

정보통신기술 교육을 위한 피지컬 컴퓨팅 학습모델

이용진

한국교육대학교 기술교육과

Physical Computing Learning Model for Information and Communication Education

Yong-Jin Lee

Korea National University of Education

요 약 본 논문은 기술교육 및 공학교육에서 정보통신기술을 가르칠 때 적용 가능한 피지컬 컴퓨팅 학습모델을 제안한다. 이 모델은 피지컬 컴퓨팅에 기반을 두고 정보의 생성과 정보의 전송을 하나의 프레임 안에서 다루는 것으로 학생들에게 정보와 통신에 대한 종합적인 이해와 실습을 제공한다. 본 연구에서 제안한 학습모델은 클라이언트-서버 구조에 기반을 둔 학습모델과 웹에 기반을 둔 학습모델로 나누어진다. 실제 구현된 학습모델에서 정보의 획득과 제어는 아두이노의 스케치를 통해 이루어지며 정보의 통신은 최근 교육용 플랫폼으로 잘 알려진 라즈베리파이의 파이썬 소켓을 이용하여 수행된다. 제안한 학습 모델은 학생들에게 전 세계적인 정보 제어와 통신을 가능하게 해주는 사물인터넷의 개념을 가르치는 데 사용될 수 있다.

주제어 : 정보통신 학습모델, 피지컬 컴퓨팅, 파이썬, 사물 인터넷

Abstract This paper aims to present the physical computing learning model applicable in teaching the information and communication technology for technology and engineering education. This model is based on the physical computing and deals with the information creation and information transfer in one framework, thus provides students with the total understanding and practice opportunity about information and communication. The proposed learning models are classified into the client-server based model and the web based model. In the implemented learning model, the acquirement and control of information is performed by sketch on Arduino and the communication of information is performed by the Python socket on Raspberry Pi well known as an education platform. Our proposed learning model can be used for teaching students to understand the concept of Internet of Things (IoT), which provides us with world wide control and communication of information.

Key Words : Information and communication learning model, Physical computing, Python, IoT

1. 서론

학교 현장에서 정보통신을 교육하기 위한 기본적인 학습 모델을 설정하는 데 있어서 중요한 사항은 정보를 생성하는 것과 정보를 전달하는 기법을 학생들에게 이해

시키되 두 가지를 균형 있게 다루는 것이다.

이러한 학습모델을 실현하기 위해서는 하드웨어 교육과 소프트웨어 교육이 모두 필요하다. 그러나 기존의 정보통신 교육에서는 소프트웨어를 수반하지 않는 하드웨어 교육만을 하거나 또는 하드웨어를 수반하지 않는 소

소프트웨어 교육만이 이루어졌기 때문에 학생들이 정보통신에 대한 전체적인 개념을 이해하는 데 한계를 보여 왔다.

피지컬 컴퓨팅(Physical Computing)이란 하드웨어와 소프트웨어를 동시에 사용함으로써 인간과 물리적 장치 사이의 상호 작용을 지원하는 개념[1]으로 최근 기술교육이나 공학 교육에서 그 적용이 활발하게 이루어지고 있다.

피지컬 컴퓨팅을 교육에 활용하게 되면 간단한 소프트웨어 프로그래밍을 통해 전기전자 부품과 센서 등을 제어할 수 있고 데이터(정보)를 얻을 수 있기 때문에 최근에 핫 이슈가 되고 있는 사물 인터넷에 대한 개념도 실습을 통해 습득할 수 있게 된다[2,3].

피지컬 컴퓨팅을 교육하기 위한 대표적인 플랫폼으로 아두이노, 갈릴레오, 그리고 라즈베리 파이 등이 이미 시장에서 판매되고 있는 데 교육 현장에서는 이 중에서도 아두이노 우노(R3)가 가장 많이 사용되고 있다.

아두이노[4,5]는 인터프리터와 운영체제가 필요 없는 매우 간단한 장비이다. C와 C++에 기반한 스케치 코드는 기계어로 컴파일 되어 칩에 실리게 되며 아두이노 내부에서 작동된다. 아두이노는 소형 제어장치(micro controller)가 외부의 기기(hardware)를 직접 제어할 수 있으므로 제어 중심에 적합한 장치이다. 즉 센서 등의 부품으로부터 데이터를 얻어 정보를 생성하고 그것을 직접 제어하는 데 적합한 장비이다. 그러나 통신 입장에서 볼 때 아두이노는 주로 시리얼 통신에 중점을 둔다. 따라서 사물 인터넷과 같이 전 세계적인 통신을 할 때 필요한 블루투스나 와이파이 그리고 이더넷 통신이 내장되지 않은 단점과 추가 실드를 사용할 때의 비용 및 코딩의 복잡함이 문제점으로 존재한다.

