Design Manual

Obstacle Robots For Swarm Robots Platform

May 2021

ISSUED BY

Department of Computer Engineering,

University of Peradeniya.

REPRESENTATIVE

E/16/369 Rusiru Thushara

E/16/366 Dinindu Thilakarathne

E/16/088 Heshan Disanayake

Content

List of Figures

1. Introduction and Background

1.1. Process Overview

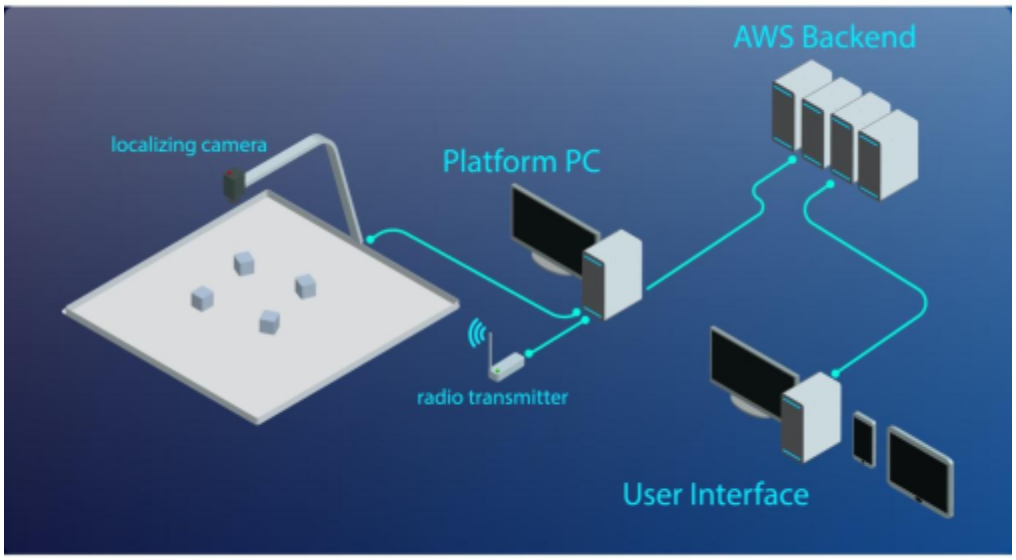


Figure 1

The system uses an overhead camera to localize the robots and the video feed is sent to the platform PC where all the algorithms are running and calculating the required parameters and the relevant data to the robots are sent using radio transmission and the other required data is sent to the server backend.

1.2. The System

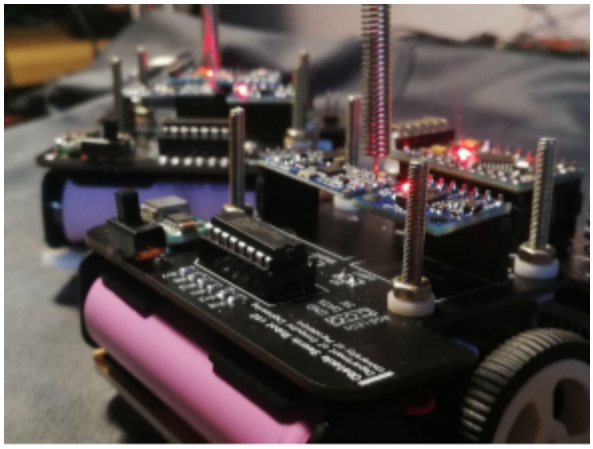
The system contains a radio transmitter module, camera module, and robots. You should have to have an arena for the given dimensions. If your arena dimensions are higher than the given dimensions you have to purchase a separate HD overhead camera module. You have to have a computer (not included with the product) with an internet connection.

2. Features

- Algorithms are capable of handling about 100 robots.
- Self homing feature when the battery levels are too low.
- Battery level monitoring feature with GUI.
- Hardware and software emergency stop buttons.
- User-friendly 3D web interface.
- User-friendly operator GUI.
- Compact robot designing.
- Highly accurate positioning of robots.
- Colour changing capability of robots.
- Automatic error detections of robots.

3. Robot Assembly

3.1. Introduction



An obstacle robot is a robot with two wheels. These robots can change their colour to any color as the need of the operator. These robots are designed according to a compact design. Assembly instructions for the robot are mentioned below.

Figure 2

3.2. Required Components

The following table shows the Items you need for a single robot assembly.

Item	Quantity
Arduino Pro Mini	1
L293D Motor Driver IC	1
HC-12 Radio Transceiver	1
MP6050 Gyroscope	1
N20 Motor (50 RPM)	2
N20 Wheel	2
N20 motor bracket	2
1K SMD resistor (805 Size)	8
220ohm SMD resistor (805 Size)	3
Male Pin Headers	2 strips

Female Pin Headers	2 strips
RGB Led	1
NeoPixel strip (30cm)	1
5V Switchmode Regulator	1
50mm M2 bolt	4
50mm M2 nuts	12

3.3. PCB manufacturing

- You can find the schematic pdf at the following link. https://github.com/cepdnaclk/e16-3yp-obstacle-bots-for-swarm-robots/blob/master/Fabrication/PCB%20Design/Schematic/Schematic_project_01_2021-05-28.pdf
- You can find the build Gerber file from the following link. https://github.com/cepdnaclk/e16-3yp-obstacle-bots-for-swarm-robots/blob/master/Fabrication/PCB%20Design/Gerber/Gerber_v3-2.zip

3.4. Assembling the PCB

- This is the PCB of the robot. Let's see how to assemble components step by step.

