

Development of Metal Polishing Dust Monitoring System using the Internet of Things and Cloud Server

<https://doi.org/10.3991/ijoe.v15i04.8971>

Zhiyao Fan

Jiangsu University, Jiangsu Sheng, China

Tianhong Pan^(✉)

Jiangsu University, Jiangsu Sheng, China

Anhui University, Anhui, China

thpan@live.com

Li Ma

Jiangsu University, Jiangsu Sheng, China

Abstract—In order to increase the management efficiency and decrease the maintenance costs in the traditional dust monitoring system, a novel real-time remote monitoring system using the Internet of Things and cloud server is proposed in this paper. The system includes several sensor nodes, a sink node and Cloud Server. The high-precision dust probe, temperature and humidity sensors, water flow sensors and hydrogen transmitters are integrated together in the sensor node to access the metal polished environmental information. Then the collected information is transmitted to sink-node using the 2.4 G wireless network. The sink-node uploads data to the Cloud Server through the communication of 4G network and TCP Socket. Based on the Browser/Server (B/S) model, a monitoring system is constructed using Tencent Cloud Server, C# language, and SQL database. As a result, the on-site metal polishing environmental information is obtained via the App and Web page.

Keywords—Internet of Things, 4G network, cloud server, socket, web service

1 Introduction

In recent years, the dust explosion in industrial processes has caused a lot of property losses and casualties. It is particularly serious in the metal polishing industry. Therefore, the safety production at the polishing site has been attracted more and more attention. How to acquire polished site data timely and provide accurate warning measurement has become a very urgent issue [1]. Furthermore, the manufacturing process cannot be effectively supervised during the metal polishing, and the common monitoring methods require a number of manpower and material resources [2].

The Internet of Things is a new information technology which is highly integrated with sensors, the Internet, and data processing technologies. It has been developed rapidly in recent years [3]. In this paper, a real-time remote monitoring system based

on the Internet of Things and Cloud Server has been developed. A number of sensor nodes are installed in the polishing site. Distributed node monitoring is used to increase the coverage rate when the sensor nodes are deployed. Each sensor node is responsible for collecting data in different local space, and the collected data is sent to the sink-node through the 2.4G wireless network. The aggregated and processed data is uploaded to the Cloud Server via the 4G network. APP and Web page view the metal polished environmental information conveniently and timely [4].

2 System Architecture

The real-time and remote monitoring functions are required in the developed system. As shown in Fig.1, the system includes three parts. The communication strategy used in sensors nodes and sink nodes is the wireless network (2.4G), which makes the equipment installation simplify. Through the data displayed in HMI, operators intuitively know the current information of polishing site. At the same time, the manager can obtain the same information through the APP and Web Page. Furthermore, the ability of self-judgment function is also required to make the power off when the polishing site is in danger for a period of time.

Using the Internet of Things technology and Cloud Server, the monitoring system is constructed as shown in Figure 1.

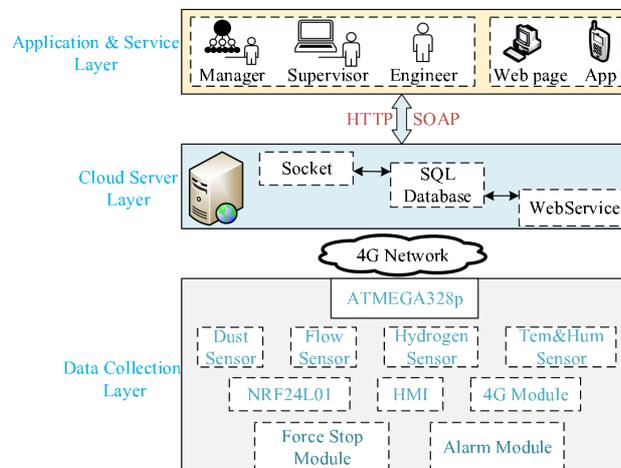


Fig. 1. The architecture of the system

The system is composed of Application & Service Layer, Cloud Server Layer, and Data Collection Layer. The functions of each part are as follows:

- **Application & service layer:** The Application & Service Layer is established for users including manager, supervisor or engineer, they can access the various vital data of the production site via the Web page and the Android App. The Web page and the App are used to interact with the Cloud Server.

- **Cloud server layer:** The Cloud Server is the core of the system which contacts the sink-node closely. In the Cloud Server, a socket is built to receive data uploaded from the sink-node and stores data in the SQL database. It also enables users to access the information with the Web page or the Android App via Web Service which provides a communication interface with HTTP protocol and the SOAP protocol.
- **Data collection layer:** The Data Collection Layer is specifically responsible for collecting various data detected by different sensor nodes, including dust concentration, temperature and humidity, flow rate, and hydrogen concentration. The data-preprocessing algorithm is also integrated into the sensor node, and final filtered data is transmitted to the sink-node through the 2.4G wireless network.

