

## **AUTONOMOUS BUS TRANSIT SIMULATION IN THE BALI TOURISM DEVELOPMENT CORPORATION (BTDC) AREA**

**Arif Devi Dwipayana<sup>1\*</sup>, Adrian Pradana<sup>1</sup>, Riz Rifai Oktavianus Sasue<sup>1</sup>, I Wayan Adiyasa<sup>2</sup>**

<sup>1</sup>Automotive Technology Study Program, Politeknik Transportasi Darat Bali, Jalan Cempaka Putih Tabanan, Bali 82111, Indonesia

<sup>2</sup>Automotive Engineering Education Study Program, Faculty of Engineering, Universitas Negeri Yogyakarta, Jl. Colombo No.1, Karang Gayam, Caturtunggal, Sleman, Yogyakarta 55281, Indonesia

\*[arif.devi@poltradabali.ac.id](mailto:arif.devi@poltradabali.ac.id)

### **ABSTRACT**

The Bali Tourism Development Corporation (BTDC) area is a tourism area located in Nusa Dua, Bali, Indonesia and has now become one of the main tourist destinations in Bali. Adopting autonomous transit bus technology in the BTDC area will show commitment to innovation and a modern image of tourism in Bali, after the Bali Provincial Government showed its seriousness in supporting the use of electric vehicles in tourism areas. The aim of this research is to find out the working steps of the electronic system for simulating an autonomous transit bus and produce a prototype of an autonomous transit bus. The research was carried out using the research and development (R&D) method which focuses on development with the aim of expanding or exploring further a theory or its application with certain developments. The results of the research show that the working steps of the electronic system for simulating an autonomous transit bus prototype are with a control system using the tracking principle of line tracer sensors with a line follower algorithm in passenger bus type mass transportation in tourism/BTDC areas. The autonomous transit bus prototype has been successfully tested by testing motor driver, motor reaction, object detection, and trajectory tracking without experiencing any problems.

Keywords: autonomous; autonomous bus transit; BTDC

### **INTRODUCTION**

The development of research on autonomous vehicles is currently quite rapid, most of the research focuses on autonomous cars, not many are looking at autonomous vehicles which have a more environmentally friendly effect, namely autonomous buses (Nenseth, Ciccone, and Kristensen, 2019). Autonomous buses also called driverless buses or automated shuttles, are being tested with pilot projects in several cities around the world (Ainsalu et al., 2018; Nordhoff, de Winter, Madigan, et al., 2018). Initial studies on the use of autonomous buses show positive attitudes among users and feelings of safety and security during travel (Nordhoff, de Winter, Madigan, et al., 2018; Rehrl & Zankl, 2018; Salonen & Haavisto, 2019). Several existing studies have been conducted in various areas including university environments, offices, and residential areas. Insight into the use of autonomous buses in tourism areas is still lacking. Such insights are important for understanding whether and how autonomous buses can be used in a tourism context, and more specifically: how tourists perceive the use of autonomous buses and in what ways tourists can use them.

The Bali Tourism Development Corporation (BTDC) area is a tourism area located in Nusa Dua, Bali, Indonesia and has now become one of the main tourist destinations in Bali. This area has various advantages, such as a strategic location with natural views, especially beautiful and clean beaches, has many five-star hotels and resorts so it is often used for international scale convention activities, has excellent infrastructure with a very high level of security and cleanliness. well, and is often used as a venue for entertainment and cultural events. As a form of increasing the accessibility of the tourism area, it is necessary to use modern public transportation that is safe, comfortable, efficient and sophisticated. With this public transportation, tourists and local residents can easily reach and move between various important

points in the BTDC area without having to rely on private vehicles. This will facilitate mobility and help reduce traffic congestion around the area. In addition, excessive private vehicles in the BTDC area can cause increased greenhouse gas emissions and air pollution.

Today's modern public transportation, especially buses, is developing not only in terms of increasing the efficiency of the energy used, but this development is now heading towards automation or autonomy. Autonomous technology is a combination of several technologies that work together to operate automatically without a driver. The development of autonomous buses encourages technological innovation in the transportation industry. Autonomous buses can increase efficiency in transportation operations. The bus can operate continuously without the need to take breaks or be tied to the driver's schedule. In addition, autonomous buses can optimize the use of space and routes by using intelligent navigation and planning systems. This can reduce traffic congestion, reduce travel times and increase road user productivity. This is in line with the current issue, namely improving the quality of transportation services, especially mass transportation such as bus rapid transit (BRT). Adopting autonomous transit bus technology in the BTDC area will show commitment to innovation and a modern image of tourism in Bali, after the Bali Provincial Government showed its seriousness in supporting the use of electric vehicles in tourism areas. Bali is known as a unique and exciting tourist destination, and the adoption of cutting-edge technology such as autonomous buses will provide additional attraction for tourists looking for a different and modern experience. The aim of this research is to find out the working steps of the electronic system for simulating an autonomous transit bus and produce a prototype of an autonomous transit bus. Supporting this, this research will focus on creating an "Autonomous Bus Transit Simulation in the Bali Tourism Development Corporation (BTDC) Area". With this research, it is hoped that it will be able to provide an overview of the conditions for implementing autonomous technology in the land transportation system, especially in the tourism area of Bali Province.

## METHOD

The research was carried out using the research and development (R&D) method which focuses on development with the aim of expanding or exploring further a theory or its application with certain developments. According to the type of research, implementation stages can be arranged which can be seen in the flow diagram as shown in the following image.

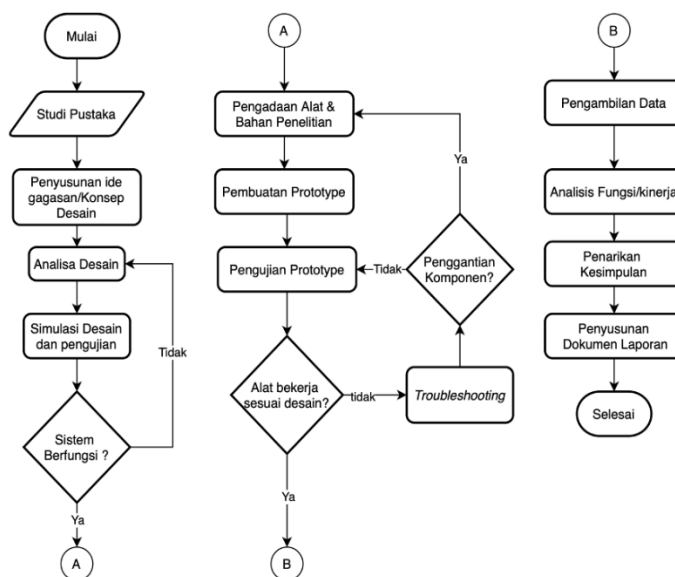


Figure 1. Research Flow Diagram

## 1. System planning

The application of autonomous vehicles for mass transportation has been carried out in various countries in the form of autonomous rail transit (ART) in the form of vehicles with a series of trains without tracks, instead lines are used to determine the direction and route of the ART whose navigation or control uses the tracking principle of line tracer sensors with algorithms. line followers. In this research, the working principle of ART navigation was adopted by applying it to passenger bus type mass transportation in the BTDC Bali area which was represented through a simulation of the Autonomous Bus Transit prototype. The designed prototype is expected to be able to detect lines as paths/routes with the additional ability to avoid objects that block the direction of movement and then return to the path when the obstacle is crossed. The prototype also has the ability to detect weather conditions, so it is possible to adjust speed dynamically based on road conditions affected by weather. (slippery road or path blocked by puddles).

