

제19회 임베디드SW경진대회 개발계획서

[webOS 기반 Car 2 Smart Home 솔루션]

□ 개발 개요

○ 요약 설명

팀 명	삼아아
작품명	SMILE : Smart Mobility Improving Life Experience
작품설명 (3줄 요약)	<ul style="list-style-type: none">- 차량 내에서 얼굴인식으로 사용자를 인식하여 대시보드 혹은 음성을 통해 스마트 홈을 제어할 수 있는 시스템을 구축한다.- 사용자는 차량의 대시보드를 통해 집 안 상황과 가전의 동작 현황을 확인할 수 있다. 이 때, 가스밸브, 보일러 등 사용자 외출 시 주의가 필요한 사항이 켜져 있는 경우에는 음성과 알람으로 사용자에게 안내한다.- 사용자는 집 혹은 차량의 대시보드로 가전의 동작스케줄을 정의할 수 있다. 스케줄은 일시적으로 혹은 반복적으로 정의가 가능하며, 날씨나 온습도를 확인하여 특정 조건에서 가전이 동작을 수행하도록 조건을 만들 수도 있다.

○ 개발 목적 및 목표

- 차량에서 대시보드로 집의 상황을 확인하고 음성으로 간편하게 집의 가전을 제어할 수 있는 시스템을 구축하여 편리한 스마트 홈 서비스를 제공한다.
- 사용자가 외출해 있을 때 유용한 기능을 제공하여 안심하고 외부일을 할 수 있도록 돕고, 스케줄링 시스템을 통해 효율적으로 가전 제어를 할 수 있도록 한다.

○ 개발 배경 및 동기

- 최근 증가하고 있는 1인 가구의 특성상 외출 시 집이 비게 되어서, 외부에서 집의 상황과 가전의 동작여부를 확인할 수 있는 시스템을 제공하면 자취하는 사람에게 큰 도움이 될 것이다.
- 생활가전의 보급률이 늘어나면서 종류와 기능들도 더 다양해지고 있다. 하지만 항상 알맞은 시간이나 상황에 가전을 가동시키는 것은 까다로운 일이다. 그래서 스마트 홈의 스케줄링 시스템을 통하여 원하는 시간이나 조건에 가전을 동작 시킨다면 편리하고 효율적인 기능이 될 것이라 생각하여 기획하게 되었다.

