

확장성과 안정성을 고려한 센서 운영체제의 구조 설계

김성현*, 이승룡**

*경희대학교 컴퓨터공학과

**경희대학교 컴퓨터공학과

e-mail: sacoku@oslab.khu.ac.kr

Sensor Operating System Architecture Designing for Reliability and Expansion

Sung-Hyun Kim*, Sung-Young Lee**

*Dept of Computer Engineering, Kyung-Hee University

**Dept of Computer Engineering, Kyung-Hee University

요 약

유비쿼터스 센서 네트워크 환경에서 센서 운영체제 기술의 중요성은 날이 갈수록 높아져 가고 있으며 센서 하드웨어의 하드웨어 자원의 부족, 극소형의 크기, 저전력으로 운용되어야 하는 특성을 보충하기 위한 노력은 이미 많은 연구가 진행되었으며 진행되고 있다. 하지만 물리적인 세계와 디지털 세계를 연결하는 역할을 하는 센서에서 데이터의 신뢰성은 반드시 보장이 되어야 하는 요소임에도 많은 부분이 간과되어 왔다. 데이터의 신뢰성을 보장하기 위해서 센서는 안정성을 확립할 수 있어야 하며 다양한 환경에 동적으로 적응이 가능해야만 한다. 그리하여 본 논문에서는 센서에 안정성을 확립할 수 있는 방법과 동적인 환경에 적응이 쉬운 형태의 확장이 쉬운 구조를 가질 수 있는 방법을 제시함으로써 센서의 신뢰성에 기여할 수 있도록 한다.

1. 서론

유비쿼터스 센서 네트워크란 기존 인간과 컴퓨터간의 커뮤니케이션에 일상 생활에 산재된 사물과 물리적 대상을 추가시켜 협력 네트워크를 구성하는 것을 말하며, 필요로 하는 곳에 수 많은 센서 노드들을 부착하여 자율적으로 정보를 수집, 관리 및 제어하는 시스템이라고 할 수 있다.[1] 이러한 인프라를 이용해서 다양하고 편리한 새로운 서비스를 제공하는 것이 유비쿼터스 사회의 궁극적인 목표라고 할 수 있으며 센서 네트워크는 물리적인 세계와 디지털 세계를 연결 할 수 있는 특징 때문에 많은 분야에 응용될 수 있다. 센서 네트워크의 핵심 기술 요소로서는 하드웨어 플랫폼 기술과 네트워크 기술 그리고 초소형 운영체제 기술을 들 수 있는데 그 중에서 센서 운영체제 기술은 하드웨어를 쉽고 효율적으로 제어할 수 있기 때문에 그 중요성은 점점 높아져 가고 있다고 할 수 있다. 이미 알려져 있듯이 센서는 작은 크기로 구성이 되어야 하며 저전력으로 운용되어야 하기 때문에 하드웨어적인 차원에서 제한적일 수 밖에 없다. 그렇기 때문에 제한적인 하드웨어 자원을 극복하기 위해서는 소프트웨어 기술인 센서 운영체제 기술의 중요성은 위에서 언급한 유비쿼터스 센서 네트워크 기술에서 중요하다고 할 수 있으며 그 중요성은 앞으로 더더욱 커질 것으로 전망하고 있다. 센서 운영체제는 센서 하드웨어 적은 메모리 때문에 극소형의 크기를 가져야 하며 센서 노드간에 무선통신을 위한 무선통

신 모듈을 포함되어야 한다. 아직까지 센서 운영체제 기술은 표준화가 이루어 지지 않았으며 상업적인 목적으로 대량 생산되어 사용되는 제품도 아직까지는 찾아보기 힘들다 하지만 국내외에서 많은 연구가 활발히 진행되고 있으며 앞으로 표준화 경쟁에 치열할 것으로 예상할 수 있다. 기존의 많은 센서 운영체제의 연구는 부족한 하드웨어 자원을 효율적으로 운용할 수 있는 연구와 저전력으로 장시간 운용이 가능하도록 하는 측면의 연구가 주를 이루었다. 즉 유연성, 병행성, 저전력에 초점이 맞추어진 연구가 대부분이었다. 하지만 센서는 유비쿼터스 센서 네트워크에서 최종단에서 상황에 따른 센싱 데이터를 추출하는 역할을 하고 있기 때문에 종단에서 센서의 오동작으로 인한 데이터의 오류는 상위 시스템에 많은 영향을 줄 수 있으며 극단적으로는 인명 피해로까지 이어질 수 있는 가능성을 가지고 있다. 그렇기 때문에 센서는 데이터에 대한 어떠한 상황에서도 데이터의 신뢰성을 보장 할 수 있어야 한다. 현재 많은 연구가 초기적인 단계이기 때문에 안정성 보다는 효율성에서 연구가 진행이 되고 있다. 하지만 앞으로는 안정성을 높이기 위한 연구 또한 활발히 진행될 것이다. 또한 유비쿼터스 환경은 다양하며 복잡한 요소를 곳곳에 내재하고 있으며 언제든지 예외상황이 발생할 수 있기 때문에 복합적인 환경에 센서가 대응하기 위해서는 센서의 실시간적인 확장성은 필수적이라고 할 수 있다.

