The Design and Implementation of a Robotic Arm Digital Twin System Based on ESP32

Li Zheng, Mo Chen*

Geely University of China

Received: July 31, 2025

Revised: August 12, 2025

Accepted: August 12, 2025

Published online: August 23, 2025

To appear in: *International Journal of Advanced AI Applications,* Vol. 1, No. 6 (October 2025)

* Corresponding Author: Mo Chen (993287003@qq.com)

Abstract. This paper designs and implements an ESP32-based digital twin system for a robotic arm. The system innovatively utilizes digital twin technology to achieve real-time monitoring, remote control, and virtual-real synchronization of the robotic arm through bidirectional mapping and real-time interaction between the virtual and real worlds. It aims to address the issues of complex operation, high safety risks, and expensive costs associated with traditional industrial robot teaching. The hardware component of the system is designed based on a 1:7 scaled-down version of the ABB IRB-460 robotic arm, with non-standard parts manufactured using 3D printing technology. ESP32 is employed as the control core, replacing the traditional high-cost industrial robot system control cabinet. In the software component, this paper not only constructs a virtual robotic arm model and realizes three-dimensional visualization through Unity3D but also specifically develops an APP-end teaching pendant. This teaching pendant not only supports users in controlling the robotic arm through a UI control panel but also introduces ChatGPT technology to enable remote control of the robotic arm via voice commands. Experimental results demonstrate that the system possesses high virtual-real consistency and operational stability, effectively lowering the learning threshold and significantly enhancing students' understanding and practical abilities in industrial robot operations. It provides an efficient and innovative solution for industrial robot teaching.

Keywords: *Digital Twin Technology; Robotic Arm Control System; ESP32; 3D Visualization; Voice Control.*

1. Introduction

With the in-depth implementation of the "Made in China 2025" and "Industry 4.0" strategies, the manufacturing industry is advancing towards intelligent and digital transformation [1]. As an important carrier of automation and intelligence, industrial robots play an irreplaceable role in improving production efficiency, reducing labor costs, and optimizing processes, and will exert an even more crucial role in future industrial development [2]. A study indicates that a 10% increase in industrial robot density leads to a 1.1% increase in high-tech jobs [3], implying that mastering industrial robot technology not only enhances students' engineering practical abilities but also boosts their employment competitiveness. Currently, there are numerous challenges in industrial robot education.

Traditional industrial robot teaching relies on manual teaching [4], programming teaching, and offline programming task planning tools such as Robot Studio [5], which involve complex operational procedures and pose certain safety risks, thereby raising the threshold for learning and operation [6]. Furthermore, the teaching pendant interface is complex, lacking in real-time and interactive capabilities, making it difficult for students to intuitively grasp the working principles and key operational steps of the robot [7]. With high-end robot training platforms priced at over 600,000 yuan, the limited procurement quantity by educational institutions further restricts opportunities for practical operation, significantly reducing students' interest and practical effectiveness [8].

Digital twin technology, through bidirectional mapping and real-time interaction between the virtual and physical worlds, enables dynamic simulation, data synchronization, and remote control of physical systems [9], providing a new technical approach for industrial robot teaching. To this end, a robotic arm control system based on digital twin technology has been designed and developed. This system employs an ESP32 to replace the traditional high-cost industrial robot system control cabinet, reducing overall hardware costs while satisfying certain computing and communication capabilities, making it suitable for small and medium-sized application scenarios with limited resources [10]. The system is equipped with multi-degree-of-freedom motion control capabilities, integrating virtual simulation, real-time communication, and 3D visualization technologies to achieve real-time monitoring, remote control, and synchronization between the virtual and physical worlds. With the aid of this control system, students can efficiently monitor and operate the robotic arm via a computer without relying on traditional teaching pendants, effectively lowering the learning threshold and enhancing their understanding and practical abilities in operating industrial robots. This provides an efficient and innovative solution for industrial robot teaching.

2. Overall design of the system

2.1. Design and implementation of robot arm hardware

This design is based on the ABB IRB-460 robotic arm [11], which has been scaled down in a 1:7 ratio. The IRB-460 is a six-degree-of-freedom industrial robotic arm with four active axes [12], as shown in Figure 1. The first three degrees of freedom (A, B, C) are driven by three MG996 servo motors, which provide high torque and load capacity, ensuring the robotic arm can perform basic rotational movements. The TPC axis is controlled by an additional MG90S servo motor, which is small in size and has a fast response time, making it suitable for more complex operations or adjustments.