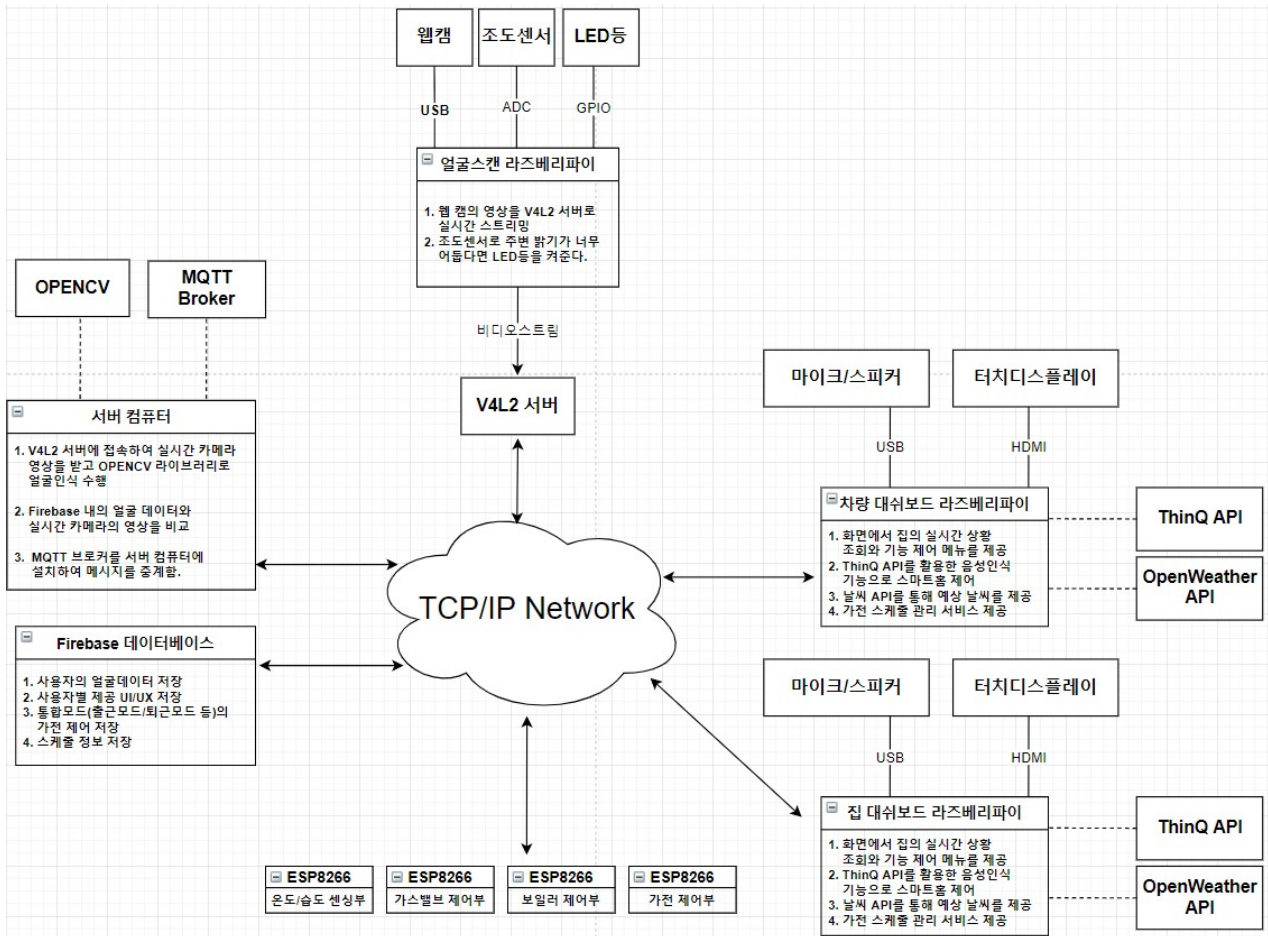
○ 작품 상세 설명

- 차량 내에서 얼굴인식으로 사용자를 인식하여 대시보드 혹은 음성을 통해 스마트 홈을 제어할 수 있는 시스템을 구축한다.
- 사용자는 차량의 대시보드를 통해 온습도와 공기 질 등의 집 안 상황과 가스밸브 및 보일러, 가전의 동작 현황을 확인할 수 있다.
- 사용자는 집 혹은 차량의 대시보드로 가전의 동작스케줄을 정의할 수 있다.

□ 개발 방향 및 전략

○ 개발 방향

- 예상 구성도



- 각 구성부 상세설명

1. 얼굴스캔 라즈베리파이

- 사용자 인증을 위한 얼굴 인식 촬영 및 스트리밍을 담당하는 부분이다. 라즈베리파이와 연결된 웹캠을 통해 실시간 영상을 V4L2 서버로 실시간 스트리밍한다.
- 사용자의 얼굴 등록은 차량 대쉬보드로부터 얼굴 등록 명령을 받아 수행한다. 웹캠으로부터 영상을 촬영하며, 얼굴의 정면, 왼쪽, 오른쪽이 잘 찍히도록 해야한다. (대쉬보드 화면에 영상과 함께 문구를 띄워주어서 등록이 쉽게 되도록 돕는다.) 이 때, 서버 컴퓨터는 해당 영상에서 사진을 캡처하고 Firebase에 저장하여 학습할 수 있도록 한다.
- 조도센서를 통해 아날로그 값을 ADC 포트로 전달받아 주변의 조도를 확인한다. 만약 얼굴을 인식하기 힘들 정도로 어둡다면 LED등을 켜주어 얼굴 인식을 향상해 도움을 줄 수 있도록 한다.