인텔의 갈릴레오[6]는 임베디드 시스템에 주로 사용되는 Yocto linux가 장착되어 있는 본격적인 컴퓨팅 시스템이다. 따라서 C언어와 파이썬 등의 컴퓨터 언어 등을 사용할 수 있고 아두이노와 동일한 핀 배열을 갖는 편리함이 있지만 와이파이나 블루투스 등의 통신을 위해서는 추가적인 부품이 필요하다.

라즈베리파이[7,8]는 리눅스에 기반을 둔 오픈 운영체제(라즈비안 등)를 장착한 프로세서(processor)로 파이썬, 스크래치, C언어 등의 컴퓨터 언어 등을 사용할 수 있다. 최근의 라즈베리파이 3은 이더넷은 물론 와이파이와 블루투스가 설치되어 있으므로 TCP/IP 소켓을 통해 전 세계적인 통신이 가능하다. 특히 내장된 파이썬은 C

언어에 비해 코드가 매우 간결하고 배우기 쉽기 때문에 교육용 언어로 적합하다. 아울러 센서 등을 제어하기 위한 GPIO (General Purpose Input/Output)가 설치되어 있으므로 독자적인 피지컬 컴퓨팅이 가능하며 교육용 설계를 적용하였기 때문에 그래픽과 게임 등도 가능하다. 하지만 이러한 범용성으로 인해 아두이노의 장점인 제어 중심의 컨트롤러 역할은 미흡하다. 또한 GPIO에서는 스케치 언어가 직접 실행될 수 없기 때문에 기존에 스케치 언어에 익숙한 학생들이 C언어나 파이썬 등의 언어를 사용해서 제어해야 하는 점도 단점이 될 수 있다.

이상에서 살펴본 바와 같이 아두이노는 정보의 획득에 강점이 있고, 라즈베리파이는 통신에 강점이 있기 때문에 균형 있는 정보 통신 교육을 위해서는 이 두 개의 장비를 융합해서 사용하는 것이 바람직하다. 따라서 본 연구에서는 정보통신 교육을 하는 데 있어서 아두이노와 라즈베리파이를 혼합해서 정보와 통신 교육을 하기 위한 학습모델들을 제안하고자 한다. 이 모델은 학생들에게 사물 인터넷을 교육하는 데도 이용가능하다.

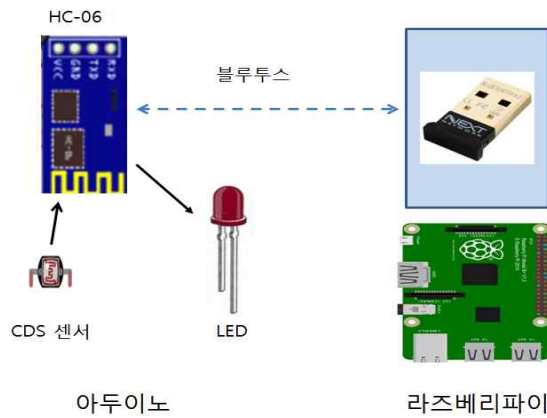
본 논문의 2장에서는 클라이언트와 서버를 이용하는 학습모델을 제시하고 3장에서는 웹기반의 학습모델을 제시하며 4장에서는 결론을 맺는다.

2. 클라이언트-서버 학습모델

아두이노는 컴퓨터와 시리얼 통신방식을 통해 정보를 주고받는다. 유선 통신은 UART, I2C, SPI 등의 방식으로 이루어지고 무선 통신은 블루투스가 주로 사용된다. 본 연구에서는 아두이노에서 읽은 CDS 센서 값을 블루투스 모듈(HC-06)을 통해 라즈베리파이의 블루투스로 전송하는 시리얼 통신을 고려한다. 라즈베리파이에서는 파이썬 블루에즈 모듈[9]을 설치한 후 블루투스 소켓모듈을 이용한다.

블루투스를 이용한 시리얼 통신의 학습모델은 [Fig. 1]과 같다.

먼저 아두이노에서 CDS 센서 값을 읽고 HC-06을 사용하여 블루투스 통신을 통해 라즈베리파이로 전송하는 스케치 코드의 일부는 [Fig. 2]와 같다. 먼저 블루투스를 초기화(bt.begin)하고 CDS로부터 아날로그 값을 읽어서 라즈베리파이로 전송한다(bt.println(message)). 다음에 라즈베리파이로부터 수신(bt.read)한 후, 값이 1이면 LED 9번 핀을 켜고 0이면 끈다.



