

Arduino Klingel/Toröffner mit Optokoppler

<https://gemini.google.com/app/eead163325380f85>

User prompt: Ich habe einen Arduino Nano mit dem ich den Tastendruck auf eine Türklingel und einen Toröffner detektieren will. An der Türklingel hängt direkt beim Taster noch ein Vibrationsmotor im selben Stromkreis, der haptisches Feedback liefert, wenn man den Klingelknopf drückt. Der Vibrationsmotor ist ein "HELTC 18000 rpm 612 3.7-5 V DC, rated current: 0.09 A". Durch das sehr lange Kabel (> 10 m) und vor allem den Vibrationsmotor fällt viel Spannung ab. Außerdem wird der Schaltkreis teilweise fälschlicherweise als geschlossen detektiert (Flares), vermutlich weil das lange Kabel wie eine Antenne wirkt. Ich möchte ein Schaltung aufbauen, die folgende Anforderungen erfüllt: - Die Flares sollen abgefangen werden. Das kann entweder durch die Schaltung selbst oder im Arduino-Code passieren. Ich habe ganz gute Erfahrungen mit einem Threshold von 200 ms gemacht. Alle Impulse darunter werden als Flare verworfen. - Für den Betrieb des Vibrationsmotors und als Ausgleich für das lange Kabel möchte ich die Schalter an den 5V-Ausgang des Arduino hängen. Macht es Sinn, den Eingang vom Signal durch einen Optokoppler zu trennen? Ich habe alle benötigten Bauteile wie PC817 oder 4N25 Optokoppler, Transistoren und Widerstände zur Verfügung. Das Eingangssignal der Klingel soll von PIN D2 gemessen werden, das Signal vom Toröffner von PIN D4. Bitte gib mir einen Vorschlag für einen zweckmäßigen, nicht unnötig komplexen Aufbau.

Response: Das ist ein klassisches Problem bei langen Leitungen im Außenbereich. Die Kombination aus der Antennenwirkung des 10-Meter-Kabels (Elektromagnetische Störungen/EMI) und dem **Vibrationsmotor als induktiver Last** ist quasi der perfekte Sturm für Fehlauslösungen und Spannungseinbrüche.

Kurze Antwort auf deine Frage: **Ja, ein Optokoppler macht absolut Sinn – und zwar doppelt!**

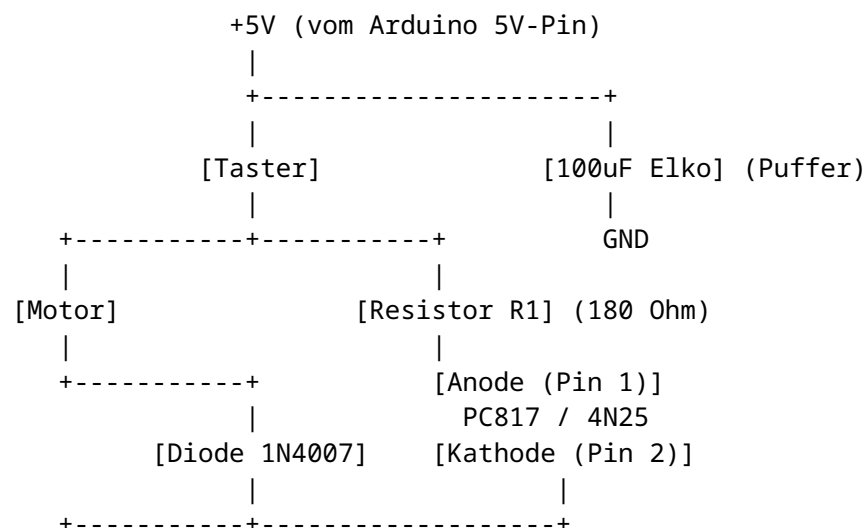
1. Er **entkoppelt** deine Mikrocontroller-Logik galvanisch von der Störquelle.
2. Der Optokoppler benötigt einen echten **Strom** (einige Milliampere), um die interne LED zu leuchten. Reine Spannungseinstrahlungen (die deine Antennen-Leitung als High-Impedanz-Glitches einfängt) haben nicht genug Energie, um den Optokoppler durchzuschalten. Das eliminiert bereits den Großteil der Flares auf Hardware-Ebene.

Zusätzlich schützt der Optokoppler deinen Arduino vor Induktionsspitzen (Back-EMF), wenn der Vibrationsmotor abschaltet.

1. Der Schaltplan (Pro Kanal)

Da du 5 Volt vom Arduino nutzen möchtest und der Motor recht viel Strom zieht (90 mA bei 10 Meter Kabel führt schnell zu Spannungseinbrüchen am Arduino, wenn du es falsch verdrahtest), teilen wir das Ganze sauber auf.

