



# STERIBOT

YOUR GUARDIAN OF PURITY IN THE  
BATTLE OF GERMS



# TEAM MEMBERS

FARIDA AMR

20P6363

HANYA HOSSAM

20P3687

SALMA AMR

20P4526

AHMED HASSABOU

19P4007

MOSTAFA SALEH

20P4538

MOHAMED MOHANNA

19P6218



# TABLE OF CONTENTS

01

Project Plan

02

Design  
Recap

03

Mechanical  
Manufacturing

04

Motors PID

05

Vision  
Module



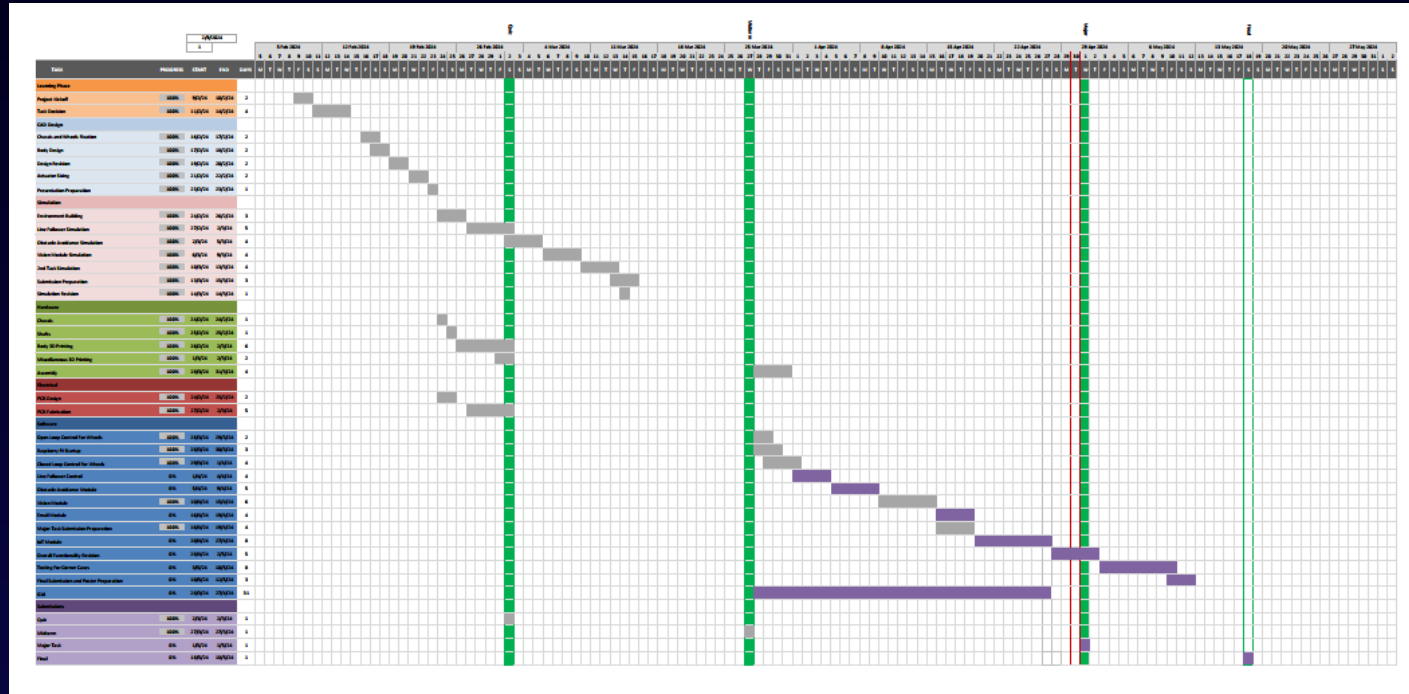


# 01

## Project Plan



## 01 Project Plan





# 02

## Design Recap





CAD  
Design

02

Design  
Recap

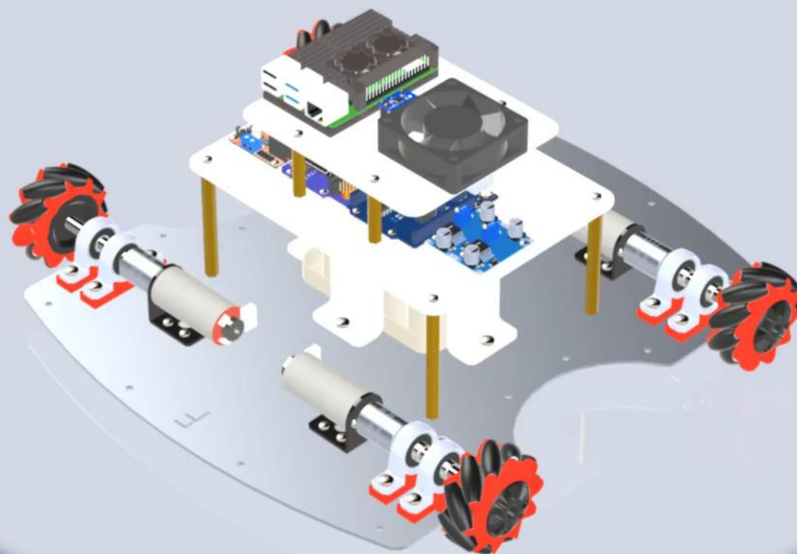




CAD  
Design

02

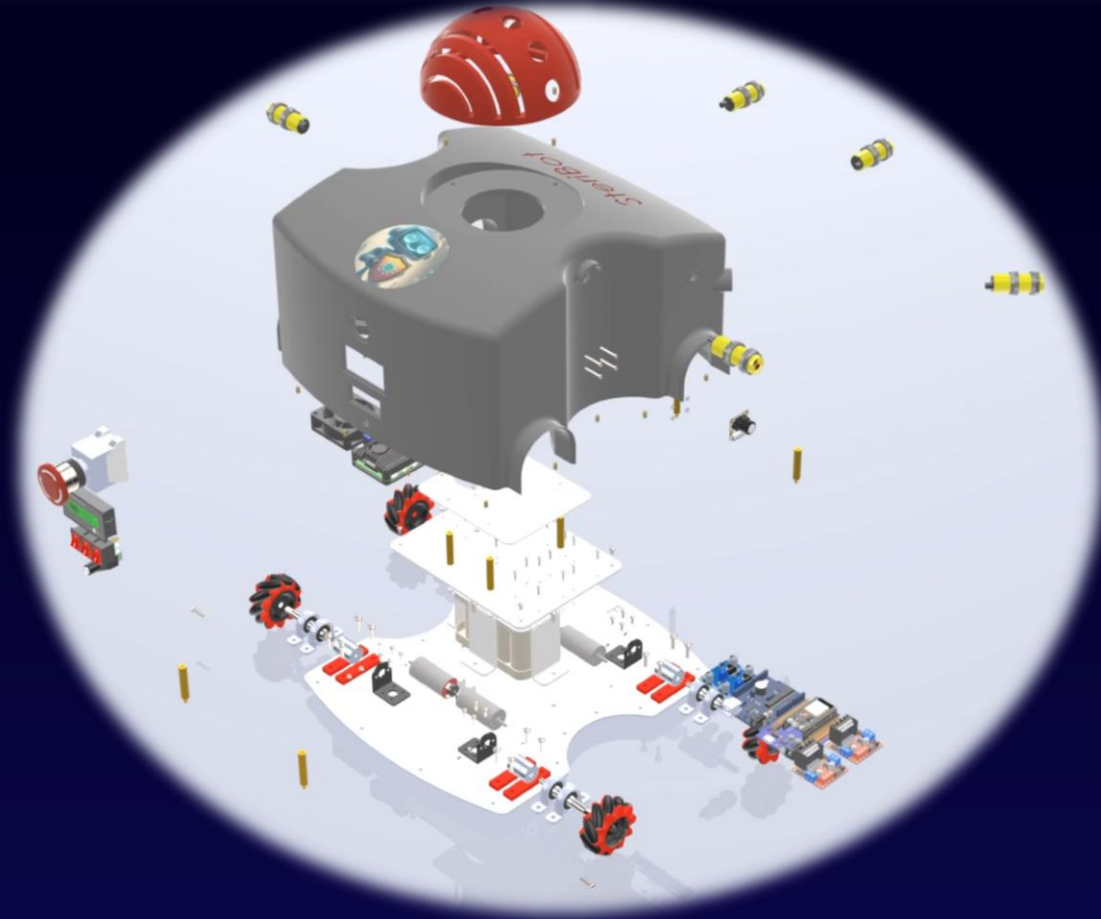
Design  
Recap







CAD  
Design



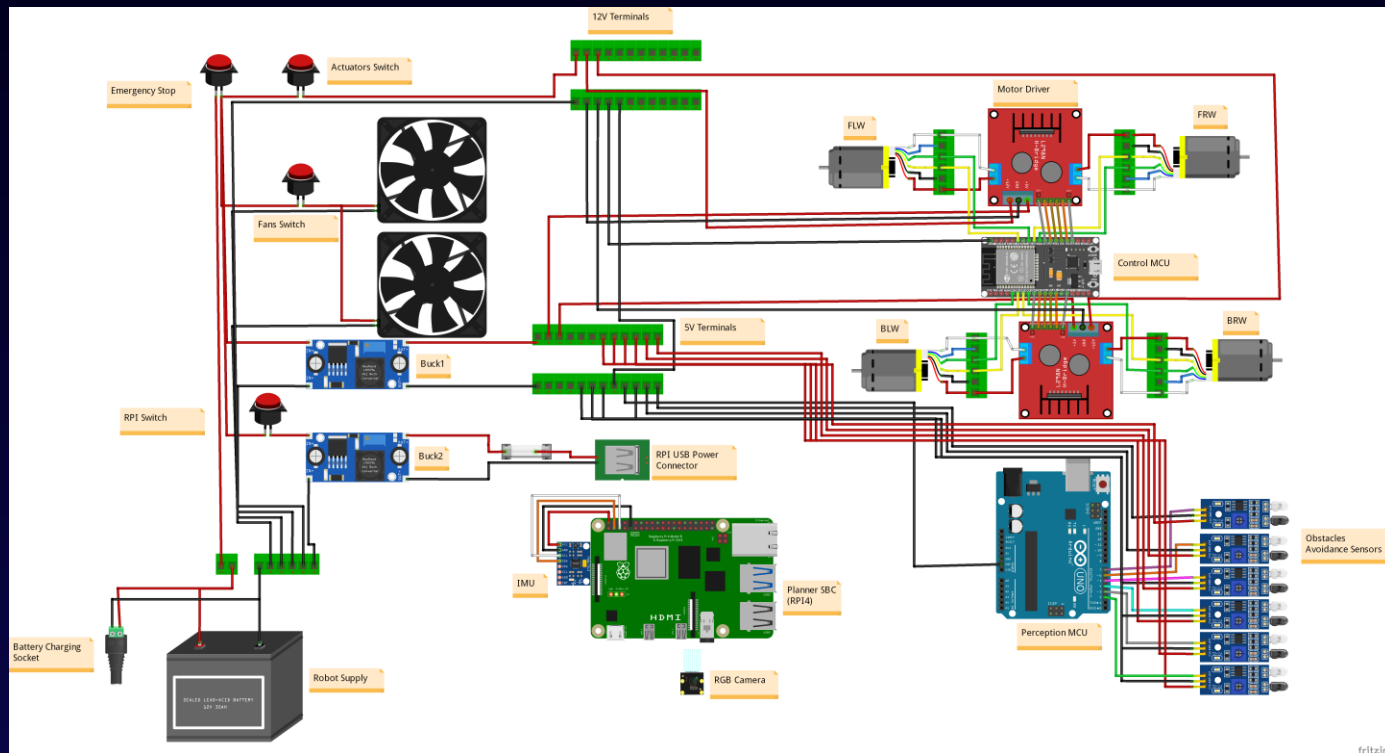
02

