

스마트 센서 응용 소프트웨어를 테스트하기 위한 효율적인 방법

조 장 우*, 정 환 철*

An Effective Method of Testing Application Software of Smart Sensors

Jang-Wu Jo *, Hwan-Cheol Joeng *

요 약

본 논문에서는 스마트 센서 응용 소프트웨어를 테스트하기 위한 효율적인 방법인 가상 센서 시스템을 제안한다. 센서 응용 소프트웨어를 테스트하는 보편적인 방법은 테스트 보드에 센서를 직접 연결시켜 테스트 환경의 센서 측정값으로 응용소프트웨어를 테스트 하는 것이다. 센서 측정값을 입력으로 센서 응용 소프트웨어를 테스트함으로써 발생하는 문제는 테스트 데이터가 제한적이라는 것이다. 즉, 테스트 데이터가 센서로부터 생성되기 때문에 소프트웨어 테스터가 테스트 데이터를 조절하지 못하는 문제가 있다. 이러한 문제를 해결하기 위해 가상 센서 시스템을 제안한다. 가상 센서 시스템은 소프트웨어 테스터가 센서의 측정값을 조절할 수 있게 한다. 가상 센서 시스템에서 센서 선택, 센서 특성화, 출력 패턴 정의의 세 단계를 통해 가상 센서를 정의한다. 가상 센서 시스템을 통해 조절 가능한 센서 측정값을 사용함으로써 센서 응용 소프트웨어에 대한 효율적인 테스트가 가능하다. 본 연구의 유용성을 보기 위해 가상 센서 시스템을 안드로이드 앱의 센서 프로그램에 적용해 보고 실험 결과를 보인다.

▶ Keywords : 센서, 가상 센서, 센서 응용 소프트웨어 테스트, 임베디드 소프트웨어 테스트

Abstract

This paper presents a virtual sensor system that is an effective method to test application software of smart sensors. The common way of testing sensor application is to build a test board, connect sensors to the board, and test sensor applications on the board with sensor's measurements as inputs. The problem of testing sensor application software with sensor's measurements as inputs is the restriction of test data. In other words, software testers cannot manipulate test data, because test data is generated by sensors. To solve this problem a virtual

•제1저자 : 조장우 •교신저자 : 조장우

•투고일 : 2013. 7. 17, 심사일 : 2013. 7. 24, 게재확정일 : 2013. 8. 14.

* 동아대학교 컴퓨터공학과(Dept. of Computer Engineering, Dong-A University)

※ 본 연구는 2012년 부산테크노파크 산학공동기술혁신사업의 지원으로 수행되었음.

sensor system is presented in this paper. The virtual sensor system enables software testers to manipulate measurements of sensors. In the virtual sensor system, generation of virtual sensors comprises three stages - sensor selection, sensor characterization, and determination of output patterns. Sensor's measurements that can be manipulated through the virtual sensor system make the process of testing efficient. To show the usefulness of our virtual sensor system, it is applied to sensor applications in Android platform and the result of experiments is shown.

▶ Keywords : sensors, virtual sensors, sensor application software testing, embedded software testing

I. 서론

센서는 외부 자극을 받아 이것을 전기 신호로 변환하는 소자로 정의할 수 있다[1]. 스마트 센서는 위의 기본 기능 이외에 전기 신호를 디지털 신호로 변환하여, 모니터링 또는 단순 제어와 같은 간단한 처리를 수행하고 주어진 통신기능을 이용하여 중앙제어장치(Central Control Unit)에 센서 측정값을 보내는 기능을 수행한다[2]. 센서 응용 소프트웨어는 중앙 제어장치에서 동작하는 프로그램을 의미하고, 스마트 센서로부터 센서측정값(디지털 신호)을 전달받아 측정 값을 모니터링하거나 분석하여, 구동기(actuator)를 구동시키는 일을 수행한다[3]. 스마트 센서와 센서 응용 소프트웨어는 현재 많은 분야에서 사용되고 있다: 자동차 시스템[4, 5], 해양분야[6, 7, 8], 의료분야[9, 10, 11], 등. 그리고 센서 관련 소프트웨어의 결함으로 인한 사고도 많이 보고되고 있다: 나로호 발사 실패, 아우디 A6 감속 센서 SW 결함, BMW 인젝션 펌프 SW 결함, 현대차 에어백 SW 결함, 토요타 프리우스 리콜, Piper Alpha 화재 등 [12].

