

Design of Modbus Communication Experiment Using a Pocket Experiment Board

Zhaonan Zhong, Yinhu A*, Zhi Weng*

College of Electronic Information Engineering, Inner Mongolia University, Hohhot, China

Email: *ayinhu@imu.edu.cn, *wzhi@imu.edu.cn

How to cite this paper: Zhong, Z. N., A, Y. H., & Weng, Z. (2024). Design of Modbus Communication Experiment Using a Pocket Experiment Board. *Creative Education*, 15, 1668-1676.

<https://doi.org/10.4236/ce.2024.158100>

Received: July 7, 2024

Accepted: August 11, 2024

Published: August 14, 2024

Copyright © 2024 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Conventional microcontroller experiments typically utilize products from educational instrument companies, where students complete tasks by wiring and programming. This approach often limits their deep understanding of the experiments and restricts practice to the laboratory. To address this, a microcontroller pocket experiment board has been developed, which requires students to solder the circuit board themselves based on a provided circuit diagram. This hands-on experience helps students thoroughly understand hardware circuit principles and acquire component soldering skills. Introducing Modbus communication, a widely used industrial automation protocol, into microcontroller experiments allows students to grasp the principles of Modbus communication, understand protocol specifications, and comprehend the communication process. This practical experience is invaluable for enhancing their competitiveness and provides a strong foundation for future careers in industrial automation, the Internet of Things, and related fields.

Keywords

Microcontroller, Modbus, Pocket Experiment Board

1. Introduction

With the rapid development of industrial automation technology, communication protocols play a crucial role in connecting and controlling various industrial equipment. Among them, the Modbus communication protocol has been widely used in the industrial field due to its openness, standardization, and ease of use. The Modbus protocol not only supports traditional RS-422, RS-485, and Ethernet devices, but also enables transmission over various media such as RS232/485, Ethernet ports, WIFI wireless, etc., making it a bridge connecting control devices

produced by different manufacturers. Therefore, mastering and proficiently applying the Modbus protocol is of great significance for improving the level of industrial automation and promoting industry development (Xie et al., 2022; Zhou, 2022).

In order to deepen the understanding of the Modbus protocol and improve its practical application capabilities, this paper designs a Modbus communication experiment based on a pocket experimental board. Pocket experiment board, as a portable and easy-to-use experimental platform, has the characteristics of low cost, easy access, and simple operation, making it suitable for students to conduct experimental operations and learn. Through this experiment, students can personally build a Modbus communication network, master the basic principles, message structures, functional codes, and other knowledge of the Modbus protocol, familiarize themselves with the application methods of the Modbus protocol in industrial control systems, and thus enhance their practical and innovative abilities (Li, 2023).

The main objectives of this experimental design include the following aspects: firstly, to enable students to understand the basic principles and working mechanisms of Modbus protocol through experiments; secondly, through experiments, students will master key knowledge such as message structure and functional codes of Modbus protocol; again, through experiments, students will become familiar with the application methods of Modbus protocol in industrial control systems, including how to configure Modbus devices, how to write Modbus communication programs, etc.; finally, through experiments, students' practical and innovative abilities are cultivated, and their comprehensive qualities are improved (Xian & Bi, 2022; Xiong & Xu, 2023).

2. Microcontroller Pocket Experiment Board

The microcontroller pocket experiment board is specifically developed for microcontroller experiments. Since it can be powered via a computer's USB interface, students can use it both during and outside of class, enhancing their learning experience.

2.1. Structure of the Experiment Board

JTAG Emulation and Debugging Unit: This unit includes a Jlink emulator for downloading and debugging programs. It features a TYPE-C interface, allowing students to use a mobile phone charging cable for connecting the computer and the experiment board, while also providing power to the board (Li et al., 2021; Wang et al., 2017) (**Figure 1**).

USB-to-Serial Communication Unit: Facilitates serial communication between the computer and the experiment board.

