Smart Media Journal / Vol.9, No.3 / ISSN:2287-1322
https://dx.doi.org/10.30693/SMJ.2020.9.2.116

오픈소스 하드웨어와 이벤트 기반 논 블로킹 I/O 알고리즘을 활용한 음성송출 시스템 설계 및 구현

(Design and implementation of Voice Transmission System using Open Source Hardware and Event based Non-Blocking I/O Algorithm)

김형우*, 이현동**
(HyungWoo Kim, Hyun Dong Lee)

요 약

Digital Information Display와 KIOSK는 전용 컨텐츠의 개발 비용으로 인한 초기 도입 비용 및 유지 비용과 제품의 특성으로 인해 설치 비용이 높다는 문제가 있다. 이러한 문제를 해결하기 위해 오픈소스 하드웨어 및 이벤트 기반 논 블로킹 I/O 알고리즘을 사용하여 음성 전송 시스템을 설계하고 구현하였다. 제안하는 오픈 하드웨어를 통한 음성송출 시스템은 시스템 초기 도입 비용과 유지 보수비용이 저렴하고, 다양한 형태로 활용할 수 있어서 정보 취약 계층의 정보에 대한 접근성을 향상할 수 있다.

■ 중심어 : 논 블로킹 입출력 ; 자동 음성 재생 ; 오픈소스 하드웨어 ; 미들웨어

Abstract

Digital Information Display and KIOSK have a problem that initial introduction cost and maintenance cost due to the development cost of dedicated contents and installation cost are high due to the characteristics of the product. In order to solve these problems, We designed and implemented of voice transmission system using Open Source Hardware and Event based Non-Blocking I/O Algorithm.

■ keywords : Non-Blocking I/O ; Auto Sound Player ; Open Source Hardware ; Middleware

Ⅰ. 서 론

현재 수많은 정보 제공 서비스들은 사용자가 관심을 가지고 스마트 디바이스를 직접 조작을 하지 않으면 정보를 얻을 수 없는 형태로 서비스가 제공되고 있으며, 공공 정보 제공을 위한 DID(Digital Information Display)나 KIOSK는 제품의 특성상 전용 콘텐츠 개발 비용 및 설치 비용으로 인한 초기 도입 비용과 유지보수 비용이 높다는 문제가 있다. 음성 및 사운드를 이용하여 정보 취약 계층의 정보에 대한 접근이 쉬우며, 오픈소스 하드웨어를 통한 시스템 구축으로 시스템 초기 도입 비용과 유지 보수비용이 저렴하고, 다양한 형태로 활용할 수 있는 새로운 형태의 음성송출 시스템을 설계 및 구현하였다.

제안하는 오픈소스 하드웨어와 이벤트 기반 논 블로킹 I/O 알고리즘을 활용한 음성송출 시스템은 외국인이 많이 거주하는 지역, 버스 정류장, 공원, 공공 화장실 등 유동인구가 적어 지자체의 예산으로 범죄 예방 활동이 어려운 공공장소 등에 설치하여 적은 설치 및 유지보수 비용으로 행정 정보 및 보안 정보 등을 제공할 수 있다.

본 논문의 구성은 다음과 같다. 2장 관련 연구에서는 오픈소스 하드웨어 및 비동기 논 블로킹 I/O 알고리즘을 살펴보고, 3장에서는 제안하는 음성송출 시스템을 살펴보고 평가한다. 마지막으로 4장에서는 결론 및 향후 연구 방향을 제시한다.

Ⅱ. 관련연구

1. 오픈소스 하드웨어 기술

오픈소스 하드웨어는 누구나 하드웨어를 배우고, 수정하고, 배포하고, 제조하고 팔 수 있는 공개된 하드웨어이다. 하드웨어를 만들기 위한 디자인 소스는 그것을 수정하기에 적합한 형태로 구할 수 있어야 한다. 오픈소스 하드웨어는 각 개인들이 하드웨어를 만들고 이 하드웨어의 사용을 극대화하기 위하여, 쉽게 구할 수 있는 부품과 재료, 표준 가공 방법, 개방된 시설, 제약이 없는 콘텐츠 그리고 오픈소스 디자인 툴을 사용하는 것이

* 정회원, 동서대학교 디자인학부
** 정회원, 동서대학교 산학협력단
이 논문은 2020년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. 2020R1I1A1A01072717)
접수일자 : 2020년 08월 10일
게재확정일 : 2020년 09월 21일
교신저자 : 이현동 e-mail : win4class@hanmail.net

이상적이다. 오픈소스 하드웨어는 디자인을 자유롭게 교환함으로써 지식을 공유하고 상용화를 장려하여 사람들이 자유롭게 기술을 제어할 수 있도록 한다[1-3].

