Design and Implementation of MCU-Based Reconfigurable Protocol Conversion Module for Heterogeneous Sensor Networks

Lei Zhao and Yan Zhang(✉)

Beijing Institute of Technology, Beijing 100081, China
zhangy@bit.edu.cn

Abstract. With the widespread applications of the Internet of Things (IoT), the heterogeneity of communication protocols and data frame formats in the sensor networks has become a significant issue. To address this problem, in this paper we design a reconfigurable protocol conversion module for heterogeneous sensor networks. Based on a microcontroller unit (MCU) hardware platform, the communication protocol conversion among RS232, CAN, and Ethernet is implemented. The upper computer software is developed to receive data from the sensor networks, configure the hardware platform communication parameters, and adjust the monitoring parameters to the normal range. Moreover, aiming to solve the problem of data frame format differences in various IoT applications, we design a reconfigurable scheme to customize data frame formats, enabling the system's support for heterogeneous sensor networks. The experimental results validate the feasibility of the reconfigurable protocol conversion module, and show that the system can transmit, receive, and manage the sensor data effectively.

Keywords: Heterogeneous sensor networks · MCU · Reconfigurable protocol conversion · Upper computer

1 Introduction

In recent years, the unprecedented connectivity in the IoT enables many areas, such as smart factories, smart cities, intelligent homes, and so on [1]. In variable IoT applications, sensor networks play a fundamental role in data acquisition, data transmission, and device monitoring [2,3]. However, as the scale and category of sensors continue to grow, the heterogeneity of sensor networks has become a significant issue to cope with. For sensors with different functions and the same sensors from different manufacturers, different communication protocols and data formats are used to transmit data between sensors and the central controller. As a result, it is difficult for the central controller to acquire data and manage the sensor networks.

B. Li et al. (Eds.): IoTaaS 2020, LNICST 346, pp. 171–181, 2021.
https://doi.org/10.1007/978-3-030-67514-1_14

The problem of heterogeneity in sensor networks exists in many application scenarios. For instance, an intelligent building in [4] needs to monitor the devices under different fieldbuses comprehensively. In the industrial field, gathering data in a heterogeneous sensor network is a challenge for the real-time monitoring center of continuous steel casting [5]. Additionally, for personal healthcare devices [6] or electric power systems [7], communications between different protocols are also a significant issue.

Some studies have been completed to try to tackle this problem. In [8], a heterogeneous network integration module was developed to integrate data transmission in the Zigbee-WiFi hybrid network. Moreover, based on microprocessor control boards, a module was designed in [9] and [10] for communication between the Controller Area Network (CAN) and Ethernet protocols. Additionally, a hardware unit was designed in [11] to take commands and data from transmitter to receiver, working on SPI bus and I2C bus, respectively. Also, conversions between multiple communication protocols are implemented in [12–14], which enables the heterogeneous systems to transfer data freely. Nevertheless, most of the works mainly focused on one-to-one protocol conversion. Even though some studies implemented multi-protocol conversion, these systems only support fixed and limited communication protocols. If an already designed system needs to add some new sensors with new communication protocols, the existing schemes cannot be applied to manage these new sensors directly. Therefore, it is necessary to develop a scalable protocol conversion, which not only supports several fixed communication protocols but also expands to support new protocols when necessary. What is more, different sensors usually use different data frame format to transmit data, which hinders data acquisition, but the differences of transmitted data frame format of different sensors have not been considered in existing studies.

In this paper, taking the smart factories as application scenarios, we design a reconfigurable protocol conversion module (PCM) to deal with data transmission in heterogeneous sensor networks. Both the scalability of communication protocol and the reconfiguration of data frame format are taken into consideration. Specifically, the PCM in this paper consists of MCU hardware platform and upper computer software. The MCU hardware platform is equipped with RS232, CAN bus, and Ethernet protocol interfaces. Meanwhile, the upper computer software is developed to receive data from the sensor networks, configure the hardware platform communication parameters, and adjust the monitoring parameters to the normal range. Besides, the sensors deployed in the factory to obtain variable operating parameters, such as temperature, humidity, etc., can communicate with the hardware platform through RS232, CAN bus interfaces. Then, the upper computer software as a central controller transmits data with the hardware platform using the Ethernet protocol.

