www.s2is.org
International Journal on Smart Sensing and Intelligent Systems
ISSN 1178-5608

A WSN-BASED ON-LINE WORKING CONDITION MONITORING SYSTEM FOR LARGE ELECTRICAL EQUIPMENT

Xuejun Chen [1,2], Yongming Yang [1]

[1] State Key Laboratory of Power Transmission Equipment & System Security and New Technology

Chongqing University, Shapingba Zone

Chongqing, China

[2] Department of Electronic Engineering

Putian University, Chenxiang Zone

Putian, China

Emails: cxjnet@126.com; yangyym@cqu.edu.cn

Submitted: Oct. 8, 2012 ***Accepted: Jan. 21, 2013*** ***Published: Feb. 20, 2013***

Abstract- To solve the problems of traditional wired on-line monitoring system which has lines too much, cost too high, fault diagnosing and maintaining difficulties and so on, an on-line working condition monitoring system for large electrical equipment based on wireless sensor network (WSN) is proposed, designed and implemented. CC2431 chips were used for hardware design of wireless sensor network node and base station, and the TinyOS transplanted into the sensor nodes and base stations

are discussed in detail. Then, a management software system based on LabVIEW and database programming technique is proposed and implemented. The experiment results show that the system can satisfy the needs for real-time data gathering, data storage, data curve drawing, low power consumption, wide coverage, no region limitation. It's a good operation performance.

Index terms*: WSN, CC2431, TinyOS, LabVIEW, SQL, On-line Monitoring.*

I. INTRODUCTION

On-line working condition monitoring of large electrical equipment is an important means to prevent electric power system failures. However, the communications of large electrical equipment on-line monitoring is wired at present [1-4], which caused the wiring complexity, vulnerability increased, maintenance and mend hard. To overcome the restrictions of wired sensor networks, a wireless sensor network for electrical equipment on line monitoring was developed in the paper.

Wireless sensor network (WSN) is a new control network technology, integrated sensor technology, embedded computer technology, modern network technology, wireless communication technology and distributed intelligent information processing technology [5-9]. The proposed network consists of the same or different functional intelligent wireless sensors. It is an intelligent network application system, which can implement data acquisition, data fusion and data transmission. Its self-organization, adaptability and flexibility topology can greatly enhance reliability of on-line monitoring system.

II. SYSTEM DESIGEN

The framework of the system is shown in Figure 1. In large electrical equipment such as transformers, WSN nodes were installed on the external of their insulators and the winding. These network nodes are ultraviolet pulse sensors, ultrasonic sensors, temperature sensors, humidity sensors and other sensors. When winding or insulator surface discharge, the nodes can detect the discharge signals and executive signal quantization, simple processing, and the IEEE802.15.4 [10] standard package framing, and then the data frames will be transmitted to the nearby nodes, or these nodes receive data frames from other nodes, and adding multi-hop information, package framing, and then the new data frames are transmitted. Until the data frames were transmitted to the base station, base station will directly upload the receiving data to computer.

In the network, all nodes' information uploaded to the computer will be done complex digital processing, and will be integrated managed by database system. Moreover, these data can be transmitted to remote host through Internet, and then the system can achieve the remote fault diagnosis.

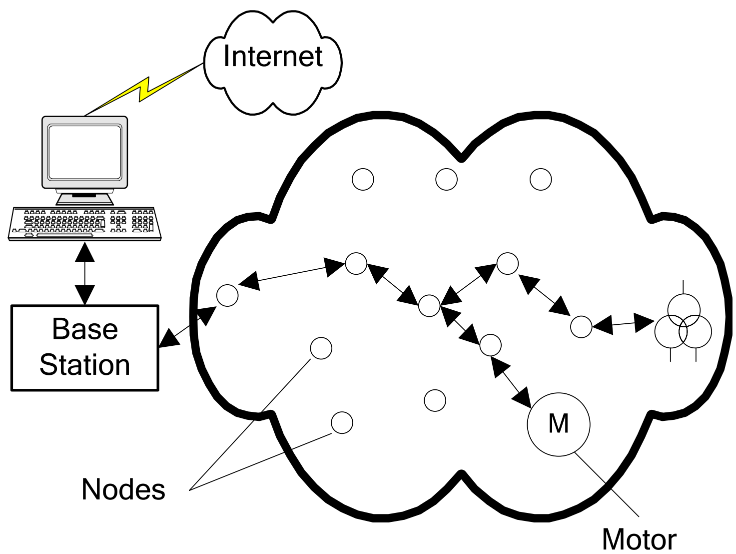


Figure 1. System framework

III. NODES AND BASE STATION DESIGN

A single node is a typical resource-constrained embedded system. It needs many functions like context awareness, data transmission and processing capabilities, receiving and executing instructions, and other network functions. Therefore, every node needs a small operating system to organize and manage the hardware, software and complete its function. TinyOS is a small operating system for wireless sensor networks which was developed by the University of California, Berkeley. However, but the core of platforms which TinyOS supports is separated with the radio frequency module, and then it must affect the nodes' energy saving. Therefore, the hardware of nodes and base station use CC2431 System-on-Chip as the processing core in this system. Because its high integration level can overcome the above shortcomings, and the node size is greatly decreased. It coupled a PCB antenna, so the system is further enhanced power conservation.

a. Hardware design

The CC2431 is highly suited for systems where ultra low power consumption is required. This is achieved by various operating modes [11]. Short transition times between these modes further ensure low power consumption. The hardware framework is shown in Figure 2. In this study, the main circuits of hardware include the power supply circuit, CC2431 external circuit, sensors circuit, flash ROM, serial interface and three light-emitting diodes. The power supply circuit consists of three 1.2 V batteries. Some filter capacitors was employed to suppress the major harmonic interference at power input. And then the power input was divided into digital and analog power supply. The digital power supply is directly connected to the power input, and the analog power supply is connected to the power input through a ferrite bead. The platform which is developed by our laboratory [12-13].

