

Design and implementation of IoT platform for collecting and managing the SmartFactory environment information

SungJin Kim*, SangYong Ra**, HwanSeog Kim***, JaeHong Choi*, JunDong Lee*

Abstract

Smart Factory is a part of and a key point of the 4th industrial revolution. It performs optimization from the whole viewpoint, using comprehensive data of the post-process data by utilizing various sensors, controllers, and mobile devices beyond the existing factory automation level.

In this paper, we design and implement an IoT platform that can detect the safety factors of the workers, the environmental factors of the factory, and real time monitoring at the control center, among the fields to implement smart factory. To accomplish this, we construct a monitoring device that provides sensor information control, server transmission of sensor information, and visualization of collected information.

By using this system, it is possible to maintain the temperature and humidity for the optimum working environment in the factory. and also, By using the beacon, it is possible to measure the working time of the worker and trace the position,

▶ Keyword: Smart Factory, Internet of Things, Platform, Raspberry pi, 4th industrial revolution

I. Introduction

4차 산업혁명이란 인공지능과 데이터 네트워크 기술인 ICBM(IoT, Cloud, Big Data, Mobile)이 결합되어 모든 산업에 융합되는 것을 의미하는 것으로, 혁명이란 용어에서 알 수 있듯 사회 전반에 대 변혁을 초래할 것이다.

특히, 스마트팩토리는 4차 산업혁명의 한 분야이자 근본으로, 기존의 공장 자동화 수준을 넘어 다양한 센서와 제어기, 모바일 기기를 활용하여 전후 공정의 데이터를 포괄적으로 활용한 총체적 관점에서의 최적화를 수행한다. 이는 기존의 공장자동화가 단위 공정별 자동화를 활용한 생산 최적화를 수행한 것과 비교될 수 있다.

스마트팩토리의 장점은 데이터기반의 의사결정, 생산 효율성 증대, 그리고 시장 대응성 증대에 있다. 즉, 경험이나 직관이 아

닌 데이터를 통해서 객관적인 생산관리를 할 수 있으며, 자동화 방식을 통한 인건비 절감과 재료와 에너지의 효율적 사용, 그리고 기존 공장의 문제점인 소품종 대량 생산에서 다품종 대량 생산이 가능한 생산 공정의 신속 전환으로 짧아진 제품 수명주기에 즉각적으로 대응할 수 있다.

본 논문에서는 스마트팩토리를 구현하기 위한 분야 중 작업자 안전과 공장의 환경요소 파악, 관제센터에서의 실시간 상황 모니터링이 가능한 사물인터넷 플랫폼의 설계 및 구현을 목표로 한다.

플랫폼에서는 다양한 외부 센서와 도구를 활용하여 스마트팩토리 환경 정보를 수집하고, 이를 가공하여 사용자에게 정보를 제공한다. 이를 위하여

• First Author: SungJin Kim, Corresponding Author: JunDong Lee

*SungJin Kim (tonyksj@naver.com), Dept. of Multimedia Engineering, GangNeung-Wonju National Univ.

**SangYong Ra (ra.sangyong@ex2i.com), Ex2i Corporation CEO

***HwanSeok Kim (hskim8805@gwnu.ac.kr), Dept. of Information Communication Engineering, GangNeung-Wonju National Univ.

*JaeHong Choi (smart_phone@daum.net), Dept. of Multimedia Engineering, GangNeung-Wonju National Univ.

*JunDong Lee (jlee@gwnu.ac.kr), Dept. of Multimedia Engineering, GangNeung-Wonju National Univ.

• Received: 2019. 04. 30, Revised: 2019. 05. 29, Accepted: 2019. 06. 06.

• This work is the extended research of 2018 KSCI Winter Conference(p77-78)

• This work (No.C0505260) was supported by Business for Cooperative R&D between Industry, Academy, and Research Institute funded Korea Small and Medium Business Administration in 2017

- 환경정보의 수집과 제어를 위하여 다양한 센서를 제어하고 센서의 정보를 수집, 관리
- 수집된 센서 정보를 서버에 전송
- 수집된 정보를 시각화하여 제공하는 모니터링 장치를 구성한다.

