
상황 인식 기반의 유비쿼터스 어플리케이션 구현

서정희* · 박홍복**

Implementation of Ubiquitous Application based on Context-Awareness

Jung-Hee Seo* · Hung-Bog Park**

요 약

정보와 통신 기술의 새로운 경향을 나타내는 유비쿼터스 컴퓨팅은 일반 환경에서 활용할 수 있도록 다량의 정보를 처리할 수 있는 컴퓨팅 기술을 내장하고 있고, 일반 환경의 모든 사물은 통신 기능과의 상호작용을 위해 센서 및 작동기기를 내장하고 있다. 따라서 유비쿼터스 컴퓨팅은 언제, 어디서나 기기에 접속하여 적절한 기능을 수행할 수 있어야 한다. 그러나 유비쿼터스 컴퓨팅과 같은 분산 환경에서 다양한 기능들과의 상호 보완 및 상호 작용을 위한 어플리케이션 설계는 매우 어렵다. 따라서 본 논문은 네트워크 기능이 추가된 장비를 이용한 임베디드 응용 소프트웨어와 LabVIEW를 사용하여 서버 모듈 및 가상 프로토타이핑과의 인터페이스를 구현하고자 한다. 그리고 상황 인식(Context-Awareness)과 위치 인식(Location Awareness)을 통해서 수집한 센서의 정보를 기반으로 상황 인식 기반의 유비쿼터스 어플리케이션을 제안하고, 이를 실험실 환경의 상황 인식을 통한 장치 제어 및 모니터링에 응용한다.

ABSTRACT

Ubiquitous computing is a new paradigm of telecommunication technology and is embedded with advanced computing technology to process a large amount of data in a normal environment. Generally, all equipment is embedded with sensors and operating devices to interaction with communication functions. That is why ubiquitous computing must be able to access any devices anywhere at anytime in order to perform appropriate functions. Unfortunately, however, it is difficult to make an optimized design for applications which can effectively interaction with various functions in distributed environment like ubiquitous computing. Therefore, this paper is aimed at deploying interface with server modules and virtual prototyping by utilizing LabVIEW and embedded application software with additional network function. In addition, given information about sensors collected from context-awareness and location-awareness, it will suggest the ideal ubiquitous application based on context-awareness and apply the advanced application to device control and monitoring through context awareness of lab.

키워드

유비쿼터스 어플리케이션, 유비쿼터스 컴퓨팅, 상황 인식

I. 서 론

정보 통신 기술의 급속한 발전과 네트워크 기능이 추

가된 장치의 개발이 확산되고 있고, 이런 장치를 활용하여 사용자들의 다양한 요구에 적절한 서비스와 환경을 지원할 수 있는 유비쿼터스 어플리케이션은 새로운 시

* 동명대학교 컴퓨터공학과

** 부경대학교 전자컴퓨터정보통신공학부(교신저자)

도로 연구되고 있다.

“유비쿼터스 컴퓨팅”란 용어를 처음 사용한 Weiser는 벽, 테이블과 같은 일상생활의 모든 사물에 컴퓨터가 내장되어야 한다고 생각했다. 따라서 사람들은 수백 개의 컴퓨터와 동시에 상호작용을 할 것이고, 컴퓨터는 시각적으로 인식되지 않는 환경에 내장되어 다른 컴퓨터들과 무선으로 통신한다[1]. 그러나 유비쿼터스 컴퓨팅과 같은 분산 환경에서의 어플리케이션 설계는 다양한 기능들과의 상호 보완 및 상호 작용할 수 있는 어플리케이션 구현의 방대함 때문에 매우 어렵다.

현재, 유비쿼터스 컴퓨팅 어플리케이션 개발을 위한 하드웨어 키트와 소프트웨어 플랫폼은 다양하게 연구되고 있다. 그러나 기존의 유비쿼터스 응용 분야는 매우 제한적이었으나 모니터링, 제어, 의료 분야 및 학습 등 다양한 분야에 접목되어 확대 연구되고 있다. 그뿐만 아니라 논문 [2]는 유비쿼터스 학습 환경에서 모든 학습자의 행동과 학습 습관을 자세히 탐지 및 관찰할 수 있는 개인화된 유비쿼터스 학습 지원 환경을 제시하고 있다. 논문 [3]은 유비쿼터스 컴퓨팅 환경에서 IFNS(Interface-based naming System)이라 불리는 위치-독립적인 Name System의 설계를 제안하고 있다. 언제 어디서나 네트워크에 연결하여 다양한 종류의 “Functional Object”와 거대한 정보량을 유용하게 활용할 수 있는 유비쿼터스 컴퓨팅 환경은 다양한 어플리케이션을 활용할 수 있게 사용자에게 가능한 새로운 플랫폼을 요구하고 있다. 즉 사용자의 어플리케이션을 사용하는 환경이 바뀌더라도 그 환경에 맞는 정보를 제공할 수 있는 방법을 제공해야 한다.

기존의 원격 제어 및 모니터링은 웹 기반으로, 유선 네트워크를 통해 데이터 수집을 위한 장치들이 일정한 장소에 고정되어 데이터 수집 및 네트워크를 구축하기 위한 환경이 매우 제한적이었다[5].

