

<https://doi.org/10.7236/JIIBC.2017.17.1.123>

JIIBC 2017-1-16

# 오픈 소스 하드웨어 기반의 스마트 센서 네트워크 시스템 구현

## Implementation of Smart Sensor Network System Based on Open Source Hardware

권오석\*, 김기환\*\*

Oh-Seok Kwon\*, Kee-Hwan Kim\*\*

**요약** 본 논문에서 오픈 소스 하드웨어인 아두이노를 기반으로 하는 스마트 센서 네트워크 시스템 모델을 제안하고 구현하였다. 제안된 스마트 센서 네트워크 시스템은 센서 및 센서의 값을 처리할 수 있는 오픈 소스 하드웨어 기반인 아두이노 등으로 구성된다. 또한 센서로부터 측정된 센서값을 활용할 수 있는 제어 장치부에 전송할 통신 모듈도 구성하였다. 제어 장치부에서는 온도, 습도, 광량과 같은 센서 데이터를 메인 프로그램으로 전송하고 메인 프로그램은 데이터를 DB에 저장하거나 특정한 제어 신호의 값을 제어장치 혹은 액츄에이터에 전송한다. 사용자는 또한 웹을 통해 스마트 센서 네트워크에서 측정된 값을 사용하는 시스템의 정보를 확인하거나 각종 액츄에이터를 원격 제어할 수 있으며 제안 시스템의 상황 인지 및 자율 제어 기능을 통해 스마트한 관리가 가능하다.

**Abstract** In this paper, we have proposed and implemented a smart sensor network system based on the Arduino open source hardware. The proposed smart sensor network system is composed of kinds of sensors and open-source hardware based, Arduino, etc. that can handle the measured sensor values. Also the communication modules that can be used to transmit the measured sensor values from the sensor control unit are configured. In the control unit the sensor data such as temperature, humidity, light intensity can be transmit to the main program and the main program will save the data in the DB or transmitting the value of the particular control signal to the control device or the actuator. The user can also check the information in the system using the measured values from the smart sensor networks through the web, or to remotely control a variety of actuators. And it is possible to manage a smart autonomous control over whether and how the proposed system.

**Key Words** : Open Source Hardware, Android, Smart Sensor Network, Arduino, Blue Tooth, ZigBee

### 1. 서론

최근 컴퓨터 및 정보통신 기술의 발전에 따라 사물인터넷에 많은 관심이 대두되고 있다. 이러한 기술의 흐름과 함께 스마트 센서 네트워크에 대한 관심과 수요가 많

이 늘었다. 사물인터넷 IoT(Internet of Things)이란 우리 주변의 사물들에 IP를 부여하고 이들을 네트워크로 연결하여 상호 정보를 주고받으며 지능화함으로써 다양한 분야에서 사용자의 요구를 충족시키는 기술이다.<sup>[1]</sup> 여기서 사물들의 상태 혹은 정보를 알기 위해서 다양한 종

\*정회원, 한국폴리텍 다솜고등학교 Smart전기과

\*\*중신회원, 세명대학교 전자공학과(교신저자)

접수일자 2016년 9월 30일, 수정완료 2017년 1월 7일

계재확정일자 2017년 2월 3일

Received: 30 September, 2016 / Revised: 7 January, 2017 /

Accepted: 3 February, 2017

\*\*Corresponding Author: khkim@semyung.ac.kr

Department of Electronic Engineering, Semyung University, Korea

류의 센서들이 구성되어야 한다. 이러한 센서들은 사용자의 사용환경 혹은 요구에 따라 서로 다르게 구성되어져야하며, 다양한 정보를 함께 모니터링하고, 분석하고, 제어할 수 있어야 한다.

이러한 센서 네트워크를 구성하기 위하여 기존에는 센서들의 집합인 센서 보드를 사용하였으며 이 센서 보드를 네트워크화 시키기 위한 센서 노드 그리고 웹서버를 따로 구성하였다. 이렇게 구성된 시스템은 비용도 높을 뿐 아니라 사용환경에 특장화 되어 있는 경우가 대부분이다. 또한 새로운 기능의 센서 인터페이스를 추가하는 경우에는 센서 네트워크 보드를 새로 설계해야하는 경우가 대부분이다.<sup>[2]</sup>

본 논문에서는 이러한 번거러움을 줄이고 사용자가 원하는 센서와 주변 기기를 장착하고 이미 공개된 오픈소스 하드웨어 및 소프트웨어를 이용하여 스스로 설계할 수 있는 시스템에 대하여 고찰하기로 한다. 이를 통해 사용자에게 최적화된 시스템을 제공하여 다양한 분야에 적용가능하게 하고자 함에 주안점을 두었다.