Hier ist der praxisnahe, robuste Aufbau für **einen** Kanal (z. B. Klingel an D2). Für das Tor (D4) nutzt du exakt dieselbe Schaltung, kannst dort aber ggf. die Freilaufdiode weglassen, wenn kein Motor dranhängt.



|
GND (Masse des Signalkreises)

----- GALVANISCHE TRENNUNG -----

```
      +5V (Arduino)
      |
      [R2: 10k Ohm]
      |
      +-----> An Arduino-Pin (D2)
      |
[Kollektor (Pin 4)]
  PC817 / 4N25
[Emitter (Pin 3)]
  |
  GND (Arduino)
```

2. Warum dieser Aufbau funktioniert & wichtige Details

1. Die Freilaufdiode (1N4007 / 1N4148) über dem Motor:

Sehr wichtig! Wenn der Taster losgelassen wird, induziert die Spule des Motors eine negative Spannungsspitze von oft mehreren Dutzend Volt. Eine Diode parallel zum Motor (Kathode an +5V nach dem Taster, Anode an GND) löscht diese Spitze sofort. Ohne Diode stirbt dir der Optokoppler über Kurz oder Lang.

2. Dimensionierung des Vorwiderstands (R1):

- Die Sende-LED im PC817/4N25 benötigt ca. 1,2 V und rund 10 – 15 mA Strom.
- Bei 5 V Versorgung und langem Kabel (Rechnen wir vorsichtig mit ca. 4 V effektiver Spannung am Taster durch Abfall):

$$R = \frac{4 \text{ V} - 1,2 \text{ V}}{0,015 \text{ A}} \approx 180 - 220 \text{ Ohm}$$

- Nimm einfach einen **180 Ω** oder **220 Ω** Widerstand vor Pin 1 des Optokopplers.

3. Invertierte Logik am Arduino (Pull-Up):

- Der Kollektor (Pin 4) geht an D2, der Emitter (Pin 3) an GND.
- Mit dem 10k Ω Pull-Up-Widerstand liest der Arduino im Normalzustand **HIGH**.
- Wird geklingelt, schaltet der Optokoppler durch und zieht D2 auf **LOW**.

4. Kondensator (Puffer) bei Spannungseinbrüchen:

- Ein kleiner Elektrolytkondensator (100 μF) direkt am Ausgang des 5V-Pins gegen GND hilft, die Anlaufstromspitze des Motors abzufangen, damit der Arduino Nano nicht neustartet (Reset).

3. Entprellung & Software-Debounce (Die 200 ms Regel)

Obwohl der Optokoppler Einstrahlungen filtert, kann der Taster mechanisch prellen oder der Motor Vibrationen zurück in den Taster leiten.

Da du gute Erfahrungen mit der 200ms-Schwelle gemacht hast, setzt du das am besten **ohne delay()** über

millis() um, damit der Arduino blockierungsfrei bleibt.

```
const int pinKlingel = 2; // PC817 Kollektor an D2
const int pinTor      = 4; // PC817 Kollektor an D4

const unsigned long DEBOUNCE_DELAY = 200; // 200 ms Threshold für Schaltsignal

// Variablen für Klingel
unsigned long lastTriggerKlingel = 0;
bool lastStateKlingel = HIGH;

void setup() {
  pinMode(pinKlingel, INPUT); // Externer 10k Pull-Up verbaut
  pinMode(pinTor, INPUT);
  Serial.begin(9600);
}

void loop() {
  unsigned long currentMillis = millis();

  // --- KLINGEL EVALUIEREN ---
  int currentStateKlingel = digitalRead(pinKlingel);

  // Signal ist Aktiv-LOW (Optokoppler zieht auf Masse)
  if (currentStateKlingel == LOW && lastStateKlingel == HIGH) {
    // Flanke nach unten erkannt - Prüfen ob Threshold erreicht ist
    if (currentMillis - lastTriggerKlingel > DEBOUNCE_DELAY) {
      lastTriggerKlingel = currentMillis;

      // HAUPTAKTION KLINGEL
      Serial.println("Klingel betätigt! (Valider Impuls)");
    }
  }
  lastStateKlingel = currentStateKlingel;

  // --- TORÖFFNER EVALUIEREN (Analog) ---
  // ... gleiche Logik für Pin D4
}
```

Zusammenfassung der Bauteile pro Kanal:

- **1x Optokoppler** (PC817 oder 4N25)
- **1x Vorwiderstand LED** (180 Ω oder 220 Ω)
- **1x Pull-Up Widerstand** (10 k Ω)
- **1x Freilaufdiode** (z. B. 1N4007 am Motor)
- Optional: **1x 100 μ F Elko** am 5V-Ausgang des Arduino.

