DESIGN MANUAL

Smart Pharmaceutical Warehouse System

May 2021

PREPARED BY

Praveen Dhananjaya (E/16/081)

Shamra Marzook (E/16/232)

Rashmika De Silva (E/16/068)

Department of Computer Engineering
University of Peradeniya

CONTENT

INTRODUCTION

Smart Pharmaceutical Warehousing is a fully automated pharmaceutical warehouse management system designed for medium to large scale pharmaceutical distributors. Carefully designed to overcome the downsides of conventional pharmaceutical warehousing methods, this system helps clients improve efficiency and profitability with minimal involvement of labor.

The system consists of three main components; an E-commerce website with inventory management system to place orders online and track inventory levels and orders, sales and deliveries, Automated Guided Vehicles (AGVs) and Automated Robotic Arms, engineered with cutting edge industrial automation techniques to handle the movements of goods inside the warehouse premises and an operating software monitors and controls the operation of these automated robots 24x7 and notifies warehouse management authorities about the progress of all operations.

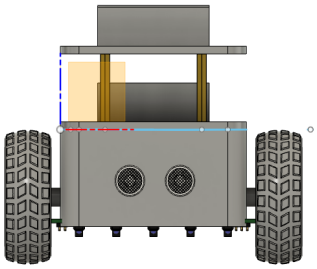
FEATURES

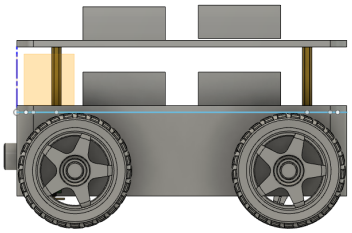
- Compact guided vehicle (AGV) and Robotic Arm designs
- Robust guidance system with optimized automated algorithms
- User-friendly controller software that monitors and controls all operations of the system
- Fully fledged E-commerce website
- An inventory management system to keep track of inventory levels and sales
- Emergency stopping mechanism for robots

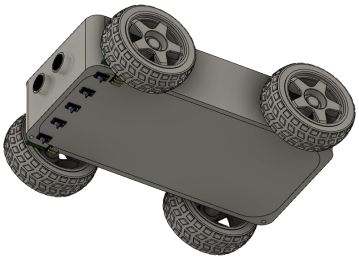
Robot Assembly

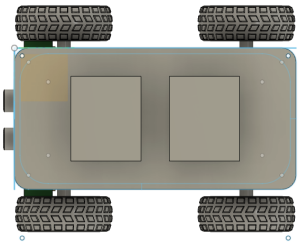
AGV

AGV Cad Model





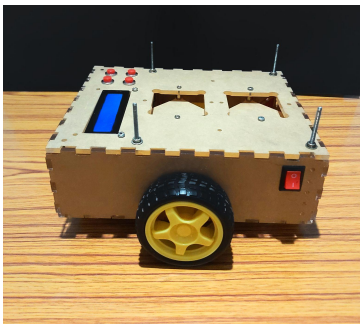


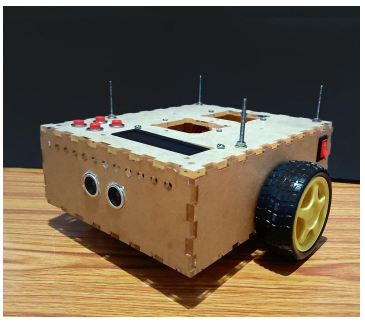


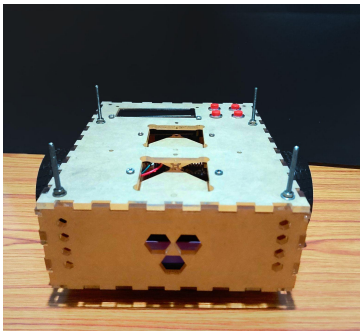
This is AGV design. It can carry multiple goods at a time.In front there is a sonar base proximity sensor. This is a 4 wheel drive vehicle. An IR base capturing method to capture the warehouse line and navigate to the destination. All communication is done by wifi base MQTT . This model manufacturing method is 3d printing.

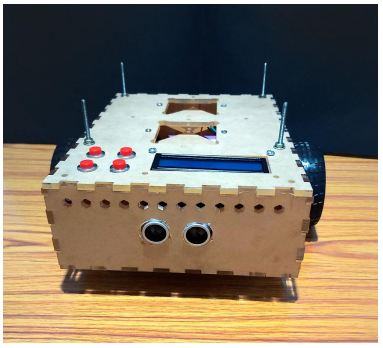
Cad link:-
https://github.com/cepdnaclk/e16-3yp-smart-pharmaceutical-warehousing/blob/main/Hardware/AGV/model/AGV.f3z

Acrylic based AGV



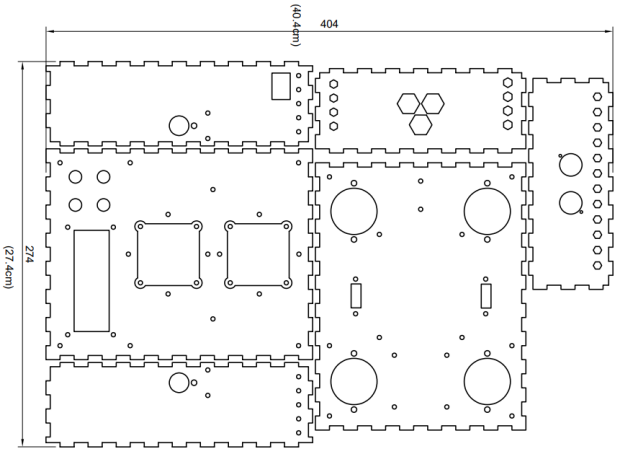






Compact AGV design . All the components are perfectly assembled into the system. Components

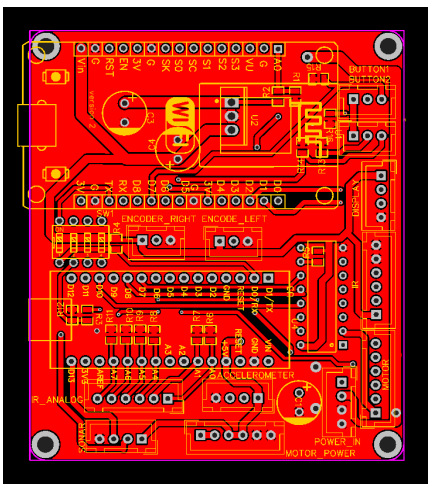
- Acrylic 28cm by 41cm , thickness 3mm or 4mm
- Two 12v motors with gearbox and wheel
- Two encoder and encoder wheel
- Tcrt5000 by 12 sensors
- Two 3.7v battery
- Sonar sensor
- Accelerometer
- 74hc595 shift register
- L298 motor drive
- Custom pcb board
- 3mm Spacers and nut and bolt

