RV College of Engineering ®

Go, change the world

EXPERIENTIAL LEARNING PROJECT

COURSE:

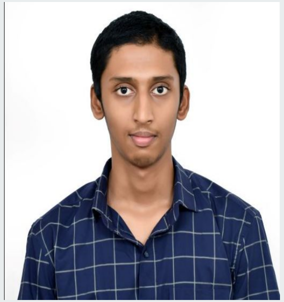
1. IDEA LAB (22ME28)

TOPIC: Autonomous Delivery Systems

SUBMITTED TO:

1. Prof. Chandra Kummar, Department of Mechanical Engineering

TEAM INTRODUCTION



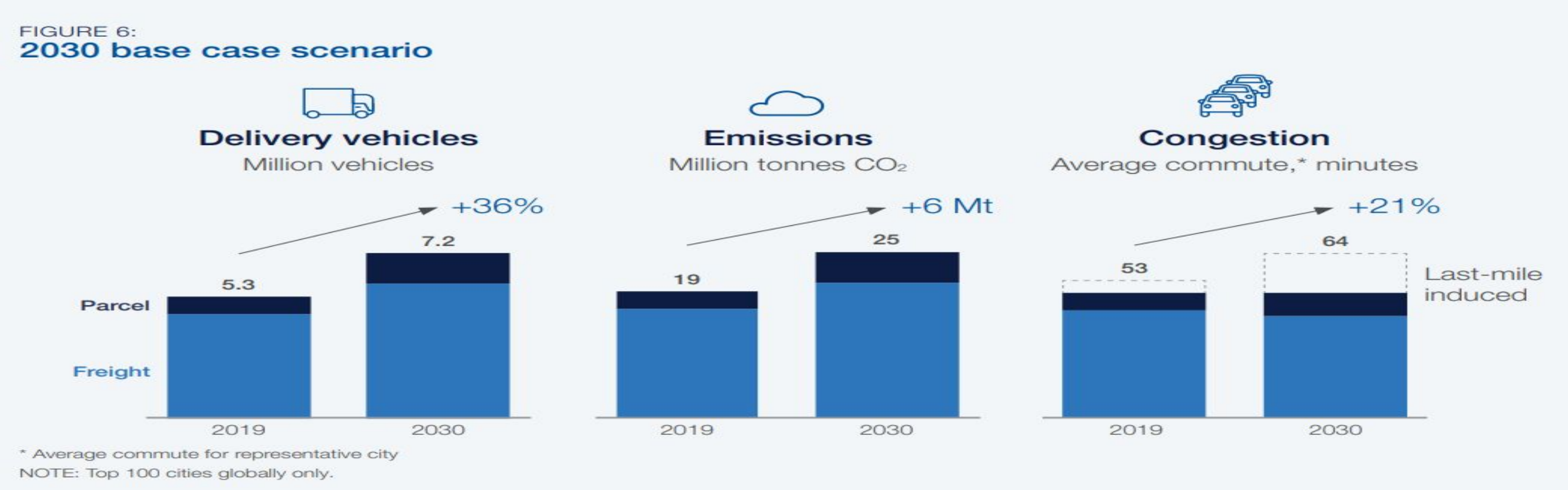

Admission number	Name	Email Id	Mobile number
RVCE22BCS202	Pranav Darshan	pranavdarshan.cs22@rvce.edu.in	8867167440
RVCE22BCS174	Priyansh Rajiv Dhotar	priyanshrajivd.cs22@rvce.edu.in	7676433089
RVCE22BCS097	Raghuveer Rajesh	raghuveernr.cs22@rvce.edu.in	8310054684
RVCE22BCS047	Pavan Shivakumar	pshivakumar.cs22@rvce.edu.in	8618860022

Contents

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

- **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 to pick up his product. It also a lot 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.

Problem Statement

BUILD AN AUTONOMOUS DELIVERY SYSTEM USING IMAGE PROCESSING TO START AND END THE DELIVERY PROCESS.

