東海大學

資訊工程研究所

碩士論文

指導教授: 楊朝棟博士

用於醫療保健的智能環境監測系統之實作

On Construction of an Intelligent Environment Monitoring System for Healthcare

研究生: 廖啟瑞

中華民國一零二年七月

摘　要

隨著衛生環境的改善、生活型態的轉變，健康促進與預防醫學是現代人們所注重的兩個觀念。配合現在現時資訊科技改善與思維轉變，如雲端技術、普及運算、Web 2.0 等。希望由建康促進與預防醫學的觀點來達成流程的改善，更能對環境的監控並提出預測與警告。因此我們提出了一個建構智慧型環境監控回饋系統及平台的方法，來長期收集工作環境空氣資料。並對於資料的分類歸檔、建立查詢介面等根據需求給予專業意見。另外對於特定需求的檔案，針對其所需資料建置格式轉換功能以降低資料交換的難度。對於該環境空氣資料的監視做專業判斷，然後依有重要影響的因素，自動發出示警的通知，以期能達成提前的預防。預防措施參考行政院環境保護署提出的室內空氣品質標準來做實驗標準。此外，也提供手機用戶端可以直接利用 App 即時監控整個系統所收集到的資訊，並提供即時的解決方案來因應特定空氣資訊到達臨界點的處理措施，運用本系統可有效掌握該區域的二氧化碳、溫度、濕度等多種訊號，並達到人員安全的保障及節省大量人力資源的耗費，測試結果顯示本系統可即時顯示二氧化碳溫溼度狀況並回傳信號執行遠端警示及控制。

Keywords: ZigBee, 環境監控, 物聯網, MySql, 無線感測網路, Android

Abstract

With the improvement of sanitary conditions and changes of lifestyle, people start paying attention to the two modern concepts of health promotion and preventive medicine. In combination with the existing information technology services and innovations, such as the grid technology, pervasive computing, WEB2.0, in this thesis, an integrated plan is proposed to achieve process improvement, environmental monitoring, forecasts and warnings for the aims of health promotion and preventive medicine. Therefore, an intelligent environment monitoring feedback system and platform is constructed to gather information about long-term hospital staff health and the air data of the working environment. For information classification and input data into the database system, we create a query interface on demand for professional advices. In addition, the format conversion capability is provided for easy exchange of information. Based on the real time ambient air monitoring information in the hospital, and according to the factors that have major impacts from the professional judgments, warning notices or signals are sent with a view to reaching early prevention; in this way, the hospital can respond in time to reduce unnecessary losses. Also, the monitoring system is able to improve the safety and security of the working environment for the medical staff, resulting in enhancement of the job efficiency.

Keywords: ZigBee, Environment Monitoring, IOT, MySql, Wireless Sensor network, Android

致謝詞

終於到了撰寫致謝詞的時刻，半年多的蒐集、閱讀資料及架設環境，兩個多月的撰寫時間，完成了這一篇的碩士論文，這是一個里程碑，是終點也是另一個起點。它代表了我們像這段學習生活的告別，也意味著我們未來將開始承擔不同的角色及責任，開始新的人生旅程！這篇論文的完成，絕非我個人獨力可達成的目標，它含藏著太多人無論精神與物質的關懷與幫助。

首先誠摯的感謝指導教授楊朝棟博士，楊教授悉心的教導使我得以一窺網路以及雲端運算領域的深奧，並讓我嘗試環境監控的這個火熱的議題，在我學習的過程中不時的督促及討論且指點我正確的方向，使我在這些年中獲益匪淺。老師對學問的嚴謹更是我輩學習的典範。

再來要感謝的是各位協助我論文更加完美的各位口試委員，感謝本系劉榮春教授對我研究上的指教評點，您讓我了解如何讓我的分析方法更加完善，並進一步給我想法及鼓勵。感謝本系朱正忠教授在口試場上用您精湛的外文能力點出我的論文語意，使其更加流暢。也感薛念林教授、李冠憬教授不辭辛苦地前來東海當我的口試委員，為我的論文提出了許多的建議。因為有你們的體諒及幫忙，使得本來鬆散的文章結構，在重整之後能夠更完整而嚴謹。

本論文的完成另外亦得感謝高效能實驗室的各位學弟妹及學長姐的大力協助，尤其是宋允文學長，在我剛進入實驗室時，是由你來帶領我學習及了解實驗室的研究領域，跟著你慢慢學習後，使我從 Java、Android 到環境監控越來越得心應手，王顯義學長不厭其煩的指出我研究中的缺失，葉威廷學長不吝嗇地讓出它的設備讓我建置我的實驗環境。呂欣汶學妹給我好的點子突破在實驗過程中遇到的瓶頸並協助處理實驗室的大小事務，幫我們減輕了不少負擔，這篇論文的完成要特別感謝吳佳澄學弟，沒有你的幫忙我就沒辦法完成這份論文，

另外還有更多更多的人要感謝，他們不只是在論文上，更是生活上的幫助，使我可以順利的完成論文。

而這兩年一起成長的朋友兼同學們文鴻、育佐、韋伸、冠龍，恭喜我們順利走過這兩年，在學校兩年的研究生活，給我留下太多難忘的回憶，在這裡，我們努力過，一起度過數不清的 meeting、報告及考試，一起熬夜讀書與早起吃炒麵，一起互相扶持及討論，大家的陪伴讓我在讀書之餘，還衝喊許多的快樂，讓我在開始進入論文撰寫時，對於即將要脫離的學生生活沒有遺憾。

謝謝爸爸媽媽從小辛苦的栽培，讓我可以完成我的求學旅程；謝謝女友歆雅的陪伴，在我求學生涯遇到不愉快的事，妳都願意聽我訴說以及給我鼓勵。

最後，謹以此文獻給一直支持我，我所摯愛的雙親、爺爺奶奶、兩個妹妹及女友歆雅。

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Day by day, medical staff must face with patients coming from all areas, taking the risk of multiple infections in the complicated environment. In addition, extreme pressures caused by quality of the working environment, overtime and high risk works affect efficiency of workers, increase the occupational injury rate and the staff turnover rate in a period of time. This research chose a medical center as the study target. Through real-time sampling, data of the air pollution level and other related factors in the medical center were recorded and analyzed to adaptively adjust the air quality in the area. .

In the meantime, we also surveyed by questionnaires on employee’ s state of health, working environment awareness, and psychological pressure to discover the connection among environmental, psychological, and physical factors. Then, we can improve the quality and health of the whole staff by human factors engineering, job description designs, and active heath cares. Besides, section supervisors of the hospital can prevent medical staff and patients from contaminating diseases by implementing a real-time monitoring and feedback system to identify problems in time. To design and implement the medical information system, the key point is how to formulate and maintain a unified data format. However, in recent years, hospitals begin to think in the views of patients. They provide major medical treatments and other related integrated services, such as the remote health care service net, by which medical staff members inquire and analyze physical information by desk-tops or mobile computers and the Internet, and provide immediate advices, revisit notes, or abnormal alerts to patients. We hope our innovative system can help forward development of integrating applications of medical, information, and communication technologies to develop a full solution of a multi-health care service for patients with urgent health care needs or chronicle diseases.

As pointed by one professor of National Taiwan University in 1992, Taiwan’s top 4 causes of death were Health care system (9.9%), biological factors (25.2%), environmental factors (19.4%), and the most important one: lifestyle (45.5%). Currently, it is no doubt that professional staff members in hospital and medical center have already striven to take forward the health lifestyle to help people promote self-care ability. However, the environmental factors need to be monitored and analyzed in a long period.

Recent trend is to include the concept of pervasive computing into all kinds of medical systems to provide integrated services. Pervasive computing focuses on human ideas, and it integrates many fields of technologies to assist human activity anytime and anywhere to let people target their purpose without waste in unnecessary processes. Thus, this project includes a wide range of technologies, including artificial intelligent (AI), embedded systems, human-machine interaction, distributed computing, cognitive networks, and mobile computing.

As the hottest application, cloud computing also plays a big role in our design. For a system with complicated structure to implement tremendous amount of computing and operations, it requires a lot of resources. Grid computing is the federation of computer resources from multiple locations to reach a common goal [7, 8, 9]. This technology not only improves computation efficiency, but makes a more complete use of remaining resources, and decreases the total cost.

Because of complex features in a medical environment, it demands high quality of air as well as a comfortable working place. An ideal indoor air system must adjust air every moment to prevent people from all kinds of air pollutions like dusts or viruses. Poor indoor air quality (IAQ) may cause serious impact on human health and the living environment. As a result, Environmental Protection Agency (EPA) recommends values of indoor air quality, including CO_2, CO, HCHO, TVOC, Bacteria, Fungi, PM_10, $PM_2.5$, and O_3. EPA also sets a strict standard for some special buildings such as hospitals.

1.2 Thesis Contributions

In addition to monitoring of the environment, we focus on the detecting and analyzing air quality in medical areas such as outpatient, emergency, ward, and administrative areas, to construct front ends of the whole system. It also can be further used to identify origins of air pollutions, record the current state of the air quality, provide medical staff with analysis support, and build a database of employee health.

In the meanwhile, we have constructed an information platform to provide necessary features, such as electronic health records storage and air monitoring, and archived data collected by our devices in a safe and standard format. Through real-time monitoring system, the administrators can learn the air quality in the whole area, and notify department members by communications systems to decrease possibilities of the infection rate. On the other side, the platform also has machine learning ability, provided with data collected and analyzed by field detection of medical facilities. Besides monitoring functions, the proposed platform can be used to prevent outburst of major diseases.

1.3 Thesis Organization

In Chapter 2, we describe the used techniques and some background knowledge. Chapter 3 describes the system architecture. Chapter 4 shows experimental results for the proposed system. Chapter 5 provides conclusions and future work of this work.

Chapter 2

Background and Related Work

2.1 Environment Monitoring

Environmental monitoring processes the need to identify and monitor of environmental quality activities in the preparation of an environmental impact assessment, and in many cases, the risk of the harmful effects of human activities on the natural environment. Monitoring strategies and programs are often designed to establish the current state of the environment or to establish the causes and reasons of the trend of environmental parameters. In all cases, the results of the monitoring will be reviewed, statistically analyzed and reported. The monitoring programs are designed to process monitoring data in real time.

Monitoring is of little use without a clear and unambiguous definition of the reasons for the monitoring and the objectives that it will satisfy. Almost all extensive and invasive monitoring (except perhaps remote sensing) research has risks of damage to the environment if not planned well before hand. It may harm rare species in wilderness areas or make the monitored people feel uncomfortable, which is an important factor to consider. Monitoring technologies, such as network gills to estimate population, can be very harmful, at least to the local residents and scientists to carry out monitoring, and can also reduce trust of the public.

Almost all mainstream environmental monitoring project forms an overall monitoring strategy or field of study of these areas and strategic goals or aspirations from high levels of an organization [10]. Unless the individual monitoring programs are uploaded into broader strategic frameworks, the result is unlikely to be published to help people understand the monitoring environment.

2.2 ZigBee

ZigBee is a specification for a suite of high level communication protocols using small, low-power digital radios based on an IEEE 802.15.4 standard for personal area networks. ZigBee devices are often used in mesh network form to transmit data over longer distances, passing data through intermediate devices to reach more distant ones. This allows ZigBee networks to be formed ad-hoc, with no centralized control or high-power transmitter/receiver able to reach all of the devices. Any ZigBee device can be tasked with running the network.

ZigBee is targeted at applications that require a low data rate, long battery life, and secure networking. ZigBee has a defined rate of 250 kbit/s, best suited for periodic or intermittent data or a single signal transmission from a sensor or input device. Applications include wireless light switches, electrical meters with in-home-displays, traffic management systems, and other consumer and industrial equipment that requires short-range wireless transfer of data at relatively low rates. The technology defined by the ZigBee specification is intended to be simpler and less expensive than other WPANs, such as Bluetooth or Wi-Fi.

ZigBee protocol layers from bottom to top are the physical layer (PHY) and media access (MAC) layer, network layer (NWK), application layer (APL). The role of the network devices can be divided into ZigBee Coordinator, ZigBee Router, and ZigBee End devices. Figures 2.1 shows a ZigBee Coordinator used in the experiment.

FIGURE 2.1: ZigBee Coordinator

2.3 Internet of Things (IOT)

Internet of Things (IOT) means a great integration including ubiquitous devices and facilities, such as build-in intelligence like sensor, mobile terminal, industrial system, floor control system, home intelligent facilities and monitoring system, or enabled device like every kind of assets on RFID label, object capable with wireless terminal, moteetc. With Web 4.0, IOT reached M2M, Grand Integration/ MAI and cloud computing-based SAAS. In Intranet, Extranet or Internet form, IOT uses appropriate information security system, providing safe, guided and even individuated real time monitoring, capable to achieve everything high efficiency, high saving energy, high security and environmental protection in an integrated service. IOT and its related Taas service compose the major structure of Web 4.0 based on Web 3.0.

As shown in Figure 2.2, the structure of Internet of Things architecture can be divided into three layers. The lowest level is the perception layer. It consists of a variety of information capturing and identifying sensing elements. The middle of the network layer consists of all kinds of wireless transmission technology. The top level application layer of Internet of Things consists of a variety of applications, such as environmental monitoring and urban management. The application support layer, a sub-layer between the network layer and application layer, is primarily responsible for providing various types of platforms, series transmission network and application services.