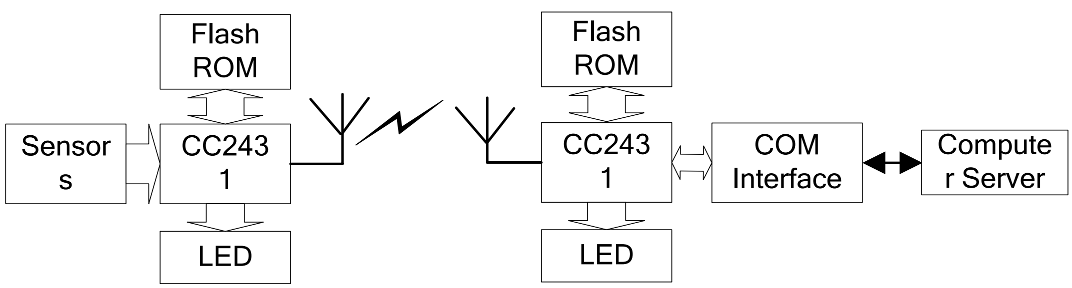


Figure 2. Hardware framework

b. Software design

Software design mainly discussed about TinyOS transplantation. Since the transplant work includes many components, the paper first discussed a typical example of radio frequency components transplantation. TinyOS transplantation in the 8051 core is different from TinyOS transplantation in AVR/MSP430/PIC/ARM core, because nesC compiler bases on the GCC (GNU Compiler Collection), but GCC does not support cross-compile for the series of MCS-51 core [12]. The whole process of this transplantation is shown in Figure 3. We illustrate the transplantation process by a typical example of radio frequency components transplantation as follows.

Write Hardware Presentation Layer and Hardware Abstraction Layer: The wiring framework of the radio frequency configuration file CC2430RadioC is in Figure 4. This file is a hardware abstraction layer file. It wires the lower level layer files CC2430RadioM、CC2430ControlM、CC2430TimerM、HPLCC2430RFInterruptM and LedsC to form the radio frequency module. This configuration provides radio frequency functions to higher level layer by the interfaces StdControl、BareSendMsg、ReceiveMsg、CC2430Control、SplitControl、MacControl、RadioCoordinator. According to the different application we can add other interfaces, but required to revise the configuration and the related modules.

The functions of every module in radio frequency configuration [11-15]:

CC2430RadioM is a hardware abstraction layer file, and most of radio frequency functions were realized in this module component. It realize data messages sending and receiving tasks, including data framing and various MAC protocols such as CSMA / CA, MAC layer conversation and so on.

CC2430ControlM is a hardware presentation layer file. It present the radio frequency controls to various functions for CC2430RadioM calling.

HPLCC2430RFInterruptM is a hardware presentation layer file, mainly dealing with the services of radio frequency-related interruption, and providing event signals for CC2430ControlM and CC2430RadioM.

CC2430TimerM is a hardware presentation layer file, which provides the MAC control clock for radio frequency component, for example, counting the frame transmitting time by capturing the start of frame delimiter signal.

LedsC module is hardware abstraction layer file. It provides radio frequency component running instructions by three light-emitting diodes.

1) NesC Compile for TinyOS: When we complete writing the hardware presentation layer files and hardware abstraction layer files, we combine with an application to compile into app.c by nesC compiler. Before compile, we must add the platform parameter into the file “\apps\makerules” and set up a platform configuration file called “.platform” in the directory of new platform.

We add the new platform name “CC2431” after the code “PLATFORMS =” in the file “makerules”.We create a new file “.platform” in the directory of CC2431 platform. Here is the content of the file:

@opts = ("-fnesc-target=keil", "-fnesc-no-debug");

Note that nesC compiler can not identify the IAR option, so we set the option by keil. After the two steps, we can use nesC compiler to compile the application.

2) Syntax Editing Script: Syntax editing script is used to couple app.c and IAR Embedded Workbench. There are much syntax in app.c can not be identify by IAR Embedded Workbench, so we must convert those codes’ syntax into the IAR syntax. In app.c, there are thousands lines of codes. So it is very cumbersome, time-consuming and prone to error if we change them by manual. So we need a text editing script to change the codes automatically. This syntax editing script is created by perl (Practical Extraction Report Language).

Perl has many applications in the Internet programming. It has simple syntax, wide use range, cross-platform, object-oriented, embedded files, C / C++ Links and easy debugging features. We utilize the powerful text editing function of Perl [15].

The change of codes:

Add “//” before the codes “typedef int sfr;” and “typedef int sbit;”DBG debugging codes, line command codes and other useless codes for IAR Embedded Workbench.

Change register definition codes like “sfr P0 __attribute ((x80));” into “SFR (P0, 0x80)”.

Change the codes accessing to external data storage “((unsigned char *) 0) [addr]” into “((unsigned char volatile __xdata *) 0) [addr]”.

Change the codes calculating the data offset like “(size_t) & ((struct TOS_Msg *) 0) ->data” into“(size_t) (unsigned long) & ((struct TOS_Msg *) 0) ->data”.

Wipe off the 64-bit “long long” type, and change all “long long” type data to 32-bit “long” type.

Change the beginning codes of interrupt service routine like “void __attribute ((interrupt)) __vector_16 (void)”or “void __attribute ((signal)) __vector_16 (void)” into “#pragma vector=RF_VECTOR__interrupt void RF_IRQ (void)”

After the script we convert app.c to IARapp.c, and then we can debug and program IARapp.c in IAR Embedded Workbench.

3) Debugging And Programming: We can debug and program directly in the Integrated Development Environment IAR Embedded Workbench for 8051. This step refers to Reference [16].

