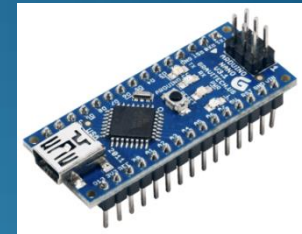
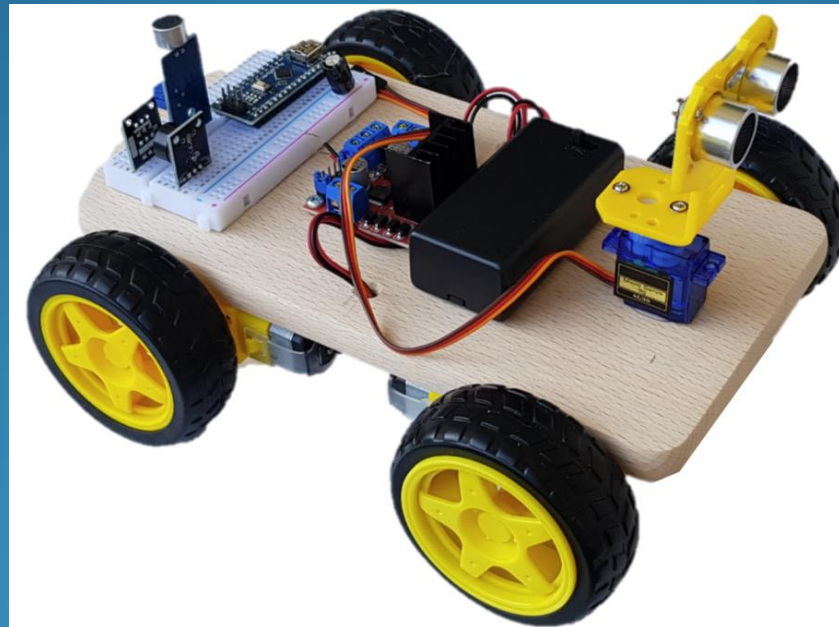


# GTA Mechatronik Teil 1

Wir bauen und programmieren ein Roboterauto und lernen dabei viel über Motoren, Servos, Sensoren, Elektronik und Software



# Inhaltsverzeichnis

---

## Tutorial

[Mechatronik](#)

[Roboterauto](#)

[Elektromotoren](#)

[Batterien und Akkus](#)

[Drähte und Litzen](#)

[USB](#)

[Löten](#)

[Breadboard](#)

[Elektronik – ESD Schutz](#)

[Mikrocontroller und Arduino](#)

[Programmiersprachen](#)

[Arduino IDE](#)

[Piezo \(Buzzer\)](#)

[Sensoren](#)

[Servo](#)

[Infrarot-Fernsteuerung](#)

[Programmablaufplan](#)

## Sketche

[Zuordnung Pins am Arduino Nano](#)

[Start und Aufbau eines Sketches](#)

[Sketch 00 Blink LED „L“](#)

[Sketch 01 Blink LED „L“ SOS](#)

[Sketch 02 Blink LED „L“ 5x](#)

[Sketch 03 Batteriespannung messen](#)

[Sketch 05 Serial Plotter-Sinuskurven](#)

[Sketch 06 Serial Plotter-Pulsmessung](#)

[Sketch 10 Auto programmiert fahren](#)

[Sketch 20 Buzzer Tone](#)

[Sketch 21 Buzzer Melody](#)

[Sketch 22 Ultraschall Abstandsmessung](#)

[Sketch 23 Auto Hinderniserkennung](#)

[Sketch 24 Geräusch erkennen](#)

[Sketch 25 Auto fahren US Geräusch](#)

[Sketch 26 Servo Mittelstellung](#)

[Sketch 27 Servo steuern](#)

[Sketch 28 Auto Hindernis Servo](#)

[Sketch 30 IR Fernsteuerung Ermittlg Code](#)

[Sketch 31 Auto IR Fernsteuerung](#)

[Sketch 35 Ampel](#)

## Anhang

[Abkürzungen](#)

[Weblinks](#)

[Udemy Arduino Lehrgang](#)

[Variablen](#)

[Bibliotheken \(Libraries\)](#)

[CH340](#)

[Bootloader für Arduino Nano](#)

[Widerstände](#)

[Multimeter](#)

[Oszilloskop Hantek](#)

[Loop Durchlaufzeit](#)

[MB102 Spannungsversorgung](#)

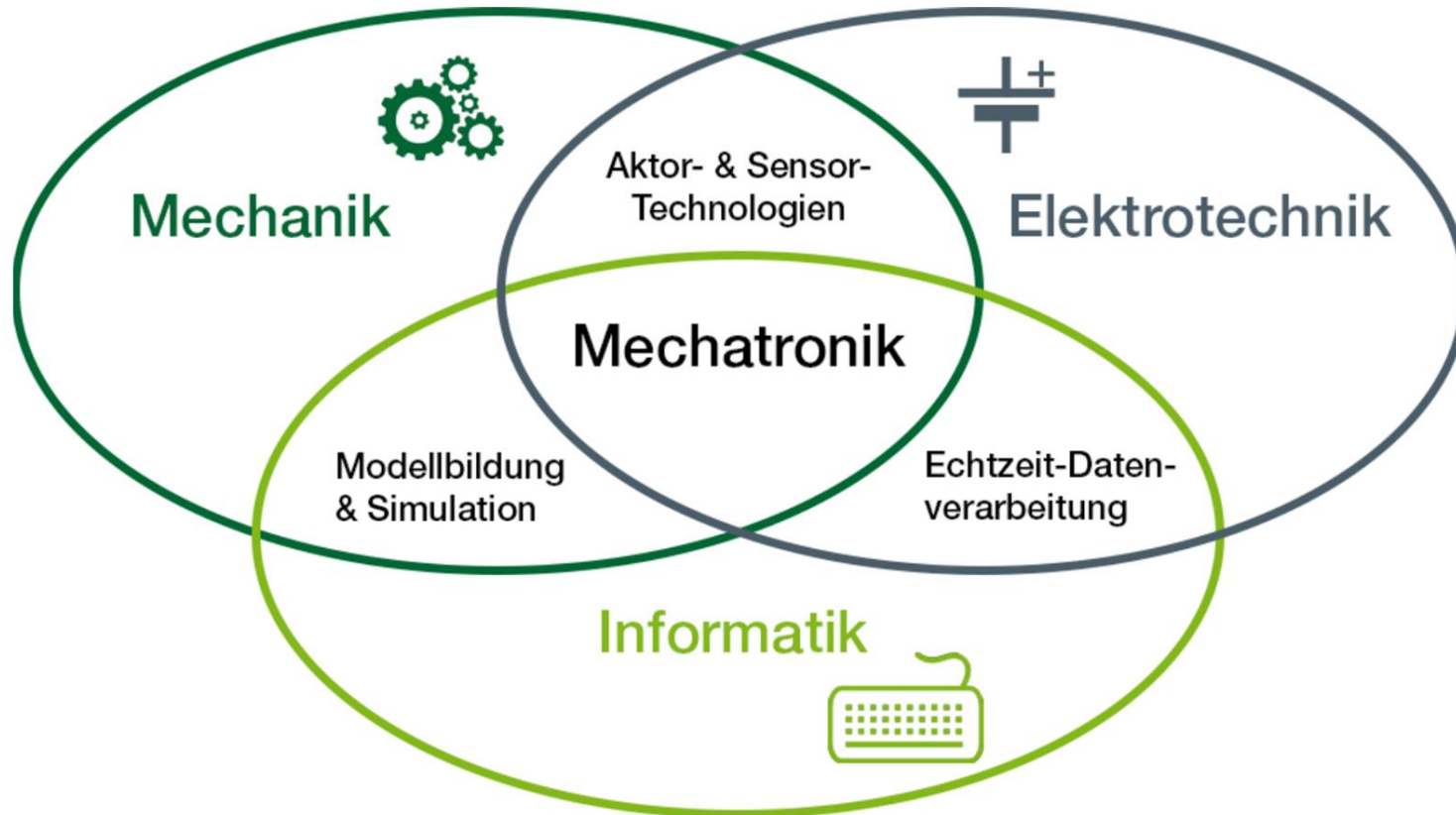
[Bezugsquellen](#)

[Simulation mit TINKERCAD](#)

[Lötbares Breadboard \(Leiterplatte\)](#)

# Mechatronik – was ist das ?

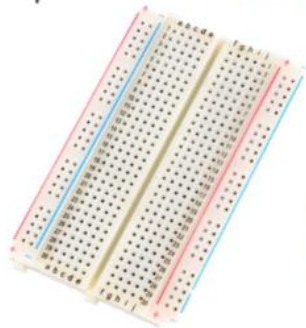
---



# Roboterauto Komponenten

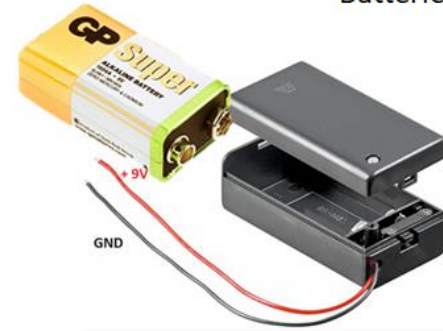


Arduino Nano



Breadboard

L298N DC Motor Driver Module



Batterie 9V



Buzzer  
(Piezo-Summer)



Servo SG90



Mikrofon



IR-Sensor



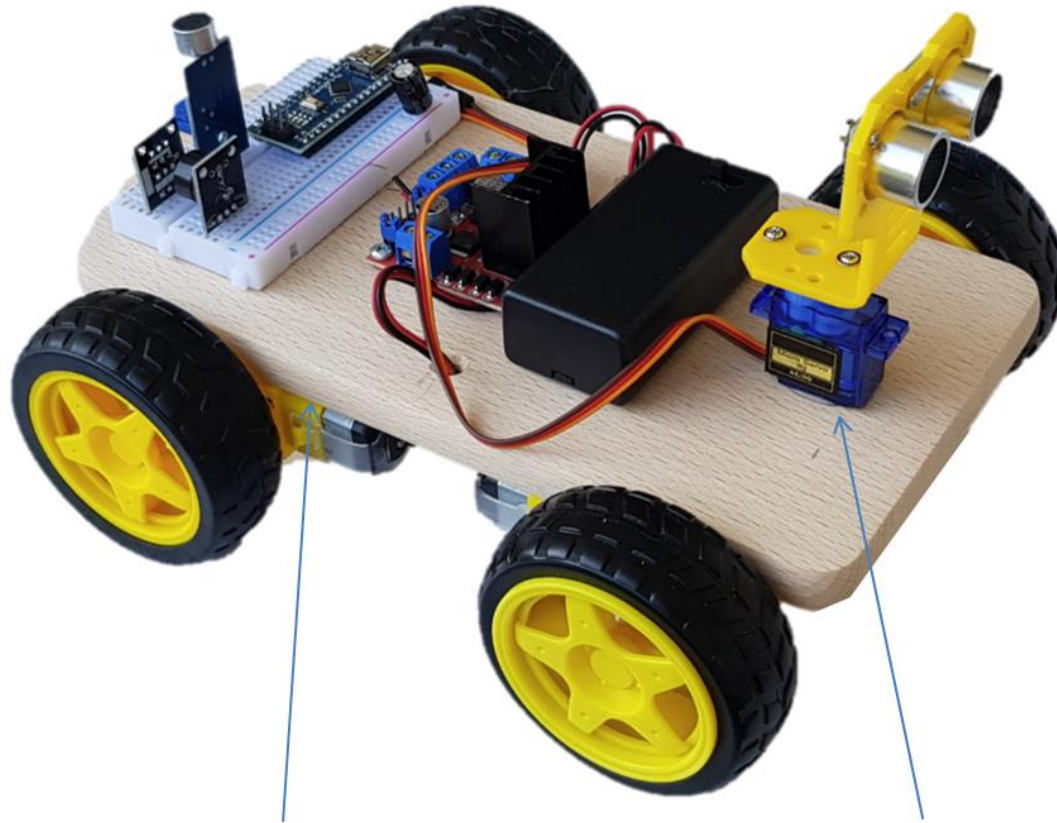
Radantriebe  
4x



US-Sensor

# Roboterauto montiert (ohne Verdrahtung)

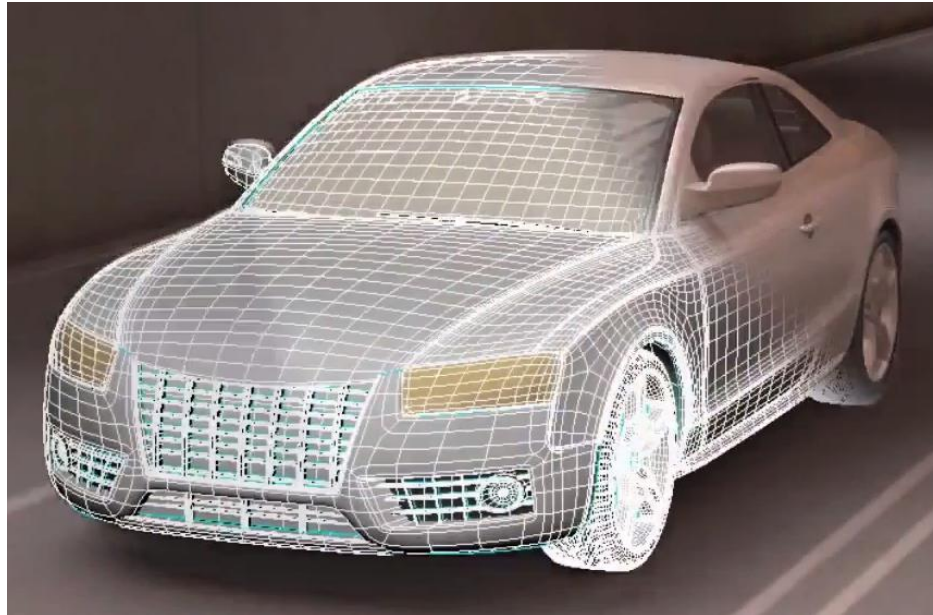
---



Heißkleber (vorher Kunststoffoberfläche mit Sandpapier anrauhen)

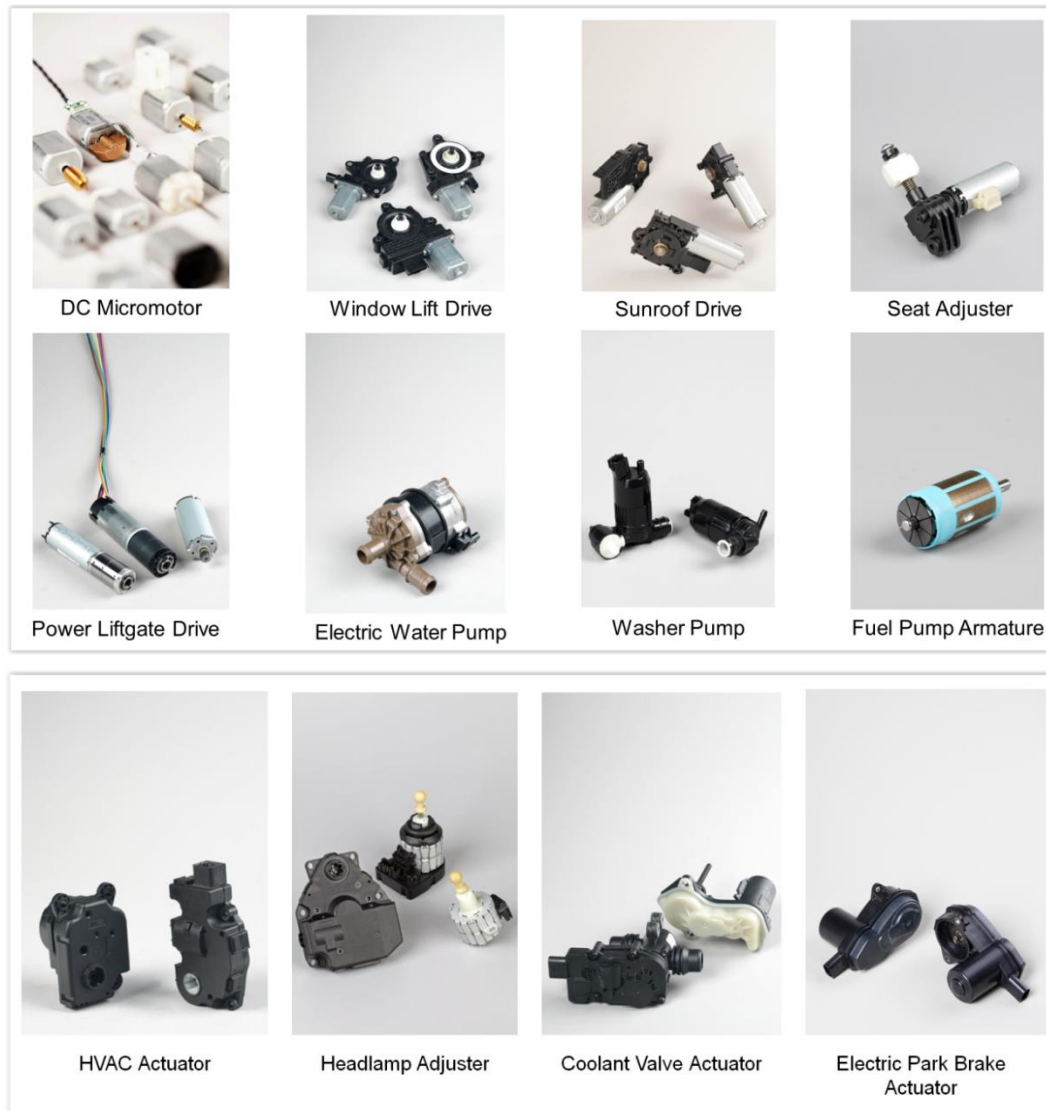
# Elektromotoren im Auto

---



<https://gtamechatronik.de/Teil1/>

# Elektromotoren im Auto



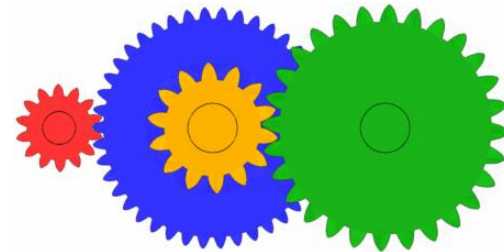
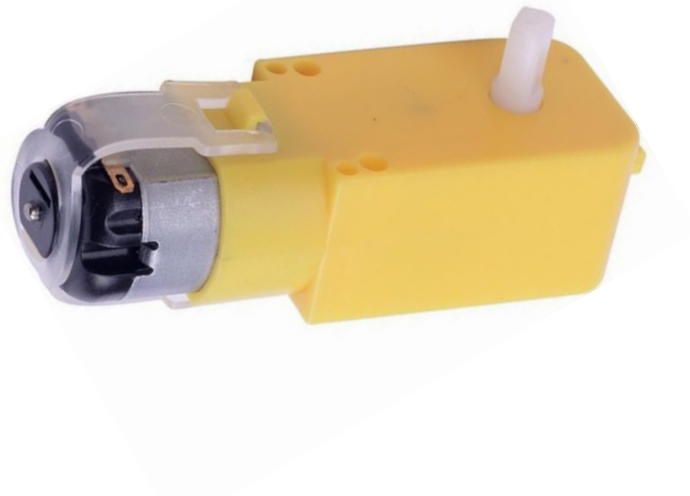
# Gleichstrom-Motor (DC-Motor) und Getriebe

---

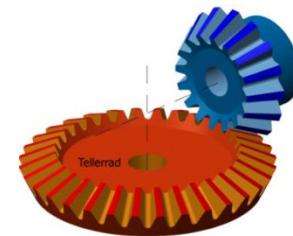
- Gleichstrom-Motor (DC Motor ... Direct Current Motor)



- Getriebe (Gearbox)



<https://fr.m.wikipedia.org/wiki/Fichier:AnimatedGears.gif>





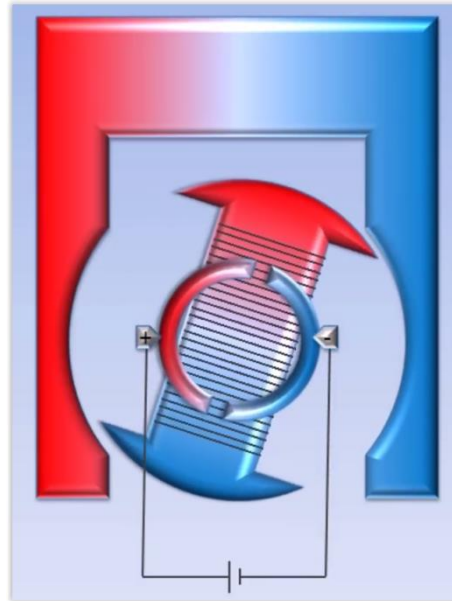
# Gleichstrom-Motor (DC-Motor) und Getriebe

---



# Gleichstrom-Motor einfach erklärt

---

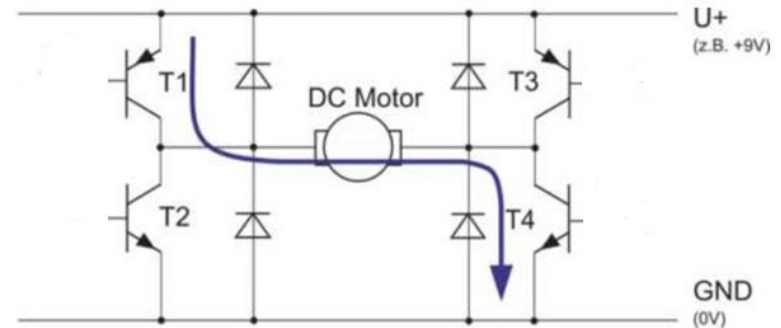
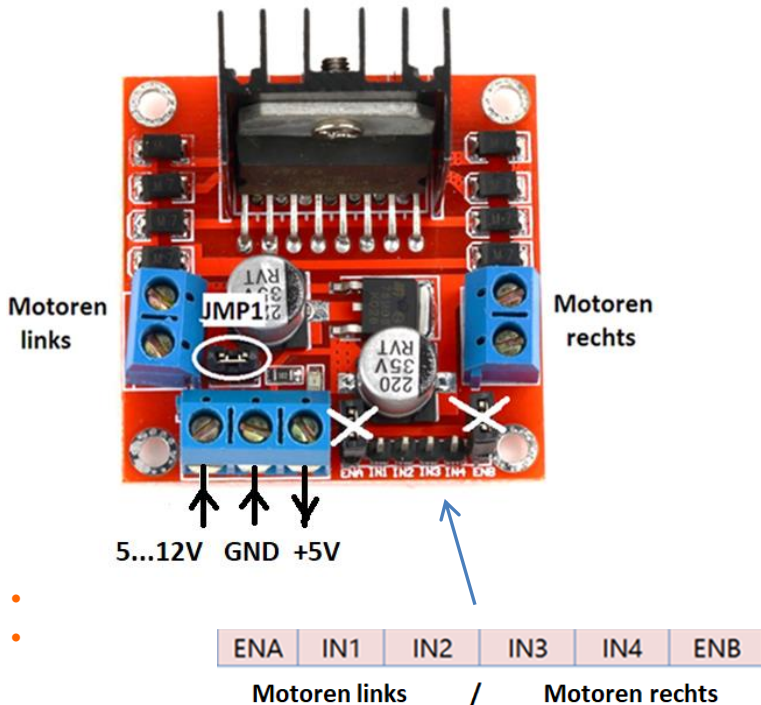


<https://www.youtube.com/watch?v=glUcvuBfV9o>

# Gleichstrom-Motor Treiber Modul

- L298N Dual H Bridge DC Motor Driver Module

Das Modul enthält zwei solche H-Brücken (H-Bridges):



Jede H-Brücke treibt zwei Motoren an.

Die Motoren rechts sowie die Motoren links werden jeweils parallel geschaltet.

Zulässige Eingangsspannung: 5 - 35V.

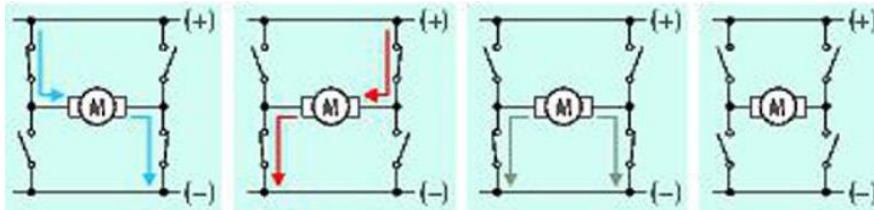
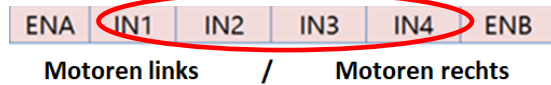
Auf dem Modul ist ein Spannungsregulator, der eine Ausgangsspannung von 5V erzeugt – aber nur, wenn die Eingangsspannung 5 – 12 V beträgt.

Die kann zur Versorgung des Arduino oder anderer Module benutzt werden.

Dazu muß die Brücke (Jumper) JMP1 gesetzt werden. Die anderen beiden Jumper sind zu entfernen.

# Steuerung des Gleichstrom-Motors

## Steuerung Start/Stop und Drehrichtung



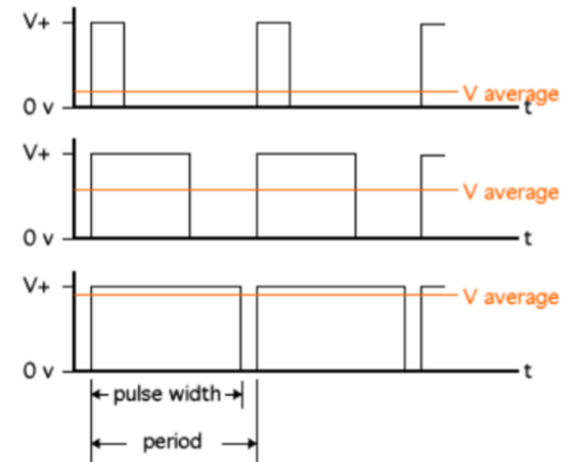
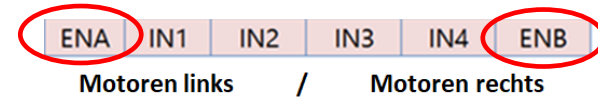
Vorwärts

Rückwärts

Stopp  
Bremsen

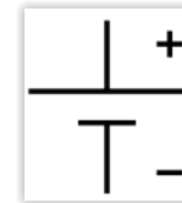
Stopp  
Auslaufen

## Steuerung Drehzahl



# Batterien und Akkus

- Eine Batterie ist ein elektrochemischer Energiespeicher.
- Bei der Entladung wird gespeicherte chemische Energie in elektrische Energie umgewandelt.
- Primärzellen können nur einmal entladen und nicht wieder aufgeladen werden.
- Sekundärzellen können wieder aufgeladen werden (Akkus).
- Gebräuchliche Baugrößen von Batterien:



- Alkali-Mangan-Zellen (Alkaline) sind der am weitesten verbreitete Batterietyp.
- Es sind Primärzellen, die Spannung pro Zelle beträgt 1,5V.
- Je größer das Volumen, desto mehr Energie kann gespeichert werden.
- Die gespeicherte Energie wird in mAh angegeben.
- Ein Wert von 1000 mAh bedeutet, daß bei Raumtemperatur ein Gerät über 10 Stunden mit 100 mA versorgt werden kann, oder über 5 Stunden mit 200mA.

# Batterien und Akkus - 9V Block

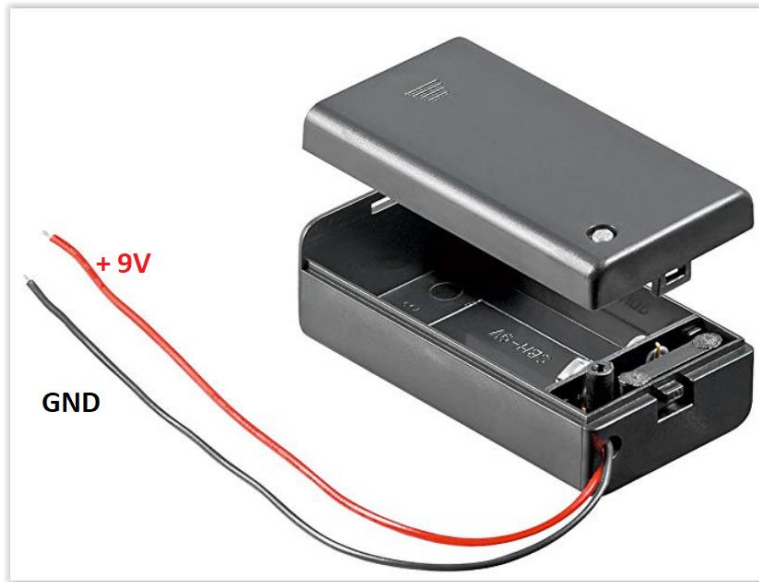
- Durch Reihenschaltung mehrerer Zellen erhält man eine höhere Batteriespannung.
- Gebräuchlich ist der sogenannte 9V Block.
- Ein 9-V-Block aus Alkali-Mangan-Zellen kann unterschiedlich aufgebaut sein: Entweder es finden sechs Rundzellen oder sechs Flachzellen Anwendung.



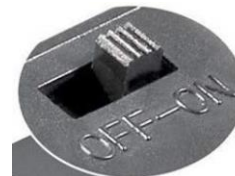
- Neben Alkaline 9V Blockbatterien gibt es auch Lithium 9V Blockbatterien.
- Diese haben eine höhere Energiekapazität und eine höhere Langzeithaltbarkeit, sind aber auch deutlich teurer.



# 9V Alkaline Batterie in Batteriegehäuse mit Schalter

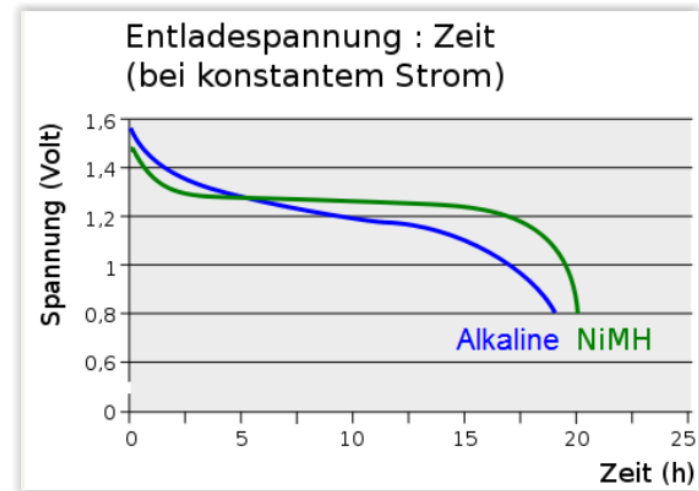


Alkaline Batterie  
nicht wiederaufladbar



# Batterien - NiMH Akkus und Akku Packs

- NiMH (Nickel Metallhydrid) Akkus sind wiederaufladbar.
- Die Spannung pro Zelle beträgt nach Wiederaufladung ca. 1,4V und fällt nach kurzer Betriebszeit auf ca. 1,2V.
- Diese Spannung bleibt längere Zeit stabil und fällt dann steil ab.
- Bei 1,0V muß der Akku wieder neu aufgeladen werden.
- Vergleich von Alkaline Batterie mit NiMH Akku:



- Im Modellbau werden NiMH Akku-Packs verwendet, mit mehreren in Reihe geschalteten Zellen.



