STERIBOT

YOUR GUARDIAN OF PURITY IN THE BATTLE OF GERMS

TEAM MEMBERS

FARIDA AMR	HANYA HOSSAM	SALMA AMR
20P6363	20P3687	20P4526
AHMED HASSABOU	MOSTAFA SALEH	MOHAMED MOHANNA
19P4007	20P4538	19P6218

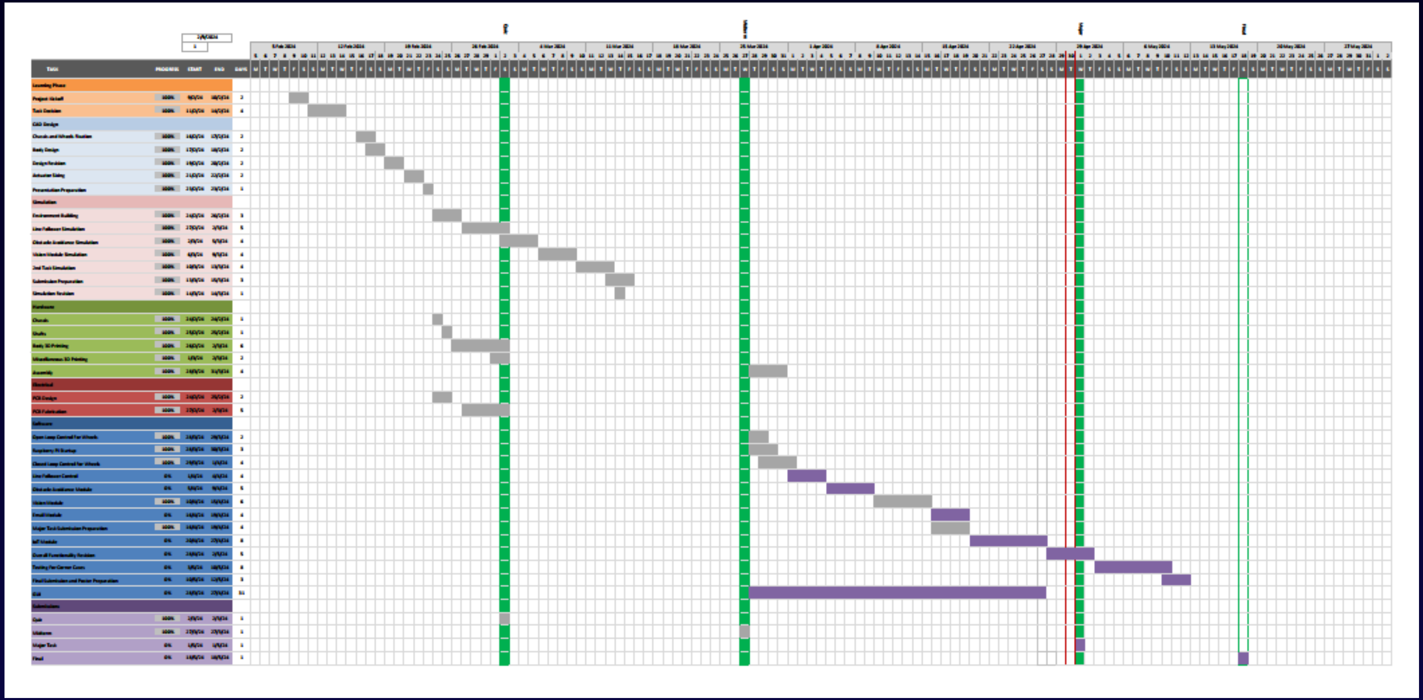
01

Project Plan

Project Gantt Chart

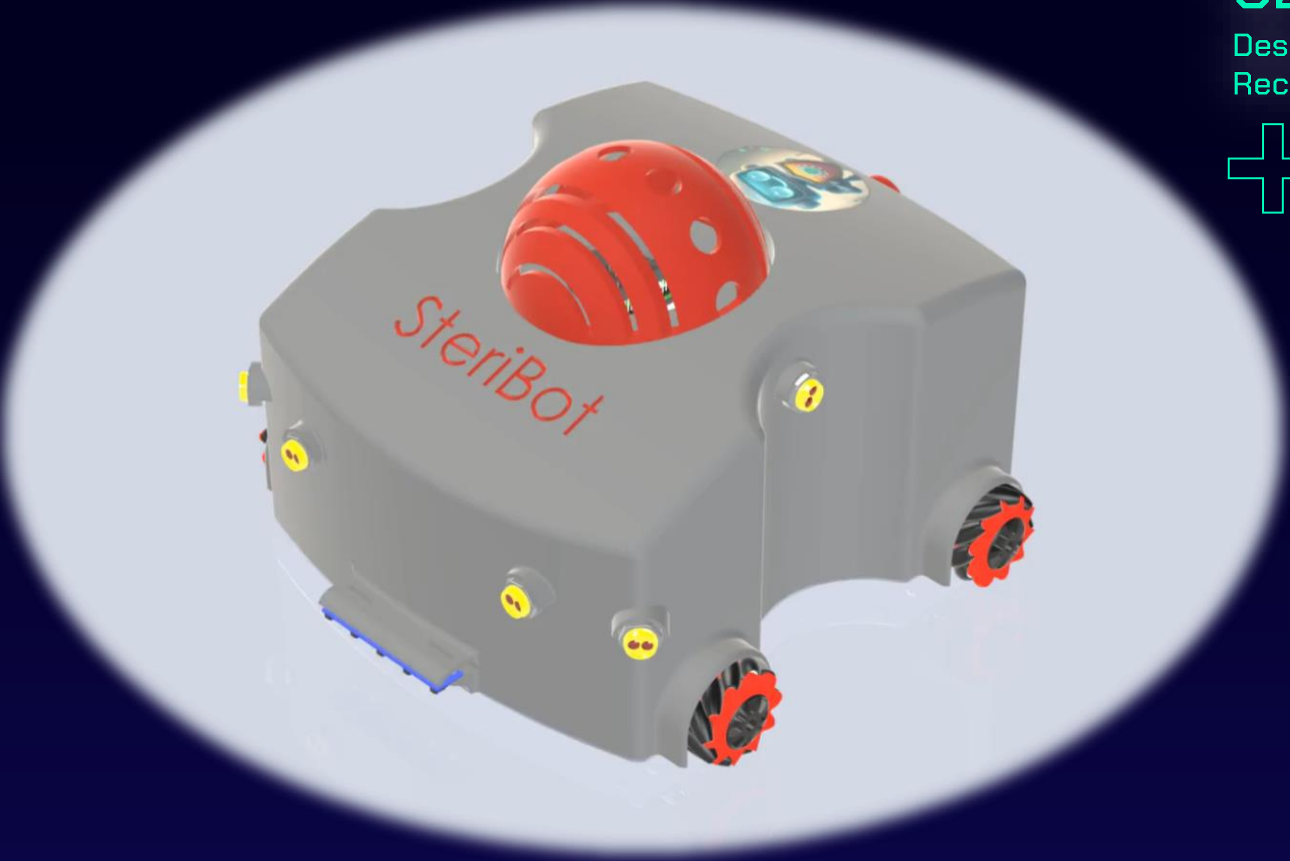
01

Project Plan

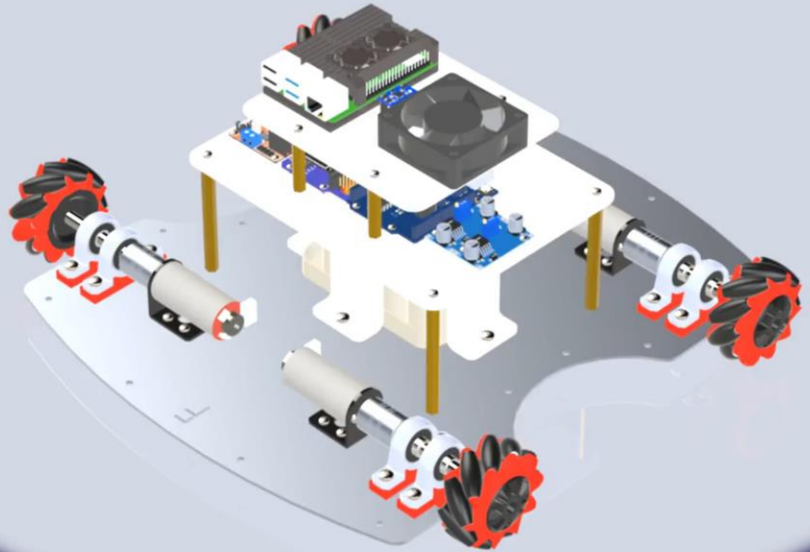


02 Design Recap

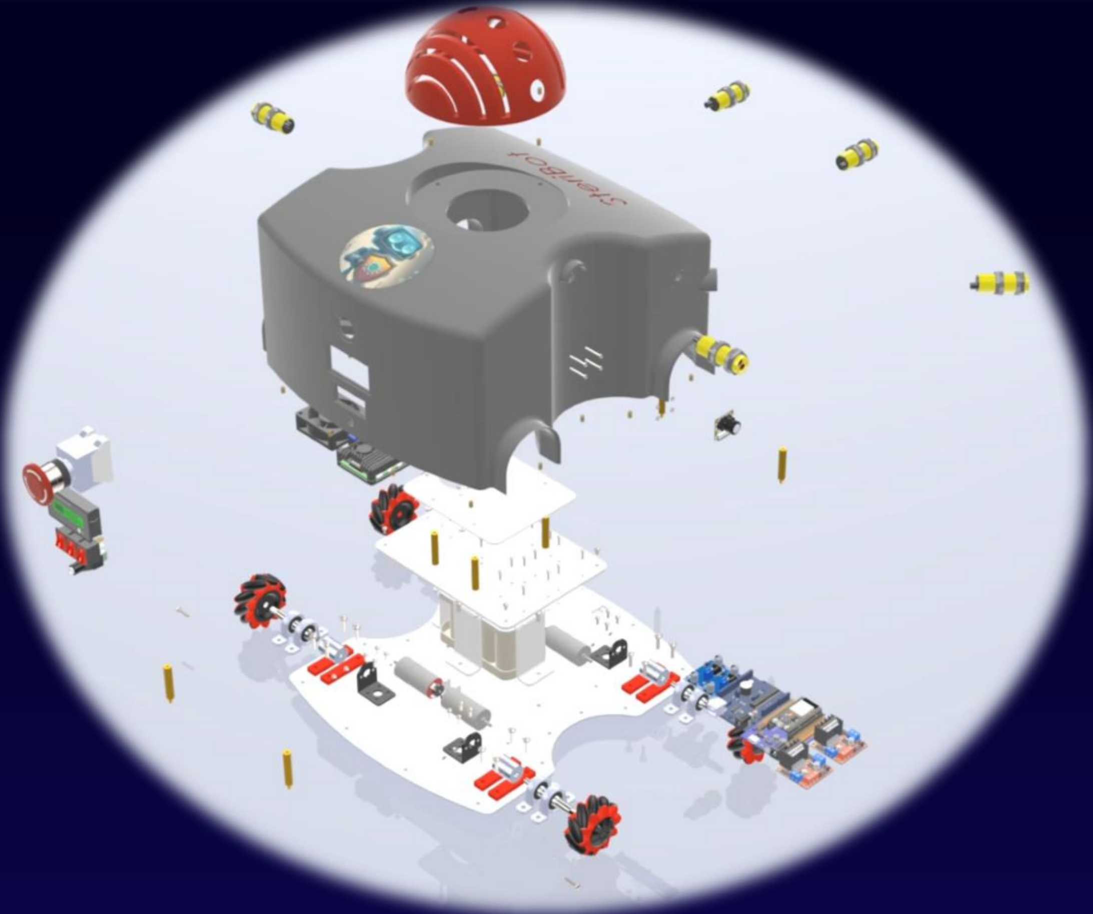
