RV Educational Institutions®
RV College of Engineering®
Autonomous Institution Affiliated to Visvesvaraya Technological University, Belagavi
Approved by AICTE, New Delhi

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

AUTONOMOUS DELIVERY SYSTEMS

Submitted by

Pranav Darshan

1RV22CS143

Priyansh Rajiv Dhotar

1RV22CS153

Raghuveer Rajesh

1RV22CS154

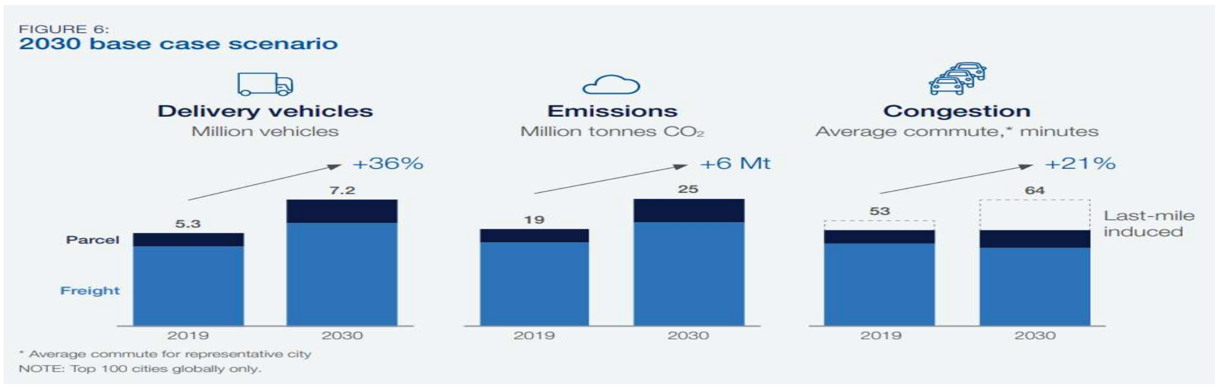
Pavan Shivakumar

1RV22CS135

Go, Change the World

Introduction

- Welcome to the future of delivery systems! In this era of technological advancement, we present to you an innovative solution that will revolutionize the way goods are delivered - the Automated Delivery System powered by OpenCV and blynk IOT.
- The world is witnessing an exponential rise in online shopping and e-commerce transactions. With this surge in demand, the traditional delivery methods struggle to keep up with the pace, leading to delays, inefficiencies, and added costs.
- With OpenCV's robust and flexible image processing algorithms, our delivery system can accurately identify and authenticate recipients, ensuring that packages reach the right hands, eliminating the risk of misplaced or lost deliveries
- The robotic arm, a marvel of engineering and automation, acts as the agile and dexterous extension of the delivery system. Its capability to manipulate objects with accuracy and finesse unlocks a whole new level of possibilities in the delivery process. Packages are handled with care, sorted efficiently, and placed with precision, ensuring safe and secure transportation from point A to point B.
- Moreover, this innovative system sets new standards for environmental sustainability. By optimizing delivery routes and minimizing human intervention, we substantially reduce carbon emissions and contribute to a greener future.



Sustainable development goals

The **Sustainable Development Goals** or **Global Goals** are a collection of seventeen interlinked objectives designed to serve as a shared blueprint for peace and prosperity for people and the planet, now and into the future. The SDGs are no poverty; zero hunger; good health and well-being; quality education; gender equality; clean water and sanitation; affordable and clean energy; decent work and economic growth; industry, innovation and infrastructure; reduced inequalities; sustainable cities and communities; responsible consumption and production; climate action; life below water; life on land; peace, justice, and strong institutions; and partnerships for the goals. The SDGs emphasize the interconnected environmental, social and economic aspects of sustainable development by putting sustainability at their center.

The SDGs were formulated in 2015 by the United Nations General Assembly (UNGA) as part of the Post-2015 Development Agenda, which sought to create a future global development framework to succeed the Millennium Development Goals, which ended that year.They were formally articulated and adopted in a UNGA resolution called the **2030 Agenda**, known colloquially as **Agenda 2030**.

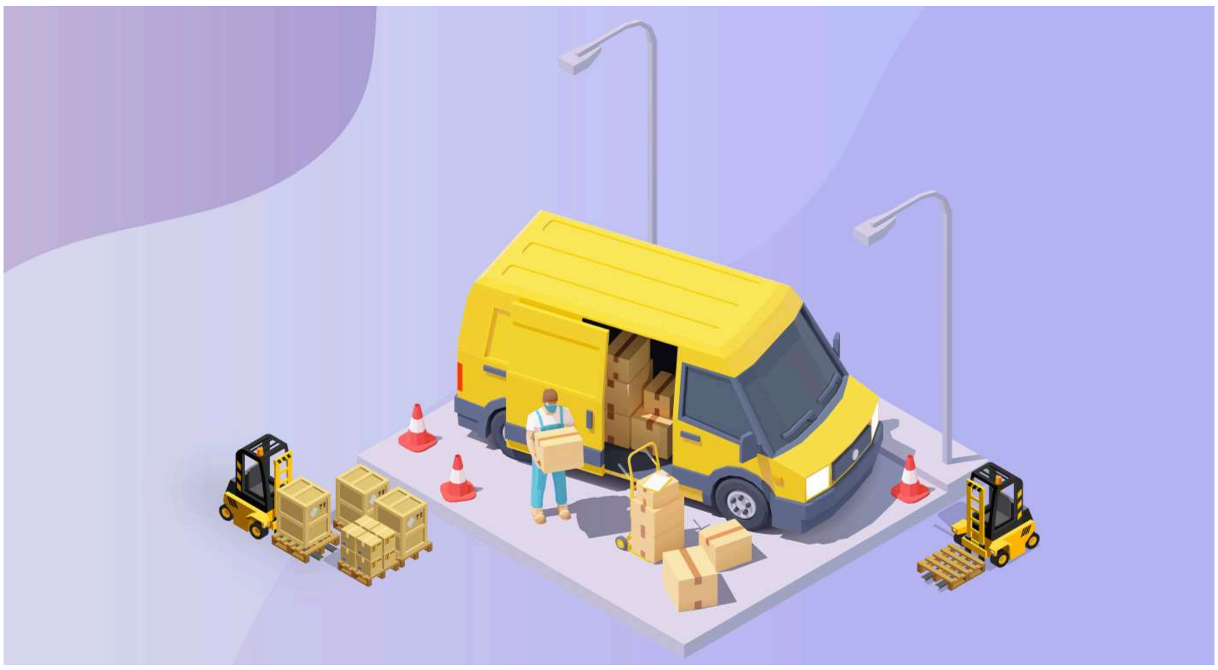
The SDGs being addressed by this project are :

- **SDG 3- Good Health and Wellbeing:** Health and safety of the delivery agent and the consumer is most important. That is why contactless autonomous delivery systems have had a huge impact delivering products in many places during COVID.
- **SDG 9- Industry, Innovation and Infrastructure:** Integrated factory floor delivery systems support the ongoing fourth industrial revolution. Supply chain management and inventory tracking can be easily achieved through these delivery systems. They can be used to deliver objects to the factory floor or they can be used for last mile delivery.
- **SDG 11- Sustainable cities and Communities:** Autonomous delivery systems to deliver final goods to the consumer helps the consumer spend a lot of time driving to the factory floor and wasting time picking up his product. It also saves a lot of energy and reduces emissions since these delivery systems will be mostly powered by clean renewable energy.
- **SDG 12- Responsible Consumption and Production:** As these delivery systems can be easily integrated into the factory floor supply chain and inventory management can be easily checked and it can order raw materials directly from the supplier whenever needed. This is one of the best ways to save mass ordering of perishable goods and then wasting them. It is one of the best possible ways to achieve Just in time delivery.