실제로 2014년 약 400억 달러에서, 2017년까지 900억에 가까운 시장을 형성 하기까지 한 오픈소스 하드웨어는 확실히 성장률 부분에서 크게 확대가 되고 있음을 파악할 수 있다. IoT 세상인 요즘은 오픈소스 하드웨어가 활용되는 분야가 크게 확대되고 있어, 더욱이 오픈소스 하드웨어의 중요성이 강조되고 있다[4-7]. 그림1은 전세계 오픈소스 하드웨어 시장규모를 나타낸다.

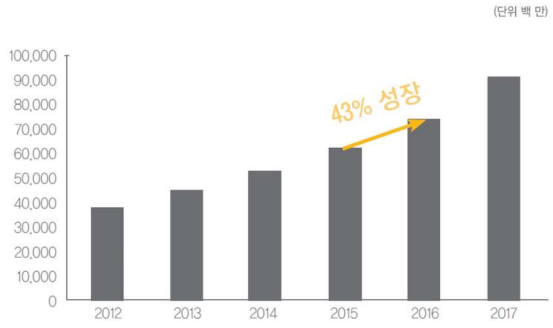

그림 1. 전세계 오픈소스 하드웨어 시장 규모

2. 비동기 논 블로킹 I/O 알고리즘

전통적인 프로그래밍 방식인 동기식 I/O 방식과 비동기식 I/O 방식의 데이터 처리에 대하여 테스트를 진행하기 위한 컴퓨터의 사양은 Intel i7 3770 CPU, RAM 16G, SSD 256G이며, 데이터 처리는 난수를 발생시켜 5번 반복 실행하는 방식으로 처리하였다. 이 부분을 하나의 프로세스로 5번 반복하는 방식과 5개의 프로세스를 시작 시 시작 이벤트와 프로세스 완료 시 완료 이벤트를 확인하여 총 5번 완료 이벤트가 발생 했을 경우 모든 프로세스를 완료하는 방식으로 나누어 테스트를 진행하였으며, 그 테스트의 결과는 아래 그림2와 같다.

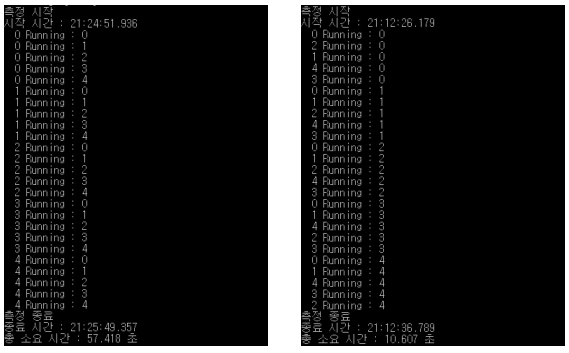

그림 2. 동기식 I/O 방식과 비동기식 I/O 방식 비교 시험

동기식 I/O 방식은 전통적인 형태의 데이터 처리 방식으로 하나의 프로세스가 하나의 작업을 완료할 때까지 다른 데이터 처리를 기다리고 있다가 데이터 처리가 끝나면 다음 데이터를 처리하는 방식이다. 비동기 I/O 방식은 데이터 처리를 시작할 때 시작 이벤트를 던지고, 다음 데이터 처리를 시작하면서 시작 이벤트를 발생시킨다.

하나의 데이터 처리가 완료되면, 데이터 처리 완료 이벤트가 발생하고, 모든 데이터 처리 완료 이벤트가 종료되었을 때 프로그램이 완료된다.

