



# 스마트 센서 노드 운영체제 기술

한국정보통신대학교 김 대 영  
한국정보통신대학교 홍 승 기



## ▷▷▷ 임베디드 S/W 특집

- 임베디드 S/W 산업 육성 추진방안
- 임베디드 S/W 산업 동향 및 향후 전망
- IP 셋톱박스 S/W 플랫폼 기술
- **스마트 센서 노드 운영체제 기술**
- 임베디드 S/W 개발도구 기술
- SCM기반의 임베디드 S/W 인력양성 정책

## 요약

u-Korea 실현을 위한 초석으로 사회 여러 분야에서 유비쿼터스 센서 네트워크(USN)에 대한 관심이 높아지고 있다. 향후 USN 기술은 가정, 물류/유통, 교통, 행정, 보건, 복지, 그리고 환경 등의 다양한 분야에 적용될 것이고, 다가올 미래 사회의 기반 인프라로 자리 잡게 될 것이다. USN이 구축된 사회에서는 모든 사물의 지능화로 자율적으로 주변 환경을 센싱하여 주변 상황을 인식하고 이들을 제어 하는 정보 환경이 형성된다. 본 고에서는 USN 구축을 위한 임베디드 S/W 기술중 가장 기본이 되는 센서 노드의 운영체제 기술 개발 동향과 운영체제가 동작하게 되는 플랫폼인 센서 노드 하드웨어에 대해 소개하고자 한다.

## I. 서론

우리나라의 IT산업 경쟁력 강화 및 세계 시장 선도를 위해 정보통신부에서 추진 중인 IT839 전략의 3대 첨단 인프라 중 하나인 RFID/USN은 미래의 유비쿼터스 사회 실현을

위한 핵심 기반 구조이다. 수많은 센서 노드들은 자율적으로 정보를 수집, 관리 및 제어하는 협력 네트워크를 자동 구성하여 다양한 산업 응용 분야에서 물리 현상계와 컴퓨터, 나아가서는 인간을 연결하는 인터페이스 역할을 담당하게 된다. 그리고 언제라도 원하는 정보를 실시간으로 이용자 요구에 맞춘 형태로 제공한다. 예를 들면, 가정에서는 실내 온도, 습도, 화재 감지, 소동, 창문 개폐의 자율적 제어 등 가정 내의 모든 기기에 대한 원격제어, 원격 점검이 가능하게 되고, 물류/유통에서는 택배, 항공/항만 수하물 등의 출발지로부터 도착지까지 이르는 모든 경로가 실시간으로 관리되어 도난, 훼손 방지, 지연 도착 등을 예방 할 수 있게 된다. 다양한 응용 분야에서 USN의 기본 요소인 센서 노드는 대부분 사물에 내장되기 때문에, 일반적으로 배터리로 동작하며 그 크기가 작고 전력 소모가 적어야 한다. 그리고 외부의 환경 변화에도 가능한 영향을 받지 않아야 한다. 센서 노드에 탑재되는 임베디드 센서 노드 운영체제는 제한된 메모리와 CPU 자원을 최대한 활용할 수 있어야 하며 노드에 할당된 작업들을 처리하고 네트워크 내에서의 통신이 원활하게 수행 될 수 있도록 보장해야 한다. 본 고에서는 센서 네트워크 관련 임베디드 소프트웨어 중 가장 핵심인 센서 노드 운영체제와 운영체제가 탑재되는 센서 노드 하드웨어에 대한 기술 동향을 살펴보기로 한다.

## II. 센서 노드 하드웨어

센서 네트워크 응용을 위한 하드웨어는 아직까지 표준화가 된다던지 혹은 상업적으로 대량 생산되어 사용되는 제품은 찾아보기 힘들다. 하지만 미국방성의 DARPA 프로젝트의 스폰서를 받아 개발된 버클리대의 Mote시리즈는 미국 정부와 관련 대학과 기업의 노력으로 미국에서 가장 널리 사용되는 하드웨어 플랫폼이 되었다. 이러한 배경에는 Mote 플랫폼이 하드웨어 뿐만

