

논문접수일 : 2013.01.05

심사일 : 2013.01.13

게재확정일 : 2013.01.25

아두이노 기반의 **DIY Pacduino** 키트 개발

Development of an Arduino-based DIY Pacduino kit

신 은 주

인천가톨릭대학교 조형예술대학 시각디자인과 교수

EunJoo Shin

Incheon Catholic University. College of Fine Art & Design. Dept. of Visual communication design

1. 서 론

- 1.1. 연구의 배경
- 1.2. 연구의 방법 및 범위

2. 미디어 저작 환경 및 개발자의 변화

- 2.1. 미디어 저작 환경의 변화
- 2.2. 개발자의 변화

3. DIY Pacduino의 기술 디자인

- 3.1. 보드 디자인
- 3.2. 샘플 코드

4. DIY Pacduino의 사용자 매뉴얼

- 4.1. 프린트 매뉴얼
- 4.2. 온라인 매뉴얼

5. 사용자와의 소통

- 5.1. 워크숍
- 5.2. 블로그

6. 결 론

참고문헌

논문요약

미디어 아트, 디자인에서는 미디어 기술을 자유롭게 활용할 수 있는 응용력과 영상, 통신, 게임, 사운드 등 다양한 영역을 자유롭게 융복합할 수 있는 창의적인 사고의 상호 습득이 요구된다. 이러한 특성이 잘 반영된 오픈소스 하드웨어 운동은 아티스트와 디자이너에게 인기를 끌고 있으며, 기술을 공개하여 이의 재사용을 촉진시키고 대규모 커뮤니티 플랫폼을 형성해 다양한 프로젝트의 기술과 아이디어를 공유할 수 있는 저작 환경을 구축하고 있다. 커뮤니티를 통해 아티스트나 디자이너 등 누구나 첨단 기술이 요구되는 프로젝트를 엔지니어나 프로그래머의 도움 없이 직접 개발하고 완성할 수 있으며 개발과정에서 공개된 기술 소스를 응용하여 재개발한 기술을 다시 공개할 뿐만 아니라 물리적인 키트로도 만들어 판매하고 공유한다. 이렇게 기술과 아이디어를 함께 제공하는 융합 지식의 순환적 공유로 이루어진 새로운 저작 환경은 미디어 아트, 디자인의 대중화 및 다변화를 이끌고 있다. 이에 본 연구는 기술을 공유하고 아이디어로 소통하기 위한 목적으로 2010년부터 2011년까지 수차례의 워크숍-빅토리아 앤 앨버트 미술관(Victoria & Albert Museum, 런던), 크리에이터스 프로젝트(The Creator's Project, 서울), 퓨처 에브리씽 미디어 페스

티벌(Future Everything, 맨체스터), 므즈테크(MzTek, 런던)-을 진행하며 체계적으로 발전시켜온 팩맨(Pacman) 게임과 오픈소스 하드웨어, 아두이노(Arduino)가 결합된 DIY Pacduino 게임 키트 개발을 케이스 스터디로 제안하고자 한다.

주제어

오픈소스 하드웨어, 아두이노, 다이아이와이(DIY) 키트 디자인

Abstract

It is required that freely applying media technology to the various areas such as moving images, communication, game and sound with creative thinking in media art and design. Reflecting the above-mentioned requirements in the fields of media art and design, open-source hardware, has been popular with artists and designers and more building a new creative environment through the large community platform where technologies opens for the public to reuse them and projects' ideas are shared. Through the community platform, anyone including artists, designers can develop his/her projects based on the knowledge of high technologies without helps of engineers or programmers and sell physical kits to share the redeveloped open source technology applications and skills in progress of their project's development. A new created environment, where the convergence of technology and creative ideas to share is occurred, leads media art and design to be more popularized and varied. In this study, 'DIY Pacduino', manipulating retro game -Pacman and open source hardware, Arduino, is proposed as a case study that has been systematically developed through the several workshops in Victoria and Albert Museum(London), The Creator's Project(Seoul), Future Everything Media Festival(Manchester), MzTek(London) to share the used technology and developed idea.

Keyword

Open source hardware, Arduino, DIY kit design

1. 서론

1.1. 연구의 배경

미디어 아트, 디자인 분야에서 사용되는 기술은 공학 분야에서 사용되는 기술 못지않게 높은 수준이 요구되며 창의적인 상상력만큼이나 중요하다. 창의적인 상상력을 표현하고 싶어도 미숙한 기술력을 가지고 있으면 실현할 수 없기 때문이다. 반면에 기술을 아무리 자유자재로 활용할 수 있어도 상투적인 생각에서 벗어나지 못하고 단지 기술 구현에만 그친다면 아트나 디자인으로서의 가치를 잃게 된다. 즉, 미디어 기술을 자유롭게 활용할 수 있는 응용력과 영상, 통신, 게임, 사운드 등 다양한 영역을 자유롭게 융복합할 수 있는 창의적인 사고의 습득이 동시에 이루어지고 구현되어야 한다. 이러한 분야의 특성으로 인하여 미디어 아티스트와 디자이너들은 소프트웨어와 소프트웨어 구성 시 원류가 되는 소스를 무료 배포함으로써 전문적인 프로그래밍 기술을 공유하는 오픈소스 소프트웨어(open source software)를¹⁾ 저작도구로서 선호하게 되었다. 더욱이 소프트웨어에서 확장된 오픈소스 하드웨어 운동은 크리에이티브 커먼즈(creative commons) 라이선스를²⁾ 기반으로 사용자가 공개된 기술을 본인의 의도에 적합하게 이용하고 발전시키도록 기술의 재사용을 적극적으로 촉진하고, 대규모 커뮤니티 플랫폼을 형성해 다양한 기술과 아이디어를 공유할 수 있도록 하는 새로운 저작 환경을 구축하고 있다. 아티스트나 디자이너 등 누구나 커뮤니티를 통해 무인기 제어, 자작 3D 프린터기, 식물과 SNS로 소통하는 보태니컬즈(Botanicals)³⁾와 같이 첨단 기술지식이 요구되는 프로젝트도 직접 개발하고 완성할 수 있다. 또한 프로젝트 진행과정에서 공개된 기술을 응용하여 재개발하고 이를 다시 공개할 뿐만 아니라 물리적인 키트로도 만들어 판매, 공유한다. 이렇게 기술과 아이디어를 함께 제공하는 융합 지식의 순환적 공유로 이루어진 새로운 저작 환경은 미디어 아트, 디자인의 대중화 및 다변화를 이끌고 있다. 하지

1) 오픈소스 소프트웨어(open source software) 운동은 1976년 데니 엘리슨(Dennis Alison)이 Tiny BASIC을 부상으로 배포하면서 시작되어 소프트웨어나 소스코드를 공개하고 공유함으로써, 누구나 공개된 코드를 기반으로 변형하여 재배포할 수 있다.