2. 서버컴퓨터

- 서버컴퓨터는 웹캠에서 스캔되는 얼굴이 사용자인지 판단하고, MQTT의 브로커 역할을 맡는다.
- 얼굴인식 활용 기술은 OpenCV와 TensorFlow의 라이브러리를 활용하여 구현한다.
- MQTT의 브로커는 Publisher와 Subscriber 사이에서 메시지의 전송을 중계하는 역할을 한다. 영상 전송을 제외한 나머지 부분의 데이터 송수신은 MQTT 프로토콜을 사용하며,

센싱 데이터, 가전 동작 제어 등의 데이터가 송수신의 예가 될 수 있다.

3. Firebase 데이터베이스

- Firebase는 구글에서 제공하는 실시간 데이터베이스 서비스로 다수의 클라이언트가 데이터를 저장하고 동기화 할 수 있는 API를 제공한다.
- 여러 기기들이 실시간으로 스케줄, 센싱 데이터, 가전 동작 여부를 편리하게 조회할 수 있어 채택하게 되었다.
- 사용자의 얼굴데이터와 사용자별 UI/UX, 통합모드의 동작 기능, 스케줄 정보, 센싱데이터와 가전 동작 여부를 Firebase에 저장하여 사용할 계획이다.

4. 집 / 차량 대쉬보드 라즈베리파이

- 대쉬보드 라즈베리파이는 터치 화면과 음성을 통해 집 내부의 상황과 가전 스케줄을 확인하고 제어할 수 있다.
- 터치 디스플레이를 통해 UI/UX 화면이 제공되며, 이 때 제공되는 UI/UX는 사용자에게 따라 설정이 가능하다.
- 가전을 개별로 제어하는 기능, 가전 스케줄 정의 및 실행하는 기능, 출근/퇴근/여행 등의 통합메뉴를 제공하는 메뉴가 있다.
- 대쉬보드 라즈베리파이는 특정시간마다 Firebase에 저장된 센싱 데이터 및 가전 ON/OFF 여부를 수집하고 스케줄을 확인하여 시간에 맞게 명령을 내릴 수 있도록 한다.
- 만약, 가스밸브, 도어락, 보일러 등이 켜진 상태로 외출 시 대시보드에 해당 내용을 알람과 음성으로 출력하여 사용자에게 알리고 제어할 수 있도록 한다.

5. ESP8266 + Arduino R3 / ESP32(집 내부 환경 센싱 및 가전 제어)

- 집 안 환경을 센싱하고 가전을 제어하는 역할로 실제 집 환경에서 유선으로 제어되는 것 보다는 와이파이를 통해 간편하게 설치되는 것이 좋아보여 ESP8266(orESP32)가 맡은 역할을 수행하도록 하였다.
- 집의 온/습도 센싱, 가전 동작, 가스밸브/보일러 동작 여부 확인을 ESP8266을 통하여 제어할 예정이다.
- ESP8266은 MQTT로부터 토픽을 받으면 해당 센싱데이터나 가전 동작 수행 성공 여부를 메시지로 전달한다.

○ SW 개발 방법 및 활용 기술

- 얼굴 학습은 OPENCV를 활용하여 수행한다. 웹캠의 영상으로부터 40~50장의 얼굴 이미지를 Firebase에 저장하고 이를 학습하여 사용자의 FaceNet을 생성한다.
- 얼굴 인식은 서버컴퓨터에서 OpenCV와 구글에서 제공하는 TensorFlow를 통해 수행한다. 이 때, 얼굴이 비슷한 사용자가 있을 수 있으므로 한 프레임마다 사용자임이 확인되었을 때 count 변수를 증가시키고 사용자와 다를 때 count 변수를 감소시킨다. 그래서 인증시간이 지났을 때, count가 특정 값 이상이라면 사용자임을 확인하고 인증을 완료한다.
- Firebase는 구글에서 제공하는 Cloud 시스템과 비슷한 데이터베이스로 다양한 기기들이 인터넷을 통해 접근할 수 있는 데이터베이스 시스템이다. 실시간으로 데이터가 갱신될 수 있기 때문에 다양한 센싱 데이터와 가전 동작 상황을 즉각적으로 보고하는 스마트홈

시스템에 적합하다고 판단하여 채택하게 되었다.

