

# Research on Education Robot Control System Based on ESP32

Yan Zhang \*, Lei Zhang

School of Information Science and Technology, Taishan University, Taian, Shandong 271021, China

\* Corresponding author: Yan Zhang (Email: xtzy@163.com)

---

**Abstract:** This study presents a comprehensive overview of the current state of educational robot control systems, examining their integration and impact in educational settings. The findings reveal the potential of these systems in enhancing learning experiences, motivating student engagement, and facilitating personalized education. The research highlights key benefits such as making abstract concepts accessible, providing interactive learning experiences, and bridging the gap in distance education. However, limitations in cost and accessibility, as well as the need for teacher training, are identified as challenges that need to be addressed. Looking ahead, the integration of artificial intelligence, machine learning, and immersive technologies like virtual and augmented reality are expected to further transform educational robot control systems, opening new possibilities for personalized and engaging learning experiences.

**Keywords:** Education Robot; ESP32; Control System.

---

## 1. Introduction

The development of educational robots has witnessed significant progress in recent years, as they have become increasingly prevalent in classrooms, laboratories, and even homes [1-2]. These robots are designed to engage students in interactive learning experiences, enhancing their understanding of complex concepts and fostering a passion for science, technology, engineering, and mathematics (STEM) subjects [3]. As a result, educational robots have become a valuable tool for supporting education innovation and enhancing student engagement.

In the realm of robot control, the ESP32 has emerged as a promising hardware platform. This low-cost, high-performance microcontroller offers an array of advanced features, including Wi-Fi and Bluetooth capabilities, making it suitable for a wide range of applications, including educational robots. The ESP32's ease of programming, coupled with its powerful processing capabilities and extensive peripheral interfaces, makes it an ideal choice for controlling educational robots.

The aim of this study is to investigate the development of an educational robot control system based on the ESP32. Specifically, we aim to design and implement a control system that leverages the ESP32's capabilities to enable robust, reliable, and cost-effective robot control. Furthermore, we aim to evaluate the cost and performance of the control system and compare it with other existing control systems.

The significance of this study lies in its potential to contribute to the advancement of educational robotics. By developing an optimized control system based on the ESP32, we aim to provide educators and students with a powerful tool for enhancing learning experiences and fostering a deeper understanding of STEM principles [5]. Additionally, this study could pave the way for future research in the field of educational robotics, leading to the development of more innovative and effective robotic teaching aids.

## 2. Introduction to the ESP32 Hardware Platform

The ESP32 is a low-cost, high-performance microcontroller that has gained popularity in various applications, particularly in the realm of robotics and Internet of Things (IoT) devices. Its popularity is attributed to its unique combination of features and performance parameters that make it an ideal choice for resource-constrained systems.

The ESP32 chip is built on the Tensilica LX6 microprocessor architecture, which provides excellent processing capabilities. It operates at a clock speed of up to 240 MHz and features a dual-core processor configuration, enabling parallel processing and improved system efficiency. Additionally, the ESP32 boasts a rich set of peripherals, including Wi-Fi and Bluetooth radios, sensors, and communication interfaces, all integrated into a compact package [6].

The ESP32's hardware architecture consists of a main processor complex, memory subsystems, and various peripherals. The processor complex includes two independent cores, each with its own dedicated instruction cache and data cache. The memory subsystems comprise internal SRAM and external flash memory, providing ample storage for program code and data. The peripherals include a wide range of interfaces such as GPIOs, UARTs, SPI, I2C, and PWM, enabling the ESP32 to interface with various sensors and actuators.

The ESP32's Wi-Fi and Bluetooth capabilities are particularly noteworthy. It supports both 2.4 GHz and 5 GHz Wi-Fi bands, allowing for high-speed wireless communication. The Bluetooth functionality supports both Classic Bluetooth and Bluetooth Low Energy (BLE), making the ESP32 suitable for a wide range of IoT applications. The ESP32's powerful processing capabilities, rich set of peripherals, and extensive wireless communication options make it an ideal choice for developing educational robot control systems.

In terms of development environment and programming

language, the ESP32 is supported by a comprehensive ecosystem of tools and software libraries. The official development framework for the ESP32 is the ESP-IDF (Espressif IoT Development Framework), which provides a comprehensive set of APIs and libraries for developing and deploying applications on the ESP32. The ESP-IDF supports multiple programming languages, including C, C++, and MicroPython[7], allowing developers to choose the language that best suits their needs and expertise. In addition, ESP32 can communicate with other programming environments or languages through network protocols, such as integrating JavaScript or establishing a twin digital system with CompeliaSim.

The cost-effectiveness and ease of programming of ESP32 further enhance its appeal for educational applications, where affordability and accessibility are key considerations.

