

논문 2013-08-01

스마트폰 USB센서 Plug & Play를 위한 통합 App 구조

(A Unified App Architecture for Plug & Play of Smart Phone USB Sensor's)

강 선 구, 소 선 섭, 김 병 호, 은 성 배*

(Seongu Kang, Sun-Sup So, Byungho Kim, Seongbae Eun)

Abstract : Performance of smart phones has increased to conduct simple PC functions that has become essential to modern people. With the development of smart phones, not only limited inner sensors but various sensors can be connected through USB. However, there is problem that app needs to be downloaded when using each sensor. In this paper, a smart phone unified app structure for USB sensor Plug&Play is presented. The structure was designed to download web pages to use various sensors in one app. Unified app communicated with the platform through ADK accessory protocol and implements structure of delivering sensor ID to the web server and downloading device driver, web page. By measurement test, it was confirmed that download time of web page in unified app was 10 times faster than original download time of original app. By managing various sensors by one app, the app is searched every time the sensor changes in which there is no need for downloading several apps.

Keywords : SmartPhone, App, USB Sensor, Plug & Play

1. 서 론

스마트폰에는 위치, 방향, 조도 등 주변 상황을 감지하는 10여개의 센서가 내장되어 있고, 이를 활용한 애플리케이션도 많다. 내장되는 센서의 종류는 한계가 있고 사용자들은 더 다양한 센서의 사용을 원한다. 또한 일상생활이나 산업현장에서 사용하는 센서의 경우 현장에 고정되어 있기 때문에 이동하거나 다른 곳에서 사용하기에 불편하다.

산업 현장에서 사용하는 센서 시스템의 경우 센서와 모니터링 장비가 하나의 세트로 구성된다. 센서만 단독으로 사용할 수 없고 모니터링 장비에 연

* Corresponding Author (sbeun@hnu.kr)

Received: 16 Nov. 2012, Revised: 13 Dec. 2012,

Accepted: 09 Jan. 2013.

S. Kang, S. Eun: Hannam University

S.S. So: Kongju University

B. Kim: Kyungsung University

※ 본 논문은 정보통신산업진흥원의 "IT/SW 창의 연구과제"의 (H0502-12-1023) 및 2012년도 한남대학교 학술연구조성비 지원에 의하여 연구하였음 (2012A194)

결하여 사용해야한다. 모니터링 장비의 경우 무겁고 이동하면서 측정하기에는 불편하다. 또한 센서들만 따로 구매하여 사용하기에는 하나의 모니터링장비에서 다양한 센서를 지원하지 않는다. 스마트폰이 손안에 PC라 불릴 정도로 성능이 발전하였고 USB 등의 인터페이스를 지원하며 휴대성이나 이동성이 편해 모니터링 장비의 대체가 가능하다.

USB 액세서리는 스마트폰의 USB포트에 액세서리형태의 기기를 연결하여 사용하는 것이다. USB액세서리는 보조배터리, 메모리카드 등 의 스마트폰에 보조적인 역할로 사용된다. 최근에는 방수능이나 나트륨, 전자파 등의 주변 환경을 탐지하는 센서를 USB 액세서리로 만들어서 사용하는 제품들이 출시되었다. 출시된 제품들의 경우 각각의 센서마다 애플리케이션이 사용되기 때문에 사용자가 일일이 다룬 받아서 설치해야하는 문제점이 있다.

본 논문에서는 스마트폰 USB센서 Plug & Play를 위한 통합 App구조를 제안한다. 제안한 구조에서는 사용자들이 센서모듈을 센서 플랫폼에 연결하면 웹서버에서 자동으로 디바이스 드라이버 및 웹페이지를 다운로드하여 처리한다. 이를 위하여 센서 플랫폼에는 기존에 계속 연구되어온 센서Plug &

Play [1-4]를 적용하여 이 프로토콜을 따르도록 설계하였다. 통합 App은 하나의 애플리케이션에서 여러 종류의 센서 값을 처리 할 수 있는 구조이다. 연결된 센서의 디바이스 드라이버 및 측정된 센서의 값을 모니터링하기 위한 웹페이지를 웹서버에서 다운로드한다. 통합 App을 통해 다양한 센서를 하나의 애플리케이션에서 사용가능하도록 그 구조를 정의 하였다.

본 논문의 구성은 다음과 같다. 2장은 USB 센서 액세서리에 대한 배경 및 관련 연구에 대해 소개한다. 3장은 통합 App 시스템의 전반적인 구조를 소개하고, 4장은 통합 App의 구현과 플랫폼 및 웹서버와의 통신 방법에 대해 정의 한다. 5장은 웹페이지 다운로드 시간에 대한 성능평가를 하고, 마지막으로 6장에서는 결론에 대해 기술한다.

