DESIGN AND DEVELOPMENT OF ROBOSWAN

A Dissertation for

Course code and Course Title: ELO-401 & Project

Credits: 8

Submitted in partial fulfilment of Master's Degree

M.Sc. in Electronics

By

Shivnath S Sangodkar

Ankita A Nigalye

Rohit Kumar Mahato

Roll Number:

21P036012

21P036005

21P036011

Under the Supervision of / Mentor

Dr. Narayan Vetrekar

School of Physical and Applied Sciences

Electronics

Goa University

Date:

10th May, 2023

Examined by:

Seal of the School

COMPLETION CERTIFICATE

This is to certify that the dissertation "DESIGN AND DEVELOPMENT OF ROBOSWAN" is a bonafide work carried out by Ms. Ankita A Nigalye, Mr Rohit Kumar Mahato, Mr Shivnath S Sangodkar under my supervision/mentorship in partial fulfilment of the requirements for the award of the degree of MSc in the Discipline Electronics at the School of Physical & Applied Science's, Goa University.

Dr. Narayan Vetrekar

Electronics Discipline

School of Physical and Applied Sciences

Date: 10.05.2023

Prof. Kaustubh R.S. Priolkar

School Of Physical And Applied Sciences

Date:

Place: Goa University

School of Physical and Applied Sciences Goa University

School Stamp

ACKNOWLEDGEMENT

We would like to thanks faculty of electronics department in specific Dr. Rajendra Gad, Dr. Jivan Parab, Dr. Narayan Vetrekar, Dr. Marlon Sequeira and Dr. Aniketh Gaonkar for there support and assistance and direction on project.The Registrar Of Goa University for providing us necessary funding.

we would like to thank Mr. Anand Lokapure Scientist at NIO for giving us the idea of RoboSwan as the project title. with his guidance the project was carried out.

We also want to thank Mr.Vishant Malik (LAB TECHNICIAN) for his on going support and assistance in providing us with the necessary components for our project.

We would specially like to express our gratitude to Mr. William D'souza for his assistance for all office related task,and non-teaching staff members Pushpa Andrade and Ashwini Velip for assistance and encouragement.

Thank you to all of our colleagues ,friends, and goa university student who have assisted and collaborated with us .We appreciate all of our family member for moral support and encouragement throughout the project.

And, of course thanks to everyone we may have overlooked. All the persons mention above are responsible for the success of our initiative

DECLARATION BY STUDENT

I hereby declare that the data presented in this Dissertation entitled, "DESIGN AND DEVELOPMENT OF ROBOSWAN" is based on the results of investigations carried out by me in the Electronics Department at the School of Physical and Applied Science's University under the Supervision/Mentorship of Dr. Narayan Vetrekar and the same has not been submitted elsewhere for the award of a degree or diploma by me. Further, I understand that Goa University or its authorities will be not be responsible for the correctness of observations / experimental or other findings given the dissertation.

I hereby authorize the University authorities to upload this dissertation on the dissertation repository or anywhere else as the UGC regulations demand and make it available to any one as needed.

Ms. Ankita A. Nigalye
Seat no:21P036005
Electronics Discipline
School Of Physical And Applied Sciences

Mr.Rohit Kumar Mahato
Seat no:21P036011
Electronics Discipline
School Of Physical And Applied Sciences

Mr. Shivnath S. Sangodkar
Seat no:21P036012
Electronics Discipline
School Of Physical And Applied Sciences

Date: 10.05.2023

Place: Goa University

Table of Contents

LIST OF FIGURES

DESIGN AND DEVELOPMENT OF ROBOSWAN

Abstract:

This project is about design and development of RoboSwan . We use robots in monitoring and improving the environment, particularly in collecting data on water quality. The development of USVs and robotic swans by various researchers and institutions is highlighted, with their advantages and disadvantages discussed. Our main focus was to design RoboSwan which resulted in the creation of seven different models of a highly efficient and practical tool for data collection. The final model incorporated several improvements, including the use of a geared DC motor, an ESP32 microcontroller, and an app for control and data collection. Through testing and analysis, the final model showed significant improvements in performance and functionality, with potential uses ranging from oceanographic to industrial applications. Our project emphasizes the importance of continuous improvement and collaboration in achieving success in the field of robotics.

DESIGN AND DEVELOPMENT OF ROBOSWAN

CHAPTER 1.

INTRODUCTION

1.1 BACKGROUND

Robots are increasingly being utilized in everyday life to monitor and improve our environments. Currently, the process of collecting readings of water temperature, pH levels, detection of pollution in water by collecting sample and water characteristics is carried out manually or with the use of crewed vessels which are expensive and provide a low data acquisition rate. This entails that water quality in many public water bodies is not monitored, or the monitoring has a considerable interval and a small coverage area.[2]

To automate the process and make it more accessible, many USVs have been developed recently, which are able to perform the metrics collection process autonomously. Researchers from the research ARC (Autonomous Robot) was conducted at Abernathy University (Penglais, UK).and VECTOR (Federal Institute of Education, Science, and Technology of Ceara, Brazil), has built ARC ship which has greater speed and maneuverability, which makes its missions faster, and also partially increases its resistance to wind and waves. [1]

The disadvantage of this project was the system was not provided with obstacles detection and detect floating garbage, add sensors for environmental monitoring, and develop an algorithm for autonomous return of the boat to base .it was easily detected in water which were causing disturbance to the birds. To overcome this issue the Researchers from the National University of

Singapore have created a bevy of robotic swans that test the quality of drinking water in Singapore's reservoir [2].

The robotic birds, collectively named NUSwan, autonomously swim across the water's surface using under-body propellers. Fitted with a number of sensors, they are designed to monitor the quality of freshwater lakes and reservoirs – such as levels of dissolved oxygen or chlorophyll – while blending in with the natural environment. Disadvantage with this project are propellers were in place of mechanical legs, it wasn’t autonomous, lacking in sensors [2].

1.2 TYPE OF VEHICLES FOR THE WATER INSPECTION.

1)Unmanned surface vehicle (SUV):

An unmanned surface vehicle (USV), sometimes known as an unmanned surface vehicle, autonomous water vehicle (ASV), unmanned surface vehicle (USV), or colloquially as a drone ship or robotic boat, is a boat that operates on the surface of the ocean without any crew autonomously. unmanned surface vehicle operates with different levels of autonomy, from remote controlled to fully autonomous [3].

A USV performs tasks in a variety of cluttered environments without human intervention and exhibits inherently highly nonlinear dynamics. Advancements in USV are expected to bring significant benefits, including Reduced development and operating costs, increased personnel safety, increased operational coverage (reliability) and accuracy, increased autonomy, and increased flexibility in demanding environments (so-called dirty and tedious). including tough and dangerous missions) [4].

With the help of more effective, compact, commercially available and affordable navigation devices such as global positioning systems (GPS) and inertial measurement units (IMUs), and more powerful and reliable wireless communication systems. Borrowing brought greater opportunities. More open than ever to USV and its applications. USVs can be cost-effectively designed for a variety of potential applications Such as scientific research, environmental missions, marine resource exploration, military use and other applications [4].

2)Unmanned underwater vehicles (UUV)

Unmanned surface vehicles are the vehicles which operates inside the water without any human inhabitant**.**

These vehicles can be divided into two categories:

1) Remotely Operated Underwater Vehicles (ROUV)
2) Autonomous Underwater Vehicles (AUV).

1)Remotely operated underwater vehicles (ROV) [5].

The ROUV is remotely operated by a human operator. A remotely operated underwater vehicle (ROV) is a type of unmanned underwater vehicle that is operated by a person on the surface using a control system. ROVs are typically tethered to a surface vessel or platform, and they are used for a variety of applications such as underwater exploration, inspection, maintenance, and scientific research.

ROVs are equipped with cameras, lights, and sensors that allow operators to remotely view and control the vehicle's movements and activities underwater. They can be used to access difficult-to-reach areas and to perform tasks such as pipeline inspection, underwater construction, and deep-sea exploration.

ROVs come in a variety of sizes and configurations, from small, portable units that can be operated by a single person, to large, complex systems that require a team of operators and support personnel. They are commonly used in industries such as oil and gas, marine research, and oceanography, and they have become an important tool for exploring and studying the ocean depths

2) Autonomous unmanned vehicle (AUV)[6].

The AUV is automated and operates independently of direct human input. An autonomous unmanned vehicle (AUV) is a type of robotic vehicle that is designed to operate without a human operator or control system. Unlike ROVs, AUVs are not tethered to a surface vessel or platform and are capable of operating independently in the water. They are equipped with sensors, cameras, and navigation systems that allow them to navigate underwater environments and perform a variety of tasks.

AUVs are used for a wide range of applications, including oceanographic research, underwater mapping, pipeline inspection, and military operations. They are particularly useful in applications where it is difficult or dangerous for humans to operate, such as in deep-sea exploration or mine countermeasures.

AUVs can be programmed to follow specific missions or routes, and they can collect data and samples as they move through the water. They are typically powered by rechargeable batteries and can operate for extended periods of time without needing to return to a base station or vessel.

Overall, AUVs are an important tool for exploring and studying the underwater environment and have the potential to revolutionize our understanding of the oceans and the life within them.

1.3 MOTIVATION AND OBJECTIVE

Motivation

The future of unmanned surface vehicle rest on the development of full autonomy, enabling unmanned surface vehicle to operate on any environment without human inspection.

Basically, development of such vehicles is challenging.at present such vehicle are in demand because of its advance autonomous feature.

The idea we got while reading research paper based on autonomous underwater vehicle and unmanned surface vehicles.so project is based on unmanned surface vehicle.so we decide to make natural bird/animal which will do the same work as the USV.

That’s why we thought of mimic swan which will do the job of USV as well as it will not disturb the natural habitat and collect all the information required for scientist.

Objective

- Development of swan will be used as essential tool for surface exploration.
- This can be also used local water bodies to perform real time measurements as per scientific needs.
- Our project focuses on biologically inspired approaches in an attempt to mimic naturals leg movement of the bird.
- Instead of using propeller-based design we have come up with mechanical leg design.
- The camera is a sensor that captures images and videos. It could be used to monitor a specific area or object and provide visual feedback to the microcontroller.
- The temperature sensor is a device that measures the temperature of the surrounding environment. It could be used to monitor the temperature of a specific location and provide feedback to the microcontroller.

1.4 SWAN DETAILS

Swans are the largest members of the waterfowl family Anatidae, and are among the largest flying birds. The largest living species, including the mute swan, trumpeter swan, and whooper swan, can reach a length of over[1- 1.5 meter] and weigh over 15 kg to 20 kg. Their wingspans can be over 3.1 meter long [7]. they have larger feet and neck compared to geese which are closely related[8].

Figure 1:show the anatomy and actual structure of Swan.

Swan legs, like the legs of other birds, are specialized for walking, swimming, and flying. They consist of several parts that work together to provide the swan with the support and mobility it needs.

Thigh:

The thigh is the upper part of the swan's leg, located between the body and the knee joint. It contains the femur, which is the largest and strongest bone in the swan's leg.

Knee joint:

The knee joint is located between the thigh and the lower leg. It is a hinge joint that allows the swan to flex and extend its lower leg.

Shank:

The shank is the middle part of the swan's leg, located between the knee joint and the ankle joint. It contains the tibia and fibula, which are the two bones that make up the lower leg.

Ankle joint:

The ankle joint is located between the shank and the foot. It is a complex joint that allows the swan to rotate its foot and adjust the angle of its toes.

Foot:

The foot of a swan is webbed, which helps it to swim efficiently. It also has four toes, with three toes pointing forward and one toe pointing backward. The toes are equipped with strong, sharp claws that help the swan to grip surfaces and defend itself.

Overall, the swan's legs are powerful and adaptable, allowing it to walk on land, swim in water, and fly through the air with ease.

SKELETON OF SWANS LEG:

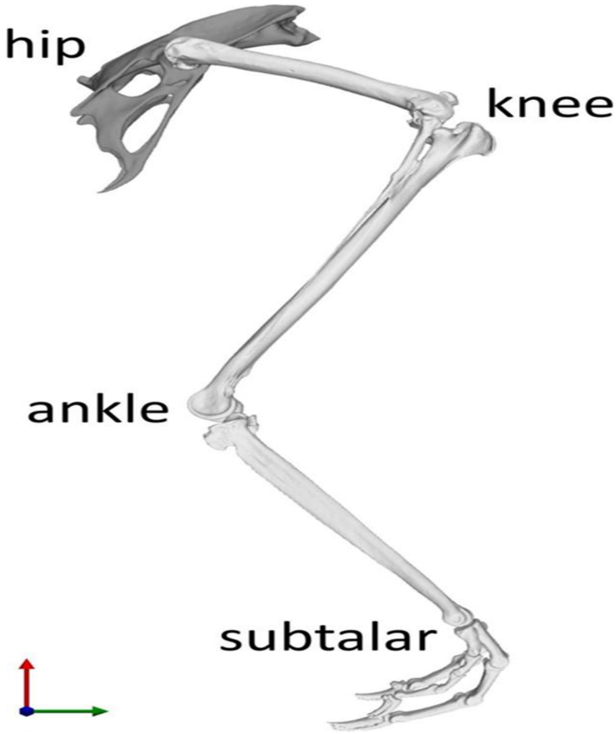


Figure 2: skeleton of swan leg.

WEBBED FEET:

Figure 3: webbed feet

WEBBED MEMBRANE:

Figure 4 : webbed feet.

Swans, like many other aquatic birds, have webbed feet that are adapted for swimming and navigating in water. The webbing is a thin membrane of skin that connects the toes of their feet, which helps to increase the surface area of their feet and distribute their weight evenly, allowing them to swim more efficiently and move through the water with ease.

The webbed feet of swans are particularly well-suited for swimming, as they have long, powerful legs and large, broad feet with a lot of surface area. This makes them strong swimmers, able to paddle through the water with great speed and agility.

In addition to helping them swim, the webbing on their feet also helps swans walk on soft, muddy ground without sinking. The webbed feet distribute their weight over a larger surface area, providing more support and preventing them from sinking too deeply into the mud.

Overall, the webbed feet of swans are a remarkable adaptation that helps them thrive in their aquatic environment, allowing them to swim, forage for food, and avoid predators with ease.

1.5 MECHANICAL WORKING OF SWAN

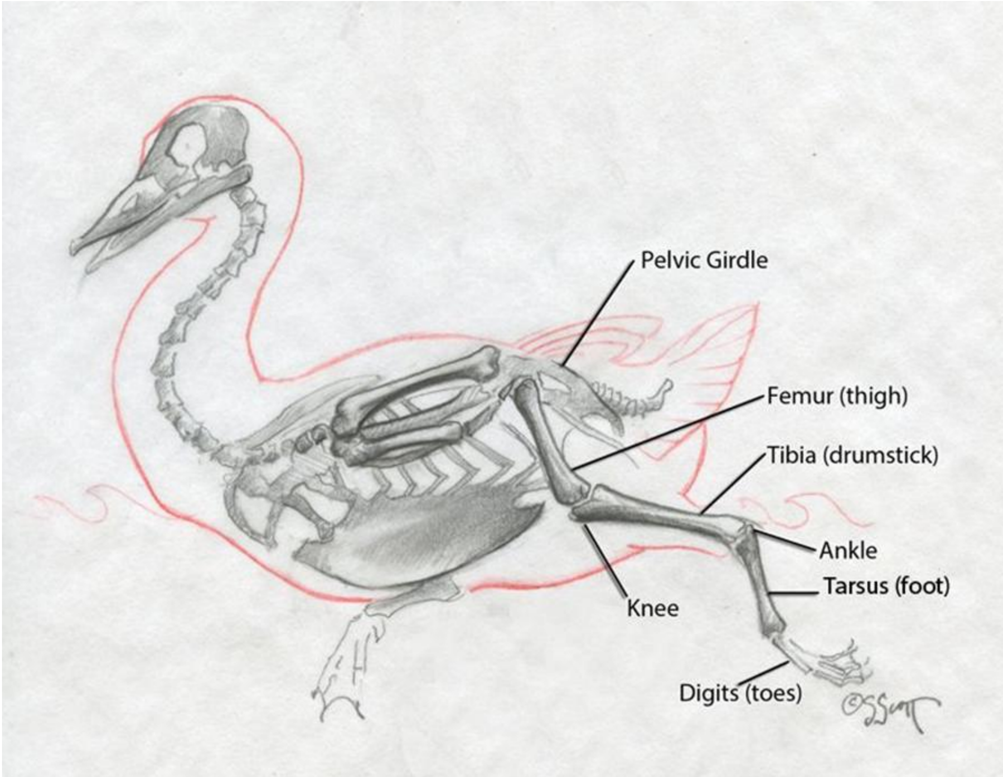


Figure 5: Leg Movement of Swan.

Swan legs are adapted for both swimming and walking on land. The mechanical working of their legs is different depending on the mode of locomotion they are using.

When swimming, swans use their powerful legs and webbed feet to propel themselves through the water. They use a combination of paddling and kicking motions to generate forward motion. The webbed feet increase the surface area of their feet, providing more thrust against the water.

The muscles in their legs are also adapted for swimming. The legs of swans have a lot of power and are able to generate a lot of force with each stroke, allowing them to swim quickly and efficiently. The leg muscles work together in a coordinated way to generate the necessary force for swimming.

When walking on land, swans use their legs and feet to support their body weight and move forward. Their legs are positioned at the back of their body, which gives them good leverage and stability. They walk by taking short, deliberate steps and shifting their weight from one leg to the other.

The mechanical working of their legs when walking on land is different from when swimming. Instead of using the powerful paddling motion of their legs, they rely on the muscles in their legs to provide support and stability. Their feet are not webbed when walking on land, as this would be a hindrance to their movement.

Overall, the mechanical working of swan legs is highly adapted to their environment and mode of locomotion, allowing them to swim and walk with ease and efficiency.

CHAPTER2:

LITERATURE SURVEY

2.1 NUSWAN: SWAN ROBOT TO MONITOR WATER QUALITY . NATIONAL UNIVERSITY OF SINGAPORE(NUS)

NUSWAN

The NUS REACHERS has developed an innovative system who observe the pollutant for proper management and sustainability. This NUSwan has an ability to eliminate the extra use of unnecessary resources which was used for the inspection of the water and it is also economical [2].

Features of NUSwan

1. Sensing Nodes: Fully autonomous, online, natural along recreational activities, rapidsampling over space and time, vertical profiling, low capital cost,scalable, rechargeable, unsupervised operations

2.Maintenance: Automatically return to charging station, low operation logistics.

3.Over the CLOUD Operations: Data available in real time, interactive mission control, simultaneously accessible to users, smart robotic intelligence.

This NUSWan has an ability collect the real time data as per the mission in located area.it is also capable of performing simultaneous multi node, detection of time variant hotspots, high speed sensing for observing concentration gradients for better characterization [2].

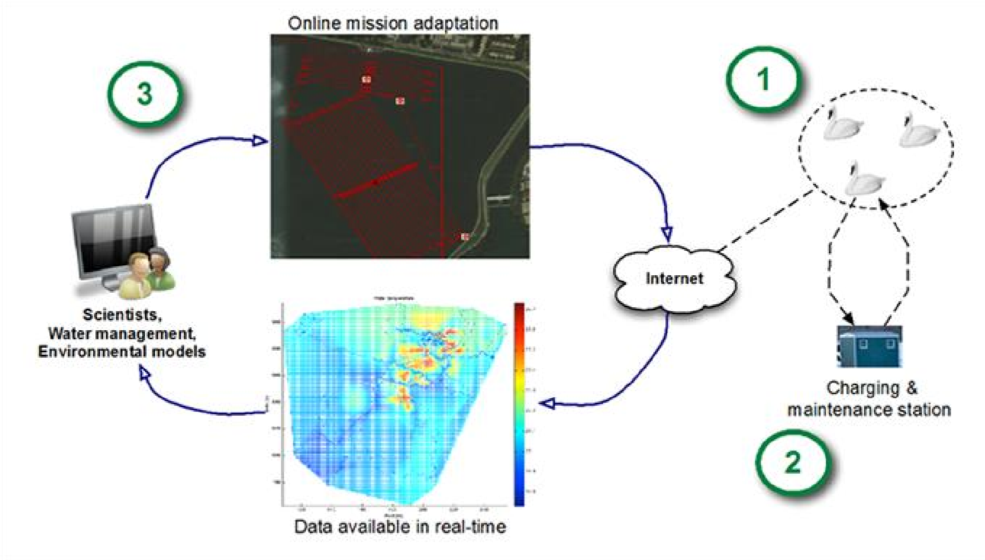


Figure 6: Real time data collection.

The NUS researchers have tested the NUSwan prototype with multi-parameter probes in nearby reservoir in line with directed mission. The facts gathered may be transmitted real-. time via Wi-Fi community inside range.

The NUS researchers have envisaged the improvement of destiny paintings on upgrading visualizations, diving functionality and adaptive sampling. Smart navigation could be hired the use of herbal water motion and cooperative sampling studies to increase the tracking patience of the cellular platform.

The upgraded overall performance could be test-bedded to assess its sampling capability, actual time facts transmission, strength utilization and navigational capabilities.

Continuous efforts are being undertaken in in search of collaborators for destiny improvement and scale-up in addition to companions who're inclined to challenge into commercialization [2].

2.2 AUTONOMOUS WATER QUALITY MONITORING AND WATER SURFACE CLEANING FOR UNMANNED SURFACE VEHICLE (Feng Chia University (FCU)).

Water is one of the maximum valuable resources. However, business improvement has made water pollutants an important trouble these days and for this reason water fine tracking and floor cleaning are critical for water aid protection [9].

