Monitoring Framework for Utility Tunnels based on BIM and IoT Technology

Kai Kang[1]; Jiarui Lin[2]; and Jianping Zhang[3]

1) B.Eng., Master Student, Department of Civil Engineering, Tsinghua University, Beijing, China. Email: gkangkai@gmail.com
2) Ph.D., Postdoc, Department of Civil Engineering, Tsinghua University, Beijing, China. Email: jiarui_lin@foxmail.com
3) Ph.D., Prof., Department of Civil Engineering, Tsinghua University, Beijing, China. Email: zhangjp@tsinghua.edu.cn

Abstract:

During the operation and maintenance period of the urban underground utility tunnels (UTs), a monitoring system should be established to monitor the internal environment and the corresponding utilities for ensuring that the tunnels are running in good condition. However, existing monitoring for UTs still need improvements in data interoperability and smart sensing. This paper established a new monitoring framework for UTs based on Building Information Modeling (BIM) and IoT (Internet of Things) technology. In the proposed framework, BIM was used to maintain the 3D data of the UT, while a time series database called InfluxDB was introduced for storage of monitoring data. Then, a monitoring server based on the MQTT protocol was established, providing a low-cost and reliable communication. Meanwhile, Arduino microcontrollers with some sensors were adopted as the nodes of the monitoring network, aiming to collected real time data of the environment as well as the utilities in the UT and upload the data to the server. Finally, a prototype system was developed to demonstrate the proposed framework. Testing of the developed system and corresponding IoT sensors shows that combination of BIM and IoT technology provides a comprehensive view of the status of the UTs and improves the efficiency of information utilization.

Keywords: Utility Tunnel; BIM; IoT; Data Sensing; Data Integration

1. INTRODUCTION

The urban underground utility tunnels (UT) is an underground urban infrastructure which accommodates a variety of city facility pipelines. It has many advantages through centralized management of pipelines, such as improved underground utilization and reduced cost. The initial investment of UTs is high, but pipeline laying cost and maintenance cost will be saved. The management during the maintenance stage is very important due to its relationship to operation effectiveness, but there still remain many gaps. In order to keep the UTs running in safe, efficient and stable condition, the environment information should be monitored constantly so supervisors can take measures in time to deal with emergencies. There should be a monitoring system deployed in UTs to monitor the temperature, humidity, air condition and water level, which can alarm and control the equipment in UTs under certain circumstances.

Currently, there are already some commercial monitoring systems for utility tunnels, but there are still some problems with these existing monitoring systems (Kim, J. R. et al. 2008; Xueming, Z. 2011). Here are some future trends with possible resolution for current problems: (1) Improvement and upgrading of monitoring equipment. With the development of IoT technology, the concept of "wireless sensor network" appeared. In this pattern, monitoring devices can easily form a multi-hop ad-hoc peer-to-peer network by wireless communication method, and devices with low power consumption can use batteries as the power supply which can work for several years without charging. These devices don't require power circuits and network cables, but only regular replacement. Such a wireless sensor network has many advantages: no cable, deployment convenience, low space requirement, and high reliability. The disadvantage of the wireless sensor network is that the wireless connection is more susceptible to electromagnetic interference, and the devices with high power consumption are not suitable for using batteries. (2) Integration with GIS and BIM technology. Some existing UT monitoring systems are combined with GIS to provide an intuitive human-computer interface, which allows supervisors getting information with its location. However, there are few monitoring systems combined with BIM. The introduction of BIM allows data to be more closely tied to actual utility tunnels and pipelines, especially in some complex parts of the tunnels. This idea requires research in data integration and analysis with BIM to enable viewing data from the model and get the state from the data. (3) Information exchange and intelligence. At present, most of the utility tunnels and pipelines have their own independent monitoring systems, making it hard to communicate with each other and forming isolated information silos, which cannot be effectively linked.

In order to solve these problems in the construction and management stage of UTs, BIM and IoT technology can be a good method to improve the monitoring systems. BIM makes it possible to integrate all kinds of information in the entire life cycle of a building, with the three-dimensional visualization of building information and efficiently data sharing among participants. (Eastman, C. et al., 2011) BIM technology can effectively cope with the complexity of the operation and maintenance of utility tunnels, improve the management efficiency, and provide convenient and effective management platforms and tools. While Internet of Things (IoT) is a hot direction of the Internet in recent years, it is dedicated to connecting various objects to the internet and exchanging information between people and things. (ITU, 2005; Luigi, 2010) The IoT accomplishes the unified and integrated access of different sensors and devices through the integration of various technologies such as wireless communications, and the computer can sense environmental information and centrally manage and control equipment and machinery, thereby providing a technical foundation for monitoring and data integration of the utility tunnels. There are already some applications in environment monitoring (Lazarescu, M.T., 2013; Kelly, S.D.T. et al., 2013; Fang, S. et al., 2014).

