | Tenergy 12 V 1.0 A / 2.0 A charger | 1 |
| IR Transmitter | Kingbright L-7113IT 5mm 2V Red LED | 1 |
| | Stanley Electric AN5307B IR LED | 3 |
| | AA Battery | 2 |
| | 3-pin on/off switch | 1 |
| | 20 Ω resistor | 3 |

| | | |
|---|---|---|
| | 50 Ω resistor | 1 |
| IR Receivers | Vishay TEFT4300 IR Phototransistor | 4 |
| | 150 kΩ resistor | 4 |
| | Molex Connector female 3-Pin | 2 |
| | Molex Connector female 4-Pin | 1 |
| US Sensors | HC-SR04 ultrasonic ranging module | 2 |
| | 3D printed support (Appendix A) | 2 |
| | Molex Connector female 4-Pin | 2 |

| REAL-SIZED TROLLEY BILL OF MATERIALS | | |
|---|---|---|
| Subassembly title | Part | Qty |
| Structure | Trolley chassis | 1 |
| | Ø 200mm x 50mm wheel | 4 |
| Electronics | PCB (Appendix B) | 1 |
| | ARM mbed NXP LPC1768 microcontroller | 1 |
| | CROUZET AUTOMATION  80807020 24 V DC Motor | 4 |
| | 3-pin on/off switch | 1 |
| | Kingbright L-7113GD 5mm 2V Green LED | 1 |
| | L298n Dual H-bridge | 1 |
| | ST L7805CV 5 V voltage regulator | 1 |
| | 4 A fuse | 1 |
| | 500 mA fuse | 1 |
| | 20 mm fuse holder | 2 |
| | Vishay BYW54-TR diode | 8 |
| | 1 kΩ resistor | 3 |
| | 1.5 kΩ resistor | 2 |
| | 2 kΩ resistor | 1 |
| | 100 kΩ resistor | 6 |
| | 150 kΩ resistor | 1 |
| | Molex Connector male 3-Pin | 2 |
| | Molex Connector male 4-Pin | 3 |
| | 2 pin 5mm PCB Screw Terminal Connector | 3 |
| | 0.33 μF capacitor | 1 |
| | 0.1 μF capacitor | 3 |
| Power supply | Enix Energies AMH9080 24V 3.2Ah Ni-MH Battery | 1 |

| | | |
|---|---|---|
| IR Transmitter | Kingbright L-7113IT 5mm 2V Red LED | 1 |
| | Stanley Electric AN5307B IR LED | 3 |
| | AA Battery | 2 |
| | 3-pin on/off switch | 1 |
| | 20 Ω resistor | 3 |
| | 50 Ω resistor | 1 |
| IR Receivers | Vishay TEFT4300 IR Phototransistor | 4 |
| | 150 kΩ resistor | 4 |
| | Molex Connector female 3-Pin | 2 |
| | Molex Connector female 4-Pin | 1 |
| US Sensors | HC-SR04 ultrasonic ranging module | 2 |
| | 3D printed support (Appendix A) | 2 |
| | Molex Connector female 4-Pin | 2 |