II . 배경

1. USB Sensor Accessory

1.1 SmartPhone의 급부상

지하철, 버스, 자가용 등 이동하는 곳이면 어디든지 스마트폰을 사용하고 있어 스마트폰이 없으면 불편을 느낄 정도로 생활필수품이 되었다 [5].

2007년, 애플이 ‘아이폰(iPhone)’을 처음 출시하면서 이른바 손안의 PC라 불리는 스마트폰의 시장이 급격히 확대되기 시작했다. 2008년 스마트폰용 안드로이드OS가 발표되었으며, 안드로이드의 경우 오픈소스로 배포했다.

다른 휴대전화 제조회사들이 안드로이드용 스마트폰을 제작하면서 스마트폰 시장에서 아이폰과 안드로이드의 대결이 본격화 되었고 그만큼 스마트폰 시장 또한 짧은 시간에 급부상하게 되었다.

1.2 USB Accessory

스마트폰이 대중화되고 다양한 접속 장치들이 상용화되면서 액세서리 시장이 활성화 되었다 [6]. 현재 많은 종류의 액세서리가 시중에 판매되고 있으며 스마트폰의 보조적인 역할로 많이 활용된다.

그 중에서 USB 액세서리는 스마트폰에 USB포트나 단자에 액세서리형태의 기기를 연결하여 사용하는 것을 말한다. 현재는 보조배터리, 메모리카드 등의 스마트폰에 보조적인 역할로 많이 사용된다.

1.3 센서 액세서리의 시장성 증가

유희스케이아와 같은 사람들의 건강과 관련되어

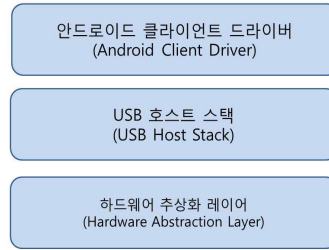


그림 1. ADK소프트웨어 내부 라이브러리 구조
Fig. 1 ADK software inner library structure

인간을 중심으로 각종 정보들을 센서 액세서리를 통하여 확인할 수 있다. 혈당 농도 센서 액세서리를 통하여 혈당을 스마트폰으로 손쉽게 체크하거나 산소포화도 및 맥박 측정 센서 액세서리를 통한 심박수 측정 등 무수히 많은 유희스케이아 센서 액세서리를 활용 할 수 있다.

사람들이 직접 행하기 어려운 작업에서 센서 액세서리를 활용하여 좀 더 쉽고 정확한 데이터를 수집할 수 있다. 기계 정비시나 세밀함이 요구되는 작업에서 미니카메라 또는 센서액세서리를 통해 모니터링 하면서 작업이 가능하다.

2. 관련연구

2.1 Android Open Accessory Development Kit

ADK(Accessory Development Kit)는 Arduino open source electronics prototyping platform 기반의 안드로이드 USB 액세서리 레퍼런스를 제공한다 [7]. 즉, 액세서리의 하드웨어 설계 파일들, 액세서리 펌웨어를 구현한 코드 그리고 액세서리와 통신하는 안드로이드 애플리케이션이 포함되어 있다.

2.2 ADK 소프트웨어 라이브러리의 구조

그림 1과 같이 ADK 보드를 크게 보면 ADK 보드를 운영하고 관리하는 부분, USB 호스트 기능을 관리하는 부분, 안드로이드 폰과 연결되었을 때 인식하고 관리하는 부분 이렇게 나눌 수 있다 [8].

2.3 USN에서의 Sensor 추상화 및 센서Plug & Play 기술

그림 2는 센서 Plug & Play를 지원하는 전체 시스템은 크게 센서 디바이스 드라이버 관리와 센서 노드에서의 처리로 나누어진다. 센서 디바이스 드라이버 관리는 센서 디바이스 드라이버 검색 프로토콜

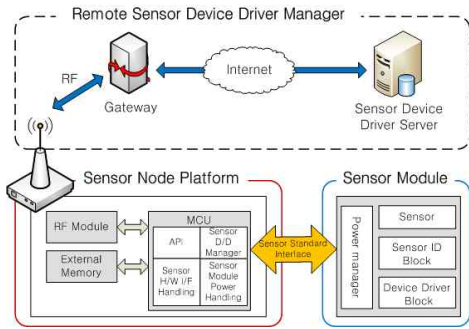


그림 2. Plug & Play 시스템 구조

Fig. 2 Plug & Play System Architecture

과 디렉터리 서비스 프로토콜, 새로운 드라이버 등록 프로토콜 등으로 이루어진다. 이 기능들은 센서 드라이버를 검색하거나 센서 드라이버를 보유하는 서버의 접근 그리고 새로운 디바이스 드라이버를 등록하는 절차들로 이루어진다.