2) 크리에이티브 커먼즈(creative commons) 라이선스는 소프트웨어와 하드웨어 등의 기술에서부터 모든 저작물에 대해 원칙적으로 사람들의 자유이용을 허락하되 몇 가지 이용방법 및 조건을 추가하는 개방적 이용허락이다.

3) 로버트 파루디(Robert Faludi), 케이티 하트먼(Kate Hartman), 케이티 런던(Kati London) 등 다수의 미디어 아티스트, 디자이너와 엔지니어로 이루어진 협업 그룹은 SNS등을 이용해 식물과 사람이 서로 소통할 수 있는 보태니컬즈를 개발하였다.

만 이러한 움직임은 여전히 일부 아티스트와 디자이너, 엔지니어의 참여에만 국한되어 있어 미디어 아트, 디자인의 보다 폭 넓은 대중화를 위해서는 기술을 공유하기 위한 효율적이고 체계적인 시스템과 아이디어의 소통을 위한 시스템 디자인의 방법에 대한 다양한 연구가 필요하며, 다수의 아티스트와 디자이너의 적극적인 참여가 요구된다. 이에 본 연구에서는 팩맨 게임과 오픈소스 하드웨어 아두이노가 결합된 대중 지향적인 하드웨어 디자인 개발과 지식 공유를 위한 정보 제공 시스템 디자인의 구축을 위해 2010년부터 2011년까지 수차례 워크숍-빅토리아 앤 앨버트 미술관(Victoria & Albert Museum, 런던), 크리에이터스 프로젝트(The Creator's Project, 서울), 퓨처 에브리씽 미디어 페스티벌(Future Everything, 맨체스터), MzTek(런던)-을 통해 대중과의 직접적인 소통으로 보안, 수정하며 발전시켜온 DIY Pacduino(팩두이노 이하 Pacduino) 게임 키트를 케이스 스티디로 제안하고자 한다.

1.2. 연구의 방법 및 범위

첫째, 대표적인 오픈소스 하드웨어 운동인 아두이노를 중심으로 미디어 아트, 디자인 분야에서 전문적인 첨단 기술이 편이하게 습득되고 이용되는 새로운 저작 환경이 구축되고 있음을 분석한다. 둘째, 아티스트와 디자이너는 창작자의 역할에서 더 나아가 기술을 응용, 개발하고 이를 아이디어에 적용하는 과정을 공유함으로써 대중과의 거리를 좁히고 폭넓은 소통을 이끌고 있음을 살펴본다. 셋째, 케이스 스티디로 진행한 Pacduino 게임 키트의 기술 디자인(전자 보드, 샘플 코드), 온-오프라인 메뉴얼을 통한 기술 지식의 전달 시스템, 워크숍과 블로그를 통한 정보 공유 및 아이디어의 소통까지 일련의 과정을 3단계에 걸쳐 살펴봄으로써 일반 사용자에게 기술을 전달하기 위한 효율적인 시스템과 상호간의 아이디어를 나눌 수 있는 방안을 제시한다.

2. 미디어 저작 환경 및 개발자의 변화

2.1 미디어 저작 환경의 변화

오픈소스 소프트웨어라는 개념을 (간단한 회로기판에서 안드로이드 전화 같은 첨단 가전제품에 이르는) 하드웨어 세계로 확대한 새로운 개념이 부상하면서 리눅스 OS나 컴퓨터 소프트웨어에서 시작된 오픈소스 운동은 하드웨어 분야에까지 적용되고 있다 (크리

스 앤더슨, 2012). 이러한 변화는 미디어 아트, 디자인에서의 무료 저작기술의 확장과 아이디어의 공유, 생산과 재배포라는 새로운 움직임을 이끌고 있다.

미디어 아트, 디자인에서 이용하는 대표적인 오픈소스 소프트웨어인 프로세싱(Processing), 포브이(vvvv)와 오픈프레임웍스(OpenFrameworks) 등은 비디오 소스나 사운드 데이터를 실시간으로 처리하여 시각화하기 위해 주로 프로그래밍 기술을 사용한다. 물론 비주얼, 오디오 데이터를 받아들이고 내 보내기 위해 웹캠, 마이크, 스피커, 프로젝터와 미디 등 간단한 인-아웃풋(in-output) 장치를 사용하지만 복잡한 피지컬 컴퓨팅(physical computing)으로⁴⁾ 제작한 하드웨어를 함께 사용하기에는 다소 어려움이 있었다. 소프트웨어와 하드웨어가 연동하려면 데이터를 주고받기 위해 프로그램 라이브러리와 통신을 지원하는 하드웨어(컨트롤러 보드)가 필요한데 사실, 다수 소프트웨어와 연동 가능한 기술 지원이 부족하거나 관련 자료도 아티스트나 디자이너가 활용하기엔 너무 공학적이었다. 그럼에도 각종 센서, 모터, LED, GPS, RFID 등과 같은 하드웨어를 피지컬 컴퓨팅 기술로 제어해 소프트웨어와 접목시킬 때 보다 더 다양하고 풍부한 기술-창의적 표현이 가능하자 미디어 아트, 디자인에서도 하드웨어의 오픈소스화, 융합 기술의 응용과 아이디어 공유의 활성화에 대한 필요성이 대두되었다.

2005년에 마시모 벤지(Massimo Banzi)와 데이비드 쿼티엘레르(David Cuartielles)는 인터랙션 디자인 프로젝트를 용이하게 배우고 제작할 수 있도록 AVR 프로세서를 사용한 오픈소스 마이크로컨트롤러 보드인 아두이노를 개발했다. 아두이노는 아티스트와 디자이너를 위한 마이크로컨트롤러 보드로서 세계적으로 인기를 끌며 오픈소스 하드웨어 운동을 주도하고 있는데 이러한 성공은 크게 3가지 측면에서 기인한다.