Design  
Recap



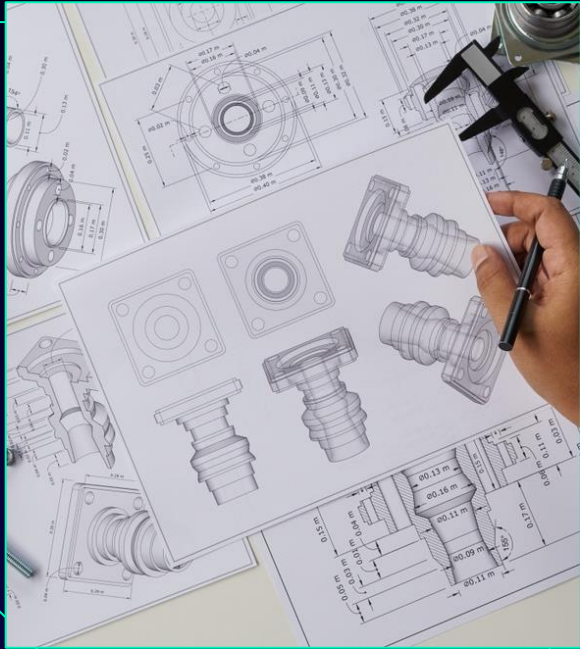


## Electrical Design



fritzing





03

# Mechanical Manufacturing



## Shafts Manufacturing

03

Mechanical  
Manufacturing





# Frame Manufacturing

03

Mechanical  
Manufacturing

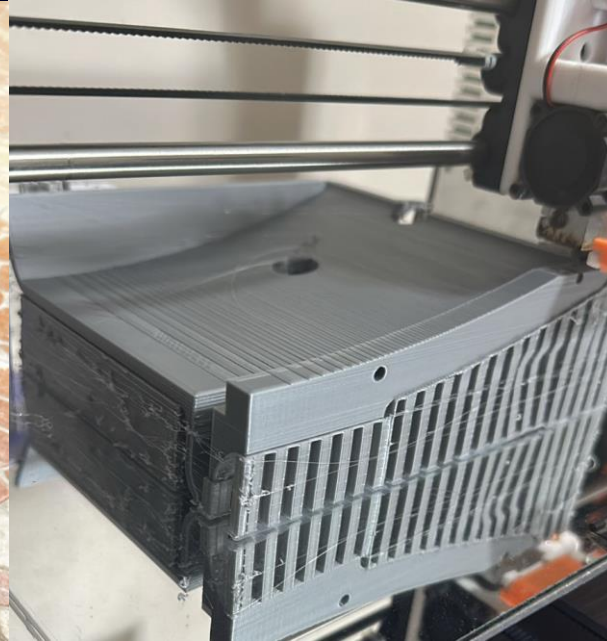




## Body Manufacturing

03

Mechanical  
Manufacturing







