

Design and Implementation of Data Processing Middleware and Management System for IoT based Services

Yon-Sik Lee*, Young-Chae Mun**

Abstract

Sensor application systems for remote monitoring and control are required, such as the establishment of databases and IoT service servers, to process data being transmitted and received through radio communication modules, controllers and gateways. This paper designs and implements database server, IoT service server, data processing middleware and IoT management system for IoT based services based on the controllers, communication modules and gateway middleware platform developed. For this, we firstly define the specification of the data packet and control code for the information classification of the sensor application system, and also design and implement the database as a separate server for data protection and efficient management. In addition, we design and implement the IoT management system so that functions such as status information verification, control and modification of operating environment information of remote sensor application systems are carried out.

The implemented system can lead to efficient operation and reduced management costs of sensor application systems through site status analysis, setting operational information, and remote control and management.

▶ Keyword: Sensor application system, IoT based monitoring and control, Middleware, IoT management system

1. Introduction

센서 응용시스템들은 다양한 설치 장소와 측정 데이터의 다양성으로 인하여 IoT 기반 서비스를 통한 원격 모니터링 및 제어기능의 지원이 필수적으로 요구된다. 이러한 센서 응용시스템과 같은 IoT 기반 서비스들은 측정 데이터의 수집, 저장, 관리, 응용을 위한 서비스 서버와 DB가 필수적으로 요구되며 [1,2], 이들의 연동과 무선통신 모듈, 컨트롤러 및 게이트웨이 미들웨어 플랫폼 등의 기능과 역할의 지원을 통하여 다양한 송수신 데이터를 요구에 적합하도록 처리한다.[1,3].

IoT 기반 서비스의 중심역할을 수행하는 IoT 서비스 서버는 측정 데이터를 게이트웨이를 통하여 수신·분류하여 DB 서버에

저장하며[3,4], IoT 관리시스템으로부터 제어 데이터를 수신·분석하여 해당 게이트웨이를 통하여 센서 응용시스템의 제어기능을 수행한다.[5]. 서버로부터 수집된 측정데이터와 관리를 위한 정보들을 저장하기 위한 DB는 IoT 서비스 서버에서만 직접 접근이 가능하도록 별도의 서버로 구축함으로써[3,6,11], 관리자가 효율적 안정적으로 모니터링 및 제어시스템을 운영할 수 있도록 한다.

본 논문에서는 게이트웨이, 컨트롤러, IoT 서비스 서버 및 IoT 관리시스템 등에서 전송되는 데이터들을 분석, 저장, 처리 및 보호를 위한 별도의 DB 서버를 구축하고, 이를 기반으로 IoT 기반 서비스를 위하여 필수적으로 요구되는 데이터 처리

• First Author: Yon-Sik Lee, Corresponding Author: Young-Chae Mun

*Yon-Sik Lee (yslee@kunsan.ac.kr), School of Computer Information and Communication Engineering, Kunsan National University

**Young-Chae Mun (ycmun@naver.com), Hailight Co., Jeonju, Korea

• Received: 2018. 12. 24, Revised: 2019. 01. 28, Accepted: 2019. 01. 28.

• This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (018R1D1A1B07051045)

미들웨어와 IoT 관리시스템을 설계 및 구현한다. 또한, 구현된 기술들의 검증과 이를 통한 IoT 기반 서비스시스템 구축의 유용성 검증을 위하여, 센서 기반 태양광 가로등시스템에 구현 기술을 실질적으로 적용하여 실험하고 그 결과를 보인다.

II. Design key points of IoT-based services

2.1 Packet protocol for data transfer

데이터 전송을 위하여 필요한 데이터 패킷 프로토콜(packet protocol)은 전송할 데이터의 시작, 시간 정보, 센서 응용시스템의 컨트롤러 식별번호, 전송 데이터 구분코드, 측정/제어/설정 데이터, 전송할 데이터의 끝 등을 포함하여 Table 1과 같이 설계한다.

Table 1. Packet protocol for data transmission

item	SDT	Date	Cont_ID	Control code	Data	EDT
Size (byte)	1	14	10	1	2~16	1
Type	int	char	char	char	char	int

전송 데이터의 시작과 끝을 알려 주는 ‘SDT’와 ‘EDT’의 값은 게이트웨이나 IoT 서비스 서버에서 실시간으로 송수신 되는 데이터를 목적에 맞게 분류하기 위하여, Table 2와 같이 컨트롤러에서 전송하는 값과 사용자가 전송하는 값으로 구분하여 설정한다.

