[UCI]I804:24011-200000235927

2008年 2月
博士學位論文

# USN 기반구현을 위한 적외선 온도감시시스템에 관한 연구

朝鮮大學校 大學院

電 氣 工 學 科

全 馨 錫

2008年 2月 博士學位論文 USN 기반구현을 위한 적외선 온도감시 시스템에 관한 연구 全 馨 錫

# USN 기반구현을 위한
# 적외선 온도감시시스템에 관한 연구

**A Study on the Infrared Thermometer Monitoring System for USN**

2008年 2月 日

朝鮮大學校 大學院

電 氣 工 學 科

全 馨 錫

# USN 기반구현을 위한
# 적외선 온도감시시스템에 관한 연구

指導敎授　吳 金 坤

**이 論文을 工學 博士學位 論文으로 提出함.**

2007年 10月　日

朝鮮大學校 大學院

電 氣 工 學 科

**全　馨　錫**

# 全馨錫의 博士學位論文을 認准함.

委員長　朝鮮大學校　敎授 _____印

委　員　朝鮮大學校　敎授 _____印

委　員　朝鮮大學校　敎授 _____印

委　員　朝鮮大學校　敎授 _____印

委　員　朝鮮大學校　敎授 _____印

*2007* 年 *12* 月　　日

朝鮮大學校 大學院

# 목 차

*ABSTRACT*

*I.* 서 론 ····· 1

*II. Ubiquitous Sensor Network* ····· 5

A. 유비쿼터스 네트워크 ····· 5

1. 유비쿼터스 ····· 6

2. 유비쿼터스의 비전 및 접근법 ····· 8

B. 국내외 기술현황 ····· 10

1. 국 내 ····· 10

2. 국 외 ····· 16

C. 유비쿼터스의 응용 ····· 26

1. 유비쿼터스 센서 네트워크의 특징 ····· 26

2. Location Based Services ····· 29

3. Home Automation ····· 36

*III.* 유선기반의 센서네트워크 ····· 39

A. Power Line Communication ····· 39

B. Controller Area Network ····· 43

1. CAN의 구성 ····· 44

2. CAN의 동작원리 ····· 46

3. CAN의 메시지 구조 ····· 51

*IV.* 적외선 온도센서 ····· 53
A. 온도센서의 개요 ····· 53
B. 적외선 센서 ····· 57
C. A2TPMI 센서 ····· 60

*V.* 실 험 ····· 63
A. 시스템의 구성 ····· 63
1. USN 서버설계 ····· 65
2. USN 노드 감지기 설계 ····· 67
3. 감지기 프로그램 개발 ····· 71
B. 실 험 ····· 75
1. 온도 보정 실험 ····· 75
2. USN 네트워크 실험 ····· 82
3. 품질측정 실험 ····· 87

*VI.* 결 론 ····· 88

참고문헌

부 록

# *List of Tables*

Table 2-1 Propellant Structure of Each Nation's Ubiquitous ····· 17

Table 2-2 Relation of LCS and LBS ····· 29

Table 3-1 Composition of CAN Protocol ····· 48

Table 4-1 Description of Temperature Sensors ····· 54

Table 4-2 A Comparison of Contact and Non-contact Temperature Sensor ····· 56

Table 4-3 Characteristics Comparison of Infrared Sensor ····· 57

Table 4-4 Characteristics of Infrared Rays Sensor ····· 60

Table 5-1 Composition of Experiment System ····· 63

Table 5-2 Measurement Data ····· 81

# *List of Figures*

Fig. 2-1 Development Composition of Ubiquitous ····· 9

Fig. 2-2 Various Application of RFID/USN ····· 13

Fig. 2-3 The Connection of Disaster Communication Technic ····· 15

Fig. 2-4 Ubiquitous's Concept ····· 27

Fig. 2-5 An Application Example of Ubiquitous ····· 28

Fig. 2-6 Relation of LBS Relation Technic ····· 31

Fig. 2-7 Middle Ware Platform Structure of LBS ····· 33

Fig. 2-8 Visualization Structure of USN Server ····· 35

Fig. 2-9 Develop Direction of HA ····· 36

Fig. 2-10 Composition of HomeVITA ····· 37

Fig. 2-11 WEB Connection Method ····· 38

Fig. 2-12 Composition of Home ····· 38

Fig. 3-1 Structural Diagram of PLC ····· 39

Fig. 3-2 PLC Network Composition ····· 40

Fig. 3-3 PLC Modem ····· 40

Fig. 3-4 Composition of CAN System ····· 45

Fig. 3-5 Structure of CAN Packet ····· 46

Fig. 3-6 Composition of CAN Protocol ····· 52

Fig. 4-1 Structure Classification of Infrared Rays Sensor ····· 58

Fig. 4-2 Photo Conductive Mechanism of Infrared Rays Sensor 59

Fig. 4-3 A2TPMI Block Diagram ····· 61

Fig. 4-4 A2TPMI Sensor ····· 61

Fig. 5-1 Block Diagram of Full System Setup ····· 64

Fig. 5-2 The Adaptive System for Process ····· 64

Fig. 5-3 Block Diagram of USN Server Control Board ····· 65

Fig. 5-4 The Photograph of USN Server Control Board ····· 66

Fig. 5-5 Infrared Sensor Driving Block Diagram ····· 67

Fig. 5-6 Prototype of Designed Sensor ····· 69

Fig. 5-7 Microprocessor Circuit ····· 69

Fig. 5-8 Infrared Thermometer Monitoring System ····· 70

Fig. 5-9 ZF5585 Characteristics Curve ····· 72

Fig. 5-10 ZF5585 Adjust Curve ····· 72

Fig. 5-11 Main Program Flow Chart ····· 74

Fig. 5-12 Temperature Adjust ····· 75

Fig. 5-13 Response Characteristics ····· 77

Fig. 5-14 Interpolation Characteristics ····· 77

Fig. 5-15 Velocity Comparison of PT100[Ω] and USN Node ····· 78

Fig. 5-16 A Test Result of Standard Temperature Generator ····· 80

Fig. 5-17 Scan Timing of USN Node ····· 84

Fig. 5-18 Composition of Server CAN Network ····· 84

Fig. 5-19 A Traffic Analysis of CAN Network ····· 85

Fig. 5-20 Server Program of USN Simulator ····· 85

Fig. 5-21 Temperature Monitoring Result ····· 86

Fig. 5-22 Error Rate of Iron Rolling Process ····· 87

# *ABSTRACT*

## *A Study on the Infrared Thermometer Monitoring System for USN*

Chun, Hyung-Suck
Advisor : Prof. Oh, Guem-Gon, Ph.D.
Department of Electrical Engineering
Graduate School of Chosun University

In this paper, We are going to propose the infrared thermometer monitoring system with using CAN(Controller Area Network). We have developed an infrared radiation thermometer to measure the temperature of hot strip in very harsh environmental conditions. The optical fiber was used to overcome the environmental troubles of the pyrometer.

We suppose that we would apply CAN(Computer Area Network) that car-business of europe uses for independent control, high reliability, various expansion, high speed communication. this system can connect with current point-to-point type detector, in the bargain we would suppose the CAN-type detector that can concurrently monitor by receiving form of analogue signal, and suppose the system that as used CAN-type communication, can operate and more easily connect with without additional connected

device.

Therefore, we construct embedded system based on CAN communication that is capable of N:N communication, and build independent system.

If the abnormality condition is occurred in the installation, the problem is that how fast we can detect the abnormality condition and put if off by using available system. this is major factor that reduces damage of our installation. therefore in this studies We would like to design more stable system than current system.

We performed extensive field test for 6 months at the POSCO in Gwang-Yang, Korea. From the experimental test, we have confirmed the reliability of the developed Infrared Thermometer Monitoring System.

To have high sensibility and to gain proper output voltage, it should have highly responsive performance.

# I. 서 론

공정시스템에서의 CAN(Controller Area Network) 네트워크 응용으로 최근 물질문명의 발전, 생활권 확대 및 인구 과밀화로 인한 사회구조의 변화는 재해 발생 요인의 복잡 다양화를 가중시키고 있다. 특히 인위적인 재난 도시형 재난은 고도 정보화 사회의 도래와 고령화 사회로 변화함에 따라 과거보다 대규모화 및 특수화 되면서 그 위험성 또한 커지고 있다. 이로 인한 재난통합관리 시스템이 매우 중요한 이슈로 다가오고 있다.

대부분의 재난 사고는 일반적으로 사고 상황 내에 있는 구성원들의 인지과정과 관련되어 있다. 인지적 판단오류, 화재 상황에 대한 필요 정보의 지각 및 관련 기억/인지 정보의 인출 실패, 패닉 상황에서의 커뮤니케이션 기술결여 등의 인지적 정보처리 오류가 많은 재난사고의 대형화에 있다. 많은 현대 장비들은 복잡하고, 개별요소들이 긴밀히 연결된 시스템이다. 이 시스템들은 많은 부분들이 자동화되어 있어서 능숙하지 못한 사람에게는 재앙을 불러올 수 있다. 문제는 이러한 시스템들이 정교해지고 복잡해지는 속도가 인간의 인지정보처리 능력이 적응하기 힘들 정도로 빠르다는데 있다. 많은 경우 이 시스템들은 성능향상을 목표로 한 기술혁신에 치중하며, 인간의 안전과 수행안정성은 그에 따르지 못하는 문제도 있다.

유비쿼터스(Ubiquitous) 환경에서는 기존 시스템과의 결합성 및 새로운 환경에 부합된 새로운 시스템을 요구하고 있다. 따라서 기존의 재난 및 방범 관련 기기들과 유비쿼터스 기술을 부여한 기기들과의 연동 내지 통합화하는 방법에 대하여 연구되어 왔다. 이러한 과정을 통해 얻어진 종합정보

를 지수화 하며 효율적으로 처리할 수 있는 재난처리 시스템의 모형도 연구하고 있다. 현재의 IT 기술을 토대로 CAN/PLC 기반의 시스템 설계, GPS/GIS(Global Positioning System/Geographic information system) 형태의 단말기, 영상 재난감지기, 소방용 로봇, 광역 재난처리 시스템들의 설계방향등을 중심으로 연구되고 있다.

특히 유비쿼터스 기술을 기반으로 주위의 환경 변화를 감지하고, 재해 상황인가를 판단하여 기기 스스로 대책을 수립하여 재난에 대응할 수 있는 시스템의 제안과 가정에서 발생하는 화재나 방범 등의 재난상황을 유비쿼터스 환경하에서 처리 가능하도록 하는 유비쿼터스 기반 각종 감지기 연구와 이를 토대로 방재 시스템에 적합한 프로토 타입 설계에 대한 연구도 활발히 진행되고 있다. 이를 위해서 유비쿼터스 시스템에 접목이 가능한 재난 인식용 센서의 연구와 함께 재난 상황발생시 대처할 수 있는 재난상황 대응 알고리즘이 탑재된 유비쿼터스 재난 처리시스템의 개발과 차세대 재난관리 시스템의 구축방안도 중요한 이슈이다.

홈네트워크 응용에서는 최근 유비쿼터스란 용어가 신축중인 아파트 광고에서 많이 선보이고 있는 것처럼 유비쿼터스 기술이 홈네트워크 위주로 연구되고 있으며, 완전한 유비쿼터스 시대를 대변하지는 못하고 있다. 유비쿼터스의 개념적인 문제인 언제 어디서나 내가 원하는 정보를 얻을 수 있느냐는 질문에 아직은 기술적으로 미완인 상태이다. 기술적인 측면에서 아직 유비쿼터스 규격이 통일 되지 않아 보일러, 가스밸브, 도어락 등 모든 것이 따로따로 개발되어졌다. 심지어 아파트 회사별로도 그 규격이 통일되지 않았기 때문에 가전제품 위주로 사용 가능하다. 심지어 같은 회사제품에서도 일부 제한적인 모델 등만 가능하기도 한다.

제지 공정에서도 잉크의 확실한 점착을 위해서 정확한 온도측정이 요구된다. 종이는 드라이어를 366[m/min]으로 통과하기 때문에 온도를 측

정하여 측정데이터를 디스플레이 하는데 걸리는 시간은 3~4초의 여유밖에 없다. Web의 온도를 약 136[℃], 드라이어의 온도를 약 192[℃]로 일정하게 유지할 목적으로 설치형 적외선 온도계를 Web으로부터 46[cm] 떨어진 거리에 설치하여 10[cm] 직경을 측정한다. 적외선 온도계를 사용하기 전까지 항상 Web의 그을림 문제에 직면해왔으나 적외선 온도계를 통한 신속하고 정확한 온도측정으로 이런 문제를 해결할 수 있다.

제철공정에서는 고온을 일정한 온도로 유지해야 한다. 제철공정의 소결공정에서 소결광은 대차 아래에 있는 밸브에서 나오는 미스트(mist)로 냉각되면서 서서히 이동된다. 미스트는 소결광의 충진층 사이로 통과시켜 소결광 온도를 저하시킴으로써 소결광 생산량을 증대시킨다. 이를 위해서 지속적으로 온도를 모니터링 해야 하는데 가장 적합한 센서는 적외선 온도계이다. 그러나 일반적인 제품은 핸디형으로 공정 모니터링의 적용을 위하여 특정 네트워크 기반에 묶여 모니터링을 수행하고, 측정된 온도에 따른 대응모델을 시스템에서 유기적으로 실현할 수 있는 시스템이 필요하다.[1-2]

제철 압연공정에 사용 중인 기존 시스템에서 열연 코일을 생산하는 열간 압연공정에서의 압연 소재 온도는 열연제품의 강도 혹은 가공성 등을 결정하는 중요한 인자이다. 따라서 열간 압연공정에서는 라인 중간에 온도계를 설치하여 목표 온도에 맞도록 공정을 제어하고 있다. 최근에는 열연 제품에 있어서 전체 폭과 전체 길이에 대해 치수 및 기계적 성질이 균일하도록 노력하고 있다.[3-8]

본 논문에서는 균일한 온도를 확보하기 위하여 폭 방향으로 소결광의 폭 방향 온도분포를 모니터링 하는 동시에 아날로그 출력을 이용하여 미스트를 발생하는 밸브를 자동으로 컨트롤할 수 있고 미스트는 소결광 층

진층 사이로 통과하여 소결광 온도를 저하시킴으로써 소결광생산량을 극대화 할 수 있는 USN(Ubiquitous Sensor Network) 기반 구현을 하고자 한다.

CAN을 통해 적외선 센서의 온도값을 모니터링하고, 이를 시스템에 전송하여 온도 조절을 함으로서 양질의 철을 생산할 수 있도록 한다. 단일 서버에서 최고 4096노드의 클라이언트의 온도 감시를 수행함으로서 효율적인 공정관리를 하고자 한다.

본 적외선 온도 감시시스템은 100~200[℃]정도의 온도 분포를 갖는 재료들을 측정하기 위한 용도이며, USN 네트워크를 구성할 수 있는 단말 감지기와 공정제어에 참여하는 서버 등의 형태로 설계하고자 한다. 또한 적외선의 경우 방사율 특성을 많이 받게 됨으로 같은 철이라도 약간씩 상이한 특성을 지니므로 이를 매뉴얼 방식으로 방사율 조정을 할 수 있는 기능을 추가하여 정확한 온도 측정을 확립하고자 한다.

# II. Ubiquitous Sensor Network

## A. 유비쿼터스 네트워크

### 1. 유비쿼터스

유비쿼터스란 라틴어로 '편재하다, 보편적으로 존재하다'라는 의미이다. 모든 곳에 존재하는 네트워크라는 것은 지금처럼 책상 위 PC의 네트워크화뿐만 아니라 휴대전화, TV, 게임기, 휴대용 단말기, 카 네비게이션, 센서 등 PC가 아닌 모든 비 PC 기기가 네트워크화 되어 언제, 어디서나, 누구나 대용량의 통신망을 사용할 수 있고, 저 요금으로 커뮤니케이션 할 수 있는 것을 가리킨다.[9-10]

1998년 유비쿼터스란 용어를 처음으로 사용한 미국 제록스 팰로앨토 연구소의 마크 와이저(Mark Weiser) 소장은 유비쿼터스 컴퓨팅이 메인프레임, PC에 이은 제3의 정보혁명의 물결을 이끌 것이라고 주장하였다.

일본의 트론(TRON) 프로젝트를 주도해 세계의 주목을 받은 바 있는 도쿄대 사카무라 켄 교수는 저서 '유비쿼터스 컴퓨팅 혁명'을 통해 '선진국의 경우 저성장 사회로의 이행이 가속화되고 있는데 유비쿼터스 컴퓨팅은 지속적 성장이 가능한 순환형 시스템의 정착을 가능하게 해줄 것'이라고 전망하고 있다. 그는 저서에서 유비쿼터스 환경에서는 정보습득과 활용이 최적화돼 소모성 자원의 효율적인 사용이 가능해질 것이며, 유비쿼터스 컴퓨팅이 대량 생산의 획일적인 '하드와이어드' 사회를 개개인의 다양성에 적절하게 대응할 수 있는 '프로그래머블' 사회로 탈바꿈시켜줄 것으로 전망하였다.

세계 최대의 소프트웨어 업체인 마이크로소프트(MS)의 빌게이츠 회장은 컴덱스 기조연설에서 'SPOT(Smart Personal Object Technology)'를 새로운 화두로 제시했다. SPOT의 스마트 오브젝트는 인터넷 기능을 구현해 언제, 어디서나 온라인에 손쉽게 접속할 수 있도록 해주는 알람시계, 부엌용 전자기기, 스테레오 장비 등과 같은 소형 전자기기. 즉 유비쿼터스를 다르게 표현한 것으로 전 세계 IT산업에 가장 큰 영향력을 행사하는 인물 중 하나인 게이츠가 유비쿼터스 시대의 본격적인 개막을 선언하였다.

이처럼 유비쿼터스는 최근 전 세계적으로 최대 화두로 다뤄지고 있으며, 유비쿼터스의 실현으로 실세계의 각종 사물들과 물리적 환경 전반 즉, 물리공간에 걸쳐 컴퓨터들이 편재되게 하되 사용자에게는 겉모습이 드러나지 않도록 환경 내에 효과적으로 심어지고 통합되는 새로운 정보통신 환경의 구축이 예상된다.[11-12]

유비쿼터스는 농업 혁명, 산업화 혁명 그리고 정보 통신 혁명에 이어 새로운 패러다임을 말하고 있고, 유비쿼터스는 단순히 혁신 기술만을 말하는 것이 아닌 모든 현실 공간의 사물(object)과 가상공간이 하나로 어울러져 조화를 이루는 그러한 혁명을 뜻한다.

예전의 정보통신화에서 말하던 자동화와 혁신 기술은 인간의 우선순위를 높게 측정하지 못했지만, 유비쿼터스에서는 주로 인간의 5감, 즉 문맥을 다루기에 모든 기술이 인간 중심적인 기술로 부상하게 될 것이므로, 인간을 중심으로 구체화되는 컴퓨팅 환경이라고 하면 이해기 쉬울 듯하다. 인간이 자연스럽게 컴퓨터에 접근하게 되고, 컴퓨터는 그에 맞게 환경을 제공하게 된다. 인간이 위치하고 있는 환경의 특성을 모색한 뒤, 그가 필요로 하는 서비스를 제공하게 되며, 더 나아가서는 인간의 심리 파악 역시 가능하여 인간의 선호도에 따른 서비스가 가능할 것이다.[13-14]

RFID와 같은 태그 기술을 사용하여 대표적으로 물류 관리에 큰 효과를 줄 수 있다. 센서 기술 센서를 사용하여, 사물 또는 인간의 위치를 파악 할 수 있고, 사물이 움직임에 즉각적으로 반응을 할 수 있다.

IPv6와 같은 네트워크망이 구축되어야 다양한 기기들을 연결하여 제어할 수 있게 된다. 이는 통신을 위한 네트워크 기반이다.

다양한 기기들을 제어하는데 있어서 필요한 프로그램 컴포넌트이다. 사용자가 여러 기기들을 제어하기 위해서는 이들에 맞는 프로그램이 필요한데, 이를 미들웨어 기술이라 한다.

사용자가 시스템과 대화하기 위해서 필수 불가결한 부분으로 운영체제라고도 말 할 수가 있겠고, 사용자가 가장 접하기 쉬운 형태의 인터페이스가 개발되어야 한다. 이와 관련하여 HCI (Human Computer Interface) 연구가 현재 진행되고 있다.

텔레메틱스는 자동차내의 안전에서 엔터테인먼트에 이르기까지 폭 넓은 정보를 통합 관리하는 최신의 무선 음성 데이터 통신 기술이다.

지능형 홈, “스마트 홈” 또는 “디지털 홈”이라고도 하는데 이는 미래 신개념의 주택으로 비바람을 피하기 위한 물리적인 의미의 주택뿐만이 아니라, 인간의 욕구를 충족시키기 위해 안전성, 쾌적성, 편리성 및 표현성을 제공한다. 그리고 인간이 필요로 하는 각종 서비스를 네트워크화 된 여러 생활 가전을 통해 질 높은 서비스를 제공하게 된다.

## 2. 유비쿼터스의 비전 및 접근법

유비쿼터스를 단지 신기술로 이해하는 것보다 유비쿼터스의 철학적 의미를 우선적으로 깨달아야지만 유비쿼터스가 무엇이고, 유비쿼터스가 우리 삶에서 어떠한 실질적인 의의를 가질 것인가에 대한 질문에 대한 답을 얻을 수 있을 것이다. 즉 유비쿼터스의 모든 프로젝트 기법들은 비전(Vision), 환경(Context), 프로세스(Process) 3가지 관점에 의해 이루어지며 이 3가지는 매우 기초적인 유비쿼터스의 원리이자 관점이다.

비전은 사용자가 환경을 보는 시각 또는 사물이 사용자를 인식하는 관점으로 서로를 인식하는 과정에 의해 사물과 사람이 특정한 임무를 협력 방법을 통해 달성 할 수 가 있다. 그렇다면, 비전이란 단지 시각적인 차원에서만 말하는 것이 아니라 특정한 유비쿼터스 제공자가 사람들에게 제공하고자 하는 서비스가 어떤 유형이 될 것인지에 대한 비전일 수도 있다. 서비스가 제공하게 될 사용자 선호도에 부합하는 조건들을 미리 예측하는 것 또한 비전이라고 말할 수 있다.

환경은 문맥이라고도 말하며 유비쿼터스 세상에 있어 가장 중요한 것이 문맥이다. 문맥을 파악하여야만 사용자에게 어떤 유형의 서비스를 제공해야 할 것인지, 어떤 서비스를 제공해야 할 것인지에 대한 의사결정을 내릴 수 있기 때문이다. 사람의 5감을 문맥으로도 표현을 할 수 있고, 각종 무선 단말기들이 이를 인식하며, 사용자가 선호하는 서비스를 제공하게 되므로, 어떤 유형의 환경 하에 서비스가 이루어질 것인지에 대한 파악 능력이 필요하게 된다.

컴퓨터 구조론에서 말하는 프로세스는 특정한 명령어가 실행되어지고 있는 간격(Duration) 또는 단계(Procedure)를 말한다. 어떤 단계에 걸쳐서 문맥을 인식을 파악하여야 하는가에 대해 고려되어야 한다. 그리고 한

단계에서 다음 단계를 거쳐 갈 경우 변수나 제약 조건을 지정하여야만, 유비쿼터스의 프로세스가 사용자의 선호도에 알맞은 유연성(Flexibility)을 가질 수 있게 된다.

유비쿼터스의 핵심 기술은 인간의 삶의 질을 높일 것이라고 학자들은 예측하고 있다. 현재까지, 이에 대한 부작용이 적어 기술력이 발달함에 따라 유비쿼터스라는 유토피아가 형성될 것이라 생각한다. 유럽, 미국, 일본에서 유비쿼터스에 대한 연구와 프로젝트를 통한 문맥을 컴퓨터로 표현하고 인지하기 위한 노력이 곧 사회를 유비쿼터스 유토피아를 형성하는데 이바지 하게 될 것이다. 그리고 이를 위해서는 정부와 기업간의 정책을 통해 기술 표준화와 여러 보안 문제를 고려해야 할 것이다. 그러므로 이 사회에서 가상공간이 아닌 현실 공간에서 일어나고 반응하는 모든 일에 대해 컴퓨터라는 장비(Device)가 인식한다면, 유비쿼터스의 유토피아는 현실화 된다는 것이다. 그림 2-1은 유비쿼터스의 발전구도를 나타내고 있다.[15-16]

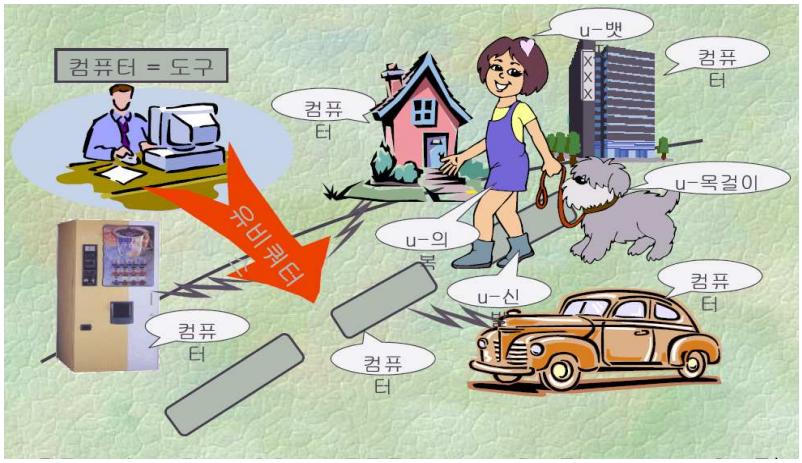



Fig. 2-1 Development Composition of Ubiquitous

## *B.* 국내외 기술현황

### *1.* 국 내

정부는 장차 e코리아를 u코리아로 업그레이드한다는 장기 구상을 세워 놓고 있다. 과학기술부가 지난 2003년 9월 새로운21세기 프런티어연구개발사업의 일환으로 추진하는 유비쿼터스 컴퓨팅 네트워크 기반기술 프로젝트는 앞으로 10년 동안 1000억원의 정부자금이 투입되는 유비쿼터스 분야의 첫 번째 대형 국책 연구 사업이다. 한편, 국내 주요 IT기업들도 유비쿼터스 시장 선점을 위한 준비 작업에 한창이다.[17-18]

삼성은 홈네트워크 브랜드 홈비타(Home VITA)를 발표, 이동전화에 TV/PC-카메라/캠코더/네비게이터/MP3플레이어/무전기 등 각종 모바일 기능을 탑재한MITs M400 출시 하였고 IPv6 활성화를 위한 한국전산원과의 공동 연구, 삼성그룹 계열사 유비쿼터스 공동 사업 등을 추진 중이다.[19-20]

LG그룹도 홈네트워크 브랜드 LG홈넷(HomNet)을 발표 하였고 건국대와 리눅스 임베디드 소프트웨어 공동 연구 추진(LG CNS) 중이며 냉장고를 홈 서버로 한 전력선통신(PLC)을 기반으로 한 LNCP(Living network control protocol) 규격의 독자개발 등을 진행 중이다.

SK텔레콤도 언제 어디서나 고객이 원하는 형태의 서비스를 제공하는 유비쿼터스 서비스 제공자로의 변신을 선언하면서 준(June), 네이트(NATE), 모네타(MONETA) 등 각종 멀티미디어 휴대폰 서비스와 함께 의류/가전/식품 등 모든 제품에 극소형 칩을 내장한 홈 네트워킹 사업을 구상중이다.

KT도 국내 최대의 통신 네트워크를 보유하고 있다는 장점을 최대한

활용해 유비쿼터스 전략을 추진 중이다. KT는 ALL IP, 브로드밴드, 유무선 통합, e포털 등 4대 분야를 중심으로 고객에게 맞춤화된 솔루션을 제공할 계획이다.

유비쿼터스 컴퓨팅은 21세기 새로운 IT혁명으로 불리며, 우리가 상상도 하지 못할 정도로 사회·경제·문화 등 모든 분야에 큰 영향력을 미치게 될 것이다. 모든 사물이 지능화되고 네트워크화 함으로써 사람과 사람, 사물과 사람, 나아가 사물과 사물 간에 의사소통이 가능한 지능기반사회(ubiquitous society)로 발전될 것이다. 우리나라 IT산업은 불과 20여년 만에 국가경제를 떠받치는 중추 산업으로 자리를 잡았을 뿐만 아니라 전 세계가 인정하는 IT강국으로 도약하였다. 미래성장이 가능한 지능기반사회로 진입하기위해 정부에서는 "IT 8-3-9 전략"을 추진하고 있다.

IT산업은 정보통신서비스와 네트워크, 기기, S/W및 콘텐츠가 밀접한 가치사슬을 이루면서 동반 성장한다. "IT 8-3-9 전략"이란 8대 신규 정보통신 서비스를 도입 및 활성화하여 3대 첨단 미래인프라에 대한 투자를 유발하고, 이를 바탕으로 9대신성장 동력산업이 동반 성장하는 발전전략을 말한다. IT분야 서비스 보급촉진을 위해 2.3[GHz] 휴대인터넷, 위성 및 지상파 DMB, 홈네트워크, 텔레매틱스, RFID(Radio frequency identification) 등 5가지신규 서비스 도입을 준비하고, W-CDMA, 지상파 DTV, 인터넷전화 등 3가지 기존 서비스의 활성화가 추진된다. BcN, USN(Ubiquitous sensor network)과 인터넷 IPv6 구축을 본격 추진하여 디지털 컨버전스 추세와 유비쿼터스 환경에 부합하는 첨단 인프라 구축을 통해 IT 신산업의 성장기반을 마련할 계획이다. 따라서 RFID 및 USN은 IT 8-3-9 전략의 중요한 역할을 맡고 있다.[21-24]

***a.*** RFID/USN

필요한 모든 것에 RFID를 부착하고 이를 통하여 사물의 인식정보를 기본으로 주변의 환경정보 온도, 습도, 오염정보, 균열정보 등을 탐지하여 이를 실시간으로 네트워크에 연결하여 정보를 관리하는 것을 말하는 것으로 궁극적으로 모든 사물에 컴퓨팅 및 통신기능을 부여하여 anytime, anywhere, anything 통신이 가능한 유비쿼터스 환경을 구현하기 위한 것이다. RFID/USN은 인식정보를 제공하는 RFID를 중심으로 발전하고 이에 감지기능이 추가되고 이들 간의 네트워크가 구축되는 USN 형태로 발전할 것이다.

