



드론 GCS 소프트웨어 개발 엔드투엔드 구현 순서

■ Notebooks	■ <u>DJI GCS.project</u>
📁 보관함	<input type="checkbox"/>
⚙️ 상태	작성 중
🕒 생성 일시	@2025년 7월 14일 오후 4:42
★ 즐겨찾기	<input type="checkbox"/>
🕒 최종 편집 일시	@2025년 7월 14일 오후 4:47

드론 GCS 소프트웨어 개발 엔드투엔드 구현 순서

본 문서는 드론 지상관제시스템(GCS) 개발을 위한 전체 소프트웨어 구현 순서를 체계적으로 정리한 가이드입니다. 각 단계별 핵심 목표와 실무 팁을 포함하여, 실제 개발 및 통합에 바로 활용할 수 있도록 구성했습니다.

1. STT(음성인식) 비교분석 및 모델 선정

- 벤치마킹 대상:
 - Google Speech Recognition
 - Vosk
 - OpenAI Whisper
- 주요 평가 기준:
 - 한국어 명령어 인식 정확도
 - 응답 속도
 - 오프라인 처리 가능성
- 실행 방법:

- 실제 드론 명령어(이륙, 착륙 등)로 각 엔진 벤치마크
- 결과 비교 및 최적 모델 선정
- **참고:**
 - STT 비교분석 보고서 (GitHub).
 - STT 비교분석 노션
- **결과:**
 - 최종 선정된 STT 엔진을 코드에 통합

2. 드론 미션 이벤트 커맨드 정의

- **핵심 명령어/이벤트 유형 선정:**
 - 이륙, 착륙, 웨이포인트 이동, 고도 유지, 객체 탐지 등
- **매핑 테이블 설계:**
 - 자연어 명령 ↔ 구조화 커맨드 ↔ 실제 SDK 함수
 - 예시: "이륙해" → **CMD: TAKEOFF** → **monitoredTakeoff()**
- **이벤트 구조체 및 데이터 포맷 표준화:**
 - JSON, Protobuf 등 활용
 - 예시:

```
json{
  "event": "TAKEOFF",
  "altitude": 50
}
```

3. 한국어 LLM 모델 비교 및 선정

- **후보군 선정:**
 - Hugging Face 등에서 경량 한국어 LLM 탐색
- **평가 기준:**
 - 성능(정확도), 파인튜닝 용이성, 라이선스, 온프레미스 배포 가능성
- **모델 선정 및 파인튜닝 전략:**
 - 최종 후보 선정 후, 미션 명령어 데이터셋 기반 파인튜닝 계획 수립

4. 드론 명령어 프롬프트 튜닝

- **입력/출력 포맷 설계:**
 - 예:
 - 입력: "고도 50미터로 이륙 후 웨이포인트 A로 이동"
 - 출력: `CMD: TAKEOFF ALT=50; GOTO_WAYPOINT A`
- **프롬프트 엔지니어링:**
 - LLM이 다양한 자연어 명령을 일관된 이벤트로 변환하도록 설계
- **테스트 및 검증:**
 - 테스트셋 기반 튜닝, 실제 명령어에 대한 변환 정확도 검증

5. 비전 모델(YOLOv8 등) 통합 및 메타데이터 연동

- **비전 모델 배포:**
 - 드론 온보드 컴퓨터에 YOLOv8 등 객체 탐지 모델 배포
- **실시간 추론 파이프라인 구축:**
 - 이미지 → 객체 탐지 → 메타데이터 생성
- **메타데이터 구조 설계:**
 - 예시:

```
json{
  "timestamp": "2025-07-14T14:22:00Z",
  "location": {"lat": 37.12345, "lon": 127.12345, "alt": 50},
  "objects": [
    {"type": "person", "confidence": 0.94, "bbox": [120, 80, 180, 200]}
  ]
}
```

- **GCS/LLM 연동:**
 - 비전 이벤트와 LLM 미션 생성의 통합 워크플로우 구현

6. 출력된 미션 SDK 연동 및 실행

- **LLM 미션 이벤트 → DJI SDK 연동:**
 - LLM이 생성한 구조화 커맨드를 DJI OSDK, ROS, Cloud API 등으로 변환 및 실행
- **상태 피드백 및 로그 관리:**
 - 명령 실행 결과 모니터링, 로그 저장

7. 사용자 쿼리 입력 후 실험 및 개선

- 실제 사용자 쿼리/음성 명령 실험:
 - 전체 시스템 통합 테스트
- 문제점 분석 및 개선 반복:
 - 인식 오류, 이벤트 매핑, 미션 실행 등 이슈 분석 및 개선

참고: DJI 드론 커맨드 및 연동 자료

- 공식 SDK/문서:
 - [DJI Onboard SDK 공식 문서](#)
 - [DJI OSDK GitHub](#)
 - [ROS dji_sdk 위키](#)
- 주요 커맨드 예시:

명령 기능	OSDK 함수/서비스	ROS 토픽/서비스
이륙	<code>monitoredTakeoff()</code>	<code>/dji_sdk/drone_task_control</code> (task=4)
착륙	<code>land()</code>	<code>/dji_sdk/drone_task_control</code> (task=6)
복귀	<code>goHome()</code>	<code>/dji_sdk/drone_task_control</code> (task=1)
위치 이동	<code>moveByPositionOffset()</code>	<code>/dji_sdk/flight_control_setpoint_ENUposition_yaw</code>
속도/자세 제어	Joystick API	<code>/dji_sdk/flight_control_setpoint_ENUvelocity_yawrate</code>
페이로드 제어	<code>cameraAction()</code>	<code>/dji_sdk/camera_action</code>

전체 워크플로우 요약

1. STT 엔진 선정 및 통합
2. 드론 미션 이벤트/커맨드 정의 및 매핑
3. 한국어 LLM 모델 선정 및 프롬프트 튜닝
4. 비전 모델 통합 및 메타데이터 연동
5. LLM 미션 이벤트 → SDK/ROS/Cloud API 연동
6. 실제 쿼리 실험 및 반복 개선