

# Meter Management System based on the Internet of Things

Zhongxing Huo, Yongjie Yang\*, and Yuelan Ji

*School of Electronics and Information, Nantong University, Nantong, 226019, China*

---

## Abstract

To realize the convenient and intelligent monitoring and management of cable manufacturers' production lines, this paper designs a set of meter management systems based on the Internet of things that can replace artificial monitoring cable production. The system uses the STM32 processor of the ARM kernel as its core, and it combines embedded technology, wireless communication technology, Internet of things technology, 485 bus technology, and upper computer software technology. The system includes the front-end signal processing equipment, the Internet of things server, and the client software, which can respectively realize the information collection and automatic control of the cable production line, real-time transmission of production data under the MQTT protocol, and storage and graphical display of production data. The system was tested in the factory and proved the timeliness of the control of the production line, the real-time nature of the data transmission, and the practicality of the client software, which plays a great role in the modernization of the traditional factory.

*Keywords:* cable production; MQTT; internet of things; STM32; intelligent management

(Submitted on May 19, 2018; Revised on June 27, 2018; Accepted on July 29, 2018)

© 2018 Totem Publisher, Inc. All rights reserved.

---

## 1. Introduction

In cable production, each production line corresponds to a meter equipment. The equipment needs to record the length of production manually and suspend production when the specified length is reached. Nowadays, in the process of construction, the demand for electric cable or communication cable is increasing, so many cable manufacturers have vigorously expanded their production lines. If manual methods are still used to record production data and control the production process, it will be complex, inefficient, and laborious work. It is also easy to make mistakes [1].

In today's powerful background of the Internet of things, the traditional production industry has been greatly impacted. The combination of information technology and the manufacturing industry is becoming more and more urgent, and it is also an inevitable trend of future industrial development. Therefore, cable manufacturers are required to realize online unified and intelligent management of each line. In this paper, the meter management system based on the Internet of things realizes the production data collection of each production line and the production start-stop control, the data analysis and storage, and the textual and graphical display of the information [2]. The client in this paper can display the production status of each production line in real time, generate statistics based on the production data, and draw the chart using the data, which is beneficial for analyzing the production capacity of the factory and the development of the company.

## 2. Overall Design of System

The meter management system based on the Internet of things is composed of three parts: front-end signal processing device, Internet of things server, and client software. The overall structure diagram of the system is shown in Figure 1.

The front-end device is placed on the cable production line, and one device corresponds to one production line. Through the 485 or 232 interface that connects the meter device and the winding machine, the speed information detected by the meter device is transmitted to the front-end device through the serial port [3]. Before workers start the production line,

---

\* Corresponding author.

E-mail address: [yang.yj@ntu.edu.cn](mailto:yang.yj@ntu.edu.cn)

the length of the production on the front-end device must be inputted. After the setting is completed, the front-end device starts to convert the speed information and time information into the production length. When abnormal production occurs or during the production process, the workers will be notified by an acousto-optic alarm, and the winding machine will be controlled to stop the operation of the production line. The communication of the whole system is wireless, avoiding the issue of having too many cables in the production plant. The stability of communication is directly related to the reliability of the system. At present, there are many reliable wireless communication methods such as ZIGBEE, WIFI, and GPRS [4]. Considering that ZIGBEE networking is complex and inflexible and GPRS requires life-long communication fees, which are troublesome to maintain, this system chooses to use WIFI communication. The company only needs to arrange a wireless network in the factory area. The network can be used for communication of the system as well as for other uses.

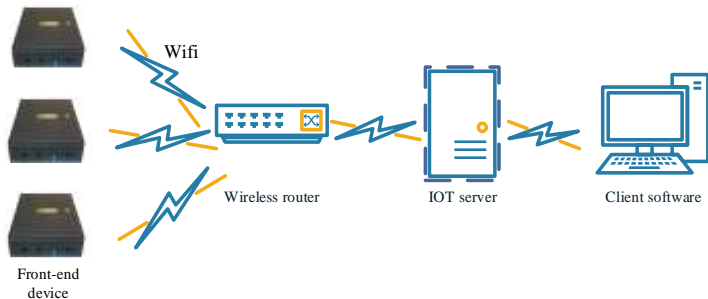


Figure 1. The overall structure of the system

Through the wireless network established by WIFI, many front-end devices can be easily connected to the Internet of things servers, and then the obtained data are sent to the Internet of things servers. The IOT server communicates with the front-end device and the client using the MQTT protocol [5]. The data acquired by the client software connected to the server is used to interact with the user. The user can view the production speed and the production length of each cable production line in real time on the client side software, and he or she can also view the historical production speed and length curve changing diagram.

