

vm_step2.dart

소스 코드

전체 코드

```
1 import 'package:buyoil/viewmodel/vm_serial_port.dart';
2 import 'package:riverpod_annotation/riverpod_annotation.dart';
3
4 import '../model/ui_state_step2.dart';
5
6 part 'vm_step2.g.dart';
7
8 @riverpod
9 class Step2 extends _$Step2 {
10   @override
11   UIStateStep2 build() {
12     return UIStateStep2.init();
13   }
14
15   pressedOpen() async {
16     ref.watch(serialPortVMProvider.notifier).open();
17   }
18 }
```

1 파일 개요

클래스: `Step2 extends _$Step2`

역할:

- Step2 화면에서 **문 열기(Open)** 동작 상태 관리
- Riverpod 상태 관리 (`@riverpod`)
- USB 통신과 직접 연결 (`SerialPortVM.open()` 호출)

2 상태 정의

모델: `UIStateStep2`

- Step2 화면에서 필요한 최소 상태 관리
- 초기 상태: `UIStateStep2.init()`
- 현재 코드는 **상태 변경 로직은 없고**, 단순히 Open 명령 전달 기능만 있음

3 주요 기능

함수	설명
<code>build()</code>	초기 상태 생성 (<code>UIStateStep2.init()</code>)
<code>pressedOpen()</code>	Open 버튼 클릭 시 호출:
	- <code>SerialPortVM.open()</code> 호출 → STM32 / USB 포트에 <code>[CMD]OPEN#</code> 전송
	- USB 응답 처리 및 화면 전환은 <code>SerialPortVM.listenByPort()</code> 에서 담당

4 특징

- Step2는 **UI 동작** → **USB 명령 전송** → **Step3 이동** 흐름에 집중
- 상태 변화는 현재 최소화되어 있음
- SerialPortVM과의 연동으로 **Open 명령 전송/응답 처리** 일원화

5 개선/확장 아이디어

- Open 명령 전송 후 **타임아웃 처리** (현재 SerialPortVM에서 재시도 로직 존재)
- UI에서 **로딩/버튼 비활성화** 상태 관리 가능
- 향후 Step2와 Step3 사이 **센서/문 상태 확인** 로직 추가 가능