ATLANTIS PRESS

Advances in Intelligent Systems Research (AISR), volume 142

2017 International Conference on Information Technology and Intelligent Manufacturing (ITIM 2017)

Design and implementation of embedded network master and slave communication based on Modbus/TCP

Chenxi Wang
Shenyang institute of Automation Chinese Academy of Sciences
Shenyang China
wangchenxi@sia.cn

Mingzhe Liu, Aidong Xu, Jilong Zhang and Ni Jin
Shenyang institute of Automation Chinese Academy of Sciences
Shenyang China
lmz@sia.cn, xad@sia.cn, zjlong@sia.cn, jinni@sia.cn

***Abstract*—By analyzing the components of the Modbus/TCP protocol, this paper designs the communication network of industrial Ethernet based on Modbus/TCP protocol. The realization method and process of client / server communication mode are given.**

Keywords—Modbus/TCP, Master/Slave, Embedded network

I. INTRODUCTION

With the rapid development of computer and network technology, information technology has gradually entered the field of industrial automation. In recent years, the application of Ethernet and TCP/IP protocol has become a hotspot in the research and development of industrial control network. Modbus/TCP protocol is a kind of industrial Ethernet standard, which uses simply, has good openness and is widely used by many manufacturers and associations.

Modbus is the world's first truly industrial fieldbus protocol. Through the Modbus protocol, the control equipment of different manufacturers can form an industrial network, and the centralized control and management of the equipment in the network is also very convenient. In the industrial control network, the computer, the controller and the instrument and so on can easily realize the interconnection and the data exchange through the Modbus.

II. MODBUS/TCP PROTOCOL

Modbus/TCP is the Modbus protocol embedded into a TCP/IP protocol. It implements the TCP/IP master/slave architecture for Modbus message communication on the Ethernet. Modbus defines a simple protocol data unit PDU that is independent of the underlying communication layer. The mapping of the Modbus protocol embedded in different bus or network will introduce some additional domains on PDU to form the application data unit. As shown in Figure 1, the Modbus request/response package on the TCP/IP protocol is coupled with a MBAP structured message header.

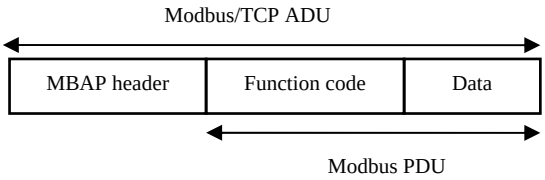

Fig.1 Modbus Data frame format

The structure of the MBAP message header is shown in Table 1

TABLE I. THE STRUCTURE OF THE MBAP MESSAGE HEADER

Field	Length	Describe
Transaction identifier	2 byte	The transaction identifier of Modbus request / response
Protocol identifier	2 byte	OX0000（Modbus protocol）
Length	2 byte	The number of bytes from the unit identifier to the end of the data field
Unit identifier	1 byte	Identification code for remote slave stations connected to serial links or other buses

The length of the PDU in frame format will change, because of the different function code. But the message frame format must strictly follow the ADU structure, to ensure that both sides of the communication can accurately identify the received data frame. The client establishes Modbus ADU, and the function code indicates to the server which operation to perform.

III. REALIZATION OF COMMUNICATION BETWEEN MODBUS/TCP MODULE AND REMOTE SLAVE STATION

A. Realization of Modbus/TCP module as master station

The design of the Modbus/TCP specification makes the realization of the main station as simple as possible. The basic transaction process is as follows: Use connect ()function to establish a connection to the TCP502 port of the target device;

Prepare Modbus message, submit a message containing the Modbus/TCP prefix, and use send () function to send; To wait for the answer in the same connection, you can also use the method of timing and use select (), so that the timing can be debug faster; Read the response with receive () function; Then use receive () function to read all reply messages; When there is no communication task, shut down the TCP connection so that the remote slave station resource is idle can be used by other devices. The maximum interval at the master station to open the connection is 1 second.

If you use a timed method to wait for an answer, close the connection from one side, open a new connection and resubmit the request message. This technology allows the master station to be Again, which is better than the default by TCP provided. At the same time, it also considered alternative methods, such as submitting the request to the standby IP address, using a fully independent communication network to prevent connection failure due to the lower structure of the network.

B. Realization of Modbus/TCP module as slave station

The basic transaction process is as follows: Use listen () function to monitor TCP502 port connection requests; If you receive a connection request, use accept () function to receive and generate a new thread to handle the connection; In the new thread, use receive () function to establish a request to receive six bytes of Modbus/TCP header, wait until the request succeeds or closes the connection.

Analyze the header, if it looks corrupted (if the protocol field is non-zero or message length is less than 256), close the connection from one side. This answer indicates that the TCP decoding is not correct.

Using receive () function to receive the remaining messages, the message length is known; In particular, receive () function with message length restrictions can withstand the communication requests continuously sent by the customer using "pipeline technology"; When the current request has complete service; The communication request for any pipe technology is left in the TCP buffer from the station end to end or master station. Handling Modbus messages, when necessary suspends the current thread until the correct answer; Final return to normal response message or exception reply; Construct the Modbus/TCP prefix, from the requested byte 0 to 1 copy the transaction identifier field and recalculate the length field; The submission includes a reply, including Modbus/TCP, and returns using send () function, waiting to read the next prefix.

Finally, when the primary station end closes the connection, the prefix obtained by the receive () function is discarded. Normal shutdown usually causes the return byte count of receive () function to be 0, and mandatory shutdown causes receive () function to produce the wrong return value. In any case, close connection can cancel the current thread.