In this study, we've got used the sensor fusion technology as a foundation to expand a multi-characteristic unmanned floor vehicle (MF-USV) for impediment avoidance, water-fine tracking, and water floor cleansing. The MF-USV contains a USV manipulate unit, locomotion module, positioning module, an impediment avoidance module, water fine monitoring system, water floor cleansing system, conversation module, electricity module, and remote human–gadget interface.

We equip the MF-USV with the subsequent functions:

(1) autonomous obstacle detection, avoidance, and navigation positioning,

(2) water fine tracking, sampling, and positioning,

(3) water floor detection and cleansing

(4) far off navigation manipulate and real-time facts display.

Lidar is used for the obstacle information. for the obstacle detection vision sensors are use and skip-enet algorithm used. For water quality check pH sensor temperature sensors etc. are used. the body of boat is composed of locomotion module, position module (GPS module), obstacle avoidance module (ultrasonic sensor), water quality monitoring system (pH sensor and water sample collection device), water surface cleaning system (vision sensor and water surface cleaning device), communication module (Bluetooth wireless module), power module, and remote human–machine interface.

The experimental outcomes verified that after the floating garbage located with inside the visible perspective ranged from−30◦to 30◦at the front of the MF-USV and the distances between the floating rubbish and the MF-USV had been forty and 70 cm, the achievement prices of floating garbage detection are all 100%.

When the space among the floating rubbish and the MF-USV was a hundred thirty cm and the floating rubbish become placed at the left side (15◦~30◦), left the front side (0◦~15◦), the front side (0◦), proper the front side (0◦~15◦), and the proper side (15◦~30◦), the achievement prices of the floating garbage series had been 70%, 92%, 95%, 95%, and 75%, respectively.

Finally, the experimental results also verified that MF-USV and applicable algorithms to impediment avoidance, water fine tracking, and water floor cleansing had been effective.

2.3 DESIGN OF SMALL UNMANNED SURFACE VEHICLE WITH AUTONOMOUS NAVIGATION SYSTEM (DEPARTMENT OF COMPUTER-AIDED DESIGN, SAINT PETERSBURG ELECTROTECHNICAL UNIVERSITY “)

Ecosystem conservation is one of the major concerns of the public today. some press A problem that threatens water resources is the pollution of water bodies by illegal flotsam. Extraction of water resources, wear and tear of underwater communications. Therefore, create Special technical solutions are urgently needed. This paper reports on model-based design of unmanned aerial vehicles.

Ground Vehicles (UPS), to control and maintain oxygen content and parameters such as acidity and temperature of rivers, lakes, inland waterways and coastal waters. Of Developed, his USV navigation autopilot is described as a two-input, one-output system.

The selected autopilot is an adaptive controller based on proportional, integral and control concept’s derivative function (PID).

The autopilot is implemented in an STM32 microcontroller and provides an accurate maintaining a constant course, adjusting speed and turning angle when the wind is blowing, etc. affect. A novel technique for sensor calibration and data acquisition is described simulation.

Results are displayed showing the performance of the autopilot algorithm as Surroundings. A numerical experiment of the model yields good agreement between prototype characteristics and simulation results. Thorough at last Field tests were conducted to verify the reliability and accuracy of the proposed solution of the developed unmanned water vehicle is used for environmental monitoring (water sampling, patrols of waterways and bodies of water). Then the resulting solution is suitable for the design of Other UPS with different size.

Advantages:

The advantages of the proposed design are:

- Potential for deployment speed in water.
- Small dimensions that allow the UPS to be transported between different bodies of water.
- Ability to use developed control and navigation systems in different types UPSs are as easy to calibrate and suitability for long-range mission considering the spherical shape of the earth.

Disadvantages:

- The drawbacks are:
- Controllers that do not consider waves or other weather phenomena.
- Lack of long-term data on application of developed systems.

2.4 AN AUTONOMOUS SURFACE VEHICLE FOR WATER QUALITY MONITORING (AUTONOMOUS SYSTEMS LABORATORY, CSIRO ICT CENTRE, UNIVERSITY OF QUEENSLAND)

Autonomous surface Vehicles that can move around the complex Measuring inland water storage Water quality characteristics and greenhouse gas emissions.

16-foot-long solar catamaran can collect this information in the country column while the vehicle is in motion Unique the feature of this ASV is to integrate into the storage-scale floating sensor network to enable remote control, task download, data download and adaptive sampling strategy. the vehicle design and operation including control, laser-based obstacle avoidance, and vision-based inspection capabilities. integration of on-board sensors, including GPS, Laser scanners, sonar and cameras enable ASV operation successfully dodge previously unmapped shallow water environments, and static and mobile obstacles

Experimental result:

it has ability to continuously collect water quality parameters that are important and complement intensive manual monitoring campaign.

2.5 Navigating high-speed unmanned surface vehicles: system approach and validations (Science and Technology on Underwater Vehicle Laboratory, Harbin Engineering University, Harbin, 150001, China b. College of Shipbuilding Engineering, Harbin Engineering University, Harbin, 150001, China c. Department of Mechanical Engineering, University College London, Torrington Place, London WC1E 7JE, UK)

With growing interest in deploying unmanned flotation vehicles (USVs) to support complex ocean operations, High-speed USV (≥ 40 knots) has become an important choice, especially for performing demanding tasks such as coastal navigation Protect. Currently, there is a large amount of research focused on the development of automated USV navigation systems and the results of most of them have only been verified by simulations or demonstrations on experimental low-speed USVs (≤20 knots). It remains unclear how high-speed UPS will perform in real-world applications. Especially during high-speed UPS navigation [10],

Reliable obstacle detection and robust motion control are always harder to achieve than under normal circumstances. Therefore, we present a new high-speed UPS autonomous navigation system with developed system software and hardware. It is based on ensuring safe and reliable operation. Within such systems are velocity-based collision avoidance algorithms Integrated Obstacle (VO) that can adjust UPS course and speed in real time. Local environment modeling method Course maintenance methods considering Coast Guard International Regulations (COLREGs) to avoid collisions at sea. suggested to improve system reliability and stability. The effectiveness and efficiency of the proposed system are verified and validated by hardware-in-the-loop simulation test and sea trial on real high-speed UPS ("Tianxing No.1") marine environment. This result indicates that the autonomous navigation system adapts to the characteristics of high-speed UPS, it can guide high speed UPS (≥40 knots) to achieve safe autonomous navigation, which greatly improves UPS Degree of autonomy.

Problems with high-speed navigation

1. The instability of obstacles.

- ❖ Radar signal noise is generated because surface waves can cause significant noise in radar images making obstacle detection uncertain such uncertainty can also be observed in visible and infrared images due to reflections from the sea surface.

2. The time lag and safety requirements in motion control.[10]

Conclusions and learned lessons

An automatic positioning system specifically designed for high-speed USVs provided with the system's stability, reliability, and real-time operability proven in numerous tests. Simulation and testing at sea. Local environment modeling method and header maintenance method have been designed and integrated into the anti-collision approach.

Some critical learned lessons are listed as follows:

1. System cooling issues:

Each installed unit has its own nominal operating temperature. H. Embedded multi-board industrial computers have a maximum operating temperature of 70°C, and his fiber optic gyroscope (FOG) based IMU has a maximum operating temperature of 60°C. Operating temperatures above ratings can cause inaccurate sensor data and permanent damage to the device.

Apart from high ambient temperatures that can affect the operating temperature of the sensor, the device itself also generates heat and contributes to the temperature rise.

2. Electromagnetic compatibility (EMC) issue:

Three communication modes, including network, radio and satellite communication, were used for the high-speed USV "Tianxing No.1". To improve the robust communication when the USV is far from the shore station, a 20W radio was chosen to improve the communication range. However, the radio antenna will have a great influence on the GPS signal, especially in rainy or foggy weather conditions, this makes GPS speed and location data errors beyond the system's tolerance in some cases. On 'Tianxing No.1', radio antenna and GPS antenna are assigned to separate locations to solve such problem. In addition, when GPS signal is affected, an alternative, Beidou signal, will be considered. [10]

3. Hydraulic actuation timing problem:

The steering control of the water jet propulsion system and the surface drive is carried out by hydraulic actuators, the delay of which is much larger than that of the engine transmission. The total time from hard port to hard starboard for 'Tianxing No.1' was approximately 10 seconds, which necessitated the development of more robust adaptive motion control strategies for high-speed USVs.

4. Shockproof problem of components and sensors:

When the high-speed USV is traveling in the planning mode, the hull will be severely impacted by the wave effect. Therefore, the shock resistance of components and sensors used for high-speed USV must be fully considered. Shock absorption measures should be applied to precision sensors, such as IMUs.

6. An Automatic Navigation System for Unmanned Surface Vehicles in Realistic Sea Environments (School of Marine Electrical Engineering, Dalian Maritime University, Dalian 116026, China).

In recent years, Unmanned Surface Vehicles (USVs) have received considerable attention due to their many advantages in both civilian and military applications. To improve the autonomy of the USV, this article describes a complete automatic navigation system (ANS) with a (PPS) and the collision avoidance subsystem 'CAS'. PPS based on Dynamic Domain Tunable Fast Marching Square (DTFMS) method can build an environment model from a real electronic board where static and dynamic obstacles are well represented. By adjusting the saturation, the generated path can be changed according to the security and path length requirements. It is then used as a guide for the CAS through a dynamic target point. In CAS, according to the Finite Control Set Model Predictive Control (FCS-MPC) theory, a collision avoidance control algorithm is developed to track the trajectory and avoid collisions based on the model of the motion. the three-degree-of-freedom (DOF) plane of the USV. Its target score and safety rating are derived from PPS's projected trajectory and environment model. In addition, the predicted CAS trajectory can guide changes in the ship's own dynamic domain model. Finally, the system was tested and validated on scenarios of three types of encounters in real marine environments.[11]

In this paper, a complete ANS is presented with PPS (DTFMS) and CAS (FCS-MPC) in a real marine environment. First, the TFMS method is proposed to model the stationary marine environment and the orbital plane, which can be adjusted by saturation of safety weight and orbital length. Then, following the FCS-MPC theory, the CAS was developed to follow the trajectory and avoid collisions based on the predictive model of the USV. Dynamic target scores and safety assessments are derived from PPS orbital planning and environmental modeling. Then, DTFMS method is proposed with dynamic domain models (path re-planning algorithm), which can represent dynamic circuits well, including own and target circuits. There, the dynamic domain models are modified according to the ship's motion state and the environment's update time. Finally, by comparing the static and dynamic tests, the simulation results validated the effectiveness of the proposed ANS in simulated and real marine environments.

Future work will first examine the issue that the feasibility of ANS can be further augmented for truly fully autonomous USV trials. First, in order to model the marine environment more realistically, the marine environment model needs to add elevation and surface currents. Second, the collision avoidance control algorithm must also consider power consumption and environmental influences (wind, waves and currents). Third, the Convention on International Regulations for the Prevention of Collisions at Sea (COLREG) should be included in the ANS.

CHAPTER 3:

HARDWARE

3.1 L298N

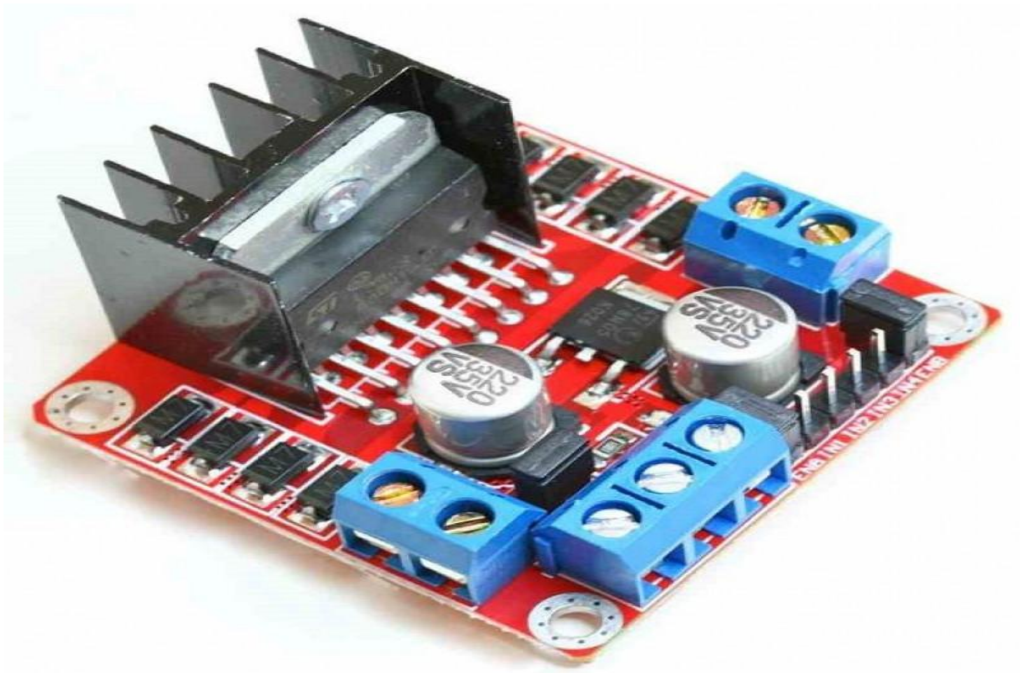


Figure 7:L298N Driver.

This L298 Based Motor Driver Module is a high-power motor driver perfect for driving DC Motors and Stepper Motors. It uses the popular L298 motor driver IC and has the onboard 5V regulator which it can supply to an external circuit. It can control up to 4 DC motors, or 2 DC motors with directional and speed control.

This motor driver is perfect for robotics and mechatronics projects and perfect for controlling motors from microcontrollers, switches, relays, etc. Perfect for driving DC and Stepper motors for micro mouse, line following robots, robot arms, etc.

An H-Bridge is a circuit that can drive a current in either polarity and be controlled by Pulse Width Modulation (PWM).

Pulse Width Modulation is a means of controlling the duration of an electronic pulse. In motors try to imagine the brush as a water wheel and electrons as the flowing droplets of water. The voltage would be the water flowing over the wheel at a constant rate, the more water flowing the higher the voltage. Motors are rated at certain voltages and can be damaged if the voltage is applied to heavily or if it is dropped quickly to slow the motor down. Thus PWM. Take the water wheel analogy and think of the water hitting it in pulses but at a constant flow. The longer the pulses the faster the wheel will turn, the shorter the pulses, the slower the water wheel will turn. Motors will last much longer and be more reliable if controlled through PWM.

PWM Control

Pulse Width Modulation (PWM) is a technique used to control the amount of power delivered to a load such as a motor. PWM works by varying the width of the pulse while keeping the frequency constant. The wider the pulse, the more power is delivered to the load. By varying the pulse width, the average power delivered to the load can be controlled, allowing for precise speed control.

The L298 2A Dual Motor Driver Module features PWM control, which can be useful for applications that require variable speed control. PWM control can be implemented using a microcontroller or other control system by varying the duty cycle of the PWM signal. The module has two input pins for PWM control, one for each channel.

Motor Connections

The L298 2A Dual Motor Driver Module has two output channels, each capable of controlling the speed and direction of a DC motor or stepper motor. The module has four output pins per channel: two for motor direction control and two for motor speed control.

To connect a motor to the L298 2A Dual Motor Driver Module, you would typically connect the motor's positive and negative leads to the module's output pins. The motor direction can be controlled using the direction control pins, and the motor speed can be controlled using the speed control pins.

Power Requirements

The L298 2A Dual Motor Driver Module requires power to operate. The module has two power input pins, one for each channel. The power input pins are typically labeled VCC and GND.

The voltage and current requirements of the power source will depend on the motor being used. The L298 IC can handle a wide range of input voltages, typically between 5V and 35V. However, the voltage should be chosen based on the voltage rating of the motor being used. The current rating of the power source should also be sufficient to handle the maximum current required by the motor.

In addition to the power input pins, the L298 2A Dual Motor Driver Module may also require additional power sources depending on the voltage and current requirements of the motor being used. For example, a stepper motor may require a separate power source for the motor coils. The power requirements for the motor should be carefully considered when designing the system.

Brief about L298N Module

The L298N Motor Driver module consists of an L298 Motor Driver IC, 78M05 Voltage Regulator, resistors, capacitor, Power LED, 5V jumper in an integrated circuit.

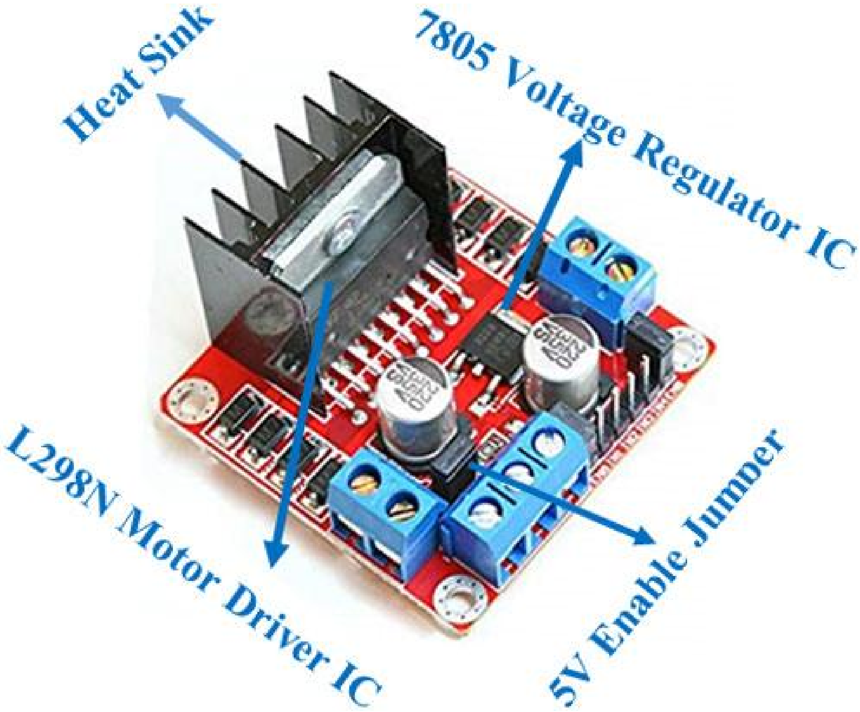


Figure 8:L298N motor driver

78M05 Voltage regulator will be enabled only when the jumper is placed. When the power supply is less than or equal to 12V, then the internal circuitry will be powered by the voltage regulator and the 5V pin can be used as an output pin to power the microcontroller. The jumper should not be placed when the power supply is greater than 12V and separate 5V should be given through 5V terminal to power the internal circuitry.

Internal Diagram of L298N:

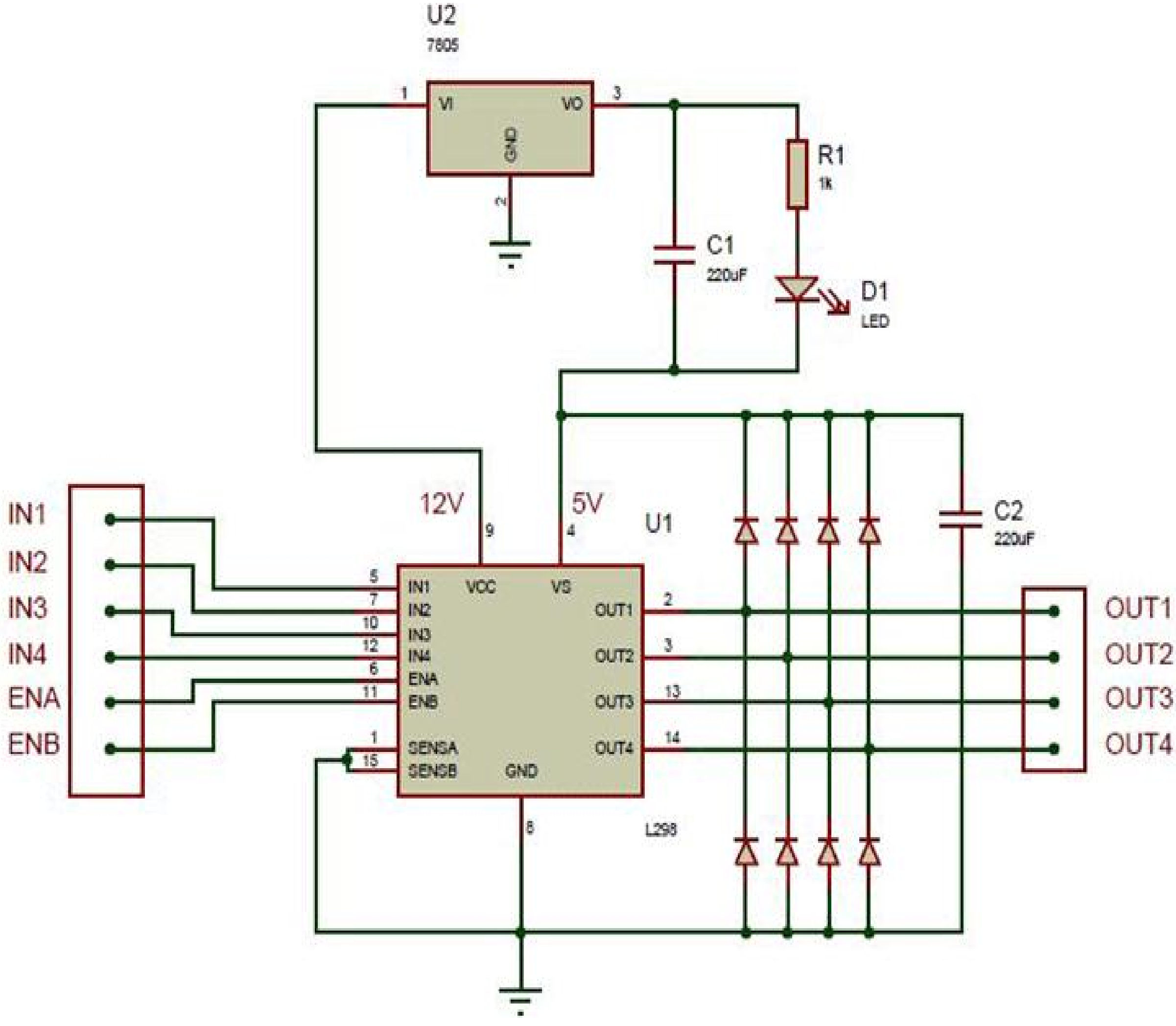


Figure 9:Internal diagram of L298N motor driver.

