



People's Democratic Republic of Algeria Ministry of
Higher Education and Scientific Research



University of Echahid Hamma Lakhdar in El-Oued

Faculty of Science and Technology Field: Electrical Engineering

End of study thesis

Presented for graduation

ACADEMIC MASTER

Domain: Science and Technology

Sector: Telecommunications

Specialty: Telecommunication Systems

Theme

**Constructing a Robot with Versatile
Capabilities: Integrating Voice Control and
Advanced Obstacle Navigation**

Realized by: Fekih awssem abd hamid , Benaissa nassim

Before the jury composed of:

Mr.	University Echahid Hamma Lakhdar	President
Mr.	University Echahid Hamma Lakhdar	Examinator
Mr.AjgouRiadh	University Echahid Hamma Lakhdar	Supervisor

2023-2024

وَمَا تَوْفِيقِي إِلَّا بِاللَّهِ



Thanks

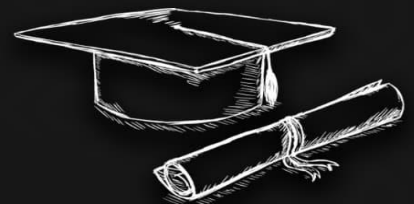
First of all, praise be to Almighty God who was with us in every step and path, we thank Him for our success.

*We would like to thank our supervisor, **Mr. Aigou Riadh**, for all the help he has given us, for his valuable advice, and for his confidence in us.*

Our sincere thanks also go to the jury members for their interest in our research and for their acceptance to examine our work.

Our thanks go to all the teachers who have supported us in our university career.

Thank you to everyone who contributed, directly or indirectly, to the realization of this project.



Dedication

Fekih Awssem Abd hamid

*To my beloved parents and my entire family,
whose unwavering dedication and endless support have been the foundation of
my upbringing and education.*

*To the teachers, mentors, and guides who have imparted wisdom and shaped
my intellect from my earliest days in primary school to the esteemed halls of
university.*

*To the brave children, resilient youth, devoted parents, and the noble martyrs of
Gaza, whose sacrifices in the land of honor and resilience inspire us all.*

*To the courageous souls who stand firm in the face of adversity, defending their
homeland with valor and integrity.*

*Your strength, perseverance, and unwavering to your enduring spirit and
boundless courage commitment fuel my aspirations and dreams.*

This work is a tribute

Dedication

Benaissa Nassim

*To my beloved parents and my entire family,
whose unwavering dedication and endless support have been the foundation of
my upbringing and education.*

*To the teachers, mentors, and guides who have imparted wisdom and shaped
my intellect from my earliest days in primary school to the esteemed halls of
university.*

*To the brave children, resilient youth, devoted parents, and the noble martyrs of
Gaza, whose sacrifices in the land of honor and resilience inspire us all.*

*To the courageous souls who stand firm in the face of adversity, defending their
homeland with valor and integrity.*

*Your strength, perseverance, and unwavering to your enduring spirit and
boundless courage commitment fuel my aspirations and dreams.*

This work is a tribute

Summary

This project explores the development of a versatile robot with integrated voice control and advanced obstacle navigation capabilities using an Arduino platform. The robot is designed to navigate autonomously, avoid obstacles, and be controlled remotely via Bluetooth and voice commands. Ultrasonic sensors are employed to detect obstacles and provide real-time data for precise navigation. The Bluetooth module allows users to control the robot using a mobile device, while voice commands offer an additional layer of convenience and flexibility. The integration of these control methods highlights the potential for creating adaptable and user-friendly robotic systems.

The project demonstrates the successful combination of various hardware components and software algorithms to achieve a functional and reliable robotic system. The methodologies and technologies used in this project can be applied to a wide range of applications, such as autonomous vehicles, delivery robots, and automated warehouse systems. By integrating multiple control systems, this project contributes to the advancement of robotics, showcasing the potential for future innovations in intelligent and interactive robotic technologies.

Key words: Versatile Robot, Voice Control, Bluetooth Control, Arduino, Obstacle Navigation, Autonomous Navigation.

المخلص

يستكشف هذا المشروع تطوير روبوت متعدد الاستخدامات يتمتع بإمكانيات التحكم الصوتي والملاحة المتقدمة لتجنب العوائق باستخدام منصة Arduino. تم تصميم الروبوت للتنقل بشكل مستقل وتجنب العوائق والتحكم فيه عن بُعد عبر البلوتوث والأوامر الصوتية. تُستخدم أجهزة الاستشعار بالموجات فوق الصوتية لاكتشاف العوائق وتوفير البيانات في الوقت الفعلي للملاحة الدقيقة. يتيح جهاز البلوتوث للمستخدمين التحكم في الروبوت باستخدام جهاز محمول، بينما توفر الأوامر الصوتية طبقة إضافية من الراحة والمرونة. يُظهر دمج طرق التحكم هذه الإمكانيات لإنشاء أنظمة روبوتية قابلة للتكيف وسهلة الاستخدام.

يُظهر المشروع التوفيق الناجح بين مكونات الأجهزة المختلفة والخوارزميات البرمجية لتحقيق نظام روبوتي وظيفي وموثوق. يمكن تطبيق المنهجيات والتقنيات المستخدمة في هذا المشروع على مجموعة واسعة من التطبيقات، مثل المركبات ذاتية القيادة والروبوتات التوصيل وأنظمة المستودعات الآلية. من خلال دمج أنظمة التحكم المتعددة، يساهم هذا المشروع في تقدم مجال الروبوتات، ويبرز الإمكانيات للابتكارات المستقبلية في تقنيات الروبوتات الذكية والتفاعلية.

الكلمات المفتاحية: روبوت متعدد الاستخدامات، التحكم الصوتي، التحكم عبر البلوتوث، أردوينو، تجنب العوائق، الملاحة الذاتية

Résumé

Ce projet explore le développement d'un robot polyvalent avec des capacités de contrôle vocal intégré et de navigation avancée pour éviter les obstacles en utilisant une plateforme Arduino.

Le robot est conçu pour naviguer de manière autonome, éviter les obstacles et être contrôlé à distance via Bluetooth et commandes vocales. Des capteurs ultrasoniques sont utilisés pour détecter les obstacles et fournir des données en temps réel pour une navigation précise. Le module Bluetooth permet aux utilisateurs de contrôler le robot à l'aide d'un appareil mobile, tandis que les commandes vocales offrent une couche supplémentaire de commodité et de flexibilité. L'intégration de ces méthodes de contrôle met en lumière le potentiel de création de systèmes robotiques adaptables et conviviaux.

Le projet démontre la combinaison réussie de divers composants matériels et algorithmes logiciels pour réaliser un système robotique fonctionnel et fiable. Les méthodologies et technologies utilisées dans ce projet peuvent être appliquées à un large éventail d'applications, telles que les véhicules autonomes, les robots de livraison et les systèmes automatisés d'entrepôt. En intégrant plusieurs systèmes de contrôle, ce projet contribue à l'avancement de la robotique, en montrant le potentiel d'innovations futures dans les technologies robotiques intelligentes et interactives.

Mots-clés : Robot Polyvalent, Contrôle Vocal, Contrôle Bluetooth, Arduino, Navigation d'Obstacle, Navigation Autonome.

Table of contents

Contents

<i>General introduction</i>	1
Chapter I: Robots using Arduino	1
I.1. Introduction:	2
I.2. Need for Arduino:.....	2
I.3. Arduino Board:	3
I.3.1 Arduino UNO:	4
I.3.2. Arduino Due:	5
I.3.3. Arduino Mega:	6
I.3.4 Arduino Leonardo:	6
I.4. Common components of Arduino boards:.....	8
I.5. Features of Arduino:	8
I.6. Selection Arduino to make projects:	9
Conclusion:	10
Chapter II: Hardware Components	10
II.1. Introduction:.....	11
II.2. Arduino Board Selection:.....	11
II.3. Motor drivers and Motors:	14
II.3.1. Brushed DC Motor (Permanent Magnet Type):	14
II.3.2. Stepper Motors:	15
II.4. Microcontroller:	16
II.5. Gear Motors:	18
II.6. Servo Motors:.....	18
II.7. L293d motor drive:	20
II.8. Ultrasonic Sensor:	21
II.9. Bluetooth module HC-05:	22
II.10. Power supply:	23
II.10. 1. Description of the selected power supply solution:	23
II.10.1.a. LiPo Battery:	23
II.10. 1. b. Voltage Regulation:	24
II.10.1.c Power Distribution:	24
II.10.1.d Safety Measures:	24

II.11.Discuss the power requirements for the robot:	24
II.11.1 Motor Power:	24
II.11.2 Voltage and Current:	25
II.11.3 Battery Capacity:	25
II.11.4 Battery Type:	25
II.11.5 Voltage Regulators:	25
II.11.6 Safety Measures:	25
II.11.Present the mechanical design of the robot:.....	26
Conclusion:	26
III.1. Introduction:.....	27
III.2. Electrical diagram:.....	27
III	
III.4 Synaptic diagram:.....	
III.1. Introduction:	27
III.2. Electrical diagram:.....	27
III.3. Assembly Implementation:	27
III.3.1.Hardware part :	28
III.3.1.a Step 1 :	28
III.3.1.b Step 2 :	28
III.3.1.c Step 3 :	29
III.3.1.d Step 4:	29
III.3.1.e Step 5:	30
III.3.1.f Step 6:	30
III.4 Synoptic diagram:.....	31
III.4.1. Synoptic diagram of obstacle avoidance:	31
III.4.2 Synoptic diagram of Bluetooth control:	32
IV.1 Program of project:	34
IV.2. Implementation:	38
IV.2.1. Steps to operate an Obstacle avoidance robot:	38
IV.2.1.a Step 1 :	38
IV.2.1.c Step 3:	39
IV.2.2. Steps to operate a Bluetooth control:	39
IV.2.2.a Step 1:	39
IV.2.2.b Step 2:	39

IV.2.2.c Step 3:	39
IV.2.2.d Step 4:	41
IV.2.3 Steps to operate a voice control:	41
IV.2.3.a Step 1:	41
IV.2.3.b Step 2:	41
IV.4.3.c Step 3 :	46
Conclusion	46
General conclusion:	47

Figures List

Figure I 1 : classification of Arduino boards	4
Figure I 2: ATmega328 Pin Diagram	4
Figure I 3:Arduino uno board	5
Figure I 4: Arduino due board.....	5
Figure I 5 :Arduino mega2560 board.....	6
Figure I 6: visual representation of the Arduino Leonardo.....	7
Figure II 1: Arduino Board Selection.....	12
Figure II 2 : DC motor.....	14
Figure II 3A typical DC motor disassembled for viewing.....	15
Figure II 4:A few bipolar stepper motors, noted as having four wires on each motor.....	16
Figure II 5: microcontroller.....	17
Figure II 6: Here you can see a small DC gear motor (left) and again with the DC motor removed from the plastic gear box.....	18
Figure II 7: A dismantled view of the larger Servo motor.....	19
Figure II 8:L293d motor drive.....	20
Figure II 9 : ultrasonic sensor.....	22
Figure II 10: Bluetooth module HC-05.....	23
Figure II 11: Robot Chassis Components.....	26
Figure III 1: electrical diagram.....	27
Figure III 2: Welding the motor wires.....	28
Figure III 3: Installing motors on the chassis	28
Figure III 4 :Arduino board with L293d motor drive.....	28
Figure III 5: connect servo motor and ultrasonic sensor.....	29

Figure III 6 : connect the Bluetooth module.....	29
Figure III 7: connecting motors.....	30
Figure III 8: battery and battery holder.....	30
Figure III 9: Synoptic diagram.....	31
Figure III 10: Synoptic diagram.....	32
Figure III. 11: Synoptic diagram.....	32
Figure IV.1: modification of code for obstacle avoiding.....	38
Figure IV.8: upload code.....	38
Figure IV.9: modification of code for Bluetooth control	39
Figure IV.10:app.....	39
Figure IV.11:	40
Figure IV.12:	40
Figure IV.13:	40
Figure IV.14:	41
Figure IV.15: modification of code for voice control	41
Figure IV.16: app.....	41
Figure IV.17:	42
Figure IV.18:	42
Figure IV.19:	43
Figure IV.20:	43
Figure IV.21:	43
Figure IV.22:	43
Figure IV.23:	44
Figure IV.24:	44
Figure IV.25:.....	45
Figure IV.26:	45

Figure IV.27:45
Figure IV.28:45

Tables List

Table I.1: comparison of Arduino board..... 26

Abbreviations

IDE : Integrated Delopment Environment

PWM : pulse width modulation

ICSP : in-cercuit serial Programming

USB : universal serial bus

AC : Alternative current

DC : Direct current

FDI USB : future technology devices international

LEDs : light emitting diodes

AREF : Analog Reference

CPU : Central processing Unit

IC : Integrated Circuit

MOSI : Master Out slave In

MISO : Maste In Slave Out

SCK : Serial Clock

PMDC : Permanent Magnet Direct Current

PLDC : Primary leadership Development Course

RAM : Random Access Memory

RPM : Revolutions Per Minute

EMF : Electromotive Force

PCB : Printed Circuit Board

SRF05 : Ultrasonic Distance Sensor

EDR : Enhanced Data Rate

ISM : Industrial , Scientific , Medical

GFSK : Gaussian Frequency Shift Keying

VCC : Voltage Common Collector

GND : Ground

TXD : Transmit Data

RXD : Receive Data

HID : Human Interface Device

LIPO : Lithium Polymer

AH : Ampere-hour

General introduction

Embedded electronics are increasingly used in our daily lives; they are used in many objects such as phones, electronic organizers, and cars. The goal of using embedded systems is to give everyday objects the ability to react to their environment.

An autonomous robot is designed to navigate an unstructured and unknown environment without any external interference. This capability is enabled by the embedded software intelligence that allows the robot to sense its surroundings, detect obstacles in its path, and maneuver around these obstacles. When designing an autonomous robot, various robotic designs can be considered, with the primary factor being the physical environment in which the robot will operate. Examples of autonomous robots include walking robots, drones, robotic cars, and snake robots.

The obstacle-avoiding robot possesses sufficient intelligence to cover the maximum area of the given space. It utilizes an ultrasonic sensor to detect obstacles in its path and then moves in a direction to avoid them.

The main aim of this technology is that it can play a significant role in today's transportation, as it can be used to avoid accidents that typically occur on congested roads by applying emergency brakes. Additionally, it can be used in developing robots for transporting orders and organizing goods in warehouses, thereby facilitating picking and distribution processes, saving time, and reducing effort.

This thesis is composed of three chapters:

Chaptre I : In this chapter, we talk Arduino in general, covering its components, types, control methods, and a brief historical overview.

Chaptre II : In this chapter, we discuss in detail the selected Arduino board used in the project, the motors, the microcontroller, and the components necessary to complete the project, as well as the car chassis.

Chapter III : In this section, we assembled the various parts of the car and connected all the components to the Arduino. We provided the working code along with a brief explanation, and finally, presented each of the three features individually.