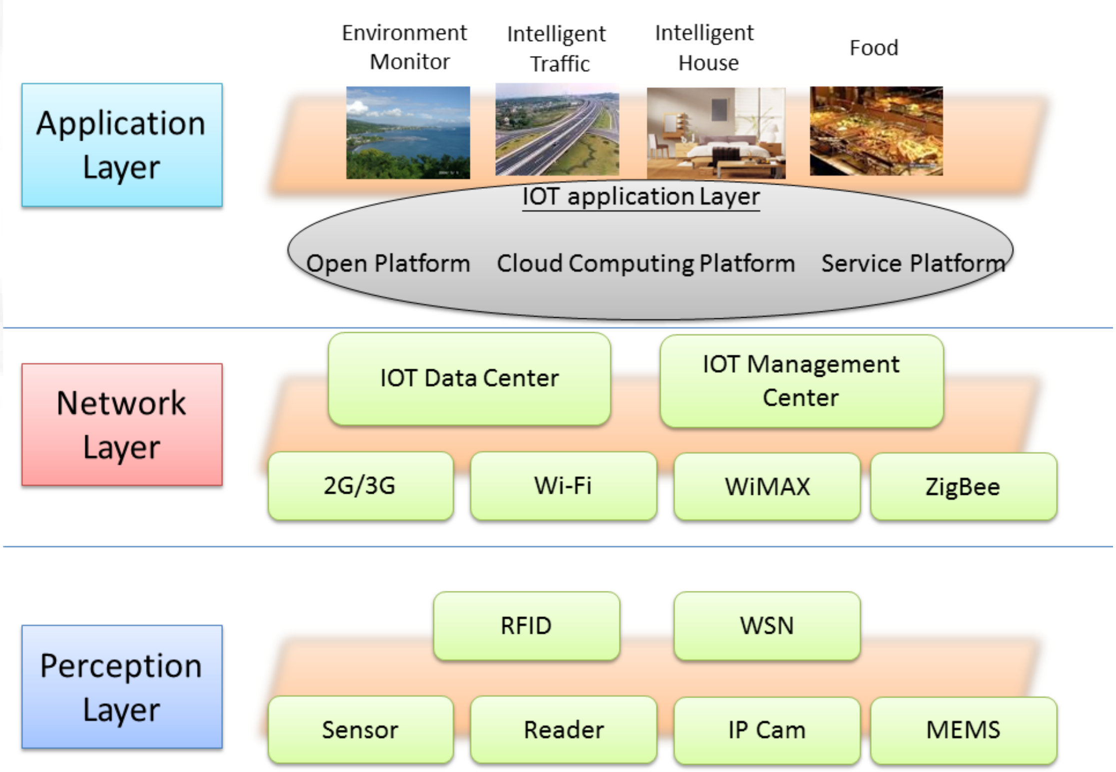

FIGURE 2.2: Internet of Things Architecture

2.4 Cloud Computing

2.4.1 Service models

Cloud computing is a computing approach based on the Internet. In this way, resources can be shared by the required hardware and software available to computers and other devices. Users no longer need to understand the ”cloud” in the details of the infrastructure, do not possess the necessary professional knowledge, without direct control. Cloud computing describes a new Internet-based services to increase IT use and delivery models, usually involving the Internet is easy to provide dynamic and often is a virtual extension of the resource. The cloud is network, Internet a metaphor. Cloud computing can be considered include the following levels of service: infrastructure as a service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). Figure 2.3 shows its architecture.

- Infrastructure as a Service (IaaS): Users can follow the required level of computer and network equipment and other resources, to the service provider subscription service, and may require changes to settings, and service provider by users of the CPU, memory, Disk space, network load to calculate the costs.
- Platform as a Service (PaaS): development of services vendors who rented to a computer, this computer has all the necessary hardware and software developers environment; or to provide application developers to market, in accordance with the amount of traffic with the use of resources Developer fees.
- Software as a Service (SaaS): the software stored in the data center to provide users network access services, according to period or pay-per-order the type of charge.

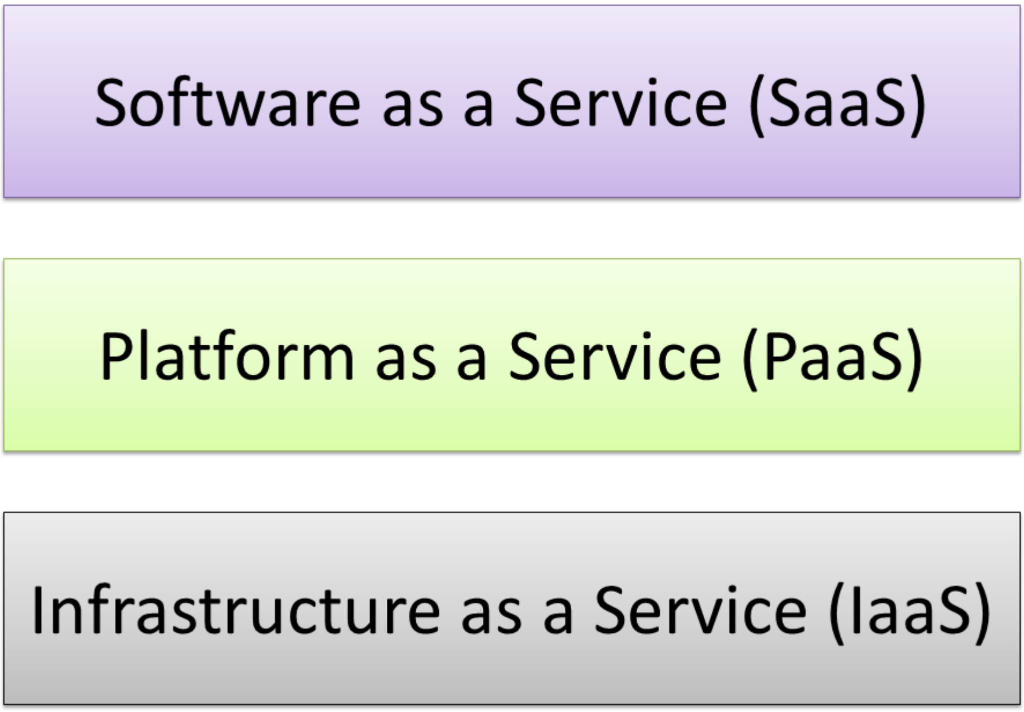

FIGURE 2.3: Cloud Computing Service models

2.4.2 Deployment models

- Public cloud: Public cloud applications, storage, and other resources available to the public by the service provider. These services are free or pay-per-use mode. In general, public cloud service providers Amazon AWS, Microsoft and Google and other infrastructure own and operate, and provide access only via the Internet of Things. Issues such as security, compliance, and judicial are common concerns in the community cloud shared between several organizations. From a particular social infrastructure, it is managed or hosted by a third party, internal or external. It has user cost fewer than a public cloud, but more than a private cloud, so only some of the potential cost savings of cloud computing.

- Hybrid cloud: Hybrid cloud refers to two or more clouds (private, community or public) remains the only entity [11], but combined together to provide the benefits of multiple deployment modes. Figure 2.4 shows deployment models. This combination of expansion of cloud services deployment options, enables IT organizations to use public cloud resources to meet temporary needs. This capability allows the hybrid cloud zoom across the cloud Cloudbursting. The increasing demand for computing power in the cloud burst, an application the application is running in a private cloud or data center and "burst" to public cloud deployment model. The blasting is a major advantage of the cloud and hybrid cloud model; organizations only need to pay the additional computing resources they need to. Cloud burst data center to create an internal IT infrastructure to support the average load and the use of public or private cloud cloud resources, during peak processing needs. Through the use of "hybrid cloud" architecture, businesses and individuals are able to obtain immediately with local Internet connection without relying on the availability of fault tolerance. Hybrid cloud architecture requires real resources and remote server-based cloud computing infrastructure. Hybrid cloud lacks of internal application flexibility, security and certainty. The hybrid cloud provides fault tolerance, flexibility and scalability of cloud-based services to internal applications.

- private cloud: The private cloud is a cloud infrastructure operations only for a single organization, whether it is internal management or hosted by a third party, internal or external. Private cloud project requires a significant level of participation, the virtualization business environment, and require organizations to re-evaluate the existing resources of the decision. It can improve the business, but at every step of the project, security issues must be addressed in order to prevent serious vulnerability. It attracts criticism because users can still buy, build, and manage things; therefore benefiting from the hands of less management. In essence, it is "the lack of economic model, makes cloud computing such an interesting concept.

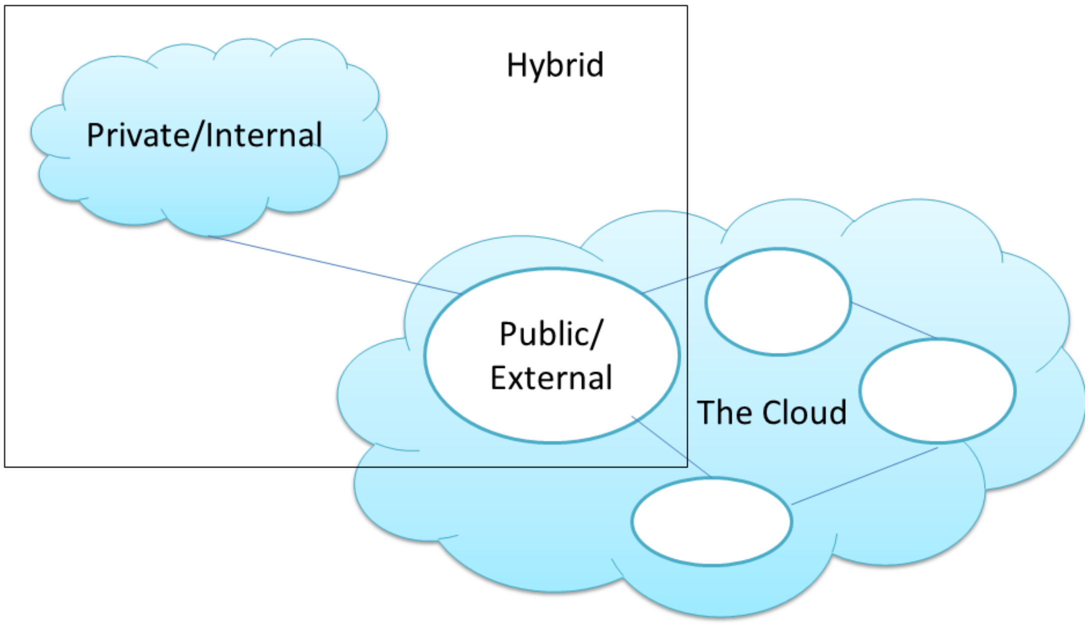

FIGURE 2.4: Cloud Computing Deployment models

2.5 Apache server

The Apache HTTP Server, commonly referred to as Apache, is a web server software program notable for playing a key role in the initial growth of the World Wide Web. In 2009 it became the first web server software to surpass the 100 million website milestone. Apache was the first viable alternative to the Netscape Communications Corporation web server (currently named Oracle iPlanet Web Server). Typically Apache is run on a Unix-like operating system, and was developed for use on Linux. Apache is developed and maintained by an open community of developers under the auspices of the Apache Software Foundation. The application is available for a wide variety of operating systems, including Unix, FreeBSD, Linux, Solaris, Novell NetWare, OS X, Microsoft Windows, OS/2, TPF, and eComStation. Released under the Apache License, Apache is open-source software. Apache was originally based on NCSA HTTPd code. The NCSA code has since been removed from Apache, due to a re-write. Since April 1996 Apache has been the most popular HTTP server software in use. As of December 2012 Apache was estimated to serve 63.7% of all active websites and 58.49% of the top servers across all domains.

2.6 MySQL

MySQL is the world' s most used open source relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases. It is named after co-founder Michael Widenius' daughter, My. The SQL phrase stands for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation. MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open source web application software stack (and other ' AMP' stacks). LAMP is an acronym for Linux, Apache, MySQL, Perl/PHP/Python. Free-softwareopen source projects that require a full-featured database management system often use MySQL. For commercial use, several paid editions are available, and offer additional functionality. Applications which use MySQL databases include: TYPO3, Joomla, WordPress, phpBB, MyBB, Drupal and other software. MySQL is also used in many high-profile, large-scale World Wide Web products, including Wikipedia, Google (though not for searches), Facebook, Twitter, Flickr, Nokia.com, and YouTube.

MySQL Cluster [12] for MySQL [13] is a distributed computing environment of high practical, high-redundancy version. It uses the NDB Cluster storage engine, allowing a cluster to run multiple MySQL servers. The storage engine is provided in the binary version of MySQL 5.1, and RPM compatible with the latest Linux version provides. (Note that, in order to get MySQL cluster feature, you must install the mysql-server and mysql-max RPM).

Operating systems able to run the MySQL cluster include Linux, Mac OS X and Solaris. Some users informed of successfully running FreeBSD on MySQL Cluster, but the MySQL AB Company has not officially supported this feature. We are making efforts to get MySQL Cluster to run in all operating systems supported by MySQL, including Windows. There are three types of cluster nodes for the lowest MySQL Cluster configuration. These three types of nodes are

- Management (MGM) node: the role of such nodes is to manage other nodes inside the MySQL cluster, such as nodes providing configuration data, starting and stopping, or doing the backup. As these nodes manage the configuration of the other nodes, they should be started before the other nodes. MGM node is evoked by the ndb_mgmd command.
- Data node: This kind of nodes is used to store data in the cluster. The number of data nodes is the same as the number of replicas and is multiple of fragments. For example, if there are two replicas and each replica has two fragments, then there are four data nodes. It does not need to have more than one replica. Data node is evoked by using the ndbd command.
- SQL node: This kind of nodes is used to access cluster data nodes. For MySQL Cluster, SQL nodes are conventional MySQL servers that use NDB Cluster storage engines. Typically, SQL node is evoked by using the mysqld-ndbcluster command, or by using mysqld with adding ndbcluster option to my.cnf.

Management servers (MGM nodes) administer the cluster log and the cluster configuration file. Every node within the cluster receives configuration data from the management server and needs a way to locate the management server. The data node will transmit messages of incidents in it to the management server to have the information written in the cluster log.

Cluster Configuration of the cluster engages configuring individual nodes in the cluster, and setting up separate communication links between nodes. For the current design of the MySQL cluster, its intention is to have homogeneous storage nodes, in terms of the processor capacity, memory space and bandwidth. Besides, to have a single point of configuration, all cluster configuration data are located in the same configuration file.

Besides, one can have any number of cluster client processes or applications of these two types:

- Standard MySQL client: MySQL Cluster is not different from the standard (non-clustered category) MySQL. In other words, one can use PHP, Perl, C, C++, Java, Python, Ruby and other existing MySQL applications to access MySQL Cluster.
- Management client: This type of clients are connected to the management server, and they provide elegant ways to start and stop nodes, backup commands, and message tracing (debug version only), and to show node versions and status.

2.7 Sensor Network

In this thesis, there are some information comes from Sensor Network, Sensor network is composed of spatially distributed automatic device consisting of a computer network, these devices use sensors to monitor different locations collaborate physical or environmental conditions (such as temperature, sound, vibration, pressure, motion or pollutants). Sensor network with each node in addition to one or more sensors, but also equipped with a radio transceiver, a microcontroller and a very small energy (usually battery). In recent year, ZigBee is popular for sensor network. Many people use ZigBee [14, 15] to deploy their sensor network in an environment. This system uses the ZigBee wireless network, also. ZigBee is a wireless network protocol, mainly by Honeywell's development component of the ZigBee Alliance, the development since 1998. The underlying IEEE 802.15.4 [16] standard specification is the use of the media access layer and physical layer. Main features are speed, low power, low cost, support a large number of network nodes to support multiple network topologies, low complexity, fast, reliable and safe.