[그림 1] 아두이노 우노(Uno)보드 및 아두이노 IDE⁵⁾

(1) 이글(Eagle) 파일로 된 회로도 와 보드 디자인, 자바(Java) 기반의 편집기, 컴파일러와 프로그램 업로더가 합쳐진 통합 개발 환경 IDE(Integrated Develop

4) 피지컬 컴퓨팅(Physical Computing)은 디지털 기술을 통해 사용자로부터 물리적인 방식으로 정보를 입력받거나 정보를 처리한 결과를 물리적인 방식으로 출력하는 컴퓨팅을 말한다.

5) 출처 : <http://arduino.cc>

ment Environment), 샘플 코드와 라이브러리 등 모든 기술 소스가 공개되어 아두이노 보드를 손쉽게 사용하고 프로그래밍할 수 있다. 이러한 오픈소스 시스템은 먼저 개발되었던 예술가용 보드, 와이어링(Wiring)에서 차용됐다. 아두이노의 원류인 와이어링도 AVR 프로세서를 사용한 마이크로컨트롤러 보드로서 많은 미디어 아티스트와 디자이너가 사용하는 오픈소스 소프트웨어인 프로세싱에 맞춘 통합 개발 환경 IDE와 개발을 위한 라이브러리를 제공하며 프로세싱과 유사하게 프로그래밍할 수 있도록 하는 장점이 있다. 아두이노는 이를 그대로 가져다 쓰면서 와이어링 보다 작고 값싼 마이크로프로세서를 사용해 최대한 간략하게 보드를 재설계하여 가격을 낮추고 비용을 절감시켰다.

(2) 아두이노의 웹 섹션-다른 소프트웨어와의 연동을 위한 인터페이스-에서는 소프트웨어는 물론 스마트 기기, 타 마이크로컨트롤러 보드와의 연동 방법을 상세히 다루고 지원한다. 이를 참조하여 프로세싱, 포브이, 오픈프레임웍스, 맥스/엠에스피 지터(Max/Msp Jitter), 플래시(Flash), 매트랩(Matlab), 오픈시비(OpenCV) 등 다수 소프트웨어와 피지컬 컴퓨팅 기술이 접목된 스크린 기반의 미디어 프로젝트 구현이 가능하고 아이폰, 안드로이드와 같은 스마트 폰의 앱(App)과 아두이노를 연동으로 외부 하드웨어를 스마트 기기로 제어하는 등 아두이노는 타 기술과의 융복합을 통한 미디어 콘텐츠의 확장을 가져왔다.

(3) 대규모 온라인 사용자 커뮤니티를 통한 기술과 아이디어의 공유로 다양한 사용자 그룹(미디어 아티스트, 디자이너, 엔지니어, 프로그래머, 메이커(maker) 등)은 새로운 아이디어로 재개발하여 재배포하는 애드온 보드나 가젯을 파생시키고 있다. 이렇게 파생된 아두이노 기반의 보드는 그 수를 셀 수 없을 만큼 스파크펀(SparkFun), 메이커 쉼드(Maker Shed), 아이다푸루트(Adafruit) 등의 온라인 숍(shop)으로 퍼져 나가고 있다.

아두이노의 철학은 디자인을 말로 설명하기보다는 직접 만드는 것과 프로토타이핑(prototyping)의 완성도를 높이기 위해서 다양한 프로토타이핑 기술을 사용해보면서 손으로 생각하는 과정을 중요시하는 것에 기반을 두고 있다(마시모 벤지, 2010). 이는 미디어 아트, 디자인 분야에서 더욱 극명하게 드러나는데 기술을 직접 익히고 충분히 이해해야지만 실현 가능한 상상도 할 수 있고 구현시킬 수 있기 때문이다. 기술과 아이디어의 하나의 융합물은 오픈소스 운동이라는 슬로건 아래 최초의 개발에서 다른 사용자에게 의해 꼬리에 꼬리를 물고 진화해 나가고 있다.

2.2 개발자의 변화

디지털 시대의 인간은 상상을 기술로 실현한다는 기술적 상상력(techno-imagination)이 시대의 새로운 상징형식으로 떠올랐다(진중권, 2012). 과거와는 달리 기술이 표현의 수단이 되고 있는 것이다. 그러나 현대의 기술은 복잡해서 혼자서 처음부터 끝까지를 모두 고안해내기가 어려워졌다. 다른 사람이 개발한 기술을 응용해 쓰지 않고는 내 아이디어를 실현하기가 어려운 시대가 된 것이다. 이에 아티스트, 디자이너, 엔지니어 등은 자발적으로 오픈 하드웨어 섬미트(Open Hardware Summit) 웹 사이트와 같은 온라인 커뮤니티와 수백 개의 오픈 프로젝트가 나와 있는 포럼 및 위키(wiki)의 토론에 참여해 오픈소스 운동을 다각도로 펼쳐 나가고 있다.

여성 엔지니어이자 오픈소스 하드웨어 운동의 리더인 라이머 프리드(Limor Fried)는 연령이나 전자 기술 경험의 유무와 관계없이 모든 메이커를 위한 최고의 DIY 키트를 소개하고 공유하기 위한 온라인 공간을 만들고자 2005년 아이다푸루트를 설립했다. 드로디오(Drwadio), 브레인 머신 키트(Brain machine kit), 트론백(Tron bag) 등 그녀가 개발한 키트를 판매하고 블로그 레이디아다(ladyada.net)에서 아이디어 드로잉, 전자 기술과 코드를 공개하고 남뱀에서부터 완성하는 과정까지를 사진으로 포스팅하여 상세히 설명해주기 때문에 초보자도 쉽게 따라 만들 수 있다.



[그림 2] 아이다푸루트에서 판매되는 브레인 머신 키트와 드로디오)

오렐리(O' Reilly) 출판사는 정기적으로 메이커 매거진(Make Magazine)을 발간해 아티스트, 디자이너와 메이커가 제작한 다양한 아이디어와 프로젝트를 소개한다. 또한 연중행사로 메이커 페어(Maker Fair)를 개최해 재미나고 기발한 생각을 실행할 수 있는 전자 기술과 관련 프로젝트를 소개하는 워크숍도 진행한다. 매거진 발행과 함께 일반 대중도 자연스럽게 흥미를 갖도록 유도하는 축제 같은 분위기의 페어 행사는 보다 많은 사람이 전자 기술에 관심을 갖고 이를 유연하게 본인들의 생각에 추가해 프로젝트를 제작할 수 있도록 촉진하는 대표적인 크리에이터 운동으로

6) 출처 : <http://www.ladyada.net>

확대되고 있다.