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 system like autonomous delivery system.

- To study the cost required to set up an autonomous delivery system and to come up with working prototype which uses OpenCV and blynk IOT whose cost is less than those available commercially and make it affordable .

Problem Survey

1. **Lost packages and delays-**Traditional delivery systems usually lead to lost packages and delays due to a human factor being involved.
2. **Increased Delivery Speed-**Automation can expedite the delivery process, reducing delivery times and improving customer satisfaction.
3. **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 CO_2 in India, three million tonnes of CO_2 in Europe, and four million tonnes of CO_2 in the US.
4. **Labor costs**- Since traditional delivery systems involve a human factor, the salaries of the labors add to the transportation costs.
5. **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

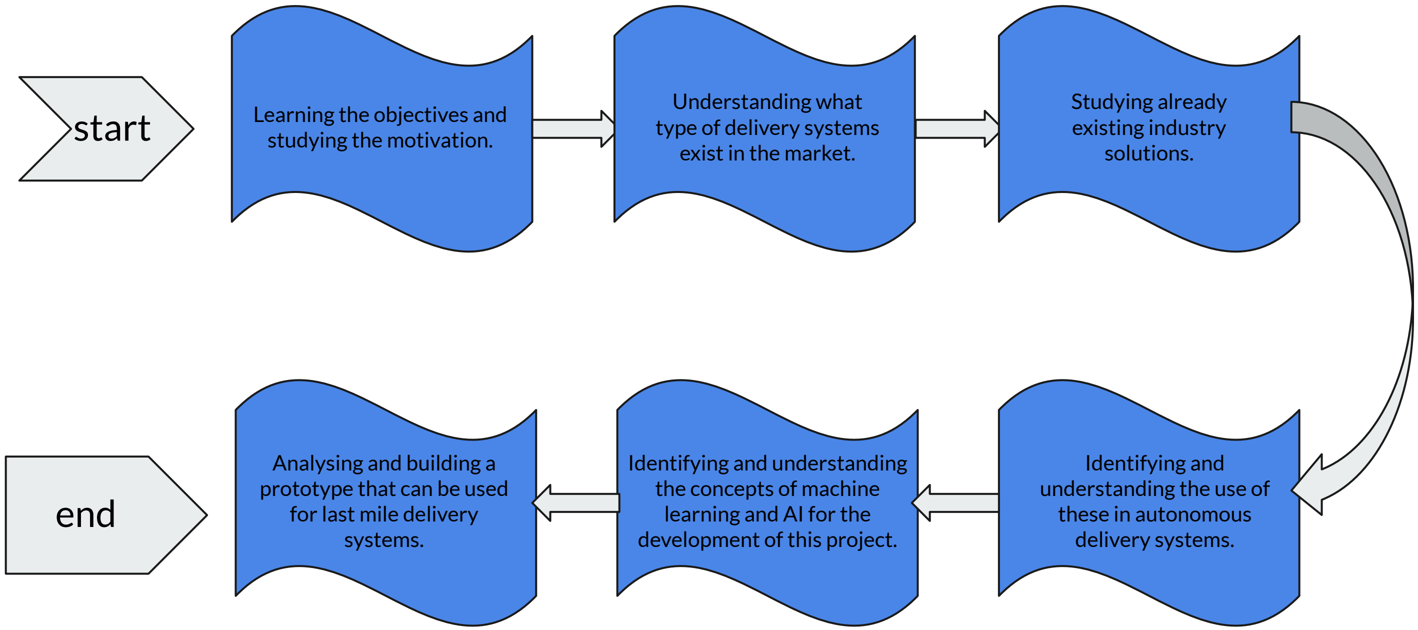
Literature Survey

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.

METHODOLOGY



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.