그림3의 결과를 보면 동기식 I/O 방식은 하나씩 순차적으로 데이터를 처리하고 있기 때문에 모든 처리가 완료될 때까지 총 25초 걸리지만, 비동기식 I/O 방식을 사용할 경우는 거의 같은 시간에 5개의 프로세스 시작 이벤트가 발생하기에 시작 순서와 수행 순서도 무작위나 동시에 데이터 처리를 하고 있어서 모든 프로세스의 종료 이벤트가 발생하는 데 5초 걸리는 것으로 확인되고 있다. 이로 인하여 비동기식 I/O 방식으로 데이터 처리를 하면 프로세싱과 IO가 동시에 일어날 수 없다는 한계점을 극복할 수 있어 CPU의 효율을 극대화 할 수 있다.

비동기 논 블로킹 입출력 알고리즘은 한 프로세스가 몇 개의 I/O 작업을 하는 것을, 일단 세우거나(block) 다른 프로세스가 끝날 때까지 대기없이 허용하고 시간이 지나거나 I/O 완료 알림을 받은 뒤에 프로세스는 I/O 결과를 받을 수 있도록 하는 입출력 알고리즘이다[8,9]. 그림3은 비동기 논 블로킹 I/O 알고리즘의 흐름을 나타낸다.

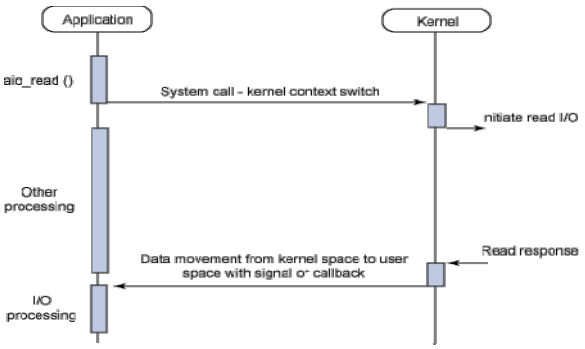

그림 3. 비동기 논 블로킹 I/O 알고리즘 흐름도

비동기 논 블로킹 I/O 알고리즘을 통하여 시스템 콜 요청(request)이 즉시 I/O 개시 여부를 반환하고, 애플리케이션은 하고 싶은 다른 일을 진행하면서, I/O는 백그라운드에서 실행할 수 있다. 비동기 논 블로킹 I/O 방식을 사용하면 더 빠르고 효율적인 I/O 애플리케이션을 만들 수 있으며, 오픈소스 하드웨어에 적용할 경우 낮은 하드웨어 성능으로도 높은 성능을 보장할 수 있다는 장점이 있다.

III. 오픈소스 하드웨어를 활용한 음성송출 시스템

1. 오픈소스 기반 음성송출 시스템 구성

본 논문에서 제안하는 오픈소스 하드웨어를 활용한 음성송출 시스템은 하드웨어와 이벤트 기반 논 블로킹 I/O 알고리즘을 적용한 미들웨어를 기본으로 동작하고 있으며, 미디어 콘텐츠 제작 애플리케이션을 통하여 미디어 콘텐츠를 업데이트하여 사용하는 방식으로 구성되어 있다. 그림 4는 제안하는 시스템 개념도를 나타낸다.

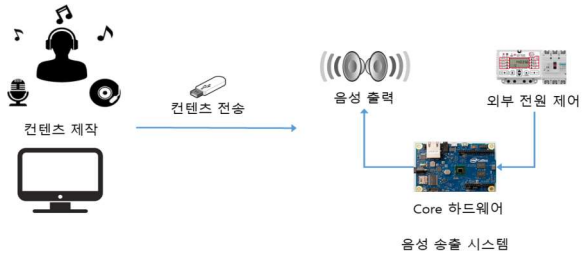

그림 4. 전체 시스템 개념도

그림 5는 제안하는 오픈소스 하드웨어를 활용한 음성 및 사운드 재생 시스템의 전체 아키텍쳐를 나타낸다.