Main Control Chip Unit: Incorporates the STM32F103RC, a high-performance 32-bit MCU based on ARM, with up to 512 KB of flash memory, USB, CAN, 11 timers, 3 ADCs, and 13 communication interfaces.

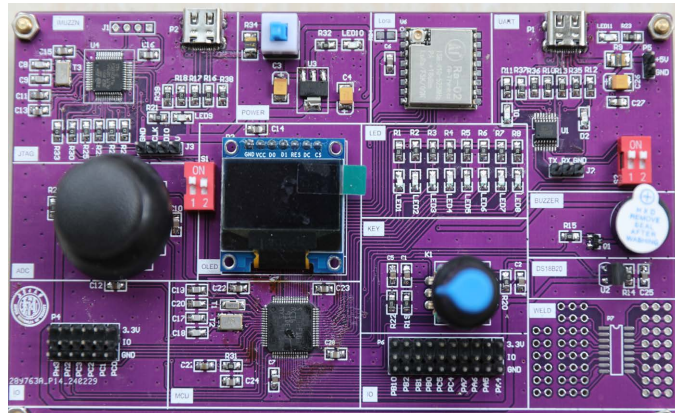


Figure 1. Pocket-sized experiment board.

Display Unit: Features a 0.96-inch OLED screen with a resolution of 128×64 .
Analog Joystick Unit: Includes a 3D model airplane joystick potentiometer, with dual potentiometers and a switch.

Keyboard Unit: Utilizes an EC11 rotary encoder.

Additional Units: The experiment board also consists of a display unit with 8 LEDs, a DS18B20 temperature sensor unit, a buzzer unit, and 16 IO expansion pins.

2.2. Preparation of the Experiment Board

In microcontroller experiment courses, each student is required to solder their own experiment board. PCB boards and components are distributed prior to class. Students solder the board following the circuit diagram, learning surface-mount soldering techniques and gaining a deeper understanding of circuit principles (Zhu et al., 2017).

Once the circuit board soldering is completed, it needs to be debugged, during which many issues may arise. Guiding students to resolve these issues based on the actual conditions helps them develop problem analysis and resolution skills. This experience is invaluable for both further academic pursuits and employment, providing lasting benefits (Figure 2).



Figure 2. Circuit board soldering.

3. Modbus Communication Protocol

Modbus is a communication protocol designed by MODICON for its PLC products. Functionally, it can be considered a fieldbus. Despite being over forty years old, the Modbus protocol remains vital. While its free availability is a significant factor, its enduring popularity is primarily due to its simple and reliable mechanism, making it highly suitable for industrial applications. Consequently, it is extensively used in the automation industry (Huang & Ding, 2016).

Through these experiments, students will gain proficiency in the following areas:

3.1. Modbus Bus Hardware Structure

The Modbus bus hardware structure follows a master-slave configuration, as depicted in **Figure 3**.

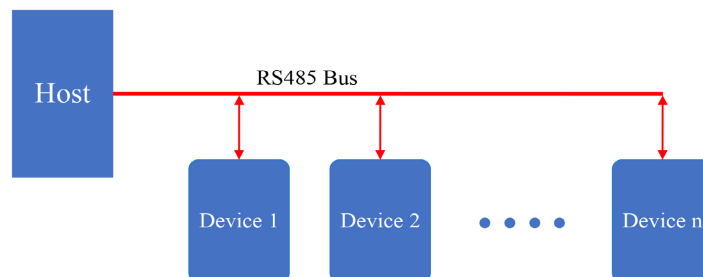


Figure 3. Modbus bus hardware structure.

In this setup, the master can be a computer or a touch screen, serving as the host device. Industrial applications commonly use the RS485 bus to connect slave modules. The master communicates with the modules based on their address numbers.

3.2. Modbus Data Model and Function Codes

The Modbus data model includes four definitions:

Discrete Input: Represents digital input, read-only, with function code 02 for reading.

