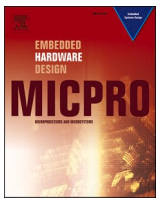
Microprocessors and Microsystems 94 (2022) 104682



Contents lists available at ScienceDirect

Microprocessors and Microsystems

journal homepage: www.elsevier.com/locate/micpro





Gigabit Modbus user datagram protocol fieldbus network integrated with industrial vision communication

Ming-Yi Lin [a], Ching-Han Chen [b,*], Zong-Bin Dong [b], Chun-Cheng Chen [c]

[a] Chinese Culture University, Department of Computer Science and Information Engineering , No.55, Hwa-Kang Road, Yang-Ming-Shan, Taipei City, 11114, Taiwan
[b] National Central University Department of Computer Science and Information Engineering Machine Intelligence and Automation Technology (MIAT) Laboratory, No. 300, Jhongda Rd., Jhongli District, Taoyuan City, 32001, Taiwan
[c] Taipei City University of Science and Technology, Department of Computer Science and Information Engineering, No.2, Xueyuan Rd., Beitou District, Taipei City, 11202, Taiwan

ARTICLE INFO

Keywords:
Embedded systems
Gigabit fieldbus
Image transmission
Industrial internet of things (IIoT)
Industry 4.0
Machine vision system
Smart manufacturing

ABSTRACT

In the era of Industry 4.0, the industrial internet of things (IIoT) is expected to transform traditional manufacturing. Machine vision, as the key enabling technology, can reduce maintenance and costs and improve production efficiency. This paper proposes an innovative machine vision system that uses the Gigabit Modbus user datagram protocol fieldbus. The protocol uses hardware and cross-layer function controllers to achieve edge computing with real-time image transmission. Results demonstrated that our innovative high-performance industrial machine vision solution can meet the requirements of IIoT applications. Moreover, the system considerably reduces the real-time response time and has superior usability compared with existing Ethernet fieldbus solutions.

1. Introduction

The internet of things (IoT) is characterized by the integration of sensing, actuating, information exchange, data processing, and other capabilities. The IoT can be used to support a wide range of applications, including smart homes, intelligent transport, and smart grids. Because of its availability, intelligence capabilities, scalability, and adaptability for connected devices, the IoT is often used in industry, which is termed the industrial internet of things (IIoT) [1–4].

The IIoT is a key topic within the rapid development of Industry 4.0. In addition, machine vision technology has great potential in the construction of powerful smart manufacturing applications for the IIoT system. The IIoT network enables any equipment or industrial device to be connected anywhere and anytime by using machine vision technology. Machine vision enables the IIoT network to efficiently gather and analyze data across machines, leading to the production of high-quality goods at reduced cost.

In addition to enhancing the operation of field devices, the integration of smart sensor arrays into such devices facilitates fault diagnosis, through which the processing parameters of the devices can be automatically adjusted using machine learning (ML) in the backend system to improve the efficiency of manufacturing and ease of maintenance. Moreover, when industrial devices integrated with machine vision are connected to the IIoT, images are collected from machines, and these images help in operation monitoring, computation, and intelligent decision making [5–11].

A complex IIoT network typically requires a hierarchical control system, with several studies [12–18] demonstrating the use of such a system in IIoT construction. Hierarchical control systems of the IIoT were presented in [12]; in the IIoT, the lower layer consists of local controllers that execute control tasks that are directly communicated to the subsystems. The control results are transmitted to a higher level for monitoring of the internal controls or effective coordination of the large systems.

A multilevel structural scheme of the IIoT was described in [13]. The scheme is mainly used to operate complex and large manufacturing systems. The perception layer comprises local controllers, which perform control functions through direct communication with the subsystems. The application layer monitors the data procured by the lower layers and coordinates the data for the large systems. A novel IIoT architecture with hierarchical control structure in the mobile edge cloud (MEC) was proposed in [14]. This architecture separates the control

* Corresponding author.
E-mail addresses: lym15@ulive.pccu.edu.tw, pk4881@gmail.com (M.-Y. Lin), pierre@g.ncu.edu.tw, pierre.miat@gmail.com (C.-H. Chen), neriot25961@gmail.com (Z.-B. Dong), jcchen@tpcu.edu.tw (C.-C. Chen).

https://doi.org/10.1016/j.micpro.2022.104682
Received 11 April 2021; Received in revised form 12 November 2021; Accepted 14 September 2022
Available online 16 September 2022
0141-9331/© 2022 Elsevier B.V. All rights reserved.

plane from the data plane on the basis of software-defined networking, with one centralized and one distributed control system. Furthermore, a deep-learning technique is implemented in the MEC to enhance the edge intelligence.

The security concerns related to information technology (IT) and operational technology (OT) in distributed IIoT-based control architecture were explored in [15]. The authors proposed a design for secure data exchange between the IT and OT domains in a smart factory of Industry 4.0. In [16], control system architecture was created for a smart factory; manufacturing execution systems and manufacturing operations management were the core of this architecture. Additionally, the authors of [17,18] developed three-layer hierarchical control of an inverter-based microgrid (MG) by using cloud-based IoT infrastructure and an ML-based islanding detection scheme. This work contributed to the successful implementation of MG hierarchical control using an IoT feedback network with low data traffic.

The bottom layer of a control system uses the fieldbus to link objects. Fieldbus network requirements in a machine vision system vary because of the different control interfaces and communication protocols of the various devices; meeting the demands of reliability and real-time communication in the IIoT network is difficult.

Regarding network scalability and computational and communication efficiency, solutions that employ a traditional fieldbus cannot deliver IIoT functionality, such as big data collection, image transmission, and image streaming. Further compounding the technical challenges of the machine vision system is the necessity that the fieldbus must manage big data collection from objects as well as meet reliable real-time requirements.

The control system and machine vision system typically operate independently, and the integration of these two systems is expected to result in increased network bandwidth costs. Therefore, solutions must be appropriately integrated to fulfill the requirements of an industrial fieldbus with machine vision for complex IIoT systems. In this paper, we propose innovative Ethernet-based fieldbus system architecture by using Gigabit Modbus architecture as the core design concept to overcome the problem of complex heterogeneous system integration.

The Gigabit Modbus User Datagram Protocol (UDP) fieldbus was designed to solve the following problems:

1) The system's high flexibility and scalability solve the problem of resource insufficiency in traditional fieldbus application.
2) The system can overcome the ineffective integration of an Ethernet-based fieldbus with a machine vision system.
3) The system comprises inexpensive, highly efficient parallel structures to address the poor performance of Ethernet-based industrial technology.

The activities and contributions of this study can be summarized as follows. In the proposed Gigabit Modbus UDP framework, reconfigurable field-programmable gate arrays (FPGAs) were combined to achieve a high-performance machine vision solution in the IIoT system. The proposed system features high-speed communication between cross-layer controllers to achieve real-time edge computing of transmitted images. Furthermore, the cross-layer controller architecture enables seamless roaming in heterogeneous systems.

The rest of this manuscript is organized as follows. In Section 2, related works on the Ethernet-based fieldbus are discussed. The development of the controller design for the Gigabit Modbus UDP protocol is presented in Section 3, and the operation of the function modules is explained. Implementation of the machine vision system is detailed in Section 4, and Section 5 offers some concluding remarks.

2. Related Work

In recent years, industrial Ethernet technology has become more ubiquitous, offering high-speed operation, increased connection distance, and the ability to connect numerous nodes. Industrial Ethernet technology is also cost effective, with common physical links, and is flexible in terms of both the network topology and number of system nodes. As such, for many IIoT networks, industrial Ethernet-based solutions are becoming preferred. Many industrial Ethernet protocols are being developed by industrial equipment manufacturers [19,20], such as the CAN bus [21], CC-Link IE [22], Ethernet for Control Automation Technology (EtherCAT) [23], EtherNet/IP [24], Modbus [25], Modbus/TCP(Transmission Control Protocol) [25], PROFIBUS [27], Process Field Network (PROFINET) [27], POWERLINK [26], and Serial Real-Time Communication System (SERCOS) III [28].

This section focuses on six popular protocols that have emerged as contenders for high-performance open standard within real-time Ethernet fieldbus applications, namely EtherCAT, EtherNet/IP, POWERLINK, PROFINET IRT [27], SERCOS III, and Modbus TCP/IP. We reviewed these protocols and analyzed their operation [20,26]. Other industrial Ethernet technologies also leverage Ethernet, but studies of their components have been insufficient in the open-source community to be considered standard and open. Thus, this section does not discuss these proprietary technologies.

The EtherCAT [23] protocol, initially developed by Beckhoff Automation, enables on-the-fly packet processing and delivers real-time Ethernet to automation applications. EtherCAT was designed for centralized controller architectures with simple field devices in a master–slave configuration and can provide scalable connectivity for entire automation systems, from large programmable logic controllers (PLCs) to the input/output (I/O) and sensor levels. Because of the low latency of each slave node, EtherCAT provides flexible, inexpensive, and network-compatible industrial Ethernet solutions.

EtherNet/IP [24] is an open industrial Ethernet protocol originally developed by Rockwell Automation. EtherNet/IP is essentially a port of the common industrial protocol (CIP), which was already used in ControlNet and DeviceNet, to the Ethernet data transfer protocol. EtherNet/IP is widely used in a range of industries, particularly in the US market, and employs both the Internet Protocol suite and IEEE 802.3 to determine features and functions for its physical, data link, network, and transport layers, employing the CIP over TCP/IP. In [19,20], the CIP's object-oriented design has been used to provide a common set of messages and services for industrial automation control systems and to promote consistent implementation of automation functions across a diverse product ecosystem. EtherNet/IP is compatible with many standard Internet and Ethernet protocols but has limited real-time and deterministic capabilities.