## II. 시스템구성

### 1. 센서 네트워크 하드웨어

본 논문에서는 센서 네트워크의 주된 하드웨어로 오픈소스 하드웨어 기반의 아두이노를 적용하였으며 더불어 라즈베리파이를 서버로 사용하여 웹기반의 모니터링이 가능하도록 시스템을 구성하였다.<sup>[3][4]</sup>

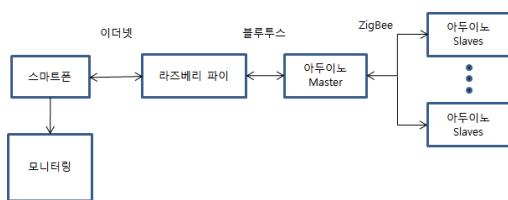


그림 1. 시스템구성도  
Fig. 1. System Diagram

전반적인 센서 네트워크의 기본 토폴로지의 구성은 그림 1과 같다. 그림 1에서 보이는 것과 같이 센서노드와 센서의 값을 읽고 처리할 수 있는 부분으로 아두이노를 사용하였다. 센서노드의 역할을 하는 것을 아두이노 Master라 칭하고 센서값을 읽고 처리하는 부분의 아두이노를 아두이노 Slave라 칭하였다.

그림 2에는 아두이노 Slave를 나타내었다. 그림 2에서 보는바와 같이 아두이노 Slave는 코디네이터와 라우터 그리고 여러 물리량을 측정할 수 있는 각종 센서들과 이를 바탕으로 제어를 하기 위한 액츄에이터들로 구성되어 있다.

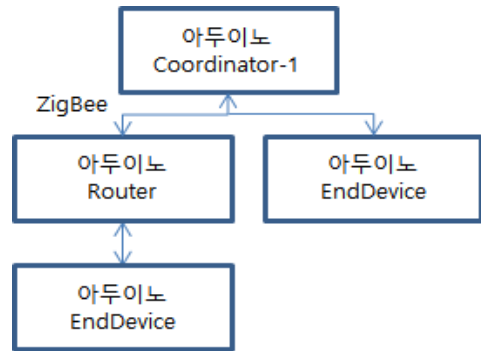


그림 2. 아두이노 Slave 구성도  
Fig. 2. Arduino Slave Diagram

여기서 아두이노 라우터와 End Device간의 통신은 지그비를 사용하였다. 그 이유는 지그비 통신을 이용하면 서로 다른 아두이노 Slave를 구성하여 여러 개의 개별적인 네트워크를 구성할 수 있다.<sup>[2]</sup>

본 연구에서 사용한 센서들은 기존의 산업체나 홈오토메이션, 공장자동화, IoT 등에 많이 사용하는 온도센서, 광량/조도 센서, 수위센서, 홀센서, 풍속센서 그리고 화재를 감지할 수 있는 화재감지 센서 등을 사용하였으며, 액츄에이터로는 기본적인 직류모터와 스위치 그리고 릴레이 등을 사용하였다.

이들 아두이노 Master와 Slave 사이에는 통신모듈마다 약간의 차이는 있지만 대략적으로 150m정도까지의 거리를 지원하고 Ad-hoc 기능을 갖는 지그비 통신을 사용하였다. Ad-hoc기능은 End Device가 Coordinator에게 데이터 전송 시 전송거리를 벗어났을 경우 통신거리내의 다른 모듈을 통해 데이터를 Coordinator에게 전달할 수 있다.