CAD Design



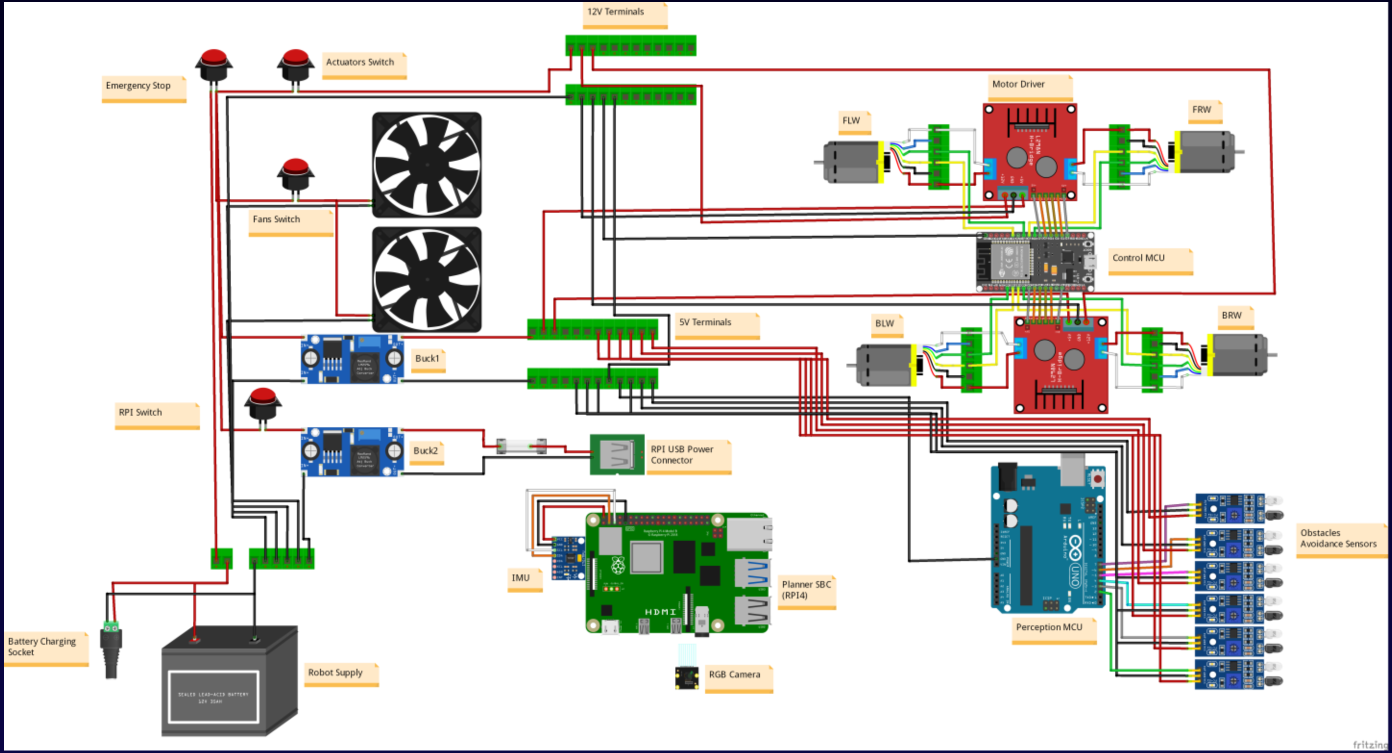
CAD Design

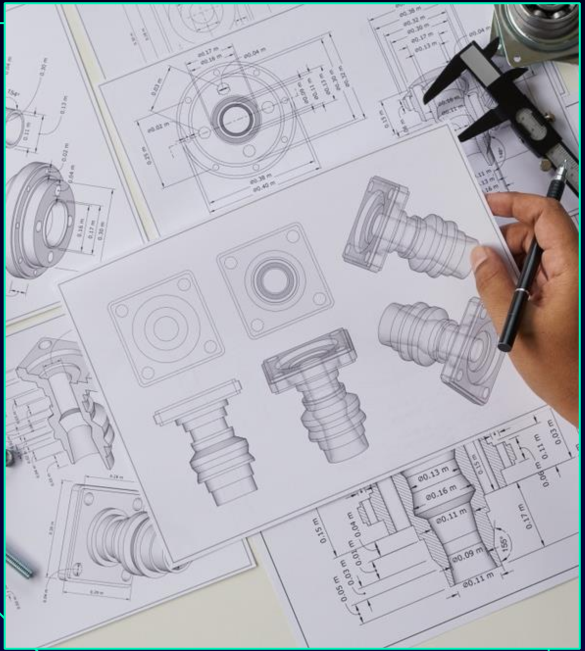


CAD Design



Electrical Design



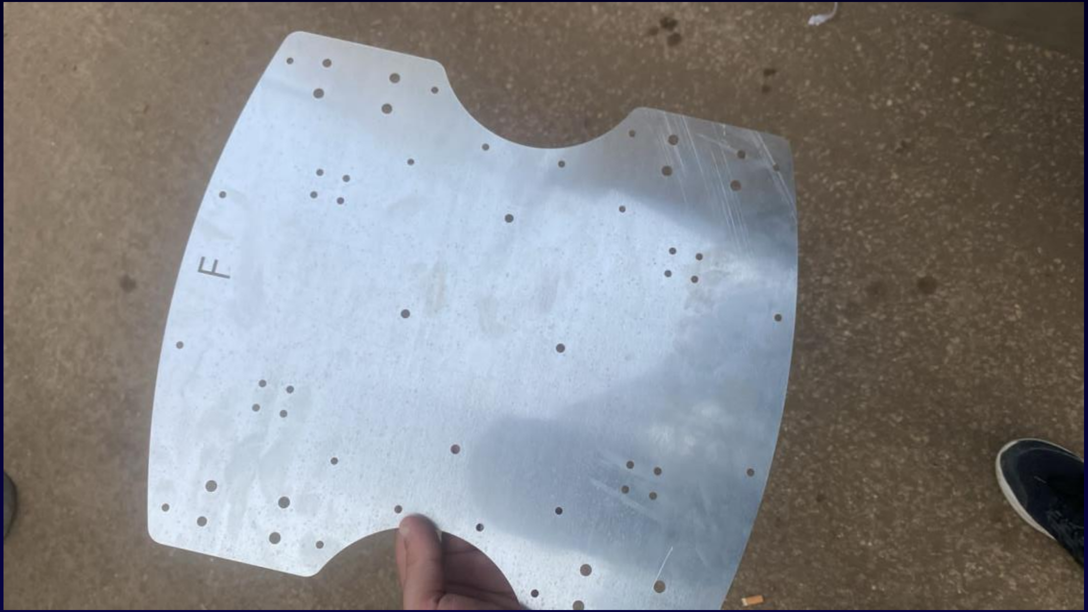


03

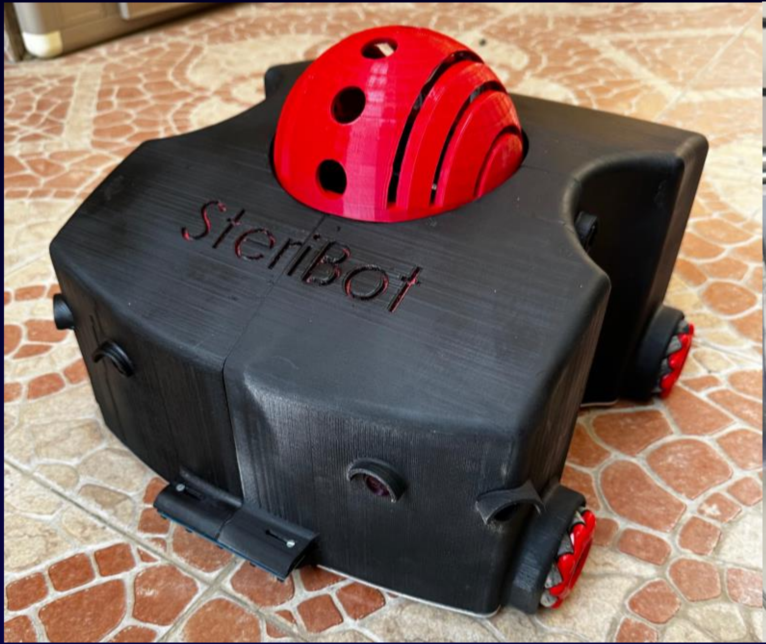
Mechanical Manufacturing

Shafts Manufacturing

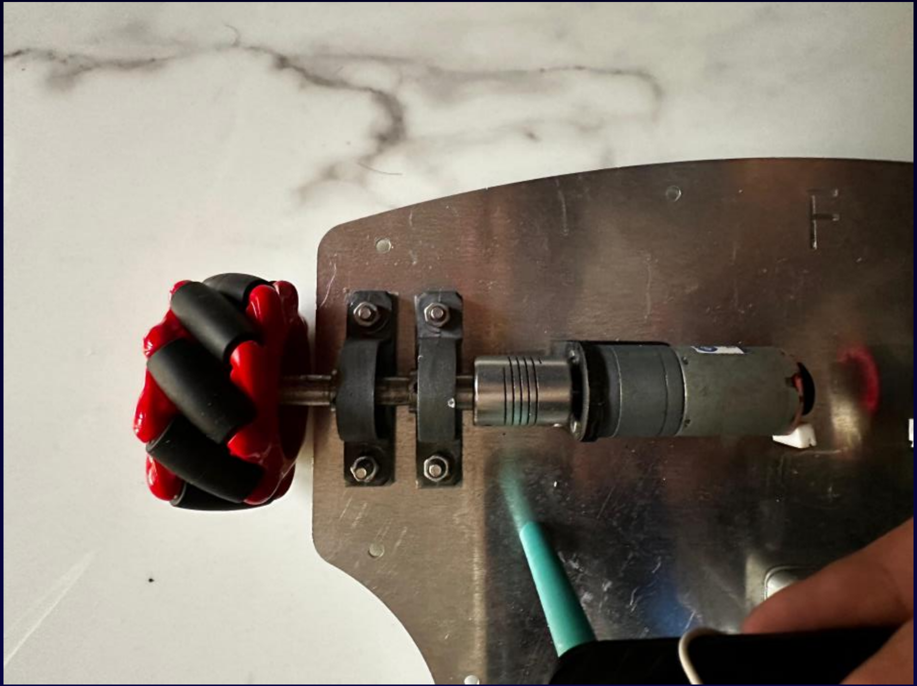
Frame Manufacturing



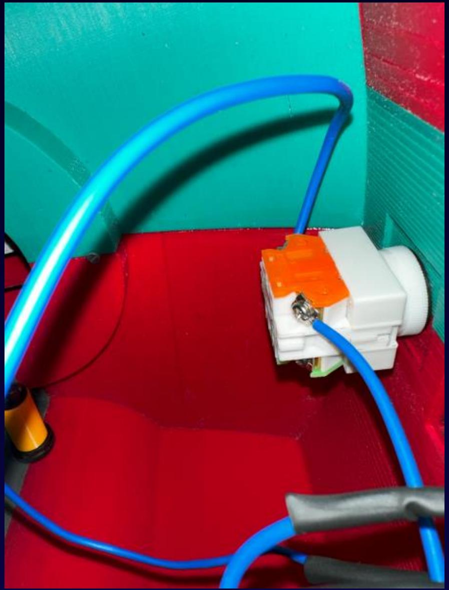
Body Manufacturing



