WRO FUTURE ENGINEERS 2025
DOCUMENTATION

AUTONEX

REG.NO-1425

TEAM MEMBERS:

1. SANJAI ARUNPRASHANA D
2. VIJAYARAYAR.S
3. V.E.LOKESH

COACH:

KARTHIKEYAN P

ABOUT AUTONEX BOT:

The Autonex Competition Kit is a 4-wheel robot DIY kit tailored for high-performance in AI and robotics competitions, especially aligned with the WRO (World Robot Olympiad) Future Engineers category. With modular hardware and AI-driven capabilities, this kit allows participants to build, program, and train their robot to tackle real-world tasks using vision and sensor data.

1. HARDWARE PROVIDED:

1. **CVPro Controller:**

The Autonex Controller is based on the ESP32 microcontroller and efficiently processes sensor data.
It includes:

- An RGB LED for status indication.
- A start pushbutton to initiate programmed tasks.
- An integrated 3200 mAh battery with a built-in charging module.

2. **Servo Motor with Ackermann Steering System:**

Provides precise directional control, replicating the steering mechanism of real-world vehicles for smooth and efficient turns.

3. **BO Motor:**

Drives the robot's forward and backward movement.

4. **Color Sensor:**

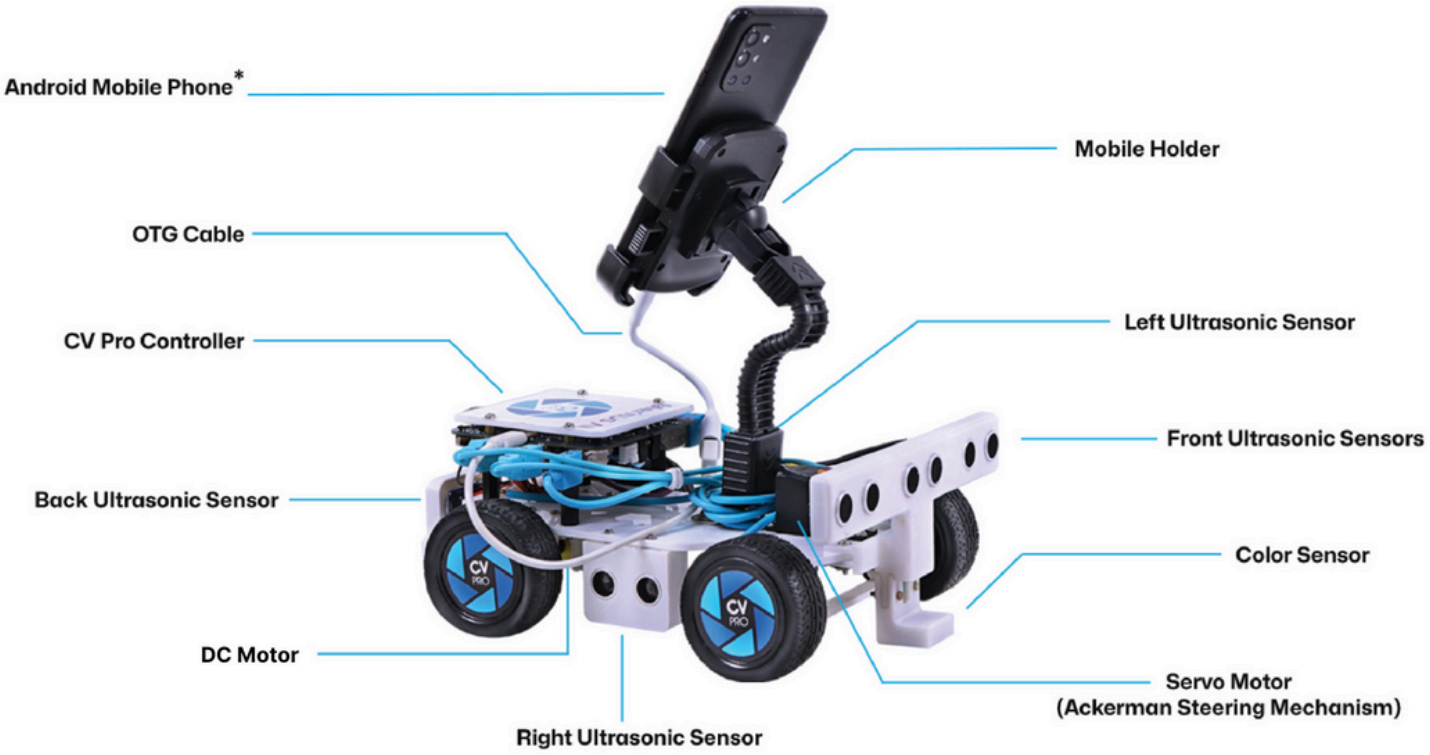
Detects colored paths or zones on the ground—ideal for line-following or zone-based navigation tasks.

5. **Ultrasonic Sensors (6 units):**

- Front (3): Detects obstacles ahead with enhanced accuracy.
- Rear (1): Improves safety during reverse movement.
- Left (1) & Right (1): Enables lateral obstacle detection and supports wall-following tasks.

6. **Mobile Phone Holder with OTG Cable:**

Securely holds a mobile device for applications such as image processing, remote control, or augmented vision.



