Design of real-time data compression wireless sensor network based on LZW algorithm

CHENG Ya-li, LI Jin-ming, CHENG Nai-peng

(*School of Instrument and Electronics, North University of China, Taiyuan 030051, China*)

Abstract: A real-time data compression wireless sensor network based on Lempel-Ziv-Welch encoding (LZW) algorithm is designed for the increasing data volume of terminal nodes when using ZigBee for long-distance wireless communication. The system consists of a terminal node, a router, a coordinator, and an upper computer. The terminal node is responsible for storing and sending the collected data after the LZW compression algorithm is compressed; The router is responsible for the relay of data in the wireless network; The coordinator is responsible for sending the received data to the upper computer. In terms of network function realization, the development and configuration of CC2530 chips on terminal nodes, router nodes, and coordinator nodes are completed using the Z-stack protocol stack, and the network is successfully organized. Through the final simulation analysis and test verification, the system realizes the wireless acquisition and storage of remote data, and reduces the network occupancy rate through the data compression, which has a certain practical value and application prospects.

Key words: wireless sensor network; ZigBee; LZW algorithm; data compression

CLD number: TP274.2 **Document code**: A

Article ID: 1674-8042(2019)02-0191-08 **doi**: 10.3969/j.issn.1674-8042.2019.02.012

0 Introduction

In wireless sensor networks, the commonly used wireless communication technology is ZigBee technology, which has characteristics of low cost, low power consumption, good transmission stability, and easy to use[1]. Under normal circumstances, the processing core on the wireless sensor network terminal node is MCU or STM32 and other low-power processing chips, but it can only be applied to the monitoring of the small amount of transmitted data. If the amount of data generated is relatively large, the use of the channel will be relatively high, then the use of MCU processing core is no longer suitable, which may affect the normal communication of other devices in the network[2].

For solving the field programmable gate array (FPGA) is used as the processing core of the terminal node in this design. In order to ease the data processing pressure and ensure the integrity of the data, the design adopts lossless compression technology to the collected data[3]. Compared to Huffman encoding, which is more complex to implement on hardware, Lempel-Ziv-Welch (LZW) algorithm is simple to encode, and compression and decompression are fast. After system testing, data compression ratio can reach 1.8 : 1 or more, the effect is better, and it is easy to implement on hardware. Therefore, a real-time data compression wireless sensor network based on the LZW algorithm is designed. This design provides new ideas for data acquisition and application in certain special testing environments such as industry, agriculture, and even military[4].

1 Overall design of system

When using wireless sensor network for data transmission, if the amount of data is large, the data should be optimized to avoid affecting the normal communication of other nodes, reduce the use of wireless bandwidth, and improve the performance of the system[5]. The system mainly consists of data acquisition module, LZW data compression module, data storage control module, data transmission module, ZigBee network module and host computer software. The overall design diagram of the system is shown in Fig. 1.

Received date: 2019-02-08

Corresponding author: CHENG Ya-li (937125521@qq.com)

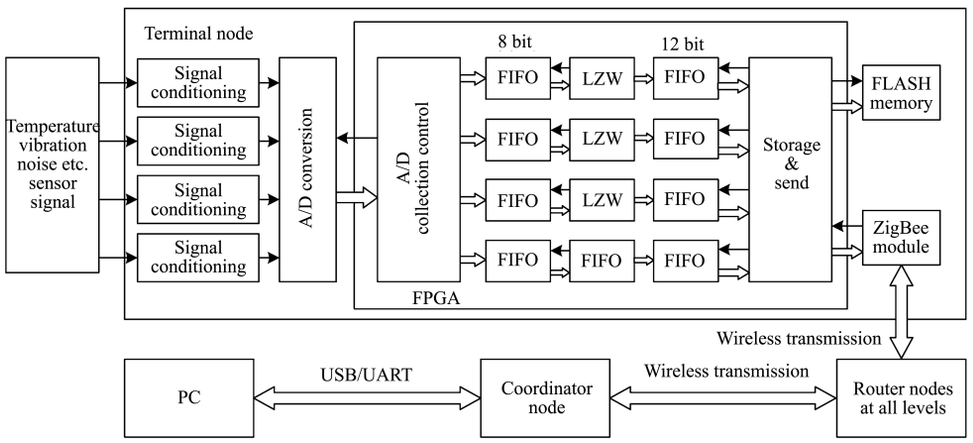

Fig. 1 General design diagram of system

To collect 4-channel analog voltage signals, the signal changes from 0 to 5 V, the signal frequency range is 0—300 Hz, and the sampling quantization precision is 8 bits. The quantized data is compressed and stored in the FLASH chip for data backup. At the same time, the compressed data is sent to the upper computer through the wireless sensor network.

