ISSN 1975-8359(Print) / ISSN 2287-4364(Online)
The Transactions of the Korean Institute of Electrical Engineers Vol. 63, No. 8, pp. 1116~1120, 2014
http://dx.doi.org/10.5370/KIEE.2014.63.8.1116

오픈소스를 활용한 지능형 수요반응 플랫폼 개발

Development of Automated Demand Response Platform Using Open Source Code

윤 재 원* · 이 인 규† · 최 중 인*
(Jae-weon Yoon · Ingyu Lee · Jung-In Choi)

Abstract - With the shift of the energy paradigm from supply side management to demand side management, demand resource management and demand response plays an important role in the energy industry. As a consequency, a lot of researches have been done to provide a suitable demand response system. However, most of the demand response systems are based on the propriety products that cannot be modified. In this paper, we are proposing an automated demand response system using an EnerNOC provided open source code. We implemented the demand response server (VTN) and demand response client (VEN), and validated the OpenADR2.0 compliances using the open source code. We also used an Arduino microcontoller to demonstrate the communication schemes to control various devices.

Key Words : Demand response, OpenADR, AutoDR, VTN, VEN

1. 서 론

전기에너지 수요는 해마다 늘어가고 있으나 공급은 이를 뒷받침하여주지 못하여, 에너지 절감을 위한 다양한 기술이 연구되어 왔다. 이러한 추세에 가장 주목받는 기술이 ICT를 접목하여 필요시에만 부하를 줄이는 방법으로 에너지 공급의 패러다임을 변화시키는 수요반응 (Demand Response) 이다. 더욱이 석유, 천연가스, 석탄 등의 화석 연료비의 급증으로 수요관리의 가치가 증대되었고, 수요자원이 지속가능한 환경 규제에 대응하는 친환경 에너지로 분류되어 최근에 더욱더 각광을 받고 있다[1,2,3]. 수요반응이 활성화 되면 피크 감소에 따른 발전설비 투자가 지연되고, 피크 및 전력수요 감소에 따라 전력시스템의 신뢰도가 증가되면서 안정적인 전력수급이 가능해진다[1,2,3]. 또한, 도매 전력시장 가격감소에 따른 소비자 편익이 증가되고, 스마트 미터, 수요반응을 위한 기기, 전력저장장치 등 에너지 효율화 관련 산업의 성장을 기대할 수 있다[1,2,3,4].

수요반응서비스는 전력망의 정보에 반응하여 소비자가 전력사용을 조정함으로써 생겨나는 새로운 부가가치를 기반으로 하는 모델이다. 기존의 일방적 공급중심의 전력망 시스템에서는 구현하기 어려운 서비스 모델로서 양방향의 스마트그리드 인프라 환경에서 새롭게 창출되는 것이다. 즉, 전력망 운영자 입장에서 전력의 거래가격이 예비율이 낮은 피크 시에는 매우 비싸고 예비율이 높은 오프피크 시에는 저렴하기 때문에, 소비자가 피크 시에 전력사용을 줄일 경우 그로 인해 발생하는 전력망에 대한 효용성이 전력요금절감액의 5배에서 10배이상 증가하게 되며 이것이 새로운 부가가치를 창출하게 되는 것이다. 이러한 새로운 가치를 효율적이고 공정하며 투명하게 소비자에게 재분배하는 기반이 스마트그리드이고 이것이 핵심서비스인 수요반응서비스이다[1,2,3].

그림 1은 ICT를 활용한 수요반응의 동작원리를 개념적으로 보여준다[1,2,6,7]. 전력 현황 상황실에서 현재의 전력 사용 현황을 모니터링하면서 전력사용량을 예측하여 전력 예비율이 기준 이하에 도달할 것으로 판단되면 (피크전력을 줄여야하는 상황이 발생 시) 수요반응서버에서 수요관리 이벤트를 발령하고 각 수요관리 참여자가 이벤트에 응답하여 피크전력을 줄일 수 있도록 한다. 수요반응 참여자는 미리 정해진 약정에 의거하여 수요관리 이벤트에 참여한 결과를 보상받는다[1,2,10].

