applied sciences

MDPI

Article

Design of Remote Upgrade System for Data Processing Unit in Marine Engine Room Simulator

Hong Zeng *, Hui Liu, Jundong Zhang, Minglu Sun and Tianjian Wang

College of Marine Engineering, Dalian Maritime University, Dalian 116026, China
* Correspondence: zenghong@dlmu.edu.cn

Abstract: With the development of ship intelligence, the frequency of upgrading the marine engine room simulator, which is essential for crew training, has increased. Traditionally, the data processing unit (DPU) of the marine engine room simulator is upgraded by manually downloading the firmware. This makes the hardware maintenance high-cost. In this paper, we first propose a WAN-based firmware upgrade system to enable secure over-the-air upgrades of DPUs and reduce operation and maintenance costs. A distributed hardware structure is given to manage DPU in the simulator via the Internet. We have designed two methods of firmware upgrades, automatic upgrades and remote upgrades. In automatic upgrades, the DPU can download new firmware upgrades from the web server through the router. By designing a series of mechanisms including code rollback, code backup and code confirmation, the In-Application Programming (IAP) technique is realized through the Internet. Firmware upgrades have good fault tolerance mechanisms to ensure that the emulator can still work in the event of an upgrade error. In remote upgrades, we upgrade the DPU firmware through the remote control center. We assessed the performance of the system by measuring the success rate of DPU upgrades, upgrade time and performance after the upgrade. The results show that the DPU upgrade success rate is close to 100% and performance is as good as expected. The results show that the remote firmware upgrade system proposed in this paper is reliable and practical.

Keywords: simulator; DPU; firmware upgrade; code backup; over-the-air

Citation: Zeng, H.; Liu, H.; Zhang, J.; Sun, M.; Wang, T. Design of Remote Upgrade System for Data Processing Unit in Marine Engine Room Simulator. *Appl. Sci.* **2022**, *12*, 9107. https://doi.org/10.3390/app12189107

Academic Editor: Eui-Nam Huh

Received: 26 July 2022
Accepted: 7 September 2022
Published: 10 September 2022

1. Introduction

The marine engine room simulator has been popular in the training and competency evaluation for seafarers [1]. With the continuous optimization of each subsystem in the marine engine room simulator, it is unavoidable to upgrade the model, control algorithm, and communication protocol. The conventional solution is to manually update the firmware in the field. However, it has high maintenance costs and poor efficiency [2]. The existing architecture no longer meets the needs of the intelligent development of the industry. On the other hand, with the new generation of information technology and the necessity to upgrade the simulator, a flexible framework is needed to support and promote development.

In the Internet of Things (IoT), remote firmware upgrade is one of the more critical research elements [3]. A Malware called Mirai infected these devices and launched the distributed denial-of-service (DDoS) attacks [4]. The number of attacks involving IoT devices during 2018 increased, with 32.7 million IoT incidents reported last year [5]. People are starting to worry about security vulnerabilities and feature upgrades in IoT placements. Therefore, the need for firmware upgrades over the air (OTA) has been added to the requirements for embedded software. In terms of security, the firmware of IoT devices is securely upgraded using the LoRa communication protocol through blockchain framework technology [6]. However, collaboration through numerous gateways is desired to increase the firmware upgrade process's reliability and performance. Heeger et al. designed a secure and reliable firmware update process for low-power wireless IoT devices using an adaptive

data rate algorithm [7]. It offers the possibility of firmware upgrades for energy-constrained IoT devices.

The firmware upgrade is a key technology in IoT [8]. Internet of Things (IoT) devices rely upon remote firmware upgrades to fix bugs and make security enhancements [9]. However, there are few research studies on the remote upgrade of the data processing unit (DPU) in the marine engine room simulator. In addition, there is also no standard firmware upgrade and specification for simulator-related fields. In order to reduce the number of times that engineers solve firmware problems on site, it also provides a new method for remote operation and maintenance of marine engine room simulators. In this paper, we propose a system for remote firmware upgrades for DPUs. The system not only solves the drawback of unstable and unreliable firmware upgrades in the marine engine room simulator but also realizes the firmware upgrade in the wide area network.

In summary, we make the following contributions:

1. We first propose a system for enabling firmware upgrades for the simulator. The problem of untimely and inconvenient firmware upgrades for existing emulators is solved. The intelligence of the marine engine room simulator is improved.
2. We have improved the availability of firmware upgrades by the remote upgrade configuration, as well as automatic upgrades by networking.
3. We have improved the reliability of program upgrades through program verification as well as program backups, ensuring that the simulator will still work in the event of an upgrade failure.

The paper includes the following content. The introduction underlines the significance and relevance of the chosen topic. Then, a remote upgrade system for marine engine room simulators is presented based on the evolution of important technologies employed in other fields. The second part, explains the simulator's components, structural layout, and network topology. In the third part, a scheme is put forward that uses functional design and technical support to remotely upgrade firmware. In the fourth part, Testing and Discussion describes the performance tests after the DPU firmware upgrade. We also analyze its test results. The results are as expected. Finally, the conclusion is a summary evaluation of the whole scheme.

2. Related Works

The evolution of IoT deployments has raised the need for effective methods of updating device firmware. As a result, innovative solutions for remote firmware upgrades have attracted the attention of many researchers. Current research covers many aspects of the upgrade process, such as the security of performing the process, the importance of remote firmware upgrades, and different aspects of upgrade efficiency [9].

Ref. [10] presents a model for LoRaWAN that can be used for firmware transfers over low data rate, constrained remote wide area networks (LoRaWAN). The proposed model ensures integrity and availability during firmware upgrades. However, it does not work very well for the transmission of larger firmware.

Ref. [11] propose a blockchain-based OCF firmware upgrade solution for IoT devices. They introduce two types of firmware upgrade protocols, direct and peer-to-peer upgrades. In the direct scheme, devices can download a new firmware upgrade from a server via an IoT gateway. In the peer-to-peer scheme, devices can query for upgrades from a nearby gateway. The results show that they can complete firmware upgrades in a reasonable amount of time, with peer-to-peer taking less time.