User prompt: Ich kann R1 und den Optokoppler nicht zwischen Taster und Motor schalten. Taster und Motor sind gemeinsam in einem Gehäuse an der Tür, die Schaltung ist am anderen Ende des Kabels, über 10 m entfernt. Außerdem ist mir nicht ganz klar, welchen Zweck der 100uF Elko erfüllt. Solange der Taster offen ist, kommt doch ohnehin kein Signal dahinter an, oder? Auch der Zweck der Diode ist mir nicht ganz klar. Was hältst du von dieser vereinfachten Variante? Ich habe einen Arduino Nano mit dem ich den Tastendruck auf eine Türklingel und einen

Toröffner detektieren will. An der Türklingel hängt direkt beim Taster noch ein Vibrationsmotor im selben Stromkreis, der haptisches Feedback liefert, wenn man den Klingelknopf drückt. Der Vibrationsmotor ist ein "HELTC 18000 rpm 612 3.7-5 V DC, rated current: 0.09 A". Durch das sehr lange Kabel (> 10 m) und vor allem den Vibrationsmotor fällt viel Spannung ab. Außerdem wird der Schaltkreis teilweise fälschlicherweise als geschlossen detektiert (Flares), vermutlich weil das lange Kabel wie eine Antenne wirkt. Ich möchte ein Schaltung aufbauen, die folgende Anforderungen erfüllt: - Die Flares sollen abgefangen werden. Das kann entweder durch die Schaltung selbst oder im Arduino-Code passieren. Ich habe ganz gute Erfahrungen mit einem Threshold von 200 ms gemacht. Alle Impulse darunter werden als Flare verworfen. - Für den Betrieb des Vibrationsmotors und als Ausgleich für das lange Kabel möchte ich die Schalter an den 5V-Ausgang des Arduino hängen. Macht es Sinn, den Eingang vom Signal durch einen Optokoppler zu trennen? Ich habe alle benötigten Bauteile wie PC817 oder 4N25 Optokoppler, Transistoren und Widerstände zur Verfügung. Das Eingangssignal der Klingel soll von PIN D2 gemessen werden, das Signal vom Toröffner von PIN D4. Bitte gib mir einen Vorschlag für einen zweckmäßigen, nicht unnötig komplexen Aufbau. "" Ich habe einen Arduino Nano mit dem ich den Tastendruck auf eine Türklingel und einen Toröffner detektieren will. An der Türklingel hängt direkt beim Taster noch ein Vibrationsmotor im selben Stromkreis, der haptisches Feedback liefert, wenn man den Klingelknopf drückt. Der Vibrationsmotor ist ein "HELTC 18000 rpm 612 3.7-5 V DC, rated current: 0.09 A". Durch das sehr lange Kabel (> 10 m) und vor allem den Vibrationsmotor fällt viel Spannung ab. Außerdem wird der Schaltkreis teilweise fälschlicherweise als geschlossen detektiert (Flares), vermutlich weil das lange Kabel wie eine Antenne wirkt. Ich möchte ein Schaltung aufbauen, die folgende Anforderungen erfüllt: - Die Flares sollen abgefangen werden. Das kann entweder durch die Schaltung selbst oder im Arduino-Code passieren. Ich habe ganz gute Erfahrungen mit einem Threshold von 200 ms gemacht. Alle Impulse darunter werden als Flare verworfen. - Für den Betrieb des Vibrationsmotors und als Ausgleich für das lange Kabel möchte ich die Schalter an den 5V-Ausgang des Arduino hängen. Macht es Sinn, den Eingang vom Signal durch einen Optokoppler zu trennen? Ich habe alle benötigten Bauteile wie PC817 oder 4N25 Optokoppler, Transistoren und Widerstände zur Verfügung. Das Eingangssignal der Klingel soll von PIN D2 gemessen werden, das Signal vom Toröffner von PIN D4. Bitte gib mir einen Vorschlag für einen zweckmäßigen, nicht unnötig komplexen Aufbau.

	+5V (vom Arduino 5V-Pin)	[Motor]		
[Taster]		(langes Kabel)		[Resistor R1] (180 Ohm)
(Pin 1)]	PC817 / 4N25	[Kathode (Pin 2)]		GND ----- GALVANISCHE
TRENNUNG -----	+5V (Arduino)	[R2: 10k Ohm]		+-----> An Arduino-
Pin (D2)	[Kollektor (Pin 4)]	PC817 / 4N25	[Emitter (Pin 3)]	GND (Arduino) ""

Response: Danke für die Klarstellung zur räumlichen Trennung! Das vereinfacht das Verständnis extrem.

Deine vereinfachte Schaltung zeigt genau das richtige Grundprinzip, hat aber **zwei kritische Schwachstellen**, die wir leicht korrigieren können. Gehen wir zuerst kurz auf deine Fragen ein, damit klar ist, *warum* wir die Schaltung etwas anpassen müssen.

Zu deinen Fragen: Warum Elko und Diode?

