

# **Arduino für Anfänger**

Elektronik, Programmierung, Praxis

---

Philip Caroli

9. Februar 2017

Fablab Karlsruhe e.V.

# Übersicht

## **Was ist ein Arduino?**

---

# Was ist ein Arduino?

- Kleine, günstige Platine.
- Mikrocontroller: Kleiner Rechner.
- Einfach über USB anzuschließen.
- Leicht zu programmieren.
- Reicht für kleine und mittelgroße Projekte vollkommen aus.
- Große Community:
  - Viele Probleme wurden schon gelöst.
  - Es gibt bewährte Lösungen.
  - “Arduino-Kompatible” Komponenten.



# Was kann man damit machen?

- Mit dem Computer kommunizieren.
- Leuchtdioden leuchten lassen.
- Motoren, Lautsprecher u.A. steuern.
- Temperatur-, Feuchtigkeits-, Lichtsensoren auslesen.
- Fertige Module benutzen:
  - Kleine Platine mit modernen Komponenten
  - Fertige Beispiele ⇒ Schnelles Experimentieren.

# Was kann man damit nicht machen?

- Bildbearbeitung.
- Komplexe Programme.
- Große WLAN- und Bluetoothanwendungen.

# **Elektronik-Grundlagen**

---

# Elektronik-Grundlagen

- Elektronik-Grundverständnis ist unabdingbar für Arduino-Projekte.
- Gefahren durch Strom:
  - Stromschlag: Keine Gefahr bei Niederspannung.
  - Brandgefahr: Umwahrscheinlich.
  - Heiße Bauteile: Möglich.
- In diesem Kurs
  - Nur grobe Einführung in die Materie.
  - Keine Erklärung der Hintergründe und Funktionsweise.
  - grauer Text: weiterführende Begriffe, auf die nicht eingegangen wird.

# Strom und Spannung

- Anschauung zum einfacheren Verständnis:
  - Luftfluß durch ein Rohrsystem.
  - Luftdruck wird durch einen Kompressor in Tank A aufgebaut.
  - Luft entströmt aus Tank A durch ein Rohr in die Umgebung.
  - Je höher der Luftdruck, desto größer der Luftstrom.
  - Wenn keine Druckdifferenz zwischen Tank und Umgebung besteht, strömt keine Luft.
  - Bei doppelter Druckdifferenz ist der Luftstrom doppelt so groß.

# Spannung - Theorie

Luftstrom-Äquivalent: Luftdruck

- Formelzeichen: U, selten V
- Einheit: Volt [V]
- Wichtig ist immer die Spannungsdifferenz.
- meist wird die Spannungsdifferenz auf die Masse bezogen.
  - "5 V" bedeutet also, dass die Spannung 5V über der Masse liegt.
  - Masse wird mit GND abgekürzt und hat die Symbole    
  - Luftstrom-Äquivalent: Umgebungsdruck.
- Auch negative Spannungen (zur Masse) sind möglich.
- Spannung kann entweder konstant anliegen (Gleichspannung, DC) oder mit einer bestimmten Frequenz (Wechselspannung, AC) ihre Polung wechseln.
- Hohe Spannungen wie 230 V AC sind lebensgefährlich.
- Spannungen unter 60V DC gelten als ungefährlich.

# Spannung - Theorie

- Alle elektrischen Projekte benötigen mindestens eine Spannungsversorgung - der Arduino kann einfach über USB mit 5V versorgt werden.
- Die meisten elektrischen Module wie der Arduino arbeiten mit 5V oder mit 3,3V.
- Durch eine zu hohe Spannung können elektrische Komponenten zerstört werden - auch wenn kein Strom fließt.

# Strom - Theorie

Luftstrom-Äquivalent: Luftstrom

- Formelzeichen: I
- Einheit: Ampere [A]
- Fließt von einer Stelle mit hoher Spannung zu einer mit niedriger Spannung.
- Benötigt einen elektrischen Leiter zum fließen.
- Zu hohe Ströme können elektrische Komponenten zerstören.
- Fließt ein Strom ungehindert von einer Spannungsquelle zu einer Senke, so spricht man von einem Kurzschluss  $\Rightarrow$  Dieser kann die Spannungsquelle beschädigen.

# Widerstände - Theorie

Luftstrom-Äquivalent: Rohr mit Verengung

- Formelzeichen: R
- Einheit: Ohm [ $\Omega$ ]
- Setzt dem elektrischen Strom einen Widerstand entgegen.
- Je größer der Widerstand, desto weniger Strom fließt bei einer gegebenen Spannung.
- Über einem Widerstand fällt bei Durchfluss eine Spannung ab.
- Ein Kupferdraht hat einen sehr geringen Widerstand.