2.8 Related Work

Environmental monitoring is more talked about than felt in our everyday lives, there will have big changes for health care and quality of the environment when we have pervasive real-time analytics of sensor data.

In recent years, some scholars implements a fast disposing wireless sensor network (WSN) with low power consumption. The system, which is based on ZigBee technology, detects environment information, such as temperature, relative humidity and pressure etc [17]. By the portable structure design, the system can be established or dismantled quickly in a desired area, such as a complete building, a temporary watched environment and emergency scene, etc.

Other presents a wireless sensor network project based on ZigBee technology to be used in greenhouse environmental monitoring [18]. Through the theoretical analysis and experimental environment test, it has been confirmed that the design of greenhouse environmental monitoring system is reasonable, the hardware system performance normal, and the software design is user-friendly that the data can be analyzed and monitored from anywhere with sensors.

The ZigBee wireless sensor networks in environmental monitoring applications is introduced in [19]. The hardware design includes microprocessor, data acquisition, antenna and peripheral circuits of the chips; and through software design composed ZigBee mesh network can do data acquisition and communication. This network has low power consumption, low cost, the effective area is big, and information transfers reliable merits. We have confirmed the network's communication applicability by the Serial Com Assistant, also testified that the network have very good pragmatism by the NS2 emulation of the network's operation.

In the health care field, personal health assessment has been a hot issue today. In response to this trend, many countries make great effort to combine different technologies and knowledge to build interactive health care systems [20]. As shown in research papers related with medical service, personal health advice and management occupy important parts of the whole medical system, and also attract great attention in research. Implementation of such a medical information system is not only a trend, but a key point in future Decision Support Systems. With the help of the advisory and management integrated health care system and personal medical data platform on mobile devices, medical staff can provide advices and services to anyone, any time, at anywhere to achieve the goals of disease prevention and health promotion. With cloud computing, we can implement more effective data exchange features and computing power. Moreover, with highly integrated and friendly interface, users can be aware of their health status in a great ease.

Chapter 3

System Design and Implementation

3.1 System Architecture

In this paper, we implemented a service platform designed to monitor air quality in a clinical center.

The platform was built with a distributed storage, but with a single entrance as wireless / wire sensors data integration to let user access these data. It supports the HL7 format in data exchange and collection, and transmits in the XML format to increase the consistency and readability [21]. Our system architecture is shown in Figures 3.1 and 3.2.

3.2 System Setup

Figure 3.3 shows a scenario of service from system to user. On the host server we wrote codes to monitor the collected data. An analysis method is programmed

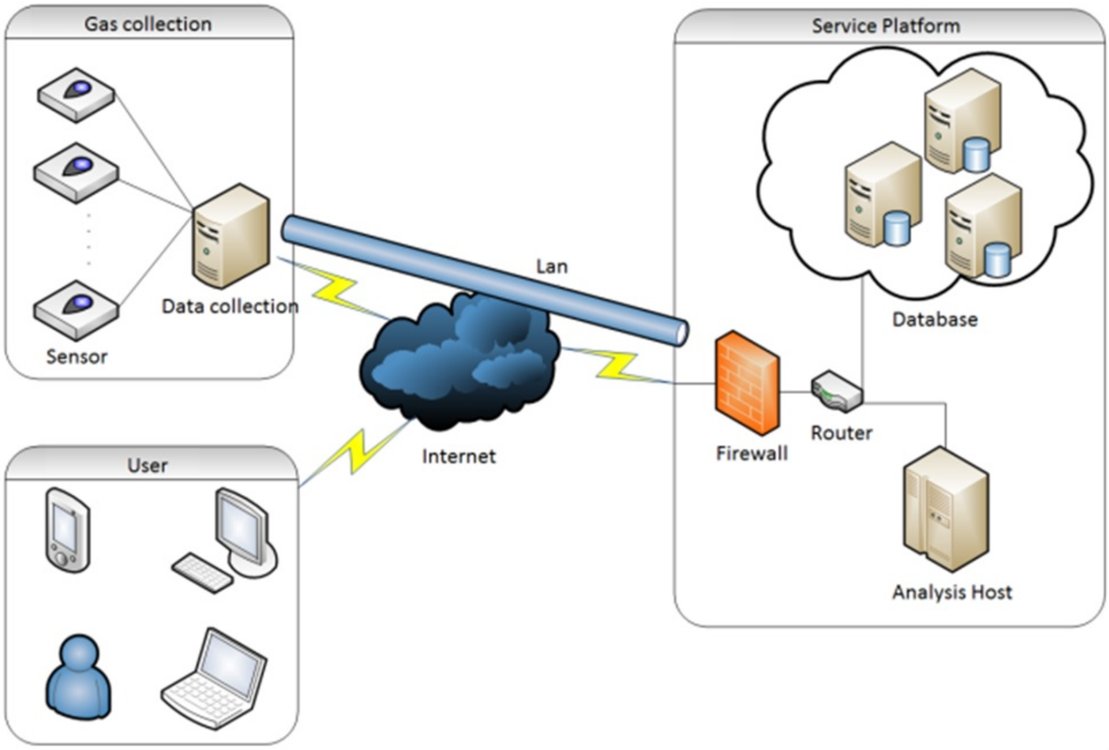

FIGURE 3.1: Primary System architecture

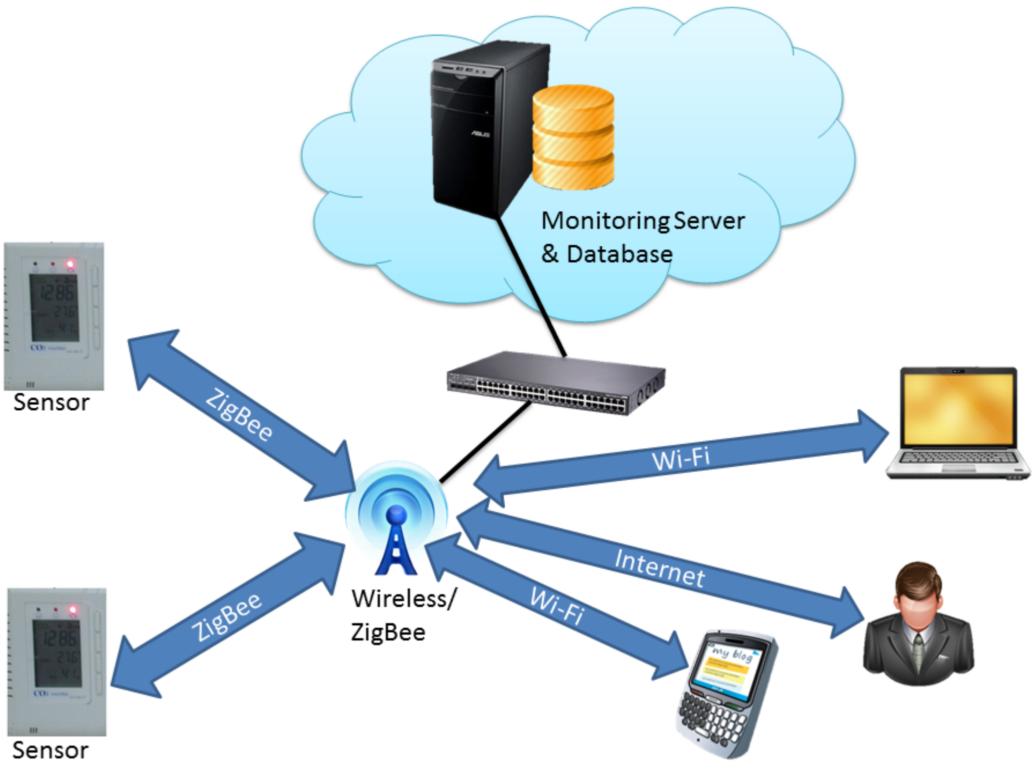

FIGURE 3.2: Secondary System architecture

on the server to analyze data collected by sensors. Thus, the plug can be automatically controlled by the program. Its details are illustrated in a flow chart in Chapter 4.

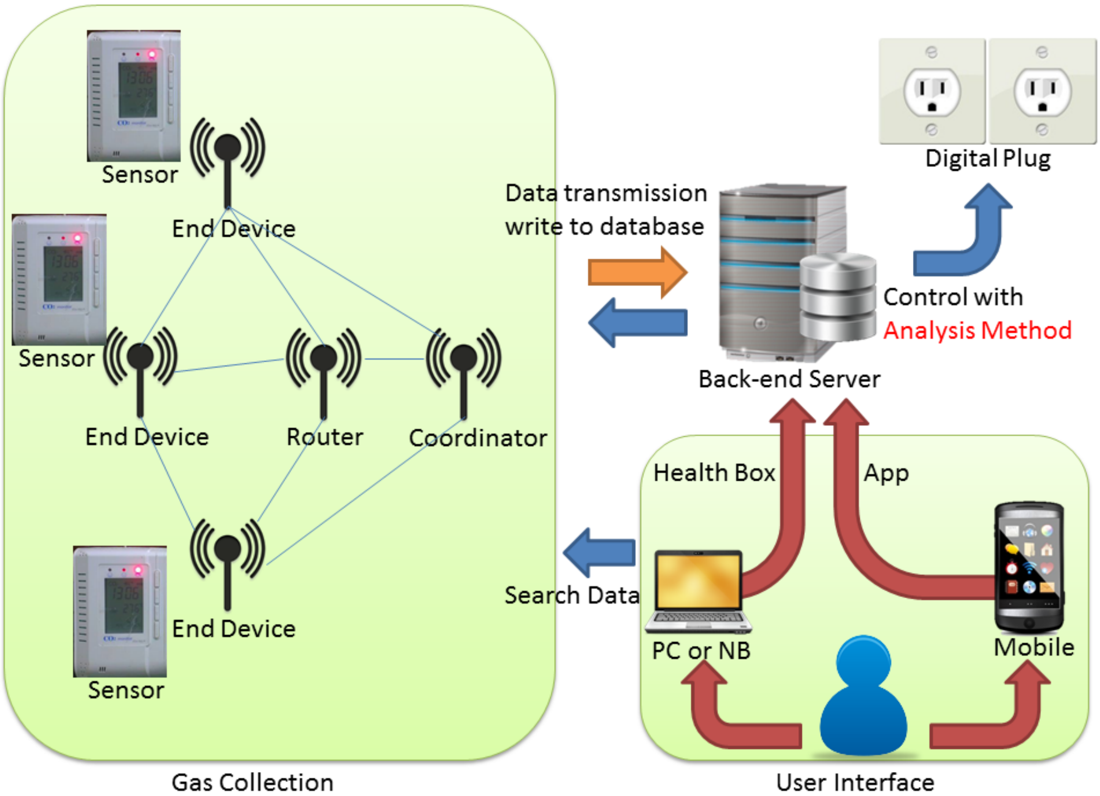

FIGURE 3.3: A scenario of Web service architecture

Through the single entrance on the platform, users can operate the system with client equipment, monitor system data from hardware devices, and plot the data for analysis by using graphic software in real time as shown in Figure 3.4. With the real-time monitoring service, the air quality is actively analyzed based on the real-time data, and, if abnormal situations are detected, alerts can be sent to clients who can make right decisions and respond instantly to solve the problems.

We used Apache Web Server Version 2.2.8 and MySQL Database Manager Version 2.10.3 on windows 7 64-bit OS to build this platform. When our sensor detects value of gas, it will write data into the database. Figure 3.5 shows CO_2 data collected by sensors. We can use mobile devices or web to load data to check values of air conditions. This idea can be achieved in a private cloud.

FIGURE 3.4: Health Box Login

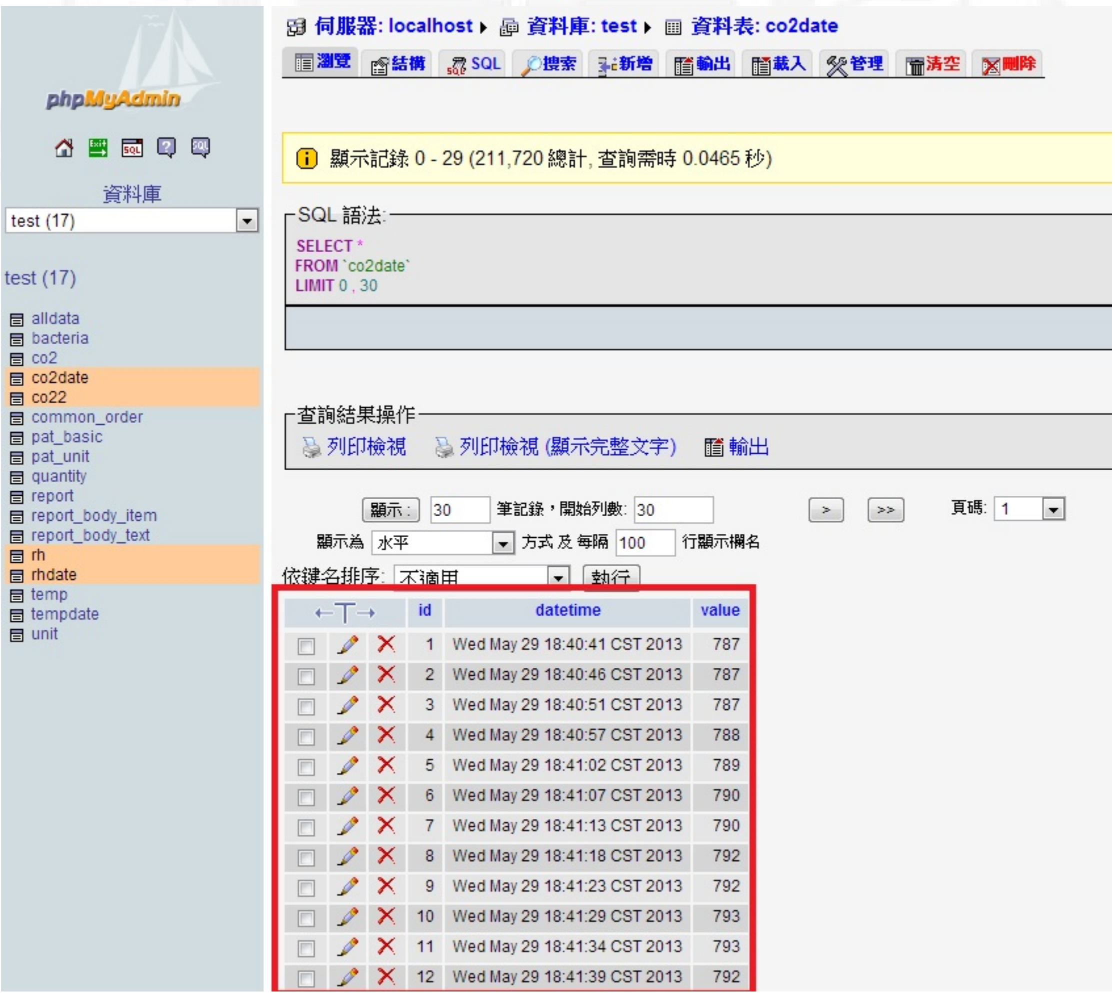

FIGURE 3.5: Detects value of CO_2 then write in database.