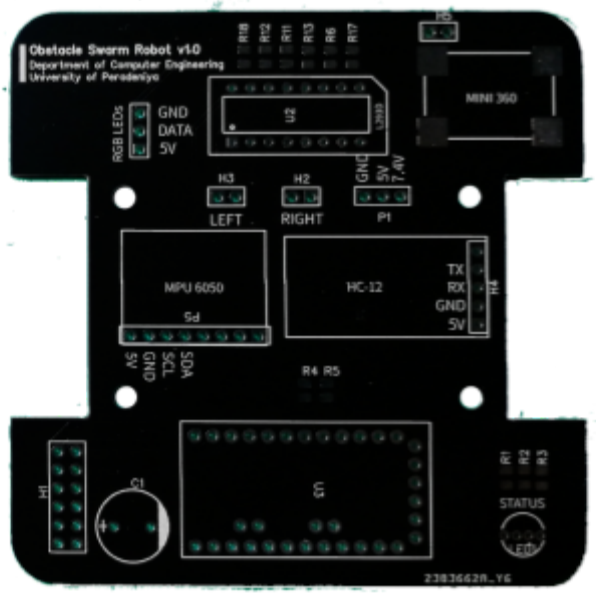


Figure 3

- Placing the components

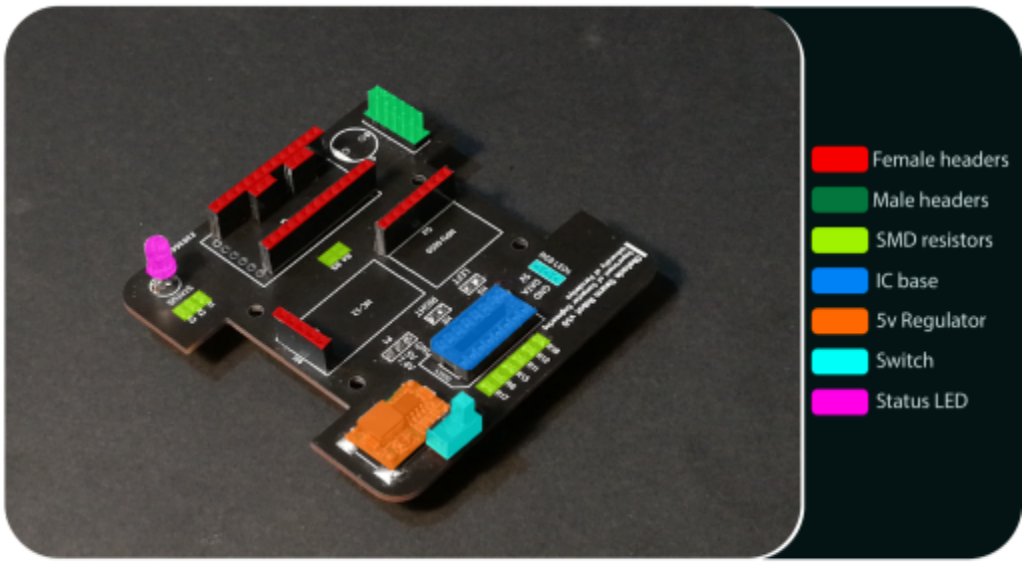
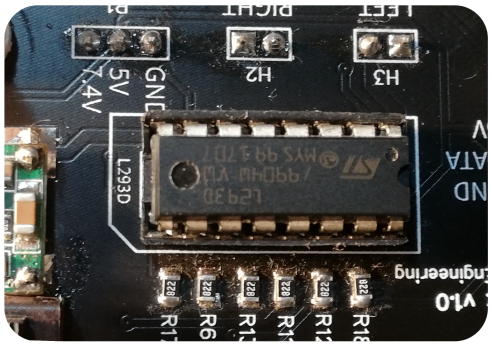


Figure 4

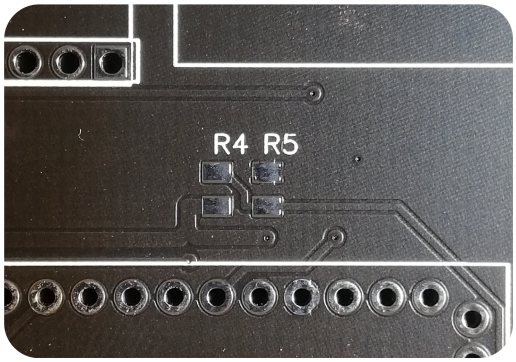
As you can see in the image, place the components accordingly.
Let's take a close view of some components.



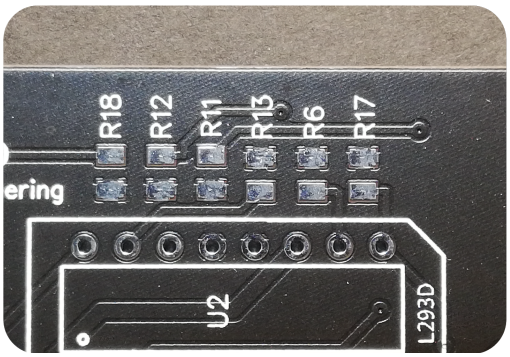
L293D Motor driver

When placing the motor driver pay attention to the orientation of the IC. the notch on the IC should be placed as shown in the image.
It is important to match the pin configuration.