II. Preliminaries

2. Related works

2.1 IoT Framework

인터넷은 Fig 1과 같이 다섯 단계로 발전해왔다. 처음에는 두 개의 컴퓨터를 서로 연결함으로 시작하였지만, 곧 WWW를 통하여 많은 수의 컴퓨터를 연결하는 인터넷으로 발전하였고, 그 후 모바일 기기를 인터넷에 연결하는 모바일 인터넷으로 발전하였다. 이후 소셜 네트워크를 통하여 사람들의 정보가 인터넷에 들어왔으며, 이제 거의 모든 객체가 인터넷에 연결되는 사물인터넷의 시대에 접어들었다[4].

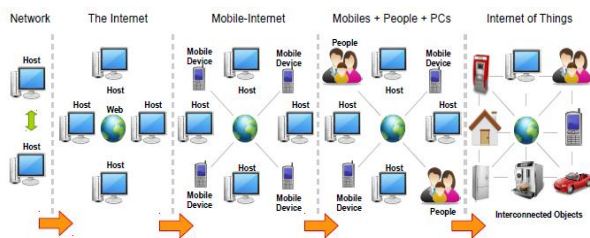


Fig. 1. Evolution of Internet

사물인터넷의 시대에 일반적인 IoT 프레임워크에서 고려하는 시스템 구조는 Fig 2와 같이 Object Sensing Layer, Data Exchange Layer, Information Integration Layer 그리고 Application Service Layer로 구성된다[2]. Object Sensing Layer는 가장 하위 계층으로 물리적 개체에 대한 감지와 데이터를 수집하는 계층이고, Data Exchange Layer는 데이터의 전송을 담당한다. Information Integration Layer는 네트워크로부터 얻어진 데이터의 재결합 및 합성을 통하여 불확실한 정보를 사용 가능한 정보로 통합한다. 마지막으로 Application Service Layer는 다양한 사용자들을 위해 콘텐츠 서비스를 제공한다.

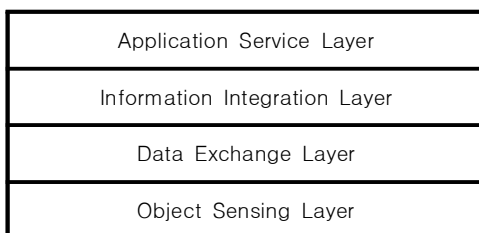


Fig. 2. Four-layer architecture of IoT

2.2 Context Awareness Technique

상황 인식이란 용어는 1994년에 Schilit 그리고 Theimer에 의해 처음으로 사용되었으며, 1990년 대 초기부터 컴퓨터 분야에서 보편적인 연구 영역으로 인정받았다[3].

상황 인식은 사용자와 주위 환경을 인식하고 이해하여 변화하는 상황을 인지하고, 이를 토대로 개인 요구에 따라 환경을 제어하는 데 활용된다. 즉, IoT로부터 발생하는 엄청난 센싱 데이터로부터 의미있는 정보를 추출하는 중요한 기술이다.

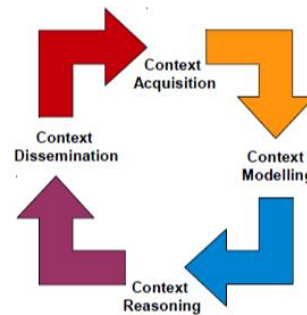


Fig. 3. Context life cycle

이러한 상황 인식을 이용하여 사용자에게 적합한 서비스를 제공하는 상황 인식 시스템(context awareness system)은 Fig 3과 같은 상황의 라이프 싸이클을 지원하여야 한다.

상황 수집(context acquisition)은 상황을 제공하는 모든 데이터 소스로부터 획득하는데, 일반적인 하드웨어 센서인 물리적 센서 외에도 가상센서나 논리적센서에서 발생하는 상황도 포함된다. 상황을 수집하는 방법에는 pull과 push의 방법이 있는데, pull 방법은 서버쪽에서 센서에 요청하여 센싱데이터를 수집하는 것이고 push방식은 센서가 데이터를 서버쪽으로 보내는 방식이다.

상황 모델링(context modeling)은 상황 표현(context representation)과 같은 의미로 사용되는데, 이를 통하여 상황의 구체적인 하위 집합을 식별할 수 있다. 상황 모델 기술로는 대표적으로 6가지가 있는데, 키 값(key-value) 방법, 마크업 기법(markup schemes), 그래픽(graphical) 방법, 객체 기반(object based) 기법, 논리 기반(logic based)기법, 온톨로지 기반(ontology based) 기법등이 있다[5].

