

센서 네트워크 노드 플랫폼 및 운영체제 기술 동향

Technical Trend of Sensor Network Node Platform & OS

임베디드 S/W 기술 동향 및
연구 개발 현황

박승민 (S.M. Park)

임베디드S/W플랫폼연구그룹 그룹장

목 차

-
- I. 서론
 - II. USN 시장 동향
 - III. 센서 노드 하드웨어 플랫폼
 - IV. 센서 네트워크 운영체제
 - V. 결론

유비쿼터스 시대가 다가옴에 따라 핵심 기술 중 하나인 유비쿼터스 센서 네트워크 (USN)에 대한 관심이 높아지고 있다. 센서 노드를 이용한 다양한 정보를 수집 처리할 수 있는 USN 기술은 물류/유통, 환경 감시, 홈 오토메이션 등 다양한 분야에 적용될 수 있기 때문에 갈수록 그 중요성은 높아질 것으로 예상된다. 이에 현재 USN의 시장 동향을 간단히 살펴보고, 센서 노드의 핵심이 되는 센서 노드 플랫폼과 센서 노드에 들어가는 센서 네트워크 운영체제의 기술 동향에 대해 살펴보고자 한다.

I. 서론

유비쿼터스 센서 네트워크(USN)는 정보통신부의 IT839 전략의 3대 첨단 인프라 중 하나로 그 중요성이 날로 증가하고 있다. 이미 인텔도 기술 로드맵 중에 센서 네트워크를 기반 기술로 예측할 정도로 미래 유비쿼터스 사회를 구현하기 위한 핵심 기반 구조로 주목 받고 있다[1].

유비쿼터스 센서 네트워크란 기존 인간과 컴퓨터 간의 커뮤니케이션에 일상 생활에 산재된 사물과 물리적 대상을 추가시켜 협력 네트워크를 구성하는 것으로, 필요로 하는 모든 곳에 수 많은 센서 노드들을 부착하여 자율적으로 정보를 수집, 관리 및 제어하는 시스템이다. 즉 물리 공간에 빛, 소리, 온도, 움직임 같은 물리적 데이터를 센서 노드에서 감지하고 측정하여 중앙의 기본 노드로 전달하는 구조를 가진 네트워크이다.

이런 인프라를 이용하여 다양하고 편리한 새로운 서비스를 제공하는 것이 유비쿼터스 사회의 궁극적인 목표로 USN은 u-Korea 실현을 위한 초석으로 유비쿼터스 시대를 이끌어갈 핵심 기술 중 하나로 각광받고 있다.

센서 네트워크는 물리적 세계와 디지털 세계를 연결할 수 있는 특징 때문에 많은 분야에 응용될 수 있다. 예를 들어 홈 네트워크에서 집안의 침입 감지나 가스 센서를 이용해 가스 안전 모니터링 등을 수행할 수 있으며, 산업 현장에서는 위치 인식 서비스나 물류 관리 등에 사용될 수 있다. 또 지능형 환경 모니터링으로 강수량 측정이나 산불 감시 등에 쓰일 수 있으며 각종 의료시스템이나 과학 분야에도 사용될 수 있어 그 가능성은 무궁무진하다.

센서 네트워크의 핵심 기술 요소로는 하드웨어 플랫폼과 노드에 들어가는 초소형 운영체제 기술을 들 수 있다.

하드웨어 플랫폼은 크게 센싱 데이터를 수집하기 위한 센서 노드, 데이터를 전달하기 위한 라우터 노드, base station과 통신을 위한 싱크 노드 등으로 나눌 수 있다. 센서 네트워크 응용을 위한 하드웨어

는 아직까지 표준화가 이뤄지지 않았으며 상업적 목적으로 대량 생산되어 사용되는 제품도 찾아보기 힘들다. 현재 하드웨어 플랫폼은 국내외에서 다양하게 개발되고 있는데 이는 본문에서 자세히 알아보기로 하겠다.

또한 해당 하드웨어 플랫폼에 들어가는 센서 네트워크용 OS 개발도 활발히 이뤄지고 있다. 센싱 정보를 처리하고 노드들 간의 통신 등을 위해서는 센서 노드를 위한 운영체제가 필수적으로 요구된다. 제한된 메모리와 CPU 자원을 가진 센서 노드의 특성상 센서 노드 OS는 극소형의 크기를 가져야만 하며 기존 RTOS보다 더 많은 제약사항을 가진다. 현재 이러한 특징을 수용하는 센서 OS의 경우 버클리의 TinyOS나 한국전자통신연구원의 나노 Qplus 등이 개발되어 사용되고 있다.

본문에서는 앞서 얘기한 USN의 발전 방향과 시장 동향에 대해 간단히 살펴보고 센서 네트워크의 핵심 기술인 하드웨어 플랫폼과 초소형 센서 네트워크 운영체제에 대한 기술 동향을 살펴보기로 한다.

