

Plug&Play 지원 센서 인터페이스 플랫폼에 관한 연구

문영백*, 김내수*, 은성배**
 *한국전자통신연구원
 **한남대학교 정보통신공학과
 e-mail : moonyb@etri.re.kr

A Research on Plug&Play support Sensor Interface Platform

Young-Bag Moon*, Nae-Soo Kim*, Seong-Bae Eun**

*RFID/USN Research Division, ETRI

**Dept. of Information Communication Engineering, Han-Nam University

요 약

Plug&Play 지원 센서 인터페이스 플랫폼을 적용하면 센서모듈의 연결이 자동으로 인식되고 동작하게 된다. Plug&Play 지원 센서 인터페이스 플랫폼은 센서노드, 센서모듈, 센서 디바이스 드라이버 매니저로 구성되고 센서모듈의 디바이스 드라이버 제공 방법에 따라 로컬 인식형과 원격 인식형으로 구분된다. 센서노드와 센서모듈간의 센서 인터페이스는 다양한 인터페이스 방식을 갖는 센서모듈을 범용적으로 수용할 수 있도록 정의된다. 본 논문은 Plug&Play 지원 센서 인터페이스 플랫폼구조 및 인식모드에 대해서 기술하고, 센서노드, 센서모듈의 구조와 각 기능에 대해서 기술한다.

1. 서론

USN 을 구성하기 위해서는 관리서버, 게이트웨이, 센서 노드 등 다양한 기능의 하드웨어로 구성된다. 또한 각 하드웨어는 통신 및 제어 모듈, 센서 모듈, 전원모듈 등 기능에 따른 다양한 형태로 구성된다. 현재 사용되는 센서 노드들은 기능 및 외형에 따라 매우 많은 형태를 갖고, 동일한 기능을 사용하는 경우에도 제작사마다 서로 다른 형태의 센서 노드 개발로 서로 호환성이 결여되며, 국가 사회적으로 개발 및 운용에 있어 많은 경제적 시간적 손실이 발생하고 있다.

Plug&Play 지원 센서 인터페이스 플랫폼은 센서모듈이 센서노드와 연결되면 자동으로 인식되고 동작하도록 하는 데 목적이 있다. 센서노드의 소프트웨어 개발자는 센서노드의 하드웨어 인터페이스와 Tiny-OS[1], Nano-Q+[2]와 같은 센서노드 OS 환경에 맞게 연결되는 센서모듈의 하드웨어 인터페이스와 디바이스 드라이버를 설계해야 한다.

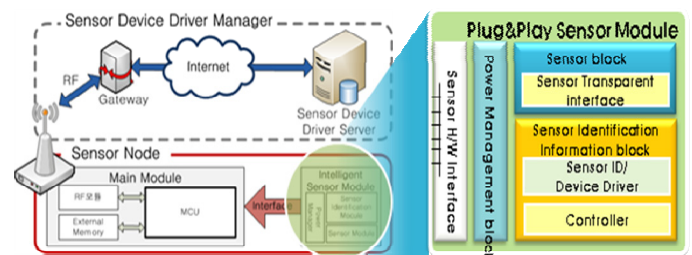
센서 인터페이스[3]를 적용하여 서로 다른 하드웨어 인터페이스를 가지는 센서모듈이 연결되더라도 범용적으로 수용할 수 있고, 디바이스 드라이버 부분도 센서모듈에서 제공하게 되어 센서노드 개발자의 부담을 줄여주게 된다. IEEE 1451[4]에서 센서인터페이스 부분에 대한 표준을 정의하고 있다. IEEE 1451 은 센서나 트랜스듀서의 특성을 기술하는 TEDS(Transducer Electronic Data Sheet)에 의해 트랜스듀서의 추상화를 지원받는 내용으로 본 논문에서 제시하는 센서노드와 센서모듈간 센서 하드웨어 인터페이스를 정의하고 Plug&play 기능을 지원하는 것과는 상이하며 실제 개발자의 구현 부분에 있어서 TEDS 를 지원하기 위해 추가적으로 구현할 부분이 부담으로 고려될 수 있다.

본 논문은 Plug&Play 지원 센서 인터페이스 플랫폼을 제시한다. 이를 구성하는 센서노드, 센서모듈, 센서 디바이스 드라이버 매니저를 구체적으로 기술하여 Plug&Play 센서 인터페이스 플랫폼의 구조 및 기능을 이해한다.