2. SOFTWARE USED:

1. **Arduino IDE**

A cross-platform programming environment used to write, compile, and upload code to the Autonex Controller.

2. **Arduino C++**

The programming language used to develop code for the Autonex bot, enabling control of its motors, sensors, and other components.

3. **Autonex Android Mobile Application**

A dedicated mobile app designed for computer vision–based tasks, such as object detection, tracking, or image processing.

3. DIMENSIONS AND WEIGHT:

- The Autonex Bot dimensions are: Length – 280 mm, Breadth – 190 mm, Height – 250 mm.
- The total weight of the bot is 975 g (without a mobile phone) and 1,120 g (with a mobile phone).

MOBILITY MANAGEMENT

1. OVERVIEW OF MOBILITY MANAGEMENT

1.1 PURPOSE AND IMPORTANCE OF MOBILITY MANAGEMENT:

- Mobility management ensures that the vehicle can move efficiently, reliably, and with control in its environment.
- It involves selecting suitable motors, designing proper control systems, and integrating mechanical elements like wheels and chassis.
- The goal is to allow the bot to move forward, backward, and steer accurately. This is essential for navigation, obstacle avoidance, and task execution.
- With correct mobility management, power is used efficiently and the bot maintains stability.
- It also extends the lifespan of mechanical components by reducing strain and improving balance.
- Ultimately, mobility management is the foundation for responsive and intelligent robotic motion.

1.2 VEHICLE MOVEMENT STRATEGIES:

- This bot uses a hybrid movement strategy combining a single BO motor for forward and backward motion, and a servo-controlled Ackermann steering mechanism for turning left and right.
- Unlike differential drive or skid-steer, where motion is based on varying wheel speeds, Ackermann steering replicates how real vehicles turn using angle-controlled front wheels.
- This makes the bot's movement smoother and more predictable. The combination reduces mechanical complexity while improving control in tighter navigation paths.

1.3 CONTROL SYSTEMS OVERVIEW:

- The movement of the bot is managed through an ESP32-based Autonexcontroller, which supports autonomous control mode.
- The controller uses inputs from multiple ultrasonic sensors (F1, F2, front, back, left, right) to detect obstacles and guide navigation.
- It also uses color sensor (TCS34725) temperature value for color-based detection tasks. A Startbutton is provided to trigger tasks, and an onboard RGB LED indicates system status.
- All components are connected via 7 USB ports, ensuring modular and streamlined integration.

2. MOTOR SELECTION AND IMPLEMENTATION

2.1 TYPES OF MOTORS CONSIDERED:

For propulsion, a **BO dual shaft DC geared motor** was chosen for its simplicity, lightweight build, and compatibility with plastic mounts. For steering, a **MG995 servo motor** was selected due to its high torque, precise 180° rotation, and proven performance in robotic applications.

SELECTION CRITERIA

Torque, Speed, Power Requirements:

- BO motor provides **300 RPM** and **0.35 kg-cm** torque, suitable for smooth forward/backward motion.
- MG995 offers 10–12 kg-cm stall torque, essential for moving the steering linkage with precision.

Voltage and Current Ratings:

- BO Motor: Operates between **3–12V** with **40–180 mA** no-load current. MG995 Servo: Operates from **4.8V** to **7.2V**, optimized around 6.6V for maximum torque.

Control Interface Compatibility:

- Both motors are easily controlled using PWM signals and are fully compatible with the Autonexcontroller's dedicated motor and servo ports.

2.2 ENGINEERING CALCULATIONS:

Torque = Force × Radius:
With a wheel radius of 3.35 cm, 0.35 kg-cm translates to approx. 0.1 N of force at the wheel surface.

Power = Torque × Angular Velocity:
Approximate angular velocity: 300 RPM=31.4 rad/s
Power = 0.035 Nm × 31.4 ≈ 1.1 W

Speed = Wheel RPM × Circumference:
Circumference = π × 6.7 cm ≈ 21 cm
Max linear speed ≈ 300 × 21 cm/min ≈ 1.05 m/s

2.3 IMPLEMENTATION DETAILS:

Motor Mounting:

- BO motor is mounted with a custom 3D-printed holder that locks it in place with nuts and screws.
- Shaft ends connect to wheels with double-D female connectors and use ball bearings for support.

Motor Driver Selection:

- The Autonexcontroller includes built-in motor driver DRV8833 control logic compatible for BO motor. Servo motor has direct gpio connection to esp32 controller.