아니라 TinyOS라는 초소형 운영체제와 TinyDB, NesC, 시뮬레이터 등의 개발환경과 함께 공개 하드웨어/소프트웨어로 제공되기 때문이다. 또한 MEMS 기술을 이용하여 Mote 하드웨어를 초소형화한 스마트 더스트는 형광등과 같은 저가의 소형 시스템에도 가격 경쟁력 있게 부착할 수 있게 될 것이다. 그 외의 대표적인 센서 노드들로는 Rockwell의 WINS 노드, Sensoria의 WINS NG 2.0, UCLA의 iBadge, MIT의 u-AMP 시리즈 등이 있으며, Intel, Crossbow, Moteiv와 같은 회사에서 추가적으로 센서 노드 하드웨어들을 출시하고 있다. 그리고 국내에서는 옥타컴에서 센서 네트워크 응용을 위해 Nano-24 센서 노드 하드웨어 플랫폼을, MAXFOR에서는 TIP 30CM과 TIP 50CM 하드웨어 플랫폼을 발표하였다.

### 1. 버클리대 Mote 시리즈

1999년 버클리대에서 WeC라는 첫 번째 Mote 플랫폼을 개발한 후, 매년 rene, dot, mica, mica2, mica2 dot과 같은 센서 노드 하드웨어가 버클리대에 의해서 공개되었으며, 이들 하드웨어 플랫폼은 미국의 Crossbow란 회사에 의해서 상업화 되어서 시장에 공급되고 있다. 그 외에도 버클리대의 TinyOS를 지원하는 호환 하드웨어 플랫폼으로 인텔의 iMote, ETH의 btNode, Moteiv의 Telos가 개발되었다. 표 1에 현재까지 개발된 Mote 시리즈 관련 하드웨어 플랫폼을 선별하여 정리하였다. 특히 Telos와 MICAz는 센서 네트워크의 표준 Radio 인터페이스로 유력한 IEEE 802.15.4 저속 WPAN(Wireless Personal Area Network) 표준을 지원하는 Chipcon사의 ZigBee 칩을 이용하고 있어 많은 주목을 받고 있다.

공개 하드웨어인 MICA는 그림 1과 같이 다양한 보드로 구성된다. 프로세서 보드는 마이크로 컨트롤러와 무선 인터페이스를 탑재한 주 보드이며, 센서 보드는 온도, 조도, 지자기 센서 등과 같이 센서를 탑재한 보드로써 기상센서 보드와 모터/서보 제어보드 등 다양한 보드

표 1. Mote 시리즈 하드웨어 사양

Mote	WeC	RENE	DOT	MICA	MICA2	iMote	btNode	Telos	MICAz
Released	1999	2000	2001	2002	2003	2003	2003	2004	2004
Clock	4MHz	4MHz	4MHz	4MHz	7MHz	12MHz	12MHz	20MHz	7MHz
CPU	Atmel	Atmel	Atmel	Atmel	Atmel	ARM	Atmel	Motorola	Atmel
Flash (code, KB)	8	8	16	128	128	512	128	60	128
RAM(KB)	0.5	0.5	1	4	4	64	4	4	4
Radio (kBaud)	10	10	10	40	40	460	460	250	250
Radio Type	RFM	RFM	RFM	RFM	ChipCon	Zeevo BT	Ericsson BT	ZigBee	ZigBee

가 제공된다. 2003년 미국 Crossbow사는 버클리대의 협력을 받아 상용 MICA2를 개발하여 판매를 하고 있다. MICA와 MICA2의 가장 큰 차이점은 RF 인터페이스가 달라졌다는 것이다. MICA는 916MHz 라디오 트랜시버를 사용하였는데 MICA2는 433/868/915MHz의 다양한 무선 밴드를 지원한다. 마이크로 컨트롤러도 MICA의 ATmega103L에서 ATmega128L로 변경되었으며 전원부와 관련된 모니터링 기능이 개선되었다.

iMote는 버클리대의 또다른 산업체 파트너인 인텔에서 개발된 Mote 시리즈로써 ARM 기반의 마이크로 프로세서를 사용한다. 표 1에서 보듯이 다른 노드에 비해서 메모리가 풍부한 것이 가장 큰 장점이다. 그 외에 블루투스를 사용하여 통신 속도도 다른 노드에 비해 뛰어나며 디지털 센서만을 이용한다. 블루투스를 사용하는 하드웨어로 스위스의 ETH에서 만든 btNode도 주목할 만하다.