Secure and reliable firmware upgrade technology for vehicles is also being investigated [12]. It can be utilized to reduce upgrade maintenance costs. In their studies, the firmware is upgraded over Wi-Fi on a vehicle that already has the most recent firmware [13]. There, vehicle owners do not need to return to a dealership for a firmware upgrade, which costs around 150 dollar [14]. The cost of an upgrade utilizing the proposed system depends on the number of users. However, they only proposed a simple TCP triple handshake for secure connections between vehicles. The security is yet to be verified. Marco Steger

proposes a wireless upgrade system based on an IEEE 802.11 s mesh network and describes related high-level requirements for such a system [15]. However, its availability is currently insufficient. In addition, its interconnection with other current automotive networks is difficult to achieve.

Marine engine room simulators can provide a highly realistic experience for seafarer training in the operation and maintenance of marine vessels. As it is a widely used simulation device, we have to consider an efficient and economical way of upgrading the firmware. We focus on the reliability and stability of the upgrade process to ensure the proper functioning of the equipment.

3. Description of Simulator

The system is designed for the DPU in the marine engine room simulator. It has been widely applied in maritime education [16]. Figure 1 is the typical layout of the marine engine room simulator. It includes a number of semi-physical simulation equipment as shown in Figure 2 and an M-side running on a computer as shown in Figure 3. In the simulator simulation system, the higher computer is mainly the M-side. It is a virtual software developed in the laboratory that is capable of highly modeling the various devices in the cabin of a ship's vessel. Its simulation is based on the mother ship's marine system, including the main engine system, main generator system, boiler system, oil and water separation system, and other subsystems. The lower computer is mainly composed of semi-physical simulation control boxes, consoles, and other simulated cabin equipment. Moreover, they have a high degree of likeness and operability to marine engine room equipment and are effective for teaching and evaluating seafarers' competencies [17].

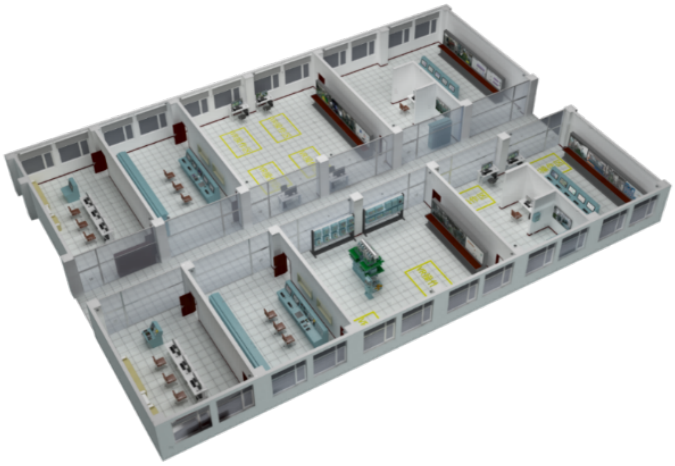

Figure 1. Typical layout of marine engine room simulator.

Figure 2. Semi-physical simulation equipment.

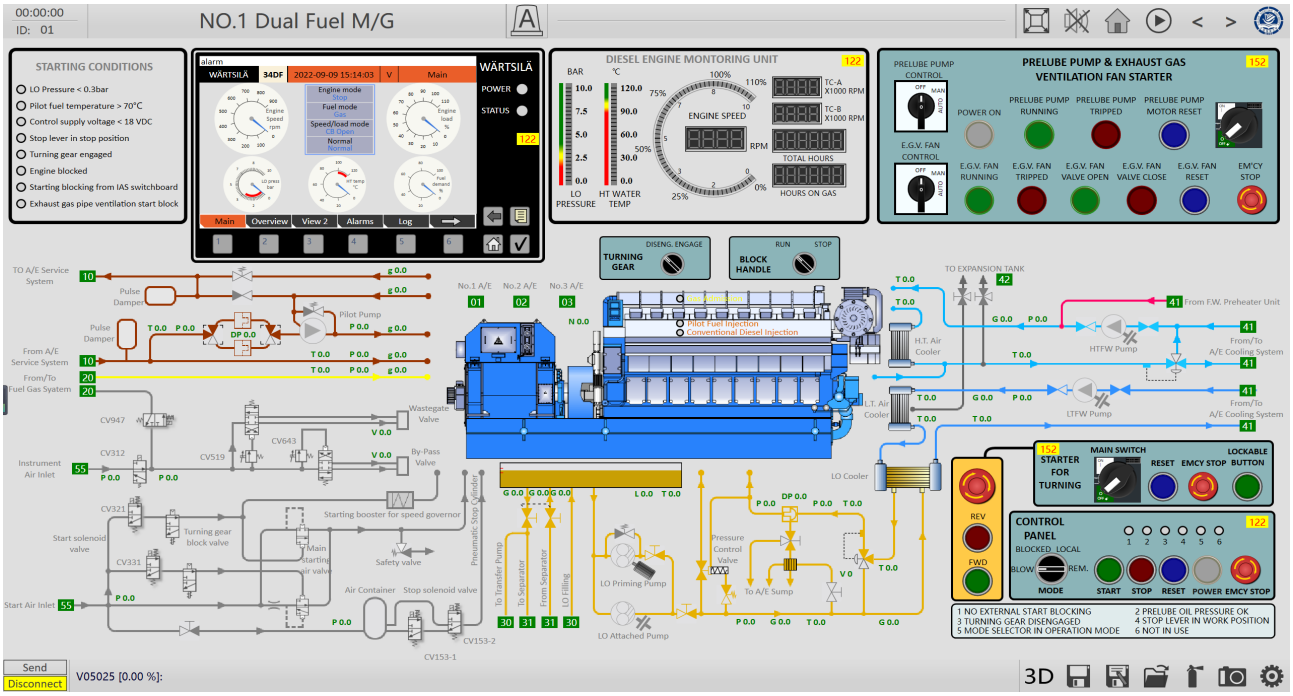

Figure 3. 2D page of dual fuel engine generator in marine engine room simulator.

