

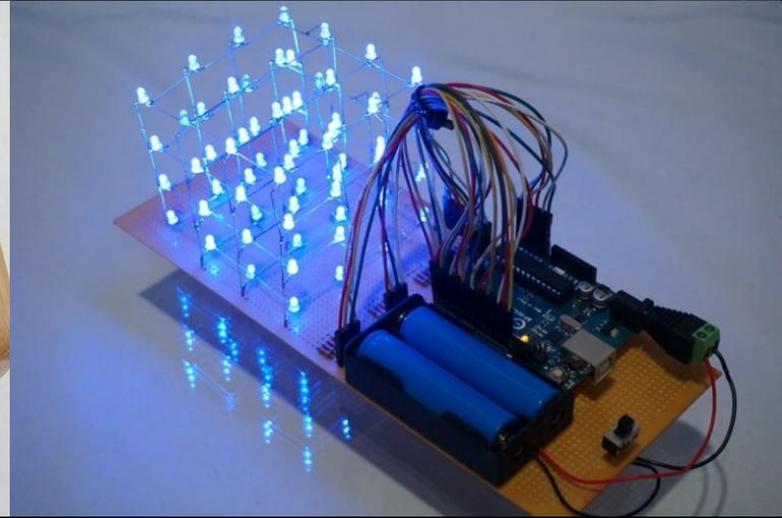
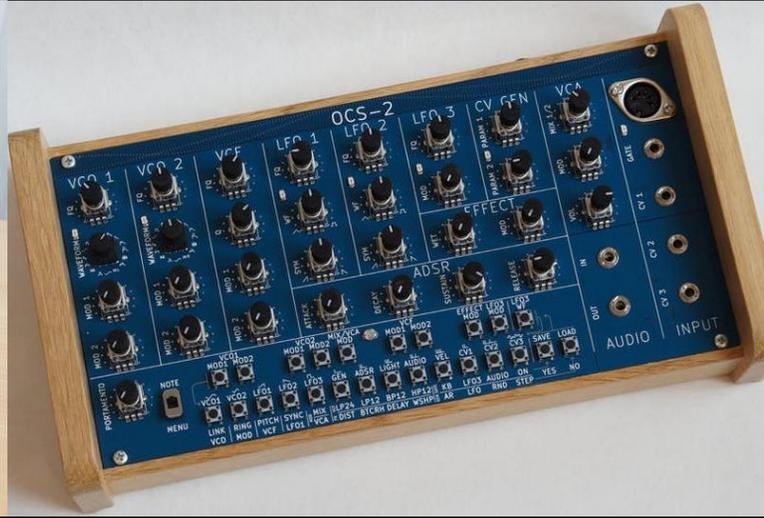
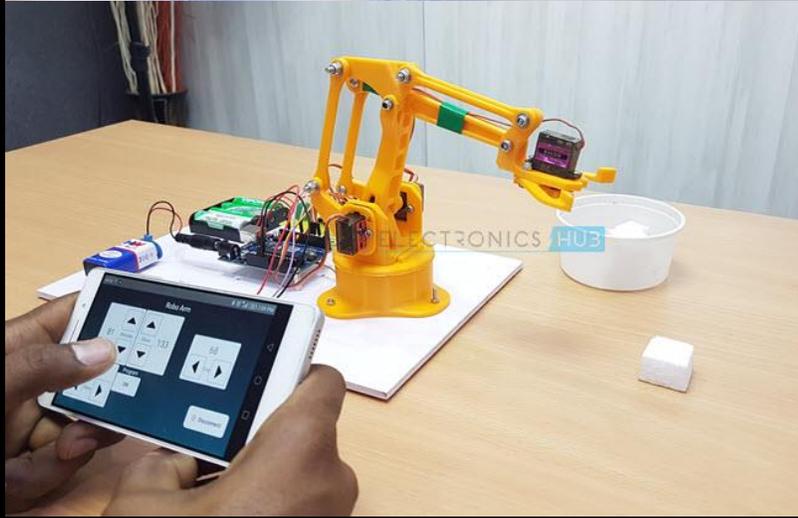
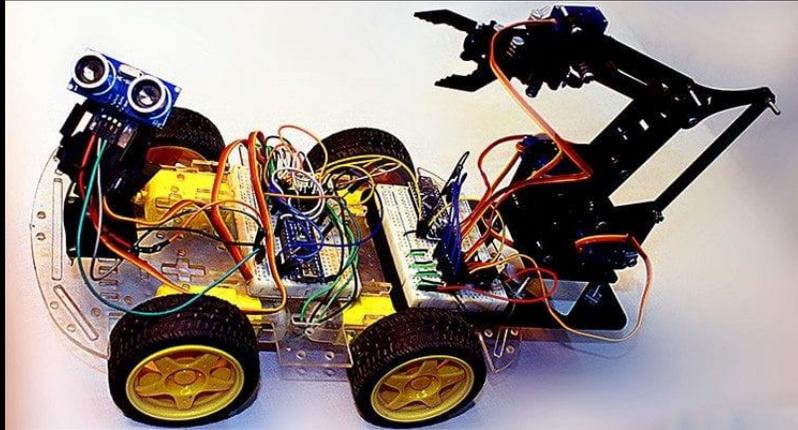
Arduino