- MQTT 프로토콜은 인터넷을 통해 센서와 기기를 연결하고 이들 사이에서 메시지를 주고받는 기법으로 현재 사물인터넷 시장에서 널리 사용되고 있다. MQTT에서 주고 받는 데이터를 메시지라고 하며, 메시지를 기다리는 Subscriber와 메시지를 발송하는 Publisher, 그리고 중간에서 중개하는 Broker로 역할이 나뉜다. Subscriber는 여러 메시지 중에 받고자 하는 메시지의 토픽을 등록하면 Publisher가 메시지를 토픽과 함께 브로커에 전송하여 등록된 가입자 중 토픽을 기다리는 구독자에게만 메시지를 전송한다. 즉, 토픽을 기반으로 구독자와 발행자 사이에서 메시지가 전송된다고 할 수 있다.
- 영상 데이터를 제외한 나머지 기기간의 통신은 모두 MQTT 프로토콜을 통해 텍스트 메시지를 주고받는다. 이 때, MQTT 브로커는 서버컴퓨터에 설치되어 사용된다. 예를 들어 Subscriber(집 / 차량 대시보드)가 토픽(temperature)를 등록하면 Publisher(ESP8266/온습도)가 토픽과 함께 온도 정보가 담긴 메시지를 브로커에 전송하여 Subscriber에 전달한다.
- OPEN Weather API 에서 제공하는 데이터는 JSON의 형태로 제공된다. 우리는 이것을 JavaScript 형태로 변환하여 활용할 계획이다.

○ HW 개발 방법 및 활용 기술 (후원 기업 제공 장비 및 팀별 사용 장비)

- RaspberryPi 4 3EA(얼굴 스캔, 차량 대시보드, 집 대시보드)
- HDMI 지원 및 터치 지원 디스플레이 2EA (차량 대시보드, 집 대시보드)
- ESP32 or ESP8266 + Arduino R3(집의 온/습도 체크 및 GPIO로 가전 제어)
- 릴레이 (가전제어)
- 마이크와 스피커 (음성인식 및 출력)
- 웹캠 (얼굴 인식을 위한 영상 촬영)
- 조도센서 (차량 밝기 측정), 온도센서/습도센서 (집 내부 환경 측정)

○ 예상되는 장애요인 및 해결방안

1. 음성인식

장애요인	음성인식의 실패로 사용자 명령을 전달할 수 없음
해결방안	1) 핸들에 음성인식 버튼을 두어 다시 입력하게 한다. 2) 대시보드에 동작 버튼을 두어 원하는 동작을 터치로 진행한다.

2. 얼굴인식

장애요인	주변환경으로 인하여 운전자의 얼굴을 제대로 인식할 수 없음
해결방안	1) 얼굴인식시 영상을 차량 대시보드 화면에 띄워주어 영상이 잘 촬영되고 있는지 디스플레이 해준다. 2) 조도 센서를 설치하여, 주변 환경이 어두워지면 등을 켜주어 얼굴인식을 항상에도움을 주도록 한다.

3. 통신끊김

장애요인	일시적인 네트워크 품질 불량으로 서버와의 연결이 끊김
해결방안	1) 명령 실행전 서버와의 연결상태를 확인하고, 연결이 끊어져있는 경우 서버에 연결된 뒤 후에 해당명령을 처리한다.

4. 센싱 데이터

장애요인	전원 불안정 등의 원인으로 인하여 조도센서, 온도센서, 습도센서의 ADC 값이 불안정 할 수 있음.
해결방안	1) ADC 값을 즉시 사용하는 것이 아니라 특정 시간동안 값을 평균내어 해당 값을 사용할 수 있도록 한다.