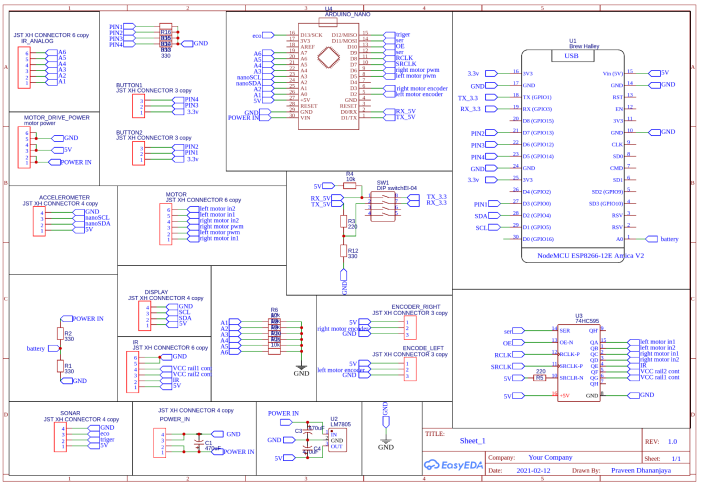


This is the AGV chassis design made with acrylic.
https://github.com/cepdnaclk/e16-3yp-smart-pharmaceutical-warehousing/blob/main/Hardware/AGV/model/agv_27cm_by_40cm.dxf

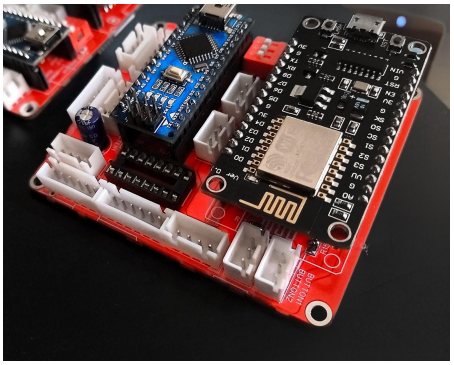
Custom PCB design

In the PCB design, all component names are given.





circuit:-https://github.com/cepdnaclk/e16-3yp-smart-pharmaceutical-warehousing/blob/main/Hardware/AGV/circuit/PCB_3y_proj_emb_agv_2021-05-28.json

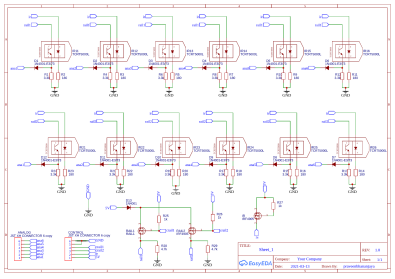


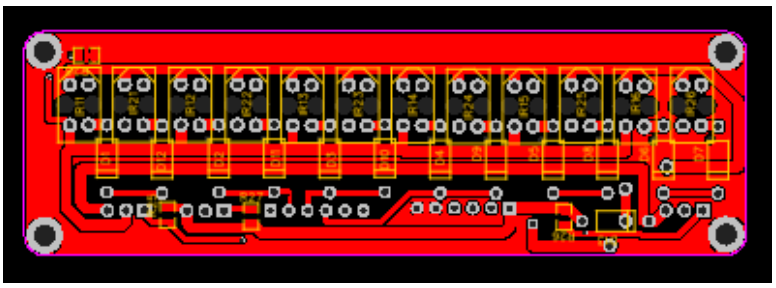
All electronic components are assembled
.

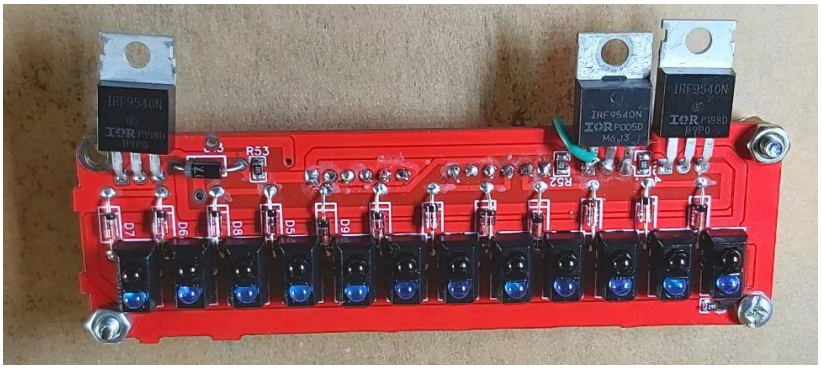
- Arduino Nano
- Esp12E
- JTS connectors
- 5V regulator
- Capacitors
- Shift registers

Every connector has a name tag.

Customly manufactured IR base sensors are used to capture floor features.

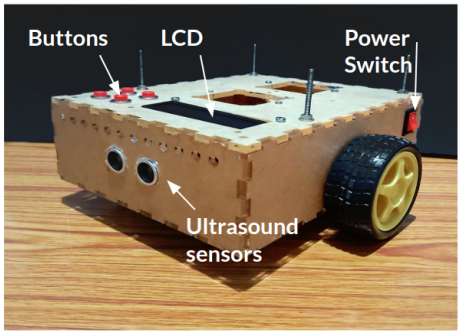


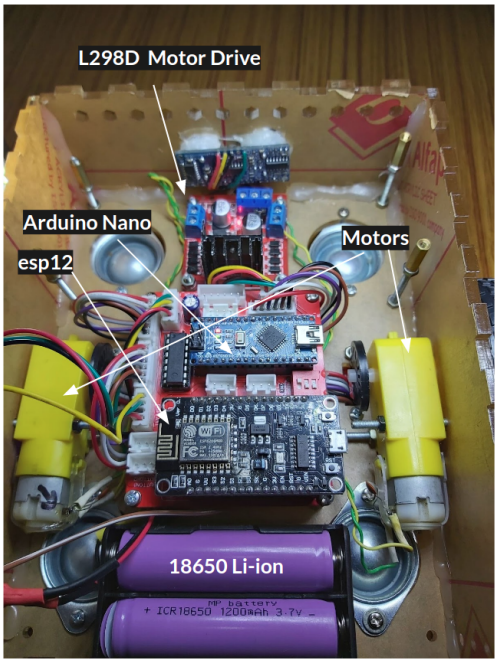




circuit:-https://github.com/cepdnaclk/e16-3yp-smart-Pharmaceutical-warehousing/tree/main/Hardware/AGV/circuit

Assembly

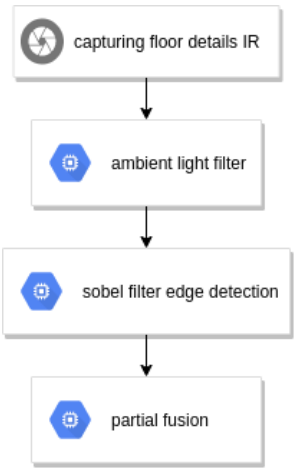


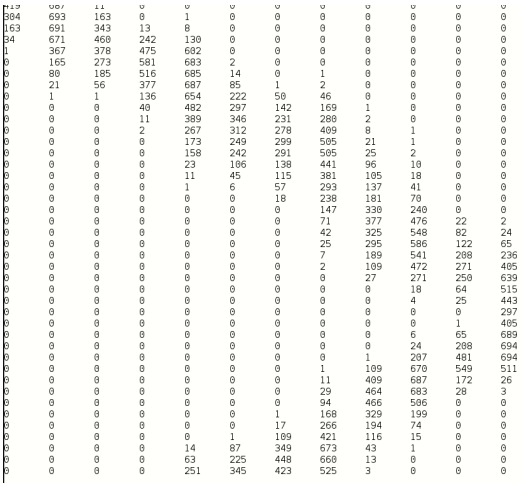


Software

- Two layer architecture behavior and hardware control
  - Behavior layer - esp12 ,
    - Wifi communication mqtt
    - LCD monitor
    - User button
    - Navigation map
  - Hardware layer - arduino nano
    - IR array
    - Sonar
    - Accelerometer
    - Motor control
    - Shift register
- single core base time scheduling multiprocessing architecture.