Features & Specifications:

1. Driver Model: L298N 2A
2. Driver Chip: Double H Bridge L298N
3. Motor Supply Voltage (Maximum): 46V
4. Motor Supply Current (Maximum): 2A
5. Logic Voltage: 5V
6. Driver Voltage: 5-35V
7. Driver Current:2A
8. Logical Current:0-36mA
9. Maximum Power (W): 25W
10. Current Sense for each motor
11. Heatsink for better performance
12. Power-On LED indicator

Pins: -

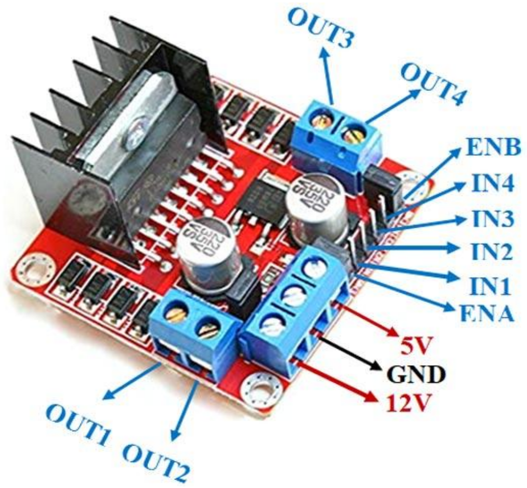


Figure 10:pins of L298N motor driver.

L298N Module Pinout Configuration

Pin Name	Description
IN1 & IN2	Motor A input pins. Used to control the spinning direction of Motor A
IN3 & IN4	Motor B input pins. Used to control the spinning direction of Motor B
ENA	Enables PWM signal for Motor A
ENB	Enables PWM signal for Motor B
OUT1 & OUT2	Output pins of Motor A
OUT3 & OUT4	Output pins of Motor B
12V	12V input from DC power Source
5V	Supplies power for the switching logic circuitry inside L298N IC
GND	Ground pin

3.2 DC MOTOR

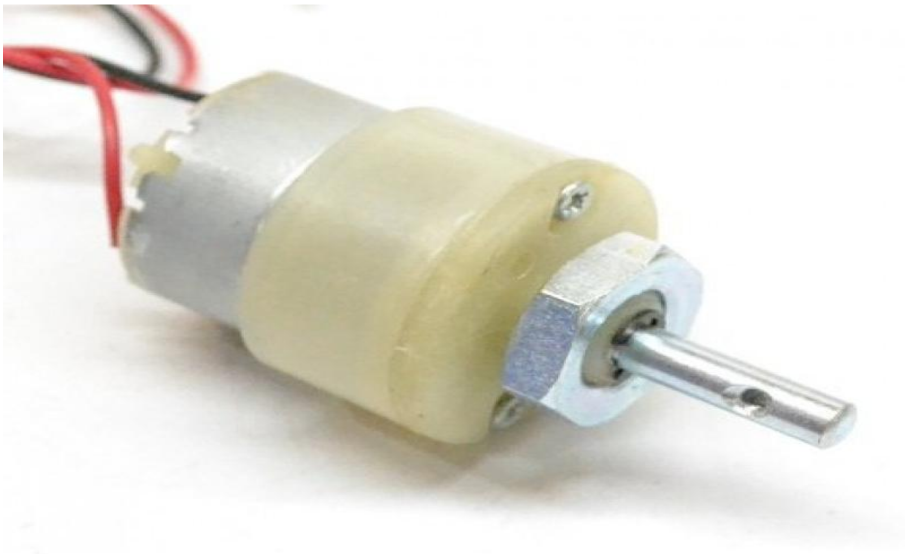


Figure 11:DC geared motor.

100 RPM 12V DC geared motors are small, powerful motors that are commonly used in a wide range of applications, including robotics, automation, and hobby projects. These motors are designed to provide high torque at low speeds, making them ideal for applications that require precise control and maneuverability. In this article, we will provide detailed information about 100 RPM 12V DC geared motors, including their specifications, features, and applications.

Specifications:

The 100 RPM 12V DC geared motor is a small, high-performance motor that is designed to provide high torque at low speeds. These motors typically have a voltage range of 6V to 12V, and a maximum speed of 100 RPM.

The torque rating of these motors is usually between 1 kg-cm to 3 kg-cm, which makes them ideal for applications that require high torque at low speeds. The motor shaft diameter is usually between 3mm to 6mm, which allows for easy integration with a wide range of mechanical components.

Feature:

100 RPM 12V DC geared motors are known for their high torque and low-speed capabilities, but they also offer a number of other features that make them ideal for a wide range of applications. Some of the key features of these motors include:

1. High efficiency: 100 RPM 12V DC geared motors are designed to be highly efficient, which means that they consume less power and generate less heat than other types of motors. This makes them ideal for applications where power consumption is a concern.
2. Low noise: These motors are designed to operate quietly, which makes them ideal for applications where noise is a concern. This is achieved through the use of high-quality gears and bearings, as well as a well-designed motor housing.
3. Compact size: 100 RPM 12V DC geared motors are typically small and compact, which makes them easy to integrate into a wide range of applications. This also makes them ideal for applications where space is limited.
4. Durability: These motors are designed to be durable and long-lasting, with high-quality gears and bearings that can withstand heavy use and high loads.

Applications:

100 RPM 12V DC geared motors are used in a wide range of applications, including:

1. Robotics: These motors are commonly used in robotics applications, where they provide the high torque and precise control required for complex movements.
2. Automation: These motors are used in a variety of automated systems, such as conveyor belts, automated doors, and robotic arms.
3. Hobby projects: 100 RPM 12V DC geared motors are commonly used in hobby projects, such as RC cars, drones, and model trains.
4. Medical devices: These motors are also used in medical devices, such as prosthetic limbs and surgical robots.

Conclusion:

In summary, 100 RPM 12V DC geared motors are small, high-performance motors that offer high torque and low-speed capabilities. These motors are highly efficient, operate quietly, and are durable and long-lasting. They are used in a wide range of applications, including robotics, automation, hobby projects, and medical devices. If you are looking for a powerful and versatile motor for your project, a 100 RPM 12V DC geared motor may be just what you need.

3.3 BATTERY: Orange 5200mAh 4S 40C/80C (14.8V) Lithium Polymer Battery Pack.

Model No: ORANGE 5200/3S-40

Figure 12: LiPo battery.

The ORANGE 5200mAH 4S 40C (14.8V) LiPo battery pack with an XT60 connector is equipped with heavy-duty discharge leads to minimize resistance and sustain high current loads. The ORANGE 5200mAH 4S 40C (14.8V) LiPo battery pack with XT60 connector have a JST-XH style balance connectors. All Orange Lithium Polymer batteries packs are assembled using IR matched cells.

Orange batteries are known for performance, reliability, and optimum price also.

It is not surprising to us that Orange Lithium polymer packs are the go-to pack for those in the know.

Orange batteries deliver the full rated capacity at a price everyone can afford.

The ORANGE 5200mAH 4S 40C (14.8V) LiPo battery pack has a matched resistance and true balance. These Orange batteries have good temperature control after high-rate discharge.

Features:

1. Product Type: Lithium Polymer Battery Pack
2. The Orange LiPo battery has matched resistance.
3. Good Temperature Control.
4. Minimum weight in Class.

Specification:

Model No.	ORANGE 42999/4S-40C
Capacity (mAh)	5200
Weight (gm)	488
Output Voltage (VDC)	14.8
Charge Rate (C)	1 ~ 3
Discharge Plug	XT-60
Balance Plug	JST-XH
Length (mm)	137
Width (mm)	44
Height (mm)	40
Max. Burst Discharge (C)	50C (10 sec).
Max. Charge Rate	5 C
Max. Continuous Discharge	40C
Shipment Weight	0.49 kg
Shipment Dimensions	17 × 6 × 5 cm

Calculation of power consumption:

DEVICES	POWER CONSUMPTION	
	VOLTAGE	CURRENT
ESP32 CAM	5V	310mA
ESP32	5V	200mA
L298N DRIVER	12V	36mA
GEAR DC MOTOR	12V	500mA
FT232RL USB TO TTL CONVERTER	MAX 5V	500mA
DS18B20 WATERPROOF TEMPERATURE SENSOR	5V	1.5mA

NOTE: BATTERY CAPACITY: 5200MAH. O/P VOLTAGE:14.8V.

TOTAL CURRENT REQUIRED: 1547.5mA

FORMULA:

BATTERY LIFE (IN MIN) = $\frac{\boldsymbol{BATTERY\ CAPACITY\ IN}\ (\boldsymbol{mAH})}{\boldsymbol{TOTAL\ \ LOAD\ CURRENT\ IN}(\boldsymbol{mAH})}$

$$= \frac{\mathbf{5200}}{1547.5}$$

$$= 216 \text{ minutes}$$

1.4 ESP32

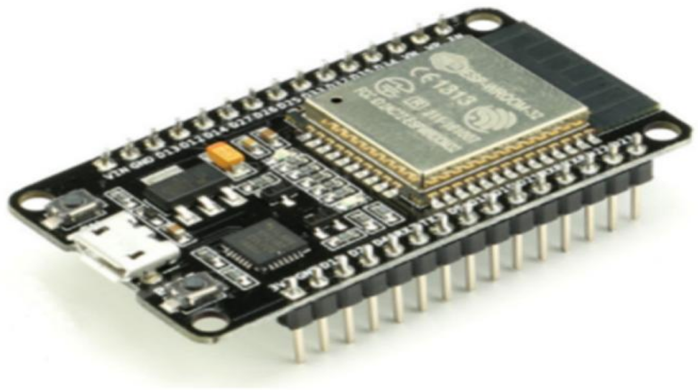


Figure 13:ESP32 Board.

The ESP32 is a powerful and versatile microcontroller designed for IoT (Internet of Things) applications. It was developed by Espressif Systems and is based on the Xtensa LX6 dual-core processor.

Here is a more detailed overview of the key features and capabilities of the ESP32:

Featured Solutions

1. Ultra-Low-Power Solution

   ESP32 is designed for mobile, wearable electronics, and Internet-of-Things (IoT) applications. It features all the state-of-the-art characteristics of low-power chips, including fine-grained clock gating, multiple power modes, and dynamic power scaling. For instance, in a low-power IoT sensor hub application scenario, ESP32 is woken up periodically and only when a specified condition is detected. Low-duty cycle is used to minimize the amount of energy that the chip expends. The output of the power amplifier is also adjustable, thus contributing to an optimal trade-off between communication range, data rate and power consumption.

2. Complete Integration Solution

   ESP32 is a highly-integrated solution for Wi-Fi-and-Bluetooth IoT applications, with around 20 external components. ESP32 integrates an antenna switch, RF balun, power amplifier, low-noise receive amplifier, filters, and power management modules. As such, the entire solution occupies minimal Printed Circuit Board (PCB)area. ESP32 uses CMOS for single-chip fully-integrated radio and baseband, while also integrating advanced calibration circuitries that allow the solution to remove external circuit imperfections or adjust to changes in external conditions. As such, the mass production of ESP32 solutions does not require expensive and specialized Wi-Fi testing equipment.

Processor:

The ESP32 features a dual-core 32-bit processor based on the Xtensa LX6 architecture, running at up to 240 MHz. The processor has a 4-stage pipeline and supports both integer and floating-point operations.

Memory:

The ESP32 includes 520 KB of SRAM (Static Random Access Memory) and 4 MB of Flash memory for program and data storage. The Flash memory can be partitioned to support multiple applications and data storage.

Wireless Connectivity:

The ESP32 features Wi-Fi and Bluetooth connectivity, making it ideal for IoT applications that require wireless communication. The Wi-Fi module supports IEEE 802.11 b/g/n/e/i standards and can operate in both Access Point and Station modes. The Bluetooth module supports both Classic Bluetooth and Bluetooth Low Energy (BLE) protocols.

Interface Options:

The ESP32 includes a wide range of interface options, including GPIO (General Purpose Input/Output) pins, I2C, SPI, UART, ADC (Analog to Digital Converter), DAC (Digital to Analog Converter), and other interfaces. These interfaces can be used to communicate with sensors, actuators, displays, and other devices.

Operating Systems:

The ESP32 can run various operating systems, including Arduino, Micro Python, and Free RTOS. These operating systems provide high-level programming interfaces and libraries that simplify application development

1.Operating Voltage:2.2 V to 3.6 V

2.Operating Temperature Range: - 40°C to +85°C

3. Power Consumption:

- Sleep current: < 5 µA
- Active mode current: 20 mA to 200 mA depending on operating frequency, RF conditions, and active peripherals.

Programming Languages:

The ESP32 can be programmed using various programming languages, including C, C++, Python, and JavaScript. There are also many development environments available for the ESP32, including the Arduino IDE, Visual Studio Code, and Eclipse. Development Environment:

- ❖ Supports various operating systems, including Arduino, Micro Python, and Free RTOS.
- ❖ Supports programming languages including C, C++, Python, and JavaScript.
- ❖ Various development tools and IDEs available, including the Arduino IDE, Visual Studio Code, and Eclipse.

Application

The ESP32 can be used in a wide range of applications, including home automation, industrial automation, smart devices, and wearable technology. Its Wi-Fi and Bluetooth capabilities make it ideal for wireless communication, while its powerful processor and extensive interface options provide great flexibility for programming and interfacing with other devices. Overall, the ESP32 is a popular and reliable microcontroller that is well-suited for a variety of IoT applications. Its powerful processor, wireless connectivity, and interface options make it a versatile platform for developing and deploying IoT solution.

The ESP32 microcontroller has a number of pins that can be used for interfacing with other devices or sensors.

PIN DIAGRAM:

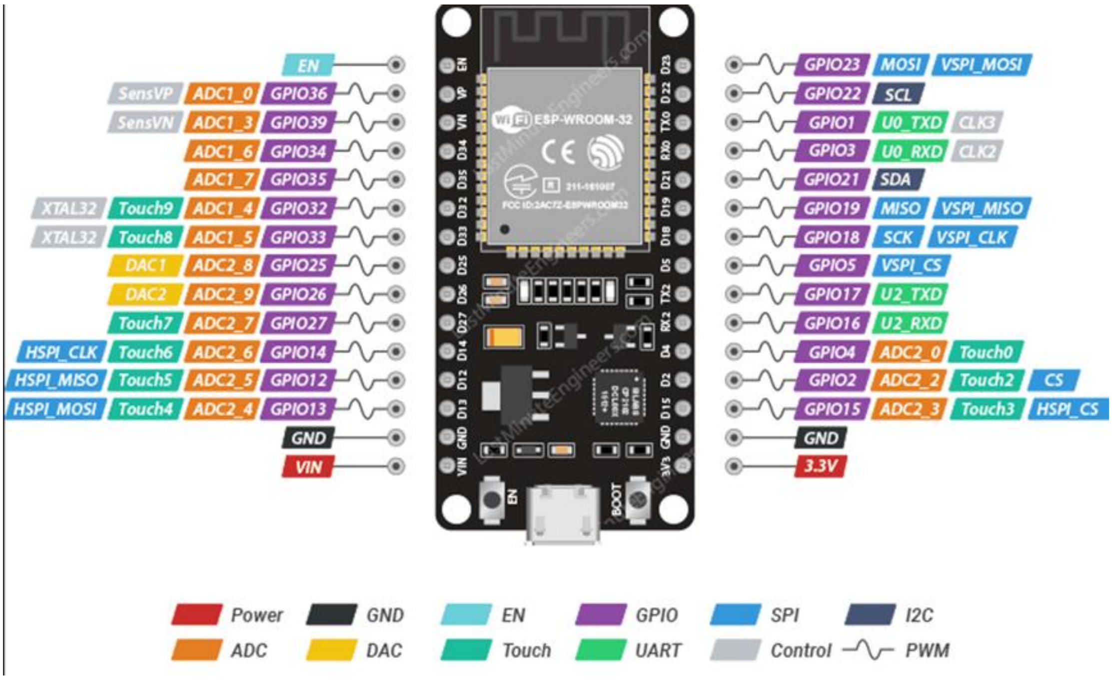


Figure 14:Pins of ESP32 board.

Here is an overview of the pinouts of the ESP32.

1.GPIO Pins:

The ESP32 has 34 GPIO (General Purpose Input/Output) pins that can be used for digital input/output, analog input, or pulse-width modulation (PWM) output. These pins are labeled GPIO0 to GPIO33.

2.Analog Input Pins:

The ESP32 has 18 analog input pins, which are labeled ADC1_CH0 to ADC1_CH7, ADC2_CH0 to ADC2_CH7, and ADC2_CH8 to ADC2_CH11. These pins can be used to measure analog voltages, such as those from sensors or potentiometers.

3.I2C Pins:

The ESP32 has two I2C (Inter-Integrated Circuit) pins, which are labeled SDA and SCL. These pins are used for two-way communication between the ESP32 and other devices that support the I2C protocol.

4.SPI Pins:

The ESP32 has four SPI (Serial Peripheral Interface) pins, which are labeled MOSI, MISO, SCK, and CS. These pins are used for communication between the ESP32 and other devices that support the SPI protocol.

5.UART Pins:

The ESP32 has three UART (Universal Asynchronous Receiver/Transmitter) pins, which are labeled TX, RX, and RTS. These pins are used for serial communication between the ESP32 and other devices.

6.DAC Pins:

The ESP32 has two DAC (Digital to Analog Converter) pins, which are labeled DAC1 and DAC2. These pins can be used to output analog signals, such as audio or voltage signals.

7.Touch Pins:

The ESP32 has ten touch pins, which are labeled T0 to T9. These pins can be used to detect touch input from capacitive touch sensors.

8. Other Pins:

The ESP32 also has various other pins, including EN (Enable), BOOT (Boot mode), and various power and ground pins.

Overall, the ESP32 offers a wide range of pinouts that can be used for interfacing with other devices and sensors, making it a versatile platform for developing IoT solutions.

1.5 ESP32 CAM

Figure 15:ESP32-cam

The ESP32-CAM is a development board that combines an ESP32-S chip with a camera module, making it a powerful platform for building IoT projects with image and video capabilities.

Here's some detailed information on the ESP32-CAM:

1. ESP32-S Chip:

   The ESP32-S chip is a powerful microcontroller with dual-core processor, Wi-Fi and Bluetooth connectivity, and a wide range of peripheral interfaces. It can operate at up to 240 MHz clock frequency and supports up to 4 MB of flash memory and 520 KB of SRAM.

2. Camera Module:

   The ESP32-CAM comes with a small camera module that can capture images and videos at resolutions up to 1600x1200 pixels. It also has a built-in lens and supports infrared (IR) illumination for low-light environments.

3. Interfaces:

The ESP32-CAM has a number of interfaces for connecting peripherals and sensors, including UART, SPI, I2C, ADC, DAC, and PWM. It also has a microSD card slot for storing images and videos, as well as a USB interface for programming and debugging.

4. Power Management:

   The ESP32-CAM includes power management features to optimize power consumption and extend battery life. It supports deep sleep modes and can wake up from sleep on predefined events, such as a button press or a sensor reading.

5. Software Development:

   The ESP32-CAM can be programmed using the Arduino IDE, which provides an easy-to-use interface and a large library of pre-built functions and examples. It also supports programming in C and C++, and can be integrated with other development tools, such as ESP-IDF and Micro Python.

Applications:

The ESP32-CAM can be used for a wide range of IoT applications that require image and video processing, such as surveillance cameras, smart doorbells, robotics, and environmental monitoring systems. Its small size and low power consumption make it ideal for battery-powered projects.

Overall, the ESP32-CAM is a versatile and powerful development board that provides a cost-effective and easy-to-use platform for building IoT projects with image and video capabilities.

Pinouts:

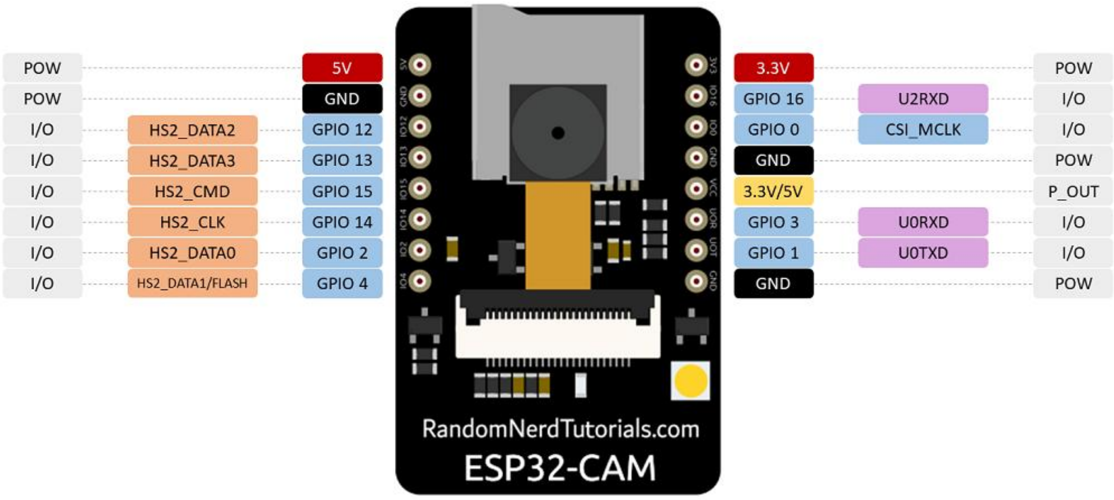


Figure 16: Pinout of ESP32-CAM.