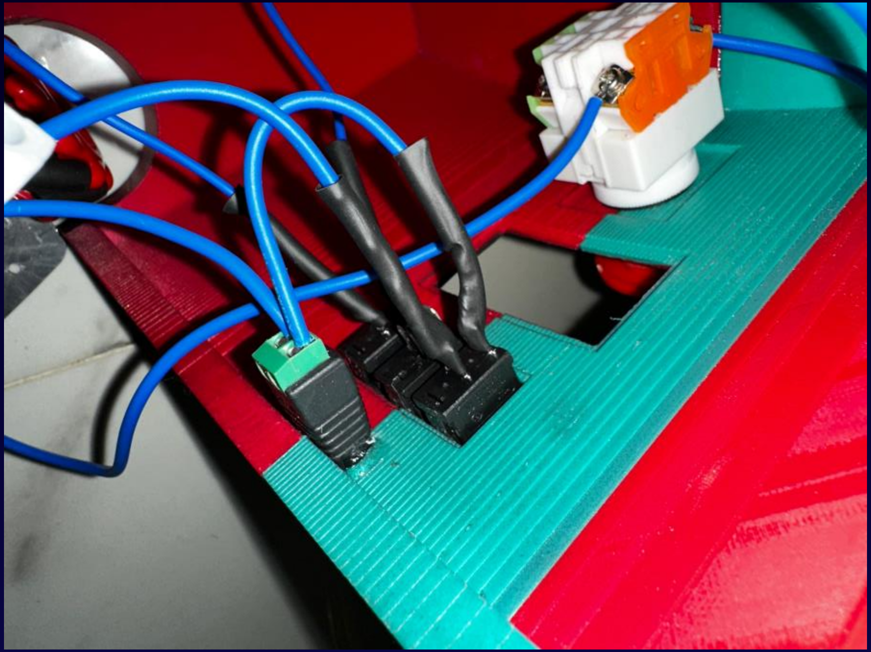
Wheel Assembly



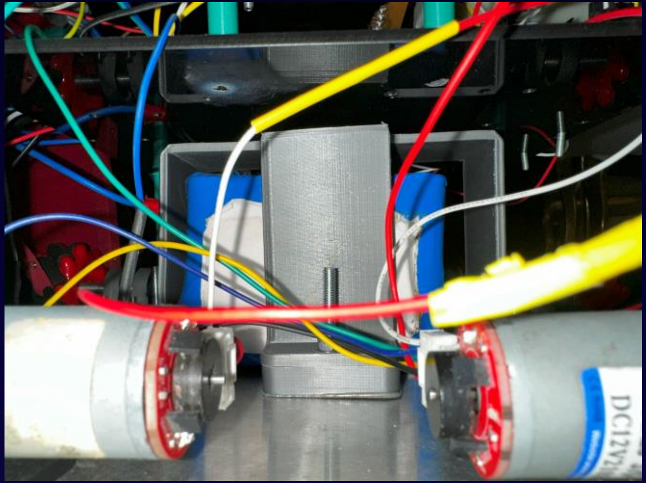
Kill Switch Fixation

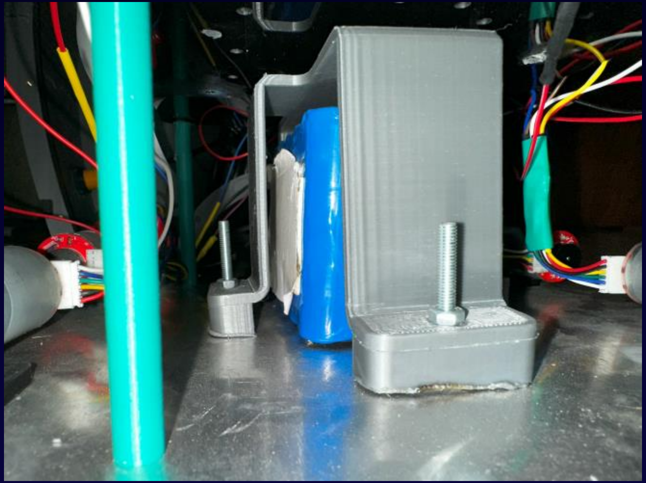


Charging Socket Fixation

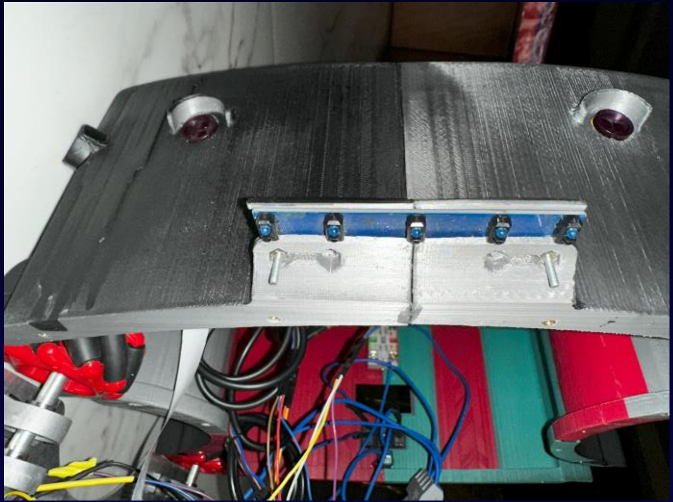


Battery Fixation

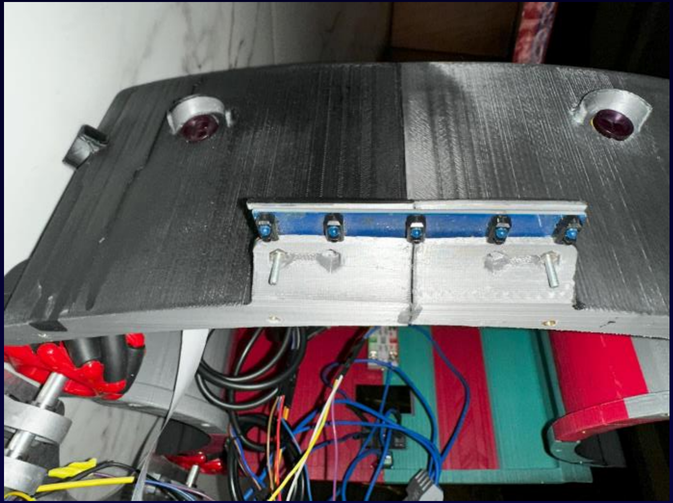


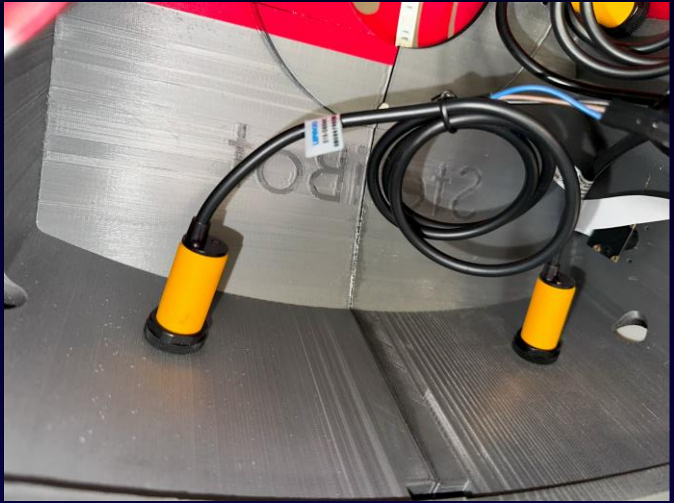


Sensors Fixation

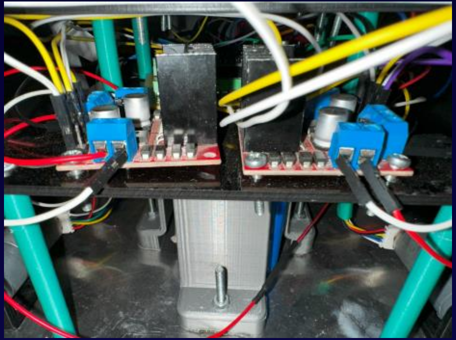
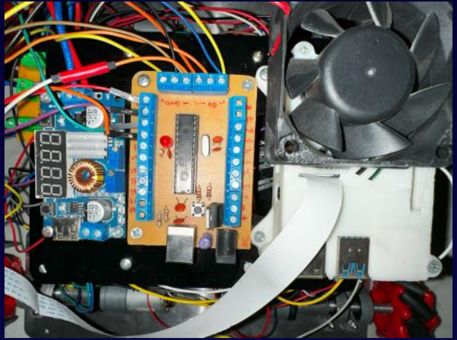