The DPU is the key for synchronizing information between the upper and lower computer [18]. The data of semi-physical simulation equipment is collected through the DPU and transmitted to the upper computer via Ethernet. Action commands from the upper computer are also transmitted to the DPU via Ethernet. Through its processing, the action commands are synchronized and reflected in the semi-physical simulation equipment based on the electrical reaction.

The marine engine room simulator has to be linked to the LAN through the network interface in order to exchange data with the top computer. Once connectivity has been established, the simulator may be utilized for maritime instruction and competency evaluation. The DPU can access the Web server when the LAN is linked to the Internet.

4. Firmware Remote Upgrade Design

4.1. Network Architecture of the Firmware Upgrade Protocol

Figure 4 shows the network architecture of the remote upgrade system. The network architecture consists of three parts: the web server, remote upgrade configuration, and the DPU in the marine engine room simulator. The web server is a cloud-based, remote network server with a public IP address. Its main work is to transfer and store upgraded data files. The remote upgrade configuration is an application created in C# that is mostly used for sending upgrade instructions and receiving returned data [19].

The DPU in the marine engine room simulator consists of a Flash memory and a communication module. Flash is mainly used to store program files and key informationr [20]. The communication module has a domestic TCP/IP protocol stack for communication with the web server [21].

When the marine engine room simulator starts up, the DPU can connect to the web server via the communication module. It sends the agreed-upon commands to the web server, requests the program version, and receives information about the upgrade file. If a firmware upgrade is necessary, the device will set the upgrade flag "UpgradeState" to 1 and simultaneously switch the work area [22]. Then, the web server is requested to deliver the most recent version of the upgrade file. After a file is received, it is validated and written to the Flash runtime. At this moment, the remote upgrade is completed.

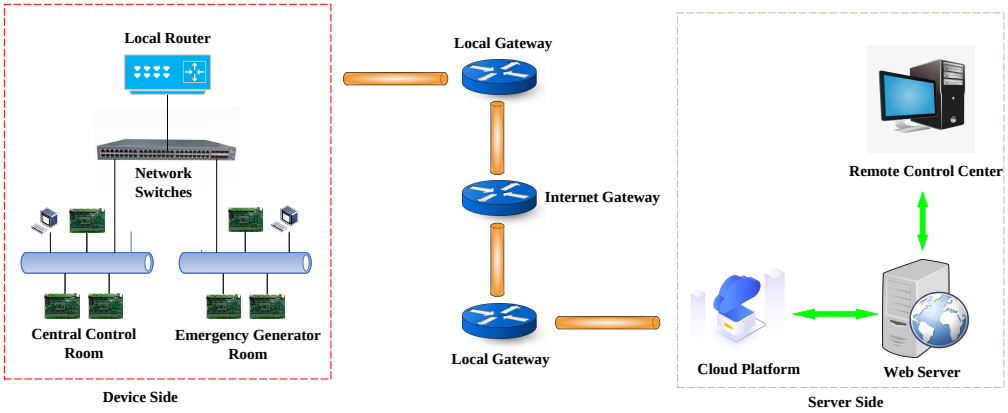

Figure 4. Network architecture of the remote upgrade system.

4.2. Function Design

4.2.1. Flash Partitions

Flash memory is a safe and fast storage entity that has become the most essential carrier of data and programs in embedded systems. In this paper, it mainly stores two parts of code: the bootloader and the user application [23]. The bootloader is the initial code when the device is started up. The user application is the code executed by the embedded system after the bootloader has completed [24]. There are eight sectors in the flash main memory block comprising four 32 KB sectors, one 128 KB sector, and three 256 KB sectors.

Combining the size of Flash space required for each part of this system, the Flash of the internal chip of the DPU is divided into four parts: the bootloader program area; the information reserved area; the APP program running area; and the APP program backup area [25]. The Flash memory partitions are shown in Table 1.

Table 1. Flash Partitions.

Title	Components	Address Interval	Size/KB
bootloader	Bootloader Program Area	0X08000000–0X0800FFFF	64
user application	Information Reserved Area	0X08010000–0X08011FFF	8
	APP Program Running Area	0X08012000–0X0807FFFF	440
	APP Program Backup Area	0X08080000–0X080FFFFF	512

The Flash size of the chip is 1024 K bytes, so the addresses reserved for the APP program running area and program backup area are enough. Since the reserved information area will not be erased during normal work, it can be for storing key information, such as upgrade flags, upgrade file versions, program download addresses, etc. To prevent the loss of critical information after an upgrade, we should allocate an appropriate amount of space to the reserved area.

4.2.2. Interface Design

Figure 5 is the interface of the remote upgrade configuration. Message Queuing Telemetry Transport (MQTT) is a lightweight communication protocol for embedded devices [26]. The remote upgrade configuration can communicate with the web server and DPU via this protocol [27]. It identifies senders and recipients by including a "Subject" tag in the application's messages. Using its unique device number, every device subscribes to at least one subject. MQTT defines three quality-of-service (QoS) levels: QoS 0, QoS 1, and QoS 2 [28]. In QoS0 (at most once), the publisher sends a message once to the broker,

which sends a message once in turn to the subscriber. The receiver does not send an acknowledgment, and the sender does not resend the message. QoS 1 (at least once) is the standard level used in MQTT, which guarantees that PUBLISH messages are delivered at least once. Finally, when messages are lost and duplicates are not acceptable, QoS 2 (exactly once) is used. QoS 2 performs a two-step acknowledgment (PUBLISH-PUBREC) and (PUBREL-PUBCOMP) process to ensure that messages are delivered once. The system employs the highest level of quality of service ("QoS 2"). This reduces the likelihood of upgrading issues caused by improper upgrade instruction.