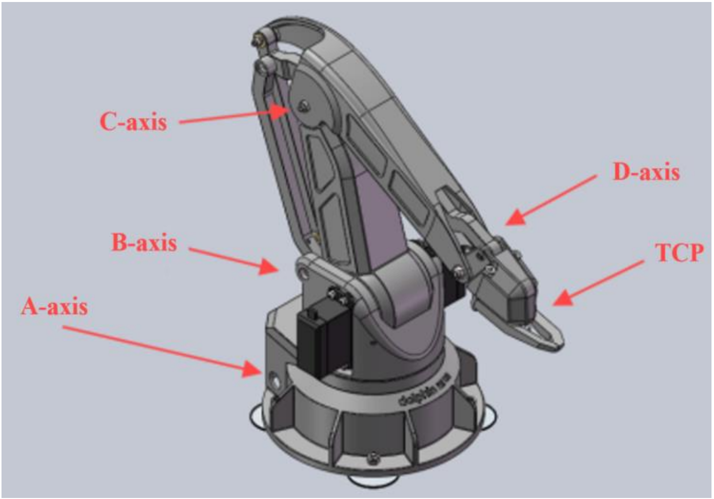


Figure 1. Schematic diagram of the position of each axis of ABB IRB-460 robotic arm.

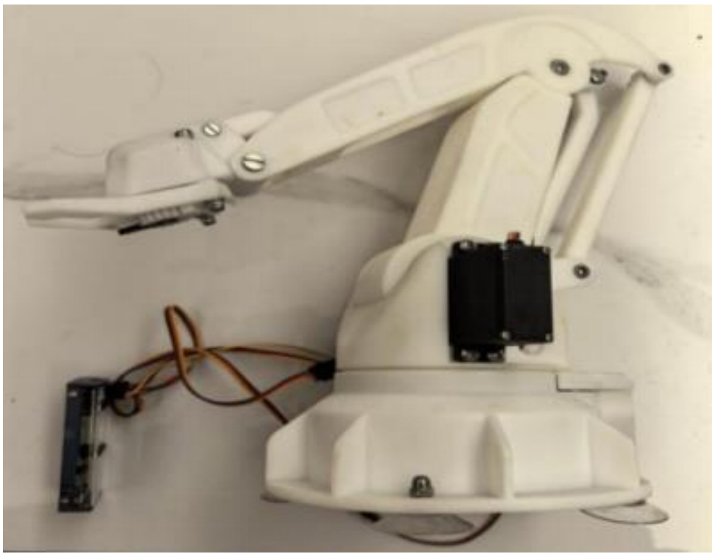


Figure 2. Schematic diagram of assembling a realistic model of a robotic arm.

The structural components of the robotic arm utilize 3D printing technology. This design approach not only ensures efficient manufacturing but also reduces production costs. According to the published ABB IRB-460 industrial technical guide, the 3D model of the robotic arm comprises 18 parts and 4 servo motors. Non-standard components such as the robotic arm can be designed using SolidWorks industrial modeling software, scaled down to a 1:7 ratio, and then manufactured using lightweight 3D printing technology. The actual assembly effect is shown in Figure 2.

2.2. Overall framework of system design

Based on the five-dimensional model theory of digital twins [13], a platform framework for the ESP32-based robotic arm digital twin system is designed, encompassing the physical layer, data layer, functional layer, and application layer.

The physical layer includes mechanical structures (such as links, joints, end effectors, etc.), control systems, drive systems, and sensor modules.

The data layer is responsible for data management and interaction between the physical layer and the virtual layer. Leveraging the Wi-Fi module of the ESP32, the system can transmit real-time data to the control end, ensuring synchronized operation between the physical and virtual worlds.

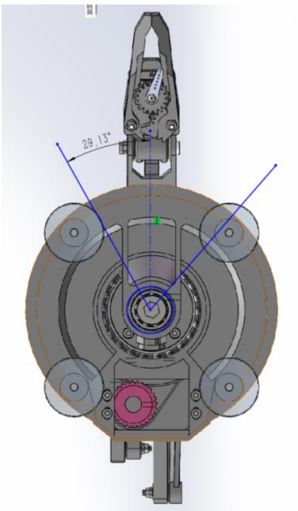
The virtual layer establishes a virtual robotic arm model based on physical layer data. Driven by the data layer, it accurately maps the operational status of the physical system, enabling functions such as motion simulation, collision detection, and task optimization.

The application layer realizes functions such as remote control through a human-machine interaction interface, employing a multimodal interaction design to capture user intentions and provide feedback on the system's status. This module integrates a speech recognition engine with a graphical interface, enabling users to input control commands through natural language or visual operations.