This paper introduced a monitoring framework for utility tunnels based on BIM and IoT technology, and developed a prototype system. Arduino microcontrollers with some sensors were used as the nodes of the monitoring network, aiming to collect data and upload the data to the server. After the data is transmitted through the monitoring network, a server is established to receive, process, store and show the monitoring data with BIM data. Finally, the prototype system was tested in a prototype case and with exploration of data analysis and data visualization.

2. FRAMEWORK

2.1 Requirements for Data Management

In a BIM based utility tunnel monitoring system, data can be classified into two categories: monitoring data and BIM data. They have different requirements on data storage system.

The monitoring data are a kind of time series data produced by working sensors. The sensors in utility tunnels collect the environmental information and upload it to the server on a certain frequency. Each record contains sensor ID, monitoring type, monitoring data and other information. Its features are (1) Simple data structure. Each record of monitoring data only contains a small amount of data such as sensor ID, monitoring indicators, time stamping, and monitoring results. (2) With continuously high-frequency data writing. The monitoring system will continuously write data sent back by monitoring devices, and a large number of monitoring devices will generate large data traffic flow. Assuming that each sensor sends a record per second, and a natural gas sensor should be deployed every 15 meters for gas pipelines so there will be 667 sensors for 10-kilometer-long pipe, then one type of monitoring will produce 667 data records per second. The linear optical fiber temperature sensor, which should be laid on the power cable, has a space accuracy of 1 meter and produces a larger amount of data too. (3) Time series. Each time the data is written is the monitoring results at that time, and it will be written into the database in chronological order on a certain frequency. (4) Rarely deleted and no modified. There is no need to delete or modify a certain record. Update operation will destroy the real data. Being able to delete data in a certain time range is enough. (5) Query in order. The data query of the monitoring result is not complicated. Users only need to query the data within a certain period of time and its maximum, minimum, average, etc.

The BIM data of UTs consists of a variety of information of various components. The features are (1) A large dataset with various elements. BIM not only contains the information of structural components such as walls, floors, and doors, but also has the information of many kinds of facilities such as pipes and cables. Elements of different categories have different attributes, which make complex data types. For example, pipelines have pipeline systems, caliber, and other information, while structural components such as walls and columns do not have these properties. (2) Complex structure. There are lots of correlations between the elements. For example, the wall should be linked with the floor, and the four sides of the room will be closely related to the location and shape of the wall.

2.2 Database Selection and Design

The existing databases can be divided into two categories: relational databases and NoSQL databases. The traditional relational database is based on the relational model, using structured query language (SQL) as the query language, and suitable for storing highly structured data. Over the past decade, with the development of the information technology and industry, especially the rise of the mobile Internet, there are more complicated data types, more diversified storage requirements, and substantially increasing data volumes. The relational databases gradually show their disadvantages and become difficult to meet these new storage needs. Therefore, NoSQL databases get into a period of rapid development. Many database systems are designed for different kinds of data, and the form of data organization becomes more flexible. According to the data type and manner in data storage, NoSQL databases can be divided into the following categories: (1) document-oriented database, which organizes the data into documents without strict requirement to the structure, such as MongoDB and CouchDB, (2) key-value database, which uses key-value model and is suitable for many scenario depend on its implementation, such as Redis, (3) column oriented database, which stores data by column to get higher performance in queries by column, such as Cassandra, (4) graph database, which is designed for graph data such as Neo4j, (5) others, which contains many databases with different design targets. For example, InfluxDB and OpenTSDB are time series databases, and this type of databases are designed for time series data to achieve unique advantages over common databases (InfluxData, 2017).

According to the database categories and characteristics above, we select MongoDB, Cassandra, and InfluxDB as candidates and do some test task on them. The test scripts written in Python language connect to the databases in a Docker container through a virtual network connection, aiming to test the performance. The resource limits on each database are 2 CPU cores (2.5GHz) and 768MB memory.