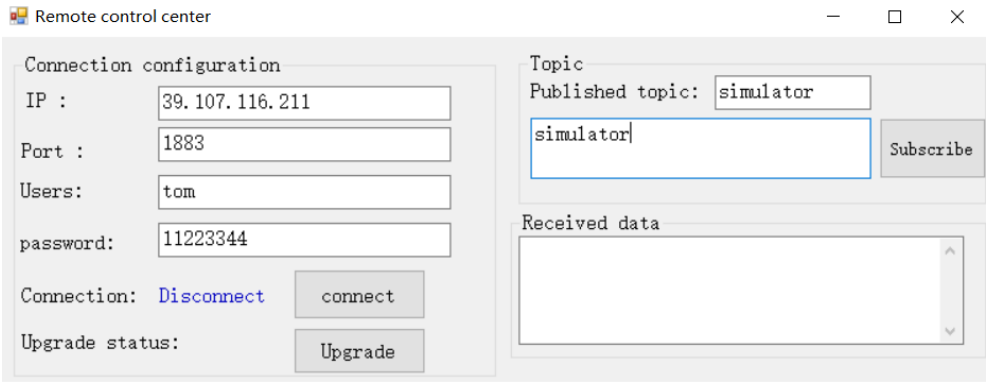

Figure 5. Remote upgrade configuration.

4.3. Remote Upgrade Program Design

Within the Local Area Network, the DPU will first accomplish the activities associated with the marine engine room simulator [29]. If the simulator is linked to the Internet and communicates with the Web server, the DPU can start a remote upgrade. First, DPU will send the "GET" seek to the Web server, which returns the file number, file length, and the beginning and ending addresses. When the DPU needs to upgrade based on a comparison of the file number, it will set the upgrade flag "UpgradeState" to 1 and simultaneously switch the work area. Then, it enters the bootloader program section and queries the command to request the server to deliver the most recent version of the program file. The server gets the command and sends the "Devicenumber1" program file to the simulator device. If DPU fails to obtain the upgrade file within the allotted time, the "Overtime" counter will increase by 1. If the "Overtime" counter value is more than 5, the remote upgrade fails and returns the corresponding fault code. If a server-sent upgrade file is received, the "Overtime" counter will be reset to 0 and the received program upgrade file will be verified. If the verification result is incorrect, the upgrade process will be stopped. If accurate, the subsequent data will be sent and written to flash. When DPU receives and verifies the "Devicenumber1" program file, it returns the command "upgrade successful" and restarts and runs the upgraded application. The system's upgrading flow chart is depicted in Figure 6.

The program upgrade has a good fault tolerance mechanism. If the program fails to receive, the backup program will be rewritten into the APP running area. Jumping between programs ensures that the marine engine room simulator will continue to work despite a failed upgrade [30].

4.4. Reliability Design

The program upgrade files are transmitted in segments during the transmission process. Consequently, packet loss [31], equipment disconnection, garbled coding, and other communication issues may occur, resulting in an incomplete delivery of the upgrade file. Furthermore, the transmission of program files differs from that of generic parameter files. It is vital to pay close attention to the reliability and real-time [32] of the transmission process; otherwise, it will interfere with the engine room simulator's typical teaching function.

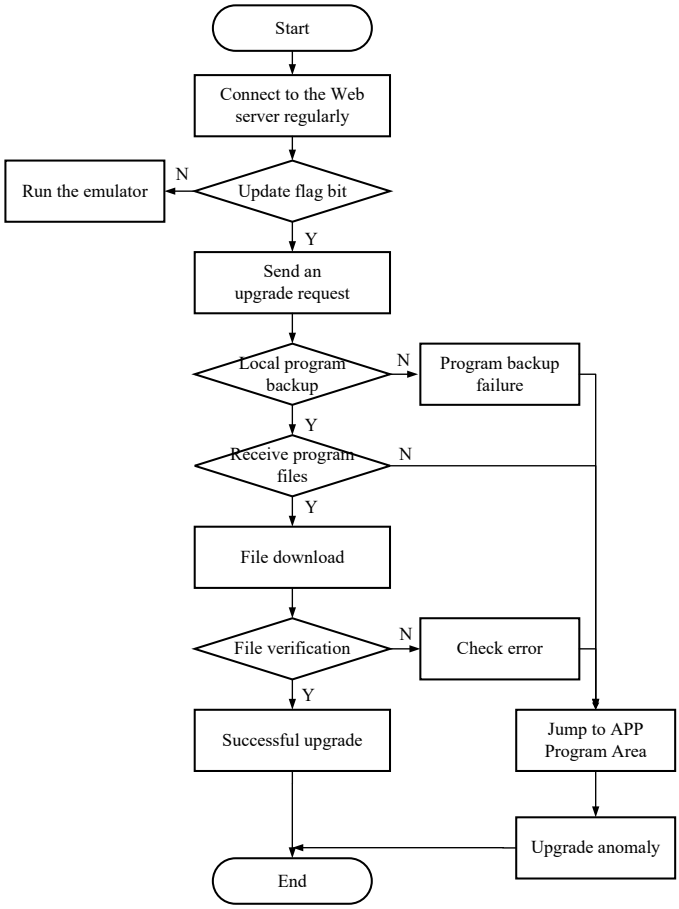

Figure 6. Flow chart of firmware upgrade.

4.4.1. Program Validation Mechanism

The program upgrade file is the code for the work of the DPU. Once a transmission error occurs, the marine engine room simulator will not work properly. Therefore, ensuring the accuracy of the transmission is the key that the upgraded marine engine room simulator can work normally. The cyclic redundancy check (CRC) is a common data confirmation algorithm that can effectively ensure the data integrity of the upgrade file during transmission [33]. In the CRC codes in this paper, the data of the transmission frame is 128 bytes. When the last byte of the transmission frame is less than 128, it can be filled with 0XFF to ensure that the data check is 128 bytes. When the DPU has received a frame of data transmission, it checks the transferred data and gains the results "Checksum". Compared with the check value "ReadCheckdate" read from the program upgrade file, If "Checksum" equals "ReadCheckdate", the data is correct and the received data is written to the Flash.

4.4.2. Backup Upgrade Mechanism

Watchdog is a timer circuit that is designed to protect the device from entering a dead cycle [34]. It is necessary to run the "watchdog" in the program runtime area. On the one hand, this is to prevent the download of buggy firmware that could prevent them from working properly. On the other hand, it is prevented from failing to upgrade the DPU remotely, which will cause the marine engine room simulator cannot work properly. If the marine engine room simulator is working properly, the program will "feed the dog" at regular intervals. Once the "dog feeding" timeout has expired, the program is running incorrectly. The DPU will be restarted and the previous version of the program in the backup area will be rewritten into the APP program run area. The work flow is shown in Figure 7.

