Research on Remote Online Firmware Upgrade System for Embedded Devices

Xian Zhang[1,2,3,4], Yiwen Liu[1,3,4*], Taiguo Qu[1], Pengju Tang[1]

[1] School of Computer Science and Engineering, Huaihua University, China
[2] College of Computer Science and Electronic Engineering, Hunan University, China
[3] Key Laboratory of Intelligent Control Technology for Wuling-Mountain Ecological Agriculture in Hunan Province, China
[4] Key Laboratory of Wuling-Mountain Health Big Data Intelligent Processing and Application in Hunan Province Universities, China
zxian@ hnu.edu.cn, 87134537@qq.com, 1849310620@qq.com, 34063641@qq.com

Abstract

To effectively reduce the cost of firmware upgrade and maintenance of embedded devices, according to the requirement of a remote firmware update of embedded devices, this paper proposes a remote online firmware upgrade technology based on ZigBee. Combined with the requirements of remote online firmware upgrade application of tower-mounted amplifier, this paper analyzes and studies the system requirements, ZigBee module selection and design, software design and implementation (communication protocol design of host and slave program, breakpoint continuation function), fault analysis and other aspects. Designed experiments and tested the ZigBee connection success rate under the device startup state and normal operation state, the relationship between transmission firmware size and transmission success rate, system CPU resource consumption, transmission distance, transmission rate, the success rate, data reuse rate and acceleration ratio of breakpoint continuation function, and other performance indicators. The experimental results show that the remote online firmware upgrade technology based on ZigBee for embedded devices can better meet the system's performance requirements in all aspects and improve the data reuse rate and acceleration ratio in the process of data transmission. The experimental results show that this scheme has the advantages of simplicity, low cost, high reliability, and broad application prospects.

Keywords: ZigBee, Online upgrade, Breakpoint continuation, Firmware, Transmission distance

1 Introduction

With the rapid development of sensors and radio frequency technology, the Internet of things (IoT) application field involves every aspect of our daily life, including the extensive application in industry, agriculture, environment, transportation, logistics, security, and other infrastructure fields [1-2]. According to the China Internet Development Report (2021) released by the Internet Society of China, China's IoT market is worth 1.7 trillion yuan.

After the IoT devices are used, they need to be maintained or upgraded for various reasons, posing a new problem for designers. If the traditional local program update and upgrade method is used, the product needs to be recalled, increasing the enterprise's cost [3-4]. It is especially when embedded devices are installed in extremely harsh environments, such as high towers and hilltops. The traditional firmware upgrade method will bring inconvenience to maintenance. Therefore, studying remote online firmware upgrade technology for embedded devices is essential [5]. It can also be regarded as an extension of the current research branch in edge computing [6].

According to the different data transmission modes of firmware upgrade for embedded devices, the commonly used online firmware upgrade methods can be divided into the wired and wireless transmission. Standard wired transmission methods are based on serial cable and Ethernet connection. The advantages of wired transmission include stable performance, convenient use, and fast speed. The disadvantage is that a dedicated cable or network cable needs to be connected between the upgrade device and the server, which is costly and cannot be used in some harsh environments [7-8]. Commonly used wireless transmission methods include data transmission based on GPRS/CDMA/LTE, data transmission based on wireless sensor network, WIFI network data transmission, ZigBee data transmission. With the help of GPRS/CDMA/LTE to connect to the Internet, the network upgrade is convenient and reliable, but the communication module is expensive, and additional data traffic charges are required [9-10]. The upgrading of wireless sensor networks mostly depends on the Bootloader provided by the operating system. This method is convenient to maintain code and has high accuracy, but the upgrading of operation is complicated, and the operating system has high requirements on hardware [11]. The advantages of WIFI network data transmission are easy deployment and good flexibility, while the disadvantages are large power consumption and low security. Zigbee-based data transmission has the advantages of low complexity, low power consumption, and low cost, while its disadvantage is low speed [12-13].

The remote online upgrade must consider the function of file breakpoint continuation. Relevant research [14-15] has

*Corresponding Author: Yiwen Liu; E-mail: 87134537@qq.com
DOI: 10.53106/160792642022122307013

been carried out to realize file breakpoint continuation by using FTP (File Transfer Protocol) protocol and HTTP (Hyper Text Transfer Protocol) protocol. These methods are suitable for downloading files based on the Web browser or file download tool. These are not well suitable for file upgrade and transmission between users in the embedded communication system.

This paper proposed a remote online firmware upgrade method for embedded devices based on ZigBee. Taking the remote online firmware upgrade scheme of a tower-mounted amplifier that the author participated in and designed as an example, this paper introduces the design and implementation process of the remote online firmware upgrade system based on ZigBee for embedded devices.

2 System Survey

Tower mounted amplifier is a high-efficiency integrated power amplifier design and digital frequency selection adaptive technology equipment. It is mainly for the adequate amplification of base station signal, expanding the coverage of base station, reducing the base station investment, and improving quality of network signal. At the same time, the tower mounted amplifier should be able to support tower mounting, which is suitable for wide-area coverage of urban villages, suburbs, hills, roads, sea areas, and other wide-area coverage occasions.

The monitoring software system of tower-mounted amplifier includes driver program and application program. The application program is designed according to the daemon module program, host module program, and slave module program. The upper Interface software consists of Object Monitor Test (OMT) and Graphical User Interface (GUI). OMT communicates with the host module program, while GUI communicates directly with the slave module program. Figure 1 shows the overall block diagram of the monitoring software system of the tower-mounted amplifier.