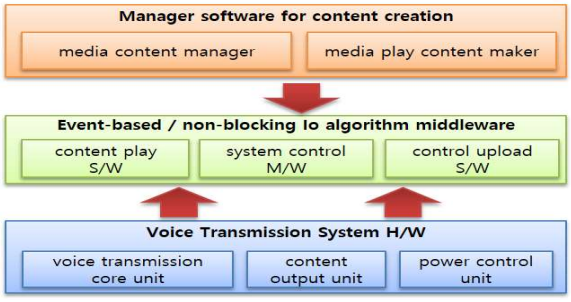

그림 5. 제안 시스템 아키텍쳐

2. 음성송출 시스템 하드웨어 구성

오픈소스 하드웨어인 라즈베리파이를 활용하여 초기 개발 비용과 개발 시간을 줄이고, 기존의 Kiosk 및 DID 시스템보다 저렴한 음성송출 시스템을 2가지(기본형, 통합형) 형태로 베이스 모델을 개발하였다. 그림 6은 음성 및 사운드 재생 하드웨어 구성을 나타낸다.

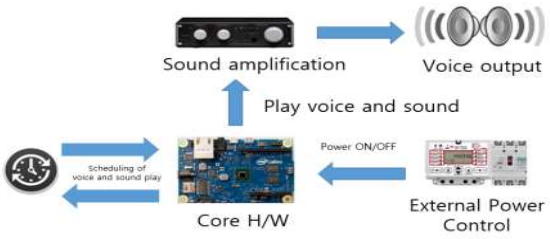

그림 6. 음성 및 사운드 재생 하드웨어 구성

■ 기본형 하드웨어 : 필수 기능을 가진 Core 하드웨어와 사운드 증폭 부분만 존재하는 형태로 제작하고, 통합형보다 더 작고, 더 저렴한 가격으로 제작하였으며, 사운드 출력, 시간 설정, 전원부로 구성하였다.

▫ 반드시 필요한 부분인 Core 하드웨어와 사운드 증폭 부분만 존재하는 형태로 제작함

▫ 통합형보다 더 작은 형태이며, 더 저렴한 가격으로 제작이 가능함

▫ 사운드 출력 : 사운드 출력을 위해서는 반드시 외부 스피커가 필요한 형태로 3.5mm 스트레오 단자를 지원하고 있음

▫ 시간 설정 : 라즈베리파이는 네트워크를 통하여 현재 날짜 및 시간을 받아오는 형태로 구성되어 있어 전원 연결이 끊어지면 시간이 리셋되는 문제가 있음

- RTC 설치 : 오픈소스 RTC 모듈을 통하여 라즈베리파이의 날짜 및 시간 설정의 한계를 극복함

- RTC의 고장 시 단순 모듈 교체를 통하여 손쉽게 수리 가능

▫ 전원부 : 사운드 및 Core 하드웨어에 전원을 인가하는 부분으로 기본형은 전원 차단 기능을 지원하지 않음

■ 통합형 하드웨어 : 하드웨어의 안정성 및 편의성을 위해서 외부 전원 제어부와 사운드 출력부를 추가한 형태의 음성 송출기이며, 사운드 출력, 시간 설정, 전원부, 전원 차단기, 전원 타이머, 자동 리셋 모듈로 구성하였다.

▫ 하드웨어의 안정성 및 편의성을 위해서 외부 전원 제어부와 사운드 출력부를 추가한 형태의 음성 송출기

▫ 사운드 출력 : 기본형의 단점인 자체 사운드 출력 불가를 보완한 형태로 제품의 하단에 10W의 스테레오 스피커를 가지고 있음

- 사운드 출력 조절 Core 하드웨어 아래에 있는 앰프를 통하여 음량을 조절할 수 있음

▫ 시간 설정 : 라즈베리파이는 네트워크를 통하여 현재 날짜 및 시간을 받아오는 형태로 구성되어 있어 전원 연결이 끊어지면 시간이 리셋되는 문제가 있음

- RTC 설치 : 오픈소스 RTC 모듈을 통하여 라즈베리파이의 날짜 및 시간 설정의 한계를 극복함

- RTC의 고장 시 단순 모듈 교체를 통하여 손쉽게 수리가 가능함

▫ 전원부 : 외부 전원이 불안정한 형태로 인가될 경우를 대비하고, 지정된 시간에 자동으로 전원을 인가할 수 있도록 전원 제어 하드웨어를 추가함

▫ 전원 차단기 : 외부 전원의 전압이나 전류가 불안정할 경우 하드웨어 큰 피해를 발생시키는 경우를 대비하기 위하여 전원 차단기를 설치함

▫ 전원 타이머 : 전자식 전원 타이머를 사용하여 메인 시스템

의 전원을 지정된 시간에만 동작되도록 설정

▫ 시스템 사용 도중 소프트웨어적 문제가 발생하여도 자동으로 전체 시스템을 리셋하기 때문에 초기화된 형태로사용이 가능하여 안정성이 높음

그림 7은 음성 및 사운드 재생 하드웨어 기본형 및 통합형을 나타낸다.