1 NO POVERTY
2 ZERO HUNGER
3 GOOD HEALTH AND WELL-BEING
4 QUALITY EDUCATION
5 GENDER EQUALITY
6 CLEAN WATER AND SANITATION
7 AFFORDABLE AND CLEAN ENERGY
8 DECENT WORK AND ECONOMIC GROWTH
9 INDUSTRY, INNOVATION AND INFRASTRUCTURE
10 REDUCED INEQUALITIES
11 SUSTAINABLE CITIES AND COMMUNITIES
12 RESPONSIBLE CONSUMPTION AND PRODUCTION
13 CLIMATE ACTION
14 LIFE BELOW WATER
15 LIFE ON LAND
16 PEACE, JUSTICE AND STRONG INSTITUTIONS
17 PARTNERSHIPS FOR THE GOALS
SUSTAINABLE DEVELOPMENT GOALS

Problem Survey

- **Lost packages and delays:** Traditional delivery systems usually lead to lost packages and delays due to a human factor being involved.
- **Increased Delivery Speed:** Automation can expedite the delivery process, reducing delivery times and improving customer satisfaction.
- **Environmental unsustainability**: Traditional delivery systems are one of the major contributors to carbon emissions, the last-mile courier industry annually emits approximately 500 thousand tonnes of CO2 in India, three million tonnes of CO2 in Europe, and four million tonnes of CO2 in the US.
- **Labor costs**: Since traditional delivery systems involve a human factor, the salaries of the laborers add to the transportation costs.
- **Energy inefficiency:** Addressing these energy inefficiencies is essential for traditional delivery systems to reduce their environmental impact, operational costs, and contribute to a more sustainable delivery ecosystem



Literature Survey

Author	Paper/book title	Publication details	Summary
Sambit Mohapatra, Senthil Yogamani, Varun Ravi Kumar, Stefan Milz, Heinrich Gotzig, Patrick Mäder	LiDAR-BEVMTN: Real-Time LiDAR Bird's-Eye View Multi-Task Perception Network for Autonomous Driving	Computer Vision and Pattern Recognition	LiDAR perception has the largest body of literature after camera perception. However, multi-task learning across tasks like detection, segmentation, and motion estimation using LiDAR remains relatively unexplored, especially on automotive-grade embedded platforms.
Yongqi Dong, Tobias Datema, Vincent Wassenaar, Joris van de Weg, Cahit Tolga Kopar, Harim Suleman	Comprehensive Training and Evaluation on Deep Reinforcement Learning for Automated Driving in Various Simulated Driving Maneuvers	under review by the 26th IEEE International Conference on Intelligent Transportation Systems (ITSC 2023)	Results show that the TRPO-based models with modified reward functions delivered the best performance in most cases, for training automated driving on the highway-env simulation platform.
Kaylene C. Stocking, Zak Murez, Vijay Badrinarayanan, Jamie Shotton, Alex Kendall, Claire Tomlin, Christopher P. Burgess	Linking vision and motion for self-supervised object-centric perception	CVPR 2023 Vision-Centric Autonomous Driving workshop	a self-supervised object-centric vision model to perform object decomposition using only RGB video and the pose of the vehicle as inputs.

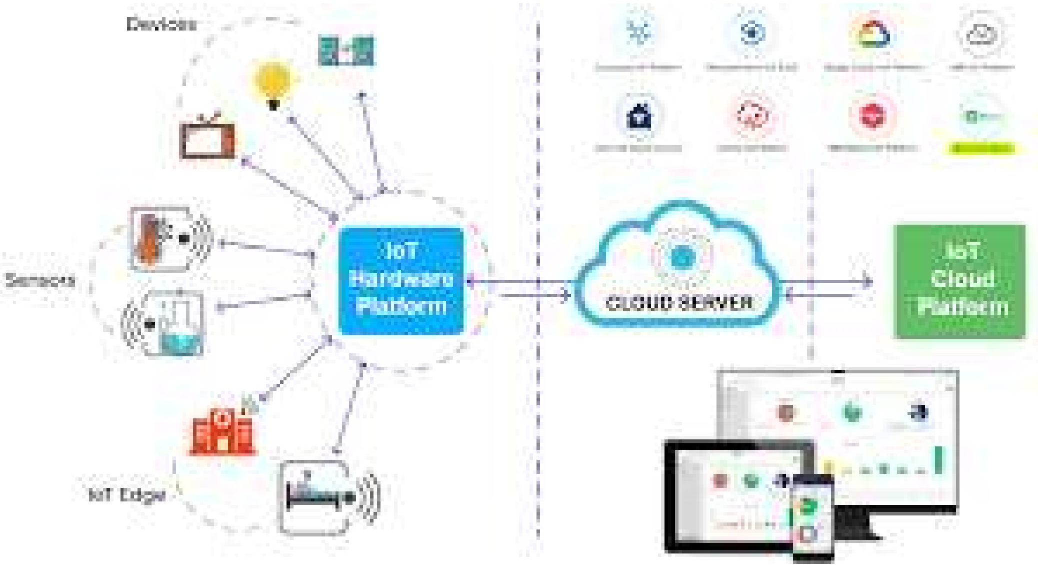
Ruohan Li, Yongqi Dong	Robust Lane Detection through Self Pre-training with Masked Sequential Autoencoders and Fine-tuning with Customized PolyLoss	IEEE Transactions on Intelligent Transportation Systems	A pipeline consisting of self pre-training with masked sequential autoencoders and fine-tuning with customized PolyLoss for the end-to-end neural network models using multi-continuous image frames.
Martin Bikandi, Gorka Velez, Naiara Aginako, Itziar Irigoien	Synthetic outlier generation for anomaly detection in autonomous driving	26th IEEE International Conference on Intelligent Transportation Systems (ITSC 2023)	Most efficient networks for anomaly detection in autonomous driving.
Faqin Gao; Yao Cheng; Ming Gao; Chen Ma	Design and Implementation of an Autonomous Driving Delivery Robot	2022 41st Chinese Control Conference (CCC)	The robot features a completely original design of the modern appearance, compactly and intelligently arranged sensor layout as well as a hardware platform with an extremely flexible four independent steering function.