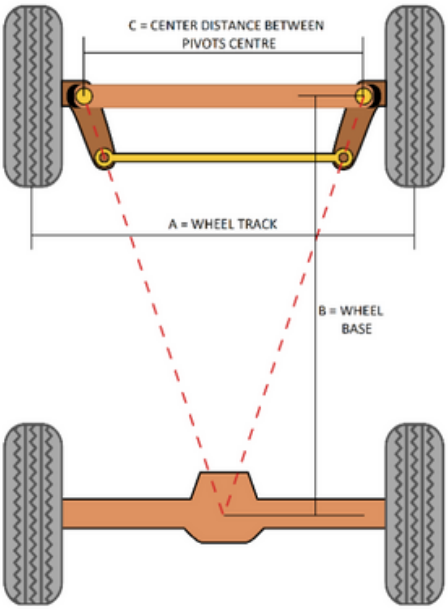
2.4 ACKERMANN STEERING SYSTEM:

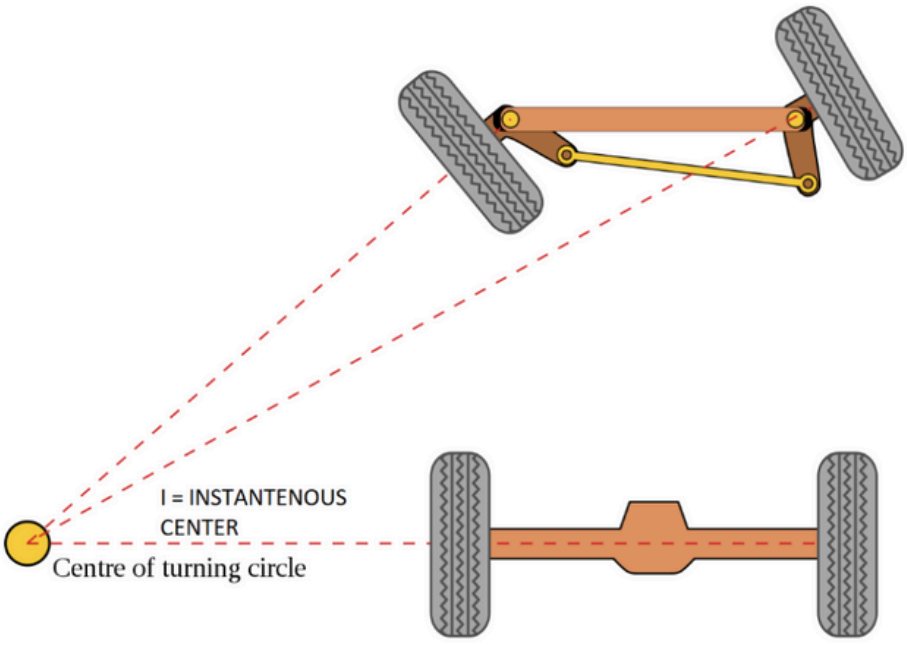
OBJECTIVE OF ACKERMAN STEERING MECHANISM:

Ackerman steering geometry is most widely used in commercial vehicles. The intention of Ackermann geometry is to avoid the need for tires to slip sideways, when following the path around a curve.

ACKERMAN PRINCIPLE:

During the turning the vehicle if I-centers of all wheels meet at a point, then the vehicle will take turn about that point which results in pure rolling action of the vehicle. The condition is called the Ackermann condition and this principle is known as Ackermann principle.





LETS CALCULATE THE ACKERMANN ANGLE, INNER & OUTER STEERING ANGLE:

Distance between pivot centers $c = 91\,\text{mm} = 0.091\,\text{m}$

Wheelbase $b = 160\,\text{mm} = 0.160\,\text{m}$

Formula for Ackermann (Steer) Angle α:

$$\alpha = \tan^{-1}\left(\frac{c}{2b}\right)$$

Substitute the values:

$$\alpha = \tan^{-1}\left(\frac{0.091}{2 \times 0.160}\right)$$

$$= \tan^{-1}\left(\frac{0.091}{0.320}\right) = \tan^{-1}(0.284375)$$

Calculate using calculator:

$$\alpha \approx \tan^{-1}(0.284375) = 15.86^\circ$$

Final Result:

$$\boxed{\text{Steer Angle or Ackermann Angle } \alpha = 15.86^\circ}$$

To calculate the **inner and outer steering angles** in an Ackermann steering setup, we'll use the **geometry of the turning circle**. You provided:

- **Common steering servo angle** (middle position) = **20°**
- **Wheelbase (L) = 160 mm**
- **Track width (T) = 156 mm**

We assume the **servo controls the center link** that represents the average or virtual center. You want to compute what the correct inner and outer angles should be.

1. Estimate the turning radius R using center angle (θ):

$$R = \frac{L}{\tan(\theta)}$$

Let's convert **20° to radians**:

$$\theta = 20^\circ = 0.3491 \text{ rad}$$

Now calculate R:

$$R = \frac{160}{\tan(0.3491)} \approx \frac{160}{0.364} \approx 439.6 \text{ mm}$$

This is the radius to the **center of the vehicle** during the turn.

2. Calculate Inner and Outer Wheel Turning Radii:

- The front wheels are **T = 156 mm** apart.
- So:
  - **Inner turning radius** = $R - \frac{T}{2} = 439.6 - 78 = 361.6$ mm
  - **Outer turning radius** = $R + \frac{T}{2} = 439.6 + 78 = 517.6$ mm

3. Calculate Inner and Outer Steering Angles:

Using:

$$\theta = \tan^{-1}\left(\frac{L}{R}\right)$$

- **Inner angle θ_i:**

$$\theta_i = \tan^{-1}\left(\frac{160}{361.6}\right) \approx \tan^{-1}(0.4423) \approx \boxed{23.84^\circ}$$

- **Outer angle θ_o:**

$$\theta_o = \tan^{-1}\left(\frac{160}{517.6}\right) \approx \tan^{-1}(0.309) \approx \boxed{17.14^\circ}$$