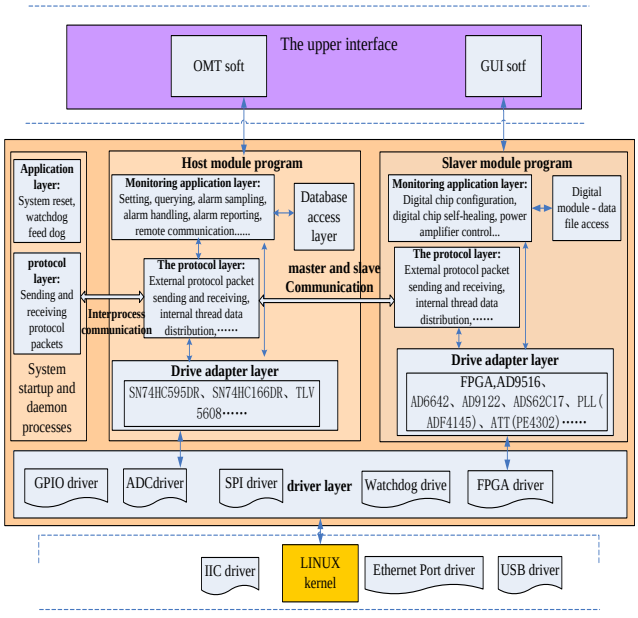

Figure 1. Block diagram of monitoring software system for tower mounted amplifier

Driver programs include GPIO driver, ADC driver, SPI driver, watchdog driver, FPGA driver, network port driver, IIC driver, USB driver, etc. The driver program compiled into the kernel includes network port drivers, IIC drivers, and USB drivers. The host module program is mainly responsible for monitoring the running state information of the tower-mounted amplifier equipment. The slave module program is responsible for the function setting and management of the FPGA of the digital slave; The host module program and the slave module program operate independently of each other, but they provide a communication interface. The communication between the master monitor program and slave monitor program adopts the mode of question and answer. Typical communication processes include query, setting, etc. The communication mode of the host and slave monitoring programs is shown in Figure 2.

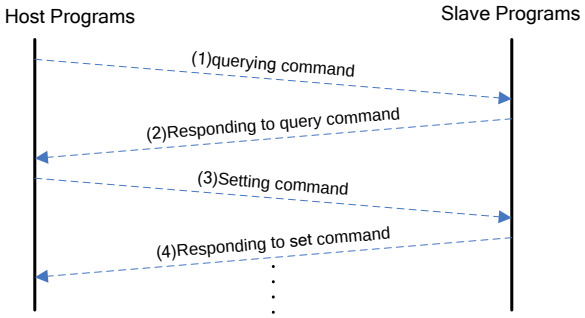

Figure 2. Communication mode between host and slave programs

The firmware upgrade of the tower-mounted amplifier mainly includes the FPGA configuration file of the slave module program, and device status query, parameter setting, and software upgrade by the host module program.

In general, tower top amplifier equipment is usually placed on the top of the mountain or tower in a relatively harsh environment. When upgrading firmware, if the wired way is adopted, people need to climb to the top of the mountain or tower for wired connection before upgrading firmware, which is highly inconvenient, unsafe, and costly. If the firmware is upgraded wirelessly, it is safer and has lower labour costs.

Comprehensively consider the advantages and disadvantages of WiFi, ZigBee, GPRS / CDMA / LTE, and other wireless transmission methods. This design decides to adopt the remote online firmware upgrade method based on ZigBee. ZigBee wireless transmission uses the ZigBee protocol. ZigBee is a wireless network protocol developed by the ZigBee Alliance (developed in 1998). The underlying layer is the media access layer and the physical layer using the IEEE 802.15.4 standard specification, and the frequency band is 2.4GHz. Its main characteristics are low speed, low power consumption, low cost, support many network nodes, support a variety of network topologies, low complexity, fast, reliability, and safety. According to their functions, devices in a ZigBee network can be classified into coordinators, routers, and end-devices. The ZigBee coordinator manages the entire ZigBee network as the initiator and maintainer of the network. Through the connection relay function of the router, the coordinator can control devices beyond their energy range [16-17].

Because there is no ZigBee module circuit in the original tower-mounted amplifier system, in designing the remote online firmware upgrade system of the tower-mounted amplifier, it is necessary to add a ZigBee module in the monitoring circuit of the tower mounted amplifier and then design the host-slave module program to realize the remote online firmware upgrade function.

3 System Design

The design of the remote online firmware upgrade system of tower-mounted amplifier mainly includes requirement description, ZigBee module selection and design, software design and Implementation (master-slave program communication protocol design [18], breakpoint continuation function), fault analysis, etc. Next, we will design from the following aspects.

3.1 Requirement Description

To facilitate the commissioning and maintenance of on-site engineers and avoid the danger and hardship of climbing towers and mountains. The project puts forward the requirement of using ZigBee to connect machines to realize remote online firmware upgrades. The GUI debugging software is connected to the ZigBee module's host-slave monitoring integrated board. And Figure 3 shows the overall structure of its communication framework.

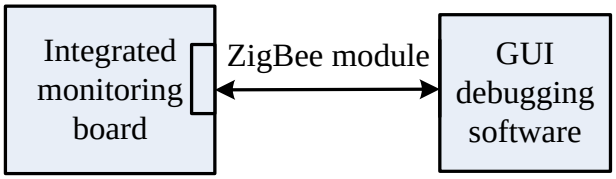

Figure 3. GUI uses ZigBee module structure diagram

To realize connection communication using the ZigBee, it needs to add the ZigBee connection communication mode to the interface of GUI debugging software. When ZigBee connection communication is selected, GUI continuously sends a connection command to try to connect until the monitoring program responds to the connection command request. GUI shows that the connection is successful. After the ZigBee model is connected successfully, you can query and set power amplifier, digital, and RADIO frequency parameters through the GUI.

Table 1 lists the specific parameters and specifications of the ZigBee-based remote online firmware upgrade system.

Table 1. System parameters

Indicator	Distance	Response time	Speed	Breakpoint continuation success rate
Parameter values	≥100m	<3s	>150Kb/S	>99%

The primary indicator of system design is that the transmission distance is greater than or equal to 100 meters, and the transmission speed is greater than or equal to 150kbps.

In the process of system firmware upgrade, equipment failure and communication failure may occur, resulting in the interruption of upgrade transmission. It is needed to support the breakpoint continuation function to avoid repeated transmission of upgrade data. The success rate of the breakpoint continuation function is required to reach more than 99%.

3.2 ZigBee Module Selection

The CC2530 chip can be directly selected as the ZigBee module's main chip according to the system indicators and parameter requirements [19]. Table 2 shows the main parameters of the CC2530 chip.