소프트웨어 테스트는 소프트웨어의 결함을 찾아내고 개선하는 일련의 절차이다[13]. 이를 위하여 소프트웨어에 입력을 주고 실행시켜 제대로 동작하는지 확인한다. 가능한 많은 입력을 실행시켜보기 위해서 테스트 데이터를 생성하는 방법이 제안되고 있다[14, 15].

본 논문에서는 센서 응용 소프트웨어를 테스트하기 위한 효율적인 방법을 제안하고자 한다. 우선 센서 응용 소프트웨어를 테스트하기 위한 기존의 방법을 살펴보고, 이의 문제점을 제안하고자 한다. 그리고 이 문제를 해결할 수 있는 방법인 가상센서 시스템에 대해서 소개하고, 안드로이드 센서 응용 앱에 적용하는 예를 보인다.

본 논문의 구성은 다음과 같다. 2 장은 연구배경으로 소프트웨어 테스트의 개념, 스마트 센서를 사용하는 시스템의 구조, 스마트 센서 응용 소프트웨어의 테스트, 기존 테스트 방법의 문제점을 기술하며, 3 장에서는 본 논문에서 제안하는 가상 센서를 이용한 테스트의 개념에 대해 소개한다. 4 장에서는 가상 센서의 사용 예를 실행 화면 위주로 소개한다. 5 장에서는 가상 센서를 안드로이드 앱에 적용하기 위하여 안드로이드 센서 프레임워크를 수정하는 부분과 안드로이드 앱에 적용한 결과를 설명한다. 아울러 6 장에서는 결론과 함께 향후 연구 내용에 대해 서술한다.

II. 연구 배경

소프트웨어 테스트는 단위테스트, 통합테스트, 시스템테스트, 인수테스트 순서로 단계적으로 수행된다. 각 단계별로 소프트웨어의 구성단위, 구성단위의 통합, 소프트웨어의 요구사항 명세의 구현, 사용자의 요구사항의 반영이 올바르게 이루어지고 있는지를 확인한다. 임베디드 소프트웨어에 대한 시스템 테스트는 하드웨어와 소프트웨어를 통합하여 요구사항 명세에 맞게 구현되었는지 확인한다[16].

센서 응용 소프트웨어는 일종의 임베디드 소프트웨어이므로, 이의 시스템 테스트는 타겟 환경과 같은 보드 수준의 시스템을 구축하여 실행시켜 보는 것이다[3]. 센서 응용 소프트웨어가 동작할 환경과 동일한 환경의 테스트 보드를 제작하고, 테스트 보드와 스마트 센서를 연결한다. 이와 같이 구축된 시스템에서 스마트 센서의 측정값에 대해 응용 소프트웨어가 제대로 동작하는지 확인한다.