\

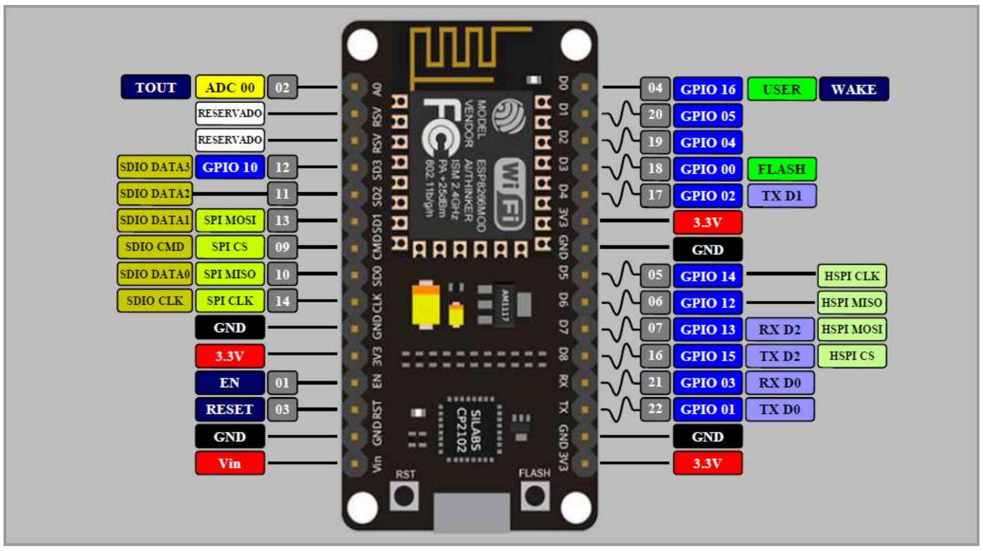
Nisarga U, Prajwal R, Reshma S M, Sonu H, Ms. Ramya B	A Review on Autonomous Delivery Robot using open CV	26th IEEE International Conference on Intelligent Transportation Paper Id : IJRASET43826 Publish Date : 2022-06-04	So, offering an autonomous robot capable to send or get hold of bodily items, in some of different scenarios. Autonomous shipping system, now no longer but smart sufficient to supply items throughout cities.
Masrul Nizam bin Mahmod1 , Mastura binti Ramli2 , Sharifah Nurulhuda Tuan Mohd Yasin3	QR code detection using OpenCV python with tello drone	26th IEEE International Conference on Intelligent Transportation	Our project aimed to develop a solution that only requires a simple vision system to achieve accurate positioning (altitude) in closed spaces. The method is developed in python environment using OpenCV library.

Working of the Prototype

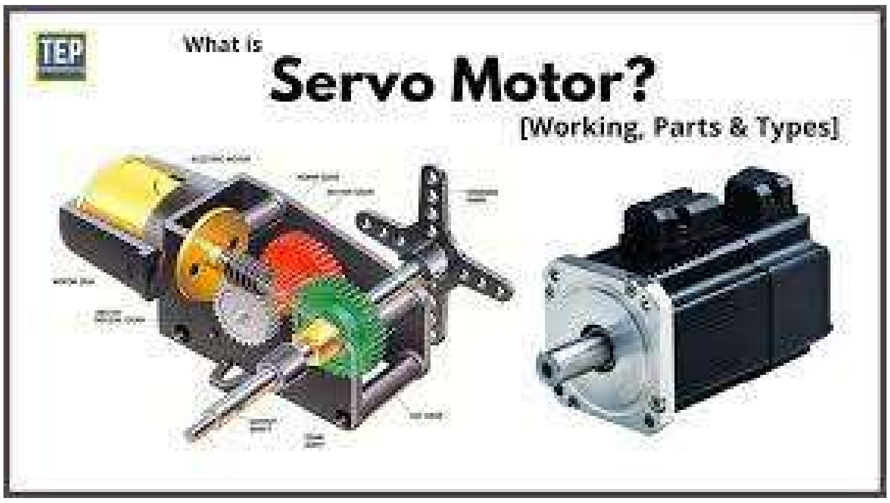
- **Application:** The functioning of the IOT based autonomous delivery system is based on the blynk IOT app. It has a robotic arm planted on its back for transferring the item onto its basket and a QR code on its body for identification of recipient and source.



- **NodeMCU 8266**:
  - NodeMCU ESP8266 is an open-source firmware and development kit that plays a vital role in designing your own IoT product. We use Arduino IDE software for programming this module.
  - The NodeMCU (***N***ode ***M***icro***C***ontroller ***U***nit) is an open-source software and hardware development environment built around an inexpensive System-on-a-Chip (SoC) called the ESP8266. The ESP8266, designed and manufactured by Espressif Systems, contains the crucial elements of a computer: CPU, RAM, networking (WiFi), and even a modern operating system and SDK. That makes it an excellent choice for Internet of Things (IoT) projects of all kinds.
  - However, as a chip, the ESP8266 is also hard to access and use. You must solder wires, with the appropriate analog voltage, to its pins for the simplest tasks such as powering it on or sending a keystroke to the "computer" on the chip. You also have to program it in low-level machine instructions that can be interpreted by the chip hardware. This level of integration is not a problem using the ESP8266 as an embedded controller chip in mass-produced electronics. It is a huge burden for hobbyists, hackers, or students who want to experiment with it in their own IoT projects.
  - The Arduino project created an open-source hardware design and software SDK for their versatile IoT controller. Similar to NodeMCU, the Arduino hardware is a microcontroller board with a USB connector, LED lights, and standard data pins. It also defines standard interfaces to interact with sensors or other boards. But unlike NodeMCU, the Arduino board can have different types of CPU chips (typically an ARM or Intel x86 chip) with memory chips, and a variety of programming environments. There is an Arduino reference design for the ESP8266 chip as well. However, the flexibility of Arduino also means significant variations across different vendors. For example, most Arduino boards do not have WiFi capabilities, and some even have a serial data port instead of a USB port.

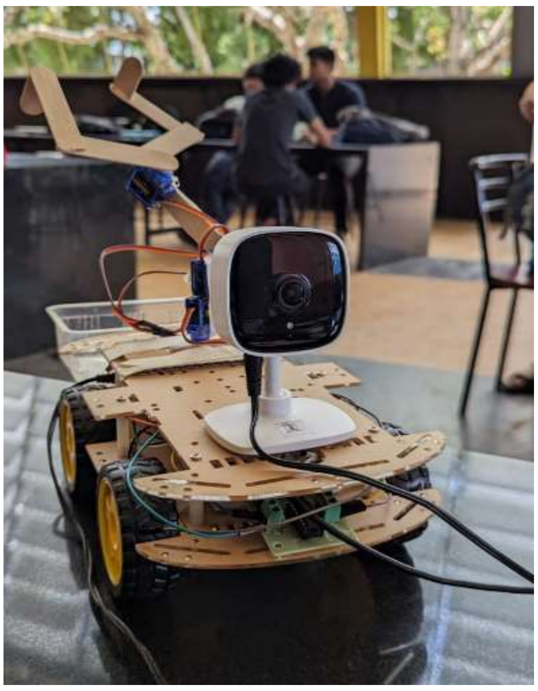


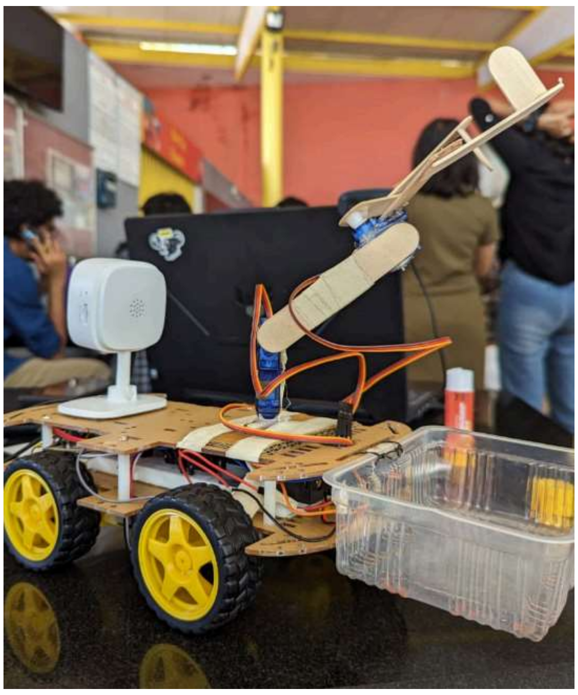
- **SERVO MOTOR**:
  - A **servo motor** is a type of motor that can rotate with great precision. Normally this type of motor consists of a control circuit that provides feedback on the current position of the motor shaft, this feedback allows the servo motors to rotate with great precision
  - It is a special kind of motor which can be programmed to rotate a specified amount (usually a maximum of 180 degrees)
  - It is an electromechanical device which produces torque and velocity based on the apple current / voltage
  - It works as a part of a closed loop system producing torque and velocity based on the servo controller's instructions and sends feedback back to the controller, closing the loop