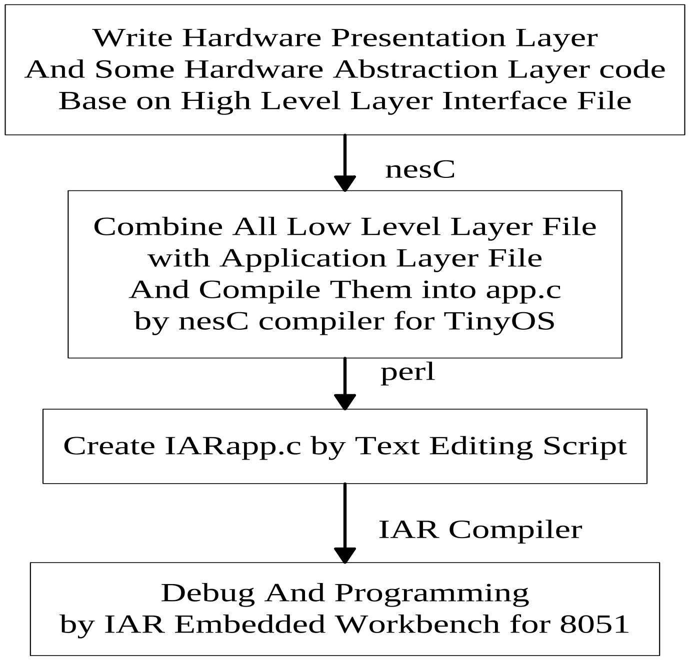


Figure 3. Transplantation process

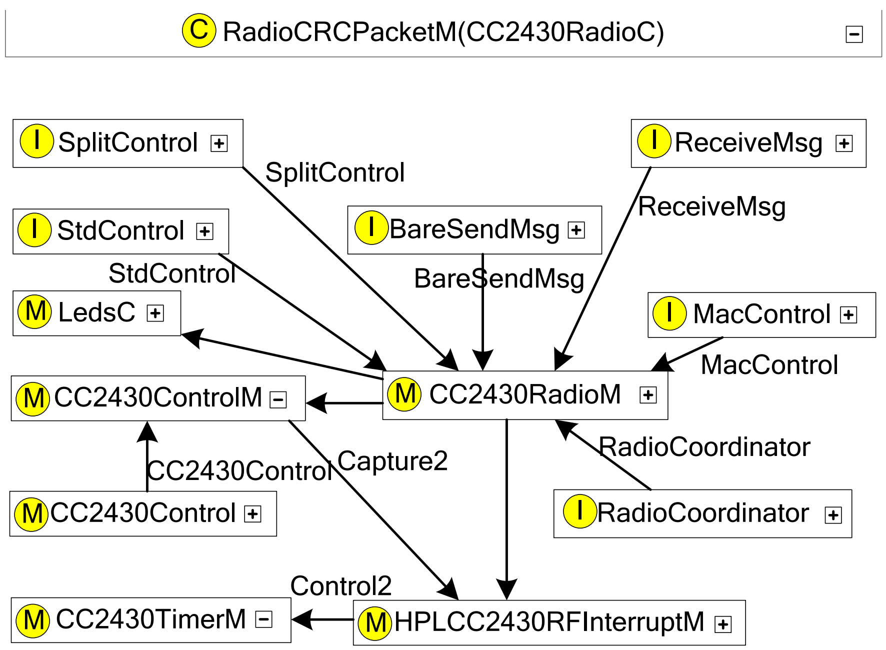


Figure 4. Radio frequency configuration framework

c. Network Model and TDMA-based protocol

In factories or power plants, there are more than one motors or electronic equipments being working. So, they need multiple nodes, which have to build an effective and synchronized wireless networks. We designed a network model which had N nodes distributed in a given area uniformity [17], and have the attributes of:

I) The sink node is the base station which have unlimited energy and always plays a cluster-head role. Other nodes have equal status and have the same capability.

II) Global time synchronization.

III) After all nodes are disposed, they won't be moved any more, and won't be intervened by human.

IV) The whole net uses a single-physical-channel.

Time synchronization is indispensable for TDMA-based protocol. And global time synchronization is assumed so that all neighbor-clusters could share the common TDMA schedule. There are already many time synchronization protocols supporting global time synchronization. The TPSN protocol [18] is used in this paper. The other three items are typical attributes of common WSNs.

The algorithm topology of TDMA-based protocol for avoiding inter-cluster interference is shown in Figure 5, the clusters CH1 and CH3 are the neighbor-cluster of CH2. CM11 and CM12 are members of CH1. CH2's member CM21 is placed in the overlapped area between CH2 and CH3, but the cluster-head CH1 can not hear it. When the cluster-head CH2 distribute TDMA slot number to CM21, it has to avoid the slots of other members in the cluster and the slots which neighbor-cluster CH3 has dominated and the slots of the cluster CH1's members CM11, CM12. Therefore, CH2 creating the TDMA schedule must get the information about which neighbor-clusters' area the members are placed, and these neighbor-clusters' TDMA schedules, for instance, CM21 is placed in CH3; and the slot numbers of the neighbor-cluster members placed in CH2's area, for example, nodes CM11 and CM12.

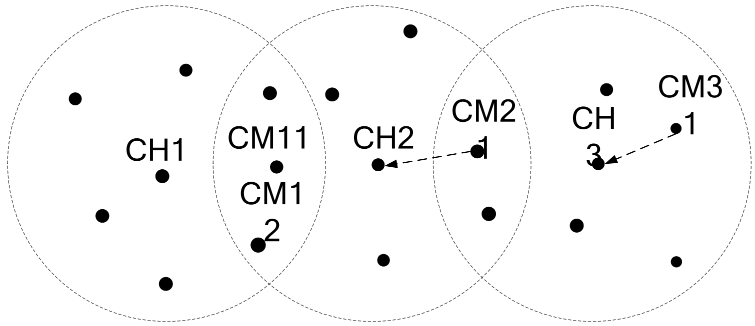


Figure 5. Inter-cluster interference

The basic idea of algorithm in this study is that, at the beginning of cluster organizing, each cluster-head keeps listening to the packets of every neighbor-cluster nodes, and constructs a neighbor-cluster member table and then feedback the information of neighbor-cluster members placed in this cluster area to their neighbor-cluster-heads. Consequently, when the cluster-heads construct the TDMA schedule, they can avoid the neighbor nodes' slots, since they have got the neighbor-nodes' information and known the situation of overlapping between local members and neighboring cluster-heads. When network turn into TDMA steady-state phase, cluster-heads exchange data with each other and upload sensing data to sink nodes both by multi-hop algorithm.