3 System Design

3.1 Data collection layer design

The Data Collection Layer contains six sensor nodes (each working station includes two sensor nodes) and a sink node. ATMEGA328p is used as MCU for each node, YTPM-2510 is selected as Dust sensor, AM2302 is selected as Tem&Hum sensor, LWGY-20mm is selected as Flow sensor, RTTPP DR-700 is selected as Hydrogen sensor, nRF24L01 is selected as a 2.4G wireless module. The hardware structure of the Data Collection Layer is shown in Figure 2.

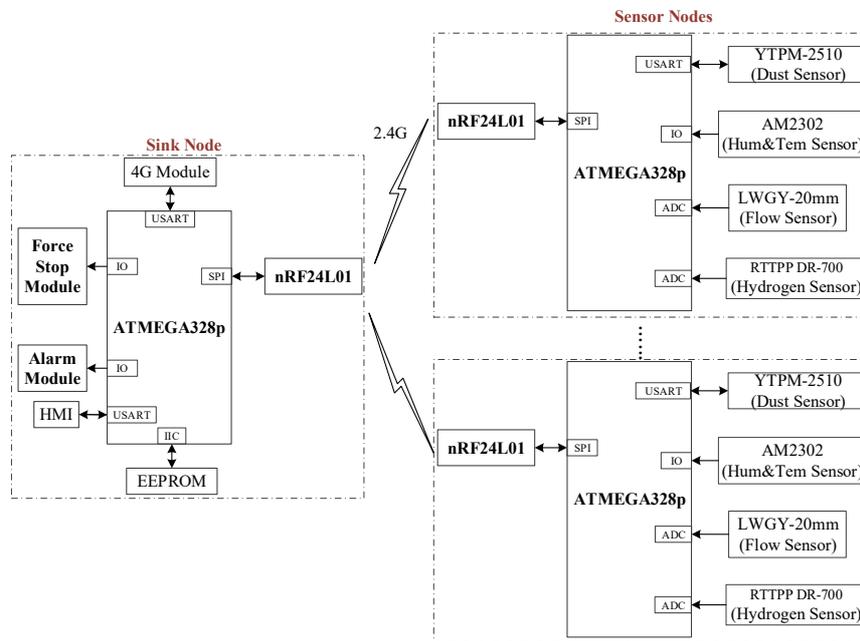


Fig. 2. The hardware structure of Data Collection Layer

The sensor nodes are deployed in different places at the metal polishing production site so that the collected data is able to describe the entire polishing environment. After all the data is collected, a simple filtering algorithm is used to eliminate abnormal data. Then, the sensor node will transmit the correct data to the sink node via 2.4G wireless network [5].

The sink node is the brain of Data Collection Layer, which uploads the received data to the Cloud Server through 4G network, updates the information on the HMI interface in time, and determines whether the data exceeds the alert threshold. If the received data exceeds the threshold for a period of time, the sink node will give an alarm and cut off the power of the polisher [6]. In the developed system, the operator can set different thresholds for each operating zone via HMI according to different weather conditions.

3.2 Working principle of 4G module

A standard 4G module, i.e. Huawei ME909S-821, is selected as the communication interface of sink node which is mature for secondary development in the market. [7]. The module provides a standard UART interface to connect to the MCU, and the MCU can directly use the 'AT' command by sending a string to manage the 4G module efficiently. The communicating procedure between the sink node and MCU is shown as follows:

- Function: test instruction, check the communication
- MCU Transmit: 'AT'
- 4G Module response: 'AT', 'OK'
- Function: check SIM Card is installed or not
- MCU Transmit: 'AT+CPIN?'
- 4G Module response: 'AT+CPIN', 'OK'
- Function: Set the SMS of Module into TEXT MODE
- MCU Transmit: 'AT+CMGF=1'
- 4G Module response: 'AT+CMGF=1', 'OK'
- Function: Set the character of Module into GSM MODE
- MCU Transmit: 'AT+CSCS=GSM'
- 4G Module response: 'AT+CSCS=GSM', 'OK'
- Function: Attach 4G network
- MCU Transmit: 'AT+CGATT=1'
- 4G Module response: 'AT+CGATT=1', '^CINIT: 4, 8192, 37', '^STN: 37', 'OK'
- Function: Open the PDP mobile scenario
- MCU Transmit: 'AT+CGACT=1,1'
- 4G Module response: 'AT+CGACT=1,1', 'OK'
- Function: Set connection mode into TCP/IP protocol, set connection IP address and port number
- MCU Transmit: 'AT+CIPSTART=TCP, 115.159.64.98, 8008'
- 4G Module response: 'AT+CIPSTART=TCP, 115.159.64.98, 8008', 'Connect OK', 'OK'

- Function: Send the specified data to Web Socket
- MCU Transmit: 'AT+CIPSEND'
- 4G Module response: 'AT+CIPSEND,' '>'
- Send the data as soon as receiving the character of '>'
- In the communication between MCU and 4G Module, the mentioned 'AT' commands should be delivered in order. After getting the specific response from the 4G module, the MCU can send next 'AT' command. In addition, a special resetting circuit is used in hardware PCB design in case the 4G module is crashed. The procedure is shown in Figure 3.