본 논문에서는 EnerNoc에서 제공하는 수요반응 오픈소스 플랫폼을 이용하여 수요반응 서버(VTN: Virtual Top Node) 및 수요반응 클라이언트(VEN: Virtual End Node) 시스템을 구축하고 OpenADR Alliance의 OpenADR2.0 규격으로 이벤트를 송·수신 할 수 있는 방법을 제시하여 수요관리 서비스

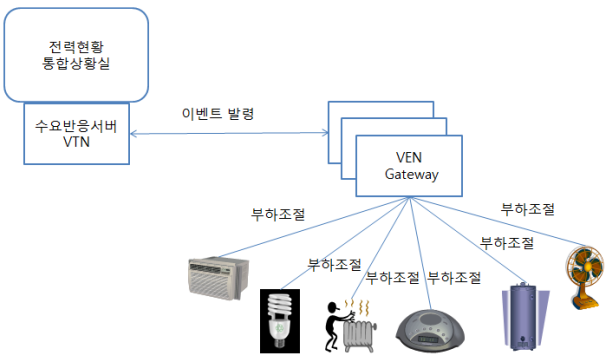


그림 1 수요반응 (Demand Response) 개념
Fig. 1 Demand Response

† Corresponding Author : Advanced Institute of Convergence Technology, Seoul National University, Korea
E-mail : inlee@snu.ac.kr
* Advanced Institute of Convergence Technology, Seoul National University, Korea
Received : June 5, 2014; Accepted : July 23, 2014

를 손쉽게 운영할 수 있는 방안을 제시한다. 또한, 수요반응 서버와 클라이언트 사이의 이벤트 흐름을 마이크로컨트롤러와 연계하여 시각적인 방법으로 표현하고 향후 마이크로컨트롤러로부터 확장되는 다양한 통신방식으로 디바이스를 제어할 수 있는 방안을 모색한다.

본 논문의 구성은 다음과 같다. 2장에서는 수요반응 오픈소스 플랫폼을 활용한 수용반응 서비스 구현 방안 및 수요반응 서비스 프로토타입의 시험 결과를 구체적으로 기술하고, 3장에서는 결론 및 향후 계획을 기술한다.

2. 오픈플랫폼을 이용한 수요반응 시스템

2.1 수요반응서비스 구성

수요반응서비스를 구현하기 위하여는 수요자원을 관리하고 현황 전력 상태를 파악하고 있는 수요반응서버와 이벤트 발생 시 이벤트를 수신하고 기기를 제어할 수 있는 수요반응클라이언트, 서버와 클라이언트간의 통신규약이 필요하다. EnerNOC에서는 수요반응 서버 역할을 하는 VTN (Vitrual Top Node)과 수요반응 클라이언트 역할을 하는 VEN (Virtual End Node)으로 구성되어 있다. 수요반응 서버와 클라이언트를 이용하여 그림 2와 같이 수요반응서비스를 구현할 수 있다.

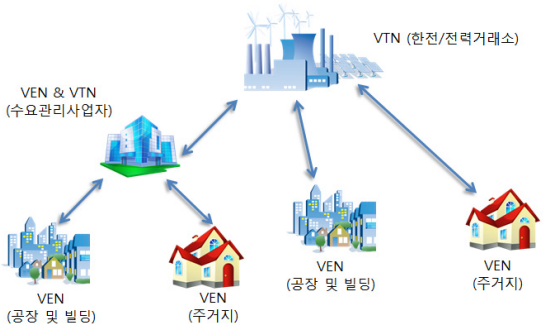


그림 2 수요반응서비스 구성 예
Fig. 2 Example of Demand Response Service

그림 2와 같이 수요반응서비스를 수용가 및 수요관리사업자로 구성한 후, 전력예비율이 일정 수준이하로 떨어질 경우에 수요반응서버에서 수요반응 이벤트를 발령한다. 이때, 수요반응 클라이언트는 약정된 규정에 따라서 수요반응 이벤트에 의거하여 전력사용량을 줄인다. 줄여진 수요자원에 대하여는 차후에 약정에 따라서 보상을 받는다. 또한, 수요반응클라이언트는 수요자원에 대한 상태정보를 수요반응서버에 전달하여 수요반응서버가 현재의 전력사용량 및 감축가능 전력량을 항시 파악할 수 있도록 한다. 수요반응클라이언트의 경우에는 기존의 외부 시스템(BACnet, OPC 등)과도 연동이 되어야 장비제어가 가능하다.

수요반응서비스를 제공하는 상용제품의 경우에는 프로토콜이 폐쇄적으로 작성이 되어 기능의 추가나 수정 및 보완이 불가능하나, EnerNOC에서 제공하는 오픈소스의 경우 OpenADR2.0 프로토콜을 준수하고 개방형 구조여서 기능의 추가나 변경이 용이하다. 따라서, 본 연구에서는 오픈 프로토콜을 사용하는 EnerNOC을 활용하여 수요관리시스템을 구축하여 수요반응서비스를 구현하고자 한다.

