

Arduino Grundkurs

Material

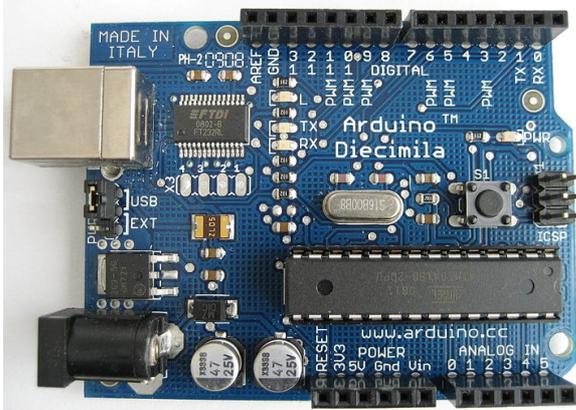
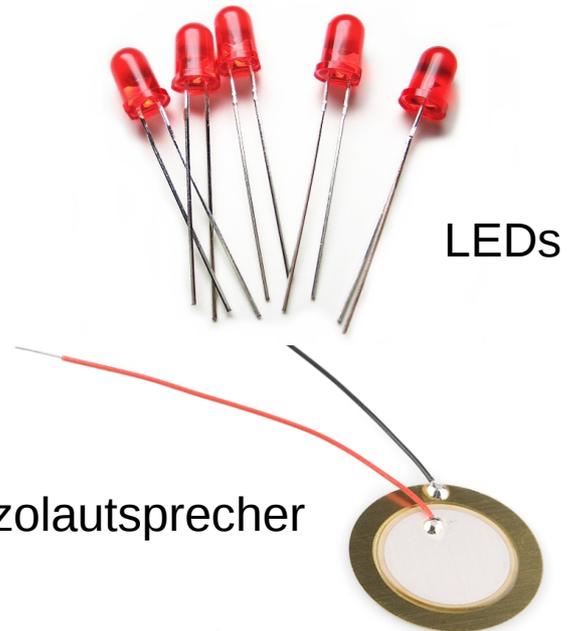


Bild: CC-BY 2.0 Remko van Dokkum

Arduino



LEDs

Piezolautsprecher

1. Mikrocontroller „Hello World“

Bring eine LED zum Blinken 1s ein / 1s aus usw.

2. Lichtmorsen - SOS

Erzeuge Morsezeichen mit deiner LED

SOS = ... - - - ...

Punkt = 0.2s

Strich = 0.5s

Wiederhole das SOC Signal mit 5s Pause

3. Schreibe eine Funktion für das Morsezeichen „S“ und eine für „O“

4. Erzeuge Töne mit dem Piezolautsprecher

Note A = 440Hz

Versuche die anderen Noten der Tonleiter

5. Ton-Morsen

Sende SOS mit Tönen

Ersetze die Funktionen für die LEDs mit solchen, die Töne ausgeben.

6. Lese Noten von der seriellen Schnittstelle und spiele sie



Arduino Grundkurs

International Morse Code

1. The length of a dot is one unit.
2. A dash is three units.
3. The space between parts of the same letter is one unit.
4. The space between letters is three units.
5. The space between words is seven units.

A ● ■
B ■ ● ● ●
C ■ ● ■ ●
D ■ ● ●
E ●
F ● ● ■ ●
G ■ ■ ●
H ● ● ● ●
I ● ●
J ● ■ ■ ■
K ■ ● ■
L ● ■ ● ●
M ■ ■
N ■ ●
O ■ ■ ■
P ● ■ ■ ●
Q ■ ■ ● ■
R ● ■ ●
S ● ● ●
T ■

U ● ● ■
V ● ● ● ■
W ● ■ ■
X ■ ● ● ■
Y ■ ● ■ ■
Z ■ ■ ● ●

1 ● ■ ■ ■ ■
2 ● ● ■ ■ ■
3 ● ● ● ■ ■
4 ● ● ● ● ■
5 ● ● ● ● ●
6 ■ ● ● ● ●
7 ■ ■ ● ● ●
8 ■ ■ ■ ● ●
9 ■ ■ ■ ■ ●
0 ■ ■ ■ ■ ■