그림 1. MICA 컴포넌트 : 프로세서 보드, 센서 보드, 초음파 보드



그림 2. Crossbow사의 MICAz 센서 노드

Moteiv 회사에서 2004년 상반기에 출시된 Telos 는 IEEE 802.15.4 표준을 지원하는 최초의 센서 노드 이다. Chipcon사에 의해서 개발된 CC2420 ZigBee 무선칩을 이용하며, 최대 50m까지 통신이 가능하다. Crossbow Technology에서 최근 원격 센서 모듈인 MICAz를 발표했다. 대체적으로 MICAz의 하드웨어 스펙은 MICA2의 스펙을 따르고 있지만, MICAz에는 위의 Telos와 마찬가지로 Chipcon사의 CC2420 RF 송수신 모듈이 포함되어 있다.

의 CC2420 RF 모듈을 채택하였다. 그리고 배터리 수명을 고려하여 상대적으로 낮은 하드웨어 전류 소모를 목적으로 설계되었고 배터리 모니터링 기능을 갖는다. 또한 조도 센서, 온도 센서, 습도 센서, 적외선 센서, 가스 누출 인식 센서, 초음파 센서, 그리고 가속도 센서로 구성된 다양한 센서 모듈들을 탑재하고 있다.

### III. 센서 네트워크 운영체제

#### 2. Nano-24 센서 하드웨어 플랫폼

국내에서 개발된 센서 노드중 주목할 만한 것은 옥타컴사에서 개발한 Nano-24 센서 하드웨어 플랫폼으로 뒷장에서 소개될 한국전자통신연구원(ETRI)에서 개발한 센서 네트워크 운영체제인 Nano-Qplus가 포팅된 유일한 플랫폼이며, TinyOS도 포팅이 되어 있다. 그림 3은 Nano-24를 구성하는 여러 모듈들을 보여준다. Nano-24의 메인 보드는 Crossbow사의 MICAz 메인보드와 유사한 하드웨어 구성과 기능 모듈들을 갖고 있다. 또한 MICAz와 마찬가지로 Nano-24는 2.4~2.4835GHz ISM 대역에서 동작하는 Chipcon사

센서 네트워크를 구성하는 각각의 센서 노드는 마이크로 컨트롤러를 내장한 소형 컴퓨터 시스템으로 센싱 응용 처리와 노드들간 통신 등을 위하여 운영체제를 필수적으로 요구한다. 센서 네트워크의 동작 환경 및 구조가 기존의 네트워크에서와는 상이하기 때문에 센서 네트워크의 운영체제 설계 시에도 특별한 조건들이 고려되어야 한다. 컴퓨팅 능력, 메모리, 전원 등 극도의 제한된 자원 하에서 무선통신 및 시스템 제어 기능을 제공함은 물론, Context-awareness, 즉 데이터 중심의 프로그래밍, 환경 및 응용 프로그램의 다양한 변화에 대한 능동적인 대처를 지원할 수 있어야 하고, 다양한 이벤트에 반응하여 실시간으로 동시 처리할 수 있어야 한다. 또한,

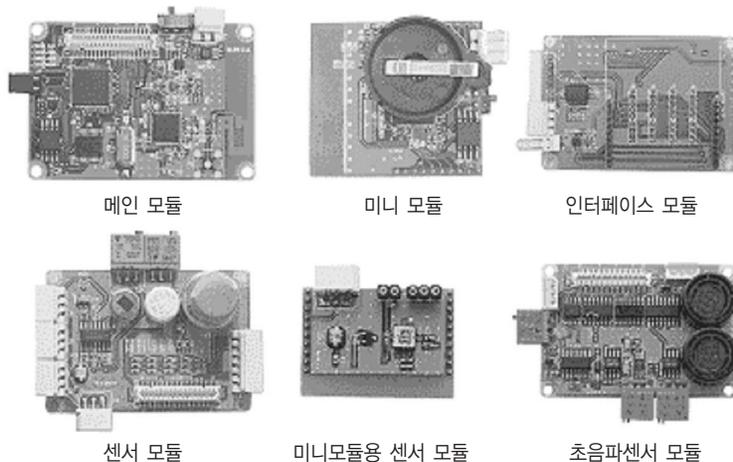


그림 3. Nano-24 센서 하드웨어 플랫폼

새로운 센서 하드웨어의 변화에 신속적으로 대처할 수 있도록 모듈 방식의 설계가 선호되고, 열악한 환경과 칩 입을 감내할 수 있으며, 분산된 환경을 지원하여 네트워크 내의 동기화 및 위치 인식 기능도 추가적으로 요구된다. 대표적인 이벤트 기반 센서 네트워크 운영체제인 UC Berkeley의 TinyOS와 한국전자통신연구원(ETRI)에서 개발된 Nano-Qplus 운영체제를 소개한다.