Chapter I: Robots using Arduino

<i>I.1. Introduction</i>	<i>2</i>
<i>I.2. Need for Arduino</i>	<i>2</i>
<i>I.3. Arduino Board.....</i>	<i>3</i>
<i>I.3.1. Arduino UNO.....</i>	<i>4</i>
<i>I.3. 2. Arduino Due.....</i>	<i>5</i>
<i>I.3.3 Arduino Meg.....</i>	<i>6</i>
<i>I.3.4. Arduino Leonardo</i>	<i>6</i>
<i>I.4. Common components of Arduino boards.....</i>	<i>8</i>
<i>I.5. Features of Arduino</i>	<i>8</i>
<i>I.6. Selection Arduino to make projects.....</i>	<i>9</i>
<i>Conclusion.....</i>	<i>10</i>

I.1. Introduction:

Arduino is an open-source microcontroller platform introduced in 2005, offering an affordable and user-friendly way to create interactive devices using sensors and actuators.

It consists of hardware, like the Arduino development board, and software, such as the Arduino IDE for code development. The IDE supports programming in C or C++, making it accessible to developers of all levels. Users can easily write code to read sensor data and control outputs.

Arduino's simplicity, affordability, and versatility have made it popular in education, hobbyist projects, and professional settings. It is widely used in IoT development by startups and companies due to its various board models supporting different communication protocols. The passionate Arduino community provides extensive resources, making project-based learning engaging for users.[1]

Overall, Arduino is the preferred platform for creating electronic projects, from simple prototypes to complex systems.

In this chapter, we discuss Arduino in general, including its components, types, and history, as well as why it was chosen for this work.

I.2. Need for Arduino:

Why is there a need to use Arduino in specific? or What makes it different from others?

Arduino is chosen for the project of constructing a robot with versatile capabilities, integrating voice control and advanced obstacle navigation due to several reasons [2]:

- **Open-Source Platform:** Arduino is an open-source platform, which means that its hardware and software specifications are freely available. This openness encourages innovation and collaboration among developers, making it an ideal choice for hobbyists, students, and professionals.
- **Versatility and Customizability:** Arduino boards provide a versatile and customizable platform for creating robots. They offer a wide range of input/output pins, analog and digital capabilities, and support for various sensors and modules. This allows for flexibility in integrating different components and functionalities according to project requirements.
- **Ease of Use and Accessibility:** Arduino boards are designed with a beginner-friendly approach, making them accessible even to those with limited programming or

electronics knowledge. The Arduino IDE (Integrated Development Environment) provides a user-friendly interface for writing and uploading code to the board, simplifying the development process.

- **Availability of Libraries and Community Support:** Arduino has a large and active community of users and developers. This translates into a vast library of pre-written code and resources that can be utilized for different functionalities, including voice control and obstacle navigation. The community support also means that troubleshooting and getting assistance is easier.
- **Cost-Effectiveness:** Arduino boards are relatively affordable compared to other microcontroller options available in the market. This makes them a cost-effective choice for hobbyists, students, and enthusiasts who may have budget constraints.
- **Integration of Voice Control and Sensor Inputs:** Arduino boards can easily interface with voice recognition modules and sensors required for obstacle navigation. By utilizing appropriate libraries and modules, the Arduino can process voice commands and sensor inputs, enabling the robot to respond and navigate autonomously.
- **Expandability and Future Development:** Arduino provides a foundation for future expansion and development. As the project evolves, additional functionalities and capabilities can be added to the robot by integrating more sensors, modules, or communication interfaces with the Arduino board.

Arduino's versatility, ease of use, community support, cost-effectiveness, and compatibility with voice control and obstacle navigation requirements make it a suitable choice for constructing a robot with versatile capabilities in this project.

I.3. Arduino Board:

The Arduino board is a versatile open-source electronics platform designed for individuals interested in creating interactive projects easily. It allows users to program various operations using an integrated development environment across different operating systems. Arduino boards can read data from sensors and buttons through their microcontrollers, enabling users to control LEDs, motors, and other components[3].

There are different types of Arduino boards available, each offering unique features and capabilities to cater to various project requirements.

In figure I.1 we show classification of Arduino board

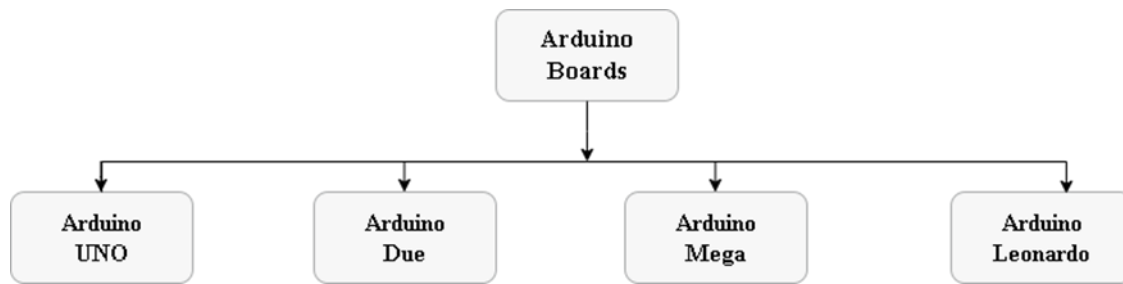


Figure I 1 : classification of Arduino boardt

I.3.1 Arduino UNO:

The Arduino Uno is a microcontroller board based on the ATmega328

In figure I.2 we show ATmega328 Pin Diagram

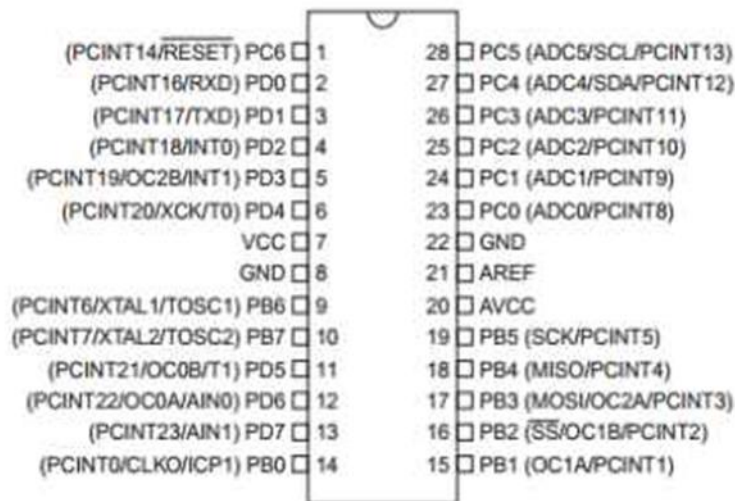


Figure I.2 : ATmega328 Pin Diagram

It features a 16 MHz ceramic resonator, 6 analog inputs, 14 digital input/output pins (six of which can be used as PWM outputs), a USB port, a power jack, an ICSP header, and a reset button.

It comes with everything needed to support the microcontroller; all you need to do is power it with a battery or an AC-to-DC adapter or connect it to a computer via a USB cable to get going.

The FTDI USB-to-serial driver chip is not used by the Uno, setting it apart from all previous boards. Rather, it has the Atmega16U2 configured as a serial-to-USB converter.[4]

“Uno” means one in Italian, and it was named to celebrate the upcoming release of Arduino 1.0. The Uno, along with version 1.0, will serve as the reference models for the Arduino platform moving forward. Uno is the latest addition to the series of USB Arduino boards, and

it plays a crucial role as the standard model for Arduino development. For a comparison with previous versions, you can refer to the index of Arduino boards.[5]

In figure I.3 we show Arduino uno board



*Figure I .3:*Arduino uno board

I.3.2. Arduino Due:

The Arduino Due is a microcontroller development board based on the AT91SAM3X8E SAM3X8E-32 BIT 32-bit ARM Cortex-M3 with a clock speed of 84MHz. This board offers a comprehensive set of features, including 54 digital I/O pins (12 of which are PWM outputs), 16 analog inputs, 4 UARTs, 2 DACs (Digital to Analog Converter), 2 CAN, 2 TWI, SPI header, JTAG header, a power jack, USB OTG capable connection, and a reset button [7-8]. This Arduino board is mostly used in powerful.[6].

In figure I.4 we show arduino due board

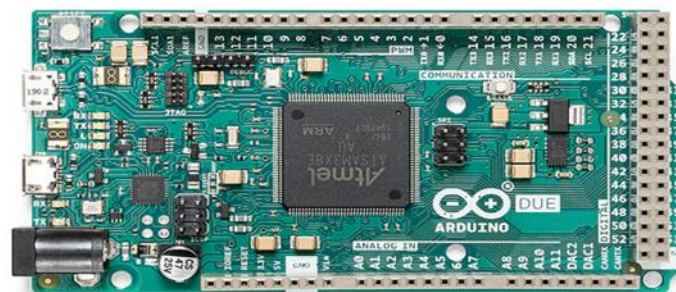


Figure I .4: arduino due board

I.3.3. Arduino Mega:

The Arduino Mega is an open-source microcontroller board built around the ATmega2560 microcontroller with a clock speed of 16 MHz. It is an 8-bit board that offers a wide range of features and I/O capabilities. With 54 digital I/O pins, including 16 analog inputs and 15 PWM output pins, it provides flexibility for connecting various components and peripherals. It also includes 4 UARTs for serial communication with other devices and an ICSP header for direct programming. The board has a power jack for external power supply, a USB connection for communication with computers or other USB devices, and a reset switch for microcontroller reset. It is compatible with shields designed for the Arduino Uno, allowing for additional functionality such as wireless communication, motor control, or sensor integration. The Arduino Mega is suitable for larger and more complex projects due to its increased memory and I/O capabilities, making it a versatile choice for applications requiring multiple sensors, actuators, or communication modules [7].

In figure I.5 we show Arduino mega2560 board

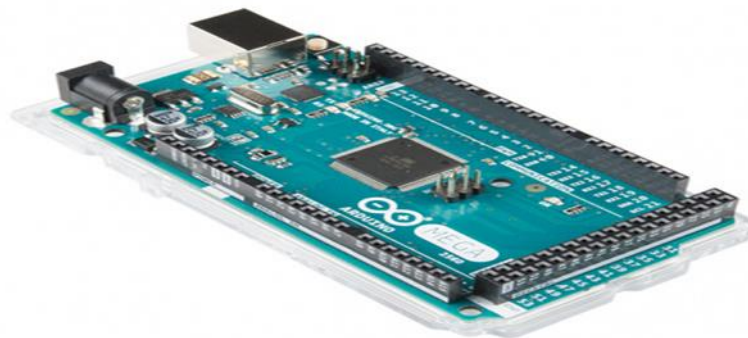


Figure I.5: Arduino mega2560 board

I.3.4 Arduino Leonardo:

The Arduino Leonardo is the first development board manufactured by the Arduino company. It is designed to be simple to use and cost-effective. The board is based on the ATmega32u4 microcontroller, which is an 8-bit AVR family member operating at a clock speed of 16 MHz. Unlike other Arduino boards, the Leonardo uses only one microcontroller with built-in USB functionality. The Arduino Leonardo features 20 digital I/O pins, with 12 of them capable of functioning as analog input pins. Additionally, it has 7 pins that can be used as PWM outputs for controlling devices with varying levels of intensity. The board includes SPI (Serial

Peripheral Interface), a micro-USB (CDC) port for connecting to a computer or other USB devices, a power jack for external power supply, and a reset switch interface [8].

In figure I.6 we show visual representation of the Arduino Leonardo.

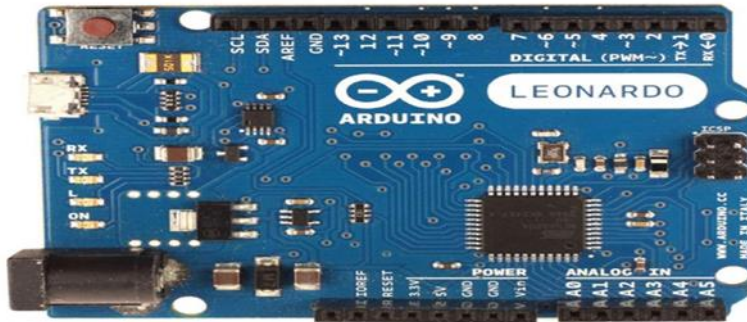


Figure I. 6: visual representation of the Arduino Leonardo.

In table I.1 we show comparison of Arduino board

Table I.1: comparison of Arduino board

Arduino	Uno	Due	Mega	Leonardo
Processor	8-bit	32-bit	8-bit	8-bit
Clock (MHz)	16	84	16	16
Flash	32,2	512,96	256,8	32,2 .5
SRAM (KB)				
Digital Pins (PWM)	14(6)	54(12)	54(15)	20(7)
Communication Interface	SPI, UART, I2C	UART, SPI, CAN, I2C	UART, SPI, I2C	TWI, SPI, USB (CDC)
Analog I/O	6/0	12/2	16/0	12/0
Voltage	5V	3.3V	5V	5V

I.4.Common components of Arduino boards:

The common components found on Arduino boards, regardless of the specific board type, include [9][10]:

- Microcontroller: Such as the ATmega328P, which acts as the main processing unit.
- Power Pins: Including 5V, 3.3V, and GND pins for power supply and grounding.
- Analog Input Pins: Used for reading analog signals from sensors like temperature sensors.
- Digital Pins: Used for digital input and output functions, including PWM pins for controlling things like LED brightness or motor speed.
- USB Port: For uploading code to the board, serial communication, and powering the board.
- Reset Button: Allows you to restart the program on the board.
- Voltage Regulator: Controls and stabilizes the voltage supplied to the board.
- Crystal Oscillator: Manages time-related functions by providing clock signals.

These components are essential for the functionality of Arduino boards and are consistent across various

I.5. Features of Arduino:

The features of Arduino are listed below [11][12]:

- Microcontroller boards capable of detecting signals (analog or digital) from different sensors and converting them into outputs, triggering actions such as activating motors, turning LEDs on or off, and more.
- Control of microcontroller board functions through the Arduino Integrated Development Environment (IDE) by sending instructions.
- No external hardware required for loading code into the board; just a USB cable is needed.
- Simplified version of C++ programming language used in Arduino, making it easy to learn.
- Standard form factor, breaking down board functions into a more accessible package.
- Arduino IDE used to control board functions and send specifications to the microcontroller.
- Arduino capable of reading both analog and digital input signals.

- Easy-to-use and implement hardware and software platform.

I.6. Selection Arduino to make projects:

Arduino is an open-source electronics platform based on easy-to-use hardware and software. It consists of a physical programmable circuit board (often referred to as a microcontroller) and a development environment for writing, compiling, and uploading code to the board. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so, you use the Arduino programming language (based on Wiring) and the Arduino Software (IDE), based on Processing.[13]

Arduino is often chosen for projects because of its simplicity and accessibility, especially for beginners and hobbyists. Here are some key reasons why it's popular:

1. **Open-Source:** Arduino's hardware and software are open-source, meaning the designs and code are freely available for anyone to use, modify, and distribute. This has led to a large community of users and developers contributing to the platform.
2. **Easy to Use:** Arduino was designed with simplicity in mind, making it accessible to beginners with no prior electronics or programming experience. The programming language is relatively straightforward, and there are plenty of resources and tutorials available online.
3. **Versatility:** Arduino boards can be used for a wide range of projects, from simple blinking LED lights to complex robotics. They can interact with a variety of sensors, motors, and other components, making them suitable for many different applications.
4. **Affordability:** Arduino boards are relatively inexpensive compared to other microcontroller platforms, making them accessible to a wide range of people, including students and hobbyists.
5. **Large Community and Ecosystem:** Arduino has a large and active community of users and developers, which means there are plenty of resources, libraries, and examples available online. This makes it easier to get started and troubleshoot problems.