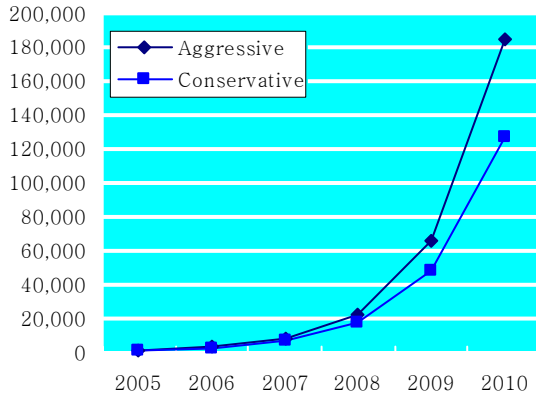
II. USN 시장 동향

현재 상업적으로 대량 생산되어 구체적으로 USN들이 쓰이고 있지는 않지만 최근 몇 년간 USN 시장은 점차 커져가고 있다. 점차 센서 네트워크를 위한 표준안이 나오고 있으며 많은 기관들이 상업화를 위한 솔루션을 제안하고 있다.

2005년 현재 노드 당 가격은 10~100달러 선이지만 시장이 더 커져 안정화가 되는 2007~2010년에는 1~10달러 정도로 진입할 것으로 예상된다[2].

또, (그림 1)에서 보는 바와 같이 현재 모든 분야에 백만 개가 조금 넘게 사용되고 있는 데 반해 2010년쯤에는 1억3천~1억8천 개의 노드가 사용될 것으로 예상되고 있다[3].

이는 다량의 센서를 배치하는 커다란 스케일의 WSN을 이용한 응용이 등장하고 있음을 나타낸다. 실제 Coronis System의 경우 프랑스 전 도시를 네트워크화 하기 위해 25,000개의 노드를 사용했으며



<자료>: On World, 2005.

(그림 1) 센서 노드 증가 추이 예측

50,000노드를 배치하기 위해 준비중이다. Eka Systems의 경우도 올해 30,000노드를 사용할 예정이며, 누리 텔레콤도 가스, 전기 원격검침을 위해 올해 60,000노드를 사용할 예정이다.

RF 모듈 같은 경우 2010년 1억6천 개의 802.15.4 RF 모듈이 생산될 것으로 추정되며, 노드의 65% 정도가 802.15.4/ZigBee를 사용할 것으로 예측된다. 이는 ZigBee가 USN에 사실상 표준으로 자리잡았음을 뜻한다.

Ⅲ. 센서 노드 하드웨어 플랫폼

1. 하드웨어 플랫폼 기술 동향

USN의 가장 큰 장점은 센서 네트워크 노드들이 독자적으로 네트워크를 구성한다는 것에 있다. 즉, 무선 네트워크 기술을 이용해 애드-혹(ad-hoc) 네트워크를 구성해야 되는데 지금까지 개발된 블루투스(Bluetooth)나 무선랜(wireless LAN) 등의 무선 네트워크 기술들은 반드시 컴퓨터나 PDA 같은 고급 컴퓨팅 장치를 필요로 했다.

하지만 센서 네트워크의 특성상 센서 노드는 가격과 유지 보수 문제 등으로 고급 컴퓨팅 장비를 사용할 수 없다. 그래서 센서 노드를 위한 하드웨어 플랫폼은 다음과 같은 특징을 가져야 한다.

- 한번 배치되면 유지보수가 어렵기 때문에 강한 구조에 저전력으로 설계되어야 한다.
- 응용 분야에 따라 크게 달라질 수 있기 때문에 어떠한 구조에도 효과적으로 사용될 수 있도록 유연성과 모듈성을 갖추고 있어야 한다.
- 유비쿼터스 컴퓨팅 환경에 맞게 무선 네트워크 기술을 지원해야 한다.
- 많은 센서 노드가 산재되기 위해 가격이 저렴하고 크기가 작아야 한다.

이와 같은 특징을 만족시키기 위해 많은 하드웨어와 노드들이 개발되고 있다. 이 중 전력을 가장 많이 소모하고 기능적으로 중요한 핵심 부품인 CPU, 무선 모듈에 대해 간단히 알아보고 개발되어 사용중인 노드들에 대해서 다루도록 한다.

가. MCU

스마트 센서 노드에 들어가는 저전력 CPU의 경우 Atmel[4], ARM[5], Motorola 등에서 개발되고 있다.

기존 국내 MCU 시장을 보면 마이크로칩스사의 PIC 계열과 인텔의 유명한 8051이 공개된 회로와 애플리케이션이 많아 사용되었었다. 하지만 센서 네트워크에서는 에너지 효율성을 위해 대부분 8bit 계열에 낮은 클럭을 가지는 RISC CPU가 많이 사용된다.

최근 들어 Atmel의 AVR ATmega 시리즈가 널리 이용되고 있는데 이는 PIC처럼 RISC 및 하버드 구조를 가지는 고성능 8bit 마이크로 컨트롤러면서 프로그램용 코드 메모리로 플래시 메모리를 내장하고 있어 쉽게 반복적인 프로그래밍이 가능하기 때문이다. 또한 8bit RISC 계열의 MCU인 AVR은 한 클럭 당 한 명령어를 처리하기 때문에 동일 클럭에서 동작 시 PIC나 8051을 채택한 MCU보다 몇 배 이상 빠르다. 그래서 가장 늦게 출시되었음에도 불구하고 단시간 내에 많은 인기를 얻었으며 현재 USN에 가장 많이 사용되고 있다.

