

# 프로젝트 제안서: Office Guardian

"환경 관리부터 안전, 스스로 충전하는 생능 능력까지 갖춘 지능형 오피스 가디언"

## 1. 프로젝트 개요

- 주제:** AIoT 기반 자율주행 오피스 올인원 케어 시스템
- 목표:** 주/야간 상시 순찰을 통해 환경(온습도)과 안전(낙상/화재/보안)을 관리하며, 배터리 상태에 따라 자율 복귀 및 자기 부착식 충전을 수행하는 로봇 구축.
- 개발 기간:** 6주 (1.5개월)
- 참여 인원:** 6명 (임베디드/로봇 4명 + 타 트랙 2명)

## 2. 핵심 시나리오 (All-Day Care Cycle)

- 주간(09-18시):** 환경 패트롤(온습도 측정 및 최적 온도 제안) 및 안전 모니터링(낙상/화재 감지).
- 야간(18시-):** 보안 순찰(미퇴실자 확인) 및 환경 정리(의자 정렬/소지품 방치 확인/소등).
- 상시:** 실시간 배터리 모니터링을 통한 자율 복귀 및 **서보-비전 협응 기반 정밀 도킹(자기 부착 충전)**.

## 3. 기능 명세 및 우선순위 (Scope & Priority)

### Core Features (필수 구현 - MVP)

- 자율주행:** 지정 경로 순찰(Patrol) 및 Nav2 기반 동적 장애물 회피.
- 환경 센싱:** 온습도/가스 데이터 수집 및 실시간 관리자 대시보드 시각화.
- 배터리 매니지먼트:** 전압 ADC 센싱을 통한 실시간 잔량 모니터링 및 저전력 복귀 로직.
- 통신 시스템:** 로봇(ESP32/Jetson) - 서버 - 앱 간 MQTT 기반 양방향 데이터 연동.

### Advanced Features (도전 과제)

- 액티브 도킹:** 서보 모터 기반 카메라 각도 제어로 스테이션 마커를 정밀 추적하여 자기 부착식 충전 수행.
- 고급 비전 AI:** 자세 추정(Pose Estimation) 기반 낙상 사고 감지 및 연기 인식 모델.
- 데이터 인사이트:** 축적된 데이터를 분석하여 에너지 절감 효과 리포트 자동 생성.

## 4. 시스템 아키텍처 (안정성 및 효율성 중심)

### 로봇 및 제어부 (Embedded Team)

- Main Computing:** **NVIDIA Jetson Orin Nano** (Vision AI 및 ROS 2 상위 제어)
- Main MCU:** **ESP32-S3 (Dual Core)**
  - Core 0:** 모터 PID 제어 및 서보 기반 카메라 트래킹 (Real-time Task)
  - Core 1:** Wi-Fi/MQTT 통신 및 배터리 전압 모니터링 (Communication Task)
- Hardware:** LiDAR, Depth Camera, 온습도/가스 센서, 전압 분배 회로, 자기 부착형 충전 단자.

### 서비스 및 확장부 (Service & Expansion Team)

- **Backend & Infra:** Spring Boot / Fast API / MQTT Broker
- **Frontend & App:** React (관제 웹) / Flutter (사용자 알람 앱)

## 5. 역할 분담 및 협업 문화 (R&R)

"주 담당 영역을 두되, 페어 프로그래밍과 지식 공유를 통해 기술 경계를 허뭅니다."

역할	담당 업무 상세	비고
시스템 통합	ROS 2 아키텍처 설계, Jetson-ESP32 통신 최적화, MQTT 브로커 구축	PM
자율주행/도킹	SLAM 지도 생성, Nav2 플러그인 튜닝, 서보-비전 협응 도킹 로직	
비전 AI	객체 인식(YOLO), ArUco 마커 기반 거리 산출, (Adv) 낙상/화재 감지	
제어/하드웨어	ESP-IDF 기반 펌웨어 개발, 배터리/충전 회로 설계, 서보 및 모터 제어	이영현
서비스 개발	실시간 관제 대시보드, 배터리/환경 데이터 시각화, 비상 알람 푸시 서비스	(타 트랙)

## 6. 팀원들을 위한 파트별 학습 키워드 (Learning Path)

프로젝트 시작 전, 각 파트별로 익혀야 할 핵심 개념입니다.

### 🌀 Embedded & Control

1. **ESP-IDF 환경:** FreeRTOS의 Task 생성 및 듀얼 코어(Task Pinning) 활용법.
2. **Peripheral 제어:** PWM(모터 속도), ADC(배터리 전압), I2C(온습도 센서) 통신 실습.
3. **제어 알고리즘:** 위치 및 속도 제어를 위한 PID 제어 루프의 이해.

### 🤖 ROS 2 & Navigation

1. **ROS 2 통신:** Topic/Service/Action의 차이와 커스텀 메시지 타입 정의.
2. **Navigation2:** SLAM(지도 생성) 및 하이퍼파라미터(Costmap, Planner) 튜닝.
3. **TF2:** 로봇의 좌표계(Base\_link, Odom, Map)와 카메라/라이다 위치 관계 이해.

### 👁 Vision AI & Perception

1. **OpenCV:** ArUco 마커 검출 라이브러리 사용법 및 카메라 캘리브레이션.
2. **Deep Learning:** YOLOv8 모델 학습 및 Jetson Orin Nano용 TensorRT 엔진 변환.
3. **협응 로직:** 비전 인식 데이터(x, y 오차)를 제어부(ESP32)로 전송하는 파이프라인 구축.

### 🌐 Service & Web

1. **MQTT:** JSON 데이터 규격 설계 및 Pub/Sub 구조의 비동기 메시징 이해.
2. **Data Visualization:** 실시간 전송되는 센서 데이터를 차트로 시각화하는 기법.
3. **Push Notification:** FCM 등 외부 서비스 연동을 통한 비상 상황 알람 로직.

## 7. 리스크 관리 및 대안 (Risk Management)

구분	잠재적 리스크 (Risk)	대응 방안 (Plan B)
기술적 난제	물리적 자기 부착 도킹 실패 시	스테이션 앞 도달을 성공 지표로 삼고, 충전은 수동 알림으로 전환.
성능 저하	젯슨 리소스 부족으로 AI 속도 저하 시	Pose Estimation 등 무거운 모델을 제외하고 가속도 센서와 단순 비전 결합.
자원 확보	딥스 카메라/젯슨 등 지원 불가 시	일반 웹캠과 초음파 센서를 조합하여 시나리오 규모 축소 (Pivot).

8. 자원 확보 및 예산 계획

- **SSAFY 인프라 활용:** 기자재실 대여 현황 확인 (Jetson, LiDAR, 3D 프린터 등).
- **예산 확보:** 필수 센서 미지원 시 팀원 간 합의를 통해 공동 구매 또는 사비 구매 후 개인 소유 결정.