http://dx.doi.org/10.5392/JKCA.2015.15.08.001

UML을 이용한 아두이노 어플리케이션 설계

A Software Design Method for Arduino Applications using UML

박기창*, 이현철**, 김은석**
동신대학교 실감미디어기반조성사업단*, 동신대학교 디지털콘텐츠학과**

Ki-Chang Park(kcpark@dsu.ac.kr)*, Hyun-Cheol Lee(hclee@dsu.ac.kr)**,
Eun-Seok Kim(eskim@dsu.ac.kr)**

요약

아두이노는 오픈소스 기반 프로토타이핑 플랫폼으로, 미디어아트, 시뮬레이션, 인터랙션 장치 등 다양한 분야에서 활용되고 있다. 아두이노 어플리케이션은 C언어를 기반으로 개발되기 때문에 프로그램의 효과적인 설계명세 양식을 제공하지 못한다. 본 논문에서는 표준 객체지향 모델링 언어인 UML의 확장 매커니즘을 이용한 아두이노 어플리케이션 설계방법을 제안한다. 제안한 방법을 통해 아두이노 어플리케이션의 주기능인 디지털 입출력, 아날로그 입출력 등을 UML 표기법을 이용하여 명세할 수 있다.

■ 중심어 : | 아두이노 | 통합모델링언어 | 소프트웨어 설계 |

Abstract

Arduino is an open-source prototyping platform. It has been used in diverse areas including media-art, simulation and interaction devices. The Arduino application does not provide an effective form of software design specification, since it is basically developed in C Programming language. In this paper, we propose a design method for the Arduino applications using the extension mechanism of the UML(Unified Modeling Language). The proposed method is enable to specify the main function of Arduino applications like digital I/O and analog I/O in UML notations.

■ keyword : | Arduino | Unified Modeling Language(UML) | Software Design |

I. 서 론

아두이노는 오픈소스기반 마이크로컨트롤러 보드로, 2005년 이탈리아 Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, David Mellis 등 에 의해 개발되었다. 아트멜(Atmel)사의 8비트 AVR 마이크로컨트롤러 칩을 사용하는 아두이노는 빛, 온도, 소리, 거리 등을 센서를 통해 감지하고, 이를 이용한 사용자 상호작용 처리에 용이하기 때문에 미디어아트, 시뮬레이션, 인터랙션 장치개발 등에 주로 사용된다. 최근에는 사용자의 생체정보를 활용한 콘텐츠 개발[1][2], 기능성 게임 인터페이스[3], 자동차 사고 대응 시스템[4], IoT 응용시스템[5], 공연예술 분야[6] 등 다양한 연구에 활용되고 있다.

* 본 연구는 미래창조과학부와 정보통신기술진흥센터의 “실감미디어산업R&D기반구축및성과확산사업”의 지원을 받아 수행된 연구결과임.
접수일자 : 2015년 06월 12일
수정일자 : 2015년 07월 15일
심사완료일 : 2015년 07월 15일
교신저자 : 김은석, e-mail : eskim@dsu.ac.kr

아두이노 기본 보드에 LCD 디스플레이 쉴드, 이더넷 쉴드, 모터 쉴드, GPS 쉴드, Wifi 쉴드 등 다양한 기능의 H/W 모듈인 쉴드 보드를 추가적으로 연결하여 그 기능을 확장할 수 있다. PCB(Printed Circuit Board)가 공개되어 현재 우노, 듀에밀라노베, 데시밀라, 메가, 나노, 블루투스, 릴리패드 등 다양한 제품군이 출시되어 판매되고 있다. 이 중 가장 대표적인 모델은 [그림 1]의 아두이노 우노로 14개의 디지털 포트와 6개의 아날로그 포트를 가지고 있다.