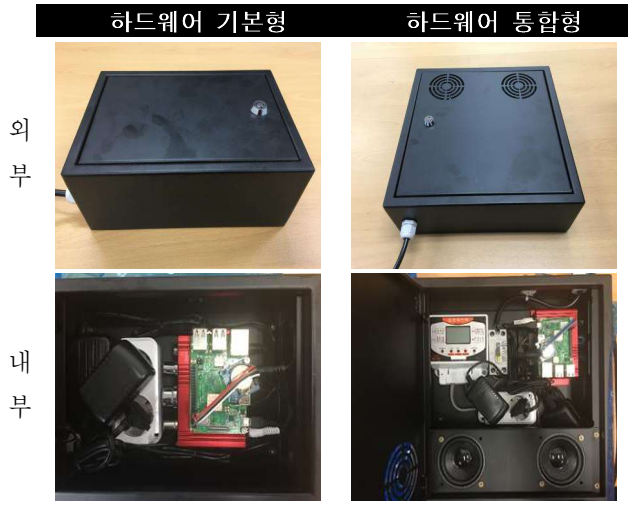

그림 7. 음성 및 사운드 재생 하드웨어

3. 음성송출 시스템 미들웨어

음성송출 시스템은 이벤트 기반 비동기 논 블로킹 I/O 방식의 미들웨어를 개발하여 Work Thread 영역에서 이벤트 처리 중에도 다른 프로세스를 처리할 수 있는 형태의 미들웨어를 설계 및 구현하였다. 그림 8은 이벤트 기반 논 블로킹 I/O 시스템 아키텍처를 나타낸다.

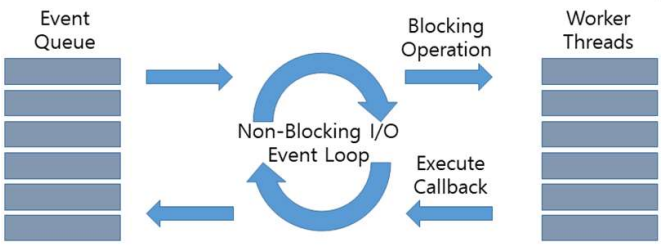

그림 8. 이벤트 기반 논 블로킹 I/O 시스템 아키텍처

■ 미디어 재생 : 사용자 미디어 재생 시나리오, 사용자 미디어 재생, 사용자 미디어 업데이트를 위한 이벤트 기반 논 블로킹 I/O 알고리즘을 활용한 소프트웨어 제공

■ 시스템 제어 : 시스템 제어 및 업데이트를 위한 이벤트 기반 논 블로킹 I/O 알고리즘 펌웨어 소프트웨어 제공

■ 비동기 논 블럭킹 이벤트 루프 : 모든 이벤트는 발생 시 Event Queue에 저장되며, Non-Blocking I/O Event Loop를 통하여 하나씩 차례대로 Work Thread 영역으로 이동하여 입출력 처리가 완료된다. Work Thread 영역에서 이벤트 처리를 하는 도중에도 이벤트 기반으로 동작하기 때문에 CallBack 이벤트가 오기 전까지는 프로세스는 다른 일을 진행하면서 동작할 수 있기때문에 시스템의 성능을 극대화할 수 있다.

그림 9는 음성송출 시스템 미들웨어의 소스코드를 나타낸다.