Table 2. SDT and EDT code table

Value	SDT/EDT	Description
0x02	SDT	Notifies the start of data transfer from the controller
0x03	EDT	Notifies the end of data transfer from the controller
0x04	C_SDT	Notifies the start of data transfer from the customer
0x05	C_EDT	Notifies the end of data transfer from the customer

‘Date’는 데이터 전송을 위한 패킷 생성 시각을 초(sec.)단위로 표현하고, ‘Cont_ID’는 센서 응용시스템의 컨트롤러를 식별할 수 있도록 10 Byte크기의 데이터로 설정한다. ‘Control code’는 센서 응용시스템의 컨트롤러와 IoT 관리시스템에서 전송된 데이터를 다음 Table 3과 같이 구분하기 위하여 사용하며, ‘Data’는 각 제어코드에 해당하는 데이터 수송을 위한 2 Byte에서 16 Byte까지의 가변형 구조로 설계한다.

Table 3. Classification of the control code

Control code	Description
EVENT	Control code for identifying all and part of the data measured on the controller
GET	Control code to obtain configuration information for the requested controller from the administrator
SET	Control code for changing controller configuration information by administrator
CMD	Control code for administrator to send sensor application system control commands to the controller

본 논문에서는 구현 기술들의 검증 및 실험을 위하여 태양광 가로등시스템을 실질적 센서 응용시스템으로 구축하여 컨트롤러에서 주기적으로 측정된 데이터를 사용한다. 이는 발전된 전압, 충전 전류, 배터리 전압과 충전 전류, 컨트롤러와 배터리 온도 등의 데이터 순으로 각각 요구되는 크기만큼 할당하여 구성하며, 2 Byte 크기의 데이터는 충·방전 이벤트가 발생하는 경우 상태 및 환경 정보를 설정하거나 설정 정보를 전송하기 위한 데이터이다.

2.2 Database

센서 응용시스템의 서버로부터 수집된 데이터와 관리를 위한 정보들의 저장, 검색 및 분석 효율성을 위하여, 또한 게이트웨이나 관리시스템 등에서 DB에 직접 접속으로 인한 보안 문제 발생을 방지하기 위하여 IoT 서비스 서버와는 별도의 서버에 DB를 물리적으로 분리하여 구축할 필요가 있다[3,6]. DB는 권한 코드로 구분된 모든 사용자 등록 정보 저장 테이블, 사용자들의 소속 정보 저장 테이블, 센서 응용시스템 구성 정보 저장 테이블, 컨트롤러에 대한 정보 저장 테이블, 컨트롤러에서 측정된 데이터 저장 테이블 등으로 구성되며, 다음 Figure 1은 각 구성 테이블들 간의 관계 모델을 나타낸다.

