

Design and Control of an Arduino-Based Multifunctional Robotic Car Using Smartphone Applications

Muhammad Junaid Khan^{1,2,*} and Rida Batool Sheraliyat^{1,3}

¹*Department of Electronics, Quaid-i-Azam University, Islamabad 45320, Pakistan*

²*Department of Physics, Kohat University of Science and Technology, Kohat, Pakistan*

³*Department of Physics, COMSATS University Islamabad, 45550, Pakistan*

(Dated: December 24, 2025)

The present work focuses on a multifunctional Arduino-based smart robotic car capable of a range of functionalities within the category of advanced control, automation, and interactivity. Wireless communication is achieved by Bluetooth, voice control through the MIT App Inventor interface, obstacle detection using ultrasonic and infrared sensors, and manual operation through a remote controller and smartphone application. The vehicle is driven by DC (BO) geared motors, controlled by an L298N motor driver connected to an Arduino UNO microcontroller. In this context, wireless communication is enabled by the use of an HC-05 Bluetooth module that allows both manual and voice-commanded navigation. The developed system with an HC-SR04 ultrasonic sensor combined with IR sensors offers obstacle avoidance capability with reliable environmental awareness. The robotic platform provides line-following and obstacle-avoiding features while remaining IR remote controllable. In this work, we demonstrate a seamless integration of hardware and software, resulting in a versatile platform for educational, research, and hobbyist applications in robotics and IoT.

Keywords: Arduino UNO, Obstacle Avoidance, Infrared (IR) Sensors, Embedded Systems

I. INTRODUCTION

A robot is an intelligent machine, which is controlled through a computer program for different tasks, thus providing a set of services. The use of a robot is a crucial aspect in the automation of many [1], including industry, military, construction, health [2], and other fields. They are widely used in accomplishing the kind of repetitive, dangerous, and precise tasks which humans find difficult or dangerous to carry out [3]. In order to address the shortcomings associated with the traditional remote control system, a number of different methods have been offered. These include the control of the robot through the application of the Bluetooth-based Android application [4, 5], voice recognition [6–8], line sensing method [9–11], IR remote control method [12, 13]. The implementation of the ultrasonic sensor has been offered in the project to make the robot detect and avoid obstacles. Robots carry out their tasks faster compared to humans. Additionally, they improve the productivity and efficiency of tasks, apart from eliminating human errors. Furthermore, robots lower the expenditure of resources, save energy, and save operational and maintenance costs. Moreover, due to the increased efficiency and effectiveness associated with robots, their application has grown significantly within various sectors [14, 15].

Robots can be generally categorized into two types. Type one is Autonomous robots [16], where the tasks performed are carried out on their own using their intelligence and sensors as aids. Type two is Remote-controlled robots [17], where control is exerted from a distant location using a control device that is operated by a human being. These types are mainly applied where there is an Exposure present like bomb diffusing, space exploration,

and other activities below water.

Embedded systems, wireless communications, and intelligent control systems integration has led to the evolution of sophisticated robotic platforms that can perform intricate tasks autonomously as well as under human control. Among various smart robotic platforms, smart robotic vehicles have been found to be equally useful in automation, surveillance, delivery systems, as well as in educational settings. Because they are adaptive, modular, as well as relatively inexpensive, they are considered to be the best platforms to work with. In this paper, the development of a smart robotic vehicle, using Arduino, featuring multiple control modes as well as intelligence derived from sensors, has been presented. At its core is the Arduino UNO microcontroller, which is the processing engine of the system that communicates through multiple sensors and various other modules like the L298N motor driver, which allows it to move by remotely controlling four DC gear motors. It has the HC-05 Bluetooth module, which gives it the capability to be controlled wirelessly by an Android application designed by the authors using the MIT App Inventor [18, 19]. This robot can function using both manual directional control and through voice recognition, offering the user a friendly and interactive environment. For the purpose of autonomous operation, the robot is equipped with obstacle detection and avoidance functions via an ultrasonic distance sensor (HC-SR04) and two IR sensors. Moreover, the RFID module adds identity verification and labeling functions [20, 21] to the robot, making this robot qualified and skilled in secure access and inventory automation. A servo motor (SG90) is also added to deliver mechanical movement functions, offering dynamic control functions like scanning or changing the orientation

of an obstacle. The proposed robotic system is an exemplary model of a holistic approach to smart automation, including real-time control functions, sensor fusion, and mobile connectivity functions. The open-source nature of this robot makes it an efficient learning and development platform for robotics, embedded systems, and the Internet of Things (IoT) [22].

II. MATERIALS AND METHODS

A. Electronic Components

Various electronic components are used to construct electronic circuits. The components employed in the proposed circuit diagrams are listed in Table 2.

B. Arduino UNO

Arduino Uno [23–27] - Arduino Uno is one of the popular microcontroller boards based on the ATmega328 microcontroller. Arduino Uno has its own IDE where users can develop code to execute on the microcontroller easily. In general, in an embedded system, there are different sensors providing data about the environment to the microcontroller. Then, the microcontroller processes the data and accordingly activates different control signals to the output peripherals or devices.

This Arduino board has a total of 28 pins, which include 14 dedicated digital input/output pins, of which six are capable of pulse-width modulation, six for analog input, and other pins for ground, 3.3V, 5V, VIN, Reset, and the analog reference, AREF. There are also three ground pins on this electronic circuit component. It is equipped with an ATmega328 microcontroller that is accompanied by 32 KB of flash memory, 2 KB of SRAM, and 1 KB of EEPROM memory that is useful for such small to medium-scale applications. Arduino Uno microcontrollers are mainly used for programming based on C++ and are equipped with various interfaces such as USB, DC jack for external DC power, ICSP programming interfaces, and a reset button that helps to restart the program.

The board operates at a voltage of 5 V and usually has a requirement for the input supply voltage of around 7 to 12 V, which has a maximum limit of 20 V.

C. L298N Motor Driver Module

L298N Motor Driver Module [28, 29]: This is a dual H-bridge motor driver module, generally utilized for the control of the direction and speed of either one or two DC motors. This motor driver module has the capability of driving DC motors of a maximum current of up to 2A

for each channel, along with an operational voltage range of 5V to 35V.

There are four output pins for DC motor connections and four input pins for receiving the signals from the microcontroller. Apart from this, it also comprises two enable pins that can be linked with the pulse width modulation output of the microcontroller, and this enables controlling the speed of the motor through this output and removing the respective enable jumpers.

The L298N module also contains a built-in 5V voltage regulator that can power and regulate external logic circuits. The built-in voltage regulator can be activated for inputs with a max motor voltage of 12V; however, for inputs above 12V, it must be shut down.

