

Zusammenfassung Physical Computing

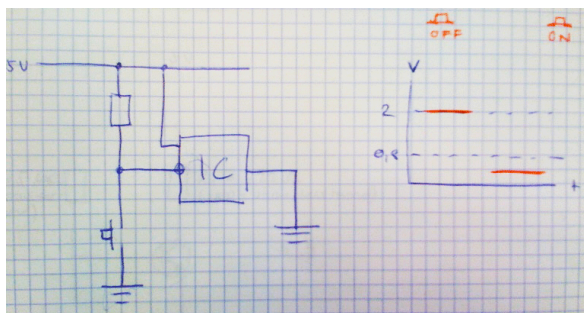
Widerstände

Farben

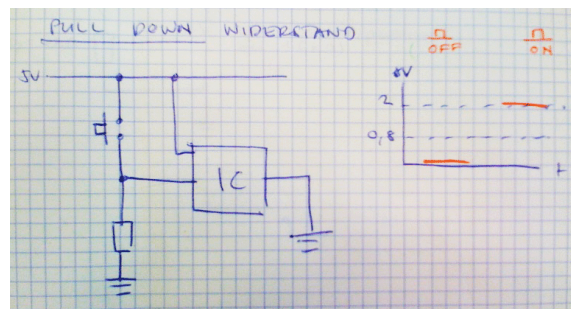
	1. Ring	2. Ring	3. Ring	4. Ring
220 Ω	Rot	Rot	Braun	Gold
	2	2	10 (Multiplikator)	5 % Toleranz

Gesamtwiderstand in einer Parallelschaltung: $R = \frac{R1 \times R2}{R1+R2}$

Pull-Up Widerstand



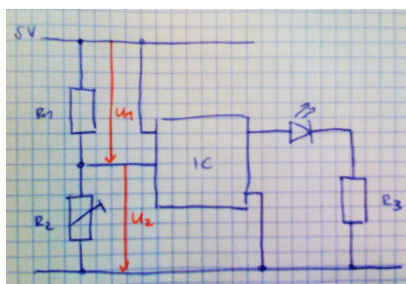
Pull-Down Widerstand



Spannungsteiler mit Variablem Widerstand

Die Spannung teilt sich in einer Serieschaltung im Verhältnis der beiden Widerstände.

Es wird ein Vorwiderstand R1 gesucht, der die Spannung für den Eingang beim IC (U2) optimal teilt. Bei einem hohen variablem Widerstand R2 sollte sich die Spannung U2 5V annähern, bei einem kleinen variablem Widerstand R2 sollte sich die Spannung U2 0V annähern.



Es wird zum Beispiel ein Vorwiderstand von 47kΩ genommen:

$$R1 = 47k\Omega$$

$$R2 = 10k\Omega \text{ bis } 300k\Omega \text{ (gemessene Werte)}$$

$$R_{Ges} \text{ bei } 10k\Omega = 57k\Omega$$

$$R_{Ges} \text{ bei } 300k\Omega = 347k\Omega$$

	Bei 10kΩ	Bei 300kΩ
$\frac{R2}{R_{Ges}} = \frac{U2}{U_{Ges}}$	$\frac{10}{57} = \frac{x}{5}$	$\frac{300}{347} = \frac{x}{5}$

	U2 = 0.8 V	U2 = 4.3 V
--	-------------------	-------------------

Die Spannung variiert zwischen 0.8 V und 4.3 V.

Diode: Lässt Strom nur in eine Richtung durch.

Kondensator: Lädt sich mit einer bestimmten Menge Strom auf und gibt sie bei Stromunterbruch wieder ab.

LED: Light Emitting Diode. Durchlassspannung bei Rot z.B. 1,6 V bei 20mA. Es braucht einen Vorwiderstand, um den Strom zu begrenzen!

Arduino:

- 13 Pins: Digitale In- und Outputs
 - 0: Receive (RX)
 - 1: Tranceive (TX)
- 6 Analoge In- und Outputs
 - Inputs: Werte von 0-1023
 - Outputs: Werte von 0-255
- Pro Pin maximal 40mA (entspricht ungefähr 2 LEDs)

Debouncing: Z.B. beim Drücken eines Tasters können Elektronen zu früh überspringen. Deshalb wird ein Delay eingebaut → Debounce

Bit Shifting « »

« Shift nach links
» Shift nach rechts

01 « ergibt 010
10 » ergibt 1

MSB / LSB

MSB: Most Significant Bit: 7654 3210
LSB: Least Significant Bit: 0123 4567

Beispiel:

3 Eingänge beim Arduino A, B und C

(MSB)

	C	B	A
	1	0	1
B «		00	
C « 2	100		

A | B | C = 001 | 000 | 100 = 101 = 5

PWM / PBM

PWM: Puls Width Modulation

PBM: Puls Breiten Modulation

Wird gebraucht um z.B. ein LED zu dimmen

