

레고 마인드스톰 NXT 를 위한 센서 API 개선 사례

정중현, 박소현, 권기현
 경기대학교 컴퓨터과학과
 {jhcheong, bsh2523, khkwon}@kyonggi.ac.kr

Empirical Study of Improving Sensor API for Lego Mindstorms NXT

Jong-Hyun Cheong, So-Hyun Park, Gihwon Kwon
 Dept. of Computer Science, Kyonggi University

요 약

레고 마인드스톰 NXT 는 다른 로봇 키트에 비해서 조작이 간편하며, 고차원적인 로봇 행위를 제어하기 위해 펌웨어인 leJOS 를 지원하며, 블루투스를 이용한 원격 제어가 가능하다. 그러나 leJOS 는 외부 파일을 제어하거나 센서 하드웨어가 정상 동작하지 않는 등의 몇 가지 문제점이 발견되었다. 본 논문에서는 leJOS 가 갖는 이러한 일련의 문제점을 살펴보고, 이에 대한 해결책을 제시한다.

1. 서론

오늘날의 로봇은 인간이 해오던 많은 일을 대신하고 있다. 로봇은 사람과 유사한 모습과 기능을 가진 기계, 또는 무엇인가 스스로 작업하는 능력을 가진 기계를 말하며, 이러한 로봇을 제어하기 위해서는 소프트웨어가 필요하다.

로봇을 제어하는 프로그램을 구현하고 검증하기 위하여 우리는 여러 로봇 하드웨어를 조사하였다. 그 중 레고 마인드스톰 NXT 는 어린이도 쉽게 다룰 수 있으며, 고차원적인 로봇의 동작을 구현하기 위한 다양한 프로그래밍 언어를 지원한다.

레고 마인드스톰 NXT 패키지는 여러 가지의 부품으로 구성되어 있으며, 사용자에게 의해 구현된 로봇 프로그램을 적재하고 명령을 수행할 수 있는 작은 컴퓨터인 NXT 지능형 브릭(Brick)이 포함 되어있다. 우리는 JAVA 를 이용하여 NXT 지능형 브릭을 제어하기 위해, 펌웨어를 leJOS(Lego JAVA Operating System)로 교체하였고 로봇 프로그램에 대한 연구를 시작하였다.

이 과정에서 우리는 leJOS 가 지원하는 JAVA API (Application Programming Interface)의 몇 가지 문제를 발견할 수 있었다. 첫째는 NXJ(NXT Java) API 에서 외부 파일을 제어하거나 제 3 자 라이브러리를 사용할 수 없다는 것, 둘째로 PC API 에서 일부 센서 하드웨어가 정상 동작하지 않는다는 문제다.

본 논문은 이러한 문제에 대한 해결방안을 제안하며 구성은 다음과 같다. 먼저 2 장에서는 논문을 이해하기 위한 배경지식으로 NXT 와 leJOS 에 대하여 간략히 소개하며, 3 장에서는 도출된 문제점에 대하여 기술한다. 4 장에서는 3 장에서 발견된 문제를 해결하기 위한 방법을 제안하며, 5 장에서는 4 장에서 제안한

방법을 이용한 사례를 보인다. 끝으로 6 장에서는 결론 및 향후 연구를 기술한다.

2. 배경 지식

2.1 레고 마인드스톰 NXT 지능형 브릭

1958 년 덴마크의 레고가 개발한 레고 블록은 어린 이들이 이용 가능한 장난감이다. 레고는 미국 MIT 대학에 연구를 지원, 사용자의 명령에 따라 움직일 수 있도록 프로그램 가능한 로봇 개념을 적용한 레고 마인드스톰 RIS 키트를 개발하였다. 이 키트에는 사용자의 명령을 수행할 수 있는 지능형 브릭이 포함되어 있으며 현재 가장 최신의 것이 NXT 지능형 브릭이다.

<표 1> NXT 지능형 브릭 제원

NXT 지능형 브릭 제원	
항목	값
프로세서	32-bit ARM7
메모리	64kb
저장장치	256kb 플래시 메모리
확장 인터페이스	RJ12 (모터 3 개/센서 4 개) USB Bluetooth
입력장치	버튼(4 개)
출력장치	100x60 pixel 흑백 LCD 8kHz 스피커

표 1 에서 보는 것과 같이 NXT 지능형 브릭의 성능은 오늘날의 PC 의 성능과 비교하여 볼 때 매우 제한적이다. 하지만, 이러한 환경에 맞춰 개발된 leJOS 펌웨어는 NXT 지능형 브릭을 이용하여 조립된 로봇을 쉽게 제어할 수 있도록 도와준다.