그러나 사용자는 다양한 환경에서의 어플리케이션 개발을 요구하고 있다. 즉, 기존의 원격 제어 및 모니터링을 지원하는 환경은 서버-클라이언트 기반의 컴퓨팅에서 사용자의 다양한 환경과 서비스를 지원하기 위해 모바일과 같은 새로운 기술과의 융합으로 지속적인 기술 향상에 의해 가속화되고 있다. 그리고 다양한 네트워크 구축을 지원하는 유비쿼터스 센서 네트워크(Ubiquitous Sensor Networks: USN)에 의해 환경 정보를 수집하여 무선 네트워크를 통해서 실시간 관리를 용이하게 하고 다

양한 응용에 활용되고 있다.

본 논문은 데이터 수집을 위한 센서와 네트워크 기능이 추가된 ZigbeX와 같은 장비를 기반으로 TinyOS 상에서 동작하는 임베디드 응용 소프트웨어를 개발하여 모트에 포팅하고, LabVIEW를 사용하여 포팅된 모트 즉, 노드 모듈과 서버 모듈 및 가상 프로토타이핑과의 인터페이스를 구현하여 상황 정보를 수집한다. 수집한 센서 정보를 기반으로 상황 인식 기반 유비쿼터스 어플리케이션을 제안하고 이를 실험실 환경의 상황 인식을 통한 장치 제어 및 모니터링에 응용한다.

본 논문의 2장은 유비쿼터스 어플리케이션에 대해 논하고, 3장은 본 논문에서 제안하는 자동 제어 및 모니터링을 위한 상황 기반 유비쿼터스 어플리케이션에 대해 설명하고, 4장은 실험 결과 및 분석, 5장 결론, 참고 문헌 순으로 기술한다.

II. 유비쿼터스 어플리케이션

현재, 다양한 어플리케이션들은 유비쿼터스 컴퓨팅 환경에서 시도되고 있다. 유비쿼터스 컴퓨팅은 정보와 통신 기술의 새로운 경향을 띠고 다량의 정보를 처리하는 컴퓨팅 기술을 일반 환경에서 활용할 수 있도록 내장하고 있다. 이런 컴퓨터들은 데이터를 교환하기 위해서 일반 환경에서의 모든 사물과 통신 기능을 상호작용하는 센서 및 작동기기를 갖추고 있다. 따라서 유비쿼터스 컴퓨팅은 언제, 어디서나 기기에 접속하여 연산을 수행할 수 있어야 한다.

그림 1은 Mori의 유비쿼터스 분류에 따라 유비쿼터스 네트워크, 유비쿼터스 사회 및 유비쿼터스 컴퓨터와 같이 유비쿼터스 분야의 관련성을 설명하고 있다[4]. 유비쿼터스 네트워크는 언제 어디서나 인터넷을 연결할 수 있는 것을 의미한다. 예를 들어, Wi-Fi와 유사하게 무선 LAN를 연결할 수 있어야 한다. 유비쿼터스 사회에서 객체(Object) 환경은 디지털 가전기기, 정보 처리 기능을 가진 운송 시스템, 태그가 부착된 식물등과 같이 통신 기능을 가지고 있음을 의미한다. 유비쿼터스 컴퓨터는 Weiser가 처음 제시한 것과 같이 컴퓨터는 사용자의 눈에 보이지 않는다.

위의 세가지 유비쿼터스 사용은 다음과 같은 세가지 개념을 공유한다. 첫째, 컴퓨터는 어디에서나 존재한다.

둘째, 객체와 환경의 분산된 상호작용은 사용자를 위한 서비스를 생성한다. 셋째, 서비스들은 사용자들의 내용과 상황 정보에 영향을 미친다. 이런 분류를 통해서 Mori는 유비쿼터스 네트워크와 유비쿼터스 사회는 기술-중심적이고, 유비쿼터스 컴퓨터는 인간-중심적이라고 주장하고 있다[4]. 따라서 차세대 유비쿼터스 사회의 주요 논제는 기술 지향적인 설계에서 인간 중심적인 설계로 전환하는 방법이다. 그러므로 유비쿼터스 어플리케이션은 Weiser의 유비쿼터스 근본 개념을 둔 설계를 지향해야 함을 상기시킬 필요가 있다.

