

# btn\_home.dart

## 소스 코드

### 전체 코드

```
1 import 'package:buyoil/viewmodel/vm_serial_port.dart';
2 import 'package:flutter/cupertino.dart';
3 import 'package:flutter/material.dart';
4 import 'package:flutter_riverpod/flutter_riverpod.dart';
5 import 'package:go_router/go_router.dart';
6
7 import '../common/app_strings.dart';
8 import '../router.dart';
9
10 class ToHomeBtn extends ConsumerWidget {
11   const ToHomeBtn({super.key});
12
13   @override
14   widget build(BuildContext context, ref) {
15     return GestureDetector(
16       behavior: HitTestBehavior.opaque,
17       onTap: () {
18         ref.watch(serialPortVMProvider.notifier).goToSplash().whenComplete(() {
19           context.goNamed(RouteGroup.Splash.name);
20         });
21       },
22       child: Container(
23         width: 99,
24         height: 49,
25         margin: EdgeInsets.symmetric(vertical: 15, horizontal: 13),
26         child: Image.asset("${AppStrings.assetPath}img_home.png"),
27       ),
28     );
29   }
30
31 }
```

## 1. 구조 분석

ToHomeBtn 은 다음 기능을 제공하는 단일-purpose UI 컴포넌트이다.

- Riverpod 의 serialPortVMProvider.notifier 를 호출하여 시리얼 포트 관련 초기화 또는 종료 처리 실행
- GestureDetector 를 사용하여 터치 이벤트를 수신
- Splash 화면으로 라우팅( go\_router )
- 정해진 크기(99×49)의 홈 버튼 이미지 렌더링

구조적 요소:

```

1 ConsumerWidget
2   └─ GestureDetector
3       └─ Container
4           └─ Image.asset(img_home)

```

## 2. 세부 동작 흐름

### 1. build(context, ref)

`ConsumerWidget`의 표준 시그니처 (`build(BuildContext context, WidgetRef ref)`) 형태가 맞음)

### 2. onTap → serialPortVMProvider.notifier.goToSplash() 호출

시리얼 포트 VM에서 splash 진입을 위한 백엔드 상태 정리 수행.

### 3. whenComplete → context.goNamed(Splash)

VM 비동기 작업 완료 이후 Splash 화면으로 이동.

### 4. UI는 단순한 image 기반의 버튼.

## 3. 문제점 및 개선 포인트

### ✓ (1) GestureDetector보다 InkWell/InkResponse 사용이 더 적절

현재는 **Material ripple 효과 없음**, 접근성(A11Y)도 부족하다.

### ✓ (2) ConsumerWidget의 build 시그니처 오류

현재 코드:

```

1 @override
2 widget build(BuildContext context, ref)

```

올바른 형태:

```

1 @override
2 widget build(BuildContext context, WidgetRef ref)

```

IDE에서 경고가 없었던 이유는 dynamic 으로 처리됐기 때문.

### ✓ (3) ref.watch → ref.read 로 변경 필요

행위 함수 호출이므로 `watch` 사용 시 rebuild 불필요하게 발생함.

잘못된 형태:

```

1 ref.watch(serialPortVMProvider.notifier).goToSplash()

```

권장:

```

1 ref.read(serialPortVMProvider.notifier).goToSplash()

```

## ✓ (4) Container + Image만 있는 단순 버튼 → StatelessWidget로 충분

추가 상태 없음.

## 4. 정제된 공식문서 스타일 리팩토링 코드

```
1 import 'package:buyoil/viewmodel/vm_serial_port.dart';
2 import 'package:flutter/material.dart';
3 import 'package:flutter_riverpod/flutter_riverpod.dart';
4 import 'package:go_router/go_router.dart';
5
6 import '../common/app_strings.dart';
7 import '../router.dart';
8
9 class ToHomeBtn extends ConsumerWidget {
10   const ToHomeBtn({super.key});
11
12   @override
13   Widget build(BuildContext context, WidgetRef ref) {
14     return Material(
15       color: Colors.transparent,
16       child: Inkwell(
17         onTap: () {
18           ref
19             .read(serialPortVMProvider.notifier)
20             .goToSplash()
21             .whenComplete(() {
22               context.goNamed(RouteGroup.Splash.name);
23             });
24         },
25       borderRadius: const BorderRadius.all(Radius.circular(4)),
26       child: Container(
27         width: 99,
28         height: 49,
29         margin: const EdgeInsets.symmetric(vertical: 15, horizontal: 13),
30         child: Image.asset(
31           "${AppStrings.assetPath}img_home.png",
32           fit: BoxFit.contain,
33         ),
34       ),
35     );
36   }
37 }
38 }
```

## 5. 리팩토링 특징 요약

- `Inkwell`로 교체하여 Material ripple 효과 제공
- build 시그니처 수정 (`WidgetRef ref`)

- `ref.watch` → `ref.read` 로 변경
- `borderRadius` 제공으로 터치 영역 시각적 안정성 확보
- 스타일, 구조, 동작 방식은 기존과 동일하게 유지