□ 본 연구는 지식 경제부의 정보통신 산업 진흥원사업의 일환으로 수행하였음. [2012-0043, 컴퓨터과학 전문프로그램]

2.2 leJOS 펌웨어

NXT 지능형 브릭은 다양한 프로그래밍 언어를 지원하는데 우리는 JAVA 언어를 사용하기 위해 leJOS 펌웨어를 사용하였다. leJOS 는 JAVA 를 이용해 NXT 지능형 브릭을 제어하는 펌웨어이다. 또한 leJOS 는 통합 개발 환경인 이클립스나 넷빈즈 같은 개발환경과 연동 할 수 있는 플러그인을 지원한다.

leJOS 는 NXT 지능형 브릭을 제어하는 두 가지 방법을 제공한다. 하나는 프로그램의 실행 주체가 NXT 지능형 브릭이 되는 방법이다. 이 방법은 leJOS 의 NXJ API 를 이용해 프로그램을 작성 및 컴파일 하고, 결과를 NXT 지능형 브릭에 적재하여 실행한다. 다른 방법으로는 프로그램의 실행 주체가 PC 가 되는 것이다. 이 방법은 leJOS 의 PC API 를 이용해 프로그램을 작성 및 컴파일 하고, 결과를 PC 에서 실행한다. 이 때 leJOS 는 내부적으로 블루투스를 통해 NXT 지능형 브릭과 연결하여, PC 와 하드웨어 제어 신호를 교환한다. 각 제어 방법을 사용하기 위해 필요한 API 의 장 단점은 다음과 같다.

- ① NXJ API: 센서 하드웨어에 구동에 대한 제약이 없지만, 파일 제어 API 의 지원이 미약하고, 제 3 자 라이브러리 사용에 제한이 있다.
- ② PC API: PC 의 JAVA 가 지원하는 모든 기능을 사용할 수 있고, 따라서 파일 제어, 제 3 자 라이브러리 사용이 용이하다. 하지만 센서 하드웨어 구동에 일부 제약이 있다.

3. leJOS 사용에 대한 문제 확인

앞서 1 장에서 leJOS 를 사용하여 로봇을 제어하는데 있어 문제점을 이야기하였다. 본 논문의 실험환경은 아래 표 2 와 같다.

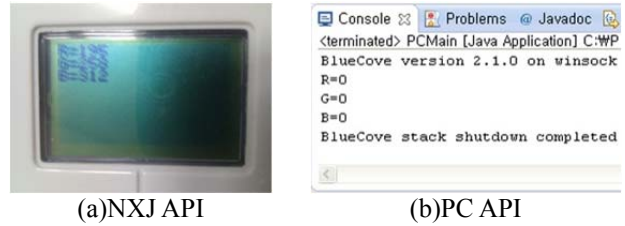
<표 2> 실험 환경

테스트 환경	
항목	값
운영체제	윈도우 XP SP3(32bit) 윈도우 7 SP1(32/64bit)
JAVA	V1.7.0_07
통합 개발 환경	Eclipse 4.2
NXT firmware	leJOS v0.9.1 beta3

leJOS 의 PC API 를 사용하던 도중 센서들의 하드웨어 값이 기대했던 값과 다르다는 점을 발견 할 수 있었다. 여러 센서 중 컬러 센서는 우리의 실험 환경에서 로봇의 상태를 결정하는 중요한 역할을 하지만, 이 값은 항상 검은색을 가리키고 있었다. 문제를 정확히 확인하기 위하여 우리는 센서 하드웨어 기능만을 추출하여 NXJ API 를 이용한 별도의 프로그램을 구현하고 실험 한 결과 PC API 에서는 일부 센서 하드웨어가 정상 동작하지 않음을 확인할 수 있었다.

우리는 문제점을 보이기 위해 NXJ API 와 PC API 에 동일한 코드를 작성 및 적용 하였고, 그 결과 API 를 제외한 다른 환경이 모두 동일하지만 아래 그림 1 과

같이, (a) 는 정상적인 값을 얻어오고 (b) 는 정상적인 값을 얻지 못하는 문제를 재현 하였다.