TIMELINE

PHASE 1

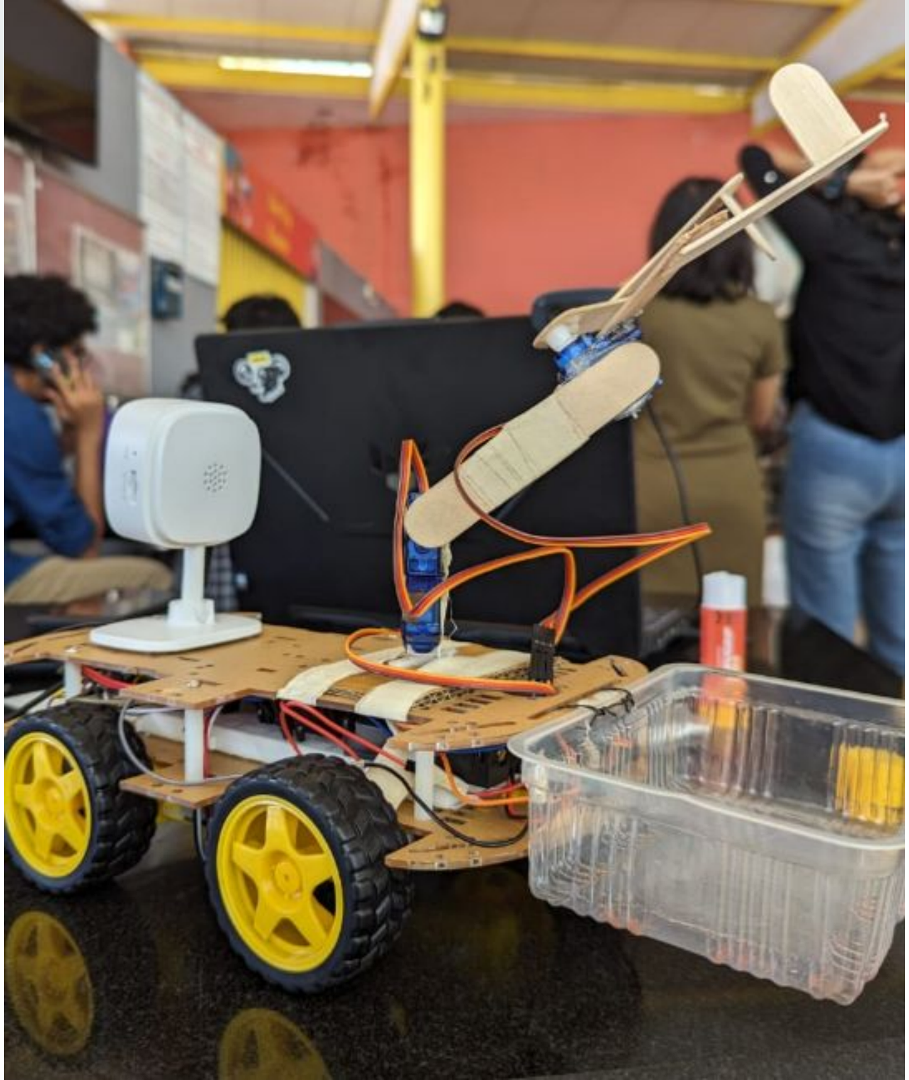
- Identified Problem Statement
- Conducted literature Reviews.
- Acquired knowledge for achieving objective.