○ 예상 결과 작품이 활용될 분야 및 방법 제시

1. 최근 증가하고 있는 1 인 가구는 거주자가 출근하거나 출장, 여행 등을 떠나게 되면 집이 오랫동안 비게 되는데 이러한 경우에 출근 모드, 출장 모드 등의 통합모드를 활성화하여 필요 없는 가전을 끄고 필수적인 가전만 가동하여 에너지 절약을 할 수 있다. 또한, 1 인 가구의 경우 가스나 보일러 등이 켜졌는지 알기 어려운 경우가 많은데 이 정보를 차에서 확인하고 제어할 수 있도록 하면 안심하고 다른 일을 하는데 큰 도움이 될 것으로 예상된다.
2. 요즘 공기청정기, 제습기, 로봇청소기 등의 가전들이 많이 사용되고 있다. 이 가전들을 일일이 제어하는 것은 까다롭다. 그래서 예상되는 날씨와 집의 환경을 확인하여 사용자가 설정한 조건과 일치하면 가전이 스스로 동작하는 스케줄링 시스템을 구축한다 예를 들어, OPEN Weather API 를 통해 비가 온다는 예보를 확인하였다면, 해당 시각에 실제로 비가 오는지 확인하고 습도 센서로 집의 습도가 80% 이상이 되면 스스로 제습기로 집의 습도를 조절하도록 한다.

□ webOS 활용 방안 및 기술 공부 내용

○ webOS 활용 방안

- 차량제어용 대쉬보드 개발

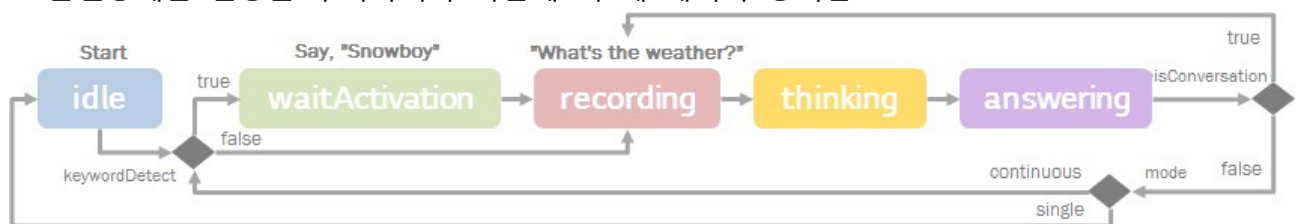
1. 기존 사용하던 html/css/js와 google material Design을 이용하여 UI/UX 개발.
2. Enact를 사용하여 사용자에게 행동에 따라 변화하는 Web-App 개발
3. 스마트홈 내에서 발생하는 이벤트를 실시간으로 사용자에게 표시
4. 스마트홈의 가전가동여부, 문단속 여부 등의 현상태를 표시해줌.

- 카메라를 연동하여 얼굴인식

1. LS2 API 內 com.webos.service.camera2와 카메라를 연동
2. 처리 서버에 촬영된 사용자의 얼굴을 전송하고, DB에 저장되어있는 각 사용자의 얼굴을 비교하여 각 사용자별 고유권한을 부여한다.
3. 사용자 인식 시작 시점은 다음과 같다. (1)사용자가 자리에 앉았을 때, (2)터치스크린의 로그인버튼을 눌렀을 때, (3)스마트홈을 제어하려고 할 때
4. 인식된 사용자에게 따라 각기다른 대쉬보드의 UI/UX를 구성한다.

- ThinQ API의 음성인식을 이용하여 스마트홈 제어

1. google cloud 계정사용 (장치등록 필요)
2. LS2 API 內 com.webos.service.ai.voice와 마이크를 이용해 사용자로부터 음성으로 명령을 입력받음
3. 입력된 음성명령을 바탕으로 DB서버와 통신하여 스마트홈 내의 기기들을 확인,제어함
4. 스마트홈 제어시 최종적으로 사용자의 얼굴인식을 진행해 권한을 인증받아야함. (예시: 대리기사의 조작 등을 미연에 방지)
5. 운전중에는 전방을 주시하여야 하는데 이 때 제어가 용이함.



