

블록코딩으로 아두이노의 막강한 기능을 제한 없이 사용하기 위한 방법 연구

강민수† · 김석전†

† 송도중학교

A method for unlimited use of Arduino's powerful functions with block coding.

Min-Su Kang† · Seok-Jeon Kim†

† Songdo Middle School

요 약

초, 중, 고 정규 교육과정에 피지컬컴퓨팅 단원이 포함 되었고 교구 지향적인 특성을 갖는다. 피지컬 단원을 위해 매우 다양한 교구가 출시 되고 있으나 학생들의 창의적인 아이디어를 모두 수용 가능하며 저렴한 교구는 찾기 힘들다. 저렴하고 막강한 기능을 가진 아두이노가 최적의 보드이지만 어려운 하드웨어 세팅과 텍스트 코딩 기반이라 외면 받고 있다. 또한 블록코딩이 가능하긴 하나 단순한 디지털, 아날로그 센서 제어 명령만 제공하다 보니 다른 보드와 기능적인 차이는 없으면서 하드웨어 세팅만 어렵게 느껴진다. 아두이노가 가진 다양한 입, 출력 모듈을 제한 없이 사용 가능한 장점을 훼손하지 않으며, 코딩이 쉬우려면 센서 제한 없는 블록코딩이 되어야 한다. 만약 아두이노가 하드웨어 세팅이 쉬우면서 거의 모든 입출력장치를 블록명령으로 제어가능하다면 초중고 학생들이 자신의 아이디어를 제한 없이 구현하여 창의적인 작품을 만들 수 있을 것이다. 따라서 블록코딩으로 아두이노의 막강한 기능을 제한 없이 사용하기 위한 방법을 연구하고, 해결책을 찾아 제시한다.

1. 서 론

학교에서 사용하는 다양한 피지컬컴퓨팅 교구가 대부분 아두이노를 개량하여 사용하기 쉽게 만들어 놓았다. 하드웨어 세팅이 쉽고, 전용 블록언어를 지원하기 때문에 사용하기는 참 편하다. 그러나 다른 센서를 사용할 수 없게 만들어 놓은 연결케이블이나 핀 번호를 변형하여 다른 보드는 사용이 불가능하다. 범용성과 호환성이 매우 떨어지고, 제공되는 센서와 액추에이터 이외에 다양한 입출력 장치를 사용하기 불가능한 단점이 있으며 블록명령어도 지원이 되지 않는다. 한마디로 제한 적인 기능 아래서 하드웨어 세팅과 코딩이 매우 쉽다.

아두이노는 오픈 소스를 기반으로 하여 마이크로 컨트롤러 칩을 사용해 만들어진 단일 보드 마이크로 컨트롤러와 관련 개발 도구 및 환경을 말한다. 2005년 이탈리아의 IDII(Interaction Design Institute Ivera) 에서 하드웨어에 익숙지 않은 학생들이 자신들의 디자인 작품을 손쉽게 제어할 수 있게 하려고 고안되었다. 아두이노는 다양한 다수의 센서와 액추에이터를 비교적 손쉽게 동작시킬 수 있어 로봇, 데이터 수집, IoT 등 다양한 분야에 접목되고 있다.

아두이노의 기본 개발 환경은 C++ 을 사용하는 avr-gcc를 기반으로 한 Sketch를 사용한다. C++ 을 사용하기 때문에 초심자나 어린 학생들이 사용하기에

는 많이 어려울 수 있다. 그래서 보통 코딩 교육 교구로 아두이노를 사용할 때는 EPL의 아두이노 확장 블록을 사용하는데 아두이노의 확장 블록은 대부분 디지털 입출력, 아날로그 입력 정도만 가능하고 추가로 초음파, 서보 모터, 부저출력을 지원한다. 하지만 이 정도의 하드웨어지원으로는 라이브러리가 필요한 다양한 센서와 액추에이터를 사용할 수 없다.