이는 무선통신의 네트워크 토폴로지가 동적으로 변하여 통신거리를 비약적으로 늘릴 수 있는 장점이 있다. 또한 라즈베리파이를 사용하여 아두이노 Master와 연결하였다. 이 라즈베리파이를 통해 인터넷으로 모니터링이 가능하게 하였다. 이때 라즈베리파이와 아두이노 Master의 연결은 빠른 전송속도 및 근거리 통신을 지원하여 스마트폰과 직접 페어링을 가능하게 하는 Bluetooth 연결

방식을 사용하였다.<sup>[5]</sup>

아두이노 Master 모듈은 Bluetooth의 통신거리 및 구성할 수 있는 네트워크 노드수에 한계가 있으므로 라즈베리 파이와 근거리에서 위치시켜 여러 가지 디스플레이 기능 등을 제공하게 하였다. 그림 3에 전반적인 라즈베리 파이에서부터 아두이노 Slave까지의 통신 방법 및 신호의 흐름을 나타내었다.

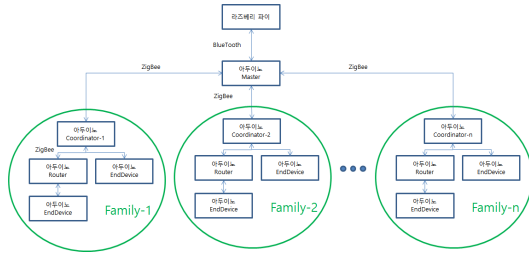


그림 3. 세부적인 시스템 구성도  
 Fig. 3. Detailed System Diagram

## 2. 서버구성

기존의 인터넷을 이용한 모니터링 시스템은 PC를 이용한 방식이 주를 이루었으나 이 방식은 서버구축에 많은 불편함을 가질 수밖에 없었다. 또한 물리적 공간 제약 및 가격, 서버구축의 어려움 등은 모니터링시스템의 구축에서 많은 제약조건을 가져온다. 하지만 최근에 나온 라즈베리 피어를 이용하여 서버를 구성할 경우 물리적 공간에 대한 제약에서 자유롭고 저가의 서버 구축가능, 서버 구축 개발기간 단축의 간편함 등이 있어 많은 이점을 가질 수 있다. 서버 구성은 그림 4에서와 같이 구성하였다. 그림 4에서 보논바와 같이 라즈베리파이의 인터넷을 통해 아두이노 Slave의 상태를 스마트폰으로 전송할 수 있으며 또한 스마트폰에서의 제어 신호를 아두이노 Slave의 액츄에이터로 보내 제어가 가능하도록 하였다.

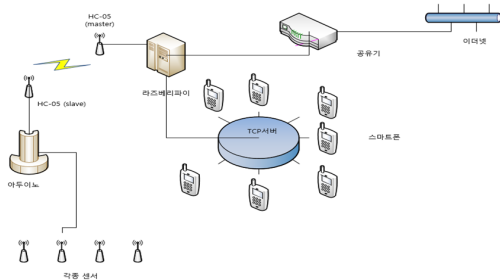


그림 4. 서버 시스템구성도  
 Fig. 4. Server System Diagram

## 3. 안드로이드 앱구성

상기에서 언급한 바와 같은 스마트폰을 이용한 서버 접속과정은 그림 5와 같이 이루어지며 안드로이드 앱에서는 필요에 따라 여러 액츄에이터 구동 및 데이터를 모니터링 할 수 있다.<sup>[6][7]</sup>

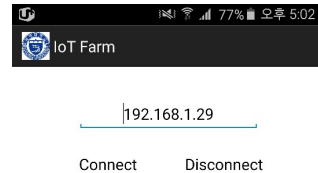


그림 5. 안드로이드 앱구성  
 Fig. 5. Android App

## III. 시스템 동작 알고리즘

각 센서노드들에서 취합되어진 센서정보 값들은 ZigBee 혹은 Bluetooth를 통하여 서버인 라즈베리 피어로 정보가 전송되어지고 라즈베리파이는 이렇게 수집된 데이터를 인터넷을 통하여 사용자가 스마트폰의 앱에서 모니터링이 가능하게 한다. 또한 사용자는 이의 역순으로 액츄에이터의 제어신호를 센서노드를 통해 아두이노 Slave에 구성되어진 액츄에이터를 구동시킬 수 있다.