IR sensor data filtering



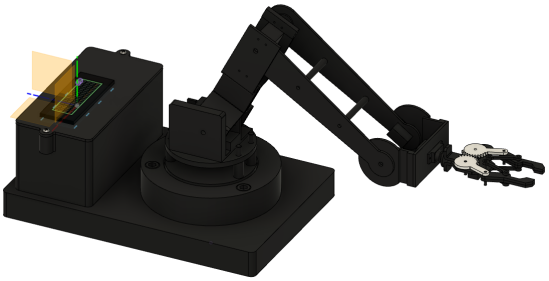


Firmware installation are done by arduino IDE
Code:-

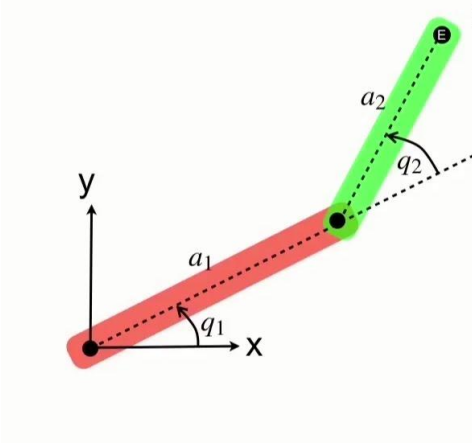
- nano:-https://github.com/cepdnaclk/e16-3yp-smart-pharmaceutical-warehousing/tree/main/Hardware/AGV/code/agv
- esp12:-https://github.com/cepdnaclk/e16-3yp-smart-pharmaceutical-warehousing/tree/main/Hardware/AGV/code/agv_esp

Robot Arm

In the warehouse , good pick and placement is done by robot arm .



This is two joint inverse kinematics with parallel joint end system. The arm has three links a1 , a2 and a3 . The a1 link is connected to base at (0,0) and the a2 link is connected to a1 at (x1,y1) and a3 is connected to a2 at **(X,Y)** . To simplify calculation consider only a1 and a2.



First calculate the forward inverse of the mechanism.

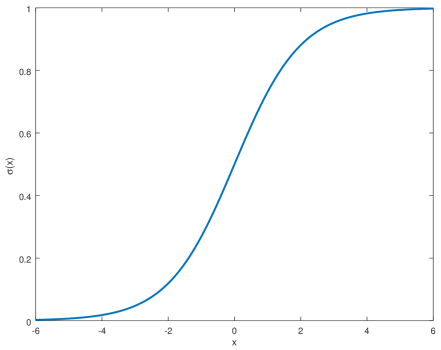
$$\cos q_2 = \frac{x^2 + y^2 - a_1^2 - a_2^2}{2a_1 a_2}$$

$$q_2 = \cos^{-1} \frac{x^2 + y^2 - a_1^2 - a_2^2}{2a_1 a_2}$$

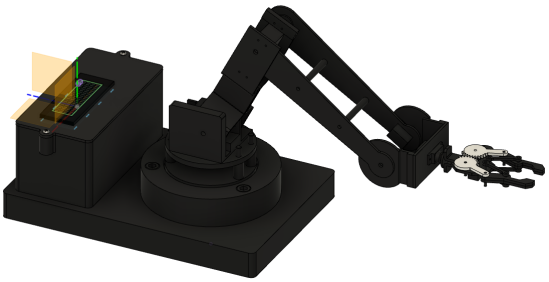
$$q_1 = \tan^{-1} \frac{y}{x} - \tan^{-1} \frac{a_2 \sin q_2}{a_1 + a_2 \cos q_2}$$

Using this function, the robot can calculate q1 and q2 angles for given X,Y . Because robot links move using servo motors(0 - 180). Using sigmoid transferring function, the robot arm moves between coordination.

$$S(x) = \frac{1}{1 + e^{-x}}$$



Arm Cad Model

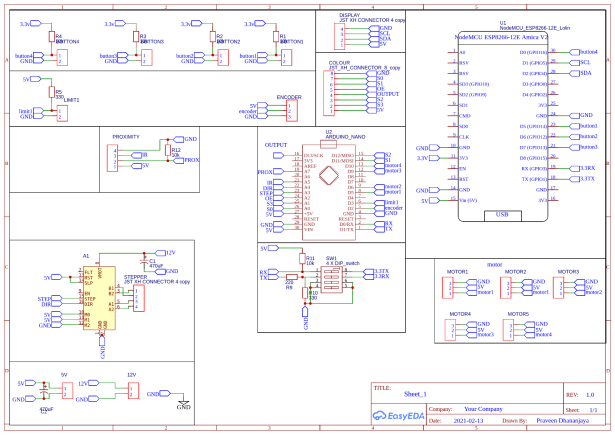


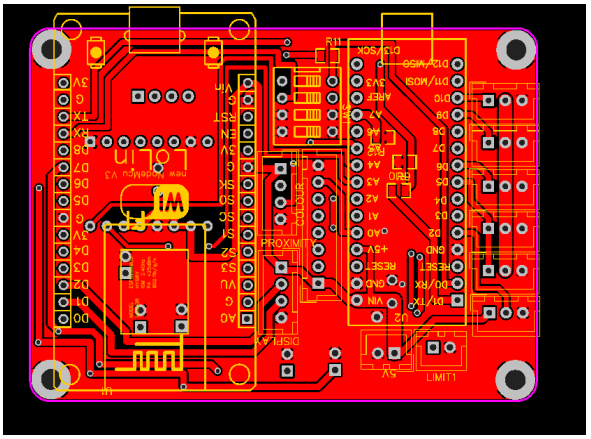
This is the Robot arm cad model .