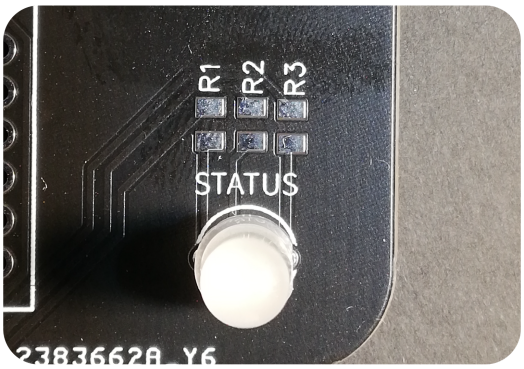
SMD resistor placement



Use 1K ohm resistor for R18, R12, R11, R13, R6, R17.



Use 1K ohm resistors for R4, R5



Use 220-ohm resistors for R1, R2, R3.

RGB status LED

As shown in the diagram the RGB led that is used is a common cathode led. Use this configuration when placing the RGB LED

5v Regulator & Switch

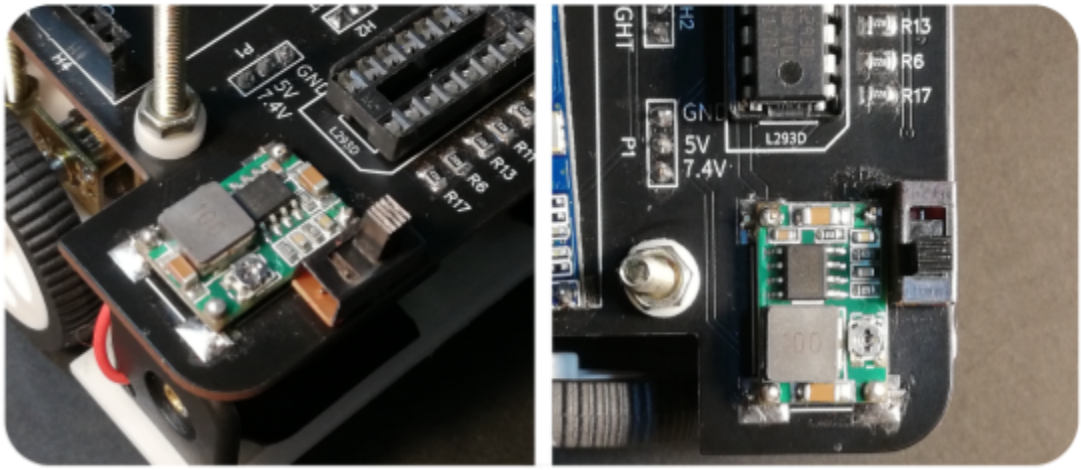
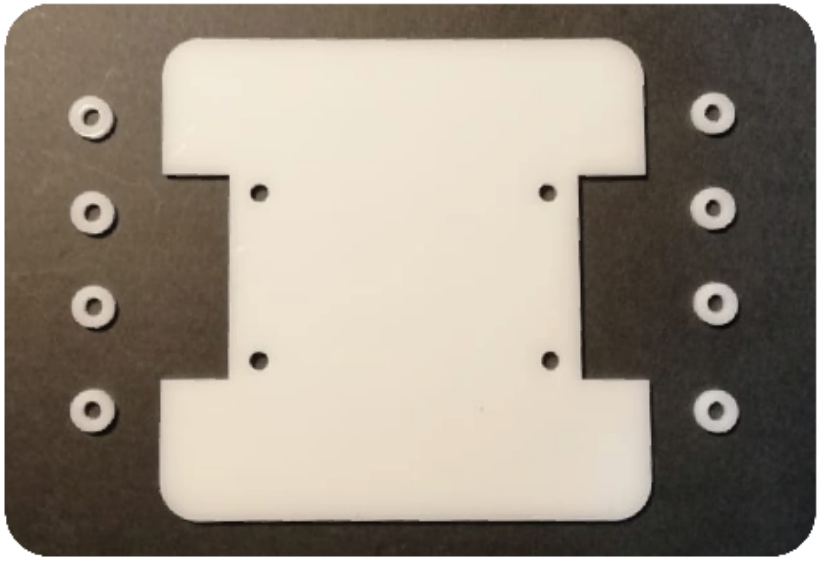


Figure 5

As shown in the figure, solder the 5v Regulator and the switch to the board.

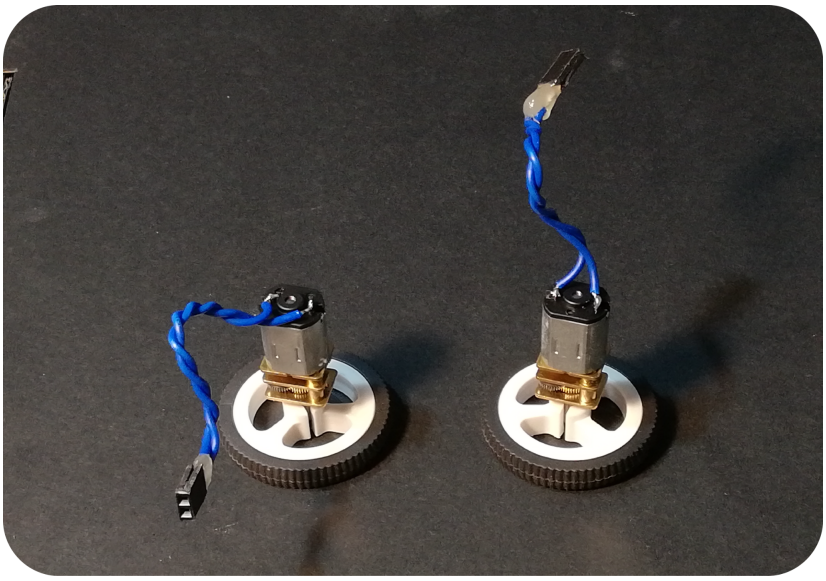
3.5. Laser Cutting

- You have to create the base plate according to the given dimensions.
- If you are willing to laser cut the part you can use the provided dwf file.