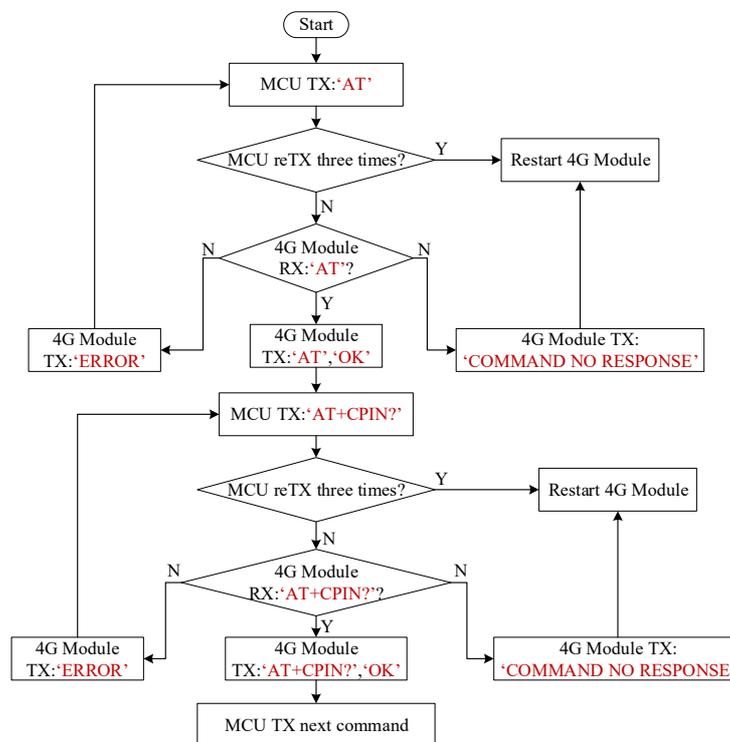


Fig. 3. Working procedure of MCU and 4G Module

3.3 Cloud database design

From the perspective of compatibility, the Windows Server 2012R2 is installed in the Cloud Computer System, so the SQL database is selected as database server which has better scalability. The data table which is the most important operational object is the basic unit for storing data in database. During the design of the database, the principle of storing different data types in different tables according to the prior regulations should be followed. Therefore, when designing the data table, the variable type of the stored data should be considered. The data table is showed in Table 1.

Table 1. SQL data table design

Column name	Data Type	Description
Dtime	smalldatetime	Data collected time
Company	smallint	Company index
Depart	tinyint	Working depart index
Num	smallint	Sensor node index
PM2.5	smallint	PM2.5 value
PM10	smallint	PM10 value
Temperature	smallint	Temperature value
Humidity	smallint	Humidity value
Hydrogen	smallint	Hydrogen value
Flow	smallint	Water flow value

3.4 Cloud socket design

The Cloud Socket is designed by using the ASP.NET technology, C# language, and Visual Studio 2015. The network communication between the Cloud Server and the 4G module is accomplished through TCP/IP protocol [8]. It is used as the basis of network transmission and used to describe the destination IP and port of the transmission through accessing socket. The sink node sends data to the socket operating on Cloud Server through the 4G network and waits for a response. TCP/IP protocol cluster is the basic data unit in the transmission. In this mode, as long as the network connection can be established reliably, the socket can create a data stream which is able to open, close, read, write, and modify the transmitted connection [9]. There are five main APIs used in creating the socket:

- The bind() function is used to put the socket and the Server’s local network address bundled together;
- The SQL Connection() function is used to establish the stable connection with the local SQL Database;
- The listen() function is used to keep the created socket always listening;
- The accept() function is used to allow the Socket to accept connection requests from various clients;
- The receive() function is used to receive the data packages sent from clients;
- The working procedure of software is shown in Figure 4.

When the thread of the socket is started, the client-listening mechanism starts to work automatically. The main program continuously receives and handles the request information heard by the listening thread. Then, the program determines whether the received data is correct or not, and stores the correct data into SQL database.

In order to identify the information from the different corporation and different polishing zone only with one all-purpose socket, a specific data format is defined in the process of data-receiving. The data stream is constituted by the following elements including< corporation code, zone code, sensor node number, PM2.5 value, PM10 value, humidity value, temperature value, sensor node number, PM2.5 value, PM10 value, humidity value, temperature value, flow value, flow value, hydrogen

value>, socket will distinguish the data stream by the symbol of ‘,’. The structure of data stream is indicated in Figure 5.