[그림 3] 메이커 매거진과 메이커 페어의 DIY CNC 기계 자작 시현 워크숍7)

취알엘(GRL-Graffiti Research Lab), 패트(FAT-Free Art & Technology Lab)와 오픈플레이믹스의 아티스트와 엔지니어가 모여 몸이 마비된 동료 작가를 위해서 오픈소스 기반의 안구 추적 시스템과 저비용의 카메라를 이용하여 눈동자로 글을 쓰거나 그림을 그려서 세상과 소통하는 아이라이터(EyeWriter)를 개발했다. 2010년, 프리스 아르스 일렉트로니카(Prix Ars Electronica)와 퓨처 에브리씽 미디어 페스티벌에서 수상하여 널리 공감대를 형성한 프로젝트임에도 불구하고 모든 기술이 공개되었다. 아이라이터는 일반 안경에 부착되어 눈동자를 추적하는 카메라와 IR센서로 구성된 하드웨어 장치와 동공의 위치를 분석해 그림을 그릴 수 있는 영상처리 소프트웨어로 구성되어 있는데, 하드웨어 장치를 만드는 데 필요한 전자부품 구입처부터 완성 방법이 단계별로 설명되어 있고 소프트웨어도 다운로드 받을 수 있어 관심이 있는 개발자는 누구나 다시 재개발할 수 있다.



[그림 4] 아이라이터로 건물 외벽에 실시간으로 그래피티 이미지를 생성8)

공개된 기술과 아이디어가 원래의 개발자가 전혀 상상하지 않은 방식으로 사용되어 예상하지 못한 것들이 끊임없이 개발되는 잠재성이야말로 오픈 소스 하드웨어 운동의 핵심이라고 할 수 있다. 최초의 개발자에서 사용자까지 모두가 개발자가 될 수 있고 또 사용자가 될 수 있다. 최초의 개발자는 존재하지만, 그 후엔 누가 먼저 다시 만들었느냐는 그리 중요치 않다. 이러한 새로운 협업 방식은 아티스트, 디자이너, 엔지니어와 메이커들의 자발적인 참여로 형성되어 이제 일반 대중에게까지 퍼져 나가고 있다.

7) 출처 : <http://makezine.com>

8) 출처 : <http://eyewriter.org>

3. DIY Pacduino의 기술 디자인

Pacduino는 팩맨 게임과 아두이노가 결합된 DIY 게임 키트로서, 일반 사용자도 쉽게 이해하고 응용해 사용할 수 있도록 고안되었다. 단순하게 변형된 팩맨 게임을 하거나 LED 매트릭스(matrix)에 이미지나 텍스트를 생성할 수 있도록 사용자가 직접 조립하고 프로그래밍한다. 일반적으로 피지컬 컴퓨팅을 배울 때, 스위치를 이용한 LED 제어를 먼저 접하는데 Pacduino는 스위치로 LED의 켜기(on)/끄기(off) 제어는 물론, 게임 인터페이스로 응용하는 방법까지 배울 수 있어 교육 효과도 기대할 수 있다.



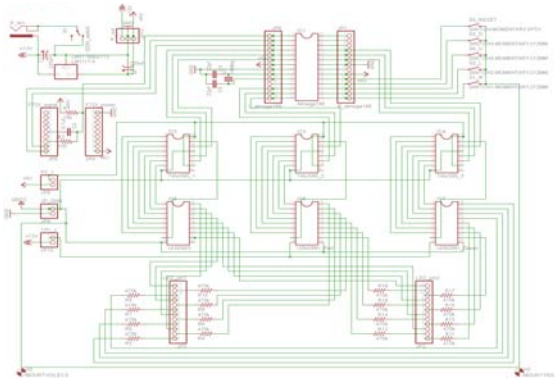
[그림 5] Pacduino로 팩맨 로고 생성 및 게임하기

3.1 보드 디자인

전자 보드 디자인은 아두이노 보드, 게임 키트로 사용하기 위한 LED 매트릭스 컨트롤 보드, 아두이노 보드와 게임 인터페이스의 분리인 3파트로 나뉜다.

(1) 아두이노 두에밀리나노브(Arduino Duemilina-nove)보드

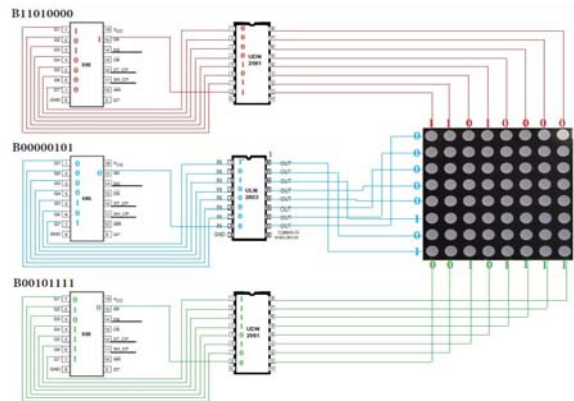
4개의 스위치의 인풋 입력을 받아 팩맨 게임의 캐릭터(떡이를 먹는 입 모양, 떡이 모양, 떡이가 잡혔을 때의 모양)를 LED 매트릭스에 구성하기 위해 오픈소스 자료로 아두이노 두에밀리나노브 보드를 만들고 FTDI 시리얼 컨버터를 사용해 아두이노와 컴퓨터 간의 시리얼 통신이 가능하도록 했다. 게임 코드 데이터 용량을 고려해 대용량 플래시 메모리(Flash Memory)가 탑재된 Atmega328을 사용하였다.