향후 RFID/USN의 이용은 칩의 가격, 크기, 성능 등 센서 기술의 발전에 따라 시장에서의 적용이 확산되면서 단계적으로 발전할 것으로 예상된다. RFID 태그가 소형화, 지능화되는 반면에 가격은 수 센트로 저가화가 실현되면서 물류, 유통분야 및 환경, 재해예방, 의료관리, 식품 관리 등 실생활의 활용이 확대될 것으로 전망된다. RFID에 통신기능이 부가되고 점차 주위환경을 감지하는 센싱 기능이 부가되어 능동적으로 정보를 처리하는 지능형 초소형 네트워크 센서로 발전할 것이다. 현재의 고정된 개체 인식코드 획득수준에서 2010년경 다기능 태그에 의한 상황인지 처리수준으로 진화하고, 2015년 이후에는 개체 간 통신기능을 갖춘 지능형 USN으로 발전할 것이다. 적용분야는 판매, 유통, 교통, 식품관리, 위조방지, 의약품관리, 환경보호, 안전진단 등 사회 모든 부분에 적용된다. 그림 2-2는 RFID/USN의 다양한 응응분야를 나타내고 있다.[13]

최근에 우리나라는 정통부, 산자부, 과기부를 중심으로 RFID/USN, 유비쿼터스 컴퓨팅 관련 프로젝트를 추진 중이다. 삼성SDS, 삼성전자, LGCNS, LG산전, 신세계I&C, KT, SKT 및 중소기업들이 각종 응용분야에 관심을 갖고 연구개발을 추진 중이며, 국내에서는 대부분

13.56[MHz]를 이용하여 생산 공정이나 교통카드, 도서관 관리 등에 사용되어지고 있다. 국내에서 선도 가능한 기술 분야에 산·학·연 연구역량을 집중하여 2010년까지 핵심적인 요소기술을 단계적으로 개발하여 세계 최고수준의 기술경쟁력을 확보할 계획이다. 다양한 응용모델 개발과 세계 최고수준의 인프라를 활용하여 초기 수요창출을 위한 시범서비스를 실시, 서비스 공급기반 확립 등 시장형성에 주력하여 수요확대와 산업경쟁력 강화도 추진할 계획이다.

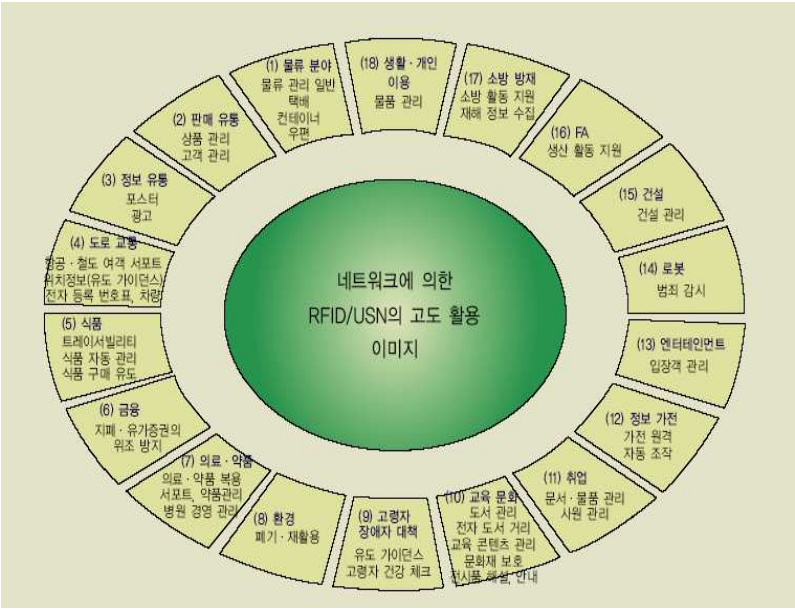



Fig. 2-2 Various Application of RFID/USN

***b.*** 재난 통신 기술

정보통신기술의 발전으로 재난 대비의 방재정보 관리시스템을 도입함으로써 적절한 사고예방과 사후관리는 물론, 보이지 않는 재난요소에 대한 경보를 제공하고, 재난이 발생하더라도 파급효과를 극소화하는데 매우 유용할 것이다. 따라서 통신시스템의 신뢰성 향상, 신속한 복구 등을 기본으로, 각종 재해 유형별 통신시설을 강화하고, 언제 어디서 발생할지 모를 재난에 대비하여 정보통신망이 제 기능을 발휘할 수 있도록 해야 한다.

재난상황에서 사용하는 통신기술의 국제표준을 제정하여, 장비간의 상호 접속 및 연동이 가능하도록 함으로써 국제적으로 협조, 공동 대처가 가능하도록 하며 차세대 통신망(NGN) 및 차세대 이동통신망에서 차세대 기술을 활용하여, 비상상황에서도 고속데이터, 디지털영상 및 음성정보를 전달하는 체계를 구현하며, 기존의 통신망 PSTN, ISDN, TRS, 셀룰라 이동통신 등을 이용한 비상통신 시스템에서, 통신망이 인터넷, NGN, 3세대 이동통신 등으로 기술이 발전함에 따라 기술적, 서비스측면에서 발전된 비상통신시스템으로 진화해야 할 것이다. 차세대 인터넷 및 차세대 이동통신기술을 활용하여, 고속 데이터, 디지털 영상 및 음성정보를 전달할 수 있도록 하기 위한 방안을 연구 중이다. 그림 2-3은 재난통신기술의 연계관계를 나타내고 있다.[23-24]

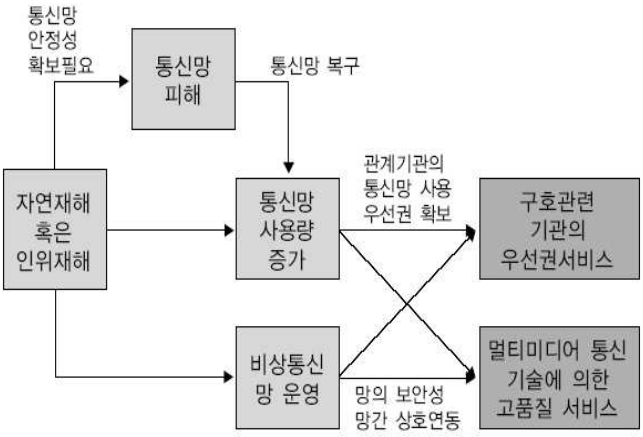



Fig. 2-3 The Connection of Disaster Communication Technic

## 2. 국 외

90년대 들어 유비쿼터스 컴퓨팅은 모든 정보통신 기술 및 서비스 분야의 신개념으로 인식됐고 미국, 일본 등 선진국들을 중심으로 이에 관한 연구를 활발하게 진행하고 있다. 제록스 팰러앨토 연구소에선 종이컴퓨터, 허리컴퓨터, 안경컴퓨터 등 이름도 생소한 이른바 유비쿼터스 컴퓨터를 개발 중이다. MIT랩, 케임브리지대학, IBM, 소니 등은 입는 컴퓨터를 개발하고 있다. 최근에는 이러한 연구들이 보다 진전되어 유비쿼터스 서비스를 상용화하기 위한 노력들이 마이크로소프트/소니/AT&T 등을 중심으로 진행되고 있다. 표 2-1은 각 나라별 유비쿼터스 추진체계를 나타내고 있다. 수년 전부터 20개의 유비쿼터스 관련 대형 프로젝트가 진행 중이고 유럽은 17개, 일본은 6개의 연구계획을 추진하고 있다. 유비쿼터스는 21세기 국가경쟁력을 좌우할 핵심기술이며 오는 2013년까지 전 세계적으로 1조3000억 달러의 부가가치를 창조할 것이라 예측 된다.[13-14]

유비쿼터스 IT는 새로운 지식정보 국가 건설과 자국의 정보산업 경쟁력 활성화를 위한 신전략으로 인식되고 있다. 따라서 주요 선진국의 정부와 기업들은 유비쿼터스 환경을 구축하고 관련 시장을 선점하기 위한 경쟁을 치열하게 전개하고 있다. 특히 미국, 일본, 유럽의 국가들은 산·관·학·연의 유기적인 협력체계를 구축하며 적극적인 투자와 활발한 연구를 진행하고 있다. 유무선 통합을 중심으로 한 네트워크, 디바이스, 서비스 분야의 디지털 컨버전스가 진행되고 있는 가운데 개념적으로만 접근되던 유비쿼터스란 용어는 점차 구체화되어가고 있는가 하며, 주요 IT 기업들과 대학을 중심으로 다양한 프로젝트가 추진되고 있으며, 실험적인 제품 및 서비스들을 테스트함으로써 새로운 비즈니스 모델을 개발하고 시장 선점을 추구하고 있다.

Table 2-1 Propellant Structure of Each Nation's Ubiquitous

| 구분 | 미 국 | 일 본 | 유 럽 | 한 국 |
|---|---|---|---|---|
| 시작 시기 | 1991년 | 2001년 | 2001년 | 2003년 |
| 계 기 | 고성능 컴퓨팅법 제공 | 유비쿼터스 네트워크 기술의 미래 전망에 대한 조사연구회 발족 | 사라지는 컴퓨팅 계획 시작 | u-Korea Forum 창립 |
| 추진 추체 | DARPA (국방부산하) | 정부주도의 산학연 연합체 | EU 주도의 전문연구기관 | 정부주도의 산학연 연합체 |
| 추진 방향 | Xerox, HP, MS, IBM, MIT, UC Berckeley 등 | NTTm 소니, NEC, 미쓰비시 등 | ETH(스위스), Tecc(독일) 등 | ETRI, SK Telecom, 삼성, LG 등 |
| 주요 프로 젝트 | 유비쿼터스 컴퓨팅 기술과 응용개발 | 마이크로 기술에 의한 유비쿼터스 네트워크기술개발 | 유비쿼터스 컴퓨팅 기술과 응용개발 | 유비쿼터스 네트워크기술과 유비쿼터스 기술개발 |
| | Smart Dust, Cool Town, TTT, Easy Living 등 | 초소형칩 네트워크, 무엇이든 MY단말 등 | Smart Its, Grocer 등 | USN 등 |

*a.* 미국

미국의 경우, 네트워크 및 정보기술개발 계획(NITRD)에 의거하여 국가차원의 IT 전략 프로그램을 각 부처·기관이 부처별 역할 분담과 제휴뿐 아니라 학계, 민간과도 폭넓은 연계를 통해 추진하고 있다. 특히 국방부 산하의 고등연구 계획국(DARPA)과 미국 기술표준의 총 본산인 국립표준기술원(NIST) 등의 정부기관이 대학 연구소 및 민간기업의 유비쿼터스 컴퓨팅 프로젝트에 대해 연구자금을 지원함으로써 유비쿼터스 컴퓨팅 연구에 중추적인 역할을 수행하고 있다.

미국은 IT기술 선진국으로서 유비쿼터스 컴퓨팅에 대한 거의 모든 분야의 기술을 주도하고 있으며, 특히 유비쿼터스 컴퓨팅화에 대한 프로젝트 추진은 정보통신업체와 대학교 연구실에서 주로 이루어지고 있다. 미국은 유비쿼터스 컴퓨팅 기술과 부분적인 조기 응용개발에 중점을 두고 있으며, 특히 일상생활 공간과 컴퓨터간의 자연스러운 통합이 가능한 HCI(Human Computer Interface) 기술과 표준 개발을 핵심요소로 인식하고 있다. 유비쿼터스 컴퓨팅화에 대한 프로젝트 추진방향은 민간 기업체의 주도로 주요 하드웨어, 네트워크, 소프트웨어 기술을 개발할 뿐만 아니라 이를 기반으로 국방, 의료, 산업, 가정 그리고 사무실 전반에 적용할 프로젝트를 진행하고 있다.

미국 연방정부는 인터넷의 폭발적인 확산으로 네트워킹에 대한 관심이 높아짐에 따라 1997년에서 2000년까지 차세대 인터넷(NGI) 등의 네트워킹 연구 분야를 중시하였으나, 2001년부터 고성능 컴퓨팅에 관한 연구개발 투자 비중을 다시 늘리고 있다. 2004년 부시행정부의 유비쿼터스 컴퓨팅 관련 예산은 약 18억 6,700만 달러로 꾸준히 증가하고 있음을 나타내준다.

미국 연방정부의 정보통신기술 연구개발 방향은 장기적으로 페타(Peta)

급의 고성능 컴퓨팅 기술 개발을 목표로 하고 있다. 이를 위해 2003년부터 나노기술, 바이오 기술과 컴퓨팅 기술을 결합한 복합화 된 신기술을 창출함으로써 최첨단 컴퓨팅에 필요한 요소기술 및 시스템 디자인기술을 획기적으로 발전시키는 한편, 최첨단 컴퓨팅 기술을 응용하기 위한 툴과 애플리케이션 등의 개발을 추진하고 있다. 대규모 네트워킹 기술도 장기적으로 페타급 통신기술 확보를 통해 최첨단 컴퓨팅과 초고속 네트워크간 컨버전스 기술 분야에서 세계적 리더십 확보를 지향하고 있다. 그리고 전광통신기술(All optical networking)을 기반으로 무선, 이동망 등을 언제, 어디서나 연결할 수 있는 하이브리드 통신기술, 수십억 개의 임베디드화 된 센서를 연결하기 위한 센서네트워크 그리고 신뢰성과 안정성을 제고하기 위한 기술개발 등을 추진하고 있다.

소프트웨어기술은 3개 프로그램에 의해 추진되는데, 주요내용은 사용자 인터페이스기술, 컴포넌트 소프트웨어, 임베디드 응용소프트웨어 그리고 소프트웨어 디자인 및 생산성 향상 기술, 고 신뢰성 소프트웨어 및 시스템기술 등이 포함된다. 특히, 소프트웨어는 인간과 유사한 수준의 기능을 수행하는 소프트웨어기술을 확보하기 위해 혁신적인 소프트웨어 개발방법과 디자인에 대한 연구를 강조한다. 다시 말해 자가진단(Self diagnosing), 자가 수정(Self correction),자가 치유(Self healing) 그리고 사용자의 실수나 외부의 무단침입에 대한 방어기능 등을 갖춘 소프트웨어를 개발함으로써 인간의 심장이나 폐와 같이 어떠한 상황에서도 자동적, 지속적으로 작동하고, 신체의 간과 같이 일부가 파괴되거나 이물질이 유입돼도 전체 시스템은 지속적으로 정상적 기능을 수행하는 고도의 소프트웨어 기술을 개발하는 것을 장기적인 연구개발 방향으로 제시하고 있다.

*b.* 일본

일본은 유비쿼터스 컴퓨팅의 역사에서 가장 앞선 나라일 수도 있었다. 실제로 1984년에 사카무라가 제시한 Tron 프로젝트 중에서 사람과 사람(Person to Person: P2P) 또는 사람과 물체(Person to Object: P2O)를 넘어서 물체와 물체(Object to Object: O2O)가 대등하게 인터페이스하는 네트워크의 개념이 등장한다. 다시 말해, 초기 Tron 구상에서는 지역분산 컴퓨팅 인프라를 배경으로 하여 소위 '지능적 물체'들이 네트워크로 연결되어 작동하는 유비쿼터스 개념의 원형이 발견된다. 그러나 Tron 구상의 이러한 측면은 한동안 주목을 받지 못하다가 1990년대 들어 미국에서의 유비쿼터스 개념의 부상에 편승하여 일본에서도 다시 관심을 끌게 되었다.[9-13]

일본은 정부 주도로 2001년도부터 전문가들로 구성된 조사연구회를 발족하여 종합적인 추진계획을 이미 수립하였다. '유비쿼터스 네트워크 기술의 미래 전망에 관한 조사 연구회'의 연구내용은 유비쿼터스 네트워크 기술의 현황, 일본과 국외의 네트워크기술의 연구개발 동향, 유비쿼터스 네트워크기술의 미래 모습, 대응해야 할 연구개발 과제와 표준화 문제, 그리고 실현에 의한 사회·경제적 효과를 검토하였다. 그리고 유비쿼터스 네트워크 실현을 위한 연구개발 추진방향을 설정하고 다음과 같은 구체적이고 야심찬 3가지 중점 프로젝트를 제안하였다. 일본은 국가차원의 IT 전략 추진을 법제화하고 수상을 본부장으로 하는 IT 전략본부를 설치하여, IT 최첨단 국가로 진입하기 위한 전략을 수립하고 있다. 정부 주도에 의한 산·학·관 연합체를 형성하여 추진하고 있으며, 통신 및 가전업체들도 적극 참여하고 있다.

이러한 U-Japan의 구상에는 두 가지의 중점 전략이 포함되어 있는데, 첫째는 '지역 안심 안전 액션 플랜'으로서, 지역의 활성화를 대전제로 신

변 주변의 생활공간에 있어서 안심·안전의 확립이 긴급한 과제이기 때문에 자주방재조직이나 커뮤니티의 주민 파워를 활용하여 지역의 안전·안심을 구축하기 위해 방재·방범 등에 있어서 광범위하게 대응하는 지역거점·네트워크를 창출하는 것이다. 그리고 다음으로는 '행정구조 개혁'으로서, 국민본위의 효과적이고 간소하고 효율적인 행정을 실현하여 민간·지방 활동의 장을 확대한다는 것이다.

정부 차원에서는 수상 직속의 'IT 전략본부'를 중심으로 신가치/신산업 창조를 목표로 U-Japan 전략을 수립하고 있으며, 작년 7월초에 특히 'U-Japan 전략 수립'에 관한 중간보고자료를 발표하였다. 이에 따르면 U-Japan 구상은 고령화 추세가 가속화되고 있는 상황을 배경으로 고령자, 신체장애자, 지적장애자 등을 포함하여 누구나가 건강하게 참가하는 사회를 구축하기 위해, ICT를 이용하여 '언제라도, 어디서나, 무엇이라도, 누구라도' 네트워크를 간단히 연결하는 유비쿼터스 사회(U-Japan)를 실현하여 활력 있는 일본을 구축함과 동시에 U-Japan으로의 원활한 이행으로 세계를 선도한다는 것이다.

첫째는, '초소형 칩 네트워크 프로젝트'이다. 의복, 서류, 유가증권, 브랜드 제품에 마이크로 칩을 내장시켜 100억개의 단말간 협조·제어가 가능한 네트워크 기술을 개발하는 것이다. 둘째, '무엇이든 My 단말 프로젝트'이다. 비접촉 카드를 통하여 순식간에 어떠한 단말이라도 마치 자신의 단말처럼 사용하게 하는 기술 개발을 목표로 하며, 개발 목표 중에는 현재 속도의 1만분의 1이하 속도로 빠른 실시간 응답과 사용자 인증을 포함한다. 마지막으로, '어디서든 네트워크 프로젝트'이다. 언제 어디라도 네트워크에 연결되어 사무실과 동일한 통신 서비스를 실현하게 해주는 환경을 구현한다.

국가 정책적 차원에서 'U-Japan' 전략의 목표인 세계 최첨단 IT 국가

의 구체적인 모습으로 유비쿼터스 네트워크 사회를 설정해 중요성을 부여하고 있다. 이러한 계획은 자국이 국제 경쟁력을 확보하고 있는 모바일, 광섬유망, 가전, IPv6 그리고 부품·재료·정밀가공기술 등과 연계시켜, 5년 이내에 세계 최첨단의 IT국가가 될 것을 목표로 하고 있으며, 기업 및 연구소를 통해 네트워크, 디바이스, 보안 및 인증기술, 소프트웨어 및 응용기술 등에 대규모 투자를 아끼지 않고 있다. 미국의 강점 분야인 컴퓨터, 소프트웨어 등 핵심기술도 중요하지만, 일본의 입장에서는 마이크로 센서 기술을 이용하여 사람과 사물간의 통신, 사물과 사물간의 통신을 지원하는 주변기술이 전략적으로 중요하다는 인식하에 추진하고 있는 것으로 보인다.

2002년 6월에는 총무성 주관으로 NTT·NHK·소니·샤프·도시바·마쓰시다전기·미쓰비시전기 등 일본의 대표적인 30여개 민간기업과 도쿄대학 등이 참여하는 '유비쿼터스 네트워크 포럼'을 출범시켰으며, 산·학·관의 연대하에 직장이나 가정은 물론 이동 중에도 휴대폰이나 PDA 등으로 인터넷 검색과 디지털 방송에 접속가능 한 유비쿼터스 네트워크 개발에 본격 착수하여 일본이 세계 최고수준을 자랑하는 이동통신에 네트워크 기술을 접목시키려 하고 있다.

일본은 '유비쿼터스 네트워크 사회'로의 단계적 발전을 계획하고 있으며, 최우선 과제로 2008년까지 휴대폰과 PDA 등 손안에 들고 다니는 휴대단말기로 하루 24시간 언제라도 인터넷에 접속해 각종 멀티미디어 정보를 검색하는 것은 물론 비디오, 컴퓨터 소프트웨어 전송, 지상파 위성방송까지 청취할 수 있는 단말기와 네트워크 시스템 개발을 추진하고 있다. 또한 이동 중에도 멀티미디어 통신을 주고받을 수 있도록 하기 위해서는 기존 시스템 보다 약 1,000배 빠른 네트워크 기술 개발이 필수 요소라고 인식하고 이에 주력할 계획이다.

노무라연구소는 일본 국내총생산에서 u네트워크 효과가 58조엔이며 이 중 주요 4대 분야인 고부가가치 부품생산액 18조엔, 통신업계 신종 서비스 18조엔, 중계 및 인티그레이션 및 서비스 0.6조엔, 설비기기의 리스 및 렌털 생산액 13조엔 이라고 분석했다. 일본은 e재팬 전략Ⅱ을 통해 새로운 IT사회 기반정비 분야에서 차세대 정보통신 기반 정비, 언제 어디서나 무엇이든 연결 가능한 유비쿼터스 네트워크 형성을 목표로 내걸고 있다. 이번 전략은 유비쿼터스를 국가전략으로 채택한 세계 첫 사례로 꼽히고 있다. 유비쿼터스에 연관 있는 부처들은 모두 각종 연구회를 설치하고, 주파 관리권을 가진 총무성과 산업전반을 관장하는 경제 산업성이 유비쿼터스를 놓고 주도권 경쟁을 벌일 정도로 일본 정부는 신속하게 움직이고 있다. 특히, 총무성은 2001년에 유비쿼터스 네트워크 기술의 장래 전망에 관한 조사 연구회를 구성, 관련 기술개발에 관한 국내외 연구동향 등을 조사/분석함과 동시에 몇 가지 중점 프로젝트를 제안했다. 최근에는 민간, 대학 전문가 그리고 통신사업자와 가전메이커 등을 중심으로 유비쿼터스 네트워크 포럼을 구성하고 유비쿼터스 국가기반 구축에 정부와 산학연이 공동 대응하는 방안을 모색 중이다.

일본 기업들이 최근 잇따라 유비쿼터스 조직을 신설하는 데서 그 힘을 느낄  수 있다. 지난해 3월 히타치가 유비쿼터스 추진센터를 설립한 것을 비롯하여, 소니의 유비쿼터스 기술연구소, 리코의 유비쿼터스 솔루션 연구소, 후지쯔의 유비쿼터스 사업추진부, 미쓰비시전기의 유비쿼터스 영상기술부, 후지제록스의 유비쿼터스 미디어사업 개발부 등 이미 표면 위로 급부상한 상태다. 이제 일본 대표기업들이 유비쿼터스 전략을 전면에 내세우고 있다.

*c.* 유럽연합(EU)

유럽은 유럽연합이 중심이 되어 2001년에 시작된 정보화사회기술계획(IST)의 일환으로 미래기술계획(FET)이 자금을 지원하는 '사라지는 컴퓨팅 계획(Disappearing Computing Initiative)을 중심으로 16개의 프로젝트 수행을 통하여 유비쿼터스 컴퓨팅 혁명에 대한 대응전략을 모색하고 있다. 이름이 의미하는 바와 같이 사람의 눈에 보이지 않는 수많은 소형 컴퓨터들을 사람들의 생활공간 곳곳에 내장시켜 인간에게 편리함을 제공하려는 것이다.

이 사업을 통하여 미래의 컴퓨터 응용에 대한 개념과 기술을 도출하려는데 목적이 있으며, 유럽의 유비쿼터스화에 대한 대응연구는 첫째, 일상 사물에 스마트한 기능을 내장하는 도구나 방법을 개발하는 것, 둘째, 일상 사물들 간의 상호작용에 대한 새로운 기능과 용도를 연구하는 것, 그리고 세 번째로는 인간 생활이 스마트 사물 환경에 밀착되고 조화롭게 생활할 수 있는지를 연구하는 것이라고 요약할 수 있다.

'사라지는 컴퓨팅 계획'을 중심으로 유비쿼터스 컴퓨팅 혁명에 대한 대응전략을 모색하고 있는 유럽은 우리가 흔히 사용하는 일상 사물에 센서 · 구동기 · 프로세서 등을 탑재해 사물 고유의 기능에 정보처리 및 정보교환기능이 증진된 정보 인공물(Information artifacts)을 개발하려고 한다. 이들 정보 인공물 상호간의 지능적이고 자율적인 감지와 무선통신을 통해 새로운 가능성과 가치를 창출하고, 궁극적으로는 인간의 일상 활동을 지원하고 향상시킬 수 있는 환경을 구축하는 것을 최종 목표로 프로젝트를 추진하고 있다. 유럽에서 추진하고 있는 거의 대부분의 프로젝트는 다국적, 전문연구 기관의 공동형태로 추진되고 있다.

유럽의 유비쿼터스 컴퓨팅 프로젝트는 정보기술을 일상사물과 환경속에 통합하여 인간의 생활을 지원하고 개선하는데 주력하고 있다. 이를 위하

여 일상 사물에 스마트한 기능을 내장하는 도구 및 방법의 개발, 일상 사물들 간의 상호작용에 대한 새로운 기능과 용도 연구, 인간생활이 스마트 사물 환경에 밀착되고 조화롭게 생활할 수 있는 방안 등이 연구되고 있다.

## *C.* 유비쿼터스의 응용

### *1.* 유비쿼터스 센서 네트워크의 특징

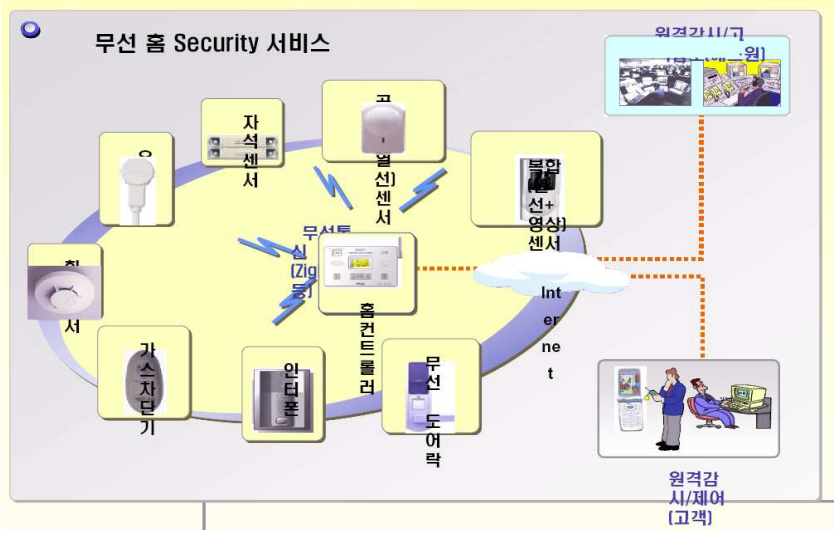
유비쿼터스는 그림 2-4와 같이 기존 망과 더불어 FTTH와 PLC 등의 유선망과 WiBRO 등의 무선망과 통합되면서 통신서비스 네트웍을 형성한다. 가입자망에서는 기존 통신에 스스로 판단하고 통신할 수 있는 마이크로프로세서가 내장된 노드들로 구성됨으로서 사용자가 언제 어디서든 원하는 정보를 실시간으로 얻을 수 있는 네트워크를 구성하는데 그 목적이 있다.[13-14]

유비쿼터스 센서 네트워크는 주변 환경의 물리적 현상을 감지하는 다수의 센서노드와 센서간의 무선 통신 및 노드들과 외부망을 연결하는 싱크로 구성된다.

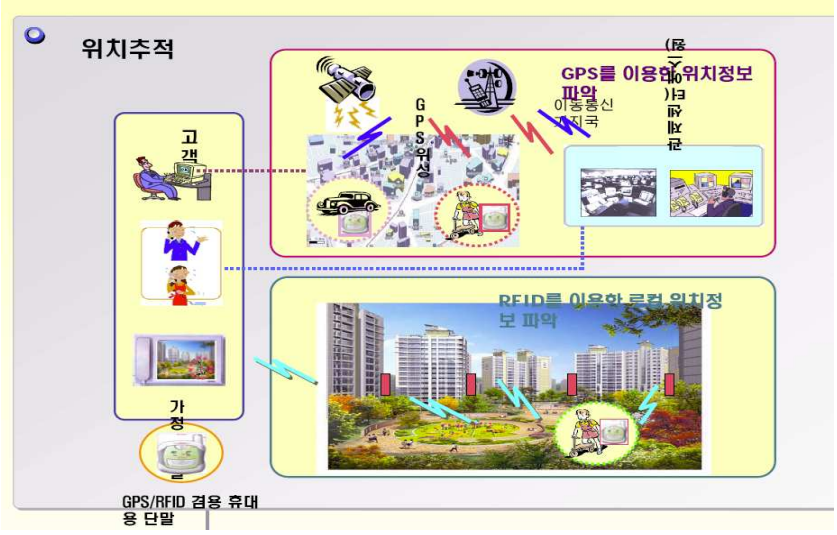
유비쿼터스 센서 네트워크는 동일 센서/이질 센서, 고정 센서/이동 센서 등 많은 수의 센서노드를 가지며 Global ID를 가지지 못한다. 또한 무선 통신방식을 이용하므로 물리적인 오류로 인하여 신뢰성이 저하되고 전파의 간섭으로 인한 메시지 손실이 있을 수 있고 CPU, Memory, Battery, Communication Bandwidth 등 원인으로 커퓨팅 자원이 제한되어 있으며 센서에 의해 감지된 값에 따라 어플리케이션 수준의 즉각적인 대처가 요구 된다.

