

종합설계 프로젝트 수행 보고서

팀 온점 S2-3
최종 수행보고서

작성날짜 : 2020.11.11

팀원 : (팀장) 김학진

문준혁

송민욱

이세민

지도교수 : 교수 공기석 (인)

문서 수정 내역

작성일	대표작성자	버전(Revision)	수정 내용	비고
2020.01.20	이세민	1.0	수행 계획서	최초 작성
2020.02.22	송민욱	2.0	상세설계 추가	2차 수정
2020.04.27	김학진	3.0	프로토타입 추가	3차 수정
2020.06.26.	문준혁	4.0	CODE/DEMO 추가	4차 수정
2020.11.11.	김학진	5.0	결론 추가	최종 수정

이 문서는 한국산업기술대학교 컴퓨터공학부의
“종합설계”교과목에서 프로젝트“온점”을 수행하는
(S2-3, 김학진,문준혁,송민욱,이세민)이 작성한 것으로 사용하기
위해서는 팀원들의 허락이 필요합니다.

목 차

I. 서론

1. 작품선정 배경 및 필요성	1p
2. 기존 연구/기술동향 분석	2p
3. 개발 목표	4p
4. 팀 역할 분담	5p
5. 개발 일정	6p
6. 개발 환경	7p

II. 본론

1. 개발 내용	9p
2. 문제 및 해결방안	10p
3. 시험시나리오	11p
4. 상세 설계	13p
5. Prototype 구현	29p
6. 시험/테스트 결과	38p
7. Coding & DEMO	45p

III. 결론

1. 연구 결과	55p
2. 작품제작 소요재료 목록	57p

참고자료	58p
------------	-----

I. 서론

1. 작품선정 배경 및 필요성

2017년 보건복지부에서 실시한 장애인 실태조사에 따르면 시각장애인 중 86%가 점자 해독이 불가능하다고 한다. 점자 해독이 불가능한 가장 큰 이유는 시각장애인 중 95% 이상이 후천적으로 발병하며, 그 중 70%는 20세 이후에 발병했기 때문에 이미 문자에 익숙해진 상태에서 새롭게 점자를 학습하기에는 어려움이 있다는 것이다.[1] 그림 1과 그림 2는 2017년 보건복지부에서 실시한 장애인 실태조사의 결과 표이다.

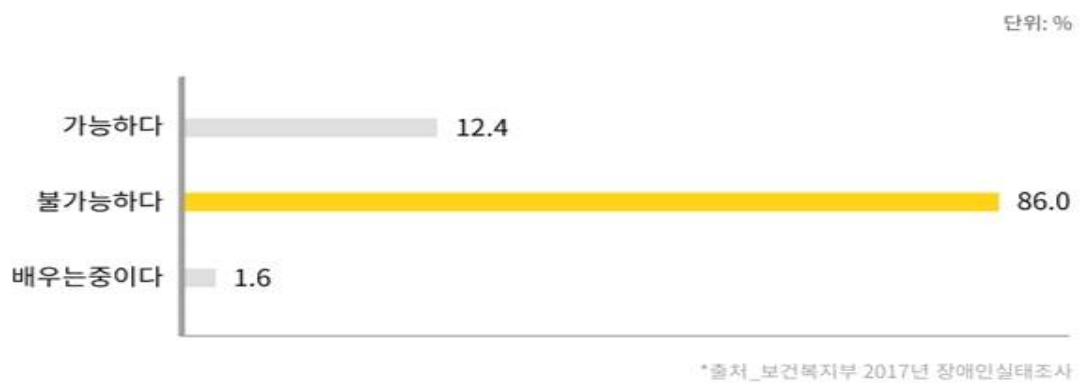


그림 1 보건복지부 2017 장애인 실태조사 - 점자 해독여부

단위: %

구분	선천적 원인	출생시 원인	후천적 원인 (질환)	후천적 원인 (사고)	원인 불명
전체	5.0	0.1	54.4	38.0	2.6

*출처_보건복지부 2017년 장애인실태조사

그림 2 보건복지부 2017 장애인 실태조사 - 발병원인

점자를 자가학습 하는 데 어려움이 많다. 전문기관에서 선천적 시각 장애인을 위한 아동용 프로그램이 전문적으로 운영되고 있지만, 점자 교육을 받기 위해선 반드시 조력자가 필요한 상황이다. 하지만 후천적 시각 장애인도 점자 교육을 받을 수 있는 조력자를 구하기 어려운 상황이고 점자를 교육받기 위해선 전문기관의 도움을 받거나 장애인 관련 복지센터에서 운영하는 프로그램에 참여해야하기 때문에 쉽고 간단하게 점자교육을 받기에 어려운 상황이다. 따라서 후천적 시각 장애인이 점자를 자가 학습으로 습득하는 방법이 필요하고 본 보

고서는 점자교육 플랫폼에 관한 내용을 기술한다.[6]

2. 기존 연구/기술동향 분석

그림 3은 2019 한국산업기술대학교 산학기술대전 출품작 “Edu.Dot”의 점자판 그림이다. 시각장애인을 위한 점자교육 임베디드 시스템으로써 라즈베리파이를 이용하여 디바이스 드라이버에 점자교육 플랫폼을 탑재한 구조이다. 음성인식과 자체 탑재 터치스크린으로 시각장애인 UI를 설계하였으며 서보모터와 낚시줄로 점자를 표현한 작품이다.



그림 3 시각장애인 점자교육 임베디드 시스템 “Edu.Dot”의 점자판

그림 4는 2018년 한국산업기술대학교 산학기술대전 출품작 “봄”이다. 봄 어플리케이션은 시각장애인을 위한 점자 교육 어플리케이션이다. 시각장애인을 위한 터치와 진동세기 조절 등 다양한 동작센서를 통한 UI로 구성되어 있고 기본 점자 교육과 더불어 다양한 요소가 포함되어있다. 따라서 해당 어플리케이션만으로도 시각장애인이 점자교육이 가능하다.[3]



그림 4 시각장애인 점자교육 어플리케이션 “봄”의 아이콘

그림 5는 “VC 진동점자교육” 어플리케이션의 화면이다. 이 어플리케이션은 스마트폰을 소유한 시각장애인이 터치스크린의 드래그 기능을 이용한 점자표현을 통해서 점자교육을 받을 수 있게 만들어진 어플리케이션이다. 별도의 디바이스가 없이 점자 교육을 받을 수 있다.[4]



그림 5 VC 진동점자교육 어플리케이션 화면

그림6은 “탭틸로”라는 점자교육 플랫폼이다. 디바이스와 어플리케이션을 통해 점자교육을 받을 수 있다. “탭틸로”는 현재 대부분의 특수학교에서 사용하고 있는 대표적인 점자 교육 플랫폼이다. 어플리케이션과 점자 디바이스를 통해 아동이 쉽게 점자를 교육받을 수 있으며 다양한 기능을 포함하고 있다.[2]



그림 6 “탭틸로”의 동작화면

기존 연구/기술동향 사례에서 살펴봤듯이 다양한 장단점이 존재한다. 디바이스가 없는 환경에서 점자를 교육 받는 플랫폼은 스마트폰의 진동이나 드래그 같은 기능을 사용해서 교육을 받는데, 이는 실제 점자가 표현되는 물리적인 속성을 완전히 배재시키는 상황이라 시각장애인이 실제 점자를 접했을 때 많은 괴리감을 느낄 수 있다는 문제점이 존재한다. 따라서 점자 디바이스의 필요성이 대두 되었고 점자키트 기능도 추가할 예정이다. 다음으로 혼자서도 학습이 가능한 구조인 사례도 있었지만 조력자가 필요한 사례도 볼 수 있었는데 본 졸업 작품의 가장 큰 목적중 하나인 조력자 없이 자가학습으로 점자를 학습할 수 있도록 하는 것에 중점을 두었다. 표 1은 관련 사례와 온점을 비교/분석한 표이다.

표 1 기존 연구와 기술동향 사례를 비교/분석한 표

	UI 적합성	점자키트 유무	단독 사용	학습 방법	웹 페이지 유무	단계별 학습
탭틸로	X	O	X	△	X	O
봄	O	X	O	O	X	O
VC 진동 점자 학습	O	X	O	X	X	X
온점	O	O	O	O	O	O

3. 개발 목표

혼자서도 점자 학습을 할 수 있도록 하기 위하여 점자키트와 어플리케이션의 조화를 이룬 점자 교육 플랫폼을 개발할 예정이다. 그리고 어플리케이션의 기능 중에 시각장애인이 점자에 관련해서 건의를 할 수 있는 음성 게시판을 구현할 예정이다. 다음 세 가지 주요 목적을 살펴본다.

첫째, 시각장애인의 점자 교육용 점자키트는 어플리케이션과 블루투스로 연동되어 교육을 실시한다. 어플리케이션에서 데이터를 받아서 점자로 표현되고 이를 시각장애인이 손쉽게 손으로 인식할 수 있는 구조로 개발한다. 점자 디바이스의 중요한 부분은 시각장애인이 점자를 분명하게 알 수 있고 명확하게 손에 느껴지는 점자를 구현하는 것이다.

둘째, 시각 장애인의 점자 학습 어플리케이션을 개발한다. 어플리케이션 개

발은 시각 장애인이 손쉽게 사용할 수 있도록 어플리케이션의 화면 전환 및 주요 기능은 제스처 기능으로 진행한다. 그리고 각 화면에서는 음성으로 안내함으로써 보이지 않아도 어떤 화면인지 알 수 있도록 구현한다. 점자 교육은 단계별 학습을 실시하는데 초성, 종성, 모음으로 나뉘어 가장 기초적인 점자부터 학습을 한다. 더하여 점자 퀴즈 기능을 포함하여 점자 학습에 재미를 더할 수 있고 더욱 기억에 오래 남을 수 있도록 도움을 주고 퀴즈의 난이도 또한 단계별로 도전할 수 있게 구현하여 학습의 성취를 증대시킨다. 퀴즈의 문제도 평소 시각장애인이 어려워하거나 많이 쓰는 단어 위주로 퀴즈 문제를 구성하여 더욱 효율적으로 학습할 수 있다. 퀴즈를 풀거나 점자 해독을 할 때는 터치 스크린의 화면 슬라이드 드래그를 통해 점자를 입력하는 방식으로 진행한다. 그리고 실제 점자를 읽던 도중 모르는 단어나 어려운 단어를 번역하는 기능도 포함하고 음성 게시판에 점자 관련 의견을 낼 수 있는 기능도 포함한다.

셋째, 시각장애인의 의견공유를 위한 점자 관련 웹페이지를 구현한다. 퀴즈에서 오답률이 높은 점자나 번역 빈도수가 높은 점자의 데이터 순위를 표시하여 시각장애인이 어려워하는 점자를 한 눈에 볼 수 있도록 시각화 하는 기능과 시각장애인이 자유롭게 음성으로 게시글을 작성할 수 있는 음성게시판 기능을 포함한다. 이 데이터는 다양한 용도로 활용할 수 있다.

예를 들어, 시각장애인이 점자가 표기되었으면 하는 물건에 관한 정보를 알 수 있고 시각장애인이 보편적으로 어려워하는 점자의 정보를 다른 시각장애인에게 알려 줄 수 있고 물건에 표기할 때 별도의 조치를 취할 수 있다.

4. 팀 역할 분담

4명의 팀원으로 졸업작품이 진행되기 때문에 역할 분담을 하였다. 팀장 역할을 맡은 김학진 학생은 데이터 통신을 위한 서버구축을 담당하고 가장 중요한 어플리케이션을 문준혁 학생과 송민욱 학생이 담당한다. 점자 디바이스 담당은 이세민 학생이 담당한다. 표 2는 팀원 역할 분담표이다.

표 2 팀원 역할 분담표

	김학진 (팀장)	문준혁	송민욱	이세민
자료 수집	앱, 서버, DB 연동방법 및 활용방안 조사	TTS, STT API 사용법 & 실례조사	- 제스처 & 드래그 인터페이스 조사 - 기존 어플 기술 조사	아두이노 부품 및 실례 조사
설계	- 서버 설계 - 회로 설계	- 앱 구조 설계 - DB 스키마 설계	- 앱 구조 설계 - DB 스키마 설계	- 회로 설계 - 서버 설계
구현	APP, DB, Web의 연동 서버 구현	TTS&STT를 사용한 음성인식 기능 구현	제스처 &드래그를 사용한 모션기능 구현	아두이노를 활용한 점자 키트 구현

5. 개발 일정

표 3은 세부 일정표이다 2월달까지 프로토타입을 완성하고 1학기가 시작되면 실제 시각장애인을 만나 테스트도 진행하는 등 실효성을 검증할 예정이다.

표 3 개발 세부일정

추진 사항	상세 내용		19	2020						
			12	1	2	3	4	5	6	7
계획	유사 콘텐츠 실례 조사 및 기능 비교									
분석	요구사항 정의 및 분석									
설계	소프트웨어 설계									
	하드웨어 설계									
	앱 기능 매뉴얼 설계									
개발	DB 구현									
	제스처 기능 개발									
	점자 키트 개발									
	음성인식 기능 개발									
	프로토 타입 개발									
	웹	클라이언트 구현								
		서버 구현								
	매뉴얼 작성									
테스트	샘플링 제작 및 피드백									
종료	최종 결과물 산출									

6. 개발 환경

아두이노 Uno를 사용하여 점자 키트를 구성한다. [7] 점자의 표현을 위해 서보모터를 48개를 사용하고 PCA9685 16채널 PWM 모듈을 사용하여 48개의 서보모터를 제어 한다. 또한 어플리케이션과 데이터 송/수신을 위해 HM-10 블루투스 모듈을 사용한다. 맞춤형 점자 표현 위해 모델링을 하고 키트를 구성하기 위해 NX10.0을 사용하여 3D프린터에 필요한 모델링을 실시한다. 표 4는 하드웨어 개발환경을 정리한 표이다.

표 4 하드웨어 개발환경

Hardware Development Environment		기능
	Arduino Uno	점자키트를 만들기 위한 아두이노 보드
	PCA9685	48개의 서보모터를 동시에 제어하기 위한 모듈
	HM-10 블루투스 모듈	아두이노와 안드로이드 간 블루투스 통신을 위한 센서
	SG90 Servo Motor	점자를 표현하기 위해 사용하는 모터
	NX10.0	3D 프린터 모델링 개발 소프트웨어[8][9]
	Arduino IDE ver 1.8.1	아두이노 개발을 위한 소프트웨어

어플리케이션을 개발하기 위하여 Android Studio를 사용한다. 시각장애인이 사용하는 스마트폰 운영체제 버전이 6.0 이하인 점을 염두한다. 서버를 개발하기 위하여 AWS를 이용한다. EC2에 Linux 운영체제 기반에 Apache Tomcat을 설치하고, RDS를 탑재 함으로써 하드웨어적인 문제를 해결하고 MySQL을 통하여 점자관련 데이터베이스 테이블 설계 및 구현을 한다[12]. 웹 페이지를 개발하기 위하여 VScode 및 Eclipse를 사용한다. 표 5는 어플리케이션 및 서버의 개발환경을 정리한 표이다.

표 5 어플리케이션 및 서버 개발환경

Software Development Environment			Version
AWS	Server	Apache Tomcat / Linux	ver 9.0 / Linux2 Ami
	DBMS	RDS / MySQL	ver 8.0
Web Development tool		VScode, Eclipse	ver 1.39, ver 4.11.0
Android Development tool		Android Studio	ver 3.5.1 (API 23)

II. 본론

1. 개발 내용

개발의 진행은 크게 어플리케이션을 중심으로 아두이노와 블루투스 통신을 통해 점자데이터를 주고받고[5] 웹서버를 통해 어플리케이션에서 받은 데이터를 데이터베이스에 저장한다. 그리고 데이터베이스에 있는 데이터를 퀴즈와 번역 데이터로 웹페이지에 표현한다. 그림 7은 온점의 시스템 구성도이다.

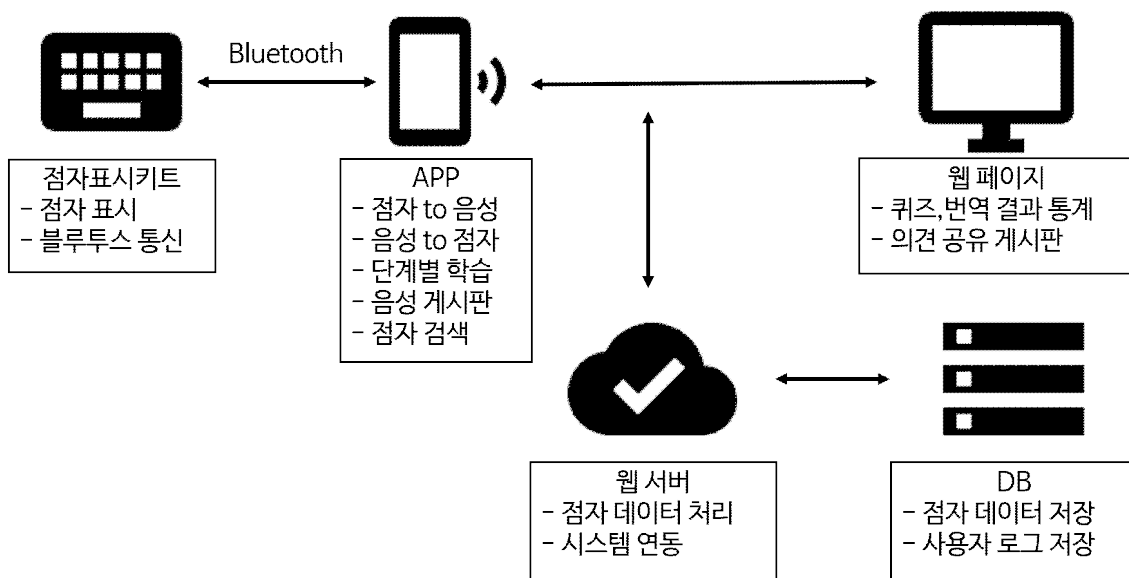


그림 7 시스템 구성도

첫째, 점자키트 개발내용은 아두이노 Uno를 사용하여 점자키트를 개발한다. 점자의 표현을 위해 48개의 서보모터를 사용해서 총 8개의 점자를 표현한다. 어플리케이션과 연동하기 위하여 블루투스 통신을 이용하여 데이터를 입출력한다. 소스코드에서 모터를 제어하기 위해 16채널 PWM모듈 라이브러리를 사용한다. 그리고 일련의 점자 데이터를 가공해서 각각 서보모터에 인가하여 확실하게 점자를 표현하는 코드를 작성한다. 서보모터에 납시줄이 달려있고 납시줄 끝에는 스프링이 달려있어 서보모터가 회전하면 납시줄을 당겨 스프링이 압축되고 스프링에 달린 점자캡을 아래로 움직여 점자를 표현하는 방식을 채택하였다. 납시줄의 엉킴을 방지하기 위하여 ㄱ자 경첩을 이용하여 엉킴을 방지하고 튜브에 납시줄을 삽입 함으로써 안전성을 높인다. 점자를 표현하기 위해 NX10.0을 이용한 맞춤형 모델링을 실시하여 점자 키트를 구성한다.

둘째, 어플리케이션 개발내용은 안드로이드 스튜디오에서 JAVA언어를 사용하여 시각장애인에게 제공하는 어플리케이션을 개발한다.[10] 어플리케이션은

웹 서버, 아두이노와 연동된다. 아두이노와는 블루투스로 연동하여 점자 데이터를 주고받고, 웹 서버와는 네트워크기반으로 데이터를 송수신한다. 아두이노와 연동하여 교육과 퀴즈 기능을 사용하는데, 어플리케이션에서 문자를 점자로 표현할 수 없기 때문에 아두이노와 연동한다. 웹 서버는 어플리케이션에서 사용하는 점자 데이터를 데이터베이스에서 가져오고, 번역 빈도수가 높은 단어와 오답률이 높은 단어의 데이터를 데이터베이스에 저장하기 위해 웹 서버와 연동한다. 어플리케이션은 시각장애인이 주로 사용하기 때문에 깔끔한 UI로 개발하기 위해 'Material design'을 사용하고, 단순한 디자인으로 구성한다. 어플리케이션 내에서 사용하는 모든 기능은 커스텀 'Drag&Drop'과 'Shaking'을 사용하여 손쉽게 사용할 수 있도록 만든다. 또한, 점자를 번역하거나, 퀴즈 기능을 사용할 때 시각장애인이 편하게 사용할 수 있도록 음성API를 사용하여 점자를 음성으로, 음성을 점자로 바꿀 수 있도록 한다.