1. **Wofür ist die Freilaufdiode?** Ein Elektromotor ist im Grunde eine große Induktivität (Spule). Wenn du den Taster loslässt, bricht das Magnetfeld im Motor schlagartig zusammen. Dabei entsteht eine kurzzeitige, **negative Spannungsspitze** (Back-EMF) von oft 20 bis 50 Volt.
 - **Die Gefahr in deiner Skizze:** Diese negative Spannung schießt ungehindert über das 10-Meter-Kabel direkt auf die LED im Optokoppler. Die Sende-LEDs in Optokopplern wie dem PC817 vertragen typischerweise **maximal 6 Volt in Sperrrichtung**. Ohne Diode brennt dir der Optokoppler nach wenigen Betätigungen durch.
2. **Wofür war der 100 µF Elko?** Ein Gleichstrommotor zieht beim Anlaufen für ein paar Millisekunden deutlich mehr Strom als im Dauerbetrieb (oft das 2- bis 3-Fache der 90 mA). Wenn der 5V-Pin des Arduino Nano diesen plötzlichen Stromimpuls liefern muss, kann die 5V-Schiene des Arduinos kurz einbrechen. Fällt sie unter $\approx 4,3$ V, macht der Nano einen ungewollten **Reset**. Der Kondensator puffert diesen ersten Einschaltmoment ab.

Das Problem an deinem Entwurf (Reihenschaltung)

In deiner Skizze liegen **Motor, R1 (180 Ω) und die Optokoppler-LED alle hintereinander in Reihe**.

Das hat zwei große Haken:

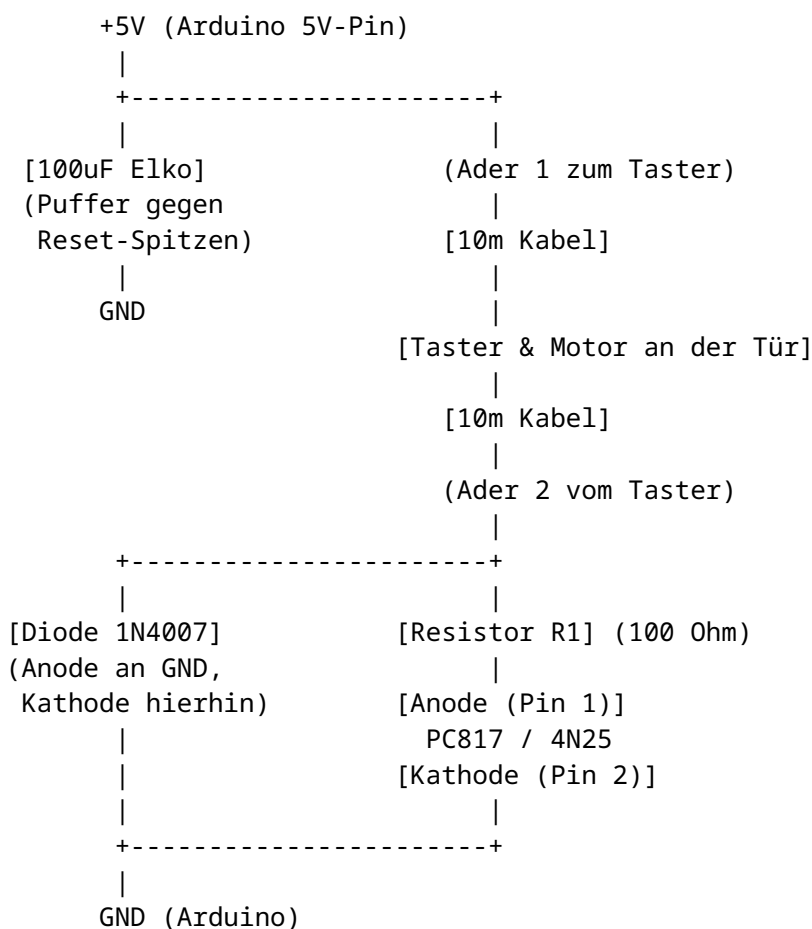
1. **Der Motor bekommt viel zu wenig Spannung:** An der LED des Optokopplers fallen bereits $\approx 1,2\text{ V}$ ab. Über den $180\ \Omega$ -Widerstand und 90 mA Strom fallen weitere $\approx 16\text{ V}$ ab (bzw. brennt der Strom komplett ein). Der Motor wird sich kaum oder gar nicht drehen.
2. **Die LED bekommt zu wenig Strom:** Vergrößerst du R1 nicht, zieht der Motor den Strom herunter, oder der Motor begrenzt den Strom so sehr, dass der Optokoppler nicht sauber durchschaltet.

Der korrigierte & optimale Aufbau (Parallel schalten!)