2 LZW algorithm

In the LZW data compression algorithm, there are three main objects: ① input data flow; ② output coding stream; ③ a string table used to encode data during the compression process, that is, a "dictionary"[6]. Input data flow refers to the original data to be compressed; The output encoding stream refers to the output encoding generated after data compression; The string table is also the index number established for the data during the data compression process. If it is the same data, only the quotation marks of the data are output, so that the data is compressed[7]. The LZW data compression coding process is shown in Fig. 2.

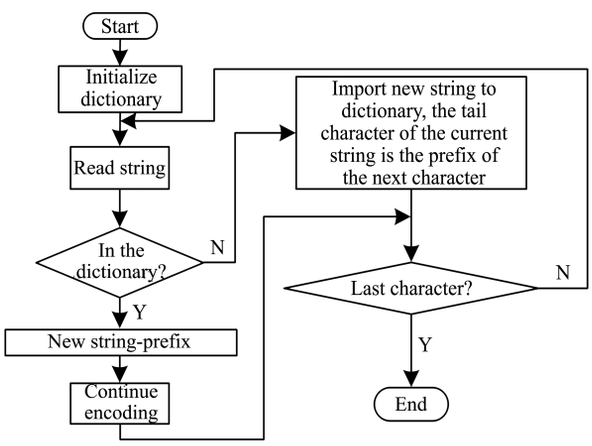

Fig. 2 Data compression process for LZW

After the data is compressed through the LZW compression algorithm, it needs to be decoded. The LZW decoding process is shown in Fig. 3.

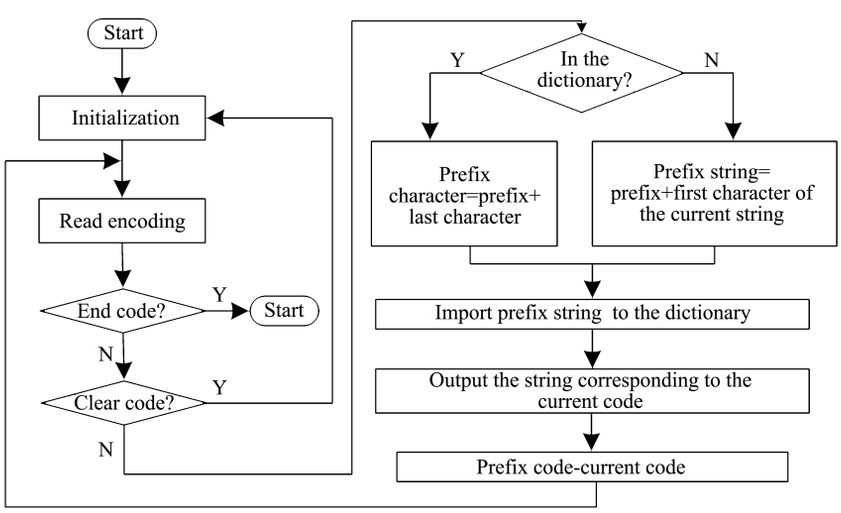

Fig. 3 LZW decoding process

3 Hardware design of wireless sensor network

With FPGA as the core chip, the main module design of the hardware system is described, and the optimization of LZW compression algorithm is introduced emphatically.

3.1 Terminal node design

The terminal node structure diagram is shown in Fig. 4.