Table 2. CC2530 chip parameters

Major parameter	Parameter values	Description
Frequency range (MHz)	2400~2480	Default 2.4GHz (32MHz crystal)
Transmitting power (dBm)	4	Maximum power (2.5mW)
Communication distance (m)	250	Minimum distance
Transmission speed (bps)	250K	Average speed
Communication interface	UART I/O	Baud rate: 115200, 57600, 38400, 19200, 14400, 9600, 4800, 2400
Supply voltage (V)	2.0~3.6	/

The parameters of the CC2530 chip in the above table can meet the requirements of transmission distance and transmission speed of the system.

Figure 4 shows the schematic circuit diagram of the ZigBee module designed with the CC2530 chip.

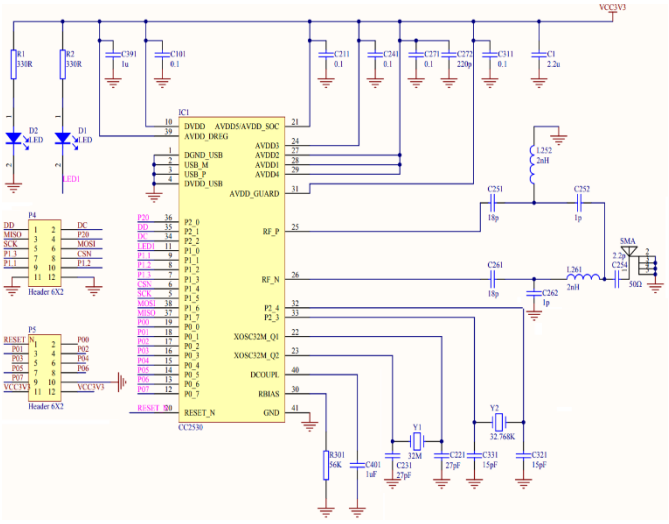

Figure 4. Schematic circuit diagram

The core circuit of the ZigBee module is mainly composed of a crystal oscillator circuit, CC2530 chip circuit, matching circuit from RF signal to antenna, and decoupling filter circuit of power supply.

At the same time, the ZigBee module is required to be designed as a pluggable interface and can have plug and play.

3.3 Software Design

The overall scheme of GUI software using the ZigBee module to realize connection communication is as follows: From the beginning, the main monitor program is responsible for monitoring the ZigBee module interface and receiving and processing commands. If the command is an OMT software connection command, the host module program controls the ZigBee module interface; if it is a GUI software connection command, the slave module program controls the ZigBee module interface. Control is returned to the main module program when the OMT packet is received from the module program, or the GUI packet is not received for 20 seconds. OMT debugging software or GUI debugging software sends connecting request commands through the ZigBee module, and the main monitoring program thread is responsible for monitoring the ZigBee module interface, receiving commands, and processing to determine whether it is OMT connecting or GUI connecting commands. The host module program processes data packets if it is an OMT software connection command. At the same time, the host module program controls the ZigBee module interface, receives and responds to the command, and tries to establish a connection with OMT software through the ZigBee module. Suppose it is a GUI software connection command. In that case, it should control the ZigBee module interface to the slave module program, which receives and processes commands and tries to establish a connection with GUI software through the ZigBee module. In addition, control is returned to the main monitor program when the OMT software command is received from the ZigBee module interface or the GUI software command is not received for 20 seconds.

Figure 5 shows the flow chart of the scheme to realize GUI software using the ZigBee module interface connection communication.

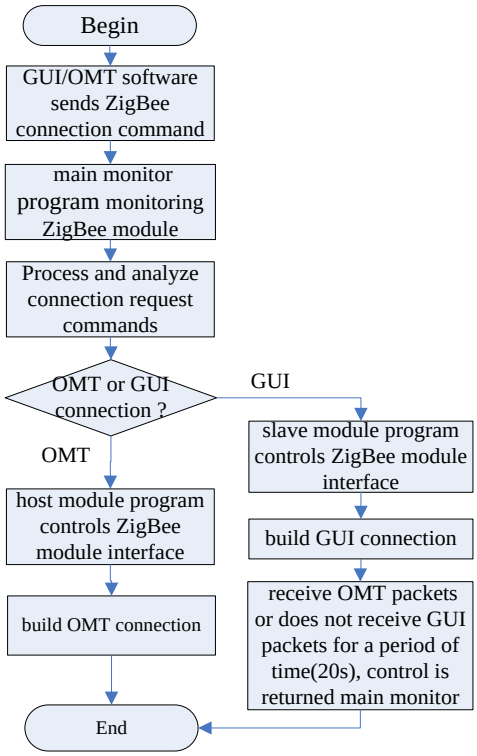

Figure 5. Flowchart of GUI software and ZigBee connection

A prerequisite for implementing the above scheme is distinguishing between command packages sent by OMT software and GUI software, which requires us to receive and judge the command packages sent by the GUI software and the OMT software. Next, we will analyze the differences between the packets sent by the OMT software and the GUI software.

To effectively distinguish whether the command package is sent by OMT or GUI software. Firstly, we analyze the protocol packet composition and relationship of the access layer, network layer, and monitoring layer when the host-slave software program connects communication.

For the monitoring software of tower-mounted amplifiers, protocol layering is carried out to facilitate the modularization of software design for the data transmitted during communication. The communication protocol of tower-mounted amplifier monitoring software is divided into four layers: media layer, monitoring control layer, network layer, and access layer [20]. The functions and uses of each layer are as follows:

Media layer: It is the actual channel of communication. This layer can provide byte-oriented packets to the access layer. This layer protocol is abbreviated as MP (medium protocol).

Monitoring control layer: It realizes the data organization oriented to monitoring functions for various monitoring functions. This layer protocol is abbreviated as CP (control protocol).

Network layer: It carries the protocol package of monitoring and control layer to realize the isolation of the monitoring and control layer from the communication channel and network structure. It can provide the monitoring instructions and data to be processed by the equipment to the monitoring control layer. It can realize the forwarding of communication packets (the forwarded packets do not need to be processed by the monitoring control layer). This layer protocol is abbreviated as NP (network protocol).

Access layer: It defines the communication transmission channel and relevant requirements to realize the information exchange with various media. The access layer carries and ensures the reliable transmission of network layer protocol data and supports a variety of access layers in this monitoring protocol. This layer protocol is abbreviated as AP (Access Protocol).