3. Hardware Design of System

The front-end device is used for the abutment between the meter equipment and the production line. It can obtain the speed and length information of the cable production in real time. As the front-end important information acquisition device of the system, it plays an important role in the whole system. According to the hardware block diagram 2, the hardware design mainly includes the MCU minimum system design, the power module design, the 485 module design, the alarm module design, and the WIFI module design.

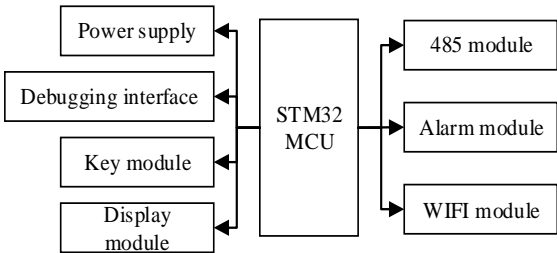


Figure 2. The hardware block diagram of the front-end device

3.1. The MCU Minimum System

The MCU minimum system is the command center of the whole circuit board, and the choice of the chip and the stability of operation are particularly important. The paper selects the STM32F103 microcontroller, which is currently widely used and has high code execution efficiency [6]. The chip has 51 universal fast I/O ports. The I/O port has good level compatibility. It can input 5V signals and can also input 3.3V signals, which is convenient for the peripheral circuit design. The power supply of the chip is 3.3V. To obtain a pure power supply, the input pin of the chip is connected in parallel with 1UF and 0.1uf capacitance to filter out low-frequency noise and high-frequency noise, respectively. In order to facilitate program downloading and debugging, the serial port and SW download interface are exported, as shown in Figure 3.

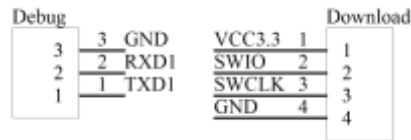


Figure 3. The download and debug interface

### 3.2. The Design of Power Supply Module

For the miniaturization and good heat dissipation of the circuit board, the power supply adopts the external 12V switching power supply. Because the modules in the front-end circuit board need 5V and 3.3V power supplies, the following power supply step-down circuit is designed, as shown in Figure 4.

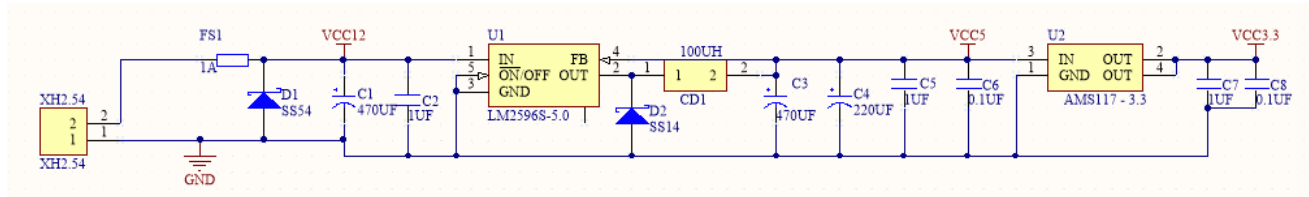


Figure 4. Power supply module

The 1A self-recovery fuse and diode of the power input terminal are used for the reverse connection protection of the power supply to prevent the wrong operation from burning the circuit board. For the purity and stability of the power supply, a power supply is added before and after the depressurization of each level, in which the LM2596-5.0 is used to generate 5V voltage to power the 485 module and the display module. The 3.3V voltage produced by AMS1117-3.3 is used in the STM32 MCU and the WIFI module. The use of multistage decompression avoids the problem of heat generated by the power chip itself due to the excessive voltage drop.

### 3.3. The Design of 485 Module

The 485 module is used to connect the counter meter device to obtain the information of the measured speed of the cable. The circuit diagram is shown in Figure 5.

