Bluetooth controlled smart car with obstacle avoiding capabilities

Taradaciuc Nicolae
Department of Automatic Contol and Applied Informatics
Faculty of Automatic Control and Computer Engineering
Iași, România.
nicolae.taradaciuc@student.tuiasi.ro

***Abstract* - This project presents the design and implementation of a versatile smart car system utilizing Arduino technology for control and interaction. The car is equipped with Bluetooth communication capabilities, enabling seamless operation via a PlayStation 5 (PS5) controller. The integration of the PS5 controller provides an intuitive and user-friendly interface, enhancing the overall experience.**

The smart car features an obstacle avoidance mode, in which the car autonomously navigates its environment, avoiding obstacles in real-time. This functionality is achieved through a combination of an ultrasonic sensor and an intelligent decision-making algorithm implemented on the Arduino board.

The ATmega328P microcontroller facilitates seamless communication between the PS5 controller, USB Host Shield and the car's motor control system.

Keywords— ATmega328P, Bluetooth, Arduino, Smart Car, PS5 Controller, USB Host Shield, Obstacle Avoidance

I. Introduction

In recent years, the intersection of embedded systems and gaming technology has given rise to innovative and engaging projects that captivate both enthusiasts and learners alike. This project explores the integration of a Bluetooth-controlled smart car, driven by an ATmega328P microcontroller, with a focus on user-friendly interaction through a PlayStation 5 (PS5) controller. The unique aspect of this project is the intermediary role played by a USB Host Shield, connecting the wireless PS5 controller to the ATmega328P microcontroller.

In the domain of connectivity, Bluetooth is a key factor. It functions as a wireless communication technology that enables devices to exchange data over short distances. In this project, the wireless link that connects the PS5 Controller and the USB Host Shield is completely done through a Bluetooth connection. For a visual representation, refer to Fig. 1, showcasing the smart car, its components, and the PS5 controller.

The smart car presented goes beyond conventional remote-controlled vehicles by incorporating an obstacle avoidance system. This system, uses an ultrasonic sensor and a set of intelligent algorithms, enabling the car to autonomously navigate its surroundings while dynamically evading obstacles in real-time.

The project is designed to provide a user-friendly experience. The smart car is a small recreation of a real car. The analog acceleration, responding to the degree of trigger pressure, simulates the control everyone will expect in a real vehicle. Similarly, the steering mechanism, executed through the left stick of the controller, allows users to decide the degree of turn, offering a lifelike simulation of steering. This project not only serves as an entertaining and educational purpose, but an actual small-scale introduction to the principles of a real car.

Fig. 1 Bluetooth controlled smart car with obstacle avoiding capabilities.

Going further, a few subjects like the "Overview of Components", "Hardware Setup", "Software Architecture" will provide a detailed exploration of the whole project, explaining the integration of various components and the whole process dedicated to the hardware and software part of the project. The obstacle avoidance system will be detailed, followed by a small presentation of the challenges encountered during development with the solution that overcame them. Additionally, a few planned ideas for the future will showcase the true nature and the whole perspective of the project itself, providing an overall understanding of the project, marking the important steps that were made during the development.

II. Overview of Components

A. Chassis (3D – Printed)

The 3D – Printed chassis provides a customized and a structure to attach all the components to. Four bearings were used to ensure smooth movement for the servo direction and wheels. The chassis was selected for its simplicity in assembly, offering a customized structure that functions as a platform [4].

B. *Arduino Uno (ATmega328P)*

The Arduino Uno contains the ATmega328P microcontroller which manages communication between components and rules the overall operation of the smart car, using 5V. The board is referred to Fig. 2 for a visual representation. It serves as the central brain of the car and by integrating it with C++ programming, resulted in much more flexibility and control over the whole project [1]. The Arduino Uno was chosen for this project because it could provide sufficient power for the project and because it is very compatible, having a wide range of shields and modules that could be used for this project. Additionally, its user-friendly and extensive documentation makes it accessible for beginners and experienced developers.

Fig. 2 Arduino Uno

C. *DC Motors (3-6V) and L289N Motor Driver*

The smart car's movement is powered by two DC motors and their speed is precisely controlled by the L298N motor driver, both being displayed in Fig. 3 for a visual representation. The L289N module with a dual H-bridge allows for speed control through PWM signals and the direction of rotation of the two DC motors [7].