센서노드에서의 처리는 센서 식별 정보 수신, 센서 디바이스 드라이버 비교 프로토콜, 센서 디바이스 드라이버 요청 절차 등으로 이루어진다. 이 기능들은 센서노드가 초기화 되었을 때 센서 식별 정보를 수신하고, 보유하고 있는 센서 디바이스 드라이버와 비교하여 없을 경우, 센서 디바이스 드라이버 관리 서버로 새로운 센서 디바이스 드라이버를 요청하는 절차들로 이루어진다.

2.4 아두이노 보드

스마트폰과 일반 센서 및 하드웨어를 연결하기 위한 보드로 가장 많이 사용되고 있는 것이 그림 3과 같은 아두이노(Arduino)보드이다.

아두이노 [9]는 오픈소스를 기반으로 한 피지컬 컴퓨팅 플랫폼으로, AVR기반의 보드와 소프트웨어 개발을 위한 통합 환경(IDE)를 제공한다. 아두이노는 많은 스위치나 센서로부터 값을 받아들여, LED나 모터와 같은 것들을 통제함으로써 환경과 상호작용이 가능한 물건을 만들어낼 수 있다. 또한 플래시, 프로세싱, Max/MSP와 같은 소프트웨어를 연동할 수 있다. 아두이노 보드의 회로도가 CCL에 따라 공개되어 있으므로, 누구나 직접 보드를 직접 만들고 수정할 수 있다.

2.5 IEEE 1451

IEEE 1451 [10]은 1451.0 ~ 1451.6의 총 7개의 표준으로 나누어 진행 중인 표준이며, 아날로그 및 디지털 인터페이스 모두를 제공하는 혼합 모드 트

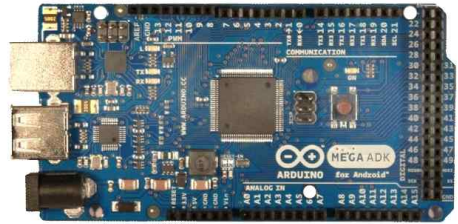


그림 3. 아두이노 보드

Fig. 3 Arduino board

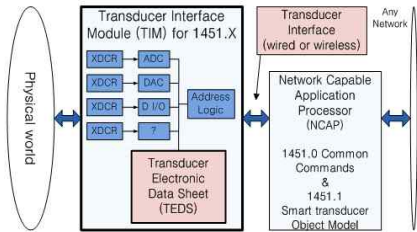


그림 4. IEEE 스마트 트랜스듀서 구조

Fig. 4 IEEE Smart Transducer Structure

랜스듀서 개념으로 정의된다. 이 표준의 핵심은 Plug & Play 작동이 가능하다는 것이다. 또한 트랜스듀서의 다양성을 극복할 수 있게 한다.

그림 4에서 보는 바와 같이 IEEE 1451의 스마트 트랜스듀서는 Transducer Interface Module(TIM)과 Network Capable Application Processor(NCAP)로 나누어진다. 여기서 TIM의 범위는 현재 사용되고 있는 센서 노드의 거의 모든 부분을 포함 하고 있는 범위이다. 그래서 센서 Plug & Play는 내부의 물리적 센서와의 연결 부분이므로 IEEE1451의 범위와는 별개의 부분이다.

3. 기존연구의 문제점

센서 Plug & Play를 지원하는 시스템은 일반적인 센서들을 다루기 위한 기술이기 때문에 스마트폰과 연결하기 위한 USB Host기능이 없고, 통신에 필요한 안드로이드 액세서리 프로토콜이 구현되어 있지 않기 때문에 스마트폰과의 통신이 불가능하다는 문제점이 있다.

아두이노는 독자적인 개발환경을 가지고 있기 때문에 기존의 센서 개발자들이 개발하기 위해서는 다시 개발 툴의 사용법을 익혀야한다. 또한 스마트폰으로부터 전원을 공급받지 못하기 때문에 외부전원을 연결해야하는 문제점이 있다.

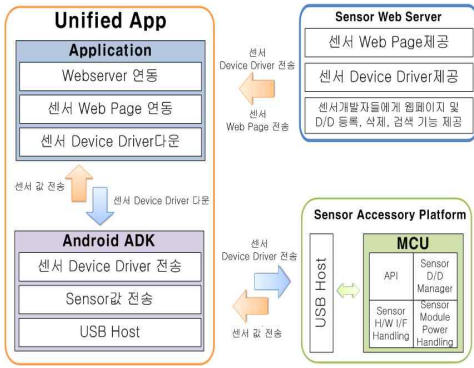


그림 5. 전체 시스템 구조
Fig. 5 Overall system structure

센서 Plug & Play나 아두이노 모두 하나의 애플리케이션으로 다양한 센서의 연결 및 처리에 관한 구조를 제시하지 못했다.