상황 추론(context reasoning)은 사용 가능한 상황을 기반으로 더 잘 이해할 수 있는 새로운 지식을 추론하는 방법이다. 추론의 필요성은 초기 상황의 불완전성과 불확실성으로부터 출발한다. 상황추론의 기법은 3단계로 구성된다. 먼저, 전처리단계로 수집된 센서 데이터를 보정한다. 통신에서 발생한 오류를 없애고, 값을 보정하는 단계를 의미한다. 다음 단계에서는 센서 데이터 퓨전이다. 다양한 센서들로부터 생성된 센서 데이터를 결합하는 단계이다. 마지막 단계는 앞 단계에서 생성된 상황을 활용하여 더 높은 단계의 상황정보를 발생시키는 상황 추론 단계이다[1].

상황배포(context dissemination)는 상황을 소비자에게 보

내주는 방법이다. 질의(query)와 구독의 방법이 있다. 질의는 상황의 소비자가 상황운영 시스템에 질의함으로써 결과를 얻는 방식이며, 구독은 시스템이 주기적으로 또는 사건이 발생했을 때 결과를 보내주는 방식이다.

III. The Proposed Scheme

3. System Architecture

스마트팩토리에서 작업자 안전과 공장의 환경요소를 파악하고, 관제센터에서의 실시간 상황 모니터링을 위한 시스템 구조는 fig. 4와 같다.

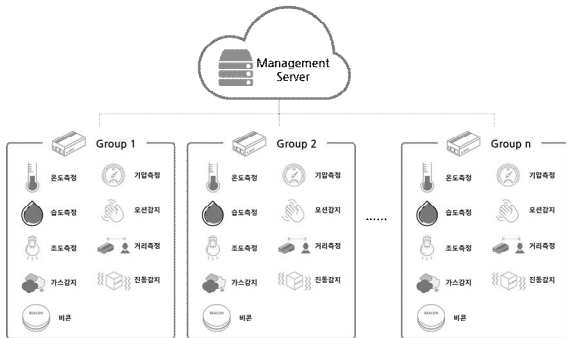


Fig. 4. System Architecture

각각의 센서들은 그룹으로 묶이어서 관리되며, 온도, 습도, 미센던지, 가스, 움직임들을 감지할 수 있다. 또한, 비콘의 사용이 가능하여 다양한 서비스를 제공하고 트리거 역할이 가능하다. 각각의 그룹에는 IoT 허브 기능의 R-Box(Recognition Box)가 있어 센서들을 관리하며 센서로부터 수집된 정보를 관리 서버로 보내는 역할을 한다. 다음절부터는 Fig 2의 IoT의 4 layer에 맞추어 시스템의 구성을 설명한다.

3.1 Object Sensing Layer

적용된 센서는 표 1과 같이 스마트팩토리에서 작업자의 안전과 공장의 환경요소 파악을 위한 센서들을 위주로 구성하였다. 이를 위하여 다양한 센서들을 테스트하였으며, 표 1과 같이 최종 6개의 센서와 비콘을 결정하였다. 초기에는 좀 더 많은 센서를 연결하려 시도하였으나 결국 전력에 문제가 생겨 수용할 수 없었으며, 같은 이유로 무선 센서의 적용도 문제가 되었다. 비콘은 스마트팩토리에서 작업자의 위치 확인이나 다양한 서비스의 트리거로 활용될 수 있을 것으로 고려되어 추가되었다.

Table 1. Application Sensor

sensor	use	feature
HC-SR501	sensing motion	infrared sensor, 0~7m
GP2Y1010AU0F	fine dust measurement	detect the light reflected from dust
MQ-2	gas measurement	LPG, hydrogen, methane, smoke
OSTSen-THL100	temperature, humidity,intensity of illumination measurement	-20°~ 120°, 0~100%RH, 0~5400Lux
RECO	beacon	bluetooth
DFR0027	vibration measurement	sensitive
HC-SR04	distance measurement	ultrasonic wave, 2m~4m

3.2 Information Integration & Data Exchange Layer

Fig 4.시스템 구성도와 같이 각 센서의 그룹에서 모여진 데이터는 관리 서버로 보내져야 한다. 각 그룹에서 자료를 교환하고 통합하는 작업은 IoT 허브인 R-Box에서 수행하게 된다.

R-Box는 라즈베리파이를 기반으로 구성하였다. 초기에는 라즈베리파이에 직접 센서들을 연결하였으나 확장에 문제가 있었고, 이를 수정하기 위하여 브레드보드를 활용하여 센서를 연결하였다.