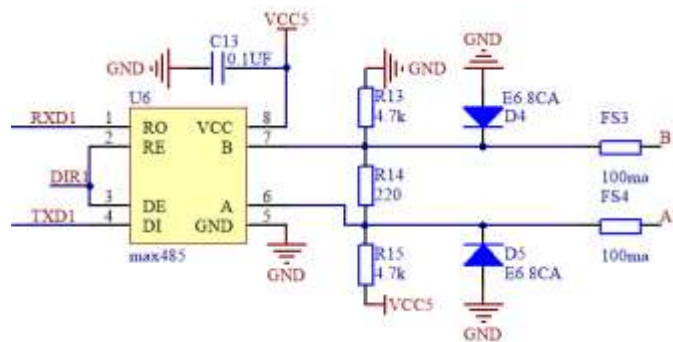


Figure 5. The circuit diagram of 485 communication interface

MAX485 connects to the MCU through the serial port. The data sent by the MCU are transmitted through two signal lines A and B, in which D4 and D5 clamp the voltage on the 485 bus below 6.8V. This prevents the fluctuation of the 485 signal line voltage, which can lead to chip burning. FS3 and FS4 are self-recovery fuses, which are used to prevent excessive currents on the bus and protect MAX485 chips.

### 3.4. The Design of Alarm Module

In the normal production process of the cable, if the cable material is exhausted or the production is finished, it is necessary to warn the staff and suspend the production line using alarms. The alarm circuit is shown in Figure 6. The warn port connects to the ordinary I/O port of the microcontroller and receives the alarm control signal of the microcontroller. U4 is the MOS tube. When the gate input is high, it can drive the alarm device. When it is low, the alarm device is turned off and

it acts as a switch. Because the warning light is driven by 12V voltage, the MCU and other modules are powered by 3.3V and 5V. To protect the MCU, the MCU and MOS are connected by an optical coupler in the circuit, isolating low voltage and high voltage, which can effectively prevent the MOS failure from causing other internal circuit damages.

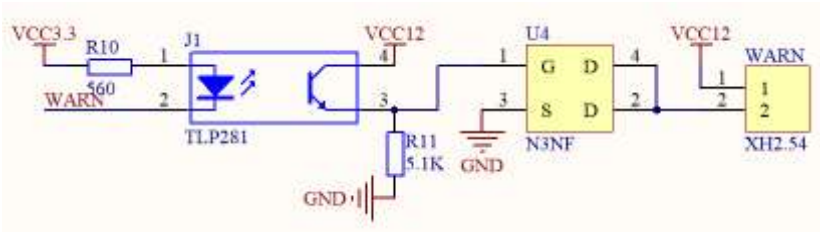


Figure 6. Alarm circuit

3.5. The Design of WIFI Module

WIFI communication is used between front-end devices and servers. The paper selects the USR-C215 module produced by You-Ren Company, and its pins are shown in Figure 7. The module is cheap and verified by a large number of entity projects. The module can be connected to the microcontroller through the serial port and support data transmission (data sent by the serial port are the data sent by the WIFI module).

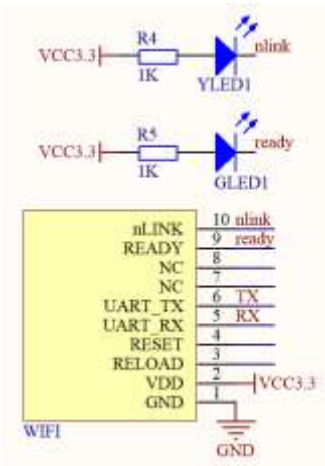


Figure 7. WIFI module

After the circuit is on electricity, the module begins to automatically connect to the preset WIFI network. After the connection is successful, both YLED1 and GLED1 will be light up. The MCU can send and receive data through the W\_TX and W\_RX pins.