Fig. 2-4 Ubiquitous's Concept

그림 2-5는 보안 및 위치추적 시스템의 예를 나타낸 것으로서 보안시스템의 경우 가내의 센서들을 통해 정보를 수집하고, 유무선 네트워크를 통해 개인이 집안의 상황을 모니터링하고, 각기 센서 노드들을 독립적으로 실행할 수 있도록 시스템이 구성된다. 위치추적 시스템의 경우는 GPS 위성위치 시스템을 사용하고 RFID와 연동시킴으로서 어린이나 노약자의 위치추적 및 신원확인이 가능하다. 앞으로서 사회 간접망 설치시 필수적으로 탑재되어 유비쿼터스 세상을 주도할 것으로 예측된다.[9-10]




(a) 무선 홈 시큐리티 서비스




(b) 위치 추적 서비스

Fig. 2-5 An Application Example of Ubiquitous

## 2. Location Based Services

위치기반서비스(LBS) 기술은 유·무선 통신망 특히, 이동 통신망 및 무선인터넷을 통해 모바일 환경에서 위치정보를 이용하여 다양한 부가적인 서비스를 제공하는 통신 인프라를 활용한 서비스 기술로서, 이 기술은 일반 대중을 위한 서비스뿐만 아니라 응급구조 시스템, 재난관리시스템 등 공공 및 개인의 안전과 효율적인 자원의 관리를 위한 국가 정보 인프라로서 정보통신 기술 활용 등 사회적인 요구 증가에 부합되고 있는 차세대 서비스이다.[15]

위치기반 서비스, 즉 LBS는 기존의 유·무선 데이터 서비스 혹은 인터넷 서비스와 구별되는 새로운 개념의 서비스임에는 분명하지만 서비스의 사용자가 휴대형태의 단말을 사용하는 것이 보편화 되고 있는 상황에서 필연적으로 탄생되고 발전될 수밖에 없는 서비스 영역이다. 따라서 LBS를 일명 MLS, 즉 이동 위치정보 서비스라 부르기도 한다. 위에서 정의하는 바와 같이 위치기반 서비스는 아래 표 2-2와 같이 위치서비스와 이를 활용하는 응용서비스로 간단히 요약 된다.

Table 2-2 Relation of LCS and LBS

| 형 태 | Location Services (LCS) | Location-Based Services (LBS) |
|---|---|---|
| 제공 서비스 | 휴대 단말의 지리적인 좌표 단위의 위치 정보 | 휴대 단말의 위치 정보를 이용한 부가가치 서비스 |
| 서비스 제공자 | 이동통신망 사업자<br>이동통신망 임대사업자<br>위치정보 제공 사업자 | 응용서비스 제공자 (위치기반 서비스 제공자 또는 위치 서비스 사용자 : 무선 서비스 제공자, 2차 ASP 사업자, 별정 이동통신 사업자, 이동통신망 임대 사업자) |
| 서비스 이용자 | 위치기반 서비스 제공자 | 이동통신 사용자, 무선 서비스 제공자, 정부 기관, 군, 소방, 보안업체 |

즉 LBS는 사용자의 위치정보, 단말 및 각종 데이터를 수집하는 장치의 위치 정보, 이들 장치에서 입력되는 각종 음성 · 영상 · 감지 데이터, 사용자 · 단말 · 데이터 · 서비스제공위치를 나타내는 지도, 이들 정보를 가공하여 처리하고 서비스를 제공하는 플랫폼 등 몇가지 중요한 기술 요소들로 구성 된다. 이러한 요소들을 활용하여 사용자, 특히 이동형 단말을 휴대한 사용자에게 사용자의 위치나, 요구에 적합하고 최적화된 서비스를 간편한 방법에 의하여 실시간으로 제공함으로서 사용자의 위치정보 활용 요구를 만족시킬 수 있는 서비스를 의미한다.

***a.*** 연관 기술과의 관계

최근에 대두되고 있는 유비쿼터스 관련 기술 및 서비스는 이 위치기반 서비스와 밀접한 관계가 있으며 유비쿼터스를 구현하기 위한 기술은 통신 인프라 및 각종 데이터 수집을 위한 센서기술 등 현재 정보, 통신 관련 기술뿐만 아니라 정밀기계 관련된 기술까지를 총 망라한 넓은 영역을 포함하게 될 것이며 이런 하부 기반 기술을 접목하여 궁극적으로 사용자에게 제공하는 서비스는 기존의 정보통신 인프라를 이용하여 제공되던 서비스와 크게 다를 바가 없지만 가장 두드러지는 차이점은 위치기반 서비스에서 가장 중요한 요소인 위치정보가 기존의 서비스에 더 추가됨으로서 서비스의 활용도가 다양해지고 정보의 가치가 더 커지게 된다. 즉 유비쿼터스 환경에서의 서비스는 위치정보를 기반으로 하는 서비스 형태를 띠게 될 것이다.

또한 위치기반 서비스는 거의 모든 경우에 이동형 단말을 이용하여 제공될 것이며 또한 이동망의 국제 로밍을 고려할 때 서비스의 글로벌라이징이 중요한 요소로 부각되고 있다.

특히 이동통신 인프라가 전 세계를 동작영역으로 하는 국제적인 로밍이

보편화 되는 경우에 위치기반서비스는 그 활용도가 극대화될 것이며 전 세계를 다양한 이동통신 인프라에서 현재의 음성통화 서비스와 같이 위치 기반 서비스도 상호 연동되어 특별한 제약 없이 이용가능 하여야 한다. 이런 요구 사항을 충족시키기 위하여 각종 정보의 표준화와 시스템 인터페이스의 표준화 정보접근 방법의 표준화 서비스 형태의 표준화, 특히 GIS 정보의 표준화가 무엇보다도 중요한 이슈로 등장하게 되었다. 그림 2-6은 LBS와 연관기술의 관계를 나타내고 있다.

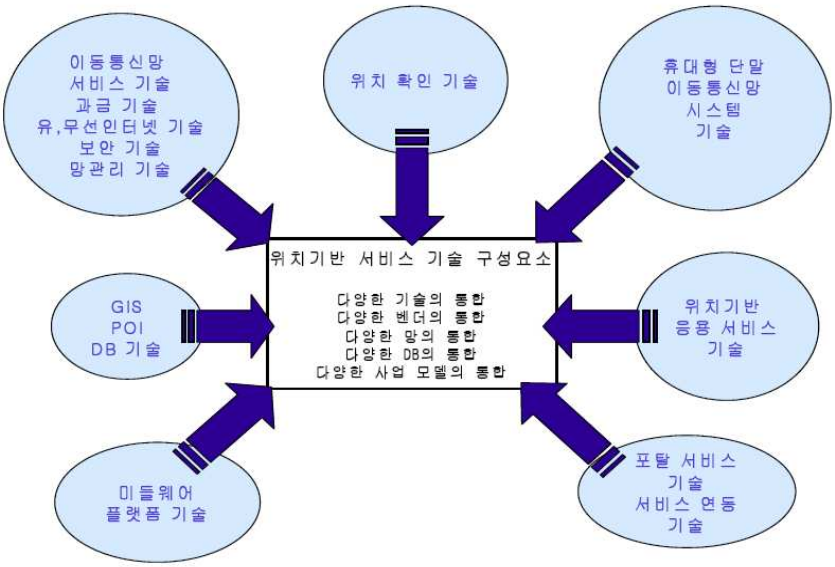



Fig. 2-6 Relation of LBS Relation Technic

***b.*** 위치 측위 기술

위치 측위 기술은 이동통신망 기반 방식과 단말 기반 방식으로 분류할 수 있으며 두 방식을 혼합한 하이브리드 형태의 방식이 있다.[15]

이동통신망 기반 방식은 망이 위치를 측위 할 수 있도록 정보를 제공하고 실제적인 위치는 망에 위치한 알고리즘에 의하여 계산 및 산출되는 방식으로 기지국에 부가적인 장치와 기능을 구현하여야 하기 때문에 초기 비용이 높다. 간단하게는 기지국에서 수신되는 신호의 세기를 가지고 알아낼 수 있는 방법이다.

단말 기반 방식은 각각의 휴대 단말에 GPS 신호를 수신하는 기능이 있어야 하며 실제적인 위치의 계산도 단말 차원에서 수행된다. 위성항법 장치라고 흔히 알려 져 있으며 위치 정보는 GPS 수신기로 3개 이상의 위성으로부터 정확한 시간과 거리를 측정하여 3개의 각각 다른 거리를 삼각 방법에 의하여 현 위치를 정확히 계산할 수 있다. 나침반과 달리 위성항법시스템은 위도, 경도, 고도의 위치뿐만 아니라 3차원의 속도정보와 함께 정확한 시간까지 얻을 수 있다.

하이브리드 방식은 GPS 신호가 고층 빌딩이나 지하에서 수신되지 않는 문제점 때문에 두가지 방식을 혼용하는 형태의 위치 측위 기술이 소개되고 있으며 향후에 가장 보편적인 방식으로 사용될 것이다.

향후에는 무선통신 기술의 발전에 따라 유선, 무선, 방송이 통합되고 아주 근거리 무선통신 기술, 즉 Blue-tooth, 무선 LAN, UWB 기술을 통합하는 위치 측위기술과 현재 사용하고 있는 이동통신망 기술들이 통합되는 4G, 그 이후의 유비쿼터스 기술까지를 고려할 때 다양한 형태의 위치 측위 기술이 소개될 것으로 예상된다.

*c.* GIS(Geographic information system) 기술

지리공간 데이터를 분석·가공하여 교통·통신 등과 같은 지형 관련 분야에 활용할 수 있는 시스템으로 과거 인쇄물 형태로 이용하던 지도 및 지리정보를 컴퓨터를 이용해 작성·관리하고, 여기서 얻은 지리정보를 기초로 데이터를 수집·분석·가공하여 지형과 관련되는 모든 분야에 적용하기 위해 설계된 종합 정보 시스템을 말한다. 그림 2-7은 위치기반서비스 미들웨어 플랫폼 구조를 나타내고 있다.

지리정보시스템이 갖추어지면 다양한 공간 분석이 가능하고, 그래픽 정보나 관련 데이터베이스 등 각종 지형정보를 상세히 알 수 있을 뿐 아니라 처리도구와 조작도구를 이용해 방대한 공간자료를 효율적으로 관리할 수 있고, 특수한 사용자를 위한 전용 매크로 언어를 이용할 수도 있다.

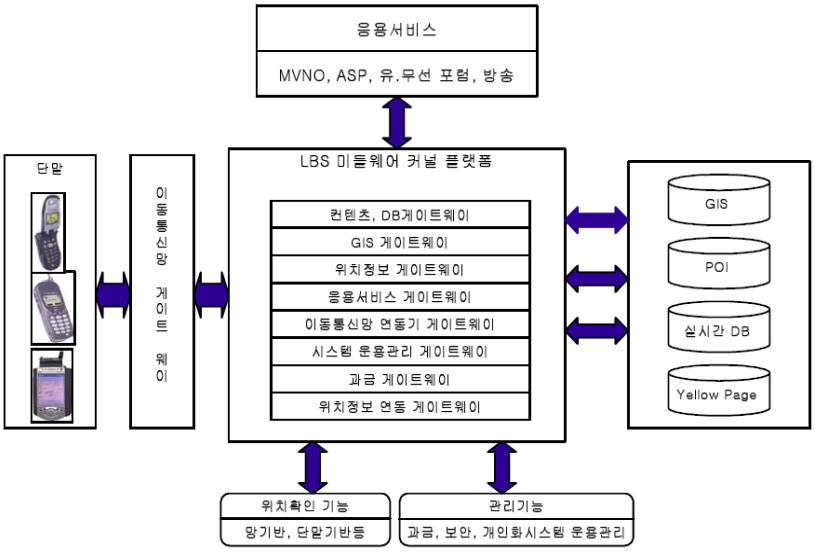



Fig. 2-7 Middle Ware Platform Structure of LBS

최근에 증대되는 위치기반 서비스에 연동되어 사용할 될 수 있도록 다양한 형태의 지도가 실생활에서 이용될 수 있도록 전자화 되고 모바일 환경에서 사용할 수 있도록 모바일 GIS기술, 지오코딩 기술 등의 연구가 추진되고 있으며 글로벌 연동이 되기 위한 표준화가 중요시되고 있다. 또한 지도정보의 실시간성을 위하여 다양한 기술의 D접목이 시도되고 있으며 지도 전문업체 간의 치열한 경쟁이 예상되고 있다.

3G 환경이 되면서 특히 멀티 벤더의 망 환경에서도 관계없이 서버간 통신이 가능하여 연속성 있는 서비스가 제공되어야 한다. 이를 위하여 중요한 사항은 산업체 표준 등을 수용할 수 있어야 하며 국내, 국제간 또는 시스템 간에 원활한 로밍이 보장 될 수 있도록 개방형태의 표준을 채택하여야 한다. 위치기반 서비스 미들웨어 플랫폼 기술에는 아래 콘텐츠 DB 연동 기술과 개인 정보보호 및 인증기술 다자간 과금 처리 기술에 대한 고려가 있어야 한다.

***d.*** 각종 콘텐츠 DB 연동 기술

다양한 DB를 상호 연동하여야 하고 이를 서버와 클라이언트 사이에 다양한 단말상으로 전달하여 똑같은 의미로 내용을 전달하여야 하므로 이를 실현하기 위하여 고려되어야 할 DB의 구조나 이 DB 데이터를 교환하는 데이터 교환 프로토콜 등이 주된 기술내용이다.

***e.*** 위치기반 서비스 관련 보안 및 개인정보보호 기술

위치기반 푸시 광고 같은 경우에는 수익창출이 기대되는 사업모델임에는 분명하지만 사용자가 선택권이 있는 풀 서비스와는 다르게 사용자에 따라서는 적절하지 못한 다수의 패킷, 즉 스팸을 수신하게 될 수 있다. 문자, 음성, 비디오 등 다양한 미디어에 의한 광고가 실행되면 더욱 심각

한 반향을 불러 일으켜 서비스 확산 자체에 걸림돌이 될 수 있다. 이 경우에는 사용자에게 선택적으로 수신을 허용할 수 있는 기능을 제공하여 주는 방법밖에는 대안이 없다. 이와 관련된 기관들로는 미국의 FTC, CTIA, FCC 등을 들 수 있다. 또한 무선 광고 협회가 이와 관련된 지침을 제시하고 있다. 유럽에는 ETSI, EU가 해당 법제를 담당하고 있다. 모바일 커머스(M-commerce) 혹은 위치기반 커머스(L-com- merce)의 경우에 보안은 상당히 중요한 이슈이다. 온라인 결재 기능이 무선망을 경유하여 제공되어야 하는 취약성을 안고 있기 때문이다. 따라서 데이터의 암호화에 초점이 맞추어 지고 있으며 다른 온라인 비즈니스와 같은 측면에서 고려되는 것 이외에 무선망을 사용한다는 측면에서 별도의 요구사항들이 있기 마련이다. 무선망상에서 이를 제공하기 위하여 비밀 보장성, 신뢰성, 인증, 지급보장 요구사항이 있다. 그림 2-8에 USN 서버의 가시화 구조의 일례를 나타내었다.

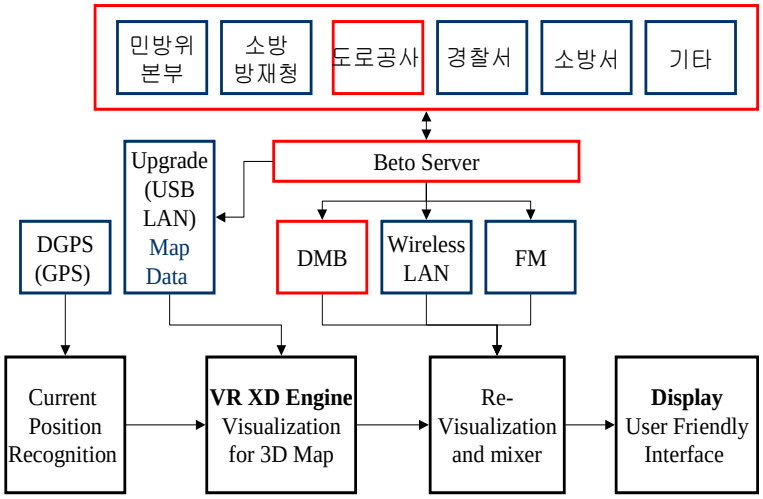



Fig. 2-8 Visualization Structure of USN Server

## 3. *Home Automation*

최근 인텔리전트 아파트의 개발로 유비쿼터스 개념이 채용된 아파트가 많이 등장하고 있는데 여기서는 삼성의 래미안의 모델을 주로 언급한다. 그림 2-9에서 보듯이 향후 추진방향은 친환경 생태단지를 추구하는 고품위의 아파트가 주로 그 대세를 이룰 것으로 예측된다. 래미안의 홈네트워크 서비스명인 홈비타에는 데이터베이스 서버, LMS(Local Management System) 서버, PMS(Public Management System) 서버, 포털 서버, 모니터링 서버 등 5종의 서버 프로그램이 있다. 데이터베이스 서버는 홈비타 운영에 관계된 모든 자료를 관리하고 LMS 서버는 댁내의 기기들을 제어하며, PMS 서버는 입주자의 출입을 감시한다. 포털 서버는 입주자에게 웹 콘텐츠를 제공하고 모니터링 서버는 홈비타의 단위별 상태를 감시한다.[13-14]

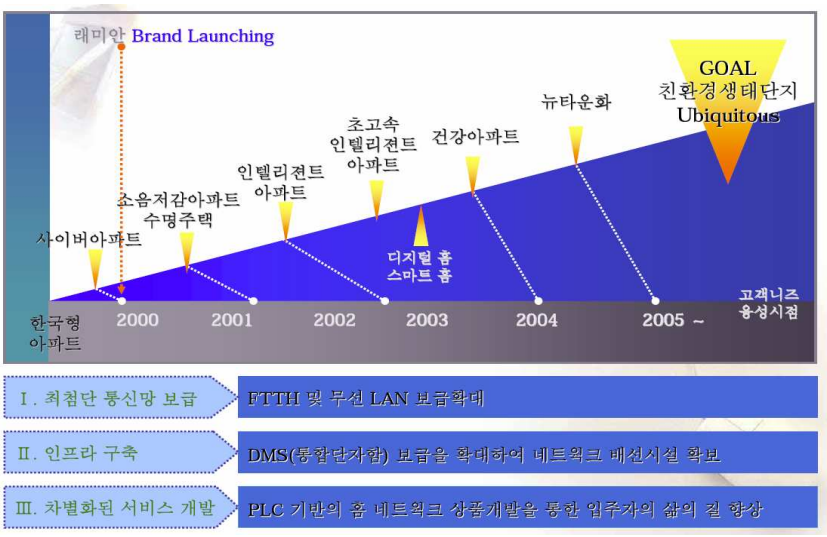
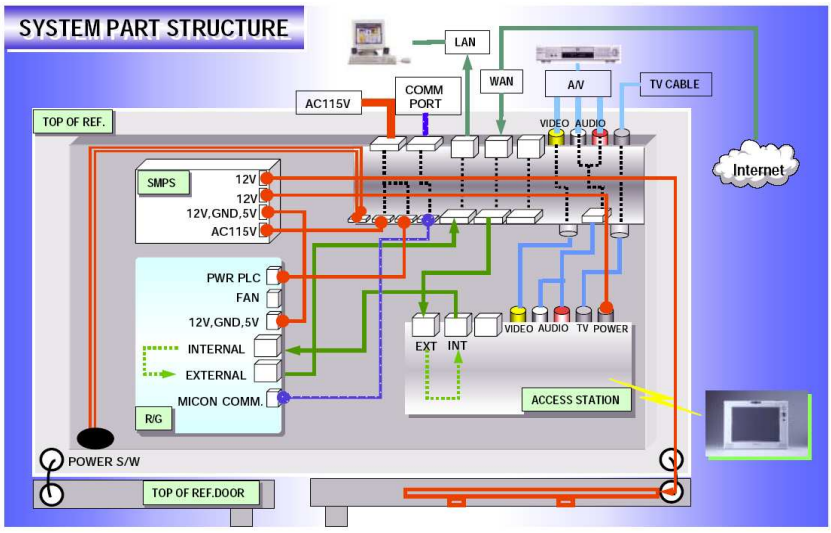
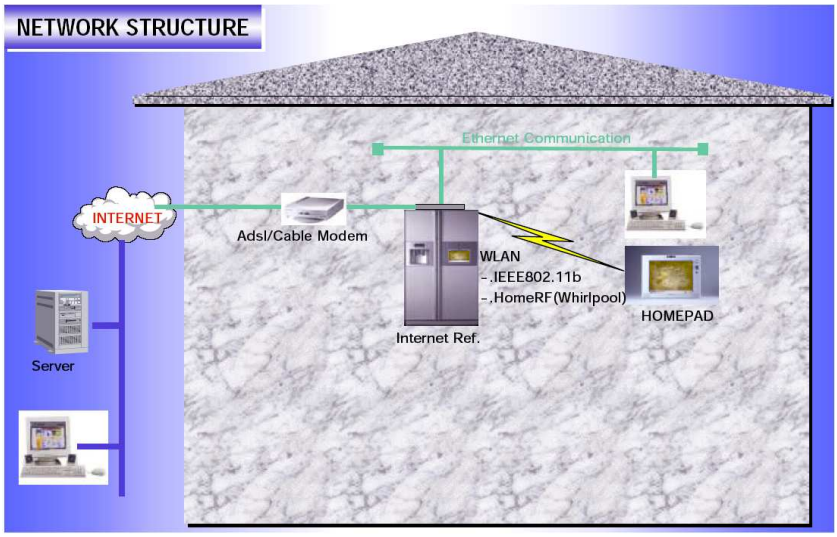



Fig. 2-9 Develop Direction of HA




(a) System Part Structure




(b) Network Part Structure

Fig. 2-10 Composition of Home VITA

그림 2-10과 같은 홈비타에서는 원격 제어를 위해 외부에서 전화선을 통해 댁내 홈서버로 진입하게 되고, 사용자는 댁내 서버와 통신을 통해 각 주변기기들을 통제하게 된다. 이를 위해 무선이동식컴퓨터, 특정위치에 상관없이 인터넷접속(반경30M), MPEG2 방식의무선TV 기능, 홈오토메이션과 연계된 출입통제기능 등을 수반하도록 설계된다. 그림 2-11은 웹 접속 방식을 나타내고 그림 2-12는 주택내 구성을 나타내고 있다.

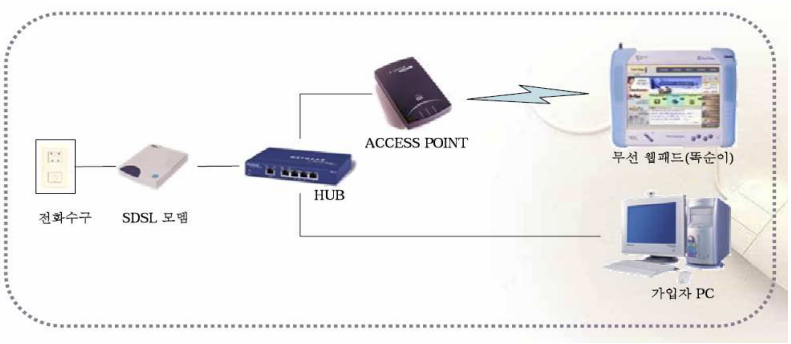



Fig. 2-11 WEB Connection Method

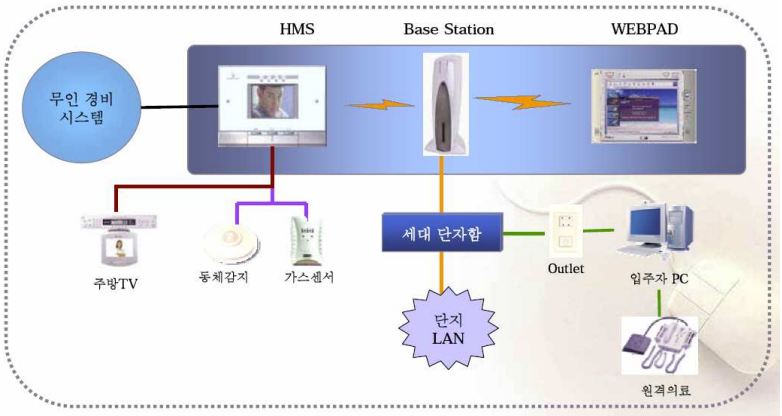



Fig. 2-12 Composition of Home

# *III.* 유선기반의 센서네트워크

## *A. Power Line Communication*

우리주위에 광범하게 깔려있는 저압 전력선을 이용하여 데이터를 전송하는 방법이 바로 전력선 모뎀(PLM: power line modem)기술로서 다수가 결합하면 네트워크를 구성할 수 있다. 이 전력선 모뎀은 각종 전력측정치를 전송하고 원거리 설비의 제어나 감시를 가능하도록 하여 빌딩자동화, 보안, 무대조명, 가로등 조명제어-보수, 전광판 제어, 인터넷 연결 등에 사용된다. PC와의 접속은 RS-232 방식을 이용한다. 그림 3-1은 PLC의 구조를 나타내고 그림 3-2는 전력선 모뎀을 이용하여 장거리 자동 검침을 시도 할 경우의 결선을 나타내고 있으며 그림 3-3은 PLC 모뎀 SE-PLM을 나타내고 있다.

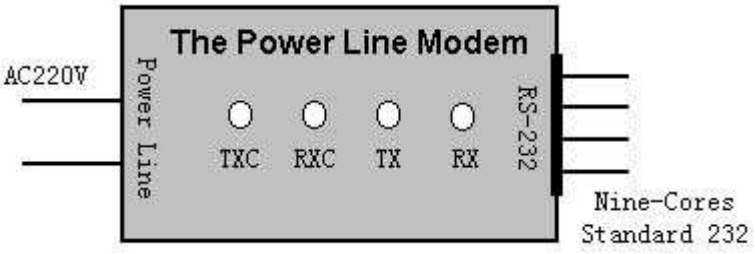



Fig. 3-1 Structural Diagram of PLC

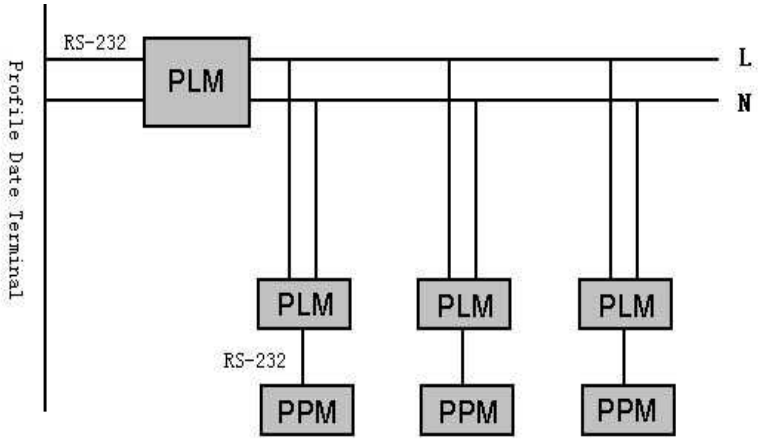



Fig. 3-2 PLC Network Composition

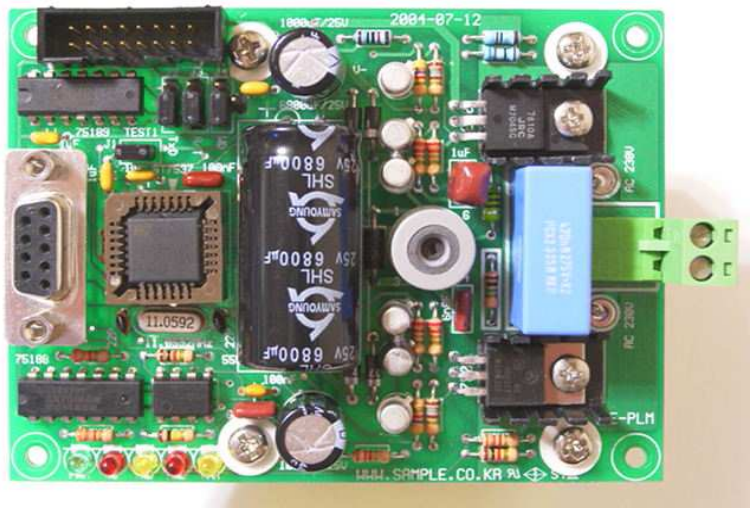



Fig. 3-3 PLC Modem

네트워크 구성은 그림 3-3의 PLC 모뎀 SE-PLM을 이용하여 구축한다. SE-PLM은 DC 10[V]로 동작하고 전원 트랜스는 15V 12VA 용량을 사용하였다. SE-PLM은 사용자 컨트롤 보드에도 전력을 공급할 수 있도록 여유 있는 전원 트랜스를 사용한다. 트랜스에서 나온 AC[15V]를 정류하여 DC +15V와 DC -15V를 만든다. -15V 발생회로는 배전압 회로를 응용한 것으로 약 50mA 의 전류 공급이 가능하다. 대부분의 경우 - 전압은 많은 전류가 필요치 않으므로 배전압 회로를 응용한 - 전압 발생 회로를 사용하고, 단권 트랜스에서도 양 전원을 얻는 방법을 사용한다. SE-PLM에서 -15V는 레벨 컨버터 SN75188의 -Vcc에 사용된다. 사용 가능한 주전원 전압은 220~230V 이고 주파수는 50~ 60Hz 이므로 유럽지역의 230V/50Hz 지역에서도 동작 가능하도록 트랜스를 설계하였다.

ST7537은 워치독 타이머를 내장하고 있다. 1.5초 이내에 WD 핀을 HIgh->Low로 하여야 한다. 1.5이상 WD가 변화하지 않으면 SR7537은 리셋 되며 RSTO 펄스가 발생한다. SE-PLM을 마이크로프로세서가 직접 제어한다면 제어 회로는 워치독 타이머와 리셋펄스 발생을 이용하는 것이 좋다. 그러나 SE-PLM을 PC의 시리얼 포트에 연결하여 사용한다면 강제적으로 WD 신호를 공급해 주어야 한다. SE-PLM은 555를 이용한 발진회로가 있어 J4 에 의하여 사용할 수 있다. 윈도우 계열의 OS는 어려우나 다른 리얼타임 OS를 사용한다면 CN1의 4번 핀을 이용하여 WD에 펄스를 공급할 수 도 있다.

SE-PLM에는 9핀 DB-9S(암) 커넥터가 부착되어 있다. PC의 COM1 또는 COM2 포트와 1:1로 직접 결선한다. 레벨 컨버터는 신뢰성이 입증된 TI사의 SN75188과 SN75189를 사용하였다. 드라이버 전압을 충분히 하기 위하여 SN75188에 +15V 와 -15V를 공급한다. SN75188 에서 약간의 열이 발생할 것이다.