그림 1. 아두이노 우노(출처: 위키피디아)

아두이노 어플리케이션은 C언어 문법을 사용하는 스케치(sketch)를 사용하여 개발된다. 아두이노는 프로세싱[7]과 Wiring[8]의 영향을 받아 개발환경이 [그림 2]와 같이 프로세싱과 거의 동일하고, 유사한 문법을 사용하며, 프로세싱, Max/Msp, Flash 등과의 연동을 통해 다양한 콘텐츠 개발도 가능하다.

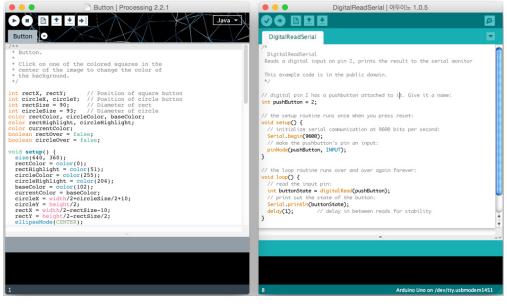


그림 2. 프로세싱(좌)과 아두이노(우)의 개발환경

대부분의 아두이노 프로젝트가 LED, 모터, 적외선 센서 등 외부 장치와 관련 있기 때문에 [그림 3]과 같은 회로도를 통해 그 구조를 가시화 한다. 이는 물리적인 장치의 연결구조는 표현할 수 있지만 소프트웨어적인 구조를 표현하지는 않는다. 또한 C언어를 기반으로 하기 때문에, 전통적인 구조적 설계양식인 [그림 4]와 같은 구조도(structure chart)를 사용하여 시스템 설계를 할 수 있으나, 구조도는 구조적 프로그램에서의 프로그램 모듈사이의 계층적 가시화가 주목적이기 때문에 아두이노 어플리케이션과 같이 모듈의 수가 적을 경우 활용도가 떨어진다.

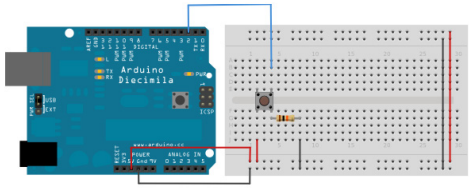


그림 3. 아두이노 어플리케이션 회로구성도의 예

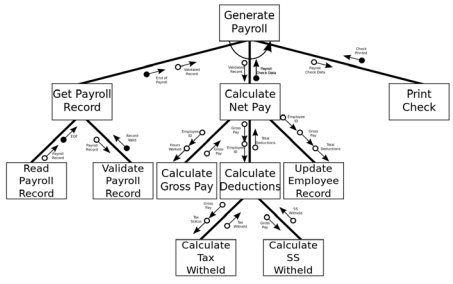


그림 4. 구조도(Structure Chart)의 예

아두이노 어플리케이션과 같은 임베디드 소프트웨어 설계 시에는 하드웨어 관련 정보 표현이 필수적인데 반해, 기존의 소프트웨어 공학기법에서는 이러한 정보를 표현할 수 있는 방법을 제공하지 못한다[9].

따라서 본 논문에서는 UML의 확장 매커니즘을 통해 아두이노 어플리케이션의 주 기능이 디지털 I/O, 아날로그 I/O 관련 어플리케이션에서 활용할 수 있는 설계 방안을 제시한다. 본 논문의 구성은 다음과 같다. 2장에서는 UML 프로파일관련 기술을 소개하고, 3장에서는 아두이노 어플리케이션의 구조를 분석하고, 이를 UML로 명세하기 위한 스테레오타입, 태그 등을 제시한다. 4장에서는 디지털 입출력과 아날로그 입출력 관련 사례연구를 통해 제안 방법의 활용방안을 살펴보고 5장에서는 결론 및 향후 연구방향을 제시한다.

II. 관련연구

UML은 유즈케이스 다이어그램, 클래스 다이어그램, 시퀀스 다이어그램, 액티비티 다이어그램 등 여러 종류의 다이어그램을 통해 대상시스템을 구조적(structural), 행위적(behavioral) 관점(view)에 따라 모델링할 수 있는 강력한 도구이다. UML은 클래스, 객체, 속성, 행위, 상속, 메시지 등 객체지향 프로그램의 설계에 필요한 요소를 가시화(visualization)하여 시스템의 관심사항을 이해관계자(stakeholder)사이에 공유하는데 도움을 준다.