POWERLINK [26], a real-time protocol for standard Ethernet and represents an open technology that is regarded worldwide as an International Electrotechnical Commission (IEC) standard and purely software-based communication system delivering effective real-time performance. POWERLINK uses a polling and time-slicing mechanism to achieve isochronous data transfer, controlling time synchronization through packet jitter to within the range of tens of nanoseconds. The manager enforces the cycle timing that synchronizes all devices and controls cyclical data communication. Such a system is suitable for all types of automation system ranging from PLC-to-PLC communication to visualization.

PROFINET [27] specifies all data transfer between I/O controllers and is widely used by industrial equipment manufacturers in their industrial Ethernet under three classes, namely Classes A, B, and C. PROFINET Class A has a cycle time of approximately 100 ms. Class B, also referred to as PROFINET Real-Time, offers a software-based real--time approach and has a cycle time of approximately 10 ms. Class C is the PROFINET isochronous real-time channel this is implemented using a custom application-specific integrated circuit and requires special hardware to reduce the cycle time to less than 1 ms to deliver sufficient performance in real-time industrial Ethernet for motion control operations.

SERCOS III [28] is a freely available real-time communication

standard for digital drive interfaces. This standard is predominantly used for communication between industrial devices and standard Ethernet nodes. The SERCOS III transmission process is based on a time slot mechanism in which bandwidth is subdivided into the real-time channel and asynchronous data traffic. In addition to the real-time channel, which uses time slots with reserve bandwidths to ensure collision-free data transfer, this system allows for an optional non-real-time channel.

In [28], collective telegrams specified using SERCOS III were transmitted as a broadcast in the real-time channel. Two main types of telegram are used within the SERCOS III communication cycle, namely the Master Data Telegram and Acknowledge Telegram. These telegrams ensure that only part of the bandwidth is used for real data exchange, which reduces the effect of the Ethernet frame overhead on network performance. Similar to in EtherCAT, a SERCOS III slave processes a packet through the on-the-fly extraction and insertion of data into the Ethernet frame to achieve low latency.

The Modbus protocol, used in the automation and sensory industries, represents a simple format of communication using the master–slave principle. Compared with other forms of industrial Ethernet technology, Modbus is advantageous because of its early emergence, ease of use, openness, favorable price–performance ratio, and reliability. The Modbus messaging structure is the application protocol that defines the rules for organizing and interpreting data independent of the data transmission medium. Modbus supports numerous network transmission media and electrical interfaces and has thus been widely used by businesses.

The Modbus TCP/IP (also Modbus/TCP) protocol works within the application layer of the TCP/IP model, providing connectivity within Modbus and TCP/IP interconnected Modbus networks. Modbus/TCP adopts the master–slave communication mode in which the master system actively writes and reads information into the controlled slave device. The master system can communicate with the slave device through one-to-one or many-to-many communication. Modbus/TCP is simply the Modbus RTU protocol with a TCP interface that operates on Ethernet. Because Modbus/TCP is a simple protocol with few function codes and commands for each communication channel, a master can have multiple outstanding transactions. In Modbus/TCP, slaves listen for incoming TCP connections on port 502 and may optionally listen on additional ports.

Modbus/TCP is a reliable service in that each packet is guaranteed to arrive at the destination. To achieve this goal, all data are positively acknowledged by the destination, with those not acknowledged within a certain time limitation retransmitted. Because of the traffic load variance on the internet, the time limitation can change dynamically. For a long trip between a source and destination, the round-trip times of transmitted data and their acknowledgements are considerable, and the time limitation must be extended. Consequently, the retransmission of lost packets may happen after a longer time.

The combination of a machine vision system and the IIoT has received increased attention in both academia and industry. The transmission of image streaming typically has bandwidth, delay, and loss requirements. Although the delay time can be controlled through schemas such as buffering or streaming systems, pauses and frame skips are unavoidable. For image streaming, in-order and timely streaming may be more critical than reliability and congestion control. The limitations of using Modbus/TCP are summarized as follows:

1 The main disadvantage is that real-time performance, such as the transmission delay, is negatively affected when using Modbus/TCP. Because Modbus/TCP is a reliable protocol, delays are introduced whenever packet loss occurs.
2 Modbus/TCP is master–slave protocol, and the slave device must routinely poll each field device to search for changes in the data. This consumes bandwidth and network time in applications in which bandwidth may be expensive, such as over a low-bit-rate link.

Therefore, Modbus/TCP is not suitable for real-time machine vision systems. Modbus/UDP was then analyzed as the next candidate for image streaming, and its advantages are described as follows:

Modbus/UDP is increasingly preferred over Modbus/TCP as a result of its suitability for real-time implementation [29–31]. Compared with Modbus/TCP, Modbus/UDP is a connectionless protocol that does not wait for data receipt acknowledgement, thereby reducing the overall latency. In Modbus/UDP, data are not retransmitted in the case of data loss, ensuring only the most recent data are updated. Whenever data are being received by the receiver in Modbus/UDP, less delay occurs because even packets that are lost during transmission are not retransmitted. In addition, the task-specific uses Modbus/UDP to prevent Modbus/TCP retransmission after errors [32].

Modbus/UDP offers the desired characteristics for stable surveillance systems with minimal latency [33]. Thus, in the present work, we employed Modbus/UDP for image transmission because in-order and timely transmission was our major goal. However, Modbus/UDP does not provide assurance of delivery of the packet and, although it is faster than Modbus/TCP, it is less reliable. Nevertheless, the overhead involved in the provision of these services is reduced in Modbus/UDP operation. Thus, Modbus/UDP, with its low overhead and high speed, is a simpler and more efficient protocol [34].

Although Modbus/UDP has numerous benefits, certain limitations must be taken into account, especially in cases of missing or duplicate datagrams or datagrams in the incorrect order. Modbus/UDP is faster but less reliable than Modbus/TCP, with packet loss representing one of UDP's major problems. For validation purposes, we benchmark tested the communication interfaces through the transference of image streaming. Moreover, reliability indices were defined to characterize reliable communication in terms of the success, retransmission, and packet loss rates. The measured results are discussed in Section 4.5.

3. Architectural Design of Gigabit Modbus UDP Protocol

In this work, we used Gigabit Ethernet and Modbus as the transport and application layer protocols, respectively. The architecture of the Gigabit Modbus UDP fieldbus network comprises two main layers, as illustrated in Fig. 1. The host is in charge of the system connection with the Modbus master controller. The Modbus master of the gateway layer is connected to multiple Modbus slaves in the device layer through Gigabit Ethernet. The core is the Gigabit Modbus UDP controller, which controls and manages the function controllers, enabling network image transfer and transmission reception.

3.1. Modeling Design for Gigabit Modbus Fieldbus Protocol

We used a high-level methodology [35–38] for designing the hardware of the fieldbus-based machine vision system based on the Gigabit Modbus UDP protocol. Through application of the hierarchical modeling tool IDEF0 (ICOM DEFinition0), a complex system was decomposed into a set of functional modules. The behavior of each module can be represented as a sequential–concurrent hybrid discrete-event system. We employed high-level synthesis rules to produce VHSIC Hardware Description Language (VHDL)-based efficient hardware for the function controllers. Finally, the very large-scale integration (VLSI) circuit for each function controller was easily obtained. This section presents the hierarchical modeling of the architecture of the proposed protocol using IDEF0.

As depicted in Fig. 1, the Gigabit Modbus UDP fieldbus network comprises multiple major controllers. The Modbus master controller is used to receive the packets transmitted from the Modbus slave controllers to the host and, through synchronization, transfer those packets to the transfer controller of each Modbus slave. The Modbus slave controllers are integrated with the transfer controller and Gigabit Modbus UDP controller, which is the core backbone for image communication. Subsequently, the receiver controller receives the

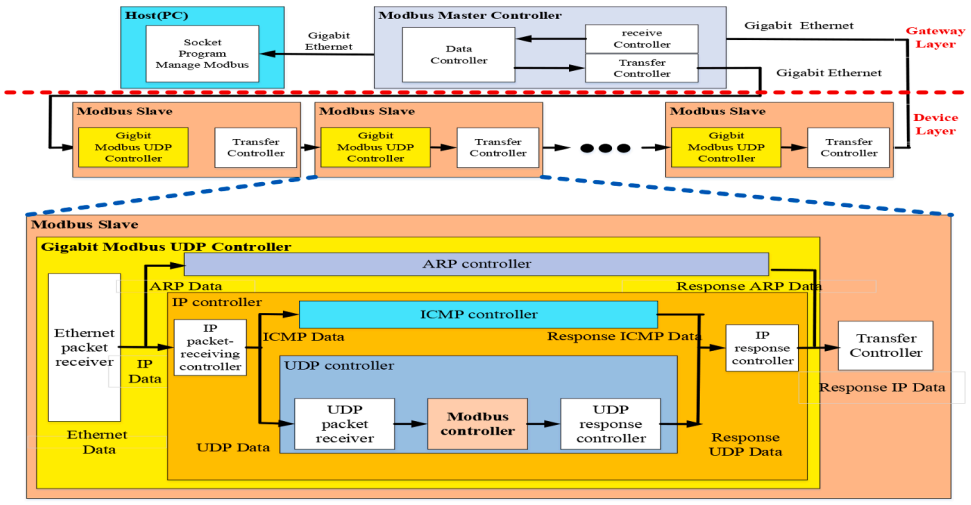


Fig. 1. Gigabit Modbus UDP fieldbus network architecture.