2.2 수요반응 서비스 구현

다음은 EnerNOC에서 공개하고 있는 OpenADR 2.0 수요반응서버 (VTN) 및 클라이언트 (VEN)를 활용하여 수요관리 시스템 프로토타입을 구현하고 마이크로컨트롤러를 통해 해당 이벤트의 실행을 확인하는 방법을 자세하게 기술한다. OpenADR2.0 수요반응서버는 OpenADR2.0a 규격으로 그림 3과 같이 수요반응 클라이언트(VEN)가 수요반응서버(VTN)에 폴링(Polling) 방식으로 호출하고 수요반응서버(VTN) 측에서 이벤트를 발생시키는 방식으로 통신이 이루어지며, HTTP(HyperText Transfer Protocol) 방식에 XML(eXtensible Mark-up Language) 형식으로 이벤트 데이터를 전송하게 된다[2,8].

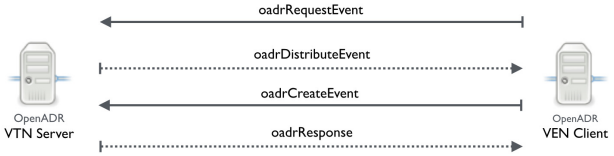


그림 3 OpenADR 이벤트절차
Fig. 3 OpenADR event flow

수요반응서버로는 인텔 제온 프로세서를 가진 시스템에 CentOS 6.5 오퍼레이팅 시스템과 NginX 웹서버를 탑재하였다. 내·외부 네트워크 통신을 사용할 수 있도록 하였으며, 이벤트 데이터와 VEN 정보 및 스케줄 정보 등을 저장하기 위하여 MySQL-Server 관계형 데이터베이스를 설치하였다. OpenADR2.0 오픈소스 프레임워크를 설치하기 위하여, 자바 1.7을 설치하고, GVM(Groovy Virtual Machine)을 통하여 Grails을 설치하여, 그림 4와 같이 구성 하였다. 또한, 수요반응서버에서 생성한 이벤트에 대한 반응을 확인하기 위하여 LED를 탑재한 마이크로컨트롤러를 피씨와 물리적인 케이블로 연결하여 이벤트 발령 시 LED를 점등 또는 소등 하여 수요반응을 실험하였다.

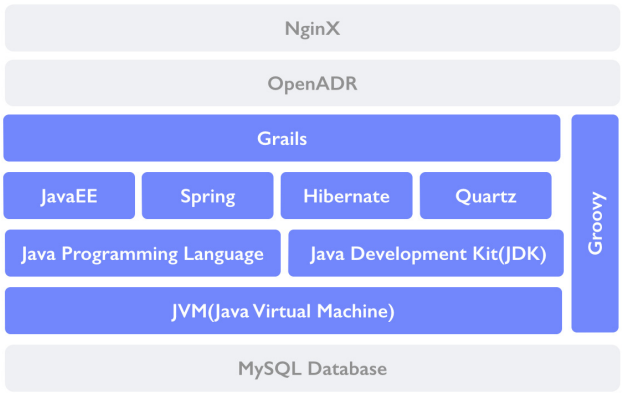


그림 4 수요반응 서버 프레임워크
Fig. 4 VTN Server Framework

OpenADR2.0 수요반응서버는 Grails 프로젝트 디렉토리 구조로 이루어져 grails-app 폴더와 web-app 폴더를 기준으로 사용 환경에 맞추어 변경 할 수 있으며, logs 폴더를 참조하여 프로젝트를 진행할 수 있다[8]. OpenADR 서비스 실행 전에 현재 시스템에 맞게 사용 플러그인, 데이터베이스 연결, 서비스 설정 등을 환경에 맞도록 수정한다. 어플리케이션 디렉토리인 grails-app/conf에서 Building.groovy, Config.groovy, DataSource.groovy 의 파일을 서비스 설정에 맞춰 표 1과 같이 설정을 변경해 준다. Building.groovy는 빌드에 필요한 플러그인을 추가적으로 인스톨할 수 있게끔 하며, Config.groovy는 Polling방식 이외에 XMPP 통신 시 필요한 XMPP 서버 정보를 설정한다. DataSource.groovy는 OpenADR2.0 수요반응서버에서 발생되는 각종 데이터를 저장할 수 있는 데이터베이스 설정을 할 수 있다. 본 실험에서는 MySQL 관계형 데이터베이스로 진행하였다. 설정을 마친 후, 프로젝트를 실행하였을 때, 그림 5와 같은 화면을 콘솔에서 볼 수 있으면 정상적으로 OpenADR2.0 수요반응서버가 실행 된 것이며, 웹브라우저에 “http://localhost:8080” 주소를 입력하면 OpenADR2.0 수요반응서버의 기본 웹페이지를 볼 수 있다.