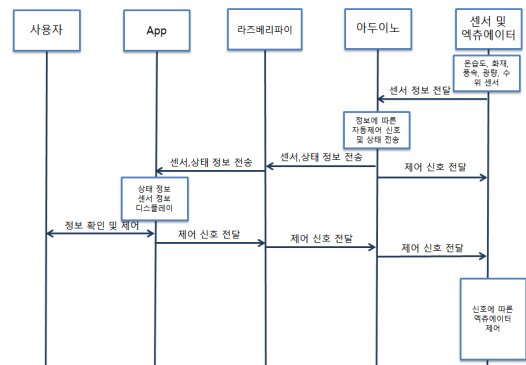


그림 6. 데이터 전송 흐름도  
 Fig. 6. Data Transport Flow

그림 6에서는 본 논문에서 구성한 정보전송의 방향과 종류에 대한 흐름을 나타내었으며 그림 7은 서버와 클라이언트 즉, 스마트폰 앱간의 인터넷 접속 과정을 나타낸다.

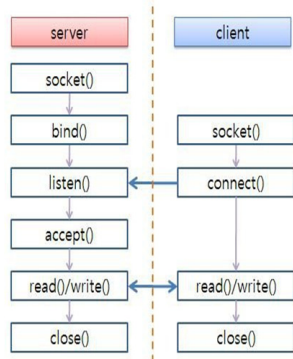


그림 7. 서버 접속 흐름도  
Fig. 7. Access Server Flow

클라이언트측에서는 열린된 포트를 통하여 접속을 요청하고 서버가 이를 수락하면 데이터를 송수신한다.

#### IV. 실험 및 결과

본 논문에서 사용한 시스템의 센서노드는 Arduino mega2560 모델이다. 아두이노는 간단한 전자연결 지식만 갖고 있으면 공개된 수많은 센서 모듈, 통신모듈, 액추에이터 등을 이용하여 전반적인 하드웨어를 구성할 수 있다. 또한 약간의 C언어 지식만으로 이미 공개된 모듈을 구동시킬 수 있는 프로그램을 작성할 수 있다.

그림 8은 오픈소스로서 약간의 변형을 통해 만든 안드로이드 프로그램 소스이다. 그림 8에서 볼 수 있듯이 간단한 프로그램이며 약간의 지식으로도 프로그램 변형이 가능하다.

```

public class Server {
    public static void main(String []args) {
        try {
            int portNumber=5001;
            int msg;

            System.out.println("서버 시작중...");
            ServerSocket s=ServerSocket.new ServerSocket(portNumber);
            System.out.println("서버 시작중... (" +portNumber+"");

            while(true){
                Socket sock=s.getServerSocket().accept();
                InetAddress clientHost=sock.getLocalAddress();
                int clientPort=sock.getPort();
                System.out.println("클라이언트 호스트: "+clientHost+" port: "+clientPort);

                ObjectInputStream inStream=new ObjectInputStream(sock.getInputStream());
                Object obj=inStream.readObject();
                Scanner input=new Scanner(System.in);
                msg=input.nextInt();
                System.out.println("수신: "+msg);

                ObjectOutputStream outStream=new ObjectOutputStream(sock.getOutputStream());
                outStream.writeObject("from Server");
                outStream.flush();
                sock.close();
            }
        } catch(Exception e){
            e.printStackTrace();
        }
    }
}
    
```

그림 8. 안드로이드 프로그램 코드  
Fig. 8. Android Program

본 논문에서는 여러 액추에이터 중에서 광량 및 온도 제어를 위한 창문의 개폐 모터구동, 게이트 개폐를 위한 모터 구동, 온도나 환기조절을 위한 팬을 구동하는 모터를 대상으로 정하였다. 이들 센서의 값과 액추에이터를 제어하는 제어량에 대한 스마트폰의 GUI 구성은 다음 그림 9와 같다.



그림 9. 안드로이드 GUI 프로그램  
Fig. 9. Android GUI Program

그림 10은 센서 노드 아두이노 보드의 프로그램 흐름도이다. 여러 가지 센서에서 측정된 데이터를 바탕으로 여러 액추에이터를 제어한다. 지그비 통신 네트워크에서 코디네이터로 동작하는 센서노드는 이 데이터를 블루투스 통신을 통하여 라즈베리파이로 전송하고 라우터 또는 엔드바이스로 동작하는 센서노드는 코디네이터로 데이터를 전송한다. 코디네이터는 라우터 혹은 엔드바이스에서 전송되어진 데이터를 라즈베리파이로 전송한다.