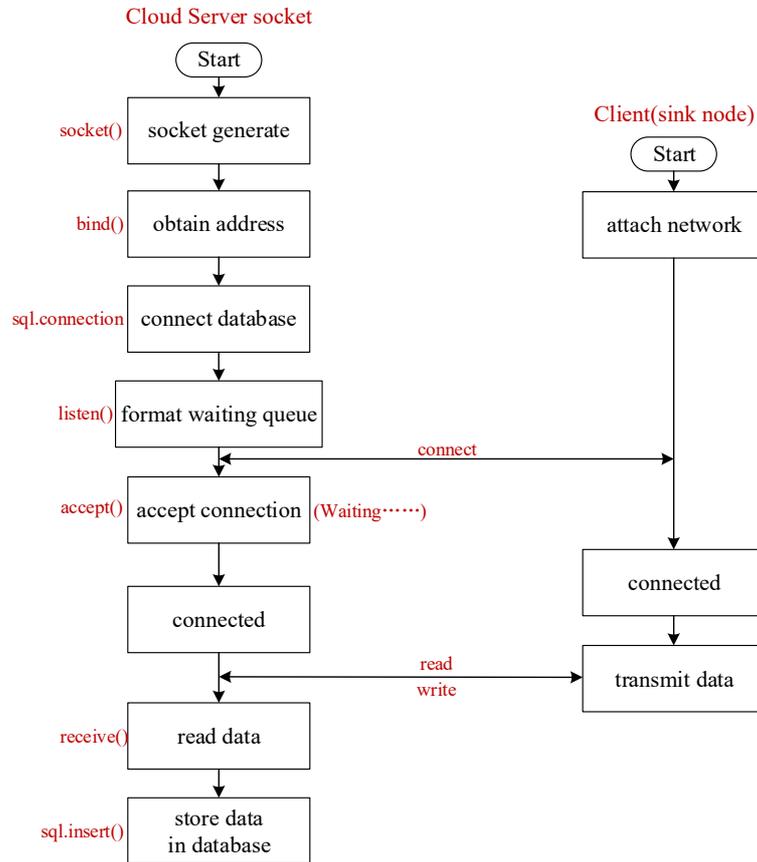


Fig. 4. Working procedure of cloud socket

Data Stream	ID of Data head		Valid data									
	First Byte	Second Byte	Third Byte	Fourth Byte	Fifth Byte	Sixth Byte	Seventh Byte	Eighth Byte	Ninth Byte	Tenth Byte	Eleventh Byte	Twelfth Byte
	corporation code	zone code	sensor node	PM2.5 value	PM10 value	hum value	tem value	sensor node	PM2.5 value	PM10 value	hum value	tem value
			Thirteenth Byte	Fourteenth Byte	Fifteenth Byte							
			flow value	flow value	hydrogen value							

Fig. 5. The structure of data stream

3.5 Cloud web service design

Web Service is a service-oriented technology, it is mainly used to interact with App in the system. Simplified standard Web protocols are used to shield differences between different platforms involving Android and iOS through XML, SOAP, and WSDL [10]. The software architecture of Web Service is shown in Figure 6.

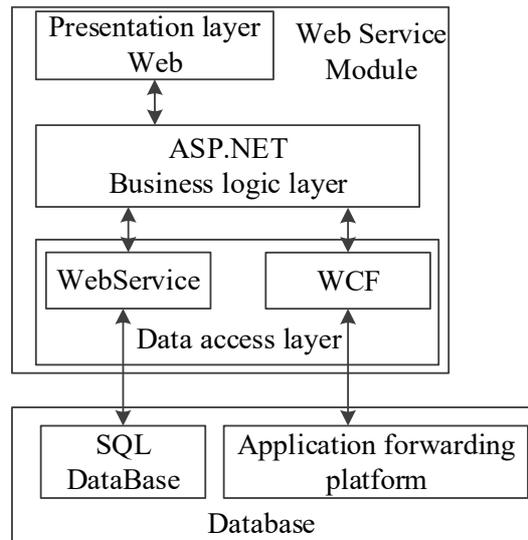


Fig. 6. The software architecture of Web Service

The idea of using the same interface and different services are adopted when programming the interface of Web Service, and all business information is encapsulated in an XML file. The unified interface is used to read business information, and the incoming and outgoing parameters are delivered in XML file [11]. The interface service module involves: Web Service interface module, XML operation module, database operation module. Operation type, SQL statement or stored procedure, the number of parameters and parameter list, etc. The XML file are loaded according to the operation ID of users. The correctness of the call of the interface is on the basis of the analysis of the value [12]. The specific process is shown in Figure 7.