4. Software Design of System

4.1. The Software Design of Front-end Device

The front-end device is responsible for not only butting the meter device and calculating the production length, but also for transmitting the data to the server steadily. Its software execution flowchart is shown in Figure 8. The whole software program is written in the C language. After the circuit is on the circuit, the program first needs to initialize the module on the hardware circuit board, including the clock frequency of the STM32, the interrupt vector initialization, and the initialization of the common IO port used by the alarm circuit and the key module. It also needs to initialize the serial port that is connected to the 485 and WIFI modules and debug the serial port. After the configuration, the user needs to input the length of production through keys. Then, the MCU begins to read the data of the meter continuously through the serial port. Normally, each data can be read successfully. If there are three failures to read the data, the alarm signal could be sent out and the cause of the alarm will be displayed on the display screen. It may be a meter damage or a problem regarding the 485 communication chip. If the data is read successfully, formula (1) must be used to integrate the read speed value  $v$  in time to obtain the length that has been produced.

$$L = \int_0^t v dt \quad (1)$$

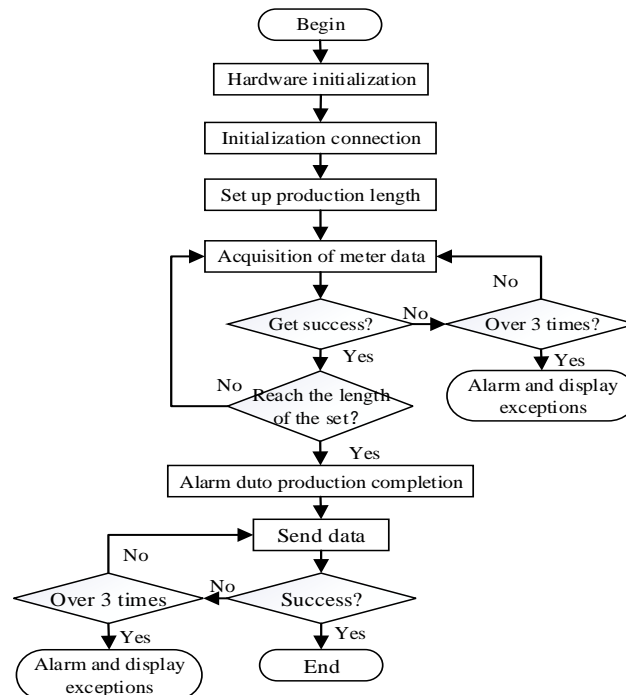


Figure 8. The flow chart of the front-end software execution

When the length reaches the value set by the user, the alarm will prompt the user that the production has been completed. The data will be displayed on the screen, and the MQTT protocol is used to send data through the WIFI network to the server. If it fails to send three times, the MCU needs to temporarily store the data in the internal storage area to prevent data loss. When the network is effective, the data will be sent again. At the same time, it will send out an alarm signal and the cause of the alarm will be displayed on the screen for relevant personnel to check.

Considering the security and correctness of data, it is necessary to encrypt and add a CRC check code before transmitting the data. Compared with the computer, the microcontroller runs slowly and cannot use complex encryption algorithms [7]. TEA (Tiny Encryption Algorithm) is a simple and efficient encryption algorithm, known for its fast encryption and decryption and simple implementation. The TEA algorithm is very simple [8]. It can operate 64-bit data at a time and use 128-bit data as a key. The algorithm is performed in an iterative manner. After receiving the data, the server first judges the correctness of the check code of the data and then uses the same key as the sender to decrypt the data to obtain the original data. Such data processing methods can prevent others from taking illegal means to steal commercial data and ensure the security and correctness of the data.

#### 4.2. The Design of the Server

The server mainly realizes the reception, storage, and forwarding of front-end device data. The background program run by the server is written in JAVA language [9-10]. The flow chart of the server software execution is shown in Figure 9. The program needs to keep running in the backstage of the server, because once the program is shut down, it will cause the front-end device to send data failure and generate alarm signals. The whole program needs to use the network. First, the socket interface needs to be initialized to access the network, and then it connects to the database to implement the database's addition, deletion, and modification [11-12]. The database chooses the MYSQL database, mainly because it is lightweight and has large storage space and wide applications. After the preparation is completed, it is always in the process of waiting for the data of the front-end device, and each data will be stored in the database to facilitate use by the client.

#### 4.3. The Design of Client Software

The general client has two architectures: B/S (browser and server) and C/S (client software and server) [13]. The former only needs the browser to access all the functions of the system. The security of the system is not high, but it is more

convenient to use without installing additional client software. The latter needs to install specially developed client software, otherwise it will not be able to connect to the server of the system, and the security is high. Therefore, this system adopts the architecture of C/S. The functional block diagram of this client is shown in Figure 10.