Arduino Grundkurs

Frequenzen und Noten

220.000000	A3	a
233.081881	A#3	ais
246.941651	B3	h
261.625565	C4	c'
277.182631	C#4	cis'
293.664768	D4	d'
311.126984	D#4	dis'
329.627557	E4	e'
349.228231	F4	f'
369.994423	F#4	fis'
391.995436	G4	g'
415.304698	G#4	gis'
440.000000	A4	a'
466.163762	A#4	ais'
493.883301	B4	h'
523.251131	C5	c''
554.365262	C#5	cis''
587.329536	D5	d''



Arduino Lichtkommunikation

Material

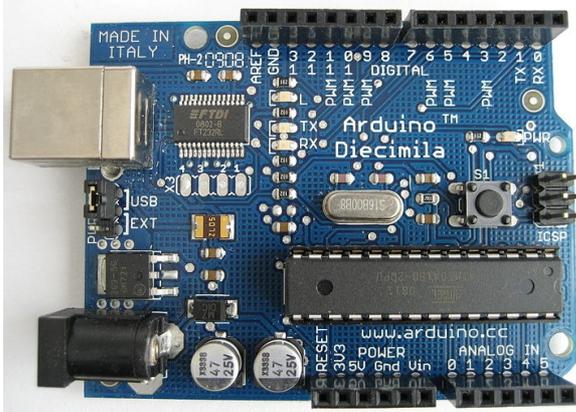
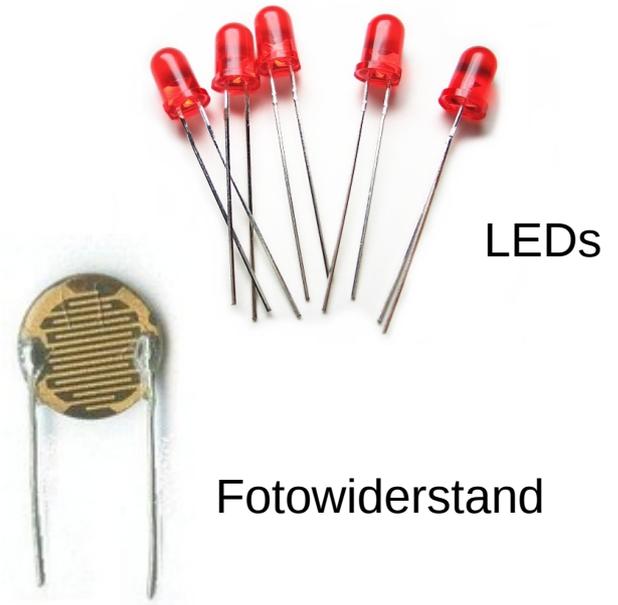


Bild: CC-BY 2.0 Remko van Dokkum



LEDs

Fotowiderstand

2x Arduino

- 1. Du empfängst Zeichen von der seriellen Schnittstelle**
Diese zerlegst du in bits
Die bits sendest du als 00100011 Folge wieder zurück an den PC
- 2. Du sendest die Bitfolge auf die LED**
Um den Start zu markieren sendest du immer am Anfang eine 1
- 3. Dein Partner empfängt Licht mit dem Fotowiderstand**
Macht euch Gedanken wie man hell/dunkel unterscheiden kann
Versucht analogRead() und digitalRead()
- 4. Dein Partner versucht aus den empfangenen Blinksignalen die originale 00100011 Folge wieder zu rekonstruieren**
- 5. Versucht aus dieser Folge wieder das entsprechende Zeichen zu erzeugen**

