A Multi-interface Data Acquisition Gateway Based on 6LoWPAN for Multi-sensor Situation

Lun Li
School of Control Science and Engineering
Dalian University of Technology
Dalian, China
248147124@qq.com

Guangwen Wang
School of Control Science and Engineering
Dalian University of Technology
Dalian, China
294914666@qq.com

Liang Zhao*
School of Control Science and Engineering
Dalian University of Technology
Dalian, China
zliang@dlut.edu.cn

***Abstract*—The IPv6 over Low Power Wireless Personal Area Network (6LoWPAN) has been used in many areas. How to reduce the number of 6LoWPAN nodes in multi-sensor situation needs to be studied. This paper describes the design and implementation of a multi-interface data acquisition gateway based on 6LoWPAN. The gateway has 6 kinds of data acquisition interfaces, including RS485, CAN, LoRa, ZigBee, 4G and GPRS. Modbus is the protocol of these interfaces' application layer. The gateway is designed with a modular approach, which is easy to be extended or tailored for other applications. Since the acquisition task quantity of the gateway changes with the configuration, and has the characteristics of multi-task synchronization and real-time response, the software design of the gateway is based on the FreeRTOS real-time system to meet the above requirements. In the 6LoWPAN network, the acquisition gateway acts as a 6LoWPAN node. After configuring the gateway through serial port, the gateway first reads sensor data from the interface and then sends the data to the 6LoWPAN module. The test results show that the gateway can read the sensor data correctly according to the configuration requirements. The purpose of using the multi-interface and bus protocol is to reduce the number of 6LoWPAN nodes required in the multi-sensor situation, thus reducing the routing complexity of the 6LoWPAN network, system cost and power consumption.**

Keywords*—*6LoWPAN, multi-interface, Modbus, FreeRTOS

I. INTRODUCTION

6LoWPAN is a low power wireless personal area network protocol based on IPv6, and its physical level standard is IEEE 802.15.4. It has the features such as low power and low rate, and this technology makes it possible for the resources limited devices to support Internet Protocol (IP). In recent years, it has attracted the attention of many researchers. 6LoWPAN has been used in many areas, such as construction, electric power, medical services, agriculture and fire monitoring. For example, 6LoWPAN was used to establish a wireless communication network of various operators and sensors in the building environment, which helps to reduce the cost and deployment complexity of smart building monitoring and controlling system [1]. GUO et al. [2] designed a remote meter reading module based on 6LoWPAN, and compared with other systems, 6LoWPAN provides better network stability and lower power consumption. Singh et al. [3] proposed using 6LoWPAN to collect sensor data with integrated temperature, electrocardiogram and pulse parameters to build an Internet of Things (IoT) based medical service framework. Kumar et al. [4] used 6LoWPAN to construct wireless sensor network to create an autonomous, self-regulating system. In literature [5], researchers designed a fire monitoring system based on multi-gas and thermal imaging, and the sensors are connected wirelessly over 6LoWPAN. These researches have made great contributions to the 6LoWPAN popularized. However, there are still some problems need to be considered. One of these problems is how to reduce the number of 6LoWPAN nodes in the multi-sensor situation. The multi-sensor situation includes two aspects, multiple sensors of the same type and multiple different types of sensors. Generating one 6LoWPAN node for each sensor in the multi-sensor situation will result in an increase in the number of 6LoWPAN nodes and the routing complexity of 6LoWPAN network will increases, and the system cost and power consumption too. This problem has attracted the attention of some researchers. Macayana et al. [6] used the combination of 6LoWPAN and CAN to build hydroponics sensor network to reduce the number of 6LoWPAN nodes in the system. Luo [7] studied the smart campus data collection based on 6LoWPAN, and the 6LoWPAN node had several types of sensor data interfaces. Their researches were helpful to reduce the amount of 6LoWPAN nodes in multi-sensor situation. However, literature [6] only considered the interface of CAN. Although there are many interfaces in [7], most interfaces were Transistor-Transistor Logic (TTL), and only RS485 supported individual devices. There is still some work to do to solve this problem, and that is the purpose of this paper.