UML은 범용적인 모델링 도구로 확장 메커니즘을 통해 다양한 도메인(domain)을 모델링하는데 사용될 수 있다[10]. UML 확장 메커니즘이란 스테레오타입(stereotypes), 제약사항(constraints), 태그정의(tag definitions), 태그값(tagged value)을 의미하며, 이들의 묶음을 프로파일(profile)이라고 한다.

표 1. UML 확장 매커니즘

구분	내용
스테레오타입 (stereotype)	· 모델링 요소에 새로운 표기법(Notation) 을 정의
제약사항 (constraint)	· 모델링 요소에 제약 사항을 통해 요소의 의미를 재정의
태그정의 (tag definition)	· 모델링 요소에 새로운 속성 항목 정의
태그값 (tagged value)	· 모델링 요소의 속성에 부가적으로 정의된 속성

UML 프로파일을 사용하여 Android[11], iOS[12], Windows Phone 7[13]과 같은 소프트웨어 개발 플랫폼에서 구동되는 응용 어플리케이션을 모델링하거나 GIS(Geographic Information System)[14], USN(Ubiquitous Sensor Network)[15]과 같은 특정 도메인을 모델링하는데 UML을 확장하여 사용할 수 있다. C언어와 같은 구조적 프로그램언어를 UML을 통해 가시화하기 위해 Douglass[16]는 FunctionalC 프로파일을 통해 시스템의 요구사항(requirements), 구조(structure), 행위(behavior)를 모델링하는데 UML을 사용할 수 있도록 확장하였으며. Bakal[17]은 C언어의 파일, 변수, 함수의 개념을 클래스, 프로퍼티, 메소드로 맵핑하여 표현하였다.

III. UML을 이용한 아두이노 어플리케이션 설계

1. 아두이노 어플리케이션 구조 분석

아두이노 어플리케이션은 일반적인 C 프로그램과 달리 main() 함수가 존재하지 않으며, 대신 setup(), loop() 함수를 포함한다. setup()함수는 프로그램이 구동될 때, 한번 실행되며 이후 loop()함수를 반복적으로 실행하게 된다. 기본 제공 함수로는 디지털 입출력, 아날로그 입출력 등의 장치인터페이스 함수와 시간, 수학, 비트연산, 삼각함수, 인터럽트 등이 있다. 아두이노 개발 레퍼런스 사이트[18]에는 함수를 기능 및 역할에 따라 [표 2]와 같이 13개의 그룹으로 구분하고 있다.

표 2. 아두이노 함수

함수 그룹	내용
Digital I/O	· 디지털 입출력 관련 함수
Analog I/O	· 아날로그 입출력 관련 함수
Due only	· 아두이노 Due 용 아날로그 API 확장 함수
Advanced I/O	· tone(), noTone() 등 입출력 함수
Time	· delay() 등 시간처리 함수
Math	· min(), max()등 수학 함수
Trigonometry	· sin(), cos(), tan() 등 삼각함수
Random Numbers	· 난수발생 함수
Bits and Bytes	· 비트 처리 함수
External Interrupts	· 외부 인터럽트 처리 함수
Interrupts	· 인터럽트 처리 함수
Communication	· 시리얼, 스트림 등 통신 함수
USB	· 키보드, 마우스 입력 처리 함수

아두이노 어플리케이션의 주요기능을 파악하기 위해 [19]에 수록된 54개의 아두이노 코드를 대상으로 기본 제공함수의 출현빈도를 조사하였다. [그림 5]에 나타난 것처럼, 54개의 코드에 아두이노 기본함수는 총 457번 나타났으며, 이 중 가장 출현빈도가 높은 함수그룹은 pinMode(), digitalRead(), digitalWrite()가 속한 Digital I/O 그룹과 Serial, Stream 객체 등이 속한 Communication 그룹으로 이 두 그룹이 314번 출현하여 전체의 68%를 차지하였다. 즉, 대부분 아두이노 어플리케이션은 외부 장치와 디지털 입출력, 통신이 주 기능으로 파악되었다.