- 5 servo motors
- 1 stepper motor
- 2 Linear guild
- Lead screw
- Color sponsor
- Limit switch
- Lcd display
- dv8856 stepper motor driver
- 805 resistor

Cad file : -
https://github.com/cepdnaclk/e16-3yp-smart-pharmaceutical-warehousing/blob/main/Hardware/Arm/model/Robotic_Arm_Final%20v22.f3z

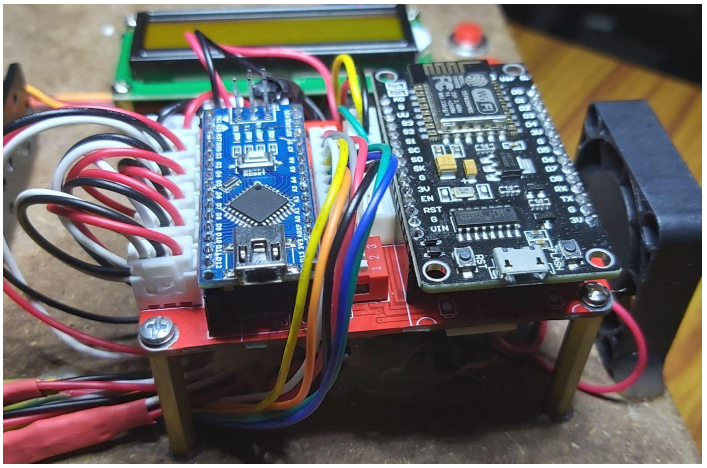
Custom PCB design



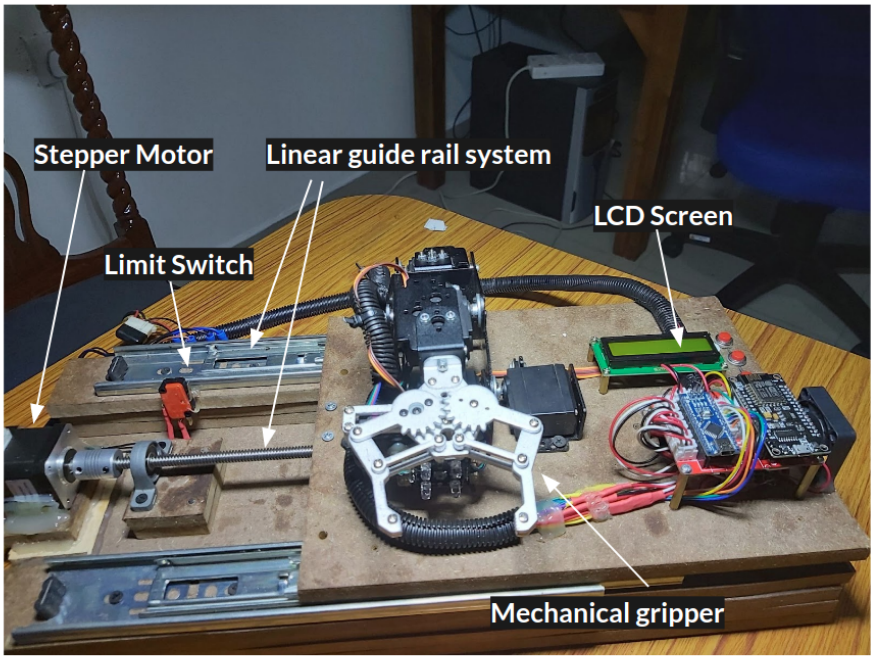


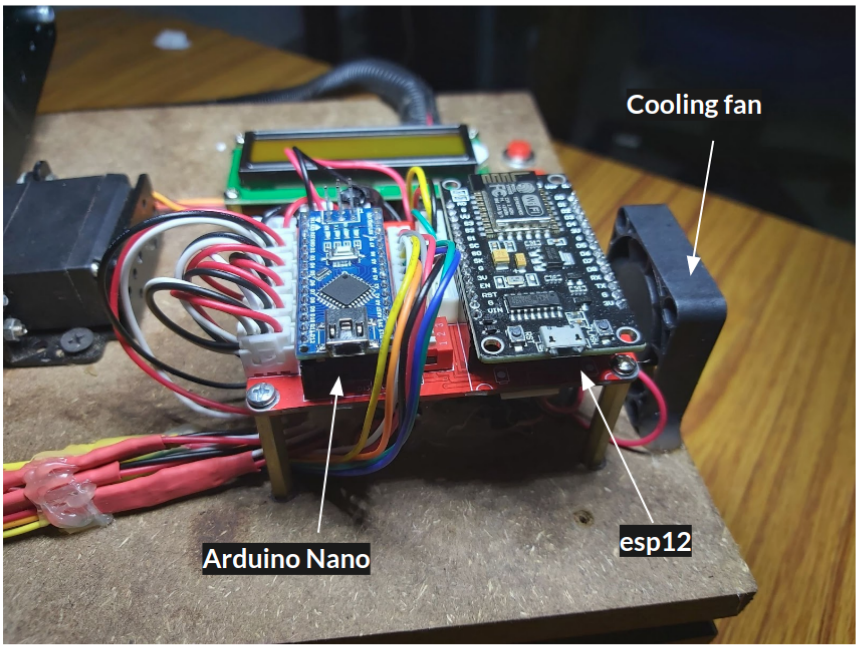
In the PCB design, all component names are given. Every connector has a name tag.

Use JST connectors for stability and reliability .
circuit:-https://github.com/cepdnaclk/e16-3yp-smart-pharmaceutical-warehousing/blob/main/Hardware/Arm/circuit/PCB_3y_proj_emb_arm_3_2021-05-28.json



Assembly


Stepper Motor
Linear guide rail system
LCD Screen
Limit Switch
Mechanical gripper


Cooling fan
Arduino Nano
esp12

Software

- Two layer architecture behavior and hardware control
    - Behavior layer - esp12 ,
        - Wifi communication mqtt
        - LCD monitor
        - User button
        - Rack position
    - Hardware layer - arduino nano
        - 5 servos
        - 1 stepper motor
        - Limit switch calibration
- single core base time scheduling multiprocessing architecture.

Code:-

- Nano:-https://github.com/cepdnaclk/e16-3yp-smart-pharmaceutical-warehousing/tree/main/Hardware/Arm/code/arm

- Esp12e:-https://github.com/cepdnaclk/e16-3yp-smart-pharmaceutical-warehousing/tree/main/Hardware/Arm/code/arm_esp/arm_esp

OPERATOR CONTROL APPLICATION

Overview

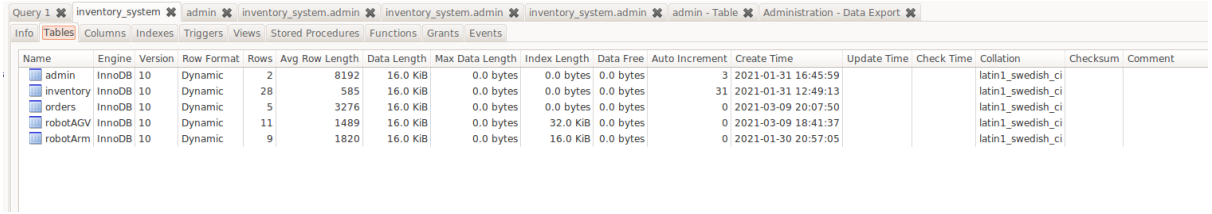
The following details guide the implementer to set up the operator control system. The operator control application contains two parts as

- Smart Pharmaceutical Warehouse Management System Application (Operator GUI)
- Inventory management database system