# Modellbau-Akkupack (NiMh) 9.6 V 800 mAh

Conrad energy Modellbau-Akkupack (NiMh) 9.6 V 800 mAh , mit Tamiya-Stecker

<https://www.conrad.de/de/search.html?search=akkupack%20%2C6v>





Bestell-Nr.: 209081 - 62  
Teile-Nr.: 209081  
EAN: 4016138647334  
Datenblatt [Downloads](#)

[Weitere Varianten zu diesem Artikel](#)

 Drucken

Online

**12,99 €**  
inkl. MwSt., zzgl. Versand

• **Online verfügbar**  
Lieferung: Mi., 24.04  
[Fälligkeit prüfen](#)

ANSMANN Akkupack Ladegerät ACS 48 – Akkupacklader für 4 bis 8 zellige NiMH-Akkupacks

[https://www.amazon.de/gp/product/B00WI4KRTK/ref=ppx\\_yo\\_dt\\_b\\_asin\\_image\\_o06\\_s00?ie=UTF8&psc=1](https://www.amazon.de/gp/product/B00WI4KRTK/ref=ppx_yo_dt_b_asin_image_o06_s00?ie=UTF8&psc=1)



von **Ansmann**  
★★★★☆ 63 Kundenrezensionen | 33 beantwortete Fragen  
Bestseller Nr. 1 in RC-Ladegeräte

Unverb. Preisempf.: EUR 14,99  
Preis: **EUR 13,99** ✓prime  
Sie sparen: EUR 1,00 (7%)  
Alle Preisangaben inkl. deutscher USt. Weitere In

**Jetzt Amazon-Konto aufladen** und noch entspannter einkaufen.  
20 neu ab EUR 13,99 1 gebraucht ab EUR 13,56

Stil: **Lader für 4-8 zellige Akkupacks**

Lader für 1-10 zellige Akkupacks EUR 38,99	Lader für 4-8 zellige Akkupacks EUR 13,99
---	--

• Ein echter Allrounder: Das Ladegerät bietet mit dem Adapterkabel Ausgangssteckern vielfältige Lademöglichkeiten bei 4-8 zelligen A

# Batterien - LiPo Akkus und Akku Packs

---

- LiPo (Lithium Polymer) Akkus haben eine hohe Kapazität (viele mAh) bei geringem Gewicht, deshalb wird zum Beispiel für Flugmodelle nur noch dieser Akku-Typ eingesetzt.
- Die Spannung pro Zelle beträgt 3,7V.
- Tiefenentladung muß vermieden werden. Ist eine Lipo- Zelle auch nur ein einziges Mal unter 2,7 Volt entladen dann ist sie in der Regel bereits dauerhaft beschädigt.
- Um diese Spannung an jeder Zelle des Akkus auch am Ende des Ladevorgangs zu haben, benutzt man spezielle Ladegeräte mit einem BALANCER.
- **Achtung:**
- Es besteht Brandgefahr, wenn die Akkus falsch geladen werden !
- Deshalb verwenden wir diesen Akku-Typ nicht.



LiPo Akku 5000mAh 7,4V 60C  
Akku Pack Hard Case

# Drähte und Litzen

---

- Drähte bestehen aus einem Kupferdraht, welcher den Strom leitet, der von einer Kunststoff-Isolierung ummantelt ist.



- Litzen bestehen aus mehreren dünnen Kupferdrähten, welche von einer Kunststoff-Isolierung ummantelt sind.
- Sie sind dadurch sehr biegsam.



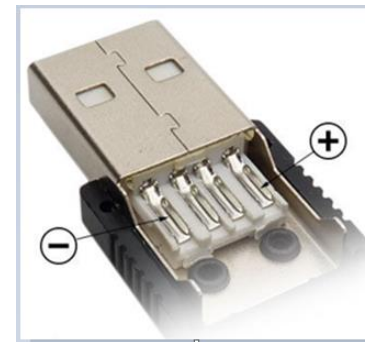
# USB

---

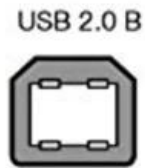
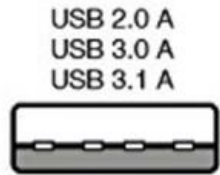
- USB (Universal Serial Bus) ist ein Bussystem, das für den direkten Betrieb von elektrischen Kleingeräten am USB-Hub eine Spannung von 5 Volt bereitstellt.
- Bei USB 2.0 darf die Last gemäß Spezifikation dort bis zu 500 Milliampere betragen, bei USB 3.0 sind es 900 Milliampere.
- Damit die elektrischen Kontakte des USB für die Lieferung von Betriebsspannung möglichst betriebssicher sind, sind die Kontakte 5 Volt (+) und eine GND (-) außen angeordnet und etwas länger als die beiden innenliegenden Kontakte für die USB-Datenleitungen. Dadurch soll beim Einstecken eines USB-Geräts die Datenverbindung erst nach der Stromverbindung hergestellt werden, beim Abziehen wird die Datenverbindung vor der Stromverbindung wieder getrennt.



- Ein Kurzschluß zwischen 5 Volt (+) und GND (-) muß unbedingt vermieden werden, sonst wird der USB-Treiber im Computer beschädigt !



# USB



- USB 2.0 (High Speed)

480 Mbit/s

USB 3.0 (Super Speed) 5 Gbit/s

- <https://handygad.de/blog/usb-arten-im-ueberblick/>



## **Sicherheitshinweise:**

- Lötkolben nur am Kunststoffgriff anfassen!
- Lötkolben nicht ruckartig bewegen – Zinn könnte sonst verspritzt werden !
- Nach dem Löten die Lötstation ausschalten !
- Lötfett darf nicht in die Augen gelangen !
- Dämpfe beim Löten sollten nicht eingeatmet werden !

# Löten

---

Lötstation mit Ersatzspitzen und Lötzinn  
60W, Regelbar 90 - 480°C, mit Anzeige der Lötspitzen-Temperatur



# Löten

---

Lote („Lötzinn“) sind für die mechanisch feste und leitende Verbindung von Metallen vorgesehen.

Das **bleihaltige** (Standard-) Lötzinn besteht aus einer Legierung aus Zinn (Sn) und Blei (Pb) und weiteren Metallen in geringer Menge.

Eine weit verbreitete Legierung ist Sn60Pb40 mit einem Schmelzpunkt um 185°C.

Blei gilt aber als Gesundheits-Risiko (deshalb auch bleifreies Benzin seit vielen Jahren).

Beim **bleifreien** Lötzinn wird das Blei durch Legierungen des Zinns mit Kupfer oder Silber ersetzt.

Diese Legierungen haben jedoch einen höheren Schmelzpunkt als bleihaltiges Lot.

Es ist eine etwas höhere Löttemperatur erforderlich.

Gängiges bleifreies Lötzinn, z. B. vom Typ Sn99Cu1+ML, hat seinen Schmelzpunkt bei 227 °C.

Es gibt Lötendraht mit bereits eingearbeitetem Flußmittel oder solchen ohne.

Dann muß ein separates Flußmittel verwendet werden, z.B.Lötfett:



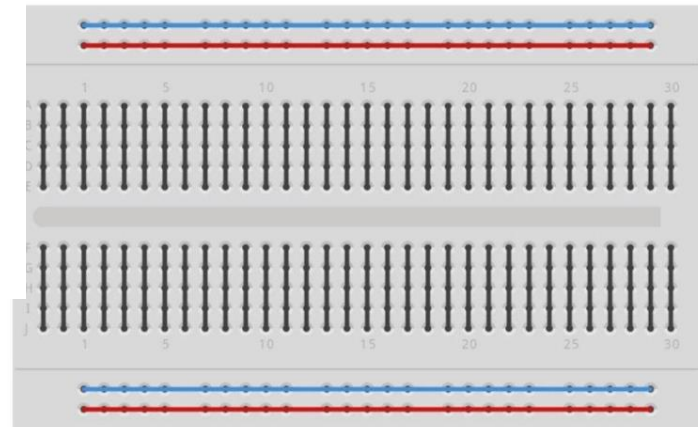
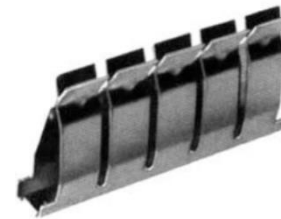
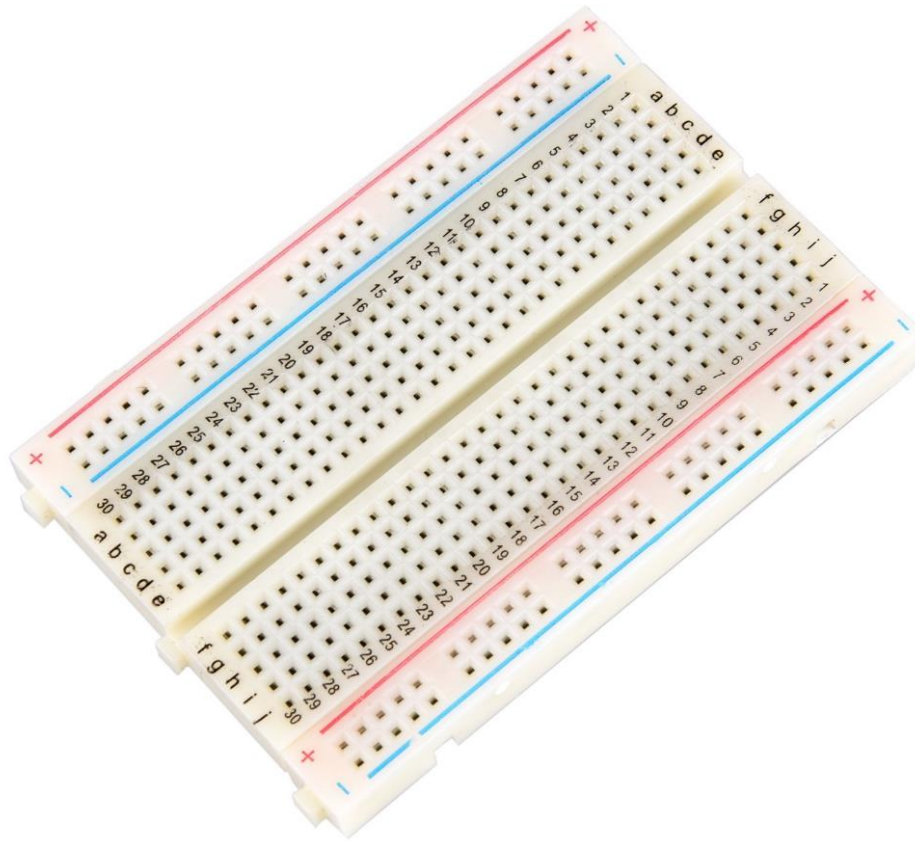
Lötfett darf nicht in die Augen gelangen.

Dämpfe beim Löten sollten nicht eingeatmet werden.

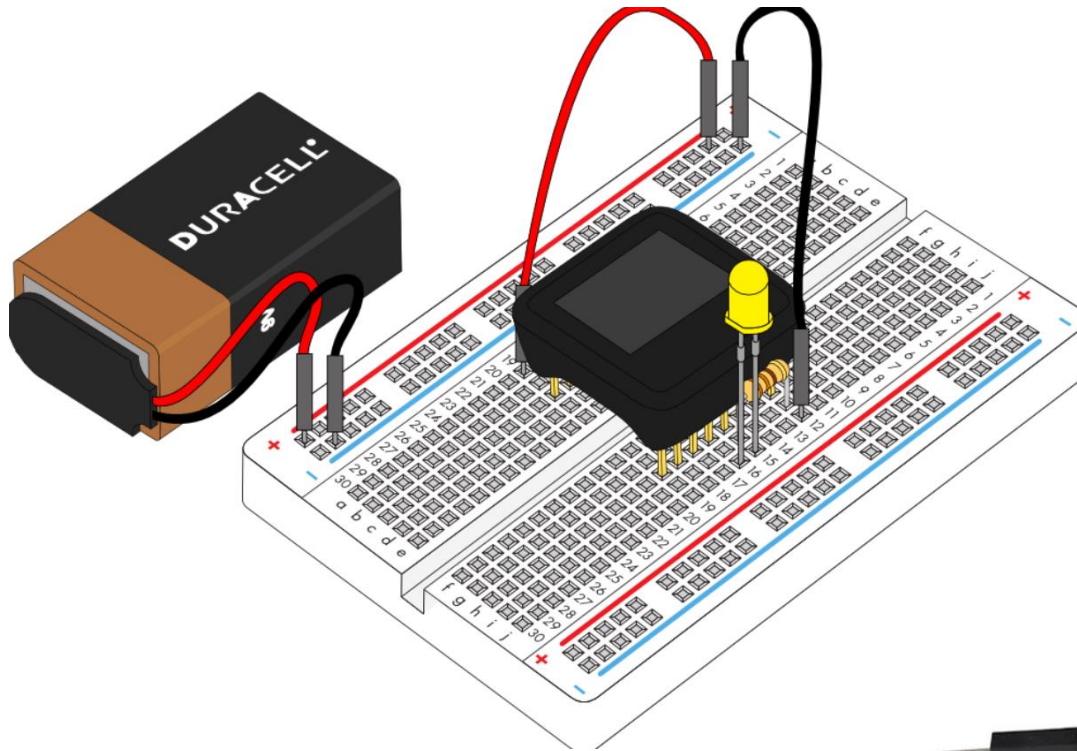




# Breadboard (Steckbrett)



# Breadboard und Jumper Wires (Verbindungsleitungen)



Male - Male

Female - Female

Male - Female





ESD (Electrostatic Discharge)  
Elektrostatische Entladung



Elektronik ist in  
leitfähigen Tüten verpackt

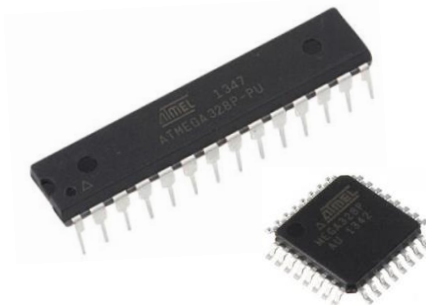
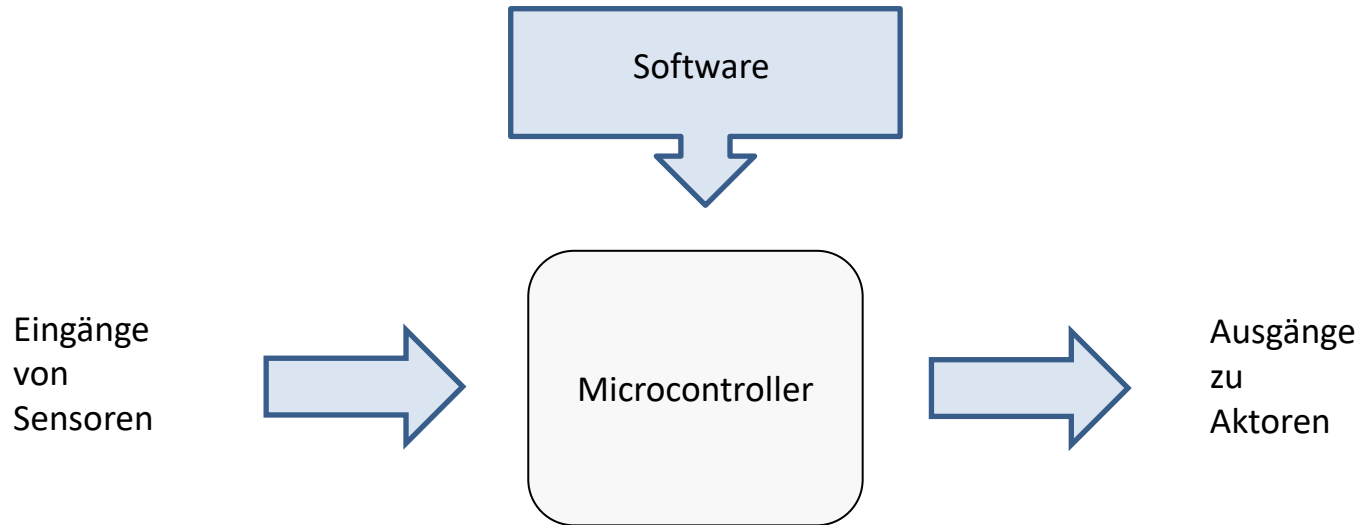


Wenn die Elektronik in Funktion ist  
nicht auf die leitfähige Tüte legen

# Mikroprozessor (Microcontroller)

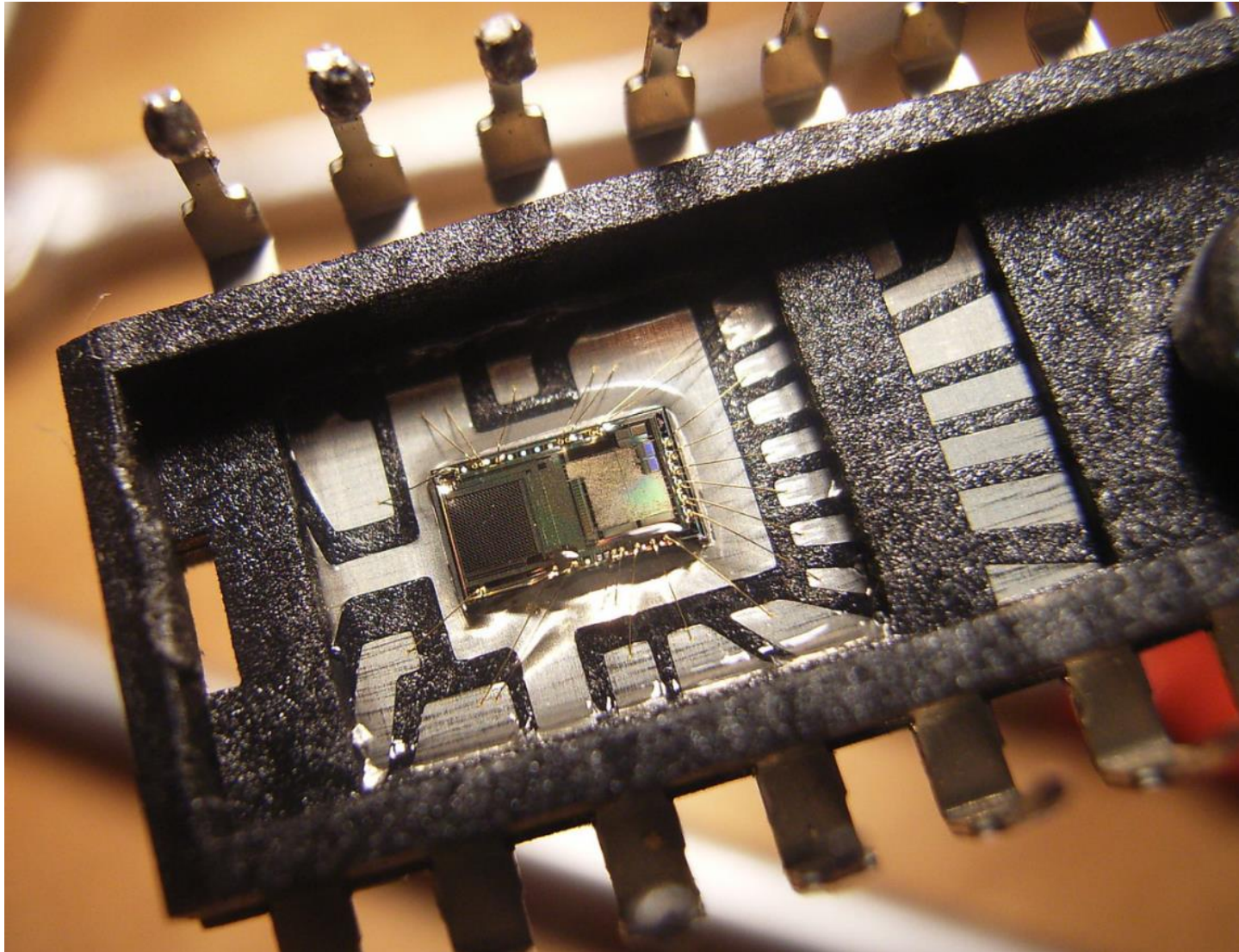
---

- Ein Mikroprozessor (englisch Microcontroller) ist eine elektronische Steuerung, die
  - auf einem Chip integriert ist
  - durch ein Programm (Software) aktiviert werden muß und geändert werden kann



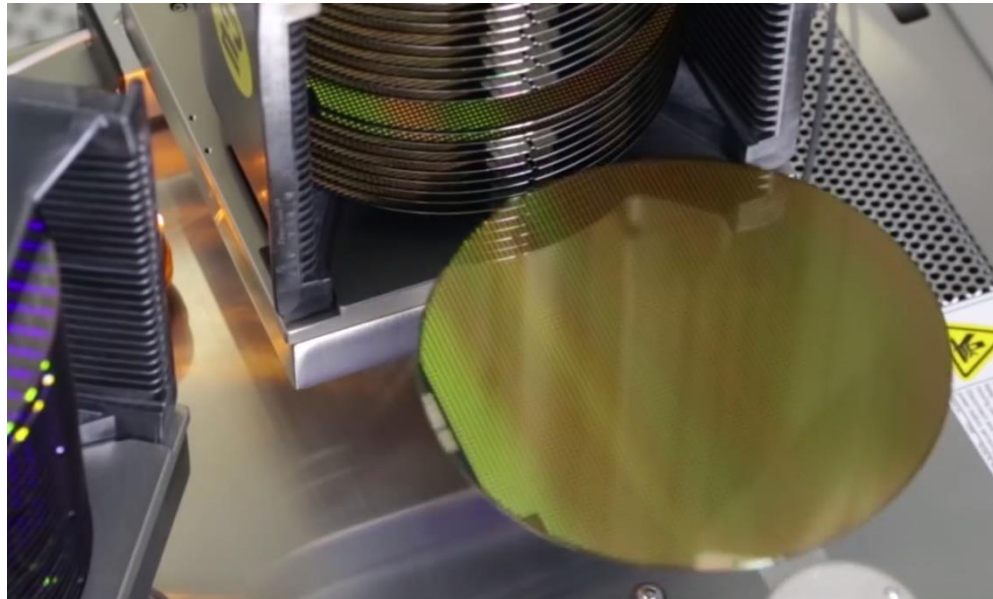
# Mikroprozessor (Microcontroller)

---



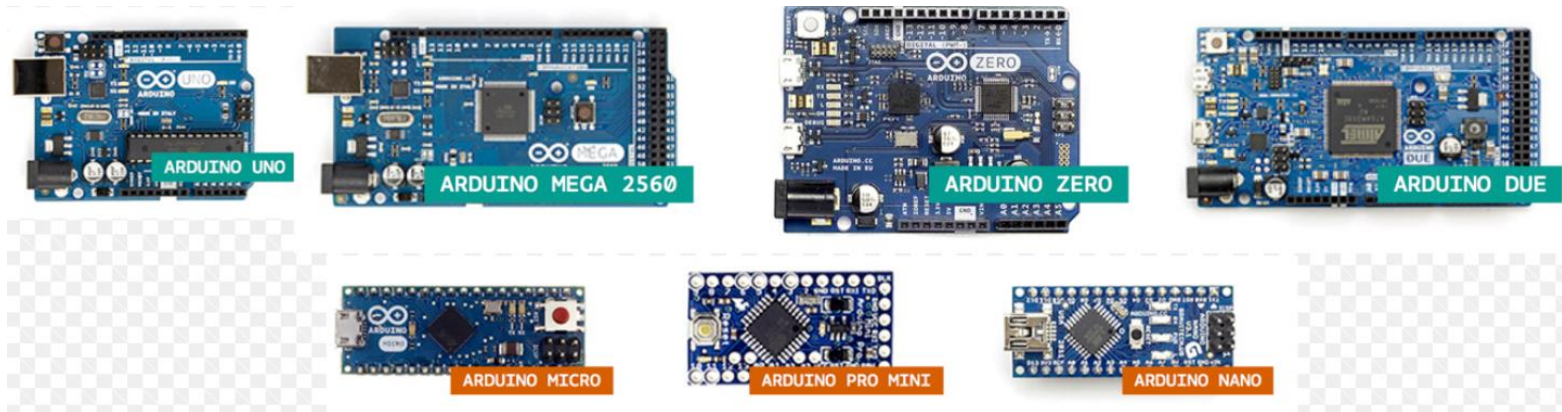
# Halbleiterfertigung

---



<https://www.youtube.com/watch?v=epk4NdyZHHc>

# Arduino Microcontroller Boards



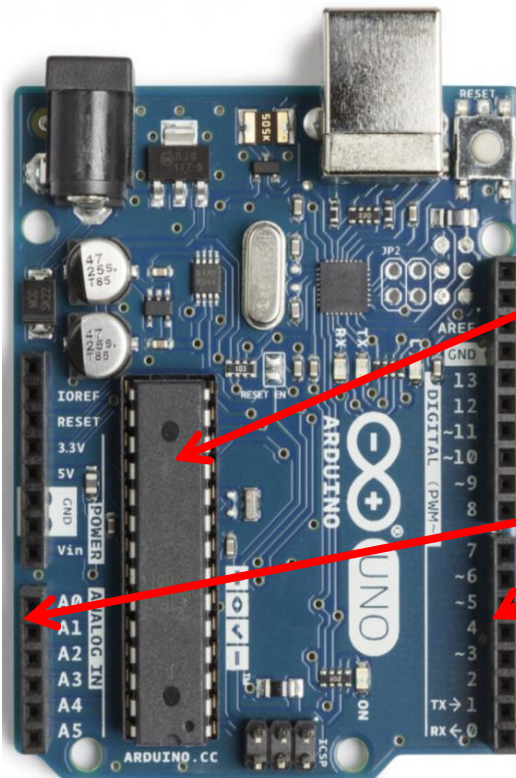
Entwickelt wurde es Ende 2005 von Prof. Massimo Banzi und David Cuartielles am *Interaction Design Institute Ivrea* (IDII) in Italien.

Namensgebend war ein Studentenlokal nahe des IDII, welche nach dem italienischen König Arduino (um 1000 n. Ch.) benannt wurde.