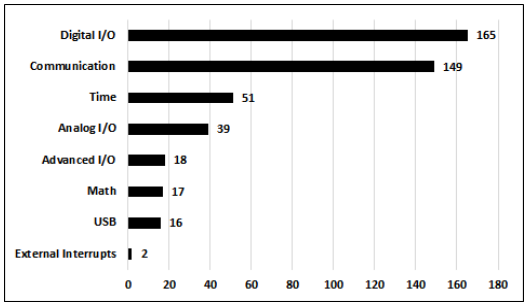


그림 5. 주요기능별 출현빈도

아두이노 개발 레퍼런스 사이트에 명시된 함수 중 총 28종의 함수가 등장하였으며, 이중 가장 출현빈도가 높은 함수는 Serial.print()로 이는 어플리케이션의 실행결과 확인을 위한 목적으로 자주 등장하였다. 다음으로 digitalWrite(), pinMode(), delay(), analogWrite(), Serial.begin(), analogRead(), digitalRead() 순으로 높은 출현빈도를 보였다. 데이터 입출력, 통신을 위한 이상 8개의 함수가 전체의 84%를 차지하였다. 주요함수의 출현빈도 조사결과는 [그림 6]과 같다.

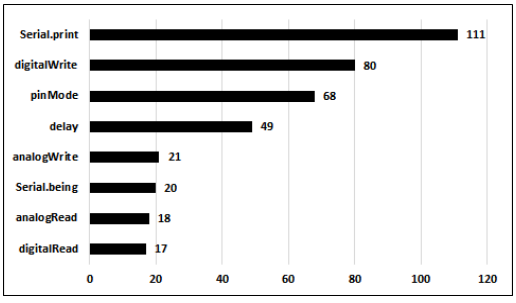


그림 6. 함수별 출현빈도

2. 아두이노 어플리케이션 설계를 위한 UML 확장

아두이노 어플리케이션의 구조분석결과를 반영하여 아두이노 어플리케이션의 주 기능이 데이터 입출력 명세를 위해 필요한 스테레오타입과 태그를 정의하였다. UML에서 스테레오타입은 기존의 메타클래스에서 추가적인 의미를 부여한 시각적인 표기법을 정의한 하나의 모델요소이다. 스테레오타입은 클래스(class), 관계(relation), 유즈케이스(use case), 액터(actor) 등 UML의 모든 구성요소에 적용할 수 있으며, 본 연구에서는 클래스와 관계에 적용한다. 스테레오타입을 표기할 때에는 모델요소에 "<<스테레오타입명 >>"형식을 붙여 확장된 의미를 부여한다. 클래스에 적용된 대표적인 스테레오 타입으로 UML 표준 프로파일에 정의된 <<boundary>>, <<control>>, <<entity>>가 있다. 여기서 boundary는 사용자 인터페이스 담당 클래스, entity는 데이터 클래스, control은 boundary와 entity를 제어하는 이벤트 처리 클래스를 의미한다. 이들은 객체지향 시스템에서 MVC (Model-View-Controller) 모델을 효과적으로 명세하는데 사용된다. 일반적으로 스테레오타입이 적용된 클래스를 도식화 하는 방법은 [표 3]과 같이 두 가지 형태로 표현할 수 있다. 일반적인 클래스표기법인 사각형대신 특정 이미지(icon)를 사용하여 의미전달의 효과를 높이기도 한다, StarUML[20]등 대부분의 UML 모델링 도구에서 두 가지 표현법 모두를 지원한다.