packets, which are then transferred to the packet controller as part of the pipeline architecture. The packets are then transmitted to the host to complete the response. The function of the host is to monitor and manage the entire system, enabling packet capture, information analysis, and diverse equipment monitoring. The master controller module comprises three submodules: the data controller, transmission controller, and receiver controller (Fig. 2). The transmission and receiver controller modules are responsible for receiving the commands sent by the host; once the Ethernet packets are received, they are transferred to the transmission controller for synchronization. To reduce the token-ring-induced delay, a determination must be made about whether the instruction sent by the host is intended for the slave controller (Fig. 3), followed by further verification of what type of communication protocol information is being transferred to the corresponding controller. As presented in Fig. 4, the Gigabit Modbus UDP controller submodule in the Modbus slave controller is the core of the fieldbus system and responsible for processing image transmission tasks, controlling the cross-layer function controllers, delivering packets, and receiving packets in conjunction with other heterogeneous-layer controllers, including the main Ethernet receiver, ARP controller, and Modbus IP (Internet Protocol) controller submodules. The Address Resolution Protocol (ARP) controller consists of the ARP receiver and ARP transmitter submodules (Fig. 5) and is responsible for processing the packets of the ARP protocol and returning the corresponding message. As depicted in Fig. 6, the IP controller was integrated with the IP receiver, IP transmitter, UDP controller, and internet control message protocol (ICMP) controller submodules. The IP controller processes the packets of the IP protocol, determines whether the IP matches, and further processes and sends the response message. The ICMP controller submodule, which comprises the ICMP receiver and ICMP transmitter submodules (Fig. 7), receives the packet, processes the ICMP protocol, and transmits the result back to the IP transmitter. The UDP controller, composed of the UDP receiver, UDP transmitter, and Modbus controller submodules (Fig. 8), receives the packet and processes the UDP protocol.

As illustrated in Fig. 9, the Modbus UDP controller submodule can be divided into the UDP receiver (A1331), Modbus controller (A1332), and UDP transmitter (A1333) submodules. The Modbus controller is the core communication component of the industrial fieldbus network. Following receipt of packets to determine their corresponding function code, the appropriate processing is performed, and the packets are sent to the UDP transmitter. The Modbus controller is the most crucial component of the system and is part of the application-level controller; it establishes the network communication protocol through the cross-layer network controller to complete packet transmission from the Gigabit Modbus UDP fieldbus network, which includes the function of image transmission. This feature must be coordinated with the upper network controller. To ensure the reliability of real-time industrial vision communication, we employ FPGAs for implementing each function controller module. The main purpose of these controllers is to complete high-speed packet transmission and management tasks under the

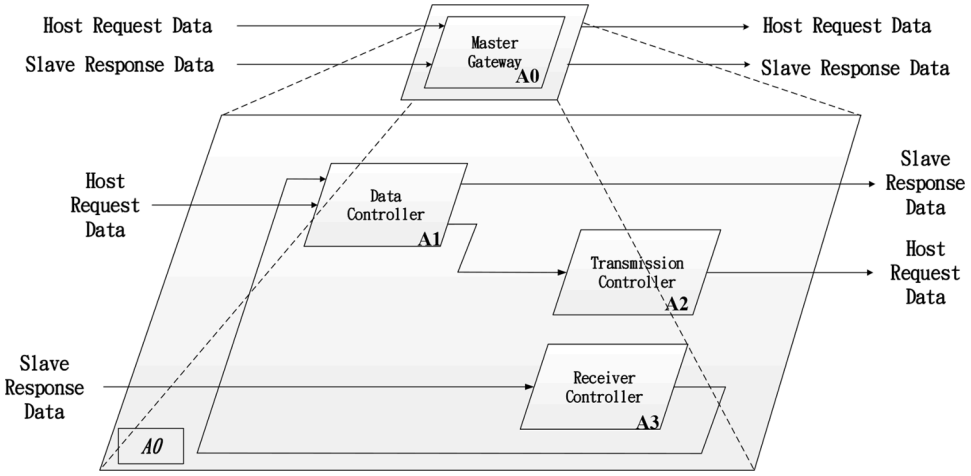


Fig. 2. Master controller.

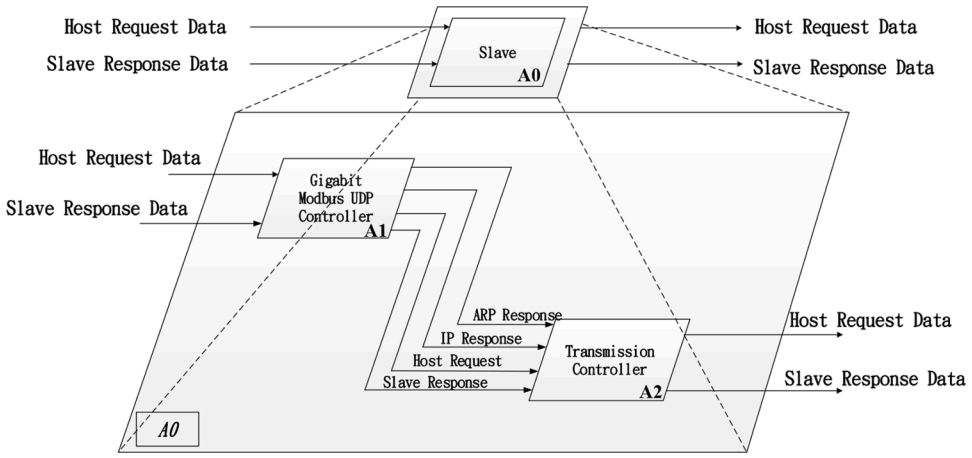


Fig. 3. Slave controller.

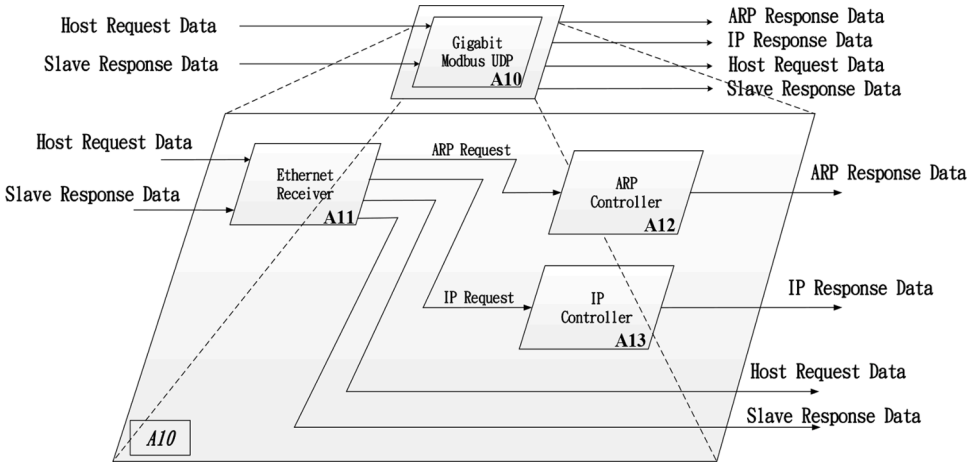


Fig. 4. Gigabit Modbus UDP controller.

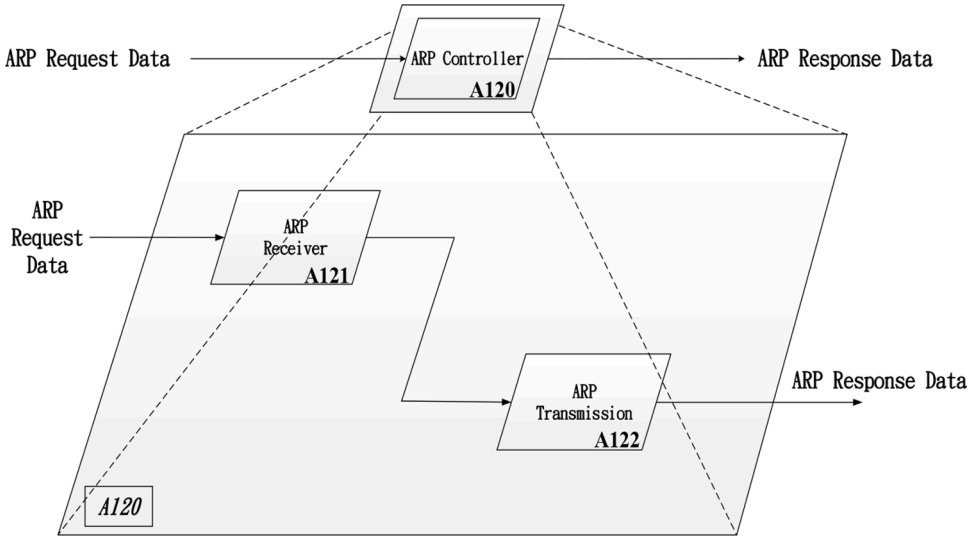


Fig. 5. ARP controller.

industrial fieldbus network settings. Therefore, the Modbus UDP controller can be used to manage image transmission between the master and slave controllers.