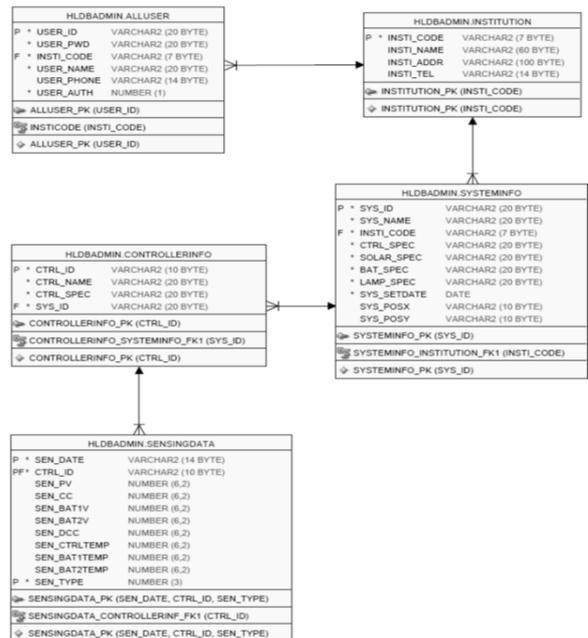


Fig. 1. Diagram of database relationship model

2.3 Data processing middleware

센서 응용시스템의 컨트롤러에서 측정된 데이터를 DB에 저장하고, 사용자의 검색, 관리 및 응용을 위하여 IoT 서비스 서버는 데이터 처리 미들웨어가 필요하다[7,8]. IoT 서비스 서버에서 동작하는 데이터 처리 미들웨어는 게이트웨이에서 전송된 데이터를 분석하여 DB에 저장하거나 IoT 관리시스템으로 데이터를 전송하는 기능을 포함하며, IoT 관리시스템에서 모니터링 데이터를 요청한 경우 DB에서 요청한 컨트롤러의 데이터를 전송하고, 컨트롤러 환경 설정 및 상태 정보나 제어를 위한 데이터를 요청한 경우에는 해당 컨트롤러가 포함된 게이트웨이에 데이터를 전송하는 기능을 포함한다[9,10,11]. Figure 2는 IoT 서비스 서버에서의 데이터 처리 미들웨어의 역할과 데이터 분석, 처리, 저장 및 검색 등의 기능 모듈들의 연관성을 나타내는 그림이다.

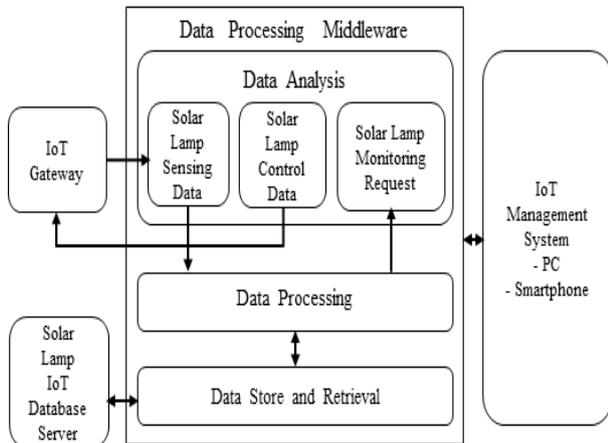


Fig. 2. Roles and features of data processing middleware

데이터 처리 미들웨어는 게이트웨이와 IoT 관리시스템에서 전송된 데이터를 분석하여, 센서 응용시스템의 측정 데이터는 데이터 패킷 항목별로 구분하여 DB에 저장하고 제어 데이터는 해당 클러스터를 관장하는 게이트웨이에 전송한다. 또한 IoT 관리시스템에서 임의의 컨트롤러에 대한 모니터링 데이터를 요청할 경우, 요청 대상 컨트롤러의 데이터를 수집·전송하여 관리자가 요청 정보를 확인할 수 있도록 한다.

III. Design and implementation

3.1 Implementation of database server and data processing middleware

본 장에서는 II장의 IoT 기반 서비스 구축을 위한 설계 주요 사항을 기반으로 DB 서버를 구현하고, IoT 관리시스템과 DB 서버 연결정보 설정 및 접속기능, 소켓 통신 서버기능, 모듈 간의 전송 데이터 분석 및 처리기능과 IoT 관리시스템에서 전송된 데이터를 게이트웨이에 전송하는 중계기능을 포함하여 데이

터 처리 미들웨어를 설계 및 구현한다.

다음 Figure 3은 본 논문에서 설계한 데이터 처리 미들웨어의 동작 흐름도이다.

