지능형 무선센서 프로그래밍에 대한 고찰

김진환* · 김광백** · 조재현***

*영산대학교 **신라대학교 ***부산가톨릭대학교

A Study on An Intelligent Wireless Sensor Programming

Jin-whan Kim* Kwang-baek Kim** · Jae-hyun Cho***

*Youngsan University **Silla University ***Catholic University of Pusan

E-mail : kjw@ysu.ac.kr, gbkim@silla.ac.kr, jhcho@cup.ac.kr

요 약

본 논문에서는 USN(Ubiquitous Sensor Network) 환경에서 많이 활용되는 운영체제로 국내외 많은 연구기관, 대학, 기업체에서 활용하고 있는 TinyOS와 지능형 무선센서에 대해서 살펴보고, 이를 제어하기 위해 활용되는 nesC 프로그래밍 기법 및 특성을 살펴보았다.

ABSTRACT

This paper is a research on an operating system(TinyOS) and nesC programming methods. TinyOS is being used for development of USN in many domestic/foreign research institutes, universities and companies. We are describing about the TinyOS and nesC programming methods and characteristics.

키워드

USN, 지능형 무선센서, TinyOS, NesC programming

Ⅰ. 서 론

사용자가 장소나 시간에 구애받지 않고 자유롭게 네트워크에 접속하여 원하는 정보를 송수신할 수 있는 통신환경을 구현하기 위한 USN(ubiquitous sensor network)은 다양한 무선센서와 컴퓨터, 통신망이 연결되어 실시간으로 각종 정보를 활용할 수 있는 형태이다. 즉, 필요한 모든 사물과 사람, 환경에 무선센서를 부착하여 네트워크나 그 주변에서 발생되는 다양한 정보를 탐지하고, 유무선 네트워크를 통하여 실시간으로 관리하고 활용하는 것이다. 이 기술은 가정, 사무실, 물류, 유통, 환경감시, 지능형 홈, 군사, 우주/항공, 선박, 자동차, 의료 등 모든 산업 분야에서 활용되고 있으며, 더욱 가속화될 전망이다.

일반적으로 무선센서는 그림 1과 같이 무선 통신칩, 메모리, 소형 운영체제(OS), 응용프로그램을 포함하고 있으며 모트(mote) 혹은 센서 노드(sensor node)라고도 부른다. 최근에는 인체에 삽입하여 질병을 진단할 수 있는 초소형 나노-바이오 센서가 개발 중에 있으며, 눈에 보이지 않는 먼지 크기의 센서도 연구되고 있다.

그림 1. 다양한 지능형 무선센서
Fig. 1 Various intelligent wireless sensors

현재 많이 알려져 있는 센서 네트워크용 운영체제로는 TinyOS, SOS, MANTIS, Contiki, T-kernel, 국내에서 개발된 Nano-Qplus 등이 있다[1]. 본 논문에서는 국내외 많은 연구기관, 대학, 기업체에서 활용하고 있는 TinyOS에 대해서 살펴보고, 여기서 동작되는 nesC 프로그래밍 기법 및 특성을 간략하게 살펴 보았다.

Ⅱ. TinyOS의 구조와 특징

TinyOS는 USN 환경 구현을 위해서 미국 UC 버클리대학에서 개발된 오픈 소스 개발 환경을 제공하는 소형 운영체제로서 모듈화가 잘 되어 있고, 제한된 자원에서 동작할 수 있는 컴포넌트 기반의 이벤트 구동(event driven) 방식으로 구현되었다.

특징은 적은 메모리를 사용하고, 적은 전력 소모, 모듈화, 동시 동작이 가능하도록 설계되었다는 것이다[2].

TinyOS에서 센서 노드를 동작 시키는 것은 이벤트(event)와 태스크(task)로 볼 수 있다. 하드웨어 이벤트는 인터럽트를 의미하며, 이런 인터럽트는 타이머, 센서, 통신 장치로부터 발생된다. 태스크들은 일반적인 프로그래밍에서 프로시저 호출을 의미 하지만 태스크 큐에 등록하여 FIFO(First In First Out) 방식의 순차적 실행이 가능하도록 구성되어 있다. 즉, 하나의 태스크 동작이 완료되기 전에는 다른 태스크가 실행 될 수 없는 비선점(Non-preemptive) 방식으로 태스크가 동작된다. 태스크 큐에 실행되어야 할 태스크가 존재하지 않을 경우, 시스템은 새로운 하드웨어 인터럽트 혹은 태스크가 발생될 때까지, 슬립(slip) 모드로 전환되어 전력 소모를 최소화한다.

TinyOS 구성은 컴포넌트 단위로 모듈화 되어있다. 컴포넌트(component)는 다른 컴포넌트들과 인터페이스(interface)를 통해 연결되어 커맨드(command)함수와 이벤트(event)함수로 실행된다.

애플리케이션 개발자들은 컴포넌트들을 라이브러리로 사용하며, 인터페이스들을 사용하여 컴포넌트들을 서로 연결하여 애플리케이션을 만들게 된다.