Sensors Fixation

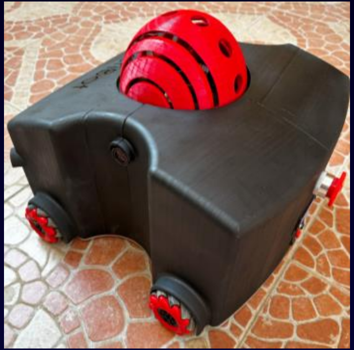


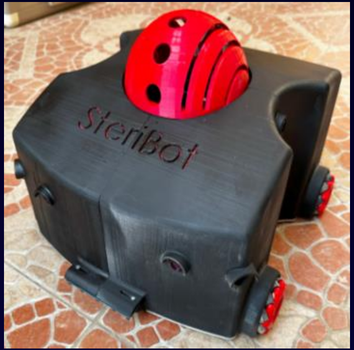


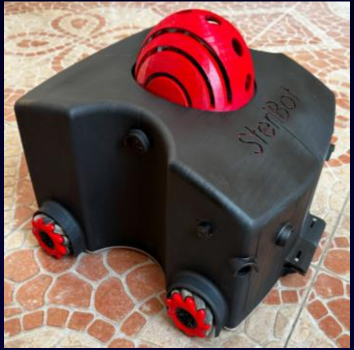
Electrical Fixation

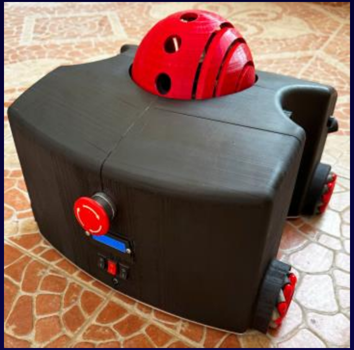
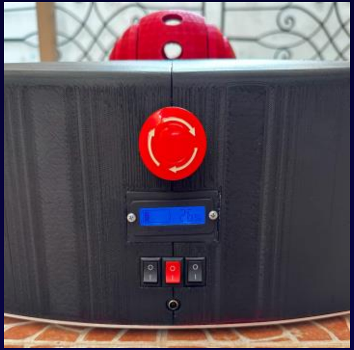
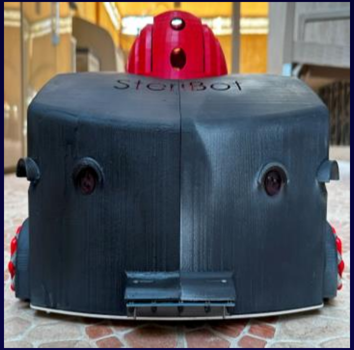



Robot Finalized Assembly







Robot Description

🤖 Meet SteriBot, Your Ultimate Hospital Room Sterilization Solution!

