ZEUS Energy and Transport Company

Design of Embedded and Intelligent Systems (DEIS)

Tollgate IV (Group 4)

Abstract:

The rise of smart cities has become a pivotal point in the development of urban landscapes following the swift progress of technology. The paradigm shift towards autonomous smart cities transforms our understanding of and experiences with urban living as we stand at the heart of innovation and urban development. As a part of the Design and Embedded Course initiated at Halmstad University, we aim to pioneer a physical working prototype of a smart city that seamlessly integrates autonomous infrastructure and intelligent vehicle robots. The project involves the creation of a sophisticated smart infrastructure and intelligent robotic vehicles developed with a set of embedded components. The emergence of autonomous smart cities and vehicles signifies a transformative shift in urban living, leveraging advanced technologies to enhance efficiency, sustainability, and overall quality of life. Benefits of this initiative include optimized traffic flow, reduced energy consumption, and improved public safety. Moreover, the autonomous infrastructure and robots are designed to communicate seamlessly with each other and with other autonomous entities, fostering a cohesive network within the smart city. This networked system enables real-time data interchange, allowing for intelligent decision-making and, eventually, contributing to the development of a more efficient and responsive urban environment.

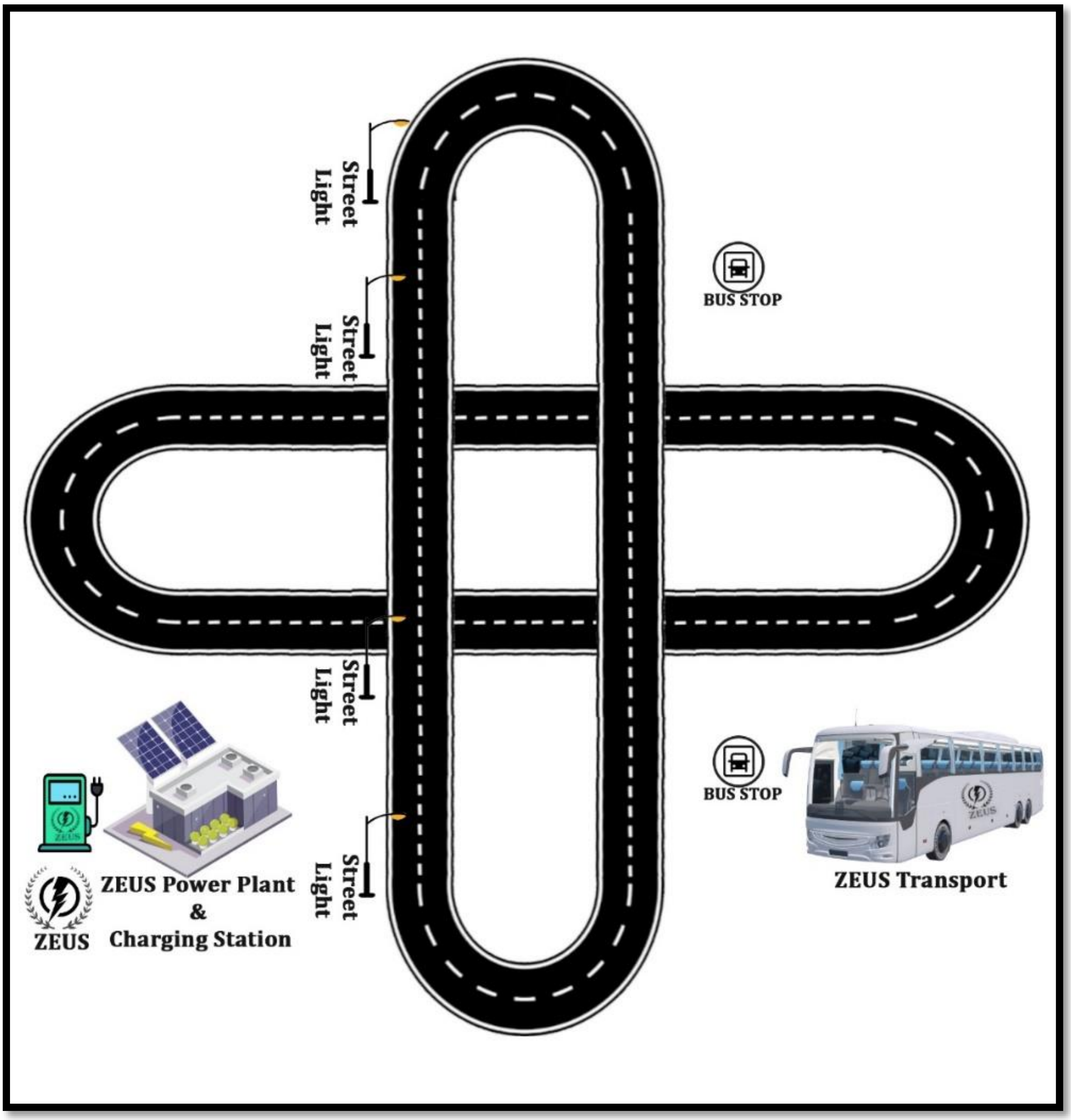
I. Introduction:

In an era marked by the relentless march of technological progress, the emergence of smart cities stands as a testament to our collective aspiration for a more efficient, sustainable, and interconnected urban future. The project aims to develop a smart infrastructure that functions as a holistic powerhouse, addressing power, energy, and transport needs in a sustainable and autonomous manner. We anticipate, the smart infrastructure is modelled as a sustainable power, and transport company that generates power and electricity through sustainable sources featuring solar panels. By harnessing this inexhaustible power, the infrastructure contributes to reducing carbon footprints and ensures a continuous and renewable energy supply. Simultaneously, the infrastructure also acts as a transport company that facilitates charging its electric vehicles, thereby weaving a seamless network of sustainable energy production and consumption. One of the prominent achievements is development of an autonomous electric bus, representing a stride towards sustainable public transportation. Functioning autonomously, it adheres to predetermined routes, halting at authorized bus stops, following traffic rules, and returning to the company for recharging, embodying a blend of innovation and eco-conscious urban mobility.

Further, as part of the infrastructural enhancement, we also integrated intelligent streetlights that adorn its roads. These smart streetlights, conceived and powered by the electric company, transcend conventional illumination by responding intelligently to their surroundings and detecting the presence of nearby robots or vehicles. By selectively illuminating in response to their proximity, they enhance road safety and conserve energy by illuminating only when needed. The project aims to integrate technology, where every component operates in harmony, which is the essence of an intelligent autonomous city. This transformative vision, fuelled by autonomous intelligence and powered by sustainable energy, brings in a new age in urban living—one in which efficiency, sustainability, and connectivity intersect to reinvent the very fabric of our communities.

Objective of this project is to implement a future city mock-up featuring small semi-autonomous robots and smart infrastructure.