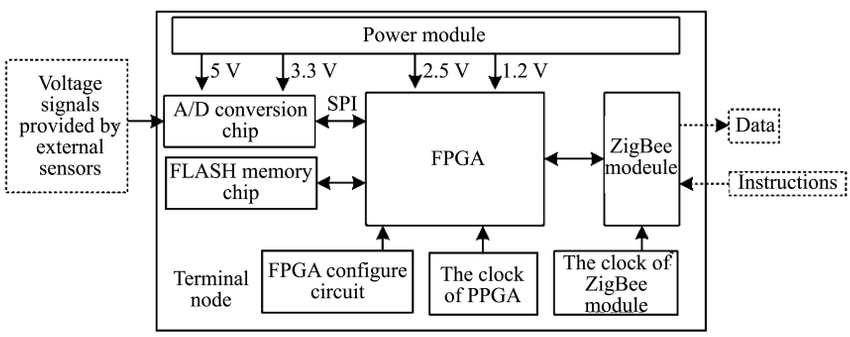

Fig. 4 Terminal node structure diagram

As shown in Fig. 4, the terminal node uses the CC2530 produced by Texas Instruments (TI) to constitute a wireless transceiver module. The module is powered by +3.3 V. The diagram of the CC2530 peripheral circuit is shown in Fig. 5.

Because the data will be sent by the FPGA through the ZigBee module, the FPGA-processed data is designed to export to the ZigBee chip through the UART0 series on CC2530. The 17th pin(P0_2 port) on the CC2530 chip is the receiving port of the ZigBee module, and the 16th pin(P0_3 port) on the CC2530 is the sending port of the ZigBee module. When designing the circuit connection, connect these two pins to the corresponding I/O port on the FPGA[8].

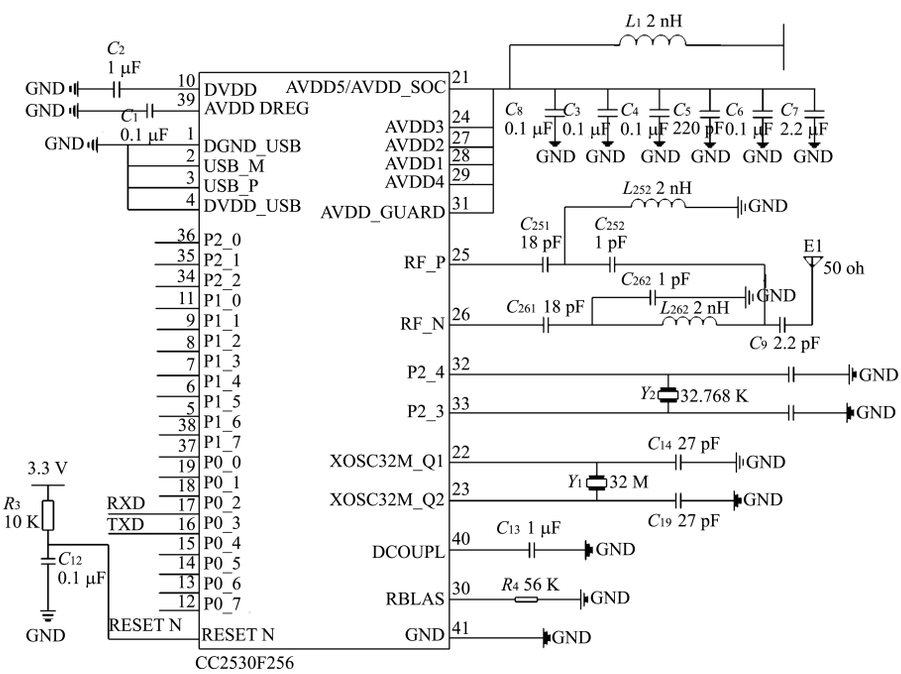

Fig. 5 CC2530 and its peripheral circuits

3.2 Coordinator node design

The main function of the coordinator node is to transfer the received data through the wireless network to the host computer. Due to the low data transmission speed of the ZigBee network and the baud rate of the transmission of 115 200 bps, serial port communication is used. The system uses the CH340T serial port/USB conversion device as a bridge to connect the coordinator and the host

computer. The CH340T chip has a separate buffer for receiving/sending data, and it supports multiple types of asynchronous serial communications. When asynchronous serial communication is performed, the format of the data is: 1 bit of starting bit (low level), 5—8 bit of data bit, 1 bit or 2 bit of stop bit (high level). In this design, the working voltage of +3.3 V is selected for the CH340T. Firstly, the +5 V voltage provided by the USB interface connected to the host computer is used, and then the voltage is converted to +3.3 V via the LM1117-3.3 chip. The CC2530 and CH340T chips are powered. The connection circuit between the CH340T, the CC2530 and USB interface is shown in Fig. 6.