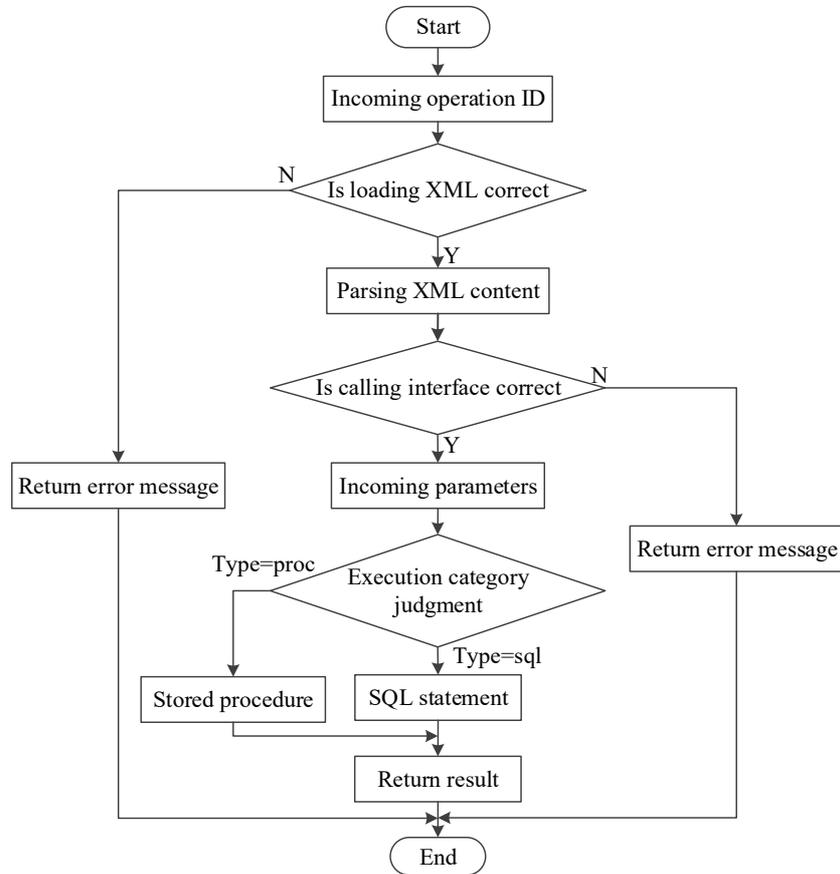


Fig. 7. Workflow chart of Web Service

After publishing on Windows platform based on IIS is shown in Figure 8.



Fig. 8. Published interface of Web Service

3.6 Remote terminal design

It is required to use the cell phone or PC to conveniently access the data stored in the Cloud Server, because the supervisor needs to know the security status of the site at any time.

App design: Java and Android Studio are used for developing the App due to it is based on Android system. The architecture of App is shown in Figure 9.

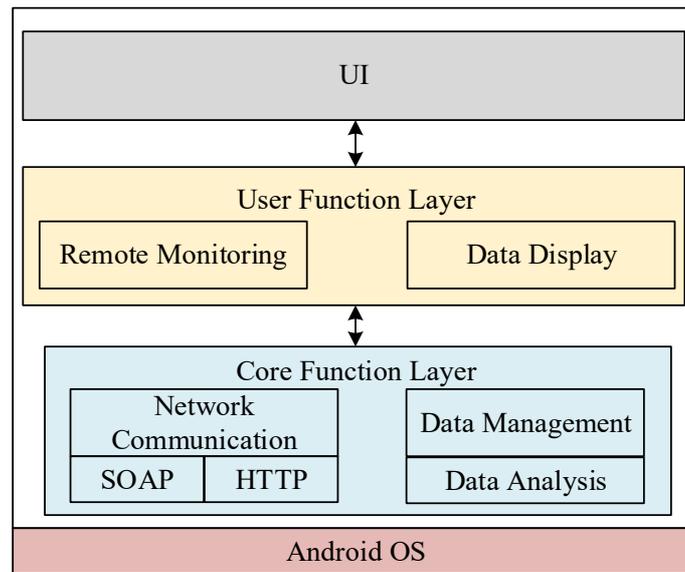


Fig. 9. The architecture of App

The design of App can be separated into three layers: UI, User Function Layer, and Core Function Layer. In Core Function Layer, the network communication is the most vital part in a software program, so the k-soap Third-party packages involving numerous APIs are used to communicate with the Web Service. The App transmits the data request to the Web Service through the soap protocol and HTTP protocol, after getting the response, the XML file will be analyzed into the available data [13]. The software design of communicating with Web Service is shown in Figure 10.

Another vital part of the App is transforming the number of data into curves. There are six steps in the curve-design which is the core part of User Function Layer:

- Import MP Android Chart;
- Initialize the background;
- Set the Entry of X axis and Y axis;
- Set the attributes of curves;
- Add curves to the display list;
- Generate curves;
- In the end, the User Function Layer interacts with the user via UI.