Modbus/TCP communication thread flow diagram as shown:

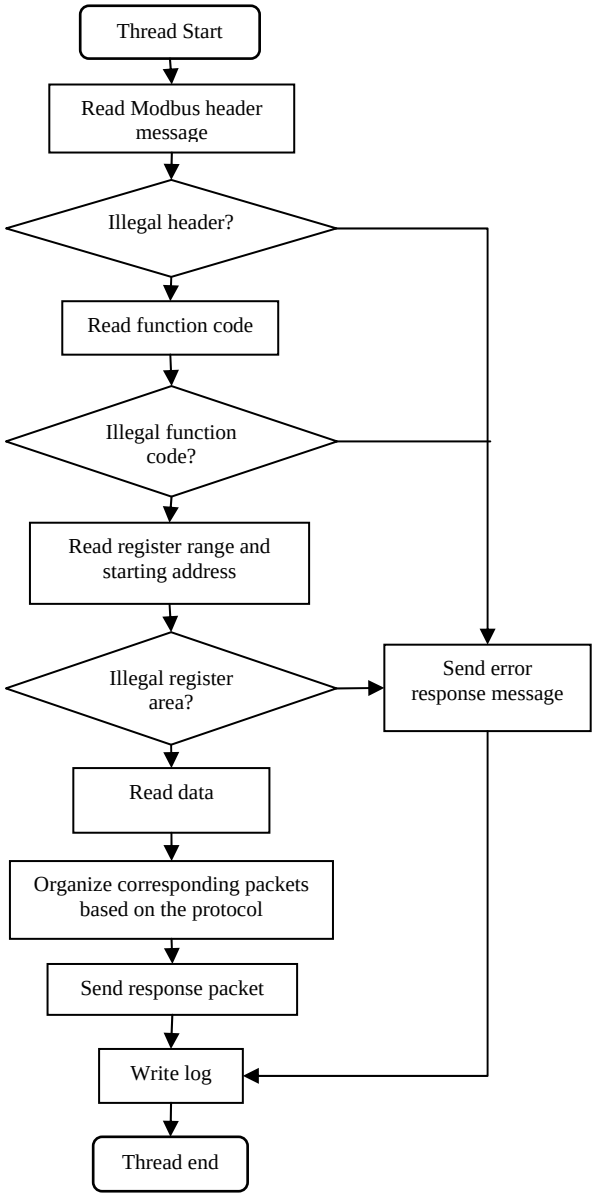

Fig 2. Modbus/TCP communication flow chart

IV. REALIZATION OF COMMUNICATION BETWEEN MODBUS/TCP MODULE AND CONTROLLER

A. Communication between the Modbus/TCP module as master station and the controller

When the Modbus/TCP module sends commands, it reads the related configuration of the master station in the configuration file to the EEPROM firstly, and then sends it to the remote slave station by the master station task. There is no data exchange between the Modbus/TCP module and the controller during this process.

When the Modbus/TCP module receives information from the remote slave station, it places the data values into the ReadData[master] in the order by the configuration file, and

sends the array to the controller. The controller stores the data values in the order by resolving the configuration file.

B. Communication between the Modbus/TCP module as salve station and the controller

Establishment of high order interrupt, so that the Modbus/TCP module and controller to continue the cycle of read and write operations. The controller transfers the data point information to the slave station module through the WriteData function, and receives the data demand of the remote master station. The slave station module writes the parameter value into the controller through ReadData[slave] function, and completes the write parameter command for the remote master station

The communication structure diagram of the remote slave station, Modbus / TCP communication module and controller is shown as Figure 3:

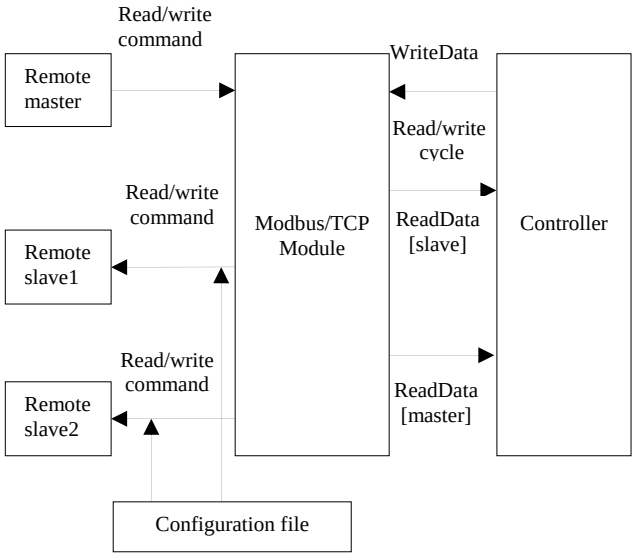

Fig 3. Modbus/TCP communication structure diagram

V. CONCLUSION

The industrial network transmission based on Modbus/TCP can realize remote monitoring in any place involved in the internet. Its advantages are remote transmission distance, high cost performance, low implementation cost, can use the existing network facilities, versatility, and can be applied to various electrical appliances needing network monitoring function. This article analyzes the key technologies of Modbus/TCP frame structure; master/slave communication mode, etc. It gives the design procedure and method of communication program between master station and slave station. And it gives the communication design process between Modbus/TCP module and controller. It has great significance for the design and optimization of Modbus /TCP network.

REFERENCES

[1] Deng.Xinru, Present situation and development of Modbus_TCP Industrial Ethernet

[2] Sun Lu, Sun Lu Application of Modbus_TCP protocol in remote monitoring.

[3] Jiang Bin, Design of industrial control network based on Modbus_TCP.

[4] Huang Jianjun, Research and implementation of communication based on ModbusTCP_IP.