Database System

Introduction

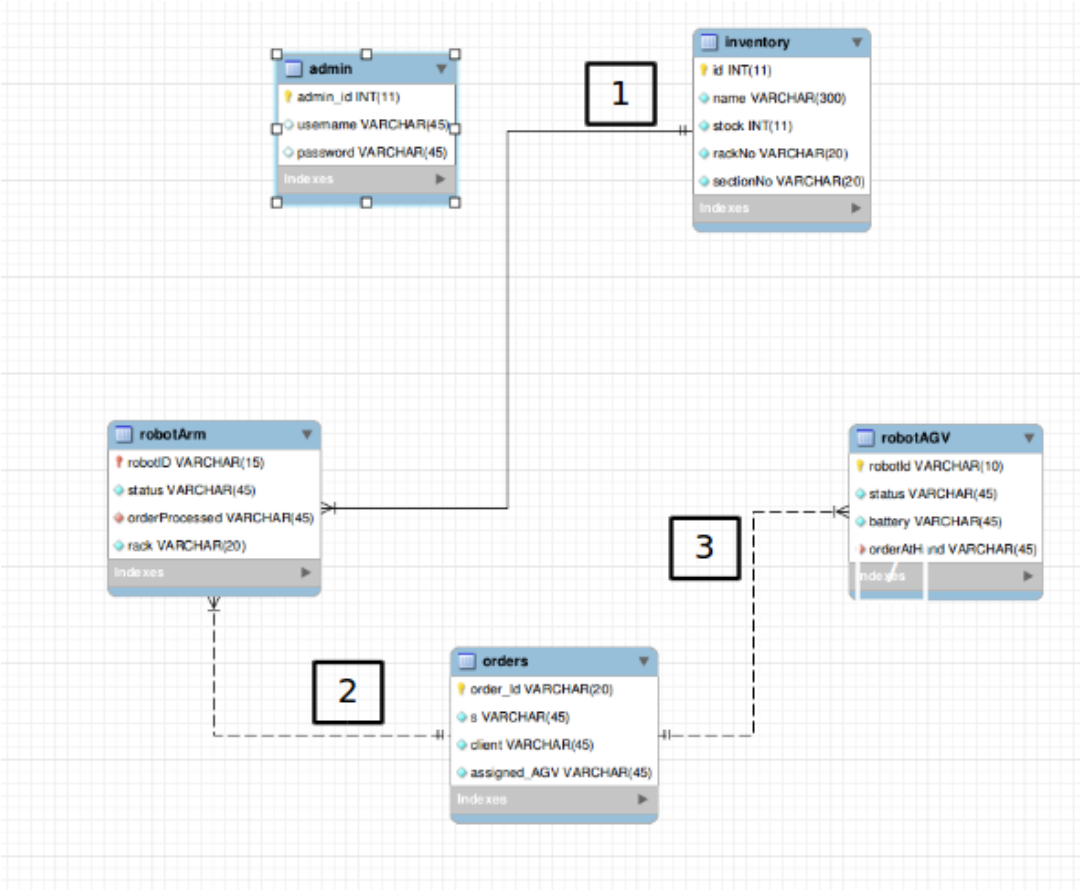
The database system has the following tables included



1. admin - contains the username and password of the operator
2. inventory - details of the stock available in the warehouse
3. orders - details of the orders placed by the customers
4. robotAGV - details of the AGVs the operate in the warehouse
5. robotArm - details of the robot arms that operate in the warehouse

The database system is designed to be managed by the operator control GUI.

The entity–relationship model (or ER model) is shown below.



Relationships are created between the tables as shown in the er-diagram.

1. robotArm - inventory : the relationship is created based on the rack no where the arm is deployed.
2. robotArm - orders : the relationship is created based on the order that is being processed by the Arm at present.
3. robotAGV - order : the relationship is created based on the order that is being processed by the Arm at present and the assigned AGV to process that order.

Configuring Steps

- As the first step, install MySQL Workbench in your system.

- Next, clone the following repository from github.
  https://github.com/cepdnaclk/e16-3yp-smart-pharmaceutical-warehousing

- Then change the directory to, /Operator Interface

```
cd /Operator Interface
```

- The **warehouse_database.sql** file contains the database system.

- Next, in the MySQL workbench create a new schema and import the warehouse_database.sql file.

- The database is now imported in the Workbench

Controller Interface

Introduction

Controller software monitors and controls all operations in the warehouse management. It also controls the automated algorithms that perform the calculations of optimum path planning and process scheduling

Configuring Steps

- As the first step, install and set up **python3** and **pip3** in your system.

- Next, clone the following repository from github.
    - https://github.com/cepdnaclk/e16-3yp-smart-pharmaceutical-warehousing

- Then change the directory to, /Operator Interface/Operator GUI/gui/

```
cd /Operator Interface/Operator GUI/gui/
```

- Next, Open a terminal in the above directory and run the following command to install project dependencies.

```
pip3 install -r req.txt
```

- To start the gui run the following command

```
python3  main.py
```

Algorithm

Automated Algorithms for optimum path planning and process scheduling

- Process Scheduling Algorithm
    - Orders are arranged in the first come first served basis
    - AGVs are assigned with jobs according to payload capacity
    - Optimum path is modeled using Traveling Salesman Problem
- Path Planning and Collision Avoidance
    - Floor map is abstracted into a graph
    - Shortest path - Dijkstra's Shortest Path algorithm
    - Two-way lane system
    - Virtual Traffic Light System - avoid collisions at intersections

Testing and functioning of the Operator GUI

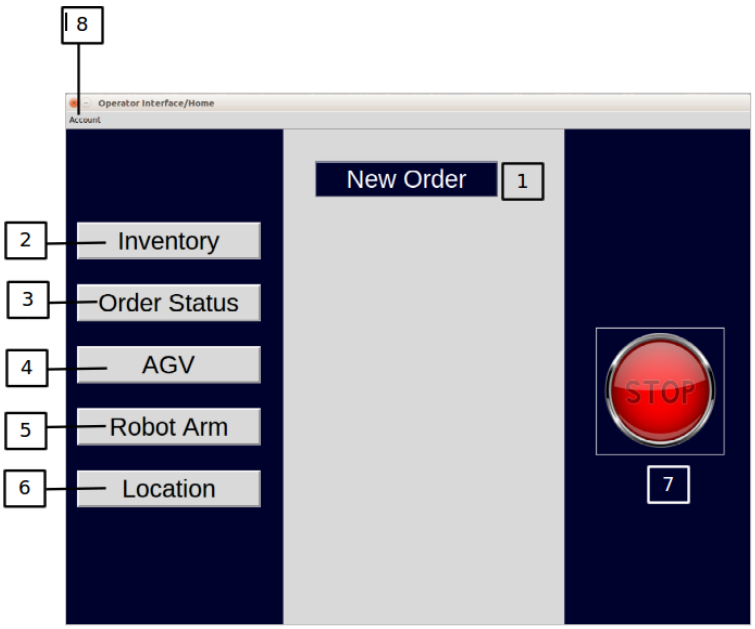
- The main tab of the operator interface is shown below

- To login into the system by clicking the 'Account/Login' in the main tab. You will be directed to this tab.

Enter the Username and Password and click the **'Login'** button.

- Login using 'admin' as both the username and the password.
- Make sure to change the login details after the initial setting.

- You will be directed to this tab.



Here the functions and operations done by the application are displayed.