D. KY-022 IR Receiver Module

KY-022 IR receiver circuit [30] is introduced inside the robotic car and is connected to Arduino pin 12. KY-022 IR receiver circuit works at an operational voltage range of 2.7 - 5.5 V and at a frequency of 37.9 kHz. Its receiving range is up to 18 m. Connectivity as well as decoding of the IR signals sent through IR receiver circuit and the IR remote is performed through the code written in Listing 1.

```

1 #include <IRremote.h>
2
3 const uint8_t IR_RECEIVE_PIN = 12; // Use
4   9 if your hardware is connected to pin
5   9
6
7 void setup() {
8   Serial.begin(9600);
9   IrReceiver.begin(IR_RECEIVE_PIN,
10    ENABLE_LED_FEEDBACK);
11 }
12
13 void loop() {
14   if (IrReceiver.decode()) {
15     Serial.print("IRCode:0x");
16     Serial.println(IrReceiver.decodedIRData
17       .decodedRawData, HEX);
18     IrReceiver.resume();
19   }
20 }

```

Listing 1: Arduino code for IR signal reception using the KY-022 module

E. Servo Motor (SG90)

The SG90 micro servo motor [31, 32] has a rotation of about 180°, of which about 90° can be achieved in each direction. The SG90 combines the controller of the motor, the gearbox, and the position feedback system in a single board, which can be precisely controlled in terms

Table 2: Motor Direction Control Table

Left Wheel	Right Wheel	Car Motion	IN1 (9)	IN2 (8)	IN3 (7)	IN4 (6)
F	F	Forward	1	0	0	1
B	B	Backward	0	1	1	0
F	B	Turn Right	1	0	1	0
B	F	Turn Left	0	1	0	1

ENA (10)	ENB (5)
ACTIVE	ACTIVE

Table 3: Enable Pins Status

of angle. The motor can be operated through standard servo libraries and control codes used for Arduino.

The servo works at an input voltage of around 5V and is controlled through a pulse-width modulation signal whose period is fixed at 20 ms and operates at a frequency of 50 Hz. For the control signal, the pulse width is usually varied between 1 ms to 2 ms for the minimum and maximum angle positions, respectively. In the proposed model, the signal pin of the SG90 servo motor is wired to digital pin 5 of the Arduino microcontroller.

F. Ultrasonic Sensor (HC-SR04)

In the proposed robotic system, the HC-SR04 ultrasonic sensor is used for the purpose of obstacle detection [33–35]. The working of the sensor is based on a DC input voltage ranging between 3.3 and 5 volts, with a typical current of around 15 mA, but the current is less than 2 mA during the standby mode. The sensor works on an ultrasonic frequency of 40 kHz, and its induction angle is less than 15 degree. The HC-SR04 is capable of detecting objects at ranges of 2 cm to 400 cm with a resolution of about 1 cm and accuracy of about 3 mm. To detect distances, the sensor requires a trigger pulse of only 10µs, and the sensor returns the detected distance in the form of an echo signal. It also supports a four-pin TTL interface and is fully compatible with Arduino technology. In our proposed system, the trigger pin of the HC-SR04 is linked to digital pin 10 of Arduino, while the echo pin is linked to digital pin 11 of Arduino.

G. Bluetooth Module HC-05

The HC-05 [36–38] is a popular Bluetooth Serial Port Protocol (SPP) wireless communication device for realizing straight wireless communication between electronic apparatus. The device has the ability to support both Master and Slave modes, thus providing appropriateness in wireless transmission of data. The module meets the specifications of the Bluetooth v2.0 + EDR (Enhanced Data Rate), with a data transfer rate of a maximum of 3 Mbps, and it also contains a complete radio transceiver and baseband controller operating on a 2.4 GHz frequency. HC-05 contains a total of six pins and connects with the microcontroller using these pins, enabling the communication process between two devices. The pins housed by HC-05 are: ENABLE, VCC, GND, TX, RX, and STATE.

This HC-05 module works on a voltage of 3.3-5V and gives a communication range of up to 90 m when used wirelessly.

H. Android Mobile Application

In this case, the android application is software created through programming languages that is capable of running on android operating systems. For the purpose of controlling the robotic car, an android application is developed in this work that is freely downloadable from the source given out in [39–41].

III. DESIGNING METHODOLOGY

A. Robotic Motor Control using L298N Module

This paper introduces the design, development, and implementation of a two-wheel driving robotic car, which uses an L298N motor driver module as the control mechanism. The major aim of this development is to ensure that the directional control of this robotic car can be done in an accurate and programmable manner through digital logic signals. The design uses an L298N H-bridge module as its major component. A detailed motor control table was designed to link the input commands with the actions of the motors, which would allow the robot to move ahead, retreat, as well as turn. The enable pins, ENA and ENB, were used for power control of the motors.

The top-down schematic diagram explains the operation of the two-motor system quite effectively with appropriate markings given for both left and right wheel rotations. The experimental result with calculations for angular velocities revealed that the vehicle made 5.5 revolutions within a time of 5 seconds with an angular velocity of $396^\circ/\text{s}$.

In conclusion, this project illustrates a viable and economical way of performing basic robotic motion control using low-cost components. The motor control logic and motion behaviors can be referred to by Tables 2 and 3. These tables show the behaviors of the robot when the motors move in different directions based on the signals. With these findings, there can be good support for further improvement work on achieving completely autonomous mobile robots with sophisticated navigation skills. Moreover, Fig. 1 shows a top-down schematic diagram of the two-wheel drive robotic platform with representation of motors A and B showing the left and right sides of the robot with their respective rotations marked by the arrow marks.

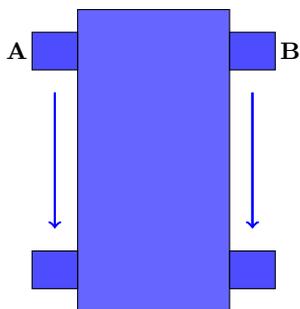


FIG. 1: Top-view schematic of a two-wheel differential drive robot. Motors A and B represent the left and right wheel drives, respectively. Arrows indicate the direction of motor rotation.

B. Angular Velocity Calculation

In order to examine the rotational motion of the vehicle, the angular velocity was calculated in terms of the revolutions made within a specific time interval. The vehicle made a total of 5.5 revolutions in 5 seconds.

The angular displacement θ (in degrees) is given by:

$$\theta = N \times 360^\circ = 5.5 \times 360^\circ = 1980^\circ \quad (1)$$

The angular velocity ω is then calculated using the relation:

$$\omega = \frac{\theta}{t} = \frac{1980^\circ}{5\text{ s}} = 396^\circ/\text{s} \quad (2)$$

Therefore, the angular velocity of the vehicle is $396^\circ/\text{s}$. The direction of the rotation depends on the sign of the angular velocity. A positive sign indicates a left turn, which is counterclockwise, and a negative sign indicates a right turn, which is clockwise. In this case, the analysis of the drive system determined the sign of the angular velocity.

IV. ULTRASONIC SENSOR-BASED OBSTACLE AVOIDANCE SYSTEM

Figure 4 shows the circuit design of the obstacle avoidance control system constructed using the HC-SR04 ultrasonic sensor. The robotic car operates autonomously and detects obstacles by adjusting its path according to the distances measured by the ultrasonic sensor. Figure 3 illustrates the circuit diagram of the obstacle avoidance system, while Figure 4 shows the physical setup of the robotic vehicle.