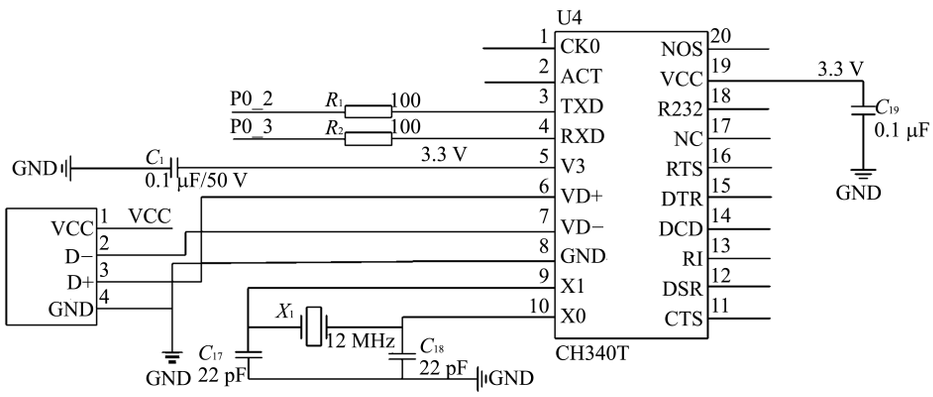

Fig. 6 Serial port/USB conversion circuit

3.3 Optimization design of LZW algorithm

Since the LZW algorithm is a dictionary based algorithm, the retrieval of the dictionary has a great impact on the compression speed of the data. In order to raise the retrieval speed, the system uses multiple hash structures to search, and establishes a hash function so that its keyword K corresponds to its storage location in the dictionary. Fig. 7 shows the dictionary retrieval process for the hash structure.

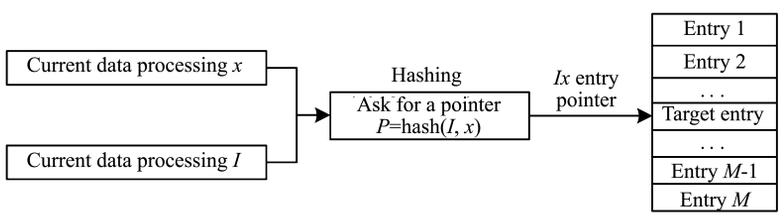

Fig. 7 Dictionary retrieval of hash structures

In the LZW data compression algorithm, when the compressed data is output, the compressed encoding of the output is actually the address of the dictionary. Therefore, after the data is processed by the LZW compression algorithm, the final encoding length is related to the size of the designed dictionary. Setting different sizes of dictionaries can affect the final compressibility. The system selects the original data with a size of 500 KB from the existing project data. Then use MATLAB software to write a computer based LZW data compression algorithm, set the dictionary size to 2, 4, 8, 16 and 32 K, and compress selected raw data. Finally, the most reasonable dictionary size is set by comparing the compressed data sizes. After experimental comparison, the relationship between the dictionary size and the compression ratio of the data is shown in Table 1.

Table 1 Relationship between dictionary size and compression ratio

Compressed data source	Dictionary size (K)	File size after compression (KB)	Compression ratio
Initial data (500 KB)	2	322.58	1.550 : 1
	4	289.18	1.729 : 1
	8	288.52	1.733 : 1
	16	286.37	1.746 : 1
	32	282.33	1.771 : 1

As seen from Table 1, the compression ratio of the original data is also increasing as the dictionary setting size continues to increase. When the dictionary size changes from 2 K to 4 K, the compression ratio of raw data grows fastest. When

the size of the dictionary gradually increases from 4 K to 32 K, the increase in the compression ratio slows down. Combining the above experimental results, the setting size of the dictionary during this compression process is 4 K, and in this dictionary, the first 256 locations are placed from 0 to 255. When the dictionary is full, it is automatically emptied.