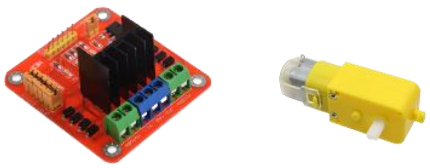

Fig. 3 L289N Motor Driver and DC Motor 3-6V

D. *Servo Motor MG90S*

The MG90S servo motor, displayed in Fig. 4, is capable of rotating up to 120 degrees and is controlled by a PWM signal [6]. Its precision and speed make it ideal for steering, making sure that the car would be able to execute precise, fast, and accurate movements, contributing to the overall responsiveness of the vehicle.

Fig. 4 Servo Motor MG90S

E. *USB Host Shield and USB Bluetooth Receiver*

This USB Host Shield (Fig. 5) provides USB Host interface, allowing full communication with USB devices such as keyboards, mice, joysticks, Bluetooth, digital camera and many other [2][3]. In this project, the USB Host Shield is used because it facilitates the communication between the car and the PS5 controller (Fig. 6), using the USB Bluetooth Receiver (Fig. 5) as a link.

The Micro Adapter Bluetooth USB receiver (Fig. 5) establishes a wireless link between the smart car and the PlayStation 5 controller. It serves as an intermediary connection point between the PS5 controller and the USB Host Shield. This connection allows access to the HID (Human Interface Device) interface.

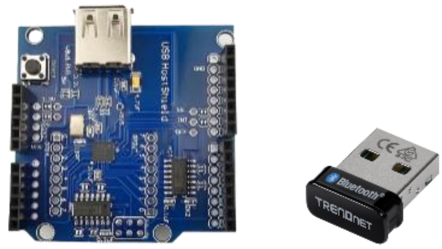

Fig. 5 USB Host Shield and USB Bluetooth Receiver

F. *PlayStation 5 Controller*

The PS5 Controller, displayed in Fig. 6, also known as DualShock, is a line of gamepads developed by Sony Interactive Entertainment for the PlayStation video game consoles. In this project, the PS5 Controller is used because it is an very intuitive component, having two adaptive triggers, a two-point touchpad, a click mechanism, two sticks, sixteen buttons and a six-axis motion sensing system with a three-axis gyroscope and a three-axis accelerometer. These features make the PS5 controller an ideal choice for the current requirements of the project, offering a versatile interface for controlling the smart and activating the avoidance system. Moreover, the vast amount of buttons and inputs offer easier future developments, without having the need to change the controller.

Fig. 6 PS5 Controller

G. *Gravity IO Expansion Shield*

The Gravity IO Expansion Shield expands the connectivity options, providing additional input and output ports. The Shield is referred to Fig. 7 for a visual representation. It also includes ports such as I2C, an SD Card interface and a Bluetooth Module Interface.

Fig. 7 Gravity IO Expansion Shield

H. Ultrasonic Sensor HC – SR04

The HC-SR04 ultrasonic sensor that can be seen in Fig. 8 acts as the eyes of the smart car's obstacle avoidance system. It measures distances, enabling the car to detect and avoid obstacles. This sensor sends ultrasonic waves to measure distances accurately, allowing the car to understand its surroundings in real time [8].

Fig. 8 Ultrasonic Sensor HC – SR04

I. Switches (3), Battery Holders (3) and LEDs (2)

Three switches have the functions to cut off the power that goes to the Servo Motor, DC Motors or Arduino Board. The inclusion of these switches serves a practical purpose, making it significantly more convenient to disable power to a specific component than removing or disconnecting the power source after each use or test of the smart car.

The three battery holders are placed under the smart car, connected to the car's chassis. The positioning of the three battery holders beneath the smart car serves a dual purpose. This placement helps with the overall design of the car, but mostly with the ease of use. Keeping the batteries securely attached to the car simplifies the process of swapping batteries when needed.

The two LEDs function as headlights, simulating a real car's illumination. More similar functionalities will be presented it the project's future development section.

III. HARDWARE SETUP

A. The Creation of the Chassis

This design choice was intentional because it provided a designated space on top of the car that serves as an attachment point for various components, allowing for easier customization [4]. For a visual representation, please refer to Fig. 9, showcasing the 3D design of a few parts of the smart car's chassis.