[Fig. 1] 블루투스를 이용한 시리얼 학습모델

이 스케치 코드는 앞으로 모든 학습모델에서 공통적으로 사용된다.

```
#include <SoftwareSerial.h>
SoftwareSerial bt (2,3); // 송신 2번핀, 수신 3번핀
int LEDPin = 9; // LED 핀
int lightPin = A0; // CDS
int btdata;

void setup()
{
  Serial.begin(9600);
  bt.begin(9600);
  pinMode (LEDPin, OUTPUT);
}

void loop()
{
  int rate;
  String message = "";
  rate = analogRead(lightPin);
  message = message + rate;

  // send the CDS value
  bt.println(message);
  fflush(stdout);

  if (bt.available()) {
    btdata = bt.read();
    if (btdata == '1')
      digitalWrite (LEDPin, HIGH);
    if (btdata == '0')
      digitalWrite (LEDPin, LOW);
  }
  delay (5000); //prepare for data
}
```

[Fig. 2] 블루투스를 이용한 아두이노 전송

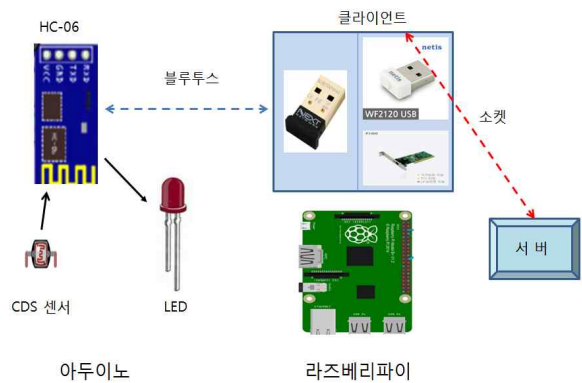
라즈베리파이의 수신코드는 [Fig. 3]과 같다. 먼저 블루투스 모듈을 임포트하고 bd_addr에 아두이노에 설치된 HC-06 블루투스 주소를 할당한다. 블루투스 소켓 (bluetooth.BluetoothSocket)을 만든 후에 아두이노에 연결(sock.connect)하고 읽어 들인 CDS 값이 400보다 작으면

아두이노로 '1'을 전송(sock.send(tosend))하여 LED의 불을 켜고, 그렇지 않으면 '0'을 전송하여 LED의 불을 끈다.

```
#!/usr/bin/env python
# -*- coding: UTF-8 -*-
import bluetooth
import time
bd_addr="20:15:12:22:88:00" #HC-06: 블루투스 주소
port = 1
sock = bluetooth.BluetoothSocket(
    bluetooth.RFCOMM)
sock.connect((bd_addr,port))
print ("server start !!")
while 1:
    print(value)
    if value < 400:
        tosend = '1'
    else:
        tosend = '0'
    if tosend != 'q':
        if tosend == '1' or tosend == '0':
            sock.send(tosend)
            time.sleep(5.0)
            message = sock.recv(1024)
            print (message)
            value = int(message)
        else:
            break
sock.close()
```

[Fig. 3] 라즈베리파이의 블루투스 수신코드

이제 라즈베리파이에서 임의의 서버로 정보를 전송하는 경우의 클라이언트-서버 학습모델을 제안한다. 이 모델은 [Fig. 4]와 같이 표현된다.



[Fig. 4] 클라이언트-서버 기반 학습모델

이 학습 모델에서는 먼저 [Fig. 3]의 코드가 클라이언트 코드로 변환되어야 한다. 즉 아두이노로부터의 정보는 블루투스를 통해 수신하고 이를 와이파이나 인터넷을 통해 파이썬 소켓 통신을 사용하여 서버로 전송한다.

[Fig. 5]는 [Fig. 3]에서 추가되어야 할 부분을 보여준다. 즉 TCP용 소켓(s)을 만들고(socket.socket) 서버

(117.17.204.120)의 8900번에 접속한 후에(s.connect) 아두이노로부터 들어온 메시지를 전송한다(s.sendall).

```
import socket
#-----이더넷 또는 와이파이 -----
HOST = '117.17.204.120'
PORT = 8900
s = socket.socket(socket.AF_INET,
                  socket.SOCK_STREAM)
s.connect((HOST, PORT))
while 1:
    if value < 400:
        tosend = '1'
    else:
        tosend = '0'
    if tosend != 'q':
        if tosend == '1' or tosend == '0':
            sock.send(tosend)
            time.sleep(5.0)
            message = sock.recv(1024)
            value = int(message)
            # send the data to server
            s.sendall(message)
        else:
            break
sock.close()
s.close()
```

