VIETNAM GENERAL CONFEDERATION OF LABOR

TON DUC THANG UNIVERSITY

FACULTY OF ELECTRICAL & ELECTRONICS ENGINEERING

NGUYEN DUC TRI

DESIGN A SMART HOUSE-SWEEPING ROBOT USING ULTRASONIC SENSORS AND INFRARED SENSOR TO AVOID OBSTACLES.

INDIVIDUAL PROJECT

ELECTRONIC AND TELECOMMUNICATIONS ENGINEERING

HO CHI MINH CITY, SEPTEMBER 2023

VIETNAM GENERAL CONFEDERATION OF LABOR

TON DUC THANG UNIVERSITY

FACULTY OF ELECTRICAL & ELECTRONICS ENGINEERING

NGUYEN DUC TRI - 41900297

DESIGN A SMART HOUSE-SWEEPING ROBOT USING ULTRASONIC SENSORS & INFRARED SENSOR TO AVOID OBSTACLES

INDIVIDUAL PROJECT

ELECTRONIC AND TELECOMMUNICATIONS ENGINEERING

Instructer:

Dr. Tu Lam Thanh

HO CHI MINH CITY, SEPTEMBER 2023

ACKNOWLEDGMENT

First of all, I would like to thank our teacher – Dr. Tu Lam Thanh has been dedicated in helping me as well as creating favorable conditions for me to complete this project. This is a very practical and necessary project for me, this knowledge will be my luggage for my next projects. In the process of implementing the thesis, there are still many mistakes, hope to receive comments from the teacher. Once again, I sincerely thank we!

Ho Chi Minh City, day month ,year 20

Author

(Signature and full name)

Nguyen Duc Tri

The work was completed at Ton Duc Thang University
Scientific instructors: Dr. Tu Lam Thanh

The specialized project was defended at the **Specialized Project Evaluation Council of Ton Duc Thang University** on … /…/20

Confirmation of the Chairman of the Evaluation Council of the Specialized Project and the Dean of the Faculty of Specialized Management after receiving the corrected Specialized Project (if any).

CHAIRMAN	***DEAN OF FACULTY***
(Signature and full name)	*(Signature and full name)*
..……………………….........	..……………………….........

DECLARATION OF AUTHORSHIP

I hereby declare that this is my own specialized project and is under the scientific guidance of **Dr. Tu Lam Thanh**. The research contents and results in this topic are honest and have not been published in any form before. The data in the tables for analysis, comments and evaluation are collected by the author himself from different sources, clearly stated in the reference section.

In addition, in the Specialized Project, a number of comments, assessments as well as data from other authors, other agencies and organizations are used, all with citations and source annotations.

If detect any fraud, I would like to take full responsibility for the content of my Individual Project. Ton Duc Thang University is not related to copyright and copyright violations caused by me during the implementation process (if any).

Ho Chi Minh City, day month ,year 20

Author

(Signature and full name)

Nguyen Duc Tri

DESIGN A SMART HOUSE-SWEEPING ROBOT USING ULTRASONIC SENSORS & INFRARED SENSOR TO AVOID OBSTACLES.

ABSTRACT

Robots are machines that can perform tasks that humans can do, but often with greater speed, accuracy, efficiency, and safety. Robots can also do things that humans cannot do, such as exploring outer space, deep sea, or hazardous environments. Robots have also been deployed to explore the ocean depths, to discover new species and resources, and to monitor the marine ecosystem. Robots can also operate in extreme conditions, such as high temperatures, radiation, or chemical exposure, that would be harmful or fatal for humans.

One of the benefits of robots is that they can increase productivity and quality in various industries. Robots can work 24/7 without getting tired, bored, or distracted. They can also perform precise and consistent operations that require high levels of skill and attention. For example, robots are used in manufacturing microelectronics, cars, airplanes, medical devices, and many other products that we use every day. By using robots, we can produce more goods and services with less waste, error, and risk. Additionally, robots can reduce the cost of labor and materials, and increase the competitiveness and profitability of businesses. Robots can also create new jobs and opportunities for human workers, such as designing, programming, maintaining, and supervising robots.

Another benefit of robots is that they can improve our health and well-being. Robots can assist us in various ways, such as helping us with household chores, personal care, education, entertainment, and companionship. By using robots, we can enhance our quality of life and happiness. For example, robots can help us clean our homes, cook our meals, do our laundry, and take care of our pets. Robots can also help us with our health, such as monitoring our vital signs, reminding us to take our medications, providing physical therapy, and performing surgery. Robots can also help us with our learning, such as tutoring us in different subjects, providing

feedback, and motivating us to achieve our goals. Robots can also help us with our leisure, such as playing games, telling stories, making jokes, and expressing emotions.

As the result of their merits brings to us, I decided to create a "Smart house-sweeping robot" which can automatically clean the floors of our homes without requiring much human intervention and they use sensors and algorithms to navigate around obstacles, avoid stairs, and return to their charging base when needed. This robot is designed to make our lives easier and more comfortable, by taking care of one of the most tedious and time-consuming chores. This robot is also designed to be smart and adaptable, by using ultrasonic sensors and infrared sensors to detect and avoid obstacles, such as furniture, walls, doors, and stairs. The robot can also map out the layout of the room, and plan the optimal path to cover the entire floor area. The robot can also sense the level of dirtiness on the floor, and adjust the suction power accordingly. The robot can also communicate with us, by using voice, lights, or sounds, to indicate its status, such as battery level, cleaning mode, or error messages. The robot can also return to its charging base when its battery is low, or when it has finished cleaning.

Vacuuming is one of those chores that can easily be forgotten or postponed, especially when we are busy or tired. However, leaving the floors dirty can affect our health and comfort. With a vacuum-cleaner robot, we do not have to worry about vacuuming ourselves.

In conclusion, robots have many benefits for our lives as they can increase productivity and quality in various industries, especially the vacuum-robot which saves time and energy by cleaning our floors automatically. Vacuum-cleaner robots are not only useful tools, while they are also fun and inspiring companions. Therefore, I determine the topic, namely "***Design a smart house-sweeping robot using ultrasonic sensors & infrared sensor to avoid obstacles***". As technology progresses, vacuum-cleaner robots will become more intelligent and capable of doing more things for us.

TABLES OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1. INTRODUCTION

1.1. THE REASON FOR CHOSING TOPIC - MOTIVATION

With each passing day, robots play an ever-more-important role in our daily lives. Their purpose is to carry out jobs that people find hazardous or impractical. Robotics has completely changed how we work, live, and communicate with other people.

Manufacturing sectors employ robots in one of their most important applications. Increased productivity and efficiency result from robots' ability to operate nonstop without tiring or making mistakes. They are also capable of doing jobs like welding and small part assembly that call for dexterity and accuracy.

In the medical field, robots are also utilized to help physicians and nurses perform surgery and other treatments. Healthcare workers can focus on more important parts of patient care by having them execute repetitive chores like monitoring vital signs or giving medicine.

Apart from their essential uses, robots are also employed in entertainment settings like theme parks and motion pictures. They demonstrate their abilities while giving people a singular experience.

All in all, robotics has, all things considered, improved safety and ease of living while creating new avenues for creativity and invention. Future applications for robots should be even more intriguing as technology develops. I have selected this topic to build a robot vacuum cleaner in order to achieve these goals.

1.2. THE TARGET IMPLEMENTATION - GOALS

The project involves the entire design and manufacturing process for an autonomous robot vacuum cleaner that can perform two main functions: vacuuming the floor and avoiding obstacles within buildings. The robot vacuum cleaner is intended to be used only in indoor environments, such as homes and offices, and on flat surfaces, such as carpets and tiles. The robot vacuum cleaner has a sleek and compact design that allows it to move easily under furniture and around corners. The robot vacuum cleaner is equipped with various sensors and modules that enable it to detect and avoid obstacles, such as tables, chairs, walls, stairs, and pets. The robot vacuum cleaner also has a powerful suction system that can remove dust, dirt, and debris from the floor. The robot vacuum cleaner is controlled by a central command center that processes the signals from the sensors and modules and operates the motor module that controls the robot's movements. At the end, the robot vacuum cleaner can be programmed to run automatically without any help from the people.

1.3. THE OBJECT AND SCOPE OF THE PROJECT

1.3.1. Subjects of research

My initial idea for this project was to design a robot that can perform two tasks simultaneously: avoiding the obstacles that it encounters on its way and vacuuming and cleaning the dirt that it finds on the floor. Moreover, as it is an automatic robot, I decided to use the Arduino platform to control its functions and movements. The Arduino is a microcontroller board that can be programmed to interact with various sensors and modules that are attached to the robot. The sensors and modules that I used for this robot are as follows: both ultrasonic sensors and infrared sensor can measure the distance between the robot and the obstacles, DC motor that can drive the wheels of the robot, a motor driver that can control the speed and direction of the DC motor, and a vacuum cleaner that can suck the dirt from the floor.

1.3.2. Range of research

The range of research of this project is to investigate and create an automated vacuum cleaner that can perform the task of cleaning indoor spaces, such as rooms, halls, and corridors, while also sensing and avoiding obstacles that might be present in the environment. The automated vacuum cleaner is a robot that can move autonomously on the floor and has a suction system that can collect dust, dirt, and debris from the surface. It is designed to be efficient, reliable, and user-friendly, and to operate with minimal human intervention.

1.4. THE RESEARCH METHODS - METHODOLOGY

The project has addressed several outcomes that are related to the theory, databases, and operating principles of robot manufacturing, based on the previous published initiatives in this field. Some of the outcomes are as follows: the robot manufacturing theory, which covers the concepts, methods, and techniques of designing, building, and testing robots; the databases, which store and manage the information and data of various aspects of robot manufacturing, such as materials, components, specifications, and performance; the operating principles of various electrical components, which explain how the components function and interact with each other in a robot, such as motors, sensors, controllers, and batteries; the many specialized meters, which measure and display the physical quantities and parameters of the robot, such as voltage, current, resistance, temperature, and speed; the programming, which involves writing and executing the code that controls the behavior and actions of the robot; and the simulation software, which allows the user to create and test virtual models of the robot and its environment before the actual implementation.

In this project, I will divide my design process into three main stages of research, which I have conducted from various sources and methods. The first stage is the research that I have obtained from the reports of the experts in the field of robot manufacturing, who have shared their knowledge and experience on how to make a robot and how it works. I have learned many useful things from their reports, such as the design principles, the components, the functions, and the challenges of robot manufacturing.

The second stage is the research that I have done on some theoretical investigation that is used for controlling the robot to avoid the objects that it encounters on its way. I have studied some mathematical models and algorithms that can help the robot to sense and react to the obstacles, such as the ultrasonic sensor, the servo motor, and the Arduino board.

The third stage is the research that I have carried out on the simulation software that people can easily use and imagine how to connect the components together that make the robot active automatic. I have explored some programming and simulation software that can allow me to create and test the code such as the Arduino IDE, Proteus, Fritzing

1.5. THE PRACTICAL SIGNIFICANCE

Robot vacuum cleaners are a sort of robot that can go across the home autonomously, avoid obstacles, and operate without human assistance.