For a vast system, it requires a lot of resources to use its complicated structure in computing and operation. Grid computing is the federation of computer resources from multiple locations to reach a common goal [22, 23]. This technology not only improves efficiency, but makes a full use of remaining resources and reduces the total cost. Our data storage was supplemented by the GRID technology to create a data base with dynamic maintenance [24, 25]. In the data mining side, In the data mining side, we used connection rules to do statistics and reference data; then we could give more suitable advice for improvement. Moreover, association rules can be used to predict potential diseases. In this part, we tried to build a database which can be used to monitor, record, and collect data about CO_2 concentration, temperature and humidity by ZigBee interface. The flow chart of the environment monitoring process is shown in Figure 3.6.

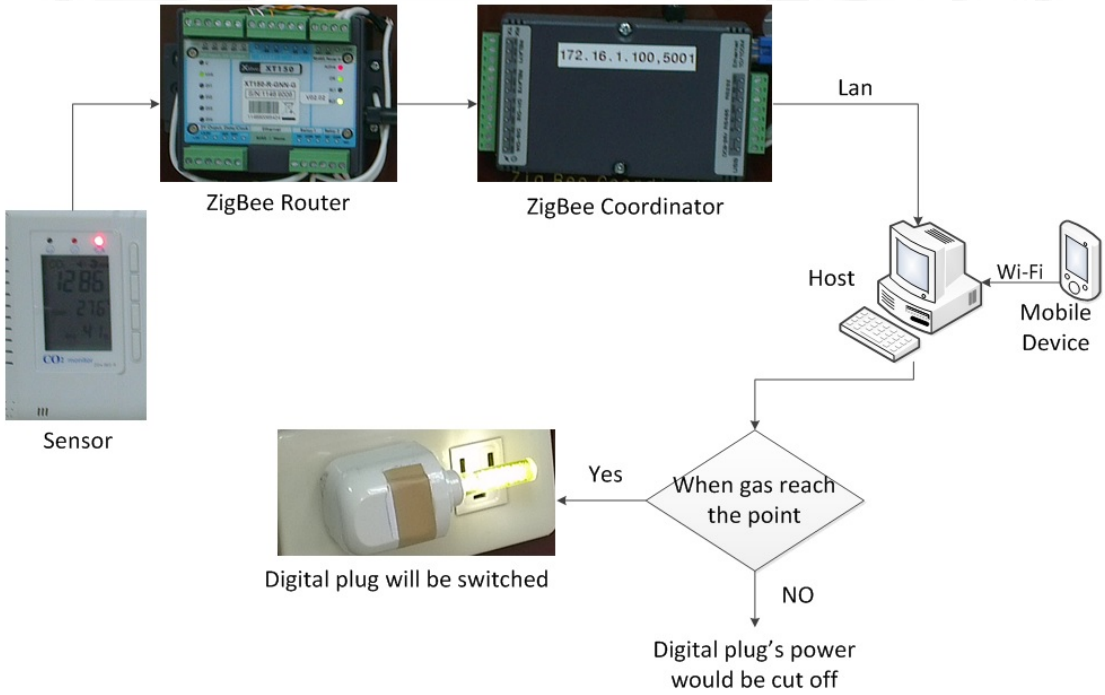

Figure 3.6: Flow chart of environment monitoring

Figure 3.7 shows pictures of devices used in environment monitoring. The right part shows sensors, a digital plug, and the connector between sensors and the Zigbee Router; the left part shows the ZigBee Router or Zigbee Coordinator using IEEE 802.15.4 standard to wirelessly transmit data in a range about 10m to 100m. The whole monitoring process is described as follows: first data are collected by sensors, then they are transmitted to Coordinator by the ZigBee Router and sent to the backend of the server through LAN for further analysis. If the collected air data are found to reach the setting limit, the system Coordinator will respond by transmitting alert signals to the ZigBee Router to trig the front-end of the digital plug.

FIGURE 3.7: Physical devices used in environment monitoring

Chapter 4

Experimental Results

4.1 Experimental Environment

In this section, we describe our Intelligent Environment Monitoring System (IEMS). By the way, some scholars also name their system as IEMS (Indoor Environment Monitoring System)[26]. In this paper, we defined this 'I' not only indoor, but also intelligent. See in Table 4.1, our monitoring server used a desktop, which has CPU Intel(R) Core^{TM} i7-2600 @3.40GHz, eight gigabyte memory and 500 gigabyte disk. We used two NICs (Network Interface Card) to build our network environment. One NIC connects to the Internet, and the other connects to the ZigBee Coordinator. Then the coordinator can use zigbee IEEE802.15.4 to connect to the ZigBee Router. At last, the gas sensor used RS232 to connect to the ZigBee Router. The hardware specification is listed in the following:

When sensor detects values of gas, it sends data through the ZigBee Router to the ZigBee Coordinator, then to the monitoring server. We set up IP address of ZigBee Coordinator as 127.16.1.100, and its port, 5001. Figure 4.1 shows our config. on the web. Our compiler uses Eclipse Java Version 1.4.2. As coded, it will catch the gas data from sensor in the ASCII format (American Standard Code for Information Interchange) [27]. The data from the sensor will be written to database; and we can immediately see the value of gas on the compiler.

The database environment used Apache Web Server Version 2.2.8, PHP Script Language Version 5.2.6, MySQL Database Version 5.0.51b and phpMyAdmin Database Manager Version 2.10.3 to build. In Table 4.2 lists software specifications.

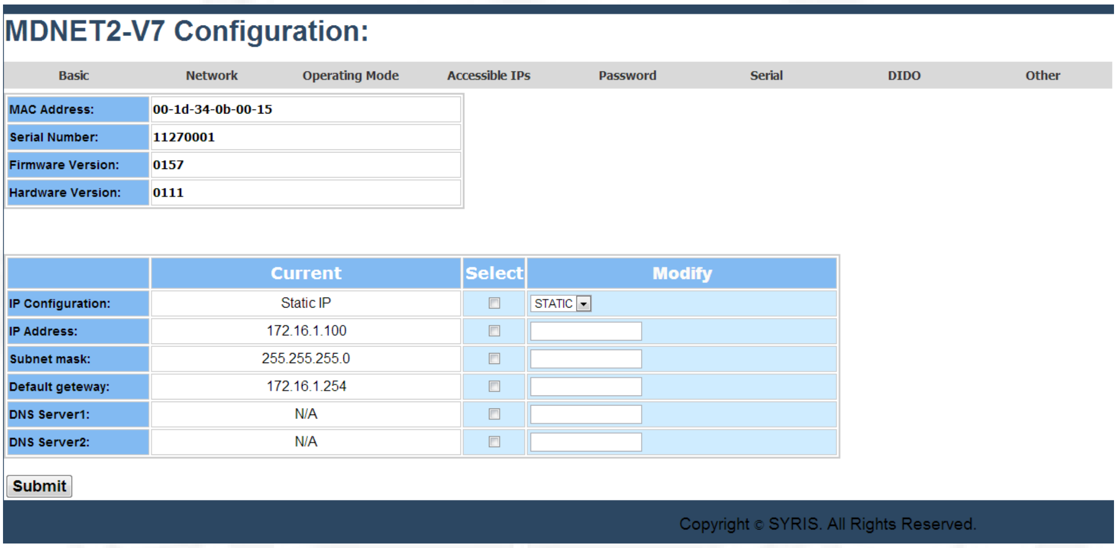

FIGURE 4.1: ZigBee Coordinator config

TABLE 4.1: Hardware specification

Hardware	
CPU	Intel(R) Core^{TM} i7-2600 @3.40GHz
Memory	DDR3 1333MHz 4GB *2
Disk	500GB
NICs	Intel(R) 82579LM Gigabit Network Connection Realtek RTL8139/810x Family Fast Ethernet NIC
Sensor	CO_2 monitor ZGw 063-R
ZigBee Router	XT15
ZigBee Coordinator	XT200-PA

TABLE 4.2: Software specification

Software	
OS	Windows 7 64-bit
Compiler	Eclipse Java EE IDE for Web Developers Version: 1.4.2
PHP	PHP Script Language Version 5.2.6
MySQL	MySQL Database Version 5.0.51b
phpMyAdmin	phpMyAdmin Database Manager Version 2.10.3

4.2 Experimental Results

4.2.1 Health Box and Web Information

We have implemented a service platform called HEALTH Box designed with the aim for air quality monitoring. The platform was built with a distributed storage, and a single entrance as wireless / wire sensors' data integration to let user access data. When user logins Health Box, a homepage will show as Figure 4.2. One can see a hospital map, because we plan to install our system on Taichung Veterans General Hospital to monitor its environment. But now we monitor our lab environment in the initial design stage. So, in this chapter, the gas values are collected at our lab. We set the Outpatient Building, 1st Floor, Information at the homepage sync our collection database. We can search the value we collected as shown in Figure 4.3.

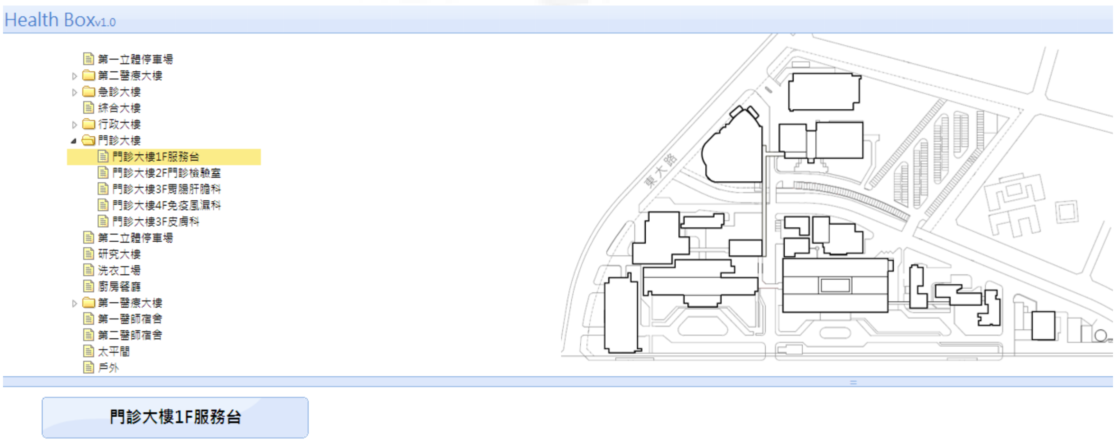

FIGURE 4.2: Health Box homepage

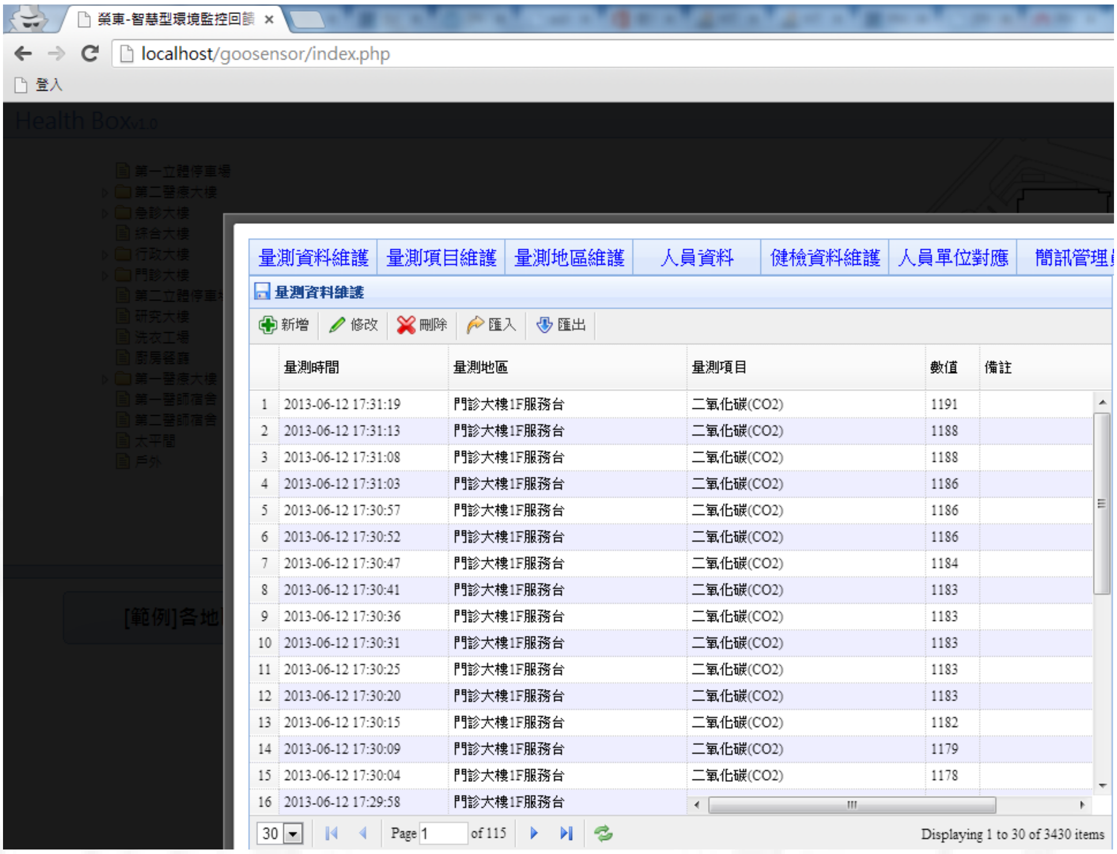

FIGURE 4.3: Search CO_2 concentration

The platform was used to collect data about CO_2 concentration, temperature and humidity for real-time monitoring, and display collected data in a web page by charts using an open source program named Java Highcharts [28]. Figure 4.4 shows the CO_2 web Information. Figure 4.5 shows temperature web Information. At last, humidity web Information is shown in Figure 4.6.