The three databases run in Docker containers separately. The write scripts use 8 threads, and each one writes 100 sensors' data for 10000 times, so there will be 8×10^6 points in total. Every test data point has sensor ID, temperature, and humidity, which are randomly generated and close to the actual value. The query scripts query the sensor's all historical data for all 100 sensors in order. Because Cassandra is a little bit hard to configure and slow to write data, it didn't take all tests. The result is shown in the Table 1.

Table 1. Test results of databases

Database	MongoDB	Cassandra	InfluxDB
CPU Usage	200%	200%	80%
Memory Usage (MB)	470	580	250
Write Time (s)	248	--	1343
Write Speed (point/s)	32258	--	5957
Query Time (s)	1116	--	23.2

For time-series data, the time-series database InfluxDB has advantages in CPU usage, disk space occupied and query performance as expected. The performance of disk usage and query time is more important for monitoring tasks that generate large amounts of data, so the weight of these two aspects should be increased when choosing a database. The monitoring data of the utility tunnels is typical time-series data, and InfluxDB already has many application cases in monitoring data storage. Therefore, InfluxDB is selected as the database for monitoring data. Document-oriented MongoDB has a flexible data structure, which is suitable for complex BIM data, so MongoDB is chosen to storage BIM data.

Monitoring data is stored using the InfluxDB database. In InfluxDB's data storage model, a piece of data is stored as a point. Each point contains a timestamp, tag keys and values that contain the attributes to be indexed, and field keys and values that contain the attributes that are not indexed. Each point belongs to a group called measurement. For the utility tunnels monitoring data used in this paper, the storage scheme should be designed with the storage model of the database considered. Taking the temperature and humidity data as an example, the storage method is designed as follows. One type of monitoring data is saved in one measurement. The "sensor" attribute is the ID of the sensor, which should be indexed in the tags group, so as to obtain better query performance when queried according to the sensor ID. Temperature and humidity data don't need to index, so they are stored as field values.

BIM model data is stored in MongoDB database. MongoDB organizes data as many documents to store, and each document is in BSON format which can have multiple properties, and multiple documents compose a collection. BIM elements usually have many attributes which record all aspects of the element's information. For each attribute of the element, the attribute name can be stored as a BSON key, and the corresponding attribute value can be stored as its value in BSON in the MongoDB. The structure of the data in the database can be consistent with the structure of the BIM element, and the stored data is relatively intuitive, simple, and easy to develop and extend.

2.3 System Design

Based on requirements for UTs monitoring and IoT technology, the system architecture is designed as Figure 1, Figure 2, and Figure 3. The system has four layers. The perception layer contains the monitoring devices. The network layer provides the communication between the monitoring devices and the server. On the server, the data layer processes the data and provides basic services to the upper layer while the application layer provides functions.

When developing the monitoring device, appropriate sensors should be selected according to the monitoring requirements. The controller periodically fetches the environmental information from the sensors. Then, the microcontroller should send the data to the server that stores and processes the monitoring data through the serial port or the network. The network for data transmission between server and monitoring devices can have multiple types. In wired transmission methods, Ethernet and some serial interfaces are widely used. In wireless network types, there are Wi-Fi, IEEE802.15.4/ZigBee network and so on can be available options.