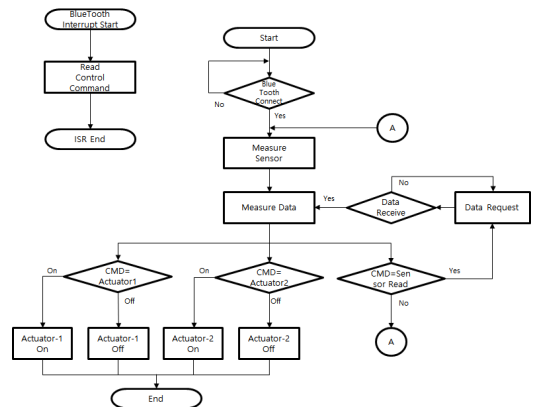


그림 10. 센서 노드 프로그램 흐름도  
Fig. 10. Sensor Node Program Flow Chart

본 논문에서 구성한 아두이노 보드에는 여러 환경요 소측정을 위한 센서 모듈, 액츄에이터 구동을 위한 모터 드라이버, 상시전원을 구동하기 위한 릴레이 모듈 등을 장착하였다. 그림 11은 아두이노 보드를 사용하여 구성한 센서노드이다.

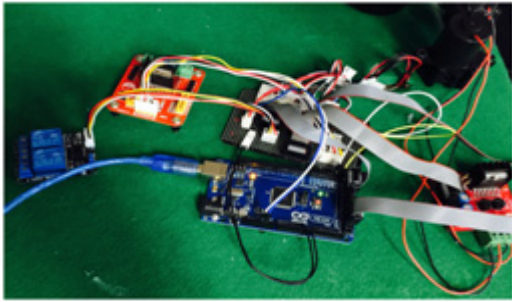


그림 11. 아두이노 센서 노드  
Fig. 11. Arduino Sensor Node

또한 그림 12에서와 같이 초음파 센서와 불꽃감지 센서 모듈을 연결 할 수 있다. 초음파 센서는 HC-SR04 모델을 사용하였다. HC-SR04 모델은 동작전원 DC5V, 12mA에서 동작하며 측정거리는 0.02m~4m정도이다. 분해능은 0.003m이다. 불꽃감지센서는 약 760nm~1100nm 파장의 불꽃을 감지할 수 있다. 실험결과 라이터 불꽃을 약 1m 정도에서 감지가 가능하였다.

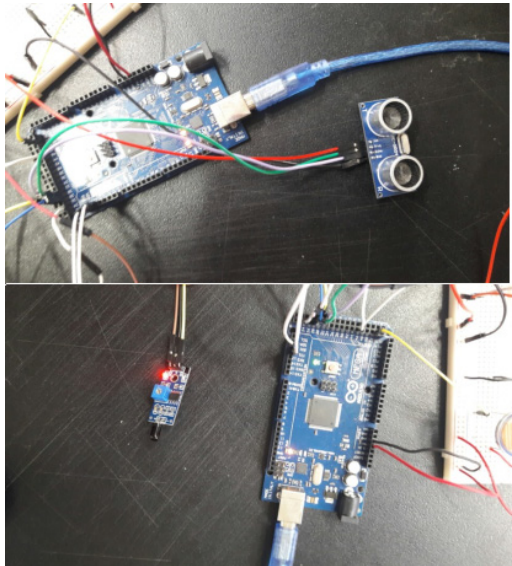


그림 12. 초음파 센서 및 불꽃감지 센서  
Fig. 12. Ultra Sonic Sensor and Fire Sensor

라즈베리 파이와 아두이노 Master 사이의 통신을 하기 위한 블루투스 모듈과 각종 액츄에이터를 구동시킬 수 있게 하는 릴레이 모듈은 그림 13에 나타내었다. 블루투스 모듈은 HC-05 모듈을 사용하였다. 이 제품은 UART 통신을 통하여 데이터 송수신이 가능하며 AT Command를 통하여 Master/Slave 설정, 비밀번호 변경, 통신속도 설정 등의 설정이 가능하다. 또한 일반적인 블루투스 모듈과 비슷하듯이 통신거리가 10m 정도이다.