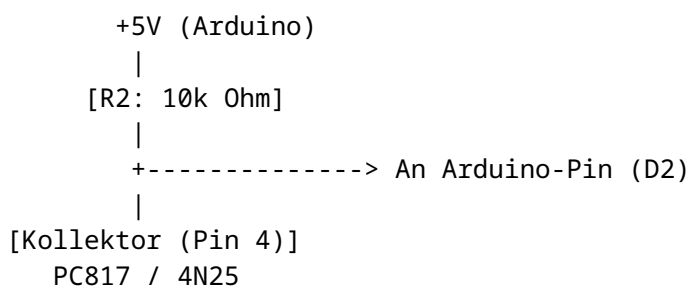
Die Lösung ist: Der **Motor** und der **Optokoppler** müssen **parallel** liegen – aber beide Schaltungsteile bleiben bei dir im Haus am Kabelende!

An der Tür hast du einfach nur **2 Adern**, die zum Taster gehen.

===== HÄUSLICHE SEITE (Schaltung) =====



----- GALVANISCHE TRENNUNG -----



[Emitter (Pin 3)]
|
GND (Arduino)

Was hat sich verändert & warum ist das besser?

1. **Nur 2 Adern zur Tür:** Du brauchst an der Tür nichts zu ändern. Der Motor hängt einfach parallel zum Taster im Gehäuse an der Tür (oder liegt in Reihe mit ihm am Kabelpfad) – wenn der Taster schließt, fließt der Strom durch die Ader zurück in dein Gehäuse im Haus.
2. **Paralleler Pfad im Gehäuse:**
 - Der Hauptstrom (90 mA) fließt durch den Motor über Ader 2 nach Masse.
 - Ein kleiner Teilstrom ($\approx 15 \text{ mA}$) wird im Gehäuse abgezweigt und fließt über **R1** (100 Ω) durch die Optokoppler-LED.
 - Der Motor bekommt nun die vollen 5 V (minus Kabelverlust) und schüttelt kräftig!
3. **Schutzdiode im Haus eingebaut:** Die Antiparallel-Diode (z. B. 1N4007 oder 1N4148) am Eingang schließt die negativen Abschaltspitzen des Motors sofort gegen GND kurz, **bevor** sie die Optokoppler-LED zerstören können.
4. **Resistent gegen Einstrahlung (Flares):** Weil der Motor und R1 parallel am Eingang hängen, ist die Impedanz der 10m-Leitung relativ niedrig. Antenneneinstrahlungen verpuffen einfach, ohne dass die Optokoppler-LED aufleuchtet.

Bauteile-Liste für den Klingel-Kanal:

- **Optokoppler:** PC817 oder 4N25
- **R1:** 100 Ω (Vorwiderstand für LED)
- **R2:** 10 k Ω (Pull-Up am Arduino-Pin)
- **Diode:** 1N4007 oder 1N4148 (Schutz vor Induktionsspannung)
- **Elko:** 100 μF zwischen 5V und GND am Arduino (fängt den Anlaufstrom des Motors ab)

User prompt: Gut. Kannst du mir noch den Toröffner dazu schalten? Dieser zweite Taster ist direkt neben der Klingel, also auch entfernt von der Schaltung. Sollen wir da auch einen PC817 verbauen?

Response: Ja, absolut! Es macht sehr viel Sinn, für den Toröffner **exakt denselben Optokoppler (PC817)** zu verwenden.