In actuality, logic circuit technology—which was used very early and geared for certain goals—has been used to make a large number of robot vacuum cleaners. However, there are several drawbacks as well, for example the robot's inability to precisely detect obstacles and move to avoid them. In order to address this flaw, I try to create a robotic vacuum cleaner that can avoid obstructions by using signals from infrared and ultrasonic sensors.
The Arduino Uno R3 microcontroller module, which is widely available and simple to use, is used in this project. The vacuum cleaner also makes use of readily accessible materials that are simple to get on the market and to construct and assemble.

CHAPTER 2. OVERVIEW

2.1. A SYNOPSIS OF THE AUTOMATED VACUUM ROBOT

2.1.1. Concept

Have you ever wished for a device that could clean your house without any human intervention? If so, you might be interested in an autonomous vacuum machine, which is a type of robot that can operate on its own by moving around and sucking up dust and dirt. An autonomous vacuum machine can recognize any barriers that might prevent it from cleaning the floor, such as walls, furniture, or wires, and move itself accordingly.

An autonomous vacuum machine consists of several components that work together to perform its function such as Microcontroller, motor control module, motor, sensor, and vacuum cleaner element are all included in an automatic robot vacuum cleaner. The main component is a microcontroller, which is a small computer that controls the behavior and actions of the robot. The microcontroller communicates with a motor control module, which regulates the speed and direction of the motor.

The motor is responsible for driving the wheels and the vacuum cleaner element of the robot. The vacuum cleaner element is the part that creates suction and collects the dust and dirt from the floor. The robot also has a sensor, which is a device that detects the presence and distance of obstacles. The sensor sends signals to the microcontroller, which then decides how to avoid them.

2.1.2. Perform

Imagine a device that can make your life easier by taking care of one of the most tedious household chores: vacuuming. This device is not a fantasy, but a reality. It is called a robot vacuum, and it is designed to clean your home autonomously, without requiring any human input or supervision. A robot vacuum is a smart and convenient gadget that can save you time and energy, and keep your home tidy and hygienic.

How does it work? The robot vacuum is equipped with a set of sensors, including ultrasonic sensors and infrared sensor, that allow it to map out your home and plan the optimal cleaning route. The ultrasonic sensors use sound waves to measure the distance between the robot and the obstacles, while the infrared sensor uses light waves to detect the presence of objects. The robot vacuum can also detect and avoid any objects that might block its path, such as chairs, tables, sofas, or even your pets. The robot vacuum can maneuver around these obstacles and continue its cleaning mission without missing a spot. We can also adjust its suction power according to the type of floor it encounters, whether it is carpet, hardwood, or tile. The robot vacuum has a suction system that can collect dust, dirt, and debris from the floor and store them in a dustbin which is inside it body, while that can be easily emptied. The robot vacuum also has a brush system that can sweep the floor and loosen the dirt before sucking it up.

The robot vacuum is controlled by a central command center that processes the signals from the sensors and operates the motor module that controls the robot's movements. The robot vacuum is powered by a rechargeable battery that can last for several hours. The robot vacuum also has a charging base that it can return to automatically when the battery is low. The robot vacuum can charge itself and resume the cleaning task when the battery is full.

2.1.3. Divide into categories

First of all, we will classify robot vacuum cleaners based on two criteria, which are robots that operate based on sensors and the other is robots that operate based on moving motors. These two criteria that are not only important for determining how the robot vacuum cleaners can navigate and clean the indoor spaces efficiently and effectively, but also help us to separate their types which are better.

On the one hand, if we sort of robot vacuum cleaners based on sensors, we will have the following types of sensors commonly used in vacuum cleaner robots today such as: Infrared Sensor (IR Sensor), Ultrasonic Sensor, Laser Sensor, to name but a few. These sensors are devices that can detect and measure the physical properties of the environment, such as light, sound, and distance, and convert them into electrical signals that can be processed by the robot's control system. As a robot using infrared sensors to avoid objects, they primarily use infrared wavelengths to identify impediments. Infrared sensors work by emitting and receiving infrared light, which is invisible to the human eye, and measuring the reflection or absorption of the light by the objects. Then commencing with the robot uses ultrasonic sensors, it will use the ultrasonic waves to measure the safe lengths which is not making them collide the objects. Ultrasonic sensors work by emitting and receiving ultrasonic sound waves, which are too high-pitched for the human ear, and measuring the time it takes for the sound waves to bounce back from the objects. Following to the next, we have the laser sensors, the vacuum cleaner robots will primarily use lasers to identify obstructions, laser sensors work by emitting and receiving laser beams, which are coherent and monochromatic light waves, and measuring the reflection or scattering of the light by the objects. It can be said that depends on the purpose of the using land, what kinds of sensor we will design for them, while people can make a robot which include all of these sensors to make sure the robot can detect obstacles in various settings. For example, if the robot is used in a dark or dusty environment, the infrared sensor might not work well, so the ultrasonic sensor or the laser sensor might be more suitable.

On the other hand, simplify robot vacuum cleaners based on their movements, there are many types of robots designed to walk on many different types of terrain such as rough terrain, moving up and down stairs,… However we will see that designing a robot that operates on flat surfaces is the easiest.

2.1.4. Product description

These days, there are various types of shapes of the vacuum cleaner robot and the most popular shapes are the D-shape and circle-shape. These shapes have their own benefit and drawback, but the circle-shape is well-liked as it helps the movement of the robot become easy as well as avoiding obstacles is also more effective when the robot can rotate on its own to find a way out. However, compared with the D-shape, the circle one has a smoothly observable drawback that because the brushes are arranged around and due to the robot's structure, the dust collector has been moved to the middle of the body of the machine, leading to the robot not being able to vacuum some areas.

As a general rule, the robot usually use the laser sensor, collision sensor, infrared sensor, ultrasonic sensor,... and in my model, I use 1 infrared sensor to limit the robot falling down the stairs outside the edge of the house, accompanied by 3 ultrasonic sensors are spaced 60 degrees apart to avoid obstacles in front and on both sides of the robot. Moreover, similar to the other robots, my design will have 2 wheel motors which are used to move the robot.

The 2 auxiliary brushes are arranged on both sides in front of the 2 wheels and close to the main brush in the middle to help the robot collect dust in places where the robot cannot enter. Thereby helping the robot overcome shortcomings and make cleaning cleaner.

Getting closer to the inside of the robot, as the processing block is the Arduino Uno R3 chip Ch340G so I decide to locate near the edge of the robot to facilitate loading and editing code. Then we have 2 switch which controls the speed of the vacuum cleaner that it located on the top of the robot and the other is controlled the speed of the brushes is placed at the end of the robot, next to the source switch.

2.2. FUNDAMENTAL PRINCIPLE OF OPERATION

2.2.1. Functional State

In order for the robots to operate in an all-around smooth manner, I have divided their operating phases into 3 essential parts with the first stage representing stability. At this stable stage, the robot does not encounter any obstacles (meaning the distance from the robot to the obstacles is above a safe distance), then the sensors will act as sending a signal is measuring distance. To take specific examples, the ultrasonic sensors will measure the distance in front of and the two side of the robot, while the infrared sensor will signal that does the robot still on the ground, if it is near stairs, it will back until infrared sensor feel safe.

In the next status, I call it a motion status, it means the robot will keep straight forward until the objects is under the safe distance. To be more clearly, it still goes a head and it will change into the next status when it is near the obstacles.

At the last standing, I named it "the obstacle state". In this state, all the sensors are calculated the safe distance which is not only wider, but also safe (as it doesn't near the stairs) the robot will move to that way.

2.2.2. Operational framework

The robot is designed to operate in the standing position for most of the time, which allows it to perform various tasks and functions. In this configuration, the robot constantly communicates with its many devices, including sensors and modules, by sending a signal to each of them and requesting a status update.

The sensors and modules then provide an immediate response to the center, indicating their functionality and any issues they might encounter. The sensor will be affected by any obstructions that are present in the environment, such as tables, chairs, walls, and other objects. The center will receive a signal from the sensor when these variables cross the operating threshold, which means that they interfere with the sensor's performance. The signal processing operations are carried out at the control center, which is the brain of the robot. In order to alter the robot's movements, the command center will simultaneously operate the motor module, which controls the robot's limbs and joints.

CHAPTER 3. DESIGN & CODING

3.1. IMPLEMENTATION

3.1.1. Conceptual Designs

At the beginning, I want to make a vacuum cleaner robot that can avoid obstacles, so the first requirement must be the ability which recognizes obstructions and makes steady, adaptable dodge motions. This ability is essential for the robot to function properly and safely in different environments, such as living rooms, bedrooms, kitchens, and bathrooms. The robot should be able to detect various types of obstacles, such as furniture, walls, doors, stairs, pets, and humans, and avoid them by changing its direction, speed, or angle. The robot should also be able to adapt to dynamic situations, such as moving objects, changing lighting, or unexpected events, and react accordingly.

Besides, making them as convenient and useful as possible, I suppose robot vacuum cleaners should be designed as compactly as possible. In addition to, as the smallest possible size can be that we can easily bring it to everywhere or the robot can easily get in to clean every corner, also the tight spaces. A compact design has many advantages for the robot vacuum cleaner, such as saving space, reducing weight, increasing mobility, and enhancing performance. A compact robot vacuum cleaner can be stored and transported easily, without taking up much room or causing inconvenience. A compact robot vacuum cleaner can also move and maneuver more freely and smoothly, without getting stuck or trapped. A compact robot vacuum cleaner can also reach and clean more areas, such as under the sofa, behind the cabinet, or along the edge of the wall, where dust and dirt tend to accumulate.

Furthermore, I also pay attention to the fact that as an autonomous robot, the robot vacuum cleaner should be able to work independently and intelligently, without requiring much human intervention or supervision. However, the robot vacuum cleaner may also encounter some interference or disturbance from other external devices, such as wireless signals, electrical appliances, or magnetic fields, which may affect its performance or functionality. Therefore, we should also consider some measures to help the robot avoid being affected by these factors, such as shielding, filtering, or isolating the robot from the sources of interference. Moreover, the robot vacuum cleaner should also have a reliable and durable power source, which can provide enough energy for the robot to complete its tasks. A battery source with a voltage of 12V is a good choice, as it can offer sufficient power and longevity for the robot to operate stably for the necessary period of time before the battery runs out.

At the end, the ease of cleaning is another important aspect of the robot vacuum cleaner, as it affects the user's satisfaction and experience. The robot vacuum cleaner should be designed to make the cleaning process as simple and convenient as possible, for both the robot and the user. The robot should have a dust collector that can store a large amount of dust and dirt, without affecting the suction power or the airflow. In addition to, our design should be convience at the cleaning time.