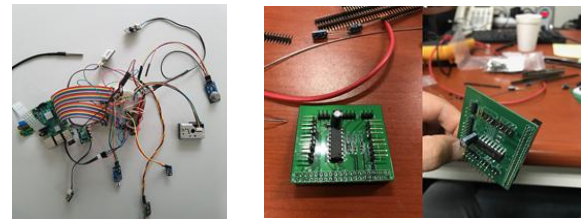


Fig. 5. PCB board construct

브레드보드의 사용은 센서의 확장성은 해결하였으나 크기와 배선의 복잡성의 문제가 생겨, 전용 PCB 보드를 설계하였다. Fig.5는 브레드보드의 사용과 전용 PCB 보드의 형태를 보여주고 있다. 전용 PCB는 센서의 확장과 크기 축소에 큰 역할을 하였다.

R-Box는 pull 방식으로 센서로부터 데이터를 수집하여 이를 관리 서버로 보내주는 역할을 하며, IoT의 4 layer 중 data exchange와 information integration 계층의 역할을 수행한다.

3.3 Application Service Layer

관리 서버(Management Server)는 응용 서비스 계층의 서비스를 제공한다. 사용자들은 앱과 웹을 통하여 손쉽게 접근하여, 센서들의 정보를 확인할 수 있으며, 수집된 정보는 데이터베이스에 누적되어 저장된다. 다수의 R-Box는 사용자 별로 관리되며, 접근 권한 기능을 가지고 있다.

Fig 6은 관리 서버와 개발 과정의 화면이다. 서버에서는 회원 관리를 위한 사용자 인터페이스를 제공한다. 사용자는 로그인 후 등록된 R-Box와 R-Box에 연결된 센서의 이름, 각 센서의 측정 정보, 작동상태, 최종 업데이트 시간, 누적 기록을 확인

할 수 있으며, 이에 대한 간단한 통계도 제공한다. 사용자 인터페이스에서는 센서 이름 변경, 센서 삭제, 새로 고침 등의 기능이 있다. 여러 개의 R-Box가 서버와 통신하고 있으므로 각 R-Box들에 연결된 센서 정보를 확인할 수 있으며 R-Box 이름 수정, R-Box 삭제, 검색, 메모 등의 기능도 제공해야 한다.

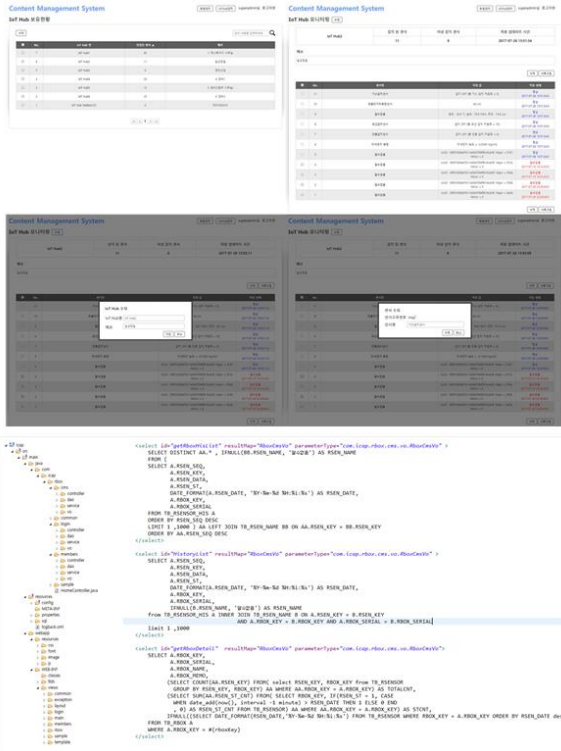


Fig 6. Management Server & Development Display

3.4 Database Design & Implementation

R-Box와 센서들의 정보를 저장하기 위한 데이터베이스의 E-R 다이어그램은 Fig 7과 같다.