Figure 6 shows the composition and relationship of each layer protocol packet when the software program of host and slave connects and communicates.

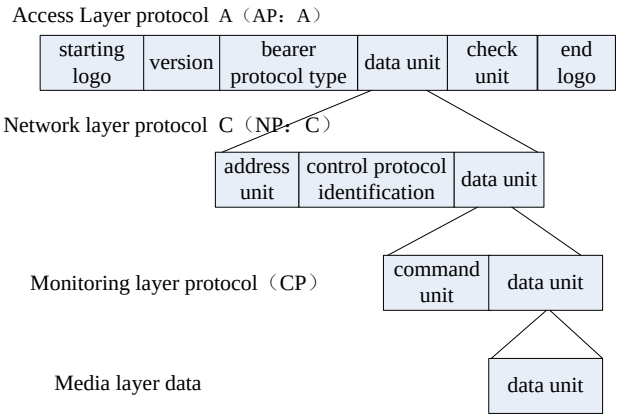

Figure 6. Packet composition and relationship of each layer protocol

The protocol layers interact in the form of data packets. The meaning of the protocol packet field of each layer is described in detail below.

Access layer protocol: a complete access layer protocol package consists of 6 parts: start flag, version, bearer protocol type, data unit, check unit, and end flag. See Table 3 for a detailed description.

Table 3. AP: a protocol package composition

Start flag	Version	Bearer protocol type	Data unit (PDU)	Check unit	End flag

The detailed description of each unit field in Table 3 is as follows:

Start flag: a flag indicating the start of a complete packet (packet); The length is 1 byte, fixed as ASCII character '~' (0x7e). The content of the end flag is the same as that of the start flag.

Version: identifies the version of the AP layer protocol; The length is 1 byte. Including AP: A, AP: B and AP: C. For example, the access layer protocol suitable for master-slave communication is AP: A and this value is 0x01.

Bearer protocol type: identifies the hosted upper layer protocol; The length is 1 byte, including NP: A and NP: C. For example, the protocol value of NP: A is 0x01, and the protocol value of NP: C is 0x03.

Data unit: it is the payload (PDU) of the access layer protocol.

Check unit: verifies the contents from "version" to "data unit" in the protocol; The CRC check value is generated according to the data packet (calculated from "version" to the last byte of "data unit"). The verification unit uses CRC verification to generate 16 bit $x^{16} + X^{12} + X^5 + 1$ (0x11021) with polynomial recommended by CCITT (Consultative Committee for International Telephone and Telegraph). The sender generates a 2-byte CRC check (the low byte is in the front and the high byte is in the back during transmission); Similarly, after receiving the complete packet, the receiver generates a new CRC check value. If the calculated CRC value is equal to the received check value, the data packet is valid. Otherwise, it is considered that an error has occurred in the transmission process of the packet. The receiver will neither process nor respond to the command with a verification error.

End flag: a flag indicating the end of a complete packet.

Network layer protocol: interact in the form of data packets. A complete network layer protocol packet comprises address unit, control protocol identification and data unit.

The format and content of the network layer protocol are shown in Table 4.

Table 4. NP: C protocol package composition

Address unit	Control protocol identification (CPID)	Data unit (PDU)

The detailed description of each unit field in Table 4 is as follows:

Address unit: the composition format of address unit is shown in Table 5.

Control protocol identification: identifies the type of upper layer protocol (i.e. CP, monitoring control protocol).

Data unit: it is the payload (PDU) of network layer protocol. The data format of network layer protocol is represented by hexadecimal number.

The composition information of the address unit is shown in Table 5.

Table 5. Address unit composition of NP: C protocol

Number	Content	Ctart position	Length
1	module number	1	1byte
2	module type	2	1byte

Module number: the unique identification of the communication address of the hardware module in a tower top amplifier.

The description of module number field is shown in Table 6.

Table 6. Module number field

Module communication address	Description
0x01～0x0F	Module No. 1 ~ 15
0x10～0x1F	Module No. 16 ~ 31

Network layer protocol is mainly used for communication between host and slave. The communication process between host and slave programs is described below. The process of OMT communicating with the slave through the host: OMT needs to use NP: A protocol to package data, in which NP: A data unit (PDU) is composed of MAP protocol package, which is composed of command unit and data unit. Here, the command ID in the command unit is "0x88" (i.e. forwarding command ID), and the data unit is the data to be forwarded; since transparent forwarding is required, the data unit must be NP: C protocol package. After the monitoring host receives the data packet sent by OMT, analyze and remove AP: A to get NP: A, then remove NP: A to get the map protocol packet, and then get the command ID "0x88" and data unit (at this time, the data unit is NP: C) in the map protocol packet. The monitoring motherboard takes out the original packet of data unit (NP: C) and then takes out the address information, data information and protocol type information to the protocol stack; according to this information, the protocol stack is packaged into NP: C again, and then packaged into AP: A based on NP: C, and then sent to the slave. It completes the process of OMT sending data to the slave. After the slave machine receives the data, it returns to the monitoring host. The monitoring host completes map packaging, NP: A packaging, AP: A packaging, and then returns to OMT. It completes the process of OMT controlling the slave.

The process of direct communication between OMT and slave: OMT adopts NP: C protocol. When the host communicates with the slave, the protocol layer packet is converted as follows:

OMT−>AP:A+NP:C−>slave ... ... slave−>AP:A+NP:C−>OMT

Monitoring control layer protocol: interact in the form of data packets. A complete monitoring and control layer protocol packet is composed of command unit and data unit. The details are shown in Table 7.

Table 7. Composition of CP protocol package

Command unit	Data unit

The details of each unit field in Table 7 are as follows:

Command unit: the composition format of the command unit is shown in Table 8.

Data unit: it is the payload (PDU) of the monitoring control layer protocol. This part can be empty (i.e., 0 bytes).

The composition of the command unit is shown in Table 8:

Table 8. Command unit composition of CP protocol

Number	Content	Start position	Length
1	Command ID	1	1byte
2	Response flag	2	1byte

The command fields in Table 8 are described as follows:

Command ID: the unique ID of the command. Details are shown in Table 9.