Arduino Lichtkommunikation

ASCII – Zeichen und ihre Nummern

32	SP	64	@	96	`
33	!	65	A	97	a
34	"	66	B	98	b
35	#	67	C	99	c
36	\$	68	D	100	d
37	%	69	E	101	e
38	&	70	F	102	f
39	'	71	G	103	g
40	(72	H	104	h
41)	73	I	105	i
42	*	74	J	106	j
43	+	75	K	107	k
44	,	76	L	108	l
45	-	77	M	109	m
46	.	78	N	110	n
47	/	79	O	111	o
48	0	80	P	112	p
49	1	81	Q	113	q
50	2	82	R	114	r
51	3	83	S	115	s
52	4	84	T	116	t
53	5	85	U	117	u
54	6	86	V	118	v
55	7	87	W	119	w
56	8	88	X	120	x
57	9	89	Y	121	y
58	:	90	Z	122	z
59	;	91	[123	{
60	<	92	\	124	
61	=	93]	125	}
62	>	94	^	126	~
63	?	95	_	127	DEL



Arduino Glockenspiel

Material

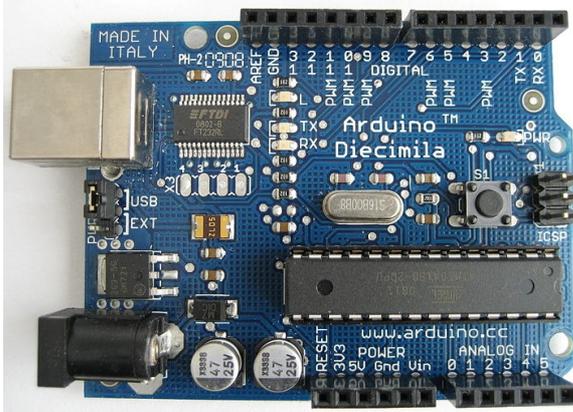


Bild: CC-BY 2.0 Remko van Dokkum



Servos



Glockenspiel

Arduino

1. Servos montieren

- Klebe das erste Servo aufrecht stehend auf die Grundplatte
 - Klebe das zweite Servo liegend auf die drehbare Scheibe des ersten
 - Befestige den Stab an der Drehscheibe des zweiten Servos
- So kannst du den Schlägel nach rechts und links und außerdem auf und ab bewegen.

2. Schließe die Servos an den Arduino an.

3. Installiere die Bibliothek für Servo-Steuerung

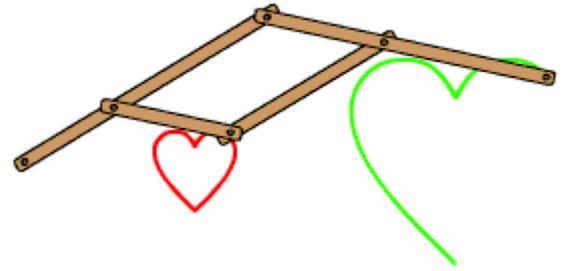
4. Finde durch Ausprobieren heraus, wie die jeden Ton anschlagen kannst.