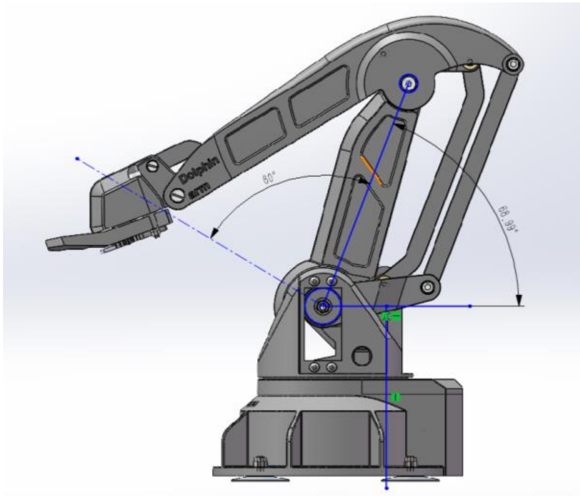
3. Key technology

3.1. Construction and implementation of virtual models

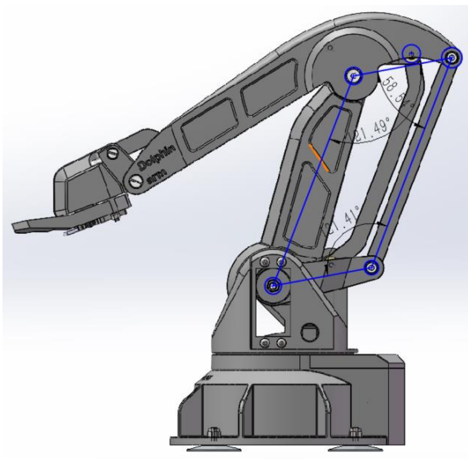
Based on the simulation and finite element analysis conducted with the industrial modeling software SolidWorks [14], the ranges of motion for the four active axes and the TCP can be calculated, as shown in Figure 3.



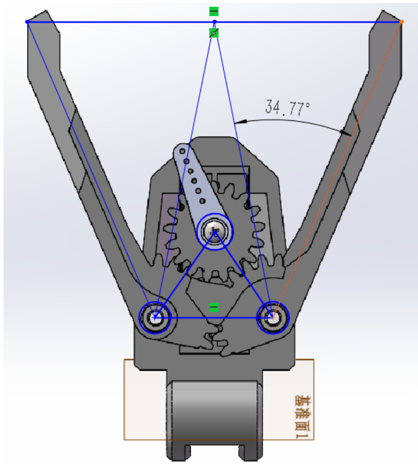
(a) the angle between A-axis and base



(b) up and down movement angle of B-axis



(c) C-axis and D-axis angles



(d)Claw Rotation Angle

Figure 3. Schematic diagram of the angle between the activity axis.

The angular range for the rotation of Axis A with the base is -35° to 35°. The angular range for the upper arm of Axis B is 0° to 80°. The angular range for the forearm of Axis C is 55° to 180°. The angular range for the gripper of Axis D is 0° to 37°.

3.2. System control

3.2.1. Design and implementation of PCB board

Since the robotic arm's movable mechanism is powered by four servo motors, the design of the PCB control board primarily focuses on controlling and driving these four servo motors. Therefore, the PCB control board consists of eight modules: the ESP32 control module, an expanded IO module, a TYPE-C power supply module, a TYPE-C debugging module, a USB-to-TTL module, an LED light wire module, an LDO voltage regulator module, and a debugging light module. The complete physical diagram of the PCB is shown in Figure 4.

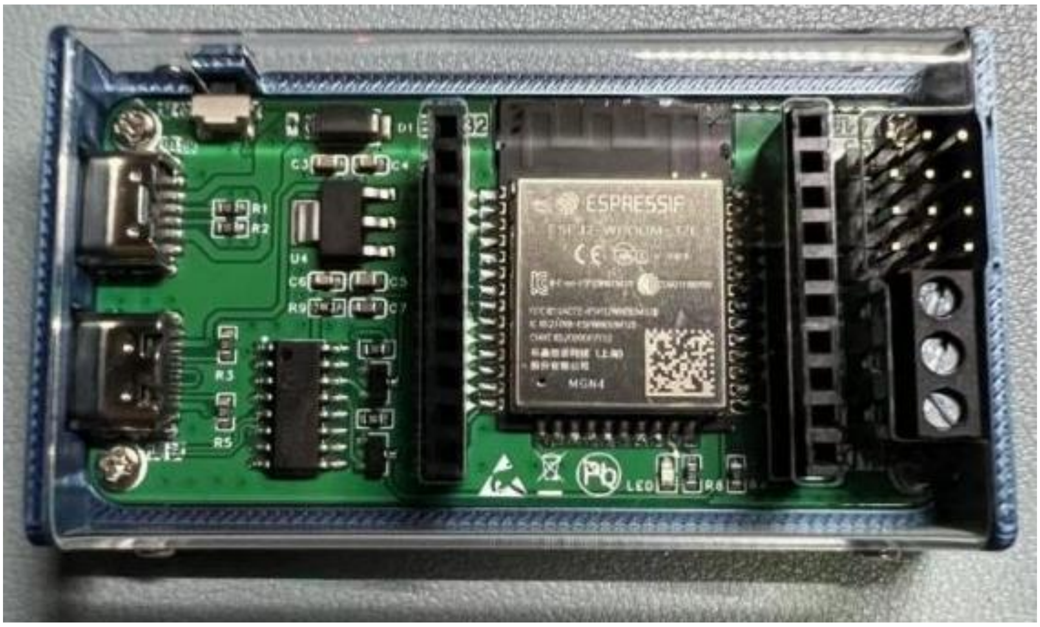


Figure 4. Complete PCB physical image.