IV. SYSTEM MANAGEMENT SOFTWARE DESIGN

In order to make the above-mentioned WSN be really used in electrical equipment on-line monitoring, management software is needed to monitor and manage the data among the sensor nodes, and then to realize the function of predicting the failure.

With the development of the computer industry, software techniques have become increasingly powerful, and virtual instrumentation (VI) technology is one of them. The VI is designed using the graphical programming language LabVIEW and is capable of performing on-line measurement functions, including data acquisition, display, and analyses in the time and frequency domains, as well as data archiving[19-20]. Therefore, the virtual instrument technology was used to develop the system. The flow chart of system software is shown in Figure 6. LabVIEW and LabSQL were used to realize the system immediate communication and data management.

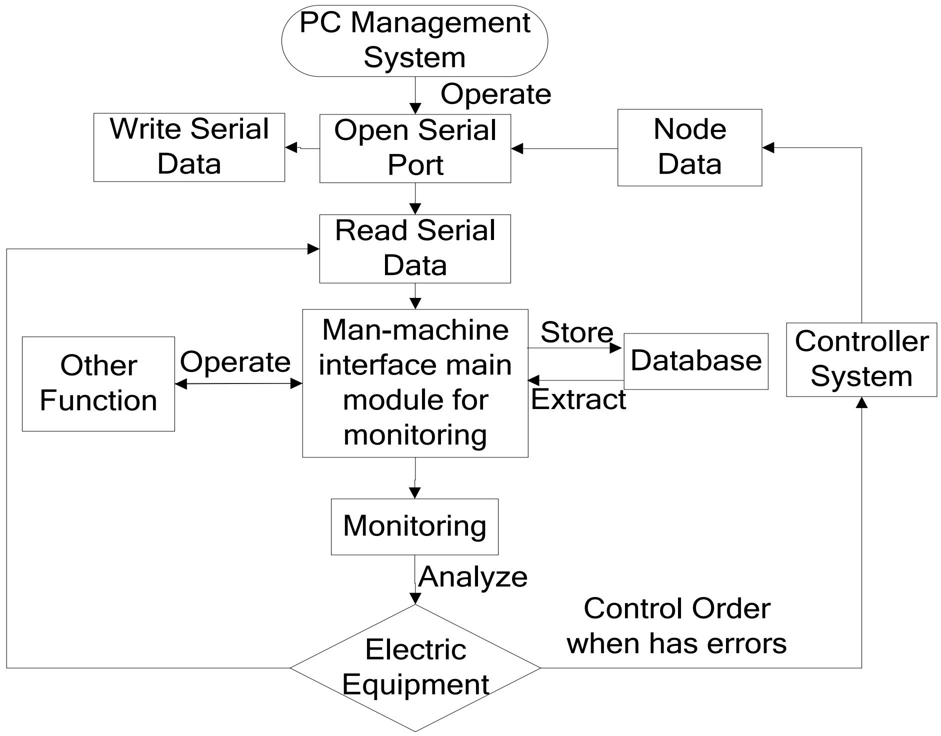


Figure 6. Flow chart of the system

Firstly the acquired data were read through serial port, and then were analyzed and extracted data information through multithreaded applications. Simultaneously, they were saved to the real-time database. Finally, waveform module of LabVIEW was used to draw and display a relative curve.

a. Multithreading Programming for serial communication

For single-threaded applications, screen updates are often slow related to other operations such as continuous high-speed data acquisition. Especially, a system attempts to acquire very large amounts of data at high speed in a single-threaded application and display all of that data in a graph on the front panel, the data buffer might overflow because the processor is forced to spend too much time on the screen update and data is lost. However, in a multithreaded application, the data acquisition can reside on a different, higher priority thread than the user interface. The acquisition can run continuously and send data into memory without interruption while the data is displayed as quickly as possible. The data acquisition thread will preempt the display thread so the system will not lose data.

Because monitoring the electrical equipment on-line monitoring system is required long time operation, many data acquisitions and high communication stability, we choose this way to communicate. Figure 7 shows the multithreaded application.

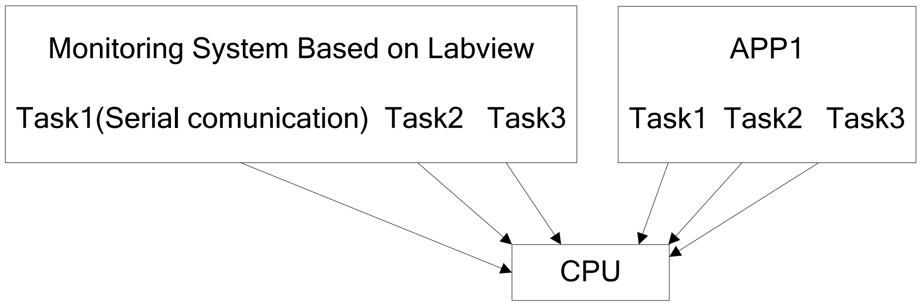


Figure 7. Multithreaded application

When the serial communication task acquires the monitoring data from the base station, these data can't be used except being analyzed. Therefore, we must extract the information from the acquisition data through some programming, and transform them into useful information for decision-making. Then the analysis and extracted data was given to the global variables. And global variables were used to access and pass data among several VIs task.

b. Real-time database

In order to conveniently search and manage the sensor data, realize the electrical equipment on-line monitoring and fault diagnosis, we need to design the real-time database to store the information of the collected sensor data. However, Labview itself has not database visited function and it will need other assistant method to carry on database visiting based on LabVIEW programming. LabSQL is a free database access toolkit of LabVIEW. LabSQL is a collection of VIs that use the ADO(ActiveX Data Objects) object collection in LabVIEW so that user can connect to almost any database, perform SQL queries, manipulate records, etc[19-21]. Therefore, LabSQL can not only access the SQLServer database can also access the Microsoft Access database. The system uses LabSQL to access the Microsoft Access database under LabVIEW. The real-time database system structure is shown in Figure 8.