본 논문은 센서 운영체제의 신뢰성을 위해서 센서에게

안정성을 부여할 수 있는 기술과 실시간 환경에서 센서의 확장성을 부여할 수 있는 기술에 대해서 기술한다. 센서의 제한적인 메모리와 프로세서의 제한적인 성능의 범위 안에서 확장이 가능한 형태로 구성함으로써 센서의 안정성과 확장성에 기여할 것으로 기대한다.

2. 관련연구

가. TinyOS[2]

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 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 응용 프로그램을 작성할 수가 있으며, 소프트웨어가 필요로 하는 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 필

요한 엔진인 TinyDB, 그리고 보안을 위하여 Tiny-Sec과 같은 모듈을 지원하고 있지만 커널과 어플리케이션에서 메모리를 공유 하고 있기 때문에 어플리케이션의 오동작이 커널마저 중지시켜 버리기 때문에 센서노드의 안정성을 확립할 수 없다.

나. SOS[3]

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 재구성

이 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. 두 번째 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. 세 번째 특징은 응용 애플리케이션은 하나 이상의 모듈로 구성되어 있으며 이러한 모듈들은 비동기 메시지 및 함수 호출을 통하여 서로 동작하게 된다. 하지만 어플리케이션과 커널이 메모리를 공유하고 있지 않기 때문에 어플리케이션의 오동작이 커널에 영향을 끼치지 않는 형태로 구성이 되어 있으나 각각의 태스크들이 함수의 형태로 존재하기 때문에 복잡한 환경에 따른 어플리케이션 제작시 각 태스크간의 연동에 있어서는 복잡하기 때문에 확장적인 측면에서는 미흡하다고 할 수 있다.

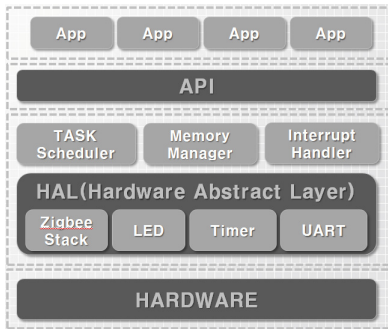
다. 나노 Qplus[4]

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, 응용프로그래머들이 손쉽게 센서 프로그램을 개발할 수 있게 C 기반의 프로그램 작성 기법을 제공하지만 위의 운영체제 또한 커널과 메모리를 공유하고 있기 때문에 어플리케이션의 오동작이 커널에 영향을 주고 있기 때문에 안정성적인 측면에서 미흡하다고 할 수 있다.

3. 연구 내용

본 논문에서 제안하는 안정성과 확장성을 고려한 센서 운영체제의 구조는 하드웨어 추상화 계층인 HAL, Zigbee Stack의 경량화를 통한 Network Stack, 커널의 요소로서는 태스크를 관리하기 위한 Task Manager와 하드웨어 인터럽트를 효율적으로 관리하기 위한 Interrupt Manager 그리고 메모리를 관리하기 위한 Memory Manager로 구성되어 있으며 안정성과 확장성은 Memory Manager와 Interrupt Manager를 통해서 구현된다. 또한 상위에는 어

플리케이션과의 인터페이스를 위한 System API로 구성되어 있다. 다음의 <그림 1>은 무선 센서 네트워크 환경에서 확장성과 안정성을 지원하기 위한 제안하는 센서 운영체제의 아키텍처이다.



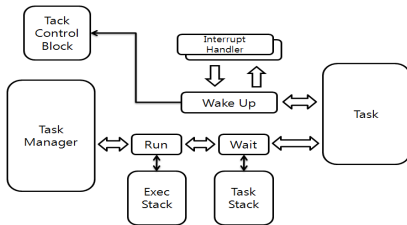
(그림 1) 센서 운영체제 아키텍처

3-1. HAL

HAL에서는 각각의 하드웨어 장치에 대한 저수준의 인터페이스를 제공하며 POSIX 802.1003.1a 기반으로 설계되었다. 또한 상위레벨과의 호환성을 무너뜨리지 않는 범위안에서 다양한 하드웨어에 이식이 용이한 구조로 설계되었다. 일반적으로 HAL이 가질 수 있는 기능은 각 디바이스의 초기화, 제어, 지정된 인터페이스에 따른 I/O함수를 제공하고 있다.

3-2. Kernel

3-2-1. Task Manager



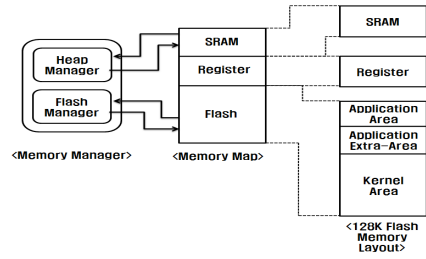
(그림 2) 태스크 관리자의 구조

태스크 관리자는 태스크 스위칭으로 인한 전력소모를 감소시키기 위해서 태스크가 지정된 시간까지 아무런 이벤트를 받을 수 없다면 주기적으로 슬립모드로 들어가며 이는 잦은 태스크 스위칭으로 인한 전력 소모를 감소시킬수 있기 때문에 저전력인 측면을 만족시킬 수 있을 것이라 예상된다.