## 1. 버클리대의 TinyOS

TinyOS는 이벤트 발생에 의한 상태 천이 방식을 채택한 상태 머신 기반의 프로그래밍 개념을 사용한 운영체제로써, 제한된 메모리 공간의 효율적인 이용과, 프로세싱의 동시성 등을 지원해 주는 운영체제이다. TinyOS에서는 시스템 자원의 제약들 때문에 기존의 IP 프로토콜, 소켓, 쓰레드 개념들을 사용하지 않는다. 이러한 TinyOS의 특징을 크게 아래 3가지로 구분해 볼 수 있다.

첫째, 재사용 가능한 소프트웨어 컴포넌트 기반의 운영체제이며, 응용 프로그램은 하드웨어 컴포넌트의

입/출력을 연결하듯 소프트웨어 컴포넌트의 입/출력 인터페이스를 연결함으로써 작성된다. 둘째, 상태 머신 기반의 구조를 가지는 운영체제로, 각각의 상태는 TinyOS의 컴포넌트가 해당된다. 각 컴포넌트의 명령(command)과 이벤트 처리기는 한 상태에서 다른 상태로 빠르게 전이를 일으키며, 기본적으로는 하드웨어의 신호 처리와 같은 특성을 가지므로 적은양의 부가처리와 논-블러킹의 특성을 지닌다. 셋째, 센서노드의 중요한 요구사항의 하나인 저전력 소모를 구현하기 위해 사용되지 않는 CPU의 사이클 동안 휴지 상태로 들어가 전력소모를 줄인다.

내부적으로 TinyOS 응용 프로그램은 그림 4와 같이 커맨드와 이벤트를 계층적으로 스케줄하는 저전력 스케줄러와 컴포넌트의 상태 천이 그래프로 구성이 된다. 그림 4에서 보이는 응용은 센서 노드가 온도와 조도를 센싱하여 무선 인터페이스를 통해 베이스노드로 전송하는 단순한 예제이며, 실제 3450바이트의 코드와 226바이트 데이터만을 사용하는 초소형 프로그램이다. 여기서 Active Messages 컴포넌트는 센서 노드와 베이스 노드와의 통신을 담당한다.