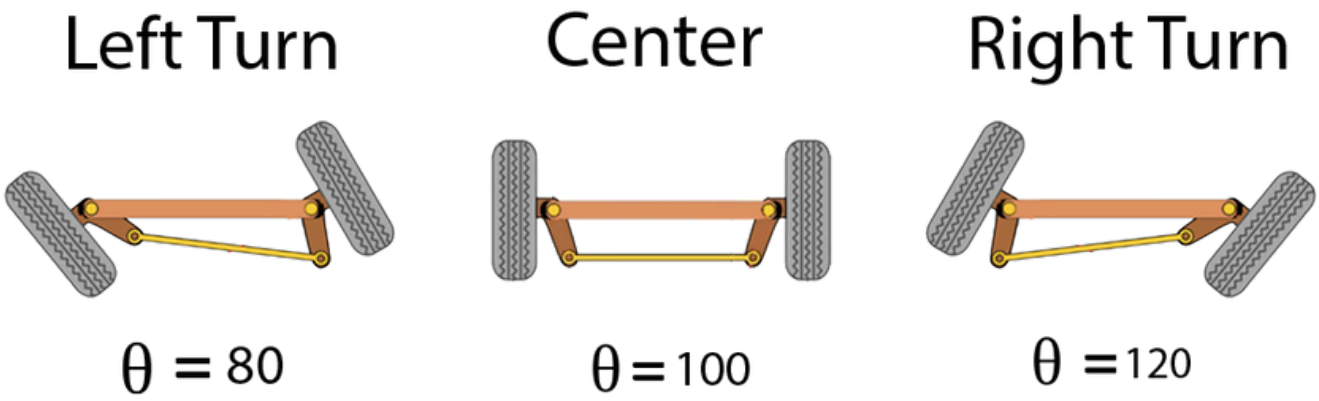
Turning Angles for Autonex Bot:

1. **F**or the Autonex Bot, the ideal turning angle was theorized and tested to be between **15°** and **20°**.

2. We chose a constant turning angle of 20° for consistent performance.

3. In the Autonex Bot, the servo's default **center position** is fixed at **100°**.

- For a **left turn**: center − 20° = **80°**
- For a **right turn**: center + 20° = **120°**



3. CHASSIS DESIGN AND COMPONENT MOUNTING

3.1 CHASSIS DESIGN OVERVIEW:

The vehicle is built on a **laser-cut 4mm thick acrylic base** which is durable, lightweight, and easy to modify. It serves as the main platform to mount the motor holder, controller, mobile phone holder and sensor holders. The layout is designed for functional symmetry and stability during movement.

3.2 STRUCTURAL LAYOUT AND DIMENSIONS:

The chassis is arranged to support a **rear-mounted BO dual shaft motor** for propulsion and a **front-mounted MG995 servo motor** for steering via an Ackermann mechanism. The **mobile holder** is **centrally mounted** on the chassis, ensuring that the phone's weight is evenly distributed. SLA 3D-printed mounts are placed at all sides for ultrasonic sensors, allowing for full 360° environmental awareness.

3.3 MATERIAL SELECTION:

Acrylic was selected for its excellent strength-to-weight ratio and availability, and its 4mm thickness offers sufficient rigidity while maintaining a low profile. A total of **11 SLA 3D-printed** parts are used for functional mounts, such as motor holders, sensor brackets, bearing supports, and structural reinforcements, optimizing the mechanical fit and performance.

3.4 MOUNTING CONSIDERATIONS:

Motor and Wheel Positioning:

- The **BO motor** is securely mounted at the rear, transferring motion to both rear wheels using double-D shaft connections and supported by ball bearings.
- The **MG995 servo** is mounted at the front, enabling left/right control of the front wheels via custom 3D-printed steering linkages.

Battery and Electronics Layout:

- A single **18650 lithium-ion** battery (3200 mAh) is housed within the Autonexcontroller, simplifying power distribution and centralizing weight.

Load Distribution and Center of Gravity:

- The **total weight** of the bot is **975g**, and **1120g** with a mobile phone.
- The central placement of the mobile holder ensures balanced mass distribution, minimizing instability or tipping during motion.

Stability and Structural Integrity:

- The design minimizes vibration and structural flex.
- Ball bearings on both motor shafts and 3D-printed reinforcements improve durability and reduce mechanical wear.

4. ENGINEERING PRINCIPLES IN DESIGN

4.1 APPLICATION OF PHYSICS:

Newton's Laws:

Newton's Second Law (F = ma) was used to guide motor selection. The BO motor delivers enough torque to overcome inertia and start moving a vehicle mass of over 1.1 kg.

Speed vs. Torque Trade-offs:

- The selected BO motor provides 300 RPM with 0.35 kg-cm of torque, which is sufficient for movement due to the low rolling resistance and optimized weight distribution.
- Higher torque would be needed for inclined surfaces, but the current setup is optimized for flat terrain.

Power Efficiency and Thermal Management:

- The motors operate within their rated voltage range (3–12V for BO, 4.8–7.2V for MG995), reducing thermal strain.
- The open acrylic chassis supports passive cooling, avoiding heat buildup.