- **300 RPM BO Motor-Straight:** Its low density, lightweight, and has low inertia, with an operating voltage between 3-12V, which are used to drive the wheels of the electric car. Its rated speed is 300 rpm(revolutions per minute with a rated torque of 35 kg cm

- **Makeshift robotic arm**: We have planted a makeshift robotic arm using ice cream sticks which is capable of picking lightweight objects and placing them in a basket. The links of the robotic arm are composed of servo motors.
- **QR code**: The QR code placed on the electric vehicle which with the help of an external camera is used to identify both the source and the destination and can be used for bookkeeping purposes by the delivery company.





Motivation

- Traditional delivery systems can be slow , time-consuming and often have limited tracking capabilities.Physical delivery systems rely heavily on vehicles and transportation, contributing to air pollution and carbon emissions.Autonomous delivery system offer faster, more efficient, and environmentally friendly alternatives to traditional methods.Hence, the need for autonomous delivery system is now more than ever.
- Recent advances in physics , mathematics and C programming offer leapfrogging opportunities to develop next generation delivery systems like autonomous delivery systems.
- To study the cost required to set up an autonomous delivery system and to come up with a working prototype which uses OpenCV and blynk IOT whose cost is less than those available commercially and make it affordable .

Objectives

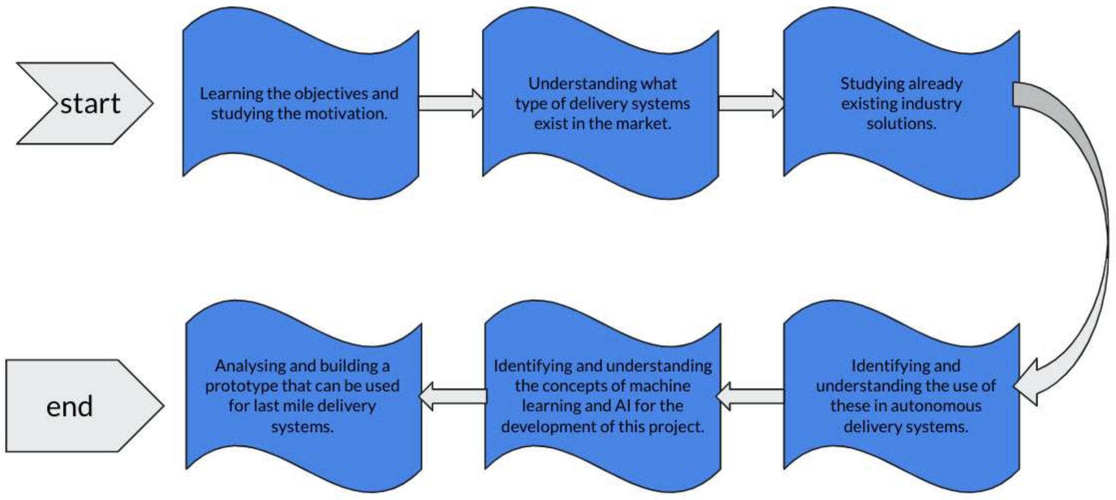
An interdisciplinary approach is used in the research to improve the efficiency of the delivery system ,which also ensures successful delivery of the package and to create a working prototype of the same.

The main objectives include :

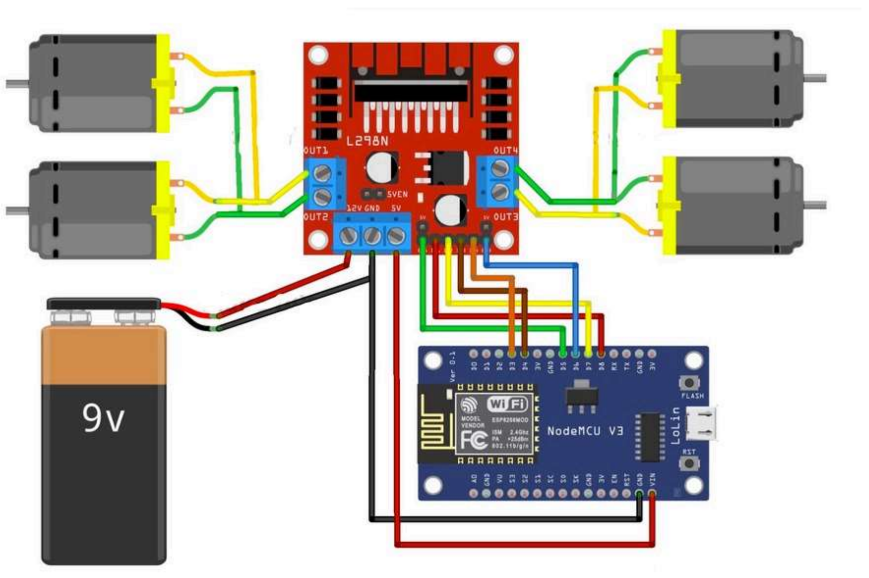
- Usage of concepts of physics and sensors to create a working model of an autonomous delivery system which is faster, more efficient and more environment-friendly.
- Usage of robotic arm which is used to pick up and drop the package.
- Usage of OpenCV which is used to detect any obstruction in the path thus making sure that the package is delivered successfully.
- Usage of blynk IOT to track and check delivery status.