III. 설 계

1. 전체 구조도

USB센서 Plug & Play를 지원하는 스마트폰 통합 App의 전체 시스템은 그림 5와 같이 센서 액세서리 플랫폼과 센서 웹서버, 통합 App으로 나누어 처리된다.

센서 액세서리 플랫폼은 USB Host, API, sensor D/D Manager, sensor H/W I/F Handling, Sensor Module Power Handling으로 이루어진다. 이 기능들은 센서모듈이 연결되면 해당 센서의 디바이스 드라이버의 유무를 확인 하거나, 센서로부터 측정된 값을 읽어오고, USB를 통해 스마트폰으로 디바이스 드라이버를 다운 받거나 센서 값을 전송하는 일을 수행한다.

센서 웹서버는 센서모듈의 디바이스 드라이버와 웹페이지를 제공한다. 통합 App에서 디바이스 드라이버나 웹페이지다운 요청이 들어오면 요청 센서에 해당되는 디바이스 드라이버나 웹페이지는 다운로드 해주는 기능을 제공한다. 웹서버는 센서 개발자에게 디바이스 드라이버나 웹페이지를 등록, 삭제, 검색 할 수 있는 기능을 제공한다.

통합 App은 ADK를 통해 센서 액세서리 플랫폼과 통신을 하고 웹서버에서 디바이스 드라이버 및 센서 웹페이지 다운의 기능을 제공한다. 센서의 값을 받아서 웹페이지와 연동하여 사용자에게 보여주고, 디바이스 드라이버를 센서 액세서리 플랫폼에 다운로드 해준다.

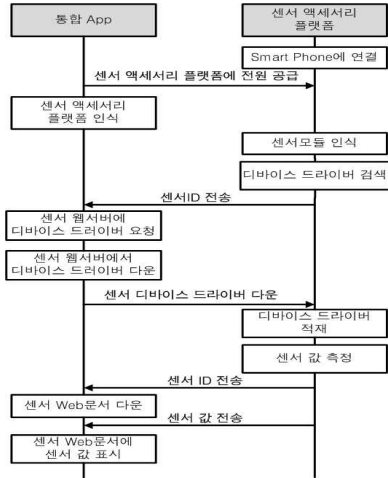


그림 6. 센서 액세서리 플랫폼 인식 절차
Fig. 6 Sensor accessory platform of recognition procedures

2. 통합 App의 구조

2.1 통합 App과 센서 액세서리 플랫폼

그림 6은 통합 App에 센서 액세서리 플랫폼이 연결 되었을 때 인식 및 디바이스 드라이버의 다운, 센서 값 전송에 관한 처리 과정을 보여준다.

통합 App은 사용자에게 센서 액세서리 플랫폼의 연결을 알리고 사용자가 확인하면 연결을 승인한다. 센서 모듈이 플랫폼에 연결되면 플랫폼은 센서모듈의 디바이스 드라이버의 유무를 확인하고 없을 경우 통합 App으로 센서ID를 전송한다.

통합 App에서는 플랫폼이 요청한 센서ID를 통해 웹서버에서 해당 센서의 디바이스 드라이버를 다운로드하고 플랫폼으로 전송한다. 플랫폼에서는 다운로드 된 디바이스 드라이버를 적재하고 센서와의 통신을 통해 측정된 값을 통합 App으로 전송한다. 이때 통합 App에서는 해당 센서의 웹페이지의 유무를 확인하고 없을 경우 웹서버에 요청하여 다운로드한다. 다운로드가 완료되면 웹페이지를 통해 센서의 값을 사용자에게 보여준다.

2.2 통합 App과 센서 웹서버

그림 7은 통합 App이 센서 웹서버에서 센서의 디바이스 드라이버 및 웹페이지 다운요청의 처리과정을 보여준다.

센서 액세서리 플랫폼에서 디바이스 드라이버의 다운 요청이 들어오면 통합 App은 웹서버에 센서ID를 전송한다. 웹서버에서는 요청이 들어온 센서ID를

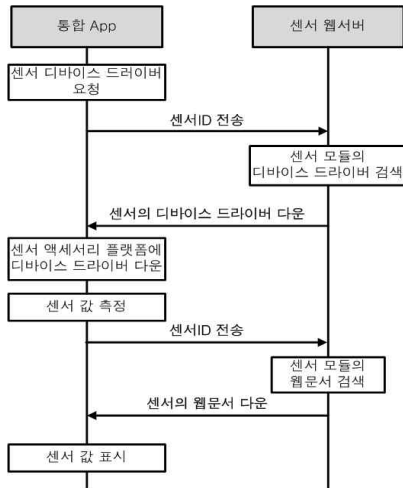


그림 7. 웹서버의 다운요청 절차
Fig. 7 Download request procedure of web server

통해 디바이스 드라이버를 검색하여 결과를 알려주면 통합 App에서 결과 값을 통해 다운로드 한다.

