

2013년 한국정보처리학회 제40회 추계학술발표대회 프로그램

The 40th Conference of the KIPS

일 자 : **2013년 11월 8일(금) ~ 9일(토)**

장 소 : **제주한라대학교**

주 최 : 사단법인 **한국정보처리학회**

주 관 :  **제주한라대학교**

제주한라대학교 정보기술교육원

협 찬 : 한국과학기술정보연구원(GSDC), 대보정보통신, 삼성SDS, SK텔레콤,
롯데정보통신, 새로운 교육, 한국IT컨설팅, 네이버, 메타빌드,
비트컴퓨터, 빅센시스템즈, 세방SDL, 알폰스테크, 엠프론티어,
원스테크넷, 이노덱, 이소프팅, 중앙정보처리학원, 토마토시스템,
한국생산성본부, 화이트정보통신, 미래큐브, 포씨

294. HTML5를 이용한 효율적 전자책 편집기와 검색기 구현에 관한 연구 KIPS_C2013J_0245
박종현*, 강현민, 황규철, 이경오(선문대학교) • 973
295. 스마트폰 기반 게임화 전략의 실시간 질의응답 시스템 KIPS_C2013J_0255
유도준*, 박현우, 권순각, 이중화(동덕대학교) • 977
296. SNS 기반의 능동적 디펜스 게임 개발 KIPS_C2013J_0261
송문철, 김창완, 박기남*(순천향대학교) • 980
297. 학습자의 인지능력 기반 개인화 학습 프로파일 설계 KIPS_C2013J_0262
지혜성, 임희석(고려대학교), 박기남*(순천향대학교) • 983
298. 구글 클라우드 메세징 서비스를 활용한 영단어 암기용 스마트폰 어플리케이션 개발
 KIPS_C2013J_0266
김기훈*, 임동현, 박창우, 전미영, 정구민(국민대학교) • 986
299. 해외사례를 통한 소프트웨어산업 발전방안 연구 KIPS_C2013J_0282
조성갑*(인천정보산업진흥원) • 989
300. 연속공정 가상설비 모델링 시스템을 위한 이산사건 알고리즘 적용에 관한 연구
 KIPS_C2013J_0289
송호섭*(POSCO ICT) • 992
301. Green500 분석을 통한 전력효율 측면에서의 슈퍼컴퓨터 개발 동향 KIPS_C2013J_0323
전인호*, 남덕윤, 류훈, 이준형, 이종숙(한국과학기술정보연구원) • 995
302. 소상공인 공동협업체를 위한 제조사별 제품정보 공유기반 협업 서비스 설계
 KIPS_C2013J_0337
홍성웅*((주)유시스템), 조여일, 김평중(충북도립대학교), 한태성((주)뉴한시스) • 998
303. 동작과 음성을 인식하는 인터페이스 기반 PC 런처 구현 KIPS_C2013J_0339
황선명*, 염희균, 김범식, 박성주, 임홍택, 강진원, 김정섭(대전대학교) • 1001
304. 성공적인 국방 프로젝트 수행을 위한 RFP 적합성에 관한 연구 KIPS_C2013J_0349
문선희*, 이석주(고려대학교) • 1004
-  305. 스마트폰과 OPRoS를 이용한 DARwin-OP 로봇 제어 어플리케이션 개발에 대한 연구
 KIPS_C2013J_0360
이주영*, 이경수, 정구민(국민대학교) • 1008
306. 개선된 Fillmore Case Grammar를 통한 Goal Use-Case 식별 방법 KIPS_C2013J_0361
박보경*, 양효석, 김영철(홍익대학교) • 1011
307. Use-Case Approach 기반의 Test Case 추출을 위한 아두이노 펌웨어 설계 연구
 KIPS_C2013J_0364
장우성*(한백전자), 박병호, 김영철(홍익대학교), 박용범(단국대학교) • 1015
308. 블랙박스 테스트를 위한 함수 추상화 기반의 테스트 케이스 생성기법 KIPS_C2013J_0416
류호동*, 이우진(경북대학교) • 1019
309. HTML5기반의 실시간 서버 및 데이터베이스 언어 KIPS_C2013J_0427
유환수*, 김성환(서울시립대학교) • 1022
310. 스프링 프레임워크를 기반으로 POJO 프로그래밍 가능기술의 구현 및 평가 KIPS_C2013J_0455
권기홍*, 변정용(동국대학교) • 1026
311. 감성 어휘 기반 인재검색을 위한 이미지 스케일과 감성 어휘 분류 체계 KIPS_C2013J_0491
김용우*, 박석천(가천대학교), 홍석우, 김태엽(화이트정보통신) • 1030
312. 조직 커뮤니케이션 관계에서 감정노동 개선을 위한 소시오그램 시각화 설계
 KIPS_C2013J_0492
김용우*, 박석천(가천대학교), 홍석우, 김태엽(화이트정보통신) • 1034
313. 모바일 웹 기반의 면접 환경 개선을 위한 감성어휘 분석 모형 설계 KIPS_C2013J_0493
김용우*, 박석천(가천대학교), 홍석우, 김태엽(화이트정보통신) • 1038