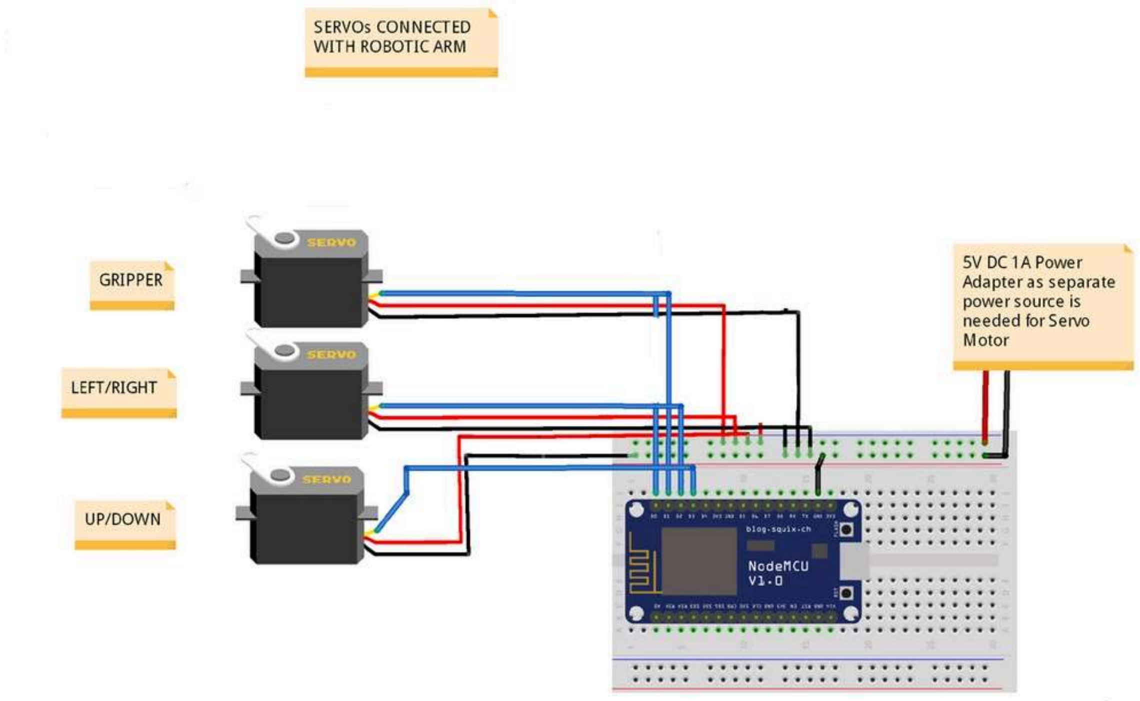
Methodology

- Learning the objectives and understanding the motivation behind the project.
- Understanding and studying the various delivery systems which exist in the market today and their advantages and drawbacks.
- Identifying the ways to implement these systems in an autonomous delivery system.
- Understanding the basics of AI/ML to implement the autonomous delivery system
- Analysing and building a prototype of a solution that can be used for last mile delivery


start
Learning the objectives and studying the motivation.
Understanding what type of delivery systems exist in the market.
Studying already existing industry solutions.
Identifying and understanding the use of these in autonomous delivery systems.
Identifying and understanding the concepts of machine learning and AI for the development of this project.
Analysing and building a prototype that can be used for last mile delivery systems.
end

Schematic Diagrams

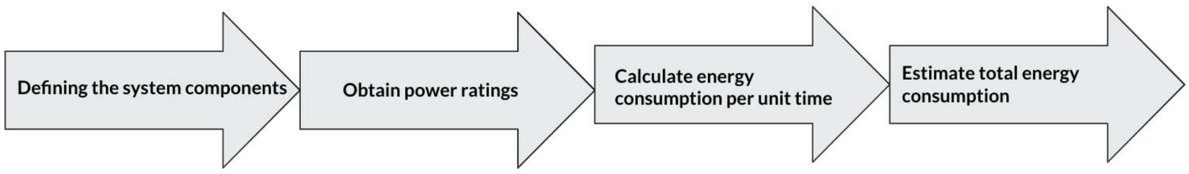



SERVOs CONNECTED WITH ROBOTIC ARM
GRIPPER
LEFT/RIGHT
UP/DOWN
SERVO
SERVO
SERVO
5V DC 1A Power Adapter as separate power source is needed for Servo Motor
NodeMCU V1.0
blog.squix.ch

Energy Consumption Analysis

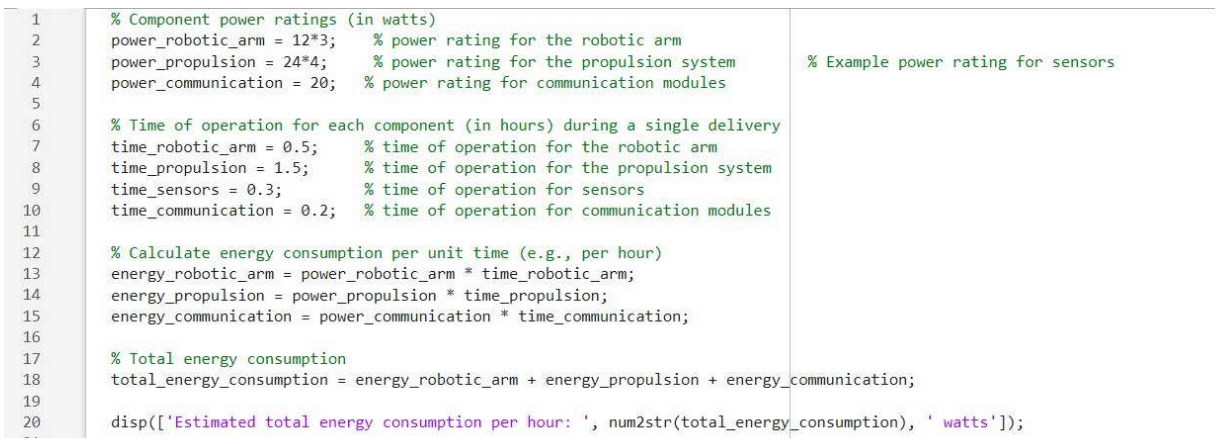
To estimate the amount of energy being used by the delivery system MATLAB can be used. A model is made using MATLAB by taking in information from various sensors about the power that is required for propulsion, sensing and other components. This simulation is then run over and over again and it is modified to be more efficient. The energy consumption in the system is optimized and its endurance is assessed.

The four main steps in the analysis are, defining the system components, obtaining the power ratings, calculating energy consumption per unit time and finally estimating the total energy consumed.

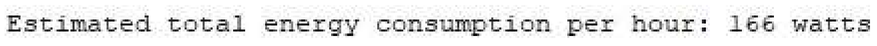

Defining the system components
Obtain power ratings
Calculate energy consumption per unit time
Estimate total energy consumption

Identify the main components of the automatic delivery system that consume energy. For each component, determine their power ratings or energy consumption specifications. Power ratings are typically given in watts (W) or kilowatts (kW)/ Once you have the power ratings, calculate the energy consumption per unit time (e.g., per hour) for each component. Sum up the energy consumption of all components to estimate the total energy consumption per unit time (e.g., per hour) for the entire automatic delivery system.

Code & Output



```
% Component power ratings (in watts)
power_robotic_arm = 12*3;     % power rating for the robotic arm
power_propulsion = 24*4;      % power rating for the propulsion system        % Example power rating for sensors
power_communication = 20;    % power rating for communication modules

% Time of operation for each component (in hours) during a single delivery
time_robotic_arm = 0.5;      % time of operation for the robotic arm
time_propulsion = 1.5;       % time of operation for the propulsion system
time_sensors = 0.3;          % time of operation for sensors
time_communication = 0.2;    % time of operation for communication modules

% Calculate energy consumption per unit time (e.g., per hour)
energy_robotic_arm = power_robotic_arm * time_robotic_arm;
energy_propulsion = power_propulsion * time_propulsion;
energy_communication = power_communication * time_communication;

% Total energy consumption
total_energy_consumption = energy_robotic_arm + energy_propulsion + energy_communication;

disp(['Estimated total energy consumption per hour: ', num2str(total_energy_consumption), ' watts']);
```



```
Estimated total energy consumption per hour: 166 watts
```

