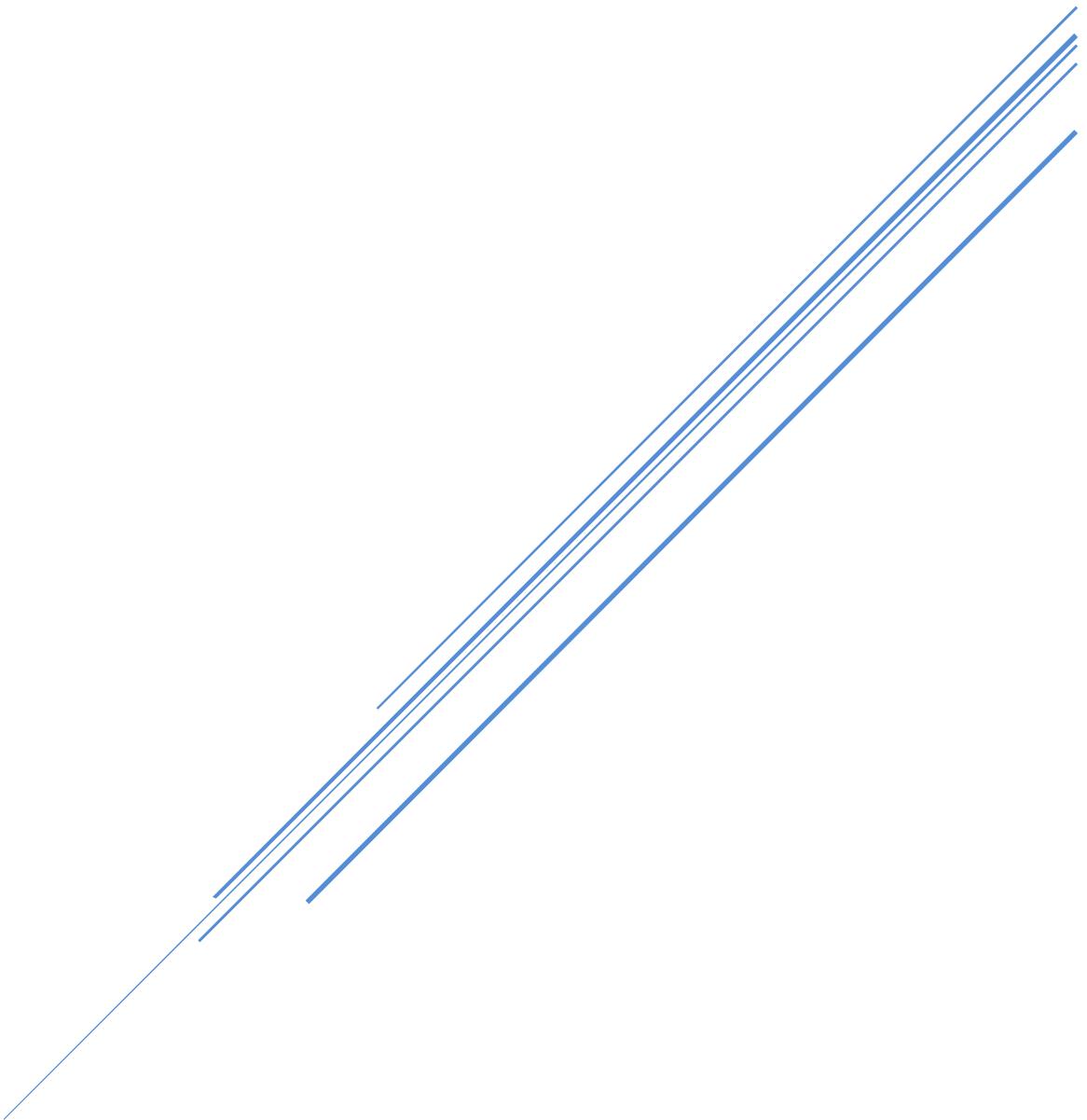


임베디드 시스템 설계 및 실험

화요일 1조 프로젝트 제안서

블루투스 기반 능동형 좌석 관리 시스템



팀 프로젝트 실행안

블루투스 기반 능동형 좌석 관리 시스템

1. 프로젝트 목적

기존 「도서관 자리비움 알리미」 아이디어를 발전시켜 도서관 열람실에서 자주 발생하는 '유령좌석(사람은 있는데 자리만 차지한 상태)' 문제를 해결하는 것을 목표로 한다. 이를 위해 여러 개의 센서를 단순 병렬이 아니라 의존성 있는 구조로 묶어 판단하고, 그 결과를 로컬(LED/LCD) + 원격(블루투스)으로 동시에 알리는 능동형 시스템을 구현한다.

본 프로젝트는 다음 5 단계 흐름으로 동작한다.

- 1 단계(감지): PIR 인체감지 센서로 사용자의 재실 여부를 1 차 판단한다.
- 2 단계(판단): 일정 시간 동안 움직임이 없으면 조도 센서 값을 추가로 읽어 짐이 놓인 상태인지 교차 판단해 센서 간 의존성을 만족시킨다.
- 3 단계(알림): 로컬(LED/LCD)과 블루투스로 현재 상태와 경과 시간을 알린다.
- 4 단계(능동 관리): 장시간 짐만 놓고 비움 상태로 판단되면서 보 모터를 작동시켜 시각적으로 알린다.
- 5 단계(수동 제어): 터치 센서(EXTI 인터럽트) 입력이 들어오면 즉시 상태를 리셋하거나 자리비움으로 전환한다.