Conclusion:

In conclusion, the Arduino platform offers a user-friendly and versatile environment for developing a wide range of projects, including robotics. Its open-source nature, ease of use, and large community make it an attractive choice for hobbyists, students, and professionals alike.

For the specific project of constructing a robot with voice control and advanced obstacle navigation, Arduino's flexibility and compatibility with various sensors and components make it a suitable platform. The integration of voice control adds a unique and interactive element to the project, while advanced obstacle navigation algorithms can be implemented using Arduino's programming capabilities.

Overall, Arduino's combination of accessibility, affordability, and extensibility makes it an ideal platform for innovative and creative projects like the one described.

Chapter II: Hardware Components

<i>II.1. Introduction</i>	<i>11</i>
<i>II.2. Arduino Board Selection</i>	<i>11</i>
<i>II.3. Motor drivers and Motors</i>	<i>14</i>
<i>II.4. Microcontroller</i>	<i>16</i>
<i>II.5. Gear Motors</i>	<i>18</i>
<i>II.6. Servo Motors</i>	<i>28</i>
<i>II.7. L293d motor drive</i>	<i>20</i>
<i>II.8. Ultrasonic Sensor</i>	<i>21</i>
<i>II.9. Bluetooth module HC-05</i>	<i>22</i>
<i>II.10. Power supply</i>	<i>23</i>
<i>II.11. Discuss the power requirements for the robot</i>	<i>24</i>
<i>Conclusion</i>	<i>26</i>

II.1. Introduction:

Arduino, an open-source electronics platform, is highly versatile and user-friendly, ideal for a wide array of projects. It can control various motors, including DC motors for driving wheels, servo motors for precise angular control, and stepper motors for accurate rotational motion. Sensors, such as ultrasonic, infrared, temperature, light, and accelerometers, can be integrated with Arduino to gather environmental data for applications like obstacle avoidance and object detection in robotics. Powering Arduino is simple, with options including USB from a computer or adapter, or a battery pack for portability. Its compatibility with different components and ease of programming makes Arduino a preferred choice for hobbyists, students, and professionals in creating innovative projects, from basic motor control to complex robotic systems with advanced sensing and navigation.[14]

What are the components used in this project, and how do we assemble them?

II.2. Arduino Board Selection:

The Arduino Uno is an open-source microcontroller board based on the ATmega328P microcontroller by Microchip. It has 14 digital I/O pins, 6 analog I/O pins, and is programmable using the Arduino IDE. The board can be powered via USB or a 7-20V power source like a 9-volt battery. It comes with a preprogrammed

The Atmega16U2 as a USB-to-serial converter. It finds applications in education, industry, and scientific projects due to its versatility. In terms of technical specifications, the Arduino Uno operates at 5V and has a recommended input voltage between 7-12V, with acceptable limits of 6-20V. It offers 14 digital I/O pins, including 6 with PWM (Pulse Width Modulation) outputs, and 6 analog input pins. The board runs at a clock speed of 16 MHz and provides 32 KB of flash memory, 2 KB of SRAM, and 1 KB of EEPROM.[15][16]

Overall, the Arduino Uno is a widely used microcontroller board known for its ease of use, flexibility, and compatibility with a wide range of projects.

In figure II.1 we show Arduino Board Selection (Arduino uno).

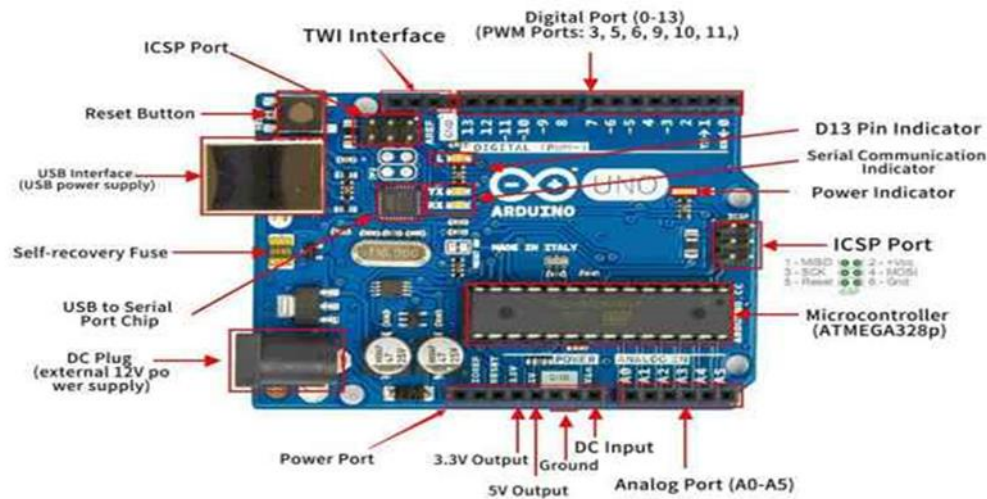


Figure II.1: Arduino Board Selection

Power USB: Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection.

- **Power (Barrel Jack):** Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack.
- **Voltage Regulator:** The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.
- **Crystal Oscillator:** The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz
- **Arduino Reset:** You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).
- **Pins (3.3, 5, GND, Vin):**
 - o 3.3V (6): Supply 3.3 output volt
 - o 5V (7): Supply 5 output volt
 - o Most of the components used with Arduino board works fine with 3.3 volt and 5 volt.

- o GND (8) (Ground): There are several GND pins on the Arduino, any of which can be used to ground your circuit.
- o Vin (9): This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.
- Analog pins: The Arduino UNO board has five analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor
- Main microcontroller: Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.
- ICSP pin: Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.
- 13 Power LED indicator: This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.
- TX and RX LEDs: On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.
- Digital I/O: The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labeled "~" can be used to generate PWM.
- AREF: AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins

II.3. Motor drivers and Motors:

Motors are electromechanical devices that convert electrical energy into mechanical motion. They are crucial components in robotics, automation, and various applications. There are different types of motors commonly used in Arduino robot projects [17]:

II.3.1. Brushed DC Motor (Permanent Magnet Type):

The brushed DC motor, also known as a Permanent Magnet Direct Current (PMDC) motor, is widely used in electronic devices, robotics, and toys [Figure II-2]. It is the simplest type of motor to drive, control, and manipulate. The motor's rotational direction can be reversed by changing the voltage polarity at the motor terminals.

In figure II.2 we show DC motors come in all shapes and sizes. The large motor is made for an electric scooter, the 2 in the center are small hobby motors, and the 2 on the right are DC servo motors.

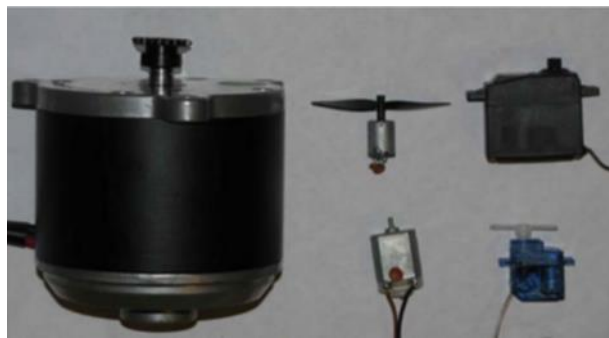


Figure II. 2 : DC motor

Most brushed DC motors have brushes that make physical contact with spinning electrical contacts called commutators, which are electrically connected to the motor coil winding. When each brush touches a different commutator, electrical current flows through the motor coil, causing the motor's output shaft to spin around its axis. In Permanent Magnet DC motors, there are typically two magnets attached to the inside of the motor casing, while the coil winding and commutator are mounted on the output shaft [Figure II-3]. The brushes are usually spring-loaded to maintain a secure connection with the commutator during operation.

In figure II.3 we show A typical DC motor disassembled for viewing

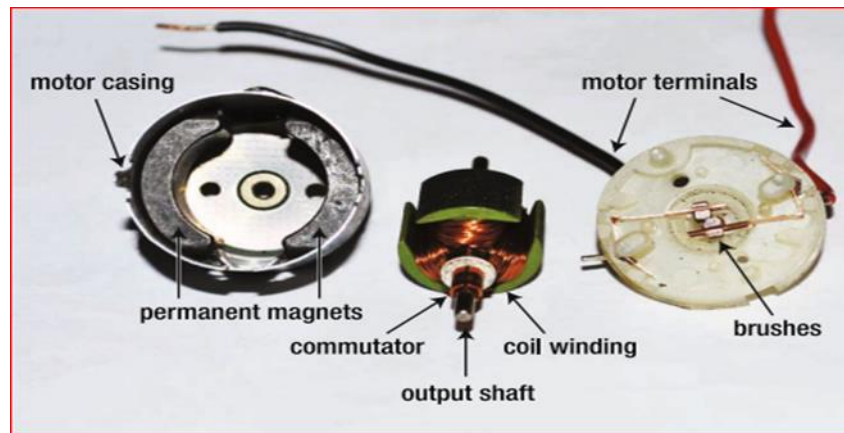


Figure II.3: A typical DC motor disassembled for viewing.

A DC motor consumes power according to its workload. It draws anywhere from 50mA to 50 amps of current and can operate at speeds ranging from 1,000RPM to 20,000RPM. By using gears, the motor's power can be transformed to provide either high-speed, low-torque output (suitable for flat surfaces) or low-speed, high-torque output (ideal for hills), thereby extending the motor's usability. Due to the physical contact involved in transferring power through the coil, it is necessary to periodically replace the motor brushes, especially for frequently used motors. Fortunately, many DC motors have easily accessible brushes that can be replaced quickly, typically within minutes.

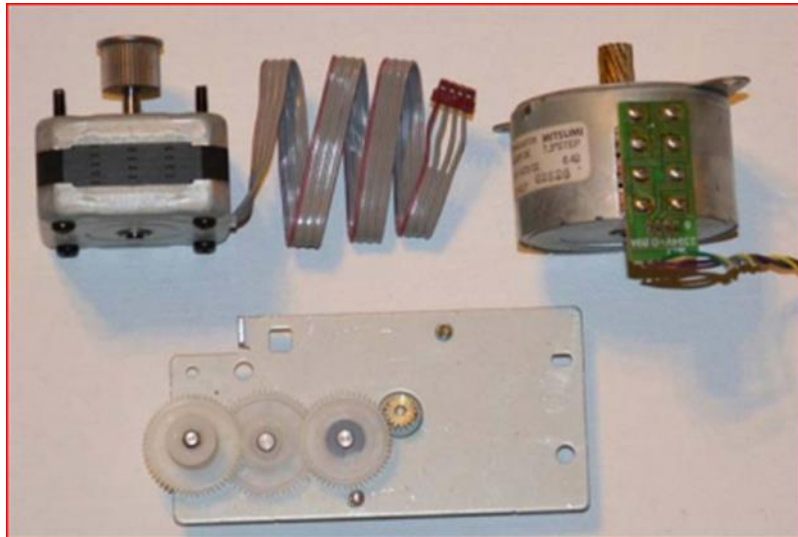
II.3.2. Stepper Motors:

Stepper motors are brushless DC motors that differ from traditional BLDC motors in that they have two or more independent coils instead of three common coils. These motors require a specific sequence and timing of energizing the coils to keep the motor shaft spinning. Applying power directly to the wires will not make the motor spin on its own. Stepper motors operate by rotating the motor shaft one step at a time, with each step corresponding to a set number of degrees. The number of steps per rotation determines the motor's resolution or precision. Stepper motors with higher step counts offer greater precision and are commonly used in CNC machines for precise positioning. There are two basic types of stepper motors: bipolar and unipolar.[18]

- Bipolar stepper motors have two coils and are typically identified by having four wires. Each coil must be switched in a specific sequence to rotate the output shaft. This requires each of the four motor wires to be driven by its own signal amplifier or a dual motor-controller.

- Unipolar stepper motors also have two coils but have additional windings in each coil, resulting in six wires (or five if the common wires are connected). These motors have a constant positive power supply to the common wire and N-type transistor switches at each negative coil wire. With the appropriate switching sequence controlled by an Arduino, unipolar stepper motors are relatively easy to drive.

In figure II.4 we show A few bipolar stepper motors, noted as having four wires on each motor



*Figure II.4:*A few bipolar stepper motors, noted as having four wires on each motor

Stepper motors are commonly used for high-precision applications like computer printer heads, CNC machines, and some robotic applications.

II.4. Microcontroller:

The most essential part of the Arduino is the Microcontroller, which is shown below [19]:

- Small and Low Power Computer:
 - The microcontroller can be thought of as a small and low-power computer system on a single chip.
 - It typically consists of key components such as a CPU (Central Processing Unit), RAM (Random Access Memory), and memory storage, similar to a traditional computer system.
- Limited Memory:
 - Microcontrollers have relatively small amounts of memory compared to full-fledged computers.

- The program memory (flash memory) available in Arduino microcontrollers typically ranges from a few kilobytes to a few hundred kilobytes.
- RAM, used for storing variables and runtime data, is also limited, usually ranging from a few hundred bytes to a few kilobytes.
- Single Program Execution:
 - Due to limited memory resources, some microcontrollers can only run a single program at a time.
 - The program stored in the microcontroller's memory is executed sequentially, performing the defined tasks and controlling the connected components.
- Integrated Peripherals:
 - Microcontrollers in Arduino boards incorporate various built-in peripherals, including GPIO (General Purpose Input/Output) pins, timers, UART (Universal Asynchronous Receiver-Transmitter) for serial communication, ADC (Analog-to-Digital Converter) for reading analog signals, and more.
 - GPIO pins serve as the interface for controlling and communicating with external electronics or circuitry, allowing the microcontroller to interact with the physical world.

Although Arduino microcontrollers are more suited for embedded systems and particular applications, their memory is usually smaller than that of full-sized computers. This enables them to work well on a variety of projects involving robotics, automation, DIY projects, and prototyping. The microcontroller is a versatile and vital part of the Arduino ecosystem, allowing users to develop interactive and personalized electronic systems. It does this by integrating memory, computing power, and I/O peripherals.

in figure II.5 we show microcontroller



Figure II. 5: microcontroller

II.5. Gear Motors:

The power of an electric motor can be adjusted to prioritize either high speed or high torque, similar to how gears on a bike can be shifted to different speeds. In a bike's low gears, such as first gear, pedaling provides high torque, allowing the rider to climb steep hills with ease but at a slower speed. On the other hand, in higher gears, like eighteenth gear, it becomes difficult to pedal uphill but allows for greater speed on flat ground or downhill. Similarly, the power output of an electric motor can be modified. This is achieved by using a series of gears connected to the motor's output shaft or by using a gear motor, which has gears built into the output shaft [Figure II-6]. The gear motor reduces the high speed of the motor's output shaft (typically ranging from 1,000 to 20,000 RPM) to a slower output speed that is more suitable for a mobile robot or other applications. A gear motor can be any type of electric motor as long as it includes a gear box that reduces the output speed. Each gear motor has a gear ratio, which specifies the relationship between the input speed and the output speed of the motor's output shaft. For example, a gear motor with a 100:1 gear ratio means that the actual DC motor output shaft must rotate 100 times to complete one revolution of the geared output shaft.[20]

In figure II.6 we show gear motor DC



Figure II. 6: Here you can see a small DC gear motor (left) and again with the DC motor removed from the plastic gear box.