셋째, 서버 및 웹페이지 개발내용은 AWS EC2를 활용하여 Linux를 설치하고 Apache Tomcat을 구축한다. JSP를 사용하여 서버 로직을 구성하며[11] DB와 WEB 그리고 APP 간의 데이터 이동을 처리한다. HTML과 CSS를 사용하여 스마트폰과 PC에 동시에 지원하는 반응형 웹페이지를 개발하고 JavaScript를 활용하여 유동적인 웹 페이지를 구현한다. 번역 빈도수가 높은 단어와 오답률이 높은 단어에 대한 데이터 통계를 시각화하여 한 눈에 볼 수 있도록 하며 시각장애인이 사용할 수 있는 음성게시판 기능을 구현한다.

마지막으로 데이터베이스 개발 내용은 AWS RDS의 Mysql을 사용하며 DA#으로 문자와 점자를 매핑하는 테이블을 설계한다.

2. 문제 및 해결방안

하드웨어의 문제점으로는 첫째, 많은 수의 서보모터를 제어하기 때문에 전압의 분배에 있어서 문제가 발생할 것으로 예상된다. 이에 외부전압을 사용할 수 있는 PWM 모듈을 이용하여 해결할 것이다. 둘째, 실제 점자를 표현하기 위해 스프링과 낚시줄로 점자를 표현할 계획인데 이를 구성하는 3D 모델링 부분에서 어려움이 존재한다. 이에 NX10.0을 통한 3D 프린터 호환 가능 모델링을 학습 중에 있다. 마지막으로 점자 데이터를 맵핑시키는 것을 구현하는 데 어려움이 있다. 이 부분은 어플리케이션 개발의 중간단계에서 팀원들과 회의를 통해 점자의 표현방식을 의논할 예정이다.

어플리케이션의 문제점으로는 '온점'만의 커스텀 제스처를 구현하는 데 어려움이 있다. 시각장애인이 사용하는 만큼 이해하기 쉽고 편리하게 사용할 수 있도록 제스처를 만드는 부분에서 어려움이 존재하는데, 이는 사용하기 쉬운 제

스쳐 방식을 연구한 후 여러 시행착오를 겪고 다른 어플리케이션의 장점을 계승 발전시키며 완성시킬 예정이다. 다음으로는 시각장애인 전용 스마트폰에서도 정상적인 작동을 하는지 고려해서 개발해야하는 문제가 있다. 이는 시각장애인 전용 스마트폰의 기능을 직접 사용해봄으로써 동작하는 방식을 숙지한 뒤 해당 기능과 마찰이 없이 개발을 진행할 예정이다.

서버 및 웹페이지 문제로는 점자데이터를 포함한 데이터베이스를 설계하는 부분에서 어려움이 있다. 이는 스키마 설계를 학습한 후, 문자와 점자데이터를 매핑시키는 방식을 팀원들과 회의를 통해 ‘온점’에 적합한 데이터베이스를 설계할 예정이다. 다음으로는 반응형 웹페이지 개발의 어려움이 있다. 반응형 웹페이지의 기법을 학습하고 오픈소스를 활용하여 개발을 진행할 예정이다.

3. 수행 시나리오

어플리케이션과 디바이스가 연동되어 동작하는 기능은 교육과 퀴즈가 있다. 어플리케이션을 실행하면 어플리케이션 메뉴얼을 음성으로 안내(생략 가능)를 받는다. 그 다음 사용자가 교육이나 퀴즈 항목을 선택한다. 먼저 퀴즈의 경우에는 음성으로 퀴즈를 푸는 방법을 안내 받는다. 안내가 끝나면 단계(초성, 종성, 모음)를 선택할 수 있다. 단계를 선택하면 해당 단계의 문제가 디바이스에 점자로 표시되며 음성을 이용해서 정답을 입력한다. 다음 교육의 경우에는 제스처를 통해 초성, 종성, 단어 등 교육 받을 항목을 선택한다. 음성으로 해당 점자를 안내 받은 후 점자키트로 확인하며 교육을 진행한다. 그림 8은 디바이스를 사용하는 어플리케이션의 시나리오이다.

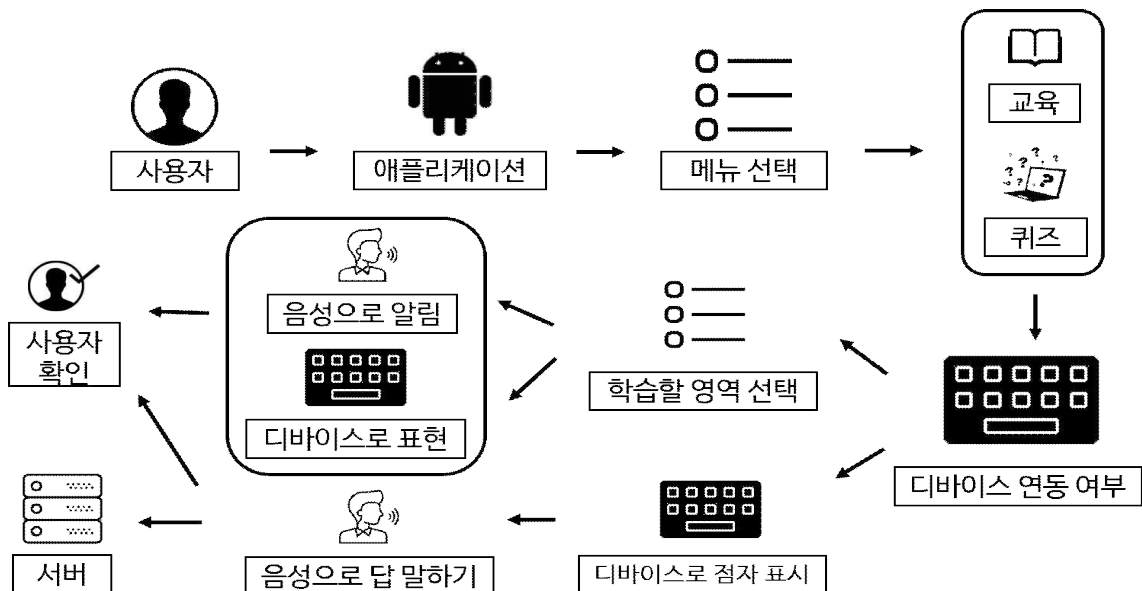


그림 8 어플리케이션 - 디바이스 동작 시나리오

디바이스와 연동되지 않아도 사용가능한 어플리케이션 기능은 번역과 음성 게시판이 있다. 어플리케이션을 실행하고 음성안내에 따라 번역이나 음성게시판 항목을 선택한다. 번역의 경우에는 사용자가 제스처를 통해 점자를 표현하면 어플리케이션이 점자를 인식하여 해당 점자가 의미하는 뜻을 음성으로 말해준다. 음성게시판 동작 시나리오는 음성 게시판 항목을 선택한 후, 음성게시판의 안내음성을 듣고 음성으로 제목과 내용을 입력한다. 작성된 글은 웹페이지에 업로드된다. 그림 9은 디바이스 없이 수행되는 어플리케이션 시나리오이다.

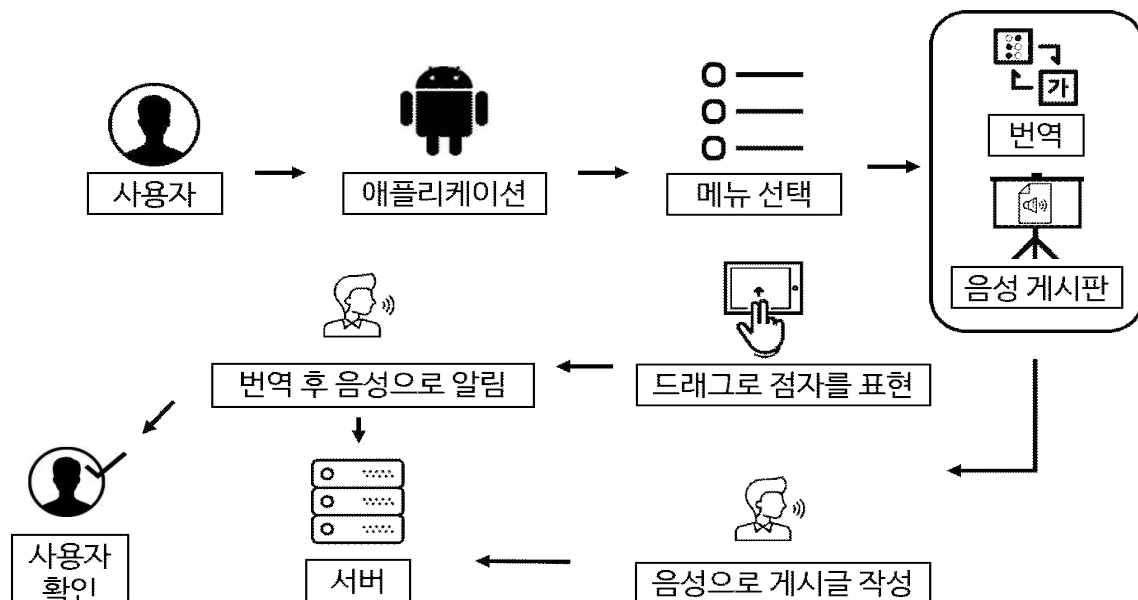


그림 9 디바이스 없이 수행되는 어플리케이션 시나리오

웹페이지에서 수행되는 시나리오는 어플리케이션의 기능 중 퀴즈, 번역, 그리고 음성게시판에서 번역 빈도수가 높은 점자, 오답률이 높은 점자, 음성게시판 게시글을 데이터화하여 한 눈에 볼 수 있도록 시각화 한다. 이를 시각 장애인에 관련된 사람이 열람할 수 있어 시각장애인이 어려워하는 부분을 알 수 있고 이를 활용할 수 있다. 예를 들어 다른 시각장애인에게 다른 시각장애인이 어려워하는 점자를 알려줄 수 있고 모여진 데이터가 시각 장애인의 점자 관련 지표로 활용될 수 있다. 그림 10은 웹페이지에서 수행되는 시나리오이다.

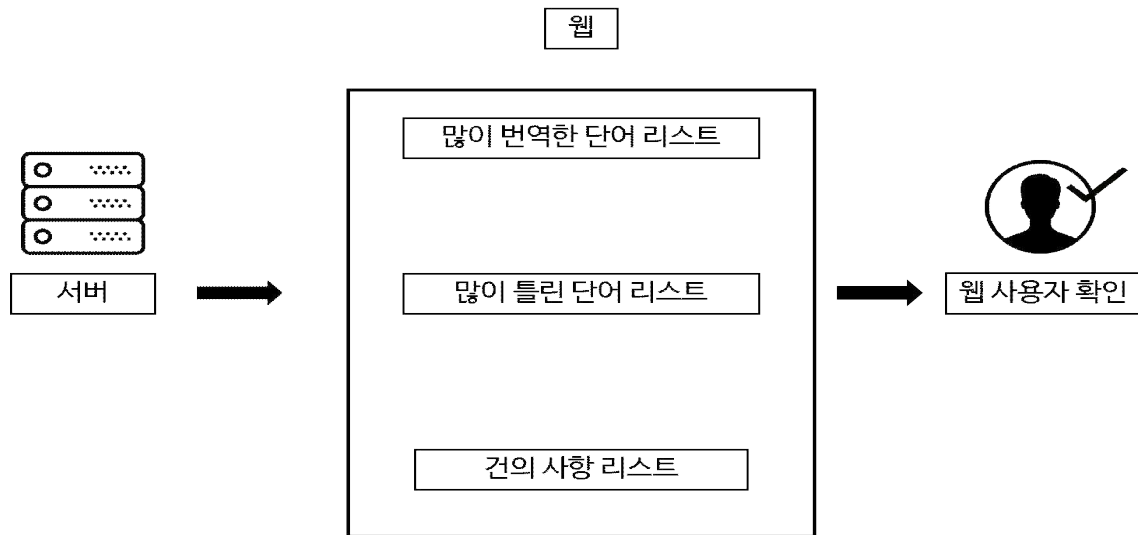


그림 10 웹페이지에서 수행되는 시나리오

4. 상세 설계

개발내용에서 서술한 커스텀 ‘Drag And Drop’ 과 ‘Shaking’을 먼저 상세히 설명 한다. 기존 안드로이드에서 제공하는 ‘TouchEvent’ 인터페이스를 사용하지 않고, 한 손가락 과 두 손가락 ‘TouchEvent’가 각기 다른 기능을 수행하기 때문에 커스텀화 하여 본 어플리케이션 목적에 맞게 개발하여 사용한다. 기존 ‘TouchEvent’가 발생하면 커스텀 터치와 연결이 되도록 하는 인터페이스를 구현하여, 이벤트가 발생 시 이벤트를 처리하는 ‘CustomTouch’ 클래스에서 상세 로직을 구현해 간다.

한 손가락과 두 손가락별 기능 제스처를 정리하면 한 손가락은 상하좌우와 롱 터치 그리고 더블 탭 기능이 있고 두 손가락 경우에는 위에서 아래로 드래그하는 ‘BACK’ 기능과 아래서 위로 드래그하는 ‘SPECIAL’ 기능이 있다. 이벤트가 발생하면 그림 11과 같이 ‘CustomTouch’ 클래스의 touchEvent에서 손가락 수에 맞는 기능을 수행하는 함수를 호출하게 된다. 한 손가락에 경우 ‘ACTION_DOWN’과 ‘ACTION_UP’에 따라 다른 기능을 수행한다.

‘ACTION_DOWN’일 때 ‘oneFingerDown’을 호출하는데, 이는 그때의 좌표를 저장하고 롱 터치를 판정하는 스레드를 진행한다. ‘ACTION_UP’일 경우 ‘ACTION_DOWN’과 마찬가지로 ‘oneFingerUp’을 호출하여 해당 함수 내에서 좌표를 저장한 후 더블 탭 판정하는 스레드를 진행한다. 그림 12의 흐름과 같

이 이후 더블 탭 판정 스레드 안에서 상하좌우, 롱 터치, 더블 탭을 처리하여 'Type'을 결정한다.

두 손가락에 경우 한 손가락과는 다르게 'ACTION_POINTER_DOWN'과 'ACTION_POINTER_UP'으로 나누는데, 이는 첫 번째 터치와 구별하기 위함이다.[14] 한 손가락과 같이 두 개의 'ACTION'에 따라 'twoFingerDown'과 'twoFingerUp' 함수를 통해 좌표를 저장하고, 좌표에 따른 제스처를 판정하는 함수를 호출하여 'Type'을 가져온다. 즉, 그림 13의 흐름으로 기능이 수행된다. 그리고 제스처에 따른 'Type'을 결정하기 위해 'enum'타입으로 'Type'을 정의한 후, 제스처를 판정하는 클래스에서 'Type'을 반환한다.

각 터치 이벤트 발생시 'FingerLocation' 클래스에 정의된 함수(setDownLocation과 setUpLocation)를 통해 액션에 맞는 좌표를 저장한다. 이후 제스처를 판정하는 클래스인 'FingerFunctionProcess'는 'FingerLocation' 객체를 통해 ACTION에 따른 좌표를 가져와 손가락 수에 맞게 상하좌우를 판정하는 로직을 수행한다. 그림 14와 그림 15의 흐름과 같이 손가락 수에 맞게 DOWN일 때 좌표와 UP일 때 좌표 각각의 차이를 구해 양의 방향인지 음의 방향인지를 파악하여 제스처를 판정한다.

그림 11은 터치 이벤트 발생 시 커스텀 터치 내에 존재하는 터치 이벤트의 흐름도이다.

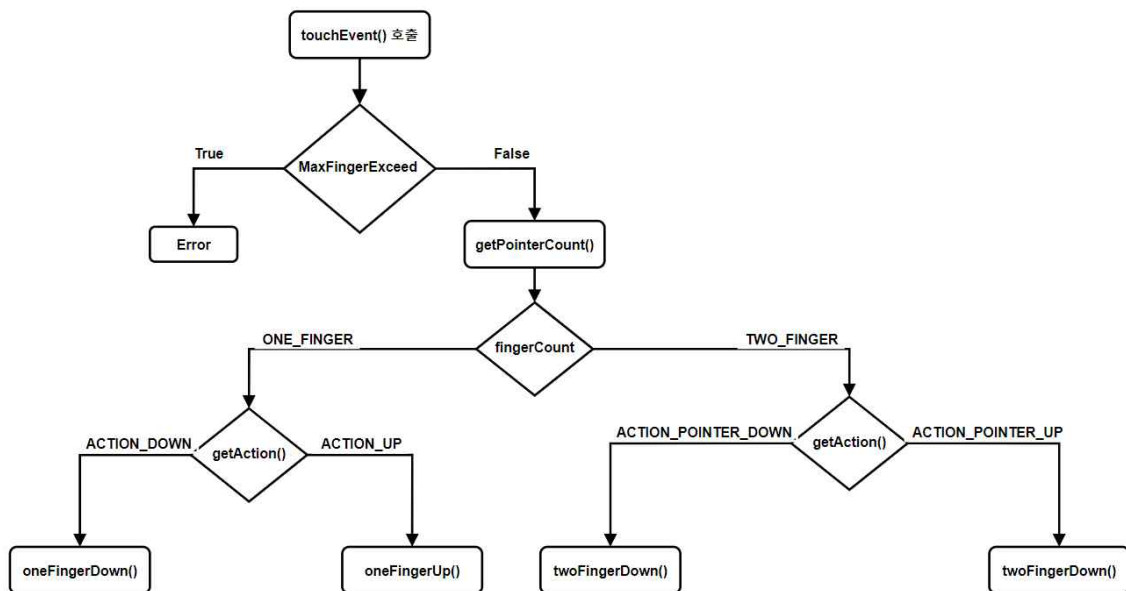


그림 11 CustomTouch의 touchEvent() 흐름도

그림 12는 터치 이벤트 발생 시 액션에 따른 한 손가락 기능 수행 흐름도이다.

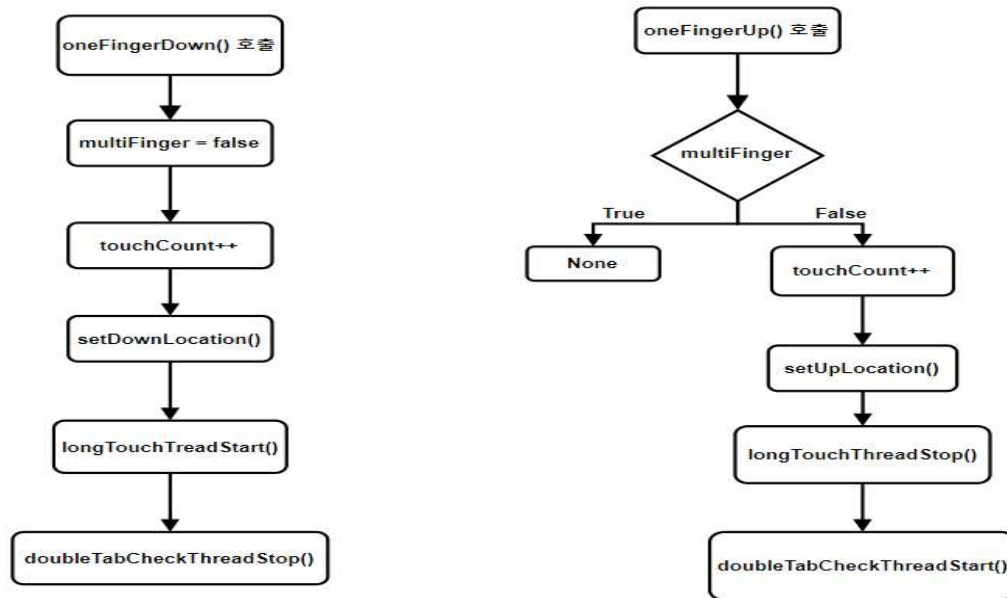


그림 12 oneFingerDown(Up) - 액션에 따른 한 손가락 수행 흐름도

그림 13은 터치 이벤트 발생 시 액션에 따른 두 손가락 기능 수행 흐름도이다.