The scale of the sensor network and the number of communication protocol types supported by our PCM can be expanded according to needs. In addition, a reconfigurable scheme is designed to customize data frame formats, enabling the system's support for heterogeneous sensor networks. As a result, the PCM can communicate with different sensors in a unified manner, making it easier to manage the whole sensor network.

The remainder of this paper is organized as follows. Section 2 describes the design architecture of the PCM. Section 3 elaborates on the implementation of the system from hardware and software aspects. Experimental results will be illustrated in Sect. 4. Finally, Sect. 5 draws the conclusions.

2 Design of the System Architecture

Given that in a smart factory scenario, many different sensors are deployed in the factory to monitor the environmental parameters, such as temperature, humidity, etc. As the communication protocols and data frame formats that each sensor uses are different, it is necessary to deploy a PCM to acquire data and manage the sensor network as a whole.

The system architecture is shown in Fig. 1, where each sensor in the sensor networks is connected with the PCM through different protocols. The PCM contains an upper computer, a network switch, and many MCU hardware platforms. Moreover, many sensors with different functions and protocols constitute a small heterogeneous network. And the upper computer can communicate with the PCM by Ethernet protocol only, which is widely used in industry and academia.

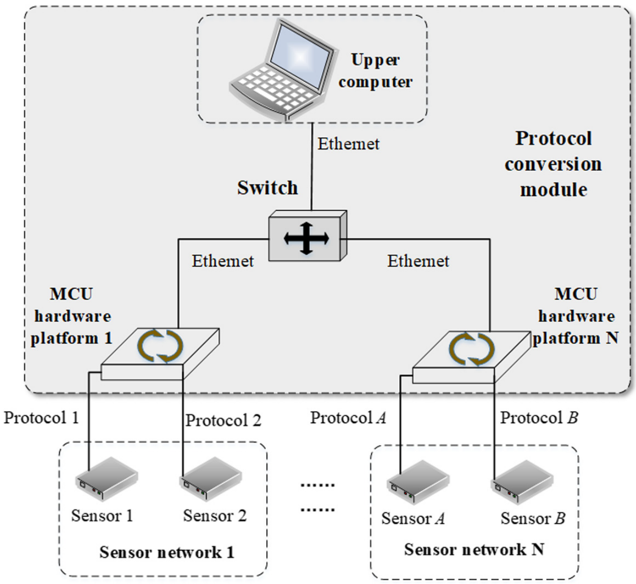

Fig. 1. The system architecture of protocol conversion module, upper computer, and sensor networks.

The details of inside functional blocks of our designed PCM are given in Fig. 2. It is noteworthy that we select the two most commonly used communication protocols in industrial fields as examples for implementation, i.e., RS232 and CAN. RS232 is a standard protocol for serial communications. Besides, CAN bus is a message-based protocol, which supports communications between microcontroller and devices and is widely used in industrial automation and vehicles. However, it should be noted that this architecture can also be used for the conversion between other protocols.

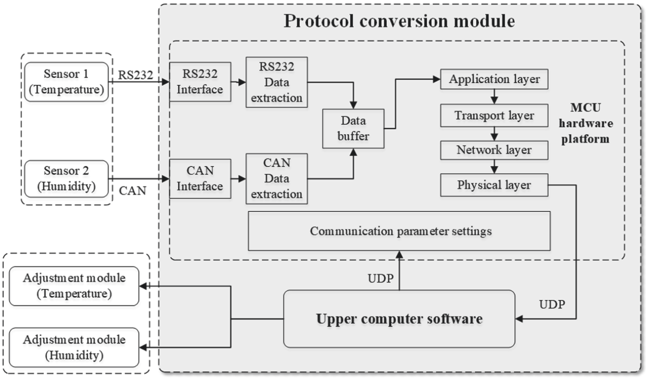

Fig. 2. A functional block diagram of the whole system.