II.6. Servo Motors:

A servo motor is a specialized type of DC motor equipped with an encoder to determine the position of its output shaft. Hobby servo motors are designed with a compact structure comprising a small DC motor, a speed-reducing gear box, a potentiometer shaft encoder, and motor drive circuitry. This design makes them easily compatible with Arduino microcontrollers (refer to Figure II-7).

The motor drive circuitry serves the purpose of decoding the input signal, usually an R/C servo pulse, and also driving the DC motor. These servo motors are capable of quickly moving to specific positions and require only three wires (signal, power, and ground) to interface with an Arduino.

The signal used to control the servo is a precisely timed pulse of electricity ranging from 1 millisecond to 2 milliseconds. A pulse duration of 1.5 milliseconds corresponds to the center position of the servo motor. The servo motor expects to receive an update with a new pulse approximately 50 times per second, or every 20 milliseconds. Originally designed for hobby applications such as cars, planes, boats, and helicopters, servo motors have gained popularity in robotics due to their accuracy, durability, and user-friendly nature.

General-purpose hobby servo motors can be found at local hobby stores for around \$15 or online retailers for approximately \$5 each. Hobby servo motors were specifically engineered to emulate the movement of an R/C transmitter control stick. When the control stick is pushed to its highest position, the servo motor's output shaft will rotate 90 degrees in one direction. On the other hand, moving the control stick to its lowest position causes the servo motor's output shaft to rotate 90 degrees in the opposite direction. Most hobby servo motors have a total range of about 180 degrees (equivalent to half of a complete rotation). Attempting to exceed the limits of the output shaft's movement will likely result in gear stripping.

In figure II.7 we show servo motor



Figure II.7: A dismantled view of the larger Servo motor

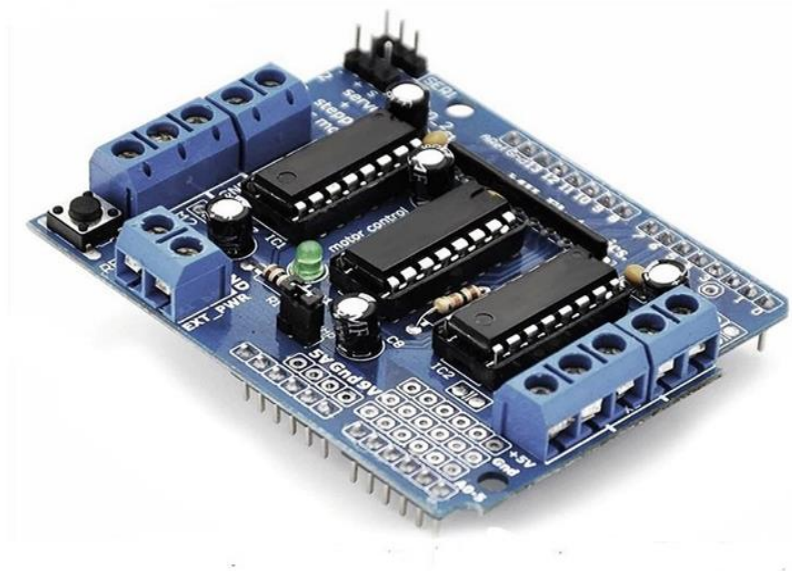
II.7. L293d motor drive:

One popular motor driver IC used for controlling DC motors and bipolar stepper motors in various electronics projects, including robotics and automation, is the L293D (shown in Figure II-8). Its direction and speed can be controlled using a microcontroller or other control logic, Key features and connections of the L293D motor driver:

- a) Dual H-bridge: The L293D contains two H-bridge circuits, enabling independent control of the direction (forward and reverse) and speed of two DC motors.
- b) Voltage compatibility: It can work with a wide range of motor voltage ratings (usually up to 36V) and logic voltage levels (usually 3.3V or 5V).
- c) Current handling: The L293D can handle moderate current loads, typically up to 600mA per channel continuously, with peak currents up to 1.2A.
- d) Built-in diodes: It has built-in flyback diodes (also called freewheeling diodes) to protect against back electromotive force (EMF) generated by the motor.

The L293D motor driver is activated when the activate input is high, allowing the outputs to interact with their information sources in real time. When the empower input is poor, the driver is rendered ineffective, and their yields suffer.[21]

In figure II.8 we show L293d motor drive



*Figure II. 8:*L293d motor drive

II.8. Ultrasonic Sensor:

Ultrasonic Sensors are widely used in Robots projects for Distance measurement as well as Obstacle detector. They are available as modules in the market.[22]

Two transducers—one for transmit and one for receive—are included in the ultrasonic module. They are both attached on a single PCB with a control circuit, making it simple to utilize in your robotics projects. From around 2 cm to 400 cm, an ultrasonic distance sensor can measure distances precisely and without contact. Animals can hear sounds at 40 KHz, which is a high frequency that humans cannot. This is known as ultrasound. Radar and ultrasonic distance measuring operate on the same basis.

Operational Principle

The ranging process is initiated by sending a brief 10 uS pulse from the Arduino to the trigger input. At 40 kHz, the Ultrasonic Module will emit an 8-cycle burst of ultrasound and elevate its echo line. To begin the range, the Arduino generates a brief 10uS pulse that is fed into the Trigger input. The Ultrasonic Module will elevate its echo line and emit an 8-cycle blast of ultrasound at 40 kHz. After that, it listens for an echo and lowers the echo line once again upon detecting one. As a result, the width of the echo line pulses in direct proportion to the object's distance from the source. It is feasible to determine the range in inches or centimeters by timing the pulse. The module will reduce its echo line in any case after around 30 milliseconds if nothing is detected. An echo pulse proportionate to distance is provided by the module. If the width of the pulse is measured in uS, then dividing by 58 will give you the distance in cm, or dividing by 148 will give the distance in inches.

$\text{uS}/58 = \text{cm}$ or $\text{uS}/148 = \text{inches}$.

The module may be activated up to 20 times per second, or once every 50 milliseconds. In the event that the SRF05 identifies a nearby object and the echo pulse is shorter, you should still wait 50 ms before the next trigger. This is to make sure that the ultrasonic "beep" has disappeared and won't interfere with the next range by creating a false echo. The sensor can detect objects within 3cm to 3m range

In figure II.9 we show ultrasonic sensor



Figure II.9 : ultrasonic sensor

II.9. Bluetooth module HC-05:

The HC-05 Bluetooth module is a versatile wireless communication module designed for various applications like wireless headsets, game controllers, and more [23]. It operates on Bluetooth version 2.0 + EDR (Enhanced Data Rate) in the 2.4 GHz ISM band, supporting a range of approximately 10 meters in open air [23]. The module uses GFSK modulation and offers a variety of baud rates for communication, ranging from 1200 to 460800.

Key Features of HC-05 Bluetooth Module:

- Supports both master and slave configurations for flexible usage.
- Utilizes serial communication via UART for interfacing with devices like microcontrollers.
- Features pins like Key/EN, VCC, GND, TXD, RXD, and State for connection and control.
- Operates on 3.3V to 5V DC, with low operating and standby currents for energy efficiency.
- Compatible with profiles like SPP (Serial Port Profile) and HID (Human Interface Device).

The HC-05 module is widely used due to its ease of use, compatibility with various devices, and reliable wireless communication capabilities. It is a popular choice for projects requiring Bluetooth connectivity, offering a cost-effective and efficient solution for wireless data transmission.

In figure II.10 we show Bluetooth module HC-05

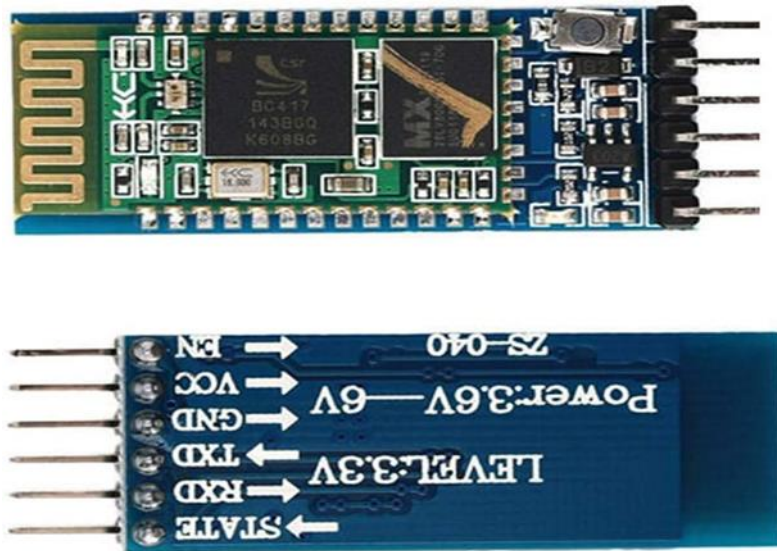


Figure II.10: Bluetooth module HC-05

II.10. Power supply:

Servomotor is in operation. The combined current drain of all servo motors operating simultaneously is 0.5A. To further enhance the project's robot mobility, an Arduino was utilized, which required 5 V. Five volts of power supply meets this the power source chosen to supply the servomotors' control circuit can offer the same current even in the event that every synchronous need.[24]

II.10. 1. Description of the selected power supply solution:

Selecting the right power source is crucial for the robot's performance and longevity. Here's a breakdown of the chosen solution [24]:

II.10.1.a. LiPo Battery:

- Lithium Polymer (LiPo) batteries are a popular choice for robotics due to their high energy density and lightweight design.
- The selected LiPo battery has a capacity of 5200mAh and operates at 11.1V. This choice provides a good balance between capacity and weight for your robot's power requirements.
- LiPo batteries are rechargeable and offer consistent voltage output during discharge, providing a stable power supply for your components.

II.10. 1. b. Voltage Regulation:

- To power the Arduino Uno and other components, a switching regulator is used.
- Switching regulators efficiently step down the LiPo battery's voltage to the required level, such as 5V for the Arduino.
- These regulators use switching techniques to minimize energy loss and maintain a stable output voltage, even as the battery voltage decreases.

II.10.1.c Power Distribution:

- Design a robust power distribution system that connects all components in the robot, such as motors, sensors, microcontrollers, and other peripherals.
- Using appropriate connectors, wires and rods to efficiently distribute power throughout the robot.
- Using a power unit or distribution panel that provides multiple power rods at voltages and specified current capabilities, allowing different components to be connected and operated simultaneously.

II.10.1.d Safety Measures:

- o Overcurrent Protection: Prevent excessive current draw that could damage components.
- o Short-Circuit Protection: Safeguard against accidental short circuits.
- o Thermal Protection: Monitor temperature to prevent overheating.
- o Properly insulate and ground components to prevent electrical hazards.

II.11. Discuss the power requirements for the robot:

When assessing the power requirements for an Arduino robot, it's important to consider various components and their power consumption [24]:

II.11.1 Motor Power:

- Determine the power consumption of the engines used in the robot. This information can usually be found in the engine datasheet or its specifications.
- Search for voltage requirements (usually in voltages) and current pull (usually in amps) for each engine.
- Calculate the power consumption of each motor by multiplying the voltage in the current. For example, if the engine is running at 12V and pulls 1A, the power consumption will be 12W.

II.11.2 Voltage and Current:

- Determine the voltage and current requirements for all the components used in the robot, including sensors, microcontrollers (such as Arduino boards), and communication modules.
- Ensure that the power supply can deliver the required voltage and current simultaneously to all components.

II.11.3 Battery Capacity:

- Estimate the runtime of the robot based on the battery capacity, which is typically measured in ampere-hours (Ah).
- Consider the expected usage of the robot, whether it will be continuously operated or used intermittently.

II.11.4 Battery Type:

- Choosing a suitable battery type based on the requirements of the robot.
- Lithium Polymer (LiPo) batteries are commonly used in mobile robots due to their high energy density.
- Nickel Metal Hydride (NiMH) batteries are suitable for lower voltage applications.
- Lead-Acid batteries are used in larger robots, such as Automated Guided Vehicles (AGVs), due to their capacity.

II.11.5 Voltage Regulators:

- Using voltage regulators, such as the LM7805, to provide a stable voltage for microcontrollers like Arduino.
- Regulate the voltage down to the required level, often 5V for Arduino boards.

II.11.6 Safety Measures:

- Implement safety measures to protect the components and ensure safe operation.
- Include overcurrent protection, short-circuit protection, and thermal protection mechanisms.
- Properly insulate and ground components to prevent electrical hazards.

II.11. Present the mechanical design of the robot:

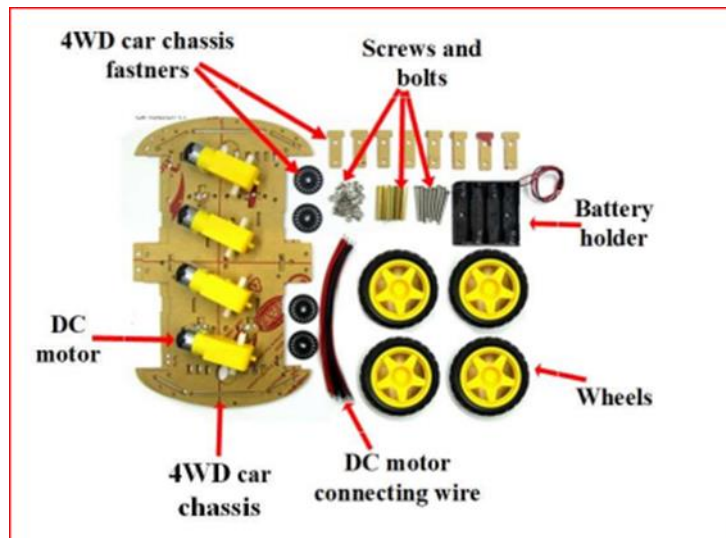


Figure II.11: Robot Chassis Components

Conclusion:

In summary, Arduino is a versatile platform that allows beginners and experts alike to create electronic projects. Whether you want to build a robot, a surveillance system, or an interactive art piece, Arduino offers a multitude of creative possibilities.

Chapter III: Hardware Implementation

III.1. Introduction:	27
III.2. Electrical diagram:	27
III.3. Assembly Implementation:	28
III.4 Synaptic diagram:	31

III.1. Introduction:

In this chapter, we will Building a versatile robot involves integrating various control methods to enhance its functionality and adaptability. In this project, we will construct a robot that can navigate obstacles, be controlled via Bluetooth, and respond to voice commands. This comprehensive approach will not only demonstrate the robot's capabilities in different scenarios but also provide a robust platform for further enhancements and applications. The journey begins with implementing obstacle avoidance, followed by adding Bluetooth control, and finally incorporating voice control to achieve a fully interactive and intelligent robotic system.

III.2. Electrical diagram:

The representation of the electrical schematic of our setup is as follows:

in figure III.1 we show electrical diagram.