나. RF 모듈

센서 네트워크 노드를 위한 RF 모듈로는 IEEE 802.15.4/ZigBee가 de facto standards로 사실상 표준으로 사용되고 있다[2]. ZigBee는 저가, 저전력을 특징으로 IEEE 802.15.4의 표준화된 PHY/MAC 레이어를 기반으로 하여 상위 프로토콜 및 애플리케이션을 규격화한 단거리 무선 통신에 적합한 스펙이다.

ZigBee가 센서 네트워크에 많이 쓰이는 이유 중 하나는 바로 저렴한 가격이다. 시스템에 ZigBee를 구현하는 비용이 2달러도 안되기 때문에 많은 센서 노드를 설치해야 하는 USN에 적합하다. 또 802.11x wireless LAN이나 IEEE 802.15.3 HR-WPAN, Bluetooth보다도 구현 측면에서 훨씬 간단한 것도 ZigBee의 장점으로 들 수 있다. 실제 ZigBee 실행 코드 사이즈는 32k 바이트 이하로 구현이 가능하며 RFD와 같은 단순한 기능을 가진 노드의 경우 4KB 이하의 사이즈로 구현이 가능하다[6]. 전력 소모를 극소화한 것도 가장 큰 장점으로 들 수 있는데 네트워크 안에 하나의 coordinator를 명하여 송/수신의 활동에 필요한 경우에만 동작하고 나머지 시간은 sleep 모드로 보내기 때문에 한 번 배터리를 장착하면 최대 2~3년 정도 사용이 가능하다.

주요 ZigBee 반도체 업체로는 국내의 경우 레이

디오필스[7]가 ZigBee 칩을 생산하고 있으며 외국 의 경우 Chipcon, Motorola, Texas Instruments, ST Microelectronics, Ember 등이 있다.

2. 센서 노드 하드웨어 플랫폼

가. Crossbow Mica Series

미국 국방성의 DARPA 프로젝트의 스폰서를 받아 개발된 미국 버클리 대학 Mote 시리즈는 미국 정부와 관련 대학과 기업의 노력으로 가장 널리 사용되는 하드웨어 플랫폼으로 자리잡고 있다[8]. 이는 하드웨어뿐만 아니라 TinyOS라는 센서 네트워크용 OS와 각종 시뮬레이터 및 공개 애플리케이션을 통한 인프라가 구축되었기 때문이다.

1999년 처음으로 WeC라는 플랫폼이 개발되었으며 매년 Rene, dot, Mica, Mica2, MicaZ와 같은 센서 노드 하드웨어가 개발되어 왔다. 해당 노드들은 버클리 출신이 만든 벤처회사 Crossbow를 통해 상업화되어 시장에 공급되고 있다.

<표 1>은 Mote 시리즈의 하드웨어 사양으로 그 동안 발표된 메인 보드별 스펙을 나타낸다. 이 메인 보드에 온도, 조도, 지자기 센스 등과 같은 센서를 탑재한 센서 보드를 스택 형식으로 장착할 수 있다.

CPU 스펙의 경우 앞서 얘기한 Atmel의 Atmega

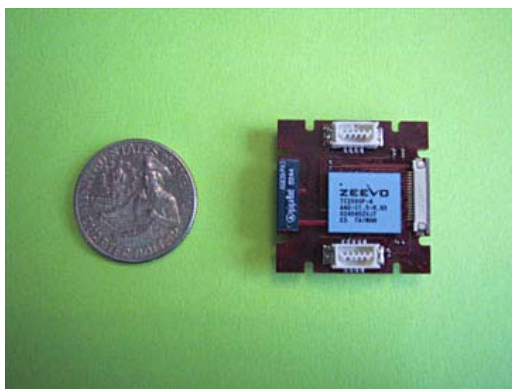
<표 1> Crossbow사의 센서 노드 하드웨어 스펙

Mote	WeC	Rene	Rene2	dot	Mica	Mica2	MicaZ
Released	1998	1999	2000	2001	2002	2003	2004
Clock	4MHz	4MHz	4MHz	4MHz	4MHz	7.37MHz	7.37MHz
CPU	Atmel 90LS8535	Atmel 90LS8535	Atmel Atmega164	Atmel Atmega163	Atmel Atmega103L	Atmel Atmega128L	Atmel Atmega128L
Program Memory(KB)	8	8	16	16	128	128	128
RAM(KB)	0.5	0.5	1	1	4	4	4
RF Module	RFM TR1000	FRM TR1000	FRM TR1000	FRM TR1000	RFM TR1000	Chipcon CC1000	Chipcon CC2420
Modulation	OOK	OOK	OOK	OOK	ASK	FSK	O-QPSK
Data Rate(KB)	10	10	10	10	40	38.4	250
TinyOS	-	V0.6.1	V0.6.1	V1.0 up	V1.0 up	V1.0 up	V1.1.7 up

시리즈를 사용하고 있으며 노드별 가장 큰 차이는 RF 인터페이스에서 찾아볼 수 있는데 Mica는 916 MHz의 라디오 트랜시버를 사용한 데 반해 Mica2는 433/868/915MHz의 다양한 무선 밴드를 지원한다. 또 MicaZ 같은 경우 Mica2가 비슷하지만 RF 모듈을 Chipcon사의 CC2420을 사용하여 ZigBee를 지원한다.