# Arduino Uno und Nano - Microcontroller-Boards

Uno

Software  
über USB



Nano

Software  
über USB

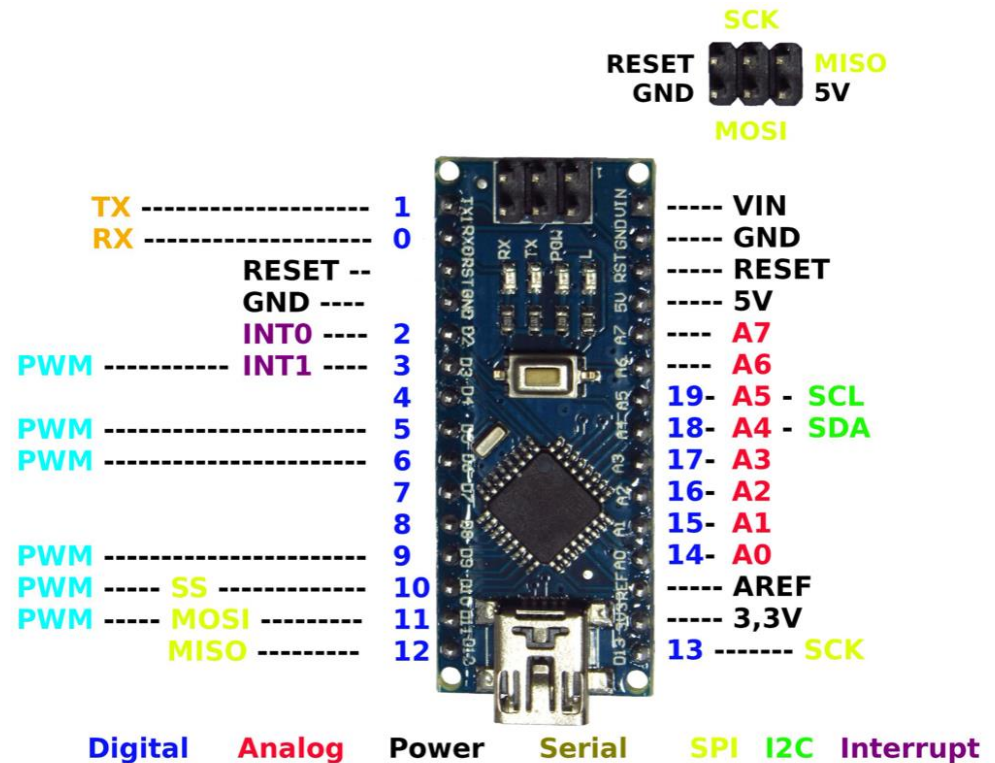
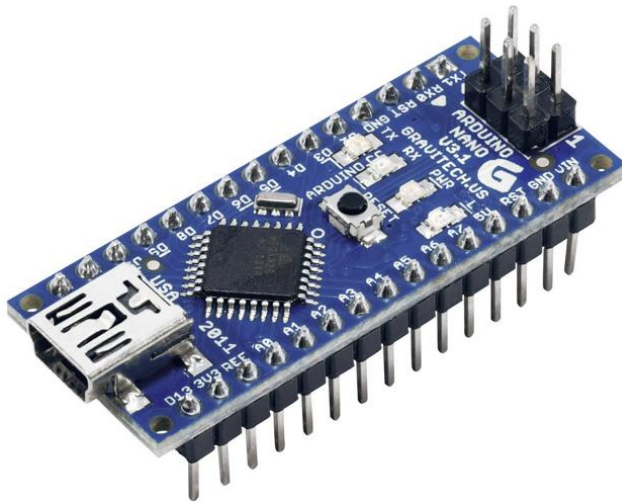


Mikrocontroller  
Atmega 328

Eingänge / Ausgänge  
Input / Output



# Arduino Nano



## Technische Daten:

Mikrocontroller: Atmel ATmega328  
 Takt (Clock): 16 MHz  
 USB-Chip: FTDI FT232RL oder CH340G  
 Eingangsspannung an VIN: 7-12 V

Digital Input/Output-Pins: 20 (davon 6 geeignet für PWM)  
 Analog Input Pins: 8 (10 Bit, d.h. 1024 Werte)  
 A0...A5 auch als Digital Input/Output D14...D19 programmierbar  
 Max. zulässiger Strom pro I/O-Pin: 40 mA  
 Max.zulässige Eingangsspannung an den Pins: 5 V

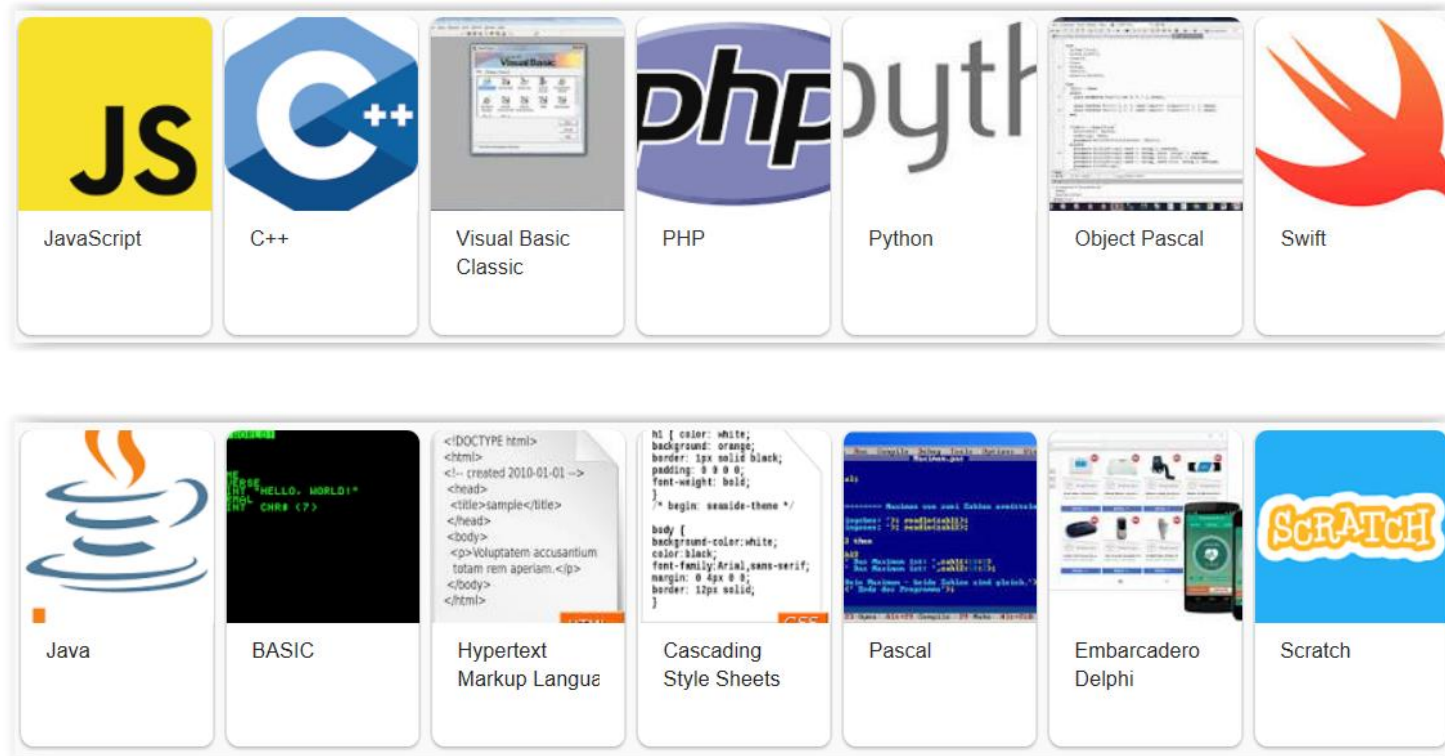


Was ist eine  
**PROGRAMMIERSPRACHE?**

<https://www.youtube.com/watch?v=g56vqSs4bS8>

# Programmiersprachen

Es gibt zahlreiche Programmiersprachen.



<https://lernprogrammieren.de/uebersicht-ueber-die-programmiersprachen/>

# Programmiersprachen - Scratch

Scratch ist eine visuelle Programmiersprache.

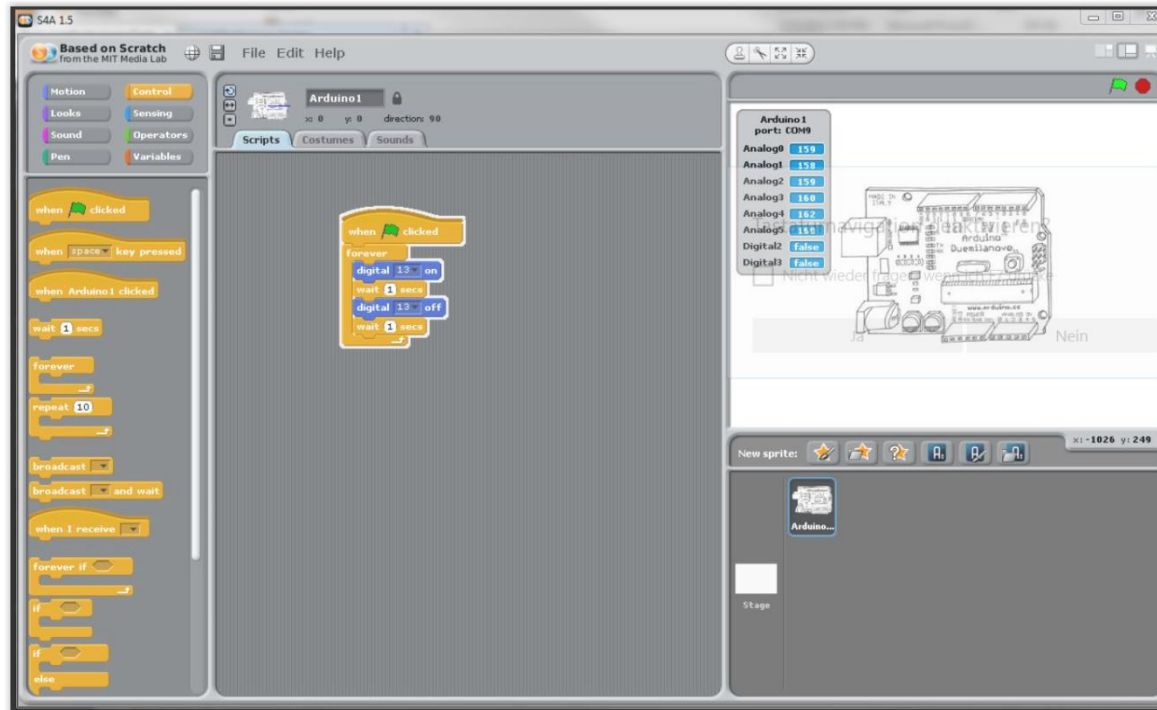
Es gibt vorgefertigte grafische Blöcke, die zusammengeschooben werden („Drag and Drop“).

Die Sprache ist englisch, kann aber auch auf deutsch eingestellt werden.



# Programmiersprachen - Arduino mit Scratch programmieren

Um den Arduino mit Scratch zu programmieren, gibt es eine angepasste Version namens „Scratch for Arduino“ (S4A). Sie stellt zusätzliche Blöcke zur Verfügung, die den Zugriff auf die Aktoren und Sensoren ermöglichen.



Das Programm läuft auf dem PC und wird nicht auf den Arduino hochgeladen, deshalb muß das USB-Kabel angeschlossen bleiben.

<http://s4a.cat/>

<https://michaelsarduino.blogspot.com/2015/07/arduino-mit-scratch-programmieren.html>

# Programmiersprachen - Python

---



Python ist im Gegensatz zu Scratch eine textorientierte Programmiersprache. Wegen seiner klaren und übersichtlichen Syntax gilt Python als einfach zu erlernen. So werden beispielsweise (Text)Blöcke nicht durch geschweifte Klammern, sondern durch Einrückungen strukturiert.

Python ist eine Interpreter-Programmiersprache.

Der Name hat mit der Python-Schlange nichts zu tun, sondern bezog sich ursprünglich auf die englische Komikertruppe Monty Python.

<http://python4kids.net/>

<http://python4kids.net/how2think/>

# Programmiersprachen – C und C++

---

Wir verwenden C++



C bzw. C++ ist ebenfalls eine textorientierte Programmiersprache.  
Sie ist ähnlich zu Python, jedoch eine Compiler-Programmiersprache.

Der Name stammt aus einer Vorgänger-Sprache mit der Bezeichnung B.

C ermöglicht eine maschinennahe Programmierung mit optimierten Programm-Laufzeiten.  
Der Mikrocontroller kann die Programme schneller abarbeiten, als wenn sie in Python oder in Scratch geschrieben sind.

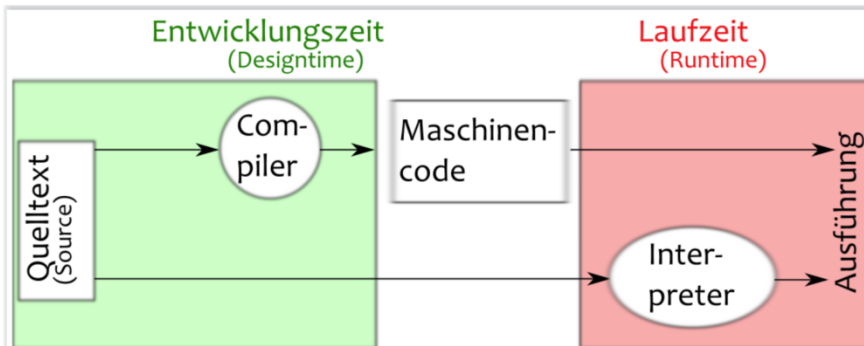
Die übliche Programmiersprache für den Arduino (Arduino IDE) verwendet ein etwas modifiziertes C++.

# Programmiersprachen - Compiler vs Interpreter

**Interpreter** sind Übersetzungsprogramme, die jeweils eine einzelne Befehlszeile des Programms übersetzen und ausführen. Die für dynamische Web-Seiten verwendbare Programmiersprache JAVA-SCRIPT benutzt beispielsweise einen Interpreter zur Übersetzung.

**Compiler** übersetzen das gesamte, in einer Programmiersprache erstellte Programm (Quellprogramm) in ein ausführbares Zielprogramm (Maschinensprache). Erst wenn das Programm komplett und fehlerfrei übersetzt wurde, kann es vom Benutzer ausgeführt werden.

Vor- und Nachteile der beiden Übersetzer sind in folgender Abbildung gegenübergestellt.



	Vorteile	Nachteile
<b>Interpreter</b>	<p>Eine Programmzeile kann sofort getestet werden.</p> <p>Fehler bei der Programmierung werden sofort erkannt und behoben.</p>	<p>Bei <u>jeder</u> Programmausführung muss das Programm erneut analysiert werden; daher längere Laufzeit.</p> <p>Es gibt kein Objektprogramm, das Quellprogramm benötigt mehr Speicherplatz.</p>
<b>Compiler</b>	<p>Programmanalyse muss nicht bei jeder Programmausführung vorgenommen werden.</p> <p>Das Objektprogramm benötigt weniger Ausführungszeit und weniger Speicherplatz.</p>	<p>Mehrere Schritte sind erforderlich, um ein Programm auszuführen.</p> <p>Bei jedem Übersetzungsvorgang (z. B. nach einer Fehlerbereinigung) muss das gesamte Programm erneut übersetzt werden.</p> <p>Fehlersuche bei der Programmentwicklung ist meist aufwändiger.</p>



# Compiler

Quellprogramm  
(Arduino-sketch)

```
#include<stdio.h>

int main() {
printf("Hello ");
printf("World\n");
return 0;
}
```



Compiler



Zielprogramm  
(binärer Maschinencode)

```
01001001001110001
11000100010101001
00101001001001110
01010111000101001
00100010111010010
00101010101110001
01010101011100111
10101011000111011
```



Zielprogramm  
(binärer Maschinencode)

```
01001001001110001
11000100010101001
00101001001001110
01010111000101001
00100010111010010
00101010101110001
01010101011100111
10101011000111011
```



# Compiler

Der Compiler übersetzt nicht nur das Quellprogramm in den Maschinencode, er prüft und erkennt auch Fehler, die beim Schreiben des Quellprogramms gemacht wurden:

```
void setup()  
{  
}  
  
void loop()  
{  
digitalWrite(LED, LOW);  
}
```

'LED' was not declared in this scope



```
int LED=13;  
  
void setup()  
{  
}  
  
void loop()  
{  
digitalWrite(LED, LOW);  
}
```

Kompilieren abgeschlossen.

# Quellprogramm („Arduino-sketch“) und IDE

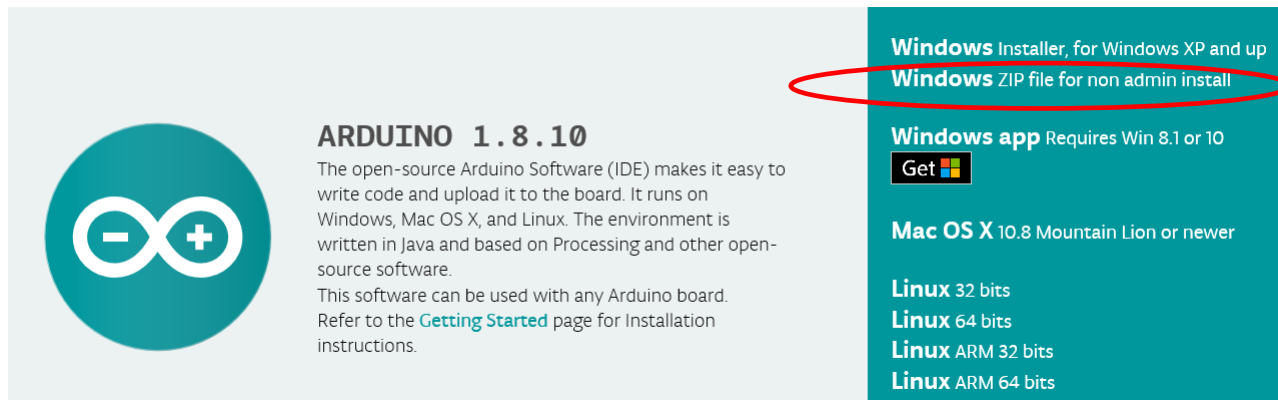
---

- Das Quellprogramm wird mit der Programmiersprache C++ geschrieben.
- Für den Arduino wurde die sogenannte IDE Integrated Development Environment (Integrierte Entwicklungsumgebung) geschaffen.
- Die Arduino-IDE besteht im wesentlichen
  - aus einem Editor für den Quelltext
  - dem Compiler zum Übersetzen in den Maschinencode
  - Bibliotheksprogrammen (Library)
- Zusätzlich hat man die Möglichkeit, sich Meldungen des Arduino-Boards auf dem Computer anzeigen zu lassen.
- Die IDE kann von der Arduino-Webseite auf den Computer heruntergeladen werden.

<https://www.arduino.cc/en/Main/Software>

# Arduino IDE - Download

---



**ARDUINO 1.8.10**

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.

This software can be used with any Arduino board. Refer to the [Getting Started](#) page for Installation instructions.

**Windows** Installer, for Windows XP and up  
**Windows** ZIP file for non admin install


**Windows app** Requires Win 8.1 or 10  
**Get**

**Mac OS X** 10.8 Mountain Lion or newer


**Linux** 32 bits  
**Linux** 64 bits  
**Linux** ARM 32 bits  
**Linux** ARM 64 bits

Wenn man keine Administratorrechte hat, sollte angeklickt werden „Download ZIP file for non admin rights“.

Der ZIP file („gepackt“ oder „komprimiert“) wird heruntergeladen und muß dann noch entpackt werden.

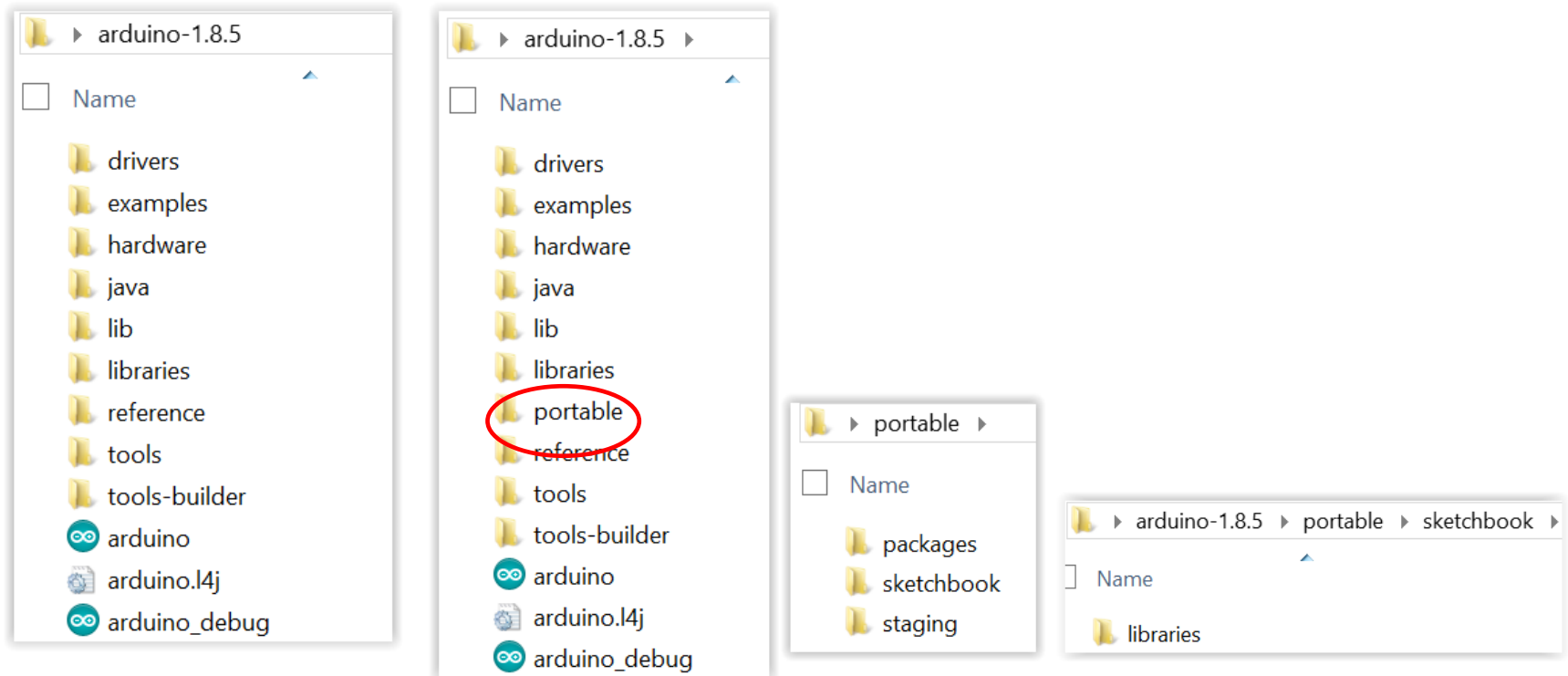
Man erhält einen Ordner  **arduino-1.8.10** , der an beliebiger Stelle abgespeichert werden kann, z.B. auf einem USB-Stick .

Die Ordnergröße beträgt ca. 480MB.

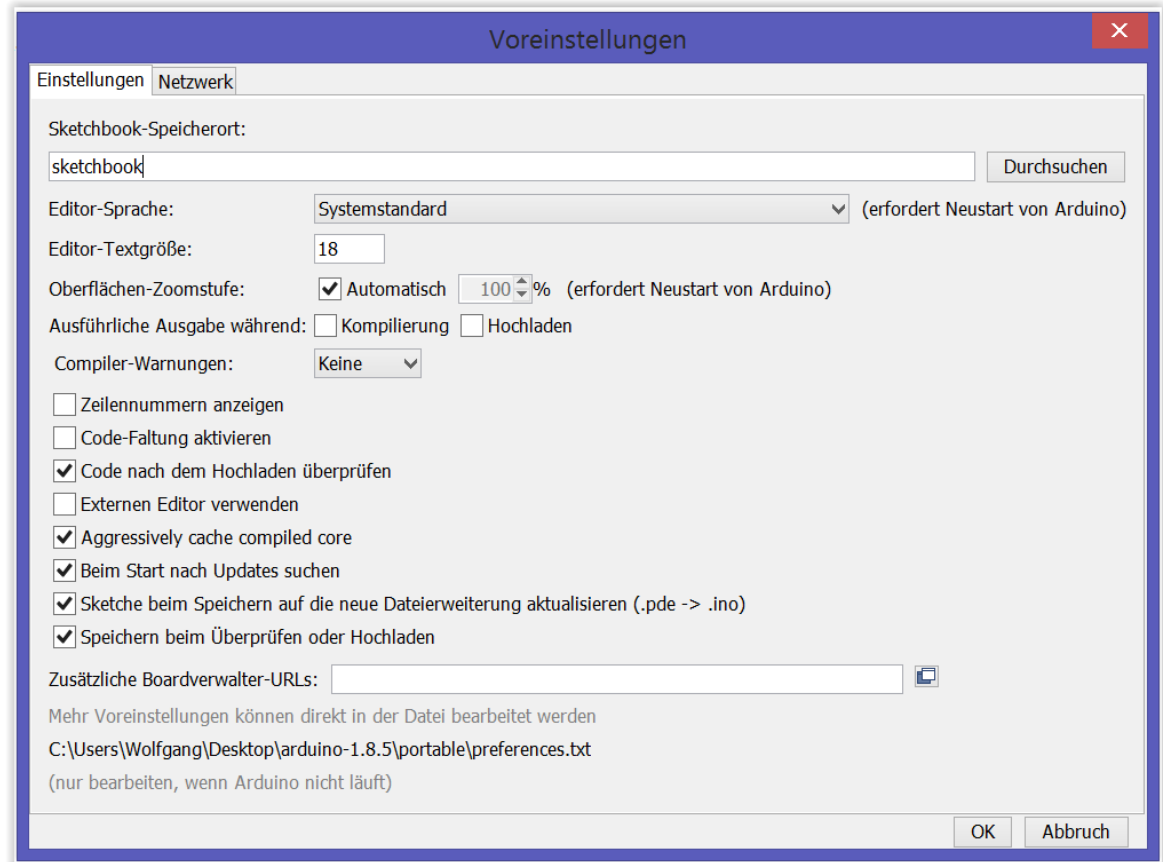
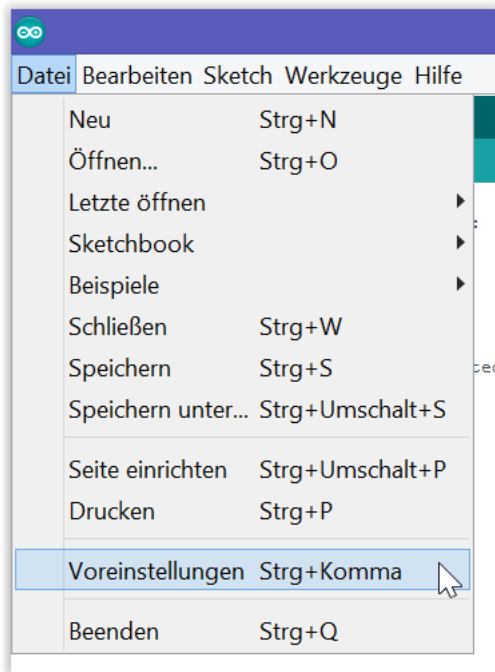
Wenn der Ordner auf C:\  **Programme (x86)** abgespeichert werden soll, benötigt man Administratorrechte.

# Arduino IDE portable

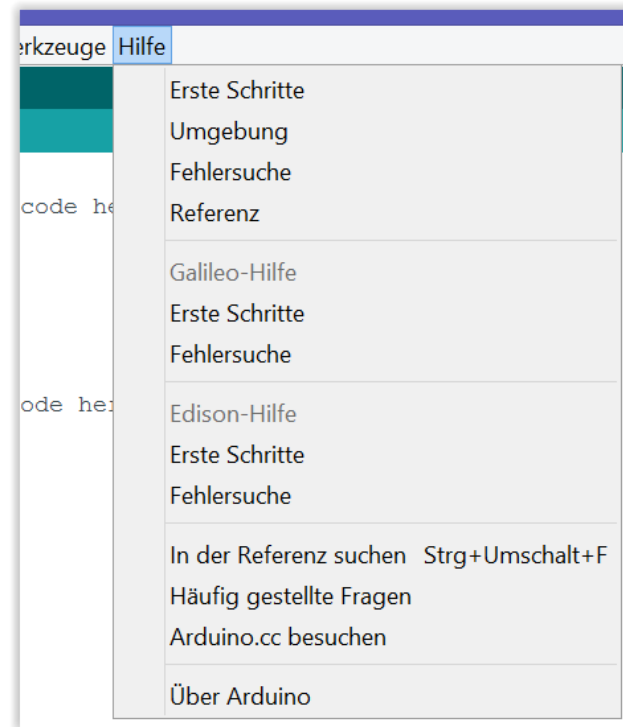
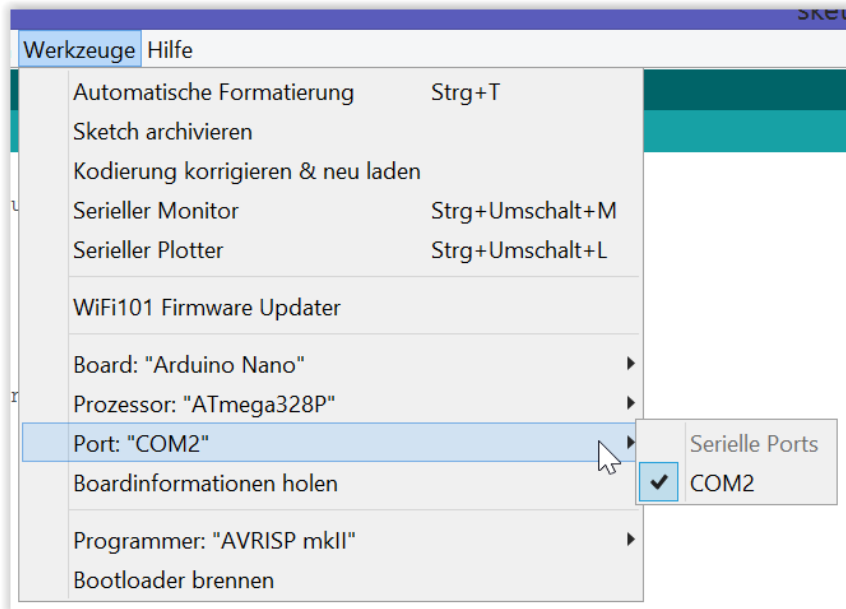
- <https://www.arduino.cc/en/Guide/PortableIDE>
- Wir legen einen einen zusätzlichen Ordner (folder) an, namens „portable“, mit einigen Unterordnern (subfoldern):
- Alle neuen Programme (sketches) und alle zusätzlichen Bibliotheksprogramme (libraries) sowie die Einstellungen (preferences) werden automatisch nur an diesem Ort gespeichert.
- Vorteil: Wenn der Ordner „arduino“ kopiert und auf einen anderen Computer übertragen wird, wird alles mit erfasst.



# Voreinstellungen (preferences)



# Auswahl des Boards und COM-Ports / Hilfe



Eventuell einstellen  
(siehe Anhang):



Den COM-Port erst prüfen, wenn der Arduino an einer USB-Buchse angesteckt ist.  
Windows weist dem USB-Adapter automatisch einen COM-Port zu.  
Die Zuordnung ist dynamisch, das heißt sie kann sich ändern nach Trennen und erneutem Anschluß.  
Nach jedem erneuten Anschluß des Arduino sollte man prüfen, ob noch der richtige Port eingestellt ist.

**Achtung:** Wenn ein Board mit CH340 verwendet wird – siehe Anhang „USB to UART Converter mit CH340“

# Arduino IDE – Ansicht

Aufruf des Seriellen Monitors (Serial Monitor)

Kompilieren (compile)

Hochladen (upload)

Dateiname (sketch name)

Quelltext (source code)

Meldung des Compilers

```
void setup()
{
  // put your setup code here, to run once:
}

void loop()
{
  // put your main code here, to run repeatedly:
}
```

Kompilieren abgeschlossen.