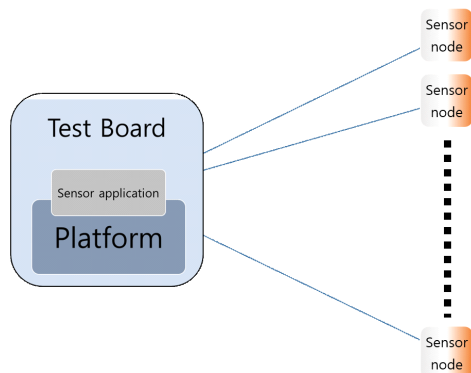


그림 1. 테스트보드와 센서의 구조
Fig. 1. Configuration of a Test board and Sensors

그림 1과 같이 테스트 보드에 스마트 센서 노드들이 연결되어 있고 테스트 보드에는 센서 응용 소프트웨어가 실행된다. 각 스마트 센서 노드에는 각자의 센서 노드 프로그램이 동작하여 센서 측정값에 대한 간단한 처리를 수행하거나, 센서 측정값을 테스트 보드의 응용 소프트웨어에 전달한다. 스마트 센서와 센서응용 소프트웨어가 연결되는 방식은 Wi-Fi, Bluetooth, Mote, RFID, 시리얼통신, 병렬통신, 이더넷통신, CAN(Controller Area Network) 통신, 등이 있다.

본 연구에서 테스트하고자 하는 소프트웨어는 그림 1의 센서 응용 소프트웨어(Sensor applications)이다. 스마트 센서가 고성능화, 소형화되는 추세로 발전하고 있으므로, 그림 1의 센서 응용 소프트웨어는 필요성이 없을 것으로 생각할 수 있으나, 스마트 센서에서는 수행할 수 없는 복잡한 처리는 센서 응용 소프트웨어에서 수행해야 한다. 예를 들어, 그림 1의 센서노드가 CO₂ 센서로 화재 감지 및 경보 기능을 제공한다고 가정하자. 각각의 센서들은 독립적으로 CO₂ 센서에서 감지된 값이 어느 한계 이상이면 화재로 인식하고 경보 기능을 수행할 수 있다. 연기가 피치면 인접한 센서들도 화재로 인식을 하게 되지만, 어느 위치에서 화재가 발생하였는지에 대해서는 알 수가 없다. 이를 위해서는 센서응용 소프트웨어가 각각의 센서로부터 정보를 받아서 화재 위치를 알 수 있다. 이와 같이 스마트 센서의 기능은 센서 측정값이 한계치(threshold)를 벗어나는 지 여부를 확인하여 그에 대한 동작을 수행하는 것이고, 센서 응용 소프트웨어의 기능은 여러 개의 스마트 센서들의 측정값들을 이용하여 필요한 정보를 추출하여 그에 대한 동작을 수행하는 것이다.

그림 1과 같은 환경에서 시스템 테스트를 수행할 경우, 센

서 응용 소프트웨어의 테스트 데이터가 제한적인 문제가 있다. 센서 응용 소프트웨어의 테스트 데이터는 센서로부터 전달되는 측정값이다. 테스트 데이터는 입력 값의 범위에서 가능한 많은 값을 포함할 수 있어야 한다. 그러나 센서 측정값은 센서가 작동하는 환경에 의존적인데, 센서가 측정할 수 있는 값들 중에서 가능한 많은 값을 제공하기 위해서는 센서가 작동하는 환경을 변경시켜야 하므로, 테스트 데이터 집합을 크게 하는 것은 비용이 많이 든다. 예를 들어, 온도 센서의 경우 측정값은 현재 환경의 온도인데, 다양한 측정값을 제공하기 위해서, 즉 테스트 데이터의 집합을 크게 하는 것은 센서 작동 환경의 온도를 변경시켜야 한다.

가상의 값을 입력으로 소프트웨어를 테스트하는 기존의 도구들을 살펴본다. 이와 같은 도구들은 테스트되는 소프트웨어가 실행되는 플랫폼에 의존적이므로, 모든 플랫폼을 지원하는 일반적인 도구들을 찾을 수 없었다. 실행 플랫폼을 안드로이드로 제한하면, DDMS(Dalvik Debug Monitor Server)[17]와 monkeyrunner[18]가 있다. DDMS는 디버깅 도구로서 메모리, 스레드, 및 메소드에 관련된 디버깅 정보를 제공할 뿐만 아니라, 가상의 GPS 정보를 이용하여 소프트웨어를 실행할 수 있다. 그리고 monkeyrunner는 테스트 도구로서, 안드로이드 기계 또는 가상기계에서 실행되는 안드로이드 앱의 입력을 안드로이드 기계 외부에서 조절할 수 있는 기능을 제공한다. 즉, 안드로이드 기계에서 실행되는 앱의 입력을 안드로이드 기계 외부인 PC에서 조절할 수 있다. 본 연구에서 제안한 방법과 기존의 도구들의 비교는 5장 4절에서 기술한다.

