

센서 네트워크 모니터링 시스템의 구현

서정민*, 호효휘**, 최정훈**, 이상문**

*지니정보시스템

**충주대학교 전자계산학과

e-mail:smlee@cjnu.ac.kr

An Implementation of Sensor Network Monitoring System

Jeong Min Seo*, Hu Xiao Wei**, Jung Hun Choi**, Sang Moon Lee**

*Jeenie Info. Sys. Co.

**Dept of Computer Science Chungju Nat'l University

요 약

유비쿼터스 센서 네트워크(USN)는 다수의 u-센서 노드간 애드 혹(Ad Hoc), 멀티 홉(Multi Hop) 통신으로 구성된다. 그러나 USN이 시작 단계인 현재 시점에서, 과거 시리얼이나 이더넷으로 구성된 유선 네트워크를 지그비, 블루투스, 와이파이 등을 이용한 포인트 투 포인트(Point To Point) 무선 네트워크로 전환하는 이슈가 많은 부분을 차지하고 있다. 하지만 적용 범위가 넓고 별도의 광역 인프라를 구성하기 곤란한 경우, USN을 이용한 애드 혹, 멀티 홉 메시 네트워크를 구성하는 사례도 점차 늘어나고 있다. 본 논문에서는 이러한 환경에서 유비쿼터스 센서 네트워크를 보다 쉽고 편리하게 모니터링 하기 위한 시스템을 제안하였다.

키워드 : 유비쿼터스, 센서, 모니터링시스템, USN

I. 서론

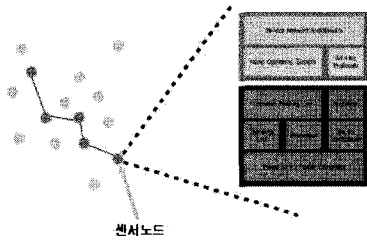
유비쿼터스 센서 네트워크(USN)는 다수의 u-센서 노드간 애드 혹(Ad Hoc), 멀티 홉(Multi Hop) 통신으로 구성된다. 그러나 USN이 시작 단계인 현재 시점에서, 과거 시리얼이나 이더넷으로 구성된 유선 네트워크를 지그비, 블루투스, 와이파이 등을 이용한 포인트 투 포인트(Point To Point) 무선 네트워크로 전환하는 이슈가 많은 부분을 차지하고 있다. 하지만 적용 범위가 넓고 별도의 광역 인프라를 구성하기 곤란한 경우, USN을 이용한 애드 혹, 멀티 홉 메시 네트워크를 구성하는 사례도 점차 늘어나고 있다. 애드 혹 메시 네트워크는 고정된 네트워크 구조를 가지지 않고 이동 단말(Mobile Host) 사이에 임의로 구성되는 형태이다. 새로운 센서 노드가 추가되거나 제거되는 경우 자동으로 노드간 최적 네트워크가 재구성되고, 각 노드가 직접 외부 네트워크와 연결되어 있지 않더라도 징검다리 형태의 홉 대 홉(Hop by Hob) 통신을 함으로써 최종 목적지인 게이트웨이 또는 베이스 스테이션으로 정보를 전달할 수 있다[1]. 본 논문에서는 이러한 복잡한

환경에서 유비쿼터스 센서 네트워크를 보다 쉽고 편리하게 모니터링 하기 위한 시스템을 제안하였다.

II. 관련 연구

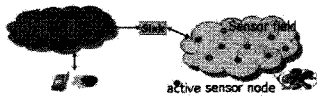
1. 센서 네트워크

센서 네트워크는 센서 노드와 계산, Sensing, 무선 통신을 하는 네트워크로 특정 지역에 센서 노드를 설치하여 주변 정보 또는 특정 목적의 정보를 획득하고, 베이스 스테이션이 수집하여 이를 활용하기 위한 서비스 환경이다. 노드당 가격이 저렴하여 대규모 네트워크의 구축이 용이하며, 특정 노드에 결함이 생겨도 다른 경로를 이용하여 네트워크 유지한다. 그러나 노드의 위치는 필요에 따라 유동적이 될 수 있고 네트워크의 수명은 각 노드의 배터리에 대한 의존이 크므로 배터리 사용이 최소화되도록 거의 대부분 Sleep 상태였다가, 인터럽트 발생 시 활성화 또는 실행 상태로 전환한다.



〈그림 1〉 USN의 구조

센서노드는 자료를 처리하는 Computation, 자연의 음성, 빛, 온도, 습도 등을 감지하는 Sensing, RF transceiver, Laser Module 등이 있는 Communication 부분으로 구성된다. 또한 싱크노드는 인터넷이나 인공위성을 통하여 사용자와 통신 서비스를 제공한다[2].