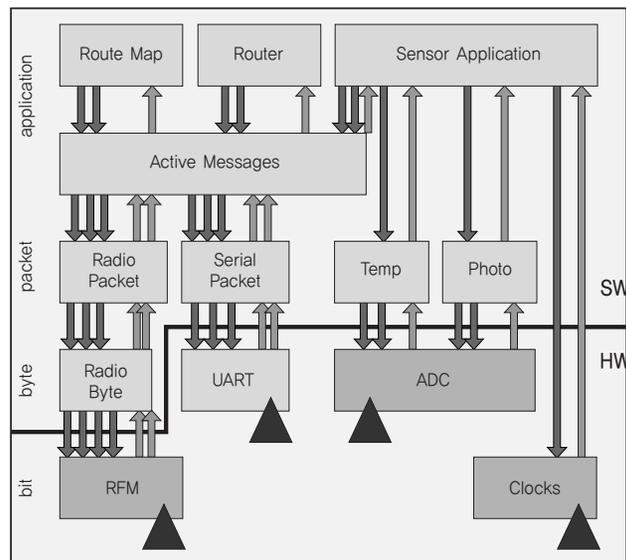


그림 4. 컴포넌트 기반 TinyOS 응용 예

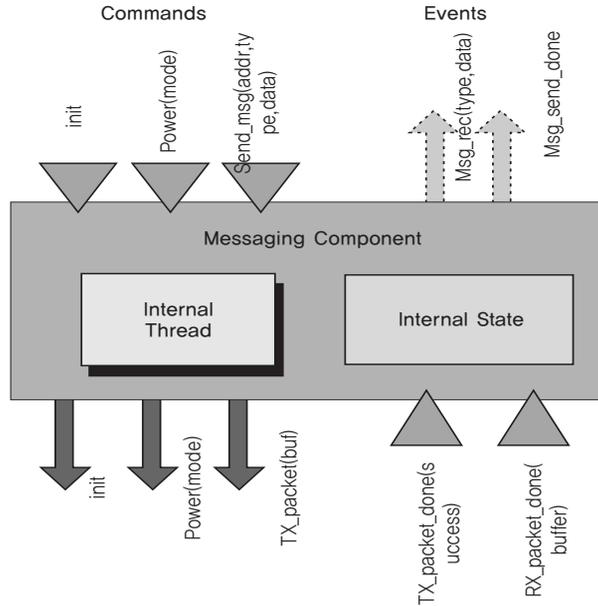


그림 5. TinyOS의 메시징 컴포넌트

그림 5의 메시징 컴포넌트 내부 구성 예에서 나타난 바와 같이 각 컴포넌트의 인터페이스는 입력 커맨드와 출력 커맨드, 입력 이벤트와 출력 이벤트 4가지로 구성된다. 입력 커맨드는 자신의 컴포넌트에서 서비스하는 함수로 상위 컴포넌트가 해당 컴포넌트의 기능이 필요할 때 호출하는 것이며, 출력 커맨드는 하위 컴포넌트의 서비스를 호출할 때 사용한다. 입력 이벤트는 해당 컴포넌트의 이벤트 핸들러에 의해 처리되며, 하위의 컴포넌트가 해당 컴포넌트에 신호를 넘겨주기 위하여 사용되는 것이며, 출력 이벤트는 해당 컴포넌트가 상위의 컴포넌트에 신호를 전달하기 위해 사용된다.

## 2. ETRI Nano-Qplus(나노 OS)

Nano-Qplus 운영체제는 한국전자통신연구원(ETRI)에서 개발중인 운영체제로 다음의 특징을 가진다.

- 에너지 소모를 최소화하기 위하여 네트워크를 구성하는 노드들간에 동기된 시간을 기반으로 주기

적으로 휴면(Sleep)과 활성(Active)모드를 반복하면서 메시지를 효율적으로 처리

- 제한된 메모리의 사용을 최소화하기 위해 멀티 스레드간 스택 공유
- 멀티 스레드 스케줄러(FIFO 방식, 시간 기반 라운드 로빈 방식, 우선 순위에 따른 스레드 선점 방식)로 다양한 무선 통신 방식(433RF, ZigBee)을 지원
- 표준형 및 마이크로 임베디드 OS와 동일한 API 세트를 지원하여 기존 시스템 프로그램 개발자에게 친밀감 제공

Nano-Qplus는 nHAL(nano Hardware Abstract Layer) 계층에서 센서 노드의 하드웨어(LED, CLOCK, RFM, UART, ADC 등)를 컴포넌트별로 추상화시켜 상위 기능 모듈들에게 하드웨어 컴포넌트 제어를 용이하게 한다. 그리고 OS의 주요 기능을 모듈단위로 구현하여 전체 시스템을 구성하는 단계에서 추가/삭제가 가능하도록 한다. 그리고 각각의 OS 모듈에서 상위 응용에게 API를 제공하도록 한다. 각 모듈의

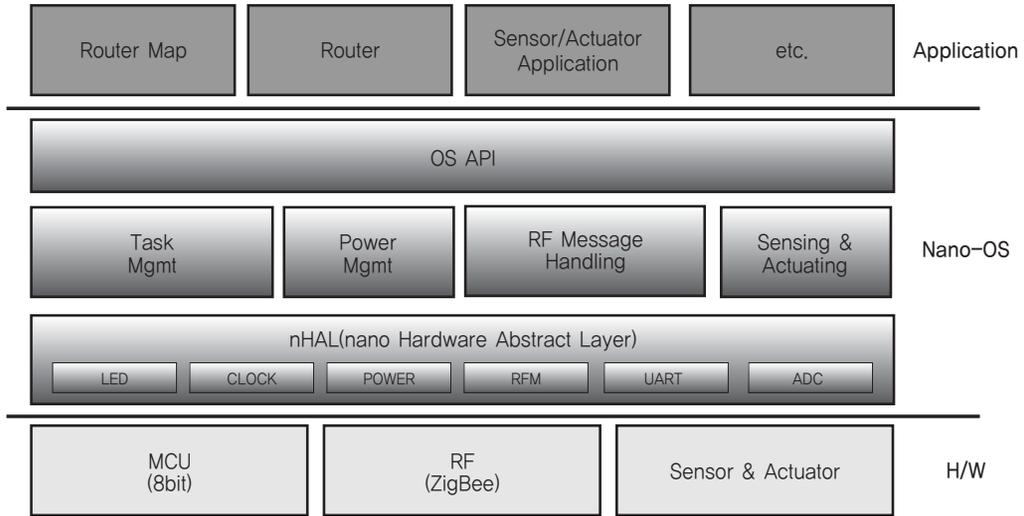


그림 6. Nano-Qplus 구조

기능을 다음과 같이 요약하였다.