Archiving built core (caching) in: C:\Users\Wolfgang  
Der Sketch verwendet 444 Bytes (1%) des Programmspe  
Globale Variablen verwenden 9 Bytes (0%) des dynamis

2 Arduino/Genuino Uno auf COM10

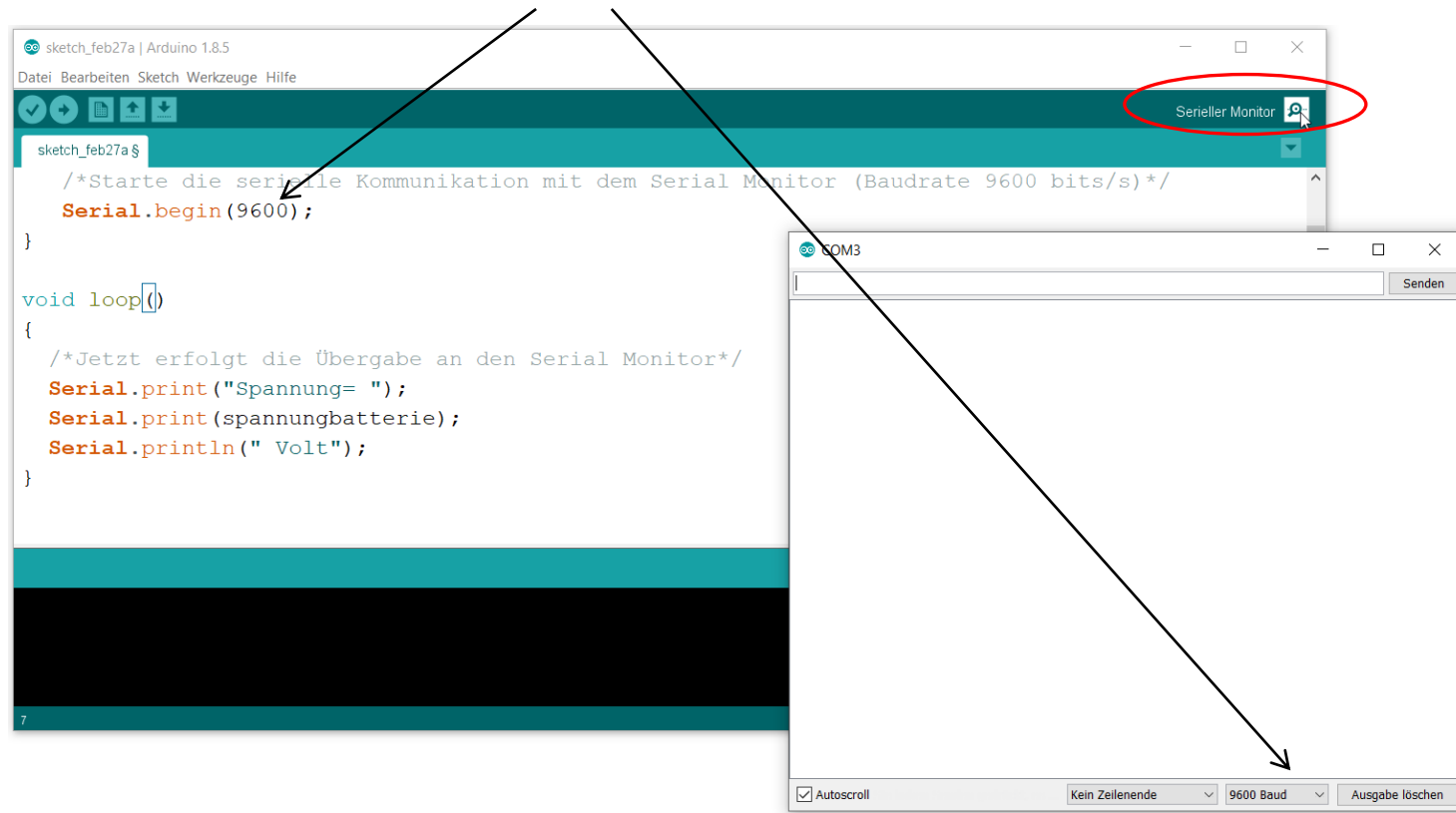
Arduino Uno, angeschlossen am Computer an Port COM10



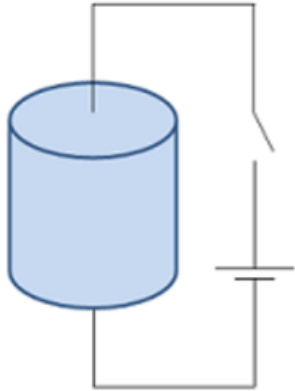
# Serial Monitor

Beachte:

- Bei jedem Aufruf des Serial Monitor wird ein Reset erzeugt, d.h. der Sketch startet neu (z.B. fährt das Auto neu los)
- Bei Arduino, aber nicht bei ESP8266 (siehe Teil 2)
- Es muß dieselbe Übertragungsrate eingestellt sein

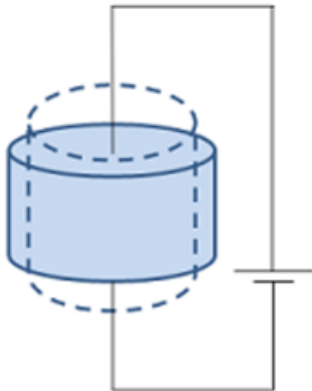


# Piezo-Signalgeber (Buzzer)



Das Wort „Piezo“ ist vom griechischen Wort für Druck abgeleitet.

1880 entdeckten Jacques und Pierre Curie, dass Druck in verschiedenen Kristallen wie Quarz und Turmalin elektrische Ladungen erzeugt; sie nannten dieses Phänomen den „Piezoeffekt“.



Später stellten sie fest, dass elektrische Felder piezoelektrische Materialien verformen können. Dieser Effekt heißt „inverser Piezoeffekt“.

Durch Anlegen einer Wechselspannung kommt der Kristall in Schwingungen und sendet ein akustisches Signal aus. Die Tonhöhe hängt von der Frequenz der Wechselspannung ab.



# Ultraschall-Distanzmessung

**Ultraschall** ist ein Hochfrequenz- Ton (meistens Frequenz 40 KHz).  
Ein kurzer Stoß von Tonwellen wird durch den Signalgeber (Trigger) ausgegeben.  
Dann lauscht der Messempfänger auf das Echo.

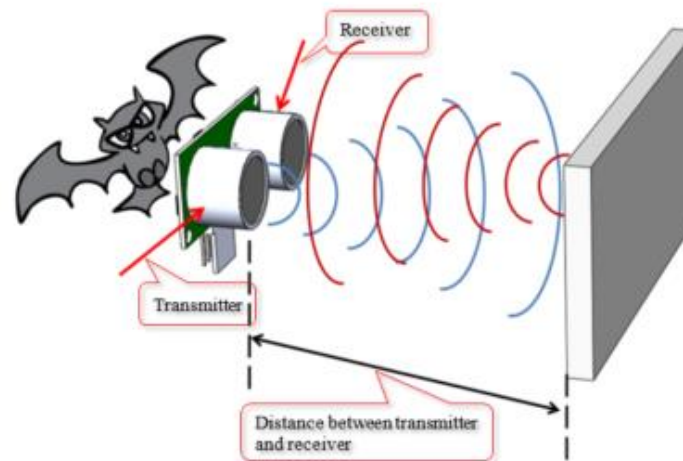
Die Ausbreitungsgeschwindigkeit von Ultraschall in Luft beträgt bei einer Lufttemperatur von 20°C etwa 343 m/s.

Zur Entfernungsmessung:

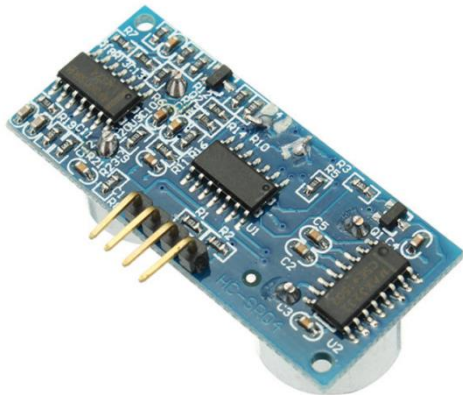
Pro gemessener Mikrosekunde  $\mu\text{s}$  wäre der Schall also 0,034 cm unterwegs gewesen.

Um die gemessene Zeit des Echo-Signals nun in cm umzurechnen, wird sie zunächst durch 2 geteilt, um die Zeit für die einmalige Strecke zu erlangen.

Danach müssten wir die Zeit (in Mikrosekunde) mit 0,03434 cm multiplizieren und wir hätten die Entfernung in cm.



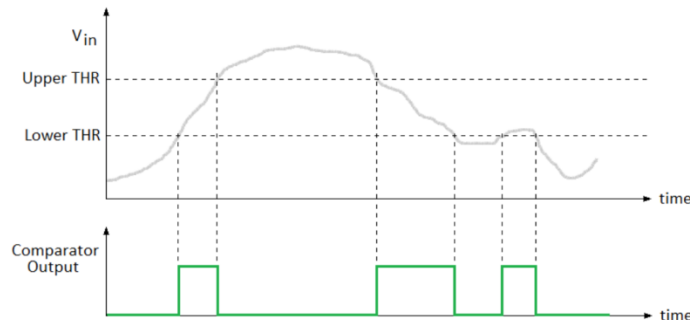
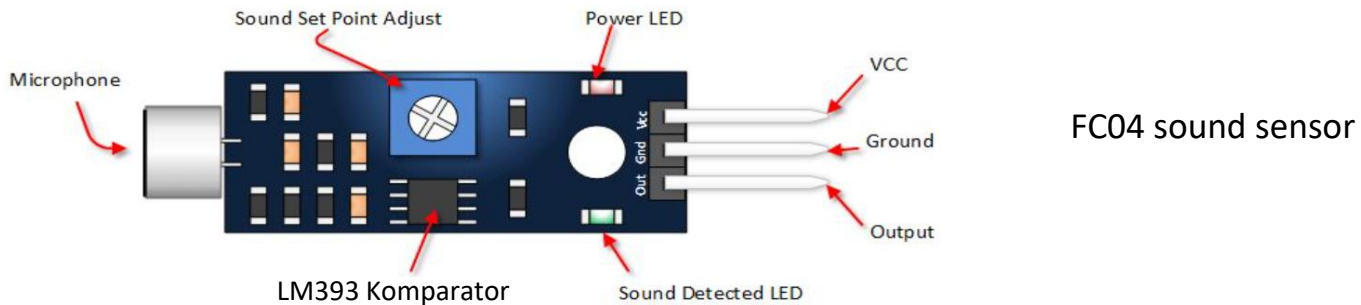
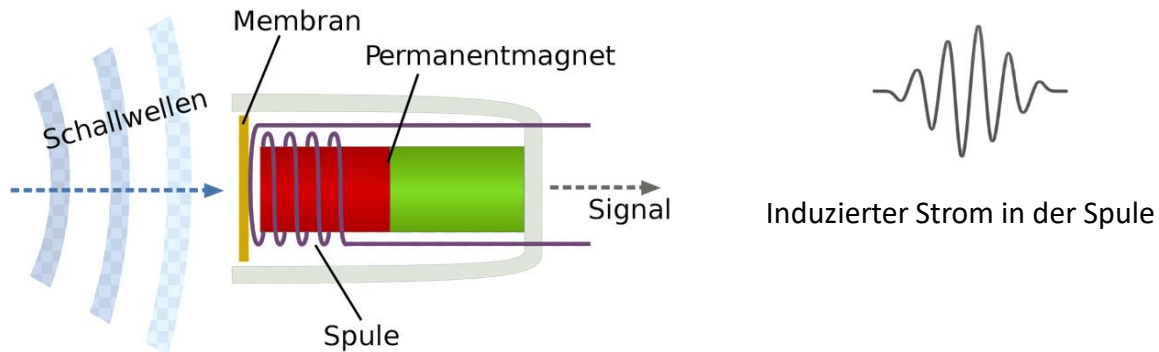
# Ultraschallsensor HC-SR04



Ultraschallsensor HC-SR04	
Betriebsspannung	5V (+/- 10%)
Strombedarf	ca. 2mA pro Messung
Signal Level	TTL-Pegel
max. messbare Entfernung	ca. 3m
min. messbare Entfernung	ca. 2 cm
Maximalen Messungen pro Sekunde	50
Ultraschallkapseln	zwei (Sender und Empfänger)
Genauigkeit	ca. 3mm
Pinbelegung	Pin 1: VCC Pin 2: Trigger Pin 3: Echo Pin 4: GND

$$\text{Distance (cm)} = \text{Measured Echo Time (in } \mu\text{sec)} / 58$$

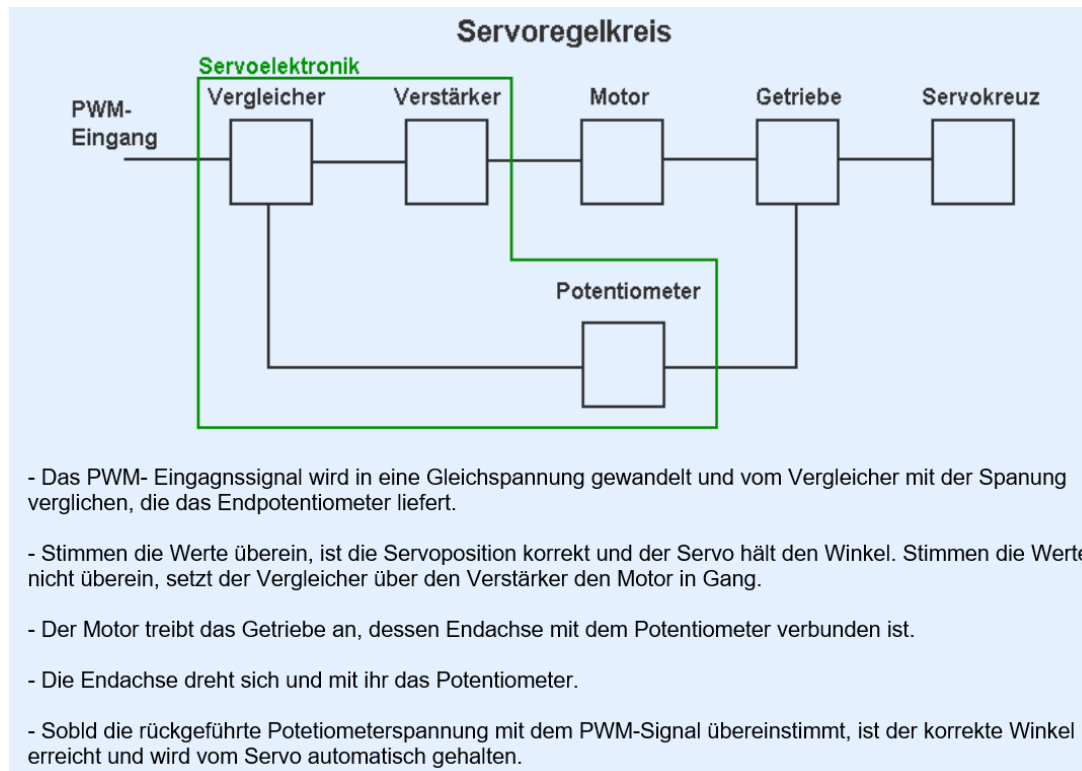
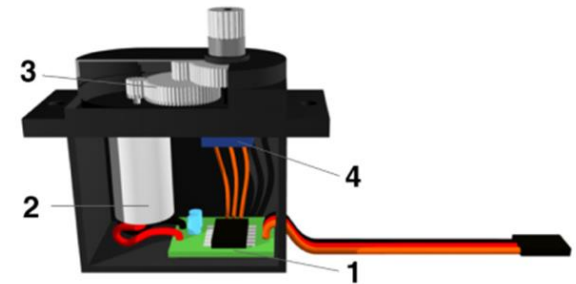
# Mikrofon – Geräuscherkennung mit FC04



Ein Komparator (von engl. "compare" = vergleichen) ist eine Schaltung, die zwei Eingangsspannungen miteinander vergleicht. Abhängig davon, welche der beiden Spannungen größer ist, wird am Ausgang entweder ein logisches „High“ oder „Low“ geliefert.

# Servo

Ein Servo besteht aus einer Motorsteuerung (1), einem Elektromotor (2), einem Getriebe (3) und einem Potentiometer zur Positionsbestimmung (4). Alle Komponenten sind in einem robusten Gehäuse untergebracht.



# Servo ansteuern

Zum Steuern des Servomotors verwendet man ein sich alle 20 Millisekunden wiederholendes Signal. Es besteht aus einem HIGH-Impuls, der zwischen etwa 0,6 und 2,6 Millisekunden lang ist, und einem LOW-Impuls. Die Dauer des HIGH-Impulses bestimmt den Zielwinkel (normaler Weise von 0 bis 180°).

In der Arduino-Software kann hierfür der Befehl `delayMicroseconds()` verwendet werden.

Da der Servo einige Zeit benötigt, um sich auf den gewünschten Zielwinkel einzustellen, muss das Signal mindestens so lange wiederholt werden, bis der Servo die Position erreicht hat.

Servomotoren überprüfen ihren eigenen Stellwinkel mit einem eingebauten Potentiometer.

Es kommt hierbei vor, dass eine gewünschte Position nicht exakt erreicht werden kann.

Der Servo korrigiert sich dann kontinuierlich selbst, was zu einem »Zittern« des Servos führt.

Um das zu verhindern, sollte der Impuls nach Erreichen des Zielwinkels auf LOW geschaltet werden.

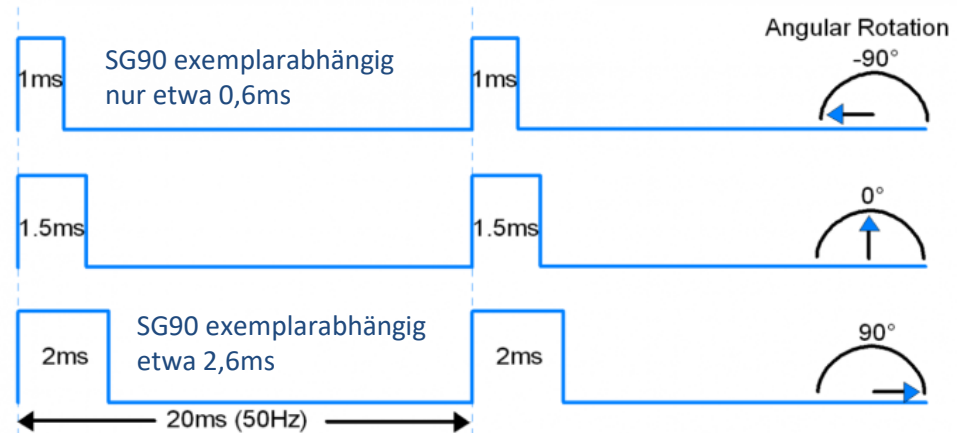


Nicht gegen einen mechanischen Anschlag bewegen – das Getriebe könnte beschädigt werden.

Siehe auch:

<https://www.youtube.com/watch?v=yQMcr3PNxV8>

<https://howtomechatronics.com/how-it-works/how-servo-motors-work-how-to-control-servos-using-arduino/>



# Servo SG90 und MG996R



Servo SG90 – für geringe Belastung  
Getriebe-Zahnräder aus Kunststoff  
Drehmoment bei Blockierung: 18 Ncm (bei 5V)  
Drehzahl: 90° in 0,15 s (bei 5V)  
Betriebsspannung: 4.2 - 6V



Servo MG996R – für hohe Belastung  
Größerer Motor, Getriebe-Zahnräder aus Metall  
Drehmoment bei Blockierung: 94 cNm (bei 4,8V)  
110 vNm (bei 6,0V)  
Strom bei Blockierung: 2,5 A (bei 6,0V)  
Drehzahl, ohne Last: 90° in 0,25 s (bei 4,8V)  
90° in 0,21 s (bei 6,0V)  
Betriebsspannung: 4.8 – 7,2V

Anschlüsse bei beiden Servos:





# HJ Digital Servo Tester

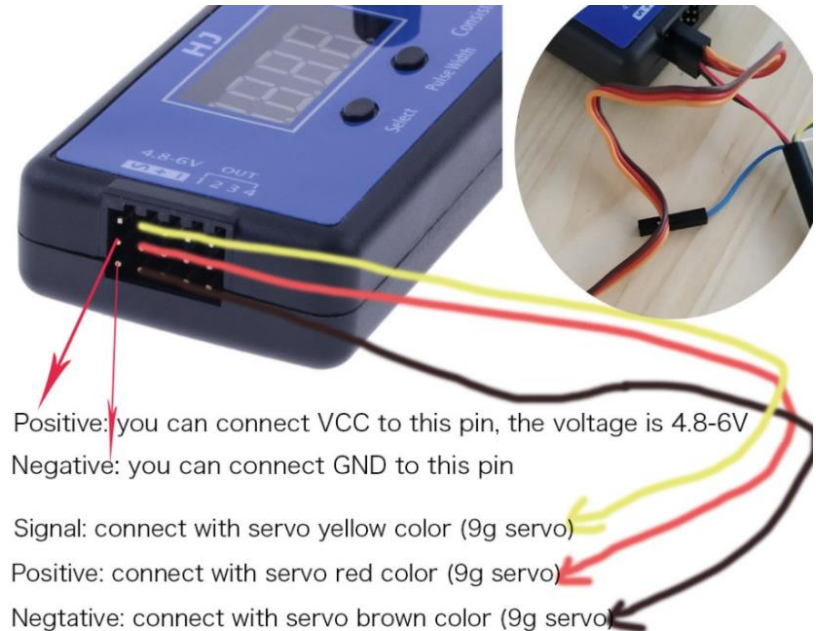
Mit dem Tester können bis zu 4 Servos angesteuert werden, ohne Arduino:

- Drehknopf- Einstellung des Winkels (PWM-Signal 0,8ms ...2,2ms)
- Automatisches Schwenken, Drehknopf verändert die Geschwindigkeit
- Einstellung Mittelposition (PWM-Signal etwa 1,5ms)

Ausserdem ist die Frequenz des Signals einstellbar 50Hz, 125Hz und 250Hz  
(50Hz entspr 20ms sollte verwendet werden)

Siehe auch:

<https://www.youtube.com/watch?v=7nuvbHuLU60>



This servo tester can connect with 4 servos at the same time

# Drahtlose Fernsteuerung – Wireless Remote Control

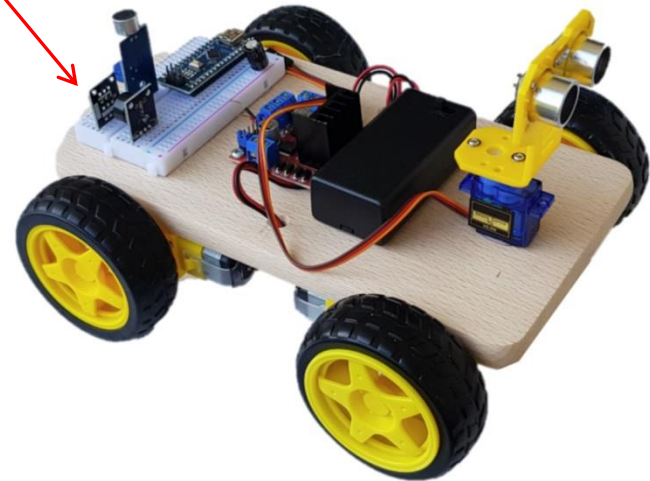
- Infrarot (IR)



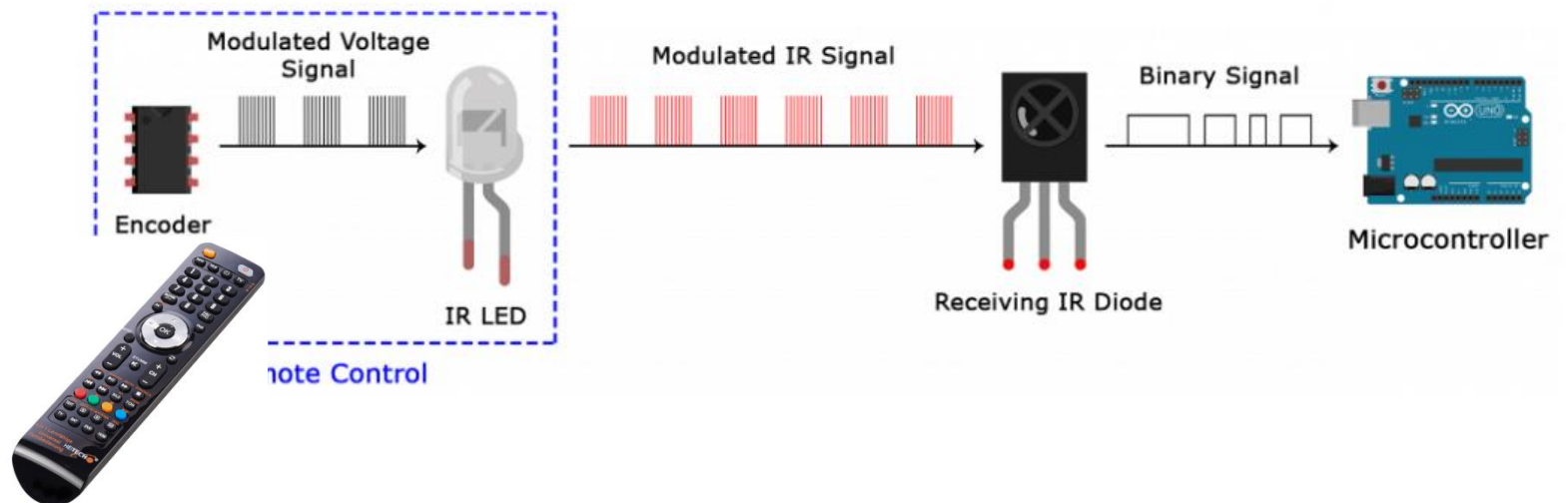
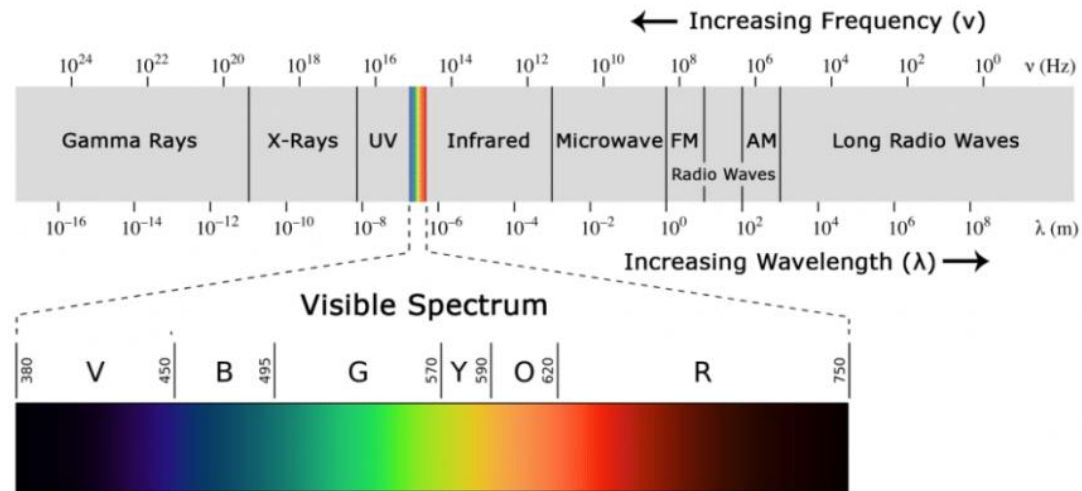
- Bluetooth



- WLAN

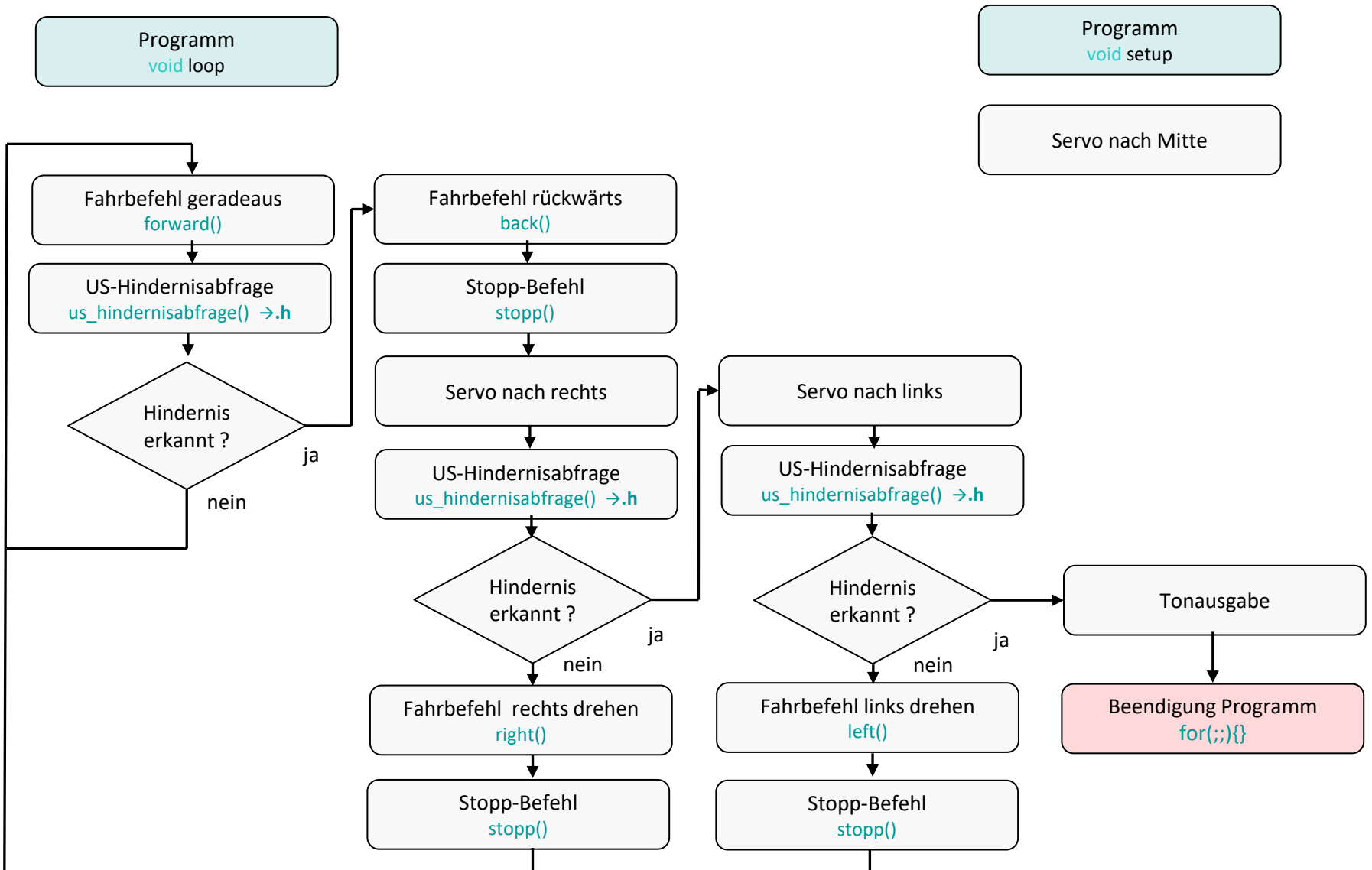


# IR Fernsteuerung



# Programmablaufplan Sketch 28

(Sketch nicht auf Website)



# Sketche

[Inhaltsverzeichnis](#)