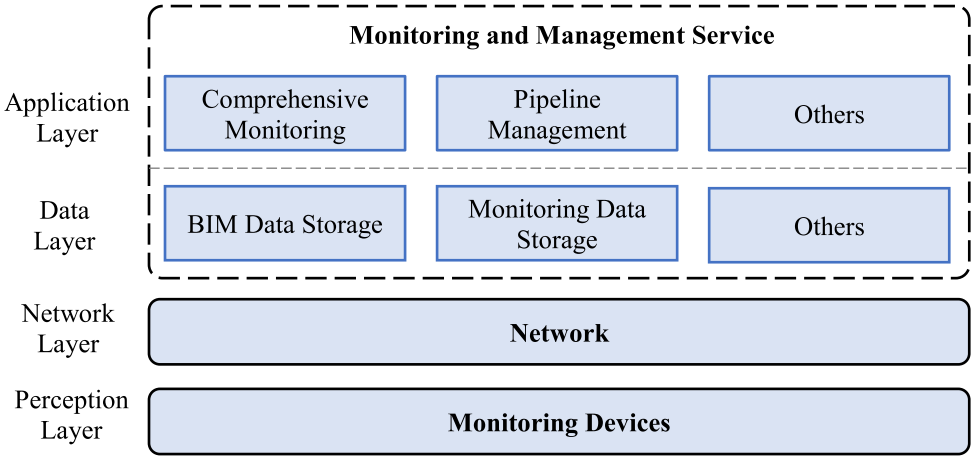

Figure 1. System architecture design

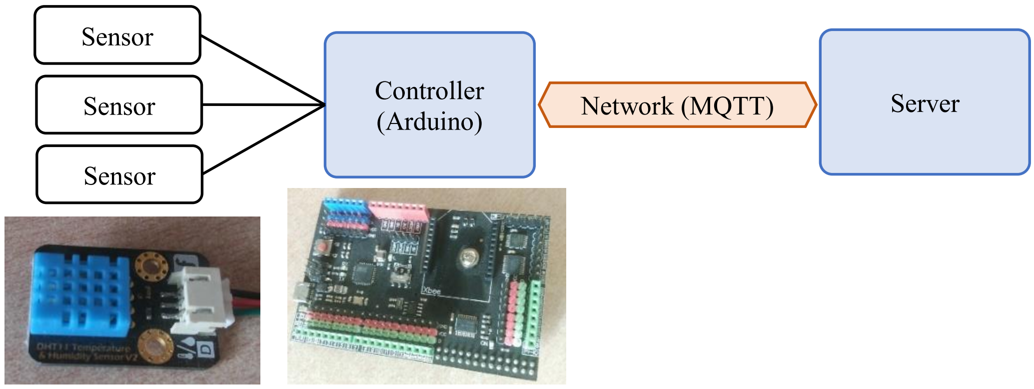

Figure 2. Monitoring device

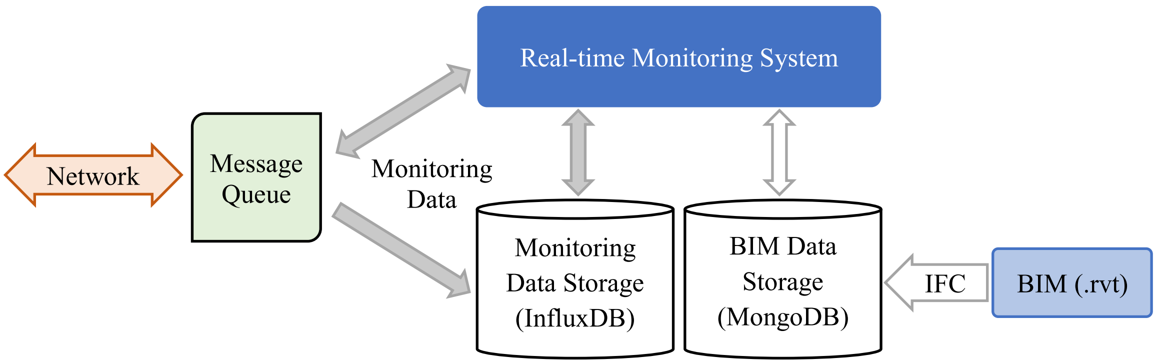

Figure 3. Server architecture design

The server-side system architecture is shown as Figure 3. The server uses a message queuing service which takes the responsibility for managing the communications with the sensor network by a certain protocol such as MQTT. When the controller uploads the monitoring data, the message queue server listens and receives the data and distributes it to all the subscribers of this topic. There are two subscribers here: the real-time monitoring system and the monitoring database. The monitoring database system is responsible for storing monitoring data, and the real-time monitoring management system is responsible for the real-time display of the current and historical environment information in the utility tunnels. If the function that sends control instructions to field devices is added in the future, the message queuing service is also capable of two-way communication. The monitoring management system is also able to query the data in the monitoring database to display and analyze historical data. The ".rvt" formatted BIM file can export the data into an IFC file and write to MongoDB following the corresponding format so that the monitoring management system can use.

3. IMPLEMENTATION

3.1 Sensor

A prototype monitoring device is developed with Arduino microcontroller as the controller, DHT11 temperature and humidity sensor and PT550 analog ambient light sensor as sensors. Some sensors for air quality and human motion are added later. Arduino is a famous open-source hardware, using Atmel AVR microcontroller, providing some basic analog and digital signal input and output interfaces. It is relatively simple and convenient to develop on Arduino which can simply connect to a variety of sensors, servo motors, and other electronic components. Arduino microcontroller can easily connect to multiple sensors due to the multiple digital and analog I/O interfaces on it, which provides a strong scalability.