Pin No.	Name	Type	Function
1	5V	POWER	5V Supply
2	GND	POWER	Ground pin
3	IO12	I/O	GPIO 12/MicroSD DATA2
4	IO13	I/O	GPIO 13/MicroSD DATA3
5	IO15	I/O	GPIO 15/MicroSD CMD
6	IO14	I/O	GPIO 14/MicroSD CLK
7	IO2	I/O	GPIO 2/MicroSD DATA0
8	IO4	I/O	GPIO 12/MicroSD DATA1/ Flash

9	GND	POWER	Ground pin
10	UOT/IO1	I/O	UART TX /GPIO 1
11	UOR/IO3	I/O	UART RX /GPIO 3
12	VCC	POWER	5V/3.3V
13	GND	POWER	Ground pin
14	IO0	I/O	GPIO 0 / Boot select
15	IO16	I/O	GPIO 16
16	3V3	POWER	3.3V Supply

Features:

1. Onboard ESP32-S module, supports Wi-Fi + Bluetooth
2. OV2640 camera with flash
3. Onboard TF card slot, supports up to 4G TF card for data storage
4. Supports WIFI video monitoring and Wi-Fi image upload
5. Supports multi sleep modes, deep sleep current as low as 6Ma
6. Control interface is accessible via pin-header, easy to be integrated and embedded into user products.

Specification:

- WIFI module: ESP-32S

- Processor: ESP32-D0WD
- Built-in Flash: 32Mbit
- RAM: Internal 512KB + External 4M PSRAM
- Antenna: Onboard PCB antenna
- Wi-Fi protocol: IEEE 802.11 b/g/n/e/i
- Bluetooth: Bluetooth 4.2 BR/EDR and BLE
- WIFI mode: Station / SoftAP / SoftAP+Station
- Security: WPA/WPA2/WPA2-Enterprise/WPS
- Output image format: JPEG (OV2640 support only), BMP, GRAYSCALE
- Supported TF card: up to 4G
- Peripheral interface: UART/SPI/I2C/PWM
- IO port: 9
- UART baud rate: default 115200bps
- Power supply: 5V
- Transmitting power:
- 802.11b: 17 ±2dBm(@11Mbps)
- 802.11g: 14 ±2dBm(@54Mbps)
- 802.11n: 13 ±2dBm (@HT20, MCS7)
- Reception sensitivity:
- CCK,1Mbps: -90 dBm
- CCK,11Mbps: -85 dBm
- 6Mbps (1/2 BPSK): -88 dBm
- 54Mbps (3/4 64-QAM): -70 dBm
- HT20, MCS7(65Mbps, 72.2Mbps): -67 dBm

- Power consumption:
- Flash off: 180mA@5V
- Flash on and brightness max: 310mA@5V
- Deep-Sleep: as low as 6mA@5V
- Modern-Sleep: as low as 20mA@5V
- Light-Sleep: as low as 6.7mA@5V
- Operating temperature: -20 ℃ ~ 85 ℃
- Storage environment: -40 ℃ ~ 90 ℃, <90%RH
- Dimensions: 40.5mm x 27mm x 4.5mm

3.6 The FT232RL USB to TTL converter

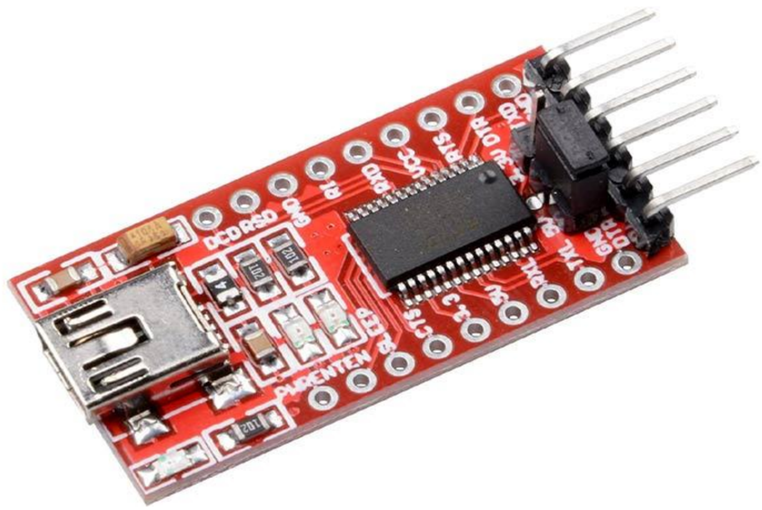


Figure 17

The FT232RL USB to TTL converter is a popular integrated circuit (IC) used to convert Universal Serial Bus (USB) signals to Transistor-Transistor Logic (TTL) signals. It is commonly used to communicate with microcontrollers or other devices that use TTL serial communication.

The FT232RL IC is manufactured by FTDI (Future Technology Devices International) and is a single-chip solution for USB to asynchronous serial data transfer. It incorporates a USB 2.0 Full Speed controller, USB transceiver, crystal oscillator, EEPROM, and a UART (Universal Asynchronous Receiver/Transmitter) interface.

The FT232RL is capable of data transfer rates up to 3Mbps and supports a wide range of data formats including 7, 8, or 9-bit word lengths, 1, 1.5 or 2 stop bits, and odd, even, mark, space, or no parity.

The converter board typically features a USB Type-A connector for connecting to a USB port on a computer, and a set of TTL serial pins (usually labeled TX, RX, VCC, and GND) for connecting to a microcontroller or other device that uses TTL serial communication. Some boards may also include additional features such as LEDs to indicate power and data transfer status.

Overall, the FT232RL USB to TTL converter is a widely used and versatile tool for interfacing with TTL serial devices using a USB connection.

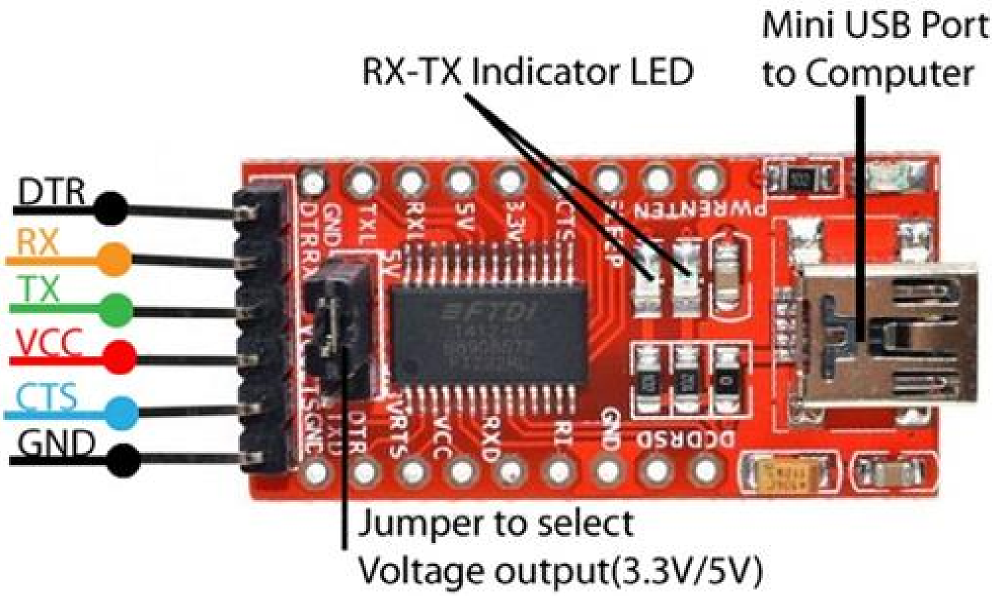


Figure 18

Features and Specifications of FT232RL USB TO TTL Converter:

This section mentions some of the features and specifications of the FT232RL USB to TTL Converter:

- Operating Voltage: 5V/3.3V DC.
- Max Current Draw: 5V - 500mA; 3.3V - 50mA.
- Connector: Mini USB.
- Transmit and receive LED drive signals.
- Fully integrated clock generation with no external crystal required.

Alternatives for FT232RL USB TO TTL Converter

FT234XD, UMFT201XB-01 XBee, TTL-232R-3V3, VNC1 Module VDIP1.

Pin Configuration of FT232RL USB TO TTL Converter

The FT232RL USB to TTL Converter has 6 pins. The table below depicts all the pin types along with the function of each pin.

Pin Type	Description
DTR	Data Terminal Ready (Output used for flow control)
RX	Serial Data Reciever
TX	Serial Data Transmit
VCC	Power Input
CTS	Clear to send (Input used for flow control)
GND	Ground

Applications of FT232RL USB TO TTL Converter

Below are the applications of FT232RL USB TO TTL Converter:

1.Programming AVR, ARM-based microcontrollers/microprocessors.

2.USB to RS232/RS422/RS485 Converters.

3. USB Smart Card Readers.

4. For serial communication with GPS devices.

5. Serial terminal on devices like Raspberry Pi.

2D Model of FT232RL USB TO TTL Converter.

The 2D model of the FT232RL USB TO TTL Converter can be seen in the image below; the dimensions of the board shown are in millimeters and it can be followed to design custom footprints of the module.

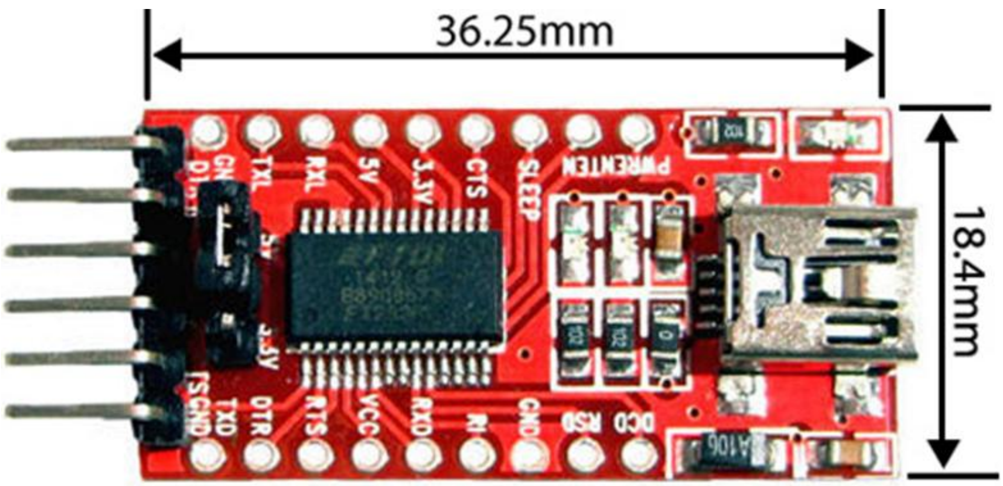


Figure 19

3.7 DS18B20 WATERPROOF TEMPERATURE SENSOR:

Figure 20:DS18B20 temperature sensor.

The DS18B20 waterproof temperature sensor is a digital temperature sensor that is designed to operate in harsh environments where exposure to water or moisture is a concern. It is based on the DS18B20 temperature sensor chip, which uses the 1-Wire communication protocol to communicate with a microcontroller or other digital device.

Here is some detailed information about the DS18B20 waterproof temperature sensor:

Temperature Range:

The DS18B20 waterproof temperature sensor has a temperature range of -55°C to +125°C, which makes it suitable for a wide range of applications. The sensor can operate at temperatures as low as -55°C, making it ideal for use in cold environments. It can also operate at high temperatures of up to +125°C, which makes it suitable for use in high-temperature applications.

Accuracy:

The sensor has an accuracy of ±0.5°C over the range of -10°C to +85°C, making it suitable for many temperature measurement applications. This level of accuracy is adequate for most applications, and it can be improved by using calibration techniques.

Waterproof Design:

The DS18B20 waterproof temperature sensor is enclosed in a waterproof stainless steel casing that protects it from moisture and other environmental factors. The casing is designed to be rugged and durable, making it suitable for use in harsh environments. The sensor has a waterproof cable that allows it to be used in wet environments such as underwater applications.

1-Wire Protocol:

The DS18B20 waterproof temperature sensor uses the 1-Wire communication protocol, which allows multiple sensors to be connected to a single data line. This makes it easy to connect multiple sensors to a microcontroller or other digital device. The 1-Wire protocol is a low-cost, low-speed bus that requires only one data line for communication.

Compatibility:

The sensor is compatible with a wide range of microcontrollers and digital devices, including Arduino boards, Raspberry Pi, and other single-board computers. It can be easily interfaced with these devices using software libraries that are readily available. The sensor can be connected to the microcontroller using a three-wire interface (data, power, and ground), which simplifies the connection process.

Power Supply:

The DS18B20 waterproof temperature sensor requires a power supply voltage of between 3V and 5.5V. The sensor can operate at low voltages, making it suitable for battery-powered applications. The sensor has a low-power mode that reduces its power consumption, making it ideal for applications where power consumption is a concern.

Working Principle of DS18B20 Waterproof Temperature Sensor

The working principle of the DS18B20 Waterproof temperature sensor is similar to any other temperature sensor. The resolution of the sensor ranges from 9-bits to 12-bits. But 12-bit is used as the default resolution to power up this sensor. It measures temperature, as well as the conversion of Analog-to-Digital (A-to-D), which can be done with a convert-T command. The output temperature value can be stored within the 2-byte register in the sensor, and after that, this sensor returns to its inactive state.

The DS18B20 Temperature Sensor has three wires/pins (Vcc, ground, and data wires) for operation. But in parasite mode, only the ground and data lines are used to operate the sensor, the input voltage of the sensor is supplied through the data line.

Technical Features:

1. Unique 1-Wire® Interface Requires Only One Digital Pin of microcontroller/microprocessor for Communication.
2. In Parasitic Power Mode, Requires Only 2 Pins for Operation (Data and GND pin).
3. Multiple temperature sensors can share one pin of microcontroller/microprocessor.
4. No External Components are Required.
5. Temperature Sensor and EEPROM.
6. Unique 64-bit address enables multiplexing.
7. Programmable alarm options.
8. It is Available in 8-Pin SO (150 mils), 8-Pin µSOP, and 3-Pin TO-92 Packages.

DS18B20 Waterproof Temperature Sensor Specifications:

Parameter	Value
Sensor Type	Programmable Digital Temperature Sensor
Operating Voltage	3.3V – 5V
Operating Current	1.5mA
Measuring temperature range	-55ºC to +125ºC (-67°F to +257°F)
Accuracy	±0.5°C Accuracy from -10°C to +85°C
Programmable Resolution	9 to 12 bit (Selectable)
Conversion Time	< 750ms
Sensing Probes	Stainless Steel tube

Sensor Probe Length/ Diameter	45mm Long, 6mm Diameter
Cable Length/ Diameter	36 Inch / 91cm long, 4mm Diameter

Application:

1. This sensor is widely used to calculate the temperature in a rigid environment like mines, chemical solutions, soil, etc.
2. Measuring Liquid temperature
3. it can be used in the thermostat controls system
4. Used in industries as a temperature-measuring device
5. It is very useful where the temperature has to be measured at multiple points

CHAPTER 4:

SOFTWARE

4.1 BLOCK DIAGRAM:

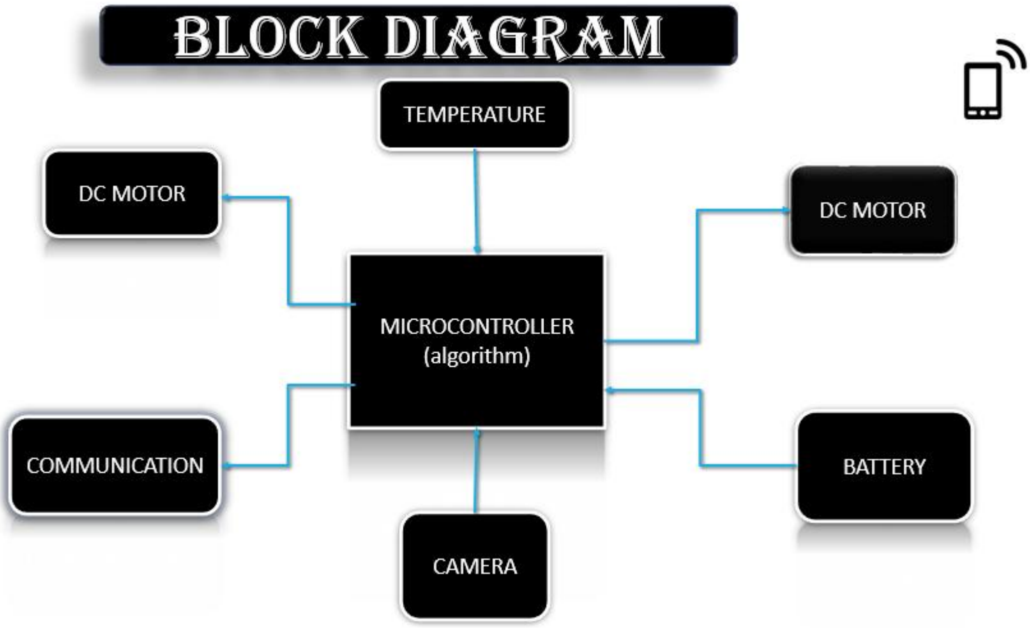


Figure 21.1: Block diagram.

The block diagram consists of a microcontroller, 2 dc geared motor, a battery, a camera, and a temperature sensor.

The microcontroller is the main processing unit of the system that controls and coordinates the operation of the other components. It receives inputs from the temperature sensor and the camera and sends commands to app.

The motor is a device that converts electrical energy into mechanical energy, and it is used to control the movement of a physical object. In this system, the motor could be used to move a leg of RoboSwan.

The battery is the power source for the system. It provides the electrical energy needed to operate the microcontroller, dc geared motor, camera, and temperature sensor.

The camera is a sensor that captures images and videos. It could be used to monitor a specific area or object and provide visual feedback to the microcontroller.

The temperature sensor is a device that measures the temperature of the surrounding environment. It could be used to monitor the temperature of a specific location and provide feedback to the microcontroller.

Together, these components form a system that could be used for a variety of applications such as a temperature-controlled, leg movement of RoboSwan or a surveillance camera with motorized movement.