## 2. Hardware Design (Hardware)

Hardware design is carried out to obtain a prototype construction form which generally includes mechanical and electronic systems which are described as follows.

### a. Construction Design

The design of the mechanical construction and chassis of the autonomous bus vehicle prototype was carried out using the help of the Sketchup 2022 computer application, so that the shape and size can be obtained as shown in the following image.

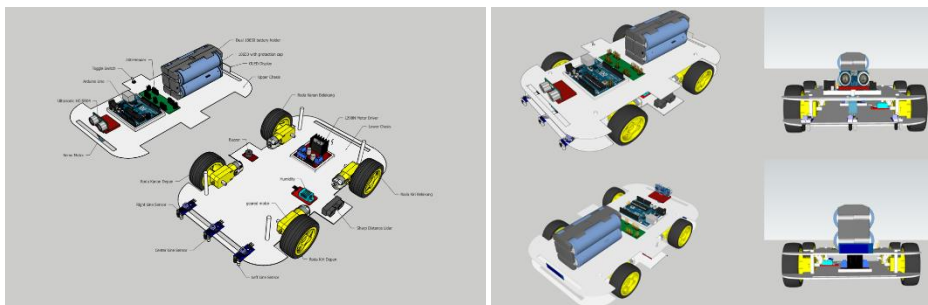


Figure 2. Lay Out Model and Full Model

### b. Electronic Systems Design

Electronic design is carried out to arrange the electronic components needed to support the mechanical system and control process via a microcontroller by referring to the following block diagram.

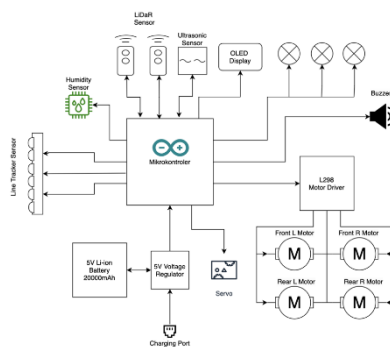


Figure 3. Electronic System Block Diagram

As can be seen in Figure 3, the autonomous transit bus prototype is composed of several sub-system parts with designs and specifications as described as follows.

1) Li-ion battery

As an electrical power source to provide supply voltage to each component, a 18650 li-ion battery with 3 serial 4.1v specifications is used to obtain 12.3 V/20,000 mAh.

2) Microcontroller

The microcontroller used is the Arduino Uno type which functions as the main controller which works based on commands/program execution to process input signals from the line tracker sensor, humidity sensor, ultrasonic sensor and LiDaR into voltage values to regulate the motor driver, servo, OLED display and buzzer. on the output side. The pin configuration used is shown in Figure 4 below.

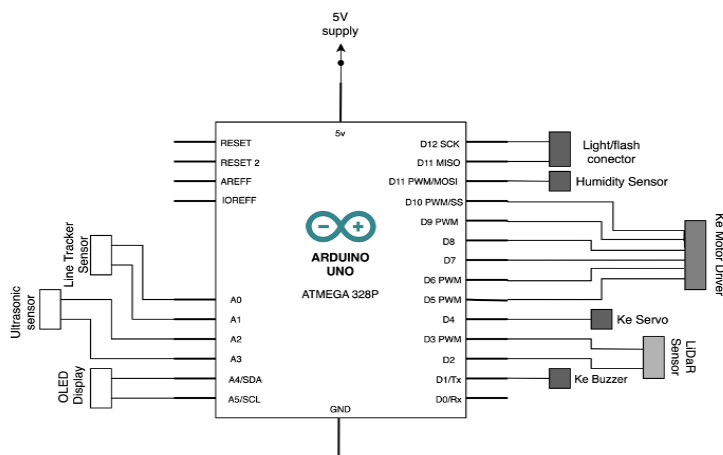


Figure 4. Arduino Uno Microcontroller Configuration

1) *Motor Driver* LN298

The motor driver circuit is used to control the rotational speed of the motor as a prototype driver. Apart from controlling the speed, the driver circuit can also control the rotational direction of the motor both clockwise and counterclockwise, thereby allowing the prototype to be moved forward or backward. The Integrated Circuit (IC) used as a driver is type LN298 which is a dual H-Bridge topology driver with operating voltage capabilities ranging from 5v to 35v and a maximum operating current of up to 2 A. This specification meets the current and voltage consumption requirements of the drive motor according to with design concepts. The driving motor is a DC motor with a gearbox with a trigger voltage of 3 SD. 6.5 V with a maximum rotation of 240 rpm. The rotation speed of the prototype is designed at 3 speed variations, namely; fast (100rpm) for straight tracks, medium (95rpm) for curved tracks and slow (85rpm) for reverse movement. The motor driver circuit configuration used can be seen in Figure 5 below.

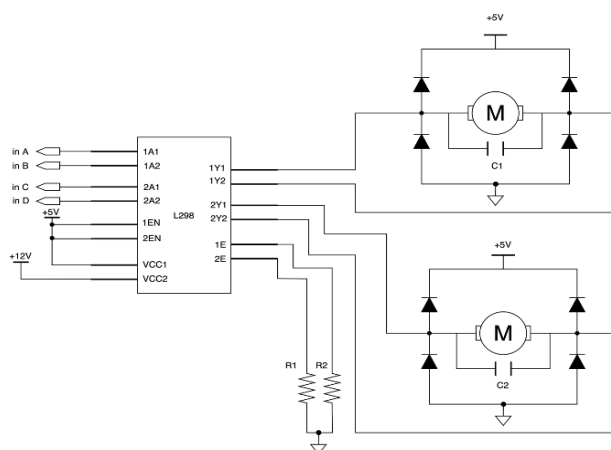


Figure 5. Motor Driver Circuit

## 2) Array Sensors

To support the performance of the prototype, four types of sensors are used, namely the IR line tracker TX5000 which is used to track paths or routes, the SRF04 ultrasonic sensor is used to detect objects in the form of obstacles, light detection and ranging (LiDaR Sensor) to detect objects in the area next to the prototype, the DHT11 sensor which is used to detect air humidity for the need to regulate the prototype's movement speed. The arrangement and configuration of all sensors is shown in Figure 6 below.

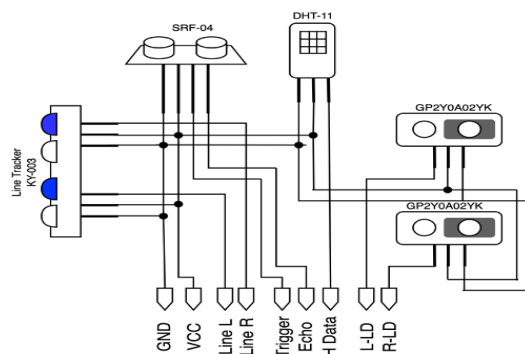


Figure 6. Sensor Array

In its function of tracing lines, the TX5000 line sensor works to detect lines/markings and every time the sensor detects a line, the sensor will send a "HIGH (I)" logic signal and vice versa a "LOW (O)" logic signal when it does not detect a line. The illustration of line tracing can be seen in Figure 7 below.

