

## Sinking Vs Sourcing



#### A) LED gegen VCC (empfohlen)

Pin  $\Rightarrow$  LED  $\Rightarrow$  Widerstand  $\Rightarrow$  GND

### B) LED gegen Masse (nicht optimal)

VCC  $\Rightarrow$  Widerstand  $\Rightarrow$  LED  $\Rightarrow$  Pin



### **Der entscheidende Punkt: Sinking vs. Sourcing**

Begriff	Bedeutung
<b>Sourcing</b>	Pin liefert Strom (Pin = HIGH)
<b>Sinking</b>	Pin zieht Strom nach Masse (Pin = LOW)



**Arduino-Pins können Strom besser „sinken“ als „sourcen“**



## Strombelastbarkeit (ATmega328P / Arduino Uno)

- Max. pro Pin: ca. 20 mA
  - Empfohlen:  $\leq 10$  mA
  - Intern:
    - Transistor nach GND ist stärker
    - Transistor nach VCC ist schwächer



**LED gegen Masse = stabiler Pegel + weniger Stress für den Pin**



### Vorteile von LEDs gegen Masse

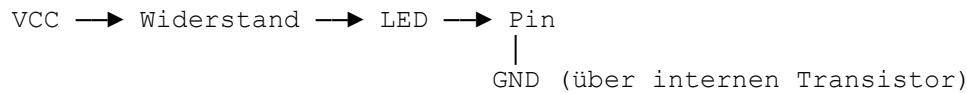
- ✓ bessere Stromaufnahme
  - ✓ geringerer Spannungsabfall im Pin
  - ✓ sauberere HIGH-Pegel
  - ✓ Standard in Datenblättern
  - ✓ identisch zu Industrie-Schaltungen

### **Merksatz:**

„Pins mögen es lieber, Strom zu schlucken als zu liefern.“

## **Korrekte Schaltung: LED gegen Masse schalten**

### **RICHTIG (Low-Side / Strom sinken)**



- ◆ **Pin = LOW → LED AN**
- ◆ **Pin = HIGH → LED AUS**

- **Der Pin zieht Strom nach Masse**
  - **Das nennt man SINKING**
  - **Genau das ist strommäÙig besser**
- 

### **Alternative (High-Side / Strom liefern)**

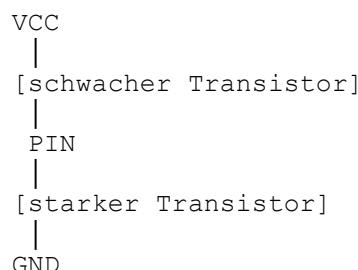
Pin --> LED --> Widerstand --> GND

- ◆ **Pin = HIGH → LED AN**
- ◆ **Pin = LOW → LED AUS**

- **Der Pin muss Strom liefern (SOURCING)**
  - **Elektrisch ungünstiger**
- 

## **3 Warum SINKING strommäßig besser ist als SOURCING**

### **Intern im Arduino-Pin (vereinfacht)**



→ Transistor nach GND ist stärker

---

⚡ Praktische Folgen

Eigenschaft	Sinking (gegen GND)	Sourcing (gegen VCC)
Max. Strom	höher / stabiler	geringer
Spannungsabfall	kleiner	größer
Verlustleistung im Pin	geringer	höher
Pegelstabilität	besser	schlechter

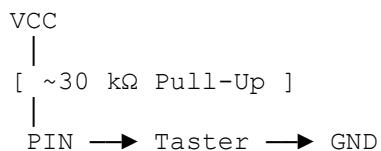
- LEDs leuchten **gleichmäßiger**
  - Weniger Stress für den µC
  - Sauberer LOW-Pegel
-

## 4 Jetzt der Bogen zu den Tastern (interner Pull-Up)

### ○ Taster-Schaltung (Standard!)

```
Pin → Taster → GND  
pinMode(pin, INPUT_PULLUP);
```

Intern:



### Zustandstabelle

#### Taster Pin

offen	HIGH
gedrückt	LOW

- Auch hier zieht der Pin nach GND
  - Gleiches Prinzip wie bei der LED
- 

## 5 Warum Embedded-Designer das lieben

- ✓ gleiche Logik für LEDs, Taster, Optokoppler, Relais
- ✓ LOW = „aktiv“ → failsafe
- ✓ weniger Störungen
- ✓ Standard in Datenblättern
- ✓ funktioniert perfekt mit internen Pull-Ups

#### 💡 Industrie-Merksatz:

„Signale sind aktiv LOW, geschaltet wird nach Masse.“

---

## 6 Kurz & korrekt zusammengefasst

### ✓ LED (empfohlen)

VCC → Widerstand → LED → Pin  
Pin = LOW → LED AN

## Taster (empfohlen)

Pin → Taster → GND  
INPUT\_PULLUP aktiv

- Pin arbeitet immer als Stromsenke
- Das ist elektrisch robuster