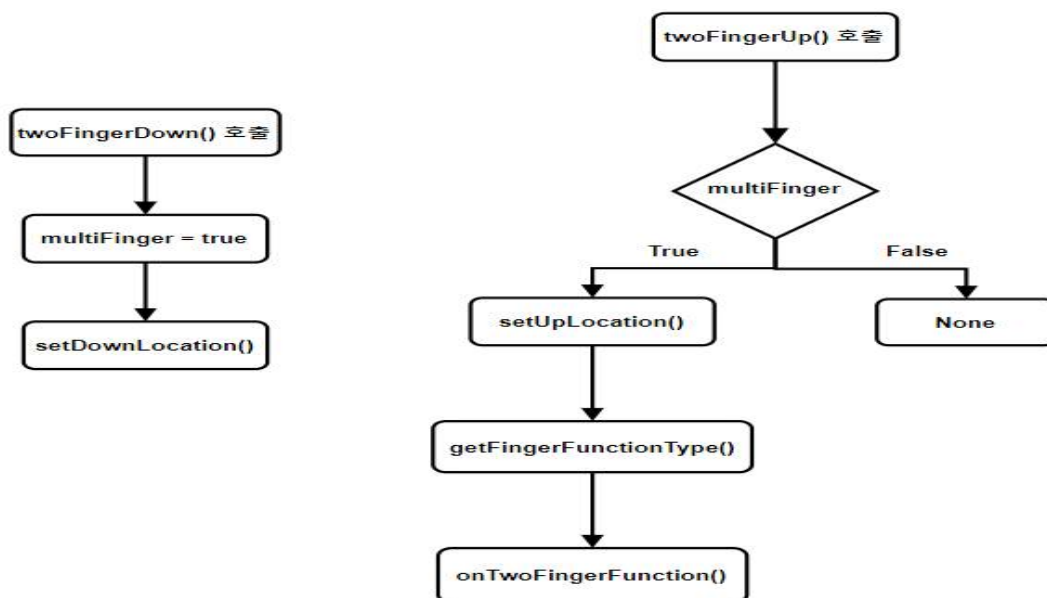


그림 13 twoFingerDown(Up) - 액션에 따른 두 손가락 수행 흐름도

그림 14는 한 손가락일 때 저장된 좌표의 차이를 이용하여 제스처를 판정하는 흐름도이다.

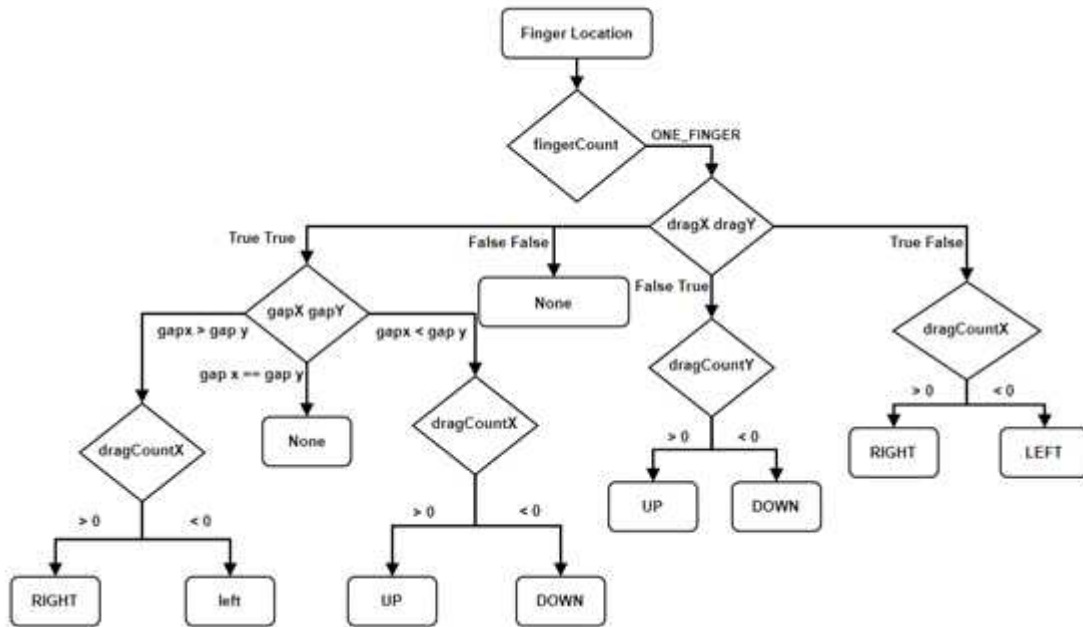


그림 14 FingerLocation - 한 손가락 제스처 판정 흐름도

그림 15은 두 손가락일 때 저장된 좌표의 차이를 이용하여 제스처를 판정하는 흐름도이다.

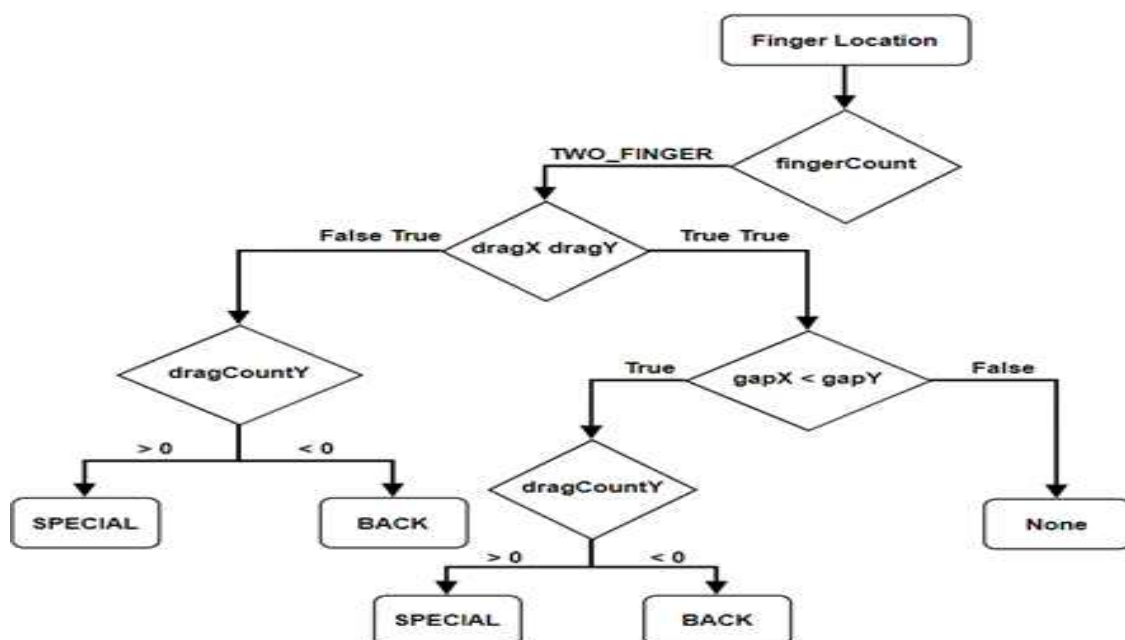


그림 15 FingerLocation - 두 손가락 제스처 판정 흐름도

한 손가락 제스처인 상하좌우를 제외한 롱 터치, 더블 탭을 판정하기 위한 판정 스레드에 대해 상세히 설명하고자 한다. 상하좌우에 롱 터치, 더블 탭을 할 때 기존 'ACTION_DOWN'과 'ACTION_UP'이 중복되어 인식이 되는 문제가 발생한다. 즉, 롱 터치와 더블 탭을 인식하기 전에 상하좌우 제스처를 먼저 인식되기 때문에 스레드를 이용하여 제스처 판정을 한다. 각 기능 수행 흐름도는 그림 16과 같이 진행이 되며, 롱 터치와 더블 탭 판정 스레드에서는 'TimerTask'와 'Timer'를 이용한다.

롱 터치 스레드는 0.5초를 기준으로 'boolean' 타입의 변수를 이용하여 판정을 한다. 한 손가락 'ACTION_DOWN'일 때 스레드를 시작하여 판정을 하며 'ACTION_UP'이 되는 경우 스레드는 정지한다. 즉 0.5초간 'ACTION_DOWN' 상태를 유지하면 'TRUE'값으로 저장된다. 이후 변수를 이용하여 더블 탭 판정 스레드에서 사용하여 처리한다.

더블 탭 스레드는 'ACTION_UP'상태에서만 시작이 되고 'ACTION_DOWN'이면 정지된다. 더블 탭 스레드는 0.2초를 기준으로 진행을 한다. 그림 14의 흐름을 보면 롱 터치 판정 스레드로 도출된 변수를 통해 판정을 1차적으로 하고, 이후 'touchCount'를 기준으로 더블 탭 판정을 하며 롱 터치와 더블 탭이 아닌 경우 '상하좌우' 제스처로 판정을 한다. 이렇게 커스텀 터치기반 한 손가락, 두 손가락에 따른 기능이 구현된다. 마지막으로 'Shaking' 제스처는 번역 기능의 점자 입력에 사용되기 때문에 해당 화면에서만 구현을 진행한다. 'Shaking'은 스마트폰의 센서를 이용하고, 번역 기능에서만 사용하기 때문에 따로 커스텀화 하지 않는다.

이후 커스텀 터치가 적용되는 액티비티에 'CustomTouchListener'를 구현함으로써 인터페이스에 정의된 함수인 'onOneFingerFunction() 과 onTwoFingerFunction'을 호출할 수 있다. 이 두 개의 함수는 해당 액티비티 내에서 터치 이벤트가 발생시 앞서 설명한 'CustomTouch'클래스에서 로직이 수행된 후 반환되는 'Type'에 따라서 기능을 구현할 수 있다.

그림 16은 커스텀 터치 내에서 더블 탭을 판정하는 스레드 흐름도이다.

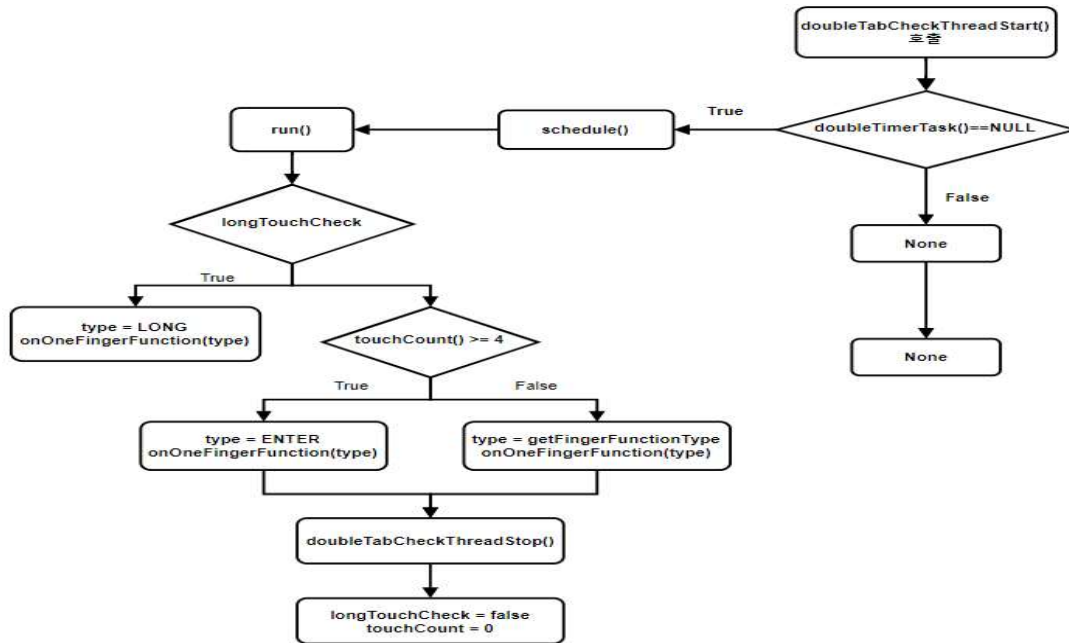


그림 16 CustomTouch - 더블 탭 판정 스레드 흐름도

표 6 커스텀 터치 관련 함수 설명표

모듈명	함수/변수명	설명
Custom TouchConnect Listener	void touchEvent (MotionEvent event)	기존의 touchEvent 를 커스텀하기 위한 인터페이스
Custom TouchEvent Listener	void onOneFingerFunction (FingerFunctionTypeefft)	FingerFunctionType을 인자로 받아 그에 맞는 기능을 수행
	void onTwoFingerFunction (FingerFunctionTypeefft)	
Finger Location	void setDownLocation (MotionEvent event, int fc)	MotionEvent 를 받아와 손가락 수에 해당하는 up/down 좌표를 저장
	void setUpLocation (MotionEvent event, int fc)	
FingerFunction Process	FingerFunctionTypeefgetFingerFunctionType(FingerLocation fl)	FingerLocation 에서 좌표를 받아서 각 좌표의 차이를 이용해 상하좌우 드래그를

		판정
FingerFunction Type	Int getNumber ()	정의된 TYPE 을 반환한다
CustomTouch Event	void touchEvent (MotionEvent event)	터치 이벤트 처리 함수
	void oneFingerDown (MotionEvent event, int fc)	한 손가락의 이벤트를 처리하는 함수
	void oneFiingerUp (MotionEvent event, int fc)	
	void twoFingerDown (MotionEvent event, int fc)	두 손가락의 이벤트를 처리하는 함수
	void twoFingerUp (MotionEvent event, int fc)	
	synchronized void doubleTabCheckThre adStart()	더블 탭 이벤트 판정 쓰레드
	synchronized void longTouchThreadStar t()	롱 터치 이벤트 판정 쓰레드

표6은 기존 터치 이벤트와 연결을 하기 위한 CustomTouchConnectListen
er를 인터페이스로 구성하고, 이를 CustomTouchEvnet 클래스 내에서 touc
hEvent 함수를 구현한다. 해당 함수 내에서 터치 이벤트 발생으로 인한 손가
락별 액션에 맞는 함수를 호출하여 이벤트를 처리한다. 이벤트를 처리하기 위
해서는 터치 이벤트를 발생시킨 손가락마다 좌표를 저장해야 하는데 이를 Fin
gerLocation 클래스에서 처리를 한다. 이 클래스는 Down 액션일 때 setDow
nLocation이 호출이 되고, Up 액션은 setUpLocation이 호출되어 저장이 된
다. 이 객체를 통해 FingerLocationProcess클래스에서 좌표마다 차이(Gap)를
이용하여 상하좌우의 제스처를 판정하고 Type을 반환한다. Type에 대한 것은
FingerFunctionType 이름의 enum 타입으로 정의한다. 이후 CustomTouch
EventListener를 제스처가 필요한 곳에 구현을 하여 반환되는 type에 따라 손
가락 별로 구현한다.

표 7 메뉴 & 음성 & DB 함수 설명 표

모듈명	함수/변수명	설명
MenuTree Manager	void setMenuImageTree	각 메뉴를 트리에 등록
	Menu getMenuName(Deque<Intege	해당 메뉴이름을 가져오는 함수

	r> deque)	
	TreeNode getMenuTreeNode(Deque<Integer> deque)	Deque 안에 있는 Index정보를 이용하여 Root노드부터 탐색 후 해당 노드를 반환하는 함수
MediaPlayer Singleton	void start(Queue<Integer> IdQue, String ttsText)	음성 API를 통해 tts 출력을 위한 함수
	void focusSoundStart()	포커스가 잡힌 음성을 출력하는 함수
	boolean checkMediaShield)	현재 음성이 멈추지 않게 하기 위한 함수
Permission CheckModule	int checkPermission()	권한 사용 여부를 판단하는 함수
	void startPermissionGuide(int permissionCheck)	권한 사용 안내를 위한 함수
DBHelper	void insert(String letterName, String dotMatrix)	데이터베이스에 데이터를 저장하는 함수
	ArrayList<OnDotData> delete(String dotMatrix)	데이터베이스에 저장되어 있는 데이터를 삭제하는 함수
	ArrayList<OnDotData> getResult()	데이터베이스에 저장되어 있는 데이터를 불러오는 함수
DBManager	void save(String letterName, String dotMatrix)	DBHelper를 통해 저장하려는 점자 데이터를 전달하는 함수
	ArrayList<OnDotData> delete(String dotMatrix)	점자데이터를 DBHelper로 현재 화면 점자 정보를 전달하여 삭제하는 함수

표7은 '온점' 어플리케이션의 메뉴, 음성 그리고 DB 함수를 설명한 표이다. 메뉴는 트리형식으로 구성하여 메뉴 탐색에 용이하게 MenuTreeManager 이름의 클래스로 구성한다. 즉, Deque로 이루어진 데이터를 이용하여 메뉴 이름, 값을 가져오는 함수로 기능을 수행한다. 카카오 음성 API를 가져와서 음성을 출력을 처리하는 것이 MediaPlayerSingleton이며, 이에 따른 권한 여부관련된 모듈이 PermissionCheckModule 이다. 데이터베이스 연동과 데이터 삽입, 삭제, 수정을 위한 DBHelper와 이를 다루는 DBManager로 구성되어 있다.

음성 API는 '카카오 I'의 음성 API를 사용하여 구현을 진행한다. 음성API에 존재하는 음성인식(STT)과 음성합성(TTS)으로 구현이 되며, 처음 화면에서는 기본적인 안내 음성이 출력이 되며 각 기능별로 음성안내가 별도로 출력된다. 이 음성안내는 별도의 음성 파일로 저장되어 해당 기능에 따라 출력이 되도록 구현한다. 대체적으로 음성합성(TTS)으로 구성되어 기존의 음성 파일이나 텍스트를 통해 안내 음성 등을 알려주는 기능을 하고, 퀴즈 답 말하기, 음성게시

판 이용은 음성인식(STT) 기능을 사용하게 된다.

음성인식 및 합성은 위험 권한인 위험 권한인 'RECORED_AUDIO, WRITE_EXTERNAL_STORAGE'를 사용자에게 동적으로 권한을 요청해야 한다. 그에 맞는 권한 안내 음성을 들려주어 앞서 설명한 커스텀터치를 이용하여 허가할 수 있게 진행을 한다. 따라서 권한 요청을 할 수 있는 모듈이 필요하다. 본 어플리케이션에서는 'PermissionCheckModule'이라는 클래스로 정의를 하였고, 해당 클래스 내에서 권한사용 여부를 판단, 권한 사용 안내, 권한 요청 화면에 대해 구현하였다. 제스처를 이용하여 기능을 수행하기 때문에 권한 허가 또한 사용자가 커스텀 제스처를 통해 할 수 있도록 설계하였다.