As shown in Fig. 2, the whole system can be roughly divided into three parts: the sensor networks, the protocol conversion module, and the adjustment modules. Specifically, the MCU hardware platform within the PCM receives data from sensors and transmits data to the upper computer software. Then, the upper computer software verifies whether the data conforms to the selected data frame formats and check whether the parameters are within the normal range. If the parameters are out of the normal range, the adjustment modules will receive commands from the upper computer software to correct the environmental parameters. For temperature, the adjustment modules can be a combination of fans and heaters. Likewise, some machines which can change the environmental humidity serve as the humidity adjustment module.

2.1 MCU Hardware Platform Design

To meet the requirements of multi-protocol conversion, the hardware platform is supposed to have multiple communication interfaces, including RS232, CAN, and Ethernet. Thus, a hardware development board in which a microcontroller unit chip and external circuits are embedded is an appropriate option for the implementation of the PCM.

The functional blocks included in the hardware platform and the workflow are presented in Fig. 2. Firstly, RS232 and CAN interfaces embedded in the board receive data from sensor 1 and sensor 2, respectively. The two data flows are transmitted through corresponding communication protocols, i.e., RS232, CAN. Secondly, the data extraction blocks of RS232 and CAN obtain data from RS232 and CAN interfaces and store these data into the data buffer block. Next, a lightweight TCP/IP (lwIP) stack is utilized to transfer the data in the data buffer block to upper computer software through four TCP/IP layers. Additionally, because of its advantages of fast speed and higher security, User Datagram Protocol (UDP) is selected as the transport layer protocol to connect the MCU hardware platform and the upper computer software.

2.2 Upper Computer Software Design

The upper computer software serves as a central controller to receive data from the MCU hardware platform and maintain the environmental parameters within the normal range. Moreover, the upper computer software provides the interfaces by which operators can configure the communication parameters settings.

Typically, different sensors use different data frame formats to transmit data. Only when the data frame formats of sensors are known, the upper computer software can obtain the transmitted data of the sensors. Assuming that the data frame formats of sensors are already known as shown in Fig. 3, four data fields of frame formats are taken into consideration. The details of these data fields are elaborated on as follows.

No.	Start bit	Data length	Data type	Precision

Fig. 3. The data fields of predefined data frame format.

- *No.* Technically, this is not the data field. Instead, it is the numerical order of the predefined data frame format.
- *Start bit.* The transmitted data will be stored in a buffer data array. And the start bit means where to start inputting data in the buffer data array.
- *Data length.* The length of transmitted data digits.
- *Data type.* Two kinds of data types are considered, i.e., integer and floating-point.
- *Precision.* Precision here means how many digits after the decimal point is retained in the data.

After receiving data through UDP protocol, the upper computer software will verify if the received data meet the requirements of data frame formats of the corresponding sensors. Then, the received data will be displayed on the software graphic user interfaces (GUIs).

As for each environmental parameter in a factory, there must have a normal range. Suppose $T_{\max}, H_{\max}$ are the maximum values of the temperature and humidity, respectively. Similarly, the minimum values of the temperature and humidity are denoted as $T_{\min}$ and $H_{\min}$, respectively. t and h represent the current values of the sensors. If $t - T_{\max} > \Delta T$ or $T_{\min} - t > \Delta T$, the upper computer software sends feedback commands to the adjustment module of temperature to restore the environment temperature to the normal range. Here, ΔT is the maximum offset of temperature. Likewise, the upper computer software is going to do the same thing when the humidity is out of the normal range.

2.3 Reconfigurable and Scalable Design

Considering that in the field of industrial IoT, it is necessary for the network operator and sensors to configure the data format of transmitted data. To establish a reconfigurable protocol conversion system, data frame formats are provided to set. The network operators and sensors are able to configure each data field in the data frame format according to their own needs.

In some scenarios, the monitoring networks need to enlarge the scale due to the addition of new sensors and communication protocols. However, a fixed network hardly supports such a demand. Therefore, it is necessary to develop a scalable protocol conversion system. In our design, the scalability of the network is taken into account from the hardware and software aspects.

As for the hardware, if there is a single sensor network, the upper computer can connect with the MCU hardware platform by the Ethernet protocol directly. Meanwhile, if there are multiple sensor networks, a network switch is needed between the upper computer and the MCU hardware platforms. As long as the new hardware platform has the corresponding protocol interfaces, the network switch can forward the data from different sensor networks to the upper computer without special configurations beyond plugging in cables. As for the upper computer software, it is easy to reserve extended functions for newly added networks.