나. Intel Mote(iMote)

역시 버클리대의 산업체 파트너인 인텔에서 개발된 Mote 시리즈로 ARM 기반의 마이크로 프로세서를 사용한다. 이는 센서 네트워크의 응용분야를 조사하고 다양한 응용을 위해 성능을 향상시킨 디자인이다. 센싱 정보의 복잡한 계산이나 라우팅 등의 상위 레벨의 정보 처리, 보안 등에 더 치중하기 위해 Intel의 iMote는 강력한 32bit ARM7TDMI CPU를 사용했다. 12MHz의 클럭으로 기존 Mica Mote 보다 4배 정도의 성능을 보이며, 메모리 측면에서도 Mica 시리즈보다 훨씬 큰 512KB의 프로그램 메모리와 64KB의 RAM을 사용한다[9]. RF 모듈은 Zeevo사의 2.4GHz band의 Bluetooth를 사용하여 최대 720kbps의 전송률을 가지기 때문에 250kbps의 전송률을 가지는 ZigBee보다 훨씬 많은 데이터를 전송할 수 있으며, 크기는 (그림 2)에서 보는 바와 같이 3×3cm 정도로 아주 작다[10].



(그림 2) Intel Mote Prototype(original size: 3×3cm)

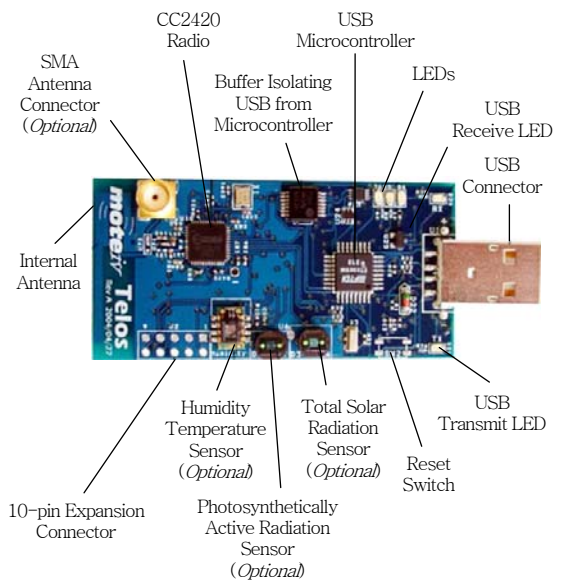
센서 네트워크 OS의 경우 Mica Mote 시리즈와 같이 TinyOS를 사용하지만 ARM instruction set에 맞게 포팅하고 새로운 로우 레벨의 디바이스 드라이버를 지원한다.

다. Moteiv Telos, Tmote SKY

2004년 초에 출시된 Telos는 IEEE 802.15.4 표준을 지원하는 최초의 센서 노드이다. MCU로는 Texas Instruments의 MSP430을 사용하며 Chipcon사에서 개발된 CC2420 ZigBee 칩을 RF 모듈로 사용한다.

Telos의 경우 Revision A, B가 있는데 두 제품의 차이는 RAM, flash가 틀리고 거의 비슷하다. 일반적으로 Telos라고 하면 rev A를 지칭하며, rev A는 현재 생산이 중단되었다. 그리고 Telos rev B의 경우 TMote Sky라는 이름으로 변경되어 팔리고 있다((그림 3) 참조)[11].

Telos에서 사용한 MSP430의 장점으로는 wake-up time이 Mica, Mica2가 180μs인데 비해 6μs면 가능하며 전력 소모 역시 active power일 때 15 mW, sleep power일 때 15μW로 Mica2의 33mW,



(그림 3) Moteiv Telos Hardware

75 μ W에 비해 소비 전력이 훨씬 적다[12]. 또 ATmega128 MCU가 2.7V 이하에서 제대로 작동을 안하는데 반해 MSP430은 1.8V까지 떨어져도 작동이 가능하다.

Tmote SKY 역시 센서 노드 OS로는 TinyOS가 사용되고 있다.

라. Maxfor TIP

한국전자부품연구원(KETI)에서 개발한 무선 센서 네트워크 플랫폼으로 TIP가 있다[13].

TIP 3X 시리즈는 900MHz대의 RF를 사용하여 38Kbaud의 데이터율을 가지며 8bit MCU로 이뤄져 있다. TIP 5X 시리즈는 2.4GHz대의 RF를 사용하며 TIP50은 16bit RISC CPU로, TIP51은 Telos에서 사용되는 MSP430 MCU를 사용한다. 역시 OS로는 TinyOS를 사용한다.

마. 옥타컴 Nano-24

국내에서 개발된 센서 노드 중 유일하게 ETRI에서 개발한 센서 네트워크 운영체제인 나노 Qplus가 포팅되어 있는 플랫폼이다. Nano-24는 Crossbow의 MicaZ 보드와 유사한 구성을 가지며 RF 역시

Chipcon사의 CC2420을 사용하여 ZigBee를 지원한다.

Nano-24 보드의 특징은 (그림 4)에서처럼 조도, IC 온도, 습도, 적외선, 가스, 초음파, 가속도 센서 등 많은 센서 노드를 제공하며 상대적으로 낮은 하드웨어 전류 소모를 목적으로 설계되었고 배터리 모니터링 기능을 갖는다.