음성인식(STT)을 처리하는 모듈에서는 처음에는 카카오에서 제공하는 SDK를 초기화하는 작업을 진행하고, 'callback' 메서드를 받기 위한 'SpeechRecognizeListener' 인터페이스를 구현함으로써 음성인식 로직을 구현한다. 음성인식을 진행한 후 SDK의 함수중 하나인 'onResults()'에서 지금까지 인식된 음성을 카카오 서버에 질의하는 과정까지 마치고나면 ArrayList를 통해 완전히 인식된 문장이나 단어를 반환받을 수 있다. 다양한 음성인식 오류처리 및 기타 이벤트의 제어도 가능하다.[15]

음성합성(TTS) 또한 음성인식과 같이 초기화 함수 'initializeLibrary()'를 호출하여 초기화 작업을 진행하고, 'callback' 메서드를 받기 위한 'TextToSpeechListener'를 구현한다. 음성인식과 달리 음성합성은 중간 결과가 없이 'onFinished'와 'onError'이벤트 밖에 없다. 모든 'callback' 함수는 'UI thread'가 아닌 'background thread'에서 호출될 수 있기 때문에, UI와 관련된 작업은 'Activity.runOnUiThread(Runnable)' 이나 'Handler.post(Runnable)'을 통해 비동기로 호출하여 구현한다. 에러처리를 위해 'onError' 함수를 통해 처리하며 음성합성이 종료된 후 'onFinished' 함수를 통해 메모리를 정리한다. 카카오 음성 API에서 제공되는 'SpeechMode'와 'VoiceType'을 적절한 것을 택하여 음성 합성을 진행한다.[16]

데이터베이스와 연동을 위한 'DBHelper'클래스와 이를 이용하는 'DBManager'클래스를 설계하였다. 'DBHelper' 클래스는 데이터베이스에 직접적으로 데이터를 삽입, 삭제, 수정의 기능이 구현되고, 'DBManager'클래스는 어플리케이션에서 가공되는 점자 데이터를 'DBHelper'객체로 전달하는 기능을 수행한다. 즉, 'DBManager' 클래스내에서 'DBHelper'객체의 함수를 이용하여 데이터베이스에 저장되어 있는 데이터를 가공할 수 있다. 또한 'DBHelper' 객체를 싱글톤(Singleton) 패턴으로 구현하여 오직 하나의 객체만을 사용하도록 설계한다.

하드웨어 상세설계에 대해서 설명한다. 그림 17은 점자키트를 3D 모델링으로 표현한 그림이다. 이 모식도는 3개의 점자를 표현하지만 실제 모델링에서는 8개의 점자를 표현할 예정이다. 그림 17처럼 점자가 올라온 부분과 내려간 부분으로 실제 점자와 같이 시각장애인이 손가락으로 만져서 인식할 수 있는 점자키트를 만든다.

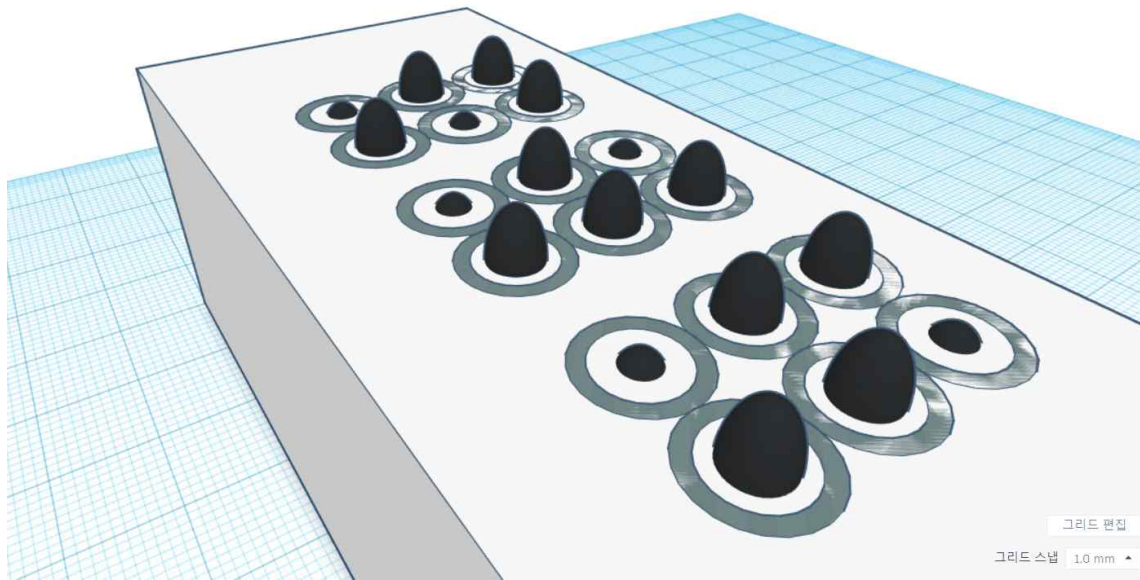


그림 17 점자키트의 모식도

그림 18과 19는 3D 모델링을 통해 점자를 표현할 그림이다. 기본적으로 2부분으로 나누어져 있으며 캡은 시각장애인이 점자를 직접 만지는 부분이고 가이드는 그 캡을 지지해주고 위 아래로 표현하는 틀 역할을 하는 부위이다 두 부위가 서로 상호 보완적으로 움직이면서 점자를 표현한다.

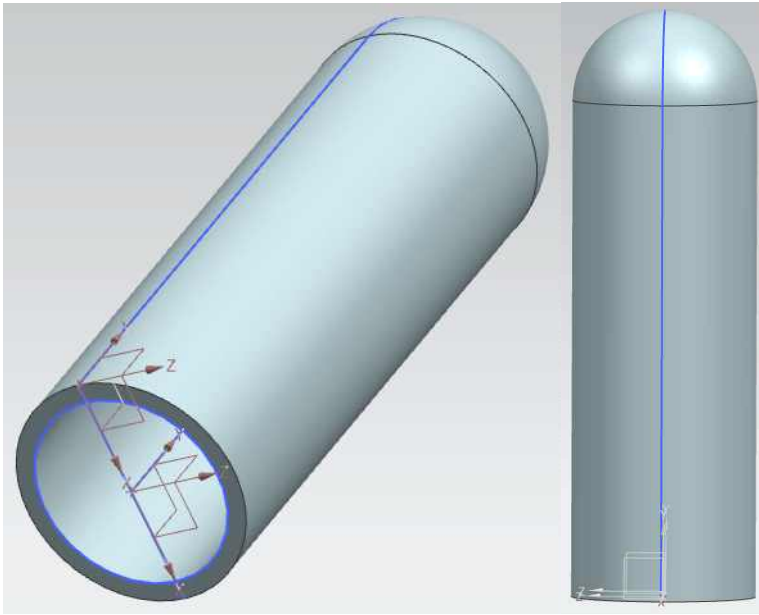


그림 18 점자를 표현하는 캡의 3D 모델링

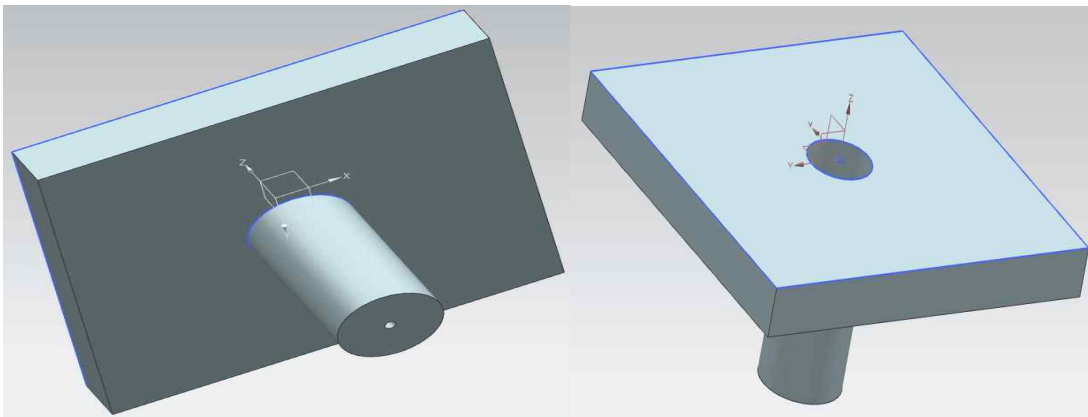


그림 19 점자를 표현하는 가이드의 3D 모델링

그림 20과 21은 점자를 표현할 3D모델링의 프로토타입이다. 총 8개의 점자를 표현하며 각각의 점자에는 6개의 가이드와 6개의 캡이 아두이노의 서보모터와 상호작용하며 점자를 표현한다.

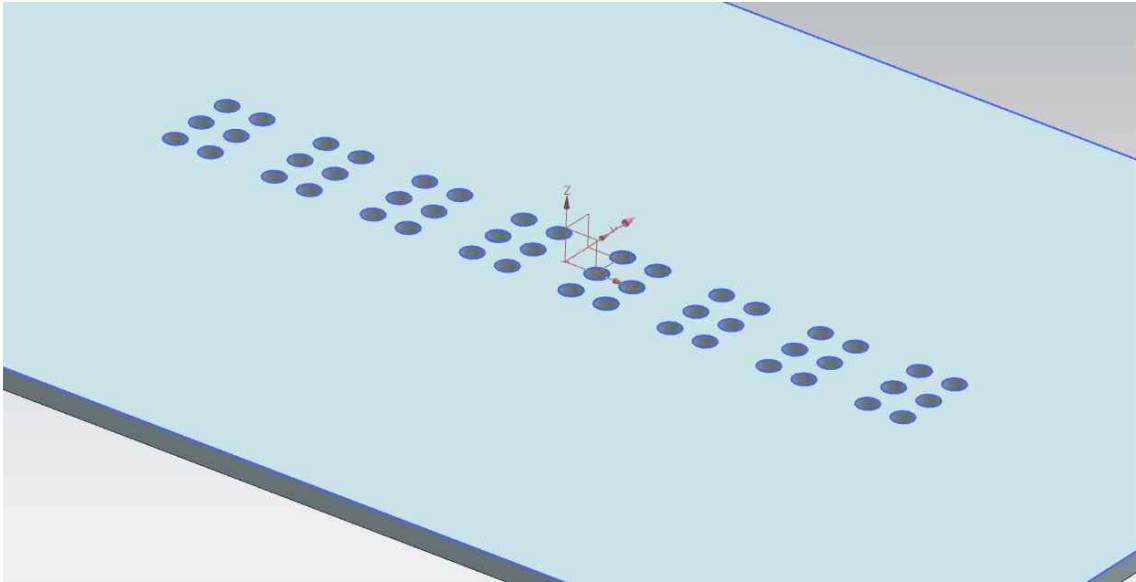


그림 20 점자를 표현할 3d모델링 프로토타입 앞면

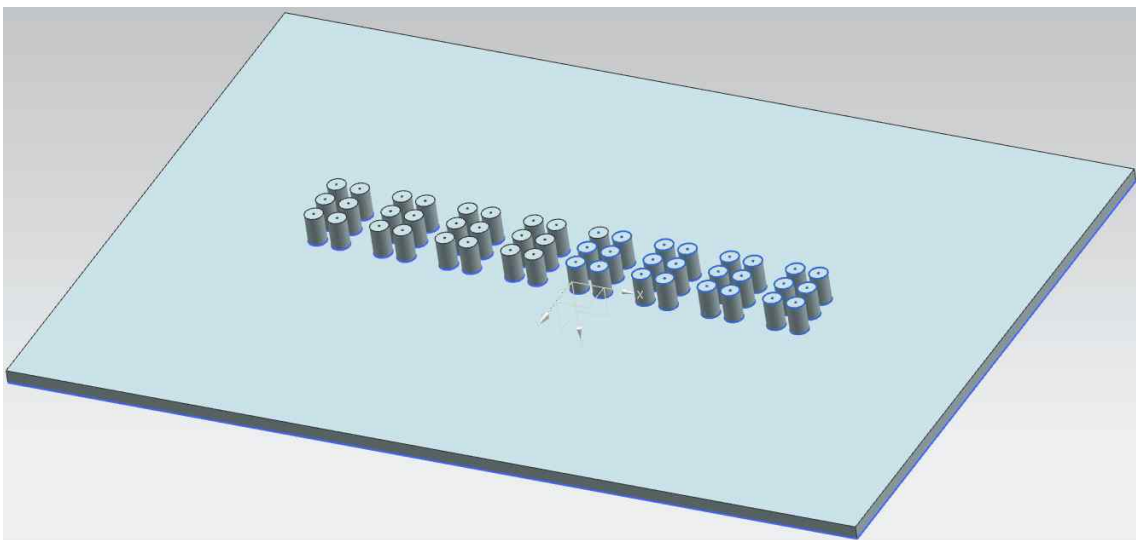


그림 21 점자를 표현할 3d모델링 프로토타입 뒷면

그림 22은 점자 디바이스의 전체 흐름도를 나타낸 흐름도이다. 아두이노 보드에 소스코드를 업로드해 프로그램을 실행하면 `setup()`에서 PWM 보드의 주소설정과 주파수설정을 하고 I2C통신을 하기위한 초기화를 하고 블루투스 모듈을 입력대기 상태로 만들어 둔 후 대기상태로 들어간다. 블루투스 연동이 성공하면 점자데이터가 외부 애플리케이션에서 인가되고 서보모터에서 점자를 표현한 후 서보모터를 다시 원상태로 복귀시킨다 그리고 점자가 8개 이상인 경우엔 점자 데이터의 잔류여부를 활용해서 단어의 끝까지 표시해주는 기능을 넣는다. 단어를 전부 표현하면 다시 점자 데이터가 입력되기 전의 상태인 입력

대기상태로 대기한다.

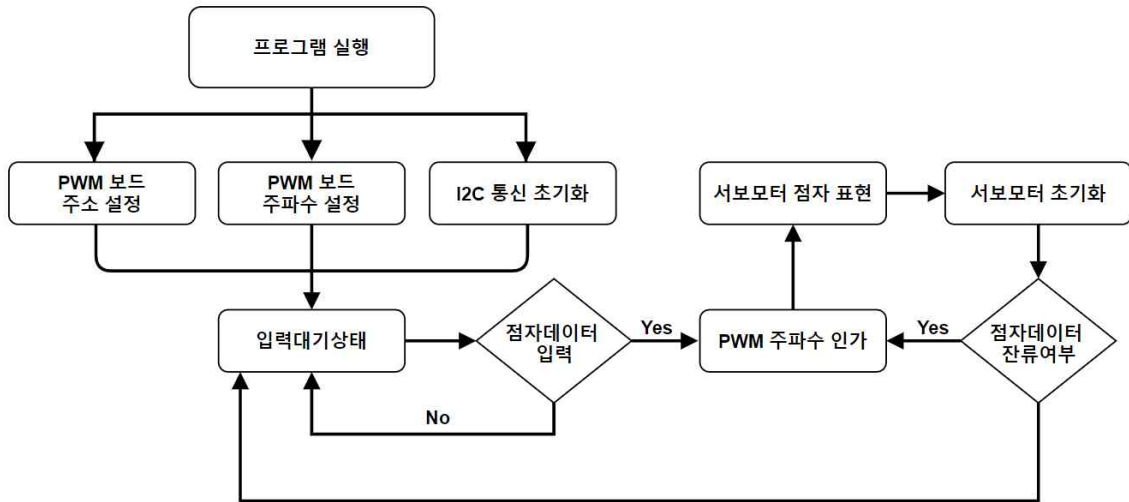


그림 22 점자 디바이스의 전체 흐름도

그림 23은 서보모터가 점자를 표현하는 흐름도이다. 서보모터는 PWM의 주파수로 움직이기 때문에 초기상태에서 최소/최대 pulse와 모터 각도를 초기화 시켜서 대기상태에 놓여진다. 점자 데이터가 인가되면 점자 데이터가 점자 8개를 초과하는지 검사 한 후 초과하면 임시배열에 초과분의 점자데이터를 저장해두고 나머지 점자 데이터를 서보모터에 PWM 주파수 인가를 통해서 표현한다. angleTopulse()를 호출하여 PWM 주파수를 각도로 변경한다. 최대각과 최소각의 표현을 하기위해 map() 함수를 호출하여 알맞은 숫자로 변형시킨 후 서보모터의 각도를 설정한다. 점자를 표현한 후 남은 데이터를 확인하여 있으면 같은 동작을 반복한다.

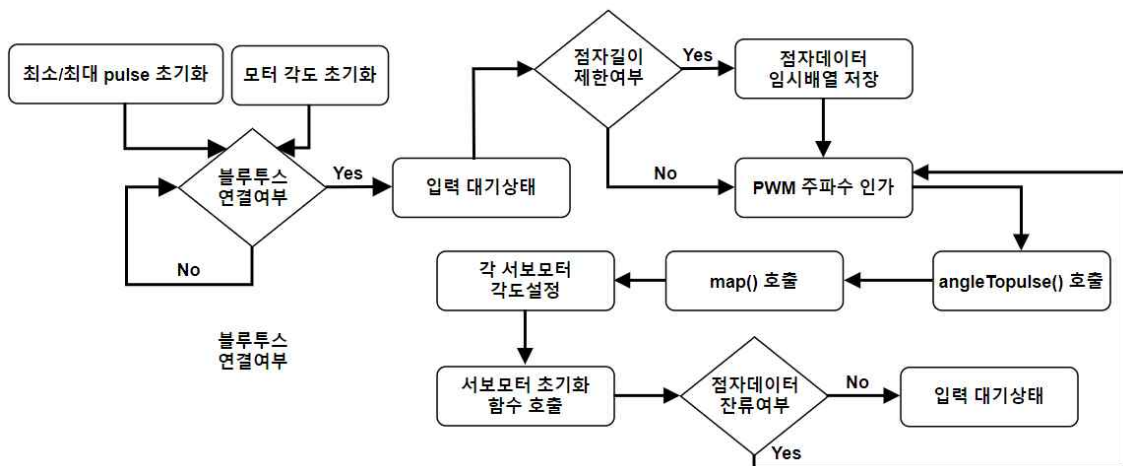


그림 23 점자를 표현하는 서보모터 Flow-chart

표 8 점자키트 관련 함수표

	모듈명	함수/변수명	설명
아두이노 함수	서보모터	Adafruit_PWMServoDriver.h	서보모터 동작 헤더파일
		int angle / int angleStep	서보모터 각도
		int servoNumber	서보모터 개수
		angleToPulse(int angle)	각도를 Pulse로 변환
		map(int angle,int min, int max, SERVOMIN,SERVOMAX)	숫자 범위 변환
		int pulse	Pulse 변수
		define SERVOMIN 100 define SERVOMAX 600	최소/최대 Pulse
	블루투스 모듈	SoftwareSerial.h	블루투스 모듈 동작 헤더파일
		int Tx / int Rx	Rx/Tx 변수
		SoftwareSerial mySerial(int Tx, int Rx)	시리얼 통신을 위한 객체선언
		mySerial.available()	블루투스 활성화
	PWM모듈	Adafruit_PWMServoDriver(int address)	PWM 객체 선언
		PWM.begin()	PWM 보드 시작 및 초기화
		PWM.setPWMFreq(int frequency)	PWM 보드 주파수 초기화
		PWM.setPWM(int pinNum, int on, angleToPulse(int angle))	PWM 주파수 인가

표 8은 아두이노에서 사용하는 함수를 정리한 표이다. 서보모터는 최소/최대 각도를 설정한 후 pulseToangle() 함수를 통해서 점자 표현에 맞는 각도로 변환시킨다. map() 함수를 이용하여 해당 서보모터에 맞는 각도로 변환시킨다. 각도를 변화시킨 후 PWM모듈 전용 함수에서 setPWM() 함수를 통해 48

개의 서보모터를 제어한다. setPWM() 함수는 48개의 서보모터를 한번에 제어할 수 있는 함수이다. PWM 모듈은 병렬적으로 구성되어 있어서 600개 이상의 서보모터도 한번에 제어할 수 있다는 장점이 존재한다.

표 9 데이터베이스 상세설계 표

	테이블명	컬럼명	자료형	용도
DB	initial_dots 초성 종성 모음 테이블	id(primary key) word type dots	int(auto) string int int	주키 문자 문자 구분 매핑 숫자
	word_dots 단어 테이블	id(primary key) word dots	int(auto) string int	주키 단어 매핑 숫자
	translatelog 많이 번역한 단어	id(primary key) word dots count	int(auto) string int(auto)	주키 단어 매핑 숫자 카운트 횟수
	wronglog 많이 틀린 단어	id(primary key) word dots count	int(auto) string int(auto)	주키 단어 매핑 숫자 카운트 횟수
	boards 게시판	id(primary key) title contents articlenum viewnum date	int(auto) string string int int date	주키 글 제목 글 내용 글 번호 조회수 게시날짜
	account 계정	id(primary key) admin_id admin_password	int(auto) string string	주키 관리자 id 관리자 pw

표 9은 ‘온점’의 DB 상세설계 표이다. 데이터베이스에서는 가장 중요한 점자관련 테이블이 주된 입출력 대상이다. 그리고 사용자의 패턴을 알 수 있는 테이블과 웹에 관련된 테이블로 이루어져 있다.