3.2.2. Multi servo control

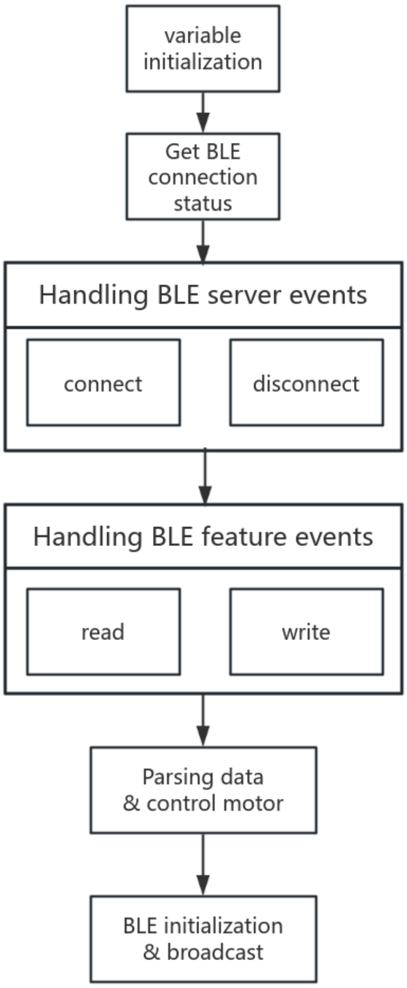


Figure 5. Logic flowchart of control card program.

The main logic of the control board's program is the ESP32's code for controlling multiple servo motors. After initializing and defining the device name, it starts broadcasting, creates a BLE server and service, and binds callback functions to handle operations such as connection, disconnection, and characteristic manipulation. This allows the teaching pendant to discover and connect to the ESP32 via Bluetooth, send and receive commands, and control multiple servo motors to perform corresponding operations. The specific logic of the control board's program is shown in Figure 5.

3.3. Interface of Flex Pendant

This interface features a UI control panel that allows users to adjust the angles of the robotic arm. Bluetooth communication is used to send control commands to the ESP32 controller, and Unity3D visualization enables users to intuitively view changes in the robotic arm's angles.

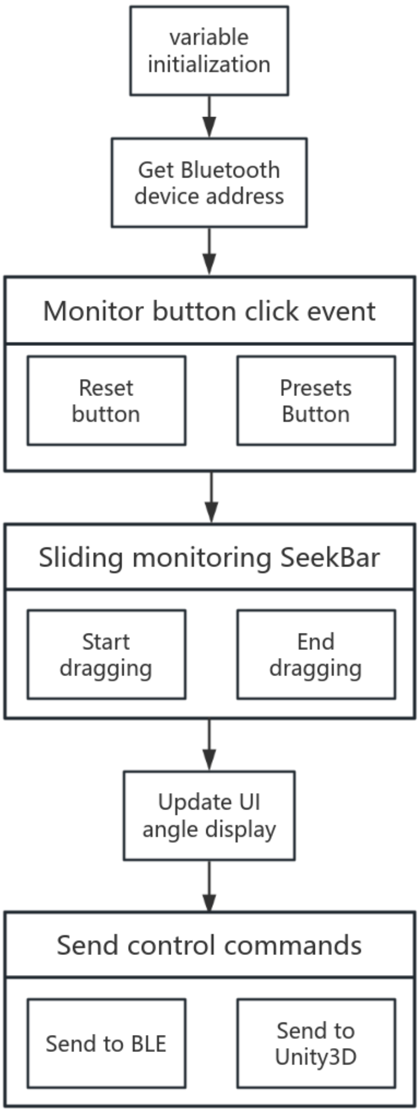


Figure 6. Logic Flow Diagram of AAP Flex Pendant Backend.