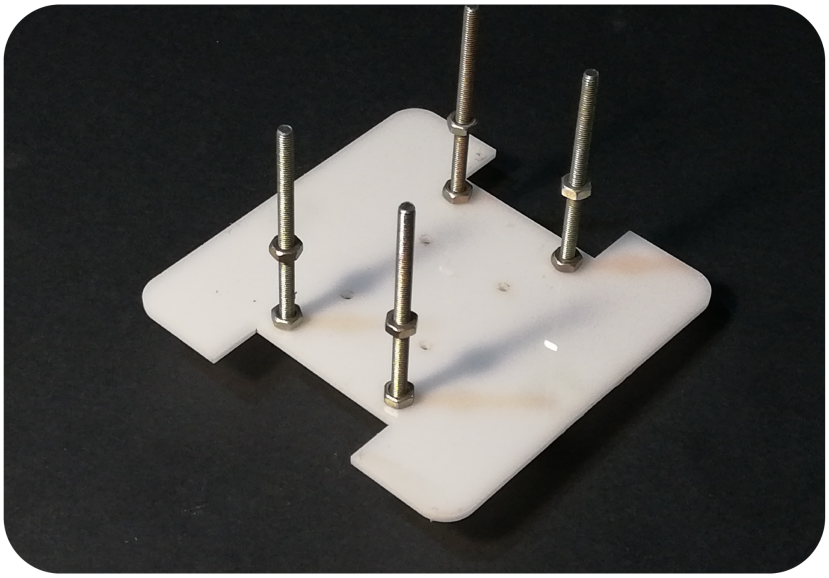
3.6. Final Assembly

- **Prepare the Motor and Wheels.**



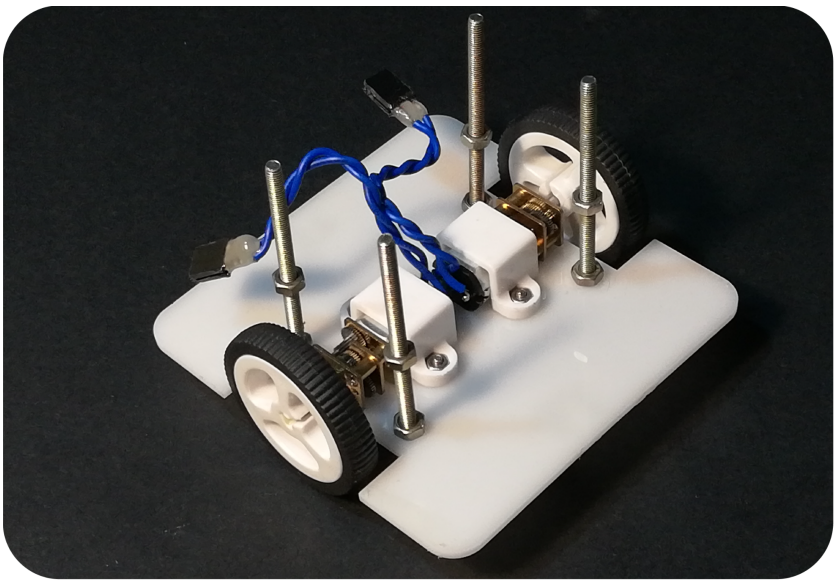
Solder wires to the motors with an approximate length of 5cm. Also solder a 2pin female header part to plug it to the main board.

Prepare the base plate



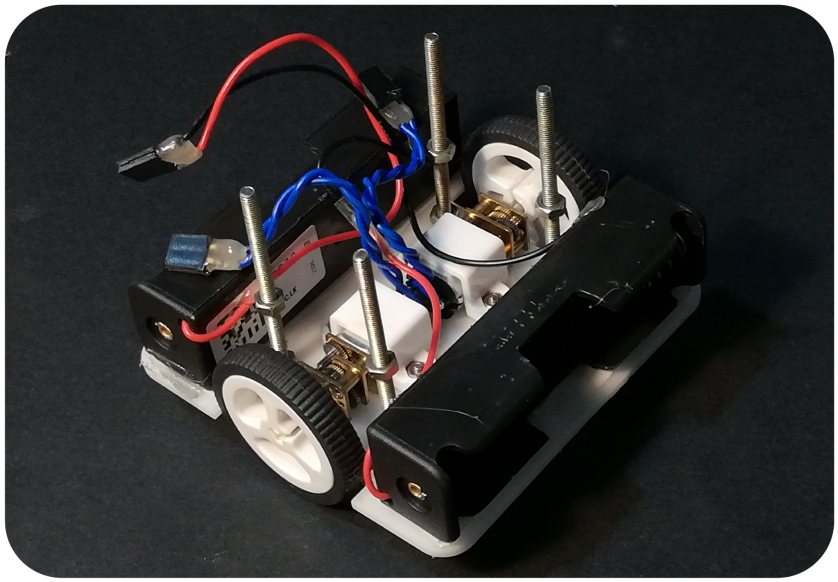
Add four 50mm M2 bolts to the base plate as shown in the figure and fasten them. Also add more extra nuts to each bolt to make the spacer which holds the pcb.