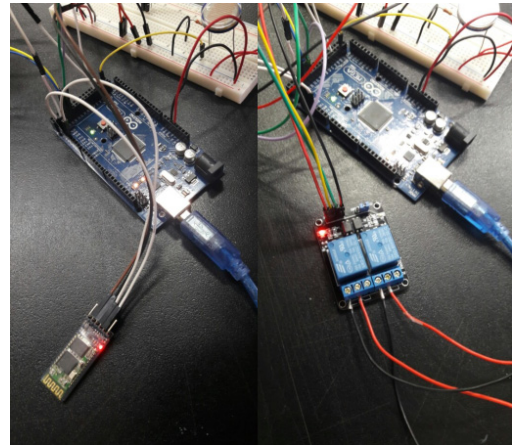


그림 13. 블루투스 모듈 및 릴레이모듈  
Fig. 13. BlueTooth and Relay Module

## V. 결 론

본 논문에서 오픈 소스 하드웨어인 아두이노를 기반으로 하는 스마트 센서 네트워크 시스템 모델을 제안하고 구현하였다. 제안된 스마트 센서 네트워크 시스템은 센서 및 센서의 값을 처리할 수 있는 오픈 소스 하드웨어 기반인 아두이노 등으로 구성된다. 주로 센서 값을 측정하고 액츄에이터의 제어 신호를 보내는 아두이노 Slave 부분은 지그비 통신을 이용하여 매쉬형태의 네트워크를 구성하였고, 상위 부분인 아두이노 Master부분은 BlueTooth 통신을 이용하여 tree 형태의 네트워크로 구성하였다. 이렇게 매쉬 및 트리 구조의 네트워크를 복합적으로 사용함으로써 시스템의 확장이 수월해짐을 알 수가 있었다.

또한, 아두이노에서는 여러 가지 센서모듈이나 디스플레이 모듈, 액츄에이터 등을 제공한다. 홈오토메이션, 공장자동화 및 기타 IoT 제품 등을 개발하기에 부족함이



없을 정도의 모듈을 갖추고 있으며 이를 프로그래밍하고 구동하기에 부족함이 없는 예제코드 및 라이브러리를 제공한다. 이는 하드웨어에 대한 이해나 지식이 부족한 사람이라도 C언어와 약간의 자바 코드를 배움으로써 홈오토메이션 등 각종 IoT 제품으로의 접근이 수월하여 전자 제어의 DIY가 가능하도록 가능성을 높여주었다.

## References

- [1] Jeong-Won Kim, "A Smart Home Prototype Implementation Using Raspberry Pi", Journal of The Korea Institute of Electronic Communication Sciences Vol. 10 No. 10 2015. 10, pp 1139-1144
- [2] Kee Hwan Kim, "Development of Complex USN Sensor for Zero Energy House with Blind System", The Journal of the Institute of Internet, Broadcasting and Communication 2013. 8, Vol. 14 No. 4 pp. 221-227  
DOI: <http://dx.doi.org/10.7236/JIIBC.2013.13.4.221>
- [3] C. Bell, "Beginning sensor networks with Arduino and Raspberry Pi", New York: Apress Media, 2013
- [4] A. Dennis, "Raspberry Pi home automation with Arduino", Birmingham: Packt Publishing, 2013
- [5] Sung-Ja Choi, Byeong-Gwon Kang, "Home Monitoring Server System Using Smart Interface over Wireless Networks", Journal of Digital Convergence Vol. 10 No. 8 2012. 9, pp. 225-231
- [6] Ji-Dong Ju, Jin-Seoung Kim, Bong-Gu Kang, Jeachang Shim "Implementation of Mobile Multi-sensor System for Measuring an Environment", Journal of Korea Multimedia Society Vol. 17, No. 8, 2014. 8. pp. 1020-1024
- [7] In-Gu Lee, Gi-Pyeong Kim, Myeon-Gyun Cho, "Arduino based Smart Home System using Smart-phone", Domestic Conference of The Institute of Electronics Engineering of Korea, 2015. 6, pp. 1667-1668

## 저자 소개

### 권 오 석(정회원)



- 2002년 : 세명대학교 교육대학원 전기전자통신 교육학 석사
- 2008년 : 세명대학교 전기전자공학과 박사 수료
- 한국폴리텍 다듬고등학교 Smart 전기과 교사

### 김 기 환(종신회원)



- 2015년 제15권 제2호 참조

※ 이 논문은 2015학년도 세명대학교 교내학술연구비 지원에 의해 수행된 연구임