4 System programming and functional implementation

The FPGA program is written with the VHDL language in the development environment Vivado of Xilinx. Fig. 8 shows the flow chart of the entire system.

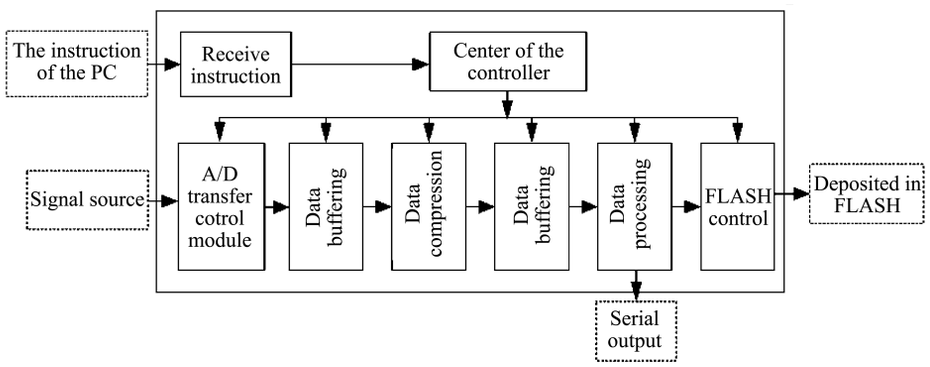

Fig. 8 FPGA internal program block diagram

4.1 LZW data compression module design

In this design, there are 4 channels of data that need to be compressed. The design diagram of the single LZW data compression algorithm is shown in Fig. 9.

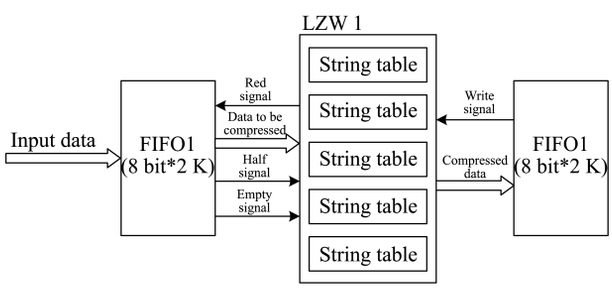

Fig. 9 LZW compression algorithm diagram

In order to prevent data confusion from causing data invalidation, a frame head and a frame tail are added to the data generated after each compression of each channel. Since the compressed data is 12 bits, the frame head and the frame tail are designed to be 12 bits. Table 2 lists the frame head and frame tail flags for the 4-channel data.

Table 2 Frame head and end flags for four channel data

Number of channels	Frame head	Frame tail
1	0x111AAA	0x111FFF
2	0x222AAA	0x222FFF
3	0x333AAA	0x333FFF
4	0x444AAA	0x444FFF

When compressing the data of each channel, a FIFO of 2 KB is set and the data in the FIFO is read when the FIFO is half full. In the compression module, if receiving the half-full signal of FIFO, firstly output the frame head "0x111AAA". Here, it should be noted that the added frame head is not compressed, but directly controls the memory of the frame head symbol to the compressed output of FIFO. After the last compression of the output data, add the uncompressed frame tail "0x111FFF", and the program control is stored in the FIFO with a compression backend width of 12 bits. Fig. 10 is the simulation result of adding the first frame head. As seen from the figure, when the half-full signal of FIFO is valid, the frame head flag "0x11AAA" in the first line is output to the back end of the FIFO during the next clock cycle.

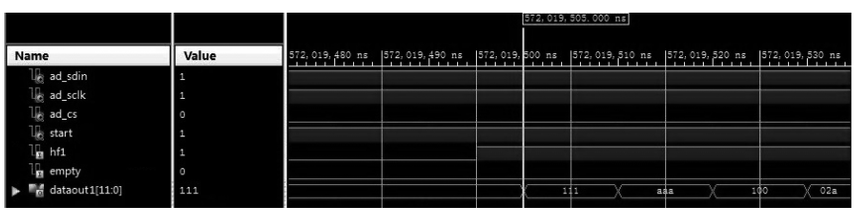

Fig. 10 Add first frame head simulation

4.2 Data transmission module design

After the compression program has processed the data, the data must be sent. Since the 4-channel data uses four compression modules, the data transmission process must be controlled, that is, the arbitration of the sent data.