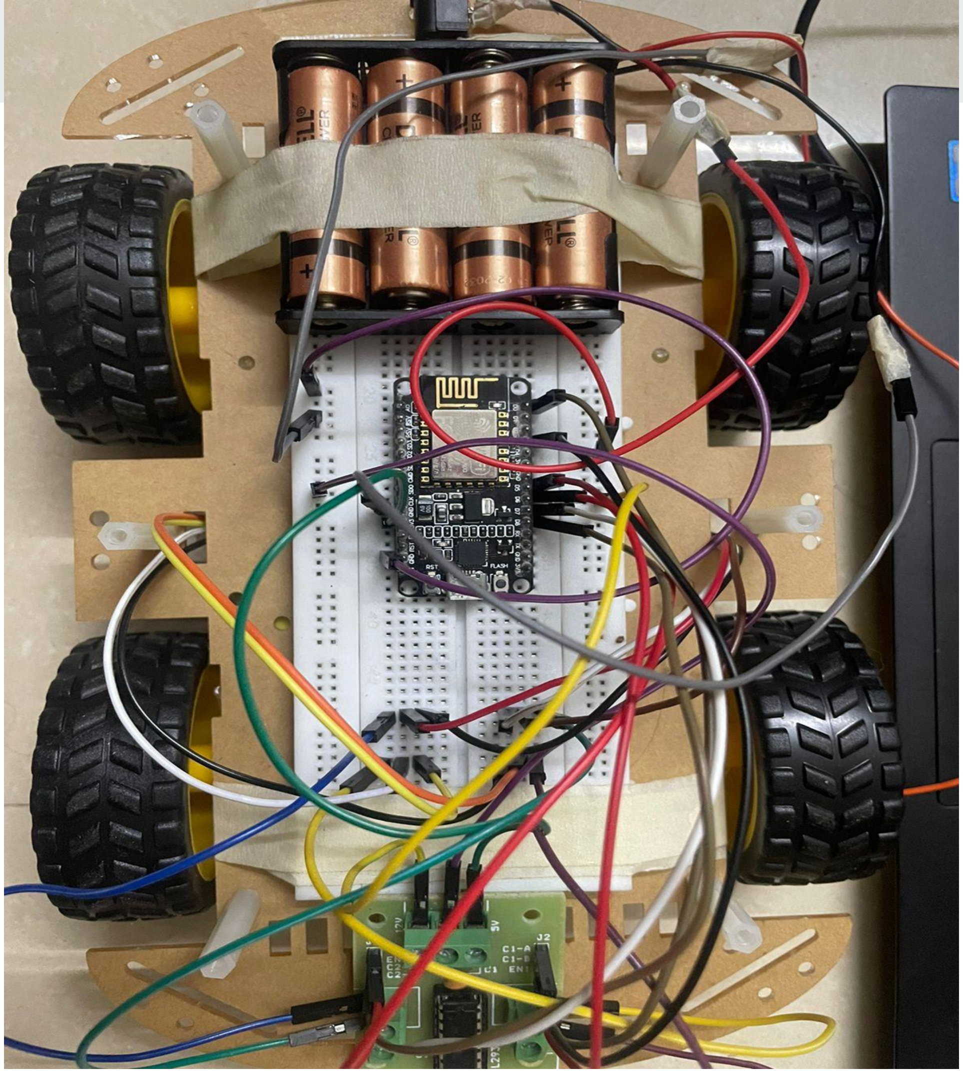
PHASE 2

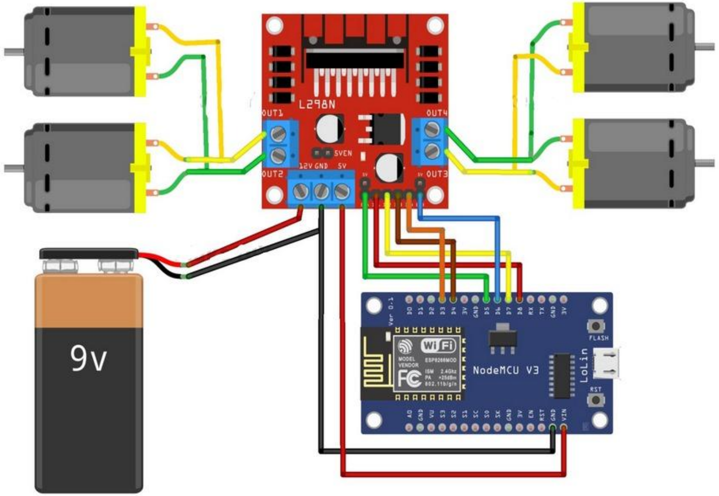
- Model on Internet Controlled Delivery System