SteriBot is a state-of-the-art, compact robot designed to keep your hospital rooms safe and sterile. Weighing just 5kg and measuring 40cm x 40cm x 30cm, SteriBot is small but mighty, packing a host of features into its sleek design.

🔋 Long-lasting Battery Life SteriBot is equipped with a high-capacity battery that provides up to 6 hours of continuous operation. Plus, it comes with an integrated battery capacity indicator, so you always know when it’s time for a recharge. And don’t worry about power interruptions - the charging circuitry is separate from the working circuitry, ensuring smooth operation at all times.

🚨 Safety First In case of emergencies, SteriBot is integrated with a kill switch for immediate shutdown, ensuring the safety of your staff and patients.

Robot Description

🔄 Omni Wheels for Varied Movement Patterns Thanks to its omni wheels, SteriBot can navigate complex environments with ease, offering varied movement patterns to suit any room layout.

💡 UV Light Sterilization SteriBot uses UV light sterilization technology to effectively eliminate harmful bacteria and viruses, ensuring a safe and clean environment for your patients.

🚧 Obstacle Avoidance Equipped with proximity sensors, SteriBot can detect obstacles in front and on the sides, avoiding collisions and ensuring smooth operation.

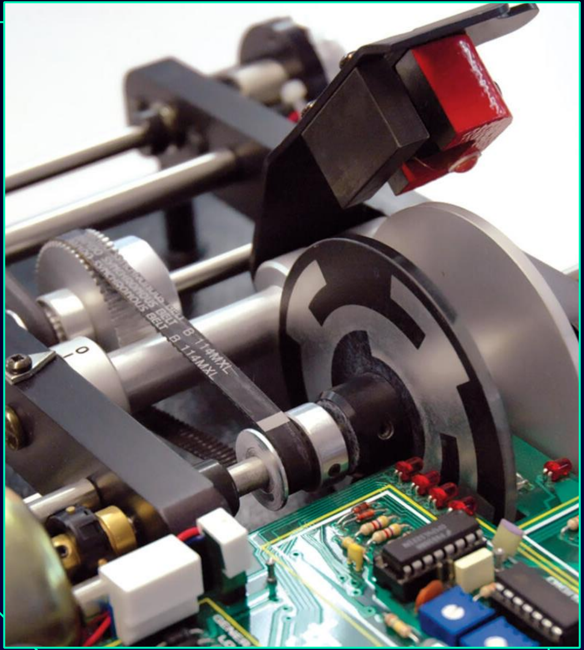
🔍 Advanced Room Detection SteriBot is integrated with a line follower module for smooth movement and low processing. It uses ArUco markers scanning for detecting the rooms in need of sterilization, with a Raspberry Pi as the master Single Board Computer (SBC).

With SteriBot, you can ensure a safe, clean, and sterile environment for your patients. Invest in SteriBot today for a healthier tomorrow! 🏥

Robot Utilization??

Robot Utilization??

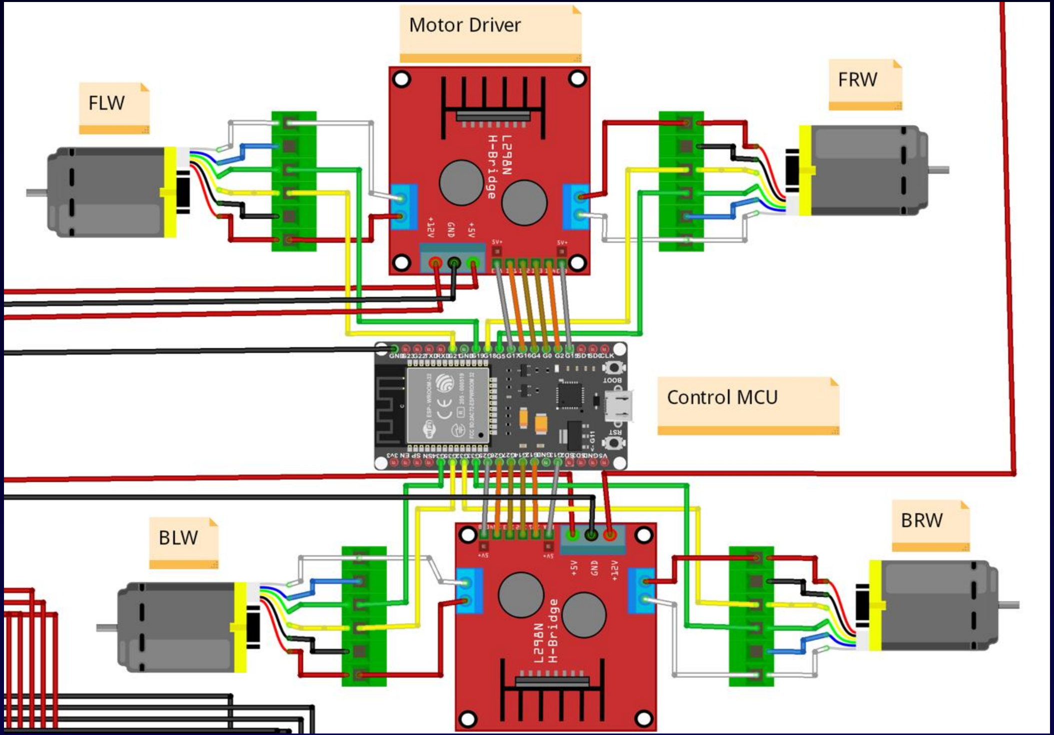
🔍 Dive Deeper with Our Team! Should you wish to explore the full potential of SteriBot, our dedicated team is just a conversation away.
Don’t hesitate to reach out for an in-depth understanding of SteriBot’s utilization.
We’re here to help you make the most of your SteriBot experience! 🤝