This paper designs a multi-interface data acquisition gateway based on 6LoWPAN, covering six interfaces, including RS485, CAN, LoRa, ZigBee, 4G and GPRS. RS485 and CAN are field bus communication technologies widely used in industry. LoRa is a wireless communication technology, which is based on spread spectrum modulation technology and enables wireless communication to transmit long distance at low rate, which only includes the physical layer. The physical layer protocol of ZigBee is IEEE 802.15.4, and the ZigBee standard defines the network layer and application layer. It supports star connection, tree connection and mesh connection and has been applied in many fields. In addition, considering that some applications require the gateway to collect remote sensor data, this paper selects 4G and GPRS for remote data acquisition. The physical layer of the above six kinds of

978-1-6654-4110-0/21/$31.00 ©2021 IEEE
DOI 10.1109/WCCCT52091.2021.00019

communication varies. In this paper, the actual application layer protocol used in these six kinds of communication is Modbus, which is due to the fact that Modbus protocol can support multiple devices and its simplicity.

II. Hardware Design

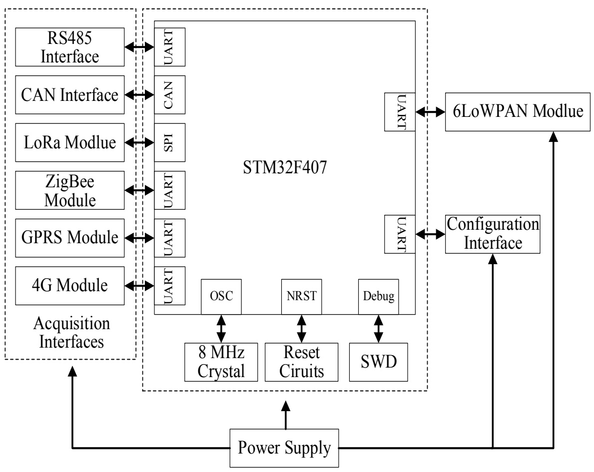

Fig. 1. Hardware architecture.

The hardware architecture of the gateway is shown in the Fig. 1, which consists of five parts: power supply, microcontroller, acquisition interfaces, configuration interface and 6LoWPAN network communication. The power supply adjusts the input DC voltage to provide a suitable and stable voltage for the components and communication modules in the system. The microcontroller is STM32F407VET6, whose kernel is ARM-CortexM4. It has sufficient storage space and contains six independent UART resources, which ensures that each module or chip of UART communication can take up one UART resource alone and can be directly connected to the UART interface of microcontroller without adding additional port multiplexers. The data acquisition interfaces are divided into fieldbus, short-range wireless and remote wireless. The fieldbus data acquisition interfaces include RS485 and CAN. The RS485 interface adopts MAX13487EESA to realize the conversion of RS485 electrical signal and TTL signal. The microcontroller has its own CAN controller, and CAN communication is available when adding a CAN driver chip, and TJA1050 is selected here. Short-range wireless data acquisition interfaces include LoRa and ZigBee. Considering the requirements of these modules for UART of microcontroller, the LoRa module that communicates with the microcontroller in SPI mode are selected, which means the LoRa module do not take up UART. ZigBee module communicates with the microcontroller in the way of UART, and ZigBee protocol stack is realized by the module, and the working mode of the module can be modified through UART interface.

In this gateway, the UART of ZigBee module can be connected to the configuration interface through the selection of jumper cap, and the ZigBee module can be configured by the computer. Remote wireless data acquisition interfaces include 4G and GPRS, and their communication with the microcontroller are UART. As well as the ZigBee module, the working mode of 4G and GPRS module can be modified through the UART interface. The configuration interface provides configuration support for the gateway. It connects to the UART of the microcontroller and converts UART communication to USB communication through USB to TTL chip (CH340G). In this way, the gateway can be connected to a computer via a USB cable, and the gateway can be configured by using software on the computer. The 6LoWPAN module is responsible for the wireless communication of 6LoWPAN. It communicates with the microcontroller through UART. The microcontroller sends the collected data to the 6LoWPAN module, through which the devices in the 6LoWPAN network can read the data collected by the gateway. The gateway is designed according to the idea of modularization, which is convenient for cutting or expanding in practical application.