표 3. 클래스 스테레오타입 표기법 예

스테레오타입	스테레오타입 표기	아이콘 표기
⟨⟨boundary⟩⟩	«entity»	
⟨⟨control⟩⟩	«control»	
⟨⟨entity⟩⟩	«boundary»	

아두이노 어플리케이션 설계를 위한 클래스 스테레오 타입은 [표 4]와 같이 정의하며. 적용대상은 데이터 입출력과 직접적인 관련이 있는 아두이노 보드, 디지털 입력장치, 디지털 출력장치, 아날로그 입력장치, 아날로그 출력장치, 입출력 값으로 한정한다.

표 4. 클래스 스테레오타입

스테레오타입	설명
⟨⟨Arduino⟩⟩	· 아두이노 보드를 표시
⟨⟨Digital Input⟩⟩	· 디지털 입력장치를 표시
⟨⟨Digital Output⟩⟩	· 디지털 출력장치를 표시
⟨⟨Analog Input⟩⟩	· 아날로그 입력장치를 표시
⟨⟨Analog Output⟩⟩	· 아날로그 출력장치를 표시
⟨⟨Value⟩⟩	· 디지털, 아날로그 입출력 값을 표시

관계 스테레오타입은 디지털 입출력, 아날로그 입출력을 표현하기 위한 <<digital>>, <<analog>>로 정의한다.

표 5. 관계 스테레오타입

스테레오타입	설명
〈〈digital〉〉	· 디지털 입출력 관계를 표시 · 아두이노와 디지털 입출력장치, 디지털입출력장치와 입출력 값 사이의 관계를 표시 · 입력의 경우 화살표 방향이 입력장치에서 아두이노로 향함 · 출력의 경우 화살표 방향이 아두이노에서 출력장치로 향함
〈〈analog〉〉	· 아날로그 입출력 관계를 표시 · 화살표 방향의 의미는 〈〈digital〉〉의 경우와 동일

정의된 스테레오타입을 도식화하기 위해 클래스 표기는 앞서 설명한 <<boundary>>, <<control>>, <<entity>> 스테레오타입의 아이콘 표기법을 적용하였다. 여기서 아두이노 보드는 <<boundary>>, LED, 모터, 적외선 센서 등 입출력장치는 <<control>>, 입출력 값은 <<entity>>로 대응하여 표기하였다. 입출력 값은 디지털 값(HIGH, LOW)과 아날로그 값(0~255)에 따라 3가지 형태로 구분하였다. 방향성 관계는 아두이노와 입출력장치, 입출력장치와 입출력 값 사이를 데이터 흐름을 화살표로 표기한다.

표 6. 제안 스테레오타입의 이미지표현

스테레오타입	이미지
〈〈Arduino〉〉	아두이노 보드를 표시
〈〈Digital Input〉〉 〈〈Digital Output〉〉 〈〈Analog Input〉〉 〈〈Analog Output〉〉	디지털/아날로그 입출력 표시
〈〈Value〉〉	디지털값 : HIGH 디지털값 : Low 아날로그값 : 0 ~ 255
〈〈digital〉〉	«digital» «digital»
〈〈analog〉〉	«analog» «analog»

태그정의는 모델요소의 속성을 나타내는 도구로, <<digital>>, <<analog>> 스테레오타입의 방향성 관계에서 핀(pin)번호와 출력 값을 나타낼 수 있는 조건값(value)으로 표현한다.

표 7. 태그정의

태그	적용대상	설명
pin	아두이노와 입출력 장치 사이의 연관 (Association)	· pinMode(), ditalRead/Write(), analogRead/Write()에 사용되는 파라미터 pin 번호를 표시 · 아두이노와 입출력 장치와 물리적으로 연결되는 핀 번호를 의미
value	출력장치와 출력값 사이의 연관 (Association)	· 출력 값을 나타내기 위한 조건 표시