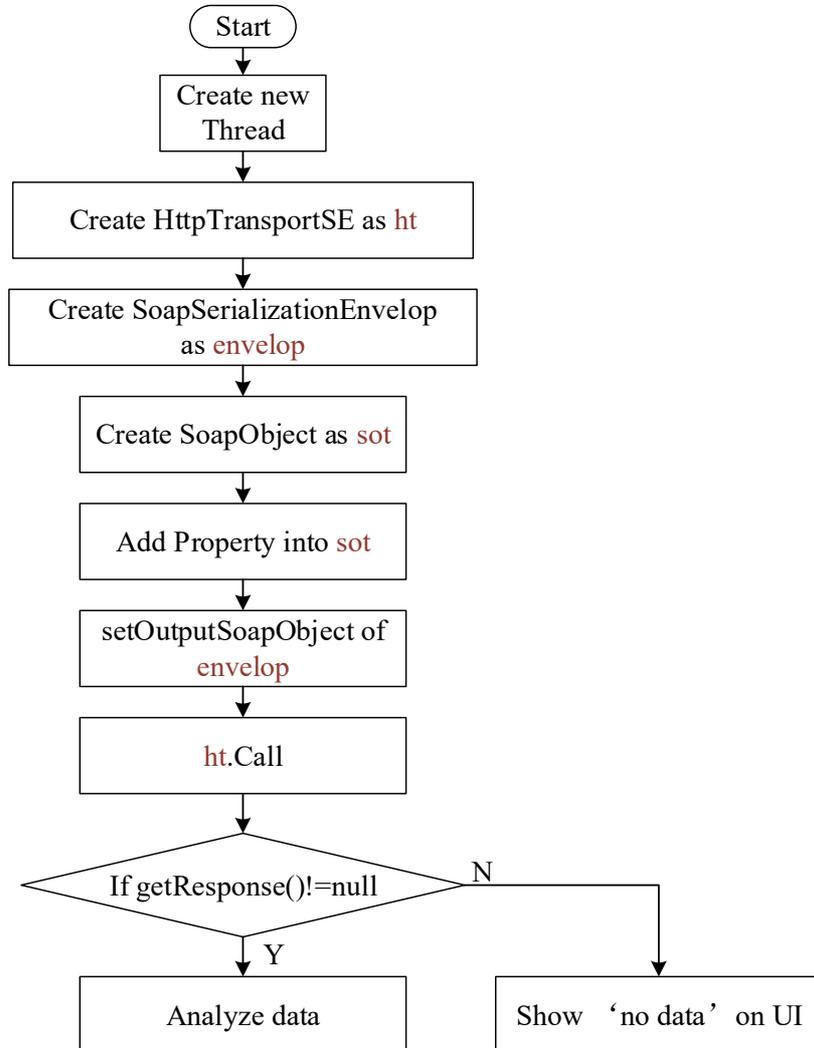


Fig. 10. Software design on communicating with Web Service

Webpage design: C# and Visual Studio are used for developing the Web page, and it is based on ASP.NET technology. SQL Data Source widget, Text Box widget, Label widget, Button widget, Chart Widget are selected for creating the Web page[14]. It consists of User Option Module and Data Display Module. The structure of Web page is shown in Figure 11.

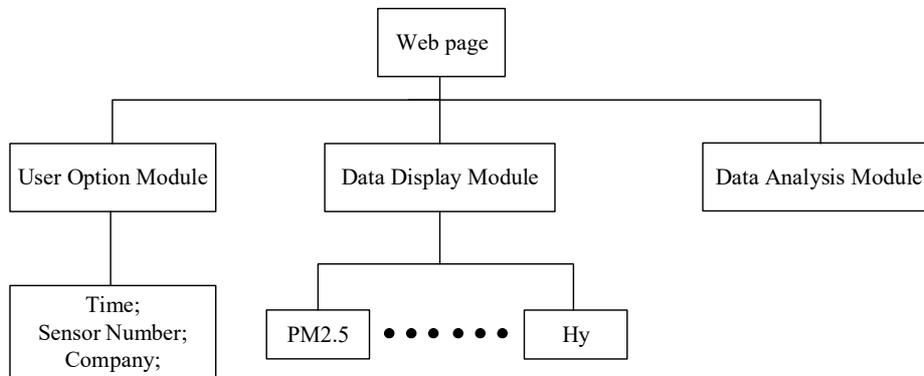


Fig. 11. Structure of the Web page

4 System Test

4.1 Prototype modules

The prototype module with sensor nodes is shown in Fig 12.



Fig. 12. Prototype module with sensor nodes

The sink node with HMI is shown in Figure 13.



Fig. 13.Sink node with HMI

4.2 Socket receiving data test

After design the socket, it needs to be encapsulated as a .exe file. Press the ‘Start’, and the Socket will be listening to the clients and wait for the data (shown in Figure 14).