3.2. High-Level Synthesis of the Gigabit Modbus Slave

To achieve high-performance communication, all function controllers of the Gigabit Modbus protocol are implemented through fully

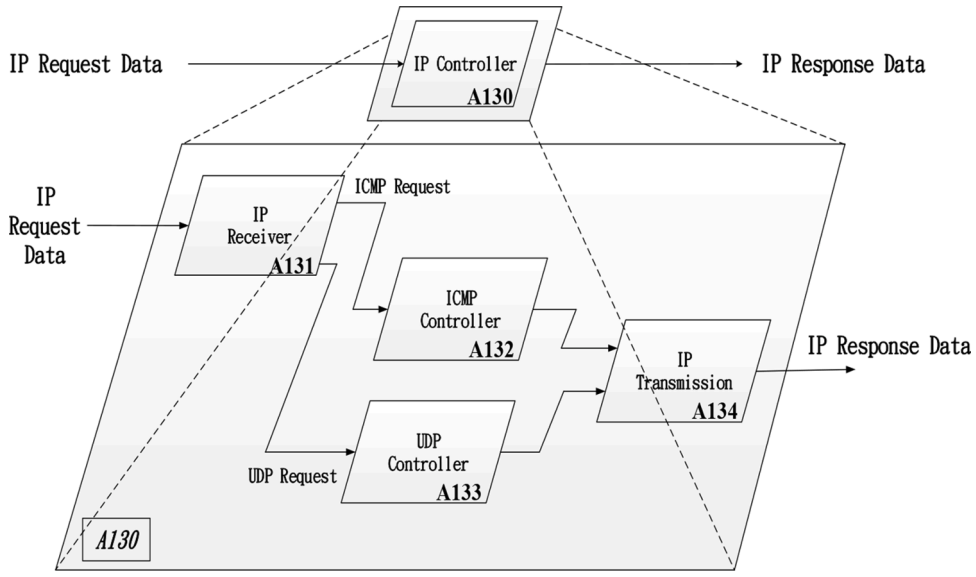


Fig. 6. IP controller.

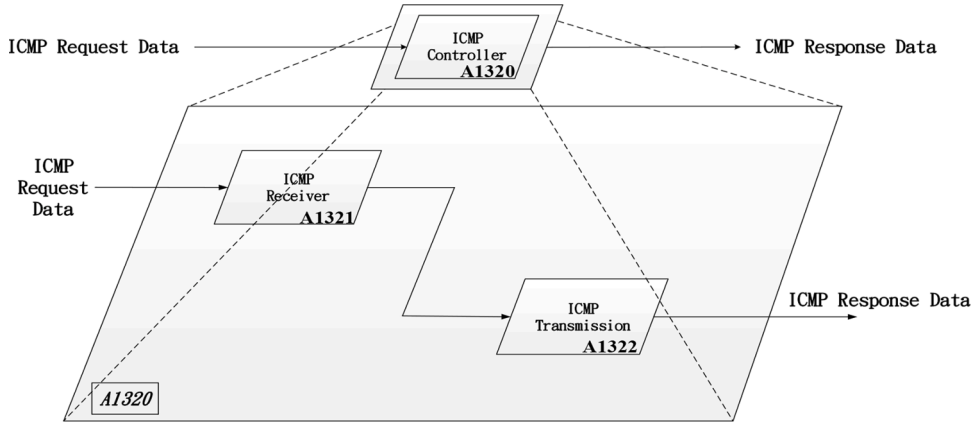


Fig. 7. ICMP controller.

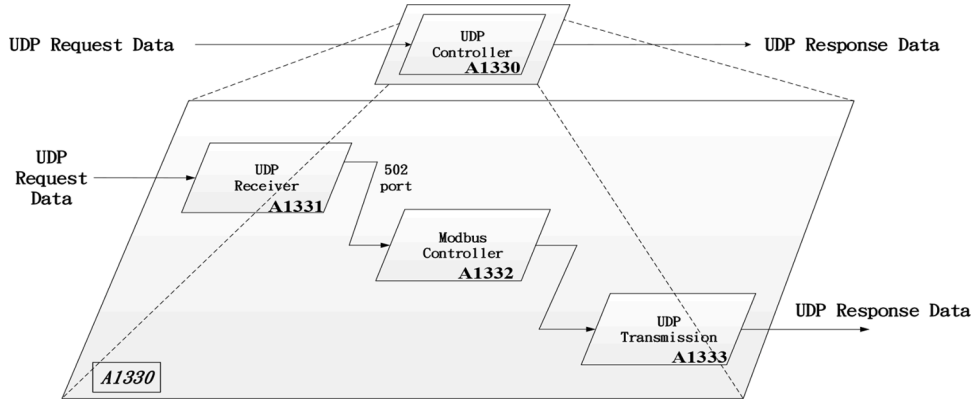


Fig. 8. UDP controller.

hardware-based pipeline architecture.

According to the modeling design detailed in Section 3.1, each controller is generated automatically to integrate all intelligent functional modules into a complex embedded system, and the VLSI circuits are then synthesized. Fig. 10 illustrates the VLSI circuit of the proposed Gigabit Modbus slave; this circuit was discovered to exhibit strong reliability in terms of real-time performance. The resource utilization of the Gigabit Modbus slave is summarized in Table 1. Details of the high-level synthesis—including the hierarchical discrete-event modeling using the IDEF0 tool, discrete-event modeling using GRAFCET language, and a set of VHDL synthesis rules—have been described in [35–38].

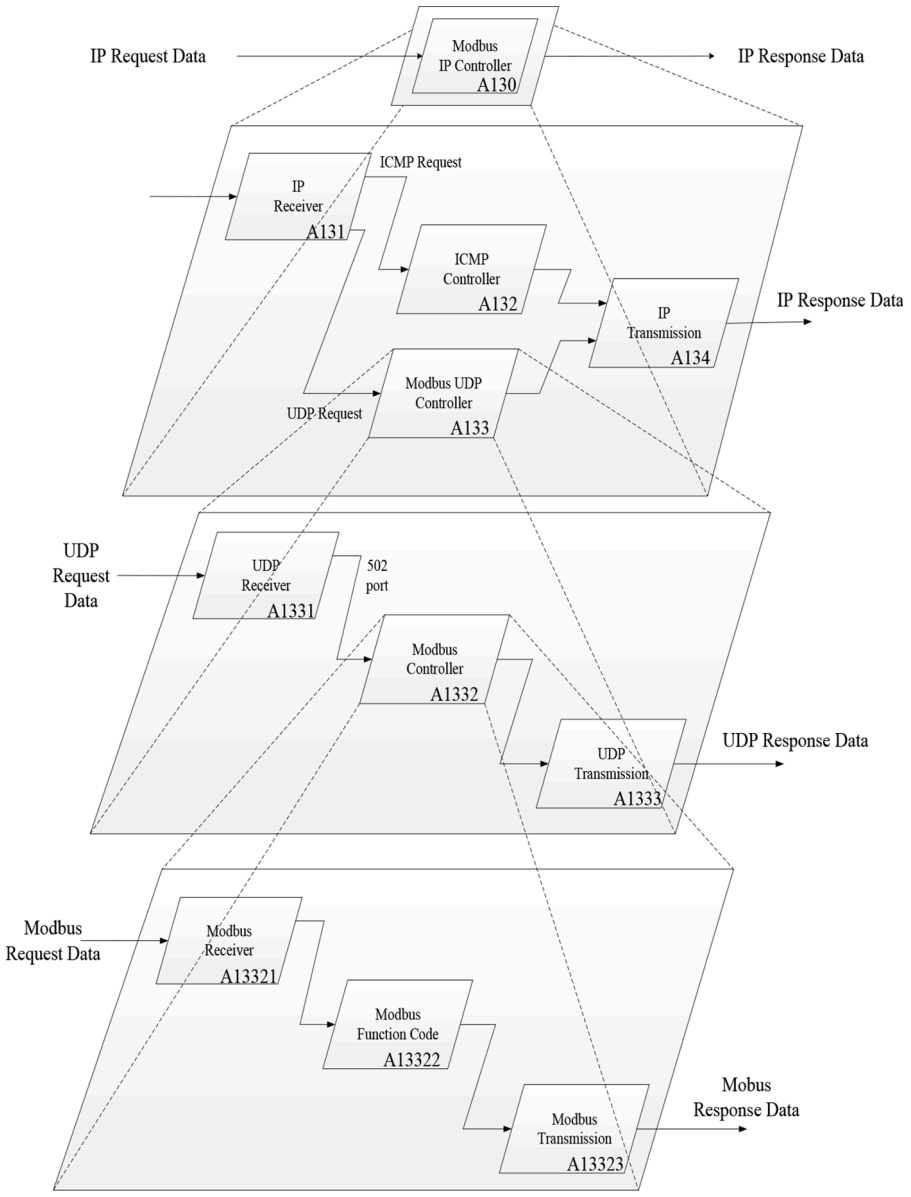


Fig. 9. Modbus controller.

4. Implementation

The system architecture of the high-speed Gigabit Modbus UDP fieldbus network comprises cross-layer interaction modules (introduced in Section 3). In this section, we discuss the prototype of the Gigabit Modbus UDP fieldbus-based machine vision system, verify the image stream processing, and analyze the performance indices in comparison of our solution with other Ethernet-based fieldbus solutions.

