ui_state_usb_port.dart

소스 코드

전체 코드

```dart
import 'package:buyoil/common/app_commands.dart';
import 'package:flutter/material.dart';
import 'package:freezed_annotation/freezed_annotation.dart';
import 'package:usb_serial/usb_serial.dart';

part 'ui_state_usb_port.freezed.dart';

@freezed
sealed class UIStateUsbPort with _$UIStateUsbPort {

  const factory UIStateUsbPort.init({
    @Default([]) List<UsbDevice> availablePorts,
    UsbDevice? connectedDevice,
    UsbPort? port,
    PORT_COMMANDS? lastCommand,
    @Default(0) int resetFailCount,
  }) = UIStateUsbPortInit;

  const factory UIStateUsbPort.connected({
    @Default([]) List<UsbDevice> availablePorts,
    required UsbDevice? connectedDevice,
    required UsbPort? port,
    PORT_COMMANDS? lastCommand,
    @Default(0) int resetFailCount,
  }) = UIStateUsbPortConnected;

  const factory UIStateUsbPort.loading({
    @Default([]) List<UsbDevice> availablePorts,
    required UsbDevice? connectedDevice,
    UsbPort? port,
    PORT_COMMANDS? lastCommand,
    @Default(0) int resetFailCount,
  }) = UIStateUsbPortLoading;

  const factory UIStateUsbPort.error({
    @Default([]) List<UsbDevice> availablePorts,
    UsbDevice? connectedDevice,
    UsbPort? port,
    required String errorMsg,
    PORT_COMMANDS? lastCommand,
    @Default(0) int resetFailCount,
  }) = UIStateUsbPortError;

}
```

1️⃣ 파일 개요

파일명: `lib/model/ui_state_usb_port.dart`
역할: USB 포트 상태 관리용 데이터 클래스 정의
주요 목적:

- Flutter에서 USB 포트 연결 상태 및 동작을 **불변 객체로 관리**
- `freezed` 패키지 활용하여 **union type / sealed class** 구조 제공
- 연결 시도, 연결 완료, 오류 상태 등을 **타입 안전하게 표현**

사용 맥락 예시:

- STM32, ESP32 등 MCU와 USB 통신 관리
- USB 연결 시 포트 검색, 연결, 로딩, 오류 상태 표시
- 상태에 따라 UI, 알림, 재시도 로직 처리

2️⃣ 주요 기능

1. **초기 상태(`init`)**
    - USB 연결 전 초기 상태
    - 연결 가능한 포트 목록 조회 가능
    - 기본값: `availablePorts = []`, `connectedDevice = null`, `port = null`
2. **연결 완료 상태(`connected`)**
    - USB 포트 연결 성공 시
    - 연결된 `UsbDevice`와 `UsbPort` 참조 유지
    - 마지막 명령(`lastCommand`)과 리셋 실패 횟수(`resetFailCount`) 관리
3. **로딩 상태(`loading`)**
    - 연결 시도 또는 명령 전송 중
    - UI에서 로딩 인디케이터 표시 가능
    - 현재 연결 상태와 포트 정보 유지
4. **오류 상태(`error`)**
    - 연결 실패, 포트 접근 오류 등 발생 시
    - 오류 메시지(`errorMsg`) 제공
    - 포트 상태는 유지, 필요 시 재시도 가능
5. **공통 필드**

필드	타입	설명
`availablePorts`	`List<UsbDevice>`	연결 가능한 USB 포트 목록
`connectedDevice`	`UsbDevice?`	현재 연결된 디바이스
`port`	`UsbPort?`	연결된 USB 포트 객체

필드	타입	설명
`lastCommand`	`PORT_COMMANDS?`	마지막 전송 명령 기록
`resetFailCount`	`int`	연결 재시도 실패 횟수

3 구조 분석

```
UIStateUsbPort (sealed class)
 ├─ UIStateUsbPortInit       : 초기 상태, 포트 검색 가능
 ├─ UIStateUsbPortConnected  : 연결 성공 상태
 ├─ UIStateUsbPortLoading    : 명령/연결 진행 중 상태
 └─ UIStateUsbPortError      : 오류 발생 상태, errorMsg 포함
```

특징:

- USB 연결 관리 로직에서 상태 기반 분기 용이
- `freezed`를 통해 **패턴 매칭** (`when` / `maybeWhen` / `map`) 가능
- 상태 불변성 유지, 안전한 ViewModel 연동

4 동작 흐름

1. 앱 실행 시 `UIStateUsbPort.init` 상태 생성 → `availablePorts` 조회
2. 사용자가 연결 시도 → 상태를 `loading`으로 전환
3. 연결 성공 시 `connected` 상태로 전환 → `connectedDevice`와 `port` 업데이트
4. 연결 실패 시 `error` 상태로 전환 → `errorMsg` 표시
5. 연결 중 마지막 명령(`lastCommand`)과 재시도 횟수(`resetFailCount`) 기록 가능
6. 필요 시 상태 변경 시 UI 리빌드

5 장점

- **상태 기반 USB 관리**: 연결, 로딩, 오류 구분 명확
- **타입 안전성**: union type으로 잘못된 상태 접근 방지
- **확장성**: 새로운 상태나 필드 추가 용이
- **ViewModel 연계 용이**: `StateNotifier`나 `Riverpod`에서 사용 가능

6 단점 / 개선점

1. **연결 상태 상세 정보 부족**
    - 전송 속도, 연결 시간, 포트 속성 등 부가 정보 없음
    - 개선: `baudRate`, `manufacturerName` 등 필드 추가 가능

2. **동기/비동기 처리 로직 분리 필요**
    - 상태 관리만 제공, 실제 포트 열기/명령 전송 로직은 별도 구현 필요
3. **Error 상태 필드 통일성**
    - `errorMsg` 외에 `errorCode` 등 추가하면 디버깅 용이

7️⃣ 개선 구조 예시 (개념)

```
@freezed
sealed class UIStateUsbPort with _$UIStateUsbPort {
  const factory UIStateUsbPort.init({
    @Default([]) List<UsbDevice> availablePorts,
  }) = UIStateUsbPortInit;

  const factory UIStateUsbPort.connected({
    required UsbDevice connectedDevice,
    required UsbPort port,
    PORT_COMMANDS? lastCommand,
  }) = UIStateUsbPortConnected;

  const factory UIStateUsbPort.loading({
    required UsbDevice? connectedDevice,
    UsbPort? port,
    PORT_COMMANDS? lastCommand,
  }) = UIStateUsbPortLoading;

  const factory UIStateUsbPort.error({
    required String errorMsg,
    UsbDevice? connectedDevice,
    UsbPort? port,
    PORT_COMMANDS? lastCommand,
    @Default(0) int resetFailCount,
  }) = UIStateUsbPortError;
}
```

- **장점:** 상태 필드 최소화, 각 상태별 필드 명확화
- **단점:** 공통 필드 일부 반복 필요