1. Smart Public Transport (Robot)
2. Energy Company (Infrastructure)

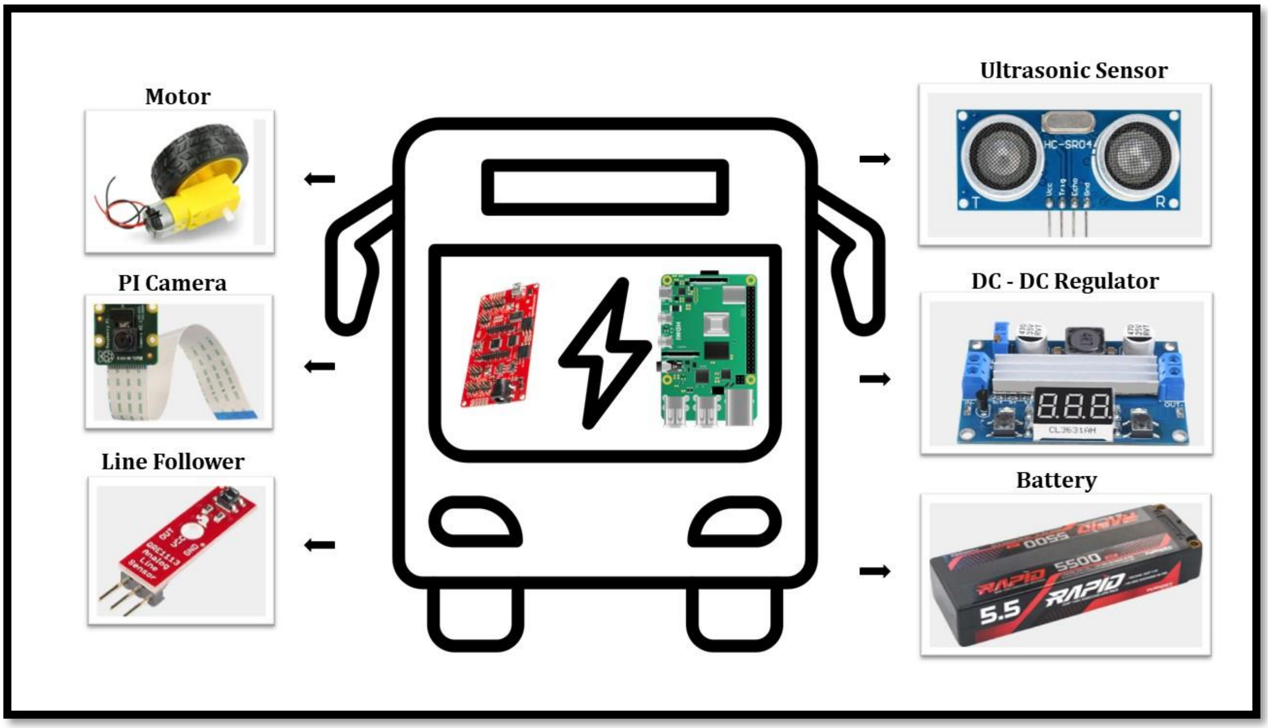


II. Hardware Overview:

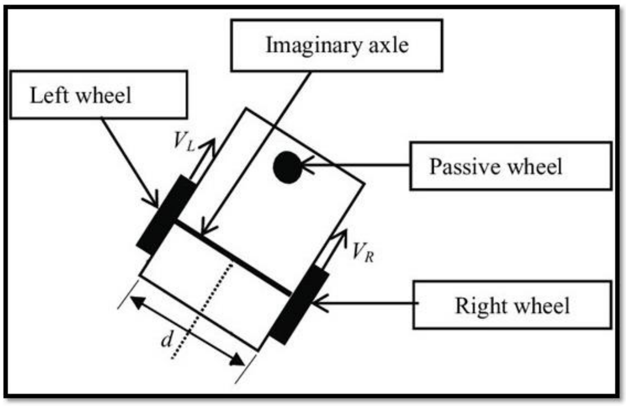
The project focuses to design and implement a prototype of an energy smart streetlights illuminate efficiently for vehicular motion, an automated robot tasked to perform as a public transport bus which is expected to follow traffic rules, stop at checkpoints (Bus stop) and a Power Infrastructure that functionalizes the streetlights.

1. Robot:

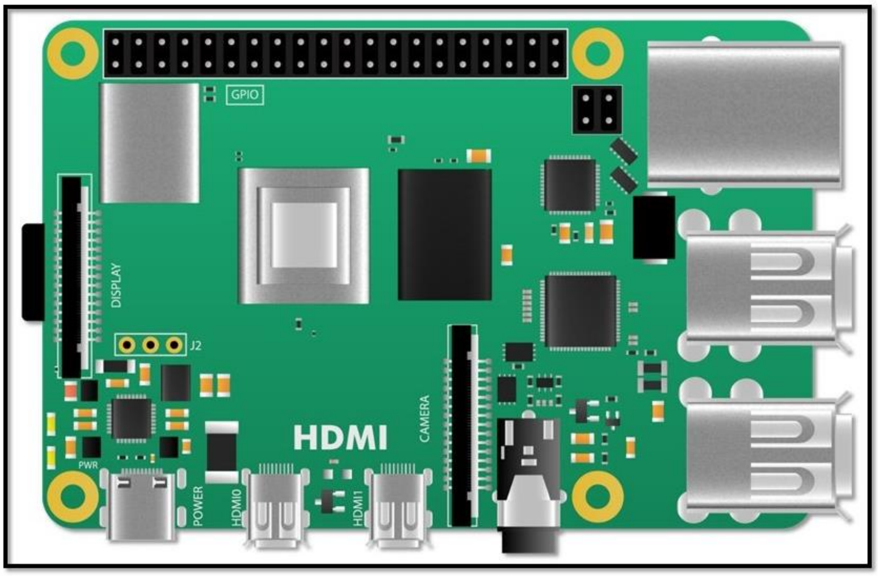
Zeus Transport, the project's robotic automobile system is a differential drive robot which modifies the speeds of its right and left wheels for its movements, by separately adjusting the wheels, this design technique enables the robot to perform a variety of actions, including forward and backward travel and rotational changes.



Notably, the robot can move in many directions for example when it turns right it stops its right wheel while moving its left. This design perfectly matches the project's goals because it is easy to operate and move, making it a favoured option for tasks like exploration, mapping, and basic navigation.



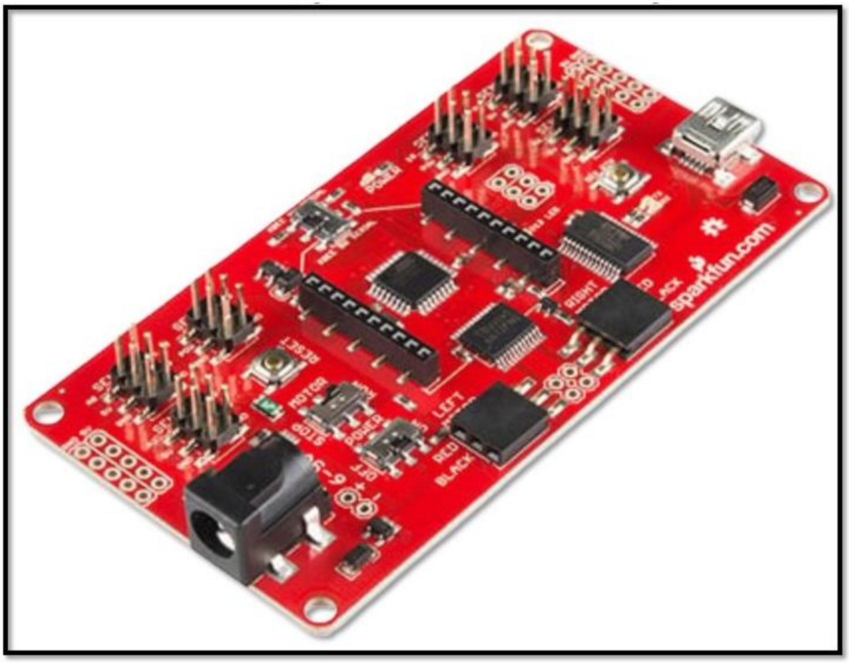
a. Raspberry Pi:



The Raspberry Pi 4 Model B with 8GB RAM, which is essential to the project's functioning, sits in its centre. With its quad-core ARM Cortex-A72 processor, this compact yet potent gadget offers improved speed and efficiency. This Raspberry Pi model supports dual band wireless networking, USB 3.0, and a large amount of RAM, allowing for smooth multitasking and high-definition display capabilities. Its adaptability and strong specifications are the foundation for carrying out the various tasks and computational operations involved in the project.

b. RedBot:

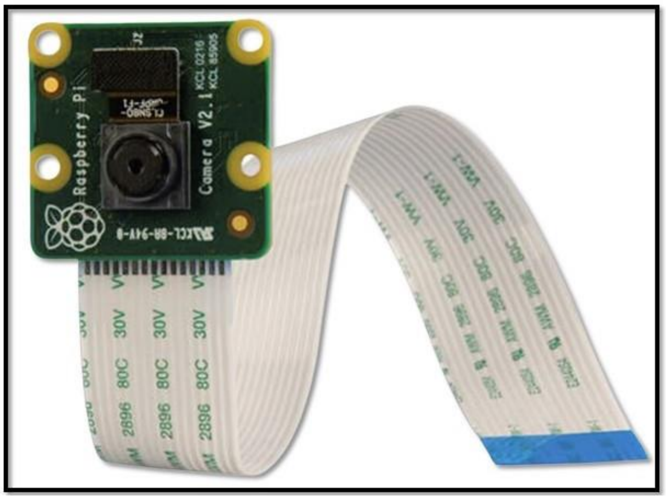
The Arduino Uno incorporated into the RedBot framework was another essential component that made this project possible. The RedBot's usefulness was improved with this Arduino microcontroller, which acted as a basic control unit and managed sensor data and motor controls.