3 System Implementation

3.1 MCU Hardware Platform Implementation

In order to receive data from different protocol interfaces, an STM32F407 MCU chip embedded development board is used. As Fig. 4 depicted, this board is appropriate for the implementation of multiple protocol conversion since it supports many communication interfaces, such as RS232, RS485, CAN bus, Ethernet, and so on. Thus, commonly used RS232 and CAN are used as physical layer protocols of the sensors in this paper. Besides, with the Ethernet related processor chips and circuits, this board is able to transfer data with the upper computer through TCP/IP protocol.

The hardware mainly consists of four parts. 1. RS232 and CAN interfaces. 2. RS232 and CAN data extraction. 3. Data buffer. 4. Ethernet block. Firstly, initializing the configurations of the RS232, CAN, and Ethernet block. Then,

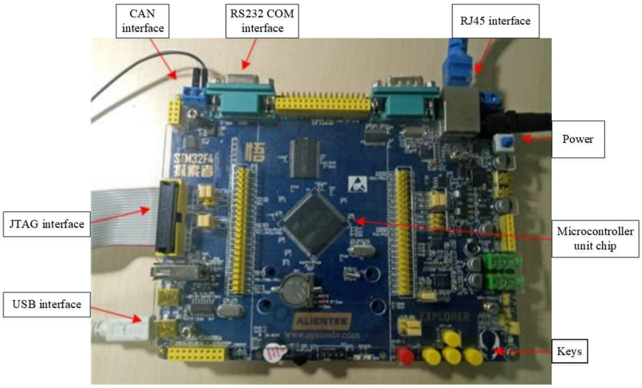

Fig. 4. MCU-based hardware development platform.

with the UART and related chips and circuits, RS232 and CAN nodes receive raw data from corresponding sensors by using interrupt functions. Next, the data extraction block obtains original transmitted data from RS232 and CAN nodes. All data are stored in the data buffer block temporarily, which is made up of arrays. As for the Ethernet block, the physical layer chip LAN8720 should be configured first. After the lwIP kernel is initialized, remote and local static IP addresses and port numbers are ought to be determined because the Dynamic Host Configuration Protocol (DHCP) is turned off here. Finally, data can be transferred from the data buffer block to the upper computer software.

3.2 Upper Computer Software Development

The upper computer software is developed in C# programming language based on Windows Forms, which is a set of libraries designed to develop applications in the .NET framework. The upper computer software will select the same data frame formats as the corresponding sensors from the list of predefined ones, which is given in Table 1.

Table 1. List of predefined data frame formats.

No	Start bit	Data length	Data type	Precision
1	2	3	Integer	0
2	1	6	Floating point	2
3	3	5	Floating point	1
4	4	5	Floating point	2

On the upper computer software side, socket class in C# is selected to implement network communications. After configuring the network parameters, including remote and local IP addresses and port numbers, the upper computer software starts to receive data from the PCM in an asynchronous way.

4 Experimental Results

In this section, some experiments are conducted to validate whether the system realizes the functions of protocol conversion. If the data at the sensors side are accurately transferred to the upper computer software, it means that the system has realized the function of communication protocol conversion successfully.

To customize the data frame formats employed by different sensors, software simulated sensors running on a separate computer are developed to mimic the behaviors of the real sensors. Therefore, the upper computer software can configure the data frame formats according to the known formats on the simulated sensor sides. The simulated sensors transmit data to the MCU hardware platform through the RS232 and CAN interfaces. Moreover, the adjustment modules are integrated into the software simulated sensors to modify the values of parameters when receiving commands from the upper computer.

To start the process of validation, the serial port related parameters should be configured first. As Table 2 presented, the parameter settings of serial ports of the temperature and humidity sensors are given, respectively.

Table 2. Serial port parameter settings of the temperature and humidity sensors.