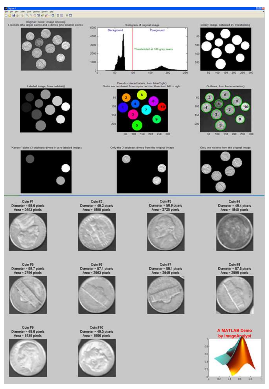
Mathematics in Image Processing

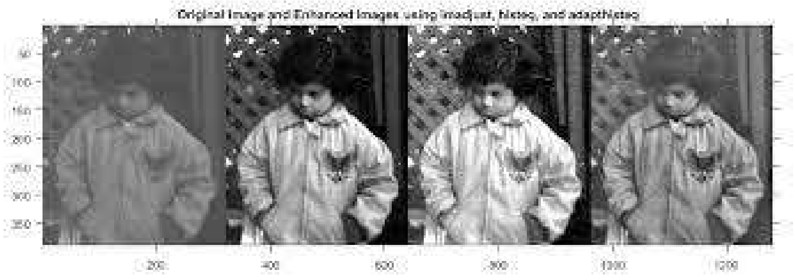
Image processing involves using mathematical analysis on an image, which can be either a static picture or a video. The resulting output may not necessarily be another image. For instance, attempting to eliminate the glare caused by the Sun's reflection in an image constitutes a type of image processing. Similarly, generating a line graph depicting the fluctuating light intensity in individual frames from a video clip is a valid image processing method. Recent years have witnessed significant advancements in image processing, largely driven by various scientific breakthroughs. Cameras have undergone a remarkable transformation over the past century, evolving from costly, grainy black and white devices to today's ubiquitous high-definition cameras found in most mobile phones. These technological strides have opened up new opportunities and posed fresh challenges. With the advent of larger telescopes enabling deep space observation, there is a disparity in the amount of data collected between distant and closer observations. Consequently, there is a growing need for enhanced techniques to reconstruct missing data and more effectively detect and eliminate noise.

In our project we have made the use of a camera to capture the image and transmit live video. To process the data that is collected, the images that are collected by the camera need to be processed. The main processing tasks include de-noising, de-blurring, enhancement, segmentation and edge detection. These processing techniques incorporate a lot of mathematical methods, models and techniques. They can be used in various different fields of study.

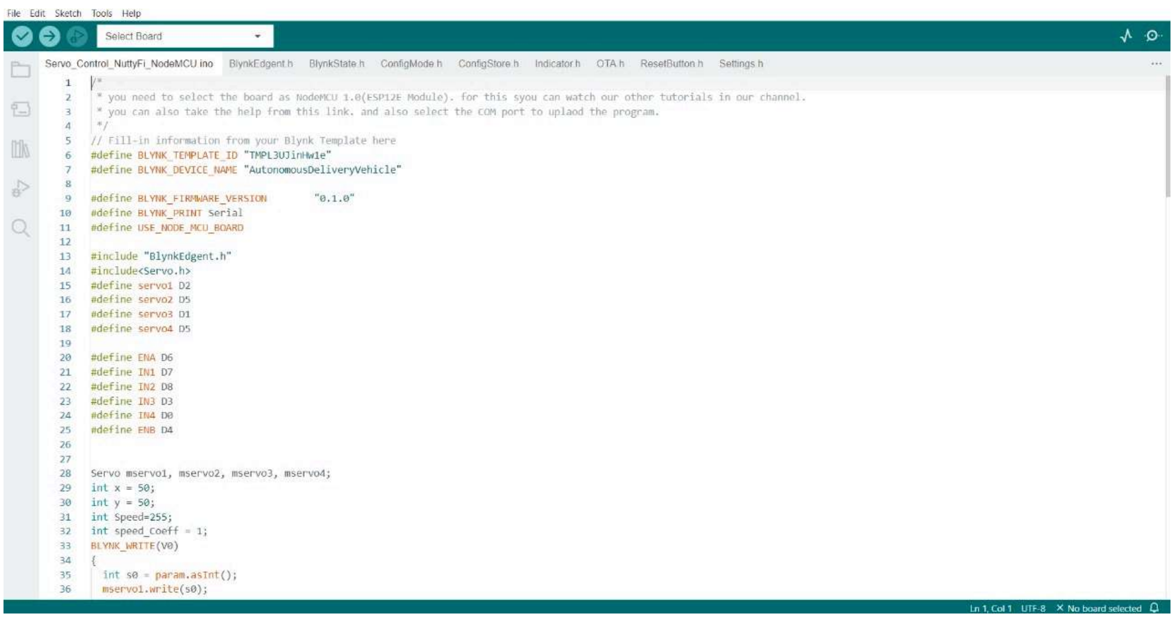
For image filtering in the spatial domain, first and second order derivatives, gradient descent algorithm, laplacian, averaging filters, order statistics filters, discrete approximations by finite differences and convolutions are used. In the frequency domain, low-pass, high-pass, median-pass filters and fourier transforms are used.

MATLAB is one of the most popular tools used for building models using AI technologies such as ANN, Fuzzy and hybrid algorithms such as ANFIS. It provides an Image Processing Toolbox as a part of its package. This toolbox provides capability to perform Image Processing operations, including: Image Segmentation, Image Enhancement, Noise Reduction, Three dimensional Image processing.