Its function went beyond simple control; it allowed for complex interactions between the Raspberry Pi, sensors, and actuators, enhancing the project's capacity for complex robotic operations, data collection, and real-time processing.

c. PI – Camera:

Another important component was the Pi Camera, a critical sensory tool used in the study to aid in bus stop recognition.



The robot was able to see and understand its surroundings in real time because to the high-resolution camera's interface with the Raspberry Pi. This camera was essential to the project's vision-based tasks because of its remarkable features, which included adjustable focus, high-definition imaging capabilities, and smooth integration with the Pi's computational power. These features allowed the robot to precisely navigate and react to its surroundings.

d. DC Motor:

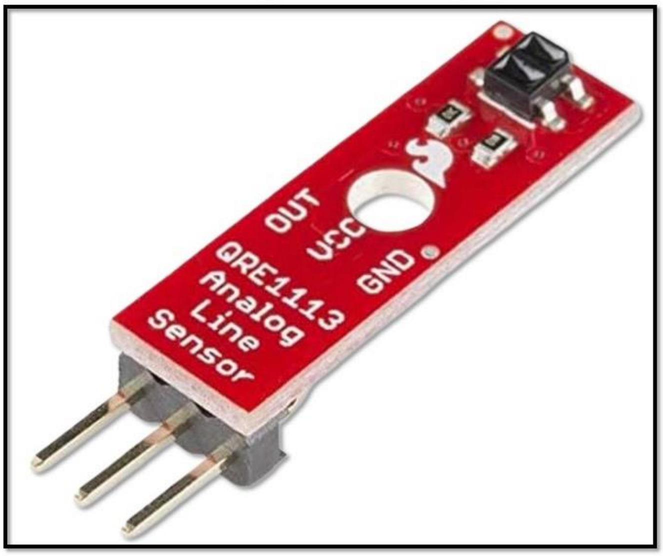
The project utilizes a DC motor equipped with a 1:42 gear ratio, selected for its ability to amplify torque while reducing rotational speed. This gearing arrangement allows the motor to rotate 42 times slower than its original speed, significantly boosting force output, which aligns perfectly with the project's specific requirements.

Instead of employing an independent motor control system, the motor is directly interfaced by connecting its pins to the designated left and right motor pins on the RedBot. This direct connection provides immediate and precise control over the motor's functionalities through the RedBot's interface, facilitating seamless modulation and manipulation of the motor's behaviour according to project needs.

The choice of integrating a motor with a 1:42 gear ratio indicates a deliberate consideration for the operational demands of the project. It implies a necessity for heightened force or torque essential for driving specific critical mechanisms crucial to the project's functionality. This motor configuration ensures the project's efficient and dependable operation by delivering the required strength and precision in motion.

e. Line Following Sensor (QRE1113):

The QRE1113 Line Sensor was used in pairs on the robot's left and right sides, and its main function was to trace a black line on the surface. The sensor has got three pins, and the pins encompass the sensor's power input, ground connection, and an analog output that relays the reflected light intensity information. Based on the principle of light reflection, these sensors detected the existence of a black road when their values were greater than 400 and a contrasting white surface when their readings were less than 100.



These tiny sensors are equipped with an infrared LED emitter and a phototransistor receiver, which allow for accurate line identification based on the intensity of reflected light. These sensors are compact, simple to integrate, and have a sensitivity that can be adjusted. They helped the robot stay on the marked black line and keep its path alignment.

- Right Sensor: A0 (Connected Pin on RedBot)
- Left Sensor: A6 (Connected Pin on RedBot)

f. Ultrasound Sensor (HC-SR04):

Combining the Spiral GPS system with the HC-SR04 Ultrasonic Sensor, which operates via a four-pin interface, is essential to our robot's obstacle avoidance mechanism. This sensor has trigger and echo pins for delivering and receiving ultrasonic signals, as well as power and ground pins. It is built into the robot and actively searches the immediate area for possible impediments to validate GPS data. The HC-SR04 sensor emits ultrasonic waves upon activation, signalling the presence of another robot nearby, as detected by the GPS.

The sensor determines how close the obstacle is by timing how long it takes for these waves to return. The robot stops moving if it detects an obstacle in front of it; otherwise, it moves onward. During the robot's navigation, this sensor's smooth integration with the RedBot platform allowed for effective data processing and decision-making.

- Trigger: 9 (Connected Pin on RedBot)
- Echo: 11 (Connected Pin on RedBot)

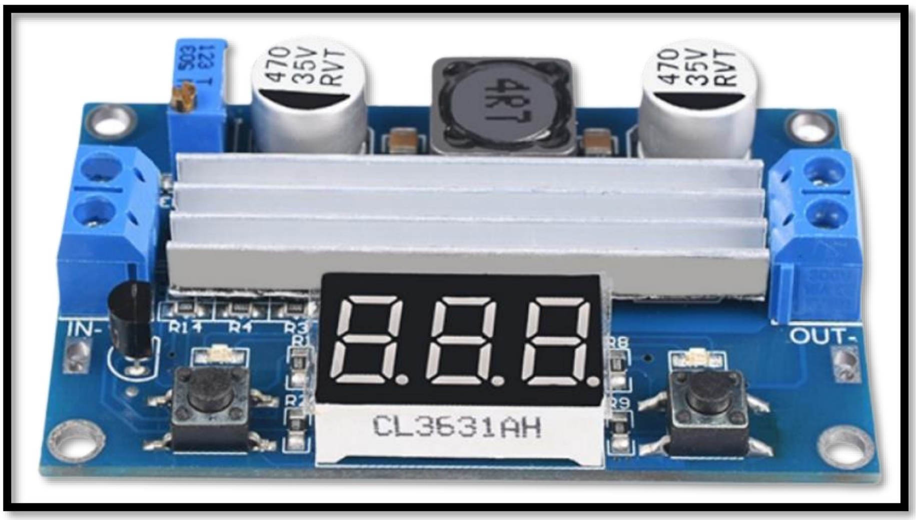
g. Battery:

A 5500mAh rechargeable battery powered the whole contraption, giving the Raspberry Pi 4 Model B with 8GB RAM, RedBot, HC-SR04 Ultrasonic Sensor, line sensors, Pi Camera, and related components the energy they needed to function.

The robot was able to accomplish its assigned tasks, such as navigation, obstacle avoidance, line following, and environmental perception using the camera and sensors, because of the high-capacity rechargeable battery that allowed for continuous operation for extended periods of time.

h. DC Voltage Regulator:

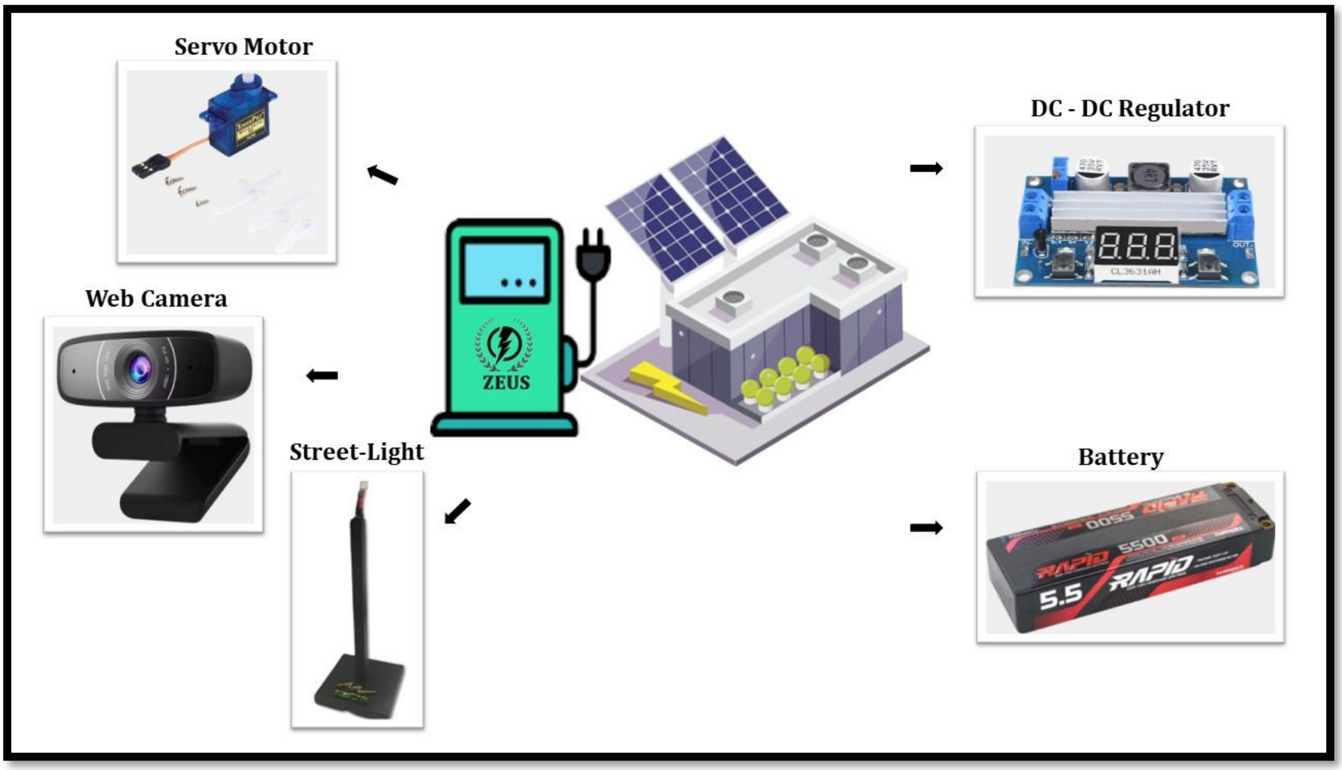
In the project setup, a DC regulator played a crucial role in managing the power flow. Specifically, it was employed to step down the voltage from 9V DC to a stable 5V. This regulation was necessary to ensure a consistent and suitable power supply to the Arduino, optimizing its performance and preventing potential damage due to voltage irregularities. By maintaining a reliable 5V output, the DC regulator enhanced the stability and functionality of the Arduino, contributing to the seamless operation of the overall system.