III. Software Design

A. Software Architecture

In order to improve the response speed of human-computer interaction at the time of the gateway configuration, and improve the ability of handling gateways with different amount of acquisition tasks, the software is designed based on FreeRTOS rather than bare metal. Six different collection tasks are designed according to the type of communication interface, and the other tasks are the configuration task and the 6LoWPAN forwarding task. These acquisition tasks have the same priority, and under the condition of task readiness, these tasks can get the same execution time. The priority of configuration task is higher than that of acquisition task. When the configuration task is ready, it preempts the currently executing acquisition task, which ensures the real-time response of human-computer interaction. Since the data is collected first and then forwarded, the priority of 6LoWPAN network forwarding task in the software is lower than that of the acquisition task.

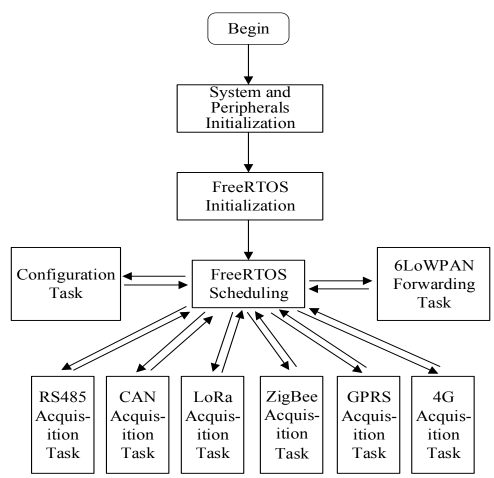

Fig. 2. Software architecture.

Software architecture is shown in Fig. 2. The first is the system and peripherals initialization, which includes initialization of clock, interrupt, UART, CAN, SPI, and so on.

The next is FreeRTOS initialization, which calls FreeRTOS API function to create six acquisition tasks, configuration task and 6LoWPAN network forwarding task, and create two event flag groups for synchronizing tasks and managing shared resources respectively. Finally, FreeRTOS scheduling is started, and each task executes under the FreeRTOS kernel scheduling.

B. *Configuration Task*

The serial port configuration task completes the functions of adding, deleting, changing and checking the configuration information table. The configuration information table contains the interface's physical layer, Modbus application layer information and acquisition cycle. The configuration information at the physical layer is for RS485, CAN and LoRa interfaces, while the configuration information at the physical layer of ZigBee, 4G and GPRS interfaces is stored in the modules' own memory, and their configuration is realized by the computer. The physical configuration information of RS485, CAN and LoRa is stored in the FLASH of microcontroller, which is set by configuration interface. The RS485 interface configuration information includes communication baud rate, data bits, stop bits, parity. The CAN interface configuration information includes communication speed and frame ID. The LoRa interface configuration information includes working frequency and the communication rate.

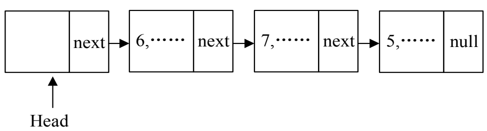

Fig. 3. Example of device Modbus information linked list on RS485 interface.

The Modbus application layer information includes device address, start register address, register length and other information. It is for devices, and each device has a configuration information table corresponding to it. For devices under the same interface, their configuration information table is stored in a linear linked list, which makes the operation of adding and deleting devices very convenient. For example, if three devices are created on the RS485 interface with the addresses of 5, 7, and 6 respectively, the list as shown in Fig. 3 will be generated. Each interface in the software has a linked list of configuration information corresponding to it, and all the linked pointer are stored in an array for quick query.