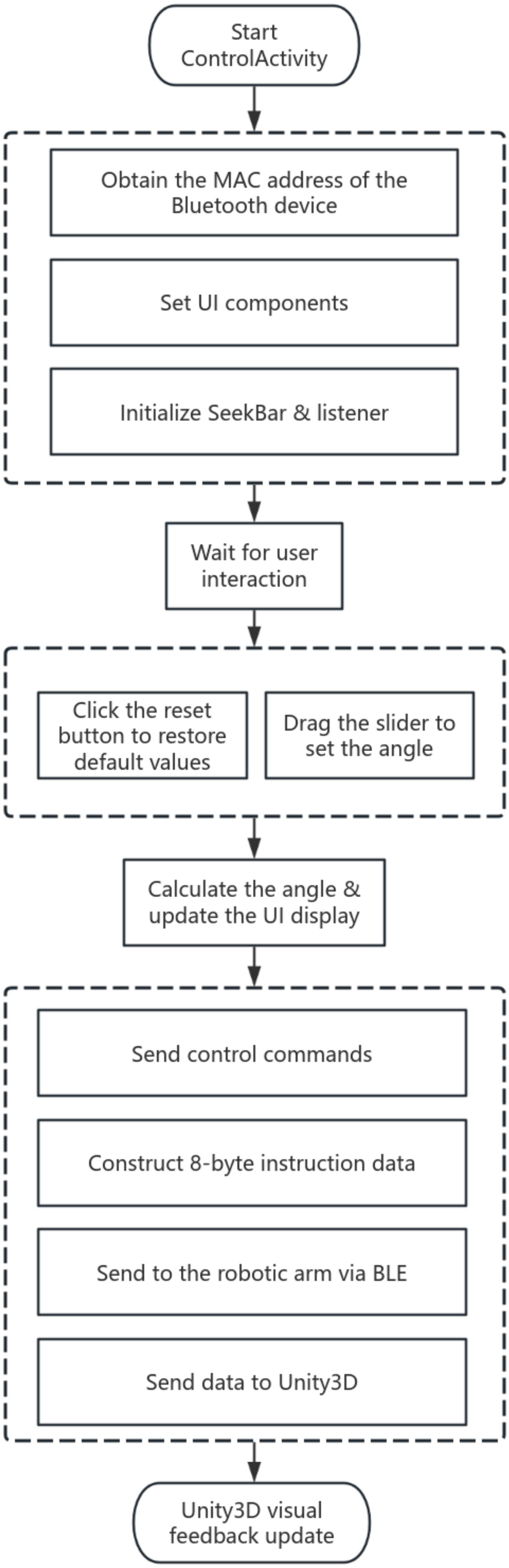


Figure 7. Logic Flow Diagram of AAP Flex Pendant Frontend.

After initializing and displaying the interface layout, the device's Bluetooth address is obtained and displayed on the UI. The maximum and initial values of the five sliders are set, and listeners are configured for each slider. When a slider detects that it has stopped being moved, it updates the angle display, sends the corresponding control command to the respective servo motor hardware, and simultaneously sends the servo motor angle value to the Dolphin Arm model in the Unity3D scene to synchronize the movement state of the virtual robotic arm [15]. The specific code logic is shown in Figure 6.

The robotic arm control interface is primarily used to adjust the angles of each joint of the robotic arm and send commands to the robotic arm control system via Bluetooth communication, while simultaneously synchronizing the data to Unity3D for visualization. The specific code logic is shown in Figure 7.

To enable users to control the robotic arm through voice commands on a mobile app, the voice signal must first be converted into text format. Considering that the ChatGPT API only supports text input, this paper employs speech recognition technology using Google's open-source Speech-to-Text API to convert users' voice commands into text. On the mobile app side, after the user presses the voice input button, the app captures the voice signal and converts it into text using speech recognition technology. This text is then sent to a local computer for processing. On the local computer, we invoke the ChatGPT API to process the converted text. To ensure that ChatGPT can accurately understand the user's intentions and generate the corresponding robotic arm control code, a series of prompts tailored for robotic arm control are designed. These prompts clearly specify the tasks that ChatGPT needs to execute, namely, invoking the movement execution code of the robotic arm API based on the text instructions input by the user.

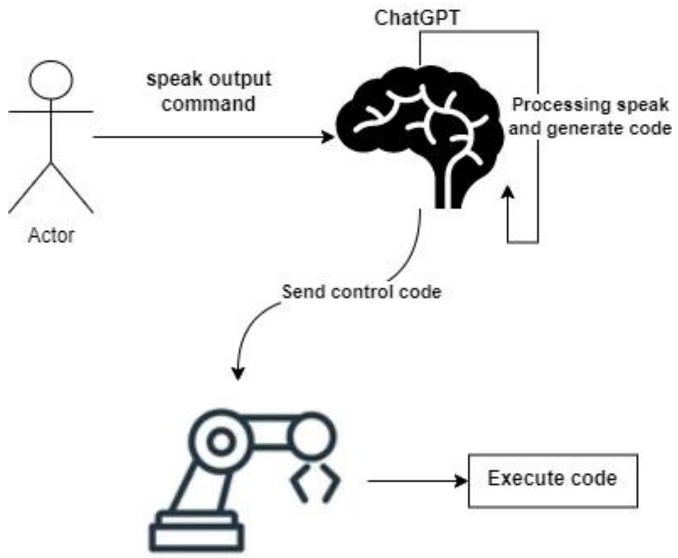


Figure 8. Schematic diagram of implementing voice control using ChatGPT.