2. Requirements Smart Infrastructure:

The requirement for our infrastructure includes switching on / off the intelligent streetlights of the city with respect to the moving robots preserving energy. Another condition is monitoring the power consumption of all other infrastructures built by our fellow groupmates and sending them alerts based on their power consumption.

Our designed infrastructure, operating under the name ZEUS Energy and Transport Company, focuses primarily on energy production and distribution. The key elements include a central building housing a Raspberry Pi, strategically placed streetlights, and bus stops. The Raspberry Pi serves as the command centre, overseeing and orchestrating all operations within the infrastructure.



a. Raspberry Pi 4

Central Hub: The Raspberry Pi is the central intelligence, programmed to manage the gate, streetlights, and communication with the robot via ROS channels. The model of this one is also same the one used on the robot.

The Raspberry Pi orchestrates the seamless functioning of the infrastructure through these two ROS channels:

- **GPS:** Coordinates with the GPS server, obtaining precise position coordinates of the robot. So that the streetlights corresponding to those robots can be switched on /off
- **Py sub:** For the internal communication with our robot

b. Webcam

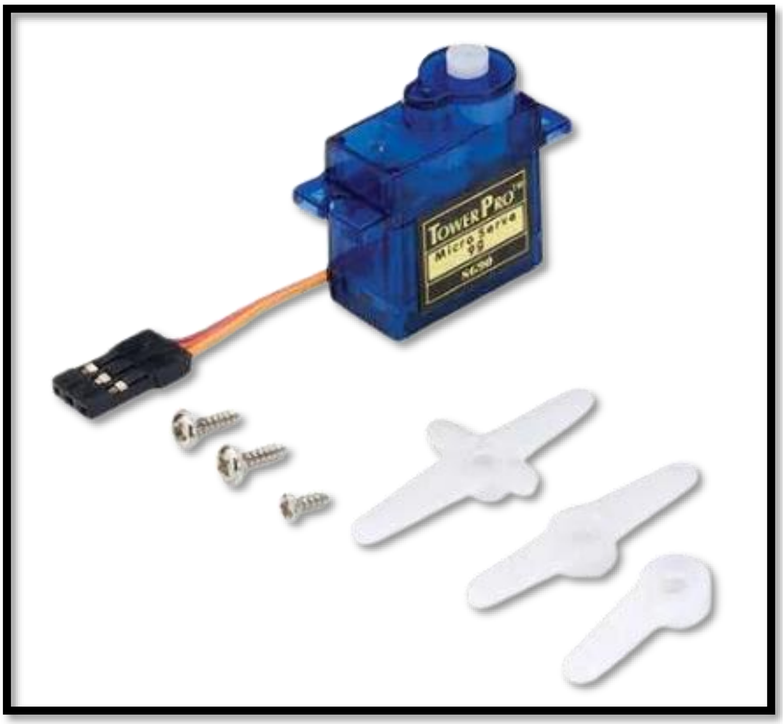
Arrival Detection: Connected to the Raspberry Pi, the webcam is positioned to capture the arrival of the robot for recharging. Image analysis is employed to identify and authorize our specific robot.

The webcam, connected on the Raspberry Pi through the USB port, plays a critical role in:

- Capturing images of the arriving robot for recharging.
- Utilizing image analysis to differentiate our own robot from others.
- Enabling exclusive access to the charging station by triggering the gate mechanism.

c. Servomotor

Gate Mechanism: The servomotor operates the gate at the charging station. Triggered by the webcam, it opens for our designated robot, closing once the robot departs to prevent unauthorized access.



The servomotor, integrated with the gate mechanism, functions by:

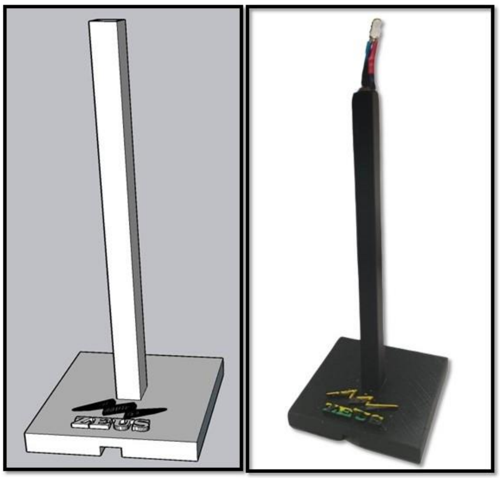
- Opening the gate upon detection of our robot by the webcam.
- Automatically closing the gate as soon as the robot exits the charging station.

- Ensuring the gate remains closed for any unidentified robots, enhancing security within the smart city environment.

d. Streetlights

Strategic Illumination: Four streetlights are strategically placed along the roads. Activated through ROS channel "GPS" based on Euclidean distance, they illuminate when our robot or any collaborating robot approaches, ensuring well-lit paths.

Design **Actual**



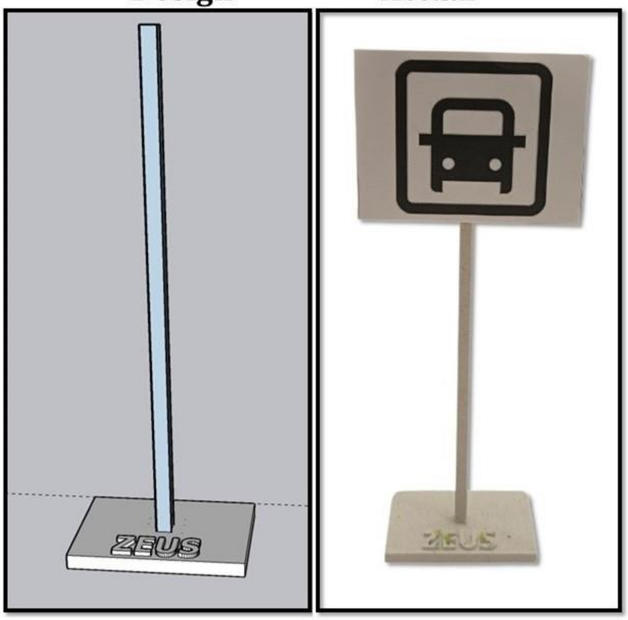
The streetlights, strategically positioned along roads, are activated, and controlled by the Raspberry PI. The activation threshold for lighting up the streetlights is set at a Euclidean distance of 100 units.

- Illuminating when our robot or any collaborating robot approaches.
- Automatically turning off when the robots move away, conserving energy and promoting efficient resource utilization.

e. Bus Stop:

A 3D modelled bus stops strategically placed within the arena serves as a designated stopping point for our robot. Activated through proximity sensors or a designated ROS channel, the robot halts its movement and remains stationary for 5 seconds upon reaching this point.