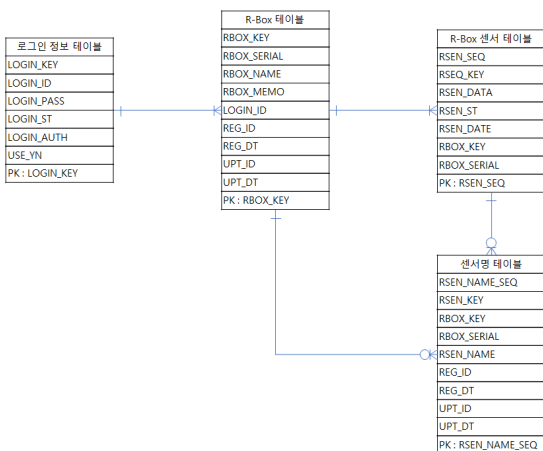


Fig. 7. E-R Diagram

로그인 정보 테이블은 로그인 기본키, 로그인 ID, 비밀번호, 로그인 상태 및 권한, 그리고 사용 여부로 구성되었으며, R-Box

테이블은 R-Box 기본키, 시리얼 번호와 이름, 메모, R-Box 사용자 ID, 등록인 ID, 등록일, 수정인 ID, 수정일로 구성되어, 사용자가 R-Box를 등록하고 이름을 붙일 수 있게 구성하였다.

센서에 관한 정보는 R-Box 센서 테이블과 센서명 테이블에 저장된다. 센서명 테이블은 센서에 관한 기본적인 정보를 저장하기 위한 테이블이며, R-Box 센서 테이블은 센서와 R-Box의 연결을 위한 테이블이다.

R-Box의 센서들은 모두 정보가 누적되어 지는데, R-Box 센서 테이블을 활용하여 계속하여 정보를 누적하게 된다.

Fig. 7의 E-R 다이어그램에 의하여 구성된 데이터베이스 스키마는 다음과 같다.

```

CREATE TABLE `TB_LOGIN` (
  `LOGIN_KEY` int(11) NOT NULL AUTO_INCREMENT,
  `LOGIN_ID` varchar(12) NOT NULL,
  `LOGIN_PASS` varchar(512) DEFAULT NULL,
  `LOGIN_ST` varchar(1) DEFAULT NULL,
  `LOGIN_AUTH` varchar(2) DEFAULT NULL,
  `USE_YN` varchar(1) DEFAULT NULL,
  PRIMARY KEY (`LOGIN_KEY`)
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT
CHARSET=utf8

CREATE TABLE `TB_RBOX` (
  `RBOX_KEY` int(11) NOT NULL AUTO_INCREMENT,
  `RBOX_SERIAL` varchar(100) NOT NULL,
  `RBOX_NAME` varchar(50) DEFAULT NULL,
  `RBOX_MEMO` varchar(512) DEFAULT NULL,
  `LOGIN_ID` varchar(12) NOT NULL,
  `REG_ID` varchar(45) DEFAULT NULL,
  `REG_DT` date DEFAULT NULL,
  `UPT_ID` varchar(45) DEFAULT NULL,
  `UPT_DT` date DEFAULT NULL,
  PRIMARY KEY (`RBOX_KEY`)
) ENGINE=InnoDB AUTO_INCREMENT=23 DEFAULT
CHARSET=utf8

CREATE TABLE `TB_RSENSOR` (
  `RSEN_SEQ` int(11) NOT NULL AUTO_INCREMENT,
  `RSEN_KEY` int(11) DEFAULT NULL,
  `RSEN_DATA` varchar(100) DEFAULT NULL,
  `RSEN_ST` int(11) DEFAULT NULL,
  `RSEN_DATE` datetime DEFAULT NULL,
  `RBOX_KEY` int(11) NOT NULL,
  `RBOX_SERIAL` varchar(100) NOT NULL,
  PRIMARY KEY (`RSEN_SEQ`)
) ENGINE=InnoDB AUTO_INCREMENT=205139 DEFAULT
CHARSET=utf8

CREATE TABLE `TB_SENSOR` (
  `RSEN_NAME_KEY` int(11) NOT NULL AUTO_INCREMENT,
  `RSEN_KEY` varchar(100) NOT NULL,
  `RBOX_KEY` varchar(50) DEFAULT NULL,
  `RBOX_SERIAL` varchar(512) DEFAULT NULL,
  `RSEN_NAME` varchar(12) NOT NULL,
  `REG_ID` varchar(45) DEFAULT NULL,
  `REG_DT` date DEFAULT NULL,
  `UPT_ID` varchar(45) DEFAULT NULL,
  `UPT_DT` date DEFAULT NULL,
  PRIMARY KEY (`RSEN_NAME_KEY`)
) ENGINE=InnoDB AUTO_INCREMENT=23 DEFAULT
CHARSET=utf8
    
```

Fig. 8. Database Schema

3.5 R-Box Protocol design & Implementation

R-Box와 연결된 센서들은 UART, I²C, SPI, Bluetooth 등 각기 다른 네트워크 프로토콜을 가지고 있다. 이러한 센서와의 통신은 R-Box가 수행하고, 이들의 정보를 모아 관리 서버와 통신을 수행한다. 서버와 통신은 TCP 기반의 새로운 통신 프로토콜을 정의하여 사용하였다. 정의된 프로토콜의 패킷 구조는 Fig. 9 Packet Structure와 같다.