1. Display of the new order placed by customer - Click **'New Order'** button
2. Inventory management of the warehouse - Click **'Inventory'** button
3. Monitoring of the orders - Click **'Order Status'** button
    a. View the list of orders placed by customers and their details
    b. View stock details
    c. Add new product
    d. Update stock of product
4. Monitoring of the AGVs - Click **'AGV'** button
    a. View the list of AGVs.
    b. Enter the **AGV ID** to get the details such as
        i. status
        ii. order that being processed currently
        iii. battery life

        of each of them.
5. Monitoring of Robot Arms - Click '**Robot Arm**' button
    a. View the list of robot arms.
    b. Enter the **Arm ID** to get the details as
        i. status
        ii. order being processed
        iii. battery life

        of each of them.
6. Locate position of AGVs in the warehouse - Click **'Location'** button
    a. View the movement and path followed by the AGVs inorder to process the orders placed by the customer in the floor map.
7. To close the floor map tab, click **'Stop'** in the Home tab.
8. Click the **Account** menu in the Home tab and select 'Logout' to logout from the account and **'Exit'** to exit from the application.

WEB APPLICATION

Overview

The following details guide the implementer to setup and deploy the web application. Further it also contains details how the administrator of the website can manage it.

Deployment steps

- As the first step, clone the following repository from github.
  https://github.com/cepdnaclk/e16-3yp-smart-pharmaceutical-warehousing/tree/main/Web%20application/aws

- Install and create a virtual environment

```
sudo apt-get install python3-venv
```

```
python3 -m venv env
```

- Activate the virtual environment and install the following libraries and dependencies.

```
source env/bin/activate
```

  - ❖ Install Django

```
pip3 install Django
```

  - ❖ Install Gunicorn (wisgi Interface)

```
pip3 install gunicorn
```

  - ❖ Install Nginx

```
sudo apt-get install nginx
```

  - ❖ Install the supervisor to keep the django website up and running

```
sudo apt-get install supervisor
```

❖ Then configure the supervisor as follows

```
cd /etc/supervisor/conf.d/
sudo touch gunicorn.conf
sudo nano gunicorn.conf
```

- Then add the following configurations in the file

```
# put these inside the file
[program:gunicorn]
directory=/home/ubuntu/e16-3yp-smart-pharmaceutical-wareho
using/Web application/aws/static_root/
command=/home/ubuntu/env/bin/gunicorn --workers 3 --bind
unix:/home/ubuntu/Sock/app.sock
MainWebApp.wsgi:application
autostart=true
autorestart=true
stderr_logfile=/var/log/gunicorn/gunicorn.err.log
stdout_logfile=/var/log/gunicorn/gunicorn.out.log

[group:guni]
programs:gunicorn
```

- After that, run the following commands to make necessary directories and run the configuration file in supervisor.

```
sudo supervisorctl reread
sudo supervisorctl update
```

```
sudo supervisorctl reread
sudo supervisorctl update
```

- Then configure the Nginx server,

```
cd /etc/nginx/sites-available/
sudo touch django.conf
```

- After that, add the following code to the created file.

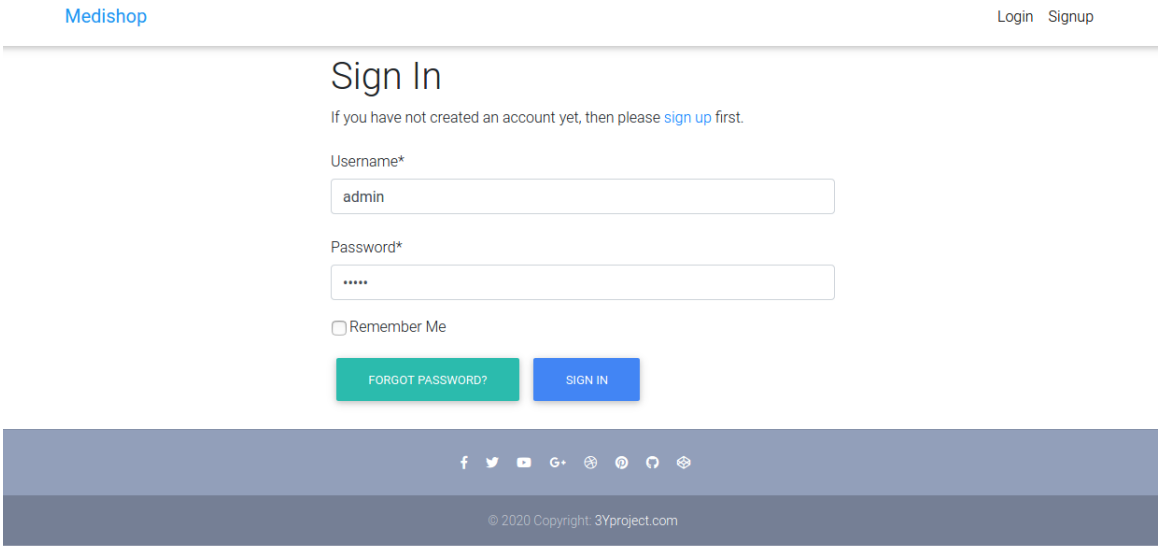
```
server {
    listen 80;
    server_name 54.87.128.186;

    location / {
            include proxy_params;
            proxy_pass http://unix:/home/ubuntu/Sock/app.sock;
    }

    location /static/ {
            autoindex on;
            alias
'/home/ubuntu/e16-3yp-smart-pharmaceutical-warehousing/Web
application/aws/static_root/';
    }
}
```

- As the final step, change the IP address of the server to the public IP and restart the Nginx server.

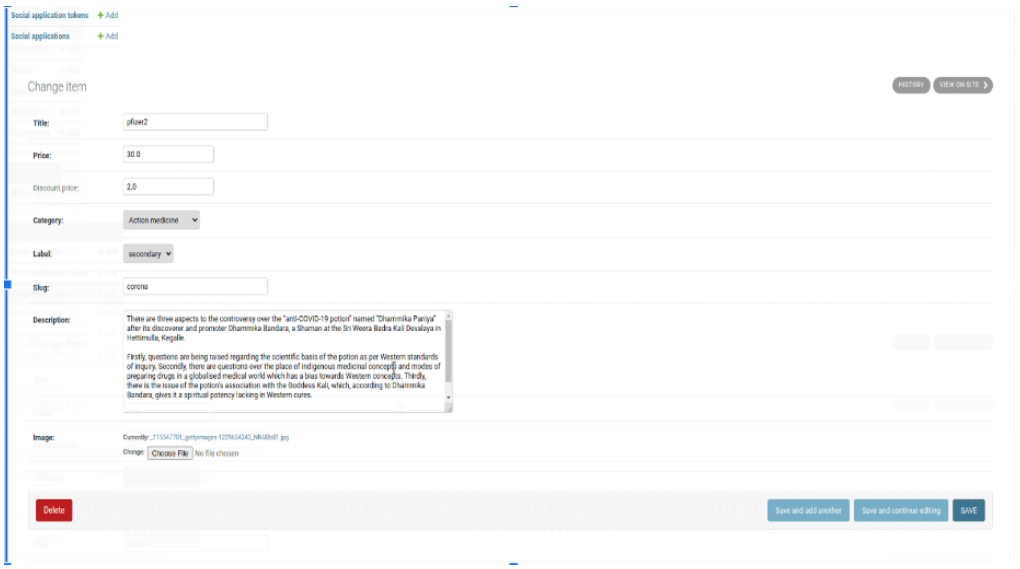
```
sudo nginx -t
sudo ln django.conf /etc/nginx/sites-enabled/
```