[Fig. 5] 라즈베리파이의 클라이언트 코드

서버측 코드는 [Fig. 6]과 같다. 먼저 소켓을 만든 후 (socket.socket)에 로컬 주소를 바인드(s.bind)하고 포트에 주의를 기울인다(listen). 클라이언트로부터의 연결을 허용하고(s.accept) 아두이노를 거쳐 클라이언트로부터 전송되어온 CDS 조도 값을 수신한다(conn.recv).

```
#!/usr/bin/env python
# -*- coding: UTF-8 -*-
import socket
HOST = ''
PORT = 8900
s = socket.socket(socket.AF_INET,
                  socket.SOCK_STREAM)
s.bind((HOST, PORT))
s.listen(1)
conn, addr = s.accept()
try:
    while 1:
        data = conn.recv(1024)
        if not data: break
        print (data) # CDS 조도값 출력
    conn.close()
```

[Fig. 6] 라즈베리파이의 서버측 코드

3. 웹기반 정보통신 학습모델

2장에서 제시한 클라이언트와 서버 기반 학습 모델은 정보의 전달과 보안이라는 측면에서 효율적이지만 임의

의 장소에서 라즈베리파이를 통해 아두이노를 제어하는 경우에는 불편한 점이 존재한다. 왜냐하면 서버 컴퓨터에 접속하기 위해 특정한 클라이언트 프로그램을 보유해야 하기 때문이다. 따라서 임의의 웹 브라우저를 사용하는 웹 기반 모델이 필요하다. 즉 라즈베리파이에 웹 서버를 구축하고 웹 브라우저를 이용하여 접속하게 되면 전 지구적인 통신과 제어가 가능해진다.

파이썬에서는 웹을 위한 마이크로 프레임워크로 플라스크[10]를 지원하므로 /home/pi/web_falsk_control 디렉터리에 플라스크를 설치한다. [Fig. 7]은 웹 플라스크를 사용한 정보통신 학습모델을 보여준다.



[Fig. 7] 웹 기반 학습모델

이 학습모델에서 아두이노의 스케치 프로그램은 [Fig. 2]의 코드와 같다. 라즈베리파이에는 블루투스 모듈과 와이파이(또는 이더넷) 모듈을 이용하여 웹 서버 프로그램 (webserver.py)을 운영해야 한다. 따라서 /home/pi/web_flask_control 디렉터리 아래에 webserver.py를 설치한다. 한편 웹 브라우저에게는 홈페이지를 제공해야 하므로 /home/pi/web_flask_control/templates 디렉터리 아래에 index.html 파일을 설치한다. index.html은 [Fig. 8]에 나타나있고 웹 브라우저에서 홈 페이지에 접속한 모습은 [Fig. 9]와 같다.

```
<!DOCTYPE html>
<html>
<head>
<title> KNUE Technology </title>
</head>
<body>
<h2> Light Sensor Reading { { value } }!</h2>
<form action="/" method="post">
<p><input type="submit" name="submit"
value="Turn On"> </p>
<p><input type="submit" name="submit"
value="Turn Off"> </p>
</form>
</body>
</html>
```

[Fig. 8] index.html



[Fig. 9] 웹 브라우저에서 접속한 모습

다음으로 웹 서버 프로그램의 개략 코드는 [Fig. 10]과 같다. 먼저 웹을 이용하기 위해 플라스크 프레임워크를 임포트해주고 아두이노와의 통신을 위해 블루투스 모듈을 임포트한다. 아두이노에서 읽은 CDS 조도 값을 블루투스를 통해 수신(sock.recv())하고, 이 값을 index.html을 통해 전달한다(render_template('index.html', value=100*(readval/1023)). 사용자가 turn on 버튼을 누르면 블루투스를 통해 '1'을 전송하여 아두이노로 하여금 LED를 키도록 하고 turn off 버튼을 누르면 '0'을 전송하여 LED를 끄도록 한다.

```

#--- webservice.py ---
from flask import Flask, render_template, request,
redirect, url_for
import time
import bluetooth
#----- 블루투스 연결 -----
# [Fig. 3] 참조
#-----
app = Flask(__name__)
time.sleep(3)
@app.route('/', methods = ['POST','GET'])
def hello_world():
    # templates/index.html
    if request.method == 'POST':
        # 홈 페이지에서 turn on 버튼을 누른 경우
        if request.form['submit'] == 'Turn On':
            # 아두이노에게 LED를 키도록 명령 보냄
            tosend = '1'
            sock.send(tosend)
        # 홈 페이지에서 turn off 버튼을 누른 경우
        elif request.form['submit'] == 'Turn Off':
            # 아두이노에게 LED를 끄도록 명령 보냄
            tosend = '0'
            sock.send(tosend)
        else:
            pass
    # 아두이노의 CDS 조도 값을 수신
    readval = int (sock.recv(1024))
    # CDS 조도 값을 홈페이지로 반환
    return render_template('index.html',
        value=100*(readval/1023.))
if __name__ == "__main__":
    # 홈 페이지 구동
    app.run(host='0.0.0.0', port = 8000)
  