기존 교구들이 아두이노를 기반으로 사용하기 쉽게 만들었지만 서로 호환이 안되고, 추가 센서도 사용할 수 없기 때문에 교구로 적합하지 않다. 이 문제를 해결하는 가장 좋은 방법은 오리지널 아두이노의 기능을 그대로 사용가능하면서 하드웨어 세팅이 쉽고, 모든 입출력 장치를 제어할 블록명령어를 제공하는 EPL이 있으면 된다. 만약 쉬운 하드웨어 세팅과 사용제한 없는 블록 명령이 갖추어지면 초중고 학생들이 자신의 아이디어를 제한 없이 구현하여 창의적인 작품을 만들 수 있을 것이다.

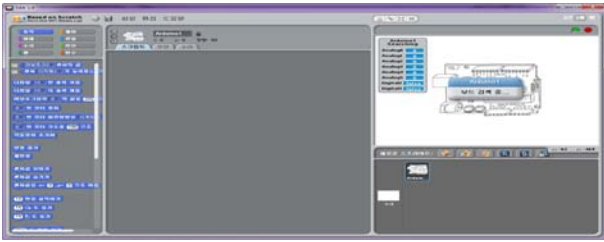
2. 이론적 배경

2.1 피지컬 컴퓨팅 교구 분석

학교에서 사용하는 다양한 피지컬 컴퓨팅 교구의 특성을 살펴보면 학생들이 사용하기 쉽게 만드는 것에 초점을 맞추었다. 거의 모든 교구가 기존 아두이노의 하드웨어 세팅 방식이 어렵다는 것을 인지하고 나름대로

로의 방식으로 하드웨어 세팅을 쉽게 만들었다. 교구별 차이점이라면 블록 명령어를 제공하는 정도의 차이가 있다. 모든 교구가 단순 아날로그/디지털 입출력 명령블록은 모두 제공하고, 추가적으로 제품 패키지 내의 센서나 액추에이터를 사용할 수 있는 명령블록이 있다.

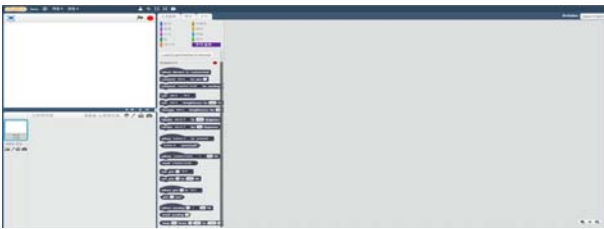
2.1 하드웨어 제어가능 EPL 분석



[그림 1] S4A

하드웨어를 제어가능한 EPL로는 ScratchX, Entry, S4A, Makecode, mBlock, Arduino on Scratch 등이 있다.

S4A는 기본적인 아날로그/디지털 명령블록만 지원하고 Scratch 1.4 기반으로 개발되어서 사용하기 불편하다. 하드웨어 제어용 EPL로는 매우 부적합하다[1].



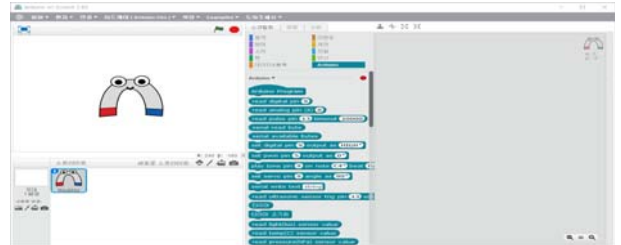
[그림 2] ScratchX

ScratchX은 기본적인 아날로그/디지털 명령블록만 지원하기 때문에 아두이노를 제어하기에는 많이 부족하다. 그래서 그 단점을 보완하기 위해 확장 블록을 추가할 수 있지만 사용자의 PC에 스크립트를 실행해서 서버를 열어 확장 블록을 추가하는 방식을 사용하기 때문에 상대적으로 개발 방법이 까다롭다. 그리고 사용자가 자체적으로 확장 블록을 개발을 해도 다른 사람과 공유할 방법이 없다[2].