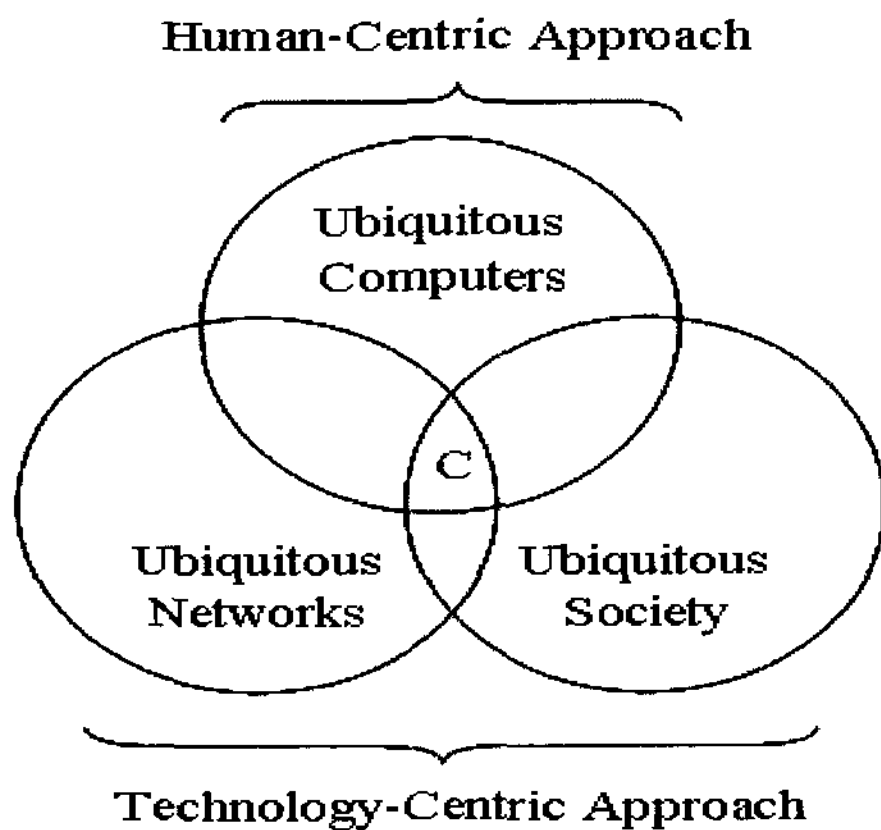


그림 1. 유비쿼터스 분야의 관련성
Fig. 1 Relationship of the ubiquitous fields

본 논문에서 제안하는 유비쿼터스 어플리케이션은 다양한 유비쿼터스의 특징들 중에서 상황 인식(Context-Awareness)과 위치 인식(Location Awareness)을 활용한다. 위치 인식은 무선 주파수(Radio Frequency: RF)를 통해서 현재 위치에서 다른 위치에 있는 노드들과 통신을 담당하고, 상황 인식은 각 지역의 RF 통신에 의해 수집한 상황 정보를 기반으로 각 장치의 제어 및 모니터링을 하는 어플리케이션을 구현한다.

III. 상황 인식 기반의 유비쿼터스 어플리케이션

3.1 장치 제어 및 모니터링

본 논문에서 제안하는 시스템 구성도는 그림 2와 같다. 노드 모듈(Nodes Module)은 각 위치 환경에서 상황 정보를 수집하기 위한 모트, 즉 노드들로 구성되고

TinyOS를 기반으로 하여 노드간의 데이터를 RF 통신을 통해서 베이스 노드로 전송한다. 이런 일련의 과정들은 스케줄링(Scheduling)에 의해서 진행된다. 서버 모듈(Server Module)은 베이스 노드와 RS232C로 연결되어 각각의 위치 환경에서 전송받는 상황 정보를 수집하고 모니터링 및 제어를 위한 어플리케이션과 가상 프로토타이핑(Virtual Prototyping)과의 TCP-IP 통신을 구축한다. 가상 프로토타이핑에서는 실험실 각 장치의 모니터링 뿐만 아니라 상황 인식을 통해서 각 장치의 제어를 수행한다.