ST7537은 항상 연속적으로 송신모드가 되지 않도록 되어있다. 약 1.5초 이상을 송신모드로 할 수 없다. 전력선은 Half Duplex 로 데이터를 전송 하여야 하므로 하나의 모뎀이 무한정 캐리어를 발생하지 않도록 한 것이다. 그러나 장비/모뎀을 테스트하기 위해서는 강제로 송신모드로 고정해 둘 필요가 있다. 점퍼핀을 쇼트하면 송수신 모드는 ST7537의 20번 핀에서 지정한 상태를 유지 한다. 점퍼핀이 쇼트되어 있고, Rx/Tx이 Low 상태이면 이 모뎀은 계속해서 송신 모드가 되어 캐리어를 발생한다. 즉 점퍼핀이 오픈 High되어 있고 Rx/Tx 가 Low이면 1.5초 동안만 캐리어를 발생하고 강제로 수신모드로 전환된다.

## *B. Controller Area Network*

CAN은 원래 1980년대 후반에 자동차 산업을 위해 독일 회사 Robert Bosch GmbH에 의해 개발되었다. 그들이 CAN을 개발하게 된 동기는 고객들의 차량에 대한 기능상의 많은 요구들 때문인데, 이는 대부분이 전자식으로 작동되도록 하여 더욱 많은 배선을 필요로 하였다. 따라서 현대 자동차에서 꾸준히 증가하고 있는 거대한 배선 작업의 문제에 대한 해결책을 제공하기 위해서 모든 온-보드 주변장치들이 부착될 수 있는 하나의 단일 네트워크 버스로 설계되었는데 이것이 바로 CAN 이었다.[2-8]

특히, 자동차 분야에서 엔진 관리 시스템, 변속장치 제어, 계기 팩, 그리고 차체 전자 기술 같은 온-보드 전자 제어 장치(ECUs)들 간의 정보 교환에 사용되곤 한다. CAN은 최대 1[Mbit/sec]의 데이터 통신을 제공하여, 실시간 제어를 촉진한다. 아울러 오류 제한과 오류 탐색 기능들은 noise-critical 환경들에서의 신뢰성을 더욱 높여 주었다.

CAN은 2가닥의 꼬임선으로 연결되어 반이중 통신방식으로 짧은 메시지를 사용하는 고속 응용 시스템에 적당하다. 더불어 외부의 요인 등에 강인성을 가져 통신 에러율을 최소화하여 높은 신뢰성을 가지고 있다. CAN은 높은 전송속도와 단지 2가닥의 통신선로만 필요로 함으로 기존 전원선과 함께 4가락의 패키징 된 선로를 이용함으로 감지기와 수신기 등 CAN 기능이 내장된 시스템에 꼽기만 하면 됨으로 배선이 간단하고 쉬우므로 경제성이 높은 방식이다.

## *1. CAN의 구성*

1993년 CAN은 표준 ISO 11898 고속 애플리케이션용과 ISO 11519저속 애플리케이션용이 되었고, 이것은 다중-마스터 직렬 통신 버스로 기본 설계 사양은 고속, 높은 잡음 면역성, 그리고 오류-검출 기능들에 적합하게 되었다. 이러한 기능들의 결과로, CAN 버스는 또한 제조 산업과 항공우주 산업들에서도 폭넓게 사용되게 되었다. 근래에는 자동차 분야뿐만 아니라 산업 전 분야에 폭 넓게 적용되고 있다.

임베디드 시스템 또는 마이크로 컨트롤러에서 일반적으로 사용되는 CAN 버스는 이론적으로는 2032개의 서로 다른 디바이스를 하나의 네트워크상에 연결하여 통신을 수행할 수 있으나 CAN 송신기의 한계로 인하여 110개까지의 Node를 연결하여 사용할 수 있다.

통신 속도는 실시간 제어가 가능한 1[Mbps]의 고속 통신을 제공하며 더불어 자동차 환경과 같은 심각한 노이즈 환경에 적합하도록 에러 검출 및 에러 보정의 기능이 있다. 기본적인 시스템 구성은 그림 3-4와 같다.

CAN 버스는 마이크로컨트롤러들 간의 통신을 위해 설계되었으며 각각의 Processor/Controller는 우선순위는 존재하지만, 각기 Master가 될 수도 있고, Slave도 될 수 있다. 즉, 한 Processer/Controller가 보낸 데이터는 필요에 따라 여러 Processor/Controller가 공유할 수 있다. 최소의 Line to line의 통신인 시리얼 타입이며, ± line으로 구성되어 485통신과 동일하며, 이와 같은 방식의 장점은 외부 전자파나 노이즈가 ±Line에 동일하게 영향을 끼쳐도 ±Line의 위상차는 영향이 없기 때문에 외부 전자파나 노이즈에 강인한 특징을 갖고 있다.

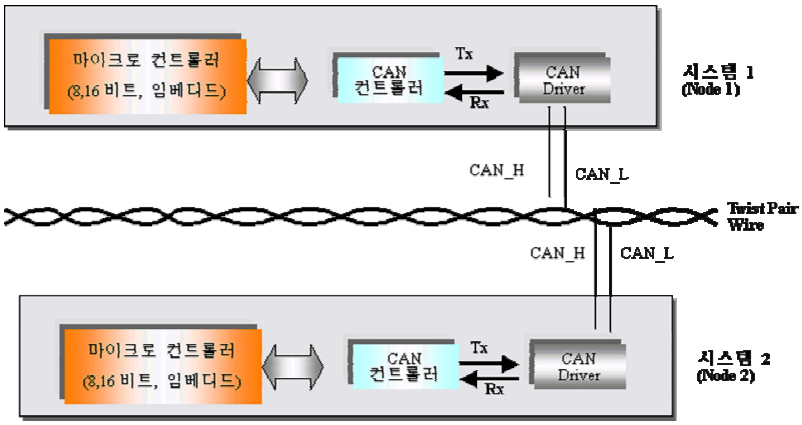



Fig. 3-4 Composition of CAN System

## 2. CAN의 동작원리

### a. 패킷

CAN은 다중통신망(Multi Master Network)이며 CSMA/CD+AMP (Carrier Sense Multiple Access/Collision Detection with Arbitration on Message Priority) 방식을 이용한다.[2-8]

먼저 CAN 노드에 메시지를 보내기 전에 CAN 버스라인이 사용 중인지를 파악한다. 또한 메시지 간 충돌 검출을 수행한다. 이러한 방식은 이더넷 통신 방식과 유사하다. 어떠한 노드로 부터 보내어진 데이터 메시지는 송신측이나 수신측의 주소를 포함하지 않는다.

대신에 각 노드의 데이터 메시지 항목에 CAN 네트워크상에서 각각의 노드를 식별할 수 있도록 각 노드마다 유일한 식별자 ID-11bit 또는 29bit를 가지고 있다. 그림 3-5는 CAN의 패킷 구조를 나타내고 있다.

네트워크상에 연결된 모든 노드는 네트워크상에 있는 메시지를 수신한 후 자신에게 필요한 메시지인지를 식별자를 통하여 평가한 후 자신이 필요로 하는 식별자의 메시지인 경우만 취하고 그렇지 않은 경우의 메시지는 무시한다.

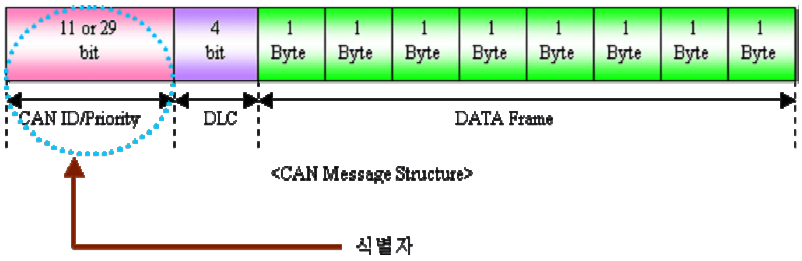



Fig. 3-5 Structure of CAN Packet

네트워크상에 흘러 다니는 여러 노드의 데이터들이 동시에 사용자가 필요로 하는 노드로 유입되는 경우에 식별자의 숫자를 비교하여 먼저 취할 메시지의 우선순위를 정한다. 식별자의 숫자가 낮은 경우가 우선순위가 가장 높다.

우선순위가 높은 메시지가 CAN 버스의 사용 권한을 보장 받으며, 이때 낮은 순위의 메시지는 자동적으로 다음 버스 사이클에 재전송을 수행한다. 이때 까지도 높은 우선순위를 가진 메시지가 완료되지 않은 상태이면 전송을 완료할 때까지 대기하고 있다.

각 CAN 메시지는 11비트의 식별자, 또는 29비트의 식별자를 가지면 CAN 메시지의 맨 처음 시작부분에 위치한다. 더불어 식별자는 메시지의 형태를 식별시켜 주는 역할과 메시지의 우선순위를 부여하는 역할을 한다.

***b.*** 프로토콜

CAN 통신 프로토콜은 CAN 버스에서 디바이스들 통신 사이로 데이터가 전달되는 방법을 명시한다. 이것은 ISO의 개방형 시스템 상호연결 모델(open system interconnection model)을 따르며, 이 모델은 통신 네트워크 표준인 일곱 계층으로 되어 있다. 이 OSI 모델은 두 개 네트워크 노드들 간의 층화된 통신 시스템을 기술하며, 이론상 각 계층은 로컬 모드에서는 오직 자신의 직접적인 위, 아래의 계층들과 통신할 수 있으며, 원격 모드에서는 동등한 계층과 통신할 수 있다. OSI 모델의 계층들은 다음 표 3-1에 나와 있다. 사실 CAN 프로토콜은 OSI 모델의 가장 낮은 두 개 층들로 설명될 수 있다. 데이터 링크 계층과 물리적 계층. 애플리케이션 계층 프로토콜들은 개별적인 CAN 사용자들에 의해 개발된 독점 구조, 또는 특정 산업 내에서 사용되는 신생 표준들 중의 하나가 될 수

있다. 프로세스-제어/제조 분야에서 사용되는 공통적인 애플리케이션 표준 층은 Device Net으로, 이것은 PLC의 네트워킹과 지능 센서들 그리고 액추에이터에 특히 적합하다. 자동차 산업에서 많은 제조업체들은 그들 자신의 독점적인 표준을 사용하고 있다.

Table 3-1 Composition of CAN Protocol

| 계층 | layer | 내 용 |
|---|---|---|
| 7 | Application Layer | 최상위층. 이것은 사용자가 네트워크와 상호 작용하는데 사용되는 소프트웨어를 설명하는 계층입니다. 예를 들어 Device Net이 해당됨. |
| 6 | Presentation Layer | 변환될 데이터의 구문을 서술합니다. - 예를 들어 서로 다른 수학 포맷을 사용하는 두 개 시스템들 간의 부동 소수점 수 변환 |
| 5 | Session Layer | 하위 계층들에 의해 처리된 패킷들보다 큰 데이터 순차들 처리에 관하여 서술. |
| 4 | Transport Layer | 두 개 통신 노드들 간의 데이터 전송 품질과 속성을 설명합니다. 재전송과 오류 복구 같은 문제 기술. |
| 3 | Network Layer | 다양한 데이터 링크를 거쳐 일련의 교환들이 어떻게 한 네트워크에서 임의의 두 개 노드들 간의 데이터를 전송할 수 있는지 설명합니다. 라우팅과 어드레싱 같은 문제들을 기술. |
| 2 | Data Link Layer | 특정 매체를 통해 전송된 데이터 비트의 배열과 조직을 서술합니다. 예를 들어, 이 계층은 checksum과 framing을 기술. |
| 1 | Physical Layer | 교환된 신호들의 해석, 전기적 특성들과 함께 통신 매체의 물리적 특징들도 서술. |

물리 계층에서, CAN은 광섬유 또는 꼬임-쌍 같은 다양한 종류의 매체를 사용하여 잠정적으로 통신할 수 있다. 꼬임-쌍 시그널링은 각각의 전선에서 서로 다른 전압들을 사용하여 실행되므로, 한 전선에서의 신호 전압은 또한 다른 전선에서 전송되지만, 반전된다. 수신기에서, 이 신호는 한 신호를 반전하고 두 개의 신호를 합해서 복원된다. 이것은 두 개 전선들에 대해 공통적인 것이므로, 이 방법은 버스 상에서 발견된 어떤 노이즈도 줄일 수 있다. 바로 이 과정에서 CAN은 자체의 잡음 면역(Noise immunity)과 결함 허용(Fault tolerance) 기능들을 유도한다. 두 개의 전선들은 CAN_H(CAN High) 와 CAN_L(CAN Low)로 불린다. 정지 상태에서 CAN_H와 CAN_L은 2.5V에 놓인다. 이것은 디지털 "1"로 표시되며 또한 "recessive" 비트로도 알려져 있다. 디지털 '0' 은 또한 "Dominant' 비트로도 알려져 있으며, CAN_L보다 큰 CAN_H에 의해 지시된다. 일반적으로 디지털 '0' 의 경우, 관련된 전압은 CAN_H = 3.5V 그리고 CAN_L = 1.5V이다.

CAN Higher level protocol은 현재의 하위 CAN 계층인 물리 계층과 데이터 링크 계층의 "위"에서 실행되는 프로토콜이다. 상위 단계 프로토콜은 CAN의 물리 계층과 데이터 링크 계층들을 개발된 응용 계층의 밑바탕으로 사용한다. 많은 시스템들은 독점적 응용 계층을 사용하지만, 많은 산업들에서, 이 방법은 비용에 비해 효과적이지 않다. 여러 단체들이 시스템 통합을 쉽게 하기 위해서 표준화된 개방형 애플리케이션 계층들을 개발하고 있다.

CAN 버스에서 두 개 노드들이 동시에 전송할 때, 한 메시지가 우선순위를 갖는 방법이 필요하다. 이것은 'Non-destructive bit-wise arbitration'을 사용하여 달성된다. CAN 버스의 각 노드는 유일한 식별자(ID)를 가지며, 그것은 11비트 또는 29비트 숫자이다. CAN은 이진 0

이 이진 1에 우선하도록 결정한다. 따라서 더 낮은 ID 번호와 더 높은 우선순위, 그러므로 식별자 0이 버스에서 최고의 우선순위를 갖는다. 이것을 알아보는 다른 방법으로 메시지 충돌 상황이 있는데, 다른 노드가 1을 보낼 때 0을 보내는 최초의 노드가 버스 제어를 획득하게 되고 성공적으로 자신의 메시지를 전송하는 것이다.

Basic CAN과 Full CAN이라 이름 붙여진, 두 개의 일반적인 CAN 접근법이 있다. 이것들은 들어오고 나가는 데이터가 처리되는 방식에서 서로 다르다. 간단히 말하면 Basic CAN은 CAN 메시지 전송과 수신을 처리하기 위해 호스트 CPU를 필요로 하며, 프레임 저장소를 다룬다. Full CAN은 메시지 전송, 메시지 수신 그리고 최대 16개 메시지들의 저장을 CAN 컨트롤러에 맡긴다. CPU는 정보가 필요할 때 CAN 컨트롤러에 문의한다. 이 방식에서, Full CAN은 CAN 처리의 책임을 CPU에서 배제시킨다. 다른 작업들을 처리할 수 있는 자유를 부여. Full CAN에서 CAN 컨트롤러는 인터럽트가 설정되어 있다면 CPU를 인터럽트 할 수 있다. 그리고 어떤 지정된 ID를 가진 메시지가 수신된다면 컨트롤러가 CPU를 인터럽트 하면서 "필터링 허용' 같은 작업들을 처리할 수 있다.

표준 CAN에서는 식별자들이 11비트 길이를 가지며 확장 CAN에서는 식별자들이 29비트 길이를 갖는다. CAN 프로토콜 버전 2.0 에서 명시된 것에 의하면, V2.0A 로 컴파일 하는 CAN 컨트롤러는 반드시 11-비트 식별자를 가져야만 한다. 반면 V2.0B 에서는 11비트 또는 29비트 아무것이나 될 수 있다. V2.0B active를 이용하면, CAN 컨트롤러는 표준 포맷과 확장 포맷 모두를 전송하고 수신할 수 있다. V2.0B passive를 이용하면, CAN 컨트롤러는 표준 프레임을 전송하고 수신하게 되며 확장 프레임은 오류 없이 무시하게 된다.

## 3. CAN의 메시지 구조

CAN 시스템에서 데이터는 메시지 프레임을 사용하여 송수신이 이루어진다. 메시지 프레임은 하나 또는 그 이상의 송신 노드로부터 데이터를 수신노드로 운반한다. CAN 프로토콜은 표준 CAN 2.0A와 확장 CAN 2.0B 두 가지 형태의 메시지 프레임을 지원한다.

본 시스템에 사용된 CAN 2.0B 확장모드는 기본적으로 데이터 프레임(data frame), 리모트 프레임(remote frame), 에러 프레임(error frame), 오버로드 프레임(overload frame)이라는 4개의 프로토콜 프레임 형태가 있다. 그림 3-6은 기본이 되는 데이터 프레임의 기본 형태로 SOF(start of frame), arvitration field, control field, data field, CRC(cyclic redundancy code) field, acknow- ledge field, end of field으로 구성된다.

버전 2.0A와 구분되기 위해 29 비트의 메시지 프레임 식별자를 갖는다. 기존에 사용 중인 2.0A와 호환되는 2.0B로 발전되었다. 2.0A와 차이점은 중재 필드가 두 개의 CAN 메시지 식별자로 구분되어 포함되며 첫 번째 기본 ID는 11 비트 길이로 2.0A와 호환되게 하며 두 번째 필드 확장 ID는 18 비트 길이로 ID는 총 29 비트로 구성된다. 두 개의 ID 필드 사이에 ID 확장자 (IDE: identifier extension)가 있어 두 개의 ID 필드를 구분한다. SRR(substitute remote request) 비트는 중재 필드에 속해 있으며 표준 데이터 프레임과 확장 데이터 프레임을 중재해야 하는 경우에 대비하기 위해 항상 "1"의 값을 전송한다. 만약 표준 데이터 프레임과 확장 데이터 프레임이 같은 기본 ID 11 비트를 가지고 있으면 표준 데이터 프레임이 우선순위를 갖는다. 그리고 2.0B 메시지 프레임에 있는 다른 필드들은 표준 메시지 포맷으로 식별된다.

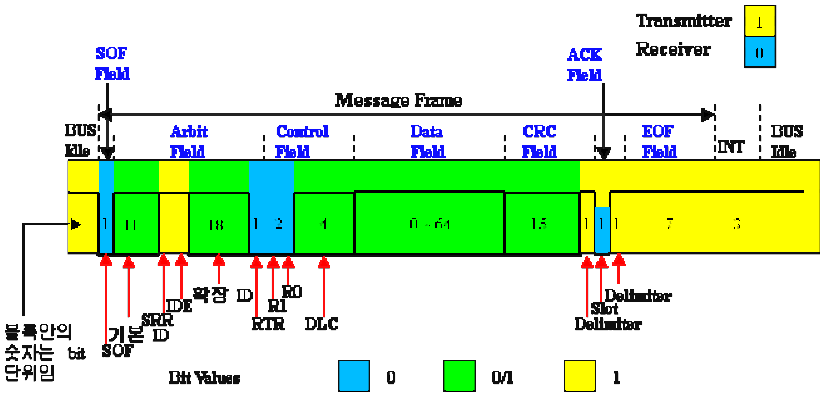



Fig. 3-6 Composition of CAN Protocol

# *IV.* 적외선 온도센서

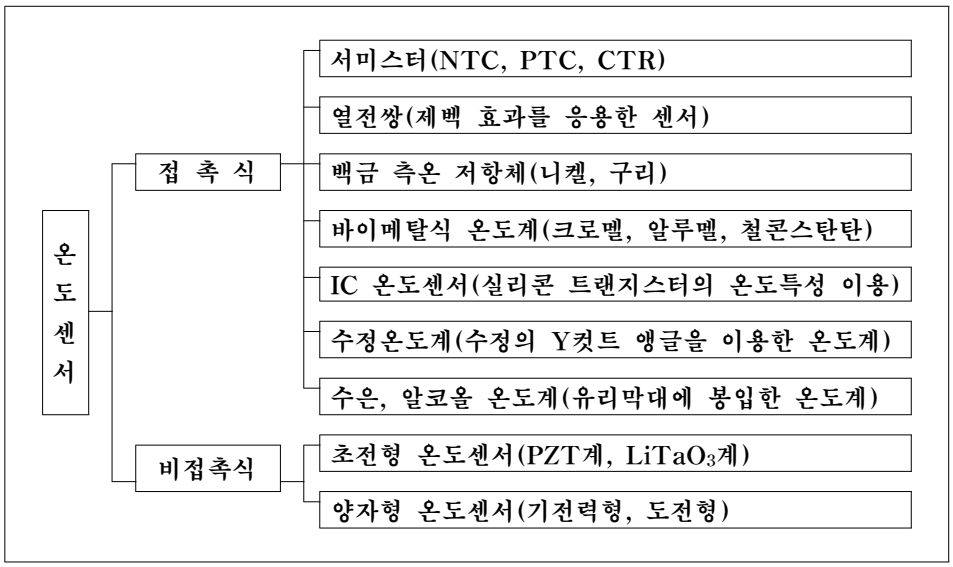
## *A.* 온도센서의 개요

지금부터 약 380년 전 이탈리아의 물리학자이며 천문학자인 갈릴레이에 의해 발명된 온도계는 여러 가지 기술 진보를 거듭하여 오늘날 가장 대중적인 계측기로 사용되고 있다. 현대 산업사회에서 온도센서에 대한 중요성은 점차 커지고 있으나 센서의 오류도 많고 그 사용범위도 광범위하여 이에 대한 이해가 부족한 상태이다. 온도센서(temperature sensor)는 열이나 온도를 전류나 전압으로 변환하여 검지하는 센서이며 다른 센서에 비해 종류가 많다.

측정대상은 기체, 액체, 고체, 플라즈마, 생체 등 다양하며 공간 역시 미생물에서 지구 및 천체에 이르기까지 광대하다. 따라서 온도센서의 종류는 매우 광범위하지만 크게 나누면 측정하고자하는 피 측정물에 센서를 직접 접촉시켜 측정하는 접촉형과 피 측정물에서 방사되는 적외선을 원격 관측하는 비접촉형으로 나눌 수 있다. 표 4-1에 온도센서의 종류를 나타내었다.

산업전반에 걸쳐 온도의 정확한 측정과 관리는 매우 중요하다. 온도를 검출하는 것은 온도 센서로서, 이것은 공업, 의학, 우주, 자원탐사, 해양개발, 방재, 공해 감시등 산업용에서부터 민생용까지 그 응용범위가 매우 넓고 갈수록 확대되고 있다.

Table 4-1 Description of Temperature Sensors




이상적인 온도센서는 측정범위가 넓고 정확하며 소형으로 값이 싸야 하지만 이러한 센서는 현실적으로 존재하지 않기 때문에 사용자는 용도에 맞는 것을 선택하여 사용할 수밖에 없다. 최근의 경향은 비교적 정밀성이 크게 요구되지 않고 다량으로 사용되는 곳에는 접촉식 온도센서들이 사용되어 왔으나 비접촉식이고 다기능화 된 온도센서들이 급격히 개발됨으로 인해 이들이 점차 시장을 확산해 가고 있는 추세이다.

온도센서는 크게 접촉식과 비접촉식으로 분류한다. 접촉식은 측정대상물과의 접촉을 통해 온도를 측정하는 방식으로 백금 저항온도센서, 서미스터, 열전대, 바이메탈등 대부분의 센서가 이에 해당하고 비접촉식에는 방사온도계, 광 고온도계가 있다.

접촉식 온도센서는 측정대상에 온도 센서를 직접 접촉시키는 방식이며, 측온의 기본형이다. 그러나 직접 접촉으로 인해 측정대상의 열에너지가 센서로 이동하기 때문에 측정대상의 온도가 저하된다. 따라서 열에너지가

작은 검출대상에서는 정확한 온도 측정이 힘들기 때문에 측정대상의 열용량이 센서 엘리멘트에 비해 충분히 커야하며, 센서 자체가 측정대상의 온도에 견디면서 동작을 할 수 있어야 한다.

비접촉식 온도센서는 측정대상에서 방사하는 열선을 계측하는 센서이기 때문에 접촉식 온도센서에서와 같은 열에너지의 손실은 없으며 상당히 떨어진 물체의 온도 측정도 가능하기 때문에 접촉식에서는 불가능했던 다양한 응용도 가능하지만 열에너지를 검출할 수 있는 광학계와 여러 가지 부재료를 필요로 하기 때문에 일반적으로 고가가 되기 쉽다. 비접촉식 온도센서는 열에너지 방사의 검출기일 뿐이므로, 방사 강도에만 의존하는 센서가 바람직하며 접촉식과는 달리 센서 출력이 온도에 의존해서는 안 된다.

자연계에 존재하는 재료는 거의 대부분 온도에 대한 의존성을 가지므로, 이론적으로는 모든 재료가 온도센서로 응용이 가능하며 온도센서 재료는 온도에만 의존해서 동작하는 것이 바람직하고 전계, 압력, 자계 등 다른 변수들에 대해서는 안정한 특성을 가져야 한다. 이러한 조건 때문에 실제의 온도센서에 사용되는 재료는 한정된다.

온도센서는 열적, 역학적, 전기적, 자기적 및 이들이 복합된 재료의 물리적 특성을 이용하고 있다. 표 4-2에 접촉형 온도센서와 비접촉형 온도센서를 비교하였다.

Table 4-2 A Comparison of Contact and Non-contact Temperature Sensor

| | 접촉방식 | 비접촉방식 |
|---|---|---|
| 필요조건 | · 측정대상에 검출소자를 접속시켰을 때 측정대상의 온도가 실용상 바뀌지 않아야 한다.<br>· 측정대상과 검출소자를 잘 접촉시켜 야 한다. | · 측정대상에서의 열에너지 방사가 검출소자에 충분히 도달하여야 한다.<br>· 측정대상의 열에너지 실효 방사율이 명확하게 알려져 있거나 재현가능한 것 |
| 장점 | · 측정개소를 임의로 할 수 있다. | · 운동하고 있는 물체의 온도도 측정 가능하다.<br>· 검출소자의 접촉을 필요로 하지 않으므로 측정대상의 측정량이 변화하는 것은 일반적으로 없다. |
| 단점 | · 운동하고 있는 물체의 온도는 측정하기 어렵다.<br>· 열용량이 작은 측정대상에서는 검출소자의 접촉에 의한 측정량의 변화가 생길 수 있다. | · 장치가 복잡하며 접촉형 온도센서에 비하여 고가이다. |
| 온도범위 | · 1000[℃] 이하의 온도 | · 일반적으로 고온의 물체를 측정하는데 적합하다. |
| 오차 | · 일반적으로 눈금 스팬의 1[%] 정도 | · 일반적으로 10[℃] 정도 |
| 지연 | · 일반적으로 크다. | · 일반적으로 작다. |

## *B.* 적외선 센서

적외선 센서는 서미스터, 서모파일 등의 열형과 PbS, InSb, HgCdTe 등의 반도체를 사용한 양자형으로 크게 나누어진다. 이들의 특성상 차이점은 대조적이므로 알기 쉽게 표 4-3에 정리하였다. 열형 센서는 감도가 낮고 응답이 늦지만 파장 전역에 걸쳐서 감도가 거의 없다는 특징이 있다. 양자형 센서는 고감도이고 응답도 빠르지만 감도의 파장 선택성이 강하고 사용 파장을 확인하여 센서를 선택하여야 한다.

Table 4-3 Characteristics Comparison of Infrared Sensor

| | 열형 | 양자형 |
|---|---|---|
| 감 도 | 낮다(수[V/W]) | 높다(103[V/W]) |
| 응답속도 | 늦다(~100[ms]) | 빠르다(~1[μs]) |
| 사용온도 | 실온(300[K]) | 냉각이 필요(77[K]) |
| 스펙트럼 특 성 | 평탄 | 선택성이 강하다 |
| 적외선 이미지용 | 부적당 | 적당 |
| 사용예 | 서미스터<br>서모파일<br>초전센서 | InSb<br>HgCdTe<br>Ge(Au) |

양자형 센서를 구조상으로 분류하면 광전도형(photo conductive type, PC형), 광기전형(photo voltaic type, PV형), 광전자형(photo electro magnetic type, PEM형), 쇼트키형(schottky type) 으로 나눌 수 있으며 이들의 구조는 그림 4-1에 나타내는 바와 같이 PC형은 반도체 박막형 저항소자이며 빛의 입사에 의해 저항 변화가 일어나서 그것을 전압 변화로서 신호를 꺼내는 방식이다. PV형은 광전지(photo cell)로 알려져 있는 것으로서 플레이너형의 pn 접합다이오드이며, 빛의 입사에 의해 접합부에 발생한 전자-정공쌍을 광전류로서 꺼내는 구조로 되어 있다. PEM형은 전계와 자계를 동시에 건 경우에 빛을 조사함으로써 기전력을 신호로써 꺼내는 것이다. 쇼트키형은 금속과 반도체의 접촉에 따라 형성되는 쇼트키 장벽(Schottky Barrier)을 이용한 다이오드이며, pn접합과 거의 같은 원리로 신호가 발생한다.

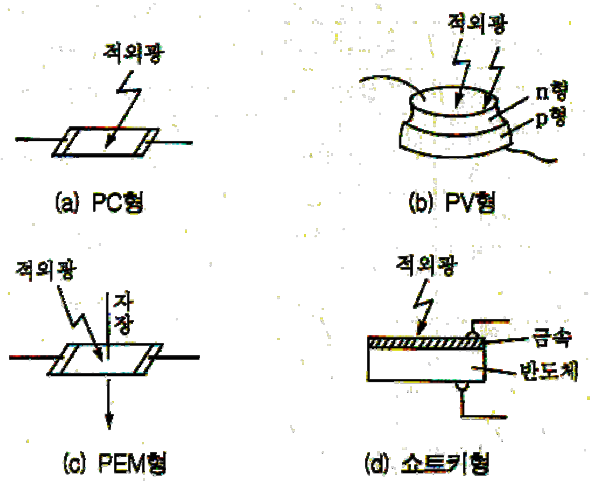



Fig. 4-1 Structure Classification of Infrared Rays Sensor

반도체 재료를 사용하여 적외선을 센싱하기 위해서는 우선 적외선이 그 재료에 효율이 좋게 흡수되어야 한다. 지금 $I_0$세기의 적외선이 반도체에 입사하여 두께 $x$점에서 광의 세기를 $I$라 하면 $I = I_0 \exp(-ax)$와 관계를 가지며, 반도체 중에 적외광은 흡수된다. 여기서 $a$는 흡수계수이지만 그 내용은 매우 복잡하다. 그림 4-2는 적외선 센서의 광도전 기구를 나타내고 있으며 이 흡수기구는 반도체의 밴드갭을 넘어서 빛에 의한 여기 전자가 전도대에 전이하는 것이 기본이며 그 에너지 이하의 광자에서는 검출되지 않는다. 따라서 이 밴드갭 $E_g$[eV]에 해당하는 파장을 차단 파장 (cut off wave length) $\lambda_c$라 하며

$$\lambda_c[\mu m] = \frac{1.24}{E_g[eV]} \tag{4-1}$$

로 주어진다. 반도로 반도체 $E_g$가 결정되면 센서 재료로서의 장파장 측의 검출한계 파장이 결정된다.