바. 한백전자 ZigbeX Mote

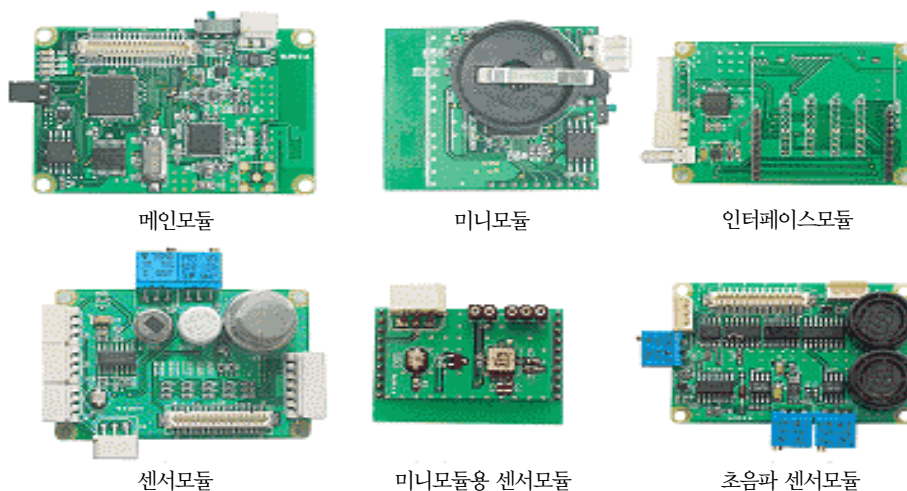
MCU는 ATmega128 CPU를 사용하고 RF는 Chipcon의 CC2420을 사용한다. 특징으로는 기본으로 온도, 조도, 습도 센서, RTC를 모드에 장착하고 있다.

다양한 옵션 센서 모듈을 지원하는데 기상센서, 방법센서, 홈 오토메이션 센서 모듈, 위치추적 센서, 구조 역학 센서, 바이오 센서, 자동차 센서 등의 응용 모듈을 제공한다.

센서 노드 OS의 경우 TinyOS와 μ C/OS를 지원한다.

사. 휴인스 UStar-2400

UStar-2400은 Chipcon CC2420 RF 모듈을 사용하여 2.4GHz ZigBee 통신을 가능하게 하고 At-



(그림 4) Nano-24 센서 노드 제품 구성

mel의 Atmega128L 프로세서를 장착하고 있다. 저전류 소모로 배터리 모니터링 기능을 제공하며 저전력 설계 보드를 사용 low-power management 기능을 제공한다.

TinyOS와 TinyDB를 지원하며 UStar-Dev라는 emulator를 포함한 개발환경과 센서 네트워크 응용 프로그램 등을 지원해준다.

IV. 센서 네트워크 운영체제

1. 기술 동향

센서 네트워크용 OS는 저전력 무선 네트워크 지원 및 소형 하드웨어 채택 등을 요구하고 있다. 이를 위하여 설계 시 요구되는 기술 동향은 다음과 같다.

첫째, 센서 네트워크에서는 자원이 제한적이기 때문에 센서 노드의 저전력 통신 특성을 갖춰야 하며, 프로세서의 메모리 영역의 효율적인 관리를 수행하여야 한다.

둘째, 센서 네트워크의 전체적인 에너지 효율을 높이기 위하여 센서 노드들 간의 시간 동기화가 이루어져야 한다.

셋째, 센서 노드는 일반적으로 제한된 처리 능력과 크기가 작은 하드웨어 구조를 가지며, 또한, I/O 장치를 지능형 제어가 아닌 원시적인 접근 방식에 의해 접근하여야 하기에 이러한 I/O의 하드웨어적인 제약 사항들을 고려해야 한다.

넷째, 센서 노드들이 실제 환경에 뿌려진 이후로 유지 보수가 어렵고, 환경의 영향을 많이 받을 수 있기 때문에, 이러한 문제점을 제공하기 위하여, 동적으로 환경에 대응할 수 있는 특성을 지녀야 한다.

다섯째, 센서 네트워크에서의 한 센서 노드의 통신 거리상의 제약으로 인하여, 센서 네트워크 운영체제를 설계할 때는 멀티 홉 라우팅 지원을 위한 프로토콜이 지원되어야 한다.

여섯째, 센서 네트워크 응용 프로그래머들의 손쉬운 프로그래밍을 위하여 용이한 운영 체제의 API를 제공하여야 한다.

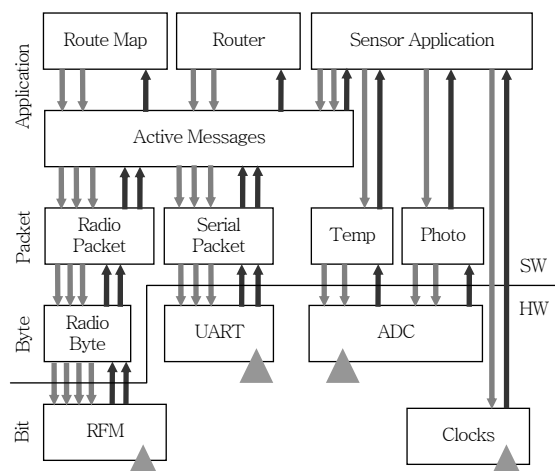
2. 센서 OS별 자료

가. TinyOS

TinyOS는 UC 버클리에서 진행해온 스마트 더스트(smart dust) 프로젝트에 이용하기 위하여 개발된 이벤트 발생 중심의 상태 변화 방식을 채택한 센서 네트워크용 운영체제로써, 동시적인 프로세싱 및 제한된 하드웨어 메모리 공간에서의 효율적인 성능을 지원해주는 운영체제이다. 특징은 다음과 같다.