CIRCUIT DIAGRAM:

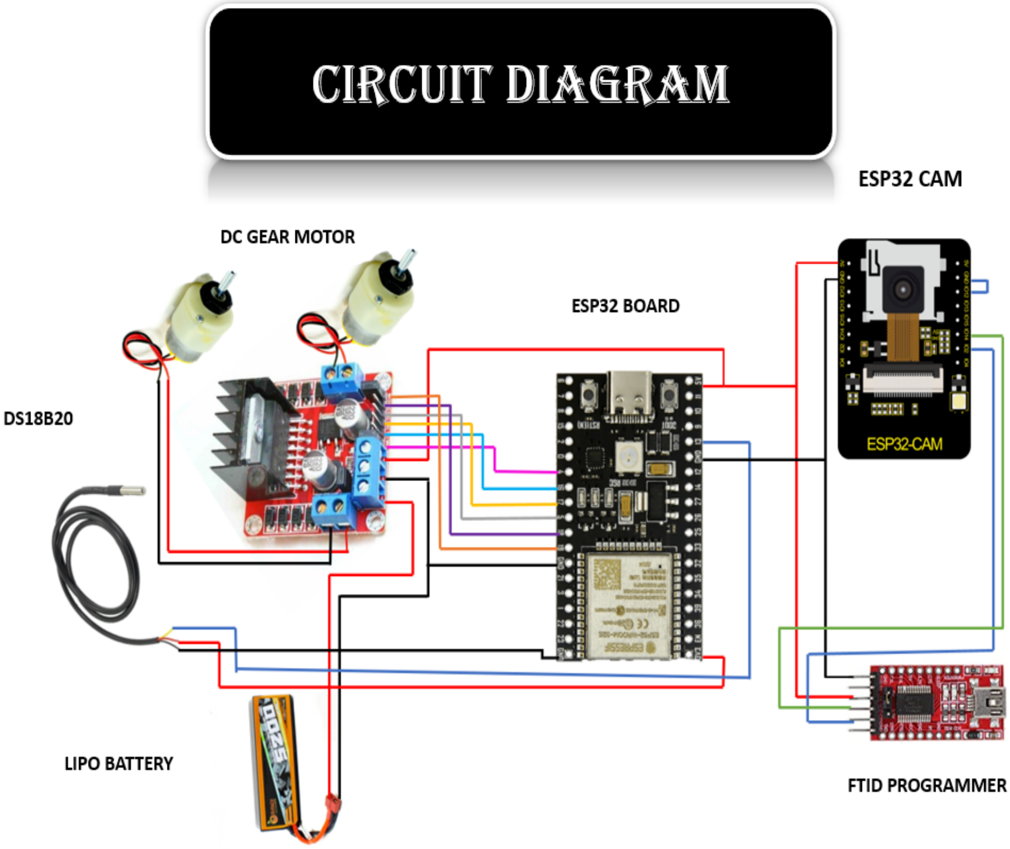
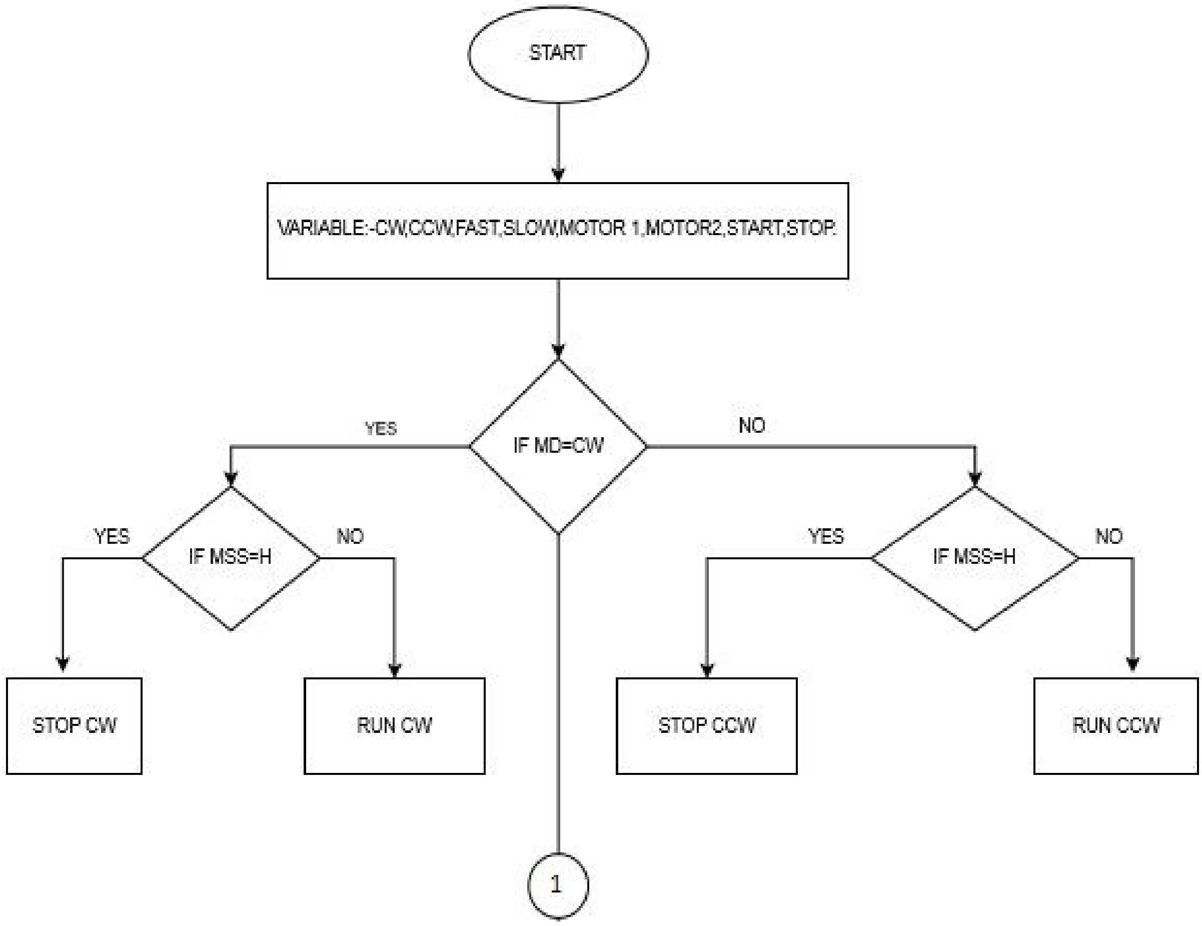


Figure 21.2: Circuit diagram.

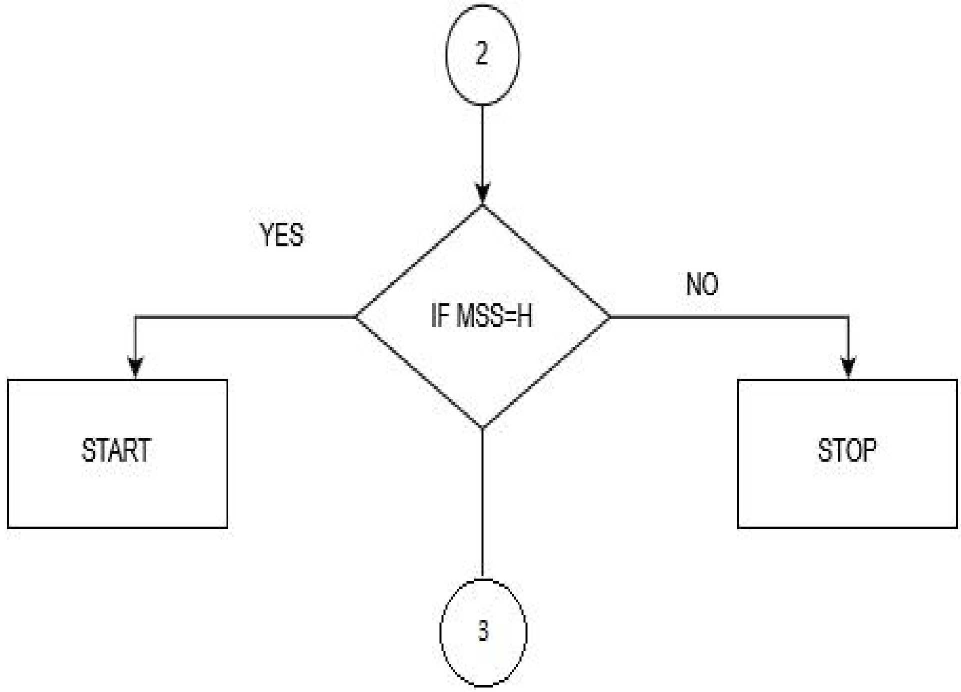
4.2 FLOWCHART

1.MOTOR DIRECTION:



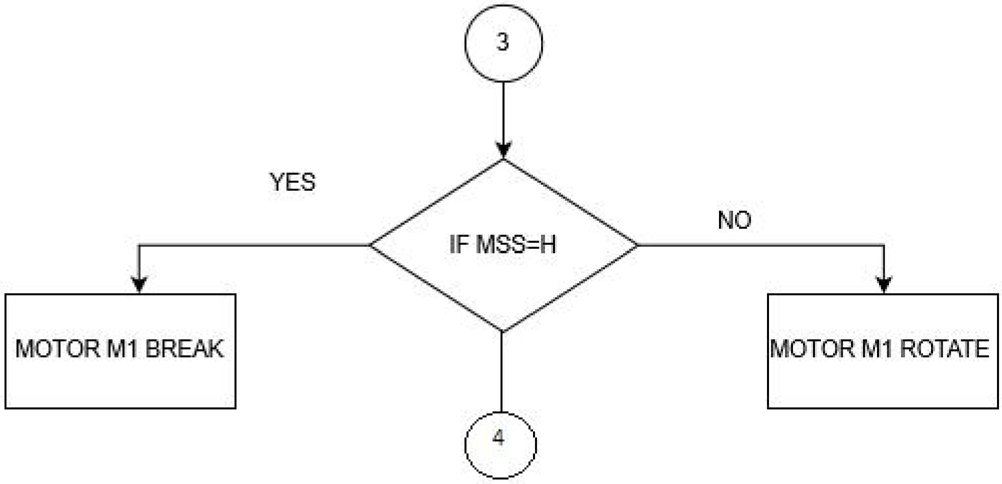
FULL FORM	
MD	MOTOR DIRECTION
CW	CLOCK WISE
MSS	MOTOR STOP STATE
CCW	COUNTER CLOCK WISE
H	HIGH STATE

2.FOR MOTOR START / STOP:



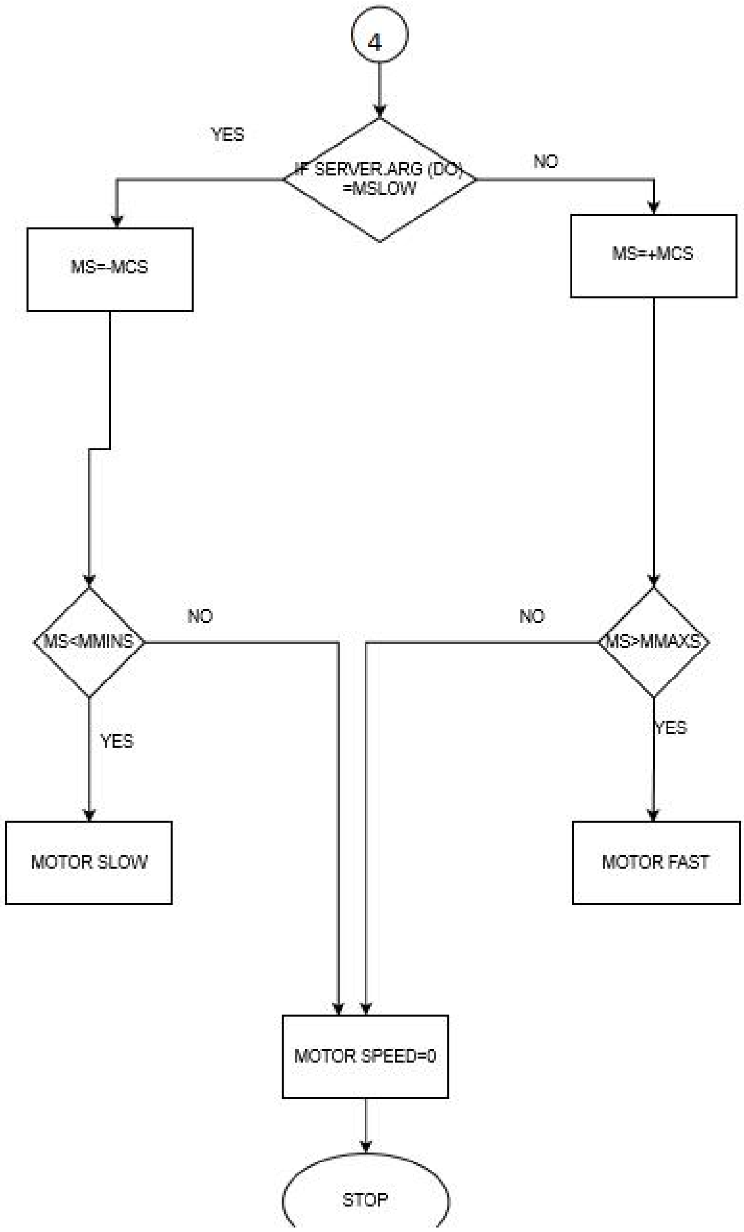
FULL FORM	
MSS	MOTOR STOP STATE

2. 3.MOTOR BREAK

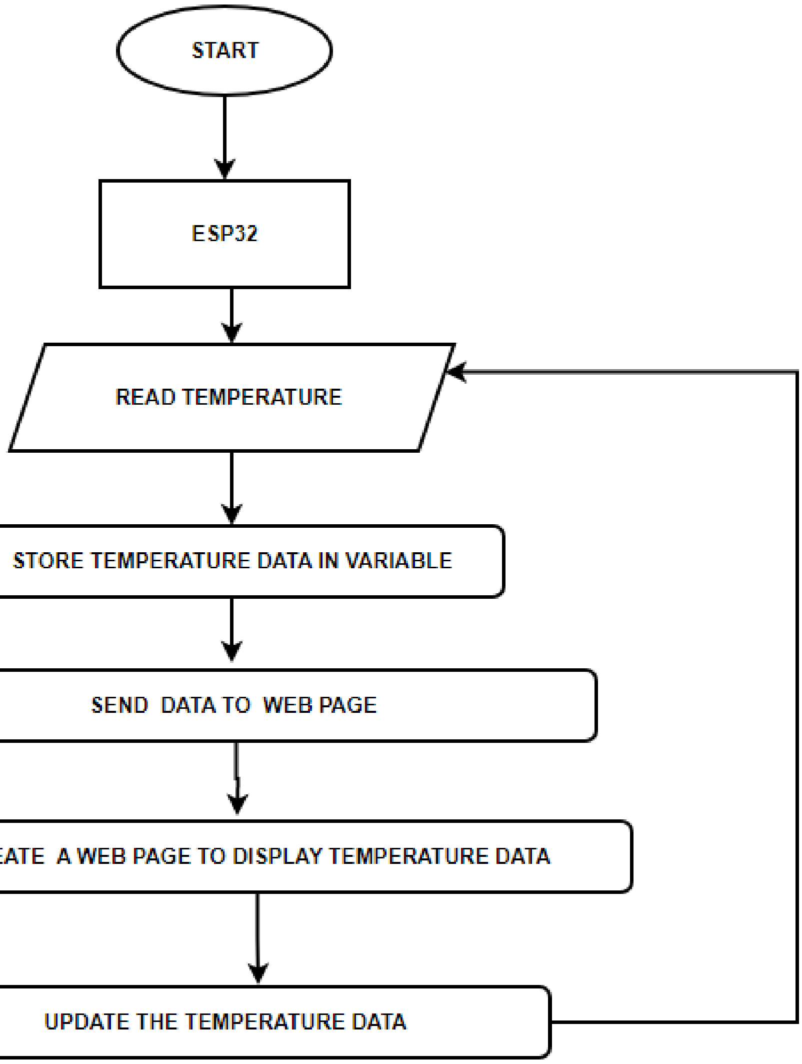


FULL FORM	
MSS	MOTOR STOP STATE

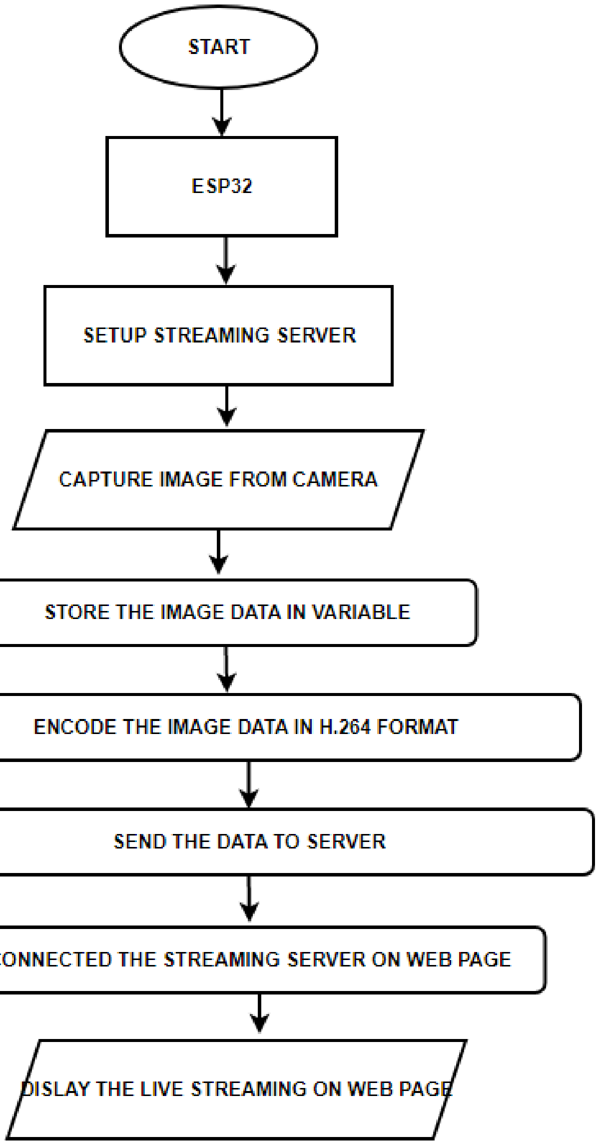
4.FOR SPEED OF MOTOR



5.FLOWCHART FOR TEMPERATURE



6.FLOWCHART FOR CAMERA



CHAPTER 5:

CONTROL

APP: RASA

5.1 WEB PAGE

Web page has different buttons for two motor like start, stop, clock wise, anticlockwise and for increasing and decreasing speed of the motor.

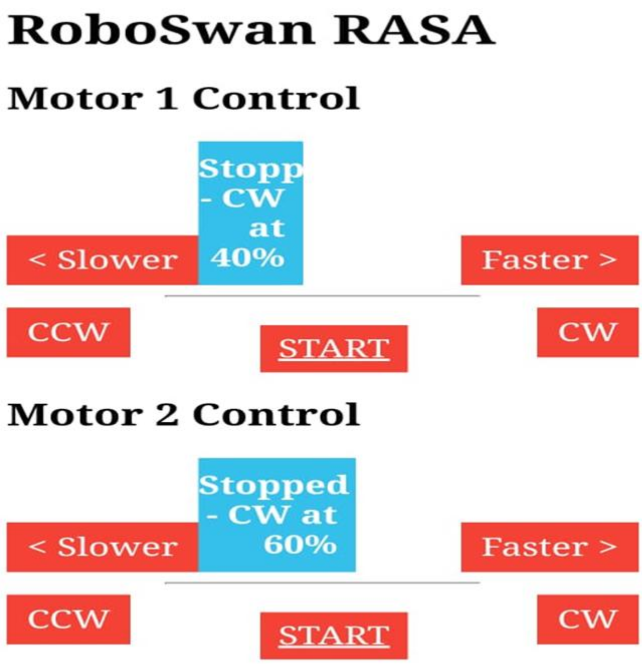


Figure 22:web page picture

Control:

1.For Forward Motion

Step1:

Click on start button to start the motor1.

Step2:

To set the direction click on clockwise button.

Step 3:

Select the speed by tapping on the slower or faster buttons.

Repeat same steps for motor 2.

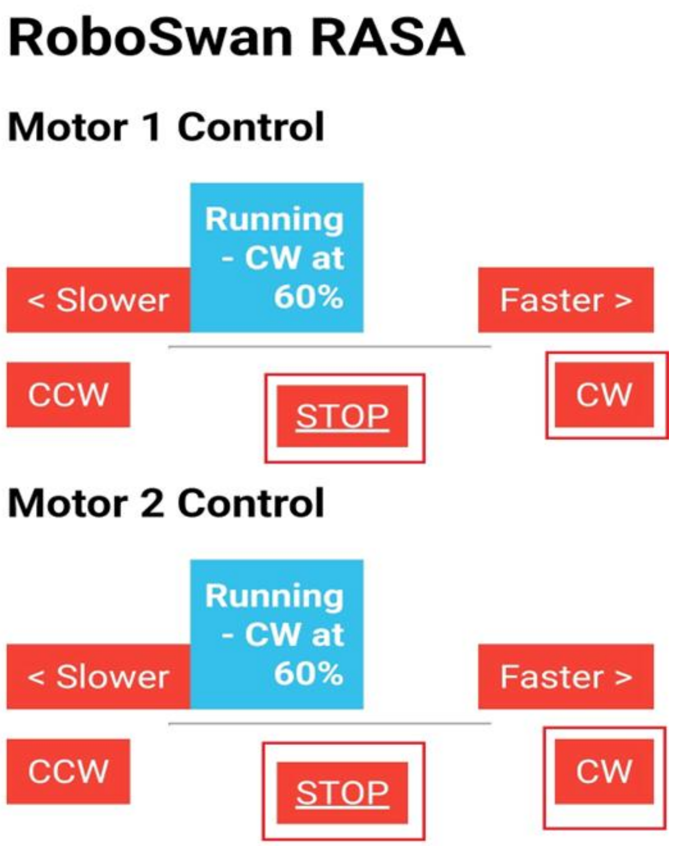


Figure 23:controls for forward motion.

2.For Backward Motion

Step1:

Click on start button to start the motor1.

Step2:

To set the direction click on Anticlock wise button**.**

Step 3:

Select the speed by tapping on the slower or faster buttons.

Repeat same steps for motor 2.

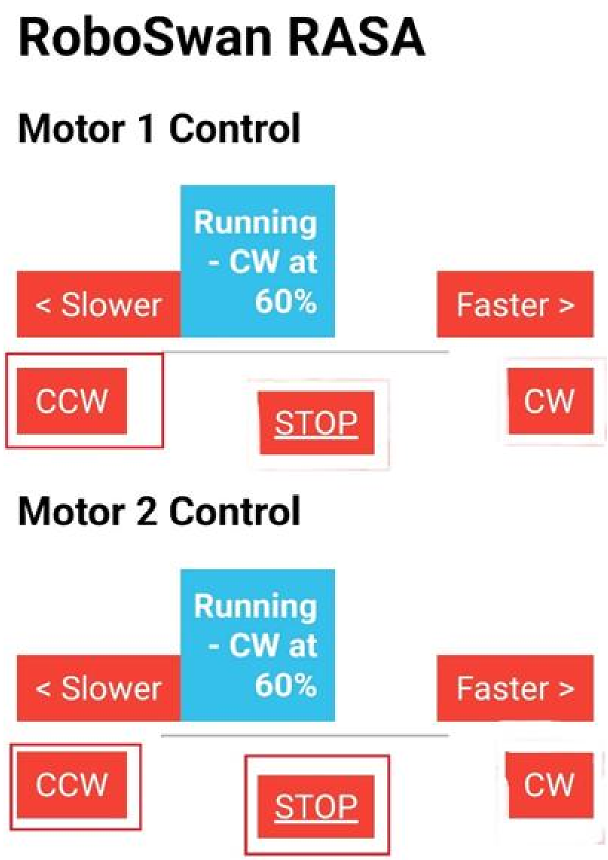


Figure 24: Control for backward motion.

3.To Take Right Turn

Step1:

click on stop button of motor2.

Step2:

click on start button of motor1.

Step3:

By keeping motor in clockwise direction.

Step4:

You can adjust the speed as per your need by tapping on faster or slower buttons.