The hardware setup for the obstacle avoidance system consists of an Arduino Uno, an HC-SR04 ultrasonic sensor, and an SG90 servo motor. The trigger pin of the ultrasonic sensor is connected to Arduino digital pin 10, and the echo pin is connected to digital pin 11. The VCC and GND of the sensor are connected to the Arduino's 5V supply and ground, respectively. The SG90 servo motor is used to rotate the ultrasonic sensor for scanning. Its signal pin is connected to Arduino digital pin 5, with VCC and GND connected to the 5V supply and ground, as shown in Figures 9 and 8.

Obstacle-detection functionality is then provided, after integrating the sensors and servos, by calling the appropriate ultrasonic sensing library. The car drive system is implemented with an L293D or L298N motor driver module along with an Arduino Uno and four DC gear motors. The motor driver control inputs have to be attached to the Arduino digital pin and extra pins for PWM have to be used in order to vary the motor speed - as can be seen in Fig. 9.

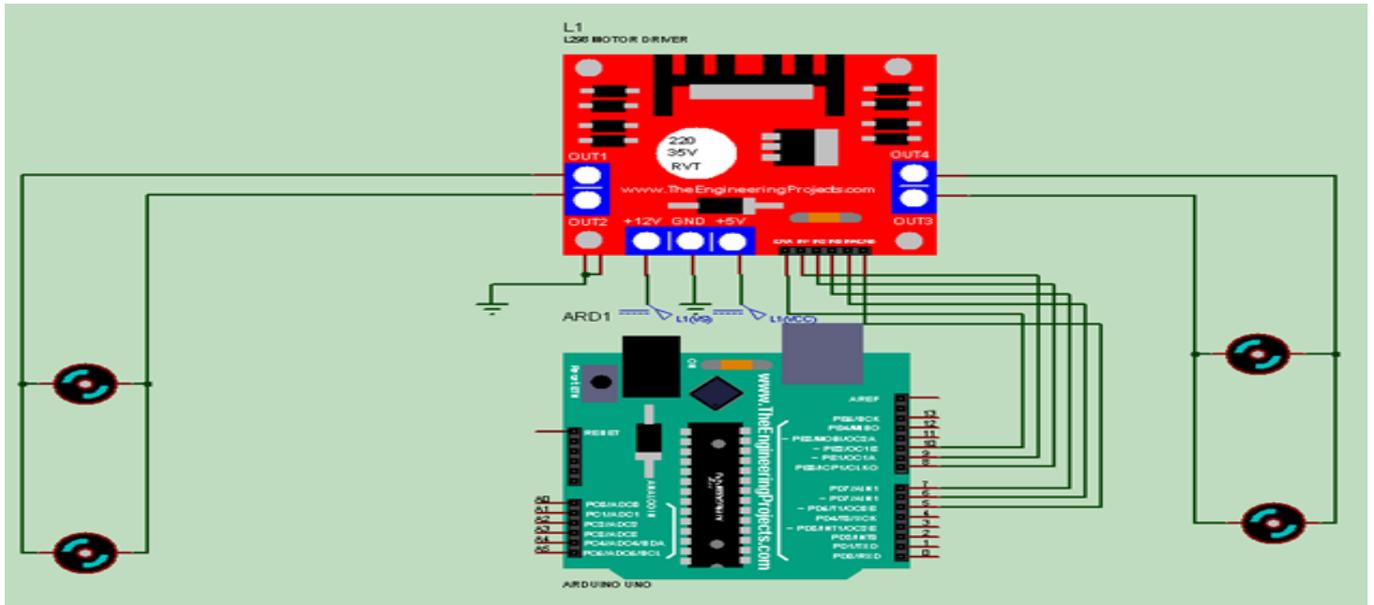


FIG. 2: L298N Motor Driver Module used to control the direction and speed of DC motors.

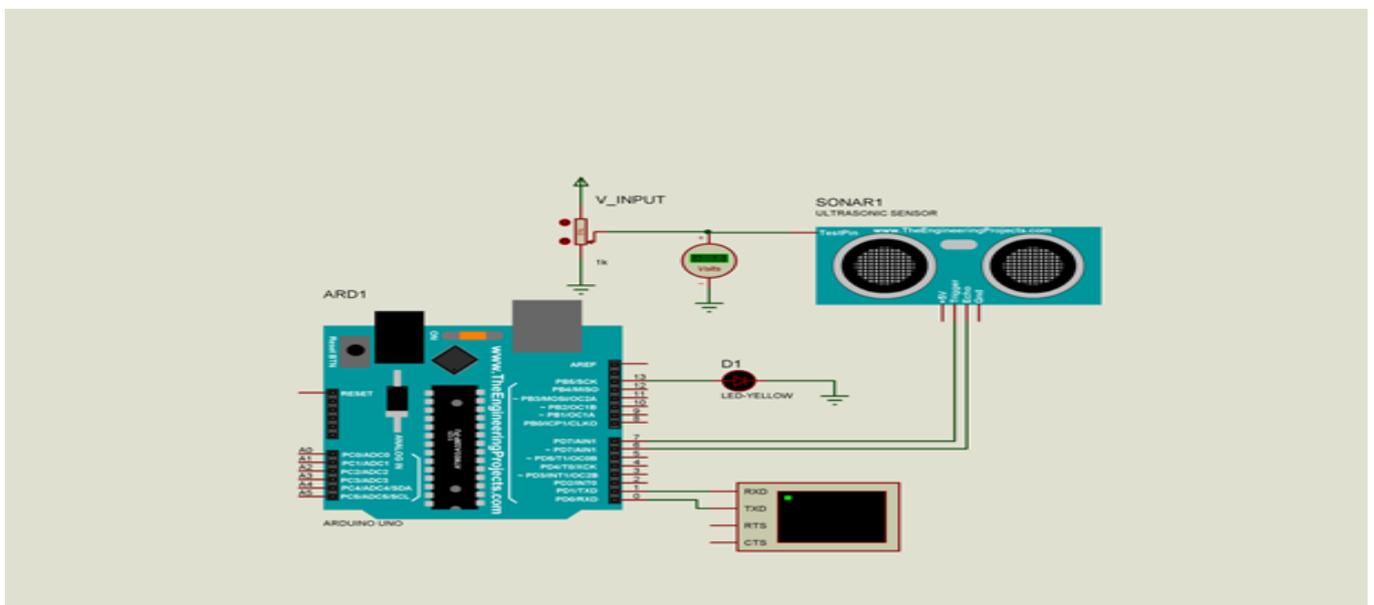


FIG. 3: HC-SR04 Ultrasonic Sensor used for obstacle detection by measuring distance using sound waves.