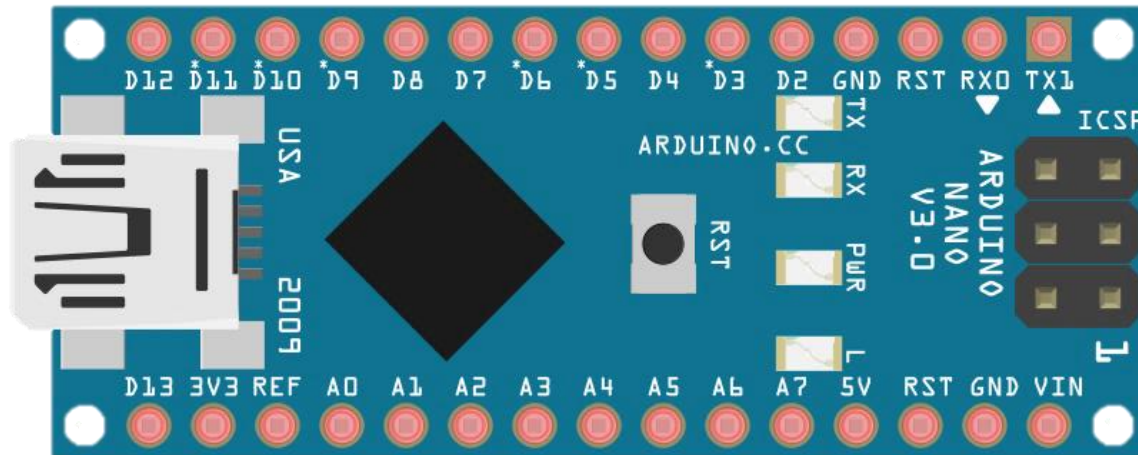
# Zuordnung Arduino Nano Pins

Nano	Funktion	Eing/Ausg			
D11~	Motoren links Brücke	Ausgang	L298N	INA1	
D8	Motoren links Brücke	Ausgang	L298N	INA2	
D10~	Servo Greifer	Ausgang		Wenn Bibliotheksprogramm Servo.h verwendet wird - reserviert	
D7	Motoren rechts Brücke	Ausgang	L298N	INA3	
D6~	Motoren links PWM	Ausgang	L298N	ENA	PWM Ausgang erford (analogWrite)
D5~	Motoren rechts PWM	Ausgang	L298N	ENB	PWM Ausgang erford (analogWrite)
D2 *	Ultraschall Trigger	Ausgang	HC-SR04		
A5=D19	Ultraschall Echo	Eingang	HC-SR04	Funktioniert nicht mit D0 oder D1 !	
D4	Buzzer Signal	Ausgang	Buzzer		
D9~	Servo	Ausgang	Servo SG9	PWM Ausgang erford; Servo.h benötigt D9	
D12	IR Empfänger	Eingang	V1838B	Funktioniert nicht mit D0 oder D1 !	
D13	Anzeige an LED „L“	Ausgang			
D3~*	Mikrophon	Eingang		Interrupt SR1	
A3=D17	Motoren rechts Brücke	Ausgang	L298N	INA4	
A0	Batteriespann. messen	Eingang		Von 9V über Spannungsteiler	

~ Ausgang PWM möglich


\* Eingang Interrupt möglich (D2...INT0 und D3...INT1)

# Anschluß an den Computer – erster Sketch



Nach Anschluß des USB-Kabels:

LED „PWR“ (oder „POW“) leuchtet rot  
LED „L“ flackert, geht dann wieder aus

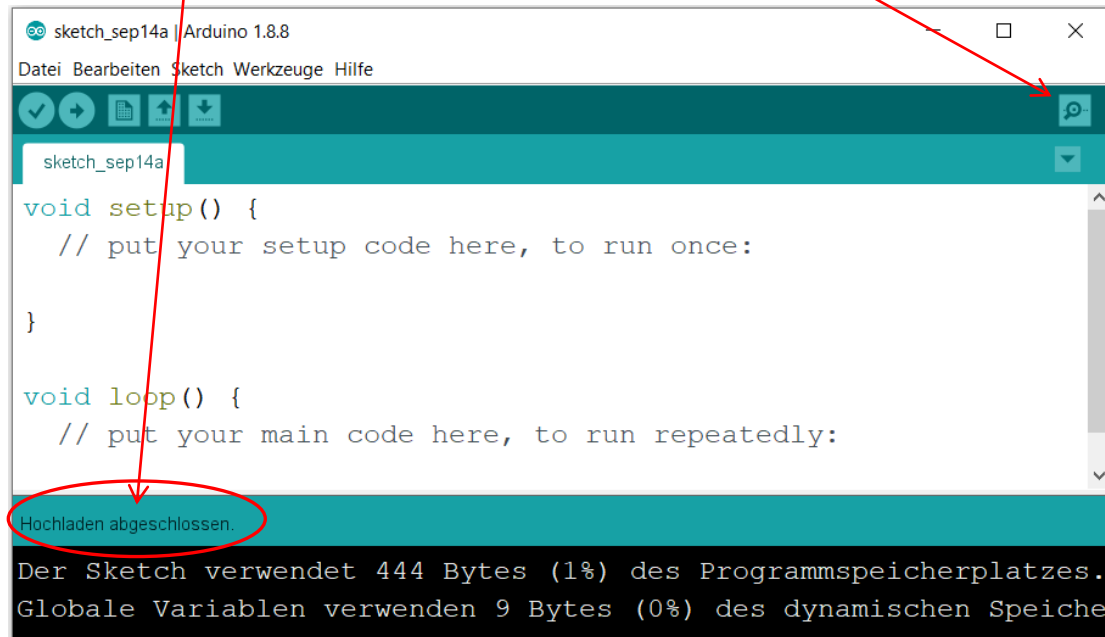
Upload sketch  :  
LED „L“ flackert einmal lang  
LEDs „RX“ und „TX“ flackern kurz für einige Sekunden

LED „L“ ist mit D13 verbunden (auf UNO als auch NANO)

# Start des sketches

Der sketch startet:

- Nachdem er hochgeladen wurde (nach Meldung „Hochladen abgeschlossen“)
- Erneut , wenn der Serial Monitor angeklickt wird
- Erneut , wenn der Reset-Knopf auf dem Arduino gedrückt



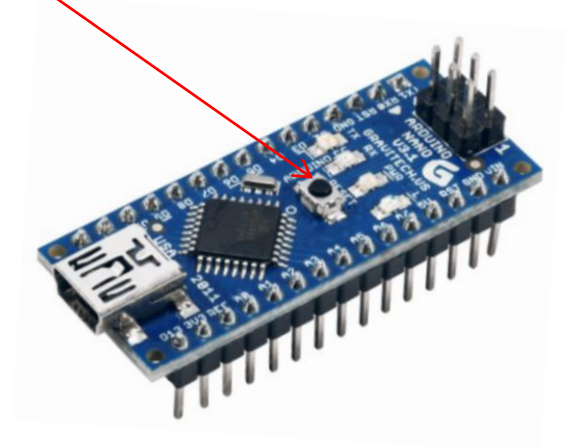
```
sketch_sep14a | Arduino 1.8.8
Datei Bearbeiten Sketch Werkzeuge Hilfe

sketch_sep14a
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}

Hochladen abgeschlossen.

Der Sketch verwendet 444 Bytes (1%) des Programmspeicherplatzes.
Globale Variablen verwenden 9 Bytes (0%) des dynamischen Speicherplatzes.
```





# Aufbau eines Sketches (eines Quellprogramms)

Genereller Aufbau eines Arduino-Sketches:

Ein Programm („sketch“) für den Arduino besteht aus drei Blöcken.

## Block 1

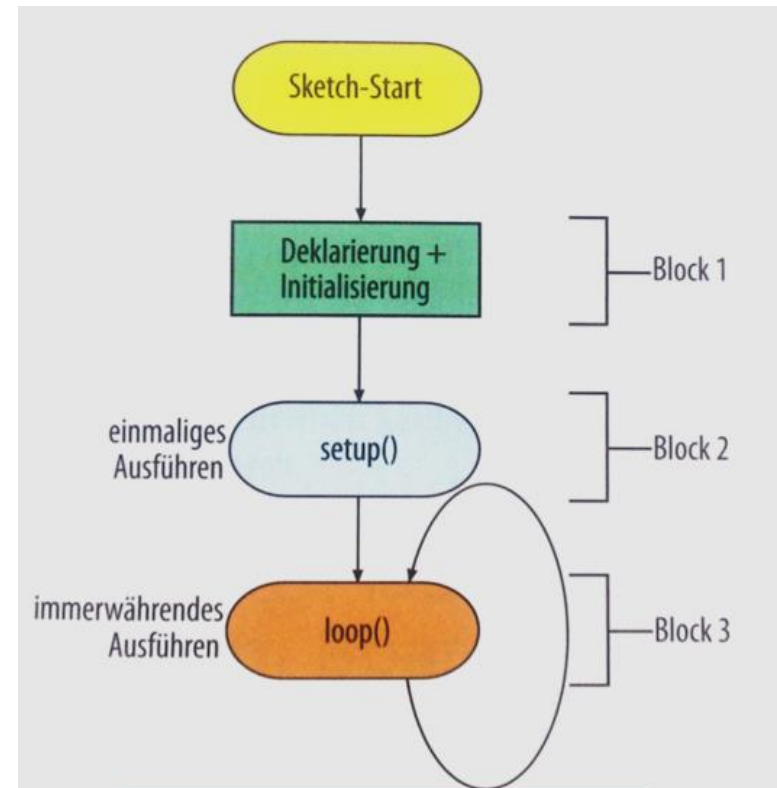
Im ersten Block weist man Variablen einen Datentyp zu, die dann im gesamten restlichen Programm verfügbar sind. Hier werden auch eventuell benötigte externe Bibliotheken mittels `#include` eingebunden.

## Block 2

Im zweiten Block, der `setup()`-Funktion, definiert man grundsätzliche Einstellungen, wie etwa welche Anschlüsse des Arduino als Eingang bzw. Ausgang verwendet werden sollen. Sie wird einmal ausgeführt und im restlichen Programm nicht mehr durchlaufen.

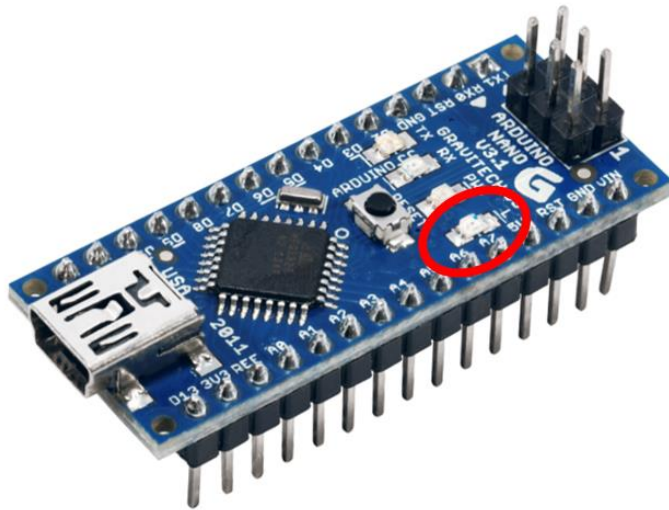
## Block 3

Der dritte und meist der Hauptteil des Sketches ist die `loop()`-Funktion. Sie wird in einer Endlosschleife wiederholt.



# Sketch 00, 01 und 02: Blink LED „L“

Dieser Sketch lässt die LED „L“ blinken.  
Sie befindet sich auf dem Arduino.

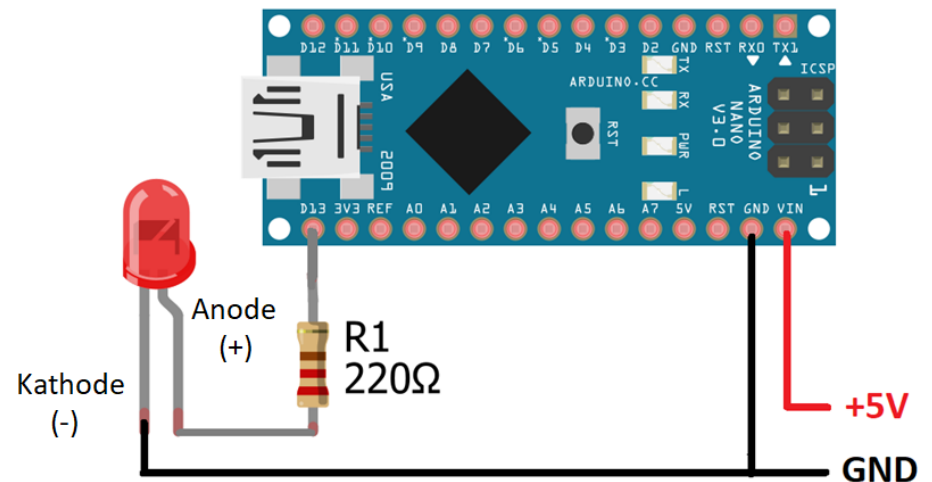


Siehe Anhang [Widerstände](#).

Das Ansteuersignal für die LED „L“ geht auch an Pin D13.

Wenn man dort eine externe LED anschließt, würde sie ebenfalls blinken.

Um aber den zulässigen Ausgangsstrom des Arduino-Pins nicht zu überschreiten, muß ein Widerstand in Reihe zur LED geschaltet werden.



# Sketch 03: Batteriespannung messen

Der Arduino kann an seinen Analog-Eingängen A0...A7 eine Spannung einlesen.  
Der Spannungswert 0...5V wird intern umgewandelt in eine Zahl 0...1023 (10 Bit).



Bei einer höheren Spannung als 5V  
kann der Arduino beschädigt werden!

Zur Messung von höheren Spannungen  
muß ein Spannungsteiler benutzt werden.

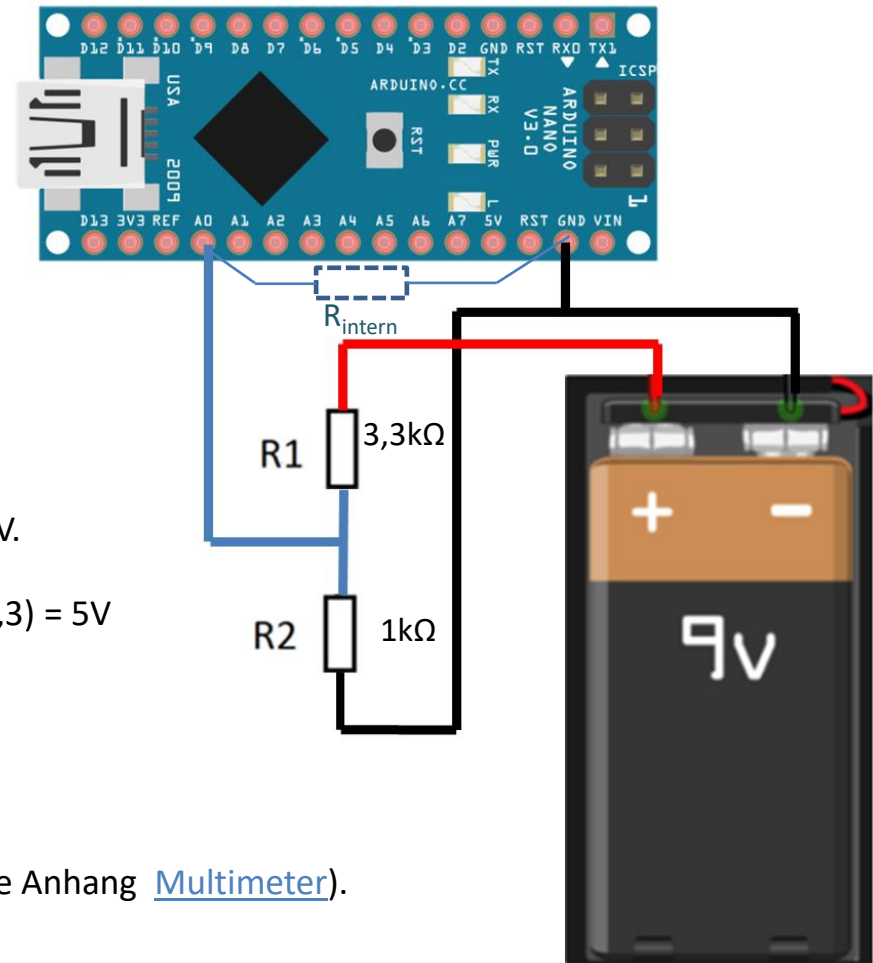
Wir benutzen dazu zwei Widerstände mit  
3,3 kOhm und 1 kOhm.

Das ermöglicht die Messung einer Spannung bis 21,5V.

Bei 21,5V liegen am Arduino-Eingang:  $21,5V * 1 / (1+3,3) = 5V$   
Welche Spannung liegt an A0 bei einer Batterie 9V ?



Vor Spannung anlegen Messung mit Multimeter (siehe Anhang [Multimeter](#)).



# Sketch 03: Batteriespannung messen

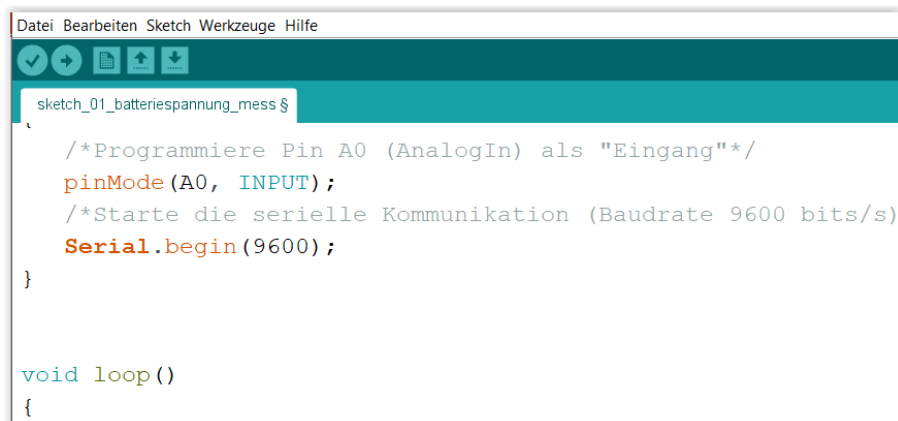
Beispiel:

Die Batteriespannung beträgt 9V.

Am Arduino Pin A0 liegt an:  $\text{Spannung} = 9V \cdot R1 / (R1+R2) = 9V \cdot 1k\Omega / (3,3k\Omega + 1k\Omega) = 2,1V$ .

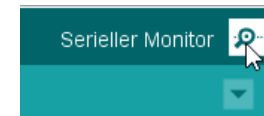
Der Arduino wandelt das um in eine Zahl  $2,1V \cdot 1023 / 5V = 430$ .

Der Sketch ermöglicht eine Anzeige im Serial Monitor:



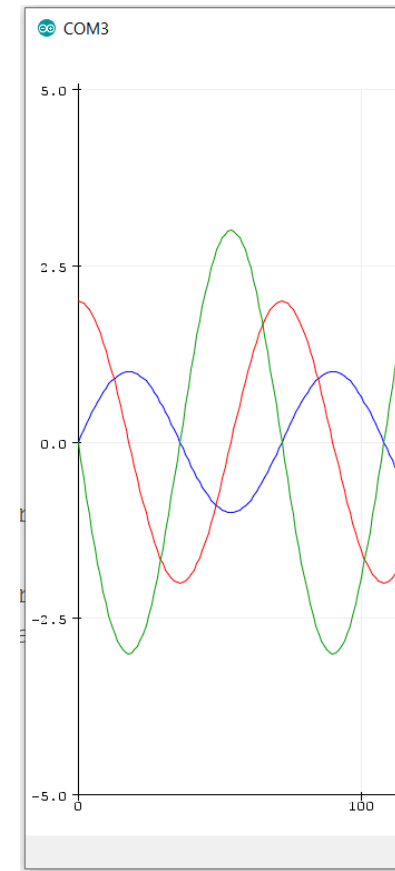
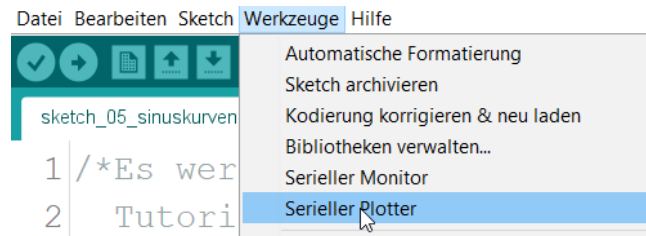
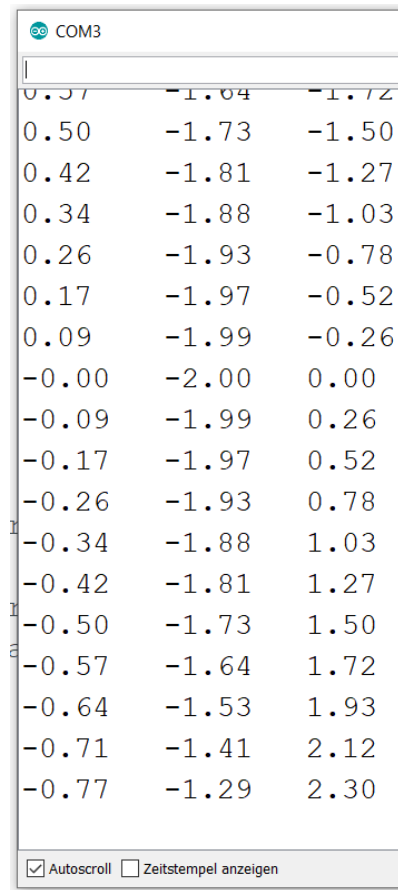
```
sketch_01_batteriespannung_mess $
/*Programmiere Pin A0 (AnalogIn) als "Eingang"*/
pinMode(A0, INPUT);
/*Starte die serielle Kommunikation (Baudrate 9600 bits/s)
Serial.begin(9600);
}

void loop()
{
```



- `analogRead()` <https://www.arduino.cc/reference/de/language/functions/analog-io/analogread/>
- `print()` <https://www.arduino.cc/reference/de/language/functions/communication/serial/print/>
- `println()` <https://www.arduino.cc/reference/de/language/functions/communication/serial/println/>

# Sketch 05: Serial Plotter - Sinuskurven

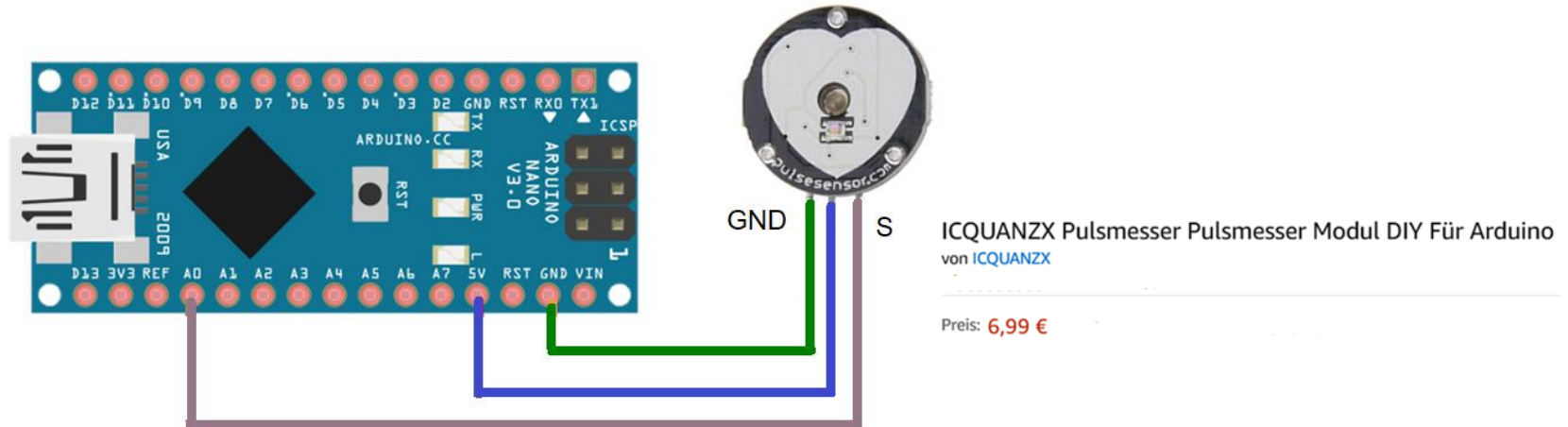


<https://arduinogetstarted.com/tutorials/arduino-serial-plotter>

<https://youtu.be/OSHMWjeFc8A>

# Sketch 06: Serial Plotter - Pulsmessung

<https://funduino.de/nr-21-herzfrequenzmessung>



Man befestigt den Sensor am Körper. Es eignen sich Stellen, an denen die super-helle LED bis zu einer pulsierenden Ader „durchleuchten“ kann (z.B. die Fingerkuppe des Ringfingers).

Der serielle Plotter nimmt zunächst ein paar Messwerte auf und stellt sich dann automatisch so ein, dass die eingelesenen Messwerte ideal dargestellt werden, es dauert einige Sekunden.

Man sieht an den Ausschlägen, wie der Grenzwert für die Einschaltung der LED 13 eingestellt werden muß (hier ca. 535).

Jedesmal, wenn der Befehl `Serial.println ()` aufgerufen wird, erfolgt eine neue Ausgabe (genau wie beim Serial Monitor).

# Sketch 06: Serial Plotter - Pulsmessung

Die Geschwindigkeit der Ausgabe kann verändert werden durch Einstellung am Serial Monitor.

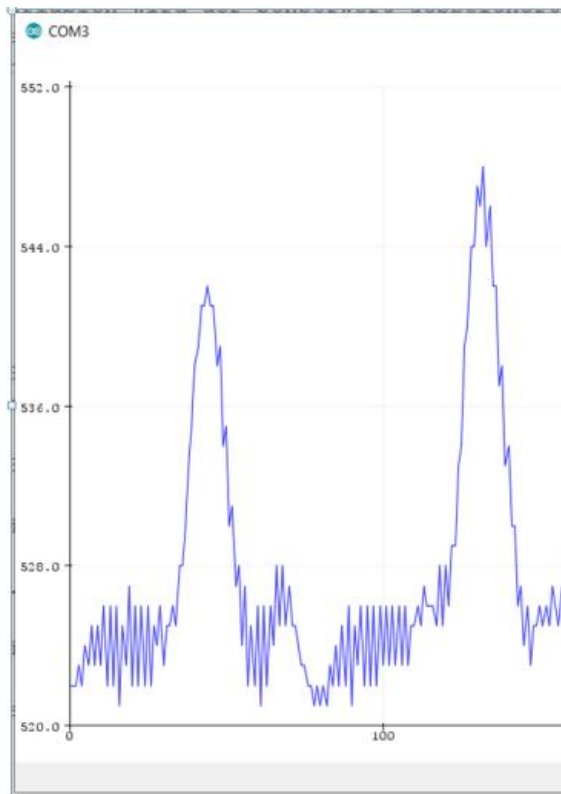


Achtung: Muß übereinstimmen mit der Einstellung im sketch, z.B. `Serial.begin(9600)`

Jedesmal, wenn der Befehl `Serial.println()` aufgerufen wird, erfolgt eine neue Ausgabe (genau wie beim Serial Monitor).

Die Zeit zwischen zwei Ausgaben entspricht also der Zeit zwischen zwei Print-Befehlen.

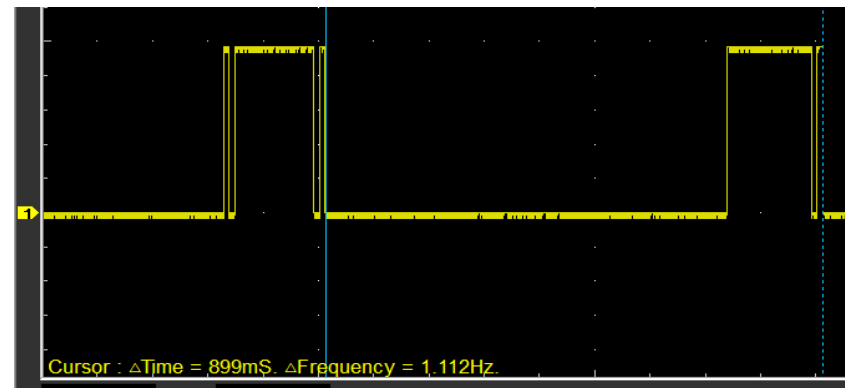
Sie kann am Serial Plotter nicht direkt ermittelt werden.



Eine Messung der Zeit kann mit einem Oszilloskop erfolgen, am Ausgang D13.

Die Pulsfrequenz beträgt 66 pro Minute.