From the LZW data compression algorithm, it can be seen that because the data on each channel is different, and the time for finding different characters is different, the time used by the 4-channel data in the compression process is also different. In this design, four different priorities are set. When a compression module with a channel has compressed the data, that is, when the corresponding channel FIFO is empty, the priority of the channel is set to the highest. In other words, the compressed data in this channel is sent out firstly and data detection is performed when reading data from the FIFO of 12-bit. If it is detected that the frametail sign of this channel composes of two consecutive 12-bit numbers, it indicates that the data of this channel has been temporarily sent, and then the data of the next priority channel compression is ready to sent. If one channel is in the process of sending data, another channel is also compressed, then later completed channel is set to the next priority and enters the waiting state until the high-priority channel sends the data. When the FPGA sends data to the CC2530, the baud rate is set to 115 200 bps, and the original data collected by the front tail is 32 768 bps. The compressed data will be smaller and must be sent within 1 s. Therefore, there is no redundancy in the process of sending the data of these four channels.

At the same time, it should be noted that the receiving end of the data is the UART0 serial port on the CC2530. The data format is: 1 bit of start bit, 8 bit of data bit, no parity bit, 1 bit of stop bit. When receiving the compressed data of 12-bit, two 12-bit data are used as a group, and the data are received by three times. As shown in Fig. 11, the serial port receive the high 8 bits data of the first 12-bit data firstly, then receives the low 4 bits data of the first 12-bit data and the high 4 bits data of the second 12-bits data, and finally receives the low 8 bits data of the second 12-bit data.

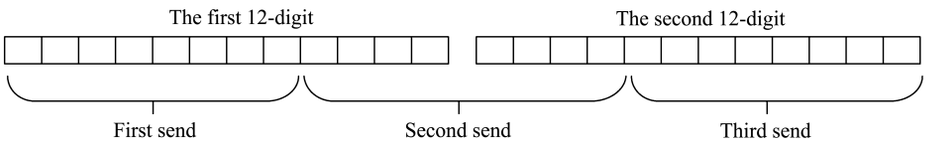

Fig. 11 Data transmission set

In terms of clock matching, since the baud rate of CC2530's serial port is set to 115,200 bps, the data sender of the FPGA is set to match the baud rates of two port. The system clock frequency is 29.491 2 MHz, so 256 frequency division is needed.

4.3 Wireless network function implementation

To realize the function of wireless sensor network, not only the function of the acquisition and compression module on the terminal node is needed, but also the development of the CC2530 chip is required to make sure it can form network successfully and realize the most basic network communication. In wireless sensor networks, three different types of network devices are installed, namely, coordinators, routers, and terminal node devices.

The coordinator is responsible for transmitting the received data to the host computer;

The router node is responsible for relaying the received data;

The terminal node device is responsible for sending the collected and compressed data.

5 Test validation

In order to verify the correctness of the real-time data compression wireless sensor network, it is necessary to conduct a comprehensive test and verify of the system.

After multiple tests, the system sets the number of dictionary hashes as 25 times and simulates it. As shown in Fig. 12, when the clock is 100 MHz, it takes 83.632 ms to compress 64 KB random data, with an average processing rate of 765.1 kB/s, which is much larger than the inflow rate of front-end data.

Multiple data collection and compression tests are performed to record the changes of the compression ratio. As shown in Table 3, the compression ratio of data is also better.