4.1. System Integration and Verification

The prototype of the proposed system comprised a host interface with Modbus master for surveillance, Gigabit Modbus slaves, and two camera modules (OV7725) for the perception layer. Both the master and slaves were implemented and tested on the high-speed parallel FPGA platform.

4.2. Function Waveform Simulation Verification

All of the function controllers were combined with the electric-only hardware circuit to verify the performance of the integrated Gigabit Modbus UDP controller. To assess the network's functions, waveform simulation verification was performed using the predefined network functional modules proposed in the Gigabit Modbus UDP fieldbus network. The simulated waveform of the Gigabit Modbus UDP controller is presented in Fig. 11.

When the Gigabit Modbus UDP controller is activated by the ff_rx_dval receiving control signal, the Ethernet packets of the ff_rx_data submodule to be received are loaded into the buffer register. Then the eth_type submodule determines the correct service submodule and initiates the corresponding submodule. During the receipt process, the packet is sent to the next Modbus slave synchronously, following which, the token-ring transmission is completed. As illustrated in Fig. 11(a) and (b), the IP controller submodule and ARP controller submodule, respectively, are triggered in the Gigabit Modbus UDP controller. Fig. 12 depicts the UDP controller submodule being triggered in the Gigabit Modbus UDP controller.

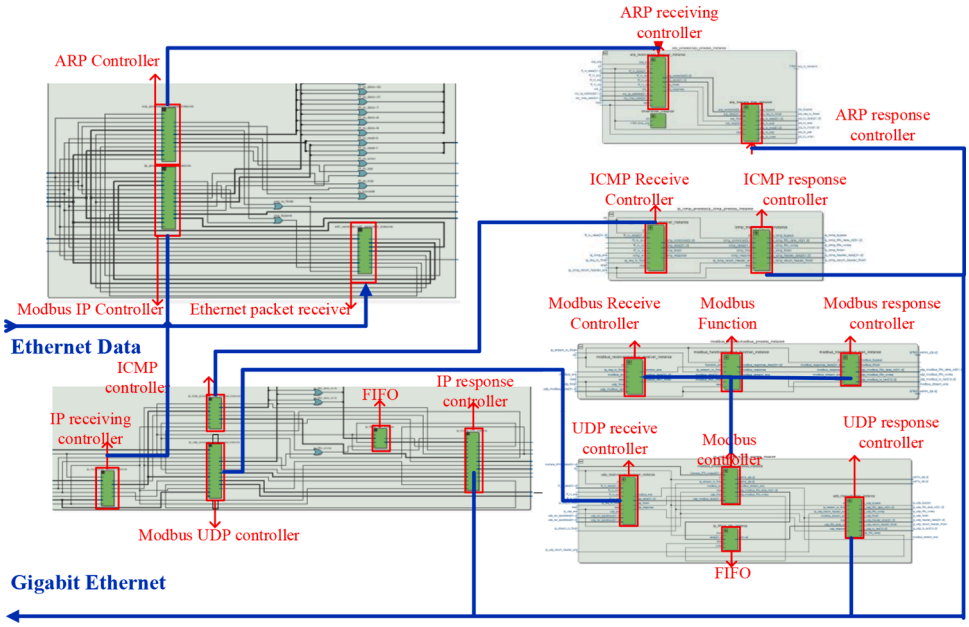


Fig. 10. VLSI hardware circuit for the proposed Gigabit Modbus slave.

Table 1
Resources used for implementing the high-speed FPGA controller of the proposed Gigabit Modbus slave. The numbers in brackets indicate device utilization.

Gigabit Modbus slave hardware specification	
FPGA chip	EP4CE115F29C7
System Clock(Max)	157.95 MHz
Total logic element	2,641/114,480 (2%)
Total register	1,550
Total Memory bits	33,280/3,981,312(1<%)
Number of I/Os	528

4.3. Image Streaming Transmission Verification

For validation purposes, we benchmark tested the communication interfaces through the transference of uncompressed raw 640 × 480 grayscale image streams from the Modbus slave of the device layer over Gigabit Ethernet to the Modbus master of the gateway layer to verify whether the proposed solution can meet the required standards. Over Gigabit Ethernet, the image is received and transmitted to the Gigabit Modbus UDP controller for processing. The received packets are then transmitted over Gigabit Ethernet to the host to verify whether the received data are correct [Fig. 13(a)]. As presented in Fig. 13(b), the Modbus function code (50, 0 × 32) was predefined for image streaming transmission; the source IP address was 192.168.0.1 and destination IP-address was 192.168.0.100. When the transmission was complete, the source sent a message of receipt that the destination transmission had been accomplished. The image streaming data comprised a series of 1280 bytes, and the response time was 0.003361 ms, as illustrated in Fig. 13(c).

To test reliability, we measured several performance indices—the success, retransmission, and packet loss rates—and response time of the proposed system. We benchmark tested the communication interfaces through transference of the error detection and reporting mechanism, Modbus TCP mode, of the 640 × 480 and 1920 × 1080 grayscale images of streamed data from the Modbus slave to Modbus master over Gigabit Ethernet. The explanation of test parameters is summarized in Table 2.

In order to provide users with convenient operation interface and flexible communications protocols, general network packet processing usually includes software control and application level design, so the communications protocols above the network level are often intervened by the operating system, which increases the uncertainty factors affecting packet processing, such as whether the software is well or poor designed and the loading of the operating system, etc., and thus increases the response time of the overall system. The proposed hardware system uses a pipelined approach to packet processing and sending in order to effectively reduce response time, assigning the unpacketing process to different controllers. The results of testing the various transference parameters for 100 consecutive transmissions are detailed in Table 3.

As illustrated in Table 3, the proposed hardware system has a better response time. In the software system, ICMP-Ping is operated directly from the system level, while the rest, including Modbus TCP Mode and Modbus image streaming transmission, have to be controlled by the upper software level, thus increasing their response time.

Since the proposed hardware system does not have the intervention of the operating system, the response time will only be affected by the size difference of the packet during the packet processing, but not be disturbed by problems such as time-sharing and interrupt, so the performance in speed is up to the response level of Gigabits. The proposed hardware solution improved the performance by 50 times and image transmission by almost 2 times compared with conventional software Modbus. For the different transmission parameters, the success rate was 100%, with retransmission and packet loss rates of 0. We designed the Gigabit Modbus UDP protocol on the basis of the goal of performance optimization, reducing the response time of the Modbus slave node from 1 ms to 69 μs for the high-performance synchronous motion control required. Therefore, it is necessary to implement Modbus/UDP Fieldbus in hardware, so regardless of how large or small the amount of data is, it can show better response speed and stability to meet the real-time requirements of Fieldbus-based machine vision system.

In addition, we measured the response time with equipment amounts ranging from 22 to 800 under different network topologies. Individual tests for the different equipment amounts were conducted over 100 consecutive transmissions, and the mean response time was determined to determine whether the proposed solution has satisfactory technology innovation indices. The results are presented in Fig. 14.

As presented in Fig. (a) and (b), the star topology outperformed the ring topology. When image transmission was conducted, the success rate was 100%, and the retransmission and packet loss rates were 0 for

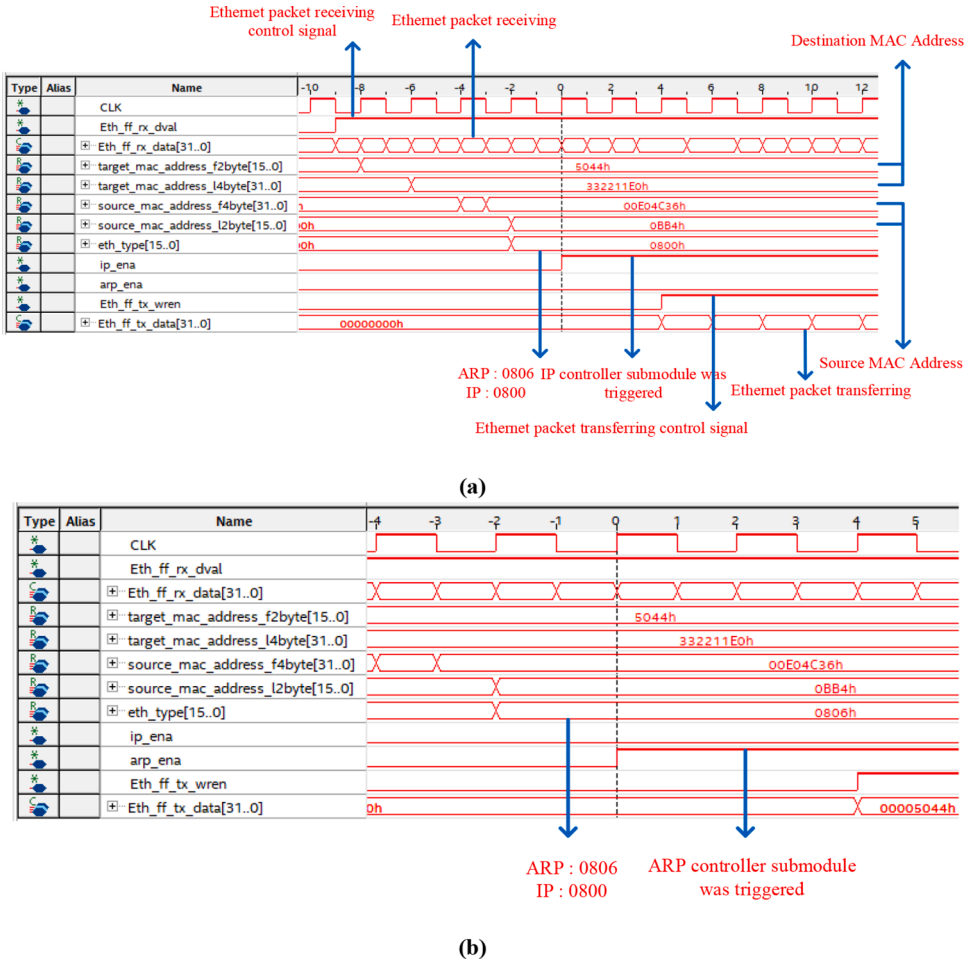


Fig. 11. Gigabit Modbus UDP controller. (a) The IP controller submodule and (b) ARP controller submodule triggered in the Gigabit Modbus UDP controller.

