

[제19회 임베디드 소프트웨어 경진대회]  
**부문별 세부 안내사항**

**- webOS 부문 -**

webOS 기반 Car 2 Smart Home 솔루션 개발

2021. 04

## 1. 부문별 목적 및 개요

가. 목적 : LG전자에서는 학생들이 webOS 오픈소스를 활용한 다양한 서비스와 기능 등에 대한 기획 및 개발, 시제품 제작 등 제품 개발의 전반적인 프로세스를 경험하게 하고 우수한 webOS 관련 작품을 발굴·선정합니다.

### 나. 개요

- 주제 : webOS 기반 Car 2 Smart Home 솔루션 개발

차량의 탑승자를 카메라를 통해 인식하여, 사용자에게 따른 차량용 대시보드 UX를 제공하고, ThinQ AI 플랫폼을 통해 음성을 인식하고 음성을 통해 가전의 기기들을 제어하는 솔루션을 개발하는 것을 목적으로 합니다.

- 개발 플랫폼 : webOS OSE 2.10.0 이상 (필수)
- 개발 언어 : 플랫폼 애플리케이션 - HTML / JavaScript 또는 QML  
플랫폼 서비스 - C/C++ (옵션) 또는 Node.js  
그 외 - 별도 제한 없음
- 개발 재료 : 플랫폼 - 라즈베리파이 4  
- HDMI 지원 FHD 터치지원 디스플레이 1~2대  
그 외 - 별도 제한 없음.  
단, 쉽게 부러지거나 다칠 위험이 있는 재료는 피해 주시기 바랍니다.

- 개발 규격

- 소프트웨어

반드시 하나 이상의 web 기반의 대시보드 앱을 개발해야 합니다.

반드시 하나 이상의 web 기반의 서비스를 개발해야 합니다. (서비스 레벨에서의 통신)

반드시 ThinQ AI 기능(Cloud API)을 활용해야 합니다.

- 하드웨어

기본으로 라즈베리파이(webOS 탑재)이외의 별도의 제한은 없습니다.

단, 결선 장소로 가져오는데 어려움이 있는 크기는 피해 주시기 바랍니다.

## 2. 참가자격

가. 참가자격 : 성인 이상 참가 가능

나. 팀 구성 : 참가 자격을 갖춘 자로 구성된 1~5인 (팀장 포함)

### <팀 구성 규칙>

- 팀장 및 팀원은 모두 홈페이지에서 회원 가입을 완료해야 합니다.
- 팀장은 홈페이지에서 팀을 생성해야 합니다.
- 팀원은 팀에 가입 신청 하고, 팀장은 신청된 팀원 중 필요 인원을 승인해야 합니다.
- 승인된 팀원만 참가자로 인정됩니다. (팀원 간 소속 무관)
- 팀 구성 시 팀명은 아래 규칙을 준수해야 합니다.
  - 20Byte 이내 (한글 기준 10글자)
  - 띄어쓰기, 특수문자, 기호 사용 불가
  - 영문 사용 시, 대문자만 사용
  - 팀명 마지막에 ‘팀’을 붙이지 않는다. / ex) CONTEST (가능), CONTEST팀 (불가능)

### <팀원 조정 규칙>

- 팀장 변경 시, 팀장이 홈페이지에서 팀장 변경을 신청해야 하며, 사무국의 승인 하에 변경 가능합니다.
- 팀원 변경 시, 팀원이 홈페이지에서 팀 변경을 신청해야 하며, 팀장 및 사무국의 승인 하에 변경 가능합니다.
- 본선을 통과한 결선진출팀은 팀원의 변경이 불가능합니다.  
(단, 개인적인 사정에 의한 일부 팀원의 중도 포기는 사무국 승인 시 가능)

다. 참가자는 타 부문과 복수의 팀에 중복하여 참가할 수 있습니다.

(단, 본선 진출 시, 1개 부문만 선택 필요)

라. 참가등록방법 : 홈페이지>참가등록>참가안내를 참고하시기 바랍니다.