V. IRREMOTE CONTROLLED ROBOT

The IR remote-controlled robot is designed with an Arduino Uno microcontroller, an L298N motor driver module, four DC gear motors with wheels, an KY-022 IR receiver module, an IR remote control, jumper wires, and a power supply system using three batteries with 2400 mAh capacity. The hardware connection of the IR control system is shown in Fig. 5. Infrared communication necessitates a combination of an infrared transmitter and an infrared receiver. The infrared remote is equipped

with an integrated infrared transmitter that sends an infrared signal each time a button is pressed on this device. This infrared signal is then received from an infrared KY-022 receiver module as depicted in Fig. 9. This infrared received signal is then processed and interpreted by an Arduino controller using an *IRremote library*, created by Ken Shirriff [42]. Each of the pushbutton switches on an infrared remote is programmed to have a predefined hexadecimal (HEX) code [42].

Based on the decoded HEX values, it allows commanding the robotic car's movements to move forward, go

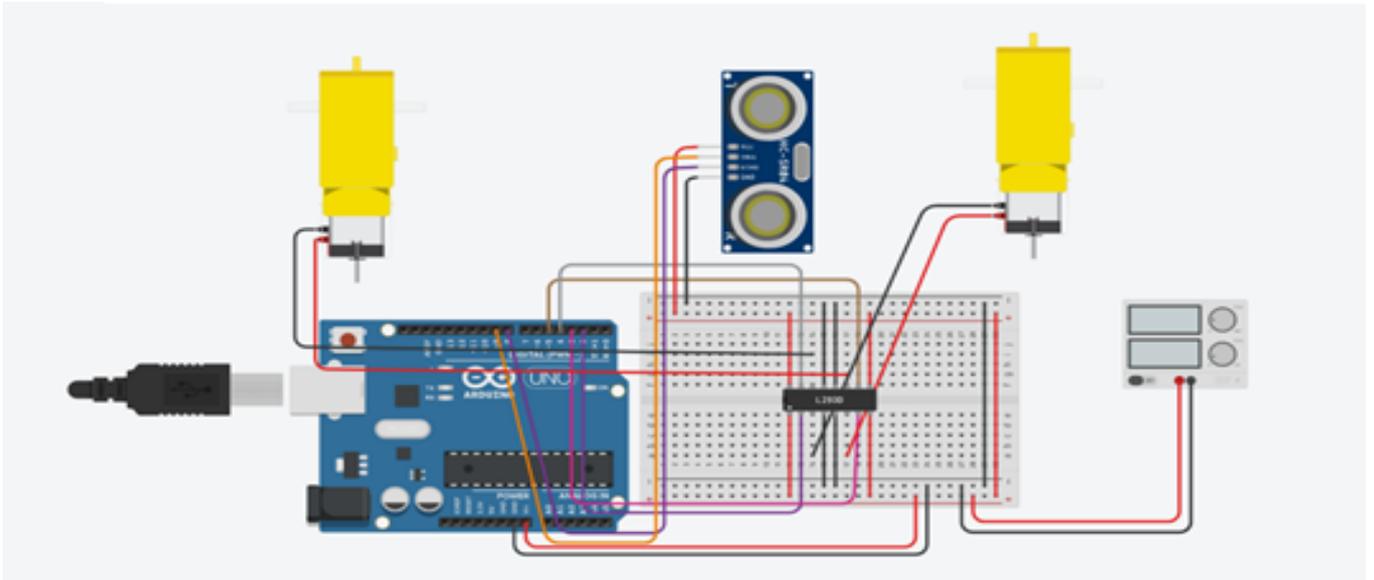


FIG. 4: Tinkercad simulation showing the interfacing of the HC-SR04 Ultrasonic Sensor with Arduino UNO for obstacle detection.

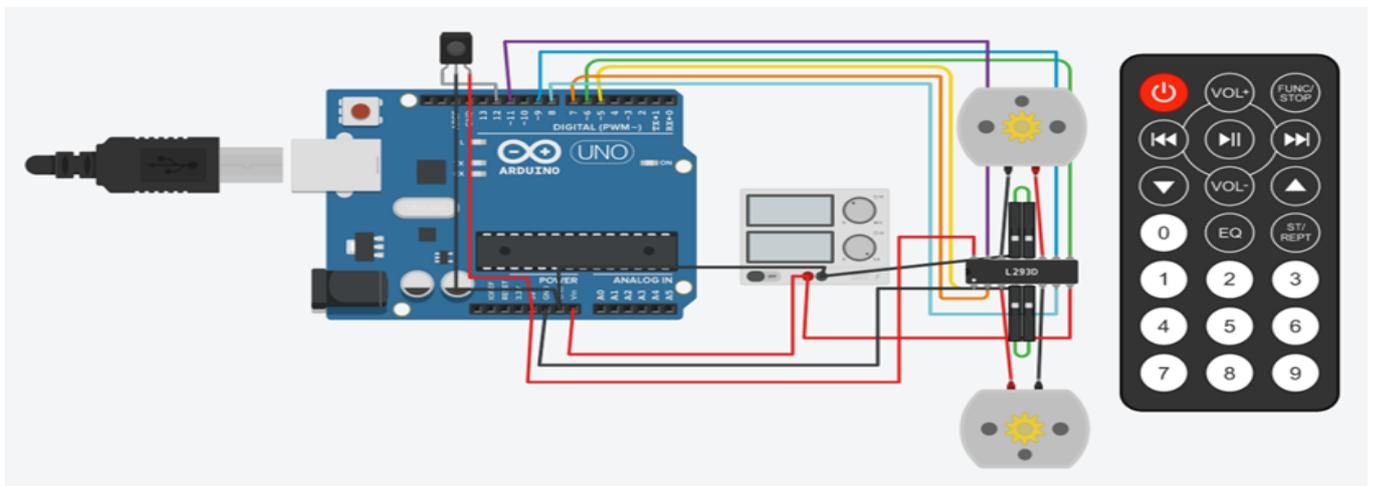


FIG. 5: Infrared (IR) control interface of the robotic car showing the KY-022 receiver module integrated with the Arduino Uno for remote-based navigation and control.

backward, turn left or right, or even stop. Moreover, the same IR-based interface enables higher-level modes of operation, such as obstacle avoidance, line-following behavior, and voice-assisted control [30, 31]. Control logic devised for these modes runs in real time, thus making the operation of robot navigation effective and trustworthy. KY-022 IR receiver module consists of three pins. These are VCC, GND and DATA, where the VCC pin is connected to the Arduino 3.3–5 V power supply.

GND to ground and DATA is interfaced with Arduino digital pin 12, which receives the infrared control signals as shown in Fig. 7 9.