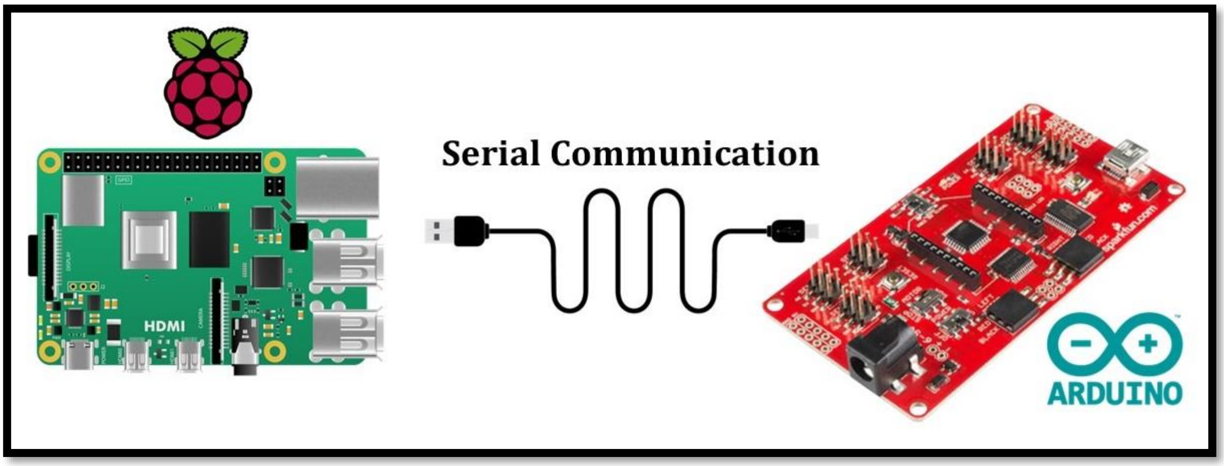
Design **Actual**



The bus stop, though a passive component, plays a crucial role in the robot's navigation strategy. Controlled by the central system, it acts as a point of brief pause for our robot, allowing for necessary actions or observations to take place.

III. Internal Communications:

The communication through a USB cable establishes critical serial communication between the RedBot and Raspberry Pi, playing a pivotal role in our system's functionality. Set at a baud rate of 115200, this configuration ensures efficient and high-speed data exchange between the two devices. Adjusting the baud rate significantly impacts communication quality; at lower rates, such as when experimented below the optimal 115200, noticeable setbacks emerged. Communication became erratic, marked by delays in the transmitting and executing commands. This emphasized the importance of the chosen baud rate in maintaining a reliable and real-time flow of data, directly impacting the system's responsiveness and performance.



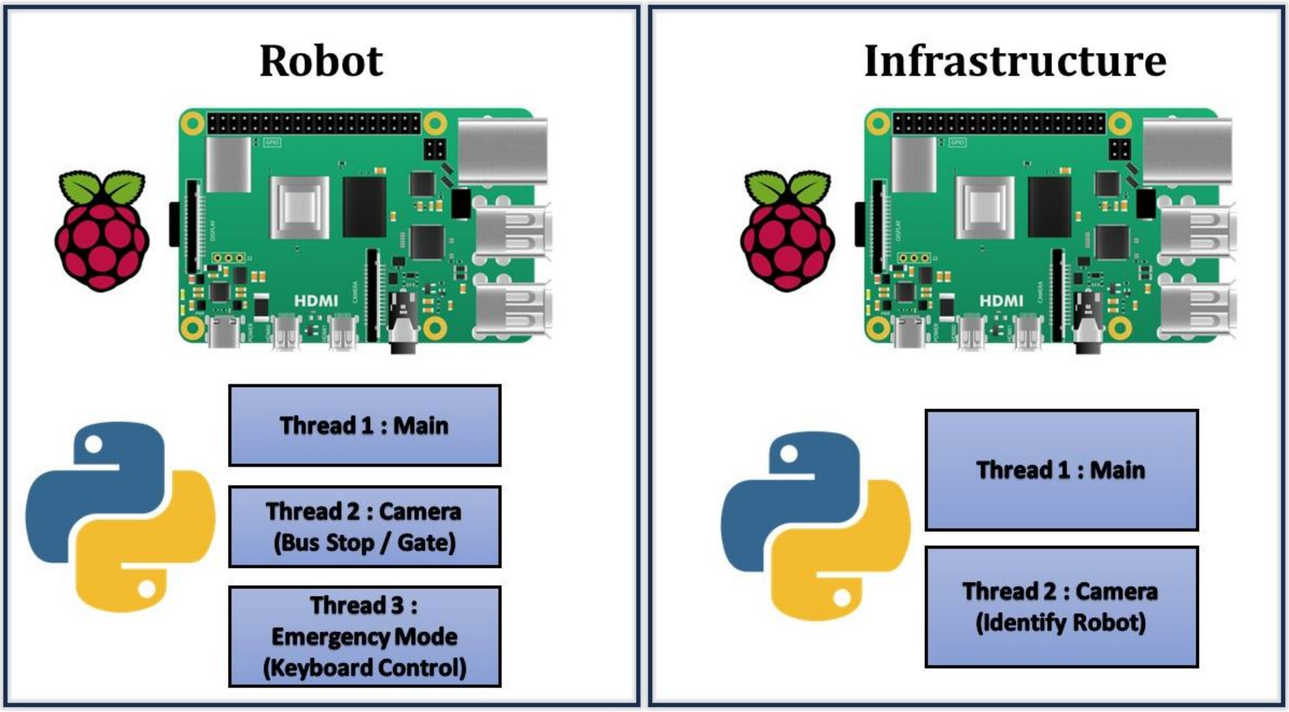
Unique Serial Communication Commands:

- D : Drive Forward
- Q : Drive Forward (Slower Speed)
- Z : Specific Time-Based Reverse after charging
- L : Left Turn
- R : Right Turn
- T : Proximity Sensor Activation (Ultrasound)
- S : Stop
- W : Emergency Forward (From the Keyboard)
- A : Emergency Left (From the Keyboard)
- F : Emergency Right (From the Keyboard)
- X : Emergency Reverse (From the Keyboard)
- P : Serial Flushing

IV. Parallel Processing (Threading in Python):

Python programming is employed for threading as a fundamental strategy in our project, enabling the execution of multiple tasks simultaneously. By leveraging threads, the system efficiently handled concurrent operations, enhancing its responsiveness and overall performance.

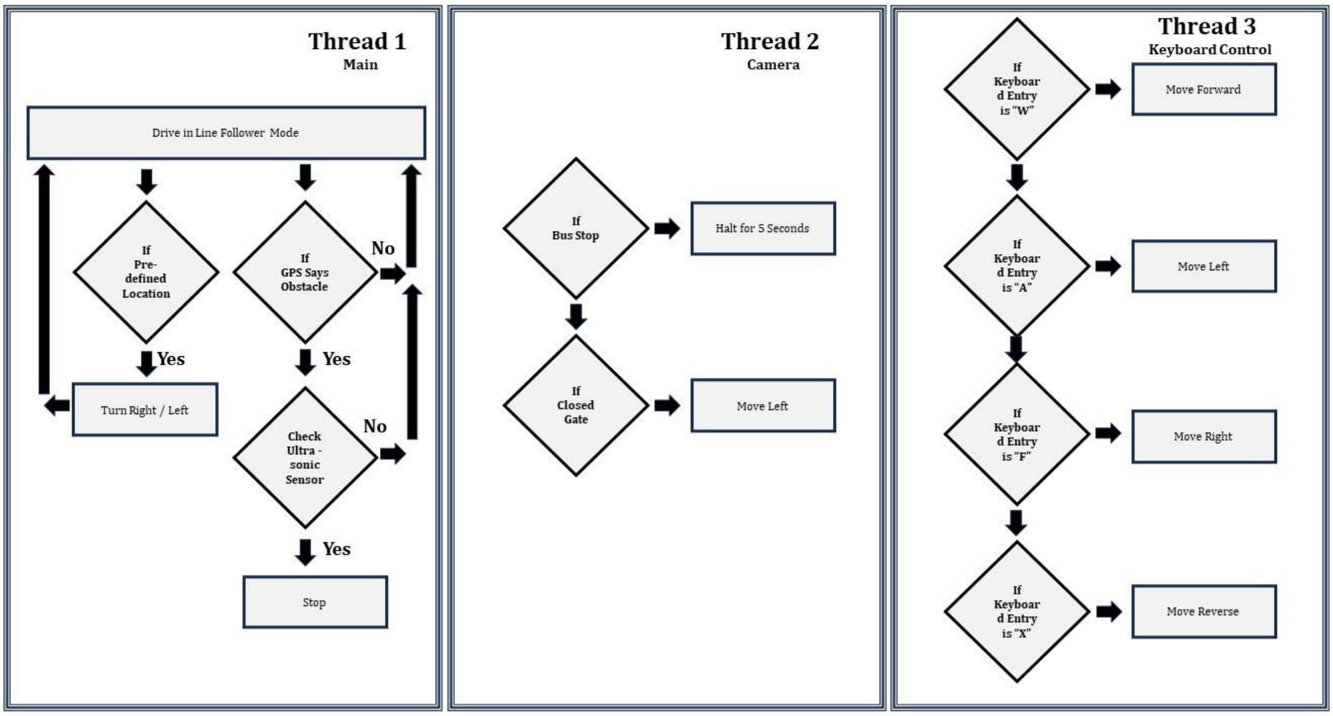
The use of threads allowed distinct tasks to run concurrently, preventing bottlenecks and ensuring smoother execution. This approach proved instrumental in managing parallel tasks, such as sensor monitoring, data processing, and motor control, optimizing system efficiency.