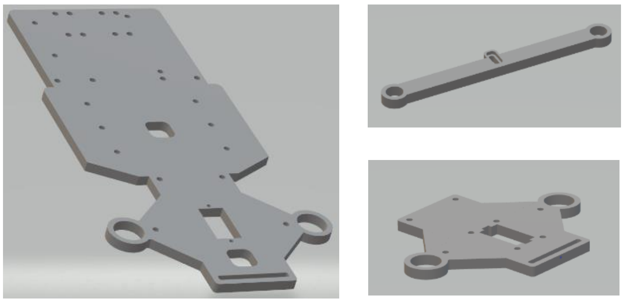

Fig. 9 The 3D Printing of the Chassis

B. The Assembly Process

Screws, nuts, bolts and 3D – printed spacers were used to securely mount the components to the chassis. After the printing of the chassis, the bearings were securely glued to their designated spots.

The assembly process started with the mounting of motors under the chassis. The Arduino Uno will be placed on the top section of the chassis, on top of the motors, raised by four 3D-printed spacers with the USB Host Shield and the Gravity IO Expansion Shield that would be stacked on top of it. The process is visually represented by Fig. 10.

Under the car's middle section, the three battery holders would be securely fixed to the chassis. Similarly, in the middle section of the car, but on the top of the chassis, the L298N motor Driver would be found.

Towards the front of the car, the steering system and servo motor would be installed beneath, while the ultrasonic sensor is positioned on top, on the very end of the car.

This arrangement was chosen to help easy modification of different pins and connections. The design allows access to any adjustment regarding the change, without the need to remove any of the parts.

All three switches have been securely fixed to the chassis of the car, using a hot glue gun. Each switch serves as a Power Control Switch. They are connected to the power supplies of the Arduino Board, the Servo Motor and the DC Motors.

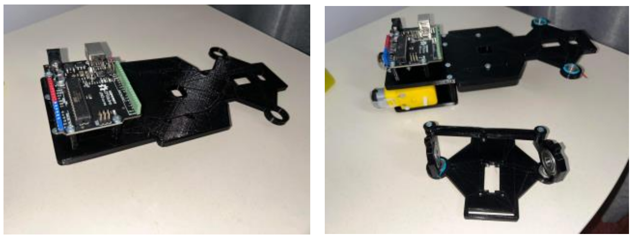

Fig. 10 The Process of mounting the smart car's components on the chassis

IV. SOFTWARE ARHITECTURE

The code for the smart car project was carefully designed with a modular approach, utilizing functions to enhance clarity and maintainability. Each hardware component, from the ultrasonic sensor to the DC motos and servo motor, has a dedicated set of functions. The smart car's code was created to be clear and easy to work with. This makes the code easier to understand and troubleshoot, if needed.

A. Motor Control and L298N Module Integration

The module L298N was used to help the Arduino send signals to the motors and control them, being able to speed them up or slow them down.

The precise control of the speed was made using the PWM (Pulse Width Modulation) that was a part from the Timer 0, allowing the motors to respond quickly and smoothly to any command.

This part contains only two functions, one for moving forward and one for moving in reverse. Each function had a condition to make sure the car wouldn't move if it did not receive any commands. The reason for creating this condition is to ensure that the car remains still until it gets a clear instruction.

B. Servo Motor Integration

The Servo motor was controlled using the PWM (Pulse Width Modulation) from Timer 1. The control was organized into three functions: one for initialization, another for modifying the PWM value and the third for updating the servo direction.

To make sure that everything operates smoothly, a condition was implemented. In the absence of commands, the wheels, along with the servo motor, will automatically return to a straight position. This condition will allow the car to stay aligned when it will not receive any specific instructions, making the use of the car more user-friendly.

C. Ultrasonic Sensor Integration

The Ultrasonic Sensor was used to measure the distance between the car and the object in front of it. Pin 10 was designated for the trigger and the pin 8 for the echo. The control was made using two functions: one for initialization and another for measuring distance. The following formula was applied, having the role of converting the time of impulse into a precise distance:

$$Distance = (Time\ of\ Impulse * 0.034) / 2 \qquad (1)$$

The sound travels are approximately 340 meters per second and the “2“ is in the formula because the sound has to travel back and forth.

D. Bluetooth Communication

For the Bluetooth Connection, the USB Host Shield was integrated with the Arduino board, by being placed on top of it. The shield utilizes the pins 10, 11, 12 and 13. The USB Receiver will be connected to the USB Host Shield and linked through a pairing process with the PS5 Controller.