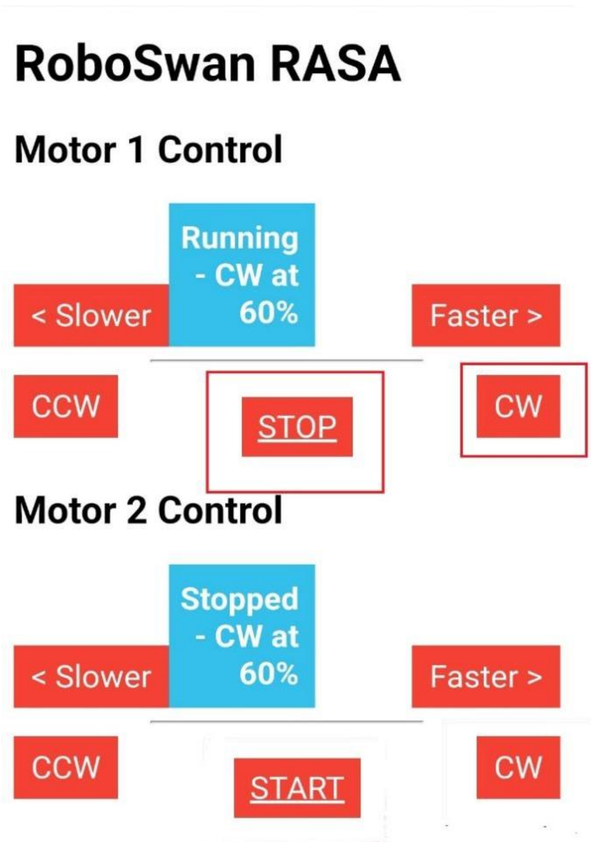


Figure 25: Control for right turn.

4.To Take Left Turn

Step1:

click on stop button of motor1.

Step2:

click on start button of motor2.

Step3:

By keeping motor in clockwise direction.

Step4:

You can adjust the speed as per your need by tapping on faster or slower buttons.

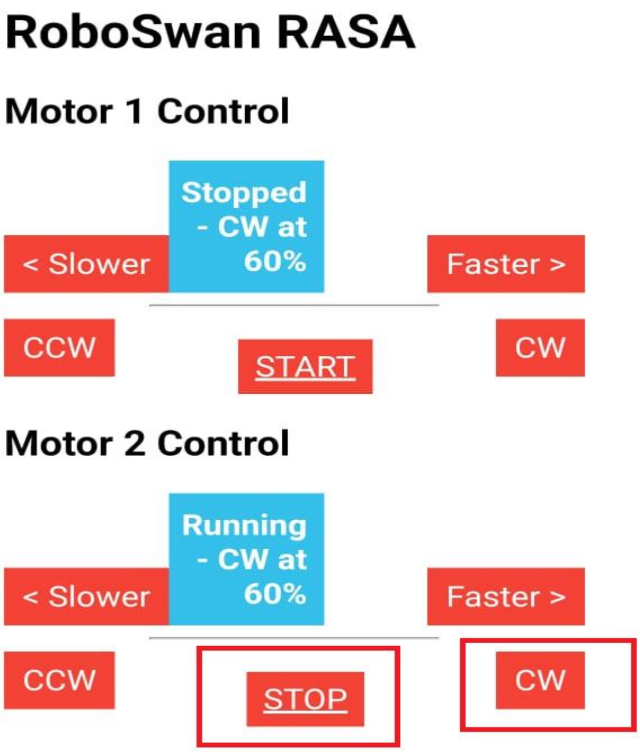


Figure 26: Control for left turn.

MODEL PIC:

Figure 27: Model picture.

FRONT VIEW:

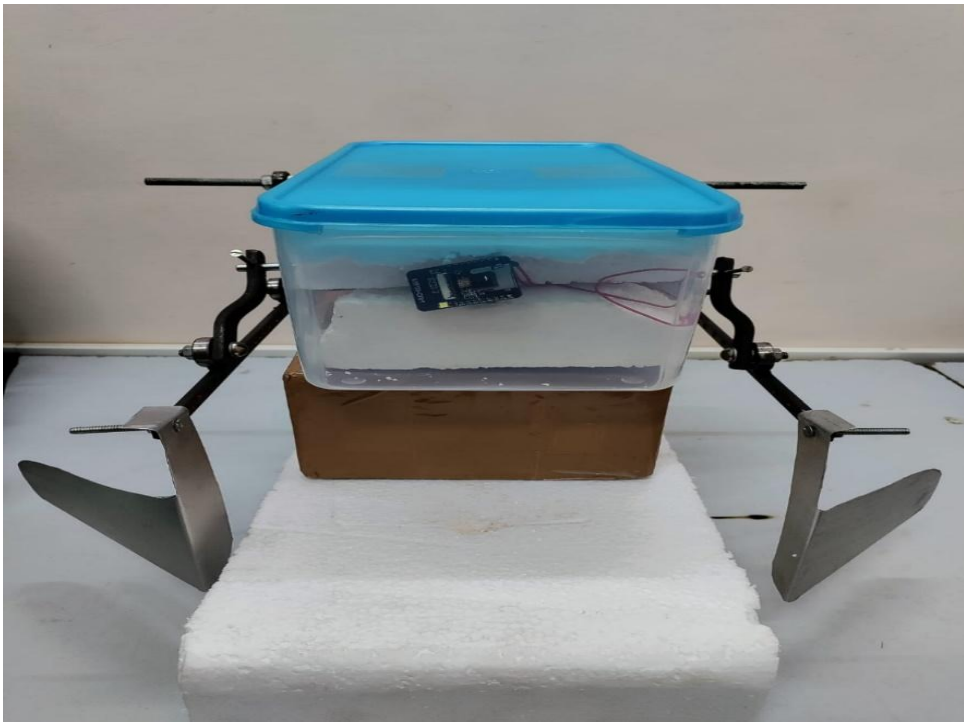


Figure 28: FRONT VIEW.

BACK VIEW:

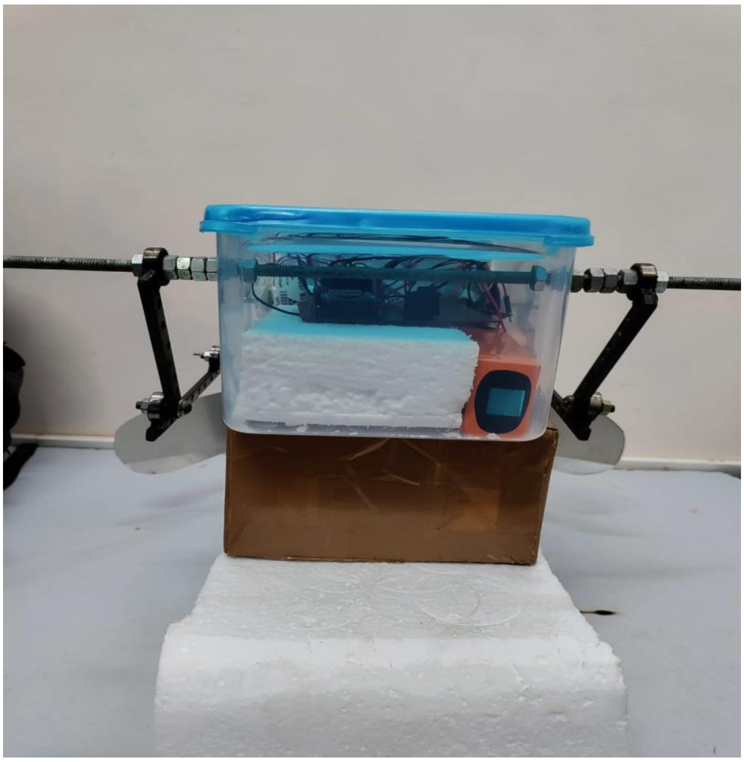


Figure 29: Back view

SIDE VIEW:

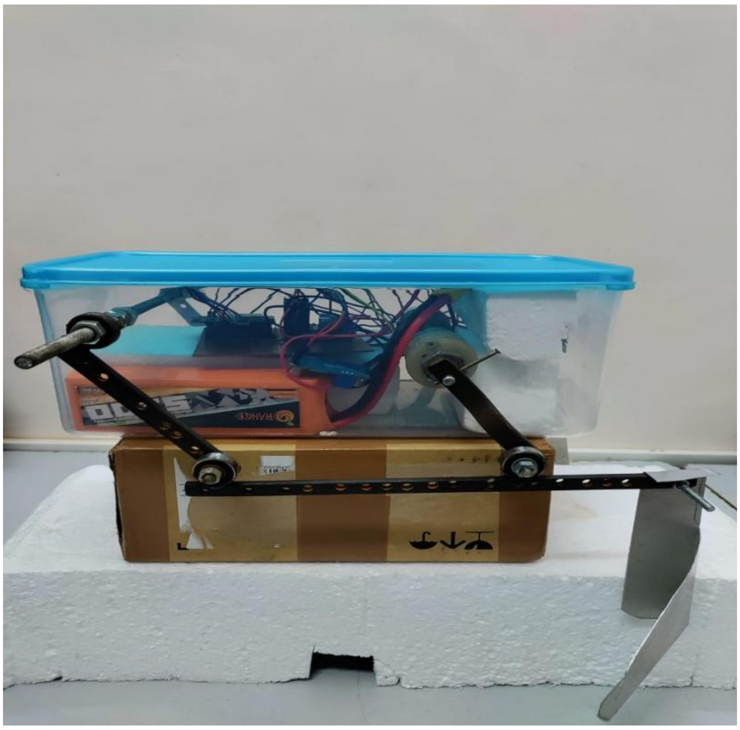


Figure 30: Side view

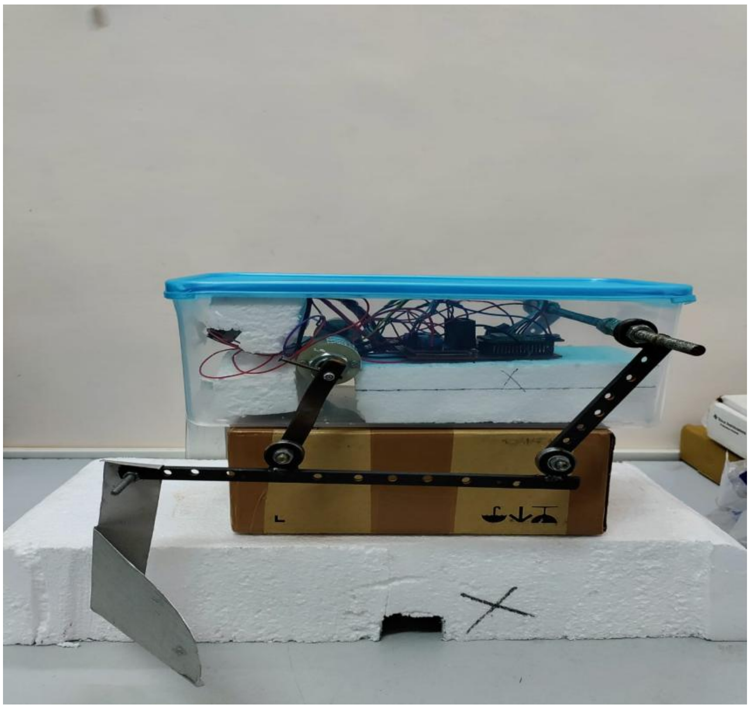


Figure 31: Side view

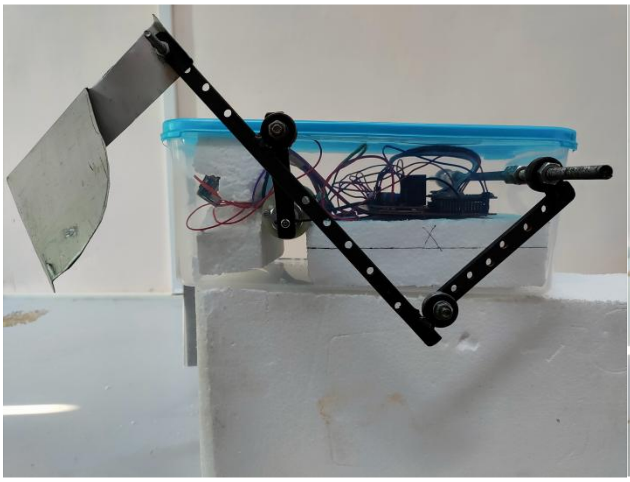
TOP VIEW:

Figure 32: Top view.

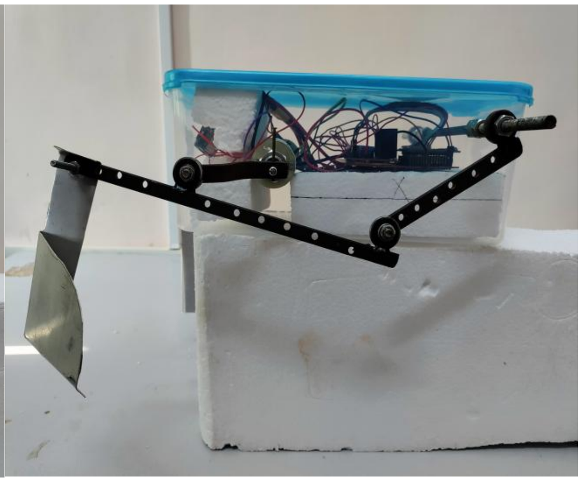
BOTTOM VIEW

Figure 33: Bottom view

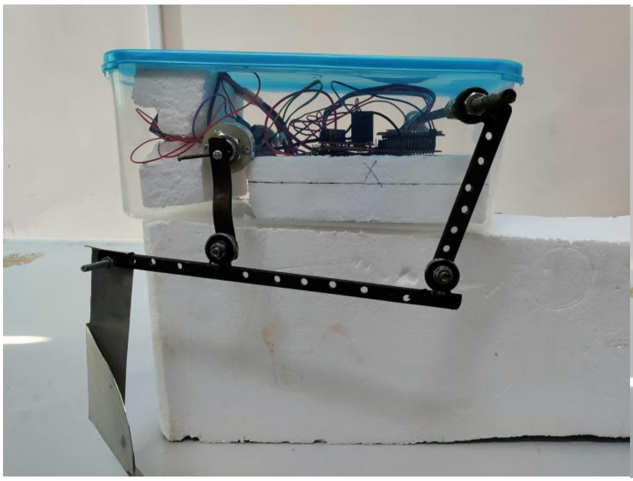
5.2 MECHANICAL LEG:



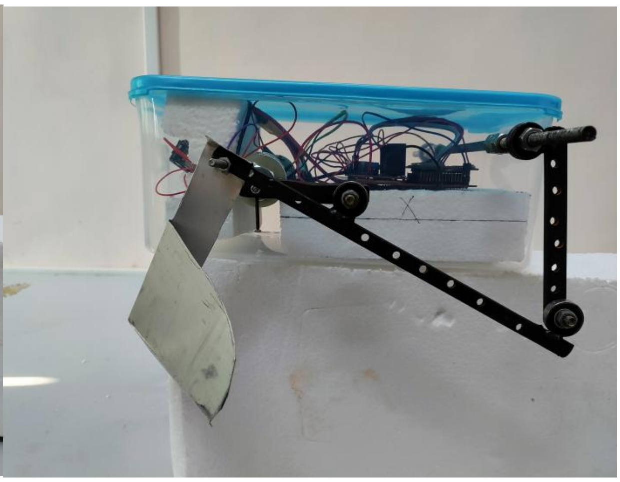
(1)



(2)



(2)



(4)

Figure 34: Demonstrate one cycle of mechanical leg.

5.3 OBSERVATION:

Model 1:

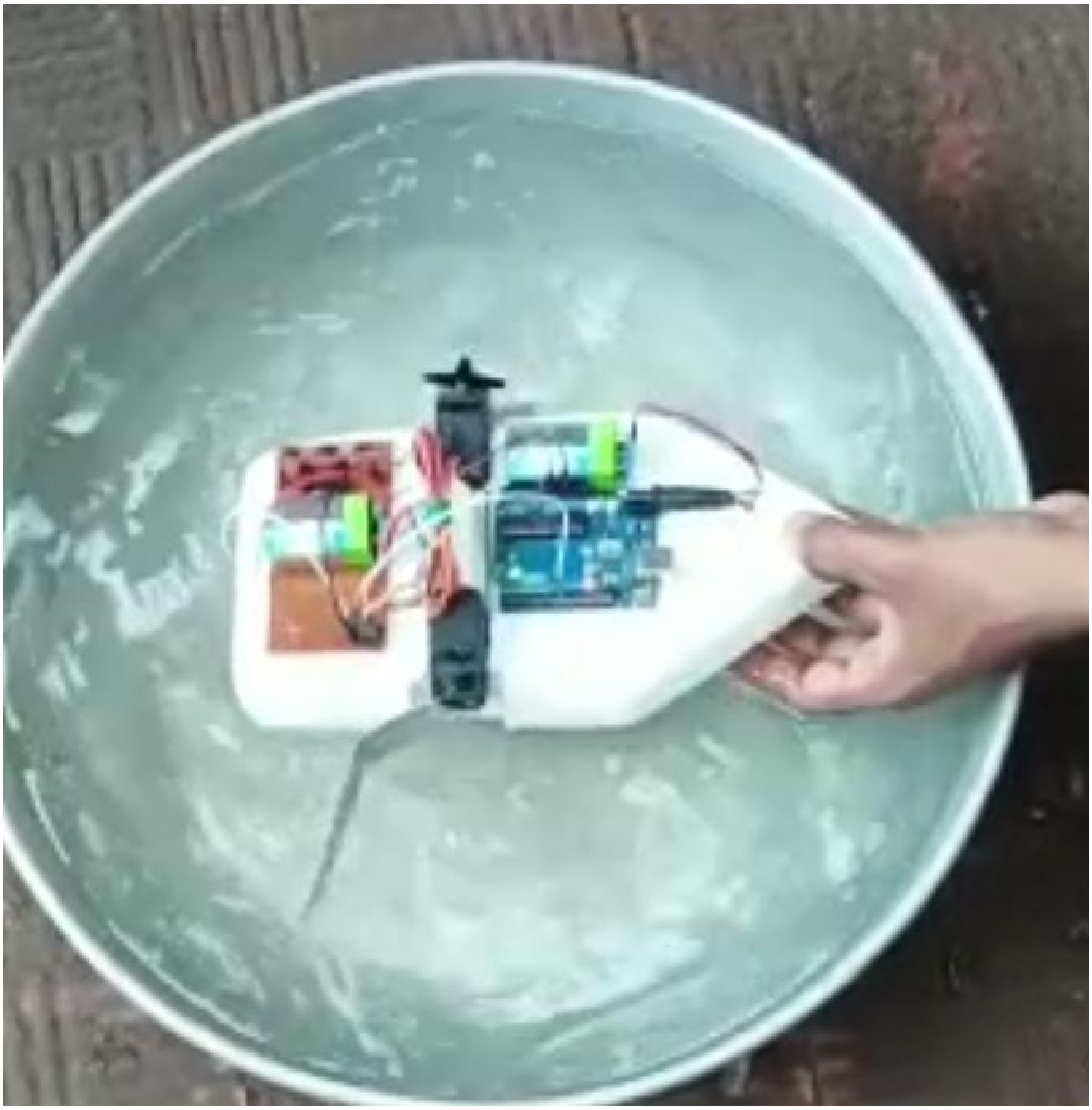


Figure 35: Model 1.1 picture.

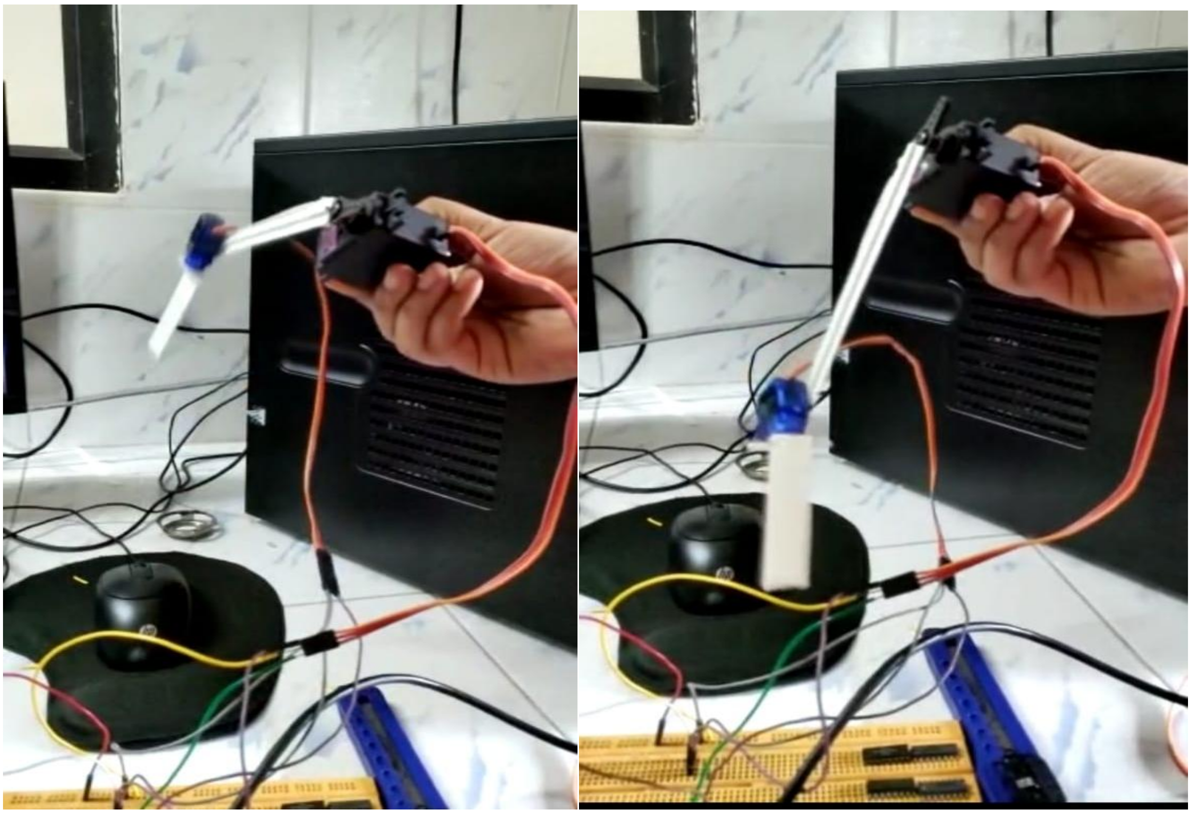


Figure 36: Model 1.2 picture

In our initial model, we utilized a servo motor and Arduino board, while the base was made of thermocol. An aluminum metallic flab was created, and control was established using an RC. However, during testing, we encountered several issues. Firstly, the flab failed due to a power issue with the servomotor, which was not able to handle the required power. Secondly, the material used for the legs was not suitable for use in water, and the overall weight of the device caused stability issues. Moreover, we faced problems with the small rechargeable battery, which only allowed for a run time of 5 minutes. Additionally, there was no provision for speed control or direction, and we attempted to use a stepper motor with no success.