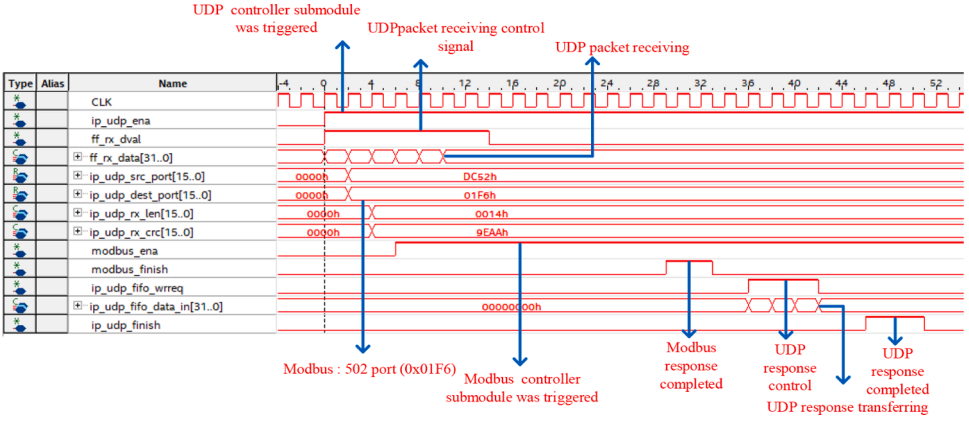
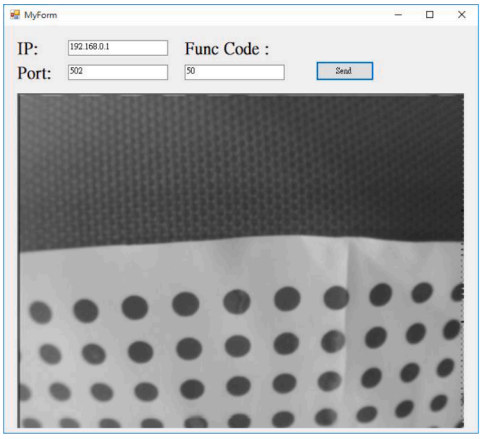


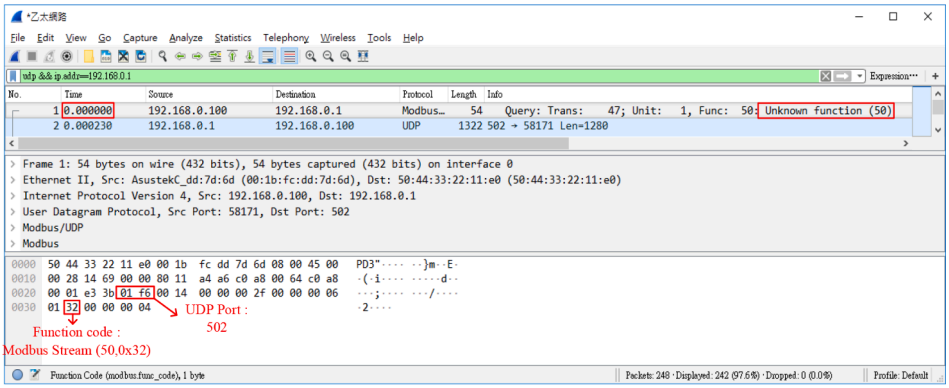
Fig. 12. UDP controller submodule triggered in the Gigabit Modbus UDP controller.

different equipment amounts, as illustrated in Fig. (c) and (d). As depicted in Fig. (c), for 640 × 480 and 1920 × 1080 grayscale image transmission using star topology, the mean response time was 3.4 and 22 ms, respectively. Because a hierarchical management mode is employed in the architecture, the response times of devices under the same level of the central device or host were approximately the same.

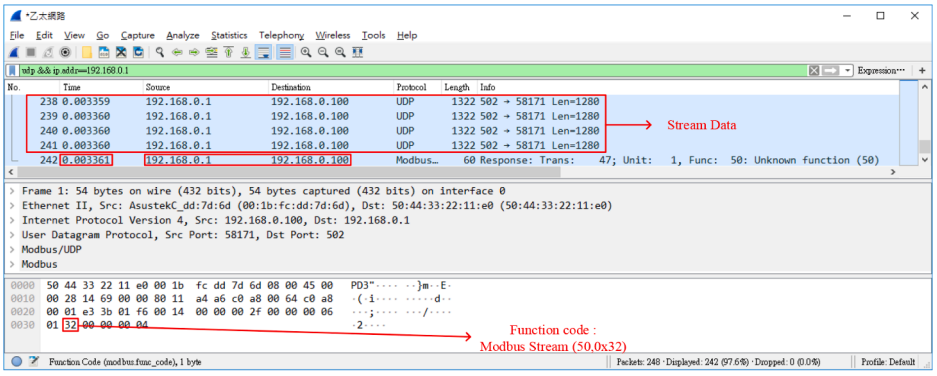
Regarding the ring topology's performance, the response time was greater as a result of the greater requirements in terms of equipment and devices [Fig. (d)]. The ring topology resulted in a larger mean response difference compared with the star topology. However, not employing ring topology in a central device would reduce the cost of the system, and, because the equipment used must only be connected to other equipment, fewer wires would be required. Furthermore, the ring topology has a wider range of scalability than the star topology. Even if the ring topology system affects the response time because of the number of devices, the average response difference for 100 devices was maintained below 0.3 ms as a result of the Gigabit Ethernet transmission interface. Within 800 units, the controlled response difference was less than 8 ms. The aforementioned improvements of the performance indices indicated that the proposed design achieves high reliability, real-time image



(a)



(b)



(c)

Fig. 13. Image streaming transmission verification: (a) using the UDP socket program and (b) and (c) using WireShark.

communication, and favorable task control in the IIoT system. Thus, the proposed machine vision system satisfies industrial fieldbus network and different topology requirements, demonstrating the practical value of the parameters.

4.4. *Innovative Advantage Analysis of Fieldbus-Based Machine Vision System*

The performance of five real-time Ethernet-based fieldbus networks is compared in [19,20], with EtherCAT determined to be the superior performance protocol. We validated our observed improvements in fieldbus-based machine vision system performance through a comparison of the system's performance with that of industry-related EtherCAT fieldbus products [40–43]. The innovative advantages of the fieldbus-based machine vision system proposed in this study were analyzed and are summarized in Table 4. The performance indices of the proposed solution were superior to those of other competing products in the market.

4.5. *Discussion*

As presented in Table 3, we compared systems documented in other

Table 2
The explanation of test parameters used in the study

Test parameter	Data size	Description
ICMP-Ping	32 byte	Error detection and reporting mechanism Internet Control Message Protocol (ICMP) The default ICMP packet size is 32 bytes [39].
Modbus TCP Mode	74 byte	To verify the functionality of Modbus/UDP, basic functions of Modbus registers are introduced according to Modbus/TCP. The Modbus message structure of Modbus TCP mode is directly modified on the basis of no increase in the communication process. Modbus/TCP embeds a standard Modbus data frame into a TCP frame, without the Modbus checksum. Modbus/TCP Application Data Unit (ADU) is composed of 7-byte Modbus Application Protocol Header (MBAP header for short) and protocol data unit (PDU). Function code (1-byte) and data fields (66-byte) are absorbed in PDU form. The MBAP header includes the following fields: Transaction Identifier (2 bytes): This identification field is used for transaction pairing; Protocol Identifier (2 bytes): This field is always 0 for Modbus services. Modbus/TCP= 0000; Length (2 bytes): Identifying the number of bytes in the message to follow; Unit Identifier (1 byte): Unit ID= 00;
640*480 Gray-scale Image	Requires 300 kB of storage	To verify the functionality of image streaming, the specialized function code (50, 0 × 32) of Modbus/UDP is predefined and the fulfillment of performance targets of the machine vision system. Uncompressed 8-bit grayscale images with a size of 640 * 480 pixels. (640 × 480 = 307200).
1920*1080 Gray-scale Image	Requires 2025 kB of storage	Uncompressed 8-bit grayscale images with a size of 1920 * 1080 pixels. (1920 × 1080= 2073600).

Table 3
Comparison of response times for different parameters.