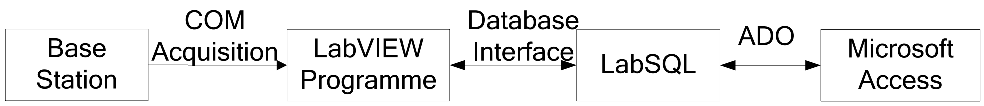


Figure 8. Real-time Database System

The database which the system is connecting to should have an ODBC (Open Database Connectivity) driver. Before using LabSQL, first is to create a DSN (data source name) in the ODBC data source of Windows operating system. LabSQL connection between the databases is built based on the DSN.

Use flow:

1) In the Windows operating system Control Panel, select "ODBC", and click "Add ..." button of "System DSN" tab in the pop-up ODBC Data Source Administrator.

2) Then the "Create Data Source" dialog box would come, select the driver "Microsoft Access Driver (*. mdb)", and click "Finish" button.

3) Create a "DSN" name in the resulting "ODBC Microsoft Access installed" dialog box, and select the database which would be accessed by LabSQL or create a new database in the "Database" column, and finally click "OK" button.

After the connector driver has been configured, LabSQL can use the DSN to access the associated database.

c. Man-machine interface for monitoring system

The monitoring system software developed by Labview is to implement data collection, processing, display, storage and other functions, as shown in Figure 9. The figure shows the human-machine interface functional structure of monitoring system software modules in the local monitoring computer. The system software is composed of a man-machine interface main module for monitoring, data acquisition module, remote data communication module, database management module, data analysis and processing module, real-time monitoring modules, parameter setting module, historical trends module and event recording module.