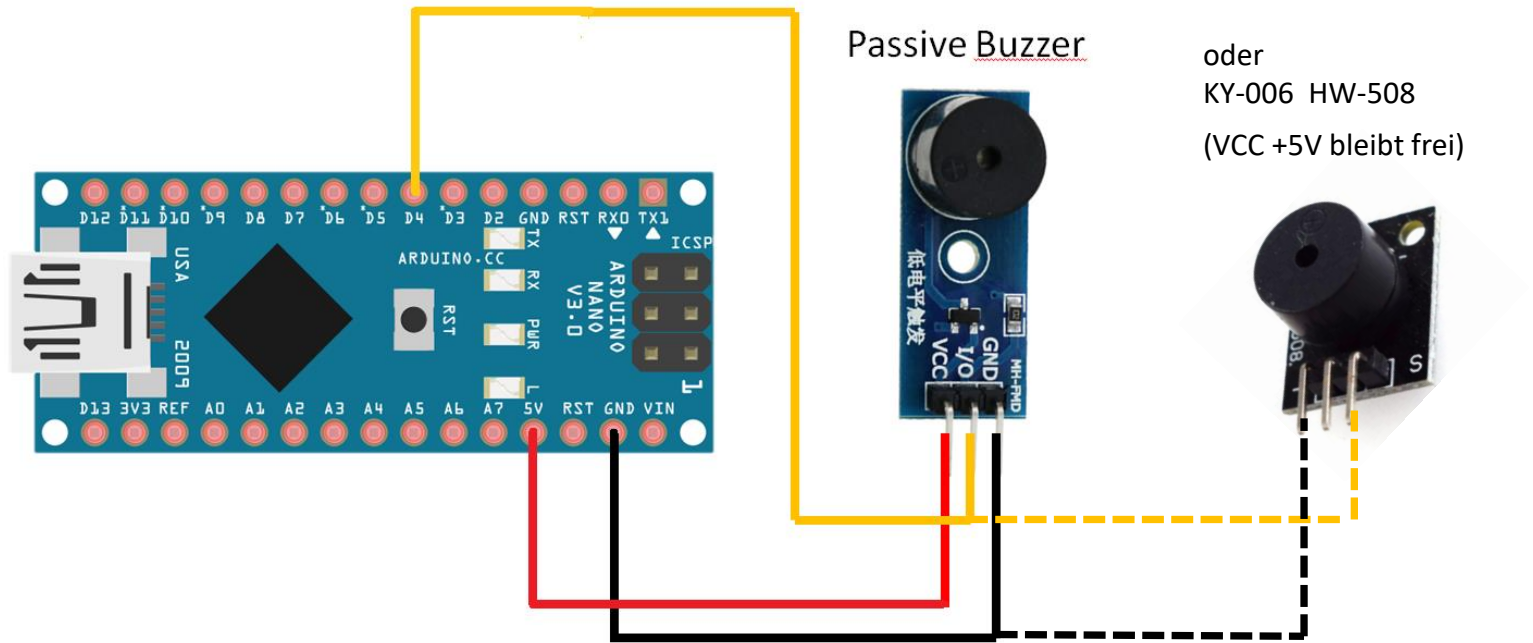
(siehe Anhang, [Oszilloskop Hantek](#) )







# Sketch 20 Buzzer Tone



```
tone() https://www.arduino.cc/reference/en/language/functions/advanced-io/tone/
```

# Tonleiter und Noten

b: des es ges as b des es ges as b des es ges as b des es ges as b des es ges as b  
#: cis dis fis gis ais cis dis fis gis ais cis dis fis gis ais cis dis fis gis ais cis dis fis gis ais

C D E F G A H c d e f g a h c' d' e' f' g' a' h' c'' d'' e'' f'' g'' a'' h'' c''

Kammerton a`  
440 Hz

Grosse Oktave kleine Oktave eingestrichene Oktave zweigestrichene Oktave

Noten  
Ganze Halbe Viertel Achtel 1/16 1/32

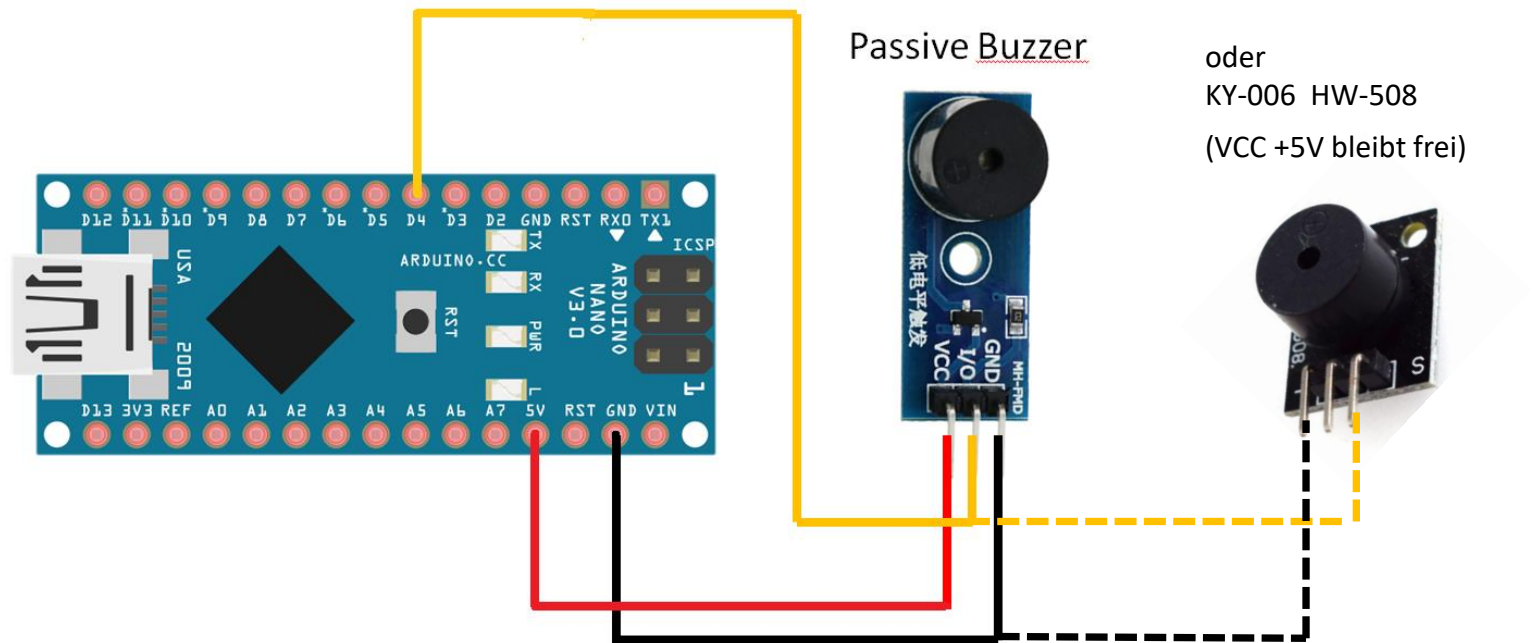
Pausen

<https://stimmungen.de/privates/musiklehre/>

<http://www.animations.physics.unsw.edu.au/jw/frequency-pitch-sound.htm>

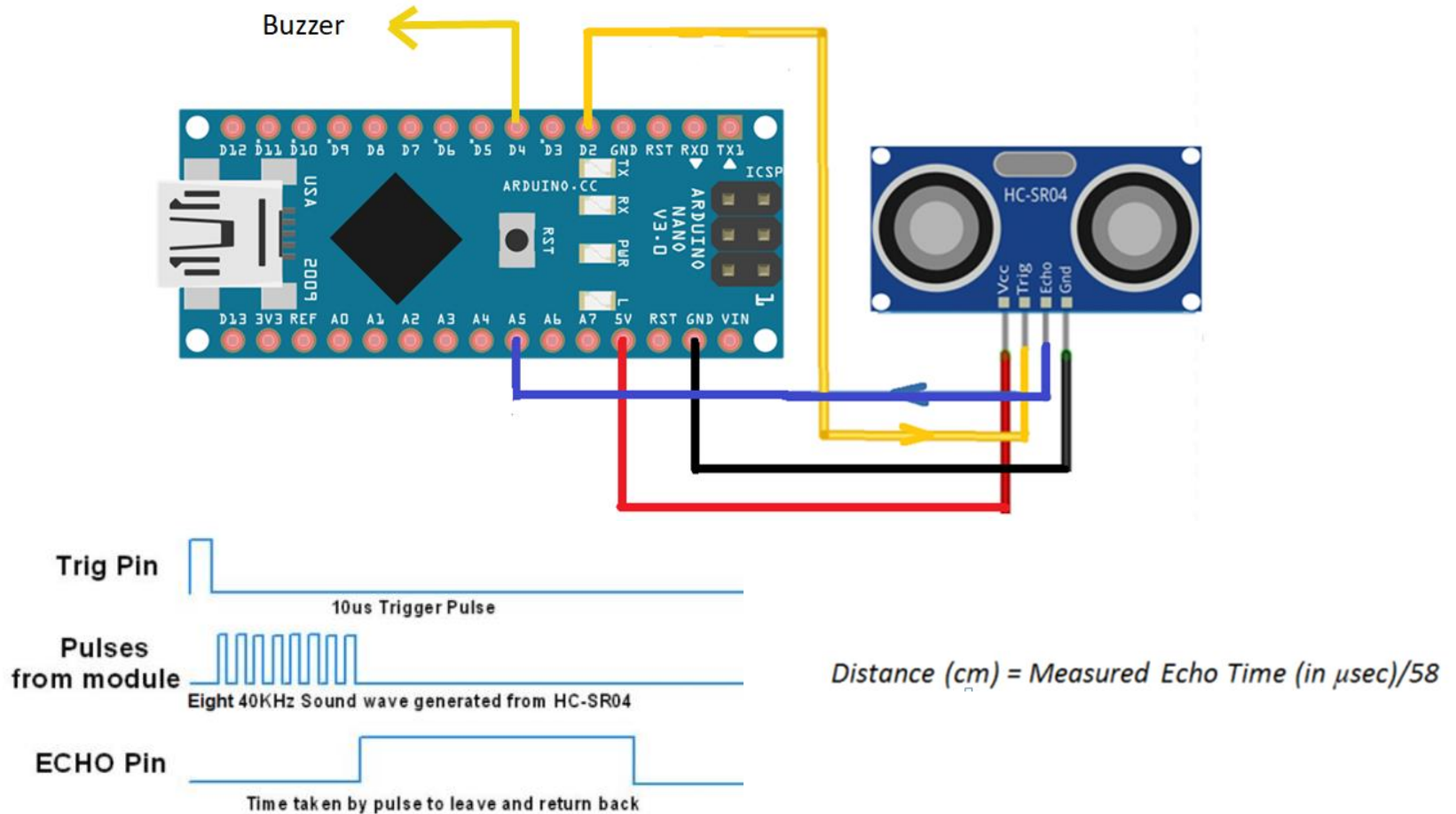


# Sketch 21 Buzzer Melody



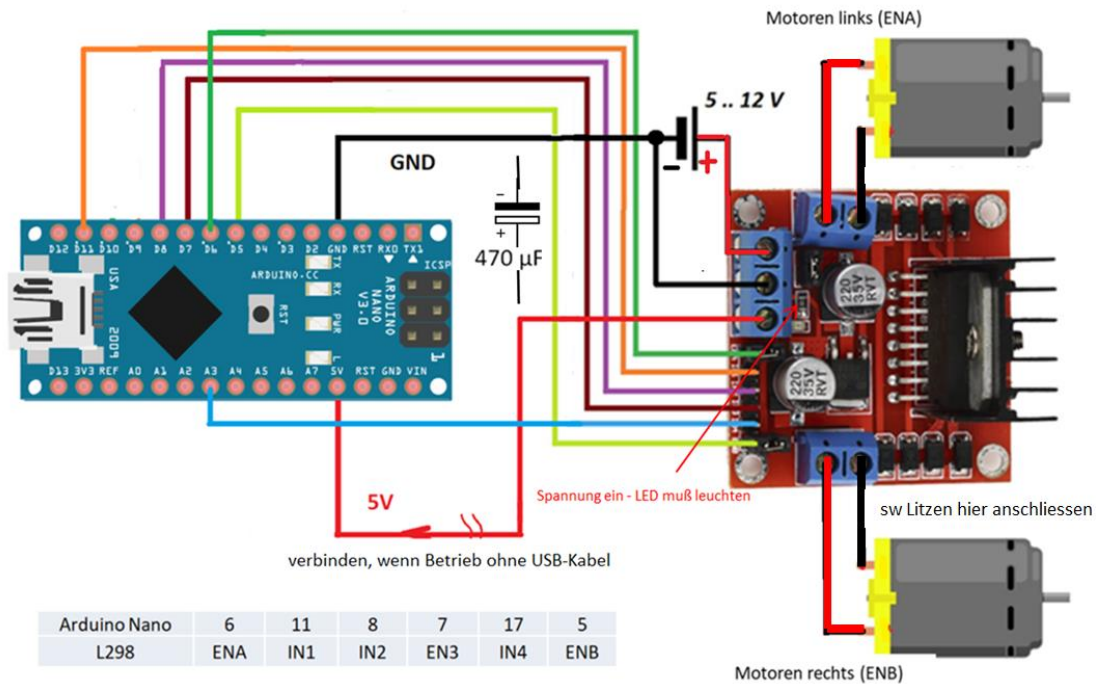
```
array https://www.arduino.cc/reference/de/language/variables/data-types/array/  
https://www.arduino.cc/de/Tutorial/ToneMelody?from=Tutorial.Tone  
#define https://www.arduino.cc/reference/de/language/structure/further-syntax/define/
```

# Sketch 22 Ultraschall Abstandsmessung

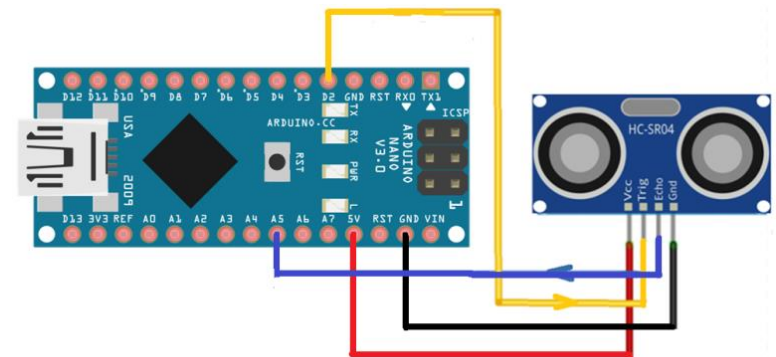


`pulseIn()` <https://www.arduino.cc/reference/de/language/functions/advanced-io/pulsein/>

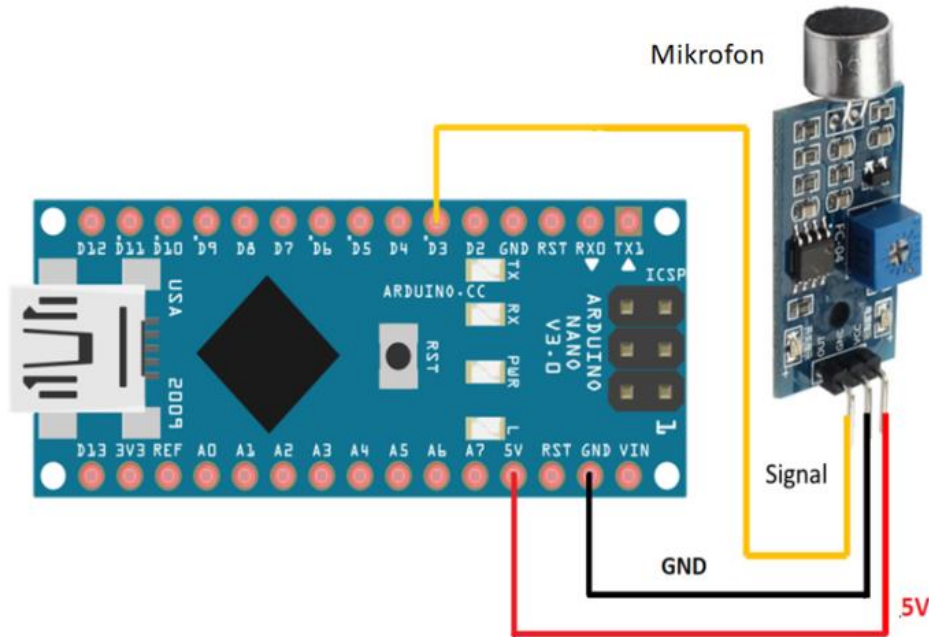
# Sketch 23 Auto programmiert fahren mit US-Hinderniserkennung



Kondensator 470 µF einfügen zum Abpuffern der Einschalt-Stromspitze

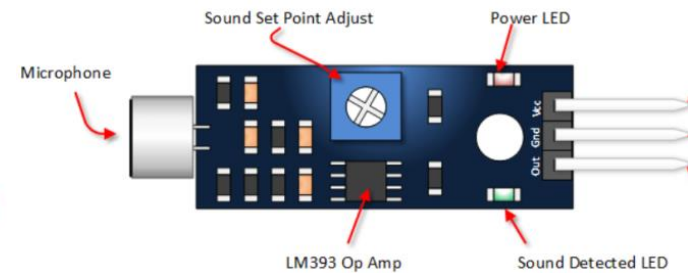


# Sketch 24 Geräuscherkennung mit FC04



FC04

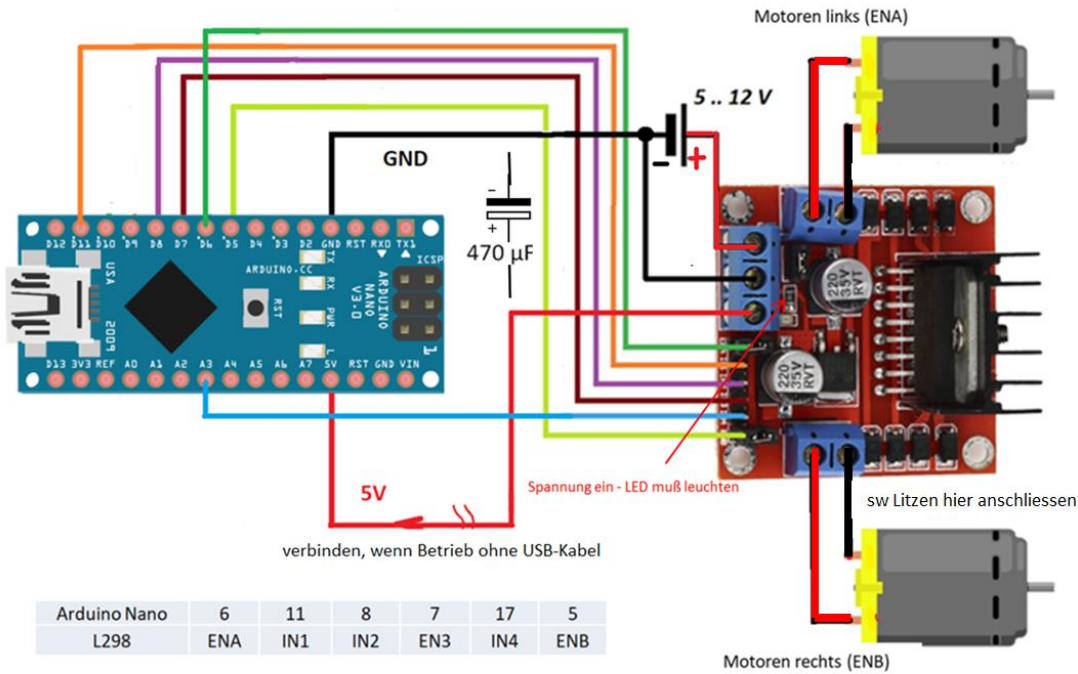
Das Poti so justieren, dass die LED bei ruhiger Umgebung gerade noch nicht leuchtet.



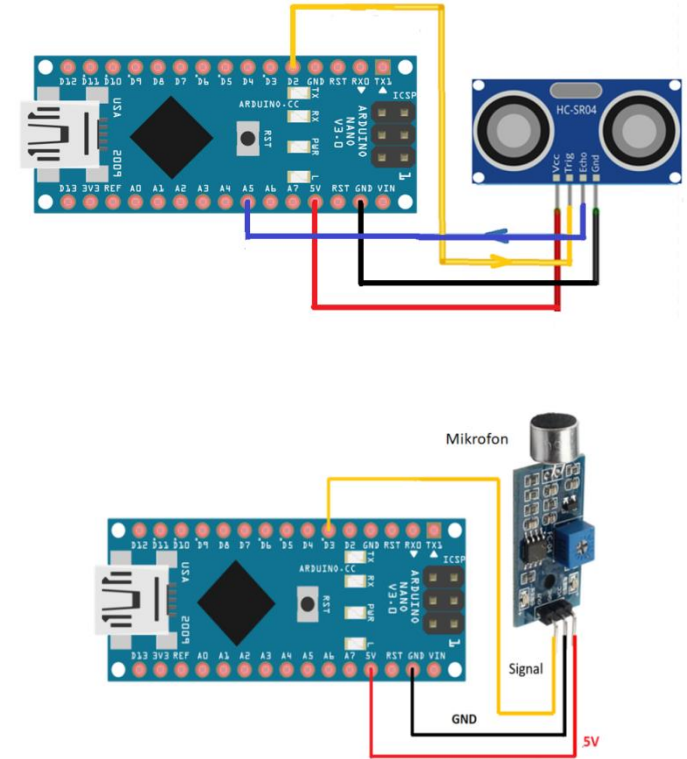
`attachInterrupt()` <https://www.arduino.cc/reference/de/language/functions/external-interrupts/attachinterrupt>

Durch einen Interrupt wird eine Interrupt Service Routine (ISR) aufgerufen, ein separates Programm. Solange wir uns in einer ISR befinden, wird kein weiterer Interrupt erkannt und es funktioniert kein *delay* und auch der Wert der *millis* wird nicht aktualisiert. Daher sollten die ISR so kurz wie möglich sein.

# Sketch 25 Auto vorwärts fahren mit Geräuscherkennung



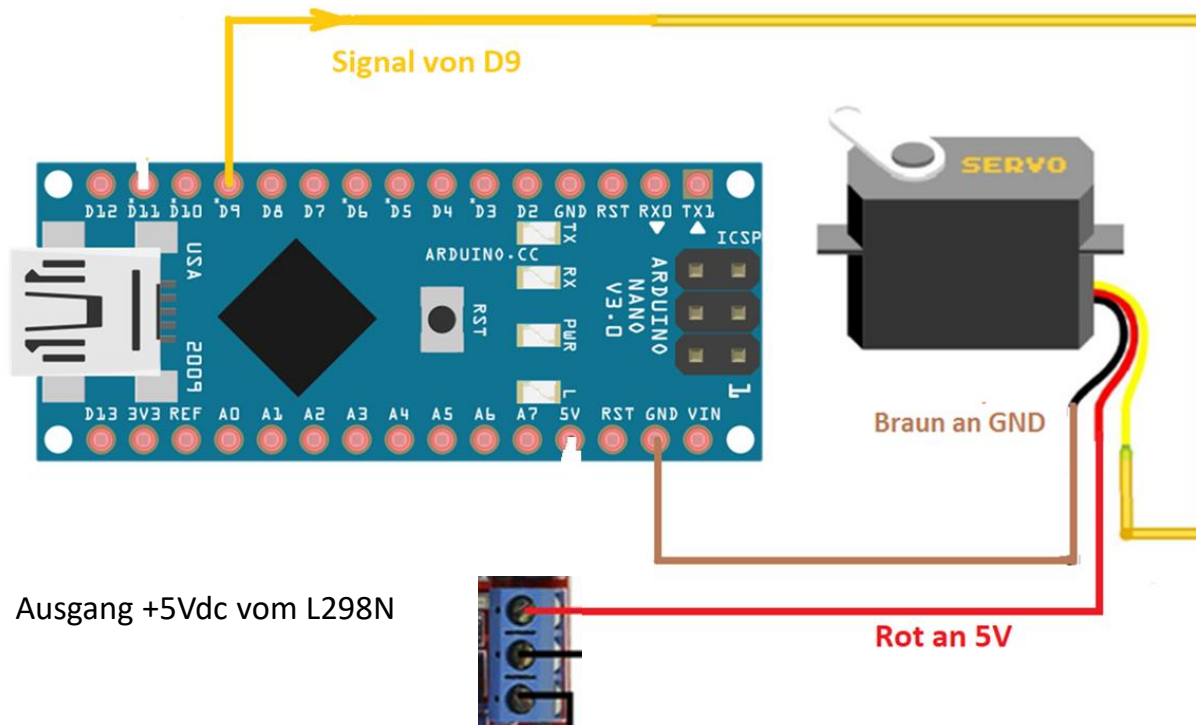
Kondensator 470 µF einfügen zum Abpuffern der Einschalt-Stromspitze



Das Poti so justieren, dass die LED bei ruhiger Umgebung noch nicht leuchtet



# Sketch 26 +27 Servo ansteuern



Zuerst die Mittelstellung durch (vorsichtiges) Verdehen per Hand austesten.

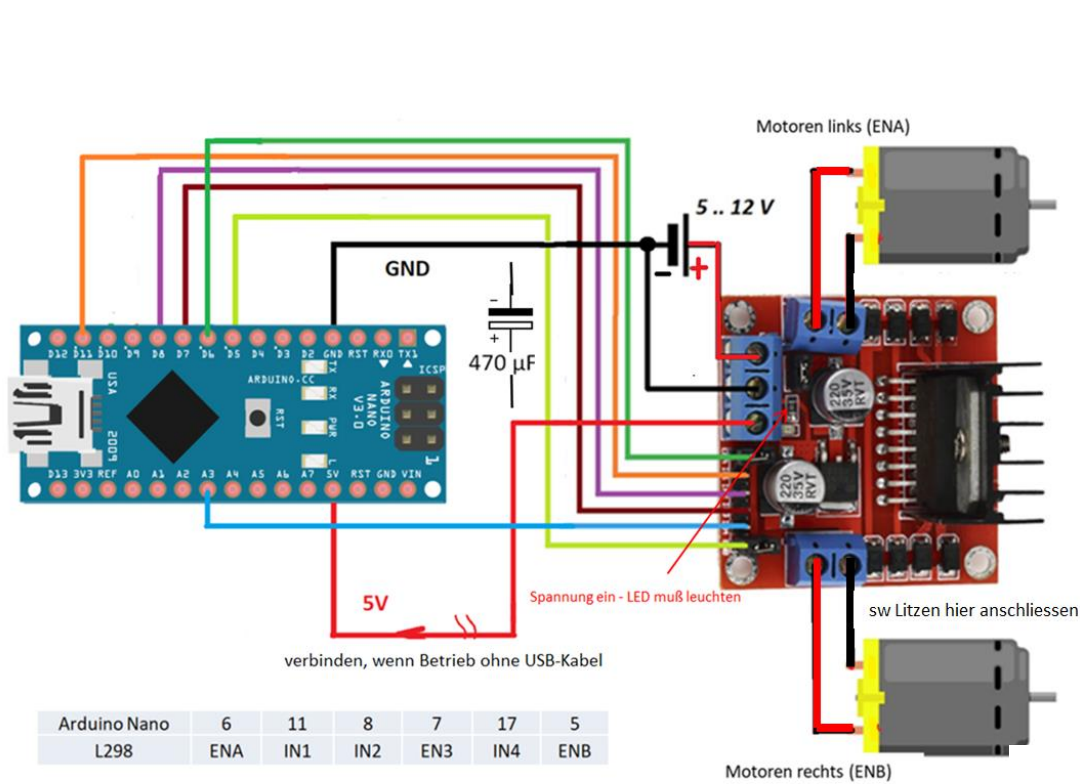
Dann austesten, ob die Mittelstellung tatsächlich bei 1,5ms erreicht wird – evtl. die Zeit etwas korrigieren.

Achtung: `delayMicroseconds()` funktioniert nur bis 16383 („integer“) !

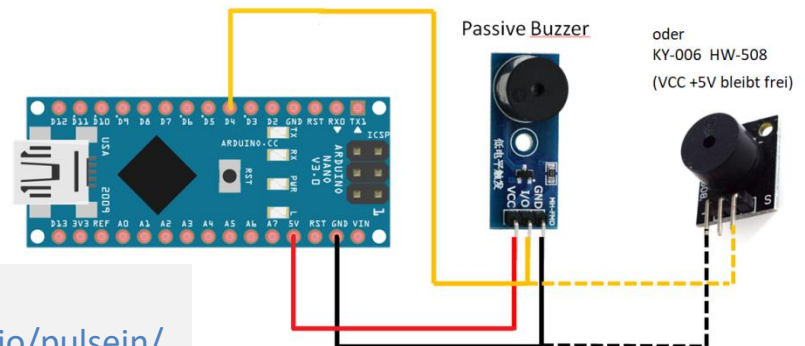
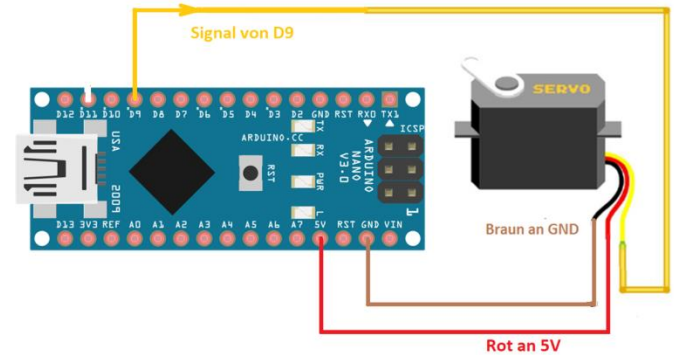
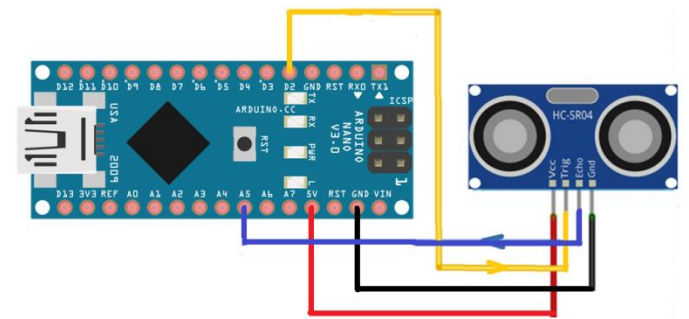
Zum Auffüllen auf 20ms: `delay (20)` verwenden.

Achtung: Achtung: +5Vdc vom L298N muß an den Servo gelegt werden (Stromaufnahme ist zu hoch, Spannungsversorgung über USB - Arduino schafft es nicht).

# Sketch 28 Auto fahren, Servo und US-Hinderniserkennung



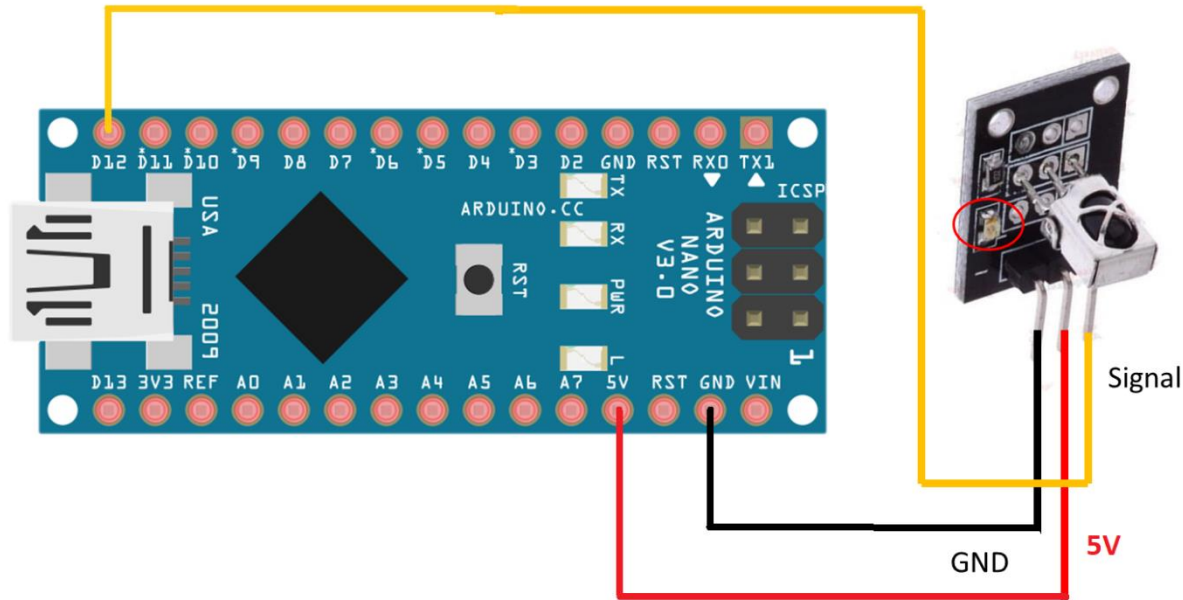
Kondensator 470 µF einfügen zum Abpuffern der Einschalt-Stromspitze



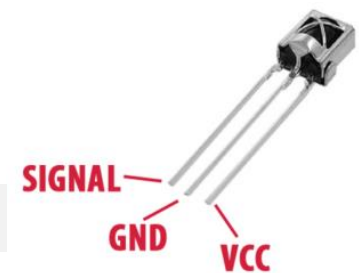
`pulseIn()`

[www.arduino.cc/reference/de/language/functions/advanced-io/pulsein/](http://www.arduino.cc/reference/de/language/functions/advanced-io/pulsein/)

# Sketch 30 IR Fernsteuerung Ermittlung Code



LED flackert wenn eine Taste der Fernbedienung gedrückt wird



VS1838B

resume() [www.arduino.cc/en/Reference/CurieTimerOneResume](http://www.arduino.cc/en/Reference/CurieTimerOneResume)

Siehe auch: <https://www.youtube.com/watch?v=kSPDHAQm9MA>

# Sketch 30 IR Fernsteuerung Ermittlung Code

Bibliotheksprogramm IRremote muß installiert und eingebunden werden.