## Wheel Assembly



03

Mechanical  
Manufacturing

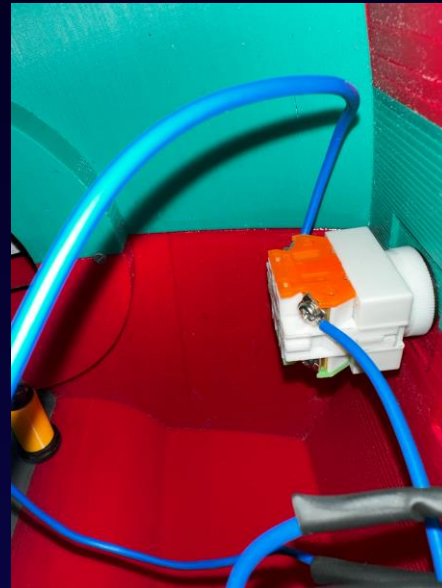




## Kill Switch Fixation

03

Mechanical  
Manufacturing



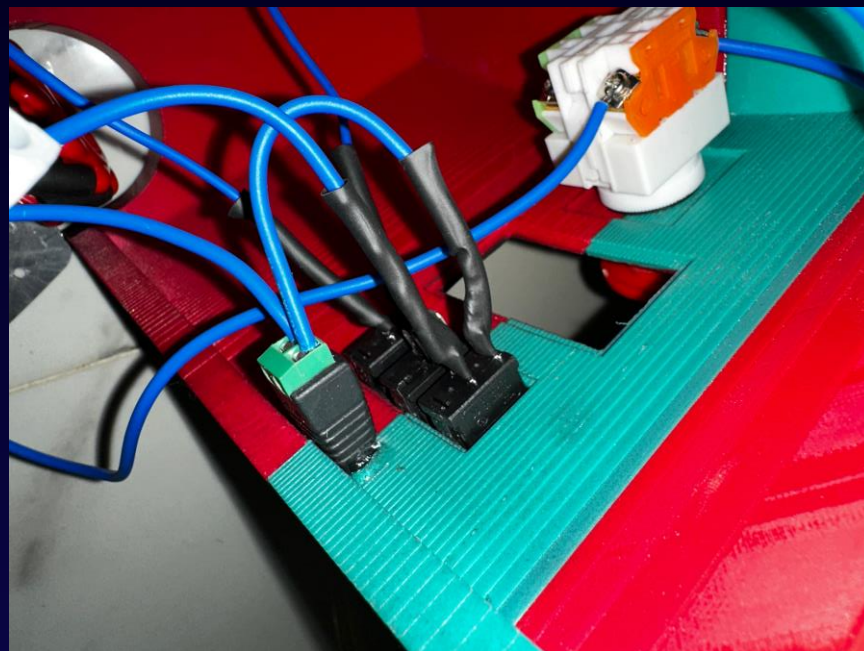




## Charging Socket Fixation

03

Mechanical  
Manufacturing



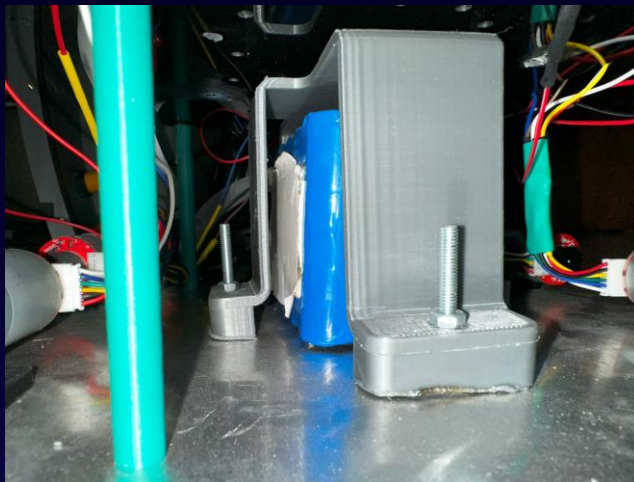
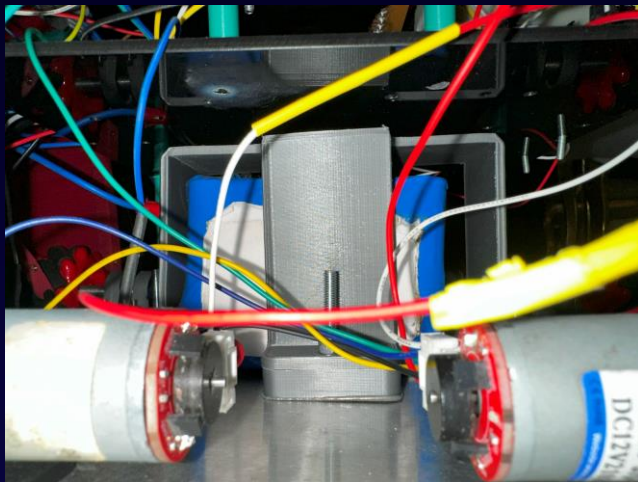