Was ist ein Arduino?



- Open-source Plattform mit **Microcontroller** zur Ansteuerung von Hardware-Komponenten
- Perfekt für **Verbindung**:
Software (Programmieren) <-> Hardware (Elektronikkurs)
- Zahlreiche Pins (Input und Output)
- **Hardware-Komponenten** (LEDs, Motoren, Abstandssensoren, ...) mit Pins verbinden und ...
- ... Verhalten der Pins **programmieren**
- **Programmiersprache**: C++ (mit einigen Mods), ähnlich wie C#
- Arduino ist *kein* Computer (wie z.B. Raspberry Pi) und benötigt Computer um programmiert zu werden
- Danach läuft eigenständig (mit Batterie)

Was kann man damit machen?

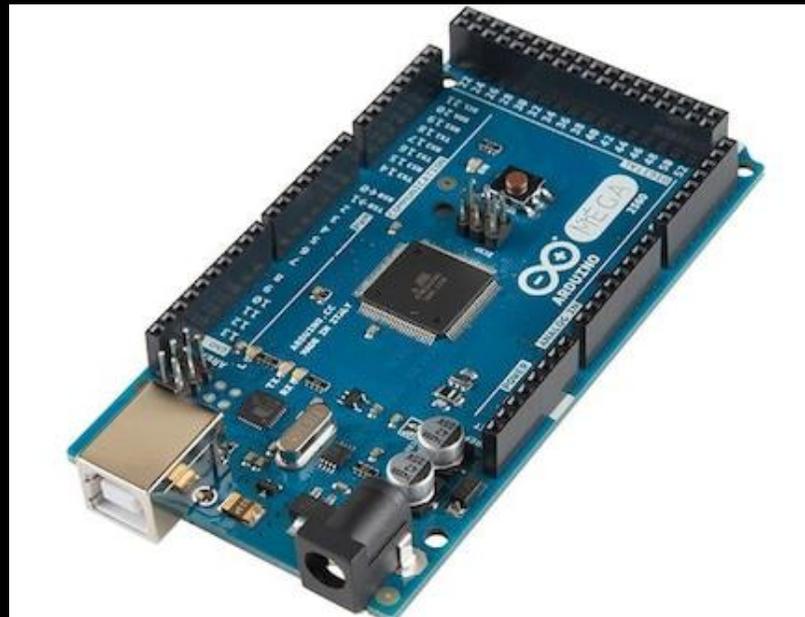


Arduino Typen

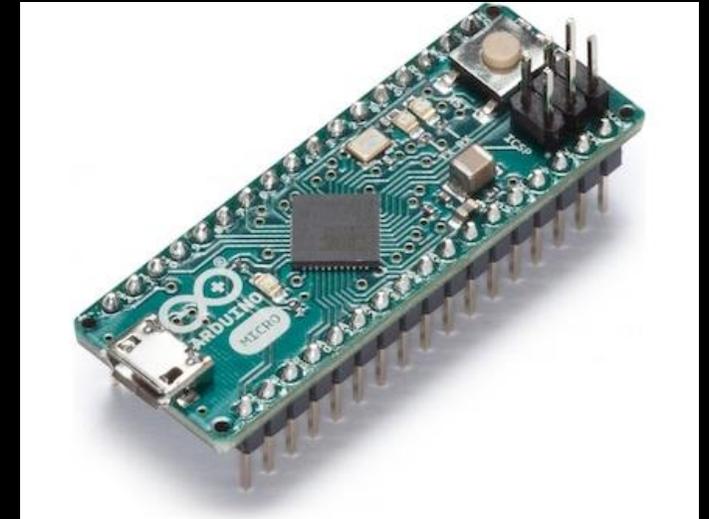
- Viele unterschiedliche Typen
- Verschiedene Grössen, unterschiedlich viele Pins, ...
- Unterschiedliche Chips: verschiedene Vor- & Nachteile je nach Projekt
- Äquivalente Produkte von anderen Herstellern