Python's threading capabilities facilitated a cohesive framework where different functionalities could operate independently, yet harmoniously, resulting in a more responsive and agile system overall. This threading methodology significantly contributed to the project's success by enabling seamless multitasking and enhancing the overall responsiveness of our application.

V. Strategy

The Mobile Robot cruises the track continuously and it has multiple tasks and checkpoints. Bus stops and Charging station serve as checkpoints and task is to board and charge respectively. Discreet algorithms were devised to combat robot interactions with checkpoints.



The overall strategy of the mobile robot can be broken down into following parts.

1. Navigation.
2. Detecting Bus Stops.
3. Traffic at cross lanes.
4. Smart Street lights.
5. Charging Station.

a. Navigation:

Our project's foundation was spiral coding, which allowed for far more functionality than just point labelling. It was essential for accurate labelling, but it also developed into a crucial component of our navigation, object detection, and collision avoidance systems.

Spirals were printed out and carefully positioned atop our robot as recognizable markers. The spirals were originally intended for precise point labelling. These spirals had one special benefit: our system could accurately identify them even when only a portion of them was visible. Our GPS-based navigation successfully made use of these physical spirals' distinctive patterns for pathfinding and location triangulation within the surroundings.

Furthermore, the spiral coding methodology demonstrated its effectiveness in the field of object detection. Its precision and distinctness in identifying multiple physical points contributed to the system's ability to distinguish between different objects irrespective of their movement or proximity.

Spiral coding's dependability in collision avoidance was a major factor in the project's success. Because of its ability to reduce false acceptance errors in both object identification and location determination, it established strong collision avoidance mechanisms that guarantee the effectiveness and safety of our system.

Moreover, spiral-coded labels' resistance to variations in distance, viewing angle, and lighting significantly improved the system's overall performance. This flexibility supported our robot's smooth movement and operation in a variety of real-world scenarios by ensuring consistent and accurate navigation regardless of environmental variations.

The navigation of the mobile robot uses line follower sensing, where it drives forward/reverse upon sensing dark lines and stops upon sensing any light lines. The robot also has four cross-over junctions where it can make a lane switch with a line follower sensor in conjunction with GPS. It also has an obstacle avoidance, where it avoids collisions by sensing other nearby robots or objects using a combination of both ultrasound sensor as well as GPS.

b. Detecting Bus Stops:

Two bus stop sign markers are randomly placed in the circuit. The mobile robot upon detecting the sign stops for 5 secs and then proceeds to next task.

c. Traffic at Cross Lanes:

Traffic lights are emulated at the crossroads. Robot checks for the emulated lights before proceeding at the junctions. It stops for red light (encoded to a digit) and proceeds at green (encoded to digit).

d. Smart Street Lights:

Four smart streetlights are kept at a predefined locations in the track and they are controlled by the infrastructure, and it is turned on by the proximity (threshold)of all the whitelisted robots and turned off by the same threshold.

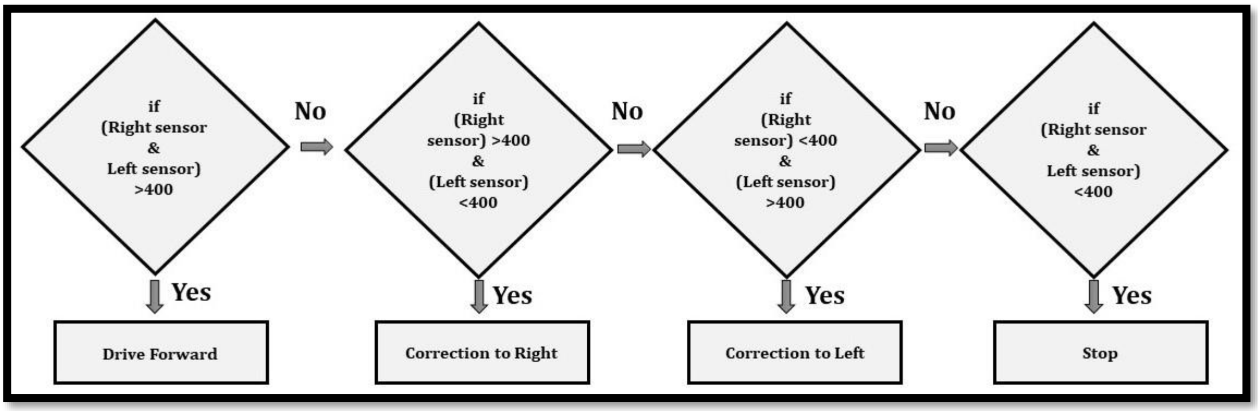
e. Charging Station:

Charging station is the infrastructure and it is controlled by a front gate. Robot enters the station to recharge after finishing a lap in the track and stays inside the infrastructure for 5 secs and then makes its way out of the infrastructure and proceeds.

VI. Algorithms and Pseudocodes

a. Line Following:

The mobile robot employs two infrared sensors, one on each side, which detect dark and light lines and provide corresponding analog values. To discriminate between the bright and dark lines, a threshold is specified. From the tests and observations light lines value is around 100 and dark lines have values above 400.



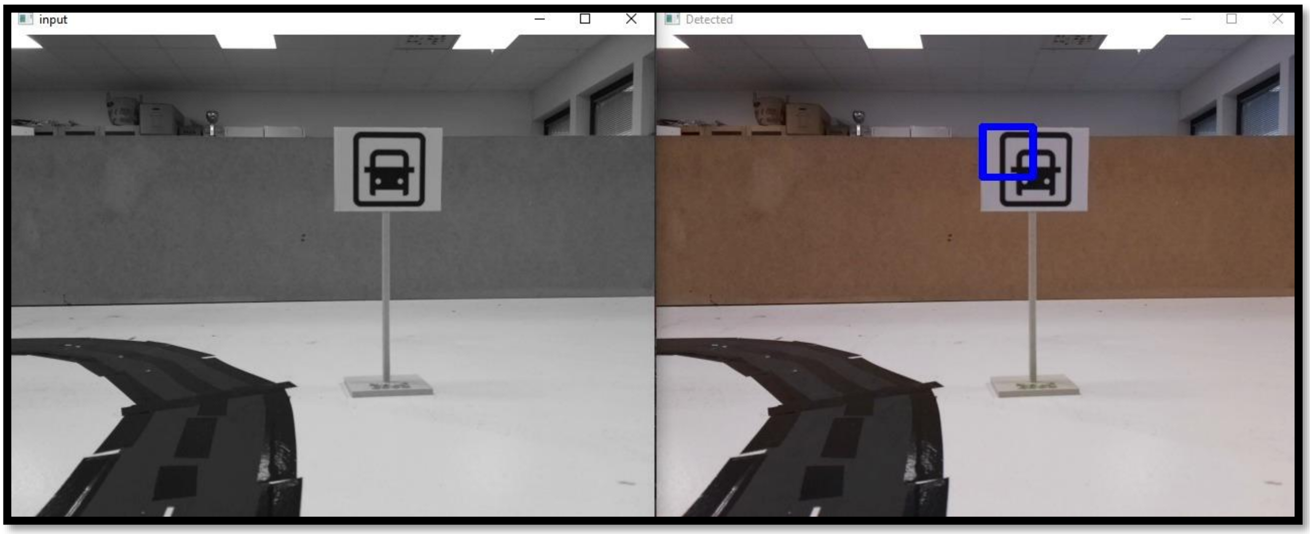
When the robot drives forward in line follower mode, it can drift away and stop by detecting light color. To steer the robot continuously, a drift correction was introduced, which corrects for the skew by rotating in the other direction. Skew is the difference between the left and right sensor values. Positive skew values indicate left skew and right skew if it's negative.

b. Cross Road Turns:

In scenarios where the robot must change lanes, it combines information from both the IR sensors. The robot stops at pre-defined junction coordinates and turns clockwise or counterclockwise until the sensor makes the dark-light-dark transition, which would effectively change the robot head to the desired lane and drive forward in line following mode.

c. Bus Stop:

Bus stop board sign is detected by the robot's on-board camera. The onboard camera continuously acquires frames, and a template of the bus stop sign was already created. When the frame grabbed by the camera matches the template (quantified by threshold), robot identifies it as bus stop and stop the motion.