### 3. Design of Educational Robot Control System

The design of an educational robot control system requires careful consideration of hardware and software components to ensure robust and effective control of the robot. The entire system architecture should be modular and scalable to accommodate future expansion and modifications.

The core of this system is the ESP32 microcontroller, which acts as the robot's brain, responsible for receiving commands from users, processing them, and sending appropriate signals to the robot's actuators. The system architecture includes various modules, including sensors, actuators, power management, and communication interfaces, all of which are interconnected through GPIO, serial buses, or wireless networks. The schematic diagram of the main control circuit is shown in Figure 1.

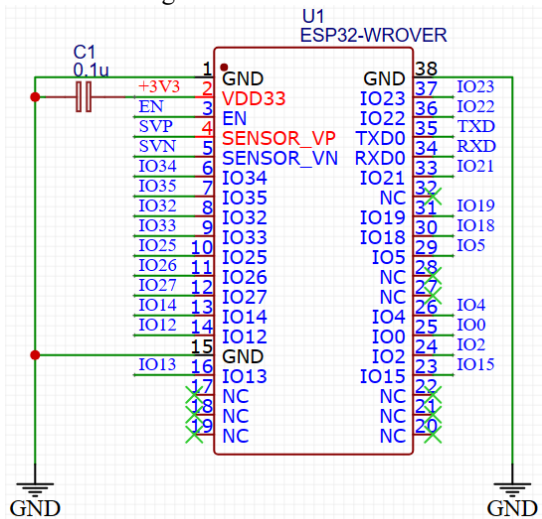


Figure 1. Microcontroller circuit schematic

The control system hardware consists of an ESP32 board, sensors, actuators, and power supply. Sensors, such as light sensors, microphones, ultrasonic distance sensors, or infrared sensors, are connected to the analog or digital input pins of ESP32 to capture environmental data. Actuators, such as motors, servos, and servo systems, interface with the PWM or GPIO pins of ESP32 through the driving chip to achieve precise control of robot motion. The power supply provides the necessary voltage and current to power the entire system. For example, the circuit design of the power supply is shown in Figure 2.

In terms of software architecture, the control system is designed using a modular approach. The main program runs on ESP32 and coordinates various modules to handle tasks such as sensor data acquisition, actuator control, and communication with external devices. This software is written in advanced programming languages such as C++ or MicroPython, utilizing the powerful processing power and rich library support of ESP32. Some programs may require higher performance CPU processing. These programs can be run in the form of apps or web apps on a PC or phone, and then communicate with ESP32 through WIFI or Bluetooth. For example, Figure 3 shows an example of programming control using MicroPython directly. Figure 4 shows an example of using Coppeliassim to construct a PC interface for a twin digital robot system [8].

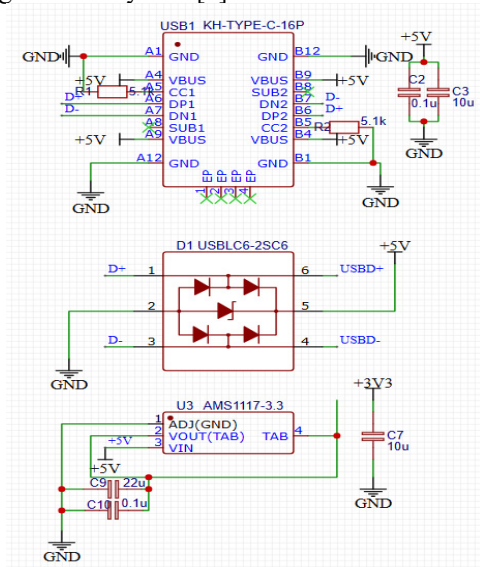


Figure 2. USB power supply schematic

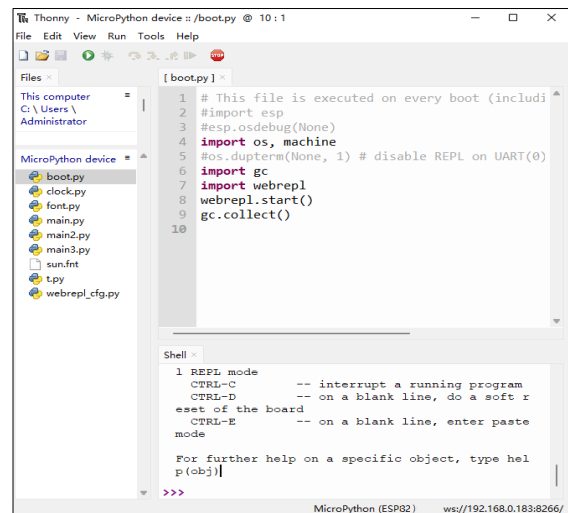


Figure 3. MicroPython on ESP32

The design philosophy behind the control system emphasizes simplicity, modularity, and scalability. The structure of the code makes it easy to understand, maintain, and modify. Each module is designed to perform specific tasks and can be easily replaced or upgraded without affecting other parts of the system. With the development of robot capabilities, this modular approach helps to integrate new features and functions.