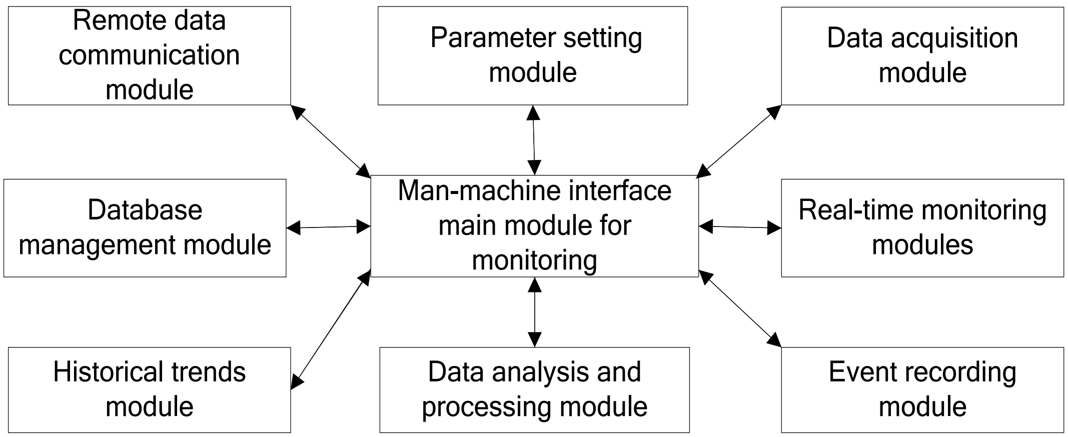


Figure 9. Software architecture diagram of monitoring system

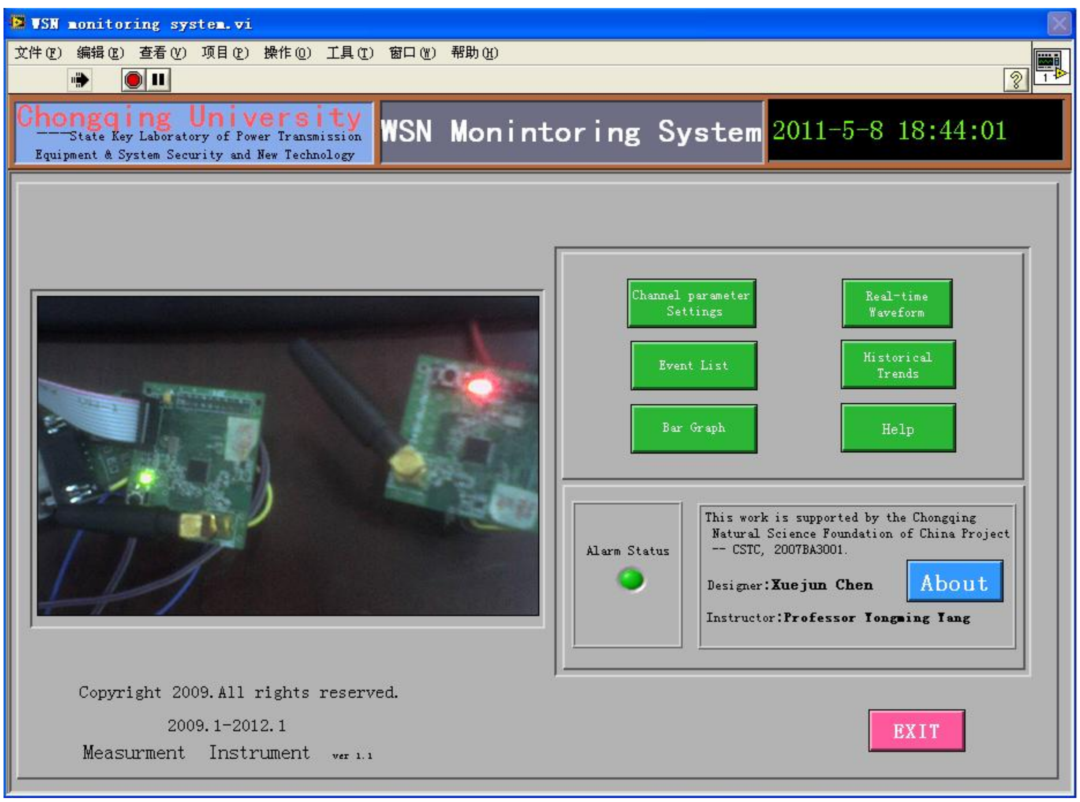


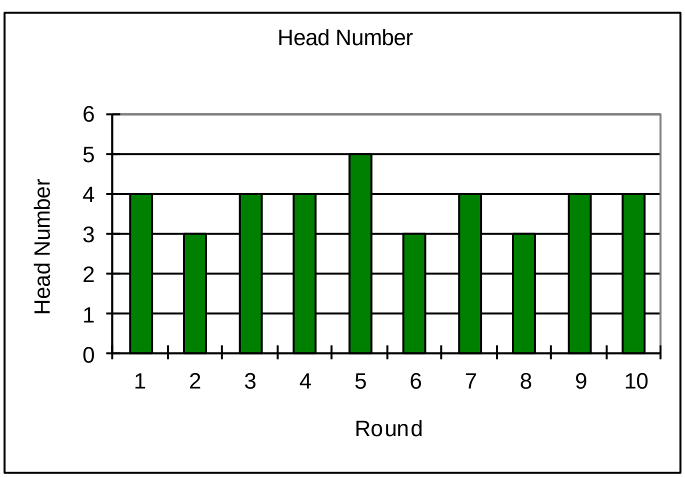
Figure 10. Interface diagram of monitoring system

Among them, data acquisition module is based on serial communication protocols to acquire data from WSN base stations; remote data communication module make the local computer send data to remote monitoring computer based on DataSocket technology; database management module is responsible for analysis and processing of data and event data storage, querying; data analysis and processing module is mainly directed against a variety of state signal digital filtering, spectral analysis, comparison the setting alarm value; Real-time monitoring module shows partial discharge, vibration, temperature, speed, pressure signal, that were processed in data analysis and processing module, which are normal or not to be expressed in the form of red, green, etc. (Red light says abnormal, Green says normal);parameter setting module is used to set the alarm parameters for various monitoring of state variables; event logging module is used to query the amount of each state signal alarm incident. The ultimate interface of the system is shown in Figure 10.

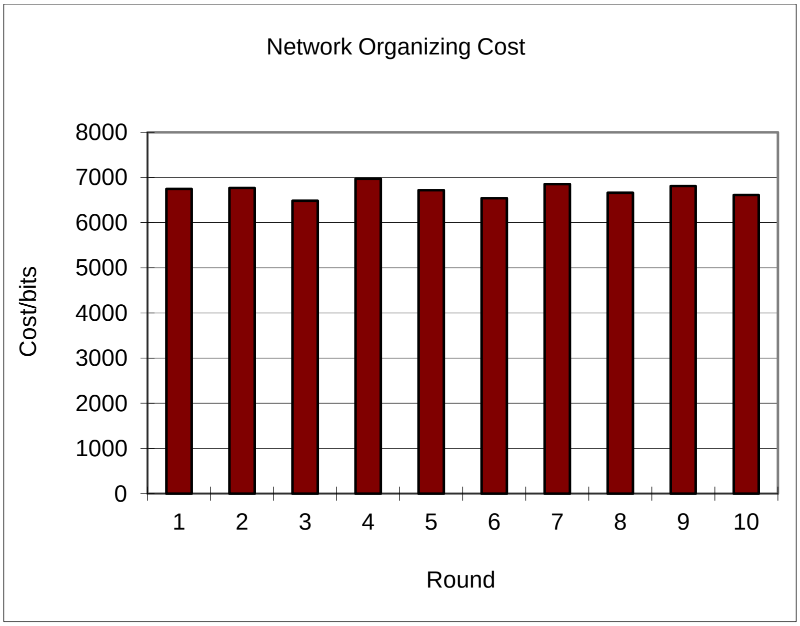
d. System experiment

1) Test of Network Model and TDMA-based protocol

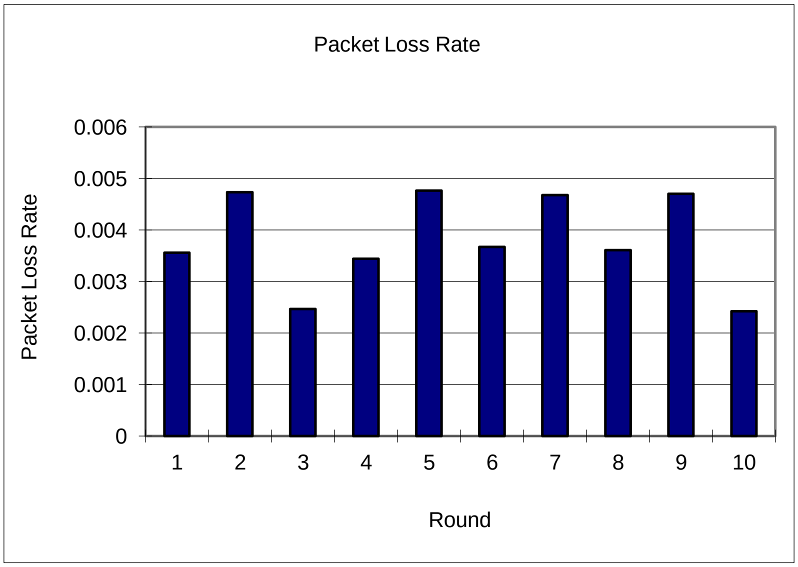
Before using the proposed system to monitor working condition for motor, the proposed TDMA-based protocol for avoiding inter-cluster interference must be tested. The test is under the circumstance of a 25 m^2 room equipped with 32 nodes, among which node 0 is a sink node, node 31 is a generic base station only for intercepting nodes' packets of network messages, others are common nodes. The sink uploads the sensing data to the upper computer, whose ID is 0, while generic base station intercepts all messages of the network and uploads them to computer, whose ID is 31, and the others' ID are in 1~30. Nodes 0~30 are programmed into the application of the protocol in this paper, while node 31 is programmed into the TinyOS standard application Generic Base. Table 1 shows the parameter sets in the experiment [22]. The results are showed in Figure 11(a),(b),(c) and Table 2.



(a) The head number of every round



(b) The cost of networking



(c) The packet loss rate of networking

Figure 11. Test results of TDMA-based protocol

Table 1 Test parameter

Test rounds	Time of every round (min)	Slot time (ms)	Slot amount	Probability of cluster-head election P (%)	Length of cluster member information list (member)	Length of neighbor-cluster information list (cluster)	Length of cluster-head information list (head)
10	10	24	256	12.5	10	10	8

Table 2 The average of test results

The average of the head	The average of the networking cost	The average of the networking packet loss rate	The average of the sensing data transmission accuracy

numbers	(bit)	(%)	(%)
3.8	6715.1	0.3803	99.6197

The cluster-heads' number were shown in Figure 11 (a) in the 10 round experiments. At the beginning of the cluster establishment of every round, each node implemented the cluster-election program, and decided to play member role or head role. The algorithm deterministic cluster-head selection (DCHS) [23] was used in the election program in this paper, which was improved from LEACH algorithm [22]. Each node created a random number between 0~1. If this number was less than a threshold, this node would elect itself to be the cluster-head. Parameter P was defined as the probability to be elected as cluster-head. Then the nodes which have been elected as cluster-heads in one of the lately 1/P rounds would not be elected cluster-heads. And thresholds of the others which were not being cluster-head would augment in every round. Then the probability P that the random number is less than the threshold would rise. The results showed that the sink node was always elected as cluster-head; the other nodes' probability P of electing cluster-head was actually 12.5%, and the remaining nodes joined the clusters evenly as cluster members.

Figure 11 (b) showed the networking costs which ranged from 6486 bits to 6973 bits. In ad hoc networks, the network was formed without any predetermined topology or shape. Therefore, any node wishing to communicate with other nodes should generate more packets than its data packets. These extra packets were generally called networking cost. Route discovery packets and route responsed packets in typical ad hoc network routing protocols were a few examples of the overhead. As the size of the network grew, more control packets would be needed to find and keep the routing paths. The network cost as the number of messages sent at the MAC layer—being communication the most prominent source of energy drain in WSNs, this measure could be considered as proportional to the system lifetime. When cluster-head number was 4 to 6 (includes the sink node), networking cost did not relate to the cluster-head number very much. However, the cluster-head was always in listening state, and consumes much energy, thus under the circumstance that the nodes' number were the same, the less the cluster-head, the longer the network lifetime would be.

The packet loss rates of the cluster establishment phase were shown in Figure 11 (c). Two to four packets is lost in every round because of the collision. But the networking still goes on successfully. The data packets from the generic base station show the fault-toleration is working. After the networking is completed, the accuracy of sensing data transmission is 99.6197% in the

TDMA steady-by phase. The average of test results is showed in Table 2. This shows that the TDMA protocol avoiding the inter-cluster interference has a good effect on the correct transmission of data.

2) System experiment

We took the system to do the experiment on a three-phase asynchronous motor. The type, power and voltage of the three-phase asynchronous motor respectively are YKK560-8, 630kW, and 6000V. The real-time temperature curve is shown in Figure 12, which is one sensor of one phase of motor windings temperature during operation. Figure 12 show that, motor sets in a normal working condition, the temperatures of P2 sensor of V-phase windings are lower than the alarm value of 115oC (the motor windings insulation for the F class) [24]. Experiments show that the acquisition module can display change values of working conditions parameters in real-time based on wireless sensor networks.

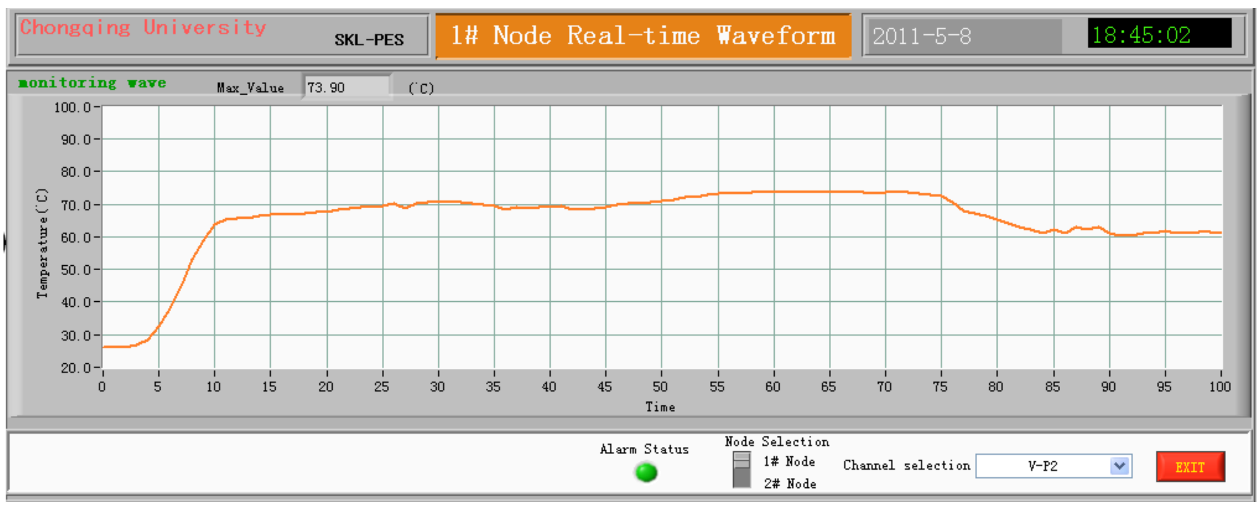


Figure 12. The real-time temperature curve

VI. CONCLUSIONS

The paper presents a new WSN, and then management software of on-line working condition monitoring system was developed for the large electrical equipment. The system has been successfully applied to do experiment on condition monitoring of a motor.