Table 9. Command identification of amplifier communication

Command	Meaning	Remarks
0x88	Forward slave communication packet	OMT↔monitoring host
normal mode, internal mode, test mode:		
0xB1	Query	OMT↔monitoring slave, monitoring host ↔monitoring slave
0xB2	Setup	OMT↔monitoring slave, monitoring host ↔monitoring slave

The details of response flag coding are shown in Table 10.

Table 10. Response flag code definition

Code	Meaning	Remarks
0x00	Success	/
0x01	The command is executed conditionally	Indicated by bit15 ~ bit13 of the monitoring object label
0x02	Command number error	Invalid command
0x03	Length error	The actual length received does not match the described in the packet.
0x04	Reserve	Reserve
0x05~0xFD	Reserve	Reserve
0xFE	Other errors	/
0xFF	Command	Represents the command package issued, not the response package of the command

Since the protocol data packet sent by the OMT software program uses NP: A protocol in the network layer, its identification number is 0x01. The data packet sent by the GUI software program uses NP: C protocol in the network layer, and its identification number is 0x03. Therefore, we can distinguish between the OMT software package and the GUI software package according to the protocol identification number used in the network layer.

After the above analysis, we can implement the scheme adopted in this paper without changing the original GUI software architecture. In the beginning, the main monitor program controls the ZigBee module interface. If GUI software packets are received, the control of the ZigBee module is handed over to the slave software program. In controlling the ZigBee module interface, if the slave software program receives the OMT software program packet or does not receive the GUI software program packet for 20 seconds, the host/slave program communication returns the control the ZigBee module to the main monitor program. Through this control process to achieve GUI software and ZigBee module connection and data communication, so as to achieve the function of using ZigBee to complete the remote online firmware upgrade.

3.4 Breakpoint Continuation

In data transmission, the file transmission may be interrupted due to the interruption of network connection or power failure. To deal with these emergencies, developing a support breakpoint continuous download function can avoid the waste of resources, speed up the download speed and improve the user experience. Therefore, in the remote online firmware upgrade system requirements for the embedded devices, it is proposed to support the breakpoint continuation function.

The essential requirement of breakpoint continuation is to record the location where the last download was disconnected and download from the location where the last download was completed when starting the next time. Therefore, the critical points for the implementation of breakpoint continuation mainly include the following two points:

One is to save the local download information when the transmission is interrupted. A temporary file needs to be established to save the local download information. The download information includes the number of downloaded bytes, file pointer, and total file size. When the transmission is interrupted, it will store the download information in the temporary file, and the file pointer will point to the end of the downloaded file.

The second is to check the local download file information when resuming transmission at the breakpoint. First, judge whether there is a file to be downloaded locally. If there is a file, read the information of the downloaded file, and move the pointer to the next byte of the downloaded file to realize the continuous transmission of the file at the breakpoint. The range field needs to be added to the request header to indicate where the client wants to continue downloading in the specific implementation. Range: bytes = XXX -, which means that you need to start downloading from XXX bytes when downloading the requested file.

The flow chart of realizing the breakpoint continuation function is shown in Figure 7.

During file upgrade transmission, the Range field in the packet request records the breakpoint serial number. When upgrading firmware transmission, directly request the transmission file from the breakpoint, improving the data reuse rate.

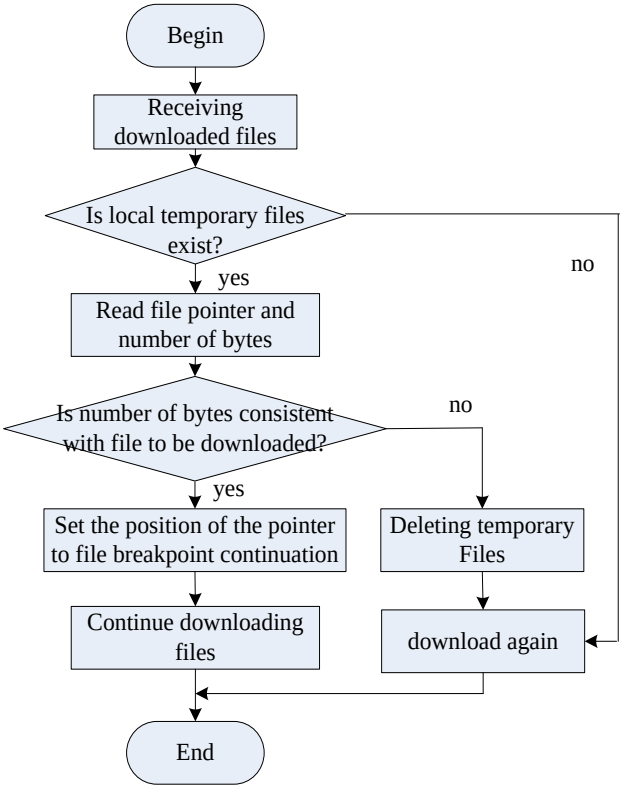

Figure 7. Flow chart of breakpoint continuation

3.5 Fault Analysis

Fault tree analysis (FTA) is a top-down deductive failure analysis method, which uses Boolean logic to combine low-order events to analyze the unwanted states in the system. FTA is mainly used in the fields of safety engineering and reliability engineering to understand the causes of system failure [21-22]. A fault tree diagram often represents the FAT method to complete system fault analysis.

To effectively realize GUI software program using ZigBee module connection and data transmission to achieve remote online firmware upgrade function, it is necessary to analyze the communication failure between GUI software program and ZigBee module. The following uses the fault tree to analyze communication failures when GUI software programs are connected to ZigBee modules. Figure 8 shows the communication fault tree between the connection of the GUI software program and the ZigBee module.

GUI software connection and communication faults using ZigBee module mainly include ZigBee module faults, host monitor program monitoring ZigBee module faults, communication faults between host and slave module programs, and slave module receiving and sending packets faults. For the failure of communication between host and slave module programs and receiving and sending data packets, we can reconnect the communication by resending the data packets after the communication failure.