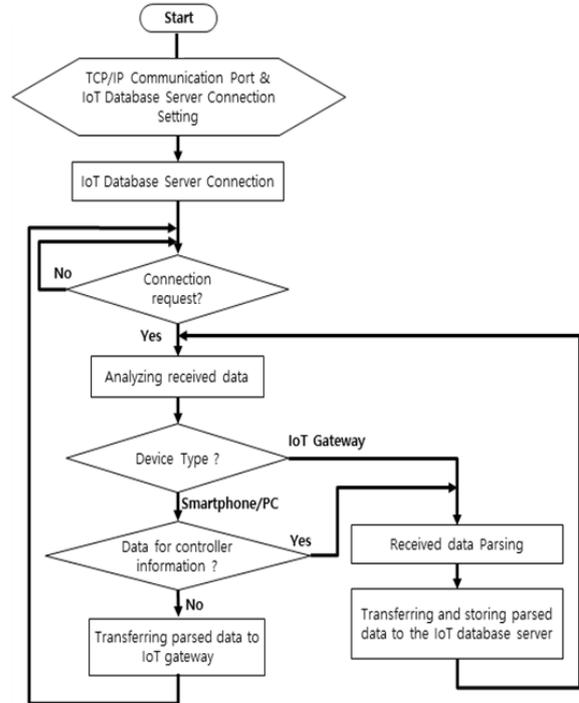


Fig. 3. Flow chart for data processing and store

데이터 처리를 위하여 먼저, TCP/IP 통신 포트와 DB 서버에 접속하기 위한 설정을 통하여 서버에 접속한 후 게이트웨이나 IoT 관리시스템으로부터 연결 요청을 기다린다. IoT 서비스 서버는 게이트웨이나 IoT 관리시스템에 연결하고, 이들은 서버에 수집된 데이터를 전송하거나 DB 서버에 저장된 데이터를 요청한다. IoT 서비스 서버는 수신된 데이터를 분석하고, 이를 DB에 저장하거나 DB의 데이터를 이들 장치들에게 전송한다. IoT 서비스 서버로 전송된 데이터는 컨트롤러에서 측정된 데이터와 IoT 관리시스템에서 모니터링이나 제어를 위하여 요청하는 데이터로 구분된다. 컨트롤러에서의 측정 데이터는 실시간 데이터와 저장된 파일 데이터로 구분되며, 이들은 DB 서버로 전송하여 저장하고, IoT 관리시스템에서 전송된 데이터는 게이트웨이로 전송하여 해당 센서 응용시스템 제어에 사용하고, 검색이나 모니터링을 위한 데이터는 DB 서버에서 필요한 데이터를 검색한 후 IoT 관리시스템에 전송한다.

다음은 각 기능들에 대한 세부적인 구현 내용이다.

▶ DB 서버 연결 정보설정 및 접속기능

DB 서버는 Oracle을 사용하여 별도의 서버로 구축하고, IoT 서비스 서버에서만 연결이 가능하도록 구현하였다. DB 서버의 IP주소, DB 이름, 접속 ID와 P/W 등의 연결 정보를 설정 후 연결함으로써 센서 응용시스템의 컨트롤러 데이터를 저장 및 검색한다. DB 서버의 구축 환경은 다음 Table 4와 같다.

Table 4. Database server implementation environment

Item	Specification
CPU	Intel Xeon Processor E5-2560V2 (2.6GHz, 8Core, 16 thread) x 2EA
Memory	32GB(2 x 16GB) PC3L-12800 1600MHz
Disk	SAS 1TB(7.2K 6Gbps 2.5inch) x 6EA (Raid 6)
Controller	Adaptec Raid Card
Power	740W x 2EA(dual)
OS	Windows Server Standard 2012 R2 64Bit
DBMS	Oracle DBMS Standard Edition ONE

▶ 소켓 통신 서버기능

게이트웨이나 IoT 관리시스템이 IoT 서비스 서버에 접속하기 위한 서버 소켓 포트를 생성하고, 연결을 요청할 경우 디바이스 별로 소켓을 생성하여 연결한다.

▶ 게이트웨이에서 전송된 데이터 분석·처리기능

전송된 데이터를 항목별로 분류하고, 제어코드를 확인하여 코드가 IoT 관리시스템에서 요청한 데이터일 경우 해당 IoT 관리시스템에 전송하고, 코드가 이벤트코드인 모니터링 데이터이면 DB에 저장한다.

▶ IoT 관리시스템에서 전송된 데이터 분석·처리기능

게이트웨이에 전송되어 파일로 저장된 모니터링 데이터는 DB에 저장하고, 컨트롤러 제어 및 환경 설정 데이터는 해당 게이트웨이에 전송하며, DB 검색 요청 데이터는 검색 결과를 요청한 디바이스에 전송한다.

Figure 4는 게이트웨이, IoT 관리시스템, DB 서버와 통신하면서 IoT 서비스 서버로 전송되는 데이터 처리를 위한 미들웨어의 클래스 다이어그램이다.