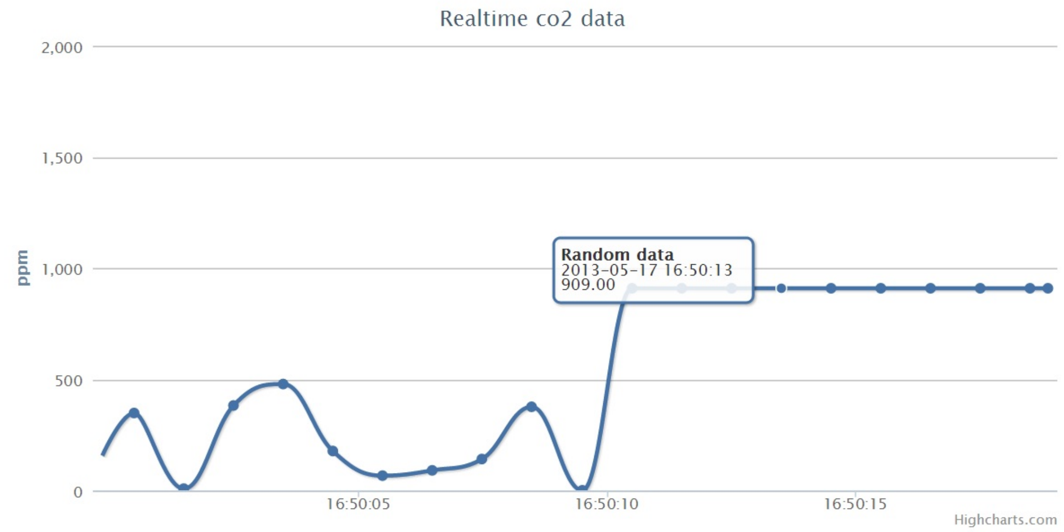

Figure 4.4: Web Information of CO_2 concentration

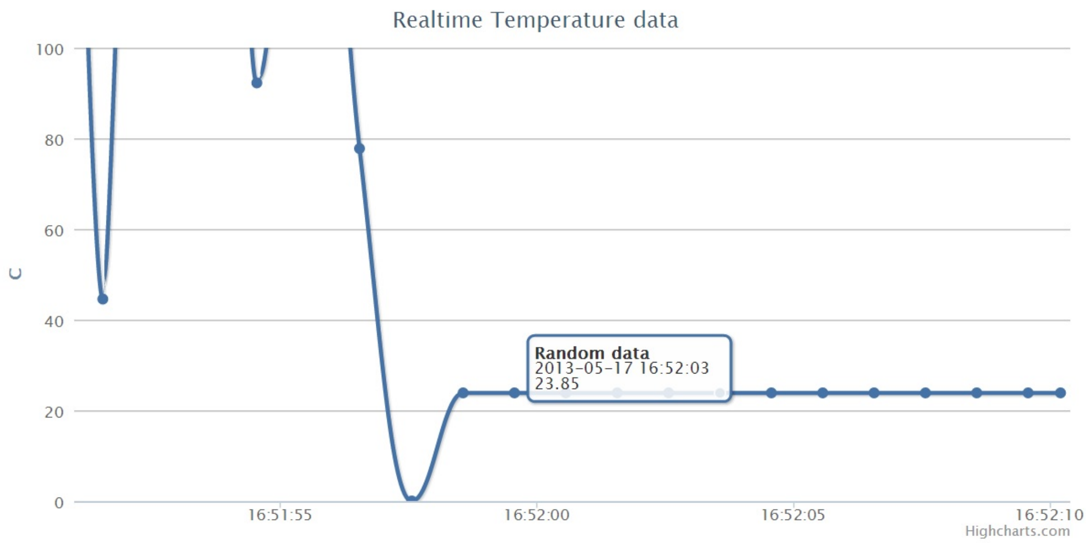

Figure 4.5: Web Information of Temperature concentration

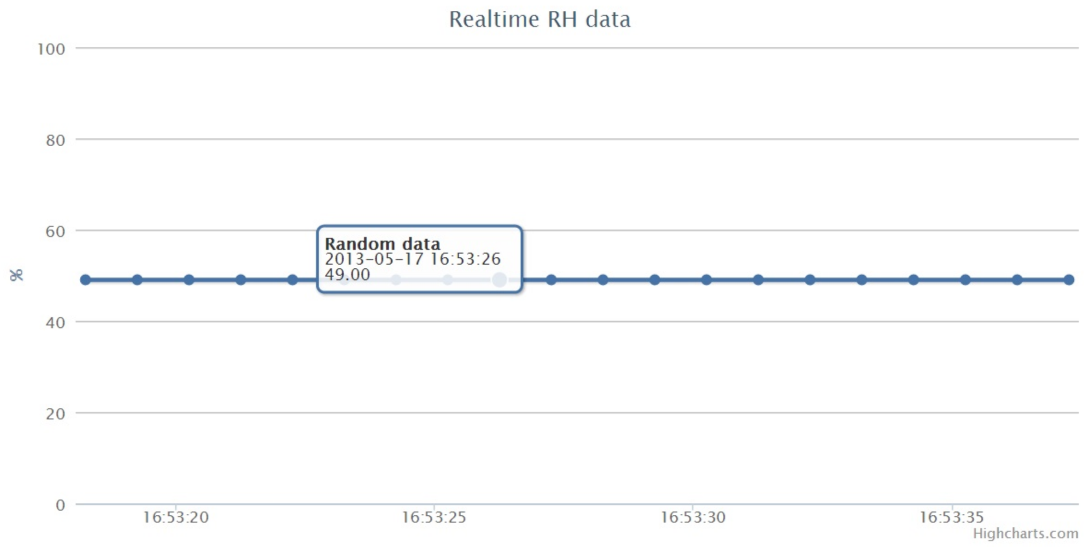

FIGURE 4.6: Web Information of Humidity concentration

4.2.2 Use IEMS to Improve the quality of the environment

In this experiment, we need to know our environment status. So we collect data about CO2 concentration, temperature, and humidity. Figure 4.7 shows the CO_2 concentration on June 11. We also collect data for a week, as shown on Figure 4.8. After we analyze these graphs, we observe the CO_2 concentration almost over 1000ppm at three o'clock to half past eight in the afternoon.

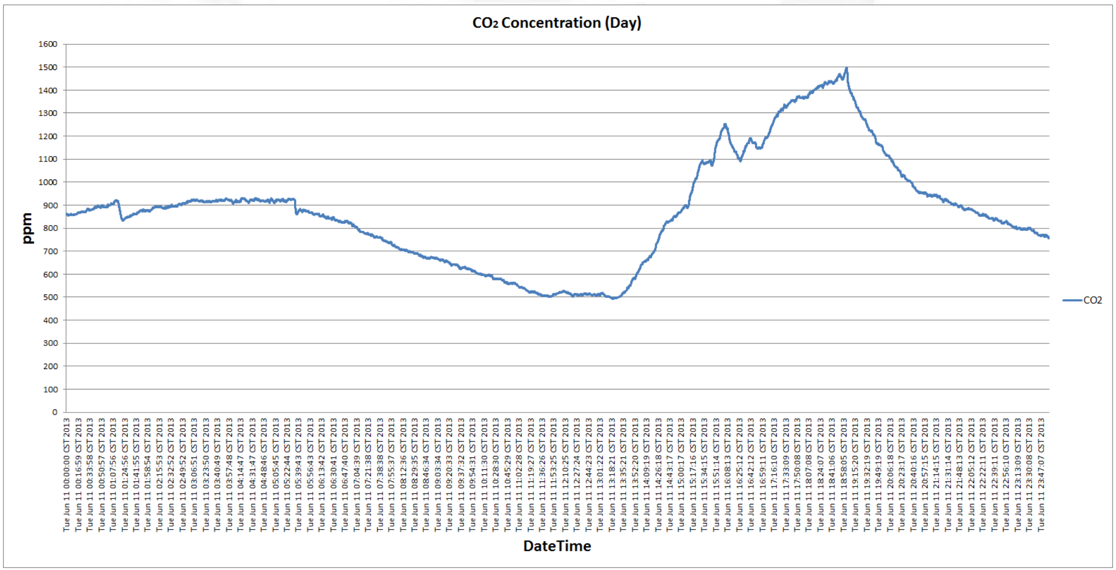

FIGURE 4.7: CO_2 concentration of a day

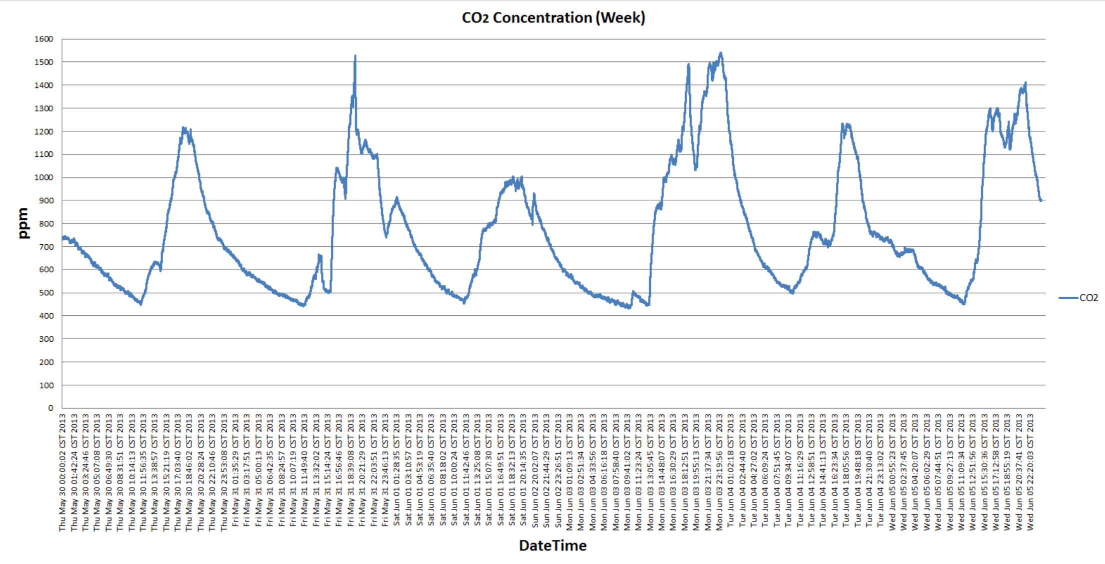

FIGURE 4.8: CO_2 concentration of a week

Figure 4.9 shows the Temperature data on June 4. We also collect data for a week, as shown on Figure 4.10. After we analyze these graphs, we observe the Temperature data almost over 29 ℃ at the morning.

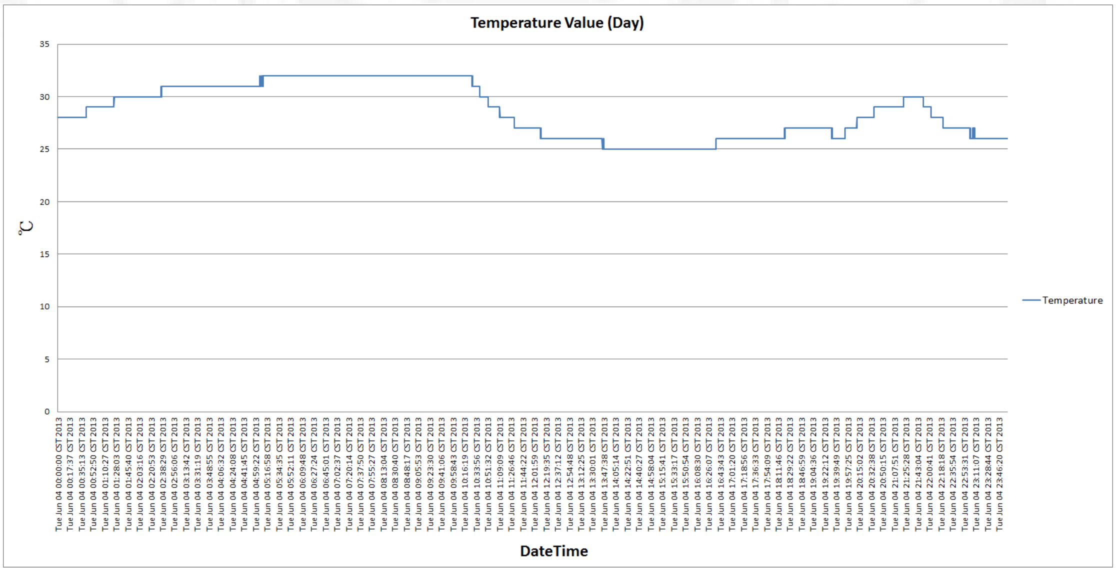

FIGURE 4.9: Temperature data of a day

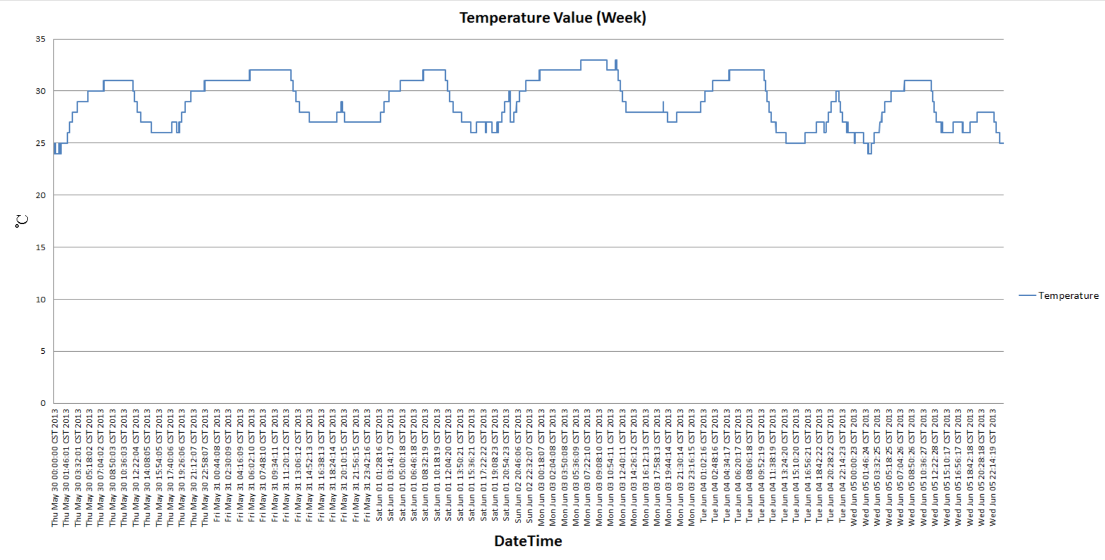

FIGURE 4.10: Temperature data of a week

Figure 4.11 shows the Humidity data on May 31. We also collect data for a week, as shown on Figure 4.12. After we analyze these graphs, we observe the Humidity data almost over 40% at AM 3:00 to PM 12:00.