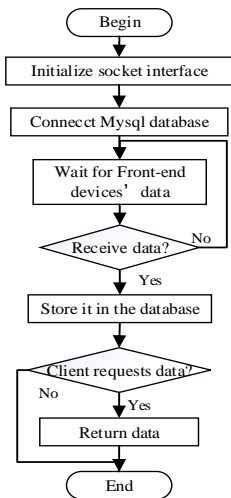


Figure 9. The flow chart of server software execution

Real-time data	Change curve
Data statistics	System configuration

Figure 10. The functional block diagram of the client

The client acquires the data that the front-end device stores on the server and displays it on the client, including the state of the front-end production line, the speed, the production length, and the time. The change curve interface of the client can make use of the historical data stored in the server database to generate the speed of the front end device and the change curve of the production length, which can help the user analyze the operation of the production line conveniently [14]. The data statistics function can generate excel files according to the time period selected by the user and the selected front-end device. The user can use the existing EXCEL editing software to process data to generate specific graphics according to their own needs [15].

5. The Test of System

The above section describes the composition and implementation process of the front-end hardware, as well as the functions of the server and the client. In combination with the above three parts, the entire system is tested below. The most important part of the whole system is data acquisition in real time, and the acquired data can be displayed and analyzed. Five front-end devices, a server, and a client were used for testing. If the system is normal, the real-time data interface of the client will display the real-time data of the five front-end devices. The login interface of the system is shown in Figure 11.

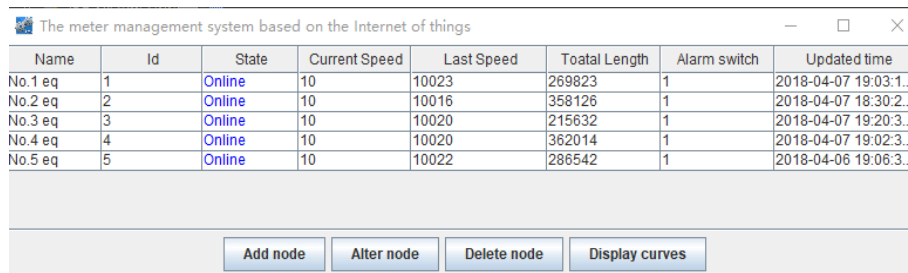


Figure 11. The login interface of the system

First, the user needs to login to the system, as shown in Figure 11, and then the user can obtain the result, as shown in Figure 12. According to the above picture, the states of five devices added by the system are all displayed normally. The interface can display the current running speed, production length, and other information of the production line under each device. At the beginning of production, the current speed can always change in real time. At the end of production, the



length of the last production and total production length can also change. In the overall process, all data are correct and the interface shows no abnormalities, achieving the purpose of this design.



Name	Id	State	Current Speed	Last Speed	Total Length	Alarm switch	Updated time
No.1 eq	1	Online	10	10023	269823	1	2018-04-07 19:03:1...
No.2 eq	2	Online	10	10016	358126	1	2018-04-07 18:30:2...
No.3 eq	3	Online	10	10020	215632	1	2018-04-07 19:20:3...
No.4 eq	4	Online	10	10020	362014	1	2018-04-07 19:02:3...
No.5 eq	5	Online	10	10022	286542	1	2018-04-06 19:06:3...

Buttons: Add node, Alter node, Delete node, Display curves

Figure 12. The interface of real-time data display

## 6. Conclusions

In view of the management needs of cable manufacturers' production information, this paper researches and implements a set of meter management systems based on the Internet of things. This system aims to monitor the production line in real time by placing a front-end device on each production line. Using the most commonly used Internet of things communication protocol MQTT, it is very convenient to send data from front-end devices to servers. The data display software uses the C/S structure to ensure the security of communication between the client and the server. The client software helps the user collect data and analyze the data, which provides important reference value for the adjustment of the next production line and the improvement of production line equipment.

## Acknowledgements

This work was supported by the Postgraduate Research & Practice Innovation Program of Jiangsu Province (KYCX17-1921) and the First Phase Project of Jiangsu University Brand Specialty Construction Project (PPZY2015B135). In addition, it was completed under the support of the Nantong University-Nantong Intelligent Information Technology Joint Research Center Open Topic (KFKT2017B05). The authors thank the three anonymous reviewers for their helpful suggestions.