3.1.2. Block shematic

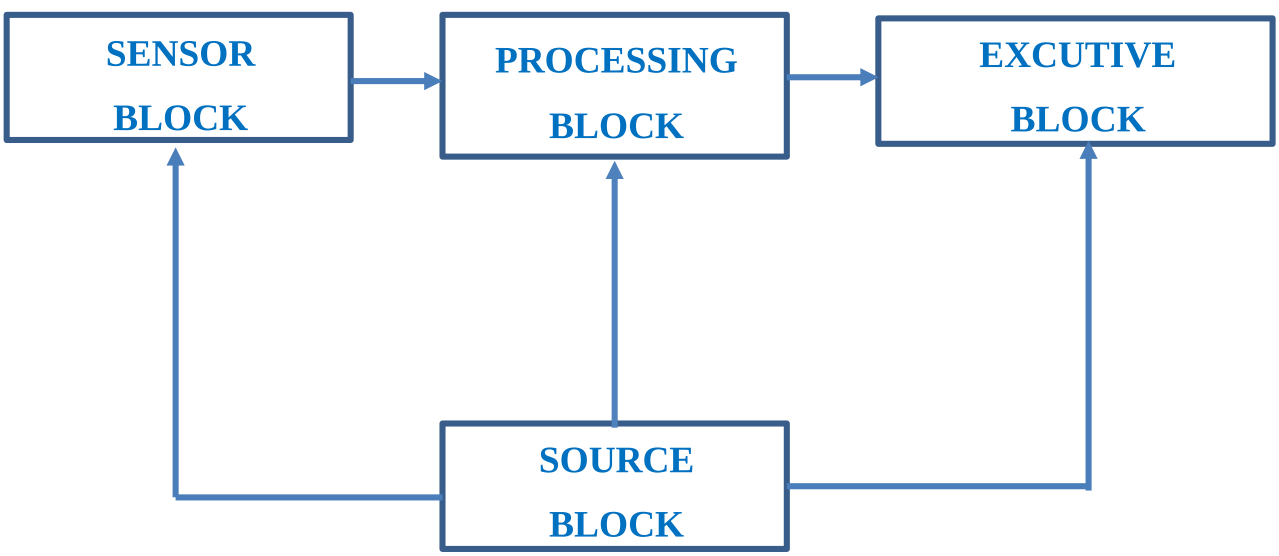

Figure 3.1. Block Diagram.

3.1.2.1. Source Block

MISSION: Supplying electricity to motors and electric circuits using BATTERY 3S LITHIUM 2400mAh.

BENEFITS: compact size, lightweight, powerful discharge current, strong capacity, consistent battery power, extended battery life, etc.

DRAWBACKS: Its modest fire rate and expensive price are offset by the fact that it might explode from overcharging or faulty charging. It also becomes quite hot while it is on fire.

Here is the specification of the Battery 3S Lithium 12V 4Ah 2400mAh:

Table 3.1. Specification of the Battery 3S Lithium 12V 4Ah 2400mAh.

Battery capacity	2400mAh (4Ah)
Minimum capacity	2350mAh
Configuration	18650
Standard voltage	11.1v
Maximum charging voltage	12.6v
Maximum discharge voltage	9v (some need to reduce to 7.5v, upon request)
Battey size	68*55*36
Battery weight	approx. 0.15kg
Overcharge protection voltage	4.2v ± 0.025v
Overload protection voltage	9v (or 7.5v)
Overcurrent protection	5-8A
Working temperature	-20 ~ 55 ℃
Charging temperature	0 ~ 45 ℃
Discharge temperature	-20 ~ 55 ℃
Storage temperature	25℃

Figure 3.2. Battery 3S Lithium 12V 4Ah 2400mAh.

3.1.2.2. Processing Block

MISSION: Process and compare the signals that were received from the sensor block to provide the actuator block with control signals.

Since the Arduino Uno R3 chip CH340G is used in this project, we will have a deeper understanding of its benefits and drawbacks in addition to learning in-depth information about the port pins of the Arduino.

BENEFITS: Widely available, inexpensive, simple to use,...

DRAWBACKS: Because it contains so many built-in features, therefore both learning to code and making circuit will be challenging for novices.

As I use the chip Arduino Uno R3, so I will introduce a little bit about it on the market. Generally, there are a two kinds of Arduino Uno R3 chip, the first is "Arduino Uno R3 chip ATMEGA328P" and the other is "Arduino Uno R3 chip CH340G" which is used in this project.

In essence, both of them have the same structure, but there is just a little bit in their main microcontroller. Therefore, I will show the difference between them below:

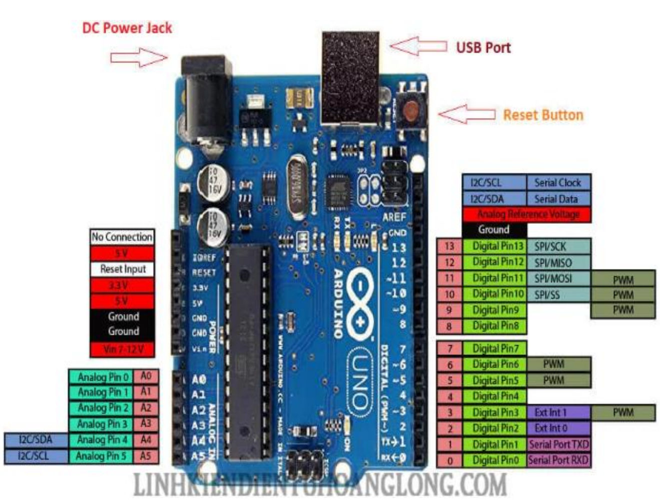

Figure 3.3. Arduino Uno R3 chip ATMEGA328P.

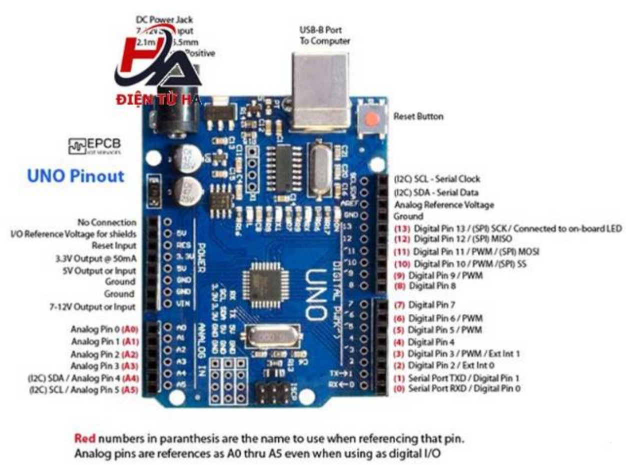

Figure 3.4. Arduino Uno R3 chip CH340G.

You can simply see the opposite between these two models. Next, I would like to show you perspicuously about their specifications.

Table 3.2. The comparison in specifications between two kinds of Arduino Uno board.

	ARDUINO UNO R3 CHIP ATMEGA328P	ARDUINO UNO R3 CHIP CH340G
Microcontroller	ATMEGA328P	ATMEGA328P
IC charging and UART communication	ATMEGA16U2	CH340
Operating voltage	5 V	5 V
Operating frequency	16 MHz	16 MHz
Current Consumption	30 mA	30 mA
Input voltage (recommended)	7-12 V	7-12 V
Input voltage (limit)	6-20 V	6-20 V
Digital I/O pin	14 (With 6 PWM output pins)	14 (With 6 PWM output pins)
Analog input pins	6	6
Maximum current per I/O pin	30 mA	30 mA
Maximum output current (5V)	500 mA	500 mA
Maximum Output Current (3.3V)	50 mA	50 mA
Flash Memory	32KB (ATmega328) with 0.5KB used by Bootloader	32KB (ATmega328) with 0.5KB used by Bootloader
SRAM	2 KB (ATmega328)	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)	1 KB (ATmega328)

Then, we will discuss in detail about the Arduino Uno R3 chip CH340G pins.

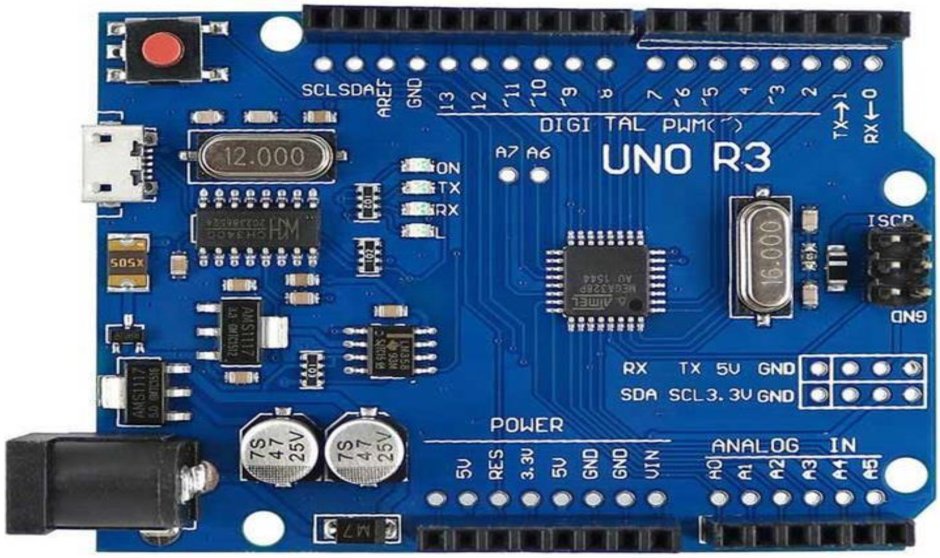

Figure 3.5. Arduino Uno R3 chip CH340G (in detail).

When it comes to electricity, the Arduino UNO R3 may be powered by 5V via the USB connection, or it can be supplied externally at a voltage between 6 and 20 volts, and the voltage between 7-12V DC which is usually suggested. An excessive power supply will harm the Arduino UNO if it over the upper limit barrier. For that reason, I will now introduce in detail the power pins in Arduino Uno R3.

Table 3.3. Arduino's power pins.

GND (Ground) pin	negative pole of the power supply to the Arduino UNO. When you use devices that use separate power sources, these pins must be connected together.
5V pin	5V output voltage level. The maximum allowable current at this pin is 500mA.
3.3V pin	3.3V output voltage level. The maximum allowable current at this pin is 50mA.
Vin (Voltage Input) pin	to supply external power to the Arduino UNO, you connect the positive pole of the source to this pin and the negative pole of the source to the GND pin.
IOREF	the operating voltage of the microcontroller on the Arduino UNO can be measured at this pin.

RESET button	pressing the Reset button on the board to reset the microcontroller is equivalent to the RESET pin connected to GND through a 10KΩ resistor.

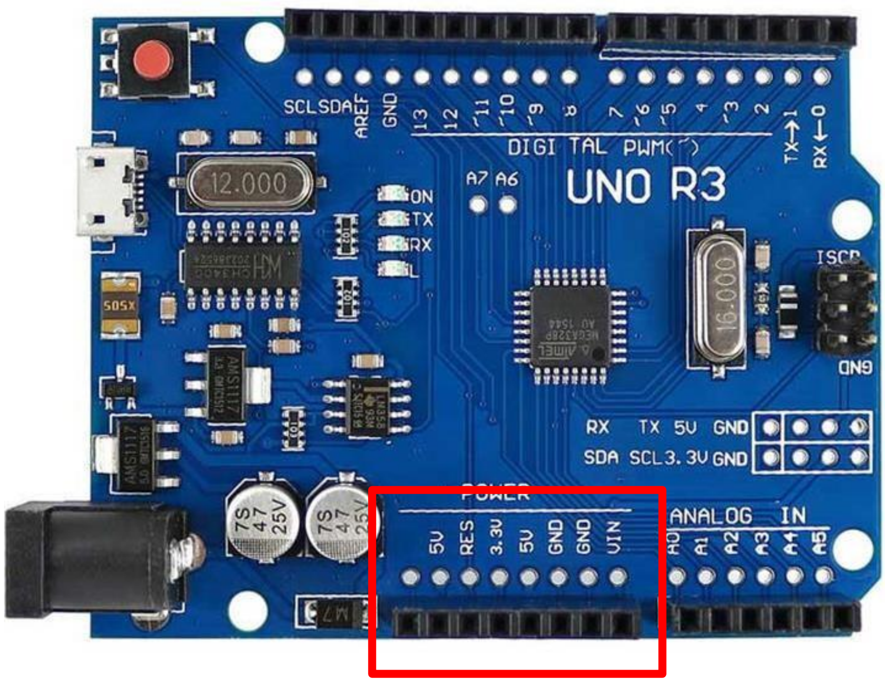

Figure 3.6. Arduino Uno R3's power pins.