마. 경진대회 참가 등록은 규정 제3장(신청 및 접수)을 기본으로 합니다.

### 3. 경진대회 진행 일정

공 고	예 선	기술교육	본선준비	본 선	결선 1차	결선 2차	시상식
팀 등록, 예선 서류 제출 (4번 항목 참고)	심사 결과 발표	지원 교육 · 1차: 7월 · 2차: 8월	본선 서류 제출 (4번 항목 참고)	심사 결과 발표	온라인 발표 심사	오프라인 발표 심사 (2일~3일간)	시상식
4월	6월	7~8월	9월	10월	11월	11월	11월

※ 상기 내용은 사정에 따라 조정될 수 있으며, 조정된 내용은 홈페이지를 통해 공지됨

가. 공 고 : 4월, 경진대회 홈페이지(eswcontest.or.kr)를 통해 공고

나. 예 선 : 6월, 접수된 제출서류(4번 항목 참고)를 토대로 후원 기업  
실무자 및 해당 분야 전문가 서류심사 진행 및 결과발표

다. 기술교육 : 7~8월, 예선에 통과한 팀에 한하여, 분야별 기술 교육 진행  
(온·오프라인 교육 / 2회 내외)

라. 본선준비 : 9월, 개발완료보고서 및 동영상 등 본선심사를 위한 서류 제출

마. 본 선 : 10월, 접수된 제출서류(4번 항목 참고)를 토대로 후원 기업  
실무자 및 해당 분야 전문가 서류심사 진행 및 결과 발표

바. 결선 1차 : 11월, 결선 1차 온라인 발표 심사

사. 결선 2차 : 11월, 결선 2차 오프라인 발표 심사

아. 시 상 식 : 11월, 시상식 진행 (수상 팀 반드시 참석)

#### 4. 경진대회 심사 안내 및 제출 서류

구분	심사형태	제출서류	진행일정	심사항목
예선	서류심사	참가신청서, 개발계획서	- 제출: 참가 신청 시 - 결과발표: 6월	
본선	서류심사	개발완료보고서, 소스코드, 시연동영상	- 제출: 9월 초 - 결과발표: 10월	
결선	발표심사	발표PPT, 개발 작품, 최종 시연동영상	- 제출: 11월 - 결과발표: 11월	

※ 상기 내용은 사정에 따라 조정될 수 있으며, 조정된 내용은 홈페이지를 통해 공지됨

##### 가. 심사형태 안내

- 서류심사 : 제출된 서류를 토대로 후원 기업 실무자 및 해당 분야 전문가로 구성된 심사위원들이 심사 진행
- 발표심사 : 결선진출팀이 개발한 SW작품을 온·오프라인으로 심사 및 오프라인 작품 시연 진행

##### 나. 제출서류

구분	제출 서류	분량	파일타입	파일명	업로드 위치
예선	참가신청서	1page	온라인	-	홈페이지
	개발계획서	10page 이내	PDF	2021ESWContest_webOS_팀번호_팀명_개발계획서	홈페이지
본선	개발완료 보고서	30page 이내	PDF	2021ESWContest_webOS_팀번호_팀명_개발완료보고서	홈페이지
	소스코드	-	-	(GitHub 주소) github.com/사용자이름/2021ESWContest_webOS_팀번호 ※ 개발완료보고서에 해당 링크 삽입	GitHub
	시연동영상	3분 이내	-	(업로드제목) 2021ESWContest_webOS_팀번호_팀명_시연동영상 ※ 개발완료보고서에 해당 링크 삽입	YouTube
결선	발표PPT	별도 안내	별도 안내	2021ESWContest_webOS_팀번호_팀명_작품설명자료	별도 안내
	개발 작품	최소 1개 이상	-	-	
	최종 시연동영상	3분 이내	MP4	2021ESWContest_webOS_팀번호_팀명_최종시연동영상	