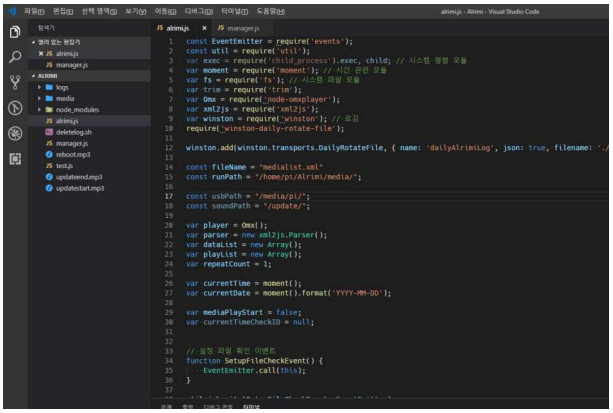

그림 9. 음성송출 시스템 미들웨어 소스코드

4. 콘텐츠 제작 및 관리 소프트웨어

관리자의 PC에서 음성 송출기에서 사용할 미디어 콘텐츠를 쉽게 생성 및 제작할 수 있는 기능을 제공한다. 관리자 프로그램에 각종 미디어를 등록, 관리하며 필요시에 바로 사용할 수 있고, 등록된 미디어 재생 시나리오와 해당 미디어를 USB 디스크로 업데이트할 수 있다. 그림 10은 콘텐츠 제작 애플리케이션의 사용자 인터페이스를 나타낸다.

그림 10. 콘텐츠 제작 애플리케이션의 사용자 인터페이스

5. 오픈소스 기반 음성송출 시스템 평가

본 논문에서 제안하는 오픈소스 하드웨어를 활용한 음성 및 사운드 재생 시스템을 콘텐츠 재생 제어, 미디어 콘텐츠 음향, 미디어 콘텐츠 업데이트 속도 면에서 평가를 진행하였으며, 이는 실제 서비스가 가능한 정도의 성능으로 판단된다.

콘텐츠 재생 제어 테스트는 설정된 시나리오에 의하여 콘텐츠가 재생될 경우 분 단위로 콘텐츠 설정이 가능한지와 콘텐츠 재생 시 오차 범위를 평가하였다.

세 개의 미디어 파일을 4:30, 4:40, 4:50분에 각각 출력되도록

세팅하고 라즈베리 콘솔 디버깅으로 해당 시간에 정확하게 송출되는 것을 확인하였다.

그림 11은 콘텐츠 재생 제어 테스트 결과를 나타낸다.

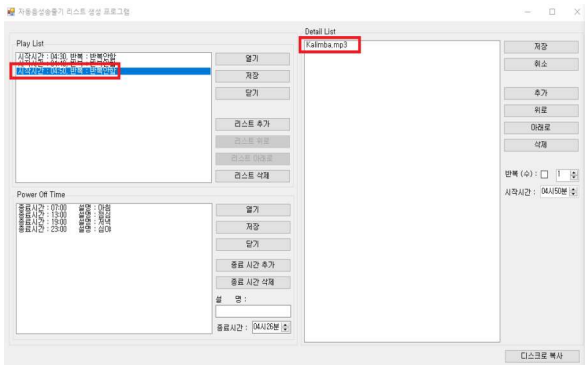

그림 11. 콘텐츠 재생 제어 테스트

미디어 콘텐츠 음량 테스트는 미디어 콘텐츠 재생 시 출력 음량을 음성 송출기와 거리 50cm, 100cm의 거리를 두고 소음 레벨 측정기 GM1357를 통하여 평가를 진행하였으며, 미디어 재생 전 소음 측정 결과는 43.5dB, 송출기 0.5m 소음 측정 결과는 102.5dB, 송출기 1m 소음 측정 결과는 98.4dB로 측정하였다. 그림 12는 음량 테스트 결과를 나타낸다.