[그림 3] mBlock

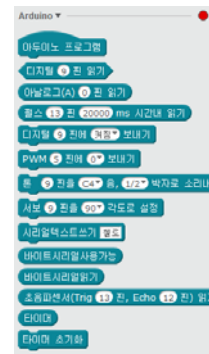
mBlock은 하드웨어 제어에 가장 최적화된 EPL이라고 할 수 있다. mBlock은 기본적인 아날로그/디지털 명령블록이외에도 초음파센서, 서보모터, 부저, 시리얼통신, 타이머를 기본으로 지원한다. 블록으로 코딩한 후 아두이노에 업로드하여 아두이노가 컴퓨터에 연결되지 않고도 독립적으로 동작하게 만들 수 있다. 그리고 mBlock의 가장 큰 장점은 확장 블록 추가가 가능하며 개발이 상대적으로 쉽다는 것과 그 확장 블록을 다른 사람과 공유 할 수 있다는 것이다. ScratchX처럼 서버를 열지 않고도 확장 블록을 추가할 수 있다. mBlock은 확장 센터를 지원하기 때문에 개발한 확장 블록을 공유하면 다른 사람도 쉽게 다운 받아서 사용할 수 있다[3].



[그림 4] Arduino on Scratch

Arduino on Scratch는 mBlock의 소스를 사용하여 만들어져서 mBlock의 기능과 매우 유사하다. 기본적으로 30가지 이상의 명령블록을 지원해주기 때문에 하드웨어 제어에 적합하다고 할 수 있다. 하지만 mBlock과 다르게 Arduino on Scratch는 확장 블록을 추가할 수 없기 때문에 보통 아두이노와 자주 쓰이는 LCD 같은 모듈은 Arduino on Scratch에서 기본 지원을 해주지 않는 이상 사용할 수 없다.

Entry는 가장 많은 하드웨어를 지원하는 EPL이다. 하지만 아두이노 확장 블록은 아날로그/디지털 명령블록 정도만 지원한다. Entry도 확장 블록을 만들어 사용할 수는 있지만 HTML 5 기반이다 보니 ScratchX처럼 서버를 열어 확장 블록을 추가하는 방식을 사용한다. 그래서 개발 방법도 상대적으로 까다롭고 만든 확장을 공유하기 위해선 Entry 측의 문의하여 확장을 등록해야하는 번거로움이 있다[4].



[그림 5] mBlock에서 기본 지원하는 명령블록

2.1 블록코딩을 활용한 아두이노 제어

아두이노를 제어할 수 있는 하드웨어 제어가 가능한 EPL로는 S4A, ScratchX, mBlock, Arduino on Scratch, Entry 등이 있다. 그러나 모두 기본적인 센서나 액추에이터만 사용가능하고, 더 이상의 고급 센서나 액추에이터는 블록코딩으로 사용이 불가능하다.

이 중에서 mBlock은 자사가 판매중인 아두이노 호환보드와 로봇을 제어할 목적의 하드웨어 제어용 확장블록을 다수 만들어 놓았다. 기능적으로 타사와 차이점은 없지만 사용자가 직접 확장블록을 만들 수 있는 기능을 가지고 있는 것이 가장 큰 특징이다. 추가적인 장점으로 블록 코딩을 하고 그것을 텍스트 코드로 변환할 수 있어 학생 입장에서 블록과 텍스트를 비교하는 것이 가능하고, 아두이노에 코드를 업로드하여 컴퓨터에 연결되어있지 않은 상태에서 독립적으로 작동시킬 수 있다.

확장 블록을 쉽게 제작할 수 있고 공유도 가능한 mBlock을 활용하면 아두이노의 다양한 센서와 액추에이터를 쉬운 블록 코딩만으로 제어가 가능해지고 그로 인해 낮은 연령층뿐만 아니라 진 연령층이 쉽게 아두이노를 사용해 창의적인 아이디어를 구현할 수 있게 될 것이다[5][7].

3. 본론

3.1 아두이노 제어 확장 블록 제작

3.1.1 Sketch의 구조 이해

<표 1> mBlock 확장 블록의 구조별 역할

구분	역할
js 폴더	mBlock의 스크래치 모드에서 작동되는 JavaScript 파일이 들어있는 폴더
.js 파일	mBlock의 스크래치 모드에서 아두이노와 통신을 위해 하드웨어 통신 방법을 정의한 JavaScript 파일
src 폴더	.s2e 파일에 포함된 Sketch에 선언된 라이브러리 파일들(.cpp, .h)이 들어있는 폴더
.s2e 파일	확장 블록의 설명과 블록의 모양, 기능을 정의하고 아두이노 모드에서 동작되는 Sketch 코드가 포함된 JSON 형식의 파일
.ino 파일 (펌웨어)	mBlock의 스크래치 모드에서 PC와 통신하기 위해 아두이노에 업로드되는 펌웨어

Sketch는 avr-gcc를 기반으로 개발된 C++ 문법을 사용하는 아두이노 프로그래밍 언어이다. Sketch는 크게 4부분으로 나눌 수 있다.