Finally, we will research about the memory use and the other remain ports.

Table 3.4. Arduino's memory usage.

32KB Flash memory	the programming instructions will be stored in the microcontroller's Flash memory. Usually a few KB of this will be used for the bootloader.
2KB for SRAM (***S***tatic ***R***andom ***A***ccess ***M***emory)	the values of variables declared when programming will be stored here. The more variables you declare, the more RAM you need. When the power goes out, the data on the SRAM will be lost.
1KB for EEPROM (***E***lectrically ***E***raseble ***P***rogrammable ***R***ead ***O***nly ***M***emory)	A place where data can be read and written to without having to worry about losing power like data on SRAM.

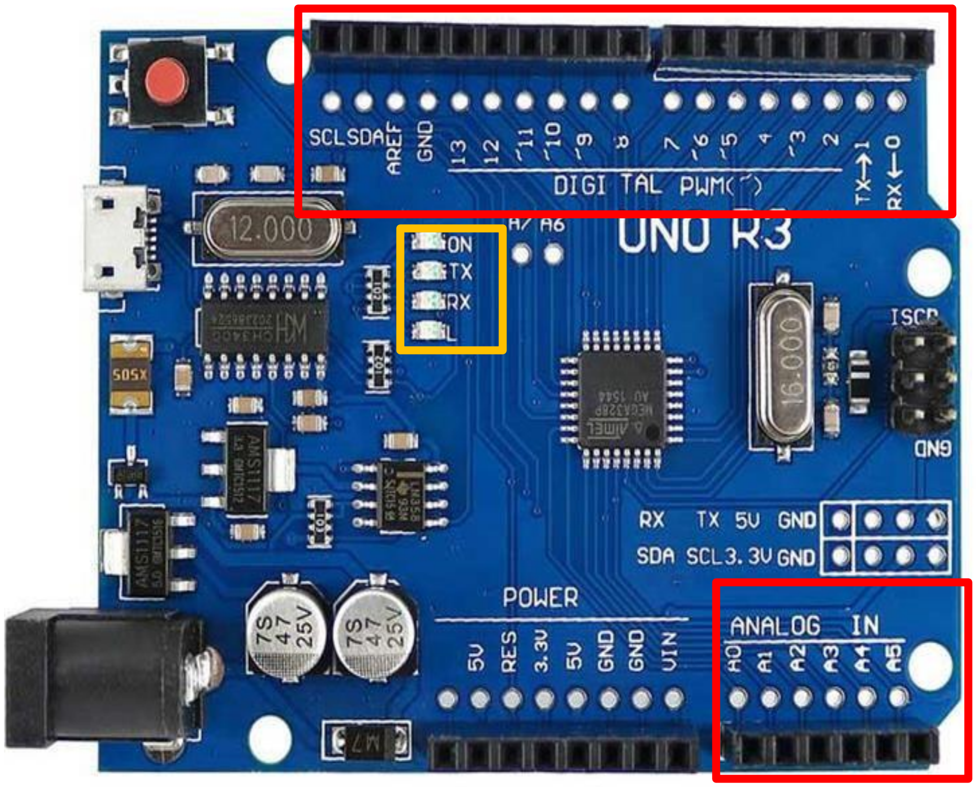

Figure 3.7. Arduino's other ports.

Table 3.5. Arduino's other ports.

2 Serial pins	TTL Serial data is sent (transmit - TX) and received (receive - RX) using the numbers 0 (RX) and 1 (TX) pin port which allow the Arduino Uno to communicate with other devices. A wireless Serial connection is the most prevalent type of Bluetooth connection, while these two pins only use when absolutely necessary
PWM pins (~)	3, 5, 6, 9, 10, and 11: these enable you to use the analogWrite function () to generate PWM pulses with 8bit resolution (values from 0 → 28-1 are equivalent to 0V → 5V). Put simply, unlike other pins that are fixed at 0V and 5V, the voltage that comes out at this pin can be adjusted between 0V and 5V.
SPI communication pins	These four pins :"11 (MOSI), 12 (MISO), 13 (SCK), and 10 (SS)" serve standard purposes as well as data transmission with other devices via the SPI protocol.
LED 13	There is one orange led (signed L) on the Arduino UNO and it will blink to indicate that you have pressed the Reset button, while it is

	attached to digital pin 13. The LED will turn on when the user uses this pin.
6 analog pins (A0 → A5)	Provide voltage values in the range of 0V to 5V with a 10bit signal resolution (0 → 210-1). When utilizing the analog pins, the reference voltage can be input using the AREF pin on the board that means you can measure the voltage in the range of 0V → 2.5V with a resolution of 10bit using the analog pins if you apply a 2.5V voltage to this pin.
To be specific, the *Arduino UNO has 2 pins A4 (SDA) and A5 (SCL)* that support I2C/TWI communication with other devices	

3.1.2.3. Sensor Block

MISSION: Calculate the robot's distance from the obstruction and translate the signal into an electrical signal.

About this section, we will have 2 kinds of sensors which are used in this project including: Infrared Sensor & Ultrasonic Sensor.

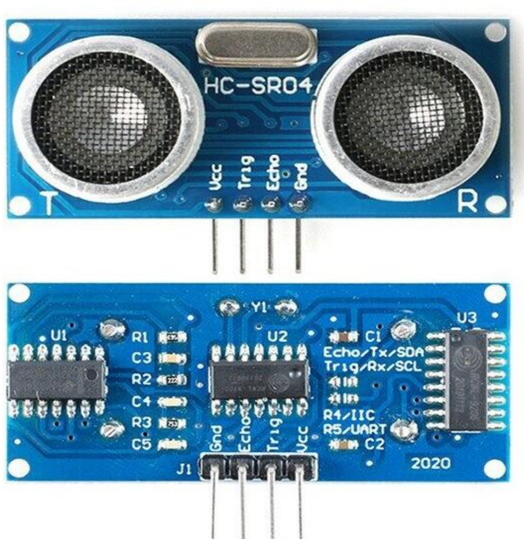

Figure 3.8. HC-SR04 _ Ultrasonic Sensor.

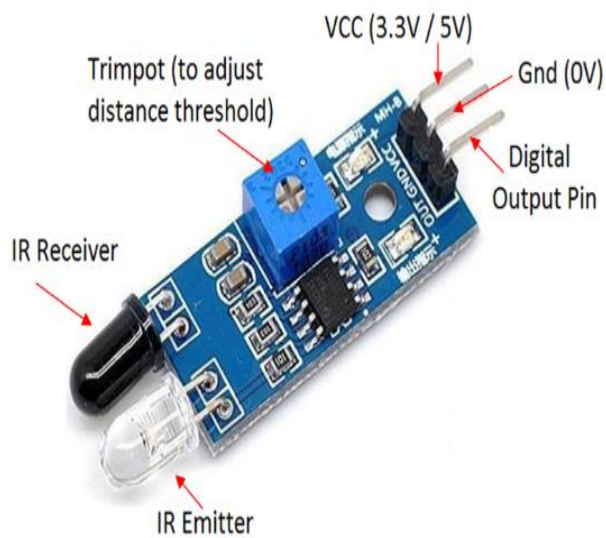

Figure 3.9. Module LM393 _ Infrared Sensor.

Moving to the following part in this section, I want to represent about the pros & cons and also their specifications of the Ultrasonic Sensor.

Table 3.6. Ultrasonic Sensor's ports.

VCC	power to the sensor (5V) or 3.3V at the 3V3.
TRIGGER	Sound wave pin which is the cycle of high/low power taking place.
ECHO	Initial state is 0V, when return signal will be 5V and then return to 0V.
GND	Connect the negative terminal of the circuit.

Table 3.7. Pros&Cons of Ultrasonic Sensor.

Merit	Less interference, determine the distance from the sensor to the obstacle, cheap price, popular in the market.
Demerit	Small scanning angle.

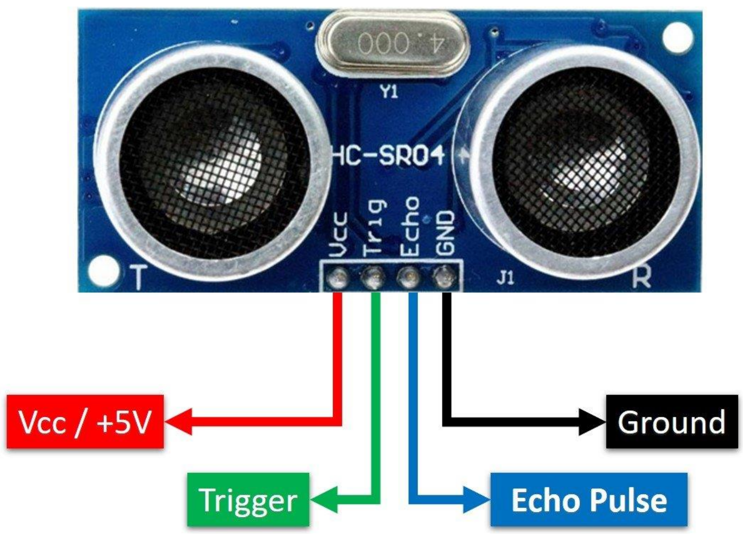

Figure 3.10. Ultrasonic Sensor's ports (in detail).

The distance calculation formula is: $$L = \frac{V * t}{2}$$

Where: L -> Distance.

V -> Speed of sound in air (V = 340m/s).

t -> Time.

With this ultrasonic sensor, we will have the transmitter (trig) on the left and the receiver (echo) on the right, while using this sensor, it allows us to measure the distance up to 3-4m with an effective scanning angle of less than 15 degrees in space through their ports.

Next, I would like to give an account of pros & cons of the Infrared sensor and also its specifications.

Table 3.8. Infrared Sensor's ports.

VCC	voltage conversion from 3.3V to 5V (can be directly connected to 5V and 3.3V microcontrollers).
OUT	digital output (0 & 1).
GND	Connect the negative terminal of the circuit.

Table 3.9. Pros&Cons of Infrared Sensor.

Merit	When detecting infrared radiation from objects in space, the infrared sensor can detect it with high sensitivity. Low cost, ease to structure and design.
Demerit	The ambient temperature has a significant impact on infrared sensors. Many dead angles and restricted scanning angles while excessive sensitivity makes it simple to make mistakes.