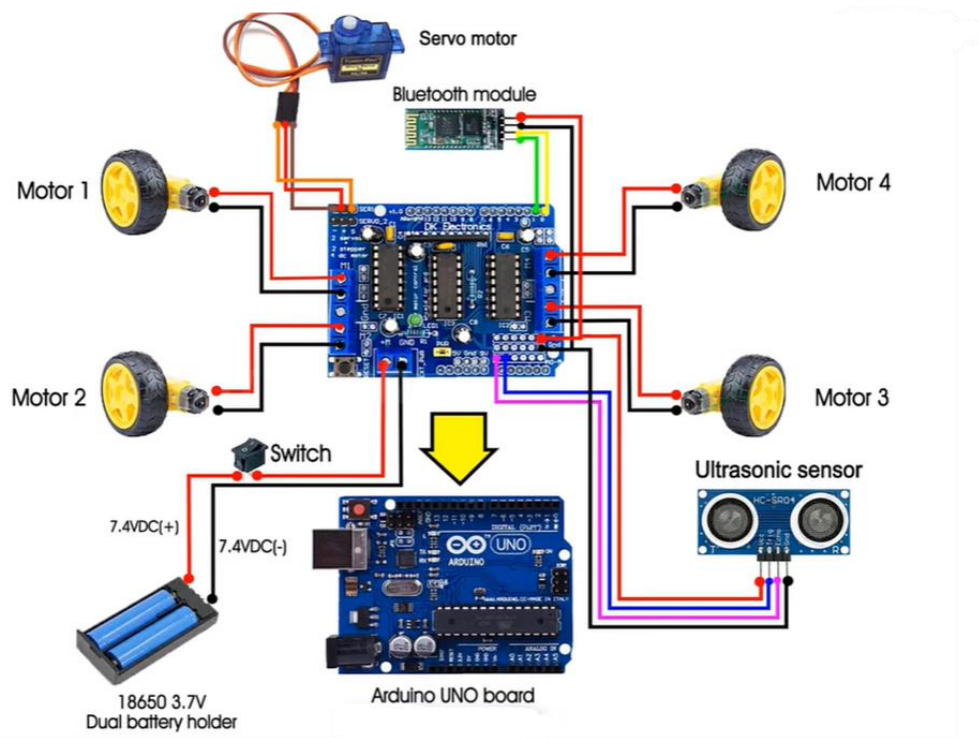


Figure III.1 : electrical diagram

III.3. Assembly Implementation:

This section is dedicated to the practical implementation of the assembly, programming of the ARDUINO module, and associated tests. It is divided into two parts: a software part and a hardware

III.3.1. Hardware part :**III.3.1.a Step 1 :**

In the first stage, you solder the wires to the four motors and then attach them to the chassis.

In figure III.2 ,III.3 we show attach for motor.

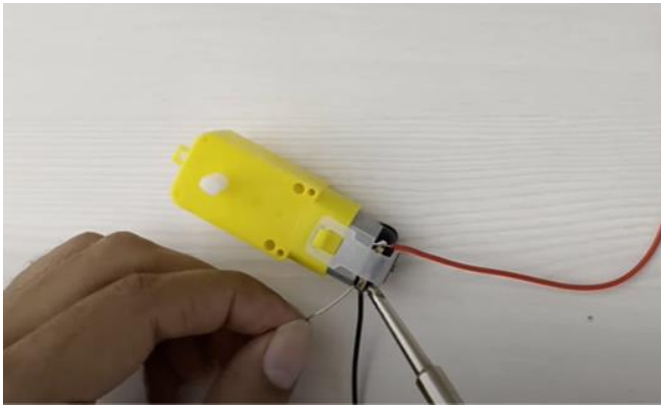


Figure III.2 : Welding the motor wires

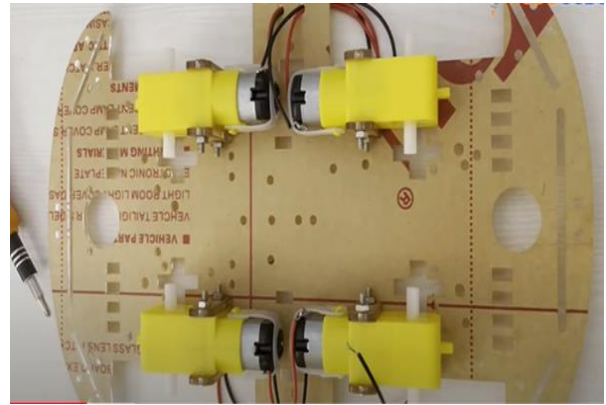


Figure III.3: Installing motors on the chassis

III.3.1.b Step 2 :

Then, attach the motor shield to the Arduino board and glue it to the robot chassis.

In figure III.4 we show Arduino board with L293d motor drive.

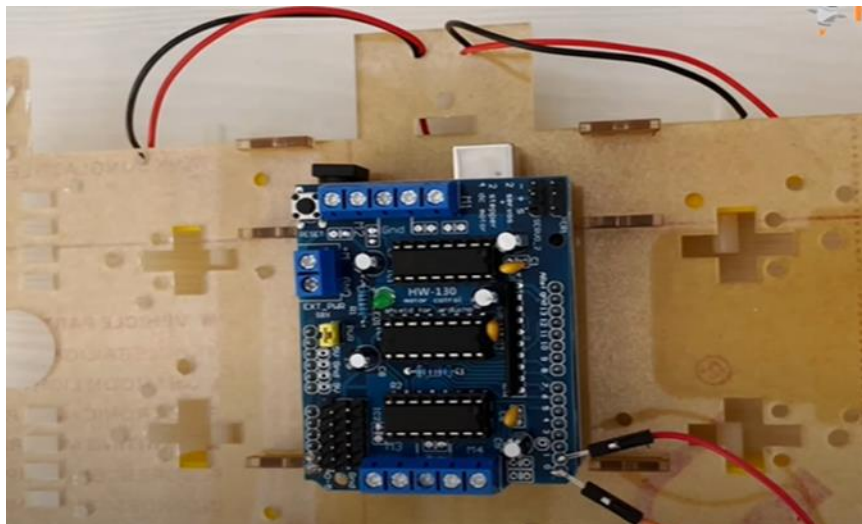


Figure III .4 :Arduino board with L293d motor drive

III.3.1.c Step 3 :

Next, connect the servo motor and the ultrasonic sensor using the circuit diagram above.

In figure III.5 we show connect servo motor and ultrasonic sensor.

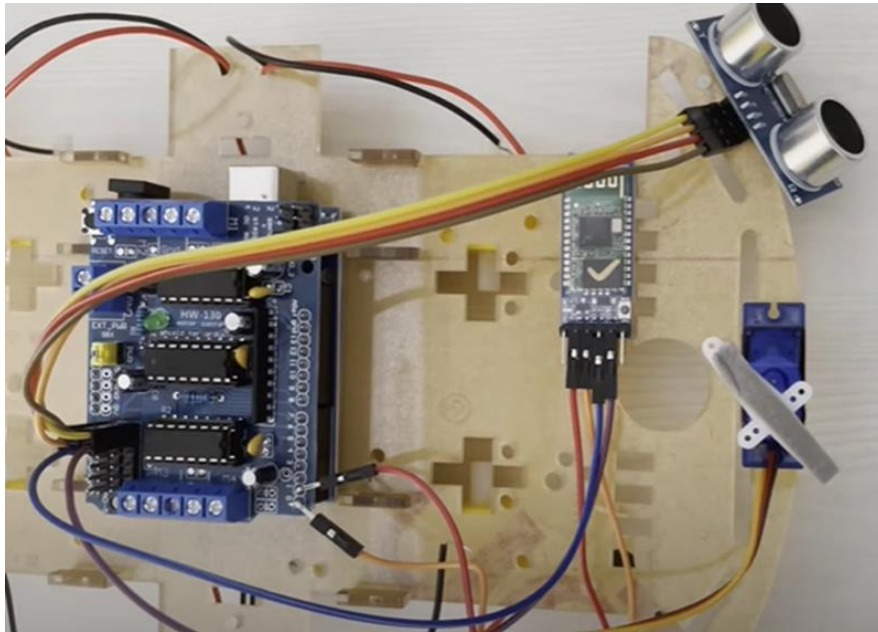


Figure III 5: connect servo motor and ultrasonic sensor.

III.3.1.d Step 4:

Then, connect the Bluetooth module to the motor driver shield and glue it to the robot chassis.

In figure III.6 we show connect the Bluetooth module.

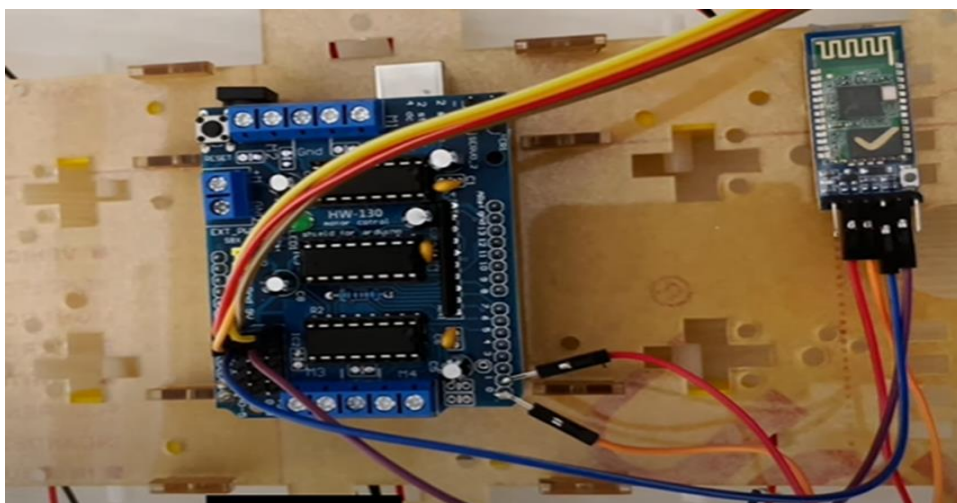


Figure III.6 : connect the Bluetooth module.

III.3.1.e Step 5:

Then, connect the motors to the motor driver shield. To do this, use the circuit diagram below.

In figure III.7 we show connecting motors

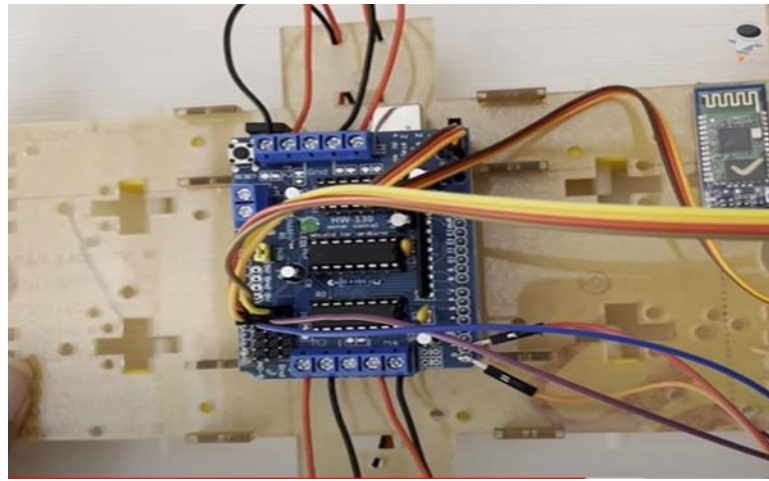


Figure III 7: connecting motors.

III.3.1.f Step 6:

Finally, attach the battery holder and connect it to the driver shield. Now, put the batteries in the battery holder.

In figure III.8 we show battery and battery holder.

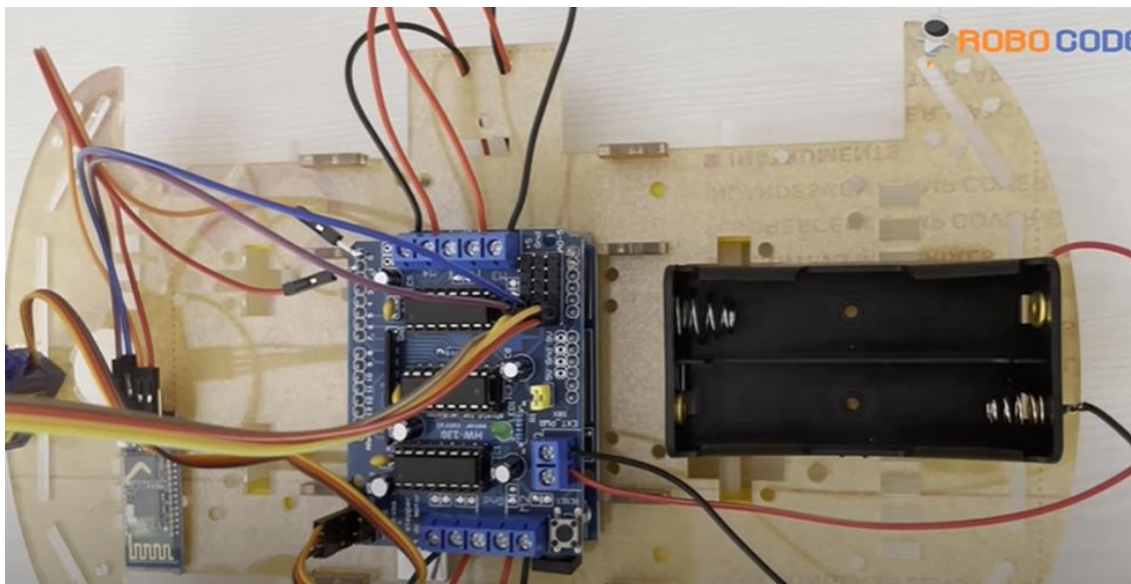


Figure III.8: battery and battery holder

III.4 Synoptic diagram:

III.4.1. Synoptic diagram of obstacle avoidance:

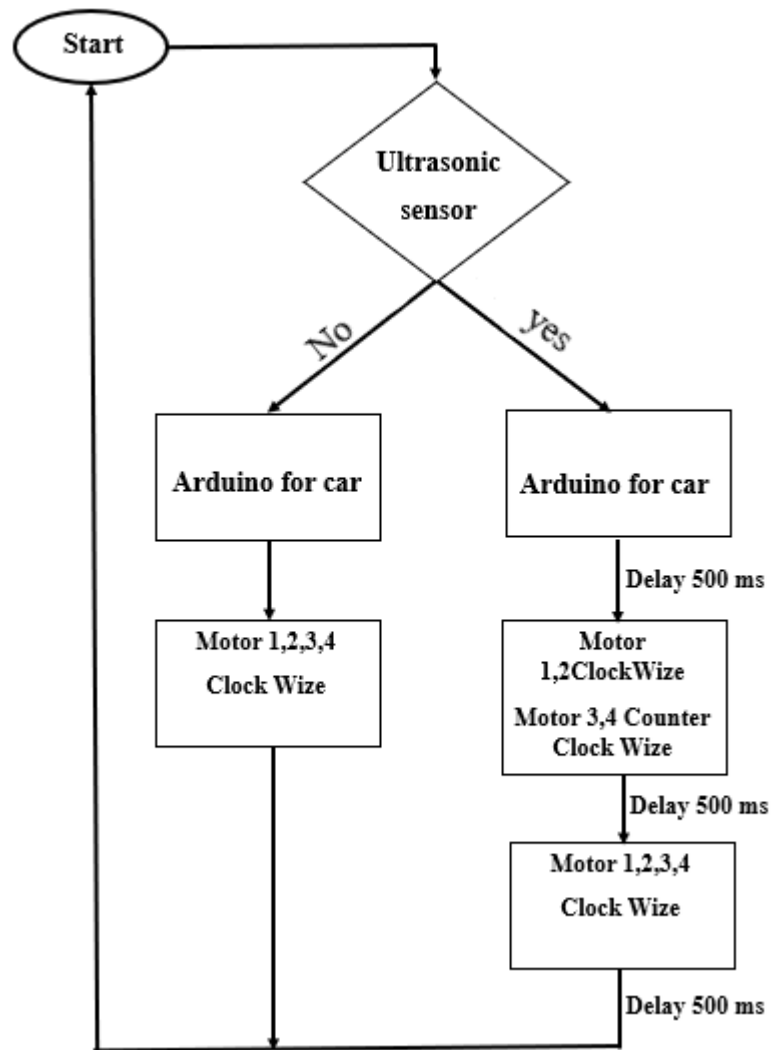


Figure III. 9: Synoptic diagram of obstacle avoidance

III.4.2 Synoptic diagram of Bluetooth control:

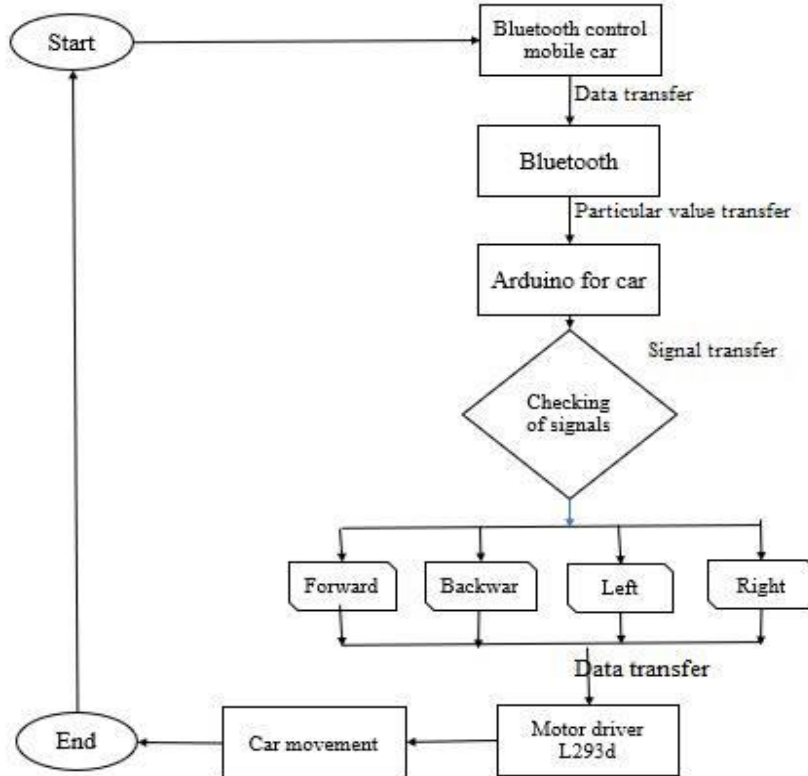


Figure III 10: Synoptic diagram of Bluetooth control

III.4.3 Synoptic diagram of voice control:

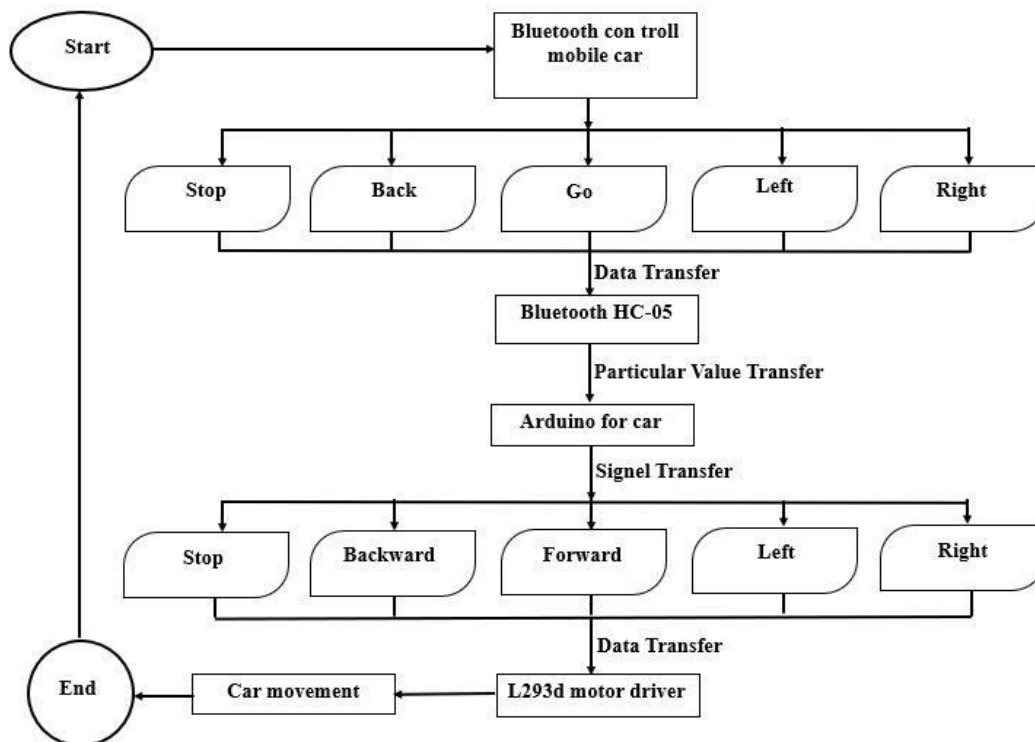


Figure III. 11: Synoptic diagram of voice control

Chapter IV: Software Implementation

IV.1 Program of project:.....34
IV.2. Implementation:.....40
Conclusion:.....48

IV.1 Program of project:

Let's create the program for this project, which includes all three functions. We can run these functions separately. The program is as follows.

```

/*obstacle avoiding, Bluetooth control, voice control robot car.
  https://srituhobby.com
*/
#include <Servo.h>
#include <AFMotor.h>
#define Echo A0
#define Trig A1
#define motor 10
#define Speed 170
#define spoint 103
char value;
int distance;
int Left;
int Right;
int L = 0;
int R = 0;
int L1 = 0;
int R1 = 0;
Servo servo;
AF_DCMotor M1(1);
AF_DCMotor M2(2);
AF_DCMotor M3(3);
AF_DCMotor M4(4);
void setup() {
  Serial.begin(9600);
  pinMode(Trig, OUTPUT);
  pinMode(Echo, INPUT);
  servo.attach(motor);
  M1.setSpeed(Speed);
  M2.setSpeed(Speed);
  M3.setSpeed(Speed);
  M4.setSpeed(Speed);
}
void loop() {
  Obstacle();
  //Bluetoothcontrol();
  //voicecontrol();
}
void Bluetoothcontrol() {
  if (Serial.available() > 0) {
    value = Serial.read();
    Serial.println(value);
  }
  if (value == 'F') {
    forward();
  }
}

```

```
} else if (value == 'B') {
  backward();
} else if (value == 'L') {
  left();
} else if (value == 'R') {
  right();
} else if (value == 'S') {
  Stop();
}
}
}
void Obstacle() {
  distance = ultrasonic();
  if (distance <= 12) {
    Stop();
    backward();
    delay(100);
    Stop();
    L = leftsee();
    servo.write(spoint);
    delay(800);
    R = rightsee();
    servo.write(spoint);
    if (L < R) {
      right();
      delay(500);
      Stop();
      delay(200);
    } else if (L > R) {
      left();
      delay(500);
      Stop();
      delay(200);
    }
  } else {
    forward();
  }
}
}
void voicecontrol() {
  if (Serial.available() > 0) {
    value = Serial.read();
    Serial.println(value);
    if (value == '^') {
      forward();
    } else if (value == '-') {
      backward();
    } else if (value == '<') {
      L = leftsee();
      servo.write(spoint);
      if (L >= 10 ) {
        left();
      }
    }
  }
}
```

```
        delay(500);
        Stop();
    } else if (L < 10) {
        Stop();
    }
} else if (value == '>') {
    R = rightsee();
    servo.write(spoint);
    if (R >= 10 ) {
        right();
        delay(500);
        Stop();
    } else if (R < 10) {
        Stop();
    }
} else if (value == '*') {
    Stop();
}
}
}
// Ultrasonic sensor distance reading function
int ultrasonic() {
    digitalWrite(Trig, LOW);
    delayMicroseconds(4);
    digitalWrite(Trig, HIGH);
    delayMicroseconds(10);
    digitalWrite(Trig, LOW);
    long t = pulseIn(Echo, HIGH);
    long cm = t / 29 / 2; //time convert distance
    return cm;
}
void forward() {
    M1.run(FORWARD);
    M2.run(FORWARD);
    M3.run(FORWARD);
    M4.run(FORWARD);
}
void backward() {
    M1.run(BACKWARD);
    M2.run(BACKWARD);
    M3.run(BACKWARD);
    M4.run(BACKWARD);
}
void right() {
    M1.run(BACKWARD);
    M2.run(BACKWARD);
    M3.run(FORWARD);
    M4.run(FORWARD);
}
void left() {
```

```
M1.run(FORWARD);
M2.run(FORWARD);
M3.run(BACKWARD);
M4.run(BACKWARD);
}
void Stop() {
  M1.run(RELEASE);
  M2.run(RELEASE);
  M3.run(RELEASE);
  M4.run(RELEASE);
}
int rightsee() {
  servo.write(20);
  delay(800);
  Left = ultrasonic();
  return Left;
}
int leftsee() {
  servo.write(180);
  delay(800);
  Right = ultrasonic();
  return Right;
}
```

IV.2. Implementation:

In this section, we will explain how to create an obstacle avoidance robot equipped with Bluetooth and voice control technology.

IV.2.1. Steps to operate an Obstacle avoidance robot:

IV.2.1.a Step 1 :

connect this robot car to the computer. Then, remove the two forward slashes in front of the “obstacle” function fig [IV.1]. Next, remove the RX and TX jumper wires connected to the Bluetooth module.

```

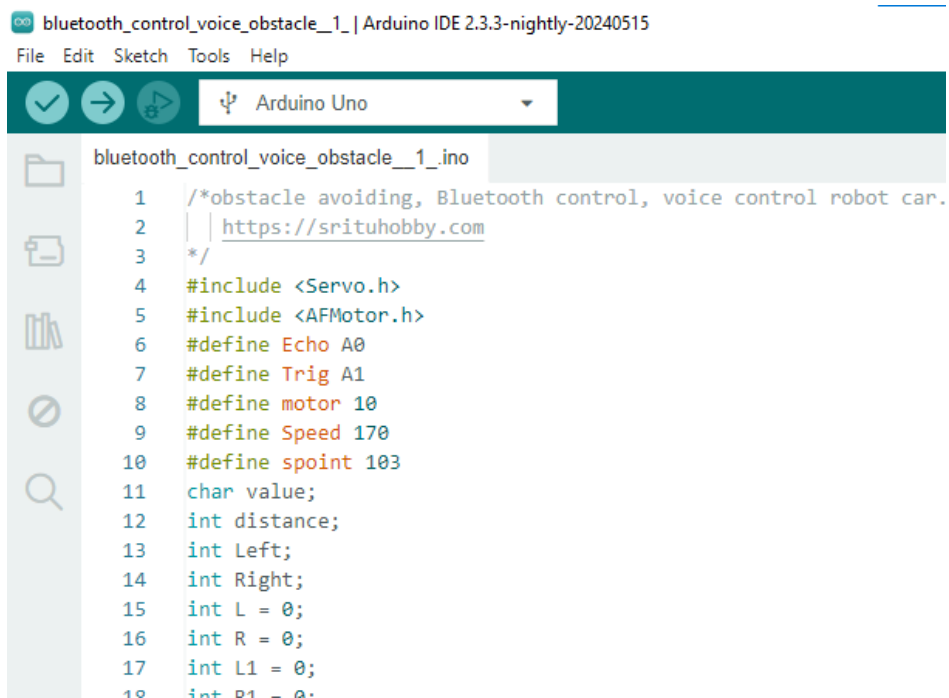
37   }
38   void loop() {
39       Obstacle();
40       //Bluetoothcontrol();
41       //voicecontrol();
42   }
43

```

Figure IV.1: modification of code for obstacle avoiding

IV.2.1.b Step 2:

Now, select board and port. After, upload this code to the robot and reconnect the RX and TX jumper wires.



```

bluetooth_control_voice_obstacle__1_ | Arduino IDE 2.3.3-nightly-20240515
File Edit Sketch Tools Help
Arduino Uno
bluetooth_control_voice_obstacle__1_ino
1  /*obstacle avoiding, Bluetooth control, voice control robot car.
2  | | https://srituhobby.com
3  */
4  #include <Servo.h>
5  #include <AFMotor.h>
6  #define Echo A0
7  #define Trig A1
8  #define motor 10
9  #define Speed 170
10 #define spoint 103
11 char value;
12 int distance;
13 int Left;
14 int Right;
15 int L = 0;
16 int R = 0;
17 int L1 = 0;
18 int R1 = 0;

```

*Figure IV.2: upload code***IV.2.1.c Step 3:**

Finally remove the connected computer cable, turn on the robot and watch how it crosses the obstacles.

IV.2.2. Steps to operate a Bluetooth control:**IV.2.2.a Step 1:**

connect this robot car to the computer. Then, remove the two forward slashes in front of the “Bluetooth control” function fig [IV.3]. Next, remove the RX and TX jumper wires connected to the Bluetooth module.

```

37   }
38   void loop() {
39       //Obstacle();
40       Bluetoothcontrol();
41       //voicecontrol();
42   }
43

```

*Figure IV.3: modification of code for Bluetooth control.***IV.2.2.b Step 2:**

select board and port. After, upload this code to the robot and reconnect the RX and TX jumper wires. Download and install the app below [Arduino Bluetooth RC car]. Then, follow the steps below.

IV.2.2.c Step 3:*Figure IV.4:app*

run this application and click the Settings button. Then, click the “Connect to Car” button and select the name of the Bluetooth module. Now, you can see the green bulb in the corner.