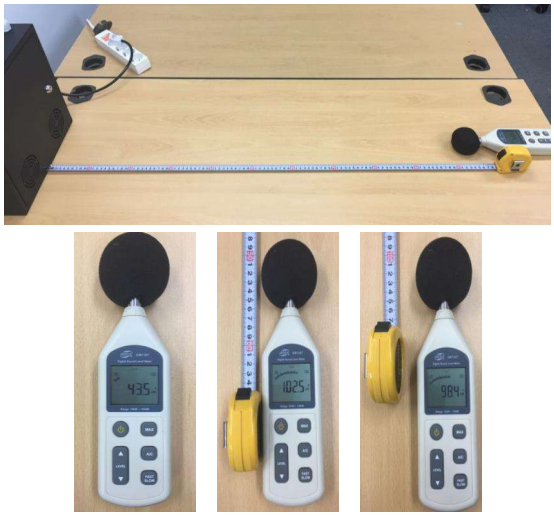

그림 12. 음량 테스트

미디어 콘텐츠 업데이트 속도 테스트는 콘텐츠 업데이트 시 100Mb, 500MB, 1GB의 파일을 전송하여 걸리는 시간을 확인하여 초당 몇 Mbps로 전송되었는지 평가를 통하여 검증하였다. 100M 더미파일 전송은 평균 전송 시간 3.5초(평균 240Mbps), 500M 더미파일 전송 테스트는 평균 전송 시간 24초(평균 전송 속도 175Mbps), 1G 더미파일 전송 테스트는 평균 전송 시간 50초(평균 전송 속도 172Mbps)로 측정되었다.

그림 13은 콘텐츠 업데이트 속도 테스트 결과를 나타낸다.

그림 13. 콘텐츠 업데이트 속도 테스트

Ⅳ. 결 론 및 향후 연구

본 논문에서는 음성 및 사운드를 이용하여 정보 취약 계층의 정보에 대한 접근이 쉬우며, 오픈 하드웨어를 통한 시스템 구축으로 시스템 초기 도입 비용과 유지 보수비용이 저렴하고, 다양한 형태로 활용할 수 있는 오픈소스 하드웨어와 이벤트 기반 논블로킹 I/O 알고리즘을 활용한 음성송출 시스템을 제안하고 평가하였다.

향후에는 제안한 음성송출 시스템의 성능 향상과 비동기 논블럭킹 이벤트 최적화 알고리즘에 관련해 연구할 예정이다.

REFERENCES

[1] S. John, and I. Kymissis, “Lab kits using the Arduino prototyping platform,” *IEEE Frontiers in Education Conference (FIE),* pT3C-1, 2010.

[2] Chang-Gyu Seong, Keel-Soo Rhyu, “Internet of things application service system with open source hardware,” *Journal of the Korean Society of Marine Engineering,* vol. 40, no. 6, pp. 542-547, 2016.

[3] Open Source Hardware Association(2014), https://www.oshwa.org/definition/korean (accessed Aug., 08, 2020).

[4] 오픈소스 하드웨어 시장의 견인차 IoT의 확산(2019), http://www.epnc.co.kr/news/articleView.html?idxno=82337(accessed Aug., 08, 2020)

[5] 김경민, “오픈소스 하드웨어를 이용한 유아의 자유선택활동관찰시스템의 설계 및 개발 연구,” *스마트미디어저널*, vol. 7, no. 2, pp. 47-53, 2018년 7월

[6] 강기욱, 이정환, 홍지만, “오픈소스 하드웨어에서 효율적인 임베디드 소프트웨어 개발을 위한 프레임워크,” *스마트미디어저널*, vol. 5, no. 4, pp. 49-56, 2016년 12월

[7] 박선, 차병래, 김종원, “오픈소스 기반 해양환경 모니터링 시스템,” *스마트미디어저널*, vol. 6, no. 3, pp. 75-82, 2017년 9월

[8] Non-blocking I/O(2018), http://wiki.sys4u.co.kr/pages/viewpage.action?pageId=7767390 (accessed Aug., 08, 2020).

[9] Asynchronous programming. Blocking I/O and non-blocking I/O(2020), https://luminousmen.com/post/asynchronous-programming-blocking-and-non-blocking (accessed Aug., 08, 2020).

저 자 소 개

김형우(정회원)

1998년 부산교육대학교 미술교육과 학사 졸업
2002년 동서대학교 디자인대학원 석사 졸업
2007년 동서대학교 디자인대학원 박사 졸업
2007년 ~ 현재 동서대학교 디자인학부 교수

<주관심분야 : 감성 디자인, 사용자 경험 디자인, 디지털 콘텐츠>

이현동(정회원)

2007년 부경대학교 컴퓨터공학과 석사 졸업.
2012년 부경대학교 컴퓨터공학과 박사 졸업.
2017년 동서대학교 산학협력단 연구교수

<주관심분야 : 상황인지 및 감성인지, 컴퓨터보안, 스마크워크>