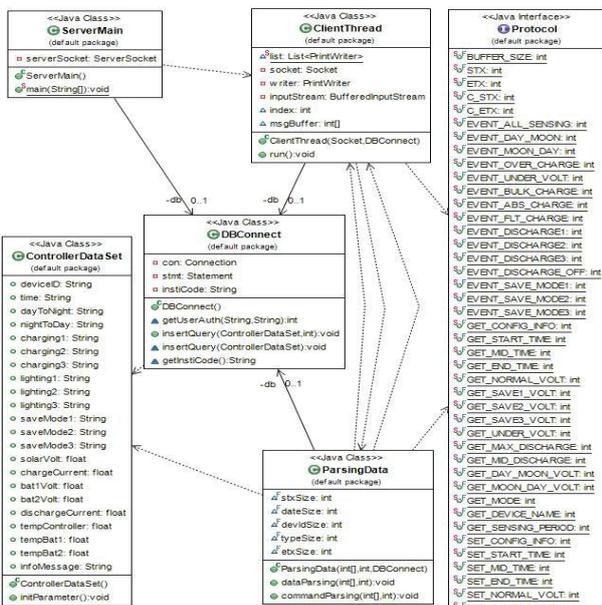


Fig. 4. Class diagram of middleware for data processing

클래스 *ServerMain*에서 미들웨어를 실행시키며, 클래스 *DBConnect*를 이용하여 DB 서버에 접속하도록 한다. 게이트웨이나 IoT 관리시스템에서 접속을 요청할 경우 해당 디바이스 객체를 클래스 *ClientThread*를 이용하여 생성함으로써 동시 처리가 가능하도록 구현하였다. 클래스 *ParsingData*는 IoT 서비스 서버로 전송된 데이터를 분석하여 DB에 저장하거나 게이트웨이에 전송한다. 클래스 *ControllerDataSet*는 측정 데이터를 컨트롤러 객체로 처리하기 위한 것이며, *Protocol* 인터페이스는 데이터 전송을 위한 패킷의 제어코드에 접근하기 위한 인터페이스기능을 수행한다.

3.2 Implementation of IoT management system

IoT 관리시스템 응용은 모듈간의 연결을 위한 통신기능, 사용자 관리기능, 데이터 분석 및 처리기능, 권한 별 데이터 표시기능, 센서 응용시스템 제어기능 등을 포함하여 IoT 관리 PC에서 응용프로그램으로 구현하였다. Figure 5는 IoT 관리시스템 응용을 위한 클래스 다이어그램이다.

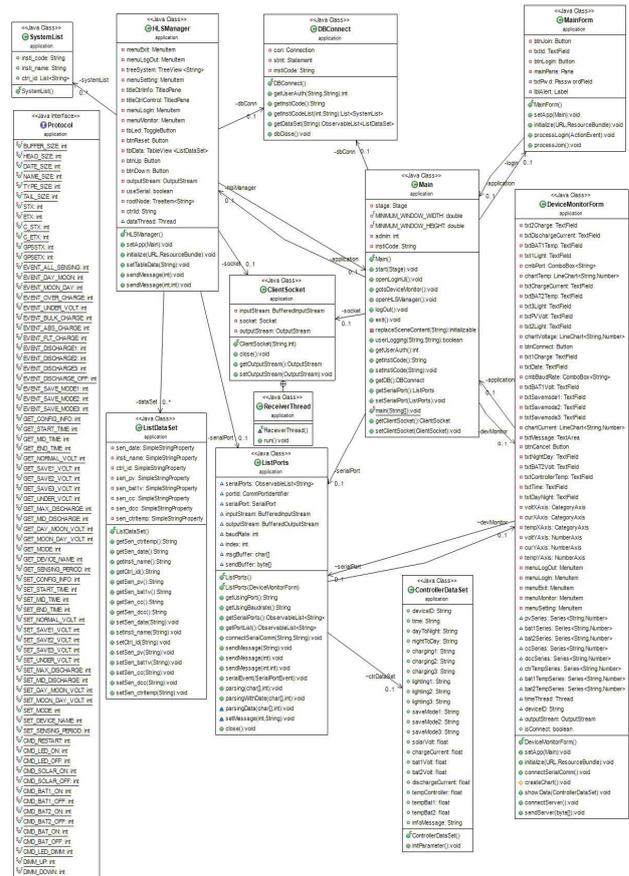


Fig. 5. Class diagram for application of IoT management system

클래스 *Main*에서 응용프로그램이 실행이 시작되고, 클래스 *DBConnect*에서 로그인 과정이 이루어진다. 로그인 후 클래스 *ClientSocket*을 사용하여 IoT 서비스 서버와 연결 및 데이터 전송을 실행하며, 클래스 *ReceiverThread*를 이용하여 IoT 서

비스 서버로부터 데이터를 수신한다. 클래스 *SysManager*를 이용하여 관리를 위한 소속 기관별 센서 응용시스템 목록을 구성하고, 클래스 *DeviceMonitorForm*을 이용하여 목록에서 선택된 시스템의 측정 데이터를 테이블 형태로 제공한다. 또한 클래스 *SystemList*를 이용하여 관리자 소속 기관에 설치된 센서 응용시스템 목록을 제공하고, 클래스 *ListDataSet*를 이용하여 컨트롤러에 제어 및 관리 명령을 설정할 수 있도록 한다. 또한, 클래스 *ListPorts*를 이용하여 시리얼 통신을 설정하고, 전송된 컨트롤러의 측정 데이터를 클래스 *ControllerDataSet*을 이용하여 컨트롤러 객체로 관리한다. *Protocol* 인터페이스는 IoT 서비스 서버의 데이터 처리 미들웨어의 *Protocol* 인터페이스를 재사용하도록 한다.