2. 시스템 구성(하드웨어)

메인 컨트롤러 STM32F107 임베디드 보드 사용

- 입력 센서
 - PIR 인체감지 센서(구매): 좌석 점유 여부 1 차 판단
 - 조도 센서(제공): 자리비움 시 짐 스탠드 존재 여부 2 차 판단
 - 터치 센서(TTP223 등 구매): 수동 리셋/상태 변경 입력 → EXTI 인터럽트 처리
- 출력 장치
 - LED (Red/Green): 보드 내장 LD1/LD3 활용 상태(사용 중/비움) 시각화
 - TFT LCD(제공): 상세 상태 경과 시간 알림 메시지 표시
 - 서보 모터(SG90, 제공): 장기 비움 시 깃발/표식 들어 올리기
 - 블루투스 모듈(FB755AC, 제공): 상태 코드 무선 전송(USART)

3. 사용 부품 목록(BOM)

구분	부품명	역할	비고
제공	블루투스 모듈(FB755AC)	상태 원격 전송	0 원
제공	3.2" TFT LCD	상태/메시지 표시	0 원
제공	서보 모터(SG90)	능동 알림(깃발)	0 원
제공	조도 센서	2 차(점) 감지	0 원
구매	PIR 인체 감지 센서	1 차(사람) 감지	약 2,000 ~ 5,000 원
구매	터치 센서 모듈(TTP223 등)	인터럽트 입력	약 1,000 ~ 2,000 원
보유	STM32F107	메인 보드	기존 보유
보유	LED (Red/Green)	상태 표시	보드 내장 사용 OR LED 이용
		총 예상 구매액	약 3,000 ~ 7,000 원

4. 핵심 시나리오

(1) 상태 1: 사용 중(Occupied)

- PIR_Task 가 주기적으로 움직임을 감지한다.
- Status_Task 는 상태를 Occupied 로 유지하고 idle_counter 를 0 으로 리셋한다.
- Display_Task 는 LED 를 녹색으로 켜고 LCD 에 “사용 중”을 표시한다.
- Bluetooth_Task 는 상태 변화를 블루투스로 전송한다.

(2) 상태 2: 자리 비움(Vacant) + 센서 의존성

- 일정 시간(PIR 미감지) 경과 시 Status_Task 의 idle_counter 가 증가한다.

- 10 분 이상 무감지 시 Light_Sensor_Task 를 깨워 조도 값을 읽는다
- 어두움 → Vacant / 밝음 → Suspicious(침만 있음) 으로 상태를 나눈다
- 상태 변경 시 StateQ 와 블루투스로 즉시 전파한다

(3) 상태 3: 능동 관리(Active)

- Suspicious 상태가 추가 5 분 지속(총 15 분) 되면 Status_Task 가 ServoQ 에 알림을 보낸다
- Servo_Task 는 서보 모터를 $0^\circ \rightarrow 90^\circ \rightarrow 0^\circ$ 로 동작시켜 시각적으로 알린다

(4) 상태 4: 수동 리셋(Interrupt)

- 사용자가 터치 센서를 누르면 EXTI 인터럽트가 발생한다
- Touch_ISR 는 Status_Task 에 “즉시 처리” 메시지를 보낸다
- Status_Task 는 idle_counter 와 무관하게 바로 상태를 갱신한다

5. RTOS(μ C/OS-III) Task 설계

1. AppTaskStart (Prio 3): 초기화 태스크 생성 큐(PirDataQ, StateQ, ServoQ) 생성
2. PIR_Task (Prio 4): 50ms 주기 샘플링 안정화 후 PirDataQ 에 전송
3. Light_Sensor_Task (Prio 4): 대기하다가 호출 시 조도 값을 읽어 공유
4. Status_Task (Prio 5): PIR/인터럽트 이벤트를 종합해 Occupied/Vacant/Suspicious 상태 결정
StateQ 에 전파 ServoQ 에 필요 시 전송
5. Display_Task (Prio 6): StateQ 를 받아 LED 와 LCD 를 동시에 갱신
6. Bluetooth_Task (Prio 6): StateQ 를 받아 상태 코드를 블루투스로 전송
7. Servo_Task (Prio 7): ServoQ 를 받아 서보 모터를 동작
8. Touch_ISR: 터치 센서 EXTI 인터럽트, Status_Task 에 즉시 이벤트 전달

6. 전체 동작 흐름(Flow)

- AppTaskStart 가 모든 태스크와 RTOS 오브젝트를 생성한다
- PIR_Task 가 50ms 마다 감지하고 20 회 누적 후 PirDataQ 로 보낸다
- Status_Task 가 PIR 데이터를 기반으로 상태를 계산하고 필요 시 조도 센서를 깨운다
- 상태가 바뀌면 StateQ 로 전파되어 Display/Bluetooth 에서 각각 처리한다
- Suspicious 가 장기화되면 Servo_Task 가 서보를 동작시킨다

- 터치 인터럽트가 발생하면 Status_Task 가 가장 먼저 이를 처리해 상태를 갱신한다