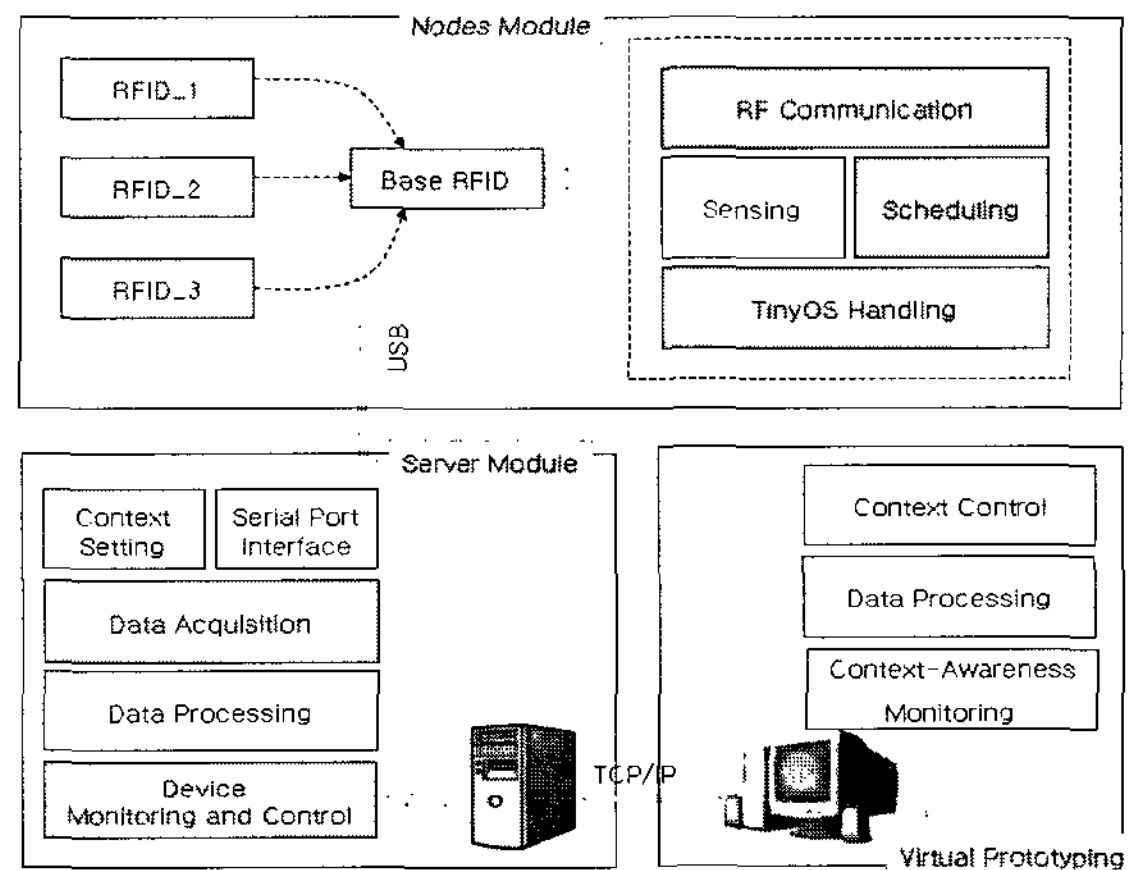


그림 2. 시스템 구성도
Fig. 2 System Architecture

독립적인 위치의 노드들 간에 전송할 상황 정보 메시지 구조는 그림 3과 같다. 두 개의 Sync_Byte는 메시지의 시작과 끝을 나타내는 바이트이고, Packet_Type은 패킷 형식, Payload Data는 사용자 정의 데이터를 나타낸다. Payload Data 구조의 세부 구성은 다음과 같다.

- (1) Address는 UART와 Broadcast 중의 하나를 의미한다.
- (2) Message Type은 Message 형태를 나타낸다.
- (3) Group ID는 베이스 노드와 각각의 노드들을 하나의 그룹으로 인식할 수 있는 ID로써 모든 노드에 동일한 ID를 내장한다.
- (4) Data Length는 Mote ID ~ Photo Diode까지의 데이터 길이를 나타낸다.
- (5) Mote ID는 각각의 노드를 식별하기 위한 노드 ID

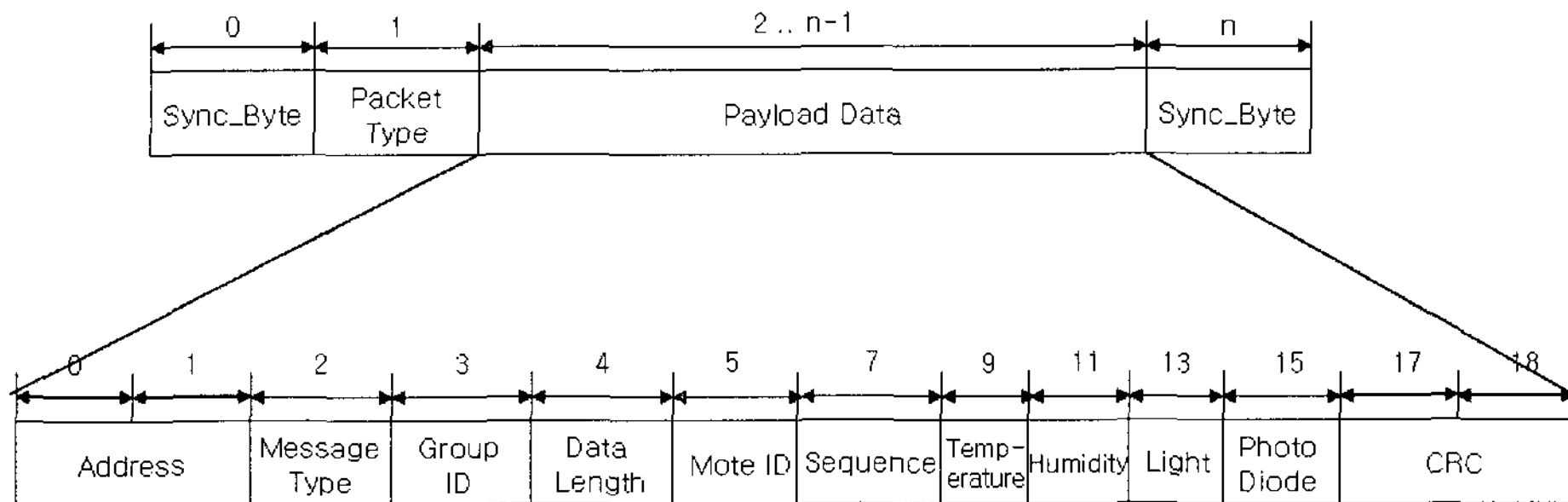


그림 3. 전송 메시지 구조
Fig. 3 Transmission Message Structure

이다. 노드에 프로그램을 포팅할 때 각 노드에 개별적인 ID를 설정한다.

- (6) Sequence는 각 노드별로 베이스 노드에 전송된 일련의 순서를 나타낸다.
- (7) Temperature는 노드에서 측정한 온도 데이터로써 2바이트로 구성된다.
- (8) Humidity는 노드에서 측정한 습도 데이터로써 2바이트로 구성된다.
- (9) Light는 노드에서 측정한 조도 데이터로써 2바이트로 구성된다.
- (10) Photo Diode는 노드에서 측정한 적외선 데이터로써 2바이트로 구성된다.
- (11) CRC는 전송 데이터의 오류 체크로 2바이트로 구성되어 있다.

본 논문에서는 환경 정보의 상황 인식을 통한 실시간 분석 및 장치 제어를 위해 모트들 간의 RF 통신을 통해서 실험실 환경에서 독립적으로 위치한 모트의 상황 데이터를 측정한다. 독립적으로 위치한 각 모트들은 센서로부터 현 위치의 상황 정보를 수집하고, RF 통신으로 ZigbeX 모트 즉, 베이스 노드에 전달한다. 베이스 노드는 서버와 시리얼 채널(Serial Channel)로 연결되어 여러 노드에서 수집된 상황 데이터를 서버측으로 전송한다. 따라서 베이스 노드는 시리얼 채널과 무선 채널(Wireless Channel) 사이의 연결을 담당한다. 서버 모듈은 베이스 노드로부터 수집한 상황 정보를 실시간으로 분석하고 클라이언트와의 TCP-IP 연결을 수행한다. 클라이언트는 가상 프로토타이핑을 설계하고 각종 장치의 상태를 모니터링하여 서버 모듈에서 분석한 상황 정보를 통해서 각 장치에 대한 제어를 수행한다.