- 이벤트 발생시 사용자에게 정보전달 (대쉬보드 및 TTS 이용)

1. google cloud 계정사용 (장치등록 필요)
2. LS2 API 內 com.webos.service.tts를 이용해 사용자에게 음성으로 정보를 전달함.
3. 어플리케이션은 스마트홈과 서버를 주기적으로 모니터링하여 이벤트를 캐치하고, 캐치된 이벤트 내용은 TTS 기능을 통해 사용자에게 음성으로 안내됨. (예시 : 가스밸브가 열렸을 때 등)
4. 음성인식 기능과 연계하여, TTS음성과 사용자가 질의식으로 대화하도록 유도해 이벤트를 처리함. (예시 : 열린 창문 닫기, 켜진 TV 끄기, 폭염경보가 내렸을때 에어컨 켜기 등)

○ webOS 기술 공부 내용

- webOS의 특징

1. LG가 지속적으로 관리하며 지원해준다.
2. LG가전 관련 API를 제공하므로, 손쉽게 연동제어가 가능하다. (스마트기기에 적합)
3. Enact / JS 프레임워크와 표준 웹 앱 서비스 등을 지원하여 빠르게 개발이 가능하다.
4. webOSE는 오픈소스로 개발자 편의와 용도에 맞게 수정이 가능하다.

- webOS의 구성

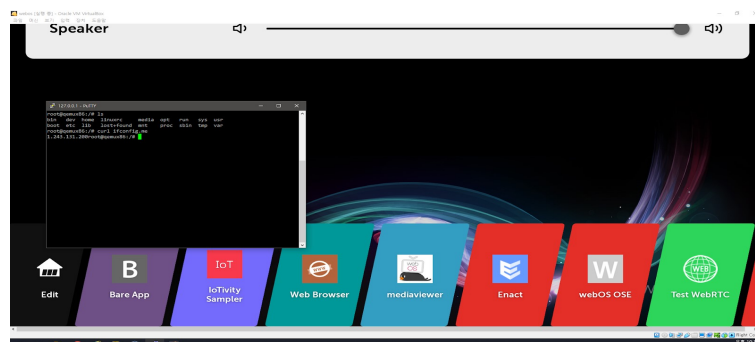
1. 하드웨어(라즈베리파이), webOS, Customize Web App 으로 구성된다.
2. webOS이미지는 다음 4가지로 구성되는데 그 종류는 WebApp, Native Service, Native App, JS Service 이다.
3. Customize Web App의 경우 WebApp과 JS Service로 구성되어있다.
4. 각각의 webOS 내의 요소들간의 통신은 'Luna Bus' 라 불리는 IPC 메커니즘을 통해 수행되며, LS2 API의 LS2가 바로 이 LunaBus 통신을 이야기 한다.

- webOS 빌드 관련

1. 이미지 빌드를 진행하는 컴퓨터의 권장사양은 CPU i7 quad-core / RAM 16 GB 이상이다. 또한 하드디스크 여유공간이 100GB이상으로 하드웨어적 성능이 어느정도 필요하다. 또한 일반적으로 쓰이지 않는 Ubuntu OS에서만 이미지 빌드가 가능하여 별도의 OS설치가 필요하다.
2. webOS 2.0버전 이상은 라즈베리파이4(공식지원)와 가상머신에서만 구동이 가능하다.
3. webOS 내에서 사용할 package, app은 빌드하여 전달해야 한다.
4. webOS는 IPK파일을 opkg명령어를 통해 지원한다. 별도의 미러서버를 지원하지 않으므로 설치파일을 호스트 컴퓨터를 통하여 직접 전달받아야 한다.