Coil: Represents digital output, both readable and writable, with function code 01 for reading, 05 for writing a single coil, and 15 for writing multiple coils.

Input Register: Represents analog input, with function code 04 for reading.

Holding Register: Represents analog output, with function code 03 for reading, 06 for writing a single register, and 16 for writing multiple registers.

Since the holding register model can be flexibly adapted to replace the other three models, this experiment uses the holding register model with function codes 03 and 06.

3.3. Modbus Serial Data Frame Definition

The data frame definition is illustrated in **Figure 4**:

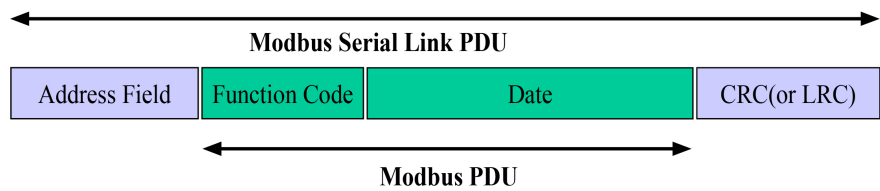


Figure 4. Data frame definition.

Address Field: Represents the module’s address number.

Function Code: Used for reading and writing operations on the data model.

Data: Represents the data of the register.

Checksum: Used to verify the correctness of the data

Modbus data frames come in two modes: RTU and ASCII. The RTU message frame uses the binary raw value, while the ASCII message frame converts the binary raw value to ASCII code, with a frame header “:” and a frame tail consisting of a carriage return and line feed.

This experiment uses the RTU message frame.

3.4. Modbus Holding Register Communication Mechanism

The microcontroller defines an int type array, illustrated as the central memory array in **Figure 5**, serving as a bridge for communication between the host and the microcontroller via the protocol stack, and for the microcontroller’s IO operations.

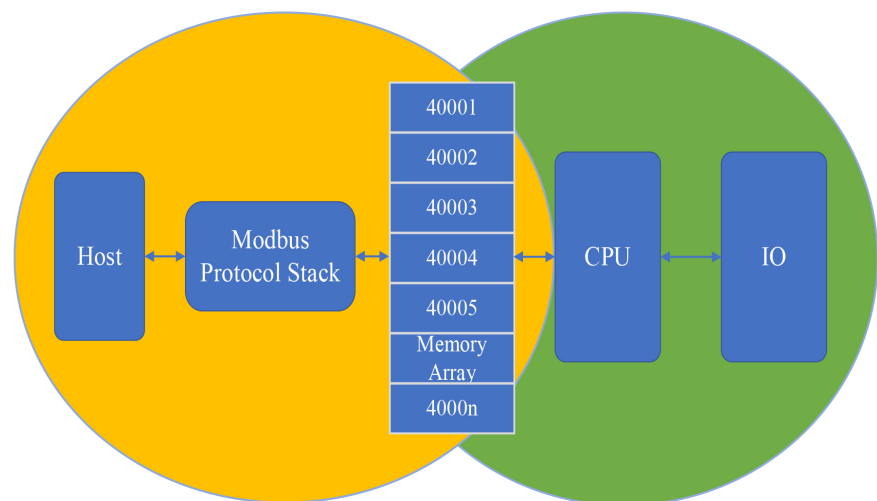


Figure 5. Holding register communication mechanism.

In this setup, the host reads the array using function code 03 in the left loop and can also modify the array using function code 06. In the right loop, the microcontroller updates the array with results from peripheral interactions. For instance, the microcontroller connected to 2 temperature sensors and 8 LEDs stores the temperature values in units 40,001 and 40,002. The host can retrieve these temperature values using function code 03 through the protocol stack. Ad-

ditionally, the host can modify the value in unit 40,003 using function code 06, and the microcontroller can drive the LEDs by monitoring changes in the value of unit 40,003 (Feng et al., 2021; Hou & Gu, 2017).