소프트웨어공학



Use-Case Approach 기반의 Test Case 추출을 위한 아두이노 펌웨어 설계 연구

장우성*, 박병호**, 김영철**, 박용범***
*한백전자

**홍익대학교 소프트웨어공학 연구실

***단국대학교

e-mail:wsjang@hanback.co.kr*, {sunsonbob,bob}@hongik.ac.kr**,
ybpark@dankook.ac.kr***

A Study on Arduino Firmware Design for Test Case Extraction based on Use-Case Approach

Woo-Sung Jang*, Byung-Ho Park**, R.Young-Chul Kim**, Y.B.Park***

*Hanback Electronics

**Dept. of Software Engineering, Hong-ik University

***Dankook University

요 약

최근 임베디드 펌웨어를 객체 지향 어플리케이션과 함께 개발하는 경우가 증가하고 있다. 하지만 기존의 펌웨어는 절차적 설계를 기반으로 하기 때문에 어플리케이션과 다른 설계 기법이 필요하여 개발 비용과 테스트 비용이 증가한다. 그렇지만 아두이노를 사용하면 펌웨어를 객체 지향 설계가 가능하다. 본 논문은 객체지향 적용 펌웨어를 설계에 초점을 두고, Gary E.Mogyorodi의 Test Case 생성 방법을 적용하여, 실제 펌웨어를 시험을 하고자 한다. 이 방법이 가능하면 일반적인 펌웨어와 어플리케이션을 같은 방법으로 설계 및 테스트가 가능하다. 적용사례로써, 한백전자 HBE-SmartCAR의 펌웨어를 아두이노 기반으로 설계 및 추출된 Test Case 실행을 통해, 설계와 구현이 동일함을 검증이 가능하다.

1. 서론

소프트웨어의 개발 비용을 줄이기 위해서는 유지보수 비용을 줄여야 한다. Boehm에 의하면 유지보수 비용이 컴퓨터 시스템의 모든 비용 중 큰 비중을 차지하고 있고, 앞으로 계속적인 유지보수 비용의 증가로 인하여 개발보다는 유지보수 비용에 많은 돈이 투자될 것이다[1]. 이 유지보수 비용을 줄이기 위해서는 체계적인 소프트웨어 테스트가 필요하다. 소프트웨어 테스트는 소프트웨어의 품질과 신뢰성 그리고 유지보수성을 향상시키는데 목적이 있다[2].

체계적인 소프트웨어 테스트는 정확한 요구사항으로부터 Test Case를 추출하여 개발 후 시스템 검증 테스트가 필요하다. 객체지향 및 Use-Case 개발환경에서 State Machine을 통하여 Test Case 발생방법, Use-Case Diagram을 통한 Test Case 발생 방법 등이 있다[3].