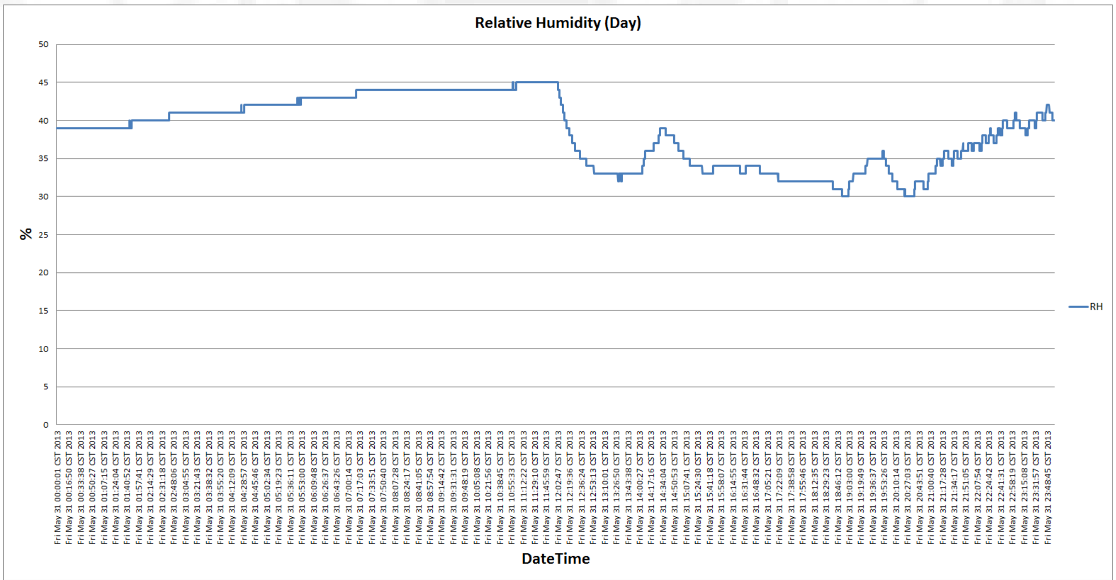

FIGURE 4.11: Humidity data of a day

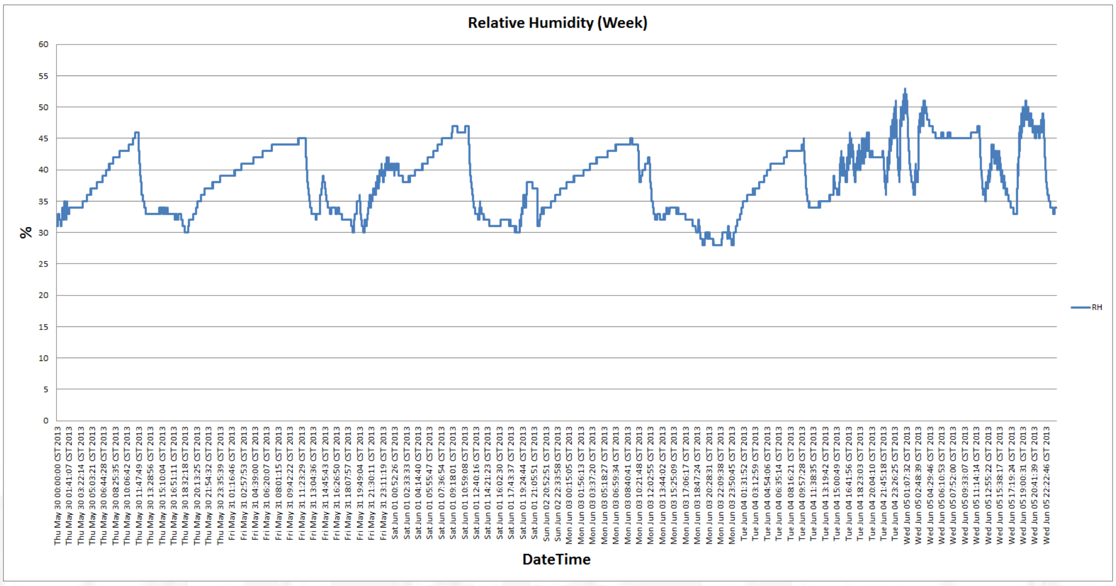

FIGURE 4.12: Humidity data of a week

After summarizing these data and referencing the Indoor Air Quality Standards set by Environmental Protection Administration shown in Table 4.3 [29], the question faced by us is how to effectively optimize our system according to the collected environmental information. Our solution is to install an intelligent plug. So we write java codes to judge and analyze data on the back-end server to automatically turn on the power when the server detects that the real-time data reach the set point or the set time.

Figure 4.13 shows the flow chart of the analysis method written in the java code on the back-end server. At the beginning, it logins MySql then connects the front sensor device to collect data and write the data into MySql. We set CO_2 concentration to control the right plug; its trigging point is CO_2 concentration of 1200ppm or the time in three o'clock to half past eight in the afternoon. Figure 4.14 shows the digital plug will be switched. The left plug controlled by the relative humidity, and its trigging point is humidity 42% or the time in AM 3:00 to PM 12:00. When the CO_2 concentration or the humidity decreases below the set value, its associated digital plug will be cut off automatically. We did not use the temperature value to control the plug, because the plug is powered by ac 110V. If we want to optimize the temperature, we need to plug the air conditioner in the plug. But most air-conditioners are powered by ac 220V. So we decide to control

CO_2 concentration and humidity by the plug. In the future, we will update our device to control temperature by the new plug that can offer ac 220V power supply.

TABLE 4.3: Indoor Air Quality set Standards by Environmental Protection Administration

Items	Standard values		Units
CO_2	Value of eight hours	1000	ppm
CO	Value of eight hours	9	ppm
Formaldehyde(HCHO)	Value of an hour	0.08	ppm
Volatile organic compound(TVOC, includ twelve kinds of sum of volatile organic compounds)	One hour values	0.56	ppm
Bacteria	Highest value	1500	CFU/m^3(Colony forming units/ Cubic meter)
Fungi	Highest value	1000 In addition to fungal ratio concentrations of indoor and outdoor were less than or equal 1.3	CFU/m^3(Colony forming units/ Cubic meter)
Particle size less than or equal to 10 microns(m) suspended particulates(PM_10)	Value of a day	75	g/m^3(microgram/ Cubic meter)
Particle size less than or equal to 2.5 microns(m) suspended particulates($PM_2.5$)	Value of a day	35	g/m^3(microgram/ Cubic meter)
O_3	Value of eight hours	0.06	ppm

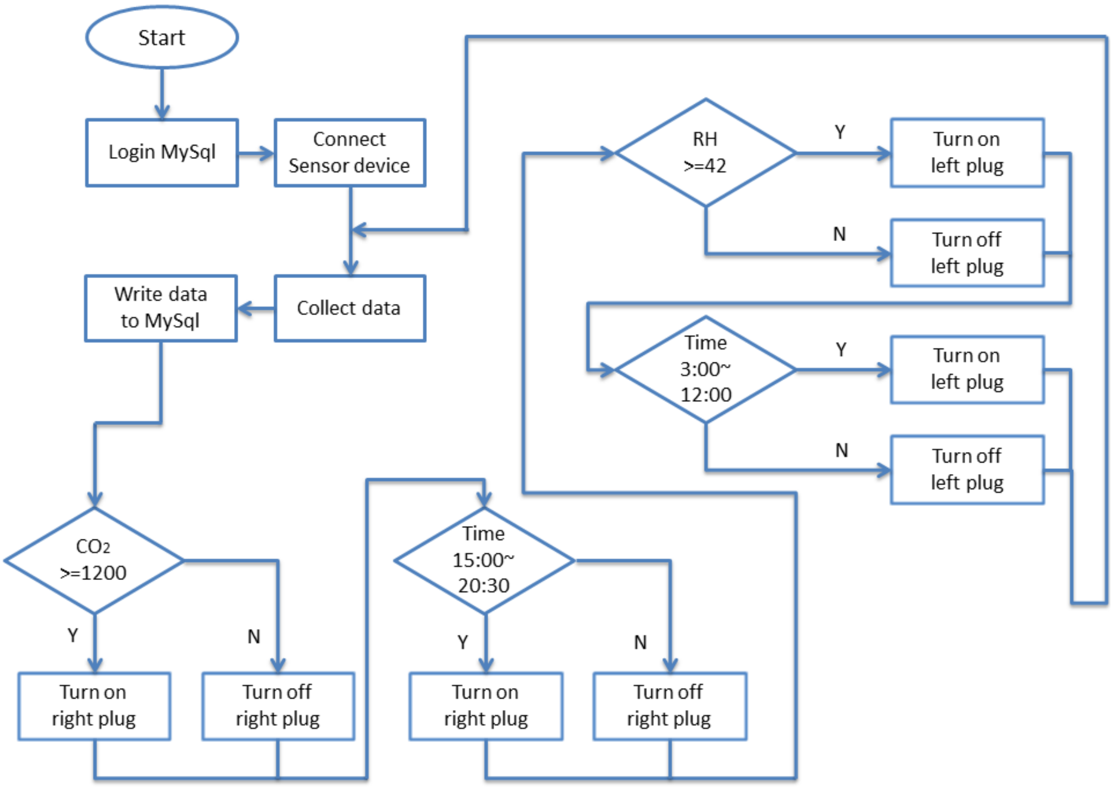

FIGURE 4.13: Flow chart of analysis method on back-end server

FIGURE 4.14: The digital plug be switched when CO_2 reaches the set point

In the previous paragraph, we mentioned the trigging point is CO_2 concentration of 1200 ppm or the time is PM 3:00 to PM 8:30. We observe the CO_2 concentration almost over 1000ppm at PM 3:00 to PM 8:30. The specifications of Indoor Air Quality set Standards by Environmental Protection Administration formulate the CO_2 concentration need to be decreased below 1000 ppm. So we set our Intelligent Environment Monitoring System (IEMS) using the analysis java code to turn on its right plug with the exhaust fan. The red line in Figure 4.15 shows one week of the new CO_2 concentration collected with IEMS, and the blue line shows one week of the old data collected without IEMS. We can observe that the $CO2$ concentration has improved significantly at PM 3:00 to PM 8:30, and the rest of the time are about 1200 ppm that is the point we set to maintain. Figure 4.16 shows the $CO2$ concentration collected in one day with and without IEMS.

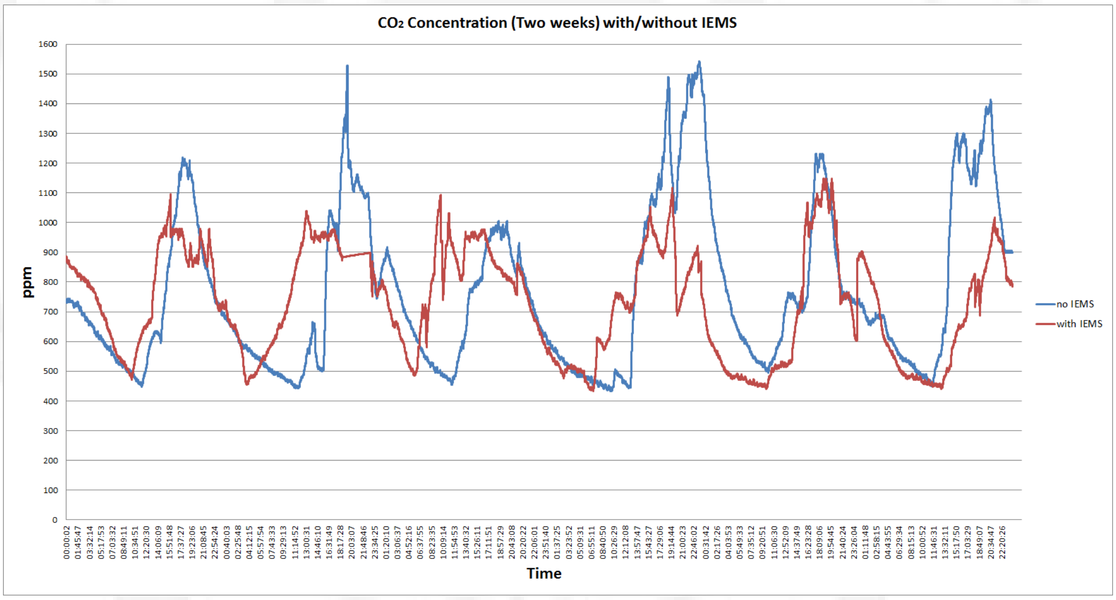

FIGURE 4.15: CO_2 concentration with/without IEMS of a week

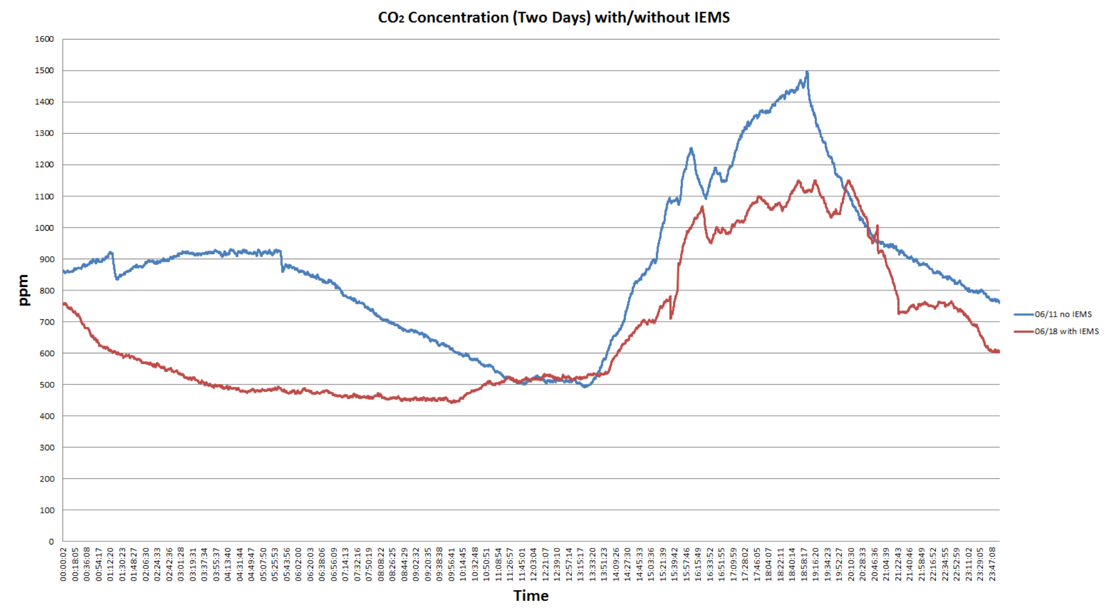

FIGURE 4.16: CO_2 concentration with/without IEMS of a day

We set our Intelligent Environment Monitoring System (IEMS) using the analysis java code to turn on its left plug with a dehumidifier. The red line in Figure 4.17 shows the new humidity data collected in one week with IEMS, and the blue line shows the data collected in one week without IEMS. We observe that the humidity has improved significantly at AM 3:00 to PM 12:00, and the rest of the time are about 42% that is the point we set to maintain. Figure 4.18 shows the humidity data collected in one day with and without IEMS.