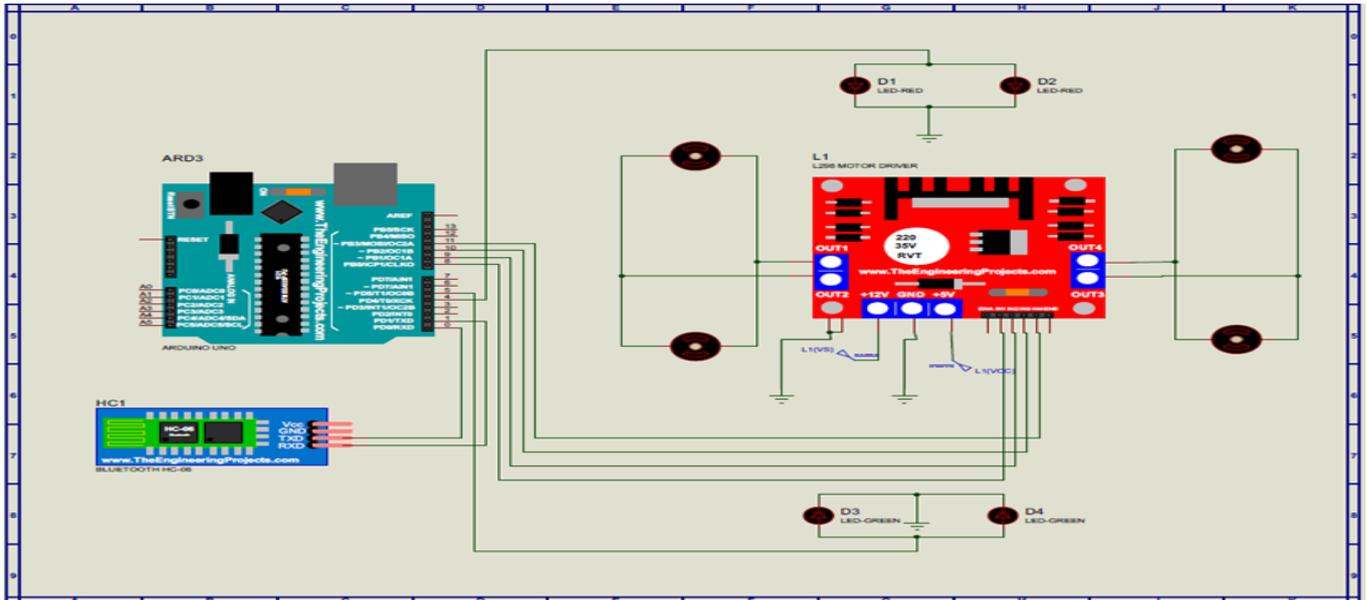


FIG. 6: Bluetooth HC-06 module used for wireless communication between the Arduino and a smartphone application.

VI. SPEECH-CONTROLLED AUTOMATION SYSTEMS (SCAS)

This paper aims to create a robotic vehicle that can be managed by voice commands directed by humans. Such technology is generally known as Speech-Controlled Automation Systems (SCAS). In this proposed system, it can be considered as an initial prototyping stage of Voice-Controlled Robotic Vehicles (VCRVs) that can be remotely controlled using an Android smartphone. Android smartphones are efficient and proficient interfaces for robotic control, as they have integral functionalities involving microphones, wire connectivity, and computing capacity. In this proposed system, an Android app is linked with a microcontroller, and communication between smartphones and robotic vehicles can be done using Bluetooth technology. The voice commands provided by users can be wirelessly received and processed by the Bluetooth module in order to be directed by the control unit.

The main purpose of VCRV is to identify the user-defined voice commands and act accordingly. Voice commands include basic motion-related commands like forward, backward, left, right, and stop. Initially, the system needs voice training or predefined command mapping after which the microcontroller interprets the received commands to execute related actions. The development of a voice-controlled robotic vehicle is supported with the motivation of decreasing man's efforting and making the operation hands-free. In conventional vehicles, man is solely responsible for all operations regarding steering, applying brakes, and giving acceleration. How-

ever, in advanced technologies, advantages can be extended regarding automation and users' convenience by including modes of operation using voice and gestures.

Voice and gesture recognition systems can greatly help in natural human machine interaction through the interpretation of voice and body language commands, thereby improving the communication channel between humans and machines. The objective of this study is to make the robot control system simpler while improving its safety and usability functions as mentioned in [43]. Voice commands can be executed through an advanced smartphone application, and the system performance can be evaluated through experiments. The feasibility of distant voice control and system response can be evaluated based on different system interaction levels. Applications of this system can be found in an industrial setup, healthcare centers, home automation, and environment monitoring labs.

Speech recognition functionality in the proposed system is achieved using smartphone voice processing capabilities, either through the smartphone's microphone or an Android application [44]. Voice-controlled robotic assistants have the potential to greatly aid humans in their everyday chores and act as Intelligent Personal Assistants (IPAs) that provide various services. In this research, an Arduino-controlled unit is employed for designing the robotic vehicle, which forms the heart of this system. Voice commands fed into the Android smartphone are processed into text messages and sent to the Arduino controller through the Bluetooth device.

Upon receiving the command, the robot performs the respective action or motion. Moreover, for advanced

designs, a camera component can be employed for object recognition within the robot environment. Subsequently, the robot can turn entirely within 360 degrees and examine pictures taken by cameras to determine the mean distance of objects. There have also been designs proposed for four-wheeled robots controlled by Android smartphones through Bluetooth technology [45]. Moreover, voice-controlled software designs always comprise recognition and monitoring interfaces through which one can confirm the interpretation of the command and resulting actions of the robot. A structured command vocabulary and grammar play an important role in successful robot control [46].

The Android app communicates, in the proposed system, directly with the interfaced HC-05 Bluetooth module to an Arduino Uno R3 microcontroller. It receives voice from a smartphone and forwards the signal to the control unit, thereby allowing voice-based or push-button control modes [47]. In summary, efficient human-robot interaction is the ultimate achievement of the proposed voice-controlled robotic vehicle, where it can listen and act accordingly to user commands intuitively and reliably.

VII. DESIGN AND IMPLEMENTATION OF A LINE-FOLLOWING ROBOT

Line Following is one of the basic applications of mobile robotics. A line-following robot is an autonomous robot that is able to detect and trace a pre-defined path that is usually demarcated on a smooth surface with a black line on a white background. The working of a line-following robot is dependent on the reflective behavior of light on a contrasting surface. White surfaces are reflective to a great extent compared to a black surface, which is usually absorbing.

A line follower robot uses infrared (IR) transmitter and receiver modules (photodiodes) for line tracking. The IR transmitter transmits infrared light toward the surface, and the reflected signal is received by the IR sensor. When the infrared light reflects off a white surface, it is picked up by the receiver, and a voltage change is detected. Black surfaces absorb infrared light, and no signal is received by the sensor.

Based on the output of the sensors, the Arduino Uno processes the input signals. Whenever the IR sensor finds a white surface, it gives the Arduino a logic high signal (1) as compared to the logic low signal (0) that is given when it finds the black line. The digital inputs help the microcontroller of the Arduino to produce the necessary signals for controlling the movements of the robots so that it follows the line successfully.