DHT11 is a type of low-cost digital temperature and humidity sensor. The sensor uses NTC thermistor temperature sensing element and the resistive humidity measuring element. The temperature measurement range is from 0 ℃ to 50 ℃, and the measurement accuracy is 2 ℃. The measured humidity data is the relative humidity of the air with a resolution of 1% RH and an accuracy of 5% RH, and the measurement range is from 20% RH to 90% RH.

The ambient light sensor used by the prototype is PT550 CdS photoresistor with the ability to sense the ambient light illuminance. The measurement range is from 1 lx to 6000 lx.

3.2 Network

This system chooses MQTT as the communication protocol for monitoring data transmission as mentioned before. MQTT (MQ Telemetry Transport or Message Queuing Telemetry Transport) is a light-weight protocol that works well with low bandwidth and low power consumption. Built on the TCP/IP protocol, MQTT operates in a publish/subscribe model that provides reliable data transfer or message push at low cost in a poorly-performing network environment. These features make it ideal for the IoT environment and mobile communication environment with low bandwidth and limited computing power, so this protocol has been widely used in the field of Internet of Things and are also commonly used in M2M communication (Machine to Machine) between embedded devices.

3.3 Prototype Development

In the development of the prototype system, the program running on Arduino microcontroller is written in Arduino language based on C/C++ language, whose work is obtaining the data from the connected sensors and uploading the data to the server.

The server program system contains several parts mainly written in Python and C# language. Mosquitto is an open-source service program written in C language that implements the MQTT protocol, so it can be used as a message queue that listens and broadcasts messages. A Python program is written for receiving the monitoring data passed by the message queue service, parsing and processing the data, and writing it to the database. A plugin written in C# language is used to display the utility tunnels' BIM model and the environmental information obtained by the sensors therein. It can receive the real-time monitoring data transmitted by the message queue and read the historical data from the monitoring database.

When analyzing monitoring data, there are a variety of programming languages that can get data from the database. Here we use Python to do some analysis, and a tool called Chronograf to show the most recent monitoring results intuitively. Chronograf is a tool for managing and querying the InfluxDB database, which can display time series data in the database as charts and make dashboards with the charts.

4. DEMONSTRATION

4.1 Case Introduction

In order to test the framework and the prototype system, a few monitoring devices with DHT11 sensors and PT550 light sensors are deployed in an office as prototype case. They have accumulated nearly a month's temperature, humidity and light data, and still running with more kinds of sensors at the room corners.

The office is equipped like a small meeting room, with a table at the center and some chairs around. The office is located at Beijing with heating in winter, so there may be little change in temperature and humidity. Because the natural light condition of this office is not good, the person who is using this office will turn on the light. This room and the adjoining room are separated by a glass wall. In the adjoining room, someone will turn on its light if there are persons, which will bring some light to the office. The adjoining room is usually manned from morning to night.

On the server, a Python script subscribes to the message from the message queue Mosquitto and outputs the received message to the console, along with writing the data to the monitoring database. The Arduino controller sends temperature, humidity and illuminance data every 10 seconds. The data is sent to the message queue service via the network and then forwarded to the subscribers by the message queue service.

After this test, the Arduino-based prototype monitor and the message queue service were verified to work properly, and the database took about 11 MB of disk space after receiving such amount of data generated in nearly a month.

4.2 Data Visualization and Analysis

These figures show some of the environment data sensed by the monitoring device.