“라이브러리 포함 선언 부”는 미리 등록한 라이브러리의 파일명을 쓰거나 사용할 라이브러리의 경로를 정

의해주어 컴파일 시 라이브러리를 불러올 수 있게 한다.

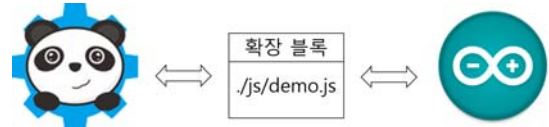
“변수, 상수 선언 부”는 변수나 상수 또는 객체(인스턴스)를 선언하는 부분이다.

“코드 초기화 부”는 처음 한 번만 실행되는 시스템 초기화 코드 부분이다. 주로 아두이노의 핀모드, 시리얼 통신 활성화, 모듈 초기화 코드 같은 것을 넣는다.

“코드 반복 실행 부”는 계속해서 반복되어 실행되어야 하는 코드 부분이다. 주로 실제 동작을 위한 코드를 넣는다.

3.1.1 mBlock 확장 블록의 구조 이해

mBlock 확장 블록은 js폴더, src폴더, .s2e파일, .ino 파일로 크게 4부분으로 나눌 수 있다.



[그림 6] PC와 아두이노의 통신 관계

3.1.1.1 js 폴더 및 파일

PC는 아두이노를 제어하기 위해 통신을 하게 되는데 이때 확장 블록 파일 중 하드웨어 통신 방법과 각각의 블록에 따라 전달 인자, 응답 처리를 정의한 JavaScript 파일을 통하여 아두이노 펌웨어와 데이터를 주고 받게 된다.

js 파일은 크게 객체 선언 부, 블록 별 함수 정의 부, 하드웨어와 통신을 위한 함수 3부분으로 나눌 수 있다.

변수 객체 선언 부는 코드 내에서 사용되는 변수와 객체를 선언되는 부분이다.

블록 별 함수 정의 부는 각각의 블록에 따라 아두이노에 요청과 함께 전달할 인자와 받은 응답을 어떻게 처리할 것인지를 정의하는 부분이다. 하드웨어 통신 함수 정의 부에서 정의된 함수를 호출해 통신 방법이 정의된다.

하드웨어 통신 함수 정의 부는 mBlock에서 제공하는 하드웨어 통신 API 이다. 액추에이터 작동 요청 함수, 센서 값 요청 함수, 데이터형 별 읽기 함수를 지원한다.

3.1.1.1 src 폴더

src 폴더 안에는 .s2e 파일에 포함된 Sketch 코드에 #include 전처리로 포함 선언된 라이브러리의 파일들 (.cpp, .h)이 들어있는 곳이다. #include 전처리에 “”를 사용할 경우 src 폴더에 있는 라이브러리들을 참조하게 된다.

3.1.1.1 .s2e 파일

.s2e 파일은 확장 블록의 설명, 확장 블록의 모양과 아

두이노 모드에서 사용되는 Sketch 코드, 각각의 블록과 연결되어 있는 .js 파일에 정의되어 있는 함수, 확장 블록의 번역을 정의하는 JSON 형식의 파일이다.

확장 블록 설명 부는 확장 블록의 이름, 설명, 버전, 작성자, 홈페이지, 태그, 정렬순위, 확장포트, 펌웨어 버전이 정의되는 부분이다.

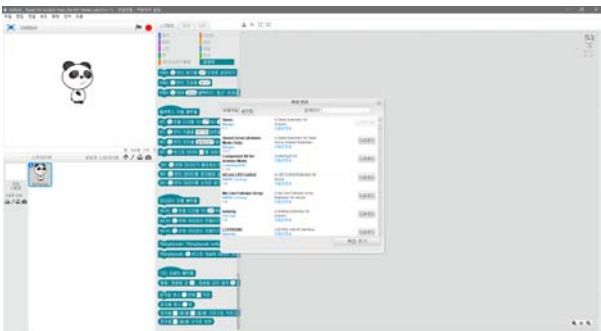


[그림 7] s2e 파일의 구조

확장 블록 정의 부는 블록의 모양, 블록의 콘텐츠, 연결된 JavaScript 함수, 텍스트 박스 기본값, Sketch 코드, 드롭 메뉴, 메뉴 콘텐츠 객체가 정의되는 부분이다. 확장 블록 번역 부는 블록의 콘텐츠를 언어 별로 번역이 정의되는 부분이다[6].