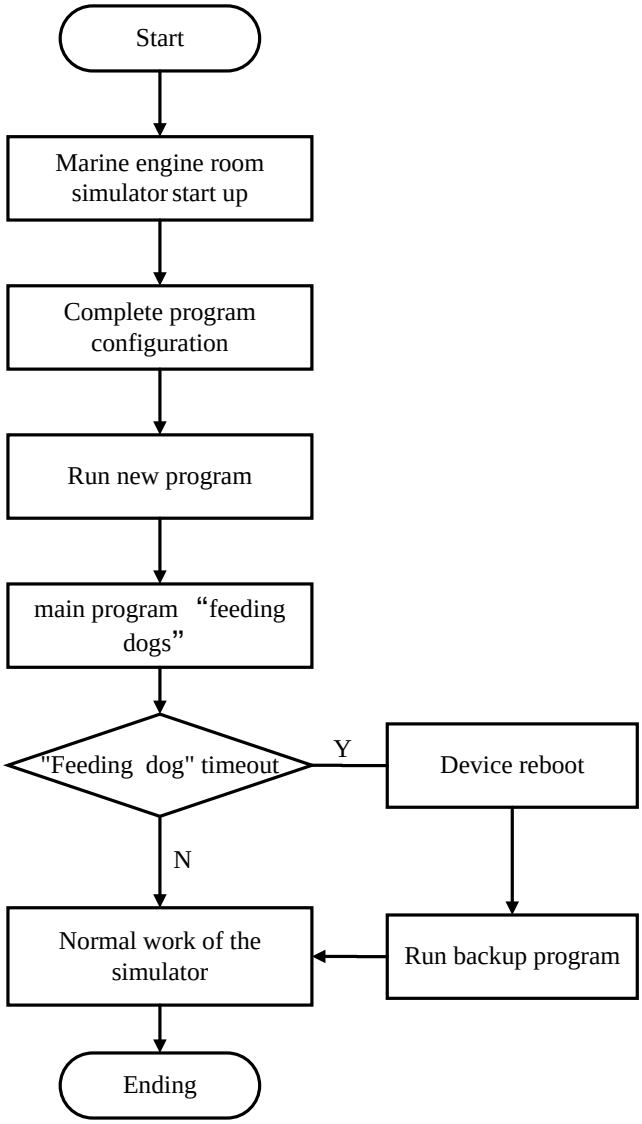

Figure 7. Flow chart of backup upgrade.

If the upgraded program fails, the backup upgrade mechanism ensures that the DPU can revert to the original successfully running program. It avoids the possibility of the marine engine room simulator that does not work properly due to program problems, which increases the reliability of remote upgrades.

5. Testing and Applications

5.1. Firmware Upgrade Function Test

To ensure the reliability and stability of the remote upgrade system, fifteen DPUs in the marine engine room simulator were selected for testing. Figure 8 is the Photo of the Data Processing Unit. We tested the DPU with firmware upgrades [35] in two ways. In normal upgrades, we select different sizes of firmware for the DPU. In abnormal upgrades, we first disconnected the communication module and the web server individually during the upgrade process to simulate a communication failure. Then, we manually downloaded the buggy firmware to simulate a human error. Table 2 shows the results of each DPU's upgrade test under different conditions.

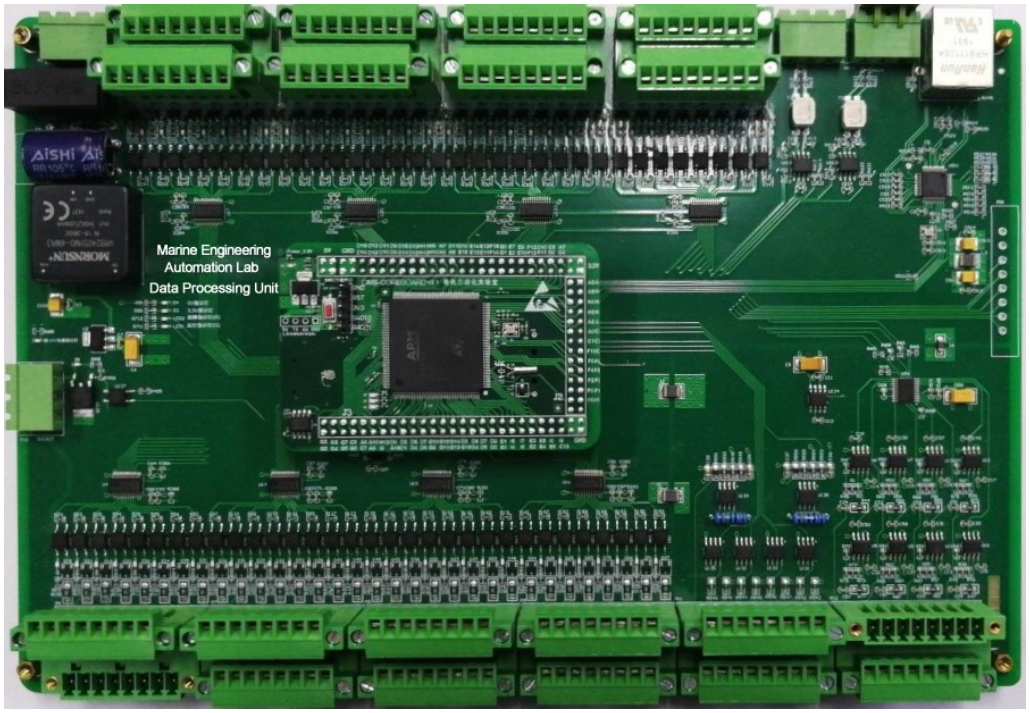

Figure 8. Data Processing Unit.

Table 2. DPU upgrade testing results.