노드들간의 무선 통신은 표 1의 상황 데이터 전송과 같이 ReceiveMsg.receive() 함수를 통해서 RF 통신으로 데이터를 받게 되고, SendMsg.send()와 SendMsg.sendDone() 함수를 통해서 상황 데이터를 무선으로 전송한다.

3.2 제안된 알고리즘 수행 절차

본 논문에서 제안하는 알고리즘 수행 절차는 표 2와 같다.

표 1. 상황 데이터 전송
Table. 1 Context Data Transmission

```

task void transmitting()
{
    AllsensorMSG *pack = (AllsensorMSG *)msg.data;
    ...
    if (TOS_LOCAL_ADDRESS != 0) {
        if (call SendMsg.send(TOS_BCAST_ADDR,
            sizeof(AllsensorMSG), &msg))
            ...
        else {
            if (call SendMsg.send(TOS_UART_ADDR,
                sizeof(AllsensorMSG), &msg))
                ...
            }
        }
    }

event TOS_MsgPtr ReceiveMsg.receive
(TOS_MsgPtr m)
{
    if (TOS_LOCAL_ADDRESS == 0)
    {
        memcpy (&Recvmsg, m, sizeof(TOS_Msg));
        call SendMsg.send((TOS_UART_ADDR,
            sizeof(AllsensorMSG), &Recvmsg);
        ...
    }
    return m;
}
    
```

서버 모듈은 TCP 네트워크 연결을 위한 Listener를 생성하고 Listener인 가상 프로토타입에서 TCP 네트워크 연결이 수락될 때까지 기다린다. 3번 단계에서 에러가 발생하지 않으면서 Listener가 수락되면 VISA resource name에 의해서 Serial Port 설정 및 초기화한다. 단계 6-7번은 서버 모듈에서 write와 read control를 선택하면 버퍼로부터 VISA resource name 또는 디바이스로 데이터를 쓰기 또는 지정된 바이트를 읽어 들인다. 8-9번 단계에서 서버 모듈은 읽어 들인 Packet Data를 해당 서버에 디스플레이 하고 TCP 네트워크 연결 요청이 수락된 Listener인 가상 프로토타입으로 Packet Data를 Write한다.

가상 프로토타입에서는 주소, 포트와 TCP 네트워크 연결을 Open하여 서버 모듈에서 전송한 Packet Data를 읽어 들이고, 현재 가상의 각 장치의 상태의 데이터를 업데이트한다.

표 2. 제안된 알고리즘
Table. 2 Proposed Algorithm

1. Create a Listener for TCP network connection
 - Open TCP Port Number
2. Waits for an accepted Listener
3. if Listener accepted
 - then TCP network connection로부터 바이트 읽기
4. Serial Port Configuration
 - serial channel로 연결된 resource를 활성화
 - write/read 가능하게 설정
 - baud rate, data bits, parity, stop bits, flow control 설정
5. Initializes the Serial Port by the resource specified settings
6. if 'write control' is selected
 - then Writes the data from write buffer to the device or interface
7. if 'read control' is selected then
 - serial port의 byte numbers를 setting
 - Reads the specified number of bytes from the device or interface
8. Server Module에서 Packet Data 읽기와 모니터링
9. Server Module의 Packet Data를 Listener로 전송하고 Listener는 모니터링 및 제어 수행

그리고 가상 프로토타입에서 설정한 환경 데이터를 바탕으로 각 장치에 대해 제어를 수행하여 상황 인식 데이터와 연관된 장치의 상태를 변경한다.

IV. 실험 결과 및 분석

본 논문의 실험 환경으로 모트는 ZigbeX로 TinyOS가 포팅되어 실시간 센서 네트워크의 구성이 가능하다. 그리고 저전력 CPU인 Atmega128과 저전력 단거리 무선 통신 표준인 802.15.4 표준화 통신을 지원할 수 있는 CC2420 칩이 내장되어 있고, 2.4GHz의 Zigbee 표준을 지원한다.

본 논문에서 제안하는 유비쿼터스 어플리케이션은 TinyOS 상에서 임베디드 소프트웨어를 개발하여 센서가 부착된 모트에 포팅하고 LabVIEW를 이용하여 USN 응용 프로그램 개발에 적용한다.

그림 4에서 각각의 독립적인 위치의 모트로부터 환경 정보를 수집한 베이스 노드는 서버 모듈로 상황 정보를 전송하여 'read string'에 Hex 코드로 디스플레이하고, '패킷데이터'는 이 데이터를 그래프로 나타내고 있다. 'read string'의 메시지 구조는 그림 3과 같다.