Test parameter	Software System (ms)	Proposed hardware system (ms)
ICMP-Ping	0.474	0.069
Modbus TCP Mode	4.048	0.077
640 * 480 Gray-scale Image	7.277	3.283
1920*1080 Gray-scale Image	38.445	21.685

work with the proposed system by identifying technological innovation indices. The analysis of the innovative advantages of the system proposed in this manuscript can be summarized as follows:

(1) Real-time operation:

The fieldbus networks used in industry, such as EtherCAT and that are constructed for high-performance synchronous motion control, required a real-time cycle time for server motion control of less than 100 μs [41,42], up to 250 μs [40], or from 250 μs to 8 ms. For most traditional industrial automation applications, the cycle time requirement for 1,000 digital inputs and outputs is 32.5 ms and for 100-axis motion control is 125 ms; in regards to high performance, the uncoordinated motion requirement is 1 ms [43].

Furthermore, in terms of the response time, a complete communication process refers to polling using the high-speed hardware master and receipt of response messages from slaves. In this work, the mean response time was 69 μs, for the 640 × 480 Gray-scale image was 3.283 ms, which is considerably smaller than the industrial automation requirement. Thus, the response time meets the real-time requirement of industry.

(2) Reliability:

Modbus/UDP is fast, simple, and efficient because of the lack of retransmission, acknowledgement, and time out, all of which are characteristics of Modbus/TCP. A comparison of Modbus/UDP with Modbus/TCP is provided as follows.

- Modbus/TCP is a reliable service; thus, delays are introduced whenever a bit error or packet loss occurs. Therefore, Modbus/UDP is appropriate for use when speed takes precedence over reliability.
- Modbus/TCP is a point-to-point connection-oriented protocol and is unsuitable for broadcast and multicast transmission. Modbus/UDP, however, is recommended when using multicast topology because of its multipoint transmission capabilities [44]. Applications operating over multicast typically use Modbus/UDP, which is connectionless.
- Modbus/TCP can have numerous features (e.g., flow control, acknowledgement, and transaction logging) that machine vision systems do not require. These features affect the available bandwidth and change the delays on a link, providing another source of jitter. Modbus/UDP is often used to provide a fast image stream flow using no error control.
- Modbus/UDP is a packet-based, connectionless, best-effort service. A large amount of data is present in the machine vision system, and a bit error or loss of a packet often only introduces a minor break in the output. If a few packets do not transmit over the link, Modbus/UDP skips those packets and continues with the overall flow of the image stream rather than attempting to locate every detail of the original source.
- Modbus/UDP often uses packets of small size and with headers. Each Modbus/UDP packet on the network is composed of a small header and user data [45]. Because of the presence of small headers, Modbus/UDP processes the packets quickly. Thus, Modbus/UDP provides a low overhead and high speed.
- Image streaming communication has the advantage of not requiring a completely reliable transport level. Because of this, Modbus/UDP enables for the fastest and most simple transmission of image streaming data to the receiver, enabling an applied machine vision system to meet real-time requirements. We also conducted benchmark testing to determine the reliability of the proposed machine vision system, with the results indicating a success rate of 100% and retransmission and packet loss rates of 0. The present experiments demonstrated that our approach performed more highly than Modbus/TCP.

(3) Power consumption:

This study employed a high-speed hardware controller constructed with FPGAs of the same specification (EP4CE115F29C7, supply voltage: 1.2 V; maximum frequency: 200 MHz) without the extra CPU and operating system to prevent the piling up and switching of software stacks; the use of distributed local computing also considerably reduced the power consumption. Unlike the EtherCAT controller in [40] without the extra CPU, the piling up of software on top of the hardware architecture was avoided in this study. Moreover, in [43], a different processing operating system, employed to avoid the switching of software stacks, had to compete for the same resources. In the proposed approach, the number of available slaves does not limit our hardware controller because the controller is truly parallel. The combination of a reconfigurable FPGA hardware and cross-layer hardware controller enables parallel computing for communication between slaves. The power consumption of the proposed design was analyzed using the Early Power Estimator for the

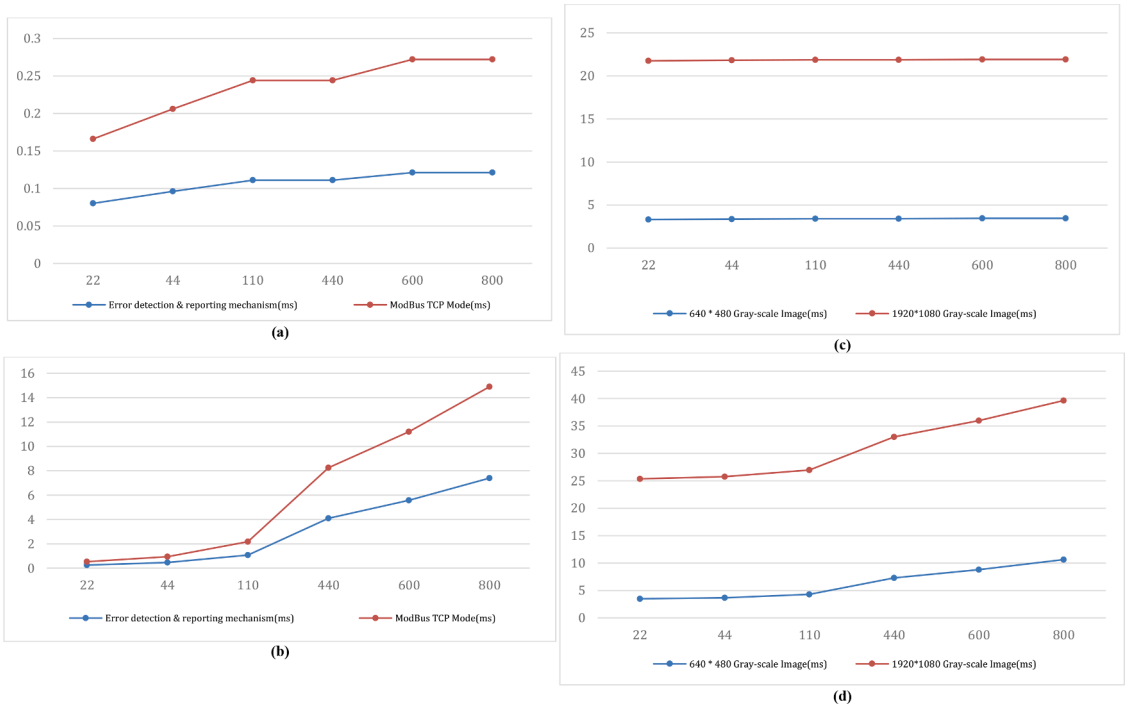


Fig. 14. Comparison of response times **for** different equipment amounts using (a) and (c) star topology and (b) and (d) ring topology.

Table 4
Comparative analysis of the proposed fieldbus-based machine vision technology.

	EtherCAT-based fieldbus technology	Proposed fieldbus-based machine vision system	Advantages
IIoT fieldbus hardware solution	Adopts EtherCAT-based Fieldbus (e.g., OMRON NX701 [40], 40 W and AXL F BK EC [41], 13.7 W)	Adopts Gigabit Modbus constructed with FPGAs of the same specification (maximum clock: 200 MHz; power consumption: 1 to 1.2 V)	High flexibility and scalability
Cycle time	From 125 to 250 μs [40, 42] Mean cycle time < 100 μs [41] Supports uncoordinated motion at 1 ms > [43]	<69 μs	High efficacy and efficiency
Edge computing	No	The edge computing of slaves is typically performed away from the centralized computing available in the cloud To effectively improve the response time of the device, the system uses pipelined hardware for both image packet processing and transmission	High computational efficiency and low bandwidth and power consumption
Integrated with machine vision	Mostly no Used with software-based applications [43]	Modbus UDP integrated with machine vision	High usability, adaptability, availability, and reliability
Time synchronous	<1 μs	<120 ns	High usability and reliability
Operating system	Requires an operating system	Does not require an operating system or software-based applications	High adaptability
Programing language	Must be written in a high-level programming language by the developer on the basis of operational demands (e.g., IEC 61131, C/C++, and MATLAB/Simulink)	With a built-in Modbus hardware controller, an additional programming language and high-level software function are not required	High usability and low cost
Network topology	Star topology	Satisfies the requirements of different topologies (e.g., star and ring topologies)	High adaptability and flexibility

entire system and was discovered to be considerably lower than that for the EtherCAT fieldbus coupler [41] and EtherCAT CPU module [40], which were 13.7 and 40 W, respectively. The experimental results revealed that the performance indices of the proposed system meet the industrial technological innovation thresholds.

5. Conclusion

This paper presents the design of an innovative fieldbus-based machine vision system using the Gigabit Modbus UDP fieldbus protocol. In the protocol, high-speed Gigabit Ethernet is integrated into a machine vision system, providing high speed, real-time operations, and reliability. Through distributed pipelined computing, the proposed system reduces the bandwidth costs of image packet transfer without employment of an extra CPU, an operating system, or software resources. Moreover, the subsequent feedback shortens the response time of the machine intelligence. In the proposed hardware architecture, cross-layer function controllers are employed to perform distributed and collaborative computing, improving the reliability of image transmission. The present results revealed that the performance of the proposed solution satisfies the requirements for technological innovation indices.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors would like to thank the anonymous referees and editor for their valuable comments.

