International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 11 | Nov - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

Smart Bluetooth and PC Controlled Robotic Arm

Dr. Sweta Raut
Department of Computer Science and Engineering Jhulelal Institute of Technology
Nagpur, India

Khushi Mukesh Basantwani
Department of Computer Science and Engineering Jhulelal Institute of Technology
Nagpur, India
ugf23cs028@jitnagpur.edu.in

Mansi Ulhas Patil
Department of Computer Science and Engineering Jhulelal Institute of Technology
Nagpur, India
ugf23cs033@jitnagpur.edu.in

Isha Mahendra Khairkar
Department of Computer Science and Engineering Jhulelal Institute of Technology
Nagpur, India
ugf23cs021@jitnagpur.edu.in

Vaishnavi Kishor Parate
Department of Computer Science and Engineering Jhulelal Institute of Technology
Nagpur, India
ugf23cs064@jitnagpur.edu.in

Suhani Satish Bagde
Department of Computer Science and Engineering Jhulelal Institute of Technology
Nagpur, India
ugf23cs059@jitnagpur.edu.in

Abstract— This paper presents a low-cost Smart Bluetooth and PC Controlled Robotic Arm developed as an educational robotics prototype. The system combines an Arduino-based microcontroller, servo motors, an HC-05/HC-06 Bluetooth communication module, a PC control interface, a mechanical arm and a gripper. The operator can issue movement commands through a Bluetooth-enabled device or a PC interface. The Arduino receives and interprets each command and generates the corresponding control signals for the required servo motor. Coordinated servo movement enables basic actions such as rotation, lifting, gripping and releasing small objects. The paper reviews related Arduino- and Bluetooth-based robotic-arm approaches and discusses the mechanical prototype shown in the supplied photograph. Reported testing indicates command response through both control paths and basic object handling. The supplied project material does not provide numerical measurements for accuracy, response time, communication range or power consumption; therefore, qualitative observations are reported without inventing performance values. Further development may include feedback sensors, computer vision, voice control and intelligent movement planning.

Keywords— Robotic Arm, Arduino, Bluetooth Control, PC Control, Servo Motor, Pick-and-Place, Wireless Robotics, Automation.

I. INTRODUCTION

Robotics combines mechanical structures, electronic control, communication and software to perform physical tasks. A robotic arm is a useful platform for demonstrating these concepts because its joints can be controlled independently and coordinated to produce a desired movement. Industrial manipulators generally require more sophisticated controllers and safety systems, while a small educational prototype allows the same basic ideas to be explored with accessible components.

The Smart Bluetooth and PC Controlled Robotic Arm addresses the need for a simple control platform that supports more than one command interface. An Arduino acts as the common controller. Servo motors are assigned to the arm joints and gripper, while commands can be supplied through Bluetooth or a PC-based interface. The controller interprets each command and sends the corresponding control signal to the appropriate servo.

The project aims to develop a low-cost robotic arm capable of basic movement and object handling. Its objectives include Bluetooth-based remote operation, PC-based command input, multiservo control, a simple operator interface and demonstration of basic automation. The intended scope is educational and small-scale object handling rather than industrial automation.

II. RELATED WORK

The literature considered in the supplied project material focuses on Arduino-based robotic arms, servo control and wireless communication. Dasgupta et al. (2021) described an Arduino-based robotic arm controlled through Bluetooth using an Android application for pick-and-place operations. Nalwandikar et al. (2022) presented a mobile-controlled robotic arm using Arduino UNO and servo motors. Kumar et al. (2023) discussed a Bluetooth-controlled robotic arm for wireless object handling with multi-axis movement. Rubyat et al. (2025) presented a low-cost Bluetooth robotic arm for pick-and-place tasks.

Across these studies, Arduino provides a practical control platform, servo motors provide controlled joint movement, and Bluetooth provides a convenient wireless command path. The project reviewed here combines Bluetooth and PC control in one prototype. The two input methods converge at the Arduino, which keeps the motor-control stage common and makes the platform useful for comparing alternative operator interfaces.

III. SYSTEM ARCHITECTURE

The system is organized into three functional parts: commandinput interfaces, the communication and control path, and the robotic actuation section. The user selects a movement through a Bluetoothenabled device or the PC interface. The command is transmitted to the Arduino, which identifies the requested action and generates the appropriate servo-control signal. The servo then moves the corresponding joint or gripper.

The major hardware elements are the Arduino microcontroller, servo motors, HC-05/HC-06 Bluetooth module, robotic arm structure, gripper, power supply, connecting wires and a PC or laptop. The software environment consists of the Arduino IDE, the microcontroller program, the PC control interface and Bluetooth communication software.

The two input paths remain separate only until the command reaches the Arduino. After reception, both paths use the same command-processing and motor-control logic. Figure 1 summarizes the overall arrangement.

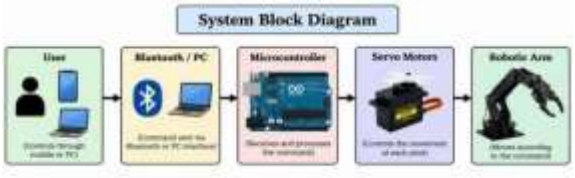


Fig. 1. System block diagram from the supplied project paper.