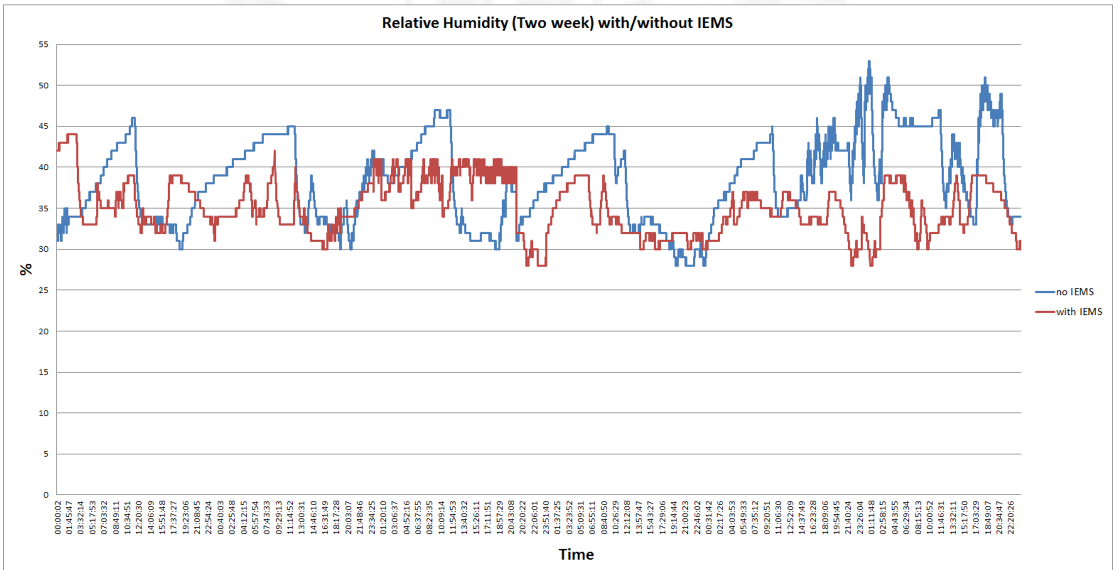

FIGURE 4.17: Humidity data with/without IEMS of a week

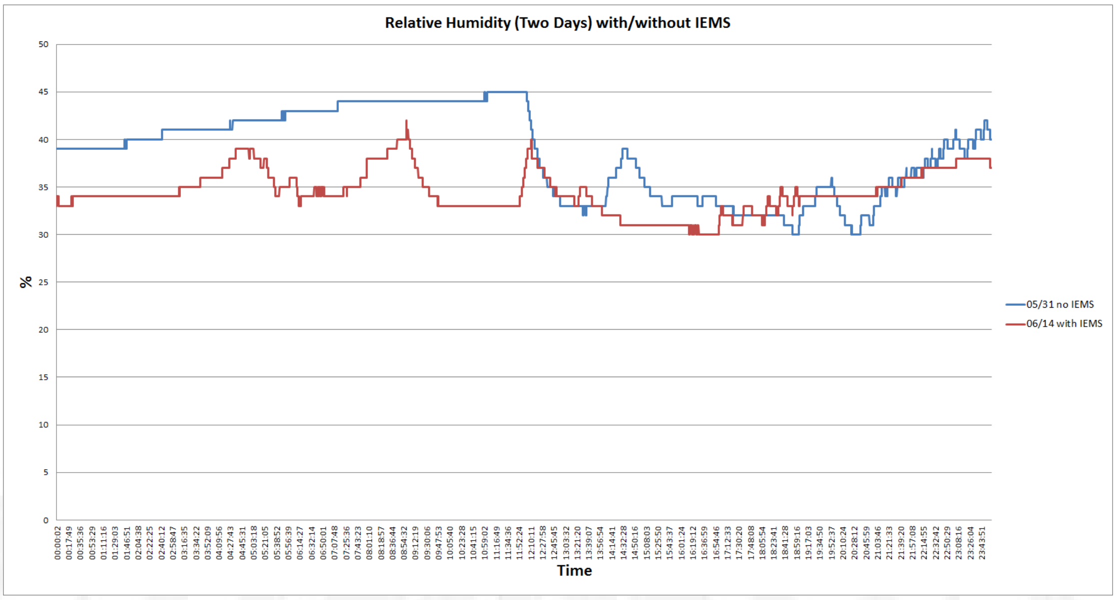

FIGURE 4.18: Humidity data with/without IEMS of a day

4.2.3 Real-time monitoring on mobile

In order to monitor data in a more convenient situation, we design a simple Android App to do the real-time monitoring by smart phones. This App has two features. The first one shows the real-time monitoring graph. Figure 4.19 shows that feature, in which you can choose the graph of value that you want to see. After user makes choice, the graph is shown as Figure 4.20. The other feature is to display digitally to allow the user to know the state of the Environment as shown in Figure 4.21. Thus the App user can be aware of the immediate CO_2 concentration even when he is out of the monitored space. The whole structure is implemented in a private cloud just as introduced in previous section.

Figure 4.19: Android App menu showing CO_2 concentration, humidity, temperature

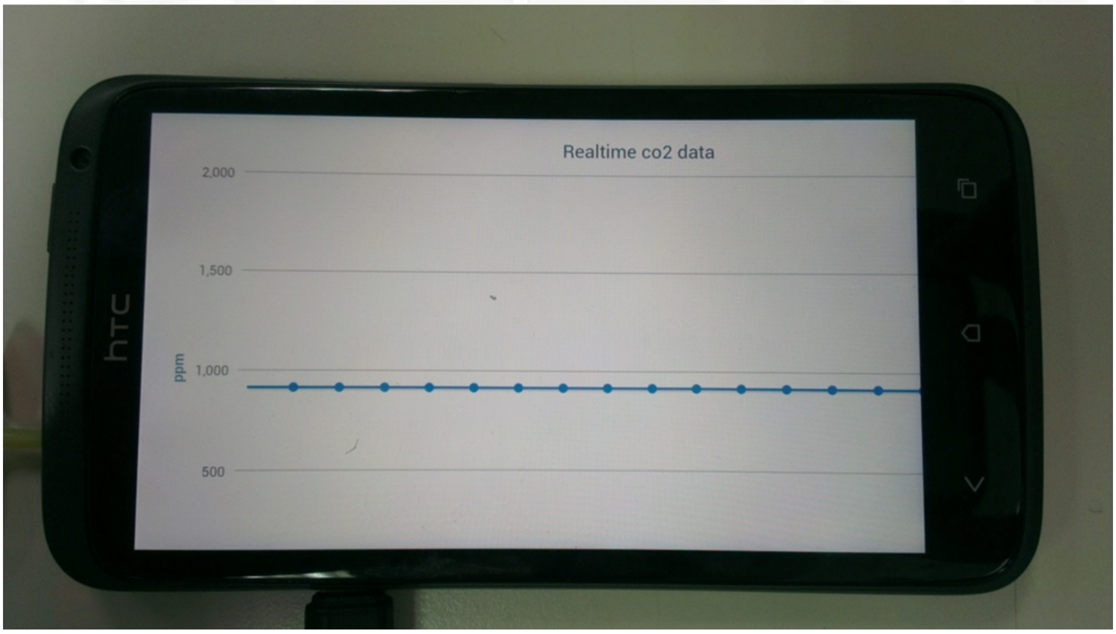

Figure 4.20: The graph of the real-time monitoring on mobile device

FIGURE 4.21: Display digital real-time monitoring on mobile device

4.3 Discussion

We have implement an environment monitoring platform that can monitor air quality and humidity, and based on the real-time data, can automatically trig devices in response. But there is still a little time delay of response from frontend devices to the back-end host. Our current solution is to use more sensitive sensors and rewrite the monitoring APP on the smart phone. Moreover, we also expect to present data in numbers instead of charts for better readability.

Besides, we hope our device can be for the lightweight demand [30]. So we tried to use Intel Atom PQ7-C100XL to replace the back-end server.

PQ7-C100XL' s development is based on popular 3.5" ESB(by Portwell). Figure 4.22 shows a built-in VGA port, a LVDS port for dual independent display, a Gigabit Ethernet port, 7 USB port, a SATA, a SDIO socket expand through a Mini PCI-E socket and a 12V DC power supply connect to DC circuit for the demand of DC-in application. PQ7-C100XL presents the most ideal solution for any kinds of low power-consuming device such as outdoor embedded system. So we decided to use this board to arrange our sensor. The board was considered as a small server, accessing and judging the accepted value PQ7-C100XL' s OS is windows 8, because of its x86 architecture. The whole composition is shown in Figure 4.23, which caters to the light weight demand.

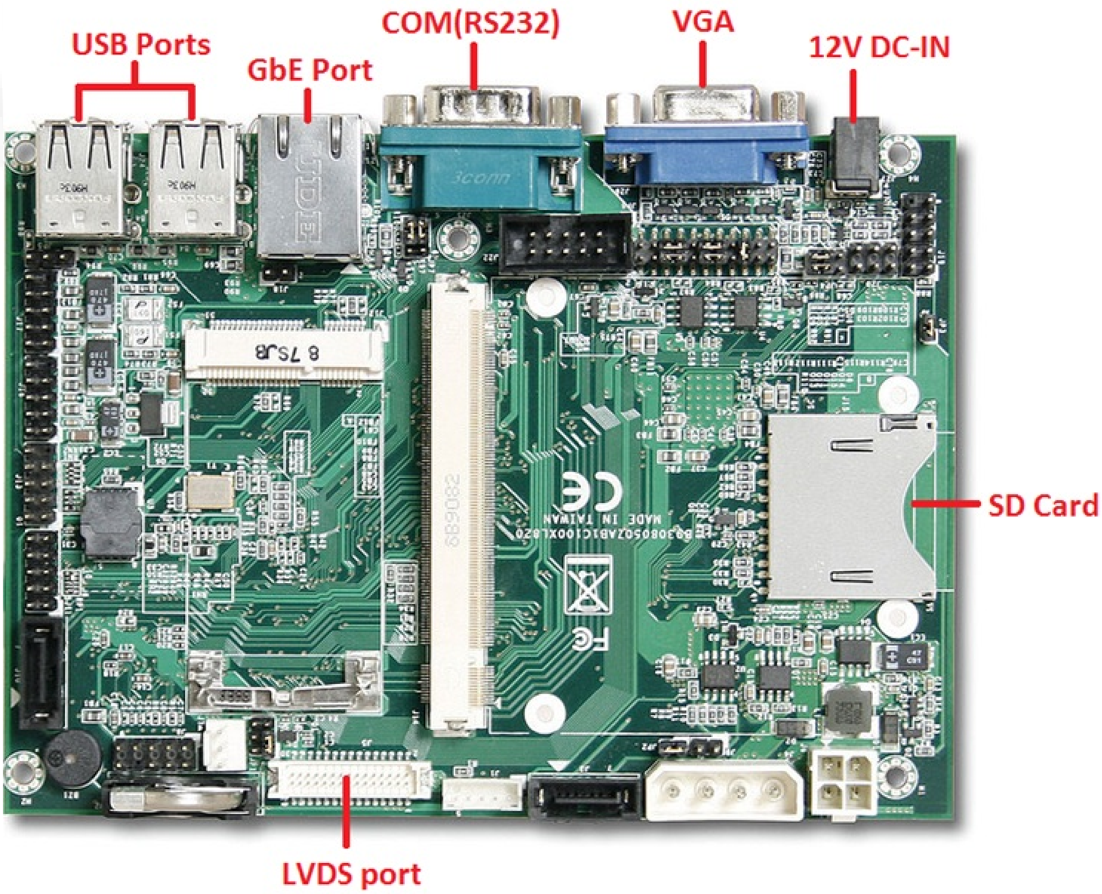

FIGURE 4.22: PQ7-C100XL

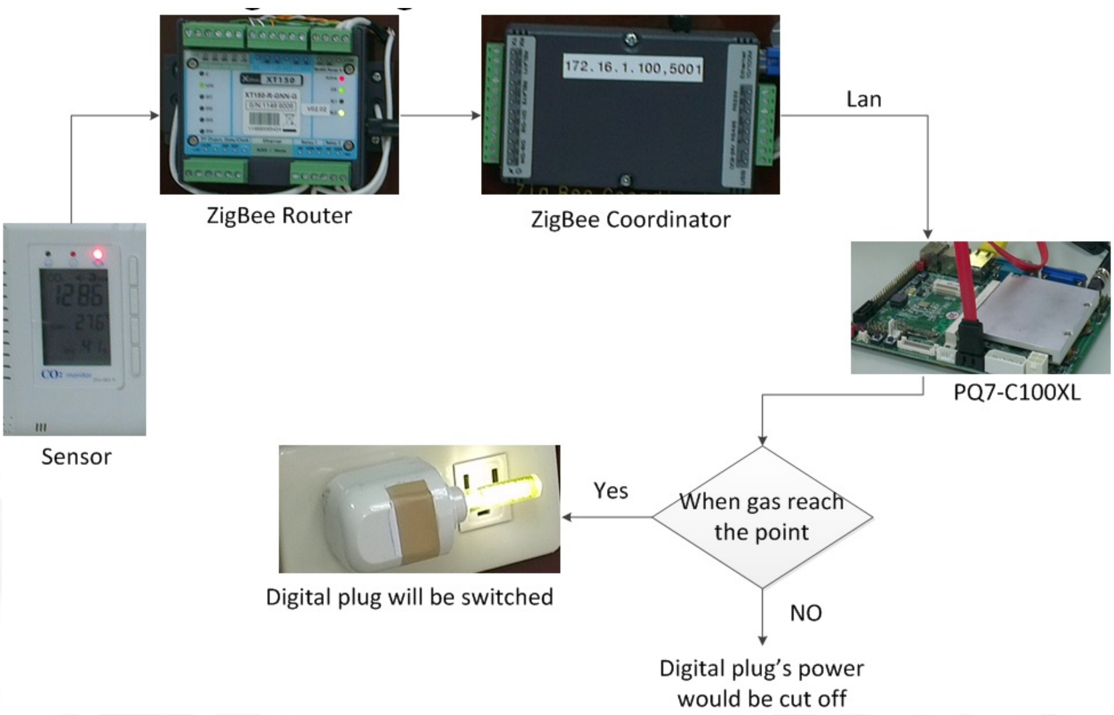

FIGURE 4.23: With Intel Atom flow chart

Besides, although Intel Atom Board actually satisfies the light weight demand, but we still be confronted with some issues. For example, we are not able to link the phone directly to the board to read monitoring data while this is the only physical board already connected to ZigBee Coordinator. Right now we are still searching for better plans for this.