최근 임베디드 소프트웨어는 단순히 펌웨어만 개발하지 않고, 펌웨어와 어플리케이션을 함께 개발하는 경우가 증가하고 있다. 스마트폰이 대중화 됨에 따라 펌웨어와 스마트폰 어플리케이션이 통신을 하면서 장비를 제어하는 시스템이 증가하고 있기 때문이다. 자동차, 보안 감시, 로봇 제어 등을 예로 들 수 있다.

기존 펌웨어는 절차식 개발 방법을 사용하고, 어플리

케이션은 주로 객체 지향 개발 방법을 사용하여 개발된다. 두 언어는 설계 의도와 사용 방법이 상이하기 때문에 같은 설계 기법과 Test Case를 적용하는데 어려움이 있다. 각 개발 방법에 맞게 테스트를 진행된다면 유지보수 비용이 증가하게 된다.

본 논문은 펌웨어와 어플리케이션을 객체 지향 개발 방법으로 통일하여 설계하고 테스트하여 시스템의 유지보수 비용을 줄이는데 목적을 둔다. 펌웨어는 아두이노 기반의 개발 환경을 사용한다. 아두이노 환경은 객체 지향 기반 언어를 사용하여 펌웨어를 제작하기 때문에 객체지향 기반 테스트를 쉽게 할 수 있는 장점이 있다. 그리고 설계도가 오픈되어 있기 때문에 쉽게 양산이 가능하다. 펌웨어와 어플리케이션을 하나의 Use-Case Diagram으로 설계하고, Gary E.Mogyorodi가 제안한 검증된 방법을 사용하여 Test Case를 추출한다[4].

2장은 관련 연구를 언급한다. 3장은 아두이노 기반 펌웨어와 안드로이드 앱을 Use-Case 다이어그램으로 설계한다. 4장은 설계한 Use-Case 다이어그램을 Gary E.Mogyorodi의 방법을 사용하여 테스트 케이스를 생성한다. 5장은 결론 및 향후 연구를 언급한다.

2. 관련 연구

2.1. 기존 Test Case 추출 절차

아래의 그림 1은 본 논문에 적용된 Test Case추출 절차이다. 절차로는 Use-Case 개발 기반으로, Use Case Diagram으로 설계된 시스템을 Gary E. Mogyorodi의 Cause-Effect Graphing 기술을 통해 Test Case를 추출한다.

Cause-Effect Graphing 기술은 최소의 Test Case로 100%의 기능적인 요구사항 커버리지를 만족시킬 수 있는 검증된 방법이다[3]. 하지만 구조적 설계를 기반으로 하기 때문에 객체지향 개념인 Use-Case Diagram을 적용하기 위해서는 Use-Case의 결정요소를 추출하는 과정이 필요하다[5].



(그림 1) Test Case 추출 프로세스

2.2. 아두이노

아두이노(Arduino)는 오픈소스를 기반으로 한 단일 보드 마이크로 컨트롤러이다. AVR을 기반으로 한 보드로 이루어졌다. 최근에는 Cortex-M3를 이용한 제품(Arduino Due)과 소프트웨어 개발을 위한 통합 환경(IDE)이 있다. 아두이노는 다수의 스위치나 센서로부터 값을 받아들여, LED나 모터와 같은 외부 전자 장치들을 통제함으로써 환경과 상호작용이 가능한 제품을 만든다. 또한 플래시, 프로세싱, Max/MSP와 같은 소프트웨어를 연동 가능하다[6].

아두이노의 가장 큰 장점은 마이크로컨트롤러를 쉽게 동작시킨다. 일반적으로 AVR 프로그래밍이 WinAVR로 컴파일하여, ISP장치를 통해 업로드 하는 번거로운 과정이 있다. 아두이노는 컴파일 된 펌웨어를 USB를 통해 쉽게 업로드 할 수 있다. 또한 다른 모듈에 비해 비교적 저렴하고, 윈도를 비롯해 맥 OS X, 리눅스와 같은 OS를 모두 지원한다. 아두이노 보드의 회로도가 CCL에 따라 공개되어 누구나 직접 보드를 제작 및 수정이 가능하다.