03

Mechanical  
Manufacturing



## Battery Fixation



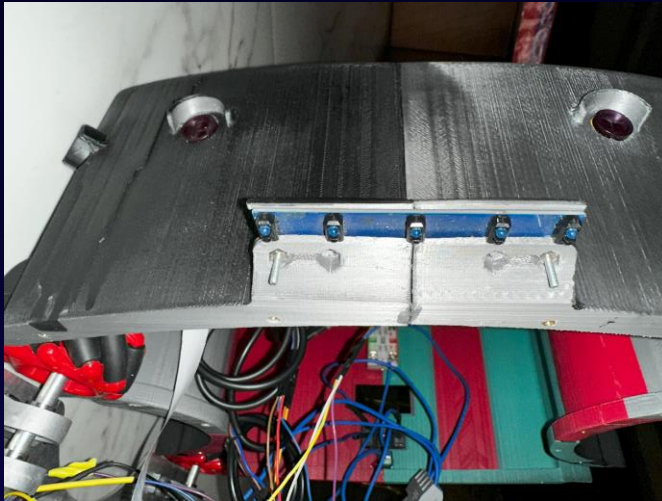


03

Mechanical  
Manufacturing



## Sensors Fixation



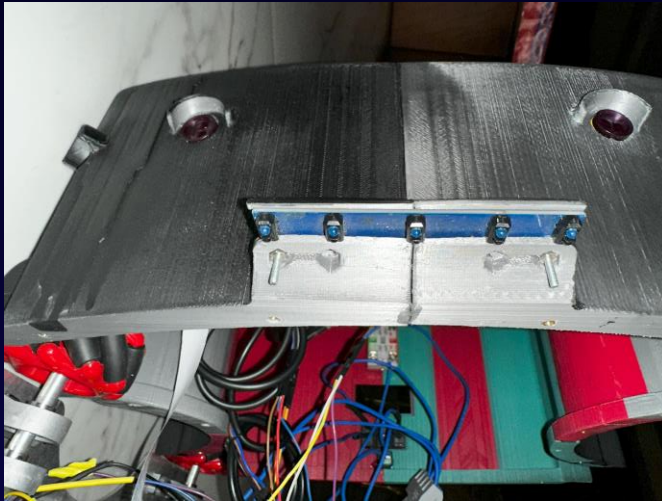


03

Mechanical  
Manufacturing



## Sensors Fixation



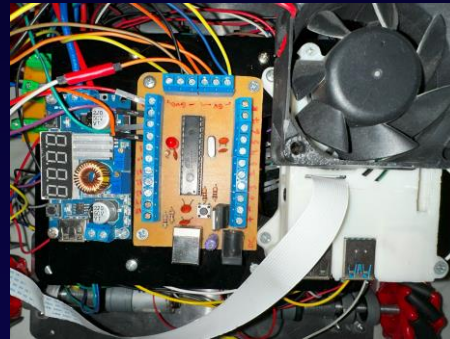
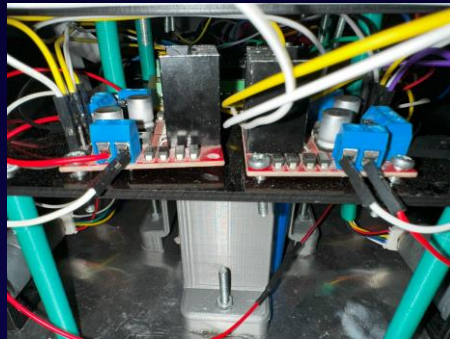