Title	Faults Description	File Size (KB)	Success Times	Success Rate (%)
abnormal upgrades	Network disconnection	270	14	93
	Web server error	270	14	93
	pgrade buggy firmware	270	15	100
normal upgrade		40	15	100
		270	15	100

When the network is disconnected, the DPU will seek to connect to the network every 30 s. If the network is disconnected for too long and the DPU is unable to access the web server, the upgrade will fail. This is the reason why DPU upgrades failed in Table 2. However, if it is connected to the network after a period of time, the DPU can restart the upgrade request to the web server and complete the remote upgrade. In summary, the remote upgrade system has a high upgrade success rate and will ensure that the DPU completes the firmware upgrade successfully.

Table 3 shows the types of faults that can occur during the upgrade process and the handling mechanism of DPU. Upgrade faults are divided into data communication faults and human operation faults during the upgrade process [36]. Data communication faults include: web server errors during remote upgrades, sudden network interruptions, etc. When such types of failure occur, DPU will automatically upgrade the upgrade flag "UpgradeState" to 1 after timeout detection and Wait for communication reestablished to complete the firmware upgrade. Undetected buggy firmware deployments are common human issues in existing systems [37]. The buggy firmware is the wrong program that will prevent the DPU from working. It can be resolved by the backup upgrade.

Table 3. Analysis and treatment of upgrade faults.

Upgrade Faults	Faults Description	Handling Mechanism
Human operation faults	Upgrade buggy firmware	Backup upgrade
Data communication faults	Code verification error Network disconnection Web server error	Code retransmission Communication reestablished Re-upgrade

5.2. Remote Upgrade Stability Testing

The main work of DPU is the collection and output of analog and digital quantity [38]. In order to judge the working capability of DPU after a firmware upgrade. In this work, we test DPU analog output and digital output under fixed load conditions and discuss the test results. T_c is the time interval value between the previous frame of data sent from the client side and the next frame of data. T_s is the time interval value between the previous frame of data sent from the server side and the next frame of data. The quality of the output data is determined by two main performance indicators:

1. By comparing two adjacent data transmission time interval values T_s with the corresponding two adjacent response data reception time interval values T_s', that is, two transceiver interval values. It is permissible to indicate the changing state of performance of the data processing element under specific load test conditions.
2. By analyzing the pattern of variation in the time difference between the packets sent and the response packets received, that is, the pattern of variation in the value of the single send/receive process. These can indicate the current data processing rate of the DPU under fixed load test conditions.

In single channel mode, the average period T_c' of the current output performed by the AO port is 60.28 ms. We use test software to send data at a fixed load, and statistics the interval between sending and receiving data by DPU. According to Figure 9a, the actual average of the two nearest data sending intervals T_s is 60.10 ms, while the average of the two nearest data receiving intervals T_s' is 60.14 ms. These two values are roughly equivalent and are closer to the output period of the port. According to Figure 9b, the response interval between each set of data sent by the software and the response data received by the software from the data interaction unit is about 33.06 ms, and the data transmission process is relatively stable.

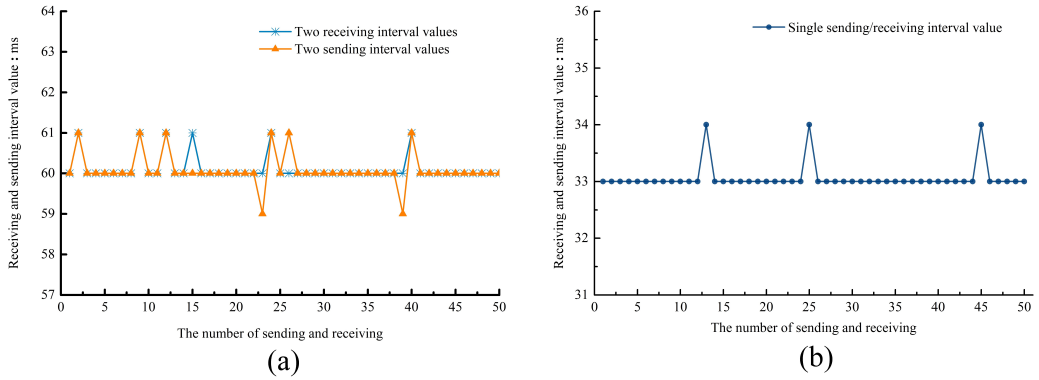

Figure 9. The interval values are the time for the test software to send and receive data under fixed load. Judging the capability of the DPU's analog output by its two interval values. (**a**) Statistics of AO sending and receiving interval values. (**b**) Statistics of AO sending and response data intervals.

In single channel mode, the DPU's DO port outputs high and low levels and the average period T_c' is 63 ms. The high voltage is 24 v, whereas the low voltage is 0 v. We also use test software to send data at a fixed load, and statistics the interval between sending and receiving data by DPU. According to Figure 10a, the software's average near data send

interval T_s is 62.04 ms and its average near data receive interval T_s' is 62.12 ms. These two values are roughly equivalent and close to the output period of the port. According to Figure 10b, the response interval for each set of data sent by the software and the response data received by the software from the DPU was approximately 42.08 ms, with less variance during the data transmission process.

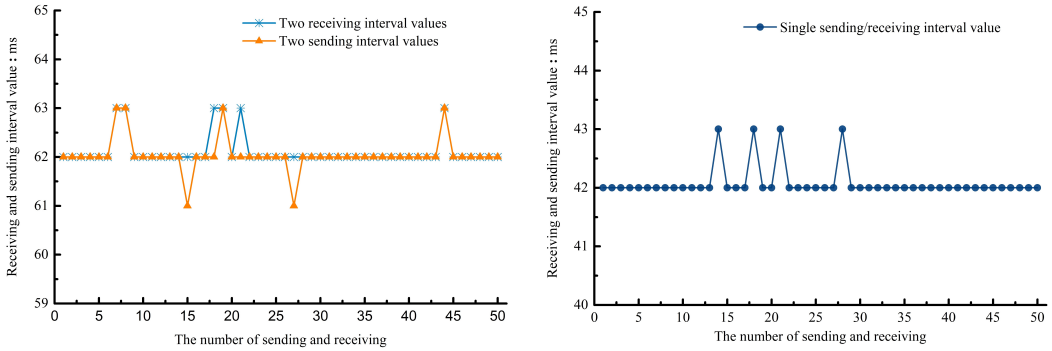

Figure 10. The interval values are the time for the test software to send and receive data under fixed load. Judging the capability of the DPU's digital output by its two interval values. (**a**) Statistics of DO sending and receiving interval values. (**b**) Statistics of DO sending and response data intervals.

