

1 현재 시스템 개요

◆ 주요 구성

구분	역할	기술 스택
AI 영상 모듈	카메라 또는 영상 파일에서 반려견을 인식, 품종 및 개체 식별 수행	Python, YOLOv8, ORB Feature Matching, ResNet50
급식 제어 모듈	급식량 계산, 서보모터 제어, 로드셀을 통한 실시간 중량 피드백	Arduino, Servo Library, HX711 Library
통신 방식	Python ↔ Arduino 간 Serial (115200 baud)	pySerial 라이브러리 사용

2 문제 상황

◆ 증상

- AI 모듈에서 개체 인식 후 Python에서 아두이노로 급식 명령(Serial.write())을 전송한 뒤
아두이노는 정상적으로 모터/로드셀 동작을 수행함.
- 그러나 Python 터미널(콘솔)에서는 다음 로그까지만 출력되고 종료됨:
 - [SERIAL][SEND] 설기,100.0,small,3.000,1.600,3500.000,2
 - [SUMMARY] 설기.mp4 breed_confirmed=말티즈 -> name=설기
 - [RESULT] ...
 - 아두이노에서 출력하는 [CYCLE 1], [WEIGHT], 목표 사료량 도달, [DONE] 등
- 모터 작동 중 실시간 로그가 Python 콘솔에 표시되지 않음.

◆ 원인 분석

1. Python 이 너무 빨리 종료됨

- 아두이노는 급식 사이클을 수행하는 동안 계속 Serial.println()으로 로그를 전송하지만,

Python 은 명령 송신 직후 main() 함수가 종료되면서 시리얼 연결이 닫힘.

- 시리얼이 닫히면 버퍼에 남아있던 로그는 더 이상 수신되지 않음.

2. 데몬 스레드(daemon thread)의 특성

- serial_reader 가 데몬 스레드로 실행되어 있었기 때문에, 메인 스레드(main)가 종료되면 자동으로 종료됨 → 로그 수신 불가.

3. 아두이노 종료 신호 미존재

- Python 이 언제까지 기다려야 하는지 기준이 없어, 급식 완료 이전에 main 루프가 끝남.

3 개선 목표

- Python 이 아두이노로 급식 명령을 보낸 이후,
아두이노가 “[DONE]” 또는 “급식 완료” 메시지를 보낼 때까지 대기하도록 설정.
- 이 기간 동안 아두이노가 출력하는 모든 로그를 실시간으로 Python 콘솔에 표시.
- [DONE] 신호 수신 후에만 Python 이 종료되어야 함.

4 해결 방안 및 구현 내용

Arduino 측 설정

1. 급식 완료 신호 추가

2. Serial.println("[DONE]"); // 급식 완료 후 Python 종료 트리거

→ performFeeding() 함수 맨 마지막에 삽입.

→ 이 줄이 출력되어야 Python 이 안전하게 종료됨.

3. 그릇 무게 자동 보정 추가

```
4. const float BOWL_WEIGHT = 100.0; // g  
5. stableWeight -= BOWL_WEIGHT;  
6. if (stableWeight < 0) stableWeight = 0;
```

→ 로드셀 측정값에서 그릇 무게(항시 고정)를 빼줌.

Python 측 설정

1. 로그 수신 함수 추가

```
2. def wait_for_arduino_done(ser, timeout=180):  
3.     """아두이노에서 [DONE] 또는 '급식 완료'가 올 때까지 읽어서 콘솔에  
    표시"""  
4.     if not ser: return False  
5.     t0 = time.time()  
6.     while time.time() - t0 < timeout:  
7.         if ser.in_waiting:  
8.             line = ser.readline().decode(errors="ignore").strip()  
9.             if line:  
10.                 print(f"[ARDUINO] {line}")  
11.                 if "[DONE]" in line or "급식 완료" in line:  
12.                     return True  
13.                 time.sleep(0.01)  
14.     return False
```

15. 시리얼 송신 후 대기 로직 추가

```
16. send_to_arduino(ser, confirmed_name, score_pct, size, weight, activeLvl,  
    calPerKg, feedingCount)  
17. wait_for_arduino_done(ser, timeout=180)
```

→ 명령 전송 직후 Python 이 즉시 종료되지 않고
아두이노의 모든 출력 로그를 수신할 때까지 블로킹.

18. 시리얼 종료 시점 조정

```
19. if ser:  
20.     ser.close() # [DONE] 신호 수신 후에만 닫힘
```

5 최종 동작 시퀀스

단계	Python	Arduino
①	반려견 인식 (YOLO + ORB + ResNet) -	
②	품종/이름 확정 후 급식 명령 송신 (Serial.write)	Serial.readStringUntil('n')로 수신
③	-	급식량 계산 (RER/DER/portion) 출력
④	-	performFeeding() 실행 — 모터 및 로드셀 제어
⑤	wait_for_arduino_done()에서 모든 로그 수신	Serial.println()으로 [CYCLE], [WEIGHT] 등 전송
⑥	[DONE] 신호 수신 → Python 결과 요약 출력	Serial.println("[DONE]") 실행
⑦	Python 종료 및 시리얼 포트 닫힘	대기 상태 복귀