## References

1. S. Avram, V. Plotenco, and L. N. Paven, "Design and Development of An Electricity Meter Test Equipment," in *Proceedings of International Conference on Optimization of Electrical and Electronic Equipmen*, pp. 96-101, 2017
2. M. Ostroverkhov, A. Silvestrov, and G. Kryvoboka, "Identification of the Parameters of the Cable Production Process," *Technology Audit and Production Reserves*, Vol. 5, No. 1(37), pp. 29-34, 2017
3. Y. Xie, M. Yu, J. Fu, and D. Chen, and C. Yang, "A Hazmat Transportation Monitoring System based on Global Positioning System / Beidou Navigation Satellite System and RS485 Bus," in *Proceedings of International Congress on Image and Signal Processing, Biomedical Engineering and Informatics*, pp. 1059-1063, 2017
4. Z. X. Yang and M. H. Zhu, "Integration Method for Wireless Communication Modes in Internet of Vehicles in the Big Data Environment," in *Proceedings of International Conference on Smart Grid and Electrical Automation*, pp. 602-607, 2017
5. Y. Ding, B. Fan, X. Kong and Q. Ma, "Design and Implementation of Mobile Health Monitoring System based on MQTT protocol," in *Proceedings of Advanced Information Management, Communicates, Electronic and Automation Control Conference*, pp. 1679-1682, 2017
6. F. Yang, C. Ma and H. Li, "Design of Wireless Video Transmission System based on STM32 Microcontroller," in *Proceedings of 11th Asian Conference on Chemical Sensors*, 2017
7. S. J. Sang, "Implementation of Cyclic Redundancy Check in Data Communication," in *Proceedings of International Conference on Computational Intelligence and Communication Networks*, pp. 529-531, 2016
8. X. Chai, K. Yang, and Z. Gan "A New Chaos-based Image Encryption Algorithm with Dynamic Key Selection Mechanisms," *Multimedia Tools & Applications*, Vol. 76, No. 7, pp. 1-21, 2016
9. S. Fukuda, S. Kurihara, S. Hamanaka, M. Oguchi and S. Yamaguchi, "Accelerated Test for Applications with Client Application and Server Software," in *Proceedings of The International Conference*, pp. 1-6, 2017
10. S. Hasan, Z. King, M. Hafiz, M. Sayagh, B. Adams, and A. Hindle, "Energy Profiles of Java Collections Classes," in *Proceedings of the 38th International Conference on Software Engineering*, pp. 225-236, 2016
11. B. Hesmans and O. Bonaventure, "An Enhanced Socket API for Multipath TCP," in *Proceedings of Applied Networking Research Workshop*, pp. 1-6, 2016
12. J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions," *Future Generation Computer Systems*, Vol. 29, No. 7, pp. 1645-1660, 2013
13. S. Y. Hao, Y. H. Qiao, Q. C. Hu, X. Y. Dai, W. J. Hao, and J. L. Liu, "Handbill Release System using B/S and C/S Hybrid Framework," in *Proceedings of Control and Decision Conference*, pp. 6789-6793, 2016

14. S. H. Park, C. S. Park, J. S. Kim, and J. G. Baek, "Principal Curve-based Monitoring Chart for Anomaly Detection of Non-Linear Process Signals," *International Journal of Advanced Manufacturing Technology*, Vol. 90, No. 9-12, pp. 1-9, 2016
15. C. Bellasio, D. J. Beerling and H. Griffiths, "An Excel Tool for Deriving Key Photosynthetic Parameters from Combined Gas Exchange and Chlorophyll Fluorescence: Theory and Practice," *Plant Cell & Environment*, Vol. 39, No. 6, pp. 1180-1197, 2016

**Zhongxing Huo** is currently a graduate student in the School of Electronic Information at Nantong University. His main research interests include the Internet of things and embedded systems.

**Yongjie Yang** is a graduate of Nanjing University of Science and Technology and the dean of the School of Electronic Information at Nantong University. He has participated in national and provincial studies, and his main research direction is in wireless communication.

**Yuelan Ji** is currently a graduate student in the School of Electronic Information at Nantong University. Her main research interests include the Internet of things and embedded systems.