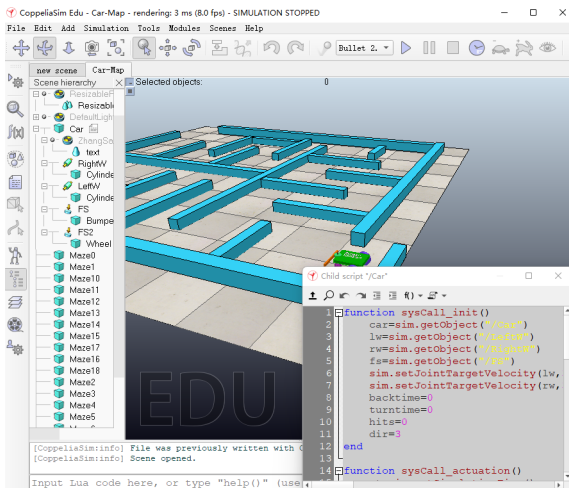


Figure 4. Digital twin based on Coppeliasim and ESP32

## 4. Research on Robot Control Algorithms Based on ESP32

### 4.1. Real-Time Control System Implementation based on ESP32

To implement real-time control of the robot using ESP32, it is necessary to consider factors such as processing speed, memory limitations, and the availability of suitable libraries or frameworks. The ESP32, with its dual-core processor and WiFi/Bluetooth capabilities, is an ideal platform for real-time control applications.

The implementation involves writing code that interfaces with the robot's hardware, acquires sensor data, computes control signals using the selected control algorithm, and sends these signals to the actuators. This code must run efficiently on the ESP32 to ensure smooth and responsive robot control. Additionally, the use of interrupt handlers and efficient data structures can help optimize performance and reduce latency.

For more complex real-time control systems, RTOS can be used to manage system resources and ensure timely execution of tasks. FreeRTOS integrates with ESP32 to provide a structured environment for real-time control tasks [9]. The Wi-Fi and Bluetooth features of ESP32 allow for wireless communication with other devices or remote controllers. This enables real-time control and monitoring of robot systems from remote locations, promoting remote learning and experimental setup.

### 4.2. Control System Performance Analysis and Optimization

To analyze and optimize the performance of the ESP32 robot control system using Arduino IDE and MicroPython, the first step is to use the Profiler tool of Arduino IDE to analyze code execution time and identify performance bottlenecks. Subsequently, optimizing the code through the concise syntax and fast execution speed of MicroPython reduces resource consumption [10]. Use the Arduino IDE's serial port monitor to view real-time performance data, such as CPU usage, memory usage, etc. If there is a communication delay issue, the wireless communication protocol can be optimized or the packet size can be reduced. In addition, hardware tools such as oscilloscopes and logic analyzers are used to assist in diagnosis. During the optimization process, continuously iterate testing and validation to ensure the effectiveness of improvement measures. By combining the ease of use of Arduino IDE with

the efficiency of MicroPython, the performance of robot control systems can be quickly improved, ensuring accuracy, stability, and rapid response.

## 5. Experimental Design and Result Analysis

### 5.1. Experimental Platform Setup and Experimental Scheme Design

In order to conduct in-depth research on the performance of robot control systems based on ESP32, an experimental platform was established. This platform includes robot hardware, ESP32 microcontroller, sensors, actuators, and necessary software components. The design focus of the experimental plan is to test the usability, accuracy, responsiveness, stability, and efficiency of the system. The experimental plan includes various tests, including hardware design and programming usability, speed control, transmission control, etc. For each test, specific metrics are defined to evaluate the performance of the system.

### 5.2. Experimental Results Presentation and Analysis

The experimental results were carefully recorded and analyzed. The data is presented in both graphical and digital forms to provide a comprehensive overview of system performance.

Analysis shows that the control system based on ESP32 has high usability, accuracy, and responsiveness. Its built-in wireless communication module makes it easier to build hardware connections and avoids long-term connection establishment and maintenance between devices, improving system efficiency. Its higher operating frequency also makes its response time faster than other systems in the same level. The rich programming methods further enhance the usability of the system.

In addition, the stability and efficiency of the system were evaluated. The system can maintain consistent control for a long time with minimal performance fluctuations. In addition, further optimization of the processing speed and memory usage of ESP32 will enable more efficient operation of related programs.