A dedicated function was implemented to initiate the pairing process, by activating the USB receiver. After connecting the PS5 Controller, the pairing process will be completed. If the pairing is successful, a flag will be generated, and it will allow to activate a condition to continuous extraction of controller’s movements information. This real-time data will be utilized to control the overall movement of the smart car.

E. LED Control

The Smart Car features two LEDs that represent a set of headlights. They are activated using the UP button on the PS5 controller, offering the ability to turn them off or on at any time, providing a much better user experience and complete control over the lightning system of the car. The negative and positive cathodes of the two LEDs were combined and connected to ground and to pin 5 in port C of the board.

F. Safety Features

As previously mentioned, the servo motor automatically returns to a straight position when no commands are received and the car will always remain stationary, preventing any unintended movements.

Furthermore, a dedicated function has been implemented for an emergency stop. This function halts any type of control and command from the PS5 controller, resulting in a complete shutdown of the smart car.

G. User Interface and Interaction

The control of the car is done through the PlayStation 5 controller, a very intuitive interface, using the following buttons for navigation and various functionalities of the project:

PS Button = Connect / Disconnect the Controller
Right Trigger = Move the car forward
Left Trigger = Move the car backward
X-axis of the Left Stick = Steering the car
UP Button = Headlights ON / OFF
Triangle Button = “Auto“ Mode ON
Circle Button = “Auto” Mode OFF

V. OBSTACLE AVOIDANCE SYSTEM

The system was designed with a set of rules and conditions, making sure that it’s adaptable and easy to modify when needed. This approach makes the system easier to improve over time and allows for small modifications and additions.

The decision-making process for the Obstacle Avoidance System is created to follow the decision graph described in Fig. 11.

The smart car obstacle avoidance system can be activated and deactivated by pressing the button O, on the PS5 controller.

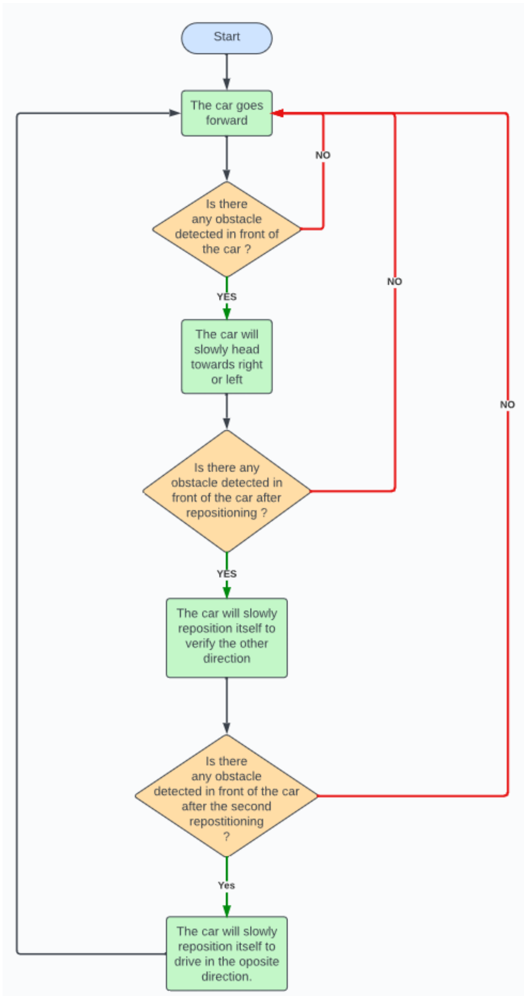

Fig. 11 Decision Graph of the Obstacle Avoidance System

This decision graph represents the final form of the algorithm, where, once the system is activated, starting from the initial position, the car will slowly start to move forward and utilizing the ultrasonic sensor, it will anticipate the presence of an obstacle in the distance, making a random decision to head towards right or left. If another obstacle is encountered in the chosen direction, the car will reposition itself and attempt to move in the opposite direction. In case in both directions have been detected obstacles, the car will turn around.