다운로드한 디바이스 드라이버는 센서 액세스리 플랫폼으로 전송하고, 플랫폼이 센서모듈을 인식하면 측정된 센서의 값을 통합 App으로 전송한다. 통합 App은 센서의 웹페이지를 확인하여 없을 경우 센서 ID를 웹서버로 전송한다. 웹서버는 센서의 웹페이지를 검색하여 결과 값을 반환하고, 통합 App에서는 결과 값을 통해 웹페이지를 다운로드 하여 센서의 값을 사용자에게 보여준다.

센서모듈이 최초로 연결 시에만 다운로드를 하기 때문에 그 이후부터는 다운로드를 하지 않는다. 다시 다운로드를 하지 않고, 바로 웹페이지를 읽기 때문에 센서 값을 보다 빠르게 표시 할 수 있다.

IV. 구현

안드로이드 기반의 스마트폰에서 통합App을 구현하였고 센서 액세스리 플랫폼은 ATmega128에 스마트폰과 USB통신을 할 수 있는 MAX3421칩을 사용하였으며 웹서버는 php와 MySQL을 통해 제작되었다.

1. 센서 액세스리 플랫폼

아두이노는 독자적인 개발환경을 가지고 있다. 독자적인 개발환경 때문에 기존의 센서 개발자들이 센서를 개발하기 위해서는 다시 아두이노의 개발

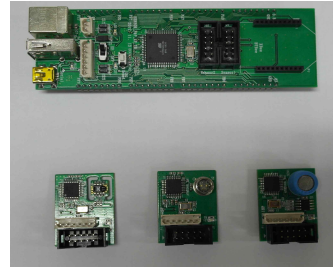


그림 8. 센서 액세스리 플랫폼의 프로토타입
Fig. 8 Prototype of sensor accessory platform

툴 사용법을 익혀야하는 번거로움이 있다. 그래서 센서 액세스리 플랫폼은 아누이노 대신에 센서 개발자들이 기존에 사용하고 있는 개발환경을 그대로 사용할 수 있도록 ATmega128에 USB통신을 할 수 있는 MAX3421 [11]칩을 연결하여 그림 8과 같이 프로토타입을 제작하였다. 또한 기존의 개발환경을 그대로 사용할 수 있기 때문에 아두이노 개발툴을 사용하지 않아도 개발이 가능하다.

USB HOST기능을 탑재한 칩인 MAX3421을 사용하고 구글 ADK API에서 제공하는 안드로이드 액세스리 프로토콜을 바탕으로 센서 액세스리 플랫폼과 통신을 할 수 있는 기능을 구현하였다. Sensor Standard H/W Interface는 센서 Plug & Play 액세스리 플랫폼과 센서모듈 간 표준화된 접속을 제공하기 위한 인터페이스로써, 센서 식별정보를 주고받고 센서 특성에 따라 ADC, I2C, SPI, Interrupt등 다양한 인터페이스를 제공하여 Plug & Play가 가능하도록 해준다.

센서모듈은 온/습도, 조도와 같은 물리적인 센서가 장착되며 센서가 갖는 인터페이스 방식에 의해 센서 H/W 인터페이스가 구성되어 Sensor Plug & Play 액세스리 플랫폼에 직접 연결된다.

2. 통합 App

통합 App은 기본적으로 USB통신을 위한 ADK를 기반으로 만들어졌다. ADK는 구글에서 안드로이드기기를 USB로 연결하여 데이터를 주고받을 수 있는 API이다.

2.1 통합 App 전체 서비스 시스템

통합 App은 그림 9와 같이 메인시스템, 파일 다운로드부분으로 구성된다. 메인 시스템은 안드로이드에서 동작하는 일반적인 애플리케이션으로 통합 App

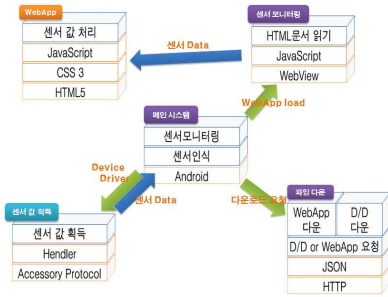


그림 9. 통합 App 전체 서비스 시스템
Fig. 9 ntegrated app service system

의 메인을 담당하게 된다. 웹페이지는 HTML5와 CSS3, JQuery로 제작되었다. 웹페이지는 안드로이드의 모든 디바이스에서 동작이 가능하고 플랫폼마다 따로 만들어야 하는 번거로움을 줄일 수 있다.