IV. 사례연구

1. 디지털 입출력

III장에서 제안한 UML확장을 이용하여 아두이노 디지털 입출력 어플리케이션을 명세해 보았다. 예제 코드는 [그림 7]과 같이 아두이노 보드와 연결된 버튼을 통해 사용자 입력을 받고, 버튼의 눌림에 따라 LED를 on/off 하는 코드를 사용하였다.

```
1  : int ledPin = 13;
2  : int buttonPin = 2;
3  : int buttonState = 0;
4  : void setup()
5  : {
6  :   pinMode(buttonPin, INPUT);
7  :   pinMode(ledPin, OUTPUT);
8  : }
9  : void loop()
10 : {
11 :    buttonState = digitalRead(buttonPin);
12 :    if (buttonState == HIGH)
13 :      digitalWrite(ledPin, HIGH);
14 :    else
15 :      digitalWrite(ledPin, LOW);
16 : }
```

그림 7. 디지털 입출력 예제 코드

코드의 1~3라인은 LED 핀, 버튼 핀, 버튼 눌림의 상태 값을 표현하기 위한 변수선언부이다. 4~8라인의 setup() 함수는 어플리케이션 구동 시 한번 실행되는 함수로 버튼과 LED를 각각 입력장치, 출력장치로 해당 핀(buttonPin, ledPin)에 연결하여 초기화 한다. 9~16라인의 loop()함수는 어플리케이션이 종료되기 까지 반복적으로 실행되는 함수로 여기서는 digitalRead()함수를

통해 버튼으로부터 눌림 상태 값을 buttonState변수에 할당한 후, buttonState 값(HIGH, LOW)에 따라 LED를 on/off 시킨다. 이러한 디지털 입출력 코드는 제안방법을 통해 [그림 8]과 같이 명세할 수 있다.

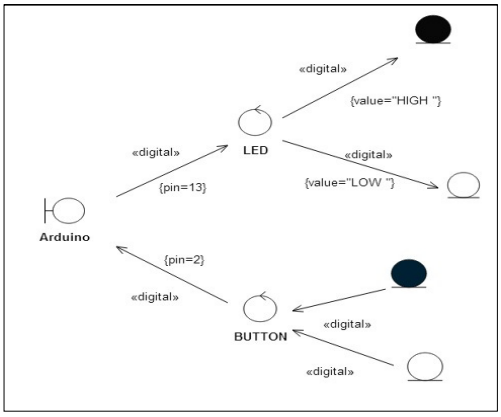


그림 8. 디지털 입출력 코드를 UML로 표현

[그림 8]의 다이어그램을 통해 다음과 같은 정보를 얻을 수 있다.

- 아두이노 보드와 연결된 입출력장치는 LED, BUTTON이다.
- BUTTON은 입력장치, LED는 출력장치이다.
- 데이터 입출력은 디지털 입출력 방식(digitalRead, digitalWrite)을 사용한다.
- BUTTON은 2번 핀에 연결되고, LED는 13번 핀에 연결된다.
- BUTTON으로 HIGH, LOW 값이 들어오고, LED는 HIGH, LOW에 따라 출력신호가 결정된다.
- pinMode(13, OUTPUT), pinMode(2, INPUT), digitalWrite(13, HIGH), digitalWrite(13, LOW) 등 주요함수와 그 파라미터 정보를 제공한다.

2. 아날로그 입출력

아날로그 입출력 예는 모터에 0~255사이의 출력값을 전달하여 모터의 구동속도를 증가시켰다가 감소시키는 예로 코드는 [그림 9]와 같다.

```
1  : int motorPin = 9;
2  : void setup()
3  : {
4  :   pinMode(motorPin, OUTPUT);
5  : }
6  : void loop()
7  : {
8  :    for (int i = 0; i < 256; i++)
9  :    {
9  :      analogWrite(motorPin, i);
10 :     delay(10);
11 :    }
12 :    for (int i = 255; i >= 0; i--)
13 :    {
14 :      analogWrite(motorPin, i);
15 :      delay(10);
16 :    }
17 :    delay(1000);
18 : }
```

그림 9. 아날로그 출력 예제 코드

1라인에서 모터 핀을 9로 지정한 후 4라인에서 출력장치로 아두이노와 연결한다. 8~10라인은 아날로그 신호 값을 0에서 255까지 점진적으로 증가시키면서 모터의 구동속도를 증가시킨 후, 12~16라인에서 다시 출력값을 255에서 점차 감소시키면서 속도를 줄이게 된다.