2. Plug & Play 지원 센서 인터페이스 플랫폼 구조 및 인식모드

Plug & Play 지원 센서 인터페이스 플랫폼은 센서 Plug&Play 기능을 지원하는 시스템으로 센서노드, 센서모듈, 센서 디바이스 드라이버 매니저로 구성된다.

센서노드와 센서모듈간에는 어떠한 센서가 적용되더라도 전원, 센싱데이터 전송 등에 문제가 없도록 하드웨어 인터페이스가 정의되어야 한다. 센서노드가 센서모듈을 인식하여 통신하기 위해서는 센서모듈의 디바이스 드라이버가 필요하다. 디바이스 드라이버를 제공하는 방식에 따라 로컬 인식형과 원격 인식형으로 구분된다. 디바이스 드라이버 매니저는 원격인식형인 경우 적용되는 부분으로 본 논문에서는 로컬인식형의 구조를 기본으로 기술한다.



(그림 1) Plug & Play 지원 센서 인터페이스 플랫폼 구조

2.1 로컬 인식형

센서 모듈 자체에 센서 디바이스 드라이버가 탑재되어 센서노드에 센서 모듈이 연결되면 자동으로 센서노드는 센서모듈의 센서 디바이스 드라이버를 제공받아 동작시킨다.

센서모듈이 연결되면 센서노드에서는 센서의 동작을 위해 센서 식별 정보 수신, 센서 디바이스 드라이버 비교, 센서

디바이스 드라이버 요청의 절차가 진행된다. 이 기능들은 센서 노드가 초기화 되었을 때 센서 모듈로부터 센서 식별 정보를 수신하고, 자신의 OS 에 맞는 센서 디바이스 드라이버를 센서 모듈로부터 얻어와 동작시키는 방식이다.

로컬 인식형을 지원하기 위해서는 센서모듈에 센서노드 OS 환경에 맞는 센서 디바이스 드라이버가 내장되어 있어야 한다.

2.2 원격 인식형

센서 모듈에 센서 디바이스 드라이버가 내장되어 있지 않거나, 센서 노드의 OS 에 맞는 센서 디바이스 드라이버가 없는 경우 센서 디바이스 드라이버 매니저로 필요한 센서 디바이스 드라이버를 요청하여 제공받고 원격으로 센서 디바이스 관리 서버로부터 디바이스 드라이버를 다운받아 센서를 인식하게 된다.

3. 센서 노드 플랫폼 구조 및 센서 H/W 인터페이스

센서노드는 MCU, RF, 센서 H/W 인터페이스로 구성된다. 센서모듈이 연결되면 센서노드는 인터럽트 신호에 의해 센서모듈의 연결을 인식하고 센서 ID 정보를 받는다. 이후 디바이스 드라이버를 다운받아 센서모듈을 제어하게 된다.

센서 H/W 인터페이스는 총 9 Pin 으로 구성되고 표 1 과 같이 정의된다.

<표 1> 센서 인터페이스 정의

Pin	명칭	설명
1	VCC	전원 Line
2	Interface 1	Data 통신형태의 센서를 위한 Digital Interface 0 Line(ADC 부가기능)
3	Interface 2	Data 통신형태의 센서를 위한 Digital Interface 1 Line(INT 부가기능)
4	Interface 3	Data 통신형태의 센서를 위한 Digital Interface 2 Line
5	Interface 4	Data 통신형태의 센서를 위한 Digital Interface 3 Line
6	Connection	Sensor Module 의 연결 확인용 INT Line
7	SCK	I2C 통신을 이용한 센서 ID 전달용 Data Line
8	DATA	I2C 통신을 이용한 센서 ID 전달용 Data Line
9	GND	Ground Line

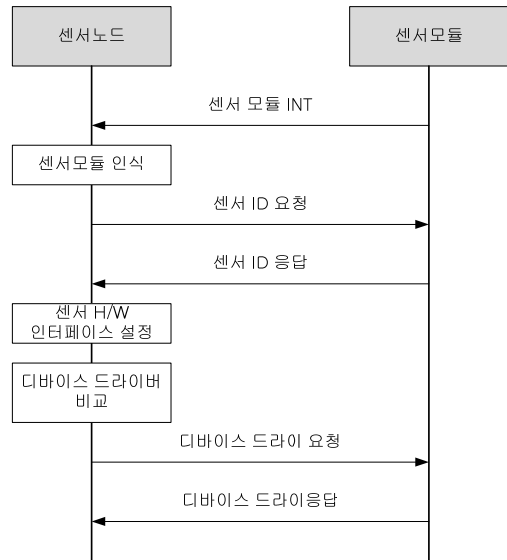
그림 2 는 센서모듈이 센서노드에 연결되면 센서노드가 센서모듈을 인식하는 과정을 보여주고 있다. 센서모듈이 연결되면 센서모듈의 INT 신호에 의해 인식하게 된다. 이후 센서 ID 정보를 수신하여 센서 모듈에 맞는 센서 H/W 인터페이스의 Digital Interface 라인을 S/W 적으로 처리하여 설정하게 된다. 센서모듈 연결방식이 설정되면 센서의 디바이스 드라이버를 다운받아 센서의 제어가 가능하게 된다.

