[제23회 임베디드 소프트웨어 경진대회]

부문별 세부 안내사항

- 스마트 가전 부문 -

오픈소스를 활용한 LG 스마트 가전 연계 AIoT 솔루션 개발

2025. 04

제23회 임베디드 소프트웨어 경진대회
The World Embedded Software Contest 2025

1. 부문별 목적 및 개요

가. 목적 : LG전자는 학생들이 스마트 가전과 AIoT 기술을 활용해 데이터를 수집하고, 이를 기반으로 사용자 맞춤형 서비스를 제공하는 임베디드 소프트웨어 솔루션 개발의 전반적인 과정을 체험할 수 있도록 합니다. 이를 통해 우수한 작품을 발굴하고 선정하는 프로그램을 운영합니다.

나. 개요

- 주제 : 오픈소스를 활용한 LG 스마트 가전 연계 AIoT 솔루션 개발

스마트 가전과 다양한 IoT장치를 제어하기 위해 가전 플랫폼에서 실행 가능한 소프트웨어 아키텍처를 구축하고, 데이터 분석을 통해 유용한 서비스를 제공하는 시스템을 개발하는 것을 목표로 함

- 개발 플랫폼 : Rasbian, Ubuntu Core 등 기술 특징에 맞는 플랫폼 이용 (별도 제한 없음)
- 개발 언어 : 플랫폼 애플리케이션 – HTML / JavaScript / Enact 등
  플랫폼 서비스 – Node.js 등
  그 외 – 별도 제한 없음
- 개발 재료 : 별도 제한 없음 (라즈베리파이 4B, TOPST보드, ESP32 등)
  단, 결선 장소로 가져오는데 어려움이 있는 크기나 재료는 피해 주시기 바랍니다.
  (전시부스 규격 : 2m × 2m, 변경될 수 있음)
- 개발 규격 : 오픈소스 플랫폼에서 동작하는 하나 이상의 web/Native 애플리케이션 제한 없는 서비스 통합 솔루션 개발

2. 참가 등록

가. 참가자격 : 대한민국 국민, 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 신청하고, 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, 사무국 승인하에 변경 가능합니다.

○ 팀원 변경 시, 팀원이 홈페이지에서 팀 변경을 신청해야 하며, 팀장 승인하에 변경 가능합니다.

○ 예선을 통과한 결선진출팀 팀원의 변경은 결선 개발완료보고서 제출 전까지 가능합니다.
(단, 개인적인 사정에 의한 일부 팀원의 중도 포기는 사무국 승인 시 가능)

다. 참가자는 타 부문과 복수의 팀에 중복하여 참가할 수 있습니다.
(단, 예선 통과 시, 1개 부문만 선택 필요)

라. 참가등록방법 : [홈페이지 〉 참가등록 〉 참가안내]를 참고하시기 바랍니다.

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

3. 경진대회 진행 일정

공 고	예 선	기술교육	결선	전시	시상식
팀 등록, 예선 서류 제출 ▶	심사 및 결과 발표 ▶	기술지원 교육 ▶	오프라인 발표 심사 ▶	오프라인 전시 ▶	시상식
4~6월	6~7월	7~9월	11월	12월	12월

가. 공　　고 : 4월 30일(수), 경진대회 홈페이지(eswcontest.or.kr)를 통해 공고

나. 예　　선 : 6~7월, 접수된 제출서류(4번 항목 참고)를 토대로 후원 기업 실무자 및 해당 분야 전문가 서류심사 진행 및 결과발표

다. 기술교육 : 7~9월, 예선에 통과한 팀에 한하여, 분야별 기술교육 및 컨설팅 진행(온·오프라인 교육 / 2회 내외)

라. 결선준비 : 10월, 개발완료보고서 및 시연 동영상 등 결선심사를 위한 서류 제출

마. 결　　선 : 11월, 접수된 제출서류(4번 항목 참고)를 토대로 오프라인 발표 평가 진행

바. 전　　시 : 12월 3~5일, 상위팀(동상 이상) 오프라인 전시 진행(수상 팀 필수 참석)

* 전시 불참 팀은 수상이 취소될 수 있음

사. 시 상 식 : 12월 3~5일, 시상식 진행 (수상 팀 필수 참석)

* 전시 및 시상식은 2025 산업기술 R&D 종합대전 내 운영되며 사정에 따라 변동될 수 있음

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

4. 경진대회 심사 안내 및 제출 서류

구분	심사형태	제출서류	진행일정	비 고
예선	서류심사	참가신청서, 개발계획서	- 서류제출: ~6월 4일 - 결과발표: 7월	온라인
결선	발표심사	개발완료보고서, 작품소개서, 소스코드, 시연동영상, 개발작품(실물)	- 서류제출: 10월 - 결과발표: 11월	오프라인

※ 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 : 현장에서 심사위원에게 작품에 대한 내용을 설명하고 답변하는 형태로 심사 진행(개발완료보고서 PPT로 발표 진행)

나. 제출서류

구분	제출 서류	분량	파일타입	파일명	업로드 위치
예선	참가신청서	1page	온라인	-	-
	개발계획서	10page 이내	PDF	2025ESWContest_스마트가전_팀번호_팀명_개발계획서	홈페이지
결선	개발완료보고서 (PPT)	20page 이내	PDF	2025ESWContest_스마트가전_팀번호_팀명_개발완료보고서	홈페이지
	작품소개서	2page	별도안내	2025ESWContest_스마트가전_팀번호_팀명_작품소개서	홈페이지
	소스코드	-	-	(GitHub 주소) github.com/사용자이름/2025ESWContest_smart_팀명 또는 팀번호(자유) ※ 작품소개서에 해당 링크 삽입	GitHub
	최종 시연동영상	3분 이내	-	2025ESWContest_스마트가전_팀번호_팀명_최종시연동영상	YouTube
	개발 작품	최소 1개 이상	-	-	-

※ 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 : 경진대회 홈페이지 내 온라인 신청(서류 접수 시 자동 생성)
- 개발계획서 : 개발 작품에 대한 계획서

- 개발완료보고서 : 개발 작품에 대한 전체적인 내용(PPT 형식)

No.	필수항목	분량
1	개발개요(프로젝트 개발 배경, 동기, 목표, 필요성 등) **소스코드 및 시연동영상 링크**	3P
2	개발 환경 설명(HW, SW구성, 기능 설계도, 개발 환경 등)	10P
3	개발 프로그램 설명(파일 구성, 함수별 기능, 흐름도, 기술적 차별성)	
4	개발 중 발생한 장애요인과 해결방안	
5	개발 결과물의 차별성(작품의 차별성 및 우수성 설명, 타 유사 제품 및 서비스와의 차별점 설명)	5P
6	개발 결과물의 파급력 및 기대효과(판매가치, 시장성, 활용성, 발전 가능성 등)	
7	개발 일정 및 업무 분장	2P

- 소스코드 : 개발에 사용한 프로그램(언어) 소스코드
- 시연동영상(화질 : 720p) : 실제 작품의 시연 영상 + 작품 설명 영상
- 작품소개서 : 발표 심사 및 전시에서 보여줄 작품에 대한 소개 자료
- 개발 작품 : 발표 심사에 보여줄 실제 작품

다. 심사 항목 안내

구분	심사 항목	배점	항목별 설명	비고
예선	독창성 및 문제해결능력	30	- 현실에서 발생되는 다양한 문제점 분석 - 문제점 분석에 따른 독창적이며 현실적인 해결방안 제시 - 아이디어가 시장에서 차별화 될 수 있는지 분석	
	기술성	30	- 임베디드SW 및 IoT 기술에 대한 적응력 및 적용 예정 기술에 대한 이해도 - 제시된 아이디어를 구현하기 위한 기술적 역량	
	문서완성도 및 활용성	30	- LG전자 가전/서비스와 연계될 수 있는 포인트에 대한 아이디어 명확성 - 개발 계획(개발의도, 개발방향, 기대효과)에 대한 명확성 - 개발 계획 문서의 가독성	
	팀 구성 및 팀 역량	10	- 개발 계획에 따른 적절한 팀 역할 배분 및 해당 팀원의 역량 - 기술 스탠 분석 및 부족한 기술 정의 및 보완 전략	
결선	완성도	60	- 개발 계획 목적에 맞는 기능의 구현 정도 및 완성도 (개발계획서에 제시된 기술의 구현정도, 개발 중 발생한 장애요인 극복방안 적절성, 개발 일정 및 업무 분장의 적절성 등)	
	독창성	20	- 제안된 기술이나 접근 방식에 대한 기존 솔루션 비교 시 독창성 정도 - 아이디어의 시장에서 차별화	
	활용성	20	- 실제 LG전자 가전/서비스와 연계될 수 있는 포인트 구현 정도 - LG전자 제품 서비스에 도입했을 때 수익성 정도	
	가산점	각 5	(1) 3가지 종류의 하드웨어(예: ESP32, RP4, TOPST)를 연동한 스마트 가전 솔루션 구현 (2) AI/Interaction 기술 적용	최대 10점

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

5. 후원 기업 지원 내용(안) : 결선 진출 10팀

가. 기술 장비 지원 (후원 기업의 사정으로 변경될 수 있음)

- RaspberryPi 4 (with Power Adapter) 1대
- SD Card 16GB
- HDMI Cable
- V4L2 지원 카메라 1대
- ESP32 CAM 1대
- HDMI 지원 FHD 터치지원 디스플레이 (7인치) 1~2대
- 아두이노 R3, ESP32 또는 ESP8266 필요수량 만큼
- TOPST R3 보드 1대

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

나. 기술 지원 세미나 및 기술 교육

- LG사이언스파크 투어
- 팀별 프로젝트 컨설팅

6. 소스코드 공개 관련 안내사항

가. 수상작은 참가팀과 사무국이 동일한 공개 여부 결정 권한을 가지며 Github를 통해 공개될 수 있습니다. (공개 정책에 대해서는 협의 가능하며 기업 후원 수상작은 후원기업에도 동일한 권한이 주어짐)

나. 본 대회에서 소스코드 공개의 의미는 (1)오픈소스로서의 공개와 (2)자작소스로서의 공개 중 하나이며, 참가팀의 동의 여부(의사 및 협의)에 따라서는 소스코드의 전체 또는 일부(선택적 또는 부분적으로)를 공개할 수 있고, (1)의 경우는 오픈소스로서 (제3자)의 활용을 제한해서는 안됩니다. (기업 후원 수상작은 후원기업의 요청에 따라 활용될 수 있음)

다. 위 내용과 같이 소스코드는 전부가 아닌 선택적 또는 부분적 공개가 가능하나 (1)의 경우는 오픈소스로서 반드시 핵심 부분이 공개범위에 포함되어야 합니다.

라. 단, 개발한 소스코드에 오픈소스가 포함되어 있을 경우 활용한 오픈소스 라이센스 규정을 위반해서는 안됩니다. (공개 대상인 수상작이 아닌 경우에도 오픈소스를 사용한 모든 참가팀에 해당)

마. 전 부문 수상작은 소스코드 관련하여 Github 공개범위를 Public(전체공개)로 유지합니다. (임의로 비공개 또는 Private으로 변경 시 수상 철회될 수 있음)

바. 제출된 소스코드는 경진대회 종료 후 홈페이지 수상작 게시판 內 개발완료보고서의 Github 링크를 통해 공개되도록 합니다.

7. 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.

바. 필요시 개인정보가 제3자에게 제공될 수 있습니다.

* 제공 시점 개별 공지 예정

〈중도 포기 시, 주의사항〉

○ 포기 팀은 개발 장비를 사무국에 반납해야 합니다.
○ 포기 팀은 반납확인증과 포기증서를 작성하여 제출해야 합니다.
○ 포기 팀은 당해년도 또는 차년도 참가에 불이익이 있을 수 있습니다.

8. 예시

※ 아래는 스마트 가전 시스템에 관한 일부 예시이며, 이밖에 더욱 다양한 기술을 적용한 스마트 가전 시스템을 구축할 수 있습니다.

1 스마트 주방 어시스턴트

스마트 주방 어시스턴트는AI, IoT, 센서 기술을 활용하여 주방에서의 요리 과정과 식재료관리를 자동화하는 시스템입니다.

음성 및 제스처 인식을 통해 사용자가 편리하게 요리를 진행할 수 있도록 가이드하고, 냉장고·오븐·전자레인지 등 스마트 가전과 연동하여 최적의 조리 환경을 제공합니다.

사용 시나리오

1. 사용자가 “파스타 레시피 알려줘”라고 하면 AI가 단계별 가이드를 제공 (AI 기반 요리 보조)
2. 오븐에 음식 넣으면 AI가 자동으로 최적 온도와 시간을 설정(스마트 가전 연동 자동 조리)
3. 냉장고 카메라가 식재료를 분석해 유통기한 임박 재료를 알려줌(AI 기반 식재료관리)
4. 가스 누출 감지 시 AI가 즉시 경고하고 환풍기를 자동 작동(주방 환경 감지 및 안전 관리)

주요기능

- AI 기반 요리 보조:음성·제스처 인식으로 요리 단계별 가이드를 제공합니다.
- 스마트 가전 연동 자동 조리:오븐·전자레인지와 연동해 자동 온도·시간을 설정합니다.
- AI 기반 식재료관리:냉장고 카메라로 식재료를분석해 유통기한을 관리합니다.
- 주방 환경 감지 및 안전 관리:가스·온습도센서로 주방 환경을 모니터링하고 경고합니다.

2 스마트 에너지 관리 시스템

스마트 에너지 관리 시스템은 가정 내 스마트 가전과 센서를 연동하여 실시간 전력 소비를 분석하고 최적화하는 솔루션입니다.

AI기반으로 전력 사용 패턴을 학습하고 자동으로 에너지 절약 전략을 추천하며, 사용자의 라이프스타일에 맞춰 에너지를 효율적으로 관리할 수 있도록 돕습니다.

사용 시나리오

1. 사용자가 앱을확인하면 AI가 가전별전력 사용량을 분석해 보여줌 (실시간 전력 소비 모니터링)
2. AI가 “TV를 밤 11시 이후 사용 안 합니다. 절전 모드 활성화할까요?“라고 제안 (AI 기반 에너지 절약 추천)
3. 외출 시 AI가 불필요한 가전을 절전 모드로 전환하고 조명을 자동 소등 (IoT기반 자동 전력 최적화)
4. 세탁기와 건조기를 전기 요금이 저렴한 시간대로 자동 예약 (에너지 효율 기반 가전 자동화)

주요기능

- 실시간 전력 소비 모니터링:가전별 전력 사용량을 분석해 절약 가능성을 제공합니다.
- AI 기반 에너지 절약 추천:사용 패턴을 분석해 최적의 절전 모드를 추천합니다.
- IoT기반 자동 전력 최적화:외출 시 조명·가전을 절전 모드로 자동 전환합니다.
- 에너지 효율 기반 가전 자동화:전기 요금이 저렴한 시간에 세탁기·건조기를 예약합니다.

[개발 방향]

오픈소스를 활용한 LG 스마트 가전 연계 AIoT솔루션 개발은 사용자 맞춤형 서비스 제공하는 것을 목표합니다. LG전자 가전 및 스마트 홈 기술과 연동되는 구조를 설계하고, Web과 Native 환경에 제한 없는 서비스를 개발할 수 있습니다. 백그라운드 서비스를 활용하면 애플리케이션이 실행되지 않아도 지속해서 데이터를 수집하고 처리할 수 있어 스마트 가전의 실시간성과 연속성을 보장할 수 있습니다. 여기에 인공지능, 빅데이터, 클라우드 기술을 융합하면 사용자 행동 분석, 자동 제어, 상황 기반 추천 기능 등을 통해 지능적이고 효율적인 사용자 경험을 제공할 수 있을 것으로 생각합니다. 다양한 센서 정보를 기반으로 이를 활용하는 시나리오를 고민해 보고 새로운 스마트 가전 사용자를 위한 Interactive UX/UI도 함께 고민하고 개발해 보기 바랍니다.

9. 참고자료

가. 플랫폼 운영체제

○ Yocto Project

Yocto Linux 기본 튜토리얼: Yocto 프로젝트를 활용한 임베디드 리눅스 배포판 생성에 대한 기본 가이드입니다.
Yocto 프로젝트 퀵 스타트 가이드: Yocto 프로젝트의 빠른 시작을 위한 공식 문서입니다.

○ Raspberry Pi OS 및 GPIO 제어 라이브러리

pigpio 를 이용한 Raspberry Pi GPIO 제어: pigpio 라이브러리를 설치하고 사용하는 방법을 설명합니다.
WiringPi 설치 및 사용법: Raspberry Pi에서 WiringPi를 설치하고 사용하는 방법을 안내합니다.

○ Zephyr RTOS

Zephyr RTOS 튜토리얼: Zephyr RTOS의 기본 개념과 사용법을 단계별로 설명하는 가이드입니다.
Zephyr 프로젝트 공식 시작하기 가이드: Zephyr RTOS의 설치 및 기본 사용법을 안내하는 공식 문서입니다.

나. 통신/인프라

○ Eclipse Mosquitto (MQTT Broker)

Eclipse Mosquitto 설치 및 실행 가이드: Mosquitto 브로커의 설치부터 MQTT 클라이언트를 이용한 테스트까지의 과정을 설명합니다.
Eclipse Mosquitto 공식 사이트: Mosquitto에 대한 공식 문서와 다운로드 링크를 제공합니다.

○ Node-RED

Node-RED 튜토리얼: Node-RED의 기본 사용법과 예제를 제공하는 공식 튜토리얼입니다.
Node-RED 시작하기 가이드: Node-RED의 설치 및 기본 흐름(flow) 생성 방법을 안내합니다.

○ Home Assistant

Home Assistant 시작하기: Home Assistant의 설치 및 기본 설정 방법을 안내하는 공식 문서입니다.
Home Assistant 문서: 다양한 기능과 설정에 대한 상세한 문서를 제공합니다.

○ open62541 (OPC UA)

open62541 튜토리얼: 간단한 서버 구축: open62541을 이용하여 OPC UA 서버를 구축하는 방법을 설명합니다.
open62541 튜토리얼 목록: 다양한 기능 구현을 위한 튜토리얼을 제공합니다. 3. 미들웨어/AI 추론/데이터 처리 레이어

다. 미들웨어/AI 추론/데이터 처리

○ TensorFlow Lite

TensorFlow Lite 시작하기: 모바일 및 임베디드 장치에서 TensorFlow 모델을 변환하고 실행하는 방법을 안내합니다.
TensorFlow Lite On ESP32: ESP32에서 TensorFlow Lite를 실행하는 방법을 설명합니다.

○ OpenCV

OpenCV 튜토리얼: 임베디드 시스템에서 OpenCV를 사용하는 방법을 설명합니다.
ESP32 CAM을 이용한 객체 탐지 및 식별: ESP32 CAM과 OpenCV를 활용한 객체 탐지 예제를 제공합니다.

○ ONNX Runtime

ONNX Runtime IoT 및 엣지 장치 배포: IoT 및 엣지 장치에서 ONNX Runtime을 활용하는 방법을 안내합니다.
Raspberry Pi에서의 ONNX Runtime 배포: Raspberry Pi에서 ONNX Runtime을 사용하여 이미지 분류를 수행하는 방법을 설명합니다.

○ Edge Impulse

Edge Impulse 시작하기: Edge Impulse의 기본 사용법과 튜토리얼을 제공합니다.
종단 간 튜토리얼: 다양한 센서를 활용한 예제를 안내합니다.

○ Apache Kafka (Edge Streaming)

Kafka Streams 튜토리얼: Kafka Streams를 활용한 스트림 처리 애플리케이션 개발 방법을 안내합니다.
Kafka Streams 시작하기: 실용 가이드: Kafka Streams의 기본 개념과 예제를 제공합니다.

라. 애플리케이션/UX

○ Electron + Vue.js

Electron과 Vue.js를 이용한 데스크탑 애플리케이션 만들기: Electron과 Vue.js를 결합하여 데스크탑 애플리케이션을 개발하는 방법을 설명합니다.
Electron과 Vue.js를 활용한 간단한 애플리케이션 구축: 단계별 가이드를 제공합니다.

○ Flutter Embedded

임베디드 장치에서의 Flutter 지원: 임베디드 시스템에서 Flutter를 활용하는 방법을 안내합니다.
Flutter를 임베디드 장치에서 실행하기: Flutter를 다양한 임베디드 장치에 적용하는 방법을 설명합니다.

○ Mycroft AI (오픈소스 음성비서)

Mycroft AI 공식 튜토리얼: Mycroft의 설치, 스킬 개발, 하드웨어 연동에 대한 자세한 문서입니다.
Mycroft 스킬 개발 가이드: 사용자 정의 음성 명령 기능 추가 방법 안내

○ Matter (CSA, Connected Standards Alliance)

Matter 개발자 가이드: Matter 표준 기반 스마트 디바이스 통신 개발 문서
ESP32와 Matter 연동 실습: Espressif의 Matter 개발 프레임워크 (ESP-IDF 기반)

마. 툴체인/개발 보조/테스트/데이터

○ PlatformIO

PlatformIO 공식 시작하기: 다양한 보드 및 프레임워크 지원
IoT 프로젝트를 위한 PlatformIO 실습

○ Arduino IDE + 생태계 라이브러리

Arduino 공식 튜토리얼
IoT Smart Kitchen 시나리오 예제

○ MQTT 테스트 툴 (MQTT.fx, MQTT Explorer 등)

MQTT.fx 사용법 가이드
MQTT Explorer 다운로드 및 사용법

○ OpenHAB Test Framework

OpenHAB 공식 테스트 문서
OpenHAB 기반 스마트홈 시뮬레이션 구성 예제