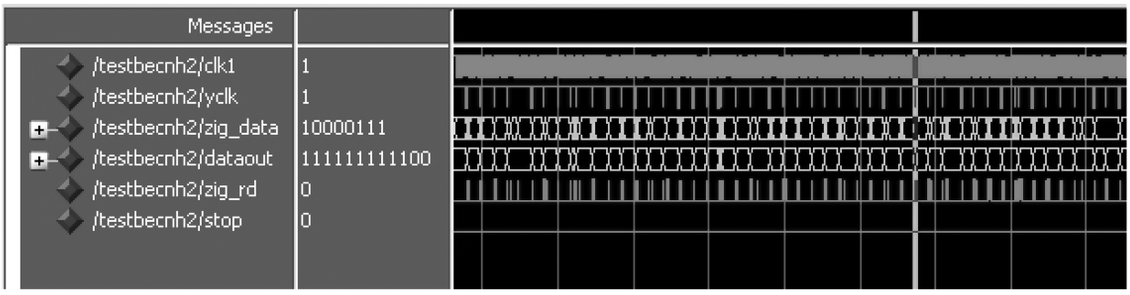

Fig. 12 Compression effect diagram

Table 3 Compression test results

Compression source	Compressed source size (KB)	Frequency range (Hz)	Compression ratio
A/D data 1	256	0—300	1.81∶1
A/D data 2	256	0—300	1.85∶1
A/D data 3	256	0—300	1.83∶1

Chipscope is used to test and verify the A/D collection function on the terminal node. The results are shown in Fig. 13. It can be seen that in the FIFO of caching the first and second channel data, the data is increasing, which is the same as the trend of sine waves. There is no data in the third and fourth channels.

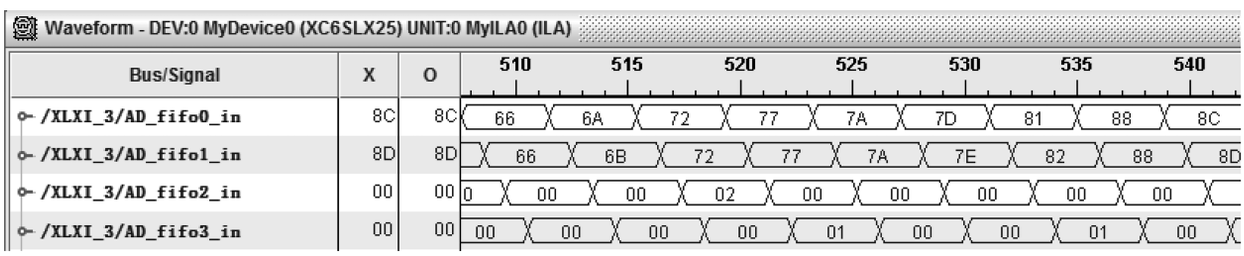

Fig. 13 A/D capture function test

LZW data compression module is the key of data processing in this design, which is related to the correctness and integrity of data. Fig. 14 shows the test of the LZW data compression module by Chipscope. As seen from the figure, the compression module has compressed the collected channel data and output 12-bits data(hexadecimal display), and due to the different time taken to consult the dictionary during the compression process, the data output time of each compression module is not uniform.

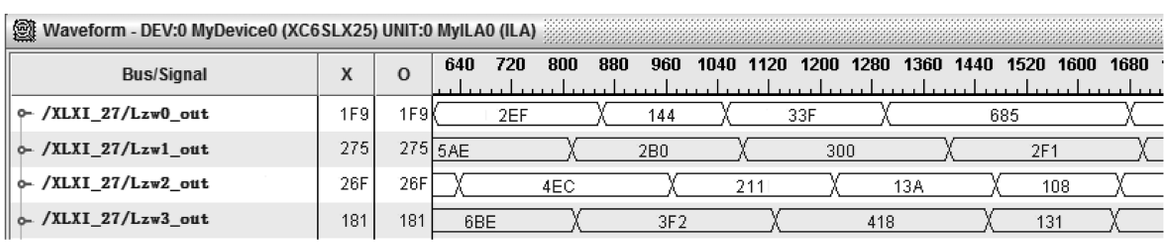

Fig. 14 LZW compression module test

When directly observing the Chipscope snapshot of the LZW data compression module, it is difficult to judge the accuracy of the compressed data. Therefore, the data received by the host computer is saved into the text file. Then analyze and verify the data.

Fig. 15 shows the data received by the host computer, from which the frame head and frame tail of the compressed data of each channel can be found.