4. 센서 모듈 플랫폼 구조

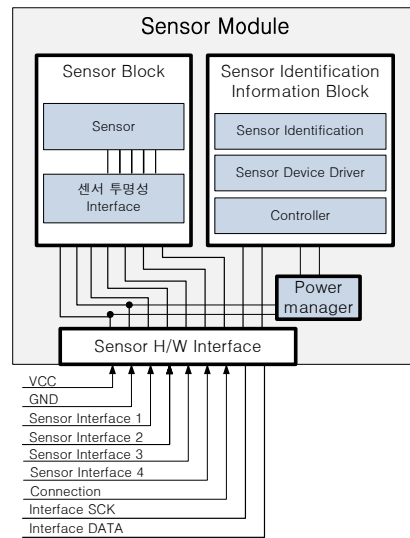
그림 3 의 센서 모듈은 센서가 탑재되어 센서노드와 연결되는 모듈을 의미하며 센서 블럭, 센서식별 정보 블럭, 전력제어, 센서 H/W 인터페이스로 구성된다.

센서 모듈과 센서노드간 H/W 인터페이스는 센서노드와 동일하다. 센서 블럭은 센서와 센서 투명성 인터페이스로 구성되고, 센서식별 정보 블럭은 센서 ID, 센서 디바이스 드

라이버, Controller 로 구성된다.



(그림 2) 센서모듈 인식 및 제어 절차



(그림 3) 센서 모듈 플랫폼 구조

4.1 센서 투명성 인터페이스

센서는 Dummy Sensor 를 그대로 사용하고, 센서의 다양성을 극복하기 위해서 인터페이스 보드에서 센서 출력 특성에 따라서 처리를 달리 한다. 표 2 는 센서 출력 특성에 따른 인터페이스의 처리를 나타낸 것이다. 이를 통해 거의 모든 센서의 데이터를 처리할 수 있게 한다.

Wire 의 개수에 따라 지원할 수 있는 인터페이스의 종류도 달라진다. 표 3 은 인터페이스 종류에 따른 Wire 개수를 나타낸다.

<표 2> 센서 투명성 인터페이스 처리 방법

센서 인터페이스	플랫폼 처리 방법
ADC 전압	ADC 포트에 직접연결
ADC 미약 전압	증폭 후 ADC 포트 연결
ADC 전류	저항 연결 후 ADC 포트 연결

주파수 입력	주파수 측정 포트 연결
인터럽트	인터럽트 포트 연결
SPI	SPI 포트 연결
I2C	I2C 포트 연결
Serial	Serial 포트 연결

<표 3> 센서 인터페이스 종류와 Wire 개수

Wire 개수	플랫폼 처리 방법
1 개	Analog, INT
2 개	I2C, RS485
3 개	Serial
4 개	SPI

4.2 Sensor Identification Information Block

- 센서 ID : 총 36 bit 로 표 4 와 같이 구성된다.

<표 4> 센서 ID 구성

Power Type	Sensor Interface Type	Sensor Number
4 bit	4 bit	28 bit

- 디바이스 드라이버 : Hardware Abstraction Layer 의 함수 들을 사용하여 만들며 전원 초기화, On/Off, 인터페이스 방식별 초기화/Read/Write 등으로 구분되며, 각 함수는 Hardware Mapping Table Index 를 함수인수로 사용하며 이는 센서 인터페이스 모듈의 인터페이스가 하드웨어의 연결상태 들을 나타내는 정보이다.
- Controller : 센서노드와의 통신을 제어하는 부분으로 센서모듈의 정보와 디바이스 드라이버 전송을 제어하게 된다.