Fig. 4-2 Photo Conductive Mechanism of Infrared Rays Sensor

## *C. A2TPMI* 센서

본 연구에 사용된 적외선 센서의 특징은 표 4-4와 같다. 그림 4-3은 Perkin Elmer 사에서 제작된 A2TPMI 적외선 온도센서로서 -20~300 [℃]의 온도 범위를 지니는 모델로 적외선 센서의 파장은 5.5~13.5[㎛]이나 여기에 편광필터를 적용하여 7.5[㎛]를 사용하도록 주문 제작하였다. 그림 4-4는 A2TPMI 센서 사진을 나타내고 있다.

Table 4-4 Characteristics of Infrared Rays Sensor

| | |
|---|---|
| 1 | Smart thermopile sensor with integrate signal processing. |
| 2 | Can be adapted to your specific measurement task. |
| 3 | Integrated, calibrated ambient temperature sensor. |
| 4 | Output signal ambient temperature compensated. |
| 5 | Fast reaction time. |
| 6 | Different optics and IR filters available. |
| 7 | Digital serial interface for calibration and adjustment purposes. |
| 8 | Analog front end / back end, digital signal processing. |
| 9 | E2PROM for configuration and data storage. |
| 10 | Configurable comparator with high/low signal for remote temperature threshold control. |
| 11 | TO 39 6 pin housing. |

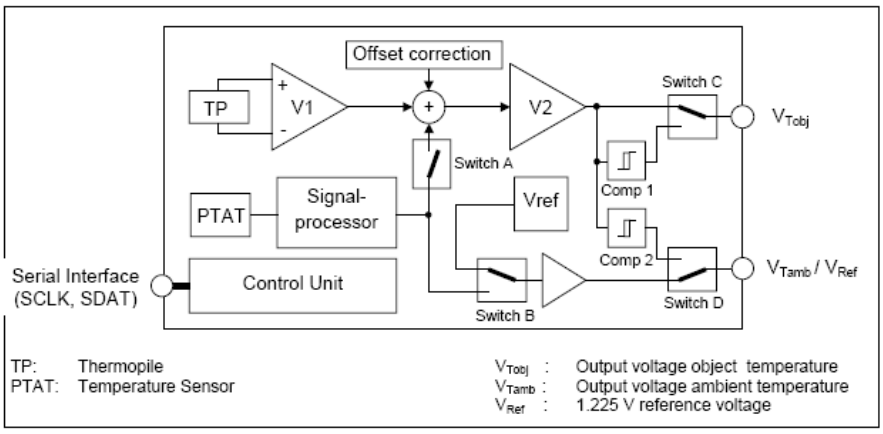



Fig. 4-3 A2TPMI Block Diagram

(a) A2TPMI 센서 (b) 편광필터 (c) 편광필터 부착

Fig. 4-4 A2TPMI Sensor

A2TPMI 센서 내부에 신호처리 회로가 들어가 있어 기존의 센서처럼 외부에 복잡한 증폭 및 필터회로를 구성할 필요가 없으며, 외부와는 I2C 통신을 통해 검출된 값을 외부 회로에 디지털로 전송한다. 내부 신호처리 레지스터는 8비트 분해능을 지니고 있다.

다만, 이 센서는 직선거리 2[m]가 적정 거리이며, 기본 ±15°의 각내에서 정확한 측정을 할 수 있다. 다른 영역은 FOV(Field of View) 영역의 50% 출력을 내기 때문에 유의해야한다.

# V. 실 험

## A. 시스템의 구성

본 연구에서 사용한 적외선 온도감시 개발시스템은 계측 PC, 임베디드 시스템 및 계측장비로 구성되었다. 소프트웨어 제작툴은 Visual C++ Ent 6.0을 이용하여 제작되었으며 임베디드 시스템 모듈은 Core, 센서, A/D 컨버터, D/A 컨버터, CAN Driver로 구성되며 본 시스템 모듈에 적용된 소자는 표 5-1에 나타내고 있다. 고속 시리얼통신으로 PC에 USN 서버와 연결하고, 서버와 USN 노드를 이용하여 USN 네트워크를 구축하고, 제작된 임베디드 시스템 모듈로 온도 측정 실험을 수행하고 그 결과를 분석하였다. 그림 5-1은 설치된 전체 실험시스템의 블록다이어그램이며, 그림 5-2는 설치된 시스템의 사진이다.

Table 5-1 Composition of Experiment System

| Composition | Standard |
|---|---|
| Core | AT91RM9200(server)<br>AVR:Atmega163(client) |
| 센서 | ZF5583, A2TPMI |
| A/D | MCP3202(12bit) |
| D/A | DAC8512(12bit) |
| CAN Driver | 82C250 |

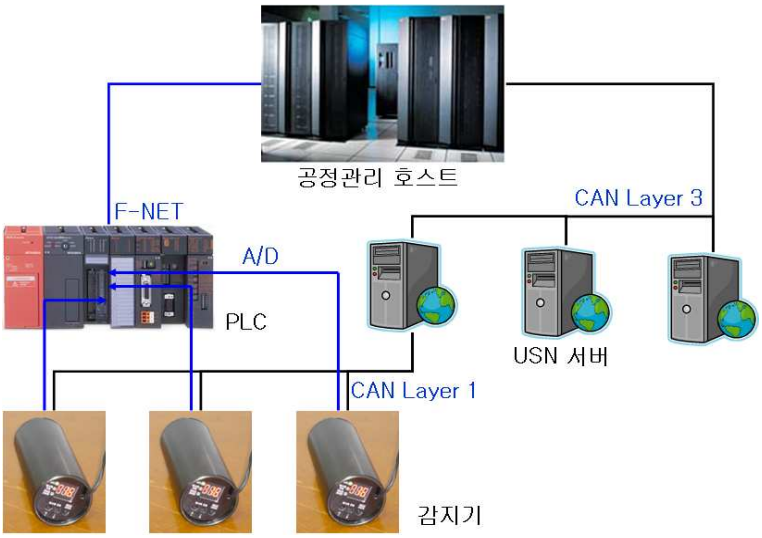



Fig. 5-1 Block Diagram of Full System Setup

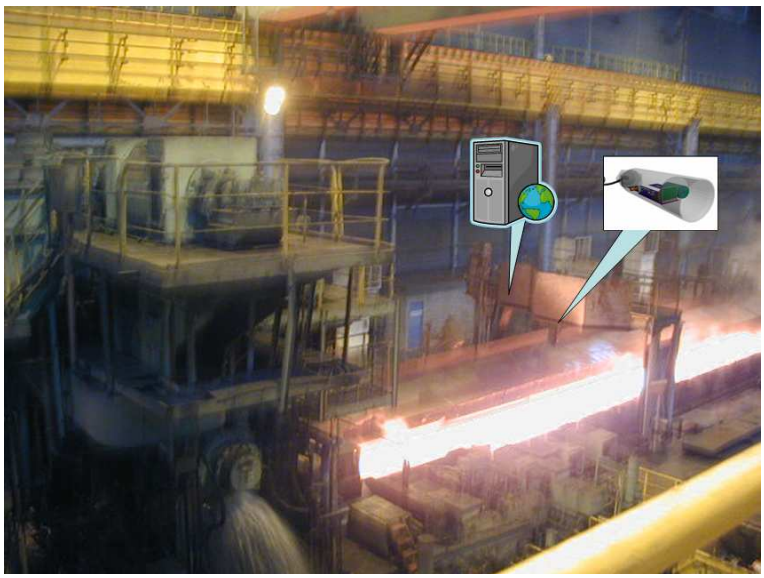


Fig. 5-2 The Adaptive System for Process

## 1. USN 서버설계

CAN 네트워크에 참여하는 노드들로부터 통신을 담당하는 CAN Driver 칩이 마이크로프로세서에 연결되어 있으며, 모니터링을 담당하는 PC에 LAN으로 100[Mbps] 전송속도로 통신을 통제할 수 있다. 서버에서는 직결되는 노들을 위해 D/A, A/D, PWM, 엔코더 카운터 기능들을 수행하도록 FPGA에를 탑재하여 처리하는 서버용 컨트롤 보드를 제작하였다. 로컬 독립적으로 7인치 TFT LCD 탑재를 고려하였으나 로컬 컨트롤 보다는 중앙 감시실에서 한 번에 통제를 하도록 소프트웨어 레이어로 분리 추진하도록 하며, 차량에서와 같이 CAN 네트워크를 4개로 분리 가능하도록 설계하였다. 그림 5-3은 USN 서버제어 보드 블록도를 나타내고 있다.

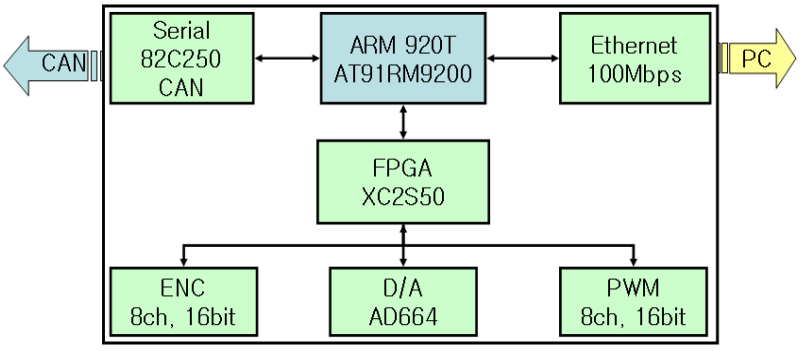



Fig. 5-3 Block Diagram of USN Server Control Board

마이크로프로세서는 Atmel사의 AT91RM9200 으로 내부처리는 400[MHz]로 처리하도록 하였고, 별도의 O/S 탑재하지 않고 자체 개발한 리얼타임 O/S로 구동한다. 플래시 롬은 AM29LV320DB, RAM은 HY57V641620HG를 사용하여 각 세그먼트는 512 [Mbyte] 공간을 각기 확보하였다. FPGA는 AT17V010A 시리얼 PROM을 이용하여 마스터 부트 모드로 사용하도록 하였다. 그림 5-4는 제작된 서버 컨트롤 보드의 사진이다.

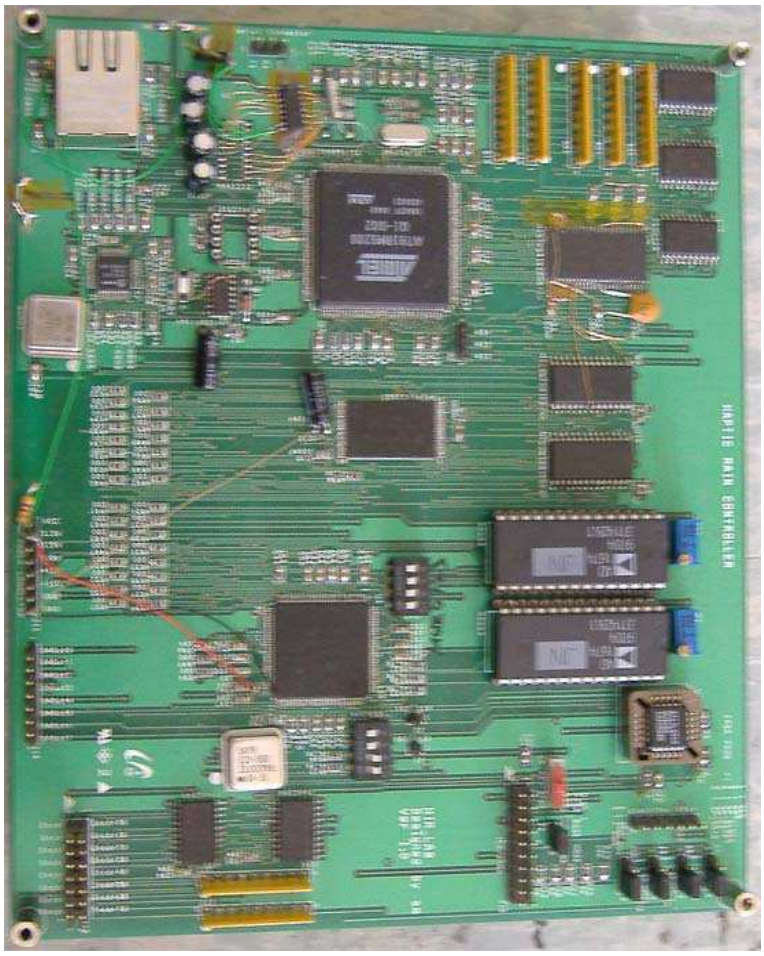



Fig. 5-4 The Photograph of USN Server Control Board

## 2. *USN* 노드 감지기 설계

본 연구에서 사용된 적외선 센서는 자체 내에 I2C 통신에 의해 감지된 값을 마이크로프로세서에 의해 디지털로 직접 받아 볼 수 있지만 본 연구에서는 분해능이 높은 방식을 위해 아날로그 값으로 출력되는 값을 12비트 분해능을 가진 MCP3202 A/D 컨버터에서 신호를 변환하여 마이크로프로세서 모듈에 전송을 하도록 구성하였다. 그림 5-5는 적외선 센서 구동 블럭도를 나타내고 있다.

사용된 적외선 센서의 응답속도는 25[ms]이다. 따라서 이 처리속도는 고가의 마이크로프로세서를 사용하지 않더라도 처리가 가능하기 때문에 저가로 RISC 구조에 안정성이 높은 Atmel사의 AVR시리즈인 Atmega 163을 사용하여 감지기를 설계 제작하였다.

공정에서는 DC24[V]를 사용하기 때문에 이를 3.3[V]또는 5[V]를 사용하기 때문에 정전압 레귤레이터 LM7805를 사용하여 전류를 0.1[A]내에서 사용하도록 설계하였다.

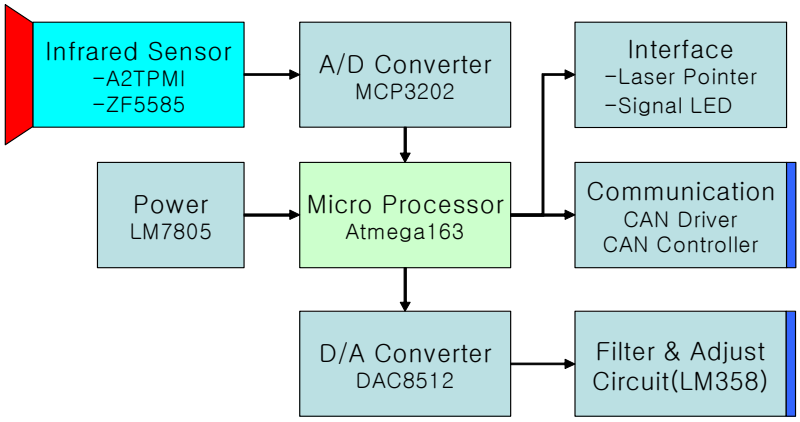



Fig. 5-5 Infrared Sensor Driving Block Diagram

공정에서 요구사항은 USN 네트워크를 구성하기 위한 CAN을 구성하고, 현장에서 직접 조작 및 모니터링이 가능해야 하기 때문에 현장 감지기의 온도 표시용 7-세그먼트 출력, 온도 데이터를 프로토콜을 통해 CAN 네트워크 출력, 아울러 PLC에 연동하기 위해 0V-10[V]의 아날로그 출력  1채널 등 3가지 공정 요구사항을 만족하였다.

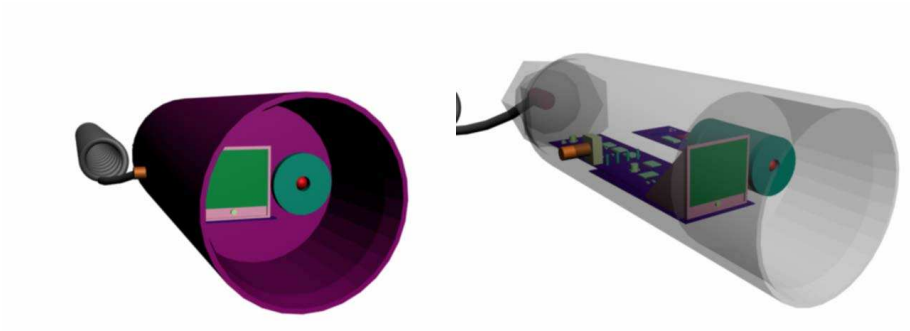
따라서 12비트 분해능의 아날로그 출력을 위해 DAC8512를 사용하여 PLC에 연결하여 모니터링 할 수 있도록 필터 및 조정 회로를 내장하였다. 특히 USN을 위해 고유 ID 설정 및 원격 모니터링 기능을 추가함으로서 공정제어에 USN을 구성할 수 있는 기본 감지기 역할을 수행할 수 있도록 하였다.

특히, 적외선 온도감지기는 측정하고자 하는 곳에서 2[m]정도 떨어진 거리에서 수직으로 입사되는 파장을 측정하는 것이 일반적이다. 따라서 측정하는 위치를 지정하도록 레이저 포인터를 통해 지시하게 한다. 본 시스템에서는 레이저를 쏘면 5초 후에 자동으로 꺼지도록 설계하였다.

그림 5-6은 3DMAX를 사용하여 감지기를 설계한 모형이다. (a) 그림의 왼쪽이 ZF5585센서이고, 오른쪽은 A2TPMI 레이저 다이오드 센서이다. (b)는 직경 30[mm]원통에 센서부와 가운데 모듈을 구성하고, 뒷면으로 단자를 빼내 결선할 수 있도록 설계를 하였다.

그림 5-7은 감지기에 사용된 마이크로프로세서의 기본회로를 나타내고 있다.

그림 5-8은 실제 제작한 감지기의 사진을 나타내고 있다. 전면에는 편광필터를 통해 적외선이 입사하도록 하였고, 뒷면에 방사율 조정 스위치와 레이저 포인터를 조절할 수 있는 S/W를 배치하였다. 세그먼트에는 측정되는 온도를 지속적으로 모니터링 할 수 있도록 하였고, 통신선을 통해 PLC 및 USN에 접속할 수 있는 케이블이 준비되었다.



(a) 감지기 내부 (b) 감지기 프레임

Fig. 5-6 Prototype of Designed Sensor

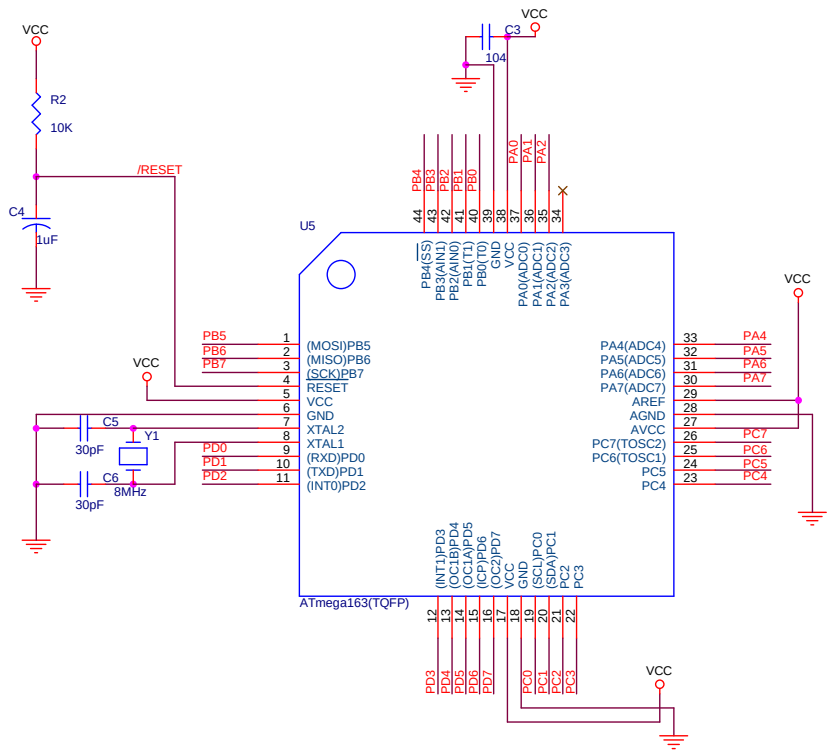



Fig. 5-7 Microprocessor Circuit

(a) 제작된 감지기 모형

(b) 전면

(c) 후면

Fig. 5-8 Infrared Thermometer Monitoring System

## 3. 감지기 프로그램 개발

감지기에서 아날로그 전압을 입력받아 온도를 산출하는 프로그램을 제작해야한다. 이를 위해 회사에서 제공하는 데이터시트를 통해 변환 함수를 제작해야 한다. 일반적으로는 MatLAB이나 수치해석 기법을 사용하지만 본 연구에서는 간편하게 이들 해법을 이용하여 다차원 함수를 뽑아주는 커브핏(Curve Fit)프로그램을 이용하여 변환함수를 찾았다. 산출된 변환함수는 식 5-1과 식 5-2에 나타내었다.

그림 5-9는 회사에서 제공하는 특성곡선이다. 온도가 300[℃]일 때 전압 출력이 3[V]를 나타내고 있지만 전 영역에서 선형적이지는 않았다. 특히 12비트 고 분해능으로 온도를 측정 할 때는 온도에 따른 시스템 응답함수와 보정함수들이 더 정확한 값을 계산하도록 준비가 되어야 한다.

그림 5-10은 150[℃]전후의 강판 온도 측정을 위해 보정을 한 특성 곡선을 나타낸다. 이 특성 곡선은 평균 3.73[℃]의 편차를 지니고 있으며, 저온보다 고온일수록 편차가 더 크게 나타난다. 따라서 이를 보정하여 정밀하게 측정하도록 수정한 특성 곡선을 그림에 나타내었다.

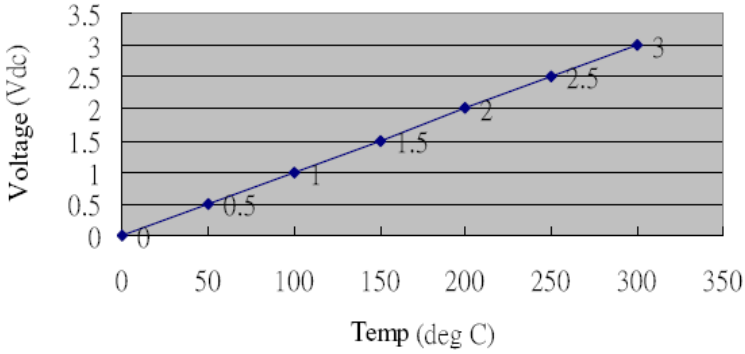



Fig. 5-9 ZF5585 Characteristics Curve

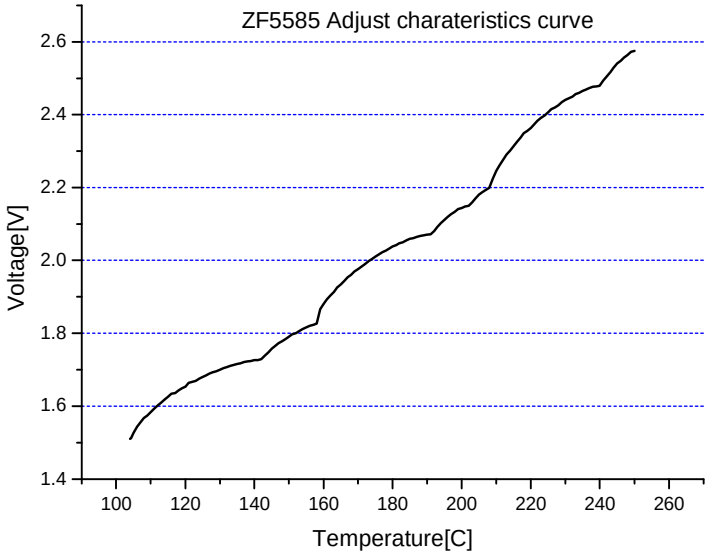



Fig. 5-10 ZF5585 Adjust Curve

그림 5-10의 특성곡선에서 2차 함수를 구하면 식 5-1과 같다. 물론 고차식일수록 정확하겠지만 마이크로프로세서에서의 계산시간 지연으로 인한 손실이나, 아니면 테이블에 의한 방법의 경우 메모리를 많이 소비하는 단점 등이 있다.

$$y = a + bx + c/x^2 \qquad (5\text{-}1)$$

여기서, a = 48.220069, b = 89.751602, c = -182.2941의 값이 산출 되었다.

$$F(x) = Rad*(y - temp) + temp \qquad (5\text{-}2)$$

출력할 온도는 방사율을 보정해야하는데 식 5-2와 같이 방사율 Rad, 현재온도 temp 등을 이용하여 보정된 출력 값을 표시하면 된다.

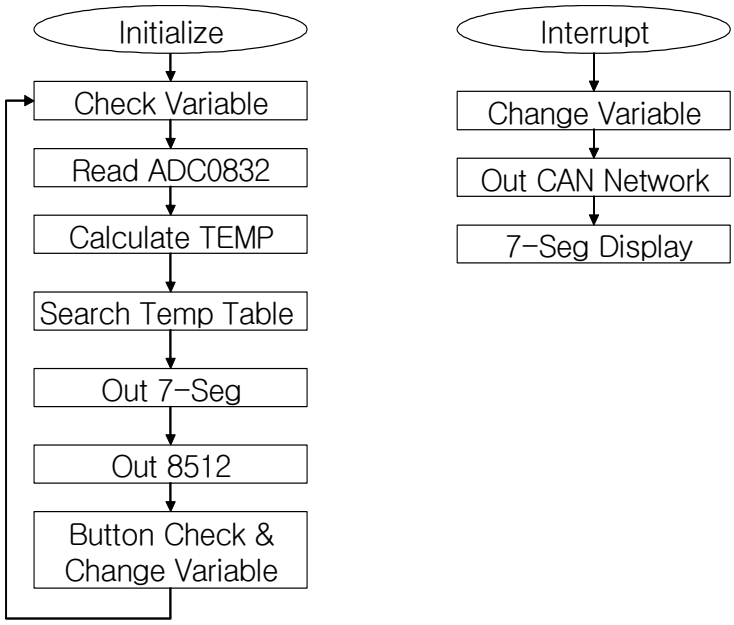
적외선은 물체의 방사율에 따라 상당한 차이를 나타낸다. 따라서 적외선 온도 감지기를 설계할 때는 측정대상물체가 무엇인가가 이 때문에 중요하다. 그렇지만 일반적인 적외선 온도계의 경우는 이 설정을 자유롭게 변경하기가 힘들다. 따라서 본 연구에서는 대상체가 주로 강철판이지만 성분에 따라서도 차이가 있기 때문에 수동적으로 방사율을 설정하는데도 개발 목적이 있었다.

방사율을 자유롭게 설정할 수 있는 본 감지기는 그렇기 때문에 더 정확한 온도를 측정할 수 있게 하였다.

그림 5-11은 앞서 보정된 온도 데이터를 45~203[℃]범위에서 테이블에 만들어 넣고, 이 데이터를 앞서 제시한 식 5-2에 의해 계산하여 그 결과를 테이블에서 찾아서 출력 하도록 프로그래밍 하였다. 이는 식에 의

해 계산할 경우보다 고속으로 처리가 가능하며 정확성도 높다. 이는 출력할 데이터가 그리 많지 않기 때문이기도 하다. 본 알고리즘의 소스는 부록에 첨부하였다.

USN 네트워크에 연결하고자 할 경우 인터럽트를 통해 시스템에 파라미터 변경과 현재 온도데이터를 요청할 수 있도록 하였고, 각기 ID는 출하 시 설정하도록 하였고, 서버에서 4096개 노드 이상을 컨트롤 할 수 있도록 하였다. 다만 서버 프로그램은 앞으로 개발예정이나 클라이언트인 감지기는 이 서버의 요청을 수행할 수 있도록 준비하였다. 차후 USN 서버프로그램에서 다중 센서를 감시할 수 있는 시스템으로 발전이 가능하다.




(a) 메인부 (b) 인터럽트부

Fig. 5-11 Main Program flow Chart

## *B.* 실 험

### *1.* 온도 보정 실험

그림 5-12에서 나타낸 보정 곡선은 본래의 데이터 시트에서는 센서가 선형적으로 출력한 것으로 예상했지만 여러 번의 실험결과 모든 구간에서 선형 출력을 나타내지 않았다.

따라서 보정 알고리즘을 사용하여 보정을 수행해야 정확한 값을 수행할 수 있음을 알 수 있었다. 그러므로 이를 만족하는 보정곡선의 정확도를 높이기 위해 9차 폴리핏 보정을 수행하였다. 그림에서처럼 적색의 직선 보간, 청색의 9차 폴리핏을 적용한 결과 이들과 서로 특성을 비교했을 때도 100[%]로 데이터가 일치 하지 않음을 알 수 있었다.

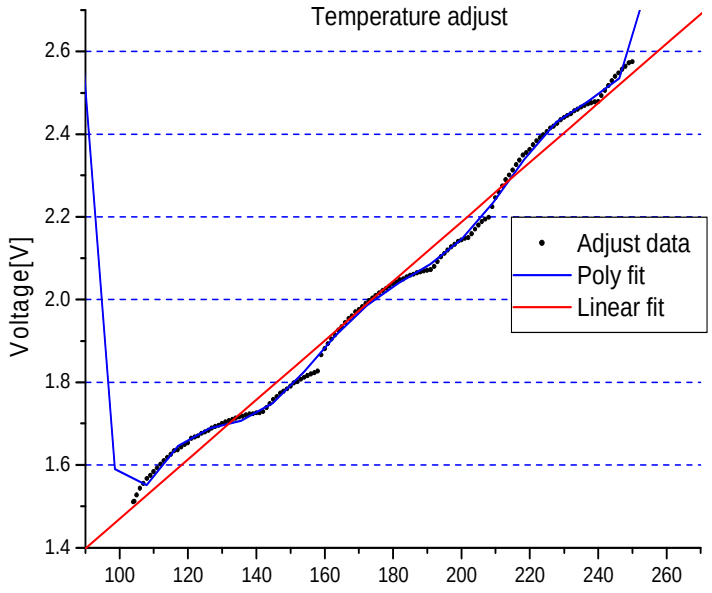



Fig. 5-12 Temperature Adjust

따라서 여러 번의 보간 및 실측결과 흑색의 보간 점을 얻을 수 있었다. 그러나 USN 노드의 실시간 처리를 위해서 입력에 따른 결과를 9차 함수를 이용해 계산하는 것은 실시간 처리를 확립할 수 없으므로 사전에 계산된 보간 점을 테이블화하여 입력대 출력값을 테이블에서 찾는 방법으로 정확하고 빠르게 온도를 측정할 수 있었다. 또한 이 보간 테이블을 통해 고속으로 온도가 산출됨으로 인해 적외선 센서의 측정주기가 25[ms]이지만 임베디드 시스템의 처리속도는 50[us]이면 계산이 가능함을 확인할 수 있었고, 이는 보다 더 성능을 개선할 수 있는 여지를 남겨놓고 있다.