TinyOS는 상태 머신의 기반의 구조를 가지는 운영체제로, 각각의 상태는 컴포넌트가 해당된다. 응용 프로그램은 (그림 5)에서처럼 컴포넌트 구현에 독립적인 연결 방법을 사용하여 각각의 TinyOS의 컴포넌트를 연결한다. 그리고, 각 컴포넌트에 명령이 내려지고 이 명령을 처리하는 이벤트 처리기는 그 명령에 따른 상태변화를 신속하게 일으켜, 필요로 하는 일을 수행하는 특징을 가진다. 다음으로는 센서 네트워크의 특징인 저전력 파워 소비를 지원하기 위하여 센서 노드들의 일이 요구되지 않을 경우 저전력 모드인 슬립 모드로 전환함으로써 효율적인 CPU의 사용을 이룰 수가 있는 것이다.

또한, nesC라는 동적 메모리를 할당하지 않는 정적인 언어를 통하여 센서 네트워크용 응용프로그램을 작성할 수가 있으며, 소프트웨어가 필요로 하는



(그림 5) 컴포넌트 기반의 TinyOS 예제

동작의 효율적인 코드를 만든다. nesC는 안정성을 위하여 전체 프로그램에 대한 분석을 통해 최적화를 수행하고, 컴포넌트 기반의 언어이며, 직접적으로 TinyOS의 이벤트 기반의 동시성 모델을 지원한다. 즉 이 언어를 이용하여 메모리 반도체 칩들과 메모리 컨트롤러를 정해진 컴포넌트와 양방향으로 갖는 인터페이스를 이용하여 프로그램 메모리 모듈을 만든다.

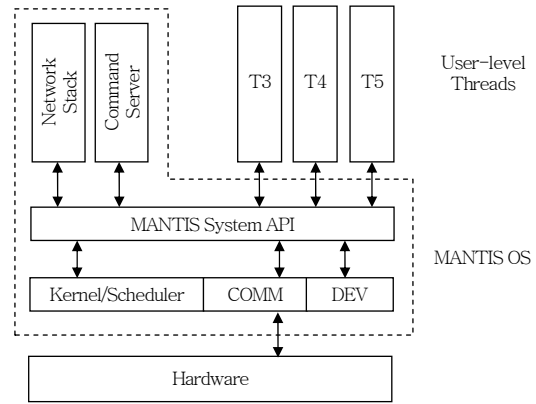
끝으로 TinyOS는 센서 네트워크의 필수적인 요소라 할 수 있는 멀티 홉 라우팅 기술을 제공함으로써 거리 제한을 해결하였으며, 여러 가지 그 외에 필요한 엔진인 TinyDB, 그리고 보안을 위하여 Tiny-Sec과 같은 모듈을 지원한다.

나. SOS

SOS는 Mote 계열 기반의 센서 네트워크를 지원하기 위한 UCLA에서 개발된 운영체제이다. SOS는 메시지 패싱, 동적 메모리 할당, 모듈을 자율적인 적재와 제거를 지원하는 공동 커널(common kernel)을 지원하며, 동적 재구성이 하나의 목표이다. 첫번째 특징은 동적 재구성 기능을 지원한다는 것으로, 이 기술은 센서 네트워크에서의 각각 개개의 센서 노드들의 소프트웨어를 무선 네트워크를 통하여 수정하고 업데이트 하는 기능이다. 이것은 센서 네트워크의 새로운 모듈 업데이트가 필요할 때, 사용될 수 있다. 두번째 특징은 센서 네트워크 프로그래머들이 필요로 하는 서비스에 맞게 응용프로그램을 제작할 수 있도록 하는 데 목적이 있다. 이를 위하여 SOS는 전통적인 운영체제의 기술을 센서 네트워크에서의 자원 제약 특성을 만족하기 위하여 설계되었으며, 그 결과로 유연한 센서 네트워크 시스템 구축을 지원하게 해준다. 세번째 특징은 응용 애플리케이션은 하나 이상의 모듈로 구성되어 있으며 이러한 모듈들은 비동기 메시지 및 함수 호출을 통하여 서로 동작하게 된다.

다. MANTIS

MANTIS는 콜로라도 대학에서 개발된 멀티 스



(그림 6) MANTIS OS 아키텍처

레드를 지원하는 센서 네트워크용 임베디드 운영체제이다. 특징은 초소형 스레드에 기반한 멀티 스레드 구조를 채택하여 TinyOS와는 달리 일반 프로그래머들이 익숙한 구조에 의하여 프로그래밍이 가능하도록 하였다. 특징은 다음과 같다.