※ 상기 서류는 사정에 따라 조정될 수 있으며, 조정된 내용은 홈페이지를 통해 공지됨

※ 파일명 : 파란 부분만 해당 팀에서 수정하여 작성

※ 서류 제출 시 주최 측이 제시한 양식 및 기간을 준수하지 않을 경우 불이익(감점, 심사 대상 제외, 장비 지급 불가 등) 제공

※ 소스코드 및 동영상 관련 상세 내용은 “홈페이지 공지사항” 게시판 참고

- 참가신청서 : 홈페이지 상 온라인 신청
- 개발계획서 : 개발 작품에 대한 계획서
- 개발완료보고서 : 개발 작품에 대한 전체적인 내용

- 소스코드 : 개발에 사용한 프로그램(언어) 소스코드
- 시연동영상(화질 : 720p) : 실제 작품의 시연 영상 + 작품 설명 영상
- 발표PPT : 온라인 발표 심사에 사용할 발표PPT 자료
- 개발 작품 : 발표 심사에 보여줄 실제 작품
- 최종 시연동영상(화질 : 720p) : 현장 심사에서 시연을 대체할 시연 영상

#### 다. 심사 항목 안내

구분	심사 항목	배점	항목별 설명	비고
예선	기술성	30%	임베디드SW에 대한 기술 적응력 및 적용 예정 기술에 대한 이해도	
	현실성	30%	적용 기술 및 아이디어 구현 가능성 및 현실성	
	문서완성도 / 팀 구성 및 팀 역량	20%	- 문서 작성에 대한 완성도 - 개발 계획에 따른 적절한 팀 역할 배분 및 해당 팀원의 역량	
	차별성	20%	문제 해결 방법에 대한 독창성 및 적용 기술의 차별성	
본선	문서완성도	35%	프로젝트에 대한 문서화의 완성도	
	개발완성도	35%	- 개발 계획서에서 제시한 기술의 구현 여부 - 현실적 문제 해결을 위한 적절한 기술 적용 여부	
	팀 구성 및 팀 역량	30%	계획 대비 팀원의 역할별 수행 정도 및 각 팀원의 역량 향상 정도	
결선	완성도	50%	작품의 목적성에 맞는 기능의 구현 정도 및 완성도	
	독창성	25%	작품의 독창적인 아이디어 및 기술적 독창성	
	참여도	25%	전시 현장에서 발표력 및 적극성	

※ 상기 내용은 사정에 따라 조정될 수 있으며, 조정된 내용은 홈페이지를 통해 공지됨

## 5. 후원 기업 지원 내용(안) : 본선 진출 10팀

가. 기술 장비 지원 (후원 기업의 사정으로 변경될 수 있음)

- RaspberryPi 4 (with Power Adapter)
- SD Card 16GB
- HDMI Cable
- HDMI 지원 FHD 터치지원 디스플레이 1~2대
- 아두이노 R3, ESP32 또는 ESP8266 필요수량 만큼

\* 대회 종료 또는 중도 포기 시, 장비 반납 / 분실 및 파손 시 동일 제품으로 반납

나. 기술 지원 세미나 및 기술 교육

- 장비 대여, webOS 기술 교육
- 진도 체크 및 질의응답

## 6. 주의사항

가. 관련 안내 사항은 임베디드SW경진대회 공통 규정을 기반으로 합니다.

나. 접수 마감 이후, 팀 주제는 변경이 불가합니다.

다. 본 세부 안내사항 내용이 조정될 경우, 경진대회 홈페이지를 통해 안내될 수 있습니다.