Assemble motors to the base plate

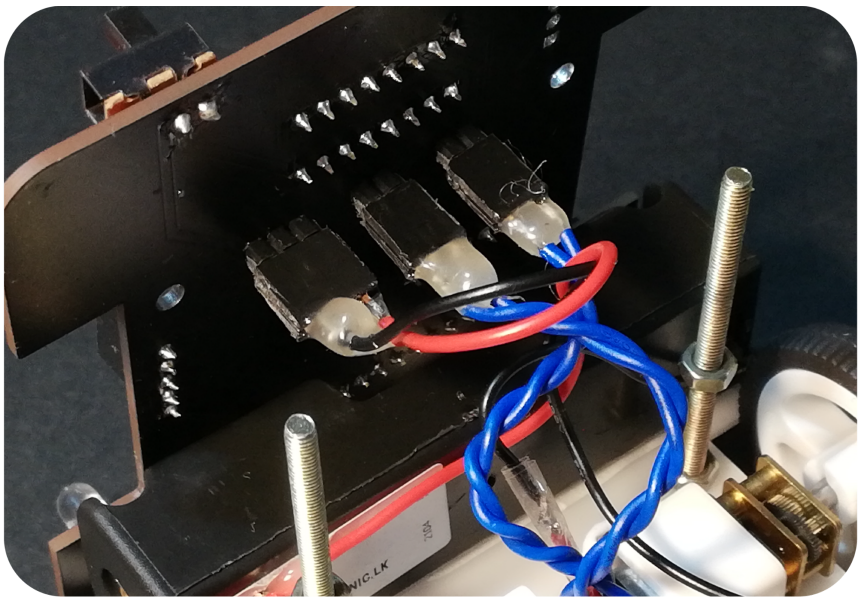


Assemble the motors to the base plate with the help of motor brackets.

Assemble battery cases

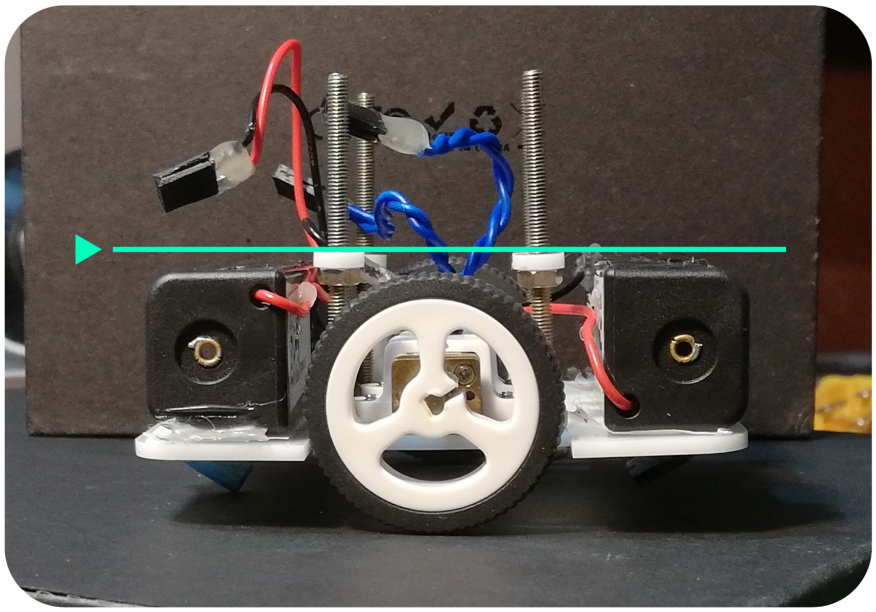


Connect the battery motors to the PCB.



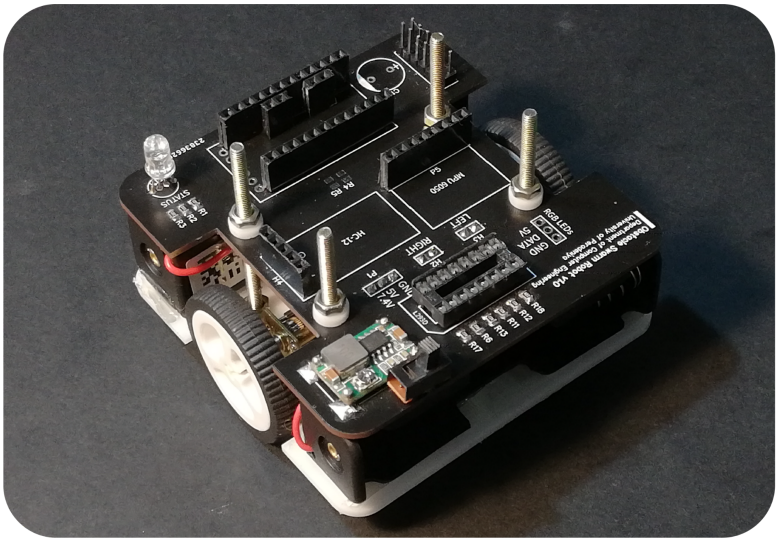
When connecting the battery connector pay attention to the VCC and GND pins. Markings of the pin configuration can be seen on the top of the PCB.