IV. PROTOTYPE MECHANICAL STRUCTURE

Fig. 2 shows the assembled robotic-arm prototype supplied for inclusion in this paper. The visible construction consists of a flat base plate, a supported rotating or mounting section, multiple servo actuators, rigid link members and a two-finger gripper. The servo housings are mounted at different points in the mechanism so that the arm can produce more than one type of movement. The linkage members transfer actuator movement through the arm structure, while the gripper provides the end-effector used to hold and release small objects.



Fig. 2. Photograph of the assembled robotic-arm prototype (user-supplied image).

The base plate provides a stable mounting surface for the mechanism. The arm links form a chain of connected members, with pivot points allowing relative movement. Servo horns and linkages convert the angular movement of each servo shaft into movement of the associated joint. At the end of the arm, the gripper fingers form the contact interface with an object. The photograph is used as a visual description of the prototype; exact servo models, dimensions, material grades and payload capacity were not specified in the supplied documents.

For a practical build, the mechanical joints should move freely without excessive looseness. Fasteners should remain secure, wires should be routed away from moving links, and the base should resist tipping during acceleration or gripping. The usable movement range should be limited in software to avoid forcing a linkage against its mechanical stops.

V. SERVO ACTUATION AND CONTROL

A hobby servo typically combines a motor, gear train and positioncontrol electronics. The Arduino Servo library provides functions to attach a servo to a control pin and request a target shaft position. The library documentation describes the use of attach() and write() for this purpose [5]. The requested angle is a setpoint; it should not be mistaken for a direct measurement of the arm's physical position. Closed-loop verification would require suitable external position feedback.

The controller maps each received command to a defined action. For example, a base-rotation command updates the servo associated with the base, a lift command changes the shoulder or linkage position, and a grip command changes the gripper servo position. In a multi-joint system, the software should maintain separate limits and calibration offsets for each actuator.

The Bluetooth path can use a serial profile to carry command characters or numeric values between a paired device and the controller. The HC-05 module documentation describes transparent serial communication through a UART interface [6]. A PC interface can send equivalent commands over a serial connection. In either case, the Arduino parses the incoming command, checks that it is valid and then updates the relevant servo setpoint.

VI. METHODOLOGY

Development is divided into component identification, mechanical assembly, electrical connection, firmware development, interface configuration, calibration and integration. First, the mechanical structure is assembled and the servo motors are mounted at the required joints and gripper. The servo signal connections and Bluetooth module are then connected to the Arduino. The power supply is selected to meet the combined current demand of the actuators.

The microcontroller program receives, interprets and maps movement commands to servo actions. Bluetooth communication and the PC control interface are configured as separate input paths. Individual servo movements are tested before the complete system is integrated. Calibration is performed so that commanded positions correspond to safe and useful physical positions.

The control sequence is: operator selects a movement; command is transmitted through Bluetooth or PC; Arduino receives and validates the command; controller maps it to a servo setpoint; servo moves the relevant joint; and the arm performs the requested action such as rotate, lift, grip or release.



Fig. 3. System working diagram from the supplied project paper.

VII. POWER, CALIBRATION AND SAFETY CONSIDERATIONS

Servo motors can draw substantial current, particularly when several actuators move at the same time or when a joint is loaded. Arduino's servo guidance recommends considering a separate supply when driving more than one or two servos and connecting the external supply ground to the Arduino ground [4]. The actual supply voltage and current rating must be selected from the specifications of the servo models used; these model details were not included in the project material.

Calibration should be carried out one joint at a time. Each servo should first be moved to a conservative centre position, after which the minimum and maximum safe positions can be identified without forcing the linkage. Software limits should be applied before full pick-and-place tests. The Servo library also warns that commanding a servo beyond its mechanical endpoint can produce a high-current condition [5].

During testing, the arm should be operated on a stable surface, with fingers kept clear of linkages and gripper pinch points. Power should be disconnected before changing wiring or adjusting mechanical parts. A simple stop or power-isolation method is recommended for supervised demonstrations. These are design and operating recommendations; the supplied report does not document a formal safety certification.

VIII. TESTING AND RESULTS

Testing is performed in stages. Individual servo movements are checked first, followed by Bluetooth communication and PC command handling. The gripper is tested for opening and closing, and finally the complete system is operated together. The supplied project report identifies six test cases: servo motor testing, Bluetooth testing, PC control testing, gripper testing, pick-and-place testing and complete system testing.

The reported observations indicate that commands received through both Bluetooth and the PC interface were processed by the controller. The servo motors performed the required movements, and the gripper was able to hold and release small objects. These observations support the basic working of the proposed control system.

Potential performance parameters include angular positioning error, command-to-motion response time, gripping capability, Bluetooth reliability, repeatability and power consumption. However, the supplied project documents do not provide numerical measurements for these parameters. Therefore, no numerical accuracy, response-time, range or payload values are claimed here. Future testing should record repeated measurements under stated load and distance conditions.