4.2 PERFORMANCE CALCULATIONS:

Estimated Speed and Acceleration:

Wheel diameter = 6.7 cm → Circumference ≈ 21 cm
Max speed = 300 RPM × 21 cm = 1.05 m/s (under no load)
Acceleration depends on total weight (1120 g) and motor torque, expected to be modest but sufficient for task-based movement.

Load Capacity and Stability:

- The system is designed to handle the full 1120 g weight, including the mobile phone, without overloading the motors.
- MG995 servo's torque (10–12 kg-cm) is more than adequate for steering the front wheels under full load.

Ground Clearance and Turning Radius:

- Ground clearance is 0.8 cm, keeping the center of gravity low for increased stability.
- Ackermann steering offers a tight turning radius, making the bot ideal for navigating constrained environments like hallways or obstacle courses.

5. ASSEMBLY GUIDE AND CAD RESOURCES

5.1 TOOLS AND MATERIALS REQUIRED:

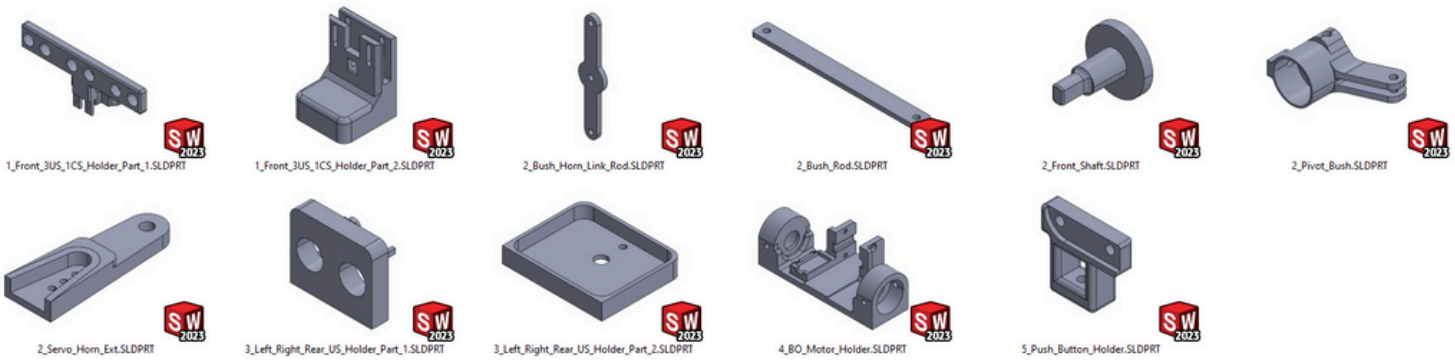
- Screwdriver.
- M3 and M4 screws, nuts and nut driver.
- Needle Nose Plier 5inch.

5.2 STEP-BY-STEP BUILD INSTRUCTIONS:

1. Secure the AutonexController to the chassis using M4 nuts. Ensure the nuts are tightly locked to prevent any movement.
2. Attach the front ultrasonic sensor holder to the chassis by fastening it with M3 nuts. Confirm that it is aligned correctly and firmly fixed.
3. Install the mobile holder in the central hole of the chassis. Make sure it is properly positioned and secured.
4. Mount all four wheels onto their respective shafts. Ensure each wheel is tightly fitted and rotates smoothly.

5.3 CUSTOM 3D-PRINTED COMPONENTS:

- Front 3 US and 1 CS Holder
- Left, Right and Rear ultrasonic sensor holder
- Pivot bush and Front shaft
- Bush Rod
- Bush-Horn Link Rod
- BO Motor Holder
- Start Button Holder



POWER AND SENSE MANAGEMENT

1. POWER SOURCE FOR THE VEHICLE

1.1 POWER SOURCE SELECTION:

Battery Specification:

- **Type:** 3.7V, 3200 mAh Li-ion single-cell battery
- **Advantages:** High energy density, lightweight, and rechargeable
- **Operation Time:** Approximately 40–50 minutes per full charge
- **Charging Module:** TP4056 lithium battery charging module (charging time: 1–2 hours)

Voltage Regulation:

- **MT3608 DC–DC Step-Up Converters** are used to regulate and supply appropriate voltages:
  - 3.3 V for low-power electronics (sensors, controller logic)
  - 5.0 V for BO motor, servo motor, ultrasonic sensors, and color sensor.

This ensures each component receives optimal voltage, improving performance and longevity.

1.2 SENSORS AND THEIR ROLES:

The vehicle uses multiple sensors to perceive and react to its surroundings:

SNo	Sensor	Purpose	Location	Voltage Requirement
1	Ultrasonic Sensors (Front, Rear, Left, Right, F1, F2)	Obstacle detection and distance measurement	Mounted around chassis	5 V
2	TCS34725 Color Sensor	Color recognition for path following or object sorting	Front-mounted	3.3 V
3	CVPro ESP32 Controller	Central control unit for sensor data processing and motor control	Center chassis	5.0 V
4	Pushbutton	Manual control input	On controller	3.3 V
5	RGB LED	Status indication	On controller	3.3 V
6	BO Motor	Bot Forward-Backward Movements	Mounted in chassis	5 V
7	Servo Motor	Act as Steering Motor	Mounted in chassis	5 V