3.1 확장 블록 등록 방법

mBlock의 장점 중 하나인 확장 블록 공유를 이용하면 제작한 확장 블록을 다른 사람과 공유하여 낮은 연령층 뿐만 아니라 전 연령층이 블록 코딩만으로 대부분의 센서와 액추에이터를 사용할 수 있게 된다. 제작한 확장 파일들을 하나의 압축파일로 만든다.

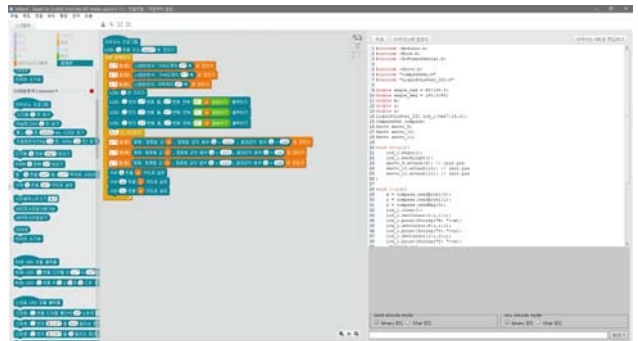


[그림 8] mBlock 확장 관리

mBlock 확장 센터(mblock.cc/extensions/)에 접속하여 GitHub 계정으로 로그인한다. GitHub 계정이 없다면 새로 발급을 받아야 한다. 그리고 로그인을 하게 되면 맨 위에 사각형 박스가 생기게 되는데 그것을 클릭하여 확장 블록 압축 파일을 선택한다. 이때 확장 블록의 문법적 오류가 있을 경우 업로드가 되지 않는다. 성공적으로 업로드가 완료되면 리스트에 업로드한 확장장이 뜨게 된다. 확장 센터에 업로드한 확장 블록은 mBlock에 확장 > 확장 관리에서 다운로드해 사용할 수 있다.

4. 결론 및 논의

확장 블록을 이용해 기존 아날로그/디지털 입출력 블록으로 제어하지 못하는 센서나 액추에이터를 제어할 수 있게 되었다. 현재까지 추가한 명령 블록은 네오픽셀, 도트매트릭스, 온습도센서, 자이로스코프, 먼지센서, DC모터, IR 송수신, LCD, FND, 와이파이, 블루투스이다. 앞으로 MP3모듈, 스텝모터 등 다양한 입출력 장치를 자유롭게 등록이 가능하다.



[그림 9] 자이로스코프 서보모터 제어 예제

확장 블록을 등록하여 사용한 결과 기본적인 아날로그/디지털 입출력 장치만 사용했을 때 보다 훨씬 다양한 작품을 만들 수 있게 되었다. 또한 교구 특성에 따른 제한적이지 않은 피지컬 컴퓨팅 실습을 할 수 있고, 학생들이 생각하는 모든 것을 블록코딩으로 쉽게 제작할 수 있어 학교의 친구들이 창의성을 발휘하는 도구로 사용되길 희망한다.

참고 문헌

- [1] S4A <http://s4a.cat/>
- [2] 스크래치X <https://scratchx.org/>
- [3] 엠블릭 <https://www.makeblock.com/>
- [4] 엔트리 <https://playentry.org/>
- [5] 김석전(2015). 중학교 정보교과의 피지컬 컴퓨팅단원 프로그래밍수업에서 아두이노를 이용한 효과적인 접근법. **컴퓨터교육학회논문지**, 21(1), 45-47.
- [6] 엠블릭 포럼 <https://www.forum.makeblock.com/>
- [7] 김석전(2016). A practical APPROACH TO Arduino programming for THE physical computing section of the Informatics Curriculum IN Korean Middle School. EDUlearn16. Barcelona (Spain).