### 5.3. Comparison and Discussion with Other Control Systems

In order to further verify the performance of the ESP32 based control system, it was compared with other commonly used control systems in the field of robotics, such as STM32, Raspberry Pi, AVR and other series of microcontrollers. These systems range from traditional microcontroller based systems to more advanced systems using machine learning or artificial intelligence algorithms.

The focus of comparison is on key performance indicators such as responsiveness, stability, cost, and efficiency. The ESP32 based system is competitive with other systems and has shown comparable or better performance in several fields. In particular, the system integrates wireless networks and Bluetooth, highlighting its ability to transmit, process sensor data, and calculate control signals in real-time as a key advantage. In addition, its 240MHz main frequency is much higher than traditional MCUs, and its price is also the lowest among MCUs in the same class. This makes the use of ESP32 as the main control system for educational robots highly cost-

effective.

## 6. Application Prospects of Educational Robot Control Systems in the Field of Education

Educational robots hold immense potential in enhancing classroom learning. They can engage students actively through interactive and hands-on experiences, making abstract concepts more tangible. For instance, robots can demonstrate complex scientific principles or historical events visually, aiding in comprehension. Additionally, they foster collaborative learning by encouraging students to work together in teams, enhancing problem-solving skills.

With the rise of remote learning, educational robots can bridge the gap between students and teachers. Robots can serve as virtual teachers, providing personalized instruction and feedback to students remotely. This allows for more flexible learning options, accommodating diverse learning styles and schedules.

Personalized education is another area where educational robots excel. By tracking students' progress and preferences, robots can tailor their instructions to individual needs, enabling differentiated learning. This approach is particularly beneficial for students with special needs or learning disabilities, as robots can provide adaptive learning experiences tailored to their unique capabilities.

Low cost educational robots based on ESP32 control systems will greatly promote these applications.

## 7. Conclusion and Prospects

The integration of ESP32-based control systems into the realm of educational robotics represents a significant milestone in the evolution of teaching and learning. The ESP32, a powerful and cost-effective microcontroller, enables the creation of sophisticated yet accessible robotic systems tailored for educational purposes.

The role of the ESP32 in these control systems is multifaceted. Firstly, it acts as the brain of the robot, processing sensor inputs and executing commands to effectuate desired actions. This allows students to gain a deeper understanding of robotics principles and programming concepts through hands-on experiences. Secondly, the ESP32's Wi-Fi and Bluetooth capabilities enable seamless communication between robots and other devices, expanding the possibilities of interactive and collaborative learning experiences.

Looking ahead, the prospects for ESP32-based educational robot control systems are bright. As technology continues to advance, we can expect even more sophisticated robots and control systems that are tailored for specific educational

objectives. These systems will likely become more accessible, affordable, and user-friendly, making robotics education more inclusive and widespread.

Moreover, the integration of AI and machine learning algorithms with ESP32-based robots will further revolutionize education. Robots will be able to adapt to students' learning styles and provide personalized feedback, enabling more effective and efficient learning.

## Acknowledgments

This work has been supported by the Micro Project on Educational Development Research in Shandong Province, [Grant No. FJ098, Research on the Application of Virtual Simulation Software CoppeliaSim EDU in Robot Course Teaching].

## References

- [1] Atman Uslu NilüferYavuz Gulay Öztüre, and Koçak Usluel Yasemin. A systematic review study on educational robotics and robots. *Interactive Learning Environments* 31. 9(2023): 5874-5898.
- [2] Vehbi Yolcu, and Veysel Demirer. The effects of educational robotics in programming education on students' programming success, computational thinking, and transfer of learning. *Computer Applications in Engineering Education* 31. 6 (2023): 1633-1647.
- [3] Nam Benjamin H., and Bai Qiong. ChatGPT and its ethical implications for STEM research and higher education: a media discourse analysis. *International Journal of STEM Education* 10.1(2023).
- [4] Espressif Systems. Available: <https://www.espressif.com/>.
- [5] Smith Kathy, et al. Principles of Problem-Based Learning (PBL) in STEM Education: Using Expert Wisdom and Research to Frame Educational Practice. *Education Sciences* 12.10 (2022):728-728.
- [6] ESP32 development board . Available : <https://www.espressif.com.cn/en/products/devkits>.
- [7] MicroPython - Python for microcontrollers. Available: <https://micropython.org>.
- [8] Robot simulator CoppeliaSim: create, compose, simulate, any robot - Coppelia Robotics. Available: <https://www.coppeliarobotics.com/>.
- [9] Arm Jakub, et al. Measuring the Performance of FreeRTOS on ESP32 Multi-Core. *IFAC PapersOnLine* 55.4(2022):292-297.
- [10] Anonymous. Arduino Cloud IoT extends ESP32 support to RISC- V MCU. *Electronics Weekly* 2833(2023):18-18.
- [11] Arduino – Home. Available: <https://www.arduino.cc/>.