[그림 6] Pacduino의 회로도

(2) 2톤 8X8 LED 매트릭스 컨트롤 보드

팩맨 로고와 같은 이미지나 텍스트를 만들려면 2톤 8X8 LED 매트릭스의 총 128개의 LED(64개의 빨강과 64개의 녹색)를 독립적으로 제어를 해야 한다. 빨강과 녹색 LED의 양극은 세로로 따로 연결되어 있고 음극은 가로로 같이 연결되어 있어 켜기/끄기 신호를 각각의 16개의 양극과 8개의 음극에 보내야 한다. 그러려면 총 24개의 디지털 포트가 필요한데 마이크로컨트롤러에서는 16개의 디지털 포트만 사용 가능하므로 74HC595로⁹⁾ 디지털 포트의 8비트(bit) 시리얼 신호를 8개의 1비트 패러럴 신호로 변환하여 포트를 확장시켰다. 8개 빨강 LED의 양극, 8개 녹색 LED의 양극, 8개 빨강-녹색 LED의 공통 음극은 각각 3개의 74HC595에 연결된다. 다시, 3개의 74HC595은 2개의 UDN2981과 1개의 ULN2803에 연결되는데 128개의 LED에 충분한 전압과 전류를 공급하기 위해 입력 전원을 12V, 2A 이상으로 설정했기 때문이다. 2개의 UDN2981은¹⁰⁾ 빨강과 녹색 LED의 양극에 12V전압을 인가해 주고 다른 1개의 ULN2803은¹¹⁾ 빨강과 녹색 LED의 공통 음극을 제어해 LED에 전류를 흘려준다.



[그림 7] 2톤 8X8 LED 매트릭스 컨트롤 보드 연결도

(3) 아두이노 보드와 게임 인터페이스의 분리

게임 인터페이스 (2톤 8X8 LED 매트릭스 컨트롤 보드와 스위치)의 구동 전원 (12V+, 5V+, GND)에 연결된 점퍼를 제거해 게임 인터페이스를 분리함으로써 Pacduino 키트를 아두이노 보드로만으로 사용할 수 있다.

9) 8비트 쉬프트 트랜지스터 (8bit shift transistor)
 10) 8채널 소스 드라이버 트랜지스터 어레이(channel Source driver transistor array)
 11) 고 전류 달링턴 트랜지스터 어레이(High currenxy darlington transistor array)

3.2. 샘플 코드

팩맨 게임과 이미지나 텍스트를 스크롤 할 수 있는 2가지 샘플 코드가 제공된다. 사용자는 샘플 코드를 그대로 사용해 팩맨 게임을 할 수도 있고 본인이 스스로 게임 코드를 만들거나 이미지 혹은 텍스트를 만들어 사용할 수 있다.

첫째, 게임 코드에서는

(1) 설정 문에서는 버튼 라이브러리를 추가하고 4개의 버튼 입력 포트와 LED 매트릭스에 보내는 신호의 출력 포트를 지정한다. 그리고 먹이, 먹이를 잡는 화살표 이미지와 먹이를 잡았을 때 색상을 바꾸기 위한 LED 데이터를 변수로 설정해 준다.

```
#include <Button.h>
Button button1 = Button(11, PULLUP);
Button button2 = Button(12, PULLUP);
Button button3 = Button(10, PULLUP);
Button button4 = Button(9, PULLUP);
Button button5 = Button(13, PULLUP);
int clockPin[] = {2,6,14};
int latchPin[] = {3,7,15};
int dataPin[] = {4,8,16};
int bc_l; int bc_r; int bc_u; int bc_d;
byte row_l[5]; byte col_l[5]; byte row_r[5]; byte col_r[5];
byte row_u[5]; byte col_u[5]; byte row_d[5]; byte col_d[5];
int row_lshift; int row_rshift; int col_ushift; int col_dshift;
byte row_p[2]; byte col_p[2]; byte r_led[8];
int random_pre = 5; int catch_p_row; int catch_p_col;
void setup() {
  for(int i=0; i < 3; i++) {
    pinMode(clockPin[i], OUTPUT);
    pinMode(latchPin[i], OUTPUT);
    pinMode(dataPin[i], OUTPUT);
    digitalWrite(latchPin[i], LOW);
    shiftOut(dataPin[i], clockPin[i], MSBFIRST, 0);
    digitalWrite(latchPin[i], HIGH); }
}
```

(2) 실행 문에서는 먹이를 잡는 화살표 이미지를 4방향(좌-우-위-아래) 중 어느 방향으로 생성할 것인지를 각각의 스위치 신호를 체크해 먼저 입력되는 방향으로 결정한다.

```
void loop() {
  set;
  bc_l = 0; bc_r = 0; bc_u = 0; bc_d = 0;
  row_lshift = 0; row_rshift = 0; col_ushift = 0; col_dshift = 0;
  r_led[0] = B00011000; //log2 of pacman
  r_led[1] = B00111100;
  r_led[2] = B01111110;
  r_led[3] = B11111111;
  r_led[4] = B11111111;
  r_led[5] = B11011011;
  r_led[6] = B10011001;
  r_led[7] = B10011001;
  for(int j=0; j<=7; j++) {
    digitalWrite(latchPin[0], 0);
    shiftOut(dataPin[0], clockPin[0], MSBFIRST, 128 >> j );
    digitalWrite(latchPin[0], 1);
    digitalWrite(latchPin[1], 0);
    shiftOut(dataPin[1], clockPin[1], MSBFIRST, r_led[j]);
    digitalWrite(latchPin[1], 1);
    digitalWrite(latchPin[2], 0);
```

```
shiftOut(dataPin[2], clockPin[2], MSBFIRST, r_led[j]);
digitalWrite(latchPin[2], 1);
for(int k=0; k <=2; k++) {
  digitalWrite(latchPin[k], 0);
  shiftOut(dataPin[k], clockPin[k], MSBFIRST, 0);
  digitalWrite(latchPin[k], 1); }
}
if(button5.uniquePress()) { goto set; }
if(button3.isPressed()) { bc_l=0; goto bleft; }
if(button4.isPressed()) { bc_r=0; goto bright; }
if(button1.isPressed()) { bc_u=0; goto bup; }
if(button2.isPressed()) { bc_d=0; goto bdown; }
goto set;
```