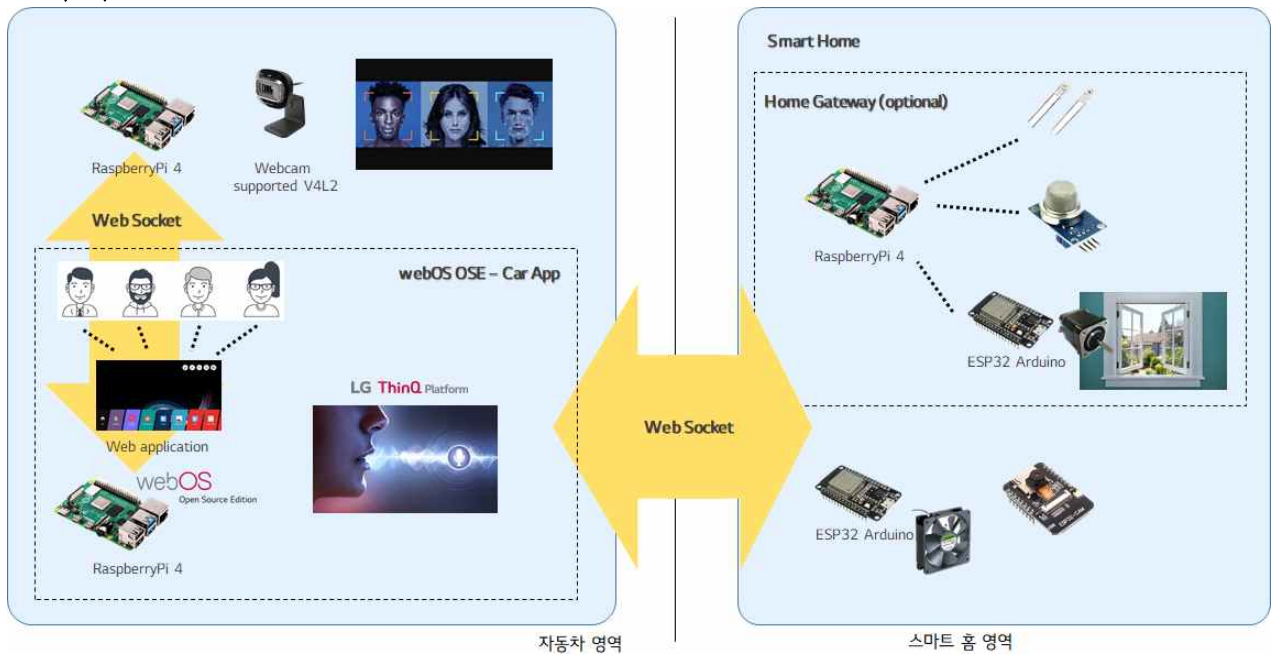
라. 참가 팀은 공통 규정 제12조 4항에 해당하는 경우, 참가자격이 상실될 수 있습니다.

마. 중도 포기 시, 참가자는 발생하는 불이익을 감수해야 합니다.

### <중도 포기 시, 주의사항>

- 포기 가능 시점은 본선서류제출 전까지 가능합니다. (개발완료보고서 제출 전)
- 포기 팀은 개발 장비를 사무국에 반납해야 합니다.
- 포기 팀은 반납확인증과 포기증서를 작성하여 제출해야 합니다.
- 포기 팀은 당해년도 또는 차년도 참가에 불이익이 있을 수 있습니다.

## 7. 예시



### (예시에 대한 실무자 코멘트)

위의 예시는 Car 2 Smart Home에서 자동차 영역과 스마트홈 영역으로 나누어 소개합니다. 자동차 영역에서는 얼굴인식을 통해 사용자를 식별하면 webOS OSE 플랫폼에서 보여지는 애플리케이션의 UI/UX가 변경됩니다. 탑승자가 AI 비서와 대화하며 스마트 홈 영역의 데이터를 확인하거나, 제어합니다. 개발팀은 기본적으로 차량과 스마트 홈 영역간의 다양한 시나리오에 대해서 생각해 보시고, 얼굴인식, 음성제어, 스마트 홈 구성 및 제어 등을 개발하시면 좋겠습니다.