MANTIS는 레이어 기반의 운영체제 기반이며, 이것은 멀티 스레딩, preemptive 스케줄링 기법, 그리고 mutual exclusion을 통한 I/O 동기화, 그리고 하드웨어를 추상화시키는 디바이스 드라이버의 특징을 가진다(그림 6) 참조). 또한 애플리케이션 프로그래머가 프로그래밍을 손쉽게 할수록 C언어 방식의 API를 지원하며, 센서 네트워크에서의 멀티 홉 통신을 위한 네트워크 스택을 지원한다. 그리고 리모트 셸 기능을 통하여 운영체제를 컨트롤하며, 프로그래밍의 재설정 또한 가능하다.

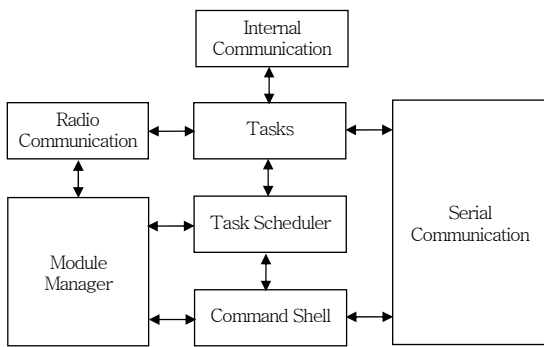
라. T-Engine Micro Kernel

T-Engine은 일본에서의 유비쿼터스 컴퓨팅 환경 구축을 위한 TRON 프로젝트에서 개발된 운영체제 기술로써, 실시간 운영체제의 특성을 가진다. 자세한 특징으로는 저전력 운영체제의 지원 및 센서 네트워크에서의 수많은 노드의 네트워크 구성을 위한 스케일러블 특성이다. 또한 태스크 매니지먼트, 태스크 동기화 및 통신, 메모리 매니지먼트 및 인터럽트 처리, 그리고 타임 매니지먼트와 관련된 OS 함수들을 특징으로 하고 있다.

마. PEEROS

PEEROS는 유비쿼터스 환경 지원을 위하여 유럽의 Twente Enshcede 대학에서 개발된 하드웨어 독립방식 기법을 채택한 센서 네트워크용 운영체제이다. 특징으로는 멀티 태스킹 및 선점형 기반의 스케줄링 기법을 제공하며, 또한 센서 노드들의 저전력 파워 소비를 위한 저전력 모드를 지원한다. 또한 C 기반의 프로그래밍 기법을 제공한다.

(그림 7)은 PEEROS의 전체적인 구조를 나타내고 있다. 이를 설명하자면, 태스크 스케줄러 부분은 EDFI 알고리즘 기반의 선점형 스케줄링 기법을 사용하며, 운영체제의 내부 컴포넌트 사이들에서의 통신을 위한 메시징 시스템, 내부 메모리의 효율적인 사용을 위한 모듈 매니저, 그리고 간단한 명령의 구현을 위하여 커맨드 셸(command shell) 기능을 탑재하였다.



(그림 7) PEEROS 아키텍처 구조도

바. ANTS

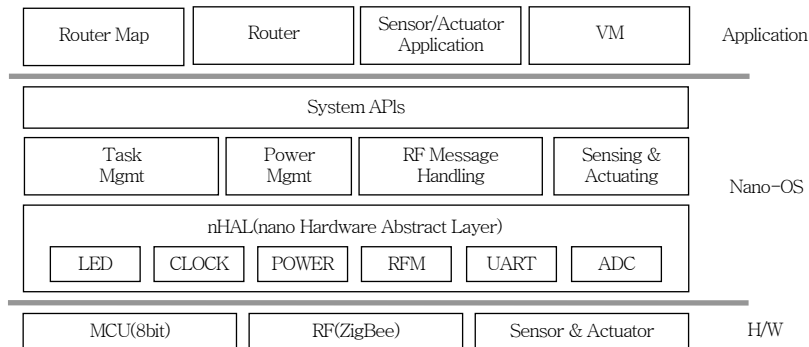
ANTS는 ICU(한국정보통신대학교)에서 센서 네트워크 환경의 쉬운 구성을 위해 개발된 운영체제이다. 특징으로는 유연한 멀티 스레딩 엔진을 탑재하였고 충분한 API를 통한 손쉬운 프로그래밍이 가능하다.

사. 나노 Qplus

나노 Qplus 운영체제는 한국전자통신연구원(ETRI)

에서 개발된 센서 네트워크용 운영체제로써 다음의 특징을 가진다. 첫째, 에너지 소모를 최소화하기 위하여, 센서 네트워크를 구성하는 노드들 간의 시간 동기화 기법을 제공하며, 또한, 슬립 모드와 활성 모드를 반복함으로써 저전력 파워 소비를 지원한다. 둘째, 제한된 메모리의 사용을 최소화하기 위하여 멀티 스레드 간의 스택을 공유한다. 셋째, 멀티 스레드 스케줄러 방식(FIFO, 시간 기반 라운드 로빈 방식, 우선 순위에 따른 스레드 선점형 방식)으로, 실시간 운영체제의 특성을 지원한다. 넷째, 응용프로그램들이 손쉽게 센서 프로그램을 개발할 수 있게 C 기반의 프로그램 작성 기법을 제공한다. 나노 Qplus의 구조를 보면 다음과 같다(그림 8) 참조.