(3) 4방향(좌-우-위-아래)의 스위치 신호 중, 먼저 입력된 신호를 받아 해당 방향의 라벨문(bleft., bright., bup., bdown:)을 실행한다. 라벨 문에서는 어떤 방향의 스위치 입력을 받던 해당 방향으로 화살표 이미지가 이동할 수 있게 사용자 함수(led_l0, led_r0, led_u0, led_d0)를 사용해 LED 켜기/끄기를 제어한다. 동시에 다른 방향의 스위치 입력도 체크해 신호를 받으면 언제든지 화살표의 방향을 전환시킨다.

```
bleft;
led_l0;
if(button3.uniquePress()) { bc_l++; if(bc_l > 5) { bc_l= 0; }
  if(button4.isPressed()) { bc_r = 5 - bc_l; row_rshift = row_lshift;
    goto bright; }
if(button1.isPressed()) { bc_u = 5 - row_lshift; col_ushift = bc_l;
  goto bup; }
if(button2.isPressed()) { bc_d = row_lshift; col_dshift = bc_l;
  goto bdown; }
if (button5.isPressed()) { goto set; }
-라벨 - bright., bup., bdown: -의 코드는 위와 상응하므로 이하 생략--
void led_l0 {
  for (int a=0; a<5; a++) {
    row_l[a] = 128 >> row_lshift + 1; //128 >> 1
    col_l[a] = 1 << bc_l;
    row_l[1] = 0; col_l[1] = 0;
    row_l[2] = 128 >> row_lshift; // 128 >> 0
    col_l[2] = 1 << bc_l+1;
    row_l[3] = 128 >> row_lshift + 2; //128 >> 2
    col_l[3] = 1 << bc_l+1;
    row_l[4] = 0; col_l[4] = 0;
    led_out(row_l[a], col_l[a]); }
}
-사용자 함수 - led_r0, led_u0, led_d0는 위와 상응하므로 이하 생략-
void led_out(int row, int col) {
  digitalWrite(latchPin[0], LOW);
  shiftOut(dataPin[0], clockPin[0], MSBFIRST, row);
  digitalWrite(latchPin[0], HIGH);
  digitalWrite(latchPin[1], LOW);
  shiftOut(dataPin[1], clockPin[1], MSBFIRST, col); //Red led
  digitalWrite(latchPin[1], HIGH);
  digitalWrite(latchPin[2], LOW);
  shiftOut(dataPin[2], clockPin[2], MSBFIRST, 0);
  digitalWrite(latchPin[2], HIGH);
  prey_led_out();
}
```

(4) prey_led_out() 함수에서는 먹이를 랜덤으로 표시 하고 먹이를 잡는 화살표와 겹치지 않게 체크한다. whichone_chatched() 함수에서는 먹이와 화살표 좌표가 일치하는지를 확인하고, prey_chated() 함수에서는 먹이를 잡았을 때, 먹이와 화살표의 색상을 오

렌지 색상으로 바꾸고 4차례 빠르게 깜박거리게 표시한다.

```

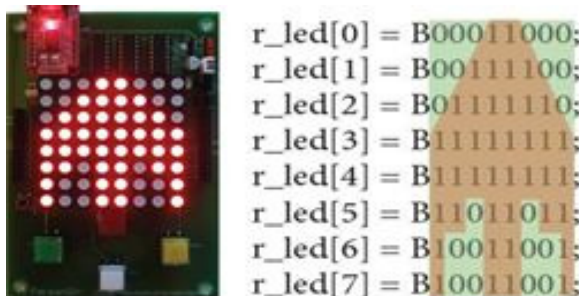
void prey_led_out() {
  for (int i = 0; i < 2; i++) {
    row_p[0] = 1 << random_pre;
    col_p[0] = 1 << random_pre;
    row_p[1] = 0; col_p[1] = 0;
    digitalWrite(latchPin[1], LOW);
    shiftOut(dataPin[1], clockPin[1], MSBFIRST, 0);
    digitalWrite(latchPin[1], HIGH);
    digitalWrite(latchPin[0], LOW);
    shiftOut(dataPin[0], clockPin[0], MSBFIRST, row_p[i]);
    digitalWrite(latchPin[0], HIGH);
    digitalWrite(latchPin[2], LOW);
    shiftOut(dataPin[2], clockPin[2], MSBFIRST, col_p[i]);
    digitalWrite(latchPin[2], HIGH); }
  whichone_catched();
}

void whichone_catched() {
  if (row_p[0] == row_u[0] && col_p[0] == col_u[0]) { prey_catched(); }
  if (row_p[0] == row_r[0] && col_p[0] == col_r[0]) { prey_catched(); }
  if (row_p[0] == row_u[2] && col_p[0] == col_u[0]) { prey_catched(); }
  if (row_p[0] == row_d[2] && col_p[0] == col_d[0]) { prey_catched(); }
}

void prey_catched() {
  for (int i = 0; i < 2; i++) {
    row_p[0] = catch_p_row; col_p[0] = catch_p_col;
    row_p[1] = 0; col_p[1] = 0;
    digitalWrite(latchPin[1], LOW);
    shiftOut(dataPin[1], clockPin[1], MSBFIRST, col_p[i]);
    digitalWrite(latchPin[1], HIGH);
    digitalWrite(latchPin[0], LOW);
    shiftOut(dataPin[0], clockPin[0], MSBFIRST, row_p[i]);
    digitalWrite(latchPin[0], HIGH);
    digitalWrite(latchPin[2], LOW);
    shiftOut(dataPin[2], clockPin[2], MSBFIRST, col_p[i]);
    digitalWrite(latchPin[2], HIGH);
    delay(300); }
  random_pre = random(1, 6);
}

```

둘째, 이미지나 텍스트 생성 코드에서는 사용자가 쉽게 이미지나 텍스트를 만들 수 있게 1바이트 데이터 어레이(1 byte data array)를 생성하였다. 마치 그림을 그리듯 숫자, 1을 넣어 원하는 패턴을 만들고 나머지 패턴이 없는 부분은 0을 입력하면 된다.



[그림 8] 팩맨 로고 이미지와 1바이트 데이터 어레이

4. DIY Pacduino의 사용자 매뉴얼

4.1. 프린트 매뉴얼

(1) 파트 1: 필요한 공구와 준비

키트 조립 전, 필요한 전자 공구 리스트와 사용용도 및 방법을 소개한다.

(2) 파트 2: 전자 부품 리스트 체크

키트 패키지의 전자 부품 리스트를 각각의 이미지로 설명해 사용자가 필요한 부품을 모두 갖췄는지 쉽게 체크 할 수 있다.

(3) 파트 3: 전자 부품 조립도

제공된 PCB 보드의 위치마다 필요한 부품의 이미지를 함께 표기해 사용자가 쉽게 이해할 수 있는 조립도를 제공한다.