Main Header (20 bytes)
Group Header (minimum 4 bytes ~ depends on protocol group type)
Protocol Header (minimum 6 bytes ~ depends on protocol type)
Protocol Body (minimum 0 ~ depends on each protocols and situations)
Checksum (4bytes, checksum between Group Header ~ Protocol Body, padding with 0)

Fig. 9. Packet Structure

사용하는 String은 항상 UTF-8 인코딩을 사용하고, 두 바이트에 문자열의 바이트 길이를 표시하고, 두 바이트 뒤에 'WO' 문자가 없는 문자열을 배열에 담아 전송한다. 패킷 구조의 각각의 헤더는 Fig 10. 패킷 헤더 구조와 같이 구성된다.

Main Header (Fixed)			
validation upper	validation lower	protocol ver upper	protocol ver lower
sender 0 : server 1 : client	protocol group type	protocol type (2 bytes)	
total length (Group Header ~ checksum) in bytes			
reserved 1			
reserved 2			
Group Header (Minimum fixed)			
reserved (4 bytes)			
Protocol Header (Minimum fixed)			
Request/Response flag (1 byte) 0 : request 1 : response	Request/Response message (2 bytes) 0 : Succeeded (for anything, it's positive) 1 ~ : Failed and reasons	reserved (1 byte)	
identifier string (2 + n bytes)			

Fig. 10. Packet Header Structure

TCP 데이터 구조는 R-Box의 정보와 센서의 정보를 구분하여 전송한다.

* R-Box 정보		* 센서의 정보	
4 int	rbox_key	4 int	rbox_key
2 + string	rbox_name	2 + string	rsen_key
2 + string	rbox_memo	2 + string	rsen_name
2 + string	login_id	2 + string	rsen_data
2 + string	reg_id	2 + string	reg_id
8 int	reg_dt	8 int	reg_dt
2 + string	upd_id	2 + string	upd_id
8 int	upd_dt	8 int	upd_dt
2 + string	rbox_serial	2 + string	rbox_serial
8 int	upd_dt	4 int	rsen_st
2 + string	rbox_serial	8 int	RSEN_DATE

Fig. 11. TCP Data Structure

데이터 보안을 위해 패킷은 암호화 후 전송되며, 암호화에 AES(Advanced Encryption Standard) 알고리즘을 활용하였다. 보안 프로토콜은 Fig 9의 TCP 기반 패킷 구조 전체를 AES 128bit 암호화하고, 보안 프로토콜 헤더를 붙였다. 보안 패킷 구조는 Fig 12와 같다.

Secured Packet Structure			
Secure Header (12 bytes)			
Encrypted Packet			
Secure Header (Fixed)			
validation upper	validation lower	protocol ver upper	protocol ver lower
protocol ver lower			
reserved			
Body (Encrypted Packet)			
AES 128bit Encrypted Packet			

Fig. 12. Secured Packet Structure

IV. Implementation and Testing

4.1 Development Environment

스마트팩토리 환경 정보를 수집하고 모니터링 하기위하여 개발한 사물인터넷 플랫폼의 개발 사양은 표 2와 같다.

Table 2. Platform Development Spec.

CPU	Intel Xeon processor E3-1230 v3 Quad core 3.3GHz
Board	ASROCK E3C222D4U Intel C222 Chipset
Memory	4GB DDR3 ECC/ Unbuffered X 2
Disk	SSD 120GB

관리 서버 구축에는 Linux계열 운영체제인 CentOS, Apache Tomcat 서버, Spring 프레임워크, MySQL을 사용하였으며, 개발 언어로 Java를 사용하였다.

4.2 Testing

스마트팩토리 환경정보 수집을 위한 사물인터넷 플랫폼을 테스트하기 위하여 플랫폼의 안정성과 에러 발생률, 센서 동시 연결확인 및 R-Box 정확도 등을 측정하였다.