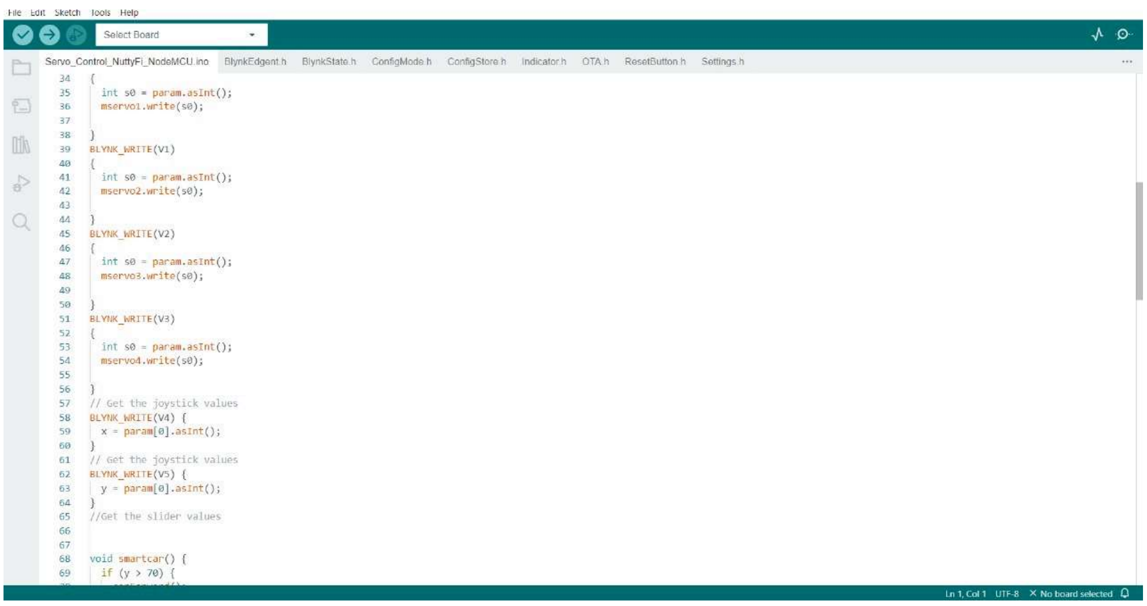



Original Image and Enhanced Images using imadjust, histeq, and adapthisteq

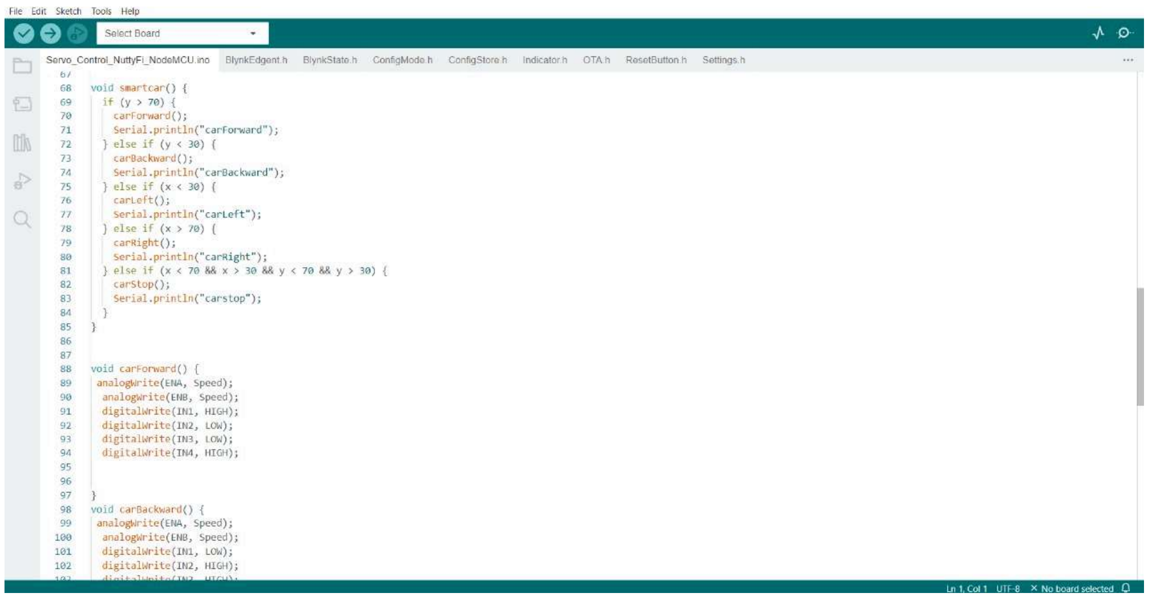
Arduino Code



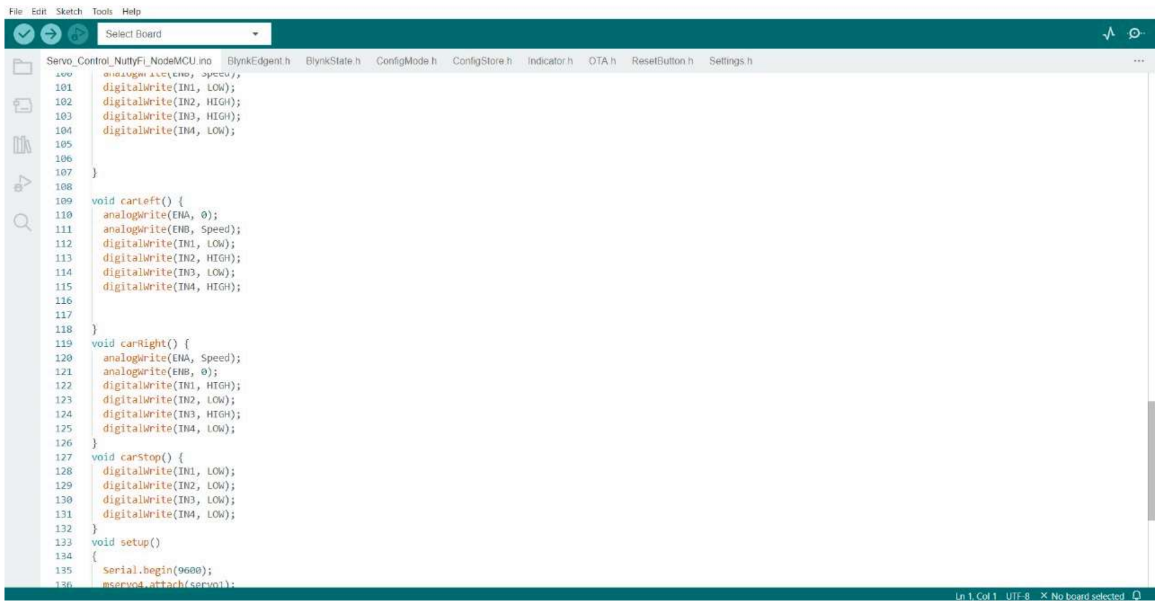
```
/*
 * you need to select the board as NodeMCU 1.0(ESP12E Module). for this syou can watch our other tutorials in our channel.
 * you can also take the help from this link. and also select the COM port to uplaod the program.
 */
// Fill-in information from your Blynk Template here
#define BLYNK_TEMPLATE_ID "TMPL3UJinHw1e"
#define BLYNK_DEVICE_NAME "AutonomousDeliveryVehicle"

#define BLYNK_FIRMWARE_VERSION        "0.1.0"
#define BLYNK_PRINT Serial
#define USE_NODE_MCU_BOARD

#include "BlynkEdgent.h"
#include<Servo.h>
#define servo1 D2
#define servo2 D5
#define servo3 D1
#define servo4 D5

#define ENA D6
#define IN1 D7
#define IN2 D8
#define IN3 D3
#define IN4 D0
#define ENB D4


Servo mservo1, mservo2, mservo3, mservo4;
int x = 50;
int y = 50;
int Speed=255;
int speed_Coeff = 1;
BLYNK_WRITE(V0)
{
  int s0 = param.asInt();
  mservo1.write(s0);
```



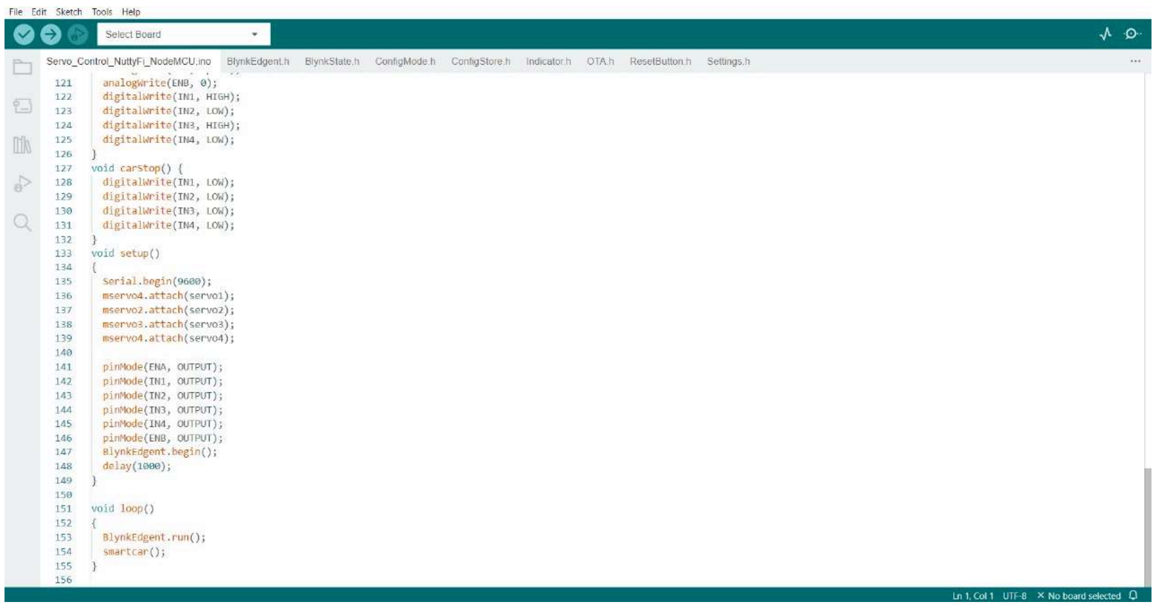
```
{
  int s0 = param.asInt();
  mservo1.write(s0);

}
BLYNK_WRITE(V1)
{
  int s0 = param.asInt();
  mservo2.write(s0);

}
BLYNK_WRITE(V2)
{
  int s0 = param.asInt();
  mservo3.write(s0);

}
BLYNK_WRITE(V3)
{
  int s0 = param.asInt();
  mservo4.write(s0);

}
// Get the joystick values
BLYNK_WRITE(V4) {
  x = param[0].asInt();
}
// Get the joystick values
BLYNK_WRITE(V5) {
  y = param[0].asInt();
}
//Get the slider values


void smartcar() {
  if (y > 70) {
```



```
void smartcar() {
  if (y > 70) {
    carForward();
    Serial.println("carForward");
  } else if (y < 30) {
    carBackward();
    Serial.println("carBackward");
  } else if (x < 30) {
    carLeft();
    Serial.println("carLeft");
  } else if (x > 70) {
    carRight();
    Serial.println("carRight");
  } else if (x < 70 && x > 30 && y < 70 && y > 30) {
    carStop();
    Serial.println("carstop");
  }
}


void carForward() {
 analogWrite(ENA, Speed);
  analogWrite(ENB, Speed);
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, HIGH);


}
void carBackward() {
 analogWrite(ENA, Speed);
  analogWrite(ENB, Speed);
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, HIGH);
  digitalWrite(IN3, HIGH);
```