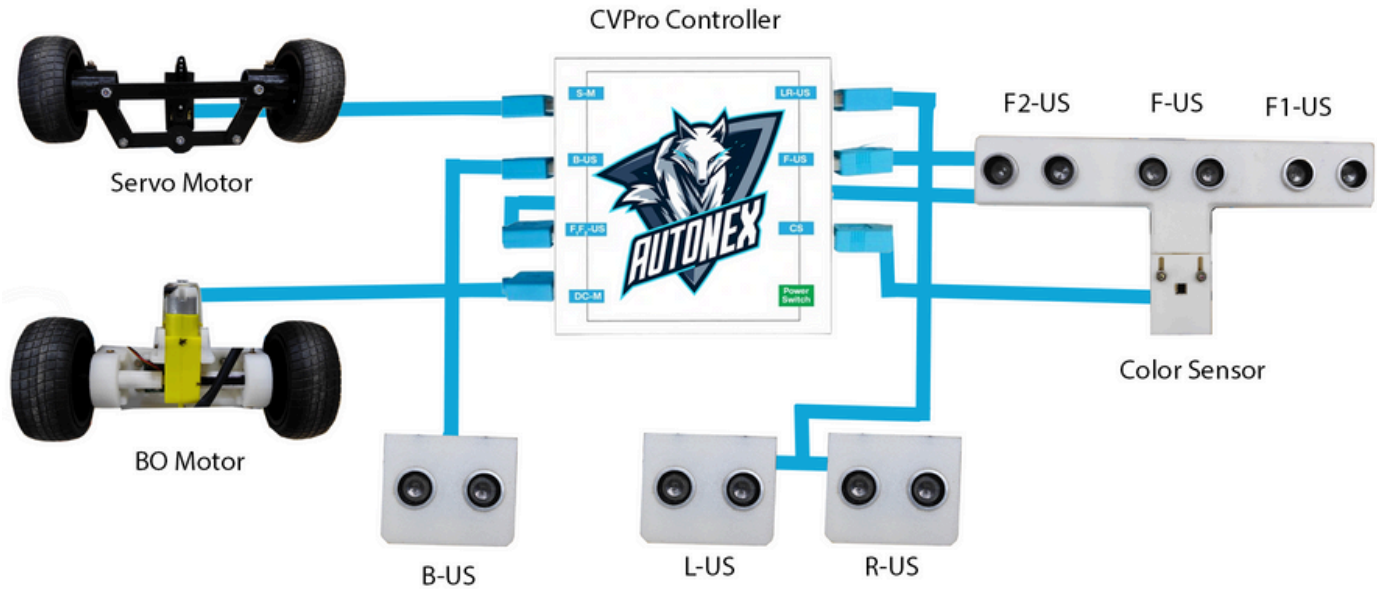
1.3 SENSOR SELECTION JUSTIFICATION:

- **Ultrasonic Sensors:** Affordable, robust in varying lighting, ideal for precise short-to-medium range obstacle detection.
- **TCS34725 Color Sensor:** Accurate color detection under varied light conditions with built-in white LED for illumination.
- **ESP32 Controller:** Handles real-time sensor data processing and motor control using its dual-core processor and integrated peripherals.
- **RGB LED & Startbutton:** Allows simple manual testing, calibration, and status feedback.

1.4 POWER CONSUMPTION CONSIDERATIONS:

- Efficient power distribution is crucial to maximize runtime.
- MT3608 converters minimize voltage loss while stepping up from 3.7 V to required levels.
- Motor current draw is the largest consumer, so motor duty cycles are optimized in software to save power.
- Sensor sampling rates are managed to balance performance and power efficiency.

1.5 WIRING DIAGRAM & COMPONENTS PROVIDED:



CVPro Controller

- Central hub for controlling motors and sensors.
- Labeled ports for each connection (S-M, DC-M, US sensors, CS).

Motors

- Servo Motor (S-M port) – Mounted on the front axle for Ackermann steering control.
- BO Motor (DC-M port) – Drives the rear wheels for forward and backward motion.

Ultrasonic Sensors (US)

- F1-US, F-US, F2-US – Three front-facing ultrasonic sensors mounted together for obstacle detection in front.
- B-US – Rear-facing ultrasonic sensor for detecting obstacles behind.
- L-US – Left-side ultrasonic sensor for side obstacle detection.
- R-US – Right-side ultrasonic sensor for side obstacle detection.

Color Sensor (CS port)

- Positioned below the front ultrasonic sensors.
- Detects surface colors for line following, object sorting, or other vision-based tasks.

Power Supply

- 18650 Li-ion battery powers the CVPro controller and connected components.
- On/Off power switch integrated into the controller for easy control.

Wiring Layout

- Blue cables represent signal/power connections from the CVPro controller to each device.
- Each sensor and motor is connected to its dedicated labeled port for organized wiring.