그림 4와 같이 수집한 디지털 온도·습도 데이터를 온도(C)와 습도(%RH) 단위로의 계산은 식(1)~(3)과 같다. 식 (1)은 디지털 온도를 온도(C) 단위로 계산하고, 식 (2)는 디지털 습도를 습도(%RH) 단위로 계산한 후 식 (3)과 같이 온도값으로 습도값을 보정하고 Compensated_Humidity(%RH)가 최대값(100)과 최소값(0.1) 사이의 값을 유지하도록 한다.

$$Temperature (C) = digital - Temp \times 0.01 - 40 \quad (1)$$

digital-Temp = 디지털 온도

$$Humidity(\%RH) = C_3 \times digital - Humi \times digital - Humi + C_2 \times digital - Humi + C_1 \quad (2)$$

digital-Humi = 디지털 습도, $C_1 = -4.0$, $C_2 = 0.0405$
 $C_3 = =0.0000028$

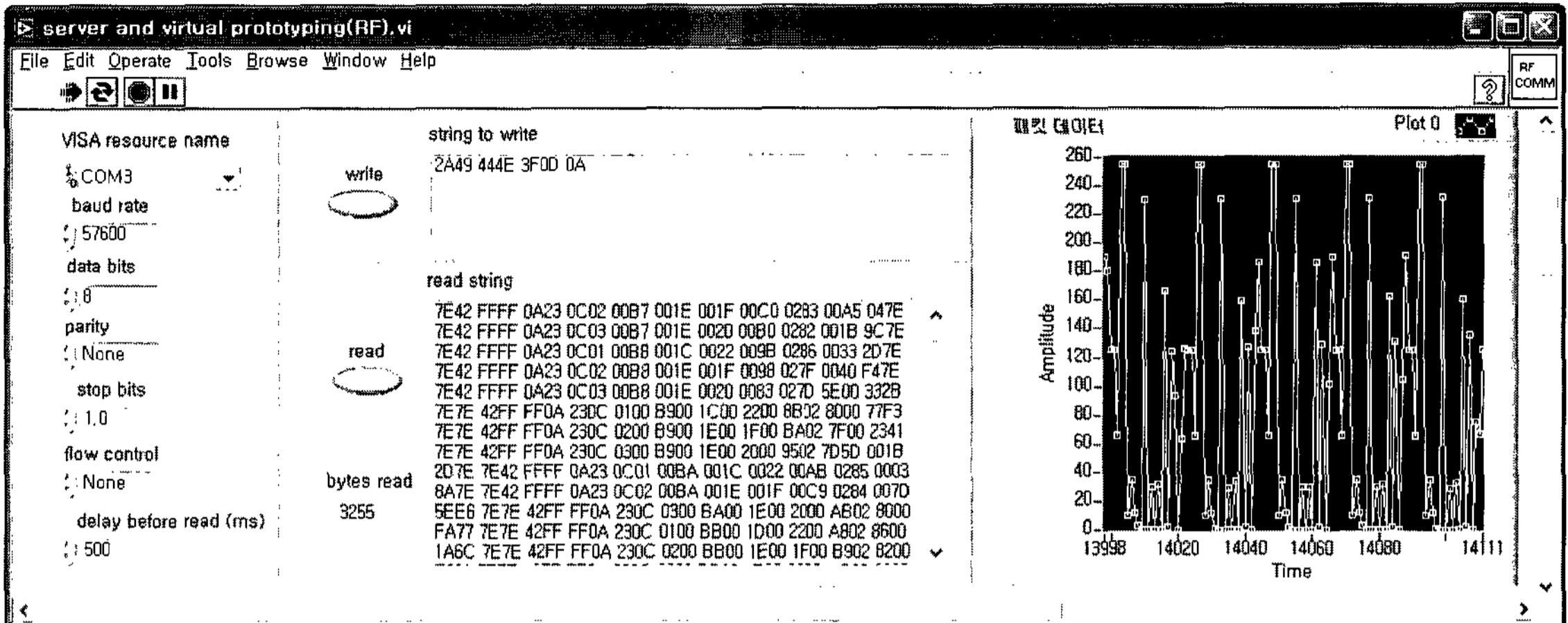


그림 4. 서버 모듈에서의 수신 패킷 데이터
Fig. 4 Receiver Packet Data in Server Module

$$Compensated - Humidity(\%RH) = (digital - Temp - 25) \times (T_1 + T_2 \times digital - Humi) + Humidity(\%RH) \quad (3)$$

$$T_1 = 0.01, T_2 = 0.00008$$

로토타이핑에서는 현재 위치한 모트의 상황 정보를 모니터링하고 클라이언트에서 설정한 환경 정보와 수집한 데이터를 분석하여 각 장치의 제어를 수행하여 상황 인식 데이터와 관련된 장치의 상태를 변경한다.