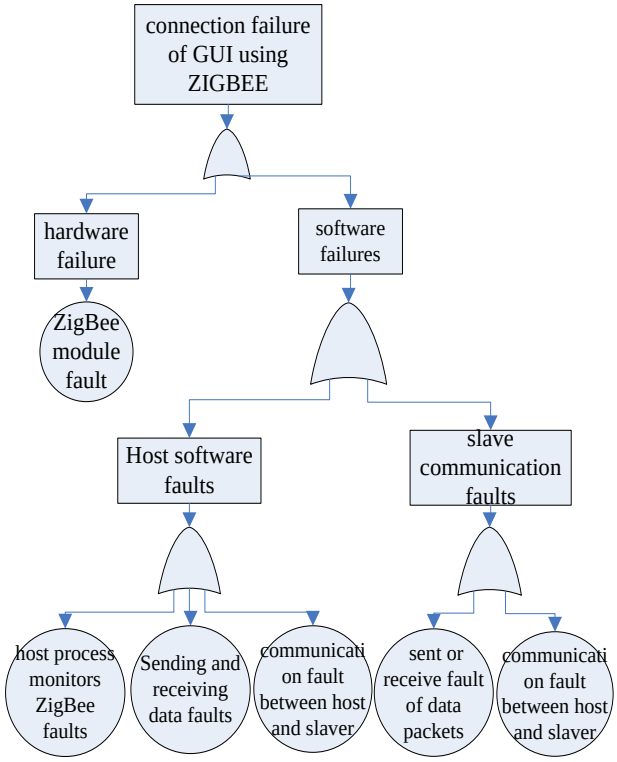

Figure 8. Communication fault tree for ZigBee and GUI connections

4 Experimental Results and Analysis

To test the remote online firmware upgrade scheme for the embedded device based on ZigBee proposed in this paper, we designed experiments to test the accuracy and timeliness of the system [23]. The ZigBee-based remote online upgrade technology of embedded devices in this scheme is integrated into the monitoring platform of the tower-mounted amplifier. The specific monitoring system framework is shown in Figure 1. First, we tested the success rate of the ZigBee module connecting to the monitoring host. Test ZigBee module connection 100 times at each startup to test whether it can be correctly connected successfully; Then test ZigBee module connection 100 times during the normal operation to test whether it can be successfully connected correctly. Here, ZigBee module connection success requires a response time within 3s. Table 11 shows the test results.

Table 11. ZigBee connection test results

State	Connection times	Success connection times
Startup state	100	100
Normal state	100	100

The test results show that the ZigBee module connection of this scheme can be successfully completed within the specified time in the state of equipment startup and normal operation, meeting the design requirements.

An important index of remote online firmware upgrade systems for embedded devices based on ZigBee is the success rate of firmware upgrade transmission. Therefore, we designed an experiment to test the transmission success rate of remote online firmware upgrades based on ZigBee. Test the

transmission success rate of firmware from small to large (1MB-300MB), and test the transmission success rate of each type of firmware size 50 times. Figure 9 shows the test results of firmware size and transmission success rate based on ZigBee transmission.

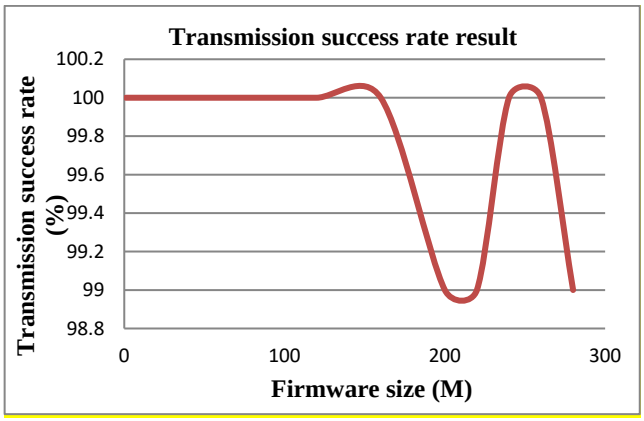

Figure 9. Test results of firmware size and transfer success rate

It can be seen from the test results that the transmission success rate is higher, and the stability is better when the firmware size is smaller than when the firmware size is larger. However, from the test results of the transmission success rate, the transmission success rate is more than 99%, which meets the design requirements of the remote online upgrade system.

Because the ZigBee module is integrated into monitoring software for tower-mounted amplifiers or other embedded devices, the consumption of system CPU resources is also an important index to evaluate the remote online firmware upgrade scheme based on ZigBee. In the actual test, the consumption degree of system CPU resource is tested respectively when the monitoring software integrates the ZigBee module's remote online upgrade method and when the ZigBee module's remote online upgrade method is not integrated. Figure 10 shows the consumption of system CPU resources.

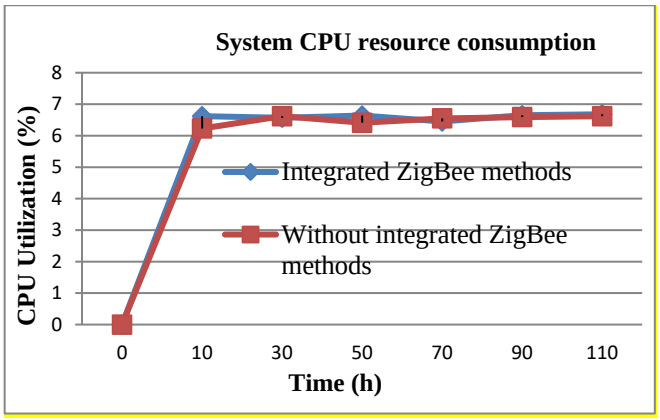

Figure 10. CPU usage of the system

It can be seen from the actual test results that the system CPU resource consumption is very close when the ZigBee module is integrated and the ZigBee module software is not integrated. Therefore, the CPU resource consumption of the system based on ZigBee module remote online upgrade technology can meet the system requirements of the tower-mounted amplifier or other embedded devices.

Transmission distance is another important indicator of a remote online firmware upgrade system for embedded devices based on ZigBee. Therefore, we designed experiments to test the relationship between transmission distance and transmission speed. In the experiment, we set the baud rate of the serial connection of the ZigBee module to 115200. The test distance is set from near to far (1m-250m) to test the transmission speed during firmware upgrade. The distance is an open direct viewing distance without obstacles. The test result is five times for each distance, and the average transmission speed is taken as the transmission speed. Figure 11 shows the relationship between ZigBee-based transmission distance and transmission rate.