- 동적 전원 관리(Dynamic Power Management) 모듈 : 시스템이 동작하는 과정에서 처리할 태스크가 없는 경우에 전원 소비가 낮은 실행 모드로 전환하여 배터리 에너지를 절약하도록 하여 노드 뿐만 아니라 네트워크 전체의 생존 시간을 늘리는 역할을 수행한다.
- 시간 동기화(Time Synchronization) 모듈 : 센서 노드들이 시간에 대한 동기를 유지하는 기능을 제공하는 모듈이다. 각 센서 노드들의 동기화 모듈간의 협력 작업을 통해 전체 센서 네트워크의 동기화를 실현할 수 있다.
- 라우팅(Routing) 모듈 : 외부의 어떤 중재나 개입 없이 센서 노드들 스스로가 네트워크를 형성하여 메시지를 교환할 수 있도록 하는 모듈로서 초기에 네트워크가 형성된 이후에도 나중에 새로운 노드가 추가되더라도 네트워크가 동적 재구성이 가능하도록 하는 역할을 수행한다.
- 스레드간 통신 및 동기화 모듈 : 나노 OS는 다중 스레드기반 운영체제로서 스레드들 서로간 협업을 위해 메시지 교환이 가능하도록 하며 또한 여러 스레드들이 공유하는 자원 접근시 충돌을 막기

위해 동기화를 맞추어 주는 역할을 수행한다.

#### IV. 결론

USN 기술이 군사 목적에서부터 가정, 물류/유통, 행정, 복지, 그리고 환경까지 사회 전반에 걸쳐 다양한 분야에 적용됨으로써 가져다주는 유용함은 우리 모두가 인식하고 있다. 따라서 센서 네트워크 기술과 관련된 시장 규모는 자동 인식 기능을 제공하는 RFID 기술과 융합되어 앞으로 천문학적으로 발전할 것이다. 그럼에도 불구하고 센서 네트워크가 실제 산업화가 되기 위해서는 풀어야 할 숙제가 많다. 아직 표준으로 공인된 센서 네트워크 기술이 없기 때문에 대량 생산과 상용 서비스와 관련하여 실제 제품 개발 단계에서 어려움이 존재한다. 이를 인식하여 현재 USN 관련 표준 제정을 위해 TTA 산하의 PG108 WG3과 PG311에서 표준화 작업을 진행중이기 때문에 조만간 국내 표준이 제정되기를 기대한다. 그리고 IT 인프라가 잘 구축된 우리 나라의 상황을 바탕으로 정부, 관련 기관 및 산업체, 학계가 서로

연계되어 노력한다면 아직 강자가 존재하지 않는 USN 시장을 선점하고 이끌어 갈 수 있을 것이다.

## 참고문헌

- [1] P. Levis, S. Madden, D. Gay, J. Polastre, R. Szewczyk, A. Woo, E. Brewer and D. Culler, The Emergence of Networking Abstractions and Techniques in TinyOS, Proceedings of the First USENIX/ACM Symposium on Networked Systems Design and Implementation(NSDI 2004)
- [2] 김대영, 양진영, 이인선, 유성은, 성종우, Tomas Sanchez Lopez, 도윤미, “무선 센서 네트워크를 위한 임베디드 소프트웨어 기술”, 전자공학회지, 2004년 11월
- [3] 김대영, 홍승기, 김현학, Do Thu Thuy, Long Minh Pham, “나노 OS의 저전력 통신을 위한

협업 메시지 엔진 개발”, 한국전자통신연구원 최종보고서, 2004년 11월

- [4] Berkeley Sensor and Actuator Center, <http://www-bsac.eecs.berkeley.edu/>
- [5] Berkely TinyOS 프로젝트, <http://today.cs.berkeley.edu/tos/>
- [6] 스마트 더스트 프로젝트, <http://www-bsac.eecs.berkeley.edu/~warneke/SmartDust/index.html>
- [7] Nano-24 옥타컴 센서 노드, <http://www.octacomm.co.kr>
- [8] Joseph Polastre, Robert Szewczyk, Cory Sharp, David Culler, “The Mote Revolution : Low Power Wireless Sensor Network Devices,” in Proceedings of Hot Chips 16 : A Symposium on High Performance Chips, Aug. 2004 