III. 가상 센서를 이용한 테스트

본 논문에서는 실제 센서를 사용하지 않고, 센서 응용 소프트웨어를 테스트하기 위하여 센서 측정값을 조절할 수 있는 가상 센서를 제안한다. 본 논문에서 제안하는 가상 센서의 목표는 다음과 같다. 1) 센서 응용 소프트웨어의 입장에서 실제 센서와 가상 센서의 사용방법은 같다. 즉, 센서 응용 소프트웨어의 수정 없이 가상 센서로 대체 가능하다. 2) 가상 센서는 실제 센서가 측정가능한 모든 범위의 값을 생성 가능 하다. 3) 여러 개의 센서들에 대해서도 센서들의 측정값을 동시에 생성 가능하다.

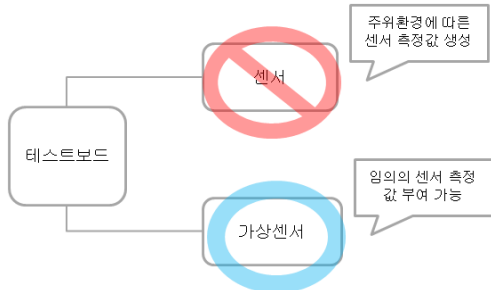


그림 2. 가상 센서를 사용한 테스트의 개념
Fig. 2. Concept of testing with virtual sensors

그림 2는 실제 센서를 이용하지 않고 가상 센서를 사용하여 테스트하는 개념이다. 실제 센서의 측정값을 가상 센서에 설정하여 타겟 보드로 전송해 응용 SW를 테스트한다. 그림 2에서 실제 센서를 사용하면 센서 측정값에 대해서만 테스트가 가능하다. 가상 센서를 사용하면 임의의 센서 측정값에 대해서도 테스트가 가능하다. 가상 센서는 센서선택, 센서특성화, 출력패턴 정의와 같이 세 단계의 과정을 통해 정의된다.

센서 선택 단계에서는 기기에서 사용할 센서들을 선택한다. 여러 개의 센서를 사용할 경우 추가 과정의 반복을 통하여 계속 추가 할 수 있다. 센서특성화 단계는 각 센서의 특성을 정의한다. 특성화 값은 센서의 특성을 정의하기 위하여 사용자가 지정해야 되는 값이다. 온도센서를 예로 들면 측정할 값의 단위가 섭씨인지 화씨인지 결정하고, 측정 가능한 온도의 범위 등을 설정한다. 출력패턴 정의 단계는 센서에서 출력할 값의 생성 패턴을 정의한다. 게이지조절 또는 값을 직접 입력하여 센서 측정값을 조절할 수 있다. 또한 랜덤 함수 또는 1차 함수와 같이 함수의 결과를 센서 측정값으로 지정할 수 있다.

본 논문에서 설정한 가상센서의 3가지 목표는 다음과 같이 달성될 수 있다. 가능한 모든 통신방식을 지원함으로써 스마트 센서 노드의 측정값을 입력받듯이 가상센서의 측정값을 사용할 수 있다. 출력패턴정의를 통해 센서의 작동환경에 의존하지 않고 센서가 측정 가능한 모든 범위의 값을 생성 할 수 있다. 독립적인 여러 개의 센서들을 센서 선택 단계에서 반복 선택하여 측정값을 동시에 생성 가능하다.