Uno



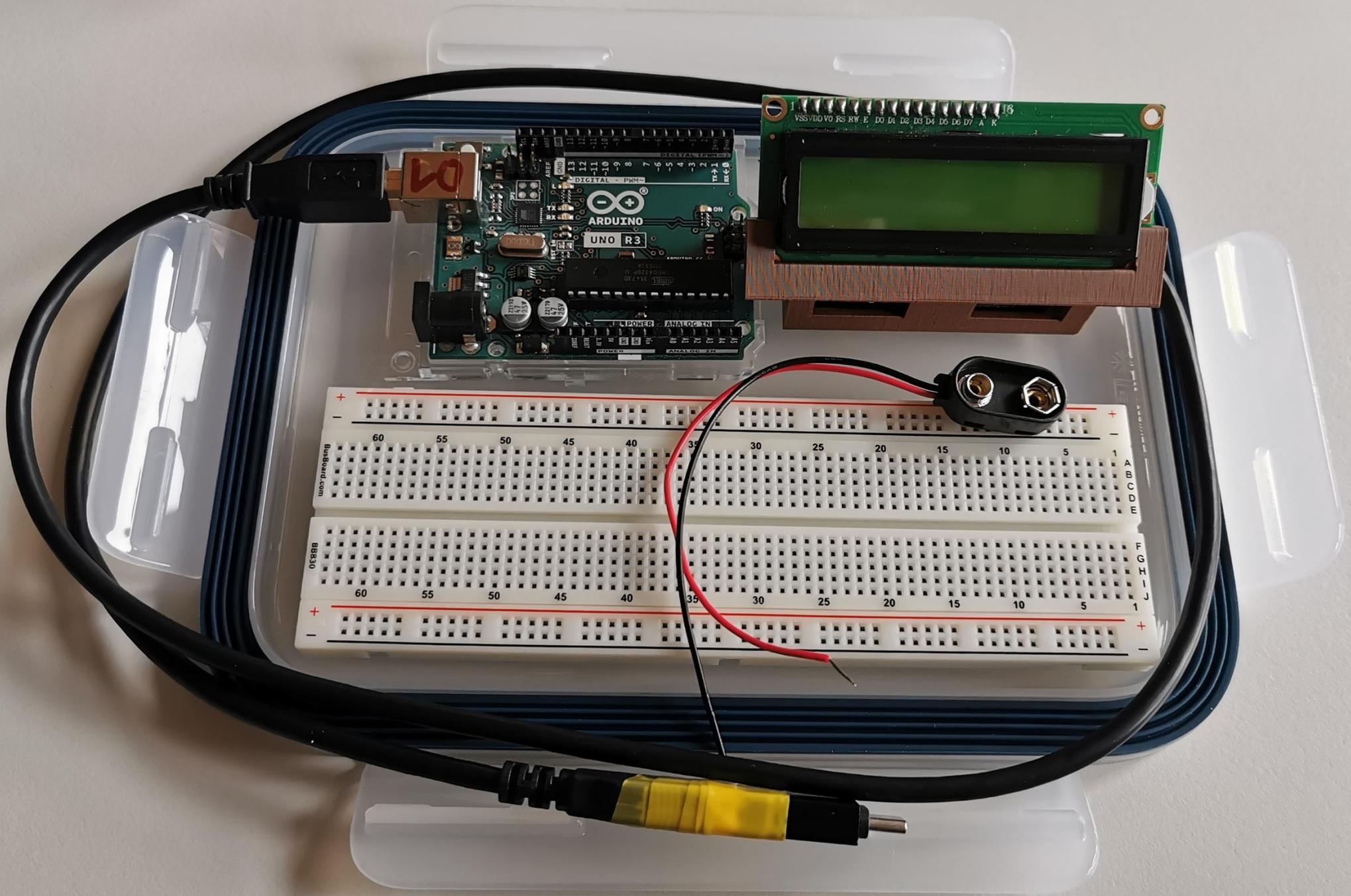
Mega



Micro

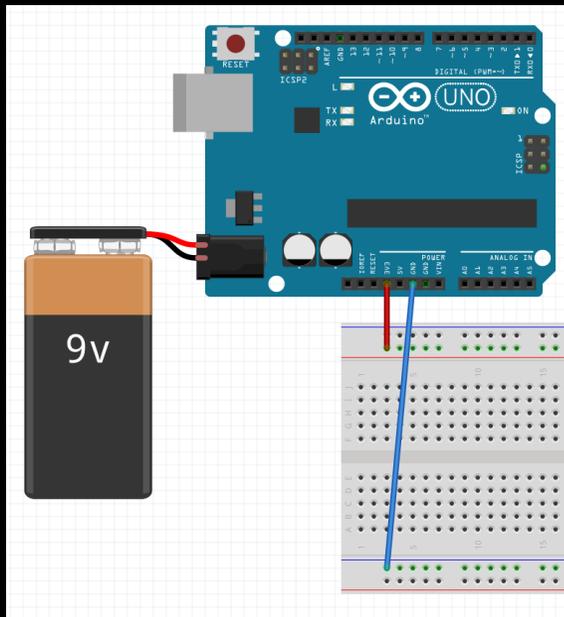
Unser Setup

- Persönliches Arduino-Kit, bestehend aus zwei **Tupperwares**:
 - T1: Arduino, Breadboard, Bildschirm mit Halterung, Batterie, USB-Kabel
 - T2: Kabel, wichtigste Komponenten (Widerstände, Knöpfe, ...)
-> siehe genaue Stückliste
- Achtung: ist Eigentum der KSR
- Kann aber für fairen Preis am Schluss gekauft werden.
- **Zwischen Lektionen kannst du dein Kit nachhause nehmen. Es ist aber deine Verantwortung, dass du dieses jeweils dabei hast!**