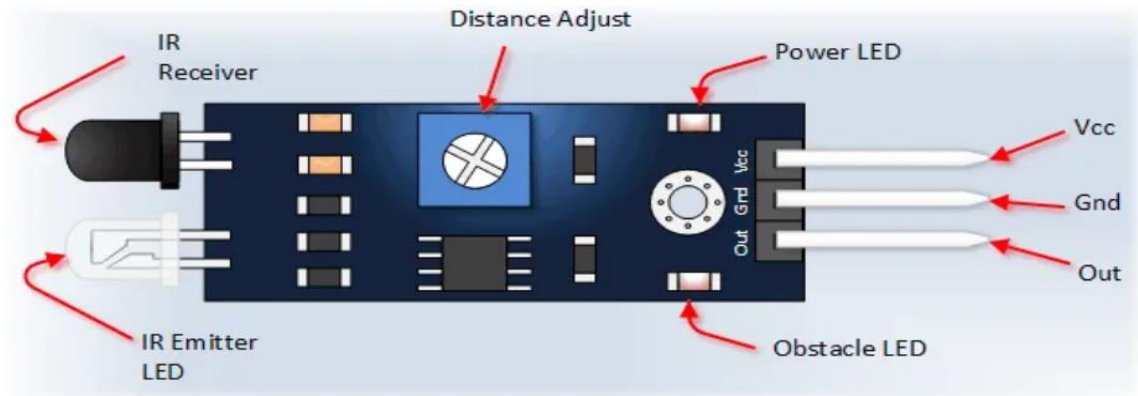

Figure 3.11. Infrared Sensor's ports (in detail).

On a regular basis, any object that can emit heat greater than 5^oK has the ability to emit and reflect infrared wavelengths. Knowing that feature, we will use infrared sensors to install robots so they can avoid obstacles (specifically, it's used to recognize the floor to avoid falling down the stairs) as the Infrared sensor can measure up to 2-3m.

3.1.2.4. Executive Block

MISSION: Assist in controlling the brush and vacuum cleaner's motors and guiding the robot to avoid obstructions.

In this part, particularly in my project, I have "Module L298N" as a circuit board controls motors; "Module DC PWM 5A, 3-35V" to control all the brushes; "Module RC Servo" to control a speed of the "Motor Nidec BLDC" which is known as a vacuum cleaner motor and with the helping of "Module ESC 30A"; "Module DC LM2596 3A" was used to reduce the voltage of the battery to fit with the other components in the circuit.

Firstly, I show the Module L298N and its specifications.

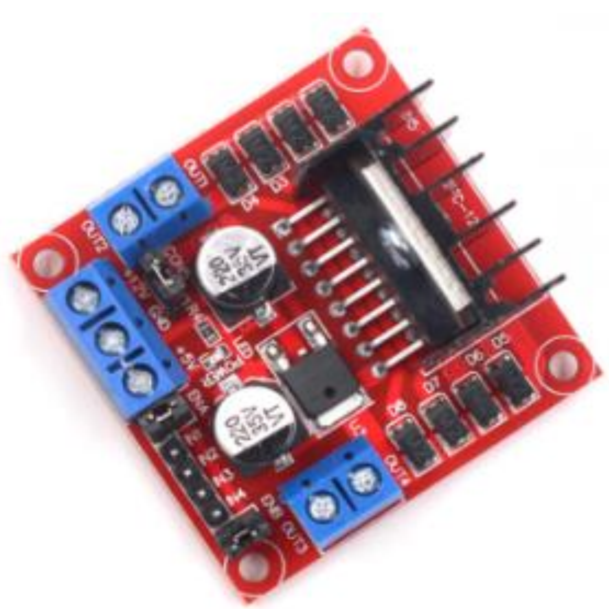

Figure 3.12. Module L298N _ Engine control module L298N | PWM | H-bridge circuit L298.

Table 3.10. Specifications of Module L298N.

Driver	L298N integrated H bridge circuit
Control voltage	5 – 12 V
Maximum current per bridge H	2 A (=> 2A per motor)
Voltage of control signal	5 – 7 V
Current of control signal	0 – 36 mA
Wasted power	20 W (when temperature T = 75^oC)
Storage temperatures	-25^oC – $+130^oC$

Module L298H has built-in 2 H-bridge circuits used to control the motor, as finger out what H-bridge circuit is, I will introduce a little bit about it. The H-bridge circuit is a common and simple circuit used to control DC motors. In general terms, an H-bridge circuit is a circuit consisting of 4 "switches" connected in an H shape which is shown below:

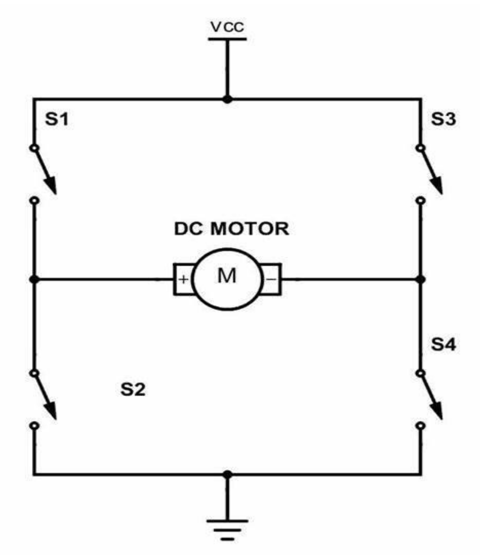

Figure 3.13. H-bridge circuit.

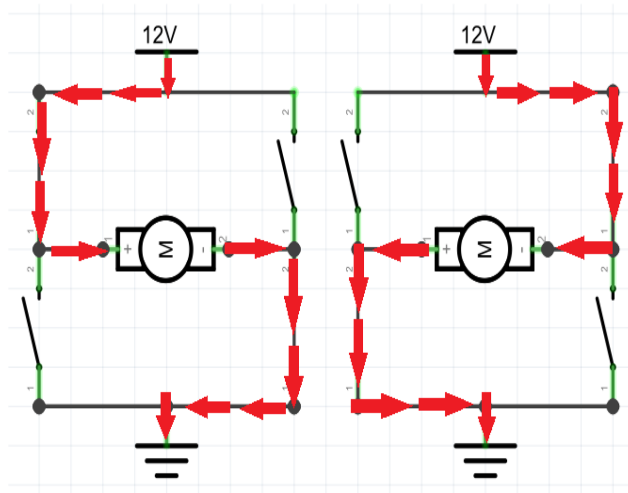

Figure 3.14. Operating principle of H-bridge circuit

Through the operating principle of the H-bridge circuit, we can visualize the mode of operation of the L298N module, through which we can see how the motor pins will operate through the 4 Input pins INA, INB , INC, IND. Here is an overview of the operating states of the two motors connected to the L298N module

- INA = 1, INB = 0: Motor 1 will rotate in the forward direction.
- INA = 0, INB = 1: Motor 1 will rotate in the opposite direction.
- INA = 0, INB = 0: Motor 1 stops/does not operate.
- INC = 1, IND = 0: Motor 2 will rotate in the forward direction.
- INC = 0, IND = 1: Motor 2 will rotate in the opposite direction.
- INC = 0, IND = 0: Motor 2 stops/does not operate.

Then, I show the Module DC PWM 5A, 3-35V and its specifications.

Figure 3.15. Module DC motor speed control circuit PWM 5A, 3-35.

Table 3.11. Specifications of Module DC PWM 5A, 3-35V.

Operating voltage	5 – 35 V
Maximum output power	90 W
Maximum output current	5 A
Line break current	0.015 A
PWM duty cycle	1 – 100 %
PWM frequency	10 kHz

The DCPWM module is based on the principle of the PWM module that we will adjust the rise time or logic high in the PWM signal thereby affecting the duty cycle, and the duty cycle is usually expressed as % in about 0-100%. To be more clarify, changing the knob will change the operating cycle, which will change the output voltage to the motor.

In this section, I will briefly talk about some related concepts to help everyone clearly understand how PWM motors work, including:

- ***Period (of a signal) = Time at high level + Time at low level***
- ***Duty cycle = Time signal is high / Full cycle time***
- Amplitude is the difference between the maximum voltage and minimum voltage of a signal. With the formula: ***Amplitude = Vmax – Vmin***
- Through the above formulas, we will know how to calculate the output voltage formula: ***Vout = Duty cycle * Vmax***

Next, I show the Module RC Servo Motor test circuit and its specification.

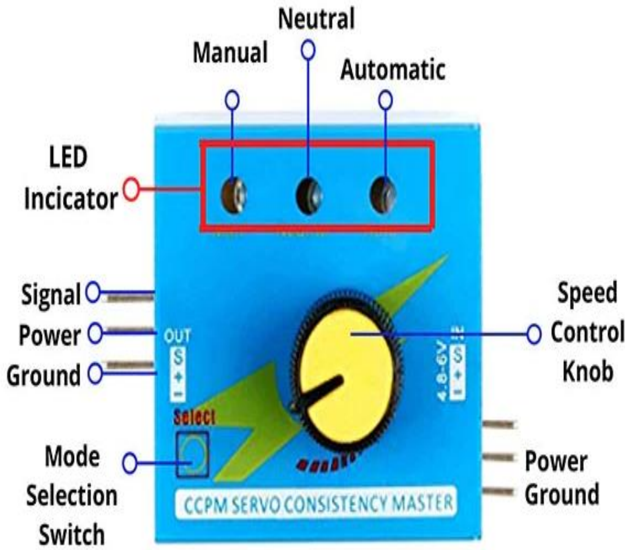

Figure 3.16. RC Servo motor test circuit.

Table 3.12. Specifications of module RC Servo.

Operating voltage	4.5 – 6 V
Control frequency	1.5 ms +/-0.5 ms
Maximum current	15 mA

For RC Servo motor test circuit, we will have components including: Input block: Input to connect power source and input signal; Output block: Output block to connect the motor to be tested or connect to the output signal; Speed change knob: The tester has a knob that manually adjusts the motor speed to help it produce an output signal of 1.5ms ± 0.5ms; Mode selection switch: used to select 3 modes of the tester: Manual, Neutral, Automatic.

With 3 mode selection switch, it is suitable for many different circumstances such as:

- “The automatic mode”: If we choose automatic mode, the servo motor will automatically rotate to its maximum rotation angle as soon as the tester is powered. The rotary motor is similar to a window wiper.
- “The neutral mode”: If we choose neutral mode, the motor will move to the position set by the manufacturer.
- “The manual mode”: If we choose manual mode, we can rotate the servo motor by turning an adjustment knob of the tester to the desired speed.

As you can see the specification of the "RC-Servo Motor Test Circuit" and "Motor driver circuit DC 5A". They are used to control the vacuum motor's power and the sweeping motor's speed. These circuits' advantages are: Accurately and swiftly adjust the motor's speed; it's inexpensive, simple to operate, and the control principle is very simple.

After that, I show the Motor Nidec BLDC and its specification.

Figure 3.17. Motor Nidec vacuum cleaner BLDC.

Table 3.13. Specifications of Motor Nidec vacuum cleaner BLDC.

Speed	62 thousand rpm
Power	360 W

In my model, the reason I chose this Nidec BLDC motor vacuum cleaner is because of its vacuuming ability. If it is supplied with a voltage of 21V, it can reach a maximum capacity of 370W equivalent to the suction force of 20kPa.