〈그림 2〉 싱크노드의 구성 예

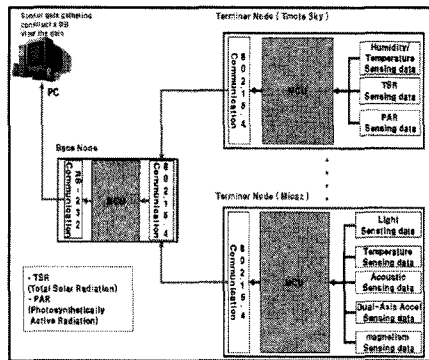
2. NesC

NesC 프로그래밍 언어는 TinyOS의 구조적인 개념들과 실행 모델들을 구현하기 위해 사용되어 졌다. 이는 어플리케이션 개발자들에게 센서 네트워크와 같은 임베디드 네트워크 시스템들의 구현을 지원하기 위한 새로운 프로그래밍 영역을 제공해 주고자 개발 되어진 프로그래밍 언어이다. NesC 프로그래밍 언어가 센서 네트워크와 같은 여러 중요한 센서 어플리케이션 분야에 널리 사용되어 질 수 있는 이유는, 새로운 분야인 임베디드 네트워크 시스템들에 의해서 요구 되어지는 복잡성과 동시성을 충분히 지원해 줄 수 있는 프로그래밍 언어이기 때문이다. NesC 는 네트워크 임베디드 시스템에서 프로그래밍언어의 효율성을 높이기 위해, 컴포넌트 기반으로 설계되어 있으며, 이로 인해 코드의 재사용성, 컴파일 시점에서의 에러 검출 등이 장점이다. 문법은 C 언어와 동일하며, 각 함수들을 묶어주는 컴포넌트, 인터페이스의 개념을 도입해서 앞에서 말한 장점들을 제공한다. NesC는 컴포넌트라는 개념에 기반 하여 TinyOS의 이벤트 기반의 동시성을 지원하고, 공유된 데이터의 동시 액세스가 가능하게 되었다. 결론적으로 네트워크 임베디드 시스템에서 효율적 개발을 가능하게 지원하고 있다. 특히 TinyOS에 공개되어 있는 많은 컴포넌트들을 쉽게 사용할 수 있다. 결국 NesC는 컴포넌트라는 개념에 기반 하여 TinyOS의 이벤트 기반의 동시성을 지원하고, 공유된 데이터의 동시 액세스가 가능하게 되었다. NesC에서의 특징들을 살펴보면, NesC는 컴포넌트로 구성되는데 이러한 컴포넌트는 '모듈'과 'Configuration'이라는 두 개의 타입으로 존재한다. 컴포넌트들은 인터페이스들을 제공하거나 사용한다. 즉, 이러한 인터페이스들이 컴포넌트에 액세스 하는 유일한 통로를 제공하는 것이다. 특히, 인터페이스는 양

방향이라는 특성을 갖는데, 이는 커맨드와 이벤트를 포함함으로써 가능하다. 예를 들어, 제공자는 인터페이스가 커맨드를 구현하는 반면에 사용자는 이벤트를 구현하는 것이다. NesC의 또 다른 큰 특징은 동시성이다. 이러한 동시성을 확보하기 위해서 태스크와 Atomic 구문의 두 가지의 방법을 사용한다[3].

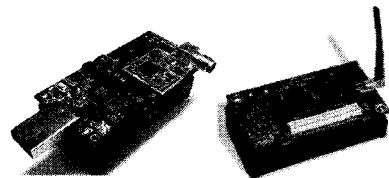
III. 시스템 설계

그림 3은 제안시스템의 H/W 구조를 나타내고 있다. 구성 부분은 센싱 자료를 저장하고 처리하기 위한 PC부분과 Base Node, Tmote Sky 4개, Micaz 2개로 구성하였다. Base Node와 PC는 RS-232를 이용하여 통신하며, 나머지 센서 부분은 802.15.4를 이용하여 통신한다.



〈그림 3〉 H/W 시스템 구조

Tmote Sky 센서는 습기와 온도 자료, TSR (Total Solar Radiation), PAR (Photosynthetically Active Radiation) 자료를 감지하며[4], Micaz 센서는 빛, 온도, 소리, 자기, 압력, 가속도 등의 자료를 감지한다[5].