COMPONENTS PROVIDED:

Sno	Component	Quantity	Voltage	Notes
1	3.7V 3200 mAh Li-ion battery	1	3.7 V	Main power source
2	BO Dual Shaft Motor	1	5 V	Drive motor
3	MG995 Servo Motor	1	5 V	Steering
4	Ultrasonic Sensor	6	5 V	Distance sensing
5	TCS34725 Color Sensor	1	3.3 V	Color detection
6	CVPro ESP32 Controller	1	3.3 V	Central control

1.6 CONCLUSION:

Effective power and sense management ensures that all electronic components receive the correct voltage and that the vehicle can gather accurate environmental data for autonomous navigation. This balance between power efficiency and sensory accuracy is vital for reliable performance in real-world conditions.

OBSTACLE MANAGEMENT

1. OBSTACLE MANAGEMENT FOR THE VEHICLE

1.1 SOFTWARE FOR BOT:

Software Used:

- **Arduino IDE** – Primary development environment for writing, compiling, and uploading code to the controller.
- **AutonexAndroid Mobile Application** – Used for Object Detection and sends the data to the bot.
- **Arduino C++** – Programming language used to implement control logic, sensor integration, and obstacle management algorithms.

1.2 OBSTACLE MANAGEMENT STRATEGY FOR AUTONEX:

The Future Engineer mat consists of four main sections:

1. **Straight-Forward Section**
2. **Corner Section**
3. **Sign Box Detection**
4. **Parking Lot**

This section explains how the Autonexbot negotiates each part of the course, using its sensors, control logic, and obstacle management techniques.

1. Strategy for Straight-Forward Section:

The Autonexbot is equipped with left and right ultrasonic sensors mounted on the respective sides. The goal in this section is to move forward without colliding with any walls.

A **counter-steering strategy** is implemented:

- **Condition 1:** If the left ultrasonic sensor detects a wall at less than 30 cm, the bot is leaning left. The servo turns right to realign.
- **Condition 2:** If the right ultrasonic sensor detects a wall at less than 30 cm, the bot is leaning right. The servo turns left to realign.
- **Condition 3:** If both left and right ultrasonic readings are greater than 30 cm, the servo remains centered and the bot continues forward.

This method allows the Autonexbot to pass through the Straight-Forward Section smoothly while maintaining a safe distance from both walls.

2. Strategy for Corner Section:

After completing the Straight-Forward Section, the bot enters the Corner Section, where it must detect a colored line (orange or blue) and take a precise turn.

- **Orange Line:** Indicates a clockwise turn → The bot takes a perfect right turn.
- **Blue Line:** Indicates a counterclockwise turn → The bot takes a perfect left turn.

By following this detection-and-turn strategy, the bot transitions into the next Straight-Forward Section.

3. Strategy for Sign Box Detection:

In the obstacle challenge round, the Autonex bot is equipped with a mobile phone mounted via a phone holder. The Autonex mobile application uses an object detection model trained to detect red and green boxes.

- The mobile phone connects to the Autonex controller via a USB-C to USB-C OTG cable.
- When the app detects a **red box**, it sends value 1 to the controller → The bot avoids it by taking a right turn.
- When the app detects a **green box**, it sends value 2 to the controller → The bot avoids it by taking a left turn.

This enables the bot to dynamically avoid obstacles based on real-time color recognition.

4. Strategy for Parking Lot

At the end of the obstacle challenge (after 3 laps), the Autonex bot must park in the Parking Lot.

- The Autonex mobile application includes an object detection model to identify parking lot walls.
- When detected, the app sends value 3 to the controller.
- The bot then executes its parking sequence and positions itself accurately within the parking area.

This structured obstacle management strategy ensures the Autonex bot can navigate all challenges on the Future Engineer mat efficiently, accurately, and without collisions.

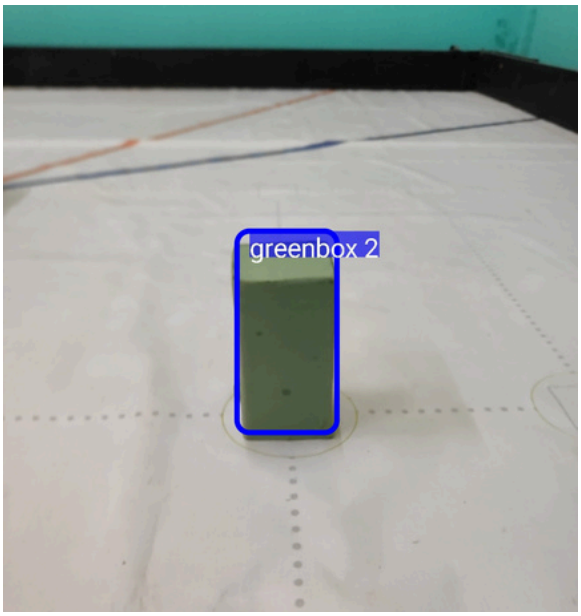
1.3 AUTONEX MOBILE APP:

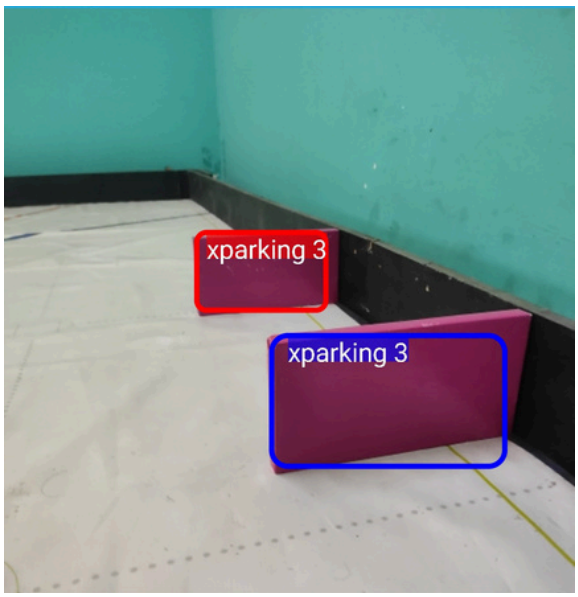
The Autonex Mobile Application is built on an object detection algorithm. In the Future Engineers category, the system uses sign boxes (red and green) as markers. At the end of the obstacle round, the bot must park in a designated parking lot. To achieve this, we trained a TensorFlow Lite (TFLite) model to detect both the sign boxes and the parking wall.

1. Red Sign Box Detection:
When the mobile application detects a red box, it sends the value "1" to the microcontroller through a C-to-C OTG cable. Based on this value, the bot executes a right turn.

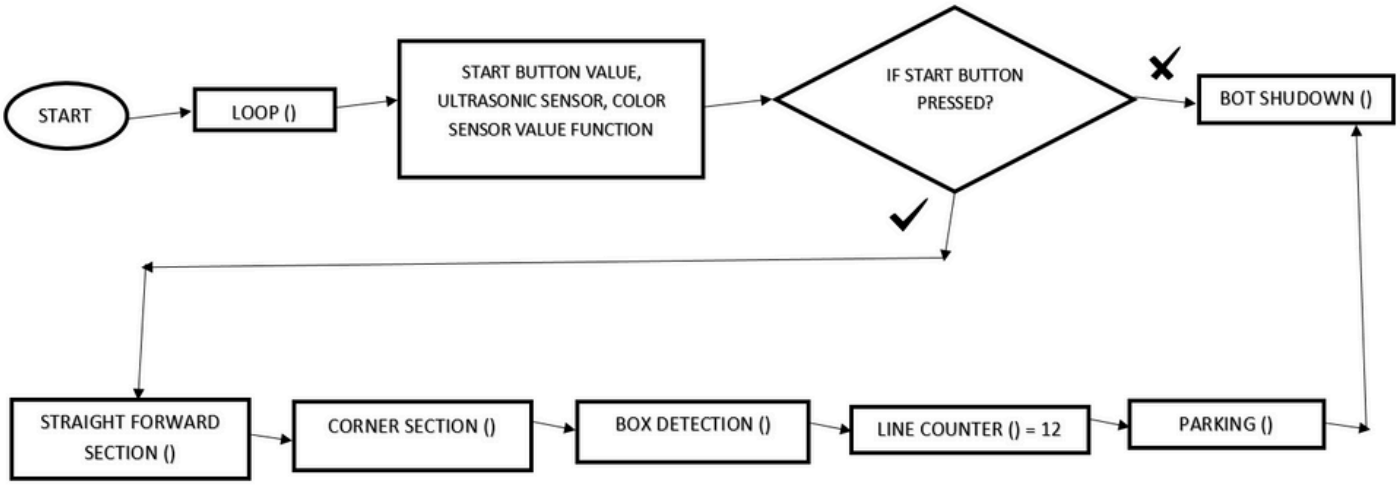
2. Green Sign Box Detection:
When the mobile application detects a green box, it sends the value "2" to the microcontroller through a C-to-C OTG cable. Based on this value, the bot executes a left turn.

3. Parking Wall Detection:
At the end of the round, when the mobile application detects the parking wall, it sends the value "3" to the microcontroller through a C-to-C OTG cable. Receiving this value triggers the bot to perform the parking maneuver.





1.4 FIRMWARE CODING LOGIC AND FLOW DIAGRAM:



Once the Autonex bot is placed on the Future Engineer (FE) Mat, the power switch is turned ON. The start button is then pressed to initiate the run.

In the firmware:

- When the Startbutton is pressed, it sets a value of 1 in the program loop.
- This value is used in a Main Gate IF control statement that acts as the entry point for all core functions and events.
- Inside the Main Gate, the program executes the vehicle's primary navigation and obstacle management routines.

1. Initial Movement and Straight-Forward Section

- The bot begins moving forward at an initial set speed.
- It navigates the Straight-Forward Section using a counter-steering strategy based on three conditions from the ultrasonic sensors (as described in the obstacle management section).

2. Corner Section Navigation

- After passing the Straight-Forward Section, the bot enters the Corner Section.
- Here, a color sensor detects the colored line (orange or blue).
- The code uses the color temperature values from the sensor to determine the correct turn direction and adjust the servo position accordingly.

3. Box Detection and Sign Response

- A box detection function continuously receives values from the mobile phone via the Autonex mobile application.
- This data is used for:
  - Sign Box Detection: Avoiding red or green boxes based on the object detection model.
  - Parking Function: Detecting parking lot walls when the final lap is reached.

This firmware structure ensures smooth gameplay logic, where the bot transitions between sections seamlessly—executing obstacle detection, path correction, and event-based actions in a controlled sequence.

The End