

# 1. Was man auf dem Foto eindeutig erkennt

## Arduino-Pins

- **Pin 2** → LED (grün)
- **Pin 3** → linker Taster
- **Pin 4** → rechter Taster
- **GND** ist gemeinsam geführt
- **5 V** liegt am Breadboard an

## Tasterbeschaltung

- Jeder Taster hat **einen Widerstand nach GND**
- Beim Drücken wird der jeweilige Pin auf **HIGH (5 V)** gezogen → **klassische Pull-Down-Schaltung**, völlig korrekt, oldschool, lehrbuchsauber ☐

☐ **Wichtig:** Das bedeutet:

- **nicht gedrückt = LOW**
- **gedrückt = HIGH**

Dein aktueller Code wertet **nur Pin 3** aus. Pin 4 existiert elektrisch, aber **softwareseitig noch gar nicht**.

# 2. Warum der vorhandene Code die Aufgabe NICHT erfüllt

Aktuell:

[snippet.cpp](#)

```
if (digitalRead(3) == HIGH)
    LED an
else
    LED aus
```

Das ist ein **Taster = Momentkontakt**. Sobald man loslässt → LED wieder aus.

Die Aufgabe verlangt aber:

- ☐ linker Taster → **einschalten**
- ☐ rechter Taster → **ausschalten**
- ☐ LED merkt sich den Zustand

Also: **kein else**, sondern **zwei unabhängige Entscheidungen**.

### 3. Minimal korrekte Lösung (exakt zur Aufgabe passend)

Kein Schnickschnack, keine Entprell-Orgie, **didaktisch sauber**:

snippet.cpp

```
void setup()
{
    pinMode(2, OUTPUT);    // LED
    pinMode(3, INPUT);    // linker Taster (EIN)
    pinMode(4, INPUT);    // rechter Taster (AUS)
}

void loop()
{
    if (digitalRead(3) == HIGH)    // linker Taster gedrückt
    {
        digitalWrite(2, HIGH);    // LED EIN
    }

    if (digitalRead(4) == HIGH)    // rechter Taster gedrückt
    {
        digitalWrite(2, LOW);    // LED AUS
    }
}
```

#### Warum das korrekt ist

- Kein `else` → Zustand bleibt erhalten
- Jeder Taster hat **eine klare Aufgabe**
- Passt **1:1 zur Verdrahtung**
- Entspricht exakt dem Aufgabentext

### 4. Typischer Denkfehler (konstruktive Kritik)

Viele machen hier reflexartig:

snippet.cpp

```
if (Taste1) LED an;
else LED aus;
```

Das ist **falsch**, weil:

- **else** bedeutet „**alle anderen Fälle**“
- Du willst aber **zwei explizite Aktionen**, keine Gegenspieler

Merksatz für die Prüfung:

**Schalten ≠ Gedrückt halten**

## 5. Wenn der Lehrer pingelig ist (Bonuswissen)

- Entprellen ist **nicht gefordert**
- INPUT\_PULLUP ist **nicht erlaubt**, weil ihr Widerstände verbaut habt
- Globale Variablen sind **nicht nötig**
- Der Code ist bewusst simpel gehalten → **prüfungsfreundlich**

From:

<http://wiki.nctl.de/dokuwiki/> - □ **Veni. Vidi. sudo rm -rf / vici.**

Permanent link:

<http://wiki.nctl.de/dokuwiki/doku.php?id=allgemein:arduino&rev=1768219323>

Last update: **12.01.2026 13:02**