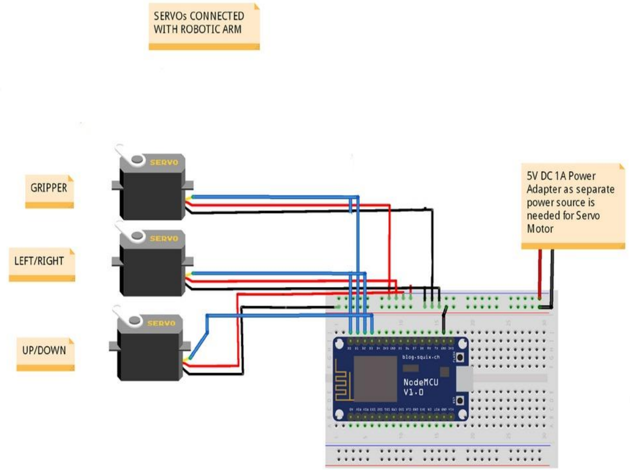
PHASE 3

- Making an Autonomous Delivery System using OpenCV



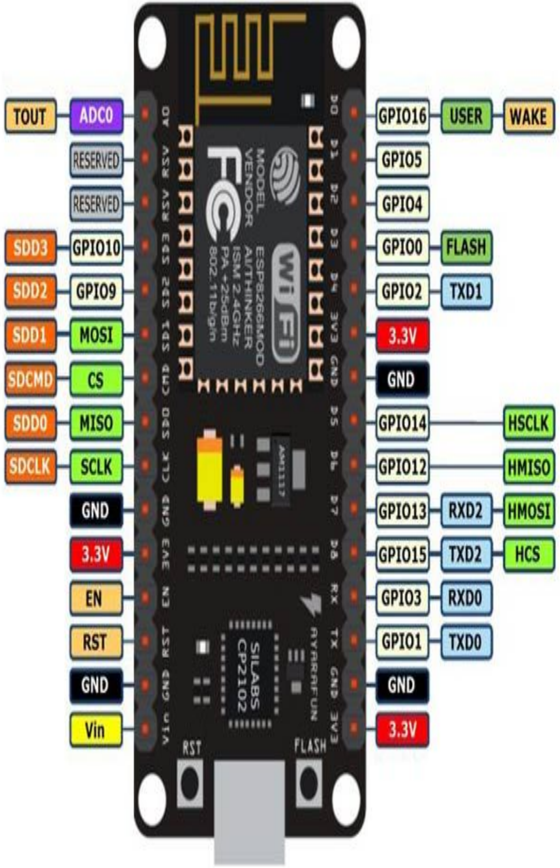

L298N
OUT1
OUT2
OUT3
OUT4
12V GND 5V
5VEN
9v
NodeMCU V3
LoLin
FLASH
RST


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

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.

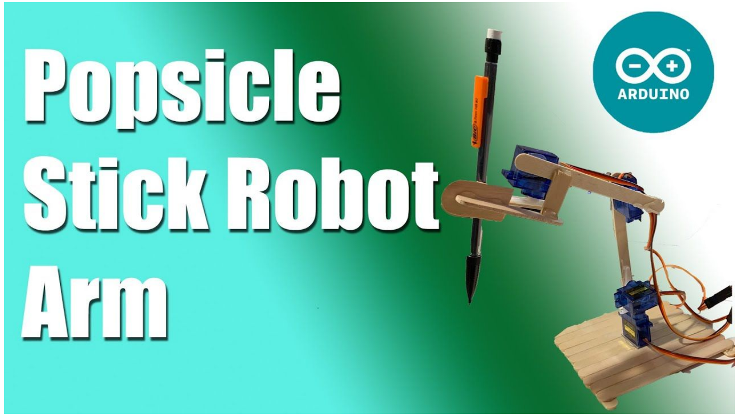
NodeMCU 8266: NodeMCU ESP8266 is an open-source firmware and development kit that plays a vital role in designing your own IoT product. We use Arduino IDE software for programming this module.



