

# Arduino Workshop - Labortage 2019: Ach ... Ich lass das jetzt so!

## Elektrotechnische Grundlagen

### Ohm'sches Gesetz

Das Ohm'sche Gesetz ist eine der wichtigsten Grundlagen und beschreibt das Zusammenspiel von Strom, Spannung und Widerstand in einem Schaltkreis. Das heißt, wenn eine gewisse Spannung anliegt wird bei einem definierten Widerstand ein Strom fließen. Dieser Strom ist das Ergebnis von Spannung geteilt durch Widerstand. Die Formeln zum Errechnen der Werte findet sich weiter unten in der Umrechnungstabelle.

### Elektrische Leistung

Der Strom kann eine gewisse Leistung liefern. Diese Leistung wird in Watt angegeben und errechnet sich als  $P = U \cdot I$ . Eine Leistung über eine gewisse Zeit (t) nennt man Energie und diese wird in Wattstunden (Wh) angegeben.

### Umrechnungstabelle

Name	Formelzeichen	Einheit	Berechnung	Beispiel
Strom	I	Ampere (A)	$I = U / R$	$0,01 \text{ A} = 3 \text{ V} / 300 \Omega$
Spannung	U	Volt (V)	$U = R \cdot I$	$3 \text{ V} = 0,01 \text{ A} \cdot 300 \Omega$
Widerstand	R	Ohm ( $\Omega$ )	$R = U / I$	$300 \Omega = 3 \text{ V} / 0,01 \text{ A}$
Zeit	t	Sekunden (s)	---	---
Elektrische Leistung	P	Watt (W)	$P = U \cdot I$	$0,03 \text{ W} = 3 \text{ V} \cdot 0,01 \text{ A}$
Elektrische Energie	E	Wattstunden (Wh)	$E = P \cdot t$	$0,03 \text{ Wh} = 0,03 \text{ W} \cdot 3600 \text{ s}$

Die elektrische Energie kann man auch in Joule als alternative Energieeinheit angeben. Die Umrechnung ist wie folgt:

1 kWh	3.600.000 J
1 J	$2,778 \cdot 10^{-7} \text{ kWh}$

### Vorwiderstand für eine LED berechnen

In dem Datenblatt einer LED steht, welche Spannung und welchen Strom eine LED benötigt.

Der genaue Verbrauch variiert je nach Halbleitermaterial der LED und Wellenlänge des Lichtes (Farbe).

Soll eine LED bei 2 V mit 10 mA betrieben werden, muss ein Vorwiderstand verwendet werden, da die LED sonst durchbrennt. Das heißt bei 5 V Versorgungsspannung aus dem Arduino müssen 3 V an Spannung abfallen (damit 2 V verbleiben, um die LED zu versorgen). Rechnung:  $3 \text{ V} / 0,01 \text{ A} = 300 \Omega$ . Bei 10 mA brauchen wir einen  $300 \Omega$  Widerstand. Wir müssen also einen Widerstand mit  $300 \Omega$  mit der LED in Reihe schalten, damit diese ihre volle Helligkeit erreicht. Je nach LED sollte der Strom 20 mA bis 50 mA nicht überschreiten.

## Begriffserklärung

HIGH – Wenn ein Spannungspegel der Versorgungsspannung entspricht (Arduino: 5V)

LOW – Wenn der Spannungspegel GND entspricht (Arduino: 0V)

RISING EDGE – Der Augenblick, wenn ein digitales Signal von LOW zu HIGH wechselt

FALLING EDGE – Der Augenblick, wenn ein digitales Signal von HIGH zu LOW wechselt