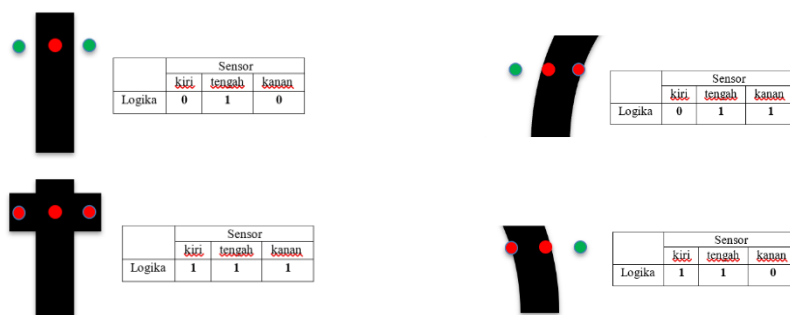


Figure 7. Illustration of Line Detection

### 3) OLED Displays and Indicator Devices

The OLED display is used as a monitor to display information on the speed and direction of motor rotation. The interface connection is connected to the SDA and SCL pins on the microcontroller as 8 bit display data communication. Apart from that, the prototype is also equipped with a buzzer as a sound indicator (horn) and light indicators for the brake/stop light and forward/reverse light with a configuration as shown in Figure 8 below.

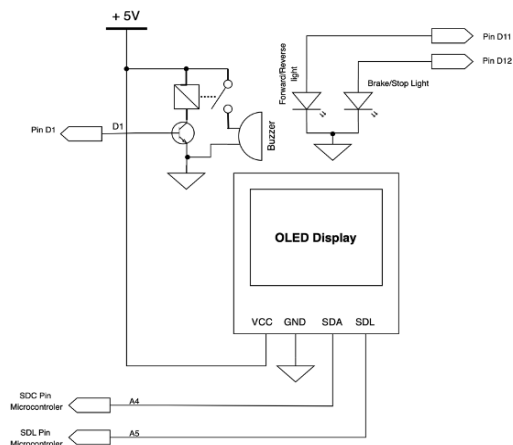


Figure 8. OLED Display and Indicator Circuit

### 3. Software Design (Software)

The program used in this research is composed of a procedure which is a sequence of sequential steps referring to the mechanism of the prototype navigation process so that a logical sequence of steps is obtained as shown in the flow diagram as follows.

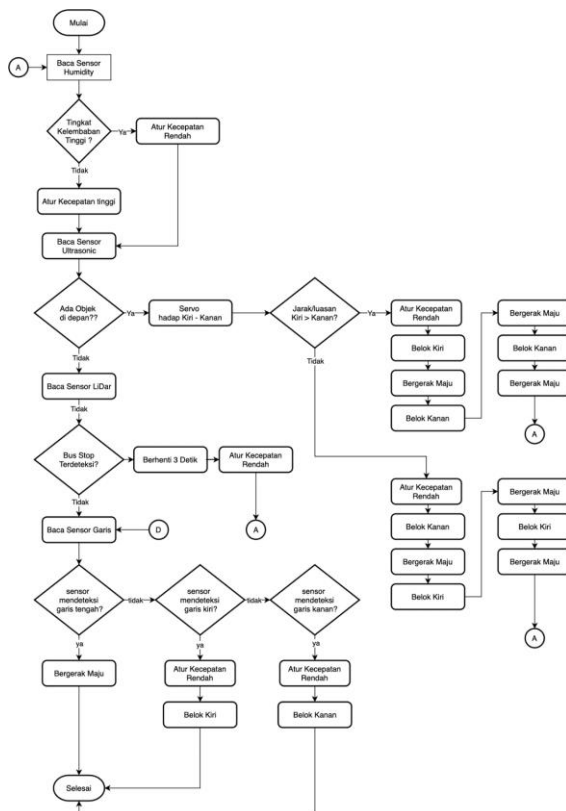


Figure 9. Program Flow Diagram

**RESULTS AND DISCUSSION**

**Route/Trajectory Determination**

Determining route location points is done by identifying land uses in the Nusa Dua ITDC area that have a high level of activity. Based on data from ITDC creating destinations, ITDC's occupancy rate will reach 68.87% in 2023 with the highest activity in hotel areas and shopping centers.

Table 1.  
 Location Points with the Highest Activity

Location Point	Land Use Management
1	Parking Lot, Club Med, Grand Whiz, Sofitel Club, Le Bar, Cinnamon Bar, Toyota Beach Club
2	(Hospitality Area) Kayu Cinnamon Private, Beach Hotel, The Westin Resort, BNI ATM, Pratama Fitness, Bali Nusa Dua, BNDCC
3	(Hospitality District) Melia Bali Hotel, The Laguna, ABBC Pacifica Museum, APVC, Devdan Show
4	Grand Hyatt, Bali Collection, Le PetitGrand Club, APVC, Devdan Show, Watersports, Peninsula Island
5	Amarterra, Nusa dua reef, Watercourt Grand Club, APVC, Devdan Show, Watersports, ITDC Mosque, IMF.
6	The Maj, Ayodya Resort, Merusaka, Novotel, Courtyard, Kagurra.
7	Indonesian Tourism Development, ITDC Mini Field, SPKLU, Central Park.
8	BIMC Hospital, ITDC Office, Fire Post,

Next, a survey is carried out on the number of people carrying out activities at the designated route location points. This aims to determine the average movement of people at that location

point so that it can determine the effectiveness of determining the route location points. The results of the survey can be seen at figure 10 following.