## Electrical Fixation

03

Mechanical  
Manufacturing





## Robot Finalized Assembly

03

Mechanical  
Manufacturing






# Robot Description


03


Mechanical  
Manufacturing



 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


03


Mechanical  
Manufacturing




 **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! 









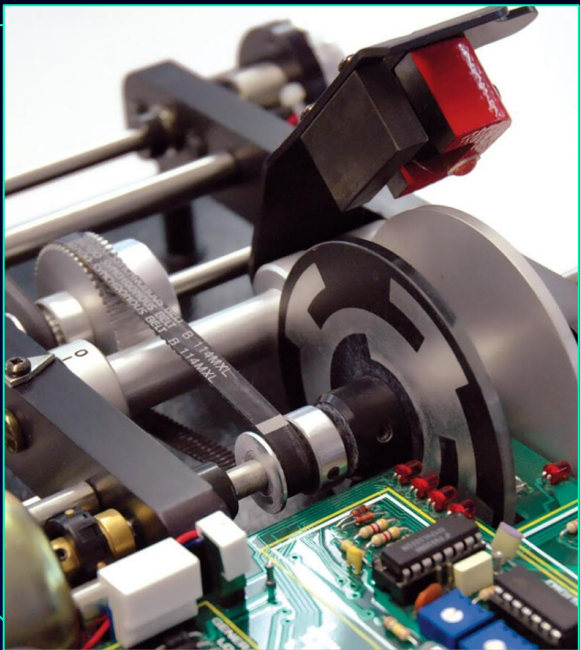
Robot  
Utilization??