	Port	Baud rate	Data bit	Stop bit	Check bit
Temperature	COM4	115200	8	2	Even
Humidity	COM1	57600	8	1	None

Table 3. IP addresses and port numbers configurations of the MCU board and upper computer software.

MCU board		Upper computer software	
IP address	Port number	IP address	Port number
192.168.1.30	8089	192.168.1.101	8089

Secondly, as shown in Table 3, the IP addresses and port numbers of the MCU-based hardware development board and the upper computer software need to be determined, respectively.

After selecting a specific data frame format for the two sensors, we can start the process of sending and receiving data. Figure 5 displays the data frame format selections, local area network configurations, and the results of received data. Comparing with the current values of temperature (23.6 °C) and humidity (41%),

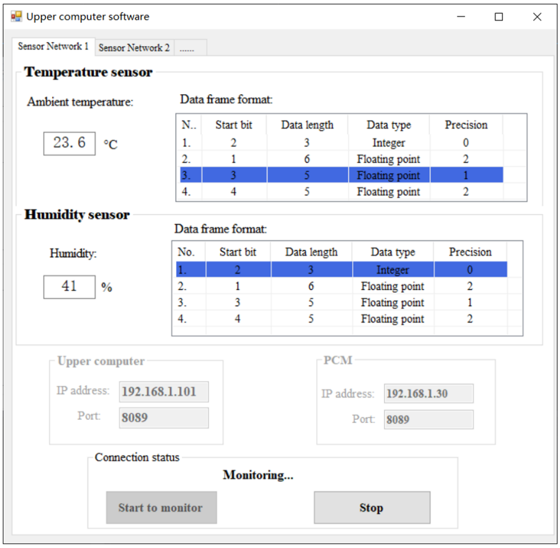

Fig. 5. Screenshot of the upper computer software.

the received data in Fig. 5 are correct. As a consequence, it turns out that the whole system has realized the function of communication protocol conversion well.

Figure 6 presents the changing process of temperature parameter over time. Here, the maximum and minimum values of temperature $T_{\max}, T_{\min}$ are set as 30 °C and 20 °C, respectively. The offset ΔT is 1 °C. As shown in Fig. 6, when

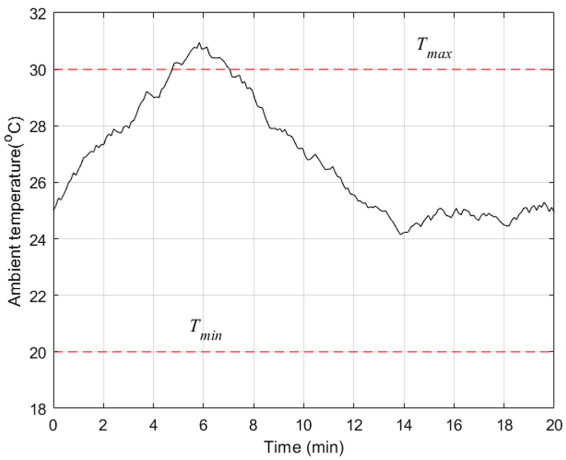

Fig. 6. Temperature parameter changes over time.

the temperature exceeds the maximum and minimum levels, the parameter will be adjusted to the normal range.

Similarly, the changing process of humidity parameter over time is illustrated in Fig. 7. Here, the maximum and minimum values of humidity $H_{\max}, H_{\min}$ are set as 45% and 40%, respectively. The offset ΔH is 2%. As shown in Fig. 7, the humidity parameter will be adjusted to the normal range if the differences between current humidity and maximum or minimum values are greater than ΔH.

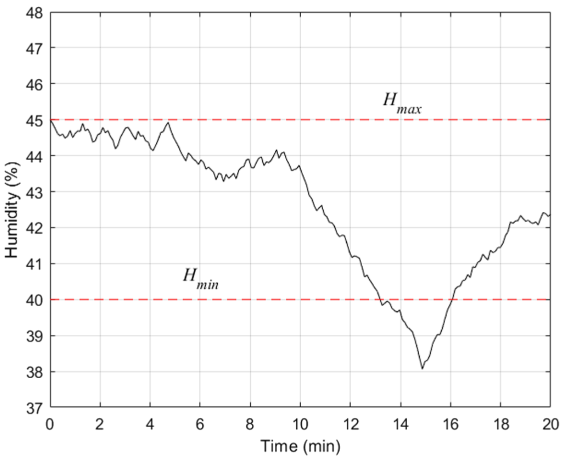

Fig. 7. Humidity parameter changes over time.