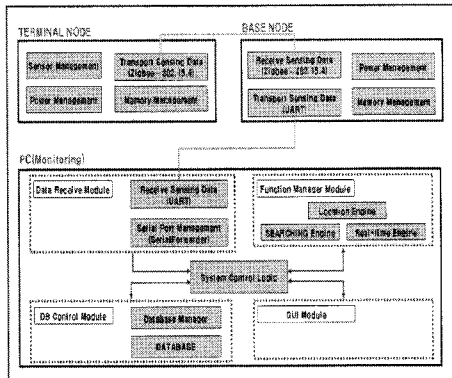
〈그림 4〉 사용한 Tmote와 Micaz 센서

그림 4는 본 논문에서 실험용으로 사용한 Tmote와 Micaz 센서이다.

그림 5는 본 논문에서 구현한 시스템의 S/W 모듈의 구조를 보여주고 있다. PC상에서 운용되는 모니터링시스템은 터미널 노드와 베이스 노드로부터 입력되는 센싱 자료를 받아 처리하는 기능을 기본으로 하여 그래프모드와 텍스트모드로 출력하며, 노드의 상대적인 위치를 보여준다.

그래프모드는 그래프를 통하여 센싱값을 보여주며, 사용

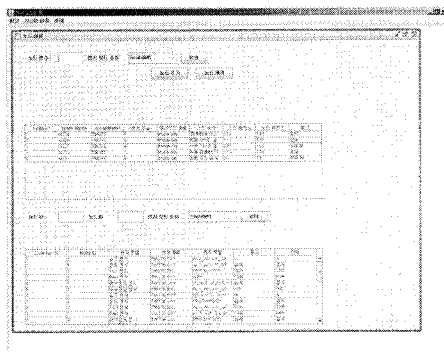
자가 이해하기 쉽게 캘리브레이션을 처리한 값과 아무런 처리를 하지 않은 원시자료를 보여준다. 텍스트 모드는 사용자의 특색에 맞게 센싱값들을 텍스트 값으로 보고자 하는 사람을 위한 모드를 제공한다. 마지막으로 노드의 상대적인 위치는 센서 네트워크를 구성하는 배경을 한 장 넣고 그 위에서 노드들의 상대적인 위치를 보여 준다(6. 7. 8. 9).



〈그림 5〉 S/W 모듈 구조

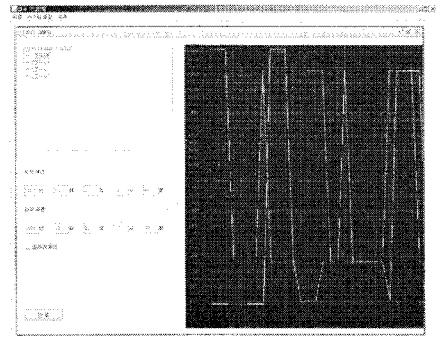
IV. 시스템 구현

그림 6은 노드 설정화면으로 노드 등록은 노드 개수를 입력 하고 노드에 대한 센서보드의 종류를 선택한 후 확인 버튼을 클릭하면 노드가 추가된다. 노드 제거는 노드의 행을 선택 후 노드제거 버튼을 누르면 노드가 제거 된다. 또한 센서 등록은 각 노드의 처음과 끝을 입력하고 대한 센서보드의 종류를 선택한 후 확인 버튼을 클릭하면 센서가 추가된다. 마지막으로 센서 제거기능은 센서의 행을 선택 후 노드제거Key를 클릭하면 노드가 제거 된다.



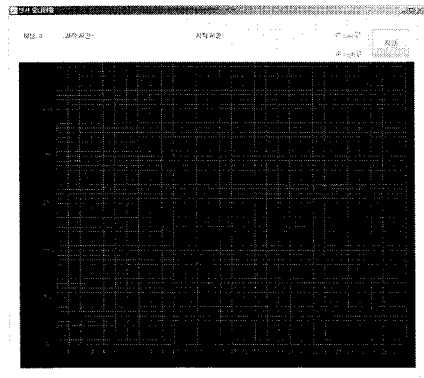
〈그림 6〉 노드 등록 화면

그림 7은 센싱 그래픽 화면으로 원하는 노드를 클릭 후 시작 시간과 종료시간을 입력 후 확인 버튼을 클릭하면 선택된 노드의 각 채널에 대한 데이터를 그래프로 보여준다.