3.4. Digital twin implementation

The digital twin effect is mainly achieved using an industrial suite developed by Unity China, which is based on the Unity engine. The overall digital model with motion limit parameters for each axis, obtained through kinematic simulation of the SolidWorks robotic arm, is converted into 3ds or fbx format and then imported into Unity3D, as shown in Figure 8. The trigger code is written in the built-in development environment of Unity3D, which listens to serial port data and updates the position of each axis of the robotic arm. This method can be used to monitor data from the robotic arm's control board. When the teaching pendant APP sends control commands to the robotic arm, the robotic arm simultaneously sends the commands to the C# program in Unity3D via the serial port. After the program parses the commands, it controls the movement of the model.

Figure 9. Unity3D running effect diagram.

This code serves as the rotation control script for the robotic arm, with core functions including controlling the rotation of four joints, setting angles through sliders, receiving external data via serial port to adjust angles, using threads to continuously monitor serial port data, updating rotation speed and angles to ensure stable movement of each joint. Each joint has independent parameters such as rotation speed, maximum/minimum angles, and current angle, which are used as global angle offset values to correct angle readings. The serial port communication is initialized, and threads are started to receive data. Event listeners are bound to the sliders to update the target angles of each joint based on the slider values.

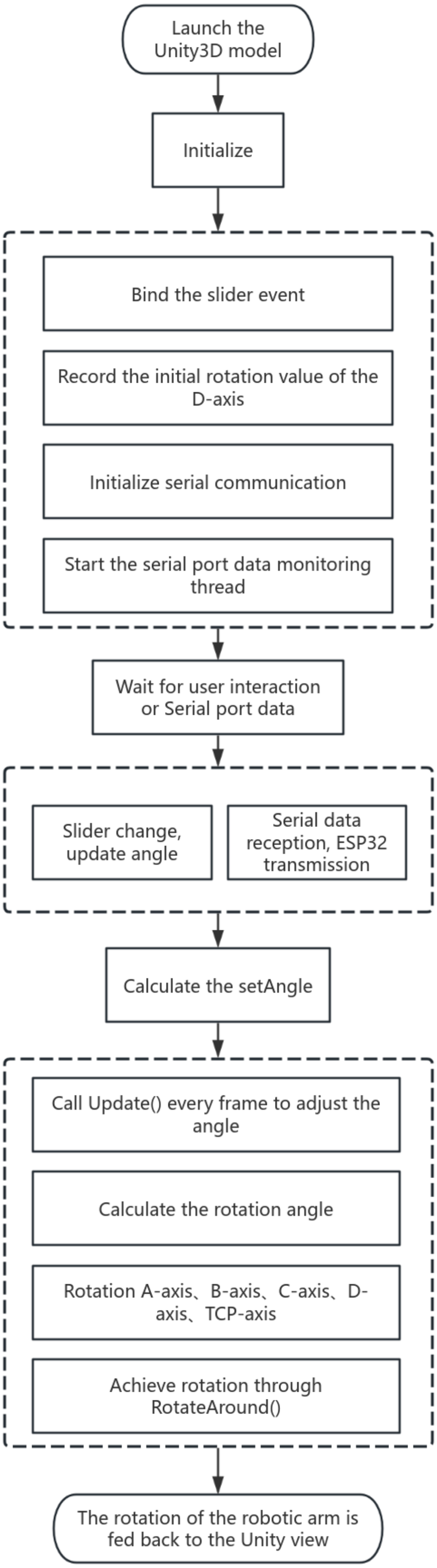


Figure 10. Logic flow diagram of Unity3D code driving logic.

When the serial port data reading thread parses the received data packets and converts them into actual angles, it updates the target angles of each joint. The joint angles can also be manually specified by parsing string commands and setting the target angles. The rotation of each axis is controlled through frame-by-frame updates. If the current angle is more than 2 degrees greater than the set angle, the rotation speed is set to -30 (reverse rotation); if it is less than 2 degrees, the rotation speed is set to 30 (forward rotation); otherwise, the rotation speed is zeroed (stabilized at the target angle). The specific Unity3D code logic is shown in Figure 10.

4. Testing

4.1. Function testing

Firstly, we tested whether the teaching pendant APP could run and meet the design requirements. The engineering files of the teaching pendant APP were generated into an APK file that could be installed on the Android system, and then installed and run.

For the state synchronization between the 3D digital model and the physical model, we first started the ArmRoteta file in the Unity Hub 3D digital model project to enable data monitoring. When the "Search Device" button was clicked, the robotic arm could be normally searched, identified, and paired. If connected to the robotic arm at this time, the robotic arm would automatically return to its initial state, indicating that the entire system was ready for operation. The specific operational effect is shown in Figure 11.