5. Schreibe eine Funktion, die die jeweilige Note anschlägt

6. Lese Noten von der seriellen Schnittstelle und spiele sie

7. Programmiere eine Melodie

Arduino Pantograf



Material

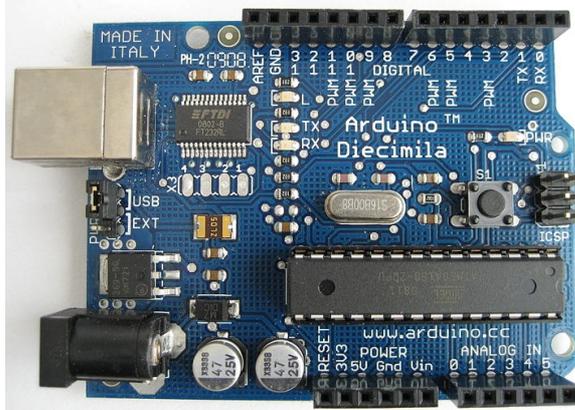


Bild: CC-BY 2.0 Remko van Dokkum

Arduino



Servos



Potentiometer

1. **Montiere die Potentiometer auf der vorbereiteten Grundplatte und den beiden Armen.**
2. **SchlieÙe die Potentiometer an den Arduino an.**
Benutze die Analogeingänge A0 und A1 um die Schleiferposition (mittlerer Pin am Potentiometer) zu messen.
Die äußeren Pins des Potis schließt du an GND bzw, +5V am Arduino an.
Mit dem Befehl `analogRead(..)` kannst du nun die Stellung des Potis als Zahlenwert zwischen 0 und 1023 auslesen.
3. **Montiere die Servos auf der Grundplatte und den vorbereiteten beiden Servoarmen.**
4. **Befestige den Stift an dem äußeren Servoarm**
5. **SchlieÙe die Servos an den Arduino an.**
6. **Installiere die Servo Bibliothek**
7. **Abhängig von der gemessenen Position der Potis steuerst du nun die Servos so, dass sie die gleiche Bewegung machen wie die Arme mit den Potis.**

Arduino Balancierplatte

Material

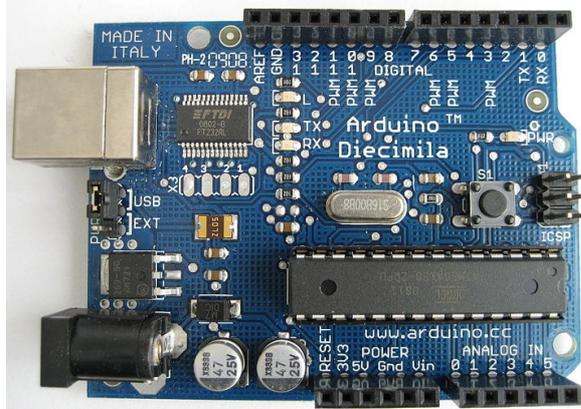


Bild: CC-BY 2.0 Remko van Dokkum



Servos



Potentiometer

Arduino

- 1. Montiere die Servos auf der vorbereiteten Grundplatte so dass die Balancierplatte in X- und Y-Richtung von den Servos gekippt werden kann.**
- 2. SchlieÙe die Potentiometer an den Arduino an.**
Benutze die Analogeingänge A0 und A1 um die Schleiferposition (mittlerer Pin am Potentiometer) zu messen.
Die äußeren Pins des Potis schließt du an GND bzw. +5V am Arduino an.
Mit dem Befehl `analogRead(..)` kannst du nun die Stellung des Potis als Zahlenwert zwischen 0 und 1023 auslesen.
- 3. SchlieÙe die Servos an den Arduino an.**
- 4. Installiere die Servo Bibliothek**
- 5. Abhängig von der gemessenen Position der Potis steuerst du nun die Servos so, dass die Balancierplatte bewegt wird.**
Versuche mit einem Partner die Kugel in das Loch der Platte zu steuern, in dem jeder mit einem Potentiometer eins der beiden Servos bewegt.

Erweiterungen: Statt der Servos könnte man das auch über sie serielle Schnittstelle steuern – oder über ein Webinterface – oder ...