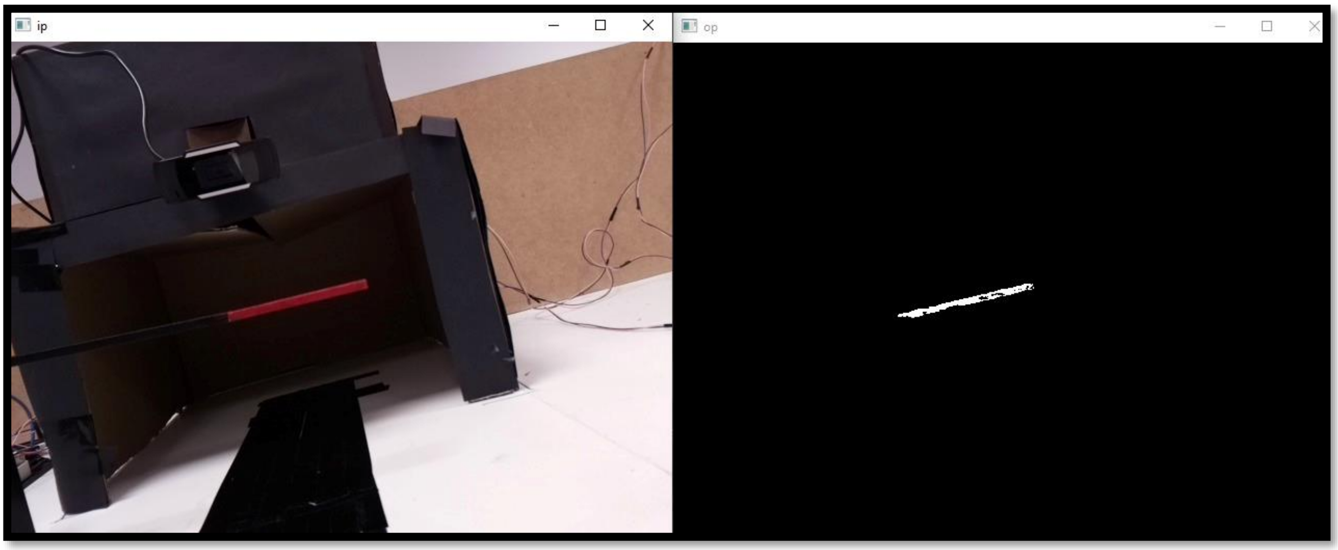


d. Charging Station:

A predefined point is sent to the robot as a home coordinate, robot will automatically make a right turn facing the infrastructure upon reaching this coordinate. The infrastructure has a security camera which controls the entry gate, its entry is restricted only to our mobile robot. The onboard camera in the mobile robot will stop the robot if the gate is closed and wait for the security camera to scan and open the gate. These two cameras serve as an interlock mechanism to avoid unnecessary crashes. This approach was better than using any timed events.

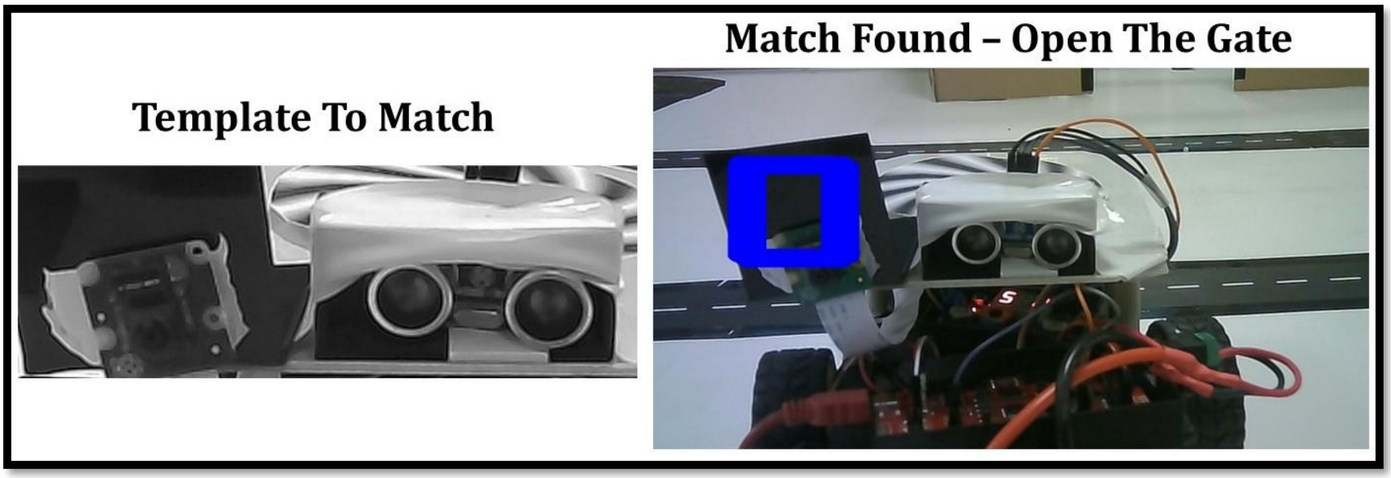
Gate detection:

The gate has a distinct red coating; when the robot activates a flag after performing a right turn, it continually applies a red threshold mask and counts the number of white pixels; if the gate is closed, it will provide a high white pixel count value and a low value conversely.



Robot detection:

The security camera also uses template matching, a unique template of the mobile robot was already created and matched with the frames acquired by the camera. When the template matching provides a score greater than the threshold, it verifies the arrival of robot and signals the infrastructure to open the gate.



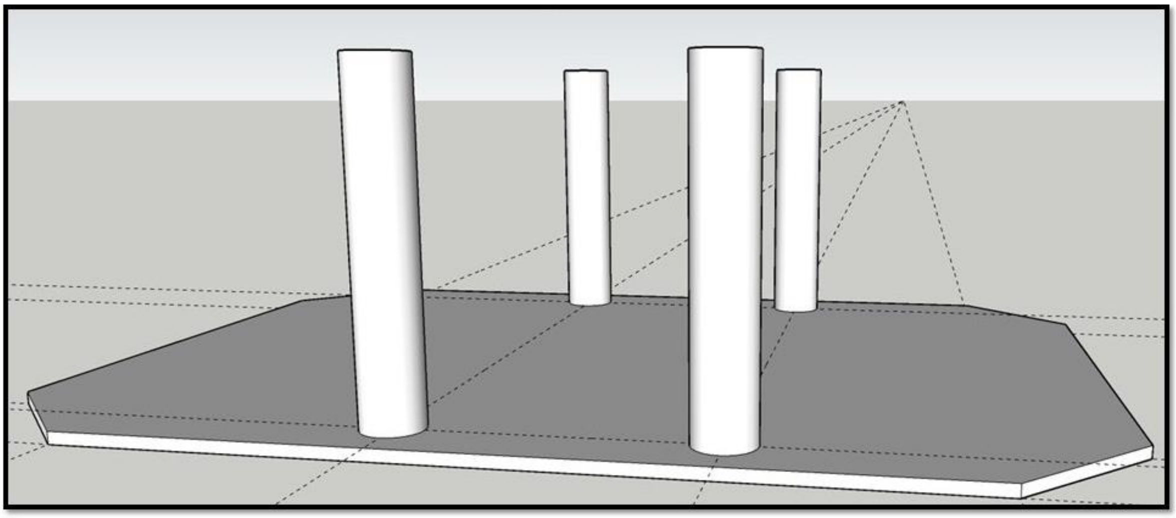
e. Manual Override:

The mobile robot also has manual override options to aid its navigation. There are four unique commands W, A, F, and X written serially to forward, left turn, right turn and reverse the robot respectively. This manual override was to give the robot an external impetus if it gets stuck in the road bumps owing to air bubbles.