표 1 OpenADR2.0 환경설정
Table 1 Environmental Settings

파일명	변경내용	비고

BuildConfig.groovy	plugins {
	compile 'mysql-connectorj:5.1.22.1'
	....
	}
		MySQL Plugins 추가

Config.groovy	xmppSvc { .. } // 해당 xmpp 접속계정 추가 rabbitmq { .... } // rabbitmq 접속계정 추가	XMPP통신 설정
DataSource.groovy	dataSource { .... } // MySQL접속 계정정보 추가	DB 접속정보 추가

다음은 수요반응클라이언트 설치를 위해 OpenADR 2.0 VEN (Python Version)을 다운받고, 설정 및 실행파일인 poll_runner.py을 열어 수요반응서버 접속정보 및 수요반응 클라이언트 정보를 기입한다. BASE_URI, VEN_ID, VTN_IDS, VTN_POLL_INTERVAL 등 설정 정보를 기입 후 수요반응클라이언트 서비스를 실행하면 그림 6과 같이 폴링(Polling) 방식으로 수요반응서버 측에 신호를 보내는 것을 볼 수 있다.

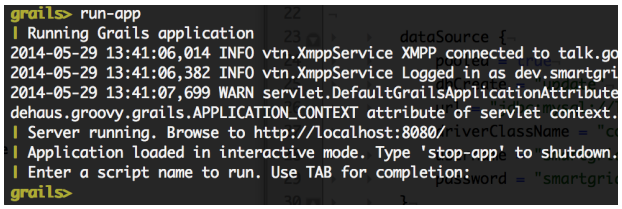


그림 5 수요반응 서버 웹페이지
Fig. 5 OpenADR2.0 Web page

수요반응서버에서 이벤트 발생 시, 이벤트 정보를 마이크로컨트롤러로 보내기 위해 시리얼 포트를 이용해야 한다. oadr2/poll.py, event.py 파일의 이벤트 발생/종료 부분에 시

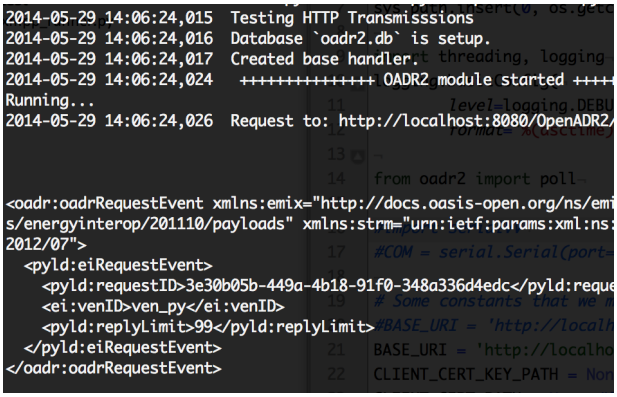


그림 6 OpenADR 수요반응 클라이언트 서비스 설정
Fig. 6 OpenADR VEN Service Settings

리얼 포트로 이벤트 신호를 전달하는 코드를 추가 한다. 본 연구에서는 시리얼 포트를 통해 이벤트 신호를 마이크로컨트롤러로 전달하도록 구성 하였다. 마이크로컨트롤러에 그림 7과 같이, LED의 긴 다리는 13번 PIN에 접속시키고, 짧은 다리는 GND에 접속시킨다. 또한 이벤트 스케줄에 따라 반응할 수 있도록 그림 8의 코드를 스케치하여, 마이크로컨트롤러에 업로드 시킨 후 케이블을 PC에 연결 시켰다. 이제 OpenADR2.0 수요반응서버에서 이벤트를 발생시키게 되면, 폴링되고 있는 OpenADR2.0 클라이언트 쪽으로 이벤트가 등록되고, 해당 이벤트는 시리얼 포트를 통해 마이크로컨트롤러로 전달되며, LED는 이벤트가 정해놓은 시간에 ON/OFF 되도록 구성 된다[11].