# Widerstände - Ohm'sches Gesetz

Das Ohm'sche Gesetz beschreibt den Zusammenhang zwischen Widerstand, Spannung und Stromstärke:

$$R = \frac{U}{I}, \text{Widerstand} = \frac{\text{Spannung}}{\text{Strom}}$$

- Eselsbrücke zum merken: Rudi  $\Rightarrow$  R gleich U durch I
- Andere Formen:  $I = \frac{U}{R}$ ,  $U = R * I$
- Wichtigste Formel in der Elektrotechnik

# Widerstände - Praxis

- Widerstände gibt es in vielen Größen und Arten.
- Wenn ein Widerstand von Strom durchflossen wird, entsteht Wärme.
- Widerstände haben keine Polarität: Es ist egal in welcher Richtung sie durchflossen werden.
- Bei Drahtwiderständen ist der Widerstandswert farbcodiert aufgedruckt.
- Die hier verwendeten Widerstände vertragen 0,25W.
- Die Wärmeleistung (in Watt [W]) wird so berechnet:  $P = U * I$

# Kondensatoren

Luftstrom-Äquivalent: Gasbehälter mit Membran

- Formelzeichen: C
- Einheit: Farad [F]
- Speichert Energie in einem elektrischen Feld.
- Je höher die anliegende Spannung, desto mehr Energie wird gespeichert.
- Werden oft zur Glättung von Spannungen verwendet.
- Elektrolytkondensatoren (Elkos) haben eine Polarität: Bei negativer Spannung explodieren sie.

# Spulen

Luftstrom-Äquivalent: Turbine mit Schwungrad

- Formelzeichen: L
- Einheit: Henry [H]
- Speichern Energie in einem elektrischen Feld.
  1. Feld wird aufgebaut: Hoher Widerstand.
  2. Feld ist aufgebaut: Niedriger Widerstand.
  3. Feld wird abgebaut: Spannungsdifferenz wird erzeugt.
- Werden in Elektromotoren verwendet.

# Dioden

Luftstrom-Äquivalent: Rückschlagventil

- Leiten Strom nur in einer Richtung.
- Haben eine Durchlassspannung, unter der sie (fast) nicht leiten.
- Über ihnen fällt eine Bauartbedingte Spannung ab: Bei Siliziumdioden zB 0,7V.
- Bei kleinem Spannungsanstieg vergrößert sich der Durchflusstrom sprunghaft.
- Bei Durchfluss entsteht Wärme, ähnlich wie bei Widerständen.
- Werden bei höherer Temperatur leitfähiger  $\Rightarrow$  Teufelskreis
- Bei zu viel Wärme werden Dioden zerstört.

# Leuchtdioden (Light emitting Diodes, LEDs)

- Sind Dioden, die außer Wärme auch Licht ausstrahlen.
- Typischer Wirkungsgrad: 20' %  $\Rightarrow$  80% der Leistung ist Wärme.
- "Typische" LEDs sind für einen Stromfluss von 20mA spezifiziert.
- Haben eine Sperrspannung von 5V.
- Werden mit einem Vorwiderstand geschützt:

Farbe	Spannung	min. Vorwiderstand bei 5V
Infrarot	1,5V	390Ω
Rot	1,6V	390Ω
Orange	2,0V	330Ω
Gelb	2,2V	330Ω
Grün	2,1V	330Ω
Blau	2,9V	220Ω
Weiβ	4,0V	100Ω