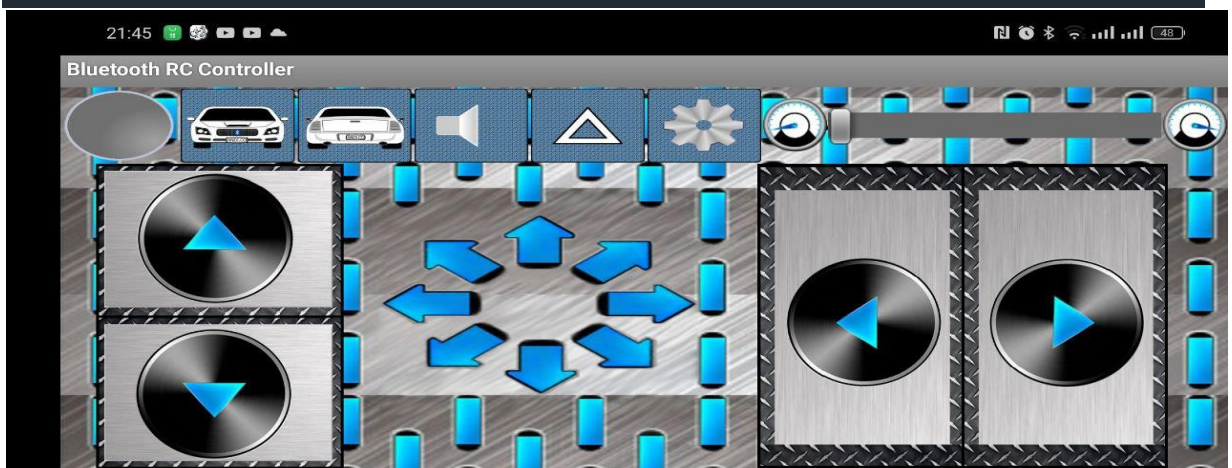


Figure IV.5

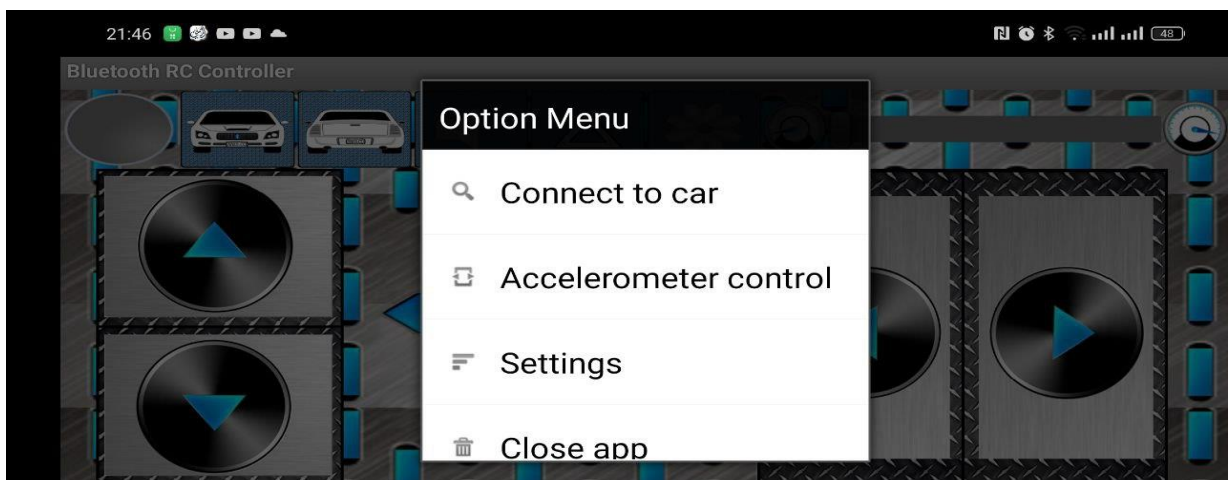


Figure IV.6

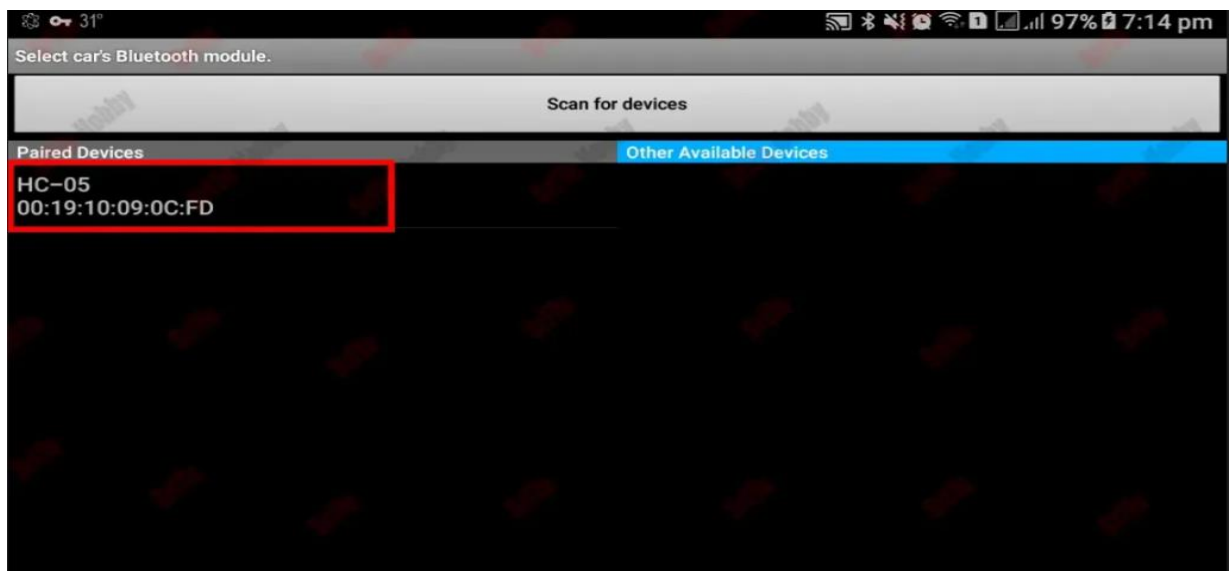


Figure IV.7



Figure IV.8

IV.2.2.d Step 4:

Click the controller buttons and move the robot.

IV.2.3 Steps to operate a voice control:**IV.2.3.a Step 1:**

connect this robot car to the computer. Then, remove the two forward slashes in front of the “voice control” function fig (IV.9). Next, remove the RX and TX jumper wires connected to the Bluetooth module.

```

37   }
38   void loop() {
39     //Obstacle();
40     //Bluetoothcontrol();
41     voicecontrol();
42   }
43

```

Figure IV.9: modification of code for voice control**IV.2.3.b Step 2:**

download and install the app below [Arduino Bluecontrol].

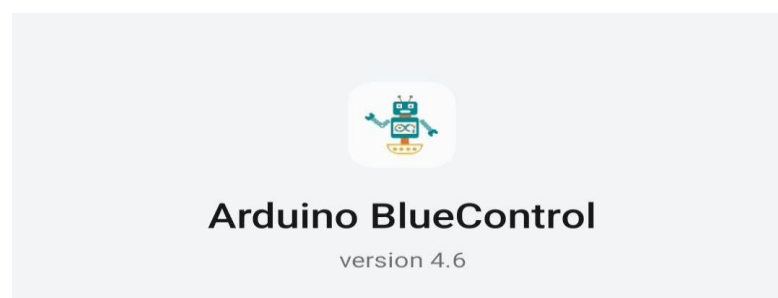


Figure IV.10 : app

Then, run this application and click the setting button. Then, click the “voice commands configuration” button and include the commands one by one. These are as follows.