The data acquisition cycle is specific to the interface, that is, all devices on an interface in the gateway have the same data acquisition cycle. The collection cycle is configured by the serial port, and each interface has a collection cycle, which is stored in an array. When a message is received from the configuration serial port, the configuration task parses the message. During parsing, if an illegal field is found, the parsing operation is terminated and an error report is printed on the serial port. After parsing, modify the configuration information table and save it to the FLASH of microcontroller, and print the configuration results in the serial port.

C. *Acquisition Task*

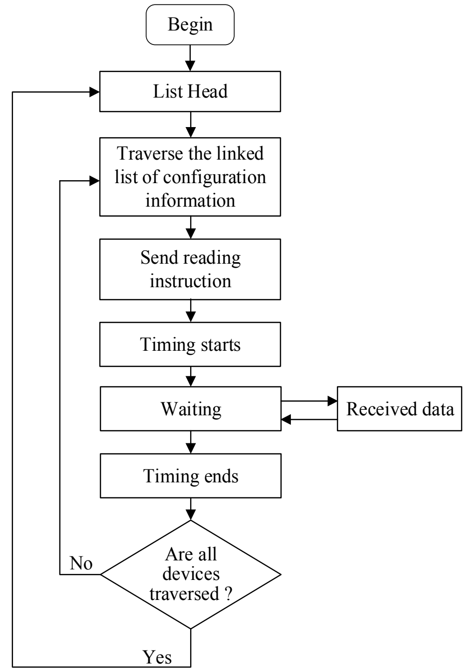

Fig. 4. The acquisition task workflow on the RS485 interface.

The acquisition task sends the reading messages on the communication interface according to the configuration table, and saves the received data. The work flow of the six acquisition tasks is the same, and the acquisition task of RS485 interface is taken as an example to illustrate here. The workflow of the acquisition task of RS485 interface is shown in Fig. 4. The acquisition task traverses the Modbus configuration information table of RS485 interface. First, the read data instruction is sent out according to the configuration table. Then, it sets a timer and waits for the response. If the response is received during the timing period, the received data is saved and then waited until the end of the timing before continuing to read the next device on the linked list. If no data is received during the timer period, the next device on the list is read directly after the timer ends. In the process of RS485 acquisition task waiting for the end of timing, it will be blocked, and the task with the highest priority in the ready task list will be executed. For example, if the CAN acquisition task gets the execution right, the devices on CAN interface will be traversed. The timing time set here is the device acquisition cycle on the RS485 interface divided by the number of devices on RS485 interface. After the traversal, the reading continues from the list's head.

D. *6LoWPAN Forwarding Task*

The 6LoWPAN network forwarding task is responsible for forwarding the data received by the acquisition task to the 6LoWPAN module, so that other devices in the 6LoWPAN network or in the Internet can read the data collected by the gateway. The process of this task is to traverse the linked list of device configuration information for each interface, and then send the collected data to the 6LoWPAN module. The microcontroller sends the gateway's last data collected to the 6LoWPAN module, and the transmission cycle is fixed at 100ms.

IV. Experiment Tests

A. Construction of the Test System

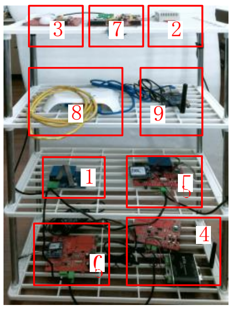

Fig. 5. Physical photos of the test system.

TABLE I. Sensor Information

Type	*Interface*	*Device address*	*Start address*	*Register length*	*Name*
PM2.5	RS485	1	0	1	"PM25(ug/ml)"
Temperature	CAN	1	0	1	"Temp(℃/100)"
Light	LoRa	1	5	1	"Light(Lux)"
VOC	ZigBee	1	19	1	"VOC(ppm)"
Humidity	4G	1	3	1	"Humi(%RH/10)"
CO_2	GPRS	2	1	1	"CO_2(ppm)"

In order to verify the function of the gateway, this paper builds a test system, which consists of multi-interface acquisition gateway, six different types of sensors, 6LoWPAN edge gateway and an IPv4 router with DHCP function. The actual test system is shown in the Fig. 5, and the information of the six sensors is shown in the Table I. The six sensors have different parameters and different communication interfaces. Modbus address, start register address and read register length are the necessary information to complete the reading. It should be noted that the physical interfaces of these sensors are different, and it will not have any effect even if these devices have the same Modbus address. Named items give physical meaning to the sensor data to increase the readability of the test results.