III. nesC 프로그래밍

TinyOS는 응용프로그램 개발을 위해 nesC라는 프로그래밍 언어를 사용한다. nesC는 C언어의 확장으로 센서네트워크와 같은 임베디드 시스템을 위해 개발되었으며, 개발 환경은 Linux 운영체제나 Cygwin을 이용하는 windows 2000/XP 등의 윈도우즈 운영체제를 지원하고 있다. nesC로 개발된 소프트웨어는 변환과정을 거쳐 C언어로 변환되고, GCC컴파일러로 컴파일되어 실행코드가 만들어지고 센서노드에 다운로드되어 실행된다.

nesC는 컴포넌트 기반의 구조를 가지고 설계되었다. 모든 프로그램은 컴포넌트로 구성되고, 컴포넌트는 모듈(Module)과 다른 컴포넌트들을 연결하는 컨피규레이션(Configuration)으로 나눌 수 있다[3, 4].

모듈은 C언어와 유사한 코드를 이용하여 컴포넌트를 구현하고 컨피규레이션은 사용하게 될 컴포넌트들과 인터페이스의 연결 관계를 명시한다. C 언어 프로그램을 작성할 때 반드시 main() 함수를 포함하듯이 nesC 프로그램을 작성할 때는 반드시 main 컴포넌트를 포함해야 한다. 그 이유는 main 컴포넌트가 시스템 초기화 루틴을 호출하고 스케줄러를 구동하기 때문이다.

3.1 컨피규레이션 (Configuration)

컨피규레이션(Configuration)에서는 사용할 컴포넌트(Component)들을 명시하고, 인터페이스(Interface)의 상호 연결 관계(->, <-, =)를 나타내며, 사용(uses)과 제공(provides)하는 관계를 표시하여 서로 다른 컴포넌트 간의 논리적인 연결 관계(wiring)를 표현한다.

nesC에는 컴포넌트 간의 연결(wire) 관계를 위한 3가지 연결 기호(->, <-, =)가 있다.

- interface aa -> interface bb: bb에서 구현되어 제공하는 함수를 aa가 사용함을 의미한다.
  bb는 제공자, aa는 사용자 인터페이스
- interface aa <- interface bb: aa에서 구현되어 제공하는 함수를 bb가 사용함을 의미한다.
  aa는 제공자, bb는 사용자 인터페이스
- interface aa = interface bb: 두 interface aa, bb가 동일함 의미한다. aa, bb는 사용자 혹은 제공자 인터페이스

위의 방법에 따라 연결되면 해당하는 컴포넌트들의 command, event, interface를 사용할 수 있다.

3.2 모듈 (Module)

모듈(module)은 미리 정의된 인터페이스(interface)를 제공하거나 사용하게 되며, 구현(implementation)에서는 실제 수행할 코드를 작성하게 된다. 모듈은 aaaM.nc 형태의 파일명을 가지며, implementation{} 내에 모든 command(event)에 해당하는 코드를 작성한다. 각 모듈은 명령(commands)을 호출(call)하고 event를 수행(signal)한다.

3.3 인터페이스 (Interface)

nesC에서 Interface는 양방향성을 가지며 제공자(provider)와 사용자(user) 컴포넌트를 연결하는 연결자 역할을 한다. 제공자 모듈에서는 모든 commands에 대한 코드를 구현하며, 사용자 모듈에서는 모든 events에 대한 코드를 구현한다.

IV. 결론

컴퓨터와 인터넷이 널리 보급됨에 따라서 업무의 효율과 생산성을 높이고, 멀리 떨어져 있는 사람들과 실시간으로 손쉽게 정보를 주고받는 등 사회에 주는 긍정적인 영향은 이루 말로 표현할 수 없을 것이다. 최근에는 모든 사물, 사람, 로봇, 환경 등에 무선센서를 부착하고, 컴퓨터, 유무선 인터넷, 다양한 통신망이 통합 연동되어 다양한 분야(가정, 사무실, 물류, 유통, 환경감시, 지능형 홈, 군사, 우주/항공, 선박, 자동차,

의료 등 모든 산업 분야)에서 연구 및 활용되고 있다. 미래의 핵심 요소 기술로 부각됨에 따라 많은 국가에서 경쟁적으로 기초기술과 응용기술을 연구하고 개발하는 사업에 참여하고 있다.

USN 관련 기술의 발전은 더욱 가속화될 것이며, 이에 대비한 기술과 비즈니스 모델을 개발하고 준비하는 것이 무엇보다 중요할 것이다.

참고 문헌

[1] 송준근, 마평수, 박승민, "센서 네트워크용 초소형 OS", 통신소프트웨어, 2007.07, pp. 26-35
[2] http://docs.tinyos.net/index.php/TinyOS_Tutorials
[3] http://www.tinyos.net/tinyos-1.x/doc/nesc/ref.pdf
[4] http://www.tinyos.net/tinyos-2.x/doc/pdf/tinyos-programming.pdf