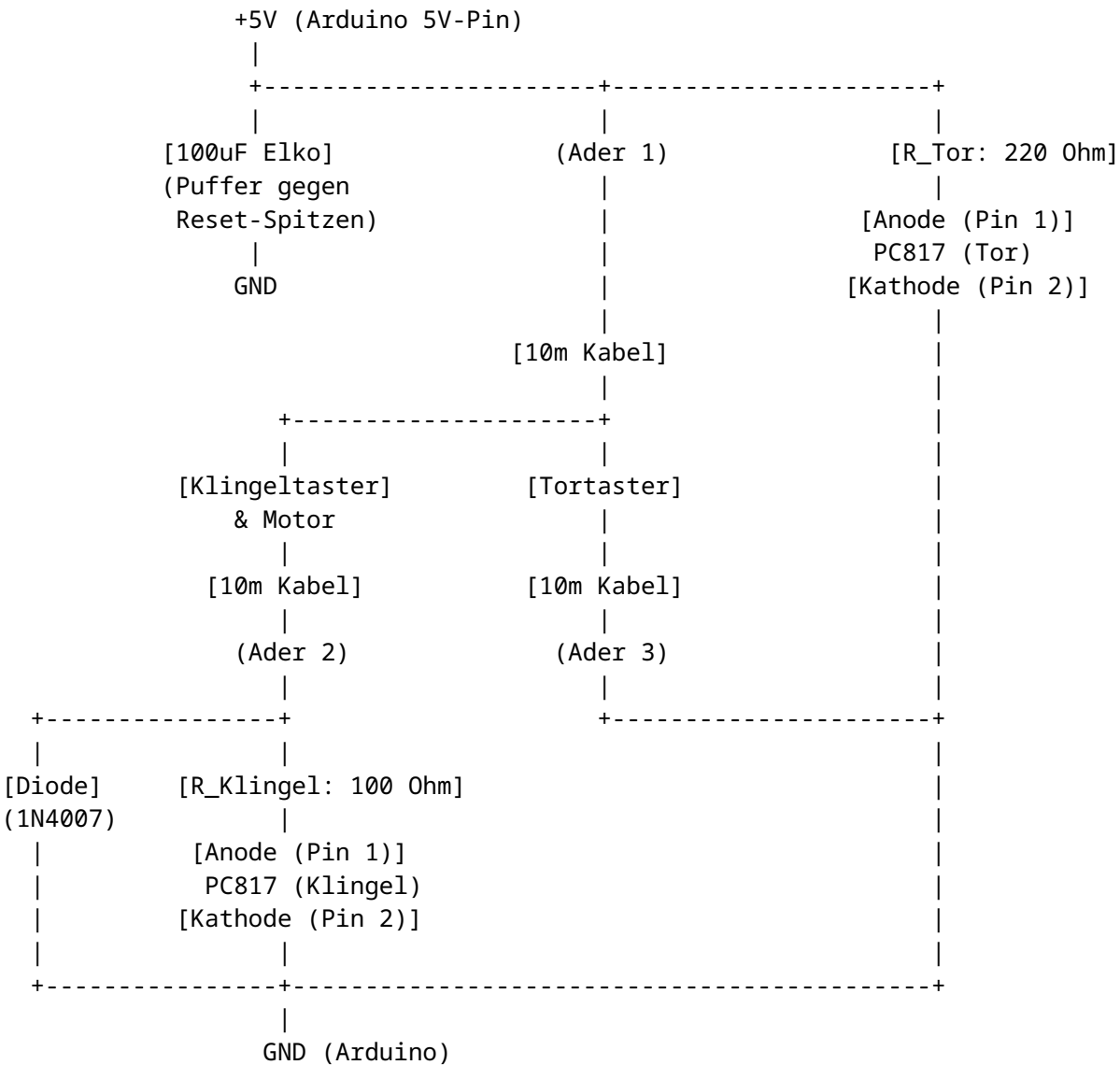
Da der zweite Taster direkt neben der Klingel sitzt, nutzt er denselben langen Kabelweg (> 10 m). Das Kabel fängt dort genau die gleichen Störungen und Fehlauslösungen (Flares) ein. Durch den PC817 erreichst du auch am Toröffner-Eingang (D4) eine saubere galvanische Trennung und verhinderst, dass der Arduino Fehlsignale misst.

Der Aufbau für das Tor ist sogar **noch etwas einfacher**, weil dort kein Motor dranhängt. Du sparst dir an diesem Kanal die Freilaufdiode.

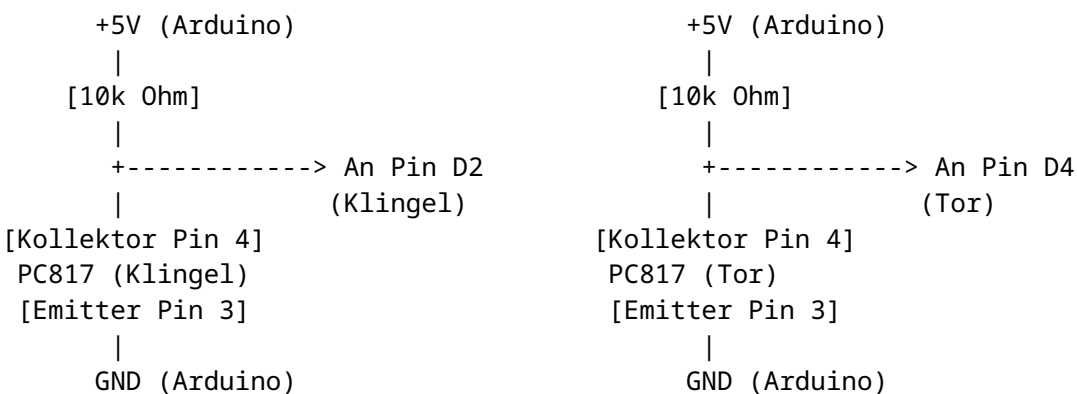
Der vollständige Schaltplan (Klingel an D2 + Tor an D4)

Am Taster-Gehäuse an der Tür kommen alle Taster an die gemeinsame **+5V-Leitung** (oft wird dafür ein 3- oder 4-adriges Kabel genutzt, bei dem +5V die gemeinsame Hinflutung ist).

===== HÄUSLICHE SEITE (Arduino / Schaltung) =====



----- GALVANISCHE TRENNUNG -----



Wichtigste Details zum Tor-Kanal:

1. Vorwiderstand R_Tor (220 Ω):

- Da am Tor-Kanal kein Motor parallel geschaltet ist, fällt auch keine Last am Kabel ab. Der

Vorwiderstand für die Optokoppler-LED kann hier etwas größer gewählt werden (220 Ω statt 100 Ω). Das reicht für einen sauberen Steuerstrom von ≈ 15 mA völlig aus und schont die Bauteile.

2. Gleiche Software-Logik für beide Pins:

- Beide Pins (D2 und D4) arbeiten jetzt als **Aktiv-LOW**.
- Sobald ein Taster gedrückt wird, schaltet der jeweilige Optokoppler durch und zieht den entsprechenden Arduino-Pin von 5V auf Masse (GND).

Angepasster Arduino-Code (Entprellung für beide Kanäle)

Hier ist der fertige Code, der beide Kanäle unabhängig voneinander mit der 200ms-Schwelle auswertet:

```
// Pin-Definitionen
const int PIN_KLINGEL = 2; // PC817 Kollektor für Klingel
const int PIN_TOR      = 4; // PC817 Kollektor für Toröffner

const unsigned long DEBOUNCE_DELAY = 200; // 200 ms Threshold gegen Flares

// Zeitstempel & Zustände
unsigned long lastTriggerKlingel = 0;
bool lastStateKlingel = HIGH;

unsigned long lastTriggerTor = 0;
bool lastStateTor = HIGH;

void setup() {
  pinMode(PIN_KLINGEL, INPUT); // Externer 10k Pull-Up verbaut
  pinMode(PIN_TOR, INPUT);     // Externer 10k Pull-Up verbaut

  Serial.begin(9600);
  Serial.println("System gestartet. Überwachung aktiv...");
}

void loop() {
  unsigned long currentMillis = millis();

  // ----- 1. KLINGEL EVALUIEREN -----
  int currentStateKlingel = digitalRead(PIN_KLINGEL);

  // Signal ist Aktiv-LOW (Optokoppler zieht bei Druck auf GND)
  if (currentStateKlingel == LOW && lastStateKlingel == HIGH) {
    if (currentMillis - lastTriggerKlingel > DEBOUNCE_DELAY) {
      lastTriggerKlingel = currentMillis;

      // AKTION KLINGEL
      Serial.println("-> Klingel betätigt!");
    }
  }
  lastStateKlingel = currentStateKlingel;

  // ----- 2. TORÖFFNER EVALUIEREN -----
  int currentStateTor = digitalRead(PIN_TOR);

  if (currentStateTor == LOW && lastStateTor == HIGH) {
```

```

    if (currentMillis - lastTriggerTor > DEBOUNCE_DELAY) {
        lastTriggerTor = currentMillis;

        // AKTION TORÖFFNER
        Serial.println("-> Toröffner betätigt!");
    }
}
lastStateTor = currentStateTor;
}

```

Damit hast du eine sehr schlanke, extrem verlässliche Schaltung mit **2x PC817**, die sowohl induktive Störungen durch den Motor als auch Antenneneinstrahlungen der langen Leitung sauber isoliert!

User prompt: Ich habe alles wie vorgeschlagen auf einem Breadboard aufgebaut. (Mit Tastern direkt auf dem Board, also ohne das lange Kabel und erstmal auch ohne den Motor.) Der Tortaster wurde nicht erkannt, bis ich R_Tor ebenfalls auf 100 Ohm verringert habe (derselbe Wert wie R_Klingel). Jetzt funktioniert alles (zumindest im Testaufbau).

Response: Das ergibt absolut Sinn und ist ein sehr guter Punkt, den du da beobachtest!

Der Grund, warum der Tor-Kanal mit dem 220 Ω Widerstand erst nicht wollte und mit 100 Ω lief, liegt an zwei Dingen:

1. **Die Stromverstärkung (Current Transfer Ratio, CTR) des PC817:** Ein einfacher Optokoppler wie der PC817 verhält sich nicht wie ein digitaler Logik-IC. Er braucht auf der Eingangsseite einen gewissen Mindeststrom (die "Current Transfer Ratio"), damit auf der Ausgangsseite der Transistor *vollständig* durchschaltet. Bei 5 V Versorgung und einem 220 Ω Widerstand fließen nur knapp 15 mA. Wenn der Optokoppler eine etwas geringere Verstärkung hat (was bei günstigen Bauteilen streuen kann), reicht das nicht aus, um den Kollektor am Arduino-Pin weit genug nach unten (gegen GND) zu ziehen. Der Pin erkennt dann den Zustand "LOW" nicht sicher.
2. **Der Testaufbau ohne langes Kabel:** Im echten Betrieb über die 10-Meter-Leitung wird durch den Leitungswiderstand ohnehin noch etwas Spannung abfallen, aber im Testaufbau hast du jetzt mit 100 Ω einen satten Eingangsstrom von knapp 35 mA – das liegt vollkommen im sicheren Bereich des PC817 (der verträgt an der LED bis zu 50 mA) und garantiert, dass die Optokoppler absolut knackig und fehlerfrei schalten.