B. Testing Process

In the test of the gateway, the gateway needs to be configured first. During the configuration, a USB cable is used to connect the gateway to the computer. After the configuration of the gateway with the help of computer software, the system test starts. The connection relationship of the system is shown in the Fig. 6.

The working principle is as follows. Firstly, the IPv4 router allocates IPv4 addresses to the computer and the 6LoWPAN edge router, generating the LAN. Then, the 6LoWPAN edge router creates the RPL network as the ROOT node. Moreover, the 6LoWPAN acquisition gateway is added to the RPL network. The 6LoWPAN edge router maps the IPv6 address and port of the 6LoWPAN acquisition gateway to a port on its own IPv4 address with the NAT technology, thus communication between the computer and the 6LoWPAN acquisition gateway is established. The resources of the 6LoWPAN acquisition gateway can be accessed on the computer using the Firefox browser with the Coap protocol debugging plug-in, where the 6LoWPAN acquisition gateway acts as the Coap server and the Firefox browser acts as the Coap client.

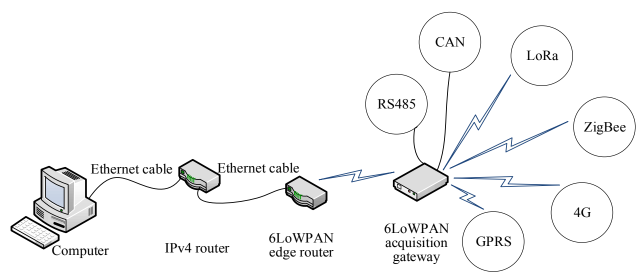

Fig. 6. The structure of the test system.

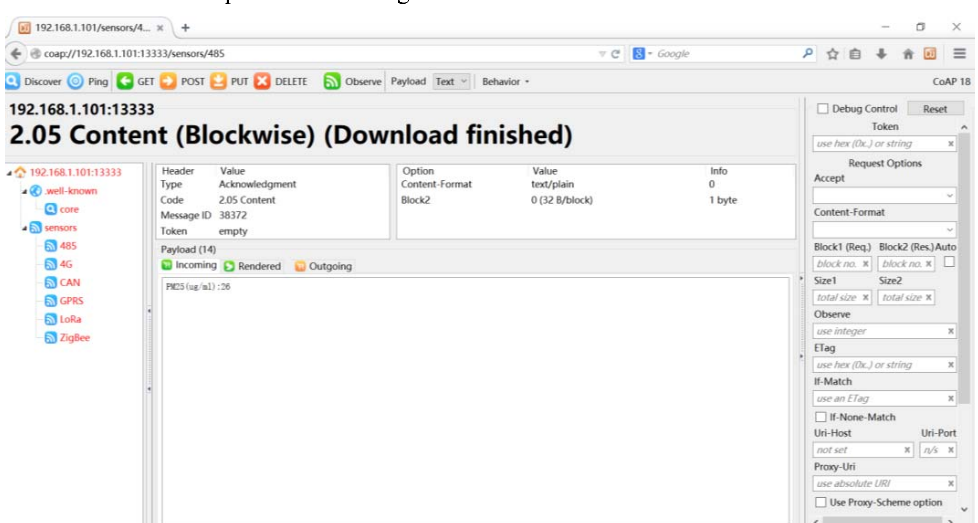

Fig. 7. The resource accessed of the RS485 interface.

C. Test Results

Fig. 7 shows the interface of 6LoWPAN sensor/485 resource accessed in Firefox. The incoming line reads "PM25(ug/ml):26". It means the PM2.5 density was read as 26 ug/ml on the RS485 interface. The access results for each interface are shown in Table II. The data collected by each interface can be correctly read on the browser, indicating that the gateway is working normally in accordance with the expected configuration and its functional indicators are qualified.