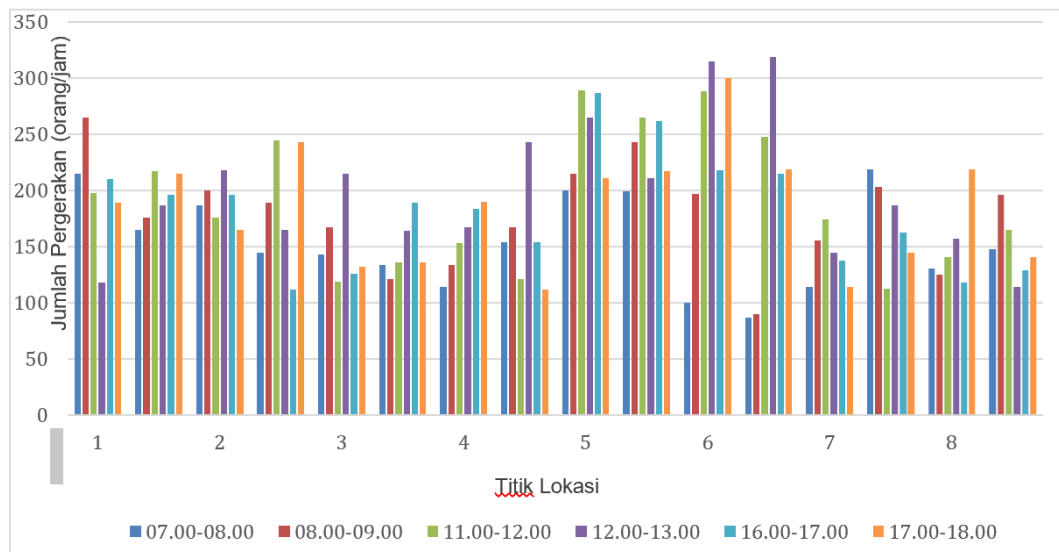


Figure 10. Graph of Number of Movements of People at Location Points

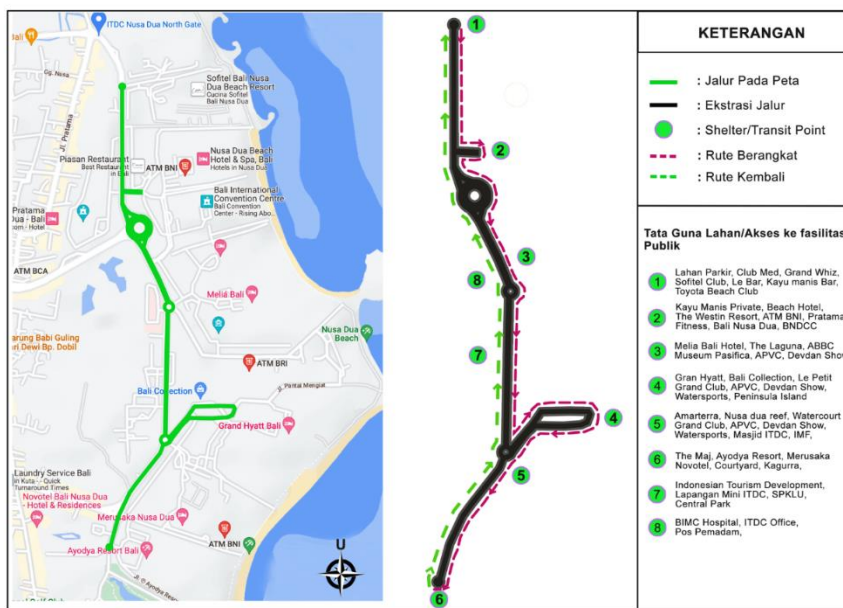


Figure 11. Route Mapping

### Track Design

To create a simulated track/track as a representation of land use in the Nusa Dua ITDC area, a track design was carried out in a form that was scaled proportionally to the autonomous bus prototype. The design was carried out using Sketch Up software. The size of the track area is 11m x 7m. To find out the bending angle, measurements are taken at each angle. The results of the design of the track shape and angle sizes are shown in Figure 12 below.

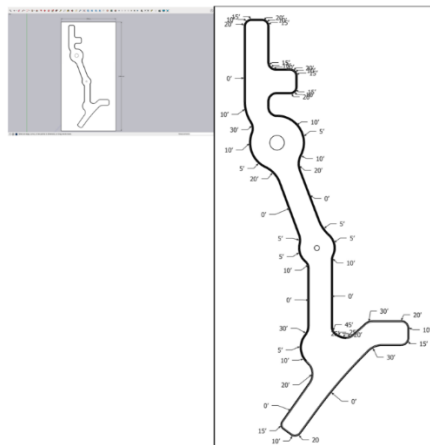


Figure 12. Simulation Trajectory

Referring to the results of the existing track design, the number and size of bending angles are calculated which are then described in the following table.