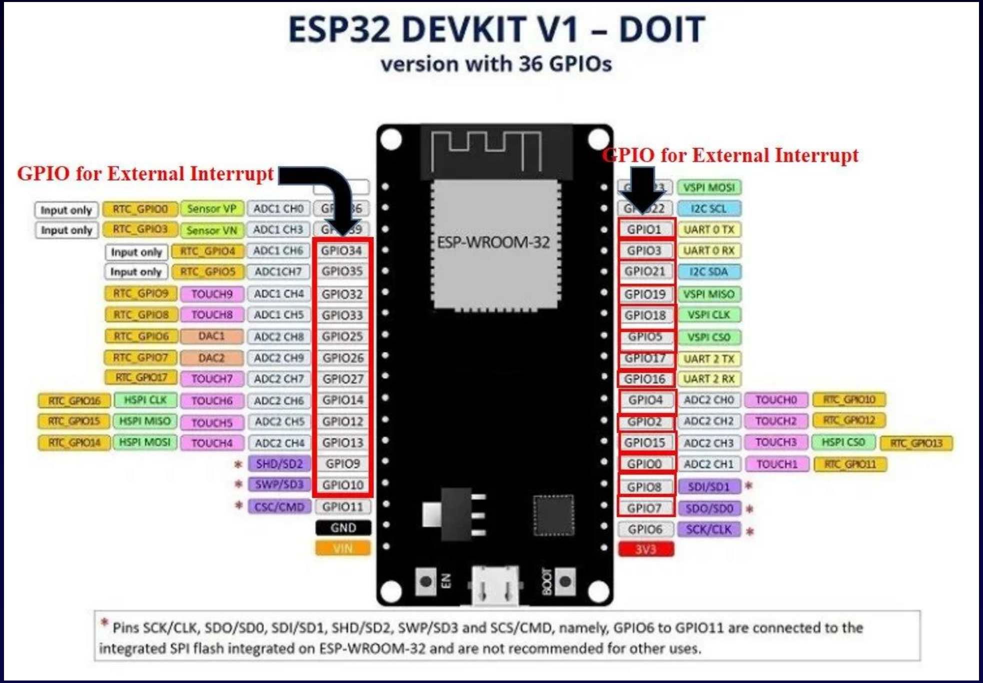
04

Motors PID

Hardware Implementation

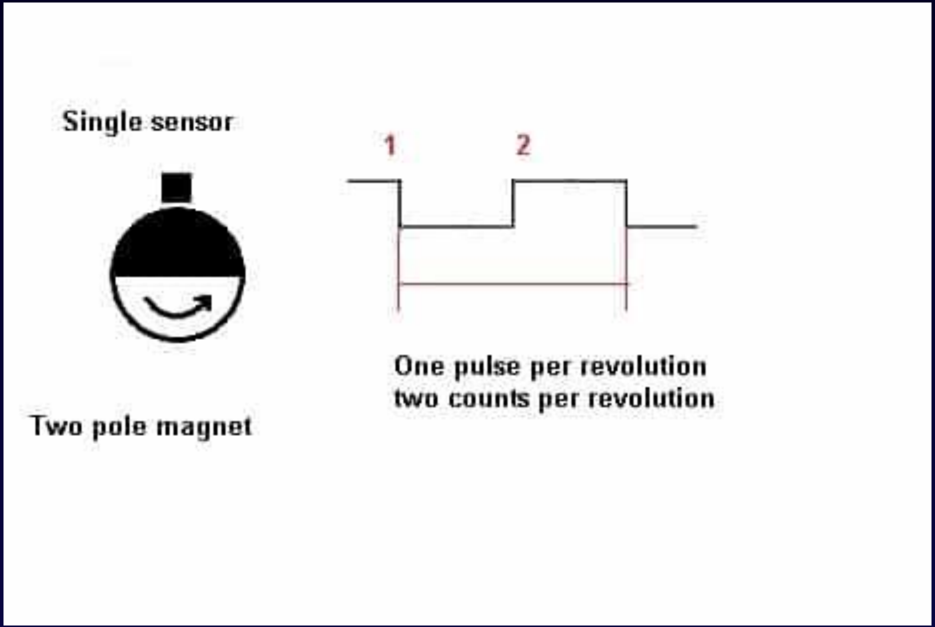


Hardware Implementation


ESP32 DEVKIT V1 – DOIT
version with 36 GPIOs
GPIO for External Interrupt
GPIO for External Interrupt
ESP-WROOM-32
Input only | RTC_GPIO0 | Sensor VP | ADC1 CH0 | GPIO36
Input only | RTC_GPIO3 | Sensor VN | ADC1 CH3 | GPIO39
Input only | RTC_GPIO4 | ADC1 CH6 | GPIO34
Input only | RTC_GPIO5 | ADC1CH7 | GPIO35
RTC_GPIO9 | TOUCH9 | ADC1 CH4 | GPIO32
RTC_GPIO8 | TOUCH8 | ADC1 CH5 | GPIO33
RTC_GPIO6 | DAC1 | ADC2 CH8 | GPIO25
RTC_GPIO7 | DAC2 | ADC2 CH9 | GPIO26
RTC_GPIO17 | TOUCH7 | ADC2 CH7 | GPIO27
RTC_GPIO16 | HSPI CLK | TOUCH6 | ADC2 CH6 | GPIO14
RTC_GPIO15 | HSPI MISO | TOUCH5 | ADC2 CH5 | GPIO12
RTC_GPIO14 | HSPI MOSI | TOUCH4 | ADC2 CH4 | GPIO13
* SHD/SD2 | GPIO9
* SWP/SD3 | GPIO10
* CSC/CMD | GPIO11
GND
VIN
GPIO23 | VSPI MOSI
GPIO22 | I2C SCL
GPIO1 | UART 0 TX
GPIO3 | UART 0 RX
GPIO21 | I2C SDA
GPIO19 | VSPI MISO
GPIO18 | VSPI CLK
GPIO5 | VSPI CS0
GPIO17 | UART 2 TX
GPIO16 | UART 2 RX
GPIO4 | ADC2 CH0 | TOUCH0 | RTC_GPIO10
GPIO2 | ADC2 CH2 | TOUCH2 | RTC_GPIO12
GPIO15 | ADC2 CH3 | TOUCH3 | HSPI CS0 | RTC_GPIO13
GPIO0 | ADC2 CH1 | TOUCH1 | RTC_GPIO11
GPIO8 | SDI/SD1 *
GPIO7 | SDO/SD0 *
GPIO6 | SCK/CLK *
3V3
EN
BOOT

* Pins SCK/CLK, SDO/SD0, SDI/SD1, SHD/SD2, SWP/SD3 and SCS/CMD, namely, GPIO6 to GPIO11 are connected to the integrated SPI flash integrated on ESP-WROOM-32 and are not recommended for other uses.

Encoder Resolution (PPR)



Encoder Resolution (PPR)

Methodology:

- Utilized ESP32Encoder library with Full quadrature mode
- Utilized external interrupt pins on the ESP32 MCU for precise pulse counting
- Rotated the motor shaft through a complete revolution (360 degrees)
- Recorded the number of pulses generated by the encoder during rotation
- Encoder PPR = Recorded PPR / 4

Results:

- Recorded PPR = 1610
- Encoder PPR = 402

RPM Calculation

Methodology:

- We only obtain pulses from the dc motor encoder

Equation:

- Δpulses = (current pulses – old pulses)
- ΔRevolutions = Δpulses / PPR
- ΔRevolutions / Δtime → gives what fraction of revolution achieved per the sample time
- To obtain the RPM → (ΔRevolutions / Δtime) / (1000 * 60)
- We divided by 1000 to convert ms to s
- And the division by 60 to convert s to min

Software Implementation

04

Motors PID

Methodology:

- CleanRTOS and PID Integration
- Motor Control Configuration
- PID Control Task for Each Wheel
- Encoder Feedback Processing
- Velocity Synchronization
- Filtering and Output Application
- Task Logging for Monitoring
- Real-Time Control and Monitoring

Software Implementation

Methodology:

- ✓ CleanRTOS and PID Integration
- ➢ Motor Control Configuration
- ➢ PID Control Task for Each Wheel
- ➢ Encoder Feedback Processing
- ➢ Velocity Synchronization
- ➢ Filtering and Output Application
- ➢ Task Logging for Monitoring
- ➢ Real-Time Control and Monitoring

Software Implementation

Methodology:

- ✓ CleanRTOS and PID Integration
- ✓ Motor Control Configuration
- ➢ PID Control Task for Each Wheel
- ➢ Encoder Feedback Processing
- ➢ Velocity Synchronization
- ➢ Filtering and Output Application
- ➢ Task Logging for Monitoring
- ➢ Real-Time Control and Monitoring

Software Implementation

Methodology:

- ✓ CleanRTOS and PID Integration
- ✓ Motor Control Configuration
- ✓ PID Control Task for Each Wheel
- ➢ Encoder Feedback Processing
- ➢ Velocity Synchronization
- ➢ Filtering and Output Application
- ➢ Task Logging for Monitoring
- ➢ Real-Time Control and Monitoring

Software Implementation

Methodology:

- ✓ CleanRTOS and PID Integration
- ✓ Motor Control Configuration
- ✓ PID Control Task for Each Wheel
- ✓ Encoder Feedback Processing
- ➢ Velocity Synchronization
- ➢ Filtering and Output Application
- ➢ Task Logging for Monitoring
- ➢ Real-Time Control and Monitoring

Software Implementation

Methodology:

- ✓ CleanRTOS and PID Integration
- ✓ Motor Control Configuration
- ✓ PID Control Task for Each Wheel
- ✓ Encoder Feedback Processing
- ✓ Velocity Synchronization
- ➢ Filtering and Output Application
- ➢ Task Logging for Monitoring
- ➢ Real-Time Control and Monitoring

Software Implementation

04

Motors PID

Methodology:

- ✓ CleanRTOS and PID Integration
- ✓ Motor Control Configuration
- ✓ PID Control Task for Each Wheel
- ✓ Encoder Feedback Processing
- ✓ Velocity Synchronization
- ✓ Filtering and Output Application
- ➢ Task Logging for Monitoring
- ➢ Real-Time Control and Monitoring

Software Implementation

04

Motors PID

Methodology:

- ✓ CleanRTOS and PID Integration
- ✓ Motor Control Configuration
- ✓ PID Control Task for Each Wheel
- ✓ Encoder Feedback Processing
- ✓ Velocity Synchronization
- ✓ Filtering and Output Application
- ✓ Task Logging for Monitoring
- ➢ Real-Time Control and Monitoring