4. Experimental Practice and Effectiveness

4.1. Hardware Composition

The setup involves a computer converting USB to TTL serial via the FT230XS chip to connect with the microcontroller's serial port. An OLED display unit is used to show information, an analog joystick simulates voltage changes for ADC sampling, and 8 LEDs are used for digital output (Feng et al., 2014; Shen et al., 2021) (Figure 6).

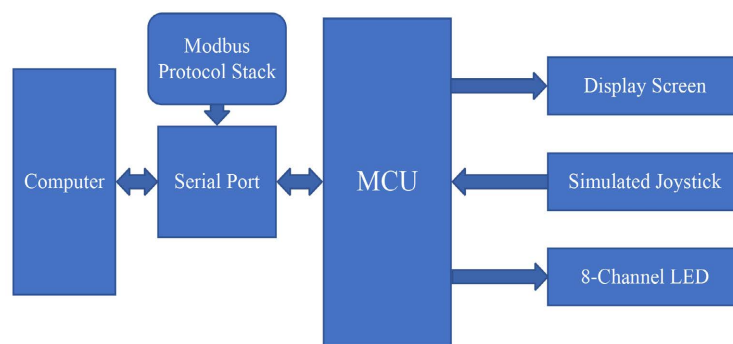


Figure 6. Hardware composition block diagram.

4.2. Lower-Level Software System Design

The microcontroller program setup is done using STM32CubeMX for configuring the clock and peripheral ports, generating the initial program framework, which can be compiled and debugged with IAR or Keil (Mo et al., 2016; Sheng et al., 2017).

Within this framework, the following steps are taken:

In ADC.c, the function uint16_t GET_ADC (uint32_t CH) is added to read the ADC conversion value, and it is declared in ADC.h.

The UART interrupt service function is modified to store data from the UART receive register into the receive array, reset the UART receive time variable, and set the receive flag variable to 1.

The timer interrupt service function is configured to trigger every 1 ms, incrementing the UART receive time variable each time.

The Modbus protocol stack files modbusRTU.c and modbusRTU.h are added to the project.

The main.c main function first initializes the working states of the system clock, timer, I/O, serial port, OLED display, etc.

In the infinite loop while (1), it first checks if the receive flag variable is 1 and if the serial receive time variable is greater than 10 ms (RTU messages are determined by the interval time to judge the received frame). If a frame of data is received, it verifies the data, checks the address, and parses the function code. If

the function code is 03, it returns the data in the memory array shown in **Figure 5** to the host computer based on the starting address and quantity of the registers. If the function code is 06, it writes the data sent by the host computer to the specified memory array.

Next, it collects the voltage values of the two potentiometers of the analog joystick through the ADC port and places them into the 40,001 and 40,002 memory arrays. It also checks for changes in the 40,003 memory array value, and the 8-bit binary value of its low byte corresponds to the on/off state of 8 LEDs, thus achieving control by the host computer.

Lastly, the OLED displays the values of the 40,001 - 40,004 memory arrays to facilitate debugging.

4.3. Integration Testing of Upper and Lower Software

First, Modscan software is used for integration testing with the experimental board to give students a preliminary understanding of the Modbus communication protocol (**Figure 7**).



Figure 7. Integration testing with software and experimental board.

Next, to deepen students' understanding of the application of the Modbus communication protocol, KingView software is used. Students use KingView software to define Modbus device drivers, set data dictionary variables, configure the interface, connect variables, and run integration testing with the experimental board, providing them with an introductory knowledge of industrial control (**Figure 8**).

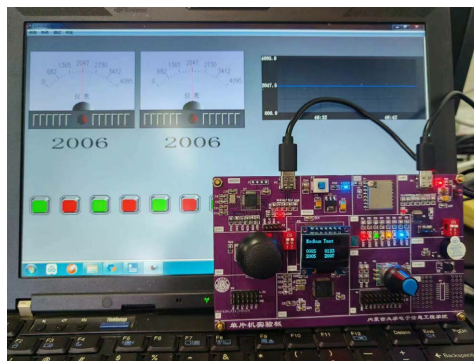


Figure 8. Integration testing with Kingview and experimental board.