However, because most of the components in the circuit have voltages that only fluctuate in the range of 5-12V, the Nidec BLDC motor can still provide us with a huge amount of power with a maximum capacity of 250W when reaching 12V, equivalent to 250W, with a suction force of 14kPa (around 2976.19 rpm), almost equal to today's modern handheld vacuum cleaners.

Subsequently, I show the Module ESC 30A and its specification.

Table 3.14. Specifications of Module ESC 30A.

Voltage	2-3S Lipo 4-12 NIMH
Continuous load current	30 A
Instantaneous load current	40 A
Type	Linear
BEC Output	5V3A

Figure 3.18. 30A BLDC ESC-Brushless Motor Speed Controller with Connector.

The 30A ESC module is a device that can control the speed and direction of a brushless DC (BLDC) motor. It works by receiving a pulse width modulation (PWM) signal from the Arduino board, which is a digital signal that varies in length and frequency.

The 30A ESC module then converts the PWM signal into a 3-phase alternating current (AC) signal, which is a type of electric current that changes direction periodically. The 3-phase AC signal is used to supply power to the BLDC motor, which has three sets of coils that are activated in sequence by the 3-phase AC signal, by doing this, the 30A ESC module can regulate the rotation speed and direction of the BLDC motor.

One important factor that affects the rotation speed of the motor is the length of the PWM pulse, which is also known as the duty cycle. The longer the PWM pulse or the higher the duty cycle, the more power is delivered to the BLDC motor, and the faster the rotation speed will be. Conversely, the shorter the PWM pulse or the lower the duty cycle, the less power is delivered to the BLDC motor, and the slower the rotation speed will be.

Therefore, by adjusting the length of the PWM pulse that is sent from the Arduino to the 30A ESC module, we can easily realize how the rotation speed of the BLDC motor is affected.

In addition, because of advantages such as: "high performance; compact structure; improved efficiency because conventional motors usually only reach 70% while if combined with a 30A ESC module, it can reach 90%; good speed control;..." so that made me choose to use it in my model.

In the end, I show the Module DC LM2596 3A and its specification.

Figure 3.19. Module DC presure reducing circuit LM2596 3A.

Table 3.15. Specifications of DC LM2596 3A.

Input voltage	3 – 30 V
Output voltage	1.5 – 30 V
Maximum response current	3 A
Efficiency	92 %
Power	15 W

In my project, I use a Nidec BLDC motor, which is a type of brushless DC motor that is commonly used in vacuum cleaners. This motor has a high speed and torque, but it also requires a high voltage of 12V from the power source to operate efficiently.

However, the other components in my circuit, such as the Arduino board, the sensors and the motor only need a lower voltage which is around 7V. Therefore, to avoid damaging the circuit or wasting energy, I use a Buck DC-DC LM2596 converter, which is a device that can step down the voltage from a higher level to a lower level.

The Buck DC-DC LM2596 converter works by switching on and off a transistor that controls the current flow through an inductor and a capacitor. By adjusting the duty cycle of the transistor, the converter can regulate the output voltage to the desired level. By using this converter, I can reduce the voltage from the power source to the circuit to match the requirements of the different components in my project.

Besides, there is one more thing that I just want to show and it is “Yellow gear motor” which I use to connect the two brushes and the wheel.

Figure 3.20. Yellow gear motor.

Because the characteristic of a vacuum cleaner robot is that it should move slowly so that the places it passes through can be cleaned of dirt in one go, so I used a yellow gear reduction motor for not only the wheels but also the robot's brushes.

The motor operates with a wide voltage range from 3-9V and the stability is highest at 6-9V, so I used a step-down circuit to adjust the voltage down to 7V from there to maximize rotation speed when there is no power. The load is 292 rpm with 66mm diameter wheels, the travel distance will be about 60m/min.

3.1.3. General model of my vacuum cleaner robot

3.1.3.1 The model

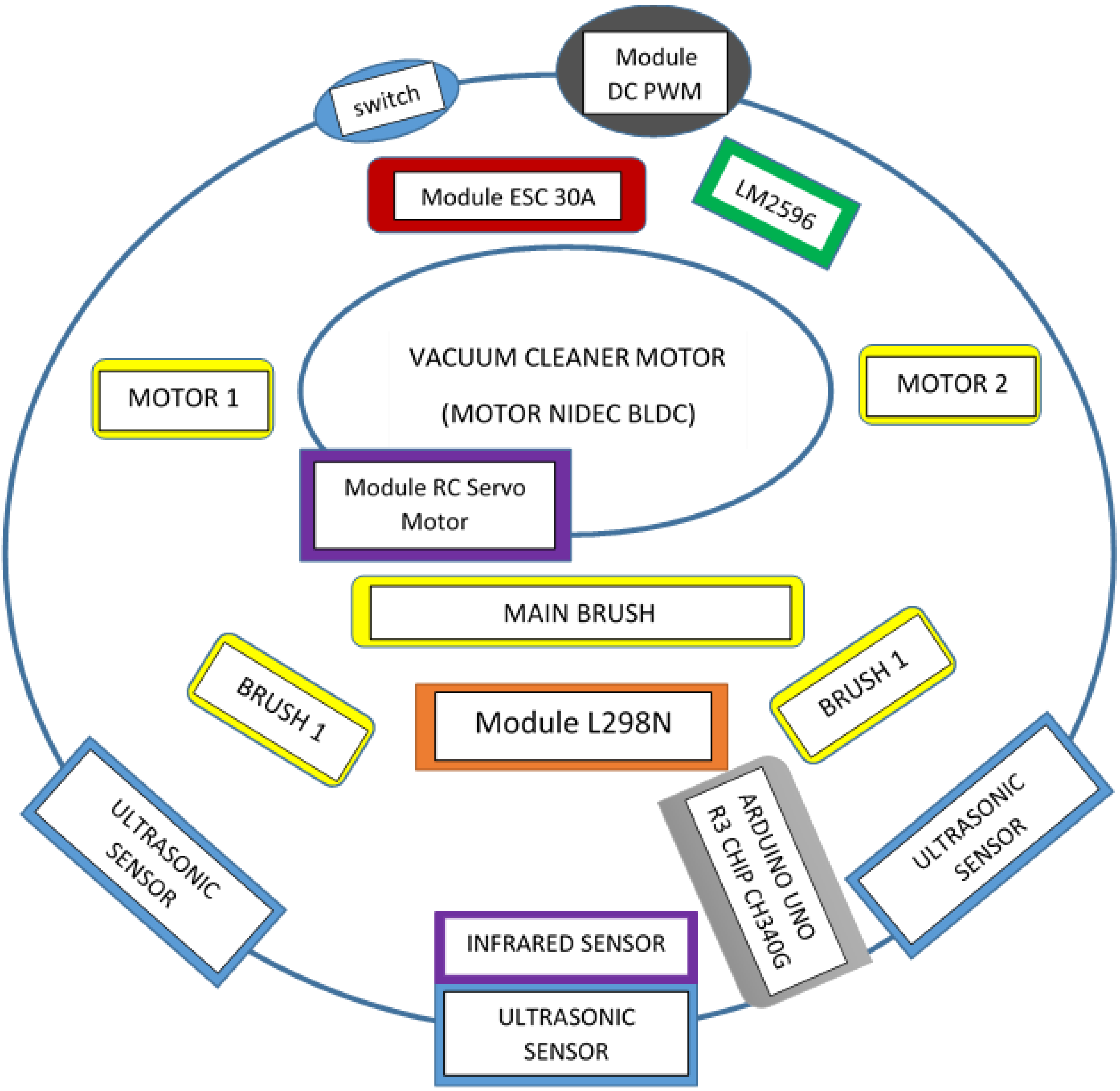

Figure 3.21. Describe the location of each component in the circuit.

3.1.3.2 Operations performed

When I started designing my robot, I had the idea of making it octagonal, because I thought that would make it easier to attach sensors to the eight sides of the robot. However, after doing a lot of research on different shapes and their advantages and disadvantages, I decided to change my design to a circular shape. I learned that a circular shape has many benefits for a robot, such as: better mobility, more stability, more flexibility,... Therefore, based on my research and analysis, I chose a circular shape for my robot, because I believed that it would improve the performance and functionality of my robot.

In terms of product materials, I noticed that Fomex cardboard material is often used in the process of making models and because I wanted to be able to conveniently observe the internal circuit structure, I used flexible mica to Make a border around the robot.

For the first preparation step, I used flexible glue to connect the brush to the yellow gear motor, then I cut the mica and fomex cardboard into the shape I wanted. Next, to make the robot more lively, I used a brush to color the Fomex cover and waited for the color to dry, then I cut and cut holes to facilitate placing components and wiring.

In the next step, I start placing the components as a component location model, and then I estimate the distance of each actual component so that I can prepare the circuit wires. After that, I used solder and wire connections to make the wires more secure, then I went through the step of checking and adjusting the voltage of the Buck DC-DC LM2596 circuit.

Finally, I soldered the wires to some components such as: wheel motor, brush motor, power switch, Buck DC-DC LM2596 circuit, 30A ESC module,... After that, I put the components in place and Use a fixed cord and then plug the components' wires into the Arduino as well as the L298N module,...