Concerning the challenged faced, a detailed overview of the obstacle avoidance system's challenges and corresponding solutions is presented in Table 1. It describes a few challenges encountered throughout the project, offering a closer look at the issues and the innovative solutions found. For and in-depth understanding of the problems faced, Table 1 serves as a valuable reference point, presenting the essence of the solved problems

Table 1. Challenges and Solutions

USB Receiver Compatibility	**Challenge:** Encounter compatibility issues with some USB receivers and the PS5 controller. Additionally, faced a problem where the receiver was unable to remember the address of the PS5 controller, resulting in a re-pairing before every use of the car. **Solution:** Tested and selected compatible USB receivers to ensure seamless connectivity with the PS5 controller.
USB Host Shield Manufacturing Issue	**Challenge:** Discovered a manufacturing problem where the USB Slot was not able to receive 5V [5]. **Solution:** Identified the manufacturing issue and re- soldered the board to make sure that the USB Slot will function properly.
Obstacle Avoidance System Limitation	**Challenge:** The obstacle avoidance system is not yet complete, currently only being able to detect and navigate around obstacles avoiding making direct contact with the detected object. The current state of the car's avoidance system capabilities is represented by Fig. 12. **Solution:** Under development, a solution is still being explored to complete and address all cases, making sure that the car can operate autonomously and independently until recharging.
Servo Motor Accuracy and Resetting	**Challenge:** Faces accuracy and resetting issues with the servo motor even code troubleshooting. **Solution:** After various tests, adding a dedicated power supply for the servo motor solved the accuracy and resetting issues.
DC Motor Power Supply	**Challenge:** Initially, the DC motors were provided with 5V from the Arduino, resulting in a not so optimal speed. **Solution:** Recognized the need for a dedicated power supply for the DC motors to gain more power and speed.
5G Control Development	**Challenge:** Initiating development for 5G control to make sure to have a broader and more expanded range of control over the smart car for outdoors usage. **Solution:** Under development, a solution still being explored. The goal is to implement a switching feature that will allow the car to switch back to Bluetooth control when it will be used indoors. The SIM8200EA-M2 5G Hat that is used for the 5G development is showcased in Fig. 13.

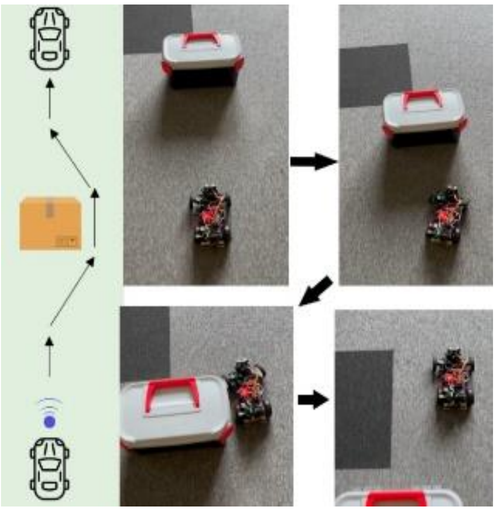

Fig. 12 The development of the Obstacle Avoidance System

Fig. 13 SIM8200EA-M2 5G Hat

VI. Future Developments and Improvements

A. *Optic Sensor for 2D Localization (Indoors)*

Plan: Integrate an optic sensor (Fig. 14) for 2D localization to track the car's position, particularly for indoor use. The optic sensor used is similar to the sensors commonly found in computer mice. This type of sensor will be able to capture precise surface details and movements for the smart car's localization.

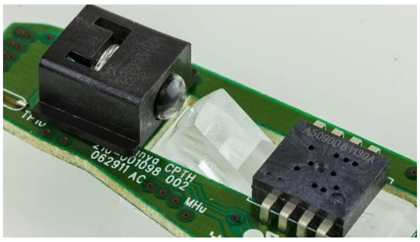

Fig. 14 Logitech M210 Optical Mouse Sensor

B. *LEDs for Stoplights and Turning Signals*

Plan: Add two LEDs for stoplights and four LEDs for turning signals for a better simulation of a real car. This functionality will add to the realism of the car and will complete the lighting system that was started with the introduction of the headlights.

C. *Addition of Camera for Point-of-View Streaming*

Plan: To install a camera (Fig. 15) that will provide live feed of the car's point of view, enabling remote control without the car being in the user's line of sight. By integrating a camera, the whole system will get a new upgraded user experience and will allow the project to be significantly improved.

Fig. 15 Raspberry Pi Camera & Arduino Camera

D. *LEDs for Stoplights and Turning Signals*

Plan: Adding two LEDs for stoplights and four LEDs for turning signals for a better simulation of a real car. This functionality will add to the realism of the car and will complete the lighting system that was started with the introduction of the headlights.