MODEL 2:

Figure 37: Model 2 picture

The 2nd model was constructed using a metallic frame box as the primary material, and we replaced the servo motor with a brushless DC motor to create a mechanical leg. However, during testing, we encountered some problems. Firstly, we faced issues related to the brushless motor's high speed, which created problems for the mechanical leg and caused damage to its parts. We attempted to mitigate this issue by controlling the motor's speed, but it was not entirely successful. Secondly, we faced problems with the small rechargeable battery, which only allowed for a run time of 5 minutes. We attempted to use a larger battery, but it added to the model's overall weight, causing stability issues in the water.

To address the issues related to the mechanical leg, we experimented with different materials to create a more robust and durable design. We also explored the possibility of adding shock absorbers to minimize the impact of the brushless motor's high speed on the leg's parts. However, these attempts did not yield satisfactory results, and we concluded that a redesign of the mechanical leg would be necessary.

Moreover, we realized that the small rechargeable battery was not sufficient to power the device for extended periods, as it would require frequent recharging, which was inconvenient. To address this issue, we explored the option of using a more powerful battery that could provide longer run time. However, this approach resulted in increased weight, which caused stability issues in the water. We attempted to optimize the overall weight of the model by using lighter materials, but it affected the device's structural integrity.

In summary, our second model provided some improvements over the initial design, but we still faced significant challenges related to the mechanical leg's durability and the device's power source. We knew that we needed to explore new approaches to address these issues and create a more functional and reliable model.

MODEL 3:

The 3rd model was constructed using a plastic box as the primary material, and we replaced the servo motor with a brushless DC motor to create a mechanical leg. However, during testing, we encountered some problems. Firstly, we faced issues related to the brushless motor's high speed, which created problems for the mechanical leg and caused damage to its parts. We attempted to mitigate this issue by controlling the motor's speed, but it was not entirely successful. Secondly, we faced problems with the small rechargeable battery, which only allowed for a run time of 5 minutes. We attempted to use a larger battery, but it added to the model's overall weight, causing stability issues in the water.

To address the issues related to the mechanical leg, we experimented with different materials to create a more robust and durable design. We also explored the possibility of adding shock absorbers to minimize the impact of the brushless motor's high speed on the leg's parts. However, these attempts did not yield satisfactory results, and we concluded that a redesign of the mechanical leg would be necessary.

Moreover, we realized that the small rechargeable battery was not sufficient to power the device for extended periods, as it would require frequent recharging, which was inconvenient. To address this issue, we explored the option of using a more powerful battery that could provide longer run time. However, this approach resulted in increased weight, which caused stability issues in the water. We attempted to optimize the overall weight of the model by using lighter materials, but it affected the device's structural integrity.

In summary, our second model provided some improvements over the initial design, but we still faced significant challenges related to the mechanical leg's durability and the device's power source. We knew that we needed to explore new approaches to address these issues and create a more functional and reliable model.

MODEL 4:

For our fourth model, we set out specific requirements that included a 5200mAh battery, a DC motor, a mechanical leg made up of metal, an Arduino board, and an L298N motor driver. We used metallic material to make the frame for the model, which enhanced its durability and stability.

We replaced the small rechargeable battery used in the previous model with a more powerful DC motor. This change allowed us to increase the device's run time and improve its overall performance. Additionally, we incorporated an L298N motor driver to provide better control over the motor's speed and direction.

However, during testing, we encountered some issues related to the mechanical leg's functionality. The motor was not capable of providing the necessary torque required, which resulted in limitations on the model's movement in water. To address this issue, we explored various solutions, including using a more powerful motor or redesigning the mechanical leg.

We also faced challenges related to controlling the motor's speed and direction. While the L298N motor driver provided better control, we experienced some issues with its usability and compatibility with the Arduino board. We attempted to troubleshoot these issues by adjusting the code, but it did not yield satisfactory results.

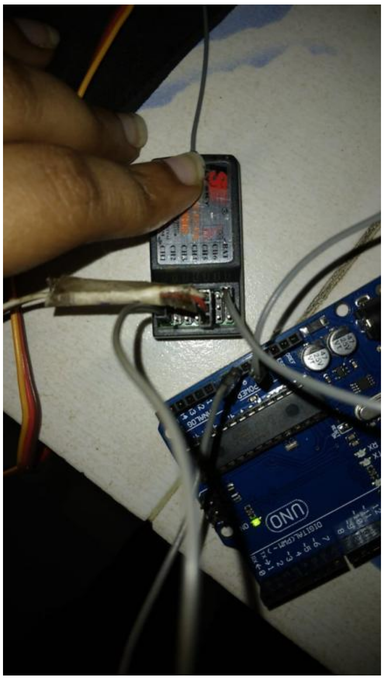


Figure 38: Model 4 picture

Overall, our fourth model represented a significant improvement over our previous designs, but we still faced significant challenges related to the mechanical leg's functionality and motor control. We knew that we needed to continue exploring new approaches and solutions to address these issues and create a more robust and reliable model.

MODEL 5:

Our fifth model was constructed using a thick plastic box, which enhanced the device's durability and resilience in water. We continued to use a DC motor, but we replaced the metallic material in the mechanical leg with plastic, which improved its efficiency and performance.

Despite these improvements, we still encountered some issues during testing, including problems with the motor's torque. We attempted to address this issue by experimenting with different motor types and sizes, but it remained a significant challenge.

Furthermore, we realized that the Arduino and RC used in previous models were not sufficient for this model's needs. Thus, we decided to shift to an ESP32 board, which offered an inbuilt Wi-Fi feature that enabled us to create a more robust and functional device. We used this feature to develop a web page that allowed for more precise control over the device's direction and speed.

To achieve this, we incorporated the L298N motor driver, which provided better control over the motor's speed and direction. We also explored the possibility of collecting data to make our model more powerful and efficient. This approach allowed us to gather valuable insights into the device's performance and optimize its design and functionality.

Overall, our fifth model represented a significant improvement over our previous designs, but we still faced challenges related to the motor's torque and control. We continued to explore new approaches and solutions to address these issues and create a more reliable and effective model.

MODEL 6:

Figure 39: Model 6 picture

The project involved the creation of a robotic leg that utilized a geared DC motor with a speed of 500 RPM, enclosed in a big thick plastic box. To ensure smooth movement, a well-designed metallic mechanical leg was built using bearings. To control the speed and direction of the motor, an L298N driver module was used. The control of the leg was implemented through a web page.

During testing, it was observed that although the geared DC motor with a speed of 500 RPM performed well, the torque generated was insufficient. Therefore, it was decided to upgrade the motor to a geared DC motor with a speed of 100 RPM, which would provide better torque and improve the overall performance of the robotic leg.

The decision to shift to a motor with a lower speed was made after careful consideration of the requirements and limitations of the project. With the new motor, the robotic leg would be able to generate the necessary torque and perform the required movements with greater efficiency. The team behind the project is optimistic about the improved performance of the upgraded robotic leg and is excited to continue testing and refining it further.

Figure 40: Model 6 testing in water.

MODEL 7:

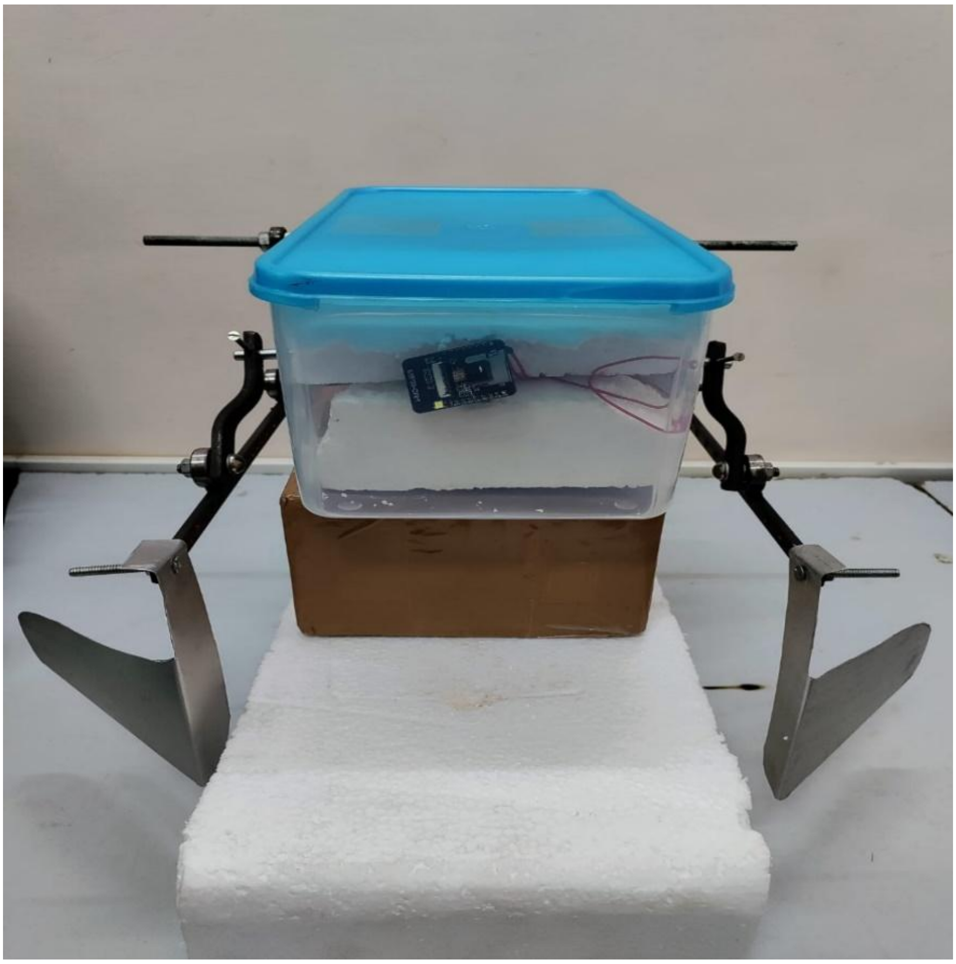


Figure 41: Model 7 picture

For the seventh and final model of the robotic leg, the team made several improvements from the previous versions. The geared DC motor with a speed of 100 RPM was used, which provided the necessary torque for the leg's movements. A smaller, thick plastic box was utilized to enclose the motor, reducing the overall size of the model and making it more practical for usage. The mechanical leg was well-designed and incorporated bearings to ensure smooth movement. An L298N driver module was used to control the speed and direction of the motor. The model was powered by a 5200mAh battery, which provided sufficient power for extended usage.

The team made use of an ESP32 microcontroller to control the leg's movements and collect data. For live video streaming, an ESP CAM with an FTDI programmer was used, which allowed for real-time monitoring of the leg's movements. A 12V to 5V converter was also incorporated to ensure that all components received the appropriate voltage.

To control the robotic leg, the team created an app that linked two web pages. This app allowed for the control of the leg's speed and direction, as well as the collection of temperature data. The temperature data was crucial in monitoring the leg's performance and ensuring that it did not overheat during usage.

Overall, the seventh model of the robotic leg was a significant improvement from the previous versions. The incorporation of better components, improved design, and increased functionality made it a highly practical and efficient tool for various applications. The team behind the project is excited about the potential uses and future developments for this model.

Figure 42: Model 7 picture

CHAPTER 6

CONCLUSION

In conclusion, the creation of the RoboSwan project was a challenging yet rewarding experience for the team involved. Through the development of seven different models, the team was able to identify and address various issues and limitations, resulting in the creation of a highly efficient and practical tool.

The final model of the RoboSwan incorporated several improvements, including the use of a geared DC motor with a speed of 100 RPM, a smaller plastic box, and the incorporation of an ESP32 microcontroller and an app for control and data collection. These improvements allowed for improved performance, increased functionality, and practicality.

The potential uses for the RoboSwan are numerous, ranging from assisting in oceanographic and industrial applications. The team behind the project is excited about the potential for future developments and applications for this technology.

Overall, the RoboSwan project demonstrated the importance of continuous improvement and adaptation in the field of robotics, and the value of collaborative efforts in achieving success.

RESULTS

The RoboSwan project resulted in the development of seven different models of a Roboswan. Each model incorporated different components and modifications, allowing the team to identify and address various issues and limitations. The final model of the Roboswan incorporated a geared DC motor with a speed of 100 RPM, a smaller plastic box, and an ESP32 micro controller with an app for control and data collection. Additionally, it had temperature monitoring as well as camera interfacing.

The team was able to successfully control and collect data on the movements of the robotic leg using the app. The temperature monitoring allowed for safe operation in various environments.

Through testing and analysis, it was found that the final model of the RoboSwan showed significant improvements in performance and functionality. The use of the geared DC motor and smaller plastic box allowed for more precise and efficient movement, while the incorporation of the ESP32 microcontroller and app allowed for easy control and data collection.

The potential uses for the robotic leg are numerous and varied, ranging from assisting in medical procedures to industrial applications. The robotic leg has the potential to be used in environments where human presence is difficult or dangerous, such as disaster zones or hazardous work sites.

The RoboSwan project was a challenging yet rewarding experience for the team involved. Through the development of seven different models, the team was able to identify and address various issues and limitations, resulting in the creation of a highly efficient and practical tool. The RoboSwan project demonstrated the importance of continuous improvement and adaptation in the field of robotics, and the value of collaborative efforts in achieving success.

The team is excited about the potential for future developments and applications for this technology.

Live streaming:

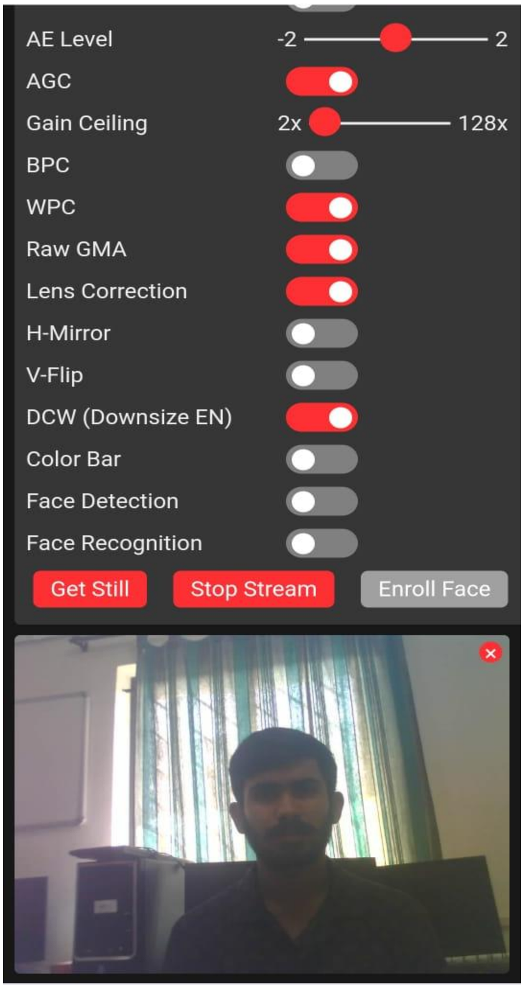


Figure 43: Live streaming

Final Web Page:

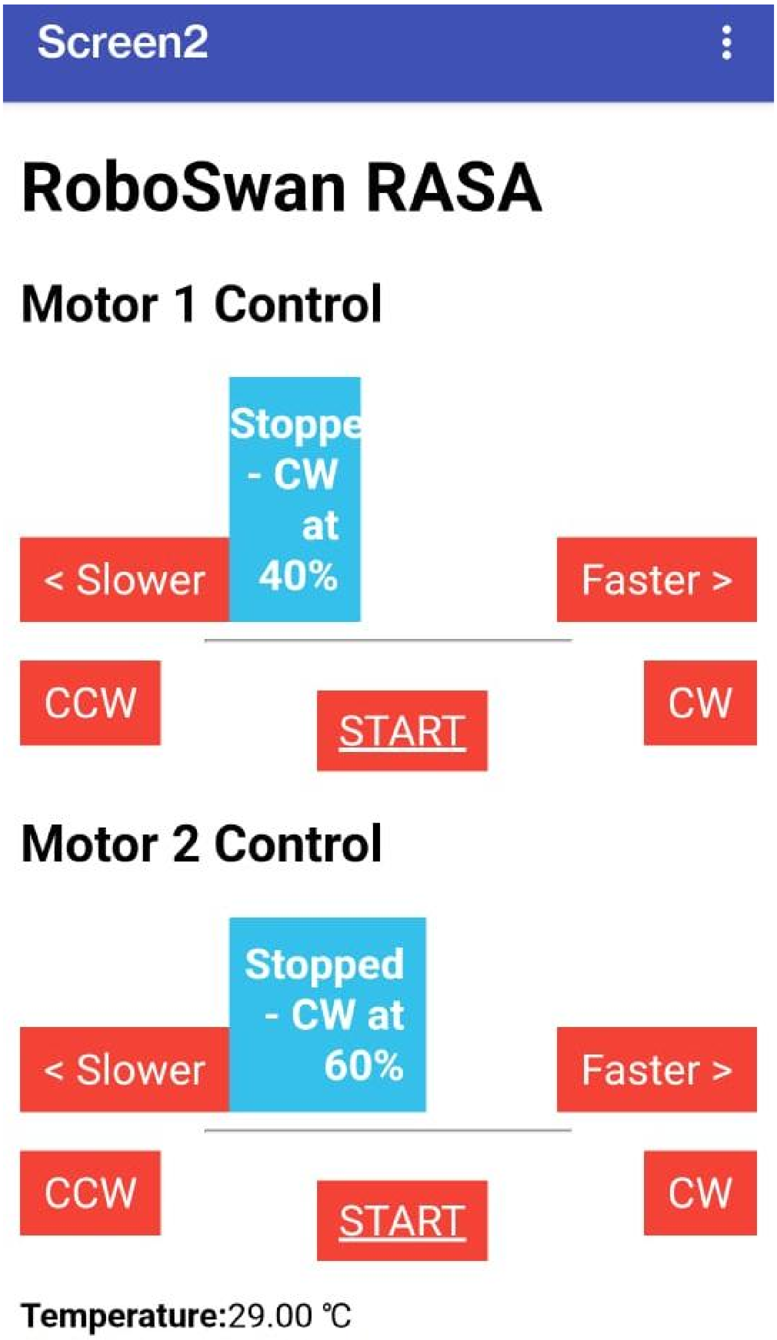


Figure 44: web page

FUTURE:

The RoboSwan can greatly benefit from the advancement of artificial intelligence (AI). By incorporating AI technologies, the RoboSwan can operate with increased efficiency and accuracy, making it more reliable for its intended use. AI can be used to identify patterns and trends in data, allowing the system to learn and optimize its performance over time. This can lead to more efficient energy utilization, route optimization, and smoother operation in general. Additionally, AI can be used to identify potential issues before they arise, allowing for preventative maintenance and minimizing the risk of equipment damage or failure.

Another crucial component of the RoboSwan is location tracking, which can be achieved using GPS modules. This allows the RoboSwan to navigate and follow a present route more accurately, while ensuring accurate tracking of its location. With an AI-powered roboSwan, the system can use GPS to take circular routes automatically or navigate along straight lines, optimizing the route for maximum efficiency. Additionally, the use of an AI and GPS combination can help the RoboSwan to avoid risks and obstacles that may be present in its path, allowing it to operate safely and effectively.

The RoboSwan model can also collect samples of water for scientific analysis using its collection mechanisms. This can help researchers study the effects of pollution, water quality, and marine life, among other things. With this feature, the RoboSwan can travel to different locations, collect water samples, and send the data back to scientists for analysis. Overall, the integration of AI, GPS tracking, and water sample collection technology can greatly enhance the performance and capabilities of the robotic model of the swan, making it an incredibly useful tool for a variety of industries and applications.

REFERENCE:

[1] Kolev, G.; Tayarani Bathaie,S.N.; Rybin, V.; Kulagin, M.; Karimov,T. Design of Small Unmanned Surface Vehicle with Autonomous NavigationSystem. Inventions 2021, 6, 91. https://doi.org/10.3390/ inventions6040091.

[2] Mr koay teong beng| NUSWAN : Swan robot to monitor water quality .Robotic labs in National University of Singapore (NUS).

[3] Niu, Hanlin; Lu, Yu; Savvaris, Al; Tsourdos, Antonios (2018). "An energy-efficient path planning algorithm for unmanned surface vehicles". Ocean Engineering. 161: 308–321. doi:10.1016/j.oceaneng.2018.01.025. hdl:1826/13249. S2CID 115280769. Archived from the original on 2022-06-15. Retrieved 2021-10-19.