아두이노가 인기를 끌면서 자동차회사 포드는 아두이노를 이용해 차량용 하드웨어와 소프트웨어를 만들어 차량과 상호작용을 할 수 있는 오픈XC라는 프로그램을 선보이기도 했다[7].

2.3. HBE-SmartCAR

본 논문의 HBE-SmartCAR는 마이크로프로세서를 이용한 교육용 바퀴형 이동 로봇으로써, 적용사례에 사용된다. 기존의 고정된 환경에서 실습하는 범위를 벗어나 모터의 구동을 통해 장비 이동과 실험 지원이 가능하다. 따라서 장비에서는 컨트롤 디바이스(MCU)를 통해 현재 상태를 파악한 후 동작을 실험할 수 있다 .

내부에 컨트롤 디바이스로는 ATmega2560로 제어한다. 이 디바이스(MCU)가 장비에 장착된 센서를 통해 현

재 상태를 파악과 DC 모터를 제어하여 장비를 구동한다[8]. 그림 2는 HBE-SmartCAR의 외형과 같다.

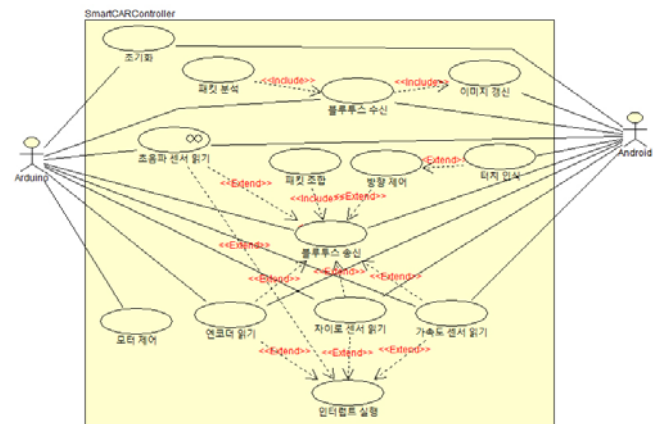


(그림 2) HBE-SmartCAR

3. 소프트웨어 설계

홍익대학교의 HiMEM 도구를 사용하여 SmartController 펌웨어와 어플리케이션을 Use-Case Diagram으로 표현하였다. 그림 3은 SmartController의 설계이다. Actor는 Arduino와 Android이며, 이는 아두이노 펌웨어와 안드로이드 어플리케이션을 의미한다. Use Case는 초기화, 패키지 분석, 블루투스 수신, 블루투스 송신, 이미지 갱신, 패키지 갱신, 방향 제어, 터치 인식, 초음파 센서 읽기, 패키지 조합, 발광 제어, 터치 인식, 블루투스 수신, 블루투스 송신, 모터 제어, 엔코더 읽기, 자이로 센서 읽기, 가속도 센서 읽기, 인터럽트 실행이다.

각 Use-Case는 Description을 가지고 있다. 본 논문은 정의된 여러 개의 Use-Case 중 초음파 센서 읽기 Use-Case만 설명한다.

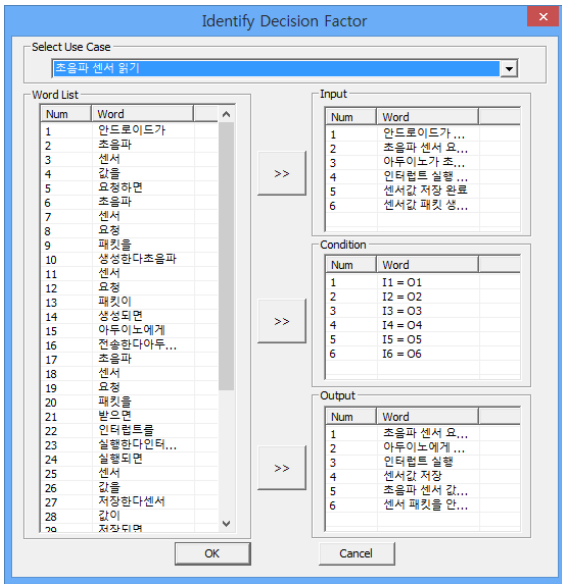


(그림 3) SmartController 설계

4. Test Case 추출

4.1. Decision Factor 추출

Use-Case에서 Decision Factor를 추출하기 위해서는 정의된 Description에서 Input, Condition, Output을 추출해내야 한다. 세 요소들을 수동으로 분류하여 Input, Condition, Output 결정한다. 그림 4는 Use-Case의 Description을 수동으로 분류하는 과정이다.