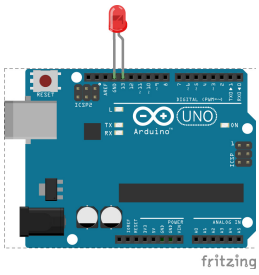


그림 7 마이크로컨트롤러 구성
Fig. 7 Microcontroller

2.3 이벤트 테스트 및 검증

OpenADR2.0 수요반응서버의 이벤트 메뉴에서 이벤트 시작시간과 종료시간(Start Date, End Date)을 설정하고, 이벤트를 발생시키면, 폴링 중인 OpenADR2.0 수요반응 클라이언트에서 이벤트를 감지하면, 콘솔상의 상태(Status)는 “near”로 표시되어 이벤트를 대기 상태로 변한다. 시작시간이 되면, 콘솔상의 상태가 “active”로 변경되고, 종료시간이 되면 콘솔에 이벤트가 삭제(Removing Cancelled event) 메시지를 띄우게 된다. 해당 이벤트 신호에 따라 마이크로컨트롤러도 동시에 제어되어 LED 점등 또는 소등하게 된다. 그림 9는 수요반응서버 및 클라이언트 간의 이벤트 흐름과 이에 따른 마이크로컨트롤러의 동작을 보여준다.

```
const int ledPin =  13;
int ledState = LOW;

void setup() {

  Serial.begin(9600);
  pinMode(ledPin, OUTPUT);
}

void loop()
{

  if (Serial.available() > 0) {

    int inByte = Serial.read();
    if(inByte >= '0' && inByte <= '5')
    {
      ledState = HIGH;
    } else {
      ledState = LOW;
    }
    digitalWrite(ledPin, ledState);
  }
}
```

그림 8 마이크로컨트롤러 제어 프로그램
Fig. 8 Microcontroller control program

그림 9 수요반응서버와 클라이언트 간의 이벤트 흐름
Fig. 9 Event flow between VTN and VEN

실행 서버의 상태는 각 콘솔상에서 OpenADR2.0 수요반응서버 및 OpenADR2.0 클라이언트의 데이터 흐름 메시지를 실시간으로 확인할 수 있으며, 등록된 이벤트 및 클라이언트, 이벤트주기, 이벤트간격 등은 그림 10과 같이 데이터베이스 접속을 통해 확인이 가능하다.

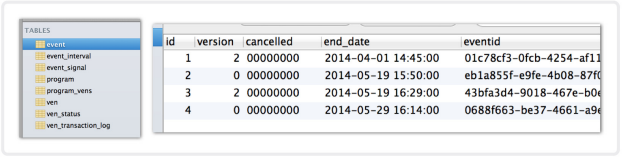


그림 10 이벤트 데이터베이스
Fig. 10 Event database

마이크로컨트롤러를 보면, 해당 이벤트가 진행되는 동안 마이크로컨트롤러의 LED는 그림 10과 같이 변경되었다. OpenADR2.0 수요반응서버에 등록된 이벤트는 16시 20분에 시작하여, 16시 21분에 종료되도록 저장되었고, 16시 14분에 이벤트가 입력되어 이벤트 대기상태로 되었고, 16시 20분에 LED는 OFF상태가 되었다. 잠시 후 16시 21분에 다시 LED가 ON상태로 변화 하면서 이벤트가 정상적으로 종료되었다. 현재는 단순히 마이크로컨트롤러의 LED제어만으로 OpenADR2.0 수요반응서버의 명령을 수행하였지만, 마이크로컨트롤러에 다양한 통신 모듈 (Zigbee, Bluetooth, Wifi, IR Remote 등) 및 센서를 결합하면 수요반응 클라이언트의 프로그래밍에 의해 클라이언트에서 관리할 수 있는 다양한 디바이스들의 제어가 가능하다. 따라서 결국 수요반응서버에서 모든 관리와 통제 하에 각 디바이스 단위의 수요관리가 가능하게 된다.

