

[제24회 임베디드 소프트웨어 경진대회]
부문별 세부 안내사항

- 스마트 가전 부문 -

사용자를 이해하고 먼저 행동하는 개인 맞춤형 AI 가전 서비스

2026. 05

1. 부문별 목적 및 개요

가. 목적 : LG전자는 학생들이 스마트 가전과 AIoT 기술을 활용하여 데이터를 수집·해석하고, 이를 기반으로 사용자 컨텍스트(발화, 행동, 시간/장소, 환경 센서, 기기 상태)를 이해하여 상황 변화에 선제적으로 반응함으로써 초개인화된 경험을 제공하는 임베디드 소프트웨어 솔루션을 설계·구현하는 전 과정을 경험할 수 있도록 합니다.

본 대회를 통해 가전 플랫폼 기반 소프트웨어 아키텍처에 대한 이해를 제고하고, 실질적인 문제 해결 역량을 갖춘 우수한 작품을 발굴·선정하고자 합니다.

나. 개요

- 주제 : 사용자를 이해하고 먼저 행동하는 개인 맞춤형 AI 가전 서비스

LG전자에서 제공하는 임베디드 SW 플랫폼을 중심으로 사용자의 맥락을 해석하고, 부분 입력과 상황 변화를 기반으로 사용자의 의도에 실시간으로 반응하는 초개인화를 위한 스마트 가전 서비스를 설계한다.

- 개발 플랫폼 : LG전자에서 제공하는 임베디드 SW 플랫폼* 기반 개발

* 해당 플랫폼은 Raspberry Pi 4 및 Raspberry Pi Zero에 사전 탑재되어 제공되며, 해당 기기 외 추가 장비 사용 시 플랫폼 선택은 제한하지 않습니다.

- 개발 언어 : 플랫폼 애플리케이션 : Flutter, React 등

플랫폼 서비스 : Node.js 등

기타 구성 요소 : 별도 제한 없음

- 개발 재료 : Raspberry Pi 4, Raspberry Pi Zero, ESP32 사용 필수

그 외 제한 없음

단, 결선 장소 반입 및 전시가 어려운 크기·형태의 재료는 지양
(전시 부스 규격 : 2m × 2m, 변경될 수 있음)

- 개발 규격 : LG전자에서 제공하는 플랫폼에서 동작하며, 애플리케이션 제한 없는 하나 이상의 서비스 통합 솔루션 개발

- 참고 사항 : 결선 진출팀은 LG전자에서 제공하는 플랫폼에 대한 비밀 유지를 위한 서약서를 작성하게 됩니다.

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>

- 서로간 협의가 완료된 1~5인으로 팀을 구성하고 팀장을 선정합니다.
- 참가신청서에 기재 및 개인정보수집·이용에 동의한 팀원만 참가자로 인정됩니다.
(팀원 간 소속 무관)
- 팀 구성 시 팀명은 아래 규칙을 준수해야 합니다.
 - 20Byte 이내 (한글 기준 10글자)
 - 띄어쓰기, 특수문자, 기호, “테스트” 및 “TEST” 문구 사용 불가
 - 영문 사용 시, 대문자만 사용
 - 팀명 마지막에 ‘팀’을 붙이지 않습니다. / ex) EMBEDDED (가능), EMBEDDED팀 (불가)

<팀원 조정 규칙>

- 팀장 및 팀원 변경은 모든 팀원의 동의하에 가능하며, 사무국에 해당 내용을 요청합니다.
- 예선을 통과한 결선진출팀의 팀장 및 팀원 변경은 결선 개발완료보고서 제출 이전까지만 허용됩니다.
(단, 제출 후 개인적인 사정에 의한 일부 팀원의 중도 포기는 사무국 승인 시 가능)

다. 참가자는 타 부문과 복수의 팀에 중복하여 참가할 수 있습니다.

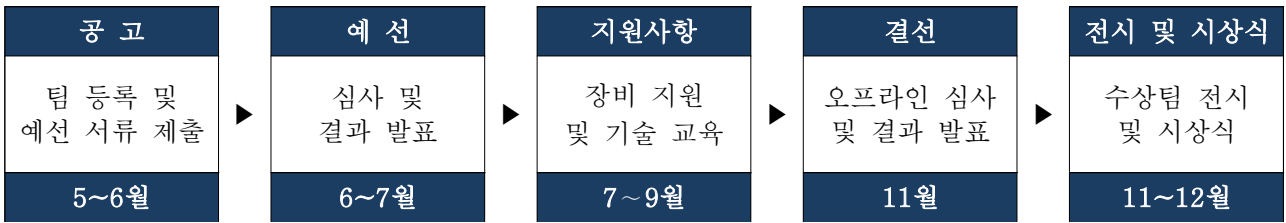
(단, 예선 통과 시, 1개 부문만 선택 필요)

라. 참가신청방법 : 구글폼을 통해 예선 서류 일체를 제출합니다.

[홈페이지 > 참가등록 > 참가안내]를 참고하시기 바랍니다.

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

3. 경진대회 진행 일정



가. 공 고 : 5월 6일(수), 경진대회 홈페이지(eswcontest.or.kr)를 통해 공고

나. 예선접수 : ~6월 10일(수), 개발계획서 등 예선심사를 위한 서류 제출

다. 예 선 : 6~7월, 접수된 제출서류를 토대로 후원 기업 실무자 및 해당 분야 전문가 서류심사 진행 및 결과발표

라. 지원사항 : 7~9월, 결선 진출팀 대상 장비 지원 및 오프라인 기술교류회/ 세미나, LG 이노베이션 갤러리 투어, 온라인 컨설팅 진행

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

바. 결 선 : 11월, 접수된 제출서류를 토대로 오프라인 발표 평가 진행

사. 전 시 : 11~12월, 본상 수상팀(동상 이상) 오프라인 전시 진행(필수 참석)

* 전시 불참 팀은 수상이 취소될 수 있음

아. 시 상 식 : 11~12월, 시상식 진행(수상팀 필수 참석)

* 전시 및 시상식은 2026 코리아 테크 페스티벌 내 운영 예정이며, 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	비고
예선	서류심사	참가신청서(자동생성), 개발계획서	- 서류제출: ~6월 10일 - 결과발표: 7월	온라인
결선	발표심사	개발완료보고서, 작품소개서, 소스코드(URL), 시연 동영상, 개발 작품	- 서류제출: 10월 - 발표심사: 11월 - 결과발표: 11월	오프라인

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

가. 심사방식

- 서류심사 : 제출 서류 기반 후원 기업 실무자 및 해당 분야 전문가로 구성된 심사위원 심사 진행
- 발표심사 : 현장 시연 및 심사위원에게 작품에 대한 내용을 설명하고 답변하는 형태로 심사 진행

나. 제출서류

구분	제출 서류	분량	파일타입	파일명	제출방법
예선	참가신청서	1page	엑셀	2026ESWContest_스마트가전_팀명_참가신청서	구글폼
	개발계획서	7page 이내	PDF	2026ESWContest_스마트가전_팀명_개발계획서	구글폼
결선	개발완료보고서 (PPT)	20page 이내	PDF	2026ESWContest_스마트가전_팀명_개발완료보고서	구글폼
	작품소개서	2page	한글/워드	2026ESWContest_스마트가전_팀명_작품소개서	구글폼
	소스코드	-	-	(GitHub 주소) github.com/사용자이름/2026ESWContest_smart_팀명 ※ 개발완료보고서 및 작품소개서에 해당 링크 삽입	GitHub
	시연동영상	3분 이내	-	(동영상 업로드제목) 2026ESWContest_스마트가전_팀명_시연동영상 ※ 개발완료보고서 및 작품소개서에 해당 링크 삽입	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)이 있을 수 있음

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

- 참가신청서 : 대회 참가를 위한 팀원 정보 작성 및 개인정보수집·이용 동의
- 개발계획서 : 개발 작품에 대한 계획서(한글/워드 형식)

- 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 및 설명 등이 포함된 영상
- 작품소개서 : 2P 분량 작품 소개서(디렉토리북, 전시, 역대 수상작 등에 활용)
- 개발 작품 : 발표 심사에 보여줄 실제 작품

다. 심사 항목 안내

구분	심사 항목	배점	항목별 설명	비고
예선	독창성 및 문제해결능력	30	- 현실에서 발생하는 다양한 문제점 분석 - 문제점 분석에 따른 독창적이며 현실적인 해결방안 제시 - 아이디어가 시장에서 차별화 될 수 있는지 분석	
	기술성	30	- 임베디드SW 및 AIoT 기술에 대한 적응력 및 적용 예정 기술에 대한 이해도 - 제시된 아이디어를 구현하기 위한 기술적 역량	
	문서완성도 및 활용성	30	- 개발 계획(개발의도, 개발방향, 기대효과)에 대한 명확성 - 개발 계획 문서의 가독성 및 구체성	
	팀 구성 및 팀 역량	10	- 개발 계획에 따른 적절한 팀 역할 배분 및 해당 팀원의 역량 - 기술 스택 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	- 일반 사용자가 직관적으로 가치를 느낄 수 있는 편리하고 자연스러운 인터렉션 - 제품 서비스에 도입했을 때 즉시 적용 가능한 수준의 비즈니스 모델 가능성	
	가산점	각 5	(1) LG전자에서 제공하는 임베디드 소프트웨어 플랫폼을 기반으로 3가지 종류의 하드웨어(예: ESP32, RPi4, RPi Zero 필수)를 연동한 AI 가전 솔루션 구현 (2) 멀티모달 확장성(음성, 영상, 조도 등 다중 센서 데이터를 융합해 상황 인지의 정확도 향상)	최대 10점

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

5. 후원 기업 지원 내용(안) : 결선 진출 10팀

※ 상황에 따라 변동될 수 있음

가. 장비 지원

- Raspberry Pi 4 (with Power Adapter) 1대
- Raspberry Pi Zero (with Power Adapter) 1대
- SD Card 16GB 2개
- HDMI Cable
- mini-HDMI 케이블
- V4L2 지원 카메라 1대
- ESP32 CAM 1대
- HDMI 및 FHD 터치지원 디스플레이(7인치) 1~2대
- 아두이노 R3, ESP32, ESP8266 각 1대

* 대회 종료 또는 중도 포기 시, 장비 반납 / 분실 및 파손 시 동일 제품 구매 후 반납

나. 기술지원교육

- 기술교류회 및 기술 세미나
- LG 이노베이션 갤러리 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 라이선스 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 개발완료 보고서의 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>

- 포기팀은 개발 장비를 사무국에 반납해야 합니다.
- 포기팀은 반납확인증과 포기증서를 작성하여 제출해야 합니다.
- 포기팀은 당해년도 또는 차년도 참가에 불이익이 있을 수 있습니다.

8. 예시

※ 아래는 초개인화 AI Agent 가전 솔루션에 관한 일부 예시이며, 이밖에 더욱 다양한 기술을 적용한 스마트 가전 시스템을 구축할 수 있습니다.

① 마이 스마트 슬립 가디언 (My Smart Sleep Guardian)

사용자의 수면 중 응얼거림, 기침, 거친 호흡 등 스트리밍 사운드를 실시간 분석하여, 수면 장애나 주변 환경 불편 요소를 인지하고 즉시 가전(조명, 가습기, 에어컨)을 제어하는 초개인화 숙면 솔루션입니다.

○ 사용 시나리오

1. 사용자가 잠결에 "아, 추워..." 혹은 "음, 너무 건조..."라고 작게 읊조리기 시작함
2. 에이전트가 문장이 끝나기 전 '온도/습도 불만족' 인텐트를 감지하고 즉시 현재 방 안의 센서값 확인
3. 클라우드에서 AI는 사용자의 과거 수면 데이터와 매칭하여 최적 온도값을 도출하고, 디바이스 측 AI는 즉시 가습기/온풍기 예열 시작
4. 사용자가 말을 마칠 즈음, 이미 기기들이 저소음 모드로 작동하며 수면 환경을 최적화함

○ 주요기능

- ESP32 수준에서도 동작하는 초경량 음성 파편 감지
- 로컬의 실시간 소음 분석과 클라우드의 개인 수면 이력 데이터 융합
- 발화 의도 파악 즉시 가전의 초기 구동(Warm-up)을 수행하여 응답 지연 최소화
- 메인 게이트웨이와 말단 제어 노드 간의 끊김 없는 명령 전달 및 하드웨어 제어

② 세이프티 셰프 에이전트 (Safety-Chef Agent)

주방의 소음 환경에서 사용자의 요리 행위와 음성을 실시간으로 연관 분석하여, 화재 위험 및 조리 단계 누락을 사전에 감지하고 가스 밸브·후드 등을 자동으로 제어하는 지능형 안전 관리 서비스입니다.

○ 사용 시나리오

1. 사용자가 요리 도중 "어? 전화 왔네?"라며 주방을 벗어나려 하거나 "깜빡했다..."라고 혼잣말함
2. 사용자가 주방을 벗어나거나 조리 기기를 켜놓고 잊은 경우, 에이전트가 이를 인식하고 인덕션·가스레인지 상태를 즉시 확인

3. 위험도가 높다고 판단될 경우, 클라우드 판단을 기다리지 않고 기기가 즉시 경고음을 발생시키거나 화력을 1단계로 낮춤
4. 사용자가 다시 돌아오면 "화력을 낮춰 두었습니다. 다시 올릴까요?"라며 연속적인 컨텍스트 제공

○ 주요기능

- 스트리밍 발화와 기기 상태(온도, 연기 등)를 실시간으로 결합한 의도 추론
- 네트워크 장애 시에도 로컬 시스템(Raspberry Pi) 독립적으로 안전 제어 수행
- 기기에서 주방 소음(물 소리, 후드 소리)을 걸러내고, 사용자가 의도한 명령어만 추출하는 전처리
- 연기가 감지되기 전, 사용자의 요리 시작 발화만으로 미리 후드 단계를 조절하는 선제적 제어

[개발 방향]

가전 기기와 AI 기술을 밀접하게 결합하여, 마치 영화 속 '자비스(Jarvis)'와 같이 사용자를 이해하고 돕는 초개인화된 에이전트 솔루션 개발을 목표로 합니다. 참가팀은 하이브리드 AI 구조(Cloud AI + Edge AI)를 기반으로 라즈베리파이와 ESP32 등 임베디드 하드웨어를 제어하며, 사용자의 발화나 행동 맥락에 따라 가전이 지능적으로 반응하는 시스템을 설계해야 합니다.

단순히 정해진 명령을 수행하는 단계를 넘어, 스트리밍되는 음성이나 텍스트 정보를 실시간으로 파악하여 사용자의 숨은 의도를 찾아내는 창의적인 접근을 권장합니다. 클라우드의 풍부한 정보와 기기의 실시간 제어 기능을 조화롭게 활용하여, 가전 기기가 단순한 도구를 넘어 사용자의 삶을 보조하는 능동적인 비서가 되는 시나리오를 구상해 보시기 바랍니다. 다양한 센서 데이터와 AI 모델을 융합해 사용자에게 이전에 없던 인터랙티브한 경험과 편리함을 제공하는 혁신적인 가전 제어 솔루션을 기대합니다.

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.

○ 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. X
[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](#)