The essential components of the line-following robot consist of an Arduino Uno, IR sensors, an L293D motor driver, four BO series DC motors, a 4-wheel drive kit, and

a 3.7V lithium-ion battery pack. In this configuration, motors 1 and 2 share a single channel of the motor driver, and motors 3 and 4 share another channel. The input pins of the motor driver, namely IN1, IN2, IN3, and IN4, can be directly connected to Arduino Uno pins 6, 7, 8, and 9, respectively. A jumper wire connects the +5V source and the enabling pins EN1 and EN2, enabling speed control using Arduino pins 5 and 10 with direct and PWM signals.

BO series DC motors have been selected due to their high torque, sufficient rotational speed, and low power consumption at a relatively low operation voltage. The motors come in both single-shaft and dual-shaft builds, and this project requires four single-shaft BO motors will be needed. Power supply used for this purpose involves three 3.7-V lithium-ion batteries connected in series to provide a total voltage of 11.1 V, besides an on-off switch enabling safe power supply. Electrical connections will be ensured through the use of male-to-female jumper wires, while all components are fixed on a robotic chassis that may be bought from the market or self-made using cardboard.

If the bot does not move in the proper direction after uploading the control code to the Arduino, motor wiring may be changed. Besides, IR sensors need to be tuned by adjusting their onboard potentiometer for accurate detection between a black line and white background. Figure 9 gives the working principle of a line follower robot with a track prepared by black tape on a white surface. Also, a corresponding block diagram 8 of the system is included for clarity.

VIII. SOFTWARE DESCRIPTION

The software environment provided by the Arduino development kit is utilized for designing and uploading the control codes that determine the overall functionality of the proposed system. Moreover, the programming code for the system is developed using the C/C++ programming language supported by the Arduino Integrated Development Environment (IDE) software version 1.8.12. After the development of the programming code, the code is uploaded into the microcontroller, wherein the code is stored in non-volatile memory for execution. This paper utilizes the Arduino development kit for firmware development. Simulations involving microcontrollers are carried out through use of the Proteus Design Suite by loading either the hex or debug file into the microcontroller in the schematic. The Proteus Design Suite supports co-simulation of microcontrollers with interconnected analog and digital electronic components. In addition to Proteus, other software tools that are employed in the development process include Tinkercad, which is used for simulation of circuits, Fritzing, which is employed to create schematic and wiring diagrams, and MIT App In-

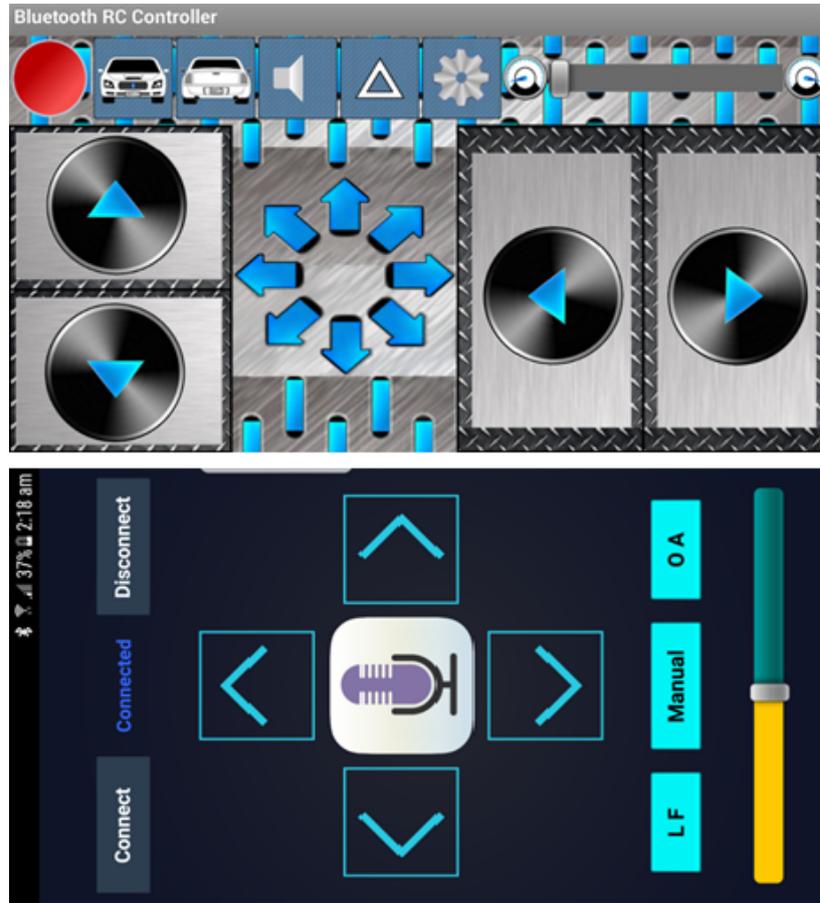


FIG. 7: Bluetooth remote control interface developed using MIT App Inventor for controlling the robotic car via smartphone.

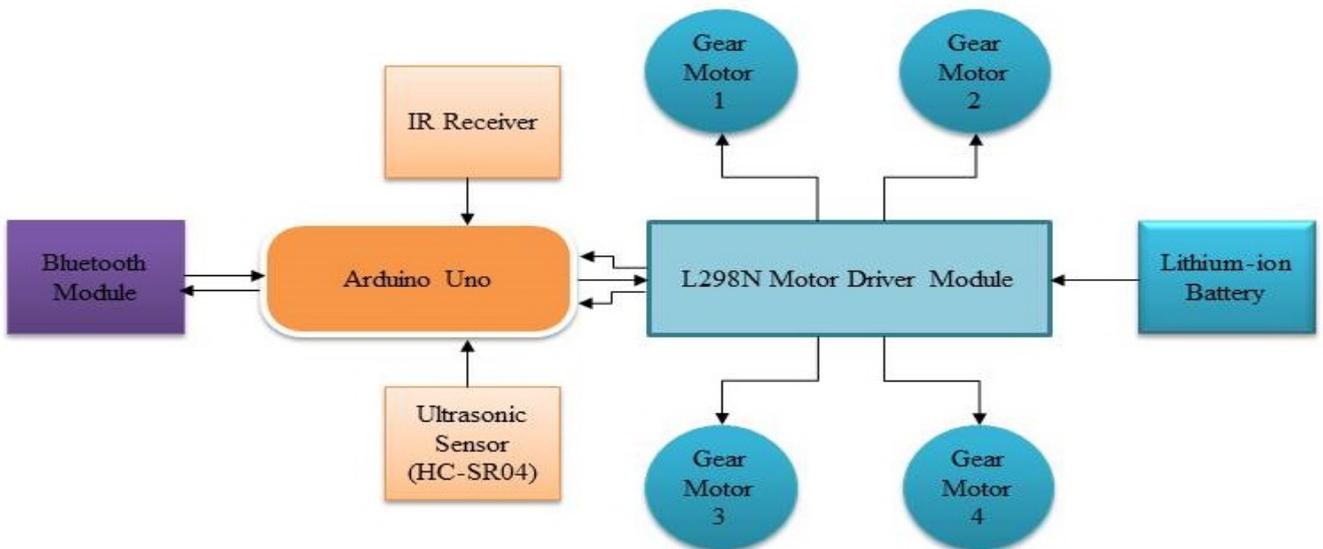


FIG. 8: Flowchart of the complete robotic car control system.

