

PROJEKT 1 – LED BLINKEN

EINFÜHRUNG

In diesem Projekt geht es um die absoluten Basics des Arduinos. Sie werden die Programmierung sowie elementare Funktionen anhand von verschiedenen, einfachen Aufgaben kennenlernen. Arbeiten Sie die Aufgaben nacheinander ab. Die benötigten Kenntnisse werden schrittweise und vor allem durch die Erklärungen in dem Quellcode der Programme (die so genannte Kommentierungen) vermittelt.

Sollte etwas nicht auf Anhieb klappen, geben Sie nicht auf! Lesen Sie die ggf. angezeigten Fehlermeldungen, die Hinweise in den Quellcodes und vor allem: probieren Sie aus.

INHALTE UND ZIELE

- Ein- und Ausschalten von LEDs
- Einstellen einer Pause (delay)
- Dimmen von LEDs mit Pulsweitenmodulation (PWM)
- Erstellen und Aufrufen von Funktionen

MATERIAL

- Arduino UNO (vgl. Abb. 1)
- 2x LED mit 220 Ω Vorwiderstand (vgl. Abb. 1)
- USB-Kabel
- Laptop mit installierter Arduino-Software
- Programm „Projekt_1_LED_Blinken.ino“



Abbildung 1: Arduino UNO mit LED samt Vorwiderstand

SCHALTPLAN

Der Schaltplan (vgl. Abb. 2) ist bei diesem Projekt sehr einfach. Eine LED wird samt 220 Ω Vorwiderstand an Pin 6 (Anode) und GND (Kathode) geschaltet.

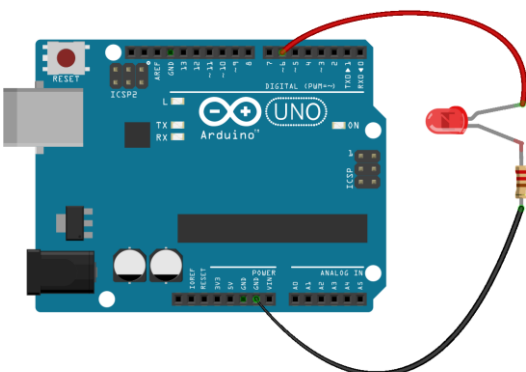


Abbildung 2: Schaltplan mit Arduino UNO mit LED samt 220 Ω Vorwiderstand

VORBEREITUNG

- Verschalten Sie die LED samt Vorwiderstand (220 Ohm) mit dem Arduino (vgl. Abb. 1 und 2). Stecken Sie die Anodenseite (hier: der Draht an der LED) in die digitale Stiftleiste an **Digital-PIN 6**. Stecken Sie die Kathodenseite (hier: der Draht am Vorwiderstand) auf der gegenüberliegenden Seite in einen der beiden **GND-Ports**.
- Verbinden Sie den Arduino mit Ihrem Laptop per USB-Kabel (der Arduino wird direkt mit Energie versorgt).
- Starten Sie die Arduino-Software auf Ihrem Computer. Wählen Sie dort den richtigen **Arduino-Typ** (Arduino UNO) und stellen anschließend den passenden **Port** ein (wo der Arduino angezeigt wird).
- Laden Sie das Programm „**Projekt_1_LED_Blinken.ino**“ in die Arduino-Software. Sie sehen den Quellcode im Fenster, das Programm ist aber noch nicht auf dem Arduino!
- Übertragen Sie das Programm mittels Klick auf den Pfeil (vgl. Abbildung 3) auf den Arduino. Nach erfolgreichem Upload wird das Programm automatisch ausgeführt. Wenn alles geklappt hat, blinkt die LED.



Abbildung 3: Bedienleiste in der Arduino-Software von *arduino.cc*. Der Pfeil lädt das Programm auf den Arduino hoch.

Was tun, wenn es nicht blinkt?

- Wurde das Arduino-Programm übertragen oder wird ggf. eine Fehlermeldung angezeigt?
- Wurde der richtige PIN verwendet? Für dieses Programm Digital-PIN 6 & GND.
- Ist die LED richtig herum eingesteckt? LEDs funktionieren nämlich als Dioden nur „in eine Richtung“.

INFOS MORSECODE

Wie funktioniert Morsen?

- Morsecodes (s. Abbildung 3) bestehen aus kurzen (DIT) und langen Signalen (DAH) in Form von z.B. Tönen oder Lichtzeichen.
- Um zwischen langen (DAH) und kurzen Signalen (DIT) zu unterscheiden, ist festgelegt, dass ein langes Signal (DAH) dreimal so lang ist, wie ein kurzes (DIT). Wenn Ihr „DIT“ also 0,5 Sekunde lang ist, muss Ihr DAH 1,5 Sekunden lang sein.
- Innerhalb eines Morsezeichens (z.B. zwischen kurz, kurz, lang) wird nur eine sehr kurze Pause verwendet. Die Pause zwischen einzelnen Buchstaben hat aber eine definierte Länge von einem DIT.

A	● —	U	● ● —
B	— ● ● ●	V	● ● ● —
C	— ● — ●	W	● — — —
D	— ● ●	X	— ● ● —
E	●	Y	— ● — —
F	● ● — ●	Z	— — ● ●
G	— — ● ●		
H	— ● ● ●		
I	● ●		
J	● — — —		
K	— ● — —		
L	— ● ● ●		
M	— — —		
N	— ●		
O	— — —		
P	● — — —		
Q	— — ● —		
R	● — ● ●		
S	— ● ●		
T	— — —		
		1	● — — — —
		2	● ● — — —
		3	● ● ● — —
		4	● ● ● ● —
		5	● ● ● ● ●
		6	— ● ● ● ●
		7	— — ● ● ●
		8	— — — ● ●
		9	— — — — ●
		0	— — — — —

Abbildung 3: Morsezeichen Quelle: en.wikipedia.org/wiki/Morse_code