Arduino Abstandsmesser

Material

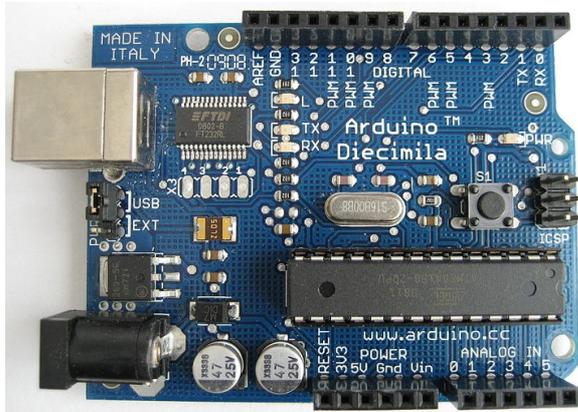


Bild: CC-BY 2.0 Remko van Dokkum

Arduino

Ultraschallsensor



Display

- 1. Schließe den Ultraschallsensor an den Arduino an.**
- 2. Steuere den Ultraschallsensor so an, wie im beigelegten Beispiel-Programm und schreibe den Wert auf die serielle Schnittstelle.**
 - Probiere aus, wie genau die Messung ist
 - Schreiben auf die serielle Schnittstelle: `Serial.println(...)`
 - Kalibriere dein Messgerät durch Vergleichen mit dem Gliedermaßstab
- 3. Schreibe einen Text auf das kleine Display**
 - Benutze hierfür die Display Bibliothek (Siehe Beispielprogramm)
- 4. Baue dir einen tragbaren Entfernungsmesser**
 - Benutze das Messprogramm von oben
 - Benutze das Anzeige Programm von oben und modifiziere es so, dass es die gemessenen Werte nicht mehr auf der seriellen Schnittstelle ausgibt sondern der LC-Anzeige.
 - Verwende eine Batterie, so dass du deinen Entfernungsmesser auch wirklich herumtragen kannst.

Arduino Abstandsmesser

Ultraschallsensor



```
// Ultraschall Sensor Library

const int pingPin = A1;
const int start_signal = A0;

long Abstand() {
    long duration, cm;

    pinMode(pingPin, OUTPUT); // Pins vorbereiten
    pinMode(start_signal, OUTPUT);
    digitalWrite(start_signal, HIGH);
    delayMicroseconds(20);

    digitalWrite(start_signal, LOW); // Starte Messung mit fallender Flanke
    digitalWrite(pingPin, LOW);
    delayMicroseconds(2);
    digitalWrite(pingPin, HIGH);
    delayMicroseconds(5);
    digitalWrite(pingPin, LOW);
    pinMode(pingPin, INPUT);

    duration = pulseIn(pingPin, HIGH); // Messung der Verzögerung bis Echo
    cm = duration / 29 / 2 ;

    Serial.print(cm); // Nur für Debug
    Serial.println("cm");
    delay(50);
    return cm;
}
```

Arduino Kompass

Material

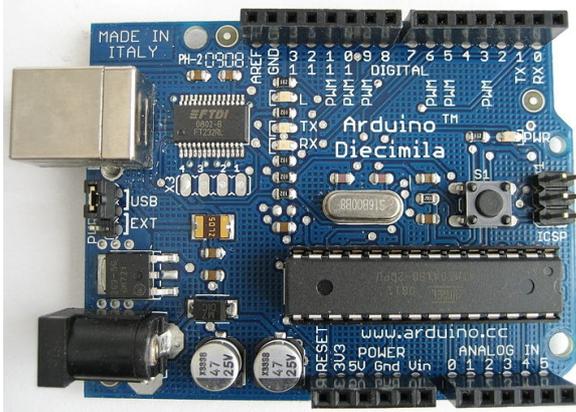


Bild: CC-BY 2.0 Remko van Dokkum

Arduino

Magnetfeldsensor



Display

- 1. Schließe den Magnetfeldsensor an den Arduino an.**
- 2. Steuere den Magnetfeldsensor so an, wie im beigelegten Beispiel-Programm und schreibe den Wert auf die serielle Schnittstelle.**
Probiere aus, wie genau die Messung ist
Schreiben auf die serielle Schnittstelle: `Serial.println(...)`
Vergleiche mit dem mechanischen Kompass
- 3. Schreibe einen Text auf das kleine Display**
Benutze hierfür die Display Bibliothek (Siehe Beispielprogramm)
- 4. Baue dir einen tragbaren Kompass**
Benutze das Messprogramm von oben
Benutze das Anzeige Programm von oben und modifiziere es so, dass es die gemessenen Werte nicht mehr auf der seriellen Schnittstelle ausgibt sondern der LC-Anzeige.
Verwende eine Batterie, so dass du deinen Kompass auch wirklich herumtragen kannst.