Table 2.  
 Number and size of bending angles on the track

Angle Size	Amount
0°	14
5°	7
10°	14
15°	7
20°	11
25°	2
30°	3
45°	2

### Software Design Results (Software)

By referring to the flow diagram of the control process in the design process in chapter III, a C language software program can be prepared which is written in accordance with the standard syntax in the Arduino IDE (integrated development environment) with the program listing as follows.

```

// Arduino IDE Program Listing
// (a) Declaration and Setup, (b) void Loop, (c) Void Obstacle Checking, (d) Navigation Void.

// Declaration and Setup
#define PIN_FORWARD 10
#define PIN_BACKWARD 11
#define PIN_LEFT 12
#define PIN_RIGHT 13
#define PIN_STOP 14
#define PIN_TURNLEFT 15
#define PIN_TURNRIGHT 16
#define PIN_DISTANCE 17
#define PIN_ANGLE 18
#define PIN_SENSOR_WHITE 19
#define PIN_SENSOR_BLACK 20
#define PIN_SENSOR_YELLOW 21
#define PIN_SENSOR_RED 22
#define PIN_SENSOR_GREEN 23
#define PIN_SENSOR_BLUE 24
#define PIN_SENSOR_VIOLET 25
#define PIN_SENSOR_MAGENTA 26
#define PIN_SENSOR_CYAN 27
#define PIN_SENSOR_PINK 28
#define PIN_SENSOR_GRAY 29
#define PIN_SENSOR_WHITE2 30
#define PIN_SENSOR_BLACK2 31
#define PIN_SENSOR_YELLOW2 32
#define PIN_SENSOR_RED2 33
#define PIN_SENSOR_GREEN2 34
#define PIN_SENSOR_BLUE2 35
#define PIN_SENSOR_VIOLET2 36
#define PIN_SENSOR_MAGENTA2 37
#define PIN_SENSOR_CYAN2 38
#define PIN_SENSOR_PINK2 39
#define PIN_SENSOR_GRAY2 40

// void loop
void loop() {
    // Read sensor data
    int distance = digitalRead(PIN_DISTANCE);
    int angle = digitalRead(PIN_ANGLE);
    int sensorWhite = digitalRead(PIN_SENSOR_WHITE);
    int sensorBlack = digitalRead(PIN_SENSOR_BLACK);
    int sensorYellow = digitalRead(PIN_SENSOR_YELLOW);
    int sensorRed = digitalRead(PIN_SENSOR_RED);
    int sensorGreen = digitalRead(PIN_SENSOR_GREEN);
    int sensorBlue = digitalRead(PIN_SENSOR_BLUE);
    int sensorViolet = digitalRead(PIN_SENSOR_VIOLET);
    int sensorMagenta = digitalRead(PIN_SENSOR_MAGENTA);
    int sensorCyan = digitalRead(PIN_SENSOR_CYAN);
    int sensorPink = digitalRead(PIN_SENSOR_PINK);
    int sensorGray = digitalRead(PIN_SENSOR_GRAY);
    int sensorWhite2 = digitalRead(PIN_SENSOR_WHITE2);
    int sensorBlack2 = digitalRead(PIN_SENSOR_BLACK2);
    int sensorYellow2 = digitalRead(PIN_SENSOR_YELLOW2);
    int sensorRed2 = digitalRead(PIN_SENSOR_RED2);
    int sensorGreen2 = digitalRead(PIN_SENSOR_GREEN2);
    int sensorBlue2 = digitalRead(PIN_SENSOR_BLUE2);
    int sensorViolet2 = digitalRead(PIN_SENSOR_VIOLET2);
    int sensorMagenta2 = digitalRead(PIN_SENSOR_MAGENTA2);
    int sensorCyan2 = digitalRead(PIN_SENSOR_CYAN2);
    int sensorPink2 = digitalRead(PIN_SENSOR_PINK2);
    int sensorGray2 = digitalRead(PIN_SENSOR_GRAY2);

    // Control logic
    if (distance < 10) {
        // Stop
        digitalWrite(PIN_FORWARD, LOW);
        digitalWrite(PIN_BACKWARD, LOW);
        digitalWrite(PIN_TURNLEFT, LOW);
        digitalWrite(PIN_TURNRIGHT, LOW);
        delay(100);
    } else if (angle > 45) {
        // Turn left
        digitalWrite(PIN_FORWARD, LOW);
        digitalWrite(PIN_BACKWARD, LOW);
        digitalWrite(PIN_TURNLEFT, HIGH);
        digitalWrite(PIN_TURNRIGHT, LOW);
        delay(100);
    } else if (sensorWhite == HIGH) {
        // Turn right
        digitalWrite(PIN_FORWARD, LOW);
        digitalWrite(PIN_BACKWARD, LOW);
        digitalWrite(PIN_TURNLEFT, LOW);
        digitalWrite(PIN_TURNRIGHT, HIGH);
        delay(100);
    } else if (sensorBlack == HIGH) {
        // Turn left
        digitalWrite(PIN_FORWARD, LOW);
        digitalWrite(PIN_BACKWARD, LOW);
        digitalWrite(PIN_TURNLEFT, HIGH);
        digitalWrite(PIN_TURNRIGHT, LOW);
        delay(100);
    } else if (sensorYellow == HIGH) {
        // Turn right
        digitalWrite(PIN_FORWARD, LOW);
        digitalWrite(PIN_BACKWARD, LOW);
        digitalWrite(PIN_TURNLEFT, LOW);
        digitalWrite(PIN_TURNRIGHT, HIGH);
        delay(100);
    } else if (sensorRed == HIGH) {
        // Turn left
        digitalWrite(PIN_FORWARD, LOW);
        digitalWrite(PIN_BACKWARD, LOW);
        digitalWrite(PIN_TURNLEFT, HIGH);
        digitalWrite(PIN_TURNRIGHT, LOW);
        delay(100);
    } else if (sensorGreen == HIGH) {
        // Turn right
        digitalWrite(PIN_FORWARD, LOW);
        digitalWrite(PIN_BACKWARD, LOW);
        digitalWrite(PIN_TURNLEFT, LOW);
        digitalWrite(PIN_TURNRIGHT, HIGH);
        delay(100);
    } else if (sensorBlue == HIGH) {
        // Turn left
        digitalWrite(PIN_FORWARD, LOW);
        digitalWrite(PIN_BACKWARD, LOW);
        digitalWrite(PIN_TURNLEFT, HIGH);
        digitalWrite(PIN_TURNRIGHT, LOW);
        delay(100);
    } else if (sensorViolet == HIGH) {
        // Turn right
        digitalWrite(PIN_FORWARD, LOW);
        digitalWrite(PIN_BACKWARD, LOW);
        digitalWrite(PIN_TURNLEFT, LOW);
        digitalWrite(PIN_TURNRIGHT, HIGH);
        delay(100);
    } else if (sensorMagenta == HIGH) {
        // Turn left
        digitalWrite(PIN_FORWARD, LOW);
        digitalWrite(PIN_BACKWARD, LOW);
        digitalWrite(PIN_TURNLEFT, HIGH);
        digitalWrite(PIN_TURNRIGHT, LOW);
        delay(100);
    } else if (sensorCyan == HIGH) {
        // Turn right
        digitalWrite(PIN_FORWARD, LOW);
        digitalWrite(PIN_BACKWARD, LOW);
        digitalWrite(PIN_TURNLEFT, LOW);
        digitalWrite(PIN_TURNRIGHT, HIGH);
        delay(100);
    } else if (sensorPink == HIGH) {
        // Turn left
        digitalWrite(PIN_FORWARD, LOW);
        digitalWrite(PIN_BACKWARD, LOW);
        digitalWrite(PIN_TURNLEFT, HIGH);
        digitalWrite(PIN_TURNRIGHT, LOW);
        delay(100);
    } else if (sensorGray == HIGH) {
        // Turn right
        digitalWrite(PIN_FORWARD, LOW);
        digitalWrite(PIN_BACKWARD, LOW);
        digitalWrite(PIN_TURNLEFT, LOW);
        digitalWrite(PIN_TURNRIGHT, HIGH);
        delay(100);
    } else if (sensorWhite2 == HIGH) {
        // Turn left
        digitalWrite(PIN_FORWARD, LOW);
        digitalWrite(PIN_BACKWARD, LOW);
        digitalWrite(PIN_TURNLEFT, HIGH);
        digitalWrite(PIN_TURNRIGHT, LOW);
        delay(100);
    } else if (sensorBlack2 == HIGH) {
        // Turn right
        digitalWrite(PIN_FORWARD, LOW);
        digitalWrite(PIN_BACKWARD, LOW);
        digitalWrite(PIN_TURNLEFT, LOW);
        digitalWrite(PIN_TURNRIGHT, HIGH);
        delay(100);
    } else if (sensorYellow2 == HIGH) {
        // Turn left
        digitalWrite(PIN_FORWARD, LOW);
        digitalWrite(PIN_BACKWARD, LOW);
        digitalWrite(PIN_TURNLEFT, HIGH);
        digitalWrite(PIN_TURNRIGHT, LOW);
        delay(100);
    } else if (sensorRed2 == HIGH) {
        // Turn right
        digitalWrite(PIN_FORWARD, LOW);
        digitalWrite(PIN_BACKWARD, LOW);
        digitalWrite(PIN_TURNLEFT, LOW);
        digitalWrite(PIN_TURNRIGHT, HIGH);
        delay(100);
    } else if (sensorGreen2 == HIGH) {
        // Turn left
        digitalWrite(PIN_FORWARD, LOW);
        digitalWrite(PIN_BACKWARD, LOW);
        digitalWrite(PIN_TURNLEFT, HIGH);
        digitalWrite(PIN_TURNRIGHT, LOW);
        delay(100);
    } else if (sensorBlue2 == HIGH) {
        // Turn right
        digitalWrite(PIN_FORWARD, LOW);
        digitalWrite(PIN_BACKWARD, LOW);
        digitalWrite(PIN_TURNLEFT, LOW);
        digitalWrite(PIN_TURNRIGHT, HIGH);
        delay(100);
    } else if (sensorViolet2 == HIGH) {
        // Turn left
        digitalWrite(PIN_FORWARD, LOW);
        digitalWrite(PIN_BACKWARD, LOW);
        digitalWrite(PIN_TURNLEFT, HIGH);
        digitalWrite(PIN_TURNRIGHT, LOW);
        delay(100);
    } else if (sensorMagenta2 == HIGH) {
        // Turn right
        digitalWrite(PIN_FORWARD, LOW);
        digitalWrite(PIN_BACKWARD, LOW);
        digitalWrite(PIN_TURNLEFT, LOW);
        digitalWrite(PIN_TURNRIGHT, HIGH);
        delay(100);
    } else if (sensorCyan2 == HIGH) {
        // Turn left
        digitalWrite(PIN_FORWARD, LOW);
        digitalWrite(PIN_BACKWARD, LOW);
        digitalWrite(PIN_TURNLEFT, HIGH);
        digitalWrite(PIN_TURNRIGHT, LOW);
        delay(100);
    } else if (sensorPink2 == HIGH) {
        // Turn right
        digitalWrite(PIN_FORWARD, LOW);
        digitalWrite(PIN_BACKWARD, LOW);
        digitalWrite(PIN_TURNLEFT, LOW);
        digitalWrite(PIN_TURNRIGHT, HIGH);
        delay(100);
    } else if (sensorGray2 == HIGH) {
        // Turn left
        digitalWrite(PIN_FORWARD, LOW);
        digitalWrite(PIN_BACKWARD, LOW);
        digitalWrite(PIN_TURNLEFT, HIGH);
        digitalWrite(PIN_TURNRIGHT, LOW);
        delay(100);
    } else {
        // Forward
        digitalWrite(PIN_FORWARD, HIGH);
        digitalWrite(PIN_BACKWARD, LOW);
        digitalWrite(PIN_TURNLEFT, LOW);
        digitalWrite(PIN_TURNRIGHT, LOW);
        delay(100);
    }
}
    
```

Figure 13. Arduino IDE Program Listing; (A) Declaration and Setup, (B) void Loop; (C) Void Obstacle Checking; (D) Navigation Void.

### Whole System Circuit Schematic

All the system parts which are arranged independently such as the Microcontroller, Sensor Array, Motor Driver and Oled Display are assembled into a complete overall system, so that the schematic arrangement of the overall circuit is obtained as shown in Figure 14.