따라서 A2TPMI 센서의 처리속도는 25[ms]일지라도 USN 노드에서 다음 입력 값이 나올 때까지 구간을 스플라인(spline) 보간을 통해 센서의 출력을 항상 1[ms]의 측정 속도로 안정화 시킬 수 있었다.

그림 5-13에서는 USN Node에서 3차원 폴리핏 기법을 적용해 센서에서 다음 데이터가 나오지 않더라도 입력 시부터 보간을 수행하여 1[ms]에 한 번씩 출력하도록 알고리즘을 개선하여 응답특성을 크게 개선하였다. 그림의 흑색선과 사각형으로 표시된 ZF5585의 출력특성을 NI A/D 보드를 이용해 캡쳐된 데이터와 같은 입력을 받아 처리하는 USN Node의 출력값을 서로 비교하여 표시하였다. 100[ms]까지의 안정화 시간중에는 보정이 부정확하나 그림 5-14와 같이 보정 후에는 안정된 출력을 보인다. 보정 이전 시간까지는 시스템 초기화후 100[ms]지난 시점부터는 안정된 응답 특성을 나타낸 반면 ZF5585는 50[ms]이후 안정되나 이후 출력이 안정되지 않고 빠른 출력을 내보내지 못하는 문제점이 개선되었다.

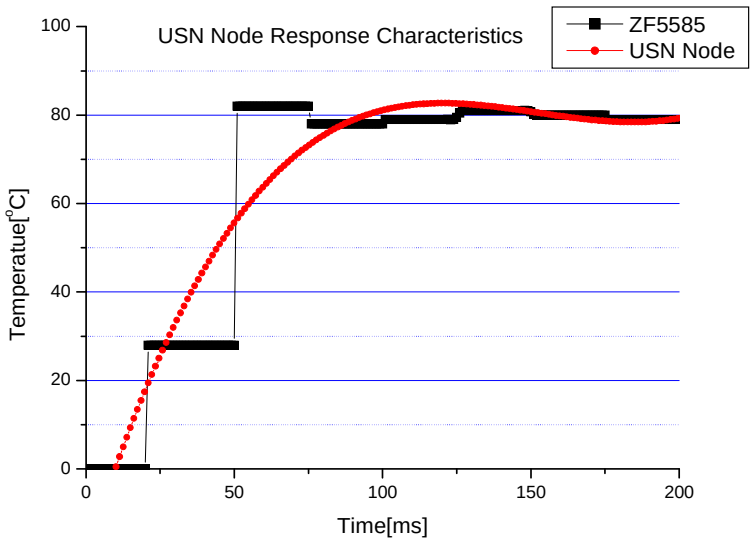



Fig. 5-13 Response Characteristics

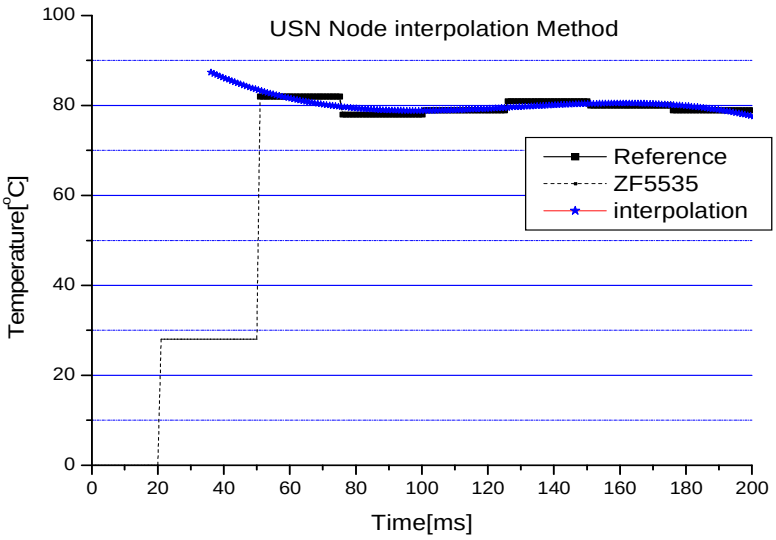



Fig. 5-14 Interpolation Characteristics

그림 5-15에 본 시스템의 우수함을 확인하기 위해 공장자동화에 자주 사용되는 PT100[Ω] 열전대를 사용하여 본 연구에서 개발된 USN 노드와 온도측정 시간을 비교하였다. 비교 결과를 그림 5-15에 정리하였다. USN 노드는 측정값과 표준 온도발생장치의 지시 값에 오차 5[%]이내에서 안정된 값을 나타내는 시간을 기준으로 나타내었고, PT100은 온도 컨트롤러를 사용하여 같은 방법으로 시간을 측정하였다. USN 노드는 적외선 방식으로 바로 측정이 가능하여 온도에 따른 시간 오차도 균일한 특성을 나타내었다. 그러나 PT100[Ω]의 경우는 자체 쉴드 가열 시간을 고려되어 온도가 높을수록 측정시간이 늘어나는 단점을 나타내고 있다. 그러나 USN 노드에 적용된 센서의 최대 온도값이 250[℃]임으로 250[℃]이상은 측정이 되지 않는 단점도 있으나 고온용 센서와 병렬처리 함으로서 이를 개선하는 방법을 연구해야한다.

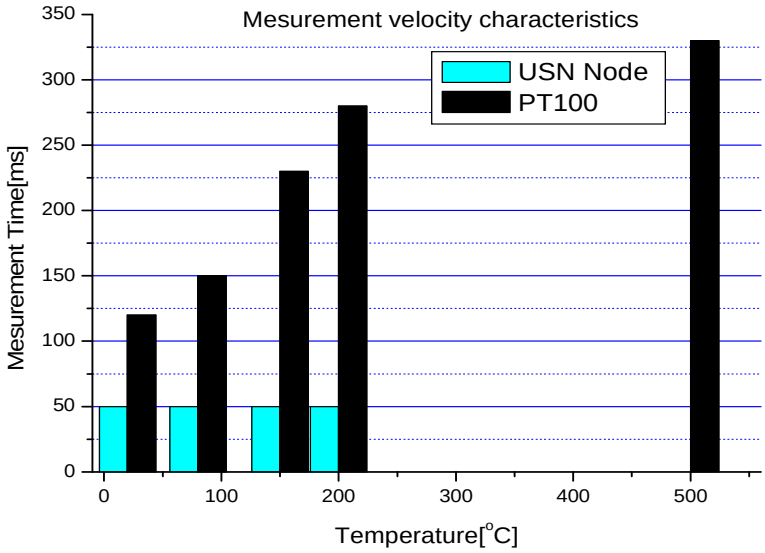
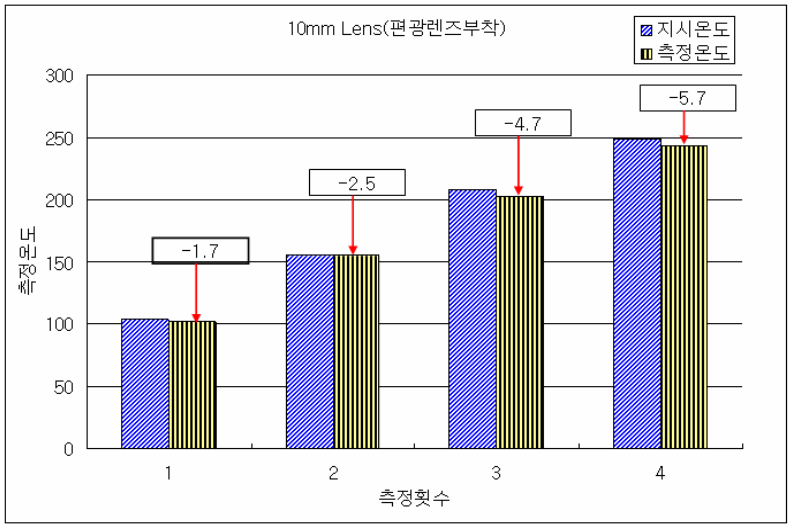
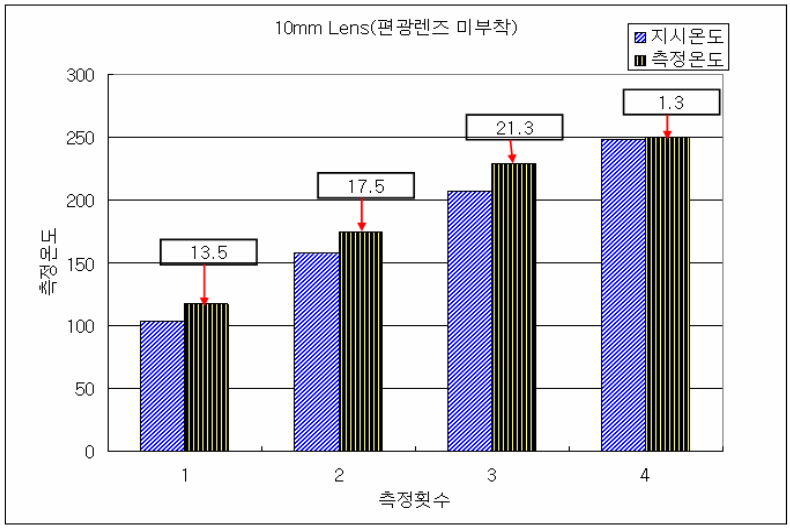



Fig. 5-15 Velocity Comparison of PT100[Ω] and USN Node

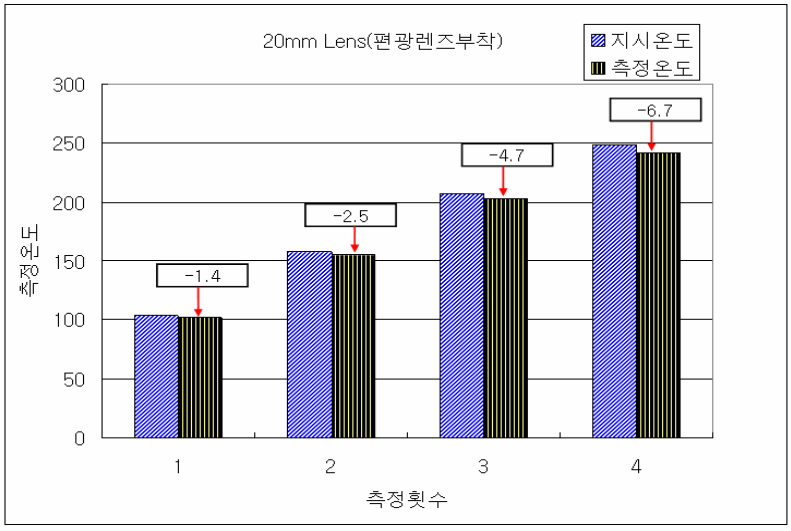
그림 5-16에 나타낸 측정결과는 표준 온도측정기에서 검증받은 자료로 실험은 표준 온도를 발생하는 온도 패널에 지시온도와 측정기의 온도를 비교 측정한 결과를 그래프에 나타내었다. (a)는 10[mm] 렌즈를 탑재하고, 루비코팅 편광렌즈를 장착한 결과를 나타낸 것으로 그래프에 온도대별 온도오차를  표시하였다. 그 결과에서 알 수 있듯이 오차는 2[%]안에 들어가고 있음으로 제시오차 5[%]이내 임으로 양호한 특성을 확인하였다. 그림 (b)는 편광 렌즈를 미부착한 상태로 상대적으로 오차가 증가한 것을 알 수 있다. 그림 (c)는 20[mm] 렌즈를 탑재하고 측정한 결과로 가장 양호한 결과를 알 수 있다. 이는 렌즈 구경이 클수록 입사하는 적외선 양이 증가하여 정확한 측정이 가능하고, 루비코팅 편광필터를 탑재할 경우 우수한 측정결과를 나타냄을 알 수 있었다.




(a) 10mm 편광렌즈 부착시




(b) 10mm 편광렌즈 미부착시




(c) 20mm 편광렌즈 부착시

Fig. 5-16 A Test Result of Standard Temperature Generator

Table 5-2 Measurement Data [단위: ℃]

| No. 1 (10φ Lens) | | | No. 2 (10φ Lens) | | | No. 3 (20φ Lens) | | |
|---|---|---|---|---|---|---|---|---|
| 지시온도 | 센서온도 | 편차 | 지시온도 | 센서온도 | 편차 | 지시온도 | 센서온도 | 편차 |
| 103.7 | 102 | -1.7 | 103.5 | 117 | 13.5 | 103.4 | 102 | -1.4 |
| 157.5 | 155 | -2.5 | 157.5 | 175 | 17.5 | 157.5 | 155 | -2.5 |
| 207.7 | 203 | -4.7 | 207.7 | 229 | 21.3 | 207.7 | 203 | -4.7 |
| 248.7 | 243 | -5.7 | 248.7 | 250 | 1.3 | 248.7 | 242 | -6.7 |

표 5-2의 데이터는 그림 5-16에 나타낸 데이터를 종합한 결과로 100, 155, 203, 244[℃]의 표준 온도 패널을 2[m]거리에서 3가지 종류의 적외선 감지기를 통해 온도를 측정한 검정결과이다. 이 감지기가 적용될 공정에서는 대부분 이 범위의 온도에서 모니터링을 수행하기 때문이다.

1번 감지기는 직경 10[mm] 편광렌즈를 부착한 센서이고, 2번 감지기는 동일하나 수직으로 받지 않은 상태이다. 3번 감지기는 직경 20[mm] 편광렌즈를 부착한 시스템이다. 앞서의 자료에서 렌즈의 구경과 편광필터 채용여부에 따라서 그 결과가 상이함을 확인하였고, 부가적인 오차의 원인을 분석하면 온도가 높을수록 편차가 벌어지는 이유는 A/D분해능에 기인한다.

따라서 본 결과까지 반영한다면 더 정확한 온도를 측정할 수 있으나, 오차가 일어나는 원인은 온도센서 내부의 컨트롤러가 8비트 분해능을 가지고 처리함으로 이런 부정확한 결과가 내포하고 있음을 의미한다. 더나가 분해능 높은 센서로 교체시 보다 정확한 감지기를 설계할 수 있을 것으로 판단된다.

## 2. *USN* 네트워크 실험

그림 5-17은 USN 각 노드의 인터럽트 측정 타임을 오실로스코프 [LeCroy 9354AM]을 이용하여 측정한 그림이다. 측정결과 각 USN 노드는 1[ms]에 동기되어 출력을 나타내고, CAN 버퍼에 이 데이터를 뿌려준다. CAN 컨트롤러는 CAN 버스가 Busy 상태가 아니면 데이터를 서버에 전송하고, 그 결과를 서버가 분석하여 PC의 모니터 프로그램에 그 처리결과를 확인할 수 있도록 제작하였다.

그림 5-18의 서버측 CAN 네트워크의 구성은 다중 레이어 방식을 적용하여 2계층으로 분리하여 구성하고, 각 계층은 서버보드를 통해 PC와 인터렉션이 가능하도록 구성되어 졌다. 이론상 노드 수는 제한이 없지만 본 실험에서는 노드수를 제한하고 실험하였고, 그 실험 결과를 그림 5-19에 나타내었다.

그림 5-19에서는 16개 노드를 100[msec]로 감시할 때와 256개의 노드를 1000[msec] 폴링 주기로 감시할 때의 경우로 실험시작 300초 후에 온도측정을 개시하여 트래픽의 증가를 모니터링 하였다. 온도 측정시는 폴링모드에서 인터럽트 모드로 전환하여 데이터 요청 및 조작 명령이 증가하나 트래픽은 2.5배 이상 증가하지 않으므로 3배 이상으로 폴링주기를 선택하면 가능할 것으로 판단된다. 그림 5-20은 PC의 서버프로그램으로 PC와 USN server 보드와는 시리얼로 접속되고, 서버보드에서 수집된 정보를 공유한다. 서버보드는 각 노드들과 CAN 네트워크로 묶이지만, 서버와 PC는 바이너리 통신을 함으로서 대역폭에 지장을 주지 않을 정도의 데이터를 가진다.

다음 프로그램에서는 각 노드의 온도 모니터링 값을 그래프로 실시간 표시하고, 각 노드를 선택적으로 표시할 수도 있다. 만일 고장이 발생시

고장기기의 정보나 상태를 로그창에 알려 줄 수 있다.

인터럽트 방식으로 그림 5-17처럼 1[ms]당 스캔하는 폴링 방식과 문제가 생겼을 때 정보를 전송하는 인터럽트 방식의 수행이 가능하고, 자동 모드는 온도 측정되는 센서들을 모두 모니터링 하는 기능과, 설정된 감지기들만 표시하는 수동 조작 방식을 지원한다. 따라서 사용자는 효과적으로 각 노드들을 모니터링 할 수 있다.

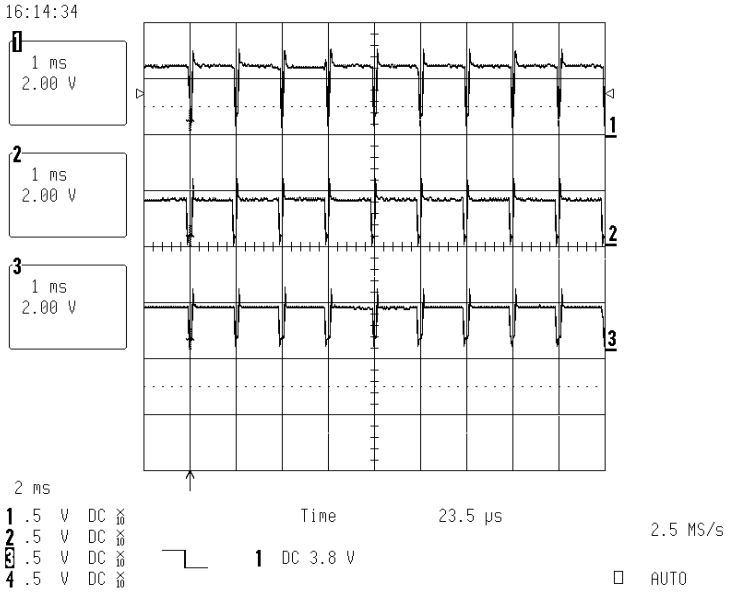



Fig. 5-17 Scan Timing of USN Node

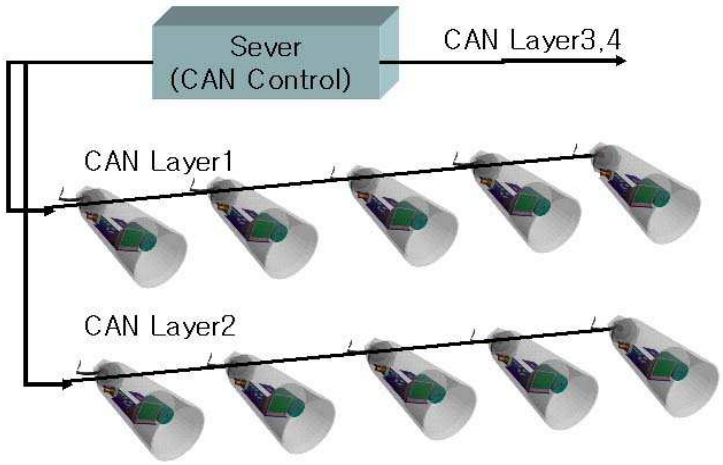



Fig. 5-18 Composition of Server CAN Network

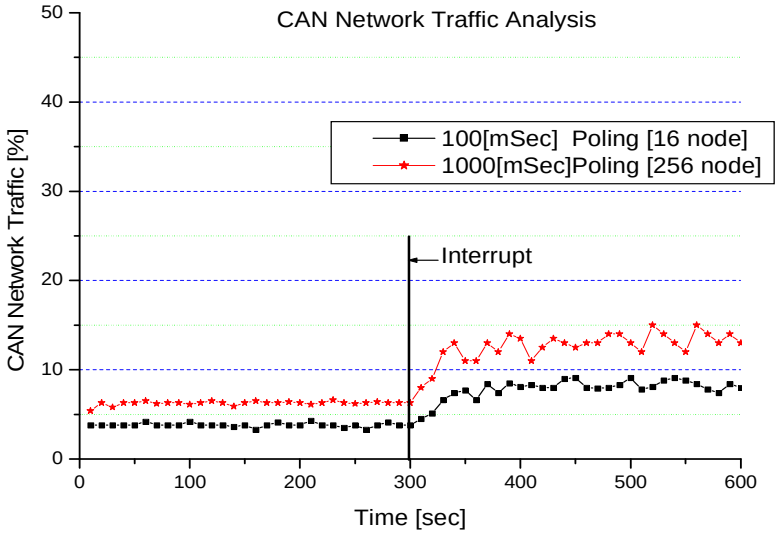



Fig. 5-19 A Traffic Analysis of CAN Network

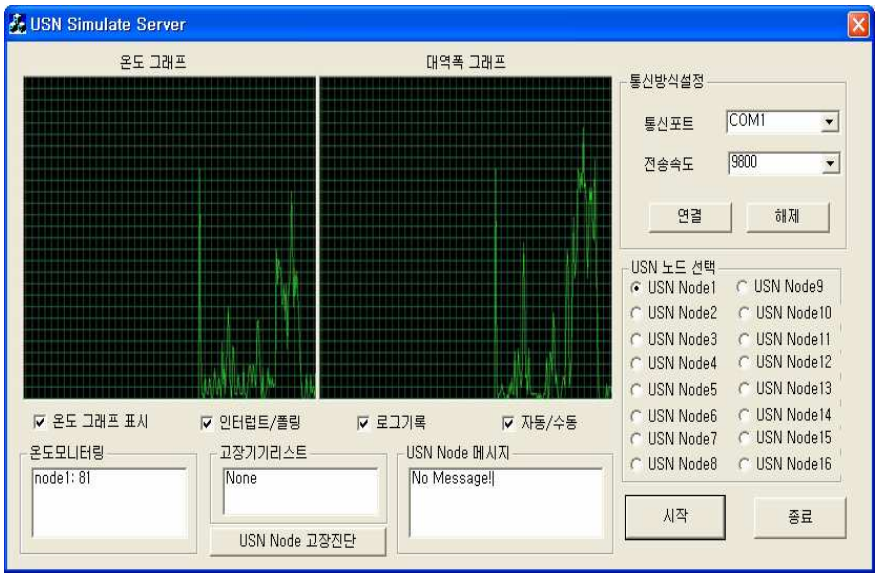



Fig. 5-20 Server Program of USN Simulator

그림 5-21은 PC의 CAN 보드를 통해 감지기 4개의 온도 값을 모니터링 한 결과이다. 감지기 1,2,3은 같은 곳을 가리키고 있으며, 감지기 4는 측정 점에서 약 30[cm]떨어진 곳의 온도를 측정하였다. 측정 방법은 표준 온도측정 패널에 103[℃] 온도를 설정하고 서서히 가열하면서 이때의 온도를 CAN 네트워크를 통해 계속적으로 모니터링 하였다. CAN 방식은 인터럽트 방식대신 PC에서 폴링방식으로 측정하였고, 측정결과 감지기 1,2,3은 거의 1[℃] 미만의 온도 편차 내에서 측정되고 있음을 실험을 통해 확인하였다. 그렇지만 감지기 4는 다른 측정 점을 지시하고 있으므로 별도의 결과를 나타내고 있다.

본 실험에서 사용한 감지기 노드 수는 4개임으로 충분히 예상된 결과를 판단할 수 있으나 차후 노드수가 1000개 이상에서는 폴링 방법으로는 제한됨으로 인터럽트 방식으로 전환해서 실험할 필요가 있다.

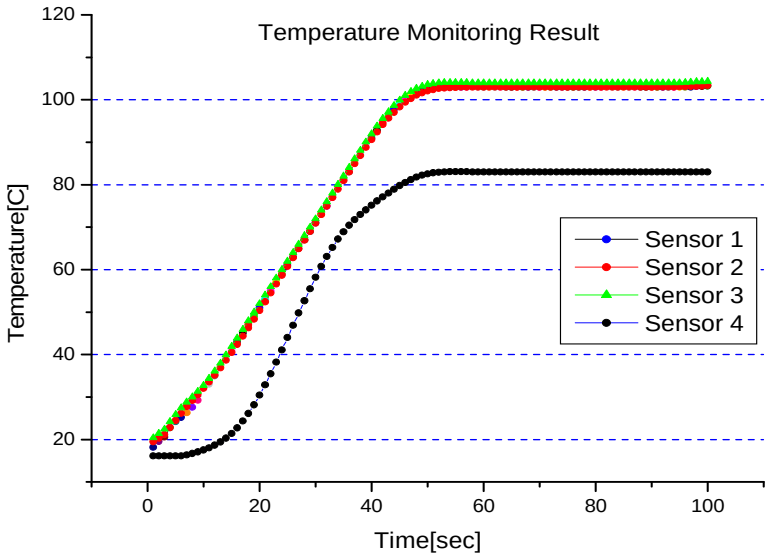



Fig. 5-21 Temperature Monitoring Result

## 3. 품질측정 실험

그림 5-22는 제철 압연공정의 에러율 변화로 적용 전에는 18[%]의 에러율이 12[%]로 감소하여 6[%]의 품질향상을 가져오고 있다. 이와 반면에 타이어 공정에 적용된 시스템은 20[%]에서 8[%]로 12[%] 품질향상을 가져오고 있는데 그 이유는 제철공정에서는 고온에서 미세한 온도조절이 불가능하기 때문에 이런 원인을 가져오는 것으로 분석되었다. 하지만 6[%]의 품질향상은 기존 대비 30[%]로 향상이며, 이는 전체적인 매출에 있어서 엄청난 가치를 동반하고 있다는 사실을 알 수 있다.

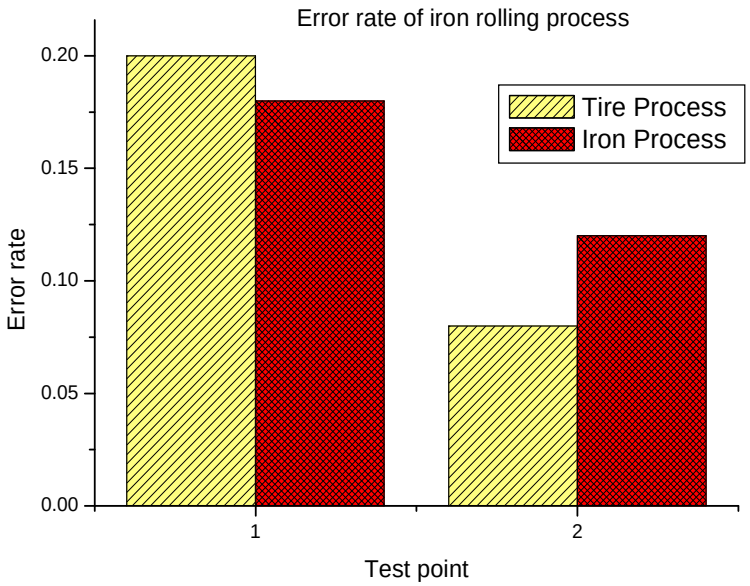



Fig. 5-22 Error Rate of Iron Rolling Process

# *VI.* 결 론

제철 압연공정에서 열연 코일을 생산하는 열간 압연공정에서의 압연 소재 온도는 열연제품의 강도 혹은 가공성 등을 결정하는 중요한 인자이다. 따라서 열간 압연공정에서는 라인 중간에 온도계를 설치하여 목표 온도에 맞도록 공정을 제어하고 있다. 최근에는 열연 제품에 있어서 전체 폭과 전체 길이에 대해 치수 및 기계적 성질이 균일하도록 노력하고 있다. 이를 위하여 제철 압연공정에 사용중인 기존 시스템에서는 균일한 온도를 확보하기 위해 스트립의 폭 방향온도 프로파일을 계측하고 평가하고 있다. 그러나 이 시스템은 정밀한 측정은 가능하지만 프로세스에 물려 자동적으로 온도 값에 따른 보정행위를 하지는 않고 단지 성능평가를 하기 위한 것이다. 따라서 분해능은 다소 떨어지더라도 기존시스템에 물려서 온도에 따른 보정작용을 할 수 있는 시스템이 필요하다.

본 논문에서는 USN 기반 구현을 위해 최근에 알려지고 있는 CAN을 통해 적외선 센서의 온도 값을 모니터링하고, 이를 시스템에 전송하여 온도 조절을 함으로서 양질의 철을 생산할 수 있도록 시스템을 구현하였다. 특히 이 시스템은 단일 서버에서 많은 노드의 클라이언트의 온도 감시를 수행함으로서 효율적인 공정관리를 하도록 지원하는 시스템이다.

실험에서는 폴링 방식과 인터럽트방식의 CAN 네트워크를 사용하여 실험을 수행했으며, 차후 인터럽트 방식으로 1000개 이상의 노드에 적용할 수 있을 것이다. 폴링 방식과 인터럽트 방식의 차이점은 인터럽트 방식은 측정 데이터 값이 이전과 같다면 데이터 전송을 하지 않고 데이터가 상이한 경우에만 인터럽트를 발생시켜 즉시 전송을 수행하여 빠른 처리를 할 수 있지만 만약 동시에 인터럽트가 발생한다면 시스템의 과부하를 유발할

수 있는 문제가 있었다. 하지만 폴링 방식은 서버가 클라이언트를 순서적으로 호출해 가면서 데이터를 취득함으로 데이터량은 많지만 순간적으로 데이터가 증가하지 않고 일정한 통신 대역폭 유지가 가능했으며, 감지기인 클라이언트의 정상 작동여부나 특정 감지기의 온도 값만을 축출하는데 적합하다.

따라서 본 시스템에서는 폴링방식이 적당하였고, 인터럽트방식은 화재감지기등과 같이 인터럽트 발생이 상대적으로 적은 화재감지기 등에 적용하는 것이 더 좋은 결과를 가져옴을 실험데이터의 비교분석결과를 통해 알 수 있었다.