## 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

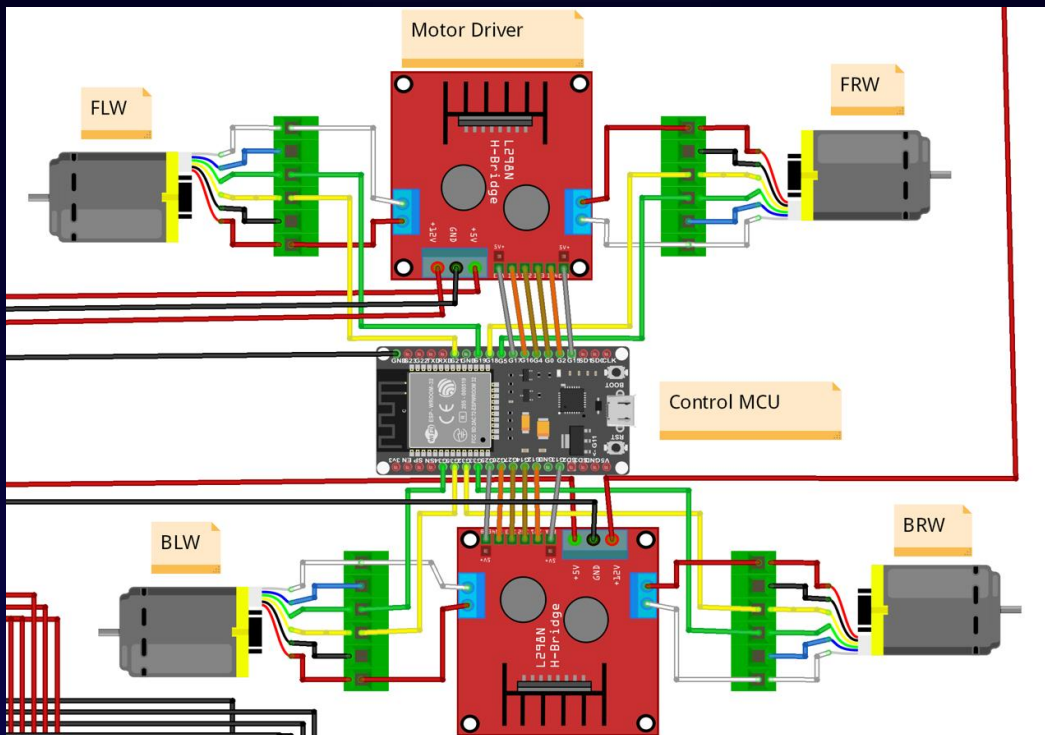
Motors PID



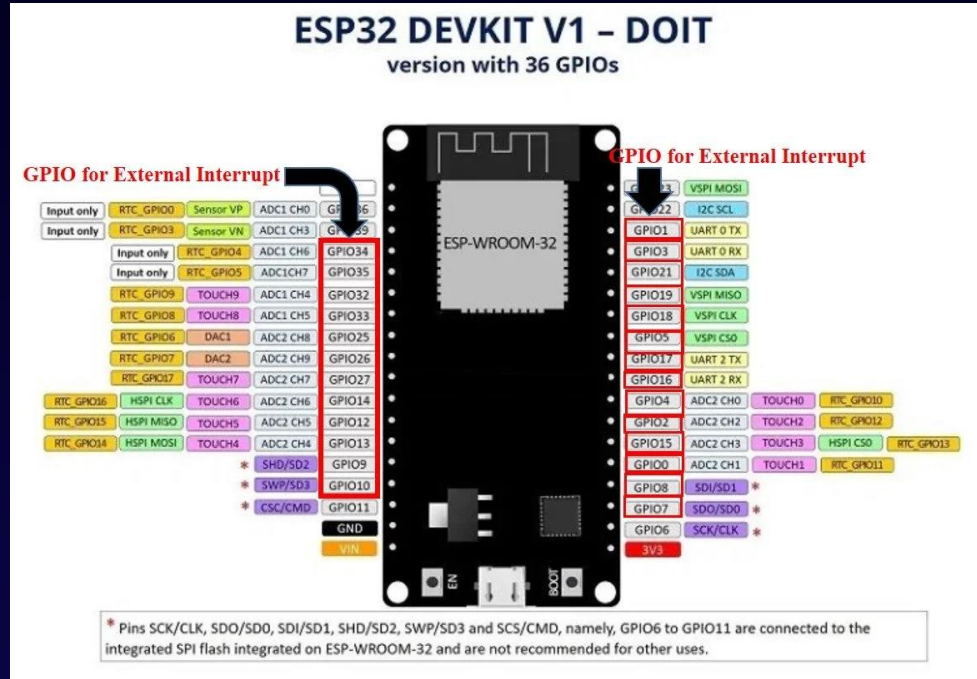
# Hardware Implementation

04

Motors PID



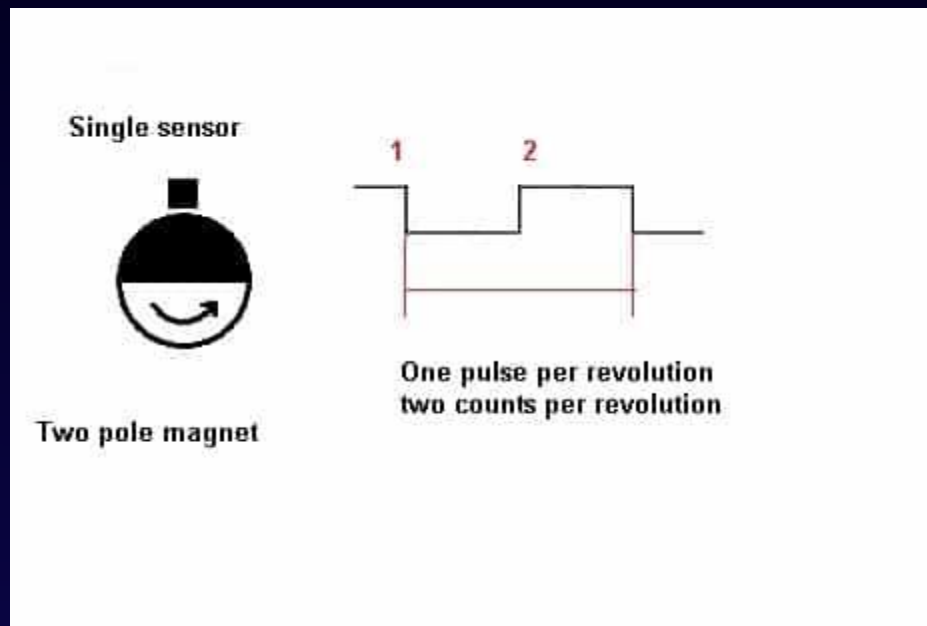
## 04 Motors PID




# Encoder Resolution (PPR)

04

Motors PID





## Encoder Resolution (PPR)

04

Motors PID



### 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

04

Motors PID



Methodology:


- We only obtain pulses from the dc motor encoder

Equation:

- $\Delta \text{pulses} = (\text{current pulses} - \text{old pulses})$
- $\Delta \text{Revolutions} = \Delta \text{pulses} / \text{PPR}$
- $\Delta \text{Revolutions} / \Delta \text{time} \rightarrow$  gives what fraction of revolution achieved per the sample time
- To obtain the RPM  $\rightarrow (\Delta \text{Revolutions} / \Delta \text{time}) / (1000 * 60)$
- 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

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





# 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





# 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



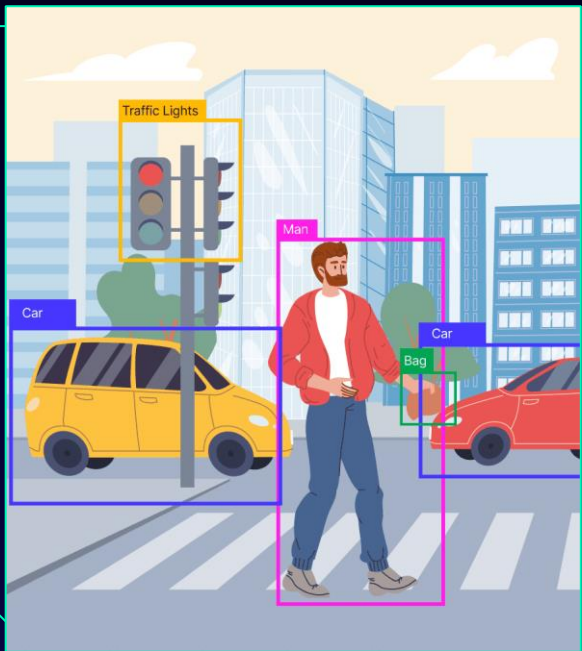


# Robot Control Result

04

Motors PID





# 05

## Vision Module





## Target of Vision Module?

05

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?