The analysis shows that the firmware upgraded for the DPU has stable performance and can complete the AO and DO functions accurately.

6. Conclusions

The Remote Upgrade System can upgrade the firmware of the DPU in the marine engine room simulator and fix software problems in a timely manner. It enhances the intelligence and practicality of the marine engine room simulator. It eliminates the need to dispatch laboratory workers to the field to deal with malfunctions of the marine engine room simulator, which significantly increases personnel productivity. It eliminates the need to dispatch laboratory workers to the field to deal with malfunctions of the marine engine room simulator, which significantly increases personnel productivity. We are also able to carry out remote maintenance of the DPU at a lower cost. The system is equipped with a program backup and data validation mechanism to enhance upgrade integrity and reliability, ensuring that the simulator will still work in the event of an upgrade failure. We have improved the availability of DPU firmware upgrades through both manual and automatic upgrades. The system also operates with the expected stability. However, the security of data transmission has not been investigated in this paper. We cannot ensure the security of data transmission. In the future, the security of data transmission is also studied, using encryption algorithms (e.g., AES, DES) and other methods to ensure the reliability of the program during transmission. we also optimize the design of the server side, which can provide more functions, including status detection of the DPU and performance analysis.

Author Contributions: Conceptualization, H.L., H.Z. and J.Z.; methodology, H.L. and H.Z.; software, H.L.; validation, H.L. and H.Z.; formal analysis, H.L.; investigation, H.Z.; resources, H.Z.; data curation, H.L.; writing—original draft preparation, H.L., M.S. and T.W.; writing—review and editing, H.L. and M.S.; visualization, T.W.; supervision, H.Z.; project administration, H.Z. and J.Z.; funding acquisition, H.Z. and J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by a grant from High Technology Ship Research and Development Program of Ministry of Industry and Information Technology of China (CJ02N20).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The processed data cannot be shared at this time as the data also forms part of an ongoing study.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Shen, H.S.; Zhang, J.D.; Yang, B.C.; Jia, B.Z. Development of an educational virtual reality training system for marine engineers. *Dev. Educ. Virt. Reality Training Syst. Mar. Eng.* **2019**, *3*, 580–602. [CrossRef]
2. Tan, Y.H.; Niu, C.Y.; Tian, H.; Zhang, J. A Digital Twin Based Design of the Semi-physical Marine Engine Room Simulator for Remote Maintenance Assistance. In Proceedings of the 5th International Conference on Vision, Image and Signal Processing, Kuala Lumpur, Malaysia, 18–20 December 2021; pp. 137–141.
3. Zandberg, K.; Schleiser, K.; Acosta, F.; Tschofenig, H.; Baccelli, E. Secure Firmware Updates for Constrained IoT Devices Using Open Standards: A Reality Check. *IEEE Access* **2019**, *7*, 71907–71920. [CrossRef]
4. Eustis, A.G. The Mirai Botnet and the Importance of IoT Device Security. In Proceedings of the 16th International Conference on Information Technology-New Generations (ITNG 2019), Las Vegas, NV, USA, 1–3 April 2019; Advances in Intelligent Systems and Computing; Volume 800. [CrossRef]
5. SonicWall. 2018 SonicWall Annual Threat Report. Available online: https://d3ik27cqx8s5ub.cloudfront.net/sonicwall.com/media/pdfs/resources/2018-snwl-cyber-threat-report.pdf (accessed on 20 July 2022)
6. Pieroni, A.; Scarpato, N.; Felli, L. Blockchain and IoT Convergence—A Systematic Survey on Technologies, Protocols and Security. *Appl. Sci.* **2020**, *10*, 6749. [CrossRef]
7. Heeger, D.; Garigan, M.; Eleni Tsiropoulou, E.; Plusquellic, J. Secure LoRa Firmware Update with Adaptive Data Rate Techniques. *Sensors* **2021**, *21*, 2384. [CrossRef]
8. Popoola, S.I.; Adebisi, B.; Ande, R.; Hammoudeh, M.; Atayero, A.A. Memory-Efficient Deep Learning for Botnet Attack Detection in IoT Networks. *Sensors* **2021**, *10*, 1104. [CrossRef]
9. Charilaou, C.; Lavdas, S.; Khalifeh, A.; Vassiliou, V.; Zinonos, Z. Firmware update using multiple gateways in LoRaWAN networks. *Sensors* **2021**, *21*, 6488. [CrossRef]
10. Mtetwa, N.S.; Tarwireyi, P.; Sibeko, C.N.; Abu-Mahfouz, A.; Adigun, M. Blockchain-based security model for LoRaWAN firmware updates. *Sensors* **2022**, *11*, 5. [CrossRef]
11. Witanto, E.N.; Oktian, Y.E.; Lee, S.G.; Lee, J.H. A blockchain-based OCF firmware update for IoT devices. *Appl. Sci.* **2020**, *10*, 6744. [CrossRef]
12. Khan, M.Z.; Alhazmi, O.H.; Javed, M.A.; Ghandorh, H.; Aloufi, K. Reliable Internet of Things: Challenges and future trends. *Electronics* **2021**, *10*, 2377. [CrossRef]
13. Odat, H.A.; Nsour, A.; Ganesan, S. In Firmware over the air ad-hoc network, FOTANET. In Proceedings of the 2015 IEEE International Conference on Electro/Information Technology, Dekalb, IL, USA, 21–23 May 2015; pp. 101–106.
14. Wang, Z.J.; Han, J.J.; Miao, T.P. An Efficient and Dependable FOTA-Based Upgrade Mechanism for In-Vehicle Systems. In Proceedings of the International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Atlanta, GA, USA, 14–17 July 2019; pp. 196–201.
15. Steger, M.; Karner, M.; Hillebrand, J.; Rom, W.; Armengaud, E.; Hansson, M.; Boano, C.A.; Römer, K. Applicability of IEEE 802.11 s for automotive wireless software updates. In Proceedings of the International Conference on Telecommunications, Graz, Austria, 13–15 July 2015; pp. 1–8.
16. Kandemir, C.; Soner, O.; Celik, M. Proposing a practical training assessment technique to adopt simulators into marine engineering education. *WMU J. Marit. Aff.* **2018**, *17*, 1–15. [CrossRef]
17. Laskowski, R.; Chybowski, L.; Gawdzińska, K. An engine room simulator as a tool for environmental education of marine engineers. In Proceedings of the New Contributions in Information Systems and Technologies, Azores, Portugal, 1–3 April 2015; pp. 311–322.
18. Jung, B.G.; So, M.O.; Eum, P.Y.; Paek, S.H.; Kim, C.H. Development of the Marine Engine Room Simulator. *J. Adv. Mar. Eng. Technol.* **2007**, *31*, 872–880. [CrossRef]
19. Hejlsberg, A.; Wiltamuth, S.; Golde, P. *C# Language Specification*, 6th ed.; Addison-Wesley Longman Publishing Co., Inc.: Saddle River, NJ, USA, 2003; ISBN 0321154916.
20. Thiers, J.P.; Nicolas, B.D.; Freudenberger, J. Read Reference Calibration and Tracking for Non-Volatile Flash Memories. *Electronics* **2021**, *10*, 2306. [CrossRef]
21. Zhang, L.; Wang, J. EtherNet/IP message analysis. In Proceedings of the 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet), Yichang, China, 21–23 April 2012; pp. 319–322. 6202024. [CrossRef]
22. Sha, C.; Lin, Z.Y. Design Optimization and Implementation of Bootloader in Embedded System Development. In Proceedings of the International Conference on Computer Science and Applications (CSA), Wuhan, China, 20–22 November 2015; pp. 151–156. [CrossRef]