Testing and functioning of the web application

- Enter the Username and Password and click the **SIGN IN** button.
    - Sign in using ‘admin’ as both the username and the password

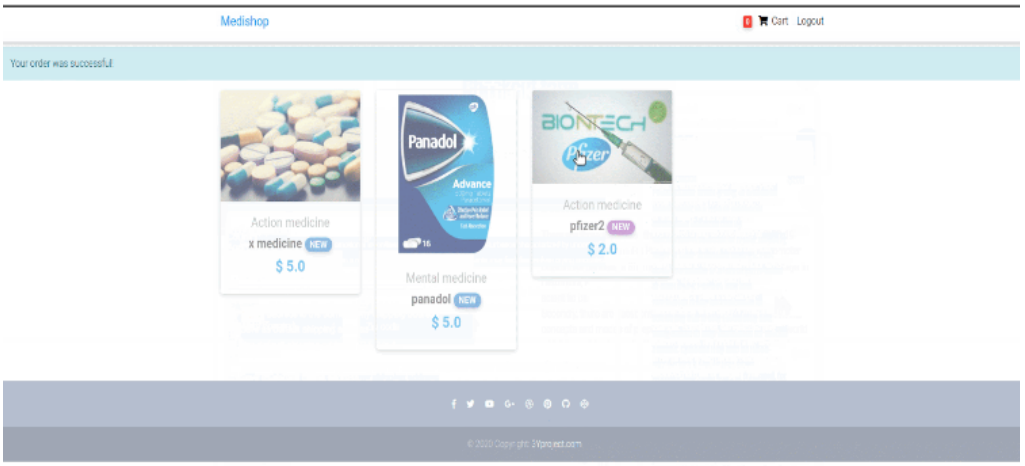


- Once you sign in into the account you can
    - Manage the website
    - Add new items
    - Update them.



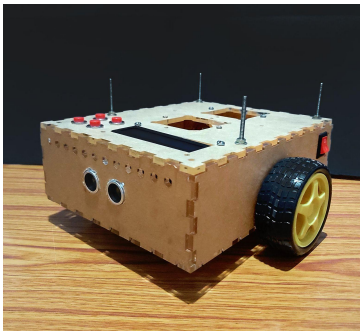
- Make sure to change the login details after the initial setting.

- Then you can view the website with the items added.



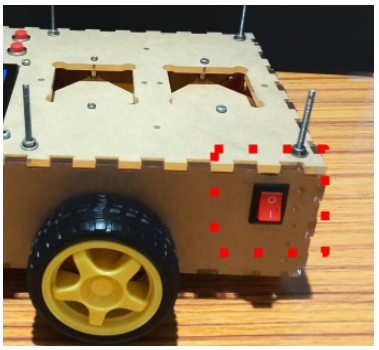
DEVICE INSTALLATION

AGV



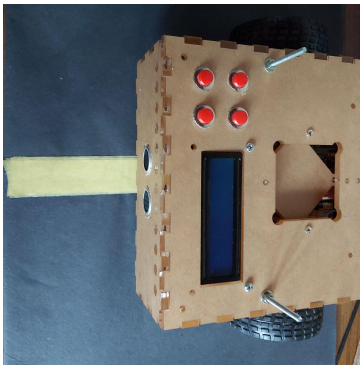
Warring | don't open and don't touch internal electronics

This is an automated guided vehicle. Its tyres must touch the ground. And check whether there are no cracks or defects.



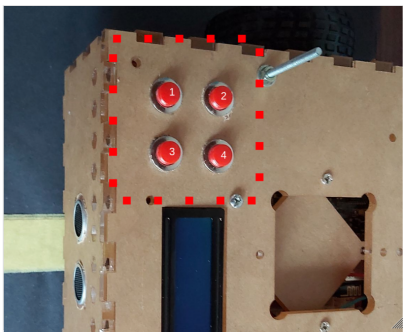
Power on button is placed In left side panel read end. Put the power switch to 1 poison to turn on. Then LCD display will turn and show " Hello AGV , ID:X " ,

ID:X is the AGV is the identification number.



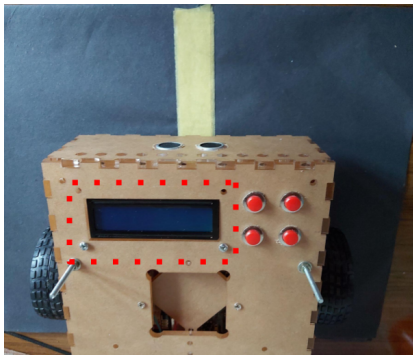
if there's any smoke turn off immediately and contact our assistance

Please place AGV in the warehouse lines. Then , the AGV will automatically calibrate to the warehouse line system. During this process AGV will move back and forward and rotate clockwise and anticlockwise a few times. forward and rotate clockwise and anticlockwise a few times.

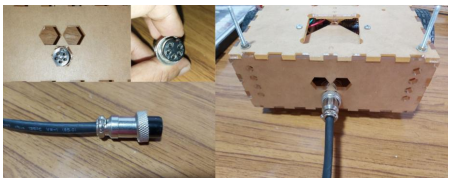


These are controller buttons.

1. OK / Select button
2. Back or Restart
    a. Single click for back
    b. Hold press will restart the system
3. Menu down
4. Menu up

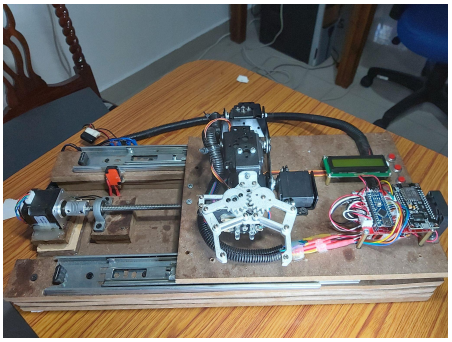


This is the user LCD display. All AGV current status and error code will show. In addition to that sensor details and calibration report can be reviewed.



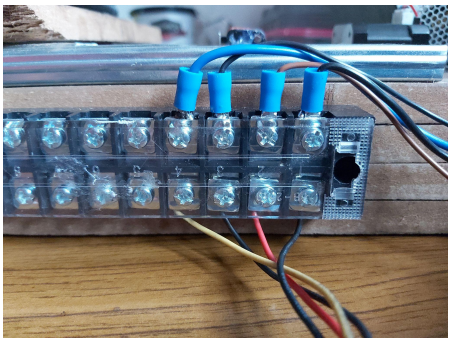
For charging purposes, the user should align the charging cable notch to charging port notch. The charger is an Industrial 5 pin connector . Charging port is located in the rear side of the AGV.

Robot Arm



This is a pick and place robot arm. It must be firmly attached to the ground. Otherwise there will be noise and vibrations. And check whether there are no cracks or defects.