(그림 4) Use Case Description 추출

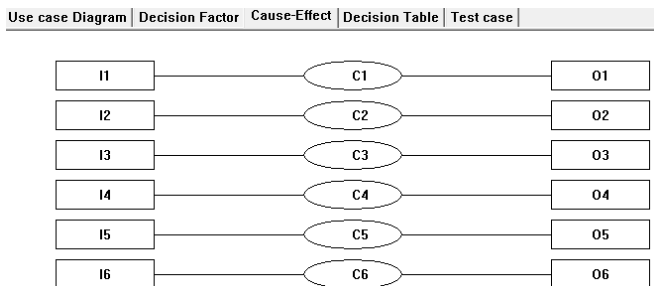
그림 5는 Use-Case Description을 설정한 결과이다. 그림 5의 I1~I6은 input, O1~O6은 output, C1~C6은 condition을 의미한다.

Use case Diagram	Decision Factor	Cause-Effect	Decision Table	Test case
List Num	Value			Type
I1	안드로이드가 조음파 센서 값 요청			input
I2	조음파 센서 요청 패킷 생성 완료			input
I3	아두이노가 조음파 센서 요청 패킷 수신			input
I4	인터럽트 실행 완료			input
I5	센서값 저장 완료			input
I6	센서값 패킷 생성 완료			input
O1	조음파 센서 요청 패킷 생성			output
O2	아두이노에게 조음파 센서 요청 패킷 전송			output
O3	인터럽트 실행			output
O4	센서값 저장			output
O5	조음파 센서 값을 패킷을 생성			output
O6	센서 패킷을 안드로이드에게 전송			output
C1	I1 = O1			condition
C2	I2 = O2			condition
C3	I3 = O3			condition
C4	I4 = O4			condition
C5	I5 = O5			condition
C6	I6 = O6			condition

(그림 5) Decision Factor 추출

4.2. Cause-Effect Diagram 생성

Decision Factor가 추출되었다면, 이를 기반으로 Cause-Effect Diagram을 생성한다. Cause-Effect Diagram의 I, C, O는 Decision Factor의 I, C, O와 동일한 의미이다. 그림 6는 Decision Factor로부터 생성된 Cause-Effect Diagram이다.



(그림 6) Cause Effect 생성

4.3. Decision Table 생성

Cause-Effect Diagram을 생성하면, 이를 기반으로 Decision Table로 변환한다. 입력의 Condition에 따라 출력의 결과가 달라지는 것을 테이블로 나타낸다. 그림 7은 Decision Table 순서를 보인다.

Use case Diagram	Decision Factor	Cause-Effect	Decision Table	Test case
			1 2 3 4 5 6 7 8 9 10 11 12	
Causes:				
11-안드로이드가 조음파 센서 값 요청	F	T		
12-조음파 센서 요청 패킷 생성 완료			F	T
13-아두이노가 조음파 센서 요청 패킷 수신			F	T
14-인터럽트 실행 완료				F
15-센서값 저장 완료				F
16-센서값 패킷 생성 완료				F
Effects:				
O1-조음파 센서 요청 패킷 생성	F	T		
O2-아두이노에게 조음파 센서 요청 패킷 전송			F	T
O3-인터럽트 실행			F	T
O4-센서값 저장				F
O5-조음파 센서 값을 패킷을 생성				F
O6-센서 패킷을 안드로이드에게 전송				F

(그림 7) Decision Table 생성

4.4. Test Case 추출

그림 7의 Decision Table을 기반으로 Test Case를 생성한다. Decision Table에서 출력된 모든 경우의 수를 Test Case로 나타낸다. 그림 8은 Test Case 생성을 보인다.