5. Conclusion

Through this experiment, students go through the entire process of circuit soldering, program writing, system debugging, and completing a project. This helps cultivate students' ability to analyze and solve real-world problems, exercises their skills in applying learned knowledge to engineering practice, allows them to accumulate valuable practical experience, and enhances their competitiveness. This lays a solid foundation for future work in fields such as industrial automation and the Internet of Things.

Acknowledgements

This work was financially supported by New Engineering Research and Practice Project of the Ministry of Education (No. E-ZDH20201607), Key University Projects for Furthering Innovation and Entrepreneurship Education Reform in 2022, and 2023 Innovation and Entrepreneurship Education Practice Project.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- Feng, T., Li, Q., Wang, C., Cui, J., Pan, Y., & Ku, D. (2021). Design and Implementation of Dual axis Control Experimental System for Tower Solar Thermal Power Generation Heliostat Based on STM32. *Laboratory Research and Exploration*, *40*, 131-137.
- Feng, Y., Liu, X., & Xia, L. (2014). Experimental Teaching of Integrated Design of Embedded Technology Based on ARM Platform. *Laboratory Research and Exploration*, *33*, 205-207, 235.
- Hou, W., & Gu, Y. (2017). Design of a Temperature Controlled Intelligent Water Cup Controlled by a Microcontroller System. *Laboratory Research and Exploration*, *36*, 70-74.
- Huang, D., & Ding, L. (2016). ModbusTCP Communication Scheme of Microcontroller and Hard Protocol Stack. *Microcontrollers and Embedded System Applications*, *16*, 17-20.
- Li, J. (2023). Modbus Communication Method Based on Message Queue. *Automation Applications*, *No. 11*, 217-219.
- Li, W., Shang, Y., Yang, H., Yuan, X., Shao, X., & Zhao, J. (2021). Experimental System Design for AC Electromagnetic Field Detection Based on STM32. *Laboratory Research and Exploration*, *No. 11*, 86-89.
- Mo, S., Wang, K., & Zhang, H. (2016). Exploration of the Design of an IoT based Environmental Monitoring Training Platform. *Laboratory Research and Exploration*, *35*, 235-237.
- Shen, Z., Shu, G., Zhang, S., & Zhang, F. (2021). Experimental Setup for Smart Home IoT System Based on STM32. *Laboratory Research and Exploration*, *No. 11*, 60-63.
- Sheng, Q., Du, Y., & Luo, F. (2017). Research on the Algorithm of Solving Rubik's Cube Based on Stm32 Robotic Arm. *Laboratory Research and Exploration*, *36*, 29-32.
- Wang, H., Yu, L., Dong, Z., & Wang, Z. (2017). Research on the Experimental Teaching

- Platform of Tracking Balance Car Based on STM32. *Experimental Technology and Management*, No. 5, 136-139.
- Xian, J., & Bi, S. (2022). Research on Embedded Teaching Based on Innovative Comprehensive Experiments. *Experimental Science and Technology*, 20, 82-85.
- Xie, H., Liu, D., & Wu, Y. (2022). Design of Modbus Immersive Teaching Experiment Project for Industrial Internet. *Experimental Technology and Management*, No. 5, 171-175.
- Xiong, J., & Xu, Y. (2023). Design of Teaching Experimental Module Based on STM32. *Comprehensive Research on Agricultural Mechanization*, No. 1, 181-183.
- Zhou, S. (2022). Research on Embedded Course Teaching Based on Smart Devices. *Computers and Information Technology*, 30, 87.
- Zhu, X., He, C., Zhu, W., & Zhong, C. (2017). Design of Communication Technology Experimental System Based on STM32 Microcontroller. *Experimental Technology and Management*, No. 8, 81-84.