Adjust the spacer nuts to give the required space for the components.



To protect the surface of the PCB add the laser cutted washers to the bolt as seen in the figure.

Assemble the top and the bottom parts.



Now you can insert the PCB to the four bolts and fasten the bolts. Place an acrylic washer to protect the PCB against the pressure of bolts.

Now the assembly is almost done, let's plug the modules to the board.

Assemble the modules.

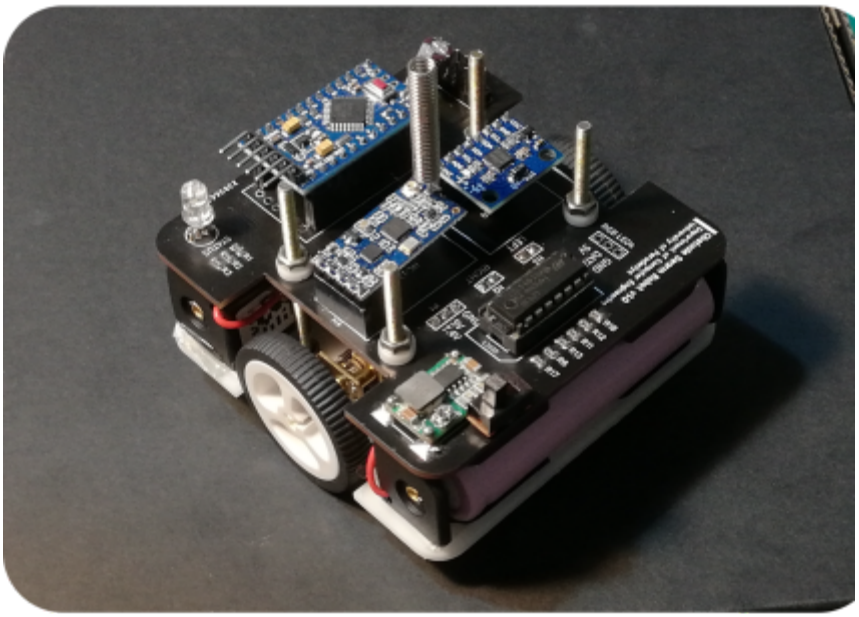


Figure 6

Assembly is complete !!

3.7. Updating the firmware

You have to upload the firmware using PlatformIo in VSCode or you can use Arduino software to upload. Code can be taken from the following link
https://github.com/cepdnaclk/e16-3yp-obstacle-bots-for-swarm-robots/blob/master/Hardware/BotCode/src/main.cpp

3.8. Radio Transceiver Module for the Platform PC

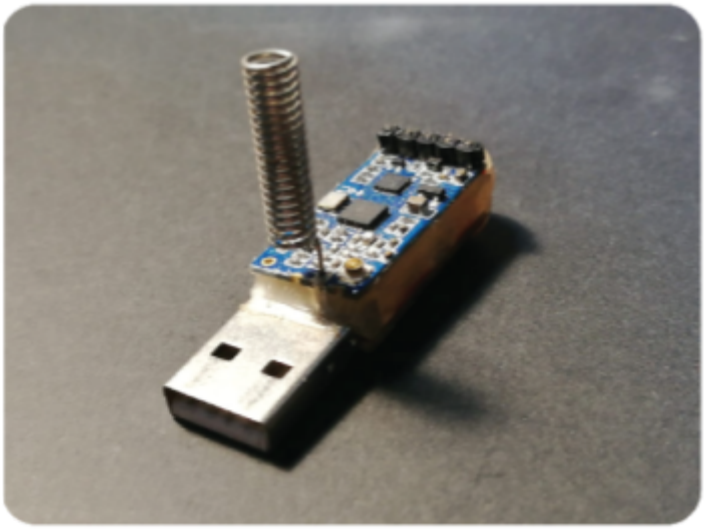


Figure 7

3.9. Generating the AR markers

Use this script to generate AR Markers
https://github.com/cepdnaclk/e16-3yp-obstacle-bots-for-swarm-robots/blob/master/Platform_PC_Software/Genarate_Ar_Markers/genarate_marker.py

Figure 8

4. Deploying the online platform

4.1. Introduction

The system facilitates a 3D user interface to deploy the algorithms. This has been created using the Three.js javascript library.

4.2. Deployment steps

This guide shows the way to set up a web interface inside the ubuntu server

- Firstly, you have to clone the Github project to the server

```
git clone https://github.com/cepdnaclk/e16-3yp-obstacle-bots-for-swarm-robots
```

- Install virtual environment

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

- Create the virtual Environment

```
python3 -m venv env
```

- Activate the virtual environment

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

- Install Django

```
pip3 install Django
```

- Install Gunicorn (wisgi Interface)

```
pip3 install gunicorn
```

- Install Nginx

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

- Installing the supervisor to keep the django site up and running

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

- Configure supervisor

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

Put the following configurations in the file

```
# put these inside the file
[program:gunicorn]
directory=/home/ubuntu/e16-3yp-obstacle-bots-for-swarm-robots/WebApp/MainWebA
```

```
pp
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
```

Run the following command to make the necessary directories

```
sudo mkdir /var/log/gunicorn
sudo mkdir /home/ubuntu/Sock
```