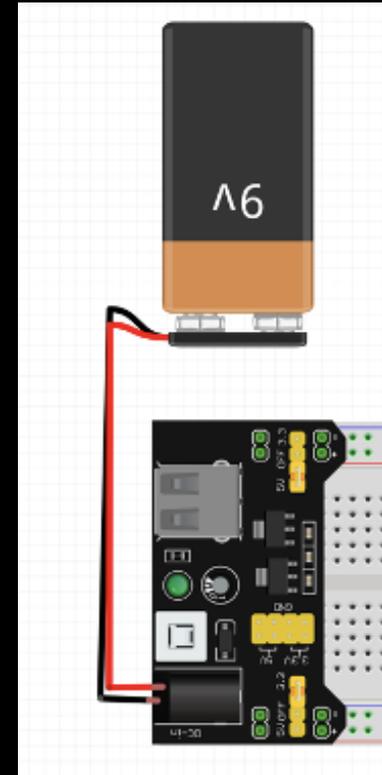


Spannungsversorgung

- 1) Über Arduino
 - Geringer Stromverbrauch
 - Batterie unnötig, wenn mit Computer verbunden
 - Tipp: + oben. - unten



- 2) Separate Stromversorgung
 - Batterie
 - oder Netzteil



Arduino Programmieren

- Programmiersprache: C++ (ähnlich wie C#)
- Programmierumgebung: Arduino IDE 2.0
- Grundstruktur Code:

Arduino auswählen

Setup-Funktion: Wird am Anfang genau 1x ausgeführt.