# (Feldeffekt)Transistoren

Luftfluß-Äquivalent: Druckgesteuertes Ventil

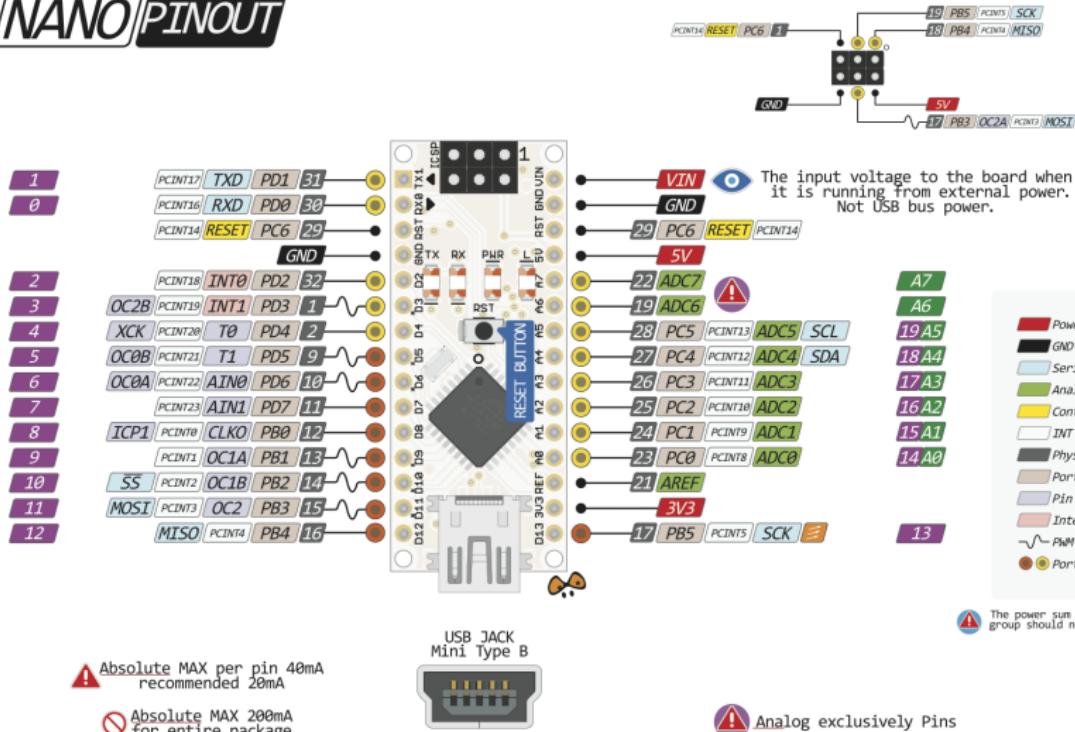
- Haben 3 Anschlüsse: Gate, Source, Drain.
- Werden als Schalter verwendet.
- Schalten durch, sobald an Gate eine Spannung anliegt.
- Können benutzt werden, um hohe Spannungen und starke Strom zu schalten.

# Der Arduino als elektrisches Bauteil

- Stromversorgung über USB-Kabel: 5V, maximal 500mA.
- Stromversorgung über VIn-Pin: 6 bis 20V.
- Stromversorgung über 5V-Netzteil: maximal 1A.
- Erzeugt eine 3,3V-Spannung, Belastbarkeit 50mA.
- Besitzt 22 digitale Eingabe/Ausgabe-Pins:
  - Eingabe: Können messen, ob eher Masse oder 5V anliegt.
  - Haben einen einschaltbaren hochohmigen Widerstand nach 5V
  - Ausgabe: Können Masse oder 5V oder nichts ausgeben.
  - Vertragen jeweils maximal 40mA, alle zusammen nicht mehr als 200mA.

# Der Arduino als elektrisches Bauteil

## NANO PINOUT



## Löten

---

# Löten

- Der Arduino Nano muss noch zusammengelötet werden.
- Dafür die Lötstation auf 300 C einstellen und eine Minute warten.
- Die langen Stiftleisten von unten einstecken, sodass der USB-Anschluss oben liegt.
- Mit dem Lötkolben eine Lötstelle 2-3 Sekunden erwärmen.
- Lötzinn hinzufügen, 2-3 Sekunden warten.
- Lötkolben wegnehmen, Lötstelle begutachten.
- Die Lötstelle muss glänzen und den Pin mit der Platine verbinden.



# Programmiergrundlagen

---

# Die Entwicklungsumgebung

- Kann unter <https://www.arduino.cc/en/Main/Software> heruntergeladen werden.
- ist für Windows, Linux und MacOS verfügbar.
- Ubuntu-Paketverwaltung beinhaltet nur alte Version - bei Problemen trotzdem sudo apt-get install arduino-core ausführen.
- Windows: Unbedingt auch die USB-Treiber installieren lassen.
- Beinhaltet viele nützliche Beispiele.
- Programme werden meist “Sketch” (Skizze) genannt.
- USB-Treiber für Arduinos werden auch mitinstalliert.

# Test nach der Installation

1. Arduino IDE starten.
2. Datei ⇒ Beispiele ⇒ 01.Basics ⇒ Blink auswählen.
3. Beide `delay(1000);` in `delay(250);` ändern.
4. Tools ⇒ Boards ⇒ Arduino Nano auswählen.
5. Unter Tools ⇒ Port den richtigen / einzigen Port wählen.
6. Mit Sketch ⇒ Upload oder [Strg]+[U] den Sketch auf den Arduino laden.
7. Der Sketch muss nicht gespeichert werden.
8. Die LED auf dem Arduino blinkt jetzt schneller.

# Programmaufbau I

- Ein Programm ist eine Liste von Befehlen.
- Jeder Befehl muss korrekt geschrieben sein.
- Jeder vollständige Befehl wird mit einem Semikolon beendet.
- Grundrechenarten können direkt benutzt werden.
- Befehle sind meist Zuweisungen, zB  $A = A + B;$
- Befehle, die andere Teilprogramme aufrufen, bestehen aus einem Text und einem Klammerpaar.
- In den Klammern werden die Parameter des Befehls angegeben.
- Der Befehl gibt einen oder keinen Wert zurück.