IX. ADVANTAGES, LIMITATIONS AND FUTURE SCOPE

The main advantages are low-cost construction, educational suitability, dual Bluetooth and PC control, modifiable Arduino control logic and servo-based joint movement. The prototype provides a practical way to understand the relationship between software commands, communication and physical motion.

The design is intended for small loads and basic demonstrations. Movement accuracy depends on servo calibration and mechanical construction. Bluetooth operation is constrained by its practical communication range, and the current system depends on operator commands rather than autonomous decision-making. Advanced sensing and computer vision are not included in the present implementation.

Future development can add sensors for object detection and feedback, a camera and computer-vision module for identifying and locating objects, voice commands as another input method, and AIbased planning for generating movement sequences. A further extension could introduce a feedback controller that compares measured joint position with the target position and corrects deviations.

X. CONCLUSION

The Smart Bluetooth and PC Controlled Robotic Arm demonstrates how a microcontroller, servo motors and communication interfaces can be combined to build a low-cost robotic prototype. The system provides two command paths— Bluetooth and PC control—while using the Arduino as the common motor-control unit. The supplied photograph illustrates a multi-link arm with servo actuators and a gripper, providing a physical basis for the movement and object-handling functions discussed in the paper.

The reported qualitative tests indicate successful command response and basic object handling, while the absence of measured numerical performance data limits quantitative evaluation. The prototype is best positioned as an educational and experimental platform. Its architecture provides a foundation for future work involving sensors, computer vision, voice interfaces and intelligent movement planning.

REFERENCES

[1] S. Dasgupta et al., "Arduino-based Bluetooth-controlled robotic arm study," 2021, as cited in the supplied project seminar material. Full publication details were not present in the supplied documents; verify the original paper before formal submission.

[2] A. Nalwandikar et al., "Mobile-controlled robotic arm using Arduino UNO and servo motors," 2022, as cited in the supplied project seminar material. Full publication details were not present in the supplied documents.

[3] A. Kumar et al., "Bluetooth-controlled robotic arm for wireless object handling," 2023, as cited in the supplied project seminar material. Full publication details were not present in the supplied documents.

[4] Arduino, "Servo Motor Basics with Arduino," Arduino Documentation. https://docs.arduino.cc/learn/electronics/servo-motors/.

[5] Arduino Servo Library, "Servo Library API Documentation," GitHub. https://github.com/arduino-libraries/Servo/blob/master/docs/api.md.

[6] ITead Studio, "HC-05 Bluetooth to Serial Port Module Datasheet." https://components101.com/sites/default/files/component_datasheet/ HC-05%20Datasheet.pdf.

[7] Atal Tinkering Labs, "Bluetooth Module – ATL Equipment Manual," Atal Innovation Mission. https://atl.aim.gov.in/ATL-Equipment-Manual/bluetoothmodule/.

[8] Arduino Project Hub, "Bluetooth Controlled Servo," project example demonstrating smartphone-to-Arduino Bluetooth servo control. https://projecthub.arduino.cc/JANAK13/bluetooth-controlled-servo-5e6688.

[9] J. J. Craig, Introduction to Robotics: Mechanics and Control, Pearson Education. (General reference for manipulator mechanics and control.)

[10] M. W. Spong, S. Hutchinson, and M. Vidyasagar, Robot Modeling and Control, Wiley. (General reference for robotic kinematics and control.)

[11] Arduino, Arduino UNO Rev3 – Hardware Documentation. https://docs.arduino.cc/hardware/uno-rev3/.

[12] Arduino, SoftwareSerial Library documentation. https://docs.arduino.cc/learn/built-in-libraries/software-serial/.

[13] Arduino, Serial – Communication Functions. https://docs.arduino.cc/language-reference/en/functions/communication/serial/.

[14] Bluetooth SIG, Bluetooth Technology Overview. https://www.bluetooth.com/learn-about-bluetooth/tech-overview/.

[15] Arduino Project Hub, "Servo Controller Using Arduino Bluetooth Module." https://projecthub.arduino.cc/digitalmonk12/servo-controller-using-arduinobluetooth-module-ea4222.

ADDITIONAL EXPERIMENTAL GUIDANCE

A repeatable evaluation should record the commanded joint position, observed final position, time between command transmission and the beginning of movement, and whether the gripper retains the test object. Repeat each test under the same load and starting position. Report the average and spread of repeated observations. For Bluetooth tests, record the connection state and test environment instead of stating a range without measurement.

The photograph shows the linkages, servo actuators and gripper, but it does not establish the exact servo model, torque rating, material specification or maximum payload. Add these details from component labels or datasheets before evaluating the prototype under a defined load.

Source and scope note: Project-specific descriptions and qualitative test observations are based on the uploaded research paper and project documents. The prototype photograph was supplied with this request. External references [4]–[8] were added for technical background. References [1]–[3] are retained from the supplied literature survey but require bibliographic verification because complete publication details were not provided.