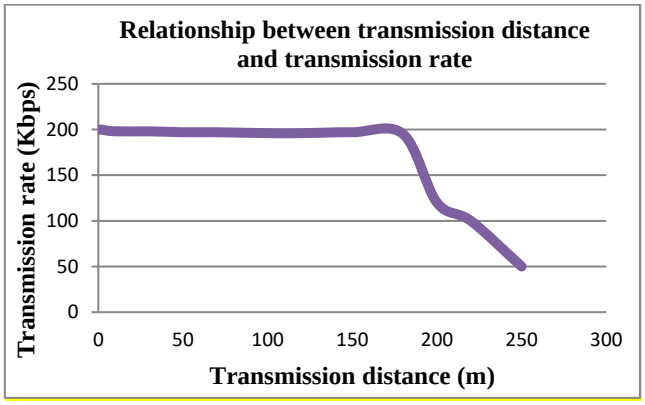

Figure 11. Relationship between transmission distance and transmission rate

From the test results, we can see that the transmission speed will decrease with transmission distance. The design requirement of our system is that the transmission distance is greater than 100 meters, and the transmission rate is greater than 150Kbps. As can be seen from the figure, when the open distance reaches about 200 meters, the transmission speed is basically maintained at 200 KBPS. When the distance is larger than 200 meters and becomes larger and larger, the transmission speed decreases sharply. Therefore, this design scheme can meet the requirements of remote online firmware upgrades.

Breakpoint continuation function is an important module of remote online firmware upgrade system for embedded devices. We designed an experiment to test the success rate of the breakpoint continuation function by suddenly disconnecting the connection in file transmission, then reconnecting and transmitting the file. A total of 100 tests were conducted. The disconnection time point was file transfer to 10%, 30%, 50%, 80% and 90% respectively. Test its breakpoint continuation success rate and save time. The test results of breakpoint continuation success rate are shown in Table 12.

Table 12. Success rate of breakpoint continuation

Transmission progress	10%	30%	50%	80%	90%
Number of tests	20	20	20	20	20
Continuation success rate	100%	100%	100%	100%	100%

From the test results, it can be seen that the success rate of the breakpoint continuation function reaches 100%, which meets the system design requirements and is more than 99%.

At the same time, we test the acceleration ratio of data transmission of the breakpoint continuation function. The time point of disconnection is file transmission to 10%, 30%, 50%, 80% and 90% respectively. Test the average acceleration ratio of 1 disconnection, three disconnections, and both average cases. The acceleration ratio for data transmission is shown in Figure 12.

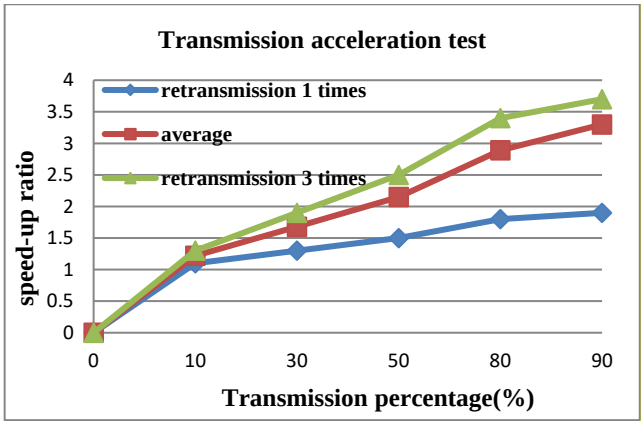

Figure 12. Data acceleration ratio of breakpoint continuation function

It can be seen from the test results that when the number of continuous transmissions at the disconnection point is more, and the percentage of data transmitted at each disconnection is higher, the acceleration ratio of data transmission is higher. The higher the data reuse rate, it will save more time.

The breakpoint continuation function can improve the speed-up ratio of data transmission because when we design breakpoint continuation, the program will detect and judge whether there is a file to download locally. If there is a file, read the information of the downloaded file, move the pointer to the next byte of the downloaded file, and then realize the breakpoint continuation of the file from the next position, which can improve the speed-up ratio of data transmission. For example, when a file is disconnected after being transferred to 50%, and when it is connected again for retransmission, retransmission starts from 50% of the file, which achieves an acceleration ratio of nearly 50%.

5 Conclusions

Based on the comprehensive consideration of resources, demand, cost performance, effectiveness, and other factors, this paper designed a remote online firmware upgrade scheme for the embedded device based on ZigBee. The system is based on the tower-mounted amplifier equipment designed and participated in by the author as a platform. The design experiments respectively test the ZigBee connection success rate of the equipment under the state of starting up and normal operation, the relationship between the firmware size of transmission and the transmission success rate, the CPU resource consumption of the system, the relationship between the transmission distance and transmission rate, the success rate of breakpoint continuation, data reuse rate and acceleration ratio of breakpoint continuation, etc. The experimental results in our design show that the remote online firmware upgrade technology for the embedded device based on ZigBee can better meet the performance indicator requirements of various aspects of the system. The design of a remote online firmware upgrade method for the embedded device based on ZigBee allows the device to update its internal code remotely online without affecting the operation of the original code, which effectively improves the firmware upgrade efficiency and reduces the upgrade and maintenance cost of the device. Therefore, it has a broad application prospect.

Acknowledgement

This work was supported by the Open Fund Project of the Key Laboratory of Intelligent Control Technology for Wuling-Mountain Ecological Agriculture in Hunan Province (ZNKZD2021-08); This work was supported by Huaihua Science and Technology Innovation Plan Project (2021R3130); This work was supported by Key Project of Hunan Provincial Department of Education (19A394); This work was supported by Xinhuasan Fund Project of Hunan Internet of Things Society (201808); This work was supported by the National Natural Science Foundation of China (62172182); This work was supported by Hunan Province Degree and Postgraduate Teaching Reform Project (2021JGYB212); This work was supported by Teaching Research Project of Primary Computer Education of China Association for Basic Computer Education in Colleges and Universities (2020-AFCEC-047); This work was partly supported by the Huaihua University Double First-Class initiative Applied Characteristic Discipline of Control Science and Engineering.