IV. Experimental results

4.1 Experiment on the operation of IoT service server

IoT 서비스 서버 구축 환경은 다음 Table 5와 같이 구축하고, IoT 기반 서비스를 위한 부품 모듈간의 통신과 발생 데이터 처리를 위한 미들웨어를 실행하도록 구현하였다.

Table 5. IoT service server implementation environment

Item	Specification
CPU	Intel Xeon Processor E3-1220V3 3.1GHz, 4Core, 4 thread
Memory	8GB PC3-12800E ECC Unbuffered
Disk	SATA3 1TB x 2EA
Power	500W
OS	Windows Server Std 2012 R2 64Bit
Tool	Eclipse
Language	JAVA

먼저 DB 서버에 연결하고 서버 소켓을 생성한 후, 게이트웨이에 접속을 요청하면 접속이 가능한 소켓을 생성하고 IoT 서비스 서버에 연결된다. 게이트웨이는 여러 개의 센서 응용시스템에서 컨트롤러로부터 측정된 데이터를 수집하여 IoT 서비스 서버로 전송하며, 패킷 형태로 전송된 스트림 데이터는 분석 과정을 통하여 분리되어 Query 실행을 통하여 DB에 저장된다.

다음 Figure 6은 게이트웨이로부터 수신된 데이터를 분석한 후 DB에 저장하는 실험 결과를 나타낸다.

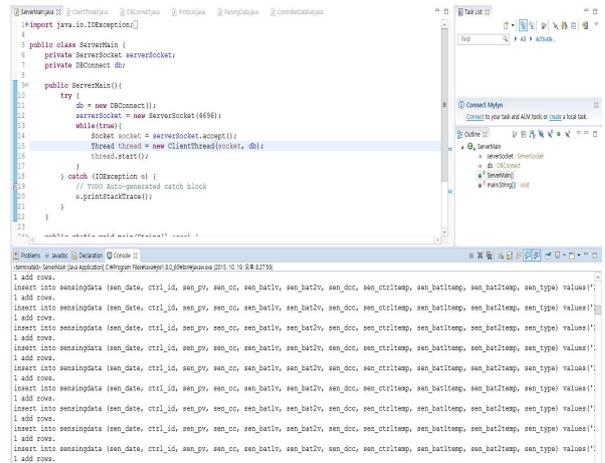


Fig. 6. Contents saved in database server after data analysis

위와 같은 실험 결과를 통하여 이기종 통신 환경에서 전송되는 데이터규격 정의와 정보 구분을 위한 제어코드 적용의 적정성과, 데이터 처리 미들웨어를 통한 DB 서버 연결 정보 설정 및 접속기능과 데이터 분석 및 처리기능, DB 구축에 의한 데이터의 효율적 저장 및 관리 응용성을 확인할 수 있다.

4.2 Experiments on the operation of IoT management system applications

IoT 기반 서비스를 위한 모니터링 및 제어시스템은 센서 응용시스템의 컨트롤러와 통신모듈, 게이트웨이, IoT 서비스 서버, DB, IoT 관리시스템 등으로 구성하고, 개발 시스템의 적용성과 유효성 검증을 위하여 IoT 관리 PC 환경에서 동작 실험을 진행하였다.