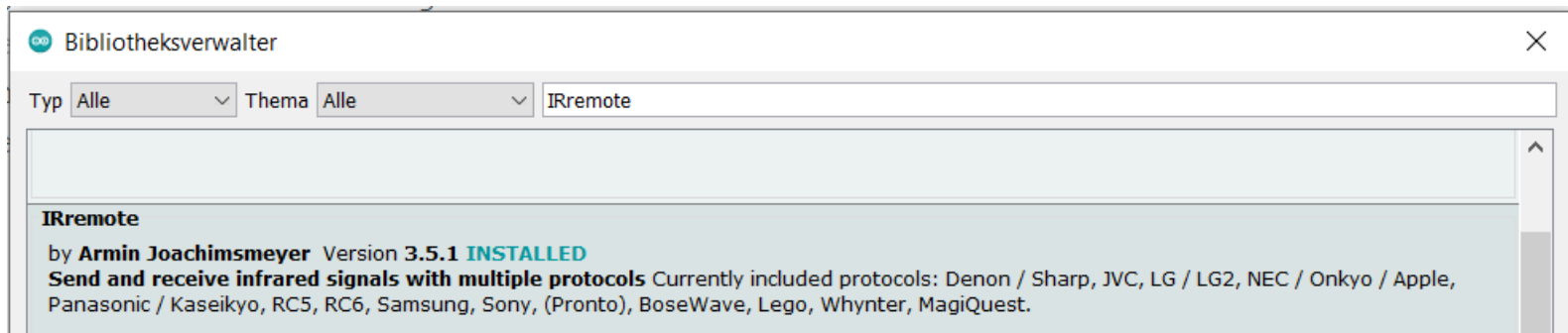
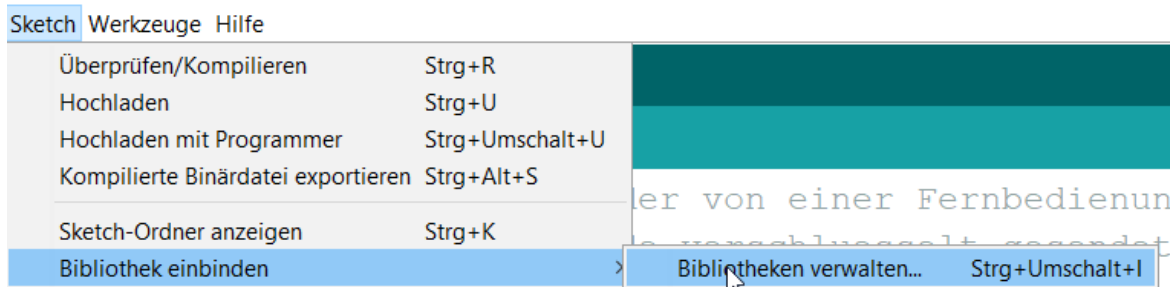
[IRremote - Arduino Reference](#)

[IRremote - Arduino Libraries](#)

## IRremote

Send and receive infrared signals with multiple protocols

**Author** [shirriff, z3t0, ArminJo](#)  
**Maintainer** [Armin Joachimsmeier](#)  
**Website** <https://github.com/Arduino-IRremote/Arduino-IRremote>  
**Github** <https://github.com/z3t0/Arduino-IRremote>



# Sketch 30 IR Fernsteuerung Ermittlung Code

[https://www.amazon.de/TOOGOO-Infrarot-Wireless-Fernbedienung-Knopfzelle/dp/B075XC71W4/ref=sr\\_1\\_63?ie=UTF8&qid=1521391130&sr=8-63&keywords=arduino+ir+fernbedienung](https://www.amazon.de/TOOGOO-Infrarot-Wireless-Fernbedienung-Knopfzelle/dp/B075XC71W4/ref=sr_1_63?ie=UTF8&qid=1521391130&sr=8-63&keywords=arduino+ir+fernbedienung)

Prüfung, ob die Fernbedienung Infrarot-Signale sendet:

Starte Kamerafunktion auf dem Smartphone und drücke einen Knopf auf der Fernbedienung. Jetzt solltest du die LED der Fernbedienung blinken sehen.

Funktioniert die Fernbedienung nicht, kann es sein das z.B. die Batterien leer sind oder noch eine Schutzfolie zwischen den Batteriekontakten klemmt.

Achtung: Einige Kameras haben einen IR Blocker vor der Linse und das IR Licht dringt nicht bis zum Objektiv vor.



- ▲ Vorwärts forward
- ▼ Rückwärts back
- ◀ Links left
- ▶ Rechts right
- OK Stopp stopp
- \* Greifer auf
- # Greifer zu
- 0 Hupen tone

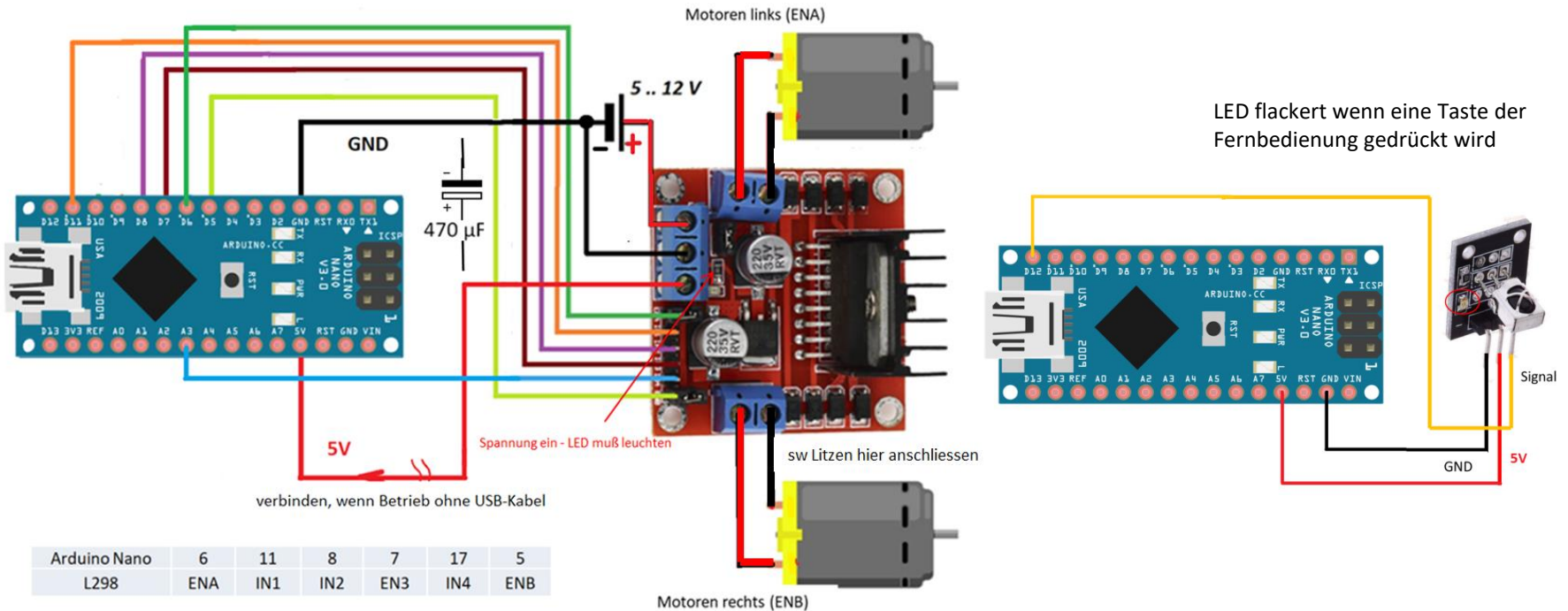
Beispiel:

Wenn Taste ▲ gedrückt wird, erscheint im Serial Monitor:

```
COM3
16718055
4294967295
```

16718055 muß dann im Sketch 31 bei codes.h eingetragen werden.

# Sketch 31 Auto fahren mit IR Fernsteuerung



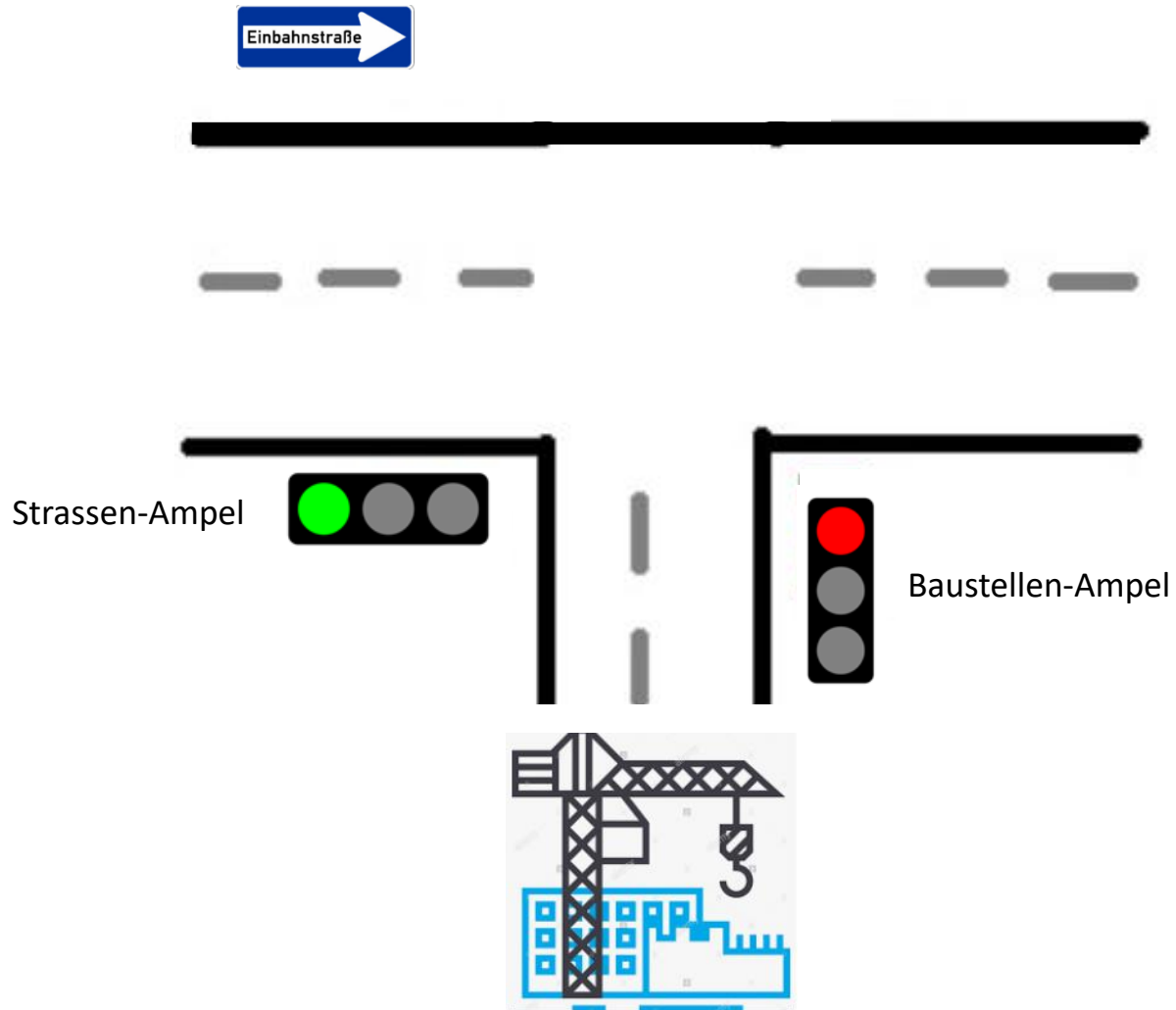
Kondensator 470 µF einfügen zum Abpuffern der Einschalt-Stromspitze

break [www.arduino.cc/reference/de/language/structure/control-structure/break/](http://www.arduino.cc/reference/de/language/structure/control-structure/break/)

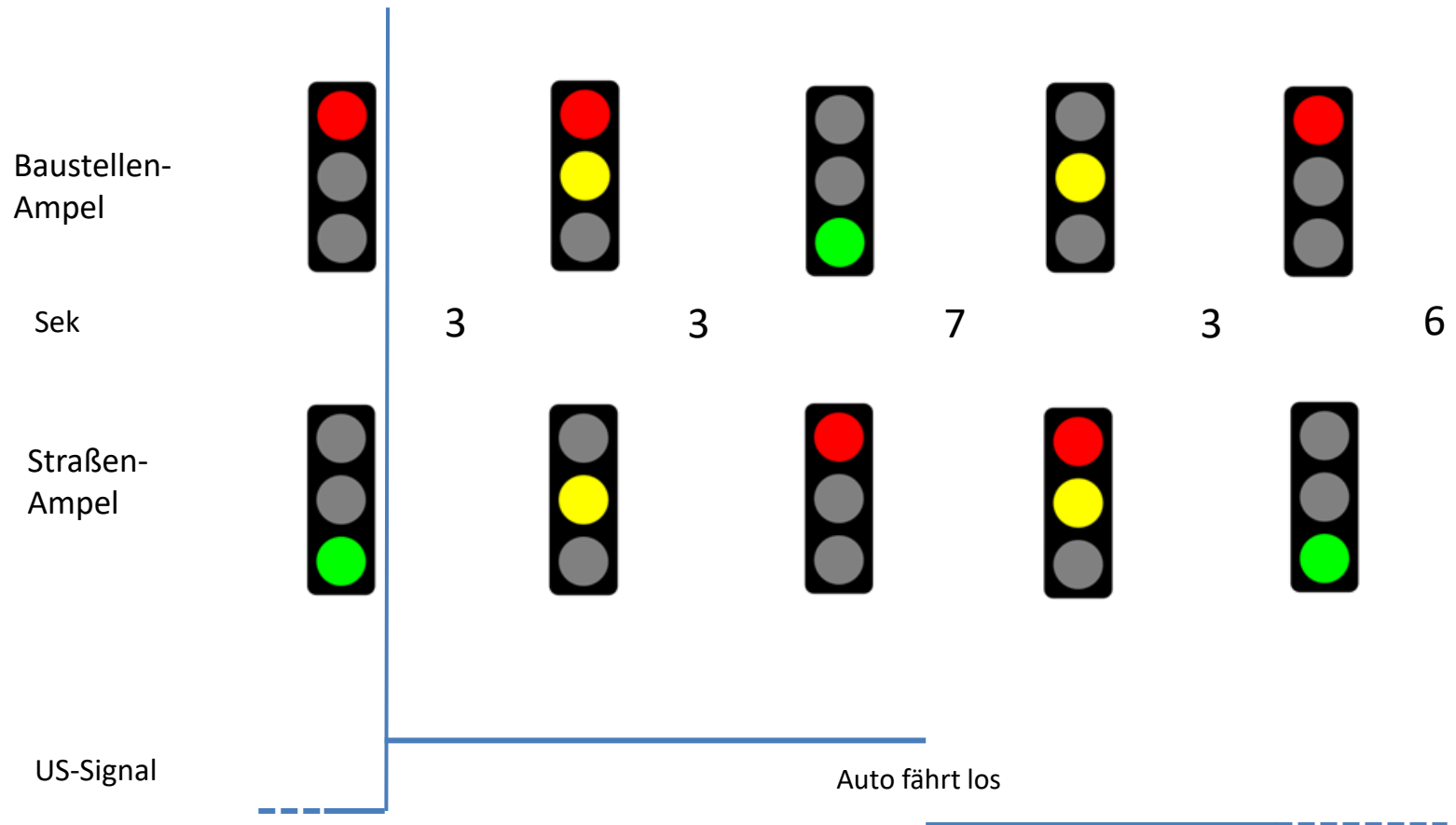
Switch...case [www.arduino.cc/reference/de/language/structure/control-structure/switchcase/](http://www.arduino.cc/reference/de/language/structure/control-structure/switchcase/)

# Sketch 35 Ampel

---

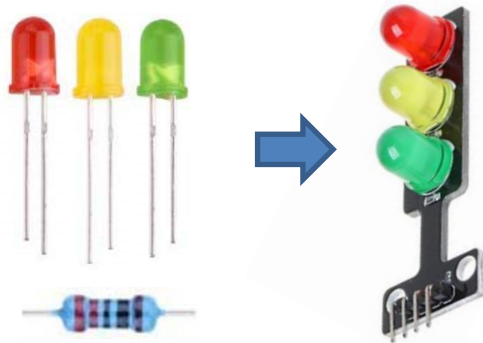


# Sketch 35 Ampel

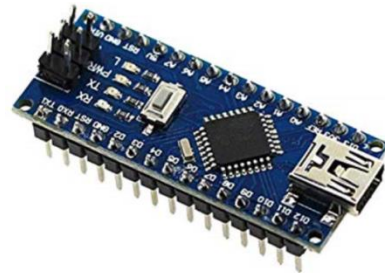




# Sketch 35 Ampel



HC-SR04



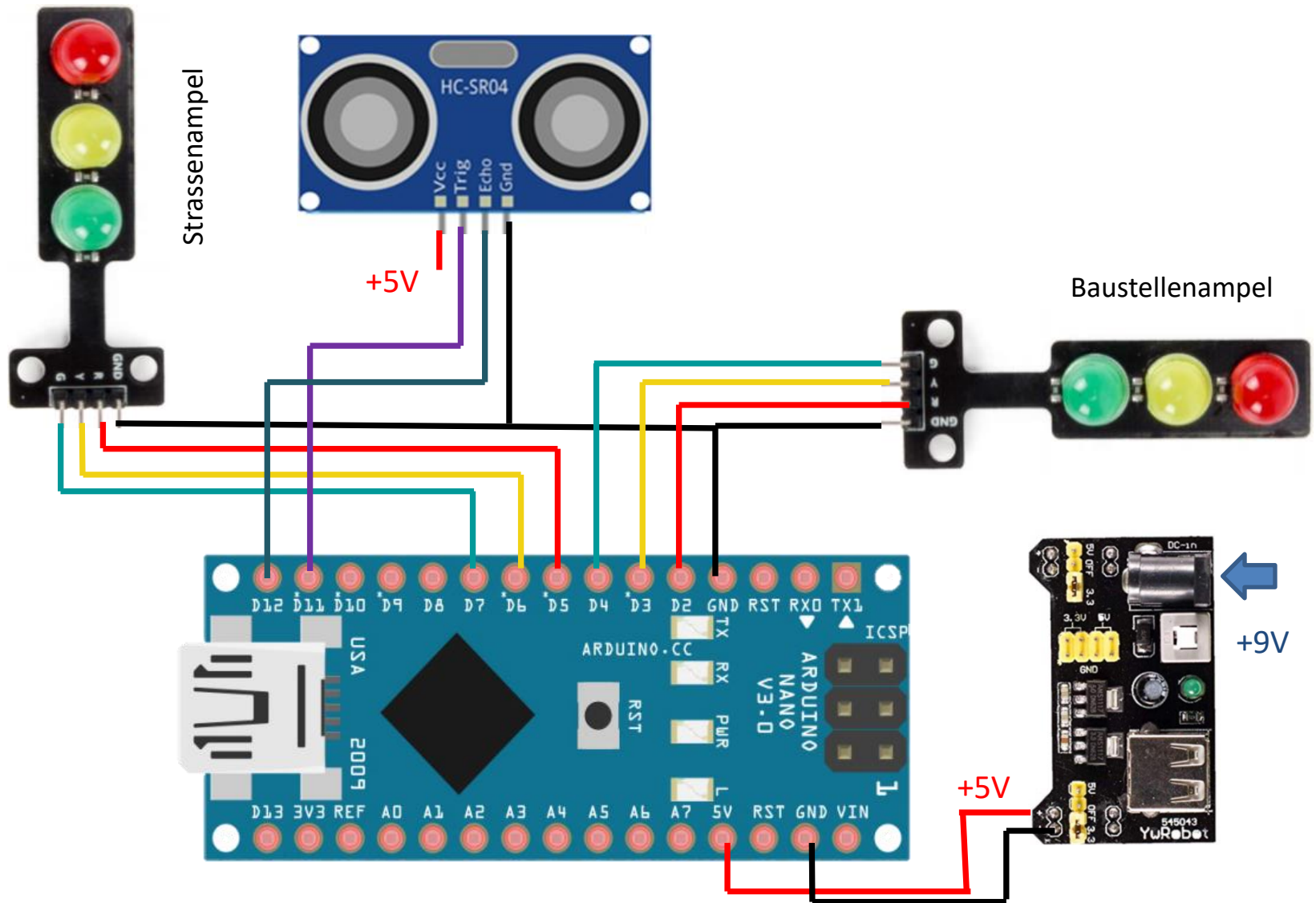
Siehe Anhang:  
[Spannungsversorgung MB102](#)

# Sketch 35 Ampel - Zuordnung Arduino Nano Pins

---

Nano	Funktion	Eing/Ausg		
D2	Rot der Baustellen-Ampel	Ausgang	LED Rot_Bst	
D3	Gelb der Baustellen-Ampel	Ausgang	LED Gelb_Bst	
D4	Grün der Baustellen-Ampel	Ausgang	LED Gruen_Bst	
D5	Rot der Straßen-Ampel	Ausgang	LED Rot_Str	
D6	Gelb der Straßen-Ampel	Ausgang	LED Gelb_Str	
D7	Grün der Straßen-Ampel	Ausgang	LED Gruen_Str	
D11	Ultraschall-Sensor Senden	Ausgang	TRIG	
D12	Ultraschall-Sensor Empfangen	Eingang	ECHO	

# Sketch 35 Ampel



# Anhang

[Inhaltsverzeichnis](#)

# Abkürzungen

---

ComPort	Communication Port (Kommunikations-Anschluß)
DC Motor	Direct Current Motor (Gleichstrom-Motor)
IDE	Integrated Development Environment (Integrierte Entwicklungsumgebung)
IR	Infrarot
LED	Light Emitting Diode
NiMH	Nickel Metallhydrid (Akku)
pdf	portable document format (übertragbares Dokument-Format)
PWM	PulsweitenModulation
USB	Universal Serial Bus



# Weblinks zum Thema Arduino

---

- Arduino homepage (en):
  - <https://www.arduino.cc/>
- Arduino Sprachreferenz (de):
  - <https://www.arduino.cc/reference/de/>
- Arduino Programmierung (de):
  - <http://www.netzmafia.de/skripten/hardware/Arduino/Programmierung/index.html>
- Arduino allgemein (de):
  - <http://www.netzmafia.de/skripten/hardware/Arduino/Intro/index.html>
- Arduino allgemein (en):
  - <https://www.tutorialspoint.com/arduino/index.htm>

- <https://about.udemy.com/de/>
- <https://www.udemy.com/course/arduino-lehrgang-vom-anfaenger-zum-profi/>



**12,99 €** ~~149,99 €~~

91 % Rabatt

🕒 Noch **1 Tag** zu diesem Preis!

In den Einkaufswagen

Jetzt kaufen

30-Tage-Geld-zurück-Garantie

Dieser Kurs bietet

📺 11 Stunden On-Demand-Video

📄 13 Artikel

📁 4 zum Download verfügbare Materialien

🌐 Uneingeschränkter lebenslanger Zugriff

📱 Zugriff auf Handy/Tablet und TV

⚡ Aufgaben

📜 Abschlussbescheinigung

# Variablen

---

- Eine globale Variable ist eine Variable, die von jeder Funktion in dem Programm (Sketch) "gesehen" werden kann.
- Sie muß außerhalb einer Funktion (z.B. setup() , loop(), etc) definiert werden. Das wird im allgemeinen zu Beginn des Sketches gemacht.

```
1 int var; //die globale Variable var wird definiert
2
3 void setup(){
4 var = 1;
5 }
6
7 void loop(){
8 var = var +1;
9 }
```

- Lokale Variablen sind nur gültig in der Funktion, in der sie deklariert wurden und können von außerhalb nicht erkannt oder verändert werden.
- Wenn Programme größer und komplexer werden, sind lokale Variablen eine Möglichkeit, sicherzustellen, dass nur die Funktion Zugriff auf die Variable hat, die dies wirklich benötigt.
- Das verhindert u.a. dass Funktionen unabsichtlich die Variablen anderer Funktionen ändern.
- Eine Variable kann auch nur innerhalb einer for-Schleife definiert werden

```
1 void setup(){
2 }
3
4 void loop(){
5 int var = var +1; //var ist nur innerhalb loop()gültig
6 }
```

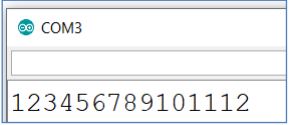


# Variablen – Übertragung zwischen Unterprogrammen

- Manchmal besteht der Wunsch, Variablen zwischen Unterprogrammen zu übergeben. Wie geht das ?
- Im Hauptprogramm wird die Variable als Globale Variable definiert („definiert“ heißt, es wird dafür auch Speicherplatz reserviert).
- Im Unterprogramm wird diese Variable nochmal deklariert. Dazu notwendig ist der Zusatz `extern` .
- Dem Compiler wird dadurch mitgeteilt, dass es sich um denselben Speicherplatz handelt.

```
sketch_hochzaehlen  unterprogramm.h
1 int var; //Globale Variable wird definiert
2 #include "unterprogramm.h";
3
4 void setup() {
5   var = 0;
6   Serial.begin (9600);
7 }
8
9 void loop() {
10  hochzaehlen();
11  Serial.print (var);
12  delay (1000);
13 }
```

```
sketch_hochzaehlen  unterprogramm.h
1 extern int var; //var wird hier nochmal deklariert
2
3 void hochzaehlen() {
4   var = var +1;
5 }
```



```
COM3
123456789101112
```

# Bibliotheken (Libraries)

---

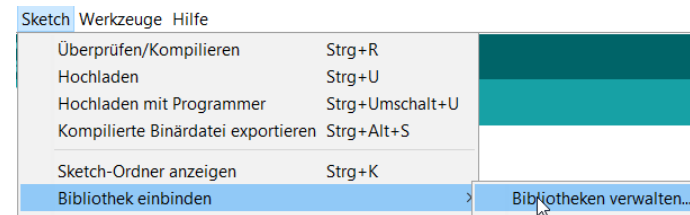
- <https://www.arduino.cc/en/Guide/Libraries>
- <https://www.heise.de/make/artikel/Arduino-Bibliotheken-einbinden-und-verwalten-3643786.html>
- Die Libraries die bei der Installation der IDE bereits mit installiert werden, sind auf verschiedene Ordner verteilt: Arduino\hardware\arduino\avr\libraries oder Arduino\libraries.
- Im Quellcode kann man erkennen, wo welche Libraries zu finden sind.

```
#include <EEPROM.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include "gfx.h"
```

- Die vorinstallierten "Standard"-Bibliotheken müssen in spitzen Klammern angegeben werden und werden automatisch orange formatiert.
- Zusätzlich über den Manager installierte Dateien (*recommended*) werden ebenfalls in spitzen Klammern notiert aber erscheinen in schwarzer Schrift.
- Selbsterstellte Libraries (*contributed*), die im gleichen sketch-Ordner liegen, stehen in Anführungszeichen (blau-grün).
- Liste aller Arduino Libraries:
- <https://www.arduinolibraries.info/>
- <https://www.arduinolibraries.info/libraries>

# Bibliothek einbinden – mögliche Fehlermeldung

Beim Versuch, ein Bibliotheksprogramm einzubinden,



Fehler beim Herunterladen von [https://downloads.arduino.cc/libraries/library\\_index.json](https://downloads.arduino.cc/libraries/library_index.json)

Das passiert, wenn die IDE abgelegt ist auf **Programme (x86)** und der Ordner **portable** angelegt worden ist.

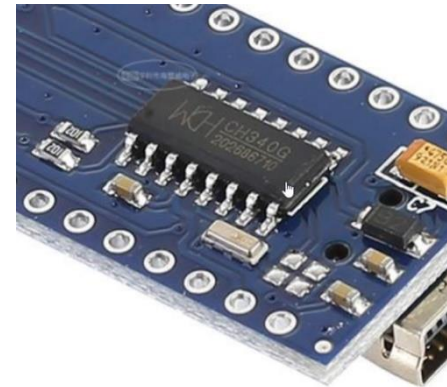
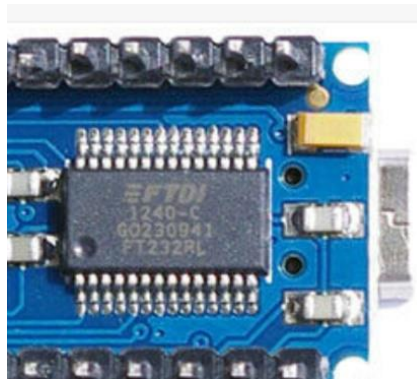
Kein Problem mehr, wenn die IDE auf einem anderen Ordner abgelegt wird oder der Ordner **portable** nicht angelegt wurde.