(4) 파트 4: 키트 조립하기

Pacduino 키트를 부품별로 조립해 완성하는 방법을 설명한다. 칩 소켓, 핀 소켓, 스위치를 순서대로 납땀하고 12V 파워 서플라이의 DC 소켓을 보드용 파워 연결선으로 교체해 준비한다.

(5) 파트 5: 게임 하기

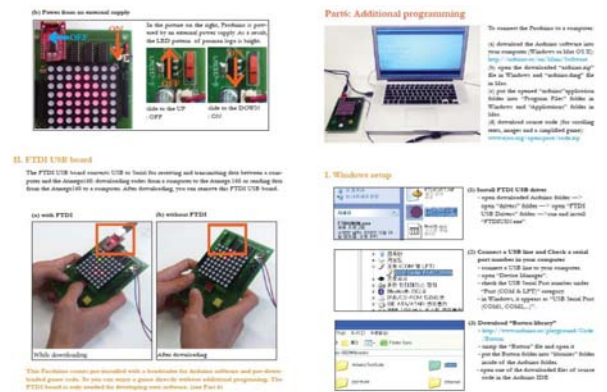
Pacduino 게임 키트를 사용할 때, 컴퓨터 USB와 외부 파워 서플라이 중에서 전원 소스를 선택할 수 있는 스위칭 모드를 설명한다.

(6) 파트 6: 직접 프로그래밍 해보기

아두이노 IDE를 다운받아 설치하는 방법, FTDI 시리얼 컨버터 드라이버 설치하는 방법, 버튼 라이브러리 설치하는 방법을 맥 OS와 윈도우 XP 버전으로 설명하고 프로그래밍시, 아두이노 보드와 시리얼 포트를 지정하는 방법을 설명한다.

(7) 파트 7: Pacduino를 아두이노로 사용하기

마지막 파트에서는 Pacduino 키트를 아두이노 보드로 사용하는 방법을 LED 켜기/끄기 제어 회로를 브레드보드에 구성해 연결하고 제어하는 코드를 예시로 설명한다.



[그림 9] 파트 6: 직접 프로그래밍 해보기

4.2. 온라인 매뉴얼

기본적으로 프린트 매뉴얼과 동일하게 구성되어 있으나 필요한 도구, 부품, 프로그램 라이브러리와 아두이노 등의 사이트와 링크되어 있어 추가 자료를 바로 확인할 수 있고, Pacduino의 전자 회로, 보드와 코드의 자료를 다운로드 받을 수 있다. 각각의 이미지 자료를 확대해 볼 수 있어 프린트 매뉴얼 보다 편리한 장점이 있으며 제공된 시현 비디오를 참조하여 조립에서 프로그래밍까지 전 과정을 초보자도 쉽게 따라할 수 있다.



[그림 10] 온라인 매뉴얼 - 조립 과정 시현 비디오

기술 공유를 위한 매뉴얼 시스템에서는 다양한 범주의 사용자(초보자부터 상급자까지)를 고려해 키트에 대한 사용 설명뿐만 아니라 이용된 소프트웨어와 하드웨어 기술의 자료를 같이 제공하여 적용된 기술의 응용과정을 이해할 수 있도록 함으로써 사용자가 기술 재사용하고 재개발할 수 있도록 촉진한다는 측면에서 효과적이라 할 수 있다. 매뉴얼의 매체 선택에서는 온라인 매뉴얼이 프린트 매뉴얼의 텍스트 설명뿐만 아니라, 비디오, 부가 자료의 링크, 선명한 이미지를 제공한다는 점에서 사용자의 접근이 용이하다.

5. 사용자와의 소통

5.1. 워크숍

워크숍은 퍼지컬 컴퓨팅 기술의 숙련도에 따라 초보자와 중급자 그룹으로 나누어 진행되었다. 먼저 초보자를 대상으로 한 퓨처 에브리씽 페스티벌과 크리에이티브 프로젝트의 워크숍에서는 상세한 전자 회로나 코드에 대한 설명보다는 참가자가 생소하고 어렵게만 느끼던 퍼지컬 컴퓨팅을 이용해 직접 키트를 만들어 봄으로써 퍼지컬 컴퓨팅이란 무엇이며 아이디어에 어떻게 접목해 프로젝트로 완성하는지에 대한 전반적인 과정을 이해할 수 있도록 하였다. 반면 퍼지컬 컴퓨팅 경험이 있는 중급자 그룹을 대상으로 한

프즈테크와 빅토리아 앤 앨버트 미술관의 워크숍에서는 Pacduino 키트에 적용된 LED 매트릭스 전자 제어 회로 원리와 구성 및 프로그래밍 코드의 응용원리를 주로 설명하고 참가자가 Pacduino에 쓰인 기술을 응용하여 발전시키고자 하는 아이디어에 대한 토론으로 이루어졌다.



[그림 11] 참가자가 Pacduino 키트 조립 후, 프로그래밍 테스트를 하는 과정 (퓨처 에브리씽 페스티벌과 빅토리아 앤 앨버트 미술관의 워크숍)

또한 워크숍 참가자들이 제안한 프린트 매뉴얼에 대한 의견과 조립과정을 보여주는 온라인 시현 비디오에 대한 의견은 개발과정에 반영되었다.

워크숍을 통한 소통은 참가자의 실질적인 요구나 필요 사항을 바로 접할 수 있어 하드웨어 개발, 아이디어 확장과 공유를 위한 시스템 개발에 바로 적용 가능하여 활용도가 높고, 참가자들도 기술정보나 아이디어의 공유과정에서의 문제점을 바로 해결할 수 있다는 점에서 적극적인 소통 및 공유의 방식이지만 물리적 공간의 제약과 1회성이라는 시간의 제한이 따르는 단점이 있었다.