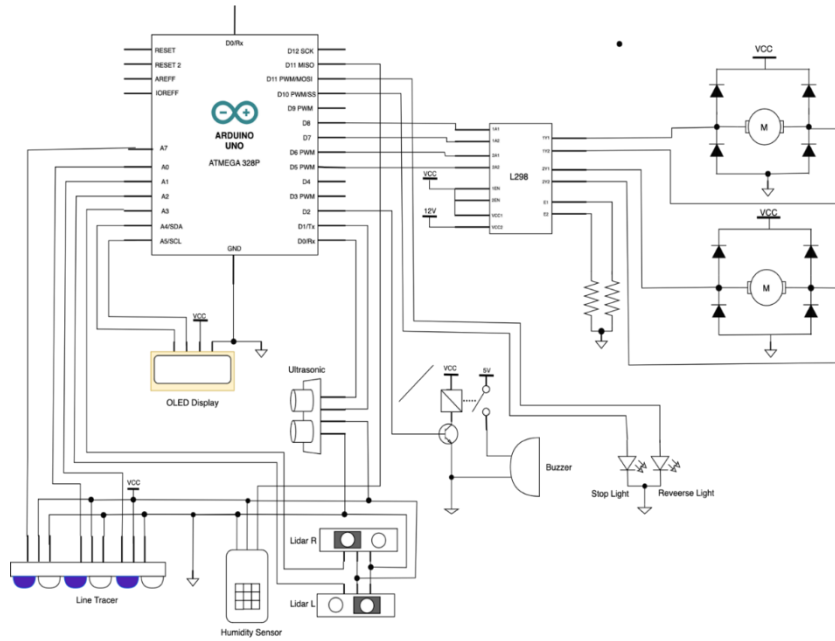


Figure 14. Schematic Diagram

The schematic image/wiring diagram above is then used as a reference in the process of installing and assembling the electronic components supporting the system so that the results of installing the hardware components for each part are obtained as shown in Figure 15 below.

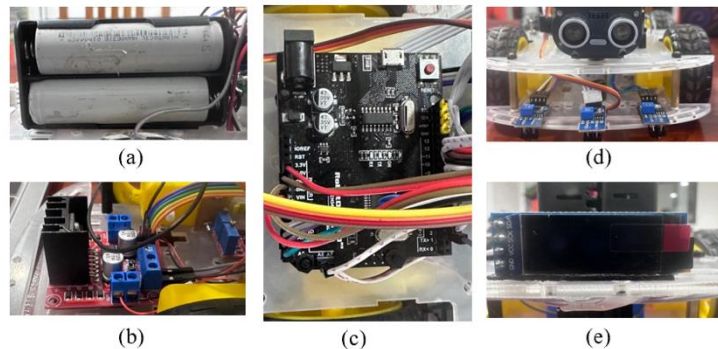
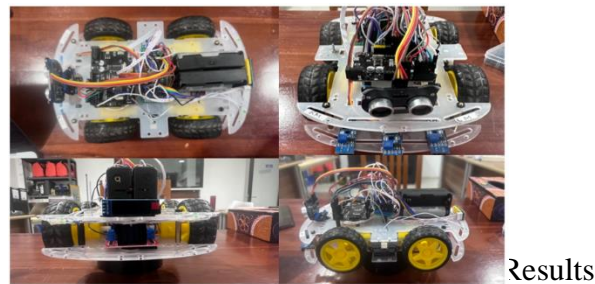


Figure 15. Installation of components for each part (sub-system); (A) Li-Ion Battery; (B) Motor Driver; (C) Controller; (D) Sensor Array; (E) OLED Display

The component parts shown in Figure 15 are then interconnected with each other and embedded in the chassis so that an arrangement of electronic components, control systems and mechanical systems is obtained to form a prototype as shown in Figure 16 below.



### Calibration and Adjustment

Before testing the prototype that has been designed, calibration and adjustment are carried out on the sensor readings, the aim is to get more accurate and precise sensor detection results. Calibration is carried out on all sensors in the sensor array module. Technical calibration is

carried out by comparing the sensor reading results with a standard that can be traced to the National standard in the form of a ruler with a resolution in millimeters (mm). The results obtained from the calibration process are shown in the following table.

Table 3.  
 Calibration Data

Sensor type	Actual size (mm)	Sensor Reading (mm)	Deviation (mm)
LiDaRGP2Y0A21YK0F	500	535	35
UltrasonicHC-SR04	500	515	15
Line Sensor TX5000	50	53	3

Through a calibration process through measurements for each type of sensor shown in the table above, the LiDaR GP2Y0A21YK0F sensor reading was obtained as an object detector worth 35 mm, for the Ultrasonic HC-SR04 sensor it was 15 mm and for the TX5000 line sensor it was 3 mm. If it is observed that the reading results of all sensors are greater than the actual value, so adjustments are made by applying a reduction operation equal to the deviation value of each sensor which is realized in the program code for each computation.

### Motor Driver Testing

The purpose of the test is to validate the success of the L298N motor driver in transforming the execution command signal from the Arduino Uno microcontroller controller with an input signal in the form of a TCX 5000 line sensor reading. The results of the command signal transformation are then converted into a voltage that is used to drive the DC motor as a prototype driver. The validation process is carried out through random treatment by placing a line in front of the sensor, then the nominal voltage at the L980 output point is measured using a volt meter. The test results can be seen in the following table.

Table 4.  
 L298N Motor Driver Measurement Data

Sensor Signals			L298 Driver Output Voltage (Volts)			
L.S	CS	RS	B2	B1	A1	A2
Left Sensor	Sensor Center	Right Sensor				
0	0	0	4.51	0	0	4.49
0	0	1	4.98	0	4.98	0
0	1	0	0	4.98	4.98	0
0	1	1	0	0	4.75	0
1	0	0	0	4.98	0	4.98
1	0	1	0	0	0	0
1	1	0	0	4.74	0	0
1	1	1	0	0	0	0

From the table above, it can be seen that the activation voltage coming out of the Motor Driver Module is in the range of 4.74V to 4.98V with an average activation voltage of 4.84V, so that it is able to meet the power requirements of the driving motor at the maximum speed of the prototype. The results of the input signal processing based on measurements can be transformed well, as shown by changes in the configuration of the voltage values 0 V and  $\approx$  4.98V which follow the sensor input configuration pattern.

### Motor Reaction Testing

The motor reaction test was carried out to see the response in the form of the direction of rotation of the motor which is used as a driver and determines the direction of movement of the prototype. During the test, changes in the rotational speed of the motor were also observed, which were expected to change according to the conditions of the prototype maneuver. The test process is carried out through random treatment by placing a line in front of the sensor, then measuring the rotation speed of the wheels on each motor. The measurement results data can be seen in the following table.