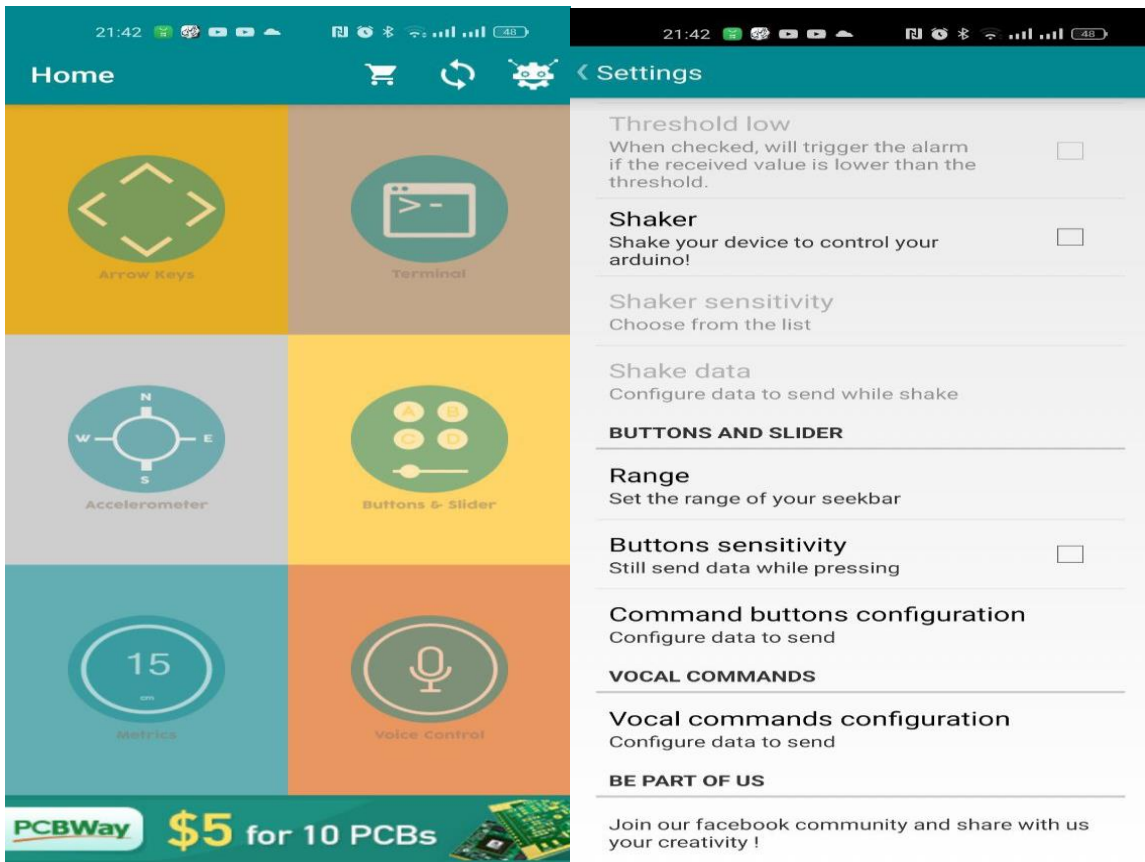


Figure IV.11

figure IV.12

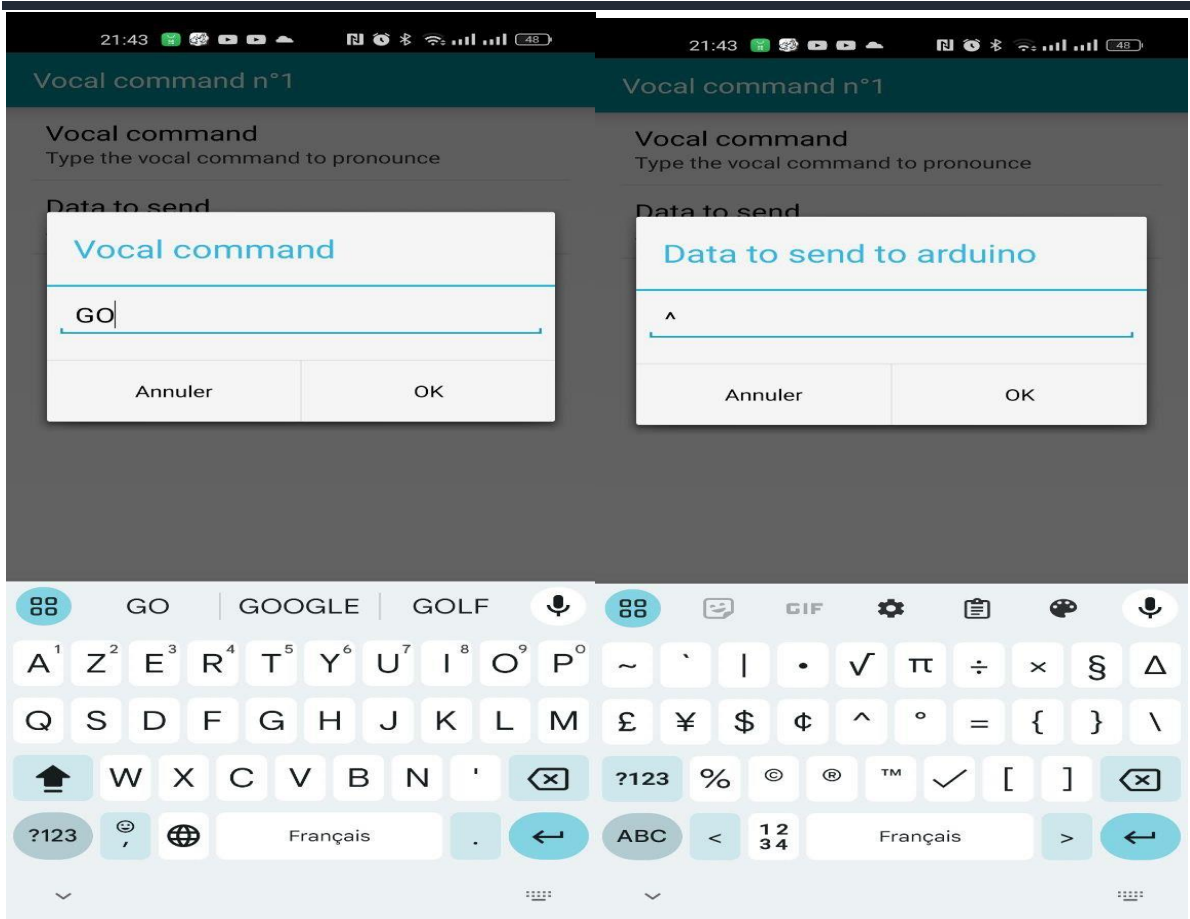


Figure IV.13

figure IV.14

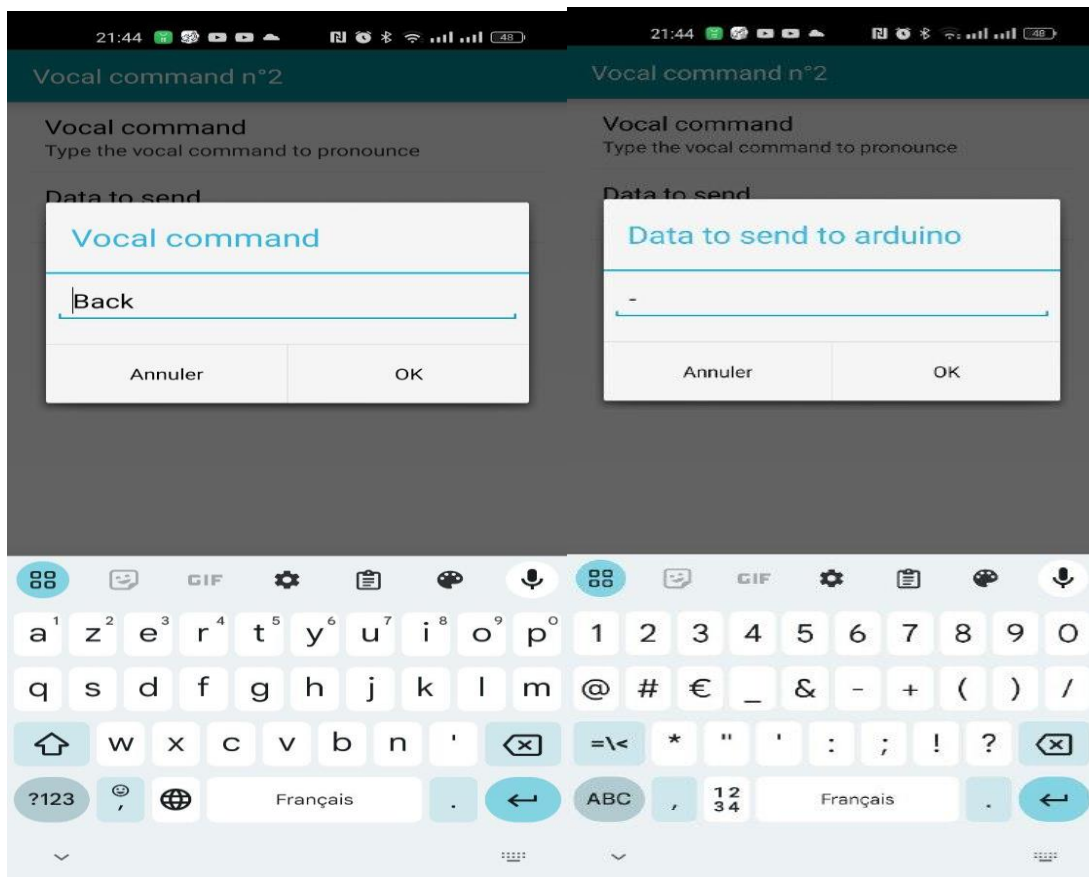


Figure IV. 15

figure IV.16

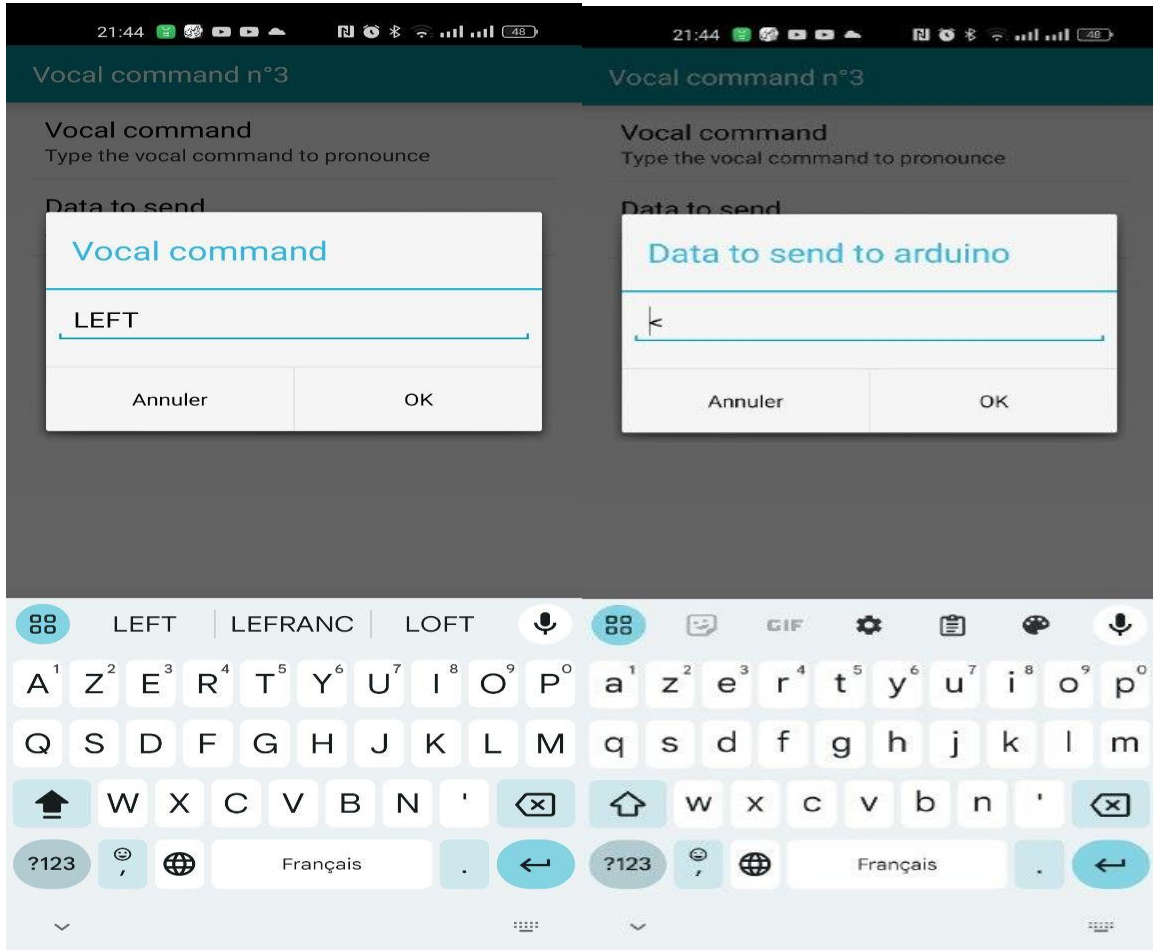


Figure IV.17

figure IV.18

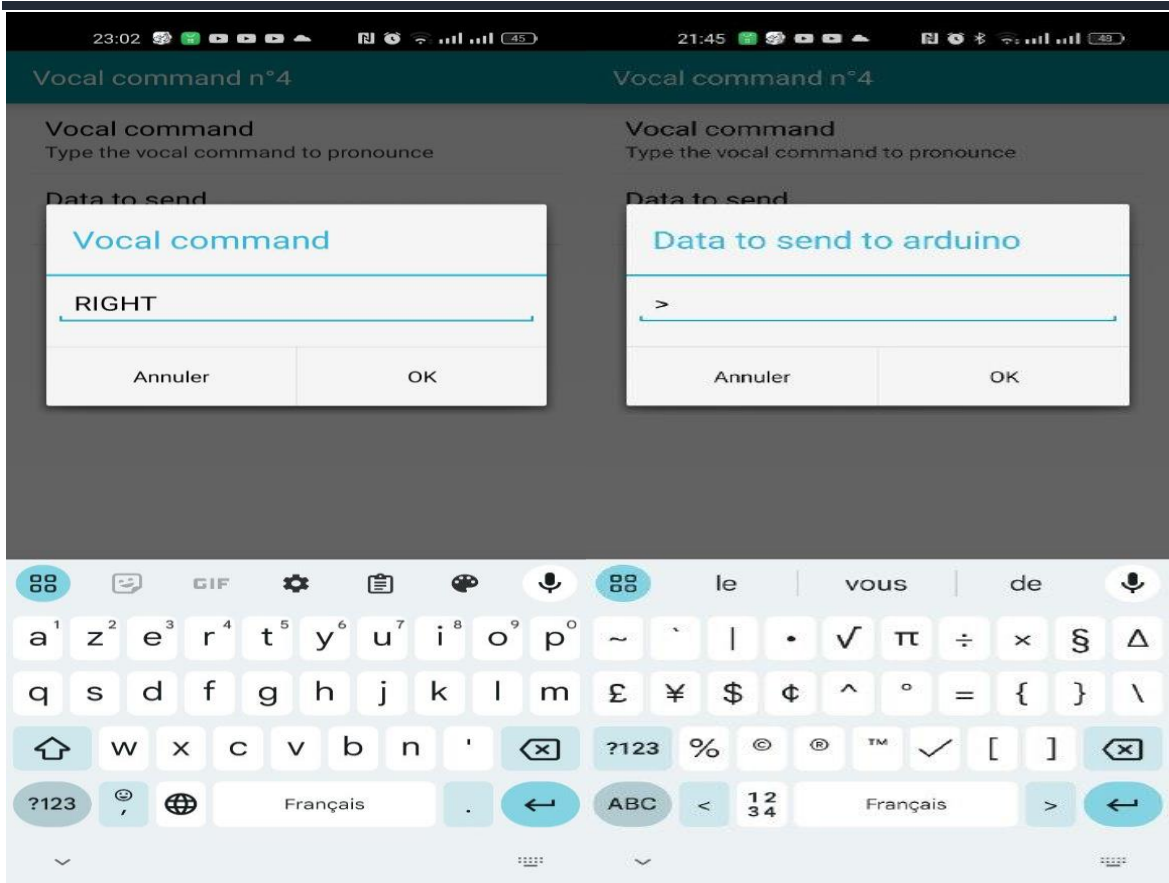


Figure IV.19

figure IV.20

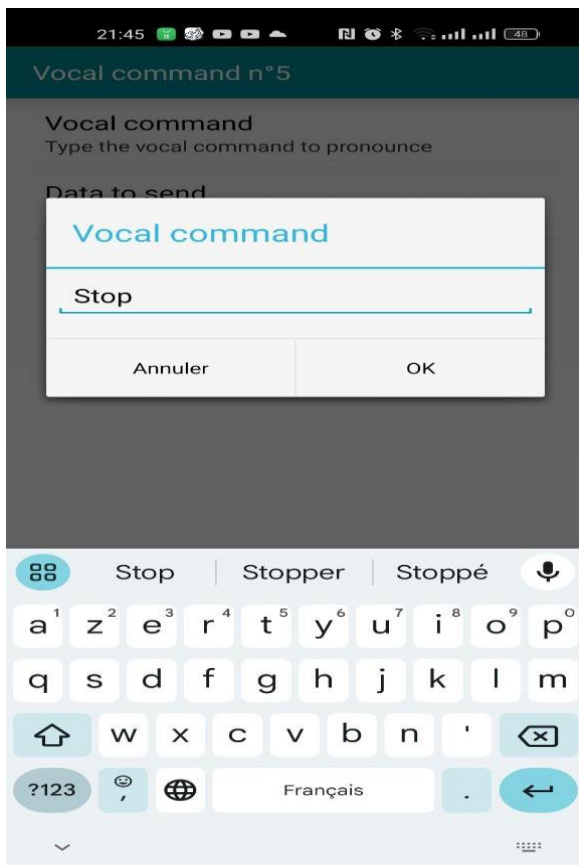


Figure IV.21



Figure IV.22

IV.4.3.c Step 3 :

now click the voice control button and enjoy this project.

Conclusion

In the end, we achieved our goal of developing an Arduino-powered robot capable of autonomously avoiding obstacles, and controlled via a mobile phone through Bluetooth as well as voice commands. This simple project enables the development of robots that can make life easier, such as self-driving cars and delivery robots. It can also be used in warehouses for goods, allowing for the organization and transportation of items remotely in the least possible time and with minimal effort.

General conclusion:

In this project, we focused on the development of an Arduino-powered robot capable of obstacle avoidance and controlled via Bluetooth and voice commands. By integrating hardware components such as ultrasonic sensors, an Arduino board, and a Bluetooth module, along with the development of algorithms for obstacle detection and communication protocols, we successfully created a versatile robot that offers enhanced control and safety in various real-world scenarios.

The robot utilizes ultrasonic sensors to detect obstacles in its path and autonomously navigates around them, ensuring efficient obstacle avoidance. Additionally, it can be remotely controlled through a mobile device connected via Bluetooth, providing flexibility and convenience to the user. The integration of voice commands further enhances the usability of the robot, allowing for intuitive and hands-free control.

After this study in the field of intelligent robots, it became clear to us that these primitive models can be developed better and can be exploited in many aspects of human life. Which made the idea of developing this type of robots

Inspiring for us, it requires us to deepen research and work in the future so that these robots become able to overcome fixed and even moving obstacles, receive instructions in multiple ways, and perform broader tasks efficiently and effectively while gaining time to serve humanity.

References:

- [1] Kondaveeti, Hari Kishan, et al. "A systematic literature review on prototyping with Arduino: Applications, challenges, advantages, and limitations." *Computer Science Review* 40 (2021): 100364.
- [2] Oellermann, Michael, et al. "Open hardware in science: The benefits of open electronics." *Integrative and comparative biology* 62.4 (2022): 1061-1075. Ozer, Jonathan, and Hugh Blemings. *Practical Arduino: cool projects for open-source hardware*. Apress, 2011.
- [3] Pajankar, Ashwin. *Arduino Made Simple: With Interactive Projects*. BPB Publications, 2018. Margolis, Michael, Brian Jepsen, and Nicholas Robert Weldin. *Arduino cookbook: recipes to begin, expand, and enhance your projects*. O'Reilly Media, 2020.
- [4] Singh, Dhawan, et al. "An overview of IoT hardware development platforms." *Int. J. Emerg. Technol* 11.5 (2020): 155-163.
- [5] Hughes, John M. *Arduino: a technical reference: a handbook for technicians, engineers, and makers*. "O'Reilly Media, Inc.", 2016.
- [6] Kondaveeti, Hari Kishan, et al. "A systematic literature review on prototyping with Arduino: Applications, challenges, advantages, and limitations." *Computer Science Review* 40 (2021): 100364.
- [7] Badamasi, Yusuf Abdullahi. "The working principle of an Arduino." 2014 11th international conference on electronics, computer and computation (ICECCO). IEEE, 2014.
- [8] Nayyar, Anand, and Vikram Puri. "A review of Arduino boards, Lilypad's & Arduino shields." 2016 3rd international conference on computing for sustainable global development (INDIACom). IEEE, 2016.
- [9] Galadima, Ahmad Adamu. "Arduino as a learning tool." 2014 11th International Conference on Electronics, Computer and Computation (ICECCO). IEEE, 2014.
- [10] Blum, Jeremy. *Exploring Arduino: tools and techniques for engineering wizardry*. John Wiley & Sons, 2019.

References

- [11] Oxer, Jonathan, and Hugh Blemings. *Practical Arduino: cool projects for open-source hardware*. Apress, 2011.
- [12] Cordova-Huaman, Allison V., Victor R. Jauja-Ccana, and Adolfo La Rosa-Toro. "Low-cost smartphone-controlled potentiostat based on Arduino for teaching electrochemistry fundamentals and applications." *Heliyon* 7.2 (2021).
- [13] FARDOUS, MD, et al. *Smart Phone Control Peck and Place Robotics Arm With Robot Car*. Diss. Sonargaon University (SU), 2022.
- [14] Islam, Riazul. *Design and Development of a Prosthetic Hand*. Diss. Department of Technical and Vocational Education (TVE), Islamic University of Technology (IUT), 2014.
- [15] rover, Radhika, et al. "A competition-based approach for undergraduate mechatronics education using the arduino platform." *Fourth interdisciplinary engineering design education conference*. IEEE, 2014
- [16] Mbarek, A., K. Ben Saad, and M. Benrejeb. "A Position Sensorless Control for a Linear Stepper Motor Based on a Sliding Mode Observer." *International Review on Modelling and Simulations (IREMOS)* 4.3 (2011): 1131-1140
- [17] Koutroulis, Eftichios, Kostas Kalaitzakis, and Nicholas C. Voulgaris. "Development of a microcontroller-based, photovoltaic maximum power point tracking control system." *IEEE Transactions on power electronics* 16.1 (2001): 46-54
- [18] yhan, Mustafa Batuhan. "A fuzzy AHP approach for supplier selection problem: A case study in a Gear motor company." *arXiv preprint arXiv:1311.2886* (2013)
- [19] Faisal, SM Saifur Rahman, et al. "Design and development of an autonomous floodgate using arduino uno and motor driver controller." *2017 4th International Conference on Advances in Electrical Engineering (ICAEE)*. IEEE, 2017
- [20] Carullo, Alessio, and Marco Parvis. "An ultrasonic sensor for distance measurement in automotive applications." *IEEE Sensors journal* 1.2 (2001): 143.
- [21] Wirawan, Nanda Tommy. "Pemanfaatan Smartphone Pada Robot Beroda Untuk Monitoring Jarak Robot Dengan Halangan Menggunakan Bluetooth Hc-05 Sebagai Media Komunikasi." *Jurnal Komtekinfo* 5.1 (2018).

References

- [22] Warren, John-David, Josh Adams, and Harald Molle. "Arduino for robotics." Arduino robotics. Berkeley, CA: Apress, 2011. 51-82.