[4] National Research Council, Division on Engineering and Physical Sciences (5 August 2005). Autonomous Vehicles in Support of Naval Operations. National Academies Press. ISBN 978-0-309-18123-5. Retrieved 15 October 2019.

[5] He, Ying; Wang, Dao Bo; Ali, Zain Anwar (2020-11-01). "A review of different designs and control models of remotely operated underwater vehicle". Measurement and Control. 53 (9–10): 1561–1570. doi:10.1177/0020294020952483. ISSN 0020-2940. S2CID 225252081.

[6] "Autonomous Underwater Vehicles | AUV Technology | Underwater Gliders". Unmanned Systems Technology. Retrieved 2021-11-04.

[7] "Autonomous Underwater Vehicles | AUV Technology | Underwater Gliders". Unmanned Systems Technology. Retrieved 2021-11-04.

[8] Kear, Janet, ed. (2005). Ducks, Geese and Swans. Bird Families of the World. Oxford: Oxford University Press. ISBN 978-0-19-861008-3.

[9] Department of Automatic Control Engineering, Feng Chia University (FCU), No. 100, Wenhwa Road, Seatwen, Taichung 40724, Taiwan. Autonomous Water Quality Monitoring and Water Surface Cleaning for Unmanned Surface Vehicle. 2021 Feb 5;21(4):1102.

[10] Jiayuan Zhuanga, b, c , Lei Zhanga,b1 , Bo Wanga, b , Yumin Sua, b , Hanbing Sunb , Yuanchang Liuc2 and Richard Bucknallc| Navigating high-speed unmanned surface vehicles: system approach and validations| a. Science and Technology on Underwater Vehicle Laboratory, Harbin Engineering University, Harbin, 150001, China b. College of Shipbuilding Engineering, Harbin Engineering University, Harbin, 150001, China c. Department of Mechanical Engineering, University College London, Torrington Place, London WC1E 7JE, UK.

[11] Xiaojie Sun [ID], Guofeng Wang *, Yunsheng Fan, Dongdong Mu [ID] and Bingbing Qiu | An An Automatic Navigation System for Unmanned Surface Vehicles in Realistic Sea Environment| School of Marine Electrical Engineering, Dalian Maritime University, Dalian 116026, China.

[12] Matthew Dunbabin's,alistar grinham,james udy |An autonomous surface vehicle for water quality monitoring |autonomous systems laboratory, csiro ict centre, university of queensland..

LINK:

1. https://en.wikipedia.org/wiki/Swan
2. https://discovery.ucl.ac.uk/id/eprint/10117148/1/
3. https://www.mdpi.com/2076-3417/8/2/193
4. https://www.electronicscomp.com/l298-2amp-motor-driver-module-with-pwm-control-india?gclid=Cj0KCQjwxMmhBhDJARIsANFGOStyAvJ-1qll72MX-_1LC9du21FoDdZEfkgpOkmoaJpd13uWqNUcpAgaAvqfEALw_wcB
5. https://www.electronicscomp.com/100-rpm-centre-shaft-dc-geared-motor-india
6. https://robu.in/product/orange-5200mah-4s-40c-lithium-polymer-battery-pack-lipo/?gclid=Cj0KCQjwxMmhBhDJARIsANFGOSsgtg0TtXt9ziuL9Nz7WzGPZl_VvUasKamWknES3L5e0UeR8SrK2hgaAqirEALw_wcB
7. https://www.electronicscomp.com/esp32-development-board-with-wifi-bluetooth-india?gclid=CjwKCAjw586hBhBrEiwAQYEnHV1Oz-OocX1aSF7GChS4sgIOuqGuP-WVRdFvtpS4NR9MRkVCQyNWhoCuIUQAvD_BwE
8. https://components101.com/modules/esp32-cam-camera-module
9. https://www.electronicscomp.com/ds18b20-water-proof-temperature-sensor-probe-india
10. https://www.electronicscomp.com/ft232rl-usb-ttl-serial-adaptor-module-for-arduino?gclid=CjwKCAjwrpOiBhBVEiwA_473dCoreQKgvl0YOYuePweyo8LRjFUgZUhstnliEu9g9DZpsqw3RkHoNxoCDo8QAvD_BwE

REFERENCE FOR IMAGE:

1. FIGURE1:https://swanlovers.net/wp-content/uploads/2016/10/MUTE-COB-Skeleton-2STIT.jpg
2. FIGURE2: https://www.frontiersin.org/articles/10.3389/fbioe.2021.676894/full
3. FIGURE3: https://depositphotos.com/stock-photos/webbed-feet.html
4. FIGURE 4: https://www.dkfindout.com/us/animals-and-nature/birds/swans/
5. FIGURE 5: https://www.pinterest.com/pin/389702174012362796/
6. FIGURE 6 : https://nus.edu.sg/neri/research/nuswan/
7. FIGURE7: https://www.electronicscomp.com/l298-2amp-motor-driver-module-with-pwm-control-india?gclid=Cj0KCQjwxMmhBhDJARIsANFGOStyAvJ-1qll72MX-_1LC9du21FoDdZEfkgpOkmoaJpd13uWqNUcpAgaAvqfEALw_wcB
8. FIGURE 8: : https://www.electronicscomp.com/l298-2amp-motor-driver-module-with-pwm-control-india?gclid=Cj0KCQjwxMmhBhDJARIsANFGOStyAvJ-1qll72MX-_1LC9du21FoDdZEfkgpOkmoaJpd13uWqNUcpAgaAvqfEALw_wcB
9. FIGURE9: : https://www.electronicscomp.com/l298-2amp-motor-driver-module-with-pwm-control-india?gclid=Cj0KCQjwxMmhBhDJARIsANFGOStyAvJ-1qll72MX-_1LC9du21FoDdZEfkgpOkmoaJpd13uWqNUcpAgaAvqfEALw_wcB
10. FIGURE 10: : https://www.electronicscomp.com/l298-2amp-motor-driver-module-with-pwm-control-india?gclid=Cj0KCQjwxMmhBhDJARIsANFGOStyAvJ-1qll72MX-_1LC9du21FoDdZEfkgpOkmoaJpd13uWqNUcpAgaAvqfEALw_wcB
11. FIGURE 11: https://www.electronicscomp.com/100-rpm-centre-shaft-dc-geared-motor-india
12. FIGURE 12: https://robu.in/product/orange-5200mah-4s-40c-lithium-polymer-battery-pack-lipo/?gclid=Cj0KCQjwxMmhBhDJARIsANFGOSsgtg0TtXt9ziuL9Nz7WzGPZl_VvUasKamWknES3L5e0UeR8SrK2hgaAqirEALw_wcB

13. FIGURE 13: https://www.electronicscomp.com/esp32-development-board-with-wifi-bluetooth-india?gclid=CjwKCAjw586hBhBrEiwAQYEnHV1Oz-OocX1aSF7GChS4sgIOuqGuP-WVRdFvtpS4NR9MRkVCQyNWhoCuIUQAvD_BwE

14. FIGURE14 : https://www.electronicscomp.com/esp32-development-board-with-wifi-bluetooth-india?gclid=CjwKCAjw586hBhBrEiwAQYEnHV1Oz-OocX1aSF7GChS4sgIOuqGuP-WVRdFvtpS4NR9MRkVCQyNWhoCuIUQAvD_BwE

15. FIGURE 15: https://components101.com/modules/esp32-cam-camera-module

16. FIGURE 16: https://components101.com/modules/esp32-cam-camera-module

17. FIGURE 17 : : https://www.electronicscomp.com/ft232rl-usb-ttl-serial-adaptor-module-for-arduino?gclid=CjwKCAjwrpOiBhBVEiwA_473dCoreQKgvl0YOYuePweyo8LRjFUgZUhstnliEu9g9DZpsqw3RkHoNxoCDo8QAvD_BwE

18. FIGURE 18: https://www.electronicscomp.com/ft232rl-usb-ttl-serial-adaptor-module-for-arduino?gclid=CjwKCAjwrpOiBhBVEiwA_473dCoreQKgvl0YOYuePweyo8LRjFUgZUhstnliEu9g9DZpsqw3RkHoNxoCDo8QAvD_BwE

19. FIGURE 19: https://www.electronicscomp.com/ft232rl-usb-ttl-serial-adaptor-module-for-arduino?gclid=CjwKCAjwrpOiBhBVEiwA_473dCoreQKgvl0YOYuePweyo8LRjFUgZUhstnliEu9g9DZpsqw3RkHoNxoCDo8QAvD_BwE

20. FIGURE 20: https://www.electronicscomp.com/ds18b20-water-proof-temperature-sensor-probe-india

APPENDIX

Code:Controlling DC Motor Using Wifi

```
#include <Robo_L298N_DC_motor.h>

// motor 1 settings

#define CHA 0

#define ENA 19

#define IN2 5


// motor 2 settings

#define IN3 17

#define IN4 16

#define ENB 4

#define CHB 1

#define ONE_WIRE_BUS 13


const int CCW = 2;

const int CW  = 1;


#define motor1 1 // do not change

#define motor2 2 // do not change


Robo_L298N_DC_motor motor(IN1, IN2, ENA, CHA, IN3, IN4, ENB, CHB, true);


int       motor1Direction = CW;//default direction of rotation

const int motor1changeStep = 10;// 10 is 10% every time button is pushed

int       motor1Speed = 40;// variable holding the light output vlaue (initial value) 40 means 40%

const int motor1MinimumSpeed=20;

const int motor1MaximumSpeed=100;

int       motor1StopState=HIGH;//Stope state of motor (HIGH means STOP) and LOW means Start


int       motor2Direction = CW;//default direction of rotation

const int motor2changeStep = 10;// 10 is 10% every time button is pushed

int       motor2Speed = 60;// variable holding the light output vlaue (initial value) 40 means 40%
```

```
const int motor2MinimumSpeed=20;
const int motor2MaximumSpeed=100;
int       motor2StopState=HIGH;//Stope state of motor (HIGH means STOP) and LOW means Start


#include "ESP32_L298N_DC_motor_wifi_page.h"



#include <WiFi.h>
#include <WiFiClient.h>
#include <WebServer.h>
#include <ESPmDNS.h>
#include <OneWire.h>
#include <DallasTemperature.h>
#include <TinyGPSPlus.h>
#include <HardwareSerial.h>



OneWire oneWire(ONE_WIRE_BUS);

// Pass our oneWire reference to Dallas Temperature sensor
DallasTemperature sensors(&oneWire);

// TinyGPSPlus gps;

const char *ssid = "RASA";
const char *password = "12345679";

WebServer server(80);

const int led = 13;

void handleRoot() {

String HTML_page = motorControlHeader_1;
```

```
HTML_page.concat(".bar1 {width: " + String(motor1Speed)  + "%;}\n");
//HTML_page.concat(motor1Speed);
//HTML_page.concat("%;}");
HTML_page.concat(".bar2 {width: " + String(motor2Speed) + "%;}\n");

HTML_page.concat(motorControlHeader_2);

HTML_page.concat(motor1Control_p1);

if(motor1Direction ==CW)
{
     if(motor1StopState ==HIGH)
     {
       HTML_page.concat("<strong>Stopped - CW at ");
     }else{
       HTML_page.concat("<strong>Running - CW at ");
     }
}else{
     if(motor1StopState ==HIGH)
     {
       HTML_page.concat("<strong>Stopped - CCW at ");
     }else{
       HTML_page.concat("<strong>Running - CCW at ");
     }
}
HTML_page.concat(motor1Speed);
HTML_page.concat(motor1Control_p2);
if(motor1StopState ==HIGH)
{
  HTML_page.concat("m1START\">START");
}else{
  HTML_page.concat("m1STOP\">STOP");
}
HTML_page.concat(motor1Control_p3);

///motor 2 begins
```

```
HTML_page.concat(motor2Control_p1);


if(motor2Direction ==CW)
{
     if(motor2StopState ==HIGH)
     {
       HTML_page.concat("<strong>Stopped - CW at ");
     }else{
       HTML_page.concat("<strong>Running - CW at ");
     }
}else{
     if(motor2StopState ==HIGH)
     {
       HTML_page.concat("<strong>Stopped - CCW at ");
     }else{
       HTML_page.concat("<strong>Running - CCW at ");
     }
}
HTML_page.concat(motor2Speed);
HTML_page.concat(motor2Control_p2);
if(motor2StopState ==HIGH)
{
  HTML_page.concat("m2START\">START");
}else{
  HTML_page.concat("m2STOP\">STOP");
}
HTML_page.concat(motor2Control_p3);
HTML_page.concat("<p><b>Temperature:</b>");
HTML_page.concat(readDHTTemperature());
HTML_page.concat("<span> &#8451;</span></p>");




HTML_page.concat("</body>\n</html>");


 server.send(200, "text/html", HTML_page);
```

```
}

void handleNotFound() {
  digitalWrite(led, 1);
  String message = "File Not Found\n\n";
  message += "URI: ";
  message += server.uri();
  message += "\nMethod: ";
  message += (server.method() == HTTP_GET) ? "GET" : "POST";
  message += "\nArguments: ";
  message += server.args();
  message += "\n";

  for (uint8_t i = 0; i < server.args(); i++) {
    message += " " + server.argName(i) + ": " + server.arg(i) + "\n";
  }

  server.send(404, "text/plain", message);
  digitalWrite(led, 0);
}

void setup(void) {
  Serial.begin(115200);
  motor.begin();
  //L298N DC Motor by Robojax.com

  // Start serial communication for debugging purposes
  Serial.begin(9600);
  // Start up the library
  sensors.begin();

  // Serial2.begin(4800, SERIAL_8N1, 16, 17);

  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
```

```
  Serial.println("");



  // Wait for connection
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println("");
  Serial.print("Connected to ");
  Serial.println(ssid);
  Serial.print("IP address: ");
  Serial.println(WiFi.localIP());

  if (MDNS.begin("robojaxESP32")) {
    Serial.print("MDNS responder started at http://");
    Serial.println("robojaxESP32");
  }

  server.on("/", handleRoot);
  server.on("/speed", HTTP_GET, handleMotorSpeed);
  server.on("/direction", HTTP_GET, handleMotorDirection);
  server.on("/stop", HTTP_GET, handleMotorBrake);
  server.onNotFound(handleNotFound);
  server.begin();
  Serial.println("HTTP server started");
}

void loop(void) {
  server.handleClient();
  if(motor1StopState ==HIGH)
  {
    motor.brake(motor1);
```

```
  }else{
     motor.rotate(motor1, motor1Speed, motor1Direction);//run motor1 at motor1Speed% speed in
motor1Direction
  }

  if(motor2StopState ==HIGH)
  {
    motor.brake(motor2);

  }else{
     motor.rotate(motor2, motor2Speed, motor2Direction);//run motor2 at motor2Speed% speed in
motor2Direction
  }

  delay(100);
}

void handleMotorSpeed() {
  if(server.arg("do") == "m1slower" )
  {
    motor1Speed -=motor1changeStep;

      if(motor1Speed < motor1MinimumSpeed)
      {
        motor1Speed = motor1MinimumSpeed;
      }
  }else if(server.arg("do") == "m1faster")
  {
    motor1Speed +=motor1changeStep;

      if(motor1Speed > motor1MaximumSpeed)
      {
        motor1Speed =motor1MaximumSpeed;
      }
  }else if(server.arg("do") == "m2slower")
  {
```

```
    motor2Speed -=motor2changeStep;

      if(motor2Speed < motor2MinimumSpeed)
      {
        motor2Speed = motor2MinimumSpeed;
      }
  }else if(server.arg("do") == "m2faster")
  {
    motor2Speed +=motor2changeStep;

      if(motor2Speed > motor2MaximumSpeed)
      {
        motor2Speed =motor2MaximumSpeed;
      }
  }else{
    motor1Speed =0;

  }

  handleRoot();
}//handleMotorSpeed() end

void handleMotorDirection() {
  if(server.arg("dir") == "m1CW")
  {
    motor1Direction =CW;

  }else if(server.arg("dir") == "m1CCW")
  {
    motor1Direction =CCW;

  }else if(server.arg("dir") == "m2CW")
  {
    motor2Direction =CW;
```

```
  }else{
    motor2Direction =CCW;


  }


  handleRoot();
}//



void handleMotorBrake() {
  if(server.arg("do") == "m1START")
  {
      motor1StopState=LOW;
  }else if(server.arg("do") == "m1STOP")
  {
      motor1StopState=HIGH;
  }else if(server.arg("do") == "m2START")
  {
      motor2StopState=LOW;
  }else{
      motor2StopState=HIGH;
  }
  handleRoot();
}//


float readDHTTemperature() {
  sensors.requestTemperatures();
  float tempInCelsius = sensors.getTempCByIndex(0);
  return tempInCelsius;
}
```

Code:HTML WEB PAGE

```
const char motorControlHeader_1[]  PROGMEM = R"header1(
<!DOCTYPE html>
<html>
<head>
<title>RoboSwan RASA </title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
* {box-sizing: border-box}
.table{
    width:100%;
    display:table;
}
.row{
    display:table-row;
}

.fixedCell {
    width:15%;
    display:table-cell;


}


.cell{
    display: table-cell;
    background: green;
}


.progress_bar {
  font-size: 20px;
  text-align: right;
  padding-top: 10px;
  padding-bottom: 10px;
  padding-right:10px;
```

```
  color: white;
  float:left;
  background-color:#34c0eb;
}

.buttonsDiv {
  display: flex;
  justify-content: center;
  float:auto;
}
.startStop{
     font-size: 20px;
   background-color: #f44336;
    color: #ffffff;
    border-color: #f44336;
    border: none;
    display: inline-block;
    padding: 7px 10px;
    vertical-align: middle;
}

)header1";


const char motorControlHeader_2[] PROGMEM = R"roboSpeed2(
.nextprev a.rj-right, .nextprev a.rj-left {
    background-color: #f44336;
    color: #ffffff;
    border-color: #f44336;
}
.nextprev a {
    font-size: 20px;
    border: 1px solid #cccccc;
}
.rj-right {
    float: right!important;
```

```
}
.rj-left {
    float: left!important;
}
.rj-btn, .rj-button {
    border: none;
    display: inline-block;
    padding: 7px 10px;
    vertical-align: middle;
    overflow: hidden;
    text-decoration: none;
    color: inherit;
    background-color: inherit;
    text-align: center;
    cursor: pointer;
    white-space: nowrap;
}
</style>
</head>
<body>

<h1>RoboSwan RASA</h1>)roboSpeed2";

////////////////for motor 1 part 1///////////////
const char motor1Control_p1[] PROGMEM = R"roboSpeed3(
<h2>Motor 1 Control</h2>
<div class="table">
    <div class="row">
        <div class="fixedCell">
            <div class="nextprev">
            <a class="rj-left rj-btn" href="/speed?do=m1slower">< Slower</a>
            </div>

        </div><!--fixedCell -->
```

```
        <div class="progress_bar bar1">)robojaxSpeed3";


////////////////for motor 1 part 2///////////////
const char motor1Control_p2[] PROGMEM = R"roboSpeed4(%</strong></div>


        <div class="fixedCell">


          <div class="nextprev">
          <a class="rj-right rj-btn" href="/speed?do=m1faster">Faster ></a>
          </div>


        </div><!--fixedCell -->
    </div><!--row -->


</div><!--table -->
<hr align="center" width="50%">
<div class="table">
    <div class="row">
        <div class="fixedCell">
            <div class="nextprev">
            <a class="rj-left rj-btn" href="/direction?dir=m1CCW">CCW</a>
            </div>


        </div><!--fixedCell -->


        <div class="buttonsDiv">
            <a class="startStop" href="/stop?do=)robojaxSpeed4";


 ////////////////for motor 1 part 3///////////////
const char motor1Control_p3[] PROGMEM = R"roboSpeed5(
            </a>
        </div>


        <div class="fixedCell">
```

```
            <div class="nextprev">
            <a class="rj-right rj-btn" href="/direction?dir=m1CW">CW</a>
            </div>

          </div><!--fixedCell -->
      </div><!--row - Robojax.com -->

</div><!-- ---- table motor 1 ended -->)roboSpeed5";

///////////////// motor 1 end******************************************
//////////////////////*************************************

////////////////for motor 2 part 1///////////////
const char motor2Control_p1[] PROGMEM = R"roboSpeed3(
<!-- ---- motor 2 started here ---- -->
<h2>Motor 2 Control</h2>

<div class="table">
    <div class="row">
        <div class="fixedCell">
            <div class="nextprev">
            <a class="rj-left rj-btn" href="/speed?do=m2slower">< Slower</a>
            </div>

        </div><!--fixedCell -->
        <div class="progress_bar bar2">)roboSpeed3";

////////////////for motor 2 part 2///////////////
const char motor2Control_p2[] PROGMEM = R"roboSpeed4(%</strong></div>

        <div class="fixedCell">

          <div class="nextprev">
```

```
          <a class="rj-right rj-btn" href="/speed?do=m2faster">Faster ></a>
          </div>

        </div><!--fixedCell -->
    </div><!--row -->

</div><!--table -->
<hr align="center" width="50%">
<div class="table">
    <div class="row">
        <div class="fixedCell">
            <div class="nextprev">
            <a class="rj-left rj-btn" href="/direction?dir=m2CCW">CCW</a>
            </div>

        </div><!--fixedCell -->
        <div class="buttonsDiv">
            <a class="startStop" href="/stop?do=)roboSpeed4";

 ////////////////for motor 2 part 3///////////////
const char motor2Control_p3[] PROGMEM = R"roboSpeed5(
            </a>
        </div>

        <div class="fixedCell">

          <div class="nextprev">
          <a class="rj-right rj-btn" href="/direction?dir=m2CW">CW</a>
          </div>

        </div><!--fixedCell -->
    </div><!--row - Robo.com -->

</div><!--table -->)roboSpeed5";
```