References

[1] C. X. Fan, L. N. Cao, *Principles of Communications*, National Defense Industry Press, 6th edition, 2012.

[2] K. M. Rao, Design of Wireless ECG Signal Monitoring System Based on GPRS and Zigbee, *Computer Measurement & Control*, Vol. 29, No. 2, pp. 20-24, February, 2021.

[3] P. Li, Design and Implementation of a Cardiovascular Function Test and Diagnosis Platform Based on the Internet of Things and Zigbee, *Computer Measurement & Control*, Vol. 29, No. 3, pp. 72-76, March, 2021.

[4] Y. P. Zhou, W. J. Ma, An Improved ZigBee Routing Algorithm and Its Application in Wireless Network of Greenhouse Monitoring, *Computer Measurement & Control*, Vol. 28, No. 10, pp. 91-95, October, 2020.

[5] X. Chen, The Intelligent Street Light Control System for Preventing Heavy Fog of Expressway Based on ZigBee, *Wireless Personal Communications*, Vol. 121, No. 1, pp. 353-359, November, 2021.

[6] S. Vimal, Y. H. Robinson, S. Kadry, H. V. Long, Y. Y. Nam, IoT Based Smart Health Monitoring with CNN Using Edge Computing, *Journal of Internet Technology*, Vo1. 22, No. 1, pp. 173-184, January, 2021.

[7] P. P. Saraswala, S. B. Patel, J. K. Bhalani, Performance metric analysis of transmission range in the ZigBee network using various soft computing techniques and the hardware implementation of ZigBee network on

ARM-based controller, *Wireless Networks*, Vol. 27, No. 3, pp. 2251-2270, April, 2021.
[8] J. Yan, J. Yang, F. Zhu, Z. Teng, Green city and government ecological environment management based on ZigBee technology, *Environmental Technology & Innovation*, Vol. 23, Article No. 101711, August, 2021.
[9] Z. R. Wang, Greenhouse data acquisition system based on ZigBee wireless sensor network to promote the development of agricultural economy, *Environmental Technology & Innovation*, Vol. 24, Article No. 101689, November, 2021.
[10] H. Sun, J. Q. Hu, Study on the Wireless Sensor Network Monitoring System Based on ZigBee Technology and Optimization Algorithm Simulation, *2021 International Wireless Communications and Mobile Computing Conference*, Harbin, China, 2021, pp. 905-909.
[11] K. Huang, B. C. Sun, Design of Port Communication Signal Management System Based on ZigBee, *Journal of Coastal Research*, Vol. 103, No. sp1, pp. 735-738, June, 2020.
[12] Y. Hu, *ZigBee wireless communication technology application development*, Beijing: Publishing House of Electronics Industry, 2020.
[13] Z. C. Chi, Y. Li, H. Y. Sun, Z. C. Huang, T. Zhu, Simultaneous Bi-Directional Communications and Data Forwarding Using a Single ZigBee Data Stream, *IEEE/ACM Transactions on Networking*, Vol. 29, No. 2, pp. 821-833, April, 2021.
[14] Z. J. Wang, W. Wang, Z. G. Zhao, Design and realization of FTP based file transfer component, *Journal of Shenyang Normal University (Natural Science)*, Vol. 30, No. 3, pp. 375-377, 2012.
[15] L. Mostarda, A. Navarra, F. Nobili, Fast File Transfers from IoT Devices by Using Multiple Interfaces, *Sensors*, Vol. 21, No. 1, pp. 1-24, January, 2021.
[16] H. A. Xie, B. Yang, Z. G. Ren, K. B. Mu, X. Q. Zhu, B. Y. Li, The information security transmission method for intelligent examination based on ZigBee communication, *International Journal of Information and Communication Technology*, Vol. 19, No. 3, pp. 258-274, 2021.
[17] J. Yuan, Y. He, Research on intelligent home design of internet of things based on ZigBee, *International Journal of Communication Networks and Distributed Systems*, Vol. 26, No. 3, pp. 272-286, 2021.
[18] S. J. Yang, T. C. Wei, Design Issues for Communication Protocols Conversion Scheme of IoT Devices, *Journal of Internet Technology*, Vol. 22, No. 3, pp. 657-667, May, 2021.
[19] Ompal, V. M. Mishra, A. Kumar, Zigbee Internode Communication and FPGA Synthesis Using Mesh, Star and Cluster Tree Topological Chip, *Wireless Personal Communications*, Vol. 119, No. 2, pp. 1321-1339, July, 2021.
[20] X. Zhang, P. J. Tang, D. Yin, T. G. Qu, Y. W. Liu, The Network Model of Internet Access Intranet Based on Embedded Platform, *Journal of Internet Technology*, Vol. 23, No. 2, pp. 201-208, March, 2022.
[21] S. H. Aldaajeh, S. Harous, S. Alrabaee, Fault-Detection Tactics for Optimized Embedded Systems Efficiency, *IEEE Access*, Vol. 9, No. 6, pp. 91328-91340, June, 2021.
[22] C. Guo, S. Ci, G. Y. L. Zhou, Y. Yang, A Survey of Energy Consumption Measurement in Embedded Systems, *IEEE Access*, Vol. 9, pp. 60516-60530, April, 2021.
[23] B. K. Dash, J. Peng, Performance Study of Zigbee Networks in an Apartment-Based Indoor Environment, *Proceedings of Sixth International Congress on Information and Communication Technology*, London, England, 2021, pp. 473-482.

Biographies

Xian Zhang received a Master's degree in software engineering from the College of Computer Science and Electronic Engineering, Hunan University, China, in 2011. From 2013 to 2021, he has been a faculty with the School of Computer Science and Engineering at Huaihua University, China. He is currently an associate professor in computers. His research interests include embedded systems, network management and network measurement, data processing and analysis, data mining, knowledge graph, Graph Processing Accelerators.

Yiwen Liu graduated from Xiamen University, for the degree of Master. His research interests include program language analysis, software test, digital image processing, etc.

Taiguo Qu graduated from the College of Computer Science and Electronic Engineering, Central South University, for the degree of Ph. D. His research interests include machine learning and bioinformatics.

Pengju Tang graduated from Guizhou University, for the degree of Master. His research interests include network management, network measurement.