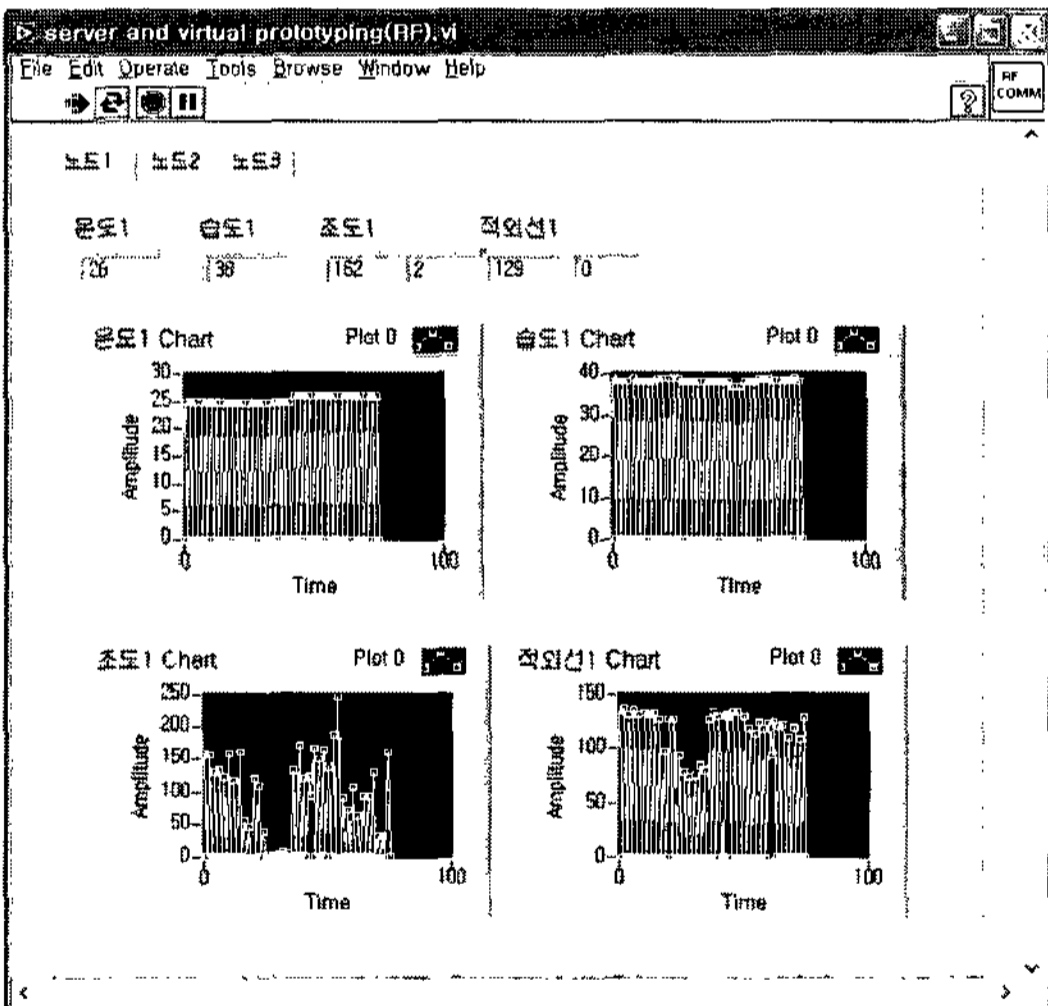


그림 5. 각 노드별 수집 데이터 모니터링
Fig. 5 Monitoring Acquisition Data of each Node

그림 5는 그림 4에서 수집한 데이터를 각각의 독립적으로 위치한 모트별로 온도, 습도, 조도, 적외선의 상황 데이터를 그래프와 텍스트로 모니터링하고 있다. 그림 6은 상황 정보를 수집하기 위해 위치한 모트의 각 장치를 제어하기 위한 가상 프로토타이핑을 나타낸다. 가상 프

표 3. 가상 프로토타이핑의 상황 인식 정보와 수행 결과

Table. 3 Action Result of Content Awareness Information and Virtual Prototyping

Content	Function Tool	Recognition	Action	
위치	RFID	node identifier number	awareness	
온도	Sensor	environment data	sensing	
습도	Sensor	environment data	sensing	
조도	Sensor	environment data	sensing	
적외선	Sensor	environment data	sensing	
에어컨	Virtual Machine	on/off or environment data	수동/자동 제어	
난방기	Virtual Machine	on/off or environment data	수동/자동 제어	
전등	Virtual Machine	on/off or environment data	수동/자동 제어	
온도계	Virtual Machine	environment data	모니터링	
습도계	Virtual Machine	environment data	모니터링	
스위치	전등	Virtual Machine	on/off	수동제어
	난방	Virtual Machine	on/off	수동제어
	에어컨	Virtual Machine	on/off	수동제어

표 3은 상황 정보와 가상 프로토타이핑을 연결한 수행 결과를 나타내고 있다. 실험실의 환경 데이터의 위치 정보는 RFID로 각 지역의 ID를 인식하고, 온도, 습도, 조도, 및 적외선은 모트에 내장된 센서에 의해서 해당 ID의 환경 데이터를 연속적으로 수집한다. 수집한 환경 데이터를 데이터 처리 과정을 통해서 환경 설정 데이터와 분석하여 가상 프로토타이핑의 가상 머신인 에어컨, 난방기, 전등에 적용하여 장치를 제어하고, 온도계 및 습도계에서 모니터링된다.