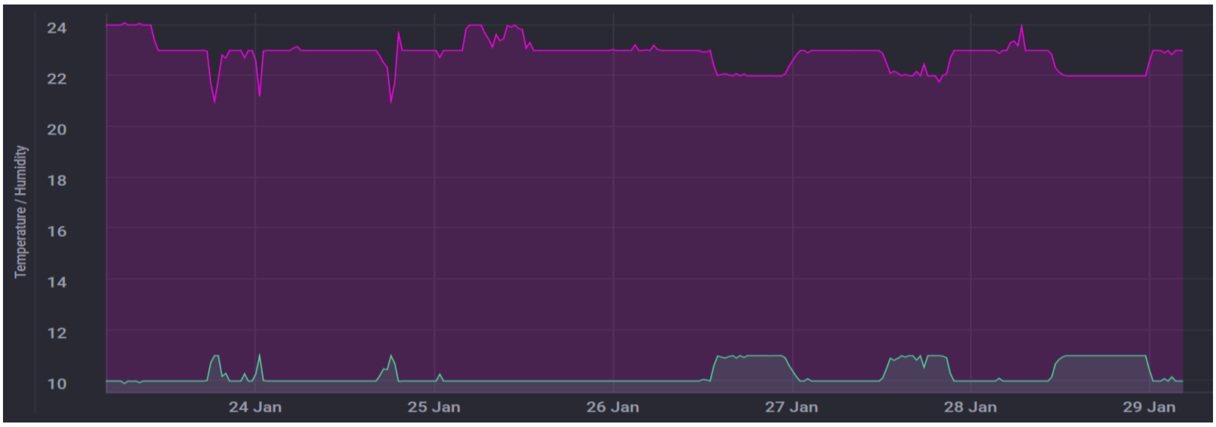

Figure 4. Temperature and humidity in past 7 days

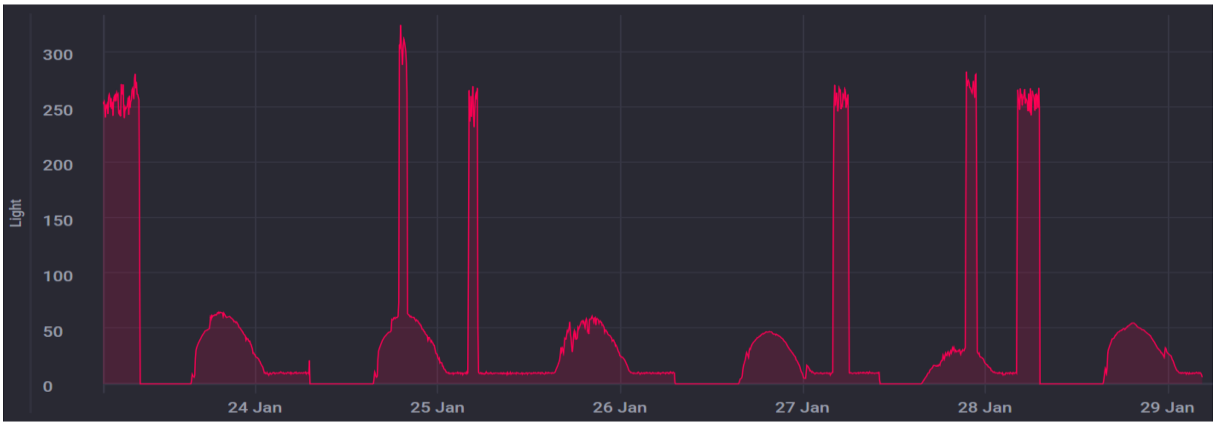

Figure 5. Light in past 7 days

The temperature and humidity have little change within a day, and the temperature is usually higher in the evening. At the beginning of the monitored period, the indoor air temperature was about 26 ℃. The daily average temperature declines over time, approaching about 23 ℃ at the end of the period. This phenomenon may be caused by the change of the outdoor air temperature. The temperature and the humidity have some relevance, but the analysis is limited by the resolution and accuracy of the sensor. There is a negative correlation between the temperature and the humidity, perhaps owing to that the temperature increase causes the relative humidity decrease with a constant water content in the air.

The light intensity curve has a smooth peak at about 50 at noon every day. It has the same pattern of outdoor light, so this part is the natural lighting effect. When some people use the office room and turn on the light, the light intensity keeps a relatively high level of around 250~300. When someone is in the adjacent office and the lamp is on, the monitoring device will report a value about 10. No light at night, the light is too weak, so the monitoring result is 0. The results are consistent with the previously analyzed office usage. In Figure 6, using smoothed z-score algorithm, it is easy to detect peak values to know whether the light in this office is on.

With the integration of the monitoring data and the BIM model, the monitoring data can be displayed in the 2D or 3D views of BIM. The room is drawn in color due to its monitoring data.