4.3 Power Manager

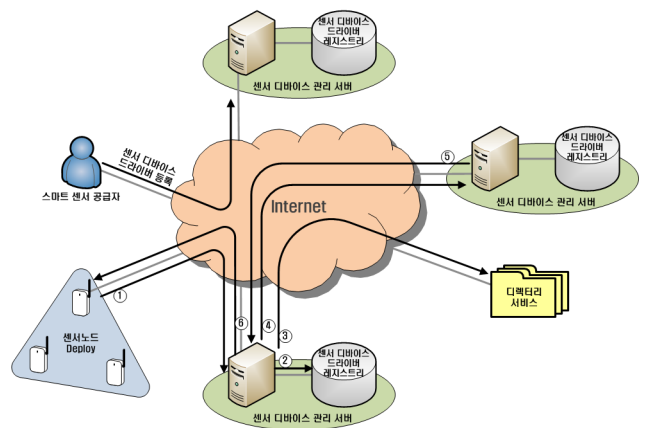
Power Manager 는 센서가 동작하는 전압은 다양하지만, 센서 모듈에서 사용하는 전압을 일정하게 유지시켜주는 역할을 한다. 센서 모듈의 동작 전압은 3V 를 기본으로 한다. 이는 대부분 센서 노드가 기본적으로 배터리 전압인 3V 로 구동되기 때문이다.

5. 원격 인식형 개념 및 구조

Plug & Play 지원 센서 인터페이스 플랫폼에서 그림 4 는 원격인식형의 구조 및 절차를 보여주고 있다. 센서모듈에 센서 디바이스 드라이버가 없거나 맞지 않은 경우 디바이스 드라이버 매니저로 센서 디바이스 드라이버를 요청한다. 센서 공급자는 센서에 맞는 센서 디바이스 드라이버를 센서 디바이스 관리서버에 등록한다. 센서노드로부터 센서 디바이스 드라이버 요청이 오면 센서 디바이스 관리 서버의 디바이스 드라이버 매니저는 센서 디바이스 드라이버 레지스트리를 검색하여 제공하고, 해당하는 디바이스 드라이버가 없는 경우 디렉터리 서비스를 통해 원하는 센서 디바이스 드라이버가 있는 센서 디바이스 관리 서버로 연결하여 센서 디바이스 드라이버를 다운받아 센서노드로 제공한다.

6. 결론

본 논문에서는 센서의 Plug&Play 기능을 위한 센서 인터페이스 플랫폼 구조와 기능을 제시하였다. 센서노드와 센서 모듈간의 하드웨어 인터페이스를 정의하였고, 센서모듈의



① 드라이버 검색 요구 ② 검색 실패 ③ 드라이버 위치 검색 요구 ④ 드라이버 요청 ⑤ 드라이버 획득 ⑥ 드라이버 다운로드

(그림 4) 원격인식형 구조 및 절차

자동인식을 위해 필요한 정보제공을 위한 센서모듈의 구성과 기능을 정의하였다. 디바이스 드라이버의 제공방식에 따라 로컬인식형과 원격인식형으로 구분하여 각 방식에 따른 구조 및 절차를 기술하였다.

기존의 개발자가 센서를 사용함에 있어 하드웨어적인 인터페이스 구성 및 디바이스 드라이버의 개발에 따른 개발 부담이 존재했으나 Plug&Play 지원 센서 인터페이스 플랫폼을 적용하여 개발기간이 단축되고 개발이 용이해지는 장점을 가지게 된다.

현재 Plug&Play 지원 센서 인터페이스 플랫폼에 대한 내용은 USN 포럼에서 표준화 절차가 진행 중에 있다. 향후 센서자체의 인식에 대한 개념을 확장하여 응용서비스를 고려한 센서모듈이 연결되면 센서가 적용되는 응용서비스에 대한 센서모듈에서 설정하거나 제공하는 정보에 대한 프로파일 등을 작성하여 제공하는 방안을 적용할 예정이다.

< Acknowledgement >

본 연구는 지식경제부의 산업원천기술개발사업 일환으로 수행되었습니다 (과제번호. 10033648). 연구비 지원에 감사드립니다.

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," Proc. of the First USENIX/ACM Symposium on Networked Systems Design and Implementation(NSDI 2004), 2004.

[2] S. Park, J. Kim, K. Lee, K. Shin, and D. Kim, "Embedded Sensor Networked Operating System," Proc. of 9th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing, 2006.

[3] 은성배, 소선섭, 김병호, "센서투명성을 지원하는 센서노드 운영체제 구조", 2008 한국컴퓨터종합학술대회 논문집 Vol.35.

[4] Institute of Electrical and Electronics Engineers, Inc., "IEEE Standard for Smart Transducer Interface for Sensors and Actuators - Network Capable Application Processor (NCAP) Information Model," Mixed-Mobile Communication Working Group of the Technical Committee on Sensor Technology TC-9 of the IEEE Instrumentation and Measurement Society, June 1999.