23. da Silveira, C.M.; T de Sousa, R., Jr.; de Oliveira Albuquerque, R.; Amvame Nze, G.D.; de Oliveira Júnior, G.A.; Sandoval Orozco, A.L.; García Villalba, L.J. Methodology for forensics data reconstruction on mobile devices with Android operating system applying in-system programming and combination firmware. *Appl. Sci.* **2019**, *10*, 4231.
24. Kang, Y.B.; Chen, J.X.; Li, B. Generic Bootloader Architecture Based on Automatic Update Mechanism. In Proceedings of the 3rd International Conference on Signal and Image Processing (ICSIP), Shenzhen, China, 13–15 July 2018; pp. 586–590. [CrossRef]
25. Kwon, J.; Seok, M.G.; Park, D. Low-Power Fast Partial Firmware Update Technique of On-Chip Flash Memory for Reliable Embedded IoT Microcontroller. *Ieice Trans. Electron.* **2021**, *E104C*, 226–236. [CrossRef]
26. *ISO Standard 20922*; Message Queuing Telemetry Transport (MQTT) V3.1.1. ISO: Geneva, Switzerland, 2016.
27. Sahlmann, K.; Clemens, V.; Nowak, M.; Schnor, B. MUP: Simplifying secure over-the-air update with MQTT for constrained IoT devices. *Sensors* **2020**, *21*, 10. [CrossRef] [PubMed]
28. Kurdi, H.; Thayananthan, V. A Multi-Tier MQTT Architecture with Multiple Brokers Based on Fog Computing for Securing Industrial IoT. *Appl. Sci.* **2022**, *12*, 7173. [CrossRef]
29. Kluj, S. The modular architecture of the engine room simulator. In Proceedings of the 17th IASTED International Conference on Modeling and Simulation, Montreal, QC, Canada, 24–26 May 2006; pp. 348–353.
30. Wang, X.; Zhao, Z.; Xu, D.; Zhang, Z.; Hao, Q.; Liu, M.; Si, Y. Two-Stage Checkpoint Based Security Monitoring and Fault Recovery Architecture for Embedded Processor. *Electronics* **2020**, *9*, 1165. [CrossRef]
31. Jung, J.Y.; Lee, J.R. Throughput and Packet Loss Probability Analysis of Long Range Wide Area Network. *Appl. Sci.* **2021**, *11*, 8091. [CrossRef]
32. Zhang, Y.; Chen, G.; Du, H.; Yuan, X.; Kadoch, M.; Cheriet, M. Real-Time Remote Health Monitoring System Driven by 5G MEC-IoT. *Electronics* **2020**, *9*, 1753. [CrossRef]
33. Kim, J.H.; Kim, S.H.; Jang, J.W.; Kim, Y.S. Low Complexity List Decoding for Polar Codes with Multiple CRC Codes. *Entropy* **2017**, *19*, 183. [CrossRef]
34. Lee, J.; Kwon, T. Distributed Watchdogs Based on Blockchain for Securing Industrial Internet of Things. *Sensors* **2021**, *21*, 4393. [CrossRef] [PubMed]
35. Jang, J.; Jung, I.Y. Sustainable and practical firmware upgrade for wireless access point using password-based authentication. *Sustainability* **2016**, *8*, 876. [CrossRef]
36. Avizienis, A.; Laprie, J.C.; Randell, B. Basic concepts and taxonomy of dependable and secure computing. *IEEE Trans. Dependable Secur. Comput.* **2014**, *1*, 11–33. [CrossRef]
37. Ayeb, N.; Rutten, E.; Bolle, S.; Coupaye, T.; Douet, M. Coordinated autonomic loops for target identification, load and error-aware Device Management for the IoT. In Proceedings of the 15th Conference on Computer Science and Information Systems (FedCSIS), Sofia, Bulgaria, 6–9 September 2020; pp. 491–500.
38. Hercog, D.; Gergič, B.A. Flexible Microcontroller-Based Data Acquisition Device. *Sensors* **2014**, *14*, 9755–9775. [CrossRef] [PubMed]