IV. 가상센서 사용 예

본 논문에서는 가상센서의 프로토타입을 제작하였고, 이의 사용 예를 보인다.

1. 센서 선택 단계:

그림 3은 센서를 선택하기 위한 초기화면이다. 그림 3에서 센서들이 나열되어 있는데, 사용할 센서들을 선택하면 다음 단계인 센서 특성화 단계로 넘어간다. 그림 4는 센서 특성화 단계로 센서들이 선택되지 않았을 때는 초기화할 센서가 없음을 보여준다.

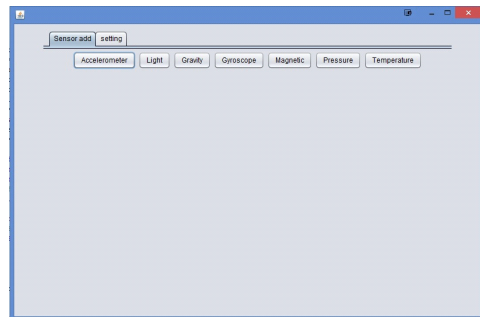


그림 3. 센서 선택 화면
Fig. 3. Screen shot of sensor selection

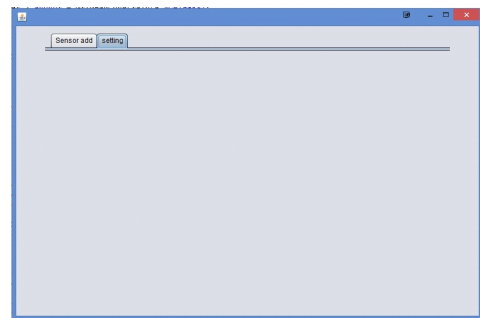


그림 4 초기의 센서 특성화 화면
Fig. 4. Screen shot of Initial phase of sensor characterization

2. 센서 특성화 단계

센서 선택 단계에서 accelerometer, light, gravity, gyroscope 센서를 선택하면, 각각에 대한 특성화 탭이 만들어진다. 그림 5는 accelerometer 센서의 특성화 화면으로 센서 측정값의 단위, 범위, 정확도, 그리고 속도를 사용자가 지정할 수 있다.

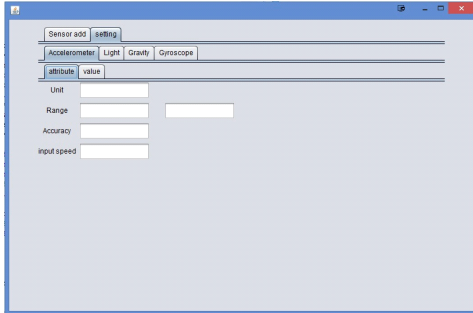


그림 5. accelerometer 센서 특성화 단계
Fig. 5. Screen shot of Characterization of an accelerometer sensor

3. 출력 패턴 정의

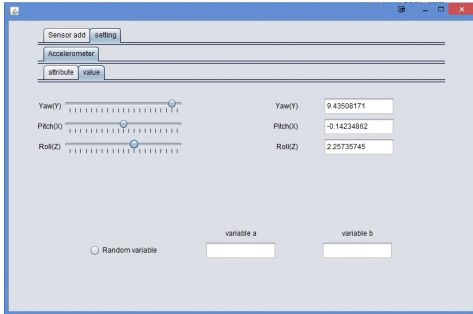


그림 6. accelerometer 센서의 측정값 조절
Fig. 6. Screen shot of Manipulating value of an accelerometer sensor