## Widerstandstabelle

5 Ringe	1. Ring (Hunderter)	2. Ring (Zehner)	3. Ring (Einer)	4. Ring (Multiplikator)	5. Ring (Toleranz)
4 Ringe		1. Ring (Zehner)	2. Ring (Einer)	3. Ring (Multiplikator)	4. Ring (Toleranz)
Silber				$10^{-2}$	
Gold				$10^{-1}$	
Schwarz		0	0	$10^0$	
Braun	1	1	1	$10^1$	± 1%
Rot	2	2	2	$10^2$	± 2%
Orange	3	3	3	$10^3$	
Gelb	4	4	4	$10^4$	
Grün	5	5	5	$10^5$	± 0.5%
Blau	6	6	6	$10^6$	± 0.25%
Violett	7	7	7		± 0.1%
Grau	8	8	8		± 0.05%
Weiß	9	9	9		

## Code Referenz

### delay(ms)

Delay verzögert die weitere Ausführung des Codes an der Stelle um die Anzahl der Millisekunden, die der Funktion übergeben werden.

### pinMode(pin, INPUT/OUTPUT/INPUT\_PULLUP)

Stellt ein, ob der Pin als Aus- oder Eingabe funktionieren soll.

### digitalWrite(pin, HIGH/LOW)

Setzt den digitalen Pin entweder auf HIGH oder auf LOW.

### digitalRead(pin)

Gibt zurück, ob der Spannungspegel an dem angegebenen digitalen Pin HIGH oder LOW ist.

### analogWrite(pin, value)

Stellt die Pulsweitenmodulation (PWM) ein. Der Wert dafür kann zwischen 0 und 255 liegen. Ausnahmen gelten dafür nur bei dem Arduino Zero oder dem Arduino MKR. Weitere Informationen findet man in der Arduino Dokumentation, welche am Ende des Handouts angegeben ist.

### analogRead(pin)

Gibt den Spannungspegel an dem Pin als Wert zwischen 0 und 1023 aus. (10 Bits)  
Der Wert ist 1023 wenn 5V anliegen und 0 bei Ground (GND).

### Serial.begin(9600)

Aktiviert die serielle Kommunikation zwischen Arduino und Computer.  
Im Serial Monitor ist eine Baud-Rate von 9600 anzugeben.

Achtung: Bei Kommunikation zwischen Arduino und PC dürfen die Pins 0 und 1 (TX und RX) nicht anderweitig verwendet werden!

### Serial.print(str) / Serial.println(str)

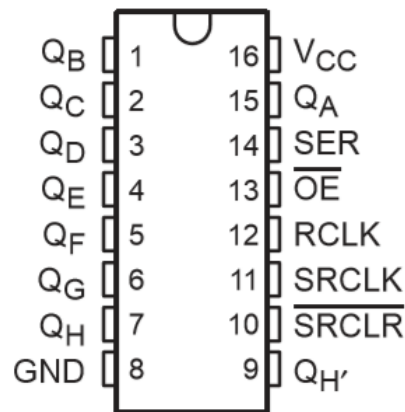
Gibt den Text auf dem Serial Monitor in der Arduino IDE aus. Die „println“-Variante erzeugt gleichzeitig eine neue Zeile.

## Pinbelegung der Bauteile

### Shiftregister

Bauteilbezeichnung: 74LS595

Hersteller: Texas Instruments



<https://www.idogendel.com/en/archives/211>

Name	Beschreibung
Vcc	Versorgungsspannung (+5V)
GND	Ground (Masse)
QA – QH	8-Bit Datenausgabe
QH'	Serielle Ausgabe zum Verketteten mehrerer Shift Register
SER	Serielle Eingabe
$\overline{OE}$	Daten werden ausgegeben, wenn der Pin LOW ist. Ist der Pin HIGH werden keine Daten aus den Pins QA bis QH zur Verfügung gestellt.
RCLK	Register Clock
SRCLK	Serial Clock
$\overline{SRCLR}$	Wenn der Wert LOW ist, wird das Register geleert

Ein Shiftregister besteht aus dem eigentlichen Shiftregister und einem internen Datenregister. Durch das Shiftregister werden Daten durchgereicht und im internen Datenregister können Daten länger vorgehalten werden. Der Verwendungszweck eines Shiftregisters besteht darin mehrere Pins zur Verfügung zu bekommen. Die Anzahl der IO Pins ist limitiert. Schließt man ein Shiftregister an, kann (je nach Register) eine Menge von drei Pins des Arduino verwendet werden um 8 Pins anzusteuern. Dies erlaubt es die Anzahl der Pins „zu erhöhen“ – dafür muss man lediglich etwas Performance eintauschen. Die Verarbeitung im Prozessor wird wegen des Verwaltungsaufwandes und der Zeiten, die es braucht Daten in die Register zu schreiben langsamer. Man kann jedoch auch beliebig viele Register hintereinander anschließen.