<https://iotboys.com/how-to-fix-arduino-ide-additional-library-installation-error/>

# USB to UART Converter mit CH340G

- Neben dem eigentlichen Mikrocontroller-IC (Atmega328P) befindet sich an der Unterseite des Nano noch ein weiterer IC.
- Er dient zur Kommunikation mit dem Computer und wandelt die Signale der USB-Schnittstelle in UART-Signale um, die vom Atmega328P verarbeitet werden können.
- *UART: Universal Asynchronous Receiver Transceiver (asynchrone serielle Schnittstelle)*

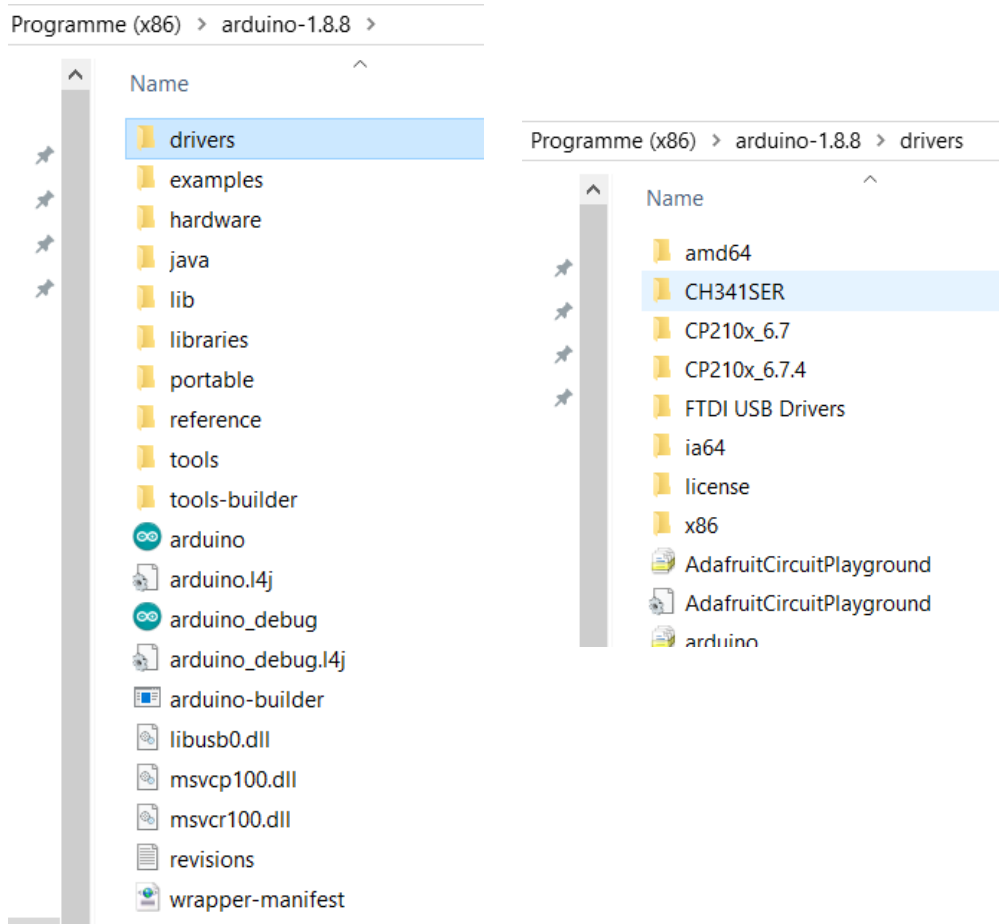
- Es gibt zwei Varianten des Nano:
- Mit FTDI
- Mit CH340G



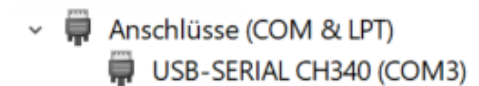
- Die Variante mit CH340 ist preiswerter (für FTDI müssen die Hersteller eine Lizenzgebühr bezahlen).
- Allerdings muß noch ein Treiber dafür installiert werden.
- Siehe dazu:
- <https://www.makershop.de/ch340-341-usb-installieren/>
- Download-Seite auf englisch: [http://www.wch-ic.com/downloads/CH341SER\\_ZIP.html](http://www.wch-ic.com/downloads/CH341SER_ZIP.html)

# Treiber (Driver) für CH340

Nach Download des Treibers von der Webseite – Installieren in den Ordner „drivers“



Nach Anschluß des Arduino Nano muß im Geräte-Manager erscheinen:

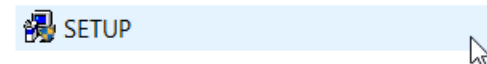


Wenn das nicht der Fall ist:

Rechter Mausklick und „Treiber aktualisieren“ („An diesem Ort nach Treibern suchen“: Ordner CH341SER und anklicken „Unterordner einbeziehen“)

Achtung:  
Das ist nur möglich als Administrator !

Oder im Ordner CH341SER aufrufen:



# Bootloader für Arduino Nano – zu beachten !

<https://www.heise.de/make/artikel/Arduino-Nano-mit-neuem-Bootloader-4011641.html>  
<https://www.arduino.cc/en/Guide/ArduinoNano#toc4>

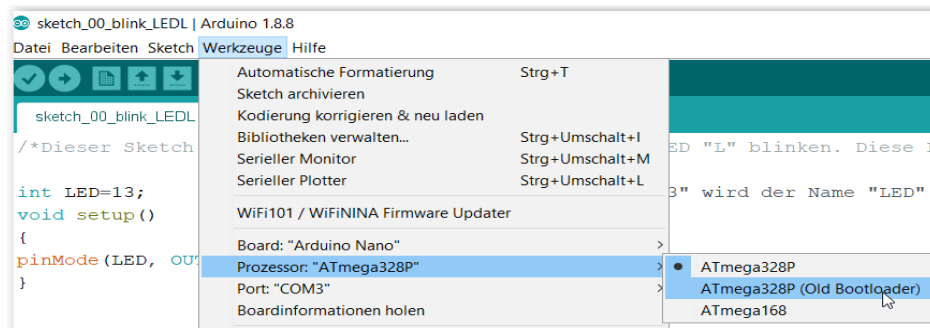
Nano Boards mit FTDI, produziert seit Jan 2018, benutzen den neuen Bootloader.

Nano Boards davor bzw. die China-Boards mit CH340G wahrsch noch für längere Zeit, benutzen noch den alten Bootloader.

Wenn die Arduino IDE neu runtergeladen und installiert wird, wird automatisch der neue Bootloader installiert.

Der Boardverwalter benötigt ein Update auf mindestens 1.6.21.

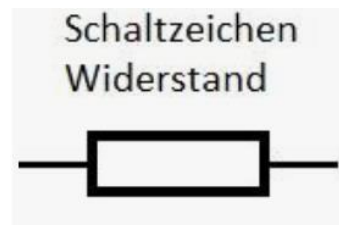
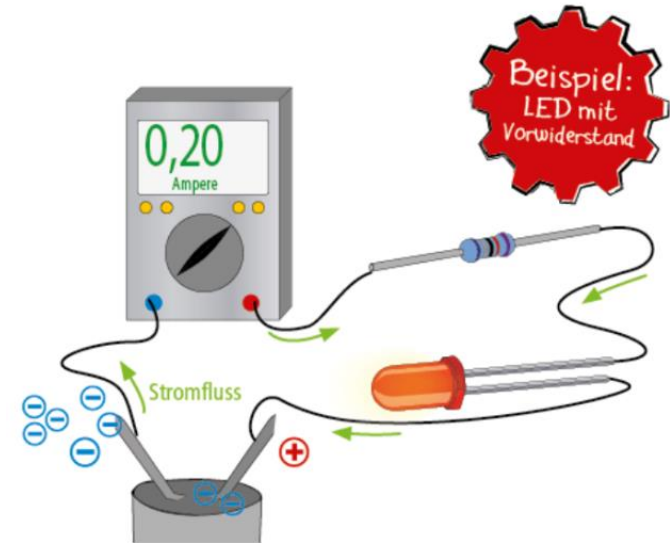
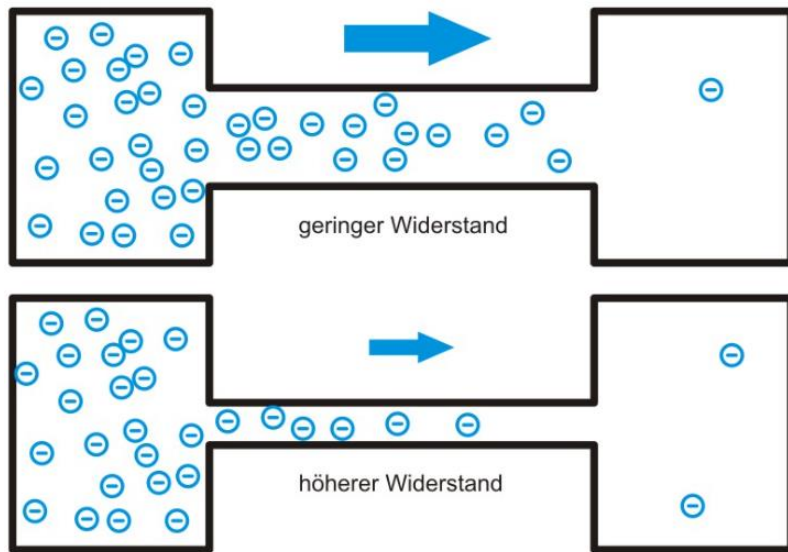
Dann stellt er zur Auswahl „Atmega328P“ (für neuen Bootloader) oder „Atmega328P (Old Bootloader)“.



Updaten des Boardverwalters:



# Widerstände



# Widerstände

Widerstandswerte – Farbcode:



**6-Band**  $274 \times 10^0 \pm 2 = 274 \Omega \pm 2\%, 250 \text{ ppm/K}$

Color	1st Digit	2nd Digit	3rd Digit	Multiplier	Tolerance	Temperature Coefficient
Black	0	0	0	1 $\Omega$		250 ppm/K
Brown	1	1	1	10 $\Omega$	$\pm 1\%$	100 ppm/K
Red	2	2	2	100 $\Omega$	$\pm 2\%$	50 ppm/K
Orange	3	3	3	1k $\Omega$		15 ppm/K
Yellow	4	4	4	10k $\Omega$		25 ppm/K
Green	5	5	5	100k $\Omega$	$\pm 0.5\%$	20 ppm/K
Blue	6	6	6	1M $\Omega$	$\pm 0.25\%$	10 ppm/K
Violet	7	7	7		$\pm 0.1\%$	5 ppm/K
Grey	8	8	8			1 ppm/K
White	9	9	9			
Gold				0.1 $\Omega$	$\pm 5\%$	
Silver				0.01 $\Omega$	$\pm 10\%$	

**4-Band**  $12 \times 10^5 \pm 5\% = 1,200 \text{ k}\Omega \pm 5\%$

**5-Band**  $100 \times 10^2 \pm 1\% = 10,000 \Omega \pm 1\%$

Widerstandsrechner:

<https://draeger-it.blog/widerstandsrechner/>

Widerstandswerte – E-Reihe:

<https://de.wikipedia.org/wiki/E-Reihe>



# Multimeter



# Oszilloskop Hantek 6022BE



Hantek 6022BE USB Digital Oszilloskop, 2-Kanal, 20MHz Storage

Prime Music das erste Mal hören und 5€ Gutscheincodes erhalten. Bedingungen gelten. [Mehr erfahren](#)

von Hantek



62 Kundenrezensionen | 23 beantwortete Fragen

Amazon's Choice für "hantek 6022be"

Preis: EUR 70,94 GRATIS Premiumversand Langstrecke für Prime-Mitglieder [Details](#)



Das Oszilloskop wird über ein USB-Kabel an einen Computer angeschlossen.  
Der Computer sorgt für die Anzeige und stellt die Betriebsspannung zur Verfügung (kein Netzteil notwendig).

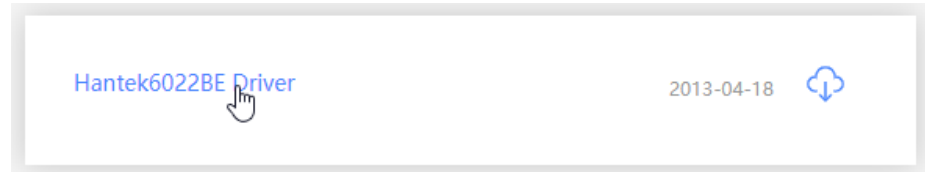
Über die Webseite von Hantek muß die Betriebs-Software heruntergeladen werden.

# Oszilloskop Hantek 6022BE – Download der Software

Webseite von Hantek:

[http://www.hantek.com/en/PagesFW\\_Vfwsc.html](http://www.hantek.com/en/PagesFW_Vfwsc.html)

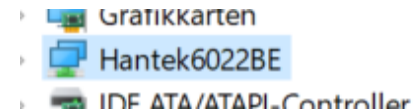
<http://www.hantek.com/download?word=6022be&sid=0&pid=0&key=fwsc>



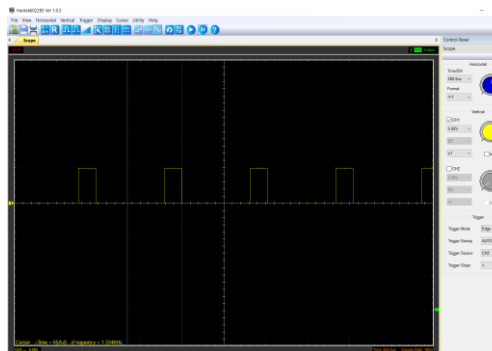
Ein Icon wird auf dem Desktop erstellt:



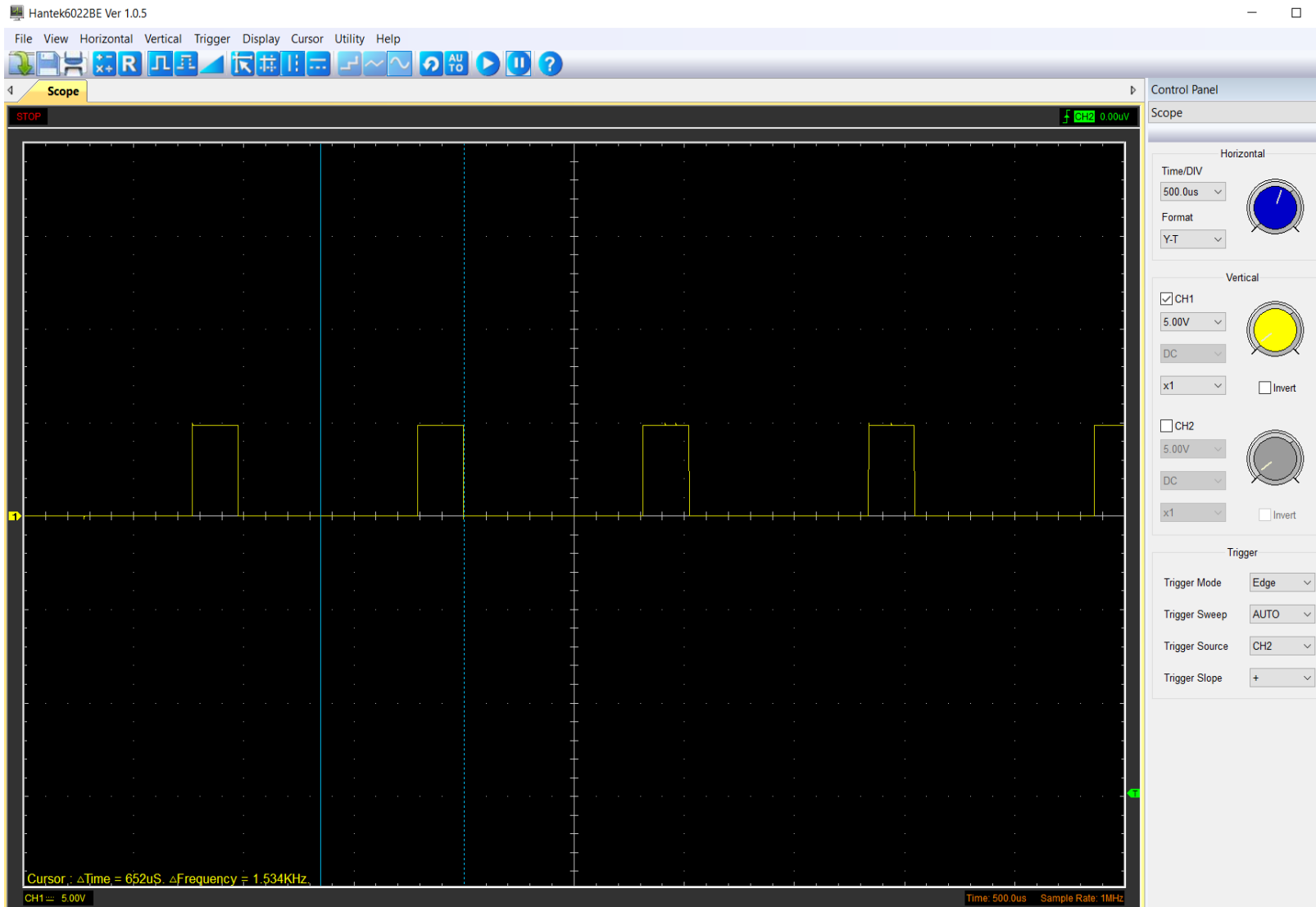
Nach Anschluß des Hantek über USB-Kabel erscheint im Geräte –Manager:



Bildschirm nach Klicken des Icons:



# Oszilloskop Hantek 6022BE – Bildschirm (Screen)



# Anschluß Hantek an Meßobjekt - Hinweise

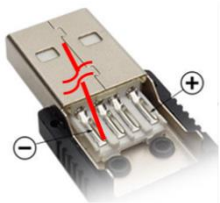


Hantek:

Schirm ist NICHT verbunden mit GND-Kontakt (-) der USB-Buchse.  
Meßleitung aussen ist verbunden mit Schirm der USB-Buchse.

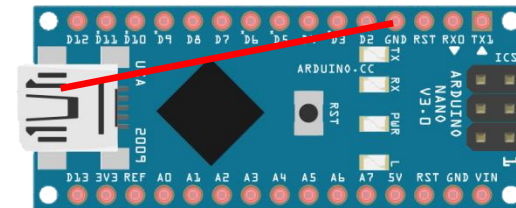
Arduino Nano:

Schirm der USB-Buchse ist verbunden mit GND des Arduino.  
GND des Arduino ist verbunden mit GND-Anschluß (-) der USB.



USB Kabel:

Schirm ist NICHT verbunden mit GND-Leitung (-).



➔ Wenn die USB-Kabel von Arduino Nano und Hantek an denselben Computer angeschlossen sind:

- wird der äußere Kontakt der Messleitung nicht benötigt (da der Kontakt schon vorhanden ist über die Schirme der beiden USB).
- Darf der äußere Kontakt nicht an ein Potential abweichend von GND angeschlossen werden.

# Messung mit Hantek - Hinweise

Time (Zeitbasis) Kann eingestellt werden zwischen 1ns und 5000s (jeweils pro DIV , zwischen gestrichelten Linien).

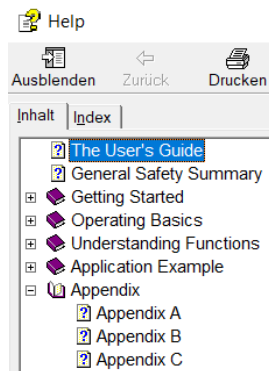
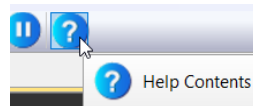
Sample rate (Abtastrate) Passt das Hantek automatisch an die gewählte Zeitbasis an.  
Beide Werte werden rechts unten angezeigt:



Der Maximalwert ist 48MHz , der Minimalwert 100kHz

Speicherumfang Das Hantek kann max 1 Million Messwerte speichern.  
Das bedeutet z.B.: Wenn ein langsamer Vorgang dargestellt werden soll und die Zeitbasis zum Beispiel auf 2s eingestellt wird, werden nur noch 5 DIV angezeigt (100kHz sind 100 Werte pro Sekunde mal 2 Sekunden mal 5 = 1 Million Werte).

Help Hilfe ist im Menü aufrufbar, unter Appendix A (Anhang A) enthält sie auch Specifications (Daten).



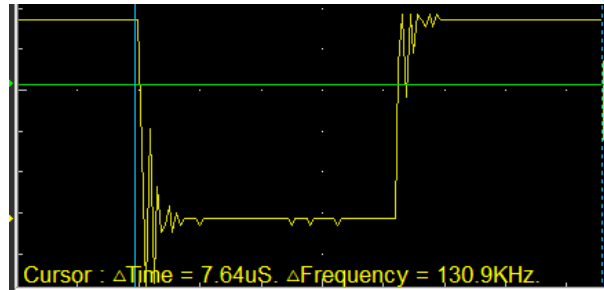
## Specifications Table:

Input	
Max.sample rate	Real-time sampling:48MS/s
Channels	2 Channels
Bandwidth	20MHz (-3dB)
Vertical resolution	8 bits/channel
Gain error	20mV - 5V/div @ x1 probe/

# Arduino - wie lange dauert der Loop-Durchlauf ?

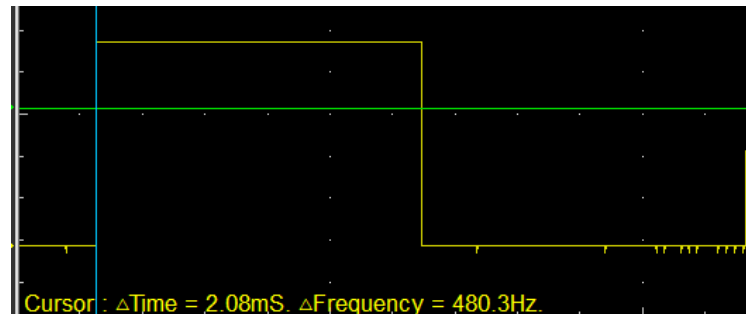
Messungen mit Oszilloskop Hantek:

```
void setup() {  
  pinMode (2, OUTPUT);  
}  
void loop() {  
  digitalWrite (2,HIGH);  
  digitalWrite (2,LOW);  
}
```



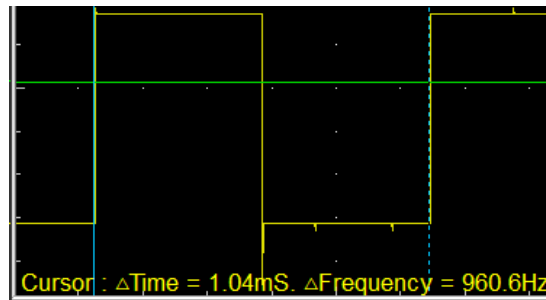
Bei diesem Sketch wird der Loop 131.000 mal pro Sekunde durchlaufen

```
void setup() {  
  pinMode (2, OUTPUT);  
  Serial.begin (9600);  
}  
void loop() {  
  digitalWrite (2,HIGH);  
  Serial.print (2);  
  digitalWrite (2,LOW);  
  Serial.print (2);  
}
```



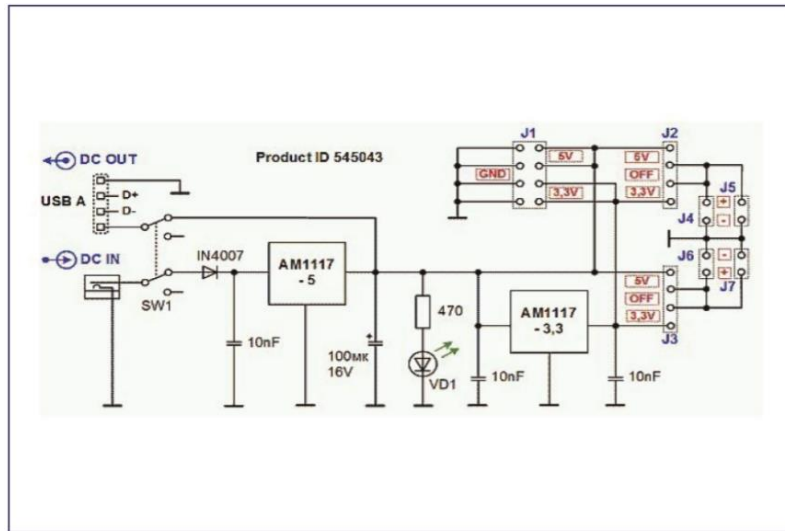
Wenn ein Serial.print-Befehl eingefügt wird, erhöht das enorm die Durchlaufzeit.

```
void setup() {  
  pinMode (2, OUTPUT);  
  Serial.begin (19200);  
}  
void loop() {  
  digitalWrite (2,HIGH);  
  Serial.print (2);  
  digitalWrite (2,LOW);  
  Serial.print (2);  
}
```



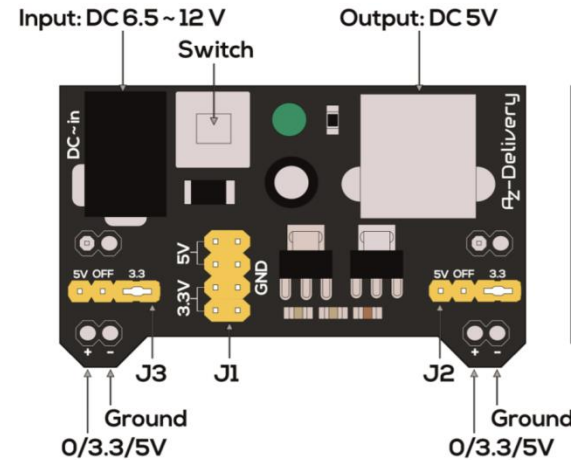
Eine Erhöhung der Übertragungsrate reduziert die Durchlaufzeit.

# MB102 Breadboard Module Adapter Shield 3.3V/5V

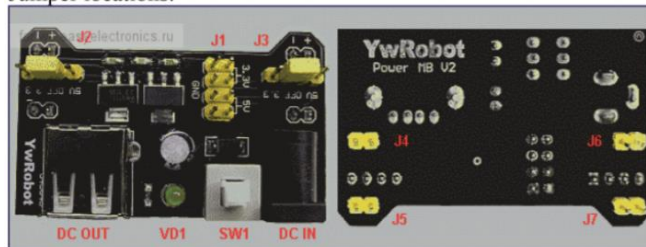


USB-Buchse nur für Spannungs-Ausgang !

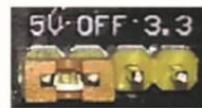
Max .Strombelastbarkeit: 700 mA



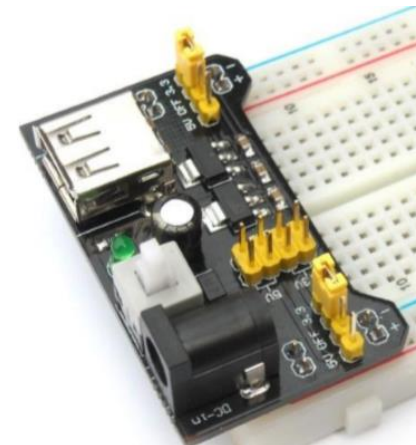
Jumper locations:



3.3 Volt Configuration



5 Volt Configuration





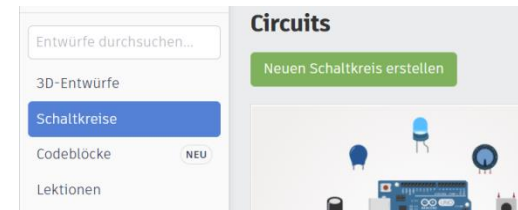
# Bezugsquellen (Auswahl)

---

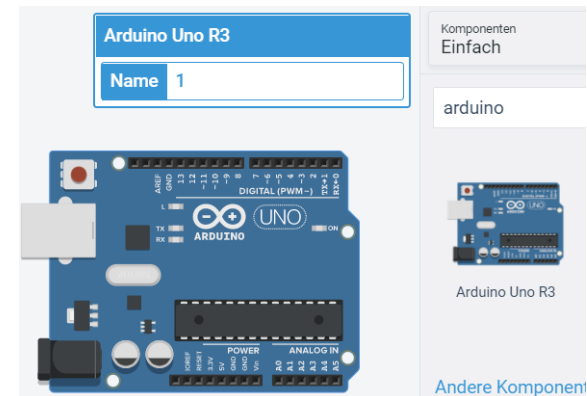
- Amazon
- [www.amazon.de](http://www.amazon.de)
  
- AZ-Delivery (in Deutschland)
- <https://www.az-delivery.de/?ls=de>
  
- Eckstein Komponente (in Deutschland)
- <https://eckstein-shop.de>
  
- Conrad
- <https://www.conrad.de/>
  
- Ali-Express (Lieferung aus China ; Webseiten in in englisch)
- <https://best.aliexpress.com/>
  
- Banggood (Lieferung aus China ; Webseiten in in englisch)
- <https://www.banggood.com>
  
- RobotShop (Lieferung aus USA ; Webseiten auch in deutsch)
- <https://www.robotshop.com/>
  
- Tomtop (China)
- <https://www.tomtop.com/de/>



- Die Tinkercad-Plattform von Autodesk ist eine umfangreiche Online-Plattform, mit der man 3D-Entwürfe, elektrische Schaltungen sowie Arduino-Sketche erstellen und deren Funktion simulieren kann.
- Erster Schritt: Kostenlose Registrierung auf <https://www.tinkercad.com>
- Nach Registrierung ruft man die Website auf:
- <https://www.tinkercad.com/dashboard>
- Links anklicken „Neuen Schaltkreis erstellen“:




- Rechts oben bei „Suchen“ eingeben „arduino“, dann das Bild nach links ins Hauptfeld ziehen:



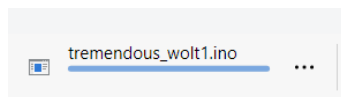
- Siehe auch:  
[Arduino UNO Tutorial 1: Einführung, LED blinken \(tinkercad simulator\) \[Deutsch\] - YouTube](#)

- Klicken auf „Code“ und auswählen „Text“:



- Jetzt wird bereits ein Sketch angezeigt (Sketch „Blink“).
- Er kann bearbeitet und dann gestartet werden („Simulation starten“).
- Der serielle Monitor ist ebenfalls simulierbar (ganz unten).
- Mit Button  kann der Sketch auf den PC runtergeladen werden. Er wird als .ino Datei abgelegt und kann mit der Arduino IDE geöffnet werden.

links unten:

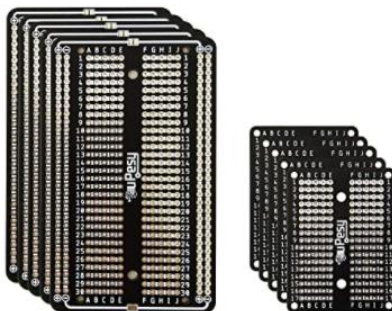
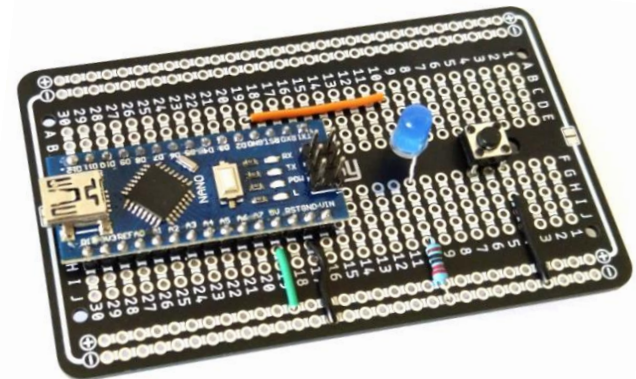


- Mit „Freigeben“ wird eine gemeinsame Bearbeitung des Projekts (hier des Sketchs) möglich.



# Lötbares Breadboard (Leiterplatte)

- Alternativ zu steckbaren Breadboards könnte man auch Lötbares Breadboards verwenden.
- Vorteil: Sichere Verbindung, Jumper-Kabel können nicht rausrutschen
- Nachteile: Umfangreiches Löten erforderlich
- Module sollten auf Adapter gesetzt werden, sonst wird der Austausch bei defekten schwierig



10 x Prototype Board, Lötbares Breadboard, Doppelseitig PCB Circuit Board (Platinen set, Leiterplatte, Lochrasterplatte Kit, Stripboard, Protoboard) for solder Arduino & DIY elektronik Projekt

Marke: uPesy

★★★★★ 47 Sternebewertungen

Preis: 18,99 € (1,90 € / Stück)

Preisangaben inkl. USt. Abhängig von der Lieferadresse kann die USt. an der Kasse variieren. Weitere Informationen.

50 € Startguthaben und keine Jahresgebühr: die **Barclaycard Visa Kreditkarte**. Mehr erfahren.

Größe: **Mini+Medium**

Medium

**Mini+Medium**

Ende  
Teil 1

[Inhaltsverzeichnis](#)