시험은 한국시험 표준원에서 수행하였으며, 준비사항으로는 PC(노트북) 1대, R-Box 1대, 센서 6개, 플랫폼소프트웨어를 준비하였다.

플랫폼 안정성 시험 방식으로는 우선 R-Box 전원을 켜 후 랜을 연결하였고, PC를 켜 후 플랫폼 소프트웨어를 실행시켰다. 그 후 테스트 시작부터 종료 시간까지 센서에서 데이터를 보낸 후 시스템에 전달된 시간이 2초 이상 누락되는 가를 4시간 동안 확인하였다. 시험 결과 모든 데이터가 누락없이 플랫폼에 표시되는 것을 확인하였다.



시점	센서 ID	센서 이름	센서 위치	센서 타입	센서 상태	센서 데이터	센서 연결 여부
00:00	001	온도	1호선	온도	정상	25.5	연결
00:05	002	습도	1호선	습도	정상	65	연결
00:10	003	진동	1호선	진동	정상	0.1	연결
00:15	004	가스	1호선	가스	정상	0.5	연결
00:20	005	압력	1호선	압력	정상	101.3	연결
00:25	006	광도	1호선	광도	정상	100	연결

Fig. 13. platform stability test

에러 발생률도 플랫폼 안정성과 유사한 방식으로 검증하였으며, 4시간 9362회의 센서 데이터 발생이 모두 에러없이 플랫폼에 표시되었다.

센서 동시 연결과 R-Box 정확도는 R-Box에 다양한 센서를 연결할 수 있는지와 센서 측정값이 R-Box에 잘 전달되는 지를

테스트하기 위하여 6개의 센서와 비콘 73개를 연결하여 테스트하였다. 두 테스트 모두 아무 이상없이 연결되고, 값이 전달되는 것을 확인할 수 있었다.

Table 3. Testing Result

Platform stability	continuous connected for 4 hours
Error Rate	receive the 9362 data for 4 hours
Sensor connectivity	connected 6 sensors and 73 beacons
R-Box Correctness	received values for 6 sensors and 73 beacons

4.3 The Last R-Box Model

스마트팩토리의 환경정보를 수집하기 위해서는 R-Box가 진동과 습기, 그리고 센서들의 전원 공급에 문제가 없어야 한다. 이를 위하여 센서들의 연결선을 모두 R-Box 안으로 구성하고, 진동에 잘 견딜 수 있도록 일체형 케이스를 설계하였다.

센서와 전자기기 특성상 방수의 역할까지는 견디지 못하지만 방진과 어느 정도의 습기는 피할 수 있으며, 특히 진동에는 강점이 있어 효과적으로 스마트팩토리 환경 정보를 수집할 수 있을 것으로 기대한다.

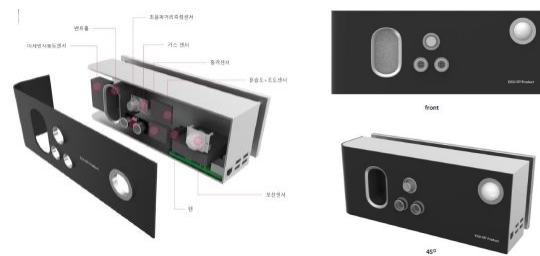


Fig. 14. The Last R-Box Model

V. Conclusions

본 논문에서는 스마트팩토리 환경정보 수집, 관리를 위한 사물인터넷 플랫폼을 설계 및 구현하였다. 이를 위하여 여러 센서를 통합, 관리하는 R-Box를 설계 및 구현하였으며, 사물인터넷 플랫폼으로 관리 서버를 설계하고 구현하였다. 또한 R-Box와 관리서버 간의 네트워크를 설계하고 구현하였다.

본 시스템을 활용하면 공장에서 최적의 작업 환경을 위한 온도, 습도 등을 측정할 수 있으며, 비콘을 활용하여 작업자의 근무 시간 측정 및 위치 확인 등을 통하여 작업자의 안전성을 높이고 근무 형태를 파악할 수 있어, 더 높은 효율을 얻을 수 있다.

본래의 목적은 스마트팩토리를 위한 ICAP(Intelligent Context Awareness Platform)의 설계 및 구현이었다. 스마트팩토리의 환경정보를 상황으로 인식하여 좀 더 지능적이고 좋은 정보를 제공하고자 하는 목적이 있었지만, 프로젝트의 성격

상 너무 오랜 시간을 R-Box에 소모하여 지능형까지는 발전시키지 못하였다.