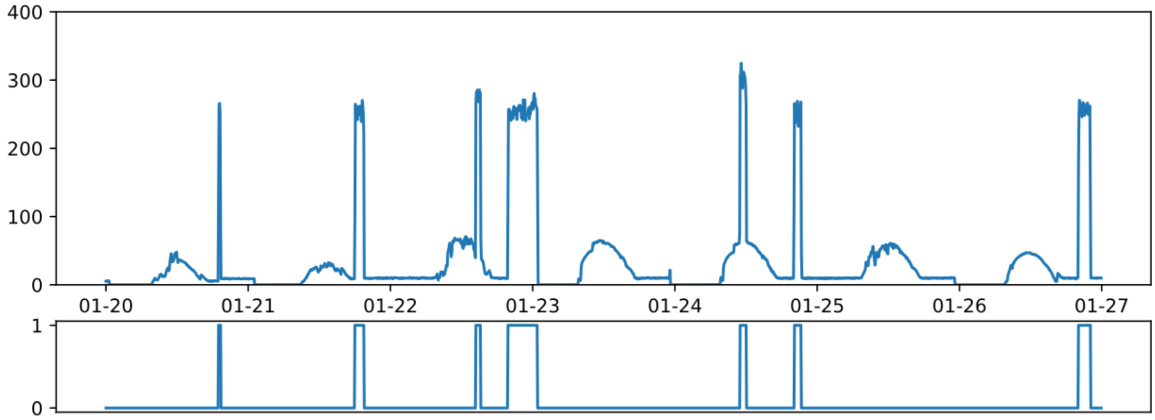

Figure 6. Light in a 7-days period and light condition detection

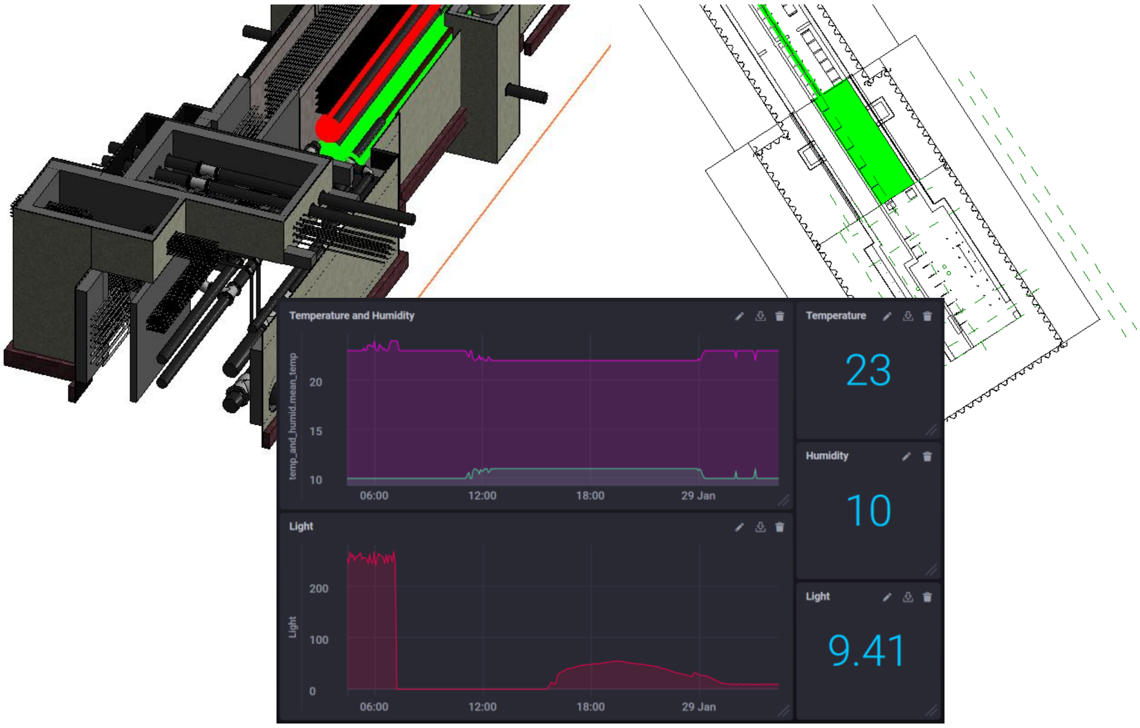

Figure 7. The monitoring data displayed in BIM model

4.3 Discussion

The prototype system works well in this case. The network can be formed wired and wirelessly. The controller sends data in JSON format by MQTT protocol. Users of real-time data can receive the data stream by registering as a subscriber at the message queue service and listening. The test shows some advantages of the framework, such as better storage performance and effective display.