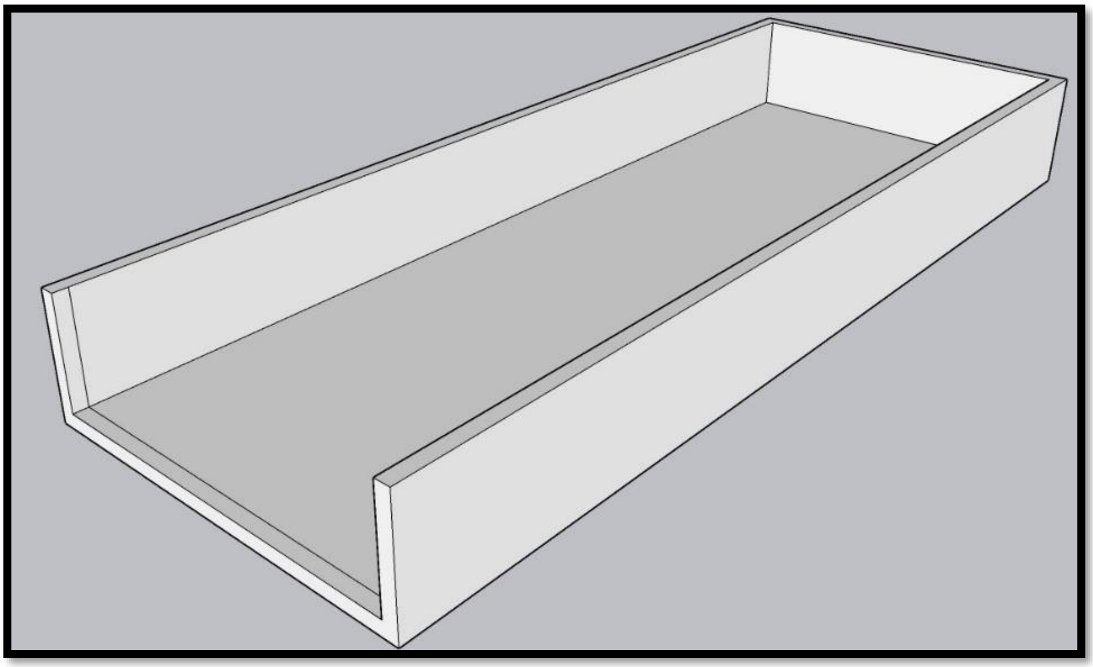
VII. 3D Modelling & Printing:

This project makes considerable use of SketchUp, a flexible and approachable 3D modelling program, to create complex designs. Users may create intricate models for different components with ease thanks to its user-friendly interface, which guarantees accuracy during the design stage. The team effectively developed and visualized the parts, components, and structures required for the robot and its accessories thanks to SketchUp's extensive feature set and versatile toolkit. The software's adaptability made it simple to modify and amend, enabling iterative improvements and precise specifications prior to the 3D printing stage, guaranteeing precise and tailored parts for the needs of the project.

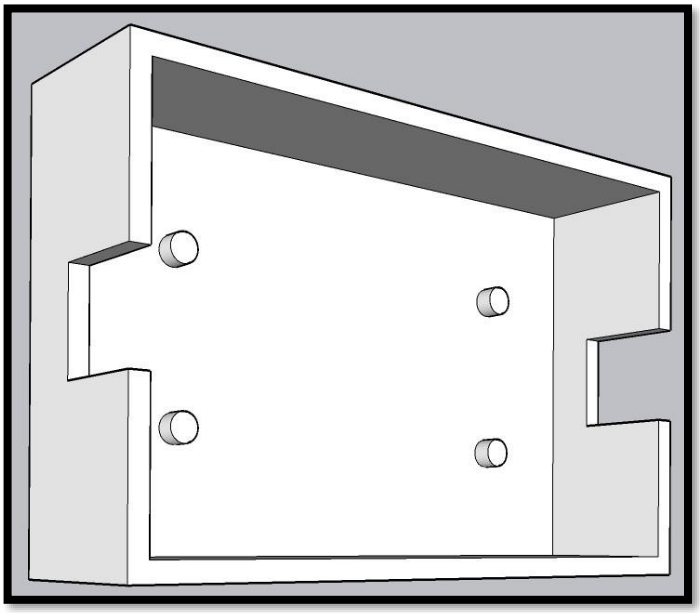
Robot Top (For attaching GPS Spiral):



Battery Holder



DC Regulator Case



VIII. Experience, Experiments and Short Comings

Over the course of this project, there were a lot of experimentations in approaches and component selections to achieve better performance and results.

1. One of the primary areas for improvement of the mobile robot is its high sequential. If this sequentially is lost owing to some factors, it drastically affects the robot's functionalities. To overcome this sequential dependency, a thought algorithm was devised to break the sequence in case of failure and return home for a restart. This idea was not implemented, can be considered as future scope.
2. Constructing line follower tracks posed some challenges. There were some shortages of the raw material and the replacements used were giving fluctuating results to different sensors. To overcome this, electric insulation tape was used to guide the sensors, which gave more consistent results. This process was highly taxing and tedious.

3. The GPS spiral had two problems: the first was false detection, which detected random clustered objects as spirals; because GPS is the primary sensor for navigation, collision avoidance, and other spatial-related tasks, this false detection affected the functionality of the robot; to avoid this situation, all robot id spirals were blacklisted, and only those available in the arena were whitelisted. The second issue was the delay in information translation in the ROS2 network. The delay was obvious and immediately detectable.
4. Changing the lanes at the crossroads was a bit of challenge, to make the algorithm consistently work required lot of fine tuning and GPS latency aspect enrolled into it. Initial approach was to use timed based event, which was unreliable and did not synchronize with the GPS.
5. A lot of time was spent automating ROS commands upon initialization of a Python script and its libraries. The approach involved reading a text file and executing commands in the bash terminal. However, all the efforts turned out to be futile as they encountered multiple snags.
6. Putty cannot register any keyboard, key press events. Manual override command is sent serially, but the actual requirement was to map the serial commands to arrow keys of the keyboard.
7. Reversing the mobile robot from the charging station was initially a timed process, but due to non-reliability in timed sequences, an ad hoc new algorithm was developed that employs the IR Sensor in the front to reverse the robot. The algorithm produced inconsistencies and there was insufficient time to fine-tune and improve it.

IX. Task division and participation

Student	Lab1-ROS	Lab2-PID	Lab3-GPS	Lab4-LCM	Project design	Project report
Dhanya	25	50	25	25	Streetlights and Infrastructure design	Infrastructure description
Kiran	25	50	25	25	Robot design and Navigation & Serial Communication. Strategy	Robot Hardware Description, Internal Communications
Mabin	25	50	25	25	Streetlights and Infrastructure Design	Introduction, Objective, and conclusion
Srinivasan	25	50	25	25	Robot design and Navigation & Image Processing. Strategy	Algorithms and Pseudocodes, Strategy, Short Comings

Conclusion:

In summary, this project has resulted in the realization of our goal of creating a prototype of a self-sufficient smart city. As a key component of sustainable urban development, the smart infrastructure powers the electric bus and illuminates smart streetlights, anticipating the harness of a sustainable energy resource. The development of the autonomous electric bus followed its route with ease using line followers and GPS spirals, obeyed traffic signals by communicating with traffic infrastructures, identified approved bus stops using camera technology, avoided collisions, and precisely returned to the infrastructure to be recharged again to continue the chain of the smart ecosystem. Utilizing the smart infrastructure to integrate smart streetlights allowed for energy conservation through selective illumination using Vehicle to infrastructure communication. This comprehensive approach reflects the successful fusion of technology and sustainability, paving the way for a future where autonomous intelligence and eco-conscious practices redefine the essence of urban living.

Reference:

1. Ubuntu Installation on Rasberry Pi (Link)
2. ROS2 2 Installation Steps (Link)
3. Putty Installation for remote accessing (Link)
4. ROS communication Python - Subscriber Publisher (Link)
5. Redbot Programming Basics (Link)

Code: (Link)
Videos & Photos: (Link)