Reading the configuration file in the supervisor

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

Configuring Nginx

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

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-obstacle-bots-for-swarm-robots/WebApp/MainWebApp/static/
;
        }
}
```

The IP address should be changed to the public IP address of the server

- Restarting the Nginx server

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

5. Deploying the MQTT Broker

5.1. Introduction

MQTT is a public subscribe system that can be used to have bidirectional communication. We have used two topics to communicate. One topic is restricted in such a way that only the platform PC can write messages to the topic. The other topic is a by-directional one and we have used AES OFB encryption.

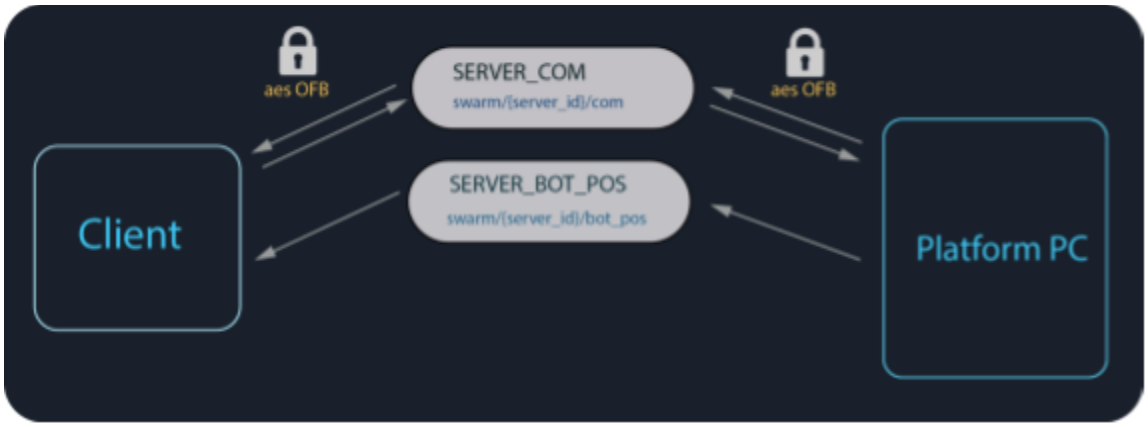


Figure 9

5.2. Deployment steps

Installing the mosquito brocker

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

6. Configuring the Operator GUI

6.1. Introduction

The operator GUI provides a responsive user interface to operate and monitor the real hardware robots.

6.2. Configuring Steps

First, you have to install and set up **python3** and **pip3** in your system.

Then you have to clone this repository.

https://github.com/cepdnaclk/e16-3yp-obstacle-bots-for-swarm-robots.git

Then change the directory to, **/Platform_PC_Software/MainCode/**

```
cd /Platform_PC_Software/MainCode/
```

Next, open a terminal in that directory and enter the following command

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

To start the program run the following command