DB의 상세 설계는 크게 6개의 테이블로 나눌 수 있다. 먼저 초성, 종성, 모음이 있는 테이블이다. 숫자 1과 2를 사용하여 점자의 돌출부와 비 돌출부를 표현한다. 예를 들어, 초성 ‘ㄱ’은 6개의 점자 중 2번 점자만 돌출되어있는 형

태이고, 이는 121111로 표현된다. 단어 테이블은 초성, 종성, 모음이 있는 테이블과 같은 방식으로 데이터를 단어 단위로 저장한다. 많이 번역한 단어와 많이 틀린 단어 테이블은 어플리케이션의 번역과 퀴즈의 기능 중 사용자가 많이 틀린 단어와 많이 번역한 단어를 별도로 저장하는 테이블로 사용자가 점자를 번역하거나 퀴즈를 틀렸을 때 카운트 횟수를 증가시킨다. 해당 데이터는 웹 페이지에 Ranking 시스템으로 활용되어 시각 장애인이 어려워하는 단어를 알려 줄 수 있다. 게시판 테이블은 어플리케이션의 음성게시판 기능을 통해 게시글을 작성할 때 필요한 데이터를 저장한 테이블이다. 계정 테이블은 관리자의 id와 password를 저장하는 테이블이고 사용자는 login 없이 사용하므로 사용자의 id와 password는 저장하지 않는다. 관리자는 음성 게시글을 삭제 할 수 있다.

표 10 웹페이지 상세설계 표

	모듈명	함수/변수명	설명
JSP	DBConnection	String DB_URL	DB 주소를 저장한 변수
		String DB_USER	DB 아이디를 저장한 변수
		String DB_PASSWORD	DB 비밀번호를 저장한 변수
		Connection conn	DB와 연결하는 객체
		Statement stmt	쿼리를 보내기 위해 필요한 객체
		ResultSet rs	테이블의 형태로 갖게 되는 객체
		PreparedStatement ps	가변적 쿼리 처리를 위한 객체
	board	public int write_board	게시글을 작성하는 함수
		public int update_board	게시글 DB를 업데이트
		public ArrayList <BoardBean> getList	DB에 있는 게시글 받아오기
		public int delete_board	게시글을 삭제하는 함수
		public int hit	게시글의 조회수를 보여주는 함수
	Quiz	public int write_quiz	많이 틀린 퀴즈를 순위화 하여 작성하는 함수
		public int delete_quiz	많이 틀린 퀴즈 순위를 삭제

	Translate		하는 함수
		public int write_translate	많이 번역한 단어를 순위화하여 작성하는 함수
		public int delete_translate	많이 번역한 단어 순위를 삭제하는 함수

표 10은 ‘온점’의 웹페이지 상세설계 표이다. 웹페이지는 JSP로 구축하며[13] DB와의 연결을 담당하는 DBConnection을 통해 DB의 주소와 아이디 비밀번호를 저장하고 쿼리문을 통해 데이터를 받는다. 웹페이지에는 음성게시판, 많이 틀린 퀴즈, 많이 번역한 단어 등 3가지의 기능이 있다. 음성게시판 기능은 write_board 함수를 사용하여 “insert into board(title, contents, articlenum, viewnum, date) values(?,?,?,?,?)”의 형태로 작성하고 getList 함수에서는 “select id from board order by id desc”와 “select * from board where id < ? order by id desc limit 10” 등의 sql 문으로 페이지를 나눈다. delete 함수에서는 “delete from board where id = ?”와 같이 Primary 키를 사용하여 삭제한다. hit 함수에서는 “update board set viewnum = viewnum + 1 where id = ?”를 통해 조회수를 업데이트 한다. 많이 틀린 퀴즈와 많이 번역한 단어 기능은 write 함수를 통해 “insert into Quiz(word) value (?) order by count desc”를 통해 count가 많은 것부터 내림차순으로 작성한다. delete 함수에서는 “delete from Quiz where id = ?”와 같이 Primary 키를 사용하여 삭제한다.

5. Prototype 구현

5.1 구현현황

표 11 구현 현황표

모듈명	세부모듈	진도/전체진도		문제점	해결방안
아두이노 (점자키트)	모터 및 PWM 제어	80%	80%	1개의 점자만 나타내는 Prototype은 원하는 결과를 도출함, 하지만 8개의 점자를 나타내는 완성품의 테스트 미실시	정교한 수치설정 및 시행착오를 거친다.
	블루투스 연동	80%			
	3D 모델링	80%			
애플리케이션	사용자 인터페이스	80%	67%	안내음성 화면과 퍼미션 화면 구현미흡	사용자 입장에서 테스트를 거쳐 맞는 화면을 구성

	커스텀 제스처		70%		정밀한 제스처 컨트롤 의 어려움	시행착오를 통한 테스트
	음성 API		60%		각 메뉴별로 STT 기능 적용의 어려움	각 기능별로 적합한 STT 모듈화 실시
	블루투스 연동		80%		다양한 점자의 데이터 송수신 미실시	점자키트 완성품과 테스 트 예정
	어플 기능	교육	80%		STT 기능 적용의 어려 움으로 인한 음성게시 판 구현미흡과 어려가 지 예외사항 발생	점자키트 완성 후 테스 트
		퀴즈	60%			
		번역	60%			
		음성게시판	50%			
웹서버 /DB	데이터베이스 설계 및 구현		80%	63%	다수의 데이터 필요	알맞은 다수의 점자데이 터 삽입
	프론트엔드		50%		세부 디자인 구현 미흡	세부 CSS적용
	백엔드		60%		유효성검사 미실시 음성게시판 미구현	유효성검사 구현 및 App의 음성게시판 구현 후 순차적 개발
전체			70%			

프로토타입 구현현황으로 하드웨어는 8개의 점자를 표현하는 점자키트의 모델링을 완료하였고 데모용으로 제작한 1개의 점자(점6개)를 표현하는 프로토타입의 3D프린팅을 완료하였다. 하드웨어 제어부분에서 아두이노와 어플리케이션간 블루투스 통신을 통해 점자 표현을 담당하는 모터를 제어하는 코드의 작성 완료하였다. 그리고 해당 블루투스 코드를 테스트하여 블루투스 기능이 정상적으로 작동하는지 확인하였다. 어플리케이션의 구현현황으로 앞서 설계한 커스텀터치 방식대로 진행이 되었다. 커스텀터치의 문제점으로 사용자의 터치 입력이 민감하게 반응하는 경우가 있어서 예외처리가 필요한 상황이다. 그리고 시각장애인뿐만 아니라 비장애인들도 사용할 수 있는 UI를 구성하였고 실제로 시각장애인도 터치와 음성만으로 기능을 사용할 수 있게 구현하였다. 음성 TTS 모듈을 통해 각 메뉴별로 메뉴의 사용법을 알려주는 안내음성을 출력하는 기능을 구현하였고 사용자의 제스처마다 알맞은 음성이 출력되도록 구현하였다. 또한 점자학습을 실시할 때 현재 학습중인 데이터에 대해서 음성으로 출력을 하는 기능을 구현하였다. 각 메뉴에서 사용될 데이터를 WAS를 통해 XML형식으로 송수신하는 코드 작성을 완료하였고 프로토타입을 통해 데이터

의 입출력 기능을 확인하였다. STT기능을 사용하는 메뉴에서 세부적인 로직이 완성되지 않아 통합테스트를 할 수 없는 상태이고 대부분의 시각장애인들이 사용하는 접근성 기능이 안드로이드 접근성 가이드라인을 준수해야 하는데 기능 구현에 초점을 맞추고 개발을 진행한 상황에서 접근성 가이드라인을 준수하지 못한 부분이 발생하였다. 따라서 접근성 기능을 구현하여 통합시키는 방향으로 진행할 예정이다. 웹서버의 구현현황으로 AWS의 EC2와 RDS의 MySQL을 이용하여 웹서버와 데이터베이스를 구축하였다. 데이터베이스의 스키마를 완성하여 데이터의 삽입을 완료하였고 웹서버와 데이터베이스를 연동하여 데이터를 송수신하는 로직을 MVC패턴 형태로 구현하였다. 또한 어플리케이션과 데이터 통신을 위해 해당 데이터를 XML형식으로 변환하는 로직을 구현하였다. 웹페이지의 개괄적인 프론트엔드를 구성하였고 많이 틀린 점자와 많이 번역한 점자를 랭킹화하여 웹페이지에 표현하였다. 웹페이지 음성게시판은 어플리케이션의 음성게시판 기능이 구현되는 시점에 개발할 예정이다.

5.2 하드웨어 Prototype

그림 24는 하드웨어 회로도이다. 48개의 서보모터를 제어하기 위해 PWM 모듈 3개가 필요하다. 각 PWM 모듈당 16개의 모터를 제어할 수 있다. 모터의 전원은 아두이노의 5V로 부족하기 때문에 외부 전원(건전지)을 통한 모터의 전압 공급을 해주었다. 어플리케이션 블루투스 통신을 위해 블루투스 모듈도 장착하였다.

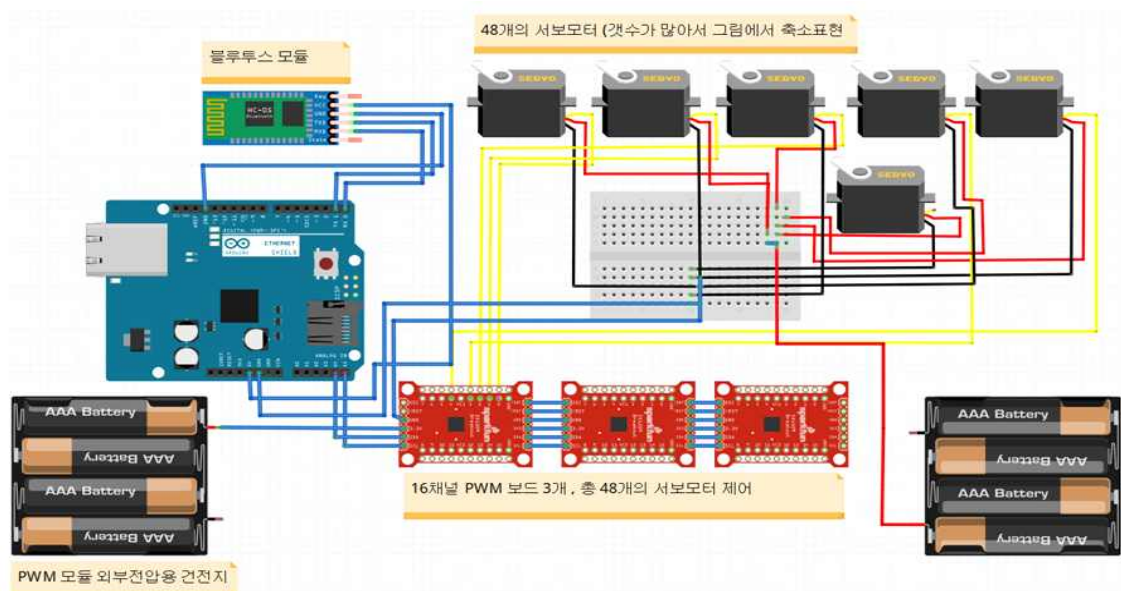


그림 24 하드웨어 회로도

그림 25와 26은 점자키트의 프로토타입이다. NX10.0을 사용하여 3D모델링을 통해 구현한 점자키트의 Prototype은 서보모터를 2개씩 3층으로 쌓아 6개의 서보모터를 한 묶음으로 하여 하나의 점자를 표현한다. 서보모터에 낚시줄을 연결하고 점자의 스프링부분과 연결한다. 서보모터의 회전범위를 고려하여 폼보드를 쌓아 회전범위를 확보하였다. 점자에 달린 낚시줄을 당길 때 사선으로 당겨지는 것을 방지하기 위해서 ㄱ자 경첩을 사용하여 수직으로 낚시줄을 당겨 점자 표현의 명확성을 확보하였다.

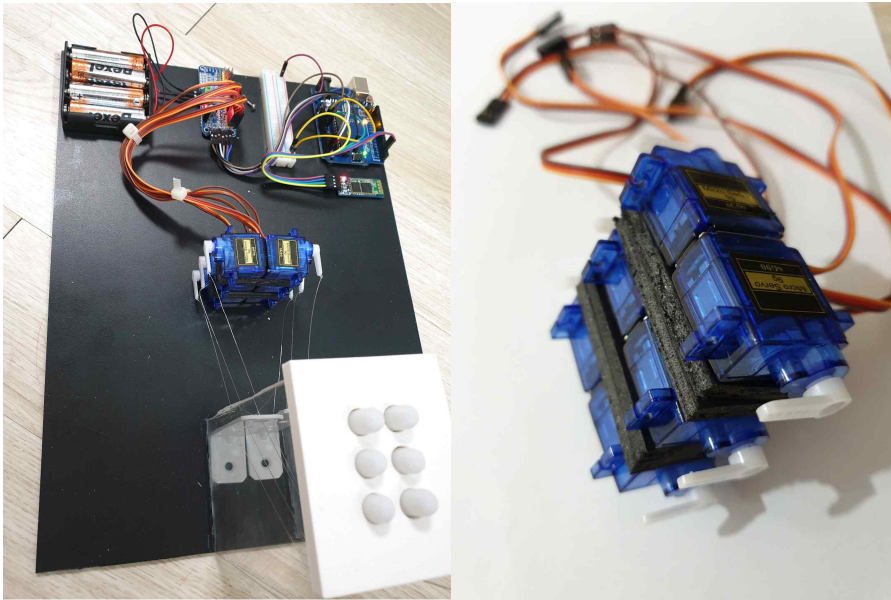


그림 25 점자키트 프로토타입 내부 그림 26 점자 1개에 대응하는 서보모터

5.3 어플리케이션 Prototype

그림 27은 어플리케이션의 화면이다. 시각장애인은 음성과 터치만으로 UI를 조작하기 때문에 화면 구성이 간단한 특징을 가지고 있다.



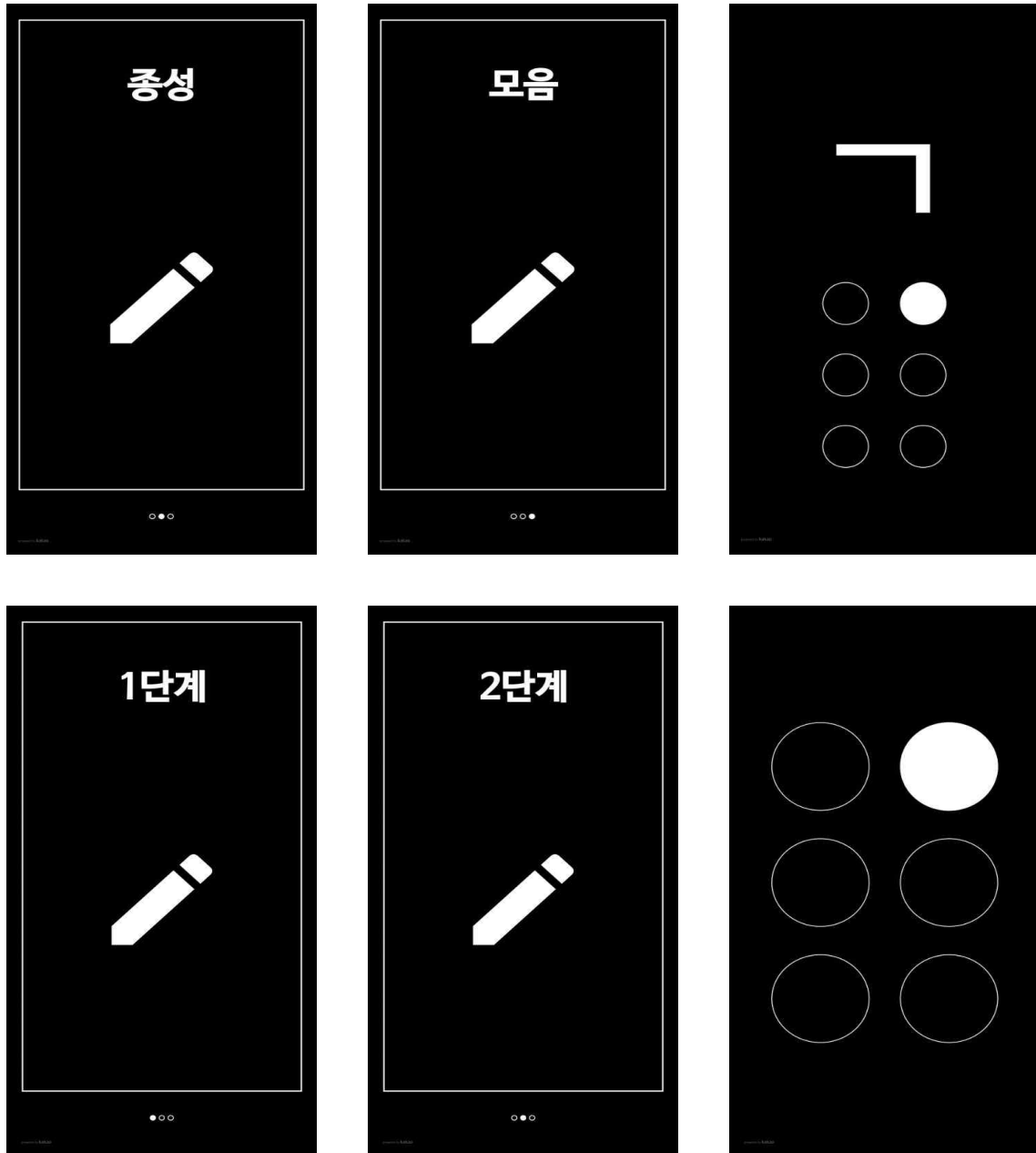


그림 27 어플리케이션 프로토타입의 화면구성

어플리케이션의 시작은 온점의 로고를 포함한 Splash 화면으로 시작한다. 각각 기능에 대한 화면을 제스처로 넘기면서 음성 안내가 출력된다. 시각장애인이 사용하는 만큼 일반적인 어플리케이션에서 사용하는 버튼이 존재하지 않고 더블탭과 상하좌우 드래그로 메뉴 이동을 할 수 있게 구현하였다.

교육 메뉴에서는 초성 종성 모음으로 나누어져 있고 실행 시 교육을 실시하는 초성, 종성, 모음이 텍스트와 점자로 표시되고 음성으로 안내를 실시한다. 교육을 받는 사용자는 음성과 점자키트에 표시된 사항을 귀와 손으로 학습한다. 퀴즈 메뉴에서는 단계별 학습을 위해 단어의 수준을 선택할 수 있고 각 단계별로 기능 실행시 교육 메뉴와 비슷한 화면이 표시되면서 음성으로 퀴즈를 실시한다는 안내를 통해 사용자에게 퀴즈의 시작을 알린다. 그리고 점자키트에 점자가 표시되고 사용자가 점자를 읽어 어떤 점자인지 음성으로 말하고 체점을 받는다. 번역 메뉴에서는 올라가는 제스처는 점자의 돌출된 부분을, 내려가는 제스처는 점자의 평평한 부분을 사용자로부터 입력을 받아 점자의 뜻을 음성으로 안내를 시켜준다. 음성게시판 메뉴에서는 제목 및 내용을 음성안내를 통해 진행방법을 안내한 뒤 사용자로부터 음성입력을 받아 글을 작성하게 한다. 작성된 글은 웹페이지에서 다양한 사람들이 확인할 수 있다.

5.4 웹 & DB Prototype

그림 28,29는 웹페이지의 화면이다. 웹페이지는 점자에 대한 간단한 소개와 어플리케이션의 실행에 따른 많이 틀린 퀴즈와 많이 번역한 단어가 랭킹화 되어있고, 음성게시판을 통해 작성한 게시글이 게시되어 있다.