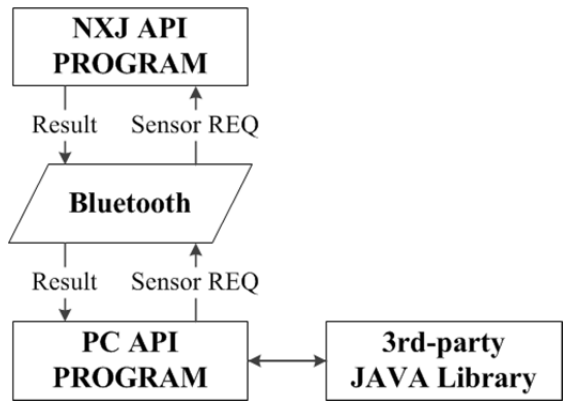
(그림 1) 동일 환경에서 API 가 동작 하는 값

이러한 문제가 발생한 원인은 leJOS 의 PC API 내부에서 찾을 수 있었다. 센서에서 값을 읽어오는 매소드는 구현되지 않았고 주석을 통해 ‘블루투스를 이용한 다중 값 전송은 아직 진행 단계이기 때문에, 이 기능은 나중에 사용할 것이다’ 라고 밝히고 있었다.

우리는 이 문제에 대한 해결책을 연구한 결과 다음 4 장에서 제안하는 센서 하드웨어 문제점을 해결할 수 있었다.

4. 센서 하드웨어 문제 해결 방안

일반적으로 NXT 로봇 프로그램 구현에는 동작 방식을 결정, 구현에 필요한 API 를 사용해 하나의 구현 프로젝트를 진행한다. 하지만 앞서 2 장 2 절에서 밝힌 것처럼 NXJ API 와 PC API 는 상호 보완적인 장/단점을 가지고 있으며, 각각의 API 는 Bluetooth 를 통해서 기종간의 통신이 가능하도록 지원하고 있다.

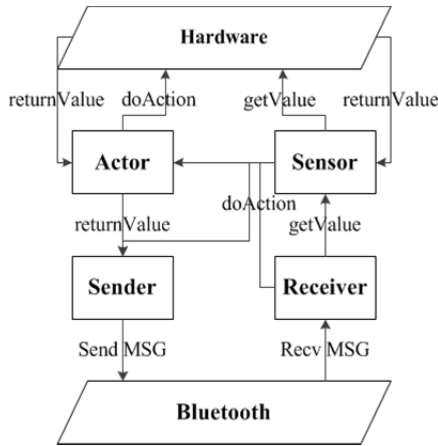


(그림 2) 문제 해결 방안 개요도

우리가 제안하는 방식은 그림 3 과 같이 상호 보완적인 장/단점을 지닌 leJOS 의 API 들을 블루투스로 연결하여 단점을 보완하고 장점을 부각시키는 것이다. 프로그램의 주 기능은 모두 PC API 에 존재하며, 센서 하드웨어와 같은 기능적 부재를 NXJ API 로 작성한 프로그램이 담당한다는 것이 핵심이다.

4.1. NXJ API 로 작성한 프로그램

leJOS 에서 지원하는 통신 API 를 이용하여 우리는 그림 3 과 같이 구현하였다.



(그림 3) NXJ API 동작 흐름도

프로그램은 다음과 같은 동작 흐름을 가진다.

- ① Receiver 에서 블루투스를 이용해 메시지를 수신하여, 명령이 수행되어야 하는 클래스에 해당 명령 값을 보낸다.
- ② 각 하드웨어 명령 수행 클래스는 명령 값을 토대로 지정된 작업을 수행하고 결과값을 돌려준다.
- ③ Sender 에 도착한 결과는 메시지가 반환 값을 요구 할 경우 블루투스를 이용해 값을 돌려주며, 그렇지 않은 경우 아무 작업도 하지 않는다.

4.2. 통신 메시지 형식 정의

4 장 1 절의 NXJ API 로 작성된 프로그램에 명령을 내리기 위해서는 블루투스로부터 명령 메시지가 도착해야 한다. 여기서 우리는 메시지의 구조를 정의하였다. 우선적으로 명령 메시지에 포함되어야 하는 요소들에 대해 논의를 거쳐 다음과 같은 요소를 정의 할 수 있었다.

- ① NXJ API 가 어떤 명령을 동작시켜야 하는가?
- ② 명령을 동작하기 위해 파라메타가 필요한가?
 - 파라메타의 자료 형은 무엇인가?
 - 파라메타의 값은 무엇인가?
- ③ 동작한 명령의 반환 값이 필요한가?

위와 같은 요소로부터 아래 그림 4 와 같은 메시지 구조를 정의 할 수 있다.

PC -> NXT Message

PACKET ID	PACKET SIZE	Method ID	Parameter Type	Parameter Value	IS RETURN
-----------	-------------	-----------	----------------	-----------------	-----------

파라메타가 여러 개일 경우, 이 묶음 원소는 반복 추가된다.