```

[Fig. 10] webservice.py

4. 결론

본 논문에서는 학교 현장에서 정보통신기술을 가르칠 때 적용가능한 단계적인 학습모델을 제안하였다. 이 모델은 현존하는 피지컬 컴퓨팅 플랫폼인 아두이노와 라즈베리파이를 혼합하여 개발되었다. 아두이노는 정보의 획득과 제어의 측면에서는 장점이 있지만 정보의 전송 측면에서는 부족한 점이 있고, 라즈베리파이는 정보의 전송에는 장점이 있지만 정보의 제어 측면에서는 부족한 점이 있다. 따라서 제안한 학습모델은 아두이노의 정보 획득과 제어기능 그리고 라즈베리파이의 정보 통신기능을 조합하는 혼합형 플랫폼 위에서 동작한다. 아두이노와 라즈베리파이 사이의 통신 기능은 블루투스를 이용하고 라즈베리파이와 다른 컴퓨터 사이의 통신에는 파이썬 소켓을 사용함으로써 최소한의 하드웨어 구성과 내재된 모듈을 이용할 수 있도록 구성하였다. 제안된 학습모델은 클라이언트-서버 기반과 파이썬 플라스크를 이용하는 웹 기반으로 나누어 개발되었다. 이 학습모델을 통해 학생들은 정보와 통신을 하나의 프레임워크 내에서 이해할 수 있고 사물 인터넷에 대한 기초 개념을 습득할 수 있게 된다. 앞으로의 연구에서는 실제 학교 현장의 학생들과 교수자의 반응을 조사하여 제안한 학습 모델을 보다 교실 친화적으로 개선하는 일과 모델의 성능을 정량적으로 평가하는 일이 필요하다.

REFERENCES

- [1] D. Surlivan, T. Igoe Physical Computing: Sensing and Controlling the Physical World with Computers, Course Technology Press, 2004.
- [2] J. Marquez, J. Villanueva, Z. Solarte, and A. Garcia, IoT in Education: Integration of Objects with Virtual Academic Communities, New Advances in Information Systems and Technologies, 201-212, 2016.
- [3] C. Bulla, B. Hunshai, and S. Mehta, "Adoption of Cloud Computing in Education System: A Survey," International Journal of Engineering Science and Computing, Vol. 6, No. 2, pp. 6375-6380, 2016.
- [4] Arduino, "Arduino open-source prototyping platform," <http://www.arduino.cc>, 2016.
- [5] H. Cheng, L. Hao, Z. Luo, and F. Wang, "Establishing the Connection between Control Theory Education and Application: An Arduino Based Rapid Control Prototyping

- Approach," International Journal of Learning and Teaching, Vol. 2, No. 1, pp.67-72, 2016.
- [6] A. Kurniawan, Getting Started with Intel IoT and Intel Galileo, Berlin, October, 2014.
- [7] The Raspberry Pi Foundation, Raspberry Pi, <http://www.raspberrypi.org>, 2016.
- [8] J. Sobota, R. Pisl, P. Balda, M. Schlegel, "Raspberry Pi and Arduino boards in control education," The Proc. of 10th IFAC Symposium Advances in Control Education," August 28-30, pp. 7-12, Sheffield, UK, 2013.
- [9] A. Huang, Introduction to Bluetooth Programming," <https://people.csail.mit.edu/albert/bluez-intro/c212.html>, 2008.
- [10] Flask, Web development, one drop at a time, <http://flask.pocoo.org/>

이 용 진(Yong-Jin Lee)

[정회원]



- 1995년 2월 : 고려대학교 전산학과 (이학박사)
- 1995년 3월 ~ 2005년 8월 : 우송대학교 컴퓨터과학과 교수
- 2005년 9월 ~ 현재 : 한국교원대학교 기술교육과 교수

<관심분야>

사물인터넷, 이동통신, 정보통신교육