메인화면은 점자에 대한 역사 및 점자의 구성이 어떻게 이루어져 있는지 소개하는 내용을 담고 있다. 왼쪽 메뉴바를 통해 각 메뉴를 이동할 수 있으며 많이 틀린 퀴즈와 많이 번역한 단어를 확인할 수 있는 메뉴들이 존재한다. 각 데이터들은 구축된 데이터베이스에서 추출해 웹페이지에 목록화 시켜준다. 음성게시판도 이와 같이 사용자로부터 입력받은 데이터를 받아와서 게시판 형태로 나타내준다. 졸업작품에 대한 개요 및 상세정보를 확인할 수 있는 메뉴도 존재하며 Github 링크 및 온점의 특징점을 확인할 수 있는 페이지와 점자에 관련된 상세한 페이지를 확인할 수 있는 점자협회의 홈페이지로 이동이 가능한 메뉴도 있다.

온점

시각장애인을 위한 점자 교육 시스템

[점자란?](#)

[점자랭킹](#)

[점자퀴즈결과](#)

[점자게시판](#)

[출업작품 정보](#)

[GitHub](#)

[출업작품 개요](#)

[점자협회](#)

점자의 의미

점자는 지면이 볼록 튀어나오게 점을 찍어 손가락끝의 촉각으로 읽도록 만들어진 특수 문자이다. 점자는 시각장애인이 지식과 정보를 습득할 수 있는 가장 중요한 도구이며 세상과 소통하는 통로이자 세상을 보는 눈이다. 점자는 프랑스의 시각장애인 루이 브라유(Louis Braille, 1809~1852)가 창안한 6점식 점자 체계에 근거한다. 6점식 점자는 현재 자국어 점자를 만들며 사용하는 모든 국가에서 공통으로 채택하고 있는 점자 체계이다. 이 6점식 점자 체계는 국제영어점자위원회(International Council on English Braille: ICEB)가 1993년 통일 영어 점자(Unified English Braille: UEB)를 규정할 때 제시한 대안책이기도 하다. 6점으로 만들어 낼 수 있는 점형의 수는 총 64개(2⁶)이다. 그중에서 점이 하나도 찍히지 않은 경우를 제외하고 점의 개수와 위치를 조합한 점형 63개를 이용해 국어, 수학, 음악, 과학 등 다양한 분야의 문자와 기호를 표기한다.

점형은 점의 개수와 위치로 구별되는 점의 모양이다. 점의 위치는 점형을 구별하는 매우 중요한 요소이므로 한 칸을 구성하는 각각의 점에는 번호가 매겨져 있다. 왼쪽 위에서 아래로 1점, 2점, 3점, 오른쪽 위에서 아래로 4점, 5점, 6점으로 구분한다. 또 1점과 4점을 상단, 2점과 5점을 중단, 3점과 6점을 하단으로 구분한다.

1	●	●	4	← 상단
2	●	●	5	← 중단
3	●	●	6	← 하단

점자의 발명

세계 최초의 점자는 프랑스 통신 부대 장교였던 샤를 바르비에(Charles Barbier)가 군사용으로 개발한 12점형 점자이다. 바르비에에는 전쟁터에서 어두운 밤에 군사용 작전 명령문을 읽을 수 있는 방법을 고민하다가 12점으로 된 야간 문자(ecriture nocturne)를 만들었다. 야간에 등불을 켜면 적에게 위치가 노출될 수 있으므로 불빛 없이 손으로 만져서 암호를 해독할 수 있는 방법을 고안한 것이다.

바르비에의 12점형 점자를 더 간략하고 실용적인 체계로 개편한 사람은 프랑스의 시각장애인 루이 브라유(Louis Braille)이다. 바르비에에는 12점형 점자(세로로 6점, 가로로 2점)가 읽기 어려워 군사용으로는 실패했지만 시각장애인들에게 도움이 될 수 있으리라 생각하고 파리맹학교에 야간 문자를 소개하였다. 이때 파리맹학교에 재학 중이던 시각장애인 학생 루이 브라유(Louis Braille)는 야간 문자가 훌륭하나 세로 6점 배열이 손가락끝으로 읽기에는 너무 어렵다는 것을 발견하고 세로로 3점, 가로로 2점인 6점 점자를 창안하였다.

바르비에의 문자가 12점으로 발음과 대응하는 데 비해 브라유의 점자는 6점으로 글자와 대응하는 점이 다르다. 여섯 점으로 된 점형은 손가락을 움직이지 않고 한 번에 모든 점의 위치를 읽어 낼 수 있는 장점이 있다. 브라유는 1829년 자신의 점자 체계에 대한 논문을 발표하였고, 1937년에는 역사 교과서를 점자본으로 출판하였다.

그림 28 웹페이지의 점자소개 화면

온점

시각장애인을 위한 점자 교육 시스템

[점자란?](#)

[점자랭킹](#)

[점자퀴즈결과](#)

[점자게시판](#)

[출업작품 정보](#)

[GitHub](#)

[출업작품 개요](#)

[점자협회](#)

많이틀린 점자랭킹

1 김문송이	11 사과
2 김학진	12 우거지
3 문준혁	13 응
4 송민욱	14 여우
5 이세민	15 아이유
6 은정	16 산기대
7 출업작품	17 컴퓨터공학부
8 나무	18 박현욱
9 아버지	19 신동원
10 어머니	20 김진엽

Developer

2015154009 김학진 2015154014 문준혁 2015154021 송민욱 2015154027 이세민

Copyright © 온점.All right reserved

그림 29 웹페이지의 점자랭킹화면

5.5 데모 시나리오

데모는 EC2서버를 활성화 시키고 어플리케이션을 실행시키는 것으로 시작한다. 어플리케이션의 교육기능에서 초성 종성 모음 중 교육받고자 하는 영역을 선택하여 교육을 실시한다. 교육기능에서 해당 문자에 관한 음성안내가 출력된 후 점자키트에서 서보모터가 구동되며 점자를 표현하는 과정이 계속해서 진행된다. 이때 프로토타입의 점자키트는 1개의 점자만 표현할 수 있으므로 간단한 자음이나 모음만 표현이 가능하다. 사용자는 음성으로 안내된 문자와 점자키트에 표현된 점자를 확인하면서 자가학습을 진행할 수 있다. 웹페이지 프로토타입은 점자에 대한 간단한 소개와 시각장애인들이 어려워하는 점자의 랭킹을 확인할 수 있다.

5.6 프로토타입 테스트

데모 시나리오대로 알파테스트를 해본 결과 몇 가지 문제점이 발생하였다. 첫 번째로 사용자가 교육을 진행할 때 매우 빠르게 점자교육을 진행하면 데이터의 송신이 점자키트의 점자표현을 하는 시간보다 빠르게 이루어졌다. 따라서 점자키트에서 점자를 표현할 때 점자가 제대로 표시되지 않는 문제가 발생하였다. 그리하여 매우 빠른 사용자의 입력을 허용하기 위해 예외처리를 실시해야 하는 필요성이 있다. 두 번째로 블루투스 통신이 불안정한 경우가 종종 생겨서 데이터의 송수신이 원활하지 않았다. 따라서 블루투스 모듈을 고급화 시켜야 하는 필요성이 있다. 세 번째로 서보모터의 토크가 강력하여 6개로 이루어진 서보모터 묶음이 제대로 지지가 되지 않았다. 따라서 서보모터 묶음을 지지해 줄 지지대를 설치해야 할 필요성이 생겼다. 네 번째로 점자키트에서 점자를 표현하는 캡이 들어갔다 나오는 동작에서 스프링과 점자 캡 사이의 폭이 좁아 스프링이 점자 캡 내부에서 수축했다 이완될 때 캡의 내부 벽을 긁으면서 올라오는 현상이 발생하였다. 따라서 캡이 완벽하게 제자리로 돌아가는 것이 아니라 조금 더 밖으로 밀리는 현상이 발생하여 캡과 스프링의 간격을 넓혀야 하는 필요성이 있다. 다섯 번째로 실제로 시각장애인이 사용하는 것처럼 눈을 감고 테스트를 해본결과 예기치 못한 제스처 방향으로 원치 않은 결과가 나타나는 경우가 간혹 발생하였다. 따라서 입력한 제스처의 정확성을 높이는 방향으로 보완할 필요성이 있다.

6. 시험/테스트 결과

표 12 구현 현황 요약표

모듈명	세부모듈		진도/전체진도	
아두이노 (점자키트)	모터 및 PWM 제어		95%	98%
	블루투스 연동		100%	
	3D 모델링		100%	
애플리케이션	사용자 인터페이스		100%	94%
	커스텀 제스처		95%	
	음성 API		90%	
	블루투스 연동		95%	
	어플 기능	교육	100%	
		퀴즈	100%	
		번역	80%	
웹서버 /DB	데이터베이스 설계 및 구현		90%	90%
	프론트엔드		90%	
	백엔드		90%	
전체			94%	

표 13 테스트 결과표

평가 항목		문제점	해결여부	해결방안
App	블루투스 연결	블루투스 연결유지	○	메인 화면 이전에 블루투스 연동 실시
	음성 출력 & 제스처	안내음성 출력중 화면 전환을 실시하면 음성 중단이 안되는 현상	○	음성출력 이전에 현재 출력중인 음성을 중단하는 코드삽입
	데이터 파싱	Json 방식의 데이터 파싱기법을 사용했을 때 어려움	○	XML 방식의 데이터 파싱기법을 사용해 비동기적으로 데이터 파싱 성공
	번역 알고리즘	복잡한 점자 번역의 예외사항 처리	△	예외 처리를 하기 위해 점자 문법, 관용 등 다양한 조사를 통해 알고리즘의 완성도를 높인다.
Device	점자 표현력	점자 캡의 명확한 점자표현, 모터와 캡의 긴밀한 연관성	○	튼튼한 낚시줄을 사용하고 자연스러운 점자의 표현을 위해 낚시줄을 90도로 꺾는 지지대 배치
	모터 전압 분배	48개의 모터가 사용하는 전압의 안정적 조정	○	1.5V 건전지 12개를 사용하여 48개의 모터의 안정적인 전압을 공급함

표13은 테스트 단계에서 각 항목별 문제점 및 해결여부를 나타내는 표이다.

테스트는 어플리케이션과 하드웨어 각각의 모듈테스트와 두가지를 동시에 진행하는 통합테스트로 진행한다. 모듈테스트 항목에서는 어플리케이션의 기능들을 각각 테스트한다. 테스트 항목으로는 첫째, 블루투스 연결 테스트, 둘째, 음성 출력(TTS) 과 커스텀 제스처 관계테스트, 셋째, 데이터 파싱 테스트, 넷째, 번역 알고리즘 테스트이다. 점자키트의 모듈테스트 항목은 첫째, 점자 하나를 표현한 모델링으로 점자 표현의 가능성을 확인하는 테스트, 둘째, 점자 여덟 개를 표현한 모델로 점자를 표현하는 테스트로 구분한다. 통합테스트는 점자키트와 어플리케이션상의 데이터 입출력을 통해 어플에서 요구하는 기능들이 점자키트에서 충실하게 진행되는지 확인하는 과정을 거친다.

어플리케이션 모듈테스트

블루투스 연결은 교육과 퀴즈 메뉴에서 사용할 점자키트와 블루투스 연동은 해당 메뉴를 들어가기 전에 블루투스 연동 여부를 결정한다. 제일 먼저 블루투스 연동을 하고 교육 메뉴의 서브 메뉴인 초성, 종성, 모음, 특수기호 및 숫자, 줄임말을 동작시켜 테스트를 하였다. 그 결과 점자 데이터에 따른 점자 키트에 올바르게 표기가 되었으나 교육 메뉴에서 메인 메뉴로 이동하고 다시 교육 메뉴를 들어오면 블루투스 연동이 끊기는 현상이 자주 발생하였다. 퀴즈 메뉴 또한 같은 현상이 발생하였다. 이와 같은 문제를 해결하기 위해서 블루투스 연동 부분을 메뉴 화면에 들어가기 이전에 연동 여부를 결정한다. 해당 화면에서 연동을 하게 되면 교육 과 퀴즈 메뉴에서 동일한 블루투스 주소를 가지고 연결을 유지한다. 이후 테스트를 한 결과 메뉴 이동간의 연결 끊김이 사라졌다.

사용자 인터페이스 측면에서 여러 가지 상황을 고려하여 음성 출력(TTS)과 커스텀 제스처 사이의 관계를 테스트 하였다. 테스트 항목으로는 첫째, 각 화면 별로 필요한 터치 제스처만으로 구성되어 있는지, 둘째, 해당 터치 제스처에 맞는 음성이 출력되는지, 셋째 각 화면에 올바른 안내 음성이 출력되는지, 넷째 메뉴 이동시 음성 출력이 적절하게 끊기는지를 확인한다. 첫 번째 테스트 결과 메뉴 화면에서 한손가락 제스처 기능 중 좌, 우, 더블탭 만을 사용하는 메뉴에서 나머지 제스처까지 판정이 되는 것을 확인 할 수 있었다. 그래서 해당 메뉴에 필요한 제스처를 남기고 나머지는 인식이 되지 않게 막았으며, 판정 불가인 드래그에 대해서는 음성 출력이 되어 사용자에게 알 수 있게 수정하였다. 두 번째 테스트 결과 대부분의 제스처에 맞게 음성이 출력 되었으나 종종 원하는 제스처와는 다르게 엉뚱한 결과가 나오는 상황이 존재하였다. 그

래서 해당 제스처 부분의 오차 범위를 수정하여 완화시킴으로서 해당 문제를 해결하였다. 세 번째와 네 번째 테스트 결과 각 화면과 특정 기능별로 안내음성은 올바르게 출력이 되었으나 메뉴 이동시에 음성 출력이 중단되지 않았다. 이를 해결하기 위해 메뉴 이동시 해당 음성이 출력이 되기 전에 앞서 출력중인 음성을 중단시키고 이후 해당 음성을 출력시키는 방식으로 해결하였다.

데이터 파싱 테스트를 하기전에 고려해야할 부분이 있다. 교육과 퀴즈 메뉴는 데이터베이스에 있는 데이터를 이용하여 초성, 종성, 모음, 특수기호 및 숫자, 줄임말의 글자와 점자를 표시하도록 설계가 되어있다. 데이터 교환을 위하여 처음에는 JSON을 사용했다. 하지만 JSON을 사용하여 데이터를 파싱하는 도중에 데이터를 동기화하는 어려움이 있어서 비동기 방식으로 사용할 수 있는 XML로 데이터 교환방식을 바꾸었다. XML을 파싱하는 방식으로 변경 후 데이터의 교환에는 아무런 문제가 발생하지 않았다.

프로젝트에서 가장 복잡한 알고리즘을 요하는 메뉴는 번역 메뉴이다. 따라서 번역 메뉴의 과정은 사용자가 점자를 입력하면 점자를 텍스트로 번역을 해야 하는 메뉴이다. 데이터 번역에는 굉장히 많은 경우의 수가 있다. 이 경우의 수를 모두 만족시키며 예외가 없는 알고리즘을 만들기 위해서 많은 노력을 기울이고, 굉장히 많은 테스트를 진행했다. 테스트를 진행할 때마다 예외는 발생했고 그 수많은 예외를 잡아내기 위해서 많은 노력을 기울였다. 예를 들어 숫자 '8'과 초성 'ㅌ'은 수표를 제외하면 똑같은 점자이다. 숫자 '8'은 앞에 '수표'가 붙고, 초성 'ㅌ'은 그렇지 않다. 그렇기 때문에 번역을 할 때는 입력한 점자가 숫자 '8'인지, 초성 'ㅌ'인지 구별할 필요가 있다. 이러한 수많은 예외처리를 해야 한다. 현재 알고리즘을 어느 정도 완성은 했지만, 이후에 더 많은 테스트를 하며 예외처리를 하여야 할 것으로 생각되고, 계속해서 수정하는 과정을 거쳐야 할 것 이다.

테스트를 거치면서 설계단계에 포함되어 있던 게시판 메뉴를 삭제하였다. 이유는 여러 가지가 있다. 첫 번째 이유는 주제와 부합하지 않는다는 것이다. 이 프로젝트의 주제는 '시각장애인을 위한 점자 교육 시스템'이다. 즉, 시각장애인이나 저시력자가 스스로 점자를 학습할 수 있도록 하는 것이 가장 큰 목적이다. 하지만 테스트를 하고 회의를 통해 피드백을 할 때마다 게시판은 주제에 어울리지 않는 느낌이 강하게 들었다. 두 번째 이유는 STT의 부정확함이다. 테스트를 할 때마다 STT는 정확도가 꽤나 낮은 편이었다. 과연 이 기능이 프

로젝트에 포함된다면 시각장애인이나 저시력자가 편하게 사용할 수 있을까에 대한 의문이 들었다. 위와 같은 두가지의 이유로 인하여 설계단계에 포함되어 있던 게시판 메뉴를 삭제하였다.

점자키트 모듈테스트

점자키트 테스트에서 가장 중요하게 고려해야 할 점은 “점자의 표현력”이다. 시각장애인이 손으로 만졌을 때 명확하게 6개의 점으로 구성된 점자의 올라간 상태와 내려간 상태를 구분할 수 있어야 한다. 그러기 위해선 점자를 표현하는 서보모터와 점자캡의 연결을 지속가능하게 그리고 명확하게 구현해야한다. 점자의 표현력을 높이기위해 여러번의 시행착오를 겪었다. 가장 첫 번째로 점자 하나를 표현하는 작은 3D모델링을 지속적으로 수정 보완하면서 테스트를 진행하였다.

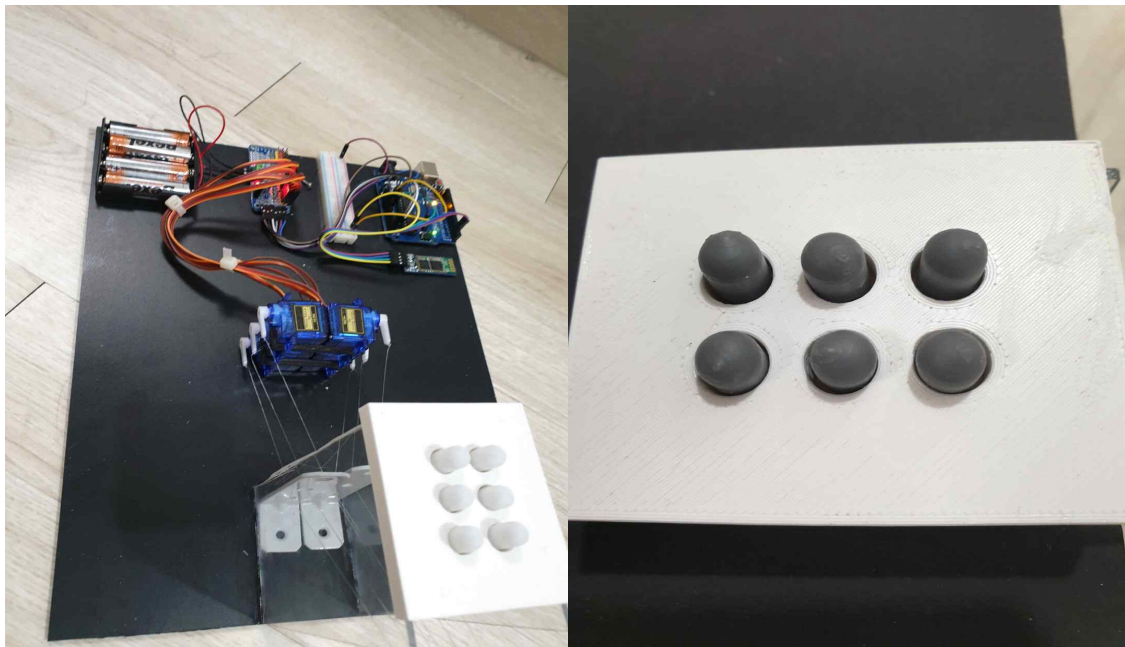


그림 30 점자키트 테스트를 위한 3D모형

그림 30은 하나의 점자를 통해 점자표현을 테스트하는 그림이다. 그림에서 볼 수 있듯이 서보모터와 각각의 캡 사이는 1대1의 관계로 납시줄이 연결하고 있다. 구성 초반에 일반 의류용 실과 치실을 이용 해봤지만 끊어지거나 늘어나는 문제가 생겼다. 그리하여 납시줄은 장력이 강력하여 끊어지지 않고 늘어나지 않는 특성이 있기에 납시줄로 선정하였다. 점자키트의 테스트 도중 발생한 문제점은 다음과 같다. 첫째, 납시줄과 캡 사이에 있는 스프링을 당길 때 내려오는 과정에서 스프링이 옆으로 휘어 벽을 긁으면서 내려오는 현상이 발생하

였다. 그에 따라 모든 캡이 일정하게 내려가지 않고 서로 각기다른 모습으로 내려가는 현상이 발생하였다. 둘째, 낚시줄이 서보모터와 묶이는 지점에서 풀림현상이 간헐적으로 발생하였다. 낚시줄은 플라스틱 재질로 되어있어서 묶을 때 다른 실처럼 완전히 묶이는 상황이 아니기에 문제가 발생하였다. 셋째, 점자 캡과 서보모터간의 간극 차이로 인해 낚시줄을 당기는 각도가 수직이 아닌 엉뚱한 방향으로 당기는 현상이 발생하였다. 그리하여 점자가 올라가고 내려갈 때 어색한 모양을 만들어서 문제점으로 지정하였다. 이 세가지 문제점을 해결하기 위해 다음과 같은 대책을 강구하였다. 첫 번째 문제를 해결하기 위해 점자 캡의 3D모델링을 대대적으로 수정하였다.