- Nano-HAL: Nano-HAL 부분은 하드웨어를 추상화하기 위하여 개발되어진 나노 Qplus의 디바이스 드라이버 영역이다. Nano-HAL은 LED, Clock, Power, RF 모듈, UART, and ADC의 하드웨어를 제공한다.
- 태스크 스케줄러(task scheduler): 나노 Qplus의 태스크 스케줄러는 전통적인 Linux 스타일의 스케줄러와 유사하지만, 저전력 및 효율적인 리소스 관리를 제공한다는 데에서 차이를 두고 있다. 태스크 스케줄러의 특징으로는 POSIX 스레드 기반의 API를 통하여 멀티 스레드 간의 메시지 교환이 가능하도록 하고 있다.
- 동적 전원관리(dynamic power management): 시스템이 동작하는 과정에서 처리할 태스크가 없는 경우에 전원 소비가 낮은 실행모드로 전환하여 배터리 에너지를 절약하도록 하여 노드뿐만 아니라 네트워크 전체의 생존 시간을 늘리는 역할을 수행한다.
- RF 메시지 핸들링(RF message handling): 이 모듈은 에너지 효율적인 무선 통신을 센서 노드들 사이에서 지원하며, IEEE 802.15.4의 기반의 멀티 홉 라우팅 형성을 지원하여 준다.
- 시간 동기화(time synchronization): 시간 동기화는 정확한 타임 스탬핑과 정밀하게 조절된 센서 노드들 사이의 듀티 사이클을 조정하기 위하



(그림 8) 나노 Qplus 구조

여 중요한 모듈이다. 나노 Qplus에서는 시간 동기화를 위하여 트리 기반의 시간 동기화 기법을 제공한다.

또한 나노 Qplus는 멀티홉 라우팅을 제공하기 위하여 클러스트 기반의 라우팅 기법인 스타-메시 라우팅 기법을 제공하며, RF 무선 통신을 위하여 ZigBee 프로토콜을 지원하는 것이 특징이다.

V. 결론

무선 센서 네트워크 기술이 물류/유통, 가정, 병원, 그리고 방재 및 환경 사회전반에 걸쳐 다양하게 이용되어짐으로써 인간 사회에 적용될 수 있는 광범위한 위험은 우리 모두가 인식하고 있다. 그리고 이러한 사회 현상을 뒷받침하기 위하여 센서 네트워크 운영체제를 통한 응용프로그램의 손쉬운 개발, 그리고 저전력 센서 네트워크 지원 및 유연한 센서 노드들의 관리가 필수적인 기술로 떠오르게 되었음을 알 수 있다. 그리하여, 이 동향지에서는 센서 네트워크의 시장 동향 및 기술, 그리고 특히 센서 네트워크 운영체제의 기술 동향 및 여러 운영체제 종류를 전반적으로 알아 보았다. 이를 통하여 우리는 센서 네트워크에서 필요로 하는 운영체제 기술이 무엇인지를 알았고, 실제적으로 이를 통하여, 센서 네트워크 운영체제의 현명한 설계 및 적용을 할 수 있을 것이라 본다.

약어 정리

ANTS	An evolvable Network of Tiny Sensors
CPU	Central Processing Unit
EDFI	Earliest Deadline First with Inheritance
HAL	Hardware Abstract Layer
HR-WPAN	High Rate Wireless Personal Area Network
KETI	Korea Electronics Technology Institute
MAC	Medium Access Control
MANTIS	Multimodal Networks of In-situ Sensors
MCU	Micro Controller Unit
nesC	Network Embedded System C
PEEROS	Preemptive EYES Real-time OS
RF	Radio Frequency
RFD	Reduced Function Device
RISC	Reduced Instruction Set Computer
RTC	Real Time Clock
RTOS	Real-Time Operating System
TIP	Tiny Interface for Physical World
USN	Ubiquitous Sensor Network
WPAN	Wireless Personal Area Network
WSN	Wireless Sensor Network

참고 문헌

- [1] <http://www.intel.com/research/exploratory>
- [2] On World, Wireless Sensor Networks: Mass Market Opportunities, 2004.
- [3] On World, Wireless Sensor Networks: Growing Markets, Accelerating Demand, July 2005.

- [4] <http://www.atmel.com>
- [5] <http://www.arm.com>
- [6] 김원수, 장기수, ZigBee 기술 동향 및 시장 전망 분석, 전자부품연구원 전자정보센터, 2003.
- [7] www.radiopulse.co.kr
- [8] ason Hill, Robert Szewczyk, Alec Woo, Seth Hollar, David Culler, and Kristofer Pister, "System Architecture Directions for Network Sensors," ASPLOS 2000, Cambridge, Nov. 2000.
- [9] Ralph M. Kling, "Intel Mote: An Enhanced Sensor Network Node," *Intel Workshop on Advanced Sensors, Structural Health Monitoring, and Smart Structures*, Nov. 2003.
- [10] <http://www.intel.com/research/exploratory/motes.html>
- [11] <http://www.moteiv.com/products/docs/an001-changes.pdf>
- [12] <http://www.moteiv.com/products/docs/an002-telos.pdf>
- [13] <http://tinyos.keti.re.kr/>