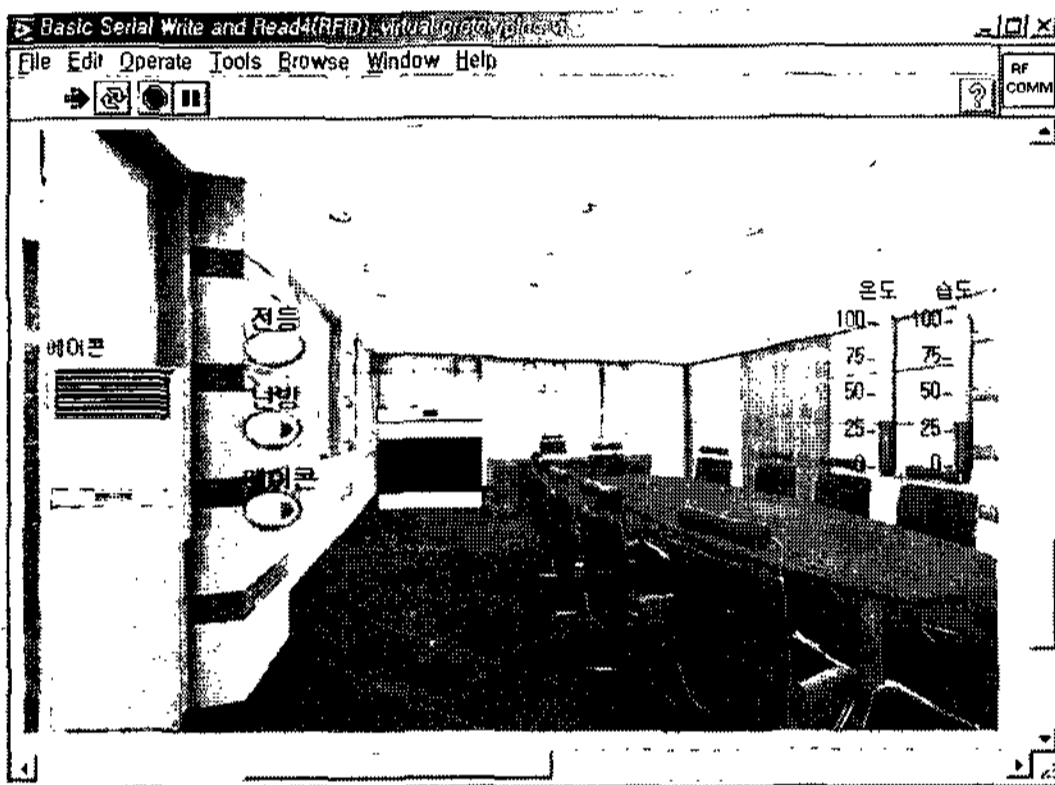


그림 6. 가상 프로토타이핑에서의 장치 제어 및 모니터링

Fig. 6 Device Control and Monitoring in the Virtual Prototyping

V. 결 론

본 논문의 상황 인식에 의한 유비쿼터스 접근은 실험실 환경 데이터와 유비쿼터스 어플리케이션 사이의 상호 작용으로 구성된다. 따라서 TinyOS 기반의 유비쿼터스 센서 네트워크 장비를 이용하여 임베디드 응용 소프트웨어를 개발하고, 서버 모듈과 가상 프로토타이핑을 통해서 수집한 센서의 정보를 기반으로 상황 기반 유비쿼터스 어플리케이션을 제안한다. 그리고 제안한 어플리케이션은 실험실 환경의 상황 인식을 통한 장치 제어 및 모니터링에 응용하였다.

앞의 실험 결과 및 분석에서 기술했듯이 실험실의 실제 상황 정보를 서버 모듈에서 모니터링하는 어플리케이션을 구현하고, 서버 모듈에서 전송한 상황 정보를 클라이언트에서 각 장치의 상태를 제어할 수 있는 가상 프

로토타이핑을 설계하여 시뮬레이션 하였다. 향후 연구 과제로써는 클라이언트에서 수행되고 있는 가상 프로토타이핑을 실물 프로토타이핑으로 구현하고 모바일 폰 또는 PDA와 같은 장치에서의 제어 및 모니터링 할 수 있는 어플리케이션을 실험할 계획이다.

참고문헌

- [1] Zahid Anwar, Jalal Al-Muhtadi, Willian Yurcik, Roy H. Campbell, "Plethora: A Framework for Converting Generic Applications to Run in a Ubiquitous Environment," Proceedings of the Second Annual International Conference on Mobile and Ubiquitous System: Network and Services, IEEE CS, 2005.
- [2] Z. Cheng, Q. Han, S. Sun, M. Kansen, T. Hosokawa, T. Huang, A. He, "A Proposal on a Learner's Content-aware Personalized Education Support method based on Principles of Behavior Science," Proceedings of the 20th International Conference on Advanced Information Networking and Applications(AINA'06), IEEE, 2006.
- [3] M. Minami, H. Morikawa, and T. Aoyama, "The Design of naming-based Service Composition System for Ubiquitous Computing Applications," Proc. of the 2004 International Symposium on Applications and the Internet Workshops(SAINTW'04), pp. 304-312, January 2004.
- [4] Kentaro Go, Naotake Hirasawa, Hideaki Kasai, and Masaaki Kurosu, "Supporting Omoiyari: A Design issue in Post-Ubiquitous Society," SICE-ICASE International Joint Conference 2006, Oct. 18-21, pp. 5444-5449, 2006.
- [5] 서정희, 박홍복, "LabView를 이용한 실시간 온실 모니터링 및 원격 제어," 한국해양정보통신학회논문지, 제7권, 4호, pp.779-787, 2003. 8월.

저자소개



서 정 희(Jung-Hee Seo)

1994년 신라대학교 자연과학대학
전자계산학과(이학사)

1997년 경성대학교 대학원 전산통계
학과(이학석사)

2006년 부경대학교 대학원 전자상거래 시스템 전공
(공학박사)

현재 동명대학교 컴퓨터공학과 전임강사

※관심분야: 멀티미디어 응용, 정보 보호, 모바일, 원격
교육



박 흥 복(Hung-Bog Park)

1982년 경북대학교 공과대학 컴퓨터
공학과(공학사)

1984년 경북대학교 대학원 컴퓨터공
학과(공학석사)

1995년 인하대학교 대학원 전자계산학전공(이학박사)

1984년~1995년 동명대학 전자계산과 부교수

2001. 2~2002. 2 The University of Arizona 객원교수

1996년~현재 부경대학교 전자컴퓨터정보통신공학부
교수

※관심분야: 모바일 시스템, 멀티미디어 응용, 컴파일
러, 원격 교육