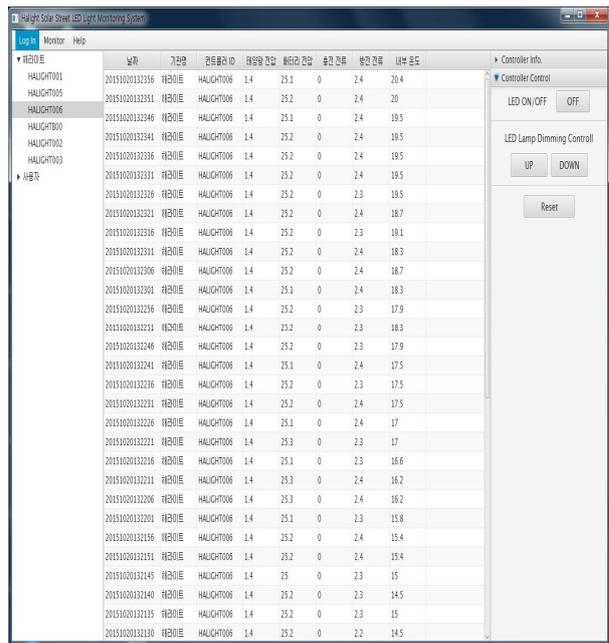


Fig. 7. Screen for data acquisition and controller control

다음 Figure 7은 IoT 관리 PC 응용을 실행하여 관리자 로 로그인 한 화면이다. 사용자 인터페이스는 설치 현장 별 설치된 센서 응용시스템 목록과 선택된 센서 응용시스템에서 전송된 컨트롤러의 측정 데이터 표시기능을 포함하고, 센서 응용시스템 제어기능과 환경 설정기능이 가능하도록 구성하였다. 센서 응용시스템의 연동 장치 제어기능 (점등 및 소등, 디밍 조절, 컨트롤러 초기화 관련 명령 등)은 관리자 등급에서 가능하도록 *Controller Control* 탭을 사용하여 제한적으로 구현하고, *Controller Info.* 탭을 사용하여 센서 응용시스템의 컨트롤러 환경 설정 정보 표시와 이를 변경 및 전송할 수 있는 기능을 구현하였다.

통신기능은 서비스 서버와 연결을 위한 TCP/IP 통신과 ZigBee 모듈과 연결을 위한 시리얼 통신으로 구현하였으며, 통신 모듈을 연결하여 수신된 측정 데이터 (태양광 모듈 전압, 배터리 전압 및 전류, 컨트롤러 및 배터리 온도)를 그래프로 표시하고 IoT 서비스 서버에 데이터를 전송하도록 한다. 또한 센서 응용시스템에서 실시간으로 측정된 데이터와 이벤트가 발생한 데이터는 이벤트 내용과 함께 다음 Figure 8과 같이 보여준다.

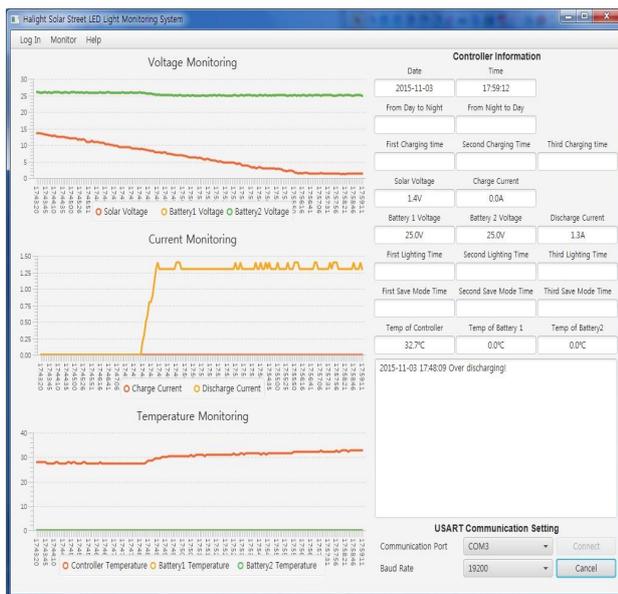


Fig. 8. Screen of serial communication settings and controller data received

위와 같은 IoT 관리시스템 응용에 대한 동작 실험을 통하여 관리자의 사용자 환경에서 센서 응용시스템의 상태 정보 확인과 운영환경 정보 설정 및 변경 등이 가능함을 보임으로써, 무선 통신 네트워크 환경에서 IoT 기반 서비스를 이용한 원격 운영 관리의 유효성을 확인할 수 있다.

V. Conclusions

센서 기반 응용시스템은 불특정 다수의 지역에 설치되어 통

신 및 유선 가설비용 발생 문제와 단일 무선통신 네트워크 구축으로 각 클러스터의 데이터 처리를 위한 서버 운영이 요구되는 문제점을 가지고 있으며, 측정 데이터의 다양성으로 인한 IoT 기반의 원격 모니터링 및 제어기능의 지원이 필수적으로 요구된다.

이러한 IoT 기반 서비스들은 측정 데이터의 처리 및 응용을 위한 서비스 서버와 DB, 데이터 처리 미들웨어, 무선통신 모듈, 컨트롤러 및 게이트웨이 미들웨어 플랫폼 등 구성요소들의 연동된 기능과 역할 지원을 통하여 다양한 데이터를 요구에 적합하도록 처리할 수 있다.