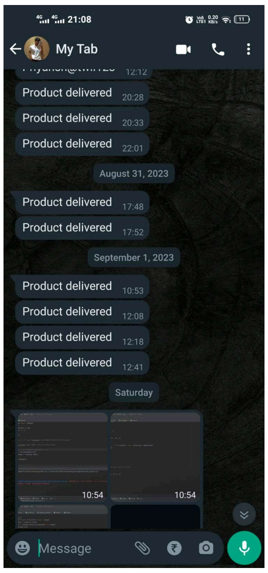
```
  analogWrite(ENB, Speed);
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, HIGH);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);


}

void carLeft() {
  analogWrite(ENA, 0);
  analogWrite(ENB, Speed);
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, HIGH);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, HIGH);


}
void carRight() {
  analogWrite(ENA, Speed);
  analogWrite(ENB, 0);
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);
}
void carStop() {
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, LOW);
}
void setup()
{
  Serial.begin(9600);
  mservo4.attach(servo1);
```



```
  analogWrite(ENB, 0);
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);
}
void carStop() {
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, LOW);
}
void setup()
{
  Serial.begin(9600);
  mservo4.attach(servo1);
  mservo2.attach(servo2);
  mservo3.attach(servo3);
  mservo4.attach(servo4);

  pinMode(ENA, OUTPUT);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
  pinMode(ENB, OUTPUT);
  BlynkEdgent.begin();
  delay(1000);
}

void loop()
{
  BlynkEdgent.run();
  smartcar();
}

```

Results

These images show the movement of the vehicle, the controller (on the phone), the picking up of the object, the live video relay being displayed on the phone and the message sent when the QR code is scanned to say that the parcel has been delivered.


21:08

My Tab

Product delivered 20:28

Product delivered 20:33

Product delivered 22:01

August 31, 2023

Product delivered 17:48

Product delivered 17:52

September 1, 2023

Product delivered 10:53

Product delivered 12:08

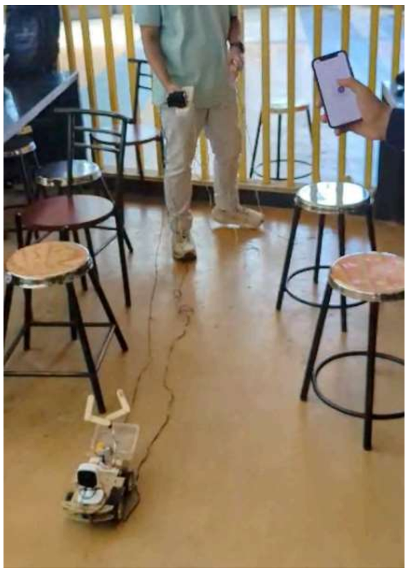
Product delivered 12:18

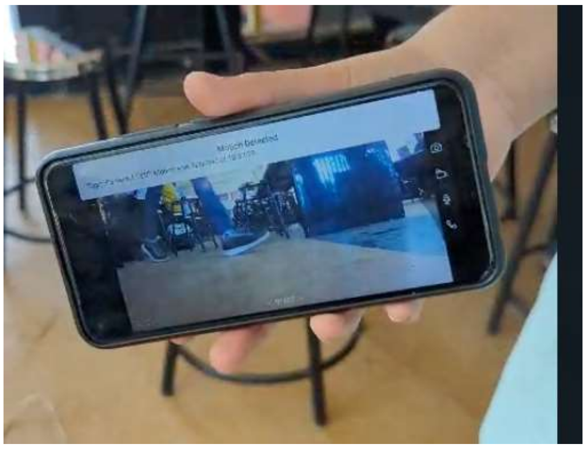
Product delivered 12:41

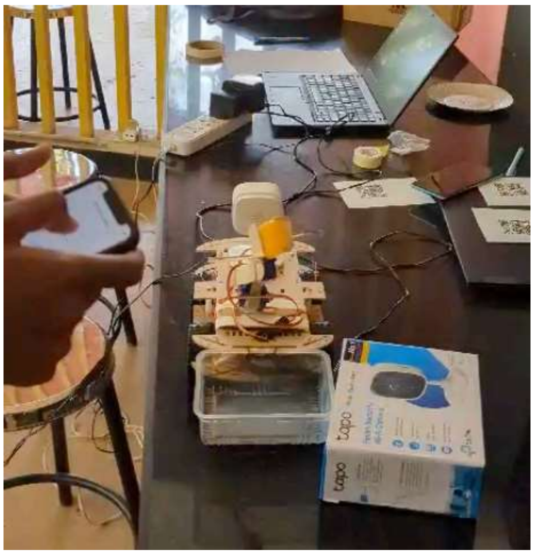
Saturday

10:54

10:54







References

- https://ottonomy.io/how-it-works/
- Sharan Srinivas, Surya Ramachandiran, Suchithra Rajendran, Autonomous robot-driven deliveries: A review of recent developments and future directions, Transportation Research Part E: Logistics and Transportation Review, Volume 165, 2022, 102834, ISSN 1366-5545, https://doi.org/10.1016/j.tre.2022.102834.
- M. M. Abrar, R. Islam and M. A. H. Shanto, "An Autonomous Delivery Robot to Prevent the Spread of Coronavirus in Product Delivery System," 2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York, NY, USA, 2020, pp. 0461-0466, doi: 10.1109/UEMCON51285.2020.9298108.
- Nisarga U, Prajwal R, Reshma S M, Sonu H, Ms. Ramya B, A Review on Autonomous Delivery Robot using Machine Learning, Ijraset Journal For Research in Applied Science and Engineering Technology, 2022-06-04, DOI Link: https://doi.org/10.22214/ijraset.2022.43826