```
python3  mainProg.py
```

6.3. Algorithm

Figure 10

For developing the algorithm, the robots are considered as the charged particles, and their destinations have opposite charges. Therefore, the robots themself and the border of the arena as a potential barrier, create a potential field.

So, it uses particle repulsion theory so that they are never gonna collide while they are moving.

Also, the system uses a modified version of the Stochastic Gradient Descent algorithm so that robots are moving in the highest gradient direction in order to reduce their potential. Also, the robots are assigned to a particular task considering the total distances which eventually optimize the efficiency of the final outcome.

6.4. Program sequence

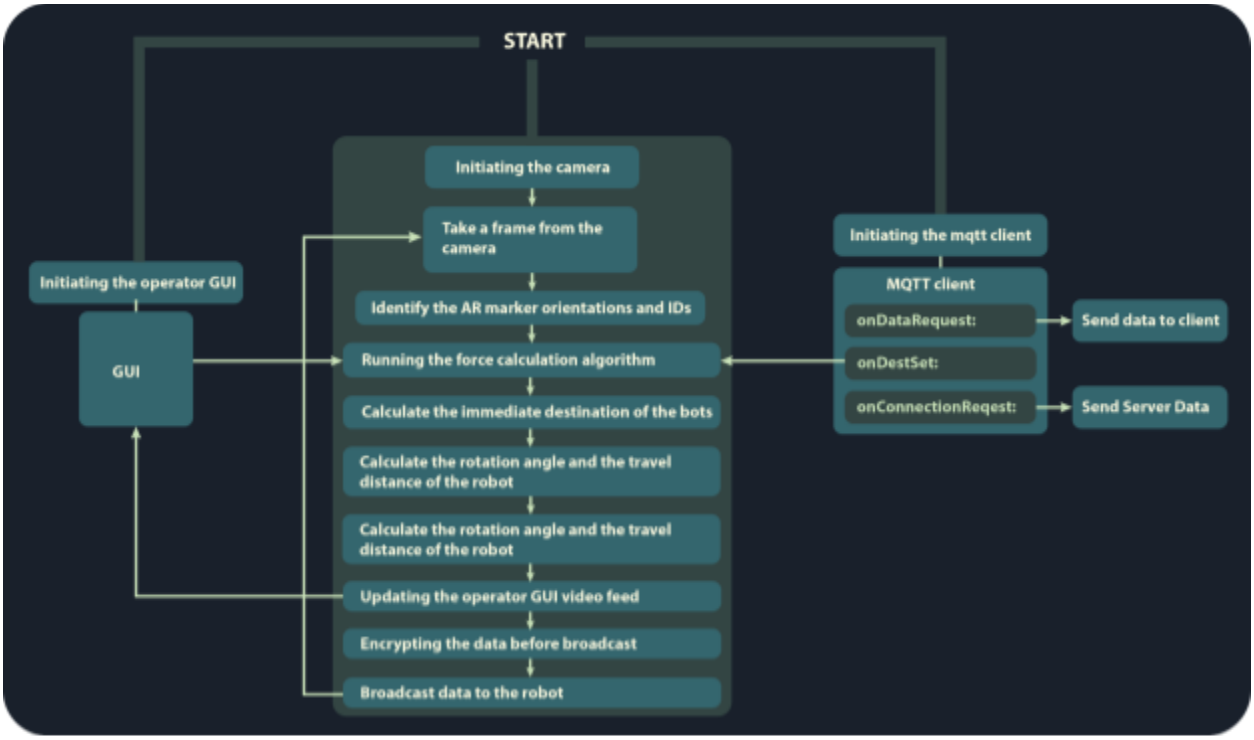


Figure 11

The program consists of three main processes

- GUI process
    - The GUI is made with HTML, CSS and hosted with Flask local server
- MQTT Process
    - Mqtt process is handling the communication between the web interface and the platform pc.
- Main Process
    - The AR markers are identified using the CV2 AR marker library.
    - Then the above-mentioned particle repulsion algorithm runs and calculates the immediate destinations of the robots.
    - Then the distance and angle calculating algorithm kicks in and calculates the robot turning angles and the distances.
    - Then the calculated angles and the distances are broadcasted over radio transmission.

Figure 12

7. Device Installation

7.1. Power Requirements

- You have to install two 18650 Li-ion batteries on a robot. You have to charge each and every battery for at least 3 hours.
- A 24v power supply should be connected to the status lamp.

7.2. Network Configuration

- Platform PCs should always be connected to the internet. (No other specific configuration needed)

7.3. Configuring Camera Setup

- The camera should be fixed, about 2m from the top surface of the arena.
- Then adjust the camera view by looking at the camera image in the Operator GUI (the whole arena should be visible through the camera view)
- Then you have to place a robot inside the camera view (Without turning on the robot)
- Make sure the robot is identified by the Operator GUI by drawing a green square around the AR marker.

(Note:- If the markers are not detecting. Please refer to the camera troubleshooting section)

7.4. Radio Transmitter Installation

- The radio transmitter should be plugged into a USB port
- You should find the COM port for the transmitter
- You will be prompted to ask the COM port of the radio transmitter when opening the Operator GUI.

7.5. Emergency Stop and Status Lamp Installation

- The status lamp should be fixed on the arena
- Emergency stop should be fixed near to the operator
- Both of them should be connected to the radio transmitter module.
- Given 24V power supply should be connected to the status lamp.

8. Security

Communication between the platform PC and the 3D user interface has been secured using multiple techniques which are provided by paho-MQTT library.

- **Authentication**

  Each device has a device ID and access token. By combining the device ID with the user name each device would have a unique ID (Different users could have devices with the same device ID, therefore, user name and device ID are combined). The combined ID acts as the MQTT user name and the access token acts as the password. These two are used to authenticate the device. The MQTT username and password are stored in the config.ini file and are read using the read function in the config parser library. Then the user name and password can be set using the username_pw_set function in the MQTT library.

- **Access Control**

  Each device has a particular topic to which it could publish. The device doesn't have permission to publish to any other topics unless the device ID and the access token are changed. The topic to which the device can publish is; user_name/device_IDand the device can publish to any subtopic under it.

- **Encryption**

  The communication uses TLS encryption to communicate with the Broker. Since the authentication sends the password this is a must. It also ensures confidential transmission of data. No application-level encryption is used since the MQTT model uses multiple subscriber publisher models and this would make key exchange much more complicated. Multiple subscribers imply that all the clients would need to know the key which makes it impractical.TLS certificate was obtained from the broker and can be set usingtls_setfunction in the MQTT library.

  Also, some channels are protected with an additional layer of security, using AES OFB. AES is based on a design principle known as a substitution–permutation network and is efficient in both software and hardware.

9. Troubleshooting

9.1. Camera position troubleshooting

Problem	Solution
Camera view detecting only part of the arena	• *Move the camera upwards until you can see the whole arena* • *Center the camera module in the Arena*
Camera not detecting	• *Unplug the USB cable and plug it again*
Camera feed interruption (or lagging)	• *Check for the CPU usage of the system. If the CPU usage is above 80%, kill unwanted processes.*

9.2. AR marker detection troubleshooting

Problem	Solution
AR Markers not detecting	● *Adjust the lighting conditions* ● *Move the camera closer to the Arena*

9.3. Local communication troubleshooting

Problem	Solution
If the blue LED on the transmitter not blinking rapidly	● *Unplug the radio transmitter and plug it again* ● *Check whether the COM port is correct*
Robots red light not blinking	● *Check whether the Robot ids are correct in the Operator GUI* ● *Reset the Robot* ● *Reduce the distance between the radio transmitter and the Arena* ● *Check the radio transmitter module is firmly attached to the PCB*

9.4. Robot troubleshooting

Problem	Solution
Not turning ON	● *Check whether the batteries are charged* ● *Check the batteries are installed in the correct way*
One or both wheels not turning	● *Check the pugs connected to the motors are firmly attached to the PCB* ● *Check the wheels are firmly attached to the motor* ● *Check the battery level.*
Robot is oscillating	● *Run the robot calibration program*