센서 액세서리 플랫폼으로부터 센서의 값을 받거나 디바이스 드라이버를 주기 위해서는 구글에서 제공하는 안드로이드 액세서리 프로토콜의 형식에 맞춰서 통신을 해야 한다. 통신이 환경이 수립되면 플랫폼과 통신하여 센서에 대한 디바이스 드라이버를 다운로드 하거나 센서의 값을 받아서 처리 해줄 수 있다.

서버에는 JSON을 통해서 정보를 주고받는다. 자신이 원하는 정보가 있을 경우, 웹 주소에 원하는 데이터의 정보를 실어서 보내면 서버에서는 그 데이터에 해당하는 결과를 반환한다. 그러면 통합 App은 반환된 결과를 해석하여 디바이스 드라이버 및 웹페이지를 다운로드 한다.

2.2 통합 App과 웹서버간의 통신

통합 App과 웹서버는 JSON을 기반으로 데이터를 주고받는다. 표 1은 통합 App에서 웹서버에 정보 요청 할 때의 포맷이다.

웹서버에 정보를 요청하기 위해서는 표 1의 형식에 맞춰야 한다. 정보요청은 기본적으로 URL 형식을 취하고 있다. 처음은 웹서버에 접속을 위한 홈페이지 주소이고 다음에 오는 것이 웹서버에서 통합 App이 원하는 서비스의 이름이다. Class 이름은 서비스에서 제공하는 기능들 중에서 통합 App이 원하는 기능에 대한 Class이름으로 바로 다음에 오는 Class 함수에서 실제로 다운로드를 하기 위한 기능이 정의 되어 있다. 그 다음에 센서 플랫폼으로부터 받은 센서ID를 입력하고, Type을 입력한다. Type은 표 2와 같이 디바이스 드라이버를 다운로드 할 것인지 웹페이지를 다운로드 할 것인가에 대한 것을 정의하는 부분이다.

표 1. 웹서버에 정보요청 포맷

Table 1. Information request format to web server

홈페이지 주소	서비스 이름	Class이름	Class함수	센서ID	Type
---------	--------	---------	---------	------	------

표 2. Type의 분류

Table 2. Categorization of type

Type	정의
1	Device Driver 다운로드 요청
2	웹페이지 다운로드 요청
3	D/D와 웹페이지 다운로드 요청

표 3. 반환 데이터 포맷

Table 3. returned data format

ID	Type	File_Path	File_name
----	------	-----------	-----------

웹서버는 통합 App에서 요청이 들어오면 DB에서 요청한 센서의 디바이스 드라이버 및 웹페이지를 검색하여 결과 값을 반환해준다. 결과 값 반환은 표 3과 같은 데이터 패킷 포맷에 따른다.

ID와 Type은 통합 App이 요청한 센서ID와 다운로드 Type이다. File_Path는 웹서버 상에 다운로드 파일이 저장되어 있는 위치를 나타내고 File_name은 다운로드할 파일의 이름을 나타낸다. 반환된 데이터를 통합 App에서 JSON을 통해 File_Path와 File_name을 추출하고, 두 개의 값을 조합하여 원하는 파일을 다운로드 한다.

2.3 통합 App과 센서 액세서리 플랫폼의 통신

일반적으로 액세서리 연결은 4가지의 절차를 수행한다. 기다리면서 연결된 장비들을 검색하고, 장비의 액세서리 모드 지원을 결정한다. 그리고 필요하다면 액세서리모드에 장비를 시작시키려고 시도하고, 만약 장비가 Android Accessory Protocol을 지원한다면 통신을 위한 환경을 수립하는 과정을 거친다.

연결 초기화 동안 액세서리는 연결된 장비의 USB 장비 기술자의 벤더와 제품 ID를 체크한다. 만약 이미 액세서리 모드로 동작하고 있다면 벤더 ID가 Google의 ID(0x18D1)와 제품 ID가 0x2D00 혹은 0x2D01인지를 체크한다. (0x2D00은 액세서리 모드를 지원하는 안드로이드 장비들을 위해 예약되어져 있고 0x2D01은 액세서리 모드뿐만 아니라 ADB(Android Debug Bridge) 프로토콜을 지원하는 장비들을 위해 예약되어져 있다.) 만약 맞으면 액세



그림 10. 온/습도 및 가스 센서의 웹문서
Fig. 10 Web document of temperature, humidity, gas sensor