NXT->PC Message

PACKET ID	PACKET SIZE	Return Type	Return Value
-----------	-------------	-------------	--------------

반환 값이 여러 개일 경우, 이 묶음 원소는 반복 추가된다.

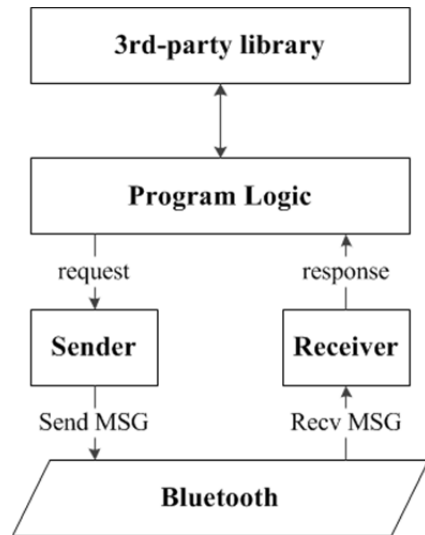
(그림 4) 송/수신 통신 메시지 형식 정의

메시지 구조를 그림 4 와 같이 정의한 가장 주된 아이디어는 통신 환경에 전달되는 하나의 명령 메시지를 패킷으로 본다는 것이다. 따라서 메시지 포함 요소 외에 명령 메시지를 식별할 수 있는 Packet ID 와 명령 메시지의 크기를 담은 Packet Size 가 메시지 구조에 추가되었다.

4.3. PC API 로 작성한 프로그램

아래 그림 5 는 PC API 로 작성된 프로그램의 흐름이다. 본 논문의 3 장 그림 1 에서 본 것과 같이 NXJ API 는 센서 하드웨어의 동작에 문제가 없다. 따라서 PC API 로 작성한 프로그램은 센서 하드웨어를 동작시키기 위해 NXJ API 를 빌려 쓰는 방식을 써야 한다.

따라서 4 장 1 절에서 작성된 NXJ API 프로그램에 블루투스로 명령 메시지를 보내 결과값을 받는 방식으로 PC API 프로그램을 구현하였다. 이 방법을 통해 프로그램은 센서의 값을 정상적으로 읽을 수 있다. 또한 PC API 는 제 3 자 라이브러리 사용에 제약이 없기 때문에 우리가 실험하고자 하는 환경의 요구사항을 충족시킬 수 있었다.



(그림 5) PC API 프로그램 동작 흐름도

5. 센서 하드웨어 문제 해결 및 외부 라이브러리 적용

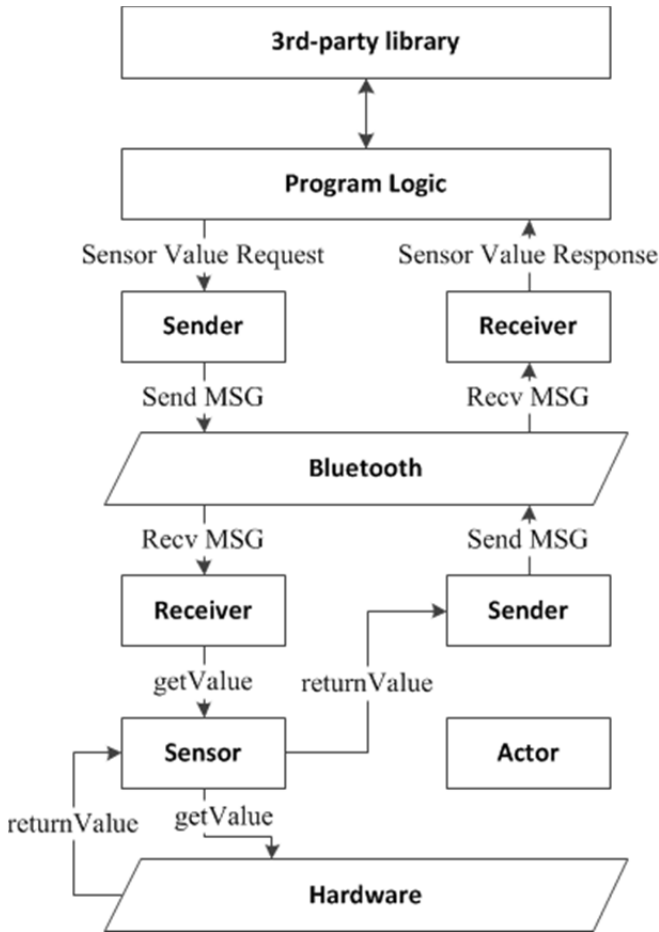
센서 하드웨어 문제 개선 방법을 사용하여 우리는 다음과 같은 시나리오를 수행하였다.

