

vm_step3.dart

소스 코드

전체 코드

```
1 import 'package:buyoil/model/ui_state_step3.dart';
2 import 'package:buyoil/viewmodel/vm_serial_port.dart';
3 import 'package:riverpod_annotation/riverpod_annotation.dart';
4
5 part 'vm_step3.g.dart';
6
7 @riverpod
8 class Step3 extends _$Step3 {
9   @override
10   UIStateStep3 build() {
11     return UIStateStep3.init();
12   }
13
14   void pressedClose() {
15     ref.watch(serialPortVMProvider.notifier).close();
16     // // todo ok 처리
17     // state = UIStateStep3.closeDoor();
18     //
19     // Future.delayed(Duration(seconds: 2), () {
20     //   state = UIStateStep3.completed();
21     // });
22   }
23 }
```

1 파일 개요

클래스: `Step3 extends _$Step3`

역할:

- Step3 화면에서 문 닫기(Close) 동작 상태 관리
- Riverpod 상태 관리 (@riverpod)
- USB 통신과 직접 연결 (`SerialPortVM.close()` 호출)

2 상태 정의

모델: `UIStateStep3`

- Step3 화면에서 필요한 상태 관리
- 초기 상태: `UIStateStep3.init()`
- 현재 코드는 상태 변경 로직은 주석 처리되어 있음 → 실제 상태 변화는 USB 응답 처리 후 구현 예정

3 주요 기능

함수	설명
<code>build()</code>	초기 상태 생성 (<code>UIStateStep3.init()</code>)
<code>pressedClose()</code>	Close 버튼 클릭 시 호출:
	- <code>SerialPortVM.close()</code> 호출 → STM32 / USB 포트에 <code>[CMD]CLOSE#</code> 전송
	- 현재 OK 응답 처리 및 상태 변경은 주석 처리 (<code>closeDoor</code> , <code>completed</code>)

4 특징

- Step3는 UI 동작 → USB 명령 전송 → 상태/화면 전환 흐름에 집중
- 실제 상태 변화 및 화면 이동은 `SerialPortVM`의 응답 처리 또는 추후 구현될 `UIStateStep3` 상태 전환 로직에서 담당

5 개선/확장 아이디어

- Close 명령 전송 후 OK 응답 확인 → Step 완료 상태로 전환
- UI에서 로딩/버튼 비활성화 상태 관리 가능
- 실패 시 재시도 로직 또는 알림/팝업 추가 가능