ventor, which is employed for designing and implementing the Android smartphone application that controls the

Arduino UNO Pin Configuration Table

Component	Pin on Component	Arduino Pin	Notes
Bluetooth Module (HC-05)	TX	RX (Pin 0)	Serial communication (Bluetooth to Arduino)
	RX	TX (Pin 1)	Use voltage divider for 3.3V logic compatibility
L298N Motor Driver	IN1	Pin 6	Motor 1 control input
	IN2	Pin 7	Motor 1 control input
	IN3	Pin 8	Motor 2 control input
	IN4	Pin 9	Motor 2 control input
	ENA	5V or PWM 5	Motor 1 speed control
	ENB	5V or PWM 10	Motor 2 speed control
Ultrasonic Sensor (HC-SR04)	Trig	Pin 10	Transmits ultrasonic trigger pulse
	Echo	Pin 11	Receives reflected echo signal
KY-022 IR Receiver Module	DATA	Pin 12	Receives infrared signals
	VCC	5 V	Power supply
	GND	GND	Ground connection
IR Sensor (Left)	OUT	Pin 2	Detects left obstacles
IR Sensor (Right)	OUT	Pin 3	Detects right obstacles
Servo Motor (SG90)	Signal	Pin 5	Rotates sensor module
Power Supply	VCC	VIN / L298N VCC	Power input
	GND	GND	Common ground

Table 4: Complete source code for the proposed robotic car control system.

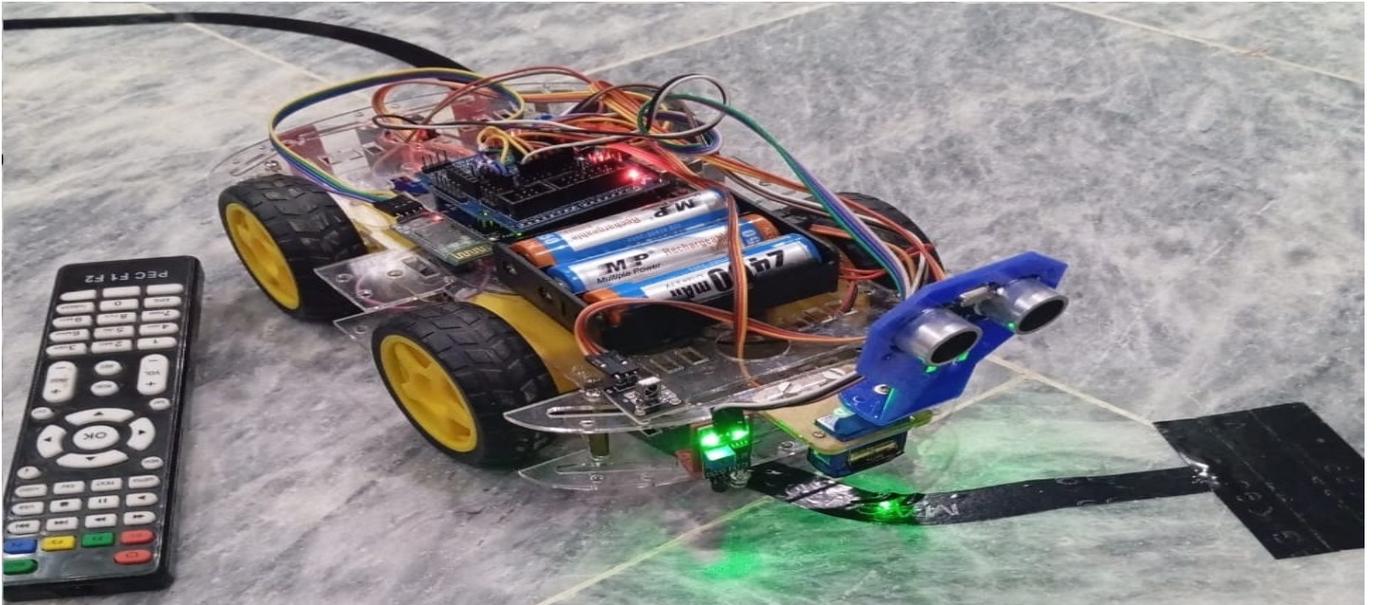


FIG. 9: Experimental setup of the proposed robotic car showing the Arduino controller, ultrasonic obstacle detection module mounted on a servo motor, DC gear motors with motor driver, power supply, and remote-control interface.

robotic car.

IX. CONCLUSION

This article has discussed the design and implementation of single hardware and multiple application robotic car control systems using the Arduino platform. To ensure multiple functionalities, the system has been designed to support various operational capabilities like line following, obstacle avoidance, IR remote control, and Android mobile control using the MIT App Inventor development platform with voice recognition capabilities. To accomplish obstacle detection, an ultrasonic sensor is employed. Line following capabilities have been incorporated using IR proximity sensors. IR remote control functionality has been added to the system using the KY-022 infrared receiver module. To establish wireless transmission of commands, Bluetooth communication has been used. Though Bluetooth allows for wireless communications with low expenses for development and operation, the actual operational range allowed for this type of system is no more than 10-15 m. This results in disconnections that lead to delay times for control commands transmitted to the system. Noise in the environment also affects the voice commands used for system control. The system has been validated through the development of a laboratory-scale hardware implementation for reliability, simplicity, flexibility, and applicability at low costs. It appears that the system is easily scalable for various application domains. The possible application domains for this system could be smart cars, home automation

robots, articulated robots, humanoid robots, quadruped robots, and hybrid robotic systems. It appears that the single-hardware multi-application robotic control system proposed has several advantages in terms of design complexity, power consumption, simplicity, and scalability for various applications.

X. FUTURE WORK

Future researchers will work towards improving the functionality and scalability of the proposed design for the robotic car system. The Global Positioning System (GPS) module will be added to the design to allow the system to track the position and navigate accordingly. Additionally, the power source of the system will be complemented with a solar source, considering that solar cells are capable of providing direct current (DC) to power the system and the electronics, which are required by the receiver.

The system can also be modified to work together with different robotic controlling systems without any major modifications. Moreover, since Bluetooth has poor communication range, wireless communication can be done using Wi-Fi technology through either the installation of local routers or by installing a GSM module. Additionally, a wireless camera can be incorporated into the system to provide live streaming of images to facilitate remote controlling of the robotic vehicle through its camera.