그림 6은 accelerometer 센서의 value 탭을 보여준다. 이 탭에서 센서 측정값을 사용자가 원하는 대로 조절 가능하다. accelerometer 센서 측정값은 yaw, pitch, roll 과 같이 세 개의 값으로 구성되는데, 세 개의 값을 조절하는 방법으로 스크롤, 값의 직접 입력, 랜덤함수, 1차 함수와 같이 네 가지 방법을 제공한다. 세 개의 스크롤을 각각 이동시켜서 각각의 측정값을 지정할 수 있고, 또한 세 개의 입력 필드에 값을 직접 입력하여 측정값을 지정할 수 있다. 랜덤함수를 선택하면 범위 내의 임의의 값이 측정값이 된다. 1차함수를 선택하고 1차 함수 $ax+b$ 의 기울기 a 와 절편 y 를 입력하면, 범위 내에서 1차 함수의 결과 값으로 측정값이 지정된다.

V. 안드로이드 앱에 적용

대부분의 안드로이드 장치들은 센서들이 내장되어 있다 [19]. 안드로이드4.0에서는 13개의 센서를 지원한다. 안드

로이드 앱은 안드로이드 장치가 지원하는 센서로부터 값을 읽어 올 수 있다. 안드로이드 센서를 이용하는 앱에 대한 테스트에서도 가상 센서를 적용할 수 있다. 안드로이드 앱이 안드로이드에 내장되어 있는 실제 센서의 측정값이 아닌 가상 센서의 값으로 테스트함으로써, 가능한 많은 센서 값에 대해 테스트를 수행할 수 있다.

1. 안드로이드 센서 프레임워크

안드로이드 앱은 안드로이드 센서 프레임워크를 이용하여 센서를 접근하거나 센서 측정값을 읽어올 수 있다. 안드로이드 센서 프레임워크는 android.hardware 패키지에 포함되고, 다음과 같은 세 개의 클래스와 하나의 인터페이스로 구성된다: `SensorManager` 클래스, `Sensor` 클래스, `SensorEvent` 클래스, `SensorEventListener` 인터페이스.

센서 프레임워크가 동작하는 방식은 다음과 같다. 센서 측정값이 변경되면 프레임워크는 이벤트를 발생시킨다. 특정 센서 측정값을 사용하기 위해서는 `SensorEventListener`에서 정의된 콜백 메소드를 구현하고, 리스너를 등록한다. 발생된 이벤트를 기다리는 리스너의 콜백 메소드가 실행된다.

2. 안드로이드 센서 프레임워크의 수정

안드로이드의 센서 프레임워크는 실제 안드로이드에 연결되어 있는 센서의 값이 변경되었을 때 이벤트를 발생시키고, 센서의 측정값을 앱에게 전달한다. 가상 센서를 사용하기 위해서는 가상 센서의 값이 변경되었을 때 이벤트를 발생시키도록 센서 프레임워크를 수정해야한다. 가장 쉽게 생각할 수 있는 센서 프레임워크 수정 방법은 기존의 클래스들을 상속받아 가상 센서의 특성에 맞게 변형하는 것이다. 그러나 `SensorManager` 클래스의 생성자가 `private`로 선언되어 있어 상속할 수가 없었다. 그래서 가상센서를 위한 센서프레임워크를 새로 작성하였다.

3. 안드로이드 앱을 위한 가상 센서 시스템 사용 예

그림 7은 가상센서를 사용한 안드로이드 앱의 실행 결과를 보여준다. 그림 7의 가상센서를 사용한 안드로이드 앱은 안드로이드 센서 프레임워크에서 제공하는 센서 관련 클래스들을 사용하지 않고, 본 연구에서 개발한 가상센서 관련 클래스들을 사용한다. 그림 7에서 가상 센서에서 사용자가 조절한 센서 측정값이 안드로이드 앱으로 전달되어 출력됨을 볼 수 있다.