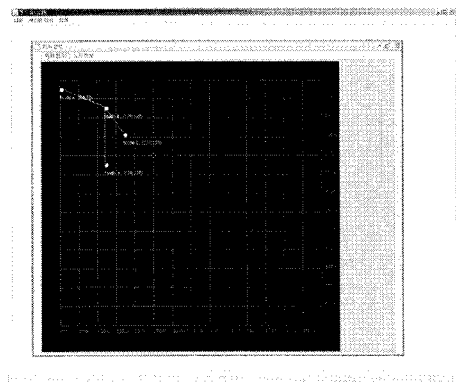
〈그림 7〉 센싱 그래픽 화면

그림 8은 센싱 그래픽을 세부적으로 보여주는 것으로 각 센서의 값을 보고 싶을 때는 센서가 속한 노드를 선택 후, 센서를 선택하고 시작 시간과 종료시간을 클릭한 후 완료 버튼을 누르면 선택된 센서의 캘리브레이싱된 값을 보여준다.



〈그림 8〉 세부적 센싱 그래픽 화면

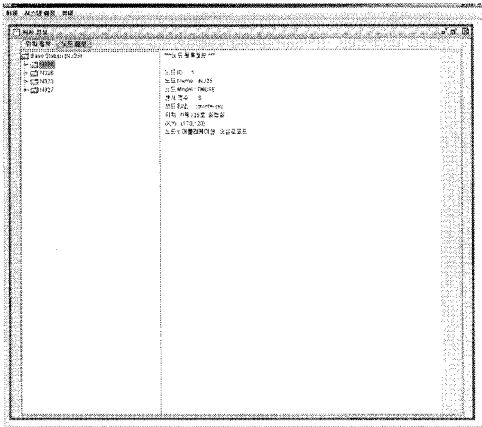
그림 9는 센서의 위치 정보를 보여주는 것으로써 각 노드에 대한 위치를 그래픽적으로 보여준다.



〈그림 9〉 센서의 위치 정보 화면

마지막으로 그림 10은 노드 정보를 보여주는 화면으로 각 노드를 선택하면 각 노드의 등록정보를 보여주고 각 노드에

속한 센서를 클릭하면 선택된 센서의 등록정보를 보여준다.



(그림 10) 센서 등록 정보 검색 화면

Dame, USA October 2005 Proceedings”, Springer-Verlag New York Inc, 2006. 6

[4] http://maxfor.co.kr/sub5_1_1.html

[5] <http://www.tinyosmall.co.kr/html/mainm.html>

[6] 김영일 외, “RFID/USN 기술을 이용한 전력설비관리 서비스 구현”, 한국정보처리학회 논문지 D, VOL.15 NO. 02 pp.263~270, 2008. 4

[7] 장동욱 외, “응급상황의 신속한 감지를 위한 u-Health 시스템 개발에 관한 연구”, 한국정보처리학회 논문지 B, VOL.14-B NO.6 pp.401~406, 2007. 10

[8] 정의민 외, “USN 을 이용한 스키장 사각지역 관리”, 한국정보처리학회 2008년 춘계학술대회, VOL.15 NO. 01 pp.520~522, 2008. 5

[9] 하은용 외, “USN을 적용한 강의실 조명 전원 관리 시스템”, 한국정보과학회 2007 추계학술대회, VOL.34 NO. 02(D) pp.500~504, 2007. 10

V. 결론

지금까지 센서 네트워크 모니터링 시스템의 구현에 관해 살펴 보았다. 본 논문에서 제안한 센서 네트워크 모니터링 시스템은 각 노드에 속한 센서의 값을 ADC 값으로 보여줄 뿐만 아니라 캘리브레이션된 값으로 보여주고, 데이터베이스에 저장되기 때문에 위험한 환경이나, 사람이 작업할 수 없는 곳에 센서를 부착하여 각 센서의 값을 보여주며 또한 데이터베이스에 저장된 값을 활용하여 통계적인 수치를 계산 할 수 있다.

현재 시스템에서는 각 센서들의 모니터링을 주된 작업을 하였다. 그래서 각 센서들의 변화 값을 이용하여 제어를 할 수는 없는 것이 현 시스템의 단점이다. 따라서 향후 개선방안으로 현재의 시스템에다가 센싱 값을 활용하여 어떠한 작업을 할 수 있는 제어 모듈을 추가하여 좀 더 효율적인 시스템으로 발전시켜야 할 것이다.

참고문헌

[1] 김관중 외, “USN 서비스 및 시장 동향”, 한국정보과학회 학회지, VOL.25 NO.12 pp.7~18, 2007. 12

[2] R.Riggio, et.al, “Topics in ad hoc and sensor networks - Hardware and software solutions for wireless mesh network testbeds”, IEEE Communications Magazine, VOL.46 NO.6 pp.156~162, 2008. 6

[3] Antsaklis et.al, “Networked Embedded Sensing And Control : Workshop Nesc’05: University of Notre