본 논문은 기 개발된 센서 응용시스템의 컨트롤러, 데이터 송수신을 위한 통신 모듈과 게이트웨이 미들웨어 플랫폼을 기반으로, IoT 기반 서비스를 위한 DB 및 서버, IoT 서비스 서버, 데이터 처리 미들웨어 및 IoT 관리시스템 응용을 설계 구현하였다. 이를 위하여, 이기종 통신 환경에서 전송되는 데이터의 규격과 센서 응용시스템의 제어, 설정 및 상태정보를 구분하기 위한 제어 코드를 정의하였고, 센서 응용시스템에서 측정된 데이터를 저장하기 위한 DB를 별도의 서버로 구축하여 데이터의 효율적 관리를 유도하였다. 또한, IoT 관리시스템의 사용자 디바이스에서 원격지의 센서 응용시스템의 상태 정보 확인, 제어, 운영 환경 정보 설정 및 변경 등의 기능을 수행하는 응용을 개발하고, 구현된 모듈들의 기능 동작을 실험을 통하여 평가하였다.

개발 시스템은 각 클러스터들로 구성된 센서 응용시스템에서 수집된 데이터를 저장 및 관리하고, 이를 통하여 현장의 상황 파악 및 분석, 운영 정보 설정, 원격 제어 및 관리 등을 효과적으로 수행함으로써, 센서 응용시스템들의 운영 및 비용 측면의 효율성을 제공한다.

REFERENCES

- [1] A. Mahmud, et. al., "A Complete Internet of Things Platform for Structural Health Monitoring," 2018 IEEE 4th World Forum on Internet of Things, pp.275-279, 2018.
- [2] M. Kim, et al., "Universal Interface for IoT Platform," Journal of KIPS, 7(1), pp.19-24, 2018.
- [3] H. O. Song, "IoT Automatic Management System based on Situational Task Control," Ph.D. thesis of Paijae University, 2017.
- [4] M. H. Park, et al., "A Study on the Development of Energy IoT Platform," Journal of KIPS, 5(10), pp.311-318, 2016.
- [5] J. B. Kim, "IoT-based Home Energy Management System and A Lightweight Household Photovoltaic System Over Dynamic Home Area Networks," Ph.D. thesis of Joongang University, 2016.
- [6] C. Lee, Y. Lai, "Design and Implementation of a Universal Smart Energy Management Gateway based on the IoT

- Platform,” 2016 IEEE Int’l Conference on Consumer Electronics, pp.67-68, 2016.
- [7] H. J. Lee, “A Study on the Application System of the IoT based on the Super Low Power Wireless Sensor Network,” Ph.D. thesis of Chonnam National University, 2017.
- [8] A. Jain, C. Nagarajan, “Efficient Control Algorithm for a Smart Solar Street Light,” 2015 9th Int’l Conference on Next Generation Mobile Applications, Services and Technologies, pp.376-381, 2015.
- [9] J. J. Lee, et al., “IoT-based mobile smart solar power monitoring system,” Journal of IEEK, 54(8), pp.55-64, 2017.
- [10] M. Saifuzzaman, et. al., “IoT Based Street Lighting And Traffic Management System,” 2017 IEEE Region 10 Humanitarian Technology Conference, pp.121-124, 2017.
- [11] B. W. Park, et al., “Development of Integrated Photovoltaic Monitoring System Using ZigBee Communication,” Journal of KIIEE, 31(7), pp.56-62, 2017.

Authors



Yon-Sik Lee received the B.S. and M.S. degrees in Computer Science from Chonnam National University, Korea, in 1982 and 1984, respectively. And, his Ph.D. degree in Computer Application Engineering from Chonbuk National

University, Korea, in 1994. Dr. Lee joined the faculty of the School of Computer Information and Communication Engineering, Kunsan National University, Kunsan, Korea, in 1986. He is interested in sensor network middleware, active rule system, agent system and cloud computing.



Young-Chae Mun received the B.S. and M.S. degrees in Computer Information Science from Kunsan National University, Korea, in 2007 and 2009, respectively. And, his Ph.D. degrees in Computer Information Engineering from Kunsan National

University, Korea, in 2018. Dr. Mun joined the technical manager at Hailight Company, Jeonju, Korea, in 2014. He is interested in wireless sensor network, IoT service management and software agent system.