Table 5.  
DC Motor Reaction Measurement Data

Sensor Signals			DC Motor Reaction				Rotating Speed (rpm)*			
L.S	CS	RS	Left		Right		Left Motor		Right Motor	
Left Sensor	Sensor Center	Right Sensor	Front	Back	Front	Behind	Front	Behind	Front	Behind
0	0	0	<i>Back off</i>	<i>Back off</i>	<i>Back off</i>	<i>Back off</i>	-85	-85	-85	-85
0	0	1	<i>Proceed</i>	<i>Proceed</i>	<i>Back off</i>	<i>Back off</i>	95	95	-95	-95
0	1	0	<i>Proceed</i>	<i>Proceed</i>	<i>Proceed</i>	<i>Proceed</i>	100	100	100	100
0	1	1	<i>Proceed</i>	<i>Proceed</i>	<i>Shut up</i>	<i>Shut up</i>	95	95	0	0
1	0	0	<i>Back off</i>	<i>Back off</i>	<i>Proceed</i>	<i>Proceed</i>	-100	-100	100	100
1	0	1	<i>Shut up</i>	<i>Shut up</i>	<i>Shut up</i>	<i>Shut up</i>	0	0	0	0
1	1	0	<i>Shut up</i>	<i>Shut up</i>	<i>Proceed</i>	<i>Proceed</i>	0	0	95	95
1	1	1	<i>Shut up</i>	<i>Shut up</i>	<i>Shut up</i>	<i>Shut up</i>	0	0	0	0

From the measurement table above, it can be seen that the DC motor responds with a rotational movement reaction in accordance with the design, where when the left sensor (LS) detects the presence of a line by issuing a "high" logic (1), the front and rear left motors will move. reverse while the right front and rear motorbikes will move forward and will form a navigation pattern turning to the left. When the right sensor (RS) detects a line by issuing a "high" logic (1), the right front and rear motors will move backwards, while the left front and rear motors will move forward and form a navigation pattern turning to the right. The same principle applies proportionally to navigation patterns moving forward and backward. Table 5 shows the change in speed when turning, where the motor rotation speed is 95rpm, when going forward 100rpm and going backwards 85rpm. The characteristics shown in the data in the table show the success of the DC motor reaction in translating command signals from the microcontroller controller and the L298N Motor Driver.

### Object Detection Testing

Testing was carried out on a dummy track by placing random objects on the track as obstacles, while shuttle detection was carried out by placing an object to represent a bus stop facility on the left side of the track at a certain point, then looking at the prototype's movement response. Sample observations of obstacle object detection were taken by creating several obstacle scenarios with the observation results shown in table 6. Observation data for bus stop detection can be seen in table 7.

Table 6.  
Obstacle Detection Observation Data

Type	Obstacle Distance (cm)			Description	Prototype Response
	Left	Front	Right		
Scenario 1	40	30	25	There is an object in front of the track at the threshold distance (30cm), the left side has a wider maneuvering area (40)isbanding right side (25)	Stopping is continued by scanning the left and right sides, turning left, moving forward then turning right then moving forward, turning right then forward and returning to the track.
Scenario 2	25	> 100	25	there are no objects in front of the track, the left and right sides have the same maneuvering area (25)	don't stop and keep moving forward
Scenario 3	35	30	50	There is an object in front of the track at the threshold distance (30cm), the right side has a wider maneuvering area (50)isbanding right side (35)	Stopping is continued by scanning the left and right sides, turning right, moving forward then turning left then moving forward, turning left then forward and returning to the track.
Scenario 4	45	30	45	There is an object in front of the track at the threshold distance (30cm), the left and right sides have the same maneuvering area (45)	Move backwards 20 cm then scan the left and right sides

Table 7.  
 Shelter/Bus Stop Detection Observation Data

Shelter Distance (cm)	Prototype Response
5	Keep moving past objects as a shuttle bus
10	Keep moving past objects as a shuttle bus
15	Keep moving past objects as a shuttle bus
20	Keep moving past objects as a shuttle bus
25	Stop for 3 seconds (assuming passenger boarding and alighting time)
30	Keep moving past objects as a shuttle bus
35	Keep moving past objects as a shuttle bus

### Trajectory Tracking Testing

To see the success of the prototype in detecting objects blocking the track, then avoiding the object and successfully returning to the track, as well as detecting the shuttle bus/bus stop, an object detection test was carried out. Testing was carried out on a dummy track by placing random objects on the track as obstacles, while shuttle detection was carried out by placing an object to represent a bus stop facility on the left side of the track at a certain point, then looking at the prototype's movement response. Through observations it can be seen that the prototype is able to detect objects well for obstacle detection in various scenarios and even succeeds in finding or returning a path when the prototype moves to avoid obstacles and move away from the track. In the shuttle/bus stop detection, the prototype also succeeded in determining the stopping point according to the detection distance programmed on the controller, namely 25cm, then stopped for 3 seconds (assumed time) to board/alight passengers according to the table and graph below..

Table 8.  
 Trajectory Tracking Test Observation Data

Track Bend Angle	Sensor Signals			Reaction				Reaction Angle	Angular Deviation	Tracking Accuracy (%)	Navigation Direction	OLED Display Designation
	L.S	CS	RS	Left DC Motor		Right DC Motor						
	Left Sensor	Sensor Center	Right Sensor	Front	Back	Front	Behind					
0°	0	1	0	Proceed	Proceed	Proceed	Proceed	0°	0°	100.00	Straight	Forward
5°	1	1	0	Shut up	Shut up	Proceed	Proceed	6°	1°	83.33	Turn left	Turn Left
10°	0	1	1	Proceed	Proceed	Shut up	Shut up	12°	2°	83.33	Turn right	Turn Right
15°	1	1	0	Shut up	Shut up	Proceed	Proceed	17°	2°	88.24	Turn left	Turn Left
20°	1	1	0	Shut up	Shut up	Proceed	Proceed	23°	3°	95.24	Turn left	Turn Left
25°	0	1	1	Proceed	Proceed	Shut up	Shut up	27°	2°	92.59	Turn right	Turn Right
30°	1	0	0	Back off	Back off	Proceed	Proceed	33°	3°	93.75	Turn right	Turn Right
45°	0	0	1	Proceed	Proceed	Back off	Back off	48°	3°	93.75	Turn left	Turn Left