3-2-1. Memory Manager

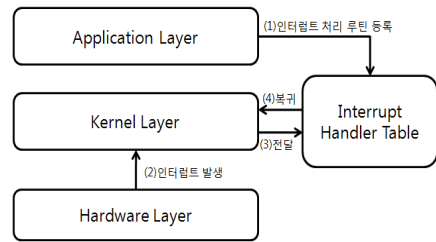
커널의 메모리 영역과 어플리케이션 메모리 영역을 분리해서 운영하기 위해서는 각각의 메모리(FLASH, SRAM)에 Write Operation 또는 Read Operation에 대한 접근 제어가 필요하다. 이를 위해서 본 센서 운영체제에서는 SRAM과 Flash메모리에 대한 접근을 관리자를 통해서 접근하도록 함으로서 메모리에 대한 접근제어가 가능하며 이로써 커널과 어플리케이션 메모리 영역의 분리 운영이 가능하다. 커널과 어플리케이션 메모리를 독립적으로 운영

함으로서 어플리케이션의 동작은 커널에 영향을 줄 수 없으며 안정성에 기여할 수 있을 것이라 예상된다.



(그림 3) 메모리 관리자의 구조

3-2-2. Interrupt Manager



<그림 4> 인터럽트 처리

기존의 대부분의 센서 운영체제는 하드웨어적으로 발생하는 모든 인터럽트에 대해서 커널수준에서 처리하고 있었다. 하지만 커널에서 모든 인터럽트를 처리하는 구조는 각 인터럽트에 대한 지정된 처리만을 하고 있기 때문에 다양한 환경에 대응하기 어려우며 예외 상황에 대한 처리 또한 힘들다고 할 수 있다. 이를 개선하기 위해서 본 운영체제에서는 인터럽트에 대한 처리를 어플리케이션 레벨에서 접근이 가능하게 함으로서 다양한 환경에 따른 어플리케이션이 제작이 가능할 것이다. 즉, 확장성을 제공할 수 있기 때문에 다양한 센서 네트워크 환경에 적응성을 높일 것으로 예상된다.

4. 결론

본 논문은 최종적 최종적으로 유비쿼터스 센서 네트워크 환경에서 센서의 안정성을 확립함으로서 센서의 작동에 신뢰성을 확립할 수 있으며 확장성을 부여함으로서 다양한 환경에 동적으로 적응할 수 있을 것이라 예상된다. 안정성을 지원하기 위해서는 어플리케이션의 오동작이 운영체제까지 연결이 되지 않아야 하며 이는 어플리케이션과 운영체제 영역이 독립적으로 운영이 되어 함을 의미한다. 영역의 분리는 가상 메모리 기술이 기본적으로 지원이 되어야 하며 운영체제 영역에 제한적으로 접근 할 수 있게 하는 특권 모드의 지원이 필요하다. 또한 스마트 센서를 위한 확장성을 제공하기 위해서 실시간으로 발생하는 인터럽트에 대해서 어플리케이션 레벨에서 많은 부분을 동적으로 처리할 수 있게 함으로서 지원이 가능하리라 예상

한다. 이러한 기술은 제한적인 하드웨어로 말미암아 기존의 일반적인 임베디드 운영체제에서 쓰이는 방법으로는 해결 될 수 없으며 본 연구에서 제안하는 방법으로서 해결 될 수 있으리라 본다.

5. 참고문헌

- [1] 박승민, “센서 네트워크 노드 플랫폼 및 운영체제 기술 동향”
- [2] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler, TinyOS: An Operating System for Sensor Networks
- [3] Chih-Chi Han, Ram Kumar Rengaswamy, Roy Shea, Eddie Kohler, and Mani Srivastava. “Sos: A dynamic operating system for sensor networks” ACM Press, 2005.
- [4] Seungmin Park, Jin Won Kim, Kee-Young Shin, and Daeyoung Kim “A Nano Operating System for Wireless Sensor Networks”
- [5] H. Abrach, S. Bhatti, J. Carlson, H. Dai, J. Rose, A. Sheth, B. Shucker, J. Deng, and R. Han. MANTIS: system support for Multimodal Networks of In-Situ sensors. In Proc. WSNA’03, 2003.
- [6] P. Levis and D. Culler. Mat’e: A tiny virtual machine for sensor networks. In Proc. ASPLOS-X, October 2002.
- [7] Chih-Chieh Han, Ram Kumar, Roy Shea, Eddie Kohler and Mani Srivastava, A Dynamic Operating System for Sensor Nodes

6. Acknowledgement

This research was supported by the MIC (Ministry of Information and Communication), Korea. It was also under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Advancement) (IITA-2008-C1090-0801-0002).