Bei einer RISING EDGE auf SRCLK wird das Signal von SER übernommen.

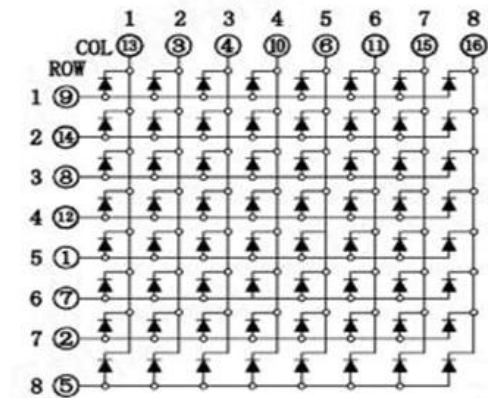
Bei einer RISING EDGE auf RCLK werden die übernommenen 8 Bit in das Register kopiert.

Bei einem HIGH-Signal auf dem OE (bzw. LOW bei einem invertiertem OE) werden die im Register gespeicherten Daten auf den Pins QA-QH ausgegeben.

Falls mehr als 8 Bit über SER ankommen, werden die zuerst gesendeten Daten über QH' aus dem IC wieder herausgeschoben, z.B. an ein weiteres angeschlossenes Shiftregister.

## LED Matrix

Bauteilbezeichnung: 1088AS

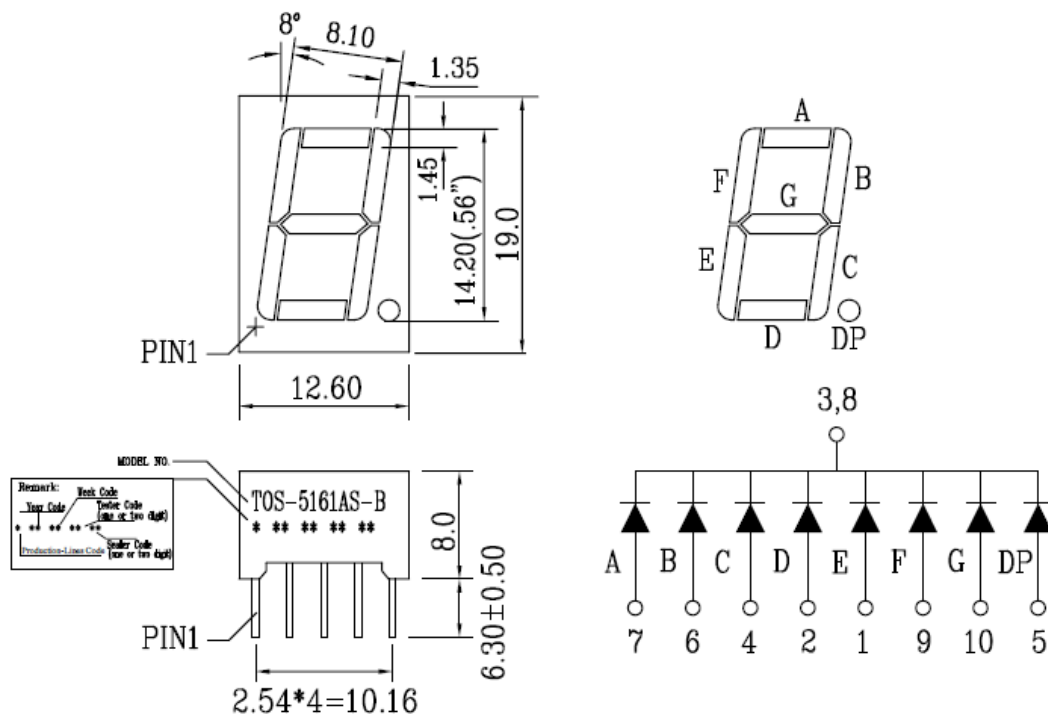


<https://rarecomponents.com/store/1484>

Die Pin-Reihenfolge (von oben gesehen, Bauteilbezeichnung frontal) von unten links (Pin 1) nach unten rechts (Pin 8) und oben rechts (Pin 9) nach oben links (Pin 16).

## 7 Segment-Anzeige

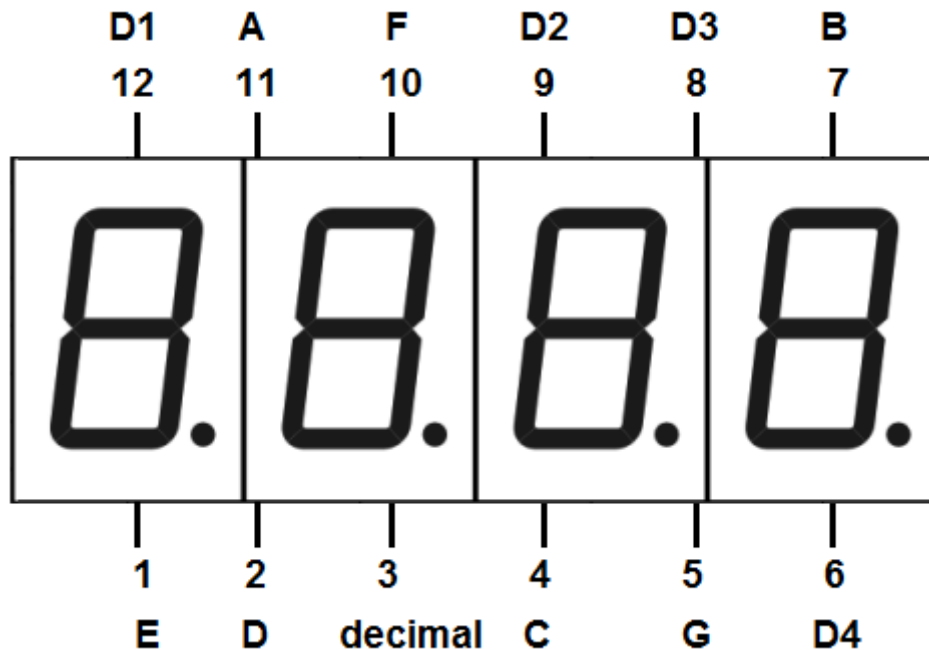
Bauteilbezeichnung: 5611AH



<http://prometec.org/displays/7-segment-display/>

## 4-fach 7 Segment-Anzeige

Bauteilbezeichnung: 5641AS-1



<http://www.learningaboutelectronics.com/Articles/4-digit-7-segment-LED-circuit-with-an-arduino.php>

## Weitere Ressourcen

[arduino.cc](http://arduino.cc)

Die Website von Arduino selbst. Dort gibt es weitere Ressourcen zu Arduino generell und ein Benutzerforum. Ebenso findet sich dort der Download zur Arduino IDE.

[arduino.cc/reference](http://arduino.cc/reference)

Hier findet sich eine Dokumentation zu der Programmierung von Arduinos, Codebeispiele und Hilfestellungen.

[mikrocontroller.net](http://mikrocontroller.net)

MikrocontrollerNet ist eine Website für an Elektronik interessierte mit einem großen Forum zum Austausch und zur Kommunikation.

[elektronik-kompodium.de](http://elektronik-kompodium.de)

Im Elektronik Kompodium kann man grundlegende Formeln und Techniken nachlesen und bekommt Schaltungen erklärt. Diese Seite ist ein generelles Wiki für Elektrotechnik.