TABLE II. THE RESULTS ACCESSED BY EACH INTERFACE

Interface	*Result*
RS485	"PM25(ug/ml):26"
CAN	"Temp(℃/100):3037"
LoRa	"Light(Lux):412"
ZigBee	"VOC(ppm):152"
4G	"Humi(%RH/10):500"
GPRS	"CO_2(ppm):614"

V. CONCLUSION

This paper presents the design and implementation of a multi-interface data acquisition gateway based on 6LoWPAN, and validates its functionality. Six physical interfaces are electrically independent and do not affect each other. Modular design improves the tailoring and expansion of gateway. At the same time, the physical layer of each interface is configurable, which expands the scope of application of gateways. The Modbus protocol is used in the application layer of the interface, so that multiple devices can be connected on one interface. Compared with the mode of one sensor with one 6LoWPAN node, the size of the 6LoWPAN node is greatly reduced if this gateway is used, which directly reduces the routing complexity, system cost and power consumption of the 6LoWPAN network. Software design is based on FreeRTOS real-time system, With the resources provided by FreeRTOS to manage and schedule each task, the program can run stably and improve the overall performance of the software.

In order to verify the function of the gateway, a test system is built. The test results show that the gateway can complete the multi-interface data acquisition task according to the expected configuration. This research has great significance on how to apply 6LoWPAN in multi-sensor situation. In the future, we will further study based on engineering needs, test, analyze and evaluate various performance indexes of the multi-interface acquisition gateway, compare and study the performance of 6LoWPAN network under two different modes, single sensor single node and multi-sensor single node, to promote the application and development of 6LoWPAN.

ACKNOWLEDGMENT

This work is supported by CERNET Innovation Project (NGII20190303), and by National Natural Science Foundation of China (61803067), and by the Fundamental Research Funds for the Central Universities of China (DUT20JC45, DUT20JC15).

REFERENCES

[1] M. Vo, N. Do, V. Tran, Q. Ma, C. Le and L. Mai, "A multi-storey building actuator and sensor system using 6L0WPAN based Internet of Things: Practical design and implementation," 2018 2nd International Conference on Recent Advances in Signal Processing, Telecommunications & Computing (SigTelCom), Ho Chi Minh City, 2018, pp. 176-181.

[2] S. Guo, W. Luan and X. Su, "The design of a remote meter acquisition module based on 6LoWPAN," 2016 China International Conference on Electricity Distribution (CICED), Xi'an, 2016, pp. 1-4.

[3] I. Singh and D. Kumar, "Improving IOT Based Architecture of Healthcare System," 2019 4th International Conference on Information Systems and Computer Networks (ISCON), Mathura, India, 2019, pp. 113-117.

[4] N. H. Kumar, S. Baskaran, S. Hariraj and V. Krishnan, "An Autonomous Aquaponics System Using 6LoWPAN Based WSN," 2016 IEEE 4th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW), Vienna, 2016, pp. 125-132.

[5] E. D. Dimaunahan, A. D. S. Santos, E. F. H. Paloma, et al., "A 6LoWPAN-Based Thermal Measurement, and Gas Leak for Early Fire Detection Using Artificial Neural Network," In Proceedings of the 2019 5th International Conference on Computing and Artificial Intelligenc, Association for Computing Machinery, New York, NY, USA, 2019, pp.170–174.

[6] Y. L. K. Macayana, V. A. Coronel, I. C. Fernandez, et al., "Implementation of 6LoWPAN and Controller Area Network for a Smart Hydroponics System," 2019 Global IoT Summit (GIoTS), Aarhus, Denmark, 2019, pp. 1-4.

[7] L. Luo, "Data Acquisition and Analysis of Smart Campus Based on Wireless Sensor," Wireless Pers. Commun., vol. 102, pp. 2897–2911, 2018.