제안 방법을 통해 아두이노, 모터, 2가지 출력을 [그림 10]과 같이 명세할 수 있다. 2가지 출력의 경우 점증적인 신호의 증가, 감소를 표현하기 위해 UML의 노트(Note) 표기법을 이용하였으며, 자세한 출력상태 제약사항이나 부가적인 정보를 기술하기 위해 UML의 일부인 OCL(Object Constraint Language)[21]을 이용하여 추가사항을 명세할 수 있다.

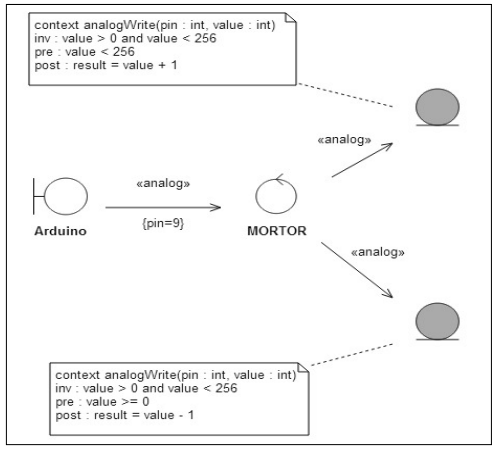


그림 10. 아날로그 출력 코드를 UML로 표현

[그림 10]의 다이어그램을 통해 다음과 같은 정보를

얻을 수 있다.

- 아두이노 보드에 모터가 출력장치로 연결된다.
- 모터는 9번 핀에 연결되어 있고 아날로그 출력(analogWrite)을 한다.
- 아날로그 출력은 2가지 방식으로 진행된다.
- pinMode(9, OUTPUT), analogWrite(9, i)등 주요 함수와 그 파라미터 정보를 제공한다.

V. 결론 및 향후 연구과제

본 논문에서는 UML을 사용한 아두이노 어플리케이션 명세방법을 제안하였다. 주 명세대상을 한정하기 위해 54개의 아두이노 소스코드에 등장한 함수의 빈도를 조사하여 주 기능이 데이터 입출력임을 확인하고, 이에 따라 UML 확장 매커니즘인 스테레오타입, 태그정의를 통해 명세 요소를 제시하였다. 또한 제안한 명세방법을 이용하여 간단한 디지털입출력, 아날로그입출력 어플리케이션의 명세 예를 보였다. 향후, 아두이노의 확장요소인 다양한 쉴드(shield)와의 관계를 표현하기 위한 연구와 명세로부터 골격코드(skeleton code)를 자동으로 생성하는 방법에 관한 연구가 필요하다.

참 고 문 헌

[1] 박소희, 박종승, “아두이노 장치를 활용한 사용자의 공포 정도 측정”, 한국컴퓨터종합학술대회 논문집, pp.1934-1935, 2014.

[2] 박소희, 박종승, “Unity3D와 아두이노를 이용한 사용자의 생체 신호 실시간 반영”, 학국정보과학회 동계학술발표회, pp.1445-1447, 2014.

[3] 고정운, 박성준, “아두이노를 활용한 노인용 인지기능 향상 기능성 게임”, 한국정보기술학회논문지, 제13권, 제4호, pp.111-119, 2015.

[4] 박상하, 김서영, 이부형, “블루투스, GPS를 이용한 아두이노기반 자동차 사고 대응 시스템”, 한국통신학회 추계종합학술발표회, pp.114-115, 2014.

[5] 강진원, 김진성, 전유빈, “아두이노를 이용한 식물관리 시스템”, 한국컴퓨터종합학술대회 논문집, pp.1818-1820, 2014.