Software Implementation

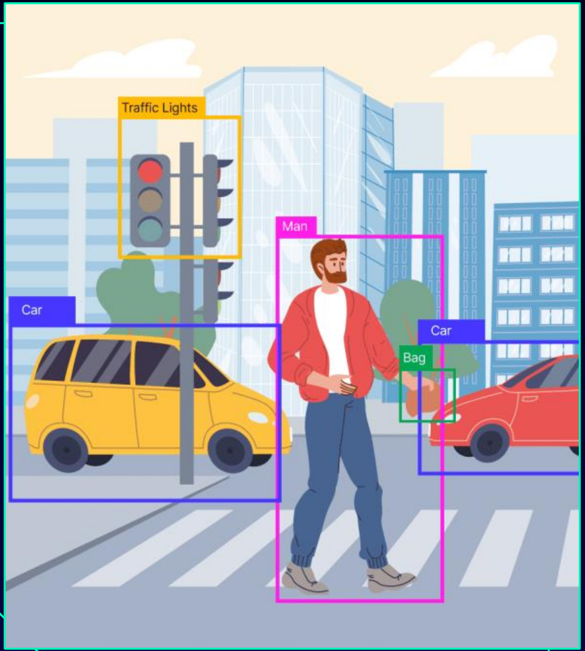
04

Motors PID

Methodology:

- ✓ CleanRTOS and PID Integration
- ✓ Motor Control Configuration
- ✓ PID Control Task for Each Wheel
- ✓ Encoder Feedback Processing
- ✓ Velocity Synchronization
- ✓ Filtering and Output Application
- ✓ Task Logging for Monitoring
- ✓ Real-Time Control and Monitoring

Robot Control Result



05

Vision Module

Target of Vision Module?

The vision module of SteriBot is one of its most advanced features. It uses ArUco marker detection, a popular method for pose estimation of square planar markers used in computer vision.

ArUco markers are easy to detect, cheap to make and provide a robust method for estimating the pose of the SteriBot in a 3D space. This makes them ideal for our application.

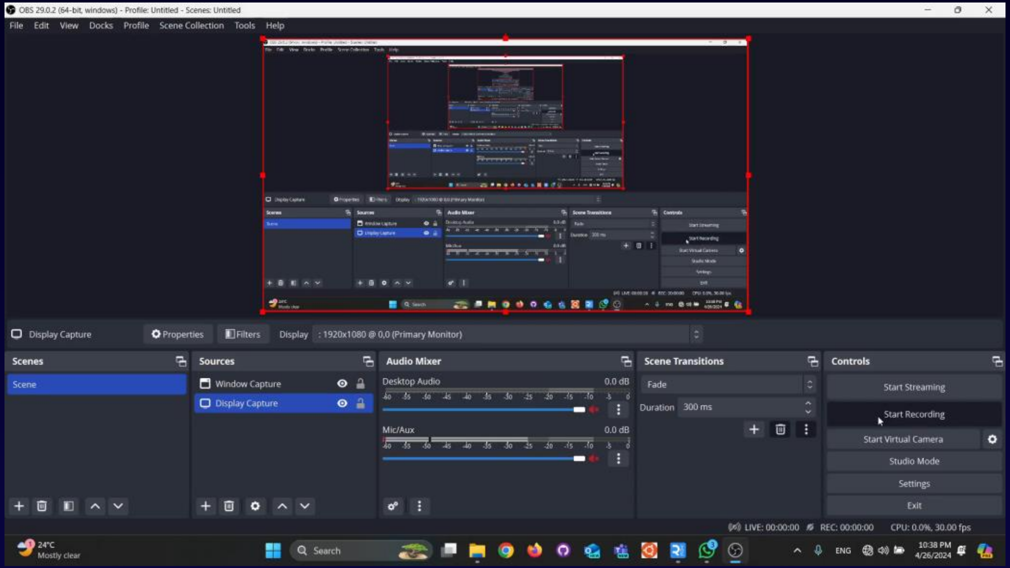
In the context of SteriBot, these markers are used to identify rooms that need sterilization. Each room is assigned a unique ArUco marker. When SteriBot detects a marker, it recognizes the corresponding room and checks whether it needs sterilization.

This system allows SteriBot to autonomously navigate through the hospital, identify rooms in need of sterilization, and efficiently carry out its task. The use of a Raspberry Pi as the master Single Board Computer (SBC) ensures smooth processing and real-time marker detection.

Methodology of Vision Module?

- Initialization: SteriBot starts by initializing a ROS (Robot Operating System) node and setting up a publisher that will broadcast a flag indicating whether a room has been sterilized or not.
- Video Capture: SteriBot then activates its onboard camera to capture real-time video feed.
- Marker Detection: Each frame of the video feed is converted to grayscale and scanned for ArUco markers - unique identifiers assigned to each room.
- Room Identification & Sterilization Status: If an ArUco marker is detected, SteriBot checks if the corresponding room ID is in the list of sterilized rooms. If it is, SteriBot broadcasts a ‘1’ (indicating the room is sterilized). If not, it broadcasts a ‘0’ (indicating the room is not sterilized).
- Display & User Interaction: The video feed, complete with detected ArUco markers, is displayed for user reference. The user can stop the program at any time by pressing ‘q’.
- Clean Up: Once the operation is complete or interrupted, SteriBot safely releases the video capture and closes all windows.

Detection Results



Display : 1920x1080 @ 0,0 (Primary Monitor)