Loop: Wird danach in Dauerschleife wiederholt.

```
1 void setup() {
2 // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7 // put your main code here, to run repeatedly:
8
9 }
10
```

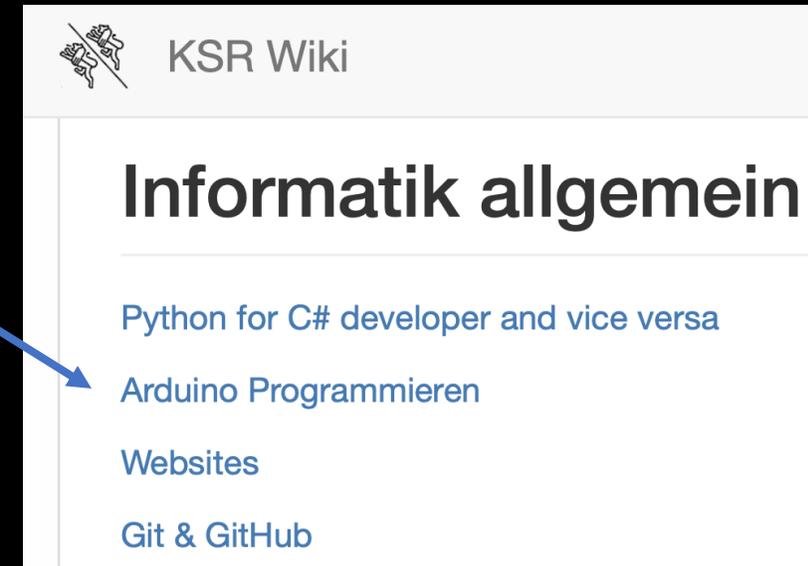
Hello World

```
1 void setup() {  
2     Serial.begin(9600); // Öffnet die serielle Schnittstelle und stellt  
3     // die Datenrate auf 9600 Bit/s ein.  
4 }  
5  
6 void loop() {  
7     Serial.println("Hello World"); // Print in Konsole mit Zeilenumbruch am Ende  
8     delay(1000);  
9 }
```

- Serial.println: sendet Text über USB-Kabel an Computer, kann dort in Konsole anzeigen.
- Konsole öffnen: Tools / Serial Monitor (oder Shortcut)

Auftrag & Aufgaben B

- Studiere Tutorial «Arduino Programmieren» auf Wiki bis und mit Kapitel 3 «Grundlagen Programmieren»
- Aufgaben B (auf Wiki)



Pins

- **Digitale & Analoge Pin**, beide können als **Input oder Output** dienen
- **Digitale Pins:**
 - LOW (0, 0V) oder HIGH (1, 3.3V)
 - Muss Modus (OUTPUT, INPUT, INPUT_PULLDOWN für Buttons) festlegen
 - Bsp. LED zum leuchten bringen
(Digital Pin 2 -> Vorwiderstand -> LED -> GND)
- **Analoge Pins:**
 - Alle Werte möglichen zwischen LOW (0) und HIGH (255)

```
1  #define PIN_LED 2
2
3  void setup() {
4      pinMode(PIN_LED, OUTPUT);
5  }
6
7  void loop() {
8      digitalWrite(PIN_LED, HIGH);
9  }
```

Auftrag & Aufgaben C

- Studiere Tutorial «Arduino Programmieren» auf Wiki Kapitel «4 Hardwarenahes Programmieren»
- Aufgaben C (auf Wiki)