1. 제철공정에서는 100[℃] 이상의 고온을 측정해야 함으로 접촉식 보다는 비접촉식 적외선 온도센서를 사용한다. 이는 접촉식 열전대를 이용해 측정할 경우 장시간 측정할 수 없으며, 정확한 측정을 위해 자주 갈아줘야 하는 문제가 있으며, 간접측정의 경우도 부정확한 측정결과를 가져오기 때문이다. 이를 위해서 지속적으로 온도를 모니터링 해야 하는데 가장 적합한 센서는 적외선 온도계이다. 그러나 일반적인 제품은 핸디형으로 공정 모니터링의 적용을 위해선 특정 네트워크 기반에 묶여 모니터링을 하고, 측정된 온도에 따른 대응모델을 시스템에서 유기적으로 실현할 수 있는 시스템이 필요하다. 따라서 제안된 시스템은 PLC에 직결할 수 있도록 표준 아날로그 출력을 통해 PLC에서 제어 프로그램을 할 수 있도록 설계하였다. USN 네트워크를 구축할 수 있는 기반도 갖추고 있으므로 공정제어에 매우 적합한 구조이다.

2. 감지기의 정확도 향상을 위해 여러 번의 보간 및 실측결과의 비교를 통해 250[℃]에서 ±1[℃]의 오차 범위 안에 들도록 보정이 되었고, 9

차 폴리핏 보정을 통해 기 계산된 결과를 테이블에서 찾도록 하여 응답속도를 1[msec]내에 처리하도록 개발되었다. 특히 제철공정에 맞도록 알맞은 방사율 선택 및 이에 따른 루비코팅렌즈의 직경과 코팅률을 적용한 실험결과를 토대로 최적의 제철공정용 모니터링 시스템으로 조율되어 본래의 적외선 센서의 측정주기가 25[msec] 이지만 임베디드 시스템의 처리속도는 1[msec]이면 계산이 가능함을 확인할 수 있었고, 이는 보다 더 성능을 개선할 수 있는 여지를 남겨놓고 있다.

3. 서버측 CAN 네트워크의 구성은 다중 레이어 방식을 적용하여 2계층으로 분리하여 구성하고, 각 계층은 서버보드를 통해 PC와 인터렉션이 가능하도록 구성되어 졌다. 실험결과에서는 16개 노드를 100[msec]로 감시할 때와 256개의 노드를 1000[msec] 폴링 주기로 감시할 경우 온도 측정시는 폴링모드에서 인터럽트 모드로 전환하여 데이터 요청 및 조작 명령이 증가하나 트래픽은 2.5배 이상 증가하지 않으므로 3배 이상으로 폴링주기를 선택하면 가능할 것으로 실험을 통하여 증명하였다.

4. USN 네트워크 실험에서는 기존에 사용되던 485통신대신에 CAN 시스템을 적용하여 안정성과 처리속도를 대폭 증가시켰으며, 한 노드가 죽더라도 다른 노드는 독립적으로 임무를 수행하며, 서버가 일정한 대역폭을 유지하면서 자기 고장진단 및 온도감시 임무를 수행하며, 기존의 PLC 네트워크와 연동됨으로 효율성을 증가할 수 있었다. 따라서 이 시스템의 적용결과 소결광의 품질이 기존대비 30[%]이상 불량률이 감소되어 생산성 및 품질향상에 기여할 수 있었다.

5. 실험결과 온도의 간접 고속 측정방법을 적용할 수 있는 곳이 타이어 공정 제지공정, 제과 공정 등 온도를 정밀하게 측정해야하는 공정에서는 본 시스템이 효과적일 것이므로 다양한 방사율 조성과 대역폭 조정을 등을 통해 다른 공정에 손쉽게 적용할 수 있는 시스템이다.

이상의 실험에서 확인하였듯이 분해능 높은 센서로 교체시 현재 발생하는 오차를 줄일 수 있음을 확인하였고, 전 측정범위에서 3.75[℃] 각 레인지당 1[℃]이내의 정확도를 지니고 있다. 따라서 좀 더 정확하거나 빠르게 수행을 위해서는 센서의 교체등 약간의 연구로 공정제어 응용할 수 있는 우수한 시스템임으로 향후 제철기술의 발달에 기여하는 시스템임을 확인하였다.

본 실험에서는 CAN 네트워크에서 통신 대역폭과 그 속도 및 안정성에 대해서만 조사하였다. 그러나 또 하나의 큰 특징인 결선량에 대하여 평가하지 못했다. 그 이유는 기존에 설치된 동일한 기능의 시스템이 없어서 평가할 대상이 없기 때문이다. 그러나 CAN 네트워크 이론에서 알 수 있듯이 CAN 네트워크의 적용은 분명 결선량 부분에 있어서도 효과를 가져올 것으로 기대됨으로 설치 공사비도 기존 다른 시스템에 비해서도 효과적 일 것으로 판단한다.

# 참 고 문 헌


[1] 안병준, “열간 사상 압연 공정 시스템의 모델링 및 장력제어”, (工學博士學位論文, 부산대학교), 2005. 8.

[2] Wook-Heon Kim, Seung-Ho Hong, "A Study on the Implementation of CAN in the Distributed Control System of Power Plant", KIEE, 48(6):.760-772, 1999.

[3] Joon-hong Jung, Soo-young Choi, Ki-heon Park, "A Study on the Power System Control and Monitoring Technique Using CAN", KIEE, 52(5):268-276, 2003.

[4] Byong-Hoon Lee, Hong-Ryeol Kim, Dae-Won Kim, "Development of an Extended EDS(Earliest Deadline Scheduling) Algorithm for the CAN-Based Real-Time System", KIEE, 51(7):294-301, 2002.

[5] Keon-Young Yi, "An Implementation of the Position Controller for Multiple Motors Using CAN", KIEE 51(2):55-60, 2002.

[6] Jong-Man Jeon, Dae-Won Kim, "Analysis of Response Characteristics of CAN-Based Feedback Control System Considering Message Time Delays", KIEE, 51(5):.190-196, 2002.

[7] Dae-Won Kim, Hwan-Soo Choi, "Performance Analysis of a Network System using the CAN Protocol", KIEE 51(5):218-225, 2001.

[8] 이준식 외, "적외선 가열로에서 가열되는 유리 패널의 온도분포 해석", 대한기계학회, pp. 278-283, 2003.

[9] 홍원표, "유비쿼터스 컴퓨팅 개념과 센서네트워크", 한국조명·전기설비학회, 제19권 제4호, pp. 4-28, 2005. 8

[10] 이동철 외, "USN 기반의 송전철탑 건전성 감시진단시스템 기본설계", KIEE, 제56P권, 제1호, pp.56-62, 2007. 3.

[11] 문희준, 홍원표, "Smart Building과 Digital Home에서의 유비쿼터스 환경", 한국조명·전기설비학회, 제19권 제4호, pp. 52-71, 2005. 8.

[12] 동성수, 이종호, "센서 네트워크 응용을 위한 무선 통신 모듈 설계 및 구현", ICASE, 12(1):60-65, 2006. 1.

[13] 차현구, 김승우, "유비쿼터스 홈 메스클린업 로봇의 구현에 관한 연구", ICASE, 11(12):1011-1019, 2005. 12.

[14] 이동활, 이만형, 박종진, 김수용, 문영성, "유비쿼터스 컴퓨팅의 실내 측위 기술을 위한 U-SAT 시스템의 적용에 관한 연구", ICASE 12(9):876-882, 2006. 9

[15] 안현식, 송재성, "실내 환경에서의 이동로봇 제어를 위한 유비쿼터스 인터페이스 시스템", ICASE 12(1):66-71, 2006.1.

[16] 김종환 외,“네트워크 환경에서 유비쿼터스 로봇의 구현”, ICASE, 11(12): 1051-1061, 2005. 12.

[17] 장경준, 정용래, 김동욱, 김승우, "유비쿼터스 RCP 상호작용을 위한 다감각 착신기능모듈의 개발", ICASE, 12(1):33-40, 2006. 1.

[18] 권영준, 김대진, 이상완, 변증남, "유비쿼터스 환경을 위한 하프미러형 인터페이스 시스템 개발과 응용", 11(12): 1020-1026, 2005. 12

[19] 심덕선, 양철관, 안규섭, 이준학 , "유비쿼터스 환경에서 개방형 제어 플랫폼에 기반한 무인탐사차량의 재형상 가능 위치제어", ICASE, 11(12): 1002-1010, 2005. 12

[20] 노광현, 이병복, 박애순, 김대식, "LonRF 지능형 디바이스 기반의 유비쿼터스 홈네트워크 테스트베드 개발", ICASE, 10(6):566-573, 2004. 6.

[21] 김영동 외, “유비쿼터스 환경을 위한 소방시스템”, 대한전기학회, 제54P권, 제 3호, pp. 141-147, 2005. 9. 1,

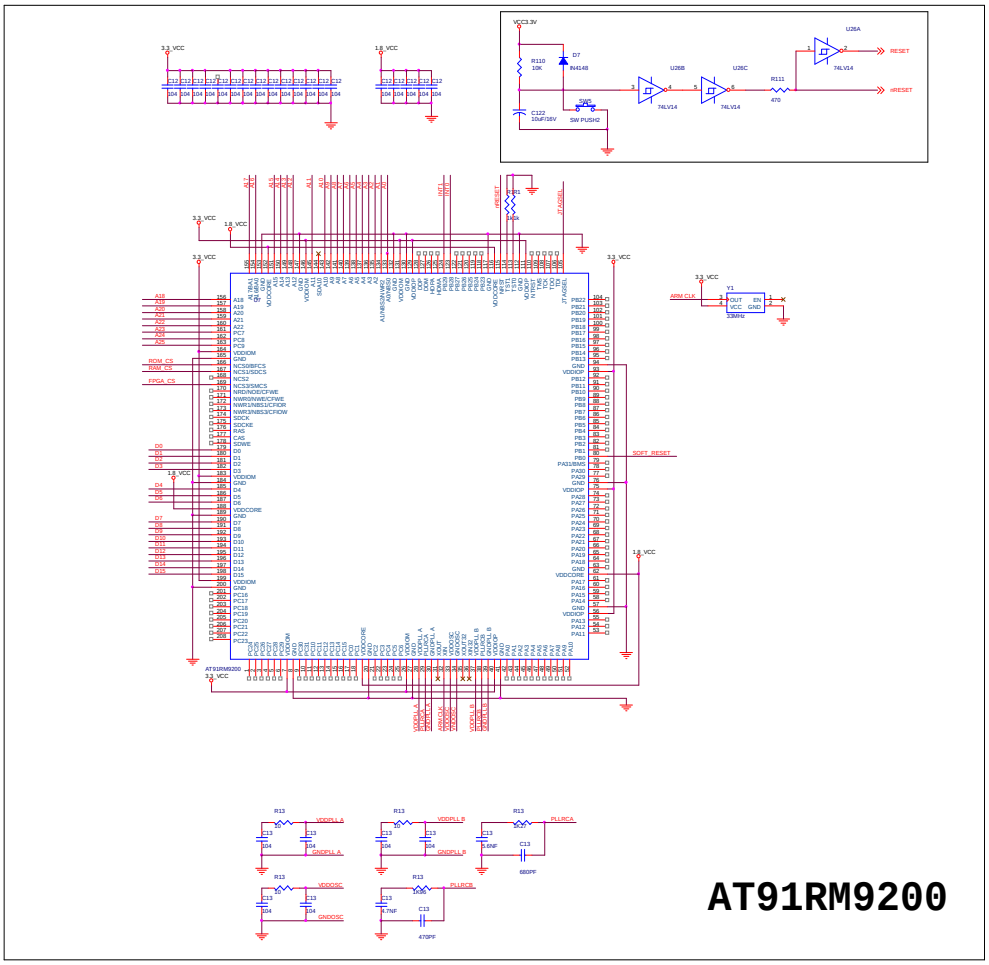
[22] 강원찬 외, “CAN 통신기반 자동 화재탐지설비에 관한 연구” 한국조명·전기설비학회, 제20권 제2호, pp. 50-59 2006. 2.

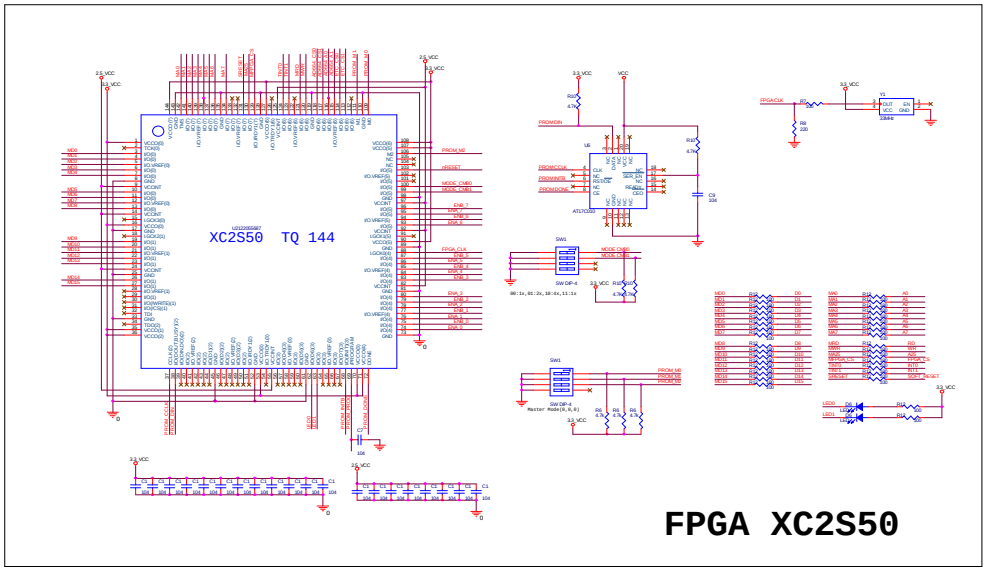
[23] A2TPMI data sheet, Perkin Elmer Inc.
http://www.perkinelmer.com

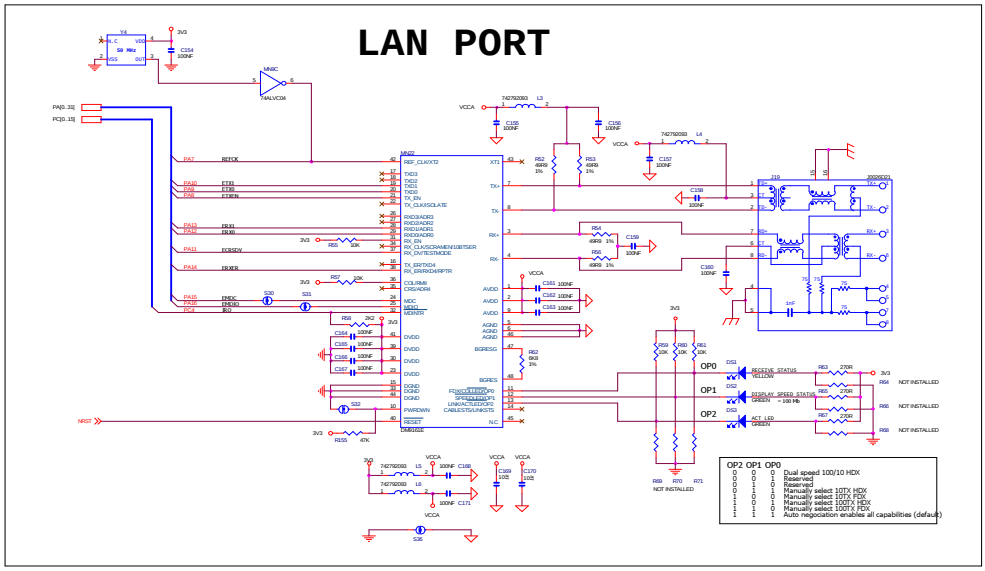
[24] ZF5583 data sheet, Radiant Innovation Inc.
http://www.zytemp.com

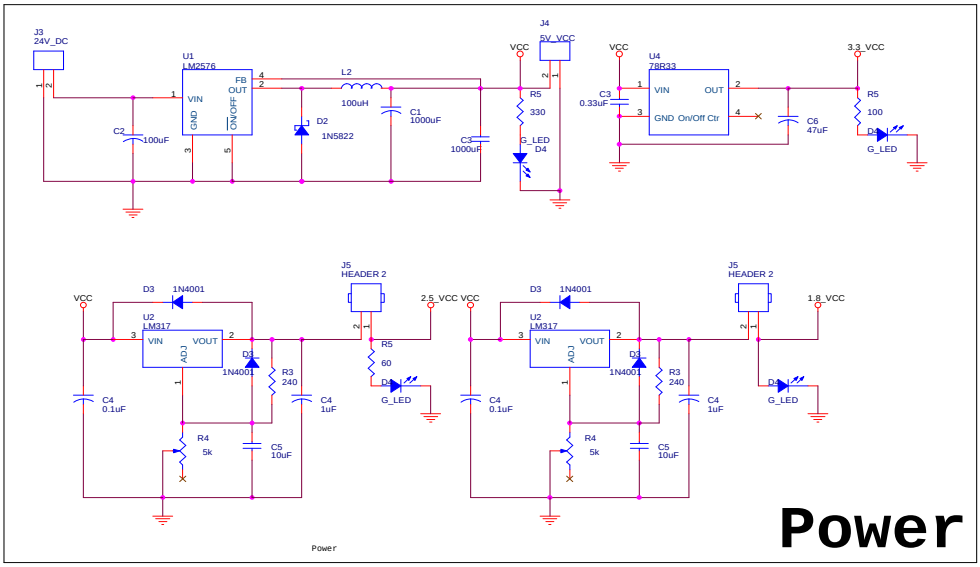
# 부 록

## *1.* 서버 보드 회로도

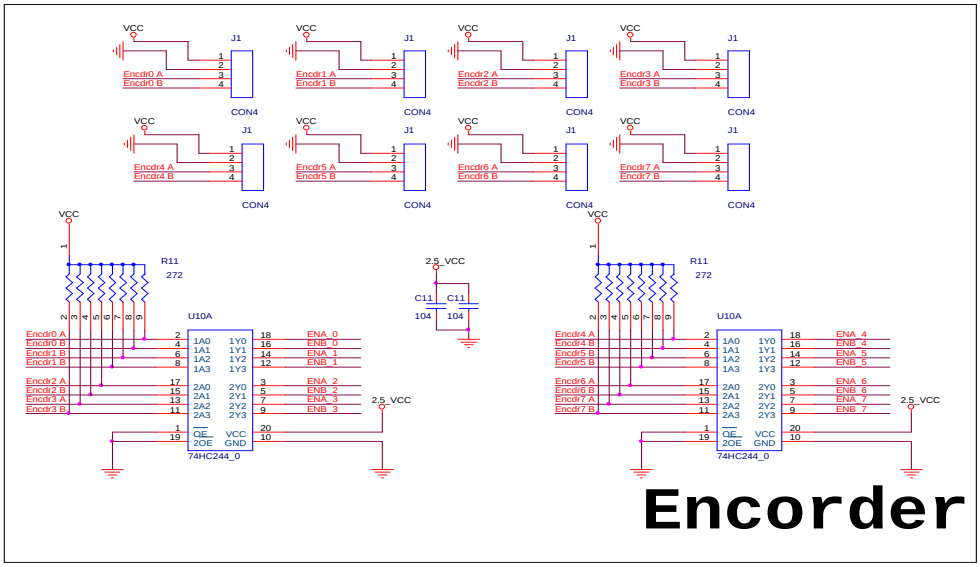

AT91RM9200
RESET
#RESET
SOFT_RESET
ARM CLK
3.3_VCC
1.8_VCC
VCC3.3V
SW PUSH2
74LV14
VDDPLL A
GNDPLL A
VDDPLL B
GNDPLL B
VDDOSC
GNDOSC
PLLRCA
PLLRCB


XC2S50 TQ 144
FPGA XC2S50

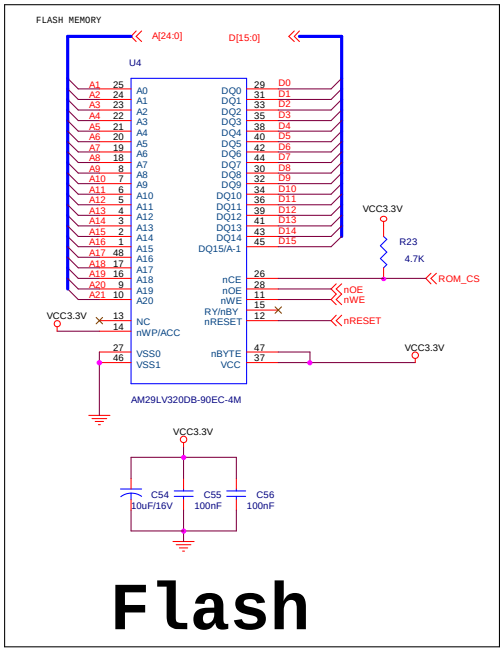

LAN PORT
OP2 OP1 OP0
0 0 0 Dual speed 100/10 HDX
0 0 1 Reserved
0 1 0 Reserved
0 1 1 Manually select 10TX HDX
1 0 0 Manually select 10TX FDX
1 0 1 Manually select 100TX HDX
1 1 0 Manually select 100TX FDX
1 1 1 Auto negotiation enables all capabilities (default)

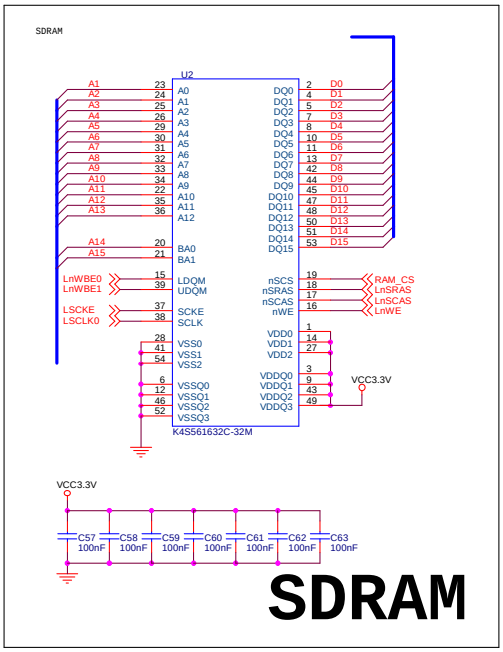



# Power

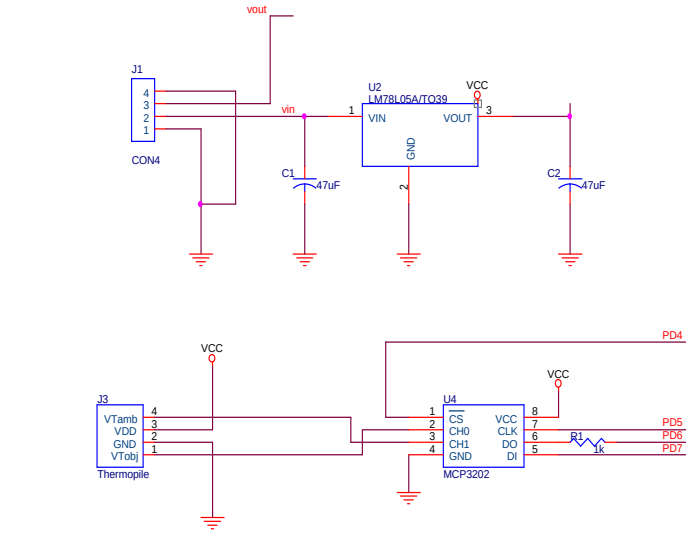


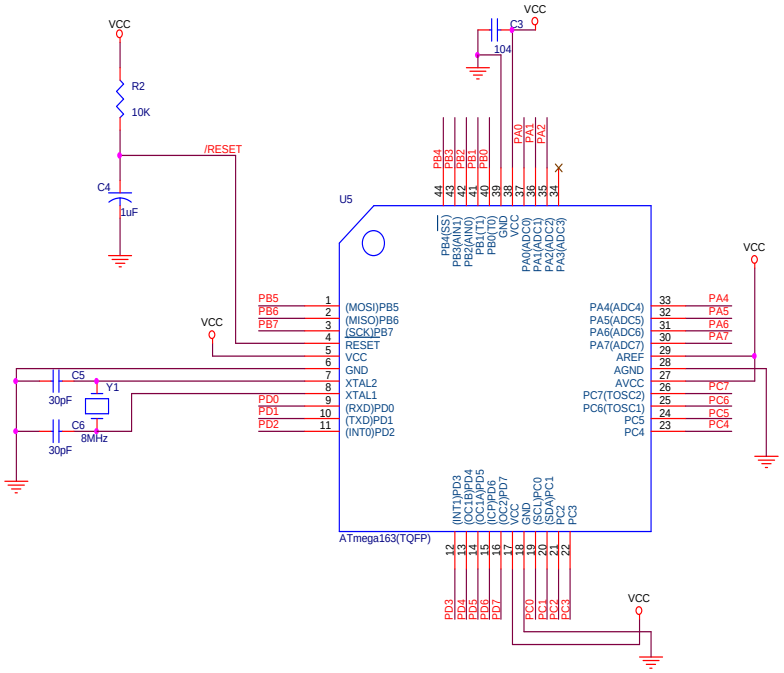

# Encorder

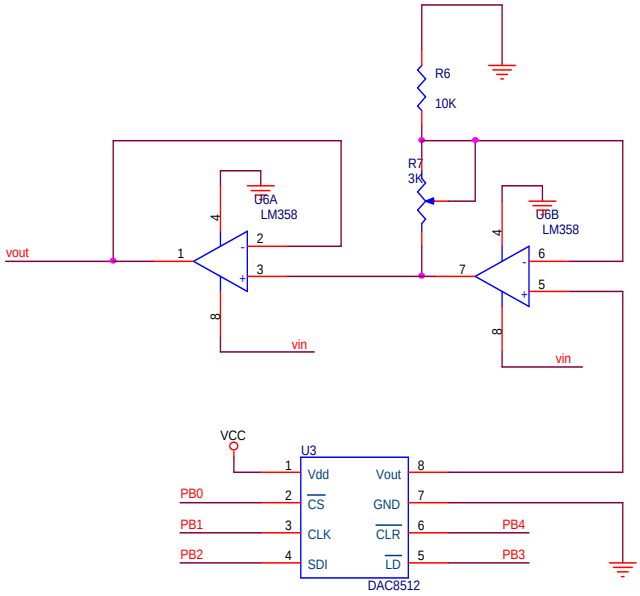

FLASH MEMORY
A[24:0]
D[15:0]
U4
A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13 A14 A15 A16 A17 A18 A19 A20
NC
nWP/ACC
VSS0
VSS1
DQ0 DQ1 DQ2 DQ3 DQ4 DQ5 DQ6 DQ7 DQ8 DQ9 DQ10 DQ11 DQ12 DQ13 DQ14
DQ15/A-1
nCE
nOE
nWE
RY/nBY
nRESET
nBYTE
VCC
D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 D10 D11 D12 D13 D14 D15
VCC3.3V
R23
4.7K
ROM_CS
nOE
nWE
nRESET
AM29LV320DB-90EC-4M
C54
10uF/16V
C55
100nF
C56
100nF
Flash

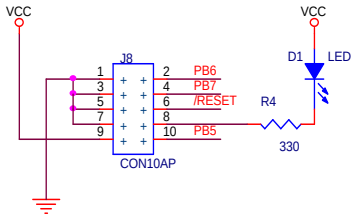

SDRAM
U2
A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12
BA0
BA1
LDQM
UDQM
SCKE
SCLK
VSS0
VSS1
VSS2
VSSQ0
VSSQ1
VSSQ2
VSSQ3
DQ0 DQ1 DQ2 DQ3 DQ4 DQ5 DQ6 DQ7 DQ8 DQ9 DQ10 DQ11 DQ12 DQ13 DQ14 DQ15
nSCS
nSRAS
nSCAS
nWE
VDD0
VDD1
VDD2
VDDQ0
VDDQ1
VDDQ2
VDDQ3
K4S561632C-32M
LnWBE0
LnWBE1
LSCKE
LSCLK0
RAM_CS
LnSRAS
LnSCAS
LnWE
VCC3.3V
C57 100nF
C58 100nF
C59 100nF
C60 100nF
C61 100nF
C62 100nF
C63 100nF
SDRAM

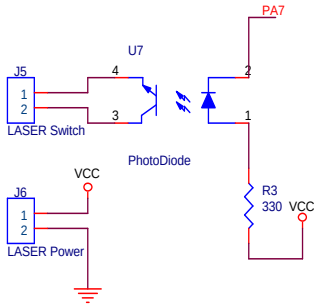
## 2. 감지기 회로도

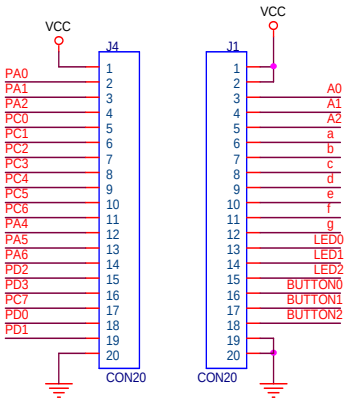

vout
J1
CON4
vin
U2
LM78L05A/TO39
VIN
VOUT
GND
VCC
C1
47uF
C2
47uF
PD4
VCC
J3
VTamb
VDD
GND
VTobj
Thermopile
U4
CS
CH0
CH1
GND
VCC
CLK
DO
DI
MCP3202
R1
1k
PD5
PD6
PD7


VCC
R2
10K
/RESET
C4
1uF
C3
104
U5
PB5
PB6
PB7
(MOSI)PB5
(MISO)PB6
(SCK)PB7
RESET
VCC
GND
XTAL2
XTAL1
(RXD)PD0
(TXD)PD1
(INT0)PD2
PD0
PD1
PD2
C5
30pF
Y1
8MHz
C6
30pF
PA4(ADC4)
PA5(ADC5)
PA6(ADC6)
PA7(ADC7)
AREF
AGND
AVCC
PC7(TOSC2)
PC6(TOSC1)
PC5
PC4
PA4
PA5
PA6
PA7
PC7
PC6
PC5
PC4
ATmega163(TQFP)


R6
10K
R7
3K
U6A
LM358
U6B
LM358
vout
vin
vin
VCC
U3
Vdd
Vout
CS
GND
CLK
CLR
SDI
LD
DAC8512
PB0
PB1
PB2
PB4
PB3