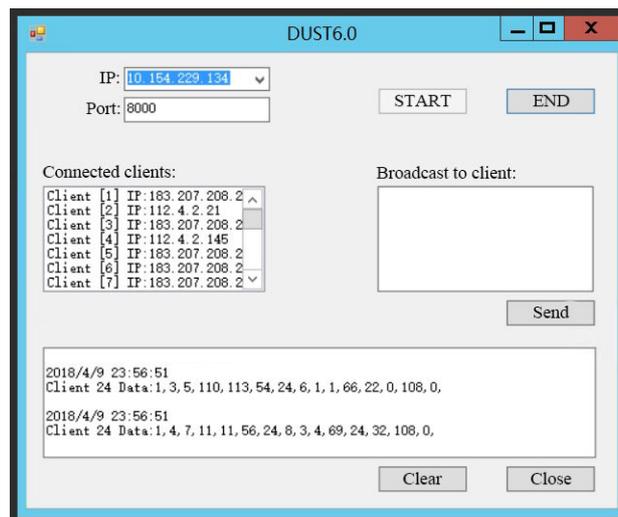


Fig. 14.Socket receiving data

4.3 Web page and App test

The remote client debugging interface is shown in Figure 15 and Figure 16. It can be seen that the data is displayed in the form of a curve, and the users can optionally select the workspace and the time to view, which primarily prove the reliability and superiority of the designed system.

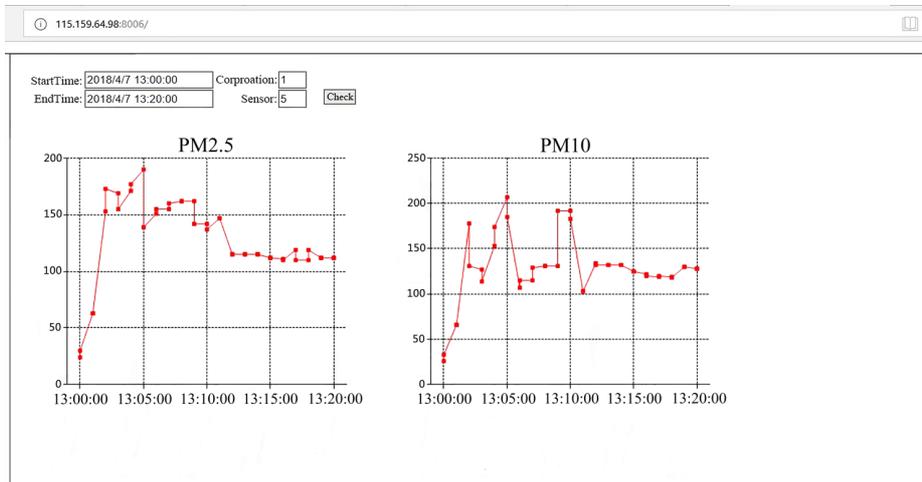


Fig. 15. Webpage Test

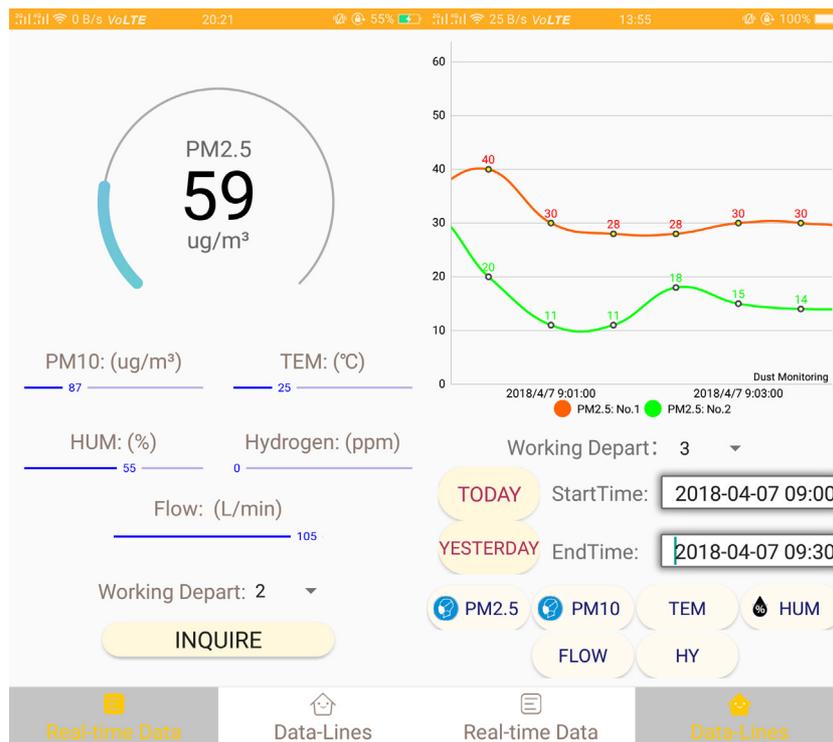


Fig. 16. App Test of real-time data and historical data

5 Conclusion

Development of Metal Polishing Dust Monitor using the Internet of Things and Cloud Server has several advantages compared with the traditional dust monitoring systems. The Internet of Things and Cloud Server technology combined with the latest 4G network makes the collection and processing of data information more reliable and stable so that regulators can remotely monitor any situation on the production site and make corresponding actions according to actual needs, which realize the 24-hour unattended stable operation. The future work is to establish a mathematical model based on big data. After analyzing the environmental data collection, we can provide targeted causes and solutions when there are problems in the field environment or monitoring equipment. Consequently, the treatment of polishing dust environment could be realized based on big data.