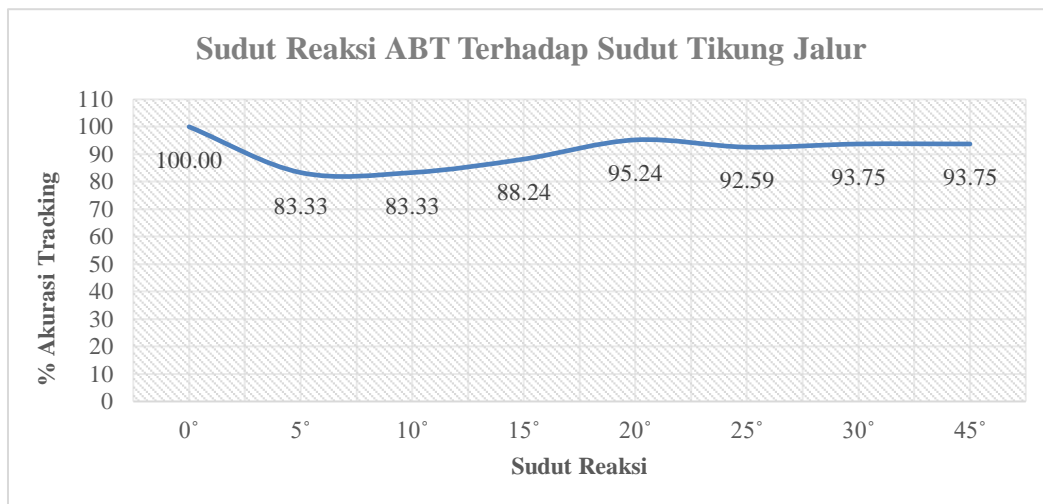


Figure 17 Prototype Reaction Accuracy Graph

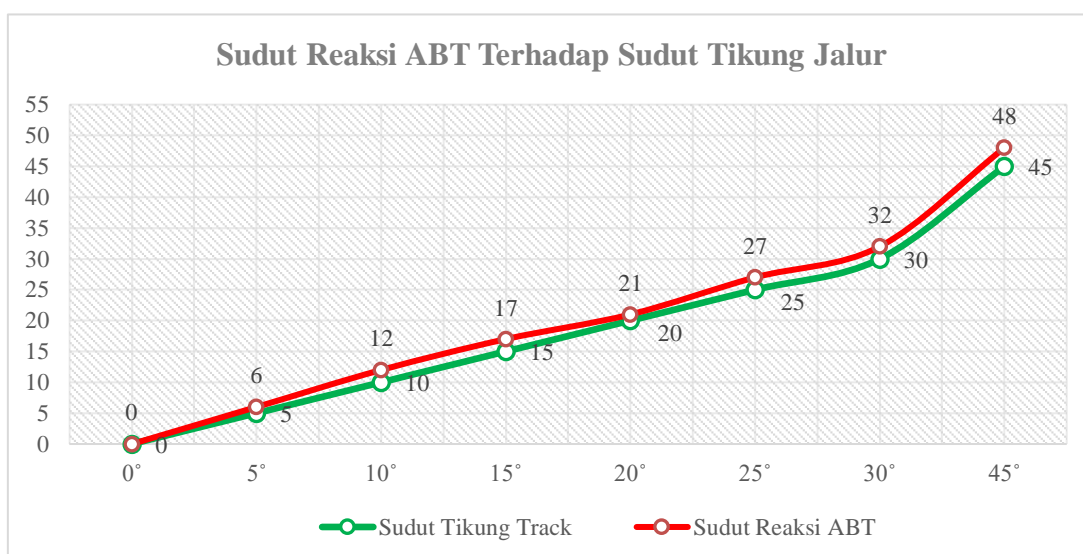


Figure 18 Graph of the Relationship Between the Reaction Angle of the Prototype and the Bending Angle of the Track

## CONCLUSION

Based on the results of the research that has been carried out, the research team can draw several conclusions as follows: 1) The working steps of the electronic system for simulating the autonomous transit bus prototype are with a control system using the tracking principle of the TX5000 line tracer sensor with the line follower algorithm in mass transportation types of passenger buses in the area tourism/BTDC; 2) The autonomous transit bus prototype has been successfully tested by testing the motor driver, motor reaction, object detection, and trajectory tracking without experiencing any problems. The suggestions that can be developed are: 1) Development of the prototype concept to the prototype stage by adding several hardware components, such as battery capacity; 2) When the simulation detects weather conditions, it should be carried out in real conditions, to test the capabilities of the system from the resulting prototype.

## REFERENCES

- Akif Fidanoglu, Ilgin Gokasar, Muhammet Deveci. 2023. Integrating shared autonomous vehicles in Last-Mile public transportation. *Sustainable Energy Technologies and Assessments* 57 (2023) 103214. journal homepage: [www.elsevier.com/locate/seta](http://www.elsevier.com/locate/seta)
- Agnihotri, N. (n.d.). GPS : Working , Applications and History. Retrieved february 2023, from Engineers Garage: <https://www.engineersgarage.com/gps-working-applications-and-history/>
- Ainsalu, J., Arffman, V., Bellone, M., Ellner, M., Haapamäki, T., Haavisto, N., ... Åman, M. (2018). State of the art of automated buses. *Sustainability*, 10(9).<https://doi.org/10.3390/su10093118>.
- Arduino.cc. (2023, March 09). Arduino Docs: Arduino.cc. Retrieved from Arduino Web site: <https://docs.arduino.cc/>
- Bayle, J. (2013). *C Programming for Arduino: Learn how to program and use Arduino boards with a series of enga*. Birmingham, United Kingdom: Packt Publishing.Ltd
- Christos Karolemeas, Stefanos Tsigdinos, Dr. Efthimios Bakogiannis, Alexandros Nikitas. (2022). Evaluating the suitability of urban road networks to facilitate autonomous buses. *Transportation Research Procedia* 62 (2022) 599–606.
- Carter, C. (1997). *Principles of GPS: A Brief Primer on the Operation of the Global Positioning System*. West Lake, CA: ALLEN OSBORNE ASSOCIATES
- Enge, P. K. (1994). *The Global Positioning System: Signals, Measurements, and Performance*. international Journal of Wireless Information Networks, 83-105.
- François Poinsignon, Lei Chen, Sida Jiang, Kun Gao, Hugo Badia, Erik Jenelius. 2022. Autonomous vehicle fleets for public transport: scenarios and comparisons. *Green Energy and Intelligent Transportation* 1 (2022) 100019. journal homepage: [www.journals.elsevier.com/green-energy-and-intelligent-transportation](http://www.journals.elsevier.com/green-energy-and-intelligent-transportation)
- French, G. T. (1996). *Understanding The GPS : An Introduction To The Global Positioning System, What it is, And How It Works*. Maryland (MD, US): GeoResearch.inc
- Gregor Leich, Joschka Bischo. (2019). Should autonomous shared taxis replace buses? A simulation study. *Transportation Research Procedia* 41 (2019) 450–460. Available online at [www.sciencedirect.com](http://www.sciencedirect.com)
- Kostas Mouratidis, Victoria Cobeña Serrano. (2021). Autonomous buses: Intentions to use, passenger experiences, and suggestions for improvement. *Transportation Research Part F* 76 (2021) 321–335. journal homepage: [www.elsevier.com/locate/trf](http://www.elsevier.com/locate/trf)
- Kaplan, E. D., & Hegarty, C. J. (2006). *Understanding GPS Principles and Applications*. Second Edition. London: Artech House.inc.
- Nenseth, V., Ciccone, A., & Kristensen, N. B. (2019). Societal consequences of automated vehicles: Norwegian scenarios (8248022374). Retrieved from Oslo: <https://www.toi.no/getfile.php?mmfileid=50576>

- Nordhoff, S., De Winter, J., Kyriakidis, M., Van Arem, B., & Happee, R. (2018). Acceptance of driverless vehicles: Results from a large cross-national questionnaire study. *Journal of Advanced Transportation*, 2018, 5382192.
- S. Tsigdinos, C. Karolemea, E. Bakogiannis, A. Nikitas. 2021. Introducing autonomous buses into street functional classification systems: An exploratory spatial approach. *Case Studies on Transport Policy* 9 (2021) 813–822. journal homepage: [www.elsevier.com/locate/cstp](http://www.elsevier.com/locate/cstp)
- Smith, A. G. (2011). *Introduction to Arduino; A piece of cake!* CreateSpace
- Limpert, R. (1999). *Brake Design and Safety 2nd Edition*. Warrendale: Society of Automotive Engineers. Inc.
- VKELElectronics. (2020). VK2828U7G5LF Datasheet. Hongkong: V.KEL Communications Equipment
- Ryosuke Abe. (2019). Introducing autonomous buses and taxis: Quantifying the potential benefits in Japanese transportation systems. *Transportation Research Part A* 126 (2019) 94–113. journal homepage: [www.elsevier.com/locate/tra](http://www.elsevier.com/locate/tra)
- Zaid Saeed Khan, Weili He, Mónica Menéndez. (2023). Application of modular vehicle technology to mitigate bus bunching. *Transportation Research Part C* 146 (2023) 103953. journal homepage: [www.elsevier.com/locate/trc](http://www.elsevier.com/locate/trc)