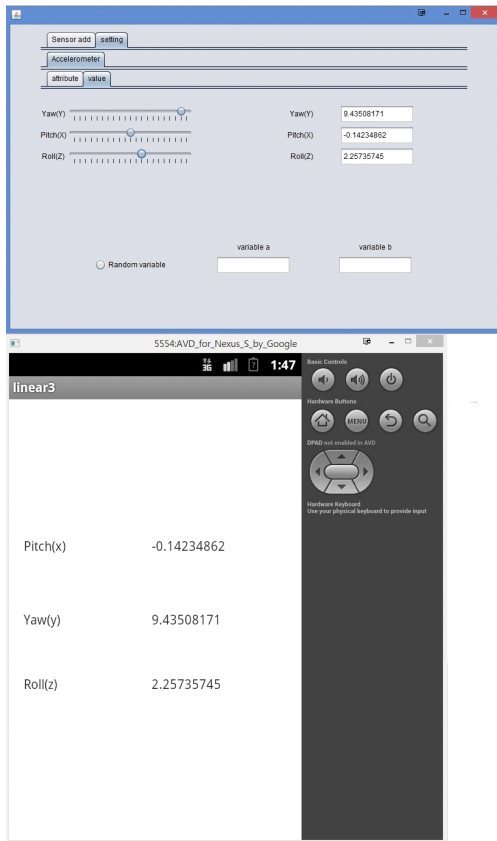


그림 7. 가상 센서를 사용한 안드로이드 앱의 실행
Fig. 7. Screen shot of execution of Android app with virtual sensors

4. 기존 방법과의 비교

DDMS는 위치정보(GPS)를 입력으로 하는 앱에게 가상의 위치정보를 제공한다. 그러므로 안드로이드 기기를 실제 움직이지 않고도 위치정보를 사용하는 앱을 테스트할 수 있다. 그러나 다른 센서들에 대해서는 가상의 센서 값을 사용하는 방법을 제공하지 않는다. 위치 정보는 안드로이드 센서 플랫폼에서 제공하지 않으므로, DDMS가 가상 센서 정보를 제공하기 위해서는 센서 플랫폼을 변경해야 한다.

monkeyrunner는 실제 기기 또는 가상 기기에서 실행되는 앱의 입력을 PC에서 조절할 수 있는 기능을 제공한다. 조절 가능한 입력은 키입력, 터치패널의 누름과 같은 사용자 인터페이스 이벤트이다. 그러나 센서의 입력은 조절할 수 없다.

본 논문에서 제안하는 가상 센서 시스템은 실행 시스템의 외부인 PC에서 센서 값 조절이 가능하다는 것이다. 이는 DDMS와 monkeyrunner의 장점들을 취한 것으로 볼 수 있

는데, DDMS의 장점인 위치정보의 가상화를 센서의 가상화에 적용하였고, monkeyrunner의 장점인 외부에서 입력 값을 조절하는 기능을 센서의 입력에 적용하였다.

IV. 결론

본 논문에서 스마트 센서 응용 소프트웨어를 테스트하기 위한 기존의 방법을 소개하고 문제점을 보였다. 그리고, 문제점을 보완하기 위한 효율적인 테스트 방법을 제안하였다. 기존의 방법에서는 실제 센서의 측정값을 이용하여 응용소프트웨어를 테스트함으로써 테스트 데이터가 제한적인 문제가 있었다. 본 논문에서 센서 측정값을 조절 가능한 가상 센서를 제안함으로써 이 문제의 해결책을 제안하였다.

현재 소프트웨어 테스트 분야에서는 테스트 케이스 자동생성 뿐만 아니라, 테스트 자동화도 실용적으로 사용되고 있다. 센서 응용 소프트웨어에 대해서도 가상 센서 개념을 도입하면 테스트의 자동화 기법들을 도입할 수 있을 것으로 생각된다.

향후 연구로는 본 논문에서 제안한 가상 센서 시스템을 자동차, 원자력 등 센서를 사용하는 많은 분야에 적용해 보는 것이다.