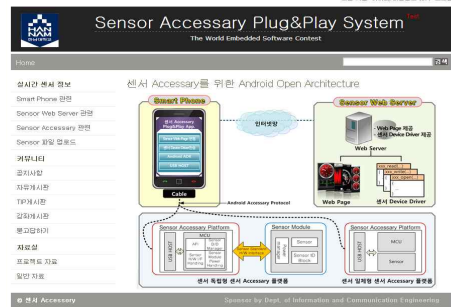


그림 12. 센서 웹서버 메인 페이지
Fig. 12 Sensor Web Server Main Page

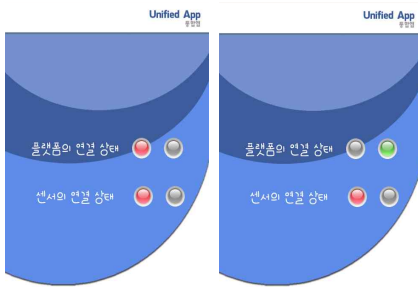


그림 11. 센서 액세서리 플랫폼의 연결
Fig. 11 Connection of Sensor Accessory Platform

서리는 통신 프로토콜에 따라 벌크(bulk) 전송 endpoint들을 통해 장비와의 통신을 할 수 있는 환경을 수립한다.

안드로이드와 장비가 통신할 수 있는 환경이 수립되면 그림 10과 같이 확인이 가능하다. 장비는 자신에게 연결된 센서모듈이 있는지 확인하고 없으면 연결 될 때 까지 기다리고 연결된 것이 있으면 센서 모듈의 디바이스 드라이버의 유무를 확인한다. 디바이스 드라이버가 있다면 바로 센서의 값을 읽어서 통합 App으로 전달해준다. 만약에 없다면 통합 App은 웹서버에 센서ID를 전송하여 디바이스 드라이버를 요청한다. 센서의 값을 읽을 때도 웹페이지 유무를 확인하여 없을 시 그림 11과 같이 각 센서의 웹페이지를 웹서버에서 다운로드 한다.

기존의 안드로이드 액세서리 플랫폼에 센서모듈의 디바이스 드라이버의 유무를 확인하고 요청하는 것과 측정된 센서의 값을 읽어 들이는 기능을 추가하여 센서 Plug & Play기술이 가능하도록 하였다.

3. 센서 웹서버

센서모듈의 디바이스 드라이버와 센서 값을 표시해주는 웹페이지를 제공해주는 웹서버를 그림 12와 같이 제작하였다. 센서 액세서리 플랫폼은 모든 센서모듈에 대한 디바이스 드라이버를 가지고 있을 수 없고 새로 만들어진 센서모듈의 디바이스 드라이버는 가지고 있지 않기 때문에 센서모듈이 연결되면 해당 센서모듈의 디바이스 드라이버를 웹서버에서 가지고 올 수 있는 기능을 웹서버에서 제공한다.

그리고 통합 App에 전송되어지는 센서들의 값을 웹페이지에 표시해 줌으로써 다양한 센서들의 값을 표시해 줄 수 있다. 통합 App에서 바로 센서들의 값을 표시해 줄 경우 센서들의 값과 단위들이 모두 달라 많은 종류의 Activity가 필요해지므로 좀더 능동적인 센서들의 값을 표시 해주기 위해 웹페이지 형식으로 제공한다.

센서개발자들이 자신이 만든 센서의 디바이스 드라이버 및 웹페이지를 서버에 등록할 수 있도록 웹서버의 홈페이지에서 등록 및 삭제, 검색 기능을 제공한다.

V. 성능 분석

본 장에서는 센서 액세서리 플랫폼에 센서모듈이 연결되어 통합 App에 센서 값을 전송할 때, 웹페이지를 다운로드해야할 경우 센서 웹서버에서 웹페이지를 다운로드하는 시간을 측정하였다. 다운로드 파일은 온/습도 및 가스 센서의 웹페이지로 하였고 시간 측정은 3G와 WI-FI, 두 자기 환경에서 실험하였다.

표 4. 3G환경에서의 다운로드 시간 측정

Table 4. Download time measurements in 3G environment

웹페이지의 크기(단위:Kbyte)	다운로드 시간(단위:ms)	
	건물 안	건물 밖
1.7	1,422	684
4.7	1,000	208
3.1	378	341
6.5	1,007	222
11.9	761	419
8.8	628	635

1. 3G환경에서 다운로드 시간 측정

3G환경에서 다운로드 시간 측정은 표 4와 같이 건물 안과 밖으로 나누어 테스트하였다. 건물 밖의 경우 일부 수신 상태가 나쁜 구역을 제외하고는 평균적으로 700(ms)을 넘지 않았다. 건물 안의 경우는 전체적으로 수신 상태가 낮아 다운로드 시간이 오래 걸리는 구역이 많았다. 다운로드 시간이 평균적으로 1초 정도의 시간이 걸리지만 사람이 느리다고 느낄 정도의 시간은 아니다.