References

[1] T. Qiu, Y. Zhang, et al., A Robust Time Synchronization Scheme for Industrial Internet of Things, IEEE Trans. Ind. Informat. (99) (2017) vol. PP1–1.
[2] C. Gündoğan et al., "The Impact of Networking Protocols on Massive M2M Communication in the Industrial IoT," in IEEE Transactions on Network and Service Management, doi: 10.1109/TNSM.2021.3089549.
[3] R. Rolle, V. Martucci, E. Godoy, Architecture for Digital Twin implementation focusing on Industry 4.0, IEEE Latin America Transactions 18 (05) (May 2020) 889–898, https://doi.org/10.1109/TLA.2020.9082917.
[4] M.S. Vikas, V. Vikram Athreya, Implementing SCADA System by Using Rexroth WinStudio for Automated Industrial Process, International Journal of Research in Engineering, Science and Management 3 (8) (2020) 287–289.
[5] S. Liu, X. Hao, Z. Zhang, 'Pedestrian retrieval via part-based gradation regularization in sensor networks, IEEE Access 6 (2018) 38171–38178.
[6] I. Al-Anbagi, M. Erol-Kantarci, H.T. Mouftah, 'Delay critical smart grid applications and adaptive QoS provisioning, IEEE Access 3 (1) (Aug. 2015) 1367–1378.
[7] Ioan Ungurean, Industrial Internet of Things with Data Distribution Service for Real Time Systems, International Research Journal of Engineering and Technology (IRJET) 02 (05 Aug 2015).
[8] Ju Ren, Hui Guo, Chugui Xu, Yaoxue Zhang, Serving at the Edge: A Scalable IoT Architecture Based on Transparent Computing, IEEE Network (2017), https://doi.org/10.1109/MNET.2017.1700030.
[9] Yaoxue Zhang, Ju Ren, Jiagang Liu, Chugui Xu, Hui Guo, Yaping Liu, A survey on Emerging Computing Paradigms for Big Data, Chinese Journal of Electronics 26 (1) (2017) 1–12.
[10] K. Xu, Y. Qu, K. Yang, 'A tutorial on the Internet of Things: Froma heterogeneous network integration perspective, IEEE Network 30 (2) (Mar. 2016) 102–108.
[11] Hugh Boyes, et al., The industrial internet of things (IIoT): An analysis framework, Computers in Industry 101 (2018) 1–12.
[12] Gannu Atharvan, et al., A way forward towards a technology-driven development of industry 4.0 using big data analytics in 5G-enabled IIoT, International Journal of Communication Systems (2021) e5014.
[13] Halim Khujamatov, et al., IoT, IIoT, and Cyber-Physical Systems Integration. Emergence of Cyber Physical System and IoT in Smart Automation and Robotics, Springer, Cham, 2021, pp. 31–50.
[14] Wenchao Xia, et al., Mobile edge cloud-based industrial internet of things: improving edge intelligence with hierarchical SDN controllers, IEEE Vehicular Technology Magazine 15 (1) (2020) 36–45.
[15] Soujanya Mantravadi, et al., Securing IT/OT Links for Low Power IIoT Devices: Design Considerations for Industry 4.0, IEEE Access 8 (2020) 200305–200321.
[16] Soujanya Mantravadi, et al., Design choices for next-generation IIoT-connected MES/MOM: An empirical study on smart factories, Robotics and Computer-Integrated Manufacturing 73 (2022), 102225.
[17] A. Dridi, et al., A Novel Deep Reinforcement Approach for IIoT Microgrid Energy Management Systems, IEEE Transactions on Green Communications and Networking (2021).
[18] Waleed Ali, et al., Hierarchical Control of Microgrid Using IoT and Machine Learning Based Islanding Detection, IEEE Access 9 (2021) 103019–103031.
[19] Real Time Automation, Inc. [Online]. Available: https://www.rtautomation.com/.
[20] APART, W. S. S. 5 Real-Time, Ethernet-Based Fieldbuses Compared. (2016).
[21] CAN in Automation. [Online]. Available: https://www.can-cia.org/.
[22] CC-Link [Online]. Available: https://www.cc-link.org/.
[23] EtherCAT Technology Group [Online]. Available: http://www.ethercat.org.
[24] ODVA. [Online]. Available: http://www.odva.org.
[25] Modbus Organization [Online]. Available: http://www.modbus.org/.
[26] POWERLINK [Online]. Available: http://www.ethernet-powerlink.org.
[27] PROFIBUS [Online]. Available: www.profibus.com.
[28] Sercos tech.[Online]. Available: https://www.sercos.org/.
[29] M.H. Syed, et al., Real-Time Coupling of Geographically Distributed Research Infrastructures: Taxonomy, Overview, and Real-World Smart Grid Applications, IEEE Transactions on Smart Grid 12 (2) (March 2021) 1747–1760, https://doi.org/10.1109/TSG.2020.3033070.
[30] B. Dean, M. Starke, M. Smith, M. Chinthavli, L. Tolbert, A Communication Testbed for Testing Power Electronic Agent Systems, in: 2021 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT), 2021, pp. 1–5, https://doi.org/10.1109/ISGT49243.2021.9372188.
[31] Alexandru Ioana, Camelia Burlacu, Adrian Korodi, Approaching OPC UA Publish–Subscribe in the Context of UDP-Based Multi-Channel Communication and Image Transmission, Sensors 21 (4) (2021) 1296.
[32] Armin Veichtlbauer, et al., Open Information Architecture for Seamless Integration of Renewable Energy Sources, Electronics 10 (4) (2021) 496.
[33] Yunhong Gu, Robert L. Grossman, UDT: UDP-based data transfer for high-speed wide area networks, Computer Networks 51 (7) (2007) 1777–1799. J. Cecilio and P. Furtado, "Architecture for uniform (re) configuration and processing over embedded sensor and actuator networks," *IEEE Transactions on Industrial Informatics*, Vol. 10, No. 1, pp. 53-60, Feb. 2014.
[34] Hongjun Zhang, et al., FastUDP: a highly scalable user-level UDP framework in multi-core systems for fast packet I/O, The Journal of Supercomputing 77 (5) (2021) 5148–5175.
[35] C.-H. Chen, M.-Y. Lin, C.-C. Liu, Edge Computing Gateway of the Industrial Internet of Things Using Multiple Collaborative Microcontrollers, IEEE Network 32 (1) (2018) 24–32.
[36] C.-H. Chen, M.-Y. Lin, X.-C. Guo, High-performance fieldbus application-specific integrated circuit design for industrial smart sensor networks, The Journal of Supercomputing (2017) 1–19.
[37] C.-H. Chen, M.-Y. Lin, W.-H. Lin, Designing and implementing a lightweight WSN MAC protocol for smart home networking applications, Journal of Circuits, Systems and Computers 26 (03) (2017), 1750043.
[38] C.-H. Chen, M.-Y. Lin, X.-C. Guo, High-level modeling and synthesis of smart sensor networks for Industrial Internet of Things, Journal of Computers and Electrical Engineering (CAEE) (2017) 48–66, https://doi.org/10.1016/j.compeleceng.2017.06.005.
[39] Sayed Qaiser Ali Shah, Farrukh Zeeshan Khan, Muneer Ahmad, The impact and mitigation of ICMP based economic denial of sustainability attack in cloud computing environment using software defined network, Computer Networks 187 (2021), 107825.
[40] Omron Co., Ltd. Omron NX701 EtherCAT CPUmodule [Online]. Available: https://www.omron.com.tw/products/family/3436/specification.html.
[41] PHOENIX CONTACT Co., Ltd.: AXL F BK EC - 2688899 [Online]. Available: https://www.phoenixcontact.com/online/portal/tw/?uri=pxc-oc-itemdetail:pid=2688899&library=twzh&pcck=P-08-12-12-01&tab=1&selectedCategory=ALL.
[42] Advantech Co., Ltd.: PCIE-1203L [Online]. Available: http://advdownload.advantech.com/productfile/PIS/PCIE-1203L/Product%20-%20Datasheet/PCIE-1203L20180910103050.pdf.
[43] KINGSTAR Co., Ltd. [Online]. Available: https://kingstar.com/products/machine-vision/.
[44] Dhiman Deb Chowdhury, Packet Timing: Network Time Protocol. NextGen Network Synchronization, Springer, Cham, 2021, pp. 103–116.
[45] Yuli Yang, Lajos Hanzo, Permutation-Based TCP and UDP Transmissions to Improve Goodput and Latency in the Internet-of-Things, IEEE Internet of Things Journal (2021).



Ming-Yi Lin received Master degree from Yuan-Ze University and received Ph.D. degree of computer science from National Central University in 2011 and 2016, respectively. He is now an Assistant Professor in the department of computer science and information engineering at Chinese Culture University, Taiwan. He was an Assistant Professor in the department of computer science and information engineering, Taipei City University of Science and Technology, Taiwan, R.O.C. from 2021. His research interests include embedded systems, smart sensor networks, and Internet of Things.



Ching-Han Chen received the Ph.D. degree in 1995 from the Franche-Comte University, France. He was an Associate Professor in the department of electrical engineering, I-Shou University, Taiwan, R.O.C. from 1995. He is now a Professor in the department of computer science and information engineering. His research interests include embedded system design, multimedia signal processing, robotics.



Zong-Bin Dong received Master degree of computer science from National Central University in 2017. His research interests include embedded system, Fieldbus protocol, and industrial sensor network.



Chun-Cheng Chen received received Ph.D. degree of National Yunlin University of Science and Technology in 2018. He is now an Assistant Professor in the department of computer science and information engineering at Taipei City University of Science and Technology, Taiwan. His research interests include high-level thinking, creative thinking, critical thinking, problem-solving ability, invention patent.