- 시나리오 1: 로봇은 정지해 있다. 로봇의 센서에 붉은 색이 인식되면 로봇은 비프음을 울린다.
- 시나리오 2: 로봇은 지정된 폐쇄공간 내에서 시계방향으로 움직인다. 폐쇄공간은 4 개의 구역으로 나뉘어 있으며, 각 구역은 바닥 면의 색으로 구분되어 있다. 로봇은 바닥 면의 색을 인식하여 현재 위치를 파악한다.

위의 시나리오를 PC API 프로그램으로 구현할 경우,

우리가 제안하는 개선된 센서 하드웨어 시스템이 없다면 구현하기가 어렵지만 우리가 제안하는 시스템을 이용할 경우, 이러한 문제를 우회할 수 있었다.

그림 6 은 위의 시나리오에서 센서 하드웨어의 결과 값이 요구되는 경우, 프로그램의 흐름도를 나타낸다.



(그림 6) 센서를 동작하기 위한 상호작용

- ① 제 3 자 라이브러리와 상호작용하는 PC API 프로그램의 로직이 동작하는 과정에서 칼라 센서의 값이 필요하다.
- ② PC API 프로그램 로직은 PC API 의 Sender 를 통해 NXJ API 프로그램에 명령 메시지를 전달한다.
- ③ NXJ API 의 Receiver 는 수신한 메시지를 해당 센서 클래스에 전달한다.
- ④ 센서 클래스는 하드웨어를 동작시켜 명령을 수행하고, 결과값을 NXJ API 의 Sender 로 전달한다.
- ⑤ NXJ API 의 Sender 는 값을 반환 여부를 판단하고, Bluetooth 를 이용해 반환 메시지를 전달한다.
- ⑥ PC API 의 Receiver 는 이 반환 메시지를 받아 프로그램 로직에 전달한다.

이러한 일련의 과정을 거쳐 그림 7 에서 보듯이 PC

API 가 올바른 칼라 센서 값을 획득할 수 있었다.

```

    Console Problems Javadoc Declaration
    <terminated> PCMain (1) [Java Application] C:\Program Files
    BlueCove version 2.1.0 on winsock
    getColor
    R=317 G=115 B=132
    getColor
    R=193 G=332 B=151
    getColor
    R=139 G=127 B=321
    BlueCove stack shutdown completed
  
```

(a) 시나리오 1

```

    Console Problems Javadoc Declaration
    PCMain (1) [Java Application] C:\Program Files (x86)\Java\jre7\bin\javaw.exe (2C
    NXT driven
    getRegion
    current:Region1 R=314 G=121 B=107
    getRegion
    current:Region1 R=303 G=99 B=157
    getRegion
    current:Region2 R=146 G=138 B=340
  
```

(b) 시나리오 2

(그림 7) 정상 동작하는 칼라 센서 값

6. 결론

PC API 에서 일부 센서 하드웨어 지원 부재는 NXT 와 leJOS 를 사용하는 사람들에게 불편을 주고 있다. eJOS 는 아직 개발 단계에 있는 펌웨어로 이러한 기능 부재는 지속적인 버전업을 통해 해결 될 것이다.

그러나 현재로서는 이러한 문제를 해결 할 수 있는 방법이 매우 제한적인 것이 한 사실이다. 본 논문에서는 레고 마인드 스톰 NXT 를 제어할 하는데 있어 아직 지원되지 않은 센서 하드웨어 기능을 NXT API 와 PC API 에서 지원하는 블루투스 통신 기술을 이용하여 해결하였다.

향후 연구로, 우리는 개선된 센서 하드웨어 시스템을 이용하여 외부 라이브러리와 연계한 고차원적인 로봇 시스템을 구현할 것이다.

참고문헌

- [1] <http://mindstorms.lego.com/en-us/Default.aspx>
- [2] <http://lejos.sourceforge.net/>
- [3] Brian Bagnall "Maximum LEHO NXT Building Robots with Java Brains", 2007
- [4] 박승성, 김종원, 이경일, 권순량, "블루투스를 이용한 홈 네트워크 서비스 설계", 한국정보과학회 가을 학술 발표논문집 Vol. 30. No.2
- [5] Sivan Toledo, "Analysis of the NXT Bluetooth Communication Protocol", 2006