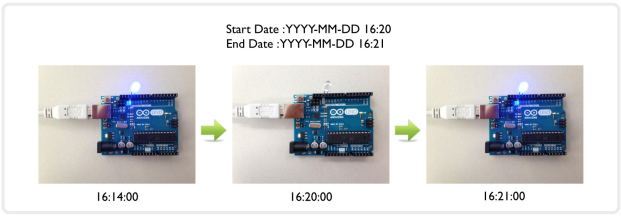


그림 11 마이크로컨트롤러 이벤트
Fig. 11 Microcontroller event

3. 결 론

본 논문에서는 오픈소스를 이용하여, 보다 쉽게 수요관리 자동화 서비스를 구축하기 위한 기본적 설계 방안을 제시하였으며, 수요관리 자동화 프로토타입을 구축하여 시험하였다. 또한, OpenADR2.0 수요반응서버와 OpenADR2.0 수요반응 클라이언트 간의 OpenADR2.0규격 이벤트 통신과 이벤트 흐름을 상세하게 기술하였다. 제안된 프레임워크는 향후 자동화된 수요반응 시스템을 구현하는데 기초로 사용될 것이다. 마지막으로 수요반응 클라이언트와 연계된 마이크로컨트롤러는 시리얼 통신을 기반으로 하여 이벤트에 따라서 LED를 컨트롤하였다. 이런 마이크로컨트롤러의 확장성을 고려해 볼 때, 수요반응 클라이언트의 프로그래밍에 따라 OpenADR2.0 수요반응서버와 OpenADR2.0 수요반응 클라이언트의 역할은 더욱 넓어질 것으로 판단된다.

현재의 프로토타입은 다양한 프로토콜을 지원하지 않아서 각종 디바이스를 제어하기에는 부적합하다. 향후에 프로토콜을 변환시키는 미들웨어를 마이크로컨트롤러에 탑재하여 수요반응서버에서 수요관리자원을 직접 통제하는 부분을 추가할 예정이다. 또한 일반사용자가 어려움이 없이 사용할 수 있도록 그래픽 기반의 사용자 인터페이스가 추가될 예정이다.

감사의 글

본 연구는 2013년도 산업통산자원부의 재원으로 한국에너지기술평가원(KETEP)의 지원을 받아 수행한 연구 과제입니다. (No. 20132010101800)

References

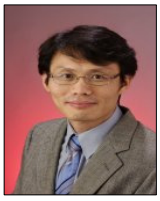
[1] OpenADR Alliance Website, http://www.openadr.org
[2] EnerNOC Website, http://www.enernoc.com

[3] SmartGrid Website, http://www.smargrid.org, Korean SmartGrid Association.
[4] Jae Jung Park, DR(Demand Reponse) Technology for Smart Grid, KERI, 2013.
[5] Jimyung Kang, Implementation of OpenADR 2.0a Profile for Demand Response in Smart Grid, KERI, 2013.
[6] Demand Reductions from the Application of Advanced Metering Infrastructure, Pricing Programs, and Customer-Based System-Initial Results, SmartGrid.gov, 2014.
[7] Demand response architectures and load management algorithms for energy-efficient power grid: a survey, Y. Law, T. Law, T. Alpcan, V. Lee, A. Lo, M. Palaniswami, Seventh International Conference on Knowledge, Information and Creativity Support Systems, 2012.
[8] OpenADR Open Source Toolkit: Developing Open Source Software for the Smart Grid, Charles McParland, IEEE Power and Energy Society General Meeting, 2011.
[9] Implementation Proposal for National Action Plan on Demand Response, FERC and DOE, 2011.
[10 PJM Website, http://pjm.com, Training mateial: PJM 101, PJM, 2013.
[11] Arduino Website, http://arduino.cc

저 자 소 개

윤 재 원 (尹 哉 元)

2008년 건국대에서 석사학위를 받은 후, 웹프로그래머로 근무하였으며, 2013년 부터는 차세대융합기술원 스마트그리드 연구센터에서 ICT를 활용한 비즈니스 플랫폼 개발 과제에 참여 중이며, 연구 관심 분야는 IoT를 활용한 에너지 효율화.



이 인 규 (李 仁 奎)

2007년 펜실베니아 주립대에서 박사학위를 받은 후, 미국 알라바마주 트로이 주립대에서 교수로 재직하였고, 2013년 부터는 차세대융합기술원 스마트그리드 연구센터에서 비즈니스 플랫폼 및 빅데이터 마이닝에 관하여 연구를 수행 중.

최 중 인 (崔 重 仁)

1987년 메사추체스 대학교에서 박사학위를 받은 후, 가천대학교에서 교수로 재직한 후, 2013년부터 차세대융합기술원 스마트그리드 연구센터에 센터장으로 근무중이며, 주요 연구 관심 분야는 스마트그리드 및 트랜잭티브 에너지.