Figure 11. Rendering of multi-device connection actual operation (robotic arm initial state 1).

When the teaching pendant APP sends an operational information chain to the ESP32, the robotic arm can receive and execute the instructions. At the same time, the digital model also receives the information chain synchronously, and the digital model mimics the posture of the physical model (robotic arm). The operational effect is shown in Figure 12, where the robotic arm is in (State 2). Therefore, based on the above tests, the overall function of the system basically meets the expected design requirements.

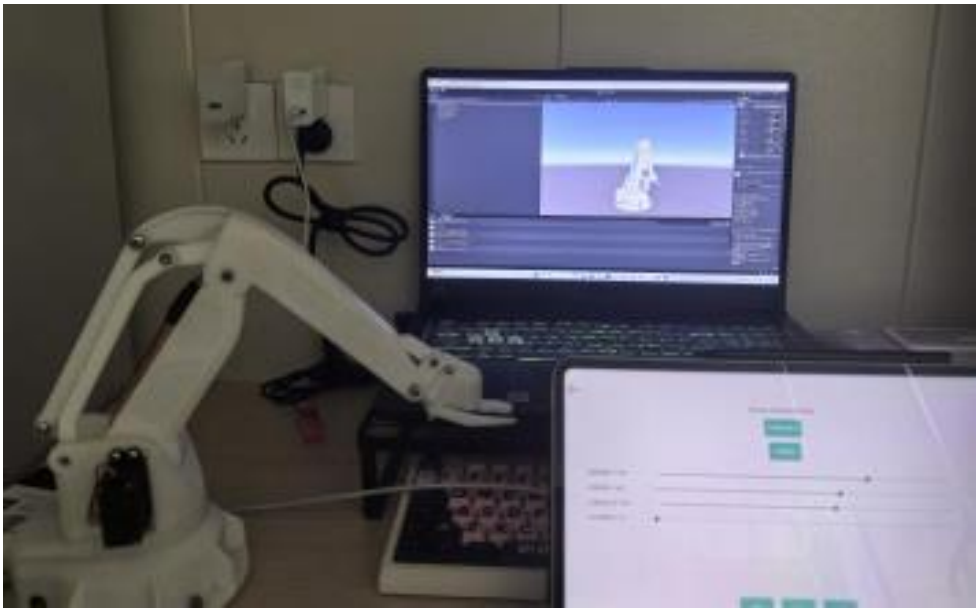


Figure 12. Rendering of multi-device connection actual operation (robotic arm operating state 2).

4.2. Error analysis

To verify the consistency of the synchronization between the virtual model and the actual robotic arm movements in the system, a comparative test of the servo angles in typical postures was conducted. The error analysis results are shown in Table 1.

Table 1. Analysis of virtual-real mapping errors.

Joint name	Commanded angle	Actual angle	Virtual angle	Virtual-real error	Description
A-axis	90	91.2	89.6	1.6	Well-synchronized
B-axis	45	44.1	46	1.9	Within the acceptable range
C-axis	120	119.5	34	1.2	High precision
D-axis	30	31.4	30.2	1.2	High precision

4.3. Performance testing

Multiple tests were designed to check serial port data throughput, response latency, and low-speed versus high-speed rotation. The experimental conclusions are presented in Table 2. Therefore, it is recommended to adjust the serial port data refresh interval to avoid frame loss due to excessively fast data at 10ms. By using an interpolation algorithm, the jitter issue during high-speed rotation can be reduced. The final conclusion is that the system functions normally, and it is suggested to operate at a refresh rate of 50ms to 100ms to avoid frame loss under high load.

Table 2. Summary of Performance Testing.

Test category	Result
Voice command response time	Normal, average response time 1300ms
UI slider	Normal, response time 50ms
Serial port data reception	An interval of 50ms~100ms is optimal
Robotic arm rotation	Smooth with no rotational jumps

5. Conclusion

This study designed and implemented an ESP32-based robotic arm digital twin system by integrating sensor data with physical models, thereby constructing an efficient and stable robotic arm monitoring platform that combines digital twin models with embedded systems. Innovatively, the platform incorporates ChatGPT technology for voice command parsing. Through virtual-real synchronization technology, it not only effectively lowers the learning threshold for industrial robotics education but also significantly enhances the system's interactivity and real-time performance. Users can interact with the system using natural language, and the system utilizes ChatGPT to convert voice commands into robotic arm control instructions, enabling a more intuitive and convenient operational experience, thus improving the intuitiveness and interactivity of the teaching process.

Experimental results demonstrate that the system possesses high virtual-real consistency and operational stability, accurately reflecting the motion state of the physical robotic arm. Particularly in terms of voice control, the system exhibits excellent response speed and accuracy, validating the effectiveness of ChatGPT technology in industrial robotics education applications. This system provides an innovative tool for robotic arm experimental teaching, helps enhance students' engineering practical abilities and system understanding capabilities, showcases good teaching application value, and offers new insights for the intelligent development of future industrial robotics education.