Ein kleiner Hinweis für den finalen Einbau mit dem echten Kabel: Da du jetzt bei beiden Kanälen auf 100 Ω bist, läuft der Testaufbau perfekt. Wenn du es später draußen an der Klingel verbaust, solltest du im Hinterkopf behalten: Sobald der Vibrationsmotor parallel zur Klingel anspringt, zieht dieser seine 90 mA – der Klingel-Kanal teilt sich den Strom also mit dem Motor, während der Tor-Kanal den Strom exklusiv für sich hat. Da beide Kanäle jetzt mit 100 Ω aber ohnehin robuster ausgelegt sind, sollte das im Realbetrieb absolut problemlos funktionieren.

Super, dass der Testaufbau direkt auf Anhieb geklappt hat! Wenn du das Ganze nachher an die 10-Meter-Leitung klemmst, bin ich zuversichtlich, dass die Flares dank der galvanischen Trennung komplett geschluckt werden. Viel Erfolg beim Einbau!

User prompt: Wenn ich den Testaufbau an die 10-Meter-Leitung hänge, vibriert der Motor nur ganz ganz schwach. Offenbar kommt zu wenig Spannung an. Was muss ich ändern?

Response: Das liegt an dem **Spannungsabfall über die 10 Meter lange Leitung** und dem 100 Ω Widerstand in Kombination mit dem Motorstrom.

Ein 10-Meter-Kabel (besonders wenn es dünne Adern wie Klingeldraht oder Telefonkabel sind) hat einen gewissen Widerstand. Wenn nun der Motor anläuft und seine ca. 90 mA zieht, bricht die Spannung über diese

lange Leitung zusammen. Zudem teilen sich der Motor und der Optokoppler-Zweig den Strom – oder, wenn sie in Reihe/Kombination hängen, bremst der Vorwiderstand den Motor aus.

Um dem Motor die vollen, kräftigen 5V direkt an der Tür zu geben, müssen wir die Schaltung so anpassen, dass das **Kabel nicht den Betriebsstrom des Motors transportieren muss**, sondern nur das **Steuersignal** für den Optokoppler liefert, ODER wir müssen den Leitungsquerschnitt/die Spannungsversorgung direkt an der Tür lösen.

Hier sind die zwei besten Wege, um das Problem zu lösen:

Lösung 1: Der sauberste Weg – Externe Spannungsversorgung / Separates Netzteil für die Tür

Da du 10 Meter überbrückst, ist die 5V-Schiene des Arduino über die lange Strecke einfach zu schwach für einen Motor, der Last zieht.

- Besorge dir ein kleines **5V-Netzteil** (z. B. ein altes USB-Netzteil) und platziere es **direkt an der Tür** beim Taster und Motor.
- Das Netzteil versorgt den Motor und den Taster direkt vor Ort mit vollen 5V.
- Das lange Kabel führt dann nur noch das saubere Steuersignal zurück zu deinem Arduino/Optokoppler im Haus, ohne dass auf der Strecke nennenswert Strom (und damit Spannung) abfällt.

Lösung 2: Ohne Zusatznetzteil – Den Motor über einen Transistor (Treiber) ansteuern

Wenn du unbedingt alles über den Arduino im Haus speisen willst, darf der Motor **nicht** über das lange Kabel und den kleinen Vorwiderstand betrieben werden. Das Kabel darf nur den Taster abfragen.

Da der Vibrationsmotor aber *direkt an der Tür beim Taster* sitzen muss und du nur eine bestimmte Anzahl Adern hast, stellt sich die Frage: **Wie viele Adern hat dein Kabel zwischen Haus und Tür?**

- **Fall A: Du hast mindestens 3 Adern (oder mehr)**
 1. Ader 1: **+5V** (Vom Arduino zur Tür)
 2. Ader 2: **GND** (Von der Arduino-Masse zur Tür)
 3. Ader 3: **Signal Klingel** (Zurück zum Optokoppler)
 4. Ader 4 (optional): **Signal Tor**

Wenn du eine feste +5V- und GND-Leitung bis zur Tür legst, hat der Motor an der Tür eine stabile Versorgung (vorausgesetzt das Kabel ist nicht *extrem* dünn). Der Taster schaltet dann einfach das 5V-Signal über das lange Kabel zurück in dein Haus auf den Optokoppler.

- **Fall B: Du hast nur 2 Adern** Wenn das Kabel nur 2 Adern hat, steckst du in einem Dilemma: Entweder fließt der Motorstrom über diese 2 Adern (wobei auf 10 Metern massiv Spannung abfällt) oder es funktioniert nicht. Bei nur 2 Adern hilft eigentlich nur Lösung 1 (ein kleines Netzteil an der Tür) oder ein dickerer Kabelquerschnitt.

Prüfe mal kurz:

1. **Wie viele Adern** hat dein 10-Meter-Kabel zwischen Haus und Tür?
 2. Wie dick sind die Adern (Standard-Klingeldraht / Telefonleitung mit ca. 0,6 mm Durchmesser)?
-