Chapter 5

Conclusions and Future Work

With the fast development of economy, it is hard for nations to evaluate, assess, and regulate industrial and commercial areas. As a result, the air pollution is always an inevitable byproduct of the modern life, and to make it worse, the origin of specific air pollution is not easy to identify because of the diffusion nature of the air.

Because of the nearness to the Taichung Industrial Park, Taichung Veterans General Hospital always suffers from serious air pollution problems. So we choose the hospital as a realistic case, collecting data in a long term, and storing the data in a database called as the Health Box, which establishes a wireless monitoring information center to assemble all kinds of sensor data for various conditions.

In the future, we will add more sensors to monitor different area at the same time. So we need to think how to analyze and store these big data. Our idea is to use Hbase to replace MySql. The amount of data per day is more 18,000, which is a very large amount for the storage in the single stage sensor. We hope we can control and itemize them effectively when more sensors are used in the future.

The Health Box combines our applications on grid computing and cloud technologies to create a system which is efficient, fault-tolerant, fast, and lightweight. For software development, we form a source code community to provide users a safe and convenient platform for data analysis, file access, and transmission. Its architecture also has high extensibility and compatibility, which will significantly reduce the time and human resource costs in the future.

Bibliography

[1] B. Allcock, J. Bester, J. Bresnahan, A. Chervenak, I. Foster, and C. Kesselman. Protocols and services for distributed data-intensive science. In *ACAT2000 Proceedings*, pages pp. 161–163, 2000.

[2] B. Allcock, J. Bester, J. Bresnahan, A. Chervenak, I. Foster, C. Kesselman, S. Meder, V. Nefedova, D. Quesnel, and S. Tuecke. Secure efficient data transport and replica management for high-performance data-intensive computing. In *Proc. of the Eighteenth IEEE Symposium on Mass Storage Systems and Technologies*, pages pp. 13–28, 2001.

[3] Chervenak, E. Deelman, I. Foster, L. Guy, W. Hoschek, A. Iamnitchi, C. Kesselman, P. Kunszt, and M. Ripeanu. A framework for constructing scalable replica location services. In *Adv. Phys*, 2002.

[4] The unicore forum. http://www.unicore.org/.

[5] The gemss project: Grid-enabled medical simulation. http://www.gemss.de/, 2002–2005.

[6] The simbio project. http://www.simbio.de/, 2000.

[7] B. Allcock, J. Bester, J. Bresnahan, A. Chervenak, I. Foster, C. Kesselman, S. Meder, V. Nefedova, D. Quesnel, and S. Tuecke. Data management and transfer in high-performance computational grid environments. In *Parallel Computing*, pages 28(5):749–771, May 2002.

[8] Chervenak, I. Foster, C. Kesselman, C. Salisbury, and S. Tuecke. The data grid: Towards an architecture for the distributed management and analysis of

large scientific datasets. In *Journal of Network and Computer Applications*, pages 23:187–200, 2001.

[9] K. Czajkowski, S. Fitzgerald, I. Foster, and C. Kesselman. Grid information services for distributed resource sharing. In *Proceedings of the Tenth IEEE International Symposium on High-Performance Distributed Computing (HPDC-10 01)*, pages 181–194, August 2001.

[10] Hui Wang, Xue-ren Dong, Yu-zhen Ma, Xiao-wei Yang, and Feng-nan Liu. Design of greenhouse environment wireless monitoring system based on zigbee. In Honghua Tan, editor, *Informatics in Control, Automation and Robotics*, volume 132 of *Lecture Notes in Electrical Engineering*, pages 579–586. Springer Berlin Heidelberg, 2011.

[11] Cloud computing. *http://en.wikipedia.org/wiki/Cloud_computing.*

[12] MySQL Cluster Architecture Overview. A mysql® technical white paper. April, 2004.

[13] David E, Paul Z, and Chris R. Copyright © 2000 elsevier inc. all rights reserved. mysql on linux. pages 283–309.

[14] Johan L and Jonas O. Zigbee for wireless networking. March 2005.

[15] Sinem c.e. ZigBee/IEEE 802.15.4 Summary, Sep 2004.

[16] Zigbee. http://en.wikipedia.org/wiki/ZigBee.

[17] Chengbo Yu, Yanzhe Cui, Lian Zhang, and Shuqiang Yang. Zigbee wireless sensor network in environmental monitoring applications. In *Wireless Communications, Networking and Mobile Computing, 2009. WiCom '09. 5th International Conference on*, pages 1–5, 2009.

[18] Yongbing Xiao, Jianxian Zhang, Tingting Liu, and Xiang Gao. An environmental monitoring system based on zigbee for emergency applications. In *Wireless Communications, Networking and Mobile Computing, 2009. WiCom '09. 5th International Conference on*, pages 1–4, 2009.

[19] Shun-qi Chai, Lei Ji, and Hong Wu. Zigbee wireless sensor network for environmental monitoring system. pages 75081R–75081R–6, 2009.

[20] Yi Zhang, Peng Xiong, Yuan Luo, and Lin Li. Design of remote home environment monitoring and health care monitoring system based on data confusion. In *Automation and Logistics (ICAL), 2011 IEEE International Conference on*, pages 35–39, 2011.

[21] Hl7. http://www.sychang.tcu.edu.tw/HL7.htm.

[22] K. Czajkowski, I. Foster, and C. Kesselman. Resource co-allocation in computational grids. In *Proceedings of the Eighth IEEE International Symposium on High Performance Distributed Computing (HPDC-8 99)*, August 1999.

[23] I. Foster, C. Kesselman, and S. Tuecke. The anatomy of the grid: Enabling scalable virtual organizations. In *Int. J. of Supercomputer Applications and High Performance Computing*, volume 15(3), pages pp. 200–222, 2001.

[24] C. T. Yang, I. H. Yang, C. H. Chen, and S. Y. Wang. Implementation of a dynamic adjustment mechanism with efficient replica selection in co-allocation data grid environments. In *Proceedings of the 21st Annual ACM Symposium on Applied Computing (SAC 2006) - Distributed Systems and Grid Computing (DSGC) Track*, volume vol. 1, pages pp. 797–804, April 23-27 2006. Dijon, France.

[25] Chao-Tung Yang, I-Hsien Yang, Kuan-Ching Li, and Ching-Hsien Hsu. Recursive-adjustment co-allocation scheme in data grid environments. In *ICA3PP 2005 Algorithm and Architecture for Parallel Processing, Lecture Notes in Computer Science*, volume vol. 3719, pages pp. 40–49, October 2005. Springer-Verlag.

[26] Suman Sankar Bhunia, Sarbani Roy, and Nandini Mukherjee. Iems: Indoor environment monitoring system using zigbee wireless sensor network. In *Proceeding ICCCS '11 Proceedings of the 2011 International Conference on Communication, Computing & Security*, pages 142–145, 2011.

[27] Ascii (american standard code for information interchange). http://zh.wikipedia.org/wiki/ASCII.

[28] Highcharts js. http://www.highcharts.com/.

[29] Indoor air quality. http://iaq.epa.gov.tw/indoorair/index.html.

[30] Jie Cao, Guang Tu, Lei Liang, and Mei Jiang. Design of power grid environment monitoring system based on wlan. In *Software Engineering (WCSE), 2010 Second World Congress on*, volume 1, pages 30–33, 2010.

Appendix A

Analysis method

I. Control plug comment

```
On  Relay 1
05 00 14 8C 01 60 12 00 01 11 46 90 06 30 31 00 00 00 00 00 00 8B A2
Off  Relay 1
05 00 14 8C 01 60 12 00 01 11 46 90 06 30 30 00 00 00 00 00 00 14 77
On Relay 2
05 00 14 8C 01 60 12 00 01 11 46 90 06 31 31 00 00 00 00 00 00 0A 1D
Off Relay 2
05 00 14 8C 01 60 12 00 01 11 46 90 06 31 30 00 00 00 00 00 00 95 C8
```

II. Login to Mysql

```
String url = "jdbc:mysql://localhost/test";
String user = "root";
String password = "0000";
Class.forName("com.mysql.jdbc.Driver");
Connection conn = DriverManager.getConnection(url, user, password);
if (!conn.isClosed()) {
    System.out.println("Connecting Database");
        }
Statement statement = conn.createStatement();
DatagramSocket serverSocket = new DatagramSocket(5001);
```

III. Connect Server with ASCII code

```
(checkData.equals("P"))   //P=50  CO2
(checkData.equals("B"))   //B=42  Temperature
(checkData.equals("A"))   //A=30  RH
```

IV. Write data to Mysql

```
Date dNow = new Date();
         SimpleDateFormat formatter = new SimpleDateFormat("HH:mm:ss ");
         System.out.print(checkData);

         String dts=formatter.format(dNow);
           int hour = Integer.parseInt(dts.substring(0,2));
           int min =  Integer.parseInt(dts.substring(3,5));
           int time = hour*100 + min;
           ..
           ..
           ..
           ..
           ..
           statement.executeUpdate("INSERT INTO co2date "
           + "VALUES (null,'"+formatter.format(dNow)+"', '"+co2Value+"')");
```

V. Turn on power of plug

```
int[] i_on2  = {0x05,0x00,0x14,0x8C,0x01,0x60,0x12,0x00,0x01,0x11,0x46,
            0x90,0x06,0x31,0x31,0x00,0x00,0x00,0x00,0x00,0x00,0x0A,0x1D};
int[] i_off2 = {0x05,0x00,0x14,0x8C,0x01,0x60,0x12,0x00,0x01,0x11,0x46,
            0x90,0x06,0x31,0x30,0x00,0x00,0x00,0x00,0x00,0x00,0x95,0xC8};
..
..
..
..

if(time>=1400 && time<1900){
    sendPacket = new DatagramPacket(b_on2, b_on2.length, IPAddress, port);
    System.out.println ("peak time" + hour + ":" + min +"\r\n");
    serverSocket.send(sendPacket);
                           }
if(co2Value>=1250)
    sendPacket = new DatagramPacket(b_on2, b_on2.length, IPAddress, port);
    serverSocket.send(sendPacket);
```

VI. Turn off power of plug

```
DatagramPacket sendPacket
= new DatagramPacket(b_off2, b_off2.length, IPAddress, port);
```

Appendix B

Show the gas data on Web

I. Compiler PHP code on apache server

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Highcharts Example</title>

<!-- 1. Add these JavaScript inclusions in the head of your page -->
<script type="text/javascript" src=
    "http://ajax.googleapis.com/ajax/libs/jquery/1.6.1/jquery.min.js"></script>
<script type="text/javascript" src="Highcharts/js/highcharts.js"></script>

<!-- 1a) Optional: add a theme file -->
<!--
<script type="text/javascript" src="../js/themes/gray.js"></script>
-->

<!-- 1b) Optional: the exporting module -->
<script type="text/javascript" src="Highcharts/js/modules/exporting.js"></script>

<!-- 2. Add the JavaScript to initialize the chart on document ready -->
<script type="text/javascript">
..
..
..
title: {
                text: 'Realtime co2 data'
                },
                xAxis: {
                type: 'datetime',
                tickPixelInterval: 150
                                },
                yAxis: {
```

```
title: {
        text: 'ppm'
                },
        max: 2000,
            tickPositions:10,
                                min: 0,
                                plotLines: [{
                                        value: 0,
                                        width: 1,
                                        color: '#808080'
                                }]
                        },
```

Appendix C

Environment Monitoring Android App

I. Show dialog when click button1

```
private void choose(){
   final CharSequence[] items = {"CO2","Temperature","Humidity"};

   AlertDialog.Builder builder = new AlertDialog.Builder(this);
   builder.setTitle選擇單位("");
   builder.setItems(items, new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int item) {
          Intent intent = new Intent();
                  intent.setClass(MonitorrrActivity.this, searchActivity.class);
          Bundle bundle = new Bundle();
          bundle.putInt("TYPE", item);
          intent.putExtras(bundle);
                  startActivity(intent);
           }
       });
       builder.create().show();
```

II. Use webview to show the graph on web

```
int result = bundle.getInt("TYPE");
       if(result==0){
Toast.makeText(getApplicationContext(), "CO2", Toast.LENGTH_SHORT).show();
         String myURL = "http://140.128.101.173/goosensor/co2data.php";
           WebView myBrowser=(WebView)findViewById(R.id.webView1);
           WebSettings websettings = myBrowser.getSettings();
           websettings.setJavaScriptEnabled(true);
```

```
            myBrowser.setWebChromeClient(new WebChromeClient() {

@Override
public void onGeolocationPermissionsShowPrompt(String origin,
                        GeolocationPermissions.Callback callback) {
              callback.invoke(origin, true, false);
                        }
                });
                myBrowser.loadUrl(myURL);



        }else if(result==1){
Toast.makeText(getApplicationContext(), "Temperature", Toast.LENGTH_SHORT).show();
                String myURL = "http://140.128.101.173/goosensor/tempdata.php";
                  WebView myBrowser=(WebView)findViewById(R.id.webView1);
                  WebSettings websettings = myBrowser.getSettings();
                  websettings.setJavaScriptEnabled(true);

                   myBrowser.setWebChromeClient(new WebChromeClient() {

@Override
public void onGeolocationPermissionsShowPrompt(String origin,
                       GeolocationPermissions.Callback callback) {
                   callback.invoke(origin, true, false);
                                }
                        });
                        myBrowser.loadUrl(myURL);
```

III. Show Environmental values when click button2

```
HttpClient client = new DefaultHttpClient();
HttpGet request = new HttpGet("http://140.128.101.173/goosensor/webtext.php");
HttpResponse response = client.execute(request);
String retSrc = EntityUtils.toString(response.getEntity());

Log.d("AAAA", "retSrc:" + retSrc);

String[] array = retSrc.split("<br>");
tv1.setText("CO2  " + array[0]+" ppm");
tv1.setTextSize(45);
tv2.setText("Temperature" + array[1] + "℃"");
tv2.setTextSize(45);
tv3.setText("Humidity  " + array[2] + "%");
tv3.setTextSize(45);
```