5.2. 블로그

워크숍의 시. 공간의 제약이라는 단점을 고려해 참가자들과의 꾸준한 기술 및 아이디어를 공유하기 위해 구성된 블로그에서는 워크숍에 참가하지 않았지만 퍼지컬 컴퓨팅이나 오픈소스 하드웨어에 관심이 있는 일반 온라인 사용자에서부터 임의로 방문한 사용자, 블로그 만들기에 관심이 있는 사용자까지 Pacduino 블로그의 자료를 이용, 공유하며 다양한 소통이 이루어졌다. 비단 공개된 게임 키트에 대한 특정 기술 정보의 공유에만 그치지 않고 정보를 공유하는 시스템 디자인에 대한 여러 의견을 주었다. 이는 사용자와 기술 및 아이디어를 공유와 더불어 컴퓨팅, 엔지니어 기술, 디자인의 융합으로 파생되고 있는 새로운 트렌드에 대한 높은 관심과 정보를 공유하는 방식에 대한 연구가 필요하다는 사실을 입증하였다. 블로그라는 웹 공간은 워크숍과 달리 물리적, 시간적 제약이 없어 불특정 다수 사용자와의 개발자간의 소통, 사용자와 사용자간의 소통이 가능한 장점이 있고 방대한 공

유 형태를 이루지만 누구나 접근이 가능해 정확하지 않은 정보를 포스팅하는 불편함을 발견했다.

6. 결론

고바야시 시게루에 의하면 만들기는 무엇을 만들까(what to make)와 어떻게 만들까(How to make)의 두 가지로 구성되며 무엇을 만들까는 갑자기 생겨나는 것이 아니고 아이디어는 처음부터 명확하고 완벽하기 보다는 애매하고 어설픈 것이며 만들어가는 도중에 점점 다듬고 정리하여 완성해 나가는 것이라고 한다(고바야시 시게루, 2012). 다시 말해서 미디어 아트, 디자인은 전문 기술의 이해와 응용을 바탕으로 끊임 없는 아이디어의 프로토타이핑 실험이 뒷받침되어야 하며 비로소 그 과정에서 온전하고 단단한 아이디어를 구축해 나갈 수 있다. 이에 따라 아티스트와 디자이너를 위한 저작 도구들이 생겨났으며 공개된 기술과 온라인 커뮤니티의 토론을 통해 기술적 문제를 해결하고 원천적인 아이디어를 공유함으로써 다른 사용자가 재사용하여 창의적인 프로젝트로 발전시킬 수 있도록 하는 새로운 저작 환경이 구축되고 있다. 이러한 저작 환경의 변화와 아티스트와 디자이너의 역할 확장을 토대로 DIY Pacduino 키트를 개발할 수 있게 되었다. 첫째, 피지컬 컴퓨팅의 기본 전자 소자인 LED 매트릭스와 스위치를 제어해 대중적인 팩맨 게임을 접목시킴으로써 보다 쉽게 아이디어와 하드웨어 기술의 적용을 습득할 수 있도록 하였다. 둘째, 샘플 코드로 팩맨 게임을 즐길 수도 있지만, 이를 응용해 사용자 스스로 게임 코드를 작성할 수도 있게 하였고 마치 그림을 그리듯이 숫자 1과 0을 이용해 이미지나 텍스트를 쉽게 만들 수 있도록 코드를 간편화하였다. 셋째, 기술과 아이디어를 전달하고 공유하기 위해 프린트와 온라인 매뉴얼 방식을 모두 채택하고, 사진과 시현 비디오로 이루어진 단계별 설명으로 사용자가 키트를 완성, 활용할 수 있게 하였다. 넷째, 위크숍과 온라인 블로그를 통해 위크숍 참여자에서부터 광범위한 일반 온라인 사용자까지 블로그의 오픈소스 자료를 이용하고 자유롭게 소통할 수 있도록 하였다.

미디어 아티스트나 디자이너들이 완성된 프로젝트로 소통하는 방식에서 더 나아가 개발 과정에서의 기술과 아이디어를 함께 공유할 때 예상하지 못한 방식으로 응용, 변형되어 새로운 프로젝트로 탄생될 수 있으며, 이는 관람자나 사용자가 수동적인 향유자에서 적극적인 창조자로 확장시켜 미디어 아트, 디자인의 다변화와 대중화를 이룰 수 있을 것이다. 또한 사용자의 창의적 개입의 가능성은 아티스트와 디자이너

에게도 프로젝트를 심도 있게 발전시킬 수 있는 계기를 제공해 줄 것이다. 초기의 오픈소스 운동이 기술의 향유에만 포커스를 맞췄다면 이젠 아이디어와 동시에 소통의 방식에 대해 생각하고 고민해야만 대중을 보다 긍정적으로 포용할 수 있고 아티스트, 디자이너에게도 새로운 기회가 되리라 생각한다.

참고문헌

- 진중권 (2012). 과학예술의 쟁점들. 엘리스온, 컬럼, 2012.10.3, <http://aliceon.tistory.com/2052>
- Flusser, V. (1993). Lob der Oberflächlichkeit: Fureine Phänomenologie der Medien. 김성재 역 (2004). 『피상성 예찬』. 서울 : 커뮤니케이션북스
- Flusser, V. (1998). Kommunikologie. 김성재 역 (2001). 『코뮤니콜로지』. 서울 : 커뮤니케이션북스
- Shigeru, G. (2010). Prototyping Lab. 서효정 역 (2012). 『프로토타이핑 랩』. 서울 : 인사이트.
- Banzi, M. (2008). Getting Started with Arduino. 이호민 역 (2010). 『손에 잡히는 아두이노』. 서울 : 인사이트.
- Anderson, C. (2006). The Long Tail: why the Future Business Is Selling Less of More. 이노무브 그룹 역 (2006). 『롱테일 경제학』. 서울 : 랜덤하우스코리아.
- Anderson, C. (2009). FREE: The Future of a Radical Price. 정준희 역 (2009). 『Free; 프리』. 서울 : 랜덤하우스코리아.
- Anderson, C. (2012). 『Makers - The New Industrial Revolution』. New York: Random House New York.
- Creative Commons. (2012.11.28.), <http://creativecommons.org/licenses>
- Botanicalls. (2012.12.3.), <http://www.botanicalls.com>
- Arduino. (2012.12.3), <http://arduino.cc>
- Hernando, B. (2004). Wiring. (2012.12.4.), <http://wiring.org.co>
- Limor, F. (2003). Project. (2012.12.7.), <http://www.ladyada.net/make>
- O'Reilly Publishing. (n.d.). Make.. (2012.12.7), <http://makezine.com>
- about EyeWriter. (2012.12.7), <http://eyewriter.org>
- EunJoo, S. (2011). DIY Pacduino. (2012.12.22.), <http://www.ejoo.org/openspace>