2. WI-FI환경에서 다운로드 시간 측정

WI-FI환경에서의 다운로드 시간 측정은 표 5와 같이 여러 장소에서 제공하는 무선공유기에 접속하여 측정하였다. 무선공유기의 수신범위를 이탈하거나 멀어질 경우 수신감도가 낮아져서 다운로드 시간이 늘어났지만 평균 다운로드 속도가 100(ms)을 넘지 않는 시간으로 사람이 인지할 정도는 아니다.

VI. 결 론

본 논문에서는 USB센서 Plug & Play를 위한 스마트폰 통합 App구조를 제안하였다. 통합 App에서 센서 액세서리 플랫폼과의 연결방법 및 센서 웹서버에서 디바이스 드라이버 다운로드 방법을 제시하였고, 연결된 센서모듈에 맞는 웹페이지를 센서 웹서버에서 다운로드 하는 방법을 제시하였다. 본 논문에서 제시한 웹페이지 다운로드 방법에서 측정된 3G환경의 다운로드 속도가 평균적으로 1초의 시간이 걸렸지만, 일반 App이 검색시간을 포함한 다운로드 시간이 1분 이상 소요되기 때문에 이보다 빠른 속도임을 확인하였다.

현재까지는 센서의 연결을 확인하고 하나의 애플리케이션에서 다양한 센서들을 처리하여 값

표 5. WI-FI환경에서의 다운로드 시간 측정

Table 5. Download time measurement in WIFI environment

웹페이지의 크기(단위:Kbyte)	다운로드 시간(단위:ms)
1.7	89
4.7	50
3.1	51
6.5	49
11.9	59
8.8	69

을 보여주기만 하였다. 추후에는 센서들로부터 얻은 값을 가공하여 다양한 정보들과 함께 제공할 수 있는 서비스에 대한 연구가 진행되어야 할 것이다.

참고문헌

- [1] S.B. Eun, S.S. So, B.H. Kim, "A Sensor Node Operating System Architecture Providing Sensor Transparency," Proceedings on General Conference of Dissertaions by KIISE, Vol. 35, No. 1, 2008 (in Korean).
- [2] S.H. Bang, S.B. Eun, "A Sensor Device Manager Supporting Sensor Transparency," Proceedings on Autumn Annual Conference of dissertations by KIISE, Vol. 15, No. 2, 2008 (in Korean).
- [3] Y.B. Park, "A Sensor Node Platform Supporting Sensor Plug & Play," Thesis of masters degree, 2009 (in Korean).
- [4] J.S. Park, S.B. Eun, N.S. Kim, Y.B. Moon, "Dynamic Loading and Linking of Sensor Device Drivers for Sensor Module Plug & Play," Proceedings on Autumn Annual Conference of IEMEK, pp.71-74, 2011 (in Korean).
- [5] Y.J. Jang, C.W. Kim, "The Evolution of Smarthphone Market & The Effect by Android," Journal of KIISE, Vol. 28, No. 5, pp.48-56, 2010 (in Korean).
- [6] http://www.koreasmart.or.kr/bbs_shop/read.htm?board_code=news&idx=291
- [7] <http://developer.android.com/tools/adk/index.html>

- [8] http://www.imaso.co.kr/?doc=bbs/gnuboard.php&bo_table=article&wr_id=38180
- [9] <http://www.arduino.cc/>
- [10] 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.
- [11] Maxim Corporation, MAX3421 Datasheet, 2006

저 자 소 개

강 선 구



2012년 한남대 정보통신 공학과 학사.
 현재, 한남대 정보통신공학과 석사과정.
 관심분야: 스마트폰, 안드로이드, WebApp

Email: kangseongu87@gmail.com

소 선 섭



1986년 이화여자대학교 전산학과 학사.
 1988년 KAIST 전산학과 석사.
 2001년 KAIST 전산학과 박사.

현재, 공주대학교 컴퓨터공학부 교수.
 관심분야: 유비쿼터스 센서 네트워크, 소프트웨어 엔지니어링.

Email: triples@kongju.ac.kr

김 병 호

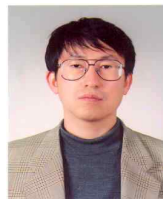


1990년 연세대학교 전산학과 학사.
 1997년 KAIST 전산학과 석박사.
 2007년-현재 경성대학교 컴퓨터학부 조교수

관심분야: 컴퓨터구조, 센서네트워크, 모바일 OS

Email: bkim@ks.ac.kr

은 성 배



1985년 서울대 전산학과 학사.
 1995년 KAIST 전산학과 석박사.
 현재, 한남대학교 정보통신공학과 교수

관심분야: 실시간시스템, 유비쿼터스, 센서네트워크

Email: sbeun@hnu.kr