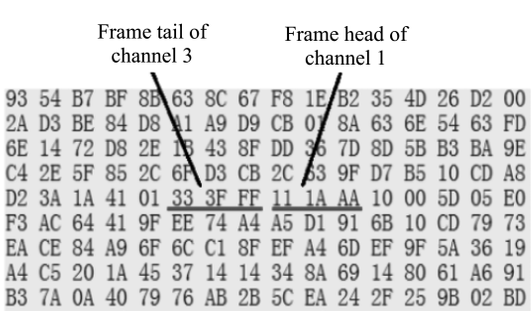

Fig. 15 Frame head and frame tail for receiving data

The data is subcontracted and decoded by using the upper computer, and the channel voltage waveforms are finally drawn using the tracing method, as shown in Fig. 16. It can be seen that the waveforms drawn by the first channel and the second channel are consistent with the data waveforms collected by A/D, and the third and fourth channels are low-level. From the test results, The validity and accuracy of the real-time data compression wireless sensor network are verified.

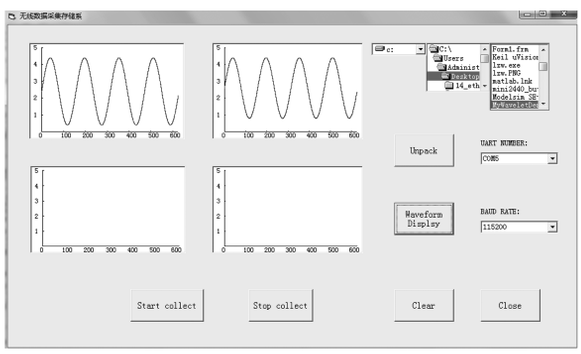

Fig. 16 System test

6 Conclusion

Combining the advantages of LZW algorithm for real-time lossless compression of data, a wireless transmission system is designed. Through several tests, it is shown that the system can achieve real-time compression with the compression ratio of more than 1.81 : 1. And it is proved that the real-time data compression wireless sensor network has good performance. After the actual test, the system is stable and has a certain practical value.

References

[1] Jia W. Research on data acquisition and information transfer technology of the internet of things. Modern Electronic Technology, 2016, 39(5): 33-37.

[2] Tian H F, Yuan M. Design of intelligent environmental monitoring system based on wireless sensor network. Measurement and Control Technology, 2014, 33(9): 36-39.

[3] Pi W J. Research on wireless sensor network positioning algorithm. Jilin: Jilin University, 2013.

[4] Leng J. Design and implementation of the Internet of things gateway platform based on M2M. Xi'an: Xidian University, 2014.

[5] Shi Q Q. Self-positioning system of wireless sensor network nodes and its algorithm research. Shanghai: Shanghai Jiaotong University, 2009.

[6] Liu W J, Fan J X, Li C S, et al. Intelligent transportation system design based on ZigBee wireless sensor network. Journal of Sensor Technology, 2013, 26(12): 1747-1751.

[7] Fan Y, Yu Y, Li Y Y, et al. Remote monitoring system based on ZigBee wireless sensor network. Laboratory Research and Exploration, 2016, 35(1): 80-84.

[8] Duan S M. Study on TDMA time slot allocation in wireless sensor networks. Kunming: Kunming University of Technology, 2010.

基于 LZW 算法的实时数据压缩无线传感器网络的设计

成雅丽，李锦明，成乃朋

（中北大学 仪器与电子学院，山西 太原 030051）

摘 要： 针对利用 ZigBee 进行远距离无线通信时，终端节点的数据量逐渐增大的特点，系统设计了一种基于 LZW 算法的实时数据压缩无线传感器网络。系统由终端节点、路由器、协调器和上位机构成。终端节点负责将采集到的数据经过 LZW 压缩算法压缩后，对数据进行存储并且发送；路由器负责无线网络中数据的中继；协调器负责将接收到的数据发送至上位机。在网络功能实现方面，利用 Z-stack 协议栈完成了终端节点、路由器节点以及协调器节点上 CC2530 芯片的开发与配置，并成功组网。经过最终的仿真分析与测试验证，系统实现了对远程数据的无线采集与存储，并通过对数据的压缩降低了网络占用率，具有一定的实用价值和应用前景。

关键词： 无线传感器网络；ZigBee；LZW 算法；数据压缩

引用格式： CHENG Ya-li, LI Jin-ming, CHENG Nai-peng. Design of real-time data compression wireless sensor network based on LZW algorithm. Journal of Measurement Science and Instrumentation, 2019, 10(2): 191-198. [doi: 10.3969/j.issn.1674-8042.2019.02.012]