The framework tested in an office can also apply to utility tunnels monitoring, though the working environment is a little bit different between an office and tunnels. In Table 2, the monitoring items in the office are similar to those in UTs, and the frequencies may have some differences. The UT environment has higher frequency requirement on some items. The framework is focused on the data storage, integration, and usage, while the differences are not so important that can affect the performance and advantages of the framework. After all, for special circumstances, the devices can be replaced by other devices with same interface and conduct.

The framework and the prototype have good scalability. Adding more sensors is simple due to the multiple digital and analog I/O interfaces on Arduino controllers. In addition, the monitoring database doesn't need to change the data scheme like SQL databases when it's extending.

Table 2. Some differences between the office and UTs

	Office	UT	Data type	Office frequency	UT frequency
Temperature	●	●	int / float	10 s	10 s ~ 1 min
Humidity	●	●	int / float	10 s	10 s ~ 1 min
Gas	○	●	float	--	5 s ~ 10 s
CO_2	○	○	float	--	10 s ~ 1 min
Light	●	●	int	10 s	10 s ~ 1 min
Water level	○	●	float	--	10 s ~ 1 min
Person	○	●	bool	--	10 s ~ 1 min
…	…	…	…	…	…

Mandatory: ● optional: ○

5. CONCLUSIONS

The urban underground utility tunnels need monitoring systems, but the existing systems have disadvantages in devices, architectures, and integration. This article presents a monitoring framework for utility tunnels based on BIM and IoT technology and tests a prototype. We use BIM to maintain the 3D data of the utility tunnels, and select a time series database InfluxDB for monitoring data storage after comparison. A few prototype monitoring devices are developed with Arduino microcontrollers and sensors. A prototype system is developed and tested in an office, and it works as expected and provides a monitoring service for the office. The analysis of monitoring data can find certain rules and patterns, therefore contributing to knowledge discovery.

Testing of the prototype system and corresponding IoT sensors in a prototype case shows that combination of BIM and IoT technology provides a comprehensive view of the status of the UTs and reveals potential for improvement of work efficiency. The monitoring data can be visualized, used statistically, and stored effectively. The above has verified the technical feasibility, but there are still some problems in the current system, such as few monitoring items and some missing functions. However, this framework is very scalable and can be easy to add monitor items and functions, which can be achieved in the future.

ACKNOWLEDGMENTS

This research is supported by the National Key R&D Program of China (No. 2017YFC0704200), the China Postdoctoral Science Foundation Grant (No. 2016M601038) and the Young Elite Scientists Sponsorship Program by the China Association for Science and Technology (No. YESS20160122).

REFERENCES

Eastman, C., Teicholz, P., Sacks, R., and Liston, K. (2011). *BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors* (2nd ed.). John Wiley & Sons.

Fang, S., Da Xu, L., Zhu, Y., Ahati, J., Pei, H., Yan, J., and Liu, Z. (2014). An Integrated System for Regional Environmental Monitoring and Management Based on Internet of Things, *IEEE Transactions on Industrial Informatics*, 10(2), 1596-1605.

Luigi, A., Antonio, I., and Giacomo, M. (2010). The Internet of things: A survey, *Computer Networks*, 2784-2805.

Lazarescu, M.T. (2013). Design of a WSN Platform for Long-Term Environmental Monitoring for IoT Applications, *IEEE Journal on Emerging & Selected Topics in Circuits & Systems*, 3(1), 45-54.

InfluxData (2017). *InfluxDB Documentation*. Retrieved from InfluxData website: https://docs.influxdata.com/influxdb/latest/, accessed on May 1, 2017.

International Telecommunication Union. (2005). *Internet Reports 2005: The Internet of Things*, ITU.

Kelly, S.D.T., Suryadevara, N.K., and Mukhopadhyay, S.C. (2013). Towards the Implementation of IoT for Environmental Condition Monitoring in Homes, *IEEE Sensors Journal*, 13(3), 3846-3853.

Kim, J. R., Yoo, H. S., Kwon, S. W., and Cho, M. Y. (2008). Integrated tunnel monitoring system using wireless automated data collection technology. *Proceedings of the 25th International Symposium on Automation and Robotics in Construction.* Vilnius: Vilnius Gediminas Technical University. Publishing House (pp. 337-3).

Xueming, Z. (2011). Design of Utility Tunnel Monitoring System in Expo Park. *Modern Architecture Electric*, (4), 21-25.