3.1.4. An alternate schematic

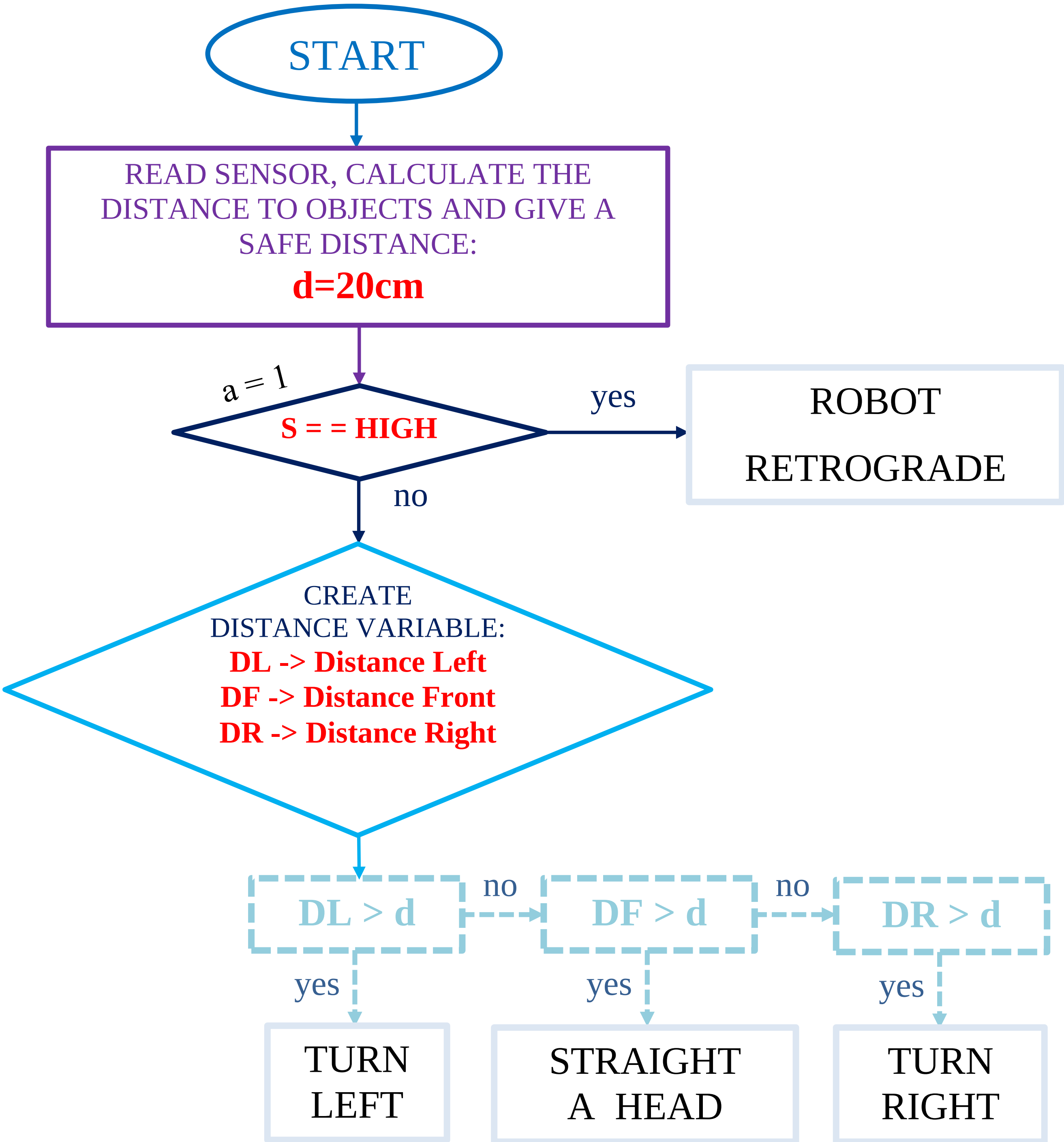

Figure 3.22. Algorithm Diagram.

3.1.5. Schematic of the circuit assembly

As a general rule, before we assemble and design the circuit, we will use simulation tools as a way to model circuit assemblies and see how they work. Simulation tools can help us to verify the functionality and performance of the circuit, as well as to identify and correct any errors or problems.

Therefore, Proteus – the most commonly used simulation tool, which I used to imitate my project. Proteus is a powerful and versatile simulation tool that can support a wide range of circuit components, such as resistors, capacitors, transistors, LEDs, sensors, and so on.

However, there are many enormous simulation that are not only allow you to simulate, but it is also show you how the components look like and the one which I would like to be account to you is Frizting. Frizting is another simulation tool that can create and test circuit assemblies, but it has a unique feature that distinguishes it from other simulation tools. Frizting can help us to design and visualize the circuit in different ways, as well as to export the circuit as an image or a file that can be used for fabrication.

Here is the circuit which is used Proteus to simulation:

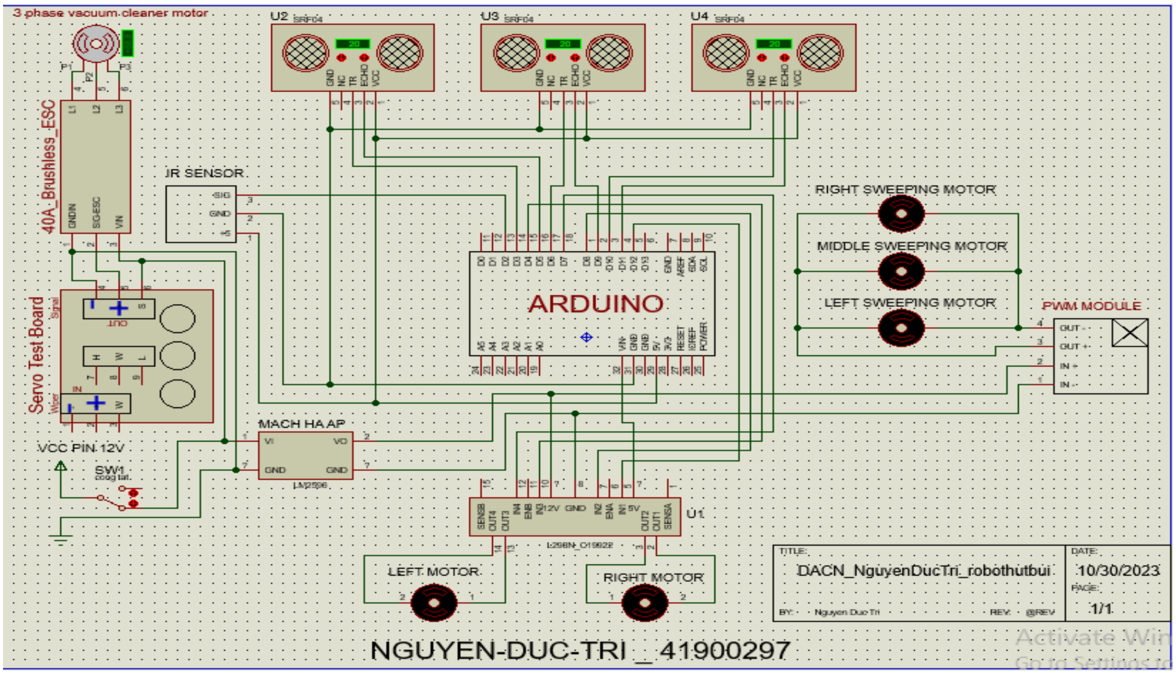

Figure 3.23. Proteus Software is used for simulation.

Here is the circuit which is used Fritzing to simulation:

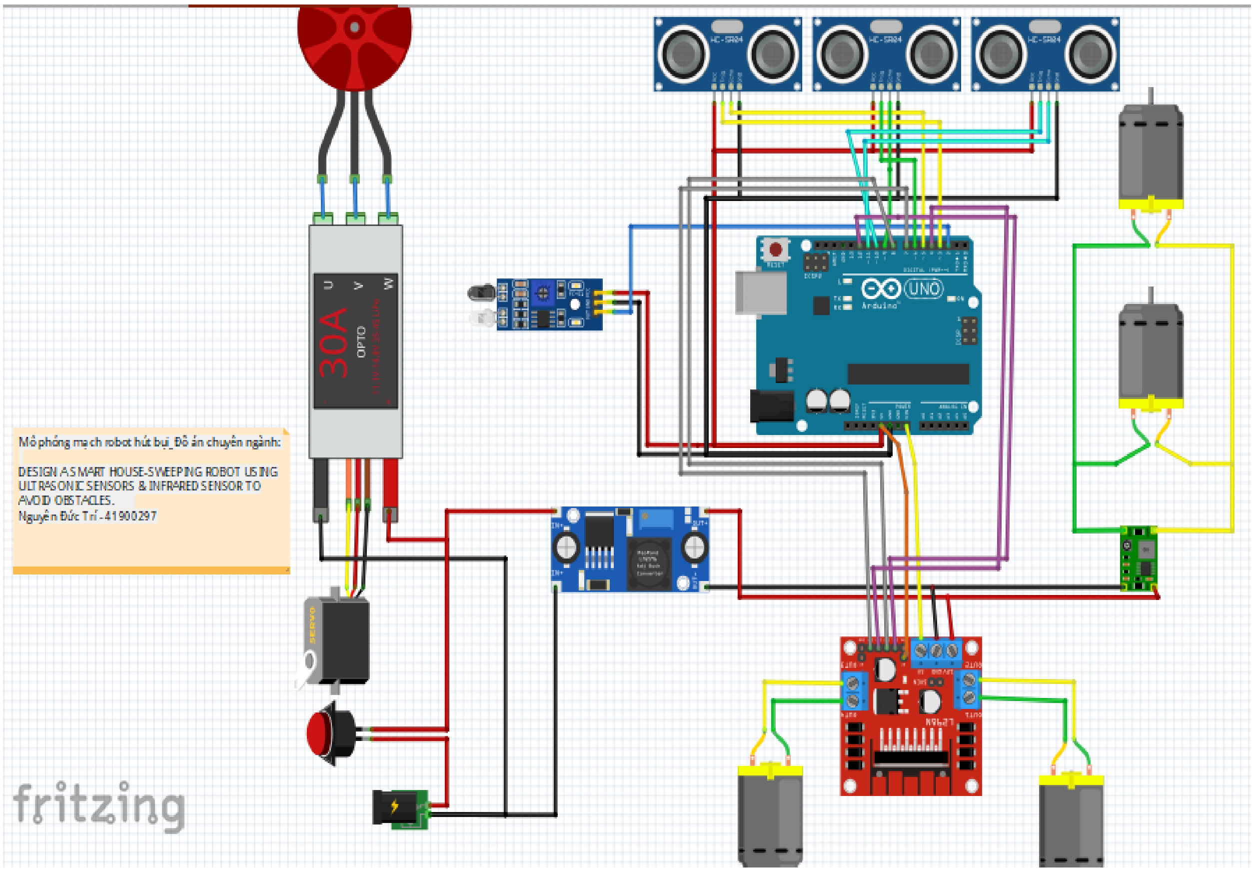

Figure 3.24. Fritzing Software is used for simulation.