참고문헌

- [1] Namki Min, "Introduction to Sensor Technology", Dongil press, pp 11-14, 2013
- [2] A. Feng, et. al, "Embedded system for sensor communication and security", IET Information Security, Vol. 6, No. 2, pp 111-121, Jun. 2012.
- [3] H. Ramamurthy, et. al, "Wireless Industrial Monitoring and Control Using a Smart Sensor Platform", IEEE SENSORS JOURNAL, Vol. 7, No. 5, pp 611-618, May. 2007.
- [4] M. H. Salah, et. al, "A smart multiple-loop automotive cooling system - model, control, and experimental study", IEEE/ASME Trans. Mechatronics, Vol. 15, No. 1, pp 117-124, Jan. 2010.
- [5] T. Bein and D. Mayer, "Smart Sensor Networks for Structural Health Monitoring", Advanced Microsystems for Automotive Applications 2013, Springer International Publishing, pp 385-394,

- 2013.
- [6] M. E. Cater, T. O'Reilly, "Promoting interoperable ocean sensors the smart ocean sensors consortium", Proc. OCEANS 2009, MTS/IEEE Biloxi - Marine Technology for Our Future, pp 1-6, Oct. 2009.
- [7] L. Ghelardoni, et. al., "Smart underwater wireless sensor networks", 2012 IEEE 27th Convention of Electrical & Electronics Engineers in Israel, pp 1-5, Nov. 2012.
- [8] S. H. Choo and H. S. Seo, "Sensor-based Alert System applying Expert System for Performance Improvement", Journal of the Korea Society of Computer and Information, Vol. 17, No. 10, pp 1-9, Nov. 2012.
- [9] M. Rusu, et. al., "Distributed e-health system with smart self-care units", Proc. IEEE Fifth Int. Conf. on Intelligent Computer Communication and Processing, pp 307-314, Aug. 2009.
- [10] M. M. Baig and H. Gholamhosseini, "Smart Health Monitoring Systems: An Overview of Design and Modeling", Journal of Medical Systems, Vol. 37, (on-line) Jan. 2013.
- [11] S. Y. Shin and Y. W. Lee, "Enhancement of Sleep Environment Using Sensor and User Information", Journal of the Korea Society of Computer and Information, Vol. 16, No. 1, pp 47-52, Jan. 2011.
- [12] Soonil Cha, "Status and prospective of SW testing industry", Communications of KIISE, Vol. 28, No. 11, pp 76-84, Nov. 2010.
- [13] C. Kameshwaran, "Software Testing", Prentice-Hall, pp 5-8, 2012.
- [14] P. McMinn, "Search-based software test data generation: a survey", Software Testing, Verification and Reliability, Vol. 14, No. 2, pp 105-156, May. 2004.
- [15] Bixin Li, et. al., "Automatic test case selection for regression testing of composite service based on extensible BPEL flow graph", Journal of Systems and Software, Vol. 85, No. 6, pp 1300-1324, Jun. 2012.
- [16] Sangwoon Kim et. al., "Survey of System testing environment of embedded SW and testing technology", Communications of KIISE, Vol. 31, No. 5, pp 63-72, May. 2013.
- [17] DDMS, <http://developer.android.com/tools/debugging/ddms.html>
- [18] monkeyrunner, http://developer.android.com/tools/help/monkeyrunner_concepts.html
- [19] Overview of Android Sensors, http://developer.android.com/guide/topics/sensors/sensors_overview.html

저자 소개



조 장 우

1992: 서울대학교
계산통계학과 이학사.
1994: 서울대학교
계산통계학과 이학석사.
2003: KAIST
전산학과 공학박사
현 재: 동아대학교
컴퓨터공학과 교수
관심분야: 프로그램 분석,
임베디드 시스템
Email : jwjo@dau.ac.kr



정 환 철

2013: 동아대학교
컴퓨터공학과 공학사.
현 재: 동아대학교
컴퓨터공학과 석사과정.
관심분야: 안드로이드 개발환경,
센서 SW
Email : karnersw@gmail.com