그림 31 점자 캡의 실제모형과 3D모델링

그림 31은 수정된 점자 캡 3D모델링이다. 본래의 방식은 스프링을 캡 안에 넣어서 스프링을 당기는 구조로 설계 하였는데 스프링을 당기면 벽을 긁는 문제가 발생하여 낚시줄을 캡 가운데에 뚫린 원형 구멍을 관통하는 구조로 설계 하였다. 이러한 구조로 캡을 만들면 스프링을 당기지 않아도 점자 표현을 가능하게 한다. 서보모터와 연결된 낚시줄이 풀리는 문제를 해결하기 위하여 기존과는 다른 방식의 매듭법을 활용해 문제를 해결하였다.

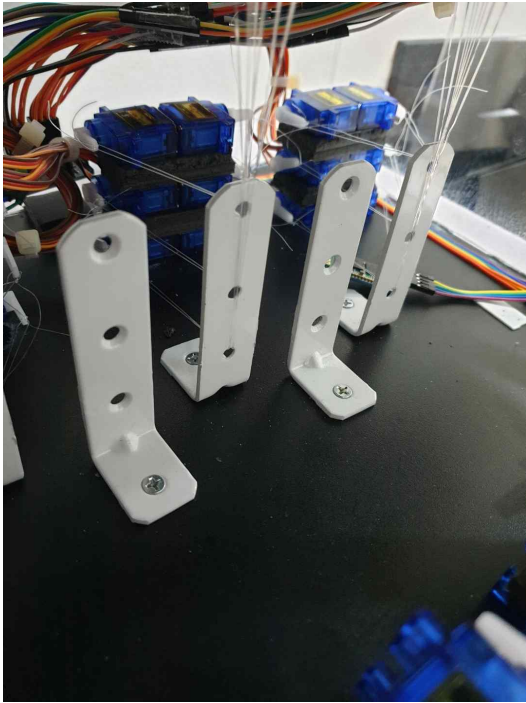


그림 32 ㄱ자 경첩 지지대

그림 32은 낚시줄을 엉뚱한 방향으로 당기는 문제점을 해결하기 위해 점자 캡 바로 아래에 3개의 구멍을 가진 지지대를 배치해 수직으로 내려 어색한 모양을 방지하였다.

위의 테스트과정을 거쳐 점자 표현의 명확성을 확보하였다. 최종적으로 표현할 점자는 8개이다. 8개의 점자모델링에 위의 시행착오 과정을 거친 결과를 녹여 점자의 명확성을 확보하였다. 최종적으로 완성된 점자키트는 다음과 같다.



그림 33 점자키트 점자부

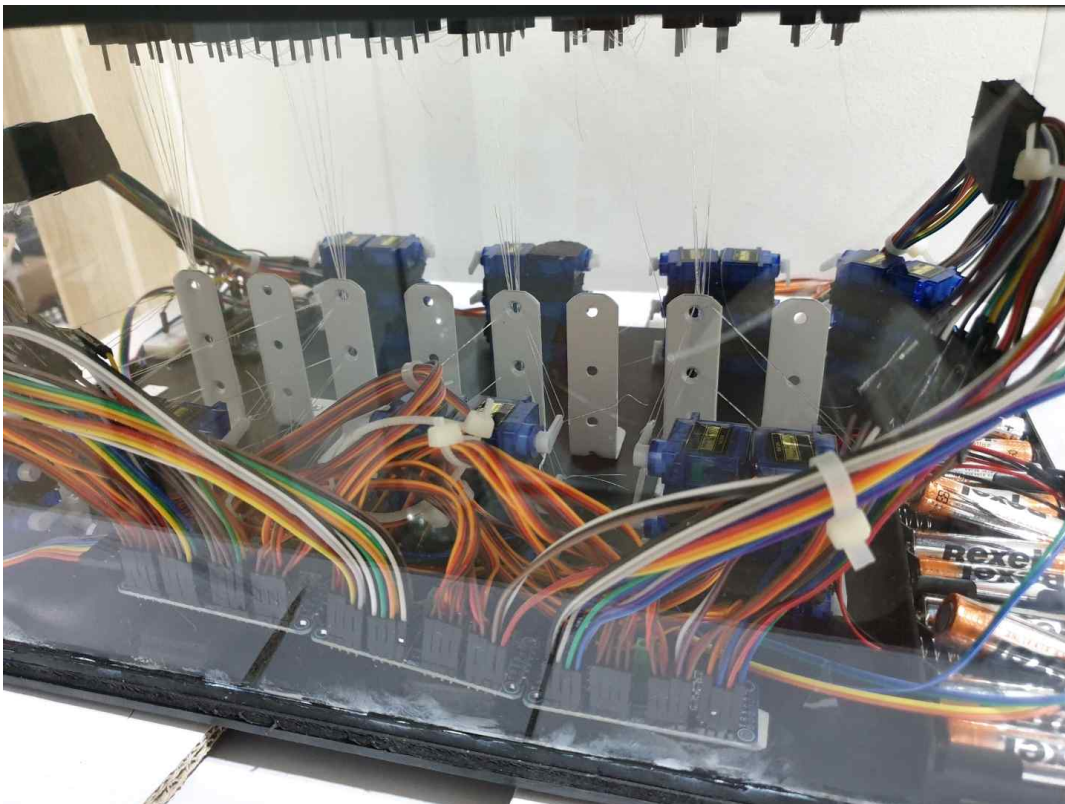


그림 34 점자키트 내부

그림 33,34는 8개의 점자를 표현한 점자 키트의 사진이다. 이 점자 키트의 테스트 중 다양한 문제가 발생하였다. 첫 번째로 서보모터의 전압 분배가 아두이노의 전원만으로 원활하게 공급되지 않는 문제가 있었는데 1.5V 건전지 12개를 이용하여 외부 전압을 공급함으로써 전압 분배 문제를 해결하였다. 두 번째로 48개의 서보모터를 연결하는 PWM 모듈의 주소 지정을 각각 다르게 지정해야 하는데 일반적으로 연결하는 방법으로는 불가능하여 납땜을 이용하여 주소 지정을 새롭게 지정하였다. 이런 문제점들을 해결하여 점자를 직관적으로 표현할 수 있는 점자 키트가 완성되었다.

통합테스트

통합 테스트는 애플리케이션과 점자 키트의 연동을 통한 데이터 입출력이 원활하게 이루어지면 성공적인 테스트가 된다. 애플리케이션의 모든 화면에서 블루투스의 연동을 검사하였고 성공적으로 연동됨을 확인하였다. 각 기능들도 점자 키트와 데이터 입출력을 잘 실시하는지 테스트하였다. 교육 기능, 퀴즈 기능 그리고 번역 기능까지 호환됨을 알 수 있었다.

7. Coding & DEMO

7.1 Class Diagram, FlowChart

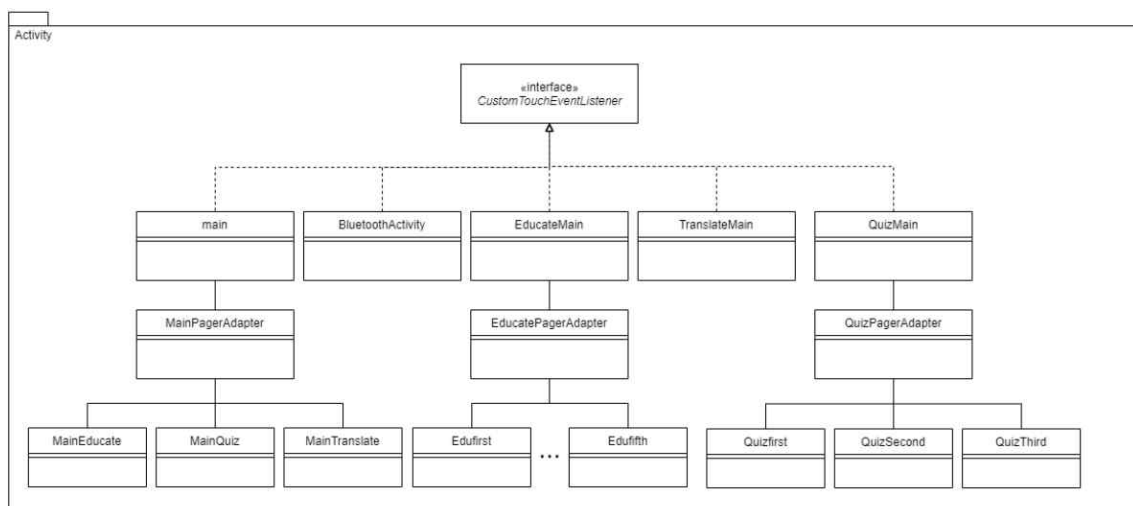


그림 35 전체 액티비티의 클래스 다이어그램

그림 35는 애플리케이션상의 전체적인 액티비티를 나타내는 클래스 다이어그램이다 모든 화면은 CustomTouchListener를 구현하게 된다. 그리고

PagerAdapter Class를 통해서 서브 액티비티를 표현한다.

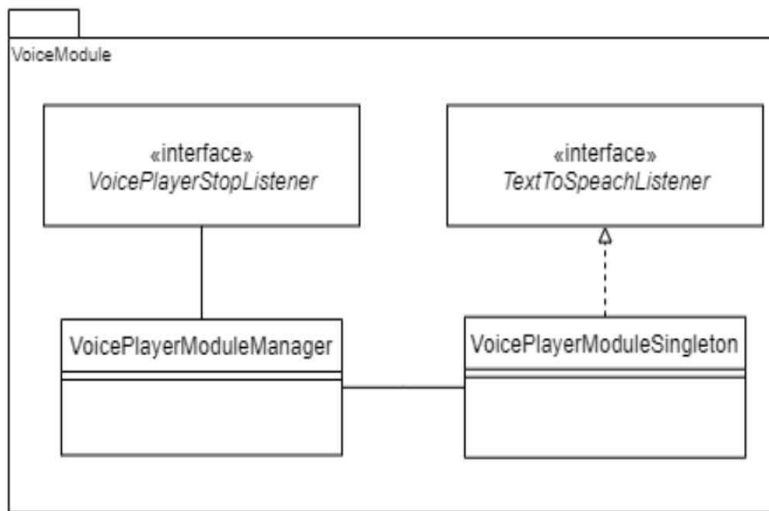


그림 36 음성모듈의 클래스 다이어그램

그림 36은 카카오 음성 API 라이브러리에 있는 TextToSpeechListener를 구현하는 VoicePlayerModuleSingleton이 존재한다. 이 Singleton을 사용하는 VoicePlayerModuleManager가 있다. 그리고 VoicePlayerStopListener를 Manager Class에 연관을 시켜 구현을 하면 음성 취소기능을 수행한다.

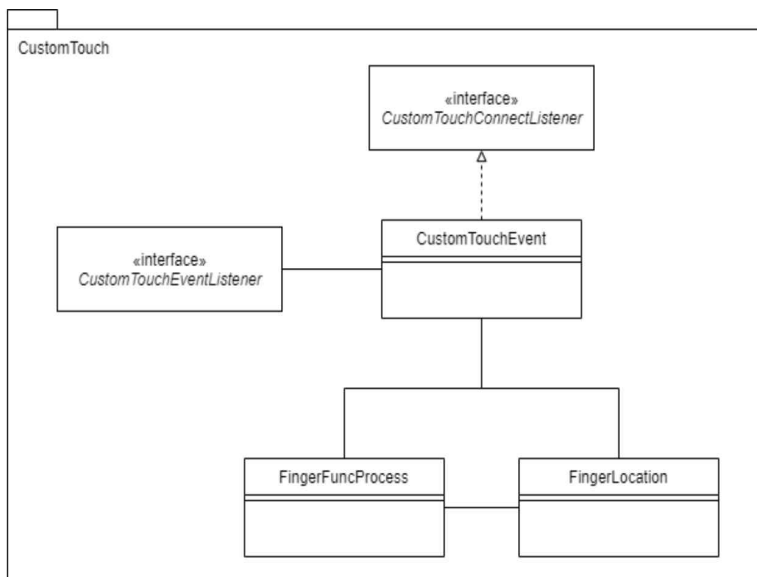


그림 37 커스텀터치의 클래스 다이어그램

그림 37은 기존 터치이벤트 연결부분 기능이 있는 CustomTouchConnectListener 인터페이스를 CustomTouchEvent로 구현을 한다. FingerFuncProcess와 FingerLocation과의 연관관계가 존재하고 해당 클래스는 CustomTouchEvent 클래스 내부에서 사용한다. 그리고 CustomTouchEventListener또한

CostomTouchEvent 클래스 내에서 사용한다. 해당 CostomTouchEventList ener를 통해서 액티비티에서 발생하는 제스처 결과를 보내는 역할을 한다.

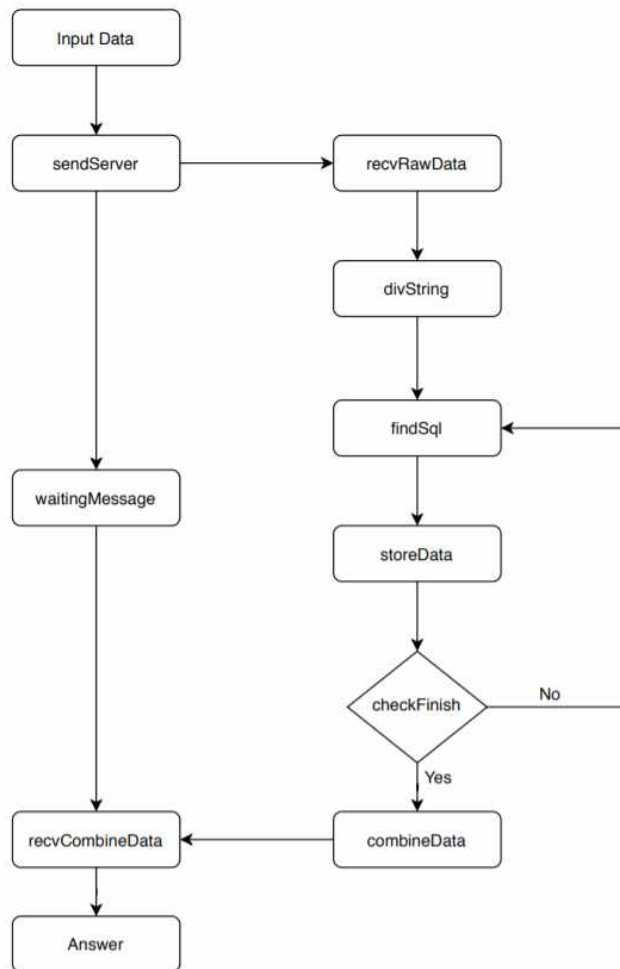


그림 38 어플리케이션 번역기능 flowchart

그림 38은 어플리케이션의 번역기능을 담당하는 흐름도이다. 번역기능에서 점자를 입력받으면 해당 데이터를 서버로 보냄과 동시에 어플리케이션은 서버로부터 데이터를 받을 때 까지 대기한다. 그리고 서버는 어플리케이션으로부터 점자 데이터를 받아 12개의 크기로 나눈다. 나누어진 점자데이터를 데이터베이스에 있는 데이터와 비교를 하고 hashMap을 이용하여 저장한다. 총 점자의 개수를 12로 나누었을 때 나머지가 0일 때 까지 반복횟수를 가진다. 나온 결과를 조합하여 애플리케이션으로 보낸다. 어플리케이션에서 서버로부터 받은 데이터를 화면에 출력한다.

7.2 데모 시나리오

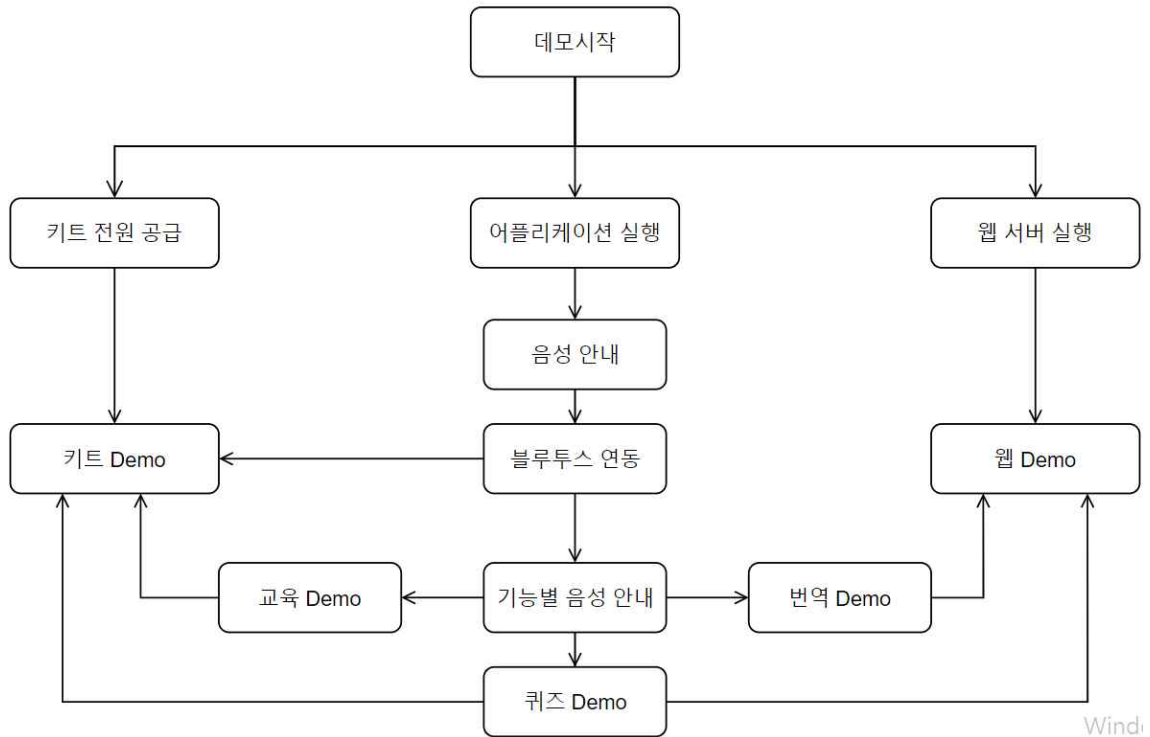


그림 39 데모시나리오 FlowChart

데모시나리오는 그림 39와 같다. 점자키트의 전원을 인가한 뒤 어플리케이션을 실행한다. 어플리케이션 실행 후 어플리케이션의 기본적인 동작환경과 사용법을 알 수 있게 음성으로 안내를 받는다. 각 기능을 실행하기 전 안내음성을 통한 전체적인 설명을 음성으로 확인한다. 블루투스가 연동이 되고 교육기능의 전 부분을 하나하나씩 실행시켜 점자키트와의 연동을 확인한다. 교육기능 연동 중 해당 점자의 음성을 들으며 손의 감각으로 어떤 점자로 표현되는지 확인한다. 퀴즈 기능으로 넘어간 뒤 2문제 가량의 점자퀴즈를 풀고 결과를 확인한다. 그리고 번역메뉴로 들어가서 1~2개의 점자를 입력하여 번역이 올바르게 이루어지는지 확인한다. 웹페이지에 퀴즈의 결과로 인해 생성된 웹페이지의 변화를 확인한다.