Desktop Audio 0.0 dB

Mic/Aux 0.0 dB

Fade

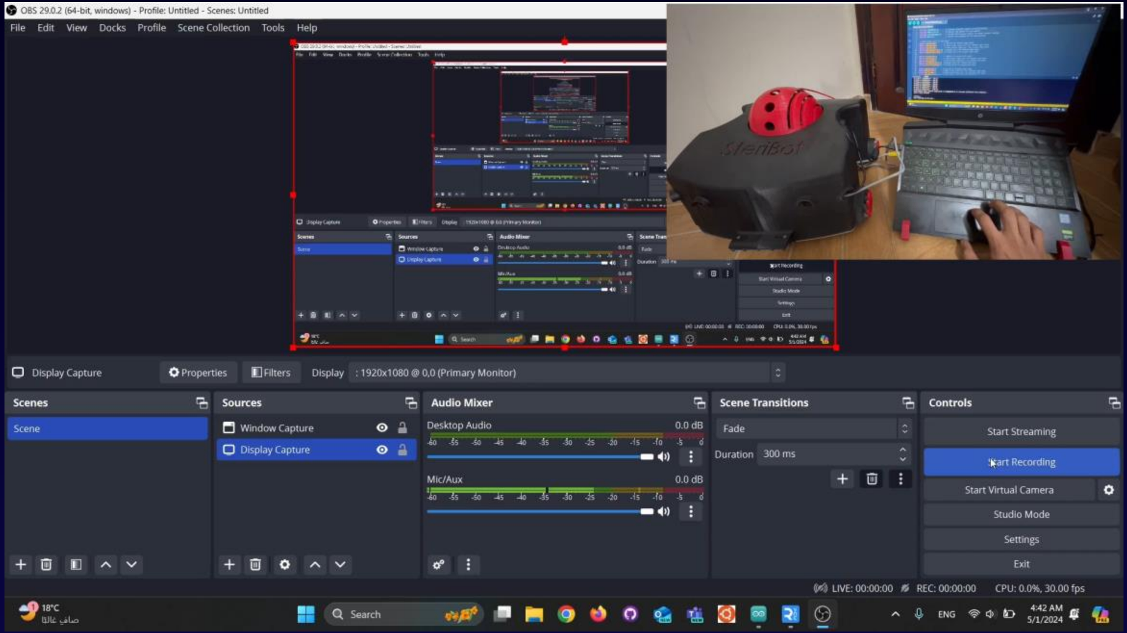
Duration 300 ms

LIVE: 00:00:00 REC: 00:00:00 CPU: 0.0%, 30.00 fps

24°C Mostly clear

10:38 PM 4/26/2024

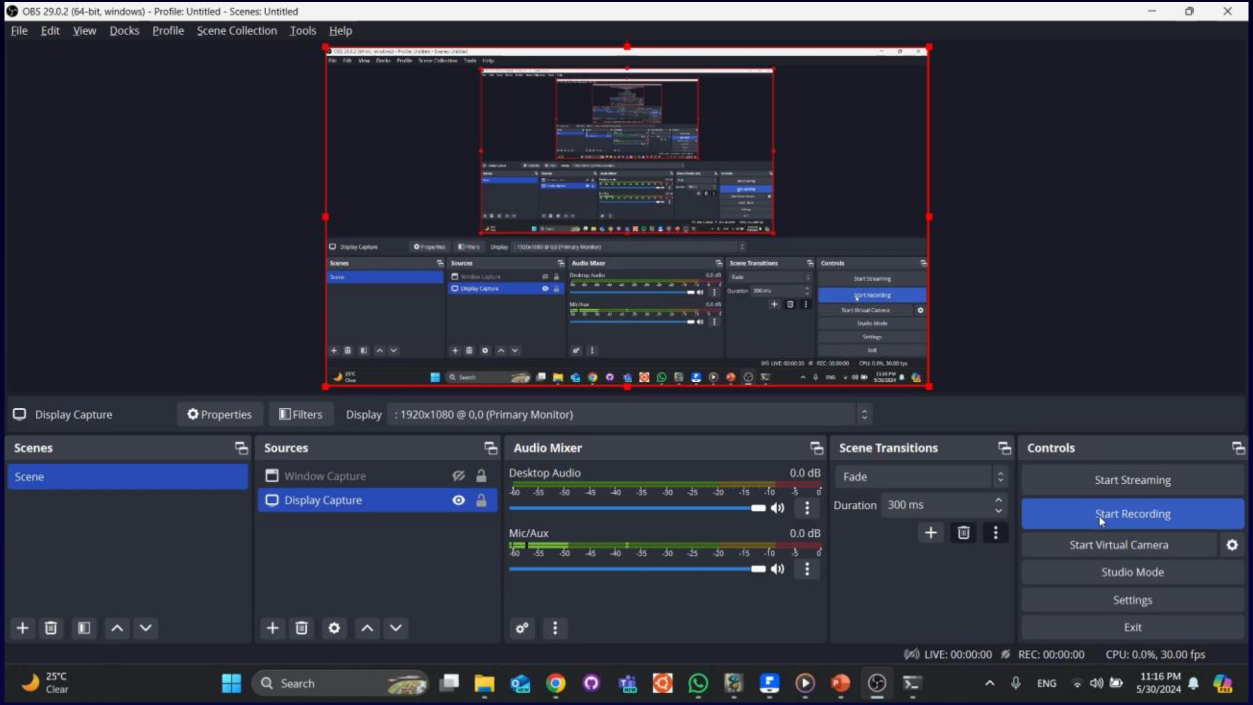
Test Results

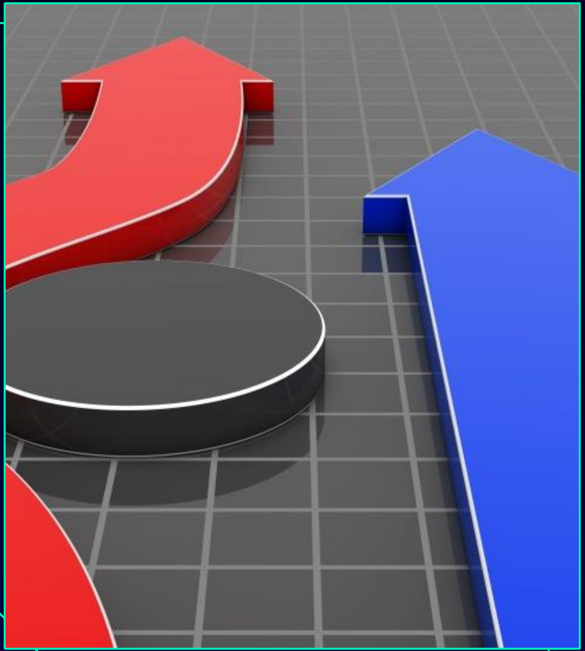


06

GUI Program

Demonstration Video

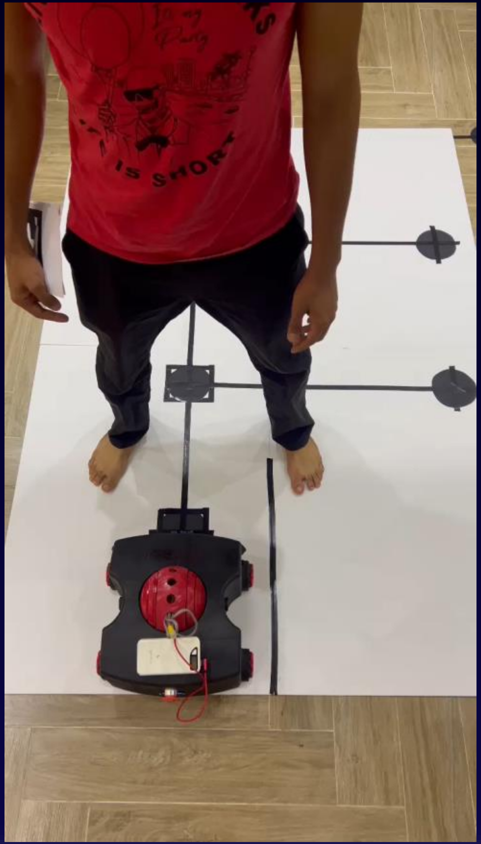




07

Obstacle Avoidance

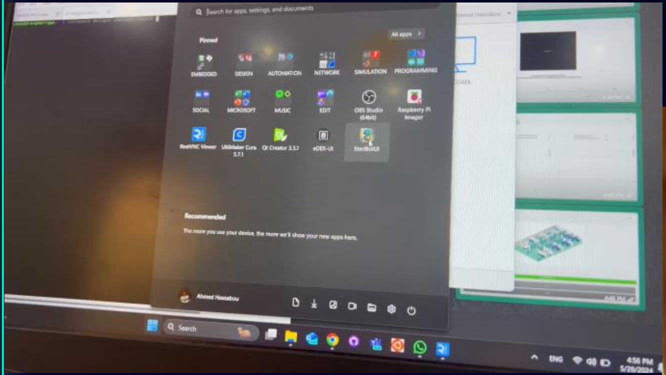
Test Results

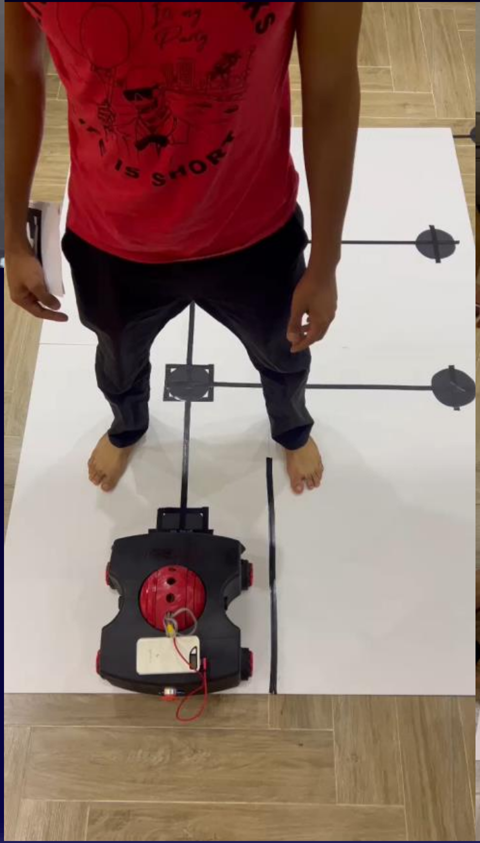


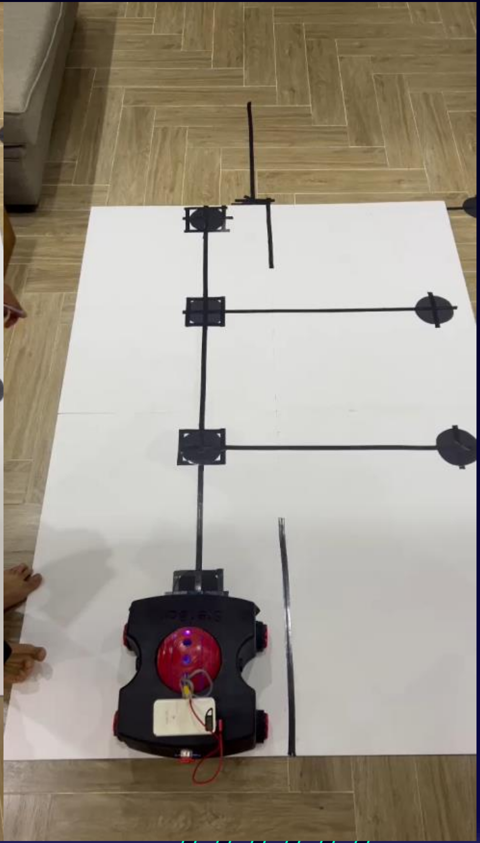
08

Full Functionality

Test Results







Extra Details ??

That's It For Today