6 Acknowledgement

The authors would like to thank the financial support provide by the key R&D Projects in Jiangsu Province under Grant BE2018370 and Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD)

7 Conflict of Interest

The authors indicated no potential conflicts of interest.

8 References

- [1] Sahidi T T, Basuki A, Tolle H. MIOT Framework, General Purpose Internet of Things Gateway using Smartphone[J]. International Journal of Online Engineering, 2018, 14(2):6. <https://doi.org/10.3991/ijoe.v14i02.7326>
- [2] Pandey S K, Zaveri M A. Event Localization in the Internet of Things Environment [J]. Procedia Computer Science, 2017, 115:691-698. <https://doi.org/10.1016/j.procs.2017.09.154>
- [3] Song Y H, , Ming, et al. An Internet of Energy Things Based on Wireless LPWAN[J]. Engineering, 2017, 3(4):460-466. <https://doi.org/10.1016/J.ENG.2017.04.011>
- [4] Xian K. Internet of Things Online Monitoring System Based on Cloud Computing[J]Computinional Journal of Online Engineering, 2017, 13(9):123. <https://doi.org/10.3991/ijoe.v13i09.7591>
- [5] Gao H, Su J. Design Of The Temperature Signal Wireless Receiver And Display System On Polishing Interface In Chemical Mechanical Polishing [J]. Procedia Engineering, 2011, 24:417-421. <https://doi.org/10.1016/j.proeng.2011.11.2668>
- [6] Zhao J, Trivedi K S, Grottke M, et al. Ensuring the Performance of Apache HTTP Server Affected by Aging[J]. IEEE Transactions on Dependable & Secure Computing, 2014, 11(2):130-141. <https://doi.org/10.1109/TDSC.2013.38>

- [7] Jalilian M, Sariri H, Parandin F, et al. Design and implementation of the monitoring and control systems for distribution transformer by using GSM network [J]. *International Journal of Electrical Power & Energy Systems*, 2016, 74:36-41. <https://doi.org/10.1016/j.ijepes.2015.07.022>
- [8] Sung W T, Chang K Y. Health parameter monitoring via, a novel wireless system [J]. *Applied Soft Computing*, 2014, 22(5):667-680. <https://doi.org/10.1016/j.asoc.2014.04.036>
- [9] Hesmans B, Bonaventure O. An enhanced socket API for Multipath TCP [J]. 2011TCP [6].
- [10] Indu I, Anand P M R, Bhaskar V. Encrypted Token based Authentication with Adapted SAML Technology for Cloud Web Services [J]. *Journal of Network & Computer Applications*, 2017, 99:131-145. <https://doi.org/10.1016/j.jnca.2017.10.001>
- [11] Li H, Pan T. Development of Physiological Parameters Monitoring System using the Internet of Things [J]. *International Journal of Online Engineering*, 2017, 13(9):87. <https://doi.org/10.3991/ijoe.v13i09.7436>
- [12] Viswanath S K, Yuen C, Tushar W, et al. System design of the internet of things for residential smart grid [J]. *IEEE Wireless Communications*, 2016, 23(5):90-98. <https://doi.org/10.1109/MWC.2016.7721747>
- [13] Fuqaha A A, Guizani M, Mohammadi M, et al. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications[J]. *IEEE Communications Surveys & Tutorials*, 2015, 17(4):2347-2376. <https://doi.org/10.1109/COMST.2015.2444095>
- [14] Fang S, Xu L D, Zhu Y, et al. An Integrated System for Regional Environmental Monitoring and Management Based on Internet of Things [J]. *IEEE Transactions on Industrial Informatics*, 2014, 10(2):1596-1605. <https://doi.org/10.1109/TII.2014.2302638>

9 Authors

Tianhong Pan (corresponding author) is a professor of the School of Electrical Engineering & Automation in Anhui University, and School of Electrical & Information Engineering in Jiangsu University. He has received Ph.D. degree in control theory and control engineering from Shanghai Jiao Tong University in 2007. His current research interests include multiple model approach and its application, machine learning, virtual metrology, predictive control and run-to-run control theory and practice, system identification (mobile: (86)15805298357, Fax: (86)511- 88790618, E-mail: thpan@ujs.edu.cn).

Zhiyao Fan is the master degree candidate of the School of Electrical & Information Engineering in Jiangsu University. His major is control engineering. His current research interests include embedded system and data fusion.

Li Ma is an associated professor of the School of Electrical & Information Engineering in Jiangsu University. She has received the B.S. and M.S. degrees in mathematics from Anhui Normal University, Wuhu, China, in 2004 and 2007, respectively. She has received Ph.D. degree in control theory and control engineering from Southeast University in 2011, Nanjing, China. Her research interests include time-delay systems, stochastic systems, networked control systems, and Markovian jumping system

Article submitted 4 June 2018. Resubmitted 26 November 2018. Final acceptance 20 January 2019. Final version published as submitted by the authors.