- webOS 빌드 후 느낀점

1. 직접 빌드를 진행해본 결과 4시간 이상이라는 긴 빌드타임이 소요됨.
2. root계정으로 빌드를 진행할 수 없으며, 더불어 build-webos 디렉토리의 권한을 현재 사용자에게 부여해야하는 등 기본적인 리눅스 지식/경험이 필요함
3. 안정적인 프로그램 구동을 위해서는 build-time에, 사용할 프로그램을 include 해야한다. 이는 webOS 개발전에 계획을 충분히 가시화 한뒤에 빌드부터 시작해야함을 의미한다.



- webOS 개발시 확인사항

1. 한번에 하나의 App 만 화면에 표시된다. (스마트폰과 동일)
2. Web-App 은 SAM/WAM 을 통해 RunTime 동안 관리된다.
3. 디버깅을 위해서는 gdb 패키지를 별도로 다운받아 적용시켜야 한다.
4. Beanviser 를 활용해 webOS 시스템을 모니터링하여 최적화한다.
5. 네이티브앱(C++) 개발시에는 NDK 를 추가로 구성해줘야한다.
6. 호스트 컴퓨터에서 개발된 웹앱은 .ipk 파일 형태로 webOS 에 배포된다.
7. 개발되는 App 은 코드, 메타데이터, 빌드스크립트, 레시피 파일로 구성 되어있다.

○ LS2 API 활용방안webOS

- 시스템 관리

com.webos.service.downloadmanager

활용 : 파일 송수신.

설명 : 장치에서 파일 업로드 및 다운로드를 관리

com.webos.service.intent

활용 : 어플간 데이터 교환시에 사용.

설명 : 인텐트를 처리하고 핸들러를 관리

- 응용프로그램 공통

com.webos.notification

활용 : 스마트홈에서 이벤트 발생시 대쉬보드로 알림.

설명 : 앱 또는 서비스에서 시스템 알림을 관리.

com.webos.service.activitymanager

활용 : 어플들의 생존여부를 확인 및 어플 간 우선순위를 지정하여 시스템 과부하 방지.

설명 : 시스템을 모니터링(작업관리자)하고 해당 이벤트가 발생하면 작업을 수행.

com.webos.service.alarm

활용 : 매 지정시간마다 특정 동작을 수행하도록 함. (실시간 서버 동기화)

설명 : 지정된 시간 간격에 특정 매개 변수를 사용해 webOS API 메서드를 호출.

- 얼굴인식 관련

com.webos.service.camera2

활용 : 카메라를 통해 얼굴인식을 진행함.

설명 : 카메라에서 이미지를 캡처하고 스트리밍하기 위한 인터페이스 제공.

- 음성인식 및 TTS 관련 (사운드 입출력)

com.webos.service.ai.voice

활용 : 음성인식을 통하여 스마트 홈 제어

설명 : 음성 인식을 사용하기 위한 인터페이스를 제공.

com.webos.service.tts

활용 : 스마트 홈에서 이벤트 발생시 음성을 통하여 사용자에게 알림.

설명 : 일반 언어 텍스트를 음성 출력으로 변환.

□ 개발 일정

[illegible]

□ 팀 구성 및 역량

No	구분	성명	팀 내 담당 업무	업무 관련 역량 (개발 언어, 프로젝트 경험 등)
1	팀장	최현식	1. 음성인식 ThinQ API 2. 활용 하드웨어 제작 및 연결 설계	○ 개발 언어 - MySQL, C언어, Python ○ 프로젝트 경험 - 스마트 홈 - 임베디드 활용 스마트 유모차 - 카카오톡 API 활용 날씨 안내 서버 개발 - 양방향 음성통화 비상벨 제작 (음성인식 기능)
2	팀원	박승운	1. 대시보드 UI / UX 제작 2. Firebase 구축	○ 개발 언어 - C, C++, JavaScript, JQUERY ○ 프로젝트 경험 - 스마트 홈 - 앱을 활용한 장애인 주차구역 표시 - 홈페이지 게시판 제작
3	팀원	이준호	1. 얼굴인식 openCV / Tensorflow 활용 2. 서버 구축	○ 개발 언어 - Python, C, C++ ○ 프로젝트 경험 - 스마트 홈 - 버스 정류장 알림 시스템