# Kommunikation mit dem PC

Vor allem um Programme zu testen ist die serielle Kommunikation nützlich. Die vom Arduino versendeten Nachrichten können über Tools  
⇒ Serieller Monitor gelesen werden.

- `Serial.begin(9600)`: Startet die Kommunikation mit dem PC.
- `Serial.println(''Hallo!'')` : Schreibt ein "Hallo" an den PC.

# Variablen

- Um in einem Programm eine Variable zu speichern oder zu ändern, müssen wir wissen wie sie heißt und was für ein Typ sie ist.
- Eine Variable muss vor der Verwendung deklariert werden:  
TYP NAME;
- Danach wird sie initialisiert: NAME = WERT;
- Beides kann kombiniert werden: TYP NAME = WERT;
- Oft verwendete Typen:

Typ	Beschreibung	Wertebereich	Beispiel
int	ganze Zahl	-32768 - 32767	24000
float	Fließkommazahl	$\pm 3.4028235E38$	3.14
char	ganze Zahl	-128 - 127	-12
byte	ganze Zahl	0 - 255	128
string	Zeichenkette		“Hallo”
bool	Wahr oder Falsch	true, false	false

# Bedingungen

Klassische “Wenn, dann”-Abfragen.

```
if (wetter == "Regen") {  
    laune = 0;  
} else {  
    laune = 100;  
}
```

Operator	Bedeutung
==	Gleichheit
!=	Ungleichheit
>	Größer wie
<=	Kleiner oder gleich wie
&&	und
	oder

# Schleifen

Mach X-Mal folgendes:

```
for(int i = 0; i < 10; i++) {  
    Serial.println("Hallo");  
}
```

Resultat: Es wird zehn Mal "Hallo" gesendet.

# Befehle für Arduino-Pins

- `pinMode(1, INPUT);` Der Pin gibt keine Ausgangsspannung aus.
- `pinMode(1, OUTPUT);` Der Pin gibt eine Ausgangsspannung aus.
- `digitalWrite(1, LOW);` Die Ausgangsspannung ist Masse (0V).
- `digitalWrite(1, HIGH);` Die Ausgangsspannung ist 5V.
- `variable = digitalRead(1);` Die Variable ist `true`, wenn 5V am Pin anliegt, und `false` wenn Masse anliegt.

# Programm-Bibliotheken

- Bibliotheken beinhalten fertige Softwarebausteine und Beispiele, wie sie verwendet werden können.
- Sie können unter Sketch ⇒ Bibliothek hinzufügen ⇒ Bibliothek verwalten installiert werden.
- Wir installieren Hier die Bibliothek **SevSeg**, Das Suchfeld erleichtert das Finden.

## kleines Beispiel

---

# Sieben-Segment-Anzeige als Handzähler

- Wir wollen mit dem Arduino eine Zahl anzeigen.
- Mit einem Knopfdruck soll die Zahl erhöht werden.
- Zur Anzeige wird eine Sieben-Segment-Anzeige verwendet.
- Dabei Wird eine Ziffer mit sieben LEDs dargestellt.
- Mit der Multiplexing-Technik können mehrere Ziffern mit wenigen Pins dargestellt werden.
- Jede der sieben LEDs benötigt ihren eigenen Vorwiderstand.

# Reaktionszeit-Tester

- Kleines alternatives Beispielprogramm.
- Eine LED geht in einer zufälligen Zeit an.
- Der Benutzer drückt als Reaktion auf eine Taste.
- Wenn er schnell genug ist, leuchtet die LED 2s schnell blinkend.
- Wenn nicht, passiert nichts.

## setup()

```
void setup() {  
    Serial.begin(9600);  
    Serial.println("Starte das Programm!");  
    pinMode(2, INPUT);  
    digitalWrite(2, HIGH);  
  
    pinMode(3, OUTPUT);  
    digitalWrite(3, LOW);  
}
```

## **loop() I**

```
void loop() {
    delay(random(1000,3000));
    long aktion = millis();
    digitalWrite(3, HIGH);
    while(digitalRead(2) != 0) {
        delay(1);
    }
    long reaktion = millis();
```

## loop() II

```
if(reaktion - aktion <= reaktionszeit) {  
    Serial.println("Geschafft");  
    reaktionszeit = reaktionszeit * 0.9;  
    for(int i = 0; i < 10; i++) {  
        digitalWrite(3, HIGH);  
        delay(100);  
        digitalWrite(3, LOW);  
        delay(100);  
    }  
} else {  
    Serial.println("Nicht_geschafft!");  
    digitalWrite(3, LOW);  
}  
delay(1000);  
}
```