그러나 지금의 스마트팩토리 환경정보 수집, 관리 시스템을 위한 플랫폼만으로도 작업자의 안전과 최적 작업환경 구성, 작업자의 작업 시간 측정 및 다양한 서비스 구성이 가능하다.

R-Box에 센서를 연결할 때 가장 문제가 되었던 것은 전원이었다. 가능한한 무선 센서를 이용하려던 계획은 각각의 무선 센서에 전원을 공급하는 문제로 인하여, R-Box에 직접 센서를 연결하는 방식으로 변경되었고 그나마 연결할 수 있는 센서의 개수도 한정되었다. 현재의 상태에서는 비콘만이 무선으로 연결하는 구조이며, 이는 비콘 자체에 배터리가 내장되어 가능한 결과이다. 연구도중 R-Box를 스마트 팜에 활용하는 계획도 잠시 있었지만 그렇게 하기 위해서는 완전한 방수 기능이 요구되어, 제한된 용도로 밖에 사용이 불가능하였다.

REFERENCES

- [1] P. Nurmi and P. Floree. Reasoning in context-aware systems. Position Paper. Department of Computer Science, University of Helsinki, 2004
- [2] Hua-Dong Ma. Internet of Things: Objectives and Scientific Challenges. Journal of Computer Science and Technology, 26(6) 919-924, Nov. 2011
- [3] KyungSik, Min, "Internet of Things", NET Term, Korea Internet & Security Agency, 2012. 6.
- [4] Charith Perer, Dimitrios Georgakopoulos, "Context aware computing for the internet of things : A survey", IEEE Communications Surveys & Tutorials, 2013.
- [5] DoHyun Kim, "IoT Context Awareness Technique Trend", The Proceedings of the Korea Electromagnetic Engineering Society 24(4), 20-26, 2013
- [6] Min Choi, KyungRok Jo, ByungChul Ahn, TaeJoon Kim, EiSin Lee, "The Prectice of Raspberry pi & ARM Embedded Programming", The Board of Publication Chungbuk National Univ. 2016.
- [7] JinKyung Kim, SangYong Ra, JaeHong Choi, JunDong Lee. "IoT Hub Implementaion." KSCI Summer Conference ,157-158, 2017.07
- [8] JinKyung Kim, SangYong Ra, JaeHong Choi, JunDong Lee. "Implementaion of Sensor N/W & CMS Server in R.Box" KSCI Winter Conference , 77-78, 2018.1
- [9] JinKyung Kim, JunDong Lee "Fine dust wireless transmission system Using Arduino" KSCI Summer Conference , 2018.7
- [10] SungWoo Yoon, "Yoon's The Heat TCP/IP Socket Programming", OrangeMedia, 2009.
- [11] <http://www.raspberrypi.org>

Authors



SungJin Kim is a Director in the department of Public Business at DataStreams Co., Ltd. He received as B.S., M.S. degree in Computer Science from Halla University Wonju, Korea in 2013, 2015 and currently receiving Ph.D. in data science part in

multimedia engineering at GangNeung-Wonju National University. He is currently a specialist in the Bigdata Consulting, at DataStreams Co., Ltd. His research interests include advanced data analyst, business process architecture.



SangYong Ra is CEO of EX2I. He finished Hanyang University of Industrial Engineering and worked for 23 years in the IT Business. He founded the company in 2014, and His company is currently working on services and research in the

IoT field.



HwanSeog Kim is a professor with the Dept. Information & Tele-communication Engineering at GangNeung-Wonju National University. He received as B.S., M.S., and Ph.D. degree in Electronic Engineering from Cheongju University, Cheongju, Korea

in 1988, 1990, and 2008, respectively. His research interests include IoT, Embedded Systems, and Medical Devices.



JaeHong Choi is a professor with the Department of Multimedia Engineering at Gangneung- Wonju National University. He received as B.S., M.S., and Ph.D. degree in Electronic Engineering from Hanyang University, Seoul, Korea in 1986, 1989, and

1997, respectively. His research interests include Mobile Business, IoT, and platform.



JunDong Lee is a professor with the Department of Multimedia Engineering at GangNeung-Wonju National University. He received as B.S., M.S., and Ph.D. degree in Computer Science from HongIk University, Seoul, Korea in 1990, 1993, and 2001,

respectively. His research interests include programming language, IoT, and platform.