Use case Diagram	Decision Factor	Cause-Effect	Decision Table	Test case
No	Post Condition	Test Condition	Expectation Result	
TC 1	I1-안드로이드가 조음파 센서 값 요청	N/A	O1-조음파 센서 요청 패킷 생성=F	
TC 2	I1-안드로이드가 조음파 센서 값 요청=T	N/A	O1-조음파 센서 요청 패킷 생성=T	
TC 3	I2-조음파 센서 요청 패킷 생성 완료=F	N/A	O2-아두이노에게 조음파 센서 요청 패킷 전송=F	
TC 4	I2-조음파 센서 요청 패킷 생성 완료=T	N/A	O2-아두이노에게 조음파 센서 요청 패킷 전송=T	
TC 5	I3-아두이노가 조음파 센서 요청 패킷 수신	N/A	O3-인터럽트 실행=F	
TC 6	I3-아두이노가 조음파 센서 요청 패킷 수신=T	N/A	O3-인터럽트 실행=T	
TC 7	I4-인터럽트 실행 완료=F	N/A	O4-센서값 저장=F	
TC 8	I4-인터럽트 실행 완료=T	N/A	O4-센서값 저장=T	
TC 9	I5-센서값 저장 완료=F	N/A	O5-조음파 센서 값을 패킷을 생성=F	
TC 10	I5-센서값 저장 완료=T	N/A	O5-조음파 센서 값을 패킷을 생성=T	
TC 11	I6-센서값 패킷 생성 완료=F	N/A	O6-센서 패킷을 안드로이드에게 전송=F	
TC 12	I6-센서값 패킷 생성 완료=T	N/A	O6-센서 패킷을 안드로이드에게 전송=T	

(그림 8) Test Case 생성

5. 결론

본 논문에서는 객체 지향 기반의 펌웨어를 설계하기 위해 아두이노 적용과 안드로이드 기반의 어플리케이션을 펌웨어 설계와 검증을 위해서 Use-Case Diagram 기반 설계와 Gary E. Mogyorodi의 Cause-Effect Graphing 기법으로 Test Case를 발생한다. 이런 절차로 펌웨어와 어플리케이션을 객체 지향 방법으로 설계와 설계된 시스템의 안정성 테스트가 가능하다.

향후 연구는 절차적 언어로 작성된 펌웨어를 객체 지향 언어로 변환하는 방법을 연구하고, 정확한 Test Case를 추출하는 방법을 연구할 예정이다.

이 논문은 2013년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(2013R1A1A2011601)

참고문헌

- [1] 정혜정, 정원태, "소프트웨어 신뢰성 테스트를 위한 테스트케이스", 한국멀티미디어학회, Vol.13, No.1, pp.348-351, 2010
- [2] Chen Minsong, Qiu Xiaokang, Li Xuandong, "Automatic test case generation for UML activity diagram", Proceedings of the 2006 international workshop on Automation of software test AST, 2006.
- [3] 우수정, 김영철, "UML 메카니즘과 원인-결과 다이어그램 기반 테스트케이스 생성을 위한 자동 도구 개발", 한국정보과학회, Vol.39, No.1, pp.205-207, 2012
- [4] Gary E. Mogyorodi, "Requirements-Based Testing Cause-Effect Graphing", 2010
- [5] 장우성, 박찬민, 이철희, 김영철, "지능형 교통 RoboCAR 운행제어 검증을 위한 Test Case 추출 및 적용 연구", 한국정보처리학회, Vol.19, No.2, pp.1452-1455, 2012
- [6] Arduino, "<http://www.arduino.cc/en/Guide/Introduction>"
- [7] 류한석, "아두이노, 상상력을 실현하는 플랫폼", 주간경향, 2013
- [8] 이상운, 박찬민, "HBE-SmartCAR Firmware 설계", 한백전자, 2013