5 Conclusions

In this paper, we have designed and implemented a reconfigurable PCM to solve the data acquisition and sensor management problems in the heterogeneous sensor networks. By introducing the predefined data frame formats, a reconfigurable scheme has been designed to customize data frame formats, enabling the system's support for heterogeneous sensor networks. Meanwhile, an MCU-based hardware platform is utilized to implement the multiple communication protocol conversion between RS232, CAN, and Ethernet. Furthermore, the upper computer software has been developed to receive data, configure the communication settings, and adjust the environmental parameters. Finally, validation experiments are conducted to demonstrate that the designed system can realize the function of multiple protocol conversions and ensure that the environmental parameters are within the normal range.

Acknowledgment. This work was supported by the National Key R&D Program of China under Grant 2019YFE010391, the National Natural Science Foundation of China under Grant 61871035, and the National Defense Science and Technology Innovation Zone.

References

1. Paniagua, C., Delsing, J.: Industrial frameworks for internet of things: a survey. IEEE Syst. J. **1–11** (2020). https://doi.org/10.1109/JSYST.2020.2993323
2. Kocakulak, M., Butun, I.: An overview of Wireless Sensor Networks towards internet of things. In: IEEE Annual Computing and Communication Workshop and Conference, pp. 1–6. IEEE, Las Vegas (2017)
3. Kuo, Y., Li, C., Jhang, J., Lin, S.: Design of a Wireless Sensor Network-Based IoT platform for wide area and heterogeneous applications. IEEE Sens. J. **18**(12), 5187–5197 (2018)
4. Xie, J., Gao, Q.: Design and implementation of embedded protocol conversion gateway for intelligent buildings. In: 2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS), pp. 1–5. IEEE, Beijing (2019)
5. Zhang, F., Liu, M., Zhou, Z., Shen, W.: An IoT-based online monitoring system for continuous steel casting. IEEE Internet Things J. **3**(6), 1355–1363 (2016)
6. Woo, M.W., Lee, J., Park, K., et al.: A Reliable IoT system for personal healthcare devices. Future Generation Comput. Syst. **78**(2), 626–640 (2018)
7. Zhu, J., Chong, X., Tao, Y.: An electric power system protocol conversion method based on embedded platform. In: 2016 2nd International Conference on Artificial Intelligence and Industrial Engineering (AIIE 2016), pp. 179–182. Atlantis Press (2016)
8. Chen, H.: Heterogeneous network integration based on protocol conversion. In: Proceedings of the 35th Chinese Control Conference, pp. 6888–6893. IEEE, Chengdu (2016)
9. Eum, S.: Implementation of protocol conversion control board for industrial communication. Int. J. Control Automa. **9**(6), 201–208 (2016)
10. Zhang, Y., Feng, X., Guo, Y.: Design of ethernet-CAN protocol conversion module based on STM32. In: International Conference on Future Generation Communication and Networking, vol. 7(1), pp. 89–96 (2014)
11. Trivedi, D., Khade, A., Jain, K.K., Jadhav, R.: SPI to i2c protocol conversion using verilog. In: 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), pp. 1–4. IEEE, Pune (2018)
12. Ma, S., Liu, F.: Multi-interface gateway protocol conversion method for nearspace-air-vehicle-ground dedicated network. DEStech Trans. Comput. Sci. Eng. CCNT **367–373** (2018)
13. Lou, G., Cai, H.: Research and implementation of ARM-based fieldbus protocol conversion method. In: International Conference on Computer and Communication Technologies in Agriculture Engineering, pp. 260–262. IEEE, Chengdu (2010)
14. Cong, P.-X., et al.: The design of multi-interface protocol adaptive conversion distribution network communication device based on wireless communication technology. In: 2017 IEEE Conference on Energy Internet and Energy System Integration (EI2), pp. 1–4. IEEE, Beijing (2017)