### [개발 방향]

webOS 플랫폼의 장점은 web Application 과 web Service를 지원한다는 점입니다. Web page를 만들 듯 Application을 개발하고, 항상 동작하는 서비스를 Node.js 기반으로 개발 할 수 있습니다. 또한 ThinQ AI에서 제공하는 Cloud API를 통해 자동 음성 인지, 자연어 처리, TTS, 감성 TTS 등을 적용할 수 있습니다. 이를 활용하는 다양한 시나리오도 고려할 수 있습니다. 또한 Smart Home 을 구성해보고, 차량에서 실시간으로 음성을 통해 제어하는 시나리오를 고민해보고 개발해 볼 수 있습니다.



## 8. 참고자료

### 가. webOS OSE가 지원하는 주요 기능들

#### ○ Storage

- web(Browser) - LocalStorage (약 5Mb 정도)
- PDM - USB 저장장치 지원
  - <https://www.webosose.org/docs/reference/ls2-api/com-webos-service-pdm>
- Google Drive 지원
  - <https://www.webosose.org/docs/reference/ls2-api/com-webos-service-storageaccess/>

#### ○ Database

- web(Browser) - Indexed DB
- DB8 - 로컬 데이터 베이스 제공
  - <https://www.webosose.org/docs/reference/ls2-api/com-webos-service-db>

#### ○ Camera Support

- 라즈베리파이 카메라 / USB 카메라 지원
  - <https://www.webosose.org/docs/reference/ls2-api/com-webos-service-camera2/>
  - <https://www.webosose.org/docs/reference/ls2-api/com-webos-service-peripheralmanager/>

#### ○ TTS

- ThinQ AI에서 제공하는 API 활용 (필수)
  - <https://www.webosose.org/docs/reference/ls2-api/com-webos-service-tts>
- Google Cloud Account 필요

#### ○ Voice Assistant

- ThinQ AI에서 제공하는 API 활용 (필수)
  - <https://www.webosose.org/docs/reference/ls2-api/com-webos-service-ai-voice>
- Google Cloud Account 필요

#### ○ Alarm

- <https://www.webosose.org/docs/reference/ls2-api/com-webos-service-alarm>

#### ○ Bluetooth

- <https://www.webosose.org/docs/reference/ls2-api/com-webos-service-bluetooth2>

#### ○ Notification

- <https://www.webosose.org/docs/reference/ls2-api/com-webos-notification>

#### ○ Notification

- <https://www.webosose.org/docs/reference/ls2-api/com-webos-notification>

## 나. 학습 자료

### ○ web application framework

- Enact: <https://enactjs.com>
- React: <https://reactjs.org>
- Vue: <https://vuejs.org>
- jQuery: <https://jquery.com>

### ○ web design Framework

- bootstrap: <http://getbootstrap.com>
- Bulma: <https://bulma.io>
- Material Design: <https://material.io/develop>

### ○ web Technology Study

- HTML5: <https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5>
- CSS3: <https://developer.mozilla.org/en-US/docs/Web/CSS>
- JavaScript: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- w3schools: <https://www.w3schools.com>
- 생활코딩: <https://opentutorialsorg/course/1>

### ○ Platform Study

- webOS OSE
  - 오픈소스 버전의 webOS 개발을 위한 개발자 사이트
  - <https://www.webosose.org>
- webOS TV
  - webOS TV 어플리케이션 개발을 위한 개발자 사이트
  - OSE 버전의 API는 다를 수 있으므로 필요한 개념만 확인/학습
  - <http://webostv.developer.lge.com>

### ○ Native HMI Framework

- QML Application 개발
  - <https://www.webosose.org/docs/guides/development/qml-apps/qml-app-overview>

### ○ Service Component

- Native Service
  - 시스템 레벨의 서비스를 개발할 수 있음 (C/C++ 개발 필요)
  - 서비스를 추가하려면 플랫폼에 추가하여 빌드 해야 함
  - <https://www.webosose.org/docs/guides/development/native-services/native-service-overview>
- JavaScript Service
  - Node.js 기반의 서비스를 개발할 수 있음(Node.js / JavaScript 개발 필)
  - 플랫폼 빌드 없이도 서비스 추가 가능
  - <https://www.webosose.org/docs/guides/development/js-services/js-service-overview>