VCC
VCC
J8
PB6
PB7
/RESET
PB5
CON10AP
D1
LED
R4
330


PA7
U7
J5
LASER Switch
PhotoDiode
VCC
J6
LASER Power
R3
330
VCC


VCC
J4
PA0
PA1
PA2
PC0
PC1
PC2
PC3
PC4
PC5
PC6
PA4
PA5
PA6
PD2
PD3
PC7
PD0
PD1
CON20
VCC
J1
A0
A1
A2
a
b
c
d
e
f
g
LED0
LED1
LED2
BUTTON0
BUTTON1
BUTTON2
CON20

### *3.* 감지기 프로그램

```
/*****************************************************
This program was produced by the
CodeWizardAVR V1.24.5 Standard

Project : IR Detector
Version : v.1.1
Date    : 2007-11-22

Chip type           : ATmega163L
Program type        : Application
Clock frequency     : 8.000000 MHz
Memory model        : Small
External SRAM size  : 0
Data Stack size     : 256
*****************************************************/
#include <mega16.h>
#include <delay.h>
#include <math.h>
#include <stdio.h>
// Declare your global variables here
#define SEG_A0          PORTA.1
#define SEG_A1          PORTA.2
#define SEG_A2          PORTC.0
#define SEG_A           PORTC.1
#define SEG_B           PORTC.2
#define SEG_C           PORTC.3
#define SEG_D           PORTC.4
#define SEG_E           PORTC.5
#define SEG_F           PORTC.6
#define SEG_G           PORTA.4
#define LED_TOP         PORTA.5
#define LED_MIDDLE      PORTA.6
#define LED_BOTTOM      PORTD.2
#define BUTTON_LEFT     PIND.0
#define BUTTON_RIGHT    PIND.3
#define BUTTON_MIDDLE   PINC.7
#define LASER_PORT      PORTA.7
#define CS3202          PORTD.4
#define CLK3202         PORTD.5
```

```
#define DO3202          PIND.6
#define DI3202          PORTD.7
#define CS8512       PORTB.0
#define CLK8512      PORTB.1
#define SDI8512      PORTB.2
#define CLR8512      PORTB.4
#define LD8512       PORTB.3
#define MODE_TEMP    0
#define MODE_EMIT    1
#define TEMP_LOW     -20
#define TEMP_HIGH    250
#define DA_OUTPUT_LOW        0
#define DA_OUTPUT_HIGH       1
#define DA_OUTPUT_NORMAL     2

int dispData = 0;
int dispEmit = 0;
long timeTick ;
long laserTimeTick ;
unsigned char curMode ;
eeprom unsigned char emittivity = 100;
unsigned int delayTime = 1000;
unsigned int laserDelayTime = 4000 ;
float Utp ;
float sensitivity ;
unsigned char chkDAOutMode ;
unsigned char chkLaser = 0;
unsigned int tempTable[300] ;

void initTable(void)
{
    int i ;
        tempTable[203] = 2041;
        tempTable[202] = 2031;
        tempTable[201] = 2017;
        tempTable[200] = 2006;
        tempTable[199] = 1996;
        tempTable[198] = 1988;
        tempTable[197] = 1978;
        tempTable[196] = 1969;
        tempTable[195] = 1963;
        tempTable[194] = 1952;
```

```
tempTable[193] = 1939;
tempTable[192] = 1931;
tempTable[191] = 1923;
tempTable[190] = 1916;
tempTable[189] = 1906;
tempTable[188] = 1896;
tempTable[187] = 1889;
tempTable[186] = 1880;
tempTable[185] = 1872;
tempTable[184] = 1863;
tempTable[183] = 1853;
tempTable[182] = 1842;
tempTable[181] = 1832;
tempTable[180] = 1822;
tempTable[179] = 1813;
tempTable[178] = 1805;
tempTable[177] = 1793;
tempTable[176] = 1788;
tempTable[175] = 1780;
tempTable[174] = 1772;
tempTable[173] = 1762;
tempTable[172] = 1750;
tempTable[171] = 1741;
tempTable[170] = 1731;
tempTable[169] = 1722;
tempTable[168] = 1714;
tempTable[167] = 1708;
tempTable[166] = 1699;
tempTable[165] = 1690;
tempTable[164] = 1682;
tempTable[163] = 1674;
tempTable[162] = 1664;
tempTable[161] = 1656;
tempTable[160] = 1648;
tempTable[159] = 1640;
tempTable[158] = 1636;
tempTable[157] = 1628;
tempTable[156] = 1620;
tempTable[155] = 1610;
tempTable[154] = 1604;
tempTable[153] = 1595;
tempTable[152] = 1590;
```

```
tempTable[151] = 1583;
tempTable[150] = 1574;
tempTable[149] = 1568;
tempTable[148] = 1562;
tempTable[147] = 1555;
tempTable[146] = 1547;
tempTable[145] = 1539;
tempTable[144] = 1531;
tempTable[143] = 1523;
tempTable[142] = 1517;
tempTable[141] = 1510;
tempTable[140] = 1504;
tempTable[139] = 1494;
tempTable[138] = 1488;
tempTable[137] = 1481;
tempTable[136] = 1474;
tempTable[135] = 1468;
tempTable[134] = 1460;
tempTable[133] = 1455;
tempTable[132] = 1449;
tempTable[131] = 1440;
tempTable[130] = 1432;
tempTable[129] = 1425;
tempTable[128] = 1422;
tempTable[127] = 1418;
tempTable[126] = 1413;
tempTable[125] = 1402;
tempTable[124] = 1394;
tempTable[123] = 1387;
tempTable[122] = 1380;
tempTable[121] = 1369;
tempTable[120] = 1369;
tempTable[119] = 1362;
tempTable[118] = 1356;
tempTable[117] = 1352;
tempTable[116] = 1346;
tempTable[115] = 1337;
tempTable[114] = 1331;
tempTable[113] = 1325;
tempTable[112] = 1318;
tempTable[111] = 1312;
tempTable[110] = 1305;
```

```
tempTable[109] = 1299;
tempTable[108] = 1294;
tempTable[107] = 1288;
tempTable[106] = 1284;
tempTable[105] = 1280;
tempTable[104] = 1274;
tempTable[103] = 1270;
tempTable[102] = 1263;
tempTable[101] = 1260;
tempTable[100] = 1253;
tempTable[99] = 1249;
tempTable[98] = 1244;
tempTable[97] = 1239;
tempTable[96] = 1234;
tempTable[95] = 1234;
tempTable[94] = 1224;
tempTable[93] = 1217;
tempTable[92] = 1213;
tempTable[91] = 1205;
tempTable[90] = 1199;
tempTable[89] = 1196;
tempTable[88] = 1189;
tempTable[87] = 1186;
tempTable[86] = 1182;
tempTable[85] = 1176;
tempTable[84] = 1171;
tempTable[83] = 1167;
tempTable[82] = 1163;
tempTable[81] = 1157;
tempTable[80] = 1154;
tempTable[79] = 1148;
tempTable[78] = 1142;
tempTable[77] = 1138;
tempTable[76] = 1135;
tempTable[75] = 1130;
tempTable[74] = 1125;
tempTable[73] = 1120;
tempTable[72] = 1114;
tempTable[71] = 1111;
tempTable[70] = 1107;
tempTable[69] = 1102;
tempTable[68] = 1099;
```

```
        tempTable[67] = 1095;
        tempTable[66] = 1092;
        tempTable[65] = 1088;
        tempTable[64] = 1082;
        tempTable[63] = 1078;
        tempTable[62] = 1076;
        tempTable[61] = 1072;
        tempTable[60] = 1069;
        tempTable[59] = 1068;
        tempTable[58] = 1062;
        tempTable[57] = 1058;
        tempTable[56] = 1056;
        tempTable[55] = 1051;
        tempTable[54] = 1047;
        tempTable[53] = 1047;
        tempTable[52] = 1046;
        tempTable[51] = 1043;
        tempTable[50] = 1036;
        tempTable[49] = 1032;
        tempTable[48] = 1030;
        tempTable[47] = 1028;
        tempTable[46] = 1025;
        tempTable[45] = 1023;
        tempTable[45] = 1023;

        for(i = 204 ; i < 300 ; i ++ )
        {
                tempTable[i] = tempTable[i-1] + 10;
        }
        for( i = 44 ; i >= 0 ; i-- )
        {
                tempTable[i] = tempTable[i+1] - 1 ;
        }
}

void out8512( unsigned int daData )
{
    LD8512 = 1 ;
    CS8512 = 0 ;
    CLK8512 = 0 ;

    if( daData >= 2048 ) SDI8512 = 1 ;
```

```
else SDI8512 = 0 ;
daData = daData % 2048 ;
delay_us(1) ;    CLK8512 = 1 ;    delay_us(1) ;    CLK8512 = 0 ;

if( daData >= 1024 ) SDI8512 = 1 ;
else SDI8512 = 0 ;
daData = daData % 1024 ;
delay_us(1) ;    CLK8512 = 1 ;    delay_us(1) ;    CLK8512 = 0 ;

if( daData >= 512 ) SDI8512 = 1 ;
else SDI8512 = 0 ;
daData = daData % 512 ;
delay_us(1) ;    CLK8512 = 1 ;    delay_us(1) ;    CLK8512 = 0 ;

if( daData >= 256 ) SDI8512 = 1 ;
else SDI8512 = 0 ;
daData = daData % 256 ;
delay_us(1) ;    CLK8512 = 1 ;    delay_us(1) ;    CLK8512 = 0 ;

if( daData >= 128 ) SDI8512 = 1 ;
else SDI8512 = 0 ;
daData = daData % 128 ;
delay_us(1) ;    CLK8512 = 1 ;    delay_us(1) ;    CLK8512 = 0 ;

if( daData >= 64 ) SDI8512 = 1 ;
else SDI8512 = 0 ;
daData = daData % 64 ;
delay_us(5) ;    CLK8512 = 1 ;    delay_us(1) ;    CLK8512 = 0 ;

if( daData >= 32 ) SDI8512 = 1 ;
else SDI8512 = 0 ;
daData = daData % 32 ;
delay_us(1) ;   CLK8512 = 1 ;    delay_us(1) ;    CLK8512 = 0 ;

if( daData >= 16 ) SDI8512 = 1 ;
else SDI8512 = 0 ;
daData = daData % 16 ;
delay_us(1) ;    CLK8512 = 1 ;    delay_us(1) ;    CLK8512 = 0 ;

if( daData >= 8 ) SDI8512 = 1 ;
else SDI8512 = 0 ;
daData = daData % 8 ;
```

```
    delay_us(1) ;    CLK8512 = 1 ;    delay_us(1) ;    CLK8512 = 0 ;

    if( daData >= 4 ) SDI8512 = 1 ;
    else SDI8512 = 0 ;
    daData = daData % 4 ;
    delay_us(1) ;    CLK8512 = 1 ;    delay_us(1) ;    CLK8512 = 0 ;

    if( daData >= 2 ) SDI8512 = 1 ;
    else SDI8512 = 0 ;
    daData = daData % 2 ;
    delay_us(1) ;    CLK8512 = 1 ;    delay_us(1) ;    CLK8512 = 0 ;
    if( daData >= 1 ) SDI8512 = 1 ;
    else SDI8512 = 0 ;
    daData = daData % 1 ;
    delay_us(1) ;    CLK8512 = 1 ;  delay_us(1) ;

    CS8512 = 1 ;
    delay_us(1);
    LD8512 = 0 ;
    delay_us(1) ;
    LD8512 = 1 ;
}

void activate3202(void)
{
        CS3202 = 0 ;   // cs enable....
        CLK3202 = 0 ; // init clock...
}

void deactivate3202(void)
{
        CS3202 = 1 ;   // cs enable....
}

void adcSendData( int channel )
{
        switch(channel)
        {
                case 0:
                        DI3202 = 1 ;   //          DI          output...../
startbit....command..
                        CLK3202 = 1 ; // clock out
```

```
                        CLK3202 = 0 ;

                        DI3202 = 1 ;   // command mux address 1 //
                        CLK3202 = 1 ; // clock out ...
                        CLK3202 = 0 ;

                        DI3202 = 0 ;   // command mux address 2 //
                        CLK3202 = 1 ;  // clock out
                        CLK3202 = 0 ;
                        // select channel 0 .....AD converter...
                break ;
                case 1:
                        DI3202 = 1 ;   //          DI          output...../
startbit....command..
                        CLK3202 = 1 ; // clock out
                        CLK3202 = 0 ;

                        DI3202 = 1 ;   // command mux address 1 //
                        CLK3202 = 1 ; // clock out ...
                        CLK3202 = 0 ;

                        DI3202 = 1 ;   // command mux address 2 //
                        CLK3202 = 1 ;  // clock out
                        CLK3202 = 0 ;
                        // select channel 1 .....AD converter...
                break ;
        }
}

unsigned int adcReadData(void)
{
        unsigned int outData = 0 ;
        int i ;

        CLK3202 = 1 ;  // clock out
    CLK3202 = 0 ;
        CLK3202 = 1 ;

        for( i = 0 ; i < 12 ; i++ )
        {
                CLK3202 = 0 ;
                outData = (outData << 1) + (int)DO3202 ;
```

```
                CLK3202 = 1 ;
        }

        delay_ms(1);
        return outData ;
}

void setNumber( unsigned char num )
{
    if( num == 0 )
    {
        SEG_A = 0 ; SEG_B = 0 ; SEG_C = 0 ; SEG_D = 0 ; SEG_E =
0 ; SEG_F = 0 ; SEG_G = 1 ;         // 0
    }
    else if( num == 1 )
    {
        SEG_A = 1 ; SEG_B = 0 ; SEG_C = 0 ; SEG_D = 1 ; SEG_E =
1 ; SEG_F = 1 ; SEG_G = 1 ;         // 1
    }
    else if( num == 2 )
    {
        SEG_A = 0 ; SEG_B = 0 ; SEG_C = 1 ; SEG_D = 0 ; SEG_E =
0 ; SEG_F = 1 ; SEG_G = 0 ;         // 2
    }
    else if( num == 3 )
    {
        SEG_A = 0 ; SEG_B = 0 ; SEG_C = 0 ; SEG_D = 0 ; SEG_E =
1 ; SEG_F = 1 ; SEG_G = 0 ;         // 3
    }
    else if( num == 4 )
    {
        SEG_A = 1 ; SEG_B = 0 ; SEG_C = 0 ; SEG_D = 1 ; SEG_E =
1 ; SEG_F = 0 ; SEG_G = 0 ;         // 4
    }
    else if( num == 5 )
    {
        SEG_A = 0 ; SEG_B = 1 ; SEG_C = 0 ; SEG_D = 0 ; SEG_E =
1 ; SEG_F = 0 ; SEG_G = 0 ;         // 5
    }
    else if( num == 6 )
    {
        SEG_A = 0 ; SEG_B = 1 ; SEG_C = 0 ; SEG_D = 0 ; SEG_E =
```

```
0 ; SEG_F = 0 ; SEG_G = 0 ;          // 6
    }
    else if( num == 7 )
    {
        SEG_A = 0 ; SEG_B = 0 ; SEG_C = 0 ; SEG_D = 1 ; SEG_E =
1 ; SEG_F = 1 ; SEG_G = 1 ;          // 7
    }
    else if( num == 8 )
    {
        SEG_A = 0 ; SEG_B = 0 ; SEG_C = 0 ; SEG_D = 0 ; SEG_E =
0 ; SEG_F = 0 ; SEG_G = 0 ;          // 8
    }
    else if( num == 9 )
    {
        SEG_A = 0 ; SEG_B = 0 ; SEG_C = 0 ; SEG_D = 0 ; SEG_E =
1 ; SEG_F = 0 ; SEG_G = 0 ;          // 9
    }
    else if(num == '-')
    {
        SEG_A = 1 ; SEG_B = 1 ; SEG_C = 1 ; SEG_D = 1 ; SEG_E =
1 ; SEG_F = 1 ; SEG_G = 0 ;          // -
    }
    else if(num == '_')
    {
        SEG_A = 0 ; SEG_B = 1 ; SEG_C = 1 ; SEG_D = 0 ; SEG_E =
1 ; SEG_F = 1 ; SEG_G = 0 ;          // _
    }
    else if(num == 'C')
    {
        SEG_A = 0 ; SEG_B = 1 ; SEG_C = 1 ; SEG_D = 0 ; SEG_E =
0 ; SEG_F = 0 ; SEG_G = 1 ;          // C
    }
    else if(num == 'H')
    {
        SEG_A = 1 ; SEG_B = 0 ; SEG_C = 0 ; SEG_D = 1 ; SEG_E =
0 ; SEG_F = 0 ; SEG_G = 0 ;          // H
    }
    else if(num == 'L')
    {
        SEG_A = 1 ; SEG_B = 1 ; SEG_C = 1 ; SEG_D = 0 ; SEG_E =
0 ; SEG_F = 0 ; SEG_G = 1 ;          // L
    }
```

```
    else
    {
        SEG_A = 1 ; SEG_B = 1 ; SEG_C = 1 ; SEG_D = 1 ; SEG_E =
1 ; SEG_F = 1 ; SEG_G = 1 ;          // all leds are turned off
    }
}

void putNumber( unsigned char address, unsigned char num)
{
    if( address == 0 )
    {
        SEG_A0 = 1 ; SEG_A1 = 0 ; SEG_A2 = 0 ;
        setNumber(num) ;
    }
    else if( address == 1 )
    {
        SEG_A0 = 0 ; SEG_A1 = 1 ; SEG_A2 = 0 ;
        setNumber(num) ;
    }
    else if( address == 2 )
    {
        SEG_A0 = 0 ; SEG_A1 = 0 ; SEG_A2 = 1 ;
        setNumber(num) ;
    }
    else
    {
        SEG_A0 = 0 ; SEG_A1 = 0 ; SEG_A2 = 0 ;  // no action.
    }
    delay_us(500) ;
}

void putSegData( int segData )
{
    unsigned char chk100s ;

    if(chkDAOutMode == DA_OUTPUT_HIGH)
    {
        putNumber(0, 'C');
        putNumber(1, '_');
        putNumber(2, 'H');
    }
    else if(chkDAOutMode == DA_OUTPUT_LOW)
```

```
    {
        putNumber(0, 'C');
        putNumber(1, '_');
        putNumber(2, 'L');
    }
    else
    {
        if( (segData >= 0) && ( segData < 1000) )         // if the data
is vaild...to display..
        {
            if( segData >= 100 )
            {
                putNumber(0, (segData / 100));
                chk100s = 1 ;
            }
            else
            {
                //putNumber(0, -1);    // turn off..LEDs
                putNumber(0, 0);    // turn off..LEDs
                chk100s = 0 ;
            }
            segData = segData % 100 ;

            if( (segData >= 10) || (chk100s == 1) )
            {
                putNumber(1, (segData / 10));
            }
            else
            {
                //putNumber(1, -1);    // turn off..LEDs
                putNumber(1, 0);    // turn off..LEDs
            }

            segData = segData % 10 ;

            putNumber(2, segData);
        }
        else if( segData < 0 )
        {
            putNumber(0, '-');
            segData = (-1*segData) % 100 ;
```

```
            if( (segData >= 10) || (chk100s == 1) )
            {
                putNumber(1, (segData / 10));
            }
            else
            {
                //putNumber(1, -1);    // turn off..LEDs
                putNumber(1, 0);    // turn off..LEDs
            }

            segData = segData % 10 ;

            putNumber(2, segData);

  } }}

// Timer 0 overflow interrupt service routine
interrupt [TIM0_OVF] void timer0_ovf_isr(void)
{
    // Place your code here
    if(chkDAOutMode == DA_OUTPUT_NORMAL)
    {
        if( curMode == MODE_TEMP )
        {
            putSegData(dispData);
            LED_TOP = 0;              // green => top
            LED_MIDDLE = 1;            // Yellow => middle
            timeTick = 0 ;
        }
        else
        {
            putSegData(dispEmit) ;
            LED_TOP = 1;              // green => top
            LED_MIDDLE = 0;            // Yellow => middle

            timeTick++ ;

            if(timeTick > delayTime)
            {
                curMode = MODE_TEMP ;
            }
```

```
        }
    }
    else
    {
        putSegData(dispData);
        LED_TOP = 1;              // green => top
        LED_MIDDLE = 1;            // Yellow => middle
    }
    laserTimeTick++ ;
    if( laserTimeTick > laserDelayTime )
    {
        chkLaser = 0 ;
        laserTimeTick = 0 ;
    }

}

// Declare your global variables here
void main(void)
{
    // Declare your local variables here
    unsigned int SenData ;
    unsigned int SenAmbData ;
    unsigned char chkLeftButton = 0;
    unsigned char chkMiddleButton = 0;

    float SenVoltage ;
    float SenAmbVoltage ;
    float tempObj ;
    float tempAmb ;
    float tempData ;
    int i, j ;
    unsigned int bufTemp1 = 0 , bufTemp2 = 0;
    unsigned int ratio1, ratio2 ;

    // Input/Output Ports initialization
    // Port A initialization
    PORTA=0x00;
    DDRA=0xFF;

    // Port B initialization
    PORTB=0x00;
```

```
DDRB=0xFF;

// Port C initialization
PORTC=0x00;
DDRC=0x7F;
// Port D initialization
PORTD=0x00;
DDRD=0xB4;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: 125.000 kHz
// Mode: Normal top=FFh
// OC0 output: Disconnected
TCCR0=0x03;
TCNT0=0x00;
OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer 1 Stopped
// Mode: Normal top=FFFFh
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer 2 Stopped
// Mode: Normal top=FFh
// OC2 output: Disconnected
```

```
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
MCUCR=0x00;
MCUCSR=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x01;

// usart 9600
UCSRA=0x00;
UCSRB=0x08;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x33;
// usart 9600

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;

// Global enable interrupts
#asm("sei")

// Init
LED_TOP = 1;              // green => top
LED_MIDDLE = 1;           // Yellow => middle
LED_BOTTOM = 1 ;          // red => bottom

SDI8512 = 0;
CLK8512 = 1 ;
CS8512 = 1 ;
LD8512 = 0 ;
CLR8512 = 1 ;
```

```
curMode = MODE_TEMP ;
dispEmit = emittivity ;
chkDAOutMode = DA_OUTPUT_NORMAL;
DDRB=0xFF;

initTable();
// Init

while (1)
{
    // Place your code here
    activate3202() ;          // activate 3202 chip
    adcSendData(0) ;                  // sensor Data
    SenData = adcReadData() ;      // read data..
    deactivate3202();
    activate3202() ;               // activate 3202 chip
    adcSendData(1) ;                      // sensor Data
    SenAmbData = adcReadData() ;// read data..
    deactivate3202();
    // read sensor data...
    for( i = 0 ; i < 5 ; i++ )
    {
            activate3202() ;          // activate 3202 chip
            adcSendData(0) ;                  // sensor Data
            SenData += adcReadData() ;    // read data..
            SenData /= 2 ;
            deactivate3202();

            activate3202() ;               // activate 3202 chip
            adcSendData(1) ;                      // sensor Data
            SenAmbData += adcReadData() ;        // read data..
            SenAmbData /= 2 ;
            deactivate3202();
            // read sensor data...
    }
    // temp Calculate Function

    SenVoltage = SenData * 5.0f / 4096.0f ;
    SenAmbVoltage = SenAmbData * 5.0f / 4096.0f ;

    // outVoltage = sensorGane*SenVoltage - SenAmbVoltage ;
```

```
        // tempObj = -0.730674*pow(SenVoltage, 6) + 13.29956*pow(SenVoltage, 5) -99.01311*pow(SenVoltage, 4) + 387.87254*pow(SenVoltage, 3) - 856.4696*pow(SenVoltage, 2) + 1095.76*pow(SenVoltage,1) - 556.58 ;
        // tempObj = 33.315249 + 0.072572527*SenData - 84200873/pow(SenData, 2) + 1.5;
        // tempObj = -0.730674*pow(outVoltage, 6) + 13.29956*pow(outVoltage, 5) -99.01311*pow(outVoltage, 4) + 387.87254*pow(outVoltage, 3) - 856.4696*pow(outVoltage, 2) + 1095.76*pow(outVoltage,1) - 556.58 ;
        tempAmb = (-100.525 + 123.8447*SenAmbVoltage)/(1+(0.962596*SenAmbVoltage) + (-0.08234*SenAmbVoltage*SenAmbVoltage));

        //tempObj = (-733.03437 + (669.61087*SenVoltage)) / ( 1 + (2.3187595 * SenVoltage) - (0.16*pow(SenVoltage, 2))) ;
        //tempObj = (17831283-20484.481*SenData)/(1-70.115181*SenData+0.005879173*SenData*SenData);

        // tempData = (( 100.0f/ emittivity)*(tempObj - tempAmb) + tempAmb);
        // tempData = ((100.0f/ emittivity)*(tempObj - tempAmb)+ tempAmb);
        // tempData = 5.253 * pow(1.0098,tempData)*pow(tempData, 0.433) ;

        // dispData = (unsigned int)tempData ;
        //dispData = tempData;
        //dispData = (unsigned int)(SenData%1000) ;
        //dispData = (unsigned int)(SenData/10) ;
        for( i = 0; i < 300 ; i++ )
        {
            if( tempTable[i] <= SenData )
                {
                        bufTemp1 = i ;
                        ratio1 = SenData - tempTable[i];
                }
        }

        for( j = 299 ; j >= 0 ; j-- )
        {
```

```
                if( tempTable[j] >= SenData )
                {
                        bufTemp2 = j ;
                        ratio2 = SenData - tempTable[j] ;
                }
        }
        if( bufTemp1 != bufTemp2 )
        {
                tempObj          =          (float)(((bufTemp2          -
bufTemp1)*ratio1)/(float)(ratio1 - ratio2) + bufTemp1) ;
        }
        else
        {
                tempObj = (float)bufTemp1 ;
        }


        dispData = (unsigned int)(tempObj) ;
        // printf("%4d\r", SenData);

        if(chkDAOutMode == DA_OUTPUT_HIGH)
        {
            out8512(4095) ;
        }
        else if(chkDAOutMode == DA_OUTPUT_LOW)
        {
            out8512(0) ;
        }
        else
        {
            out8512((unsigned int)(tempData*15.16667 + 303.3333)) ;
        }
        // temp Calculate Function

        // up/down button function...
        if(( BUTTON_LEFT == 0 ) && (BUTTON_MIDDLE == 0) &&
(BUTTON_RIGHT == 0))
        {
            // 조절 모드..
            if(chkDAOutMode == DA_OUTPUT_HIGH )
            {
                chkDAOutMode = DA_OUTPUT_LOW ;
```

```
        }
        else if(chkDAOutMode == DA_OUTPUT_LOW )
        {
            chkDAOutMode = DA_OUTPUT_NORMAL  ;
        }
        else
        {
            chkDAOutMode = DA_OUTPUT_HIGH ;
        }

        delay_ms(50);
        chkLaser = 0 ;
        LED_BOTTOM = 1 ;
        LASER_PORT = 1 ;
    }

    if( BUTTON_LEFT == 0 )
    {

        curMode = MODE_EMIT ;
        emittivity++ ;
        if(emittivity > 100 )
        {
            emittivity = 100 ;
        }
        dispEmit = emittivity ;

        chkLeftButton++ ;

        if( chkLeftButton < 5 )
        {
            delay_ms(50);
        }

        timeTick = 0 ;
    }
    else
    {
        chkLeftButton = 0 ;
    }

    if( BUTTON_MIDDLE == 0 )
```

```
{
    curMode = MODE_EMIT ;
    if(emittivity != 0 )
    {
        emittivity-- ;
    }
    dispEmit = emittivity ;

    chkMiddleButton++ ;

    if( chkMiddleButton < 5 )
    {
        delay_ms(50);
    }
    timeTick = 0 ;
}
else
{
    chkMiddleButton = 0 ;
}
// up/down button function...

// laser pointer function...

if(chkDAOutMode == DA_OUTPUT_NORMAL )
{
    if((BUTTON_RIGHT == 0) && ( chkLaser == 0 ))
    {
        chkLaser = 1 ;
        laserTimeTick = 0 ;
        delay_ms(50);
    }
    else if((BUTTON_RIGHT == 0) && ( chkLaser == 1 ))
    {
        chkLaser = 0 ;
        laserTimeTick = 0 ;
        delay_ms(50);
    }

    if( chkLaser )
    {
        LED_BOTTOM = 0 ;        // red => bottom
```

```
                LASER_PORT = 0 ;
            }
            else
            {
                LED_BOTTOM = 1 ;        // red => bottom
                LASER_PORT = 1 ;
            }
        }
        else
        {
            LED_BOTTOM = 1 ;
            LASER_PORT = 1 ;
        }
        // laser pointer function...
    };
}
```

# 저작물 이용 허락서

| 학 과 | 전기공학과 | 학 번 | 20047366 | 과 정 | 박사 |
|---|---|---|---|---|---|
| 성 명 | 한글 : 전형석 한문 : 全馨錫 영문 : Chun, Hyung-Suck | | | | |
| 주 소 | | | | | |
| 연락처 | E-MAIL : | | | | |
| 논문제목 | 한글 : USN 기반 구현을 위한 적외선 온도감시 시스템에 관한 연구<br>영어 : A Study on the Infrared thermometer Monitoring System for USN | | | | |

본인이 저작한 위의 저작물에 대하여 다음과 같은 조건아래 조선대학교가 저작물을 이용할 수 있도록 허락하고 동의합니다.

- 다 음 -

1. 저작물의 DB구축 및 인터넷을 포함한 정보통신망에의 공개를 위한 저작물의 복제, 기억장치에의 저장, 전송 등을 허락함
2. 위의 목적을 위하여 필요한 범위 내에서의 편집·형식상의 변경을 허락함. 다만, 저작물의 내용변경은 금지함.
3. 배포·전송된 저작물의 영리적 목적을 위한 복제, 저장, 전송 등은 금지함.
4. 저작물에 대한 이용기간은 5년으로 하고, 기간종료 3개월 이내에 별도의 의사 표시가 없을 경우에는 저작물의 이용기간을 계속 연장함.
5. 해당 저작물의 저작권을 타인에게 양도하거나 또는 출판을 허락을 하였을 경우에는 1개월 이내에 대학에 이를 통보함.
6. 조선대학교는 저작물의 이용허락 이후 해당 저작물로 인하여 발생하는 타인에 의한 권리 침해에 대하여 일체의 법적 책임을 지지 않음
7. 소속대학의 협정기관에 저작물의 제공 및 인터넷 등 정보통신망을 이용한 저작물의 전송·출력을 허락함.

**동의여부 : 동의( 0 ) 반대( )**

2008년 2 월 일

저작자: 전 형 석 (서명 또는 인)

**조선대학교 총장 귀하**