Arduino Schüttelwürfel

Material

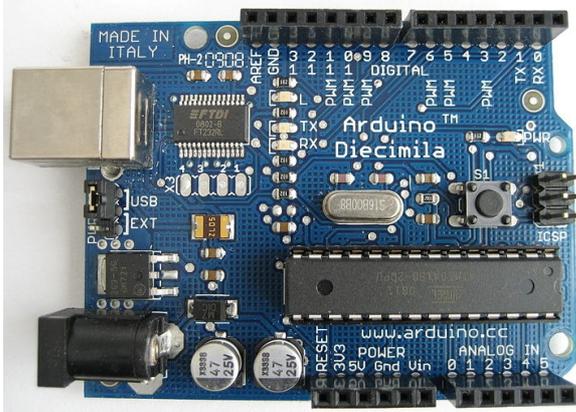
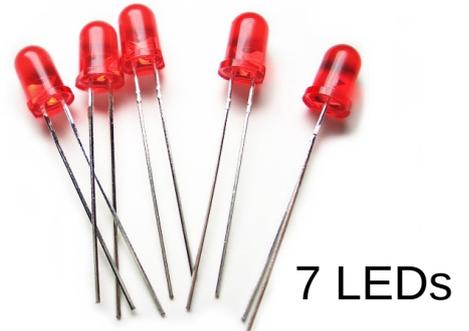


Bild: CC-BY 2.0 Remko van Dokkum

Arduino



7 LEDs



Tilt-Sensor

- 1. Schließe den Tilt-Sensor an den Arduino an.**
- 2. Benutze eine Steckbrett, um die 7 LEDs in der typischen Würfelform anzuordnen**
Schließe die LEDs über 150 Ohm Vorwiderstände an 7 Ausgänge des Arduino an.
- 3. Schreibe ein Programm, dass immer dann aktiv wird, wenn der Tilt-Sensor Schütteln erkennt.**
Das Programm soll dann eine Zufallszahl zwischen 1 und 6 ermitteln, das die entsprechenden LEDs de Würfelaugen zum Leuchten bringt.

Arduino Reaktionsspiel

Material

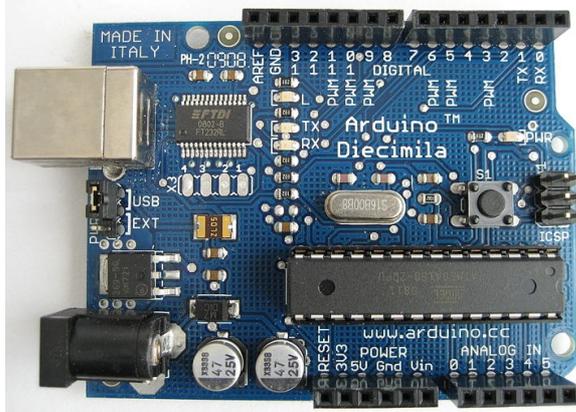
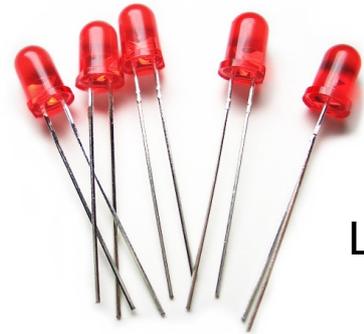


Bild: CC-BY 2.0 Remko van Dokkum

Arduino



LEDs



Taster

1. **Stecke die 6 LEDs in beliebiger Anordnung mit 150 Ohm Vorwiderständen auf das Steckbrett.**
2. **Neben jede LED steckst du einen der Tastschalter mit Pull-up Widerstand**
3. **Schließe die LEDs und Schalter an den Arduino an**
 - Die 6 LEDs über Vorwiderstände
 - Die 6 Pull-up Widerstände an +5V
 - Die Schalter an Signalpins des Arduino
4. **Programmieraufgabe**
 - Nun lässt du in zufälliger Folge jeweils eine der LEDs aufleuchten
 - Der Spieler muss versuchen, möglichst schnell den zur LED passenden Taster zu drücken.
 - Der Arduino misst die Zeit bis zum Druck der Taste.
 - Nach 10 LEDs wird die Gesamtreaktionszeit ausgewertet.
 - Ist sie klein genug, dann blinken die LEDs - wenn nicht, dann leuchten sie dauerhaft.