데모 단계별 실행화면



그림 40 Splash 화면



그림 41 어플 안내 화면

그림 40은 Splash화면이다. 로고를 몇초간 띄우고 어플리케이션 설명을 듣는 음성안내 화면으로 넘어간다.

그림 41은 어플리케이션의 전체적인 동작에 관한 가이드를 듣는 화면이다. 제스처를 사용하여 메뉴이동 및 기능수행을 할 수 있는 전반적인 내용을 안내 받는다.



그림 42 블루투스 연동 화면

그림 42는 블루투스 연동을 위한 음성안내 화면이다. 점자키트와 블루투스 연동을 위한 안내음성을 들을 수 있다

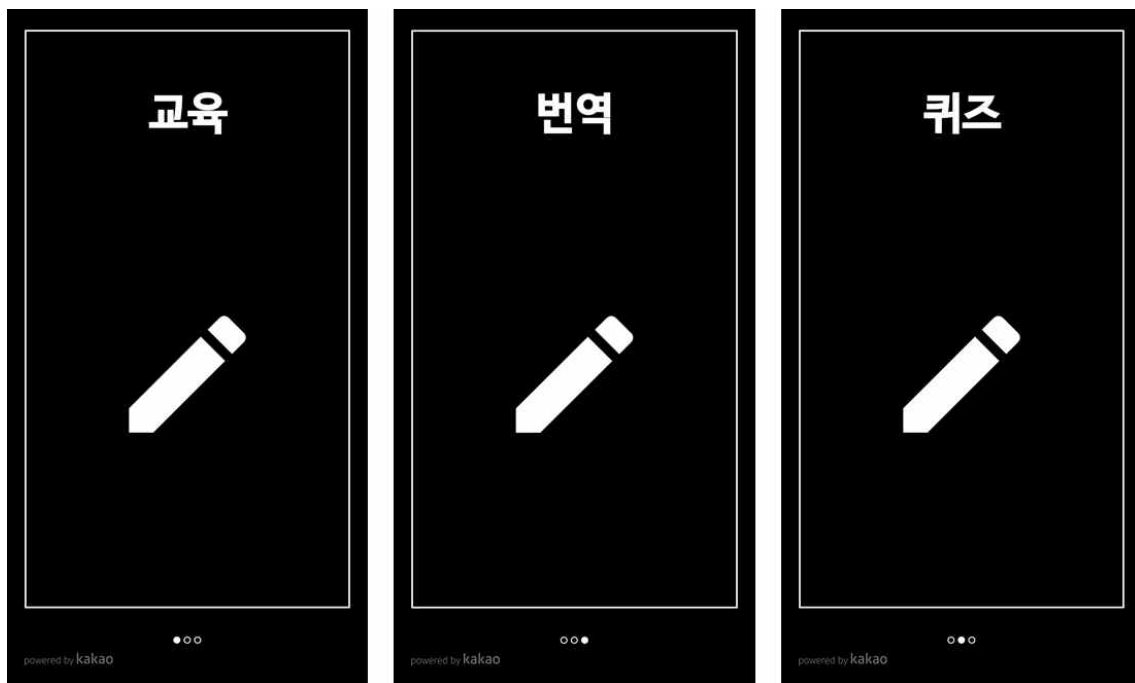


그림 43 메뉴 항목

그림 43은 메뉴를 확인할 수 있는 화면이다. 메뉴이동시 해당 메뉴 이름을 음성으로 안내를 받는다.



그림 44 교육 안내화면
그림 44는 교육메뉴 제스처 사용 안내음성을 알려주는 화면이다.

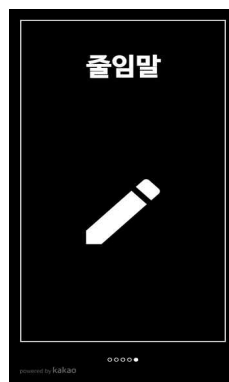
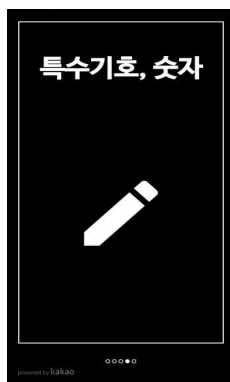
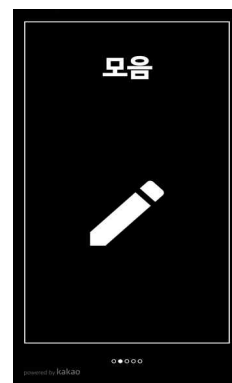
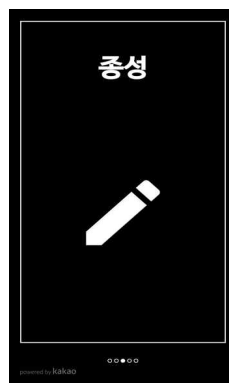
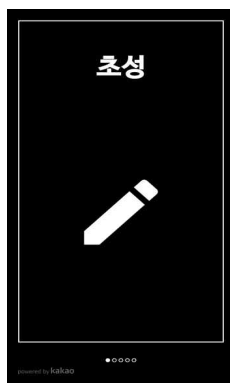


그림 45 메뉴 항목이다
그림 45는 교육메뉴의 서브메뉴 목록이다.

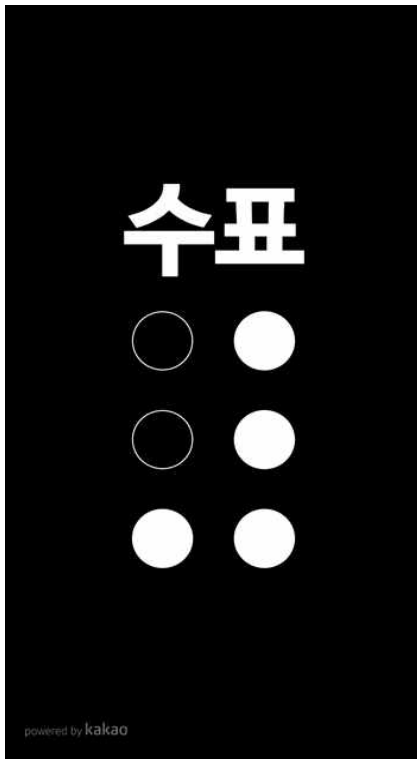


그림 46 특수기호, 숫자 화면

그림 46은 그림 45 메뉴의 특수기호 및 숫자 결과 화면이다.

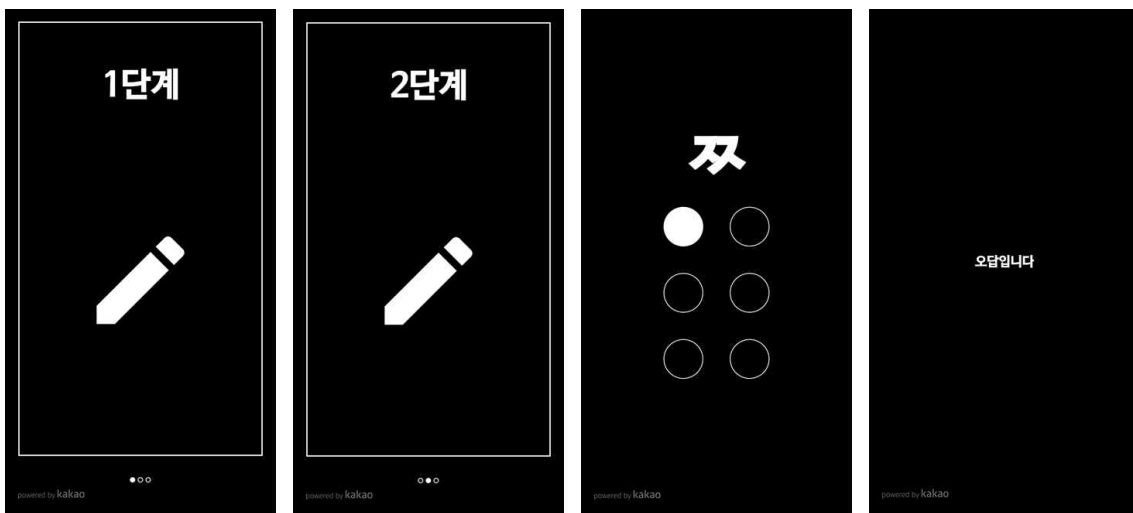


그림 47 퀴즈 메뉴의 점자입력 및 결과화면

그림 47은 퀴즈메뉴의 점자입력부와 결과화면이다. 제스처로 점자를 입력하면 퀴즈의 답을 확인하는 결과화면으로 전환된다.

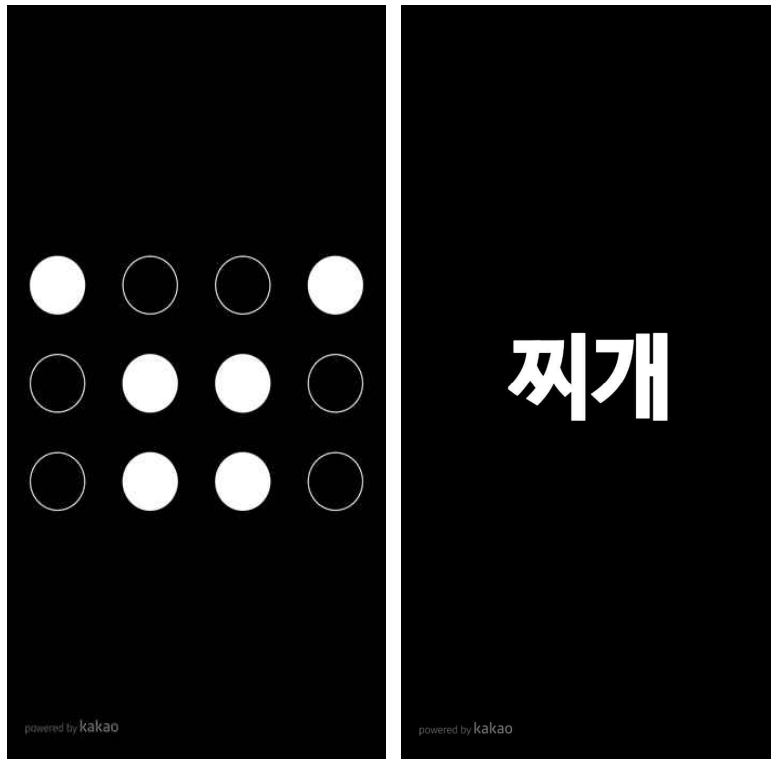


그림 48 번역메뉴의 점자입력 및 결과 화면

그림 48은 번역메뉴의 점자입력부와 결과화면이다. 제스처로 점자를 입력해서 해당점자의 내용을 확인하는 결과화면과 전환된다.

많이들린 점자랭킹

1 우거지	11 취소
2 여우	12 관람
3 용	13 새우
4 야유	14 책값
5 어머니	15 샷
6 찌개	16 낚시
7 바나나	17 버섯
8 아버지	18 꽃
9 나들이	19 법률
10 권리	20 국어

그림 49 많이 틀린 점자랭킹 입출력 전 화면

많이들린 점자랭킹

1 우거지	11 취소
2 여우	12 관람
3 용	13 새우
4 야유	14 책값
5 찌개	15 샷
6 어머니	16 낚시
7 바나나	17 버섯
8 아버지	18 꽃
9 나들이	19 법률
10 권리	20 국어

그림 50 많이 틀린 점자랭킹 입출력 후 화면

그림 49와 50은 많이 틀린 점자랭킹을 웹페이지에 나타낸 화면이다. 현재 5위 찌개 6위 어머니를 확인할 수 있다. “찌개”라는 단어가 틀려서 순위가 역전됨을 확인할 수 있다.

Ⅲ. 결론

1. 연구 결과

1.1 실험 및 결과

본 프로젝트에서는 시각장애인이 OnDot을 사용하는 것을 고려하여 사용성을 높이기 위해 다음과 같은 실험을 진행하였다. 첫 째, 터치 제스처만으로 동작하는 어플리케이션의 명확한 사용을 위해 터치의 인식범위를 실험하였다. 둘 째, 겹받침, 된소리, 어려운 단어 등의 점자들을 정확하게 번역할 수 있는지 확인하는 과정을 테스트하였다. 셋 째, 동작 모터의 개수에 따라 문제없이 동작하는지 실험하였다.

터치 제스처로 모든 기능을 실행할 수 있음에 따라 메뉴 이동과 같은 기본적인 제스처를 사용하여 정상적인 터치 입력과 잘못된 터치 입력을 구분하는 실험을 다음과 같이 진행하였다. 실제 사용하는 상황을 가정하여 한 손가락은 엄지손가락을 이용한 경우와 검지손가락을 이용한 경우로 나눠서 실험한다. 두 손가락은 검지와 중지만 이용하여 실험한다. 총 20회의 터치를 보편적으로 스마트폰 사용자가 슬라이드를 하는 방식으로 4명이 각각 5회씩 진행하였다. 실험결과, 손가락을 올려둔 시작점의 X, Y축의 좌표와 손가락을 떼는 X, Y축의 좌표의 차가 스마트폰 화면 비율의 30% 이하일 경우 실패하는 경우가 생겼고, 30% 이상일 경우 의도하고자 한 터치 제스처의 입력이 정상적으로 실행됨을 알 수 있었다. 표 14는 한 손가락과 두 손가락의 터치 제스처의 판정범위를 실험하여 올바른 터치 입력이 적용되는지 나타낸 표이다.

표 14. 손가락의 제스처 성공 횟수

X, Y 축 비율차이	성공회수
5%	8
10%	9
15%	12
20%	17
25%	18
30%	20
35%	20
40%	20

번역 알고리즘에서 언급한 겹받침, 된소리, 어려운 단어 등의 점자들을 정확하게 번역할 수 있는지 확인하는 과정을 테스트하였다. 표 15은 점자 번역에 대한 결과를 정리한 표이다.

표 15. 점자 번역 결과

번역 단어	입력된 점자 문자열	번역 결과
야유	112221211212	야유
법률	111221122211221111111212112121111	법률
스웨터	111112121212222211222121221121122211	스웨터
옹달샘	22222121211121111111112222121121112	옹달샘
힘껏	121221212121121112111112111222122211	힘껏

표 16은 모터의 개수에 따른 점자키트의 부하테스트 표이다. SMPS로 전류를 공급하는 중 부하전류(Stall Current)를 받는 서보모터의 개수를 점점 늘려가며 테스트를 하였다. 결과적으로 현재 장착중인 SMPS로 모든 점자표현이 가능함을 알 수 있다.

표 16. 점자키트 동작 성공 횟수

동작 모터 개수	성공회수
6	10
12	10
18	10
24	10
30	10
36	10
42	10
48	10

1.2 결 론

시각장애인은 사회적 약자로서 점자라는 특수한 문자 체계를 이용하여 글을 읽을 수 있다. 이에 다양한 점자 학습의 방법이 있지만 스스로 학습하는 것에 어려움이 존재해 조력자가 필요하고, 현재 출시된 점자학습 어플리케이션은 점자를 만져보며 할 수 없다는 단점이 존재한다.

이에 본 프로젝트에서는 시각장애인도 혼자서 스마트폰과 점자키트를 이용하여 점자의 기초가 되는 자음, 모음부터 약어까지 학습할 수 있고 이를 퀴즈로 풀어 효율적으로 점자를 학습할 수 있게 하였다.

실험결과에서도 알 수 있듯이 제스처 드래그의 비율을 높여 터치 인식률을 높였고, 번역 알고리즘을 통해 의도한대로 번역이 되는 것을 확인했다. 또한, 점자키트에 전류가 적절하게 공급되는지 확인하기 위해 부하전류를 받는 서보모터의 개수를 늘려가며 점자표현이 정확하게 되는 것을 확인하여 점자 학습에 어려움이 없다는 것을 알 수 있다.

OnDot의 점자키트는 많은 수의 서보모터를 제어하기 때문에 점자키트의 크기가 필연적으로 커질 수밖에 없다. 또한 220V 교류 전원의 필요로 휴대성을 확보하지 못하였다. 하지만 점자키트를 소형화 하고 휴대성을 확보하면 어디서든지 점자교육을 받을 수 있을 것이라 판단된다.

2. 작품제작 소요재료 목록

표 17은 소요재료의 목록을 정리한 표이다.

표 17 점자 디바이스 소요재료 목록

점자 디바이스 소요재료 목록	
재료명	가격
아두이노 Uno	이전 프로젝트로 보유중
서보모터 sg90 x 48개	57240원
PWM 서보모터 드라이버 x 3개	14250원
디바이스 바닥제 (포맥스)	12000원
모터 높이 보충제 (폼보드)	5300원
점자 표현 스프링 x 50개	13000원
점자 표현 ㄱ자꺾쇠 x 16개	12740원
전원공급 건전지 x 12개	3600원
건전지 홀더 x 4개	7200원
아크릴 칼 (바닥제 절단용)	2000원
납시줄	2000원
옆면 투명 아크릴판	30000원
예시용 점자표현 3d 모델링	300000만원

점자표현 3d 모델링	600000만원
-------------	----------

참고문헌

- [1] 보건복지부, “2017년 장애인 실태조사”, 정책보고서 2017-90, 발간등록 11-1352000-000568-12
- [2] 탭틸로, “탭틸로 사용설명서”, <https://www.taptilo.com/kor>, (2019/11/10)
- [3] 봄 - 시각장애인을 위한 점자학습, https://github.com/ch-Yoon/refactoring_braille , ch-Yoon
- [4] 뉴스토마토, “VC 진동점자학습”, <http://www.newstomato.com/ReadNews.aspx?no=906184>, (2019/11/18)
- [5] 한글 - 점자 번역 파이썬 프로그램 , <https://github.com/hyonzin/hangul-braille-converter> , hyonzin
- [6] 한국시각장애인 연합회 , “시각장애인의 이해”, <http://www.kbuwel.or.kr/Blind/Braille> , (2019/11/15)
- [7] JEREMY BLUM, 익스플로링 아두이노, 한빛아카데미, 2014/05/26,
- [8] 권영은, 『NX 3D모델링 실기 활용서』, 메카피아, 2018
- [9] Siemens Nx Tutorials, "Unigraphics Siemens UG Nx 10 Basic Modeling Tutorial for Beginner - 1", YouTube, 2017.04.19.
- [10] 정재곤, 『Do it! 안드로이드 앱 프로그래밍』, 이지스퍼블리싱, 2019
- [11] 송미영, 『쉽게 배우는 JSP 웹 프로그래밍』, 한빛 아카데미, 2018
- [12] AWS - Amazon RDS 기능 <https://aws.amazon.com/ko/rds/features/> , (2020/01/24)
- [13] JoShi - JSP 게시판 만들기 - 네이밍, 데이터베이스 설계 http://www.joshi.co.kr/index.php?mid=board_wUIV71&document_srl=301683 , (2020/01/28)
- [14] 안드로이드 Developers API 멀티터치 동작 처리 , GoogleDeveloper

<https://developer.android.com/training/gestures/multi?hl=ko>
(2020/02/01)

[15] 카카오 음성인식 API SDK 기술문서 , KAKAO

<https://developers.kakao.com/docs/android/speech#%EC%9D%8C%EC%84%B1-%EC%9D%B8%EC%8B%9D> , (2020/02/02)

[16] 카카오 음성합성 API SDK 기술문서 , KAKAO

<https://developers.kakao.com/docs/android/speech#%EC%9D%8C%EC%84%B1-%ED%95%A9%EC%84%B1> , (2020/02/02)