E. *"Coming Back to Home" mode*

Plan: To develop a mode that will allow the user to set a "home" position and after pressing a button, the car will use the obstacle avoidance system to safely navigate the home position that was saved earlier.

F. *Rotating Ultrasonic Sensor*

Plan: To attach a new Servo Motor to that car and set the Ultrasonic on top of it. This configuration will allow the Sensor to rotate independently, to scan the area in front of the car, without requiring the entire car to move. At the same time, this setup will create more flexibility and improve the smart car's overall navigation capabilities.

G. *Improvement of Obstacle Avoidance System*

Plan: The improvement of the Obstacle Avoidance System, using the Rotating Ultrasonic Sensor to make sure that the car can handle any situation it can encounter.

H. *Changing the Arduino Uno Board*

Plan: There is a possibility for a potential transition to an alternative board such as Arduino Mega or even Raspberry Pi. Both boards are visually represented in Fig. 16. The Arduino Mega has a surplus of ports, opening the possibilities for additional components, while the Raspberry Pi could be a better fit for 5G modules. This shift seeks to optimize the smart car's capabilities, offering opportunities for a lot more sophisticated features.

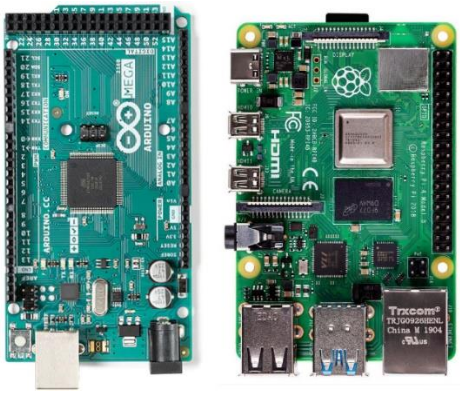

Fig. 16 Arduino Mega 2560 & Raspberry Pi 4 Model B

VII. RESULTS

The implementation of the Bluetooth controlled smart car with obstacle avoiding capabilities had some significant outcomes, showing successful integration of various components. One of the key results is that the smart car responded effectively to the commands from the PlayStation5 controller, offering a very intuitive experience. The analog acceleration functionality allows precise commands and control from the user, offering a more pleasant driving experience. The safety features worked as intended and the servo motor's straightening mechanism in the absence of commands worked as it should. At the same time, the emergency stop functionality with complete controller disconnection did not show any issues during testing. The USB receiver compatibility issue was fixed by trying out different receivers.

All these results mark a successful implementation of the smart car project, its user-friendly interface being approved even by other people that had the chance of testing it out.

The development of the 5G integration would expand the car's control range for outdoors applications which would open a more broader area for exploration, learning and development.

The car's autonomous capabilities can be improved and a lot more features can be added to make the project even more advanced than it currently is.

VIII. CONCLUSIONS

This smart car project showcases a fusion of creativity and technology. From 3D-printed chassis to the PlayStation5 Controller, it merges innovation and user-friendly design and experience. The addition of an analog acceleration elevates the car's functionality, responding dynamically to user input. The obstacle avoidance system adds a layer of intelligence, making it more than just a remote-controlled car.

This smart car project serves as a step towards simulating a real-world car with similar driving experience with features that can be expanded and developed over time.

REFERENCES

[1] Data Sheet, "A000066 Arduino UNO R3" from https://docs.arduino.cc/hardware/uno-rev3

[2] Data Sheet and Library " USB Host Shield Library 2.0" from https://www.arduino.cc/reference/en/libraries/usb-host-shield-library-2.0/

[3] Oleg M., Alexei G., Kristian Sloth L., guruthree, Yuuichi A., Aran V. " USB Host Library Rev. 2.0" from https://felis.github.io/USB_Host_Shield_2.0/index.html

[4] Chassis Model, Rodney Majola " 4WD RC Smart Car Chassis for Arduino" from https://www.thingiverse.com/thing:2575467

[5] Video "Arduino USB Host Shield no power fix" from https://www.youtube.com/watch?v=1dxwU87GQVM

[6] Data Sheet " MG90S Servo Motor Datasheet" from https://www.alldatasheet.com/

[7] Course "Controlling DC Motors with the L289N Dual H-Bridge and an Arduino" from https://dronebotworkshop.com/dc-motors-l298n-h-bridge/

[8] Data Sheet "Ultrasonic Ranging Module HC – SR04" from https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf