

Research Article

Study on Wireless Network Communication in Stage Hydraulic Monitoring System Based on Internet of Things

Yue Dong,¹ Hui Ren,¹ Jianghui Dong,² and Liping Wang³

¹*School of Information Engineering, Communication University of China, Beijing 100024, China*

²*School of Natural and Built Environments, University of South Australia, Adelaide, SA 5095, Australia*

³*Sansom Institute for Health Research, School of Pharmacy and Medical Sciences, University of South Australia, Adelaide, SA 5001, Australia*

Correspondence should be addressed to Yue Dong; yue_dong9@163.com and Jianghui Dong; jianghui.dong@unisa.edu.au

Received 1 December 2014; Revised 28 February 2015; Accepted 2 March 2015

Academic Editor: Filippo Cacace

Copyright © 2015 Yue Dong et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A novel stage hydraulic monitoring system based on Internet of Things (IoT) is proposed in this paper. Compared with the traditional wired system, the proposed system is a flexible working method and can save the cost. Furthermore, it has the low power consumption, high safety, and large scale network. The real-time pressure and flow data can be collected by using the nodes in ZigBee network. The fault detection and diagnosis process was used in this study, which was facilitated by measuring pressure of flow. When the monitored data exceeds the normal range, some failure may occur in the stage hydraulic system. If any failure occurs in the circuit, the maintainers can be informed immediately, which can greatly improve maintenance efficiency, ensuring the failure to be eliminated in time. Meanwhile, we can take advantage of wireless sensor network (WSN) to connect the multiple loops and then monitor the loops by using ZigBee technology, which greatly improves the efficiency of monitoring.

1. Introduction

It is well known that the lifting platform is the indispensable part in a theatre. It has many important functions, such as changing the set rapidly, meeting the process arrangement of the stage, making special atmosphere and effect, and changing the stage form according to the needs of performance. Chen and Wang studied the hydraulic control system design of the test platform, and the hydraulic control system of cutter disk/segment erector/screw conveyer loading simulation and driving experimental unit, simulation of thrust control system unit, and motion control test system of hydraulic cylinder in multi-DOF segment erector unit were discussed separately [1]. Li et al. designed a tripartite hydraulic lifting platform control system by which the position of the platform can be controlled with high precision and its speed can be synchronized perfectly [2]. Miao and Wang established the failure modules and collaborative failure simulation platform of hydraulic lifting system, and the failure simulation platform can realize the integrated failure mode and effect

analysis [3]. Lifting platform includes main stage lifting platform, orchestra pit lifting platform, and the audience lifting platform. Furthermore, in order to cooperate with the use of equipment such as ride, there also exists movable stage that moves in vertical direction. The compensation table and micro desk lamp also can be classified as lifting platform. Based on the stage hydraulic system, lifting platform plays an important role in the stage. The development of an end-to-end system model for a complex electromechanical-hydraulic system was discussed [4]. Dong et al. established a nonlinear model of a hydraulic automatic gauge control (AGC) system for fault detection and isolation (FDI) [5]. Tan and Sepehri proposed a parametric fault diagnosis (FD) methodology and showed that it provides an effective means of extracting important information to aid detection and isolation of faults for the class of highly nonlinear and dynamic hydraulic drive systems [6]. The applicability of Hilbert-Huang transform (HHT) for internal leakage detection in valve-controlled hydraulic actuators was investigated for the first time by Goharrizi and Sepehri [7]. A FD scheme uses measured

input-output signals to establish information regarding possible faults in a monitored system [8, 9].

Because the stage of the hydraulic system is relatively large, the failure is unavoidable. The causes of the failure are varied, such as stained aging of hydraulic components, all kinds of interference of impurity, oil corrosion, and instability of flow and pressure of the hydraulic loop. Once there is a failure in the stage hydraulic system, it is difficult to accurately lock the location of the failure and find out the reasons of the failure. It is also very difficult to detect the fault of the hydraulic components, which will result in low work efficiency, affecting the speed of the maintenance, and even it is likely to affect the performance of the whole system. Therefore, it is very necessary to design an effective stage hydraulic safety monitoring system so as to locate the position of the failure in time and then take measures to cut losses to a minimum.

IoT is an interconnected network, in which the objects can transfer data over the network. It is through radio frequency identification (RFID), sensors, global positioning systems, and other information sensing devices to connect anything in the world with the Internet and conduct information exchange and communication between the object and network according to the agreed protocol. In [10], a semantic data model was proposed to store and interpret IoT data, and a resource-based data accessing method (UDA-IoT) was designed to acquire and process IoT data ubiquitously to improve the accessibility to IoT data resources. Lazarescu proposed the functional design and implementation of a complete WSN platform that can be used for a range of long-term environmental monitoring IoT applications [11]. Amendola et al. conducted a survey on the state of the art of RFID for application to body-centric systems and for gathering information about the user's living environment [12]. A new method was proposed by Chi et al. to design a reconfigurable smart sensor interface for industrial WSN in IoT environment, in which complex programmable logic device (CPLD) was adopted as the core controller [13]. As an important part of IoT, WSNs are gradually expanded applications in smart home, precision agriculture, forestry monitoring, intelligent transportation, and the other areas. A novel source-initiated on-demand routing mechanism for efficient data transmission in mobile wireless sensor networks was presented by Mao and Zhu [14]. Mao et al. applied the game theory to solve the network security problem of wireless network, and then a novel intrusion detection framework for WSNs was presented [15, 16]. Briff et al. proposed a lower bound on the energy required for synchronizing nodes in a WSN by using statistical estimation techniques [17]. Lu et al. described the system architecture and design methodology of an ASCI-based sensor network device to meet some attributes for a class of applications [18]. A novel Tyndall Heterogeneous Automated Wireless Sensor (THAWS) based automated routing selection method to fully utilize the system resource and save sensor system energy was proposed by Shen et al. in [19]. Choi et al. described a Bluetooth WSN for security systems, which includes the implementation issues about system architecture, power management, self-configuration of network, and routing in [20].

Compared with other IoT technology, ZigBee has a number of advantages, for example, low power consumption, high safety performance, the maximum network size and low cost. In addition, ZigBee wireless network also has a low complexity and data rate. These features make ZigBee suitable for automatic control and remote control and can be embedded in a variety of devices. In addition to a stable two-way and multipoint communications capability, ZigBee technology also has a flexible adaptability and scalability, and therefore it does not require complex server and other equipment. Full function device and simplified function device are the only part of any ZigBee network. The network can be expanded by increasing the full function device or simplified function device, which provides great flexibility for a variety of applications. Yi et al. have thoroughly evaluated ZigBee performance under WiFi interference for smart grid applications. A theoretical model has been introduced, followed by a corresponding simulation model, which completely reflects the ZigBee and WiFi coexistence features via MATLAB or Simulink in [21]. A network repair scheme consisting of a regular repair and an instant repair scheme was proposed in [22]. If a router loses its parent, it tries instant repair to reconnect to a new parent. Ding et al. studied the efficient and simple data broadcast in IEEE 802.15.4 standard, and proposed self-pruning and forward node selection algorithms to exploit the hierarchical address space in ZigBee networks [23]. A Dual Radio ZigBee Homecare Gateway (DR-ZHG) has been proposed and implemented to support remote patient monitoring in [24].

According to the above advantages of ZigBee technology, we choose ZigBee to build a wireless network for data communication. Traditional method for monitoring the stage can be only performed in one loop, and when the other circuit fails, the monitoring system cannot find the fault in time. Using ZigBee technology can take advantage of WSN to connect multiple loops and then monitor multiple loops, greatly improving the efficiency of monitoring. By using the nodes in ZigBee network, we can collect real-time pressure and flow data. With establishing and running the stage network monitoring system, the stage management department can achieve the real-time operating status. When a fault occurs in one loop, the nearest maintenance personnel can be immediately informed. So it can greatly improve the response speed of stage operation failure, ensuring the failure excluded in time, and personnel can receive timely rescue, in order to protect the safe operation of the stage. By analyzing and counting stage log data, we can know the type and probability of the failure during the operation of stage. It helps to analyze the potential insecurity, provide direct data to support routine maintenance, improve the pertinence of maintenance work, and reduce maintenance costs.

This paper is organized as follows. In Section 2, we present the composition of the stage hydraulic monitoring system. The monitoring theories are described in Section 3. In Section 4, the design of each part of the stage hydraulic monitoring system is introduced in detail. The monitoring data and analysis are discussed in Section 5, and the conclusions are drawn in Section 6.

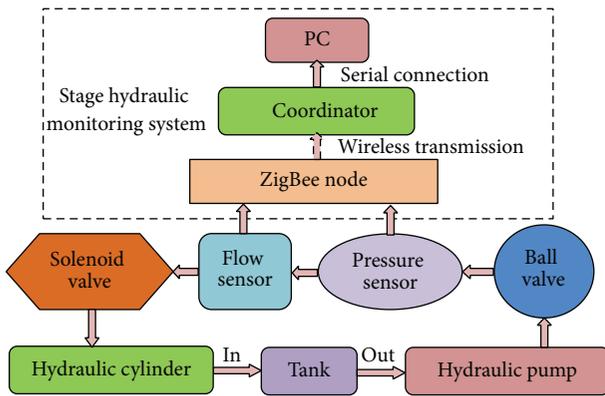


FIGURE 1: The diagram of stage hydraulic system.

2. Composition of the Stage Hydraulic Monitoring System

ZigBee network has three topology forms, namely, star topology, tree topology, and mesh topology [25]. Combined with the actual situation of the stage hydraulic system, and due to the limited conditions, we choose star network topology in this paper, which includes a coordinator node and a plurality of end nodes. Each end node can only communicate with the coordinator node, and communication between two end nodes must be forwarded by the coordinator node.

The system consists of three parts, that is, sensor node part, coordinator part, and PC part. It consists of a fully functional coordinator, multiple end nodes equipped with pressure, and flow sensor to achieve point to point transmission. Coordinator is connected with the PC via the serial port, and end nodes are arranged in different locations of environmental monitoring area to monitor environmental parameters through sensors on it. Finally, the monitored data of the stage hydraulic system is sent to the coordinator through the antenna in wireless method. Due to the connection between coordinator and PC, environmental monitoring results can be presented on the PC, realizing monitoring of pressure and flow of the node. The diagram of stage hydraulic system is shown in Figure 1.

The main task of this system is to monitor the pressure and flow of the stage. When the particular node index is higher than the threshold value, the monitoring system will emit an alarm to the staff. The PC receives and processes the data and then draws real-time dynamic curves, while the host computer can save the measured data of all nodes up to facilitate postviewing.

3. Monitoring Theories

Software design of the system is carried out in C language based on the Z-Stack protocol stack [26] on the programming; the software is the IAR Embedded Workbench for 8051. The system includes sensor node and coordinator node. After the formation of the coordinator electricity networks, sensor nodes automatically discover and join the network. Coordinator node is responsible for sending and receiving

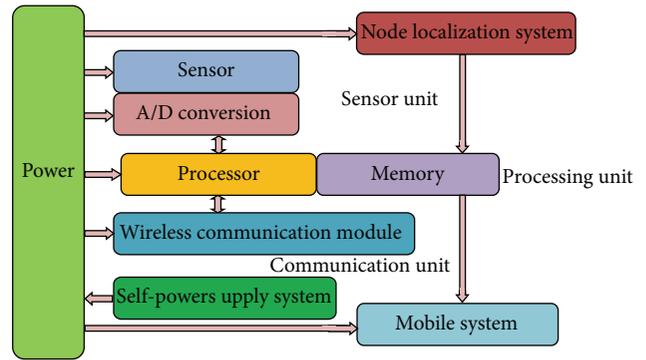


FIGURE 2: Model of the wireless sensor nodes.

data and communicating with the PC. While the sensor node can send and receive data, the data cannot be forwarded.

During the experiment, let each node (including the coordinator and a plurality of end devices) download IAR program via an emulator which has been compiled. As the download completed, reset the node and install the antenna, and then reboot the node, so the node can start working.

The sensor part consists of pressure sensor and flow sensor. When the stage hydraulic system works, the two sensors work as well. We can connect the sensors with one IO port of CC2530 chip via data line. When oil flows through pressure sensor and flow sensor, the real-time data can be transmitted to CC2530 chip. The measured data is analog signal and can be converted into digital signal by the AD conversion section in CC2530, thus continuing subsequent data transmission. In this paper, we select IO port P07 as the access point.

The coordinator is the core part of the stage hydraulic monitoring system. Its role is to establish the network and open the allowed binding function after power on. The sensor nodes join the network after power on and initiate binding request automatically. After the binding between sensor nodes and coordinator, the converted digital signal is transmitted to the coordinator in wireless transmission method. At last, the measured data are transmitted to the PC via serial connection and then conduct the data processing. Figure 2 displays the model of the wireless sensor nodes, and the workflow of the ZigBee network is presented in Figure 3.

4. Design of the Stage Hydraulic Monitoring System

4.1. Coordinator. Each ZigBee network has one and only one core part to build a ZigBee network. The network architecture diagram is shown in Figure 4. When a node joins, the address is allocated to the child node, and it is usually defined as equipment that cannot be powered down, which has no low power consumption state. Each ZigBee network needs only one coordinator, and different networks have different PAN IDs. The coordinator conducts judgment when capturing the interrupt: if it is the request for the new node joining the network, then the coordinator assigns a network address and binds it; if it is the control request for an added node (such as

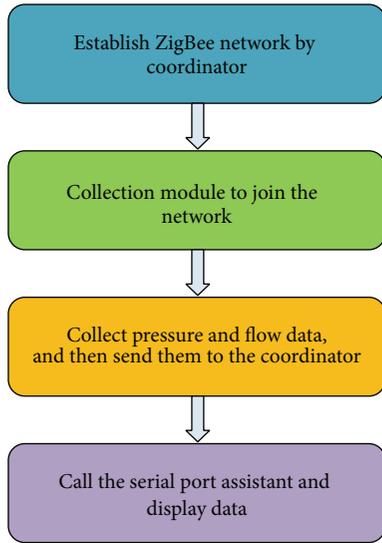


FIGURE 3: Workflow of the ZigBee network.

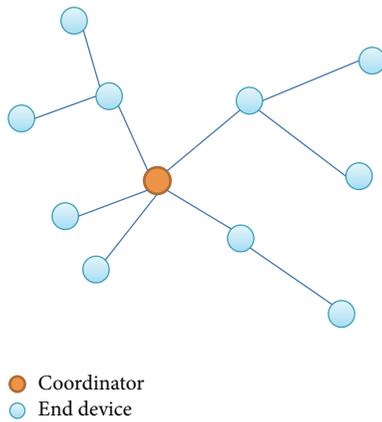


FIGURE 4: Network architecture diagram.

serial port communication request), then call corresponding processing function for processing. When the data processing is completed, the coordinator returns to wait for request to the listening state. Figure 5 displays the flow chart of the coordinator.

The coordinator plays a very important role in building up the entire network, binding with the sensor nodes to receive data, and transmitting the received data via the serial port to the connected PC.

According to the function requirements of the coordinator node, a SampleApp task is defined in the application layer of the node to complete data collection and communication in the user layer. User tasks are defined as the serial communication events, wireless communication events, and sleep events. Serial communication events are mainly for data communication with the host computer; wireless communication events are mainly mutual communication between the nodes, including flow and pressure data acquisition [27, 28], control commands, and routing and topology information

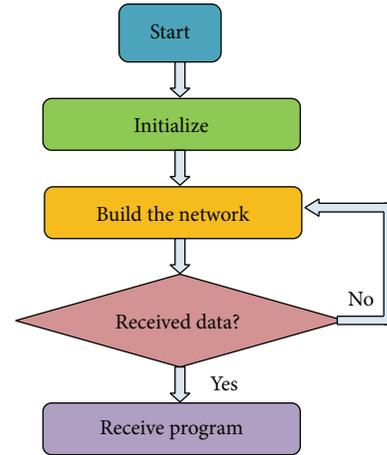


FIGURE 5: The flow chart of the coordinator.

of the nodes; sleep events are to inform the node into the dormant state.

Communication between the coordinator node and each sensor node is in a single-point transmission method; end devices communicate only with the coordinator. For the realization of this function, the coordinator must know the network address of each collection node, which requires each node to send its network address to the coordinator after joining in the network. After receiving the network addresses, the coordinator establishes a network address table to store these addresses, so that the user collects data based on the address table for each sensor, providing convenient communication between the coordinator node and each sensor node.

Firstly, the coordinator node needs to complete the initialization of serial port, network operating system, and such procedures. There are mainly the initialization of the operating system, the serial port, and the hardware. The initialization is performed with functions `osal_init_system()`, `MT_UartInit()`, and `HAL_BOARD$_$_INIT()`, respectively. Secondly, based on IEEE 802.15.4 standard, a free channel in 16 channels was scanned out by using the 2.4 GHz to build the new network. The channel scanning process is completed by `MAC_MlmeScanReq(maeMlmeScanReq_t *pData)` function. The third step is to establish a network. Calling the function `NLME_NetworkDiscoveryRequest()` to establish the network after the channel scan is successful. The fourth step is receiving data, and the entire WSN enters into the monitoring state after the node successfully joins the network. When it receives information transmitted from the child node, the coordinator deals with information through `SampleApp_MessageMSGCB(aflncomingMSGPacket_t *pkt)` and `*pkt` points to the received packet. If the received cluster ID is the information of `SAMPLEAPP_POINT_TO_POINT_CLUSTERID`, then the function of writing string to the serial port `HalUARTWrite(0, &pkt->cmd.Data[0], 5)` will be performed.

Since the Z-Stack protocol stack has provided the framework of agreement, the coordinator's code in IAR EW8051 only needs to be modified in the App layer. The ZigBee

network follows the beacon-enabled mode, in which communications are synchronized by specific frames (beacons) which are periodically emitted by the coordinator. For distributing the beacon interval of a ZigBee cluster tree among the superframes of the clusters, the algorithm follows the time division policy. In order to avoid those periods without any activity in any cluster, the goal is to maximize the use of the beacon interval of the network. So the assignment of the superframe orders should be considered.

In ZigBee sensor networks, the data are forwarded from the end nodes to a gateway or central node which most probably will reside in the coordinator. This centralization may cause the coordinator to become a traffic bottleneck. To avoid this situation, the active time of the coordinator should be privileged. A simple algorithm that prioritizes the role of the coordinator is to design its superframe order (SO_1) with twice the value of that of the rest of the cluster coordinators:

$$SO_i = SO \quad \forall i \in [2, N_C]; \quad SO_1 = 2 \cdot SO, \quad (1)$$

where N_C is the number of coordinators in the network. The aggregation of all the superframes must not be greater than the global beacon interval:

$$BI = a \cdot 2^{BO} \geq \sum_{i=1}^{N_C} SD_i = \sum_{i=1}^{N_C} a \cdot 2^{SO_i}, \quad (2)$$

where SD_i and SO_i are the superframe duration and superframe order of the i th coordinator, respectively. So we obtain

$$a \cdot 2^{BO} \geq \sum_{i=1}^{N_C} a \cdot 2^{SO_i} = a \cdot 2^{SO} + a \cdot (N_C - 1) \cdot 2^{SO} \quad (3)$$

from which we deduce the following quadratic equation:

$$(2^{SO})^2 + (N_C - 1) \cdot 2^{SO} - 2^{BO} \leq 0. \quad (4)$$

The maximum values of SO that avoid beacon collision can be deduced from the positive root of this equation:

$$SO = \left\lfloor \log_2 \left(1 - N_C + \sqrt{(N_C - 1)^2 + 4 \cdot 2^{BO}} \right) - 1 \right\rfloor. \quad (5)$$

As SD is defined as a power of SO , the above policy will tend in most cases to overestimate the superframe of the coordinator. To compensate this, we propose a variation of the previous method in which the SD of the coordinator is just twice the SD of the routers. To achieve this condition, the superframe order of the coordinator (SO_1) is set to the value of the order of the routers plus one unit in the following way:

$$SO_i = SO \quad \forall i \in [2, N_C]; \quad SO_1 = SO + 1. \quad (6)$$

Using these values and solving (2), we obtain the following:

$$SO = \left\lfloor \log_2 \left(\frac{2^{BO}}{N_C + 1} \right) \right\rfloor = \lfloor BO - \log_2(N_C + 1) \rfloor. \quad (7)$$

In order to operate as a coordinator in a beacon-enabled cluster network, a node has to transmit beacons and receive the contention access period for communicating with the nodes associated with it. In addition, a coordinator maintains synchronization with its parent by receiving beacons from it. The activity of a coordinator depends significantly on its location in the network and hence the requested throughput. As the power consumption of radio transmission and reception modes is quite similar, we estimate that the power consumption of a coordinator during contention access period equals the reception mode power consumption. The data flow to the uplink direction is performed by long MAC payloads containing sensing items per each frame. The duty cycle of a coordinator (DC_{COOR}) is modeled as follows:

$$DC_{COOR} = \frac{t_{TXB} + t_{RXB}}{I_B} + \frac{(t_{TXDL} + t_{RXA})(n_{DL} + n_D + 1)u}{I_U I_B A} + \frac{(t_{TXDS} + t_{RXA} + t_{RXDD} + t_{TXA})u}{I_D I_B} + \frac{t_{CAP}}{I_B} + \frac{t_{NS}}{I_{NS}}, \quad (8)$$

where DC_{COOR} is the duty cycle of a coordinator, t_{TXB} is the beacon transmission time, t_{CAP} is the CAP (contention access period) length, A is sensing items transmitted in a long frame (value 12), t_{TXDL} is the data transmission time for long frames according to the paper defined backoff models, n_{DL} is the number of the nodes hierarchically below a coordinator, and n_D is the number of the devices.

Similarly, the average power consumption of a coordinator can be calculated as follows:

$$P_{COOR} = \frac{E_{TXB} + E_{RXB}}{I_B} + \frac{t_{CAP} P_{RX}}{I_B} + \frac{(E_{TXDL} + E_{RXA})(n_{DL} + n_D + 1)u}{I_U I_B A} + \frac{(E_{TXDS} + E_{RXA} + E_{RXDD} + E_{TXA})u}{I_D I_B} + \frac{t_{NS}}{I_{NS}} + (1 - DC_{COOR}) P_S, \quad (9)$$

where P_{COOR} is the average power consumption of a coordinator, P_{RX} is the symbol of the measured platform power consumption, and E_{TXDL} is the energy of a data transmission for long frames according to the paper defined backoff models.

The average coordinator power consumption is a function of the uplink data transmission interval. As in the case of the device power consumption analysis, the network scanning interval equals 3 hours. With very low data rate network, coordinator power consumption can go below $200 \mu W$. In typical applications coordinator power consumption is between 1 and 10 mW.

4.2. End Device. The program of the end device is almost the same with the coordinator. There will be a little difference in

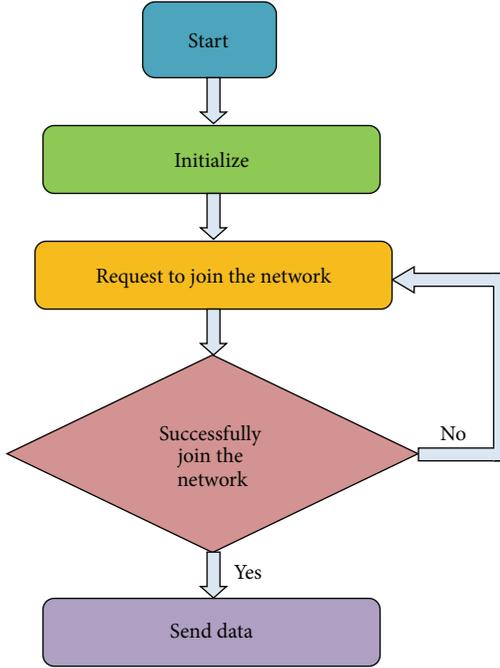


FIGURE 6: The flow chart of end device.

a few key points, such as the selection of the equipment type. The type of the collector node is the coordinator, while the sensor node is only served as an end device node to start. The flow chart of end device is shown in Figure 6.

As mentioned above, the entire network can support more than 64000 ZigBee nodes ideally. Multiple ends are utilized in the experiment, and we only need to download the “Coordinator” program, respectively, to these ends. All the PAN ID of the end nodes is the same as the coordinator, which will be automatically connected to the same ZigBee network.

When the establishment of the network is complete, end nodes can be open to join the network. Finish the initialization of the operating system of a node and the serial port, and then join the network through sending the request and calling the corresponding function. Finally, the sensor nodes also need to be bound to the coordinator node.

The first step is to initialize the operating system of each sensor node, LCD, and UART serial port. `osal_init_system()`, `InitLcd()`, `HalUARTInit()`, and other functions are needed to conduct the initialization. Secondly, when there is a network, the network layer will give sensor nodes feedback information of the ZDO layer. Requesting to join the network through the network layer and the function `NLME_NetworkFormationRequest()` is required to join the network. Thirdly, the binding of the sensor nodes and coordinator node is necessary. After receiving the response of joining the network, the sensor nodes call `ZDP_EndDeviceBindReq()` function to request binding. After the binding, sending and receiving data can be realized. The fourth step is sending the data. We need to register the event, set the number, and send time. The function `SampleApp_TaskID` is for registration event, function `SAMPLEAPP_AA_PERIODIC_MSG_EVT` is

for setting number, and function `SAMPLEAPP_SEND_PERIODIC_MSG_TIMEOUT` is used to set time for periodic sending data. When a registered event occurs, `SampleApp_SendPeriodicMessage()` is required to set the transmitted information.

To be able to operate in a ZigBee network, an end device receives beacons and exchanges data with a coordinator. If a communication link to the coordinator is lost an end device performs a network scanning. The rest of the time a device is in the sleep mode. The duty cycle of a device (DC_{DEV}) is calculated with beacon receptions, uplink and downlink data exchanges, and network scanning. The average network scanning interval (INS) depends on device speed and radio link quality. DC_{DEV} can be computed as follows:

$$DC_{DEV} = \frac{t_{RXB}}{I_B} + \frac{t_{TXDS} + t_{RXA}}{I_U I_B} \mu + \frac{(t_{TXDS} + t_{RXA} + t_{RXDD} + t_{TXA}) \mu}{I_D I_B} + \frac{t_{NS}}{I_{NS}}, \quad (10)$$

where DC_{DEV} is the duty cycle of a device which is calculated with beacon receptions, uplink and downlink data exchanges, and network scanning. I_B is the beacon interval. I_U is the uplink data transmission interval. I_D is the downlink data transmission interval. I_{NS} is the average network scanning interval which depends on device speed and radio link quality. t_{NS} is the network scanning time. t_{RXA} is the ACK reception time. t_{RXB} is the beacon reception time. t_{RXA} is the ACK transmission time. t_{RXDD} is the indirect data transmission time after a data request. t_{TXDS} is the data transmission time for short frames according to the paper defined backoff models. μ is the average number of transmission attempts per frame.

Similarly, the device power consumption is expressed as follows:

$$P_{DEV} = \frac{E_{RXB}}{I_B} + \frac{(E_{TXDS} + E_{RXA} + E_{RXDD} + E_{TXA}) \mu}{I_D I_B} + \frac{E_{TXDS} + E_{RXA}}{I_U I_B} \mu + \frac{E_{NS}}{I_{NS}} + (1 - DC_{DEV}) P_S, \quad (11)$$

where P_{DEV} is the average device power consumption. E_{NS} is the network scanning energy. E_{RXA} is the ACK reception energy. E_{RXB} is the beacon reception energy. E_{RXA} is the ACK transmission energy. E_{RXDD} is the energy for an indirect data transmission after a data request. t_{TXDS} is the energy of a data transmission for short frames according to the paper defined backoff models. P_S is the symbol of the measured platform power consumption.

The average end device power consumption is a function of the uplink data transmission interval. The network scanning interval is approximated to be averagely 3 hours, which corresponds to a deployment with low dynamics and good link qualities. In general, the power consumption decreases with longer beacon and uplink data transmission intervals, since the energy required for beacon receptions and data transmissions diminishes. At the longer beacon

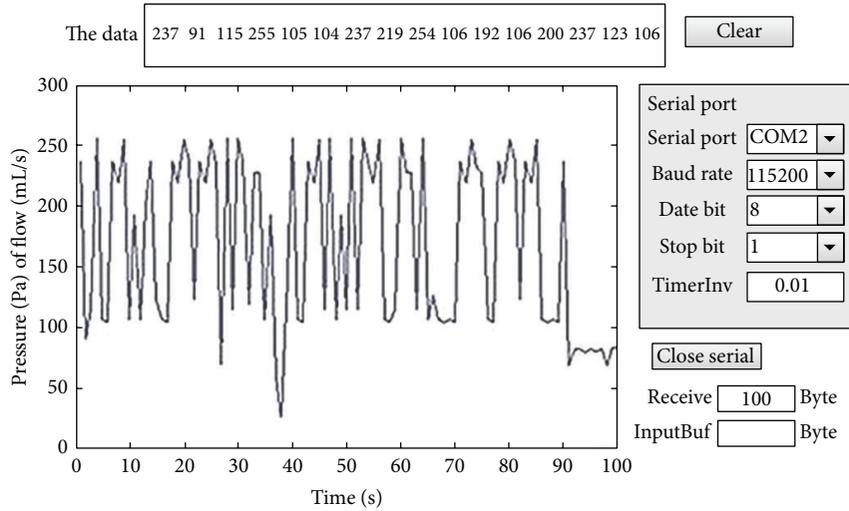


FIGURE 7: GUI monitoring interface under normal circumstances.

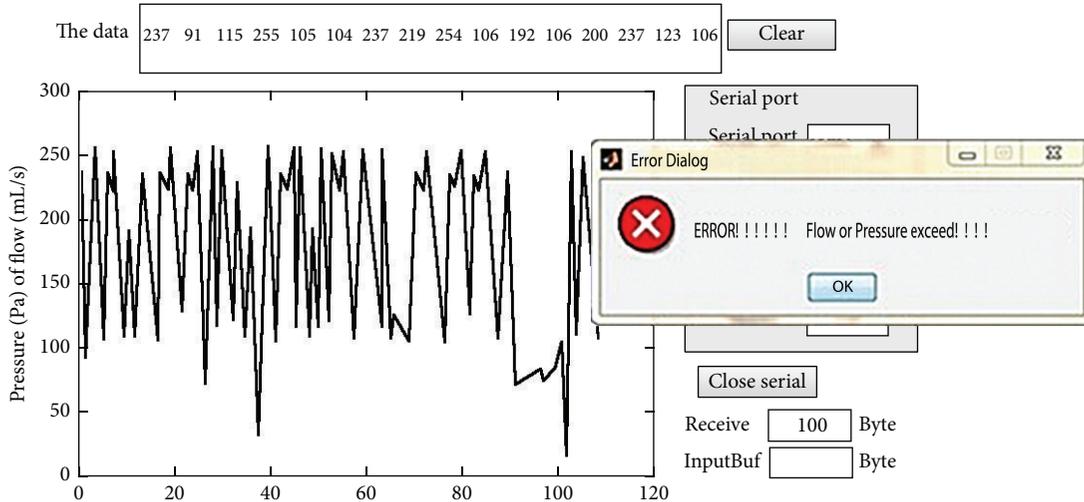


FIGURE 8: GUI alarm interface when pressure or flow exceeds normal.

intervals, the network scanning power becomes significant, since the network scanning energy increases directly the beacon interval.

4.3. *PC Part.* As for the PC part, the serial debugging assistant can be used to view data. The serial port, baud rate, parity bit, data bit, and stop bit have to be set up when using them. The received data can be displayed in the data box after opening the serial port. In order to strengthen the functions of the monitoring system and intuitively display the operating status of the monitoring system, we design a stage hydraulic monitoring system based on MATLAB GUI interface. The interface of stage hydraulic monitoring system includes real-time display of the received data, real-time data curve drawing, selection of COM port, selection of baud rate, selection of data bits and stop bits, definition of data read interval, open serial, off serial, and other functions.

5. Monitoring Data and Analysis

To obtain the monitoring data, the serial port, baud rate, parity bit, data bit, and stop bit need to be set up. If the serial port is open, the received data can be displayed in the data box of the serial debugging assistant. All the parameters of serial port in MATLAB GUI interface are set up as well. Details of the test data are as follows: Alert data: {32 32 49 46 48 86 32 32 49 46 48 86 32 32 49 46 48 86}. Normal data 1: {80 82 69 83 85 82 69 58 56 53 32 32 10 70 76 79 87 58 56 53 10 77} (received byte = 330). Normal data 2: {84 69 77 80 49 58 50 56 32 67 10 71 65 78 58 50 56 32 67 10} (received byte = 2200). Real-time data can be displayed, and real-time curve can be plotted according to these data. Figure 7 shows the GUI monitoring interface under normal circumstances.

An alarm module is also designed in the PC software monitoring system. When the monitored data exceeds the normal range as Figure 8 shows, a warning box will jump

out to notify the staff to carry out the appropriate emergency treatment for the stage hydraulic system.

Compared with traditional wired system, the proposed structure not only has a flexible working way, but also saves space and costs and is easy to implement. We can conduct real-time monitoring of pressure and flow data. If any failure occurs in the circuit, the maintenance personnel can be informed immediately, which can greatly improve maintenance efficiency and ensure the failure to be eliminated in time and success of the performance.

6. Conclusions

With the help of ZigBee technology, the proposed system can carry out real-time monitoring of the stage hydraulic system and successfully achieve data collection and real-time display of dynamic curve. A MATLAB GUI interface is designed to intuitively display the operating statue of the monitoring system. Once a failure occurs, the alarm will be triggered to inform the maintainers to find the location of the failure in time. Therefore, this system can improve maintenance efficiency and ensure the safety of the performers and the success of the performance.

Conflict of Interests

The authors declare that they have no conflict of interests.

Acknowledgment

This study is supported by the National Science-Technology Support Plan Projects (no. 2012BAH38F01-04).

References

- [1] K. Chen and H.-X. Wang, "The electro-hydraulic control system design of shield test platform," in *Proceedings of the International Conference on Measurement, Information and Control (MIC '12)*, vol. 2, pp. 900–906, Harbin, China, May 2012.
- [2] Z. M. Li, R. Z. Chen, and J. Li, "The synchronized coordinated control of tripartite hydraulic lifting platform," in *Proceedings of the 3rd International Conference on Power Electronics and Motion Control*, pp. 1456–1458, Beijing, China, 2000.
- [3] Y. Miao and S.-P. Wang, "Failure diagnosis of hydraulic lifting system based on multistage telescopic cylinder," in *Proceedings of the International Conference on Fluid Power and Mechatronics (FPM '11)*, pp. 828–834, Beijing, China, August 2011.
- [4] S.-J. Fu, M. Liffiring, and I. S. Mehdi, "Integrated electro-hydraulic system modeling and analysis," *IEEE Aerospace and Electronic Systems Magazine*, vol. 17, no. 7, pp. 4–8, 2002.
- [5] M. Dong, C. Liu, and G.-Y. Li, "Robust fault diagnosis based on nonlinear model of hydraulic gauge control system on rolling mill," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 2, pp. 510–515, 2010.
- [6] H.-Z. Tan and N. Sepeshri, "Parametric fault diagnosis for electrohydraulic cylinder drive units," *IEEE Transactions on Industrial Electronics*, vol. 49, no. 1, pp. 96–106, 2002.
- [7] A. Y. Goharrizi and N. Sepeshri, "Internal leakage detection in hydraulic actuators using empirical mode decomposition and Hilbert spectrum," *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 2, pp. 368–378, 2012.
- [8] R. Isermann and P. Ballé, "Trends in the application of model-based fault detection and diagnosis of technical processes," *Control Engineering Practice*, vol. 5, no. 5, pp. 709–719, 1997.
- [9] R. Isermann, "Supervision, fault-detection and fault-diagnosis methods—an introduction," *Control Engineering Practice*, vol. 5, no. 5, pp. 639–652, 1997.
- [10] B. Y. Xu, L. D. Xu, H. M. Cai, C. Xie, J. Y. Hu, and F. Bu, "Ubiquitous data accessing method in iot-based information system for emergency medical services," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1578–1586, 2014.
- [11] M. T. Lazarescu, "Design of a WSN platform for long-term environmental monitoring for IoT applications," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 3, no. 1, pp. 45–54, 2013.
- [12] S. Amendola, R. Lodato, S. Manzari, C. Occhiuzzi, and G. Marrocco, "RFID technology for IoT-based personal healthcare in smart spaces," *IEEE Internet of Things Journal*, vol. 1, no. 2, pp. 144–152, 2014.
- [13] Q.-P. Chi, H.-R. Yan, C. Zhang, Z.-B. Pang, and L. D. Xu, "A reconfigurable smart sensor interface for industrial WSN in IoT environment," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1417–1425, 2014.
- [14] Y.-X. Mao and P. Zhu, "A source-initiated on-demand routing algorithm based on the thorup-zwick theory for mobile wireless sensor networks," *The Scientific World Journal*, vol. 2013, Article ID 283852, 6 pages, 2013.
- [15] Y.-X. Mao, P. Zhu, and G.-Y. Wei, "A game theoretic model for wireless sensor networks with hidden-action attacks," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 836056, 9 pages, 2013.
- [16] Y.-X. Mao, "A semantic-based intrusion detection framework for wireless sensor network," in *Proceedings of the 6th International Conference on Networked Computing (INC '10)*, pp. 1–5, 2010.
- [17] P. Briff, A. Lutenberg, L. R. Vega, F. Vargas, and M. Patwary, "A primer on energy-efficient synchronization of WSN nodes over correlated Rayleigh fading channels," *IEEE Wireless Communications Letters*, vol. 3, no. 1, pp. 38–41, 2014.
- [18] S.-L. Lu, X. Huang, L. Cui, Z. Zhao, and D. Li, "Design and implementation of an ASIC-based sensor device for WSN applications," *IEEE Transactions on Consumer Electronics*, vol. 55, no. 4, pp. 1959–1967, 2009.
- [19] C. Shen, S. Harte, E. Popovici, B. O'Flynn, R. Atkinson, and A. Ruzzelli, "Automated protocol selection for energy efficient WSN applications," *Electronics Letters*, vol. 45, no. 21, pp. 1098–1099, 2009.
- [20] S.-H. Choi, B.-K. Kim, J. Park, C.-H. Kang, and D.-S. Eom, "An implementation of wireless sensor network for security system using Bluetooth," *IEEE Transactions on Consumer Electronics*, vol. 50, no. 1, pp. 236–244, 2004.
- [21] P.-Z. Yi, A. Iwayemi, and C. Zhou, "Developing ZigBee deployment guideline under WiFi interference for smart grid applications," *IEEE Transactions on Smart Grid*, vol. 2, no. 1, pp. 110–120, 2011.
- [22] M.-S. Pan and Y.-C. Tseng, "A lightweight network repair scheme for data collection applications in zigbee WSNs," *IEEE Communications Letters*, vol. 13, no. 9, pp. 649–651, 2009.

- [23] G. Ding, Z. Sahinoglu, P. Orlik, J. Zhang, and B. Bhargava, "Tree-based data broadcast in IEEE 802.15.4 and ZigBee networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 11, pp. 1561–1574, 2006.
- [24] H.-Y. Tung, K.-F. Tsang, H.-C. Tung, K.-T. Chui, and H.-R. Chi, "The design of dual radio ZigBee homecare gateway for remote patient monitoring," *IEEE Transactions on Consumer Electronics*, vol. 59, no. 4, pp. 756–764, 2013.
- [25] A. Wheeler, "Commercial applications of wireless sensor networks using ZigBee," *IEEE Communications Magazine*, vol. 45, no. 4, pp. 70–77, 2007.
- [26] M. R. Palattella, N. Accettura, X. Vilajosana et al., "Standardized protocol stack for the internet of (important) things," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1389–1406, 2013.
- [27] Z. Zhou, "Mining strongly associated rules," in *Proceedings of the 6th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD '09)*, vol. 2, pp. 99–102, August 2009.
- [28] Z. Zhou, "Mining frequent independent patterns and frequent correlated patterns synchronously," in *Proceedings of the 5th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD '08)*, vol. 5, pp. 552–556, October 2008.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