05

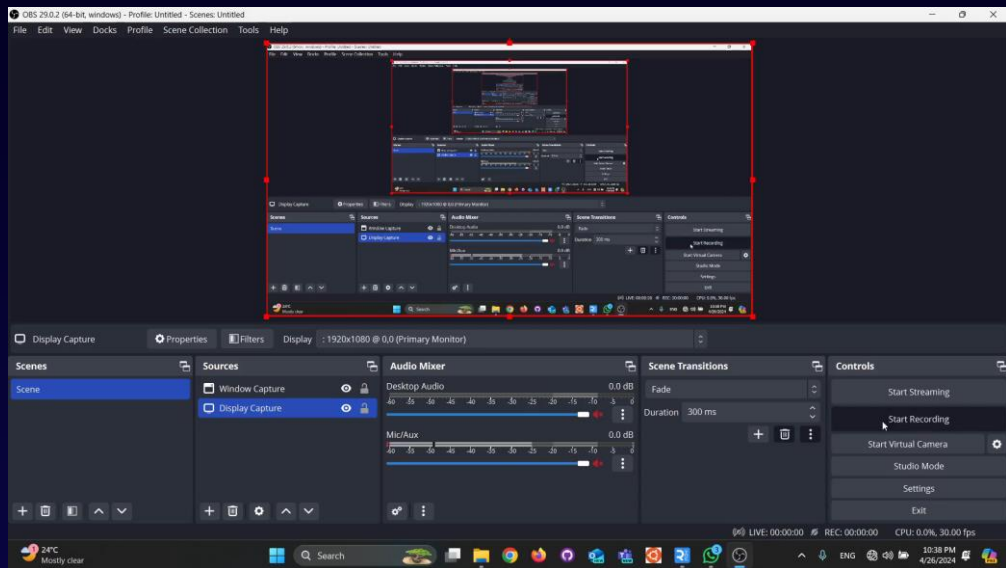
Vision Module

- 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.

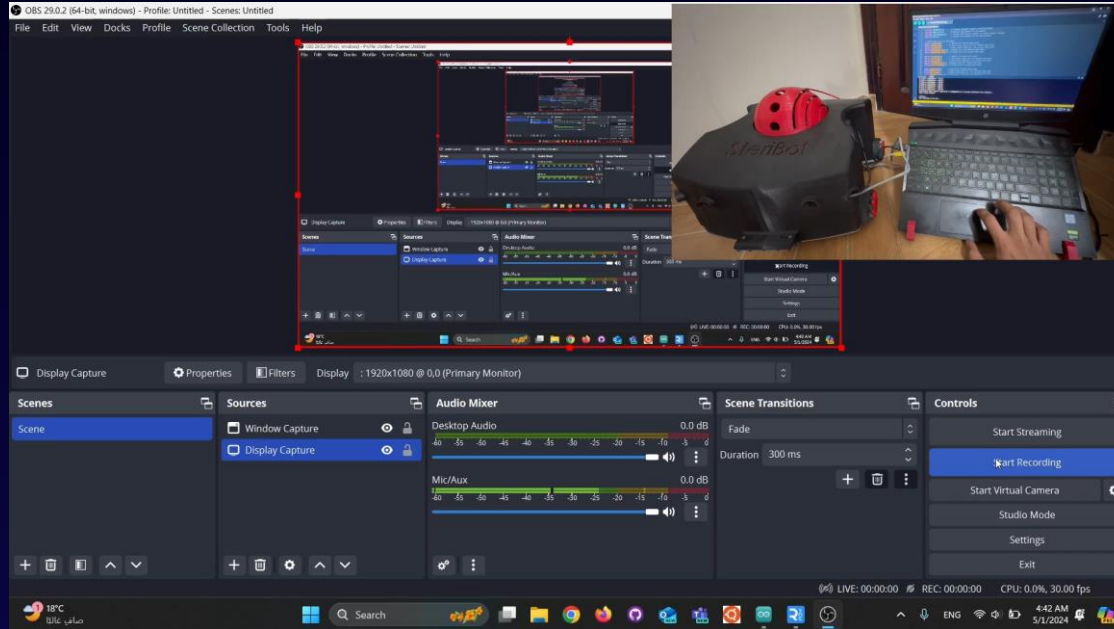


# Detection Results

05  
Vision Module



# Test Results





Extra Details ??



That's  
It For  
Today