[6] 하유이, 김이경, “공연예술에서의 표현 확장을 위한 웨어러블 컴퓨터 연구”, HCI 학술대회, pp.633-636, 2014.

[7] https://processing.org/reference/environment/#Overview

[8] http://wiring.org.co/about.html

[9] 이진호, 심재환, 최진영, 고병령, “임베디드 소프트웨어 설계 명세서의 개발”, 한국컴퓨터종합학술대회 논문집, pp.212-215, 2011.

[10] L. F. Fernandez and A. V. Moreno, “An Introduction to UML Profiles,” The European Journal for the Informatics Professional, Vol.4, No.2, pp.6-13, 2004.

[11] 고민혁, 국승학, 윤상필, 김현수, “안드로이드 애플리케이션 개발을 위한 UML 프로파일”, 한국컴퓨터종합학술대회 논문집, 제38권, 제1호, pp.148-151, 2011.

[12] 서용진, 문대건, 국승학, 김현수, “iOS 애플리케이션을 위한 UML 프로파일”, 한국컴퓨터종합학술대회 논문집, 제38권, 제1호, pp.116-119, 2011.

[13] 민법기, 국승학, 김현수, “윈도우폰7 플랫폼 기반 스마트 디바이스 애플리케이션의 모델링을 위한 UML 프로파일”, 한국컴퓨터종합학술대회 논문집, 제38권, 제1호, pp.120-123, 2011.

[14] J. L. Filho, G. B. Sampaio, F. R. Nalon, and K. A. de V. Borges, “A UML Profile for Conceptural Modeling in GIS Domain,” Workshop DE@CAiSE, pp.18-31, 2010.

[15] 이우진, 최일우, “효과적인 USN 어플리케이션 설계를 위한 UML 프로파일”, Entrue Journal of Information Technology, 제11권, 제2호, pp.215-226, 2012.

[16] B. P. Douglass, *UML for the C programming language*, IBM, 2009.

[17] http://www.drdobbs.com/cpp/uml-for-c -programmers/184401948

[18] http://www.arduino.cc/en/Reference/HomePage
[19] J. Blum, *Exploring Arduino*, wiley, 2013.
[20] http://staruml.io/
[21] http://www.omg.org/spec/OCL/

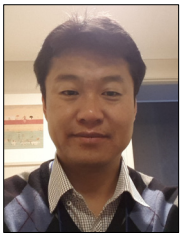
저 자 소 개

박 기 창(Ki-Chang Park) 정회원

- 2001년 2월 : 전남대학교 물리학과(이학사)
- 2003년 2월 : 전남대학교 전산학과(이학석사)
- 2013년 8월 : 전남대학교 전산학과(이학박사)
- 2008년 4월 ~ 2013년 9월 : 동신대학교 디지털콘텐츠 협동연구센터 선임연구원
- 2013년 10월 ~ 현재 : 동신대학교 실감미디어기반조성사업단 선임연구원

<관심분야> : 소프트웨어공학, 실감미디어

이 현 철(Hyun-Cheol Lee) 정회원



- 1989년 2월 : 동신대학교 디지털콘텐츠학과(이학사)
- 1998년 2월 : 동신대학교 계산통계학과(이학석사)
- 2003년 2월 : 동신대학교 컴퓨터학과(이학박사)
- 2005년 ~ 현재 : 동신대학교 디지털콘텐츠학과 조교수
- 2012년 ~ 현재 : 동신대학교 실감미디어기반조성사업단 실장

<관심분야> : 멀티미디어통신, 증강현실, 실감미디어

김 은 석(Eun-Seok Kim) 종신회원

- 1995년 2월 : 전남대학교 전산학과(이학사)
- 1997년 2월 : 전남대학교 전산통계학과(이학석사)
- 2001년 2월 : 전남대학교 전산통계학과(이학박사)
- 2002년 ~ 현재 : 동신대학교 디지털콘텐츠학과 부교수

<관심분야> : 컴퓨터그래픽스, 가상현실, 실감미디어