References

[1] Li, Bh., Hou, Bc., Yu, Wt. et al. (2017). Applications of artificial intelligence in intelligent manufacturing: a review. Frontiers Inf Technol Electronic Eng 18, 86–96. https://doi.org/10.1631/FITEE.1601885

[2] Ritchie, E. ., & Landis, E. A. (2021). Industrial Robotics in Manufacturing . Journal of Leadership, Accountability and Ethics, 18(2). https://doi.org/ 10.33423/jlae.v18i2.4258

[3] Zhao, Yantong, Said, Rusmawati, Ismail, Normaz Wana, Hamzah, Hanny Zurina. (2022). Effect of Industrial Robots on Employment in China: An Industry Level Analysis, Computational Intelligence and Neuroscience, 13. https://doi.org/10. 1155/2022/2267237

[4] Valeria Villani, Fabio Pini, Francesco Leali, Cristian Secchi. (2018).Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications, Mechatronics, 55, 248-266, https://doi.org/10.1016/j.mechatronics. 2018.02.009.

[5] Kaczmarek, W., Lotys, B., Borys, S., Laskowski, D., & Lubkowski, P. (2021). Controlling an Industrial Robot Using a Graphic Tablet in Offline and Online Mode. Sensors, 21(7), 2439. https://doi.org/10.3390/s21072439

[6] Liu, Y., Kukkar, A. & Shah, M. (2022). Study of industrial interactive design system based on virtual reality teaching technology in industrial robot. Paladyn, Journal of Behavioral Robotics, 13(1), 45-55. https://doi.org/10.1515/pjbr-2022 -0004

[7] Chen, X. (2023). Bilingual course of industry robot basing digital twin technology. Journal of Education and Educational Research, 3(1), 29-31. https://doi.org/10.54097/jeer.v3i1.8059

[8] Huang, Y., Wu, X., Yang, B., Li, Z. (2020). The Application of Industrial Robot Technology in the Teaching of Applied Undergraduate Colleges and Universities. In: Abawajy, J., Choo, KK., Islam, R., Xu, Z., Atiquzzaman, M. (eds) International Conference on Applications and Techniques in Cyber Intelligence ATCI 2019. ATCI 2019. Advances in Intelligent Systems and Computing, vol 1017. Springer, Cham. https://doi.org/10.1007/978-3-030- 25128-4_3

[9] Garg, G., Kuts, V., & Anbarjafari, G. (2021). Digital Twin for FANUC Robots: Industrial Robot Programming and Simulation Using Virtual Reality. Sustainability, 13(18), 10336. https://doi.org/10.3390/su131810336

[10] Zhang, Y., & Zhang, L. (2024). Research on Education Robot Control System Based on ESP32. Journal of Education and Educational Research, 7(2), 299-302. https://doi.org/10.54097/3x86qp78

[11] Domènech, G.F. (2016). Construcció d'un micro braç articulat: Part mecànica.

[12] Lundqvist, R., & Söreling, T. (2005). New Interface for Rapid Feedback Control on ABB-Robots.

[13] Khadiri, H., Sekkat, S., & Herrou, B. (2023). Digital Twin Based SUDIHA Architecture to Smart Shopfloor Scheduling. Journal of Manufacturing and Materials Processing, 7(3), 84. https://doi.org/10.3390/jmmp7030084

[14] Hindroyuwono, R., Marvianto, R. A., Almaida, E. F., & Arafat, Y. . B. (2024). Static Loading Simulation on Temporary Platform and Ladder Absorbent Chamber Design. International Journal of Engineering Continuity, 3(1), 118–125. https://doi.org/10.58291/ijec.v3i1.247

[15] Khan, A. (2021). Study of an at-home upper-extremity stroke rehabilitation system (Master's thesis, University of Calgary, Calgary, Canada). Retrieved from https://prism.ucalgary.ca.

[16] He, B., Bai, KJ. (2021). Digital twin-based sustainable intelligent manufacturing: a review. Adv. Manuf,9, 1–21 .https://doi.org/10.1007/s40436-020-00302 -5

[17] Sun, X., Zhang, F., Wang, J. et al.(2025). Digital twin for smart manufacturing equipment: modeling and applications. Int J Adv Manuf Technol 137, 4929–4946.https://doi.org/10.1007/s00170-025-15468-0

[18] Wang, C., Liu, X., Feng, F. et al.(2025). A modeling method of digital twin connection model based on virtual sensor. J Intell Manuf. https://doi.org/10. 1007/s10845-025-02639-9

[19] Craig, J.J. (1986). Introduction to Robotics Mechanics and Control.