* Electronic address: mjkh@ele.qau.edu.pk

- [1] X.-Y. Gong, H. Su, D. Xu, Z.-T. Zhang, F. Shen, and H.-B. Yang, *International Journal of Automation and Computing* **15**, 656 (2018).
- [2] M. O. Qureshi and R. S. Syed, *Safety and health at work* **5**, 198 (2014).
- [3] Y. Fernando, A. Mathath, and M. A. Murshid, *International Journal of Robotics Applications and Technologies (IJRAT)* **4**, 43 (2016).
- [4] S. Kanere, R. Shinde, V. Tornekar, D. Zagade, and V. Sonar, *Int. J. Adv. Res. Comput. Commun. Eng* **6**, 44 (2017).
- [5] R. Amrin, M. M. Ahmed, A. Al Zubaer, and M. M. H. Naim, *Global Scientific Journals* **8** (2020).
- [6] M. Saravanan, B. Selvababu, A. Jayan, A. Anand, and A. Raj, in *IOP Conference Series: Materials Science and Engineering* (IOP Publishing, 2020) p. 012125.
- [7] S. Srivastava and R. Singh, *International Research Journal of Engineering and Technology (IRJET)* **7**, 4033 (2020).
- [8] A. Chaudhry, M. Batra, P. Gupta, S. Lamba, and S. Gupta, in *2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)* (IEEE, 2019) pp. 415–417.
- [9] D. Punetha, N. Kumar, V. Mehta, *et al.*, *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)* **2**, 2446 (2013).
- [10] O. O. Ebiesuwa, Y. A. Adekunle, L. A. Akinyemi, and O. D. Oyerinde, *International Journal of Engineering Research & Technology* (2013).
- [11] O. Gumus, M. Topaloglu, and D. Ozelik, *Procedia Computer Science* **102**, 202 (2016).
- [12] M. S. Rao and M. Shivakumar, *Journal of Robotics and Control (JRC)* **2**, 244 (2021).
- [13] J.-D. Warren, J. Adams, and H. Molle, in *Arduino robotics* (Springer, 2011) pp. 51–82.
- [14] M. J. Khan and R. B. Sheraliyat, arXiv preprint arXiv:2512.12241 (2025).
- [15] O. Bukin, D. Proshenko, A. Chekhlenok, S. Golik, I. Bukin, A. Mayor, and V. Yurchik, *Sensors* **18**, 1680 (2018).
- [16] S. C. Diamantas, A. Oikonomidis, and R. M. Crowder, in *2010 IEEE International Conference on Imaging Systems and Techniques* (IEEE, 2010) pp. 426–430.
- [17] S.-Y. Juang and J.-G. Juang, *Applied Sciences* **6**, 82 (2016).
- [18] A. Top and M. Gökbulut, *Wireless Personal Communications* **126**, 1403 (2022).
- [19] K. E. Hendrickson, *Writing and connecting IoT and mobile applications in MIT App Inventor*, Ph.D. thesis, Massachusetts Institute of Technology (2018).
- [20] S. Dewanto, M. Munir, B. Wulandari, and K. Alfian, in *Journal of Physics: Conference Series* (IOP Publishing, 2021) p. 012012.
- [21] Y. Roza, K. Musliadi, and Y. Pernando, *Scientific Literacy Innovation and Technology Journal* **1**, 26 (2024).
- [22] S. Al-Sarawi, M. Anbar, K. Alieyan, and M. Alzubaidi, in *2017 8th International conference on information technology (ICIT)* (IEEE, 2017) pp. 685–690.
- [23] L. Louis, *International Journal of Control, Automation, Communication and Systems (IJCACS)* **1**, 21 (2016).
- [24] A. Mohamed, C. Yang, and A. Cangelosi, *IFAC-PapersOnLine* **49**, 155 (2016).
- [25] P. Sihombing, N. Karina, J. Tarigan, and M. Syarif, in *Journal of Physics: Conference Series*, Vol. 978 (IOP Publishing, 2018) p. 012014.
- [26] M. Yusro, N. A. Guntoro, and Rikawarastuti, in *AIP conference proceedings* (AIP Publishing LLC, 2021) p. 060004.
- [27] Y. A. Badamasi, in *2014 11th international conference on electronics, computer and computation (ICECCO)* (IEEE, 2014) pp. 1–4.
- [28] P. Peerzada, W. H. Larik, and A. A. Mahar, *International Journal of Electrical Engineering & Emerging Technology* **4**, 21 (2021).
- [29] V. A. Akpan and A. S. Eyefia, *sensors* **15**, 19 (2021).
- [30] A. Perera, K. Botirov, H. Sallouha, and M. Katz, *IEEE Sensors Journal* (2025).
- [31] M. Lu, *International Journal of Robotics and Automation* (2021).
- [32] M. Shuai, *Soft Matter* (2019).
- [33] A. K. Kumar, A. Sarangi, D. Singh, S. Dash, and I. Mani, *Journal of Scientific & Industrial Research (JSIR)* **82**, 1091 (2023).
- [34] N. Fardin, M. M. R. Masrafee, and M. A. Upoma, in *2022 25th International Conference on Computer and Information Technology (ICCIT)* (IEEE, 2022) pp. 1080–1085.
- [35] T. Gluck, M. Kravchik, S. Chocron, Y. Elovici, and A. Shabtai, *Sensors* **20**, 6157 (2020).
- [36] A. Nelwan, P. Manembu, A. Wauran, F. Manoppo, and C. Mamahit, *JURNAL EDUNITRO Jurnal Pendidikan Teknik Elektro* **3**, 9 (2023).
- [37] A. Giorgio and F. Paris, *International Journal of Advanced Engineering Research and Applications (IJ-ERA)* **3** (2017).
- [38] G. N. Mori and P. R. Swaminarayan, in *Data Science and Intelligent Applications: Proceedings of ICDSIA 2020* (Springer, 2020) pp. 375–382.
- [39] A. Yusuf and M. A. Baba, *International Journal of Computer Applications* **975**, 8887 (2014).
- [40] A. ALFoudery, A. A. Alkandari, and N. M. Almutairi, *Indonesian Journal of Electrical Engineering and Computer Science (IJECS)* **10**, 120 (2018).
- [41] F. M. Lopez-Rodriguez and F. Cuesta, *Applied Sciences* **11**, 48 (2020).
- [42] M. Moshiri, *Hardware Implementation of Robotic Arm for Motion Control Using Infrared Communication*, Ph.D. thesis, California State University, Northridge (2020).
- [43] G. Lunghi, R. Marin, M. Di Castro, A. Masi, and P. J. Sanz, *IEEE Access* **7**, 127290 (2019).
- [44] P. Nath and U. C. Pati, in *2018 3rd International Conference for Convergence in Technology (I2CT)* (IEEE, 2018) pp. 1–4.
- [45] M. Yan and H. Shi, *International journal of wireless & mobile networks* **5**, 65 (2013).
- [46] E. Iadanza, G. Pasqua, D. Piaggio, C. Caputo, M. Gherardelli, and L. Pecchia, *Health and Technology* **13**, 285 (2023).
- [47] K. A. Azizi, A. Sembiring, and M. Arfisyari, *International Journal of Health Engineering and Technology (IJHET)* **1** (2022).