3.2. CODING

```
// defining the pins
  //Khai báo chân HC-SR 04 - Cảm biến siêu âm
  //trái
const int trigPin1 = 3;
const int echoPin1 = 5;
  //giữa
const int trigPin2 = 6;
const int echoPin2 = 9;
  //phải
const int trigPin3 = 10;
const int echoPin3 = 11;
  //Khai báo chân IR sensor - Cảm biến hồng ngoại
int irpin =2;

// defining variables
long duration1; // biến đo thời gian
long duration2;
```

```
long duration3;
int distanceleft; // biến lưu khoảng cách
int distancefront;
int distanceright;
int a=0; // đặt a = 0 -> a=1 khi cảm biến hồng ngoại phát hiện chướng ngại
vật

//Chuong trinh chinh - khai bao bien, thong so
void setup()
{
  pinMode(trigPin1, OUTPUT);
  pinMode(trigPin2, OUTPUT);
  pinMode(trigPin3, OUTPUT); // Sets chân trigPin là Output - phát tín hiệu
  pinMode(echoPin1, INPUT); // Sets chân echoPin là Input - nhận tín hiệu
  pinMode(echoPin2, INPUT);
  pinMode(echoPin3, INPUT);
  pinMode(irpin, INPUT); // Sets chân cảm biến hồng ngoại là Input
  pinMode(4, OUTPUT); //Sets chân động cơ là Output
  pinMode(7, OUTPUT);
  pinMode(8, OUTPUT);
  pinMode(12, OUTPUT);
}

//Chuong trinh lap sau khi chuong trinh chinh bat dau
void loop()
{
//Cảm biến trái
  digitalWrite(trigPin1, LOW); //ngừng gửi tín hiệu
  delayMicroseconds(2);
  digitalWrite(trigPin1, HIGH);  //gửi tín hiệu để đo độ rộng xung – khoảng
cách
  delayMicroseconds(10);
  digitalWrite(trigPin1, LOW); //ngừng gửi tín hiệu -> để tín hiệu vừa được
phát đi quay về -> thu tín hiệu vô chân echo
  duration1 = pulseIn(echoPin1, HIGH); //đo độ rộng sung bằng micro giây ->
thu tín hiệu
  distanceleft = duration1 * 0.034 / 2; // chuyển đổi khoảng cách từ âm thanh
thành cm
  Serial.print("Distance1: ");
  Serial.println(distanceleft); //in khoảng cách đo được lên màn hình trực
tiếp
//Cảm biến giữa
  digitalWrite(trigPin2, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin2, HIGH);
  delayMicroseconds(10);
```

```
  digitalWrite(trigPin2, LOW);
  duration2 = pulseIn(echoPin2, HIGH);
  distancefront = duration2 * 0.034 / 2;
  Serial.print("Distance2: ");
  Serial.println(distancefront);
//Cảm biến phải
  digitalWrite(trigPin3, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin3, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin3, LOW);
  duration3 = pulseIn(echoPin3, HIGH);
  distanceright = duration3 * 0.034 / 2;
  Serial.print("Distance3: ");
  Serial.println(distanceright);
  int s = digitalRead(irpin);
  if(s==HIGH) // không có vật cản
  {
    digitalWrite(4, LOW);
    digitalWrite(7, HIGH);
    digitalWrite(8, LOW);
    digitalWrite(12, HIGH);
    delay(1000);// 1 giây
    a=1;
    }
   if  ((a==0)&&(s==LOW)&&(distanceleft  <=  20  &&  distancefront  >  20  &&
distanceright <= 20) || (a==0)&&(s==LOW)&&(distanceleft > 20 && distancefront
> 20 && distanceright > 20))
//Nếu (a=0-VÀ-slow-VÀ-(trái<=20cm,giữa>20cm,phải<=20cm) HOẶC a=0-VÀ-slow-VÀ-
(trái >20,giữa>20,phải<=20)) -> Chỉ cần 1 trong 2 cái, nếu cái nào TRUE ->
TRUE, nếu cả 2 cùng FALSE -> FALSE
//S low -> ko có vật cản
  { //đi thẳng
    digitalWrite(4, HIGH);
    digitalWrite(7, LOW);
    digitalWrite(8, HIGH);
    digitalWrite(12,LOW);
  }
  if ((a==1)&&(s==LOW)||(s==LOW)&&(distanceleft <= 20 && distancefront <= 20
&& distanceright > 20)||(s==LOW)&&(distanceleft <= 20 && distancefront <= 20
&& distanceright > 20)||(s==LOW)&& (distanceleft <= 20 && distancefront > 20
&&  distanceright  >  20)||(distanceleft  <=  20  &&  distancefront  >  20  &&
distanceright > 20))
//Nếu (a=1-VÀ-slow-VÀ-(trái<=20cm,giữa>20cm,phải<=20cm) HOẶC a=0-VÀ-slow-VÀ-
(trái >20,giữa>20,phải<=20)) -> Chỉ cần 1 trong 2 cái, nếu cái nào TRUE ->
TRUE, nếu cả 2 cùng FALSE -> FALSE
```

```
  { //rẻ phải
    digitalWrite(4, HIGH);
    digitalWrite(7, LOW);
    digitalWrite(8, LOW);
    digitalWrite(12, HIGH);
    delay(100);
    a=0;
  }
  if ((s==LOW)&&(distanceleft > 20 && distancefront <= 20 && distanceright <=
20) ||(s==LOW)&& (distanceleft > 20 && distancefront > 20 && distanceright <=
20) ||(s==LOW)&& (distanceleft > 20 && distancefront <= 20 && distanceright
> 20) )
  { //rẻ trái
    digitalWrite(4, LOW);
    digitalWrite(7, HIGH);
    digitalWrite(8, HIGH);
    digitalWrite(12, LOW);
  }
}
```

3.3. REQUIREMENTS AND RESOURCES

Table 3.16. Supplies & Components

STT	NAME OF COMPONENT	QUANTITY
1	Arduino Uno R3 chip CH340G	1
2	Ultrasonic Sensor HC-SR04	3
3	IR Sensor (Infrared obstacle sensor LM393)	1
4	Module L298N	1
5	Module DC LM2596 3A	1
6	Module DC PWM 5A, 3-35V	1
7	Module RC Servo	1
8	Module ESC 30A	1
9	Yellow gear motor	4
10	Motor Nidec BLDC	1
11	Battery 3S Lithium	1
12	Switch	1
13	Mini brush	2
14	Main brush	1
15	Wheel	2

CHAPTER 4. RESULTS & EVALUATION

4.1 Completed product

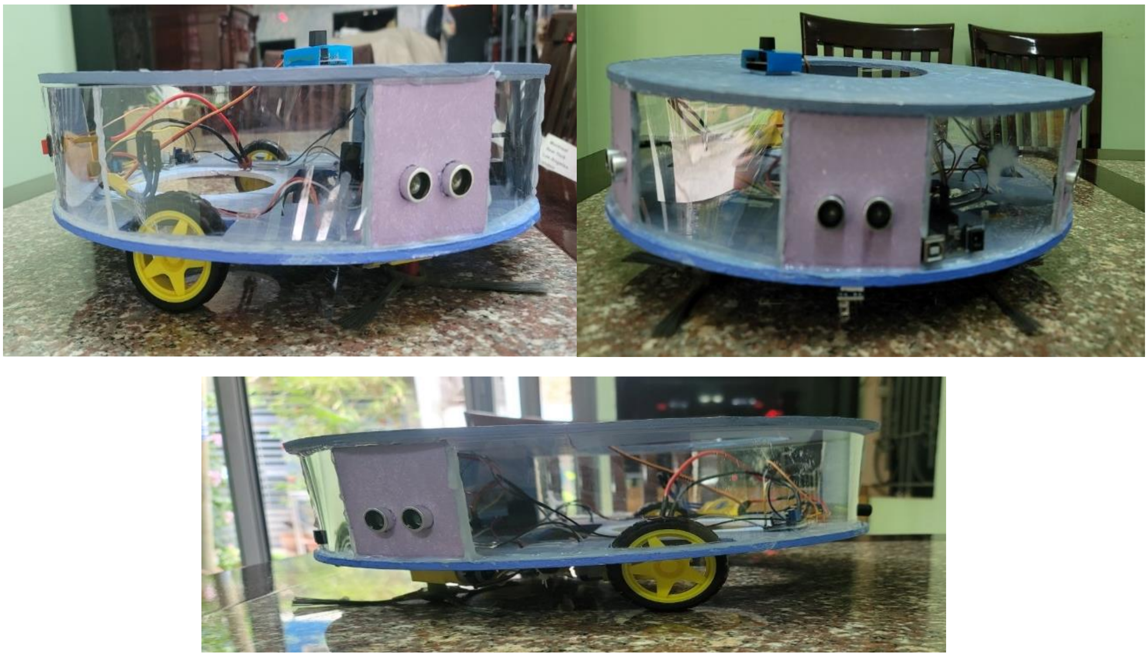

Figure 4.1. Model in reality.

4.2 Evaluation

The entire assessment below will be based on the main goal I initially want to aim for:

Target	Consequence
Compact design	☑
Ability to avoid obstacles	☒
Ability to avoid falling	☑
Vacuuming ability	☑

Table 4.1. Evaluation table.

REFERENCES

[1]. Design, manufacture, and simulate vacuum cleaner robots

[2]. Making a robot vacuum cleaner to avoid falling down the stairs

[3]. Mechano smart floor cleaning robot using Arduino

[4]. How to make a simple smart robot vacuum cleaner with Arduino

[5]. Instructions for simulating and adding component libraries in Proteus

[6]. Fritzing circuit design software

[7]. Instructions for simulating on Fritzing

[8]. Fritzing Forum

[9]. How to export files from Arduino IDE for simulation

[10]. Research and manufacture automatic vacuum cleaner robots

APPENDIX CODE

A. Defining the pins

```
const int trigPin1 = 3;
const int echoPin1 = 5;
const int trigPin2 = 6;
const int echoPin2 = 9;
const int trigPin3 = 10;
const int echoPin3 = 11;
int irpin =2;
```

B. Defining variables

```
long duration1;
long duration2;
long duration3;
int distanceleft;
int distancefront;
int distanceright;
int a=0;
```

C. Main program

```
void setup()
{
  pinMode(trigPin1, OUTPUT);
  pinMode(trigPin2, OUTPUT);
  pinMode(trigPin3, OUTPUT);
  pinMode(echoPin1, INPUT);
  pinMode(echoPin2, INPUT);
  pinMode(echoPin3, INPUT);
  pinMode(irpin, INPUT);
  pinMode(4, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(8, OUTPUT);
  pinMode(12, OUTPUT);
}

void loop()
{
  digitalWrite(trigPin1, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin1, HIGH);
  delayMicroseconds(10);
```

```
  digitalWrite(trigPin1, LOW);
  duration1 = pulseIn(echoPin1, HIGH);
  distanceleft = duration1 * 0.034 / 2;
  Serial.print("Distance1: ");
  Serial.println(distanceleft);
  digitalWrite(trigPin2, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin2, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin2, LOW);
  duration2 = pulseIn(echoPin2, HIGH);
  distancefront = duration2 * 0.034 / 2;
  Serial.print("Distance2: ");
  Serial.println(distancefront);
  digitalWrite(trigPin3, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin3, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin3, LOW);
  duration3 = pulseIn(echoPin3, HIGH);
  distanceright = duration3 * 0.034 / 2;
  Serial.print("Distance3: ");
  Serial.println(distanceright);
  int s = digitalRead(irpin);
  if(s==HIGH)
  {
    digitalWrite(4, LOW);
    digitalWrite(7, HIGH);
    digitalWrite(8, LOW);
    digitalWrite(12, HIGH);
    delay(1000);
    a=1;
    }
   if ((a==0)&&(s==LOW)&&(distanceleft <= 20 && distancefront > 20 && distanceright <= 20) || (a==0)&&(s==LOW)&&(distanceleft > 20 && distancefront > 20 && distanceright > 20))
  {
    digitalWrite(4, HIGH);
    digitalWrite(7, LOW);
    digitalWrite(8, HIGH);
    digitalWrite(12,LOW);
  }
```

```
  if ((a==1)&&(s==LOW)||(s==LOW)&&(distanceleft <= 20 && distancefront <= 20 && distanceright > 20)||(s==LOW)&&(distanceleft <= 20 && distancefront <= 20 && distanceright > 20)||(s==LOW)&& (distanceleft <= 20 && distancefront > 20 && distanceright > 20)||(distanceleft <= 20 && distancefront > 20 && distanceright > 20))
  {
    digitalWrite(4, HIGH);
    digitalWrite(7, LOW);
    digitalWrite(8, LOW);
    digitalWrite(12, HIGH);
    delay(100);
    a=0;
  }
  if ((s==LOW)&&(distanceleft > 20 && distancefront <= 20 && distanceright <= 20) ||(s==LOW)&& (distanceleft > 20 && distancefront > 20 && distanceright <= 20) ||(s==LOW)&& (distanceleft > 20 && distancefront <= 20 && distanceright > 20) )
  {
    digitalWrite(4, LOW);
    digitalWrite(7, HIGH);
    digitalWrite(8, HIGH);
    digitalWrite(12, LOW);
  }
}
```