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.

300 RPM BO Motor-Straight: Its low density, lightweight, and has low inertia, with and 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 kgcm

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.

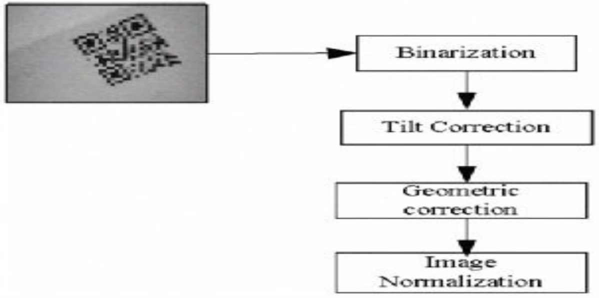


Fig. 1. Recognition Process

File Edit Sketch Tools Help

Select Board

Servo_Control_NuttyFi_NodeMCU.ino | BlynkEdgent.h | BlynkState.h | ConfigMode.h | ConfigStore.h | Indicator.h | OTA.h | ResetButton.h | Settings.h

```
/*
 * 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);
```

Ln 1, Col 1 UTF-8 × No board selected

File Edit Sketch Tools Help

Select Board

Servo_Control_NuttyFi_NodeMCU.ino | BlynkEdgent.h | BlynkState.h | ConfigMode.h | ConfigStore.h | Indicator.h | OTA.h | ResetButton.h | Settings.h

```
{
  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) {
```

Ln 1, Col 1 UTF-8 ✕ No board selected

File Edit Sketch Tools Help

Select Board

Servo_Control_NuttyFi_NodeMCU.ino | BlynkEdgent.h | BlynkState.h | ConfigMode.h | ConfigStore.h | Indicator.h | OTA.h | ResetButton.h | Settings.h

```
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);
```

Ln 1, Col 1 UTF-8 ✕ No board selected

File Edit Sketch Tools Help

Select Board

Servo_Control_NuttyFi_NodeMCU.ino BlynkEdgent.h BlynkState.h ConfigMode.h ConfigStore.h Indicator.h OTA.h ResetButton.h Settings.h

```
  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);
```

Ln 1, Col 1 UTF-8 ✕ No board selected

File Edit Sketch Tools Help

Select Board

Servo_Control_NuttyFi_NodeMCU.ino | BlynkEdgent.h | BlynkState.h | ConfigMode.h | ConfigStore.h | Indicator.h | OTA.h | ResetButton.h | Settings.h

```
  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();
}
```

Ln 1, Col 1 UTF-8 × No board selected

Energy Consumption Analysis

MATLAB can be used to estimate the energy consumption of the delivery system. Create a model that takes into account the power required for propulsion, sensing, and other system components. By simulating different delivery scenarios, you can optimize the system's energy efficiency and assess its endurance.

1. Defining the system components
2. Obtain power ratings
3. Calculate energy consumption per unit time
4. 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.

Mathematics used in image processing

Processing tasks: just like, De-noising, De-blurring, Enhancement, Segmentation and Edge Detection etc. The study of such Image Processing tasks provides a unique opportunity of incorporating Mathematical tools and techniques to address several of the Image Processing applications in various scientific fields of study.

Multiple Mathematical Techniques are used, i.e. for Image Filtering in the Spatial domain (using first- and second order partial derivatives, the gradient, Laplacian, and their discrete approximations by finite differences, averaging filters, order statistics filters, convolution), and in the frequency domain (Fourier transform, low-pass and high-pass filters), zerocrossings of the Laplacian, etc.

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 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

Input Image	Output Image	Integer Coordinates
f (x, y)	g (x, y)	(x, y) with $0 \leq x \leq M - 1$ and $0 \leq y \leq N - 1$.

References

1. https://ottonomy.io/how-it-works/
2. 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.
3. 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.
4. 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