We will take full advantage of the CC2431 node location engine to add localization procedures for monitoring network node for future work. According to nodes working in different environment, we will add solar power generation and thermal power generation to enable power management of wireless sensor more flexible, and enhance life cycle of WSN.

ACKNOWLEDGEMENTS

This work is supported by State Key Laboratory of Power Transmission Equipment&System Security and New Technology of China (Project – 2007DA10512709206). And the authors wish to thank Chongqing Electric Machine Federation Co., Ltd for their support.

REFERENCES

[1] H. C. Lin, "Intelligent Neural Network-Based Fast Power System Harmonic Detection," IEEE Trans. on Industrial Electronics, vol. 54, no. 1, 2007, pp. 43-52.

[2] H.Su, K.T. Chong, "Induction Machine Condition Monitoring Using Neural Network Modeling," IEEE Trans. on Industrial Electronics, vol. 54, no. 1, 2007, pp. 241-249.

[3] M. Blodt, D. Bonacci, J. Regnier, M. Chabert, J. Faucher, "On-Line Monitoring of Mechanical Faults in Variable-Speed Induction Motor Drives Using the Wigner Distribution," IEEE Trans. on Industrial Electronics, vol. 55, no. 2, 2008, pp. 522-533.

[4] A.M.Gaouda et al.,"On-Line Detection and Measurement of Partial Discharge Signals in a Noisy Environment," IEEE Trans. on Dielectrics and Electrical Insulation, vol. 15, no. 4, 2008, pp.1162-1173.

[5] J. Yick, B. Mukherjee, D. Ghosal,"Wireless sensor network survey", Computer networks,vol.52,2008,pp.2292-2330.

[6] X. Cao, J. Chen, Y. Zhang, Y. Sun,"Development of an integrated wireless sensor network micro-environmental monitoring system", ISA Transactions,vol.47,no.3,2008,pp.247-255.

[7] G. Anastasia, M. Contib, M. Di Francescoa, A. Passarellab,"Energy conservation in wireless sensor networks: A survey", Ad Hoc Networks, vol.7, no.3, 2009, pp.537-568.

[8] J. Wu, S. Yuan, S. Ji, G. Zhou, Y. Wang, Z. Wang,"Multi-agent system design and evaluation for collaborative wireless sensor network in large structure health monitoring", Expert Systems with Applications,vol.37,no.3, 2010,pp.2028-2036.

[9] Y. Liu, K. Liu, M. li,"Passive diagnosis for wireless sensor networks", IEEE/ACM Transactions on Networking, vol.18, no.4, 2010, pp. 1132-1144.

[10] A. Ghose, J. Grobklags and J. Chuang. “Resilient data-centricstorage in wireless ad-hoc sensor networks”, in Porceedings of the 4th International Conference on Mobile Data Management (MDM03), pp.45-62, 2003.

[11] J. Hill, “System Architecture for Wireless Sensor Networks,” USA: University of California, Berkeley,CA, 2003.

[12] X. Zhuang, Y. Yang and W. Ding, “The wireless sensor network node design for electrical equipment on-line monitoring”, in Proc. Int. Conf. Industrial Technology, Chengdu, China, April 2008, pp. 1-4.

[13] Yongming Yang, Xuejun Chen, Xiong Zhuang,Wei Ding, “A WSN-based Large Electrical Equipment On-line Monitoring System”, in Proc. Int. Conf. Industrial Electronics and Applications, Taichung, China, June 2010, pp. 1849-1854.

[14] Yu Haibin et al, “Intelligent Wireless Sensor Networks,” Beijing, Science Publishing Company, pp. 286-292, 2006.

[15] Long Yang, Shi Po et al, “Perl5 Practical Programming,” Beijing, China, Water Power Press, 2001.

[16] IAR Systems. (December 2004). IAR Embedded Workbench IDE User Guide.

[17] X. Zhuang,Y. Yang and D. Wei, “A TDMA-based Protocol and Implementation for Avoiding Inter-Cluster Interference of Wireless Sensor Network”, in Proc. Int. Conf. Industrial Technology, Gippsland, Australia, February 2009, pp. 1-6.

[18] S. Ganeriwal, R. Kumar and Mani B. Srivastava., “Timing-sync protocol for sensor networks,” in Proc. Int. Conf. Embedded Networked Sensor Systems, Los Angeles, America, November 2003, pp. 138-149.

[19] C. Wang and R.X. Gao: “A virtual instrumentation system for integrated bearing condition monitoring”, Instrumentation and Measurement, IEEE Transactions on, Vol.49, No.2, 2000, pp. 325 – 332.

[20] National Instruments Corporation. LabVIEW Advanced: Performance & Communication Course Manual. National Instruments Corporation, USA, 2001.

[21] X. Chen, H. Tao, Y. Gao, C. Yu,"Study on the On-line Monitor System of Stator Bar of Generator",Piezoelectectrics Acoustooptics, vol.29,no.6,2007, pp.749-754.

[22] X. Zhuang,Y. Yang and D. Wei, “A TDMA-based Protocol and Implementation for Avoiding Inter-Cluster Interference of Wireless Sensor Network”, in Proc. Int. Conf. Industrial Technology, Gippsland, Australia, February 2009, pp. 1-6.

[23] Handy MJ, Haase M and Timmermann D, “Low energy adaptive clustering hierarchy with deterministic cluster-head selection”, in Proc. Int. Conf. Mobile and Wireless Communications Networks, Stockholm, Sweden, Sept. 2002, pp. 368-372.

[24] IEEE Power Engineering Society: “A IEEE standard test procedure for polyphase induction motors and generators”, Revision of IEEE Std 112-1996 (1997).