Warring | don't open and don't touch internal electronics



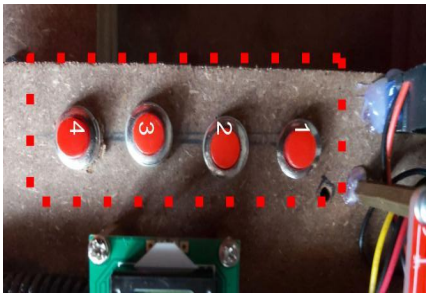
Power delivery system of the Robot arm. You should wire colour match as follows.

1. Black to Black - Ground wire
2. Brown to Red - 5v line
3. Yellow to Blue - 12V line

And power on the System, The Robotic arm starts to calibrate its sensors and actuators . It will take several seconds .

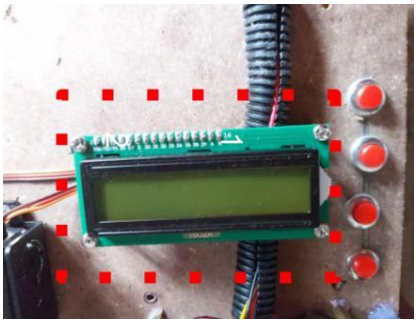
During place stay away from Robot ARM

if there's any smoke turn off immediately and contact our assistance



These are controller buttons.

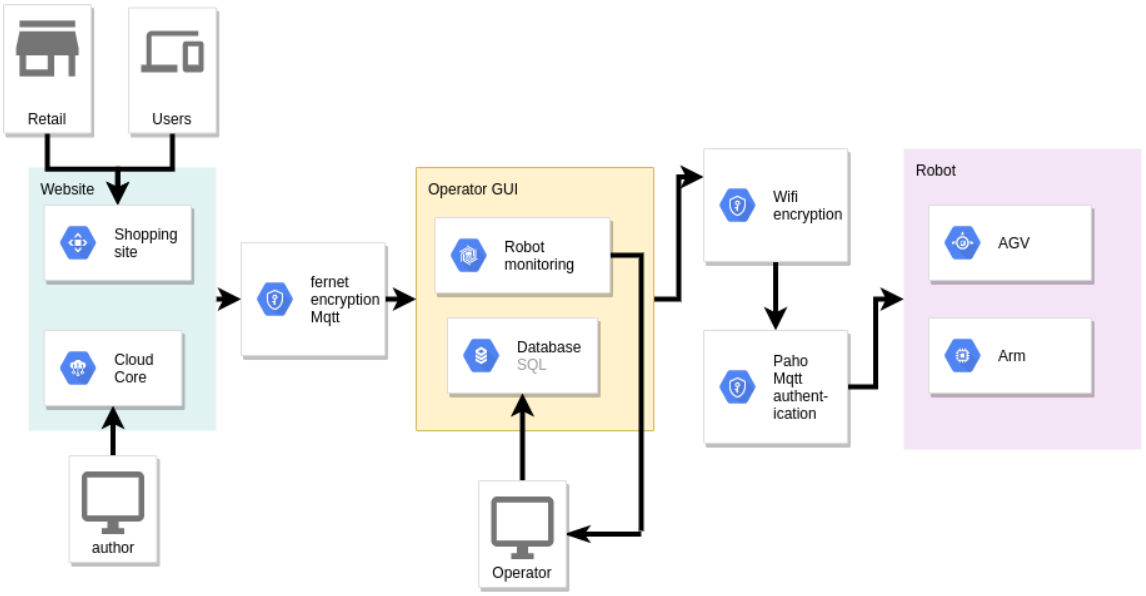
1. OK / Select button
2. Back or Restart
    a. Single click for back
    b. Hold press will restart the system
3. Menu down
4. Menu up



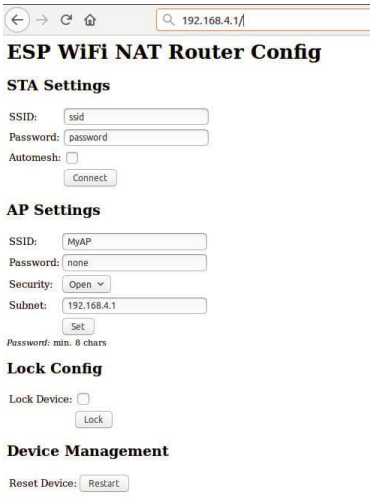
This is the user LCD display. All AGV current status and error code will show. In addition to that sensor details and calibration report can be reviewed.

DEPLOYING MQTT BROKER

This is the overall mqtt interaction between the rest of the system.



AGV and ROBOT wifi setting



In the first boot , All AGV and ROBOT ARM IN AP mode operator can login to robot using

SSID : AGVX or ARMX
Password : Admin

Then the operator should add warehousing **local wifi** SSID and Password. In STA settings. Reboot the system using device management restart button.

Warring | please change the AP password details

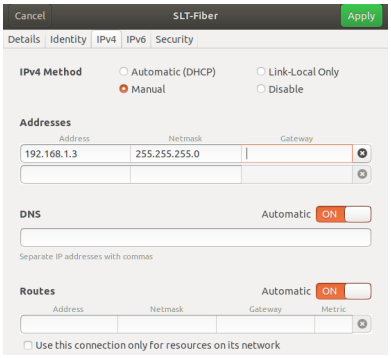
Local Mqtt broker

Ubuntu base instruction . The 1st operator should install the mqtt mosquitto.

```
sudo apt-get update
sudo apt-get install mosquitto
```

Install MQTT clients

```
sudo apt-get install mosquitto-clients
```



Operator pc Local wifi network setting change into static IP 192.168.1.3 (AGV and ARM mqtt port)
And pass the Apply button

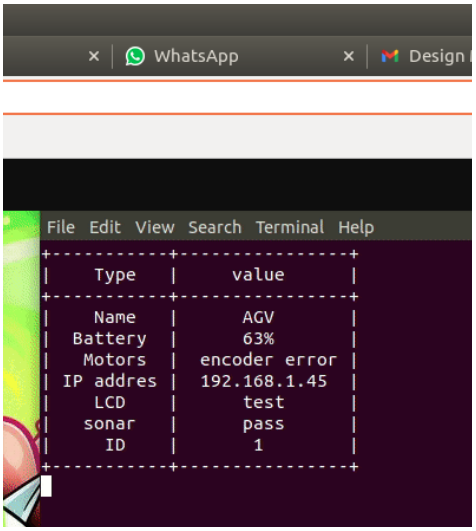
Warring | if you have any problem please contact our assistant

Communication test

For the AGV and ARM communication test . First operator should download the following files.

- AGV:-https://github.com/cepdnaclk/e16-3yp-smart-pharmaceutical-warehousing/tree/main/Software/mqtt/mqtt_agv

- ARM:-https://github.com/cepdnaclk/e16-3yp-smart-pharmaceutical-warehousing/tree/main/Software/mqtt/mqtt_arm



To run the AGV and ARM test file . You should enter this command on the Local server computer.

python arm.py or
python agv.py

Then automatically this program test robot prints detatis and diagnostic results .

Warring | if you have any problem please contact our assistant

END